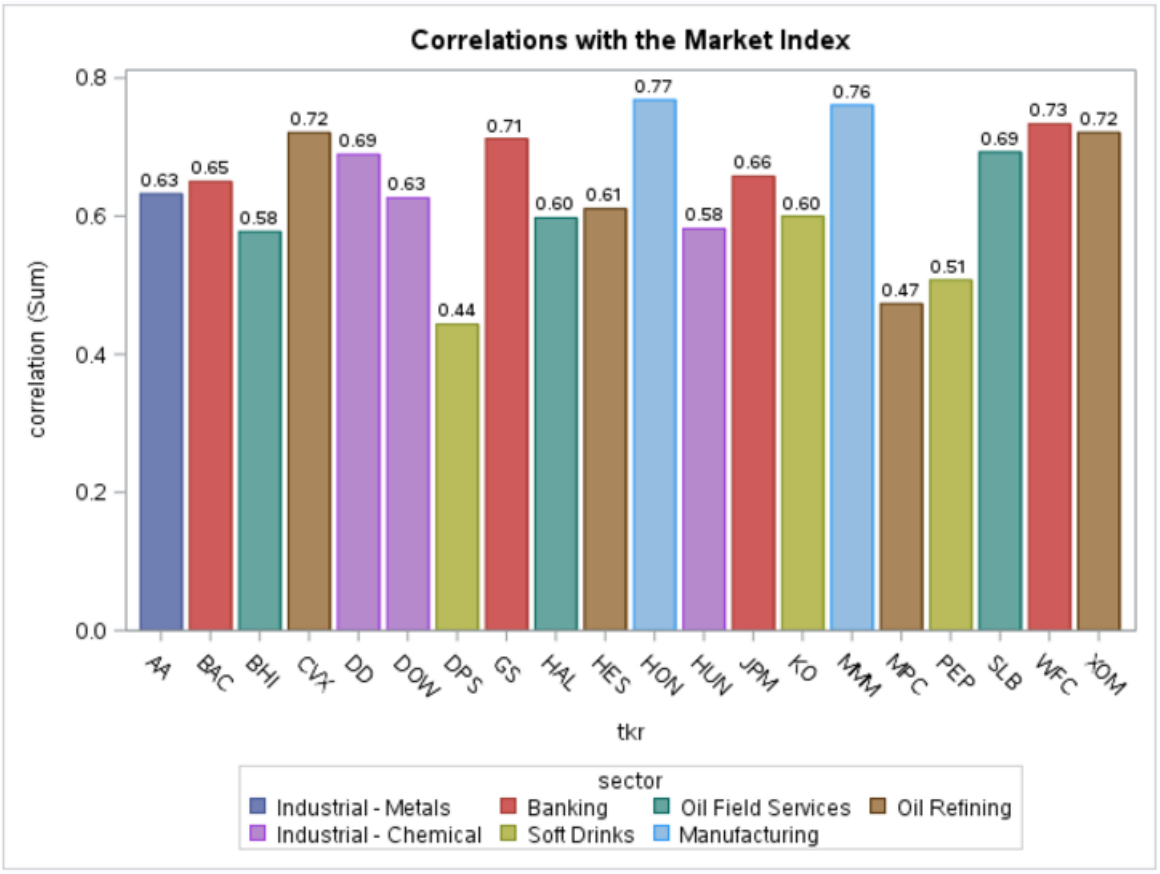
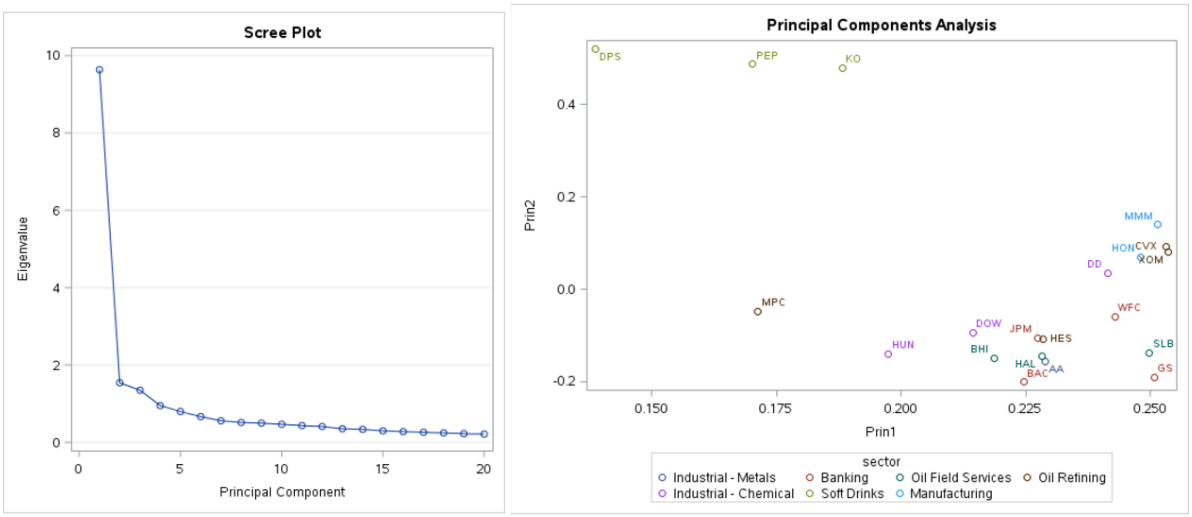
Correlation Analysis

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.

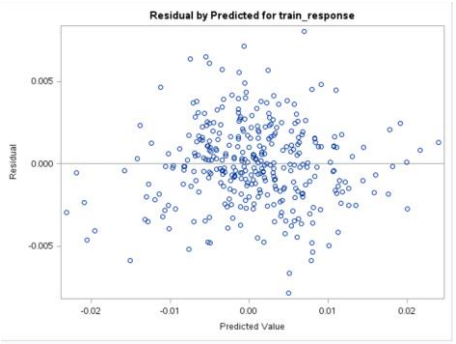


Regression Models Leveraging Principal Component Analysis

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.



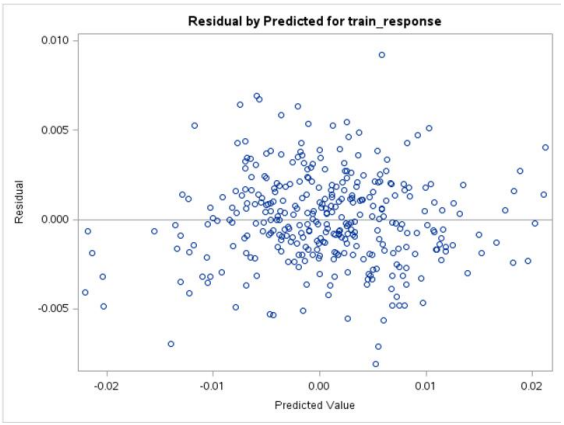
Root MSE	0.00253	R-Square	0.8983
Dependent Mean	0.00061635	Adj R-Sq	0.8919
Coeff Var	410.18453		

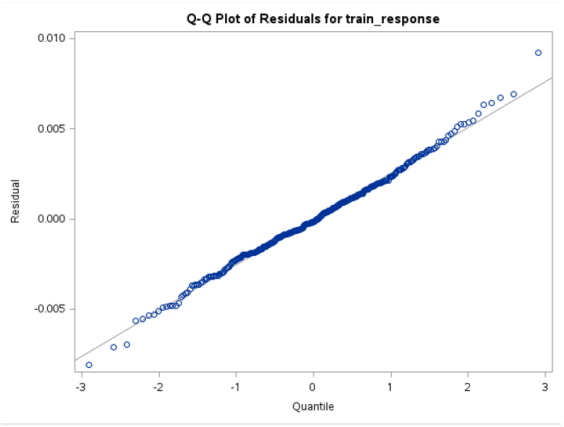
Parameter Estimates

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

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.





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

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

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.

Model_Stocks MSE MAE

train	N Obs	Variable	Mean	Minimum	Maximum	Median	Sum
0	164	abserr	0.0021449	0.000036770	0.0151721	0.0017160	0.3496194
		sqerr	9.3058716E-6	1.3519968E-9	0.000230192	2.9445408E-6	0.0015169
1	338	abserr	0.0019020	6.5094738E-7	0.0080258	0.0015583	0.6428870
		sqerr	5.9944678E-6	4.237325E-13	0.000064414	2.4283559E-6	0.0020261

Model_PCA MSE MAE

train	N Obs	Variable	Mean	Minimum	Maximum	Median	Sum
0	164	abserr	0.0021792	0.000034029	0.0150649	0.0017652	0.3552177
		sqerr	9.6765058E-6	1.1579772E-9	0.000226950	3.1157957E-6	0.0015773
1	338	abserr	0.0019752	0.000018256	0.0091922	0.0016208	0.6676307
		sqerr	6.410289E-6	3.332899E-10	0.000084496	2.6270651E-6	0.0021667

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.

Model_Stocks Prediction Grade

Prediction_Grade	Frequency	Percent	Cumulative Frequency	Cumulative Percent
01: Grade 1 within 0.0 to 10	15	9.15	15	9.15
02: Grade 2 within 10 to 20	12	7.32	27	16.46
03: Grade 3 within 20 to 30	14	8.54	41	25.00
04: Grade 4 within 30 to 40	12	7.32	53	32.32
05: Grade 5 above 40	111	67.68	164	100.00

Model_PCA Prediction Grade

Prediction_Grade	Frequency	Percent	Cumulative Frequency	Cumulative Percent
01: Grade 1 within 0.0 to 10	16	9.76	16	9.76
02: Grade 2 within 10 to 20	15	9.15	31	18.90
03: Grade 3 within 20 to 30	9	5.49	40	24.39
04: Grade 4 within 30 to 40	13	7.93	53	32.32
05: Grade 5 above 40	111	67.68	164	100.00

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'));
```