R Notebook

BIS581 This assignment will look at some stock market data. Answer each question, provide your r code also. You will need to edit the line below to point it at where you saved the CSV file

chooseCRANmirror(graphics = FALSE, ind = 1)

install.packages("dplyr", repos = "https://cran.rstudio.com/")

## Installing package into 'C:/Users/demet/AppData/Local/R/win-library/4.3'  
## (as 'lib' is unspecified)

## package 'dplyr' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'dplyr'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying  
## C:\Users\demet\AppData\Local\R\win-library\4.3\00LOCK\dplyr\libs\x64\dplyr.dll  
## to C:\Users\demet\AppData\Local\R\win-library\4.3\dplyr\libs\x64\dplyr.dll:  
## Permission denied

## Warning: restored 'dplyr'

##   
## The downloaded binary packages are in  
## C:\Users\demet\AppData\Local\Temp\Rtmp82Ckjo\downloaded\_packages

library(dplyr)

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

##   
## 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

my\_stocks <- read.csv("stocks.csv",header=TRUE, stringsAsFactors = FALSE)

Your job first, is to ‘know your data’. Run whatever code you feel necessary to get to know your data. Add more code blocks as needed, leave this part in your document when you submit it.

For the questions below, show both the code used to answer along with text that shows the answer:

#How many different stocks are there (by stock symbol) (5)?

num\_stocks <- my\_stocks %>% distinct(stock\_symbol) %>% nrow()  
print(num\_stocks)

## [1] 25

#I used the distinct funtion to try to find all the unique symbols in the dataset. From there, I counted nrow() to get the number of unique stocks.

#What are those stocks (5)?

stock\_symbol <- my\_stocks %>% distinct(stock\_symbol) %>% pull(stock\_symbol)  
print(stock\_symbol)

## [1] "XL" "XFJ" "XCO" "XVG" "XFB" "XOM" "XKN" "XVF" "XTO" "XJT" "XAA" "XFP"  
## [13] "XKO" "XKK" "XEC" "XRX" "XFL" "XIN" "XFH" "XKE" "XCJ" "XRM" "XEL" "XFD"  
## [25] "XFR"

#I used the distinct function again. I extracted all unique stock symbols from the data set. From there, it displayed a list of stock symbols from me using the pull() function.

#Which stock had the highest single gain in a day, which day was it (10)?

my\_stocks <- my\_stocks %>%  
 arrange(stock\_symbol, date) %>%  
 group\_by(stock\_symbol) %>%  
 mutate(daily\_change = stock\_price\_high - stock\_price\_open) %>%  
 ungroup()  
  
max\_gain <- my\_stocks %>% filter(daily\_change == max(daily\_change, na.rm = TRUE))  
print(max\_gain)

## # A tibble: 1 × 10  
## exchange stock\_symbol date stock\_price\_open stock\_price\_high stock\_price\_low  
## <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 NYSE XFJ 2008-… 11.4 25 11.4  
## # ℹ 4 more variables: stock\_price\_close <dbl>, stock\_volume <int>,  
## # stock\_price\_adj\_close <dbl>, daily\_change <dbl>

#Calculating the daily change in stock price by subtracting stock\_price\_open from the stock\_price\_high.This would be for each stock, then filtering for the highest daily change which then would display the stock symbol and date of the highest gain.

#Which stock had the highest single loss in a day, which day was it (10)?

max\_loss <- my\_stocks %>% filter(daily\_change == min(daily\_change, na.rm = TRUE))  
print(max\_loss)

## # A tibble: 18,079 × 10  
## exchange stock\_symbol date stock\_price\_open stock\_price\_high stock\_price\_low  
## <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 NYSE XAA 1995… 12.1 12.1 12.1  
## 2 NYSE XAA 1995… 12 12 11.9  
## 3 NYSE XAA 1995… 12 12 11.9  
## 4 NYSE XAA 1995… 11.9 11.9 11.9  
## 5 NYSE XAA 1995… 12 12 11.8  
## 6 NYSE XAA 1995… 11.6 11.6 11.5  
## 7 NYSE XAA 1995… 11.6 11.6 11.5  
## 8 NYSE XAA 1995… 11.8 11.8 11.6  
## 9 NYSE XAA 1995… 11.8 11.8 11.8  
## 10 NYSE XAA 1995… 11.9 11.9 11.9  
## # ℹ 18,069 more rows  
## # ℹ 4 more variables: stock\_price\_close <dbl>, stock\_volume <int>,  
## # stock\_price\_adj\_close <dbl>, daily\_change <dbl>

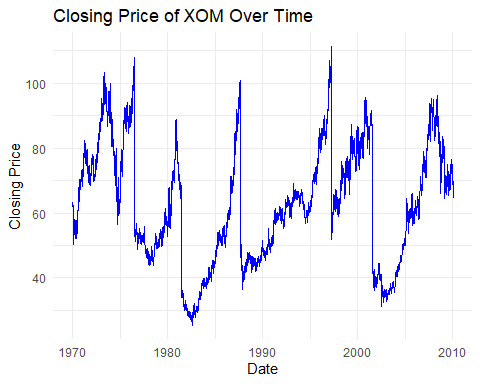
#I followed the same process from before when looking at gains but instead, use the filter to find the minimum daily change. The would represent any of the highest loss.

#Create a plot of closing price for XOM over time (5)

library(ggplot2)

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

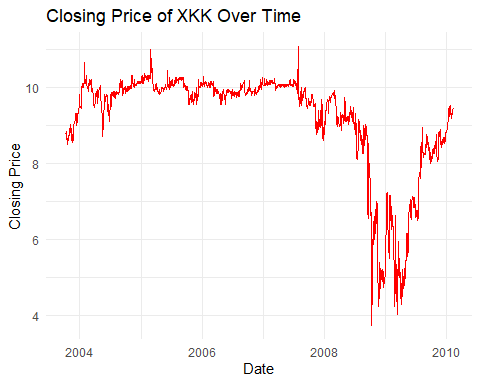
ggplot(my\_stocks %>% filter(stock\_symbol == "XOM"),   
 aes(x = as.Date(date), y = stock\_price\_high)) +  
 geom\_line(color = "blue") +  
 labs(title = "Closing Price of XOM Over Time", x = "Date", y = "Closing Price") +  
 theme\_minimal()



#I used the ggplt filter for XOM stock and plotted the stock\_high\_price vs the date column.

#Create a plot of closing price for XKK over time (5)

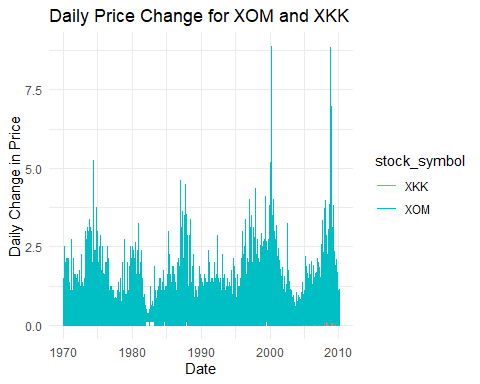
ggplot(my\_stocks %>% filter(stock\_symbol == "XKK"),   
 aes(x = as.Date(date), y = stock\_price\_high)) +  
 geom\_line(color = "red") +  
 labs(title = "Closing Price of XKK Over Time", x = "Date", y = "Closing Price") +  
 theme\_minimal()



#This is almost identical to my previous answer. Filter the data set XKK stock and plot the stock\_price\_high.

#Create a plot of the change in price per day for both XOM and XKK (10)

ggplot(my\_stocks %>% filter(stock\_symbol %in% c("XOM", "XKK")),   
 aes(x = as.Date(date), y = daily\_change, color = stock\_symbol)) +  
 geom\_line() +  
 labs(title = "Daily Price Change for XOM and XKK", x = "Date", y = "Daily Change in Price") +  
 theme\_minimal()



#First you would calculate the daily change in price. Then, use the ggplot to plot the daily price change. You would do this for both, XOM and XKK over time.