Data Project - Stock Exchange Data

Samuel Kihuguru

12/10/2021

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
==============================================
```

Stock Index Project

This is R Markdown data project analyzes stock index data across 11 exchanges. The data files being used for this report are indexData.csv, indexProcessed.csv, and indexInfo.csv. These comma-separated values files were imported from Kaggle, a community interface of data scientists and machine learning practitioners. For more details on accessing the Kaggle data, click the link here.

```
indexD <- read_csv("indexData.csv")
```

```
## Rows: 112457 Columns: 8
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): Index, Open, High, Low, Close, Adj Close, Volume
## date (1): Date
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
indexI <- read_csv("indexInfo.csv")
```

```
## Rows: 14 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): Region, Exchange, Index, Currency
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
indexP <- read_csv("indexProcessed.csv")
```

```
## Rows: 104224 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): Index
## dbl  (7): Open, High, Low, Close, Adj Close, Volume, CloseUSD
## date (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Before undertaking the data analysis process, the datasets of interest must be cleaned and made consistent. There are three datasets we are working on: indexD, indexI, and indexP. Each of these tables has variables that require changes to their data type. For example, the Index column in indexD which shows the ticker symbols for each of the exchanges would need to move from a character("chr") variable to a factor variable.

```
# Index in indexD needs to be a Factor
indexD$Index <- factor(indexD$Index)

# Open, High, Low, Close, Adj Close, Volume need to be numeric
indexD$Open <- as.numeric(indexD$Open)
indexD$High <- as.numeric(indexD$High)
indexD$Low <- as.numeric(indexD$Low)
indexD$Close <- as.numeric(indexD$Close)
indexD$`Adj Close` <- as.numeric(indexD$`Adj Close`)
indexD$Volume <- as.numeric(indexD$Volume)

# Data cleaning for indexInfo
indexI$Region <- factor(indexI$Region)
indexI$Exchange <- factor(indexI$Exchange)
indexI$Index <- factor(indexI$Index)
indexI$Currency <- factor(indexI$Currency)

# Data cleaning for indexProcessed
indexP$Index <- factor(indexP$Index)
```

Throughout the report, you will encounter Index symbols associated with the Exchanges. The following are the ticker symbols with their exchange name:

- NYA — New York Stock Exchange
- IXIC — NASDAQ
- HSI — Hong Kong Stock Exchange
- 000001.SS — Shanghai Stock Exchange
- N225 — Tokyo Stock Exchange
- N100 — Euronext
- 399001.SZ — Shenzhen Stock Exchange
- GSPSTE — Toronto Stock Exchange

- NSEI — National Stock Exchange of India
- GDAX1 — Frankfurt Stock Exchange
- KS11 — Korea Exchange
- SSMI — SIX Swiss Exchange
- TWII — Taiwan Stock Exchange
- J203.JO — Johannesburg Stock Exchange

```
indexI %>%
  select(Index,Exchange,Region)
```

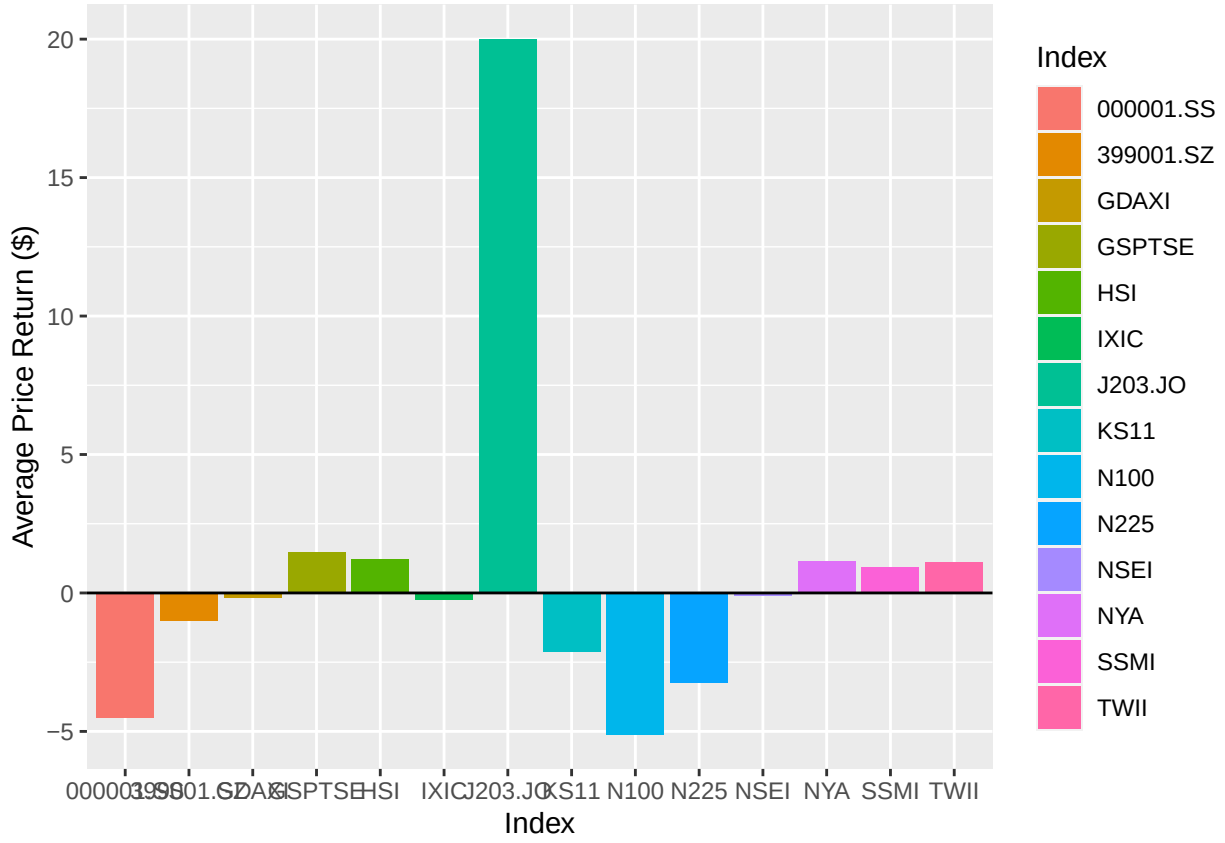
```
## # A tibble: 14 x 3
##    Index     Exchange                          Region
##    <fct>     <fct>                             <fct>
##  1 NYA       New York Stock Exchange           United States
##  2 IXIC      NASDAQ                            United States
##  3 HSI       Hong Kong Stock Exchange          Hong Kong
##  4 000001.SS Shanghai Stock Exchange           China
##  5 N225      Tokyo Stock Exchange              Japan
##  6 N100      Euronext                          Europe
##  7 399001.SZ Shenzhen Stock Exchange           China
##  8 GSPTSE    Toronto Stock Exchange            Canada
##  9 NSEI      National Stock Exchange of India  India
## 10 GDAXI     Frankfurt Stock Exchange          Germany
## 11 KS11      Korea Exchange                    Korea
## 12 SSMI      SIX Swiss Exchange                Switzerland
## 13 TWII      Taiwan Stock Exchange             Taiwan
## 14 J203.JO   Johannesburg Stock Exchange       South Africa
```

Including Plots

Q1

What is the average price return by Index? What is the Daily Return and Daily Return (%) for each index price?

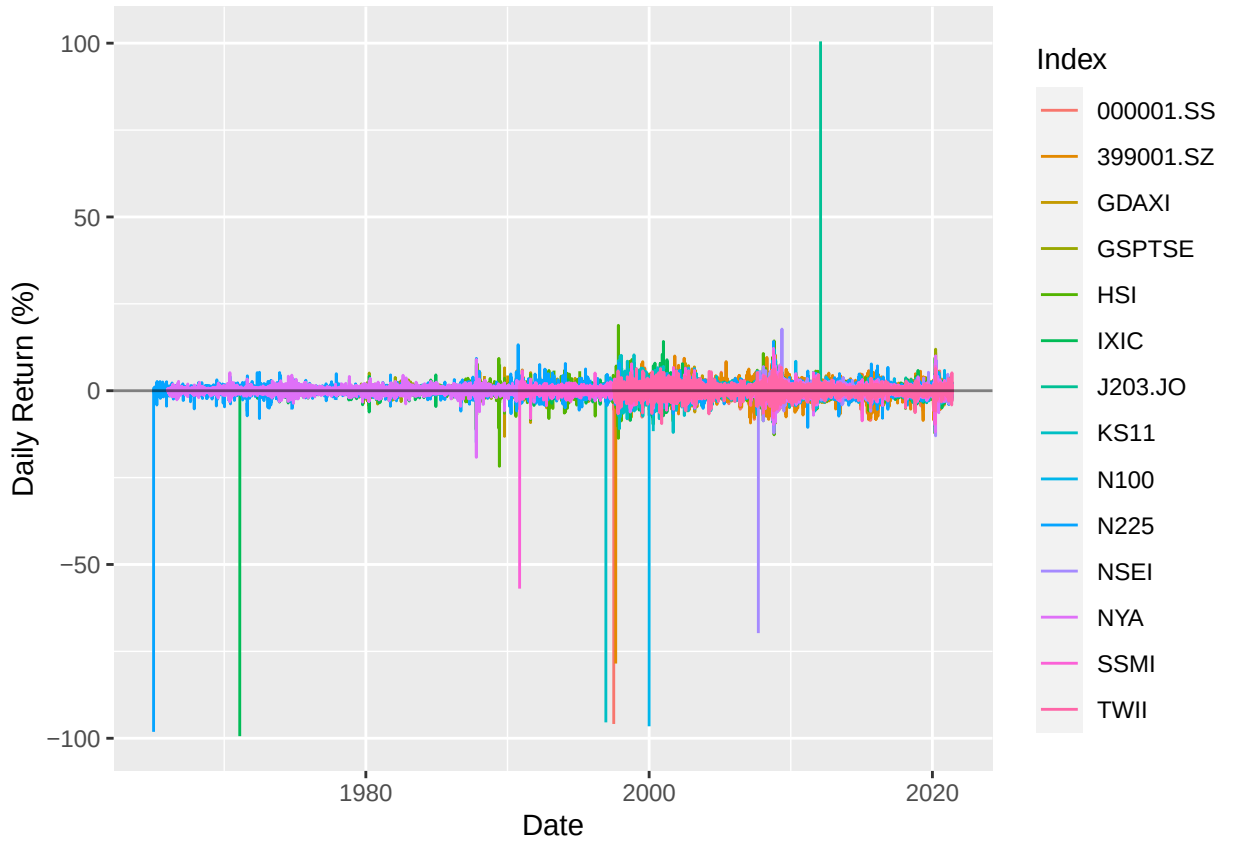
```
indexD %>%
  mutate(Daily_Return = `Adj Close`-lag(`Adj Close`)) %>%
  mutate(Daily_Return_Perc. = (`Adj Close`-lag(`Adj Close`))*100/lag(`Adj Close`)) %>%
  select(Index, Date, Open, High, Low, Close, Daily_Return,Daily_Return_Perc.) %>%
  group_by(Index) %>%
  summarize(Avg_Return = mean(Daily_Return, na.rm = TRUE), Avg_Return_Perc. =   mean(Daily_Return_Perc.
  ggplot(aes(Index, Avg_Return, fill = Index))+
  geom_bar(stat = "identity")+
  geom_hline(yintercept = 0)+
  ylab("Average Price Return ($)")
```



Q2

Let's take a look at the exchange returns across time. What is the Daily Return(%) over time by Index? Are there noticeable anomalies?

```
# Daily Return(%) against time (color by index)
indexD %>%
  mutate(Daily_Return = `Adj Close`-lag(`Adj Close`)) %>%
  mutate(Daily_Return_Perc. = ((`Adj Close`-lag(`Adj Close`))*100)/lag(`Adj Close`)) %>%
  ggplot(aes(Date, Daily_Return_Perc.)) +
  geom_line(aes(color = Index))+
  geom_hline(aes(yintercept = 0), color = "black", alpha = 0.5)+
  ylab("Daily Return (%)")
```



Using Daily Return and Daily Return Percentage is very useful in standardizing the data and allow us to work with mean and standard deviation data in later plots.

```
# Incorporating Daily_Return and Daily_Return_Perc. into dataset
indexD$Daily_Return <- indexD$`Adj Close`-lag(indexD$`Adj Close`)
indexD$Daily_Return_Perc. <- (indexD$`Adj Close`-lag(indexD$`Adj Close`))*100/lag(indexD$`Adj Close`)
```

Left joining indexD data with indexI that contains the Exchange name and countries associated with each Index. New index would be called "fullindex" and extracts "Year" from the Date column.

```
# Left join indexI for countries by Index
fullindex <- indexI %>%
  left_join(indexD)
```

```
## Joining, by = "Index"
```

```
fullindex$Year <- as.POSIXlt(fullindex$Date)$year + 1900
```

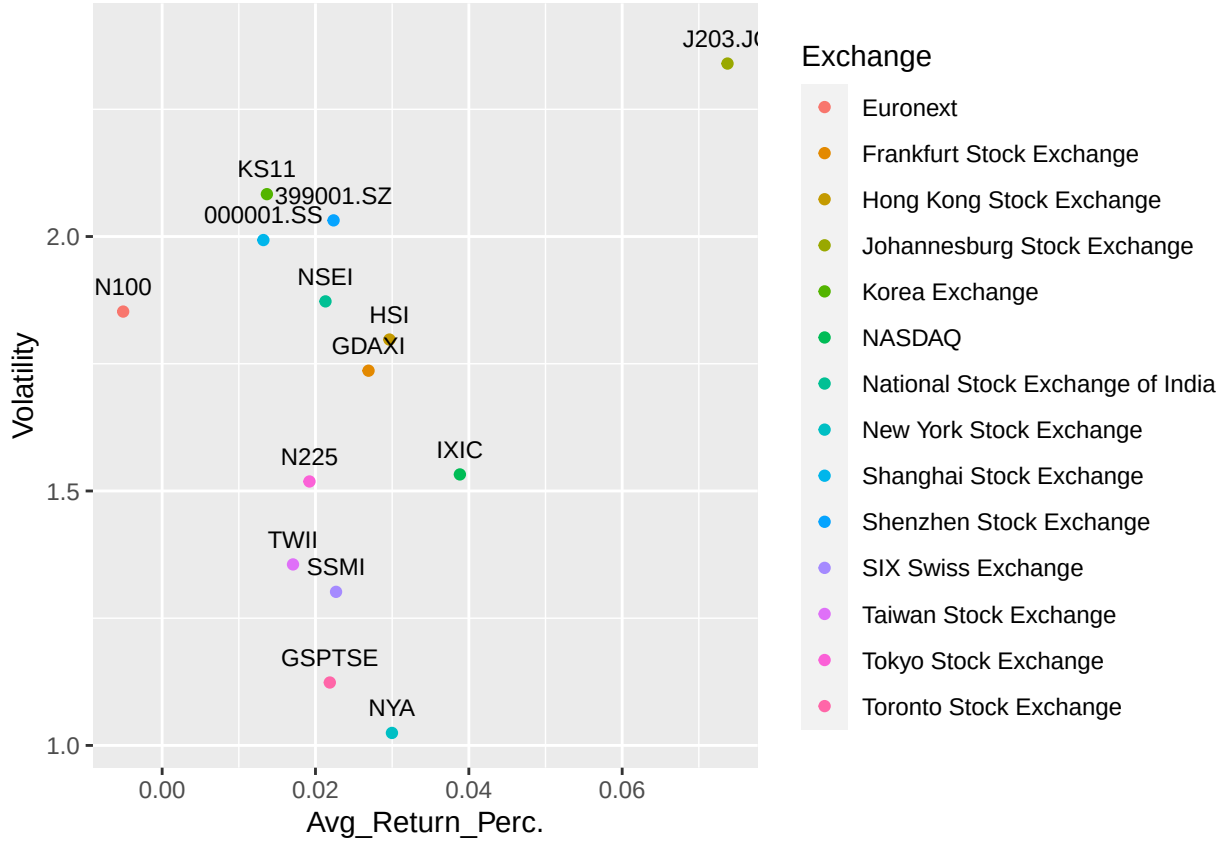
Q3

Stock exchanges play into the risk-reward relationship. Ideally, the more risk you take on, the higher the reward to compensate for higher risk. Is there a correlation in the data between average return and volatility (standard deviation)? Do some indexes experience higher reward for lower risk?

```
fullindex%>%
  group_by(Region,Index,Exchange) %>%
  summarize(Count = n(), Avg_Return = mean(Daily_Return, na.rm = TRUE),
            Avg_Return_Perc. = mean(Daily_Return_Perc., na.rm = TRUE),
            Volatility = sd(Daily_Return_Perc., na.rm = TRUE),  #,
```

```
            Duration = year(today()) - min(as.POSIXlt(Date)$year + 1900))%>%
  arrange(desc(Avg_Return_Perc.)) %>%
  ggplot(aes(Avg_Return_Perc., Volatility))+
  geom_point(aes(color = Exchange))+
  geom_text(aes(label = paste0(Index)), nudge_y = 0.05, size=3)
```

```
## `summarise()` has grouped output by 'Region', 'Index'. You can override using the `.groups` argument
```



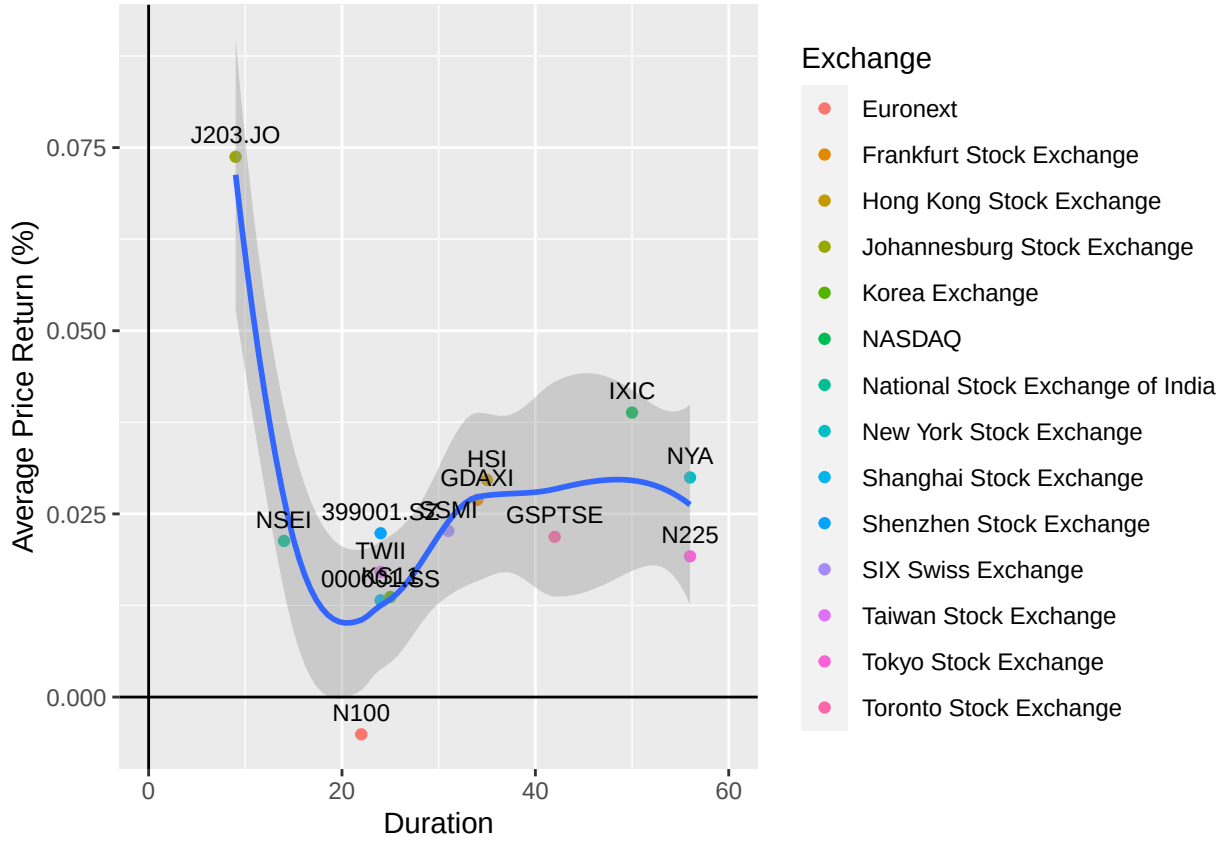
It appears that there is no clear direct correlation with the exchange data provided between average return percentage and standard deviation. As a result, you have indexes like the NYA (New York Stock Exchange) that generates the same mean price return for lower volatility than GSPTSE, SSMI, TWII, etc.

Q4

It is noticeable that some exchanges have been in place longer than others, and this seems to affect the return results. The Johannesburg Stock Exchange, for example, tends to show strong volatility and average return compared to large long-running exchanges. Is there an inverse relationship between index duration and avg_return_perc.?

```
fullindex %>%
  group_by(Region,Exchange,Index) %>%
  summarize(Count = n(), Avg_Return = mean(Daily_Return, na.rm = TRUE),
            Avg_Return_Perc. = mean(Daily_Return_Perc., na.rm = TRUE), #,
            Duration = year(today()) - min(as.POSIXlt(Date)$year + 1900)) %>%
  ggplot(aes(Duration,Avg_Return_Perc.))+
  geom_point(aes(color = Exchange))+
  geom_smooth()+
  geom_text(aes(label = paste0(Index)),nudge_y=0.003, size=3)+
```

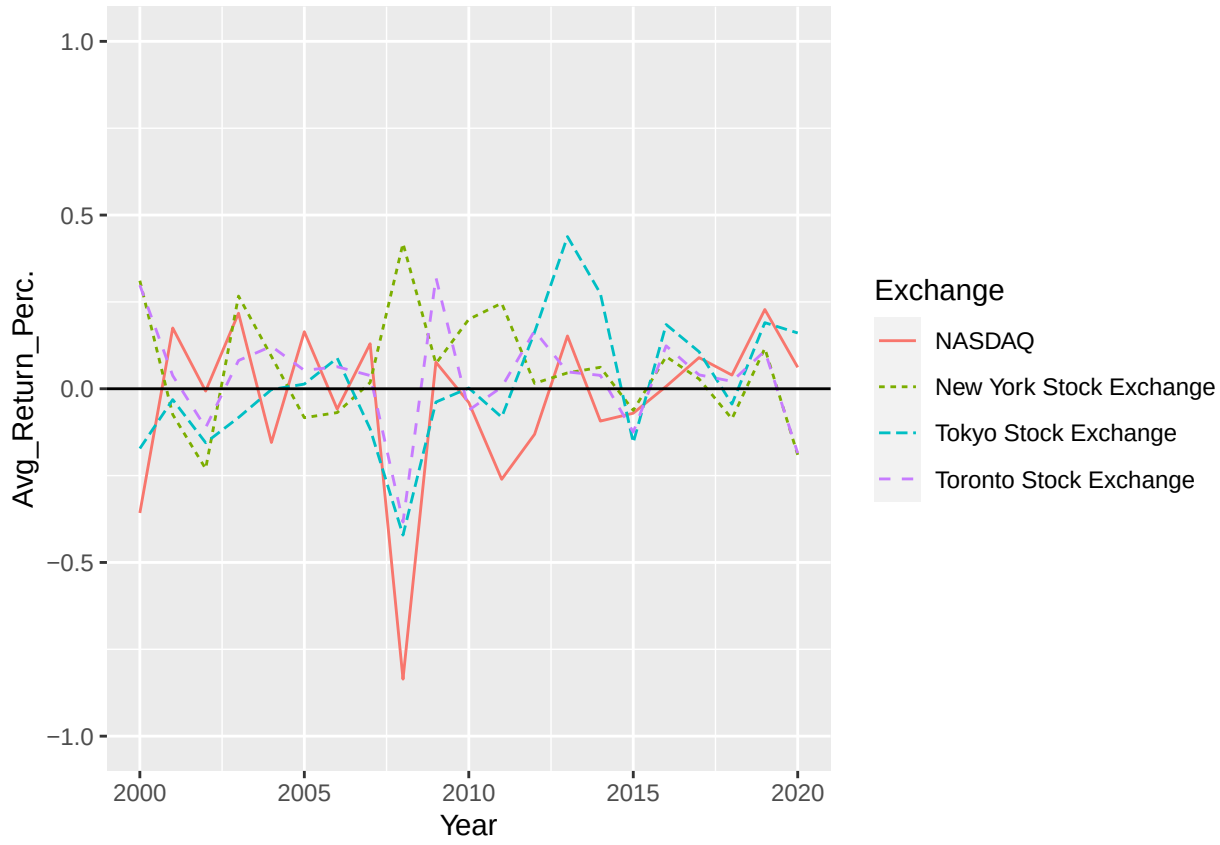
```
  geom_hline(yintercept=0)+
  geom_vline(xintercept=0)+
  coord_cartesian(xlim=c(0,60))+
  ylab("Average Price Return (%)")
```



Q5

How did the four largest indices(NYA, IXIC, GSPTSE, N225) perform annually from 2000 to 2021?

```
fullindex %>%
  filter(Index == c("NYA","IXIC","GSPTSE","N225")) %>%
  group_by(Index, Exchange, Year) %>%
  summarize(Avg_Return_Perc. = mean(Daily_Return_Perc., na.rm = TRUE)) %>%
  ggplot(aes(Year, Avg_Return_Perc.))+
  geom_line(aes(color = Exchange, linetype = Exchange))+
  geom_hline(yintercept=0)+
  xlim(2000,2020)+
  ylim(-1,1)
```



Q6

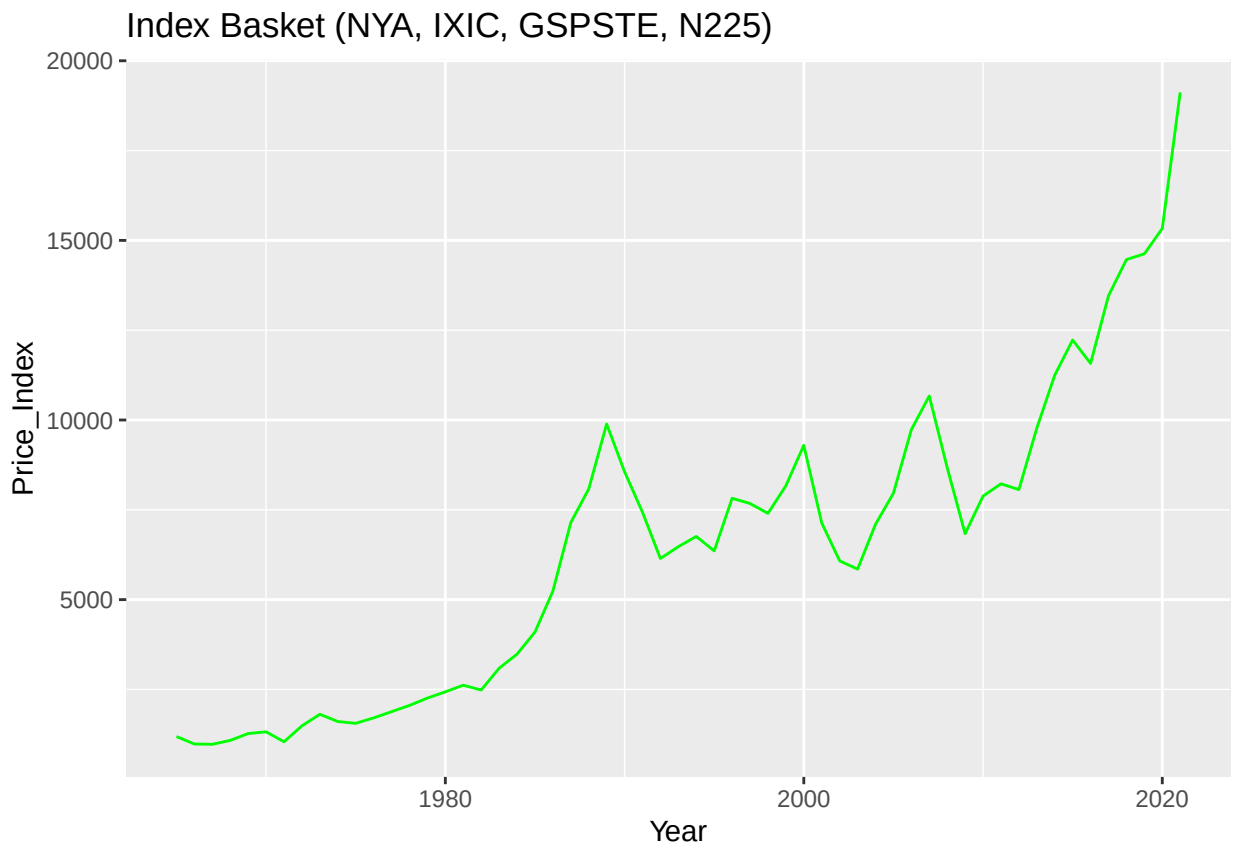
Create a new index composed of the four chosen indices and track its projection over time.

```
new_index_large <- fullindex %>%
  filter(Index == c("NYA","IXIC","GSPTSE","N225")) %>%
  group_by(Year) %>%
  summarize(Price_Index = mean(`Adj Close`, na.rm = TRUE)) %>%
  ggplot()+
  geom_line(aes(Year, Price_Index), color="green")+
  labs(title = "Index Basket (NYA, IXIC, GSPSTE, N225)")
```

```
## Warning in `==.default`(Index, c("NYA", "IXIC", "GSPTSE", "N225")): longer
## object length is not a multiple of shorter object length
```

```
## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length
```

```
new_index_large
```



Q7

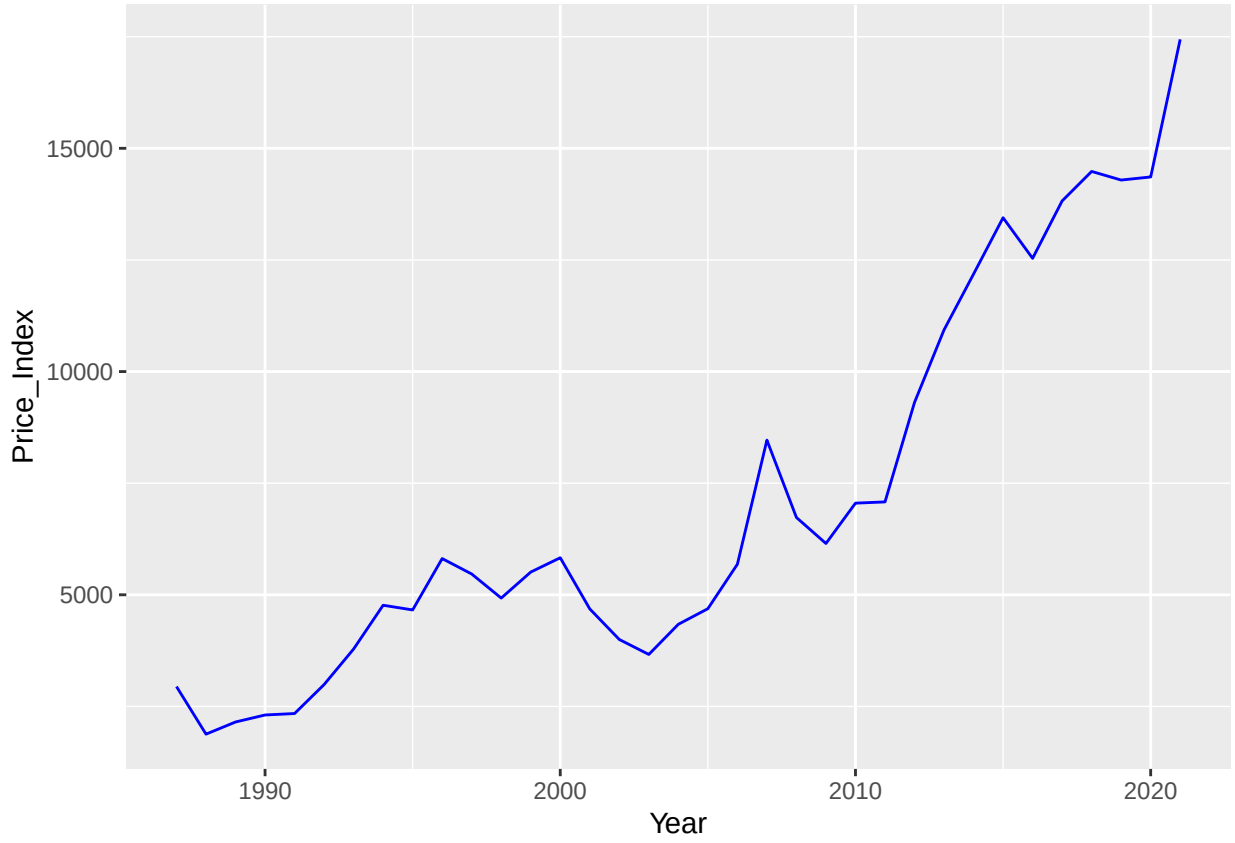
Compare the performance of the other 10 stocks from 2000 to 2010 under the new index "new_index_small".

```
new_index_small <- fullindex %>%
  filter(Index == c("HSI","000001.SS","N100","399001.SZ","NSEI","GDAXI","KS11","SSMI","TWII","J203.JO"))
  group_by(Year) %>%
  summarize(Price_Index = mean(`Adj Close`, na.rm = TRUE)) %>%
  ggplot()+
  geom_line(aes(Year, Price_Index),color="blue")
```

```
## Warning in `==.default`(Index, c("HSI", "000001.SS", "N100", "399001.SZ", :
## longer object length is not a multiple of shorter object length
```

```
## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length
```

```
new_index_small
```



Q8

```
#Compare the variability of the four chosen stocks from 2000 to 2010. The lowest index
#price is counted as 0 and the highest is a 1.
scale_level <- function(index){
  (index - min(index,na.rm=TRUE))/(max(index,na.rm=TRUE) - min(index,na.rm=TRUE))
}

#Spread the index column
four_indices <- fullindex %>%
  filter(Index == c("NYA","IXIC","GSPTSE","N225"), Year<=2021, Year>=2000) %>%
  select(Date, Index, Daily_Return_Perc.) %>%
  spread(Index,Daily_Return_Perc.) %>%
  mutate(var_GSPTSE = scale_level(GSPTSE),
         var_IXIC = scale_level(IXIC),
         var_N225 = scale_level(N225),
         var_NYA = scale_level(NYA)) %>%
  arrange(desc(Date))

four_indices
```

```
## # A tibble: 4,047 x 9
##    Date       GSPTSE    IXIC    N225    NYA var_GSPTSE var_IXIC var_N225 var_NYA
##    <date>      <dbl>   <dbl>   <dbl>  <dbl>      <dbl>    <dbl>    <dbl>   <dbl>
##  1 2021-05-31 NA     NA      -0.993  NA         NA       NA        0.507  NA
##  2 2021-05-28 NA      0.0907 NA      NA         NA        0.468   NA      NA
##  3 2021-05-26  0.927 NA      NA      NA          0.685   NA       NA      NA
```

```
##  4 2021-05-25 NA     NA       0.668  -0.452     NA       NA        0.608   0.376
##  5 2021-05-24 NA      1.41    NA       NA        NA        0.518   NA      NA
##  6 2021-05-19 -0.462 NA      -1.28    -0.639     0.597   NA        0.490   0.365
##  7 2021-05-18 NA     -0.564  NA       NA        NA        0.444   NA      NA
##  8 2021-05-13  0.147 NA      -2.49     0.864     0.635   NA        0.417   0.452
##  9 2021-05-12 NA     -2.67   NA       NA        NA        0.364   NA      NA
## 10 2021-05-07  0.942 NA       0.0902  0.795     0.686   NA        0.573   0.448
## # ... with 4,037 more rows
```

Q9

Generate density plots of the scaled values to compare the variability of the chosen indices. Plot one per facet.

```
four_indices %>%
  gather(GSPTSE:NYA, key=Index,value=Daily_Return_Perc.) %>%
  gather(var_GSPTSE:var_NYA, key=varIndex,value=Scaled_Price) %>%
  group_by(Date,Scaled_Price) %>%
  ggplot(aes(Scaled_Price,color=varIndex))+
  geom_density()+
  facet_wrap(~varIndex)
```