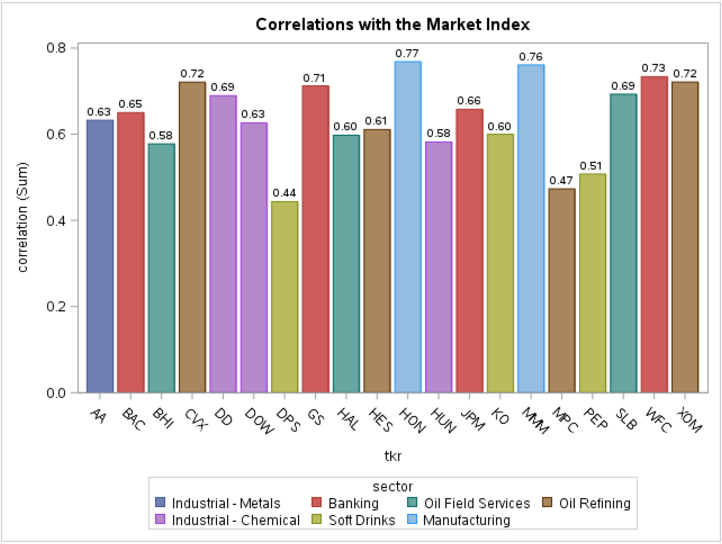
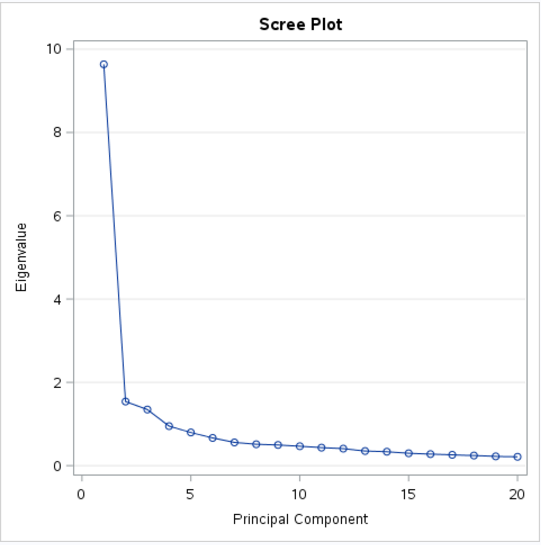
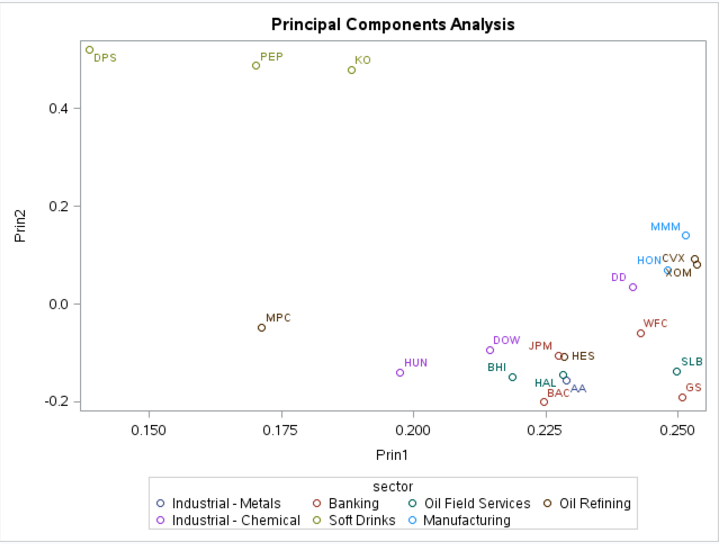
**Correlation Analysis**

The first step in our analysis is to perform the log transformation on the twenty stocks and the Vanguard market index fund. Next, we will plot a Pearson correlation matrix for each stock against the market index . As you can see below, many of the variables have a positive correlation with the market index. You can see several stocks in the same sector have similar correlations with the response variable (eg. CVX and XOM, HON and MMM).



**Principal Component Analysis (PCA)**

The next phase in this assignment is to perform Principal Component Analysis (PCA) on the predictor variables. After running the PRINCOMP procedure in SAS, you can see the scree plot below. The Kaiser Rule tells us to drop all components with eigenvalues below 1.0, which in this case would suggest we should retain only three components.

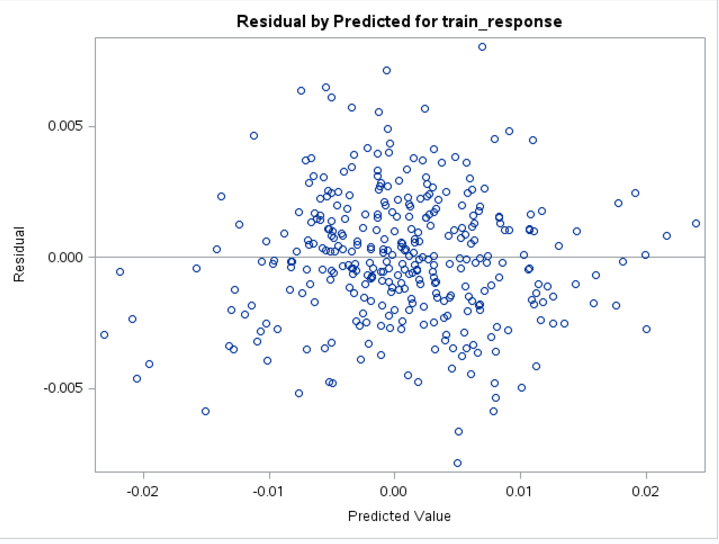
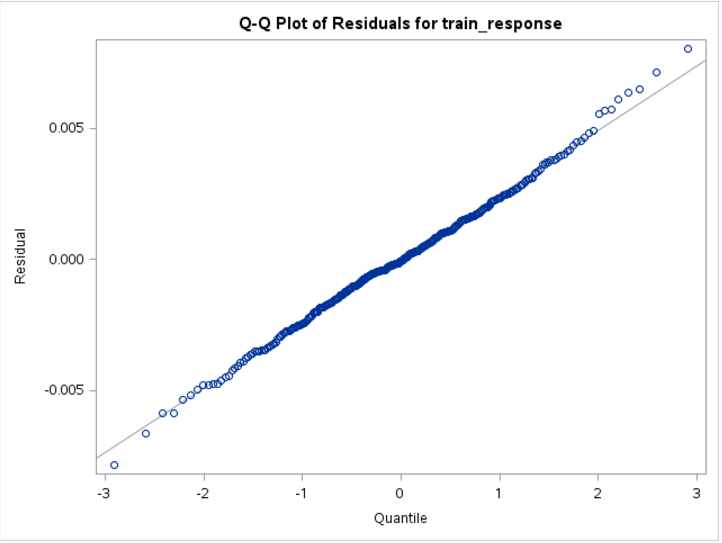
 

**Regression Models Leveraging Principal Component Analysis**

In this section, we will build two regression models in order to assess if performing PCA improves the predictive accuracy of the model.

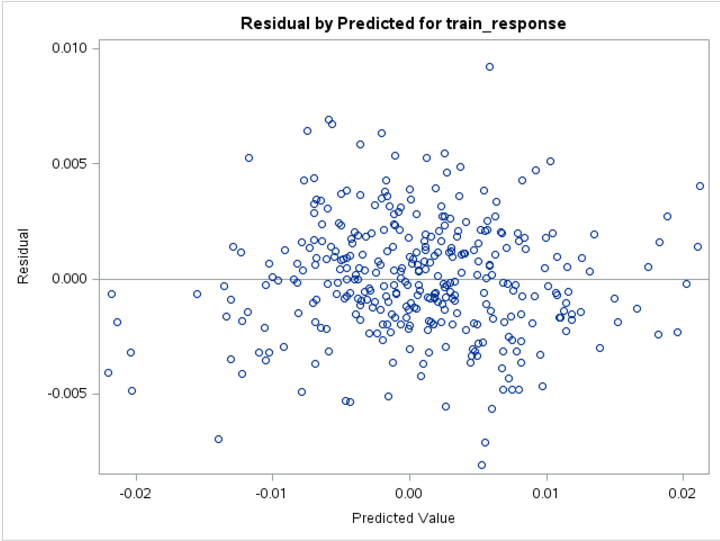
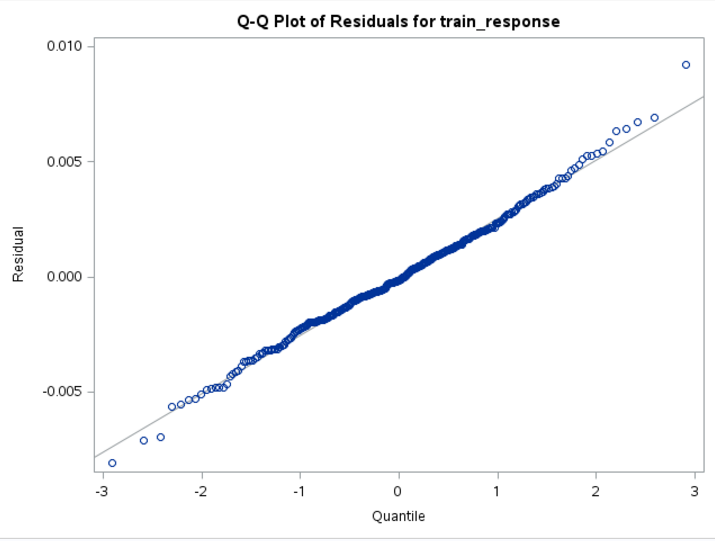
*Model 1 – All Log-Returns for each Stock against the Log-Return of the Market Index - VV*

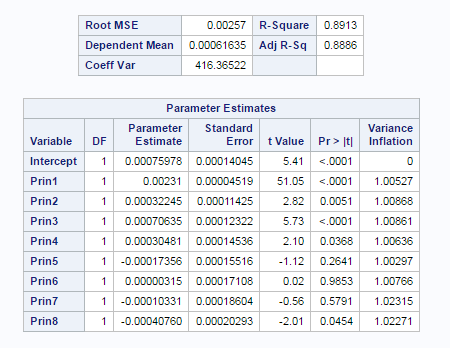
The first model we will assess include all log-return values for each stock as the predictor variables and the log-return of the Vanguard market index (response\_VV). From a goodness-of-fit perspective, the residual plots all show constant variance for all observations as shown in Figure 5. Another indication that the model has a good fit is the straight line in the QQ plot for the model. Lastly, the adjusted R-squared is 0.8919, which indicates a very good fit to the data. VIF values are well beyond 5 so we can conclude that no multicollinearity exists among the predictor variables.

*Model 2 – 8 components from PCA against the Log-Return of the Market Index - VV*

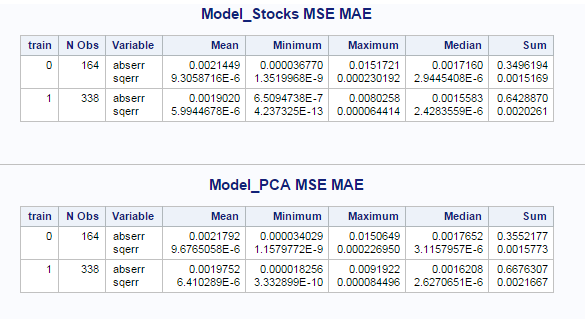
In the second model, we have gone down the path of performing principal component analysis. As can be seen below, the residuals appear to have constant variance in the model, similar to previous model. The QQ plot has a straight line, which is an indication of good model fit. Lastly, the adjusted R-squared for Model 2 is 0.8886, which is also a good indication of model fit. VIF values are much lower than in the previous model.



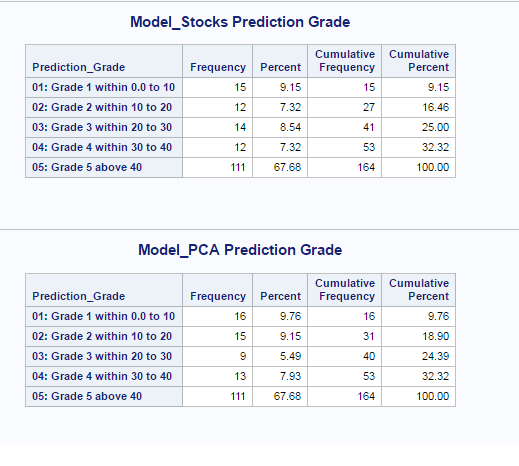
**Comparison of Model 1 and Model 2**

As you can see below, the MSE, and MAE are slightly better in Model 1 than in Model 2 for both the training and test data sets Given we did not see any indications that multicollinearity was present among the predictor variables, this would make sense why we did not see any performance improvement from leveraging PCA.



**Predictive Accuracy & Final Model Selection**

Now that we have assessed the performance of the models in the statistical sense, we will now compare the models based on their predictive accuracy. By comparing the predicted values vs the actual values for both Model 1 and Model 2, we can evaluate models from business point of view.



As we can see by comparing the frequency tables for the results of the predicted values against the actual values, there is not much of a difference between both models. Model 1 performs slightly better having larger portion of correct predictions within 30%.

**Conclusions**

I could not find any strong evidence of multicollinearity so both models performed pretty much the same. Overall predictive accuracy for both models was rather lousy so I think that other predictive algorithms than linear regressions should be tried out.

libname mydata '/scs/crb519/PREDICT\_410/SAS\_Data/' access=readonly;

proc datasets library=mydata; run; quit;

ods graphics on;

ods noproctitle;

title 'Assignment 6';

data temp;

set mydata.stock\_portfolio\_data;

run;

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));

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;

\* 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 output PearsonCorr=portfolio\_correlations;

proc corr data=temp;

var return\_:;

with response\_VV;

title 'Portfolio Correlations';

run; quit;

proc print data=portfolio\_correlations;

title 'Portfolio Correlations';

run; quit;

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;

\* Merge on sector id and make a colored bar plot;

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;

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;

title 'Correlations with the Market Index';

proc sgplot data=long\_correlations;

format correlation 3.2;

vbar tkr / response=correlation group=sector groupdisplay=cluster datalabel;

run; quit;

proc means data=long\_correlations;

class Sector;

title 'Correlations by Market Sector';

run; quit;

data return\_data;

set temp (keep= return\_:);

\* What happens when I put this keep statement in the set statement?;

\* Look it up in The Little SAS Book;

run;

proc print data=return\_data(obs=10); run;

title 'Principal Components Analysis';

proc princomp data=return\_data out=pca\_output outstat=eigenvectors plots=scree(unpackpanel);

run; quit;

\* Notice that PROC PRINCOMP produces a lot of output;

\* How many principal components should we keep?;

\* 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(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;

proc sort data=long\_pca; by tkr; run;

data long\_pca;

merge long\_pca (in=a) sector (in=b);

by tkr;

if (a=1) and (b=1);

run;

proc sgplot data=long\_pca;

scatter x=Prin1 y=Prin2 / datalabel=tkr group=sector;

run; quit;

\* Do we see anything interesting here? Why would we make such a plot?;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* 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);

\* 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;

\* create macros for printing the estimators and calculating MSE and MAE;

%macro data\_validation(indata,outdata);

data &outdata.;

set &indata.;

err = response\_VV - yhat;

abserr = abs(err);

sqerr = (err) \*\* 2;

if(abserr <= response\_VV \* 0.1) then

Prediction\_Grade = '01: Grade 1 within 0.0 to 10';

else if (abserr <= response\_VV \* 0.2) then

Prediction\_Grade = '02: Grade 2 within 10 to 20';

else if (abserr <= response\_VV \* 0.3) then

Prediction\_Grade = '03: Grade 3 within 20 to 30';

else if (abserr <= response\_VV \* 0.4) then

Prediction\_Grade = '04: Grade 4 within 30 to 40';

else

Prediction\_Grade = '05: Grade 5 above 40';

%mend;

%macro print\_mse\_mae(indata,mytitle);

proc means data=&indata. mean min max median sum;

var abserr sqerr;

class train;

title &mytitle;

run; quit;

%mend;

%macro print\_grade(indata,mytitle);

proc freq data=&indata.;

tables Prediction\_Grade;

where train=0;

title &mytitle.;

run; quit;

%mend;

proc reg data = cv\_data plots = diagnostics(unpack) outest = Model\_Stocks\_est;

Model\_Stocks: model train\_response =

return\_AA return\_BAC return\_BHI return\_CVX

return\_DD return\_DOW return\_DPS return\_GS

return\_HAL return\_HES return\_HON return\_HUN

return\_JPM return\_KO return\_MMM return\_MPC

return\_PEP return\_SLB return\_WFC return\_XOM

/ vif;

output out=Model\_Stocks\_out (keep=train response\_VV yhat) predicted=yhat;

title 'Model\_Stocks';

run;

proc reg data = cv\_data plots = diagnostics(unpack)outest = Model\_PCA\_est;

Model\_PCA: model train\_response =

Prin1 Prin2 Prin3 Prin4

Prin5 Prin6 Prin7 Prin8

/ vif;

output out=Model\_PCA\_out (keep=train response\_VV yhat) predicted=yhat;

title 'Model\_PCA';

run;

%data\_validation(indata=%str(Model\_Stocks\_out),outdata=%str(Model\_Stocks\_validation));

%print\_mse\_mae(indata=%str(Model\_Stocks\_validation),mytitle=%str('Model\_Stocks MSE MAE'));

%data\_validation(indata=%str(Model\_PCA\_out),outdata=%str(Model\_PCA\_validation));

%print\_mse\_mae(indata=%str(Model\_PCA\_validation),mytitle=%str('Model\_PCA MSE MAE'));

%print\_grade(indata=%str(Model\_Stocks\_validation),mytitle=%str('Model\_Stocks Prediction Grade'));

%print\_grade(indata=%str(Model\_PCA\_validation),mytitle=%str('Model\_PCA Prediction Grade'));