Case Study Down Jones

Leonel Salazar, Seth Harris, Joaquin Ramirez, Collin Real

Warning: package 'ggplot2' was built under R version 4.3.3

Warning: package 'tidyr' was built under R version 4.3.3

Warning: package 'dplyr' was built under R version 4.3.3

Warning: package 'stringr' was built under R version 4.3.2

Warning: package 'lubridate' was built under R version 4.3.2

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.4 ✔ readr 2.1.4  
✔ forcats 1.0.0 ✔ stringr 1.5.1  
✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
✔ purrr 1.0.2   
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

Warning: package 'here' was built under R version 4.3.3

here() starts at C:/Users/Leonel/Desktop/MSDA/MSDA/MS Data Analytics/Current Class/DA 6813/Week 6  
  
Attaching package: 'gridExtra'  
  
The following object is masked from 'package:dplyr':  
  
 combine

Warning: package 'tseries' was built under R version 4.3.3

Registered S3 method overwritten by 'quantmod':  
 method from  
 as.zoo.data.frame zoo   
Loading required package: xts  
Loading required package: zoo  
  
Attaching package: 'zoo'  
  
The following objects are masked from 'package:base':  
  
 as.Date, as.Date.numeric  
  
  
######################### Warning from 'xts' package ##########################  
# #  
# The dplyr lag() function breaks how base R's lag() function is supposed to #  
# work, which breaks lag(my\_xts). Calls to lag(my\_xts) that you type or #  
# source() into this session won't work correctly. #  
# #  
# Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #  
# conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop #  
# dplyr from breaking base R's lag() function. #  
# #  
# Code in packages is not affected. It's protected by R's namespace mechanism #  
# Set `options(xts.warn\_dplyr\_breaks\_lag = FALSE)` to suppress this warning. #  
# #  
###############################################################################  
  
Attaching package: 'xts'  
  
The following objects are masked from 'package:dplyr':  
  
 first, last  
  
Loading required package: TTR

Warning: package 'TTR' was built under R version 4.3.2

[1] "C:/Users/Leonel/Desktop/MSDA/MSDA/MS Data Analytics/Current Class/DA 6813/Week 6"

quarter stock date open high low close volume  
1 1 AA 1/7/2011 $15.82 $16.72 $15.78 $16.42 239655616  
2 1 AA 1/14/2011 $16.71 $16.71 $15.64 $15.97 242963398  
3 1 AA 1/21/2011 $16.19 $16.38 $15.60 $15.79 138428495  
4 1 AA 1/28/2011 $15.87 $16.63 $15.82 $16.13 151379173  
5 1 AA 2/4/2011 $16.18 $17.39 $16.18 $17.14 154387761  
6 1 AA 2/11/2011 $17.33 $17.48 $16.97 $17.37 114691279  
 percent\_change\_price percent\_change\_volume\_over\_last\_wk previous\_weeks\_volume  
1 3.792670 $16.71 $15.97   
2 -4.428490 1.380223028 239655616  
3 -2.470660 -43.02495926 242963398  
4 1.638310 9.355500109 138428495  
5 5.933250 1.987451735 151379173  
6 0.230814 -25.71219489 154387761  
 next\_weeks\_open next\_weeks\_close percent\_change\_next\_weeks\_price  
1 -4.42849 26 0.182704  
2 $16.19 $15.79 -2.470660  
3 $15.87 $16.13 1.638310  
4 $16.18 $17.14 5.933250  
5 $17.33 $17.37 0.230814  
6 $17.39 $17.28 -0.632547  
 days\_to\_next\_dividend percent\_return\_next\_dividend  
1 NA NA  
2 19 0.187852  
3 12 0.189994  
4 5 0.185989  
5 97 0.175029  
6 90 0.172712

'data.frame': 750 obs. of 16 variables:  
 $ quarter : int 1 1 1 1 1 1 1 1 1 1 ...  
 $ stock : chr "AA" "AA" "AA" "AA" ...  
 $ date : chr "1/7/2011" "1/14/2011" "1/21/2011" "1/28/2011" ...  
 $ open : chr "$15.82 " "$16.71 " "$16.19 " "$15.87 " ...  
 $ high : chr "$16.72 " "$16.71 " "$16.38 " "$16.63 " ...  
 $ low : chr "$15.78 " "$15.64 " "$15.60 " "$15.82 " ...  
 $ close : chr "$16.42 " "$15.97 " "$15.79 " "$16.13 " ...  
 $ volume : int 239655616 242963398 138428495 151379173 154387761 114691279 80023895 132981863 109493077 114332562 ...  
 $ percent\_change\_price : num 3.79 -4.43 -2.47 1.64 5.93 ...  
 $ percent\_change\_volume\_over\_last\_wk: chr "$16.71 " "1.380223028" "-43.02495926" "9.355500109" ...  
 $ previous\_weeks\_volume : chr "$15.97 " "239655616" "242963398" "138428495" ...  
 $ next\_weeks\_open : chr "-4.42849" "$16.19 " "$15.87 " "$16.18 " ...  
 $ next\_weeks\_close : chr "26" "$15.79 " "$16.13 " "$17.14 " ...  
 $ percent\_change\_next\_weeks\_price : num 0.183 -2.471 1.638 5.933 0.231 ...  
 $ days\_to\_next\_dividend : int NA 19 12 5 97 90 83 76 69 62 ...  
 $ percent\_return\_next\_dividend : num NA 0.188 0.19 0.186 0.175 ...

quarter stock   
 0 0   
 date open   
 0 0   
 high low   
 0 0   
 close volume   
 0 0   
 percent\_change\_price percent\_change\_volume\_over\_last\_wk   
 0 0   
 previous\_weeks\_volume next\_weeks\_open   
 0 0   
 next\_weeks\_close percent\_change\_next\_weeks\_price   
 0 0   
 days\_to\_next\_dividend percent\_return\_next\_dividend   
 30 30

'data.frame': 750 obs. of 16 variables:  
 $ quarter : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...  
 $ stock : Factor w/ 30 levels "AA","AXP","BA",..: 1 1 1 1 1 1 1 1 1 1 ...  
 $ date : Date, format: "2011-01-07" "2011-01-14" ...  
 $ open : num 15.8 16.7 16.2 15.9 16.2 ...  
 $ high : num 16.7 16.7 16.4 16.6 17.4 ...  
 $ low : num 15.8 15.6 15.6 15.8 16.2 ...  
 $ close : num 16.4 16 15.8 16.1 17.1 ...  
 $ volume : num 2.40e+08 2.43e+08 1.38e+08 1.51e+08 1.54e+08 ...  
 $ percent\_change\_price : num 3.79 -4.43 -2.47 1.64 5.93 ...  
 $ percent\_change\_volume\_over\_last\_wk: num NA 1.38 -43.02 9.36 1.99 ...  
 $ previous\_weeks\_volume : num 1.60e+01 2.40e+08 2.43e+08 1.38e+08 1.51e+08 ...  
 $ next\_weeks\_open : num 4.43 16.19 15.87 16.18 17.33 ...  
 $ next\_weeks\_close : num 26 15.8 16.1 17.1 17.4 ...  
 $ percent\_change\_next\_weeks\_price : num 0.183 -2.471 1.638 5.933 0.231 ...  
 $ days\_to\_next\_dividend : num NA 19 12 5 97 90 83 76 69 62 ...  
 $ percent\_return\_next\_dividend : num NA 0.188 0.19 0.186 0.175 ...

Loaded data for stock: AA   
Loaded data for stock: AXP   
Loaded data for stock: BA   
Loaded data for stock: BAC   
Loaded data for stock: CAT   
Loaded data for stock: CSCO   
Loaded data for stock: CVX   
Loaded data for stock: DD   
Loaded data for stock: DIS   
Loaded data for stock: GE   
Loaded data for stock: HD   
Loaded data for stock: HPQ   
Loaded data for stock: IBM   
Loaded data for stock: INTC   
Loaded data for stock: JNJ   
Loaded data for stock: JPM   
Loaded data for stock: KRFT   
Loaded data for stock: KO   
Loaded data for stock: MCD   
Loaded data for stock: MMM   
Loaded data for stock: MRK   
Loaded data for stock: MSFT   
Loaded data for stock: PFE   
Loaded data for stock: PG   
Loaded data for stock: T   
Loaded data for stock: TRV   
Loaded data for stock: UTX   
Loaded data for stock: VZ   
Loaded data for stock: WMT   
Loaded data for stock: XOM

'data.frame': 750 obs. of 16 variables:  
 $ quarter : int 1 1 1 1 1 1 1 1 1 1 ...  
 $ stock : chr "AA" "AA" "AA" "AA" ...  
 $ date : chr "2011-01-07" "2011-01-14" "2011-01-21" "2011-01-28" ...  
 $ open : num 15.8 16.7 16.2 15.9 16.2 ...  
 $ high : num 16.7 16.7 16.4 16.6 17.4 ...  
 $ low : num 15.8 15.6 15.6 15.8 16.2 ...  
 $ close : num 16.4 16 15.8 16.1 17.1 ...  
 $ volume : int 239655616 242963398 138428495 151379173 154387761 114691279 80023895 132981863 109493077 114332562 ...  
 $ percent\_change\_price : num 3.79 -4.43 -2.47 1.64 5.93 ...  
 $ percent\_change\_volume\_over\_last\_wk: num NA 1.38 -43.02 9.36 1.99 ...  
 $ previous\_weeks\_volume : num 1.60e+01 2.40e+08 2.43e+08 1.38e+08 1.51e+08 ...  
 $ next\_weeks\_open : num 4.43 16.19 15.87 16.18 17.33 ...  
 $ next\_weeks\_close : num 26 15.8 16.1 17.1 17.4 ...  
 $ percent\_change\_next\_weeks\_price : num 0.183 -2.471 1.638 5.933 0.231 ...  
 $ days\_to\_next\_dividend : int NA 19 12 5 97 90 83 76 69 62 ...  
 $ percent\_return\_next\_dividend : num NA 0.188 0.19 0.186 0.175 ...

'data.frame': 25 obs. of 7 variables:  
 $ stock : chr "GE" "GE" "GE" "GE" ...  
 $ date : chr "2011-01-07" "2011-01-14" "2011-01-21" "2011-01-28" ...  
 $ open : num 18.5 18.6 19 19.9 20.1 ...  
 $ high : num 18.7 18.8 20 20.7 21 ...  
 $ low : num 18.1 18.5 18.2 19.7 20.1 ...  
 $ close : num 18.4 18.8 19.7 20.2 20.6 ...  
 $ volume: int 280146510 215626229 449508777 437317986 246591055 252254376 158691572 268500170 261175412 303566341 ...

'data.frame': 750 obs. of 16 variables:  
 $ quarter : int 1 1 1 1 1 1 1 1 1 1 ...  
 $ stock : chr "AA" "AA" "AA" "AA" ...  
 $ date : chr "2011-01-07" "2011-01-14" "2011-01-21" "2011-01-28" ...  
 $ open : num 15.8 16.7 16.2 15.9 16.2 ...  
 $ high : num 16.7 16.7 16.4 16.6 17.4 ...  
 $ low : num 15.8 15.6 15.6 15.8 16.2 ...  
 $ close : num 16.4 16 15.8 16.1 17.1 ...  
 $ volume : int 239655616 242963398 138428495 151379173 154387761 114691279 80023895 132981863 109493077 114332562 ...  
 $ percent\_change\_price : num 3.79 -4.43 -2.47 1.64 5.93 ...  
 $ percent\_change\_volume\_over\_last\_wk: num NA 1.38 -43.02 9.36 1.99 ...  
 $ previous\_weeks\_volume : num 1.60e+01 2.40e+08 2.43e+08 1.38e+08 1.51e+08 ...  
 $ next\_weeks\_open : num 4.43 16.19 15.87 16.18 17.33 ...  
 $ next\_weeks\_close : num 26 15.8 16.1 17.1 17.4 ...  
 $ percent\_change\_next\_weeks\_price : num 0.183 -2.471 1.638 5.933 0.231 ...  
 $ days\_to\_next\_dividend : int NA 19 12 5 97 90 83 76 69 62 ...  
 $ percent\_return\_next\_dividend : num NA 0.188 0.19 0.186 0.175 ...

'data.frame': 750 obs. of 6 variables:  
 $ date : Date, format: "2011-01-07" "2011-01-14" ...  
 $ open : num 15.8 16.7 16.2 15.9 16.2 ...  
 $ high : num 16.7 16.7 16.4 16.6 17.4 ...  
 $ low : num 15.8 15.6 15.6 15.8 16.2 ...  
 $ close : num 16.4 16 15.8 16.1 17.1 ...  
 $ volume: num 2.40e+08 2.43e+08 1.38e+08 1.51e+08 1.54e+08 ...

library(dplyr)  
  
  
  
# Ensure columns are properly formatted  
clean\_stock <- clean\_stock %>%  
 mutate(  
 date = as.Date(date, format = "%m/%d/%Y"),  
 open = as.numeric(gsub("\\$", "", open)),  
 high = as.numeric(gsub("\\$", "", high)),  
 low = as.numeric(gsub("\\$", "", low)),  
 close = as.numeric(gsub("\\$", "", close)),  
 volume = as.numeric(volume)  
 )  
  
# Check for issues in data conversion  
print(summary(data)) # Provides an overview of the data

quarter stock date open   
 Min. :1.00 Length:750 Length:750 Length:750   
 1st Qu.:1.00 Class :character Class :character Class :character   
 Median :2.00 Mode :character Mode :character Mode :character   
 Mean :1.52   
 3rd Qu.:2.00   
 Max. :2.00   
   
 high low close volume   
 Length:750 Length:750 Length:750 Min. :9.719e+06   
 Class :character Class :character Class :character 1st Qu.:3.087e+07   
 Mode :character Mode :character Mode :character Median :5.306e+07   
 Mean :1.175e+08   
 3rd Qu.:1.327e+08   
 Max. :1.453e+09   
   
 percent\_change\_price percent\_change\_volume\_over\_last\_wk previous\_weeks\_volume  
 Min. :-15.42290 Length:750 Length:750   
 1st Qu.: -1.28805 Class :character Class :character   
 Median : 0.00000 Mode :character Mode :character   
 Mean : 0.05026   
 3rd Qu.: 1.65089   
 Max. : 9.88223   
   
 next\_weeks\_open next\_weeks\_close percent\_change\_next\_weeks\_price  
 Length:750 Length:750 Min. :-15.4229   
 Class :character Class :character 1st Qu.: -1.2018   
 Mode :character Mode :character Median : 0.1353   
 Mean : 0.2135   
 3rd Qu.: 1.6936   
 Max. : 9.8822   
   
 days\_to\_next\_dividend percent\_return\_next\_dividend  
 Min. : 0.00 Min. :0.06557   
 1st Qu.: 24.00 1st Qu.:0.53409   
 Median : 47.00 Median :0.68014   
 Mean : 52.26 Mean :0.69156   
 3rd Qu.: 69.25 3rd Qu.:0.85402   
 Max. :329.00 Max. :1.56421   
 NA's :30 NA's :30

# Group by 'date' and calculate averages, handling NAs  
averaged\_data <- clean\_stock %>%  
 group\_by(date) %>%  
 summarize(  
 avg\_open = mean(open, na.rm = TRUE),  
 avg\_high = mean(high, na.rm = TRUE),  
 avg\_low = mean(low, na.rm = TRUE),  
 avg\_close = mean(close, na.rm = TRUE),  
 avg\_volume = mean(volume, na.rm = TRUE)  
 )  
  
# View the cleaned data  
print(head(averaged\_data))

# A tibble: 6 × 6  
 date avg\_open avg\_high avg\_low avg\_close avg\_volume  
 <date> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 2011-01-07 51.3 52.4 50.5 51.4 164199230.  
2 2011-01-14 51.2 52.3 50.6 51.9 109024554.  
3 2011-01-21 51.9 52.9 51.2 52.3 122358483.  
4 2011-01-28 52.2 53.7 51.4 52.1 150735323.  
5 2011-02-04 52.2 53.6 51.7 53.3 119958491.  
6 2011-02-11 53.3 54.7 52.8 54.1 137143768.

Removed 'stock' column for: AA   
Removed 'stock' column for: AXP   
Removed 'stock' column for: BA   
Removed 'stock' column for: BAC   
Removed 'stock' column for: CAT   
Removed 'stock' column for: CSCO   
Removed 'stock' column for: CVX   
Removed 'stock' column for: DD   
Removed 'stock' column for: DIS   
Removed 'stock' column for: GE   
Removed 'stock' column for: HD   
Removed 'stock' column for: HPQ   
Removed 'stock' column for: IBM   
Removed 'stock' column for: INTC   
Removed 'stock' column for: JNJ   
Removed 'stock' column for: JPM   
Removed 'stock' column for: KRFT   
Removed 'stock' column for: KO   
Removed 'stock' column for: MCD   
Removed 'stock' column for: MMM   
Removed 'stock' column for: MRK   
Removed 'stock' column for: MSFT   
Removed 'stock' column for: PFE   
Removed 'stock' column for: PG   
Removed 'stock' column for: T   
Removed 'stock' column for: TRV   
Removed 'stock' column for: UTX   
Removed 'stock' column for: VZ   
Removed 'stock' column for: WMT   
Removed 'stock' column for: XOM

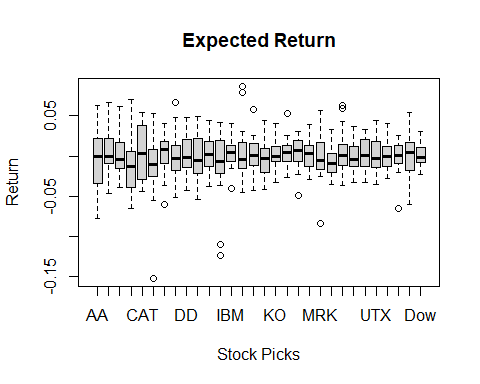
[1] "C:/Users/Leonel/Desktop/MSDA/MSDA/MS Data Analytics/Current Class/DA 6813/Week 6"

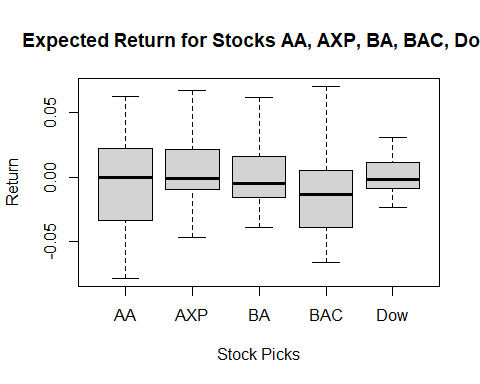
Data cleaning completed for all stocks.

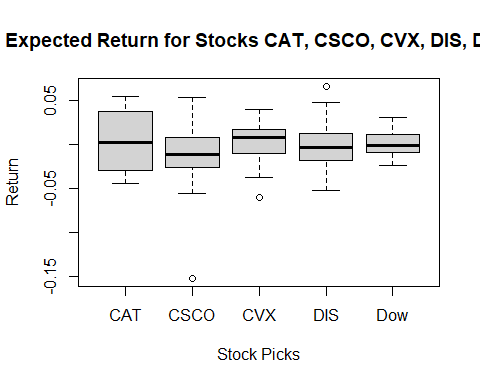
Removed the 'stock' column for all stocks.

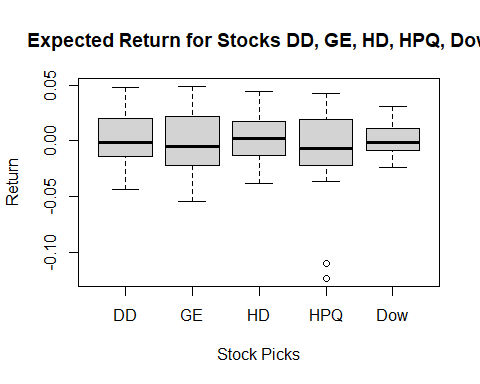
Computed returns and removed missing values for all stocks.

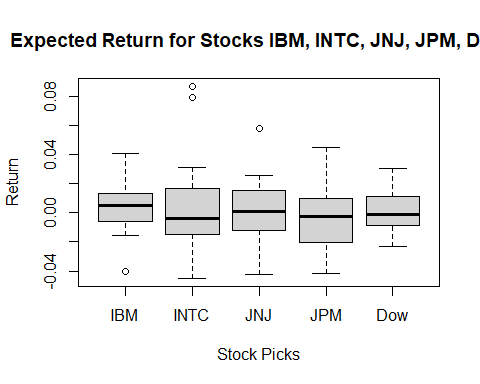
AA AXP BA BAC CAT  
[1,] -0.027405603 0.0426059513 0.009945229 0.070175439 0.002987304  
[2,] -0.011271133 -0.0054054054 0.022977023 -0.065573770 -0.013402829  
[3,] 0.021532616 -0.0465217391 -0.034179688 -0.045614035 0.031590296  
[4,] 0.062616243 -0.0009119927 0.031055901 0.050735294 0.040865385  
[5,] 0.013418903 0.0668644455 0.010647240 0.033589923 0.039662617  
[6,] -0.005181347 -0.0260962567 0.012475742 -0.001354096 0.022406799  
 CSCO CVX DIS DD GE  
[1,] 0.01144492 0.017984428 -0.004055767 0.0008038585 0.021161150  
[2,] -0.02310231 0.010233761 0.011453296 -0.0291164659 0.048884166  
[3,] 0.01013514 -0.004371934 -0.022395571 0.0401240951 0.023302938  
[4,] 0.05351171 0.040055692 0.047876448 0.0445416584 0.017821782  
[5,] -0.15192744 -0.006796416 0.066322771 0.0390253189 0.037451362  
[6,] 0.00802139 0.023535511 0.003455425 0.0256504214 0.005157056  
 HD HPQ IBM INTC JNJ  
[1,] 0.043920884 0.0257263251 0.0139931048 0.020329138 -0.000798722  
[2,] 0.017275007 0.0211891892 0.0366666667 -0.012333966 0.001758593  
[3,] 0.005204054 -0.0364175312 0.0238585209 0.030739673 -0.042291733  
[4,] 0.002724796 0.0421885300 0.0300860499 0.010251631 0.013831028  
[5,] 0.018478261 0.0255112798 -0.0009146341 0.003690037 -0.002301118  
[6,] 0.026680896 0.0006167763 0.0060421117 0.017463235 0.006754530  
 JPM KO KRFT MCD MMM  
[1,] 0.029101742 0.003337572 0.004809234 -0.004168347 0.021686188  
[2,] 0.008461367 -0.005702519 0.000319081 0.012827437 0.013507378  
[3,] -0.016559947 -0.008921459 -0.026156300 -0.023063592 -0.020719005  
[4,] 0.001122586 0.005626105 0.021618081 0.010507642 0.009720952  
[5,] 0.044404575 0.016144501 -0.016992626 0.028224173 0.039755352  
[6,] 0.030706463 0.015416077 0.008153947 -0.000131337 0.012636166  
 MRK MSFT PFE PG TRV  
[1,] -0.083534137 -0.0104895105 0.000000000 0.015968992 0.024376524  
[2,] -0.009640666 -0.0098939929 0.001090513 0.005798871 0.006772835  
[3,] -0.024483776 -0.0096359743 -0.011437908 -0.025944470 0.014727273  
[4,] -0.005443000 0.0007207207 0.063360882 -0.009190031 0.028668697  
[5,] 0.005472788 -0.0187252431 -0.024352332 0.017607294 0.027521338  
[6,] -0.006652555 -0.0069724771 0.019118428 -0.006642979 0.032717410  
 UTX VZ WMT XOM Dow  
[1,] 0.000000000 -0.013080991 0.013498521 0.0297658420 0.009633087  
[2,] 0.014162873 -0.014382403 0.016785258 0.0146454265 0.007159092  
[3,] 0.015336658 0.019456366 0.017405347 0.0001266143 -0.004054545  
[4,] 0.013385730 0.019085041 -0.011816578 0.0543106722 0.022691631  
[5,] 0.032476975 0.002203250 -0.006068178 -0.0055235351 0.014990299  
[6,] -0.002230047 0.006320418 -0.005566529 0.0202849553 0.009588999

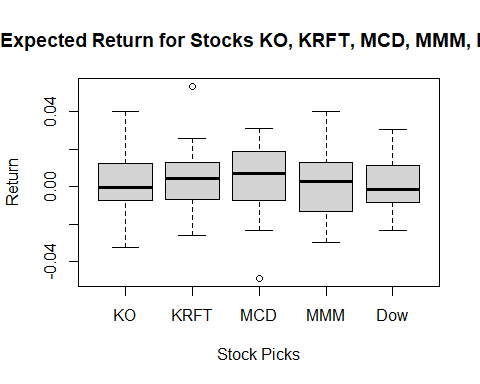


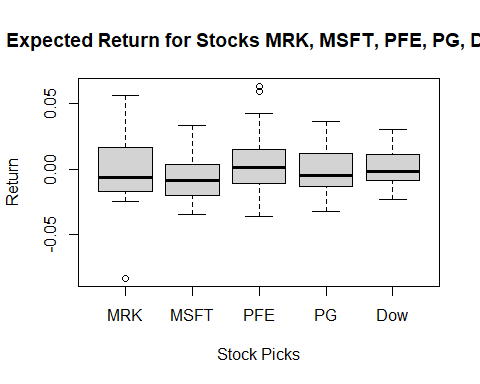


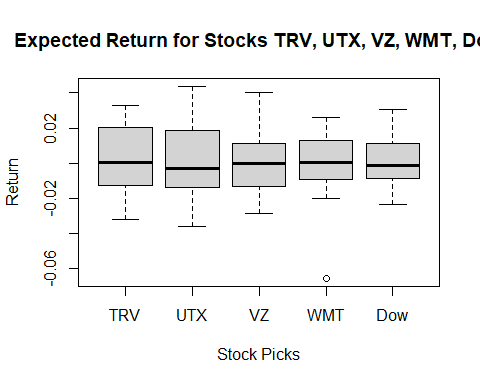


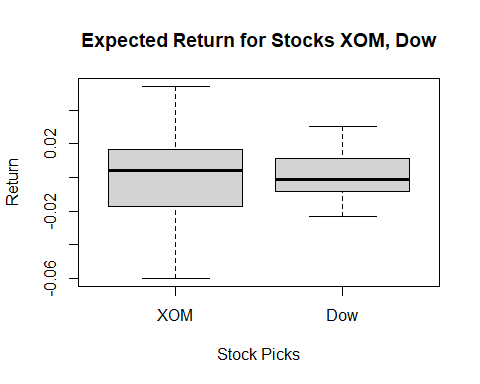












# Load the here package  
library(here)  
  
# Define the number of stocks to compare in each plot  
stocks\_per\_plot <- 4  
  
# Get the column names excluding "Dow"  
stock\_columns <- colnames(MyData)[colnames(MyData) != "Dow"]  
  
# Define the output folder using the here package  
output\_folder <- here("plots")  
  
# Create the folder if it doesn't exist  
if (!dir.exists(output\_folder)) {  
 dir.create(output\_folder, recursive = TRUE)  
}  
  
# Loop through the stocks in chunks of 4 and create boxplots  
for (i in seq(1, length(stock\_columns), by = stocks\_per\_plot)) {  
 # Subset the stocks for the current plot and include Dow  
 selected\_columns <- c(stock\_columns[i:min(i + stocks\_per\_plot - 1, length(stock\_columns))], "Dow")  
   
 # Define the filename for the current plot  
 plot\_filename <- file.path(output\_folder, paste0("Boxplot\_", i, ".png"))  
   
 # Save the boxplot as a PNG file  
 png(plot\_filename, width = 800, height = 600)  
 boxplot(  
 MyData[, selected\_columns, drop = FALSE],  
 main = paste("Expected Return for Stocks", paste(selected\_columns, collapse = ", ")),  
 xlab = "Stock Picks",  
 ylab = "Return"  
 )  
 dev.off() # Close the graphics device  
}

DataMean DataSD  
AA -0.0025562614 0.03450483  
AXP 0.0039787288 0.02867944  
BA 0.0014706601 0.02740461  
BAC -0.0120307527 0.03342655  
CAT 0.0032371593 0.03336199  
CSCO -0.0132726501 0.03920114  
CVX 0.0032523639 0.02448295  
DIS -0.0016628865 0.02746833  
DD 0.0021434643 0.02664204  
GE -0.0007380844 0.02560781  
HD 0.0011017053 0.02335630  
HPQ -0.0098428976 0.03914964  
IBM 0.0047343827 0.01806670  
INTC 0.0015586246 0.03211640  
JNJ 0.0018110711 0.02068670  
JPM -0.0039127044 0.02246357  
KO 0.0014376506 0.01628497  
KRFT 0.0044823469 0.01777007  
MCD 0.0041737383 0.01919738  
MMM 0.0023999818 0.02069039  
MRK -0.0028985966 0.02654569  
MSFT -0.0065998397 0.01858821  
PFE 0.0041023671 0.02598915  
PG -0.0011006261 0.01777888  
TRV 0.0027203616 0.01934159  
UTX 0.0028884330 0.02084871  
VZ 0.0002326181 0.01783121  
WMT -0.0011305653 0.01894121  
XOM 0.0009597025 0.02533114  
Dow 0.0010161594 0.01438800

Stock: KRFT, Beta: 0.19, R-Squared: 0.0237  
Stock: PG, Beta: 0.4855, R-Squared: 0.1544  
Stock: MRK, Beta: 0.5415, R-Squared: 0.0862  
Stock: BAC, Beta: 0.6528, R-Squared: 0.0789  
Stock: CSCO, Beta: 0.6541, R-Squared: 0.0576  
Stock: KO, Beta: 0.6811, R-Squared: 0.3622  
Stock: WMT, Beta: 0.686, R-Squared: 0.2716  
Stock: MSFT, Beta: 0.7148, R-Squared: 0.3061  
Stock: PFE, Beta: 0.7237, R-Squared: 0.1605  
Stock: MCD, Beta: 0.7446, R-Squared: 0.3114  
Stock: JNJ, Beta: 0.7601, R-Squared: 0.2795  
Stock: VZ, Beta: 0.7612, R-Squared: 0.3772  
Stock: CVX, Beta: 0.8434, R-Squared: 0.2457  
Stock: JPM, Beta: 0.8679, R-Squared: 0.309  
Stock: IBM, Beta: 0.9226, R-Squared: 0.5398  
Stock: HD, Beta: 0.9871, R-Squared: 0.3697  
Stock: TRV, Beta: 1.0005, R-Squared: 0.5539  
Stock: AXP, Beta: 1.09, R-Squared: 0.299  
Stock: DD, Beta: 1.129, R-Squared: 0.3717  
Stock: XOM, Beta: 1.1974, R-Squared: 0.4626  
Stock: UTX, Beta: 1.1999, R-Squared: 0.6857  
Stock: MMM, Beta: 1.2392, R-Squared: 0.7426  
Stock: INTC, Beta: 1.2524, R-Squared: 0.3148  
Stock: AA, Beta: 1.3034, R-Squared: 0.2954  
Stock: GE, Beta: 1.3552, R-Squared: 0.5798  
Stock: HPQ, Beta: 1.4257, R-Squared: 0.2745  
Stock: DIS, Beta: 1.4881, R-Squared: 0.6076  
Stock: CAT, Beta: 1.493, R-Squared: 0.4146  
Stock: BA, Beta: 1.6256, R-Squared: 0.7285

Attaching package: 'flextable'

The following object is masked from 'package:purrr':  
  
 compose

| Stock | Beta | R\_Squared | Dow\_Beta |
| --- | --- | --- | --- |
| KRFT | 0.1900 | 0.0237 | 1.0000 |
| PG | 0.4855 | 0.1544 | 1.0000 |
| MRK | 0.5415 | 0.0862 | 1.0000 |
| BAC | 0.6528 | 0.0789 | 1.0000 |
| CSCO | 0.6541 | 0.0576 | 1.0000 |
| KO | 0.6811 | 0.3622 | 1.0000 |
| WMT | 0.6860 | 0.2716 | 1.0000 |
| MSFT | 0.7148 | 0.3061 | 1.0000 |
| PFE | 0.7237 | 0.1605 | 1.0000 |
| MCD | 0.7446 | 0.3114 | 1.0000 |
| JNJ | 0.7601 | 0.2795 | 1.0000 |
| VZ | 0.7612 | 0.3772 | 1.0000 |
| CVX | 0.8434 | 0.2457 | 1.0000 |
| JPM | 0.8679 | 0.3090 | 1.0000 |
| IBM | 0.9226 | 0.5398 | 1.0000 |
| HD | 0.9871 | 0.3697 | 1.0000 |
| TRV | 1.0005 | 0.5539 | 1.0000 |
| AXP | 1.0900 | 0.2990 | 1.0000 |
| DD | 1.1290 | 0.3717 | 1.0000 |
| XOM | 1.1974 | 0.4626 | 1.0000 |
| UTX | 1.1999 | 0.6857 | 1.0000 |
| MMM | 1.2392 | 0.7426 | 1.0000 |
| INTC | 1.2524 | 0.3148 | 1.0000 |
| AA | 1.3034 | 0.2954 | 1.0000 |
| GE | 1.3552 | 0.5798 | 1.0000 |
| HPQ | 1.4257 | 0.2745 | 1.0000 |
| DIS | 1.4881 | 0.6076 | 1.0000 |
| CAT | 1.4930 | 0.4146 | 1.0000 |
| BA | 1.6256 | 0.7285 | 1.0000 |

# Load the required libraries  
library(flextable)  
library(here)  
library(webshot)

Warning: package 'webshot' was built under R version 4.3.3

# Install webshot dependencies if not already installed  
# webshot::install\_phantomjs()  
  
# Define the output folder using the here package  
output\_folder <- here("tables")  
  
# Create the folder if it doesn't exist  
if (!dir.exists(output\_folder)) {  
 dir.create(output\_folder, recursive = TRUE)  
}  
  
# List of stocks to process (excluding "Dow")  
stocks <- colnames(MyData)[colnames(MyData) != "Dow"]  
  
# Initialize an empty data frame to store results  
results <- data.frame(Stock = character(), Beta = numeric(), R\_Squared = numeric(), Dow\_Beta = numeric(), stringsAsFactors = FALSE)  
  
# Loop through each stock and compute the linear model  
for (stock in stocks) {  
 # Construct the formula dynamically  
 formula <- as.formula(paste(stock, "~ Dow"))  
   
 # Fit the linear model  
 lm\_model <- lm(formula, data = as.data.frame(MyData))  
   
 # Extract Beta (coefficient for Dow) and R-squared  
 beta <- summary(lm\_model)$coefficients[2, 1]  
 r\_squared <- summary(lm\_model)$r.squared  
   
 # Append results to the data frame  
 results <- rbind(results, data.frame(Stock = stock, Beta = beta, R\_Squared = r\_squared, Dow\_Beta = 1)) # Dow\_Beta is always 1  
}  
  
# Order results by Beta (safest to riskiest)  
results <- results[order(results$Beta), ]  
  
# Split results into chunks of stocks per table  
stocks\_per\_table <- 10  
num\_tables <- ceiling(nrow(results) / stocks\_per\_table)  
  
# Loop to create and save each table as an image  
for (i in seq(1, num\_tables)) {  
 # Subset the data for the current table  
 start\_row <- (i - 1) \* stocks\_per\_table + 1  
 end\_row <- min(i \* stocks\_per\_table, nrow(results))  
 table\_data <- results[start\_row:end\_row, ]  
   
 # Create the flextable for the current chunk  
 ft <- flextable(table\_data) %>%  
 set\_caption(paste("Beta vs Dow Beta (Table", i, ")")) %>%  
 bg(j = "Dow\_Beta", bg = "lightblue", part = "body") %>%  
 colformat\_double(j = c("Beta", "R\_Squared", "Dow\_Beta"), digits = 4) %>%  
 autofit()  
   
 # Define the filename for the current table  
 image\_filename <- file.path(output\_folder, paste0("Table\_", i, ".png"))  
   
 # Save the flextable as an image  
 save\_as\_image(ft, path = image\_filename)  
}

| Stock | Beta | R\_Squared | Dow\_Beta | Expected\_Return |
| --- | --- | --- | --- | --- |
| KRFT | 0.1900 | 0.0237 | 1.0000 | 0.0395 |
| PG | 0.4855 | 0.1544 | 1.0000 | 0.0543 |
| MRK | 0.5415 | 0.0862 | 1.0000 | 0.0571 |
| BAC | 0.6528 | 0.0789 | 1.0000 | 0.0626 |
| CSCO | 0.6541 | 0.0576 | 1.0000 | 0.0627 |
| KO | 0.6811 | 0.3622 | 1.0000 | 0.0641 |
| WMT | 0.6860 | 0.2716 | 1.0000 | 0.0643 |
| MSFT | 0.7148 | 0.3061 | 1.0000 | 0.0657 |
| PFE | 0.7237 | 0.1605 | 1.0000 | 0.0662 |
| MCD | 0.7446 | 0.3114 | 1.0000 | 0.0672 |
| JNJ | 0.7601 | 0.2795 | 1.0000 | 0.0680 |
| VZ | 0.7612 | 0.3772 | 1.0000 | 0.0681 |
| CVX | 0.8434 | 0.2457 | 1.0000 | 0.0722 |
| JPM | 0.8679 | 0.3090 | 1.0000 | 0.0734 |
| IBM | 0.9226 | 0.5398 | 1.0000 | 0.0761 |
| HD | 0.9871 | 0.3697 | 1.0000 | 0.0794 |
| TRV | 1.0005 | 0.5539 | 1.0000 | 0.0800 |
| AXP | 1.0900 | 0.2990 | 1.0000 | 0.0845 |
| DD | 1.1290 | 0.3717 | 1.0000 | 0.0864 |
| XOM | 1.1974 | 0.4626 | 1.0000 | 0.0899 |
| UTX | 1.1999 | 0.6857 | 1.0000 | 0.0900 |
| MMM | 1.2392 | 0.7426 | 1.0000 | 0.0920 |
| INTC | 1.2524 | 0.3148 | 1.0000 | 0.0926 |
| AA | 1.3034 | 0.2954 | 1.0000 | 0.0952 |
| GE | 1.3552 | 0.5798 | 1.0000 | 0.0978 |
| HPQ | 1.4257 | 0.2745 | 1.0000 | 0.1013 |
| DIS | 1.4881 | 0.6076 | 1.0000 | 0.1044 |
| CAT | 1.4930 | 0.4146 | 1.0000 | 0.1047 |
| BA | 1.6256 | 0.7285 | 1.0000 | 0.1113 |

CAPM calculations and table generation completed successfully.

`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'  
`geom\_smooth()` using formula = 'y ~ x'

# Load required libraries  
library(flextable)  
library(here)  
library(webshot)  
  
# Install webshot dependencies if not already installed  
# webshot::install\_phantomjs()  
  
# Inputs for CAPM  
risk\_free\_rate <- 0.03 # Example: 3% annualized (e.g., 10-year Treasury bond yield)  
market\_return <- 0.08 # Example: 8% annualized expected return of Dow Jones index  
market\_risk\_premium <- market\_return - risk\_free\_rate  
  
# Assuming `results` contains Beta values calculated earlier  
# Add Expected Return based on the CAPM formula  
results$Expected\_Return <- risk\_free\_rate + (results$Beta \* market\_risk\_premium)  
  
# Order by Expected Return (safest to riskiest stock based on CAPM)  
results <- results[order(results$Expected\_Return), ]  
  
# Create a flextable for results with proper styling  
ft <- flextable(results) %>%  
 set\_caption("CAPM: Expected Returns vs Dow Beta") %>%  
 bg(bg = "white", part = "all") %>% # Set white background for the entire table  
 color(color = "black", part = "all") %>% # Set text color to black  
 border\_outer() %>%  
 colformat\_double(j = c("Beta", "R\_Squared", "Dow\_Beta", "Expected\_Return"), digits = 4) %>%  
 autofit()  
  
# Define the output folder for the image  
output\_folder <- here("Week 6")  
if (!dir.exists(output\_folder)) {  
 dir.create(output\_folder, recursive = TRUE)  
}  
  
# Define the image path  
image\_path <- file.path(output\_folder, "CAPM\_Results\_Table.png")  
  
# Save the flextable as an image with proper rendering  
save\_as\_image(ft, path = image\_path)

[1] "C:/Users/Leonel/Desktop/MSDA/MSDA/MS Data Analytics/Current Class/DA 6813/Week 6/Week 6/CAPM\_Results\_Table.png"

# Print a message to indicate successful saving

