Homework 1

42205 / Marc Gehring

2022-02-14

1.

```
library(tidyverse)
library(readxl)
library(lubridate)
library(reshape2)
library(psych)
library(ggh4x)
library(QRM)
library(moments)
```

(a)

```
path = paste0(getwd(), "/data_assignment1.xlsx")
data = lapply(excel_sheets(path), read_excel, path = path)

data = data[[1]]
data = data %>% mutate(DATE = ymd(DATE))

head(data)
```

```
## # A tibble: 6 x 6
##   DATE       COCACOLA      GE     IBM   VWRET   EWRET
##   <date>        <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 1963-01-31  0.0587   0.0195  0.0865  0.0518  0.0891
## 2 1963-02-28 -0.00277 -0.0703 -0.0543 -0.0219 -0.0145
## 3 1963-03-29  0.0464   0.0326  0.0532  0.0329  0.0120
## 4 1963-04-30 -0.0107   0.0553  0.102   0.0474  0.0269
## 5 1963-05-31  0.00540  0.0667  0.0313  0.0201  0.0323
## 6 1963-06-28  0.0153  -0.0536 -0.0859 -0.0180 -0.0104
```

(b)

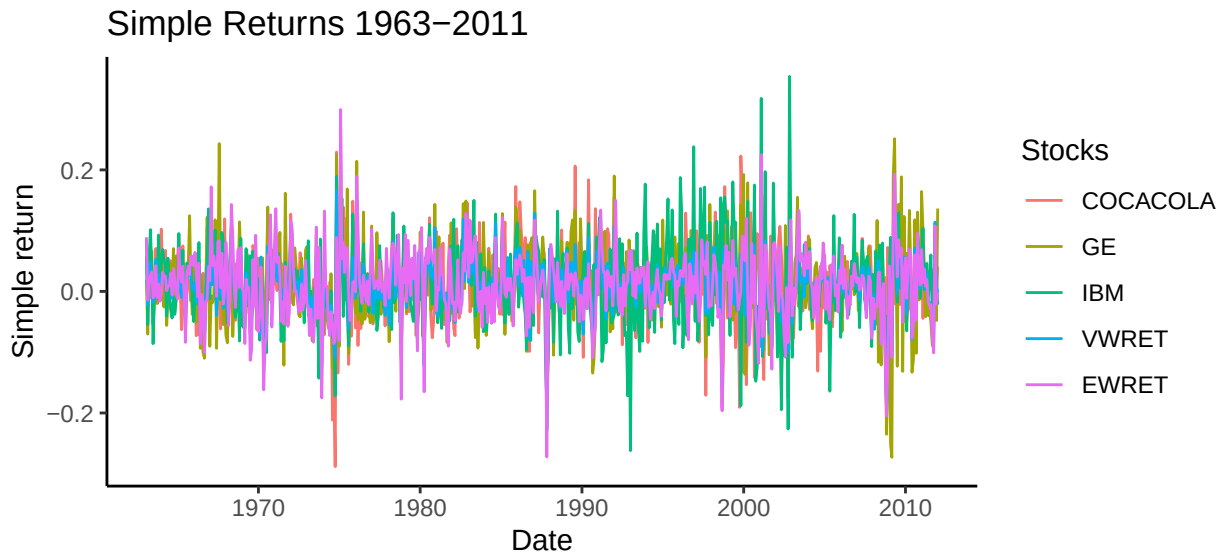
```
simpleToLog = function(returns) {return(log(returns + 1))}
for (i in (2:ncol(data))) {
```

```
  data[paste0("log",names(data)[i])] = simpleToLog(data[, i])
}

head(data)
```

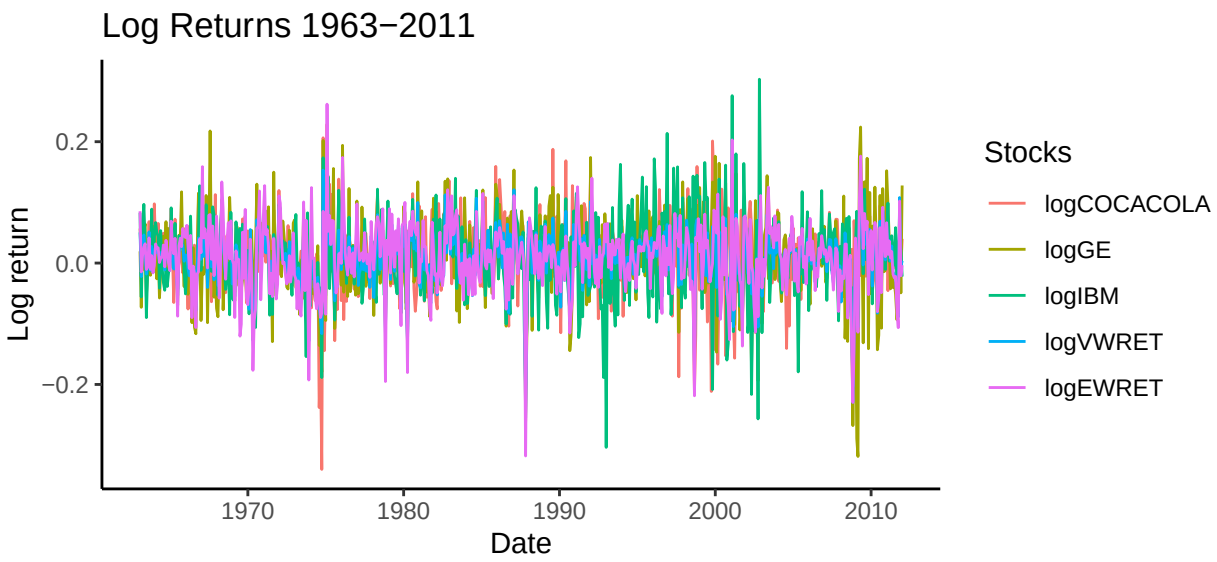
```
## # A tibble: 6 x 11
##   DATE       COCACOLA      GE     IBM    VWRET   EWRET logCOCACOLA   logGE
##   <date>        <dbl>   <dbl>   <dbl>   <dbl>   <dbl>       <dbl>   <dbl>
## 1 1963-01-31  0.0587   0.0195  0.0865  0.0518  0.0891     0.0570   0.0194
## 2 1963-02-28 -0.00277 -0.0703 -0.0543 -0.0219 -0.0145    -0.00277 -0.0729
## 3 1963-03-29  0.0464   0.0326  0.0532  0.0329  0.0120     0.0453   0.0321
## 4 1963-04-30 -0.0107   0.0553  0.102   0.0474  0.0269    -0.0108   0.0538
## 5 1963-05-31  0.00540  0.0667  0.0313  0.0201  0.0323     0.00539  0.0645
## 6 1963-06-28  0.0153  -0.0536 -0.0859 -0.0180 -0.0104     0.0152  -0.0551
## # ... with 3 more variables: logIBM <dbl>, logVWRET <dbl>, logEWRET <dbl>
```

(c)

```
ggplot(melt(data[,1:6], id.vars = "DATE"), aes(DATE, value, color = variable)) +
geom_line() +
ggtitle("Simple Returns 1963-2011") +
labs(x = "Date", y = "Simple return") +
guides(color = guide_legend(title = "Stocks")) +
theme_classic()
```



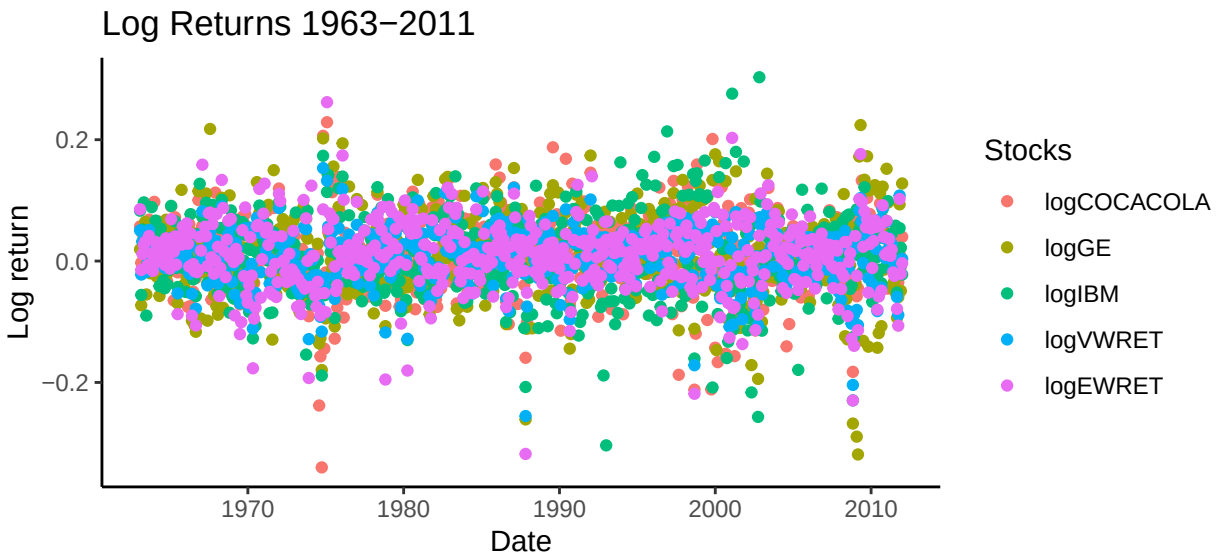

```
ggplot(melt(data[,c(1,7:ncol(data))], id.vars = "DATE"),
  aes(DATE, value, color = variable)) +
geom_line() +
ggtitle("Log Returns 1963-2011") +
labs(x = "Date", y = "Log return") +
guides(color = guide_legend(title = "Stocks")) +
theme_classic()
```



There are no striking differences between the patterns of the simple and the log returns. That comes as no surprise, as simple and log returns are similar for small numbers. Individual time series are hard to distinguish. One can, however, see periods of higher volatility and peaking returns. At the beginning of the 2000s, for example, you can see a relatively broad range of spikes, corresponding to the dot-com bubble. Similarly, you can see large negative spikes around the time of the global financial crisis. In general, the individual stocks, compared to the stock indices, peak more extremely, more often. The inner band of the data seems to be dominated by the colors of the two stock indices. This is to be expected, as a consequence of diversification within a stock index.

(d)

```
ggplot(melt(data[,c(1,7:ncol(data))], id.vars = "DATE"),
  aes(DATE, value, color = variable)) +
geom_point() +
ggtitle("Log Returns 1963-2011") +
labs(x = "Date", y = "Log return") +
guides(color = guide_legend(title = "Stocks")) +
theme_classic()
```



This plot is similar in content to the previous one. In this plot, though, it perhaps becomes more apparent that on the whole, the returns of the indices tend to be closer to 0. The data points of COCACOLA, GE, and IBM tend to lie outside of the cloud of index data points.

(e)

```
descriptiveStats = describe(data[, 7:ncol(data)]) %>%
  mutate(excessKurtosis = kurtosis - 3)
descriptiveStats
```

```
##             vars   n mean   sd median trimmed  mad   min  max range  skew
## logCOCACOLA    1 588 0.01 0.06   0.01    0.01 0.05 -0.34 0.23  0.57 -0.56
## logGE          2 588 0.01 0.07   0.01    0.01 0.06 -0.32 0.22  0.54 -0.30
## logIBM         3 588 0.01 0.07   0.01    0.01 0.06 -0.30 0.30  0.61 -0.17
## logVWRET       4 588 0.01 0.05   0.01    0.01 0.04 -0.26 0.15  0.41 -0.80
## logEWRET       5 588 0.01 0.06   0.01    0.01 0.05 -0.32 0.26  0.58 -0.61
##             kurtosis se excessKurtosis
## logCOCACOLA     2.96  0          -0.04
## logGE           1.81  0          -1.19
## logIBM          1.89  0          -1.11
## logVWRET        2.86  0          -0.14
## logEWRET        3.45  0           0.45
```

kurtosis is already excess kurtosis!

The equally-weighted index has a higher average return than the value-weighted index; 0.0101 vs 0.0077. At the same time, the equally-weighted index is more volatile, as measured by variance, than the value-weighted index; 0.0033 vs 0.0021. Both return series are negatively skewed, indicating that the tail is on the left side of the distribution. The value-weighted index exhibits more skewness than the equally-weighted index; -0.7994 vs -0.6136. Excess kurtosis describes the shape of the distribution tails. The normal distribution has kurtosis of 3, so the kurtosis value of a distribution minus 3 is the excess kurtosis. The value-weighted index has negative excess kurtosis of -0.1418, making it platykurtic; the distribution's tails are thinner than those of a normal distribution. Extreme events occur less frequently than predicted by a normal distribution. The equally-weighted index has positive excess kurtosis of 0.4465, making it leptokurtic; the distribution's tails are thicker than those of a normal distribution. Extreme events occur more frequently than predicted by

a normal distribution. Finally, the value-weighted index has a broader range of returns, from -0.2554 to +0.1532. The returns of the equally-weighted index, on the other hand, range from -0.3178 to +0.2618. This seems to agree with the excess kurtosis values.

(f)

```
for (i in 7:ncol(data)) {
  print(names(data)[i])
  print(t.test(data[, i], mu = 0))
  print(agostino.test(data[[i]]))
  print(anscombe.test(data[[i]]))
}
```

```
## [1] "logCOCACOLA"
##
##  One Sample t-test
##
## data:  data[, i]
## t = 4.2605, df = 587, p-value = 2.375e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.005840269 0.015829680
## sample estimates:
##  mean of x
## 0.01083497
##
##
##  D'Agostino skewness test
##
## data:  data[[i]]
## skew = -0.5565, z = -5.2280, p-value = 1.714e-07
## alternative hypothesis: data have a skewness
##
##
##  Anscombe-Glynn kurtosis test
##
## data:  data[[i]]
## kurt = 5.9830, z = 6.8069, p-value = 9.973e-12
## alternative hypothesis: kurtosis is not equal to 3
##
## [1] "logGE"
##
##  One Sample t-test
##
## data:  data[, i]
## t = 2.7473, df = 587, p-value = 0.006193
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.002245946 0.013508527
## sample estimates:
##   mean of x
## 0.007877237
```

```
##
##
##  D'Agostino skewness test
##
## data:  data[[i]]
## skew = -0.30106, z = -2.95499, p-value = 0.003127
## alternative hypothesis: data have a skewness
##
##
##  Anscombe-Glynn kurtosis test
##
## data:  data[[i]]
## kurt = 4.8222, z = 5.2573, p-value = 1.462e-07
## alternative hypothesis: kurtosis is not equal to 3
##
## [1] "logIBM"
##
##  One Sample t-test
##
## data:  data[, i]
## t = 2.8135, df = 587, p-value = 0.005065
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.002429429 0.013663617
## sample estimates:
##   mean of x
## 0.008046523
##
##
##  D'Agostino skewness test
##
## data:  data[[i]]
## skew = -0.17286, z = -1.72021, p-value = 0.08539
## alternative hypothesis: data have a skewness
##
##
##  Anscombe-Glynn kurtosis test
##
## data:  data[[i]]
## kurt = 4.9032, z = 5.3885, p-value = 7.105e-08
## alternative hypothesis: kurtosis is not equal to 3
##
## [1] "logVWRET"
##
##  One Sample t-test
##
## data:  data[, i]
## t = 4.1232, df = 587, p-value = 4.275e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.004049061 0.011415117
## sample estimates:
##   mean of x
## 0.007732089
```

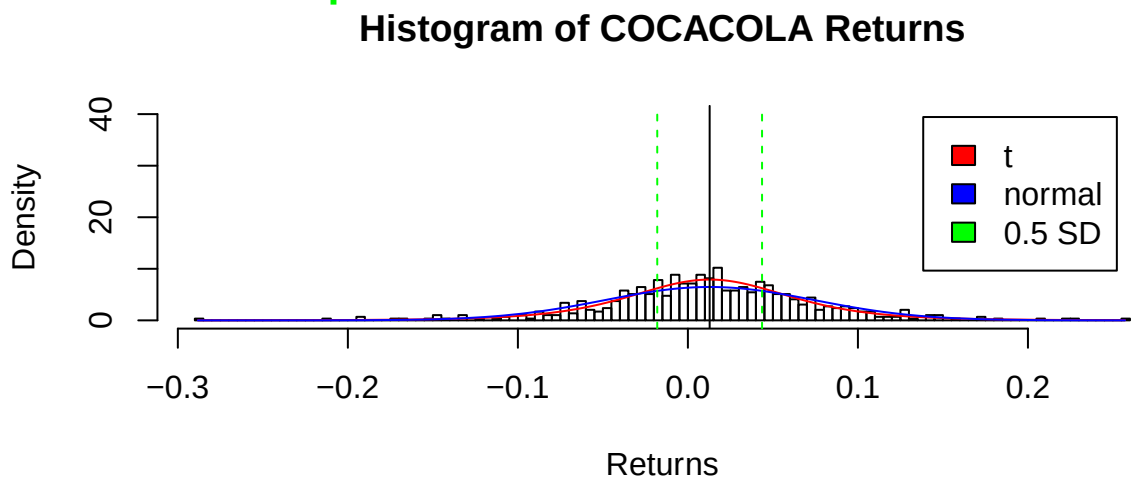
```
##
##
##  D'Agostino skewness test
##
## data:  data[[i]]
## skew = -0.80145, z = -7.13082, p-value = 9.976e-13
## alternative hypothesis: data have a skewness
##
##
##  Anscombe-Glynn kurtosis test
##
## data:  data[[i]]
## kurt = 5.8782, z = 6.6908, p-value = 2.22e-11
## alternative hypothesis: kurtosis is not equal to 3
##
## [1] "logEWRET"
##
##  One Sample t-test
##
## data:  data[, i]
## t = 4.276, df = 587, p-value = 2.221e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.005481622 0.014794833
## sample estimates:
##  mean of x
## 0.01013823
##
##
##  D'Agostino skewness test
##
## data:  data[[i]]
## skew = -0.61515, z = -5.70864, p-value = 1.139e-08
## alternative hypothesis: data have a skewness
##
##
##  Anscombe-Glynn kurtosis test
##
## data:  data[[i]]
## kurt = 6.4685, z = 7.3003, p-value = 2.872e-13
## alternative hypothesis: kurtosis is not equal to 3
```

For all log series, the mean is statistically significant from 0 at the 5% significance level. Whether the skewness is statistically significantly different from 0 and the kurtosis from 3, can be tested with the D'Agostino skewness test and the Anscombe-Glynn kurtosis test, respectively. For COCACOLA and GE, we can reject the null hypotheses of no skewness and no excess kurtosis at the 1% significance level. For IBM, we can reject the hypothesis of no skewness at the 10% significance level and the hypothesis of no excess kurtosis at the 1% significance level. Apparently, the returns of logIBM are less skewed. For both stock indices, we can reject the hypotheses of no skewness and no excess kurtosis at the 1% significance level. So, overall, the return series are skewed and show excess kurtosis, indicating non-normal distributions. This also confirms the observations made in the previous item.

(g)

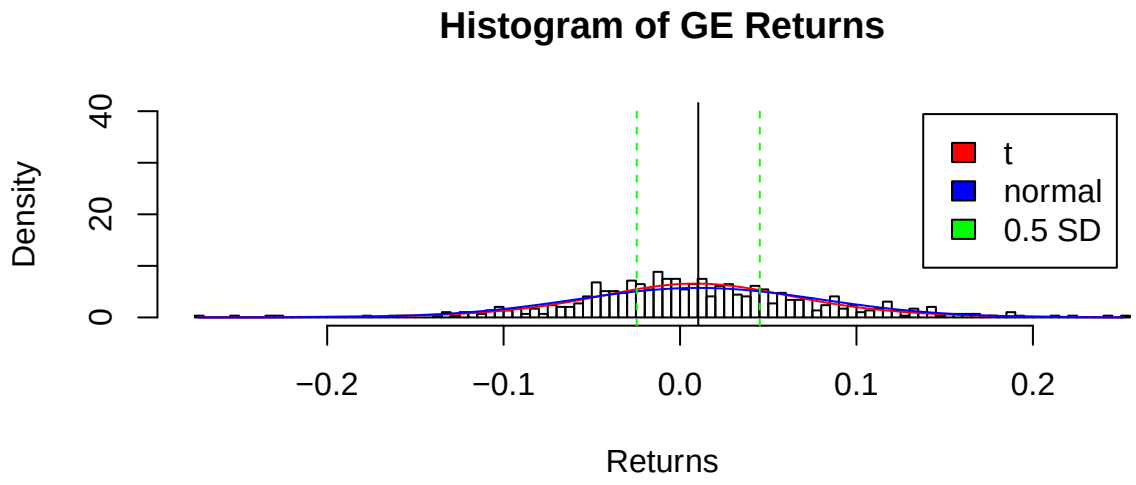
```
for (i in 2:ncol(data)) {
  xvals = sort(data[[i]])
  tfit = fit.st(xvals)
  tpars = tfit$par.ests
  nu = tpars[1]
  mu = tpars[2]
  sigma = tpars[3]

  dtvals = dt((xvals - mu)/sigma, df = nu)/sigma
  normvals = dnorm(xvals, mean(xvals), sd(xvals))

  hist(xvals, nclass = 100, probability = TRUE, ylim = range(0, 40),
    main = paste("Histogram of", names(data)[i], "Returns"),
    xlab = "Returns", ylab = "Density")
  lines(xvals, dtvals, col = "red")
  lines(xvals, normvals, col = "blue")
  legend("topright", inset = 0.05, legend = c("t", "normal", "0.5 SD"),
    fill = c("red", "blue", "green"), horiz = FALSE)
  abline(v = mean(xvals))
  abline(v = mean(xvals) - 0.5 * sd(xvals), col = "green", lty = 2)
  abline(v = mean(xvals) + 0.5 * sd(xvals), col = "green", lty = 2)

  print(names(data)[i])
  print(jarque.test(xvals))
}
```

the normal distribution has to be peakier than the t-distribution

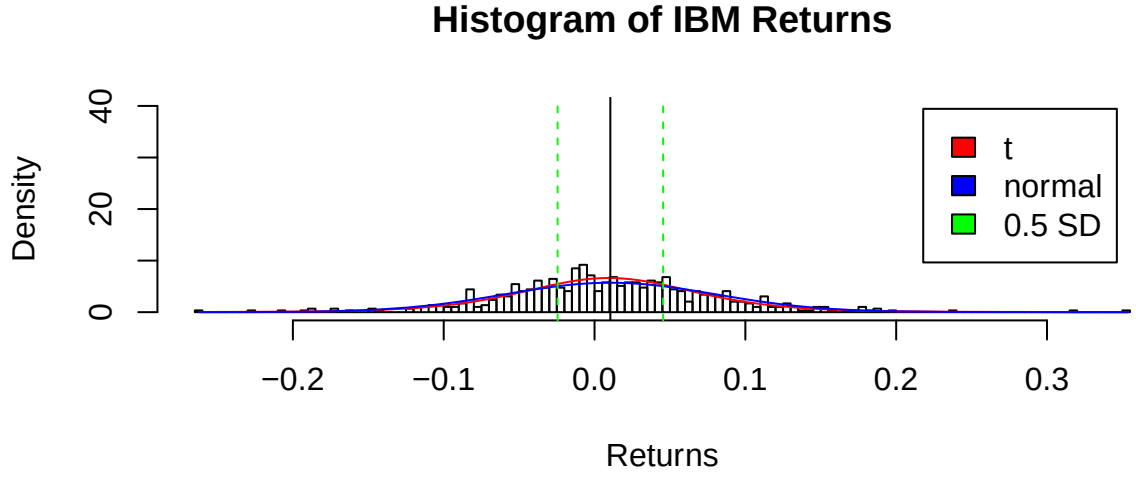



```
## [1] "COCACOLA"
##
##  Jarque-Bera Normality Test
##
## data:  xvals
```

```
## JB = 115.67, p-value < 2.2e-16
## alternative hypothesis: greater
```

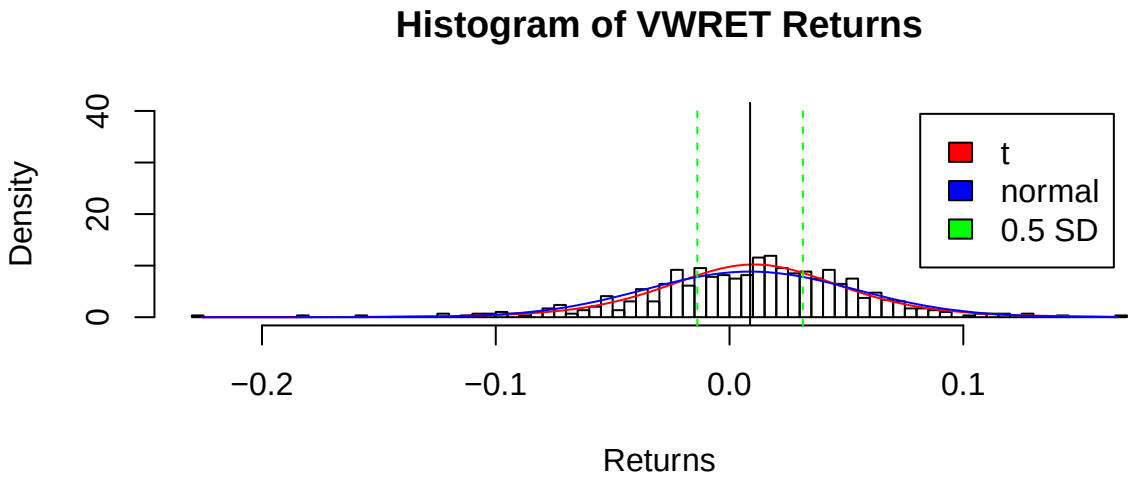



```
## [1] "GE"
##
##  Jarque-Bera Normality Test
##
## data:  xvals
## JB = 37.488, p-value = 7.238e-09
## alternative hypothesis: greater
```

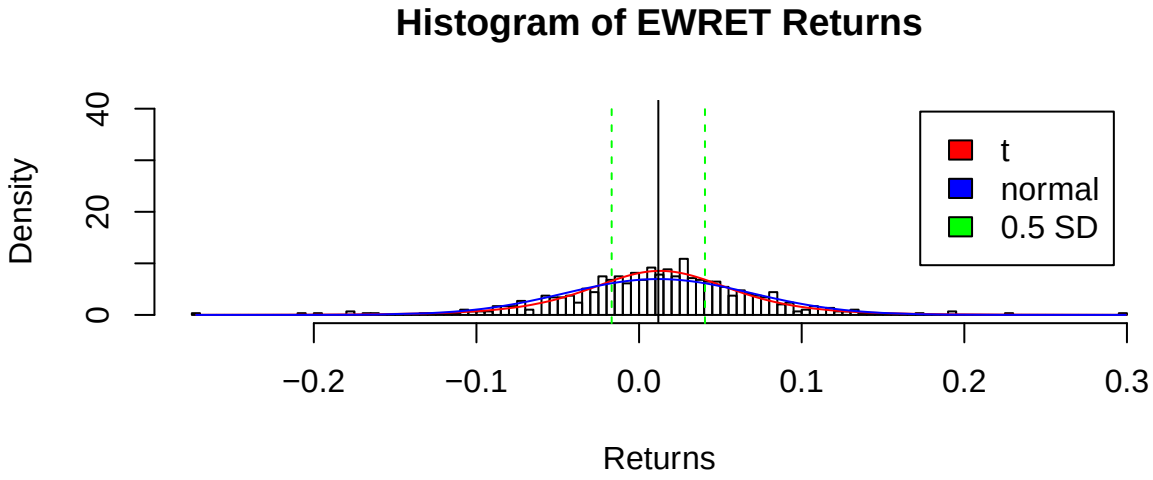



```
## [1] "IBM"
##
##  Jarque-Bera Normality Test
```

```
##
## data:  xvals
## JB = 102.37, p-value < 2.2e-16
## alternative hypothesis: greater
```

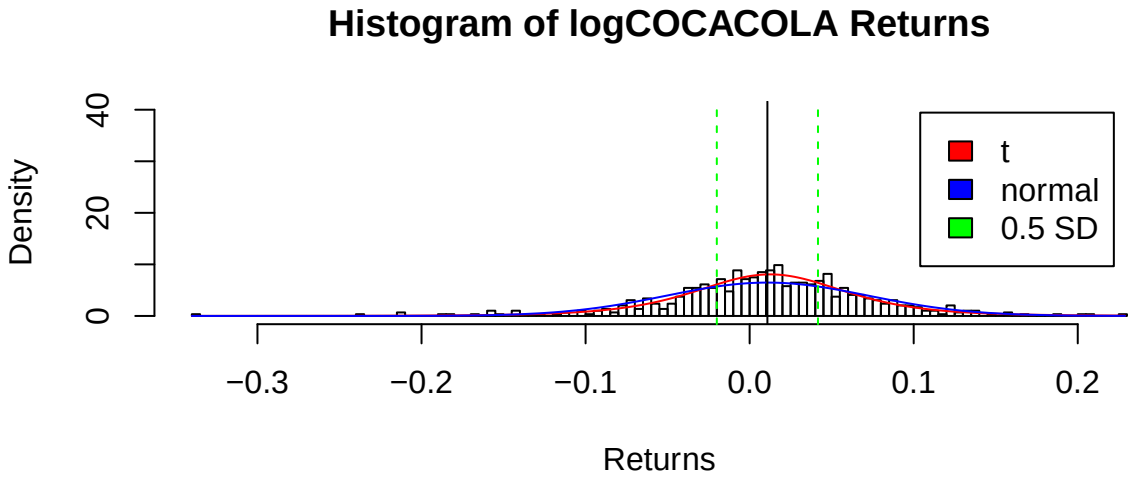



```
## [1] "VWRET"
##
##  Jarque-Bera Normality Test
##
## data:  xvals
## JB = 124.2, p-value < 2.2e-16
## alternative hypothesis: greater
```

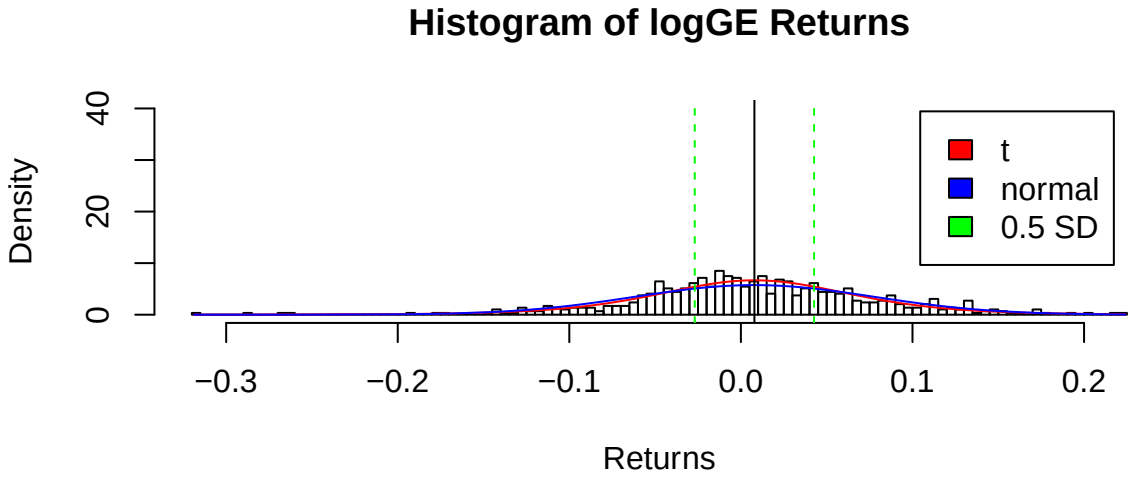



```
## [1] "EWRET"
```

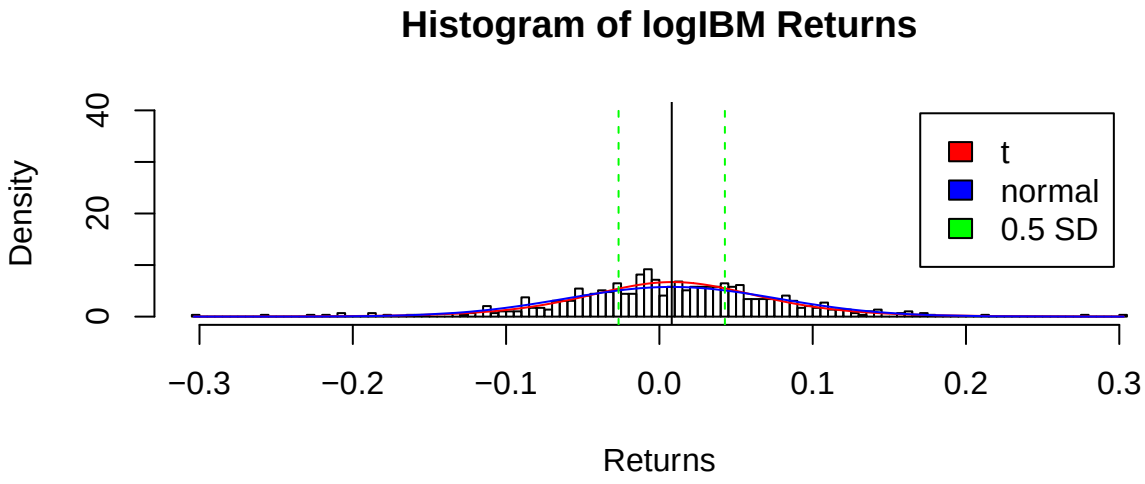
```
##
##  Jarque-Bera Normality Test
##
## data:  xvals
## JB = 202.95, p-value < 2.2e-16
## alternative hypothesis: greater
```



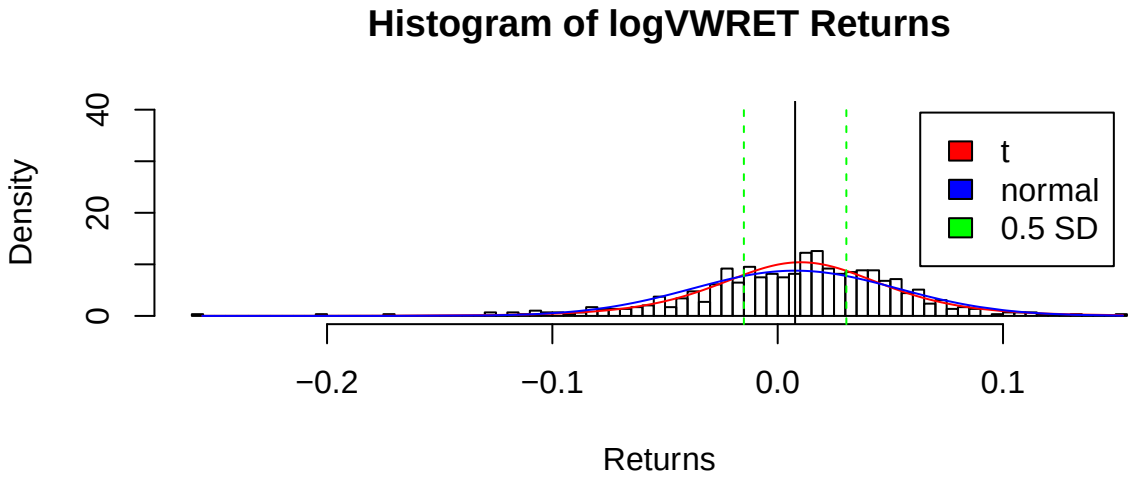

```
## [1] "logCOCACOLA"
##
##  Jarque-Bera Normality Test
##
## data:  xvals
## JB = 248.35, p-value < 2.2e-16
## alternative hypothesis: greater
```



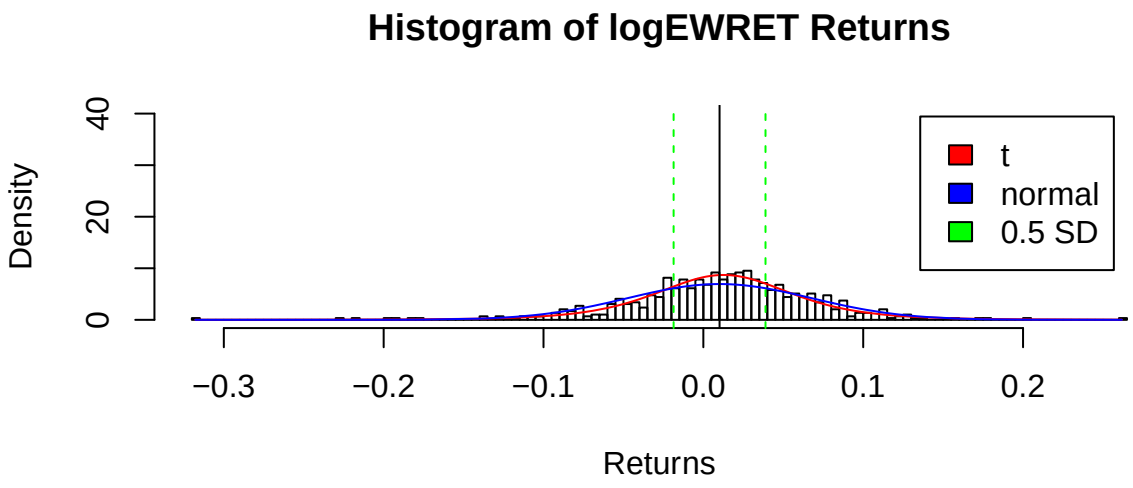
```
## [1] "logGE"
##
##  Jarque-Bera Normality Test
##
## data:  xvals
## JB = 90.23, p-value < 2.2e-16
## alternative hypothesis: greater
```




```
## [1] "logIBM"
##
##  Jarque-Bera Normality Test
##
## data:  xvals
## JB = 91.674, p-value < 2.2e-16
## alternative hypothesis: greater
```



```
## [1] "logVWRET"
##
##  Jarque-Bera Normality Test
##
## data:  xvals
## JB = 265.91, p-value < 2.2e-16
## alternative hypothesis: greater
```




```
## [1] "logEWRET"
##
##  Jarque-Bera Normality Test
##
## data:  xvals
## JB = 331.83, p-value < 2.2e-16
## alternative hypothesis: greater
```

the Shapiro-Wilk test is better than the Jarque-Bera test

At first glance, the histograms of the different actual and fitted distributions look mostly normal, in that they are unimodal and to a degree "peaky". One can see that for every return series, the t-distribution is more peaky around the mean and dips below the normal distribution at around 0.5 SDs. Though not quite discernible, the t-distribution is expected to have fatter tails than the normal distribution. The outliers, however, likely render the distributions non-normal. One can also see a negative skew in every distribution. This can also be seen at the intersections at the 0.5 SD vertical lines. In terms of kurtosis, the distributions exhibit extreme data points at the tail ends, where we would not expect such values according to a normal distribution. The t-distribution is more appropriate in this regard. When one increases the number of bins to 100, it becomes apparent that the t-distribution does a better job at describing the data than the normal distribution. Comparing the log stock returns to the log index returns, the index returns look closer to the t-distribution than the stock returns. In the Jarque-Bera-test, the null hypothesis is a joint hypothesis of the skewness being 0 and the excess kurtosis being 0. If the statistic is far from 0, it signals the data do not resemble a normal distribution. For every return series, the JB statistic is very large and we can reject the null hypothesis at the 1% significance level in each case. This means that the skewness and kurtosis values are (combined) non-normal for each distribution.

(h)

```
for (i in (2:4)) {
  print(names(data)[i])
  lm = lm(data[[i]] ~ VWRET, data)
  print(summary(lm))
}
```

```
## [1] "COCACOLA"
##
## Call:
## lm(formula = data[[i]] ~ VWRET, data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.216102 -0.029532 -0.001404  0.030982  0.171847
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.006496   0.002206   2.944  0.00336 **
## VWRET       0.716446   0.048027  14.918  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05251 on 586 degrees of freedom
## Multiple R-squared:  0.2752, Adjusted R-squared:  0.274
## F-statistic: 222.5 on 1 and 586 DF,  p-value: < 2.2e-16
##
## [1] "GE"
##
## Call:
## lm(formula = data[[i]] ~ VWRET, data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.174175 -0.030702 -0.003161  0.029411  0.198822
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0004826  0.0020217   0.239    0.811
## VWRET       1.1199868  0.0440096  25.449   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04812 on 586 degrees of freedom
## Multiple R-squared:  0.525,  Adjusted R-squared:  0.5242
## F-statistic: 647.6 on 1 and 586 DF,  p-value: < 2.2e-16
##
## [1] "IBM"
##
## Call:
## lm(formula = data[[i]] ~ VWRET, data = data)
##
```

```
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.280296 -0.031656 -0.001447  0.028898  0.285129
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.002762   0.002421   1.141    0.254
## VWRET       0.879652   0.052696  16.693   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05762 on 586 degrees of freedom
## Multiple R-squared:  0.3223, Adjusted R-squared:  0.3211
## F-statistic: 278.7 on 1 and 586 DF,  p-value: < 2.2e-16
```

```
for (i in (7:9)) {
  print(names(data)[i])
  lm = lm(data[[i]] ~ logVWRET, data)
  print(summary(lm))
}
```

```
## [1] "logCOCACOLA"
##
## Call:
## lm(formula = data[[i]] ~ logVWRET, data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.262378 -0.028580 -0.000216  0.031576  0.152872
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.005318   0.002196   2.422   0.0157 *
## logVWRET    0.713470   0.047642  14.976   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05249 on 586 degrees of freedom
## Multiple R-squared:  0.2768, Adjusted R-squared:  0.2756
## F-statistic: 224.3 on 1 and 586 DF,  p-value: < 2.2e-16
##
## [1] "logGE"
##
## Call:
## lm(formula = data[[i]] ~ logVWRET, data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.200676 -0.029510 -0.001773  0.029920  0.197660
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0007123  0.0020001  -0.356    0.722
## logVWRET     1.1108939  0.0433988  25.597   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04781 on 586 degrees of freedom
## Multiple R-squared:  0.5279, Adjusted R-squared:  0.5271
## F-statistic: 655.2 on 1 and 586 DF,  p-value: < 2.2e-16
##
## [1] "logIBM"
##
## Call:
## lm(formula = data[[i]] ~ logVWRET, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32032 -0.03062  0.00005  0.02967  0.24081
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.001322   0.002385   0.554     0.58
## logVWRET    0.869711   0.051755  16.805   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05702 on 586 degrees of freedom
## Multiple R-squared:  0.3252, Adjusted R-squared:  0.324
## F-statistic: 282.4 on 1 and 586 DF,  p-value: < 2.2e-16
```

I decided to run the regression on the value-weighted index, since value-weighted indices tend to reflect better the actual value allocation in a portfolio than equally-weighted indices. For GE and IBM, we fail to reject the null hypothesis that the alpha is 0 at the 10% significance level. The values are 0.0005 and 0.0028, respectively. For COCACOLA, we find that alpha, 0.0065, is statistically significantly different from 0 at the 1% significance level. This information could be useful in an investment decision or in setting up an investment strategy. COCACOLA's beta is 0.7164, GE's is 1.1200, and IBM's is 0.8797. All are statistically significantly different from 0 at the 1% significance level. Accordingly, GE can be considered a cyclical stock and both COCACOLA and IBM counter-cyclical stocks. A similar pictures emerges from the log returns, though the coefficients and standard errors change slightly and the alpha of logCOCACOLA is now statistically significant at the 5% significance level, but no longer at the 1% significance level.

2.

```
library(haven)
library(car)
library(lmtest)
library(sandwich)
```

(a)

```
path = paste0(getwd(), "/pension.dta")
data = read_dta(path)
```

```
data[c("e401k", "marr", "male", "p401k", "pira")] =
lapply(data[c("e401k", "marr", "male", "p401k", "pira")], as.factor)

head(data)
```

```
## # A tibble: 6 x 11
##   e401k   inc marr  male    age fsize nettfa p401k pira  incsq agesq
##   <fct> <dbl> <fct> <fct> <dbl> <dbl>  <dbl> <fct> <fct> <dbl> <dbl>
## 1 0      13.2 0     0        40     1   4.57 0     1      173.  1600
## 2 1      61.2 0     1        35     1 154    1     0     3749.  1225
## 3 0      12.9 1     0        44     2   0    0     0      165.  1936
## 4 0      98.9 1     1        44     2  21.8  0     0     9777.  1936
## 5 0      22.6 0     0        53     1  18.5  0     0      511.  2809
## 6 0      15   1     0        60     3   0    0     0      225   3600
```

(b)

```
paste0("There are ", toString(sum(data$fsize == 1)),
  " single households. This is equivalent of ",
  toString(round(sum(data$fsize == 1)/nrow(data)*100)), "% of the sample.")
```

```
## [1] "There are 2017 single households. This is equivalent of 22% of the sample."
```

(c) & (d)

```
dataLM = data %>% dplyr::filter(fsize == 1)
lm = lm(nettfa ~ inc + age, dataLM)
summary(lm)
```

```
##
## Call:
## lm(formula = nettfa ~ inc + age, data = dataLM)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -179.95  -14.16   -3.42    6.03 1113.94
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -43.03981    4.08039 -10.548   <2e-16 ***
## inc           0.79932    0.05973  13.382   <2e-16 ***
## age           0.84266    0.09202   9.158   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.68 on 2014 degrees of freedom
## Multiple R-squared:  0.1193, Adjusted R-squared:  0.1185
## F-statistic: 136.5 on 2 and 2014 DF,  p-value: < 2.2e-16
```

For single-person households, the net financial wealth increases positively with both (family) income (beta of 0.7993) and age (beta of 0.8427). The intercept of -43.0398, on average, is perhaps nonsensical, though, since negative net financial wealth at age 0 and income 0 does not make much sense (unless the individual inherits a considerable amount of financial debt). This calls into question the external validity of this regression model for low age and/or low income individuals living on their own. Nonetheless, an individual of a single-person household is expected to increase her/his financial wealth by 799 USD for every additional 1,000 USD income, on average, certeris paribus. Similarly, individuals that are 1 year older are expected to have financial wealth greater by 842 USD, on average, certeris paribus. The intercept and the coefficients are statistically significant at the 1% significance level.

(e)

```
mu = coef(lm)[3]
se = sqrt(diag(vcov(lm)))[3]
pt(-abs((mu - 1)/se), lm$df)
```

```
##        age
## 0.04371514
```

We can reject the null hypothesis that the coefficient of age is equal to 1 at the 5% significance level against the one-sided alternative hypothesis that the coefficient is smaller than 1. We fail to reject the null hypothesis at the 1% significance level, though.

(f)

```
2 * pt(-abs((mu - 1)/se), lm$df)
```

```
##        age
## 0.08743028
```

Against the two-sided alternative hypothesis, we fail to reject the null hypothesis at the 5% significance level.

(g)

```
summary(lm(nettfa ~ inc, dataLM))
```

```
##
## Call:
## lm(formula = nettfa ~ inc, data = dataLM)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -185.12  -12.85   -4.85    1.78 1112.66
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -10.5709     2.0607   -5.13 3.18e-07 ***
## inc           0.8207     0.0609   13.48  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 45.59 on 2015 degrees of freedom
## Multiple R-squared:  0.08267,    Adjusted R-squared:  0.08222
## F-statistic: 181.6 on 1 and 2015 DF,  p-value: < 2.2e-16
```

Running the model on only income, the coefficient of income increases slightly from 0.7993 to 0.8207. Both the intercept and the coefficient are statistically significant at the 1% significance level. This can be explained by the very slight positive correlation (0.0391) between age and income in the restricted sample. Age having a positive coefficient in the full model and the correlation between the two variables being positive, as well, implies a positive bias on the coefficient in the restricted model. Is is questionable, whether this difference is economically significant, though. The adjusted R-squared decreases slightly, which is to be expected.

(h)

```
lm = lm(nettfa ~ inc + age + incsq + agesq + fsize, data)
summary(lm)
```

```
##
## Call:
## lm(formula = nettfa ~ inc + age + incsq + agesq + fsize, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -523.05  -15.85   -2.97    5.62 1466.00
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.9925513  9.9919763   1.400   0.1614
## inc         -0.1266417  0.0732205  -1.730   0.0837 .
## age         -1.2580134  0.4918240  -2.558   0.0105 *
## incsq        0.0094893  0.0005837  16.257  < 2e-16 ***
## agesq        0.0263771  0.0056527   4.666 3.11e-06 ***
## fsize       -1.9792085  0.4011045  -4.934 8.18e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 57.26 on 9269 degrees of freedom
## Multiple R-squared:  0.1989, Adjusted R-squared:  0.1985
## F-statistic: 460.4 on 5 and 9269 DF,  p-value: < 2.2e-16
```

In this model the intercept is not statistically significantly different from 0 at the 10% significance level. Family income is now statistically significant at the 10% significance level, but not at the 5% significance level. Age is statistically significant at the 5% significance level, but not at the 1% significance level. All new coefficients are statistically significant at the 1% significance level. If all model variables are equal to 0 the model predicts net financial wealth of 13.9926, which seems more plausible than a negative value (though the intercept is statistically insignificant). Income and age now have negative values. This can be explained by the polynomial terms. There appear to be increasing marginal returns to both income and

age. So the older or the more income an individual (or family household) has, the greater the incremental wealth for additional income and/or "additional age", on average, ceteris paribus. For income, for example, additional income starts to increase wealth starting at income of 2 * 0.0095 * x - 0.1266 = 0 <=> x = 6.6631 (6,6631 USD), on average, ceteris paribus. Finally, the coefficient of fsize is negative, -1.9792, meaning that financial wealth decreases with family size. For every additional family member, net financial wealth decreases by 1,979 USD, on average, ceteris paribus. This could seem counterintuitive, since more family members could provide more labor and hence income, but large family size tends to correlate with poverty, in general. There are other factors and explanations, of course. Here, it would also be interesting to investigate potential increasing/decreasing marginal effects. The adjusted R-squared also increases compared to model (1), though the sample changes, as well, making the comparison invalid.

(i)

```
linearHypothesis(lm ,c("incsq = 0","agesq = 0"))
```

```
## Linear hypothesis test
##
## Hypothesis:
## incsq = 0
## agesq = 0
##
## Model 1: restricted model
## Model 2: nettfa ~ inc + age + incsq + agesq + fsize
##
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1   9271 31361180
## 2   9269 30394935  2    966245 147.33 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The null hypothesis of beta3 = beta4 = 0 can be rejected at the 1% significance level. This means that both variables add statistically significantly to the model by increasing the explained variance in the model. The SSR (sum of squared residuals) is significantly larger in the reduced model than in the full model. Nevertheless, it could still be that the variables are individually statistically insignificant and only work in conjunction.

(j) convert the variables back

```
dataRE = data %>% mutate(inc = inc / 10, incsq = incsq / 100)
lm = lm(nettfa ~ inc + age + incsq + agesq + fsize, dataRE)
summary(lm)
```

```
##
## Call:
## lm(formula = nettfa ~ inc + age + incsq + agesq + fsize, data = dataRE)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -523.05  -15.85   -2.97    5.62 1466.00
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.992551   9.991976   1.400   0.1614
## inc         -1.266417   0.732205  -1.730   0.0837 .
## age         -1.258013   0.491824  -2.558   0.0105 *
## incsq        0.948932   0.058370  16.257  < 2e-16 ***
## agesq        0.026377   0.005653   4.666 3.11e-06 ***
## fsize       -1.979209   0.401105  -4.934 8.18e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 57.26 on 9269 degrees of freedom
## Multiple R-squared:  0.1989, Adjusted R-squared:  0.1985
## F-statistic: 460.4 on 5 and 9269 DF,  p-value: < 2.2e-16
```

The model stays the same, but the coefficient for income is now 10 times its previous size. The coefficient of incsq is scaled by 100 (10^2). Statistical significance stays unaffected, since the standard errors are scaled by the same factor for inc (x10) and incsq (x100), reversing the effect. The economic interpretation under consideration of the scaling also stays the same. The scaling has no material effect.

(k)

```
bptest(nettfa ~ inc + age, data = dataLM, studentize = FALSE)
```

```
##
##  Breusch-Pagan test
##
## data:  nettfa ~ inc + age
## BP = 1697.2, df = 2, p-value < 2.2e-16
```

The null hypothesis of homoskedasticity can be rejected at the 1% significance level. The coefficients are jointly statistically significant and hence there still seems to be a relationship between the squared residuals and at least one variable. We thus found evidence for heteroskedasticity.

(l)

```
lm = lm(nettfa ~ inc + age, dataLM)
coeftest(lm, vcov. = vcovHC(lm ,type = "HC0"))
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept) -43.03981    5.52022 -7.7968 1.011e-14 ***
## inc           0.79932    0.10070  7.9372 3.401e-15 ***
## age           0.84266    0.11929  7.0642 2.217e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Including robust standard errors does not change the estimates of the coefficients, but only increases the standard errors, for example from 0.0597 to 0.1007 for income. Thereby, the t-values become smaller, but the intercept and the coefficients are still statistically significant at the 1% significance level.

(m)

```
zScore = function(x) {return((x-mean(x))/sd(x))}
lm = lm(zScore(nettfa) ~ zScore(inc) + zScore(age), dataLM)
summary(lm)
```

```
##
## Call:
## lm(formula = zScore(nettfa) ~ zScore(inc) + zScore(age), data = dataLM)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7811 -0.2975 -0.0718  0.1267 23.4068
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.102e-17  2.091e-02   0.000        1
## zScore(inc)  2.800e-01  2.093e-02  13.382   <2e-16 ***
## zScore(age)  1.916e-01  2.093e-02   9.158   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9389 on 2014 degrees of freedom
## Multiple R-squared:  0.1193, Adjusted R-squared:  0.1185
## F-statistic: 136.5 on 2 and 2014 DF,  p-value: < 2.2e-16
```

The intercept is effectively 0, as expected. The intercept is, of course, not statistically significantly different from 0, but the coefficients still are, at the 1% significance level. Both coefficients have become tremendously small. If income increases by 1 standard deviation, then net financial wealth increases by 0.2800 standard deviations, ceteris paribus. Similarly, if age increases by 1 standard deviation, net financial wealth increases by 0.1916 standard deviations, ceteris paribus. One unit is now a standard deviation.

3.

```
library(broom)
```

(a)

```
path = paste0(getwd(), "/ceo_salary.dta")
data = read_dta(path)
data[c("college", "grad")] = lapply(data[c("college", "grad")], as.factor)

head(data)
```

```
## # A tibble: 6 x 15
##   salary   age college grad  comten ceoten sales profits mktval lsalary lsales
##    <dbl> <dbl> <fct>   <fct>  <dbl>  <dbl> <dbl>   <dbl>  <dbl>   <dbl>  <dbl>
## 1   1161    49 1       1          9      2  6200     966  23200    7.06   8.73
## 2    600    43 1       1         10     10   283      48   1100    6.40   5.65
## 3    379    51 1       1          9      3   169      40   1100    5.94   5.13
## 4    651    55 1       0         22     22  1100     -54   1000    6.48   7.00
## 5    497    44 1       1          8      6   351      28    387    6.21   5.86
## 6   1067    64 1       1          7      7 19000     614   3900    6.97   9.85
## # ... with 4 more variables: lmktval <dbl>, comtensq <dbl>, ceotensq <dbl>,
## #   profmarg <dbl>
```

```
lm = lm(lsalary ~ lsales + lmktval + ceoten + ceotensq, data)
summary(lm)
```

```
##
## Call:
## lm(formula = lsalary ~ lsales + lmktval + ceoten + ceotensq,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.41976 -0.28791  0.00253  0.28615  1.74966
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.3685503  0.2587397  16.884  < 2e-16 ***
## lsales       0.1646331  0.0386393   4.261 3.35e-05 ***
## lmktval      0.1085285  0.0488257   2.223  0.02753 *
## ceoten       0.0451169  0.0141169   3.196  0.00166 **
## ceotensq    -0.0012102  0.0004747  -2.549  0.01167 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4969 on 172 degrees of freedom
## Multiple R-squared:  0.343,  Adjusted R-squared:  0.3277
## F-statistic: 22.45 on 4 and 172 DF,  p-value: 6.257e-15
```

The intercept and all coefficients are statically significant at the 5% significance level. lsalary increases with lsales, lmktval, and ceoten, on average, ceteris paribus. The coefficient of ceotensq is negative, indicating decreasing marginal returns to ceoten.

(b)

```
descriptiveStats =
  describe(data[c("sales", "mktval", "lsales", "lmktval", "ceoten", "ceotensq")])
descriptiveStats
```

```
##          vars   n    mean      sd  median trimmed     mad    min      max
## sales       1 177 3529.46 6088.65 1400.00 2179.81 1577.49  29.00 51300.00
## mktval      2 177 3600.32 6442.28 1200.00 2029.79 1040.79 387.00 45400.00
```

```
## lsales      3 177    7.23    1.43    7.24    7.25    1.36   3.37    10.85
## lmktval     4 177    7.40    1.13    7.09    7.27    1.15   5.96    10.72
## ceoten      5 177    7.95    7.15    6.00    6.82    5.93   0.00    37.00
## ceotensq    6 177  114.12  212.57   36.00   64.22   47.44   0.00  1369.00
##              range  skew kurtosis     se
## sales    51271.00  4.14    23.27 457.65
## mktval   45013.00  3.85    17.84 484.23
## lsales       7.48 -0.10    -0.21   0.11
## lmktval      4.76  0.84    -0.06   0.09
## ceoten      37.00  1.63     3.02   0.54
## ceotensq  1369.00  3.63    15.69  15.98
```

Taking the natural log can mitigate or eliminate problems arising from strictly positive variables that have heteroskedastic or skewed conditional distributions. Also, narrowing the range of the dependent and independent variables can make OLS estimates less sensitive to outliers. A common rules of thumb is that when a variable is a positive currency amount, take the log. That is also the case for lsales and lmktval, both of which are quoted in a currency. To compare, I have included also the variables without the log: sales and mktval. We can see that the variables without the logs exhibit strikingly larger skewness (4.14 vs -0.10 and 3.85 vs 0.84) and kurtosis (23.27 vs -0.21 and 17.84 vs -0.06) than the log variables. Thus, their distributions are becoming closer to a normal distribution, making them more viable for statistical inference. In addition, the original variables have larger means, larger standard deviations, and a larger range.

(c)

```
coeftest(lm, vcov. = vcovHC(lm ,type = "HC0"))
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  4.36855032  0.26010837 16.7951 < 2.2e-16 ***
## lsales       0.16463314  0.03791554  4.3421 2.404e-05 ***
## lmktval      0.10852852  0.04877684  2.2250  0.027382 *
## ceoten       0.04511688  0.01412824  3.1934  0.001672 **
## ceotensq    -0.00121019  0.00054608 -2.2161  0.027994 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The White standard errors are more conservative and thus larger than normal standard errors. Hence, the t-statistics are expected to decrease in absolute value for every coefficient and the intercept. Apparently, for the log variables, lsales and lmktval, that is not the case. Their standard errors decrease and the t-statistics increase for White standard errors.

(d)

```
fullLM = augment(lm)
fullLM %>% dplyr::filter(abs(.std.resid) > 1.96) %>% count
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1     9
```

should be 8, not 9 => what went wrong?

```
177* (2 * (1 - pnorm(2, 0, 1)))
```

```
## [1] 8.053547
```

There are 9 standardized residuals, whose absolute value is greater-equal to 1.96. If the standardized residuals were i.i.d draws from a standard normal distribution, we would expect about 8 our of 177 to be above 2 in absolute value. This can be considered slight evidence against normally distributed standardized residuals.

(e)

```
lm = lm(lsalary ~ lsales + lmktval + ceoten + ceotensq + college + college:lsales, data)
summary(lm)
```

```
##
## Call:
## lm(formula = lsalary ~ lsales + lmktval + ceoten + ceotensq +
##     college + college:lsales, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.45977 -0.29501  0.00112  0.28871  1.74673
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      7.9442688  1.7085885   4.650 6.65e-06 ***
## lsales          -0.2999761  0.2205561  -1.360  0.17560
## lmktval          0.1179878  0.0486657   2.424  0.01638 *
## ceoten           0.0452778  0.0140763   3.217  0.00155 **
## ceotensq        -0.0011771  0.0004723  -2.492  0.01365 *
## college1        -3.6618822  1.7267023  -2.121  0.03539 *
## lsales:college1  0.4661815  0.2179206   2.139  0.03384 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4933 on 170 degrees of freedom
## Multiple R-squared:  0.3602, Adjusted R-squared:  0.3376
## F-statistic: 15.95 on 6 and 170 DF,  p-value: 1.652e-14
```

The intercept and all coefficients, but the one for lsales are statistically significant at the 5% significance level. According to the intercept, a CEO with no sales, leading a company with no market value, with no tenure, who didn't go to college earns exp(7.9443) = 2819.4580 in the respective currency, on average. The salary increases by 0.1180% when the market value increases by 1%, on average, ceteris paribus. Bigger companies pay higher salaries. As mentioned earlier, there are decreasing marginal effect to ceoten. We cannot tell the specific marginal effect, since it is non-constant. We can, however, mention the first derivative with respect to ceoten, which equals 0.0453 - 0.0024 * ceoten. This terms turns negative for ceoten >= 18.8750. It is

questionable whether it makes economic sense that salary decreases marginally with tenure. Whether a ceo has gone to college has a differential flat impact on salary. The intercept for college goers becomes 7.9444 - 3.6619 = 4.2825, while it stays at 7.9444 for non-college-goers. Having gone to college decreases the salary 366.1882%, on average, ceteris paribus. This finding is surprising to me. Looking at the data more closely, however, I find that 172 of the 177 CEOs went to college. The mean lsalary among those CEOs is 6.5761, while it is 6.8134 among the non-college-goers. Thus, the sample is unbalanced in this regard, making the (external) validity of this finding questionable. Finally, the coefficient of the interaction term is positive. This means that the coefficient on lsales is -0.3000 + 0.4662 = 0.1662 for CEOs who went to college and still -0.3000 for non-college CEOs. For CEOs who went to college, a 1% increase in sales increases the salary by 0.1662%. For CEOs who didn't go to college, a 1% increase in sales decreases the salary by 0.3000%, though this coefficient is insignificant. Additionally, it would not make much sense that the salary decreases with sales.

(f)

```
dataAD = data %>% mutate(lsales_adjusted = 0.9 * lsales)

lm = lm(lsalary ~ lsales_adjusted + lmktval + ceoten + ceotensq, dataAD)
summary(lm)
```

```
##
## Call:
## lm(formula = lsalary ~ lsales_adjusted + lmktval + ceoten + ceotensq,
##     data = dataAD)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.41976 -0.28791  0.00253  0.28615  1.74966
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.3685503  0.2587397  16.884  < 2e-16 ***
## lsales_adjusted  0.1829257  0.0429326   4.261 3.35e-05 ***
## lmktval          0.1085285  0.0488257   2.223  0.02753 *
## ceoten           0.0451169  0.0141169   3.196  0.00166 **
## ceotensq        -0.0012102  0.0004747  -2.549  0.01167 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4969 on 172 degrees of freedom
## Multiple R-squared:  0.343,  Adjusted R-squared:  0.3277
## F-statistic: 22.45 on 4 and 172 DF,  p-value: 6.257e-15
```

The new coefficient is 0.1829. Multiplying it by the scaling factor 0.9 returns the old coefficient, 0.1646. The standard error increases proportionally and the t-statistic, logically, stays the same. This is to be expected, since the scaling does nothing to the estimation precision. The other coefficients and statistics are unaffected, as well. This shows that when the measurement error is systematic (the bias is constant over all observations), one can correct for it with an appropriate scaling factor. The model now predicts a higher percentage change in salary as a response to a 1%-increase in sales. The precision stay the same, though.

(g)

```
dataRE = fullLM %>% dplyr::filter(abs(.std.resid) < 1.96)

lm = lm(lsalary ~ lsales + lmktval + ceoten + ceotensq, dataRE)
summary(lm)
```

```
##
## Call:
## lm(formula = lsalary ~ lsales + lmktval + ceoten + ceotensq,
##     data = dataRE)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.84325 -0.27539  0.00062  0.28180  0.88039
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.1381178  0.2038259  20.302  < 2e-16 ***
## lsales       0.1538080  0.0301393   5.103 9.21e-07 ***
## lmktval      0.1534926  0.0378702   4.053 7.81e-05 ***
## ceoten       0.0359998  0.0115910   3.106  0.00224 **
## ceotensq    -0.0008125  0.0004160  -1.953  0.05250 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3788 on 163 degrees of freedom
## Multiple R-squared:  0.5041, Adjusted R-squared:  0.492
## F-statistic: 41.43 on 4 and 163 DF,  p-value: < 2.2e-16
```

9 observations were dropped. ceotensq is now no longer statistically significant at the 5% significance level, while lmktval is now statistically significant at the 1% significance level. While the intercept and the coefficients show noticeable changes, the R-squared exhibits the most considerable change. The adjusted R-squared value increases from 0.3277 to 0.4920. Seemingly, the 9 dropped observations were outliers, since they deviated strongly from the regression line. As we can see, they had a negative effect on the model accuracy. It would be interesting to see what variable(s) made them outliers and whether they were truly outliers or just naturally extreme observations.

4.

```
library(plm)
library(lfe)
library(tseries)
```

(a)

```
path = paste0(getwd(), "/local_returns.dta")
data = read_dta(path)
data[c("city", "ind", "date", "year", "permno")] =
  lapply(data[c("city", "ind", "date", "year", "permno")], as.factor)
data = pdata.frame(data, c("permno", "date"))
head(data)
```

```
##           permno year         ret city ind date city_returns     MRP     HML
## 10001-480  10001 2000 -0.04411765    4   8  480 -0.024553183 -0.0474  0.0026
## 10001-481  10001 2000  0.01538462    4   8  481  0.045873884  0.0245 -0.1261
## 10001-482  10001 2000 -0.01575758    4   8  482  0.039057303  0.0520  0.0765
## 10001-483  10001 2000  0.01171875    4   8  483 -0.009810019 -0.0640  0.0909
## 10001-484  10001 2000 -0.02316602    4   8  484 -0.014000371 -0.0442  0.0372
## 10001-485  10001 2000  0.02766798    4   8  485  0.017856063  0.0464 -0.1002
##               RF     MOM      indret
## 10001-480 0.0041  0.0188  0.06220017
## 10001-481 0.0043  0.1838 -0.04722921
## 10001-482 0.0047 -0.0680  0.04546719
## 10001-483 0.0046 -0.0852  0.06599442
## 10001-484 0.0050 -0.0906  0.01546620
## 10001-485 0.0040  0.1649 -0.04741758
```

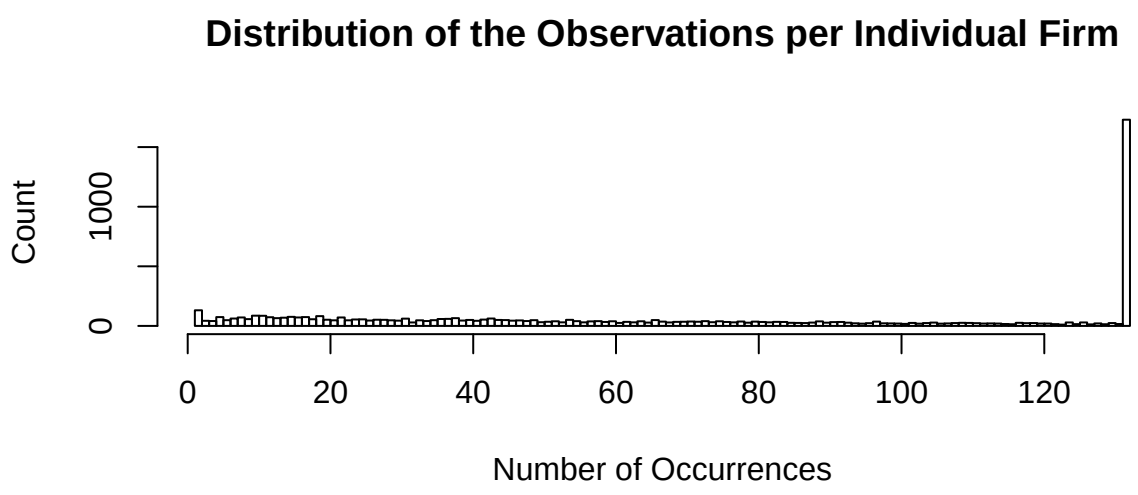
```
pdim(data)
```

```
## Unbalanced Panel: n = 6783, T = 1-132, N = 487582
```

```
# Balance among permnos
data %>% count(permno) %>%
  summarize(min = min(n), max = max(n), sd = SD(n), shareMax = mean(n == max(n)))
```

```
##   min max      sd  shareMax
## 1   1 132 46.8766 0.2550494
```

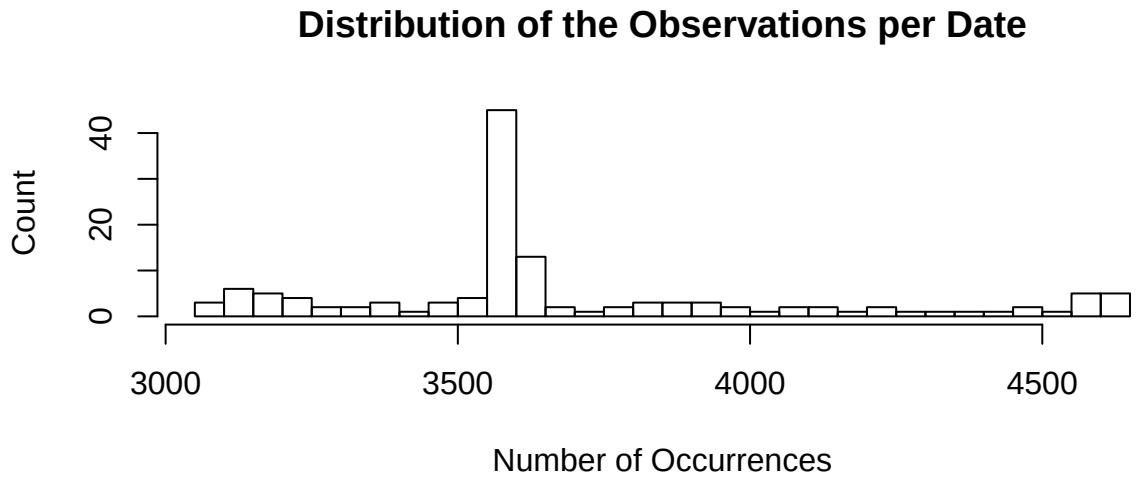
```
n = data %>% count(permno) %>% select(n)
n = as.numeric(n[[1]])
hist(n, breaks = 132, main = paste("Distribution of the Observations per Individual Firm"),
  xlab = "Number of Occurrences", ylab = "Count")
```




```
# Balance among date
data %>% count(date) %>%
  summarize(min = min(n), max = max(n), sd = SD(n), shareMax = mean(n == max(n)))
```

```
##    min  max       sd    shareMax
## 1 3073 4642 402.8351 0.007575758
```

```
n = data %>% count(date) %>% select(n)
n = as.numeric(n[[1]])
hist(n, breaks = 30, main = paste("Distribution of the Observations per Date"),
  xlab = "Number of Occurrences", ylab = "Count")
```




```
data %>% count(year)
```

```
##    year     n
```

```
## 1  2000 55099
## 2  2001 50860
## 3  2002 46862
## 4  2003 43771
## 5  2004 42992
## 6  2005 43272
## 7  2006 43059
## 8  2007 42941
## 9  2008 42011
## 10 2009 39200
## 11 2010 37515
```

```
# Balance among cities
data %>% count(city) %>% summarize(min = min(n), max = max(n), sd = SD(n))
```

```
##    min   max       sd
## 1 3598 91399 20858.89
```

```
data %>% count(city)
```

```
##    city     n
## 1     1 19626
## 2     2 41574
## 3     3 39269
## 4     4  9437
## 5     5 24336
## 6     6 14437
## 7     7  9822
## 8     8 25303
## 9     9  5550
## 10   10 40527
## 11   11 14683
## 12   12 15790
## 13   13 91399
## 14   14  3598
## 15   15 32908
## 16   16  7781
## 17   17 49044
## 18   18  9599
## 19   19  6844
## 20   20 26055
```

```
# Balance among industries
data %>% count(ind) %>% summarize(min = min(n), max = max(n), sd = SD(n))
```

```
##    min    max      sd
## 1 7262 144480 41665.4
```

```
data %>% count(ind)
```

```
##    ind      n
```

```
## 1    1  18835
## 2    2   7698
## 3    3  32621
## 4    4  15687
## 5    5   9269
## 6    6  94846
## 7    7  13409
## 8    8   7262
## 9    9  38427
## 10  10  52124
## 11  11 144480
## 12  12  52924
```

According to the bdim function from the plm package, the panel is unbalanced. To go a little further, we can look at two dispersion metrics, range and standard deviation, of the dimensions. First, looking at the distribution of the number of observations per individual firms, we see that the number differs widely from a minimum of 1 to a maximum of 132 with a standard deviation of 46.8766. Looking at the histogram, we see a mode at 132 observations; 25.5049% of the observations have this number of observations. On the whole, the panel looks far from balanced when it comes to the number of observations per individual firm. The panel looks slightly more balanced when we look at the number of observations per date. Still, with a minimum of 3,073, a maximum of 4,642, and a standard deviation of 402, the data exhibits quite a degree of dispersion. In the histogram, we can also see that the count per date peaks at certain dates, serving as evidence against balance. We can also see that the number of observations per year decrease monotonously at an almost constant rate. This might be a sign of attrition, a phenomenon that occurs when you only observe a subset in subsequent periods. This leads to selection bias of unknown nature if the data is not missing at random and the selection is based on unobservables. A similar picture emerges when we look at the balance across cities and industries. Imbalance here is to be expected though, since there are larger and smaller cities and industries.

(b)

```
data %>% group_by(city) %>% summarize(mean = mean(ret), sd = SD(ret))
```

```
## # A tibble: 20 x 3
##    city     mean    sd
##    <fct>   <dbl> <dbl>
##  1 1     0.00721 0.197
##  2 2     0.00942 0.201
##  3 3     0.00797 0.144
##  4 4     0.0129  0.162
##  5 5     0.0110  0.195
##  6 6     0.0104  0.208
##  7 7     0.00967 0.198
##  8 8     0.0165  0.208
##  9 9     0.0105  0.162
## 10 10    0.00922 0.224
## 11 11    0.00586 0.231
## 12 12    0.0129  0.189
## 13 13    0.00849 0.194
## 14 14    0.00633 0.235
## 15 15    0.00872 0.163
```

```
## 16 16    0.0116  0.213
## 17 17    0.00779 0.240
## 18 18    0.00893 0.236
## 19 19    0.0155  0.180
## 20 20    0.0108  0.219
```

```
data %>% group_by(city) %>% summarize(mean = mean(ret), sd = SD(ret)) %>%
  summarize(min(mean), max(mean), sd(mean))
```

```
## # A tibble: 1 x 3
##   `min(mean)` `max(mean)` `sd(mean)`
##         <dbl>       <dbl>      <dbl>
## 1     0.00586      0.0165    0.00278
```

```
data %>% group_by(city) %>% summarize(mean = mean(ret), sd = SD(ret)) %>%
  summarize(min(sd), max(sd), sd(sd))
```

```
## # A tibble: 1 x 3
##   `min(sd)` `max(sd)` `sd(sd)`
##       <dbl>     <dbl>    <dbl>
## 1     0.144     0.240   0.0275
```

Among the 20 cities the average monthly firm returns differ widely. The minimum is 0.0059 in city 11 and the maximum is 0.0165 in city 8. The standard deviation of mean returns is quite considerable at 0.0028. The standard deviations also differ from a minimum of 0.1444 in city 3 to a maximum of 0.2401 in city 17 with a standard deviation of standard deviations of 0.0275. Strikingly, the minimum and maximum returns do not occur in the cities with the lowest and highest standard deviations, respectively. On a preliminary basis, this implies a violation of market efficiency (in terms of Sharpe ratio), though further investigations are needed.

(c) & (d)

```
pooledLM = plm(ret ~ city_returns + indret, model = "pooling", data)
summary(pooledLM)
```

```
## Pooling Model
##
## Call:
## plm(formula = ret ~ city_returns + indret, data = data, model = "pooling")
##
## Unbalanced Panel: n = 6783, T = 1-132, N = 487582
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -1.1929686 -0.0739028 -0.0067361  0.0576988 15.6428882
##
## Coefficients:
##                 Estimate  Std. Error  t-value  Pr(>|t|)
## (Intercept)  -0.00145716  0.00026866  -5.4239 5.834e-08 ***
## city_returns  0.21216676  0.00516315  41.0925 < 2.2e-16 ***
```

```
## indret        0.85583696  0.00498702 171.6127 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    19778
## Residual Sum of Squares: 16807
## R-Squared:      0.15022
## Adj. R-Squared: 0.15022
## F-statistic: 43095.7 on 2 and 487579 DF, p-value: < 2.22e-16
```

In the pooled regression, all coefficients and the intercept are statistically significant at the 1% significance level. According to this model, firm returns increase with city and industry returns, respectively, on average, ceteris paribus. There also seems to be a negative default return of -0.0015 that occurs, on average, for city and industry returns of 0.

(e)

```
bptest(pooledLM, studentize = FALSE)
```

```
##
##  Breusch-Pagan test
##
## data:  pooledLM
## BP = 85052, df = 2, p-value < 2.2e-16
```

Here again, we apply the Breusch-Pagan test. The null hypothesis of homoskedasticity can be rejected at the 1% significance level. The coefficients are jointly statistically significant and hence there still seems to be a relationship between the squared residuals and at least one variable. We thus found evidence for heteroskedasticity.

(f)

```
pooledLM_Date1 = felm(ret ~ city_returns + indret | date | 0 | 0, data)
summary(pooledLM_Date1)
```

```
##
## Call:
##    felm(formula = ret ~ city_returns + indret | date | 0 | 0, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2073 -0.0739 -0.0068  0.0573 15.6614
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## city_returns 0.170679   0.010145   16.82   <2e-16 ***
## indret       0.874796   0.006046  144.69   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1855 on 487448 degrees of freedom
## Multiple R-squared(full model): 0.1521   Adjusted R-squared: 0.1519
## Multiple R-squared(proj model): 0.04126   Adjusted R-squared: 0.041
## F-statistic(full model):657.6 on 133 and 487448 DF, p-value: < 2.2e-16
## F-statistic(proj model): 1.049e+04 on 2 and 487448 DF, p-value: < 2.2e-16
```

Using pooled OLS with time fixed effects, the coefficients of city and industry returns stay statistically significant at the 1% significance level. The coefficient of city_returns decreases slight from 0.2122 to 0.1707 and the coefficient of indret increases slightly from 0.8558 to 0.8748. The standard errors of both coefficients increase slightly.

(g)

```
pooledLM_Date_Firm = felm(ret ~ city_returns + indret | date + permno | 0 | 0, data)
summary(pooledLM_Date_Firm)
```

```
##
## Call:
##    felm(formula = ret ~ city_returns + indret | date + permno |      0 | 0, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3172 -0.0742 -0.0073  0.0562 15.0179
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## city_returns 0.172773   0.010199   16.94   <2e-16 ***
## indret       0.870429   0.006082  143.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1851 on 480666 degrees of freedom
## Multiple R-squared(full model): 0.1677   Adjusted R-squared: 0.1557
## Multiple R-squared(proj model): 0.04095   Adjusted R-squared: 0.02715
## F-statistic(full model):14.01 on 6915 and 480666 DF, p-value: < 2.2e-16
## F-statistic(proj model): 1.026e+04 on 2 and 480666 DF, p-value: < 2.2e-16
```

Compared to (f), the model changes barely. The coefficient on city_returns increases slightly and the coefficient on indret decreases slightly. The standard errors also increase slightly.

(h)

```
pooledLM_Date_Firm_ad = felm(ret ~ city_returns + indret | date + permno | 0 |
  date + permno, data)
summary(pooledLM_Date_Firm_ad)
```

the 0 could be replaced by an IV

```
##
## Call:
##    felm(formula = ret ~ city_returns + indret | date + permno |      0 | date + permno, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3172 -0.0742 -0.0073  0.0562 15.0179
##
## Coefficients:
##              Estimate Cluster s.e. t value Pr(>|t|)
## city_returns  0.17277      0.03089   5.592 1.25e-07 ***
## indret        0.87043      0.03285  26.493  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1851 on 480666 degrees of freedom
## Multiple R-squared(full model): 0.1677   Adjusted R-squared: 0.1557
## Multiple R-squared(proj model): 0.04095   Adjusted R-squared: 0.02715
## F-statistic(full model, *iid*):14.01 on 6915 and 480666 DF, p-value: < 2.2e-16
## F-statistic(proj model): 351.2 on 2 and 131 DF, p-value: < 2.2e-16
```

Again, the coefficients barely change compared to (f) and (g). The standard errors, as expected, increase; Clustered standard errors are more conservative. The coefficients are still statistically significant at the 1% significance level. Among these models, the adjusted R-squared barely changes, as well.

(i)

```
feLM_Date_Firm = plm(ret ~ city_returns + indret, model = "within",
  index = c("permno", "date"), data)
summary(feLM_Date_Firm)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = ret ~ city_returns + indret, data = data, model = "within",
##     index = c("permno", "date"))
##
## Unbalanced Panel: n = 6783, T = 1-132, N = 487582
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -1.3028196 -0.0743791 -0.0076208  0.0565786 14.9961571
##
## Coefficients:
##               Estimate Std. Error t-value  Pr(>|t|)
## city_returns 0.2111358  0.0051836  40.732 < 2.2e-16 ***
## indret       0.8540338  0.0050137 170.341 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    19431
```

```
## Residual Sum of Squares: 16496
## R-Squared:      0.15108
## Adj. R-Squared: 0.1391
## F-statistic: 42782.4 on 2 and 480797 DF, p-value: < 2.22e-16
```

(j) & (k)

```
fdLM_Date_Firm = plm(ret ~ city_returns + indret, model = "fd",
  index = c("permno", "date"), data)
summary(fdLM_Date_Firm)
```

```
## Oneway (individual) effect First-Difference Model
##
## Call:
## plm(formula = ret ~ city_returns + indret, data = data, model = "fd",
##     index = c("permno", "date"))
##
## Unbalanced Panel: n = 6783, T = 1-132, N = 487582
## Observations used in estimation: 480799
##
## Residuals:
##        Min.     1st Qu.      Median     3rd Qu.        Max.
## -1.3635e+01 -9.6285e-02 -2.5202e-04  9.4355e-02  1.6300e+01
##
## Coefficients:
##                 Estimate  Std. Error t-value Pr(>|t|)
## (Intercept)  -0.00024950  0.00038326  -0.651    0.515
## city_returns  0.21734628  0.00556280  39.071   <2e-16 ***
## indret        0.84736463  0.00526530 160.934   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    38744
## Residual Sum of Squares: 33955
## R-Squared:      0.1236
## Adj. R-Squared: 0.12359
## F-statistic: 33903.2 on 2 and 480796 DF, p-value: < 2.22e-16
```

In terms of adjusted R-squared the pooled regression model in (h) has the highest value with 0.1557. The model in (i) has an adjusted R-squared of 0.1391 and the model in (j) 0.12359. In terms of coefficients, the models in (i) and (j) are similar. The coefficients for city_returns are slightly higher and for indret slightly lower than in (h). In all 3 models, both coefficients are statistically significant at the 1% significance level. Of these 3 models, the first difference model in (j) is the only one with an estimate for the intercept, -0.0002. This value is not statistically significant at the 10% significance level, though.

(l)

```
adf.test(data$ret, k = 2)
```

```
## Warning in adf.test(data$ret, k = 2): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  data$ret
## Dickey-Fuller = -414.47, Lag order = 2, p-value = 0.01
## alternative hypothesis: stationary
```

```
dataRE = data %>% mutate(dummySF = ifelse(city == "10", 1, 0))
newLM = plm(ret ~ city_returns + indret + dummySF:city_returns, model = "within",
  index = c("permno", "date"), dataRE)
summary(newLM)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = ret ~ city_returns + indret + dummySF:city_returns,
##     data = dataRE, model = "within", index = c("permno", "date"))
##
## Unbalanced Panel: n = 6783, T = 1-132, N = 487582
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -1.3030519 -0.0743536 -0.0076351  0.0565589 14.9963717
##
## Coefficients:
##                        Estimate Std. Error  t-value Pr(>|t|)
## city_returns          0.2091046  0.0052961  39.4825  < 2e-16 ***
## indret                0.8540236  0.0050136 170.3398  < 2e-16 ***
## city_returns:dummySF 0.0239767  0.0128200   1.8703  0.06145 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    19431
## Residual Sum of Squares: 16495
## R-Squared:      0.15108
## Adj. R-Squared: 0.1391
## F-statistic: 28522.9 on 3 and 480796 DF, p-value: < 2.22e-16
```

Running the augmented Dickey-Fuller test on the outcome variable returns a p-value of less than 1%. This means that with reasonable confidence, no unit root is present and a fixed effect estimator is probably better here. This is also true considering that the panel is unbalanced. According to the fixed effect model, the coefficient gamma is statistically significant at the 10% significance level, but not at the 5% significance level. The model can be interpreted as follows. For all cities but San Francisco, the firm returns increase by 0.2091 for an increase of 1 (100%) in the variable city_return and by 0.8540 for an increase of 1 in the variable indret, on average, ceteris paribus, respectively. We can see that both types of returns have a positive effect on the firm return, but the industry effect is stronger than the city effect (for all cities but San Francisco). For San Francisco, the same holds for the indret coefficient, but for city_returns, the effect is slightly stronger at 0.2331. An increase of 1 in the city_returns variable leads to a 0.2331 increase in the firm return, on average, ceteris paribus. This could imply that there are also city fixed effects that need to be accounted for.