Report - Project 1

DATA130021 Financial Econometrics

Deng Qisheng

2021/03/31

1 Introduction

A return, also known as a financial return, in its simplest terms, is the money made or lost on an investment over some period of time. This project aims to analysis returns by statistical modeling on historical stock price data. Stock price data of Google Inc. from 2012/01/01 to 2018/12/31, downloaded from Yahoo! Finance, which includes daily, weekly and monthly data, are used to analyze in this project.

Before we analyze the stock data, environment should be set up in R.

```
library(DistributionUtils)
library(ggplot2)
library(goftest)
library(gridExtra)
library(lubridate)
library(tseries)
```

2 Data preparation

Data preparation is the process of cleaning and transforming raw data prior to processing, analysis and modeling. In this project, prices, returns and log returns are key variables to concern about. Load in the dataset respectively.

```
# Daily data
daily.data <- read.csv("./data/GOOG_daily.csv")

daily.len <- dim(daily.data)[1]

daily.date <- as.Date(daily.data$Date)
daily.year <- as.factor(year(daily.date))
daily.price <- daily.data$Close
daily.return <- c(0,
    daily.price[2:daily.len] / daily.price[1:(daily.len - 1)] - 1)
daily.log.return <- log(1 + daily.return)

daily.stock <- data.frame(
    date=daily.date,
    year=daily.year,
    price=daily.price,
    return=daily.return,
    log.return=daily.log.return)
```

```r
# Weekly data
weekly.data <- read.csv("./data/GOOG_weekly.csv")

weekly.len <- dim(weekly.data)[1]

weekly.date <- as.Date(weekly.data$Date)
weekly.year <- as.factor(year(weekly.date))
weekly.price <- weekly.data$Close
weekly.return <- c(0,
    weekly.price[2:weekly.len] / weekly.price[1:(weekly.len - 1)] - 1)
weekly.log.return <- log(1 + weekly.return)

weekly.stock <- data.frame(
    date=weekly.date,
    year=weekly.year,
    price=weekly.price,
    return=weekly.return,
    log.return=weekly.log.return)

# Monthly data
monthly.data <- read.csv("./data/GOOG_monthly.csv")

monthly.len <- dim(monthly.data)[1]

monthly.date <- as.Date(monthly.data$Date)
monthly.year <- as.factor(year(monthly.date))
monthly.price <- monthly.data$Close
monthly.return <- c(0,
    monthly.price[2:monthly.len] / monthly.price[1:(monthly.len - 1)] - 1)
monthly.log.return <- log(1 + monthly.return)

monthly.stock <- data.frame(
    date=monthly.date,
    year=monthly.year,
    price=monthly.price,
    return=monthly.return,
    log.return=monthly.log.return)
```

Following are summaries of stock data.

```r
# Daily data
knitr::kable(x=summary(daily.stock[, c(-1, -2)]),
    caption="Summary for Daily Stock Data of Google Inc.",
    col.names=c("Price", "Returns", "Log Returns"), align=c("c", "c", "c"))
```

Table 1: Summary for Daily Stock Data of Google Inc.

Price	Returns	Log Returns
Min. : 278.5	Min. :-0.0837751	Min. :-0.0874934
1st Qu.: 451.7	1st Qu.:-0.0061061	1st Qu.:-0.0061248
Median : 599.6	Median : 0.0004119	Median : 0.0004118
Mean : 671.4	Mean : 0.0007539	Mean : 0.0006485
3rd Qu.: 835.5	3rd Qu.: 0.0082377	3rd Qu.: 0.0082040
Max. :1268.3	Max. : 0.1605243	Max. : 0.1488719

```r
# Weekly data
knitr::kable(x=summary(weekly.stock[, c(-1, -2)]),
    caption="Summary for Weekly Stock Data of Google Inc.",
    col.names=c("Price", "Returns", "Log Returns"), align=c("c", "c", "c"))
```

Table 2: Summary for Weekly Stock Data of Google Inc.

Price	Returns	Log Returns
Min. : 281.2	Min. :-0.100517	Min. :-0.105935
1st Qu.: 452.9	1st Qu.:-0.014348	1st Qu.:-0.014452
Median : 600.7	Median : 0.004451	Median : 0.004441
Mean : 671.4	Mean : 0.003731	Mean : 0.003189
3rd Qu.: 829.6	3rd Qu.: 0.022720	3rd Qu.: 0.022466
Max. :1238.5	Max. : 0.269368	Max. : 0.238519

```r
# Monthly data
knitr::kable(x=summary(monthly.stock[, c(-1, -2)]),
    caption="Summary for Monthly Stock Data of Google Inc.",
    col.names=c("Price", "Returns", "Log Returns"), align=c("c", "c", "c"))
```

Table 3: Summary for Monthly Stock Data of Google Inc.

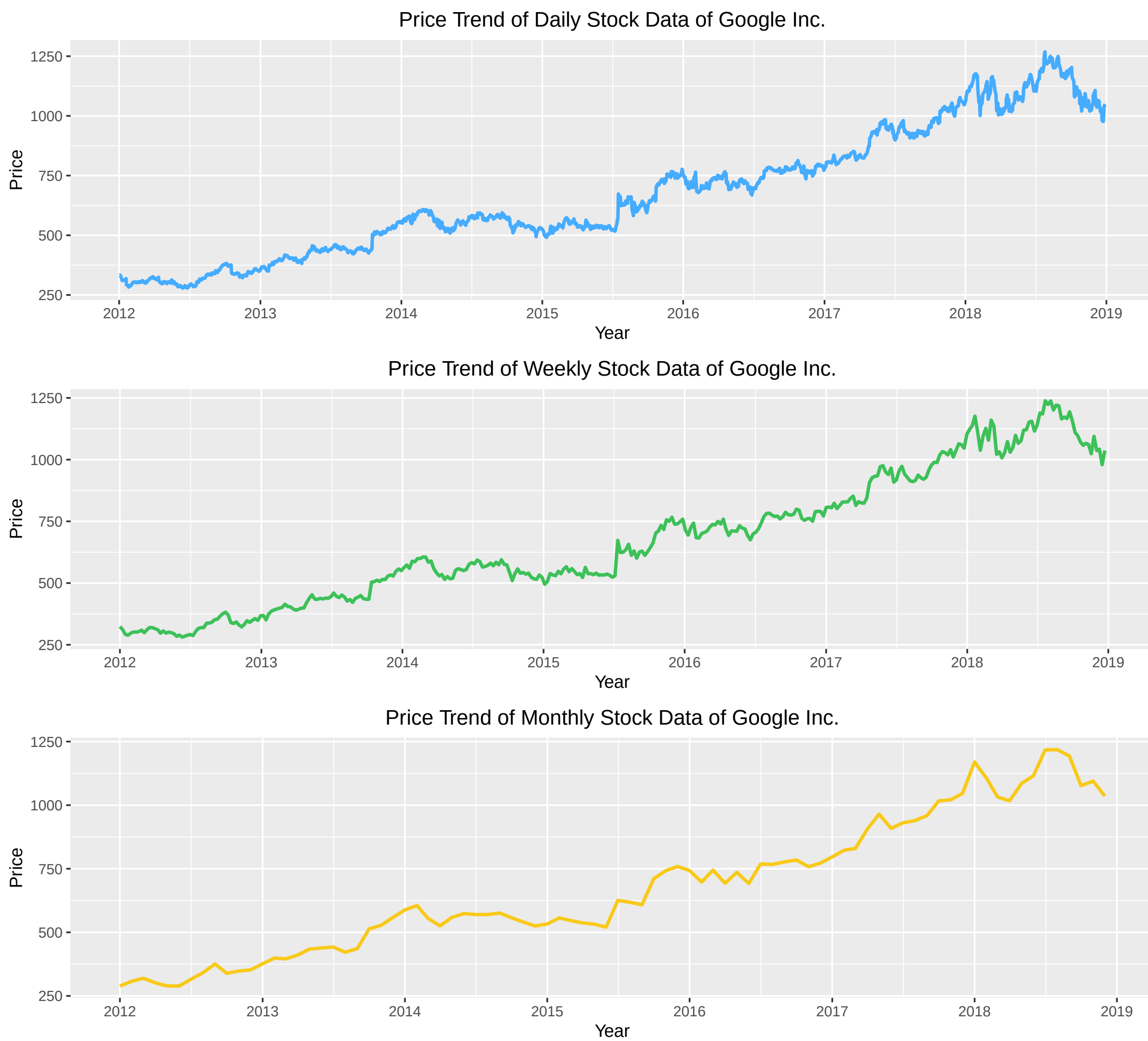
Price	Returns	Log Returns
Min. : 289.0	Min. :-0.09834	Min. :-0.10352
1st Qu.: 495.6	1st Qu.:-0.01904	1st Qu.:-0.01923
Median : 607.0	Median : 0.01313	Median : 0.01304
Mean : 674.9	Mean : 0.01691	Mean : 0.01520
3rd Qu.: 848.7	3rd Qu.: 0.05447	3rd Qu.: 0.05304
Max. :1218.2	Max. : 0.20192	Max. : 0.18392

3 Descriptive Data Analysis

Usually, the price trends of stock data are the most concerned in financial analysis. Price trends are shown as below.

```r
# Daily data
daily.price.trend <- ggplot(data=daily.stock, aes(x=date, y=price)) +
    geom_line(aes(colour="price"), size=1, group=0) +
    labs(title="Price Trend of Daily Stock Data of Google Inc.", x="Year", y="Price") +
    theme(plot.title=element_text(hjust=0.5), legend.position="none") +
    scale_colour_manual(values=c("price"="#45ACFF")) +
    scale_x_date(date_labels="%Y", date_breaks="1 year")

# Weekly data
weekly.price.trend <- ggplot(data=weekly.stock, aes(x=date, y=price)) +
    geom_line(aes(colour="price"), size=1, group=0) +
    labs(title="Price Trend of Weekly Stock Data of Google Inc.", x="Year", y="Price") +
    theme(plot.title=element_text(hjust=0.5), legend.position="none") +
    scale_colour_manual(values=c("price"="#3DC159")) +
    scale_x_date(date_labels="%Y", date_breaks="1 year")
```

```r
# Monthly data
monthly.price.trend <- ggplot(data=monthly.stock, aes(x=date, y=price)) +
    geom_line(aes(colour="price"), size=1, group=0) +
    labs(title="Price Trend of Monthly Stock Data of Google Inc.", x="Year", y="Price") +
    theme(plot.title=element_text(hjust=0.5), legend.position="none") +
    scale_colour_manual(values=c("price"="#F9C918")) +
    scale_x_date(date_labels="%Y", date_breaks="1 year")

# Plot
grid.arrange(daily.price.trend, weekly.price.trend, monthly.price.trend, nrow=3)
```

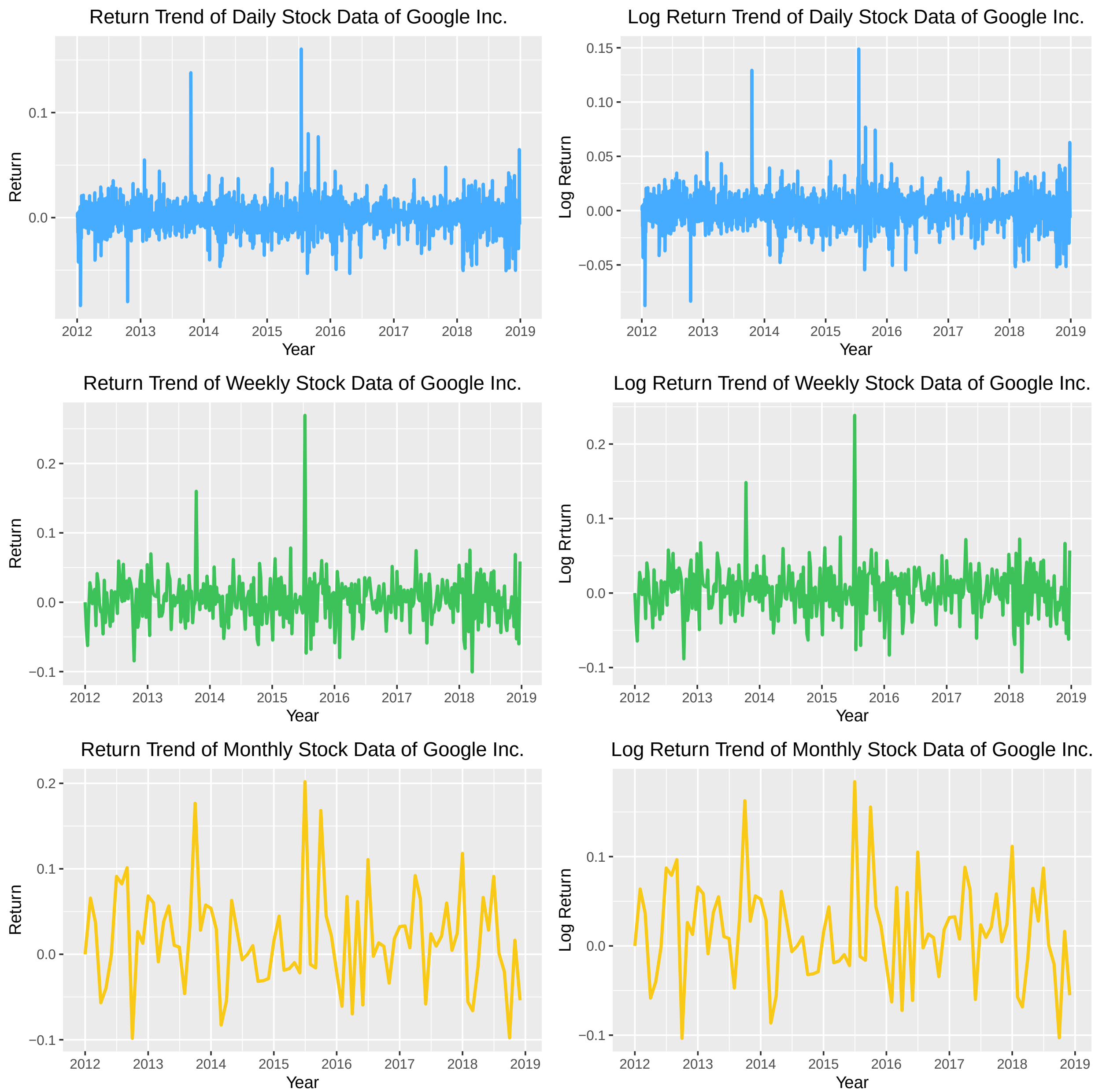


The prices of stock data of Google Inc. presents an increasing trend from 2012 to 2018. In detail, we find that prices were fluctuated in 2014 and 2018. Around June 2015, prices suddenly surged in a short time. However, prices had been falling during second half of 2018, which means a long bear market.

Similarly, we also consider the trend of returns and log returns as below.

```r
# Daily data
daily.return.trend <- ggplot(data=daily.stock, aes(x=date, y=return)) +
    geom_line(aes(colour="return"), size=1, group=0) +
    labs(title="Return Trend of Daily Stock Data of Google Inc.",
         x="Year", y="Return") +
    theme(plot.title=element_text(hjust=0.5), legend.position="none") +
    scale_colour_manual(values=c("return"="#45ACFF")) +
    scale_x_date(date_labels="%Y", date_breaks="1 year")

daily.log.return.trend <- ggplot(data=daily.stock, aes(x=date, y=log.return)) +
    geom_line(aes(colour="log.return"), size=1, group=0) +
    labs(title="Log Return Trend of Daily Stock Data of Google Inc.",
         x="Year", y="Log Return") +
    theme(plot.title=element_text(hjust=0.5), legend.position="none") +
    scale_colour_manual(values=c("log.return"="#45ACFF")) +
    scale_x_date(date_labels="%Y", date_breaks="1 year")

# Weekly data
weekly.return.trend <- ggplot(data=weekly.stock, aes(x=date, y=return)) +
    geom_line(aes(colour="return"), size=1, group=0) +
    labs(title="Return Trend of Weekly Stock Data of Google Inc.",
         x="Year", y="Return") +
    theme(plot.title=element_text(hjust=0.5), legend.position="none") +
    scale_colour_manual(values=c("return"="#3DC159")) +
    scale_x_date(date_labels="%Y", date_breaks="1 year")

weekly.log.return.trend <- ggplot(data=weekly.stock, aes(x=date, y=log.return)) +
    geom_line(aes(colour="log.return"), size=1, group=0) +
    labs(title="Log Return Trend of Weekly Stock Data of Google Inc.",
         x="Year", y="Log Rrturn") +
    theme(plot.title=element_text(hjust=0.5), legend.position="none") +
    scale_colour_manual(values=c("log.return"="#3DC159")) +
    scale_x_date(date_labels="%Y", date_breaks="1 year")

# Monthly data
monthly.return.trend <- ggplot(data=monthly.stock, aes(x=date, y=return)) +
    geom_line(aes(colour="return"), size=1, group=0) +
    labs(title="Return Trend of Monthly Stock Data of Google Inc.",
         x="Year", y="Return") +
    theme(plot.title=element_text(hjust=0.5), legend.position="none") +
    scale_colour_manual(values=c("return"="#F9C918")) +
    scale_x_date(date_labels="%Y", date_breaks="1 year")

monthly.log.return.trend <- ggplot(data=monthly.stock, aes(x=date, y=log.return)) +
    geom_line(aes(colour="log.return"), size=1, group=0) +
    labs(title="Log Return Trend of Monthly Stock Data of Google Inc.",
         x="Year", y="Log Return") +
    theme(plot.title=element_text(hjust=0.5), legend.position="none") +
    scale_colour_manual(values=c("log.return"="#F9C918")) +
    scale_x_date(date_labels="%Y", date_breaks="1 year")

# Plot
grid.arrange(
```

```
    daily.return.trend,
    daily.log.return.trend,
    weekly.return.trend,
    weekly.log.return.trend,
    monthly.return.trend,
    monthly.log.return.trend,
    nrow=3, ncol=2)
```

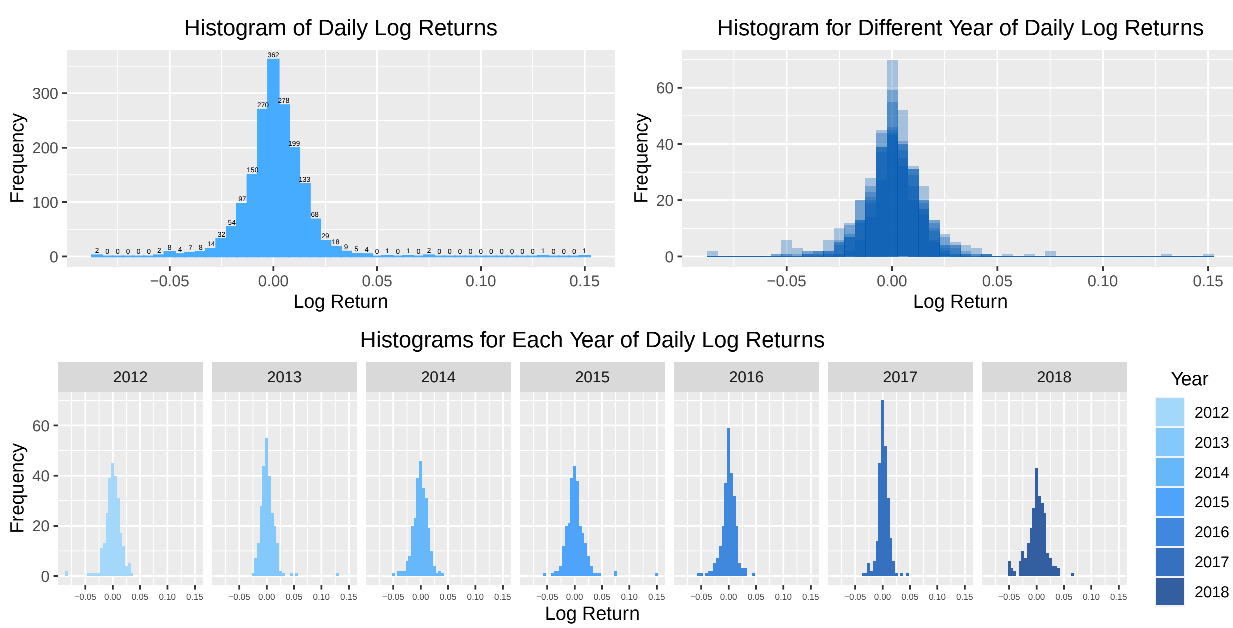


From the figures above, we find that the corresponding log returns fluctuate around zero, except for several points.

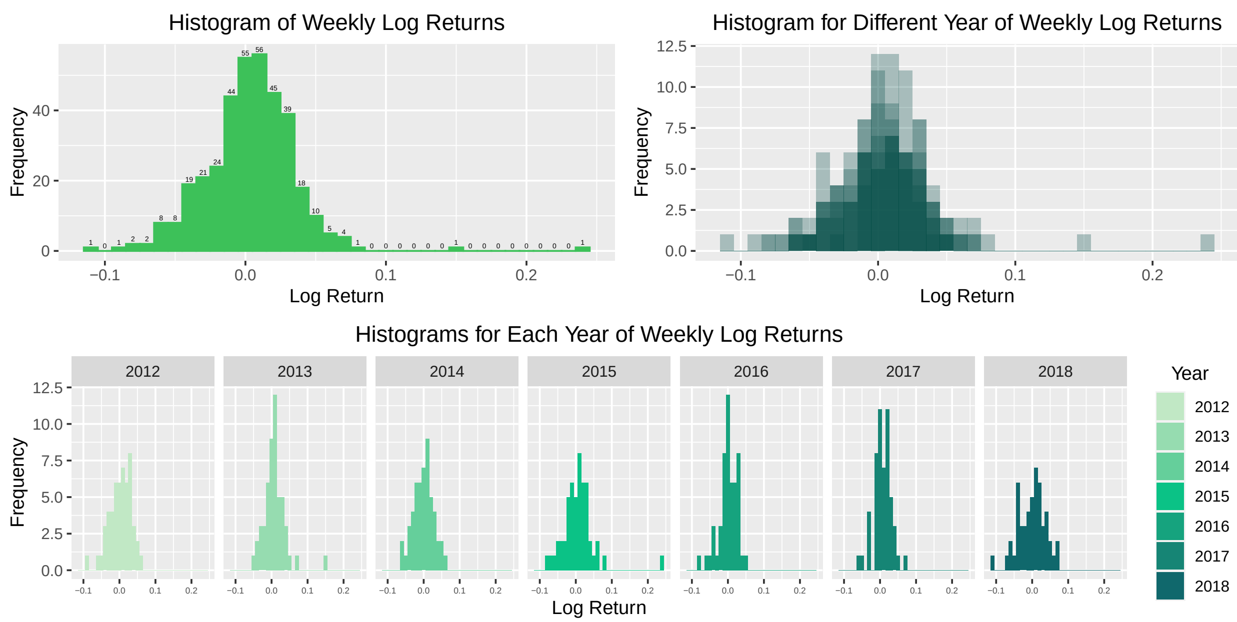
Normal distribution is often used for demostration in mathematical modeling. Hence, we assume that the log returns of stock data satisfy normal distribution. We should use histograms and QQ-plots to verify our assumptions.

```r
# Daily data
daily.total.hist <- ggplot(data=daily.stock, aes(x=log.return)) +
    geom_histogram(binwidth=0.005, fill="#45ACFF", color="#45ACFF") +
    geom_text(aes(label=as.character(..count..)),
        stat="bin", binwidth=0.005, vjust=-0.5, size=1.5) +
    labs(title="Histogram of Daily Log Returns",
        x="Log Return", y="Frequency") +
    theme(plot.title=element_text(hjust=0.5))

daily.overlay.hist <- ggplot(data=daily.stock, aes(x=log.return, fill=year)) +
    geom_histogram(binwidth=0.005, alpha=0.3, position="identity") +
    labs(title="Histogram for Different Year of Daily Log Returns",
        x="Log Return", y="Frequency") +
    theme(plot.title=element_text(hjust=0.5), legend.position="none") +
    scale_fill_manual(values=c(
        "#0059B2", "#0059B2", "#0059B2", "#0059B2", "#0059B2", "#0059B2", "#0059B2"))

daily.year.hist <- ggplot(data=daily.stock, aes(x=log.return, fill=year)) +
    geom_histogram(binwidth=0.005) +
    labs(title="Histograms for Each Year of Daily Log Returns",
        x="Log Return", y="Frequency") +
    theme(plot.title=element_text(hjust=0.5),
        axis.text.x = element_text(size=5)) +
    facet_grid(~year) +
    scale_fill_manual(name="   Year", values=c(
        "#A4D8FB", "#84C9FB", "#66B8FB", "#4EA4FB", "#4088DD", "#3571BE", "#335FA0"))

# Weekly data
weekly.total.hist <- ggplot(data=weekly.stock, aes(x=log.return)) +
    geom_histogram(binwidth=0.01, fill="#3DC159", color="#3DC159") +
    geom_text(aes(label=as.character(..count..)),
        stat="bin", binwidth=0.01, vjust=-0.5, size=1.5) +
    labs(title="Histogram of Weekly Log Returns",
        x="Log Return", y="Frequency") +
    theme(plot.title=element_text(hjust=0.5))

weekly.overlay.hist <- ggplot(data=weekly.stock, aes(x=log.return, fill=year)) +
    geom_histogram(binwidth=0.01, alpha=0.3, position="identity") +
    labs(title="Histogram for Different Year of Weekly Log Returns",
        x="Log Return", y="Frequency") +
    theme(plot.title=element_text(hjust=0.5), legend.position="none") +
    scale_fill_manual(values=c(
        "#044E48", "#044E48", "#044E48", "#044E48", "#044E48", "#044E48", "#044E48"))

weekly.year.hist <- ggplot(data=weekly.stock, aes(x=log.return, fill=year)) +
    geom_histogram(binwidth=0.01) +
    labs(title="Histograms for Each Year of Weekly Log Returns",
        x="Log Return", y="Frequency") +
    theme(plot.title=element_text(hjust=0.5),
        axis.text.x = element_text(size=5)) +
    facet_grid(~year) +
    scale_fill_manual(name="   Year", values=c(
        "#C1E8C5", "#96DCB0", "#65CF9B", "#0BC286", "#16A37E", "#168575", "#10686C"))
```

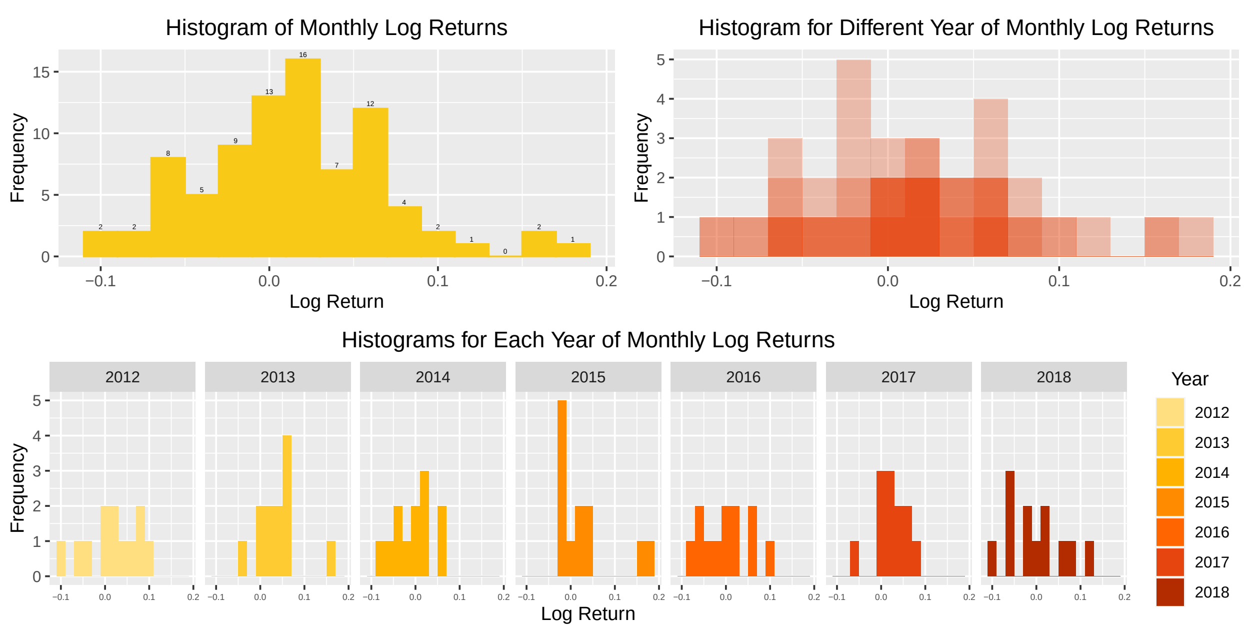
```
# Monthly data
monthly.total.hist <- ggplot(data=monthly.stock, aes(x=log.return)) +
    geom_histogram(binwidth=0.02, fill="#F9C918", color="#F9C918") +
    geom_text(aes(label=as.character(..count..)),
        stat="bin", binwidth=0.02, vjust=-0.5, size=1.5) +
    labs(title="Histogram of Monthly Log Returns",
         x="Log Return", y="Frequency") +
    theme(plot.title=element_text(hjust=0.5))

monthly.overlay.hist <- ggplot(data=monthly.stock, aes(x=log.return, fill=year)) +
    geom_histogram(binwidth=0.02, alpha=0.3, position="identity") +
    labs(title="Histogram for Different Year of Monthly Log Returns",
        x="Log Return", y="Frequency") +
    theme(plot.title=element_text(hjust=0.5), legend.position="none") +
    scale_fill_manual(values=c(
        "#E6450F", "#E6450F", "#E6450F", "#E6450F", "#E6450F", "#E6450F", "#E6450F"))

monthly.year.hist <- ggplot(data=monthly.stock, aes(x=log.return, fill=year)) +
    geom_histogram(binwidth=0.02) +
    labs(title="Histograms for Each Year of Monthly Log Returns",
        x="Log Return", y="Frequency") +
    theme(plot.title=element_text(hjust=0.5),
        axis.text.x = element_text(size=5)) +
    facet_grid(~year) +
    scale_fill_manual(name="   Year", values=c(
        "#FFDF80", "#FFCB33", "#FFB200", "#FF8C00", "#FF6500", "#E6450F", "#B22C00"))

# Plot
grid.arrange(daily.total.hist, daily.overlay.hist, daily.year.hist,
             layout_matrix=rbind(c(1, 2), c(3, 3)))
```



```r
grid.arrange(weekly.total.hist, weekly.overlay.hist, weekly.year.hist,
             layout_matrix=rbind(c(1, 2), c(3, 3)))
```

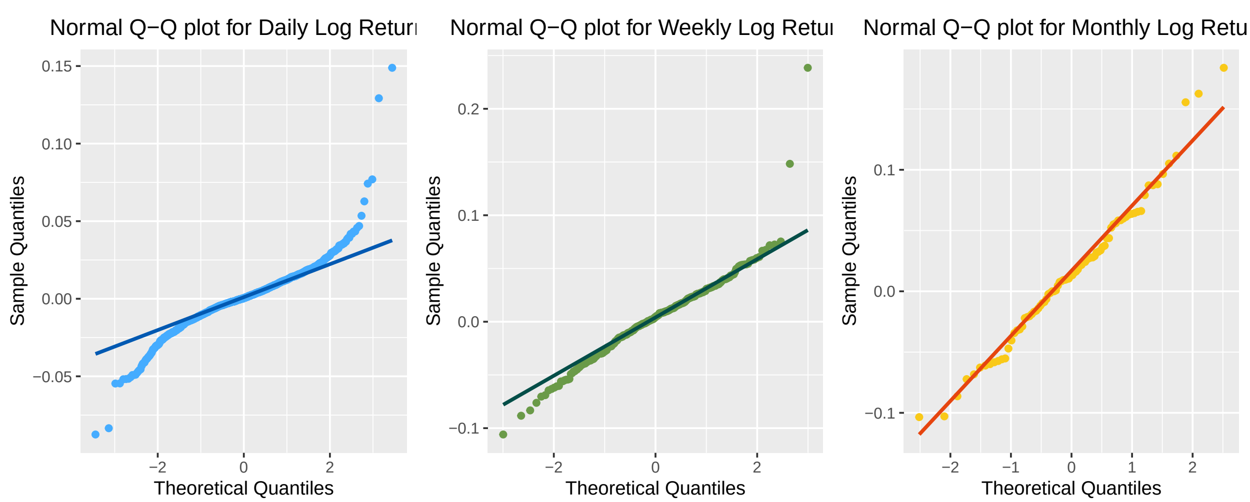



```r
grid.arrange(monthly.total.hist, monthly.overlay.hist, monthly.year.hist,
             layout_matrix=rbind(c(1, 2), c(3, 3)))
```



For each figure, the plots on the left upper corner show the histogram for whole data. The plots on the right upper corner show the overlay of histograms for different year, which means that the darker area represent the more common points. The plots at the bottom show the histograms for each year. According to these results, we find that all histograms have some features of normal distribution. In advanced, we should use QQ-plots to check the normality.

```r
# Daily data
daily.qqplot <- ggplot(data=daily.stock, aes(sample=log.return)) +
    geom_qq(color="#45ACFF") +
    geom_qq_line(color="#0059B2", size=1) +
    labs(title="Normal Q-Q plot for Daily Log Returns",
        x="Theoretical Quantiles", y="Sample Quantiles") +
    theme(plot.title=element_text(hjust=0.5))

# Weekly data
weekly.qqplot <- ggplot(data=weekly.stock, aes(sample=log.return)) +
    geom_qq(color="#6A9A48") +
    geom_qq_line(color="#044E48", size=1) +
    labs(title="Normal Q-Q plot for Weekly Log Returns",
        x="Theoretical Quantiles", y="Sample Quantiles") +
    theme(plot.title=element_text(hjust=0.5))

# Monthly data
monthly.qqplot <- ggplot(data=monthly.stock, aes(sample=log.return)) +
    geom_qq(color="#F9C918") +
    geom_qq_line(color="#E6450F", size=1) +
    labs(title="Normal Q-Q plot for Monthly Log Returns",
        x="Theoretical Quantiles", y="Sample Quantiles") +
    theme(plot.title=element_text(hjust=0.5))

# Plot
grid.arrange(daily.qqplot, weekly.qqplot, monthly.qqplot, nrow=1)
```



From the QQ-plots above, we find that only monthly log returns data satisfy normal distribution approximately, which means that the normality assumptions may not be correct.

4 Statistical Modeling

In this part, we are going to use some common models to test for normality.

```r
# Daily data
s.daily <- skewness(daily.log.return)
```

```r
k.daily <- kurtosis(daily.log.return)

# weekly data
s.weekly <- skewness(weekly.log.return)
k.weekly <- kurtosis(weekly.log.return)

# Monthly data
s.monthly <- skewness(monthly.log.return)
k.monthly <- kurtosis(monthly.log.return)

# Plot
knitr::kable(
    x=data.frame(
        "Type"=c("Daily", "Weekly", "Monthly"),
        "Skewness"=c(s.daily, s.weekly, s.monthly),
        "Kurtosis"=c(k.daily, k.weekly, k.monthly)),
    caption="Estimation for Skewness and Kurtosis of Log Returns",
    align=c("c", "c", "c"))
```

Table 4: Estimation for Skewness and Kurtosis of Log Returns

Type	Skewness	Kurtosis
Daily	0.6829710	12.538519
Weekly	0.9303101	7.806097
Monthly	0.3975733	0.459198

From the table above, we find that all of three log returns dataset satisfy positively skewed distribution. Comparing to normal distribution, daily log returns and weekly log returns have an extremely large kurtosis, which means they should not satisfy normal distribution. For monthly log returns, the kurtosis is 0.46, which may satisfy normal distribution. Next step, we are going to use Kolmogorov-Smirnov Test, Cramer-von Mises Test and Jarque-Bera Test to test normality.

```r
# Daily data
ks.daily <- ks.test(daily.log.return, "pnorm",
                    mean=mean(daily.log.return), sd=sd(daily.log.return))
cvm.daily <- cvm.test(daily.log.return, "pnorm",
                    mean=mean(daily.log.return), sd=sd(daily.log.return))
jarque.bera.daily <- jarque.bera.test(daily.log.return)

# Weekly data
ks.weekly <- ks.test(weekly.log.return, "pnorm",
                     mean=mean(weekly.log.return), sd=sd(weekly.log.return))
cvm.weekly <- cvm.test(weekly.log.return, "pnorm",
                     mean=mean(weekly.log.return), sd=sd(weekly.log.return))
jarque.bera.weekly <- jarque.bera.test(weekly.log.return)

# Monthly data
ks.monthly <- ks.test(monthly.log.return, "pnorm",
                      mean=mean(monthly.log.return), sd=sd(monthly.log.return))
cvm.monthly <- cvm.test(monthly.log.return, "pnorm",
                      mean=mean(monthly.log.return), sd=sd(monthly.log.return))
jarque.bera.monthly <- jarque.bera.test(monthly.log.return)
```

```r
# Plot
knitr::kable(
    x=data.frame(
        "Statistic"=c(
            "Kolmogorov-Smirnov Test",
            "Cramer-von Mises Test",
            "Jarque-Bera Test"),
        "Daily"=c(
            ks.daily$statistic,
            cvm.daily$statistic,
            jarque.bera.daily$statistic),
        "Weekly"=c(
            ks.weekly$statistic,
            cvm.weekly$statistic,
            jarque.bera.weekly$statistic),
        "Monthly"=c(
            ks.monthly$statistic,
            cvm.monthly$statistic,
            jarque.bera.monthly$statistic)),
    caption="Statistics in Three Tests for Log Returns",
    col.names=c("Statistic", "Daily", "Weekly", "Monthly"),
    align=c("c", "c", "c", "c"))
```

Table 5: Statistics in Three Tests for Log Returns

	Statistic	Daily	Weekly	Monthly
D	Kolmogorov-Smirnov Test	0.073385	0.0567711	0.0635014
omega2	Cramer-von Mises Test	3.448177	0.3265847	0.0486979
X-squared	Jarque-Bera Test	11692.012436	993.9777922	3.3260336

```r
knitr::kable(
    x=data.frame(
        "P.value"=c(
            "Kolmogorov-Smirnov Test",
            "Cramer-von Mises Test",
            "Jarque-Bera Test"),
        "Daily"=c(
            format(ks.daily$p.value, scientific=T, digits=3, nsmall=3),
            format(cvm.daily$p.value, scientific=T, digits=3, nsmall=3),
            if (jarque.bera.daily$p.value < 2.2e-16) {"< 2.2e-16"}),
        "Weekly"=c(
            format(ks.weekly$p.value, scientific=F, digits=3, nsmall=3),
            format(cvm.weekly$p.value, scientific=F, digits=3, nsmall=3),
            if (jarque.bera.weekly$p.value < 2.2e-16) {"< 2.2e-16"}),
        "Monthly"=c(
            format(ks.monthly$p.value, scientific=F, digits=3, nsmall=3),
            format(cvm.monthly$p.value, scientific=F, digits=3, nsmall=3),
            format(jarque.bera.monthly$p.value, scientific=F, digits=3, nsmall=3))),
    caption="P-value in Three Tests for Log Returns",
    col.names=c("P-value", "Daily", "Weekly", "Monthly"),
    align=c("c", "c", "c", "c"))
```

Table 6: P-value in Three Tests for Log Returns

P-value	Daily	Weekly	Monthly
Kolmogorov-Smirnov Test	1.18e-08	0.190	0.887
Cramer-von Mises Test	7.55e-09	0.114	0.885
Jarque-Bera Test	$<$ 2.2e-16	$<$ 2.2e-16	0.190

In all hypothesis tests for normality, if P-value is smaller than the significance level, then we reject the null hypothesis and consider that the samples are not sampling from normal distribution.

In Kolmogorov-Smirnov Test, we conclude that daily log returns do not satisfy normal distribution over 0.01 level of significance. But we can not give a conclusion on weekly log returns and monthly log returns.

In Cramer-von Mises Test, we have a similar conculsion that daily log returns do not satisfy normal distribution over 0.01 level of significance. But we also can not give a conclusion on weekly log returns and monthly log returns.

In Jarque-Bera Test, we have sufficient evidences to conclude that daily log returns and weekly log returns do not satisfy normal distribution over 0.01 level of significance. The P-value of monthly log returns is 0.19, which means that we can not reject the null hypothesis at 0.05 level of significance.

5 Conclusion

In this project, we have analyzed the prices, returns and log returns data in daily, weekly and monthly views. In descriptive analysis, we can conclude that daily log returns and weekly log returns do not satisfiy the normal distribution by QQ-plots. In statistical modeling, we use several normality tests to check the log returns data. Eventually, we know that daily log returns and weekly log returns do not have any normality over 0.01 level of significance. However, we can not apply this conclusion to monthly log returns data and we need more analysis to check its normality.

In another view, although all stock price data are sampling from same company (Google Inc.) and same period (From 2012/01/01 to 2018/12/31), the results for normality tests are different, which means that we should fit the a suitable model by choosing appropriate periods.