Statistical Computing with R - Assignment 1

Colin Yip, Student No: 3953629

Exercise 1

Formatting and output config.

```r
library(knitr)
library(formatR)
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 80))
```

Data Preparation

1.

```r
curr_dir <- getwd()
file_name <- "coca_cola.csv"
catted_file_path <- paste(curr_dir, file_name, sep = "/")
file_host <- "https://raw.githubusercontent.com"
file_path <- "kalilurrahman/Coca-ColaStockdata/main"
remote_file_name <- "Coca-Cola_stock_history.csv"
file_url <- paste(file_host, file_path, remote_file_name, sep = "/")
curl::curl_download(file_url, destfile = catted_file_path)
```

2.

```r
coca_cola_df <- read.csv(catted_file_path)
original_column_names <- names(coca_cola_df)
space_stripped_columns <- gsub(" ", "_", original_column_names, gsub)
datetime_converted_dt <- as.Date(coca_cola_df$Date, format = "%Y-%m-%d")
coca_cola_df$Date <- datetime_converted_dt
start_date <- as.Date("01-01-2000", format = "%d-%m-%Y")
end_date <- as.Date("31-12-2019", format = "%d-%m-%Y")
bool_date_index <- (coca_cola_df$Date >= start_date) & (coca_cola_df$Date <= end_date)
downselected_coca_cola_df <- subset(coca_cola_df, bool_date_index)
```

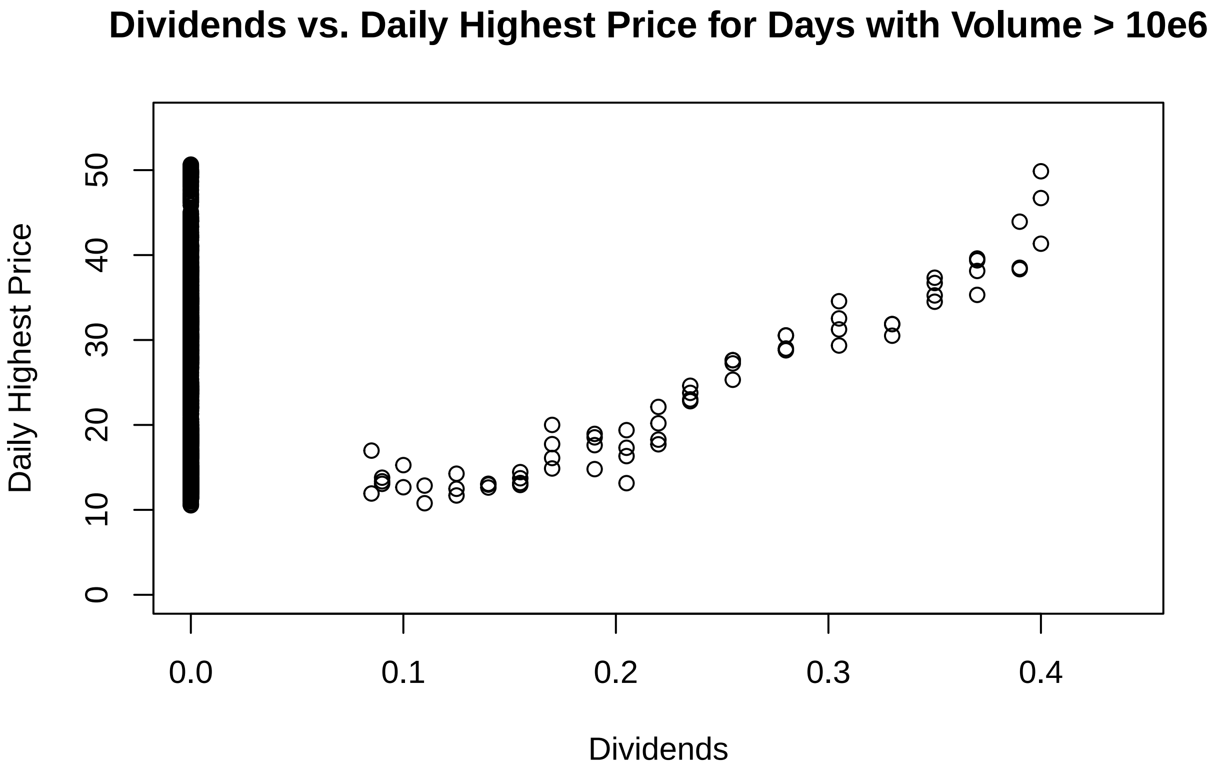
Data Manipulation

3.

```r
volume_exceeds_bool <- downselected_coca_cola_df$Volume > 9e6
downselected_coca_cola_df$HighVolume <- volume_exceeds_bool
```

4.

```r
high_volume_days_df <- subset(downselected_coca_cola_df, volume_exceeds_bool)
plot(high_volume_days_df$Dividends,
  high_volume_days_df$High,
  xlim = c(0, max(high_volume_days_df$Dividends) * 1.1),
  ylim = c(0, max(high_volume_days_df$High) * 1.1),
  xlab = "Dividends",
  ylab = "Daily Highest Price",
  main = "Dividends vs. Daily Highest Price for Days with Volume > 10e6"
)
```



There are two groups of points - the first being days with no dividends but persisting a high volume of trades, and the second group with a > 0 dividend value. The latter shows a strong positive correlation between dividends and daily highest price, which is most likely explained by the market reacting positively to high dividend yield days. Of course, not all days are dividend yield days and stock price is variable to more than just dividends, so the price fluctuates greatly and volume can still be high on days with no dividends.

5.

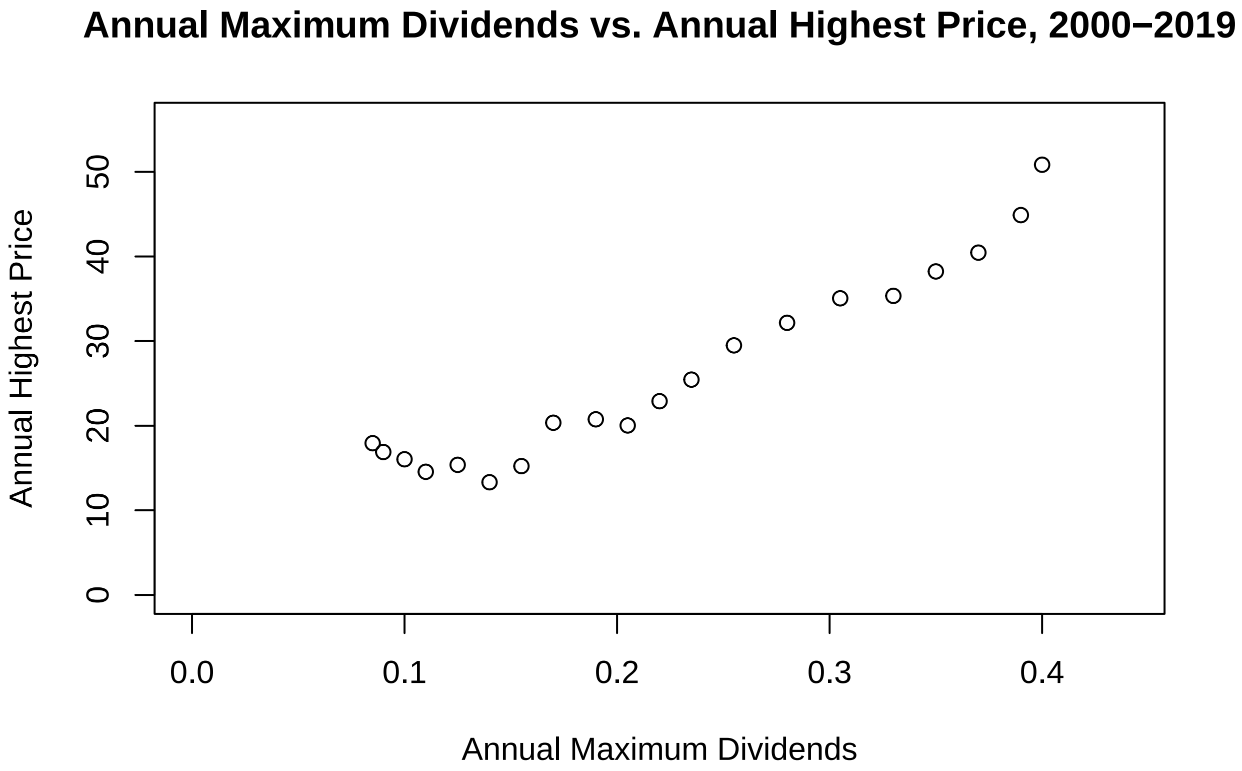
```r
downselected_coca_cola_df$DateYear <- format(downselected_coca_cola_df$Date, "%Y")
columns_classes <- sapply(downselected_coca_cola_df, class)
numeric_columns <- names(columns_classes[columns_classes %in% c("integer", "numeric")])
max_by_col <- function(r) apply(r, 2, max)
summary_by <- by(
  downselected_coca_cola_df[numeric_columns],
  downselected_coca_cola_df$DateYear,
  max_by_col
)
summary_df <- data.frame(do.call("rbind", summary_by))

knitr::kable(summary_df, caption = "Highest value of each numeric column, by year")
```

Table 1: Highest value of each numeric column, by year

	Open	High	Low	Close	Volume	Dividends	Stock.Splits
2000	17.89677	17.93028	17.62865	17.93028	27540400	0.085	0
2001	16.55177	16.89128	16.36503	16.51985	23672000	0.090	0
2002	15.89098	16.03218	15.80516	15.95742	37382000	0.100	0
2003	14.45425	14.55433	14.41137	14.51145	31069400	0.110	0
2004	15.24548	15.37480	15.02995	15.23111	61406200	0.125	0
2005	13.29848	13.31614	13.20434	13.31320	39747600	0.140	0
2006	15.15908	15.22386	15.06036	15.11589	53627400	0.155	0
2007	20.24963	20.35087	20.16736	20.27810	50341600	0.170	0
2008	20.68626	20.75270	20.47427	20.74321	79214400	0.190	0
2009	19.92781	20.02889	19.85033	19.91433	124169000	0.205	0
2010	22.77954	22.89421	22.75173	22.85598	63648600	0.220	0
2011	25.33793	25.44443	25.21014	25.28469	50096800	0.235	0
2012	29.37658	29.49261	29.23515	29.41647	98967500	0.255	2
2013	31.87937	32.16078	31.68684	31.90900	33575600	0.280	0
2014	34.60277	35.05459	34.45476	34.92216	55953500	0.305	0
2015	35.26362	35.34412	35.01409	35.28778	44065300	0.330	0
2016	38.04524	38.23181	37.72076	38.03712	40360300	0.350	0
2017	40.31787	40.45420	39.85777	40.41160	37526500	0.370	0
2018	44.81048	44.88933	44.30789	44.79157	32477700	0.390	0
2019	50.61188	50.84110	50.48352	50.74941	58905400	0.400	0

6.

```
plot(summary_df$Dividends,
  summary_df$High,
  xlim = c(0, max(summary_df$Dividends) * 1.1),
  ylim = c(0, max(summary_df$High) * 1.1),
  xlab = "Annual Maximum Dividends",
  ylab = "Annual Highest Price",
  main = "Annual Maximum Dividends vs. Annual Highest Price, 2000-2019"
)
```



Taking annual maximums instead of daily grain data, it is evident that there is a positive correlation between dividends and high prices - however because this does not respect day wise indexes (i.e. the maximum daily dividend of a given year is not necessarily aligned to the highest daily price of a given year), this is a very broad stroke association. Furthermore, this eliminates the large amount of variance visible in daily price even when there are no dividends. Without that information, it appears that the only dependent data point that share price depends on is dividends, which is false.

7.

```r
return_df <- data.frame(row.names = c("Lowest Value", "Date of Lowest Value", "High-Low Diff."))
for (i in 1:length(numeric_columns)) {
  col_name <- numeric_columns[i]
  col_min <- min(downselected_coca_cola_df[col_name])
  col_max <- max(downselected_coca_cola_df[col_name])
  indexes_of_min <- downselected_coca_cola_df[col_name] == col_min
  date_of_min <- subset(downselected_coca_cola_df$Date, indexes_of_min)
  min_date_of_min <- format(min(date_of_min), "%Y-%m-%d")
  col_diff <- col_max - col_min
  return_df[col_name] <- c(col_min, min_date_of_min, col_diff)
}

knitr::kable(return_df, caption = "Summary statistics of numeric columns")
```

Table 2: Summary statistics of numeric columns

	Open	High	Low	Close	Volume	Dividends	Stock.Splits
Lowest Value	10.39588929	10.53599404	10.37066808	10.38748169	2147400	0	0
Date of Lowest Value	2003-03-11	2003-03-10	2003-03-05	2003-03-10	2003-12-26	2000-01-03	2000-01-03
High-Low Diff.	40.21598925	40.30510497	40.1128473	40.36192703	122021600	0.4	2

Exercise 2

```r
beer_generator <- function(x_vec) {
  generic_error_msg <- "beer_generator requires a vector of length 1 containing integer x, x > 0"
  if (length(x_vec) != 1) {
    stop(paste(generic_error_msg, ". Length(x_vec) > 1"))
  }
  x <- x_vec[1]
  if (x %% 1 != 0) {
    stop(paste(generic_error_msg, ". x_vec[1] is not integer"))
  }
  if (!(x > 0)) {
    stop(paste(generic_error_msg, ". x_vec[1] <= 0"))
  }

  header_text <- sprintf("%s Bottles of Beers Song", x)
  ge_1_first_clause_fmt <- "%s bottles of beer on the wall, %s bottles of beer."
  e_1_first_clause_fmt <- "%s bottle of beer on the wall, %s bottle of beer."
  plural_text_fmt <- "Take one down, pass it around, %s bottles of beer on the wall..."
  singular_plus_one_text_fmt <- "Take one down, pass it around, %s bottle of beer on the wall..."
  singular_text_fmt <- "Take one down, pass it around, no more bottles of beer on the wall."

  out_string <- header_text

  while (x > 0) {
    if (x == 1) {
      fmted_first_string <- sprintf(e_1_first_clause_fmt, x, x)
      fmted_second_string <- singular_text_fmt
    } else if (x == 2) {
      fmted_first_string <- sprintf(ge_1_first_clause_fmt, x, x)
      fmted_second_string <- sprintf(singular_plus_one_text_fmt, x - 1)
    } else {
      fmted_first_string <- sprintf(ge_1_first_clause_fmt, x, x)
      fmted_second_string <- sprintf(plural_text_fmt, x - 1)
    }
    fmted_string <- paste(fmted_first_string, fmted_second_string)
    out_string <- paste(out_string, "\n\n", fmted_string)
    x <- x - 1
  }

  writeLines(strwrap(out_string, width = 100))
}
```

```r
test_vec <- c(5)
beer_generator(test_vec)
```

```
## 5 Bottles of Beers Song
##
## 5 bottles of beer on the wall, 5 bottles of beer. Take one down, pass it around, 4 bottles of beer
## on the wall...
##
## 4 bottles of beer on the wall, 4 bottles of beer. Take one down, pass it around, 3 bottles of beer
## on the wall...
##
## 3 bottles of beer on the wall, 3 bottles of beer. Take one down, pass it around, 2 bottles of beer
## on the wall...
##
## 2 bottles of beer on the wall, 2 bottles of beer. Take one down, pass it around, 1 bottle of beer
## on the wall...
##
## 1 bottle of beer on the wall, 1 bottle of beer. Take one down, pass it around, no more bottles of
## beer on the wall.
```