

COMPARISON STUDY ON SOME CLASSICAL LACK-OF-FIT TESTS IN REGRESSION MODEL

by

TEJ B SHRESTHA

B.Sc., Tribhuvan University, Nepal, 1988

M.Sc., Tribhuvan University, Nepal, 1991

M.S., Kansas State University, 2005

A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Statistics
College of Arts and Sciences

KANSAS STATE UNIVERSITY

Manhattan, Kansas

2010

Approved by:

Major Professor
Weixing Song

Copyright

TEJ SHRESTHA

2010

Abstract

The relationship between a random variable and a random vector is often investigated through the regression modeling. Because of its relative simplicity and ease of interpretation, a particular parametric form is often assumed for the regression function. If the pre-specified function form truly reflects the truth, then the resulting estimators and inference procedures would be reliable and efficient. But if the regression function does not represent the true relationship between the response and the predictors, then the inference results might be very misleading. Therefore, lack-of-fit test should be an indispensable part in regression modeling. This report compares the finite sample performance of several classical lack-of-fit tests in regression models via simulation studies. It has three chapters. The conception of the lack-of-fit test, together with its basic setup, is briefly introduced in Chapter 1; then several classical lack-of-fit test procedures are discussed in Chapter 2; finally, thorough simulation studies are conducted in Chapter 3 to assess the finite sample performance of each procedure introduced in Chapter 2. Some conclusions are also summarized in Chapter 3. A list of MATLAB codes that are used for the simulation studies is given in the appendix.

Table of Contents

Table of Contents	iv
List of Figures	v
List of Tables	vi
Acknowledgements	vii
Dedication	viii
1 Introduction	1
2 Classical Lack-of-Fit Tests	3
2.1 Reduction Method	8
2.2 General von Neumann Test	9
2.3 The von Neumann Test	13
2.4 Cusum Test	14
2.5 Summary of tests	16
3 Simulation studies	17
3.1 Replication Case Lack-of-fit Test:	17
3.2 Non-replication Case Lack-of-fit Test:	19
3.3 Reduction:	22
3.4 General von Neumann Test:	23
3.5 von Neumann and Buckley:	27
3.6 Conclusion	33
Bibliography	36
A Matlab Codes	37

List of Figures

3.1	Replication Power	18
3.2	Non-replication Specified Alternative	19
3.3	Non-replication	21
3.4	Non-replication	21
3.5	Reduction Test Power	22
3.6	General von Neumann Null Distribution	23
3.7	General von Neumann Test	25
3.8	von Neumann Null Distribution	30
3.9	Buckley Null Distribution	31
3.10	von Neuman Test Power	31
3.11	von Neumann test power(Bootstrap)	32
3.12	Buckley Test Power	32
3.13	Buckley Test Power(Bootstrap)	32

List of Tables

3.1	Replication Power	18
3.2	Non-Replication Specified Alternative Power(σ_d^2)	20
3.3	Non-Replication Specified Alternative Power(σ_e^2)	20
3.4	Non-Replication Specified Alternative Power(Bootstrap, σ_d^2)	20
3.5	Non-Replication Specified alternative Power(Bootstrap, σ_e^2)	20
3.6	Reduction Power Table	22
3.7	General von Neumann Critical Values	23
3.8	General von Neumann Test Power(Crit Distribution)	24
3.9	General von Neumann Test Power(Theorem 5.1)	24
3.10	General von Neumann Test Power(Bootstrap)	24
3.11	Critical Values	27
3.12	von Neumann Power(σ_d^2)	28
3.13	von Neumann Power(σ_e^2)	28
3.14	von Neumann Power(Bootstrap, σ_d^2)	28
3.15	von Neumann Power(Bootstrap, σ_e^2)	28
3.16	Buckely test Power(σ_d^2)	29
3.17	Buckely test Power(σ_e^2)	29
3.18	Buckely test Power(Bootstrap, σ_d^2)	29
3.19	Buckely test Power(Bootstrap σ_e^2)	29

Acknowledgments

I would like to thank my major advisor, Dr. Weixing Song. Without his valuable, continuous guidance and help, this work would be at this end. I am indebted to him from my heart. I would like to thank Dr. Haiyan Wang and Dr. David N Yetter for serving in my committee.

I would like to thank all members of statistics department of Kansas State University whose kindness and friendliness are priceless to me and will not be forgotten.

TO MY SON...

Chapter 1

Introduction

The relationship between a random variable and a random vector is often investigated through regression modeling. Because of its relative simplicity and ease of interpretation, parametric regression is commonly used in practice. By specifying a particular parametric form for the regression function, the main problem is to estimate the regression parameter. After obtaining estimators for the parameters, the fitted regression then can be used to make some statistical inferences. If the pre-specified function form truly reflects the actual relationship among those interested variables, then the resulting estimator or inference procedure would be reliable and efficient. But if the regression model is wrongly specified, then the following statistical inference might be very misleading, even disastrous. Therefore, to assess the adequacy of the parametric form for the regression function should become an indispensable part in regression modeling. The test procedure of fulfilling this objective is called lack-of-fit test.

More formally, let Y denote the response variable, and X denote the explanatory variable, the conditional expectation of $r(x) = E(Y|X = x)$ is called the regression function of Y on X . The regression model can be written as

$$Y = r(X) + e$$

where e is the random error. In classical regression model, e is often assumed to have a

normal distribution. Let Θ be a parameter space, and $S_{\Theta} = \{r(\cdot; \theta) : \theta \in \Theta \subset \mathbb{R}^n\}$ be a class of parametric functions. We want to test the following hypothesis

$$H_0 : r(x; \theta) \in S_{\Theta} \text{ v.s. } H_a : H_0 \text{ is not true.}$$

If $r(x)$ has finite unknown parameters then the model is called parametric otherwise non-parametric. Graphical tools are often used to check the appropriateness of the specified regression function. For example, a variety of residual plots can be used to identify the appropriateness of the regression function by observing the scatter plot of the residuals about the line $y = 0$. But the conclusion based on the residual plots are subject to personal judgement, and not very objective. To overcome this drawback, many formal numerical lack-of-fit test procedures are proposed. In this report, we shall investigate some popular classical lack-of-fit test procedures and numerical comparisons are made via simulation among these test procedures.

Chapter 2

Classical Lack-of-Fit Tests

We begin with the general framework of hypothesis testing in statistics. A hypothesis H_0 with respect to a population quantity, say θ , of a statistical model is proposed, one needs to develop a procedure to check the plausibility of the hypothesis based on the data. Usually, when we propose a hypothesis H_0 , we also set up an alternative hypothesis H_a , which is opposite or partly opposite to the null hypothesis. Therefore, the formal hypothesis framework is to find a procedure to test

$$H_0 : \theta \in \Theta_0, \quad \text{versus} \quad H_a : \theta \in \Theta_a, \quad (2.1)$$

where $\Theta_0 \cap \Theta_a = \emptyset$.

In fact, developing a test procedure for the above hypothesis is to specify a set, say C , of the data values. C is often called critical region or rejection region. If the sample belongs to C , then reject H_0 , otherwise, accept H_0 . It is well known that for any test procedure, we are likely to commit two types of mistakes: type I error and type II error. Type I error is a mistake that reject H_0 when H_0 is true, and type II error is a mistake that fail to reject H_0 when actually it is false. A test is of significance level α implies $\sup_{\theta \in \Theta_0} P_\theta(C|H_0 \text{ is true}) \leq \alpha$, and the power function of this test is defined as $\pi(\theta) = P_\theta(C|H_0 \text{ is not true}), \theta \in \Theta_a$. A test with significance level α is called the most powerful test if it has the biggest power among all tests with significance level α .

If Θ_0 contains only one value, then the hypothesis H_0 is called simple hypothesis. In the

case of simple H_0 and H_a , Neyman-Pearson lemma claims that the likelihood ratio test is most powerful.

Neyman-Pearson Lemma: *Let X_1, X_2, \dots, X_n be a random sample from a population with density function $f(x; \theta)$. Suppose we want to test the following hypothesis,*

$$H_0 : \theta = \theta_0, \quad \text{versus} \quad H_a : \theta = \theta_1.$$

Define

$$C = \left\{ (X_1, X_2, \dots, X_n) : \frac{L(\theta_0; X_1, X_2, \dots, X_n)}{L(\theta_1; X_1, X_2, \dots, X_n)} \leq k \right\}$$

where k is some constant such that $P_{\theta_0}(C) = \alpha$ and $L(\theta; X_1, X_2, \dots, X_n)$ is the likelihood function $\prod_{i=1}^n f(X_i; \theta)$. Then the test given by the critical region C is the most powerful test with significance level α .

In general case, H_0 and H_a may be composite, that is, they contain more than one point.

A natural generalization of above Neyman-Pearson test statistic is

$$\Lambda = \frac{\sup_{\theta \in \Theta_0} L(\theta; X_1, X_2, \dots, X_n)}{\sup_{\theta \in \Theta_a} L(\theta; X_1, X_2, \dots, X_n)}.$$

Although we can not guarantee the test based on Λ is most powerful, but certainly it is a desirable test. In the context of regression, we shall assume that (x_i, Y_i) , $i = 1, 2, \dots, n$ obey the following regression model

$$Y_i = r(x_i; \theta) + \varepsilon_i, \tag{2.2}$$

where Y_i is the response, x_i is the predictor which may be multidimensional, ε_i 's are random errors which are independent random variables with mean 0 and variance $\sigma^2 < \infty$. In this report, the regression model (2.2) is assumed to have a fixed design, that is, x_i is fixed constant. Without loss of generality, we will assume x_i 's are equally spaced on $[0, 1]$.

Suppose it is of interest to test the following hypothesis

$$H_0 : \theta \in \Theta_0 \quad \text{versus} \quad H_a : \theta \in \Theta_a = \Theta - \Theta_0,$$

where Θ is the parameter space. If $\varepsilon_i \stackrel{\text{i.i.d.}}{\sim} g(\cdot; \phi)$, where g is a known function up to some unknown parameter ϕ . The general likelihood ratio test statistic is given by

$$\Lambda_n = \frac{\sup_{\theta \in \Theta_0, \phi \in \Phi} L(\theta, \phi; Y)}{\sup_{\theta \in \Theta, \phi \in \Phi} L(\theta, \phi; Y)},$$

where

$$L(\theta, \phi; Y) = \prod_{i=1}^n g(Y_i - r(x_i, \theta), \phi).$$

If the null model is appropriate, Λ_n should be larger, smaller value of Λ_n is an evidence indicating the null model may not be appropriate. To test the hypothesis with specific significance level, we need to find out the critical region, which in turn depends on the distribution of Λ_n under null hypothesis. The exact distribution of Λ_n might be difficult to find. It is well known that under H_0 and other regularity conditions, $-2 \ln(\Lambda_n) \sim \chi^2_\alpha(p - p_0)$, where p and p_0 are the number of free parameters in the full model and the null model, respectively. That is, for large sample sizes, one can reject H_0 if $-2 \ln(\Lambda_n) \geq \chi^2(p - p_0)$. Kendall and Stuart¹(1979) and Chernoff²(1954) provide sufficient conditions for the validity of above asymptotic result.

Although likelihood ratio test is a powerful test, it is designed to be for parametric models and the distribution of error should be known. Moreover, in non-nested cases, $-2 \ln(\Lambda_n) \sim \chi^2(p - p_0)$ is not valid in general. So general likelihood ratio test is not a universal test. Cox³(1962) proposed a modified likelihood ratio test procedure for the non-nested cases, and White⁴(1982) provides conditions for this modified test statistic to be asymptotically normal. For a detailed treatment of the non-nested models, see Pace and Salvani⁵(1990).

Construction of a Test Statistic: Variance Ratio

To derive a general method for the lack-of-fit test, let's consider a special case in which ε_i 's are i.i.d. from normal distribution with mean 0 and variance σ^2 . Then the likelihood ratio Λ_n has the following form:

$$\Lambda_n = \frac{\sup_{\theta \in \Theta_0, \sigma > 0} \sigma^{-n} \exp \left\{ - \sum_{i=1}^n [Y_i - r(x_i; \theta)]^2 / 2\sigma^2 \right\}}{\sup_{\theta \in \Theta, \sigma > 0} \sigma^{-n} \exp \left\{ - \sum_{i=1}^n [Y_i - r(x_i; \theta)]^2 / 2\sigma^2 \right\}}$$

which can be reduced to a simple ratio

$$\Lambda_n = \left(\frac{\sigma^2(\hat{\theta})}{\sigma^2(\hat{\theta}_0)} \right)^{n/2},$$

where $\hat{\theta}$ and $\hat{\theta}_0$ are the estimators of θ under the full model and the null model which minimizes

$$\sigma^2(\theta) = \frac{1}{n} \sum_{i=1}^n [Y_i - r(x_i; \theta)]^2,$$

when $\theta \in \Theta$ and $\theta \in \Theta_0$, respectively. This implies that one can reject H_0 if $\sigma^2(\hat{\theta}_0)/\sigma^2(\hat{\theta})$ is large.

Therefore, the above variance ratio test is equivalent to general likelihood ratio test when the random error follows a normal distribution with mean 0 and variance σ^2 . Although this equivalence does not hold for general cases, it does provide a general way to construct lack-of-fit test procedures: For a regression model, one can obtain one good estimator, say $\hat{\sigma}_M^2$, of the variance of the random error when the null model is true, another good estimator, say $\hat{\sigma}^2$, of the variance of the random error regardless of the validity of the null model. Obviously, $\hat{\sigma}_M^2$ is model dependent, and $\hat{\sigma}^2$ is not or less model dependent. Then the ratio $\hat{\sigma}_M^2/\hat{\sigma}^2$ can be used as a lack-of-fit test procedure. In fact, many classical lack-of-fit test procedures are of the variance ratio types.

Example (Pure Experimental Error and Lack-of-Fit). Consider the following regression model:

$$Y_{ij} = r(x_i) + \epsilon_{ij} \quad j = 1, 2, \dots, n_i; i = 1, 2, \dots, n. \quad (2.3)$$

where $n_i > 1$ at least at some treatment point i , that is, we have replicated observations at least at one design point. Suppose we want to test $H_0 : r(x) = r(x; \theta)$ for some $\theta \in \Theta_0$. If $\hat{\theta}$ is an estimator of θ under the null hypothesis, then a model dependent estimator under H_0 is

$$\hat{\sigma}_M^2 = \frac{SSE_M}{n-p} = \frac{\sum_{i=1}^n n_i (\bar{Y}_i - \hat{Y}_i)^2}{n-p} = \frac{\sum_{i=1}^n n_i (\bar{Y}_i - r(x_i; \hat{\theta}))^2}{n-p},$$

where p is the dimension of θ . A model-free estimator of σ^2 can be construct using the pure experimental errors at each design point, that is

$$\hat{\sigma}^2 = \frac{SSE_P}{N-n} = \frac{\sum_{i=1}^n \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_i)^2}{N-n}$$

with $N = \sum_{i=1}^n n_i$. In fact, one can show that $\hat{\sigma}^2 = SSE_P/(N - n)$ is an unbiased estimator of the error variance σ^2 .

Hence the test statistic can be taken as

$$V_R = \frac{SSE_M/(n - p)}{SSE_P/(N - n)}.$$

The distribution of V_R depends on the distribution of the random error and the regression function under the null hypothesis. If the random error is normally distributed, r is linear in p unknown parameters and the parameters are estimated by least squares, then $V_R \sim F_{n-p, N-n}$.

Estimator Based on Pseudo-Residuals

The variance test statistic discussed in the above example is not applicable if no replications are available. In this case, if the regression function is sufficiently smooth, then one can use pseudo-residuals to construct an estimator of the error variance which does not or less depends on the regression function under H_0 , see Gasser, Sroka, Jennen-Steinmetz⁶(1986).

Method 1: $\hat{\sigma}_d^2$

If the regression is smooth enough, then $Y_i - Y_{i-1}$ can be approximated by $\varepsilon_i - \varepsilon_{i-1}$. An estimator of σ^2 can be constructed as follows

$$\hat{\sigma}_d^2 = \frac{1}{2(n-1)} \sum_{i=2}^n (Y_i - Y_{i-1})^2. \quad (2.4)$$

Method 2: $\hat{\sigma}_e^2$

One can also use the following pseudo-residuals:

$$\hat{e}_i = \frac{x_{i+1} - x_i}{x_{i+1} - x_{i-1}} Y_{i-1} + \frac{x_i - x_{i-1}}{x_{i+1} - x_{i-1}} Y_{i+1} - Y_i = a_i Y_{i-1} + b_i Y_{i+1} - Y_i.$$

The resulting estimator of the error variance is

$$\hat{\sigma}_e^2 = \frac{1}{n-2} \sum_{i=2}^{n-1} \frac{\hat{e}_i^2}{a_i^2 + b_i^2 + 1}. \quad (2.5)$$

If we denote

$$\mathbf{H} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & -1 & 1 \end{pmatrix},$$

then $\hat{\sigma}_d^2 = Y'HY/(2n - 2)$. If the design points x_i 's are equally spaced, then $a_i = b_i = 1/2$.

Let

$$\mathbf{D} = \begin{pmatrix} 1 & -2 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & \cdots & 0 & 0 & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & -2 & 1 \end{pmatrix},$$

then $\hat{\sigma}_e^2 = Y'D'DY/(6n - 12)$.

But to apply them to the lack-of-fit test, we need to know the distribution of the test statistic under H_0 which can not be obtained easily.

2.1 Reduction Method

If both null hypothesis and alternative hypothesis are linear regression models, and the null hypothesis is nested in the alternative hypothesis, that is, the hypothesis of interest is

$$H_0 : r(x) = \sum_{j=1}^p \theta_j r_j(x) \text{ vs. } H_a : r(x) = \sum_{j=1}^{p+k} \theta_j r_j(x) \quad k \geq 1; 0 \leq x \leq 1,$$

then one can use the reduction method to construct the test statistic. Denote SSE_0 and SSE_a as the sums of squares of errors under null and alternative models, respectively. Then we can use

$$F_R = \frac{(SSE_0 - SSE_a)/k}{SSE_a/(n - p - k)} \quad (2.6)$$

to test the hypothesis. If further assume that errors are i.i.d. $N(0, \sigma^2)$, then $F_R \sim F_{k, n-p-k}$.

F_R is a variance ratio test statistic, the numerator is unbiased estimator of variance of errors under H_0 and the denominator is unbiased estimator of error variance under both

null and alternative hypothesis. If $\epsilon_i \stackrel{i.i.d.}{\sim} N(0, \sigma^2)$, this variance ratio test is equivalent to the Gaussian errors likelihood ratio test.

Lehmann⁷(1959) shows that among the classes of invariant tests, the reduction test is uniformly most powerful for testing the above hypothesis.

Note that the reduction test is especially design for the above alternative model. If the alternative model is not the form of the alternative model as above, then the reduction test might have very low power.

For example consider $Y_i = r(i/n) + \epsilon_i \quad i = 1, 2, \dots, n$ and

$$H_0 : r(x) = \theta_0 + \theta_1 x \quad H_a : r_a(x) = \sum_{j=1}^3 \gamma_j x^j, \quad \int_0^1 r_a(x) x^j dx = 0; \quad j = 0, 1, 2.$$

Clearly, $\gamma_j \approx 0$ for $j = 0, 1, 2$ and the model is neither a linear nor a quadratic but H_0 is linear. In this case the reduction test has no power at all.

2.2 General von Neumann Test

Now let's consider the hypothesis $H_0 : r(x) = \sum_{j=1}^p \theta_j r_j(x)$. Suppose the design matrix

$$\mathbf{R} = \begin{pmatrix} r_1(x_1) & r_2(x_1) & \cdots & r_p(x_1) \\ r_1(x_2) & r_2(x_2) & \cdots & r_p(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ r_1(x_n) & r_2(x_n) & \cdots & r_p(x_n) \end{pmatrix}$$

is of full rank which ensures that there is a unique set of least square estimators $\hat{\theta}_0, \hat{\theta}_1, \dots, \hat{\theta}_p$ of the parameters $\theta_0, \theta_1, \dots, \theta_p$. Denote $R(R'R)^{-1}R'$ by M , the perpendicular projection matrix onto the column space of R , and the i th component of residuals by $e_i = Y_i - \sum_{j=1}^p \hat{\theta}_j r_j(x_i) \quad i = 1, 2, \dots, n$. Then the residual vector can be written as

$$e = [I - M]Y \text{ where } Y = (Y_1, Y_2, \dots, Y_n)'$$

I is the unit matrix of size n . Then a model based unbiased estimator of error variance is

$$\hat{\sigma}_M^2 = e'e/(n - p) = \frac{1}{n - p} Y'(I - M)Y.$$

If $r(x)$ is smooth, a less model-dependent estimator of error variance is

$$\hat{\sigma}^2 = \frac{1}{a_n} \sum_{i=2}^n (e_i - e_{i-1})^2 = \frac{1}{a_n} e' H e$$

where $a_n = \text{trace}((I - M)H(I - M)) = 2(n - 1) - \text{trace}(HM)$, is unbiased estimator of σ^2 under H_0 and is consistent for σ^2 as long as the null model and $r(x)$ are piecewise smooth. Note that, we are not restricting $r(x)$ to be a linear.

Hence, a lack-of-fit test of H_0 without specifying any alternative models can be implemented by the variance ratio statistic

$$V_n = \frac{\hat{\sigma}_M^2}{\hat{\sigma}^2}.$$

We call this testing procedure the general von Neumann test. Other candidates for the denominator are $\hat{\sigma}_e^2$ and $\hat{\sigma}_d^2$ defined in (2.4) and (2.5). Note that, if our null model is linear, the test based on V_n with denominator $\hat{\sigma}_e^2$ and $\hat{\sigma}_d^2$ is equivalent to the test discussed in previous example but without replication.

It is easy to see that the test statistic V_n can be written as the ratio of two quadratic forms of Y :

$$V_n = \frac{Y'AY}{Y'BY}, \text{ where } A = \frac{1}{n-p}(I-M) \quad B = \frac{(n-p)^2}{a_n}AHA.$$

Under H_0 , $AY = A\epsilon$, so

$$V_n = \frac{\epsilon' A \epsilon}{\epsilon' B \epsilon}.$$

To find the critical values for testing H_0 , one needs to find the following probability,

$$P(V_n \geq u) = P[\epsilon'(A - uB)\epsilon \geq 0].$$

If ϵ follows a normal distribution with mean 0 and variance σ^2 , then Theorem 2.1 in Box⁸(1954) shows that

$$P[\epsilon'(A - uB)\epsilon \geq 0] = P\left[\sum_{j=1}^r \lambda_j(u) \chi_j^2 \geq 0\right]$$

where $r = \text{rank}(A - uB)$, $\lambda_1, \lambda_2, \dots, \lambda_r$ are the non-zero eigenvalues of $A - uB$ and $\chi_j^2 \stackrel{iid}{\sim} \chi_{(1)}^2$, $j = 1, \dots, r$. Unfortunately, this theoretical result is hard to use. Therefore, we shall use the following simulation algorithm to find the critical value:

Method I: Since the distribution of $V_n = \frac{\varepsilon' A \varepsilon}{\varepsilon' B \varepsilon}$ is invariant with respect to the variance of ε , without loss of generality, we can take the variance of ε to be 1. Based on the simulation set up, it will be sufficient to consider the values of u only in the interval $[-3, 3]$. In our simulation, 1000 equally spaced values of u from the interval are used. In some cases, for more precise result, we can choose subinterval of $[-3, 3]$. The following algorithm can be used to obtain the critical value of the variance ratio test when significance level is α .

- (i) Let u be one of 1000 selected values from $[-3, 3]$, calculate $r = \text{rank}(A - uB)$;
- (ii) Calculate the non-zero eigenvalues of $A - uB$ and denote them as $\lambda_j(u), j = 1, 2, \dots, r$
- (iii) Generate r independent sample from χ_1^2 , denoted as $X_j, j = 1, 2, \dots, r$, and calculate the sum $x_u = \sum_{j=1}^r \lambda_j(u) X_j$;
- (iv) Repeat step (ii)-(iii) a large number of times, say R , to obtain R x_u 's, then calculate $p_u = \#\{u : x_u \geq 0\} / R$. In our simulation $R=1000$.
- (v) Repeat step (i)-(iv) for each u selected from $[-3, 3]$, then we get 1000 pairs of (u, p_u) . Fitting the data set $\{(u, p_u)\}$ with kernel smoother, which gives as estimated CDF curve of V_n . For any given significance level α , one can find the $100(1 - \alpha)$ th percentile of the estimated CDF which is the desired critical value.

If ε follows a non-normal distribution or simply the distribution is unknown, we can use the following result to obtain a large sample critical value:

Theorem 5.1 [Hart⁹]: Suppose the regression model holds $Y_i = r(x_i) + \varepsilon_i \quad i = 1, 2, \dots, n$ with r having the linear form and ε_i are independent random variables with common variance σ^2 and $E|\varepsilon_i|^{2+\delta} < M$ for all i and positive constants δ and M . If $\hat{\theta}_0, \hat{\theta}_1, \dots, \hat{\theta}_p$ satisfy

$$E(\hat{\theta}_j - \theta_j)^2 = O(1/n) \quad j = 1, 2, \dots, p \text{ and } |r_j(x)| < \alpha$$

a constant for all j and $x \in [0, 1]$, then

$$\sqrt{n}(V_n - 1) \xrightarrow[n \rightarrow \infty]{D} N(0, 1).$$

Therefore, if the sample size is large enough, we can use the following method to find the critical value:

Method II: Based on Hart's theorem, the critical value for the test statistic V_n is given by $1 + z_{1-\alpha}/\sqrt{n}$, where $z_{1-\alpha}$ is the upper $100(1 - \alpha)$ th percentile of the standard normal distribution.

Another way to obtain the critical values is using bootstrap. Generally, bootstrap method behaves very well when the sample size is small or moderate. Unfortunately, if we do not set up the alternative, developing a good bootstrap algorithm is infeasible. The following algorithm, called naive bootstrap algorithm, do not require the information on alternative model, but it does need data from the null model.

Method III:

1. Bootstrap from the empirical distribution of the residuals

$e_i = Y_i - \sum_{j=1}^p \hat{\theta}_j r_j(x_i), i = 1, 2, \dots, n$. Denote $\{e_i^*\}_{i=1}^n$ as the bootstrap sample from $\{e_i\}_{i=1}^n$. Here $(Y_i, x_i), i = 1, 2, \dots, n$ are the data from null model.

2. Compute $\sqrt{n}(V_n - 1)$ based on the bootstrap sample or directly the V_n .
3. Repeat above steps m times and then $100(1 - \alpha)$ th percentile of bootstrap values $\sqrt{n}(V_n - 1)$ or of V_n can be used for the critical value for the α significant level test.

In the simulation study, we will use the above three methods to find the critical values, then comparison will be made among different testing procedures.

The reduction method needs the specification of the alternative hypothesis, but the general von Neumann test does not need the alternative model. Therefore, the general von Neumann test is an omnibus test.

2.3 The von Neumann Test

We are interested in constructing a test which can be used for wide class of alternatives. Consider the simple, maybe, simplest hypothesis

$$H_0 : r(x) = C; C \text{ is an unknown constant and } 0 \leq x \leq 1. \quad (2.7)$$

On this model, x has no effect on regression function at all. In case of smooth departure of r from constancy, von Neuman¹⁰ proposed the test statistic

$$F_N = \frac{s^2}{\hat{\sigma}_d^2}, \quad (2.8)$$

to test above hypothesis, where

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (Y_i - \bar{Y})^2,$$

and $\hat{\sigma}_d^2$ is as defined on page 7, Method 1. Since

$$E(\hat{\sigma}_d^2) = \sigma^2 + O(n^{-2}),$$

if H_0 is false and r is smooth, we can consider it as a unbiased estimator of variance(at least for large sample size). Also under H_0 ,

$$E(s^2) = \sigma^2 + \frac{1}{n-1} \sum_{i=1}^n [r(x_i) - \bar{r}_n]^2 = \sigma^2 + O(1/n),$$

where $\bar{r}_n = \sum_{i=1}^n r(x_i)$, can be regarded as unbiased estimator of σ^2 . Thus the F_N is a test statistic for the lack-of-fit test.

Further, if Y_1, Y_2, \dots, Y_n are i.i.d Gaussian distribution, the test statistic under null hypothesis can also be written as

$$F_N = 2 \frac{Y'(I - n^{-1}J_n)Y}{Y'HY}. \quad (2.9)$$

This representation of the variance ratio suggests us to the approximation of critical value as the algorithm discussed above. The obvious other candidate for denominator is σ_e^2 . In this case, the test statistic can be written as

$$F_N = \frac{(2n - 12)Y'(I - n^{-1}J_n)Y}{(n - 1)Y'D'DY}. \quad (2.10)$$

where D is the matrix defined before.

2.4 Cusum Test

Cusum-based procedures have been used in quality control to detect in the mean of sequence of variables. Buckley¹¹ has shown that a cusum-based test is locally most powerful in certain Bayesian models. For a regression model,

$$Y_i = r(x_i) + \epsilon_i, \quad i = 1, 2, \dots, n,$$

the cumulative sum or cusum, S_j is defined by

$$S_j = \sum_{i=1}^j (Y_i - \bar{Y}) \quad i = 1, 2, \dots, n. \quad (2.11)$$

Buckley¹¹ suggests the test statistic

$$T_B = \frac{\sum_{j=1}^n S_j^2}{n^2 \hat{\sigma}_d^2}. \quad (2.12)$$

Since

$$E\left(\frac{1}{n^2} \sum_{j=1}^n S_j^2\right) = \frac{\sigma^2(n-1)(n+1)}{6n^2} + \frac{1}{n^2} \sum_{j=1}^n \sum_{i=1}^j [r(x_i) - \bar{r}]^2,$$

under H_0 , the estimate of numerator is proportional to σ^2 and tends to be larger than as it is on null hypothesis but the denominator is consistent. Since it uses the cumulative sum, this test measures the departure in a smooth way different from the other tests we discussed. Hence, it is called the smooth test. The numerator can also be written as

$$\frac{1}{n^2} \sum_{j=1}^n S_j^2 = \frac{1}{n^2} Y'(I - n^{-1}J_n)C_n(I - n^{-1}J_n)Y \quad (2.13)$$

where

$$\mathbf{C}_n = \begin{pmatrix} n & n-1 & n-2 & n-3 & \cdots & 2 & 1 \\ n-1 & n-1 & n-2 & n-3 & \cdots & 2 & 1 \\ n-2 & n-2 & n-2 & n-3 & \cdots & 2 & 1 \\ n-3 & n-3 & n-3 & n-3 & \cdots & 2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 2 & 2 & 2 & 2 & \cdots & 2 & 1 \\ 1 & 1 & 1 & 1 & \cdots & 1 & 1 \end{pmatrix}.$$

Thus, we can approximate the critical value of the test using algorithm discussed in section 2.1.3.

2.5 Summary of tests

All the information about the test statistics corresponding to lack-of-fit tests we discussed can be put together as the following table. The second column contains the test statistic for the method of test in the first column, the third column accounts the methods of obtaining critical value and the last column is the null model we are going to select for simulation studies of the lack-of-fit test.

Test	Test Statistic	Critical value	Simulation Model
Variance ratio in Replication case	$\frac{SSE_M/(n-p)}{\hat{\sigma}_P^2}$	F-distribution	$H_0 : r(x) = \beta_0 + \beta_1 X$ $H_a : r(x) = \beta_0 + \beta_1 X + \beta_2 X^2$
Variance ratio in Nonreplication case	$\frac{Y'(I-M)Y/(n-p)}{\sigma_d^2, \sigma_e^2}$	Simulation Bootstrap	$H_0 : r(x) = \beta_0 + \beta_1 X$ $H_a : r(x) = \beta_0 + \beta_1 X + \beta_2 X^2$
Reduction	$\frac{SSE_0 - SSE_a/k}{SSE_a/(n-p-k)}$	F-distribution	$H_0 : r(x) = \beta_0 + \beta_1 X$ $H_a : r(x) = \beta_0 + \beta_1 X + \beta_2 X^2$
General von Neumann	$\frac{Y'(I-M)Y/(n-p)}{\bar{\sigma}^2}$	Simulation, Theorem 5.1, Bootstrap	$H_0 : r(x) = \beta_0 + \beta_1 X$ $H_a : r(x) = \beta_0 + \beta_1 X + \beta_2 X^2$
von Neumann	$\frac{s^2}{\sigma_d^2, \sigma_e^2}$	Simulation Bootstrap	$H_0 : r(x) = \beta_0$ $H_a : r(x) = \beta_0 + \beta_1 X$
Buckley	$\frac{n^{-2} \sum S_j^2}{\sigma_d^2, \sigma_e^2}$	Simulation Bootstrap	$H_0 : r(x) = \beta_0$ $H_a : r(x) = \beta_0 + \beta_1 X$

Chapter 3

Simulation studies

Consider the regression model

$$Y_i = r(x_i) + \varepsilon_i$$

where the ε_i is a normal random number with mean zero and variance unit. The hypothesis is chosen as

$$H_0 : r(x) = \theta_0 + \theta_1 x \quad vs. \quad H_a : r(x) = \beta_0 + \beta_1 x + \beta_2 x^2. \quad (3.1)$$

For simulation, we chose $\theta_0 = 1, \theta_1 = 2, \beta_0 = 1, \beta_1 = 2, \beta_2 = c$, and $c = 0, 1, 2, \dots, 5$. The sample of sizes 50, 100, 200, 300, 400, and 500 were chosen for all tests. The empirical power of the lack-of-fit test is the probability of rejecting null hypothesis if null is not true and is calculated in 1000 identical trials.

3.1 Replication Case Lack-of-fit Test:

Consider 3 replication case at each design point with regression function $r(x) = \beta_0 + \beta_1 x$. The test statistic is $\frac{SSE_M/(n-p)}{\hat{\sigma}_P^2}$ and is defined on (2.4). Since the test statistic has F-distribution $F(n-p, N-n)$ under the null hypothesis, the critical value for the test is $F(n-p, N-n, 1-\alpha)$, where $\alpha = 0.05$, the significance level and , $N = 3n, p = 2$.

c	50	100	200	300	400	500
0	0.054	0.039	0.055	0.046	0.046	0.046
1	0.047	0.055	0.065	0.069	0.075	0.091
2	0.102	0.100	0.144	0.169	0.175	0.208
3	0.172	0.235	0.324	0.416	0.517	0.587
4	0.268	0.415	0.640	0.757	0.857	0.921
5	0.486	0.649	0.896	0.974	0.994	0.998

Table 3.1: *Replication Power*

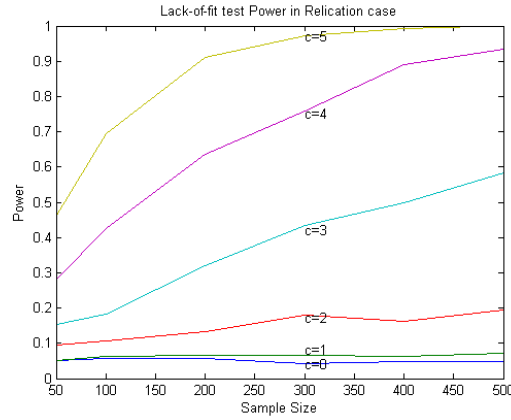


Figure 3.1: *Replication Power*

This empirical power table (Table 3.1) and graph (Figure 3.1) say that as the alternative model is deviating more and more from null model, the power of the Replication test becomes bigger and bigger. For larger sample sizes, this test has a very good power. But for the alternatives which are close to the null model, there may be chances of making wrong decision but as the alternative model is departing from null much, this test has good power to make good decision.

3.2 Non-replication Case Lack-of-fit Test:

Let us denote the design matrix by X and the perpendicular projection matrix onto column space of X is by M . The critical values for the Non-replication variance ratio test are generated by matlab code(listed on "Appendix A") using the algorithm(see section 2.2) discussed before for null distribution under the Method I and Method III, where $A = (I - M)/(n - p)$, n , the sample size taken and $p = 2$, the number of parameter on the model used, and $B = H/(2n - 2)$ for σ_d^2 case or $B = D'D/(6n - 12)$ for σ_e^2 case which are defined in the discussion of variance ratio("Estimator Based on Pseudo-Residual"). The simulation result for critical values and the distribution of the test statistic under null model are given bellow:

n	50	100	200	300	400	500
σ_d^2	1.2889	1.2222	1.1333	1.1111	1.1000	1.0889
σ_e^2	1.5152	1.2727	1.1970	1.1667	1.1515	1.1364

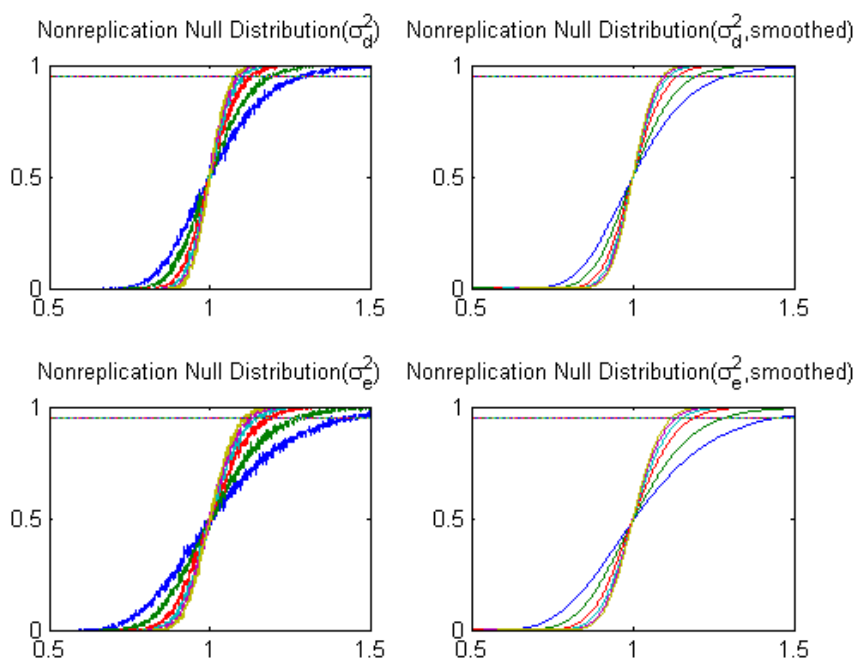


Figure 3.2: Non-replication Specified Alternative

The empirical power of the test obtained by using above critical values and using σ_a^2 and σ_e^2 for the above model considered are as follow:

c	50	100	200	300	400	500
0	0.048	0.051	0.045	0.049	0.037	0.048
1	0.070	0.087	0.089	0.097	0.073	0.135
2	0.145	0.251	0.341	0.386	0.433	0.595
3	0.348	0.549	0.737	0.883	0.939	0.976
4	0.600	0.867	0.979	0.999	0.999	1.000
5	0.844	0.973	1.000	1.000	1.000	1.000

Table 3.2: *Non-Replication Specified Alternative Power(σ_a^2)*

c	50	100	200	300	400	500
0	0.028	0.054	0.040	0.033	0.038	0.020
1	0.046	0.070	0.054	0.068	0.044	0.040
2	0.096	0.174	0.197	0.213	0.228	0.235
3	0.197	0.369	0.523	0.622	0.714	0.807
4	0.405	0.696	0.865	0.961	0.976	0.997
5	0.624	0.920	0.992	0.999	1.000	1.000

Table 3.3: *Non-Replication Specified Alternative Power(σ_e^2)*

c	50	100	200	300	400	500
0	0.063	0.050	0.045	0.047	0.046	0.047
1	0.052	0.040	0.057	0.072	0.041	0.057
2	0.080	0.083	0.097	0.118	0.124	0.134
3	0.108	0.124	0.170	0.216	0.283	0.277
4	0.182	0.195	0.328	0.418	0.505	0.550
5	0.255	0.341	0.531	0.686	0.775	0.863

Table 3.4: *Non-Replication Specified Alternative Power(Bootstrap, σ_a^2)*

c	50	100	200	300	400	500
0	0.037	0.049	0.044	0.063	0.049	0.053
1	0.035	0.051	0.044	0.046	0.048	0.059
2	0.049	0.057	0.058	0.096	0.087	0.103
3	0.077	0.085	0.115	0.184	0.155	0.204
4	0.077	0.147	0.213	0.286	0.320	0.402
5	0.171	0.249	0.350	0.457	0.548	0.650

Table 3.5: *Non-Replication Specified alternative Power(Bootstrap, σ_e^2)*

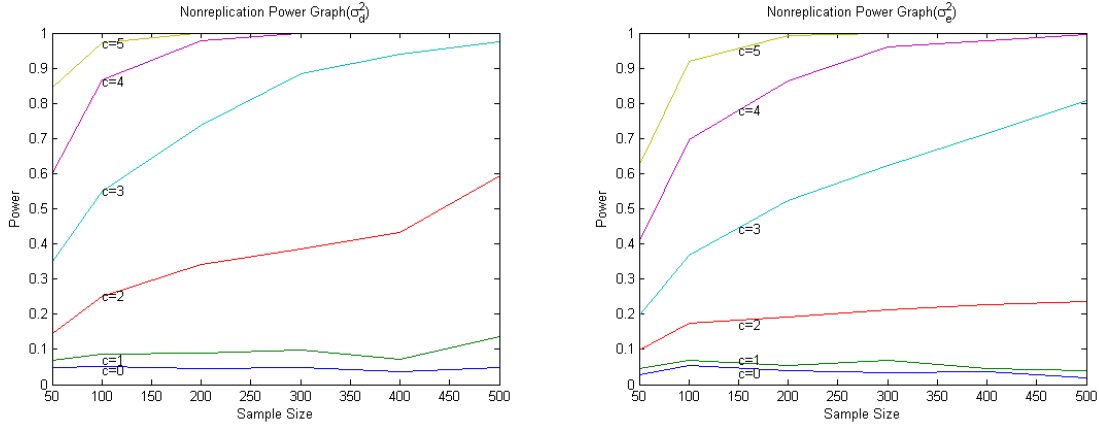


Figure 3.3: *Non-replication*

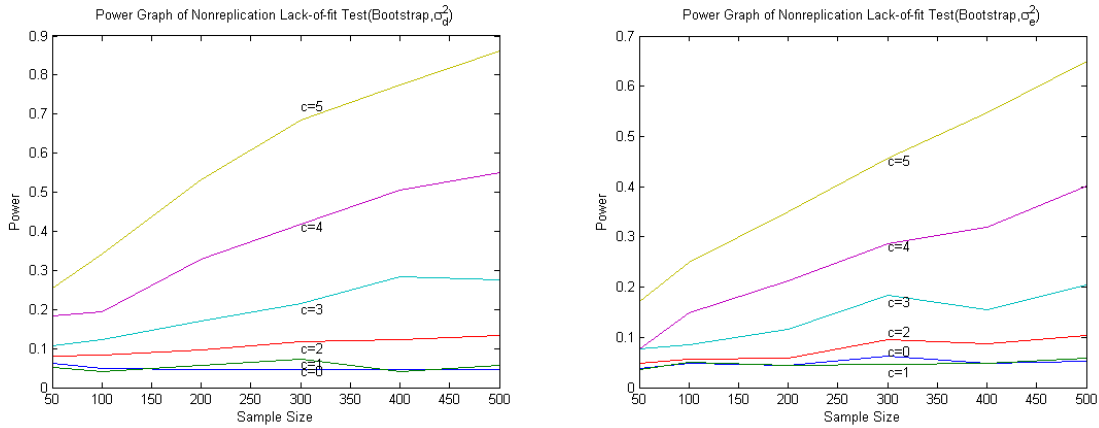


Figure 3.4: *Non-replication*

The empirical power table (Tables 3.2-3.5) and their graphs (Figure 3.3-3.4) show that the variance ratio test in non-replication case can not detect small deviation of alternative model. The power of the test obtained from the critical value using the distribution (see Figure 3.2) of the test statistic under the null directly is higher than the values using critical value from bootstrap approximation of test statistic under the null model. In case of bootstrapping case, the power values under σ_d^2 is higher than using σ_e^2 . This test can not detect small deviation of alternative model.

3.3 Reduction:

For the model considered, $p = 2, k = 1$ for the lack-of-fit test statistic with the Reduction case. Since, $\frac{(SSE_0 - SSE_a)/k}{SSE_a/(n - p - k)} \sim F(k, n - p - k)$, so the critical value for the lack-of-fit test is $F(k, n - p - k, 1 - \alpha)$ we use significance level $\alpha = 0.05$, the power values (Table 3.6) and power graph (Figure 3.5) of the test in 1000 identical trials are given follow:

c	50	100	200	300	400	500
0	0.048	0.059	0.048	0.047	0.052	0.056
1	0.090	0.113	0.178	0.269	0.340	0.391
2	0.178	0.333	0.564	0.717	0.833	0.901
3	0.367	0.613	0.890	0.981	0.992	0.998
4	0.587	0.839	0.988	1.000	1.000	1.000
5	0.776	0.958	1.000	1.000	1.000	1.000

Table 3.6: *Reduction Power Table*

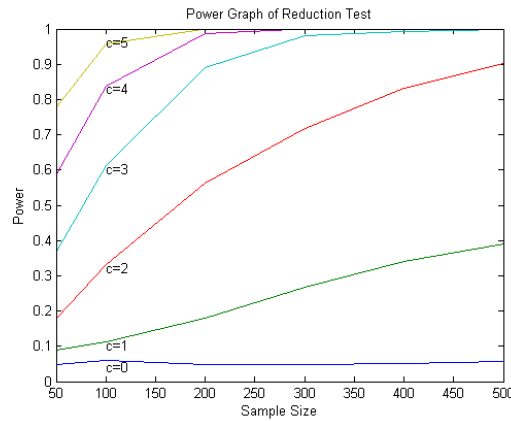


Figure 3.5: *Reduction Test Power*

Table 3.6, and the corresponding graph, Figure 3.5 say that reduction test has high power. It detects the deviation of alternative for the null even small deviation of alternative model and power grows faster as the sample size gets larger.

3.4 General von Neumann Test:

The critical values for the test statistic are calculated by the simulation method, by the Theorem 5.1[Hart⁹], and bootstrap. For the distribution of test statistic under null model, the numerator matrix is $A = (I - M)/(n - 2)$ and the denominator matrix is $B = \frac{(I - M)H(I - M)}{(2n - 2 - \text{trace}(HM))}$ where H and M matrices are discussed before(see 2.2). The critical values (Table 3.7) and distribution graph (raw and smoothed, Figure 3.6) obtained are given follow:

n	50	100	200	300	400	500
Distribution	1.2778	1.2071	1.1364	1.1162	1.0960	1.0859
Theorem 5.1	1.2326	1.1645	1.1163	1.0950	1.0822	1.0736
Bootstrap	1.3515	1.2234	1.1511	1.1156	1.0947	1.0863

Table 3.7: *General von Neumann Critical Values*

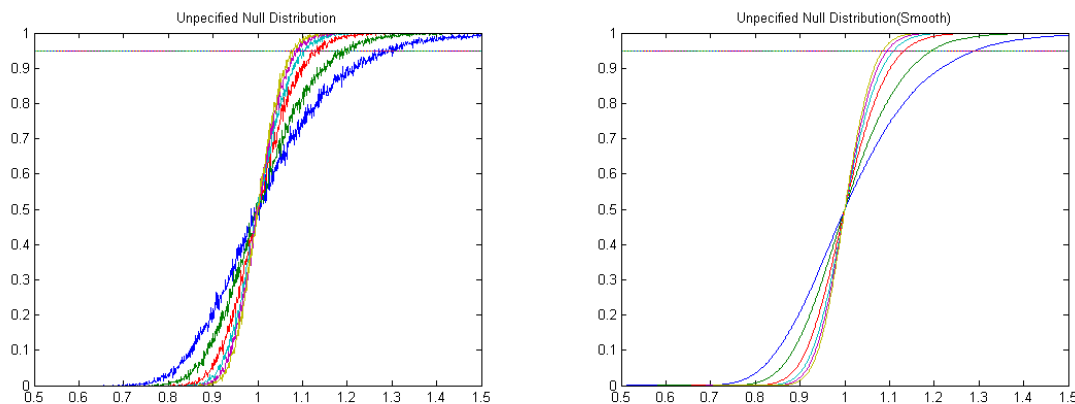


Figure 3.6: *General von Neumann Null Distribution*

Using distribution critical value

c	50	100	200	300	400	500
0	0.038	0.047	0.037	0.039	0.018	0.025
1	0.045	0.058	0.060	0.070	0.034	0.029
2	0.066	0.080	0.102	0.115	0.067	0.081
3	0.102	0.147	0.170	0.228	0.174	0.216
4	0.141	0.258	0.304	0.440	0.409	0.457
5	0.216	0.398	0.511	0.675	0.690	0.754

Table 3.8: *General von Neumann Test Power(Crit Distribution)*

c	50	100	200	300	400	500
0	0.045	0.046	0.048	0.036	0.032	0.035
1	0.059	0.059	0.039	0.051	0.052	0.040
2	0.076	0.061	0.089	0.089	0.087	0.094
3	0.118	0.142	0.141	0.181	0.226	0.247
4	0.184	0.219	0.324	0.396	0.485	0.534
5	0.251	0.369	0.505	0.629	0.741	0.821

Table 3.9: *General von Neumann Test Power(Theorem 5.1)*

c	50	100	200	300	400	500
0	0.025	0.028	0.026	0.039	0.042	0.030
1	0.040	0.037	0.043	0.036	0.056	0.042
2	0.054	0.059	0.069	0.087	0.101	0.100
3	0.071	0.098	0.130	0.169	0.232	0.256
4	0.123	0.183	0.265	0.361	0.460	0.537
5	0.171	0.295	0.463	0.619	0.757	0.825

Table 3.10: *General von Neumann Test Power(Bootstrap)*

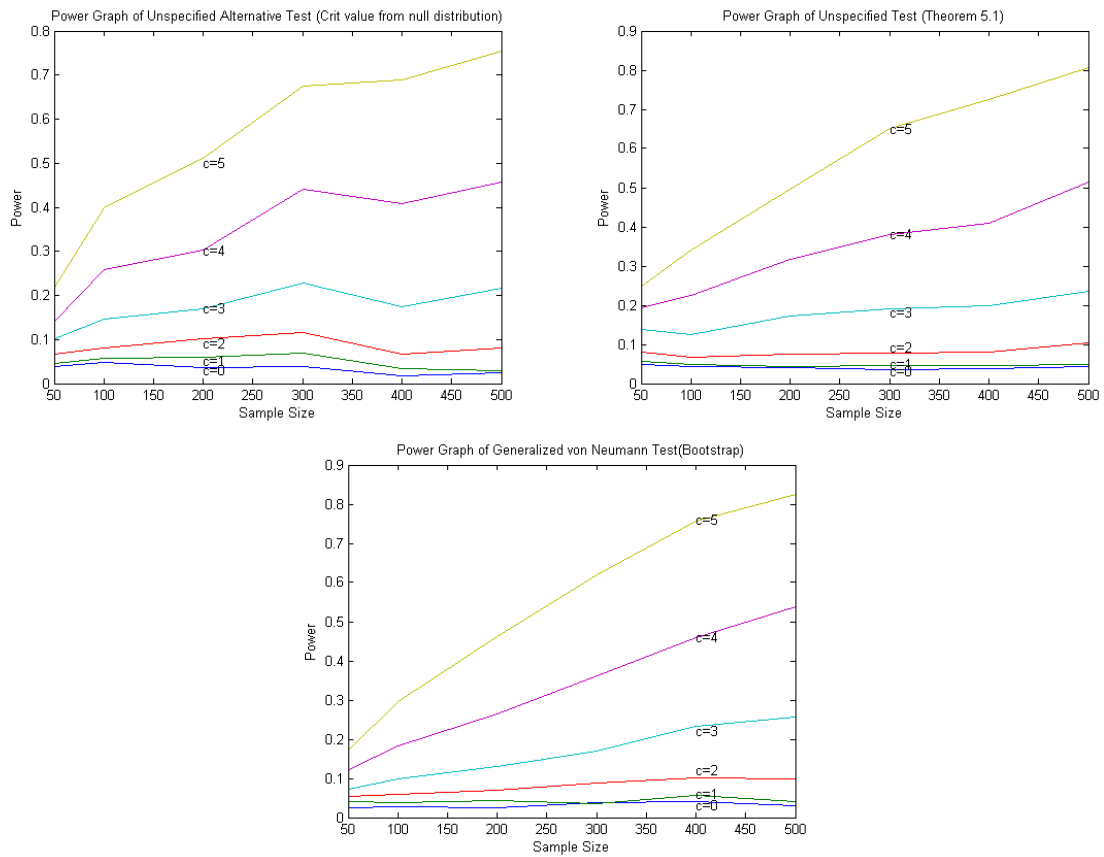


Figure 3.7: *General von Neumann Test*

The results from tables 3.8-3.10 and graph (Figure 3.7) show that the General von Neumann Test small power for smaller sample sizes and has good power for larger sample sizes when we use the critical values using the result of Theorem 5.1[Hart⁹]. There is greater probability of making wrong decision even for large sample sizes if alternative model has little deviation from the null model. In case of bootstrap critical value, the power of the test is higher than the critical values from other methods.

3.5 von Neumann and Buckley:

We have the hypothesis for these tests is

$$H_0 : r(x) = C, \text{ a constant. vs } H_a : r(x) = C + cx$$

We use the simulation algorithms(see 2.2) Method I and Method III to calculate the critical value for the tests. For the Buckley test, the numerator matrix is $A = (I - n^{-1}J_n)/(n - 1)$ and that for the von Neumann test it is $A = (I - n^{-1}J_n)Cn(I - n^{-1}J_n)$.The denominator matrices are matrices corresponding to σ_d^2 and σ_e^2 (see *Estimator Based on Pseudo-Residual*). The critical values (Table 3.11) for the test are obtained by using simulation of distribution(Distribution curve, Figure 3.8,3.9) of test statistic under null are given below:

$\sigma^2 \backslash n$	50	100	200	300	400	500
vonNeu(d)	1.2677	1.2071	1.1263	1.0960	1.1091	1.0859
vonNeu(e)	1.4293	1.2980	1.1667	1.1364	1.1465	1.1263
vonNeu(boot-d)	1.3299	1.1910	1.1289	1.1092	1.0859	1.0815
vonNeu(boot-e)	1.4316	1.2719	1.1912	1.1485	1.1236	1.1104
Buckley(d)	0.4455	0.4384	0.4313	0.4101	0.4343	0.4101
Buckley(e)	0.4384	0.4172	0.3818	0.4737	0.3394	0.4737
Buckley(boot-d)	0.5075	0.4475	0.4815	0.4581	0.4844	0.4638
Buckley(boot-e)	0.4888	0.5106	0.4973	0.4505	0.4701	0.4868

Table 3.11: *Critical Values*

von Neumann power table

c	50	100	200	300	400	500
0	0.044	0.057	0.029	0.024	0.021	0.030
1	0.159	0.193	0.238	0.332	0.359	0.385
2	0.596	0.798	0.958	0.991	0.999	1.000
3	0.953	0.996	1.000	1.000	1.000	1.000
4	0.998	1.000	1.000	1.000	1.000	1.000
5	1.000	1.000	1.000	1.000	1.000	1.000

Table 3.12: *von Neumann Power(σ_d^2)*

c	50	100	200	300	400	500
0	0.060	0.059	0.067	0.045	0.042	0.062
1	0.140	0.149	0.260	0.246	0.286	0.395
2	0.465	0.582	0.885	0.948	0.981	0.994
3	0.860	0.965	1.000	1.000	1.000	1.000
4	0.983	1.000	1.000	1.000	1.000	1.000
5	0.999	1.000	1.000	1.000	1.000	1.000

Table 3.13: *von Neumann Power(σ_e^2)*

c	50	100	200	300	400	500
0	0.034	0.049	0.051	0.042	0.052	0.045
1	0.111	0.233	0.302	0.380	0.507	0.498
2	0.549	0.815	0.972	0.997	1.000	1.000
3	0.926	0.998	1.000	1.000	1.000	1.000
4	0.997	1.000	1.000	1.000	1.000	1.000
5	1.000	1.000	1.000	1.000	1.000	1.000

Table 3.14: *von Neumann Power(Bootstrap, σ_d^2)*

c	50	100	200	300	400	500
0	0.059	0.051	0.044	0.052	0.051	0.063
1	0.129	0.174	0.175	0.277	0.295	0.380
2	0.413	0.639	0.868	0.954	0.990	0.992
3	0.814	0.978	1.000	1.000	1.000	1.000
4	0.983	1.000	1.000	1.000	1.000	1.000
5	1.000	1.000	1.000	1.000	1.000	1.000

Table 3.15: *von Neumann Power(Bootstrap, σ_e^2)*

Buckley's test power table

c	50	100	200	300	400	500
0	0.044	0.067	0.060	0.058	0.049	0.057
1	0.442	0.844	0.989	0.998	1.000	1.000
2	0.970	1.000	1.000	1.000	1.000	1.000
3	1.000	1.000	1.000	1.000	1.000	1.000
4	1.000	1.000	1.000	1.000	1.000	1.000
5	1.000	1.000	1.000	1.000	1.000	1.000

Table 3.16: *Buckely test Power(σ_d^2)*

c	50	100	200	300	400	500
0	0.050	0.067	0.057	0.054	0.061	0.060
1	0.541	0.820	0.986	0.997	1.000	1.000
2	0.971	1.000	1.000	1.000	1.000	1.000
3	1.000	1.000	1.000	1.000	1.000	1.000
4	1.000	1.000	1.000	1.000	1.000	1.000
5	1.000	1.000	1.000	1.000	1.000	1.000

Table 3.17: *Buckely test Power(σ_e^2)*

c	50	100	200	300	400	500
0	0.043	0.070	0.054	0.061	0.053	0.034
1	0.472	0.817	0.981	0.999	0.999	1.000
2	0.976	1.000	1.000	1.000	1.000	1.000
3	1.000	1.000	1.000	1.000	1.000	1.000
4	1.000	1.000	1.000	1.000	1.000	1.000
5	1.000	1.000	1.000	1.000	1.000	1.000

Table 3.18: *Buckely test Power(Bootstrap, σ_d^2)*

c	50	100	200	300	400	500
0	0.057	0.048	0.039	0.075	0.044	0.045
1	0.517	0.794	0.974	0.999	1.000	1.000
2	0.981	1.000	1.000	1.000	1.000	1.000
3	1.000	1.000	1.000	1.000	1.000	1.000
4	1.000	1.000	1.000	1.000	1.000	1.000
5	1.000	1.000	1.000	1.000	1.000	1.000

Table 3.19: *Buckely test Power(Bootstrap σ_e^2)*

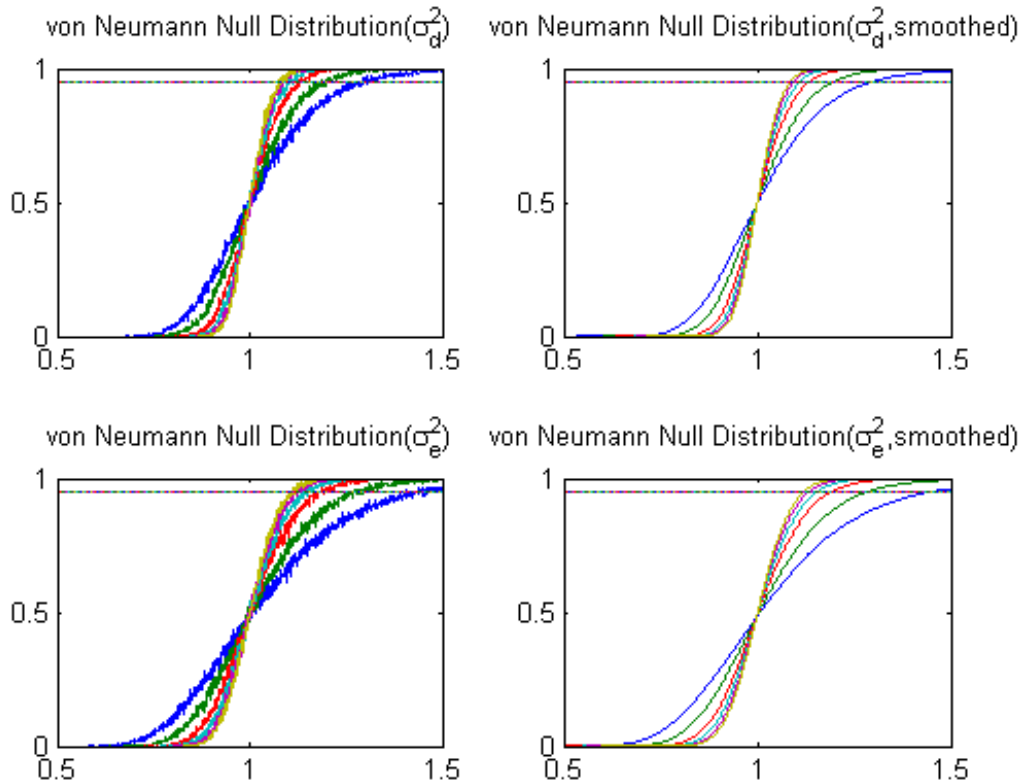


Figure 3.8: *von Neumann Null Distribution*

Empirical power of von Neumann test (Table 3.12-3.15) and that of Buckley test (Table 3.16-3.119) show that both the test have very high power in both the cases either we use σ_d^2 or σ_e^2 . Buckley Test has higher power than that of the von Neumann test. It is also observed that under the “no effect” condition in other lack-of-fit tests, Buckley test has the highest power and von Neumann test has second to Buckley test. From the null distribution curve of the Buckley test, it is clear that it has almost same critical value for all sample sizes. Thus, this test is the best for smaller sample sizes.

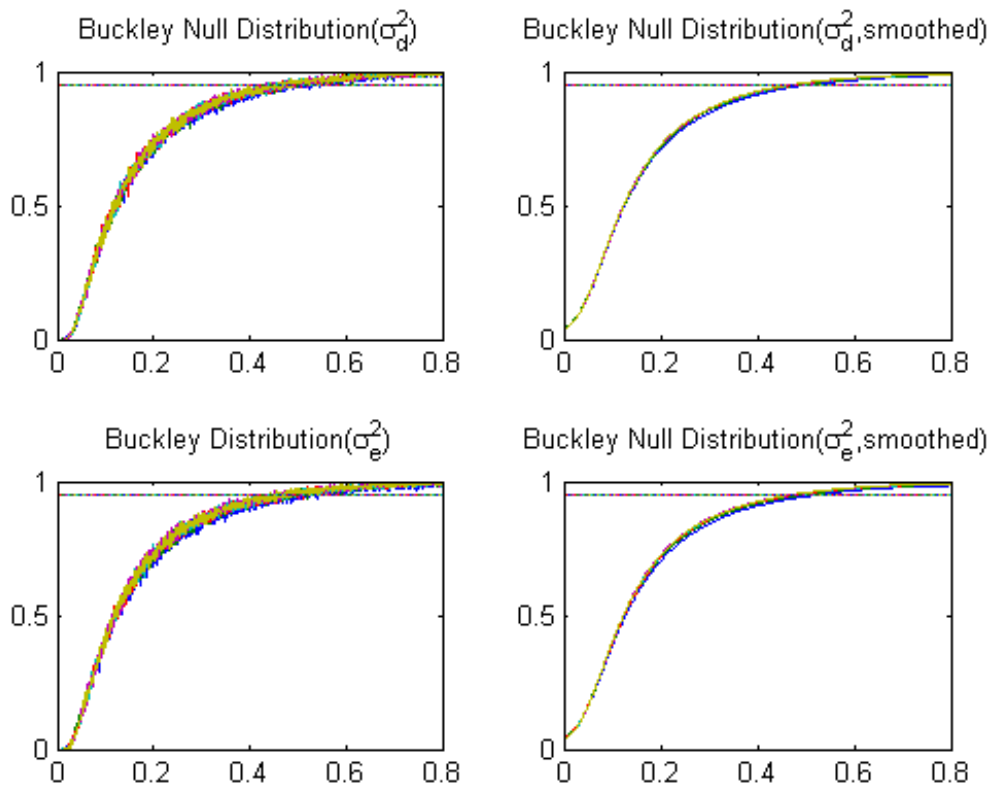


Figure 3.9: *Buckley Null Distribution*

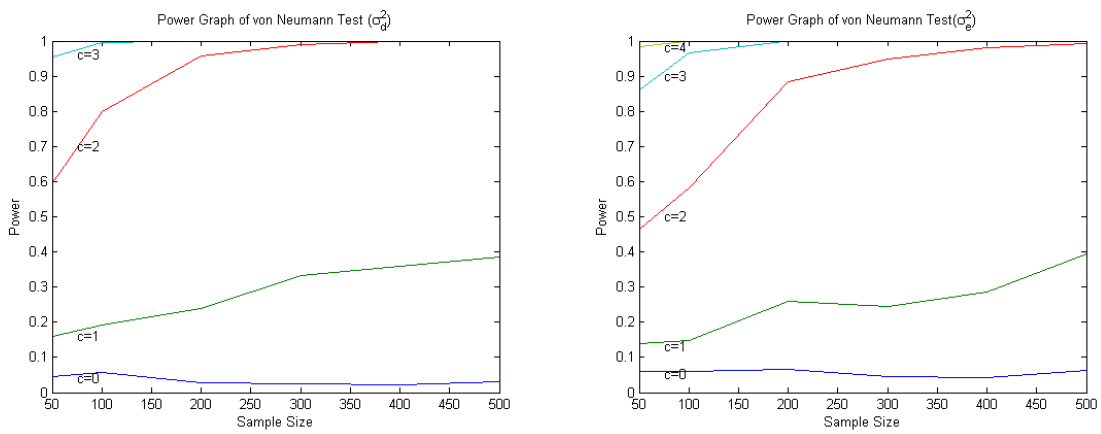


Figure 3.10: *von Neuman Test Power*

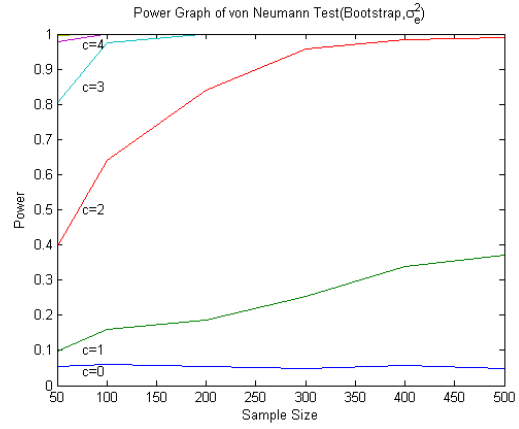
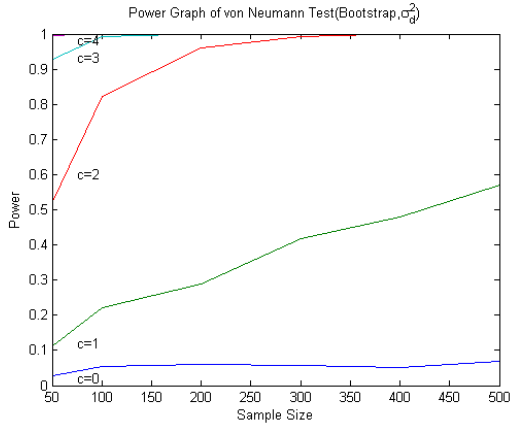


Figure 3.11: *von Neumann test power(Bootstrap)*

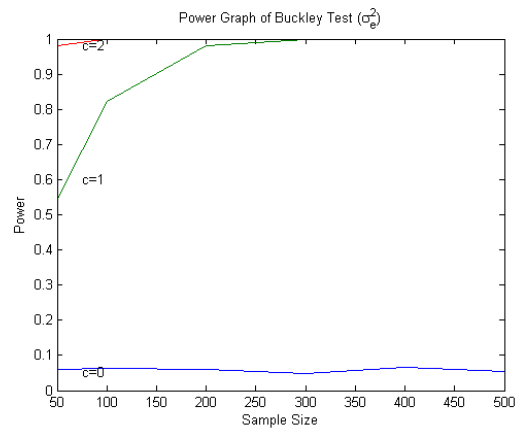
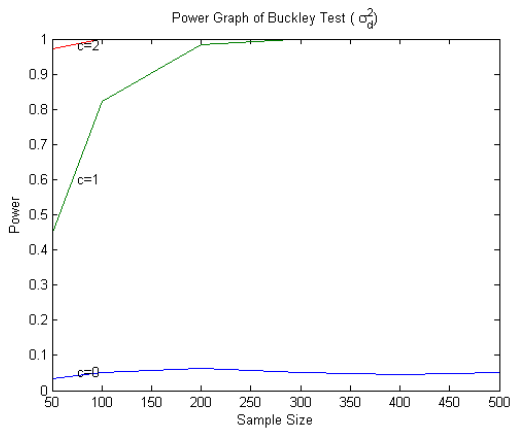


Figure 3.12: *Buckley Test Power*

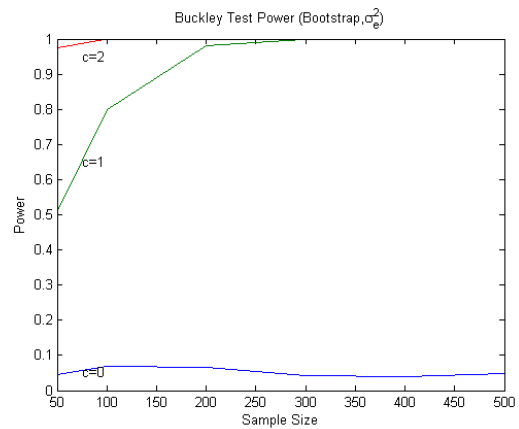
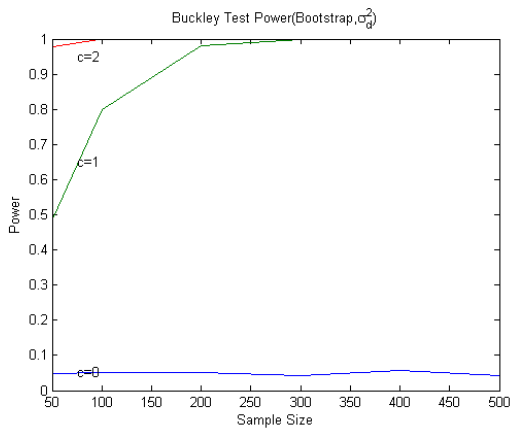


Figure 3.13: *Buckley Test Power(Bootstrap)*

3.6 Conclusion

Simulation results(section 3.1-3.6) reveal that the the Replication variance ratio test has much higher power to detect the deviation of alternative from the null model. It can make good decision even for a small sample of size 50 and $c \geq 1$ for the model selected.

In comparison to the power of lack-of-fit tests in Specified alternative non-replication case, the Reduction and the General von Neumann cases, the Reduction test has the highest power among the three. It is observed that the power of lack-of-fit test in case of specified alternative non-replication case using the critical value from the distribution of the test statistic under the null hypothesis is higher than that in the case of bootstrap critical value. It is also observed that the variance ratio, in specified alternatives non-replication case, does not have good power to make decision for smaller sample sizes and the power of test is slightly higher in case of denominator σ_d^2 than that in case of denominator σ_e^2 . It needs sample of size about 300 or more to make good decision and can make wrong decision if our c is close to 1 or in between 0 and 1. In case of the Generalized von Neumann test, the power of the test using the critical values from different method are almost consistent with each other. The graphs of empirical power of the test in case of critical values from Theorem 5.1 and bootstrap are more or less the same.

In comparison to the von Neumann test, the Buckley's test has higher power. Both of these tests have good sensitivity to detect the deviation of hypothesis from the null to the alternative. Both the σ_d^2 and σ_e^2 serve equally well for the denominator of test statistics. Also, it is seen that both the tests show same behavior regardless of whether we choose the critical value from null distribution or from the bootstrap method. There is almost no difference in case of the Buckley's test but in case of the von Neumann test, the power of the test due to σ_e^2 is a little higher than that due to σ_d^2 .

In case of replication, we don't have any tests other than the Replication case lack-of-fit test. With specified linear alternative and non-replication cases, if the null model is nested in the alternative model, we use the Reduction test. If the regression model is smooth we

can use either Non-replication lack-of-fit test or the Generalized von Neumann test. The former test needs specified alternative model but the later one does not. In case of “no effect” model, the Buckley test is considered the best.

Bibliography

- [1] M. Kendall and A. Stuart, *The Advanced Theory of Statistics*, New York: Charles Griffin & Company Ltd., 1979.
- [2] H. Chornoff, *On the distribution of the likelihood ratio*, Ann. Math. Statist.,25, 573–578, 1954.
- [3] D. R. Cox, *Further results on tests of separate families of hypotheses*, J. Roy. Statist. Soc. Ser.,B24, 406 – 424, 1962.
- [4] H. White, *Regularity conditionns for Cox’s test of non-nested hypothesis*, J. Econometr.,19, 301 – 318, 1962.
- [5] L. Pace and A. Salvan, *Best conditional tests of separate families of hypotheses*, J. Roy Statist. Soc. Ser.,B52, 125 – 134, 1990.
- [6] T. Gasser, L. Sroka, and G. Jennen-Steinmetz, *Residual variance and residual pattern in nonlinear regression*, Boimetrika,73, 625 – 633, 1986.
- [7] E. Lehmann, *Testing Statistical Hypotheses*, New York: John Wiley & Sons, 1959.
- [8] G. E. P. Box, *Some theorems on qudratic forms applied in the study of variance problems I.Effect of inequality of variance in the one-way classification*, Ann. Math. Statist.,25, 290 – 302, 1991.
- [9] J. D. Hart, *Nonparametric Smoothing and Lack-of-Fit Tests*, volume 1st Ed., Berlin; New York: Springer Series in Statistics, 1997.
- [10] J. von Neuman, *Distribution of the ratio of the mean squared successive difference of the variance*, volume 12, 367 – 395, 1941.

- [11] M. J. Buckley, *Detecting a Smooth Signal: Optimality of Cusum Based Procedures*, volume 78, Boimetrika, 1991, 253-262.

Appendix A

Matlab Codes

1. Code for Replication Variance Ratio Test

```
function freq=replication(n,r,a,b,c)
%this code gives the frequency of ratio less than the critical value and
%takes n, the sample size, r the number of replication, a, b, c are the
%parameters on the null model and p is number of parameters.
f=finv(.95,n-r,r*n-n); %gives f-value for critical value of the test.
freq=0;
for i=1:1000
    I=eye(n);
    x=(0:1/(n-1):1)';%x is column vector of n-equally space points on (0,1).
    e=normrnd(0,1,n,r);%generates n*r N(0,1) random matrix.
    y=[a+b*x+c*x.*x a+b*x+c*x.*x a+b*x+c*x.*x]+e;%random matrix of n*p.
    J=ones(n,1);
    Jr=ones(r,r);
    X=[J x x.*x];
    my=y*Jr/r;
    m=my(:,1);
    S=(y-my).*(y-my);
    SSEP=sum(sum(S'));
```

```

M=X*inv(X'*X)*X';
L=(sum((M*y)'))'/r;
sigmam2=r*(m-L)'+(m-L)/(n-r);
sigmap2=SSEP/(n*r-n);
if sigmam2/sigmap2>f;
    freq=freq+1;
else
    freq=freq;
end
end
end

```

2. Code for Power of Replication Variance Ratio Test

```

function replicationp(V,C)
%This code generates a power matrix with number of rows equal to the length of
%parameter matrix C and number of columns equal to the length of the sample
%size matrix V. The first row correspond to the first entry of C, second
%the second element of C and so on. Similarly columns correspond as
%elements of V. The entries of C are numbers on [0,1].
[m u]=size(V);%V is a row matrix of sample sizes like [5 20 30 40 50].
[r t]=size(C);
Power=zeros(t,u);%to generate power matrix corresponding to the sample sizes.
for k=1:u
    n=V(1,k);
    x=(0:1/(n-1):1)';%n equally spaced column matrix on interval (0,1).
    f=finv(.95,n-2,3*n-n);
for q=1:t
    c=C(1,q);

```

```

freqqk=0;%initializing frequency out of 1000 trials.
for i=1:1000
    e=normrnd(zeros(n,3),ones(n,3),n,3);%normal random matrix of size nx3.
    ya=[1+2*x+c*x.^2 1+2*x+c*x.^2 1+2*x+c*x.^2]+e; %ya is alternative
    %model with coefficient a and parameter c. For suitable alternative
    %of  $y=1+10*x$ , we need to choose a as an input parameter.
    J=ones(n,1);%matrix of ones of size nx1.
    J3=ones(3,3);%matrix of ones of size 3x3.
    X=[J x];%determining matrix.
    my=ya*J3/3;%mean matrix of size n*r with ith-rows of ith mean.
    m=my(:,1);%column vector of means.
    S=(ya-my).*(ya-my);%squares of deviation from means.
    SSEP=sum(sum(S'));%sum of squares of deviation from means.
    L=(sum((X*inv(X'*X)*X'*ya)'))'/3;%gives mean of rows.
    sigmam2=3*(m-L)'.*(m-L)/(n-2);
    sigmap2=SSEP/(n*3-n);
    if sigmam2/sigmap2 > f;
        freqqk=freqqk+1;
    else
        freqqk=freqqk;
    end
end
end
Power(q,k)=freqqk/1000;
end
end
Power
y1=Power(1,:);

```

```

y2=Power(2,:);
y3=Power(3,:);
y4=Power(4,:);
y5=Power(5,:);
y6=Power(6,:);
plot(V,y1,V,y2,V,y3,V,y4,V,y5,V,y6)
title('The Graph of Power of Replication Test[1+2x]')
end

```

3. Critical value code for Unspecified alternative test

```

function critunspecified(v)
m=length(v);
pr=zeros(1000,m);
P=zeros(1000,m);
crt=zeros(1,m);
crt1=zeros(1,m);
for t=1:m
    n=v(1,t);
    x=(0:1/(n-1):1)';
    I=eye(n);
    X=[ones(n,1) x];
    M=X*inv(X'*X)*X';
    I=eye(n);
    A=(I-M)/(n-2);
    H=diag([1 2*ones(1,n-2) 1])+diag(-ones(1,n-1),-1)+diag(-ones(1,n-1),1);
    B=(I-M)*H*(I-M)/(2*n-2-trace(H*M));
    U=linspace(0.5,1.5,100);
    valt=zeros(100,1);

```



```

for l=1:1000
    ul=U(1,l);
    eil=eig(A-ul*B);
    epl=eil(eil~=0);
    kl=length(epl);
    xl=chi2rnd(1,1000*kl,1);
    X=reshape(xl,1000,kl);
    Y=X*epl;
    pr(l,t)=1-sum(Y>=0)/1000;
end
k=1;
for x0=U
    valt(k)=sum(pr(:,t).*exp(-(x0-U').^2/(2*.02^2)))...
    /sum(exp(-(x0-U').^2/(2*.02^2)));
    k=k+1;
end
P(:,t)=valt;
crt(1,t)=U(min(find(pr(:,t)>.95)));
crt1(1,t)=U(min(find(P(:,t)>.95)));
end
[crt;crt1]
subplot(1,2,1);...
plot(U,pr(:,1)',U,pr(:,2)',U,pr(:,3)',U,pr(:,4)',U,pr(:,5)',U,pr(:,6)',U,.95)
title('Unpecified Null Distribution curve')
subplot(1,2,2);...
plot(U,P(:,1)',U,P(:,2)',U,P(:,3)',U,P(:,4)',U,P(:,5)',U,P(:,6)',U,.95)
title('Unpecified Smooth Null Distribution curve')

```

end

4. Power Code for Unspecified Alternative test using critical value from above code and theorem 5.1

```
function unspecifiedppthm(v,C)
u=length(v);
q=length(C);
Pp=zeros(q,u);
Pthm=zeros(q,u);
a=norminv(.95);
U=[1.2727 1.2121 1.1111 1.1111 1.0707 1.1111
    0 0 0 0 0 0];
for t=1:u
    n=v(1,t);
    U(2,t)=1+a/sqrt(n);
    x=(0:1/(n-1):1)';
    H=diag([1 2*ones(1,n-2) 1])+diag(-ones(1,n-1),1)+diag(-ones(1,n-1),-1);
    J=ones(n,1);
    X=[J x];
    M=X*inv(X'*X)*X';
    I=eye(n);
    for s=1:q
        c=C(1,s);
        freqst=0;
        freqste=0;
        for i=1:1000
            e=normrnd(0,1,n,1);
            ya=1+c*x+e;
```

```

    sigmam2=ya'*(I-M)*ya/(n-2);
    sigma2=ya'*(I-M)*H*(I-M)*ya/(2*n-2-trace(H*M));
    if sigmam2/sigma2 > U(1,t);
        freqst=freqst+1;
    else
        freqst=freqst;
    end
    if sigmam2/sigma2 > U(2,t);
        freqste=freqste+1;
    else
        freqste=freqste;
    end
end
end
Pp(s,t)=freqst/1000;
Pthm(s,t)=freqste/1000;
end
end
U
Pp
Pthm
subplot(2,1,1);plot(v,Pp(1,:),...
v,Pp(2,:),v,Pp(3,:),v,Pp(4,:),v,Pp(5,:),v,Pp(6,:))
title('Power Graph of Generalized von Neumann Test')
subplot(2,1,2);plot(v,Pthm(1,:),...
v,Pthm(2,:),v,Pthm(3,:),v,Pthm(4,:),v,Pthm(5,:),v,Pthm(6,:))
title('Power Graph of Generalized von Neumann Test Using Theorem 5.1')
end

```

5. Power code for Reduction test

```
function reductionlinp(v,C)
u=length(v);
q=length(C);
P=zeros(q,u);
for j=1:u
    n=v(1,j);
    x=(0:1/(n-1):1)';
    X=[ones(n,1) x];
    Xa=[ones(n,1) x x.^2];
    M=X*inv(X'*X)*X';
    Ma=Xa*inv(Xa'*Xa)*Xa';
    f=finv(.95,1,n-3);
    I=eye(n);
    for i=1:q
        c=C(1,i);
        freqij=0;
        for k=1:1000
            e=normrnd(0,1,n,1);
            y0=1+2*x+e;
            ya=1+2*x+c*x.^2+e;
            sigmar2=ya'*(Ma-M)*ya;
            sigmaf2=ya'*(I-Ma)*ya/(n-3);
            if sigmar2/sigmaf2 > f;
                freqij=freqij+1;
            else
                freqij=freqij;
            end
        end
    end
end
```

```

        end
    end
    P(i,j)=freqij/1000;
end
end
P
plot(v,P(1,:), v,P(2,:),v,P(3,:),v,P(4,:),v,P(5,:),v,P(6,:))
title('Power Graph of Reduction Test')

end

```

6. Code for critical value Nonreplication test[both σ_d and σ_e]

```

function critnonreplicated(v)
m=length(v);
pr=zeros(1000,m);
P=zeros(1000,m);
crt=zeros(1,m);
crt1=zeros(1,m);
prd=zeros(1000,m);
Pd=zeros(1000,m);
crt1d=zeros(1,m);
crt1d1=zeros(1,m);
for t=1:m
    n=v(1,t);
    x=(0:1/(n-1):1)';
    X=[ones(n,1) x];
    M=X*inv(X'*X)*X';
    I=eye(n);

```

```

A=(I-M)/(n-2);
M=zeros(n-2,n);
for i=1:n-2
    M(i,i)=1; M(i,i+1)=-2; M(i,i+2)=1;
end
B=M'*M/(6*n-12);
H=diag([1 2*ones(1,n-2) 1])+diag(-ones(1,n-1),-1)+diag(-ones(1,n-1),1);
Bd=H/(2*n-2);
valt=zeros(100,1);
U=linspace(.5,1.5,1000);
for l=1:1000
    ul=U(1,l);
    eil=eig(A-ul*B);
    epl=eil(eil~=0);
    kl=length(epl);
    xl=chi2rnd(1,1000*kl,1);
    Xl=reshape(xl,1000,kl);
    Y=Xl*epl;
    pr(l,t)=1-sum(Y>=0)/1000;
end
k=1;
for x0=U
    valt(k)=sum(pr(:,t).*exp(-(x0-U').^2/(2*.03^2)))...
    /sum(exp(-(x0-U').^2/(2*.03^2)));
    k=k+1;
end
P(:,t)=valt;

```

```

crt(1,t)=U(1,min(find(P(:,t)>.95)));
crt1(1,t)=U(1,min(find(pr(:,t)>.95)));
for l=1:1000
    ul=U(1,l);
    eil=eig(A-ul*Bd);
    epl=eil(eil~=0);
    kl=length(epl);
    xl=chi2rnd(1,1000*kl,1);
    Xl=reshape(xl,1000,kl);
    Y=Xl*epl;
    prd(l,t)=1-sum(Y>=0)/1000;
end
k=1;
for x0=U
    valt(k)=sum(prd(:,t).*exp(-(x0-U').^2/(2*.03^2)))/...
    /sum(exp(-(x0-U').^2/(2*.03^2)));
    k=k+1;
end
Pd(:,t)=valt;

crtcd(1,t)=U(1,min(find(Pd(:,t)>.95)));
crtcd1(1,t)=U(1,min(find(prd(:,t)>.95)));
end
[crtcd;crtcd1]
subplot(2,2,3);plot(U,prd(:,1)',U,prd(:,2)',U,prd(:,3)',
U,prd(:,4)',U,prd(:,5)',U,prd(:,6)',U,.95)
title('Nonreplication Null Distribution(\sigma_d^2)')

```

```

subplot(2,2,4);plot(U,Pd(:,1)',U,Pd(:,2)',U,Pd(:,3)',
U,Pd(:,4)',U,Pd(:,5)',U,Pd(:,6)',U,.95)
title('Nonreplication Null Distribution(\sigma_d^2 smoothed)')
[crt;crt1]
subplot(2,2,1);plot(U,pr(:,1)',U,pr(:,2)',U,pr(:,3)',...
U,pr(:,4)',U,pr(:,5)',U,pr(:,6)',U,.95)
title('Nonreplication Null Distribution(\sigma_e^2)')
subplot(2,2,2);plot(U,P(:,1)',U,P(:,2)',U,P(:,3)',...
U,P(:,4)',U,P(:,5)',U,P(:,6)',U,.95)
title('Nonreplication Null Distribution(\sigma_e^2 smoothed)')
end

```

7. Power Code for Non-replication lack-of-fit test[both σ_d^2 and σ_e^2]

```

function nonreplicationped(v,C)
u=length(v);
q=length(C);
Pd=zeros(q,u);
Pe=zeros(q,u);
U=[1.2778    1.1566    1.1061    1.1061    1.1061    1.0859
    1.4394    1.2778    1.1667    1.1667    1.1263    1.0960]
for t=1:u
    n=v(1,t);
    x=(0:1/(n-1):1)';
    J=ones(n,1);
    %H=diag([1 2*ones(1,n-2) 1])+diag(-ones(1,n-1),1)+diag(-ones(1,n-1),-1);
    X=[J x];
    M=X*inv(X'*X)*X';

```



```

I=eye(n);
D=zeros(n-2,n);
for i=1:n-2
    D(i,i)=1; D(i,i+1)=-2; D(i,i+2)=1;
end
B=D'*D/(6*n-12);
for s=1:q
    c=C(1,s);
    freqst=0;
    freqste=0;
    for i=1:1000
        e=normrnd(0,1,n,1);
        ya=1+2*x+c*x.^2+e;
        sigmae2=ya'*B*ya;
        sigmam2=ya'*(I-M)*ya/(n-2);
        %sigmad2=ya'*H*ya/(2*n-2);
        sigmad2=diff(ya)'.diff(ya)/(2*n-2);
        if sigmam2/sigmad2 > U(1,t);
            freqst=freqst+1;
        else
            freqst=freqst;
        end
        if sigmam2/sigmae2 > U(2,t);
            freqste=freqste+1;
        else
            freqste=freqste;
        end
    end
end

```

```

        end
        Pd(s,t)=freqst/1000;
        Pe(s,t)=freqste/1000;
    end
end
    Pd
    Pe
subplot(2,1,1);plot(v,Pd(1,:), v,Pd(2,:),v,Pd(3,:),...
v,Pd(4,:),v,Pd(5,:),v,Pd(6,:))
title('Power Graph of Nonreplication Test Using \sigma_d^2')
subplot(2,1,2);plot(v,Pe(1,:), v,Pe(2,:),v,Pe(3,:),...
v,Pe(4,:),v,Pe(5,:),v,Pe(6,:))
title('Power Graph of Nonreplication Test Using \sigma_e^2')
end

```

8. Critical vlaue code for von Neumann test

```

function critvonneumanned(v)
m=length(v);
pr=zeros(1000,m);
P=zeros(1000,m);
crt=zeros(1,m);
crt1=zeros(1,m);
prd=zeros(1000,m);
Pd=zeros(1000,m);
crted=zeros(1,m);
crted1=zeros(1,m);
for t=1:m

```

```

n=v(1,t);
x=(0:1/(n-1):1)';
I=eye(n);
A=(I-ones(n)/n)/(n-1);
M=zeros(n-2,n);
for i=1:n-2
    M(i,i)=1; M(i,i+1)=-2; M(i,i+2)=1;
end
B=M'*M/(6*n-12);
H=diag([1 2*ones(1,n-2) 1])+diag(-ones(1,n-1),-1)+diag(-ones(1,n-1),1);
Bd=H/(2*n-2);
valt=zeros(1000,1);
U=linspace(.5,1.5,1000);
for l=1:1000
    ul=U(1,l);
    eil=eig(A-ul*B);
    epl=eil(eil~=0);
    kl=length(epl);
    xl=chi2rnd(1,1000*kl,1);
    Xl=reshape(xl,1000,kl);
    Y=Xl*epl;
    pr(l,t)=1-sum(Y>=0)/1000;
end
k=1;
for x0=U
    valt(k)=sum(pr(:,t).*exp(-(x0-U').^2/(2*.03^2)))...
    /sum(exp(-(x0-U').^2/(2*.03^2)));
end

```

```

        k=k+1;
end
P(:,t)=valt;
crt(1,t)=U(1,min(find(P(:,t)>.95)));
crt1(1,t)=U(1,min(find(pr(:,t)>.95)));
for l=1:1000
    ul=U(1,l);
    eil=eig(A-ul*Bd);
    epl=eil(eil~=0);
    kl=length(epl);
    xl=chi2rnd(1,1000*kl,1);
    Xl=reshape(xl,1000,kl);
    Y=Xl*epl;
    prd(l,t)=1-sum(Y>=0)/1000;
end
k=1;
for x0=U
    valt(k)=sum(prd(:,t).*exp(-(x0-U').^2/(2*.02^2)))/...
        /sum(exp(-(x0-U').^2/(2*.02^2)));
    k=k+1;
end
Pd(:,t)=valt;

crt(1,t)=U(1,min(find(Pd(:,t)>.95)));
crt1(1,t)=U(1,min(find(prd(:,t)>.95)));
end
[crt; crt1]

```

```

subplot(2,2,1);plot(U,prd(:,1)',U,prd(:,2)',U,prd(:,3)',...
U,prd(:,4)',U,prd(:,5)',U,prd(:,6)',U,.95)
title('von Neumann Null Distribution(\sigma_d^2)')
subplot(2,2,2);plot(U,Pd(:,1)',U,Pd(:,2)',U,Pd(:,3)',...
U,Pd(:,4)',U,Pd(:,5)',U,Pd(:,6)',U,.95)
title('von Neumann Null Distribution(\sigma_d^2 smoothed)')
[crt;crt1]
subplot(2,2,3);plot(U,pr(:,1)',U,pr(:,2)',U,pr(:,3)',...
U,pr(:,4)',U,pr(:,5)',U,pr(:,6)',U,.95)
title('von Neumann Null Distribution(\sigma_e^2)')
subplot(2,2,4);plot(U,P(:,1)',U,P(:,2)',U,P(:,3)',...
U,P(:,4)',U,P(:,5)',U,P(:,6)',U,.95)
title('von Neumann Null Distribution(\sigma_e^2 smoothed)')
end
[function nonreplicationd(v,C)
%This code generates a power matrix with number of rows equal to the length of
%parameter matrix C and number of columns equal to the length of the sample
%size matrix V. The first row correspond to the first entry of C, second
%the second element of C and so on. Similarly columns correspond as
%elements of V. The entries of C are numbers on [0,1]. Although we are naming
%function as replication, but it non-replication case.
u=length(v);%V is a row matrix of sample sizes like [5 20 30 40 50].
t=length(C);
P=zeros(t,u);%to generate power matrix corresponding to the sample sizes.
crt=critnonreplicationd(v);
for k=1:u
    n=v(1,k);

```

```

x=(0:1/(n-1):1)';%n equally spaced comumn matrix on interval (0,1).
for q=1:t
    c=C(1,q);
    freqqk=0;%initializing frequency out of 1000 trials.
    for i=1:1000
        e=normrnd(0,.5,n,1);%normal random matrix of size nx3.
        %y=1+2*x+e;
        ya=1+2*x+c*x.^2+e;
        J=ones(n,1);
        X=[J x];
        M=X*inv(X'*X)*X';
        I=eye(n);
        A=(I-M)/(n-2);
        H=diag([1 2*ones(1,n-2) 1])+diag(-ones(1,n-1),-1)...
        +diag(-ones(1,n-1),1);
        B=H/(2*n-2);
        if ya'*A*ya/(ya'*B*ya) >crt(1,k);
            freqqk=freqqk+1;
        else
            freqqk=freqqk;
        end
    end
    end
    P(q,k)=freqqk/1000;
end
end
crt
P

```

```

plot(v,P(1,:),v,P(2,:),v,P(3,:),v,P(4,:),v,P(5,:),v,P(6,:))
title('Nonreplication Power Graph using sigmad2(nonreplicationdpd)')
end

```

```

function crt=critnonreplicationd(v)
m=length(v);
pr=zeros(1000,m);
P=zeros(1000,m);
crt=zeros(1,m);
for t=1:m
    n=v(1,t);
    x=(0:1/(n-1):1)';
    X=[ones(n,1) x];
    M=X*inv(X'*X)*X';
    I=eye(n);
    A=(I-M)/(n-2);
    H=diag([1 2*ones(1,n-2) 1])+diag(-ones(1,n-1),-1)+diag(-ones(1,n-1),1);
    B=H/(2*n-2);
    U=linspace(.5,1.6,1000);
    valt=zeros(1000,1);
    for l=1:1000
        ul=U(1,l);
        eil=eig(A-ul*B);
        epl=eil(eil~=0);
        kl=length(epl);
        xl=chi2rnd(1,1000*kl,1);
        Xl=reshape(xl,1000,kl);
    end
end

```

```

        Y=X1*ep1;
        pr(1,t)=1-sum(Y>=0)/1000;
    end
    k=1;
    for x0=U
        valt(k)=sum(pr(:,t).*exp(-(x0-U').^2/(2*.01^2)))...
            /sum(exp(-(x0-U').^2/(2*.01^2)));
        k=k+1;
    end
    P(:,t)=valt;
    crt(1,t)=U(1,min(find(P(:,t)>.95)));
end
%C=(P>.95);
%D=[U' C]
%crt

end]

```

9. Power code for von Neumann test

```

function vonneumannped(v,C)
[m u]=size(v);
[p q]=size(C);
Pp=zeros(q,u);
Pe=zeros(q,u);
U=[1.2677    1.2071    1.1263    1.0960    1.1061    1.0859
    1.4293    1.2980    1.1667    1.1364    1.1465    1.1263]
for t=1:u

```



```

n=v(1,t);
ut=U(1,t);
Cn=diag(n:-1:1);
for i=1:n
    for j=1:i
        Cn(j,i)=Cn(i,i);Cn(i,j)=Cn(i,i);
    end
end
for s=1:q
    c=C(1,s);
    freqst=0;
    freqste=0;
    for i=1:1000
        x=(0:1/(n-1):1)';
        e=normrnd(0,1,n,1);
        ya=1+c*x+e;
        J=ones(n,1);
        H=diag([1 2*ones(1,n-2) 1])+diag(-ones(1,n-1),1)...
        +diag(-ones(1,n-1),-1);
        X=[J x];
        I=eye(n);
        M=zeros(n-2,n);
        for i=1:n-2
            M(i,i)=1; M(i,i+1)=-2; M(i,i+2)=1;
        end
        B=M'*M/(6*n-12);
        sigmae2=ya'*B*ya;
    end
end

```

```

    sigmam2=ya'*(I-ones(n)/n)*ya/(n-1);
    %sigmad2=ya'*H*ya/(2*n-2);
    sigmad2=diff(ya)'\diff(ya)/(2*n-2);
    if sigmam2/sigmad2 > U(1,t);
        freqst=freqst+1;
    else
        freqst=freqst;
    end
    if sigmam2/sigmae2 > U(2,t);
        freqste=freqste+1;
    else
        freqste=freqste;
    end
end
end
Pd(s,t)=freqst/1000;
Pe(s,t)=freqste/1000;
end
end
Pd
Pe
subplot(2,1,1);plot(v,Pd(1,:), v,Pd(2,:),v,Pd(3,:),...
v,Pd(4,:),v,Pd(5,:),v,Pd(6,:))
title('Power Graph of von Neumann Test Using \sigma_d^2')
subplot(2,1,2);plot(v,Pe(1,:), v,Pe(2,:),v,...
Pe(3,:),v,Pe(4,:),v,Pe(5,:),v,Pe(6,:))
title('Power Graph of von Neumann Test Using \sigma_e^2')
end

```

10. Critical value code for Buckley test

```
function critBuckleyed(v)
m=length(v);
pr=zeros(1000,m);
P=zeros(1000,m);
crt=zeros(1,m);
crt1=zeros(1,m);
prd=zeros(1000,m);
Pd=zeros(1000,m);
crt1d=zeros(1,m);
crt1d1=zeros(1,m);
for t=1:m
    n=v(1,t);
    x=(0:1/(n-1):1)';
    X=[ones(n,1) x];
    M=X*inv(X'*X)*X';
    I=eye(n);
    Cn=diag(n:-1:1);
    valt=zeros(1000,1);
    for i=1:n
        for j=1:i
            Cn(j,i)=Cn(i,i);Cn(i,j)=Cn(i,i);
        end
    end
    A=(I-ones(n)/n)*Cn*(I-ones(n)/n)/n^2;
    M=zeros(n-2,n);
    for i=1:n-2
```

```

    M(i,i)=1; M(i,i+1)=-2; M(i,i+2)=1;
end
B=M'*M/(6*n-12);
H=diag([1 2*ones(1,n-2) 1])+diag(-ones(1,n-1),-1)+diag(-ones(1,n-1),1);
Bd=H/(2*n-2);
valt=zeros(1000,1);
U=linspace(0,.7,1000);
for l=1:1000
    ul=U(1,l);
    eil=eig(A-ul*B);
    epl=eil(eil~=0);
    kl=length(epl);
    xl=chi2rnd(1,1000*kl,1);
    Xl=reshape(xl,1000,kl);
    Y=Xl*epl;
    pr(l,t)=1-sum(Y>=0)/1000;
end
k=1;
for x0=U
    valt(k)=sum(pr(:,t).*exp(-(x0-U').^2/(2*.04^2)))/...
    /sum(exp(-(x0-U').^2/(2*.04^2)));
    k=k+1;
end
P(:,t)=valt;
crt(1,t)=U(1,min(find(P(:,t)>.95)));
crt1(1,t)=U(1,min(find(pr(:,t)>.95)));
for l=1:1000

```

```

    ul=U(1,1);
    eil=eig(A-ul*Bd);
    epl=eil(eil~=0);
    kl=length(epl);
    x1=chi2rnd(1,1000*kl,1);
    X1=reshape(x1,1000,kl);
    Y=X1*epl;
    prd(1,t)=1-sum(Y>=0)/1000;
end
k=1;
for x0=U
    valt(k)=sum(prd(:,t).*exp(-(x0-U').^2/(2*.04^2)))...
    /sum(exp(-(x0-U').^2/(2*.04^2)));
    k=k+1;
end
Pd(:,t)=valt;

crt(1,t)=U(1,min(find(Pd(:,t)>.95)));
crt1(1,t)=U(1,min(find(prd(:,t)>.95)));
end
[crt;crt1]
subplot(2,2,1);plot(U,pr(:,1)',U,pr(:,2)',U,pr(:,3)',U,...
pr(:,4)',U,pr(:,5)',U,pr(:,6)',U,.95)
title('Buckley Distribution(\sigma_e^2)')
subplot(2,2,2);plot(U,P(:,1)',U,P(:,2)',U,P(:,3)',...
U,P(:,4)',U,P(:,5)',U,P(:,6)',U,.95)
title('Buckley Null Distribution(\sigma_e^2 smoothed)')

```

```

[crtid;crtid1]
subplot(2,2,3);plot(U,prd(:,1)',U,prd(:,2)',U,prd(:,3)',...
U,prd(:,4)',U,prd(:,5)',U,prd(:,6)',U,.95)
title('Buckley Null Distribution(\sigma_d^2)')
subplot(2,2,4);plot(U,Pd(:,1)',U,Pd(:,2)',U,Pd(:,3)',U,...
Pd(:,4)',U,Pd(:,5)',U,Pd(:,6)',U,.95)
title('Buckley Null Distribution(\sigma_d^2 smoothed')
end

```

11. Power code for Buckley test

```

function Buckleyped(v,C)
u=length(v);
q=length(C);
Pp=zeros(q,u);
Pe=zeros(q,u);
%U=[.4798 .4495 .4747 .4596 .4343 .4394];
U=[0.4455    0.4384    0.4313    0.4101    0.4313    0.4101
    0.4384    0.4172    0.3818    0.4737    0.3394    0.4737]
for t=1:u
    n=v(1,t);
    ut=U(1,t);
    Cn=diag(n:-1:1);
    for i=1:n
        for j=1:i
            Cn(j,i)=Cn(i,i);Cn(i,j)=Cn(i,i);
        end
    end
end
for s=1:q

```

```

c=C(1,s);
freqst=0;
freqste=0;
for i=1:1000
    x=(0:1/(n-1):1)';
    e=normrnd(0,1,n,1);
    ya=1+c*x+e;
    J=ones(n,1);
    H=diag([1 2*ones(1,n-2) 1])+diag(-ones(1,n-1),1)...
    +diag(-ones(1,n-1),-1);
    X=[J x];
    I=eye(n);
    M=zeros(n-2,n);
    for i=1:n-2
        M(i,i)=1; M(i,i+1)=-2; M(i,i+2)=1;
    end
    B=M'*M/(6*n-12);
    sigmae2=ya'*B*ya;
    sigmam2=(cumsum((I-ones(n)/n)*ya))'*(cumsum(I-ones(n)/n)*ya)/n^2;
    %sigmad2=ya'*H*ya/(2*n-2);
    sigmad2=diff(ya)'*diff(ya)/(2*n-2);
    if sigmam2/sigmad2 > U(1,t);
        freqst=freqst+1;
    else
        freqst=freqst;
    end
    if sigmam2/sigmae2 > U(2,t);

```

```

        freqste=freqste+1;
    else
        freqste=freqste;
    end
end
end
Pd(s,t)=freqst/1000;
Pe(s,t)=freqste/1000;
end
end
Pd
Pe
subplot(2,1,1);plot(v,Pd(1,:), v,Pd(2,:),v,Pd(3,:),v,Pd(4,:),v,Pd(5,:),v,Pd(6,:))
title('Power Graph of Buckley Test Using \sigma_d^2')
subplot(2,1,2);plot(v,Pe(1,:), v,Pe(2,:),v,Pe(3,:),v,Pe(4,:),v,Pe(5,:),v,Pe(6,:))
title('Power Graph of Buckley Test Using \sigma_e^2')
end

```

12. Power code for General von Neumann test using bootstrap

```

function generalvonnbootstrapp(v,C)
u=length(v);
q=length(C);
Pp=zeros(q,u);
crt=zeros(1,u);
for t=1:u
    n=v(1,t);
    x=(0:1/(n-1):1)';
    H=diag([1 2*ones(1,n-2) 1])+diag(-ones(1,n-1),1)+diag(-ones(1,n-1),-1);
    J=ones(n,1);

```



```

X=[J x];
M=X*inv(X'*X)*X';
I=eye(n);
y=1+2*x+normrnd(0,1,n,1);
e=(eye(n)-M)*y;
mt=bootstrp(1000,@(u)[((u'*u*(2*n-2-trace(H*M)))/(u'*H*u*(n-2))),e);
D=sort(mt);
%pr(:,t)=D;
crt(1,t)=D(950);
for s=1:q
    c=C(1,s);
    freqst=0;
    for i=1:1000
        ya=1+2*x+c*x.^2+normrnd(0,1,n,1);
        sigmam2=ya'*(I-M)*ya/(n-2);
        sigma2=ya'*(I-M)*H*(I-M)*ya/(2*n-2-trace(H*M));
        if sigmam2/sigma2> crt(1,t);
            freqst=freqst+1;
        else
            freqst=freqst;
        end
    end
    end
    Pp(s,t)=freqst/1000;
end
end
crt

```

```

Pp
plot(v,Pp(1,:), v,Pp(2,:),v,Pp(3,:),v,Pp(4,:),v,Pp(5,:),v,Pp(6,:))
title('Power Graph of Generalized von Neumann Test(Bootstrap)')
xlabel('Sample Size'),ylabel('Power')
end

```

13. von Neumann test code using bootstrap(σ_d^2)

```

function vonneumannbootstrapp(v,C,nb,a)
u=length(v);
q=length(C);
P=zeros(q,u);
crt=zeros(1,u);
for t=1:u
    n=v(1,t);
    x=(0:1/(n-1):1)';
    H=diag([1 2*ones(1,n-2) 1])+diag(-ones(1,n-1),1)+diag(-ones(1,n-1),-1);
    J=ones(n,1);
    y=1+normrnd(0,1,n,1);
    mt=bootstrp(nb,@(u)[2*(u'*(eye(n)-ones(n)/n)*u)/(u'*H*u)],y);
    D=sort(mt);
    crt(1,t)=D((1-a)*nb);
    for j=1:q
        Mt=zeros(1,nb);
        for l=1:nb
            yal=1+normrnd(0,1,n,1)+C(1,j)*x;
            Mt(1,l)=2*(yal'*(eye(n)-ones(n)/n)*yal)/(yal'*H*yal);
        end
        P(j,t)=sum(Mt>crt(1,t))/nb;
    end
end

```

```
end
```

```
end
```

```
plot(v,P(1,:),v,P(2,:),v,P(3,:),v,P(4,:),v,P(5,:),v,P(6,:))
```

```
title('Power Graph of von Neumann Test using Bootstrap')
```

```
xlabel('Sample Size'),ylabel('Power')
```

```
crt
```

```
[v;P]
```

```
end
```

14. Buckley test code using bootstrap(σ_d^2)

```
function Buckleybootstrapp(v,C)
```

```
u=length(v);
```

```
q=length(C);
```

```
P=zeros(q,u);
```

```
crt=zeros(1,u);
```

```
for t=1:u
```

```
    n=v(1,t);
```

```
    x=(0:1/(n-1):1)';
```

```
    H=diag([1 2*ones(1,n-2) 1])+diag(-ones(1,n-1),1)+diag(-ones(1,n-1),-1);
```

```
    J=ones(n,1);
```

```
    y=1+normrnd(0,1,n,1);
```

```
    Cn=diag(n:-1:1);
```

```
    I=eye(n);
```

```
    Jn=ones(n);
```

```
    for i=1:n
```

```
        for j=1:i
```

```
            Cn(j,i)=Cn(i,i);Cn(i,j)=Cn(i,i);
```

```

        end

    end

    mt=bootstrp(1000,@(u) [2*(n-1)*u'*(I-Jn/n)*Cn*(I-Jn/n)*u/(n^2*u'*H*u)],y);
    D=sort(mt);
%    [fi xi]=ksdensity(D);
%    plot(xi,fi)
%    hold on
    crt(1,t)=D(950);
    for j=1:q
        Mt=zeros(1,1000);
        for l=1:1000
            yal=1+normrnd(0,1,n,1)+C(1,j)*x;
            Mt(1,l)=2*(n-1)*(yal'*(I-Jn/n)*Cn*(I-Jn/n)*yal)/(n^2*yal'*H*yal);
        end
        P(j,t)=sum(Mt>crt(1,t))/1000;
    end

end

end

hold off

plot(v,P(1,:),v,P(2,:),v,P(3,:),v,P(4,:),v,P(5,:),v,P(6,:))
title('Buckley Test Power Using Bootstrap')
xlabel('Sample Size'),ylabel('Power')

crt
[v;P]

end

```