HW2 Part 2

July 31, 2021

1 MGT6203 HW2 Part 2

1.1 Instructions:

For Homework 2 Part 2, please use this R notebook in Vocareum to submit your solutions. Vocareum is an educational cloud platform for programming in several languages; it is based on the Jupyter notebook environment. This platform allows us to move homework assignments to the cloud. The advantages are that all of you will be working in the same coding environment AND peer reviewers will be able to run your R code easily. This way we eliminate some issues we might encounter when working on an individual/local basis, such as library installations and Rstudio OS requirements; R notebooks work on mobile platforms and tablets.

With R notebooks, you will be learning a new way of presenting data analysis reports, that is neat and flexible, where formatted (English) text and (R) code can easily coexist on the same page. Notebooks can be also collaborative when needed. For now, we are asking each of you to do your own work for homework. Think of R notebooks as interactive program-based Google docs or MS-Office 360 docs; these are gradually replacing local files on our computers.

Many of you are new to the R notebooks and Vocareum platforms. We will provide TA help in Piazza with specific code if you have questions. Here we list some important things to get you started. Please read through them carefully.

Even though we are moving from your local envrionment to the cloud, **our expectations from your homework will remain the same**. Same goes for the rubrics.

Vocareum has its own cloud based file system, the data files you will be using for the assignments will be stored in the cloud with path **"../resource/asnlib/publicdata/FILENAME.csv"**. You will be able to import them with the same method as you do in RStudio, simply substitute the path name to the one specified in the instructions. You won't be able to modify these data files.

You will be able to find the data files on Canvas/EdX if you would like to explore them offline.

For coding questions, you will be graded on the R code as well as the output in your submission.

For interpretations or short response questions, please type the answers in the notebook's markdown cells. To change a code cell to a markdown cell, click on the cell, and in the dropdown menu above, switch the type of the cell block from "code" to "markdown". **Adding print statements to code cells for short response/interpretation questions is also fine, as long as we can clearly see the output of your response**.

You don't need to, but if you would like to learn more about how to format your markdown cells, visit the following site: https://www.earthdatascience.org/courses/intro-to-earth-data-science/file-formats/use-text-files/format-text-with-markdown-jupyter-notebook/. Jupyter notebook also support LaTeX.

Feel free to delete or add as many additional cells as you need. But please try to keep your notebook clean and keep your solution to a question directly under that question to avoid confusions.

You may delete the #SOLUTION BEGINS/ENDS HERE comments from the cell blocks, they are just pointers that indicates where to put you solutions.

When you have finished the assignment, remember to rerun your notebook to check if it runs correctly. You can do so by going to **Kernel-> Restart & Run All**. You may lose points if your solutions does not run successfully.

Click the "Submit" button on the top right corner to turn in your assignment. Your assignment will enter the next phase for peer review.

**You are allowed a total of 2 submissions for this assignment. So make sure that you submit your responses carefully. You will be able to come back and resubmit your assignment as long as it is before the start of the peer review period.

Please remember to finish the peer reviews after you have submitted your assignment. You are responsible for grading the work of three of your peers thoroughly, and in addherence to the rubrics. And you will be held accountable for peer grading. **There will be a 30% penalty to your grade if you fail to complete one or more peer reviews in proper fashion.**

Good Luck!

2 About Package Installation:

Most of the packages (if not all) that you will need to complete this assignment are already installed in this environment. An easy way to check this is to run the command: library(PackageName). If this command runs successfully then the package was already installed and has been successfully attached to the code. If the command gave an error saying the Package was not found then follow the steps below to successfully install the package and attach it to the code:

Use *installed.packages()* command to return a table of the packages that are preinstalled in the environment.

To attach a preinstalled library in Vocareum, simply use *library(PackageName)*

To install a package that does not come with the provided environment, please use the following syntax:

install.packages("PackageName", lib="../work/")

To attach a library you just installed, use the following syntax:

library(PackageName, lib.loc="../work/")

Make sure the file location is the same as the above code snippets *("../work/")*

2.1 Q1. (30 Points Total)

2.1.1 For part A and B:

"PlantGrowth" is a dataset in R that contains crop weights of a control group and two treatment groups.

#Code to Get Data
library(datasets)
data(PlantGrowth)

Perform the following operations: Create two separate datasets: dataset1 contains datapoints of treatment 1 group and control group, and dataset2 contains datapoints of treatment 2 group with the control group.

2.1.2 A. Now compute the difference estimator for each dataset that were just created. (treatment 1 vs control group for dataset1, and treatment 2 vs control group for dataset2) (5 Points)

```
In [18]: # SOLUTION BEGINS HERE
         if(!require("datasets")) {install.packages("datasets")
                                   library("datasets")}
         if(!require("dplyr")) {install.packages("dplyr")
                                library("dplyr")}
         data(PlantGrowth)
```

```
Loading required package: dplyr


Attaching package: dplyr


The following objects are masked from package:stats:

    filter, lag


The following objects are masked from package:base:

    intersect, setdiff, setequal, union
```

```
In [26]: data = PlantGrowth
         str(data)
         unique(data$group)
         head(data)
```

```
'data.frame':        30 obs. of  2 variables:
 $ weight: num  4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...
 $ group : Factor w/ 3 levels "ctrl","trt1",..: 1 1 1 1 1 1 1 1 1 1 ...
```

1. ctrl 2. trt1 3. trt2
Levels: 1. 'ctrl' 2. 'trt1' 3. 'trt2'

A data.frame: 6 Œ 2

	weight	group
	<dbl>	<fct>
1	4.17	ctrl
2	5.58	ctrl
3	5.18	ctrl
4	6.11	ctrl
5	4.50	ctrl
6	4.61	ctrl

```
In [27]: data1 = subset(data, group == "trt1" | group == "ctrl")
         str(data1)
         unique(data1$group)
```

```
'data.frame':        20 obs. of  2 variables:
 $ weight: num  4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...
 $ group : Factor w/ 3 levels "ctrl","trt1",..: 1 1 1 1 1 1 1 1 1 1 ...
```

1. ctrl 2. trt1
Levels: 1. 'ctrl' 2. 'trt1' 3. 'trt2'

```
In [31]: data2 = subset(data, group == "trt2" | group == "ctrl")
         str(data2)
         unique(data2$group)
```

```
'data.frame':        20 obs. of  2 variables:
 $ weight: num  4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...
 $ group : Factor w/ 3 levels "ctrl","trt1",..: 1 1 1 1 1 1 1 1 1 1 ...
```

1. ctrl 2. trt2
Levels: 1. 'ctrl' 2. 'trt1' 3. 'trt2'

```
In [33]: model1 = lm(weight~group, data1)
         summary(model1)
```

```
Call:
lm(formula = weight ~ group, data = data1)

Residuals:
    Min      1Q  Median      3Q     Max
-1.0710 -0.4938  0.0685  0.2462  1.3690

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.0320     0.2202  22.850 9.55e-15 ***
grouptrt1    -0.3710     0.3114  -1.191    0.249
---
```

```
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.6964 on 18 degrees of freedom
Multiple R-squared:  0.07308,Adjusted R-squared:  0.02158
F-statistic: 1.419 on 1 and 18 DF,  p-value: 0.249
```

```
In [35]: model2 = lm(weight~group, data2)
         summary(model2)
         # SOLUTION ENDS HERE
```

```
Call:
lm(formula = weight ~ group, data = data2)

Residuals:
   Min     1Q Median     3Q    Max
-0.862 -0.410 -0.006  0.280  1.078

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.0320     0.1637  30.742   <2e-16 ***
grouptrt2     0.4940     0.2315   2.134   0.0469 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.5176 on 18 degrees of freedom
Multiple R-squared:  0.2019,Adjusted R-squared:  0.1576
F-statistic: 4.554 on 1 and 18 DF,  p-value: 0.04685
```

2.1.3 B. From the 'PlantGrowth' dataset, what is the average crop weight of the control group, treatment 1 group, and treatment 2 group? Comment on which group has the highest average. (6 Points)

```
In [48]: # SOLUTION BEGINS HERE
         avg_weight_ctrl = mean(data[which(data$group == "ctrl"),]$weight)
         avg_weight_trt1 = mean(data[which(data$group == "trt1"),]$weight)
         avg_weight_trt2 = mean(data[which(data$group == "trt2"),]$weight)

         print("Average weight for:")
         print(paste0("Control: ", avg_weight_ctrl))
         print(paste0("Treatment Group 1: ", avg_weight_trt1))
         print(paste0("Treatment Group 2: ", avg_weight_trt2))
         # SOLUTION ENDS HERE
```

```
[1] "Average weight for:"
[1] "Control: 5.032"
[1] "Treatment Group 1: 4.661"
[1] "Treatment Group 2: 5.526"
```

Treatment group 2 has the highest average weight as compared to Control and Treatment 1. Difference estimator from the 2 models correspond to differences in average weight between each group.

2.1.4 For Parts C,D, and E:

Use the Dataset "Min_Wage.csv" and the information below to answer the questions.

The Minimum Wage Law protects the right of workers to get a minimum wage. Consider a scenario where the law of minimum wage was changed just in the state of New Jersey (i.e., law has not been changed in other states). We want to use the data from company XYZ to observe the difference in hours worked by full time employees in New Jersey before and after the law was changed.

Note: The variable 'State' indicates the citizenship of the worker, i.e., State = New Jersey ,the worker is from NJ, else worker is not from NJ (is from Philadelphia).

Note: The variable 'fte' contains the number of hours worked by a full time employee.

Note: The variable 'd' indicates whether or not the data was collected before or after the law changed, i.e. d = 1 indicates the data was collected after the law was changed, and d = 0 indicates the data was collected before the law was changed.

Note: The file "Min_Wage.csv" can be accessed at the path: ("../resource/asnlib/publicdata/Min_Wage.csv"). E.g. read.csv("../resource/asnlib/publicdata/Min_Wage.csv")

2.1.5 C. In the above problem, classify the workers into four groups and assign the corresponding group with the group title (A,B,C, and D) where the group titles are as follows: (8 Points)

	Before	After
Control	A	C
Treated	B	D

Note: This is a theoretical question, you just need to identify each group with a short description of what the group is. You don't need to create a column categorizing the datapoints into groups.

```
In [87]: # SOLUTION BEGINS HERE
         data = read.csv("../resource/asnlib/publicdata/Min_Wage.csv")
         str(data)
         head(data)
         unique(data$State)
         unique(data$d)
```

```
'data.frame':        768 obs. of  14 variables:
 $ d       : int  0 0 0 0 0 0 0 0 0 0 ...
 $ d_nj    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ fte     : num  15 15 24 19.2 21.5 ...
 $ bk      : int  1 1 0 0 1 0 0 1 0 0 ...
 $ kfc     : int  0 0 0 0 0 1 0 0 0 0 ...
 $ roys    : int  0 0 1 1 0 0 0 0 0 0 ...
 $ wendys  : int  0 0 0 0 0 0 1 0 1 1 ...
 $ co_owned: int  0 0 0 1 0 0 0 0 0 0 ...
 $ centralj: int  1 1 1 0 0 0 0 0 0 0 ...
 $ southj  : int  0 0 0 0 0 0 0 0 1 1 ...
 $ pa1     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ pa2     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ demp    : num  12 6.5 -1 2.25 13 1 -27.5 17 8 6 ...
 $ State   : chr  "New Jersey" "New Jersey" "New Jersey" "New Jersey" ...
```

A data.frame: 6 Œ 14

	d <int>	d_nj <int>	fte <dbl>	bk <int>	kfc <int>	roys <int>	wendys <int>	co_owned <int>	centralj <int>	s <
1	0	0	15.00	1	0	0	0	0	1	0
2	0	0	15.00	1	0	0	0	0	1	0
3	0	0	24.00	0	0	1	0	0	1	0
4	0	0	19.25	0	0	1	0	1	0	0
5	0	0	21.50	1	0	0	0	0	0	0
6	0	0	9.50	0	1	0	0	0	0	0

1. 'New Jersey' 2. 'Philadelphia'
1. 0 2. 1

```
In [88]: data$State = relevel(as.factor(data$State), ref = "Philadelphia")
         str(data)
```

```
'data.frame':        768 obs. of  14 variables:
 $ d       : int  0 0 0 0 0 0 0 0 0 0 ...
 $ d_nj    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ fte     : num  15 15 24 19.2 21.5 ...
 $ bk      : int  1 1 0 0 1 0 0 1 0 0 ...
 $ kfc     : int  0 0 0 0 0 1 0 0 0 0 ...
 $ roys    : int  0 0 1 1 0 0 0 0 0 0 ...
 $ wendys  : int  0 0 0 0 0 0 1 0 1 1 ...
 $ co_owned: int  0 0 0 1 0 0 0 0 0 0 ...
 $ centralj: int  1 1 1 0 0 0 0 0 0 0 ...
 $ southj  : int  0 0 0 0 0 0 0 0 1 1 ...
 $ pa1     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ pa2     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ demp    : num  12 6.5 -1 2.25 13 1 -27.5 17 8 6 ...
 $ State   : Factor w/ 2 levels "Philadelphia",..: 2 2 2 2 2 2 2 2 2 2 ...
```

If this was a natural experement, changes in New Jersey would be treated as the treatment group, and states without changes would be the control. Group A: State = "Philadelphia", d = 0 Group B: State = "New Jersey", d = 0 Group C: State = "Philadelphia", d = 1 Group D: State = "New Jersey", d = 1

```
In [78]: data = data %>% mutate(group = case_when((State == "Philadelphia" & d == 0) ~ "A",
                                                  (State == "New Jersey" & d == 0) ~ "B",
                                                  (State == "Philadelphia" & d == 1) ~ "C",
                                                  (State == "New Jersey" & d == 1) ~ "D"))
         table(data$group)
         # SOLUTION ENDS HERE
```

```
  A   B   C   D
 75 309  75 309
```

2.1.6 D. To estimate the difference in difference we need four averages for the above categorized groups i.e., control group before change, control group after change, treatment group before change and treatment group after change. Compute the following: (7 Points)

i. Calculate the mean of the 'fte' variable for each of the four groups in R and print them. (4 Points)

```
In [79]: # SOLUTION BEGINS HERE
         mean_A = mean(data[which(data$group == "A"),]$fte)
         mean_B = mean(data[which(data$group == "B"),]$fte)
         mean_C = mean(data[which(data$group == "C"),]$fte)
         mean_D = mean(data[which(data$group == "D"),]$fte)

         print(paste0("Mean fte for Group A: ", mean_A))
         print(paste0("Mean fte for Group B: ", mean_B))
         print(paste0("Mean fte for Group C: ", mean_C))
         print(paste0("Mean fte for Group D: ", mean_D))
         # SOLUTION ENDS HERE
```

```
[1] "Mean fte for Group A: 23.38"
[1] "Mean fte for Group B: 20.4305825242718"
[1] "Mean fte for Group C: 21.0966666666667"
[1] "Mean fte for Group D: 20.8972491909385"
```

ii. Using these averages, estimate the value of the difference in difference. (3 Points)

```
In [80]: # SOLUTION BEGINS HERE
         diff_ctrl = mean_C - mean_A
         diff_trt = mean_D - mean_B
         diff_in_diff = diff_trt - diff_ctrl
         print(paste0("Difference in difference = ", diff_in_diff))
         # SOLUTION ENDS HERE
```

```
[1] "Difference in difference = 2.75"
```

2.1.7 E. Estimate the DID (Difference in Difference) using regression model. (4 Points)

```
In [91]: # SOLUTION BEGINS HERE
         model = lm(fte ~ State + d + State*d, data)
         summary(model)
         # SOLUTION ENDS HERE
```

```
Call:
lm(formula = fte ~ State + d + State * d, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-21.097  -6.472  -0.931   4.603  64.569 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)         23.380      1.098  21.288   <2e-16 ***
StateNew Jersey     -2.949      1.224  -2.409   0.0162 *  
d                   -2.283      1.553  -1.470   0.1419    
StateNew Jersey:d    2.750      1.731   1.588   0.1126    
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 9.511 on 764 degrees of freedom
Multiple R-squared:  0.007587,Adjusted R-squared:  0.00369 
F-statistic: 1.947 on 3 and 764 DF,  p-value: 0.1206
```

2.2 Q2. (10 Points Total)

For the following questions, use the dataset Berkshire.csv with the following variables:

	Column Name	Description
1	Date	Calendat Date
2	BRKret	Berkshire Hathaway's monthly return
3	MKT	the return on the aggregate stock market
4	RF	the risk free rate of return

You may/may not need the following dependencies:

"lubridate" package which is already installed in R in Vocareum.

"PerformanceAnalytics" package which can be installed as indicated at the beginning of this homework.

Note: The file "Berkshire.csv" can be accessed at the path: ("../resource/asnlib/publicdata/Berkshire.csv"). E.g. read.csv("../resource/asnlib/publicdata/Berkshire.csv")

Return data is in decimal format, 0.01=1%. Round all answers to the nearest hundredth, for example 12.34%.

2.2.1 A. Find the standard deviation of Berkshire Hathaway over the sample period. (2 Points)

```
In [29]: # SOLUTION BEGINS HERE
         if(!require("lubridate")) {install.packages("lubridate")
                                    library("lubridate")}
         if(!require("PerformanceAnalytics")) {install.packages("PerformanceAnalytics")
                                               library("PerformanceAnalytics")}
         if(!require("xts")) {install.packages("xts")
                              library("xts")}
```

```
Loading required package: lubridate


Attaching package: lubridate


The following objects are masked from package:base:

    date, intersect, setdiff, union


Loading required package: PerformanceAnalytics

Loading required package: xts

Loading required package: zoo


Attaching package: zoo


The following objects are masked from package:base:

    as.Date, as.Date.numeric



Attaching package: xts


The following objects are masked from package:dplyr:

    first, last



Attaching package: PerformanceAnalytics
```

```
The following object is masked from package:graphics:

    legend
```

```
In [33]: data = read.csv("../resource/asnlib/publicdata/Berkshire.csv")
         str(data)
         head(data)
```

```
'data.frame':        500 obs. of  4 variables:
 $ Date  : chr  "11/30/1976" "12/31/1976" "1/31/1977" "2/28/1977" ...
 $ BrkRet: num  0.1544 0.1465 0 0 0.0778 ...
 $ MKT   : num  0.0076 0.0605 -0.0369 -0.016 -0.0099 0.0053 -0.0109 0.0511 -0.0127 -0.0131 ...
 $ RF    : num  0.004 0.004 0.0036 0.0035 0.0038 0.0038 0.0037 0.004 0.0042 0.0044 ...
```

A data.frame: 6 Œ 4

	Date	BrkRet	MKT	RF
	<chr>	<dbl>	<dbl>	<dbl>
1	11/30/1976	0.1544	0.0076	0.0040
2	12/31/1976	0.1465	0.0605	0.0040
3	1/31/1977	0.0000	-0.0369	0.0036
4	2/28/1977	0.0000	-0.0160	0.0035
5	3/31/1977	0.0778	-0.0099	0.0038
6	4/30/1977	-0.0103	0.0053	0.0038

```
In [34]: data$Date = mdy(data$Date)
         str(data)
         head(data)
```

```
'data.frame':        500 obs. of  4 variables:
 $ Date  : Date, format: "1976-11-30" "1976-12-31" ...
 $ BrkRet: num  0.1544 0.1465 0 0 0.0778 ...
 $ MKT   : num  0.0076 0.0605 -0.0369 -0.016 -0.0099 0.0053 -0.0109 0.0511 -0.0127 -0.0131 ...
 $ RF    : num  0.004 0.004 0.0036 0.0035 0.0038 0.0038 0.0037 0.004 0.0042 0.0044 ...
```

A data.frame: 6 Œ 4

	Date	BrkRet	MKT	RF
	<date>	<dbl>	<dbl>	<dbl>
1	1976-11-30	0.1544	0.0076	0.0040
2	1976-12-31	0.1465	0.0605	0.0040
3	1977-01-31	0.0000	-0.0369	0.0036
4	1977-02-28	0.0000	-0.0160	0.0035
5	1977-03-31	0.0778	-0.0099	0.0038
6	1977-04-30	-0.0103	0.0053	0.0038

```
In [12]: print(paste0("Standard deviation of Berkshire Hathaway: ", round(sd(data$BrkRet)*100,
```

```
[1] "Standard deviation of Berkshire Hathaway: 6.75%"
```

```
In [ ]: # SOLUTION ENDS HERE
```

2.2.2 B. Find Berkshire Hathaway's average return over the sample period. Use arithmetic average (2 Points)

```
In [16]: # SOLUTION BEGINS HERE
         print(paste0("Average return of Berkshire Hathaway: ", round(mean(data$BrkRet)*100, 2
         # SOLUTION ENDS HERE
```

```
[1] "Average return of Berkshire Hathaway: 1.9%"
```

2.2.3 C. By what percentage per month on average has Berkshire Hathaway outperformed the market? (2 Points)

```
In [28]: # SOLUTION BEGINS HERE
         data = data %>% mutate(outperform = ifelse(BrkRet > MKT, 1, 0))
         print(paste0("Berkshire Hathaway has on average outperformed the market ", sum(data$ou
         # SOLUTION ENDS HERE
```

```
[1] "Berkshire Hathaway has on average outperformed the market 54% of the time"
```

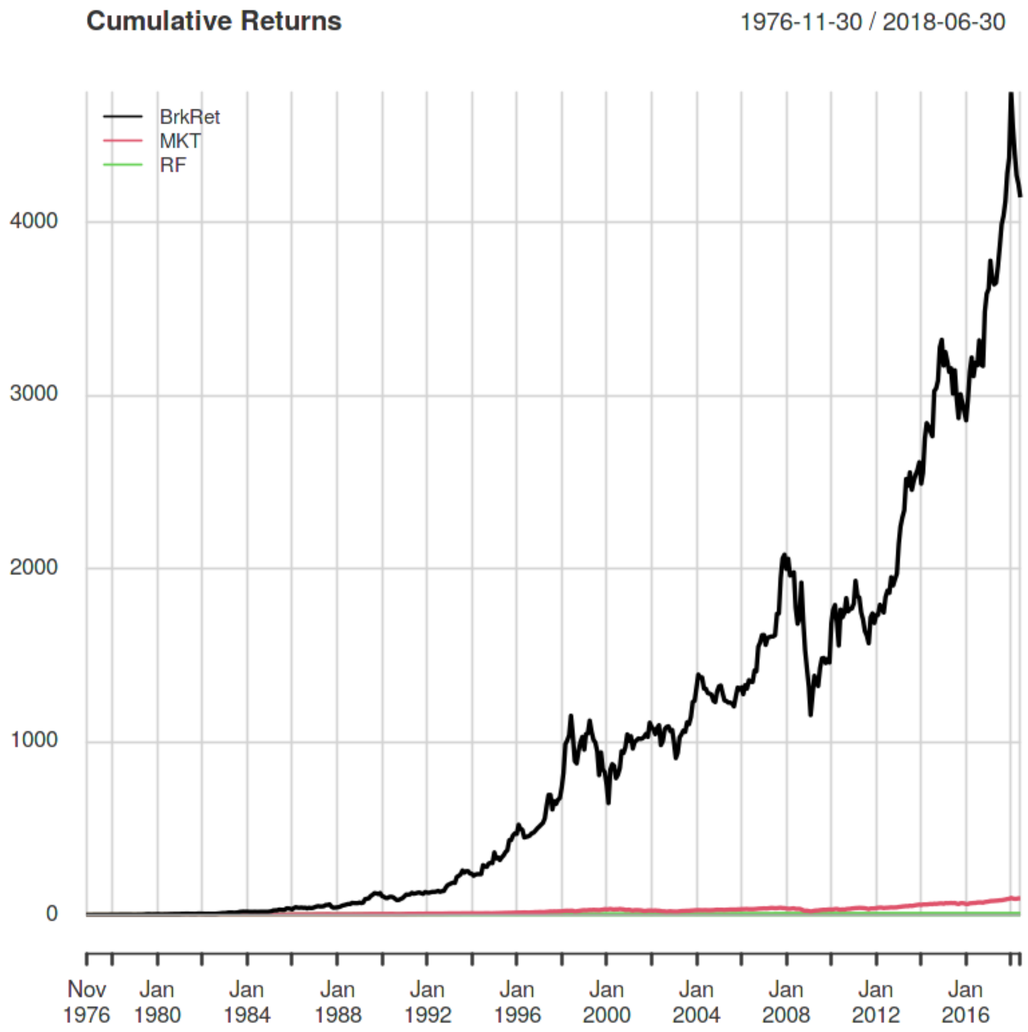
2.2.4 D. If $10,000 were invested in Berkshire Hathaway at the start of the sample period, how much would the investment have grown to by the end of the sample period? (2 Points)

```
In [80]: # SOLUTION BEGINS HERE
         data_ordered = data[order(data$Date),]
         data_xts = xts(data_ordered[,-1], order.by=data_ordered$Date)
         print(paste0("$10,000 investment from the start of the sample period would have grown
                      format(as.numeric((Return.cumulative(data_xts$BrkRet, geometric = TRUE)+
         # SOLUTION ENDS HERE
```

```
[1] "$10,000 investment from the start of the sample period would have grown to $41,439,895.55
```

2.2.5 E. Plot the cumulative return of Berkshire and Market across all years and include a legend. Describe your observation. (2 Points)

```
In [92]: # SOLUTION BEGINS HERE
         chart.CumReturns(data_xts,
                          main = "Cumulative Returns",
                          wealth.index = FALSE,
                          legend.loc = TRUE)
```



The Berkshire fund vastly outperformed (~400,000%) the market (~10,000%) on cumulative returns from the start of the sample period.

```
In [ ]: # SOLUTION ENDS HERE
```