Solutions to Laboratory Exercise 3

Correlation Analysis

Problem 3.1 Clean the data for any missing or erroneous values in order to include only the stocks that have complete data available on the above variables at the end of 2004. Create the corresponding variables.

Sol: Use the package `readxl` to import the Excel data and function `na.omit` to remove the observations with NAs. Then, rename the variables to a single word.

```
# 3.1
library(readxl)
raw = read_excel('lab3_data.xlsx',
                 sheet = 'New Search Criteria',
                 range = 'A1:L3984')

tempData = na.omit(setDT(raw))
setnames(tempData, c("name", "ticker", "exchange",
                     "be", "clsgPrice", "size",
                     "cP", "eP", "ebitdaP", "fcfP", "sP",
                     "yr1Ret"))

head(tempData)
```

```
##                      name ticker                        exchange          be clsgPrice
## 1: 3D Systems Corporation    DDD NEW YORK STOCK EXCHANGE, INC.    53065197      9.94
## 2:                  3M Co    MMM NEW YORK STOCK EXCHANGE, INC. 10378010967     82.07
## 3:         7-Eleven, Inc.     SE NEW YORK STOCK EXCHANGE, INC.   464457580     23.95
## 4:   99 Cents Only Stores    NDN NEW YORK STOCK EXCHANGE, INC.   488284620     16.16
## 5:      A.M. Castle & Co.    CAS NEW YORK STOCK EXCHANGE, INC.   119010356     11.94
## 6: A.O. Smith Corporation    AOS NEW YORK STOCK EXCHANGE, INC.   590598967     19.96
##           size           cP          eP     ebitdaP         fcfP        sP      yr1Ret
## 1:   266546089 -0.003287668 -0.02716297 -0.01068591 -0.005724872 0.4813462 -0.09456722
## 2: 63894267160  0.064420994  0.04398684  0.06751629  0.051094626 0.2994319 -0.03498928
## 3:  2704339349  0.187733635  0.03799583  0.05189160  0.082927335 4.1618112  0.56450928
## 4:  1123212402  0.074543644  0.02722772  0.07143036  0.009231181 0.8225010 -0.35272282
## 5:   188609457  0.083026137  0.04103852  0.17753125  0.050786737 3.5150373  0.82914550
## 6:   882857546  0.028269630  0.04909819  0.08614979 -0.023464914 1.8017693  0.19880601
```

Problem 3.2 Since the existence of a single outlying value can markedly influence the results of correlation coefficients, remove any outliers present in the variables, which we have previously defined as data values that are 3 standard deviations away from their means.

Sol: Create a function to identify the outliers. Use the apply family to calculate the means and standard deviations of the explanatory variables and remove any outlying values from the data.

```
# 3.2
Data = tempData[, .SD, .SDcols = -c("name", "exchange")]

is.out = function(x) abs(x - mean(x)) > 3 * sd(x)
sumData = sapply(Data[, .(cP, eP, ebitdaP, fcfP, sP, yr1Ret)],
                  function(x) c(mean = mean(x), sd = sd(x),
                                nOut = sum(is.out(x))))
sumData
```

```
##                cP         eP    ebitdaP        fcfP        sP     yr1Ret
## mean  0.09520172 0.04808802  0.1178378  0.02231821  1.153064  0.1467654
## sd    0.10044826 0.42421765  0.1355957  0.17031667  1.579924  0.3901031
## nOut 27.00000000 2.00000000 27.0000000 17.00000000 25.000000 24.0000000
```

```
outliersIndxA = sapply(Data[, .(cP, eP, ebitdaP, fcfP, sP, yr1Ret)], is.out)
outliersIndxB = apply(outliersIndxA, 1, any)

(Nout = sum(outliersIndxB))
```

```
## [1] 102
```

```
Data = Data[!outliersIndxB, ]
rm(sumData, outliersIndxA, outliersIndxB)
```

We find 102 outliers, and this leaves us with a 1382-by-10 dataset.

Problem 3.3 Regression models intended to explain the cross-section of average stock returns tend to make certain assumptions regarding the collected data in order to avoid distortion in the interpretation of the results. Given the difficulty in interpreting stocks with negative book values and negative earnings, we will exclude them from our data sets. However it is important to note that stocks with negative book values and earnings, behave like stocks with low book value of equity-to-market value of equity (BE/ME) and low earnings-to-price (E/P) ratios, which tend to have lower average returns. Also to avoid that the data sets are dominated by small cap stocks, we will exclude stocks with a closing price lower than $5 and with a market capitalization lower than $100m. Filter the data to avoid stocks with negative book values and earnings, with a closing price lower than $5, and with a market capitalization lower than $100m (any of these three).

Sol: In this example, we are keeping only those stocks with book values and P/E ratios greater than 0, with prices greater than $5, and with a market capitalization of $100m or greater.

```
# 3.3
Data = Data[be > 0 & eP > 0 & clsgPrice > 5 & size > 100000000]
```

This leaves us with an 1198-by-10 dataset.

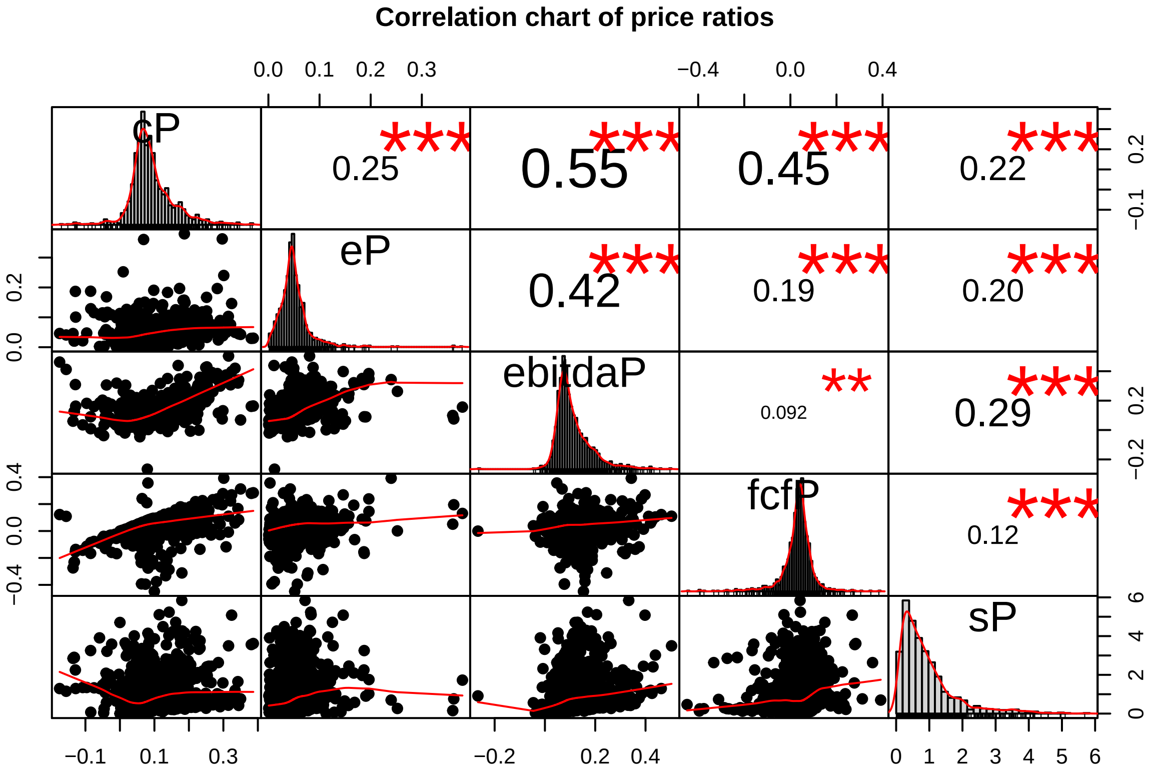
Problem 3.4 Perform a correlation analysis to assess whether a linear association exists between each pair of explanatory variables.

Sol: The aim of this correlation analysis is to look for significant and/or spurious linear associations between each pair of explanatory variables. When these sorts of linear associations are present in regression models, the models tend to encounter problems.

(a) Use the function `chart.Correlation` in the package `PerformanceAnalytics` to visualize the relationship between each pair of the variables.

Sol: The output is a plot matrix. In the lower triangular, this function creates the scatter plot in the (i, j)th position of the array. In addition, the plot in the ith diagonal position is a histogram of the ith variable. The upper triangular features the correlation between two variables and its significance level.

```
# 3.4 (a)
library(PerformanceAnalytics)
chart.Correlation(as.matrix(Data[, .(cP, eP, ebitdaP, fcfP, sP)]),
                  histogram = TRUE, pch = 19, method = "pearson",
                  main = "Correlation chart of price ratios",
                  cex.main = 0.80)
```



(b) Compute the correlation coefficients to quantify the direction and strength (if any) of linear association between each pair of the variables. You may use the package `psych` to create a correlation matrix.

Sol: The correlation coefficients for the above pairs of explanatory variables can be calculated using the function `corr.test` in the package `psych`, which returns a list of several useful results. The first element of the list is the matrix of the correlation coefficients. They tell us the direction and extent of association. The diagonal matrix elements represent the perfect correlation of each variable with itself and are equal to 1. The off-diagonal elements indicate statistical correlations between the variables. We can interpret these off-diagonal elements in the following way: a 0.1000 value can be interpreted as a low correlation, a 0.3000 value as a moderate correlation, and a 0.5000 as a high (or very) strong correlation between the pairs of variables. A correlation coefficient greater than 0.5000 could exist between two different measures of the same variable. Such large correlations might indicate a spurious (or artificial) relationship.

```
# 3.4 (b)
library(psych)
correlation = corr.test((Data[, .(cP, eP, ebitdaP, fcfP, sP)]))
correlation[["r"]]
```

```
##                cP        eP    ebitdaP       fcfP        sP
## cP      1.0000000 0.2467193 0.55456290 0.45375304 0.2193445
## eP      0.2467193 1.0000000 0.41545351 0.19298401 0.2038081
## ebitdaP 0.5545629 0.4154535 1.00000000 0.09225386 0.2941417
## fcfP    0.4537530 0.1929840 0.09225386 1.00000000 0.1240637
## sP      0.2193445 0.2038081 0.29414166 0.12406369 1.0000000
```

For this sample, there is a positive linear association between the pairs of variables. This may not be surprising since for instance, firms with high C/P ratios tend to have high E/P ratios. As the previous scatter plots showed, we can see that the correlations between **cP** and **ebitdaP**, and between **cP** and **fcfP** may represent some kind of artificial relationship. Also there is a moderate to large correlation between **eP** and **ebitdaP**. The matrix also shows some small correlations between **ebitdaP** and **fcfP**. The rest of the correlations appear to be between the [0.1000, 0.4000] range.

(c) Create a matrix of p-values in order to test the hypothesis of no correlation between each pair of the variables against the alternative of significant correlation between each pair of the variables. Interpret the results.

Sol: The results generated from `corr.test` also contain the significance of the correlation coefficients by calculating a matrix of p-values for purposes of testing the hypothesis of no correlation (i.e., the correlation between the variables is due to chance relationships in this particular data sample) against the alternative that there is a nonzero correlation. Each p-value is the probability of getting a correlation as large as the observed value by random chance, when the true correlation is zero. If *p(i,j)* is small, say less than 0.05, then the correlation *r(i,j)* is significantly different from zero.

```
# 3.4 (c)
correlation[["p"]]
```

```
##                   cP           eP    ebitdaP         fcfP           sP
## cP      0.000000e+00 0.000000e+00 0.00000000 0.000000e+00 8.104628e-14
## eP      0.000000e+00 0.000000e+00 0.00000000 4.883094e-11 4.263256e-12
## ebitdaP 0.000000e+00 0.000000e+00 0.00000000 1.390630e-03 0.000000e+00
## fcfP    0.000000e+00 1.627698e-11 0.00139063 0.000000e+00 3.320561e-05
## sP      1.620926e-14 1.065814e-12 0.00000000 1.660280e-05 0.000000e+00
```

The following construct finds the significant correlations and displays it by their (row, column) indices.

```
which(correlation[["p"]] < 0.05, arr.ind = T)
```

```
##         row col
## cP        1   1
## eP        2   1
## ebitdaP   3   1
## fcfP      4   1
## sP        5   1
## cP        1   2
## eP        2   2
## ebitdaP   3   2
## fcfP      4   2
## sP        5   2
## cP        1   3
## eP        2   3
## ebitdaP   3   3
## fcfP      4   3
## sP        5   3
## cP        1   4
## eP        2   4
## ebitdaP   3   4
## fcfP      4   4
## sP        5   4
## cP        1   5
## eP        2   5
## ebitdaP   3   5
## fcfP      4   5
## sP        5   5
```

This test tells us that there is a significant correlation between most of the pairs of variables in this sample. Therefore we may reject hypothesis of no correlation, which discards the possibility of spurious correlations between the variables due to chance relationships in this particular sample.

Problem 3.5 Run a correlation analysis to evaluate the presence of linear associations between the explained variable and the explanatory variables.

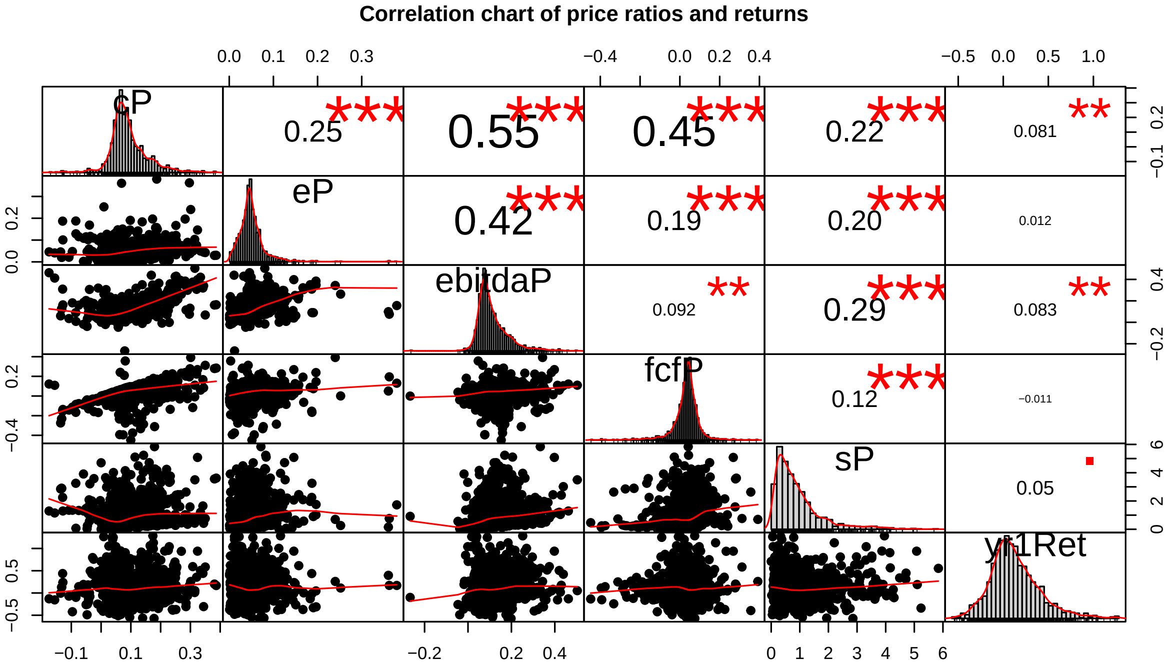
Sol: The aim of this correlation analysis is to look for linear associations between the explained variable and each pair of the explanatory variables. Statistically significant correlation coefficients may provide valuable information about the behavior of returns. When there is no linear association between the variables, there is no reason to include them in a regression model as they will not provide useful information about the explained variable.

(a) Use the function `chart.Correlation` in the package `PerformanceAnalytics` to visualize the relationship between the explained variables and each of the explanatory variables.

Sol: Again, in the lower triangular, this function creates the scatter plot in the (i, j)th position of the array. In addition, the plot in the ith diagonal position is a histogram of the ith variable. The upper triangular features the correlation between two variables and its significance level. The resulting plot matrix gives us an idea of how the explanatory variables fluctuate against the explained variable.

From the scatter plots and the correlations we can see that there may be some positive relationships between the explained variables and the explanatory variables. In particular, cP and ebitdaP seem to reveal strong predictability.

```
# 3.5 (a)
chart.Correlation(as.matrix(Data[, .(cP, eP, ebitdaP, fcfP, sP, yr1Ret)]),
                  histogram = TRUE, pch = 19,method = "pearson",
                  main = "Correlation chart of price ratios and returns",
                  cex.main = 0.80)
```



(b) Compute the correlation coefficients. You may use the package `psych` to create a correlation matrix.

Sol: The correlation matrix between the explained variable and each pair of the explanatory variables is computed using the function `corr.test`. In this case the correlations are shown in the last row and column of the matrix.

```
# 3.5 (b)
correlation1 = corr.test(Data[, .(cP, eP, ebitdaP, fcfP, sP, yr1Ret)])
correlation1[["r"]]
```

```
##                 cP         eP    ebitdaP        fcfP         sP      yr1Ret
## cP      1.00000000 0.24671928 0.55456290  0.45375304 0.21934453  0.08076978
## eP      0.24671928 1.00000000 0.41545351  0.19298401 0.20380812  0.01209021
## ebitdaP 0.55456290 0.41545351 1.00000000  0.09225386 0.29414166  0.08301055
## fcfP    0.45375304 0.19298401 0.09225386  1.00000000 0.12406369 -0.01110134
## sP      0.21934453 0.20380812 0.29414166  0.12406369 1.00000000  0.05001146
## yr1Ret  0.08076978 0.01209021 0.08301055 -0.01110134 0.05001146  1.00000000
```

With a positive (negative) correlation coefficient between the explained variable and the explanatory variable, we can expect a positive (negative) regression coefficient for the explanatory variable in a regression on the explained variable.

(c) Create a matrix of p-values in order to test the hypothesis of no correlation between each pair of the variables against the alternative of significant correlation between each pair of the variables. Interpret the results.

Sol: We can assess the significance of the relationships to confirm our prior assessment by calculating the matrix of the p-values for the correlation coefficients. In large samples, even relatively small correlations can be significantly different from zero. The test showing the significance of the correlations coincides with our assessment that only **cP** and **ebitdaP** have significant correlations with the explained variable **yr1Ret**.

```
# 3.5 (c)
correlation1[["p"]]
```

```
##                   cP           eP     ebitdaP         fcfP           sP     yr1Ret
## cP      0.000000e+00 0.000000e+00 0.000000000 0.000000e+00 1.620926e-13 0.02061473
## eP      0.000000e+00 0.000000e+00 0.000000000 1.302158e-10 9.592327e-12 1.00000000
## ebitdaP 0.000000e+00 0.000000e+00 0.000000000 8.343781e-03 0.000000e+00 0.02019264
## fcfP    0.000000e+00 1.627698e-11 0.001390630 0.000000e+00 1.162196e-04 1.00000000
## sP      1.620926e-14 1.065814e-12 0.000000000 1.660280e-05 0.000000e+00 0.25074120
## yr1Ret  5.153683e-03 6.759131e-01 0.004038528 7.010883e-01 8.358040e-02 0.00000000
```

```
which(correlation1[["p"]] < 0.05, arr.ind = T)
```

```
##         row col
## cP        1   1
## eP        2   1
## ebitdaP   3   1
## fcfP      4   1
## sP        5   1
## yr1Ret    6   1
## cP        1   2
## eP        2   2
## ebitdaP   3   2
## fcfP      4   2
## sP        5   2
## cP        1   3
## eP        2   3
## ebitdaP   3   3
## fcfP      4   3
## sP        5   3
## yr1Ret    6   3
## cP        1   4
## eP        2   4
## ebitdaP   3   4
## fcfP      4   4
## sP        5   4
## cP        1   5
## eP        2   5
## ebitdaP   3   5
## fcfP      4   5
## sP        5   5
## cP        1   6
## ebitdaP   3   6
## yr1Ret    6   6
```

Problem 3.6 Compute the z-scores for each of the explanatory variables and compute the correlation coefficients between each pair of the explanatory variables. Do these results change your answers in problems 3.4? Why? Aggregate the z-scores for the explanatory variables and compute the correlation coefficient between the aggregated explanatory variables and the explained variable. Interpret the results.

Sol: Use the function `scale` to calculate the z-scores for each of the explanatory variables. Then compute their pairwise correlations.

```
# 3.6
zscore = sapply(Data[, .(cP, eP, ebitdaP, fcfP, sP)], scale)
zscore = scale(Data[, .(cP, eP, ebitdaP, fcfP, sP)])
correlationZ = corr.test(zscore)
correlationZ[["r"]]
```

```
##                cP        eP    ebitdaP       fcfP        sP
## cP      1.0000000 0.2467193 0.55456290 0.45375304 0.2193445
## eP      0.2467193 1.0000000 0.41545351 0.19298401 0.2038081
## ebitdaP 0.5545629 0.4154535 1.00000000 0.09225386 0.2941417
## fcfP    0.4537530 0.1929840 0.09225386 1.00000000 0.1240637
## sP      0.2193445 0.2038081 0.29414166 0.12406369 1.0000000
```

The aggregate z-score is an equal weighted average (or mean) of the single factor z-scores. When we compute the correlation coefficient with the aggregate z-score we must understand that correlations computed from aggregated data, where the values are averaged across stocks, would be larger than what they would be if the individual variables were used. This is due to increased reliability and therefore we should never interpret aggregated correlation as if it were correlations from individual variables.

```
zAggr = apply(zscore, 1, mean)
cor(Data[, yr1Ret], zAggr)
```

```
## [1] 0.06598751
```

Problem 3.7 Based on the correlation coefficients and their significance tests in problem 3.4, is there any variable that may introduce multicollinearity to a regression that combines the 5 fundamental factors so as to explain the average returns for the next year? Identify the variable(s) and explain how we can correct for multicollinearity.

Sol: Multicollinearity arises when two or more explanatory variables (or combinations of explanatory variables) are highly (but not perfectly) correlated with each other. Coefficient estimates for multiple linear regression models rely on the independence of the model variables. When the explanatory variables are correlated and they have an approximate linear dependence, the matrix inverse needed to calculate the least-squares estimates becomes highly sensitive to random errors in the observed explained variable, thereby producing a large variance. Consequently with multicollinearity we can estimate the regression, but the interpretation of the regression estimates becomes problematic. Moreover, tests of regressions with highly correlated predictors lack explanatory power in order to separate the effects of one predictor from the effects of the other in the explained variable. Some symptoms of multicollinearity may include slope coefficients with high standard errors and low significance levels for the t-statistics or high multiple R^2 (and significant F-statistic) levels, but coefficients with wrong signs or implausible magnitudes.

Using the correlation coefficients in problem 3.4, we identified the explanatory variables with high correlations with each other's. Given the significance tests, we concluded that there were significant correlations between each pair of the explanatory variables. However, not all of those variables had high correlations with each other's. We can compute the lower and upper bounds for a 95% confidence interval for each coefficient using the results from the function `coef.test` to assess which of the variables may introduce some sort of multicollinearity in the regression model.

```r
# 3.7
(CI = round(correlation[["ci"]], 4))
```

```
##              lower      r  upper      p
## cP-eP       0.1928 0.2467 0.2992 0.0000
## cP-ebtdP    0.5141 0.5546 0.5926 0.0000
## cP-fcfP     0.4076 0.4538 0.4976 0.0000
## cP-sP       0.1648 0.2193 0.2726 0.0000
## eP-ebtdP    0.3675 0.4155 0.4612 0.0000
## eP-fcfP     0.1379 0.1930 0.2469 0.0000
## eP-sP       0.1489 0.2038 0.2575 0.0000
## ebtdP-fcfP 0.0358 0.0923 0.1481 0.0014
## ebtdP-sP    0.2415 0.2941 0.3450 0.0000
## fcfP-sP     0.0679 0.1241 0.1794 0.0000
```

For instance, the confidence bounds tell us that we could keep **cP**, **eP**, and **sP** since their 95% confidence intervals for their correlation coefficients lie between [0.1489, 0.2992], (i.e., [0.1928, 0.2992] for **cP** and **eP**, [0.1648, 0.2726] for **cP** and **sP**, and [0.1489, 0.2575] for **eP** and **sP**). However we suspect the correlations between each pair of **cP**, **eP**, and **sP** are due to spurious relations between them. These spurious relationships are induced in part from (i) the direct relationship between them, as they are derived from the others, which will cause a greater proportion of shared variance; and in part from (ii) their relation to third component, which in this case is price, as they are all scaled versions of price. Spurious relationships of this sort result in larger correlations than should be. Consequently the most direct solution to correct for multicollinearity is to exclude one or more of the variables and in this case, leave only the one with the highest correlation together with the explained variable.

Problem 3.8 Analyzing the correlation coefficients and their significance tests of problem 3.5 and the results in problem 3.7, would we run the risk of overfitting a regression model that incorporates all of the 5 fundamental factors to explain average returns for the next year? Why? Is there any way to prevent overfitting a regression model?

Sol: Overfitting is the use of models that include more explanatory variables than are necessary, i.e., models that include irrelevant as well as the needed predictors. Adding irrelevant predictors can make predictions worse because the regression coefficients fitted to them add random variation to the subsequent predictions. Given the results in problems 3.5 and 3.7, if we were to fit a model that incorporates the 5 explanatory variables we would definitely run the risk of overfitting the regression model. First, as we saw in problem 3.5, when the correlation coefficients are interpreted, as the approximate proportion of variation explained, only 2 predictors (**cP** and **ebitdaP**) showed statistically significant levels. Therefore, the other 3 predictors (**cP**, **eP**, and **sP**) will just add noise to the regression, which will decrease the adjusted R^2 measure. However as we discussed in problem 3.7, the confidence interval for the correlation between **cP** and **ebitdaP** is [0.5141, 0.5926], which means that including both variables in a model would make it prone to multicollinearity since both variables will compete for explanatory power.

Therefore we can compute the lower and upper bounds for a 95% confidence interval of the coefficients between these two variables and the explained variable, in order to assess which one to include in the regression model.

```
# 3.8
correlation1[["ci"]][c("cP-ebtdP", "cP-yr1Rt", "ebtdP-yr1Rt"), ]
```

```
##                  lower          r     upper           p
## cP-ebtdP    0.51407239 0.55456290 0.5925874 0.000000000
## cP-yr1Rt    0.02424383 0.08076978 0.1367809 0.005153683
## ebtdP-yr1Rt 0.02649828 0.08301055 0.1389939 0.004038528
```

Since the confidence intervals of the correlation coefficients are similar for both variables, we can conduct a test of significance specifying a higher confidence level of 99.5% to decide which of the variables to include. **ebitdaP** is the final variable we are going to pick. The regression result is also included, and we will do more about regressions in lab 4.

```
correlation2 = corr.test(Data[, .(cP, ebitdaP, yr1Ret)], alpha = 0.005)
correlation2[["ci"]]
```

```
##                     lower          r     upper           p
## cP-ebtdP     0.4958178473 0.55456290 0.6082558 0.000000000
## cP-yr1Rt    -0.0002553174 0.08076978 0.1607413 0.005153683
## ebtdP-yr1Rt  0.0020005797 0.08301055 0.1629381 0.004038528
```

```
model = lm(yr1Ret ~ ebitdaP, data = Data)
summary(model)
```

```
## 
## Call:
## lm(formula = yr1Ret ~ ebitdaP, data = Data)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.71712 -0.17880 -0.04017  0.14579  1.17883 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   0.0809     0.0152   5.323 1.22e-07 ***
## ebitdaP       0.3261     0.1132   2.881  0.00404 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.2788 on 1196 degrees of freedom
## Multiple R-squared:  0.006891,   Adjusted R-squared:  0.00606 
## F-statistic: 8.299 on 1 and 1196 DF,  p-value: 0.004039
```