Noé Flores
Principal Components in Predictive Modeling

Introduction:

The objective of this analysis is to define and utilize principal component analysis (PCA), to examine and run a regression model on our dataset. The data is made up of daily closing prices for 20 stocks and a large-cap index ETF from Vanguard. For reference, a large-cap index is composed of stocks with large market capitalization values. The utilization of PCA should allow us to incorporate dimension reduction and minimize multicollinearity between our variables.

Results and Analysis:

Our examination begins by making some general observations within our dataset. The dataset has closing prices for the 20 different stocks and the large-cap ETF, with the date range for these closing prices extending from January 3, 2012, through December 31, 2013. What we will actually be examining here is the daily returns for the stocks, specifically, we will explore the log-returns of the individual stocks to help us explain the variation in the log-returns of the market index. The table below provides a brief example of the data and the subsequently transformed returns for four of the stocks and our index.

Figure 1: Exterior Quality Variable.

Closing Prices					
Date	**AA**	**BAC**	**BHI**	**CVX**	**VV**
3-Jan-12	9.23	5.8	51.02	110.37	58.18
4-Jan-12	9.45	5.81	51.53	110.18	58.25
5-Jan-12	9.36	6.31	50.82	109.1	58.44
Returns					
Date	**return_AA**	**return_BAC**	**return_BHI**	**return_CVX**	**response_VV**
3-Jan-12	.	.	.	.	.
4-Jan-12	0.023556	0.001723	0.009946	-0.001722966	0.001202439
5-Jan-12	-0.009569	0.082555	-0.013874	-0.009850499	0.003256494

We continue by examining the correlations between the daily returns of each stock and the stock index response variable, VV. The statistical analysis tool we are incorporating in this investigation is SAS. Typically, correlations in SAS are output in a comprehensive format, making it difficult to view and quickly assess. We will transpose the returned correlation analysis into an extended format to better assist us in making initial visual quantifications.

Figure 2: Correlations.

Obs	correlation	tkr	Sector
1	0.63241	AA	Industrial - Metals
2	0.65019	BAC	Banking
3	0.5775	BHI	Oil Field Services
4	0.7209	CVX	Oil Refining
5	0.68952	DD	Industrial - Chemical
6	0.62645	DOW	Industrial - Chemical
7	0.4435	DPS	Soft Drinks
8	0.71216	GS	Banking
9	0.5975	HAL	Oil Field Services
10	0.6108	HES	Oil Refining
11	0.76838	HON	Manufacturing
12	0.58194	HUN	Industrial - Chemical
13	0.65785	JPM	Banking
14	0.5998	KO	Soft Drinks
15	0.76085	MMM	Manufacturing
16	0.47312	MPC	Oil Refining
17	0.50753	PEP	Soft Drinks
18	0.69285	SLB	Oil Field Services
19	0.73357	WFC	Banking
20	0.72111	XOM	Oil Refining

In order to get a better visual of the possible similarities in the correlations between the stocks and the sector they operate in we provide a graph below. In this graph, we see the correlations grouped by individual sectors, as well as by individual stocks. Our sample of stocks is relatively small, so we can see that there actually is a bit of difference in stocks correlations within their specific sectors.

Figure 3: Correlations for each stock.

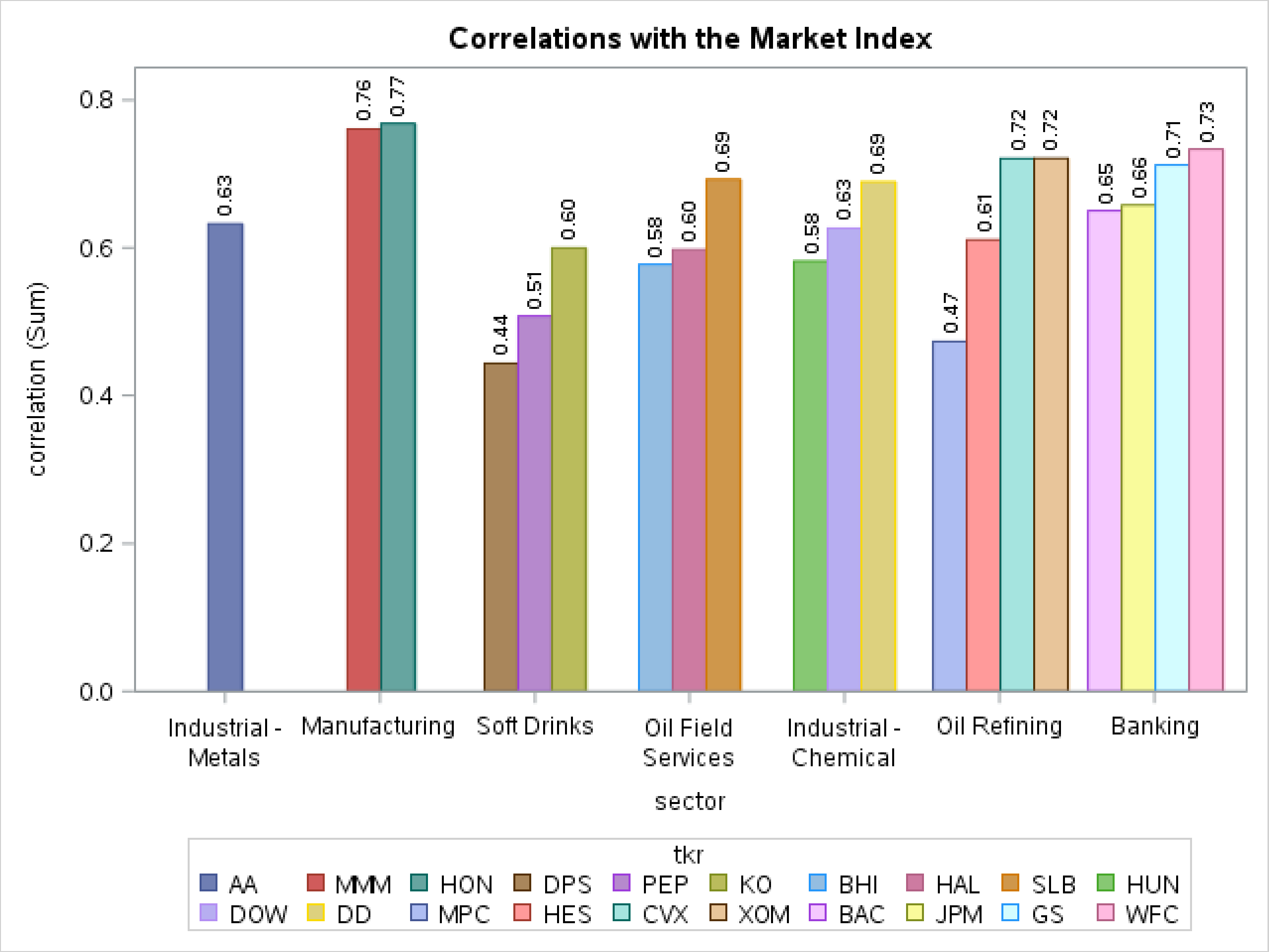
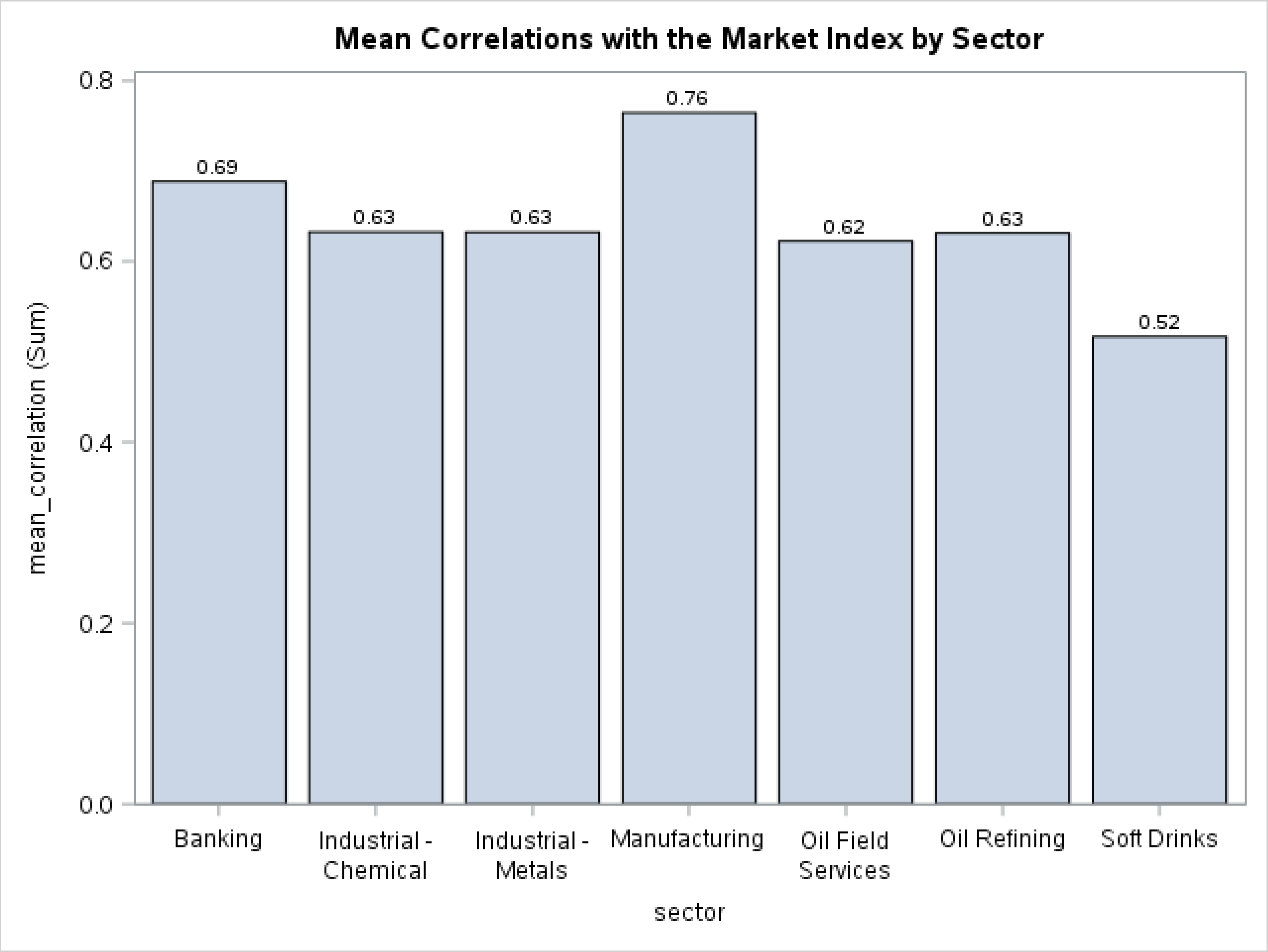


Figure 4: Correlations for each stock grouped by sector.



This graphic provides some interesting observations. It appears that the manufacturing sector has the highest correlations to our response variable, VV, followed by the banking industry.

The main objective up to this point has been to get a feel for the type of data we are working on in this analysis. What we will now set out to do is define one of the primary objectives of our analysis, which is to determine which principal components to use for our future regression modeling. Principal Component Analysis (PCA) is used to find the underlying variables that can best differentiate our data, it combats multicollinearity by reducing multidimensional data, and it is used as a means to convert possibly correlated variables into a set of variables that are uncorrelated. The table below provides a set of components and their eigenvalues. It also provides us with the proportion of variation each one accounts for, as well as the cumulative total of components added.

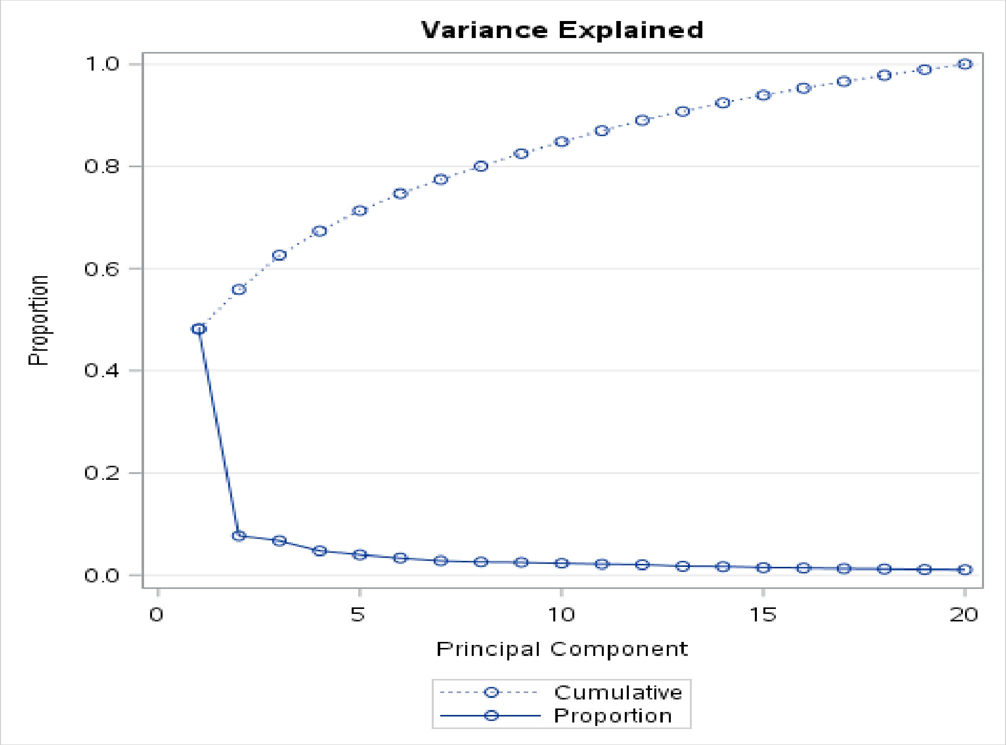
Figure 5: Exterior Quality Variable.

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	0.1290986	0.0398	0.7133
6	0.66640793	0.1079874	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	0.825
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	0.2762875	0.01692712	0.0138	0.9532
17	0.25936037	0.01730228	0.013	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

What we want to achieve is the ability to retain just enough components to explain "some" specified large percentage of the total variation of the original variables. One such rule we can implement is to use enough components to get us between 70% and 90% of the total variability in the dataset. With that in mind, keeping the first 8 components, totaling 80% of the variation in the dataset allows us to split the 70% - 90% recommendation right down the middle. This also allows us to keep the model relatively simple.

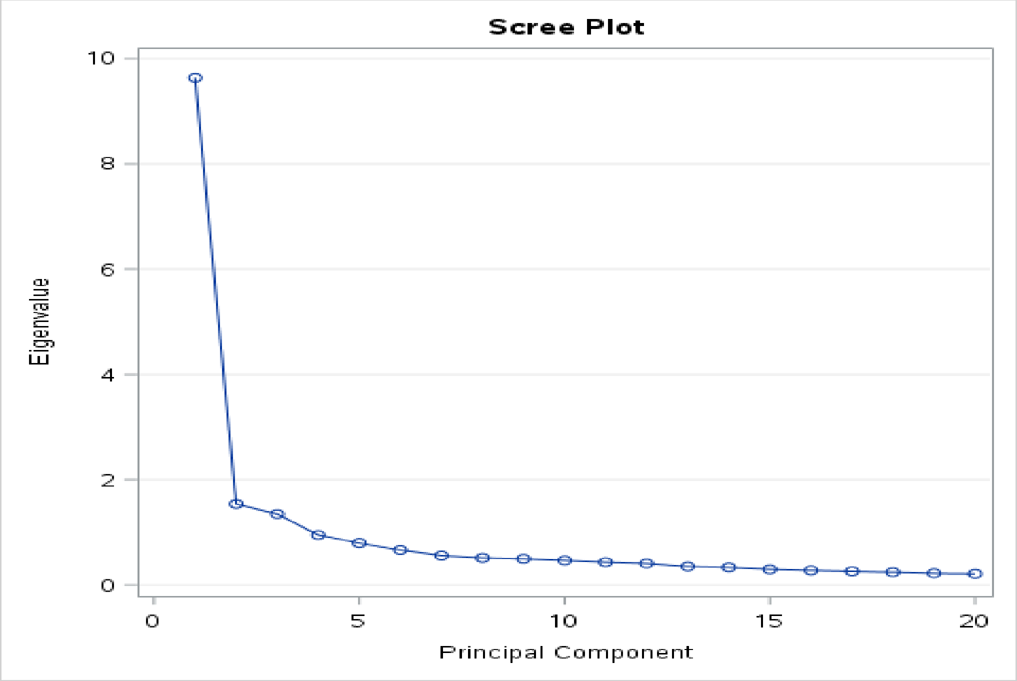
Another way to check which components to include in the analysis is through the use of graphics. The plot in Figure.6 gives us a visual representation of the eigenvalue chart we previously viewed.

Figure 6: Variance Plot



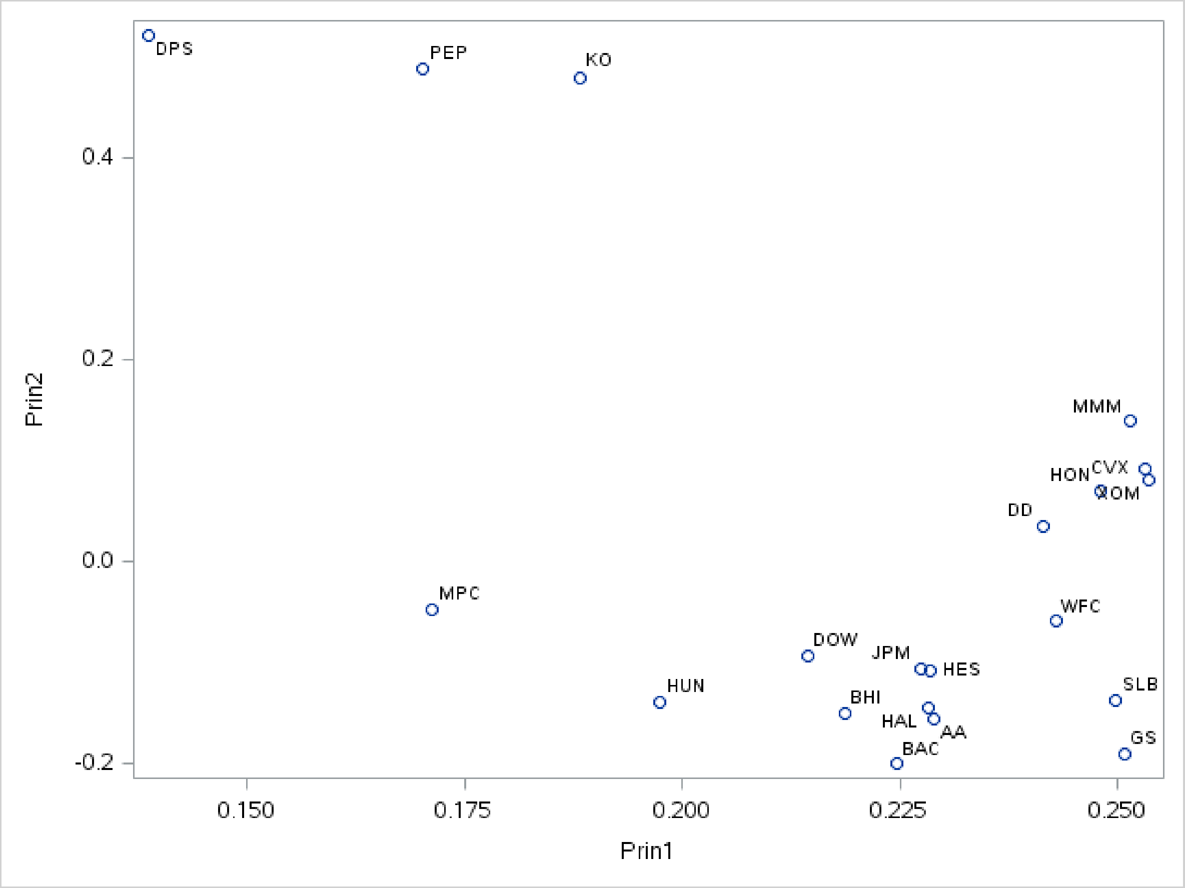
If we look at the Scree plot below, you can see that after the eighth component the slope of the line begins to flatten out, meaning that every component after that is accounting for less of the variability.

Figure 7: Scree Plot



The final graphic related to assessing the overall fit of our components is based on the first 2 loadings (Prin1 and Prin2). We do this to examine the relationship between the variables. Essentially, variables that find themselves grouped closely to one another have a higher correlation. If the variables find themselves on opposite ends, they have negative correlations. These relationships are based on the variable's correlations to the first two principal components. In our plot below, we do see some grouping of data points present. At the top of the graph, we can see that the soft drink stocks are group together. This is not surprising, given the nature of their business. Towards the bottom right of the graph, we do have some more compelling groupings taking place. BHI, HAL, AA, and BAC are all found plotted relatively close to one another. This is interesting for the simple fact that these stocks come from three different business sectors.

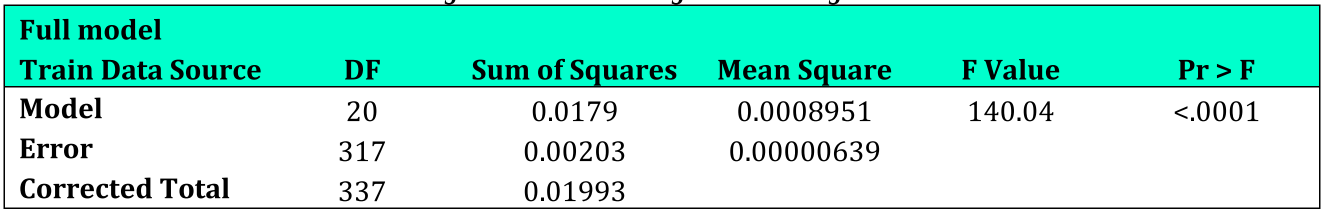
Figure 8: Plot of Loadings (Prin1 vs Prin2)



Modeling Comparisons:
To assess the benefits, validity, and accuracy of using principal components in regression modeling, we will use cross-validation techniques. To accomplish this goal, we will be creating a training and test split of our dataset on a 70-30 line, which will allow us to run our model using 70% of the dataset and examine the model's overall accuracy on the remaining 30% of the data. Also, for evaluation purposes, we will first fit a model without the use of PCA. This will give us a baseline from which we can compare and contrast the benefits, or lack of advantages in using PCA in model building.

As stated, we will run a regression model using all of the individual stocks in our dataset against the training dataset. Below are some of the results from the ANOVA analysis.

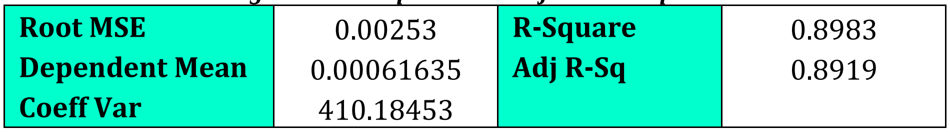
Figure 9: Full model against Training data.



Full model Train Data Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	20	0.0179	0.0008951	140.04	<.0001
Error	317	0.00203	0.00000639		
Corrected Total	337	0.01993			

The F-Value in Figure.9 is positive and strong while our P-Value is below our desired benchmark. Both metrics are indicators of the model's strength.

Figure 10: R-Square and Adjusted R-Square.

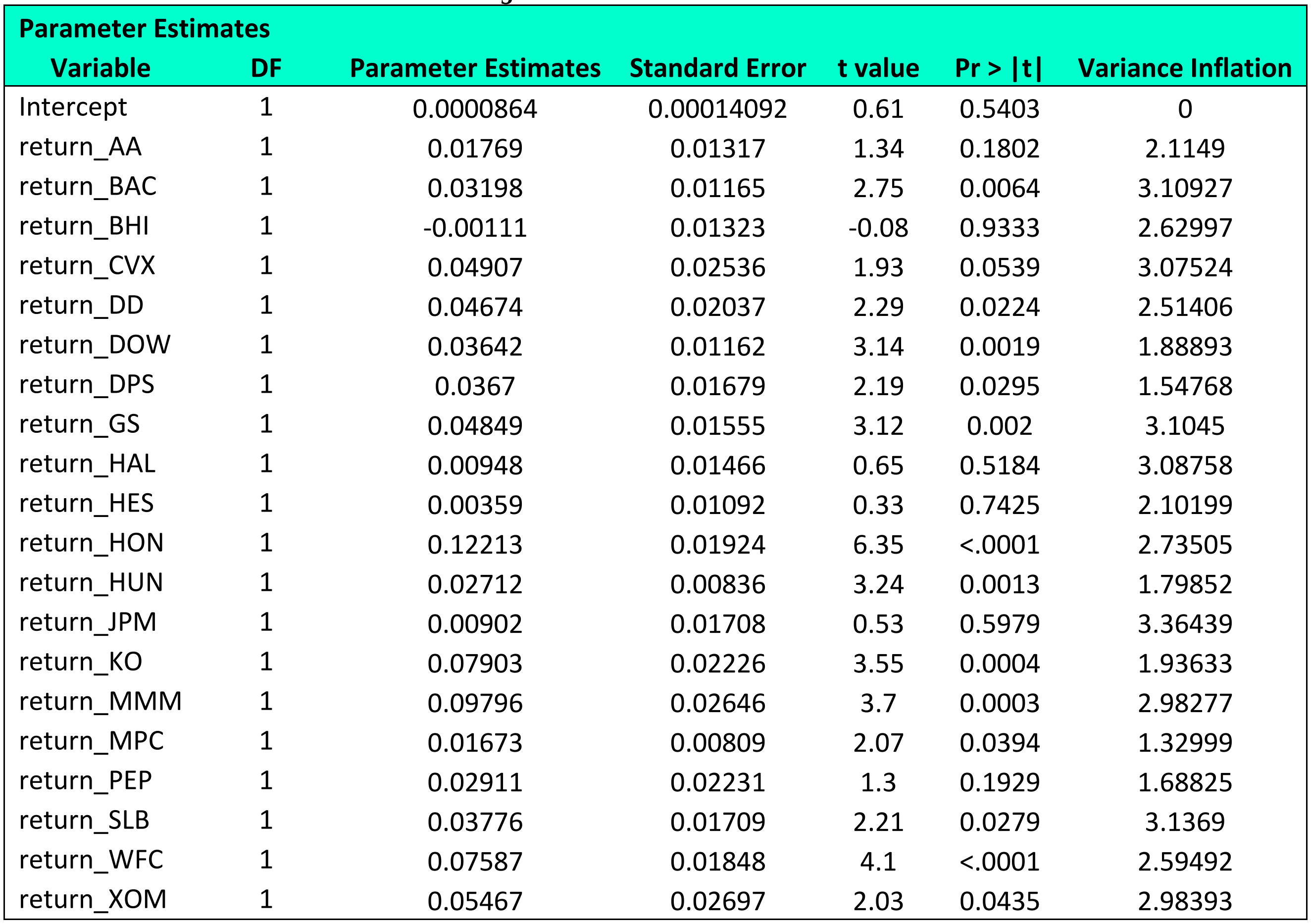


Root MSE	0.00253	**R-Square**	0.8983
Dependent Mean	0.00061635	**Adj R-Sq**	0.8919
Coeff Var	410.18453		

One component to assessing the goodness of fit for a model is to analyze the results of the R-Square and Adjusted R-Square. The results in Figure.10 show high values for R-Square and Adjusted R-Square. We typically want to see values over 50%, and we are well above those marks.

The parameter estimates below are accompanied by the variance inflation factors for each component of the model. Variance inflation factors (VIF) measure how much variance of the estimated regression coefficients are inflated as compared to when these independent variables are not linearly correlated to one another. VIF values that are equal to or close to 1.0 are considered acceptable and indicate that multicollinearity is not an issue. VIF values close to 4.0 can be regarded as problematic and indicative of multicollinearity. In Figure.11 we don't have any values that reach 4.0 but a lot of variables are higher than 1.0 which tells me that there is at least a presence of multicollinearity in our model

Figure 11: Parameter Estimates.



Parameter Estimates

Variable	DF	Parameter Estimates	Standard Error	t value	Pr > |t|	Variance Inflation
Intercept	1	0.0000864	0.00014092	0.61	0.5403	0
return_AA	1	0.01769	0.01317	1.34	0.1802	2.1149
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	0.0367	0.01679	2.19	0.0295	1.54768
return_GS	1	0.04849	0.01555	3.12	0.002	3.1045
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	3.7	0.0003	2.98277
return_MPC	1	0.01673	0.00809	2.07	0.0394	1.32999
return_PEP	1	0.02911	0.02231	1.3	0.1929	1.68825
return_SLB	1	0.03776	0.01709	2.21	0.0279	3.1369
return_WFC	1	0.07587	0.01848	4.1	<.0001	2.59492
return_XOM	1	0.05467	0.02697	2.03	0.0435	2.98393

We can now use the residuals to calculate the mean square error (MSE) and mean absolute error (MAE) for the training sample model we just evaluated and compare those values to the testing sample model's values of MSE and MAE. The MAE provides a simple and effective way for us to determine the forecast accuracy for our model, while MSE is the average of the square of differences between the actual observations and those predicted. What we want to see is similar values between our training and testing model. Examining the information in Figure.12 shows that our MAE is very close in value in both versions of the model. There is and should be a slight difference in the magnitude on the testing model, but overall it appears that the training model is a good fit, and presents evidence that it could serve a predictive purpose when compared to the testing data.

Figure 12: Mean Absolute Error for full model against Training and Test data.

Training Model		
N	**Mean Absolute Error**	**Mean Square Error**
338	0.001902	0.00000639
Testing Model		
N	**Mean Absolute Error**	**Mean Square Error**
163	0.0019514	0.00000828

We can now move forward and build a regression model utilizing PCA to fit the model with the first 8 principal components. The model will also be fitted initially against the training data set and later compared with the testing data set, to allow us to make an assessment of the model's predictive value.

Below are the results from the ANOVA analysis of our PCA model.

Figure 13: PCA model against Training data.

PCA model					
Train Data 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			

The F-Value in Figure.13 is positive and strong and shows an improvement over our first model and our P-Value is below our benchmark of 0.05.

Figure 14: R-Square and Adjusted R-Square for PCA model.

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

The goodness of fit results in Figure.14 shows high values for R-Square and Adjusted R-Square. as previously stated, we want to see values over 50% and we are well above those marks.

The parameter estimates below are once again accompanied by the variance inflation factors for each of the 8 component of the model. Our goal of utilizing PCA to build our model is to eliminate multicollinearity. VIF values that are equal to or close to 1.0 are considered acceptable and indicate to us that multicollinearity is not an issue. The values in Figure.15 are all close to 1.0, which suggests that multicollinearity is not present in this PCA model. This is a significant improvement over our first model.

Figure 15: Parameter Estimates for PCA model.

Parameter Estimates						
Variable	**DF**	**Parameter Estimates**	**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	2.1	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	-0.0004076	0.00020293	-2.01	0.0454	1.02271

Since the PCA model appears to be outperforming the full model in terms of fit and statistical strength, the final assessment with respect to its usefulness is to evaluate its predictive ability. We will fit the model against the testing data and attempt to make that judgment.

Figure 16: Mean Absolute Error for PCA model against Training and Test data.

Training Model		
N	**Mean Absolute Error**	**Mean Square Error**
338	0.0019752	0.00000659
Testing Model		
N	**Mean Absolute Error**	**Mean Square Error**
163	0.0020142	0.00000884

Figure.16 shows that our MAE and MSE is very close in value in both versions of the PCA model. These figures indicate to us that our PCA model does, in fact, have good predictive ability.

As a final check of the strengths and merits of our PCA model, we have a table with some summary data we can use to compare the PCA model to the Full model. The table below also includes the results of both models when fitted with the training and test datasets.

Figure 17: Model Comparisons.

Model	R-Square	Adjusted R-Square	F-Value	Pr > F	MAE	MSE
Full Training	0.8983	0.8919	140.04	<.0001	0.0019020	0.00000639
Full Testing	0.8679	0.8492	46.63	<.0001	0.0019514	0.00000828
PCA Training	0.8913	0.8886	337.13	<.0001	0.0019752	0.00000659
PCA Testing	0.8470	0.8390	106.56	<.0001	0.0020142	0.00000884

From the information above we can determine that there is very little difference between the models, especially with respect to the MAE values, which means that all the models serve their predictive purpose in sample and out of sample. The F-Statistic is higher in the PCA model, and we must point out the fact that there were no issues of multicollinearity in our PCA model. The full model had issues with multicollinearity. Also, we should note that are far fewer variables in the PCA model making it easier to interpret, while maintaining its predictive power. These facts lead us to the determination that the PCA model is a preferred choice over the Full model.

Conclusion:

We examined 2 different linear models in this analysis. In each case we found the models to be statistically significant when it came to the regression analysis numeric fit statistics. The issue of multicollinearity is what distinguished the models from one another. Mutlicollinearity is an issue that can lead us to misleading results. The use of Principal Component Analysis combats multicollinearity by reducing multidimensional data, utilizing less than the full set of principal components in the model, while retaining most of the information.

Apendix:SAS Code

```
/*Read file into SAS and identify it as mydata*/
libname mydata "/scs/wtm926/" access=readonly;

data temp;
set mydata.stock_portfolio_data; run;

proc contents data=temp order=varnum;
run;

proc print data= temp;

/***1.Compute the log-returns - log of the ratio of today's price to yesterday's price***/
/***Note that the data needs to be sorted************************************************/
proc sort data=temp; by date; run; quit; data temp;
set temp;
return_AA = log(AA/lag1(AA));
return_BAC = log(BAC/lag1(BAC));
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=3); run; quit;

/*** 2.look at the correlations between the individual stocks and the market index***/
/***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;

/***3.SAS has two “types” of data sets or data formats long/wide. use PROC TRANSPOSE to transform***/
/***Note that the output has much more output than we need/want are the correlations between the returns***/
data wide_correlations;
set portfolio_correlations (keep=return_:);
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);
drop _NAME_;
rename COL1=correlation;
run;

proc print data=long_correlations; run; quit;


/***4.When working with a large amount of predictor variables, it can be helpful to use visualizations***/
/*** instead of tables. As a visualization we will create a colored bar plot of the correlations***/
data sector;
input tkr $ 1-3 sector $ 4-35;
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;

/*** Sort data by correlations ***/
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 ***/
ods graphics on;
title 'Correlations with the Market Index';
proc sgplot data=long_correlations;
format correlation 3.2;
vbar tkr / response=correlation group=sector groupdisplay=cluster datalabel categoryorder =
respasc;
run; quit;
ods graphics off;

/***bar plot grouped by sector***/
ods graphics on;
title 'Correlations with the Market Index';
proc sgplot data=long_correlations;
  format correlation 3.2;
  vbar sector / response=correlation group=tkr groupdisplay=cluster datalabel categoryorder =
respasc;
run; quit;
ods graphics off;
```

```
* SAS can produce bar plots by sector of the mean correlation;
proc means data=long_correlations nway noprint;
class sector;
var correlation;
output out=mean_correlation mean(correlation)=mean_correlation;
run; quit;

proc print data=mean_correlation; run;

ods graphics on;
title 'Mean Correlations with the Market Index by Sector';
proc sgplot data=mean_correlation;
        format mean_correlation 3.2;
        vbar sector / response=mean_correlation datalabel;
run; quit;
ods graphics off;

/*** 5. Compute the (PCA)principal components for the return
data****************************/
/*** How many componenst do we
need********************************************************/
data return_data;
set temp (keep= return_:);
run;

/***PCA***/
ods graphics on;
proc princomp data=return_data out=pca_output outstat=eigenvectors plots=scree(unpackpanel);
run; quit;
ods graphics off;
* Notice that PROC PRINCOMP produces a lot of output;
proc print data=pca_output(obs=10); run;
proc print data=eigenvectors(where=(_TYPE_='SCORE')); run;
/***** Display the two plots and the Eigenvalue table from the output *****/

/***Plot the first two eigenvectors***/
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 principal components***/
ods graphics on;
proc sgplot data=long_pca;
scatter x=Prin1 y=Prin2 / datalabel=tkr;
run; quit;
ods graphics off;

/***6. using principal components in regression modeling. Split the data into train/test data sets***/
/***create a training data set and a testing data set from the PCA output***/
/***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);
* 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=.;
  if (u > 0.70) then test = 1;
  else test = 0;
  if (test=1) then test_response=response_VV;
  else test_response=.;
run;
proc print data=cv_data(obs=10); run;

/***7. Fit a regression model using all of the stocks with train_response as the response
variable***/
/***Have SAS output the predicted values and the variance inflation factors***/
proc reg data=cv_data;
model train_response = return_: / vif ;
output out=modelall_output predicted=Yhat;

/***Find the mean square error (MSE) and mean absolute error(MAE)***/
proc print data=modelall_output(obs=10);
Data modelall_outputb;
set modelall_output;
mae = abs(yhat - train_response);

proc print data=modelall_outputb(obs=5);
proc means data=modelall_outputb;
var mae;
title 'MAE calculation Training';
```

```
/***test response information***/
/******************************/
proc reg data=cv_data;
model test_response = return_: / vif ;
output out=modeltest_output predicted=Yhat;

/***Find the mean square error (MSE) and mean absolute error(MAE)***/
proc print data=modeltest_output(obs=10);
Data modeltest_outputb;
set modeltest_output;
mae = abs(yhat - test_response);

proc print data=modeltest_outputb(obs=5);
proc means data=modeltest_outputb;
var mae;
title 'MAE calculation Training';


/***8. Fit a regression model using first 8 PCA and train_response as the response variable***/
/***Have SAS output the predicted values and the variance inflation factors***/
proc reg data=cv_data;
model train_response = prin1-prin8 / vif ;
output out=modelPCA_output predicted=Yhat ;

/***Find the mean square error (MSE) and mean absolute error(MAE)***/
proc print data=modelPCA_output(obs=10);
Data modelPCA_outputb;
set modelPCA_output;
mae = abs(yhat - train_response);

proc print data=modelPCA_outputb(obs=5);
proc means data=modelPCA_outputb;
var mae;
title 'MAE calculation Training';

/***test response***/
proc reg data=cv_data;
model test_response = prin1-prin8 / vif;
output out=modelPCAt_output predicted=Yhat;

/***Find the mean square error (MSE) and mean absolute error(MAE)***/
proc print data=modelPCAt_output(obs=10);
Data modelPCAt_outputb;
set modelPCAt_output;
mae = abs(yhat - test_response);
```

```
proc print data=modelPCAt_outputb(obs=5);
proc means data=modelPCAt_outputb;
var mae;
title 'MAE calculation Training';

quit;
```

References

(1) Black, K. (2008). *Business statistics: For contemporary decision making*. Hoboken, NJ: Wiley.

(2) Montgomery, D. C., Peck, E. A., Vinning, G. G., (2012). *Introduction to Linear Regression Analysis* Hoboken, NJ: Wiley.

(3) Wilcox, R. R. (2009). *Basic statistics: Understanding conventional methods and modern insights*.Oxford: Oxford University Press.

(4) Cody, R. (2011). *SAS: Statistics by Example*. Carey, NC: SAS Institute Inc.

(5) Evaluating forecast Accuracy. https://www.otexts.org/fpp/2/5 (accessed February 4, 2017)

(6) Principal Component Analysis Loading Plot. http://wiki.q-researchsoftware.com/wiki/Principal_Components_Analysis_Loading_Plot (accessed February 11, 2017)

(7) What in The World is a VIF. http://blog.minitab.com/blog/starting-out-with-statistical-software/what-in-the-world-is-a-vif (accessed February 11, 2017)