Assignment #6
Scott M. Morgan

Introduction:

The purpose of this assignment is to use principal components analysis as a method of dimension reduction and as a remedial measure for multicollinearity in Ordinary Least Squares regression. To accomplish this objective, I use the stock portfolio data set provided by the course instructor and SAS Studio to construct regression models using individual stocks to explain performance of the market index, which is represented by the Vanguard Large-Cap ETF. The data set consists of the daily closing prices of 20 publicly traded companies between the 2-year period of January 3, 2012 to December to 31, 2013. The independent variables will be stock prices (or some derivative of) and the dependent variable will be the return of the market index. Understanding what factors impact market performance is of great interest to investors not only as they dynamically position portfolios throughout market cycles, but also when they communicate with clients. Quantitative tools such as principal component analysis help decision makers see through the noise of large data sets and provide them insight into less obvious relationships. I expect this assignment will be an excellent application of this technique as financial market analysis can easily become overwhelming given the volume, variety and velocity of the data involved.

Results:

In the subsequent sections, we prepare the stock portfolio data for analysis, visualize and examine correlations with the market, perform principal component analysis and, finally, construct several regression models to explain market returns. A key element of constructing the regression models will be identifying and remedying multicollinearity if present.

Data Preparation. In this section, we perform our initial data preparation. The raw data consists of daily closing stocks prices for 20 publicly traded U.S. equities and the Vanguard Large-Cap ETF (VV). The VV is used as a proxy for the broad U.S. stock market as it currently holds 614 individual stocks and exhibits a trailing 5-year tracking error of 0.08% to its benchmark, the CRSP US Large Cap Index, according to Morningstar as of March 31, 2017. The CRSP US Large Cap Index includes U.S. companies that comprise the top 85% of investable market capitalization. It includes both mid and mega capitalization companies and is what I believe to be a relatively suitable proxy for the investable universe of the U.S. stock market. Tracking error, or active risk, is the annualized standard deviation of daily return differences between the return performance of the fund and the return performance of its underlying index. As ETFs are intended to mirror non-investable market indexes, we want this number as small as possible. For context, a trailing 5-year tracking error of 0.08% is considered exceptional so we are comfortable with the VV as a market proxy.

We calculate the return on the stocks as follows:

$$r_i = \frac{p_i - p_j}{p_j}$$

Where *r* is the return at time *i* and *p* is the closing price of the security at time *i* and *j*. We use return instead of price to measure all variables in a comparable metric. We also compute the natural log of returns. Log returns are preferred for financial analysis for reasons of normalization and they are usually not auto-correlated while prices can be (Aldridge, 2013). Analysis might also benefit from log transformations if there are a wide range of x values; which is the case with our sample of 20 different stocks and their prices over the 2-year period.

Correlations of Security Log Return to Market Index Log Return. Following a series of structural transformations to the data set, we arrive at Table 1 which provides the correlations of the individual equities to the market index over the specified period, sorted in alphabetical order. We also include the sector in which the company operates; the idea being that the stock price of competing firms within the same sector tend to behave similarly over a market cycle. A U.S. stock market index is comprised of any number of sectors depending on the classification schema (i.e. GICS, ICB, TRBC, SIC, Russell, etc.) that are intended to reflect, ceteris paribus, the current conditions of the economy. As sector performance thematically drives the performance of the overall market, it would be advantageous to understand if and how stocks in the same sector are correlated with each other as well as how the sector itself is correlated with the market index.

Table 1: Correlations of Security Log Return to Market Index Log Return

Obs.	Correlation	Ticker	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

Table 1 in its current format is difficult to digest and draw conclusions from. When working with a large amount of predictor variables, it can be helpful to use visualizations instead of tables. Exhibit 1 below provides our first visualization in the form of a bar chart that is sorted by company ticker and color-coded by sector. While much more appealing to the eye relative to Table 1, it is still difficult to discern any meaningful insight.

Exhibit 1: Correlations with the Market Index

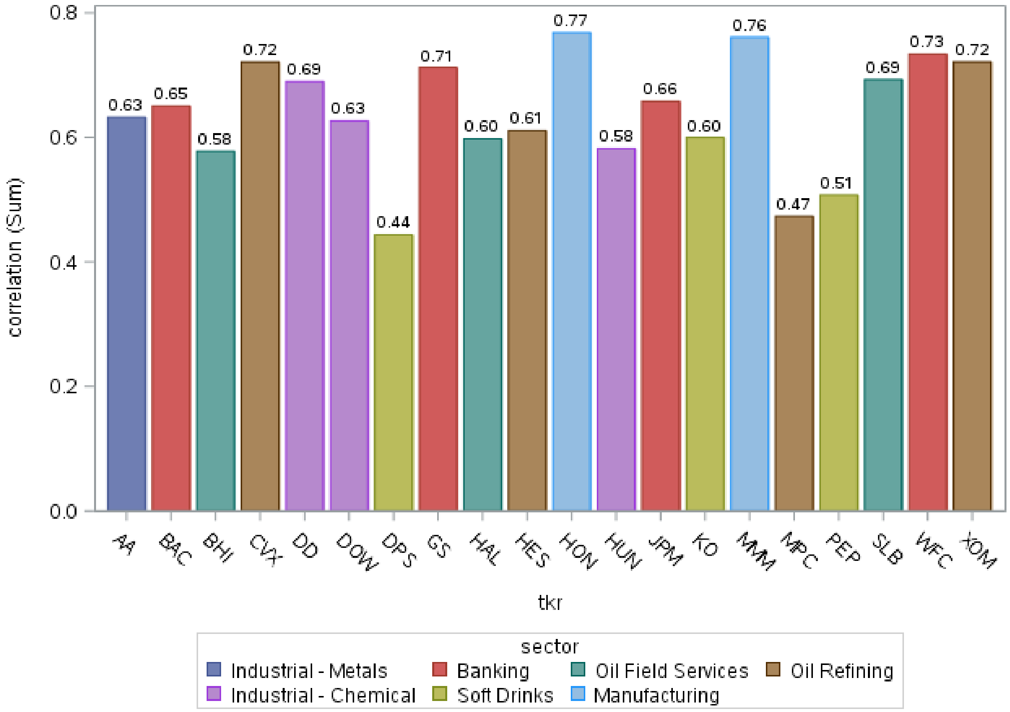
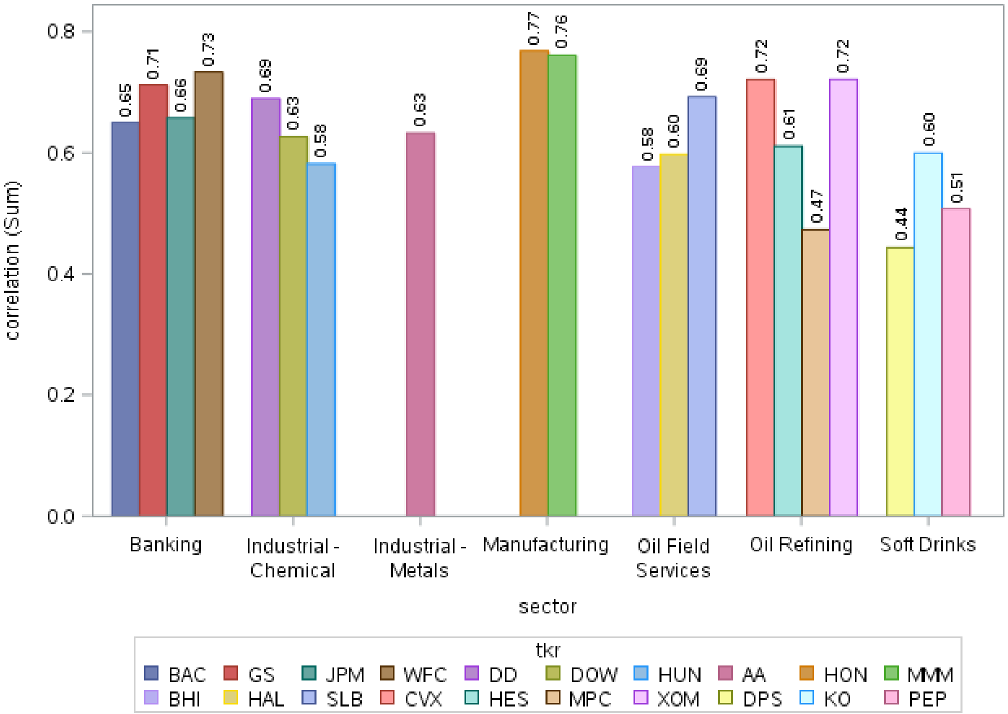


Exhibit 2 attempts to improve upon our initial graph by grouping the stocks together by sector. Stocks within Banking and Manufacturing have relatively similar intra-sector correlations to the market index. Conversely, stocks within the Industrial-Chemical, Oil Field Services, Oil Refining and Soft Drinks sectors have wider ranges of correlations to the market index. There is only one stock in the Industrials – Metals sector so comparisons with related stocks is not possible. This bifurcation in correlation patterns could be for many reasons. We do not fully understand how the current portfolio of 20 stocks was constructed in terms of characteristics. For example, the portfolio could be a mix between large and mega cap names, the latter being less volatile and likely more positively correlated with the broad market. Also, while sectors like Oil Field Services and Oil Refining tend to exhibit more cyclicality; we cannot be sure why the correlations of the underlying equities in the sectors differ so much without additional information on the securities themselves or the screening technique used. Furthermore, we are attempting to explain total market performance by using a subset of sectors. This is problematic as other sectors not included in the sample, such as Heath Care and Technology, likely have considerable explanatory power as they represent over 30% of the market index according to the Vanguard website. We are left with the assumption that the stocks, and therefore the sectors, were cherry picked (i.e. selection bias) based on some unknown criteria. This could bring into question the validity of any predictive model. A superior approach would be to use sector focused ETFs of all underlying sectors as they represent large unbiased, baskets of stocks. While this is beyond the scope of this exercise, I have included a recommended list of ETFs in the appendix for future assignments.

Exhibit 2: Correlations with the Market Index by Sector



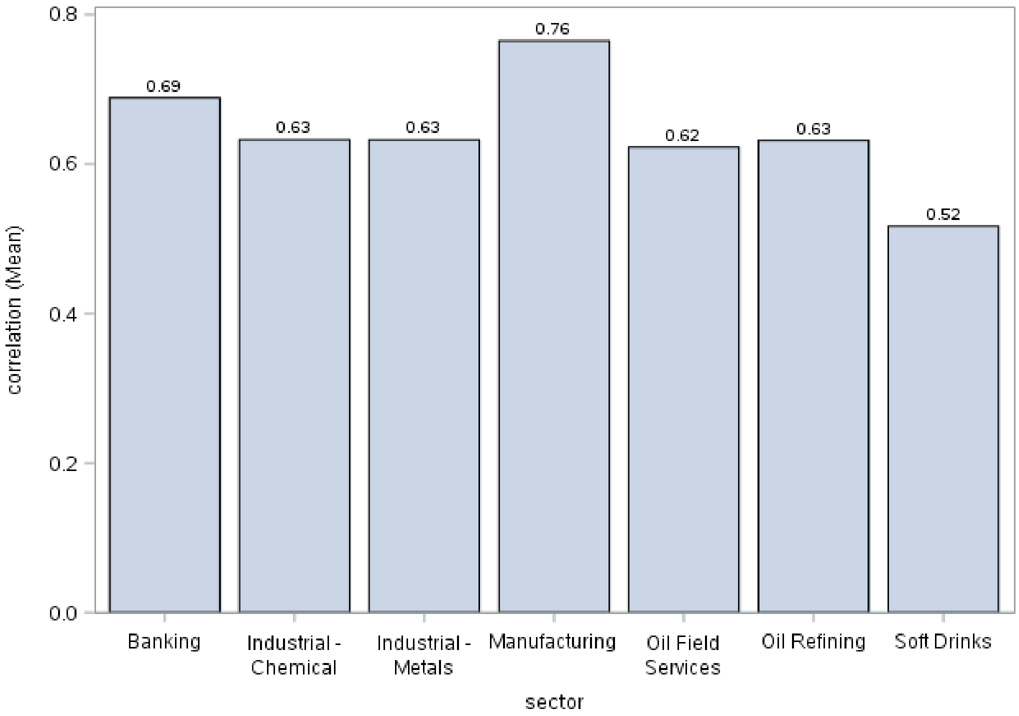
While Exhibit 2 perhaps introduces more questions than answers, we continue forward with our analysis and attempt to reduce the noise caused by listing the individual securities. We do this by grouping at the sector level and calculating the arithmetic mean of the underlying equities (Table 2). This method assumes an equal weighting of the securities in the portfolio.

Table 2: Correlations with the Market Index by Sector

Obs.	Sector	Type	Frequency	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

Exhibit 3 below is the visual representation of Table 2. Examining the graph, the disjointed individual correlations appear to have been successfully. Manufacturing and Banking, respectively, have the highest correlation to the market index. This makes sense intuitively as these two sectors are more sensitive than other parts of the market to changing demand for production/labor and fluctuations in interest rates as well as consumer financial health throughout economic cycles. The commodity and raw material based sectors (Industrial – Chemical, Industrial - Metals, Oil Field Services and Oil Refining) all have very similar correlations and are also considered economically sensitive, albeit to a lesser extent. The Soft Drinks sector likely has the lowest correlation with the market index as it is a subset of Consumer Staples, which is generally less cyclical as these are goods people are unable or unwilling to cut regardless of financial situation. In the following section, we begin our work with principle components.

Exhibit 3: Mean Correlations with the Market Index by Sector



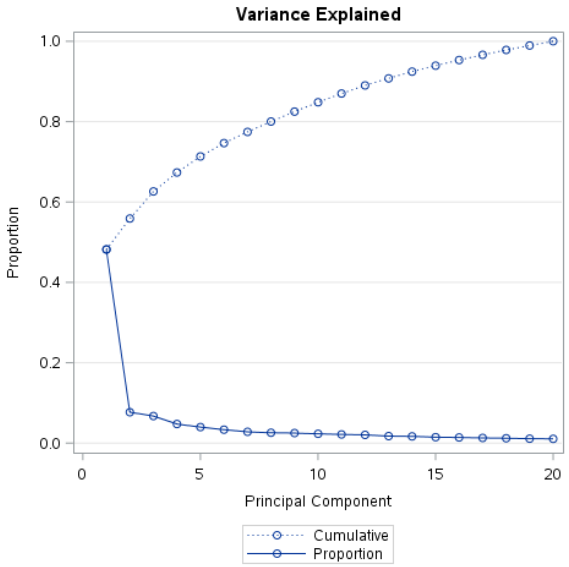
Principal Components Analysis. We first create a data set that consists solely of the log return information of the independent variables. This is done by using the (keep=return_:) function in SAS. The response variable is not kept as the log transformed variable contains the "response_" prefix. We then use the SAS function princomp to calculate the Eigenvectors, or latent roots, as well as generate a scree plot. Table 3 provides the Eigenvalues of the correlation matrix. There is 1 principal component for every variable in the data set. For interpretative purposes, starting with the initial observation, the first principal component is equivalent to about 9.6 of the original variables.

Table 3: Eigenvalues of the Correlation Matrix

Obs.	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

Exhibit 4 below provides the scree plot by principal component as well as a line illustrating the cumulative variance explained. Scree plots are useful for finding an upper bound for the number of components that should be retained for modeling. It appears that the variance explained drops considerably after the second principal component.

Exhibit 4: Scree Plot with Variance Explained



From both Table 3 and Exhibit 4, we decide which principal components to include for regression modeling. There are several options for deciding how many principal components to use:

- **Option 1)** Eigenvalues greater than 1.0;
- **Option 2)** Scree test (i.e. "elbow");
- **Option 3)** Percent of total variance (~60% in this case);
- **Option 4)** A priori specification;
- **Option 5)** Parsimony.

For this exercise, the recommendation is to include 3 principal components based on Options 1, and 3; the first 3 Eigenvalues are greater than 1.0 and the cumulative variance explained is slightly greater than 60% at 3 Eigenvalues. We will also plot the first 2 Eigenvectors from the principal components analysis in Exhibit 5 below. The graph itself is relatively easy to interpret and display graphically. In terms of groupings, there are two distinct clusters and single point separated from both clusters. The Soft Drinks sector has a distinct grouping in the upper left portion of the graph. The more cyclical sectors are grouped together, which is to be expected, in the lower region of the chart. The one surprising point is the isolated location of Marathon Petroleum Corporation (MPC), which is part of the Oil Refining sector. As a side note, it would be helpful in future exercises to color the data points by sector in a scatterplot.

Exhibit 5: Principal Component 1 and Principal Component 2

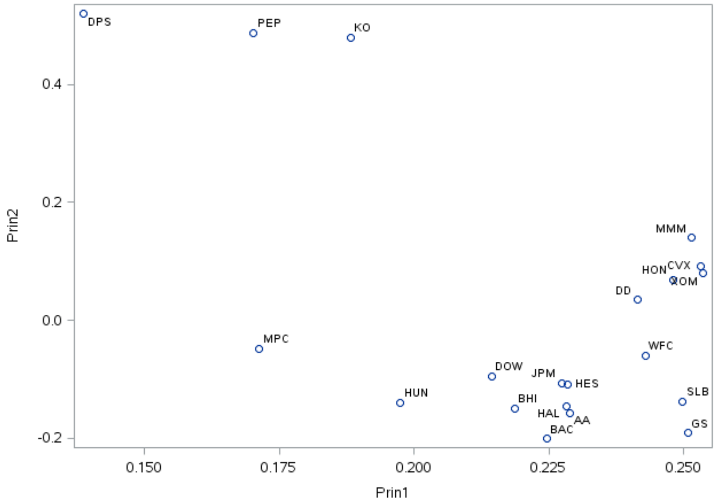


Table 4 below delineates all variables in the data set and how they are fitting with the first 3 principal components. The figures represent the Eigenvectors or loadings and we are focused on understanding the relative magnitude of each variable by principal component. For the first principal component, we see that the values for DPS, KO, MPC and PEP are materially different from the other variables; this is supported by our observations from Exhibit 5 above. We would interpret that all the other variables simultaneously are contributing to the interpretation of the first principal component. That commonality, as we have discussed, is likely some factor involving sectors with increased sensitivity to economic cycles. With the second principal component, the companies in the Soft Drinks sector (DPS, PEP, and KO) all have relatively high Eigenvectors. We posit the second principal component is definable by companies that are counter-cyclical. Lastly, MPC is an interesting case as it clearly stood out in Exhibit 5 yet it's Eigenvector does not become notable until the fifth principal component. I speculate the Eigenvector is much larger than the others for this principal component due to some idiosyncratic, or company specific, characteristic. MPC is much smaller in terms of market capitalization compared to XOM and CVX, and the company profiles on Yahoo!Finance suggests that XOM, CVX and HES are all integrated oil and gas firms where as MPC is involved specifically in refining and marketing. Therefore, principal component 5 might be related to some market factor that's specific to oil refining companies; sensitivity to input prices and gasoline inventories are two examples.

Table 4: Abbreviated Eigenvectors

Obs.	Prin1	Prin2	Prin3
return_AA	0.228948	-0.157413	-0.035471
return_BAC	0.22464	-0.200812	0.340968
return_BHI	0.218616	-0.150172	-0.43866
return_CVX	0.253203	0.091157	-0.167116
return_DD	0.241403	0.034337	0.122273
return_DOW	0.214303	-0.094422	0.172226
return_DPS	0.138615	0.520404	-0.015629
return_GS	0.250754	-0.19111	0.190148
return_HAL	0.228119	-0.146007	-0.401851
return_HES	0.228429	-0.108316	-0.224758
return_HON	0.247959	0.068924	0.085061
return_HUN	0.197471	-0.14011	0.182943
return_JPM	0.227353	-0.106563	0.347821
return_KO	0.1882	0.47871	0.030244
return_MMM	0.251487	0.139607	0.011853
return_MPC	0.171144	-0.048239	-0.003188
return_PEP	0.170206	0.487599	0.019364
return_SLB	0.249733	-0.137574	-0.32663
return_WFC	0.24282	-0.060107	0.284424
return_XOM	0.253542	0.080845	-0.135687

Regression Model 1: Raw Predictors. The next step towards using principal components in regression modeling is to use SAS to split the data set into training and test sets and then fit a regression model using all the raw predictor variables and VV as the response variable. Using the SAS procedure for correlation and the train data set, we produce an analysis of variance table. The output is presented in Table 5 below.

Table 5: Model 1 Parameter Estimates

Source	DF	Sum of Squares	Mean Square	F Value	Pr > |t|
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-Squared**	0.8983
Dependent Mean	0.00061635	**Adj R-Sq**	0.8919
Coeff Var	410.18453		

Of the 502 observations in the train sample, the number of observations with missing values is 164, which is about 33%. This makes sense given the 70/30 splitting of the data. The F Value is statistically significant but we reserve commentary on the metric until viewed in the context of another model. Both R-squared and Adjusted R-Squared are high. We next shift our attention to the automatically generated ODS output (not provided) from SAS to assess the goodness-of-fit for this model. The scatterplot for residual values is slightly concentrated toward the center of the plot but generally random. The Q-Q plot and histogram of the residuals suggests a relatively normal distribution. Cook's Distance (Cooks' D), which quantifies influential data points, illustrates a handful of possible outliers. The left pane of the Fit-Mean Residual plot is taller than the right, indicating that the independent variables account for a greater portion of the variation in the model.

Using the SAS regression procedure, we next generate the following parameter estimates for Model 1 (Table 6):

Table 6: Model 1 Parameter Estimates

Variable	DF	Parameter	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

A little over half of the coefficients are statistically significant. To check for multicollinearity, we use SAS to calculate the variance inflation factors (VIF) to measure how much variance of the coefficients is "inflated" by the existence of correlation among the predictor variables in the model. For the sake of interpretability, using return_AA as an example, the VIF tells us that the variance of the log return of Alcoa Corporation (AA) is inflated by a factor of 2.1149 because it is highly correlated with at least one of the other predictors in the model. As the VIFs in this initial model generally range between 1 and 3, we conclude that the predictors are moderately correlated. The guidelines we used for interpreting the VIF are from Minitab.com and are presented in Table 7 below for reference. VIFs larger than 10 imply serious problems with multicollinearity (Montegomery, Peck & Vinning, 2012).

Table 7: Guidelines to Interpret VIF

VIF	Status of Predictors
VIF = 1	Not correlated
1 < VIF < 5	Moderately Correlated
VIF > 5 to 10	Highly Correlated

Lastly, the MSE and MAE are very low for the training and testing samples (Table 8). For reference, the MSE (Mean Squared Error) is a measure of how close a fitted line is to data points. It is the average of the squares of the difference between the actual observations and predicted data points. The lower the MSE, the better the model is fit to the data. The MAE (Mean Absolute Error) measures the average magnitude of the errors in a group of predictions. It is the average of the absolute difference between predicted and observed points where all have equal weight. The lower the MAE, the better.

Table 8: Model 1 Additional Fit Statistics

Obs	Train	MSE_1	MSE_2
1	0	0.0000009306	0.002144904
2	1	0.000005994	0.001902032

Regression Model 2: Principal Components. We now fit a regression model using the first 3 principal components and VV as the response variable. Given the inclusion of the principal components in the regression model we would expect the VIF scores to drop notably. Using the SAS procedure for correlation and the train data set, we produce an analysis of variance table. The output is presented in Table 9 below.

Table 9: Model 2 Parameter Estimates

Source	DF	Sum of Squares	Mean Square	F Value	Pr > |t|
Model	3	0.01770	0.00590	882.42	<.0001
Error	334	0.00223	0.00000668		
Corrected Total	337	0.01993			

Root MSE	0.00259	R-Squared	0.8880
Dependent Mean	0.00061635	Adj R-Sq	0.8870
Coeff Var	419.47762		

Of the 502 observations in the train sample, the number of observations with missing values is 164, identical to Model 1. The F Value is statistically significant and much larger than Model 1. Both R-squared and Adjusted R-Squared are high but marginally lower than Model 1. We next shift our attention to the automatically generated ODS output (not provided) from SAS to assess the goodness-of-fit for this model. The scatterplot for residual values is still slightly concentrated toward the center of the plot but again generally random. The Q-Q plot and histogram of the residuals suggests a relatively normal distribution, however there now appears to be a few additional possible outliers compared to Model 1. This is corroborated by Cook's D. The left pane of the Fit-Mean Residual plot is again taller than the right, indicating that the independent variables account for a greater portion of the variation in the model.

Using the SAS regression procedure, we next generate the following parameter estimates for Model 2 (Table 10):

Table 10: Model 2 Parameter Estimates

Variable	DF	Parameter	Standard Error	t Value	Pr > |t|	Variance Inflation
Intercept	1	0.00079501	0.00014088	5.64	<.0001	0
Prin1	1	0.00231	0.00004547	50.72	<.0001	1.00252
Prin2	1	0.00029846	0.00011466	2.60	0.0097	1.00099
Prin3	1	0.00071709	0.00012382	5.79	<.0001	1.00332

All coefficients are statistically significant. Additionally, the VIFs of the coefficients are about 1.0 This model does not appear to have a multicollinearity problem. The MSE and MAE are both small and we compare the metrics between the 2 models in Table 11 below. The difference between the models in terms of these metrics is minimal.

Table 11: MSE and MAE Comparisons

Obs	Train	MSE_1	MAE_1	MSE_2	MAE_2
1	0	0.0000009306	0.002144904	0.000009603	0.002195650
2	1	0.000005994	0.001902032	0.000006605	0.001998245

Reflections / Conclusions:

The conclusion for Assignment #6 is that Model 2 is superior in terms of fit and interpretability. Model 2 tells us that economically sensitive sectors (principal component 1) have greater impact on explaining market returns than non-cyclical sectors (principal component 2) and oil refining-related companies (principal component 3). Model 1 had moderate problems with multicollinearity and was difficult to interpret given the number of coefficients.

Certain elements of this study are problematic and worth mentioning. First, the sample data uses only 2 years of historical information. From personal experience, industry practice dictates 3, 5 and 10 year windows for analysis while more academically orientated work generally requires as much historical data as possible though it depends on the scope of the study. The use of rolling correlation windows should also be considered for visualization purposes as this study uses correlations over a single point in time. This is a dangerous assumption as correlations change over economic cycles, especially during systematic market moves up and down. Also, as discussed earlier, an alternative approach to constructing the market proxy should be advocated. Sector based ETFs would provide a larger sample given they invest in hundreds of individual securities and would likely provide a more accurate representation of the U.S. stock market. This would also alleviate the concern of cherry picking stocks.

Reference(s):

Aldridge, I. (2013). *High-frequency trading: a practical guide to algorithmic strategies and trading systems*. Hoboken, NJ: Wiley.

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

Appendix:

List of Alternative ETFs for Market Index Proxy

Name	Morningstar Category	Ticker
Consumer Discret Sel Sect SPDR® ETF	US Fund Consumer Cyclical	XLY
Consumer Staples Select Sector SPDR® ETF	US Fund Consumer Defensive	XLP
Energy Select Sector SPDR® ETF	US Fund Equity Energy	XLE
Financial Select Sector SPDR® ETF	US Fund Financial	XLF
Health Care Select Sector SPDR® ETF	US Fund Health	XLV
Industrial Select Sector SPDR® ETF	US Fund Industrials	XLI
Materials Select Sector SPDR® ETF	US Fund Natural Resources	XLB
Real Estate Select Sector SPDR®	US Fund Real Estate	XLRE
Technology Select Sector SPDR® ETF	US Fund Technology	XLK
Utilities Select Sector SPDR® ETF	US Fund Utilities	XLU

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;

/* PART 1*/

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));

        *return_VV  = log(VV/lag1(VV));

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

run;


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



/* PART 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;



/* PART 3*/



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;

* SAS has two 'standard' data formats - wide and long;

* 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;


/* PART 4*/


* 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;
ods graphics on;
title 'Exhibit 1: 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;

* Still not the correct graphic.  We want the tickers grouped and color coded by sector;
* We want ticker labels directly under the x-axis and sector labels under the ticker
        labels denoting each group.  Looks like we have an open SAS graphics project.;
ods graphics on;
title 'Exhibit 2: Correlations with the Market Index by Sector';
proc sgplot data=long_correlations;
        format correlation 3.2;
        vbar sector / response=correlation group=tkr groupdisplay=cluster datalabel;
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;

* Note that we have been using PROC SGPLOT to display a data summary, and hence we have not
	been able to make the display that we want.  In reality PROC SGPLOT is designed to take an
	input data set, perform some routine data summaries, and display that output.  Unfortunately,
	routine data summaries are typically frequency counts for discrete data or averages for
```

```
        contiuous data.  Here is an example of the default use of PROC SGPLOT.;
ods graphics on;
title 'Exhibit 3: Mean Correlations with the Market Index by Sector';
proc sgplot data=long_correlations;
        format correlation 3.2;
        vbar sector / response=correlation stat=mean datalabel;
run; quit;
ods graphics off;


* Reset title statement to blank;
title '';


/* PART 5 */



************************************************************************************;
* Begin Modeling;
************************************************************************************;
* Note that we do not want the response variable in the data used to compute the
        principal components;


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;



********************************************************************************************;
* Compute Principal Components;
********************************************************************************************;
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;
* 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;


* 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;
title 'Exhibit 5: Principal Component 1 and Principal Component 2';
ods graphics on;
proc sgplot data=long_pca;
scatter x=Prin1 y=Prin2 / datalabel=tkr;
run; quit;
```

```
ods graphics off;



/* PART 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);
        * 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;
```

```
* You can double check this merge by printing out the original data;

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



*******************************************************************************************
*******;

* Fit a regression model using the raw predictor variables;

*******************************************************************************************
*******;

* Fit a regression model using all of the raw predictor variables and VV as the response variable;


ods graphics on;

proc reg data=cv_data;

model train_response = return_: / vif ;

output out=model1_output predicted=Yhat ;

run; quit;

ods graphics off;

* Examine the Goodness-Of-Fit for this model.  How well does it fit?  Are there any problems?;



*******************************************************************************************
*******;

* Fit a regression model using the rotated predictor variables (Principal Component Scores) ;

*******************************************************************************************
*******;

* Now fit a regression model using your selected number of principal components and VV as
```

```
        the response variable;
* Examine the Goodness-Of-Fit for this model.  How well does it fit?  Are there any problems?;


ods graphics on;
proc reg data=cv_data;
model train_response = prin1-prin3 / vif ;
output out=model2_output predicted=Yhat  ;
run; quit;
ods graphics off;



******************************************************************************************
*******;
* Compare fit and predictive accuracy of the two fitted models ;
******************************************************************************************
*******;
* Use the Mean Square Error (MSE) and the Mean Absolute Error (MAE) metrics for your comparison.;


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


* Model 1;
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;


* Model 2;
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;


* Merge them together in one table;
data error_table;
        merge model1_error(drop= _TYPE_ _FREQ_) model2_error(drop= _TYPE_ _FREQ_);
        by train;
run;

proc print data=error_table; run;
```