COST EFFECTIVE ANALYSIS OF BIG DATA

By

STEVEN J. LITZSINGER

NEHAD DABABO

A Senior Project submitted

in partial fulfillment

of the requirements for the degree of

Bachelor of Science in Industrial Engineering

California Polytechnic State University

San Luis Obispo

Graded by: ________________________________ Date of Submission: _________________

Checked by: _______________________________ Approved by: _______________________

Table of Contents

Table of Figures

Executive Summary

Big data is everywhere and businesses that can access and analyze it have a huge advantage over those who can’t. One option for leveraging big data to make more informed decisions is to hire a big data consulting company to take over the entire project. This method requires the least effort, but is also the least cost effective. The problem is that the know-how for starting a big data project is not commonly known and the consulting alternative is not very cost effective. This creates the need for a cost effective approach that businesses can use to start and manage big data projects. This report details the development of an advisory tool to cut down on consulting costs of big data projects by taking an active role in the project yourself. The tool is not a set of standard operating procedures, but simply a guide for someone to follow when embarking on a big data project. The advisory tools has three steps that consist of data wrangling, statistical analysis, and data engineering.

Data wrangling is the process of cleaning and organizing data into a format that is ready for statistical analysis. The guide recommends using the open source software and programming language of R. The next step is the statistical analysis portion of the process which takes the form of exploratory data analysis and the use of existing models and algorithms. The use of existing methods should always be attempted to the highest performance before justifying the costs to pay for big data analytics and the development of new algorithms. Data engineering consists of creating and applying statistical algorithms, utilizing cloud infrastructure to distribute processing, and the development of a complete platform solution.

The experimentation for the design of our advisory toolwas carried out through analysis of many large data sets. The data sets were analyzed to determine the best explanatory variables to predict a selected response. The iterative process of data wrangling, statistical analysis, and

model building was carried out for all the data sets. The experience gained, through the iterations of data wrangling and exploratory analysis, was extremely valuable in evaluating the usefulness of the design. The statistical analysis improved every time the iterative loop of wrangling and analysis was navigated.

In house data wrangling, before submission to a data scientist, is the primary cost justification of using the advisory tool. Data wrangling typically occupies 80% of data scientist's time in big data projects. So, if data wrangling is self-performed before a data scientist receives the data, then less time will be spent wrangling by the data scientist. Since data scientists are paid very high hourly wages, extra time saved wrangling equates to direct cost savings. This is assuming that the data wrangling performed before a data scientist takes over is of adequate quality.

The results of applying the advisory tool may vary from case to case, depending on the critical skills the user possesses and the development of such skills. The critical skills begin with coding in R and Python as well as knowledge in the statistical methods of choice. Basic knowledge of statistics, and any programming language is a must to begin utilizing this guide. Statistical proficiency is the limiting factor in the advisory tool. The best start for doing a big data project on one's own is to first learn R and become familiar with the statistical libraries it contains. This allows data wrangling and exploratory analysis to be performed at a high level. This project pushed the boundaries of what can be done with big data using traditional computer framework without cloud usage. Storage and processing limits of traditional computers were tested and in some cases reached, which verified the eventual need to operate in the cloud environment.

Acknowledgements

We would like to thank our professors, advisors, and colleagues from California Polytechnic State University (Cal Poly) who provided insight and expertise to help this project.

We thank Dr. Reza Pouraghabagher, a professor in the Industrial and Manufacturing Engineering department at Cal Poly, for being our technical advisor and aiding us in the development of this project. We could not have done it without him.

We would also like to thank Dr. Beth Chance, a professor in the Statistics department at Cal Poly, for her wisdom and guidance in our statistical analysis.

We would also like to show our gratitude to our colleague Ken Kauffman, a student in the Industrial and Manufacturing Engineering department at Cal Poly, for engaging in many helpful discussions and sharing his experiences with big data and data wrangling.

Introduction

Big data is a frequently used buzzword, but it’s not just hype. Big data is everywhere and businesses that can access and analyze it have a huge advantage over those who can’t. The fact is that big data is changing the way that business decisions are made and the way that enterprises are managed. The gap between those who use data efficiently and those who don’t is growing and no one wants to be on the wrong side of the gap. So, what does it take to leverage big data and begin to make more informed data-driven decisions?

One option is to hire a big data consulting company to take over the entire project for you. This option is the one that requires the least effort on your part it, but it is far from the most cost effective approach. These big data consulting companies charge hefty fees for their services and many businesses can’t afford them. Therefore there is a need for a way for businesses to approach big data projects in a cost effective manner.

The problem is that the knowledge of what it takes to begin your own big data project is not commonly known and the consulting alternatives are not cost effective. To solve this problem the following objectives must be achieved:

- Understand how big data consulting companies carry out big data projects and the costs of such projects
- Carry out big data analysis
- Apply findings from research and analysis to make recommendations for those who wish to carry out their own big data projects

First big data consulting companies will be researched to determine the methods and costs of big data projects. This will allow for a thorough understanding of what a big data project entails and what it costs to outsource the entire project. After an understanding of big data projects is reached, the next step is to carry out analysis of large data sets. This will take the form for multiple regression and the selection of variables for prediction models. This will give practical experience and allow for recommendations to be made for starting your own big data project. The completion of these tasks will lead to an advisory tool for starting and managing your own big data projects in a cost effective manner.

The goal of the advisory tool it to cut down on consulting costs of big data projects by taking an active role in the project yourself. The tool is not a set of standard operating procedures, but simply a guide for someone to follow when embarking on a big data project. The tool will not get rid of the need for data science professionals, but it will allow the user to effectively use these professionals in a cost effective manner. Since each big data project is unique the tool will not make specific recommendations, but instead focus on educating the user to the available methods and tools so they can make more informed decisions for their project.

Background

Variable selection is the first step in creating a multiple regression model and it is often the most difficult. It is also the most important because you are selecting the independent explanatory variables that will best predict the dependent variable or response. In the selection process we want to choose variables that have the strongest association with the response variable. Scatterplots and added variable plots are two graphical methods that can be used to evaluate strength of association between explanatory and response variables. However our modern day software JMP, Minitab, SAS, R and others can effectively and efficiently recommend variables for the model. With the large amount of available variables in big data there are an even larger amount of models that can be created from any number of the variables. One can't possibly hope to explore all possible models, especially in big data, so variable selection techniques provide a way to explore some of the possible models.

Three of these screening techniques include forward selection, backward elimination, and stepwise regression. Forward selection adds variables to the model one at a time until the fit is no longer significantly improved by any more additions. Criteria for selection can be based on optimizing a model measure criterion or from a hypothesis test that produces a p-value (strength of evidence) to compare to a predetermined level of significance. A .05 level of significance (alpha or α) will be used for the entirety of the project. This means that we accept a 5% chance of making a type 1 error, which is incorrectly rejecting a null hypothesis. A null hypothesis states a relationship between two measured phenomena, and is compared to an alternative hypothesis. Hypothesis testing is meant to assess the strength of evidence (p-value) against the null hypotheses. The null hypothesis can only be rejected or deemed plausible, but it can never be accepted. If we reject the null hypothesis we are acting in favor of the alternative hypothesis

The criteria for variable selection used in this project will be based on hypothesis testing, because it is required for backward elimination and stepwise regression. The p-value is obtained by a partial F-test which evaluates the change in variability explained by the model, while taking into consideration the different number of explanatory variables present. Backward elimination uses the same partial F-test, but instead begins will all the variables in the model. It then proceeds to remove the variables making the smallest contribution, one by one, until all the variables that remain are significant to the model. A variable is significant to the model if the p-value of the partial F-test (for that variable) is less than the level of significance. Stepwise regression applies both of the methods iteratively, one after the other, until a best subset for the model is found. This method evaluates, over and over again, if the earlier variables are still needed in the model.

These techniques offer up best subsets of independent variables. However, it is not unusual for several subsets to be found that are equally "good," and thus additional considerations must be made when selecting the subset to use in a model. It truly is a balancing act because the subset should be small enough so that maintenance costs and analysis are feasible, but also large enough so that acceptable prediction is attained [1]. The selection process should be subjective to the analyst's experience, judgment, and knowledge of the topic being studied. In addition to this, good regression model selection should follow the principle of parsimony which states, "A model should contain the smallest number of variables necessary to fit the data" [2]. If there are still choices between models at this point then additional criterion should be considered to reach your desired end result. This will vary depending if you goal is prediction or parameter estimates, among other things.

Once the explanatory variables are selected we run the model and check if the model assumptions are met. The model assumptions for linear regression are linearity of the regression function, normal distribution of the prediction errors, equal variance of the prediction errors, and independence of the error terms (in time sequenced data). In addition to the normal linear regression assumptions, multiple regression has the additional assumptions of no autocorrelation and little or no multicollinearity. (See Table of Terms explained at the end of the literature review) Recalling that we live in a world of variability these assumptions are almost never exactly true. Therefore we must assess the degree to which the assumptions are violated, the nature of the violations, and take corrective action if needed [3]. Corrective action takes the form of various transformations that include log and power transformations of both the response variable and/or the explanatory variable(s). Making transformations to meet the model assumptions is often an iterative process where many different things will be tried separately, and together. Often a combined solution of these efforts will lead to the model assumptions being met. If the assumptions of multiple linear regression absolutely cannot be met then non-parametric methods, such as ridge regression and distribution-free rank methods, should be considered instead.

Literature Review

Progressive Sampling

Progressive sampling is a process in which the sample size of analysis is increased until the accuracy of the model is maintained. Progressive sampling is based on the idea that sub-sampling the data can give the same accuracy of the whole data set at much lower computational costs [4]. This process is especially useful in big data because of the high costs of storing and processing the large amounts of data. Progressive sampling was used in this project to study variable selection trends as the sub-sample size was increased. To use progressive sampling, two fundamental aspects must be determined. A sampling schedule must be determined and convergence must be detected [5].

A sampling schedule can take many forms, arithmetic or geometric for example, and determines the size and increasing increments of the samples that will be tested. Geometric sampling was used in this project. Convergence in progressive sampling is a phenomenon where the stability of results is reached. In this project convergence takes the form of consistent variable selection. This means that the variables selected for the model are the same across different replicates of a given sub-sample size.

Data Analytic Costs and Alternatives

Data analytic consulting organizations today charge their clients for big data modeled solutions with respect to their specific needs, regardless of quantitative aspects of pricing. With the predicted growth in big data technology and services rising at a 27% compound annual growth rate to $32.4 billion through 2017 as forecasted by IDC [7], it’s important for these organizations to focus on high-quality solutions that will return growth in revenue. To retain high-quality solutions that will bring back their evangelist clientele along with inspired affiliates, questioning whether or not additional observations or explanatory variables doesn't revolve around pricing but rather the quality either variable adds in value to the solution they will deliver. Putting aside all the skills and time spent creating an accurate big data model, it could be difficult adding marginal cost for more explanatory variables or observations if the model doesn't serve useful to a client and in most cases clientele refusing to pay for time when results do not produce larger revenue. [8]

The marginal costs for additional variables or observations can vary individually for a big data project. To understand the pricing scheme better, big data projects could be scene as 3-step process to produce information that clients will see valuable. There are a lot of organizations out there today that may define these three steps in their own way to better correlate their trademark with the final product, but they all follow a general method. The first step involves establishing a scope and work breakdown structure, as well as gathering and organizing the data. The second step is the process is finding the variables necessary to answer to the problem statement, and finally creating an algorithm to carry out the statistical analysis. Within the second step architecture is created to enhance the carrying out of the algorithms and make the processing more efficient. The third step is creating the platform tool to provide visualization of the analysis and reporting of important metrics and results.

Renat Khasanshyn, the CEO and Founder of Altoros Systems in Sunnyvale California, explained in a phone interview [9], his company's 3-step process and the associated costs. The big data projects they do could vary in cost from $20,000 to $200,000 depending on the size and scope of the project. Usually they have an architecture engineer and data scientist working throughout the longevity of the 3-step process of a project where they are each paid around $150 -$200 per hour depending on their level of experience. As mentioned before their three step process is very similar, but to be specific with pricing, here is how Renat Khasanshyn described his three step process while including work durations and costs:

1. Scope of the project and defining the architecture (3-5 days) is the process of validating the project, creating a proof of concept with a real data set ($3,000-$6,000).
2. Implementation of architecture is where they define queries to navigate through the data pipeline, which includes accessing the data, running the algorithms, and sending the analysis results to the front end platform tool. That process usually takes around 1-2 months ($30,000-$60,000).
3. Implementation of production: bringing the tool into production, fine-tuning, making the tool user friendly so that users can add more data to further analyze their progress. The workload duration will be based on the time spent in step 2, which also will determine the costs for step 3 ($30,000-$60,000).

Step 2 of this process is usually the bulk of the project and will dominate the definition of costs to a client. An approach at cost optimization for this step process has been examined by Gary M. Weiss and Ye Tian in their article, “Maximizing classifier utility when there are data acquisition and modeling costs” [6]. It describes an optimization methodology that uses progressive sampling and cost optimization functions. It takes into account the major costs of acquiring and analyzing big data and compares the costs to the learning gained from the model. This optimization ideology is fundamental to addressing our goals in the optimization of big data regression analysis.

Big data projects are not done equally and the cost of big data projects are quite expensive for smaller businesses that don’t necessarily need large and expensive analytics to return information that could be of use to them. In general, the use of smaller data analytics should always be attempted to the highest performance before justifying the costs to pay for big data analytics. The tools and methods in the data science industry today are of reasonable pricing and can help businesses organize and make sense of the data they have.

Tools like Google Analytics have free data analytic starting packages that display metrics on website traffic for small business. Google Analytics can extract and use long-term data pertaining to visitor behavior and social media traffic to create trends and other vital information that could lead to data-driven business decisions [10]. Google Analytics is free, and through use of the software, Google can eventually collect enough information to gain insight for other premium tools that will manipulate data to highlight potential improvements for a company and for the premium package costs are incurred but they are minimal in comparison to the costs for big data.

Another source for data analytics that smaller businesses can prosper from is IBM’s Watson Analytics. By taking the time to understand the user, their services focus on illustrating the key driving factors that will give better data and predictive models that users can use to anticipate trends that will help businesses lead to better data-driven decisions. To stay competitive, IBM offers their services for free with storage limitations for flat files of up to 100,000 rows, 50 columns, and a 500 MB in size [11]. According to IBM, all of the functionality in the free version is the same as what is offered in the premium version with the only difference being storage available, however after 30 days of the free usage, IBM does require a credit card from its users to hold even though they will not charge the users [11]. For more advanced users that know how to code in languages like Ruby and Java, the use of Watson Analytics can be performed on IBM's free platform as a service (PaaS), known as Bluemix, to use free services like: Message Resonance, Q&A, User Modeling, Visualization Rendering, Concept Expansion, Language Identification, Machine Translation, and Relationship Extraction [12].

An online platform most commonly used today by many small businesses due to their infamous impact (in industry) in data storage and in the big data industry is Amazon Web Services (AWS). Rather than hiring a data analytics company to do a big data project, AWS provides all the tools necessary for every step in the process from collecting, to storing, analyzing, programing and more [13]. AWS is familiar with various file types and thus makes file conversions a breeze and allow users to quickly move into AWS platforms they've designed. To help users getting started, AWS offers tutorials that include practice assignments, and even provide walk through case study examples. AWS's general concept for payment is to pay only for the storage and processing users actually use with no minimum fees or setup costs [14]. Between all the data analytic services offered out there, AWS creates the possibility for average businesses to perform independent small scale big data projects at affordable rates in comparison to using a big data consulting firm [15]. Using services like AWS with the design we will propose will enable users to get the most out of their big data projects for the best price.

Cloud Computing Infrastructure

AWS architecture uses frameworks such as Hadoop and NoSQL to run their Elastic Cloud Computing (EC2) service, which is designed to help users with their computing needs at varying capacities based on the size of the processing power required. When a user begins to use AWS, they will extract data from their computer to send data and algorithms to Amazon's Simple Storage Service (S3) where it is prompted for processing in EC2 (Figure 1A). The data and algorithms then get sent from S3 to EC2 to get processed, and converted into the user’s designed analysis format, and from there get sent back into S3. EC2 uses the implemented architecture to distribute computing power across parallel servers to execute the algorithms in the most efficient manner. For projects that have multiple data sources that individually need to be processed in order to be collaborated for larger processing, services like Amazon’s Elastic Map Reducing (EMR) will be used to increase the speed of computing time by assigning an EC2 to each data source (Figure 1B). When the data computations are finished, the results of the analysis are sent from EMR to S3, and from S3 sent back to the user to view in the user’s designed platform which is programed with corresponding languages to comprehend the data on the computer (Figure 1C) [16].

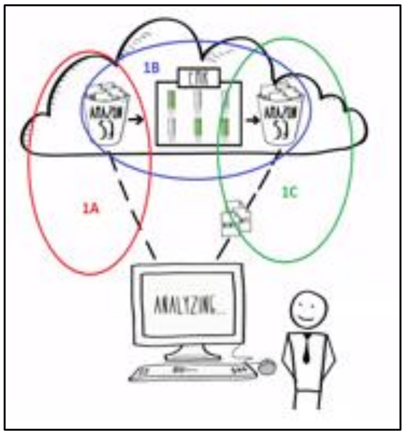


Figure 1: AWS Cloud (S3 and EC2) [16]

Table of Terms

Statistical Terms

Algorithm—A statistical procedure or set of operations

Alpha (α)—The probability of incorrectly rejecting a null hypothesis

Autocorrelation—Occurs when the residuals are not independent from each other

Data wrangling— The process of cleaning and organizing data into a format that is ready for statistical analysis

Geometric sampling schedule— Equation: $S_g = a^i \times n_0$ ["n_0" is the starting sample size and "a" is the common ratio] the sampling schedule starts at n_0 and is multiplied by the common ratio to get the next number in the schedule ex.) n_0=50 and common ratio of 2 yields a sampling schedule of 50,100,200,400, etc.

Iterative loop—The iterations of data wrangling, exploratory analysis, and model building

Multicollinearity—When two or more explanatory variables in a prediction model relate strongly to each other

Parametric vs. nonparametric methods—Parametric methods make assumptions about the probability distributions of the variables and nonparametric methods do not

Multiple linear regression— Modeling the relationship between a response variable and many explanatory variables, with the goal of prediction of the response

Sequential partial F-test—A hypothesis test used to compare the amount of variability explained by a model compared to a reduced version of the model

Sub-sample—A sample of a sample and not a sample of a population

Traditional statistical methods (linear)—Statistical methods that are less robust than the new algorithms being developed today

Software, Cloud and Other Terms

3Vs: Volume, Velocity, Variety—The three dimensions of big data that are used to define how much (volume), how many different types of data (variety), and the processing speed (velocity)

API (Application Program Interface)—Set of protocols and tools for the building of software applications

Cloud—A virtual network of servers that have different functions

Data engineering—Creating statistical algorithms, utilizing cloud infrastructure, and developing a complete platform solution

Hadoop—Open source cloud framework written in Java for distributed storage and processing

Infrastructure—Everything that supports the processing of information in a virtual environment

JMP—A statistical software developed by SAS Institute with an easy to use graphical interface

Statistical library—A collection of statistical functions

MapReduce—Programming model that distributes processing over multiple servers in the cloud

Open source—Software that is freely available and may be used without a software license

Python—Open source high-level general-purpose programming language

R—Open source statistical programming language and software environment

RStudio—Open source development environment and user interface for using R in the cloud

SAS (Statistical Analysis System)—Software suite and programming language for data management and statistical analysis

Structured data—Data with a high degree or organizations, usually tabular

Unstructured data—Data that is not organized in a pre-defined way

Vectorized operations—Operations that are applied to an array (vector) instead of individually to the items within an array

Amazon Web Services (AWS) Terms

AWS—Variety of cloud resources in a pay-as-you-go pricing model

EC2 (Elastic Cloud Computing)—Virtual computer web service by AWS that provides computing capacity in the cloud

Amazon EMR (Elastic MapReduce)—Amazon’s MapReduce (see MapReduce)

Get & Put requests—Requests to access or alter information in S3

S3 (Simple Storage Service)—Online file storage web service offered by AWS

Design - Advisory Tool for Starting a Big Data Project

This advisory tool serves as a starting point for big data projects. The goal of the advisory tool is to allow the user to cut down on big data consulting costs by taking an active role in the project. This is assuming that the user has the time and resources to do so. This guide serves as a tool to anyone interested in learning from big data in a cost effective manner. This tool is similar to the 3-step process used by big data consulting companies, but in a slightly different order. Big data companies often use the same services that are already available to anyone such as Amazon Web Services' (AWS) EC2, and S3. AWS and many other big data services charge on a pay-as-you-go basis for any user. Therefore by managing the project yourself, you avoid the markups that big data consulting companies charge in addition to the actual cost of services provided

Managing your own project gives you the ability to directly control, the level of data engineering performed and thus the costs associated with it. The total costs of the project are largely determined by the skills possessed in statistical analysis, data engineering, and platform development. Every stage of the process, with the exception of creating statistical algorithms, can be learned and applied without formal education. Step three will explain why only a data scientist should create statistical algorithms. In self-managed big data projects, decision nodes will be reached to decide on whether to self-perform or to hire a subject matter expert.

The overall approach in any big data project is to begin with questions of interest that address specific business requirements. These should be clearly defined and must be aligned with specific business goals. This leads into evaluating the data requirements to determine what data needs to be retained and what can be discarded. Once the data requirements are determined then the data science process can begin.

The data science process is summarized in Figure 2. Data science is the process of collecting raw data and turning it into information and eventually intelligence. Once the raw data is collected it must be cleaned and re-cleaned until it is ready for statistical models and algorithms. This is an iterative process where data wrangling and exploratory data analysis are continually performed until it is determined that models and algorithms can be applied. Models and algorithms output a data product that can then be used to make decisions. The data product that is outputted is communicated through visualizations and reports so that intelligent data-driven decisions can be made.

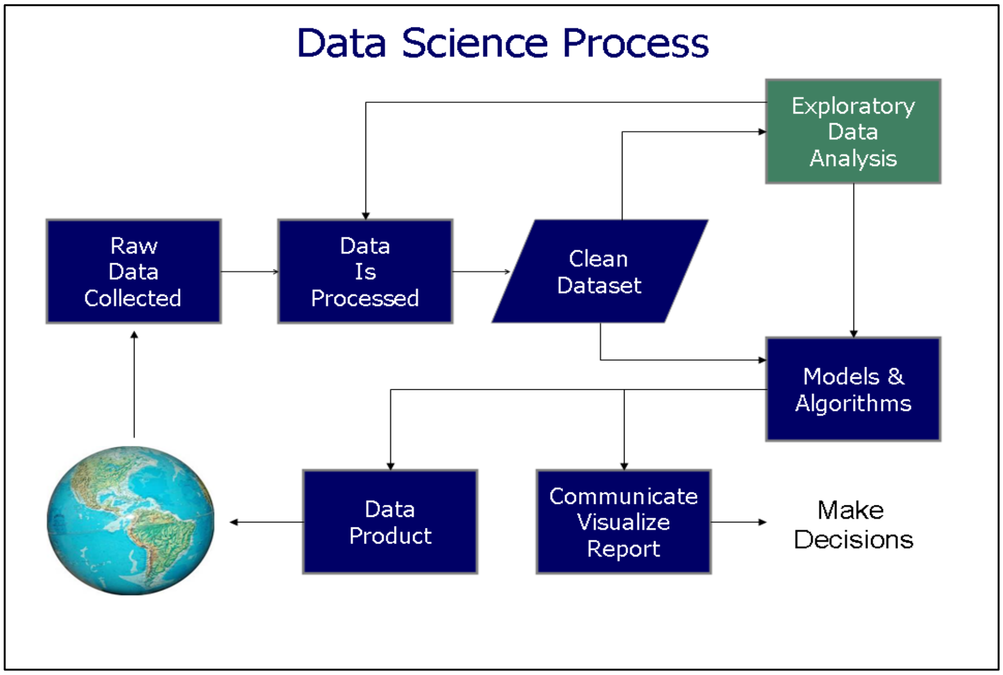


Figure 2: Data Science Process Flow Chart [17]

Data Wrangling

Data wrangling is the process of changing the data from one form to another so that it is ready for statistical analysis. When data wrangling in a big data project, it’s important to consider the problem statement when looking for data that could be used for analysis. Once the problem statement is clearly stated and understood, you can begin to define the data needed to address the problem statement. The search for data now has a specific direction and the collection and storage of the data should be organized and structured to best compliment the statistical method of choice.

The data will come from many sources and in many formats, so it is likely that the data will need to be reformatted prior to analysis. Sometimes this process can be as easy as a copy and paste operation from one source to another. However it may not always be this simple, and the data found often needs to be structured so that it’s ready for analysis. Even after the data is cleaned and ran through an analysis, the cleaning may not be done. It is not uncommon to repeat the cleaning process multiple times throughout the analysis as new problems come to light or new data is collected. There are many tools out there to aid in the wrangling of data and these tools specialize in different aspects of the wrangling process. The different aspects of the wrangling process are data enrichment, ETL/blending, and data integration [18].

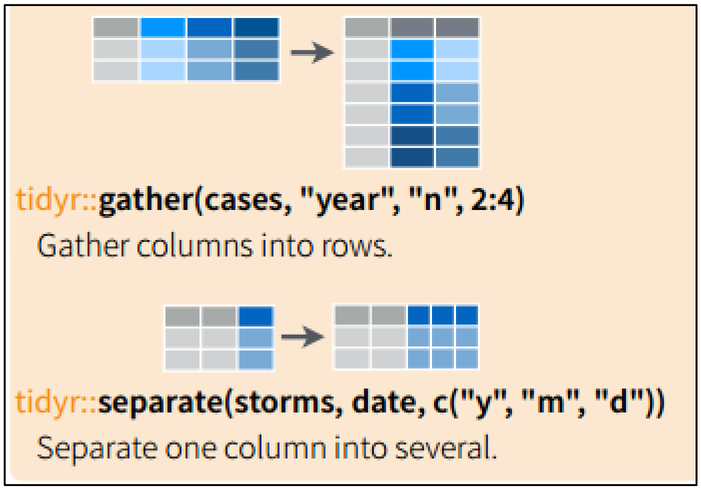
Data enrichment is the cleaning of data. This can take the form of human or automated enrichment, both of which have advantages and disadvantages. Human enrichment is based on the premise that there are tasks that people are better at than machines, such as image classification and language processing (sarcasm, irony, or slang). The challenge here is to find the right combination of both as to be most efficient without sacrificing accuracy.

ETL/Blending (extract, transform, and load blending) is the process where data is joined together from different sources. An example of this is joining company credit card spending information with the types of expenses that employees are allowed to write off. ETL is very similar to data integration. The difference is that data integration focuses more with connecting data applications and specific formats rather than combining data sets with one another.

Two open source tools for data wrangling are Kettle by Pentaho and OpenRefine (formerly Google Refine). Both tools offer easy click and drag user interface to support data wrangling methods like ETL and data integration. OpenRefine is an organization started by google to run ETL and data integration tools [18]. More often today, companies are building their own unique wrangling tools using Python. Python is an open source programming language, and one that can be utilized for other parts of big data projects as well [20].

The techniques for data wrangling vary depending on the initial and desired structure of the data. This is driven by the method of analysis chosen, however the goal remains to get the data in a structured form ready for analysis. Data tidying is an important part of data cleaning, because tidy data sets are much easier to manipulate, model, and visualize. A key aspect of tidy data is that it complements the vectorized operations used in R. The specific structure of tidy data sets is each variable is a column, each observation is a row, and each type of observational unit is a table [21]. The third attribute becomes relevant with larger volumes of data.

See appendix A for a cheat sheet to data wrangle with graphics and syntax in R. The cheat sheet provides graphics and code examples for the techniques discussed below and for many other techniques. Tidy data is what the final product should look like but the actual content of the variables is the important part. Reshaping the data is one technique in data wrangling in which you change the layout of a data set. This includes transforming, separating, uniting, sorting, and more. Transforming is where you turn rows into columns or columns into rows (see Figure 3). Separating is very important because it allows you to further break down information into new variables which allows for more thorough analysis. The reshaping methods facilitate organization of the data into a tidy data set.



```
tidyr::gather(cases, "year", "n", 2:4)
```

Gather columns into rows.

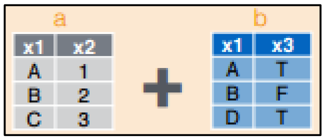
```
tidyr::separate(storms, date, c("y", "m", "d"))
```

Separate one column into several.

Figure 3: Transforming and Separating graphics and R code [19]

Subsetting observations and variables are useful techniques and they help to extract and select variables or observations that meet certain criteria. Subsetting is fundamental to data storage and processing because it can enable or hinder efficient use of computing resources. Summarizing data also can greatly enable efficient processing and storage because a data summary contains information about great volumes of data without the impact of storage volume. Some common summary functions are means, standard deviations, and counts. New variables can be made to better explain other variables or aspects of them. One example of this is binary flags that define a characteristic of another variable. Grouping and ungrouping data are techniques that enable categorical effects to be better visualized and understood.

The techniques so far have focused on manipulation of data within a data set, but combining and separating data sets are also crucial techniques to data wrangling. Data often comes from many sources and it follows that data sets must be combined. One approach is to merge data sets in which a common variable (unique identifier) is used to match attributes of the data sets being merged (see Figure 4). Stacking data sets is a method where data sets are simply stacked on top of each other. This also requires the data sets to share the same unique identifier variable.



a

x1	x2
A	1
B	2
C	3

b

x1	x3
A	T
B	F
D	T

Figure 4: Combining Data Sets [19]

As mentioned earlier data wrangling is an iterative process in which the data is continually cleaned and structured. This must be repeatedly done because new problems with the data will come to light during the analysis stages of the process. Changes in the problem statement of the project and the direction of data collection will also drive the need for continuous data wrangling throughout the project.

Statistical Analysis

The statistical analysis portion of the process takes the form of exploratory data analysis and the use of existing models and algorithms. This step is the limiting factor for most users because of the complex nature of statistics. The key word here is existing, and it means that the algorithm already exists in a library of the software. The use of existing methods should always be attempted to the highest performance before justifying the costs to pay for the development of new algorithms. Fundamental errors can arise from the use of statistical methods without the proper understanding of them. Some common errors are due to a lack of understanding in areas of sampling populations, confounders, bias, overfitting, model assumptions, and sensitivity analysis.

Exploratory analysis creates information, while the use of statistical models and algorithms creates intelligence. As such one must begin with exploratory analysis, which most commonly takes the form of descriptive statistics and data visualization. Means and standard deviations are examples of descriptive statistics are. Data visualization is the examination of data in graphical format. Once the data is better understood, statistical models called algorithms can be applied to the data to create intelligence. Statistical methods for big data can be categorized into supervised and unsupervised learning methods.

Supervised learning methods are mainly concerned with prediction of some criterion. The basis of these methods is to use an algorithm to attempt "to 'learn' a rule or set or rules for predicting a criterion based on observed data" [22]. Unsupervised learning methods do not aim to predict a criterion but instead seek to summarize and explain key features of the data. An important distinction between the two learning methods is that supervised learning methods aim to optimize some measure of performance. Unsupervised learning methods typically lack absolute measures of performance.

Within both supervised and unsupervised learning methods, a further breakdown of parametric and nonparametric methods exist. The main difference is that parametric methods have assumptions about types of probability distributions and inferences of parameters. Non-parametric methods do not have probability distribution assumptions and the parameters of a model are not fixed. If the assumptions of a parametric model are not met then the results can be very misleading and for that reason parametric methods are not very robust. The robustness of methods is further challenged by the volume of big data.

Analysis of data takes many forms depending on the question at hand and the data available. Some examples are predictive analytics, text analytics, network analytics, and web analytics. Predictive analytics is the field of predicting an outcome based on current and historical inputs. Text analytics is the process of deriving high quality information from the analyzing of patterns and trends in text. Network analytics is a method of describing and visualizing connections among objects. Web analytics is the application of various methods to analyze web data such as website views, hits, and clicks. Data analytics are carried out by traditional statistical methods, new big data methods, or a combination of both. The user will need to determine the appropriate methods for their big data project. If existing methods are not adequate for the results required from the analysis, then new statistical algorithms will need to be developed to achieve the required result.

The exact statistical methods will vary for each big data project, but regardless of the statistical methods chosen there are many important things to keep in mind in during analysis. Confounding is a fundamental part of statistical analysis in which a variable not in the model is causing a large amount of bias on the model. Confounding is a large contributor to spurious relationships, which in this case are perceived relationships between variables that are estimated incorrectly because of a confounding factor. An example of this is ice-cream consumption and the number of drowning deaths. At first they may seem to have a positive relationship and one may start to think a cause-and-effect relationship exists. This is obviously erroneous, and a closer look at confounding variables reveals that season and outside temperature influence both ice-cream consumption and the number of drowning deaths (more people swimming).

The above example of cause-and-effect thinking leads to an even bigger potential pitfall in statistical analysis. The pitfall, which too many fall victim to, is confusing correlation with causation. This is absolutely false because this type of cause-and-effect relationship can only be determined if a controlled experiment is designed and carried out. Spurious relationships can also be caused entirely by chance. This is due to the volume of big data and that coincidences of strong relationships will appear even though the relationship has no predictive power. Another caution of big data analysis that relates to relationships between variables is multicollinearity. This is when two or more explanatory variables in a prediction model relate strongly to each other. This is a major issue in traditional linear methods and one that is magnified by big data because of the vast amount of information available, and the fact that so much of it is related.

There are many more caveats, warnings, and cautions of statistical analysis and big data but this guide can't address them all. The best advice is to fully understand the limitations and pitfalls of the statistical methods of choice. Once these are understood it can be determined if the statistical method of choice is robust enough for the application or if more robust methods are needed. This determination will lead you to the pivotal decision in the data engineering step of whether the need for development of new statistical algorithms exists.

Data Engineering

Data engineering consists of creating and applying statistical algorithms, utilizing cloud infrastructure to distribute processing, and the development of a complete platform solution. The creation of statistical algorithms is something that only a data scientist can do, so if the need exists for algorithm development then a data scientist must be hired to do so. However, the utilization of cloud infrastructure and the development of a complete platform solution can be done without the direct hiring of a data scientist. This is carried out by applying open source tools in combination with pay-as-you-go cloud services. Hadoop is an example of an open source infrastructure software. According to Mike Gualtieri's video on "What is Hadoop", he explains that Hadoop (open source software under Apache) is an infrastructure software for storing and processing large data sets [23].

Hadoop stores data through HDFS (Hadoop Distributed File System). A distributed file system is an arrangement of network computers in which files are located on many computers.

This allows you to store a large quantity of files that are too large for traditional computers. Hadoop processes data through a framework called MapReduce. It processes all the data on the servers by sending the processing software to where the data is stored, rather than moving the data to where the software is stored. The reason this is done is because the data is very large and takes too long to send to the software to process.

Hadoop is used by large companies for storing and processing big data. It uses Hive to query data sets for processing, and can deal with storing structured and unstructured data. This software however is not easy to use for just anybody. It has to be used by those with JAVA programming skills or support tools to help guide users through the software. Hadoop works with many vendors to create support tools that will expand the versatility in the types of queries that could be applied to the data. Since Hadoop focuses on being able to distribute workloads of storage and processing, other vendors like Amazon Web Services (AWS) exist to allow users to utilize Hadoop framework in a more user friendly environment.

The easiest way to get started with AWS is to create an account and access the AWS Management Console to manage your use of AWS services. Within the console will include documentation and detailed guides for working with the resource groups AWS provides along with managing the console itself. It’s suggested using the free tutorials that offer interactive labs for each AWS tool to enhance understanding and applicability.

The account administration resource of the console is where the cost breakdown statistics of AWS usage in time, storage, and computing power will be. Taking the time early to understand your usage is essential to managing how money should be spent with the service. The resource groups in AWS include storage, computing, analytics, databases, deployment and management, and app services, however for this part of the guide will focus on storage, computing, and analytics.

Storage space begins with S3's free tier option which offers a 5GB of free storage and monthly limitations of 20,000 get requests, 2,000 put requests, and 15GB of data transfer. To start computing, the EC2 On-Demand Instance t2.micro option should be chosen because it's free for the first year, with a limit of 750 hours per month. It may have a slow computing speed being that it's the most basic selection, but this is perfect for starting out and gaining experience.

The analytics resource group has a couple options to choose from in terms of how data is processed. The best start is to make use of the open source RStudio IDE and incorporate it into EC2. Simple lines of code are written into the AMI in EC2 to allow for the integration of RStudio. There are open sources with easy installation instructions for RStudio. (http://www.louisaslett.com/RStudio_AMI/). For bigger data down the road, RStudio can also be enabled to work with Amazon's EMR. Also understand that RStudio has plenty of pre-installed statistical libraries available for use in performing a wide variety of analysis. Researching the type of analysis within the libraries is necessary to utilize the existing algorithms.

The tools and tutorials to begin utilizing cloud infrastructure for big data projects are out there in open source formats with plenty of support. By utilizing these tools in combination with AWS, you can develop a big data approach where you only pay for what you use. This is a much more cost mindful alternative to the option of opening up your wallet to a big data consulting company. It also allows you to make more informed decisions throughout the lifecycle of the project without the contractual commitment of a consulting relationship. See appendix B for a detailed list of vendors for data sources, data wrangling, and data applications.

Methods - Big Data Analysis

The experimentation for the advisory tool was carried out through wrangling and analysis of many large data sets. Due to scope and time constraints, the experimentation was not carried out for the entire design. The experimentation is based on the data wrangling and statistical analysis parts of the guide. The data sets found online were mostly structured but data wrangling was an ongoing process needed throughout the analysis. The data engineering step was not experimented with due to scope and time constraints for this project.

The goal of the statistical analysis was to find the variables that best predicted a chosen response. This was done by using backward variable elimination in multiple regression. Automated variable selection methods like backward elimination can only operate with quantitative continuous variables so categorical variables were excluded from the analysis. It is not standard in regression to ignore categorical variables, but it had to be done in this project to achieve the volume and variety of analysis that were desired. This is another example of scope and time limitations. Due to the exploratory nature of this project, in depth analysis and model building for each data set was not carried out. However, the variable selection algorithm of backward elimination was evaluated with replication in the context of big data. The variable selection trends were analyzed to better understand how variables are selected for a multiple regression model with the algorithm. This experimentation allowed for the robustness of the variable selection technique to be evaluated.

The backward elimination variable selection algorithm is carried out by first adding all the potential variables to the model. It then proceeds to remove the variables, one by one, until all the variables that remain are significant to the model. A variable is significant in the model if the sequential partial F-test yields a p-value less than the level of significance (alpha or α). A .05

level of significance was used in all analysis. The sequential partial F-test compares the amount of variability explained by a model to that of a reduced version of the model. The test takes into account the different number of explanatory variables in each model, because more variables will always explain more variability. The goal of the test is determine whether the “reduced model” is good enough, or if there is a big enough difference in variability explained to warrant the use of the “full model”. The null hypothesis for this test is that the reduced model is adequate. The corresponding alternative hypothesis is that the reduced model is not adequate. The sequential aspect of the test refers to the fact that the variable in question is added to the model last, and as such the difference in variability explained (reduced vs. full model) is due to that variable alone. In other words it is the difference in variability explained by the variable in question, after taking into account the other variables in the model.

Progressive sampling was used with the variable selection algorithm to allow for the trends to be analyzed across different sample sizes. Progressive sampling is a process in which the sample size of analysis is increased until the accuracy of the model is maintained. In the variable selection context, accuracy of the model takes the form of consistent variable selection. From this point on, the sample in progressive sampling will be referred to as the sub-sample. This is because the samples being analyzed were randomly sampled from a data set and not a population, and thus they are a sample of a sample.

Progressive sampling was carried out on a geometric sampling schedule beginning with a sub-sample size of 50 and using 2 as the common ratio. This resulted in the sample size doubling throughout the schedule (50, 100, 200, etc.). Ten replicates were performed at each sub-sample size. When the sub-sample size was greater than 10% of the data set size, less replicates were collected in order to preserve unique sub-samples. Sub-samples were collected by randomly sampling the entire dataset using simple random sampling. Simple random sampling is sampling without replacement. This ensured that each sub-sample was unique. Once the sub-samples were created, the backward variable elimination algorithm was applied to the sub-samples. For each replicate the variables selected by the algorithm were recorded in a table of all potential variables. If a variable was selected for a given replicate then the corresponding cell was populated with a one. The percentage of times a variable was selected for each sub-sample size was calculated and recorded in a summary table.

Once the variable selection percentages for each sub-sample size were collected, the data was graphed (see Figure 5). The vertical axis is the variable selection percentages from 0% to 100% and the values represent the percent of times that a variable was selected in a given sub-sample size. The horizontal axis is the sub-sample size as it is increased through progressive sampling. The “ALL Obs.” point at the end of the horizontal axis is where the entire dataset was run through the variable selection algorithm. This was not a sub-sample and there is no replication at that point so the corresponding selection percentage can only be 0% or 100%. Each of the lines in the graph represents a different variable.

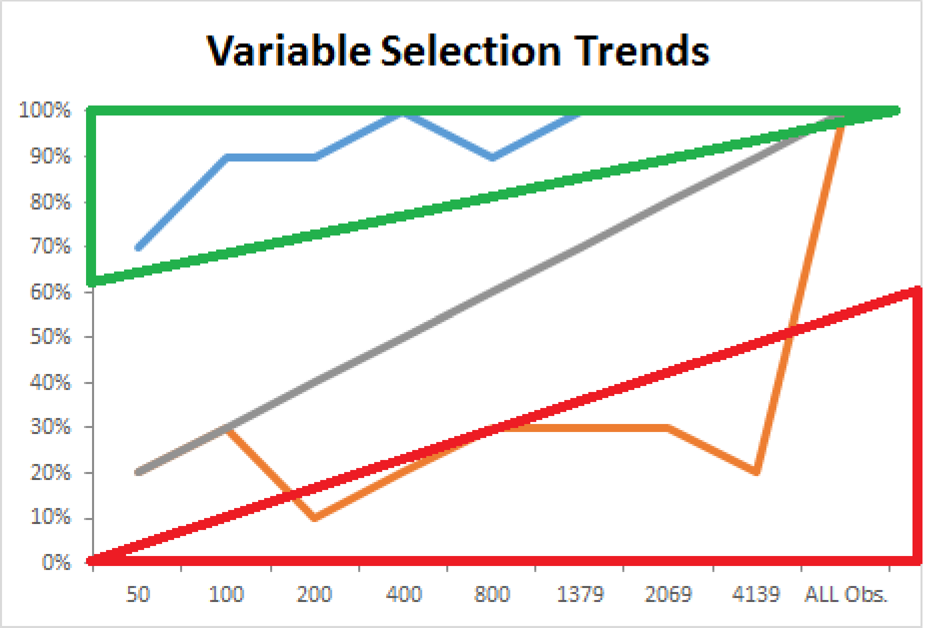


Figure 5: Variable Selection Trends

The two triangular regions and the region between them represent the breakdown of the three variable trends that were identified and further investigated. The shape and size of the three regions are not exact, and in fact the shape and size of the regions differ for each data set analyzed. In addition, the lines will rarely fall exactly into one region, as in the case of the blue line above. Often the lines will cross over more than one region, but nonetheless the trend can still be identified, as in the case of the orange line. The green triangle represents the trend of the variable that had high selection percentages in small sub-sample sizes. This trend will be referred to as "above," and it identifies the variables that showed their importance to the model early on in the progressive sampling. The red triangle represents the trend that the variable had low selection percentages in large sub-sample sizes. This trend identifies variables that did not show their importance until late in the progressive sampling, or in other words they are variables that "sneak" into the model. This trend will be referred to as "below." The region between the red and green triangles represents the trend that the selection percentage of the variable gradually increased as the sub-sample size increased. This trend will be referred to as “middle.” These trends only apply to the variables that were selected for the model in “ALL Obs.” The analysis is based around recognizing these trends for variables selected in the model when the entire dataset was entered into the variable selection algorithm.

Once the trends are graphed and categorized as above, below, or middle, the next step taken was to further investigate the variables by category. One additional category is variables that did not make it into the model when the entire dataset was entered into the variable selection algorithm. This category will be referred to as “not in model”. First, the response is graphed against each explanatory variable and the corresponding correlation coefficient is recorded. The correlation coefficient quantifies the strength and direction of the linear relationship between two variables. Multicollinearity issues were also explored through matrix scatter plot graphs and variance inflation values (VIFs). Multicollinearity is when two or more explanatory variables in a prediction model relate strongly to each other. Multicollinearity is bad because it causes problems in the estimation and interpretation of parameters. In fact, one of the assumptions of multiple linear regression is that little or no multicollinearity exists in the model.

Matrix scatterplot graphs were used to visually evaluate the severity of multicollinearity before the variable selection algorithm was applied. This was done prior to the application of the algorithm because multicollinearity confuses variable selection and therefore makes the method less robust. VIF values can only be collected after a model is created. VIF values in excess of 10 are evidence that multicollinearity is a problem. Some analysis was repeated twice in its entirety, once when multicollinearity was ignored from the start, and again when it was reduced or eliminated from the start. The other differences between the first and second analysis of data sets arose from exploratory analysis and the iterations of loop seen in Figure 6.

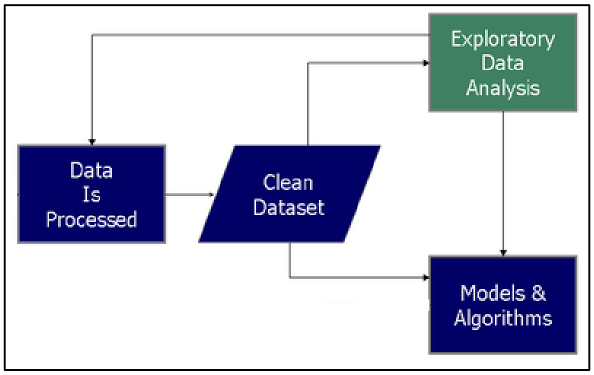


Figure 6: Iterative loop of data wrangling, exploratory analysis, and model building [17]

Efficiency was key in being able to carry out many analyses and to provide replications within each analysis. Efficiency was obtained through the use of SAS (Statistical Analysis System). SAS is a software suite that, among many things, has the ability to wrangle and analyze data. SAS was used extensively in the analysis to automate data wrangling, random sampling, and replications of variable selection. JMP was also used in this project to compliment SAS when a graphical user interface served the analysis needs better. JMP was used to create final models and report model measures that would have been more time consuming to produce in SAS. Excel was also used in the data wrangling and data transferring processes. It was used to wrangle data because excel has many easy to use formulas for organizing and filtering data. It was then used to convert the data to a comma separated values (.csv) file format to then be read in by SAS. In addition to the tools used to process the data, the documentation of the data sets was extremely useful as well. It allowed for a thorough understanding of the meaning of the variables present as well as the unique attributes about the data.

Results

The goal was to get experience applying parts of the advisory tool, and to test the usefulness of it. The iterative loop of data wrangling, exploratory analysis, and model building was the only part of the guide carried out due to scope and time constraints. The model building was limited and only done to record and evaluate measures that were affected by the iterations of data wrangling. Thus the analysis focused on carrying out the iterative loop of data wrangling and analysis to the highest performance. The iterative loop was applied to many data sets across many industries. The data came from hospital and manufacturing operations, housing and energy, and various national accounts such as GDP. Appendices C1 through C13 contain summary information of the analysis performed on each data set. The data came mostly structured, but this only confirmed the need for iterative data wrangling and exploratory analysis.

The results presented are centered on the analysis of the hospital operations data, because it best represents the use of the guide and the iterations of data wrangling and exploratory analysis. See appendices C8 and C9 for summarized results. The annual hospital financial data came from the office of statewide health planning and development for California. There was a dataset for each year from 1995 to 2014, so the first step was combining the datasets. The format and structure of the data varied from year to year so the first step was gaining a thorough understanding of the data and how to standardize the information. The key difference in the datasets was that later versions of the data had new variables that were categorizations of old variables by health care plans. The data was standardized by summarizing the categorized variables into totals. These totals then matched the format of the variables in the earlier datasets, and then all of the data was combined into one large set. At this point there were 8,781 observations with a total of 225 variables, including the unique identifier for each observation. The exploratory analysis began once the datasets were standardized, tidy, and combined.

Exploratory analysis took the form of selecting the variables in the dataset that could be used with the backward elimination variable selection algorithm. This algorithm only works for categorical continuous variables, so the first step was to identify these variables. There were 59 quantitative and continuous available. Some of which include labor statistics, income statement data, and a wide variety of operational data ranging from time spent in the operating room to number of beds in the facility. The response variable chosen was total operational expenses, which are the total costs incurred for providing patient care at the facility. Once the response was chosen the next step was to examine the rest of the variables to see what could best predict total operational expenses. This is the most important step in regression because the quality of the explanatory variables determines the quality of the predictive analytics.

Upon visual inspection of the data, it was noticed that some of the variables had a value of zero for many observations. This was further evaluated by quantifying how many observations had a value of zero for a given variable. A percentage was calculated to represent this proportion. It was found that 26 of the variables had values of zero in over 50% of the observations. It was determined that these variables should not be entered into the variable selection algorithm due to the lack of robustness in the variable selection algorithm in the presence of this phenomenon. The algorithm looks for strength of the linear relationship, and the high concentration of zeros does not allow for the linear relationship to be evaluated correctly. Figure 7 shows a scatterplot graph of a variable with too many zeros. After the variables with too many zeros were removed there were 33 quantitative continuous variables left. Further examination of the variables showed that some variables were linear combinations of other an algorithm.

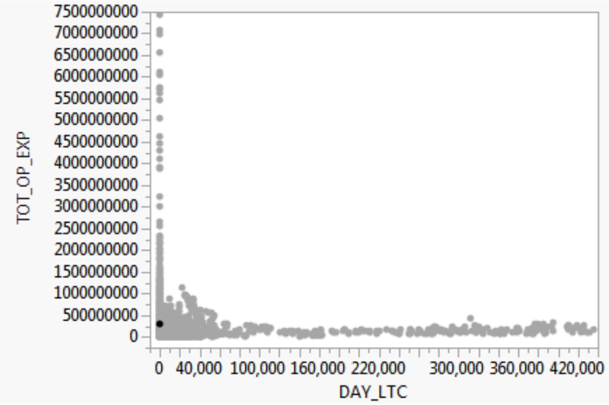


Figure 7: Scatterplot of High Zero Concentration

The Backwards variable elimination algorithm was performed in SAS, using replicates at increasing sub-sample sizes on a geometric sampling schedule. The sub-samples were collected using the simple random sampling algorithm in SAS. Ten sub-samples were created for each sub-sample size on the sampling schedule (50, 100, 200, 400, and 800). In addition there were six sub-samples taken of size 1463, four of size 2195, and two of size 4390. Variable selection of the dataset as a whole was also performed. Each sub-sample was run through the variable selection algorithm and the variables chosen were recorded. The variables selected when the entire set was run were the ones that are referred to as making it into the model.

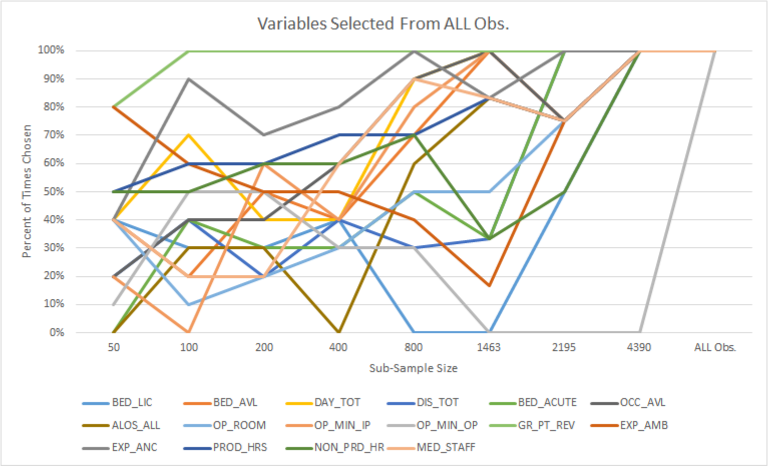


Figure 8: Medical Dataset All Variables Selected in “ALL. Obs.”

Once all the results were recorded the variable selection trends were graphed. Across the horizontal axis is the sub-sample size and across the vertical axis is the percent of times the variable was chosen in the sub-sample size. Each line represents a different variable, and the trends were analyzed only for variables that were selected when the entire dataset was run (ALL. Obs.). Upon first glance it just looks like a mess of lines (Figure 8), but after the graphs are categorized, the trends of above below and middle can be seen clearly. The above trend is seen in Figure 9, for the gross patient revenue variable. It makes sense that gross patient revenue would be a good predictor, and its importance was evident even in small sample sizes. The bottom two graphs (Figure 10) show the middle trend (left graph) and the below trend (right graph). The below trend is harder to visualize, but it can be seen that these are the variables that do not show their importance until late in the progressive sampling. It’s important when a variable shows up in the progressive sampling schedule because it is a reflection of how much data is needed to be processed to learn the importance of the variable. A variable in the above trend is recognized early, and with less processing. This is important when processing in the cloud environment because more time processing costs more money.

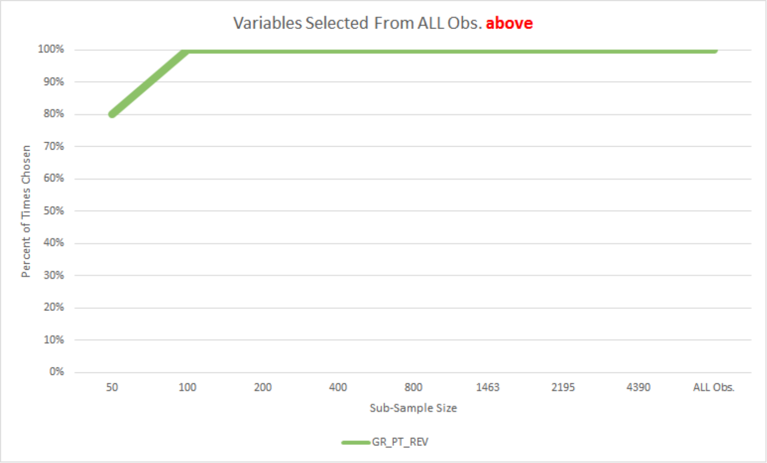


Figure 9: Medical Dataset Variables Selected Early

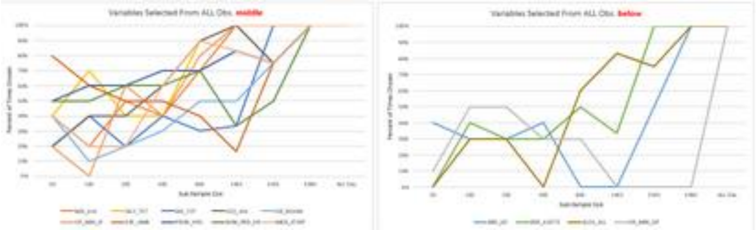


Figure 10: Medical Dataset Variables Selected (middle and below trends)

The next step is to create a model and make transformations if necessary to meet the model assumptions of multiple linear regression. This step was not carried out to the extent of transformations and ensuring model validity. However, the variables were entered into a model so that the VIF measures could be recorded and multicollinearity could be explored. The VIF values indicated that there was evidence of multicollinearity. BED_LIC (licensed beds) and BED_AVL (available beds) both had VIF values around 25. VIF values in excess of 10 indicate that multicollinearity is a problem. These two variables are extremely similar, in that they are both counts of beds but with different classifications. It is not surprising that they display multicollinearity. Little or no multicollinearity is one of the assumptions of multiple linear regression, so this issue must be fixed before moving on with the building of the prediction model. Thus the iterative loop of data wrangling, exploratory analysis, and model building continues to turn.

One way to address the problem of multicollinearity is to remove the problematic variables from the model. However, since multicollinearity affects the estimation of the effects of variables, then the entire variable selection algorithm was affected by the multicollinearity as well. Thus, more data wrangling was performed to resolve the multicollinearity issue. Through this iteration of data wrangling it was found that an additional 11 variables displayed problematic multicollinearity. Some of the variables were removed entirely. Others, like the categories of beds, were averaged into a new variable. The end result of this iteration of wrangling is that 15 variables were entered into the variable selection algorithm instead of 26. The result of this iteration of wrangling is that no problematic multicollinearity was seen in the variables selected by the algorithm. The analysis of this dataset ended with the second run of the variable selection algorithm.

Multicollinearity problems were seen in many of the data sets analyzed and this often led to repeating analysis of entire data sets multiple times. The tax data from 990 forms is a great example where the Multicollinearity problem was removed entirely in the second analysis. See Appendix C1 and C2 for the differences in the first and second analysis of the tax data. Multicollinearity could not always be removed as in the example above. Multicollinearity was a big problem in the GDP and National Accounts data sets. This is often seen in economic data because of the similarities of economic measures. There was severe Multicollinearity in both analyses of the GDP data sets. See Appendices C4 and C5 for the different analyses of the GDP data sets. See Appendices C6 and C7 for the different analyses of the National Accounts data sets.

The iterative process of data wrangling, statistical analysis, and model building was carried out for all the data sets. The hope was to see commonalities across data sets in the way the variable selection trends were affected by the iterations of wrangling and analysis. Specifically, the interest was in identifying patterns in different industry sectors so that the iterative wrangling and analysis could be standardized for future users of the advisory tool. There were no conclusive findings that indicated these methods could be standardized across different industry sectors. However, it was observed that the iterations of data wrangling and analysis are better associated with the statistical algorithm of choice. This is not to say that the iterations of wrangling and analysis can’t be standardized for specific problems that the data presents.

Discussion

Big data is a term typically reserved for data sets that are so big or complex that traditional data processing and storage techniques are inadequate. The term big data is used loosely in this paper to also include data sets that are just below the boundary of what traditional processing and storage techniques are capable of. The statistical analysis was not carried out to the point of creating prediction models. This was acceptable because the goal of the experimentation was to apply parts of the design and to challenge the robustness of a traditional statistical method in the presence of big data

The iterations of data wrangling and analysis were the focus of the experimentation. The goal was to test the usefulness of the advisory tool. Scope and time constraints of the project limited the design testing to the iterations of data wrangling and analysis. This covers approximately the first half of the guide, leaving out the final model building and data engineering. The experience gained, through the iterations of data wrangling and exploratory analysis, was extremely valuable in evaluating the usefulness of the design. One key point is that a fundamental understanding of what the data represents is crucial to enhancing the results of data wrangling and analysis. The statistical analysis improved every time the iterative loop was navigated.

The wrangling and analysis iterations of many different data sets would not have been possible without an efficient approach. This was attained through the utilization of various SAS libraries that provided the many algorithms used for data wrangling, random sampling, and variable selection. SAS is efficient at analyzing large amounts of data because it uses vectorized operations, as does R. Efficiency in processing and storage is key in big data analysis because of the pay-as-you-go model for storage and processing.

The robustness of the variable selection algorithm was challenged through the experimentation of wrangling and analysis. The robustness of a statistical method refers to the effectiveness of the method in the presence of outliers and other departures from the model assumptions. In statistics, classical methods rely heavily on model assumptions which are not always met in practice. An example of this is the little or no multicollinearity assumption of multiple linear regression. This assumption was not initially met in some of the data sets, but it was combatted through iterations of data wrangling and analysis. If the assumptions of a model can’t be met, then a more robust statistical method will be needed. An example of this is when the assumptions of parametric models are not met and nonparametric methods must be attempted instead. Big data challenges the robustness of traditional statistical methods. When the robustness of existing methods does not allow for the required accuracy of results, then the creation of new algorithms must be explored. The robustness of existing methods and the statistical knowledge of the analyst are the limiting factors of the guide. This is something that will likely be faced when using the advisory tool.

The results of applying the advisory tool will vary from case to case, depending on the critical skills the user possesses and the development of such skills. The critical skills begin with R, Python, and knowledge in the statistical methods of choice. Basic knowledge of statistics, and any programming language is a must to begin utilizing this guide. This can be as simple as VBA and a few statistics courses. These fundamental skills are the building blocks needed to begin using the guide. In addition, the knowledge of a statistical programming language such as R or SAS will give you an even better foundation to start with. Experience in the cloud is the next skill that will greatly increase the usability to the advisory tool.

The best start is to first get good at R and become familiar with the statistical libraries it contains. This allows data wrangling and exploratory analysis to be performed at a high level. However, this is just the start. To maximize the usability of the guide, a strong understanding and control over the statistical methods of choice is crucial. The results of the analysis are entirely dependent of the accuracy of the statistics used. It is the results of the statistical analysis that will lead to more informed data driven decision making.

Statistical proficiency is the limiting factor in the advisory tool. This is because the programming languages of R and Python, can be learned without much formal education. However, statistical proficiency can’t be acquired by reading forums online and patching together code examples, as can be done with learning programming languages. That being said, it is not unlikely that outside help may be needed from a statistician or data scientist. However, if the analysis is carried out thoroughly, and valid models can be created, then there may not be a need for a statistician.

The advisory tool is a great way to begin harnessing the power of learning from big data. However, continuous improvement in all things is always necessary and as such there is room for improvement in the guide. First and foremost, the guide does not include any information on data collection methods. This is very important because without data to analyze, there is no use for the guide. Data collection was left out due to scope and time constraints. In addition to data collection, the advisory tool should include pseudo code and more code excerpts (R and Python). This would greatly add to the usefulness and applicability of the tool. The guide did not include specifics about the AWS tools available, but that was not necessary because of the vast amount of educational resources available by AWS.

This project pushed the boundaries of what can be done with big data using traditional computer framework (no cloud usage). Storage and processing limits of traditional computers were tested and in some cases reached, which only further verifies the eventual need to operate in the cloud environment. For example some data sets were too big to be downloaded. The limits of excel were the first to be reached in the form of excel crashing when trying to open the data set. In these cases SAS was used to read in and explore the data. The processing limits of SAS were never reached, even when millions of observations were analyzed using the variable selection algorithms. This is because of the efficiency of vectorized operations.

Cost Justification

In comparison to hiring a big data consulting company, using the advisory tool may take longer to carry out the project to completion. However, the experience gained by carrying out the project using the advisory tool is invaluable. Just the experience alone, of carrying out the iterations of data wrangling and exploratory analysis, will lead to an enhanced comprehension of data and the power it possess. Self-performing data wrangling has great cost saving benefits. Data wrangling typically occupies 80% of data scientist’s time in big data projects, and the biggest hurdle for them is receiving dirty data [24]. So if a data scientist is needed in the project, wrangling the data first by yourself will result in less time spent wrangling by the data scientist. Since data scientists are paid very high hourly wages ($150-$200 per hour) [9], extra time saved wrangling equates to direct cost savings. This is assuming that the data wrangling performed before a data scientist takes over is of adequate quality.

The next savings realized by use of the advisory tool is in avoiding the markups that consulting companies charge in addition to the actual cost. These markups exist anytime a third party is utilized for project management or services. Going to the source eliminates these markups. This is done by accessing AWS directly and only paying for the data and processing performed, not the third party markup. This is also done when subject matter experts such as data scientists or statisticians need to be hired. This assumes that they can be hired at cost and for the duration of the job. This is not a far stretch because in all industries there are specialists who operate on their own as a private contractor. This assumption of outsourced work obtained at cost applies for any aspects of the project that need to be sub-contracted out.

Exact numerical comparisons of the total cost of ownership of a big data project for the advisory tool versus a big data consulting company could not be obtained. This is due to the varying nature of each big data project and the critical skills possessed by the user of the advisory tool. If the advisory tool had been carried out from start to finish, then a more exact cost justification could have been carried out.

Conclusion

This project addresses the need for a cost effective approach to big data projects. The problem is that the knowledge of what it takes to self-perform a big data project is not commonly known and the consulting alternatives are not cost effective. To solve this problem it was first understood how big data consulting companies carry out big data projects and the corresponding costs. The next objective was to carry out big data analyses and apply findings from the research and experimentation to make recommendations for those who wish to carry out their own big data projects. The main conclusions are:

- Self-performing data wrangling can save lots of money in a big data project
- Statistical knowledge is the limiting factor in self-performing a big data project
- Open source tools are available and powerful enough for big data analysis
    - R for data wrangling and statistical algorithms
    - Python for implementing cloud architecture

The experimentation was very important in realizing that the limitation of the design is in the statistical knowledge of the user. This was realized as the robustness of the traditional method of multiple linear regression was challenged. Multicollinearity, which is heightened in big data, was the biggest contributor to challenging the robustness of this method. More robust methods for multiple regression exist and should be utilized. However these methods require more statistical knowledge and thus the main limitation is the analyst's statistical proficiency.

Big data is everywhere and this project serves as a starting point for taking on a big data project. It explains the tools that are needed to carry out a big data project and how to access them. Much more can be detailed about big data projects but due to the uniqueness of each project it is best to seek out specific case studies that can be applied to each application. However, this project is very useful in driving home the point that big data projects can be self-managed in a cost effective manner.

If we could do this project differently next time there are some things we would change. Instead of analyzing many data sets we would focus on one set and attempt to use the data to drive a business decision. Furthermore we would develop a cloud application to store and process the data. These changes would allow for an entire case study to be developed to more accurately estimate the costs and benefits of self-performing a big data project. However, this project was still very beneficial as the results are a tool to start and manage your own big data project. This brings the power of big data to businesses in a cost effective manner.

Bibliography

[1] Kutner, Michael H., Chris Nachtsheim, John Neter, and William Li. "Selection of Independent Variables." Applied Linear Statistical Models. Boston: McGraw-Hill Irwin, 2005. 418. Print.

[2] Navidi, William. "Model Selection." Engineering Statistics. N.p.: McGraw-Hill Companies, 2011. 619. Print.

[3] Oehlert, Gary W. "Checking Assumptions." A First Course in Design and Analysis of Experiments. New York: W.H. Freeman, 2000. 114. Print.

[4] Chawla, Nitesh, Thomas E. Moore, Jr., Kevin W. Bowyer, Lawrence O. Hall, Clayton Springer, and Philip Kegelmeyer. "Bagging Is A Small-Data-Set Phenomenon." (n.d.): n. pag. IEEE. Web. 18 Mar. 2015.

[5] Provost, Foster, David Jensen, and Tim Oats. "Efficient Progressive Sampling." ACM Digital Library. N.p., 1999. Web.

[6] Weiss, Gary M., and Ye Tien. "Maximizing Classifier Utility When There Are Data Acquisition and Modeling Costs." Data Mining and Knowledge Discovery 17.2 (2007): 253-82. Rpt. in N.p.: Springer Link, n.d. Web. 19 Mar. 2015.

[7] "Big Data Consulting Services Top Analytics Opportunities for Channel." SearchITChannel. N.p., n.d. Web. 19 Mar. 2015. <http://searchitchannel.techtarget.com/feature/Big-data-consulting-services-top-analytics-opportunities-for-channel>.

[8] Granville, Vincent. "Big Data Is Cheap and Easy." Web log post. - Data Science Central. N.p., 12 Feb. 2014. Web. 19 Mar. 2015. <http://www.datasciencecentral.com/profiles/blogs/big-data-is-cheap-and-easy>.

[9] Khasanshyn, Renat. "Big Data Company Investigation." Telephone interview. 19 Mar. 2015.

[10] "Analytics." Google Official Website – Web & Reporting. N.p., n.d. Web. 18 Mar. 2015.

[11] "Brilliant." IBM Watson Analytics. N.p., n.d. Web. 19 Mar. 2015.

[12]Townsend, Patrick. "Re: Functionality of Watson in Free Mode?" Web log comment. N.p., n.d. Web. <https://community.watsonanalytics.com/discussions/questions/1922/functionality-of-watson-in-free-mode.html>.

[13] "Top 38 Questions about the Watson Services on Bluemix." Bluemix. N.p., 03 Nov. 2014. Web. 19 Mar. 2015. <https://developer.ibm.com/bluemix/2014/11/03/watson-webinar-qa-responses/>.

[14] "AWS | Big Data Analytics - Cloud Computing & Cloud Storage for Big Data." Amazon Web Services, Inc. N.p., n.d. Web. 19 Mar. 2015. <http://aws.amazon.com/big-data/>.

[15] "AWS | Amazon Simple Storage Service (S3) - Online Cloud Storage for Data & Files." Amazon Web Services, Inc. N.p., n.d. Web. 19 Mar. 2015. <http://aws.amazon.com/s3/>.

[16] [Figure 1A, 1B, 1C] "AWS | Amazon Elastic MapReduce (EMR) | Hadoop MapReduce in the Cloud." Amazon Web Services, Inc. N.p., n.d. Web. 17 May 2015. <http://aws.amazon.com/elasticmapreduce/>.

[17] Farcaster. Data Visualization Process V1.png. Digital image. Data Visualization. Wikipedia, Sept. 2014. Web. 29 May 2015. <http://en.wikipedia.org/wiki/Data_visualization>.

[18] Biewald, Lukas. "The Data Science Ecosystem Part 2: Data Wrangling."Computerworld. N.p., 1 Apr. 2015. Web. 30 May 2015. <http://www.computerworld.com/article/2902920/the-data-science-ecosystem-part-2-data-wrangling.html>.

[19] R Studio. Data Wrangling with Dplyr and Tidyr Cheat Sheet. Program documentation. Rstudio. Rstudio, n.d. Web. 30 May 2015. <http://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf>.

[20] "KDnuggets." KDnuggets Analytics Big Data Data Mining and Data Science. N.p., n.d. Web. 30 May 2015. <http://www.kdnuggets.com/2014/08/four-main-languages-analytics-data-mining-data-science.html>.

[21] Wickham, Hadley. "Journal of Statistical Software — Index." Journal of Statistical Software 59.10 (2014): n. pag. Journal of Statistical Software — Index. 05 May 2014. Web. 31 May 2015.

[22] Oswald, F. L., & Putka, D. J. (in press). Statistical methods for big data. In S. Tonidandel, E. King,& J. Cortina. (2015). Big data at work: *The data science revolution and organizational psychology*. New York: Routledge.

[23] "Mike Gualtieri's Blog." What Is Hadoop? N.p., 7 June 2015. Web. 01 June 2015. <http://blogs.forrester.com/mike_gualtieri/13-06-07-what_is_hadoop>.

[24] "The Data Behind Today's Data Scientists: An Infographic." CrowdFlower. N.p., 9 Feb. 2015. Web. 04 June 2015. <http://www.crowdflower.com/blog/the-data-behind-todays-data-scientists>.

Data Bibliography

[25] Becker, R. Gray, W. and Marvakov, J. (2013). *NBER-CES Manufacturing Industry Database, 1958-2009* [data file and codebook]. National Bureau of Economic Research, Census Bureau's for Economic Studies. <http://www.nber.org/nberces/>.

[26] Singer, J. David, Stuart Bremer, and John Stuckey. (1972). "*Capability Distribution, Uncertainty, and Major Power War, 1820-1965*." [data file and codebook] in Bruce Russett (ed) Peace, War, and Numbers, Beverly Hills: Sage, 19-48.

[27] OSHPD. Healthcare Information Division. (2015)."*Hospital Annual Financial Data*". 1995-2013. [data file and codebook]<http://www.oshpd.ca.gov/hid/Products/Hospitals /AnnFinanData/SubSets/SelectedData/default.asp>.

[28] Residential Energy Consumption Survey (RECS). U.S. Energy Information Administration - EIA - Independent Statistics and Analysis, (2013). "*2009 RECS Survey Data*". 1978-2009. [data file and codebook] <http://www.eia.gov/consumption/residential/data /2009/index.cfm?view=microdata>.

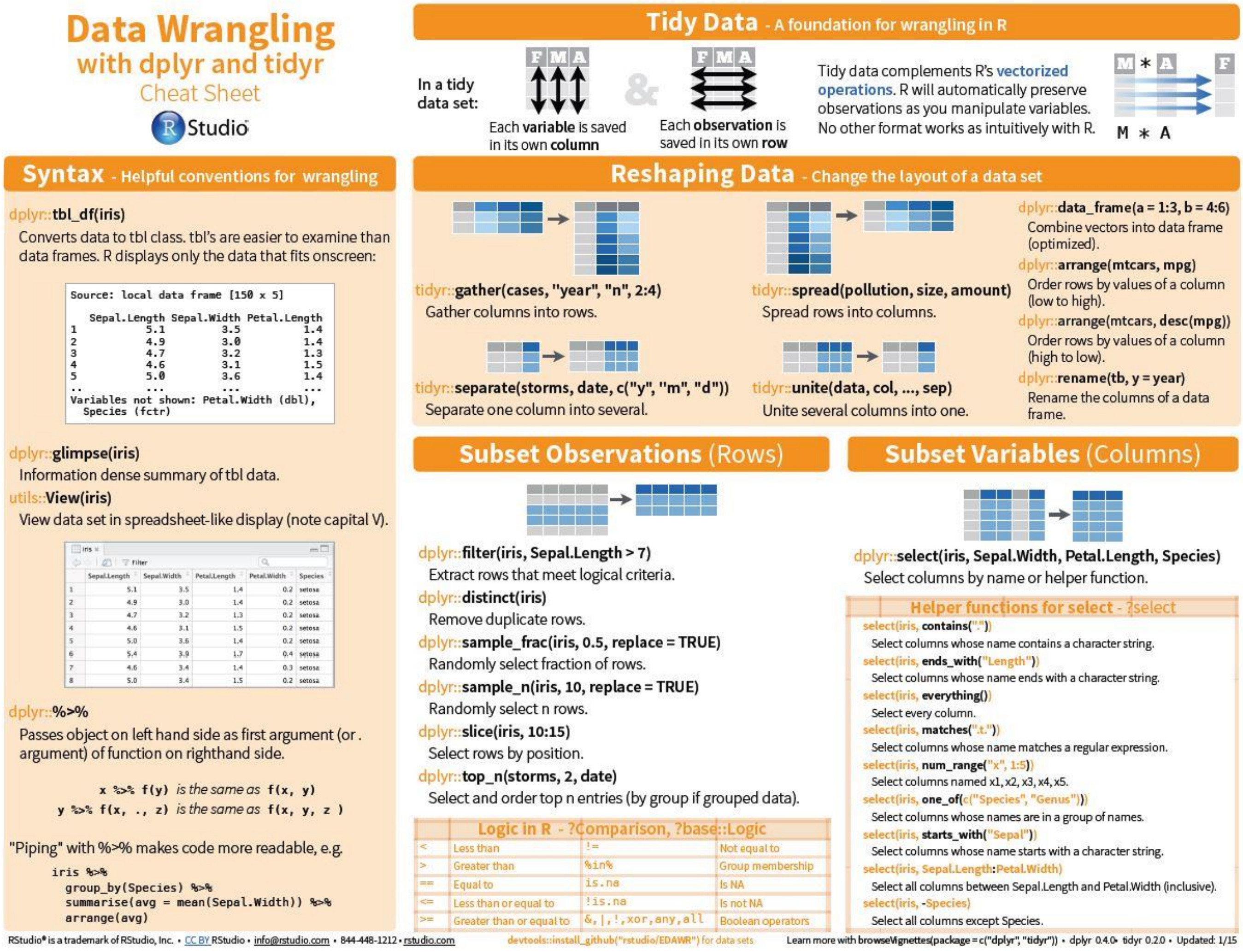
[29] Feenstra, Robert C., Robert Inklaar and Marcel P. Timmer (2015), "*The Next Generation of the Penn World Table*" [data file and codebook] forthcoming American Economic Review, available for download at www.ggdc.net/pwt

[30] Roth. J. (2013) "*IRS Form 990 Data - SOI Tax Stats - Annual Extract of Tax-Exempt Organization Financial Data*" [data file and codebook] National Bureau of Economic Research, Census Bureau's for Economic Studies. http://users.nber.org/data/soi-tax-stats-annual-extracts-form-990.html

[31] Heysem Kaya, Pınar Tüfekci , Sadık Fikret Gürgen: "*Local and Global Learning Methods for Predicting Power of a Combined Gas & Steam Turbine"*, [data file and codebook]Proceedings of the International Conference on Emerging Trends in Computer and Electronics Engineering ICETCEE 2012, pp. 13-18 (Mar. 2012, Dubai)

Appendices

Appendix A: Data Wrangling Cheat sheet [19]



Data Wrangling
with dplyr and tidyr
Cheat Sheet

RStudio

Syntax - Helpful conventions for wrangling

dplyr::**tbl_df(iris)**

Converts data to tbl class. tbl's are easier to examine than data frames. R displays only the data that fits onscreen:

```
Source: local data frame [150 x 5]

  Sepal.Length Sepal.Width Petal.Length
1          5.1         3.5          1.4
2          4.9         3.0          1.4
3          4.7         3.2          1.3
4          4.6         3.1          1.5
5          5.0         3.6          1.4
..         ...         ...          ...
Variables not shown: Petal.Width (dbl),
  Species (fctr)
```

dplyr::**glimpse(iris)**

Information dense summary of tbl data.

utils::**View(iris)**

View data set in spreadsheet-like display (note capital V).

iris ×

Filter

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

dplyr::**%>%**

Passes object on left hand side as first argument (or . argument) of function on righthand side.

```
  x %>% f(y)       is the same as  f(x, y)
y %>% f(x, ., z)   is the same as  f(x, y, z )
```

"Piping" with %>% makes code more readable, e.g.

```
iris %>%
  group_by(Species) %>%
  summarise(avg = mean(Sepal.Width)) %>%
  arrange(avg)
```

Tidy Data - A foundation for wrangling in R

In a tidy data set:

Each **variable** is saved in its own **column**

&

Each **observation** is saved in its own **row**

Tidy data complements R's **vectorized operations**. R will automatically preserve observations as you manipulate variables. No other format works as intuitively with R.

Reshaping Data - Change the layout of a data set

tidyr::**gather(cases, "year", "n", 2:4)**

Gather columns into rows.

tidyr::**spread(pollution, size, amount)**

Spread rows into columns.

tidyr::**separate(storms, date, c("y", "m", "d"))**

Separate one column into several.

tidyr::**unite(data, col, ..., sep)**

Unite several columns into one.

dplyr::**data_frame(a = 1:3, b = 4:6)**

Combine vectors into data frame (optimized).

dplyr::**arrange(mtcars, mpg)**

Order rows by values of a column (low to high).

dplyr::**arrange(mtcars, desc(mpg))**

Order rows by values of a column (high to low).

dplyr::**rename(tb, y = year)**

Rename the columns of a data frame.

Subset Observations (Rows)

dplyr::**filter(iris, Sepal.Length > 7)**

Extract rows that meet logical criteria.

dplyr::**distinct(iris)**

Remove duplicate rows.

dplyr::**sample_frac(iris, 0.5, replace = TRUE)**

Randomly select fraction of rows.

dplyr::**sample_n(iris, 10, replace = TRUE)**

Randomly select n rows.

dplyr::**slice(iris, 10:15)**

Select rows by position.

dplyr::**top_n(storms, 2, date)**

Select and order top n entries (by group if grouped data).

Logic in R - ?Comparison, ?base::Logic

<	Less than	!=	Not equal to
>	Greater than	%in%	Group membership
==	Equal to	is.na	Is NA
<=	Less than or equal to	!is.na	Is not NA
>=	Greater than or equal to	&, |, !, xor, any, all	Boolean operators

Subset Variables (Columns)

dplyr::**select(iris, Sepal.Width, Petal.Length, Species)**

Select columns by name or helper function.

Helper functions for select - ?select

select(iris, **contains(".")**)

Select columns whose name contains a character string.

select(iris, **ends_with("Length")**)

Select columns whose name ends with a character string.

select(iris, **everything()**)

Select every column.

select(iris, **matches(".t.")**)

Select columns whose name matches a regular expression.

select(iris, **num_range("x", 1:5)**)

Select columns named x1, x2, x3, x4, x5.

select(iris, **one_of(c("Species", "Genus"))**)

Select columns whose names are in a group of names.

select(iris, **starts_with("Sepal")**)

Select columns whose name starts with a character string.

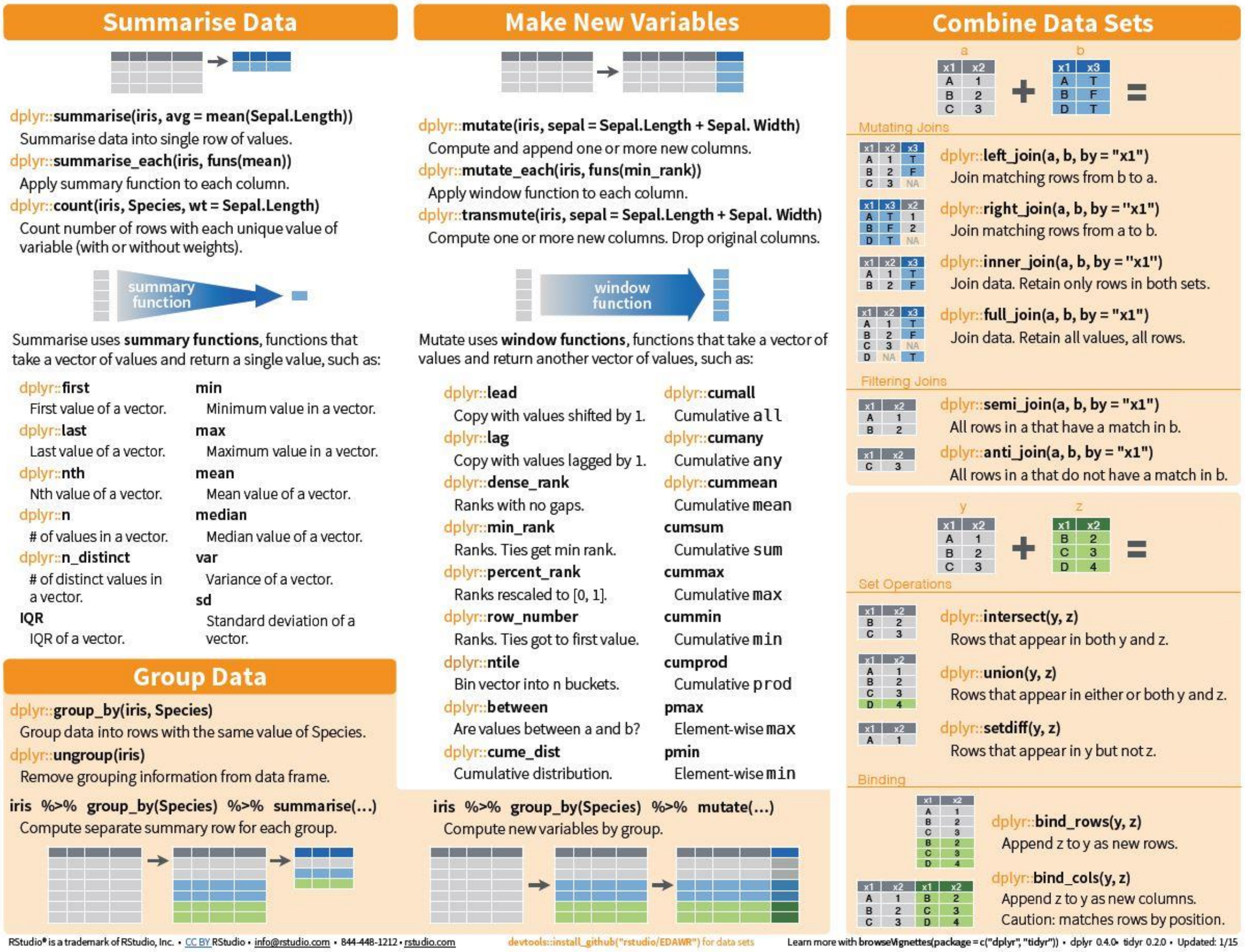
select(iris, Sepal.Length:Petal.Width)

Select all columns between Sepal.Length and Petal.Width (inclusive).

select(iris, -Species)

Select all columns except Species.

RStudio® is a trademark of RStudio, Inc. • CC BY RStudio • info@rstudio.com • 844-448-1212 • rstudio.com   devtools::install_github("rstudio/EDAWR") for data sets   Learn more with browseVignettes(package = c("dplyr", "tidyr")) • dplyr 0.4.0• tidyr 0.2.0 • Updated: 1/15



Summarise Data

dplyr::**summarise(iris, avg = mean(Sepal.Length))**
Summarise data into single row of values.

dplyr::**summarise_each(iris, funs(mean))**
Apply summary function to each column.

dplyr::**count(iris, Species, wt = Sepal.Length)**
Count number of rows with each unique value of variable (with or without weights).

summary function

Summarise uses **summary functions**, functions that take a vector of values and return a single value, such as:

dplyr::**first**
First value of a vector.

dplyr::**last**
Last value of a vector.

dplyr::**nth**
Nth value of a vector.

dplyr::**n**
# of values in a vector.

dplyr::**n_distinct**
# of distinct values in a vector.

IQR
IQR of a vector.

min
Minimum value in a vector.

max
Maximum value in a vector.

mean
Mean value of a vector.

median
Median value of a vector.

var
Variance of a vector.

sd
Standard deviation of a vector.

Group Data

dplyr::**group_by(iris, Species)**
Group data into rows with the same value of Species.

dplyr::**ungroup(iris)**
Remove grouping information from data frame.

iris %>% group_by(Species) %>% summarise(…)
Compute separate summary row for each group.

Make New Variables

dplyr::**mutate(iris, sepal = Sepal.Length + Sepal. Width)**
Compute and append one or more new columns.

dplyr::**mutate_each(iris, funs(min_rank))**
Apply window function to each column.

dplyr::**transmute(iris, sepal = Sepal.Length + Sepal. Width)**
Compute one or more new columns. Drop original columns.

Mutate uses **window functions**, functions that take a vector of values and return another vector of values, such as:

dplyr::**lead**
Copy with values shifted by 1.

dplyr::**lag**
Copy with values lagged by 1.

dplyr::**dense_rank**
Ranks with no gaps.

dplyr::**min_rank**
Ranks. Ties get min rank.

dplyr::**percent_rank**
Ranks rescaled to [0, 1].

dplyr::**row_number**
Ranks. Ties got to first value.

dplyr::**ntile**
Bin vector into n buckets.

dplyr::**between**
Are values between a and b?

dplyr::**cume_dist**
Cumulative distribution.

dplyr::**cumall**
Cumulative `all`

dplyr::**cumany**
Cumulative `any`

dplyr::**cummean**
Cumulative `mean`

cumsum
Cumulative `sum`

cummax
Cumulative `max`

cummin
Cumulative `min`

cumprod
Cumulative `prod`

pmax
Element-wise `max`

pmin
Element-wise `min`

iris %>% group_by(Species) %>% mutate(…)
Compute new variables by group.

Combine Data Sets

a

x1	x2
A	1
B	2
C	3

+

b

x1	x3
A	T
B	F
D	T

=

Mutating Joins

x1	x2	x3
A	1	T
B	2	F
C	3	NA

dplyr::**left_join(a, b, by = "x1")**
Join matching rows from b to a.

x1	x3	x2
A	T	1
B	F	2
D	T	NA

dplyr::**right_join(a, b, by = "x1")**
Join matching rows from a to b.

x1	x2	x3
A	1	T
B	2	F

dplyr::**inner_join(a, b, by = "x1")**
Join data. Retain only rows in both sets.

x1	x2	x3
A	1	T
B	2	F
C	3	NA
D	NA	T

dplyr::**full_join(a, b, by = "x1")**
Join data. Retain all values, all rows.

Filtering Joins

x1	x2
A	1
B	2

dplyr::**semi_join(a, b, by = "x1")**
All rows in a that have a match in b.

x1	x2
C	3

dplyr::**anti_join(a, b, by = "x1")**
All rows in a that do not have a match in b.

y

x1	x2
A	1
B	2
C	3

+

z

x1	x2
B	2
C	3
D	4

=

Set Operations

x1	x2
B	2
C	3

dplyr::**intersect(y, z)**
Rows that appear in both y and z.

x1	x2
A	1
B	2
C	3
D	4

dplyr::**union(y, z)**
Rows that appear in either or both y and z.

x1	x2
A	1

dplyr::**setdiff(y, z)**
Rows that appear in y but not z.

Binding

x1	x2
A	1
B	2
C	3
B	2
C	3
D	4

dplyr::**bind_rows(y, z)**
Append z to y as new rows.

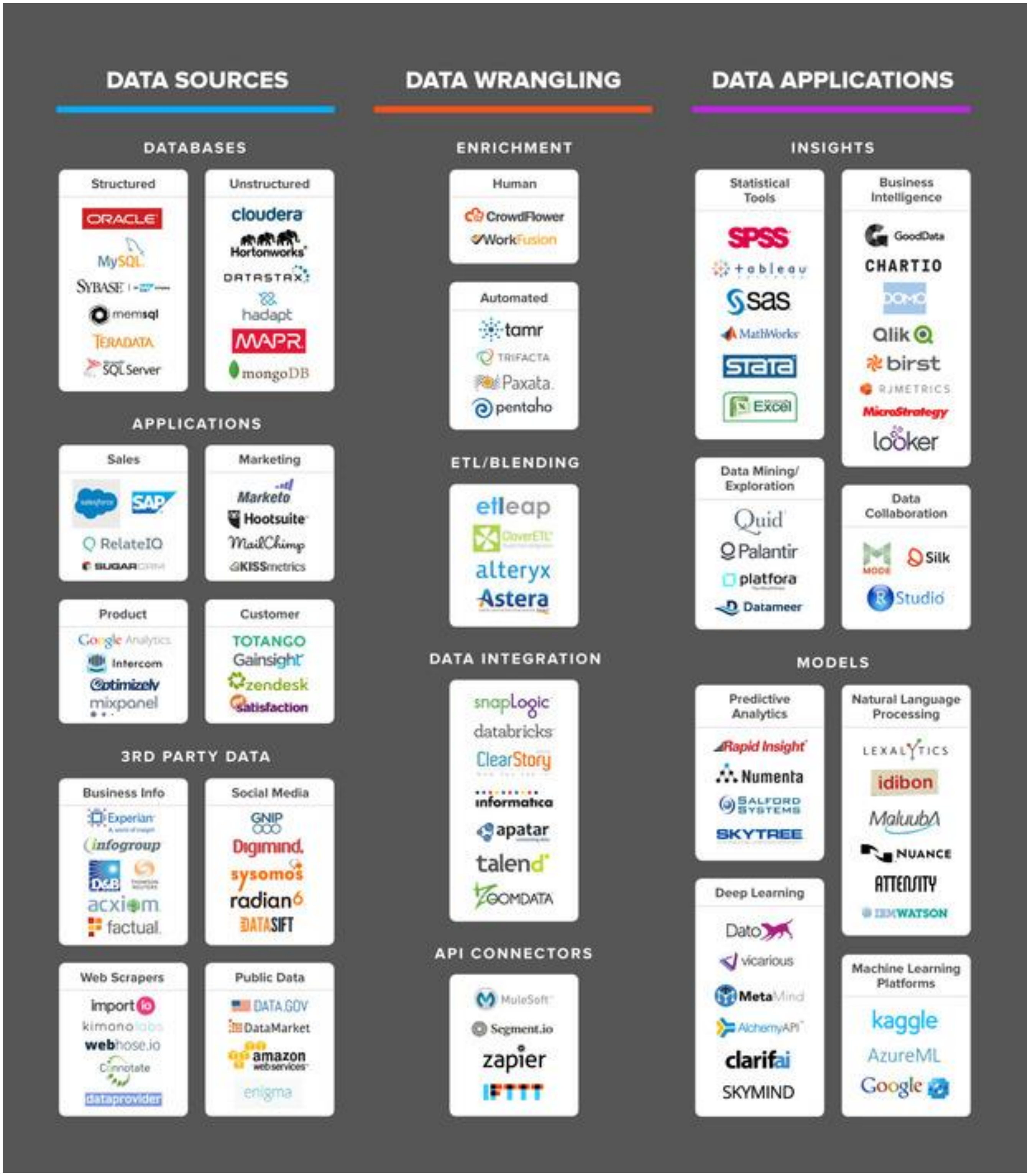
x1	x2	x1	x2
A	1	B	2
B	2	C	3
C	3	D	4

dplyr::**bind_cols(y, z)**
Append z to y as new columns.
Caution: matches rows by position.

devtools::install_github("rstudio/EDAWR") for data sets

Learn more with browseVignettes(package = c("dplyr", "tidyr")) • dplyr 0.4.0• tidyr 0.2.0 • Updated: 1/15

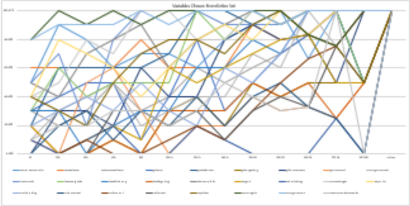
Appendix B: Big Data Services Vendor List [18]

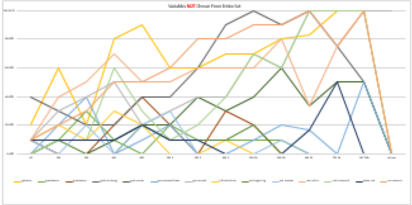


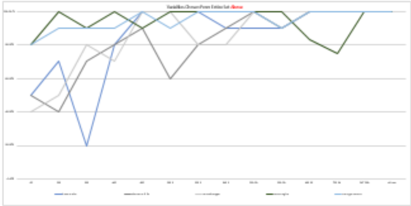
Appendix C1: 990 Tax (1st Analysis)

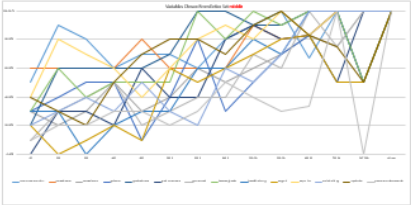
Model before backwards selection takes place:

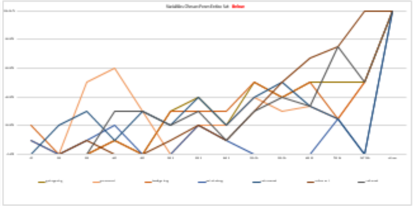
totfuncexpns = compnsatncurrofcr cstbasisecur cstbasisothr gnlsecur gnlsothr grsalesecur grsalesinvent grsalesothr grsincfndrsng grsincgaming grsincmembers grsincother grsrcptspublicuse grsrntsprsnl grsrntsreal initiationfees invstmntinc lesscstofgoods lessdirfndrsng lessdirgaming miscrevtot11e netgnls netincfndrsng netincgaming netincsales netrntlinc othrsalwages payrolltx profndraising rntlexpnsprsnl rntlexpnsreal rntlincprsnl rntlincreal royaltsinc subseccd totcntrbgfts totprgmrevnue totrevenue txexmptbndsproceeds











Variables:

- Not in model: gnlsecur, grsalesecur, grsalesothr, grsincfndrsng, grsincother, grsrcptspublicuse, grsrntsreal, initiationfees, netincgaming, netincsales, netrnlinc, rntlexpnsprsnl, subseccd, totrevenue
- In model:
  - above (bottom left): investmntinc, miscrevto11e, othrsalwages, totcntrbgfts, totprgmrevenue
  - below (bottom right): grsincgaming, grsrntsprsnl, lessdirgaming, netincfndrsng, rntlexpnsreal, rntlincprsnl, rntlincreal
  - Middle(bottom middle): compnsatncurrofcr, cstbasisecur, cstbasisothr, gnlsothr, grsalsinvent, grsincmembers, lesscstofgoods, lessdirfndrsng, netgnls, payrolltx, profndraising, royaltsinc, txexmptbndsproceeds

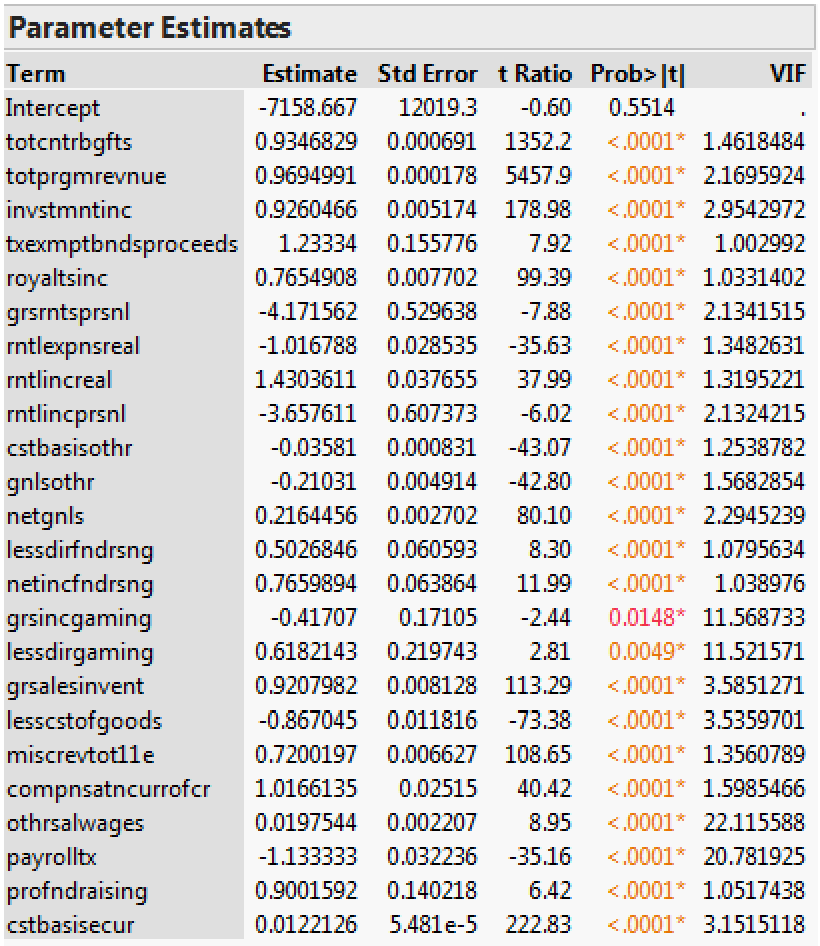
Multicollinearity:

grsincgaming: VIF=11.5

lessdirgaming: VIF=11.5

othrsalwages: VIF=22.1

payrolltx: VIF=20.7

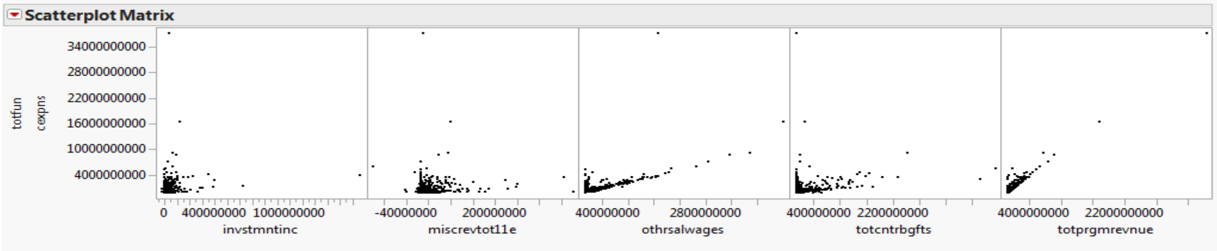


Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	-7158.667	12019.3	-0.60	0.5514	.
totcntrbgfts	0.9346829	0.000691	1352.2	<.0001*	1.4618484
totprgmrevnue	0.9694991	0.000178	5457.9	<.0001*	2.1695924
invstmntinc	0.9260466	0.005174	178.98	<.0001*	2.9542972
txexmptbndsproceeds	1.23334	0.155776	7.92	<.0001*	1.002992
royaltsinc	0.7654908	0.007702	99.39	<.0001*	1.0331402
grsrntsprsnl	-4.171562	0.529638	-7.88	<.0001*	2.1341515
rntlexpnsreal	-1.016788	0.028535	-35.63	<.0001*	1.3482631
rntlincreal	1.4303611	0.037655	37.99	<.0001*	1.3195221
rntlincprsnl	-3.657611	0.607373	-6.02	<.0001*	2.1324215
cstbasisothr	-0.03581	0.000831	-43.07	<.0001*	1.2538782
gnlsothr	-0.21031	0.004914	-42.80	<.0001*	1.5682854
netgnls	0.2164456	0.002702	80.10	<.0001*	2.2945239
lessdirfndrsng	0.5026846	0.060593	8.30	<.0001*	1.0795634
netincfndrsng	0.7659894	0.063864	11.99	<.0001*	1.038976
grsincgaming	-0.41707	0.17105	-2.44	0.0148*	11.568733
lessdirgaming	0.6182143	0.219743	2.81	0.0049*	11.521571
grsalesinvent	0.9207982	0.008128	113.29	<.0001*	3.5851271
lesscstofgoods	-0.867045	0.011816	-73.38	<.0001*	3.5359701
miscrevtot11e	0.7200197	0.006627	108.65	<.0001*	1.3560789
compnsatncurrofcr	1.0166135	0.02515	40.42	<.0001*	1.5985466
othrsalwages	0.0197544	0.002207	8.95	<.0001*	22.115588
payrolltx	-1.133333	0.032236	-35.16	<.0001*	20.781925
profndraising	0.9001592	0.140218	6.42	<.0001*	1.0517438
cstbasisecur	0.0122126	5.481e-5	222.83	<.0001*	3.1515118

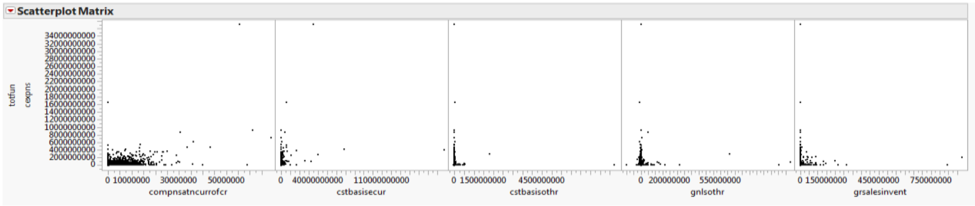
Response vs. Explanatory

Above: R^2=.0719, R^2=.0570 R^2=.534 R^2=.118 R^2=.9475

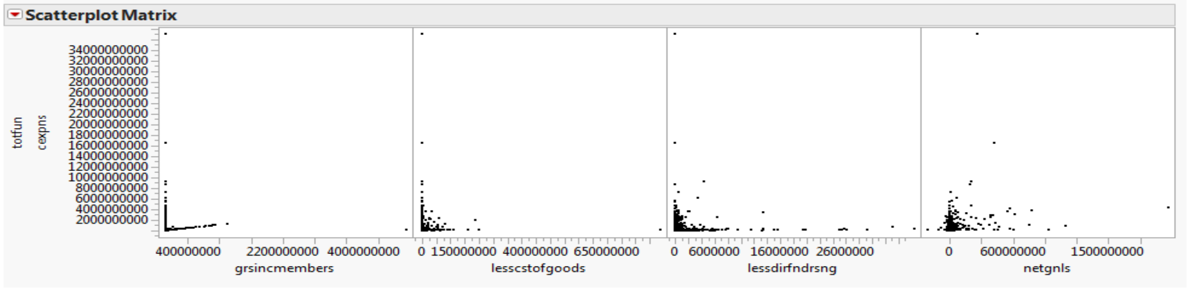


Average R^2 of above trend=.3456

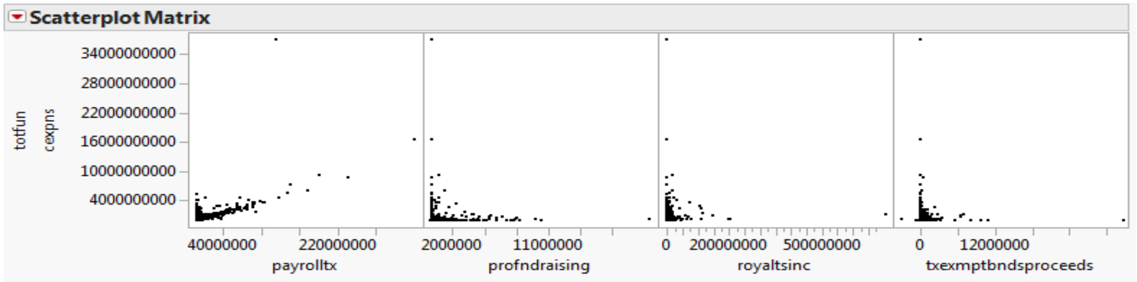
Middle: R^2= .263, R^2=.073, R^2<.001, R^2=.00156, R^2=.00272



R^2<.001, R^2<.001, R^2=.00255, R^2=.0839,

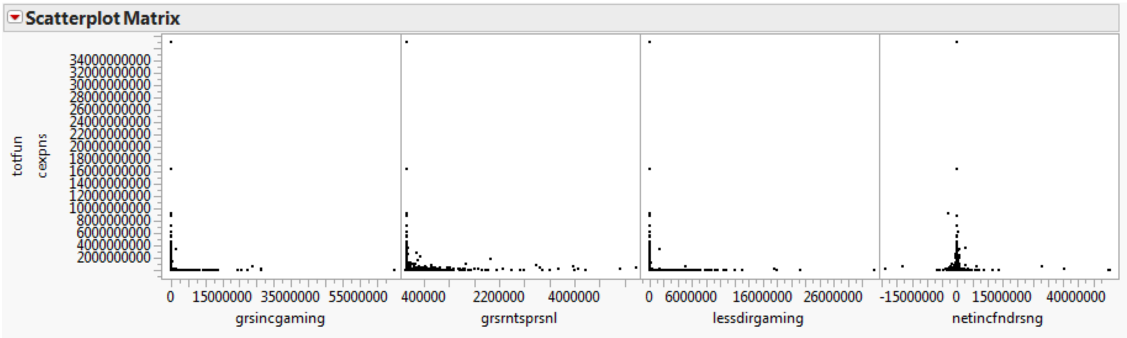


R^2=.537, R^2=.00267, R^2=.00598, R^2= .00118,

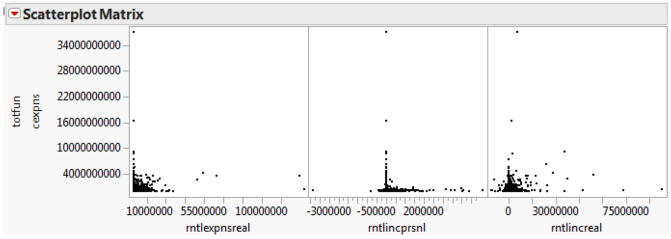


Average R^2 of above trend=0.1501

Below: R^2<.001, R^2<.001, R^2<.001, R^2<.001,



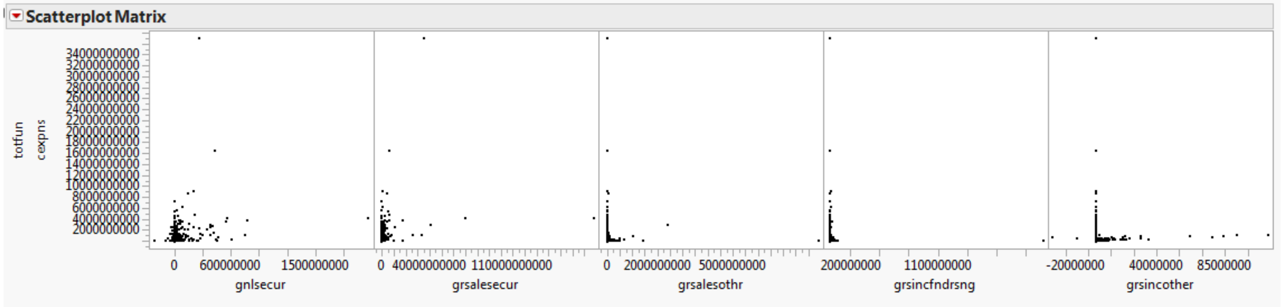
R^2=.023908, R^2<.001, R^2=.034022



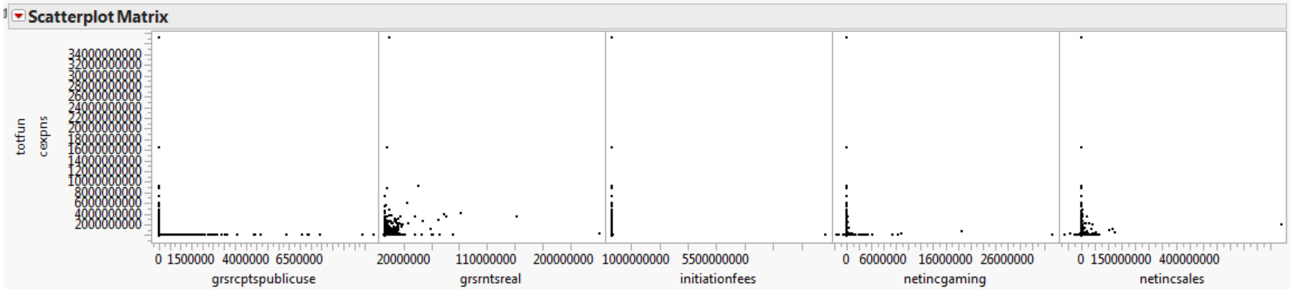
Average R^2 of above trend=.008463

Not in Model:

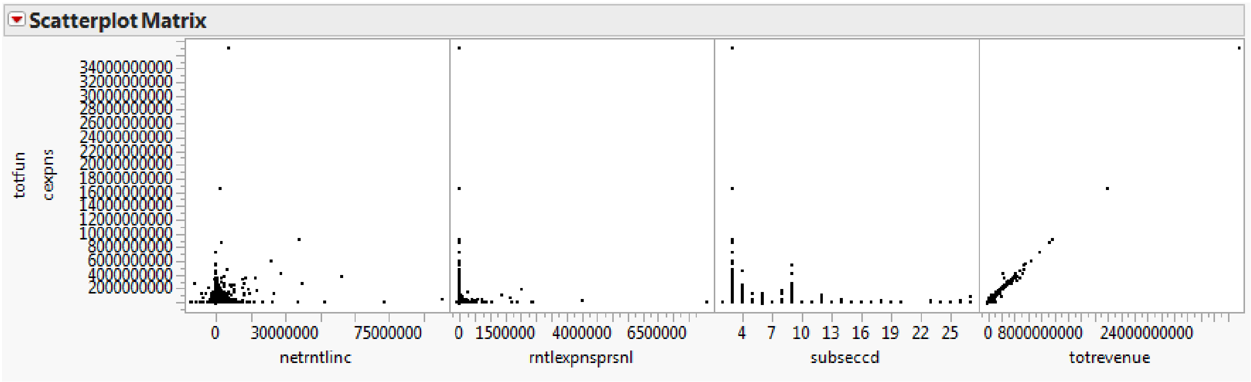
R^2= 0.096799, R^2= 0.074602, R^2<.001, R^2<.001, R^2=0.002171



R^2<.001, R^2=0.038553, R^2<.001, R^2<.001, R^2=0.00301,



R^2=0.034491, R^2<.001, R^2<.001, R^2=0.992698

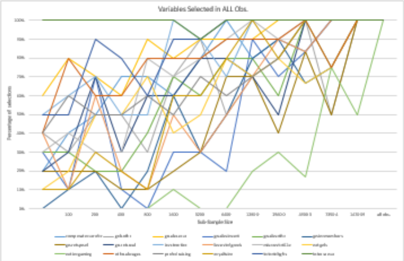


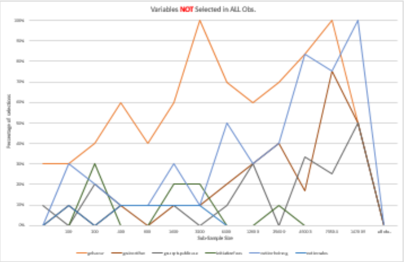
Average R^2 of above trend=0.08879717

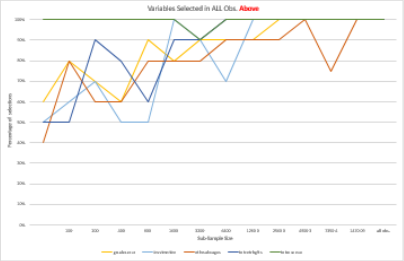
Appendix C2: 990 Tax (2nd Analysis)

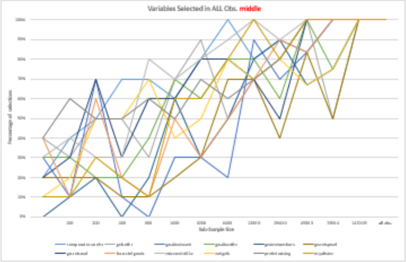
Model before backwards selection:

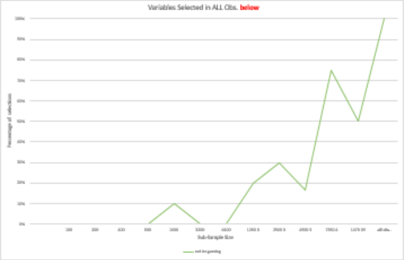
totfuncexpns = compnsatncurrofcr gnlsecur gnlsothr grsalesecur grsalesinvent grsalesothr grsincmembers grsincother grsrcptspublicuse grsrntsprsnl grsrntsreal initiationfees invstmntinc lesscstofgoods miscrevtot11e netgnls netincfndrsng netincgaming netincsales othrsalwages profndraising royaltsinc totcntrbgfts totrevenue







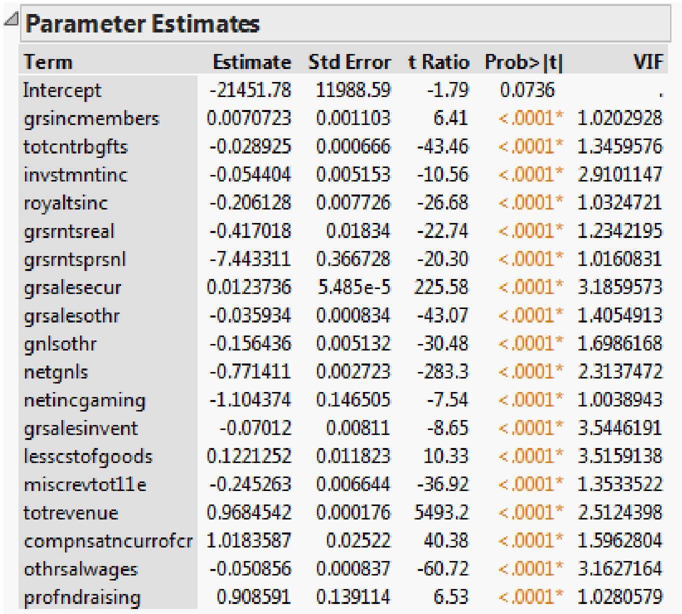




Variables:

- Not in model (top right): gnlsecur, grsincother, grsrcptspublicuse, initiationfees, netincfndrsng, netincsales
- In model
  - Above (bottom left): grsalesecur, invstmntinc, othrsalwages, totcntrbgfts, totrevenue
  - Below (bottom right): netincgaming
  - Middle (bottom middle): compnsatncurrofcr, gnlsothr, grsalesinvent, grsalesothr, grsincmembers, grsrntsprsnl, grsrntsreal, lesscstofgoods, miscrevtot11e, netgnls, netincfndrsng, profndraising, royaltsinc

Multicollinearity: No multicollinearity in the model

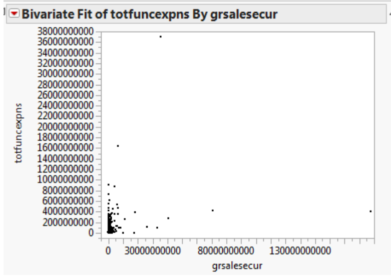


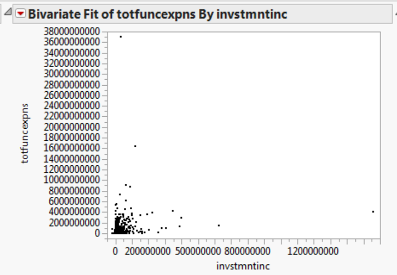
Parameter Estimates

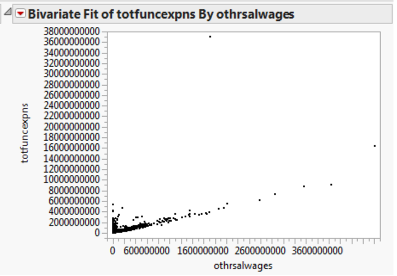
Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	-21451.78	11988.59	-1.79	0.0736	.
grsincmembers	0.0070723	0.001103	6.41	<.0001*	1.0202928
totcntrbgfts	-0.028925	0.000666	-43.46	<.0001*	1.3459576
invstmntinc	-0.054404	0.005153	-10.56	<.0001*	2.9101147
royaltsinc	-0.206128	0.007726	-26.68	<.0001*	1.0324721
grsrntsreal	-0.417018	0.01834	-22.74	<.0001*	1.2342195
grsrntsprsnl	-7.443311	0.366728	-20.30	<.0001*	1.0160831
grsalesecur	0.0123736	5.485e-5	225.58	<.0001*	3.1859573
grsalesothr	-0.035934	0.000834	-43.07	<.0001*	1.4054913
gnlsothr	-0.156436	0.005132	-30.48	<.0001*	1.6986168
netgnls	-0.771411	0.002723	-283.3	<.0001*	2.3137472
netincgaming	-1.104374	0.146505	-7.54	<.0001*	1.0038943
grsalesinvent	-0.07012	0.00811	-8.65	<.0001*	3.5446191
lesscstofgoods	0.1221252	0.011823	10.33	<.0001*	3.5159138
miscrevtot11e	-0.245263	0.006644	-36.92	<.0001*	1.3533522
totrevenue	0.9684542	0.000176	5493.2	<.0001*	2.5124398
compnsatncurrofcr	1.0183587	0.02522	40.38	<.0001*	1.5962804
othrsalwages	-0.050856	0.000837	-60.72	<.0001*	3.1627164
profndraising	0.908591	0.139114	6.53	<.0001*	1.0280579

Response vs. Explanatory Variables:

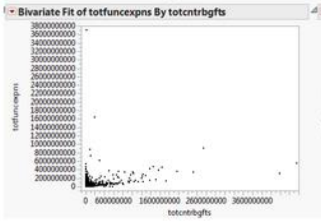
Above: R^2= .0746, R^2= .07191, R^2=.5346,

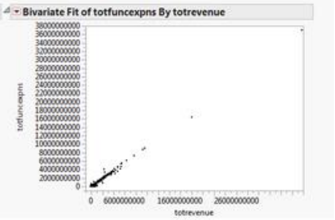






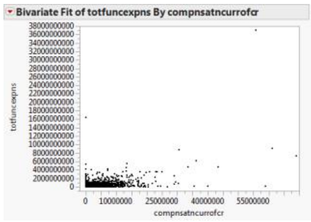
R^2=.11803, R^2=.9926

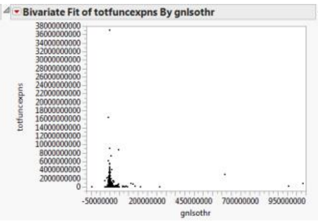


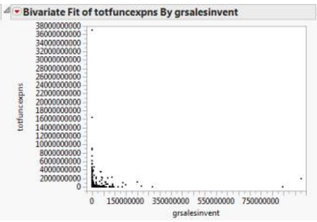


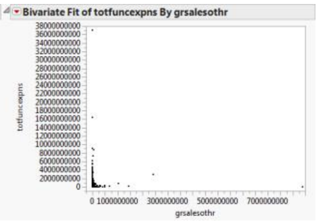
Average R^2 of above trend=0.3583

Middle: R^2= .26378, R^2=.001567, R^2=.002722, R^2<.001,

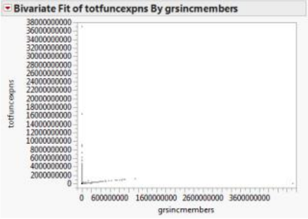


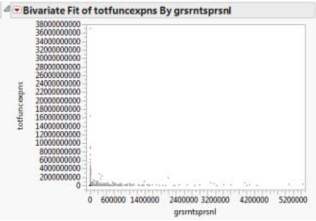


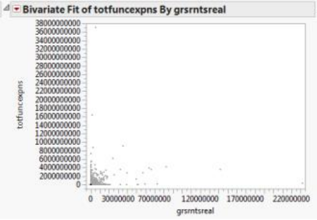


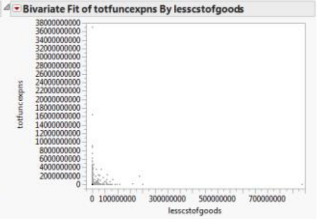


R^2=.00179, R^2<.001, R^2=.038553, R^2<.001,

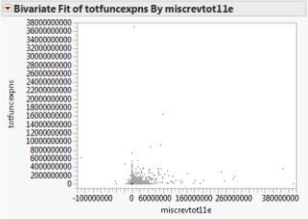


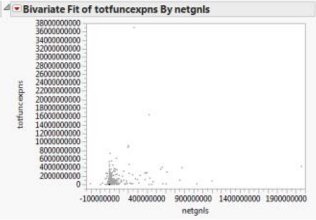


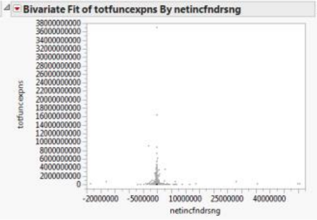


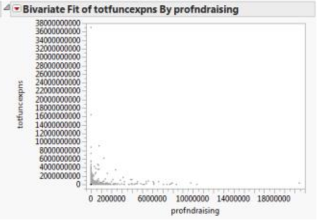


R^2=.057027, R^2=.083931, R^2<.001, R^2=.002376

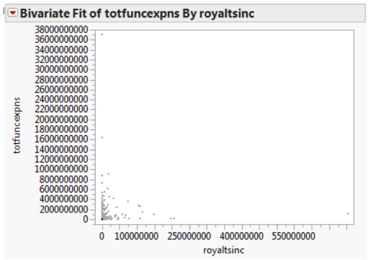






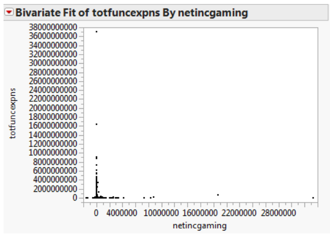


R^2=.005983

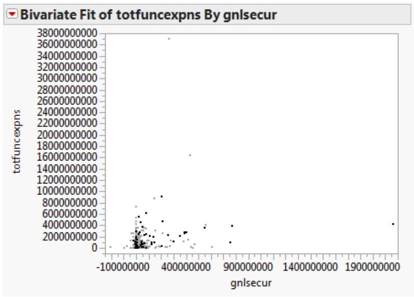


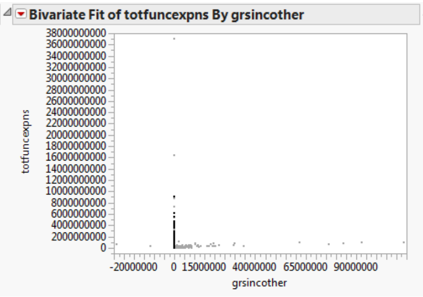
Average R^2 of above trend=0.3583

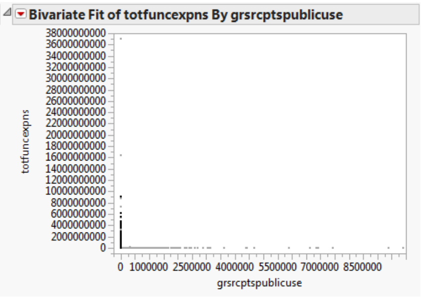
Below: R^2<0.001



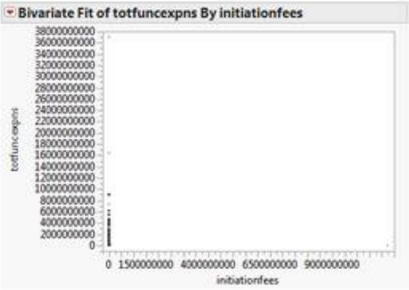
Not in Model: R^2= .096799, R^2=.002171, R^2<0.001

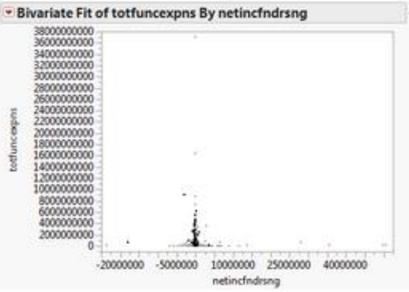


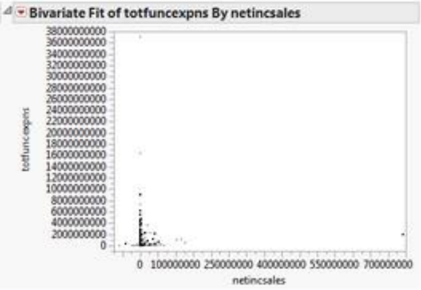




R^2<.001, R^2<.001, R^2=.00301





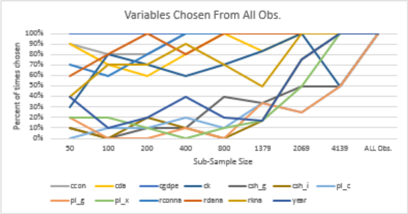


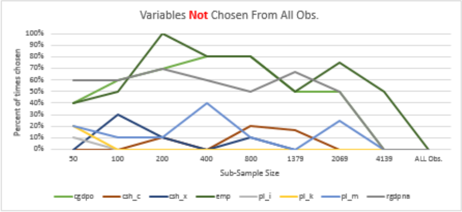
Average R^2 of above trend=0.0170

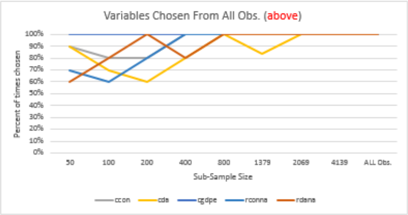
Appendix C4: GDP (1st analysis)

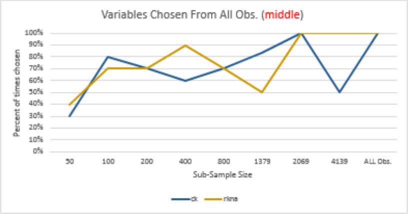
Model before backwards selection:

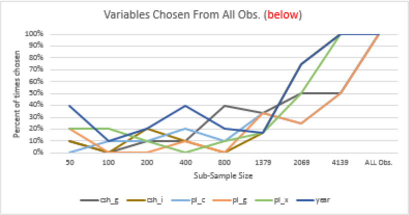
rgdpe = ccon cda cgdpe cgdpo ck csh_c csh_g csh_i csh_x emp pl_c pl_g pl_i pl_k pl_m pl_x rconna rdana rgdpna rkna year











Variables:

- Not in model (top right): cgdpo, csh_c, csh_x, emp, pl_i, pl_k, pl_m, rgdpna
- In model
    - above (bottom left): ccon, cda, cgdpe, rconna, rdana
    - below (bottom right): csh_g, csh_i, pl_c, pl_g, pl_x, year
    - middle (bottom middle): ck, rkna

Multicollinearity:

ccon: VIF=1231

cda: VIF=3671

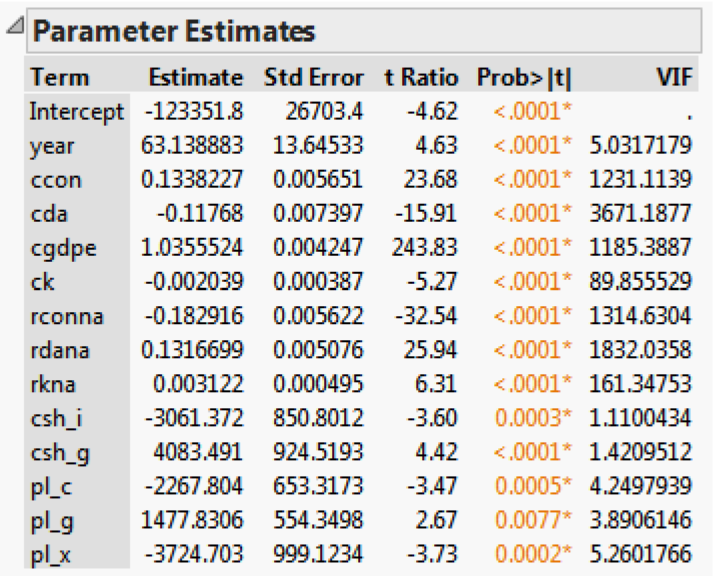
cgdpe: VIF=1185

ck: VIF=90

rconna: VIF=1315

rdana: VIF=1832

rkna: VIF=161

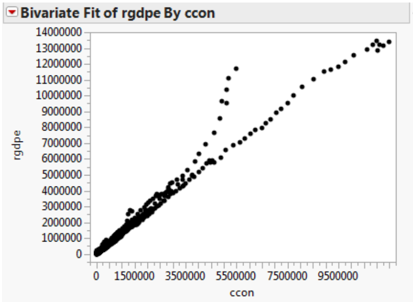


Parameter Estimates

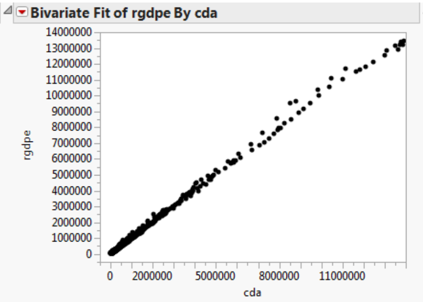
Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	-123351.8	26703.4	-4.62	<.0001*	.
year	63.138883	13.64533	4.63	<.0001*	5.0317179
ccon	0.1338227	0.005651	23.68	<.0001*	1231.1139
cda	-0.11768	0.007397	-15.91	<.0001*	3671.1877
cgdpe	1.0355524	0.004247	243.83	<.0001*	1185.3887
ck	-0.002039	0.000387	-5.27	<.0001*	89.855529
rconna	-0.182916	0.005622	-32.54	<.0001*	1314.6304
rdana	0.1316699	0.005076	25.94	<.0001*	1832.0358
rkna	0.003122	0.000495	6.31	<.0001*	161.34753
csh_i	-3061.372	850.8012	-3.60	0.0003*	1.1100434
csh_g	4083.491	924.5193	4.42	<.0001*	1.4209512
pl_c	-2267.804	653.3173	-3.47	0.0005*	4.2497939
pl_g	1477.8306	554.3498	2.67	0.0077*	3.8906146
pl_x	-3724.703	999.1234	-3.73	0.0002*	5.2601766

Response vs. Explanatory Variables:

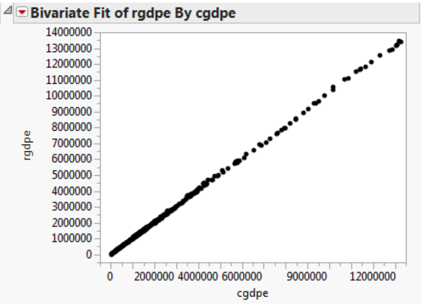
Above: R^2= .9797, R^2=.9980, R^2=.9998



Bivariate Fit of rgdpe By ccon

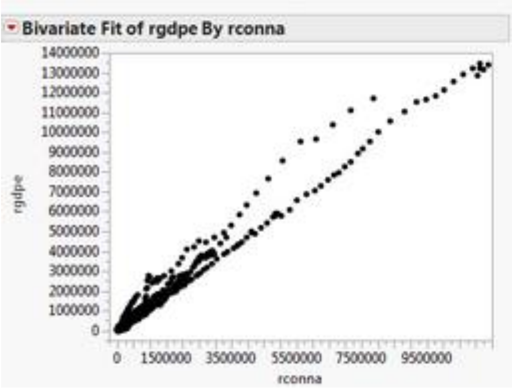


Bivariate Fit of rgdpe By cda

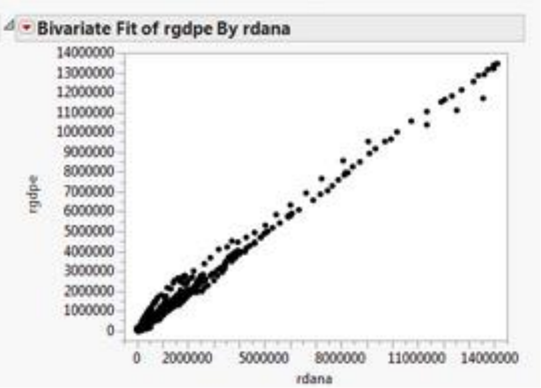


Bivariate Fit of rgdpe By cgdpe

R^2= .9821, R^2=.9917



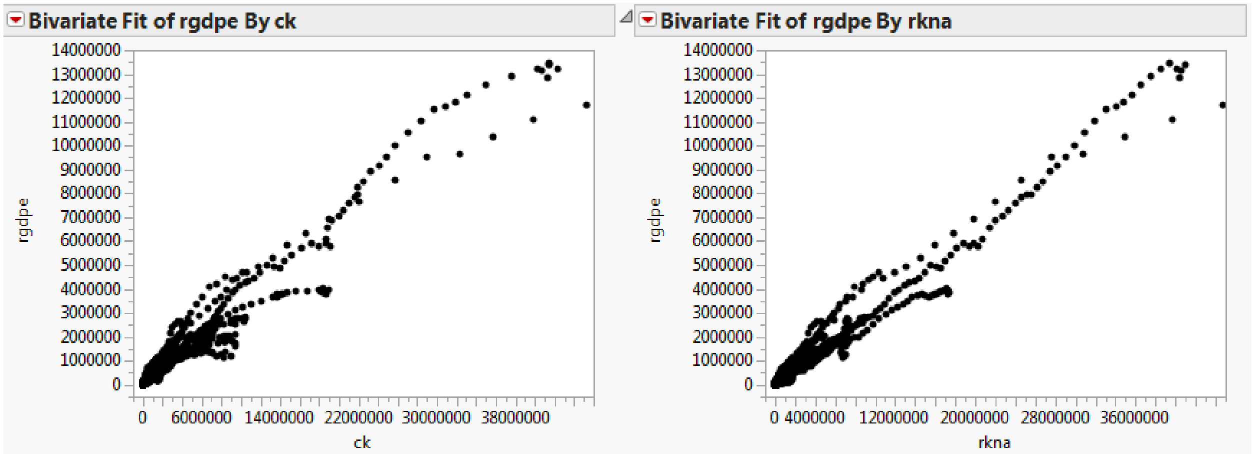
Bivariate Fit of rgdpe By rconna



Bivariate Fit of rgdpe By rdana

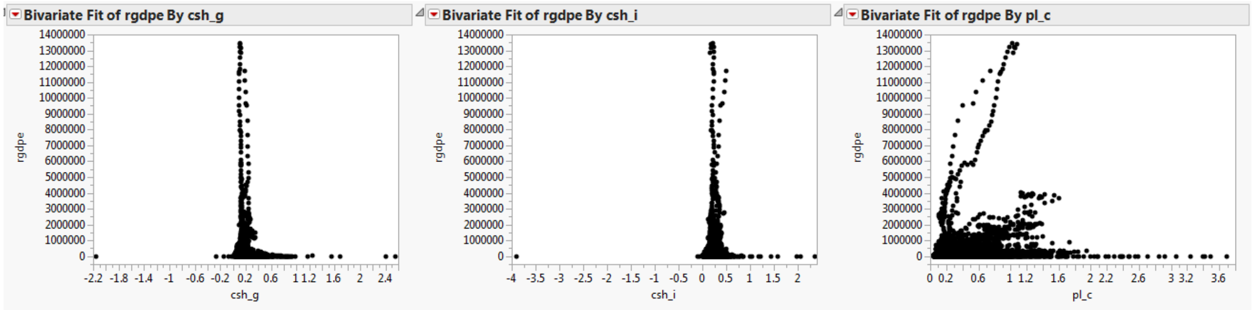
Average R^2 of above trend= 0.990

Middle: R^2= .961, R^2=.9735

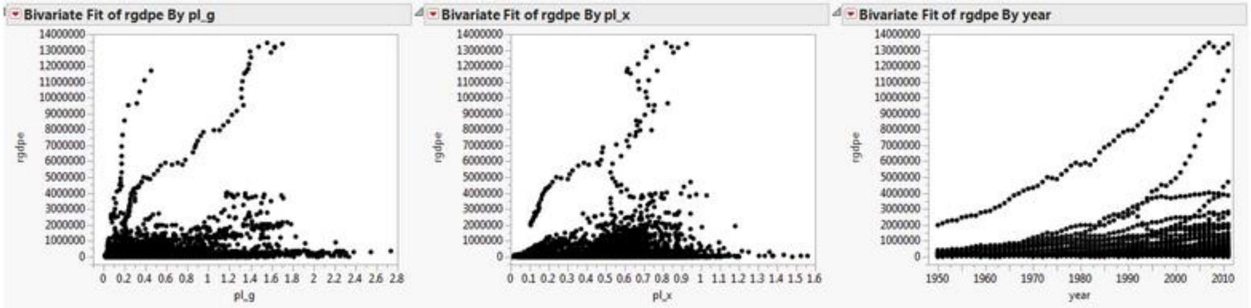


Average R^2 of above trend=.967

Below: R^2= .00756, R^2=.00763, R^2=.0262,



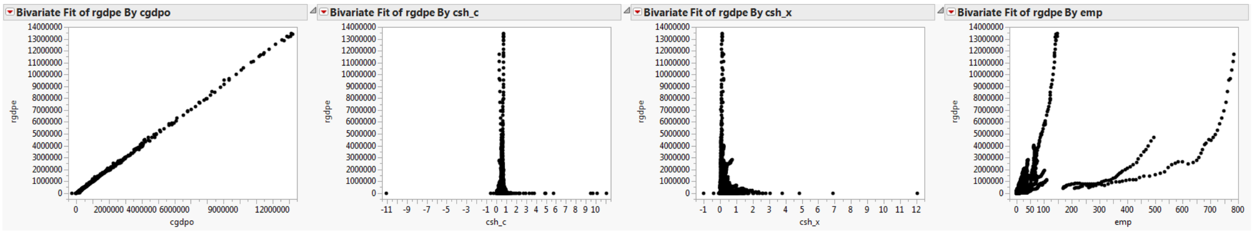
R^2=.0629, R^2=.0181, R^2=.0140



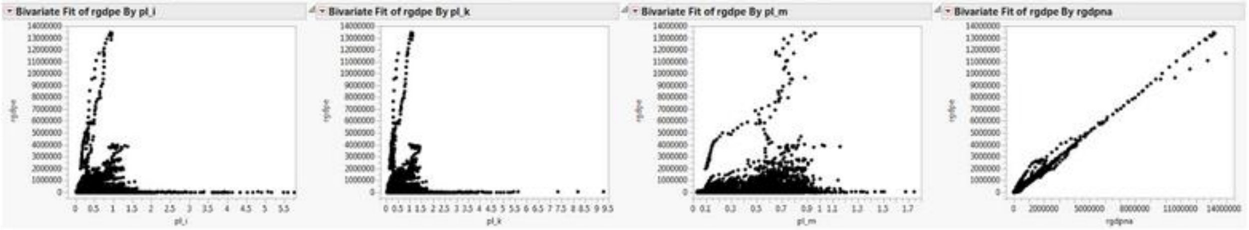
Average R^2 of above trend=.0227

Not in Model:

R^2= .999, R^2=.00181, R^2=.00102, R^2=.3083,



R^2=.00213, R^2<.001, R^2=.0269, R^2=.991

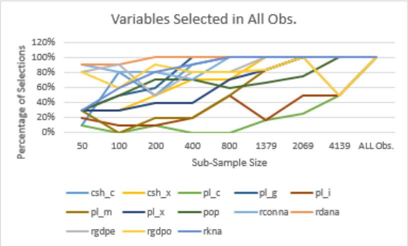


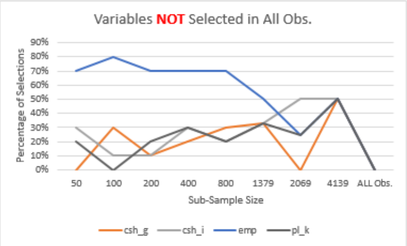
Average R^2 of above trend=.291

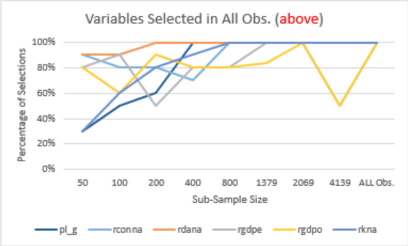
Appendix C5: GDP (2nd Analysis)

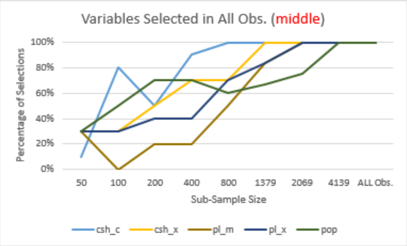
Model before backwards selection:

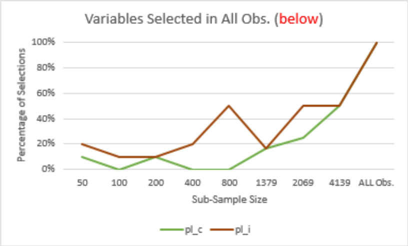
rgdpna = csh_c csh_g csh_i csh_x emp pl_c pl_g pl_i pl_k pl_m pl_x pop rconna rdana rgdpe rgdpo rkna











Variables:

- Not in model (top right): csh_g, csh_i, emp, pl_k
- In model
  - above (bottom left): pl_g, rconna, rdana, rgdpe, rgdpo, rkna
  - below (bottom right): pl_c, pl_i
  - middle (bottom middle): csh_c, csh_x, pl_m, pl_x, pop

Multicollinearity:

rgdpe: VIF= 2567

rgdpo: VIF=2618

rconna: VIF=126

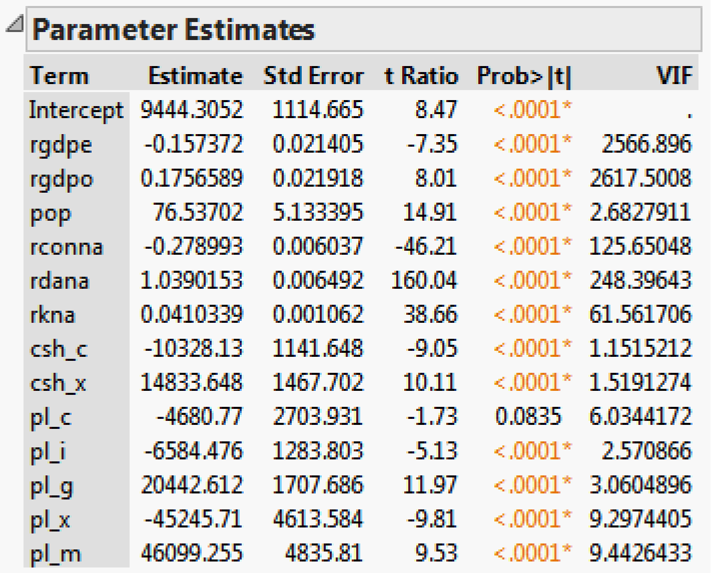
rdana: VIF=248

rkna: VIF=62

pl_x: VIF=9

pl_m: VIF=9

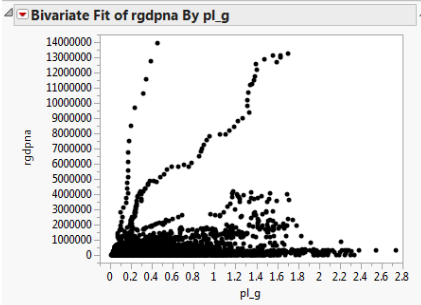
Response vs. Explanatory Variables:

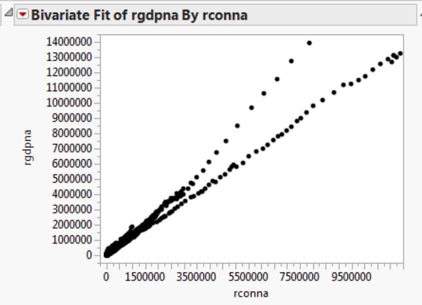


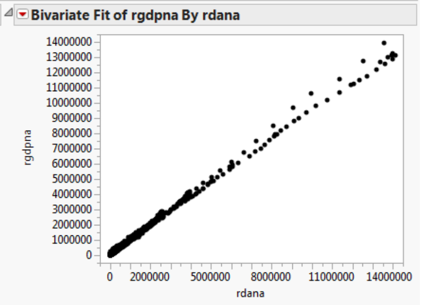
Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	9444.3052	1114.665	8.47	<.0001*	.
rgdpe	-0.157372	0.021405	-7.35	<.0001*	2566.896
rgdpo	0.1756589	0.021918	8.01	<.0001*	2617.5008
pop	76.53702	5.133395	14.91	<.0001*	2.6827911
rconna	-0.278993	0.006037	-46.21	<.0001*	125.65048
rdana	1.0390153	0.006492	160.04	<.0001*	248.39643
rkna	0.0410339	0.001062	38.66	<.0001*	61.561706
csh_c	-10328.13	1141.648	-9.05	<.0001*	1.1515212
csh_x	14833.648	1467.702	10.11	<.0001*	1.5191274
pl_c	-4680.77	2703.931	-1.73	0.0835	6.0344172
pl_i	-6584.476	1283.803	-5.13	<.0001*	2.570866
pl_g	20442.612	1707.686	11.97	<.0001*	3.0604896
pl_x	-45245.71	4613.584	-9.81	<.0001*	9.2974405
pl_m	46099.255	4835.81	9.53	<.0001*	9.4426433

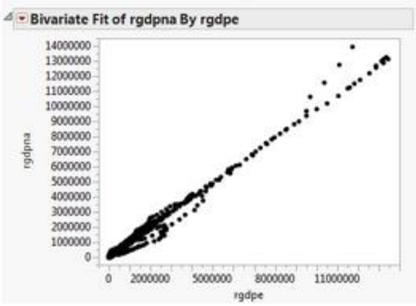
Above: R^2=.06759, R^2=.981, R^2=.9971,

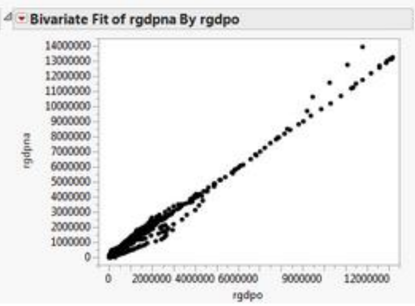


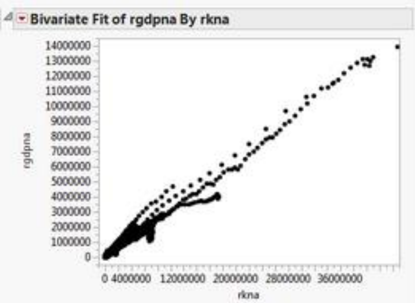




R^2=.9905, R^2=.9906, R^2=.9808

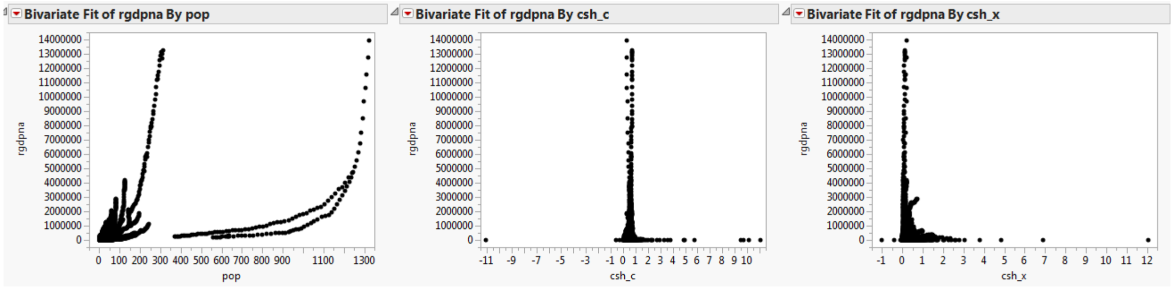




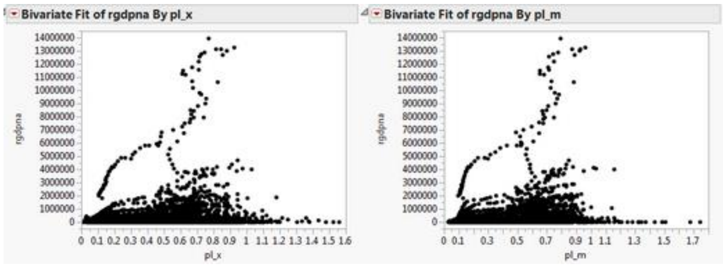


Average R^2 of above trend=.835

Middle: R^2= .2526, R^2=.002126, R^2<.001,

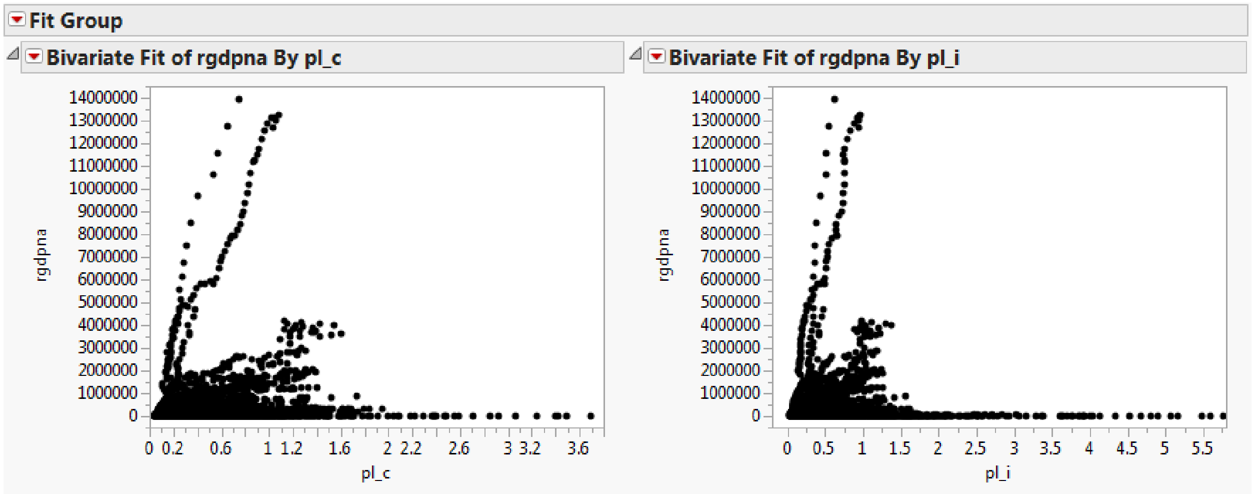


R^2=.01756, R^2=.0266



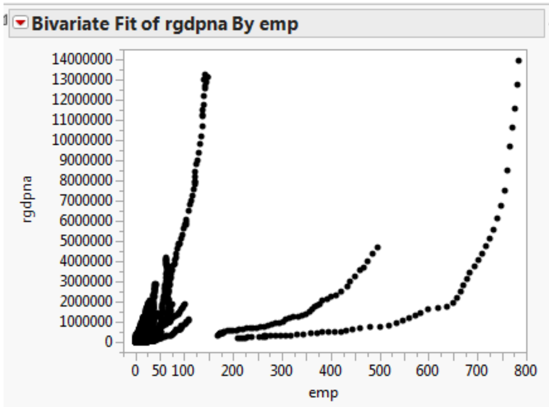
Average R^2 of above trend=.0599

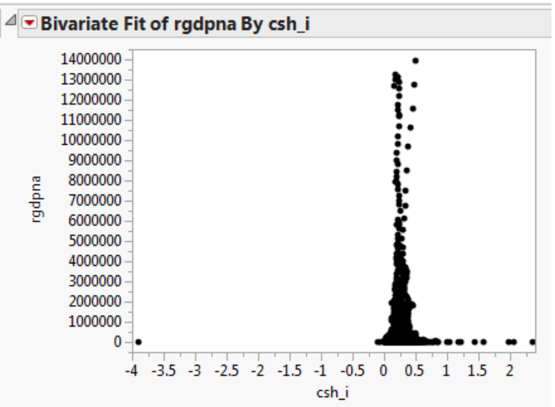
Below: R^2= .02877, R^2=.002464



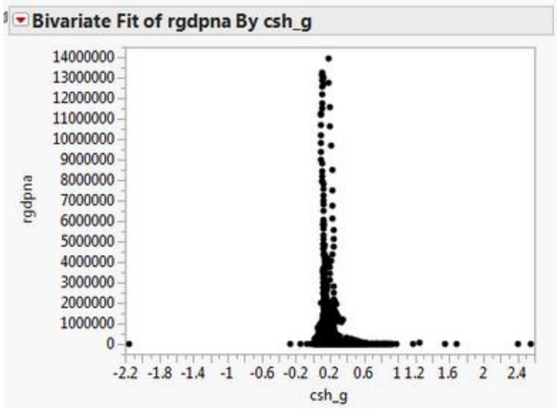
Average R^2 of above trend=.01561

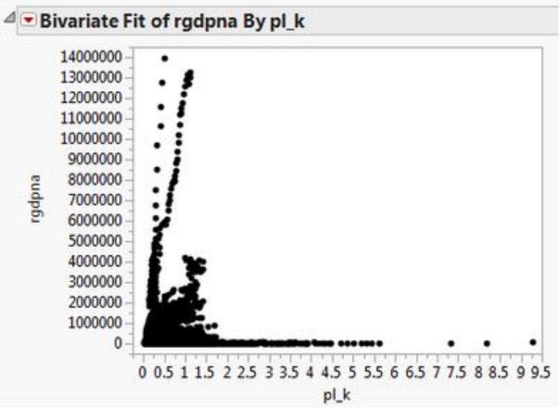
Not in Model: R^2= .2789, R^2=.00875,





R^2=.00837, R^2=.0011



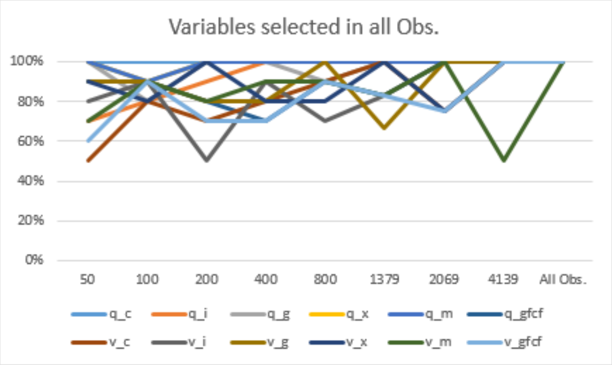


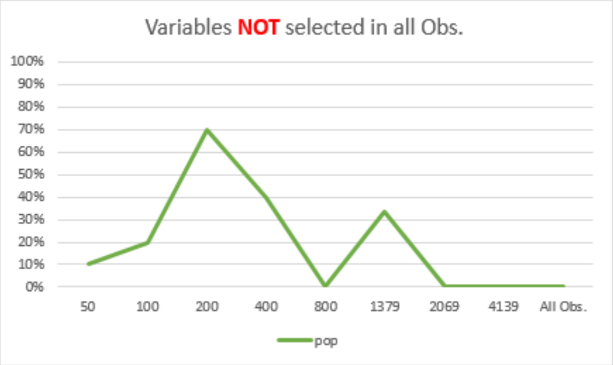
Average R^2 of above trend=.07428

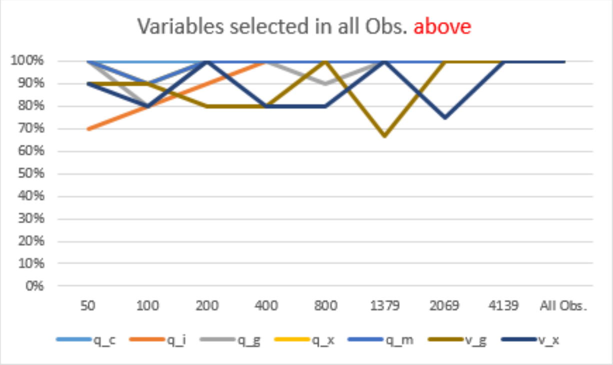
Appendix C6: National Accounts (2nd analysis)

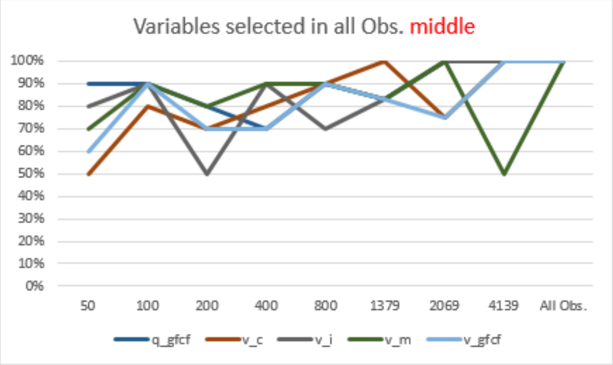
Model before backwards selection:

q_gdp = q_c q_i q_g q_x q_m pop q_gfcf v_c v_i v_g v_x v_m v_gfcf


Variables selected in all Obs.


Variables **NOT** selected in all Obs.

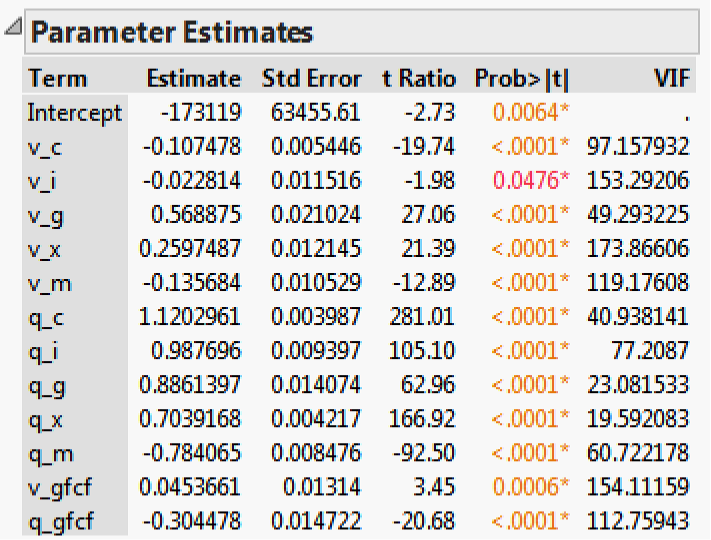

Variables selected in all Obs. above


Variables selected in all Obs. middle

Variables:

- **Not in model (top right): pop**
- **In model**
  - **above (bottom left): q_c, q_i, q_g, q_x, q_m, v_g, v_x**
  - **below: N/A**
  - **middle (bottom right): q_gfcf, v_c, v_i, v_m, v_gfcf**

Multicollinearity: All variables have multicollinearity issues

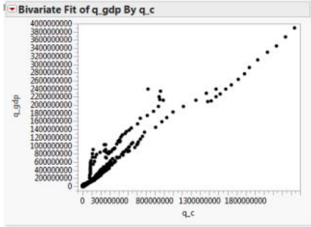


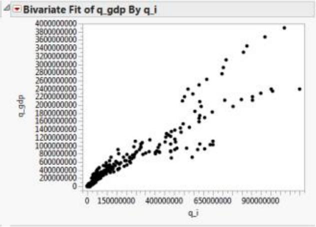
Parameter Estimates

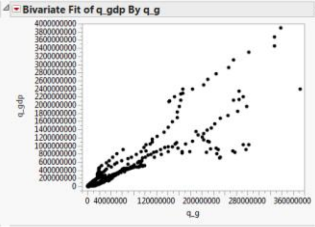
Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	-173119	63455.61	-2.73	0.0064*	.
v_c	-0.107478	0.005446	-19.74	<.0001*	97.157932
v_i	-0.022814	0.011516	-1.98	0.0476*	153.29206
v_g	0.568875	0.021024	27.06	<.0001*	49.293225
v_x	0.2597487	0.012145	21.39	<.0001*	173.86606
v_m	-0.135684	0.010529	-12.89	<.0001*	119.17608
q_c	1.1202961	0.003987	281.01	<.0001*	40.938141
q_i	0.987696	0.009397	105.10	<.0001*	77.2087
q_g	0.8861397	0.014074	62.96	<.0001*	23.081533
q_x	0.7039168	0.004217	166.92	<.0001*	19.592083
q_m	-0.784065	0.008476	-92.50	<.0001*	60.722178
v_gfcf	0.0453661	0.01314	3.45	0.0006*	154.11159
q_gfcf	-0.304478	0.014722	-20.68	<.0001*	112.75943

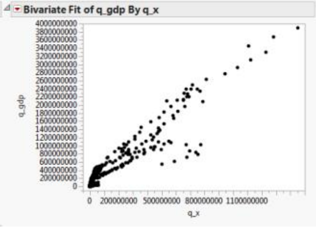
Response vs. Explanatory Variables:

Above: R^2=.945, R^2=.918, R^2=.848, R^2=.916,

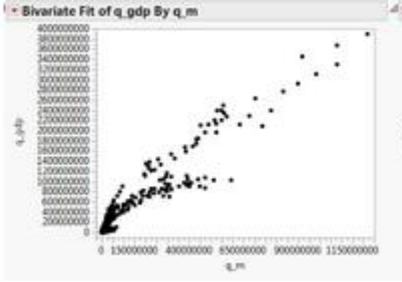


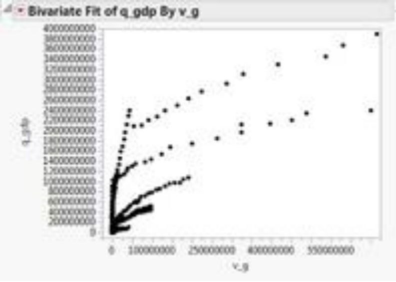


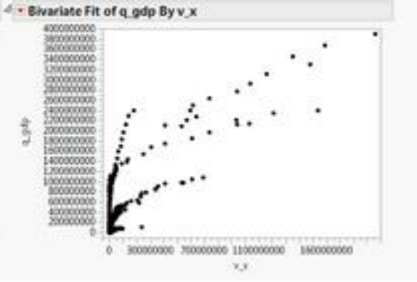




R^2=.930, R^2=.626, R^2=.665

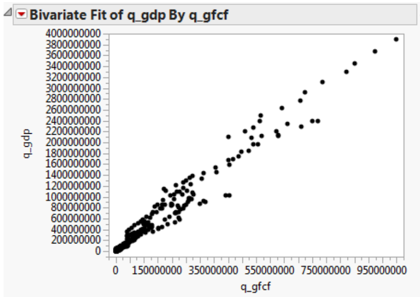




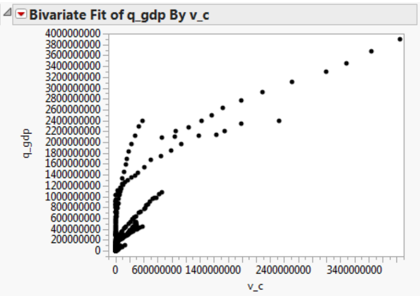


Average R^2 of above trend=.836

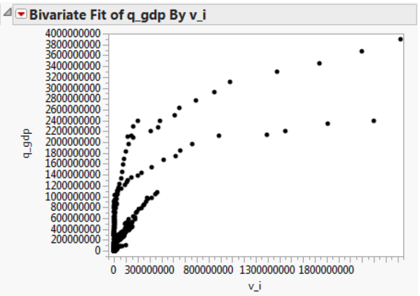
Middle: R^2= .977,



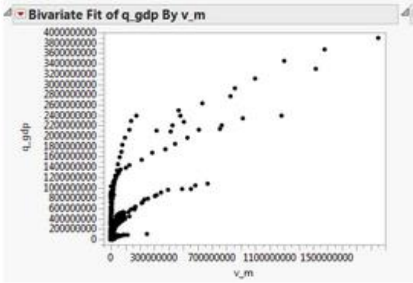
R^2=.677,



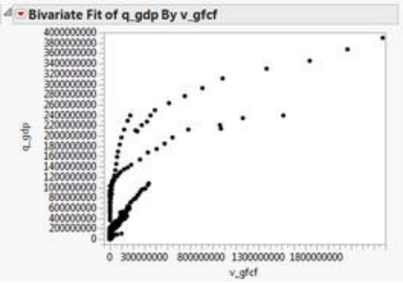
R^2=.587,



R^2=.657,



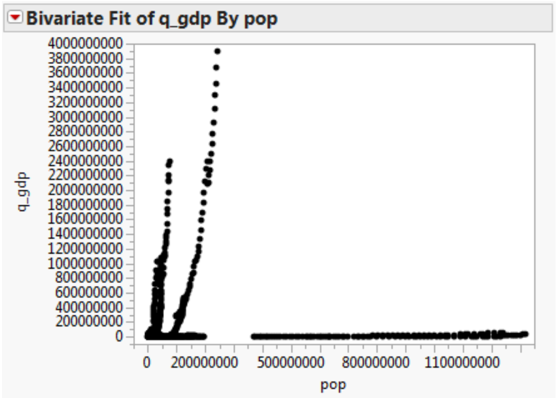
R^2=.623



Average R^2 of above trend=.704

Below: N/A

Not in Model: R^2= .006645

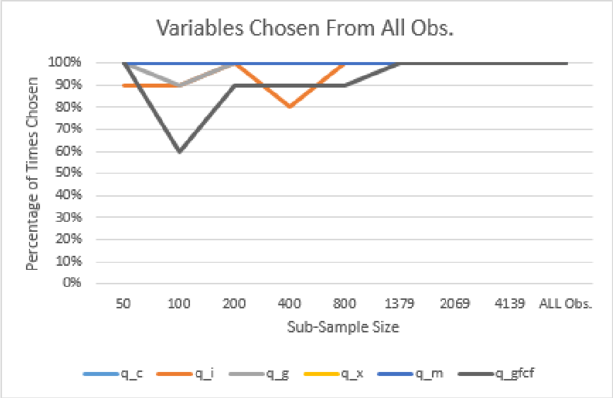


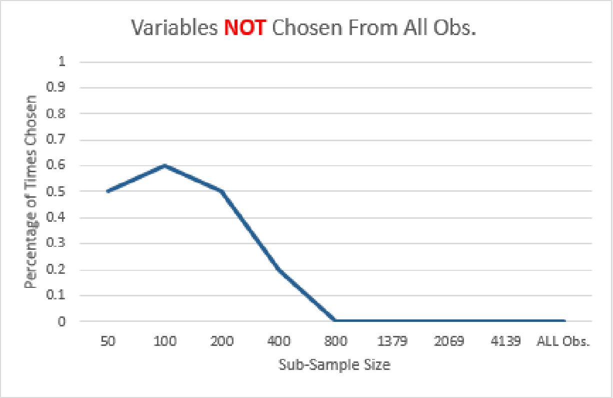
Average R^2 of above trend=.00665

Appendix C7: National Accounts (1st analysis)

Model before backwards selection:

q_gdp = q_c q_i q_g q_x q_m pop q_gfcf

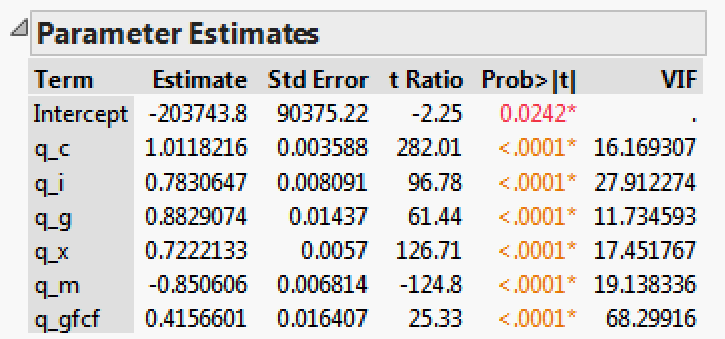




Variables:

- Not in model (right): pop
- In model
    - above (left): q_c, q_i, q_g, q_x, q_m, q_gfcf
    - below: N/A
    - middle: N/A

Multicollinearity: All variables have multicollinearity issues

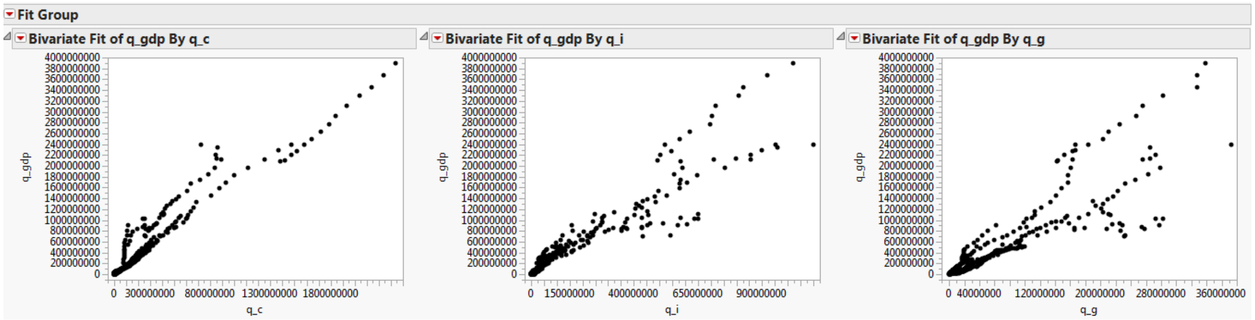


Parameter Estimates

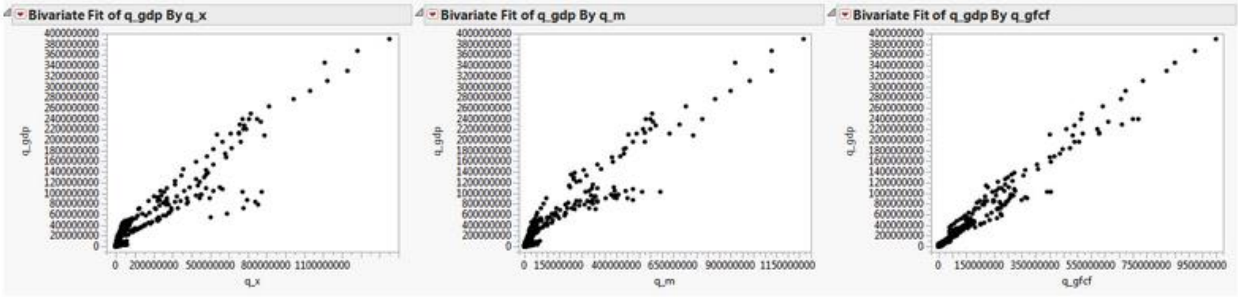
Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	-203743.8	90375.22	-2.25	0.0242*	.
q_c	1.0118216	0.003588	282.01	<.0001*	16.169307
q_i	0.7830647	0.008091	96.78	<.0001*	27.912274
q_g	0.8829074	0.01437	61.44	<.0001*	11.734593
q_x	0.7222133	0.0057	126.71	<.0001*	17.451767
q_m	-0.850606	0.006814	-124.8	<.0001*	19.138336
q_gfcf	0.4156601	0.016407	25.33	<.0001*	68.29916

Response vs. Explanatory Variables:

Above: R^2= .945, R^2=.918, R^2=.848,



R^2=.916, R^2=.930, R^2=.977

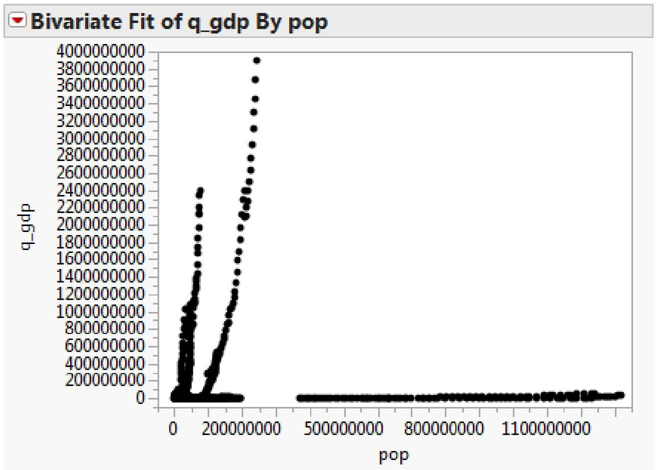


Average R^2 of above trend=.922

Middle: N/A

Below: N/A

Not in Model: R^2= .006645

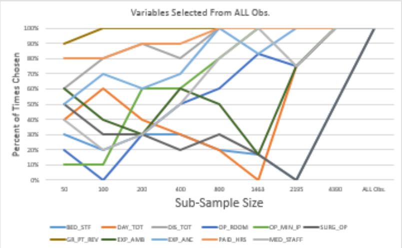


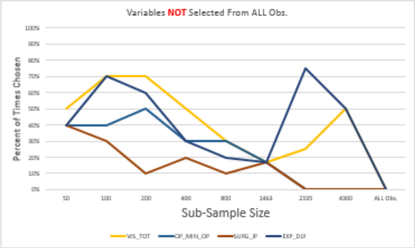
Average R^2 of above trend=.00665

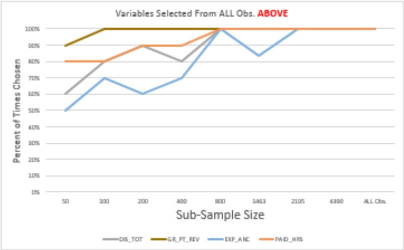
Appendix C8: Medical Operations (2^{nd} analysis)

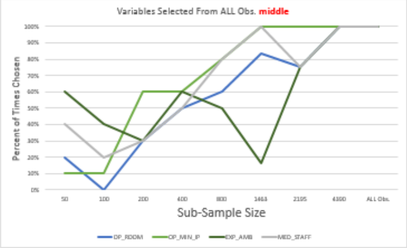
Model before backwards selection:

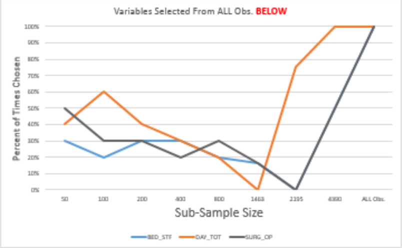
TOT_OP_EXP = BED_STF DAY_TOT DIS_TOT VIS_TOT OP_ROOM OP_MIN_IP OP_MIN_OP SURG_IP SURG_OP GR_PT_REV EXP_DLY EXP_AMB EXP_ANC PAID_HRS MED_STAFF


Variables Selected From ALL Obs.


Variables NOT Selected From ALL Obs.


Variables Selected From ALL Obs. ABOVE


Variables Selected From ALL Obs. middle

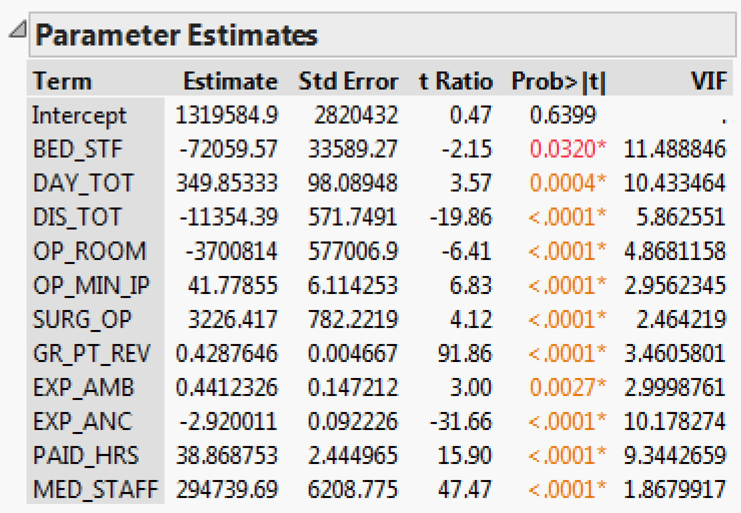

Variables Selected From ALL Obs. BELOW

Variables:

- Not in model (top right): VIS_TOT, OP_MIN_OP, SURG_ IP, EXP_DLY
- In model
  - above (bottom left): DIS_TOT, GR_PT_REV, EXP_ANC, PAID_HRS
  - below (bottom right): BED_STF, DAY_TOT, SURG_OP
  - middle (bottom middle): OP_ROOM, OP_MIN_IP, EXP_AMB, MED_STAFF

Multicollinearity:

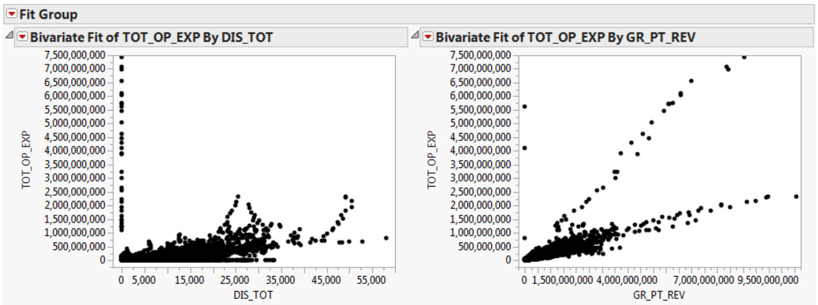
BED_STF: VIF= 11.5
DAY_TOT: VIF=10.4
EXP_ANC: VIF=10.2
PAID_HRS: VIF=9.3


Parameter Estimates

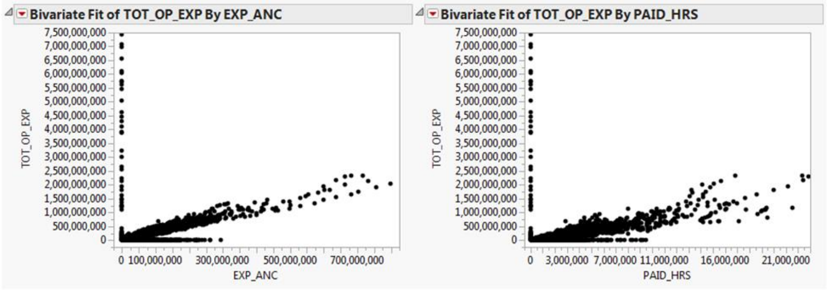
Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	1319584.9	2820432	0.47	0.6399	.
BED_STF	-72059.57	33589.27	-2.15	0.0320*	11.488846
DAY_TOT	349.85333	98.08948	3.57	0.0004*	10.433464
DIS_TOT	-11354.39	571.7491	-19.86	<.0001*	5.862551
OP_ROOM	-3700814	577006.9	-6.41	<.0001*	4.8681158
OP_MIN_IP	41.77855	6.114253	6.83	<.0001*	2.9562345
SURG_OP	3226.417	782.2219	4.12	<.0001*	2.464219
GR_PT_REV	0.4287646	0.004667	91.86	<.0001*	3.4605801
EXP_AMB	0.4412326	0.147212	3.00	0.0027*	2.9998761
EXP_ANC	-2.920011	0.092226	-31.66	<.0001*	10.178274
PAID_HRS	38.868753	2.444965	15.90	<.0001*	9.3442659
MED_STAFF	294739.69	6208.775	47.47	<.0001*	1.8679917

Response vs. Explanatory Variables:

Above: R^2=.12, R^2=.59,

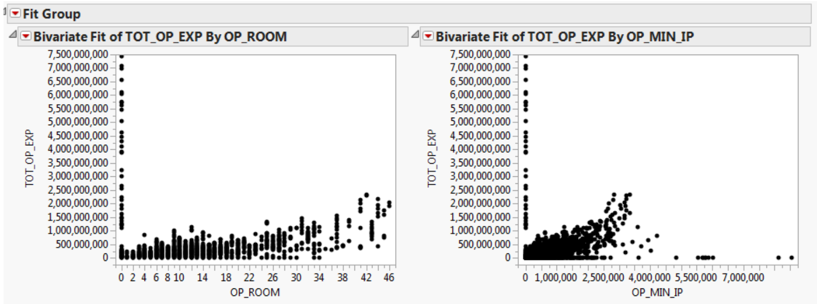


R^2=.23, R^2=.21

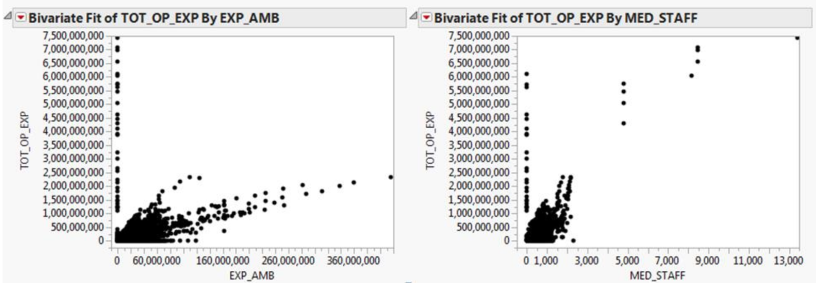


Average R^2 of above trend=.29

Middle: R^2= .12, R^2=.12,

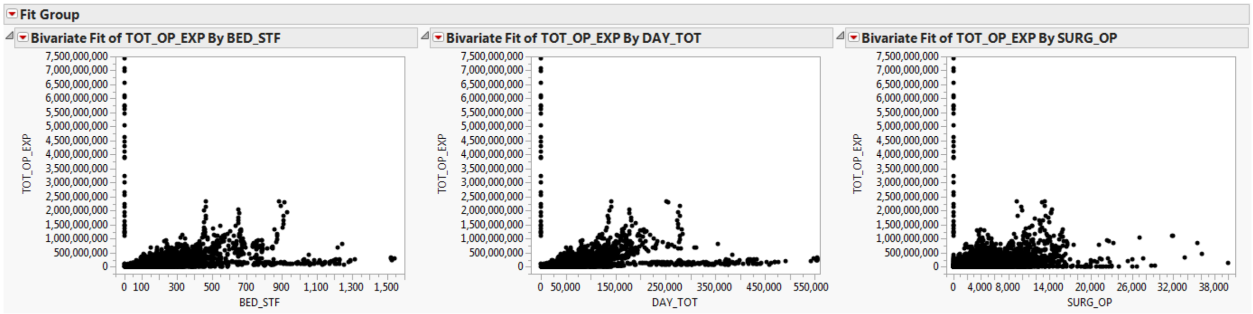


R^2=.16, R^2=.39



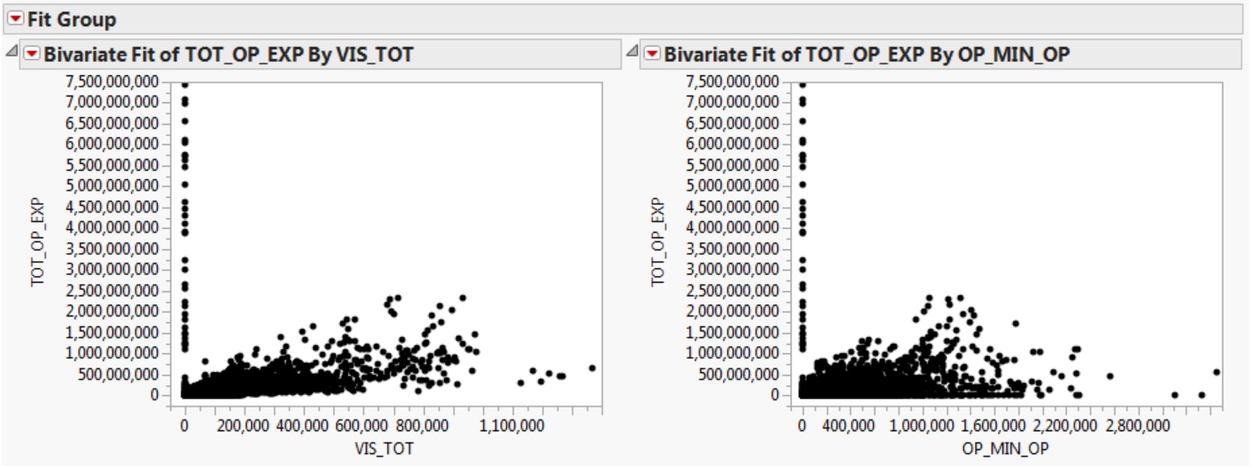
Average R^2 of above trend=.20

Below: R^2= .07, R^2=.06, R^2=.05

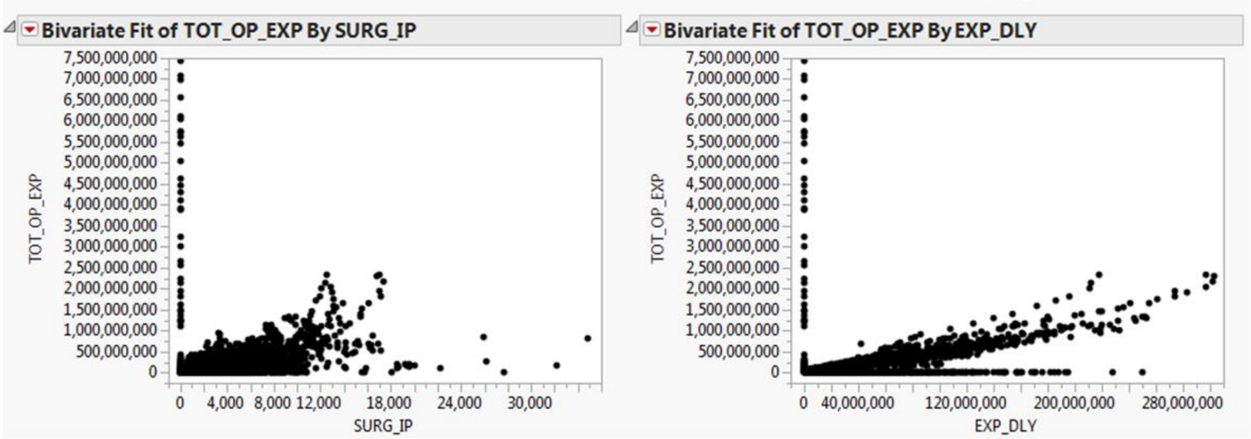


Average R^2 of above trend=.06

Not in Model: R^2=.15, R^2=.06,



R^2=.12, R^2=.19

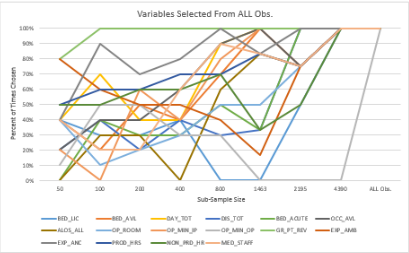


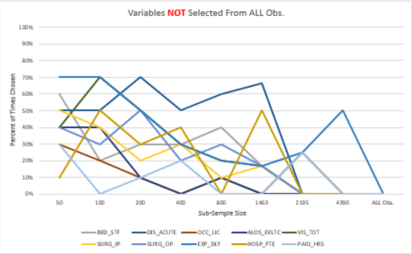
Average R^2 of above trend=.13

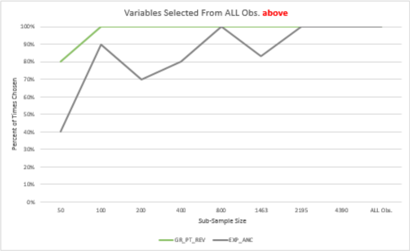
Appendix C9: Medical Operations (1st analysis)

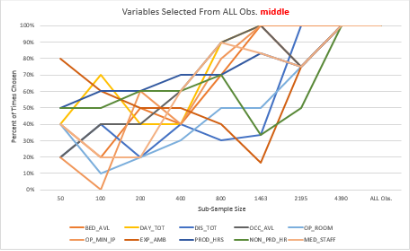
Model before backwards selection:

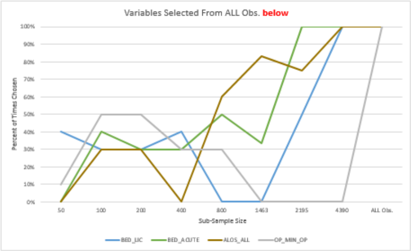
TOT_OP_EXP = BED_LIC BED_AVL BED_STF DAY_TOT DIS_TOT BED_ACUTE DIS_ACUTE OCC_LIC OCC_AVL ALOS_ALL ALOS_EXLTC VIS_TOT OP_ROOM OP_MIN_IP OP_MIN_OP SURG_IP SURG_OP GR_PT_REV EXP_DLY EXP_AMB EXP_ANC HOSP_FTE PROD_HRS NON_PRD_HR PAID_HR MED_STAFF











Variables:

- Not in model (top right): BED_STF, DIS_ACUTE, OCC_LIC, ALOS_EXLTC, VIS_TOT, SURG_IP, SURG_OP, EXP_DLY, HOSP_FTE, PAID_HRS
- In model
  - above (bottom left): GR_PT_REV, EXP_ANC
  - below (bottom right): BED_LIC, BED_ACUTE, ALOS_ALL, OP_MIN_OP
  - middle (bottom middle): BED_AVL, DAY_TOT, DIS_TOT, OCC_AVL, OP_ROOM, OP_MIN_IP, EXP_AMB, PROD_HRS, NON_PRD_HR, MED_STAFF

Multicollinearity:

BED_LIC: **VIF= 25.5**

BED_AVL: **VIF=28.2**

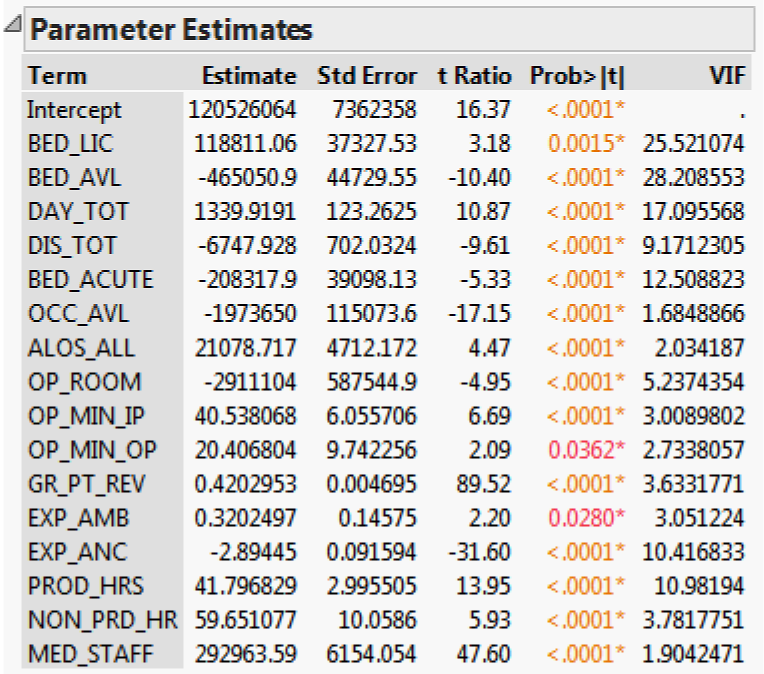
DAY_TOT: **VIF=17.1**

DIS_TOT: **VIF=9.2**

BED_ACUTE: **VIF=12.5**

EXP_ANC: **VIF=10.4**

PROD_HRS: **VIF=11.0**

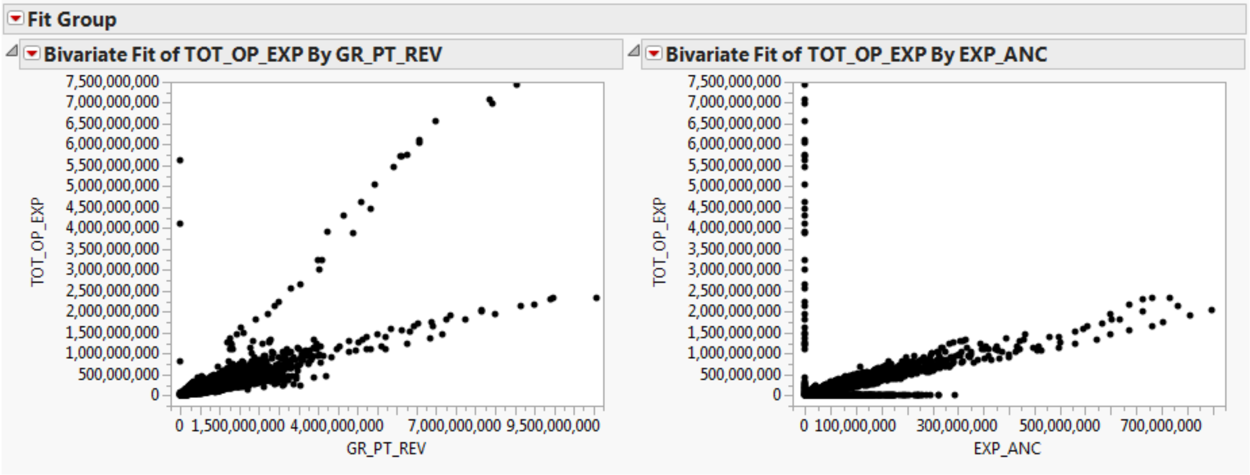


Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	120526064	7362358	16.37	<.0001*	.
BED_LIC	118811.06	37327.53	3.18	0.0015*	25.521074
BED_AVL	-465050.9	44729.55	-10.40	<.0001*	28.208553
DAY_TOT	1339.9191	123.2625	10.87	<.0001*	17.095568
DIS_TOT	-6747.928	702.0324	-9.61	<.0001*	9.1712305
BED_ACUTE	-208317.9	39098.13	-5.33	<.0001*	12.508823
OCC_AVL	-1973650	115073.6	-17.15	<.0001*	1.6848866
ALOS_ALL	21078.717	4712.172	4.47	<.0001*	2.034187
OP_ROOM	-2911104	587544.9	-4.95	<.0001*	5.2374354
OP_MIN_IP	40.538068	6.055706	6.69	<.0001*	3.0089802
OP_MIN_OP	20.406804	9.742256	2.09	0.0362*	2.7338057
GR_PT_REV	0.4202953	0.004695	89.52	<.0001*	3.6331771
EXP_AMB	0.3202497	0.14575	2.20	0.0280*	3.051224
EXP_ANC	-2.89445	0.091594	-31.60	<.0001*	10.416833
PROD_HRS	41.796829	2.995505	13.95	<.0001*	10.98194
NON_PRD_HR	59.651077	10.0586	5.93	<.0001*	3.7817751
MED_STAFF	292963.59	6154.054	47.60	<.0001*	1.9042471

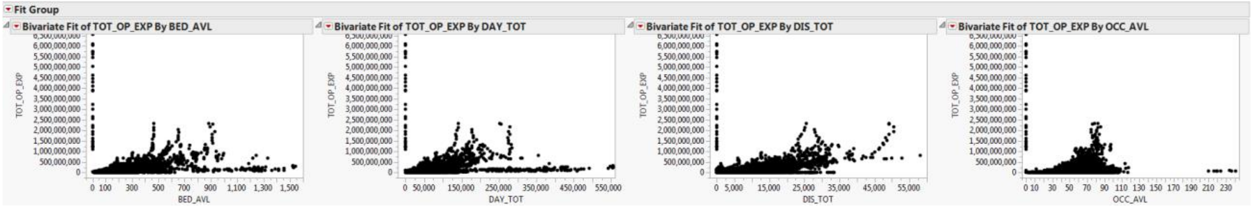
Response vs. Explanatory Variables:

Above: R^2=.592, R^2=.226/

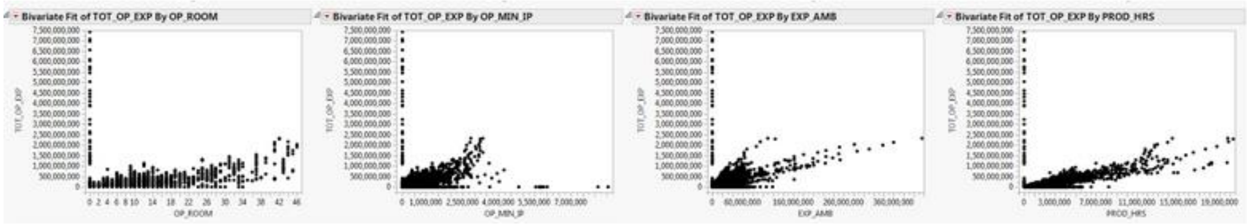


Average R^2 of above trend=.409

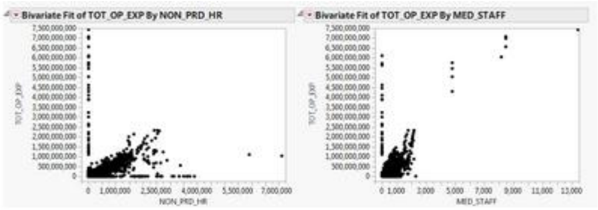
Middle: R^2= .0639, R^2=.0649, R^2=.122, R^2=.0012



R^2=.121, R^2=.115, R^2=.165, R^2=.209,

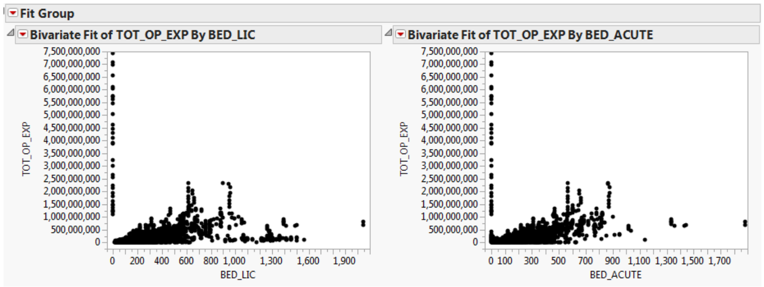


R^2=.141, R^2=.392

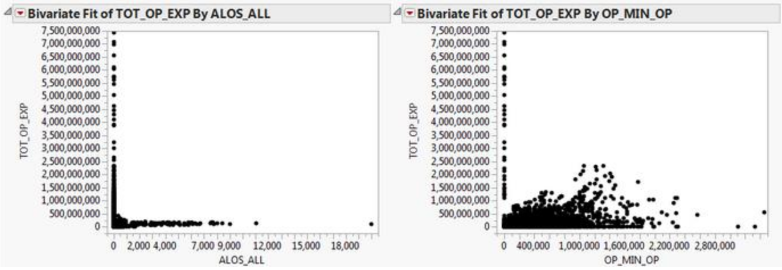


Average R^2 of above trend=0.1395

Below: R^2=.0629, R^2=.121



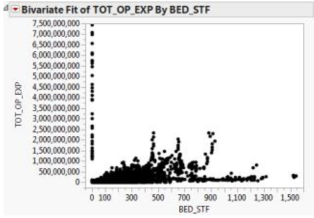
R^2< .001, R^2=.0644

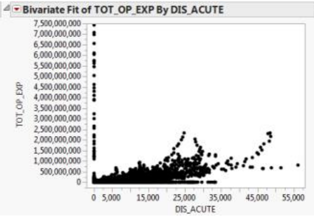


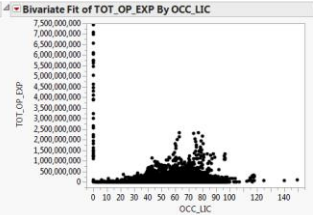
Average R^2 of above trend=.0621

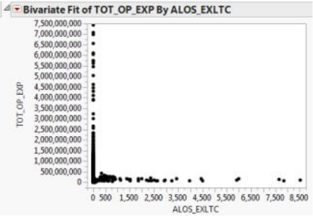
Not in Model:

R^2= .0661, R^2=.122, R^2=.0036, R^2<.001,

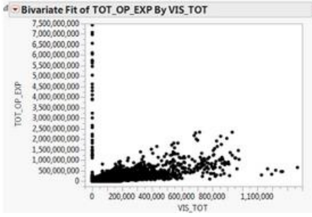


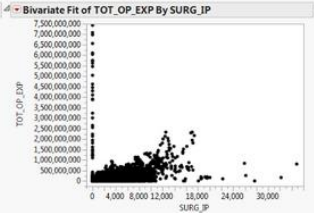


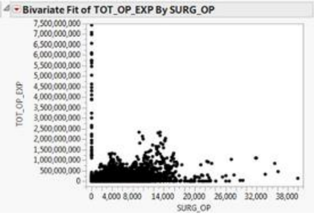


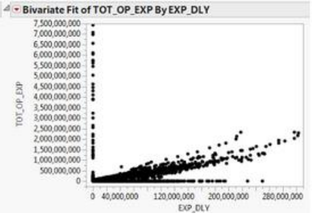


R^2=.1485, R^2=.121, R^2=.0545, R^2=.1872,

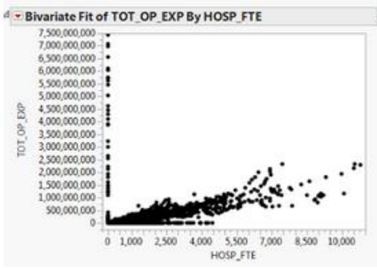


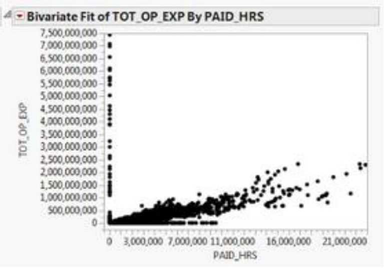






R^2=.2045, R^2=.2065





Average R^2 of above trend= .1113