Project 2

Matthew Stacy

1/26/2020

## Set Up

install.packages(“rmarkdown”) #If you do not have R Markdown, you may install it using this command.  
pacman::p\_load(pacman, rmarkdown) #To load R Markdown, we can use pacman to conviently load the necessary packages.

### Question 1

Read the dataset in SP500.xlsx into R. Call the loaded data SP500. Make sure that you have the directory set to the correct location for the data.

library(readxl)  
SP500 <- read\_xlsx("C:/Users/stacy/OneDrive/School/Maryville/2020 - Spring/SDCI 502 - Introduction to R/Module 2/SP500.xlsx",  
 sheet = "Data")

The library() function allows us to read excel files. Then, we supply the name of the new object and we use the read\_xlsx() function to import the data and assign it to the object named at the beginning of the command line. In the path to the excel file, we must specify the worksheet we want to work with as there are more than one worksheet in the excel file.

### Question 2

How many rows are in the data set? How many columns are in the data set?

dim(SP500)

## [1] 1773 6

The dim() function displays the number of rows and columns found within the data set. It will display the numbers of rows first, followed by the number of columns. There are 1773 rows and 6 columns. Each row represents values associated with a month and year starting at January 1871 and ending at September 2018. Each column represents a variable.

### Question 3

Select the following three columns: SP500, CPI, and Rate.

SP500[ , c("SP500", "CPI", "Rate")]

## # A tibble: 1,773 x 3  
## SP500 CPI Rate  
## <dbl> <dbl> <dbl>  
## 1 4.44 12.5 5.32  
## 2 4.5 12.8 5.32  
## 3 4.61 13.0 5.33  
## 4 4.74 12.6 5.33  
## 5 4.86 12.3 5.33  
## 6 4.82 12.1 5.34  
## 7 4.73 12.1 5.34  
## 8 4.79 11.9 5.34  
## 9 4.84 12.2 5.35  
## 10 4.59 12.4 5.35  
## # ... with 1,763 more rows

By writing the data frame’s name with a bracket, it selects that data frame. Then, it follows an x, y pattern. The x is the row selection; by leaving the x selection blank, we are selecting all of the rows within the data frame. Following is the y selection; this is where we specify the three columns within the question (SP500, CPI, and Rate). We use the concatenante function, which combines the desired selections into one command. Since we will not call on this selection later, we will not assign it to an object. All rows and the three selected variables will be printed into the R Markdown output file.

### Question 4

Select the 10th, 100th, 500th, and 1500th rows.

SP500[c(10, 100, 500, 1500), ]

## # A tibble: 4 x 6  
## Date SP500 Dividend Earnings CPI Rate  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1871. 4.59 0.26 0.4 12.4 5.35  
## 2 1879. 3.77 0.187 0.333 8.18 4.17  
## 3 1912. 9.81 0.477 0.663 9.70 4.27  
## 4 1995. 615. 13.8 34.0 154. 5.71

By writing the data frame’s name with a bracket, it selects that data frame. Then, it follows an x, y pattern. The x is the row selection; this is where we specify the four rows within the question (10, 100, 500, and 1500). We use the concatenate function, which combines the desired selections into one command. Following is the y selection; by leaving the y selection blank, we are selecting all of the columns within the data frame. Since we will not call on this selection later, we will not assign it to an object. The four rows and all of the variables will be printed into the R Markdown output file.

### Question 5

Select all the observations such that SP500 is greather than 2000 or CPI is less than 100.

subset(SP500, (SP500 > 2000) | (CPI < 100))

## # A tibble: 1,395 x 6  
## Date SP500 Dividend Earnings CPI Rate  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1871. 4.44 0.26 0.4 12.5 5.32  
## 2 1871. 4.5 0.26 0.4 12.8 5.32  
## 3 1871. 4.61 0.26 0.4 13.0 5.33  
## 4 1871. 4.74 0.26 0.4 12.6 5.33  
## 5 1871. 4.86 0.26 0.4 12.3 5.33  
## 6 1871. 4.82 0.26 0.4 12.1 5.34  
## 7 1871. 4.73 0.26 0.4 12.1 5.34  
## 8 1871. 4.79 0.26 0.4 11.9 5.34  
## 9 1871. 4.84 0.26 0.4 12.2 5.35  
## 10 1871. 4.59 0.26 0.4 12.4 5.35  
## # ... with 1,385 more rows

The subset() function allows the command to be cleaner by following the (x, …) system. The desired data frame is specified as the x selection, then the arguments follow. When using the subset() function, R uses the selected data frame as the parent object and does not require the arguments to be dataframe$variable. In this question, the arguments are to select any row with a value of SP500 larger than 2000 OR any row with a CPI value smaller than 100. Since we will not call on this selection later, we will not assign it to an object. The result will populate 1395 rows and 6 variables into the R Markdown output file.

### Question 6

Select the data such that Earnings greater than 50 and Rate less than 3 with columns SP500 and Dividend only.

subset(SP500, (Earnings > 50) & (Rate < 3), select = c(SP500, Dividend))

## # A tibble: 89 x 2  
## SP500 Dividend  
## <dbl> <dbl>  
## 1 1087. 22.2  
## 2 1122. 22.4  
## 3 1172. 22.5  
## 4 1199. 22.6  
## 5 1185. 24.9  
## 6 1174. 25.2  
## 7 1207. 25.6  
## 8 1226. 26.0  
## 9 1243. 26.4  
## 10 1301. 26.7  
## # ... with 79 more rows

The subset() functions allows the command to be cleaner by following the (x, …) system. The desired data frame is specified as the x selection, then the arguments follow. When using the subset() function, R uses the selected data frame as the parent object and does not require the arguments to be dataframe$variable. In this question, the arguments are to select any row with a value of Earnings larger than 50 AND a Rate value smaller than 3. Then, it will only select to show the SP500 and the Dividend variables in the result. Since we will not call on this selection later, we will not assign it to an object. The result will populate 89 rows and 2 variabes into the R Markdown output file.

### Question 7

Remove the entire column of Rate.

SP500$Rate <- NULL  
colnames(SP500)

## [1] "Date" "SP500" "Dividend" "Earnings" "CPI"

By writing the data frame name followed by a $ symbol, we are specifying a column name within that desired data frame. In this quesiton, the Rate column is being selected and the logical statement NULL is being applied to make the chosen column undefined. Next, the colnames() function returns the names of all of the column headers. This is performed to ensure the Rate column is no longer present within the data frame.The result will populate the column names Date, SP500, Dividend, Earnings, and CPI into the R Markdown output file.

### Question 8

Real price is the inflation adjusted price, which is given by the following formula:  
Real Price at time t = (SP500 price at time t)\*CPI(t)/CPI(2018.09), where CPI(2018.09) is the latest consumer price index in the data set. Based on the formula above, you need to add one more column, RealPrice.

SP500$RealPrice <- (SP500$SP500)\*(SP500$CPI)/(SP500$CPI[which.max(SP500$Date)])  
print(SP500)

## # A tibble: 1,773 x 6  
## Date SP500 Dividend Earnings CPI RealPrice  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1871. 4.44 0.26 0.4 12.5 0.219  
## 2 1871. 4.5 0.26 0.4 12.8 0.229  
## 3 1871. 4.61 0.26 0.4 13.0 0.238  
## 4 1871. 4.74 0.26 0.4 12.6 0.236  
## 5 1871. 4.86 0.26 0.4 12.3 0.236  
## 6 1871. 4.82 0.26 0.4 12.1 0.231  
## 7 1871. 4.73 0.26 0.4 12.1 0.226  
## 8 1871. 4.79 0.26 0.4 11.9 0.226  
## 9 1871. 4.84 0.26 0.4 12.2 0.234  
## 10 1871. 4.59 0.26 0.4 12.4 0.225  
## # ... with 1,763 more rows

By writing the data frame name followed by a $ symbol, we are specifying a column name within that desired data frame. In this question, the specified column name does not exist within the data frame. The command is assinging a calculated value to represent the new column. The calculated value is taken by multiplying the SP500 value at any given date by the CPI value at the same date and dividing the result by the latest CPI value. The latest CPI value is derived by the which.max() function, which returns the maximum value within a specified object. In this data frame, that is 252.439 at September 2018. The resulting value will populate the rows within the newly assigned column RealPrice. The result of this command will be displayed using the print() function within the R Markdown output file.

When comparing the real price of the SP500 of any date previous to the baseline date of September 2018, it makes sense that it would be less than that of today’s SP500 price when taking into account inflation. The price of the SP500 will gradually increase so long as the CPI increases. Once the date has reached past the baseline date, if the CPI increases, it should be expected that the real price also increases. Also, if the CPI decreases, it should be expected that the real price also decreases.

### Question 9

Real earnings are the inflation adjusted earnings, which is given by the following formula:  
Real earnings at time t = (Earnings at time t)\*CPI(t)CPI(2018.09), where CPI(2018.09) is the latest consumer price index in the data set. Based on the formula above, you need to add one more column, RealEarnings.

SP500$RealEarnings <- (SP500$Earnings)\*(SP500$CPI)/(SP500$CPI[which.max(SP500$Date)])  
print(SP500)

## # A tibble: 1,773 x 7  
## Date SP500 Dividend Earnings CPI RealPrice RealEarnings  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1871. 4.44 0.26 0.4 12.5 0.219 0.0197  
## 2 1871. 4.5 0.26 0.4 12.8 0.229 0.0204  
## 3 1871. 4.61 0.26 0.4 13.0 0.238 0.0207  
## 4 1871. 4.74 0.26 0.4 12.6 0.236 0.0199  
## 5 1871. 4.86 0.26 0.4 12.3 0.236 0.0194  
## 6 1871. 4.82 0.26 0.4 12.1 0.231 0.0191  
## 7 1871. 4.73 0.26 0.4 12.1 0.226 0.0191  
## 8 1871. 4.79 0.26 0.4 11.9 0.226 0.0188  
## 9 1871. 4.84 0.26 0.4 12.2 0.234 0.0193  
## 10 1871. 4.59 0.26 0.4 12.4 0.225 0.0196  
## # ... with 1,763 more rows

By writing the data frame name followed by a $ symbol, we are specifying a column name within that desired data frame. In this question, the specified column name does not exist within the data frame. The command is assigning a calculated value to represent the new column. The calculated value is taken by multiplying the Earnings value at any given date by the CPI value at the same date and dividing the result by the latest CPI value. The latest CPI value is derived by the which.max() function, which returns the maximum value within a specified object. In this data frame, that is 252.439 at September 2018. The resulting value will populate the rows within the newly assigned column RealEarnings. The result of this command will be displayed using the print() function within the R Markdown output file.

When comparing the real earnings of the SP500 of any date previous to the baseline date of September 2018, it makes sense that it would be less than that of today’s SP500 earnings when taking into account inflation. The earnings of the SP500 will gradually increase so long as the CPI increases. Once the date has reached past the baseline date, if the CPI increases, it should be expected that the real earnings also increases. Also, if the CPI decreases, it should be expected that the real earnings also decreases.

### Question 10

Price to earnings ratio is given by the following formula:  
P/E Ratio = RealPrice/RealEarnings  
Based on the formula above, please add one more column, PERatio.

SP500$PERatio <- (SP500$RealPrice/SP500$RealEarnings)  
print(SP500)

## # A tibble: 1,773 x 8  
## Date SP500 Dividend Earnings CPI RealPrice RealEarnings PERatio  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1871. 4.44 0.26 0.4 12.5 0.219 0.0197 11.1  
## 2 1871. 4.5 0.26 0.4 12.8 0.229 0.0204 11.2  
## 3 1871. 4.61 0.26 0.4 13.0 0.238 0.0207 11.5  
## 4 1871. 4.74 0.26 0.4 12.6 0.236 0.0199 11.9  
## 5 1871. 4.86 0.26 0.4 12.3 0.236 0.0194 12.1  
## 6 1871. 4.82 0.26 0.4 12.1 0.231 0.0191 12.0  
## 7 1871. 4.73 0.26 0.4 12.1 0.226 0.0191 11.8  
## 8 1871. 4.79 0.26 0.4 11.9 0.226 0.0188 12.0  
## 9 1871. 4.84 0.26 0.4 12.2 0.234 0.0193 12.1  
## 10 1871. 4.59 0.26 0.4 12.4 0.225 0.0196 11.5  
## # ... with 1,763 more rows

By writing the data frame name followed by a $ symbol, we are specifying a column name within that desired data frame. In this question, the specified column name does not exist within the data frame. The command is assigning a calculated value to represent the new column. The calculated value is taken by dividing the RealPrice value at any given date by the RealEarnings value at the same date. The resulting value will populate the rows within the newly assigned column PERatio. The result of this command will be displayed using the print() function within the R Markdown output file.

As time progresses forward, it appears that the SP500 is earning less as the price increases. The larger the ratio, the less is earned. It appears that the Price to Earnings ratio has doubled from 11.10000 to 22.25247. This indicates that each SP500 stock costs twice as much but is earning the stockholder less money. Typically, when utilizing baseline statistics involving dates, it is standard to set the baseline as a static value. In this case, we would dictate the baseline to be September 2018. As more data was captured and was added to the data source, the same value from Semptermber 2018 for CPI would stay static at 252.439 to indicate a proper comparison from the future’s data to the past’s data. It is clear that there is an upward trend in all of the values indicating inflation at work.

## Clean Up

Once completed, you may clear your R Studio with the following:

Clear Environments  
rm(list = ls())

Clear Console  
cat(“\014”) #Or ctrl/cmd + L