A Comparison of R, Python, and Excel Power Query with Open-Source Financial Data

Ronald Crowe, Kathryn M. Winney, and Michelle I. Avila

Department of Business, School of Business and Leadership

Our Lady of the Lake University

A Comparison of R, Python, and Excel Power Query with Open-Source Financial Data

The purpose of this paper is to illustrate the data mining class exercises that compare Power Query (M), R, and Python. We are teaching a skill set and modeling.[i] The comparison is based on scraping data from Yahoo^{TM} Finance, generating returns, and calculating the beta. We use Python version 3.9.7, R version 4.1.1, and Excel version 365. We illustrate the elements of Power Query that are in the assignment, the method for mining with R, then the method with Python. We will discuss the efficiency of each method and share what we have learned from the experience. We realize there are packages that will mine financial data with minimal input. Our intent is to have students be able to understand the URL to be mined and an efficient code to accomplish the task with an optimal output.

Data Mining

FinTech is here. A recent search at SSRN lists 1601 papers with a "Fintech" search and 723 papers with a "Machine Learning" search. Web data is an important source of information. Machine Learning in finance is a coming trend. There are obvious sources of financial market data that users download whether from CRSP^{TM}, Bloomberg^{TM}, Reuters^{TM}, Yahoo^{TM} Finance, et. al. In addition, alternative data (social media texts, blogs, etc.) are "one of the most important sources of insights for asset managers about market trends and investment opportunities (Dilmegani, 2021)."

Dilmegani lists the following schemes for finance

- analyzing the current status of the financial market
- uncovering market changes and trends
- monitoring national and global news which may affect stocks and economics
- evaluating consumer sentiment and behavior

John Hull, in his editorial on Machine Learning states, “Data scientists will have an increasing impact on business and society in the future and learning the language and the methodologies of data science is going to be increasingly important for all who work in the financial sector (Hull, 2020).”

Why Power Query?

Why Power Query? The answer is in “Why Excel?” Having a desktop application in most business settings is ample reason to learn and use it. Timothy Mayes states that Power Query

> “…is a powerful tool for importing and processing data files. It can import data from Excel workbooks, many databases, text files (.csv, .txt, and .xml), Web sites, and so on. Once the data is imported, it can be transformed in many ways. Furthermore, each step of the processing is recorded like a macro so that it can be refreshed at any time. This tool can literally save a data analyst hours of tedious work (Mayes, 2020).”

Our Lady of the Lake University has a subscription to Microsoft 365. Students and faculty can and do download the Excel and other applications.

Why Python? Why R?

Why Python? Why R? Demand for skills with these specific applications, price, and learning paradigms. Open-source software dominates data science as seen in the KDnuggests Software Poll, Table 1. Python’s % share is 65.8% and R is 46.6.% (Piatestsky, 2019). The absence of SAS, SPSS, or Stata may suggest that those who use those did not respond to the survey. SAS is a leader in the commercial analytical software (Jain, 2017). There are many texts available on using SAS for data mining, e.g., *Getting started with SAS enterprise miner 5.2*. (SAS Institute., 2006) and *Applied data mining for forecasting using SAS* (Rey, 2012). SAS, SPSS, and Stata have various platforms for academic use at less a than full price or free, too. However, the software may not be used at employers' workplaces. Clearly, the advantage of open-source software is its price. This is a factor in academic use. The demand for knowing skills has increased. Except for certification testing, we have opted for Excel, R, and Python over just teaching the financial calculator.

Table 1. Top Analytics/Data Science/ML Software in 2019

Software	2019 % share	2018 % share	2017 % share
Python	65.8%	65.6%	59.0%
RapidMiner	51.2%	52.7%	31.9%
R Language	46.6%	48.5%	56.6%
Excel	34.8%	39.1%	31.5%
Anaconda	33.9%	33.4%	24.3%
SQL Language	32.8%	39.6%	39.2%
Tensorflow	31.7%	29.9%	22.7%
Keras	26.6%	22.2%	10.7%
scikit-learn	25.5%	24.4%	21.9%
Tableau	22.1%	26.4%	21.8%
Apache Spark	21.0%	21.5%	25.5%
Source: KDNuggets (Piatestsky, 2019)			

Hull concludes that "coding skills (ideally Python) and some knowledge of data science are necessary for obtaining and holding down jobs in the finance sector (Hull, 2020)."

Yan advocates for R because it serves as a learning environment (Yan, 2022). I purport that web searching for "How do I do …. In R?" will bring up coding solutions and videos. YouTube, as well as other online video sources, support the use of Excel, R, and Python.

Yahoo Finance Mining

The elements of mining, or scraping, the stock price data at Yahoo™ involve selecting the ticker, the start and end dates, interval, and whether one wants the adjusted closing price.[ii] The URL for daily stock prices of Apple between August 1, 2021, and August 1, 2022, follows (251 trading days).

https://query1.finance.yahoo.com/v7/finance/download/AAPL?period1=1629651662&period2=1661187662&interval=1d&events=history&includeAdjustedClose=true

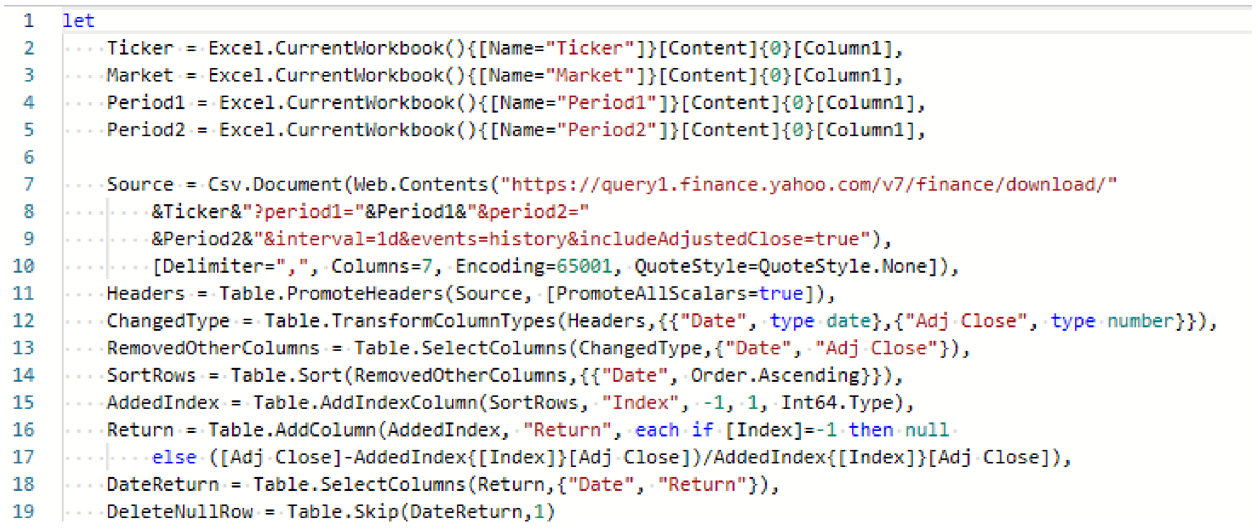
The Ticker is the stock identifier on the exchange where the stock trades. The periods in the URL are based on Unix Epoch Time, UET, based on the number of seconds from a base of January 1, 1970 (Unix time was zero).

The Power Query Approach

We teach the students how to accomplish a Get and Transform in Excel. [iii] We progress to using Excel to calculate the returns and then the beta with slope function. We then show them how create a power query in M ("advance editor") (Microsoft Corporation, 2022). [iv]

For this paper, we created a table in Excel with named cells to enter the Ticker, the market proxy ("^GSPC" in this case), and the start and end date in Excel format. The table has an Excel formula to change the Excel data to UET. In Figure 1, lines 2 to Lines 5 use the data entered in the spreadsheet. Lines 7-10 get the data from an edited URL request. Line 12 changes the strings to the dates and numbers. Line 14 removes other columns and leaves us with Date and Adjusted Close price. Lines 16 and 17 create the return data. Then, the first row of data (with no return) is deleted.

Figure 1. Stock Price Query



```
1   let
2       Ticker = Excel.CurrentWorkbook(){[Name="Ticker"]}[Content]{0}[Column1],
3       Market = Excel.CurrentWorkbook(){[Name="Market"]}[Content]{0}[Column1],
4       Period1 = Excel.CurrentWorkbook(){[Name="Period1"]}[Content]{0}[Column1],
5       Period2 = Excel.CurrentWorkbook(){[Name="Period2"]}[Content]{0}[Column1],
6
7       Source = Csv.Document(Web.Contents("https://query1.finance.yahoo.com/v7/finance/download/"
8           &Ticker&"?period1="&Period1&"&period2="
9           &Period2&"&interval=1d&events=history&includeAdjustedClose=true"),
10          [Delimiter=",", Columns=7, Encoding=65001, QuoteStyle=QuoteStyle.None]),
11      Headers = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
12      ChangedType = Table.TransformColumnTypes(Headers,{{"Date", type date},{"Adj Close", type number}}),
13      RemovedOtherColumns = Table.SelectColumns(ChangedType,{"Date", "Adj Close"}),
14      SortRows = Table.Sort(RemovedOtherColumns,{{"Date", Order.Ascending}}),
15      AddedIndex = Table.AddIndexColumn(SortRows, "Index", -1, 1, Int64.Type),
16      Return = Table.AddColumn(AddedIndex, "Return", each if [Index]=-1 then null
17          else ([Adj Close]-AddedIndex{[Index]}[Adj Close])/AddedIndex{[Index]}[Adj Close]),
18      DateReturn = Table.SelectColumns(Return,{"Date", "Return"}),
19      DeleteNullRow = Table.Skip(DateReturn,1)
```

As the students participate in the assignments, we show them how to create a function in M, Figure 2. The code “(Ticker,StartDate,EndDate) =>” will ask for these parameters to be entered when the function is used – “invoked” in M. In Figure 2, the UET periods are calculated (lines 3 and 4).

```
Period1 = Number.ToText(Duration.TotalSeconds(StartDate-#datetime(1970,1,1,0,0,0))),
Period2 = Number.ToText(Duration.TotalSeconds(EndDate-#datetime(1970,1,1,0,0,0))),
```

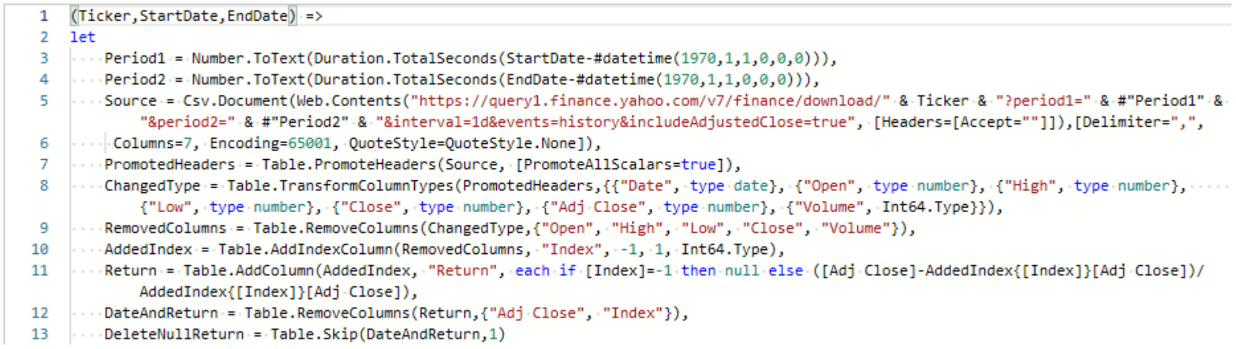
We use a method suggested by Damodaran to calculate the CAPM Beta. The difference in the price (including dividends) consecutively divided by the previous price. Damodaran warns that daily, weekly, and monthly estimations will have different results for Beta (Damodaran, 2014). The return code in line 11 is

```
Return = Table.AddColumn(AddedIndex, "Return", each if [Index]=-1 then null else ([Adj Close]-
AddedIndex{[Index]}[Adj Close])/AddedIndex{[Index]}[Adj Close])
```

The Index column is to use the previous price in the return. The “if” command returns null as a value for the first row. The math is just a simple one period return between the parentheses. The resulting table appears in the spreadsheet. See the first 10 returns for AAPL in Figure 3, Panel (a).

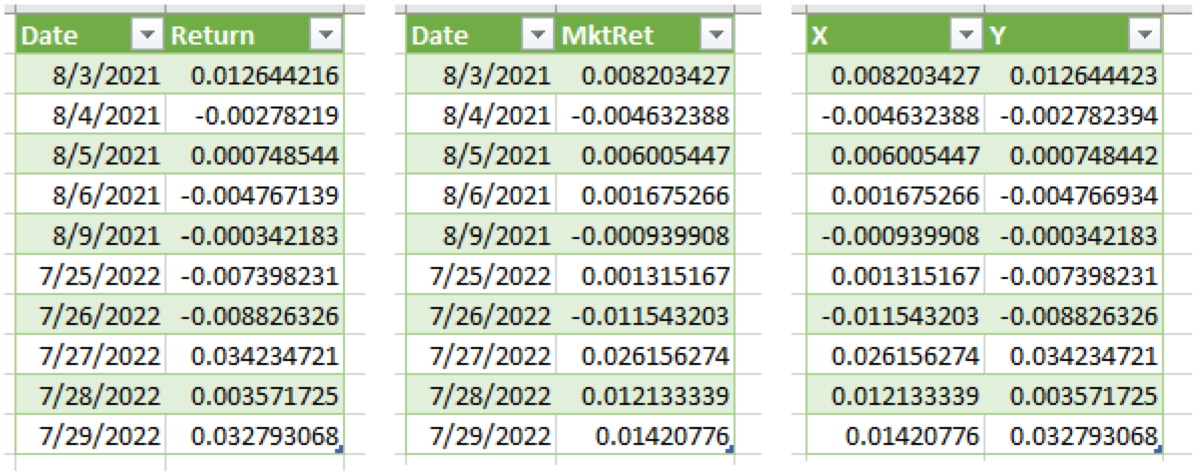
Figure 2. Function of Stock Price Query

c



```
1   (Ticker,StartDate,EndDate) =>
2   let
3       Period1 = Number.ToText(Duration.TotalSeconds(StartDate-#datetime(1970,1,1,0,0,0))),
4       Period2 = Number.ToText(Duration.TotalSeconds(EndDate-#datetime(1970,1,1,0,0,0))),
5       Source = Csv.Document(Web.Contents("https://query1.finance.yahoo.com/v7/finance/download/" & Ticker & "?period1=" & #"Period1" &
            "&period2=" & #"Period2" & "&interval=1d&events=history&includeAdjustedClose=true", [Headers=[Accept=""]]),[Delimiter=",",
6        Columns=7, Encoding=65001, QuoteStyle=QuoteStyle.None]),
7       PromotedHeaders = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
8       ChangedType = Table.TransformColumnTypes(PromotedHeaders,{{"Date", type date}, {"Open", type number}, {"High", type number},
            {"Low", type number}, {"Close", type number}, {"Adj Close", type number}, {"Volume", Int64.Type}}),
9       RemovedColumns = Table.RemoveColumns(ChangedType,{"Open", "High", "Low", "Close", "Volume"}),
10      AddedIndex = Table.AddIndexColumn(RemovedColumns, "Index", -1, 1, Int64.Type),
11      Return = Table.AddColumn(AddedIndex, "Return", each if [Index]=-1 then null else ([Adj Close]-AddedIndex{[Index]}[Adj Close])/
            AddedIndex{[Index]}[Adj Close]),
12      DateAndReturn = Table.RemoveColumns(Return,{"Adj Close", "Index"}),
13      DeleteNullReturn = Table.Skip(DateAndReturn,1)
```

Figure 3. Daily Returns by Date



Date	Return
8/3/2021	0.012644216
8/4/2021	-0.00278219
8/5/2021	0.000748544
8/6/2021	-0.004767139
8/9/2021	-0.000342183
7/25/2022	-0.007398231
7/26/2022	-0.008826326
7/27/2022	0.034234721
7/28/2022	0.003571725
7/29/2022	0.032793068

Panel (a) AAPL

Date	MktRet
8/3/2021	0.008203427
8/4/2021	-0.004632388
8/5/2021	0.006005447
8/6/2021	0.001675266
8/9/2021	-0.000939908
7/25/2022	0.001315167
7/26/2022	-0.011543203
7/27/2022	0.026156274
7/28/2022	0.012133339
7/29/2022	0.01420776

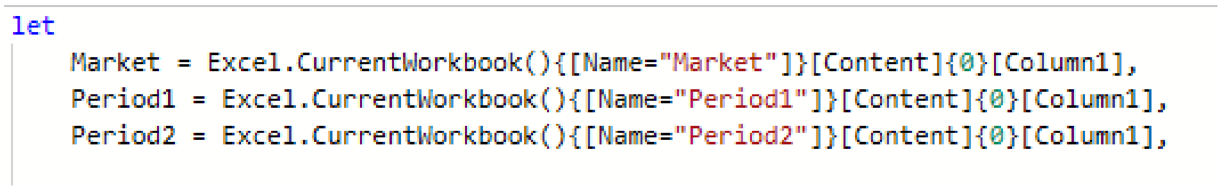
Panel (b) ^GSPC

X	Y
0.008203427	0.012644423
-0.004632388	-0.002782394
0.006005447	0.000748442
0.001675266	-0.004766934
-0.000939908	-0.000342183
0.001315167	-0.007398231
-0.011543203	-0.008826326
0.026156274	0.034234721
0.012133339	0.003571725
0.01420776	0.032793068

Panel (c) X = ^GSPC and Y = AAPL

The process is continued to produce the Market returns as shown in Figure 4. Note the *M* code uses the named range "Market" in the Excel spreadsheet in this case. The results of the first 10 returns are shown in Figure 3, Panel (b).

Figure 4. Market Return Query.



```
let
    Market = Excel.CurrentWorkbook(){[Name="Market"]}[Content]{0}[Column1],
    Period1 = Excel.CurrentWorkbook(){[Name="Period1"]}[Content]{0}[Column1],
    Period2 = Excel.CurrentWorkbook(){[Name="Period2"]}[Content]{0}[Column1],
```

In preparing for linear regression, the data is merged to a table, Figure 5, panel (a). We convert the columns names to X and Y. Then, we remove the Date column.

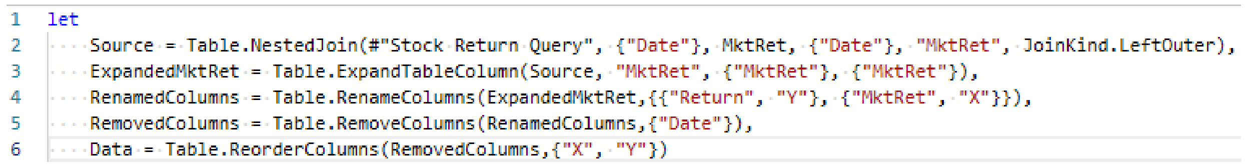
The regression is done with the computational method[v] with equation below.

$$Slope = \hat{\beta}_1 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})\,(Y_i - \bar{Y})}{\sum_{i=1}^{n}(X_i - \bar{X})^2} = \frac{n(\sum_{i=1}^{n} XY) - (\sum_{i=1}^{n} X)(\sum_{1=1}^{n} Y)}{n(\sum_{i=1}^{n} X^2) - \left(\sum_{i=1}^{n} X\right)^2}$$

$$Intercept = \hat{\beta}_0 = \bar{Y} - \hat{\beta}_1\bar{X} = \frac{\left(\sum_{i=1}^{n} Y - \hat{\beta}_1(\sum_{i=1}^{n} X)\right)}{n}$$

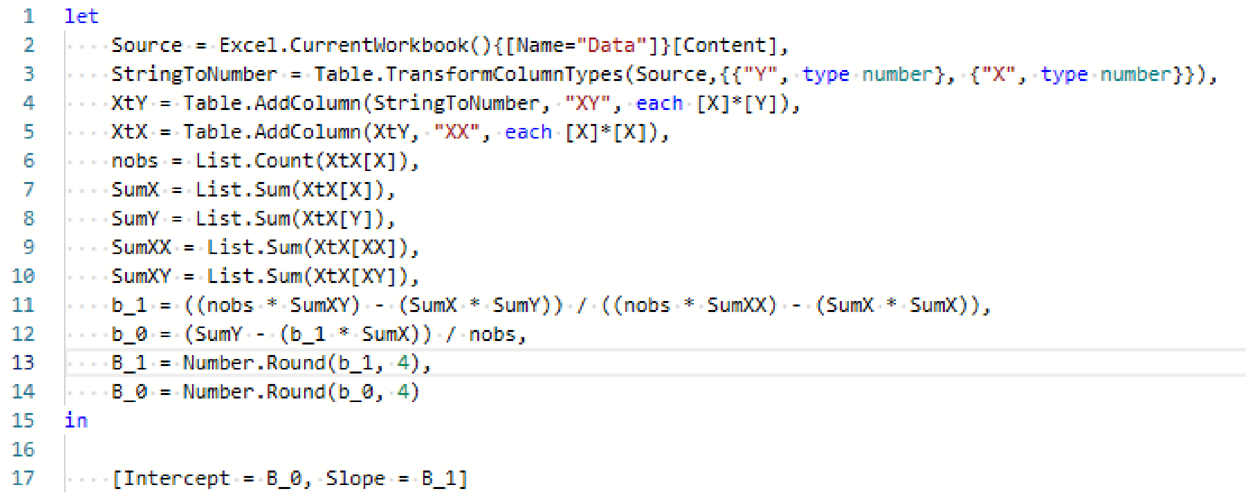
The code to estimate the intercept and the slope in Figure 5, Panel (b) shows a process of creating columns, i.e., ***XtX***, ***XtY***. Then those columns are summed, e.g., ***SumX***, ***SumY***, ***SumXX***, and ***SumXY***. The calculation for the slope, $\hat{\beta}_1$, and intercept, $\hat{\beta}_0$, occur in lines 11 and 12.

Figure 5. Merging & Regression.



```
let
    Source = Table.NestedJoin(#"Stock Return Query", {"Date"}, MktRet, {"Date"}, "MktRet", JoinKind.LeftOuter),
    ExpandedMktRet = Table.ExpandTableColumn(Source, "MktRet", {"MktRet"}, {"MktRet"}),
    RenamedColumns = Table.RenameColumns(ExpandedMktRet,{{"Return", "Y"}, {"MktRet", "X"}}),
    RemovedColumns = Table.RemoveColumns(RenamedColumns,{"Date"}),
    Data = Table.ReorderColumns(RemovedColumns,{"X", "Y"})
```

Panel (a) Merging



```
1   let
2       Source = Excel.CurrentWorkbook(){[Name="Data"]}[Content],
3       StringToNumber = Table.TransformColumnTypes(Source,{{"Y", type number}, {"X", type number}}),
4       XtY = Table.AddColumn(StringToNumber, "XY", each [X]*[Y]),
5       XtX = Table.AddColumn(XtY, "XX", each [X]*[X]),
6       nobs = List.Count(XtX[X]),
7       SumX = List.Sum(XtX[X]),
8       SumY = List.Sum(XtX[Y]),
9       SumXX = List.Sum(XtX[XX]),
10      SumXY = List.Sum(XtX[XY]),
11      b_1 = ((nobs * SumXY) - (SumX * SumY)) / ((nobs * SumXX) - (SumX * SumX)),
12      b_0 = (SumY - (b_1 * SumX)) / nobs,
13      B_1 = Number.Round(b_1, 4),
14      B_0 = Number.Round(b_0, 4)
15  in
16
17      [Intercept = B_0, Slope = B_1]
```

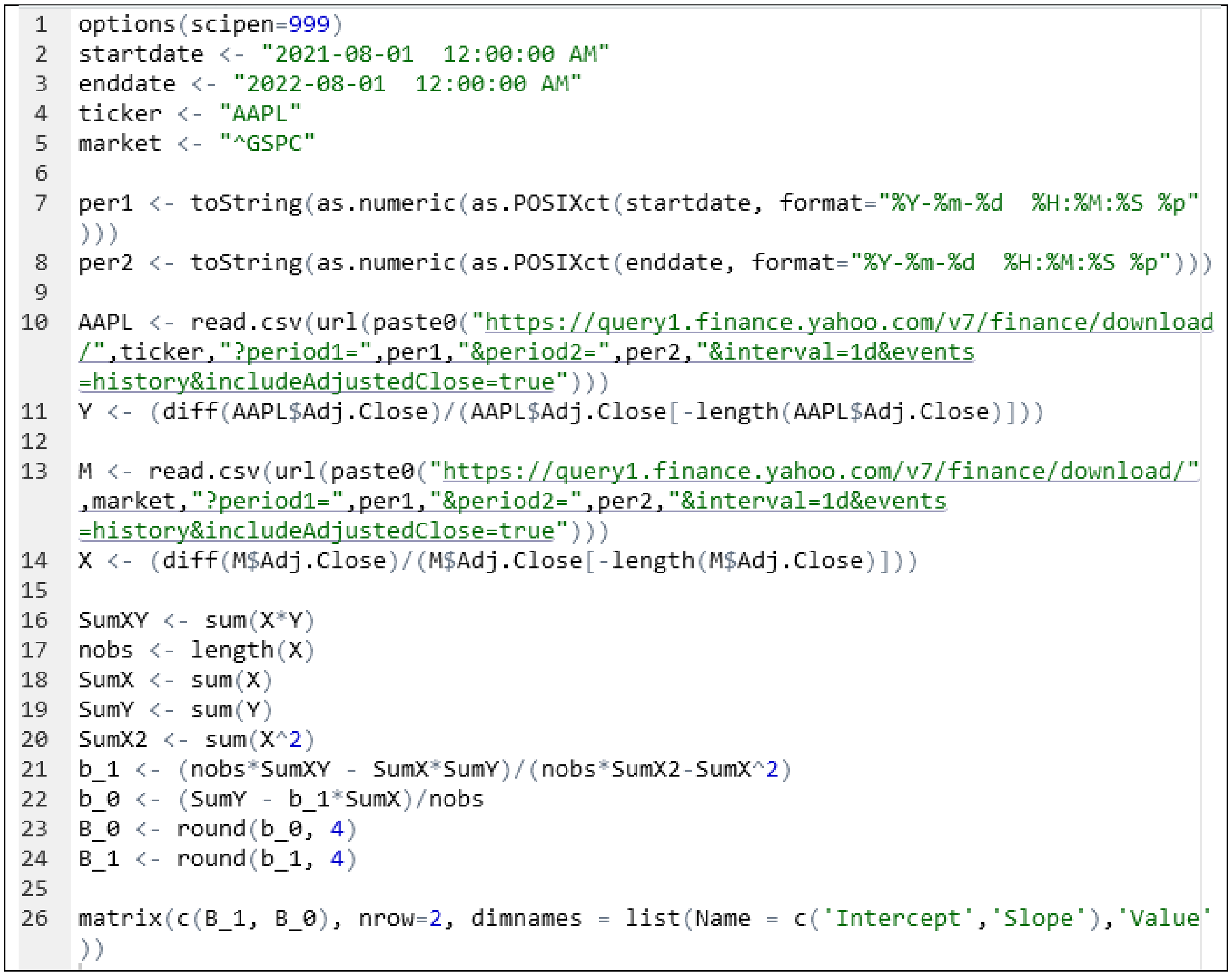
Panel (b) Regression

The intercept and slope are rounded to four decimal places. The output table is coded in line 17 in Panel (b), see Table 8, Panel (a). The beta is 1.2631 and the intercept is 0.0008.

The R Approach

For the application in R, only the base package is used; no other packages are needed (R Core Team, 2013). In the first line of the R script, an option command is done to show values without scientific notation. Next, the ticker, market, start date, and end date are entered. The dates are converted UET. The data is read, and the returns are calculated. The regression inputs are set up for the computational method. Then, the intercept and slope are calculated. The data are rounded to four decimal places. Finally, a table is formed for output. The R script is run in R Studio (R Studio Team, 2020).

Figure 6. The R script file.

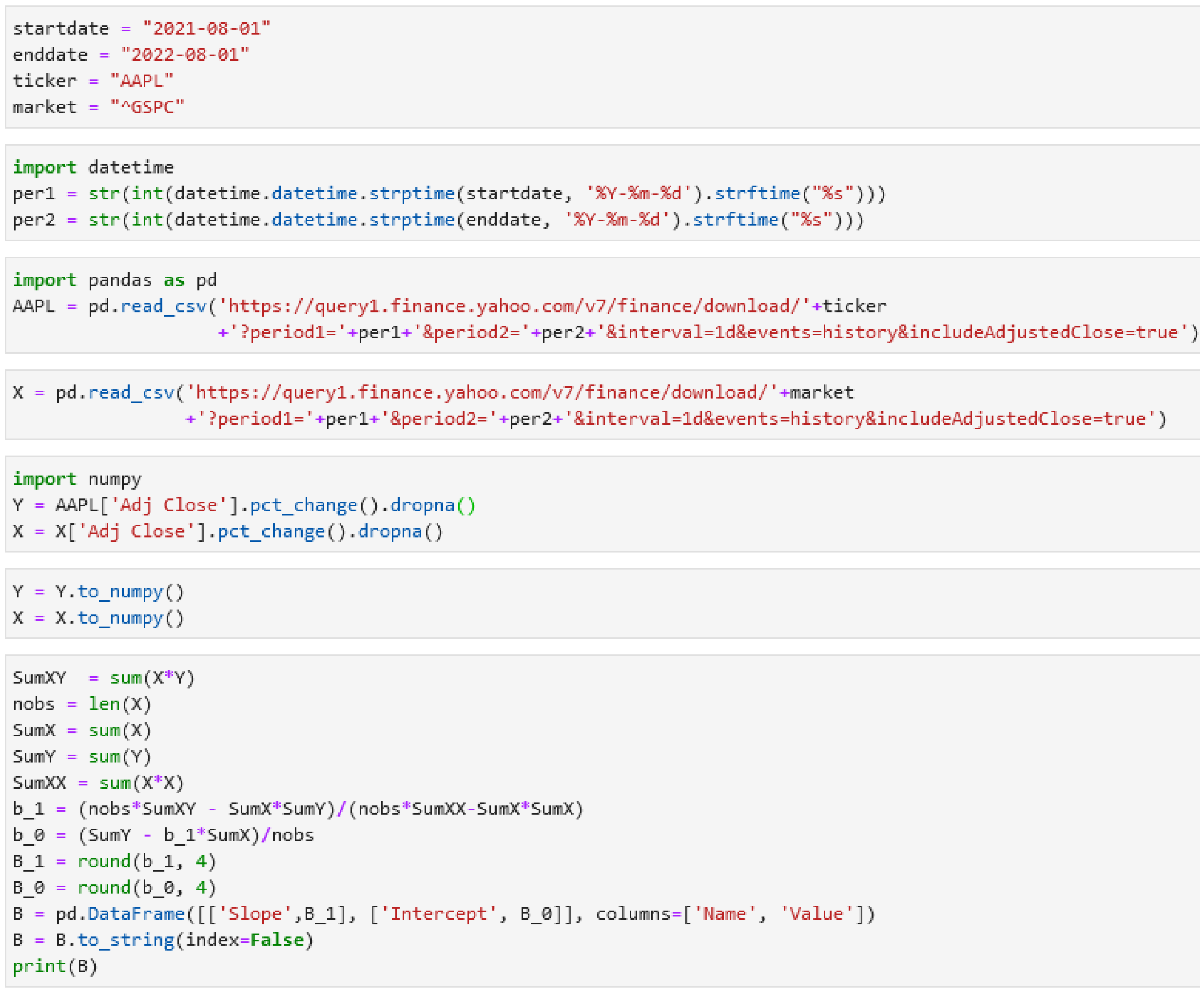


```r
options(scipen=999)
startdate <- "2021-08-01  12:00:00 AM"
enddate <- "2022-08-01  12:00:00 AM"
ticker <- "AAPL"
market <- "^GSPC"

per1 <- toString(as.numeric(as.POSIXct(startdate, format="%Y-%m-%d  %H:%M:%S %p"
)))
per2 <- toString(as.numeric(as.POSIXct(enddate, format="%Y-%m-%d  %H:%M:%S %p")))

AAPL <- read.csv(url(paste0("https://query1.finance.yahoo.com/v7/finance/download
/",ticker,"?period1=",per1,"&period2=",per2,"&interval=1d&events
=history&includeAdjustedClose=true")))
Y <- (diff(AAPL$Adj.Close)/(AAPL$Adj.Close[-length(AAPL$Adj.Close)]))

M <- read.csv(url(paste0("https://query1.finance.yahoo.com/v7/finance/download/"
,market,"?period1=",per1,"&period2=",per2,"&interval=1d&events
=history&includeAdjustedClose=true")))
X <- (diff(M$Adj.Close)/(M$Adj.Close[-length(M$Adj.Close)]))

SumXY <- sum(X*Y)
nobs <- length(X)
SumX <- sum(X)
SumY <- sum(Y)
SumX2 <- sum(X^2)
b_1 <- (nobs*SumXY - SumX*SumY)/(nobs*SumX2-SumX^2)
b_0 <- (SumY - b_1*SumX)/nobs
B_0 <- round(b_0, 4)
B_1 <- round(b_1, 4)

matrix(c(B_1, B_0), nrow=2, dimnames = list(Name = c('Intercept','Slope'),'Value'
))
```

The Python Approach

The process in Python is like M and R. The Python coding first sets variables as strings. (Van Rossum, 1995). For the teaching of the coding process, students use a “Jupyter Notebook” to allow sectioning the code into relevant parts (Kluyver, 2016). Three python modules are needed: Datetime. Numpy (Harris, 2020). Pandas (McKinney, 2010) via import commands. The Datetime module has the commands to turn the date string into UET. The Pandas module will mine the data by URL. The Numpy module commands are used to create the returns, drop the row without a return, and change an indexed data frame to an array for mathematical operations. After the intercept and slope are estimated, a Pandas command sets up the output of a table.

Figure 7. Python Code



```
startdate = "2021-08-01"
enddate = "2022-08-01"
ticker = "AAPL"
market = "^GSPC"
```

```
import datetime
per1 = str(int(datetime.datetime.strptime(startdate, '%Y-%m-%d').strftime("%s")))
per2 = str(int(datetime.datetime.strptime(enddate, '%Y-%m-%d').strftime("%s")))
```

```
import pandas as pd
AAPL = pd.read_csv('https://query1.finance.yahoo.com/v7/finance/download/'+ticker
                   +'?period1='+per1+'&period2='+per2+'&interval=1d&events=history&includeAdjustedClose=true')
```

```
X = pd.read_csv('https://query1.finance.yahoo.com/v7/finance/download/'+market
                +'?period1='+per1+'&period2='+per2+'&interval=1d&events=history&includeAdjustedClose=true')
```

```
import numpy
Y = AAPL['Adj Close'].pct_change().dropna()
X = X['Adj Close'].pct_change().dropna()
```

```
Y = Y.to_numpy()
X = X.to_numpy()
```

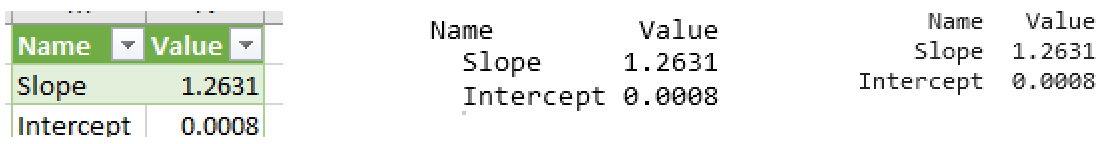
```
SumXY  = sum(X*Y)
nobs = len(X)
SumX = sum(X)
SumY = sum(Y)
SumXX = sum(X*X)
b_1 = (nobs*SumXY - SumX*SumY)/(nobs*SumXX-SumX*SumX)
b_0 = (SumY - b_1*SumX)/nobs
B_1 = round(b_1, 4)
B_0 = round(b_0, 4)
B = pd.DataFrame([['Slope',B_1], ['Intercept', B_0]], columns=['Name', 'Value'])
B = B.to_string(index=False)
print(B)
```

Comparison between methods

Having the visual of data in an Excel spreadsheet is an obvious difference to using Excel versus R or Python. The use of the ticker, market, start date and end date represent string or text information. All the methods treat these in similar fashions. Converting to UET requires coding unique to each method. The concatenation of the URL coding requires ampersands in M, commas in R, and plus signs in Pandas.

The data in M is treated by default as a string. Coding it as numeric is required before math operations. Quantitative data variables are columns in M, vectors in R , and arrays in Numpy. The number of observations calculation is a "List.count()" command in M, and a length function in R and Ptyhon ("length()" and "len()", respectively). The summation functions to set up the computation of the intercept and slope are the same in R and Ptyhon with a sum() function. In M, the command requires List.sum() function. The intercept and slope calculations are similar for the most part. In M and Python the sum of X squared requires multiplication, in R, squaring with "^2" is allowed.
The output of the code is in Figure 6. The Beta for AAPL over the period was 1.2631 for all three methods.

Figure 8. Regression output.



Name	Value
Slope	1.2631
Intercept	0.0008

```
Name           Value
  Slope       1.2631
  Intercept  0.0008
```

```
     Name    Value
    Slope   1.2631
Intercept   0.0008
```

Panel (a) M Panel (b) R Panel (c) Python

With the use of M, the output is loaded to a location in Excel as a table. The color and style are arbitrarily chosen. The table header is set up to filter in Excel. These allow users to sort and show column information based on content related searches. These functions would be coded in both R and Python.

In R, the output table is accomplished with a “matrix()” command to label the column and rows. In Python, Pandas “.Dataframe()” command is used. There are other ways to accomplish this in both R and Python, some require packages for R and modules for Python.

Student response

Students have two assignments. The first assignment is to create an Excel spreadsheet, use Power Query to get prices, create returns and find the beta. The second assignment is to use R and Python for the process. The following are comments on the exercises.

“It was an easier way than searching for the data on Yahoo Finance and importing it manually.”

“I enjoy looking at the regression models”

“it looks like another all-purpose programming language that would be a great introduction to coding. It was simple enough to read and understand and I think that had I used this language when first becoming familiar with coding, I would have learned quickly and understood some of the coding basics more easily.”

Conclusion

We present three methods to mine financial data. We use the respective systems to calculate returns of a stock and a market index. We used linear regression to find the CAPM Beta. The computational method is used to find the ordinary least squares estimation of the slope and intercept. All methods reported the same numbers. The process in M requires slightly more coding than do R and Python. The tradeoff of using Excel is having visual results in the spreadsheet. Student comments were positive. For future analysis, a survey of students on views of efficiency and preference is planned.

[i] Appendix A has the MS in Business Analytics program objectives
[ii] “AdjustedClose” is the closing price that accounts for dividends and stock splits (Yahoo, n.d.).
[iii] Excel 365 has a function to insert historical data in an array. See https://support.microsoft.com/en-us/office/stockhistory-function-1ac8b5b3-5f62-4d94-8ab8-7504ec7239a8.
[iv] The amount of code will be discussed shortly. The amount of code is reduced as student’s learn M.
[v] There are no power function operators in Power Query.

References

Damodaran, A. (2014). *Applied corporate finance* (Fourth Edition ed.). Hoboken, NJ: Wiley Global Education US.

Dilmegani, D. (2021, August 19). *In-depth guide to web scraping for finance in 2022*. Retrieved Aug 1, 2022, from AIMultiple: https://research.aimultiple.com/web-scraping-for-finance/cond

Harris, C. M. (2020). Array programming with NumPy. *Nature 585*, 357–362. doi:10.1038/s41586-020-2649-2

Hull, J. (2020). Machine learning and finance. *Journal of Risk Management in Financial Institutions, 13*(2), 104-105.

Jain, K. (2017). *Python vs. R vs. SAS – which tool should I learn for data science?* Retrieved from www.analyticsvidhya.com/: https://www.analyticsvidhya.com/blog/2017/09/sas-vs-vs-python-tool-learn/

Kluyver, T. R.-K. (2016). Jupyter notebooks – a publishing format for reproducible computational workflows. (F. Loizides, & B. Schmidt, Eds.) *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pp. 87-90.

Mayes, T. (2020). *Financial analysis with Microsoft Excel* (9th ed.). New York: Cengage Learning US.

McKinney, W. &. (2010). Data structures for statistical computing in Python. *In Proceedings of the 9th Python in Science Conference, 445*, 51-56.

Microsoft Corporation. (2022). *Microsoft Excel for Microsoft 365.* Retrieved from https://office.microsoft.com/excel

Piatestsky, G. (2019, May 30). *KDnuggets*. Retrieved from Python leads the 11 top data science, machine learning platforms: Trends and Analysis: https://www.kdnuggets.com/2019/05/poll-top-data-science-machine-learning-platforms.html

R Core Team. (2013). *R: A language and environment for statistical computing.* Vienna, Austria: R Foundation for Statistical Computing. Retrieved from http://www.R-project.org

R Studio Team. (2020). *RStudio: Integrated Development for R*. Retrieved from http://www.rstudio.com/.

Rey, T. K. (2012). *Applied data mining for forecasting using SAS.* SAS Institute.

SAS Institute. (2006). *Getting started with SAS enterprise miner 5.2.* SAS Publications.

Van Rossum, G. &. (1995). *Python reference manual.* . Amsterdam: Centrum voor Wiskunde en Informatica.

Yahoo. (n.d.). *What is the adjusted close?* Retrieved from in.help.yahoo.com: https://in.help.yahoo.com/kb/adjusted-close-sln28256.html?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAEbGoBOwYYMquYGZ72fOdxwBqUxwyVHoia7P5Dpj_2XZ-FcWx2b8Pc8NeAhYnq6qFLHLp5acgjMMXNFhAItWne20Fayf1Di2od4j0kwtQtN5IhteNAlo

Yan, Y. (2022, 6 1). An innovative way to teach courses in finance, economics, and data analytics.