Assignment #6

Andrea Bruckner

Introduction

In this report we are using the stock portfolio data set to explore the use of Principal Components Analysis (PCA) as a method of dimension reduction and a remedy of multicollinearity in Ordinary Least Squares (OLS) regression. We used the log-returns of 20 individual stocks to explain the variation in the log-returns of the Vanguard market index fund, and we fit a couple regression models to compare how a model created using PCA differs from a model using all of the individual stocks. We assessed the goodness-of-fit (GOF) of each model using diagnostic plots and statistical tables, examined the models for the presence of multicollinearity by assessing the variance inflation factors (VIFs), and assessed the predictive accuracy of our models by cross-validating the training set statistics with the testing set statistics. By comparing the MAE and MSE statistics between the training set and testing set for both of our models, we discovered that both models performed better in-sample that out-of-sample. We found that the model using all the stocks and the model using just eight principal components both produced very similar diagnostic plots and statistics, with the exception being that the model created using PCA had a much higher F-statistic and consistently lower VIFs than the all stocks model. Given that the models had very similar outputs aside from the F-statistic and VIFs, we used these two measures to determine that the PCA model helped correct the multicollinearity in the all stocks model and was therefore more reliable and a better fit.

Data

The stock portfolio data set consists of the daily closing stock prices for twenty stocks, shown below in **Table 1**, as well as a large-cap index fund from Vanguard. It contains 502 observations with 22 variables concerning stock prices between January 2012 and December 2013. All the variables in the data set are continuous, except for the date variable, which is ordinal.

Ticker Symbol	Name	Ticker Symbol	Name
AA	Alcoa Inc.	HON	Honeywell International Inc.
BAC	Bank of America Corporation	HUN	Huntsman Corporation
BHI	Baker Hughes Incorporated	JPM	JPMorgan Chase & Co.
CVX	Chevron Corporation	KO	The Coca-Cola Company
DD	E. I. du Pont de Nemours and Company	MMM	3M Company
DOW	The Dow Chemical Company	MPC	Marathon Petroleum Corporation
DPS	Dr Pepper Snapple Group, Inc.	PEP	Pepsico, Inc.
GS	The Goldman Sachs Group, Inc.	SLB	Schlumberger Limited
HAL	Halliburton Company	WFC	Wells Fargo & Company
HES	Hess Corporation	XOM	Exxon Mobil Corporation

Table 1: Stocks in Portfolio

Examining Correlations Between Individual Stocks and the Market Index

In order to try to explain the variation in the market index, we computed the log-returns (i.e., the ratio of the current day's price with the previous day's price) of the 20 individual stocks and the

large-cap index fund, Vanguard. **Figure 1** shows the correlations between the individual stocks, identified by their ticker symbols, with the market index. The bar chart reveals that the Manufacturing sector stocks have the strongest correlation with the market index, while the Soft Drinks sector stocks have the weakest correlation. Most of the stocks regardless of sector have a 60 percent or greater positive correlation with the market index.

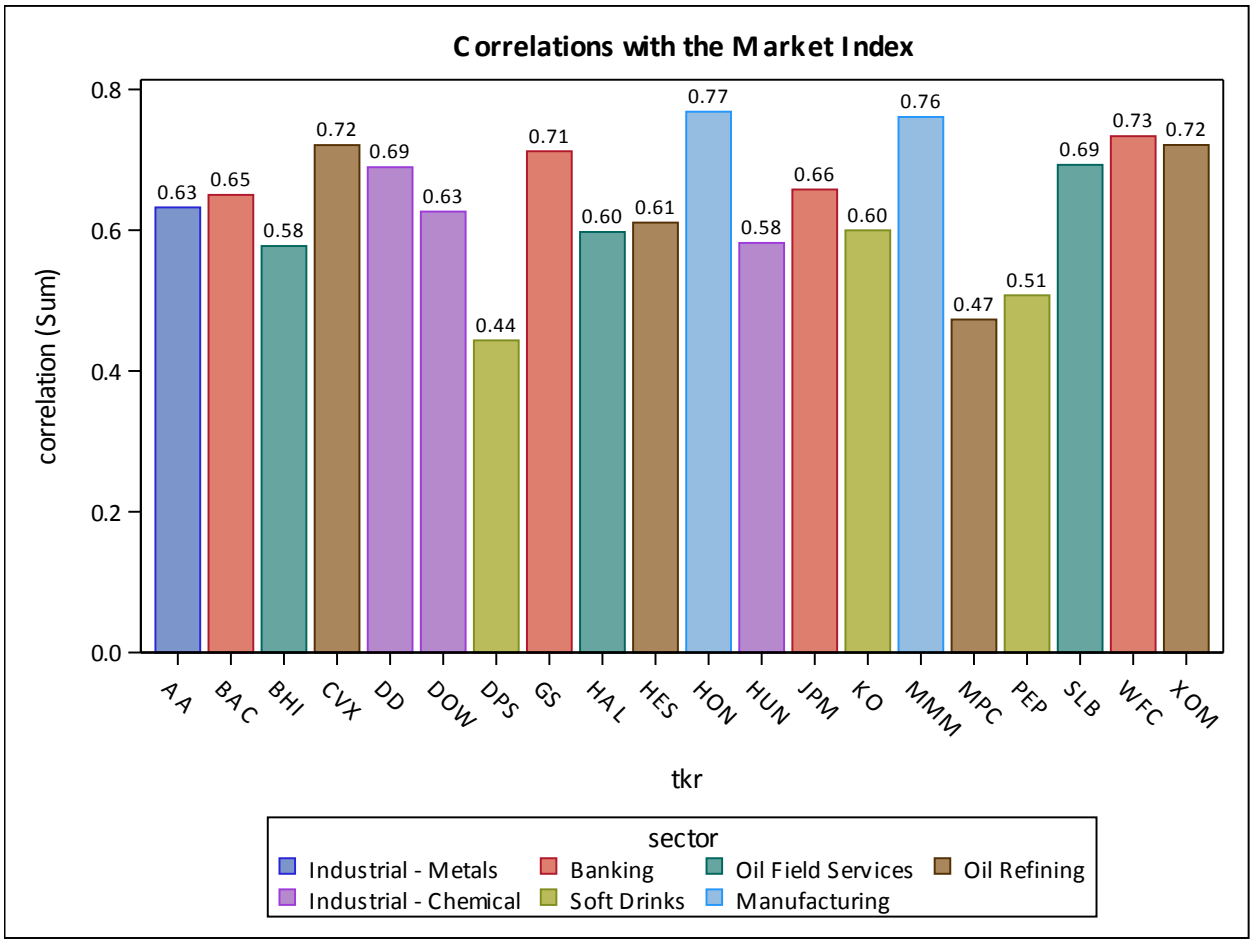

Figure 1: Pearson Correlation Coefficients Between the Log-Returns of Individual Stocks and the Log-Returns of the Vanguard Large-Cap Index Fund

Not all sectors are equally represented in the bar chart above, which is something to keep in mind if we are trying to make an inference about an entire sector from just the data in the stock portfolio. Although the stocks in the Manufacturing sector exhibit the strongest correlation with the market index based on **Figure 1**, adding more stocks from this sector or changing some of the stocks in the other sectors could indicate a different sector having the strongest correlation with the market index.

Selecting the Number of Principal Components

In **Table 2**, we can see that the first principal component accounts for 48.18 percent, or nearly half, of the variance in the data. The second principal component accounts 7.69 percent of the variance in the data. Together, the first two principal components capture 55.87 percent of the total variance in the data. The rest of the table shows how much variance each of the remaining principal components captures in the data.

Eigenvalues of the Correlation Matrix				
	Eigenvalue	Difference	Proportion	Cumulative
1	9.63645075	8.09792128	0.4818	0.4818
2	1.53852947	0.19109235	0.0769	0.5587
3	1.34743712	0.39975791	0.0674	0.6261
4	0.94767921	0.15217268	0.0474	0.6735
5	0.79550653	0.12909860	0.0398	0.7133
6	0.66640793	0.10798740	0.0333	0.7466
7	0.55842052	0.04567198	0.0279	0.7745
8	0.51274854	0.01590728	0.0256	0.8002
9	0.49684126	0.03250822	0.0248	0.8250
10	0.46433304	0.03089374	0.0232	0.8482
11	0.43343929	0.02568332	0.0217	0.8699
12	0.40775598	0.05667006	0.0204	0.8903
13	0.35108592	0.01597897	0.0176	0.9078
14	0.33510695	0.03813712	0.0168	0.9246
15	0.29696984	0.02068234	0.0148	0.9394
16	0.27628750	0.01692712	0.0138	0.9532
17	0.25936037	0.01730228	0.0130	0.9662
18	0.24205809	0.02020002	0.0121	0.9783
19	0.22185807	0.01013445	0.0111	0.9894
20	0.21172363		0.0106	1.0000

Table 2: Eigenvalues of the Stock Returns Correlation Matrix

Deciding how many principal components to keep depends on the application of this knowledge. The goal of PCA is to capture the most variance with the fewest number of components. In general, we should aim to capture at least 70 percent of the variance.[1] One of the ways we can decide on how many principal components to keep is by examining a scree plot to see where the plot begins to form an "elbow." The scree plot for the values represented in the table above is shown in **Figure 2**. The "elbow" in the plot falls around the eighth principal component. Prior to the "elbow," the line is steeper and more curved. However, the line begins to flatten out around the eighth principal component, indicating that the successive components account for less and less of the total variance in the data. As **Table 2** shows, the first eight principal components each capture at least 2.5 percent of the variance, while the remaining components capture less. Since this particular data set deals with returns on stock investments, keeping more than eight principal components would result in diminishing returns. **Figure 3** supports this by showing that the line for the cumulative variance explained is more curved and rises more quickly for the first eight components but has a gentler slope for the remaining 12 components.

[1] Everitt, B. (2009). *Multivariable Modeling and Multivariate Analysis for the Behavioral Sciences*. Boca Raton, FL: CRC Press

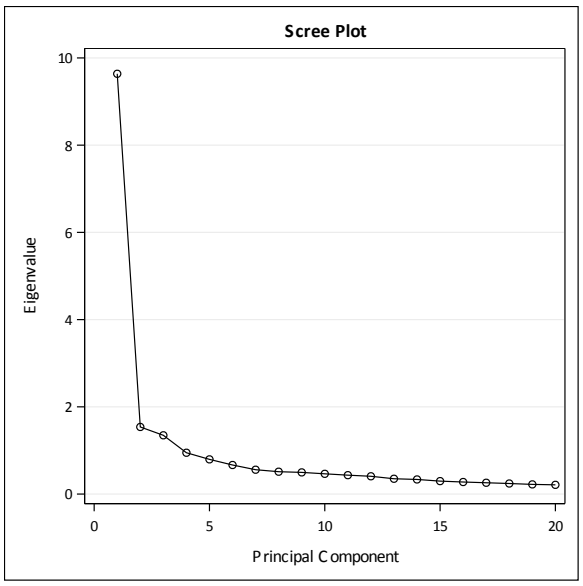

Figure 2: Scree Plot of Principal Components

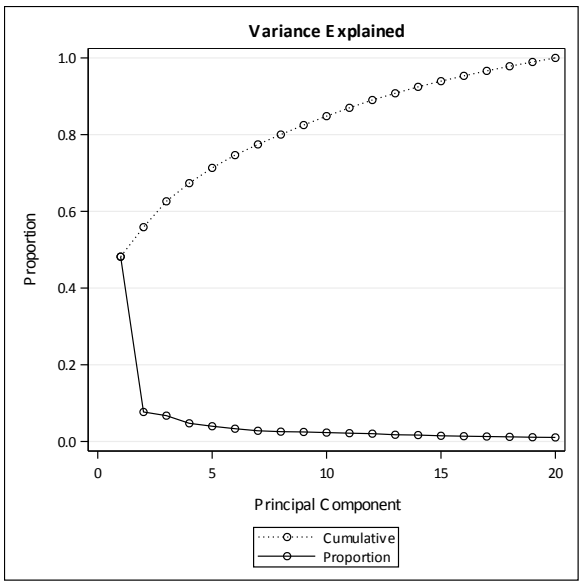

Figure 3: Plot of Variance Explained

By keeping eight principal components, we are able to reduce the dimensions of the data by more than half yet still explain approximately 80 percent of the variance in the data.

A Closer Look at the First Two Principal Components

As we discussed in the previous section, we decided that keeping eight principal components was a good way to reduce the stock portfolio data dimensions while retaining most of the variance in the data. We learned from **Table 2** that the first two principal components capture a cumulative 55.87 percent of the total variance in the data. Since together these components account for more than half of the variance in the data, we decided to take a closer look at them to better understand the relationships in the data.

Obs	Prin1	Prin2	Ticker Symbol	Market Sector	Obs	Prin1	Prin2	Ticker Symbol	Market Sector
1	0.22895	-0.15741	AA	Industrial - Metals	**11**	0.24796	0.06892	HON	Manufacturing
2	0.22464	-0.20081	BAC	Banking	**12**	0.19747	-0.14011	HUN	Industrial - Chemical
3	0.21862	-0.15017	BHI	Oil Field Services	**13**	0.22735	-0.10656	JPM	Banking
4	0.25320	0.09116	CVX	Oil Refining	**14**	0.18820	0.47871	KO	Soft Drinks
5	0.24140	0.03434	DD	Industrial - Chemical	**15**	0.25149	0.13961	MMM	Manufacturing
6	0.21430	-0.09442	DOW	Industrial - Chemical	**16**	0.17114	-0.04824	MPC	Oil Refining
7	0.13862	0.52040	DPS	Soft Drinks	**17**	0.17021	0.48760	PEP	Soft Drinks
8	0.25075	-0.19111	GS	Banking	**18**	0.24973	-0.13757	SLB	Oil Field Services
9	0.22812	-0.14601	HAL	Oil Field Services	**19**	0.24282	-0.06011	WFC	Banking
10	0.22843	-0.10832	HES	Oil Refining	**20**	0.25354	0.08085	XOM	Oil Refining

Table 3: First Two Principal Components for Individual Stocks

In **Table 3** above, we can see that the first principal component, Prin1, has a much higher value than Prin2 for all stocks except for the stocks in the Soft Drinks sector. For these three stocks,

Prin2 is much higher. This suggests that we can infer Prin1 as an index of the overall stock returns, while Prin2 is a measure of the Soft Drinks sector versus the other sectors.

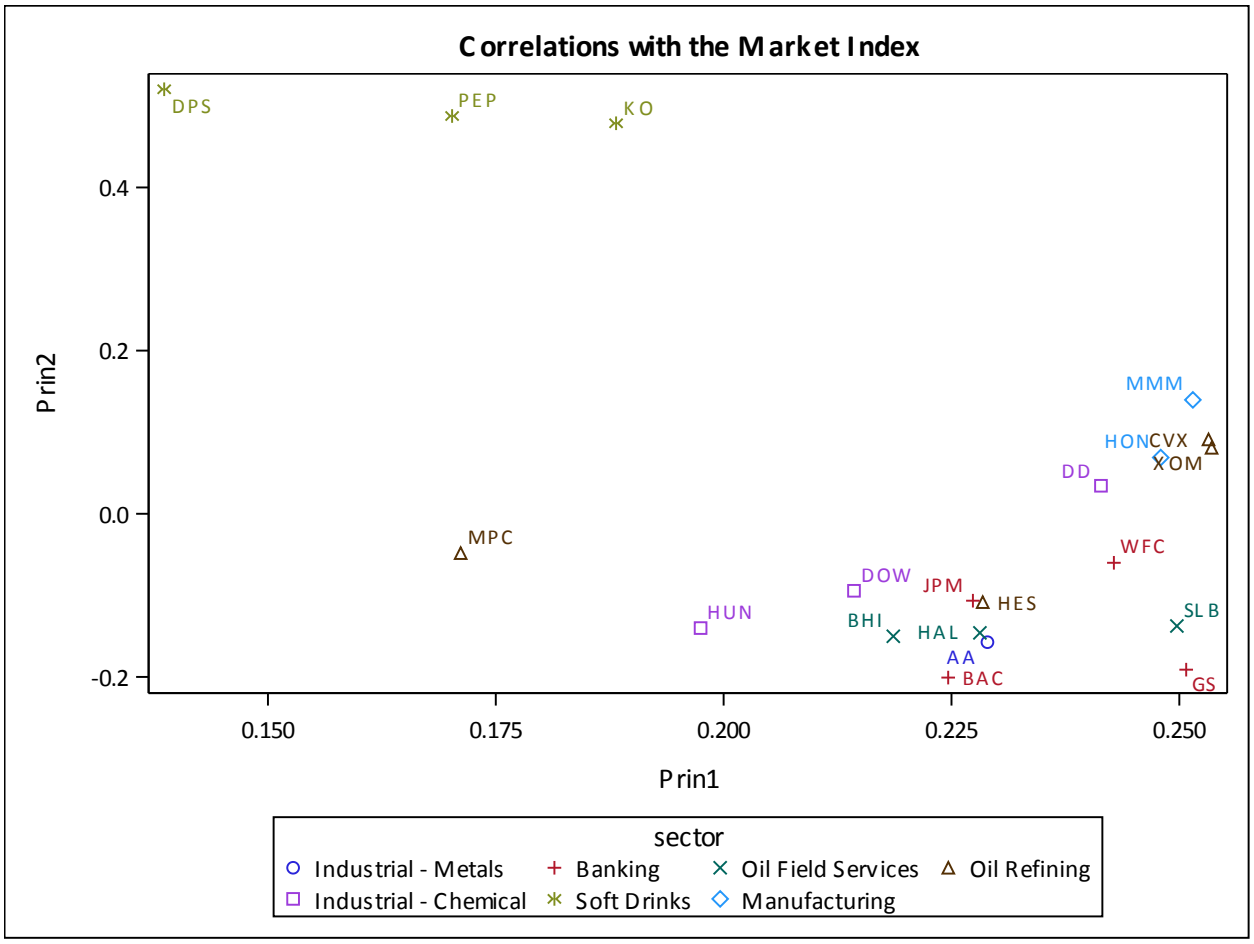

Figure 4: Correlations with the Market Index for the First Two Principal Components

In the above plot, we can see that the pattern we detected in **Table 3** concerning the Soft Drinks stocks is also apparent in **Figure 4**. Although the Soft Drinks stocks are not as close to each other as some other stocks are to stocks in a shared sector, the three Soft Drinks stocks clearly form a cluster as they are the only stocks in that region of the plot. The two Manufacturing stocks form a tight cluster, while the other sectors all form loose clusters. For instance, all the Banking stocks are in the same region of the plot, but they are not as close to each other as the Manufacturing stocks. We can interpret them as a loose cluster since they are proximate to each other rather than dispersed throughout the graph. The only sector that does not show any clustering among all stocks is Oil Refining. Stocks CVX and XOM are very near each other, but HES and MPC spread into different regions of the plot. Looking at **Figure 4** and similar plots can help us detect relationships in the data that rise to the surface when we use PCA.

Partitioning the Data for Cross-Validation

In order to assess the predictive accuracy of the two models we will introduce in the next couple sections, we split the stock portfolio data into a training set that contained approximately 70 percent of the data and a testing set that contained the rest of the data. **Table 4** shows the number of observations in each set.

train	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	164	32.67	164	32.67
1	338	67.33	502	100.00

Table 4: Observation Counts for the Cross-Validation Data Partition

Once we trained the models, we tested the predictive accuracy of the models on the remaining 30 percent of the sample population. In the discussion that follows, we will review the two models we created and assess their GOF and multicollinearity.

Creating a Model Using All Individual Stock Returns

In this section we are evaluating the GOF of a model we fit using all 20 of the individual stocks with the response variable, train_response, that we defined while partitioning the data for cross-validation. We will refer to this model as the All Stocks Model. Based on the output in **Table 5**, we can conclude that the individual stocks have a significant effect on the market index, train_response, since the p-value is very low (<.0001) and the F-statistic is high (140.04). The adjusted R-squared value (0.8919) shown in **Table 6** indicates that 89.19 percent of the variation in the response variable can be explained by the predictor variables in the model.

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	20	0.01790	0.00089510	140.04	<.0001
Error	317	0.00203	0.00000639		
Corrected Total	337	0.01993			

Table 5: ANOVA Table for the All Stocks Model

Root MSE	0.00253	R-Square	0.8983
Dependent Mean	0.00061635	Adj R-Sq	0.8919
Coeff Var	410.18453		

Table 6: Overall Model Fit Table for the All Stocks Model

Per **Table 7**, the fitted equation for the All Stocks Model is train_response = 0.00008640 + 0.01769*return_AA + 0.03198*return_BAC – 0.00111*return_BHI + 0.04907*return_CVX + 0.04674*return_DD + 0.03642*return_DOW + 0.03670*return_DPS + 0.04849*return_GS + 0.00948*return_HAL + 0.00359*return_HES + 0.12213*return_HON + 0.02712*return_HUN + 0.00902*return_JPM + 0.07903*return_KO + 0.09796*return_MMM + 0.01673*return_MPC + 0.02911*return_PEP + 0.03776*return_SLB + 0.07587*return_WFC + 0.05467*return_XOM. Except for the variable return_BHI, every unit increase in a predictor variable produces an increase in the response variable when all the other predictor variables are held constant. However, every unit increase in return_BHI causes a 0.00111 decrease in train_response. Although it is logical that an increase in the daily closing stock prices would cause the market index to increase, why the BHI stock causes it to decrease warrants further investigation.

Parameter Estimates						
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variance Inflation
Intercept	1	0.00008640	0.00014092	0.61	0.5403	0
return_AA	1	0.01769	0.01317	1.34	0.1802	2.11490
return_BAC	1	0.03198	0.01165	2.75	0.0064	3.10927
return_BHI	1	-0.00111	0.01323	-0.08	0.9333	2.62997
return_CVX	1	0.04907	0.02536	1.93	0.0539	3.07524
return_DD	1	0.04674	0.02037	2.29	0.0224	2.51406
return_DOW	1	0.03642	0.01162	3.14	0.0019	1.88893
return_DPS	1	0.03670	0.01679	2.19	0.0295	1.54768
return_GS	1	0.04849	0.01555	3.12	0.0020	3.10450
return_HAL	1	0.00948	0.01466	0.65	0.5184	3.08758
return_HES	1	0.00359	0.01092	0.33	0.7425	2.10199
return_HON	1	0.12213	0.01924	6.35	<.0001	2.73505
return_HUN	1	0.02712	0.00836	3.24	0.0013	1.79852
return_JPM	1	0.00902	0.01708	0.53	0.5979	3.36439
return_KO	1	0.07903	0.02226	3.55	0.0004	1.93633
return_MMM	1	0.09796	0.02646	3.70	0.0003	2.98277
return_MPC	1	0.01673	0.00809	2.07	0.0394	1.32999
return_PEP	1	0.02911	0.02231	1.30	0.1929	1.68825
return_SLB	1	0.03776	0.01709	2.21	0.0279	3.13690
return_WFC	1	0.07587	0.01848	4.10	<.0001	2.59492
return_XOM	1	0.05467	0.02697	2.03	0.0435	2.98393

Table 7: Parameter Estimates Table for the All Stocks Model

When assessing a model's fit, it is important to examine the model for signs of multicollinearity. To do this, we calculated the VIFs for each predictor variable present in the All Stocks Model. Typically a VIF greater than 10 is the threshold for determining if a model has a serious problem with multicollinearity.[2] In **Table 7** above, all of the predictor variables have a VIF less than 4. This suggests that multicollinearity is not a major issue for this model. However, a VIF greater than 1 indicates that there is correlation among the predictor variables. There is not a clear pattern to explain why some stocks have higher VIFs than other stocks. For instance, if all the Soft Drinks sector stocks had VIFs of 3 while all the other stocks had VIFs closer to 1, we would want to consider removing at least one of the Soft Drinks stocks in order to reduce the amount of multicollinearity in the model. However, this is not the case, so removing some of the variables with the highest VIFs might remedy the moderate multicollinearity issues in this model.

Although the results in the above tables indicate a strong linear correlation between the predictor variables and response variable in the All Stocks Model, analysis of the diagnostic

[2] Montgomery, D.C., Peck, E.A., and Vining, G.G. (2012). *Introduction to Linear Regression Analysis*. (5th Edition). New York, NY: Wiley

plots is also necessary to understand the model's overall GOF. Ideally the residuals should support the OLS assumptions that the errors are normally distributed and homoscedastic.

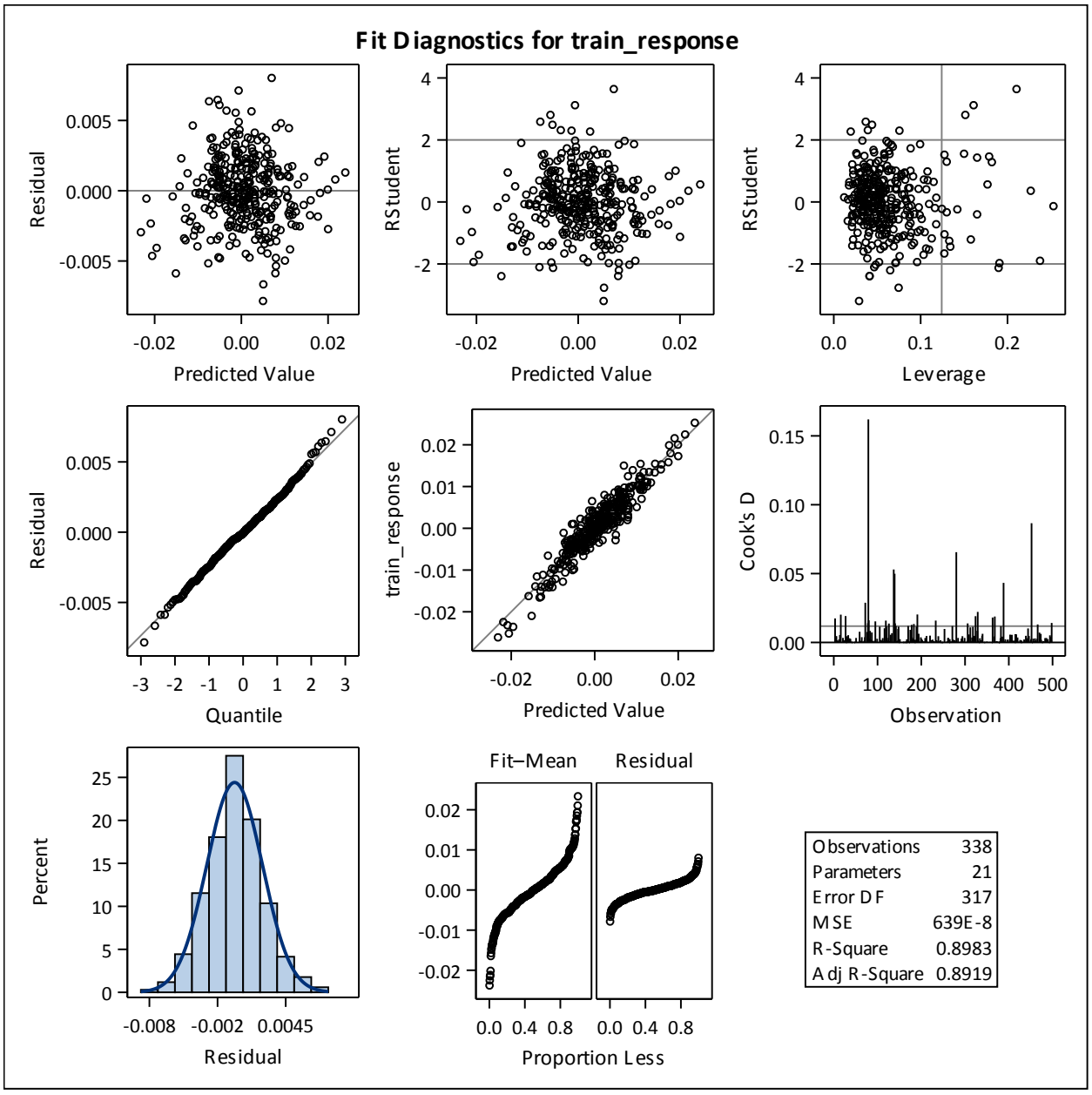

Figure 5: Fit Diagnostic Plots for the All Stocks Model

In **Figure 5** above, the Q-Q plot indicates a mostly normal distribution. Except for a few observations on each end that drift slightly from the 45-degree line, the observations form a nearly straight line. The plotted line does not contain heavy tails or exhibit any curvature, which would indicate a skewed distribution. The residuals histogram confirms the conclusion that the model generally supports the normality assumption.

The residual-leverage plot in **Figure 5** indicates that there are several observations that may be outliers (the points that fall outside the horizontal lines), as well as high-leverage points (the points to the right of the vertical line). Examining these observations closer and removing some

of them may help improve the GOF, but the plot reveals that the vast majority of the observations do not exhibit strong leverage or influence in the model. The Cook's D plot suggests the same conclusions since it shows only a few high peaks, but the majority of observations have low peaks that fall below the horizontal line and are thus not indicative of being influential in the model.

The plot of the response variable by the predicted values in the center of **Figure 5** indicates that the model is a good, but not perfect, fit. The points should ideally all fall on the line, which would support that the model's predicted output and the actual observations of the response variable have a linear relationship. The majority of observations, however, are very close to the line, and even the handful of observations that are not part of the main tight cluster still fall close to the line. In the residual-fit plot directly below this plot, the fit side is much taller than the residual side, indicating that the model accounts for most of the variation in the data.

Both the residuals and studentized residuals by the predicted values plots in **Figure 5** indicate that the OLS assumption of homoscedasticity is mostly true since the residuals in the plot are spread randomly. However, for the residuals to have perfect constant variance and have a mean of zero, they would need to form a band around mean zero rather than a cluster. A band of residuals would indicate that the residuals exhibit independence and identical distribution.

In the next section we will fit a model using just the first eight principal components we discussed earlier and compare its fit to the All Stocks Model.

Creating a Model Using the First Eight Principal Components

After fitting a regression model using all of the individual stocks, we decided to fit another regression model using just the first eight principal components. We will refer to this model as the Principal Components Model. The regression output for this model has some similarities to the previous model. **Table 8** reveals that using the first eight principal components produces a very low p-value, just like the All Stocks Model. However, the F-statistic in **Table 8** for the Principal Components Model is much higher than the F-statistic shown in **Table 5** for the All Stocks Model, indicating that the predictor variables in the Principal Components Model have a more significant effect on the response variable than the predictor variables in the All Stocks Model. In **Table 9**, we can see that the adjusted R-squared value is 0.8886, which indicates that the Principal Components Model accounts for 88.86 percent of the variation in the response variable. Although this value is slightly less than the 89.19 percent of variation that the All Stocks Model explains, the values are very close and cannot be interpreted as a conclusive indicator of which model fits better.

Analysis of Variance					
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	8	0.01776	0.00222	337.13	<.0001
Error	329	0.00217	0.00000659		
Corrected Total	337	0.01993			

Table 8: ANOVA Table for the Principal Components Model

Root MSE	0.00257	**R-Square**	0.8913
Dependent Mean	0.00061635	**Adj R-Sq**	0.8886
Coeff Var	416.36522		

Table 9: Overall Fit Table for the Principal Components Model

Per **Table 10** below, the fitted equation for the Principal Components Model is train_response = 0.00075978 + 0.00231*Prin1 + 0.00032245*Prin2 + 0.00070635*Prin3 + 0.00030481*Prin4 – 0.00017356*Prin5 + 0.00000315*Prin6 – 0.00010331*Prin7 – 0.00040760*Prin8. Some of the parameter estimates, such as the values for Prin5, Prin7, and Prin8, reveal an inverse relationship with the response variable. Examining each principal component more closely may reveal why certain parameter estimates are positive while others are negative.

Parameter Estimates						
Variable	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**	**Variance Inflation**
Intercept	1	0.00075978	0.00014045	5.41	<.0001	0
Prin1	1	0.00231	0.00004519	51.05	<.0001	1.00527
Prin2	1	0.00032245	0.00011425	2.82	0.0051	1.00868
Prin3	1	0.00070635	0.00012322	5.73	<.0001	1.00861
Prin4	1	0.00030481	0.00014536	2.10	0.0368	1.00636
Prin5	1	-0.00017356	0.00015516	-1.12	0.2641	1.00297
Prin6	1	0.00000315	0.00017108	0.02	0.9853	1.00766
Prin7	1	-0.00010331	0.00018604	-0.56	0.5791	1.02315
Prin8	1	-0.00040760	0.00020293	-2.01	0.0454	1.02271

Table 10: Parameter Estimates Table for the Principal Components Model

In the table above we also computed the VIFs for each principal component to determine if multicollinearity is present in the model. All of the VIF values are close to 1, which indicates that multicollinearity is not an issue in the Principal Components Model. In comparison to the VIF values for the All Stocks Model in **Table 7**, the VIF values above in **Table 10** are lower and more consistent, indicating that the Principal Components Model is a more stable and better model than the All Stocks Model. However, before drawing definitive conclusions, we must examine the fit diagnostic plots to see how the two models compare.

In the following discussion we will present side-by-side comparisons of various plots from the All Stocks Model and the Principal Components Model. As the plots will reveal, reducing the dimensions of the data using PCA did not result in drastically different fit diagnostic plots from those produced by the All Stocks Model.

Figures 6 and 7 below show the Q-Q plots for each model. Both of the Q-Q plots suggest a normal distribution of the residuals since the observations fall mostly in a straight line without any curvature or heavy tails. Each plot shows that a few points slightly drift from the 45-degree line at the ends, but neither plot suggests that one model is clearly superior to the other model. Examining the data to identify which observations are causing some of the points to stray slightly from the line might help improve the fit of both models.

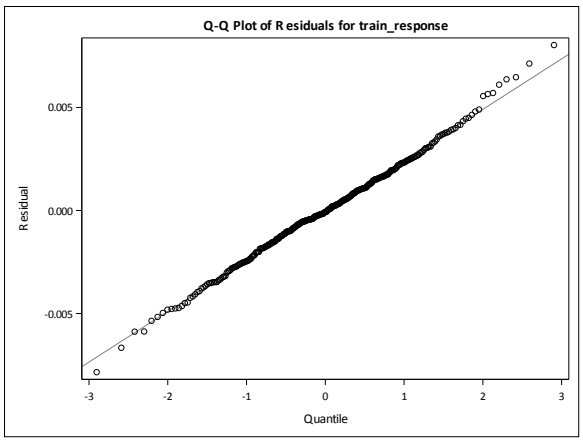

Figure 6: Q-Q Plot for All Stocks Model

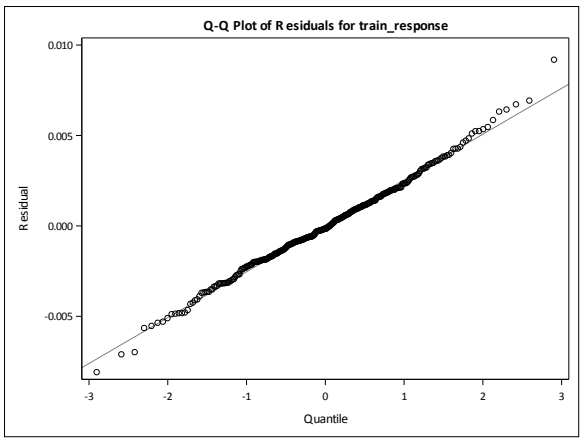

Figure 7: Q-Q Plot for PC Model

Just as the Q-Q plots above suggest a normal distribution, the distribution of residuals histograms in **Figures 8 and 9** below also suggest a mostly normal distribution. Fitting a model using PCA has changed the distribution of the residuals, but not enough to skew the data from a mostly normal distribution.

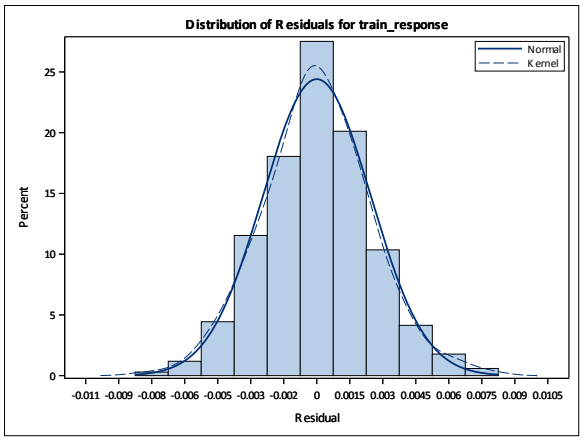

Figure 8: Histogram for All Stocks Model

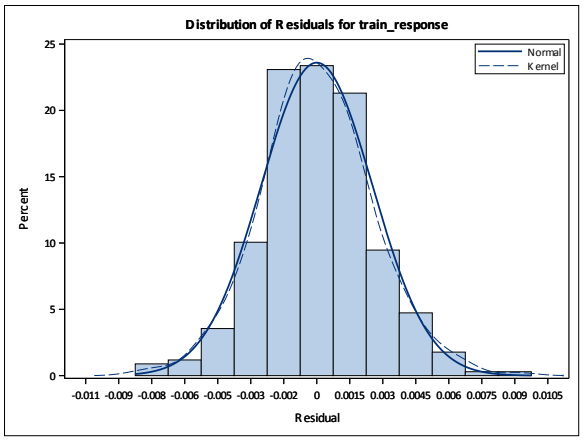

Figure 9: Histogram for PC Model

In **Figures 10 and 11** below, the plots of the studentized residuals by the predicted values for both the All Stocks Model and the Principal Components Model are nearly identical. Neither plot strongly indicates which model is better, but they show that performing PCA did indeed result in model that captured the majority of the variance in the data since the residuals are similarly distributed in the plots for both models.

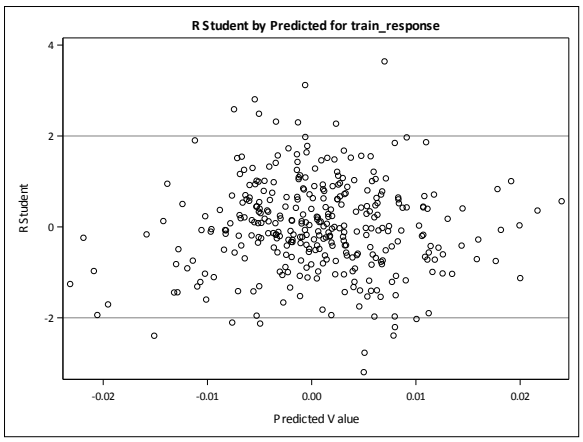

Figure 10: Residuals Plot for All Stocks Model

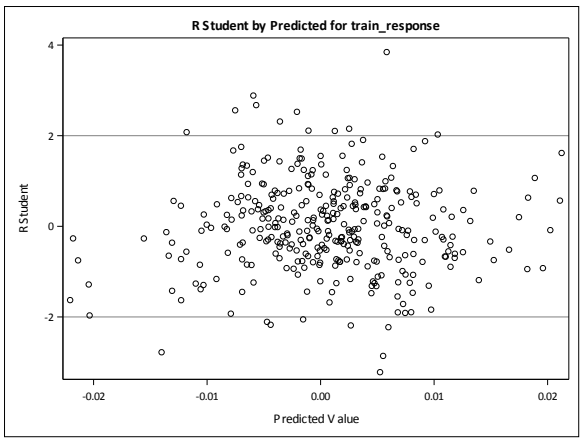

Figure 11: Residuals Plot for PC Model

The observed by predicted plots in **Figures 12 and 13** suggest that each model can predict the response variable with about the same accuracy as the other model, thus indicating that both models are equally adequate.

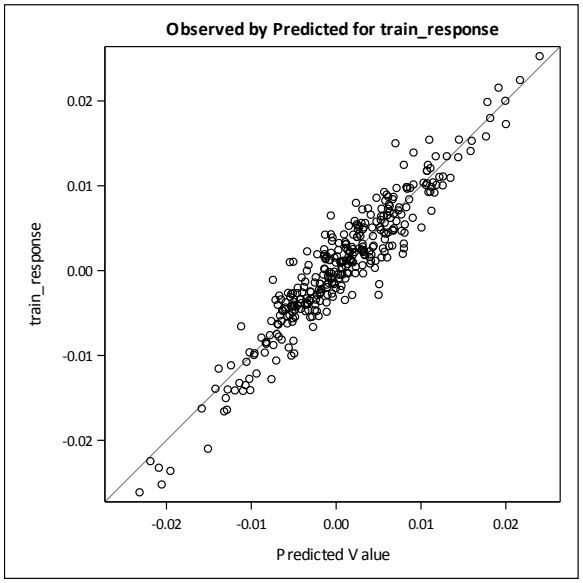

Figure 12: Accuracy Plot for All Stocks Model

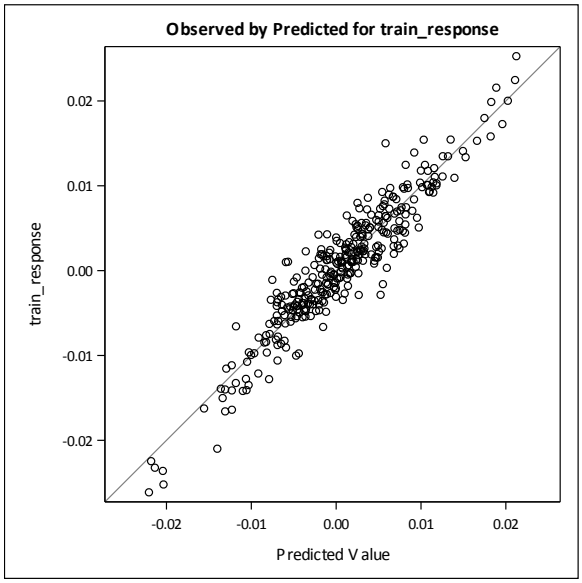

Figure 13: Accuracy Plot for PC Model

In **Figures 14 and 15**, both models show a higher fit side than residual side, indicating that both models explain most of the variance in the data. Again, neither plot strongly suggests one model is better than the other model.

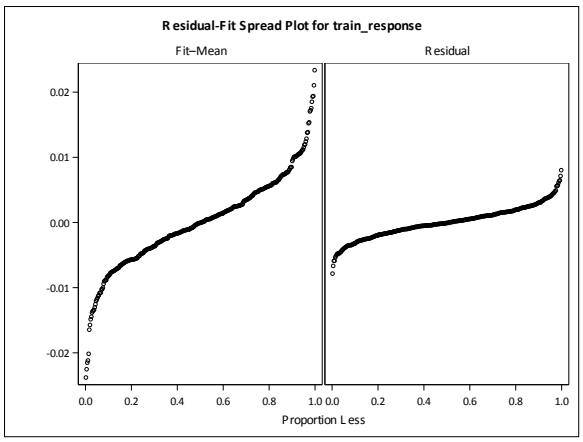

Figure 14: Residual-Fit Spread for All Stocks Model

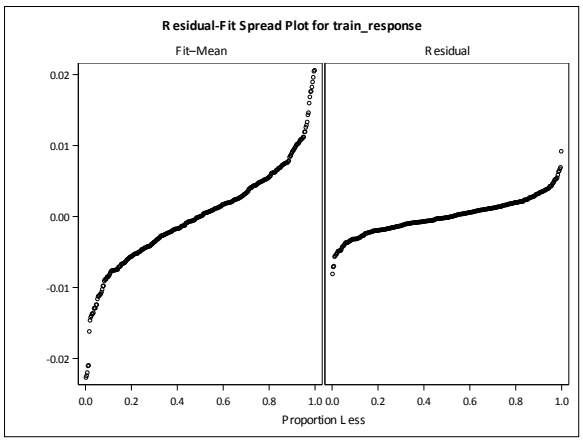

Figure 15: Residual-Fit Spread for PC Model

In the outlier detection plots in **Figures 16, 17, 18, and 19**, we are able to more clearly see how the two models differ than we could from looking at the plots discussed above. **Figure 17** suggests that the Principal Components Model contains more outliers than the All Stocks Model, and the leverage points in **Figure 17** appear to be more clustered than the leverage points in **Figure 16**. Reducing the dimensions of the data using PCA has altered the distribution of the leverage and influence points, but we would need to examine the data closer to understand if this affects the overall performance of the model.

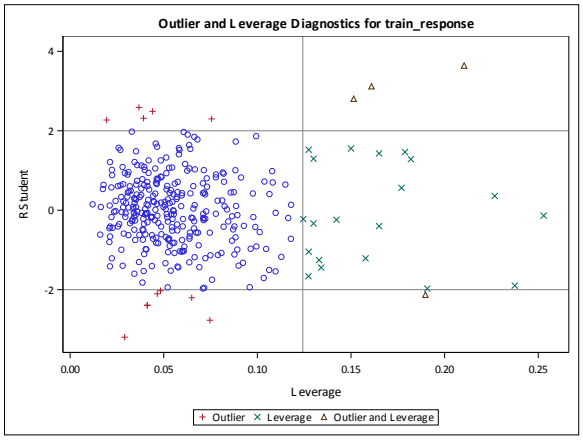

Figure 16: Leverage Plot for All Stocks Model

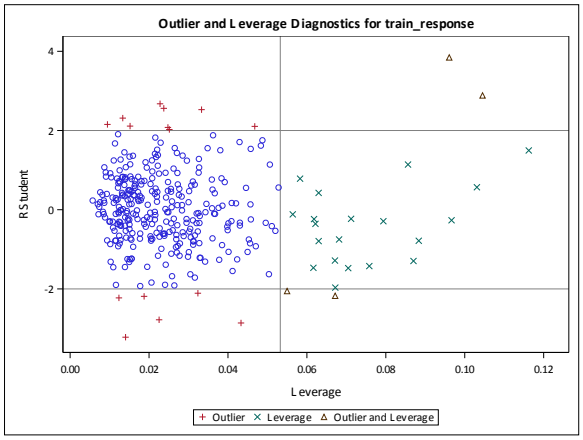

Figure 17: Leverage Plot for PC Model

In the Cook's D plots below, we see that **Figure 19** has fewer high peaks than **Figure 18**. This suggests that there are fewer potential influential observations in the Principal Components Model than the All Stocks Model, which suggests that the Principal Components Model fits the data better.

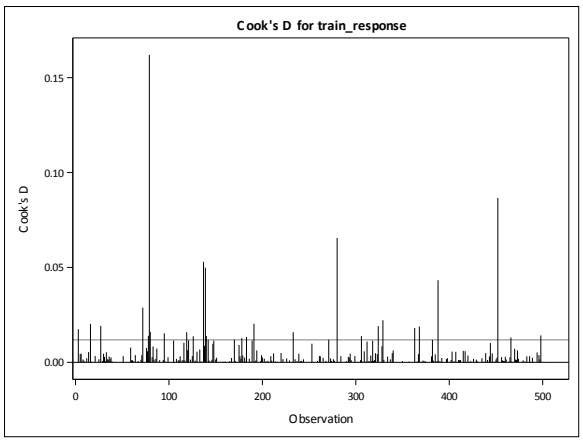

Figure 18: Cook's D Plot for All Stocks Model

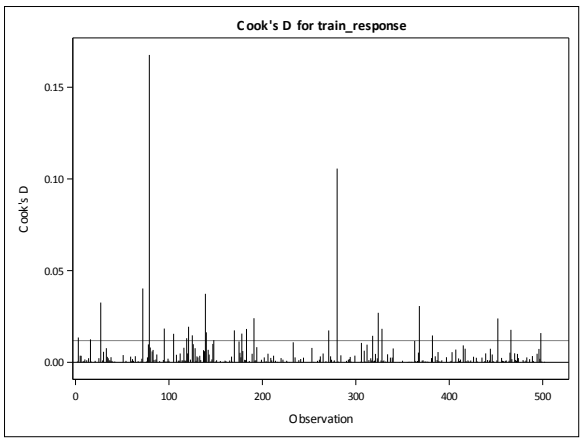

Figure 19: Cook's D Plot for PC Model

MSE and MAE of the All Stocks and Principal Components Models

In addition to looking at the summary tables and diagnostic plots of the two models, we decided to compute and compare the MSE and MAE values for both models to try to determine which model fits better.

In **Tables 11 and 12** below, we computed the MSE and MAE for both the testing and the training data sets for each model. The first row in each table shows the statistics for the testing set, while the second row shows the statistics for the training set. For both models the MSE and MAE are higher for the testing set than the training set, which indicates that the models performed worse out-of-sample than in-sample. Ideally the MSE and MAE for the testing set would be similar in value to the MSE and MAE for the training set.

Obs	train	_TYPE_	_FREQ_	MSE	MAE
1	0	1	164	.000009306	.002144904
2	1	1	338	.000005994	.001902032

Table 11: MSE and MAE Computations for the All Stocks Model

Obs	train	_TYPE_	_FREQ_	MSE	MAE
1	0	1	164	.000009677	.002179249
2	1	1	338	.000006410	.001975239

Table 12: MSE and MAE Computations for the Principal Components Model

When we compare the MSE and MAE values for both models, we see that the All Stocks Model has slightly lower error scores both in-sample and out-of-sample than the Principal Components Model. This suggests that the All Stocks Model fits slightly better than the Principal Components Model since smaller errors indicate better accuracy. However, when determining which model fits better overall, it is important to consider all the statistical and graphical output. Most of the diagnostic plots for the two models were nearly identical, and none of the plots that had noticeable differences showed a clear advantage of one model over the other. Examining the statistical output reveals that the adjusted R-squared, MSE, and MAE values are all very close between the two models, but that the statistics for the All Stocks Model suggest a slightly better fit. However, when we consider the F-statistics and VIFs of the two models, the Principal Components Model is clearly a stronger and more stable model than the All Stocks Model.

Because the graphical and statistical output between the two models is so similar, we decided that choosing the model with a higher F-statistic and less multicollinearity made the most sense. Both models are strong, but the Principal Components Model is better because it accomplishes what the All Stocks Model does but with fewer data and less multicollinearity.

Conclusions

In this report we explored the use of PCA as a method of dimension reduction and a remedy of multicollinearity in OLS regression. We used the log-returns of 20 individual stocks from the stock portfolio data set to explain the variation in the log-returns of the Vanguard market index fund, and we fit two regression models to analyze how well a model using principal components performed in comparison to a model using all of the individual stocks. By examining the diagnostic plots and statistical tables produced by the models, we were able to assess the GOF of each model and determine if multicollinearity was an issue in either model. Although both models produced nearly identical diagnostic plots and had very similar statistics, such as adjusted R-squared, MSE, and MAE, we learned that the Principal Components Model has much better VIFs, indicating that multicollinearity is not as much of an issue for this model as it is for the All Stocks Model. The lower VIFs and much higher F-statistic for the Principal Components Model suggests that this model is more reliable and fits better than the All Stocks Model. Deciding which model to deploy, however, should also take into consideration the intended application of the model, as well as subject matter knowledge of the problem the model is trying to solve. We did not incorporate these considerations into our decision for this report but would do so before deploying the model in a real-world context.

Code

```
*************************************************************************;
* Preliminary Steps;
*************************************************************************;

* Access library where stock portfolio data set is stored;
libname mydata '/scs/crb519/PREDICT_410/SAS_Data/' access=readonly;
proc datasets library=mydata; run; quit;

*************************************************************************;
* Step 1;
*************************************************************************;

* Set stock portfolio data set to short name;
data temp;
set mydata.stock_portfolio_data;
run;

proc contents data=temp; run;
proc print data=temp (obs=10); run;

* AA   BAC   BHI   CVX   DD    DOW   DPS   GS    HAL   HES   HON   HUN   JPM
       KO    MMM   MPC   PEP   SLB   WFC   XOM   VV;

proc sort data=temp; by date; run; quit;
data temp;
set temp;
* Compute the log-returns - log of the ratio of today's price
to yesterday's price;
* Note that the data needs to be sorted in the correct
direction in order for us to compute the correct return;
return_AA = log(AA/lag1(AA));
return_BAC = log(BAC/lag1(BAC));
* Continue to compute the log-returns for all of the stocks;
return_BHI = log(BHI/lag1(BHI));
return_CVX = log(CVX/lag1(CVX));
return_DD = log(DD/lag1(DD));
return_DOW = log(DOW/lag1(DOW));
return_DPS = log(DPS/lag1(DPS));
return_GS = log(GS/lag1(GS));
return_HAL = log(HAL/lag1(HAL));
return_HES = log(HES/lag1(HES));
return_HON = log(HON/lag1(HON));
return_HUN = log(HUN/lag1(HUN));
return_JPM = log(JPM/lag1(JPM));
return_KO = log(KO/lag1(KO));
return_MMM = log(MMM/lag1(MMM));
return_MPC = log(MPC/lag1(MPC));
return_PEP = log(PEP/lag1(PEP));
```

```
return_SLB = log(SLB/lag1(SLB));
return_WFC = log(WFC/lag1(WFC));
return_XOM = log(XOM/lag1(XOM));

* Name the log-return for VV as the response variable;
response_VV = log(VV/lag1(VV));
run;
proc print data=temp(obs=10); run; quit;
*The log-returns have only 501 observations instead of 502 because they were created by
computing the ratio from the next day to the previous day. Day 1 doesn't have a previous day
and therefore contains missing return values/a blank row.;

**********************************************************************;
* Step 2;
**********************************************************************;

* We can use ODS TRACE to print out all of the data sets available to ODS for a particular SAS
procedure.;
* We can also look these data sets up in the SAS User's Guide in the chapter for the selected
procedure.;
*ods trace on;
ods output PearsonCorr=portfolio_correlations;
proc corr data=temp;
*var return: with response_VV;
var return_:;
with response_VV;
run; quit;
*ods trace off;
proc print data=portfolio_correlations; run; quit;
* This prints the p-values of the log-returns in addition to the correlation between the response
and log-returns;

**********************************************************************;
* Step 3;
**********************************************************************;

data wide_correlations;
set portfolio_correlations (keep=return_:); * This gets rid of the p-values;
run;

* Note that wide_correlations is a 'wide' data set and we need a 'long' data set;
* We can use PROC TRANSPOSE to convert data from one format to the other;
proc transpose data=wide_correlations out=long_correlations;
run; quit;

data long_correlations;
set long_correlations;
tkr = substr(_NAME_,8,3); * Little SAS pg 78-19. What does 8 mean? This does not seem right:
The correlation column is 7 characters, so the tkr column needs to start at the 8th character
and it needs to be 3 characters long to fit the names of the stocks;
drop _NAME_;
```

```
rename COL1=correlation;
run;

proc print data=long_correlations; run; quit;

**********************************************************************;
* Step 4;
**********************************************************************;

* Merge on sector id and make a colored bar plot;
data sector;
input tkr $ 1-3 sector $ 4-35; * numbers define how many characters per column;
datalines;
AA Industrial - Metals
BAC Banking
BHI Oil Field Services
CVX Oil Refining
DD Industrial - Chemical
DOW Industrial - Chemical
DPS Soft Drinks
GS Banking
HAL Oil Field Services
HES Oil Refining
HON Manufacturing
HUN Industrial - Chemical
JPM Banking
KO Soft Drinks
MMM Manufacturing
MPC Oil Refining
PEP Soft Drinks
SLB Oil Field Services
WFC Banking
XOM Oil Refining
VV Market Index
;
run;
proc print data=sector; run; quit;
proc sort data=sector; by tkr; run;
proc sort data=long_correlations; by tkr; run;
data long_correlations;
merge long_correlations (in=a) sector (in=b);
by tkr;
if (a=1) and (b=1);
run;

proc print data=long_correlations; run; quit;

* Make Grouped Bar Plot;
* p. 48 Statistical Graphics Procedures By Example;
ods graphics on;
title 'Correlations with the Market Index';
```

```
proc sgplot data=long_correlations;
format correlation 3.2; * Little SAS pg. 110-11: The 3 is the width ('0.77' is width 3), and .2 is
how many decimal places to keep for the correlation values;
vbar tkr / response=correlation group=sector groupdisplay=cluster datalabel;
run; quit;
ods graphics off;

**********************************************************************;
* Step 5;
**********************************************************************;

data return_data;
set temp (keep= return_:); * This gets rid of the date and name columns (AA, BAC, BHI, etc.).
VV is not part of the table since it's a response variable (response_VV v. return_stock);
* What happens when I put this keep statement in the set statement?;
* Look it up in The Little SAS Book; * pg 198-199: helps you avoid reading lots of variables you
don't need to read;
run;

proc print data=return_data(obs=10); run;

ods graphics on;
proc princomp data=return_data out=pca_output outstat=eigenvectors
plots=scree(unpackpanel); *Do plots=all to see more plots and decide which are important;
run; quit;
ods graphics off;
* Notice that PROC PRINCOMP produces a lot of output;
* How many principal components should we keep?; * 8 because this is where the elbow is in
the scree plot;
* Do the principal components have any interpretability?;
* Can we display that interpretability using graphics?;

proc print data=pca_output(obs=10); run;

* proc print data=eigenvectors without specifying _TYPE_='SCORE' prints a huge table
with the mean, std, n, correlation matrix, eigenvalues, and the PCs. We need just the PCs.;

proc print data=eigenvectors(where=(_TYPE_='SCORE')); run;
* Display the two plots and the Eigenvalue table from the output;

data pca2;
set eigenvectors(where=(_NAME_ in ('Prin1','Prin2')));
drop _TYPE_ ;
run;

proc print data=pca2; run;

proc transpose data=pca2 out=long_pca; run; quit;
proc print data=long_pca; run;

data long_pca;
```

```
set long_pca;
format tkr $3.;
tkr = substr(_NAME_,8,3);
drop _NAME_;
run;

proc print data=long_pca; run;

* Plot the first two eigenvectors;
* Note that SAS has been calling them Prin* but giving them type SCORE;
data long_pca;
merge long_pca (in=a) sector (in=b);
by tkr;
if (a=1) and (b=1);
run;
proc print data=long_pca; run; quit;

ods graphics on;
proc sgplot data=long_pca;
scatter x=Prin1 y=Prin2 / datalabel=tkr group=sector;
run; quit;
ods graphics off;
* Do we see anything interesting here? Why would we make such a plot?;

**********************************************************************;
* Step 6;
**********************************************************************;

* Create a training data set and a testing data set from the PCA output;
* Note that we will use a SAS shortcut to keep both of these 'datasets' in one data set that we
will call cv_data (cross-validation data). ;

data cv_data;
merge pca_output temp(keep=response_VV); * pca_output doesn't contain response_VV, but
temp does--appending response_VV to pca_output;
* No BY statement needed here. We are going to append a column in its current order;
* generate a uniform(0,1) random variable with seed set to 123;
u = uniform(123);
if (u < 0.70) then train = 1;
else train = 0;
if (train=1) then train_response=response_VV;
else train_response=.;
run;

proc print data=cv_data(obs=10); run;

* View frequency test/train data;
proc freq data=cv_data;
tables train;
title 'Observation Counts for the Cross-Validation Data Partition';
run; quit;
```

```
*************************************************************************;
* Step 7;
*************************************************************************;

* Fit a regression model using all of the individual stocks with train_response as the response
variable.;
ods graphics on;
title;
proc reg data=cv_data outest=Model_All_out;
Model_All: model train_response = return_: / selection=rsquare start=20 stop=20 adjrsq aic bic
mse cp vif;
output out = Model_All_fit pred=yhat;
run;
ods graphics off;

proc print data=Model_All_out; run;

data Model_All_fit;
set Model_All_fit;
residual = response_VV - yhat;
abs_res = abs(residual);
sq_res = residual*residual;
run;

proc print data=Model_All_fit (obs=10);
run;

title "Model_All's MSE and MAE";
proc means data=Model_All_fit nway noprint;
class train;
var sq_res abs_res;
output out = msemae_Model_All_fit
mean(sq_res abs_res) = MSE MAE;
proc print data = msemae_Model_All_fit;
run; quit;

*************************************************************************;
* Step 8;
*************************************************************************;

* Fit a regression model using the first 8 principal components with train_response as the
response variable.;
ods graphics on;
title;
proc reg data=cv_data outest=Model_PC8_out;
Model_PC8: model train_response = Prin1-Prin8 / selection=rsquare start=8 stop=8 adjrsq aic
bic mse cp vif;
output out = Model_PC8_fit pred=yhat;
run;
ods graphics off;
```

```
proc print data=Model_PC8_out; run;

data Model_PC8_fit;
set Model_PC8_fit;
residual = response_VV - yhat;
abs_res = abs(residual);
sq_res = residual*residual;
run;

proc print data=Model_PC8_fit (obs=10);
run;

title "Model_PC8's MSE and MAE";
proc means data=Model_PC8_fit nway noprint;
class train;
var sq_res abs_res;
output out = msemae_Model_PC8_fit
mean(sq_res abs_res) = MSE MAE;
proc print data = msemae_Model_PC8_fit;
run; quit;

*************************************************************************;
* Create Unpacked Models;
*************************************************************************;

ods graphics on;
title;
proc reg data=cv_data outest=Model_All_out plots(unpack);
Model_All: model train_response = return_: / selection=rsquare start=20 stop=20 adjrsq aic bic
mse cp vif;
output out = Model_All_fit pred=yhat;
run;
ods graphics off;

ods graphics on;
title;
proc reg data=cv_data outest=Model_PC8_out plots(unpack);
Model_PC8: model train_response = Prin1-Prin8 / selection=rsquare start=8 stop=8 adjrsq aic
bic mse cp vif;
output out = Model_PC8_fit pred=yhat;
run;
ods graphics off;

*************************************************************************;
* End;
*************************************************************************;
```