Summer 4-2019

KOLMOGOROV-SMIRNOV TYPE TESTS UNDER SPATIAL CORRELATIONS

WENJUN ZHENG
UTHealth School of Public Health

KOLMOGOROV-SMIRNOV TYPE TESTS UNDER SPATIAL CORRELATIONS

by

WENJUN ZHENG, BEC

APPROVED:

DEJIAN LAI, PHD

J. MICHAEL SWINT, PHD

MOMIAO XIONG, PHD

DEAN, THE UNIVERSITY OF TEXAS
SCHOOL OF PUBLIC HEALTH

DEDICATION

To my families

KOLMOGOROV-SMIRNOV TYPE TEST WITH SPATIAL ADJUSTMENT VIA MORAN'S I

by

WENJUN ZHENG
BEc, Jiangxi University of Finance and Economics, 2014

Presented to the Faculty of The University of Texas

School of Public Health

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS
SCHOOL OF PUBLIC HEALTH
Houston, Texas
April, 2019

ACKNOWLEDGEMENTS

I would like to thank Dr. Dejian Lai, my mentor and chair of my committee, who has been supportive and generous since the day I joined the school of public health. His mentorship made this dissertation possible. I would also like to thank Dr. K. Lance Gould for providing the access to his trial and my committee members: Dr. J. Micheal Swaint and Dr. Momiao Xiong for heling me through the process.

I would like to express my deepest gratitude to my families and friends. Without their support, I would not have the courage to purse my career in the first place. This dissertation would not have been possible without their warm love, continued patience, and endless support.

KOLMOGOROV-SMIRNOV TYPE TESTS UNDER SPATIAL CORRELATIONS

Wenjun Zheng, BEc, PhD
The University of Texas
School of Public Health, 2019

Dissertation Chair: Dejian Lai, PhD

Kolmogorov-Smirnov test is a non-parametric hypothesis test that measures the probability of deviations, that the interested univariate random variable is drawn from a pre-specified distribution (one-sample KS) or has the same distribution as a second random variable (two-sample KS). The test is based on the measure of the supremum (greatest) distance between an empirical distribution function (EDF) and a pre-specified cumulative distribution function (CDF) or the largest distance between two EDFs. KS test has been widely adopted in statistical analysis due to its virtue of more general assumptions compared to parametric test like t-test. In addition, the p-value derived from the KS test is more robust and distribution-free for a large class of random variables. However, the fundamental assumption of independence is usually overlooked and may potentially cause inaccurate inferences. The KS test in its original form assumes the interested random variable to be independently distributed while it's not true in a lot of nature datasets, especially when we are dealing with more complicated situations like imgage analysis, geostatistical which may involve spatial dependence.

I proposed a modified KS test with adjustment via spatial correlation. The dissertation concerns the following three aims. First, I conducted a systematical review on the KS test, the Cramer von Mise test, the Anderson-Darling test and the Chi-square test and evaluate their performance under normal distributions, Weibull distributions and multinomial distributions. In the review, I also studied how these tests perform when random variables are correlated. Second, I proposed a modified KS test that corrects the bias in estimating CDF/EDF when spatial dependence exists and calculate the informative sample size. Finally, I conducted a

revisit analysis of coronary flow reserve and pixel distribution of coronary flow capacity by Kolmogorov-Smirnov with spatial correction to evaluate the efficiency of dipyridamole and regadenoson.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

Introduction

Kolmogorov-Smirnov Test

Andrey Kolmogorov (1903-1987) was a mathematician born in the Soviet Union. His study covered areas of probability theory, topology, intuitionistic logic, turbulence, classical mechanics, algorithmic information theory and computational complexity (Stephens, 1992). Among his prominent contributions to many fields of mathematics and statistics, the Kolmogorov statistic is a commonly-used statistic to test the equality of an empirical distribution function (EDF) and a given cumulative distribution function (CDF) (Stephens, 1992). In the year of 1933, Kolmogorov published a short but landmark paper, in which he formally defined empirical distribution function (EDF), in the *Italian Giornale dell'Istituto Italiano degli Attuari* (Kolmogorov, 1933).

To define the empirical distribution function, let set $x_1, x_2, \ldots, x_{i-1}, x_i, \ldots, x_n$ be the realizations of random variables X having the $F(x) = pr(X < x)$. Similarly, let $y_1, y_2, \ldots, y_{i-1}, y_i, \ldots, y_m$ be the realizations of random variables Y having the $G(y) = pr(Y < y)$. Put

$$\epsilon(x) = I(x_i \leq x)$$

Then the EDF of X is defined as:

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} \epsilon(x_i)$$

It could be easily seen that the EDF $F_n(x)$ is the portion of $x_1, x_2, \ldots, x_{i-1}, x_i, \ldots, x_n$ of X below x. It comes naturally to ask how close EDF $F_n(x)$ is to its corresponding CDF $F_(x)$. To answer this question, Kolmogorov studied and gave the asymptotic distribution of EDF. This led to the definition of Kolmogorov statistic (or Kolmogorov-Smirnov statistic) D and the distribution of D given finite sample size n was derived (Kolmogorov, 1933).

$$D = \sup_x |F_n(x) - F(x)|$$

where the $\sup_x$ is the supremum function defined as the least upper bound of all absolute distance sets between the EDF $F_n(x)$ and CDF $F_(x)$.

Kolmogorov's sutdent Smirnov extended Kolmogorov's original one-sample KS statistic into the two sample version of the KS statistic, which is defined as (N. V. Smirnov, 1939)

$$D_{n,m} = \sup_x |F_n(x) - G_m(x)|,$$

where $G_m(x)$ is the EDF of random variable Y.

The Kolmogorov-Smirnov Type Statistics and Its Variants

Later, Smirnov proposed the Cramer-von Mises statistic (CvM statistic) ω^2, which can be viewed as an extension of KS statistic, based on Cramer's work in 1928 and von Mises's work in 1931 (von Mises, 1931; N. V. Smirnov, 1937; Mises, 1928). In which, Smirnov also found the asymptotic distribution of ω^2, in the form of a sum of weighted chi-squared variables.

$$\omega^2 = \int_{-\infty}^{\infty} [F_n(x) - F(x)]^2 f(x) dx$$

Anderson commented on the distribution of the two-sample CvM statistic, which is defined as followed (Anderson, 1962).

$$\omega_2^2 = \frac{nm}{n+m} \int_{-\infty}^{\infty} [F_n(x) - G_m(x)]^2 dH(x)$$

Where H(x) is the empirical function of the combination of two samples together,

$$H(x) = \frac{nF(x) + mG(x)}{n+m}$$

Anderson also worked out the expected value, $E(\omega_2^2)$, and variance, $var(\omega_2^2)$, of the asymptotic distribution of ω_2^2.

$$E(\omega_2^2) = \tfrac{1}{6} + \tfrac{1}{6(m+n)}$$
$$Var(\omega_2^2) = \tfrac{1}{45} \times \tfrac{m+n+1}{(m+n)^2} \times \tfrac{4mn(m+n)-3(m^2+n^2)-2mn}{4mn}$$

The way to use the asymptotic distribution of two-sample CvM statistic for hypothesis testing will be based on the standardized statistic W^2 defined as

$$W^2 = \frac{\omega_2^2 - E(\omega_2^2)}{[45Var(\omega_2^2)]^{\frac{1}{2}}} + \frac{1}{6}$$

Reject H_0 if $W^2 > W_\alpha^2$. The critical value W_α^2 at the significance level of $\alpha = 0.01$ and $\alpha = 0.05$ has been worked out by Anderson (Anderson, 1962).

Extension of Kolmogorov-Smirnov Type Statistic on Discontinuous Distribution

Researchers extended the discrete CvM into the scope of k-sample CvM for discrete distribution or continuous distribution being grouped. Consider ordered observations $Z_1^*, \ldots Z_L^*$ as the L distinct pooled sample of X and Y (Brown, 1982, 1994; Lockhart, Spinelli, & Stephens, 2007).

Let

$$k_1 = n$$

$$k_2 = m$$

The two-sample CvM for discrete distribution is defined as followed

$$W_d^2 = \sum_{i=1}^{2} k_i \sum_{j=1}^{L} (S_{ij} - T_{ij})^2 p_j$$

where for ordered observations $Z_1^*, \ldots Z_L^*$, p_j is the probability of falling into group j. S_{1j} is the number of observations in X not greater than Z_j^* , S_{2j} is the number of observations in Y not greater than Z_j^*.

$$T_i j = k_i \sum_{i=1}^{j} p_l$$

and $(n+m)p_j$ is the number of observations of a pooled sample of X and Y coinciding with z_j^*. The asymptotic distribution has been worked out by Sun. If $W_d^2 > \omega_{(d,\alpha)}^2$, then we reject H_0.

By modifying the weight factor of CvM statistic, T. W. Anderson and D. A. Darling (1952) proposed the Anderson Darling statistic (AD statistic) A.

$$A^2 = n \int_{-\infty}^{\infty} \frac{[F_n(x) - F(x)]^2}{F(x)[1 - F(x)]} f(x) dx$$

Later in 1987, F.W. Scholz and M. A. Stephens proposed an extension for k-sample AD statistic. In this paper, we only used the two-sample version which has the form as followed. (Scholz & Stephens, 1987)

$$A^2_{n,m} = \frac{mn}{N} \int_{-\infty}^{\infty} \frac{[F_m(x) - G_n(x)]^2}{H_N(x)[1 - H_N(x)]} dH_N(x)$$

where

$$H_N(x) = \frac{mF_m(x) + nG_n(x)}{N}, \text{ with } N = m + n$$

The asymptotic distribution of $A^2_{n,m}$ under H_0 is

$$A^2_{n,m} = \sum_{j=1}^{\infty} \frac{1}{j(j+1)} \chi^2_j$$

where χ^2_j are independent chi-squared random variables with 1 degree of freedom. In order to compute the statistic given sample X and Y. Given ordered observations $Z_1, \ldots, Z_N$ as the pooled sample of X and Y. Formulas on how to calculate the AD statistic under the assumption that samples were from continuous and discrete parent population is given as followed,

$$A^2_{n,m} = \frac{1}{N} \sum_{i=1}^{2} \frac{1}{k_i} \sum_{j}^{N-1} \frac{(NM_{ij} - jk_i)^2}{j(N-j)}$$

where M_{1j} is the number of observations in X not greater than Z_j and M_{2j} is the number of observations in Y not greater than Z_j and

$$k_1 = n$$

$$k_2 = m$$

In order to deal with the situation when X and Y are from the discrete population, or from the continuous population but being grouped, let ordered observations $Z^*_1, \ldots, Z^*_L$ as the L

distinct pooled sample of X and Y. AD statistic under discrete setting is defined as follows.

$$A^2_{n,m} = \sum_{i=1}^{2} \frac{1}{k_i} \sum_{j}^{L-1} \frac{l_j}{N} \frac{(NM_{ij} - B_j k_i)^2}{B_j(N - B_j)}$$

Where f_{1j} be the number of observations in X coinciding with Z^*_j, f_{2j} be the number of observations in Y coinciding with Z^*_j and let

$$l_j = f_{1j} + f_{2j}$$

$$M_{ij} = f_{i1} + \cdots + f_{ij}$$

$$B_j = l_1 + \cdots + l_j$$

Pettitt worked out an approximation formula to calculate the variance of $A^2_{n,m}$. (Pettitt & Stephens, 1977)

$$var(A^2_{n,m}) = \frac{2(\pi^2 - 9)}{3} \times (1 - \frac{3.1}{N})$$

The test procedure for AD test is as follows,

1. Compute $A^2_{n,m}$ by the formula in respect to its parent distribution

2. Compute

$$T_N = \frac{(A^2_{n,m} - 1)}{\sigma_N}$$

where

$$\sigma^2_N = var(A^2_{n,m})$$

3. Reject H_0 if

$$T_N > t_\alpha$$

The critical value t_α has been derived by Pettitt (Pettitt & Stephens, 1977) and confirmed through the Monte Carlo simulation by Scholz (Scholz & Stephens, 1987).

Choulakian extended the Cramer-von Mises statistic into the scope for discrete distributions or continuous distributions being grouped. (Choulakian, Lockhart, & Stephens, 1994) Consider $x_1^*, \ldots, x_L^*$ as the ordered L-distinct sample of X.

$$W_2^2 = \frac{1}{n}\sum_{j=1}^{L}(S_j - T_j)^2 p_j$$

Where o_j is the number of observations coinciding with x_j^*, then

$$S_j = \sum_{i=1}^{j} o_i$$
$$T_j = \sum_{i=1}^{j} N p_i$$

Reject the null hypothesis if the statistic is larger than the critical values of W_2^2.

On the other hand, the Chi-squared test is also a popular test that has been widely adopted. Similar to the EDF based tests, Chi-squared tests also has One-sample and Two- sample version.

$$\chi^2 = \sum_{i=1}^{2^k} \frac{(O_i - E_i)^2}{E_i}$$

From the formula above, we can see that χ^2 statistic is the summation of deviations of the observed number and expected number in i_{th} bin divided by the expected number in i_{th} bin. One sample χ^2 statistic is asymptotically distributed in chi-squared distribution with k-1 degrees of freedom.

$$\chi_2^2 = \sum_{i=1}^{k} \frac{(K_1 O_{1i} - K_2 O_{2i})^2}{O_{1i} + O_{2i}}$$
$$K_1 = \sqrt{n_2/n_1}$$
$$K_2 = \sqrt{n_1/n_2}$$

Asymptotically, the two-sample statistic χ^2 follows a chi-square distribution with $(k-c)$ degrees of freedom where k is the number of non-empty bins and $c = 1$ if the sample sizes of X and Y are equal, $c = 0$ otherwise. Critical value will be $\chi^2_{(1-\alpha,k-c)}$, at the nominal level of α.

The chi-squared test used here has two versions, one for continuous data and one for discrete data. The discrete data one is directly from the popular package stats and has been reported to be reliable. (Arnold & Emerson, 2011) The continuous one is from a categorized version chi-squared test, the grouping algorithm in which the test is reported to be one of the optimization algorithms. (D'Agostino & Stephens, 1986)

1. If sample size n $\leq$ 35, then the number of bins

$$B_n = \lfloor \frac{n}{5} \rfloor$$

   B_n which is the largest integer not greater than n/5. Therefore to ensure there's at least 5 samples in each bin

2. If sample size $n > 35$, then the number of bins

$$B_n = \lfloor 1.88 \times n^{\frac{2}{5}} \rfloor$$

   which is the largest integer not greater than $1.88 \times n^{\frac{2}{5}}$.

3. Cut the range of data into n bins $(x_1, x_{\lfloor \frac{n}{b_n} \rfloor}), (x_{1+\lfloor \frac{n}{b_n} \rfloor}, x_{\frac{n}{b_n}}), \ldots, (x_{1+\lfloor \frac{n}{b_n} \rfloor}, x_n)$

4. Test if the number of samples in each bin same as expected. Reject if such statistic is large than the critical value.

Since Kolmogorov's introduction of the EDF based test, Kolmogorov-Smirnov test has been increasingly popular in analyzing data from clinical trials. By the virtue of its relatively less strict assumptions on the dataset to be applied, e.g. its distribution-free properties. The

nature advantage of being generally more powerful than χ^2 test (Pettitt & Stephens, 1977). The KS test has been widely appreciated for test the distribution equality.

In many ways, the KS test seems like a safe choice and popular for spatial statistics analysis. Researchers have been applying it for testing the equality of sample distributions of realizations across map (Berman, 1986; P. Clifford, Richardson, & Hémon, 1989) . It is also common to see KS test being applied to test the histogram frequency similarities and for discriminate images (Demidenko, 2004).

However, the independence assumption is one of the very fundamental and easily overlooked assumptions of a statistical model. Without taking care of the effect of correlations between samples, positive linear correlations may result in the underestimation of type I error of the KS test and vice versa (Weiss, 1978) .

Kolmogorov-Smirnov test has been used to discriminate image difference. Published papers have confirmed the efficiency of KS test being applied in the imaging process and histogram analysis (Lampariello, 2000). Lim showed that the KS test has relatively higher power compared to Wilcoxon and t-test when the variation is relatively large (Lim & Jang, 2002). Geman used KS test for discriminating homogeneous maps by pixel gray levels distribution (Geman, Geman, Graffigne, & Dong, 1990). The interpretation ability rendered its favourable position in clinical fields. Clinically, published reports suggested that KS test were valid for analyzing MR scans comparison (Chen, Sans, Bogdanov, & Weissleder, 2006; F. Baselice, 2017; Rajan, Dekker, & Sijbers, 2014). Kipritidis used KS test for CT/PET scans and Brook applied histogram analysis with KS for spectral CT scans to evaluate the artifacts reduction (Kipritidis et al., 2016; Brook et al., 2012) .

Measure of Dependence

Directly measure the relationship between variables is relatively hard and usually inaccessible. One of the statistical tools involving dependence is the measure of correlation. Correlation coefficient has been used to measure correlations; it is usually being standardized from -1 to 1. A value of 1 of correlation coefficient means a perfect positive correlation between samples and vice versa. A weak correlation is indicated by a correlation coefficient with a value close to 0.

Linear Correlation Coefficient

The linear correlation measures the correlation relationship between samples linearly. Published reports have introduced multiple linear correlation coefficients includes Pearson's r, Spearman's ρ, intra-class correlation coefficient and other coefficients for different purposes and situations.

Pearson's r correlation coefficient

The most widely used measure of correlation in statistics is Pearson's r. It is a coefficient measuring the correlation introduced by Karl Pearson in 1895 in Proceeding of the Royal Society of London with his landmark paper Note on the regression and inheritance in the case of two parents (Pearson, 1895).

Give a population of n subjects with bivariate outcome X and Y for each subject in the population. Originally, Pearson's r for X and Y is defined as

$$r = \frac{\sum_{i=1}^{n}(y_i - \mu_X)(y_i - \mu_Y)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_X)}\sqrt{\sum_{i=1}^{n}(y_i - \mu_Y)}}$$

Where μ_x and μ_y are the mean values for X and Y, respectively.

Pearson's r is the most popular correlation coefficient due to the reasons that it is easy to calculate and interpret and it is invariant to linear transform. However, sound inference of linear correlation between two random variables dependends on strict assumptions, such as continuous and normally distributed. When, unfortunately, random variables do not meet the these assumptions, though one can still calculate the Pearson's r, it is hard to interpret and thus not be informative.

Intra-class Correlation Coefficient

Similar to Pearson's r, intra-class correlation (ICC) is a measure of how good one variable resembles the other. It is commonly used to measure the agreement for continuously paired outcomes. Ronald Fisher (1925) first proposed the original idea of ICC in Statistical Methods for Research Workers (Fisher, 1925).

Consider two paired random variables $X = x_1, x_2, \ldots, x_i$ and $Y = y_1, y_2, \ldots, y_i$, Fisher's original ICC was defined as

$$r = \frac{1}{ns^2} \sum_{i=1}^{n} (x_i - \mu)(y_i - \mu)$$

where

$$\begin{aligned} \mu &= \frac{1}{2n} \sum_{i=1}^{n} (x_i + y_i), \\ s^2 &= \frac{1}{2n} \sum_{i=1}^{n} [(x_i - \mu)^2 + (y_i - \mu)^2] \end{aligned}$$

Later in 1934, Fisher introduced a form of ICC based on analysis of variance model (Fisher, 1934). More recently in 1980, Donner introduced a form of ICC within the scope of the linear

mixed model that has been more popular credited to its virtue of parsimony (Donner & Koval, 1980).

Consider a linear mixed-effects model with n subjects from k groups. Let y_{ij} denotes i^{th} subject from j^{th} group,

$$Y_{ij} = \mu + \beta_i + \varepsilon_{ij}; i = 1, 2, \ldots, n; j = 1, 2, \ldots, k$$

where

$$\begin{aligned} \beta_i &\sim N(0,\ \sigma_\beta^2) \\ \varepsilon_{ij} &\sim N(0,\ \sigma^2) \end{aligned}$$

It is easy to derive that ICC in the linear mixed-effects model is defined as the ratio of variance within the group and total variance

$$\rho_{ICC} = \frac{\sigma_{beta}^2}{\sigma_{beta}^2 + \sigma^2}$$

Under the linear mixed-effects model setting, ICC and Pearson's r are comparable as standardized coefficients that measure the linear correlation between random variables when k = 2. ICC has advantages over Pearson's r due to following factors, 1. Unlike the calculation of Pearson's r, where each variable is centered and scaled by its own mean and standard deviation, ICC calculated mean based on pooled population. When the interested variables are paired, a mean from the pooled population would be more reasonable. 2. When ICC is calculated from the linear mixed model, it can be applied to cases where there are more than 2 groups, whereas Pearson's r can only measure the correlation of bivariate variables. One of the common negative

aspects of linear correlation coefficient that need to be noticed is that they may suffer from assumptions of linear correlation and normal distribution of interested random variables.

Non-linear Correlation Coefficient

The scope of non-linear correlation coefficient includes a variety type of measures on the correlation in samples. Similarly to the linear correlation coefficient, most of the values from standardized non-linear correlation coefficients range from -1 to 1. A non-linear correlation coefficient of 1 is interpreted as the perfect correlating of samples, and vice versa.

Spearman's ρ

Spearman's ρ is another popular correlation coefficient introduced by Charles Spearman (1904). (Spearman, 1904) Spearman published the article *The Proof and Measurement of Association between Two Things* in the American Journal of Psychology as "a commencement at attempting to remedy". Unlike linear correlation coefficients concerning with continuous outcomes, ρ is calculated through ranks of random variables which make it available to the discrete or grouped outcome.

Give a population of sample size n with random variables X and Y. Spearman defined the correlation coefficient by

$$\rho = \frac{cov(R_x, R_y)}{\sigma_x \sigma_y}$$

where

- R_x and R_y are ranks of random variables X and Y,
- $cov(R_x, R_y)$ is the covariance of R_x and R_y,

- σ_x and σ_y are standard deviations of ranks R_x and R_y.

It is worth noticing that there is another popular form of ρ as

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}$$

where

$d_i = R_x - R_y$ is the difference between each pair of ranks.

Kendall's τ

Kendall's τ is a correlation coefficient measuring the non-linear correlation among bivariate random variables. It is commonly used when a researcher is curious about the non-parametric property. (Kendall, 1938) Kendall's τ was first introduced by Maurice Kendall (1938) titled *A New Measure of Rank Correlation* in Biometrika. Consider a population of n subject with bivariate random variables X and Y. Kendall's τ is defined by

$$\tau = \frac{n_c - n_d}{\frac{1}{2}n(n-1)}$$

Any pair of (x_i, y_i) and (x_j, y_j), where $i \neq j$, are concordant if $X_i < X_j$ and $Y_i < Y_j$ or if $X_i > X_j$ and $Y_i > Y_j$. Pairs are considered discordant if $X_i < X_j$ and $Y_i > Y_j$ or if $X_i > X_j$ and $Y_i < Y_j$. The number of concordant and discordant pairs are denoted as n_c and n_d, respectively.

Spatial Correlation Coefficient

In the setting of spatial statistics, the correlation relationship between samples are not only in values but also depend on the spatial locations. Assume realizations from each location were sampled from the same parent distribution, the correlation relathionship between each

realizations in each location were the same as the correlation of a variable with itself through space. Therefore, the correlation relationship under spatial setting is usually referred as the spatial autocorrelation.

Before the introduction of spatial autocorrelation, firstly we need to define the spatial data. There are three main categories of spatial data (N. Cressie, 1992):

- Point pattern:

  - When a spatial process is observed at a set of locations and the locations themselves are of interest. e.g. galaxies in space

- Geostatistical data:

  - When a spatial process that varies continuously is observed only at a few points e.g. mineral concentrations at various drilling locations

- Lattice data:

  - When a spatial process is observed on a regular or irregular grid. Often this arises due to aggregation of some sort, e.g. averages over a pixel in an image

Many spatial correlation coefficients have been proposed to evaluate the spatial autocorrelation relationship. In order to define the spatial relationship mathematically, a good amount of correlation functions has been introduced as followed.

Moran's I

In the field of spatial statistics, things got more complicated when researchers are trying to calculate the correlation coefficient. Because there are random variables and there is also distance between each pair of subjects. To account for the effect of distance, Patrick Moran

(1950) proposed a spatial autocorrelation coefficient in his paper of Notes on Continuous Stochastic Phenomena in Biometrika. (Moran, 1950)

Give a population of N spatial subjects with random variable X, w_{ij} denotes the preset weight between i^{th} and j^{th} subjects. Moran's I is defined as

$$I = \frac{N}{S} \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij}(x_i - \mu)(x_j - \mu)}{\sum_{j=1}^{N}(x_i - \mu)^2}$$

Where

$$\begin{aligned} S &= \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \\ \mu &= E(X) \end{aligned}$$

Later in 1995, a local Moran's I was introduced by Anselin (Anselin, 1995). After the introduction of local Moran's I, researchers are able to analyze not only the global spatial autocorrelation of the geostatistical data but also be provided with a tool to analyze the local spatial relationship.

$$I_i = \frac{\sum_{j=1, j\neq i}^{N} w_{ij}(x_i - \mu)(x_j - \mu)}{\frac{\sum_{j=1, j\neq i}^{N} w_{ij}}{N-1} - \mu^2}$$

It is easy to show that under large sample, the global Moran's I is the average of local Moran's I,

$$\frac{\sum_{i=1}^{N} I_i}{N} = I$$

Different from global Moran's I, the value of local Moran's I is calculated for each observation unit. Different patterns or processes may occur in different parts of the region, local Moran's I provide us tool to precisely identify regions that have serious spatial autocorrelation influence.

D Statistic

Similar to the rank statistic for traditional samples, Walter proposed a statistic to account for the auto-correlation relationship.

Let $Y_i = y_1, y_2, \ldots, y_n$ be realizations in the map location $s_1, s_2, \ldots, s_n$. The D statistic is defined as followed,

$$D = \sum_{i=1,\ldots,n} \sum_{j=1,2,\ldots,n,i=j} w_{ij} h(rank(x_i), rank(x_j))$$

where the w_{ij} is the weight function. The weight function w_{ij} may be the inverse distance function or the neighboring weight function.

Dejian showed the asymptotic distribution of the standardized D statistic, which is defined as D statistic subtract mean and divided by its standard deviation. (Lai, 1997) However, the standardized D statistic ranges from $-\infty \text{to} \infty$ and therefore not be able to be directly used for comparing the autocorrelation relationship in different maps.

Cariovascular Disease and Nuclear Stress Test

Cardiovascular disease

Cardiovascular disease (CVD) generally refers to conditions that involve narrowed or blocked blood vessels that can lead to a heart attack, chest pain (angina) or stroke. It is an umbrella term that commonly includes the coronary artery disease (CAD), cerebrovascular disease, peripheral arterial disease, rheumatic and congenital heart diseases and venous thromboembolism.(Stewart, Manmathan, & Wilkinson, 2017) CVD is the top killer in the US that accounts for more than 836,000 deaths in 2018. The deaths caused by CVD accounts for 1 of every 3 deaths in the US

and is more than the deaths caused by all forms of cancer combined. Among the total deaths caused by CVD, coronary heart disease(CHD) or coronary artery disease accounts for more than 40% of total CVD deaths and is the leading cause of CVD. Published reports projected that in the year of 2018, about 720,000 Americans had a new coronary event and half of them will have recurrent coronary events. (AHA guideline 2018)

CAD is usually caused by the plaque builds up in cardio arteries. As plaque builds up in the arteries of a person with heart disease, the inside of the arteries begins to narrow, which lessens or blocks the flow of blood. Plaques can also rupture and when they do a blood clot can form on the plaque, blocking the flow of blood. Over time, CAD can weaken the heart muscle. This may lead to heart failure, a serious condition where the heart can't pump blood the way that it should, or an irregular heartbeat, or arrhythmia, also can develop. The amount of damage to the heart muscle is positively correlated with the time untreated.

Risk factors of CVD are the use of tobacco, unhealthy diet habits, physical inactivity, obesity, Cholesterol, and psychosocial stress. However, the WHO estimated that about 75% of the total cases of CVD is preventable.(technical report series, 2003) The time of diagnosis of a premature coronary event is essential to prevent CVD deaths. The diagnosis strategy includes electrocardiogram (ECG), echocardiogram, stress test, cardiac catheterization and angiogram, and heart scan.

Electrocardiogram (ECG)

The nature behind the ECG theory is that the heart beats are stimulated by electrical impulses that are generated by certain cells in heart. To record the heart rhythm, electrical impulses were recorded. Then the heart rhythm may be indirectly computed. ECG is a common test to use for diagnosis of heart problems and monitor heart health status by recording the electrical signal. During the ECG test, sensors will be placed on the chest or limbs of the patients. Electrical signal will be collected by the sensors and report in almost simultaneous results. ECG is a popular diagnostic strategy for CAD as its nice property of non-invasive and able to record heart

activity continuously.(Liang et al., 2017) However, due to the natural limitation, ECG may only record electrical signals.(Gulamhusein et al., 1982)

Echocardiogram

An echocardiogram is a test that uses ultrasound waves to produce heart images. The natural theory of the echocardiogram is that the sensor may receive reflected ultrasound signals transmitted through various locations on the chest wall. An echocardiogram image is able to provide physicians with a comprehensive and detailed image of the whole heart and in continuous time. However, the quality of the image may be affected by various factors and may suffer from poor quality or reproductive issue.(Gottdiener, 2003)

Cardiac catheterization and angiogram

Cardiac catheterization is an invasive strategy for diagnosis of CAD. The invasive strategy means that different from the non-invasive diagnosis method, cardiac catheterization involves putting sensors directly into the heart vessels. To perform the catheterization, a thin, hollow tube is implemented to a large blood vessel that leads to heart.(Swan et al., 1970) Then it records the blood flow. Usually, an angiogram will be done simultaneously and provide an x-ray image of heart for physicians. The advantage of cardiac catheterization is that the process let the physician analyze the blood flow in heart and cardiac angiogram in real time. However, published reports claimed multiple risk factors such as chemically diagnosed acute renal dysfunction(Rich & Crecelius, 1990) and minor problems like bruises, feel of itchy or hives or sick in stomachs.(Kern et al., 2006; Cosman, Arthur, & Natarajan, 2011) Cardiac catheterization is a direct and accurate way to evaluate the heart's function. (de Bruyne et al., 1988)

Stress test and heart scan

A stress test, by its name, is a test that helps physicians to understand how the heart responds to external stress. Usually, a stress test is carried out through obtaining the heart activity in rest compare to in exercise. A common way of the activity form is to ask patients to run in a treadmill or pedaling on a stationary bicycle. Throughout the exercise stress test, patients are

attached with several sensors on the chest, arm and other places on the body to measure the hearts activity. Usually, ECG, breathing, blood pressure, heart rhythm will be recorded for the diagnosis purpose. During the exercise pressure, heart is required to pump more blood and therefore physicians may learn the function of the heart. The exercise stress test is popular due to the simplicity to implement, however, it may lack generality for patients who cannot exercise or the heartbeat did not increase enough with exercise. The alternative of exercise stress test are the nuclear stress test and combined nuclear-exercise stress test.(Lette et al., 1995; Dowsley et al., 2013; Dahan et al., 2002)

Nuclear Stress Test

To account for the needs of stress test for patients without the ability to do the exercise on pedaling machine or heart rate did not go up enough, a nuclear stress test may be done instead.

To evaluate the ability that heart responds to stress, we may involve an invasive strategy, such as coronary angiography, as well as non-invasive strategy such as positron emission tomography (PET)/ computed tomography (CT). Published reports find the non-invasive strategy to be both efficient and accurate. (Danad et al., 2017; Raff, Gallagher, O'Neill, & Goldstein, 2005)

Myocardial perfusion PET is a non-invasive imaging tool for diagnosis of cardiovascular disease.(Carli et al., 2007) In order to take the rest image, patients were given a dose of radiotracer. After a suitable waiting period to ensure proper distribution of the radiotracer, a PET image is taken for rest image. it is a non-invasive way to take photos of the blood flow in your heart. To take the stress PET scan, a medication, for example, adenosine, will be administered. it helps open coronary arteries and causes more blood to flow and simulates the effect of exercise for patients who cannot exercise on a treadmill. Then the image will be taken again as the in stress condition. The nuclear imaging process provides a strategy that quantifies the absolute values of myocardial blood flow. In addition, with the absolute myocardial values, it is possible

to use certain statistical methods to assist the diagnosis process and improve sensitivity.(Cremer, Hachamovitch, & Tamarappoo, 2014)

The medication that typically used for nuclear stress test includes adenosine and regadenoson. Dipyridamole was first introduced in 1959 as an antianginal medication and was used for vasodilator stress imaging after proved to have vasodilator properties (Picano, 1989). Later, adenosine was introduced as an alternative to dipyridamole in 1994 (Cerqueira, Verani, Schwaiger, Heo, & Iskandrian, 1994). In 2005, an adenosine A2A receptor agonist was developed as regadenoson.(Hendel et al., 2005) Dipyridamole, adenosine, and regadenoson served as alternatives to each other and there were trade-offs and arguments in terms of cost, efficiency and timing protocol.(Johnson & Gould, 2015; Vasu et al., 2013; Pijls & van Lokien X Nunen, 2015; Gibbs & Lip, 1998; Goudarzi, Fukushima, Bravo, Merrill, & Bengel, 2011; Bravo, Pozios, & Abraham, 2012)

Attenuation correction

Attenuation is a condition when the coincidence events were not recorded because of their absorption in the body or other reasons. In a nuclear stress test that produces scans for rest and stress, attenuation correction (AC) is commonly involved to reduce the effects of attenuation and to ensure better alignment.

Coronary flow reserve and physiology beyond it

Myocardial blood flow

In the nuclear stress test, physicians were able to track a consistent portion in the left ventricle continuously. With the help of PET/CT, we were able to measure the myocardial blood flow (MBF) quantitatively in ml/min/g. By comparing the absolute difference or ratio of MBF for patients in rest and MBF for patients in stress, physicians could evaluate the hearts function and diagnosis for any abnormal condition.

Coronary flow reserve

Coronary flow reserve (CFR) is a relative value of stress and rest myocardial blood flow. The concept of CFR was firstly introduced by Gould et al. in 1974.(K. Lance Gould, Lipscomb, & Hamilton, 1974) The introduction of CFR provided a quantitative measurement to evaluate the ability of the heart to pump blood increasingly when the body demands it. Mathematically, it is calculated as the ratio of MBF in stress and MBF in rest.

$$CFR = \frac{MBF_{instress}}{MBF_{inrest}}$$

MBF and CFR are effective tools that help physicians understand how the heart functions and respond to outside pressure.(Klocke & Lee, 2011) Published reports suggested that the absolute myocardial perfusion analysis outperformed the relative analysis of myocardial perfusion.(Wichmann et al., 2015) In order to have a more comprehensive diagnosis method to follow. The concept of coronary flow capacity which compared both absolute and relative value of myocardial perfusion is proposed.

Coronary flow capacity (CFC)

In order to integrate the CFR with absolute blood flow, a new concept of was approved by the Food and Drug Administration (FDA) on September 22, 2017. The approval was based on the comprehensive scientific review from 2012 to 2017. Several published reports(See Gould 2018) validated the concept and proved its effects to be treat as a biomarker for CAD diagnosis.

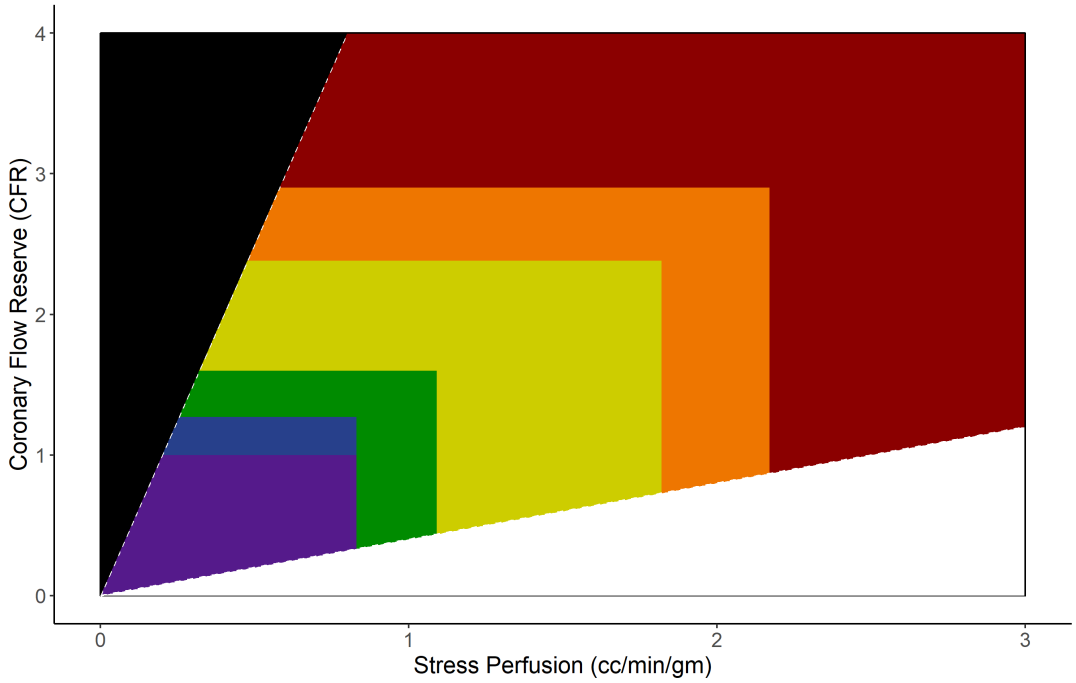


Figure 1.1: CFC Scatter Plot of CFR versus Absolute Stress Flow

CFC	CFR	Stress perfusion	Color Code
Excellent	$CFR > 2.9$	$perfusion > 2.17$	Red
Typical	$2.9 \geq CFR > 2.38$	$2.17 \geq perfusion > 1.82$	Orange
Mildly reduced	$2.38 \geq CFR > 1.6$	$1.82 \geq perfusion > 1.09$	Yellow
Moderately reduced	$1.6 \geq CFR > 1.27$	$1.09 \geq perfusion > 0.83$	Green
Severely reduced	$1.27 \geq CFR > 1$	$0.83 \geq perfusion$	Blue
Myocardinal steal	$CFR < 1$	$0.83 \geq perfusion$	Purple

Table 1.1: Coronary flow capacity

From the table we know that when CFR is larger than 2.9 (ml/g/min) or stress perfusion greater than 2.17 then the CFC is coded as excellent and the color code is red, when the CFR from 2.38 to 2.9 or the perfusion is from 1.82 to 2.17 then the CFC is coded as typical and the color code is orange, when the CFR is from 1.6 to 2.38 or the stress perfusion from 1.09 to 1.82 then the CFC is coded as mildly reduced and color code is yellow, when the CFR is from 1.27 to 1.6 or the perfusion from 0.83 to 1.09 then the CFC is recorded as moderately reduced and the color is coded as green, when the CFR is from 1 to 1.27 or the perfusion is less than 0.83 then the CFC is coded as severely reduced and the denoting color is blue, lastly when CFR is less than 1, the CFC is coded as myocardial steal and the color code is purple. The triangle in the upper left and bottom with black and white color were the lower limit of rest flow for viability and the upper limit of clinically observed rest flow, respectively.

Methods

Data Simulation

Simulating Distribution

Having studied previous works on KS type test, I have learned more about the advantage and limitation of such kind of test. Together with other goodness of fit tests, Chi-squared, Shapiro-Wilk tests, and other popular ones, researchers are given a considerable library of tests to pick from. Though it is a good thing to be provided with varieties of methods to apply for different problems, one may find it hard to decide which methods to apply. Therefore to address such issues, I have conducted a systematic review of the performance of the original KS test, CvM test, AD test, and Chi-squared test. The assessment will be both on one sample and two sample tests.

Tests mentioned above are fall in the category of "distribution-free method" which means they are robust under different distributions. However, the virtue of "distribution-free" sometimes may cause problems. When the parameter or even the distribution of our interested random variables unknown, it is hard to estimate the sample size required for certain power of the test. Therefore, I set up an environment with manually controlled various sample sizes. To evaluate the performance of the tests, I used certain characteristics of the power of hypothesis testings mentioned above under different sample size and at significance levels of 0.05. In order to study the robustness of the above tests in the presence of dependence pattern, I generated

subjects that are linearly correlated and autocorrelated. I simulated samples from the Weibull distribution $W(\gamma, \lambda)$ with two parameters, as it is commonly being applied in survival analysis, engineering and geology, normal distribution $N(\mu, \sigma^2)$ and multinomial distribution $Mult(n, p)$. Meanwhile, Weibull distribution of shape parameter γ and scale parameter λ makes us able to control the skewness of the testing distribution.

$$f(x) = \frac{\gamma}{\lambda}(\frac{x}{\lambda})^{\gamma-1}e^{(\frac{x}{\lambda})^{\gamma}}$$
$$F(x) = 1 - e^{(-\frac{x}{\lambda})^{\gamma}}$$

It is possible for me to control the actual magnitude of the difference between the two distributions by using theoretical distributions with known parameters. Thereafter I will compare the power of above tests under certain circumstances stated as followed.

Monte Carlo simulations will be used to evaluate the statistical power of KS, CvM, AD and Chi-squared statistics. Consider random variable $X : x_1, x_2, \ldots, x_n$ from

$$W(\gamma, \lambda), where \gamma = 0.5, 1, 2, 3, 5; \lambda = 1, 2, 3$$
$$N(\mu, \sigma^2), where \mu = 0, 1, 3, 5; \sigma = 0.1, 0.5, 2$$
$$Mult(n, P)$$

where

$$P = \begin{cases} (p_1, p_2) = (0.5, 0.5), & \text{Symmetric} \\ (p_1, p_2) = (0.1, 0.9), & \text{Heavily Skewed} \\ (p_1, p_2) = (0.3, 0.7), & \text{Skewed} \\ (p_1, p_2, p_3, p_4, p_5) = (0.1, 0.2, 0.4, 0.2, 0.1), & \text{Symmetric} \\ (p_1, p_2, p_3, p_4, p_5) = (0.7, 0.2, 0.05, 0.03, 0.02), & \text{Skewed} \\ (p_1, p_2, p_3, p_4, p_5) = (0.3, 0.15, 0.1, 0.15, 0.3), & \text{Symmetric with Heavy Tails} \end{cases}$$

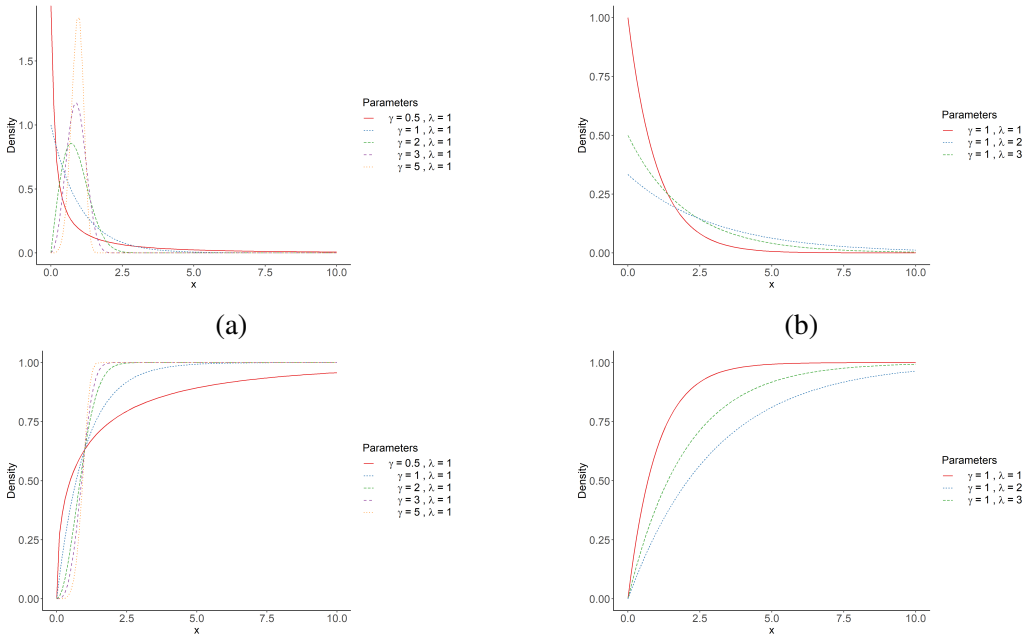


Figure 2.2: PDF and CDF for Weibull Distributions

Left column of figures are samples from distribution of N(1, 1), while right samples are from N(1, 4). Figure (a), (b) are the alternative is different variance. Figure (c), (d) are the alternative is different mean. Figure (e), (f) are the alternative is different mean.

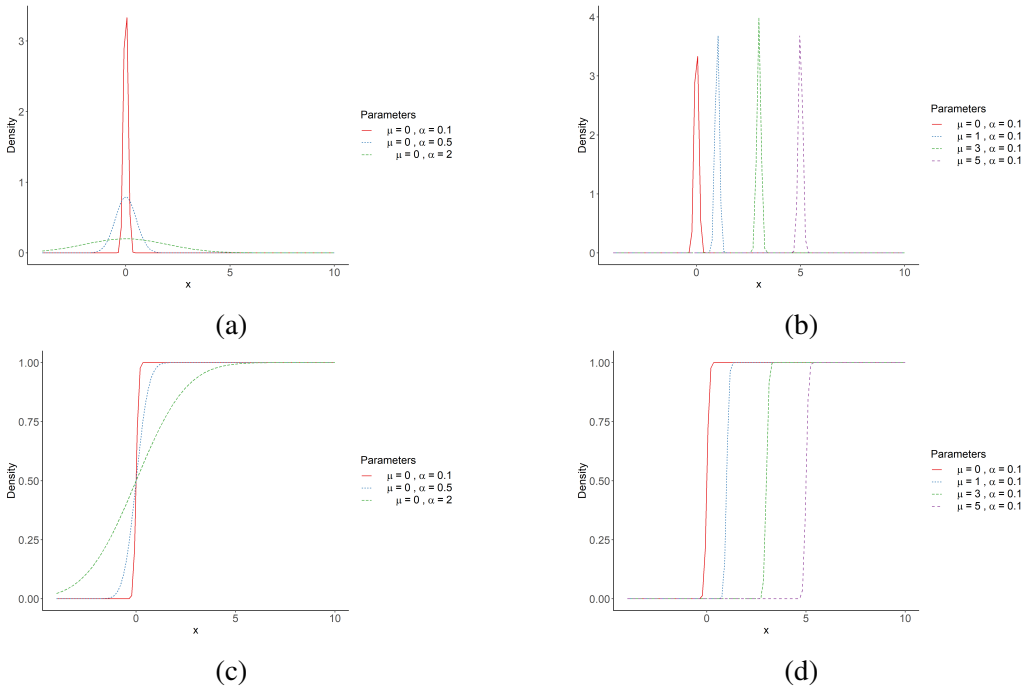


Figure 2.3: PDF and CDF for Normal Distributions

Left column of figures are samples from distribution of N(1, 1), while right samples are from N(1, 4). Figure (a), (b) are the alternative is different variance. Figure (c), (d) are the alternative is different mean. Figure (e), (f) are the alternative is different mean.

From the density and cumulative density plots of Weibull distribution, it is clear that the shape parameter controls density in tail and skewness, scale parameter only stretches or compresses on x and y-axis. Setup of parameter above ensures us to test the Weibull of heavy left tailed, minor left tailed, symmetric and right-tailed scenarios. Meanwhile, parameter ratio change in the mean of normal distribution will result in the location shift in PDF and CDF. The increase in variance will result in more flat CDF curve and PDF curve. The null and alternative hypothesis to be tested is as followed,

$$H_0 : F(x) = G(x) \tag{2.1}$$

$$H_1 : F(x) \neq G(x) \tag{2.2}$$

G(x) is the pre-specified distribution function of $W(\gamma+\Delta, \lambda+\Delta), N(\mu, \sigma^2)andMult(n, p)$, where the difference ratio Δ is

$$\Delta = 0.05, 0.1, 0.2, 0.5, 1$$

The sample size of observations generated from $W(\gamma, \lambda)$ will be $n = (10, 20, 30, 100, 500)$. Power will be obtained based on tested results of 10,000 generate samples.

Meanwhile, σ controls the shape and density of the probability curve in normal distributed data. The mean parameter μ from normal distribution shifts the entire curve while not changing shape and density distribution. Therefore, the change in σ and μ provide us an opportunity to test the performance under shape differences and location differences, or both differences.

Lastly, in the multinomial distributed data group, I had a chance to evaluate the performance of KS, CvM and AD tests when data is indeed discrete. When, unfortunately, certain parameters of the distribution were not available and we are left with no option on the table but to estimate these parameters from the sample, then results from Kolmogorov-Smirnov test will be conservative. To adjust for the effects bring by discontinuous in samples, methods were proposed to extend EDF tests on discrete data (Simpson, 1951; Crutcher, 1975; Lilliefors, 1967). Therefore, I simulated data from multinomial distribution under different conditions.

In the comparison of two-sample tests, Monte Carlo simulations will be used to evaluate the type I error and statistical power of KS, CvM, AD and Chi-squared statistics in testing if both samples are from the same certain distribution.

Correlated Realizations

Consider two random variables, X follows $W(\gamma, \lambda)$, Yfollows$W(\gamma + \Delta, \lambda + \Delta)$. To study the performance of above tests under dependency, random variables X and Y are sampled independently or in the existence of linear dependence, Pearson's $r = (-0.8.-0.5, -0.2, -0.1, 0.1, 0.2, 0.5, 0.8)$.

Sample size for random variables $X$and$Y$ will include balanced and imbalanced groups in detail as followed table.

In order to simulate correlated samples, I applied the copula method (Joe 1997). For the sake of easy computation and estimation, I chose a Gaussian copula method for its relatively high accuracy. The procedure of copula methods to simulate bivariate correlated Weibull distribution is as followed.

1. First, choose a covariance matrix Σ that reflects the correlations relationship in our targeted samples. Based on the covariance structure one would like to achieve, draw correlated samples $X_1 = (x_1^1, x_2^1, x_3^1, \ldots, x_n^1)$ and $X_2 = (x_1^2, x_2^2, x_3^2, \ldots, x_n^2)$ from standard bivariate Gaussian distribution. Therefore we may have

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim MVN \left(\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 1 & r^2 \\ r^2 & 1 \end{pmatrix} \right)$$

2. Find the CDF of X_1 and X_2 as $\phi(X_1), \phi(X_2)$.

3. In order to simulate correlated samples $Z_1 = (z_1^1, z_2^1, z_3^1, \ldots, z_n^1)$ and $Z_2 = (z_1^2, z_2^2, z_3^2, \ldots, z_n^2)$ from the targeted distribution, we find the targeted inver-CDF function as $F^{-1}(Z_1)$ and $F^{-1}(Z_2)$

4. Compute the following function and our interested correlated samples may be obtained

$$\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \begin{bmatrix} F^{-1}(\phi(X_1)) \\ F^{-1}(\phi(X_2)) \end{bmatrix}$$

There are several choices for the correlation matrix to simulate the bivariate Gaussian distribution. Rank correlation coefficients, such as Kendall's τ and Spearman's ρ, are usually preferred as they are invariant to strictly increasing transformations (Ding & Li, 2013). The linear

correlation coefficient, on the other hand, may not be invariant to non-linear transformations but have the virtue of able to be applied directly to simulate normal distribution in the first step. In addition, the trend of the correlation relationship between samples is invariant. Dithinde used a translation-based lognormal model with Pearson's r to capture the correlation structure between two hyperbolic curve-fitting parameters and have relatively well results (Dithinde, Phoon, De, & Retief, 2011). Genest report the simulation with Pearson's r measuring the correlation structure to be performing reasonably well when simulated sample size n is 50 or larger. Therefore, I applied Pearson's r to simulate the bivariate normal distribution (Genest & Rivest, 1993).

In real data analysis, we may find data to be in chaos and usually given in imbalanced sample size. For the purpose of evaluating the performance of tests under the imbalanced sample size condition, I have simulated our data in the sample size as showed in the following table.

Sample size of (X, Y)				
10, 10	20, 20	50, 50	100, 100	500, 500
10, 20	20, 50	50, 100	100, 500	
10, 50	20, 100	50, 500		
10, 100	20, 500			
10, 500				

Table 2.2: Simulation Sample Size

The performance of EDF based tests and the chi-squared test will be evaluated by their simulation results of type I error and power. Type I error and power will be analyzed from realization results of 10,000 repeated iterations.

Spatial Analysis

In previous chapters, I have discussed that the PET-CT image data is gridded spatial data in nature. In this section, I focused on the method to generate a spatial field that simulates the PET-CT image data with pre-defined auto-correlation structure. First, we need to define a few spatial statistics concepts.

Let $S : s_i \in \boldsymbol{R^d}$ be interested location in d-dimensional Euclidean space, $Z(s_i)$ can be viewed as the random process in such location s_i.The notation $z(s_i)$ is defined as a realization of such random process $Z(s_i)$. Without loss of generality, we may assume that the random process $Z(s_i)$ as followed

$$Z(s_i) = \mu + \varepsilon_i$$

Where μ is defined as the mean value of such process and the error term follows a normal distribution, $\varepsilon_i \sim N(0, \sigma^2)$. For the purpose of statistically analyzing the image data, intrinsic stationary distribution is a critical assumption for the spatial random process. The intrinsic stationery is defined as followed

$$\begin{aligned} E(Z(s+h) - Z(s)) &= 0 \\ var(Z(s+h) - Z(s)) &= 2\gamma(h) \end{aligned}$$

where h is the Euclidean distance, $2\gamma(h)$ is an important spatial statistics parameter is known as variogram and $\gamma(h)$ is the semivariogram.

Meanwhile, the second order stationary ensures the distribution of such random process not depend on the location s_i, therefore all realizations across the map were from the same

distribution.

$$E(Z(s_i)) = \mu \tag{2.3}$$

$$cov(Z(s_i + h), Z(s_i)) = C(h) \tag{2.4}$$

where $C(h)$ is the covariogram that only depend on the distance between location s_i and s_j. After $C(h)$ is defined, the autocorrelation structure of such spatial process may be determined.

With the aim of creating a positive-definite covariance structure for the spatial analysis, a valid covariance structure depend on geometry location needs to be defined. Matern (1960) constructed a few valid covariogram models in$\boldsymbol{R^d}, d > 1$. Assumed a valid isotropic covariogram structure in $\boldsymbol{R^3}$.

$$C(h) = \frac{\sigma^2(\frac{\alpha^2||h||}{2})^\nu 2K_\nu(\alpha^2||h||)}{\Gamma(\nu)}, \nu > 0$$

where K_ν is the modified Bessel function of the second kind, $||h||$ is the Euclidean distance. Specifically, $\nu = 1/2$ may yield into a special case

$$C(h) = \sigma^2 exp(-\alpha^2||h||)$$

Cholesky Decomposition Method

With knowledge of covariogram structure Σ, we were able to apply Cholesky decomposition methods to simulate valid autocorrelated data on the interested fields. (N. Cressie, 1992; Joe, 1997) In order to get the targeted simulated realizations, we decomposed the covariogram matrix with Cholesky decomposition, in which

$$\Sigma = LL'$$

Where L is a lower triangular $n \times n$ matrix. Then the targeted realizations could be obtained as

$$Z(s) = \mu + LE \tag{2.5}$$

Where E is the error term in matrix form. Note that E is from the identical independent normal distribution with zero mean and unit variance, $E \sim N(0, 1)$. By applying the Cholesky decomposition method, I was able to simulate auto-correlated spatial realizations, with pre-defined covariogram structure, from independent simulated spatial data points.

A Moran's I in Covariogram Form

With the Cholesky decomposition method from section , I was able to simulate spatially correlated realizations once the covariogram Σ structure is defined. In order to measure the spatial autocorrelation, a more general correlation coefficient is required. Moran's I has been introduced in section and considered to evaluate the degree of autocorrelation of my simulation. However, the original Moran's I was defined as a measurement for realizations, which is inaccessible before simulation. With the purpose of simulating spatially autocorrelated samples with respect to certian Moran's I. With given spatial covariogram known, I used an approximation form of Moran's I with the weighted covariogram matrix.

$$I_A = \frac{N}{W} \frac{\sum_i \sum_j w_{i,j} cov(Z(s_i), Z(s_j))}{\sum_i var(Z(s_i))}$$

where N is the sample size, $w_{i,j}$ is the weight for location s_i and s_j, $W = \sum_i \sum_j w_{i,j}$.

In order to see if I_A generates desired spatially autocorrelated samples in a given spatial space, I have run a Monte Carlo simulation with 10,000 replications. Given the valid variogram for $\boldsymbol{R^3}$,

$$C(h) = \sigma^2 exp(-\alpha^2 ||h||) \tag{2.6}$$

Samples were generated regarding given covariogram 3.17 and spatial structure stated in figure 3.21. The Moran's I in covariogram form was calculated before simulation. The Moran's I in original form for simulated samples were computed after simulation. The Moran's I in covariogram form and the simulated Moran's I were compared in plot 3.13. It shows a satisfied rate of fit.

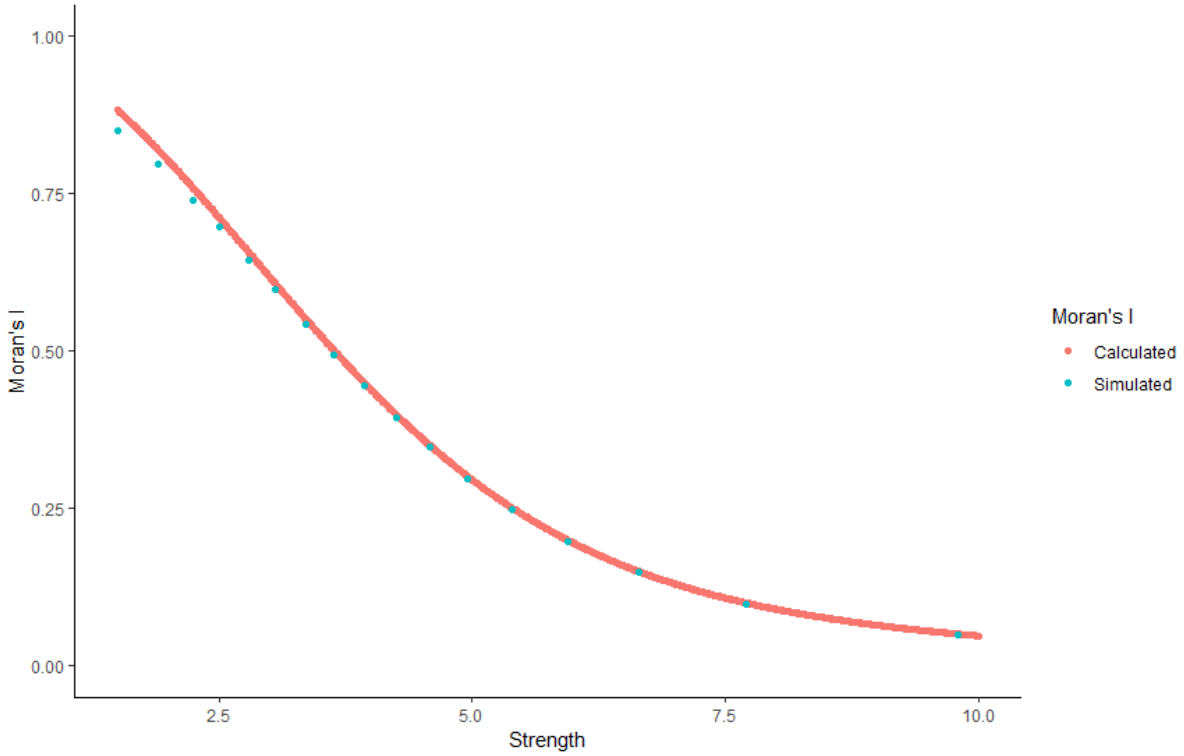


Figure 2.4: I_A vs. Simulated Moran's I

Published reports suggested that when the KS test was applied directly without adjustment on the existed spatial autocorrelation will be liberal with an underestimated p-value (Weiss, 1978). Therefore, it is reasonable for me to assume that a adjustment on the sample size may provide us a closer guess to the truth. I defined the sample size after adjustment as informative sample size.

For spatial realizations $y_1, y_2, \ldots, y_n$ of $1^{st}, 2^{nd}, \ldots, n^{th}$ locations. Notice that n is the sample size. Assume

$$Y = \mu + \varepsilon$$

where μ denotes the population mean of Y, ε is the spatially auto-correlated error term independent of μ. We may rewrite the error term in independent form ε^*, and $\varepsilon^* \sim i.i.d.N(0, \sigma^2_{\varepsilon^*})$. let

$$C(Y_i, Y_j) = \sigma^2 V^{-1}$$

then

$$Y = \mu + V^{-\frac{1}{2}}\varepsilon^*$$

where V is the identity matrix, $V = I$, if and only if Y is spatially independent.

Griffith (2005) gave that the expectation of the variance of Y is

$$E(\hat{\sigma^2_Y}) = \frac{\frac{tr(V^{-1})}{n}\sigma^2_\varepsilon}{\frac{tr(V^{-1})}{1^t V^{-1} 1} n}$$

where 1 is the $n \times 1$ matrix of 1, $tr(V^{-1})$ is the trace matrix of V^{-1}.

Then he notes that the informative sample size n^*(the equivalent number of samples without autocorrelation) is

$$n^* = \frac{tr(V^{-1})}{1^t V^{-1} 1} n$$

Griffith reported findings for an approximation of n^* when the spatial realizations Y is normally distributed given the spatial autocorrelation coefficient $\hat{\rho}$ estimated from Spatial autoregressive (SAR) models as followed

$$n^* = n \times [1 - \frac{1}{1 - \exp -1.92} \frac{n-1}{n} (1 - \exp -2.12\hat{\rho} + 0.2\sqrt{\hat{\rho}})] \tag{2.7}$$

where the KS statistic was still obtained as the supremum of the absolute distance between two EDFs.

Another KS test with adjustment for the violation of independence assumption is the ICC adjusted KS test (N. Cressie, 1992). Similar to Griffith's adjustment, the ICC adjusted KS has an adjusted sample size. The KS statistic was still obtained as the supremum of the absolute distance between two EDFs. The informative sample size is defined as:

$$n^* = ICC * n \tag{2.8}$$

With previous knowledge, we may assume a general form that the informative sample size $n'$with adjustment by the spatial autocorrelation coefficient of Moran's I be

$$n^{'} = n \times \frac{2}{1 + e^{g(I)}}$$

Where $g(I)$ is the function of I, $g(I) = \beta_1 I + \beta_2 I^2 + \cdots + \beta_i I^i$. For the sake of parsimony, I only consider $g(I) = \beta_1 I + \beta_2 I^2 + \beta_3 I^3$.

Therefore to simplify the model I considered

$$A = \frac{n'}{n} = \frac{2}{1 + e^{g(I)}}$$

The original one-sample and two-sample KS statistic has the supremum form as followed

$$\begin{aligned} K_n &= \sqrt{n} \sup_x |F_n(X) - G_n(X)| \\ K_{m,n} &= \sqrt{\frac{mn}{m+n}} \sup_{x,y} |F_n(X) - G_m(Y)| \end{aligned}$$

The KS statistic with adjustment for spatial autocorrelation is defined as followed

$$\begin{aligned} K^{'}_{n^*} &= \sqrt{n^*}\sup_x |F_n(X) - G_n(X)| \\ K^{'}_{m^*,n^*} &= \sqrt{\frac{m^*n^*}{m^*+n^*}}\sup_{x,y} |F_n(X) - G_m(Y)| \end{aligned}$$

A generalized linear model (GLM) may be considered to estimate the βs. Assuming a link function $l(A) = \log(\frac{1}{A} - 1)$, the adjustment ratio may be rewrite into the following general linear form

$$E(l(A)) = g(I)$$

Parameters were estimated with the maximum likelihood. In order to simplify our model with emphasizing on the most influential variables. I used the lasso to select for dimension reduction. A valid hypothesis test requires controlled type I error rate, which should be near the pre-claimed nominal level. After the type I error is controlled, a satisfied power to discriminate against differences between tested distributions is desired. Therefore, I used type I error under the most popular nominal level of 0.05 and power of my adjusted KS test as benchmarks to evaluate the KS test.

In order to provide a clear picture of how the spatially adjusted KS test performed compared to the other KS type tests. I have evaluated the traditional KS test without spatial autocorrelation adjusted sample size, KS test adjusted with ICC, KS test with Griffith's adjustment and lastly, my adjusted KS test. The designed nature of image scans limit the sample locations, in other word, the sample size is fixed at 1344. Therefore, the power of KS tests was analyzed for differences in parameters of distributions. I was able to test the distribution change in mean, μ, at the ratio of $0.05, 0.1, 0.2, 0.5, 1$. Same differences ratio was analyzed for the variance, σ as well as in both mean and variance.

Spatial Coordinates and Geometry Characteristics of Human Heart

The geometry of the heart plays a critical role in the mechanics of cardiology. Back in 1892, Wood has used a spherical coordinate system to mimic the heart shape. Since then the sphericity index system has been popularly used by several studies to reconstruct the shape of the heart. (Mitchell, Lamas, Vaughan, & Pfeffer, 1992a) Azhari 1998 used a special normalized helical shape descriptor, denoted "geometrical cardiogram", to determine the shape of left ventricular. As the spherical shape has been proved to provide a simulation in shape that is close enough to the heart. (Azhari, Beyar, & Sideman, 1999)

In this study, I focused on the reconstruction of cardiac geometry locations with PET-CT image data. For each PET scan, electric signal values for CFR were recorded in a matrix form with 21 rows and 64 radials. In order to reconstruct the cardiac locations from PET image, I simulated a gridded map with a shape of a truncated ellipsoid, similar to a half football.

Gridded Map

Once the simulation shape of heart is decided, I simulated fixed locations D along the fields to represent the electronic recording points in the image location. The nature of gridded spatial data in $\boldsymbol{R^3}$ can be viewed as a two-way table. (N. Cressie, 1992) Locations $s_i \in D$, D is the subset of $\boldsymbol{R^3}$ and the realization in such location is $Z(s_i)$.

Given the spherical coordinates system

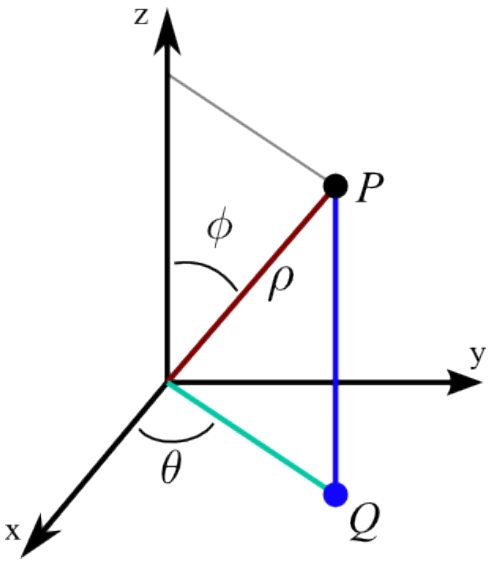


Figure 2.5: Spherical Coordinates

The procedure to generate the 3-D gridded map is as followed

1. Define the radius of the half football we want as

$$\rho = 1.$$

2. Then the define θ on the circle as 64 equal cuts of 2π

$$\Theta = (\theta_1, \theta_2, \ldots, \theta_6 4) = (\frac{1}{32}\pi, \frac{2}{32}\pi, \ldots, 2\pi).$$

3. Similarly define ϕ as 21 equal cuts of $(\pi/2, \pi)$

$$\Phi = (\phi_1, \phi_2, \ldots, \phi_2 1) = (\frac{21}{42}\pi, \frac{22}{42}\pi, \ldots, \frac{41}{42}\pi).$$

4. Transfer spherical coordinates into catesian coordinates

$$\begin{aligned} x &= \rho \sin\phi \cos\theta \\ y &= \rho \sin\phi \sin\theta \\ z &= \rho \cos\phi \end{aligned}$$

The generate 3-D space is realized as followed.

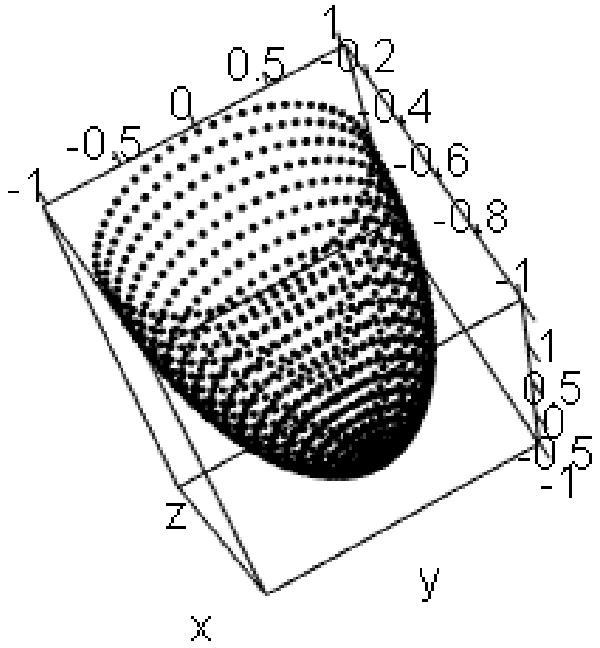


Figure 2.6: Generated Coordinates for Reconstructing PET into Heart shape

After the 3-D space is simulated, the distance between each unique pair of locations may be calculated. I defined the arc length between two locations as the interested distance. The distance between two location $s_i = (x_i, y_i, z_i) = (\rho \sin\phi_i \cos\theta_i, \rho \sin\phi_i \sin\theta_i, \rho \cos\phi_i)$ and $s_j = (x_j, y_j, z_j) = (\rho \sin\phi_j \cos\theta_j, \rho \sin\phi_j \sin\theta_j, \rho \cos\phi_j)$ is defined as

$$Acos = \arccos\left(\cos\phi_i \cos\phi_j + \sin\phi_i \sin\phi_j \cos\left(\theta_i - \theta_j\right)\right) \tag{2.9}$$

$$dist(s_i, s_j) = \begin{cases} \rho \times \arccos\left(1\right), & \text{Acos} \geq 1 \\ \rho \times \arccos\left(-1\right), & \text{Acos} \leq 1 \\ \rho \times Acos, & \text{otherwise} \end{cases} \tag{2.10}$$

The weight function w_{ij} is defined as the squared inverse distance

$$w_{ij} = \frac{1}{(dist(s_i, s_j))^2}$$

The weight matrix **W** is therefore defined as

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1n} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & w_{n3} & \dots & w_{nn} \end{bmatrix} \tag{2.11}$$

After the setup of the spatial environment, the procedure for simulating spatially autocorrelated samples in grid map figure 3.21 is followed as

1. Simulate N samples from $i.i.d$normal distribution$N(\mu, \sigma^2)$. In this study, $N = 1344,\ \mu = (0.5, 1)$ and $\sigma = (0.5, 1, 2)$.

2. Refer to the Cholesky decompsition method in section , calculate L from given covariogram structure Σ.

3. Refer to transformation equation 3.16, transfer N $i.i.d$ samples into the grid with respect the weight matrix **W**.

Study Design

In order to study the efficiency of dipyridamole, adenosine, and regadenoson and provide arguments for which one outperforms others. The Weatherhead PET Center for Preventing and Reversing Atherosclerosis of the University of Texas Medical School at Houston and Hermann Hospital conducted an investigator-initiated, single-centered, diagnostic accuracy trial between December 2012 to June, 2014.(Johnson & Gould, 2015) Subjects were recruited with following but not limited to entry criteria

1. Subjects were 40 years or older
2. Subjects with written informed consent

Subjects met any of the following but not limited to exclusion criteria will not be included in the trial

1. Any absolute contraindication to dipyridamole or regadenoson
2. Pregnancy or active breastfeeding
3. Current participation in another clinical research study
4. inability to undergo 2 PET scans within 2 months, but at least 1 day apart

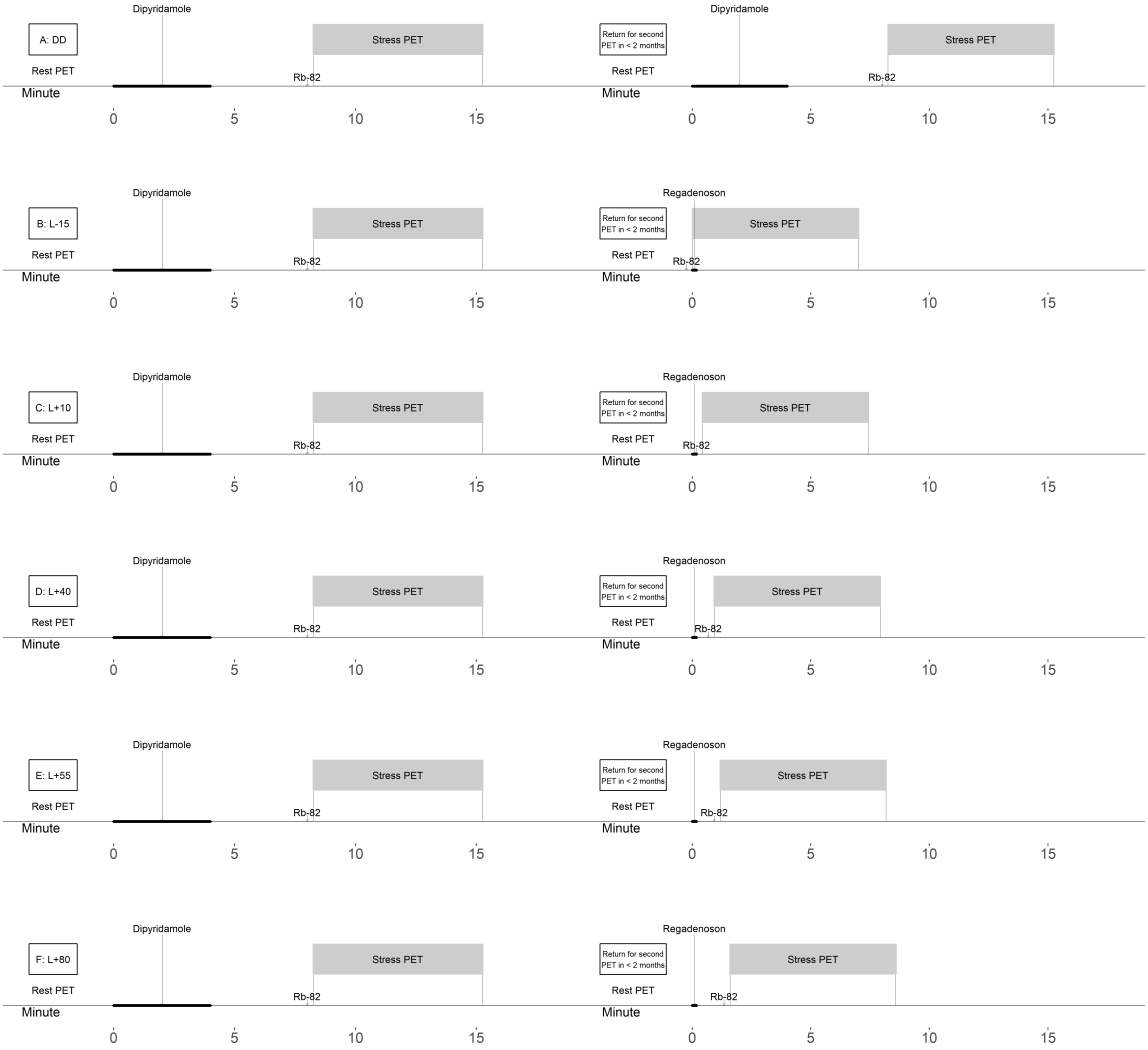
Protocol

Recruited subjects were split into 6 groups, each group went through a two-stage PET imaging procedure. The first group of subjects was administered with dipyridamole in both the first stage and second stage of PET scans. The second group of subjects was administered with the procedure of Rb-82 activated 15s before injection of regadenoson in one stage and with dipyridamole in the other stage. Similarly, the third, fourth, fifth and sixth group of subjects

were administered regadenoson with a certain time of activation of Rb-82 in one stage and administered with dipyridamole in the other stage.

Different dipyridamole protocol timing has been studied. Researchers applied the current optimal protocol of 4 mins dipyridamole protocol in the trail.(Harel, Finnerty, Authier, & Pelletier-Galarneau, 2018) According to the dipyridamole protocol guideline, dipyridamole ($142ug/kg/min$) was infused for 4 min. After dipyridamole is infused, Rb-82 generator was activated. PET stress scan starts 15s after Rb-82 generator activation.

Regadenoson protocol indicates that a single-use, pre-filled, 5-ml syringe of regadenoson was administered for 10s via a peripheral vein. Time of Rb-82 generator activation varies by protocols. Similarly, 10s after Rb-82 generator activation, PET scan was performed.



Notes: The bold black line in the timeline denotes the duration of medication, either dipyridamole or regadenoson, infusion. Protocols in the left is the baseline with dipyridamole, protocols in the right are study group with dipyridamole, Rb-82 activated 15s before regadenoson administration and Rb-82 activated 10s/40s/55s/80s after regadenoson administration.

Figure 2.7: Description of Protocols

Protocol	Description
DD	Repeated dipyridamole
L - 15	Regadenoson group with Rb-82 activated 15 seconds prior to injection of regadenoson
L + 10	Regadenoson group with Rb-82 activated 10 seconds after injection of regadenoson
L + 40	Regadenoson group with Rb-82 activated 40 seconds after injection of regadenoson
L + 55	Regadenoson group with Rb-82 activated 55 seconds after injection of regadenoson
L + 80	Regadenoson group with Rb-82 activated 80 seconds after injection of regadenoson

Table 2.3: Protocols

In this single-subject design, subjects using dipyridamole was used as the baseline and compared with themselves using either dipyridamole repeatedly in DD protocol or using regadenoson in L-15, L+10, L+40, L+55, L+80.

Journal Articles

A Simulation Study of A Class of Nonparametric Test Statistics: A Close Look of Continuous, Discrete and Correlated Variables

Journal of Statistical Computation and Simulation

Abstract

Kolmogorov-Smirnov test is a non-parametric hypothesis test that measures the probability of deviations, that the interested univariate random variable is drawn from a pre-specified distribution (one-sample KS) or has the same distribution as a second random variable (two-sample KS). The test is based on the measure of the supremum (greatest) distance between an empirical distribution function (EDF) and a pre-specified cumulative distribution function (CDF) or the largest distance between two EDFs. KS test, as well as other EDF based tests such as Anderson-Darling test and Cramer-von Mises test, have been widely adopted in statistical analysis due to its virtue of more general assumptions compared to parametric test like t-test. However, it is unclear under which condition will different EDF based test works best. Therefore to address such issues, I have conducted a systematic review of the performance of the original KS test, CvM test, AD test, and Chi-squared test. The assessment will be both on one sample and

two sample tests. We concluded that if we do not have prior information about the distributions going to be tested, EDF-based tests are better. However, so long as we have prior information about tested distribution and the distribution is bell-shaped and we are expecting differences in variance/sparseness, then the Chi-squared test may be more preferable. When correlation exists between tested samples, adjustment on the informative sample size is important and required.

Introduction

Together with other goodness of fit tests, Chi-squared, Shapiro-Wilk tests, and other popular ones, researchers are given a considerable library of tests to pick from. Though it is a good thing to be provided with varieties of methods to apply for different problems, one may find himself/herself hard to decide which methods to apply. In order to address such issues, we conducted a systematic review of the performance of the original KS test, CvM test, AD test, and Wilcoxon rank-sum test. The assessment will be both on one sample and two sample tests.

In the year of 1933, Kolmogorov published a short but landmark paper, in which he formally defined empirical distribution function (EDF), in the *Italian Giornale dell'Istituto Italiano degli Attuari* (Kolmogorov, 1933).

To define the empirical distribution function, let set $x_1, x_2, \ldots, xi-1, x_i$ be the realizations of random variables X having the $F(x) = pr(X < x)$. Put

$$\epsilon(x) = I(x_i \leq x)$$

Then the EDF is defined as:

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} \epsilon(x_i)$$

It could be easily seen that the EDF $F_n(x)$ is the portion of $x_1, x_2, \ldots, xi-1, x_i$ of X below x. It comes naturally to ask how close EDF is to its corresponding CDF. To answer this question, Kolmogorov studied and gave the asymptotic distribution of EDF. This led to the definition of Kolmogorov statistic (or Kolmogorov-Smirnov statistic) D and the distribution of D given finite sample size n was derived.

$$D = \sup_x |F_n(x) - F(x)|$$

The two sample version of the KS statistic is defined as

$$D_{n,m} = \sup_x |F_n(x) - G_m(x)|$$

Later, Smirnov proposed the Cramer-von Mises statistic (CvM statistic) ω^2, which can be viewed as an extension of KS statistic, based on Cramer's work in 1928 and von Mises's work in 1931. (von Mises, 1931; N. V. Smirnov, 1937; Mises, 1928) In which, Smirnov also found the asymptotic distribution of ω^2, in the form of a sum of weighted chi-squared variables.

$$\omega^2 = \int_{-\infty}^{\infty} [F_n(x) - F(x)]^2 f(x) dx$$

Choulakian extended the Cramer-von Mises statistic into the scope for discrete distributions or continuous distributions being grouped. (Choulakian et al., 1994) Consider $x_1^*, \ldots, x_L^*$ as the ordered L-distinct sample of X.

$$W_2^2 = \frac{1}{n} \sum_{j=1}^{L} (S_j - T_j)^2 p_j$$

Where o_j is the number of observations coinciding with x_j^*, then

$$S_j = \sum_{i=1}^{j} o_i$$

$$T_j = \sum_{i=1}^{j} Np_i$$

Researchers extended the discrete CVM into the scope of k-sample CVM for discrete distribution or continuous distribution being grouped. Consider ordered observations $Z_1^*, \ldots Z_L^*$ as the L distinct pooled sample of X and Y. (Brown, 1982, 1994; Lockhart et al., 2007)

Let

$$k_1 = n$$

$$k_2 = m$$

The two-sample CVM for discrete distribution is defined as followed

$$W_d^2 = \sum_{i=1}^{2} k_i \sum_{j=1}^{L} (S_{ij} - T_{ij})^2 p_j$$

Where S_{1j} is the number of observations in X not greater than Z_j^* , S_{2j} is the number of observations in Y not greater than Z_j^*,

$$T_i j = k_i \sum_{i=1}^{j} p_l$$

and $(n+m)p_j$ is the number of observations of a pooled sample of X and Y coinciding with z_j^*. The asymptotic distribution has been worked out by Sun. If $W_d^2 > \omega_{(d,\alpha)}^2$, then we reject H_0.

By modifying the weight factor of CvM statistic, T. W. Anderson and D. A. Darling (1952) proposed the Anderson Darling statistic (AD statistic) A.

$$A^2 = n \int_{-\infty}^{\infty} \frac{[F_n(x) - F(x)]^2}{F(x)[1 - F(x)]} f(x)dx$$

AD statistic under discrete setting is defined as follows.

$$A^2_{n,m} = \sum_{i=1}^{2} \frac{1}{k_i} \sum_{j}^{L-1} \frac{l_j}{N} \frac{(NM_{ij} - B_j k_i)^2}{B_j(N - B_j)}$$

Where f_{1j} be the number of observations in X coinciding with Z^*_j, f_{2j} be the number of observations in Y coinciding with Z^*_j and let

$$l_j = f_{1j} + f_{2j}$$

$$M_{ij} = f_{i1} + \cdots + f_{ij}$$

$$B_j = l_1 + \cdots + l_j$$

Pettitt worked out an approximation formula to calculate the variance of $A^2_{n,m}$. (Pettitt & Stephens, 1977)

$$var(A^2_{n,m}) = \frac{2(\pi^2 - 9)}{3} \times (1 - \frac{3.1}{N})$$

Methods

Tests mentioned above are fall in the category of "distribution-free method" which means they are robust under different distributions. However, the virtue of "distribution-free" sometimes may cause problems. When the parameter or even the distribution of our interested random variables unknown, it is hard to estimate the sample size required for certain power of the test. Therefore, I set up an environment with manually controlled various sample sizes. To evaluate

the performance of the tests, we used certain characteristics of the power of hypothesis testings mentioned above under different sample size and at significance levels of 0.05. In order to study the robustness of the above tests in the presence of dependence pattern, we generated subjects that are linearly correlated and autocorrelated.

Simulation

Simulated samples were drawn from the Weibull distribution $W(\gamma, \lambda)$ with two parameters, as it is commonly being applied in survival analysis, engineering and geology, normal distribution $N(\mu, \sigma^2)$ and multinomial distribution $Mult(n, p)$. Meanwhile, Weibull distribution of shape parameter γ and scale parameter λ makes us able to control the skewness of the testing distributions.

$$f(x) = \frac{\gamma}{\lambda}(\frac{x}{\lambda})^{\gamma-1}e^{(\frac{x}{\lambda})^{\gamma}}$$

$$F(x) = 1 - e^{(-\frac{x}{\lambda})^{\gamma}}$$

It is possible for me to control the actual magnitude of the difference between the two distributions by using theoretical distributions with known parameters. Thereafter I will compare the power of above tests under certain circumstances stated as followed.

Monte Carlo simulations will be used to evaluate the statistical power of KS, CvM, AD and Chi-squared statistics. Consider random variable $X : x_1, x_2, \ldots, x_n$ from

$$W(\gamma, \lambda), where \gamma = 0.5, 1, 2, 3, 5; \lambda = 1, 2, 3$$

$$N(\mu, \sigma^2), where \mu = 0, 1, 3, 5; \sigma = 0.1, 0.5, 2$$

$$Mult(n, P)$$

where

$$P = \begin{cases} C_1 = (p_1, p_2) = (0.5, 0.5), & \text{Symmetric} \\ C_2 = (p_1, p_2) = (0.1, 0.9), & \text{Heavily Skewed} \\ C_3 = (p_1, p_2) = (0.3, 0.7), & \text{Skewed} \\ C_4 = (p_1, p_2, p_3, p_4, p_5) = (0.1, 0.2, 0.4, 0.2, 0.1), & \text{Symmetric} \\ C_5 = (p_1, p_2, p_3, p_4, p_5) = (0.7, 0.2, 0.05, 0.03, 0.02), & \text{Skewed} \\ C_6 = (p_1, p_2, p_3, p_4, p_5) = (0.3, 0.15, 0.1, 0.15, 0.3), & \text{Symmetric with Heavy Tails} \end{cases}$$

The null and alternative hypothesis to be tested is as followed,

$$H_0 : F(x) = G(x) \tag{3.12}$$

$$H_1 : F(x) \neq G(x) \tag{3.13}$$

G(x) is the pre-specified distribution function of $W(\gamma + \Delta, \lambda + \Delta), N(\mu + \Delta, (\sigma + \Delta)^2)$ and $Mult(n, p + \Delta)$, where the difference ratio Δ is

$$\Delta = 0.05, 0.1, 0.2, 0.5, 1$$

Meanwhile, σ controls the shape and density of the probability curve in normally distributed data. The mean parameter μ from normal distribution shifts the entire curve while not changing shape and density distribution. Therefore, the change in σ and μ provide us an opportunity to test the performance under shape differences and location differences, or both differences.

Lastly, in the multinomial distributed data group, we will have a chance to evaluate the performance of KS, CvM and AD tests when data is indeed discrete. When, unfortunately, certain parameters of the distribution were not available and we are left with no option on the table but to estimate these parameters from the sample, then results from Kolmogorov-Smirnov test will be conservative. (Simpson, 1951; Crutcher, 1975; Lilliefors, 1967) Methods were

proposed to extend EDF tests on discrete data. Therefore, we simulated data from multinomial distribution under different conditions.

Correlated Realizations

In order to simulate correlated samples, we applied the copula method (Joe, 1997). For the sake of easy computation and estimation, we choose a Gaussian copula method for its relatively high accuracy. The procedure of copula methods to simulate bivariate correlated Weibull distribution is as followed.

1. First, we choose a covariance matrix Σ that reflects the correlations relationship in our targeted samples. Based on the covariance structure we would like to achieve, we draw correlated samples $X_1 = (x_{1,1}, x_{1,2}, x_{1,3}, \ldots, x_{1,n})$ and $X_2 = (x_{2,1}, x_{2,2}, x_{2,3}, \ldots, x_{2,m})$ from standard bivariate Gaussian distribution. Therefore we may have

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim MVN \left(\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 1 & r^2 \\ r^2 & 1 \end{pmatrix} \right)$$

2. Find the CDF of X_1 and X_2 as $\phi(X_1), \phi(X_2)$.

3. In order to simulate correlated samples $Z_1 = (z_{1,1}, z_{1,2}, z_{1,3}, \ldots, z_{1,n})$ and $Z_2 = (z_{2,1}, z_{2,2}, z_{2,3}, \ldots, z_{2,m})$ from the targeted distribution, we find the targeted inver-CDF function as $F^{-1}(Z_1)$ and $F^{-1}(Z_2)$

4. Compute the following function and our interested correlated samples may be obtained

$$\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \begin{bmatrix} F^{-1}(\phi(X_1)) \\ F^{-1}(\phi(X_2)) \end{bmatrix}$$

There are several choices for the correlation matrix to simulate the bivariate Gaussian distribution. Rank correlation coefficients, such as Kendall's τ and Spearman's ρ, are usually preferred as they are invariant to strictly increasing transformations (Ding & Li, 2013). The linear correlation coefficient, on the other hand, may not be invariant to non-linear transformations but have the virtue of able to be applied directly to simulate normal distribution in the first step. In addition, the trend of the correlation relationship between samples is invariant. Dithinde used a translation-based lognormal model with Pearson's r to capture the correlation structure between two hyperbolic curve-fitting parameters and have relatively well results. (Dithinde et al., 2011) Genest report the simulation with Pearson's r measuring the correlation structure to be performing reasonably well when simulated sample size n is 50 or larger. We used Pearson's r to simulate the bivariate normal distribution. (Genest & Rivest, 1993)

The performance of EDF based tests and the Chi-squared test will be evaluated by their simulation results of type I error and power. To evaluate the effects of sample size on type I error and power, we simulated samples of size $n = (10, 20, 30, 100, 500)$. Type I error and power will be analyzed from realization results of 10,000 repeated iterations.

Results

Analysis of Type I error

Comparison of one-sample tests

From the simulation results of the continuous distribution, such as normal distribution and Weibull distribution in our case, the EDF type tests achieved the type I error that is reasonably close to nominal level even when the sample size is relatively small (n = 10). When sample size $n \geq 30$, all tests achieve a type I error around the nominal level of 0.05.

Table 3.5: Type I Error for One-Sample Tests of Multinomial Distributions

Sample Size	Test	Test Sets					
		C_1	C_2	C_3	C_4	C_5	C_6
10	KS	0.022	0.013	0.011	0.022	0.013	0.028
	CvM	0.106	0.071	0.011	0.039	0.079	0.049
	AD	0.022	0.071	0.075	0.043	0.075	0.046
	Chi-squared	0.022	0.071	0.075	0.047	0.078	0.050
20	KS	0.041	0.043	0.024	0.014	0.028	0.026
	CvM	0.116	0.043	0.024	0.048	0.053	0.045
	AD	0.116	0.043	0.081	0.052	0.041	0.047
	Chi-squared	0.041	0.043	0.024	0.053	0.064	0.046
30	KS	0.046	0.028	0.028	0.015	0.029	0.054
	CvM	0.098	0.028	0.028	0.044	0.054	0.048
	AD	0.098	0.123	0.070	0.046	0.046	0.047
	Chi-squared	0.046	0.028	0.070	0.047	0.067	0.050
100	KS	0.007	0.000	0.003	0.006	0.003	0.016
	CvM	0.057	0.031	0.059	0.049	0.046	0.052
	AD	0.057	0.068	0.059	0.049	0.051	0.052
	Chi-squared	0.057	0.068	0.059	0.049	0.053	0.049
500	KS	0.006	0.000	0.004	0.006	0.003	0.012
	CvM	0.066	0.027	0.046	0.051	0.049	0.050
	AD	0.078	0.085	0.057	0.052	0.049	0.052
	Chi-squared	0.053	0.041	0.046	0.049	0.045	0.048

From table 3.5 we may see that when the data is multinomial distributed, the KS test, as Conover mentioned in his paper, is more accurate when the sample size is less than 30. (Conover, 1972a) On the other hand, when the sample size $n > 30$, the modified KS test produced a conservative type I error. In addition, we found that Conover's KS test performs better when the discrete distribution is symmetric and have heavy tails. It is more conservative when the data is skewed. Moreover, EDF based tests are heavily influenced by the number of groups. They seem to perform better in multinomial distribution with 5 groups than that of 2 groups. As Chi-squared tests are for discrete samples, it performs the most stable among the 4 tests, it tends to be more accurate when the sample is symmetric and with more number of groups. In addition, the influence in symmetricity and number of groups were canceled out when the sample size is large than 100.

Comparison of two-sample tests

From table 3.6, we may see that when data is normally distributed, the KS and the Chi-square produced conservative statistics if the sample size is small, say n $<$ 100. When n = 100, the Chi-squared test has a controlled type I error while KS test does not. When sample size is large, n = 500, KS, AD, and chi-squared tests all have controlled type I error. However, CvM tests seem to be a little conservative.

Table 3.6: Type I Error for Two sample tests

		Sample Size				
Distribution	Test	10	20	30	100	500
Normal	KS	0.01	0.03	0.04	0.04	0.05
	CvM	0.05	0.04	0.04	0.04	0.04
	AD	0.05	0.05	0.05	0.05	0.05
	Chi-squared	0.01	0.03	0.03564	0.04	0.05

Weibull	KS	0.04	0.03	0.03	0.04	0.05
	CvM	0.05	0.04	0.04	0.04	0.04
	AD	0.05	0.05	0.05	0.05	0.05
	Chi-squared	0.01	0.01	0.01	0.02	0.03
Multinomial	KS	0	0.01	0.01	0.01	0.01
	CvM	0	0	0	0	0
	AD	0.06	0.05	0.05	0.05	0.05
	Chi-squared	0.03	0.04	0.04	0.05	0.05

Normal distribution is from $N(0, 4)$.

Weibull distribution is from $W(1, 2)$.

Multinomial distribution from $C_4 = (0.1, 0.2, 0.4, 0.2, 0.1)$.

When simulated data is from Weibull distribution, results from table 3.6 are similar to that of normal distributions. However, it is noticeable that Chi-squared test was conservative when the shape parameter of Weibull is 0.5 and 1(heavily skewed), even though test slowly be more accurate when sample size increased, it still is very conservative when sample size reached 500. Meanwhile, the chi-squared test is more accurate when the shape parameter is large than 1. Therefore, from the simulated results we can confirm that the chi-squared test is not as stable in skewed distributed distributions as in symmetric cases.

In the multinomial tested results, the modified AD test seems to be the most stable one. Chi-squared is not accurate when the number of groups is 2 or the sample size is small. When the number of groups is 2, sample size $n = 500$ reaches satisfied accuracy. Meanwhile, it performs relatively well when the number of groups is 5 and symmetric. CvM is always not as accurate but not in group 6, which has symmetric and heavy-tailed distributed samples.

Correlated Samples

From the results from table 3.7, we may see that for normal distribution and Weibull distribution, when X and Y were sampled from correlated distributions and we did not address for such effects when applying the hypothesis testing, all the tests produced untrue type I errors. When the correlation between tested samples is positive then the type I error is overestimated. On the other hand, when correlation negative then we are more likely to have a liberal type I error. (Cribbie & Keselman, 2003) When the Pearson's $r \geq 0.5$, the EDF-based tests had a type I error of almost 0, however, Chi-squared test still had some rejection ability at the nominal level of 0.05. When the Pearson's $r = -0.8$, the type I error almost doubled.

Table 3.7: Type I Error for Correlated Samples

		Pearson's r			
Distribution	Test	0.5	0.8	-0.5	-0.8
Normal	KS	0.01	0	0.09	0.12
	CvM	0.01	0	0.09	0.12
	AD	0.01	0	0.10	0.14
	Chi-squared	0.02	0.01	0.06	0.08
Weibull	KS	0	0	0.10	0.12
	CvM	0	0	0.09	0.12
	AD	0	0	0.10	0.14
	Chi-squared	0.02	0.01	0.06	0.08

Sample size $N = 500$

Normal distribution is from $N(0, 4)$.

Weibull distribution is from $W(1, 2)$.

Analysis of Power

Comparison of one-sample tests

Results for normal distributions is listed in table 3.8, when under the alternative with same mean and different variance, when the sample size is relatively small, $n = 10$, the chi-squared test is the most powerful one while significantly higher than the EDF ones. Under relatively large sample size, $100 > n > 20$, the Chi-squared test is still the most powerful when the change ratio in variance is below 50%, while when the change ratio in variance large than 100% then the AD test is more powerful.

Varaince				Mean					Mean			
Null	Alternative	Sample Size	Test	0	1	3	5	Sample Size	0	1	3	5
2.0	2.100	10	KS	0.05	0.04	0.04	0.05	100	0.05	0.05	0.04	0.05
			CvM	0.04	0.04	0.04	0.05		0.04	0.05	0.05	0.05
			AD	0.04	0.04	0.04	0.04		0.04	0.05	0.04	0.05
			Chi-Squared	0.04	0.04	0.04	0.04		0.06	0.05	0.06	0.06
	3.000		KS	0.04	0.04	0.04	0.04		0.59	0.58	0.59	0.60
			CvM	0.02	0.02	0.02	0.02		0.76	0.75	0.76	0.76
			AD	0.01	0.01	0.01	0.01		0.92	0.92	0.93	0.92
			Chi-Squared	0.11	0.11	0.11	0.10		0.92	0.91	0.91	0.91
	4.000		KS	0.05	0.05	0.05	0.05		1.0	1.00	1.00	1.00
			CvM	0.03	0.03	0.03	0.03		1.00	1.00	1.00	1.00
			AD	0.01	0.01	0.01	0.01		1.00	1.00	1.00	1.00
			Chi-Squared	0.27	0.27	0.27	0.27		1.00	1.00	1.00	1.00

Table 3.8: Power for One-sample Tests in Normal Distributed with Identical Mu

Power analsyis for Weibull distributions is listed in table 3.9, when the alternative is scale difference, even under small sample size, n = 10, the EDF based tests were more powerful than the chi-squared tests. Among the EDF tests, CvM and AD share almost identical power

under various alternatives. KS has a slightly low power but almost the same as the other two EDF ones. However, when the sample size is relatively large, the gap between AD, CvM and KS are greater, while the order is AD test $>$ CvM test $>$ KS test. When the alternative is the shape difference, similar to scale difference, the AD is the most powerful test in detecting the difference. However, we found that KS and CvM are not always better than the Chi-squared test.

Scale				Shape						Shape				
Null	Alternative	Sample Size	Test	0.5	1	2	3	5	Sample Size	0.5	1	2	3	5
1	1.05	10	KS	0.049	0.046	0.055	0.059	0.085	100	0.052	0.059	0.104	0.190	0.452
			CvM	0.045	0.049	0.058	0.062	0.089		0.054	0.069	0.125	0.231	0.542
			AD	0.047	0.045	0.054	0.060	0.084		0.051	0.068	0.127	0.236	0.554
			Chi-Squared	0.041	0.038	0.044	0.048	0.058		0.051	0.057	0.071	0.101	0.222
	2.00		KS	0.126	0.387	0.963	1.000	1.000		0.781	1.000	1.000	1.000	1.000
			CvM	0.138	0.441	0.984	1.000	1.000		0.856	1.000	1.000	1.000	1.000
			AD	0.127	0.412	0.980	1.000	1.000		0.869	1.000	1.000	1.000	1.000
			Chi-Squared	0.072	0.189	0.772	0.998	1.000		0.458	0.997	1.000	1.000	1.000

Table 3.9: Power for One-sample Tests in Weibull Distributed with Identical Shape

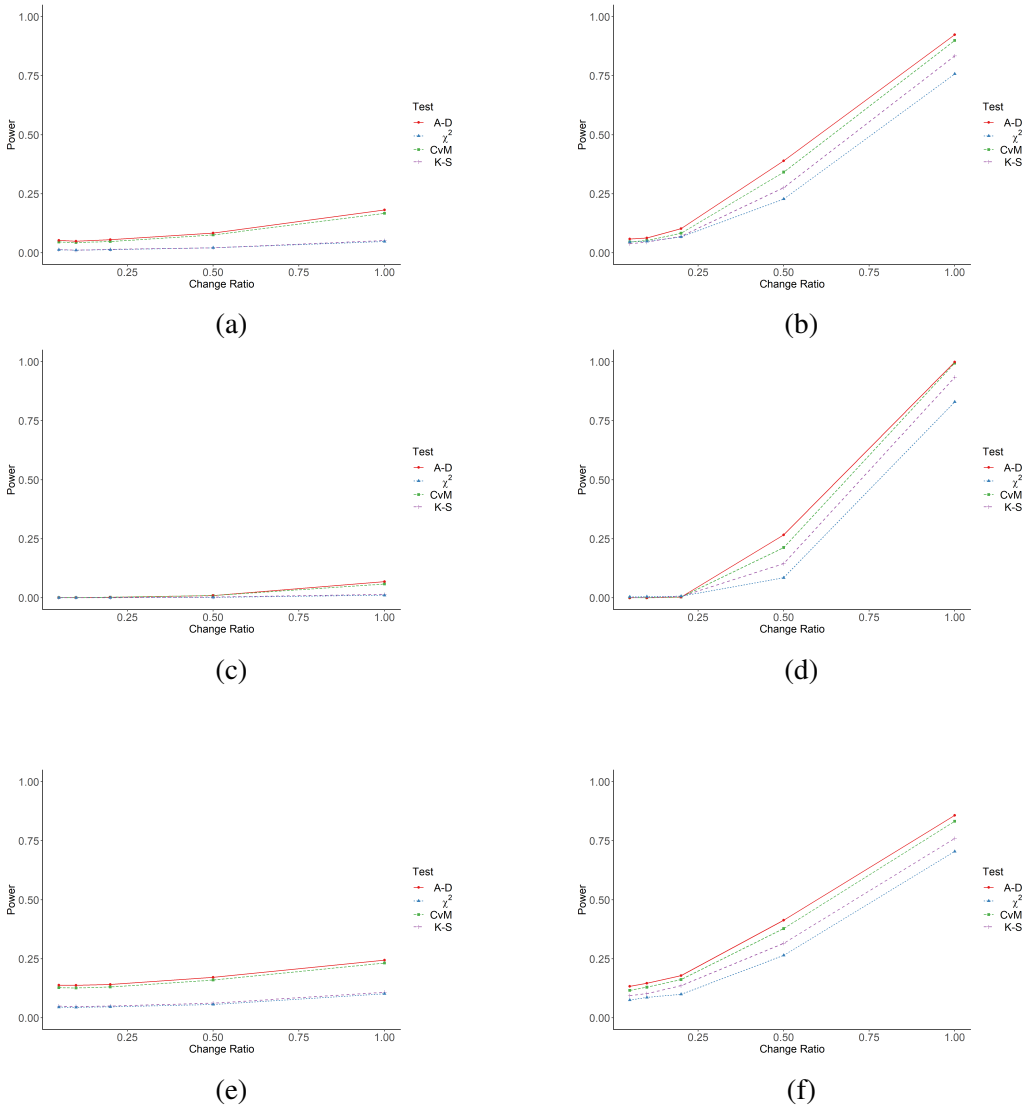
From the simulation results of multinomial cases in table 3.10, we may see that EDF-based tests have higher power when the sample distribution is not symmetric. When categories of multinomial distribution is more than 5, EDF based tests achieved comparable or higher power than the Chi-squared test. However, when the multinomial distribution is bell-shaped, then the Chi-squared test is the most powerful one.

Table 3.10: Type I Error for One-Sample Tests of Multinomial Distributions

		Test Sets					
Sample Size	Test	C_1	C_2	C_3	C_4	C_5	C_6
	KS	0.022	0.013	0.011	0.022	0.013	0.028

10	CvM	0.106	0.071	0.011	0.039	0.079	0.049
	AD	0.022	0.071	0.075	0.043	0.075	0.046
	Chi-squared	0.022	0.071	0.075	0.047	0.078	0.050
20	KS	0.041	0.043	0.024	0.014	0.028	0.026
	CvM	0.116	0.043	0.024	0.048	0.053	0.045
	AD	0.116	0.043	0.081	0.052	0.041	0.047
	Chi-squared	0.041	0.043	0.024	0.053	0.064	0.046
30	KS	0.046	0.028	0.028	0.015	0.029	0.054
	CvM	0.098	0.028	0.028	0.044	0.054	0.048
	AD	0.098	0.123	0.070	0.046	0.046	0.047
	Chi-squared	0.046	0.028	0.070	0.047	0.067	0.050
100	KS	0.007	0.000	0.003	0.006	0.003	0.016
	CvM	0.057	0.031	0.059	0.049	0.046	0.052
	AD	0.057	0.068	0.059	0.049	0.051	0.052
	Chi-squared	0.057	0.068	0.059	0.049	0.053	0.049
500	KS	0.006	0.000	0.004	0.006	0.003	0.012
	CvM	0.066	0.027	0.046	0.051	0.049	0.050
	AD	0.078	0.085	0.057	0.052	0.049	0.052
	Chi-squared	0.053	0.041	0.046	0.049	0.045	0.048

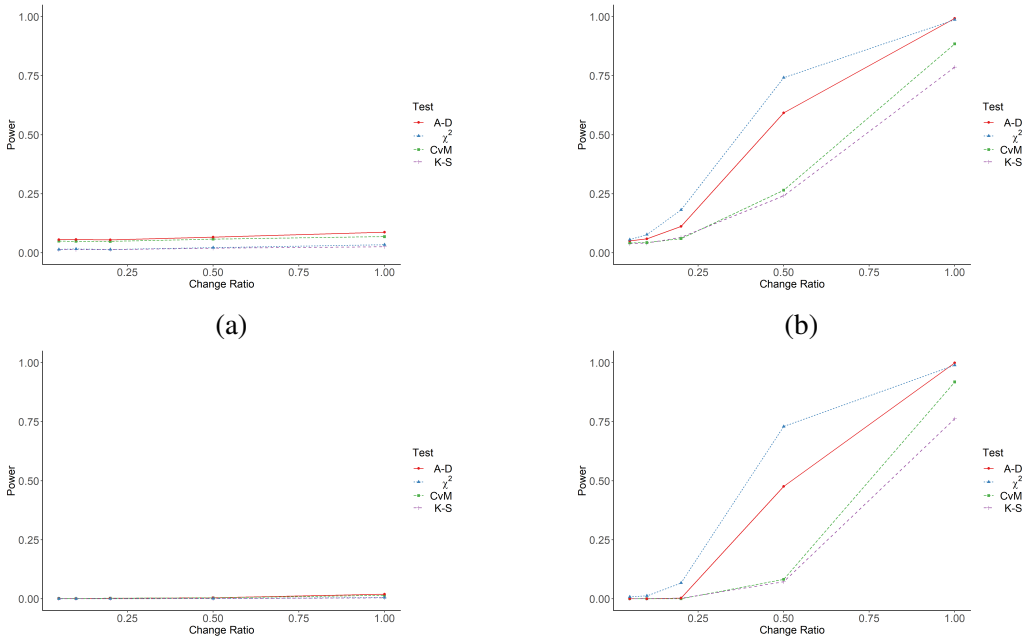
Two sample tests comparison



Top left, shows power analysis for $N(\mu_1 = 1, \sigma^2 = 4)$ and $N(\mu_2, \sigma^2 = 4)$, where $\mu_2 = \mu_1 * (1 + \Delta)$, sample size $N = 10$. Top right shows power analysis for $N(\mu_1, \sigma^2 = 4)$ and $N(\mu_2, \sigma^2 = 4)$, where $\mu_2 = \mu_1 * (1 + \Delta)$, sample size $N = 100$. Middle left was the power for the correlated case with $r = 0.8$, $N(\mu_1, \sigma^2 = 4)$ and $N(\mu_2, \sigma^2 = 4)$, where $\mu_2 = \mu_1 * (1 + \Delta)$, sample size $N = 10$. Middle right is the power for the correlated case with $r = 0.8$, $N(\mu_1, \sigma^2 = 4)$ and $N(\mu_2, \sigma^2 = 4)$, where $\mu_2 = \mu_1 * (1 + \Delta)$, sample size $N = 100$. Bottom left is the power for the correlated case with $r = -0.8$, $N(\mu_1, \sigma^2 = 4)$ and $N(\mu_2, \sigma^2 = 4)$, where $\mu_2 = \mu_1 * (1 + \Delta)$, sample size $N = 10$. Bottom right is the power for the correlated case with $r = -0.8$, $N(\mu_1, \sigma^2 = 4)$ and $N(\mu_2, \sigma^2 = 4)$, where $\mu_2 = \mu_1 * (1 + \Delta)$, sample size $N = 100$.

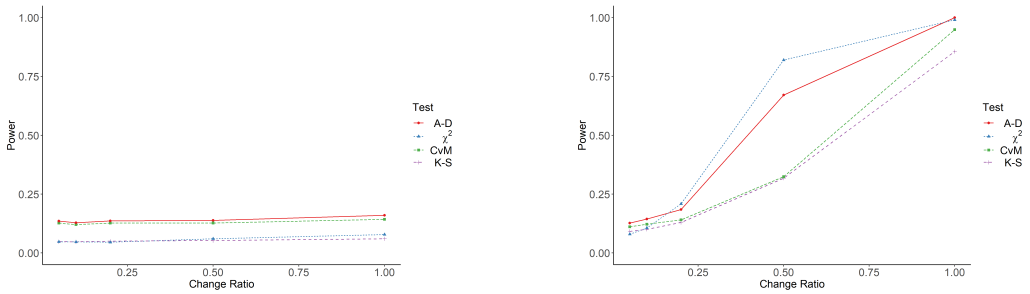
Figure 3.8: Power Analysis for Two-sample Tests on Normal distributions

From figure 3.8, we find that when the alternative was the difference in location (μ) shift, then the EDF based tests are more powerful than the Chi-squared test. Similarly to the previous power analysis on the variance difference, when the assumption of independence among samples are violated, the power of the four tests was relatively lower when there exist positive correlation and relatively higher power when samples were negatively correlated.



(a) (b)

(c) (d)

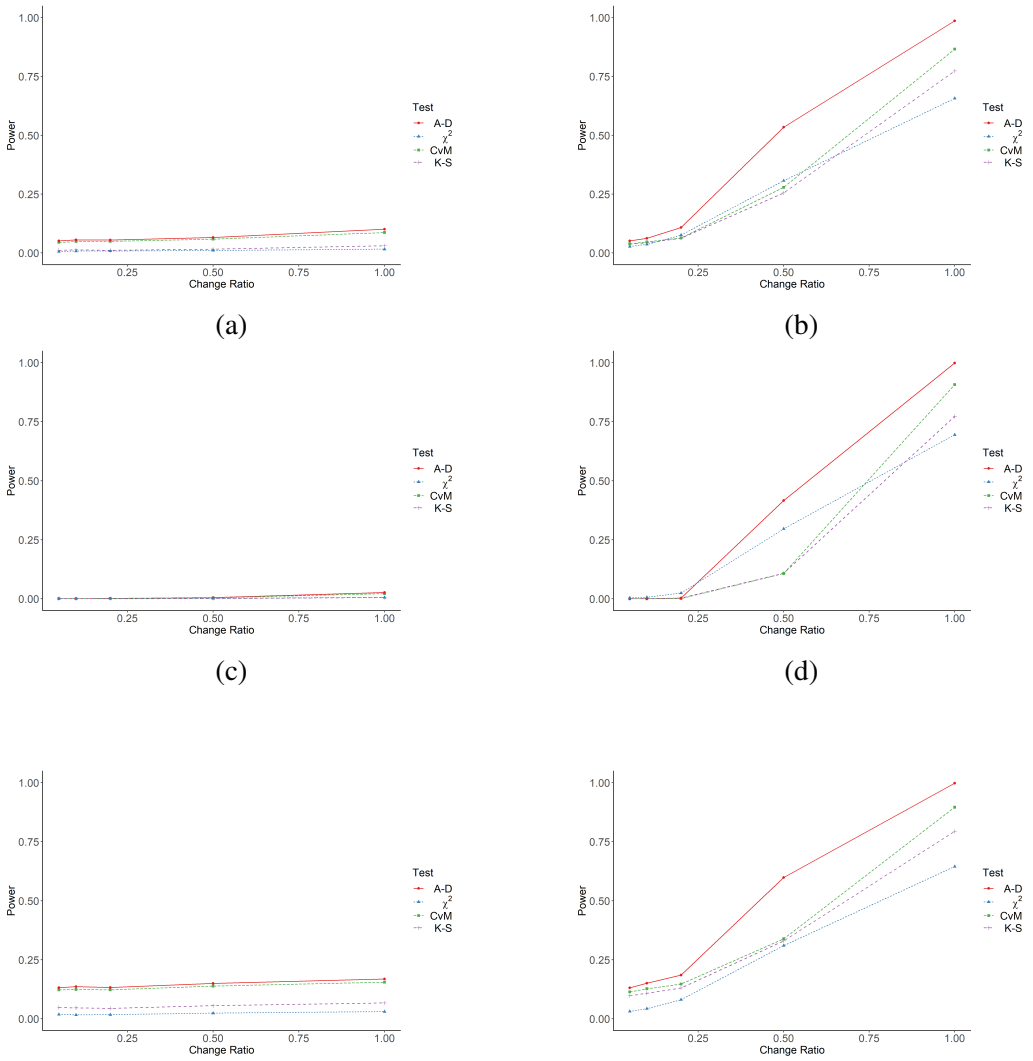


(e) (f)

Top left, shows power analysis for $N(0, \sigma_1^2 = 4)$ and $N(0, \sigma_2^2)$, where $\sigma_2 = \sigma_1 * (1+\Delta)$, sample size $N = 10$. Top right shows power analysis for $N(0, \sigma_1^2 = 4)$ and $N(0, \sigma_2^2)$, where $\sigma_2 = \sigma_1 * (1+\Delta)$, sample size $N = 100$. Middle left was the power for the correlated case with $r = 0.8$, $N(0, \sigma_1^2 = 4)$ and $N(0, \sigma_2^2)$, where $\sigma_2 = \sigma_1 * (1 + \Delta)$, sample size $N = 10$. Middle right is the power for the correlated case with $r = 0.8$, $N(0, \sigma_1^2 = 4)$ and $N(0, \sigma_2^2)$, where $\sigma_2 = \sigma_1 * (1 + \Delta)$, sample size $N = 100$. Bottom left is the power for the correlated case with $r = -0.8$, $N(0, \sigma_1^2 = 4)$ and $N(0, \sigma_2^2)$, where $\sigma_2 = \sigma_1 * (1 + \Delta)$, sample size $N = 10$. Bottom right is the power for the correlated case with $r = -0.8$, $N(0, \sigma_1^2 = 4)$ and $N(0, \sigma_2^2)$, where $\sigma_2 = \sigma_1 * (1 + \Delta)$, sample size $N = 100$.

Figure 3.9: Power Analysis for Two-sample Tests on Normal distributions

The results from figure 3.9 showed that under the distribution of normal, the two-sample tests have almost identical power to the one-sample conditions. When the alternative is the difference in dispersion rate (σ) then the Chi-squared test is the most powerful one. However, under the two-sample condition, the AD test has an acceptable rate to rightly discriminate among alternatives. When the underlying assumption of independence between samples is violated, $r = 0.8$, then the four tests achieved relatively lower powers than the independent cases. However, when $r = -0.8$ then the four tests were relatively more powerful to discriminate among alternative.

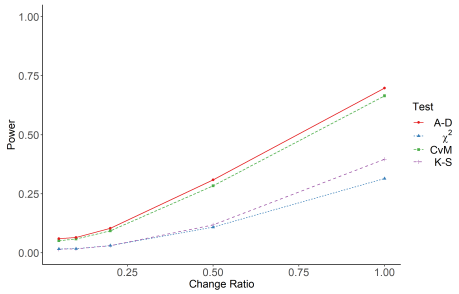


Top left, shows power analysis for $W(\gamma_1 = 1, \lambda = 2)$ and $W(\gamma_2, \lambda = 2)$, where $\gamma_2 = \gamma_1 * (1 + \Delta)$, sample size $N = 10$. Top right shows power analysis for $W(\gamma_1 = 1, \lambda = 2)$ and $W(\gamma_2, \lambda = 2)$, where $\gamma_2 = \gamma_1 * (1 + \Delta)$, sample size $N = 100$. Middle left was the power for the correlated case with $r = 0.8$, $W(\gamma_1 = 1, \lambda = 2)$ and $W(\gamma_2, \lambda = 2)$, where $\gamma_2 = \gamma_1 * (1 + \Delta)$, sample size $N = 10$. Middle right was the power for the correlated case with $r = 0.8$, $W(\gamma_1 = 1, \lambda = 2)$ and $W(\gamma_2, \lambda = 2)$, where $\gamma_2 = \gamma_1 * (1 + \Delta)$, sample size $N = 100$. Bottom left was the power for the correlated case with $r = -0.8$, $W(\gamma_1 = 1, \lambda = 2)$ and $W(\gamma_2, \lambda = 2)$, where $\gamma_2 = \gamma_1 * (1 + \Delta)$, sample size $N = 10$. Bottom right was the power for the correlated case with $r = -0.8$, $W(\gamma_1 = 1, \lambda = 2)$ and $W(\gamma_2, \lambda = 2)$, where $\gamma_2 = \gamma_1 * (1 + \Delta)$, sample size $N = 100$.

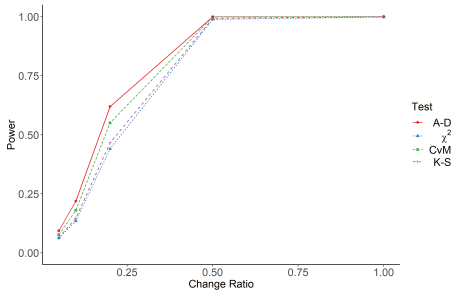
Figure 3.10: Power Analysis for Two-sample Tests on Weibull distributions

Figure 3.10 showed that when tested samples were from Weibull distribution, the simulation results showed that EDF tests were more powerful than the chi-squared tests when the tested

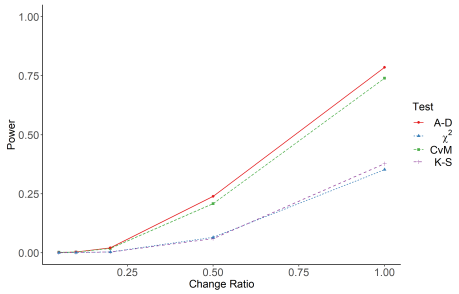
distributions were significantly different. Given the alternative that X and Y sampled from that of Weibull distribution with identical scale parameter, λ, but different shape parameter, γ_1 and γ_2, CvM, KS and Chi-squared tests were almost as powerful when the change ratio was less than 50%. However, when the change ratio in the shape parameter of tested Weibull populations was significant, more than 50%, then the EDF-based tests were much more powerful.



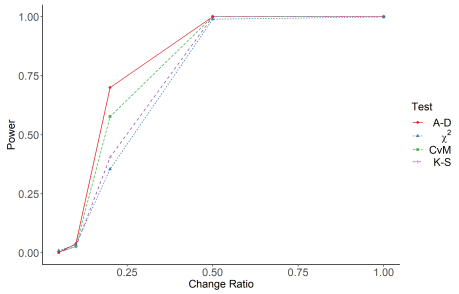
(a)



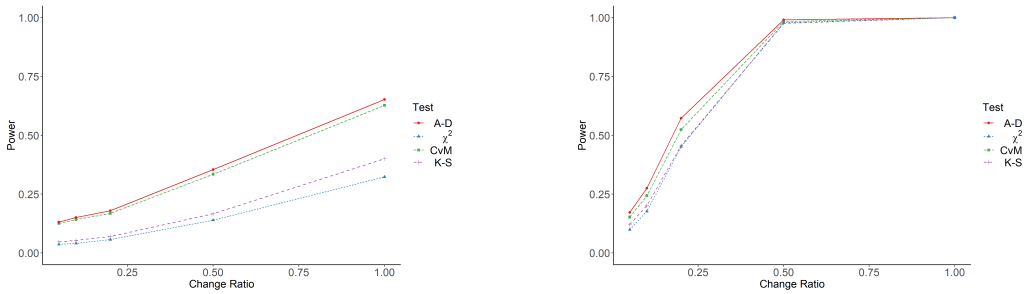
(b)



(c)



(d)

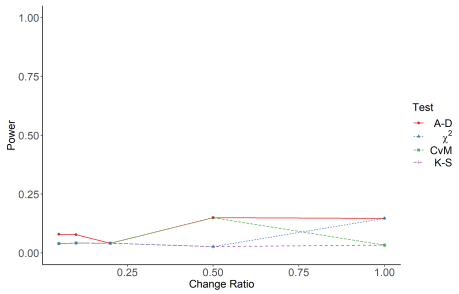


(e) (f)

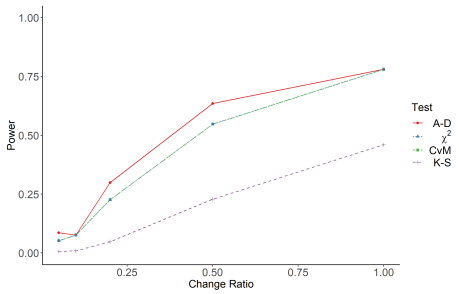
Top left, shows power analysis for $W(\gamma = 1, \lambda_1 = 2)$ and $W(\gamma, \lambda_2)$, where $\lambda_2 = \lambda_1 * (1 + \Delta)$, sample size $N = 10$. Top right shows power analysis for $W(\gamma = 1, \lambda_1 = 2)$ and $W(\gamma, \lambda_2)$, where $\lambda_2 = \lambda_1 * (1 + \Delta)$, sample size $N = 100$. Middle left was the power for the correlated case with $r = 0.8$, $W(\gamma = 1, \lambda_1 = 2)$ and $W(\gamma, \lambda_2)$, where $\lambda_2 = \lambda_1 * (1 + \Delta)$, sample size $N = 10$. Middle right was the power for the correlated case with $r = 0.8$, $W(\gamma = 1, \lambda_1 = 2)$ and $W(\gamma, \lambda_2)$, where $\lambda_2 = \lambda_1 * (1 + \Delta)$, sample size $N = 100$. Bottom left was the power for the correlated case with $r = -0.8$, $W(\gamma = 1, \lambda_1 = 2)$ and $W(\gamma, \lambda_2)$, where $\lambda_2 = \lambda_1 * (1 + \Delta)$, sample size $N = 10$. Bottom right was the power for the correlated case with $r = -0.8$, $W(\gamma = 1, \lambda_1 = 2)$ and $W(\gamma, \lambda_2)$, where $\lambda_2 = \lambda_1 * (1 + \Delta)$, sample size $N = 100$.

Figure 3.11: Power Analysis for Two-sample Tests on Weibull distributions

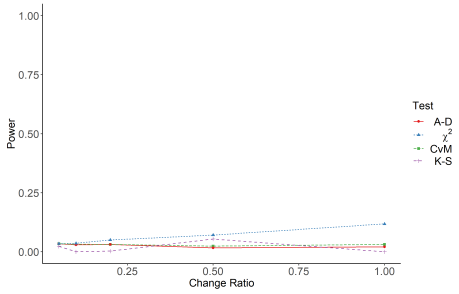
Figure 3.11 showed results from Weibull distribution with identical shape parameter, γ, while different scale parameter, λ, generally, the EDF based tests were more powerful than the Chi-squared test. It was worth noticing that when the independence assumption for the tested population was violated, the positive correlation leads to a conservative probability of rejecting of the null hypothesis when the difference between tested populations are not significant, while the rejecting probability increased drastically when the difference was more significant.


Power
1.00
0.75
0.50
0.25
0.00
0.25
0.50
0.75
1.00
Change Ratio
Test
A-D
χ^2
CvM
K-S

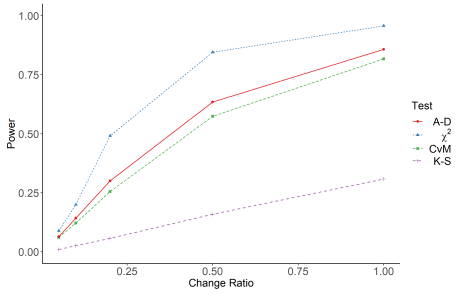
(a)


Power
1.00
0.75
0.50
0.25
0.00
0.25
0.50
0.75
1.00
Change Ratio
Test
A-D
χ^2
CvM
K-S

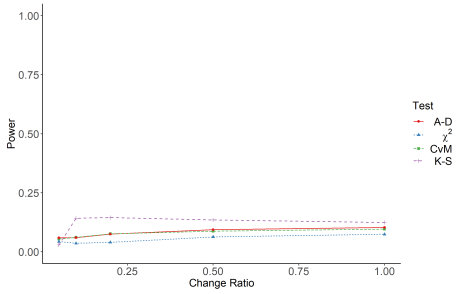
(b)


Power
1.00
0.75
0.50
0.25
0.00
0.25
0.50
0.75
1.00
Change Ratio
Test
A-D
χ^2
CvM
K-S

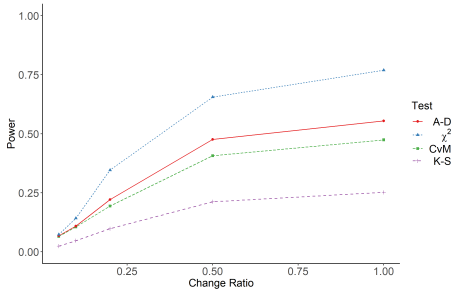
(c)


Power
1.00
0.75
0.50
0.25
0.00
0.25
0.50
0.75
1.00
Change Ratio
Test
A-D
χ^2
CvM
K-S

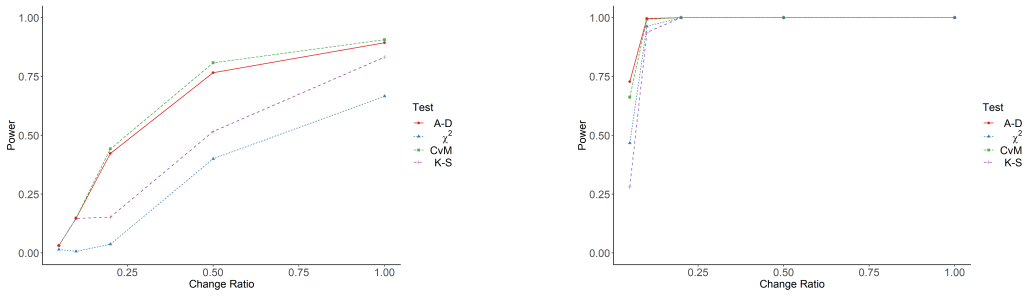
(d)


Power
1.00
0.75
0.50
0.25
0.00
0.25
0.50
0.75
1.00
Change Ratio
Test
A-D
χ^2
CvM
K-S

(e)


Power
1.00
0.75
0.50
0.25
0.00
0.25
0.50
0.75
1.00
Change Ratio
Test
A-D
χ^2
CvM
K-S

(f)



(g) (h)

Figure 3.12 (a), shows power analysis for skewed case with $P_2^1 = (p_1^1 = 0.3, p_2^1 = 0.7)$ and $P_2^2 = (p_1^2, p_2^2)$, where $p_n^2 = p_n^1 \times (1+\Delta)/\sum_{i=1}^n p_n^2$, sample size $N = 10$. Figure 3.12 (b) shows power analysis for $P_2^1 = (p_1^1 = 0.3, p_2^1 = 0.7)$ and $P_2^2 = (p_1^2, p_2^2)$, where $p_n^2 = p_n^1 \times (1+\Delta)/\sum_{i=1}^n p_n^2$, sample size $N = 100$. Figure 3.12 (c) was the power for symmetric case with $P_2^1 = (p_1^1 = 0.1, p_2^1 = 0.2, p_3^1 = 0.4, p_4^1 = 0.2, p_5^1 = 0.1)$ and $P_2^2 = (p_1^2, p_2^2, p_3^2, p_4^2, p_5^2)$, where $p_n^2 = p_n^1 \times (1+\Delta)/\sum_{i=1}^n p_n^2$, sample size $N = 10$. Figure 3.12 (d) was the power for $P_2^1 = (p_1^1 = 0.1, p_2^1 = 0.2, p_3^1 = 0.4, p_4^1 = 0.2, p_5^1 = 0.1)$ and $P_2^2 = (p_1^2, p_2^2, p_3^2, p_4^2, p_5^2)$, where $p_n^2 = p_n^1 \times (1+\Delta)/\sum_{i=1}^n p_n^2$, sample size $N = 100$. Figure 3.12 (e) was the power for symmetric multinomial distribution with heavy tails $P_2^1 = (p_1^1 = 0.3, p_2^1 = 0.15, p_3^1 = 0.1, p_4^1 = 0.15, p_5^1 = 0.3)$ and $P_2^2 = (p_1^2, p_2^2, p_3^2, p_4^2, p_5^2)$, where $p_n^2 = p_n^1 \times (1+\Delta)/\sum_{i=1}^n p_n^2$, sample size $N = 10$. Figure 3.12 (f) was the power for $P_2^1 = (p_1^1 = 0.3, p_2^1 = 0.15, p_3^1 = 0.1, p_4^1 = 0.15, p_5^1 = 0.3)$ and $P_2^2 = (p_1^2, p_2^2, p_3^2, p_4^2, p_5^2)$, where $p_n^2 = p_n^1 \times (1+\Delta)/\sum_{i=1}^n p_n^2$, sample size $N = 100$. Figure 3.12 (g) was the power for skewed multinomial distribution with heavy tails $P_2^1 = (p_1^1 = 0.7, p_2^1 = 0.2, p_3^1 = 0.05, p_4^1 = 0.03, p_5^1 = 0.02)$ and $P_2^2 = (p_1^2, p_2^2, p_3^2, p_4^2, p_5^2)$, where $p_n^2 = p_n^1 \times (1+\Delta)/\sum_{i=1}^n p_n^2$, sample size $N = 10$. Figure 3.12 (h) was the power for $P_2^1 = (p_1^1 = 0.7, p_2^1 = 0.2, p_3^1 = 0.05, p_4^1 = 0.03, p_5^1 = 0.02)$ and $P_2^2 = (p_1^2, p_2^2, p_3^2, p_4^2, p_5^2)$, where $p_n^2 = p_n^1 \times (1+\Delta)/\sum_{i=1}^n p_n^2$, sample size $N = 100$.

Figure 3.12: Power Analysis for Two-sample Tests on Multinomial distributions

Interesting results from figure 3.12 were found from the power plots for multinomial distributions. When group numbers in multinomial are small or when the distributions are skewed, EDF-based tests were more powerful than the Chi-Squared test. When the multinomial distributions are symmetric and sample size large than 30, Chi-squared test has the highest power. The number of groups increases in a multinomial distribution, the more powerful the KS, the CvM, the AD and the Chi-squared test will be. Interestingly, the more skewed the multinomial distributions are, the more powerful the KS, the CvM, the AD and the Chi-squared test will be.

Discussion and Concluding Remarks

As compared to the Chi-squared test, the EDF-based tests have a steeper discriminate curve, in another word, EDF- based test may not perform as powerful to minor differences between tested populations but very powerful towards more significant differences. In addition, from the simulation results, we have shown that the Anderson-Darling test has the most satisfactory controlled type I error and power under sample sizes ranged from small to large and across multiple distributions.

The bell-shape assumption of distribution is critical for the Chi-squared test. We have noticed a considerable decline of accuracy for Chi-squared test when the tested distributions were from an unsymmetrical distribution family. On the other hand, EDF-based tests were consistent across distributions.

When correlation exists between tested samples, none of the tests was a suitable choice. The KS test in its original form, the CvM test, the AD test and the Chi-squared test have conservative type I error when the correlation was positive and liberal type I error when the correlation was negative, the degree of conservative/liberal of the tests increases when the degree of correlation increases and vice versa. Noticeably, Chi-squared test was less vulnerable to the violation of the independence assumption of tested samples than EDF-based tests, in another word, the Chi-squared test has less performance reduced when correlation exists among tested samples.

We may conclude that if we do not have prior information about the distributions going to be tested, EDF-based tests are better. However, so long as we have prior information about tested distribution and the distribution is bell-shaped and we are expecting differences in variance/sparseness, then the Chi-squared test may be more preferable. When correlation exists between tested samples, adjustment on the informative sample size is important and required.

Our simulation results for the one-sample KS test in discrete distribution is from Conover's method. Conover has mentioned in his paper that his discrete KS test is inaccurate when the

sample size n is larger than 30. In the two sample KS test simulation, we applied the original KS test which is known to be conservative when the tested distribution is discontinuous. Further research on the two-sample KS test for discontinuous distributions is needed.

The Chi-squared test has a relatively better power for continuous distribution when applying an optimal grouping algorithm. However, our simulation results have shown that the EDF-based tests, such as KS, CvM and AD, were more powerful and robust than the Chi-squared test. Only under certain conditions like the difference only exists in variation and the distribution is bell-shaped, Chi-squared test to be preferred. Among the EDF-based tests, the CvM and AD outperformed the KS in most cases as they have cumulative the difference while KS used the supremum of the density difference as the testing statistic. When the data is discrete, we may still apply the EDF based tests due to their higher power. Under the condition that tested samples are correlated, the tests are inaccurate and adjustments account for such effect is necessary.

References

Kolmogorov, A. (1933). Sulla determinazione empirica di una lgge di distribuzione. *Inst.Ital.Attuari, Giorn. 4*, 83–91.

von Mises, R. (1931). *Vorlesungen aus dem gebiete der angewandten mathematik*. F. Deuticke.

Smirnov, N. V. [N. V.]. (1937). Sur la distribution de w2 (criterium de m.r. von mises). *C. R. Acad. Sci. (Paris), 202*, 449–452.

Mises, R. V. (1928). *Wahrscheinlichkeit, statistik und wahrheit*. J. Springer.

Brown, B. M. (1982). Cramér—von mises distributions and permutation tests. *Biometrika*, *69*(3), 619–624.

Brown, B. M. (1994). Grouping corrections for circular goodness-of-fit tests. *Journal of the Royal Statistical Society. Series B (Methodological)*, *56*(1), 275–283.

Lockhart, R. A., Spinelli, J. J., & Stephens, M. A. (2007). Cramér–von mises statistics for discrete distributions with unknown parameters. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, *35*(1), 125–133.

Pettitt, A. N., & Stephens, M. A. [M. A.]. (1977). The kolmogorov-smirnov goodness-of-fit statistic with discrete and grouped data. *Technometrics*, *19*(2), 205–210.

Choulakian, V., Lockhart, R. A., & Stephens, M. A. (1994). Cramér-von mises statistics for discrete distributions. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, *22*(1), 125–137.

Simpson, P. B. (1951). Note on the estimation of a bivariate distribution function. *The Annals of Mathematical Statistics*, *22*(3), 476–478.

Crutcher, H. L. (1975). A note on the possible misuse of the kolmogorov-smirnov test. *Journal of Applied Meteorology*, *14*(8), 1600–1603. doi: 10.1175/1520-0450(1975)0142.0.CO;2; 25.

Lilliefors, H. W. (1967). On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, *62*(318), 399–402.

Ding, A. A., & Li, Y. (2013). Copula correlation: An equitable dependence measure and extension of pearson's correlation. *arXiv preprint arXiv:1312.7214*.

Dithinde, M., Phoon, K. K., De, W. M., & Retief, J. V. (2011). Characterization of model uncertainty in the static pile design formula. *Journal of Geotechnical and Geoenvironmental Engineering*, *137*(1), 70–85. doi: 10.1061/(ASCE)GT.1943-5606.0000401; 17.

Genest, C., & Rivest, L.-P. (1993). Statistical inference procedures for bivariate archimedean copulas. *Journal of the American Statistical Association*, *88*(423), 1034–1043.

Joe, H. (1997). *Multivariate models and multivariate dependence concepts*. Chapman and Hall/CRC.

Conover, W. J. (1972a). A kolmogorov goodness-of-fit test for discontinuous distributions. *Journal of the American Statistical Association*, *67*(339), 591–596.

Cribbie, R. A., & Keselman, H. J. (2003). The effects of nonnormality on parametric, nonparametric, and model comparison approaches to pairwise comparisons. *Educational and Psychological Measurement*, *63*(4), 615–635. doi: 10.1177/0013164403251283; 17.

An Adjustment of Kolmogorov-Smirnov Test Under Spatial Autocorrelation

Journal of Statistical Planning and Inference

Abstract

Kolmogorov-Smirnov (KS) test is a non-parametric hypothesis test that measures the probability of deviations, that the interested univariate random variable is drawn from a pre-specified distribution (one-sample KS) or has the same distribution as a second random variable (two-sample KS). KS test, as well as other EDF based tests such as Anderson-Darling test and Cramer-von Mises test, have been widely adopted in statistical analysis due to its virtue of more general assumptions compared to parametric test like t-test. However, the independence assumption is one of the very fundamental and easily overlooked assumptions of a statistical model.Without taking care of the effect of correlations between samples, positive linear correlations may result in the conservative estimation of type I error of the KS test and vice versa. In order to address the effects of autocorrelation, I introduced a novel approach of reconstruction of grid map with spherical coordinates. I studied the true distribution of KS statistic under sptial autocorrelation from Monte Carlo simulation and introduced a KS test with spatial adjustment from modelling

on the simulation results. Our KS test with spatial adjustment has a controlled type I error and satisfied power.

Introduction

Kolmogorov-Smirnov test has been a popular test in many fields of applications. It is a non-parametric method under simply settings. It measures the supremum divergence of EDF difference between an interested dataset and the second dataset. By the virtue of its relatively generous on the assumptions of the dataset to be applied, e.g. it is distribution-free which means it does not require knowledge of the samples. The test has been widely appreciated for test the distribution equality. In addition, the EDF test tends to give more power than the χ^2 test. (Pettitt & Stephens, 1977)

The original one-sample and two-sample K-S statistic has the supremum form as followed

$$\begin{aligned} K_n &= \sqrt{n}\sup_x |F_n(X) - G_n(X)| \\ K_{m,n} &= \sqrt{\frac{mn}{m+n}}\sup_{x,y} |F_n(X) - G_m(Y)| \end{aligned}$$

However, the independence assumption is one of the very fundamental and easily overlooked assumptions of a statistical model. Without taking care of the effect of correlations between samples, positive linear correlations may result in the conservative estimation of type I error of the KS test and vice versa (Weiss, 1978). We conducted a comprehensive simulation to study the KS test in its original form on distributions under correlations. Under the significance level of 0.05, we found the KS test in its original form have a uncontrolled small type I error under positive correlations and uncontrolled large type I error under negative type I error (Zheng & et al, 2019a). When the KS test is applied in the spatial analysis, spatial autocorrelation may cause the KS test to have a larger type I error if no adjustments for spatial correlation are applied.

In order to apply the KS test in the right form, adjustments have been studied and proposed. ICC adjustment (N. Cressie, 1992; Kitkungvan et al., 2017). Marc suggested modifying the KS statistic as a function of the original KS statistic and the linear correlation coefficient of r (Weiss, 1978). Adjustment for KS test considering the spatial structure has not been studied.

One of our primary goals in this article is to apply the KS test in analyzing the cardiac PET scans. Therefore, the geometry characteristics of the human heart were studied and a simulated spatial structure was proposed.

First, we need to define a few spatial statistics concepts. Let $S : s_i \in \boldsymbol{R^d}$ be interested location in d-dimensional Euclidean space, $Z(s_i)$ can be viewed as the random process in such location s_i.The notation $z(s_i)$ is defined as a realization of such random process $Z(s_i)$. Without loss of generality, we may assume that the random process $Z(s_i)$ as followed

$$Z(s_i) = \mu + \varepsilon_i$$

Where μ is defined as the mean value of such process and the error term follows a normal distribution, $\varepsilon_i \sim N(0, \sigma^2)$. For the purpose of statistically analyzing the image data, intrinsic stationary distribution is a critical assumption for the spatial random process. The intrinsic stationery is defined as followed

$$\begin{aligned} E(Z(s+h) - Z(s)) &= 0 \\ var(Z(s+h) - Z(s)) &= 2\gamma(h) \end{aligned}$$

where h is the Euclidean distance, $2\gamma(h)$ is an important spatial statistics parameter is known as variogram and $\gamma(h)$ is the semivariogram.

Meanwhile, the second order stationary ensures the distribution of such random process not depend on the location s_i, therefore all realizations across the map were from the same

distribution.

$$E(Z(s_i)) = \mu \tag{3.14}$$

$$cov(Z(s_i + h), Z(s_i)) = C(h) \tag{3.15}$$

where $C(h)$ is the covariogram that only depend on the distance between location s_i and s_j. After $C(h)$ is defined, the autocorrelation structure of such spatial process may be determined.

With the aim of creating a positive-definite covariance structure for the spatial analysis, a valid covariance structure depend on geometry location needs to be defined. Matern (1960) constructed a few valid covariogram models in$\boldsymbol{R^d}, d > 1$. Assumed a valid isotropic covariogram structure in $\boldsymbol{R^3}$.

$$C(h) = \frac{\sigma^2(\frac{\alpha^2||h||}{2})^\nu 2K_\nu(\alpha^2||h||)}{\Gamma(\nu)}, \nu > 0$$

where K_ν is the modified Bessel function of the second kind, $||h||$ is the Euclidean distance. Specifically, $\nu = 1/2$ may yield into a special case

$$C(h) = \sigma^2 exp(-\alpha^2||h||)$$

Methods

The KS test with spatial autocorrelation were found by using the Monte Carlo simulation. In this section, we introduced some methods and elaborated on the procedures we applied.

Cholesky Decomposition Method

With knowledge of covariogram structure Σ, we were able to apply Cholesky decomposition methods to simulate valid autocorrelated data on the interested fields. (N. Cressie, 1992; Golub &

Loan, 2012) In order to get the targeted simulated realizations, we decomposed the covariogram matrix with Cholesky decomposition, in which

$$\Sigma = LL'$$

Where L is a lower triangular $n \times n$ matrix. Then the targeted realizations could be obtained as

$$Z(s) = \mu + LE \tag{3.16}$$

Where E is the error term in matrix form. Note that E is from the identical independent normal distribution with zero mean and unit variance, $E \sim N(0, 1)$. By applying the Cholesky decomposition method, I was able to simulate auto-correlated spatial realizations, with pre-defined covariogram structure, from independent simulated spatial data points.

Moran's I and A Moran's I in Covariogram Form

In order to measure the spatial autocorrelation with a coefficient, Patrick Moran (1950) proposed a spatial autocorrelation coefficient in his paper of Notes on Continuous Stochastic Phenomena in Biometrika. (Moran, 1950)

Give a population of N spatial subjects with random variable X, w_{ij} denotes the preset weight between i^{th} and j^{th} subjects. Moran's I is defined as

$$I = \frac{N}{S} \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij}(x_i - \mu)(x_j - \mu)}{\sum_{j=1}^{N}(x_i - \mu)^2}$$

Where

$$\begin{aligned} S &= \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \\ \mu &= E(X) \end{aligned}$$

With the Cholesky decomposition method from section , we were able to simulate spatially correlated realizations once the covariogram Σ structure is defined. In order to measure the spatial autocorrelation, a more general correlation coefficient is required. However, the original Moran's we were defined as a measurement for realizations, which is inaccessible before simulation. With the purpose of simulating spatially autocorrelated samples with respect to certian Moran's I. With given spatial covariogram known, we used an approximation form of Moran's I with the weighted covariogram matrix.

$$I_A = \frac{N}{W} \frac{\sum_i \sum_j w_{i,j} cov(Z(s_i), Z(s_j))}{\sum_i var(Z(s_i))}$$

where N is the sample size, $w_{i,j}$ is the weight for location s_i and s_j, $W = \sum_i \sum_j w_{i,j}$.

In order to see if I_A generates desired spatially autocorrelated samples in a given spatial space, we have run a Monte Carlo simulation with 10,000 replications. Given the valid variogram for $\boldsymbol{R^3}$,

$$C(h) = \sigma^2 exp(-\alpha^2 ||h||) \tag{3.17}$$

Samples were generated regarding given covariogram 3.17 and spatial structure stated in figure 3.21. The Moran's I in covariogram form was calculated before simulation. The Moran's I in original form for simulated samples were computed after simulation. The Moran's I in covariogram form and the simulated Moran's I were compared in plot 3.13. It shows a satisfied rate of fit.

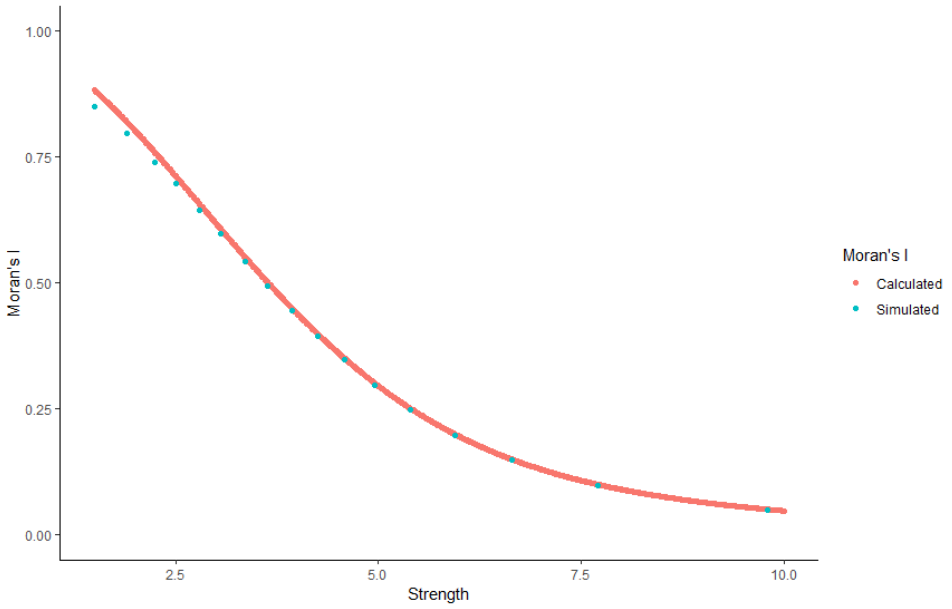


Figure 3.13: I_A vs. Simulated Moran's I

Spatial Coordinates and Geometry Characteristics of Human Heart

The geometry of the heart plays a critical role in the mechanics of cardiology. Back in 1892, Wood has used a spherical coordinate system to mimic the heart shape. Since then the sphericity index system has been popularly used by several studies to reconstruct the shape of the heart. (Mitchell et al., 1992a) Azhari 1998 used a special normalized helical shape descriptor, denoted "geometrical cardiogram", to determine the shape of left ventricular. As the spherical shape has been proved to provide a simulation in shape that is close enough to the heart. (Azhari et al., 1999)

In this study, we focused on the reconstruction of cardiac geometry locations with PET-CT image data. For each PET scan, electric signal values for CFR were recorded in a matrix form with 21 rows and 64 radials. In order to reconstruct the cardiac locations from PET image, we simulated a gridded map with a shape of a truncated ellipsoid, similar to a half football.

Once the simulation shape of heart is decided, we simulated fixed locations D along the fields to represent the electronic recording points in the image location. The nature of gridded

spatial data in $\boldsymbol{R^3}$ can be viewed as a two-way table. (N. Cressie, 1992) Locations $s_i \in D$, D is the subset of $\boldsymbol{R^3}$ and the realization in such location is $Z(s_i)$.

Given the spherical coordinates system

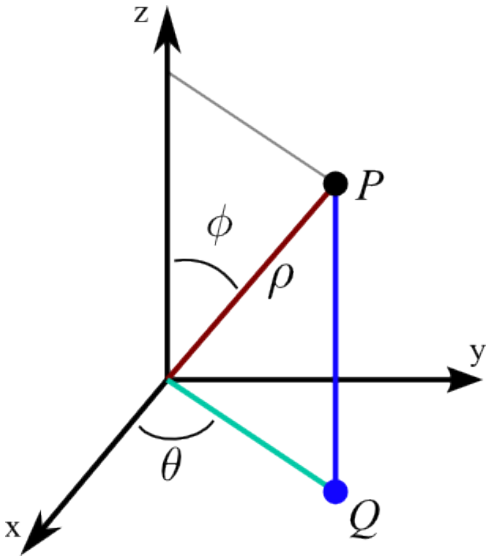


Figure 3.14: Spherical Coordinates

The procedure to generate the 3-D gridded map is as followed

1. Define the radius of the half football we want as

$$\rho = 1.$$

2. Then the define θ on the circle as 64 equal cuts of 2π

$$\Theta = (\theta_1, \theta_2, \ldots, \theta_{64}) = (\frac{1}{32}\pi, \frac{2}{32}\pi, \ldots, 2\pi).$$

3. Similarly define ϕ as 21 equal cuts of $(\pi/2, \pi)$

$$\Phi = (\phi_1, \phi_2, \ldots, \phi_{21}) = (\frac{21}{42}\pi, \frac{22}{42}\pi, \ldots, \frac{41}{42}\pi).$$

4. Transfer spherical coordinates into catesian coordinates

$$\begin{aligned} x &= \rho \sin\phi \cos\theta \\ y &= \rho \sin\phi \sin\theta \\ z &= \rho \cos\phi \end{aligned}$$

The generate 3-D space is realized as followed.

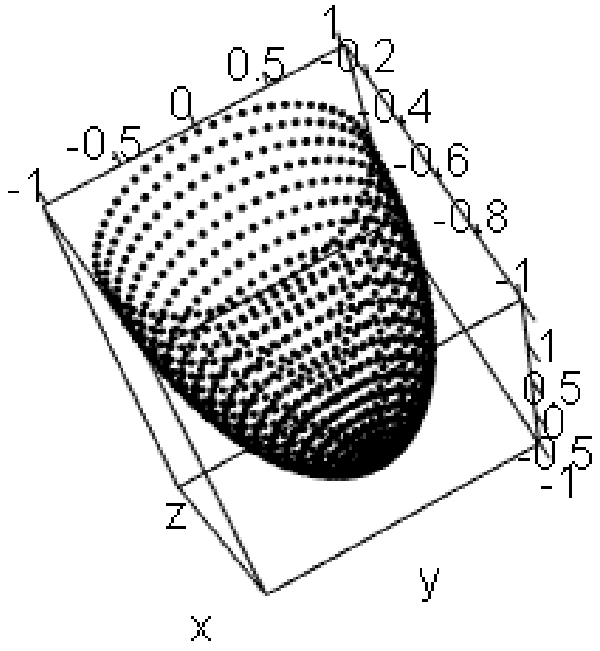


Figure 3.15: Generated Coordinates for Reconstructing PET into Heart shape

After the 3-D space is simulated, the distance between each unique pair of locations may be calculated. I defined the arc length between two locations as the interested distance. The distance between two location $s_i = (x_i, y_i, z_i) = (\rho \sin\phi_i \cos\theta_i, \rho \sin\phi_i \sin\theta_i, \rho \cos\phi_i)$ and $s_j = (x_j, y_j, z_j) = (\rho \sin\phi_j \cos\theta_j, \rho \sin\phi_j \sin\theta_j, \rho \cos\phi_j)$ is defined as

$$Acos = \arccos\left(\cos\phi_i \cos\phi_j + \sin\phi_i \sin\phi_j \cos\left(\theta_i - \theta_j\right)\right) \tag{3.18}$$

$$dist(s_i, s_j) = \begin{cases} \rho \times \arccos\left(1\right), & \text{Acos} \geq 1 \\ \rho \times \arccos\left(-1\right), & \text{Acos} \leq 1 \\ \rho \times Acos, & \text{otherwise} \end{cases} \tag{3.19}$$

The weight function w_{ij} is defined as the squared inverse distance

$$w_{ij} = \frac{1}{(dist(s_i, s_j))^2}$$

The weight matrix **W** is therefore defined as

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1n} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & w_{n3} & \dots & w_{nn} \end{bmatrix} \tag{3.20}$$

After the setup of the spatial environment, the procedure for simulating spatially autocorrelated samples in grid map figure 3.21 is followed as

1. Simulate N samples from $i.i.d$normal distribution$N(\mu, \sigma^2)$. In this study, $N = 1344,\ \mu = (0.5, 1)$ and $\sigma = (0.5, 1, 2)$.

2. Refer to the Cholesky decompsition method in section , calculate L from given covariogram structure Σ.

3. Refer to transformation equation 3.16, transfer N $i.i.d$ samples into the grid with respect the weight matrix **W**.

A KS Test with Spatial Adjustment

Published reports suggested that when the KS test was applied directly without adjustment on the existed spatial autocorrelation will be liberal with an underestimated p-value (Weiss, 1978). Therefore, it is reasonable to assume that a adjustment on the sample size may provide us a closer guess to the truth. The sample size after adjustment is called the informative sample size in this article.

For spatial realizations $y_1, y_2, \ldots, y_n$ of $1^{st}, 2^{nd}, \ldots, n^{th}$ locations. Notice that n is the sample size. Assume

$$Y = \mu + \varepsilon$$

where μ denotes the population mean of Y, ε is the spatially auto-correlated error term independent of μ. We may rewrite the error term in independent form ε^*, and $\varepsilon^* \sim i.i.d.N(0, \sigma^2_{\varepsilon^*})$. let

$$C(Y_i, Y_j) = \sigma^2 V^{-1}$$

then

$$Y = \mu + V^{-\frac{1}{2}}\varepsilon^*$$

where V is the identity matrix, $V = I$, if and only if Y is spatially independent under Gaussian.

Griffith (2005) gave that the expectation of the variance of Y is

$$E(\hat{\sigma^2_Y}) = \frac{\frac{tr(V^{-1})}{n}\sigma^2_\varepsilon}{\frac{tr(V^{-1})}{1^t V^{-1} 1} n}$$

where 1 is the $n \times 1$ matrix of 1, $tr(V^{-1})$ is the trace matrix of V^{-1}.

Then he notes that the informative sample size n^*(the equivalent number of samples without autocorrelation) is

$$n^* = \frac{tr(V^{-1})}{1^t V^{-1} 1} n$$

The approximation of n^* when the spatial realizations Y is normally distributed given the spatial autocorrelation coefficient $\hat{\rho}$ estimated from Spatial autoregressive (SAR) models as followed

$$n^* = n \times [1 - \frac{1}{1 - \exp -1.92} \frac{n-1}{n} (1 - \exp -2.12\hat{\rho} + 0.2\sqrt{\hat{\rho}})] \tag{3.21}$$

where the KS statistic was still obtained as the supremum of the absolute distance between two EDFs.

Another KS test with adjustment for the violation of independence assumption is the ICC adjusted KS test (N. Cressie, 1992). Similar to Griffith's adjustment, the ICC adjusted KS has an adjusted sample size. The KS statistic was still obtained as the supremum of the absolute distance between two EDFs. The informative sample size is defined as:

$$n^* = ICC * n \tag{3.22}$$

With previous knowledge, we assumed a general form that the informative sample size n' with adjustment by the spatial autocorrelation coefficient of Moran's I be

$$n^{'} = n \times \frac{2}{1 + e^{g(I)}}$$

Where $g(I)$ is the function of I, $g(I) = \beta_1 I + \beta_2 I^2 + \cdots + \beta_i I^i$. For the sake of parsimony, I only consider $g(I) = \beta_1 I + \beta_2 I^2 + \beta_3 I^3$.

Therefore to simplify the model I considered

$$A = \frac{n'}{n} = \frac{2}{1 + e^{g(I)}}$$

For j^{th} individual we may have

$$\begin{aligned} A_j &= \frac{n'_j}{n_j} \\ &= \frac{2}{1+e^{\beta_j I_j + \varepsilon_j}} \end{aligned}$$

In order to find the informative sample size and the true distribution of KS statistic under spatial autocorrelation, we used the Monte Carlo procedure as followed.

1. Simulate spatial autocorrelated samples in grid map 3.21 with respect to Moran's I at certain levels. In this study we used Moran's I = (0.2, 0.4, 0.6, 0.8), sample size n = 1344, sample distribution of $N(0,1)$.

2. Compute the KS statistic from simulated samples in step 1.

3. Find the 95 percentile of the KS statistics, denote as KS_{sim} from step 2. Assume KS_{sim} is the critical value of true distribution of KS statistic under spatial autocorrelation at the 95 percentile, find the corresponding sample sizes n'.

After we have obtained the informative sample size n', generalized linear model (GLM) with L1 regularization (Lasso) was used to estimate the βs. The L1 regularization ensured our model with virtue of parsimony by emphasizing on the most influential variables. Assuming a link function $l(A) = \log(\frac{1}{A} - 1)$, the adjustment ratio may be rewrite into the following general linear form

$$E(l(A)) = g(I)$$

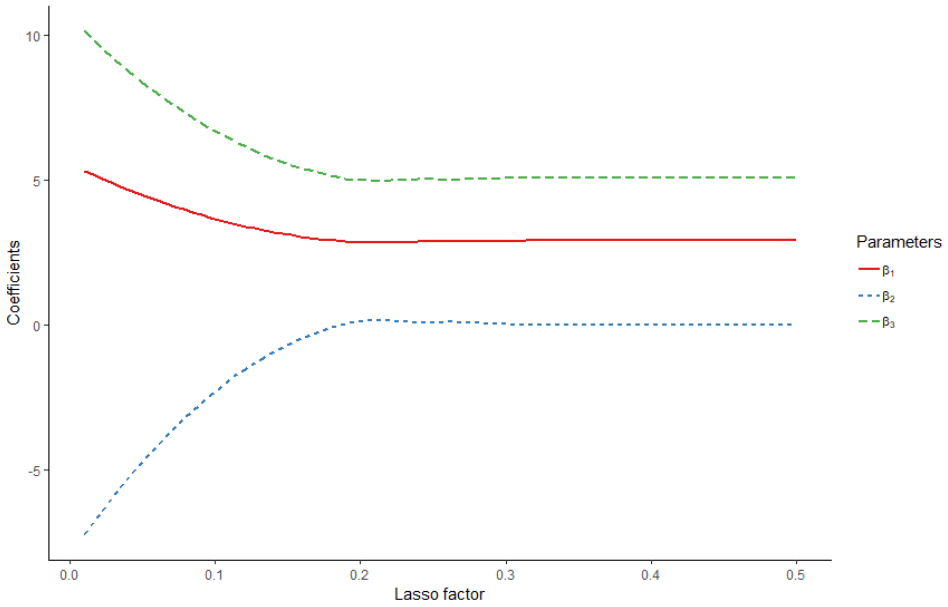


Figure 3.16: GLM with Lasso

After the lasso procedure **??**, we have I and I^3 in the model and I^2 were eliminated from proposed model.

$$g(I) \quad = \quad \beta_1 I + \beta_3 I^3 \tag{3.23}$$

The estimated parameters are as followed,

$$n^{'} = n \times \frac{2}{1 + e^{3.934I + 3.172I^3}} \tag{3.24}$$

The KS statistic with adjustment for spatial autocorrelation is defined as followed

$$\begin{aligned} K^*_{n'} &= \sqrt{n^*} \sup_x |F_n(X) - G_n(X)| \\ K^*_{m',n'} &= \sqrt{\frac{m'n'}{m' + n'}} \sup_{x,y} |F_n(X) - G_m(Y)| \end{aligned}$$

A valid hypothesis test requires controlled type I error rate, which should be near the pre-claimed nominal level. After the type I error is controlled, a satisfied power to discriminate against differences between tested distributions is desired. Therefore, I used type I error under the most popular nominal level of 0.05 and power of my adjusted KS test as benchmarks to evaluate the KS test.

In order to provide a clear picture of how the spatially adjusted KS test performed compared to the other KS type tests. I have evaluated the traditional KS test without spatial autocorrelation adjusted sample size, KS test adjusted with ICC, KS test with Griffith's adjustment and lastly, the KS test with spatial adjustment. The designed nature of image scans limit the sample locations, in other word, the sample size is fixed at 1344. Therefore, the power of KS tests was analyzed for differences in parameters of distributions. I was able to test the distribution change in mean, $\mu = 1 + \Delta$, at the ratio of $0.05, 0.1, 0.2, 0.5, 1$. Same differences ratio was analyzed for the variance, $\sigma = (0.5, 1, 2) + \Delta$.

Results

Type I Error

Moran's I	Test	Parameters (μ, σ^2)			Moran's I	Test	Parameters (μ, σ^2)		
		(1, 0.25)	(1, 1)	(1, 4)			(1, 0.25)	(1, 1)	(1, 4)
0.2	KS	0.169	0.167	0.173	0.6	KS	0.693	0.704	0.699
	KS(1)	0.050	0.052	0.048		KS(1)	0.033	0.037	0.037
	KS(2)	0.064	0.067	0.064		KS(2)	0.197	0.209	0.201
	KS(3)	0.167	0.165	0.172		KS(3)	0.687	0.698	0.694
0.4	KS	0.407	0.411	0.412	0.8	KS	0.928	0.927	0.927
	KS(1)	0.049	0.049	0.053		KS(1)	0.032	0.032	0.032
	KS(2)	0.110	0.110	0.112		KS(2)	0.374	0.381	0.369
	KS(3)	0.402	0.407	0.410		KS(3)	0.921	0.919	0.921

* KS(1) = KS adjusted with Moran's I

* KS(2) = Griffith's adjusted KS

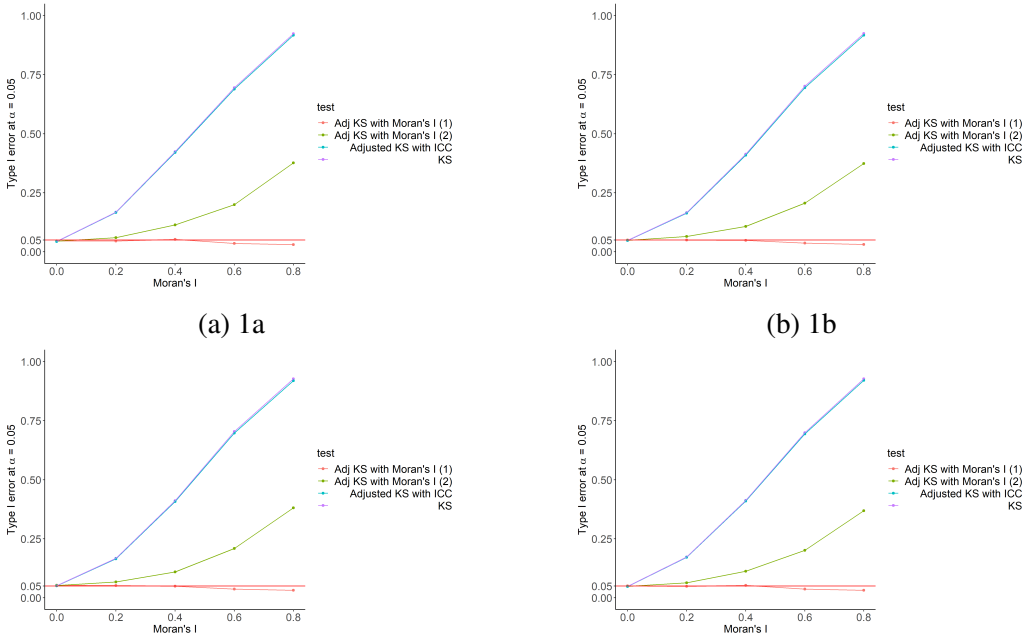
* KS(3) = Adjusted KS with ICC

Table 3.11: Type I Error for Two sample tests of Spatial Normal Distributed Samples

The traditional KS test without any adjustment was unable to achieve the exact type I error when the spatial correlation exists. The type I error for traditional KS test without adjustment and KS test with ICC adjustment have an uncontrolled type I error larger than 0.15 when the Moran's I is 0.2. When the spatial autocorrelation is more serious, a Moran's I of 0.4, the type I error is more than 0.4. The KS tests without adjustment or adjusted by ICC were unable to be used.

KS test with Griffith's adjustment was able to eliminate the unwanted autocorrelation effects when Moran's I is small. When the spatial autocorrelation is more serious, above 0.2, the type I error is liberal.

Our proposed KS statistic with adjustment of Moran's I has proved to have a controlled type I error rate while previous KS statistic from Griffith's tends to have liberal Moran's I when the Moran's I is relatively large. When the Moran's I is small, less than 0.5, we have a type I error of 0.5. When the Moran's I is relatively large, Moran's I larger than 0.6, our proposed test may be rather conservative, with a type I error of 0.03.



(a) 1a (b) 1b

(c) 1c (d) 1d

Figure 3.17: Type I error under the nominal level of 0.05

Power analysis

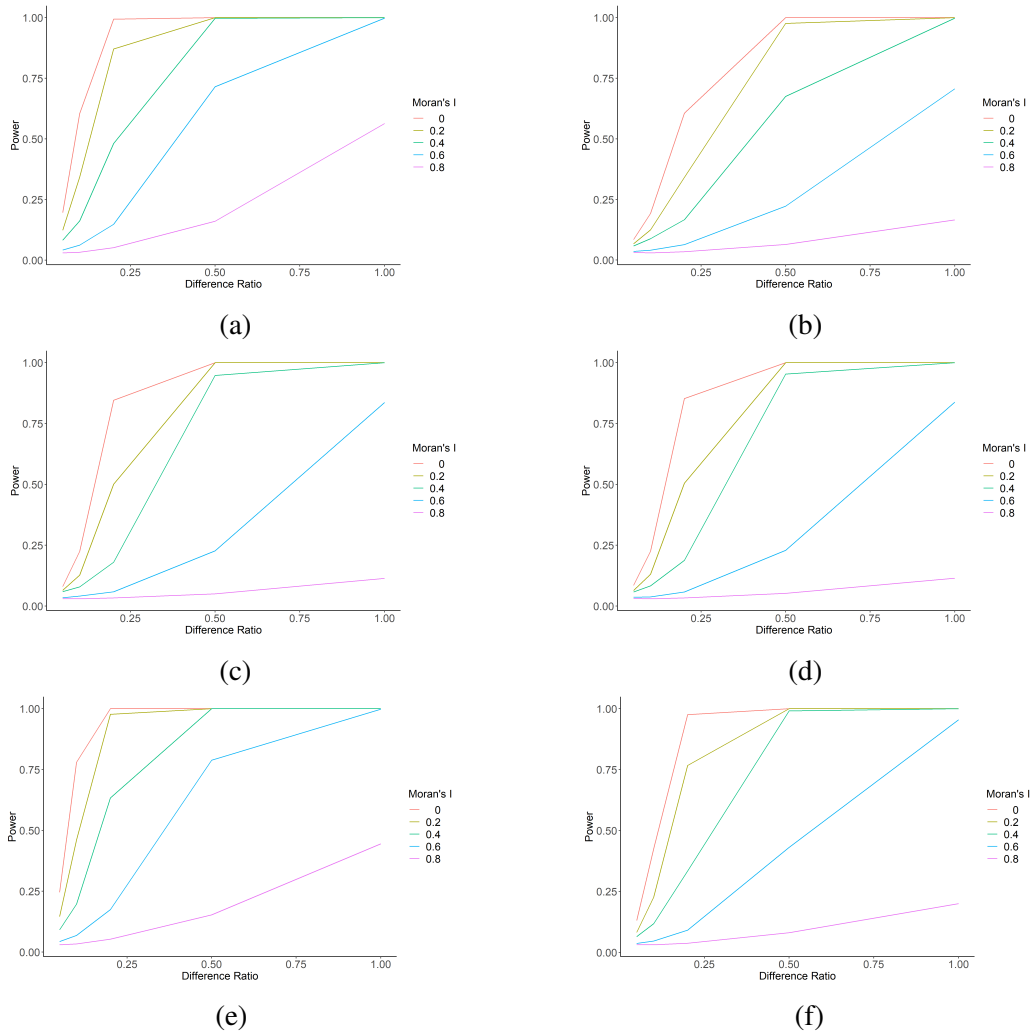
In order to evaluate the ability of rightfully rejecting null hypothesis, we conducted power analysis for the proposed KS test via MC simulation. The power analysis were evaluated on several mean and variance sets to study the performance under different normal distributions.

From the power analysis we may see that the power of our proposed KS test were able to achieve satisfied power.

When the alternative hypothesis are parameters difference in both mean and variance and the Moran's I is moderate, less than 0.2, the proposed test was able to have a power of 0.8 when the parameter difference ratio is 0.1. When the spatial autocorrelation is more serious, Moran's I is 0.4, proposed KS test was able to achieve a power of 0.9 when parameter difference is 0.5, when the parameters difference ratio is 0.2, the power is less than 0.5. When the spatial autocorrelation is very serious with a Moran's I of 0.6, the power of rejecting null when the parameters difference ratio is 0.2 is 0.2, when the parameters difference is 0.5, the power is 0.6. When the spatial autocorrelation is extreme among samples, with a Moran's I of 0.8, then the power is very low and unable to discriminate the null.

When the alternative hypothesis is parameters differences in mean, power was consistent among different variances. Given a relatively weak spatial autocorrelation of 0.2, our proposed KS test was almost as powerful as independent cases. As the spatial autocorrelation increases in samples, the power of our proposed KS test decreased. If the spatial autocorrelation is extremely severe, the proposed test may be unpowerful to discriminate null when it is false.

When the alternative hypothesis is parameters difference in variance, we were able to find a similar conclusion as for when the alternative hypothesis is parameters difference in mean. Given a relatively moderate spatial autocorrelation, Moran's I less than 0.6, then our proposed test was powerful to reject null when the parameter differences are larger than 0.5.



Left column of figures are samples from distribution of N(1, 1), while right samples are from N(1, 4). Figure (a), (b) are the alternative is different variance. Figure (c), (d) are the alternative is different mean. Figure (e), (f) are the alternative is different mean.

Figure 3.18: Power analysis for proposed KS test with spatial autucorrelation adjustment

Discussion and Concluding Remarks

In this paper, we provide a relatively simple way of applying the KS test for samples with spatial autocorrelations. Griffith's adjustment on the informative sample size is specifically for SAR model which may have caused the inadequately shrink in sample size to reflect the true informative samples.

We noticed an uncontrolled type I error in the case of extreme spatial autocorrelation. It was interpreted as even though our KS test was proposed to eliminate the effect of spatial autocorrelation, it may fail when the auto-correlation is extremely large. When the Moran's I is close to 1, the similarities among samples may be too serious. The informative sample size may be too small for the KS test to produce a reasonable result. Our proposed test may serve as a rescue when the spatial independence assumption is violated.

The importance of addressing the right correction correspondence to the correlation structure is self-evident. In our simulation, we have full knowledge of what degree and structure may the Moran's I be. However, in real life data analysis, it may be difficult to identify the exact weight matrix that corresponds to the spatial autocorrelation structure. Therefore, an algorithm that assigns weight automatically based on observed data may need to be studied in future researches.

Future study of adjusting informative sample size for spatial autocorrelation in discrete spatial samples is desired. In the study of the image scan, we find the interested variables were separate in groups. The KS test was rather conservative when tested samples were from grouped or discrete populations. Therefore, our proposed test may direct to conservative type I error. In addition, the Moran's I can only capture the autocorrelation of continuous spatial realizations. The Moran's I may be difficult to apply and uninterpretable when the samples are discrete. D statistic is able to measure the autocorrelation in discrete samples but the null distribution of D statistic is not general and therefore may not be applied directly. In order to solve this issue, a standardized D statistic ranges from -1 to 1 needs to be addressed in future researches.

Meanwhile, multi-dimensional KS tests has been studied. (Justel, Peña, & Zamar, 1997; Fasano & Franceschini, 1987; Peacock, 1983) In the introduction I have suggested that published articles proved that the effectiveness and power for such tests in analyzing images. In future studies, we may focused on proposing a multi-dimensional KS type test with spatial adjustment via Moran's I.

References

Pettitt, A. N., & Stephens, M. A. [M. A.]. (1977). The kolmogorov-smirnov goodness-of-fit statistic with discrete and grouped data. *Technometrics*, *19*(2), 205–210.

Weiss, M. S. (1978). Modification of the kolmogorov-smirnov statistic for use with correlated data. *Journal of the American Statistical Association*, *73*(364), 872–875.

Cressie, N. (1992). *Statistics for spatial data*.

Moran, P. A. P. (1950). Notes on continuous stochastic phenomena. *Biometrika*, *37*(1/2), 17–23.

Mitchell, G. F., Lamas, G. A., Vaughan, D. E., & Pfeffer, M. A. (1992a). Left ventricular remodeling in the year after first anterior myocardial infarction: A quantitative analysis of contractile segment lengths and ventricular shape. ID: 271027.

Azhari, H., Beyar, R., & Sideman, S. (1999). On the human left ventricular shape. *Computers and Biomedical Research, an International Journal*, *32*(3), 264–282.

Zheng, W., & et al. (2019a). A simulation study of a class of nonparametric test statistics: A close look of continuous, discrete and correlated variables.

Kitkungvan, D., Lai, D., Zhu, H., Roby, A. E., Johnson, N. P., Steptoe, D. D., … Gould, K. L. (2017). Optimal adenosine stress for maximum stress perfusion, coronary flow reserve, and pixel distribution of coronary flow capacity by kolmogorov–smirnov analysis. *Circulation: Cardiovascular Imaging*, *10*(2), e005650.

Golub, G. H., & Loan, C. F. V. (2012). *Matrix computations*. JHU Press.

Justel, A., Peña, D., & Zamar, R. (1997). A multivariate kolmogorov-smirnov test of goodness of fit. *Statistics & Probability Letters*, *35*(3), 251–259. ID: 271498.

Fasano, G., & Franceschini, A. (1987). A multidimensional version of the kolmogorov-smirnov test. *Monthly Notices of the Royal Astronomical Society*, *225*, 155–170.

Peacock, J. A. (1983). Two-dimensional goodness-of-fit testing in astronomy. *Monthly Notices of the Royal Astronomical Society*, *202*, 615–627.

Comparing Heart PET Scans: A Revision of Komogorov-Smirnov Test

Computational Statistics & Data Analysis

Abstract

Kolmogorov-Smirnov (KS) test has been a popular test in many fields of applications. Published papers have confirmed the efficiency of KS test being applied in the imaging process, histogram analysis and PET/CT scan analysis. However, the independence assumption is one of the very fundamental and easily overlooked assumptions of a statistical model. Without taking care of the effect of correlations between samples, positive linear correlations may result in the conservative estimation of type I error of the KS test and vice versa. When the KS test is applied in the spatial analysis, spatial autocorrelation may cause the KS test to have a larger type I error if no adjustments for spatial correlation are applied. We revisited a trial comparing the efficiency of regadenoson under different timeing and dipyridamole by the Weatherhead PET Imaging Center in Houston. In order to study the PET scans with spatial autocorrelation, we have introced a novel way of reconstructing the shape of human heart by using spherical coordinates. Meanwhile, the KS test in its original form does not have a controlled type I error and therefore we used the KS test with spatial adjustment. We compared the KS test with spatial

adjustment with other KS test with adjustment for correlation. The results showed that the KS test with spatial adjustment has a controlled type I error and a satisfied power.

Introduction

In order to integrate the CFR with absolute blood flow, a new concept was approved by the Food and Drug Administration (FDA) on September 22, 2017. The approval was based on the comprehensive scientific review from 2012 to 2017. Several published reports validated the concept and proved its effects to be treated as a biomarker for CVD diagnosis (K. Lance Gould & Johnson, 2018).

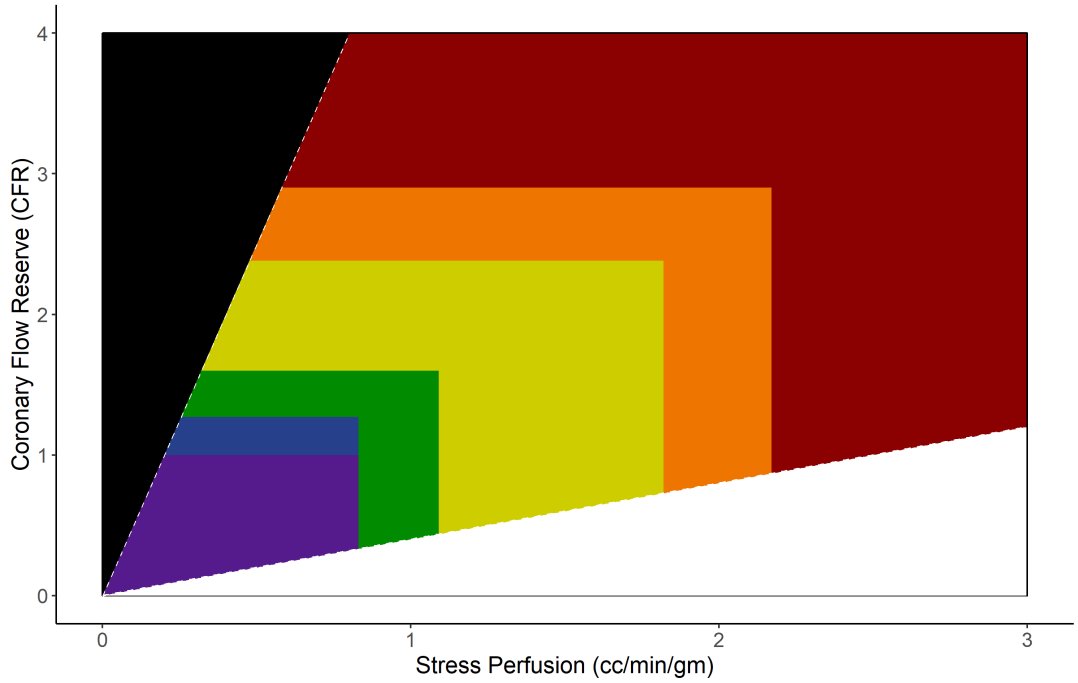


Figure 3.19: CFC Scatter Plot of CFR versus Absolute Stress Flow

CFC	CFR	Stress perfusion	Color Code
Excellent	$CFR > 2.9$	$perfusion > 2.17$	Red
Typical	$2.9 \geq CFR > 2.38$	$2.17 \geq perfusion > 1.82$	Orange
Mildly reduced	$2.38 \geq CFR > 1.6$	$1.82 \geq perfusion > 1.09$	Yellow
Moderately reduced	$1.6 \geq CFR > 1.27$	$1.09 \geq perfusion > 0.83$	Green
Severely reduced	$1.27 \geq CFR > 1$	$0.83 \geq perfusion$	Blue
Myocardial steal	$CFR < 1$	$0.83 \geq perfusion$	Purple

Table 3.12: Coronary flow capacity

From the table 3.12 and figure 3.19, we know that when CFR is larger than 2.9 $(ml/g/min)$ or stress perfusion $>$ 2.17 then the CFC is coded as excellent and the color code is red, when the CFR from 2.38 to 2.9 or the perfusion is from 1.82 to 2.17 then the CFC is coded as typical and the color code is orange, when the CFR is from 1.6 to 2.38 or the stress perfusion from 1.09 to 1.82 then the CFC is coded as mildly reduced and color code is yellow, when the CFR is from 1.27 to 1.6 or the perfusion from 0.83 to 1.09 then the CFC is recorded as moderately reduced and the color is coded as green, when the CFR is from 1 to 1.27 or the perfusion is less than 0.83 then the CFC is coded as severely reduced and the denoting color is blue, lastly when CFR is less than 1, the CFC is coded as myocardial steal and the color code is purple. The triangle in the upper left and bottom with black and white color were the lower limit of rest flow for viability and the upper limit of clinically observed rest flow, respectively.

Kolmogorov-Smirnov test has been a popular test in many fields of applications. It is a non-parametric method under simply settings. It measures the supremum divergence of EDF difference between an interested dataset and the second dataset. By the virtue of its relatively generous on the assumptions of the dataset to be applied, e.g. it is distribution-free which means it does not require knowledge of the samples. The test has been widely appreciated for test the

distribution equality. In addition, the EDF test tends to give more power than the χ^2 test. (Pettitt & Stephens, 1977)

The original one-sample and two-sample K-S statistic has the supremum form as followed

$$\begin{aligned} K_n &= \sqrt{n}\sup_x |F_n(X) - G_n(X)| \\ K_{m,n} &= \sqrt{\frac{mn}{m+n}}\sup_{x,y} |F_n(X) - G_m(Y)| \end{aligned}$$

Kolmogorov-Smirnov test has been used to discriminate image difference. Published papers have confirmed the efficiency of KS test being applied in the imaging process and histogram analysis (Lampariello, 2000). Lim showed that the KS test has relatively higher power compared to Wilcoxon and t-test when the variation is relatively large (Lim & Jang, 2002). Geman used KS test for discriminating homogeneous maps by pixel gray levels distribution (Geman et al., 1990). The interpretation ability rendered its favourable position in clinical fields. Clinically, published reports suggested that KS test were valid for analyzing MR scans comparison (Chen et al., 2006; F. Baselice, 2017; Rajan et al., 2014). Kipritidis used KS test for CT/PET scans and Brook applied histogram analysis with KS for spectral CT scans to evaluate the artifacts reduction (Kipritidis et al., 2016; Brook et al., 2012).

However, the independence assumption is one of the very fundamental and easily overlooked assumptions of a statistical model. Without taking care of the effect of correlations between samples, positive linear correlations may result in the conservative estimation of type I error of the KS test and vice versa (Weiss, 1978). When the KS test is applied in the spatial analysis, spatial autocorrelation may cause the KS test to have a larger type I error if no adjustments for spatial correlation are applied.

Under positive spatial autocorrelation, the locations closer tend to be similar and dependent, locations further away tend be more independent. Therefore, the sample size in effect under spatial autocorrelation may be different from the original sample size (N. Cressie, 1992). We

called the true sample size under spatial autocorrelation as informative sample size $n^{'}$. In order to adjust for the spatial autocorrelation, we worked out the KS test with spatial adjustment (Zheng & et al, 2019b).

$$n^{'} = n \times \frac{2}{1 + e^{3.934I + 3.172I^3}} \tag{3.25}$$

The KS statistic with adjustment for spatial autocorrelation is defined as followed

$$\begin{aligned} K^{*}_{n'} &= \sqrt{n^{*}} \sup_{x} |F_n(X) - G_n(X)| \\ K^{*}_{m',n'} &= \sqrt{\frac{m'n'}{m' + n'}} \sup_{x,y} |F_n(X) - G_m(Y)| \end{aligned}$$

The other popular test in analyzing the PET scan is the t-test (Kershah et al., 2013).

$$t = (\bar{X} - \mu)/(\frac{\sigma}{\sqrt{n}})$$

where $\bar{X}$ is the sample mean of $X: \ x_1, x_2, \ldots, x_n$, σ is the standard deviation and μ is the population/hypothesized mean. The most used type of t-test used is the paired t-test **??**.

$$t = (\bar{X}_d - 0)/(\frac{\sigma_d}{\sqrt{n}})$$

where $\bar{X}_d$ is the sample mean of the difference of paired samples $X_d: \ (x_{1,1} - x_{2,1}), (x_{1,2} - x_{2,2}), \ldots, (x_{1,n} - x_{2,n})$, σ_d is the standard deviation of the paired differences.

In order to provide analysis on the cardiac PET scans. We applied the KS test with spatial adjustment via Moran's I on the averaged pixel distribution of CFC and compared the results from t-test in its original form.

Methods

The geometry of the heart plays a critical role in the mechanics of cardiology. Back in 1892, Wood has used a spherical coordinate system to mimic the heart shape. Since then the sphericity index system has been popularly used by several studies to reconstruct the shape of heart (Mitchell, Lamas, Vaughan, & Pfeffer, 1992b). Azhari (1999) used a special normalized helical shape descriptor, denoted "geometrical cardiogram", to determine the shape of left ventricular.(Azhari et al., 1999) As the spherical shape has been proved to provide a simulation in shape that is close enough to the heart. (Hansen, Marinucci, Natoli, & Vittorio, 2002)

In this study, we focused on the reconstruction of cardiac geometry locations with PET-CT image data. For each PET scan, electric signal values for CFR were recorded in a matrix form with 21 rows and 64 radials. In order to reconstruct the cardiac locations from PET image, we simulated a gridded map with a shape of a truncated ellipsoid, similar to a half football.

Once the simulation shape of heart is decided, we simulated fixed locations D along the fields to represent the electronic recording points in the image location. The nature of gridded spatial data in $\boldsymbol{R^3}$ can be viewed as a two-way table. (N. Cressie, 1992) Locations $s_i \in D$, D is the subset of $\boldsymbol{R^3}$ and the realization in such location is $Z(s_i)$.

Given the spherical coordinates system

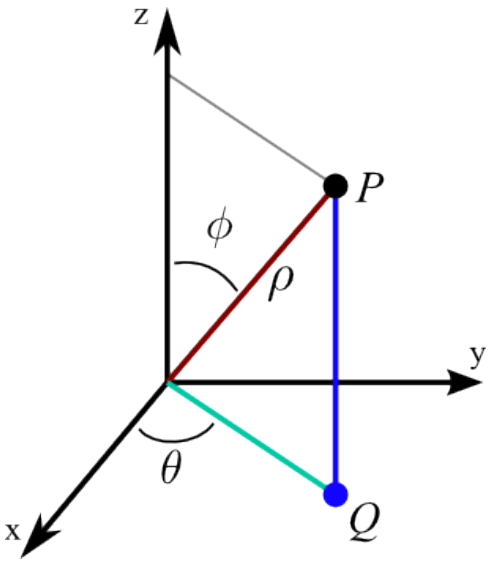


Figure 3.20: Spherical Coordinates

The procedure to generate the 3-D gridded map is as followed

1. Define the radius of the half football we want as

$$\rho = 1.$$

2. Then the define θ on the circle as 64 equal cuts of 2π

$$\Theta = (\theta_1, \theta_2, \ldots, \theta_6 4) = (\frac{1}{32}\pi, \frac{2}{32}\pi, \ldots, 2\pi).$$

3. Similarly define ϕ as 21 equal cuts of $(\pi/2, \pi)$

$$\Phi = (\phi_1, \phi_2, \ldots, \phi_2 1) = (\frac{21}{42}\pi, \frac{22}{42}\pi, \ldots, \frac{41}{42}\pi).$$

4. Transfer spherical coordinates into Cartesian coordinates

$$\begin{aligned} x &= \rho \sin\phi \cos\theta \\ y &= \rho \sin\phi \sin\theta \\ z &= \rho \cos\phi \end{aligned}$$

The generate 3-D space is realized as followed.

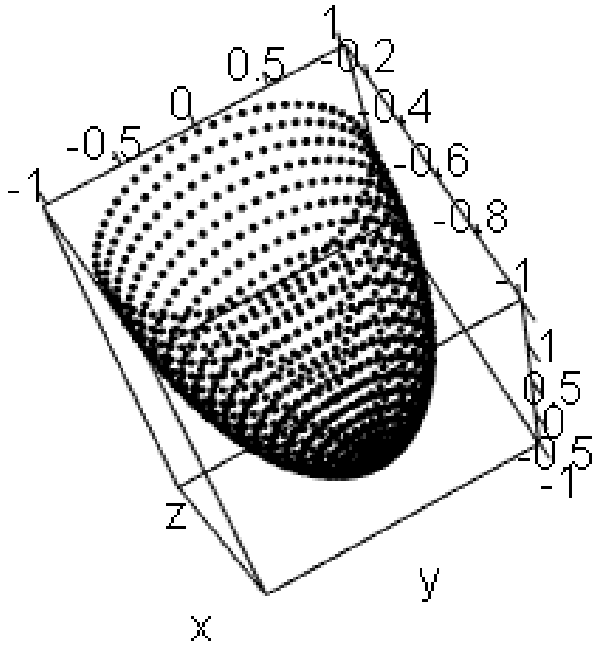


Figure 3.21: Generated Coordinates for Reconstructing PET into Heart shape

After the 3-D space is simulated, the distance between each unique pair of locations may be calculated. We defined the arc length between two locations as the interested distance. The distance between two location $s_i = (x_i, y_i, z_i) = (\rho \sin\phi_i \cos\theta_i, \rho \sin\phi_i \sin\theta_i, \rho \cos\phi_i)$ and $s_j = (x_j, y_j, z_j) = (\rho \sin\phi_j \cos\theta_j, \rho \sin\phi_j \sin\theta_j, \rho \cos\phi_j)$ is defined as

$$Acos = \arccos\left(\cos\phi_i \cos\phi_j + \sin\phi_i \sin\phi_j \cos\left(\theta_i - \theta_j\right)\right) \tag{3.26}$$

$$dist(s_i, s_j) = \begin{cases} \rho \times \arccos\left(1\right), & \text{Acos} \geq 1 \\ \rho \times \arccos\left(-1\right), & \text{Acos} \leq 1 \\ \rho \times Acos, & \text{otherwise} \end{cases} \tag{3.27}$$

The weight function w_{ij} is defined as the squared inverse distance

$$w_{ij} = \frac{1}{(dist(s_i, s_j))^2}$$

The weight matrix **W** is therefore defined as

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1n} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & w_{n3} & \dots & w_{nn} \end{bmatrix} \tag{3.28}$$

After the reconstruction 3.21 is finished, PET scan data will be put into the coordinates in respect to the column and row order. Spatial autocorrelation coefficient can be computed therefore.

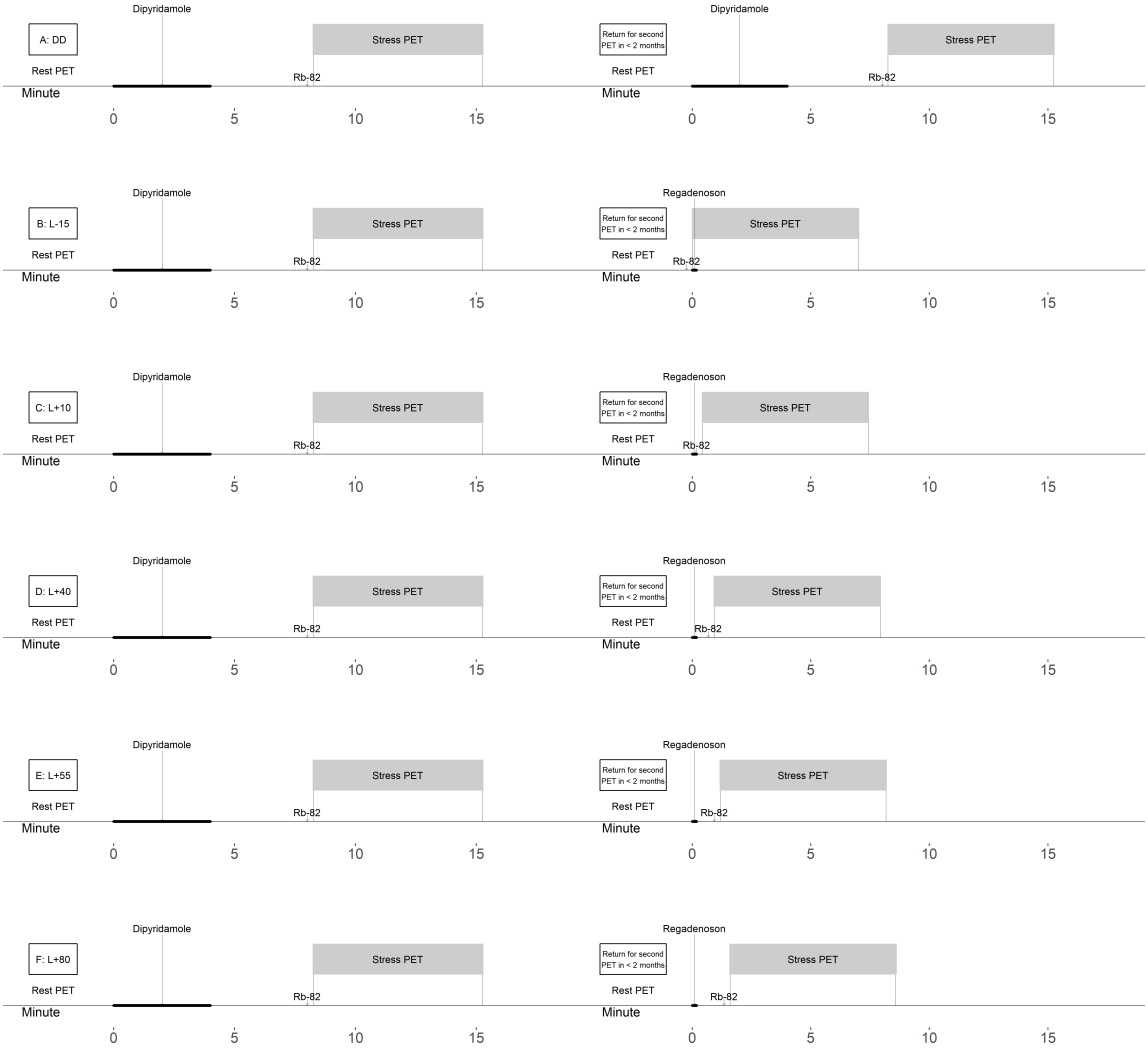
Data Collection

Recruited subjects were split into 6 groups, each group went through a two-stage PET imaging procedure. The first group of subjects was administered with dipyridamole in both the first stage and second stage of PET scans. The second group of subjects was administered with the procedure of Rb-82 activated 15s before injection of regadenoson in one stage and with dipyridamole in the other stage. Similarly, the third, fourth, fifth and sixth group of subjects

were administered regadenoson with a certain time of activation of Rb-82 in one stage and administered with dipyridamole in the other stage.

Different dipyridamole protocol timing has been studied. Researchers applied the current optimal protocol of 4 mins dipyridamole protocol in the trail.(Harel et al., 2018) According to the dipyridamole protocol guideline, dipyridamole ($142ug/kg/min$) was infused for 4 min. After dipyridamole is infused, Rb-82 generator was activated. PET stress scan starts 15s after Rb-82 generator activation.

Regadenoson protocol indicates that a single-use, pre-filled, 5-ml syringe of regadenoson was administered for 10s via a peripheral vein. Time of Rb-82 generator activation varies by protocols. Similarly, 10s after Rb-82 generator activation, PET scan was performed.



Notes: The bold black line in the timeline denotes the duration of medication, either dipyridamole or regadenoson, infusion. Protocols in the left is the baseline with dipyridamole, protocols in the right are study group with dipyridamole, Rb-82 activated 15s before regadenoson administration and Rb-82 activated 10s/40s/55s/80s after regadenoson administration.

Figure 3.22: Description of Protocols

Protocol	Description
DD	Repeated dipyridamole
L - 15	Regadenoson group with Rb-82 activated 15 seconds prior to injection of regadenoson
L + 10	Regadenoson group with Rb-82 activated 10 seconds after injection of regadenoson
L + 40	Regadenoson group with Rb-82 activated 40 seconds after injection of regadenoson
L + 55	Regadenoson group with Rb-82 activated 55 seconds after injection of regadenoson
L + 80	Regadenoson group with Rb-82 activated 80 seconds after injection of regadenoson

Table 3.13: Protocols

The protocol for the trial is described in figure 3.22 and table 3.13. In this single-subject design, subjects using dipyridamole was used as the baseline and compared with themselves using either dipyridamole repeatedly in DD protocol or using regadenoson in L-15, L+10, L+40, L+55, L+80.

Statistical Analysis

Statistical analysis was conducted with R version 3.5.1(The R Foundation for Statistical Computing Platform: x86_64-w64-mingw32/x64 (64-bit)). Descriptive tables including means, standard deviations, percentages, and p-values will be presented. For the categorical variable, multiple chi-squared tests will be applied. For variables with counts less than 5, a Fisher's exact test will be applied. For continuous variables, t-tests will be carried out.

Frequency plots for the averaged pixel distribution of CFC were presented for each protocol. In addition, cumulative frequency plots for the averaged pixel distribution of CFC were presented for each protocol. The primary approach to analyze the PET scans is to evaluate the differences in the averaged pixel distribution of CFC for baseline and test protocols via spatially adjusted

KS test. In addition, in order to evaluate the traditional approaches, we conducted a comparison for a paired t-test, original KS test, KS test with ICC adjustment and the spatially adjusted KS test. P-values for each test were reported and analyzed.

Results

There were 188 patients recruited in the trial and 176 of them finished the trial. Exclusions of subjects include 7 subjects had severe side effects, intravenous access of 2 subjects were unable to be obtained and another 2 subjects had other reasons. Table 3.15 shows the number of patients in each protocol, demographic, clinical and relative PET uptake results.

The test for age and BMI were significant. However, we could see that the differences were small from mean and standard deviation. Subjects have similar risk factors and history conditions including smoking, myocardial infarction (MI), hypertension, dyslipidemia, diabetes, cardiac catheterization, percutaneous intervention (PCI) or bypass surgery (CABG). The percentage of interested medication used were comparable. For the baseline cardiac characters, there were statistically significant differences across protocols for cholesterol and low-density lipoprotein cholesterol (LDL). No significant difference was detected from low-density lipoprotein cholesterol (HDL). We noticed a relatively high percentage of missing in cholesterol (32.10%), LDL (33.24%) and HDL (32.10%). The PET uptake was consistent across protocols. In addition, significant differences in rest heart rate and stress heart rate were reported. We noticed that the L-15 protocol was having lower rest and stress heart rate.

Table 3.15: Descriptive Table

	Population	Protocols						P-value
		DD	L-15	L + 10	L + 40	L + 55	L + 80	
Clinical characteristics								
Age	60 ± 9	62 ± 10	64 ± 8	57 ± 10	61 ± 7	60 ± 10	58 ± 6	0.02
BMI	29 ± 5	28 ± 5	27 ± 5	28 ± 4	30 ± 4	28 ± 5	31 ± 6	< 0.001
Risk factors and history								
Smoking	52(0.3)	16(0.32)	3(0.2)	17(0.34)	5(0.33)	9(0.29)	2(0.13)	0.66

MI	15(0.09)	4(0.08)	3(0.2)	4(0.08)	1(0.07)	2(0.06)	1(0.07)	0.72
Hypertension	81(0.46)	23(0.46)	10(0.67)	21(0.42)	7(0.47)	14(0.45)	6(0.4)	0.68
Dyslipidemia	132(0.75)	40(0.8)	10(0.67)	34(0.68)	14(0.93)	24(0.77)	10(0.67)	0.32
Diabetes	17(0.1)	7(0.14)	3(0.2)	4(0.08)	1(0.07)	2(0.06)	0(0)	0.39
Catheterization	38(0.22)	12(0.24)	4(0.27)	10(0.2)	5(0.33)	4(0.13)	3(0.2)	0.68
PCI	28(0.16)	8(0.16)	5(0.33)	8(0.16)	3(0.2)	1(0.03)	3(0.2)	0.19
CABG	8(0.05)	3(0.06)	2(0.13)	2(0.04)	0(0)	1(0.03)	0(0)	0.48
Medications								
Statin	89(0.51)	25(0.5)	10(0.67)	23(0.46)	11(0.73)	15(0.48)	5(0.33)	0.23
ACEI/ARB	48(0.27)	14(0.28)	3(0.2)	12(0.24)	7(0.47)	9(0.29)	3(0.2)	0.55
Antiplatelet	85(0.48)	17(0.34)	9(0.6)	27(0.54)	9(0.6)	17(0.55)	6(0.4)	0.20
Beta Blocker	50(0.28)	15(0.3)	8(0.53)	11(0.22)	5(0.33)	7(0.23)	4(0.27)	0.27
Diuretic	25(0.14)	7(0.14)	3(0.2)	7(0.14)	3(0.2)	4(0.13)	1(0.07)	0.91
Calcium blockers	14(0.08)	3(0.06)	1(0.07)	4(0.08)	2(0.13)	4(0.13)	0(0)	0.67
Nitrate	3(0.02)	1(0.02)	0(0)	1(0.02)	1(0.07)	0(0)	0(0)	0.65
Baseline Cardiac								
Cholesterol	180 ± 46	183 ± 50	153 ± 42	179 ± 38	155 ± 44	193 ± 43	216 ± 45	0.01
LDL	100 ± 36	102 ± 36	84 ± 30	98 ± 35	85 ± 39	105 ± 31	136 ± 32	0.01
HDL	54 ± 16	51 ± 16	54 ± 16	54 ± 14	50 ± 15	62 ± 19	51 ± 16	0.21
Rest Systolic blood pressure	115 ± 17	119 ± 19	117 ± 16	113 ± 16	114 ± 15	115 ± 16	112 ± 12	0.59
Rest Diastolic blood pressure	65 ± 10	68 ± 10	63 ± 10	63 ± 9	67 ± 14	64 ± 12	68 ± 6	0.26
Rest Heart Rate	63 ± 11	61 ± 10	60 ± 10	63 ± 11	64 ± 13	65 ± 12	66 ± 14	0.37
Stress Systolic blood pressure	119 ± 15	122 ± 17	111 ± 15	117 ± 15	121 ± 13	120 ± 15	120 ± 14	0.21
Stress Diastolic blood pressure	63 ± 10	64 ± 9	57 ± 12	61 ± 9	65 ± 14	64 ± 11	63 ± 8	0.19
Stress Heart Rate	89 ± 13	87 ± 13	83 ± 13	90 ± 13	92 ± 13	91 ± 13	93 ± 15	0.17
Non-baseline Cardiac								
Cholesteral	180 ± 46	185 ± 50	158 ± 43	178 ± 39	155 ± 44	193 ± 42	205 ± 46	0.03
LDL	100 ± 36	103 ± 36	87 ± 30	97 ± 36	85 ± 39	107 ± 31	127 ± 36	0.04
HDL	54 ± 17	51 ± 16	56 ± 16	55 ± 16	50 ± 15	61 ± 19	50 ± 15	0.29
Rest Systolic blood pressure	117 ± 16	117 ± 15	116 ± 18	116 ± 17	116 ± 24	117 ± 13	117 ± 14	0.99
Rest Diastolic blood pressure	67 ± 11	67 ± 9	63 ± 9	66 ± 12	68 ± 14	67 ± 9	70 ± 10	0.61
Rest Heart Rate	63 ± 12	60 ± 10	59 ± 8	65 ± 13	61 ± 9	67 ± 12	68 ± 15	0.03
Stress Systolic blood pressure	119 ± 19	120 ± 14	111 ± 18	119 ± 22	114 ± 21	124 ± 19	122 ± 18	0.29
Stress Diastolic blood pressure	62 ± 12	64 ± 10	61 ± 14	60 ± 14	62 ± 14	62 ± 11	63 ± 9	0.68
Stress Heart Rate	91 ± 15	85 ± 15	82 ± 12	96 ± 15	88 ± 11	98 ± 14	93 ± 13	< 0.001

Continuous variables were presented as mean $\pm$ standard deviation, categorical variables were presented as count(percentage)

BMI in kg per m^2

Systolic/Diastolic Blood pressure in mm Hg

Heart rate in beats per minute

Table 3.16 lists the averaged rest perfusion, averaged stress perfusion and averaged CFR. It was clear that the rest perfusion for subjects in non-base condition and base condition is comparable. This indicates no significant effects other than protocol difference existed. As we expected, the stress perfusion for subjects using dipyridamole in the baseline group and subjects using different timing protocols of regadenoson were different. Subjects using dipyridamole have relatively higher stress perfusions. The trends in averaged CFR were similar to stress

perfusion. Subjects with dipyridamole had relatively higher CFR. A weak but noticeable positive correlation could be spotted between Rb-82 activation time and CFR. In other word, subjects in protocol with Rb-82 activated later tended to have a higher CFR.

Table 3.17 reported the p-values from paired t-test and KS test with spatial adjustment. From p-value we can make similar conclusion we had in table 3.16. We may see that the spatial adjusted KS were more sensitive than the paired t-test. The paired t-test analyzed the global CFR and global flow and therefore minor differences were overlooked.

Protocol	Rest Perfusion			Stress Perfusion			CFR		
	Non-Base	Base	Δ	Non-Base	Base	Δ	Non-Base	Base	Δ
DD	0.79 ± 0.28	0.81 ± 0.27	-0.02 ± 0.2	2.13 ± 0.7	2.22 ± 0.65	-0.09 ± 0.46	2.78 ± 0.73	2.86 ± 0.76	-0.09 ± 0.7
L-15	0.73 ± 0.22	0.76 ± 0.23	-0.02 ± 0.18	1.3 ± 0.46	1.87 ± 0.61	-0.57 ± 0.4	1.78 ± 0.48	2.52 ± 0.73	-0.74 ± 0.75
L + 10	0.79 ± 0.28	0.78 ± 0.25	0.01 ± 0.24	1.71 ± 0.52	2.15 ± 0.61	-0.44 ± 0.48	2.25 ± 0.55	2.88 ± 0.79	-0.63 ± 0.72
L + 40	0.77 ± 0.24	0.76 ± 0.23	0.01 ± 0.22	1.79 ± 0.52	2.1 ± 0.55	-0.31 ± 0.38	2.43 ± 0.65	2.87 ± 0.69	-0.43 ± 0.79
L + 55	1.01 ± 0.37	0.96 ± 0.34	0.05 ± 0.21	2.28 ± 0.68	2.49 ± 0.71	-0.21 ± 0.42	2.36 ± 0.61	2.73 ± 0.78	-0.36 ± 0.77
L + 80	0.89 ± 0.32	0.87 ± 0.35	0.02 ± 0.23	2.14 ± 0.56	2.43 ± 0.74	-0.28 ± 0.49	2.53 ± 0.66	2.91 ± 0.65	-0.39 ± 0.56

Δ: The difference between base and Non-Base.

Table 3.16: Averaged Rest Flow, Averaged Stress Flow and Averaged CFR by Protocol

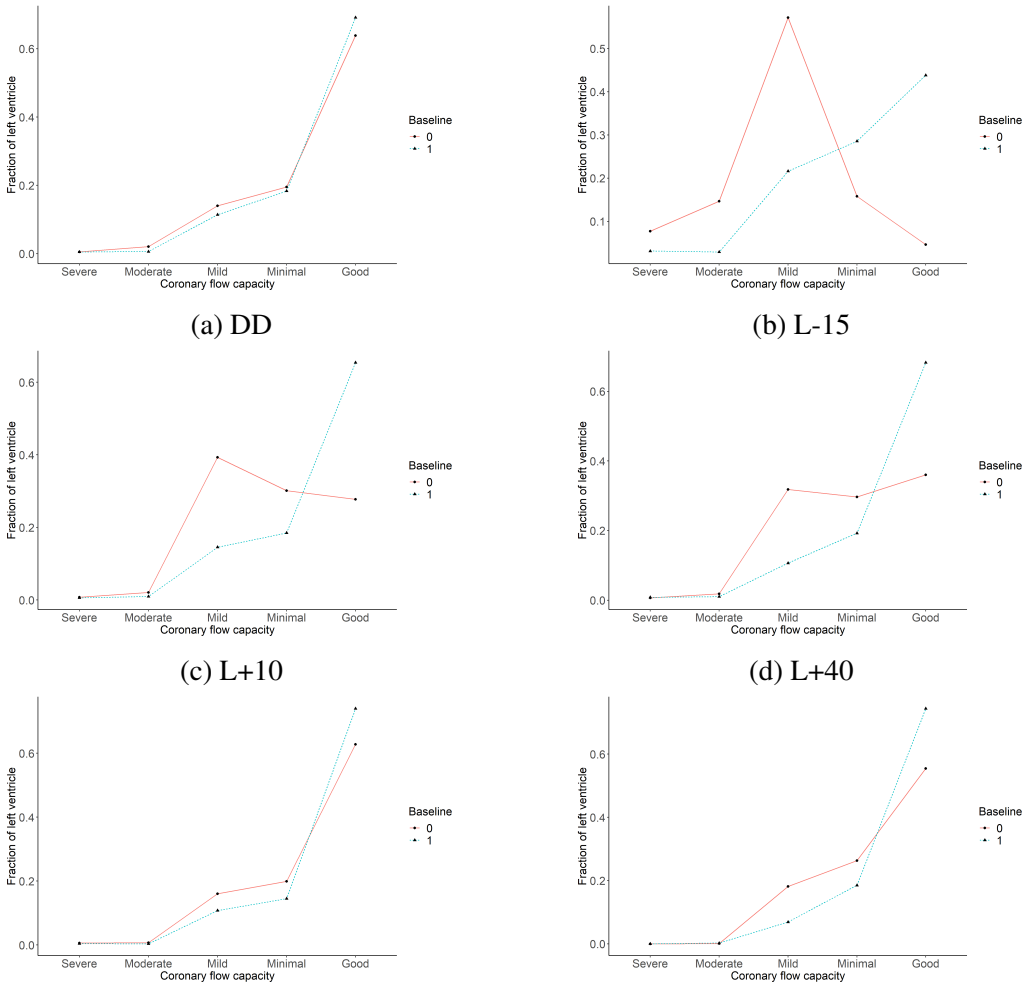
Protocol	Rest Perfusion		Stress Perfusion		CFR	
	Paired t-test	Spatial KS	Paired t-test	Spatial KS	Paired t-test	Spatial KS
DD	0.483	0.288	0.094	0.004^{**}	0.221	$< 10^{-7***}$
L-15	0.589	0.285	$< 0.001^{**}$	$< 10^{-16***}$	$< 0.001^{**}$	$< 10^{-16***}$
L+10	0.691	0.635	$< 10^{-10***}$	$< 10^{-16***}$	$< 10^{-10***}$	$< 10^{-16***}$
L+40	0.879	0.361	$< 0.001^{**}$	$< 10^{-16***}$	0.013^{*}	$< 10^{-16***}$
L+55	0.105	0.002^{**}	0.001^{**}	$< 10^{-9***}$	0.004^{**}	$< 10^{-16***}$
L+80	0.676	0.384	0.019^{*}	$< 10^{-13***}$	$< 0.001^{**}$	$< 10^{-16***}$

* p-value < 0.05

** p-value < 0.005

*** p-value < 0.0005

Table 3.17: P - values from Paired t-test and Spatially Adjusted KS test



(a) DD

(b) L-15

(c) L+10

(d) L+40

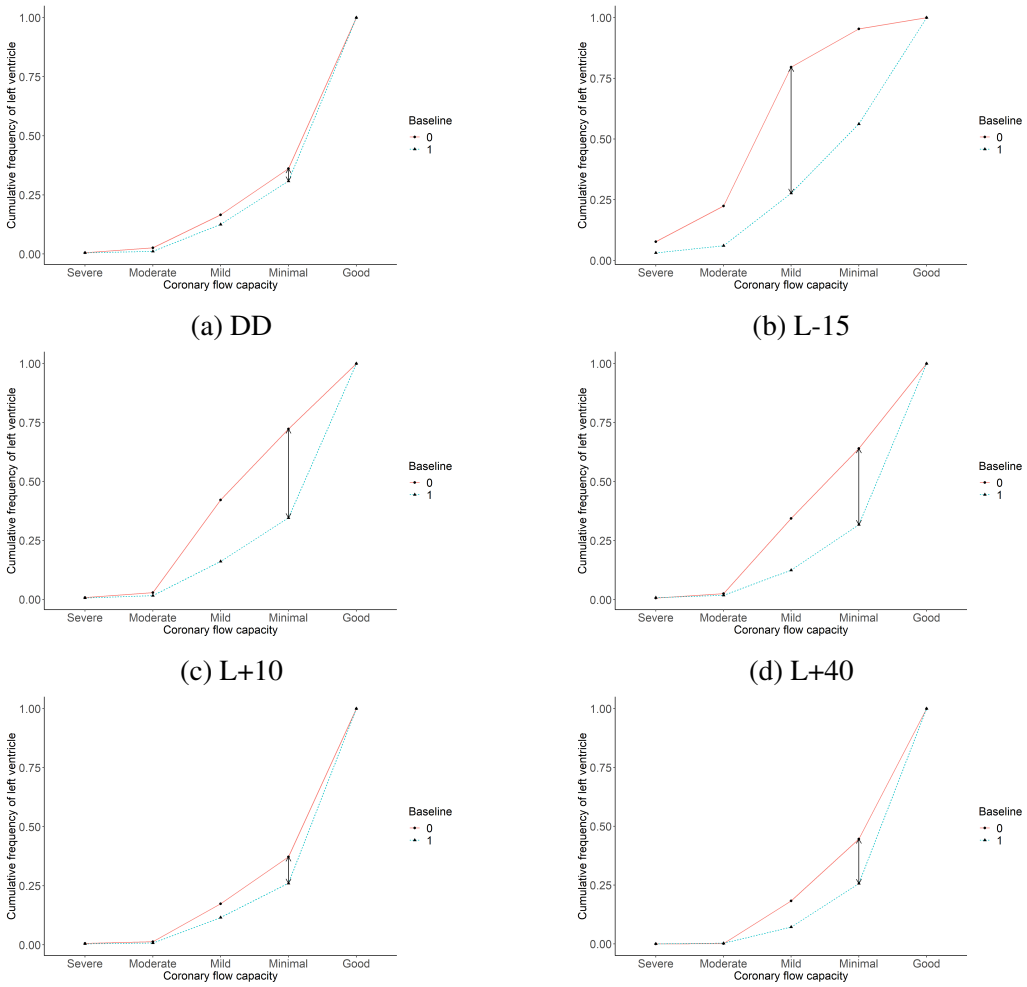
(e) L+55

(f) L+80

Figure 3.23: CFC frequency plots of protocols

Figure 3.23 shows the averaged CFC frequency distribution for each protocol. From the sub-plot 3.23a, we may see that the average CFC distribution for subjects in DD protocol was almost comparable. Therefore, we may conclude that if there were differences between baseline(dipyridamole) and non-baseline(regadenoson with different timing), the differences were due to the medication/timing difference as the trial controlled other effects pretty well. Major discrepancy was noticed between dipyridamole and regadenoson in L-15 protocol in sub-plot 3.23b. The frequency plot showed that subjects administered with regadenoson and Rb-82 activated 15s prior to the drug administration in the baseline had a much higher frequency

of mild/minimal reduced flow but a much lower frequency of good CFC compared to subjects administered with dipyridamole. Similar trends were also presented in L+10 protocol and L+40 protocol. Protocols with a suitable delay, 55s, to activate Rb-82 after regadenoson was administered had the average pixel distribution of CFC comparable to its baseline of dipyridamole. While a relatively lower frequency of pixels of good CFC was found in subjects with Rb-82 activated 80s after regadenoson bolus compared to their CFC using dipyridamole.



(a) DD

(b) L-15

(c) L+10

(d) L+40

(e) L+55

(f) L+80

Figure 3.24: Cumulative Averaged CFC Pixel Frequencies

Protocol	KS statistic	P-Values		
		Spatial Adjusted KS	Original KS	ICC adjusted KS
DD	0.05	0.96	0.047^{*}	0.29
L - 15	0.52	$< 10^{-16***}$	$< 10^{-16***}$	$< 10^{-16***}$
L + 10	0.38	$< 10^{-10***}$	$< 10^{-16***}$	$< 10^{-16***}$
L + 40	0.32	$< 10^{-7***}$	$< 10^{-16***}$	$< 10^{-16***}$
L + 55	0.11	0.24	$< 10^{-16***}$	0.0004^{***}
L + 80	0.19	0.004^{**}	$< 10^{-16***}$	$< e - 10^{***}$

* p-value < 0.05

** p-value < 0.005

*** p-value < 0.0005

Table 3.18: Kolmogorov-Smirnov Tests for Averaged Pixel Distribution of CFC

From the results of figure 3.24 and table 3.18 we may see that the original KS test without any adjustment tends to give smaller p-values. Liberal p-values lead to the overestimation of the significance of the test result. Hence, from the original KS test, before any adjustment, we may untruely conclude that the all protocols, including the repeated dipyridamole group, reported a statistically significant difference in CFC distribution between subjects baseline, administered dipyridamole, and test stage, either regadenoson or repeated dipyridamole.

With adjustment on the informative sample size, both the ICC adjusted KS test and the spatially adjusted KS test were able to report a higher p-value. It is worth noticing that the p-value from ICC adjusted KS was relatively lower than that of spatially adjusted KS. The averaged pixel distribution of CFC of subjects in L+55 protocol showed no statistically significant difference, based on the p-value reported from spatially adjusted KS test, between stages with dipyridamole administration and that of regadenoson administration. However,

Formula	Mode of action	Administration	Dose	Duration of infusion	Terminal half-life	Time to peak	Duration of action	Elimination	Antidote
$C_{15}H_{18}N_8O_5H_2O$	Selective A_2A	IV bolus	400 ug	10-s bolus	33–108 min	33 s	2.3 min	Renal (57%)	Aminophylline

Table 3.19: Regadenoson Pharmacokinetic and Pharmacodynamic Properties in Human Volunteers

From pharmacokinetic and pharmacodynamic table 3.19 we may see that the peak time of regadenoson concentration in blood is 33s (Jaroudi & Iskandrian, 2009). The lack of time for the medication to be absorbed by the organ may have lead to insufficient stress perfusion in protocols of early Rb-82 generator activation.

The KS tests for the L+80 protocol showed significant differences ($p = 0.004$) between the averaged pixel distribution of CFC for subjects administered with dipyridamole and regadenoson. Resutls from CFC could be supported with the absolute differences in stress perfusion and CFR from table 3.17. Compared with their baseline characteristics, the ordered protocols of absolute difference of stress perfusion are L - 15 $>$ L + 10 $>$ L + 40 $>$ L + 80 $>$ L + 55 $>$ DD.

Discussion and Concluding Remarks

The original KS overestimated the significance scale and produced a p-value that was too small. ICC adjustment in the KS test adjusts the p-values in the right direction. However, it is not as effective as the KS test with spatial adjustment. Spatial adjusted KS is able to adjust for the effect of autocorrelation in spatial settings and therefore produced a p-value closer to the true scale of significance. Regardless of the scale of the existing correlation, the original KS test did not adjust the sample size. The ICC adjusted KS test was able to shrink the sample size linearly while the spatially adjusted KS test was able to adjust the sample size exponentially. The KS statistics from original KS, ICC adjusted KS and spatially adjusted KS were the same. The differences in p-value are caused by the difference in informative sample size.

Our results partially agreed with results from mixed-effects ANOVA on stress flow (Johnson & Gould, 2015). The ANOVA results failed to detect the differences in the protocol of Rb-82 activated 80s after regadenoson bolus time. Analysis of averaged pixel distribution of CFC has proved to be more accurate than only considering CFR or absolute flow. Our analysis on the CFC provides an evaluation of the effectiveness of dipyridamole and different timing protocol of regadenoson. Even though the difference of averaged pixel distribution of CFC between dipyridamole and L+80 regadenoson is statistically significant, the clinical meaning of such difference needs more in-depth evaluation. Based on our findings, physicians may evaluate the cost-effect trade-off from each protocol and decide or inform patients with the findings so they could decide which protocol may be optimal in each case.

A bell shape hyperemia produced by different timing of regadenoson bolus time can be concluded from reported results of the trial. The stress perfusion increased as Rb-82 activation time delays, as the medication takes time to be distributed in blood and absorbed by organ. Then the stress perfusion decreased as the medication peak time and effectiveness time passed.

Our approach of analyzing PET scans may provide assistance in future image analysis as it is simple to apply and easy to understand. In our trial, the CFC is defined as a discontinuous variable determined by the value of CFR and stress flow. The KS test is a powerful tool in analyzing the pixel distribution. However, it may lack power and be conservative when the underlying pixel distribution was discrete (Conover, 1972a; Gleser, 1985). A two-sample spatially adjusted KS test for discontinuous distribution is desired. Meanwhile, the multi-dimensional KS tests were studied by researchers (Justel et al., 1997). Multi-dimensional KS test has been proved to be a sensitive and powerful tool in discriminating images.(Metchev & Grindlay, 2002) Therefore, in future studies, we may consider proposing a multi-dimensional KS test with adjustment for spatial autocorrelation based on such findings. Then a direct analysis could be carried on CFR and stress flow simultaneously.

This single-subject designed trial was imbalanced and therefore may have been vulnerable to insufficient power. The researchers did not blind any party in the trial. Therefore, there may be uncontrolled confounders that need to be addressed. In addition, subjects recruitment was carried out by convenience. There was no randomization in recruitment. Hence the conclusion from the trial may be potentially questionable in nature. In addition, the imbalanced trial design and the small sample sizes in L-15, L+40, and L+80 arm could potentially reduce the results reliability.

The spatial autocorrelation coefficient is one of the fundamental pillars of the spatially adjusted KS test. However, currently, there are no certain 'absolute' coefficients that account for spatial autocorrelation. By saying 'absolute' we mean that the spatial correlation coefficient was defined without any human-defining structure. Currently available coefficients were subjective in the sense that one has to define the spatial structure and the correlation scale regards to the spatial relationship between locations. For example, in this article, we assumed that the correlation between locations decay in proportion to the square of the distance. Another popular spatial correlation is the neighboring correlation, weight function w_{ij} equal to 1 if X_i and X_j is adjacent and equal to 0 otherwise. A method that could evaluate the spatial correlation absolutely, without any subjective definition is needed.

From the results of spatially adjusted KS test, we found that the regadenoson protocol with Rb-82 activated 55s after the injection of regadenoson has similar performance as dipyridamole. The protocols that activate Rb-82 15 seconds before, 10 seconds after, 40 seconds after or 80 seconds after regadenoson bolus time were sub-optimal compared to the hyperemia of dipyridamole.

References

Pettitt, A. N., & Stephens, M. A. [M. A.]. (1977). The kolmogorov-smirnov goodness-of-fit statistic with discrete and grouped data. *Technometrics*, *19*(2), 205–210.

Weiss, M. S. (1978). Modification of the kolmogorov-smirnov statistic for use with correlated data. *Journal of the American Statistical Association*, *73*(364), 872–875.

Lampariello, F. (2000). On the use of the kolmogorov-smirnov statistical test for immunofluorescence histogram comparison. *Cytometry*, *39*(3), 179–188.

Lim, D. H., & Jang, S. J. (2002). Comparison of two-sample tests for edge detection in noisy images. *Journal of the Royal Statistical Society. Series D (The Statistician)*, *51*(1), 21–30.

Geman, D., Geman, S., Graffigne, C., & Dong, P. (1990). Boundary detection by constrained optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(7), 609–628.

Chen, J. W., Sans, M. Q., Bogdanov, J. A., & Weissleder, R. (2006). Imaging of myeloperoxidase in mice by using novel amplifiable paramagnetic substrates1. *Radiology*, *240*(2), 473.

Baselice, F. [F.]. (2017). Ultrasound image despeckling based on statistical similarity. *Ultrasound in Medicine and Biology*, *43*(9), 2065–2078. Cited By :5.

Rajan, J., Dekker, A. J. D., & Sijbers, J. (2014). A new non-local maximum likelihood estimation method for rician noise reduction in magnetic resonance images using the kolmogorov-smirnov test. *Signal Processing*, *103*, 16–23. Cited By :21.

Kipritidis, J., Hofman, M. S., Siva, S., Callahan, J., Roux, P.-Y. L., Woodruff, H. C., … Keall, P. J. (2016). Estimating lung ventilation directly from 4d ct hounsfield unit values. *Medical Physics*, *43*(1), 33–43.

Brook, O. R., Gourtsoyianni, S., Brook, A., Mahadevan, A., Wilcox, C., & Raptopoulos, V. (2012). Spectral ct with metal artifacts reduction software for improvement of tumor visibility in the vicinity of gold fiducial markers. *Radiology*, *263*(3), 696–705.

Cressie, N. (1992). *Statistics for spatial data*.

Johnson, N. P., & Gould, K. L. [K. Lance]. (2015). Regadenoson versus dipyridamole hyperemia for cardiac pet imaging. *JACC: Cardiovascular Imaging*, *8*(4), 438–447.

Azhari, H., Beyar, R., & Sideman, S. (1999). On the human left ventricular shape. *Computers and Biomedical Research, an International Journal*, *32*(3), 264–282.

Harel, F., Finnerty, V., Authier, S., & Pelletier-Galarneau, M. (2018). Comparison of two dipyridamole infusion protocols for myocardial perfusion imaging in subjects with low likelihood of significant obstructive coronary artery disease. *Journal of nuclear cardiology : official publication of the American Society of Nuclear Cardiology*, 1–9.

Conover, W. J. (1972a). A kolmogorov goodness-of-fit test for discontinuous distributions. *Journal of the American Statistical Association*, *67*(339), 591–596.

Justel, A., Peña, D., & Zamar, R. (1997). A multivariate kolmogorov-smirnov test of goodness of fit. *Statistics & Probability Letters*, *35*(3), 251–259. ID: 271498.

Gould, K. L. [K. Lance], & Johnson, N. P. (2018). Coronary physiology beyond coronary flow reserve in microvascular angina. *Journal of the American College of Cardiology*, *72*(21), 2642–2662.

Zheng, W., & et al. (2019b). An adjustment of kolmogorov-smirnov test under spatial autocorrelation.

Kershah, S., Partovi, S., Traughber, B. J., Jr., R. F. M., Schluchter, M. D., O'Donnell, J. K., & Faulhaber, P. (2013). Comparison of standardized uptake values in normal structures between pet/ct and pet/mri in an oncology patient population. *Molecular Imaging and Biology*, *15*(6), 776–785.

Mitchell, G. F., Lamas, G. A., Vaughan, D. E., & Pfeffer, M. A. (1992b). Left ventricular remodeling in the year after first anterior myocardial infarction: A quantitative analysis of contractile segment lengths and ventricular shape. *Journal of the American College of Cardiology*, *19*(6), 1136–1144.

Hansen, F. K., Marinucci, D., Natoli, P., & Vittorio, N. (2002). Testing for non-gaussianity of the cosmic microwave background in harmonic space: An empirical process approach. *Physical Review D - Particles, Fields, Gravitation and Cosmology*, *66*(6). Cited By :14.

Jaroudi, W. A., & Iskandrian, A. E. (2009). Regadenoson: A new myocardial stress agent. *Journal of the American College of Cardiology*, *54*(13), 1123–1130.

Gleser, L. J. (1985). Exact power of goodness-of-fit tests of kolmogorov type for discontinuous distributions. *Journal of the American Statistical Association*, *80*(392), 954–958.

Metchev, S. A., & Grindlay, J. E. (2002). A two-dimensional kolmogorov–smirnov test for crowded field source detection: Rosat sources in ngc 6397. *Monthly Notices of the Royal Astronomical Society*, *335*(1), 73–83.

REFERENCES

Stephens, M. A. [M. A.]. (1992). Introduction to kolmogorov (1933) on the empirical determination of a distribution. (pp. 93–105). Breakthroughs in Statistics: Methodology and Distribution. ID: Stephens1992.

Kolmogorov, A. (1933). Sulla determinazione empirica di una lgge di distribuzione. *Inst.Ital.Attuari, Giorn. 4*, 83–91.

Smirnov, N. V. [N. V.]. (1939). On the Estimation of the Discrepancy Between Empirical Curves of Distribution for Two Independent Samples. *Bul. Math. de l'Univ. de Moscou*, *2*, 3–14.

von Mises, R. (1931). *Vorlesungen aus dem gebiete der angewandten mathematik*. F. Deuticke.

Smirnov, N. V. [N. V.]. (1937). Sur la distribution de w2 (criterium de m.r. von mises). *C. R. Acad. Sci. (Paris), 202*, 449–452.

Mises, R. V. (1928). *Wahrscheinlichkeit, statistik und wahrheit*. J. Springer.

Anderson, T. W. (1962). On the distribution of the two-sample cramer-von mises criterion. *The Annals of Mathematical Statistics*, *33*(3), 1148–1159.

Brown, B. M. (1982). Cramér—von mises distributions and permutation tests. *Biometrika*, *69*(3), 619–624.

Brown, B. M. (1994). Grouping corrections for circular goodness-of-fit tests. *Journal of the Royal Statistical Society. Series B (Methodological)*, *56*(1), 275–283.

Lockhart, R. A., Spinelli, J. J., & Stephens, M. A. (2007). Cramér–von mises statistics for discrete distributions with unknown parameters. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, *35*(1), 125–133.

Scholz, F. W., & Stephens, M. A. [M. A.]. (1987). K-sample anderson-darling tests. *Journal of the American Statistical Association*, *82*(399), 918.

Pettitt, A. N., & Stephens, M. A. [M. A.]. (1977). The kolmogorov-smirnov goodness-of-fit statistic with discrete and grouped data. *Technometrics*, *19*(2), 205–210.

Choulakian, V., Lockhart, R. A., & Stephens, M. A. (1994). Cramér-von mises statistics for discrete distributions. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, *22*(1), 125–137.

Arnold, T. B., & Emerson, J. W. (2011). Nonparametric goodness-of-fit tests for discrete null distributions. *R Journal*, *3*(2).

D'Agostino, R. B., & Stephens, M. A. [Michael A.]. (1986). *Goodness-of-fit techniques*. Marcel Dekker.

Berman, M. (1986). Testing for spatial association between a point process and another stochastic process. *Applied Statistics*, *35*(1), 54.

Clifford, P. [P.], Richardson, S., & Hémon, D. (1989). Assessing the significance of the correlation between two spatial processes. *Biometrics*, *45*(1), 123–134.

Demidenko, E. (2004). Kolmogorov-smirnov test for image comparison. In A. Laganá, M. L. Gavrilova, V. Kumar, Y. Mun, C. J. K. Tan, & O. Gervasi (Eds.), *Computational science and its applications – iccsa 2004* (pp. 933–939). Berlin, Heidelberg: Springer Berlin Heidelberg.

Weiss, M. S. (1978). Modification of the kolmogorov-smirnov statistic for use with correlated data. *Journal of the American Statistical Association*, *73*(364), 872–875.

Lampariello, F. (2000). On the use of the kolmogorov-smirnov statistical test for immunofluorescence histogram comparison. *Cytometry*, *39*(3), 179–188.

Lim, D. H., & Jang, S. J. (2002). Comparison of two-sample tests for edge detection in noisy images. *Journal of the Royal Statistical Society. Series D (The Statistician)*, *51*(1), 21–30.

Geman, D., Geman, S., Graffigne, C., & Dong, P. (1990). Boundary detection by constrained optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(7), 609–628.

Chen, J. W., Sans, M. Q., Bogdanov, J. A., & Weissleder, R. (2006). Imaging of myeloperoxidase in mice by using novel amplifiable paramagnetic substrates1. *Radiology*, *240*(2), 473.

Baselice, F. [F.]. (2017). Ultrasound image despeckling based on statistical similarity. *Ultrasound in Medicine and Biology*, *43*(9), 2065–2078. Cited By :5.

Rajan, J., Dekker, A. J. D., & Sijbers, J. (2014). A new non-local maximum likelihood estimation method for rician noise reduction in magnetic resonance images using the kolmogorov-smirnov test. *Signal Processing*, *103*, 16–23. Cited By :21.

Kipritidis, J., Hofman, M. S., Siva, S., Callahan, J., Roux, P.-Y. L., Woodruff, H. C., ... Keall, P. J. (2016). Estimating lung ventilation directly from 4d ct hounsfield unit values. *Medical Physics*, *43*(1), 33–43.

Brook, O. R., Gourtsoyianni, S., Brook, A., Mahadevan, A., Wilcox, C., & Raptopoulos, V. (2012). Spectral ct with metal artifacts reduction software for improvement of tumor visibility in the vicinity of gold fiducial markers. *Radiology*, *263*(3), 696–705.

Pearson, K. (1895). Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, *58*, 240–242.

Fisher, R. A. (1925). *Statistical methods for research workers* (I). ID: 1929-00958-000. Oxford, England: Stechert.

Fisher, R. A. (1934). *Statistical methods for research workers, 5th ed*. ID: 1934-15010-000. Oliver: Edinburgh.

Donner, A., & Koval, J. J. (1980). The estimation of intraclass correlation in the analysis of family data. *Biometrics*, *36*(1), 19–25.

Spearman, C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology*, *15*(1), 72–101. ID: 1926-00292-001.

Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, *30*(1/2), 81–93.

Cressie, N. (1992). *Statistics for spatial data*.

Moran, P. A. P. (1950). Notes on continuous stochastic phenomena. *Biometrika*, *37*(1/2), 17–23.

Anselin, L. (1995). Local indicators of spatial association—lisa. *Geographical Analysis*, *27*(2), 93–115.

Lai, D. (1997). Spatial statistical analysis of chinese cancer mortality. *Scandinavian Journal of Public Health*, *25*(4), 258–265.

Stewart, J., Manmathan, G., & Wilkinson, P. (2017). Primary prevention of cardiovascular disease: A review of contemporary guidance and literature.

technical report series, W. (2003). Diet, nutrition, and the prevention of chronic diseases. (Vol. 916). WHO technical report series, Geneva: World Health Organization.

Liang, S., Yang, F., Wen, T., Yao, Z., Huang, Q., & Ye, C. (2017). Nonlocal total variation based on symmetric kullback-leibler divergence for the ultrasound image despeckling. *BMC Medical Imaging*, *17*(1). Cited By :1.

Gulamhusein, S., Naccarelli, G. V., Ko, P. T., Prystowsky, E. N., Zipes, D. P., Barnett, H. J. M., ... Klein, G. J. (1982). Value and limitations of clinical electrophysiologic study in assessment of patients with unexplained syncope. *The American Journal of Medicine*, *73*(5), 700–705.

Gottdiener, J. S. (2003). Overview of stress echocardiography: Uses, advantages, and limitations. *Current Problems in Cardiology*, *28*(8), 485–516.

Swan, H. J., Ganz, W., Forrester, J., Marcus, H., Diamond, G., & Chonette, D. (1970). Catheterization of the heart in man with use of a flow-directed balloon-tipped catheter. *The New England Journal of Medicine*, *283*(9), 447–451.

Rich, M. W., & Crecelius, C. A. (1990). Incidence, risk factors, and clinical course of acute renal insufficiency after cardiac catheterization in patients 70 years of age or older: A prospective study. *Archives of Internal Medicine*, *150*(6), 1237–1242.

Kern, M. J., Lerman, A., Bech, J.-W., Bruyne, B. D., Eeckhout, E., Fearon, W. F., ... Spaan, J. A. E. (2006). Physiological assessment of coronary artery disease in the cardiac catheterization laboratory: A scientific statement from the american heart association committee on diagnostic and interventional cardiac catheterization, council on clinical cardiology. *Circulation*, *114*(12), 1321–1341.

Cosman, T. L., Arthur, H. M., & Natarajan, M. K. (2011). Prevalence of bruising at the vascular access site one week after elective cardiac catheterisation or percutaneous coronary intervention. *Journal of Clinical Nursing*, *20*(9-10), 1349–1356.

de Bruyne, B., Dorsaz, P. A., Doriot, P. A., Meier, B., Finci, L., & Rutishauser, W. (1988). Assessment of regional coronary flow reserve by digital angiography in patients with coronary artery disease. *International journal of cardiac imaging*, *3*(1), 47–55.

Lette, J., Tatum, J. L., Fraser, S., Miller, D. D., Waters, D. D., Heller, G., ... Nattel, S. (1995). Safety of dipyridamole testing in 73,806 patients: The multicenter dipyridamole safety study. *Journal of Nuclear Cardiology*, *2*(1), 3–17.

Dowsley, T., Al-Mallah, M., Ananthasubramaniam, K., Dwivedi, G., McArdle, B., & Chow, B. J. W. (2013). The role of noninvasive imaging in coronary artery disease detection, prognosis, and clinical decision making. *Canadian Journal of Cardiology*, *29*(3), 285–296.

Dahan, M., Viron, B. M., Poiseau, E., Kolta, A. M., Aubry, N., Paillole, C., ... Mignon, F. E. (2002). Combined dipyridamole-exercise stress echocardiography for detection of myocardial ischemia in hemodialysis patients: An alternative to stress nuclear imaging. *American Journal of Kidney Diseases*, *40*(4), 737–744.

Danad, I., Raijmakers, P. G., Driessen, R. S., Leipsic, J., Raju, R., Naoum, C., ... Knaapen, P. (2017). Comparison of coronary ct angiography, spect, pet, and hybrid imaging for diagnosis of ischemic heart disease determined by fractional flow reserve. *JAMA Cardiology*, *2*(10), 1100–1107.

Raff, G. L., Gallagher, M. J., O'Neill, W. W., & Goldstein, J. A. (2005). Diagnostic accuracy of noninvasive coronary angiography using 64-slice spiral computed tomography. *Journal of the American College of Cardiology*, *46*(3), 552–557.

Carli, M. F. D., Dorbala, S., Meserve, J., Fakhri, G. E., Sitek, A., & Moore, S. C. (2007). Clinical myocardial perfusion pet/ct. *The Journal of Nuclear Medicine*, *48*(5), 783–793.

Cremer, P., Hachamovitch, R., & Tamarappoo, B. (2014). Clinical decision making with myocardial perfusion imaging in patients with known or suspected coronary artery disease. *Seminars in Nuclear Medicine*, *44*(4), 320–329.

Picano, E. (1989). Dipyridamole-echocardiography test: Historical background and physiologic basis. *European Heart Journal*, *10*(4), 365–376.

Cerqueira, M. D., Verani, M. S., Schwaiger, M., Heo, J., & Iskandrian, A. S. (1994). Safety profile of adenosine stress perfusion imaging: Results from the adenoscan multicenter trial registry. *Journal of the American College of Cardiology*, *23*(2), 384–389.

Hendel, R. C., Bateman, T. M., Cerqueira, M. D., Iskandrian, A. E., Leppo, J. A., Blackburn, B., & Mahmarian, J. J. (2005). Initial clinical experience with regadenoson, a novel selective a2a agonist for pharmacologic stress single-photon emission computed tomography myocardial perfusion imaging. *Journal of the American College of Cardiology*, *46*(11), 2069.

Johnson, N. P., & Gould, K. L. [K. Lance]. (2015). Regadenoson versus dipyridamole hyperemia for cardiac pet imaging. *JACC: Cardiovascular Imaging*, *8*(4), 438–447.

Vasu, S., Bandettini, W. P., Hsu, L.-Y., Kellman, P., Leung, S., Mancini, C., … Arai, A. E. (2013). Regadenoson and adenosine are equivalent vasodilators and are superior than dipyridamole- a study of first pass quantitative perfusion cardiovascular magnetic resonance. *Journal of cardiovascular magnetic resonance : official journal of the Society for Cardiovascular Magnetic Resonance*, *15*(1), 85.

Pijls, N. H., & van Lokien X Nunen. (2015). Fractional flow reserve, maximum hyperemia, adenosine, and regadenoson. *Cardiovascular Revascularization Medicine*, *16*(5), 263–265.

Gibbs, C. R., & Lip, G. Y. H. (1998). Do we still need dipyridamole? *British Journal of Clinical Pharmacology*, *45*(4), 323–328.

Goudarzi, B., Fukushima, K., Bravo, P., Merrill, J., & Bengel, F. (2011). Comparison of the myocardial blood flow response to regadenoson and dipyridamole: A quantitative analysis in patients referred for clinical 82rb myocardial perfusion pet. *European Journal of Nuclear Medicine and Molecular Imaging*, *38*(10), 1908–1916.

Bravo, P. E., Pozios, I., & Abraham, T. P. (2012). Comparison and effectiveness of regadenoson versus dipyridamole on stress electrocardiographic changes during positron emission tomography evaluation of patients with hypertrophic cardiomyopathy. *American Journal of Cardiology, The*, *110*(7), 1033–1039.

Gould, K. L. [K. Lance], Lipscomb, K., & Hamilton, G. W. (1974). Physiologic basis for assessing critical coronary stenosis: Instantaneous flow response and regional distribution during coronary hyperemia as measures of coronary flow reserve. *The American Journal of Cardiology*, *33*(1), 87–94.

Klocke, F. J., & Lee, D. C. (2011). Absolute myocardial blood flow. *JACC: Cardiovascular Imaging*, *4*(9), 999–1001.

Wichmann, J. L., Meinel, F. G., Schoepf, U. J., Lo, G. G., Choe, Y. H., Wang, Y., … Cecco, C. N. D. (2015). Absolute versus relative myocardial blood flow by dynamic ct myocardial perfusion imaging in patients with anatomic coronary artery disease. *American Journal of Roentgenology*, *205*(1), W72.

Simpson, P. B. (1951). Note on the estimation of a bivariate distribution function. *The Annals of Mathematical Statistics*, *22*(3), 476–478.

Crutcher, H. L. (1975). A note on the possible misuse of the kolmogorov-smirnov test. *Journal of Applied Meteorology*, *14*(8), 1600–1603. doi: 10.1175/1520-0450(1975)0142.0.CO;2; 25.

Lilliefors, H. W. (1967). On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, *62*(318), 399–402.

Ding, A. A., & Li, Y. (2013). Copula correlation: An equitable dependence measure and extension of pearson's correlation. *arXiv preprint arXiv:1312.7214*.

Dithinde, M., Phoon, K. K., De, W. M., & Retief, J. V. (2011). Characterization of model uncertainty in the static pile design formula. *Journal of Geotechnical and Geoenvironmental Engineering*, *137*(1), 70–85. doi: 10.1061/(ASCE)GT.1943-5606.0000401; 17.

Genest, C., & Rivest, L.-P. (1993). Statistical inference procedures for bivariate archimedean copulas. *Journal of the American Statistical Association*, *88*(423), 1034–1043.

Joe, H. (1997). *Multivariate models and multivariate dependence concepts*. Chapman and Hall/CRC.

Mitchell, G. F., Lamas, G. A., Vaughan, D. E., & Pfeffer, M. A. (1992a). Left ventricular remodeling in the year after first anterior myocardial infarction: A quantitative analysis of contractile segment lengths and ventricular shape. ID: 271027.

Azhari, H., Beyar, R., & Sideman, S. (1999). On the human left ventricular shape. *Computers and Biomedical Research, an International Journal*, *32*(3), 264–282.

Harel, F., Finnerty, V., Authier, S., & Pelletier-Galarneau, M. (2018). Comparison of two dipyridamole infusion protocols for myocardial perfusion imaging in subjects with low likelihood of significant obstructive coronary artery disease. *Journal of nuclear cardiology : official publication of the American Society of Nuclear Cardiology*, 1–9.

Conover, W. J. (1972a). A kolmogorov goodness-of-fit test for discontinuous distributions. *Journal of the American Statistical Association*, *67*(339), 591–596.

Cribbie, R. A., & Keselman, H. J. (2003). The effects of nonnormality on parametric, nonparametric, and model comparison approaches to pairwise comparisons. *Educational and Psychological Measurement*, *63*(4), 615–635. doi: 10.1177/0013164403251283; 17.

Zheng, W., & et al. (2019a). A simulation study of a class of nonparametric test statistics: A close look of continuous, discrete and correlated variables.

Kitkungvan, D., Lai, D., Zhu, H., Roby, A. E., Johnson, N. P., Steptoe, D. D., ... Gould, K. L. (2017). Optimal adenosine stress for maximum stress perfusion, coronary flow reserve, and pixel distribution of coronary flow capacity by kolmogorov–smirnov analysis. *Circulation: Cardiovascular Imaging*, *10*(2), e005650.

Golub, G. H., & Loan, C. F. V. (2012). *Matrix computations*. JHU Press.

Justel, A., Peña, D., & Zamar, R. (1997). A multivariate kolmogorov-smirnov test of goodness of fit. *Statistics & Probability Letters*, *35*(3), 251–259. ID: 271498.

Fasano, G., & Franceschini, A. (1987). A multidimensional version of the kolmogorov-smirnov test. *Monthly Notices of the Royal Astronomical Society*, *225*, 155–170.

Peacock, J. A. (1983). Two-dimensional goodness-of-fit testing in astronomy. *Monthly Notices of the Royal Astronomical Society*, *202*, 615–627.

Gould, K. L. [K. Lance], & Johnson, N. P. (2018). Coronary physiology beyond coronary flow reserve in microvascular angina. *Journal of the American College of Cardiology*, *72*(21), 2642–2662.

Zheng, W., & et al. (2019b). An adjustment of kolmogorov-smirnov test under spatial autocorrelation.

Kershah, S., Partovi, S., Traughber, B. J., Jr., R. F. M., Schluchter, M. D., O'Donnell, J. K., & Faulhaber, P. (2013). Comparison of standardized uptake values in normal structures between pet/ct and pet/mri in an oncology patient population. *Molecular Imaging and Biology*, *15*(6), 776–785.

Mitchell, G. F., Lamas, G. A., Vaughan, D. E., & Pfeffer, M. A. (1992b). Left ventricular remodeling in the year after first anterior myocardial infarction: A quantitative analysis of contractile segment lengths and ventricular shape. *Journal of the American College of Cardiology*, *19*(6), 1136–1144.

Hansen, F. K., Marinucci, D., Natoli, P., & Vittorio, N. (2002). Testing for non-gaussianity of the cosmic microwave background in harmonic space: An empirical process approach. *Physical Review D - Particles, Fields, Gravitation and Cosmology*, *66*(6). Cited By :14.

Jaroudi, W. A., & Iskandrian, A. E. (2009). Regadenoson: A new myocardial stress agent. *Journal of the American College of Cardiology*, *54*(13), 1123–1130.

Gleser, L. J. (1985). Exact power of goodness-of-fit tests of kolmogorov type for discontinuous distributions. *Journal of the American Statistical Association*, *80*(392), 954–958.

Metchev, S. A., & Grindlay, J. E. (2002). A two-dimensional kolmogorov–smirnov test for crowded field source detection: Rosat sources in ngc 6397. *Monthly Notices of the Royal Astronomical Society*, *335*(1), 73–83.

Smirnov, N. V. [Nikolai V.]. (1939). On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bull.Math.Univ.Moscou*, *2*(2), 3–14.

Lilliefors, H. W. (1969). On the kolmogorov-smirnov test for the exponential distribution with mean unknown. *Journal of the American Statistical Association*, *64*(325), 387–389.

Cramér, H. (1928). On the composition of elementary errors. *Scandinavian Actuarial Journal*, *1928*(1), 141–180.

Conover, W. J. (1972b). A kolmogorov goodness-of-fit test for discontinuous distributions. *Journal of the American Statistical Association*, *67*(339), 591–596.

Anderson, T. W., & Darling, D. A. (1952). Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *The Annals of Mathematical Statistics*, *23*(2), 193–212.

S, M. (1986). Testing correlated "ecg-like" data for normality using a modified kolmogorov-sminov statistic. *IEEE Transactions on Biomedical Engineering*, *BME-33*(12), 1114–1120.

Spearman, C. (1987). The proof and measurement of association between two things. *The American Journal of Psychology*, *100*(3/4), 441–471.

Clifford, P. [Peter], Richardson, S., & Hémon, D. (1989). Assessing the significance of the correlation between two spatial processes. *Biometrics*, 123–134.

Pardo-Igúzquiza, E., & Dowd, P. A. (2004). Normality tests for spatially correlated data. *Mathematical Geology*, *36*(6), 659–682.

Sampson, P. D., & Guttorp, P. (1992). Nonparametric estimation of nonstationary spatial covariance structure. *Journal of the American Statistical Association*, *87*(417), 108.

Gnedenko, B. V. (1951). *Limit theorems for sums of independent random variables*. American Mathematical Society.

Fernández, V. A., Gamero, M. D. J., & García, J. M. (2008). A test for the two-sample problem based on empirical characteristic functions. *Comput. Stat. Data Anal. 52*(7), 3730–3748.

Li, C., & Shepherd, B. E. (2010). Test of association between two ordinal variables while adjusting for covariates. *Journal of the American Statistical Association*, *105*(490), 612–620.

Olea, R. A., & Pawlowsky-Glahn, V. (2009). Kolmogorov–smirnov test for spatially correlated data. *Stochastic Environmental Research and Risk Assessment*, *23*(6), 749–757.

DeKemp, R. A., Ruddy, T. D., Hewitt, T., Dalipaj, M. M., & Beanlands, R. S. B. (2000). Detection of serial changes in absolute myocardial perfusion with 82rb pet. *Journal of Nuclear Medicine*, *41*(8), 1426–1435.

Aston, J. A. D., Gunn, R. N., Worsley, K. J., Ma, Y., Evans, A. C., & Dagher, A. (2000). A statistical method for the analysis of positron emission tomography neuroreceptor ligand data. *NeuroImage*, *12*(3), 245–256.

Cressie, N. A. C. (2015). Spatial point patterns. (pp. 575–723). Statistics for Spatial Data. John Wiley Sons, Ltd.

Gould, K. L. [K L]. (2013). Coronary flow and physiology beyond the stenosis. *Heart and metabolism : management of the coronary patient*, *58*, 4–9.

Aguirre, G. K., Zarahn, E., & D'Esposito, M. (1998). A critique of the use of the kolmogorov-smirnov (ks) statistic for the analysis of bold fmri data. *Magnetic Resonance in Medicine*, *39*(3), 500–505.

Baselice, F. [Fabio], Ferraioli, G., Pascazio, V., & Sorriso, A. (2019). Denoising of mr images using kolmogorov-smirnov distance in a non local framework. *Magnetic Resonance Imaging*, *57*, 176–193.

Feher, A., Srivastava, A., Quail, M. A., Boutagy, N. E., Khanna, P., Wilson, L., ... Sinusas, A. J. (2018). Serial assessment of coronary flow reserve by rubidium-82 positron emission tomography predicts mortality in heart transplant recipients. *JACC. Cardiovascular imaging*.

Stolker, J. M., Lim, M., Shavelle, D. M., Morris, D. L., Angiolillo, D. J., Guzman, L. A., ... Neumayr, R. H. (2015). Pooled comparison of regadenoson versus adenosine for measuring fractional flow reserve and coronary flow in the catheterization laboratory. *Cardiovascular Revascularization Medicine*, *16*(5), 266–271.

Cassuto, L., & Jay, P. (2015). The phd dissertation. *Pedagogy: Critical Approaches to Teaching Literature, Language, Composition, and Culture*, *15*(1), 81–92.

Cho, S.-G., Lee, S. J., Na, M. H., Choi, Y. Y., & Bom, H. H.-S. (2018). Comparison of diagnostic accuracy of pet-derived myocardial blood flow parameters: A meta-analysis. *Journal of nuclear cardiology : official publication of the American Society of Nuclear Cardiology*, 1–12.

Driessen, R. S., Danad, I., Stuijfzand, W. J., Raijmakers, P. G., Schumacher, S. P., van Diemen, P. A., ... Knaapen, P. (2019). Comparison of coronary computed tomography angiography, fractional flow reserve, and perfusion imaging for ischemia diagnosis. *Journal of the American College of Cardiology*, *73*(2), 161–173.

Benjamin, E., Virani, S., Callaway, C., Chamberlain, A., Chang, A., Cheng, S., … Muntner, P. (2018). Heart disease and stroke statistics—2018 update: A report from the american heart association. *Circulation*, *137*(12), e492.

Appendices

A A Simulation Study of A Class of Nonparametric Test Statistics: A Close Look of Continuous, Discrete and Correlated Variables: R Codes

A.1 One-sample Simulation

```
##############################################################################################
##############################################################################################
################################### Author: Wenjun Zheng #####################################
################################### Date: 03-19-2018 #########################################
################################### Title: A simulation study of KS ##########################
##############################################################################################

# load necessary packages
if (!is.loaded("mpi_initialize")) {
  library("Rmpi")
}
library(snow)

# generate cluster in MPI type
ncs <- parallel::detectCores()
cl <- makeCluster(ncs - 1, type = "MPI")

# pass necessary packages to load in clusters
clusterEvalQ(cl, library(psych))
clusterEvalQ(cl, library(MASS))
clusterEvalQ(cl, library(cramer))
clusterEvalQ(cl, library(goftest))
clusterEvalQ(cl, library(EWGoF))
clusterEvalQ(cl, library(kSamples))
clusterEvalQ(cl, library(zoo))
clusterEvalQ(cl, library(dgof))
clusterEvalQ(cl, library(KSgeneral))
clusterEvalQ(cl, library(EnvStats))

##############################################################################################
################################### Preparing functions ######################################
########################################## One-Sample ########################################
##############################################################################################

################################### Type I error #############################################

Comp1err.1s <- function(itn = 1000, sh1 = 1,
                        sc1 = 0.5, size = 500, probm = c(0.1, 0.9), dist = 'Weibull'){
  options(warn=-1)
  test.results <- lapply(vector("list", itn), function(x) vector("list", 4))
  if (dist == "Weibull"){
    for (i in 1:itn){
      x1 <- rweibull(size, shape = sh1, scale = sc1)

      ks_1sam <- stats::ks.test(x1, 'pweibull', shape = sh1, scale = sc1)$p.value
      cvm_1sam <- goftest::cvm.test(x1, 'pweibull', shape = sh1, scale = sc1)$p.value
      ad_1sam <- goftest::ad.test(x1, 'pweibull', shape = sh1, scale = sc1)$p.value
      chisq_1sam <- EnvStats::gofTest(x1, test = "chisq", distribution = "weibull",
                                      param.list = list(shape = sh1, scale = sc1))$p.value
      # chisq_1sam <- chisq1s(x1, sh1, sc1, dist)

      test.results[[i]] <- list("One-sample Kolmogorov-Smirnov Test"=ks_1sam,
                                "One-sample Cramer-von Mises Test"=cvm_1sam,
                                "One-sample Anderson-Darling Test" = ad_1sam,
                                "One-sample Chi-Squared Test" = chisq_1sam)
    }
  }
  else if (dist == "Normal"){
    for (i in 1:itn){
      x1 <- rnorm(size, mean = sh1, sd = sc1)

      ks_1sam <- stats::ks.test(x1, 'pnorm', mean = sh1, sd = sc1)$p.value
      cvm_1sam <- goftest::cvm.test(x1, 'pnorm', mean = sh1, sd = sc1)$p.value
      ad_1sam <- goftest::ad.test(x1, 'pnorm', mean = sh1, sd = sc1)$p.value
      chisq_1sam <- EnvStats::gofTest(x1, test = "chisq", distribution = "norm",
```

```
                                    param.list = list(mean = sh1, sd = sc1))$p.value
      # chisq_1sam <- chisq1s(x1, sh1, sc1, dist)

      test.results[[i]] <- list("One-sample Kolmogorov-Smirnov Test"=ks_1sam,
                                "One-sample Cramer-von Mises Test"=cvm_1sam,
                                "One-sample Anderson-Darling Test" = ad_1sam,
                                "One-sample Chi-Squared Test" = chisq_1sam)
    }
  }else if (dist == "Multinomial"){
    for (i in 1:itn){
      x1 <- rmultinom(n=1, size, prob = probm)
      # categorize data
      x1_dt <- unlist(apply(as.data.frame(1:length(x1)), 1,
                            function(l){rep(l, x1[l])}))
      null_ecdf <- stepfun(1:length(x1), cumsum(c(0, probm)))

      ks_1sam <- dgof::ks.test(x1_dt, null_ecdf, simulate.p.value = T)$p.value
      cvm_1sam <- dgof::cvm.test(x1_dt, null_ecdf, type = "W2")$p.value
      ad_1sam <- dgof::cvm.test(x1_dt, null_ecdf, type = "A2")$p.value
      chisq_1sam <- chisq.test(x1, p = probm, rescale.p = T)$p.value

      test.results[[i]] <- list("One-sample Kolmogorov-Smirnov Test"=ks_1sam,
                                "One-sample Cramer-von Mises Test"=cvm_1sam,
                                "One-sample Anderson-Darling Test" = ad_1sam,
                                "One-sample Chi-Squared Test" = chisq_1sam)
    }
  }
  err_list <- lapply(test.results, function(c) c < 0.05)
  ks_err <- mean(sapply(err_list, function(l) l[[1]]))
  cvm_err <- mean(sapply(err_list, function(l) l[[2]]))
  ad_err <- mean(sapply(err_list, function(l) l[[3]]))
  chisq_err <- mean(sapply(err_list, function(l) l[[4]]))# l[[4]]))
  type1err <- list("Type I error of Kolmogorov-Smirnov Test"=ks_err,
                   "Type I error of Cramer-von Mises Test"=cvm_err,
                   "Type I error of Anderson-Darling Test" = ad_err,
                  "Type I error of Chi-Squared Test" = chisq_err)
  options(warn=0)
  if(dist == "Multinomial"){
    outlist<-c(probm)}else{ outlist<- c(sh1, sc1)}
  return(list('Parameters' = outlist,
              'size' = size, 'Iteration times' = itn, 'distribution' = dist,
              'Type I error'= type1err, 'P-value List' = test.results))
}

################################### Power Calculation #####################################
ComPower.1s <- function(itn = 1000, sh1 = 1, sc1 = 0.5,
                        sh2 = 1, sc2 = 0.5, probm = c(0.1, 0.9),
                        probm2 = c(0.1, 0.9), size = 500, dist = 'Weibull'){
  options(warn=-1)
  test.results <- lapply(vector("list", itn), function(x) vector("list", 4))
  if (dist == "Weibull"){
    for (i in 1:itn){
      x1 <- rweibull(size, shape = sh1, scale = sc1)

      ks_1sam <- stats::ks.test(x1, 'pweibull', shape = sh2, scale = sc2)$p.value
      cvm_1sam <- goftest::cvm.test(x1, 'pweibull', shape = sh2, scale = sc2)$p.value
      ad_1sam <- goftest::ad.test(x1, 'pweibull', shape = sh2, scale = sc2)$p.value
      chisq_1sam <- EnvStats::gofTest(x1, test = "chisq", distribution = "weibull",
                                      param.list = list(shape = sh2, scale = sc2))$p.value
      # chisq_1sam <- chisq1s(x1, sh2, sc2, dist)

      test.results[[i]] <- list("One-sample Kolmogorov-Smirnov Test"=ks_1sam,
                                "One-sample Cramer-von Mises Test"=cvm_1sam,
                                "One-sample Anderson-Darling Test" = ad_1sam,
                                "One-sample Chi-Squared Test" = chisq_1sam)
    }
  }
  else if (dist == "Normal"){
    for (i in 1:itn){
      x1 <- rnorm(size, mean = sh1, sd = sc1)

      ks_1sam <- stats::ks.test(x1, 'pnorm', mean = sh2, sd = sc2)$p.value
      cvm_1sam <- goftest::cvm.test(x1, 'pnorm', mean = sh2, sd = sc2)$p.value
      ad_1sam <- goftest::ad.test(x1, 'pnorm', mean = sh2, sd = sc2)$p.value
      chisq_1sam <- EnvStats::gofTest(x1, test = "chisq", distribution = "norm",
                                      param.list = list(mean = sh2, sd = sc2))$p.value
      # chisq_1sam <- chisq1s(x1, sh2, sc2, dist)

      test.results[[i]] <- list("One-sample Kolmogorov-Smirnov Test"=ks_1sam,
                                "One-sample Cramer-von Mises Test"=cvm_1sam,
                                "One-sample Anderson-Darling Test" = ad_1sam,
                                "One-sample Chi-Squared Test" = chisq_1sam)
    }
  }
  else if (dist == "Multinomial"){
    for (i in 1:itn){
      x1 <- rmultinom(n=1, size, prob = probm)
      # categorize data
```

```
      x1_dt <- unlist(apply(as.data.frame(1:length(x1)), 1,
                            function(l){rep(l, x1[l])}))
      null_ecdf <- ecdf(unlist(apply(as.data.frame(1:length(probm2)), 1,
                                     function(l){rep(l, probm2[l]*100)})))

      ks_1sam <- dgof::ks.test(x1_dt, null_ecdf, simulate.p.value = T)$p.value
      cvm_1sam <- dgof::cvm.test(x1_dt, null_ecdf, type = "W2")$p.value
      ad_1sam <- dgof::cvm.test(x1_dt, null_ecdf, type = "A2")$p.value
      chisq_1sam <- chisq.test(x1, p = probm2, rescale.p = T)$p.value

      test.results[[i]] <- list("One-sample Kolmogorov-Smirnov Test"=ks_1sam,
                                "One-sample Cramer-von Mises Test"=cvm_1sam,
                                "One-sample Anderson-Darling Test" = ad_1sam,
                                "One-sample Chi-Squared Test" = chisq_1sam)
    }
  }
  power_list <- lapply(test.results, function(c) c < 0.05)
  ks_power <- mean(sapply(power_list, function(l) l[[1]]))
  cvm_power <- mean(sapply(power_list, function(l) l[[2]]))
  ad_power <- mean(sapply(power_list, function(l) l[[3]]))
  chisq_power <- mean(sapply(power_list, function(l) l[[4]]))
  powerlist <- list("Power of Kolmogorov-Smirnov Test"=ks_power,
                    "Power of Cramer-von Mises Test"=cvm_power,
                    "Power of Anderson-Darling Test" = ad_power,
                    "Power of Chi-Squared Test" = chisq_power)
  options(warn=0)
  if(dist == "Multinomial"){
    outlist<-c('null'=probm, 'alternative'=probm2)}else{ outlist<- c(sh1, sc1)}
  return(list('Parameters' = outlist,
              'size' = size, 'Iteration times' = itn, 'distribution' = dist,
              'MC power' = powerlist, 'P-value List' = test.results))
}

# pass function to clusters
clusterExport(cl, list('Complerr.1s'))
clusterExport(cl, list('ComPower.1s'))

# example: sample weibull distributed observations
# x <- rweibull(100, shape = 1, scale = 1)
# shape = (0.5, 1, 2, 3, 5), scale = (1, 2, 3)
# delta teps: "shape: 0.1-1 by 0.1 ;scale:0.1-0.5 by 0.1"
# generate correlated variables first
# use Gaussian copula, due to the property of copula, it may change correlation
set.seed(831111)
# shape and scale parameters
shape_para <- c(0.5, 1, 2, 3, 5)
scale_para <- c(1, 2, 3)
# generate unique combinations for shape and scale
para_list <- t(expand.grid(shape_para, scale_para))


# delta , 5 levels of change in original parameter to see the power
para_dlt <- c(0.05, 0.1, 0.2, 0.5, 1)
# generate unique list for delta
weibull_dlt_list <- t(expand.grid(para_dlt, shape_para, scale_para))
weibull_dlt_list <- rbind(weibull_dlt_list, weibull_dlt_list[1,]*weibull_dlt_list[2,],
                          weibull_dlt_list[1,]*weibull_dlt_list[3,] )
weibull_dlt_list[4,] <- weibull_dlt_list[2,] + weibull_dlt_list[4,]
weibull_dlt_list[5,] <- weibull_dlt_list[3,] + weibull_dlt_list[5,]
rownames(weibull_dlt_list) <- c('dlt', 'nul_shape', 'nul_scale', 'al_shape', 'al_scale')
weibull_dlt_list <- weibull_dlt_list[-1,]

# for normal distribution
mu_para <- c(0, 1, 3, 5)
sigma_para <- c(0.1, 0.5, 2)
norm_para_list <- t(expand.grid(mu_para, sigma_para))


norm_dlt_list <- t(expand.grid(para_dlt, mu_para, sigma_para))
norm_dlt_list <- rbind(norm_dlt_list, norm_dlt_list[1,]*norm_dlt_list[2,],
                       norm_dlt_list[1,]*norm_dlt_list[3,] )
norm_dlt_list[4,] <- norm_dlt_list[2,] + norm_dlt_list[4,]
norm_dlt_list[5,] <- norm_dlt_list[3,] + norm_dlt_list[5,]
rownames(norm_dlt_list) <- c('dlt', 'nul_mu', 'nul_sd', 'al_mu', 'al_sd')
norm_dlt_list <- norm_dlt_list[-1,]
for (i in 1:3){
  norm_dlt_list[3, ((i-1)*20+1):((i-1)*20+5)] <- norm_dlt_list[3, ((i-1)*20+1):((i-1)*20+5)] + c(0.01, 0.02, 0.03, 0.04,
      0.05)
}


# MC iteration times
tot_itn <- 10000
# calculate iterations needed for each computing core
it_n <- round(tot_itn/(ncs-1))

# sample size
size_n <- c(10, 20, 30, 100, 500)
```

```
# try different corr coef to make sure we have a good simulation sample
# rho <- c(-0.8, -0.5, -0.2, -0.1, 0.1, 0.2, 0.5, 0.8)

# a function for simulation, note itn is the simulation numbers, sh is shape parameter
# sc is the scale parameter, sig is the correlation matrix, make sure it's 2*2 if two sample
# Two-sample simulation, weibull

# set cluster random number generator to each nodes.
clusterSetupRNG(cl)


err1_norm <- lapply(1:ncol(norm_para_list), function(l) {
  lapply(1:ncs, function(l) {
  list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})})
err1_norm_list <- lapply(1:5, function(j){
  lapply(1:ncol(norm_para_list), function(l) {
    list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
})
##################11 cores, parallel###############
# parallel version is 6 times faster than the usual one
start_t <- Sys.time()
for (q in 1:5){
  err1_norm <- apply(norm_para_list, 2, function(l) {
    clusterCall(cl, Complerr.1s, itn = it_n, sh1 = l[1],
                sc1 = l[2], dist = "Normal", size = size_n[q])
  })
  err1_norm_list[[q]] <- err1_norm
}
end_t <- Sys.time()
jobtime <- end_t - start_t
jobtime

# clusterExport(cl, "it_n")
# clusterExport(cl, "para_list")
# clusterExport(cl, "size_n")
#
# start_t <- Sys.time()
#
# err1_weibull_list <- parRapply(cl, para_list, function(l){
#   err1_weibull<- Complerr.1s(itn = it_n,
#                sh1 = l[1], sc1 = l[2], size = size_n[q])
#   return(err1_weibull)
# } )
save(err1_norm_list, file = 'T1E_Norm.RData')

#######################perform power study###############
pow1_norm <- lapply(1:ncol(norm_para_list), function(l) {
  list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
pow1_norm_list <- lapply(1:5, function(l) {
  list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
start_t <- Sys.time()
for (q in 1:5){
  pow1_norm <- apply(norm_dlt_list, 2, function(l) {
    clusterCall(cl, ComPower.1s, itn = it_n, sh1 = l[1],
                sc1 = l[2], sh2 = l[3], sc2 = l[2], dist = "Normal", size = size_n[q])
  })
  pow1_norm_list[[q]] <- pow1_norm
}
end_t <- Sys.time()
jobtime <- end_t - start_t
jobtime

save(pow1_norm_list, file = 'POW_norm_Nulvar.RData')

# null: nul_shape nul_scale, alternative: nul_shape, alt_scale
pow1_norm <- lapply(1:ncol(norm_para_list), function(l) {
  list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
pow1_norm_list <- lapply(1:5, function(l) {
  list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
start_t <- Sys.time()
for (q in 1:5){
  pow1_norm <- apply(norm_dlt_list, 2, function(l) {
    clusterCall(cl, ComPower.1s, itn = it_n, sh1 = l[1],
                sc1 = l[2], sh2 = l[1], sc2 = l[4], dist = "Normal", size = size_n[q])
  })
  pow1_norm_list[[q]] <- pow1_norm
}
end_t <- Sys.time()
jobtime <- end_t - start_t
jobtime

save(pow1_norm_list, file = 'POW_norm_Nulmu.RData')

# null: nul_shape nul_scale, alternative: alt_shape, alt_scale
pow1_norm <- lapply(1:ncol(norm_para_list), function(l) {
  list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
pow1_norm_list <- lapply(1:5, function(l) {
  list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
```

```
start_t <- Sys.time()
for (q in 1:5){
  pow1_norm <- apply(norm_dlt_list, 2, function(l) {
    clusterCall(cl, ComPower.1s, itn = it_n, sh1 = l[1],
                sc1 = l[2], sh2 = l[3], sc2 = l[4], dist = "Normal", size = size_n[q])
  })
  pow1_norm_list[[q]] <- pow1_norm
}
end_t <- Sys.time()
jobtime <- end_t - start_t
jobtime

save(pow1_norm_list, file = 'POW_norm_alt.RData')

# close cluster
stopCluster(cl)

# Tell all slaves to close down, and exit the program
mpi.quit()
```

A.2 Two-sample Simulation

```
#!/usr/bin/env Rscript
getwd()

if (!is.loaded("mpi_initialize")) {
  library("Rmpi")
}
library(snow)

# generate cluster in MPI type
ncs <- parallel::detectCores()
avilable_mpi_ncs <- ncs -1
cl <- makeCluster(avilable_mpi_ncs, type = "MPI")

# pass necessary packages to load in clusters
clusterEvalQ(cl, library(psych))
clusterEvalQ(cl, library(MASS))
clusterEvalQ(cl, library(cramer))
clusterEvalQ(cl, library(goftest))
clusterEvalQ(cl, library(EWGoF))
clusterEvalQ(cl, library(kSamples))
clusterEvalQ(cl, library(zoo))
clusterEvalQ(cl, library(dgof))
clusterEvalQ(cl, library(KSgeneral))
clusterEvalQ(cl, library(EnvStats))
clusterEvalQ(cl, library(dplyr))

##############################################################################################
################################### Preparing functions ######################################
########################################## Two-Sample ########################################
##############################################################################################

# binning mechanism is actually very scientific
# The small-n(N<35) part is a rule of thumb that says you should have on average
# at least five data points per bin (a rule which is not always followed in practice).
# The large-n part(n>=35) has a real basis in statistical theory.  A reference for it is in
# Goodness-of-Fit Tests by Ralph D'Agostino and Michael Stephens (Dekker 1986), page 70.

chisq2s <- function(x1, x2, dists = 'Weibull'){
  if(is.null(x1)|is.null(x2)){
    stop("Insert a valid test data.")
  }
  if(min(length(x1), length(x2)) < 35){
    n_bin <- round(length(x1)/5, 0)
  }else{
    n_bin <- floor(1.88*(min(length(x1), length(x2))^(2/5)))
  }
  # set the binning range
  range_para <- ifelse(dists == "Normal", 1.1, 0.9)
  while(n_bin>2 ){
    brks <- seq(min(x1, x2)-.01,max(x1, x2)+.01, length.out = n_bin)
    p1 <- hist(x1, breaks=brks, right=FALSE, plot = F)
    p2 <- hist(x2, breaks=brks, right=FALSE, plot = F)
    if (sum(p2$counts < 5) ==0){
      break
    }
    n_bin = n_bin-1
  }
  if (n_bin==2){
    brks <- seq(min(x1, x2)-.01,max(x1, x2)+.01, length.out = n_bin+1)
    p1 <- hist(x1, breaks=brks, right=FALSE, plot = F)
    p2 <- hist(x2, breaks=brks, right=FALSE, plot = F)
```

```
  }
  # calculate expected pr for each bins
  return(chisq.test(cbind(p1$counts, p2$counts))$p.value)
}

Asym.Cvm.2s <- function(x1, x2, alpha = 0.05){
  if(is.null(x1)|is.null(x2)){
    stop("Insert a valid test data.")
  }

  m <- length(x1)
  n <- length(x2)
  N <- m + n

  rank_xy <- rank(c(x1,x2), ties.method = "min")
  rank_x <- sort(rank_xy[1:m])
  rank_y <- sort(rank_xy[-(1:m)])

  component_xy <- (4*m*n-1)/(6*N)

  component_xx <- (1/(N*n))*sum(sapply(1:m, function(l) {
    (l-rank_x[l])^2
  }))
  component_yy <- (1/(m*N))*sum(sapply(1:n, function(l) {
    (l-rank_y[l])^2
  }))

  t_stat <- -(component_xy-component_xx-component_yy)
  exp_t_stat <- 1/6 + 1/(6*N)
  var_t_stat <- ((N + 1)/(180*(N^2)))*(4*(N-1)-(3*(N^2))/(m*n))
  z_stat <- (t_stat-exp_t_stat)/sqrt(var_t_stat) + 1/6
  ################## need to compute the significance value ###################################
  # in paper(Curry, Dang, 2018) it suggest using d = 4 or 10, we try 4 here.
  t_sig <- (sqrt(45)/(pi^2))*sum(sapply(c(1:2), function(k){
    (1/(k^2))*(qchisq(1-0.05, df=1)-1)
  }))
  if (z_stat > t_sig){
    test_result <- 0.04
  }else{
    test_result <- 0.06
  }
  return(list('Ranked-CvM statistic' = z_stat, "Significance value" = t_sig,
              'Significance'=test_result))
}

disc_cvm <- function(x1, y1, alpha = 0.05){
  n_x <- length(x1)
  n_y <- length(y1)
  N <-  n_x + n_y
  N_xy <- c(n_x, n_y)
  # pooled x and y
  obs_xy <- as.data.frame(sort(c(x1, y1)))
  colnames(obs_xy) <- 'obs'
  # compute L distinct ordered observations, l_j  = f_ct[,2]
  f_ct <- dplyr::add_count(obs_xy, obs) %>% distinct(obs, n)
  distinct_f <- f_ct[,1]
  # compute f_1j
  f1_ct_temp <- dplyr::add_count(as.data.frame(x1), x1) %>% distinct(x1, n)
  colnames(f1_ct_temp)[1] <- 'obs'
  f1_ct <- merge(f_ct[,1], f1_ct_temp, all.x = T)
  f1_ct[is.na(f1_ct)] <- 0
  # compute f_2j
  f2_ct_temp <- dplyr::add_count(as.data.frame(y1), y1) %>% distinct(y1, n)
  colnames(f2_ct_temp)[1] <- 'obs'
  f2_ct <- merge(f_ct[,1], f2_ct_temp, all.x = T)
  f2_ct[is.na(f2_ct)] <- 0
  # pool f_1 and f_2
  f_ij <- rbind(t(f1_ct[,2]), t(f2_ct[,2]))
  # compute L
  c_l <- nrow(f_ct)
  # compute l = sum(f_ij)
  l <- t(f_ct[,2])
  # compute M_aij
  M_a1j <- sapply(1:c_l, function(l) sum(f_ij[1,1:l]))
  M_a2j <- sapply(1:c_l, function(l) sum(f_ij[2,1:l]))
  M_aij <- rbind(M_a1j, M_a2j)
  # compute T_ij
  T_ij <- as.matrix(N_xy, nrow = 2) %*% t(as.matrix(sapply(1:c_l, function(l) sum(f_ct[1:l,2]))/N))
  # Compute statistic
  p_j <- unlist(f_ct[,2]/N)
  W_k <- sum(sapply(1:2, function(i){
    (1/N_xy[i])*sum(sapply(1:c_l, function(j){
      ((M_aij[i,j] - T_ij[i,j])^2)*p_j[j]
    }))
  }))
  # to standarize the statistic we need to calculate mu and var, capital p(P), capital d(D), capital q(Q)
  c_p <- matrix(0, nrow = c_l, ncol = c_l)
  c_p[lower.tri(c_p, diag = T)] <- 1
```

```
  c_d <- diag(p_j)
  c_q <- c_p%*%(c_d - as.matrix(p_j)%*%t(as.matrix(p_j)))%*%t(c_p)
  mu_T <- psych::tr(c_q)
  var_T <- psych::tr(c_q^2)
  # standardize T
  T_w <- (W_k - mu_T)/sqrt(var_T)
  # calculate critical value
  # critical value given in table
  critical_list <- t(matrix(c(0.25, .1, .05, .025, .01,
                              .295, 1.252, 2.012, 2.791, 3.838),
                            nrow = 2, byrow = T))
  critical <- critical_list[critical_list[,1]==alpha/2, 2]
  #compare ad statistic with critical value
  rej <- (T_w >= critical)
  p_val <- ifelse(rej == T, 0, 1)
  results <- list('Statistic' = W_k, 'Rejection' = rej, 'P-value' = p_val)
  return(results)
}
# take x1, y1 in contigency table as well
disc_ad <- function(x1, y1, alpha = 0.05){
  n_x <- length(x1)
  n_y <- length(y1)
  N <-  n_x + n_y
  N_xy <- c(n_x, n_y)
  # compute the variance of statistic
  g_v <- sum(sapply(1:(N-2), function(l){
    sum(sapply((l+1):(N-1), function(k){
      1/((N-l)*k)
    }))
  }))
  # H: capital h,
  c_h_v <- do.call(sum, lapply(c(n_x, n_y), function(l) 1/l))
  # h
  h_v <- sum(sapply(1:(N-1), function(l) 1/l))
  # a, b, c, d parameters according to paper
  a_v <- (4*g_v-6) + (10-6*g_v)*c_h_v
  b_v <- (2*g_v-4)*(2^2) + 8*h_v*2 + (2*g_v-14*h_v-4)*c_h_v - 8*h_v + 4*g_v - 6
  c_v <- (6*h_v+2*g_v -2)*(2^2) + (4*h_v - 4*g_v+6)*2 + (2*h_v-6)*c_h_v + 4*h_v
  d_v <- (2*h_v+6)*(2^2)-4*h_v*2
  # compute the variance
  var_n <- (a_v*(N^3) + b_v*(N^2)+c_v*N+d_v)/((N-1)*(N-2)*(N-3))

  # before compute statistic, first we define the variables for statistic
  obs_xy <- as.data.frame(sort(c(x1, y1)))
  colnames(obs_xy) <- 'obs'
  # compute L distinct ordered observations, l_j  = f_ct[,2]
  f_ct <- dplyr::add_count(obs_xy, obs) %>% distinct(obs, n)
  distinct_f <- f_ct[,1]
  # compute f_1j
  f1_ct_temp <- dplyr::add_count(as.data.frame(x1), x1) %>% distinct(x1, n)
  colnames(f1_ct_temp)[1] <- 'obs'
  f1_ct <- merge(f_ct[,1], f1_ct_temp, all.x = T)
  f1_ct[is.na(f1_ct)] <- 0
  # compute f_2j
  f2_ct_temp <- dplyr::add_count(as.data.frame(y1), y1) %>% distinct(y1, n)
  colnames(f2_ct_temp)[1] <- 'obs'
  f2_ct <- merge(f_ct[,1], f2_ct_temp, all.x = T)
  f2_ct[is.na(f2_ct)] <- 0
  # pool f_1 and f_2
  f_ij <- rbind(t(f1_ct[,2]), t(f2_ct[,2]))
  # compute L
  c_l <- nrow(f_ct)
  # compute l = sum(f_ij)
  l <- t(f_ct[,2])
  # compute M_aij
  M_a1j <- sapply(1:c_l, function(l) ifelse(l == 1, f_ij[1,l]/2, sum(f_ij[1,1:(l-1)], f_ij[1,l]/2)))
  M_a2j <- sapply(1:c_l, function(l) ifelse(l == 1, f_ij[2,l]/2, sum(f_ij[2,1:(l-1)], f_ij[2,l]/2)))
  M_aij <- rbind(M_a1j, M_a2j)
  # compute B_aj
  B_aj <- sapply(1:c_l, function(k) ifelse(k == 1, l[k]/2, sum(l[1:(k-1)], l[k]/2)))
  # compute statistic
  A_a2N <- ((N-1)/N)*sum(sapply(1:2, function(i) {
    (1/N_xy[i])*sum(sapply(1:c_l, function(j){
      (l[j]/N)*(((N*M_aij[i,j] - N_xy[i]*B_aj[j])^2)/((B_aj[j]*(N-B_aj[j]))-(N*l[j])/4))
    }))
  }))

  T_a2N <- (A_a2N - 1)/sqrt(var_n)
  # calculate critical value
  # actually it should gose to infinity, but I choose to go 3 as it should be enough
  # derive critical value
  critical_list <- t(matrix(c(0.25, .1, .05, .025, .01,
                              .326, 1.225, 1.96, 2.719, 3.752),
                            nrow = 2, byrow = T))
  critical <- critical_list[critical_list[,1]==alpha/2, 2]

  #compare cvm statistic with critical value
  rej <- (T_a2N >= critical)
```

```r
  p_val <- ifelse(rej == T, 0, 1)
  results <- list('Statistic' = A_a2N, 'Rejection' = rej, 'P-value' = p_val)
  return(results)
}
#################################################################################################
################################## wrapper for tests ############################################
#################################################################################################
Comp1err.2s <- function(itn = 1000, sh1 = 1,
                        sc1 = 0.5, size = 500, rho = 0,
                        probm = c(0.1, 0.9), dist = 'Weibull'){
  options(warn=-1)
  test.results <- lapply(vector("list", itn), function(x) vector("list", 4))
  if (dist == "Weibull"){

    test.results <- lapply(1:itn, function(q){
      if (rho == 0) {
        x1 <- rweibull(size, shape = sh1, scale = sc1)
        x2 <- rweibull(size, shape = sh1, scale = sc1)}
      else{
        covar <- matrix(c(1, rho, rho, 1), ncol=2)
        z <- MASS::mvrnorm(1000 ,mu=rep(0, 2),Sigma=covar,empirical=T)
        # get the inv-cdf of z
        u <- pnorm(z)
        # generate weibull distribution use gaussian copula
        x1 <- qweibull(u[,1], shape = sh1, scale = sc1)
        x2 <- qweibull(u[,2], shape = sh1, scale = sc1)
      }

      ks_2sam <- stats::ks.test(x1, x2)$p.value
      cvm_2sam <- Asym.Cvm.2s(x1, x2)[[3]]
      # cvm_2sam <- cramer::cramer.test(x1,x2)$p.value
      ad_2sam <- kSamples::ad.test(x1, x2, method = "asymptotic")$ad[1,3]
      chisq_2sam <- chisq2s(x1, x2, dist)

      test.results.temp <- list("Two-sample Kolmogorov-Smirnov Test"=ks_2sam,
                                "Two-sample Cramer-von Mises Test"=cvm_2sam,
                                "Two-sample Anderson-Darling Test" = ad_2sam,
                                "Two-sample Chi-Squared Test" = chisq_2sam)
    })

  }
  else if (dist == "Normal"){

    test.results <- lapply(1:itn, function(q){
      if (rho == 0) {
        x1 <- rnorm(size, mean = sh1, sd = sc1)
        x2 <- rnorm(size, mean = sh1, sd = sc1)}
      else{
        covar <- matrix(c(sc1*sc1, rho*sc1*sc1, rho*sc1*sc1, sc1*sc1), ncol=2)
        z <- MASS::mvrnorm(1000 ,mu=rep(sh1, 2),Sigma=covar,empirical=T)
        # generate weibull distribution use gaussian copula
        x1 <- z[,1]
        x2 <- z[,2]
      }

      ks_2sam <- stats::ks.test(x1, x2)$p.value
      cvm_2sam <- Asym.Cvm.2s(x1, x2)[[3]]
      # cvm_2sam <- cramer::cramer.test(x1,x2)$p.value
      ad_2sam <- kSamples::ad.test(x1, x2, method = "asymptotic")$ad[1,3]
      chisq_2sam <- chisq2s(x1, x2, dist)

      test.results.temp <- list("Two-sample Kolmogorov-Smirnov Test"=ks_2sam,
                                "Two-sample Cramer-von Mises Test"=cvm_2sam,
                                "Two-sample Anderson-Darling Test" = ad_2sam,
                                "Two-sample Chi-Squared Test" = chisq_2sam)
    })

  }else if (dist == "Multinomial"){

    test.results <- lapply(1:itn, function(q){
      x1 <- rmultinom(n=1, size, prob = probm)
      # categorize data
      x2 <- rmultinom(n=1, size, prob = probm)

      x1_dt <- unlist(apply(as.data.frame(1:length(x1)), 1, function(l){rep(l, x1[l])}))
      x2_dt <- unlist(apply(as.data.frame(1:length(x2)), 1, function(l){rep(l, x2[l])}))
      x2_ecdf <- stepfun(1:(length(x2)), cumsum(c(0, x2))/sum(x2))


      ks_2sam <- tryCatch({dgof::ks.test(x1_dt, x2_dt)$p.value},
                          error = function(e){ return(NA)} )
      cvm_2sam <- tryCatch({disc_cvm(x1_dt, x2_dt)[[3]]},
                           error = function(e){ return(NA)} )
      ad_2sam <- tryCatch({disc_ad(x1_dt, x2_dt)[[3]]},
                          error = function(e){ return(NA)} )
      chisq_2sam <- chisq.test(as.table(cbind(x1, x2)))$p.value

      test.results.temp <- list("Two-sample Kolmogorov-Smirnov Test"=ks_2sam,
```

```
                                "Two-sample Cramer-von Mises Test"=cvm_2sam,
                                "Two-sample Anderson-Darling Test" = ad_2sam,
                                "Two-sample Chi-Squared Test" = chisq_2sam)
    })

  }
  err_list <- lapply(test.results, function(c) c < 0.05)
  ks_err <- mean(sapply(err_list, function(l) l[[1]]), na.rm = T)
  cvm_err <- mean(sapply(err_list, function(l) l[[2]]), na.rm = T)
  ad_err <- mean(sapply(err_list, function(l) l[[3]]), na.rm = T)
  chisq_err <- mean(sapply(err_list, function(l) l[[4]]), na.rm = T)
  type1err <- list("Type I error of Kolmogorov-Smirnov Test"=ks_err,
                   "Type I error of Cramer-von Mises Test"=cvm_err,
                   "Type I error of Anderson-Darling Test" = ad_err,
                   "Type I error of Chi-Squared Test" = chisq_err)
  options(warn=0)
  if(dist == "Multinomial"){
    outlist<-c(probm)}else{ outlist<- c(sh1, sc1)}
  return(list('Parameters' = outlist,
              'size' = size, 'Iteration times' = itn, 'distribution' = dist,
              'Type I error'= type1err,
              'P-value List' = test.results))
}

################################### Power Calculation #######################################
ComPower.2s <- function(itn = 1000, sh1 = 1, sc1 = 0.5,
                        sh2 = 1, sc2 = 0.5, probm = c(0.1, 0.9),
                        probm2 = c(0.1, 0.9), size = 500, rho = 0, dist = 'Weibull'){
  options(warn=-1)
  test.results <- lapply(vector("list", itn), function(x) vector("list", 4))

  if (dist == "Weibull"){

    test.results <- lapply(1:itn, function(q){
      if (rho == 0) {
        x1 <- rweibull(size, shape = sh1, scale = sc1)
        x2 <- rweibull(size, shape = sh2, scale = sc2)}
      else{
        covar <- matrix(c(1, rho, rho, 1), ncol=2)
        z <- MASS::mvrnorm(1000 ,mu=rep(0, 2),Sigma=covar,empirical=T)
        # get the inv-cdf of z
        u <- pnorm(z)
        # generate weibull distribution use gaussian copula
        x1 <- qweibull(u[,1], shape = sh1, scale = sc1)
        x2 <- qweibull(u[,2], shape = sh2, scale = sc2)
      }

      ks_2sam <- stats::ks.test(x1, x2)$p.value
      cvm_2sam <- Asym.Cvm.2s(x1, x2)[[3]]
      # cvm_2sam <- cramer::cramer.test(x1,x2)$p.value
      ad_2sam <- kSamples::ad.test(x1, x2, method = "asymptotic")$ad[1,3]
      chisq_2sam <- chisq2s(x1, x2, dist)

      test.results.temp <- list("Two-sample Kolmogorov-Smirnov Test"=ks_2sam,
                                "Two-sample Cramer-von Mises Test"=cvm_2sam,
                                "Two-sample Anderson-Darling Test" = ad_2sam,
                                "Two-sample Chi-Squared Test" = chisq_2sam)
    })

  }
  else if (dist == "Normal"){

    test.results <- lapply(1:itn, function(q){
      if (rho == 0) {
        x1 <- rnorm(size, mean = sh1, sd = sc1)
        x2 <- rnorm(size, mean = sh2, sd = sc2)}
      else{
        covar <- matrix(c(sc1*sc1, rho*sc1*sc2, rho*sc1*sc2, sc2*sc2), ncol=2)
        z <- MASS::mvrnorm(1000 ,mu=rep(sh1, 2),Sigma=covar,empirical=T)
        # generate weibull distribution use gaussian copula
        x1 <- z[,1]
        x2 <- z[,2]
      }

      ks_2sam <- stats::ks.test(x1, x2)$p.value
      cvm_2sam <- Asym.Cvm.2s(x1, x2)[[3]]
      # cvm_2sam <- cramer::cramer.test(x1,x2)$p.value
      ad_2sam <- kSamples::ad.test(x1, x2, method = "asymptotic")$ad[1,3]
      chisq_2sam <- chisq2s(x1, x2, dist)

      test.results.temp <- list("Two-sample Kolmogorov-Smirnov Test"=ks_2sam,
                                "Two-sample Cramer-von Mises Test"=cvm_2sam,
                                "Two-sample Anderson-Darling Test" = ad_2sam,
                                "Two-sample Chi-Squared Test" = chisq_2sam)
    })

  }else if (dist == "Multinomial"){

```

```
    test.results <- lapply(1:itn, function(q){
      x1 <- rmultinom(n=1, size, prob = probm)
      # categorize data
      x2 <- rmultinom(n=1, size, prob = probm2)
      # generate categorize data
      x1_dt <- unlist(apply(as.data.frame(1:length(x1)), 1, function(l){rep(l, x1[l])}))
      x2_dt <- unlist(apply(as.data.frame(1:length(x2)), 1, function(l){rep(l, x2[l])}))
      x2_ecdf <- ecdf(unlist(apply(as.data.frame(1:length(x2)), 1, function(l){rep(l, x2[l])})))


      ks_2sam <- tryCatch({dgof::ks.test(x1_dt, x2_dt)$p.value},
                          error = function(e){ return(NA)} )
      cvm_2sam <- tryCatch({disc_cvm(x1_dt, x2_dt)[[3]]},
                           error = function(e){ return(NA)} )
      ad_2sam <- tryCatch({disc_ad(x1_dt, x2_dt)[[3]]},
                          error = function(e){ return(NA)} )
      chisq_2sam <- chisq.test(as.table(cbind(x1, x2)))$p.value

      test.results.temp <- list("Two-sample Kolmogorov-Smirnov Test"=ks_2sam,
                                "Two-sample Cramer-von Mises Test"=cvm_2sam,
                                "Two-sample Anderson-Darling Test" = ad_2sam,
                                "Two-sample Chi-Squared Test" = chisq_2sam)
    })

  }

  power_list <- lapply(test.results, function(c) c < 0.05)
  ks_power <- mean(sapply(power_list, function(l) l[[1]]))
  cvm_power <- mean(sapply(power_list, function(l) l[[2]]))
  ad_power <- mean(sapply(power_list, function(l) l[[3]]))
  chisq_power <- mean(sapply(power_list, function(l) l[[4]]))
  powerlist <- list("Power of Kolmogorov-Smirnov Test"=ks_power,
                    "Power of Cramer-von Mises Test"=cvm_power,
                    "Power of Anderson-Darling Test" = ad_power,
                    "Power of Chi-Squared Test" = chisq_power)
  options(warn=0)
  if(dist == "Multinomial"){
    outlist<-c('null'=probm, 'alternative'=probm2)}else{ outlist<- c(sh1, sc1)}
  return(list('Parameters' = outlist,
              'size' = size, 'Iteration times' = itn, 'distribution' = dist,
              'MC power' = powerlist, 'P-value List' = test.results))
}

# pass function to clusters
clusterExport(cl, list('chisq2s'))
clusterExport(cl, list('Asym.Cvm.2s'))
clusterExport(cl, list('disc_cvm'))
clusterExport(cl, list('disc_ad'))
clusterExport(cl, list('Comp1err.2s'))
clusterExport(cl, list('ComPower.2s'))

set.seed(831111)

# sample size
size_n <- c(10, 20, 30, 100, 500)

# decide the total number of iterations needed
tot_itn <- 10000
# calculate iterations needed for each computing core
it_n <- round(tot_itn/avilable_mpi_ncs)
# delta , 5 levels of change in original parameter to see the power
para_dlt <- c(0.05, 0.1, 0.2, 0.5, 1)
# generate unique list for delta
problist <- list(c(0.5, 0.5), c(0.1, 0.9), c(0.3, 0.7),
                 c(0.1, 0.2, 0.4, 0.2, 0.1), c(0.7, 0.2, 0.05, 0.03, 0.02),
                 c(0.3, 0.15, 0.1, 0.15, 0.3))
# probability in alternative
prob_dlt_list <- lapply(problist, function(k)
{apply(as.data.frame(para_dlt), 1, function(l) (return(list(k, round((k + l)/sum(k+l), 2))))})
##############################################################################################
######################################### Type I Error Analysis ##############################
##############################################################################################
err1_multn_2s <- lapply(1:length(problist), function(l) {
  list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
err1_multn_list_2s <- lapply(1:5, function(j){
  lapply(1:length(problist), function(l) {
    list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
})
start_t <- Sys.time()

for (q in 1:5){
  err1_multn_2s <- lapply(problist, function(l) {
    clusterCall(cl, Comp1err.2s, itn = it_n,
                probm = l, dist = "Multinomial", size = size_n[q])})
  err1_multn_list_2s[[q]] <- err1_multn_2s
}
# save(err1_weibull, file = 'T1E_Wei.RData')
save(err1_multn_list_2s, file = 'T1E_multn_size_2s.RData')
```

```
502
503 end_t <- Sys.time()
504 jobtime <-difftime(end_t, start_t, unit = "hours")
505 outline <- paste(end_t, ": T1E_multn_size_2s.RData"," is finished. Time difference is ", jobtime,sep="")
506 print(outline)
507 flush.console()
508 # rm(err1_multn_list_2s, err1_multn_2s)
509 #
510 # ##############################################################################################
511 # ##################################### Power study ############################################
512 # ##############################################################################################
513 # # Nul shape while alternative scale
514 pow_multn2s <- lapply(1:(length(prob_dlt_list)*5), function(l) {
515   list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
516 pow_multnlist_2s <- lapply(1:5, function(j){
517   lapply(1:(length(prob_dlt_list)*5), function(l) {
518     list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
519 })
520
521 start_t <- Sys.time()
522 for (q in 1:5){
523   pow_multn2s <- lapply(prob_dlt_list, function(l) {
524     lapply(1:5, function(k){
525       clusterCall(cl,ComPower.2s, itn = it_n,
526                   probm = l[[k]][[1]], probm2 = l[[k]][[2]], dist = "Multinomial", size = size_n[q])})
527     })
528   pow_multnlist_2s[[q]] <- pow_multn2s
529 }
530 # save(err1_multn2s, file = 'T1E_Wei.RData')
531 Pow_multn_2s <- pow_multnlist_2s
532 save(Pow_multn_2s, file = 'Pow_multn_2s.RData')
533
534 end_t <- Sys.time()
535 jobtime <-difftime(end_t, start_t, unit = "hours")
536 outline <- paste(end_t, ": Pow_multn_2s.RData"," is finished. Time difference is ", jobtime,sep="")
537 print(outline)
538 flush.console()
539 # remove unecessary things causing system slowing down
540 rm(pow_multn2s, pow_multnlist_2s, Pow_multn_2s)
541
542 # Nul scale while alternative shape
543 # pow_multn2s <- lapply(1:ncol(multndlt_list), function(l) {
544 #   list(vector("list", 4), lapply(1:it_n, function#!/usr/bin/env Rscript
545
546 # close cluster
547 stopCluster(cl)
548
549 # Tell all slaves to close down, and exit the program
550 mpi.quit()
```

B An Adjustment of Kolmogorov-Smirnov Test Under Spatial Autocorrelation: R Codes

B.1 Simulation and Adjustment Estimation for Distributions with Spatial Autocorrelation

```
#!/usr/bin/env Rscript
print(getwd())
set.seed(1234)
##############################################################################################
##############################################################################################
################################### Author: Wenjun Zheng #####################################
#################################### Date: 09-04-2018 ########################################
#################### Title: A simulation study of Spatial Adjustment #########################
##############################################################################################
##############################################################################################
# use Gaussian copula, due to the property of copula, it may change correlation
# if (!is.loaded("mpi_initialize")) {
#   library("Rmpi")
# }
library(snow)

# generate cluster in MPI type
ncs <- parallel::detectCores()
avilable_mpi_ncs <- ncs
cl <- makeCluster(avilable_mpi_ncs, type = "SOCK")

# pass necessary packages to load in clusters
clusterEvalQ(cl, library(psych))
clusterEvalQ(cl, library(MASS))

##############################################################################################
################################### Preparing functions ######################################
############################################ Two-Sample ######################################
##############################################################################################
Spa_DP_Gen <- function(weights.dis, dist_p = 'Normal', N_sam,
                       para1 = 0, para2 = 1, mult_p = C(0.5, 0.5)){

  # spatially correlated errors
  # could be directly used as observations in locations if necessary
  if (dist_p == 'Normal'){
    sim_points <- para1 + weights.dis %*% rnorm(N_sam, mean = 0, sd = 1)
  }else if(dist_p == 'Weibull'){
    sim_points <- weights.dis %*% rweibull(N_sam, shape = para1, scale = para2)
  }else if(dist_p == 'Multinomial'){
    sim_points_cont <- weights.dis %*% rnorm(N_sam)
    mult_p_cum <- sapply(1:length(mult_p), function(l) sum(mult_p[1:l]))
    multi_P <- c(-Inf, qnorm(mult_p_cum))
    # cut points into interval
    sim_points <- as.numeric(cut(sim_points_cont, breaks = multi_P, include.lowest = T))
  }
  # sim_points <- 1 + errors
  # Moran.I(as.numeric(sim_points), dists.inv)
  return(sim_points)
}

# function to compute the global and local Moran's I
lisa_Moran <- function(x, w, scaled = T, na.rm = F){
  # remove missing values
  N <- length(x)
  if(na.rm == T){
    x <- as.numeric(na.omit(x))}
  # create standard weighting matrix/vector
  if(scaled == T){
    ROWSUM <- rowSums(w)
    ROWSUM[ROWSUM == 0] <- 1
    w <- w/ROWSUM
  }

  # compute the deviations
  deviation_mean <- x - mean(x)

  # compute the local Moran's I, lisa_M
  # to speed up the procedure, we use matrix form
```

```
  lisa_M <- c((deviation_mean/(sum(deviation_mean^2)/N))*(w%*%deviation_mean))

  # compute the global Moran's I, M.I
  # to speed up the procedure, we use matrix form
  M.I <- as.numeric((N/sum(w))*(t(deviation_mean)%*%w%*%deviation_mean)/sum(deviation_mean^2))

  return(list('Anselin Local Moran I' = lisa_M, 'Moran I' = M.I))
}

MI.adj.ks.test <- function(x, y, alternative = "two.sided", G_Moran_I = c(NULL, NULL),
                           L_Moran_I = list(NULL, NULL), adj_method = NULL){
  x <- x[!is.na(x)]
  y <- y[!is.na(y)]
  n.x <- length(x)
  n.y <- length(y)

  # stop the process if data is not enough
  if (n.x < 1L)
    stop("not enough 'x' data")
  if (isTRUE(adj_method == "Global") || isTRUE(adj_method == "Local")){
    if (is.null(G_Moran_I) && is.null(L_Moran_I[[1]]) && is.null(L_Moran_I[[2]]))
      stop("please insert valid global Moran's I and local Moran's I")}
  w <- c(x, y)
  # compute the superemum distance between tested ecdf/cdf
  z <- cumsum(ifelse(order(w) <= n.x, 1/n.x, -1/n.y))
  z <- z[c(which(diff(sort(w)) != 0), n.x + n.y)]
  STAT_VAL <- switch(alternative, two.sided = max(abs(z)),
                     greater = max(z), less = -min(z))
  PVAL <- NULL
  if (is.null(adj_method))
    PVAL <- 1 - .Call(stats:::C_pSmirnov2x, STAT_VAL, n.x, n.y)
  else if (adj_method == "Global"){
    G_n.x <- (1-G_Moran_I[1])*n.x
    G_n.y <- (1-G_Moran_I[2])*n.y
    PVAL <- 1 - .Call(stats:::C_pSmirnov2x, STAT_VAL, G_n.x, G_n.y)}
  else if (adj_method == "Local"){
    # adjust sample sizes by local Moran's I
    L_n.x <- sum(L_Moran_I[[1]] >= G_Moran_I[1])
    L_n.y <- sum(L_Moran_I[[2]] >= G_Moran_I[2])
    PVAL <- 1 - .Call(stats:::C_pSmirnov2x, STAT_VAL, L_n.x, L_n.y)}
  else if (adj_method == "ICC"){
    # adjusted sample size by ICC
    ICC.xy <- psych::ICC(as.data.frame(matrix(c(x,y), ncol = 2)))$results[2][[1]][3]
    ICC.n.x <- (1-ICC.xy)*n.x
    ICC.n.y <- (1-ICC.xy)*n.y
    PVAL <- 1 - .Call(stats:::C_pSmirnov2x, STAT_VAL, ICC.n.x, ICC.n.y)}

  output <- list('statistic' = STAT_VAL, "p.value" = PVAL)
  return(output)
}
################################################################################################
################################## wrapper for tests ###########################################
################################################################################################
Spa.Complerr.2s <- function(itn = 1000, sh1 = 1,
                        sc1 = 0.5, probm = c(0.1, 0.9), dist = 'Normal',
                        spa_mat, corstr = 0.1, dists_inv = dists.inv, alpha.level = 0.05){
  options(warn=-1)
  test.results <- lapply(vector("list", itn), function(x) vector("list", 4))
  N_mat <- nrow(spa_mat)
  # here p is the strength of autocorrelation
  # if |p| is large then the autocorrelation is weak
  p <- corstr
  # distance matrix between points
  # already have it as dist_sph
  # weights matrix
  # compute the cholesky decomposition
  if (dist == "Weibull"){
    Omega <- exp(-(p^2)*spa_mat)}
  else if (dist == "Normal"){
    Omega <- (sc1^2)*exp(-(p^2)*spa_mat)}
  weights_sph <- chol(Omega)
  weights_inv <- t(weights_sph)

  # this section is for true sample size, however I realized it is too liberal
  # indi.matrix <- matrix(rep(1, nrow(Omega)), ncol = 1)
  # adj.rat <- psych::tr(Omega)/(t(indi.matrix)%*%Omega%*%indi.matrix)

  if (dist == "Weibull"){

    test.results <- lapply(1:itn, function(q){
      # simulate data by Cholesky
      Sim_sph1 <- Spa_DP_Gen(weights.dis = weights_inv, N_sam = N_mat,
                             para1 = sh1, para2 = sc1, dist_p = dist)
      Sim_sph2 <- Spa_DP_Gen(weights.dis = weights_inv, N_sam = N_mat,
                             para1 = sh1, para2 = sc1, dist_p = dist)

      # to compute the Moran's I therefore to adjust
      MoranI_1_bug <- lisa_Moran(Sim_sph1, dists_inv, scaled = T, na.rm = T)
```

```
      MoranI_2_bug <- lisa_Moran(Sim_sph2, dists_inv, scaled = T, na.rm = T)

      GM1 <- MoranI_1_bug[[2]]
      GM2 <- MoranI_2_bug[[2]]


      LM1 <- MoranI_1_bug[[1]]
      LM1_R <- sum(abs(LM1) <= abs(MoranI_1_bug[[1]]))/N_mat
      LM2 <- MoranI_2_bug[[1]]
      LM2_R <- sum(abs(LM1) <= abs(MoranI_1_bug[[2]]))/N_mat

      ks_2sam <- stats::ks.test(Sim_sph1, Sim_sph2)


      test.results.temp <- list(list("Local Moran's I" = list(LM1_R, LM2_R),
                                     "Global Moran's I" = list(GM1, GM2)),
                                list("Original Two-sample Kolmogorov-Smirnov Statistic"= ks_2sam$statistic))
    })
  }else if (dist == "Normal"){

    test.results <- lapply(1:itn, function(q){

      # simulate data by Cholesky
      Sim_sph1 <- Spa_DP_Gen(weights.dis = weights_inv, N_sam = N_mat,
                             para1 = sh1, para2 = sc1, dist_p = dist)
      Sim_sph2 <- Spa_DP_Gen(weights.dis = weights_inv, N_sam = N_mat,
                             para1 = sh1, para2 = sc1, dist_p = dist)

      # to compute the Moran's I therefore to adjust
      MoranI_1_bug <- lisa_Moran(Sim_sph1, dists_inv, scaled = T, na.rm = T)
      MoranI_2_bug <- lisa_Moran(Sim_sph2, dists_inv, scaled = T, na.rm = T)

      GM1 <- MoranI_1_bug[[2]]
      GM2 <- MoranI_2_bug[[2]]

      LM1 <- MoranI_1_bug[[1]]
      LM1_R <- sum(abs(LM1) <= abs(MoranI_1_bug[[1]]))/N_mat
      LM2 <- MoranI_2_bug[[1]]
      LM2_R <- sum(abs(LM1) <= abs(MoranI_1_bug[[2]]))/N_mat

      ks_2sam <- stats::ks.test(Sim_sph1, Sim_sph2)


      test.results.temp <- list(list("Local Moran's I" = list(LM1_R, LM2_R),
                                     "Global Moran's I" = list(GM1, GM2)),
                                list("Original Two-sample Kolmogorov-Smirnov Statistic"= ks_2sam$statistic))
    })

   }#else if (dist == "Multinomial"){


  options(warn=0)
  if(dist == "Multinomial"){
    outlist<-c(probm)}else{ outlist<- c(sh1, sc1)}
  return(list('Parameters' = outlist,
              'Correlation Strength' = corstr, 'Iteration times' = itn, 'distribution' = dist,
              'Results List' = test.results))
}

# pass function to clusters

clusterExport(cl, list('Spa_DP_Gen'))
clusterExport(cl, list('lisa_Moran', 'MI.adj.ks.test'))
clusterExport(cl, list('Spa.Comp1err.2s'))

# set seed to ensure reproduction
parallel::clusterSetRNGStream(cl, iseed = 1234)

# decide the total number of iterations needed
tot_itn <- 10000
# calculate iterations needed for each computing core
it_n <- ceiling(tot_itn/(avilable_mpi_ncs))

# generate parameter list for normal distribution
mu_para <- c(0, 1)
sigma_para <- c(1, 2)

#  generate normality distribution parameter list
norm_para_list <- t(expand.grid(mu_para, sigma_para))


# Spatial coordinates
spher_to_cart <- function(r, theta, phi) {
  list(r_sph = r,
       theta_sph = theta,
       phi_sph = phi,
       x_car=r*sin(phi)*cos(theta),
       y_car=r*sin(phi)*sin(theta),
```

```
       z_car=r*cos(phi))
}

arcL <- function(p1, p2, r){
  cos_prod <- as.numeric(cos(p1[3])*cos(p2[3]) + sin(p1[3])*sin(p2[3])*cos(p1[2] - p2[2]))
  if (cos_prod > 1 ){
    arclength <- r*(acos(1))
  }else if( cos_prod < -1){
    arclength <- r*(acos(-1))
  }else{
    arclength <- r*(acos(cos_prod))
  }
  names(arclength) <- 'Arclength'
  return(arclength)
}
# this will generate a matrix of 64 columns and 21 rows.
# deleting the first and last observation of phi as phi = 0 or phi =pi was not what we want
coord <- list(phi=c(seq(pi/2, pi, length =23)[-c(1,23)]),
              theta = seq(0,2*pi,length=65)[-c(1)])

scan_matrix <- expand.grid(coord$theta, coord$phi)
# label scan matrix
names(scan_matrix) <- c('theta', 'phi')
# generate spherical coordinates
# first we assign the radius we want as r
radius_t <- 1
spher_coord <- spher_to_cart(radius_t, scan_matrix$theta, scan_matrix$phi)

# distance calculated from xy locations
# dist_sph <- as.matrix(dist(xy))
sph_coords <- as.data.frame(spher_coord)
# compute the arclength for each pair of the locations
# the greatest distance between points is pi(3.141593)
dist_sph <- apply(sph_coords[,1:3], 1, function(i){
  apply(sph_coords[,1:3], 1, function(j){
    arcL(i, j , radius_t)
  })
})

# inverse distance
dists.inv <- 1/dist_sph
# making the inverse distance matrix
diag(dists.inv) <- 0
# distance decreasing strength, weight matrix to the second power
weight.matrix <- exp(dists.inv)
diag(weight.matrix) <- 0
# cor_list <- c(-0.01, -0.1, -0.38, -0.83, -2.9, -6 )
# Moran's I: 0.6, 0.55, 0.4, 0.3, 0.25, 0.15, 0.1, 0.05, 0
cor_list <- c(0.01, 0.02, 1, 1.8, 2.5, 3, 4, 5.5, 8, 50)
# plot the coordinates
clusterExport(cl, "dist_sph")
clusterExport(cl, "dists.inv")
clusterExport(cl, "weight.matrix")
clusterExport(cl, "sph_coords")
clusterExport(cl, "cor_list")

# a function for simulation, note itn is the simulation numbers, sh is shape parameter
# sc is the scale parameter, sig is the correlation matrix, make sure it's 2*2 if two sample
# Two-sample simulation, weibull
# perform the simulation on all parameters # shape = (0.5, 1, 2, 3, 5), scale = (1, 2, 3)

err1_spatial_2s <- lapply(1:ncol(norm_para_list), function(l) {
  list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
err1_spatial_list_2s <- lapply(1:length(cor_list), function(j){
  lapply(1:ncol(norm_para_list), function(l) {
    list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
})

start_t <- Sys.time()
for (q in 1:length(cor_list)){

  err1_spatial_2s <- lapply(1:ncol(norm_para_list), function(l) {
    list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})

  err1_spatial_2s <- apply(norm_para_list, 2, function(l) {
    clusterCall(cl, Spa.Comp1err.2s, itn = it_n, corstr = cor_list[q],
                sh1 = l[1], sc1 = l[2], dist = 'Normal', spa_mat = dist_sph,
                dists_inv = weight.matrix, alpha.level = 0.05)})
  err1_spatial_list_2s[[q]] <- err1_spatial_2s
}
# save(err1_spatial_2s, file = 'T1E_Wei.RData')
save(err1_spatial_list_2s, file = 'T1E_Spa_size_2s_Oct30.RData')


end_t <- Sys.time()
jobtime <-difftime(end_t, start_t, unit = "hours")
outline <- paste(end_t, ": Tests for spatial distributed samples"," is finished. Time difference is ", jobtime,sep="")
print(outline)
```

```
flush.console()

# release memory
rm(err1_spatial_2s, err1_spatial_list_2s)

# close cluster
stopCluster(cl)

# Tell all slaves to close down, and exit the program
# mpi.quit()
```

B.2 Simulation for Distributions with Spatial Autocorrelation

```
#!/usr/bin/env Rscript
print(getwd())
set.seed(1234)
##############################################################################################
##############################################################################################
################################### Author: Wenjun Zheng #####################################
################################### Date: 09-04-2018 #########################################
##################### Title: A simulation study of Spatial Adjustment ########################
##############################################################################################
##############################################################################################
# use Gaussian copula, due to the property of copula, it may change correlation
# if (!is.loaded("mpi_initialize")) {
#   library("Rmpi")
# }

library(snow)
# suppressPackageStartupMessages(library(gmailr))
# generate cluster in MPI type
ncs <- parallel::detectCores()
avilable_mpi_ncs <- ncs
cl <- makeCluster(avilable_mpi_ncs, type = "SOCK")

# pass necessary packages to load in clusters
clusterEvalQ(cl, library(psych))
clusterEvalQ(cl, library(MASS))

##############################################################################################
################################### Preparing functions ######################################
########################################## Two-Sample ########################################
##############################################################################################
Spa_DP_Gen <- function(weights.dis, dist_p = 'Normal', N_sam,
                       para1 = 0, para2 = 1, mult_p = C(0.5, 0.5)){

  # spatially correlated errors
  # could be directly used as observations in locations if necessary
  if (dist_p == 'Normal'){
    sim_points <- para1 + weights.dis %*% rnorm(N_sam, mean = 0, sd = 1)
  }else if(dist_p == 'Weibull'){
    sim_points <- weights.dis %*% rweibull(N_sam, shape = para1, scale = para2)
  }else if(dist_p == 'Multinomial'){
    sim_points_cont <- weights.dis %*% rnorm(N_sam)
    mult_p_cum <- sapply(1:length(mult_p), function(l) sum(mult_p[1:l]))
    multi_P <- c(-Inf, qnorm(mult_p_cum))
    # cut points into interval
    sim_points <- as.numeric(cut(sim_points_cont, breaks = multi_P, include.lowest = T))
  }
  # sim_points <- 1 + errors
  # Moran.I(as.numeric(sim_points), dists.inv)
  return(sim_points)
}

# function to compute the global and local Moran's I
lisa_Moran <- function(x, w, scaled = T, na.rm = F){
  # remove missing values
  N <- length(x)
  if(na.rm == T){
    x <- as.numeric(na.omit(x))}
  # create standard weighting matrix/vector
  if(scaled == T){
    ROWSUM <- rowSums(w)
    ROWSUM[ROWSUM == 0] <- 1
    w <- w/ROWSUM
  }

  # compute the deviations
  deviation_mean <- x - mean(x)

  # compute the local Moran's I, lisa_M
  # to speed up the procedure, we use matrix form
  lisa_M <- c((deviation_mean/(sum(deviation_mean^2)/N))*(w%*%deviation_mean))
```

```
  # compute the global Moran's I, M.I
  # to speed up the procedure, we use matrix form
  M.I <- as.numeric((N/sum(w))*(t(deviation_mean)%*%w%*%deviation_mean)/sum(deviation_mean^2))

  return(list('Anselin Local Moran I' = lisa_M, 'Moran I' = M.I))
}

MI.adj.ks.test <- function(x, y, alternative = "two.sided", G_Moran_I = c(NULL, NULL),
                           L_Moran_I = list(NULL, NULL), adj_method = NULL){
  x <- x[!is.na(x)]
  y <- y[!is.na(y)]
  n.x <- length(x)
  n.y <- length(y)

  # stop the process if data is not enough
  if (n.x < 1L)
    stop("not enough 'x' data")
  if (isTRUE(adj_method == "Global") || isTRUE(adj_method == "Local")){
    if (is.null(G_Moran_I) && is.null(L_Moran_I[[1]]) && is.null(L_Moran_I[[2]]))
      stop("please insert valid global Moran's I and local Moran's I")}
  w <- c(x, y)
  # compute the superemum distance between tested ecdf/cdf
  z <- cumsum(ifelse(order(w) <= n.x, 1/n.x, -1/n.y))
  z <- z[c(which(diff(sort(w)) != 0), n.x + n.y)]
  STAT_VAL <- switch(alternative, two.sided = max(abs(z)),
                     greater = max(z), less = -min(z))
  PVAL <- NULL
  adj_MI <- G_Moran_I + c(1/(n.x - 1), 1/(n.y - 1))
  if (is.null(adj_method))
    PVAL <- 1 - .Call(stats:::C_pSmirnov2x, STAT_VAL, n.x, n.y)
  else if (adj_method == "Global"){
    G_n.x <- ceiling((2/(1+exp(4.018401*adj_MI[1] + 3.881034*adj_MI[1]^3)))*n.x)
    G_n.y <- ceiling((2/(1+exp(4.018401*adj_MI[2] + 3.881034*adj_MI[2]^3)))*n.y)
    PVAL <- 1 - .Call(stats:::C_pSmirnov2x, STAT_VAL, G_n.x, G_n.y)}
  else if (adj_method == "Local"){
    # adjust sample sizes by local Moran's I
    L_n.x <- ceiling((2/(1+exp(1.894057*adj_MI[1] + 5.932520*adj_MI[2]^2)))*n.x)
    L_n.y <- ceiling((2/(1+exp(1.894057*adj_MI[2] + 5.932520*adj_MI[2]^2)))*n.y)
    PVAL <- 1 - .Call(stats:::C_pSmirnov2x, STAT_VAL, L_n.x, L_n.y)}
  else if (adj_method == "ICC"){
    # adjusted sample size by ICC
    ICC.xy <- psych::ICC(as.data.frame(matrix(c(x,y), ncol = 2)))$results[2][[1]][3]
    ICC.n.x <- (1-ICC.xy)*n.x
    ICC.n.y <- (1-ICC.xy)*n.y
    PVAL <- 1 - .Call(stats:::C_pSmirnov2x, STAT_VAL, ICC.n.x, ICC.n.y)}

  output <- list('statistic' = STAT_VAL, "p.value" = PVAL)
  return(output)
}
################################################################################################
################################## wrapper for tests ###########################################
################################################################################################
Spa.Complerr.2s <- function(itn = 1000, sh1 = 1,
                        sc1 = 0.5, probm = c(0.1, 0.9), dist = 'Weibull',
                        spa_mat, corstr = 0.1, dists_inv = dists.inv){
  options(warn=-1)
  test.results <- lapply(vector("list", itn), function(x) vector("list", 4))
  N_mat <- nrow(spa_mat)
  # here p is the strength of autocorrelation
  # if |p| is large then the autocorrelation is weak
  p <- corstr
  # distance matrix between points
  # already have it as dist_sph
  # weights matrix
  # compute the cholesky decomposition
  if (dist == "Weibull"){
    Omega <- exp(-(p^2)*spa_mat)}
  else if (dist == "Normal"){
    Omega <- (sc1^2)*exp(-(p^2)*spa_mat)}
  weights_sph <- chol(Omega)
  weights_inv <- t(weights_sph)
  if (dist == "Weibull"){

    test.results <- lapply(1:itn, function(q){
      # simulate data by Cholesky
      Sim_sph1 <- Spa_DP_Gen(weights.dis = weights_inv, N_sam = N_mat,
                             para1 = sh1, para2 = sc1, dist_p = dist)
      Sim_sph2 <- Spa_DP_Gen(weights.dis = weights_inv, N_sam = N_mat,
                             para1 = sh1, para2 = sc1, dist_p = dist)

      # to compute the Moran's I therefore to adjust
      MoranI_1_bug <- lisa_Moran(Sim_sph1, dists_inv, scaled = T, na.rm = T)
      MoranI_2_bug <- lisa_Moran(Sim_sph2, dists_inv, scaled = T, na.rm = T)

      GM1 <- MoranI_1_bug[[2]]
      GM2 <- MoranI_2_bug[[2]]
```

```r

      LM1 <- MoranI_1_bug[[1]]
      LM2 <- MoranI_2_bug[[1]]

      ks_2sam <- stats::ks.test(Sim_sph1, Sim_sph2)$p.value
      ks_GM_2sam <- MI.adj.ks.test(Sim_sph1, Sim_sph2,
                                   G_Moran_I = c(GM1, GM2), adj_method = 'Global')$p.value
      ks_LM_2sam <- MI.adj.ks.test(Sim_sph1, Sim_sph2, G_Moran_I = c(GM1, GM2),
                                   L_Moran_I = list(LM1, LM2), adj_method = 'Local')$p.value
      ks_ICC_2sam <- MI.adj.ks.test(Sim_sph1, Sim_sph2, adj_method = 'ICC')$p.value

      test.results.temp <- list(list("Global Moran's I" = list(GM1, GM2)),
                                list("Original Two-sample Kolmogorov-Smirnov Test"= ks_2sam,
                                     "Global Moran's I adjusted Two-sample Kolmogorov-Smirnov Test"= ks_GM_2sam,
                                     "Local Moran's I adjusted Two-sample Kolmogorov-Smirnov Test"= ks_LM_2sam,
                                     "ICC adjusted  Two-sample Kolmogorov-Smirnov Test"= ks_ICC_2sam))
    })
  }else if (dist == "Normal"){

    test.results <- lapply(1:itn, function(q){

      # simulate data by Cholesky
      Sim_sph1 <- Spa_DP_Gen(weights.dis = weights_inv, N_sam = N_mat,
                             para1 = sh1, para2 = sc1, dist_p = dist)
      Sim_sph2 <- Spa_DP_Gen(weights.dis = weights_inv, N_sam = N_mat,
                             para1 = sh1, para2 = sc1, dist_p = dist)

      # to compute the Moran's I therefore to adjust
      MoranI_1_bug <- lisa_Moran(Sim_sph1, dists_inv, scaled = T, na.rm = T)
      MoranI_2_bug <- lisa_Moran(Sim_sph2, dists_inv, scaled = T, na.rm = T)

      GM1 <- MoranI_1_bug[[2]]
      GM2 <- MoranI_2_bug[[2]]

      LM1 <- MoranI_1_bug[[1]]
      LM2 <- MoranI_2_bug[[1]]

      ks_2sam <- stats::ks.test(Sim_sph1, Sim_sph2)$p.value
      ks_GM_2sam <- MI.adj.ks.test(Sim_sph1, Sim_sph2,
                                   G_Moran_I = c(GM1, GM2), adj_method = 'Global')$p.value
      ks_LM_2sam <- MI.adj.ks.test(Sim_sph1, Sim_sph2, G_Moran_I = c(GM1, GM2),
                                   L_Moran_I = list(LM1, LM2), adj_method = 'Local')$p.value
      ks_ICC_2sam <- MI.adj.ks.test(Sim_sph1, Sim_sph2, adj_method = 'ICC')$p.value

      test.results.temp <- list(list("Global Moran's I" = list(GM1, GM2)),
                                list("Original Two-sample Kolmogorov-Smirnov Test"= ks_2sam,
                                     "Global Moran's I adjusted Two-sample Kolmogorov-Smirnov Test"= ks_GM_2sam,
                                     "Local Moran's I adjusted Two-sample Kolmogorov-Smirnov Test"= ks_LM_2sam,
                                     "ICC adjusted Two-sample Kolmogorov-Smirnov Test"= ks_ICC_2sam))
    })

   }#else if (dist == "Multinomial"){
#
  err_list <- lapply(test.results, function(c) c[[2]] < 0.05)
  ks_err <- mean(sapply(err_list, function(l) l[[1]]), na.rm = T)
  ks_G_err <- mean(sapply(err_list, function(l) l[[2]]), na.rm = T)
  ks_L_err <- mean(sapply(err_list, function(l) l[[3]]), na.rm = T)
  ks_ICC_err <- mean(sapply(err_list, function(l) l[[4]]), na.rm = T)

  type1err <- list("Type I error of Original Kolmogorov-Smirnov Test"=ks_err,
                   "Type I error of Global Moran's I adjusted Kolmogorov-Smirnov Test"=ks_G_err,
                   "Type I error of Local Moran's I adjusted Kolmogorov-Smirnov Test"=ks_L_err,
                   "Type I error of ICC adjusted Kolmogorov-Smirnov Test"=ks_ICC_err)
  options(warn=0)
  if(dist == "Multinomial"){
    outlist<-c(probm)}else{ outlist<- c(sh1, sc1)}
  return(list('Parameters' = outlist,
              'Correlation Strength' = corstr, 'Iteration times' = itn, 'distribution' = dist,
              'Type I error'= type1err,
              'Results List' = test.results))
}

################################### Power Calculation ######################################
Spa.ComPower.2s <- function(itn = 1000, sh1 = 1, sc1 = 0.5,
                            sh2 = 1, sc2 = 0.5, probm = c(0.1, 0.9),
                            probm2 = c(0.1, 0.9), dist = 'Weibull',
                            spa_mat, corstr = 0.1, dists_inv = dists.inv){
  options(warn=-1)
  test.results <- lapply(vector("list", itn), function(x) vector("list", 4))
  N_mat <- nrow(spa_mat)
  # here p is the strength of autocorrelation
  # if |p| is large then the autocorrelation is weak
  p <- corstr
  # distance matrix between points
  # already have it as dist_sph
  # weights matrix
  # compute the cholesky decomposition
  if (dist == "Weibull"){
```

```
    Omega <- exp(-(p^2)*spa_mat)
    weights_sph <- chol(Omega)
    weights_inv <- t(weights_sph)}
  else if (dist == "Normal"){
    Omega1 <- (sc1^2)*exp(-(p^2)*spa_mat)
    Omega2 <- (sc2^2)*exp(-(p^2)*spa_mat)
    weights_sph1 <- chol(Omega1)
    weights_inv1 <- t(weights_sph1)
    weights_sph2 <- chol(Omega2)
    weights_inv2 <- t(weights_sph2)}


  if (dist == "Weibull"){

    test.results <- lapply(1:itn, function(q){
      Sim_sph1 <- Spa_DP_Gen(weights.dis = weights_inv, N_sam = N_mat,
                             para1 = sh1, para2 = sc1, dist_p = dist)
      Sim_sph2 <- Spa_DP_Gen(weights.dis = weights_inv, N_sam = N_mat,
                             para1 = sh2, para2 = sc2, dist_p = dist)

      # to compute the Moran's I therefore to adjust
      MoranI_1_bug <- lisa_Moran(Sim_sph1, dists_inv, scaled = T, na.rm = T)
      MoranI_2_bug <- lisa_Moran(Sim_sph2, dists_inv, scaled = T, na.rm = T)

      GM1 <- MoranI_1_bug[[2]]
      GM2 <- MoranI_2_bug[[2]]

      LM1 <- MoranI_1_bug[[1]]
      LM2 <- MoranI_2_bug[[1]]

      ks_2sam <- stats::ks.test(Sim_sph1, Sim_sph2)$p.value
      ks_GM_2sam <- MI.adj.ks.test(Sim_sph1, Sim_sph2,
                                   G_Moran_I = c(GM1, GM2), adj_method = 'Global')$p.value
      ks_LM_2sam <- MI.adj.ks.test(Sim_sph1, Sim_sph2, G_Moran_I = c(GM1, GM2),
                                   L_Moran_I = list(LM1, LM2), adj_method = 'Local')$p.value
      ks_ICC_2sam <- MI.adj.ks.test(Sim_sph1, Sim_sph2, adj_method = 'ICC')$p.value

      test.results.temp <- list(list("Global Moran's I" = list(GM1, GM2)),
                                list("Original Two-sample Kolmogorov-Smirnov Test"= ks_2sam,
                                     "Global Moran's I adjusted Two-sample Kolmogorov-Smirnov Test"= ks_GM_2sam,
                                     "Local Moran's I adjusted Two-sample Kolmogorov-Smirnov Test"= ks_LM_2sam,
                                     "ICC adjusted  Two-sample Kolmogorov-Smirnov Test"= ks_ICC_2sam))
    })

  }
  else if (dist == "Normal"){

    test.results <- lapply(1:itn, function(q){
      Sim_sph1 <- Spa_DP_Gen(weights.dis = weights_inv1, N_sam = N_mat,
                             para1 = sh1, para2 = sc1, dist_p = dist)
      Sim_sph2 <- Spa_DP_Gen(weights.dis = weights_inv2, N_sam = N_mat,
                             para1 = sh2, para2 = sc2, dist_p = dist)

      # to compute the Moran's I therefore to adjust
      MoranI_1_bug <- lisa_Moran(Sim_sph1, dists_inv, scaled = T, na.rm = T)
      MoranI_2_bug <- lisa_Moran(Sim_sph2, dists_inv, scaled = T, na.rm = T)

      GM1 <- MoranI_1_bug[[2]]
      GM2 <- MoranI_2_bug[[2]]

      LM1 <- MoranI_1_bug[[1]]
      LM2 <- MoranI_2_bug[[1]]

      ks_2sam <- stats::ks.test(Sim_sph1, Sim_sph2)$p.value
      ks_GM_2sam <- MI.adj.ks.test(Sim_sph1, Sim_sph2,
                                   G_Moran_I = c(GM1, GM2), adj_method = 'Global')$p.value
      ks_LM_2sam <- MI.adj.ks.test(Sim_sph1, Sim_sph2, G_Moran_I = c(GM1, GM2),
                                   L_Moran_I = list(LM1, LM2), adj_method = 'Local')$p.value
      ks_ICC_2sam <- MI.adj.ks.test(Sim_sph1, Sim_sph2, adj_method = 'ICC')$p.value

      test.results.temp <- list(list("Global Moran's I" = list(GM1, GM2)),
                                list("Original Two-sample Kolmogorov-Smirnov Test"= ks_2sam,
                                     "Global Moran's I adjusted Two-sample Kolmogorov-Smirnov Test"= ks_GM_2sam,
                                     "Local Moran's I adjusted Two-sample Kolmogorov-Smirnov Test"= ks_LM_2sam,
                                     "ICC adjusted  Two-sample Kolmogorov-Smirnov Test"= ks_ICC_2sam))
    })

  }else if (dist == "Multinomial"){

    test.results <- lapply(1:itn, function(q){
      x1 <- rmultinom(n=1, 1344, prob = probm)
      # categorize data
      x2 <- rmultinom(n=1, 1344, prob = probm2)

      x1_dt <- unlist(apply(as.data.frame(1:length(x1)), 1, function(l){rep(l, x1[l])}))
      x2_dt <- unlist(apply(as.data.frame(1:length(x2)), 1, function(l){rep(l, x2[l])}))
      x2_ecdf <- ecdf(unlist(apply(as.data.frame(1:length(x2)), 1, function(l){rep(l, x2[l])})))

```

```
      ks_2sam <- tryCatch({KSgeneral::disc_ks_test(x1_dt, x2_ecdf, exact = T)$p.value},
                          error = function(e){ return(NA)} )


      test.results.temp <- list("Two-sample Kolmogorov-Smirnov Test"=ks_2sam)
    })

  }

  power_list <- lapply(test.results, function(c) c[[2]] < 0.05)

  ks_power <- mean(sapply(power_list, function(l) l[[1]]), na.rm = T)
  ks_G_power <- mean(sapply(power_list, function(l) l[[2]]), na.rm = T)
  ks_L_power <- mean(sapply(power_list, function(l) l[[3]]), na.rm = T)
  ks_ICC_power <- mean(sapply(power_list, function(l) l[[4]]), na.rm = T)

  powerlist <- list("Power of Kolmogorov-Smirnov Test"=ks_power,
                    "Power of Global Moran's I adjusted Kolmogorov-Smirnov Test"=ks_G_power,
                    "Power of Local Moran's I adjusted Kolmogorov-Smirnov Test"=ks_L_power,
                    "Power of ICC adjusted Kolmogorov-Smirnov Test"=ks_ICC_power)
  options(warn=0)
  if(dist == "Multinomial"){
    outlist<-c('null'=probm, 'alternative'=probm2)}else{ outlist<- c(sh1, sc1)}
  return(list('Parameters' = outlist, 'Iteration times' = itn, 'distribution' = dist,
              'MC power' = powerlist, 'P-value List' = test.results))
}

# pass function to clusters

clusterExport(cl, list('Spa_DP_Gen'))
clusterExport(cl, list('lisa_Moran', 'MI.adj.ks.test'))
clusterExport(cl, list('Spa.Comp1err.2s'))
clusterExport(cl, list('Spa.ComPower.2s'))


parallel::clusterSetRNGStream(cl, iseed = 1234)


# decide the total number of iterations needed
tot_itn <- 10000
# calculate iterations needed for each computing core
it_n <- ceiling(tot_itn/(avilable_mpi_ncs))

# generate parameter list for normal distribution
# mu_para <- c(0.5, 2)
# sigma_para <- c(0.9, 1.5, 3)

mu_para <- c(0)
sigma_para <- c(1)
#  generate normality distribution parameter list
norm_para_list <- t(expand.grid(mu_para, sigma_para))
# generate normality distritbuion list for power analysis
para_dlt <- c(0.05, 0.1, 0.2, 0.5, 1)
norm_dlt_list <- t(expand.grid(para_dlt, mu_para, sigma_para))
norm_dlt_list <- rbind(norm_dlt_list, norm_dlt_list[1,]*norm_dlt_list[2,],
                       norm_dlt_list[1,]*norm_dlt_list[3,] )
norm_dlt_list[4,] <- norm_dlt_list[2,] + norm_dlt_list[4,]
norm_dlt_list[5,] <- norm_dlt_list[3,] + norm_dlt_list[5,]
rownames(norm_dlt_list) <- c('dlt', 'nul_mu', 'nul_sd', 'al_mu', 'al_sd')
norm_dlt_list <- norm_dlt_list[-1,]

# Spatial coordinates
spher_to_cart <- function(r, theta, phi) {
  list(r_sph = r,
       theta_sph = theta,
       phi_sph = phi,
       x_car=r*sin(phi)*cos(theta),
       y_car=r*sin(phi)*sin(theta),
       z_car=r*cos(phi))
}

arcL <- function(p1, p2, r){
  cos_prod <- as.numeric(cos(p1[3])*cos(p2[3]) + sin(p1[3])*sin(p2[3])*cos(p1[2] - p2[2]))
  if (cos_prod > 1 ){
    arclength <- r*(acos(1))
  }else if( cos_prod < -1){
    arclength <- r*(acos(-1))
  }else{
    arclength <- r*(acos(cos_prod))
  }
  names(arclength) <- 'Arclength'
  return(arclength)
}
# this will generate a matrix of 64 columns and 21 rows.
# deleting the first and last observation of phi as phi = 0 or phi =pi was not what we want
coord <- list(phi=c(seq(pi/2, pi, length =23)[-c(1,23)]),
              theta = seq(0,2*pi,length=65)[-c(1)])
```

```r

scan_matrix <- expand.grid(coord$theta, coord$phi)
# label scan matrix
names(scan_matrix) <- c('theta', 'phi')
# generate spherical coordinates
# first we assign the radius we want as r
radius_t <- 1
spher_coord <- spher_to_cart(radius_t, scan_matrix$theta, scan_matrix$phi)

# distance calculated from xy locations
# dist_sph <- as.matrix(dist(xy))
sph_coords <- as.data.frame(spher_coord)
# compute the arclength for each pair of the locations
# the greatest distance between points is pi(3.141593)
dist_sph <- apply(sph_coords[,1:3], 1, function(i){
  apply(sph_coords[,1:3], 1, function(j){
    arcL(i, j , radius_t)
  })
})

# inverse distance
dists.inv <- 1/dist_sph
# making the inverse distance matrix
diag(dists.inv) <- 0
# inverse distance to the second power
weight.matrix <- exp(dists.inv)
diag(weight.matrix) <- 0

# Moran's I: 1.00 0.90 0.85 0.80 0.75 0.70 0.65 0.60
#            0.55 0.50 0.45 0.40 0.35 0.30 0.25
#            0.20 0.15 0.10 0.05 0.00
cor_list <- c(0.01, 1, 1.5, 1.9, 2.25, 2.5, 2.8, 3.05,
              3.36, 3.64, 3.93, 4.25, 4.58, 4.96, 5.4,
              5.95, 6.65, 7.7, 9.8, 30)
# Moran's I:  0.90 0.70 0.50 0.30 0.10 0.00
cor_list <- c(1, 2.5, 3.64, 4.96, 7.7, 30)
# plot the coordinates
clusterExport(cl, "dist_sph")
clusterExport(cl, "weight.matrix")
clusterExport(cl, "sph_coords")
clusterExport(cl, "cor_list")

# a function for simulation, note itn is the simulation numbers, sh is shape parameter
# sc is the scale parameter, sig is the correlation matrix, make sure it's 2*2 if two sample
# Two-sample simulation, weibull
# perform the simulation on all parameters # shape = (0.5, 1, 2, 3, 5), scale = (1, 2, 3)

err1_spatial_2s <- lapply(1:ncol(norm_para_list), function(l) {
  list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
err1_spatial_list_2s <- lapply(1:length(cor_list), function(j){
  lapply(1:ncol(norm_para_list), function(l) {
    list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
})

start_t <- Sys.time()
for (q in 1:length(cor_list)){
  err1_spatial_2s <- apply(norm_para_list, 2, function(l) {
    clusterCall(cl, Spa.Comp1err.2s, itn = it_n, corstr = cor_list[q],
                sh1 = l[1], sc1 = l[2], dist = 'Normal', spa_mat = dist_sph, dists_inv = weight.matrix)})
  err1_spatial_list_2s[[q]] <- err1_spatial_2s
}
# save(err1_spatial_2s, file = 'T1E_Wei.RData')
save(err1_spatial_list_2s, file = 'T1E_Spa_size_2s_test_NOV08.RData')

end_t <- Sys.time()
jobtime <-difftime(end_t, start_t, unit = "hours")
outline <- paste(end_t, ": Tests for spatial distributed samples"," is finished. Time difference is ", jobtime,sep="")
# finish_mail <- mime(
#   To = "wenjun.zheng@aol.com",
#   From = "van0604@gmail.com",
#   Subject = "Simulation Job Finished",
#   body = outline)
# send_message(finish_mail)
print(outline)
flush.console()
# remove unecessary things causing system slowing down
rm(err1_spatial_2s, err1_spatial_list_2s)

#############################################################################################
################################### Power study #############################################
#############################################################################################
# Nul shape while alternative scale
pow_spatial_2s <- lapply(1:ncol(norm_dlt_list), function(l) {
  list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
pow_spatial_list_2s <- lapply(1:length(cor_list), function(j){
  lapply(1:ncol(norm_dlt_list), function(l) {
    list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
})
```

```
start_t <- Sys.time()
for (q in 1:length(cor_list)){
  pow_spatial_2s <- apply(norm_dlt_list, 2, function(l) {
    clusterCall(cl,Spa.ComPower.2s, itn = it_n, corstr = cor_list[q],
                sh1 = l[1], sc1 = l[2], sh2 <- l[3], sc2 <- l[2],
                dist = 'Normal', spa_mat = dist_sph, dists_inv = weight.matrix)})
  pow_spatial_list_2s[[q]] <- pow_spatial_2s
}
# save(err1_spatial_2s, file = 'T1E_Wei.RData')
Pow_Spa_2s_Nullmu <- pow_spatial_list_2s
save(Pow_Spa_2s_Nullmu, file = 'Pow_Spa_2s_Nullmu.RData')

end_t <- Sys.time()
jobtime <-difftime(end_t, start_t, unit = "hours")
outline <- paste(end_t, ": Tests for spatial distributed samples"," is finished. Time difference is ", jobtime,sep="")
print(outline)
flush.console()
# finish_mail <- mime(
#   To = "wenjun.zheng@aol.com",
#   From = "van0604@gmail.com",
#   Subject = "Simulation Job Finished",
#   body = outline)
# send_message(finish_mail)
# remove unecessary things causing system slowing down
rm(pow_spatial_2s, pow_spatial_list_2s, Pow_Spa_2s_Nullmu)

# Nul scale while alternative shape
pow_spatial_2s <- lapply(1:ncol(norm_para_list), function(l) {
  list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
pow_spatial_list_2s <- lapply(1:length(cor_list), function(j){
  lapply(1:ncol(norm_para_list), function(l) {
    list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
})

start_t <- Sys.time()
for (q in 1:length(cor_list)){
  pow_spatial_2s <- apply(norm_dlt_list, 2, function(l) {
    clusterCall(cl,Spa.ComPower.2s, itn = it_n, corstr = cor_list[q],
                sh1 = l[1], sc1 = l[2], sh2 <- l[1], sc2 <- l[4],
                dist = 'Normal', spa_mat = dist_sph, dists_inv = weight.matrix)})
  pow_spatial_list_2s[[q]] <- pow_spatial_2s
}
# save(err1_spatial_2s, file = 'T1E_Wei.RData')
Pow_Spa_2s_Nullvar <- pow_spatial_list_2s
save(Pow_Spa_2s_Nullvar, file = 'Pow_Spa_2s_Nullvar.RData')

end_t <- Sys.time()
jobtime <-difftime(end_t, start_t, unit = "hours")
outline <- paste(end_t, ": Tests for spatial distributed samples"," is finished. Time difference is ", jobtime,sep="")
print(outline)
flush.console()
# finish_mail <- mime(
#   To = "wenjun.zheng@aol.com",
#   From = "van0604@gmail.com",
#   Subject = "Simulation Job Finished",
#   body = outline)
# send_message(finish_mail)
# remove unecessary things causing system slowing down
rm(pow_spatial_2s, pow_spatial_list_2s, Pow_Spa_2s_Nullvar)

# Alternative scale, alternative shape
pow_spatial_2s <- lapply(1:ncol(norm_para_list), function(l) {
  list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
pow_spatial_list_2s <- lapply(1:length(cor_list), function(j){
  lapply(1:ncol(norm_para_list), function(l) {
    list(vector("list", 4), lapply(1:it_n, function(k){vector("list", 4)}))})
})

start_t <- Sys.time()
for (q in 1:length(cor_list)){
  pow_spatial_2s <- apply(norm_dlt_list, 2, function(l) {
    clusterCall(cl,Spa.ComPower.2s, itn = it_n, corstr = cor_list[q],
                sh1 = l[1], sc1 = l[2], sh2 <- l[3], sc2 <- l[4],
                dist = 'Normal', spa_mat = dist_sph, dists_inv = weight.matrix)})
  pow_spatial_list_2s[[q]] <- pow_spatial_2s
}
# save(err1_spatial_2s, file = 'T1E_Wei.RData')
Pow_Spa_2s_alt <- pow_spatial_list_2s
save(Pow_Spa_2s_alt, file = 'Pow_Spa_2s_alt.RData')

end_t <- Sys.time()
jobtime <-difftime(end_t, start_t, unit = "hours")
outline <- paste(end_t, ": Tests for spatial distributed samples"," is finished. Time difference is ", jobtime,sep="")
print(outline)
flush.console()
# finish_mail <- mime(
#   To = "wenjun.zheng@aol.com",
```

```
#   From = "van0604@gmail.com",
#   Subject = "Simulation Job Finished",
#   body = outline)
# send_message(finish_mail)

# remove unecessary things causing system slowing down
rm(pow_spatial_2s, pow_spatial_list_2s, Pow_Spa_2s_alt)
#
# close cluster
stopCluster(cl)
```

B.3 Comparison of I_A vs. Moran's I

```
library(ggplot2)

w <- weight.matrix
ROWSUM <- rowSums(w)
ROWSUM[ROWSUM == 0] <- 1
w <- w/ROWSUM

out_temp <- matrix(ncol = 2)
for (i in seq(0.01, 10, by = 0.01)){
  Omega <- exp(-(i^2)*dist_sph)
  # calculate expected Moran's I in respect to given strength of autocorrelation
  weighted.cov.matrix <- w * Omega
  M.I <- sum(weighted.cov.matrix)/1344
  out_temp <- rbind(out_temp, t(as.matrix(c(i, M.I), ncol = 1)))
}
out_plot <- as.data.frame(out_temp[-1,])
ggplot(out_plot, aes(x = V1, y = V2)) +
  geom_point() +
  labs( x = 'Strength', y = "Moran's I") +
  xlim(0, 5) +
  ylim(0, 1) +
  theme_classic()

simulateM <- as.data.frame(cbind(cor_list, unique(moranS)))
colnames(simulateM)<- c("V1", "V2")
ggplot(simulateM, aes(x = cor_list, y = simulated_M)) +
  geom_point() +
  labs( x = 'Strength', y = "Simulated Moran's I") +
  xlim(0, 5) +
  ylim(0, 1) +
  theme_classic()

#Moran's I plot
out_plot$grp <- 'Calculated'
simulateM$grp <- "Simulated"
MIP <- rbind(out_plot, simulateM)
ggplot(MIP, aes(x = V1, y = V2, group = grp, col = as.factor(grp))) +
  geom_point() +
  labs( x = 'Strength', y = "Moran's I", col = "Moran's I") +
  xlim(1.5, 10) +
  ylim(0, 1) +
  theme_classic()
```

C Comparing Heart PET Scans: A Revision of Komogorov-Smirnov Test: R Codes

C.1 Pre-Defined Functions

```r
# library to attach and load
library(ape)
library(rgl)
# first we write a function to generate spherical coordinates
# formula reference: https://mathinsight.org/spherical_coordinates
spher_to_cart <- function(r, theta, phi) {
  list(r_sph = r,
       theta_sph = theta,
       phi_sph = phi,
       x_car=r*sin(phi)*cos(theta),
       y_car=r*sin(phi)*sin(theta),
       z_car=r*cos(phi))
}

arcL <- function(p1, p2, r){
  cos_prod <- as.numeric(cos(p1[3])*cos(p2[3]) + sin(p1[3])*sin(p2[3])*cos(p1[2] - p2[2]))
  if (cos_prod > 1 ){
    arclength <- r*(acos(1))
  }else if( cos_prod < -1){
    arclength <- r*(acos(-1))
  }else{
    arclength <- r*(acos(cos_prod))
  }
  names(arclength) <- 'Arclength'
  return(arclength)
}
# this will generate a matrix of 64 columns and 21 rows.
# deleting the first and last observation of phi as phi = 0 or phi =pi was not what we want
coord <- list(phi=c(seq(pi/2, pi, length =23)[-c(1,23)]),
              theta = seq(0,2*pi,length=65)[-c(1)])

scan_matrix <- expand.grid(coord$theta, coord$phi)
# label scan matrix
names(scan_matrix) <- c('theta', 'phi')
scan_matrix$row <- rep(c(1:21), each = 64)
scan_matrix$radial <- rep(c(1:64), 21)

# generate spherical coordinates
# first we assign the radius we want as r
radius_t <- 1

spher_coord <- spher_to_cart(radius_t, scan_matrix$theta, scan_matrix$phi)

# plot the coordinates, unmark if not necessary
heart_plot <- rgl::plot3d(spher_coord$x_car,spher_coord$y_car,spher_coord$z_car, xlab = "x", ylab = "y", zlab = "z")

# dist_sph <- as.matrix(dist(xy))
sph_coords <- as.data.frame(spher_coord)
# compute the arclength for each pair of the locations
start.time <- Sys.time()
# the greatest distance between points is pi(3.141593)
dist_sph <- apply(sph_coords[,1:3], 1, function(i){
  apply(sph_coords[,1:3], 1, function(j){
    arcL(i, j , radius_t)
  })
})
end.time <- Sys.time()
jobtime <-difftime(end.time, start.time, unit = "auto")
jobtime
# inverse distance
dists.inv <- 1/dist_sph
# making the inverse distance matrix
diag(dists.inv) <- 0

# inverse distance to the alpha's power, dists,inv^a
weight_Matrix <- dists.inv^2
diag(weight_Matrix) <- 0

ROWSUM <- rowSums(weight_Matrix)
ROWSUM[ROWSUM == 0] <- 1
```

```
w <- weight_Matrix/ROWSUM

# function to compute the global and local Moran's I
lisa_Moran <- function(x, w, scaled = T, na.rm = F){
  # remove missing values
  N <- length(x)
  if(na.rm == T){
    x <- as.numeric(na.omit(x))}
  # create standard weighting matrix/vector
  if(scaled == T){
    ROWSUM <- rowSums(w)
    ROWSUM[ROWSUM == 0] <- 1
    w <- w/ROWSUM
  }

  # compute the deviations
  deviation_mean <- x - mean(x)

  # compute the local Moran's I, lisa_M
  # to speed up the procedure, we use matrix form
  lisa_M <- c((deviation_mean/(sum(deviation_mean^2)/N))*(w%*%deviation_mean))

  # compute the global Moran's I, M.I
  # to speed up the procedure, we use matrix form
  M.I <- as.numeric((N/sum(w))*(t(deviation_mean)%*%w%*%deviation_mean)/sum(deviation_mean^2))

  return(list('Anselin Local Moran I' = lisa_M, 'Moran I' = M.I))
}

MI.adj.ks.test <- function(x, y, alternative = "two.sided", G_Moran_I = c(NULL, NULL),
                           L_Moran_I = list(NULL, NULL), adj_method = NULL){
  x <- x[!is.na(x)]
  y <- y[!is.na(y)]
  n.x <- length(x)
  n.y <- length(y)

  # stop the process if data is not enough
  if (n.x < 1L)
    stop("not enough 'x' data")
  if (isTRUE(adj_method == "Global") || isTRUE(adj_method == "Local")){
    if (is.null(G_Moran_I) && is.null(L_Moran_I[[1]]) && is.null(L_Moran_I[[2]]))
      stop("please insert valid global Moran's I and local Moran's I")}
  w <- c(x, y)
  # compute the superemum distance between tested ecdf/cdf
  z <- cumsum(ifelse(order(w) <= n.x, 1/n.x, -1/n.y))
  z <- z[c(which(diff(sort(w)) != 0), n.x + n.y)]
  STAT_VAL <- switch(alternative, two.sided = max(abs(z)),
                     greater = max(z), less = -min(z))
  PVAL <- NULL
  adj_MI <- G_Moran_I + c(1/(n.x - 1), 1/(n.y - 1))
  if (is.null(adj_method))
    PVAL <- 1 - .Call(stats:::C_pSmirnov2x, STAT_VAL, n.x, n.y)
  else if (adj_method == "Global"){
    # adjusted sample size by global Moran's I
    # 1-(1/(1-exp(-1.92369)))*(1343/1344)*(1-exp(-2.12373*0.2+0.20024*sqrt(0.2)))
    # G_n.x <- (1-(1/(1-exp(-1.92369)))*(1343/1344)*(1-exp(-2.12373*G_Moran_I[1]+0.20024*sqrt(G_Moran_I[1]))))*n.x
    # G_n.y <- (1-(1/(1-exp(-1.92369)))*(1343/1344)*(1-exp(-2.12373*G_Moran_I[2]+0.20024*sqrt(G_Moran_I[2]))))*n.y
    G_n.x <- ceiling((2/(1+exp(3.934*adj_MI[1] + 3.172*adj_MI[1]^3)))*n.x)
    G_n.y <- ceiling((2/(1+exp(3.934*adj_MI[2] + 3.172*adj_MI[2]^3)))*n.y)
    PVAL <- 1 - .Call(stats:::C_pSmirnov2x, STAT_VAL, G_n.x, G_n.y)}
  else if (adj_method == "Local"){
    # adjust sample sizes by local Moran's I
    adj_MI2 <- ifelse(adj_MI < 0, 0, adj_MI)
    L_n.x <- ceiling((1-(1/(1-exp(-1.92369)))*((n.x -1)/n.x)*(1-exp(-2.124*adj_MI2[1] + 0.2*sqrt(adj_MI2[1]))))*n.x)
    L_n.y <- ceiling((1-(1/(1-exp(-1.92369)))*((n.x -1)/n.x)*(1-exp(-2.124*adj_MI2[2] + 0.2*sqrt(adj_MI2[2]))))*n.y)
    PVAL <- 1 - .Call(stats:::C_pSmirnov2x, STAT_VAL, L_n.x, L_n.y)}
  else if (adj_method == "ICC"){
    # adjusted sample size by ICC
    ICC.xy <- 0.5
    ICC.n.x <- (1-ICC.xy)*n.x
    ICC.n.y <- (1-ICC.xy)*n.y
    PVAL <- 1 - .Call(stats:::C_pSmirnov2x, STAT_VAL, ICC.n.x, ICC.n.y)}

  output <- list('statistic' = STAT_VAL, "p.value" = PVAL)
  return(output)
}

MI.adj.ks.test.discret <- function(x, y, alternative = "two.sided", G_Moran_I = c(NULL, NULL),
                           L_Moran_I = list(NULL, NULL), adj_method = NULL){
  x <- x[!is.na(x)]
  y <- y[!is.na(y)]
  n.x <- sum(x)
  n.y <- sum(y)

  # stop the process if data is not enough
  if (n.x < 1L)
    stop("not enough 'x' data")
  if (isTRUE(adj_method == "Global") || isTRUE(adj_method == "Local")){
```

```
    if (is.null(G_Moran_I) && is.null(L_Moran_I[[1]]) && is.null(L_Moran_I[[2]]))
      stop("please insert valid global Moran's I and local Moran's I")}
  w <- c(x, y)
  # compute the superemum distance between tested ecdf/cdf
  z <- cumsum(x)/sum(x) - cumsum(y)/sum(y)
  STAT_VAL <- switch(alternative, two.sided = max(abs(z)),
                     greater = max(z), less = -min(z))
  PVAL <- NULL
  adj_MI <- G_Moran_I + c(1/(n.x - 1), 1/(n.y - 1))
  if (is.null(adj_method))
    PVAL <- 1 - .Call(stats:::C_pSmirnov2x, STAT_VAL, n.x, n.y)
  else if (adj_method == "Global"){
    # adjusted sample size by global Moran's I
    # 1-(1/(1-exp(-1.92369)))*(1343/1344)*(1-exp(-2.12373*0.2+0.20024*sqrt(0.2)))
    # G_n.x <- (1-(1/(1-exp(-1.92369)))*(1343/1344)*(1-exp(-2.12373*G_Moran_I[1]+0.20024*sqrt(G_Moran_I[1]))))*n.x
    # G_n.y <- (1-(1/(1-exp(-1.92369)))*(1343/1344)*(1-exp(-2.12373*G_Moran_I[2]+0.20024*sqrt(G_Moran_I[2]))))*n.y
    G_n.x <- ceiling((2/(1+exp(3.934*adj_MI[1] + 3.172*adj_MI[1]^3)))*n.x)
    G_n.y <- ceiling((2/(1+exp(3.934*adj_MI[2] + 3.172*adj_MI[2]^3)))*n.y)
    PVAL <- 1 - .Call(stats:::C_pSmirnov2x, STAT_VAL, G_n.x, G_n.y)}
  else if (adj_method == "Local"){
    # adjust sample sizes by local Moran's I
    adj_MI2 <- ifelse(adj_MI < 0, 0, adj_MI)
    L_n.x <- ceiling((1-(1/(1-exp(-1.92369)))*((n.x -1)/n.x)*(1-exp(-2.124*adj_MI2[1] + 0.2*sqrt(adj_MI2[1]))))*n.x)
    L_n.y <- ceiling((1-(1/(1-exp(-1.92369)))*((n.x -1)/n.x)*(1-exp(-2.124*adj_MI2[2] + 0.2*sqrt(adj_MI2[2]))))*n.y)
    PVAL <- 1 - .Call(stats:::C_pSmirnov2x, STAT_VAL, L_n.x, L_n.y)}
  else if (adj_method == "ICC"){
    # adjusted sample size by ICC
    ICC.xy <- 0.5
    ICC.n.x <- (1-ICC.xy)*n.x
    ICC.n.y <- (1-ICC.xy)*n.y
    PVAL <- 1 - .Call(stats:::C_pSmirnov2x, STAT_VAL, ICC.n.x, ICC.n.y)}

  output <- list('statistic' = STAT_VAL, "p.value" = PVAL)
  return(output)
}
# lisa_Moran(datapoints, weight_Matrix, scaled = T, na.rm = T)
```

C.2 Main Analysis

```
library(tidyverse)
library(readxl)
library(dplyr)
library(sqldf)
# read general patient info
patient_info <- read_excel("C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\PET Research Records-Sept
    2018.xlsx", guess_max =  7000)
# select patient participated in the study
study_Pat_info <- subset(patient_info, rprotocol_sub %in% c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80'))
# pat protocol info
pat_protocol_info <- study_Pat_info %>% select(pet_no, pat_id, pet_date, rprotocol, rprotocol_sub, pet_stressor)
# get the scan counts
pet_time <- tally(group_by(study_Pat_info, pat_id))
# table the scan counts
table(pet_time$n)
# select those who only took 1 PET scan
exclude_pat <- subset(pet_time, n == 1)
# select those taking two scans
interest_Pat_info <- subset(study_Pat_info, !(pat_id %in% exclude_pat$pat_id))
# sort by patient ID
interest_Pat_info_srt <- interest_Pat_info[order(interest_Pat_info$pat_id),]
# remove caffine
interest_Pat_final <- subset(interest_Pat_info_srt, !(pat_id == 'pat_08170') )
# table(interest_Pat_no_caf$rprotocol_sub)
# DD L-15 L+10 L+40 L+55 L+80
# 100   30  100   30   62   30
# create pet id and protocol
# get the baseline scan and mark it as count: 1
pat_protocol_info <- interest_Pat_final %>%
  group_by(pat_id) %>%
  mutate(ct = ifelse(pet_date < max(pet_date), 1, 2))
# table(pat_protocol_info$ct, pat_protocol_info$pat_id)
pet_protocol <- pat_protocol_info[,c(5, 2, 17, 22, 24, 202)]
pet_protocol <- pet_protocol %>% group_by(pat_id) %>% arrange(rprotocol_sub, pet_stressor)
pet_protocol <- pet_protocol %>% mutate(baseline = ifelse(rprotocol_sub == 'DD',
                                                          ifelse(ct == 2, 0, 1),
                                                          ifelse(pet_stressor == 'Dipyridamole', 1, 0)))
tt <- pet_protocol %>% select(pat_id, pet_no, rprotocol_sub, pet_stressor, baseline, pet_date)
# get the scan number
scan_num <- pet_protocol$pet_no

# create matrix for Moran's I
pet_scan_moran_matrix <- as.data.frame(matrix(data = NA, nrow = 352, ncol = 6))
colnames(pet_scan_moran_matrix) <- c("Pet_ID", "value0_M", "value1_M", "cfr_M", "capacity_M", "AVG_M")
```

```
44
45 # patient scan location
46 pat_loc <- c("C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Scan_Pixels\\Pooled\\")
47
48 for (i in 1:length(scan_num)){
49     # read patients imaging scan
50     infile <- paste(pat_loc,
51                     scan_num[i], '.csv', sep="" )
52     Pat_data <- read.csv(file = infile, header = T)
53
54     # subset data into before(pat_data_B) and after(pat_data_A) treatment
55     pat_data_B <- subset(Pat_data, state == 0)
56     pat_data_A <- subset(Pat_data, state == 1)
57
58     # merge patients data into coordinates
59     pat_coor_B <- merge(pat_data_B, scan_matrix, by = c('row', 'radial'))
60     pat_coor_A <- merge(pat_data_A, scan_matrix, by = c('row', 'radial'))
61
62     # sorting data
63     attach(pat_coor_B)
64     pat_coor_B_srt <- pat_coor_B[order(row, radial),]
65     detach(pat_coor_B)
66     attach(pat_coor_A)
67     pat_coor_A_srt <- pat_coor_A[order(row, radial),]
68     detach(pat_coor_A)
69
70     # calculate Moran's I for patients imaging data
71     # weight matrix is calculated by the inverse distance matrix of our spherical distance
72     # correlating strength could be adjusted by different p.
73     # before treatment
74
75     M_cfr_1 <- lisa_Moran(pat_coor_B_srt$cfr, weight_Matrix, scaled = T, na.rm = T)[2][[1]]
76
77     M_value_1 <- lisa_Moran(pat_coor_B_srt$value, weight_Matrix, scaled = T, na.rm = T)[2][[1]]
78     # capacity is a character variable with normal and minimal, translate it into numerical form
79     # table(pat_coor_B_srt$capacity)
80     M_Capacity_1 <- lisa_Moran(as.numeric(pat_data_A$capacity), weight_Matrix, scaled = T, na.rm = T)[2][[1]]
81
82     # after treatment
83     M_value_2 <- lisa_Moran(pat_coor_A_srt$value, weight_Matrix, scaled = T, na.rm = T)[2][[1]]
84
85     # average M of CFR, Value 0 & 1
86     avg_M <- mean(c(M_value_1, M_value_2, M_cfr_1), na.rm = T)
87     pet_scan_moran_matrix[i,] <- c(scan_num[i], M_value_1, M_value_2, M_cfr_1, M_Capacity_1, avg_M)
88
89
90 }
91
92 # save the Morans' I matrix
93 saveRDS(pet_scan_moran_matrix, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pet_m.rds')
94 pet_scan_moran_matrix <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pet_m.rds')
95 # average Moran's I for ks test
96 pet_scan_moran_matrix_protocol <- sqldf(
97   "SELECT T.rprotocol_sub, T.pet_stressor, T.pat_id, T.baseline, R.*
98       FROM pet_scan_moran_matrix AS R
99       LEFT JOIN pet_protocol AS T
100      ON R.Pet_ID = T.pet_no
101  "
102 )
103 # save the Morans' I matrix with protocol and stressor used
104 saveRDS(pet_scan_moran_matrix, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pet_m.rds')
105 saveRDS(pet_scan_moran_matrix_protocol, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pet_
      scan_moran_matrix_protocol.rds')
106 pet_scan_moran_matrix_protocol <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pet_
      scan_moran_matrix_protocol.rds')
107 pet_scan_moran_matrix <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pet_m.rds')
108
109 # creae average
110 avg_M_ks <- pet_scan_moran_matrix_protocol %>%
111               group_by(rprotocol_sub, baseline) %>%
112               summarise(value0_M = mean(value0_M),
113                         value1_M = mean(value1_M),
114                         cfr_M = mean(cfr_M),
115                         capacity_M = mean(capacity_M, na.rm = T))
116
117 # after having the Moran's I, deal with the average frequency pet scan
118 p <- 1
119 protocol_pet_list <- vector("list", 6)
120 # protocol: 'DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80'
121 for (i in c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')){
122   for (k in c(0, 1)){
123     pet_value0 <- as.data.frame(matrix(data = NA, nrow = 1344, ncol = 1))
124     pet_value1 <- as.data.frame(matrix(data = NA, nrow = 1344, ncol = 1))
125     pet_cfr <- as.data.frame(matrix(data = NA, nrow = 1344, ncol = 1))
126     pet_capacity <- as.data.frame(matrix(data = NA, nrow = 1344, ncol = 1))
127     q <- 2
128     for (j in subset(pet_protocol, rprotocol_sub == i & baseline == k)$pet_no){
129       infile <- paste(pat_loc,
```

```
                    j, '.csv', sep="" )
      Pat_data <- read.csv(file = infile, header = T)
      # subset data into before(pat_data_B) and after(pat_data_A) treatment
      pat_data_B <- subset(Pat_data, state == 0)
      pat_data_A <- subset(Pat_data, state == 1)

      pet_value0 <- cbind(pet_value0, pat_data_B$value)
      colnames(pet_value0)[q] <- j
      pet_value1 <- cbind(pet_value1, pat_data_A$value)
      colnames(pet_value1)[q] <- j
      pet_cfr <- cbind(pet_cfr, pat_data_B$cfr)
      colnames(pet_cfr)[q] <- j
      pat_data_B <- pat_data_B %>% mutate(capacity1 = ifelse(capacity == 'severe', 1,
                                                             ifelse(capacity == 'moderate', 2,
                                                                    ifelse(capacity == 'mild', 3,
                                                                           ifelse(capacity == 'minimal', 4, 5)))))
      pet_capacity <- cbind(pet_capacity, pat_data_B$capacity1)
      colnames(pet_capacity)[q] <- j
      q <- q + 1
    }
    pet_value0 <- cbind(pet_value0, rowMeans(pet_value0[,-1]))
    colnames(pet_value0)[q] <- 'avg_value0'
    pet_value1 <- cbind(pet_value1, rowMeans(pet_value1[,-1]))
    colnames(pet_value1)[q] <- 'avg_value1'
    pet_cfr <- cbind(pet_cfr, rowMeans(pet_cfr[,-1]))
    colnames(pet_cfr)[q] <- 'avg_cfr'
    pet_capacity <- cbind(pet_capacity, rowMeans(pet_capacity[,-1]))
    colnames(pet_capacity)[q] <- 'avg_capacity'
    protocol_pet_list[[p]][[(k+1)]] <- list(pet_value0[,-1], pet_value1[,-1], pet_cfr[,-1], pet_capacity[,-1])
  }

  p <- p + 1
}
saveRDS(protocol_pet_list, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\protocol_pet_list.
    rds')
protocol_pet_list <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\protocol_pet_list.
    rds')

# get the PET capacity info
p <- 1
protocol_pet_capacity_list <- vector("list", 6)
for (i in c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')){

  for (k in c(0, 1)){
    protocol_pet_capacity <- as.data.frame(matrix(data = as.factor(c(1, 2, 3, 4, 5)), nrow = 5, ncol = 1))
    colnames(protocol_pet_capacity) <- c('capacity')
    protocol_pet_capacity_temp <- protocol_pet_list[[p]][[(k+1)]][[4]][,1:(length(protocol_pet_list[[p]][[(k+1)]][[4]])-1)]
    q <- 2
    for (j in colnames(protocol_pet_capacity_temp)){

      pet_capacity_frq <- as.data.frame(table(protocol_pet_capacity_temp %>% select(j)))
      colnames(pet_capacity_frq) <- c('capacity', j)
      pet_capacity_frq[,1] <- as.character(pet_capacity_frq[,1])
      protocol_pet_capacity <- left_join(x = protocol_pet_capacity, y = pet_capacity_frq)
      q <- q + 1

    }
    pet_capacity_avg <- cbind(protocol_pet_capacity, rowSums(protocol_pet_capacity[,-1], na.rm = T)/length(protocol_pet_
        capacity_temp))
    colnames(pet_capacity_avg)[q] <- 'avg_capacity'

    protocol_pet_capacity_list[[p]][[(k+1)]] <- list(pet_capacity_avg)
  }

  p <- p + 1
}
saveRDS(protocol_pet_capacity_list, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\protocol_
    pet_capacity_list.rds')
protocol_pet_capacity_list <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\protocol_
    pet_capacity_list.rds')

# go ahead to create ks test
# value 0
for (i in c(1:2)){
  pet_value0_ks_t <- data.frame(matrix(data = c(unlist(protocol_pet_list[[1]][[i]][[1]][length(protocol_pet_list[[1]][[i
      ]][[1]])]),
                                                unlist(protocol_pet_list[[2]][[i]][[1]][length(protocol_pet_list[[2]][[i
                                                    ]][[1]])]),
                                                unlist(protocol_pet_list[[3]][[i]][[1]][length(protocol_pet_list[[3]][[i
                                                    ]][[1]])]),
                                                unlist(protocol_pet_list[[4]][[i]][[1]][length(protocol_pet_list[[4]][[i
                                                    ]][[1]])]),
                                                unlist(protocol_pet_list[[5]][[i]][[1]][length(protocol_pet_list[[5]][[i
                                                    ]][[1]])]),
                                                unlist(protocol_pet_list[[6]][[i]][[1]][length(protocol_pet_list[[6]][[i
                                                    ]][[1]])])),
                                       nrow = 1344, ncol = 6))
  pet_value0_ks_t <- pet_value0_ks_t %>% mutate(baseline = i - 1)
```

```
  colnames(pet_value0_ks_t)[1:(ncol(pet_value0_ks_t)-1)] <- c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')
  if (i == 1){
    pet_value0_ks <- pet_value0_ks_t
  }else{
    pet_value0_ks <- rbind(pet_value0_ks, pet_value0_ks_t)
  }
}
saveRDS(pet_value0_ks, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pet_value0_ks.rds')
pet_value0_ks <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pet_value0_ks.rds')
# value 1
for (i in c(1:2)){
  pet_value1_ks_t <- data.frame(matrix(data = c(unlist(protocol_pet_list[[1]][[i]][[2]][length(protocol_pet_list[[1]][[i
      ]][[1]])]),
                                       unlist(protocol_pet_list[[2]][[i]][[2]][length(protocol_pet_list[[2]][[i]][[1]])]),
                                       unlist(protocol_pet_list[[3]][[i]][[2]][length(protocol_pet_list[[3]][[i]][[1]])]),
                                       unlist(protocol_pet_list[[4]][[i]][[2]][length(protocol_pet_list[[4]][[i]][[1]])]),
                                       unlist(protocol_pet_list[[5]][[i]][[2]][length(protocol_pet_list[[5]][[i]][[1]])]),
                                       unlist(protocol_pet_list[[6]][[i]][[2]][length(protocol_pet_list[[6]][[i]][[1]])])),
                              nrow = 1344, ncol = 6))
  pet_value1_ks_t <- pet_value1_ks_t %>% mutate(baseline = i - 1)
  colnames(pet_value1_ks_t)[1:(ncol(pet_value1_ks_t)-1)] <- c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')
  if (i == 1){
    pet_value1_ks <- pet_value1_ks_t
  }else{
    pet_value1_ks <- rbind(pet_value1_ks, pet_value1_ks_t)
  }
}
saveRDS(pet_value1_ks, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pet_value1_ks.rds')
pet_value1_ks <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pet_value1_ks.rds')
# cfr
for (i in c(1:2)){
  pet_cfr_ks_t <- data.frame(matrix(data = c(unlist(protocol_pet_list[[1]][[i]][[3]][length(protocol_pet_list[[1]][[i]][[1]])
      ]),
                                       unlist(protocol_pet_list[[2]][[i]][[3]][length(protocol_pet_list[[2]][[i]][[1]])]),
                                       unlist(protocol_pet_list[[3]][[i]][[3]][length(protocol_pet_list[[3]][[i]][[1]])]),
                                       unlist(protocol_pet_list[[4]][[i]][[3]][length(protocol_pet_list[[4]][[i]][[1]])]),
                                       unlist(protocol_pet_list[[5]][[i]][[3]][length(protocol_pet_list[[5]][[i]][[1]])]),
                                       unlist(protocol_pet_list[[6]][[i]][[3]][length(protocol_pet_list[[6]][[i]][[1]])])),
                              nrow = 1344, ncol = 6))
  pet_cfr_ks_t <- pet_cfr_ks_t %>% mutate(baseline = i - 1)
  colnames(pet_cfr_ks_t)[1:(ncol(pet_cfr_ks_t)-1)] <- c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')
  if (i == 1){
    pet_cfr_ks <- pet_cfr_ks_t
  }else{
    pet_cfr_ks <- rbind(pet_cfr_ks, pet_cfr_ks_t)
  }
}
saveRDS(pet_cfr_ks, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pet_cfr_ks.rds')
pet_cfr_ks <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pet_cfr_ks.rds')

# capacity in Dr. Lai's
for (i in c(1:2)){
  pet_capacity_ks_t <- data.frame(matrix(data = c(unlist(protocol_pet_capacity_list[[1]][[i]][[1]][length(protocol_pet_
      capacity_list[[1]][[i]][[1]])]),
                                       unlist(protocol_pet_capacity_list[[2]][[i]][[1]][length(protocol_pet_capacity_list[[2]][[i
                                           ]][[1]])]),
                                       unlist(protocol_pet_capacity_list[[3]][[i]][[1]][length(protocol_pet_capacity_list[[3]][[i
                                           ]][[1]])]),
                                       unlist(protocol_pet_capacity_list[[4]][[i]][[1]][length(protocol_pet_capacity_list[[4]][[i
                                           ]][[1]])]),
                                       unlist(protocol_pet_capacity_list[[5]][[i]][[1]][length(protocol_pet_capacity_list[[5]][[i
                                           ]][[1]])]),
                                       unlist(protocol_pet_capacity_list[[6]][[i]][[1]][length(protocol_pet_capacity_list[[6]][[i
                                           ]][[1]])])),
                              nrow = 5, ncol = 6))
  pet_capacity_ks_t <- pet_capacity_ks_t %>% mutate(baseline = i - 1)
  colnames(pet_capacity_ks_t)[1:(ncol(pet_capacity_ks_t)-1)] <- c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')
  if (i == 1){
    pet_capacity_ks <- pet_capacity_ks_t
  }else{
    pet_capacity_ks <- rbind(pet_capacity_ks, pet_capacity_ks_t)
  }
}
saveRDS(pet_capacity_ks, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pet_capacity_ks.rds')
pet_capacity_ks <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pet_capacity_ks.rds'
    )


for (i in 1:6){
    test1 <- c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')[i]
    KS_P <- data.frame(MI.adj.ks.test.discret(unlist(subset(pet_capacity_ks %>% select(test1, baseline), baseline == 0)[1]),
                                              unlist(subset(pet_capacity_ks %>% select(test1, baseline), baseline == 1)[1]),
                                              G_Moran_I = c(mean(subset(avg_M_ks, rprotocol_sub == test1)$value0_M[1],
                                                                 subset(avg_M_ks, rprotocol_sub == test1)$value1_M[1],
                                                                 subset(avg_M_ks, rprotocol_sub == test1)$cfr_M[1]),
                                                            mean(subset(avg_M_ks, rprotocol_sub == test1)$value0_M[2],
                                                                 subset(avg_M_ks, rprotocol_sub == test1)$value1_M[2],
                                                                 subset(avg_M_ks, rprotocol_sub == test1)$cfr_M[2])),
```

```
                                          adj_method = 'Global'))

    KS_P <- KS_P %>% mutate(original_ks = MI.adj.ks.test.discret(unlist(subset(pet_capacity_ks %>% select(test1, baseline),
        baseline == 0)[1]),
                                                                 unlist(subset(pet_capacity_ks %>% select(test1, baseline),
                                                                      baseline == 1)[1]),
                                                                 G_Moran_I = c(0, 0),
                                                                 adj_method = 'Global')$p.value)

    KS_P <- KS_P %>% mutate(ICC_ks = MI.adj.ks.test.discret(unlist(subset(pet_capacity_ks %>% select(test1, baseline),
        baseline == 0)[1]),
                                                                 unlist(subset(pet_capacity_ks %>% select(test1, baseline),
                                                                      baseline == 1)[1]),
                                                                 adj_method = 'ICC')$p.value)

    KS_P <- KS_P %>% mutate(test_grp = paste(test1))

    if (i == 1){
      pooled_KS_P_ap1 <- KS_P
    }else{
      pooled_KS_P_ap1 <- rbind(pooled_KS_P_ap1, KS_P)
    }
}
pooled_KS_P_ap1 <- pooled_KS_P_ap1 %>% mutate(sig = ifelse(p.value < 0.05, 1, 0))
pooled_KS_P_ap1 <- pooled_KS_P_ap1[,c(5, 1:4, 6)]
saveRDS(pooled_KS_P_ap1, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pooled_KS_P_ap1.rds')
pooled_KS_P_ap1 <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pooled_KS_P_ap1.rds'
    )
pooled_KS_P_ap1[,2] <- round(pooled_KS_P_ap1[,2], digits = 2)
pooled_KS_P_ap1[,3:5] <- round(pooled_KS_P_ap1[,3:5], digits = 4)

# KS on CFR
for (i in 1:6){
  test1 <- c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')[i]
  KS_cfr_P <- data.frame(MI.adj.ks.test(unlist(subset(pet_cfr_ks %>% select(test1, baseline), baseline == 0)[1]),
                                        unlist(subset(pet_cfr_ks %>% select(test1, baseline), baseline == 1)[1]),
                                        G_Moran_I = c(subset(avg_M_ks, rprotocol_sub == test1)$cfr_M),
                                        adj_method = 'Global'))
  KS_cfr_P <- KS_cfr_P %>% mutate(test_grp = paste(test1))
  if (i == 1){
    pooled_KS_cfr_P_ap1 <- KS_cfr_P
  }else{
    pooled_KS_cfr_P_ap1 <- rbind(pooled_KS_cfr_P_ap1, KS_cfr_P)
  }
}
pooled_KS_cfr_P_ap1 <- pooled_KS_cfr_P_ap1 %>% mutate(sig = ifelse(p.value < 0.05, 1, 0))
saveRDS(pooled_KS_cfr_P_ap1, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pooled_KS_cfr_P_
    ap1.rds')
pooled_KS_cfr_P_ap1 <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pooled_KS_cfr_P_
    ap1.rds')

for (i in 1:6){
  test1 <- c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')[i]
  KS_value0_P <- data.frame(MI.adj.ks.test(unlist(subset(pet_value0_ks %>% select(test1, baseline), baseline == 0)[1]),
                                        unlist(subset(pet_value0_ks %>% select(test1, baseline), baseline == 1)[1]),
                                        G_Moran_I = c(subset(avg_M_ks, rprotocol_sub == test1)$value0_M),
                                        adj_method = 'Global'))

  KS_value0_P <- KS_value0_P %>% mutate(original_ks = ks.test(unlist(subset(pet_value0_ks %>% select(test1, baseline),
      baseline == 0)[1]),
                                                              unlist(subset(pet_value0_ks %>% select(test1, baseline))))$p.
                                                                   value)

  KS_value0_P <- KS_value0_P %>% mutate(ICC_ks = MI.adj.ks.test(unlist(subset(pet_value0_ks %>% select(test1, baseline),
      baseline == 0)[1]),
                                                                unlist(subset(pet_value0_ks %>% select(test1, baseline),
                                                                     baseline == 1)[1]),
                                                                adj_method = 'ICC')$p.value)

  KS_value0_P <- KS_value0_P %>% mutate(test_grp = paste(test1))
  if (i == 1){
    pooled_KS_value0_P_ap1 <- KS_value0_P
  }else{
    pooled_KS_value0_P_ap1 <- rbind(pooled_KS_value0_P_ap1, KS_value0_P)
  }
}
pooled_KS_value0_P_ap1 <- pooled_KS_value0_P_ap1 %>% mutate(sig = ifelse(p.value < 0.05, 1, 0))
pooled_KS_value0_P_ap1 <- pooled_KS_value0_P_ap1[,c(5, 1:4, 6)]
saveRDS(pooled_KS_value0_P_ap1, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pooled_KS_
    value0_P_ap1.rds')
pooled_KS_value0_P_ap1 <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pooled_KS_
    value0_P_ap1.rds')

for (i in 1:6){
  test1 <- c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')[i]
  KS_value1_P <- data.frame(MI.adj.ks.test(unlist(subset(pet_value1_ks %>% select(test1, baseline), baseline == 0)[1]),
                                        unlist(subset(pet_value1_ks %>% select(test1, baseline), baseline == 1)[1]),
                                        G_Moran_I = c(subset(avg_M_ks, rprotocol_sub == test1)$value1_M),
```

```
                                        adj_method = 'Global'))
  KS_value1_P <- KS_value1_P %>% mutate(test_grp = paste(test1))
  if (i == 1){
    pooled_KS_value1_P_ap1 <- KS_value1_P
  }else{
    pooled_KS_value1_P_ap1 <- rbind(pooled_KS_value1_P_ap1, KS_value1_P)
  }
}
pooled_KS_value1_P_ap1 <- pooled_KS_value1_P_ap1 %>% mutate(sig = ifelse(p.value < 0.05, 1, 0))
saveRDS(pooled_KS_value1_P_ap1, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pooled_KS_
    value1_P_ap1.rds')
pooled_KS_value1_P_ap1 <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pooled_KS_
    value1_P_ap1.rds')
# statistic       p.value test_grp sig
# 1 0.1979167  4.457124e-03       DD   1
# 2 0.8214286  3.330669e-16     L-15   1
# 3 0.7016369 -6.661338e-16     L+10   1
# 4 0.4724702  6.661338e-16     L+40   1
# 5 0.3824405  1.604372e-10     L+55   1
# 6 0.4538690  3.641532e-14     L+80   1
##############################################################################################################
##############################################################################################################
############################################## an alternative approach ########################################
##############################################################################################################
##############################################################################################################
# begin ks test, the moran's I average is avg_M_ks
# My thought: instead of taking the average of capacity directly.
#             take average of value and cfr to calculate the average capacity
# now the problem is transferred to more calculation
j <- 1
for (i in c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')){
  pooled_pet_ks_temp <- data.frame(cbind(pet_value0_ks[,j], pet_value1_ks[,j], pet_cfr_ks[,j]))
  colnames(pooled_pet_ks_temp) <- c('avg_value0', 'avg_value1', 'avg_cfr')
  pooled_pet_ks_temp <- pooled_pet_ks_temp %>% mutate(sub_protocol = i)
  if (j == 1){
    pooled_pet_ks <- pooled_pet_ks_temp
  }else{
    pooled_pet_ks <- rbind(pooled_pet_ks, pooled_pet_ks_temp)
  }
  j <- j + 1
}
# manipulate data
pooled_pet_ks_md <- pooled_pet_ks %>%
  mutate(pet_avg_cap = ifelse(avg_value1 >= 2.17 | avg_cfr >= 2.9, 5,
                              ifelse(avg_value1 >= 1.82 | avg_cfr >= 2.38, 4,
                                     ifelse(avg_value1 >= 1.09 | avg_cfr >= 1.6, 3,
                                            ifelse(avg_value1 >= 0.83 | avg_cfr >= 1.27, 2, 1)))))
# do the ks test, an alternative approach
# Note this approach is the average of capacity defined different than Dr. Lai's version
for (i in 1:5){
  for(j in (i+1):6){
    test1 <- c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')[i]
    test2 <- c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')[j]
    KS_P <- data.frame(MI.adj.ks.test(subset(pooled_pet_ks_md, sub_protocol == test1)$pet_avg_cap,
                           subset(pooled_pet_ks_md, sub_protocol == test2)$pet_avg_cap,
                           G_Moran_I = c(subset(avg_M_ks, rprotocol_sub == test1)$cfr_M,
                                         subset(avg_M_ks, rprotocol_sub == test2)$cfr_M),
                           adj_method = 'Global'))
    KS_P <- KS_P %>% mutate(test_grp = paste(test1, 'vs', test2))
    if (i == 1 & j == 2){
      pooled_KS_P_ap2 <- KS_P
    }else{
      pooled_KS_P_ap2 <- rbind(pooled_KS_P_ap2, KS_P)
    }

  }
}
saveRDS(pooled_KS_P_ap2, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pooled_KS_P_ap2.rds')
pooled_KS_P_ap2 <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pooled_KS_P_ap2.rds'
    )

# descriptive
for (i in 1:6){
  test1 <- c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')[i]
  temp_data <- unlist(subset(pet_value0_ks %>% select(test1, baseline), baseline == 0)[1])
  temp_data2 <- unlist(subset(pet_value0_ks %>% select(test1, baseline), baseline == 1)[1])

  print(round(c(mean(temp_data)), 2))
  #print(round(c(mean(temp_data) - mean(temp_data2)), 2))
}
for (i in 1:6){
  test1 <- c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')[i]
  temp_data <- unlist(subset(pet_value1_ks %>% select(test1, baseline), baseline == 0)[1])
  temp_data2 <- unlist(subset(pet_value1_ks %>% select(test1, baseline), baseline == 1)[1])

  #print(round(c(mean(temp_data)), 2))
  print(round(c(mean(temp_data) - mean(temp_data2)), 2))
}
```

```r
for (i in 1:6){
  test1 <- c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')[i]
  temp_data <- unlist(subset(pet_cfr_ks %>% select(test1, baseline), baseline == 0)[1])
  temp_data2 <- unlist(subset(pet_cfr_ks %>% select(test1, baseline), baseline == 1)[1])

  #print(round(c(mean(temp_data)), 2))
  print(round(c(mean(temp_data) - mean(temp_data2)), 2))
}
# few plots
# resting flow
for (i in 1:6){
  test1 <- c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')[i]
  temp_data <- ecdf(unlist(subset(pet_value0_ks %>% select(test1, baseline), baseline == 1)[1]))
  if (i == 1){
    plot(temp_data, xlim = c(0.4, 1.5))
  }else{
    plot(temp_data, verticals=TRUE, do.points=FALSE, add=TRUE, col=i)
  }
}

for (i in 1:6){
  test1 <- c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')[i]
  temp_data <- ecdf(unlist(subset(pet_cfr_ks %>% select(test1, baseline), baseline == 0)[1]))
  if (i == 1){
    plot(temp_data, xlim = c(2, 3.5))
  }else{
    plot(temp_data, verticals=TRUE, do.points=FALSE, add=TRUE, col=i)
  }
}

library(knitr)
library(kableExtra)
library(dplyr)
# descriptive tables
# interested population dataset: pat_protocol_info

# table 1, baseline characteristics
# interested variables: (note: checked variable with * sign)
#   clinical characteristics;
#     age*, sex*, bmi&, CAD: prior (bypass surgery(CABG)*, percutaneous intervention(hx_PCI)*,
#     myocardial infarction(hx_MI_recent?)*)
#     Dyslipidemia*, diabetes mellitus*, hypertension(hx_htn)*, current smoking(pet_stressor?)*
#   Current medications;
#     Statins*, ACEI or ARB*, antiplatelet use*, beta blocker*, calcium channel blockers(med_ccb)*, diuretics*, nitrate*
#   Extened clinical characteristics:
#     total cholesterol*, LDL*, HDL*, resting (sbp*, dbp*, heart rate, pressure-rate product)
#     stress (sbp, dbp, heart rate, pressure-rate product), rest and stress homogeneity index
tb1_data <- pat_protocol_info %>% select(pet_no, pat_id, pet_date, rprotocol, rprotocol_sub, pet_stressor,
                                         age, male, BMI, rest_sbp, rest_dbp, rest_hr, stress_sbp, pet_cotinine, pet_nicotine,
                                         stress_dbp, stress_hr, Cholest, LDL, HDL, med_statin, med_ACEIorARB, med_nitrate,
                                         med_antiplatelet, med_betablocker, med_diuretic, med_ccb, hx_dyslipidemia,
                                         hx_smoking, hx_diabetes, hx_MI_recent, hx_MI_distant, hx_PCI, hx_CABG, hx_htn,
                                         hx_prior_cath, pet_angina)
# with baseline indication variable added
tb1_data_ba <- sqldf(
     "SELECT T.baseline, R.*
      FROM tb1_data AS R
      LEFT JOIN pet_protocol AS T
      ON R.pet_no = T.pet_no
  ")
# first part
desc_table_pt1.1.1 <- tb1_data_ba %>%
  summarise(age_avg = mean(age), age_sd = sd(age),
            BMI_avg = mean(BMI), BMI_sd = sd(BMI))

desc_table_pt1.1.2 <- tb1_data_ba %>% group_by(baseline) %>%
  summarise(rest_sbp_avg = mean(rest_sbp), rest_sbp_sd = sd(rest_sbp),
            rest_dbp_avg = mean(rest_dbp), rest_dbp_sd = sd(rest_dbp),
            rest_hr_avg = mean(rest_hr), rest_hr_sd = sd(rest_hr),
            stress_sbp_avg = mean(stress_sbp), stress_sbp_sd = sd(stress_sbp),
            stress_dbp_avg = mean(stress_dbp), stress_dbp_sd = sd(stress_dbp),
            stress_hr_avg = mean(stress_hr), stress_hr_sd = sd(stress_hr),
            Cholest_avg = mean( as.numeric(Cholest), na.rm = T), Cholest_sd = sd( as.numeric(Cholest), na.rm = T),
            LDL_avg = mean( as.numeric(LDL), na.rm = T), LDL_sd = sd( as.numeric(LDL), na.rm = T),
            HDL_avg = mean( as.numeric(HDL), na.rm = T), HDL_sd = sd( as.numeric(HDL), na.rm = T))

desc_table_pt1.2.1 <- tb1_data_ba %>% group_by(rprotocol_sub) %>%
  summarise(age_avg = mean(age), age_sd = sd(age),
            BMI_avg = mean(BMI), BMI_sd = sd(BMI))

desc_table_pt1.2.2 <- tb1_data_ba %>% group_by(rprotocol_sub, baseline) %>%
  summarise(rest_sbp_avg = mean(rest_sbp), rest_sbp_sd = sd(rest_sbp),
            rest_dbp_avg = mean(rest_dbp), rest_dbp_sd = sd(rest_dbp),
            rest_hr_avg = mean(rest_hr), rest_hr_sd = sd(rest_hr),
            stress_sbp_avg = mean(stress_sbp), stress_sbp_sd = sd(stress_sbp),
            stress_dbp_avg = mean(stress_dbp), stress_dbp_sd = sd(stress_dbp),
            stress_hr_avg = mean(stress_hr), stress_hr_sd = sd(stress_hr),
            Cholest_avg = mean( as.numeric(Cholest), na.rm = T), Cholest_sd = sd( as.numeric(Cholest), na.rm = T),
```

```
            LDL_avg = mean( as.numeric(LDL), na.rm = T), LDL_sd = sd( as.numeric(LDL), na.rm = T),
            HDL_avg = mean( as.numeric(HDL), na.rm = T), HDL_sd = sd( as.numeric(HDL), na.rm = T)) %>%
  arrange(baseline, rprotocol_sub)

saveRDS(desc_table_pt1.1.1, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\desc_table_pt1_1_1.rds')
saveRDS(desc_table_pt1.1.2, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\desc_table_pt1_1_2.rds')
saveRDS(desc_table_pt1.2.1, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\desc_table_pt1_2_1.rds')
saveRDS(desc_table_pt1.2.2, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\desc_table_pt1_2_2.rds')
desc_table_pt1.1.1 <- round(readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\desc_table_pt1_1_1.rds')
                            , digits = 0)
desc_table_pt1.1.2 <- round(readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\desc_table_pt1_1_2.rds')
                            , digits = 0)
desc_table_pt1.2.1 <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\desc_table_pt1_2_1.rds')

desc_table_pt1.2.2 <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\desc_table_pt1_2_2.rds')

desc_table_pt1.2.1[,-1] <- round(readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\desc_table_pt1_2_1.rds')[,-1]
                                 , digits = 0)
desc_table_pt1.2.2[,-1] <- round(readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\desc_table_pt1_2_2.rds')[,-1]
                                 , digits = 0)
# desc_table_pt1H_l: descriptive table first half(1H) latex file
desc_table_pt1_1H_l <- data.frame(matrix(nrow = 2, ncol = 7))
colnames(desc_table_pt1_1H_l) <- c('population', unlist(desc_table_pt1.2.1 %>% distinct(rprotocol_sub)))
for (i in 1:7){
  for (j in 1:2){
    if (i == 1){
      desc_table_pt1_1H_l[j, i] <- paste(desc_table_pt1.1.1[i, (j)*2-1], '+', desc_table_pt1.1.1[i, ((j)*2)], sep = '')
    }else{
      desc_table_pt1_1H_l[j, i] <- paste(desc_table_pt1.2.1[i-1, (j)*2], '+', desc_table_pt1.2.1[i-1, ((j)*2 + 1)], sep = '')
    }
  }
}

# reordered table variables
desc_table_pt1.1.2_srt <- desc_table_pt1.1.2[,c(1, 14:19, 2:13)]
desc_table_pt1.2.2_srt <- desc_table_pt1.2.2[,c(1, 2, 15:20, 3:14)]
# desc_table_pt2H_l: descriptive table second half(1H) latex file
desc_table_pt2H_l <- data.frame(matrix(nrow = 18, ncol = 7))
colnames(desc_table_pt2H_l) <- c('population', unlist(desc_table_pt1.2.1 %>% distinct(rprotocol_sub)))
for (i in 1:7){
  for (j in 1:18){
    if (j <= 9){
      if (i == 1){
        desc_table_pt2H_l_temp <- subset(desc_table_pt1.1.2_srt, baseline == 1)
        desc_table_pt2H_l[j, i] <- paste(desc_table_pt2H_l_temp[i, (j)*2], '+', desc_table_pt2H_l_temp[i, ((j)*2+1)], sep = '')
      }else{
        desc_table_pt2H_l_temp <- subset(desc_table_pt1.2.2_srt, baseline == 1)
        desc_table_pt2H_l[j, i] <- paste(desc_table_pt2H_l_temp[i-1, (j)*2 + 1], '+', desc_table_pt2H_l_temp[i-1, ((j)*2 + 2)], sep = '')
      }
    }else{
      k <- j - 9
      if (i == 1){
        desc_table_pt2H_l_temp <- subset(desc_table_pt1.1.2_srt, baseline == 0)
        desc_table_pt2H_l[j, i] <- paste(desc_table_pt2H_l_temp[i, (k)*2], '+', desc_table_pt2H_l_temp[i, ((k)*2+1)], sep = '')
      }else{
        desc_table_pt2H_l_temp <- subset(desc_table_pt1.2.2_srt, baseline == 0)
        desc_table_pt2H_l[j, i] <- paste(desc_table_pt2H_l_temp[i-1, (k)*2+1], '+', desc_table_pt2H_l_temp[i-1, ((k)*2 + 2)], sep = '')
      }
    }
  }
}

desc_table_1 <- rbind(desc_table_pt1_1H_l, desc_table_pt2H_l)
desc_table_1$cha <- c('Age', 'BMI',
                      'Cholesteral', 'LDL', 'HDL',
                      'Rest Systolic blood pressure', 'Rest Diastolic blood pressure', 'Rest Heart Rate',
                      'Stress Systolic blood pressure', 'Stress Diastolic blood pressure', 'Stress Heart Rate',
                      'Cholesteral', 'LDL', 'HDL',
                      'Rest Systolic blood pressure', 'Rest Diastolic blood pressure', 'Rest Heart Rate',
                      'Stress Systolic blood pressure', 'Stress Diastolic blood pressure', 'Stress Heart Rate')
desc_table_1 <- desc_table_1[, c(ncol(desc_table_1), 1:(ncol(desc_table_1)-1))]
desc_table_1_kable <- kable(desc_table_1, "latex", booktabs = T, align = "c",
                            caption = "Type I Error for Two sample tests of Spatial Normal Distributed Samples",
                            digits = 0, longtable = T)
```

```
608
609 cat(desc_table_1_kable, file = paste('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\',
        "desc_table_1_kable.txt", sep='')
610     , sep = "n", append = T)
611
612
613 # list categorical variable
614 desc_table_pt1.3 <- tb1_data_ba %>% distinct(rprotocol_sub)
615
616 for (i in c('hx_htn', 'hx_dyslipidemia', 'hx_diabetes', 'hx_prior_cath', 'hx_PCI', 'hx_CABG',
617             'med_statin', 'med_ACEIorARB', 'med_antiplatelet', 'med_betablocker', 'med_diuretic', 'med_ccb', 'med_nitrate',
618             'hx_smoking', 'hx_MI_recent')){
619   # set intersted table
620   if ( i == 'hx_smoking'){
621     desc_table_pt_int <- tb1_data_ba %>% mutate(smk = ifelse(as.numeric(eval(as.symbol(i))) > 0 , 1, 0)) %>%
622       group_by(rprotocol_sub, smk) %>% summarise(n = ceiling(n()/2)) %>%
623       ungroup %>% group_by(rprotocol_sub) %>% mutate(total = sum(n), rel.prob = n/total)
624   } else if ( i == 'hx_MI_recent'){
625     desc_table_pt_int <- tb1_data_ba %>% mutate(MI = ifelse(as.numeric(hx_MI_recent) > 0 | hx_MI_distant >0 , 1, 0)) %>%
626       group_by(rprotocol_sub, MI) %>% summarise(n = ceiling(n()/2)) %>%
627       ungroup %>% group_by(rprotocol_sub) %>% mutate(total = sum(n), rel.prob = n/total)
628   }else{
629     desc_table_pt_int <- tb1_data_ba %>% group_by(rprotocol_sub, eval(as.symbol(i))) %>% summarise(n = ceiling(n()/2)) %>%
630       ungroup %>% group_by(rprotocol_sub) %>% mutate(total = sum(n), rel.prob = n/(total))
631   }
632
633   colnames(desc_table_pt_int)[2] <- i
634   desc_table_pt_int2 <- subset(desc_table_pt_int, eval(as.symbol(i)) == 1)
635   desc_table_pt_int_temp <- desc_table_pt_int2[,c(1, 3, 5)]
636   colnames(desc_table_pt_int_temp)[2:3] <- c(paste(i, '.n'), paste(i, '.pct'))
637
638   desc_table_pt1.3 <- merge(desc_table_pt1.3, desc_table_pt_int_temp, by = 'rprotocol_sub', all = T)
639 }
640 saveRDS(desc_table_pt1.3, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\desc_table_
      pt1_3.rds')
641 desc_table_pt1.3 <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\desc_table_
      pt1_3.rds')
642 desc_table_pt1.3[, -1] <- round(desc_table_pt1.3[, -1], digits = 2)
643 desc_table_pt1.3_srt <- desc_table_pt1.3[, c(1, 28:31, 2:27)]
644 desc_table_pt1.3_srt[is.na(desc_table_pt1.3_srt)] <- 0
645
646 desc_table_pt1.3_srt_t <- colSums(desc_table_pt1.3_srt[,-1])
647 desc_table_pt1.3_srt_t <- c('Population', desc_table_pt1.3_srt_t)
648 for (i in 1:15){
649   desc_table_pt1.3_srt_t[2*i+1] <- round(as.numeric(desc_table_pt1.3_srt_t[2*i])/176, digits = 2)
650 }
651 desc_table_pt1.3_srt <- rbind(desc_table_pt1.3_srt_t, desc_table_pt1.3_srt)
652
653 desc_table_pt1.3_l <- data.frame(matrix(nrow = 15, ncol = 7))
654
655 colnames(desc_table_pt1.3_l) <- unlist(desc_table_pt1.3_srt %>% distinct(rprotocol_sub))
656 for (i in 1:15){
657   for (j in 1:7){
658     desc_table_pt1.3_l[i, j] <- paste(desc_table_pt1.3_srt[j, (i)*2], '(', desc_table_pt1.3_srt[j, ((i)*2 + 1)], ')', sep = '
          ')
659   }
660 }
661
662 desc_table_pt1.3_l$cha <- c('Smoking', 'MI', 'Hypertension', 'Dyslipidemia', 'Diabetes', 'prior_cath', 'PCI', 'CABG',
663                             'Statin', 'ACEI/ARB', 'Antiplatelet', 'Beta Blocker', 'Diuretic', 'Calcium blockers', 'Nitrate')
664 desc_table_pt1.3_l <- desc_table_pt1.3_l[, c(ncol(desc_table_pt1.3_l), 1:(ncol(desc_table_pt1.3_l)-1))]
665 desc_table_1_3_kable <- kable(desc_table_pt1.3_l, "latex", booktabs = T, align = "c",
666                               caption = "Type I Error for Two sample tests of Spatial Normal Distributed Samples",
667                               digits = 2, longtable = F)
668
669 cat(desc_table_1_3_kable, file = paste('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\
      ', "desc_table_1_3_kable.txt", sep='')
670     , sep = "n", append = T)
671
672
673 # p values for table 1
674 # continuous: age bmi desc_table_pt1.2.1,
675 p1.1 <- c(summary(aov( BMI ~ factor(rprotocol_sub), data = tb1_data_ba))[[1]][[5]][[1]],
676         summary(aov( age ~ factor(rprotocol_sub), data = tb1_data_ba))[[1]][[5]][[1]])
677 names(p1.1) <- c('age', 'bmi')
678 round(p1.1, digits = 2)
679 # continuous pet uptake: desc_table_pt1.2.2,
680 p1.2.1 <- sapply(c('Cholest', 'LDL', 'HDL', 'rest_sbp', 'rest_dbp',
681                    'rest_hr', 'stress_sbp', 'stress_dbp', 'stress_hr'), function(k)
682                      summary(aov( as.numeric(eval(parse(text = k))) ~ factor(rprotocol_sub),
683                                   data = subset(tb1_data_ba, baseline == 1)))[[1]][[5]][[1]])
684 p1.2.2 <- sapply(c('Cholest', 'LDL', 'HDL', 'rest_sbp', 'rest_dbp',
685                    'rest_hr', 'stress_sbp', 'stress_dbp', 'stress_hr'), function(k)
686                      summary(aov( as.numeric(eval(parse(text = k))) ~ factor(rprotocol_sub),
687                                   data = subset(tb1_data_ba, baseline == 0)))[[1]][[5]][[1]])
688 p1.2 <- c(p1.2.1, p1.2.2)
689 names(p1.2) <- c('Cholesteral', 'LDL', 'HDL',
690                  'Rest Systolic blood pressure', 'Rest Diastolic blood pressure', 'Rest Heart Rate',
```

```
                'Stress Systolic blood pressure', 'Stress Diastolic blood pressure', 'Stress Heart Rate',
                'Cholesteral', 'LDL', 'HDL',
                'Rest Systolic blood pressure', 'Rest Diastolic blood pressure', 'Rest Heart Rate',
                'Stress Systolic blood pressure', 'Stress Diastolic blood pressure', 'Stress Heart Rate')
round(p1.2, digits = 2)
# categorical count: desc_table_pt1.3
p1.3_temp <- desc_table_pt1.3[,c(1, 2*(1:15))]
p1.3_temp[is.na(p1.3_temp)] <- 0
p1.3_temp <- p1.3_temp %>% mutate(size = c(50, 15, 50, 15, 31, 15))
p1.3_temp <- p1.3_temp[,c(1, 17, 15, 16, 2:14)]
p1.3 <- sapply(3:17, function(k)
  chisq.test(cbind(p1.3_temp[,2] - p1.3_temp[,k], p1.3_temp[,k]))$p.value)
names(p1.3) <- c('Smoking', 'MI', 'Hypertension', 'Dyslipidemia', 'Diabetes', 'prior_cath', 'PCI', 'CABG',
                 'Statin', 'ACEI/ARB', 'Antiplatelet', 'Beta Blocker', 'Diuretic', 'Calcium blockers', 'Nitrate')
round(p1.3, digits = 3)
# table 2, myocardial absolute flow and CFR, break into whole, anterior, septal, lateral, inferior.
# Use both P-value from t-test (the traditional approach) and spatially adjusted KS (My new approach)
for (i in 1:6){
  test1 <- c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')[i]
  nsize <- c(50, 15, 50, 15, 31, 15)[i]
  desc_table_pt2_1_temp <- data.frame(rprotocol_sub = test1)
  for (j in 1:3){
    name_j <- c('rest', 'stress', 'cfr')[j]
    value_nonbase <- protocol_pet_list[[i]][[1]][[j]][,1:nsize]
    value_base <- protocol_pet_list[[i]][[2]][[j]][,1:nsize]

    #print(round(c(mean(temp_data)), 2))
    desc_table_pt2_2_temp <- data.frame(rprotocol_sub = test1,
                                        nonbase_temp_mean = mean(unlist(value_nonbase)),
                                        nonbase_temp_sd = sd(unlist(value_nonbase)),
                                        base_temp_mean = mean(unlist(value_base)),
                                        base_temp_sd = sd(unlist(value_base)),
                                        diff_temp_mean <- mean(unlist(value_nonbase - value_base)),
                                        diff_temp_sd <- sd(unlist(value_nonbase - value_base)))
    colnames(desc_table_pt2_2_temp)[2:7] <- c(paste(name_j,'_test_mean'), paste(name_j, '_test_sd'),
                                              paste(name_j,'_base_mean'), paste(name_j, '_base_sd'),
                                              paste(name_j,'_diff_mean'), paste(name_j, '_diff_sd'))
    desc_table_pt2_1_temp <- merge(desc_table_pt2_1_temp, desc_table_pt2_2_temp,
                                   by = 'rprotocol_sub', all = T)
  }
  if (i ==1 ){
    desc_table_pt2 <- desc_table_pt2_1_temp
  }else{
    desc_table_pt2 <- rbind(desc_table_pt2, desc_table_pt2_1_temp)
  }
}
saveRDS(desc_table_pt2, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\desc_table_pt2.
    rds')
desc_table_pt2 <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\desc_table_
    pt2.rds')

desc_table_pt2_l <- data.frame(matrix(nrow = 6, ncol = 9))
colnames(desc_table_pt2_l) <- unlist(desc_table_pt2 %>% distinct(rprotocol_sub))
for (i in 1:9){
  for (j in 1:6){
    desc_table_pt2_l[j, i] <- paste(desc_table_pt2[j, (i)*2], '+', desc_table_pt2[j, ((i)*2 + 1)], sep = '')
  }
}
test1 <- as.data.frame((c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')))
colnames(test1) <- NULLs
desc_table_pt2_l <- cbind(test1, desc_table_pt2_l)
desc_table_pt2_kable <- kable(desc_table_pt2_l, "latex", booktabs = T, align = "c",
                              caption = "Type I Error for Two sample tests of Spatial Normal Distributed Samples",
                              digits = 2, longtable = F)

cat(desc_table_pt2_kable, file = paste('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\
    ', "desc_table_pt2_kable.txt", sep='')
    , sep = "n", append = T)

# p-values for table 2
tb2_pvalues <- cbind(pooled_KS_value0_P_ap1[,1:3], pooled_KS_value1_P_ap1[,1:2], pooled_KS_cfr_P_ap1[,1:2])
colnames(tb2_pvalues)[2:7] <- c('rest_statistic', 'rest_p',
                                'stress_statistic', 'stress_p',
                                'cfr_statistic', 'cfr_p')
saveRDS(tb2_pvalues, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\tb2_pvalues.rds')
tb2_pvalues <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\tb2_pvalues.rds'
    )
for (i in 1:6){
  test1 <- c('DD', 'L-15', 'L+10', 'L+40', 'L+55', 'L+80')[i]
  nsize <- c(50, 15, 50, 15, 31, 15)[i]
  desc_table_pt2_1_temp <- data.frame(rprotocol_sub = test1,
                                      rest = NA,
                                      stress = NA,
                                      cfr = NA)
  for (j in 1:3){
    name_j <- c('rest', 'stress', 'cfr')[j]
    value_nonbase <- protocol_pet_list[[i]][[1]][[j]][,1:nsize]
    value_base <- protocol_pet_list[[i]][[2]][[j]][,1:nsize]
```

```
775
776     #print(round(c(mean(temp_data)), 2))
777     p <- t.test(colMeans(value_nonbase), colMeans(value_base), paired = T)$p.value
778     desc_table_pt2_1_temp[1,(j+1)] <- p
779   }
780   if (i ==1 ){
781     desc_table_pt2 <- desc_table_pt2_1_temp
782   }else{
783     desc_table_pt2 <- rbind(desc_table_pt2, desc_table_pt2_1_temp)
784   }
785 }
786
787 # combine tb2_pvalues into
788 tb2_pvalues[,c(2, 4, 6)] <- desc_table_pt2[,2:4]
789 tb2_pvalues[, 2:7] <- round(tb2_pvalues[, 2:7], digits = 3)
790 colnames(tb2_pvalues)[2:7] <- c('ks_rest_p', 'rest_p',
791                                 'ks_stress_p', 'stress_p',
792                                 'ks_cfr_p', 'cfr_p')
793 saveRDS(tb2_pvalues, 'C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\tb2_pvalues.rds')
794 tb2_pvalues <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\tb2_pvalues.rds'
      )
795 tb2_pvalues_l <- kable(tb2_pvalues, "latex", booktabs = T, align = "c",
796                               caption = "Type I Error for Two sample tests of Spatial Normal Distributed Samples",
797                               digits = 3, longtable = F)
798
799 cat(tb2_pvalues_l, file = paste('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\Tables\\', "tb2
      _pvalues_kable.txt", sep='')
800     , sep = "n", append = T)
801 # table 3, capacity and KS tests
802 pooled_KS_P_ap1 <- readRDS('C:\\Users\\wzheng1\\Dropbox\\Dissertation\\Simulation\\Aim3\\Patients\\Data\\pooled_KS_P_ap1.rds'
      )
803 desc_table_pt3_kable <- kable(pooled_KS_P_ap1, "latex", booktabs = T, align = "c",
804                               caption = "Type I Error for Two sample tests of Spatial Normal Distributed Samples",
805                               digits = 2, longtable = F)
```