Assignment #6
Crystal Mosley

Introduction:

The goal of this data examination and modeling is to utilize the Principal Components Analysis as a method of dimension reduction, and as a remedial measure for multicollinearity in OLS regression.

The data being examined consists of 20 daily closing stock prices and an index fund from Vanguard—VV. The data range is from 01/03/12 – 12/31/13 which totals to 501 days without any noticeable gaps of days skipped, or null values. The stock values will be the independent variables, and Vanguard as the dependent variable.

We will begin with prepping the data and using the log-returns of each stock to explain the variation in the log-returns of the market index. We'll then review the correlations between the individual stocks and the market index. Next, we'll compute the PCA for the return data and move on to regression modeling.

Results:

Compute the log-return of each stock/index variable—

Return: r_i at a time i, where p_i is the price at a time i; j = (i-1)

$$ri = \frac{pi - pj}{pj}$$

Log-return: log(r_i); time i is in days

Return is used instead of price because it provides a mechanism of normalization—this allows for a measurement of all variables in a comparable metric, which will enable us to evaluate the relationships between two or more variables regardless of originating from a series of unequal values.

Obs	Date	AA	BAC	BHI	CVX	DD	DOW	DPS	GS	HAL	HES	HON	HUN	JPM	KO	MMM	MPC	PEP	SLB	WFC	XOM	VV
1	31DEC2013	10.63	15.57	55.26	124.91	64.97	44.4	48.72	177.26	50.75	83	91.37	24.6	58.48	41.31	140.25	91.73	82.94	90.11	45.4	101.2	84.8
2	30DEC2013	10.53	15.54	54.45	124.23	64.65	44.6	48.84	175.73	50.4	82.61	91	24.33	57.95	41.09	139.42	88.5	82.91	89.17	45.5	100.31	84.37
3	27DEC2013	10.69	15.67	54.77	125.23	64.25	44.6	48.8	176.35	51.08	83.07	91.14	24.24	58.14	40.66	139.35	89.13	82.71	89.9	45.5	101.51	84.45
4	26DEC2013	10.43	15.65	54.58	124.81	64.25	44.86	48.59	176.45	51.21	82.6	91.1	24.01	58.2	40.49	138.29	89.54	82.45	89.39	45.54	100.9	84.45
5	24DEC2013	10.36	15.7	54	123.51	63.83	44.42	48.5	176.16	50.68	81.14	90.45	24.14	58.25	40.19	136.99	89.35	82.04	88.31	45.39	99.22	84.07
6	23DEC2013	10.13	15.69	53.64	122.8	62.74	43.88	48.19	176.47	50.3	80.22	89.72	23.94	58.24	40.16	136.8	88.77	81.86	87.32	45.21	98.51	84.31
7	20DEC2013	9.94	15.6	53.81	122.78	62.55	43.78	48.42	175.16	50.54	80.17	89.55	23.77	57.7	40.04	136.72	87.26	81.81	87.27	44.96	98.68	83.85
8	19DEC2013	9.97	15.75	53.19	123.22	62.71	43.9	47.8	174.77	49.42	80.54	88.64	23.77	57.23	39.86	136.41	85.71	81.66	86.47	45.02	99.43	83.36
9	18DEC2013	9.83	15.69	53.66	121.6	62.19	43.51	48.22	174.84	49.91	80.83	88.45	23.79	57.24	40.02	135.8	87.42	81.42	87.26	44.93	99.54	83.5
10	17DEC2013	9.7	15.18	52.66	118.74	61.18	42.2	47.35	170.49	49.35	79.51	86.48	23.55	55.72	39.1	131.39	86.09	80.46	85.54	43.59	96.75	82.07

Obs	Date	AA	BAC	BHI	CVX	DD	DOW	DPS	GS	HAL	HES	HON	HUN	JPM	KO	MMM	MPC	PEP	SLB	WFC	XOM	VV
1	03JAN2012	9.23	5.8	51.02	110.37	46.51	29.79	38.34	95.36	34.15	58.4	55.58	9.95	34.98	35.07	83.49	33.41	66.4	70.09	28.43	86	58.18
2	04JAN2012	9.45	5.81	51.53	110.18	47.02	29.95	38.55	94.74	35.12	59	55.53	9.87	34.95	34.85	84.18	33.76	66.74	69.56	28.56	86.02	58.25
3	05JAN2012	9.36	6.31	50.82	109.1	46.7	30.14	38.79	94.58	34.56	57.6	55.59	9.82	35.68	34.68	83.8	31.92	66.22	68.07	29.02	85.76	58.44
4	06JAN2012	9.16	6.18	51.26	108.31	46.04	30.32	38.52	93.42	34.98	56.42	55.18	9.9	35.36	34.46	83.37	31.66	65.39	67.78	28.94	85.12	58.32
5	09JAN2012	9.42	6.27	51.58	109.49	46.43	30.31	38.52	94.69	35.38	56.9	55.64	9.84	35.3	34.46	83.87	30.96	65.73	68.82	29.3	85.5	58.45
6	10JAN2012	9.44	6.63	51.95	109.06	47.14	30.98	38.62	98.33	36.33	58.56	56.58	10.2	36.05	34.67	84.3	31.83	65.66	70.75	29.41	85.72	58.99
7	11JAN2012	9.63	6.87	50.21	107.77	47.3	31.43	38.15	99.76	35.38	57.94	56.46	10.59	36.66	34.03	83.77	32.89	65.01	70.16	29.62	85.08	59.06
8	12JAN2012	9.93	6.79	48.29	104.97	48.1	32.56	37.96	101.21	34.73	57.01	57.19	11.03	36.85	33.78	84.28	32.64	64.62	69.7	29.61	84.74	59.2
9	13JAN2012	9.8	6.61	48.02	106.09	48.4	32.02	37.82	98.96	33.94	56.55	56.7	11.21	35.92	33.5	83.6	33.03	64.4	67.99	29.61	84.88	58.95
10	17JAN2012	9.76	6.48	47.7	106.72	48.54	32.64	37.63	97.68	33.86	57.39	57.16	10.85	34.91	33.68	84.23	33.54	64.65	67.64	29.82	85.69	59.1

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
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0.023556	0.001723	0.009946	-0.001723	0.010906	0.005357	0.005462	-0.006523	0.028008	0.010222	-0.000900	-0.008073	-0.000858	-0.006293	0.008230499
-0.009569	0.082555	-0.013874	-0.009850	-0.006829	0.006324	0.006206	-0.001690	-0.016074	-0.024015	0.001080	-0.005079	0.020672	-0.004890	-.004524356
-0.021599	-0.020817	0.008621	-0.007267	-0.014234	0.005954	-0.006985	-0.012341	0.012080	-0.020699	-0.007403	0.008114	-0.009009	-0.006364	-.005144475
0.027989	0.014458	0.006223	0.010836	0.008435	-0.000330	0.000000	0.013503	0.011370	0.008472	0.008302	-0.006079	-0.001698	0.000000	0.005979449
0.002121	0.055828	0.007148	-0.003935	0.015176	0.021864	0.002593	0.037721	0.026497	0.028757	0.016753	0.035932	0.021024	0.006076	0.005113884
0.019927	0.035559	-0.034068	-0.011899	0.003388	0.014421	-0.012245	0.014438	-0.026497	-0.010644	-0.002123	0.037522	0.016779	-0.018632	-.006306917
0.030677	-0.011713	-0.038990	-0.026325	0.016772	0.035322	-0.004993	0.014430	-0.018543	-0.016181	0.012847	0.040709	0.005169	-0.007374	0.006069641
-0.013178	-0.026867	-0.005607	0.010613	0.006218	-0.016724	-0.003695	-0.022482	-0.023010	-0.008101	-0.008605	0.016187	-0.025561	-0.008323	-.008101069
-0.004090	-0.019863	-0.006686	0.005921	0.002888	0.019178	-0.005036	-0.013019	-0.002360	0.014745	0.008080	-0.032641	-0.028521	0.005359	0.007507632

return_MPC	return_PEP	return_SLB	return_WFC	return_XOM	response_VV
.	.	.	.	.	.
0.010421	0.005107	-0.007590	0.004562	0.000232531	0.001202439
-0.056044	-0.007822	-0.021653	0.015978	-.003027130	0.003256494
-0.008179	-0.012613	-0.004269	-0.002761	-.007490672	-.002055499
-0.022358	0.005186	0.015227	0.012363	0.004454350	0.002226600
0.027713	-0.001066	0.027658	0.003747	0.002569795	0.009196250
0.032759	-0.009949	-0.008374	0.007115	-.007494180	0.001185938
-0.007630	-0.006017	-0.006578	-0.000338	-.004004245	0.002367666
0.011878	-0.003410	-0.024840	0.000000	0.001650749	-.004231915
0.015323	0.003874	-0.005161	0.007067	0.009497638	0.002541297

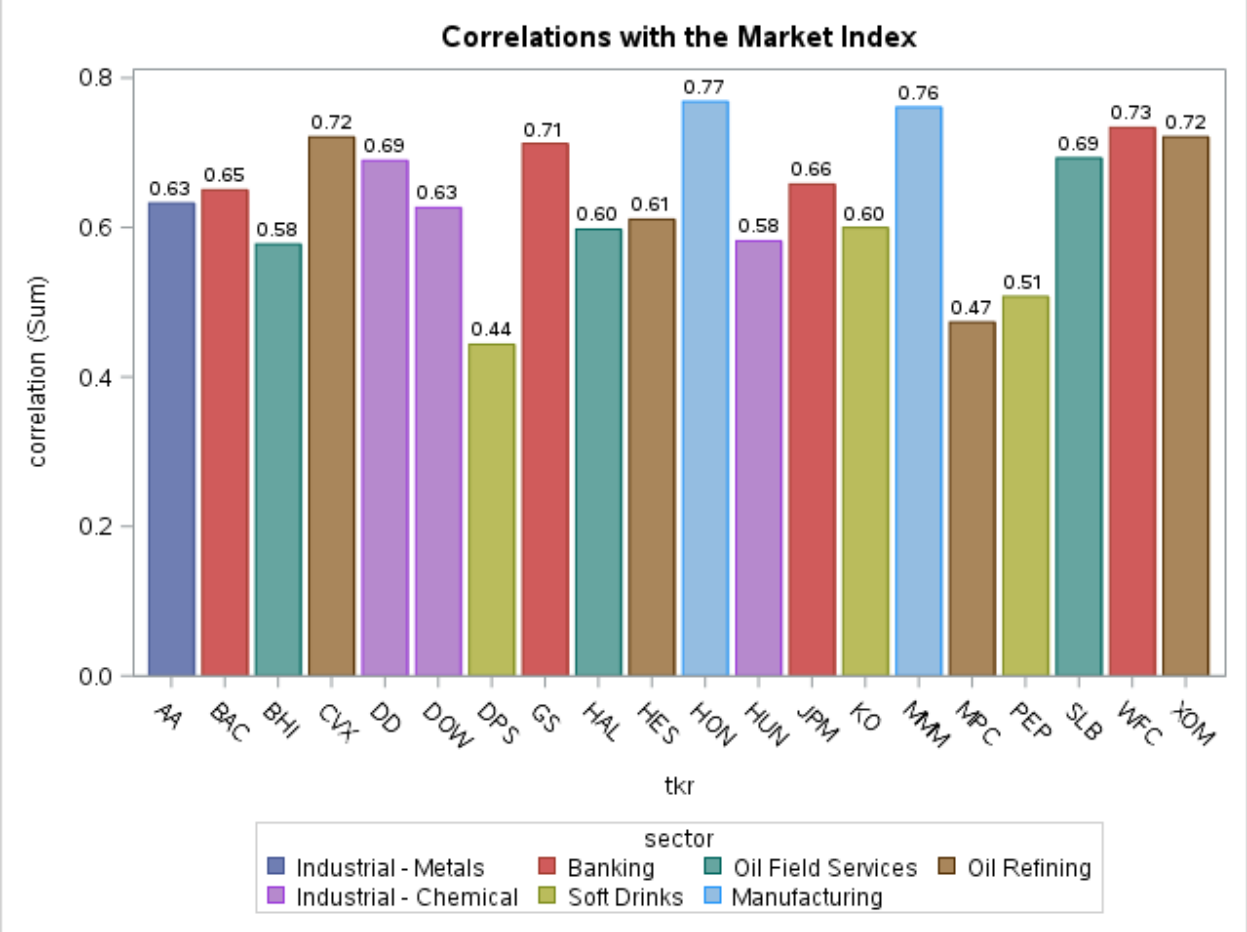
Correlation between Log-Return and Market Index—

After performing the PearsonCorr procedure the results show a wide dataset, meaning the variables in this vector are in a left to right orientation. We'll need to transpose this dataset into the long format since it's required for the graphics being created later; it's also not best practice to examine data in the wide format.

Once we transposed the data, we created a separate dataset with each ticker (tkr) and their individual sector. We have sorted the transposed to long correlation dataset, as well as the new ticker and sector data by the ticker, then merged each. Afterwards, we were able to print a table of the Pearson Correlations of log-return to the Vanguard Index, to include each ticker's individual sector:

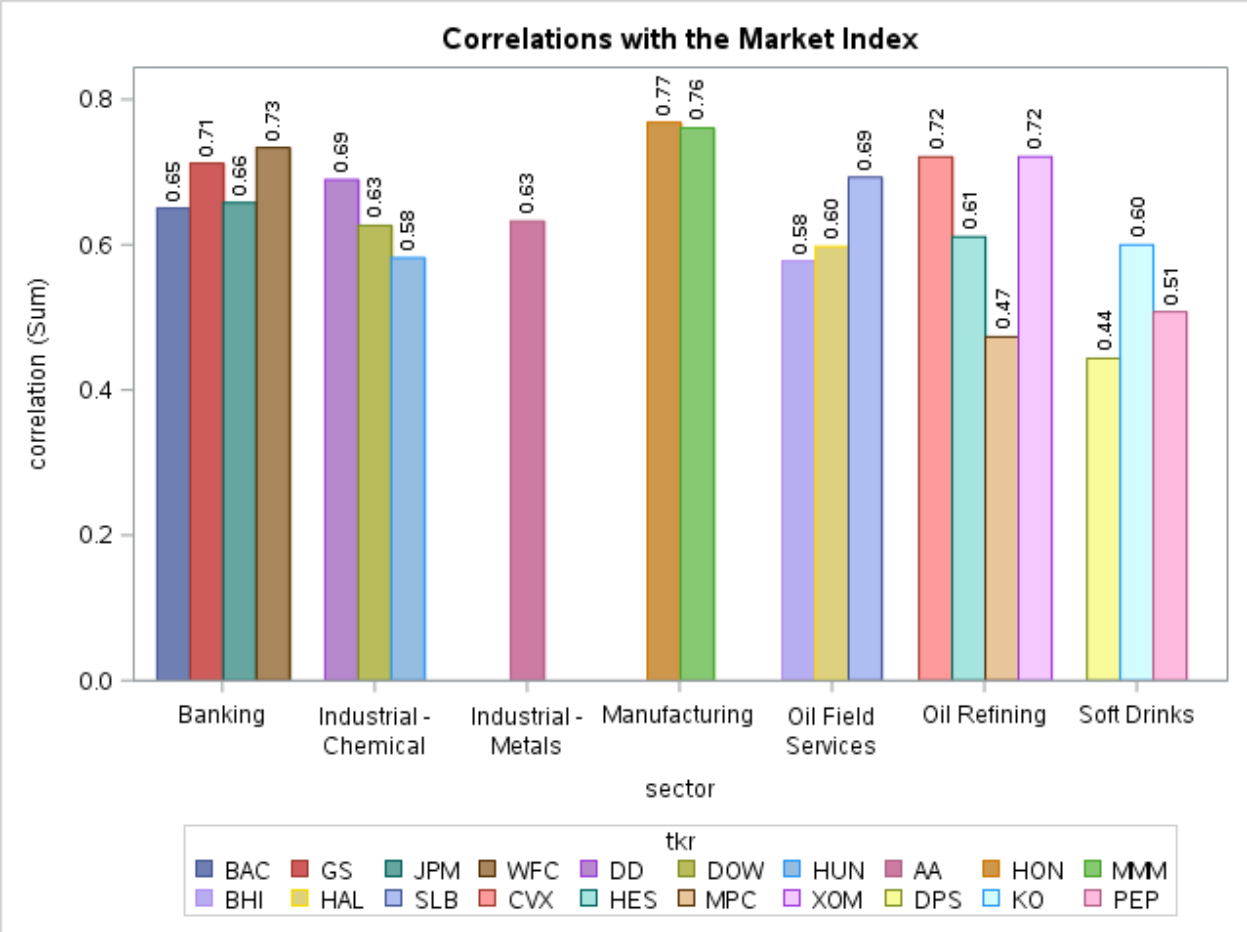
Obs	correlation	tkr	sector
1	0.63241	AA	Industrial - Metals
2	0.65019	BAC	Banking
3	0.57750	BHI	Oil Field Services
4	0.72090	CVX	Oil Refining
5	0.68952	DD	Industrial - Chemical
6	0.62645	DOW	Industrial - Chemical
7	0.44350	DPS	Soft Drinks
8	0.71216	GS	Banking
9	0.59750	HAL	Oil Field Services
10	0.61080	HES	Oil Refining
11	0.76838	HON	Manufacturing
12	0.58194	HUN	Industrial - Chemical
13	0.65785	JPM	Banking
14	0.59980	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

Next, we examine the above table by creating a grouped, colored bar plot of the correlation on the Y-Axis, but group the *tkr* by sector on the X-Axis:



The Banking sector shows similar correlation, as well as the Manufacturing sector. The rest do not show similar correlation.

This graphic is still not correct though, because we want the tickers grouped and color coded by sector; therefore, we edit the graph by assigning the ticker labels under the X-Axis, and the sector labels in the graph legend:

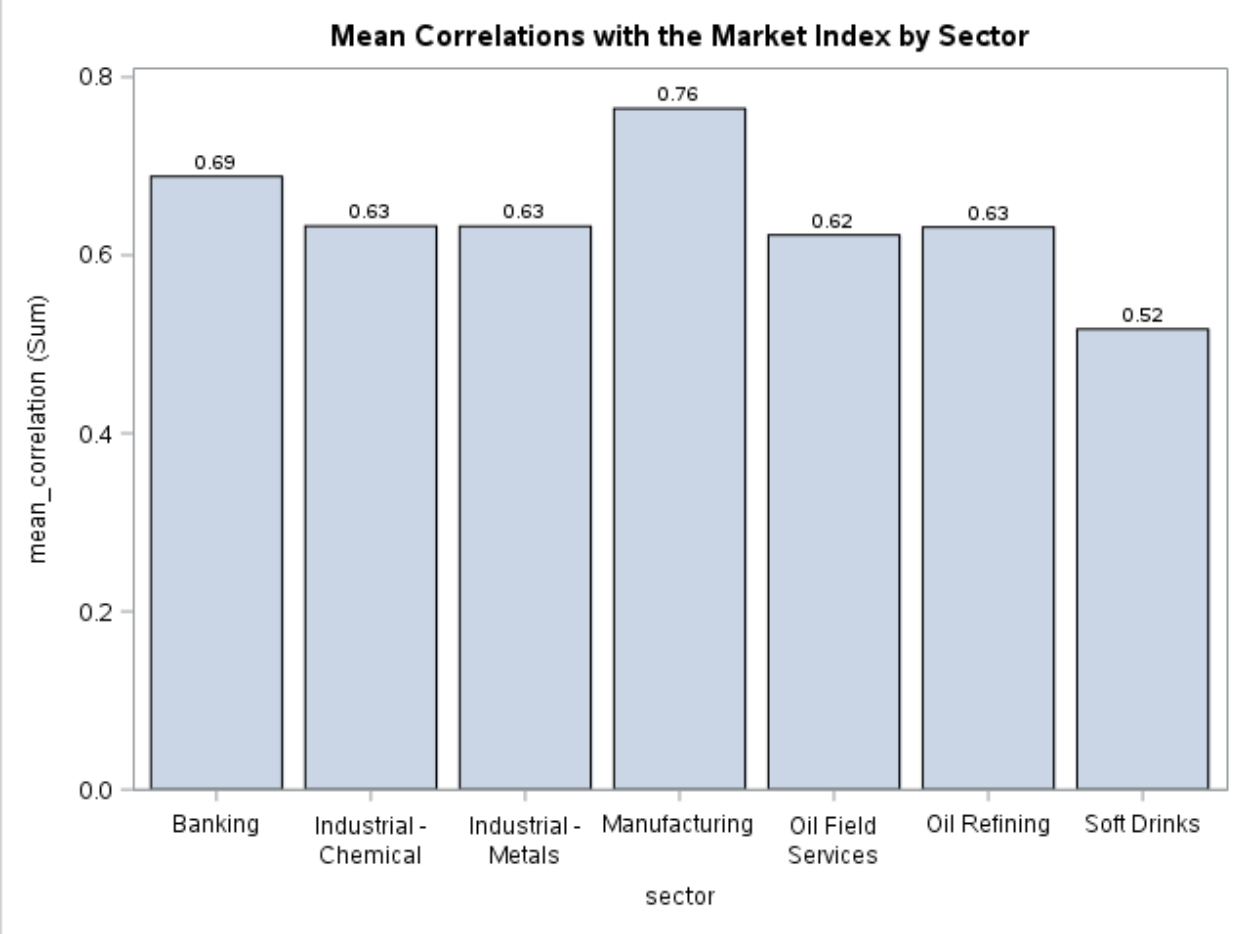


Next, we examine the mean correlation by sector:

Obs	sector	_TYPE_	_FREQ_	mean_correlation
1	Banking	1	4	0.68844
2	Industrial - Chemical	1	3	0.63264
3	Industrial - Metals	1	1	0.63241
4	Manufacturing	1	2	0.76461
5	Oil Field Services	1	3	0.62262
6	Oil Refining	1	4	0.63148
7	Soft Drinks	1	3	0.51694

We notice that the sectors with heavy price regulation all have similar correlation to a large market identifier.

We'll now examine the above mean correlation table by creating a bar chart:



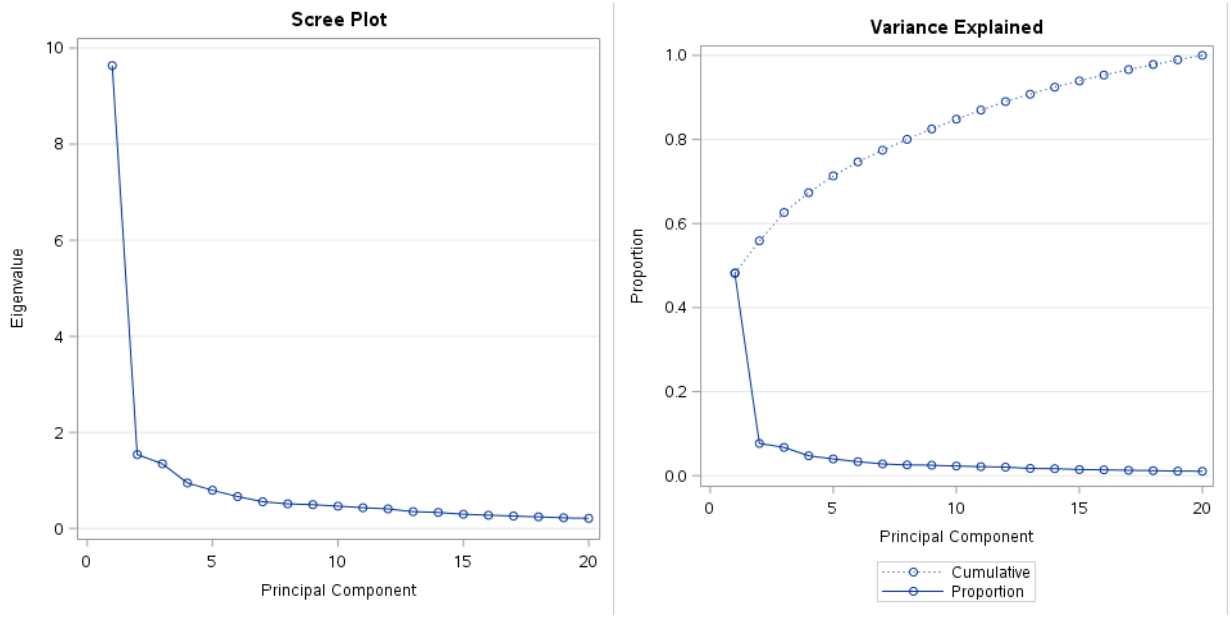
We see that the Manufacturing sector shows the highest correlation of all the sectors; we also notice via this mean correlation with market index Vanguard graph, that all other sectors aside from Soft Drinks, show similar correlation.

Principal Components—

We will create a new dataset to exclude the Market Index as we calculate our principal component because we don't want the dependent variable included in this dataset. We are able to utilize a trick when we call the *set directive*: passing in the *keep* statement, as well as *return_:* — the colon is used as a wildcard to match all previously calculated log-return tickers. *Princomp* will be used to calculate the *eigenvectors*, as well as produce a screen plot. Once complete, we should be able to come up with criteria for which principal components we can take forward into our regression modeling.

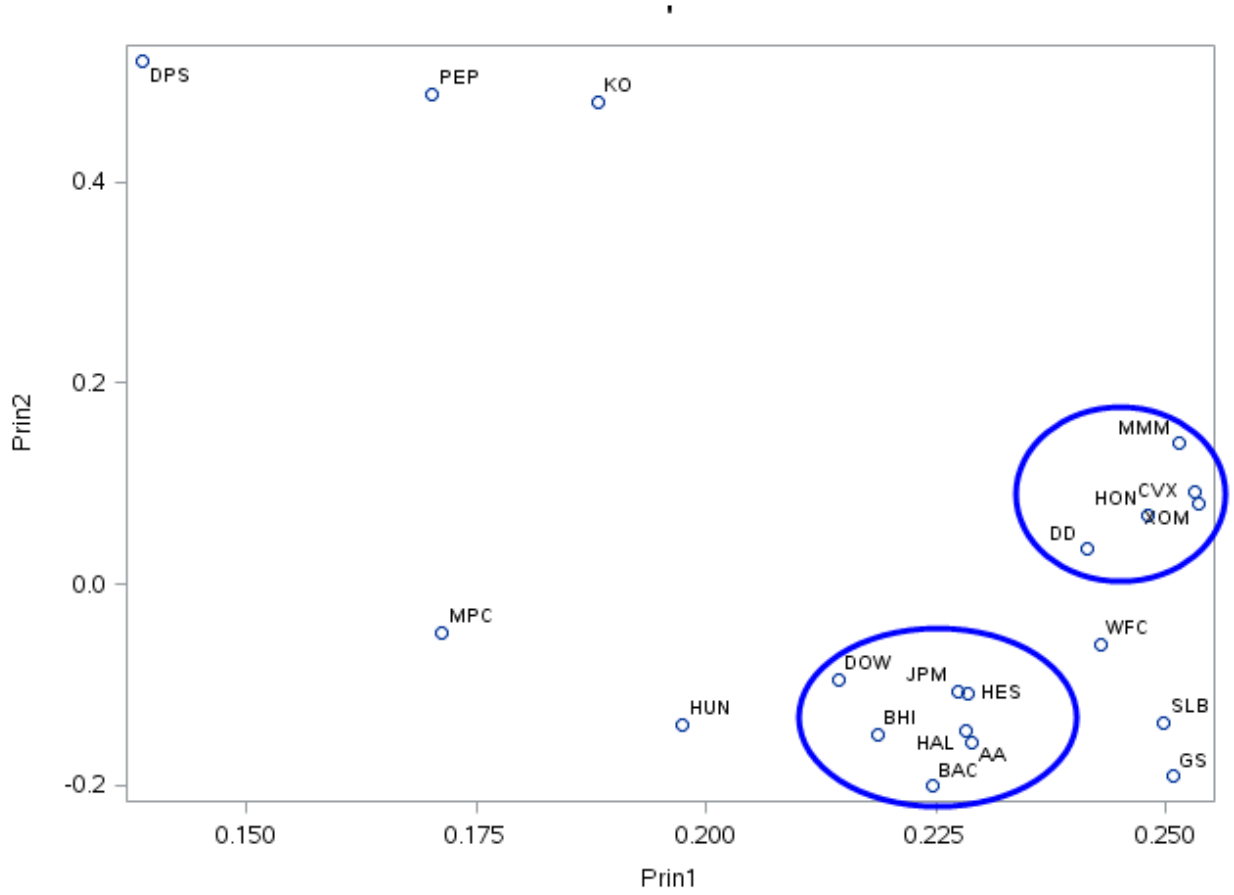
Eigenvalues of the Correlation Matrix				
	Eigenvalue	Difference	Proportion	Cumulative
1	9.63645075	8.09792128	0.4818	0.4818
2	1.53852947	0.19109235	0.0769	0.5587
3	1.34743712	0.39975791	0.0674	0.6261
4	0.94767921	0.15217268	0.0474	0.6735
5	0.79550653	0.12909860	0.0398	0.7133
6	0.66640793	0.10798740	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

Next, we'll look at a scree plot that's overlay with the cumulative values from the eigenvalue table above:



The scree plot gives a quick and simple mechanism for evaluating the set of calculated principal components and how each of them contribute to the overall variability within the dataset.

Next, we examine the first two principal component vectors to see if we notice any relationship within the data:



In observing this graph, we see two clusters (outlined in blue). After examining the tickers within each cluster to see if they belong to the same sector, we find that they do not belong to the same sector; consequently, there's no reason at the moment as to why these clusters are occurring except that their log-returns are relatively close in value.

Combine PCA Dataset into Train/Test—

Next, we'll split the original dataset into training and test groups; 70% of the data being selected will be at random within the training group. We'll use the *merge* feature to merge the original log-return calculations with the principal components. Rows flagged as part of the training set will be assigned *response_VV* to a new variable, *train_response*.

Regression Model with All Stocks—

First, we fit a regression model using all of the raw predictor variables. (ANOVA, Model Performance, and Parameter Estimates)

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			

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

R-Square and Adj. R-Square are high, but this is expected. AVOVA is showing that the model is statistically significant.

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

From reviewing these estimates, we detect the presence of multicollinearity. We have a significant F Value, and almost all independent variables have a non-significant t-test.

Next, we examine the Variance Inflation Factor (VIF) of the k^{th} predictor:

$$VIF_k = \frac{1}{1-R^2k}$$ (Note: R^2_k = R-Square value obtained by regressing the k^{th} predictor on remaining predictors.)

VIF_k represents how much the variance of the estimated regression coefficient β_k is inflated by the existence of correlation among the predictor variables in the model. If $VIF_k > 10$, multicollinearity is considered to be high. Reviewing the estimates table, none of the predictors have a VIF higher than 4; however, many of the predictors are close to 4 – in general, it's best for the VIF_k value to be close to 1.0.

Regression Model with Principal Components—

First, we fit a regression model using the eight selected principal components. (ANOVA, Model Performance, and Parameter Estimates)

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			

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

The R-Square and Adj. R-Square values have decreased slightly, but our F Value has more than doubled over the last model.

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

The first three independent variables show significance. This is an improvement over the last model. Independent variables four – eight are not significant, but the VIF_k of each predictor variable are all close to 1 – which is the goal for VIF values.

Conclusions:

Between the two models, the first model with the predictor variables had significant signs of multicollinearity. The second model with the selected principal components had less indication of multicollinearity, but had a significant rise in the F Value; however, both model's F Values were very high.

To further compare the models, it's best to compute the MSE (mean square error) and the MAE (mean absolute error). MSE assesses the quality of an estimator or set of predictions in terms of its variation and degree of bias. MAE is an average of the absolute error within the model.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2 \qquad MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{Y}_i - Y_i|$$

Obs	train	MSE_1	MAE_1	MSE_2	MAE_2
1	0	.000009306	.002144904	.000009677	.002179249
2	1	.000005994	.001902032	.000006410	.001975239

After computing the MSE and MAE, the difference between the models are small. Overall, the second model is best for the following reasons:

- Significantly fewer parameters—makes the model simpler to interpret
- Less presence of multicollinearity with low VIF_k on each parameter
- Better goodness-of-fit with higher F Value and almost the same R-Square and Adj. R-Square values

Since the first model showing multicollinearity, using the PCA as a preprocessor for regression gives the ability to address multicollinearity by reducing the amount of parameters used in the model.

Code:

```
libname mydata '/scs/wtm926/' access=readonly;

data temp;
        set mydata.stock_portfolio_data;
run;

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

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));

response_VV=log(VV/lag1(VV));
run;

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

ods trace on;

ods output PearsonCorr=portfolio_correlations;

proc corr data=temp;
        var return_:;
        with response_VV;
run;
quit;

ods trace off;

proc print data=portfolio_correlations;
run;
quit;
```

```
data wide_correlations;
	set portfolio_correlations (keep=return_:);
run;

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;

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;

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;
run;
quit;

ods graphics off;

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;
run;
quit;

ods graphics off;

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 on;

title 'Mean Correlations with the Market Index by Sector - SGPLOT COMPUTES MEANS';
proc sgplot data=long_correlations;
        format correlation 3.2;
        vbar sector / response=correlation stat=mean datalabel;
run;
quit;

ods graphics off;

title '';

ods graphics off;

data return_data;
        set temp (keep= return_:);
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);
run;
quit;

ods graphics off;

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

```
run;

proc print data=eigenvectors(where=(_TYPE_='SCORE'));
run;

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;

ods graphics on;

proc sgplot data=long_pca;
	scatter x=Prin1 y=Prin2 / datalabel=tkr;
run;
quit;

ods graphics off;

data cv_data;
	merge pca_output temp(keep=response_VV);
	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;

proc print data=temp(keep=response_VV obs=10);
run;
quit;

ods graphics on;

proc reg data=cv_data;
	model train_response = return_: / vif;
	output out=model1_output predicted=Yhat;
run;
quit;

ods graphics off;

ods graphics on;

proc reg data=cv_data;
	model train_response = prin1-prin8 / vif;
	output out=model2_output predicted=Yhat;
run;
quit;

ods graphics off;

proc print data=model1_output(obs=10);
run;

data model1_output;
	set model1_output;
	square_error = (response_VV - Yhat)**2;
	absolute_error = abs(response_VV - Yhat);
run;

proc means data=model1_output nway noprint;
	class train;
	var square_error absolute_error;
	output out=model1_error
		mean(square_error)=MSE_1
		mean(absolute_error)=MAE_1;
run;
quit;

proc print data=model1_error;
run;
```

```
data model2_output;
	set model2_output;
	square_error = (response_VV - Yhat)**2;
	absolute_error = abs(response_VV - Yhat);
run;

proc means data=model2_output nway noprint;
	class train;
	var square_error absolute_error;
	output out=model2_error
		mean(square_error)=MSE_2
		mean(absolute_error)=MAE_2;
run;
quit;

proc print data=model2_error;
run;

data error_table;
	merge model1_error(drop= _TYPE_ _FREQ_) model2_error(drop= _TYPE_ _FREQ_);
	by train;
run;

proc print data=error_table;
run;
```