Homework2

Linfeng Zhou

September 23, 2015

Notes to students:

1. Going forward, all homework assignments will be produced as notebooks independent of those presented in class. You may feel free to use the notebooks presented in class to assist you with the homework.

2. If you have already run a "CAPM" model for some other stock than AAPL and submitted it with Homework 1, please copy and paste your code and results for this homework assignment. If you did not submit a CAPM for Homework 1, you will not be penalized. Your "CAPM" model will be grade for this homework assignment. If you are struggling because, for example, the New York Stock Exchange has a potentially discontinuous price series, please try using Yahoo, which also trades on the NASDAQ. The goal is not to turn you into finance professionals, but to quickly get you engaged in statistical learning (using a well-understood model) and hypothesis testing.

3. The question has been raised as to whether students of my class and Dr. Sobolevsky's class may work together on their Foundations Project. Subject to the other constraints in the syllabus, we approve this request.

Please answer the questions below, disregarding any homework assignments in existing notebooks.

4. This is a very challenging set of questions, but they address several key topics in data analytics. You may work with other students on a solution with the recognition that you may not complete this set of questions. I have frequently used the phrase "data generating process" (or "DGP") to describe the hypothetical process by which observations of data arise in the real world. We discussed at some length the bivariate linear regression model,

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

   . In this problem, we will work with a specific DGP and evaluate features of $\widehat{\beta}_1$, the least squares estimate of β_1. Suppose your DGP is

$$y_i = 1 + 2x_i + \epsilon_i$$

   , where $x \sim N(0,1)$ and $\epsilon \sim N(0,1)$. Using R or Python, write code to generate 1,000 draws for x and ϵ. Use these draws to generate y in accordance with the given DGP. Using R or Python, write code to estimate the bivariate model,

$$y_i = \beta_0 + \beta_1 x_i$$

   and to summarize the findings.

5. Repeat 4 above five different times for a new set of random draws for each replication. (This effort is called Monte Carlo simulation. Each time you generate a new set of data and estimate a model, you have a replication. For example, here you have five replications.)

6. Write code to automatically repeat 5 above 1,000 times (or 1,000 replications), each time automatically recording the estimated value of β_1. Generate a histogram of these 1,000 replications of your estimates of β_1. What does the dispersion of these replications measure?

7. Suppose that you were not interested in the estimate of β_1, but instead in some functional transformation, such as the estimate of $\exp(\beta_1)$. What might you do with your 1,000 replications from 6 above to inform you about the distribution of this transformation of β_1?

Submit code and results.

Solution

Question 1

1. We discussed at some length the bivariate linear regression model,

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

   Using R or Python, generate two series of 1,000 random integers with values between 0 and 9. Call one series y and the other x. Using R or Python, fit the bivariate linear regression model. Examine the t-statistic on the coefficient that captures the relationship between y and x to evaluate whether it is greater than two in absolute value. Would you reject or fail to reject that there is any relationship between these two series?

```r
## generate samples
x <- sample(c(0:9), 1000,replace = TRUE)
y <- sample(c(0:9), 1000,replace = TRUE)

## formula a linear model
fit1 <- lm(y ~ x)
summary(fit1)
```

Solution

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.711 -2.508 -0.255  2.542  4.745
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.25498    0.16967  25.078   <2e-16 ***
## x            0.05064    0.03201   1.582    0.114
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.895 on 998 degrees of freedom
## Multiple R-squared:  0.002502,   Adjusted R-squared:  0.001503
## F-statistic: 2.503 on 1 and 998 DF,  p-value: 0.1139
```

Comment:

Question 2

2. Download the file train.dta from the course website. These data are formatted as a Stata dataset. Read this dataset into R or Python. (For R, you may find the "foreign" library of use. For Python, we have already installed the Pandas library. The goal here is to get you familiar with reading datasets with

alternative formatting standards.) Generate summary statistics for the two variables in the data: (1) d, which is an indicator for whether a particular email is spam; and, (2) $x1$, which is an attribute of the email. Using least squares, regress d on $x1$. Is the constant statistically significantly different than zero at a 95% level confidence? Is the coefficient associated with $x1$ statistically significantly different that zero at a 95% level confidence? Suppose you determine a threshold as to whether an email is spam if the predicted value of d exceeds 1. In turn, I give you a new email with an attribute value of 0.65 but do not provide a label for the email as to whether it is, in truth, spam. Given the attribute value of 0.65, would you classify the incoming email as spam or not spam? Suppose instead I give you another new email with an attribute value of 1.01, but again do not provide a label for the email. Would you classify it as spam or not spam? #### Solution

```
setwd("~luke/Dropbox/Applied_Data_Science/Assignment 2")
## import data
data <- read.dta("train.dta")

### summary data
summary(data)
```

```
##        d               x1
##  Min.   :0.000   Min.   :0.0002863
##  1st Qu.:0.000   1st Qu.:0.2493821
##  Median :0.000   Median :0.4846455
##  Mean   :0.477   Mean   :0.4873756
##  3rd Qu.:1.000   3rd Qu.:0.7324941
##  Max.   :1.000   Max.   :0.9990059
```

```
### fit model
fit2 <- lm(d ~ x1,data)
summary(fit2)
```

```
##
## Call:
## lm(formula = d ~ x1, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.96928 -0.29151 -0.01333  0.28998  0.95981
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.01795    0.02573  -0.698    0.486
## x1           1.01554    0.04564  22.250   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4088 on 998 degrees of freedom
## Multiple R-squared:  0.3316, Adjusted R-squared:  0.3309
## F-statistic:   495 on 1 and 998 DF,  p-value: < 2.2e-16
```

```
### predict
data[1001:1002,2] <- c(0.61,1.01)
pred1 <- predict(fit2,data)
pred1[1001:1002] > 1
```

```
##  1001  1002
## FALSE  TRUE
```

Comment

Qustion 3

3. Using an API such as Quandl, download a daily price series for a particular publicly-traded stock of your choice for a five-year time period (don't use Apple), as well as the daily price series on the exchange on which it trades. Using R or Python, calculate the log returns of each series as the natural log of the ratio of (price today/price yesterday). Use adjusted closing prices as they reflect so-called stock splits. Using R or Python, generate a histogram of log returns of the stock of your choice. Using R or Python, generate a scatterplot that relates the log returns of your stock of choice to the log returns of the exchange on which it is traded. Finally, using R or Python, fit a linear model to obtain estimates of what finance folks call the "alpha" and the "beta". Is "alpha" significantly different than zero at a 95% level of confidence? Does a 95% confidence level for "beta" include one? (Note that your results will depend on the stock price you use.) ####Solution

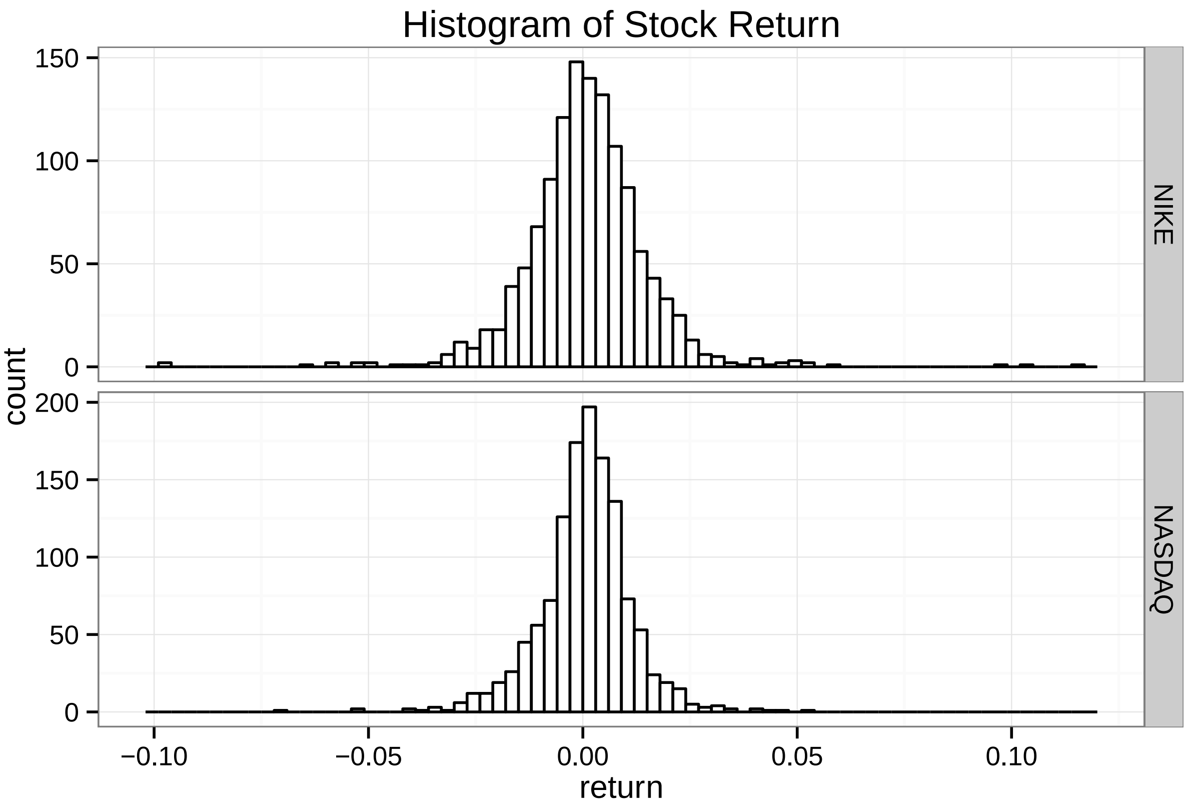
```r
NIKE_INDEX<-getSymbols("NKE",from="2010-09-20", to='2015-09-20',src="yahoo")
```

```
##     As of 0.4-0, 'getSymbols' uses env=parent.frame() and
##  auto.assign=TRUE by default.
##
##  This  behavior  will be  phased out in 0.5-0  when the call  will
##  default to use auto.assign=FALSE. getOption("getSymbols.env") and
##  getOptions("getSymbols.auto.assign") are now checked for alternate defaults
##
##  This message is shown once per session and may be disabled by setting
##  options("getSymbols.warning4.0"=FALSE). See ?getSymbols for more details.
```

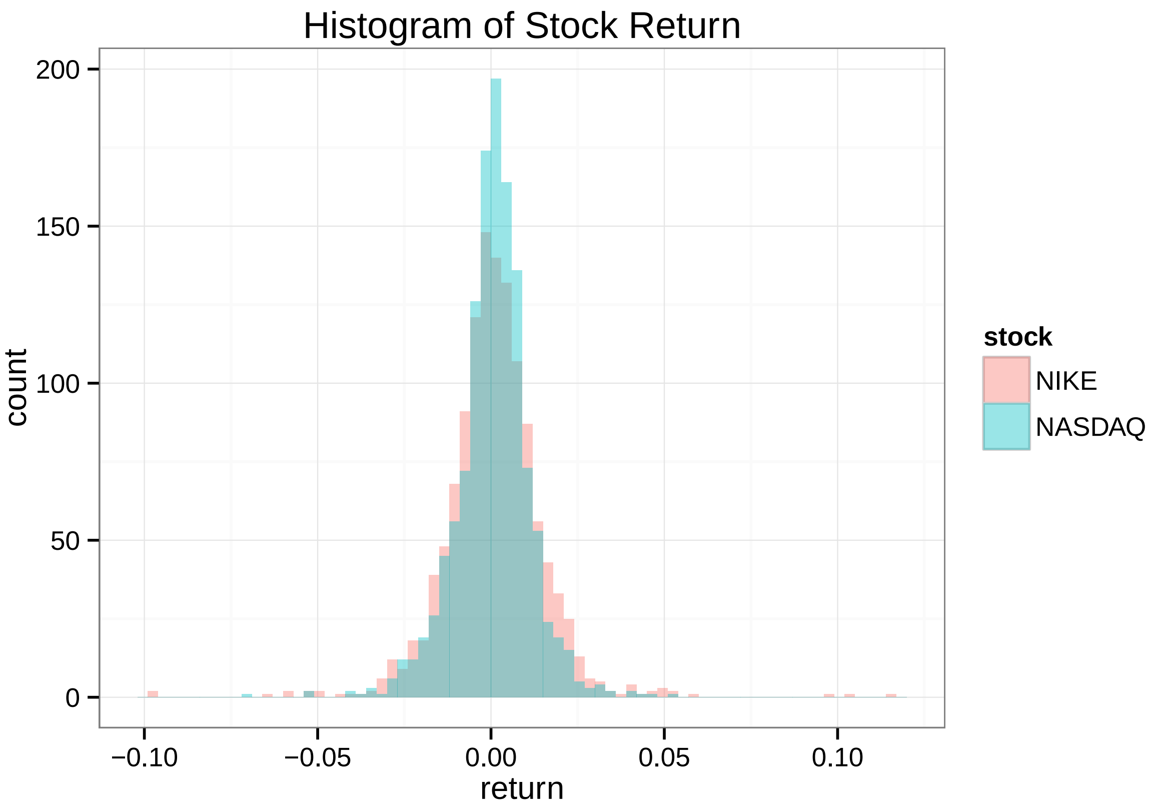
```r
NASQ_INDEX<-getSymbols("^IXIC",from="2010-09-20", to='2015-09-20',src="yahoo")
NIKE_return=log(NKE$NKE.Adjusted/lag(NKE$NKE.Adjusted))
NASQ_return=log(IXIC$IXIC.Adjusted/lag(IXIC$IXIC.Adjusted))

alldata <- cbind(NIKE_return[,1], NASQ_return[,1])

### eliminate format
write.csv(alldata,"allclean.csv",row.names=FALSE)
d<-read.csv("allclean.csv",header=TRUE)

### plot histogram
NIKE_f = cbind(d$NKE.Adjusted[-1],0)
NASQ_f = cbind(d$IXIC.Adjusted[-1],1)
alldata <- rbind(NIKE_f,NASQ_f)
alldata <- data.frame(alldata)
colnames(alldata) <- c("return","stock")
alldata$stock<-as.factor(alldata$stock)
levels(alldata$stock)
```

```
## [1] "0" "1"
```

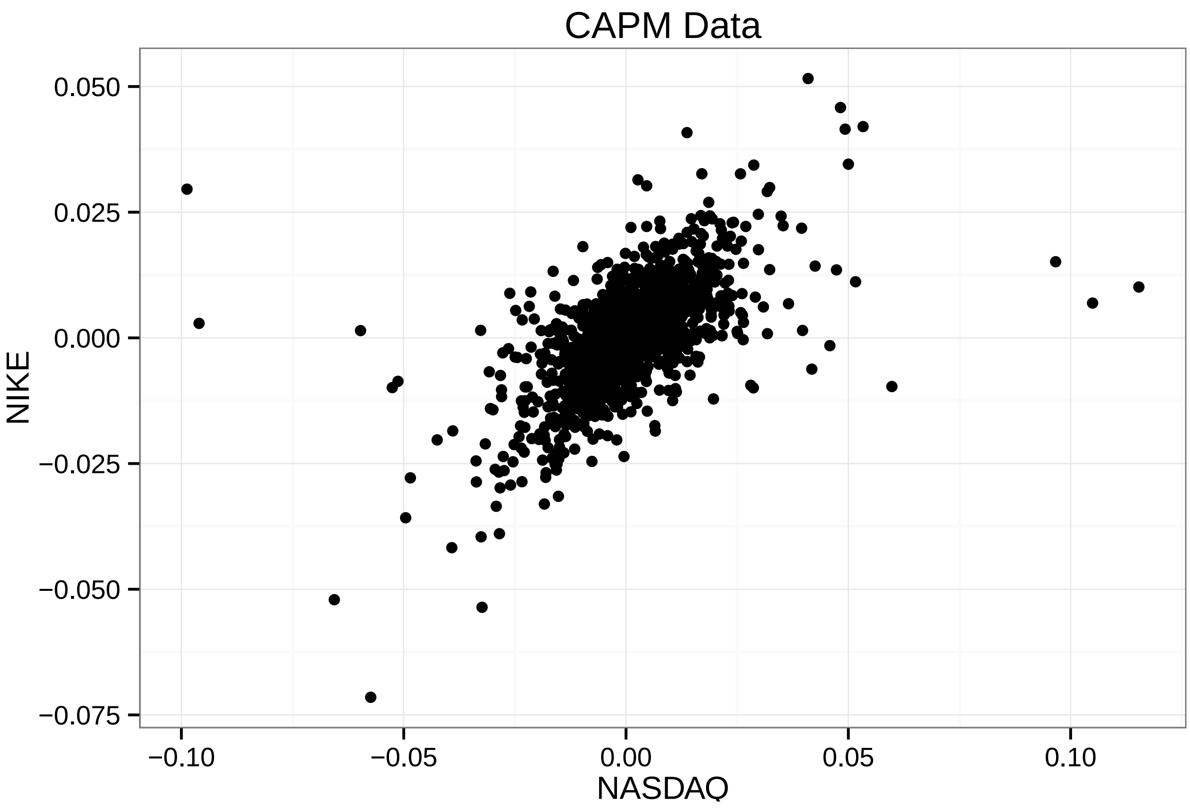
```
alldata$stock<-revalue(alldata$stock,c("0"="NIKE","1"="NASDAQ"))
ggplot(alldata,aes(x=return)) +
  geom_histogram(binwidth=0.003,fill="white",colour="black")+
  facet_grid(stock~., scales="free")+
  ggtitle("Histogram of Stock Return")+
  theme_bw()
```



```
ggplot(alldata,aes(x=return,fill=stock)) +
  geom_histogram(binwidth=0.003,position='identity', alpha=0.4)+
  ggtitle("Histogram of Stock Return")+
  theme_bw()
```



```
### plot scatter plot
alldata1<-cbind(NASQ_f[,1],NIKE_f[,1])
alldata1<-data.frame(alldata1)
colnames(alldata1)<-c("NIKE","NASDAQ")
ggplot(alldata1,aes(x=NASDAQ,y=NIKE)) + geom_point()+
  theme_bw()+ggtitle("CAPM Data")
```



```
### fit linear model
fit3 <- lm(NIKE~NASDAQ,alldata1)
summary(fit3)
```

```
##
## Call:
## lm(formula = NIKE ~ NASDAQ, data = alldata1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.046460 -0.004647  0.000190  0.004767  0.072751
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0001712  0.0002451   0.699    0.485
## NASDAQ      0.4388958  0.0167309  26.233   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.008675 on 1256 degrees of freedom
## Multiple R-squared:  0.354,  Adjusted R-squared:  0.3534
## F-statistic: 688.2 on 1 and 1256 DF,  p-value: < 2.2e-16
```

```
### alpha beta f test
residuals1 <- fit3$residuals
residuals2 <- NASQ_f[,1] - NIKE_f[,1]
var.test(residuals1,residuals2)
```

```
##
##  F test to compare two variances
##
## data:  residuals1 and residuals2
## F = 0.52757, num df = 1257, denom df = 1257, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.4723235 0.5892737
## sample estimates:
## ratio of variances
##          0.5275678
```

```
##
```

Comment