Assignment #7

Crystal M. Mosley

Introduction:

The purpose of this examination is to run factor analysis to identify sectors in the stock market. The dataset being utilized consists of daily closing stock prices for 20 stocks, and an index fund from Vanguard. Data ranges from 01/03/12 – 12/31/13; 501 days, which is one record for each day in the dataset—zero days were skipped (no missing values per day).

We'll begin by computing the log-return of each stock/index variable.

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

Note: r_i = at a time i

p_i = price at a time i

$j = (i - 1)$

log-return = $\log(r_i)$

time i is in days

We will use *return* instead of *price* because it provides a mechanism of normalization which will allow for a measurement of all variables in a comparable metric, enabling evaluation of analytical relationship amongst two or more variables despite originating from a prices series of unequal values.

Factor Analysis will be used to identify sectors in the stock market. To gather better factor analysis results, some values from the dataset will be dropped. After eliminating these values, we are left with: Banking, Oil Field Services, Oil Refining, and Industrial-Chemical. Within the context of factor analysis, we hypothesize that we have three or four factor (industry sectors) within this dataset. The set criteria for significance of the factor loadings is going to be 0.5—this value is a random choice, but is a starting point for significance criteria and will enable more exclusivity within the selection. This threshold is saying that we choose to have at least half the variance accounted for by the factor for each variable.

Results:

Principal Factor Analysis—

We'll begin by performing a PCA without a factor rotation. The SAS procedure used will automatically select the number of factors to retain. The "keep" statement that falls within the "set" statement will only keep the log-returns of each variable.

Factor analysis begins by substituting the diagonal of the correlation matrix with what are called "prior communality estimates". The communality estimate for a variable is the estimate of the proportion of the variance of the variable that is both error free and shared amongst the other variables within the matrix. This calculation was completed using the SMC method which uses the squared multiple correlation between the variable and all other variables.

In observing the *prior communality estimates*, there are some values that are getting close to one (>0.6); consequently, the SMC method might not be the appropriate method for the modeling.

Prior Communality Estimates: SMC											
return_BAC	return_BHI	return_CVX	return_DD	return_DOW	return_HAL	return_HES	return_HUN	return_JPM	return_SLB	return_WFC	return_XOM
0.58577906	0.61046627	0.64179539	0.54681402	0.47197670	0.64986770	0.49976057	0.39225125	0.58034671	0.68269067	0.57372531	0.62696933

Next, examine the eigenvalues of the reduced correlation matrix:

Eigenvalues of the Reduced Correlation Matrix: Total = 6.86244298 Average = 0.57187025				
	Eigenvalue	Difference	Proportion	Cumulative
1	6.04732583	5.16261770	0.8812	0.8812
2	0.88470813	0.52262870	0.1289	1.0101
3	0.36207942	0.05735386	0.0528	1.0629
4	0.30472556	0.29429115	0.0444	1.1073
5	0.01043441	0.06365245	0.0015	1.1088
6	-.05321803	0.01517115	-0.0078	1.1011
7	-.06838918	0.03291807	-0.0100	1.0911
8	-.10130725	0.01600696	-0.0148	1.0763
9	-.11731422	0.00866270	-0.0171	1.0593
10	-.12597692	0.01040221	-0.0184	1.0409
11	-.13637913	0.00786652	-0.0199	1.0210
12	-.14424565		-0.0210	1.0000

The Scree plot could be examined; however, it's already noticeable from the above eigenvalue table that the first two eigenvalues have a large proportion of the variance, especially the first eigenvalue. This is evidence that the variables within the model are all highly correlated with each other and there is some hidden quality/trait that is fundamental with the correlation.

Next, examine the loadings of the factor pattern and their respective factor variance:

Factor Pattern		
	Factor1	Factor2
return_BAC	0.68475	0.36021
return_BHI	0.69984	-0.39498
return_CVX	0.77402	-0.10833
return_DD	0.71605	0.16703
return_DOW	0.64548	0.19801
return_HAL	0.72630	-0.38221
return_HES	0.70361	-0.15709
return_HUN	0.58030	0.18186
return_JPM	0.67874	0.34813
return_SLB	0.79382	-0.30815
return_WFC	0.72445	0.30517
return_XOM	0.76500	-0.08361

Variance Explained by Each Factor	
Factor1	Factor2
6.0473258	0.8847081

SAS retained two factors under its default settings. Since a MINEIGEN parameter was not specified when calling the FACTOR statement, the MINEIGEN will be calculated as:

MINEIGEN = Total Weighted Variance / Number of Variables

For this dataset, the MINEIGEN results in:

6.86244208 / 12 = 0.57

We are able to differentiate the factors into two groups due to the difference in the factor signs. The first group incorporates both the Banking and Industrial-Chemical sectors (BAC, DD, DOW, HUN, JPM, WFC), and the second group incorporates the Oil Refining and Oil Field Services sectors (BHI, CVX, HAL, HES, SLB, XOM). All of these variables are highly loaded for the first factor; however, the variables for the second factor do not meet the pre-specified criteria (0.5) for loading.

Interpreting these results systematically would produce the following equation for each variable within the analysis:

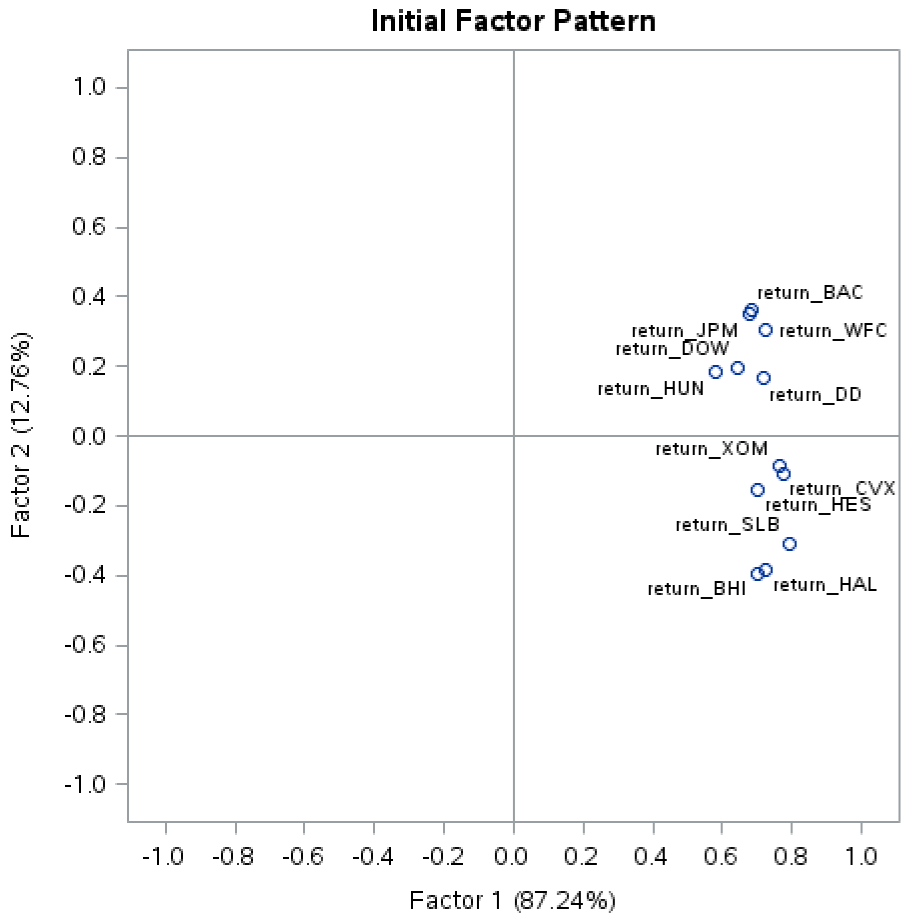
$$X_1 – \lambda 1 \int_1 + \lambda 2 \int_2 + \ldots + \lambda k \int_k + u_1$$

A good example using the variable, BAC:

$$\text{Return\_BAC} = 0.68475 \times f_1 + 0.36021 \times f_2$$

If we were strict with the loading criteria, we would not have included the second factor and loading coefficient within the equation. It is common to choose the largest factor and say that the variable is explained more by the larger factor (return_BAC is explained more by Factor1 than Factor2).

Next, examine the two factors within the graph:



The first factor has variables that all pass the loading threshold; however, the second factor does not. Since this doesn't provide much interpretability, we'll look at the sign within the second factor, only.

Principal Factor Analysis with Rotation (VARIMAX)—

We'll now perform a factor analysis with rotation. The rotation improves the interpretability of the model by seeking a *simple structure*, defined as a pattern of loadings where items load most strongly on one factor, and much more weakly on the other factors. By default, the un-rotated output maximizes the variance accounted for by the first and subsequent factors and forces the factors to be orthogonal. The Varimax rotation was chosen to maximize the variance of the squared loadings of a factor (column) on all variables (rows) in a factor matrix.

The factor analysis outputs are the same as above, but with new outputs for the rotation method:

Rotated Factor Pattern		
	Factor1	Factor2
return_BAC	0.73912	0.22875
return_BHI	0.21634	0.77394
return_CVX	0.47133	0.62344
return_DD	0.62482	0.38759
return_DOW	0.59675	0.31582
return_HAL	0.24408	0.78359
return_HES	0.38705	0.60822
return_HUN	0.53921	0.28120
return_JPM	0.72634	0.23305
return_SLB	0.34419	0.77886
return_WFC	0.72835	0.29575
return_XOM	0.48241	0.59958

Variance Explained by Each Factor	
Factor1	Factor2
3.4711423	3.4608916

After observing this model, we see that the rotation has given the ability to consider each factor as providing close to the same explanatory value for the variance within the model. Interpreting with the loading threshold of 0.5 allows us to see that Factor1 is comprised of BAC, DD, DOW, HUN, JPM, WFC. Factor2 is comprised of BHI, CVX, HAL, HES, SLB, XOM. Factor1 is comprised of Banking and Industrial-Chemical sectors, and Factor2 is comprised of Oil Refining and Oil Field Services sectors.

What is not seen is a factor that is loaded for a single variable. If this was visible, we would know to drop that variable from the model and consider it independently from the factor analysis.

Maximum Likelihood Factor Analysis with Rotation (VARIMAX)—

We'll now perform a maximum likelihood factor analysis with varimax rotation. Maximum likelihood is a formal estimation procedure that provides us with the formal inference for factor loadings and goodness-of-fit criteria. The method computes an initial set of eigenvalues to assess the convergence criterion.

MINEIGEN: 18.8960127/12 = 1.574667725

This means we'll be receiving a model with two factors.

Once the criterion was met, the model shows two separate statistical hypothesis tests with the null hypothesis stated as ‘no common factors’ and ‘2 Factors are sufficient’. Both tests allow us to accept the null hypothesis.

Significance Tests Based on 501 Observations			
Test	DF	Chi-Square	Pr > ChiSq
H0: No common factors	66	3656.2617	<.0001
HA: At least one common factor			
H0: 2 Factors are sufficient	43	319.3192	<.0001
HA: More factors are needed			

Rotated Factor Pattern		
	Factor1	Factor2
return_BAC	0.76122	0.21969
return_BHI	0.21664	0.79932
return_CVX	0.49806	0.57530
return_DD	0.59542	0.38748
return_DOW	0.56395	0.31884
return_HAL	0.24256	0.80907
return_HES	0.40289	0.59153
return_HUN	0.50588	0.29457
return_JPM	0.75054	0.22277
return_SLB	0.35223	0.79376
return_WFC	0.75994	0.27534
return_XOM	0.51113	0.55362

Variance Explained by Each Factor		
Factor	Weighted	Unweighted
Factor1	8.7156851	3.55022275
Factor2	10.1803287	3.42320994

The same amount of common factors is suggested by the ML method. The factor loadings between PFA and ML with rotations are similar, leaving no difference in interpretability. The added benefit of utilizing the ML method is the goodness-of-fit criteria—this allows for model comparisons.

Maximum Likelihood Factor Analysis with Rotation and Max Priors—

With the Max Priors parameter set, the threshold for accepting factor may be drastically different. Max will set the prior communality estimate for each variable to its maximum absolute correlation with any other variable.

Recalculated MINEIGEN: 27.8241868/12 = 2.318682233

From this recalculated eigenvalue, we would expect to see two factors from this method; however, we see five:

Rotated Factor Pattern					
	Factor1	Factor2	Factor3	Factor4	Factor5
return_BAC	0.19300	0.75425	0.26803	0.17215	0.09285
return_BHI	0.75597	0.14970	0.18684	0.24628	-0.01722
return_CVX	0.37688	0.25354	0.26440	0.70383	0.02658
return_DD	0.24372	0.27524	0.66859	0.31138	-0.13337
return_DOW	0.19396	0.25931	0.64481	0.23505	-0.00701
return_HAL	0.82071	0.18978	0.20801	0.16916	-0.00609
return_HES	0.47834	0.23976	0.25785	0.40900	0.24903
return_HUN	0.22592	0.26677	0.60996	0.06709	0.16770
return_JPM	0.20547	0.77151	0.22874	0.17842	-0.03102
return_SLB	0.72537	0.25575	0.24707	0.30301	0.05701
return_WFC	0.20847	0.61032	0.35934	0.29285	-0.00631
return_XOM	0.37166	0.29603	0.24083	0.66560	-0.02404

Variance Explained by Each Factor		
Factor	Weighted	Unweighted
Factor1	9.48177257	2.55119512
Factor2	6.95572063	2.08400430
Factor3	5.26449075	1.82173920
Factor4	5.80237050	1.59069819
Factor5	0.31984016	0.12246466

The above suggests that the factor analysis is highly dependent upon the prior estimates of communalities. Max Priors is a highly inclusive method for computing communalities. After observing the rotated factors above, it's noticeable that some of the factors will only be inclusive of a small subset of the variables (based on loading conditions).

The method chosen for *priors calculation* and communalities is highly influential over the chosen factors from the model. Using Max Priors led us closer to what was initially expected from familiarity with the dataset.

Code:

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

data temp;
 set mydata.stock_portfolio_data;
 drop AA HON MMM DPS KO PEP MPC GS ;
run;
```

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

proc sort data=temp;
	by date;
run;
quit;


data temp;
  set temp;
  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_HAL = log(HAL/lag1(HAL));
  return_HES = log(HES/lag1(HES));
  return_HUN = log(HUN/lag1(HUN));
  return_JPM = log(JPM/lag1(JPM));
  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;

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

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

ods graphics on;
proc factor data=return_data method=principal priors=smc rotate=none
  plots=(all);
run;
quit;

ods graphics off;

ods graphics on;
```

```
proc factor data=return_data method=principal priors=smc rotate=varimax
  plots=(all);
run;
quit;

ods graphics off;

ods graphics on;
proc factor data=return_data method=ML priors=smc rotate=varimax
  plots=(loadings);
run;
quit;

ods graphics off;

ods graphics on;
proc factor data=return_data method=ML priors=max rotate=varimax
  plots=(loadings);
run;
quit;

ods graphics off;
```