Data Analysis

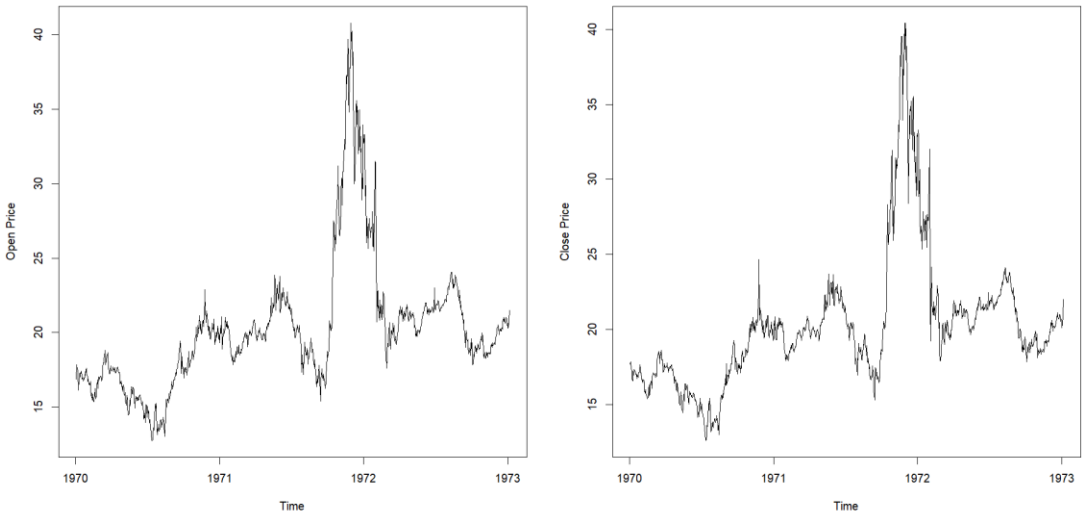
To discover the characteristics of each stock intuitively, we first conducted some basic data analysis before designing strategies. This data analysis part includes concrete feature analysis of particular stocks (taking stock 01 – "01.csv" as the example in this report) as well as comparisons between different stocks.

Part 1 – Data Visualizing

We run the following R code to read the stock data and plot its important features in a graph. Taking "01.csv" as an example:

```
Stock <- read.csv("01.csv", header=TRUE, sep=",", dec=".", fileEncoding="UTF-8-
BOM")
StockDates <- as.Date(Stock$Index)

par(mfrow=c(1,2))
plot(StockDates, Stock$Open, type="l", xlab="Time", ylab="Open Price")
plot(StockDates, Stock$Close, type="l", xlab="Time", ylab="Close Price")
```

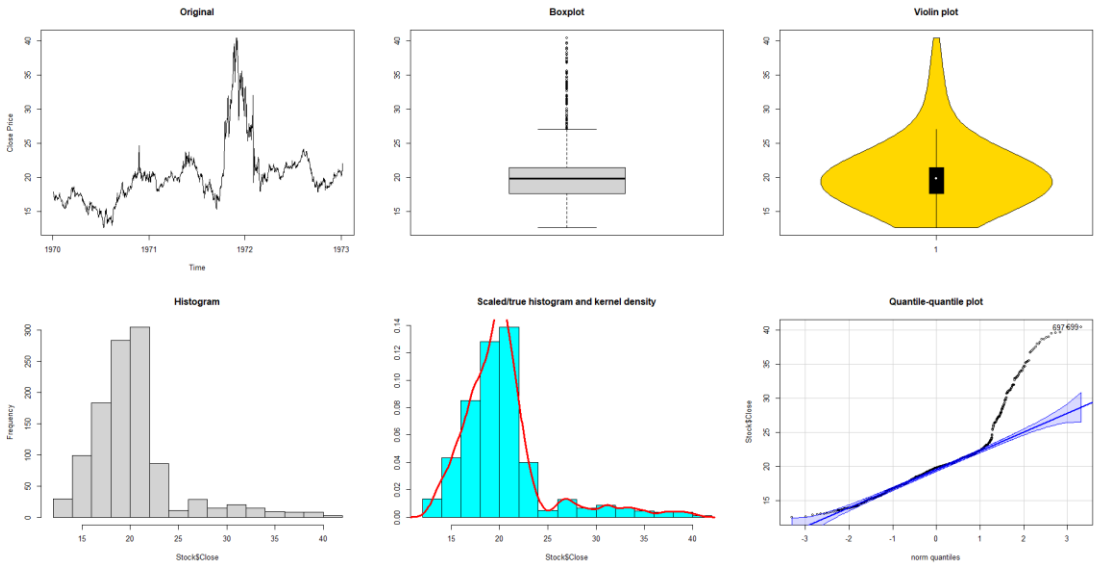


Stock 1 experienced a downturn period in the middle of the 1970s and great prosperity in the 1972s. This stock price remained stable in other observation years. We can also see that the general trends of stock 1's open price and close price are roughly similar, whereas obvious difference occurs in the late 1970s and early 1972s, which indicates great slippages.

We continue with data visualization. We generate six graphs using the following functions to make us a better understanding of each stock. In this part, we take the close

price of stock 1 as the target data.

```
par(mfrow=c(2,3))
plot(StockDates, Stock$Close, type="l", xlab="Time", ylab="Close Price", main="Orig
inal")
boxplot(Stock$Close, main="Boxplot")
library(vioplot)
vioplot(Stock$Close, col="gold", main="Violin plot")
hist(Stock$Close, main="Histogram")
library(MASS)
truehist(Stock$Close, main="Scaled/true histogram and kernel density")
lines(density(Stock$Close),col="red", lwd=3)
library(car)
qqPlot(Stock$Close, main="Quantile-quantile plot")
```



The above six graphs indicate the quantile of the stock data. We introduce quantile following with its definition: Let $0<p<1$, then the pth quantile of a random variable X is defined as any real value x satisfying both

$$P(X \geq x) \geq 1-p \text{ and } P(X \leq x) \geq p$$

Note that we call 0.5 quantile the median. We can view the median as the midpoint of the distribution.

The first graph is the line chart of stock 1, which is convenient for us to compare with other following graphs.

The second graph “box plot” graphically displays the quantile distribution of this stock. The bounds of the box represent the first sample quartile (Q1, 25% quantile) and the third sample quartile (Q3, 75% quantile), indicating that half of the close prices fall into

the interquartile range (IQR) 17 to 22. The line inside the box represents the sample median (50% quantile), which is 19.8 in this scenario. Points above the upper horizontal line are observed to be outliers. In this scenario, this can be interpreted that when the stock closing price rises above 27, The company is in a special boom period and is not "normal" from an overall-period perspective.

The third graph "violin plot" is similar to the box plot and displays the same outcome. In addition, it shows the distribution of the data more intuitively with the yellow area. The stock price is more likely to fall into the index with greater width.

The fourth graph "histogram" indicates the number of times an observation (The stock's close price at a particular date) falls into each of the bins. We can see that majority of close price falls into the range 28-30 and 30-32, which is constant with the IQR we calculated previously. The fifth graph approximates a probability density function base on the histogram.

The last graph "quantile-quantile plot" plots the quantiles of two distributions against each other. In this scenario, we compare the sample quantiles of the original stock price with an estimated normal distribution. If the two distributions are similar, we expect to see a straight line through $y=x$ in this plot. As the blue-colored area indicates the difference and occurs around $y=x$ in this graph, we can draw a conclusion that this stock price distribution is not similar to a normal distribution. The points outside the blue-colored area indicate outliers.

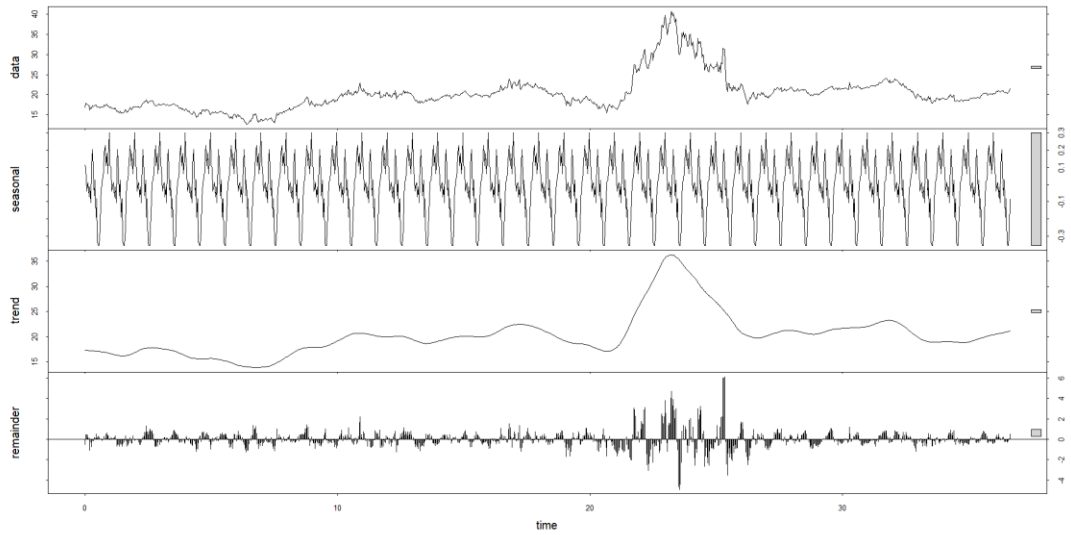
Part 2 - The classical decomposition model

The classical decomposition model divides the data into three compositions, defined as

$$X_t = m_t + s_t + Y_t$$

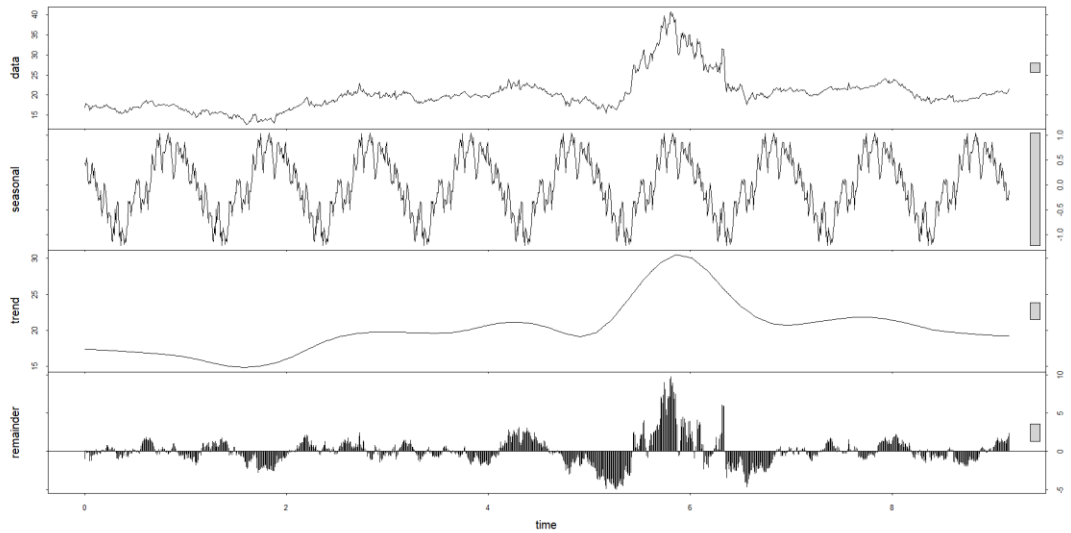
where m_t is the trend component (slowly changing function), s_t is the seasonal component (periodic function) and Y_t is a stationary stochastic process. Through this model, we can clearly display the overall trend and the fluctuation of this stock. Taking stock 1 as example, we set the seasonal period as 30 days, and get the following results.

```
StockTS <- ts(Stock$Open, start=0, frequency=30)
plot(stl(StockTS, s.window="periodic"))
```



The first graph part of the graph is the original stock data, which is made up of the following seasonal component, trend and fluctuation. We can conclude from the trend and remainder (fluctuation) section that the stock price is generally stable except during the special boom period. When the stock price rises and fall significantly, its fluctuation increases correspondingly and leads to those major outliers mentioned above.

Note that in the above decomposition analysis we have set the seasonal period as 30 days. If we set the period longer, we can observe a smoother overall trend. The following graph is the result of 120-day seasonal period.



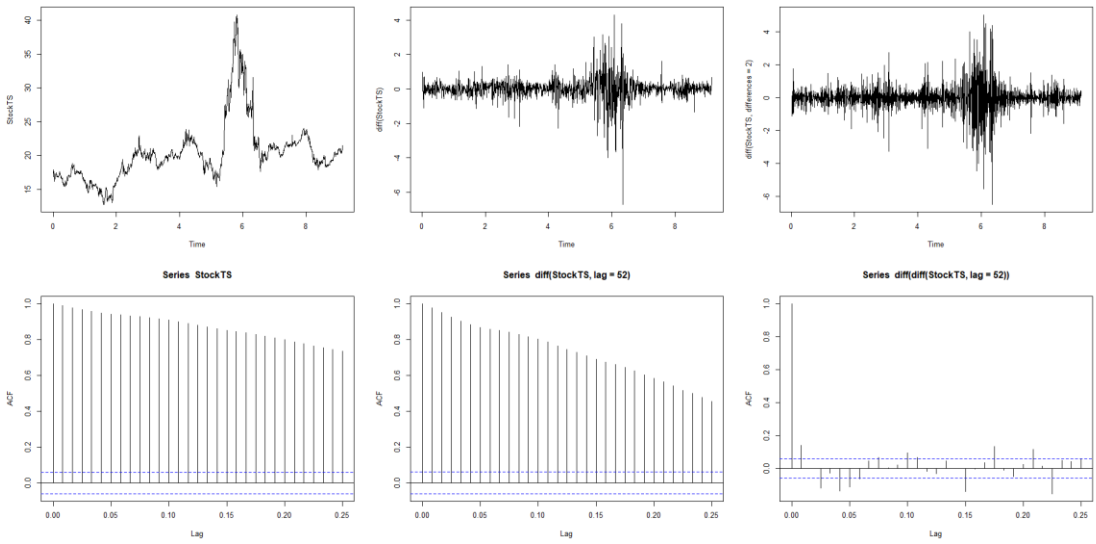
\

Part 3 – Autoregression or not?

We want to remove the seasonal component and the trend of the stock and analyze if it has autoregression characteristics. In order to achieve that, we difference the stock data to generate a stationary time series. Through empirical observation, we notice that the second differential of the stock data is generally similar to its fluctuation. Thus, we make the following graph (Again, taking stock 1 as example):

```
par(mfrow=c(2,3))
plot(StockTS) #The original
plot(diff(StockTS), type="l", pch=19) #The first order difference
plot(diff(StockTS, differences=2), type="l", pch=19) #The second order difference

#Plot the corresponding empirical autocorrelation functions
acf(StockTS)
acf(diff(StockTS, lag=52))
acf(diff (diff(StockTS, lag=52)))
```



The two graphs on the right side are the stock's fluctuation display and its ACF diagram. The blue dotted line represents the 0.95 confidence interval. The acf index does not always fall into the 0.95 confidence interval area even if we remove lag0. Thus, we can draw the conclusion that the stock series have the characteristic of autoregression, which means that there exists a linear relationship between a current close price and its past close price in stock 1.

Part 4 – Cross Comparison Between Stocks

The above three parts focus on the features and characteristics of one particular stock

with R language. In order to better compare those ten data, we use Python to conduct even more data analysis.

```
import pandas as pd
import os
import numpy as np
np.set_printoptions(suppress=True)

dfs = {}
for file in os.listdir():
    split_name = file.split(".")
    if  split_name[-1] == "csv":
        dfs[int(split_name[0])] = pd.read_csv(file, index_col ='Index')
        dfs[int(split_name[0])].index = pd.to_datetime(dfs[int(split_name[0])].inde
                                        x)

open_to_open = {}
close_to_close = {}
open_to_close = {}
high_to_low = {}
for name in dfs:
    open_to_open[name] = np.log(dfs[name]["Open"]).diff().shift(-1)
    close_to_close[name] = np.log(dfs[name]["Close"]).diff().shift(-1)
    open_to_close[name] = np.log(dfs[name]["Open"]) - np.log(dfs[name]["Close"])
    high_to_low[name] = np.log(dfs[name]["High"]) - np.log(dfs[name]["Low"])
open_to_open = pd.concat(open_to_open, axis=1)
close_to_close = pd.concat(close_to_close, axis=1)
open_to_close = pd.concat(open_to_close, axis=1)
high_to_low = pd.concat(high_to_low, axis=1)
return_list = [
    "open_to_open",
   "close_to_close",
    "open_to_close",
    "high_to_low"
]

for return_ in return_list:
    print(return_, "Data Analysis")
    print(locals()[return_].describe()[1:].T, "\n")

other_info = pd.concat([
    open_to_open.skew(),
    open_to_open.kurt()
], axis=1)
```

```
other_info.columns = ["skew", "kurt"]
other_info
```

We get the following output:

open_to_open Data Analysis:

	mean	std	min	25%	50%	75%	max
1	0.000220	0.026972	-0.241745	-0.013458	0.000919	0.013596	0.135427
2	0.000116	0.027669	-0.265703	-0.012202	0.000000	0.013615	0.150823
3	-0.000101	0.017751	-0.120953	-0.009527	0.000000	0.009431	0.099018
4	-0.000009	0.016194	-0.115143	-0.009069	-0.000844	0.008307	0.116433
5	-0.000176	0.018291	-0.096515	-0.008249	0.000000	0.008193	0.134442
6	0.000071	0.003032	-0.014997	-0.001546	0.000344	0.001781	0.016323
7	-0.000055	0.007012	-0.041964	-0.003590	-0.000392	0.003531	0.033572
8	-0.000307	0.020870	-0.145016	-0.010480	0.000000	0.010861	0.091937
9	-0.000281	0.013734	-0.085507	-0.008047	0.000752	0.008368	0.059285
10	-0.000021	0.007508	-0.056541	-0.002825	0.000590	0.003675	0.032374

close_to_close Data Analysis

	mean	std	min	25%	50%	75%	max
1	0.000198	0.029452	-0.400478	-0.012132	0.000765	0.012462	0.135724
2	0.000111	0.029703	-0.390942	-0.010572	0.002086	0.013296	0.139942
3	-0.000104	0.016854	-0.123481	-0.008226	0.000392	0.008395	0.072225
4	-0.000008	0.015147	-0.073338	-0.008717	0.000502	0.007673	0.064143
5	-0.000179	0.019207	-0.135403	-0.007468	0.000277	0.006989	0.238970
6	0.000070	0.002888	-0.017042	-0.001279	0.000360	0.001652	0.019916
7	-0.000055	0.007083	-0.041325	-0.003808	-0.000398	0.003603	0.039457
8	-0.000315	0.020250	-0.110105	-0.010047	0.000000	0.010342	0.084023
9	-0.000275	0.013152	-0.056242	-0.006312	0.000000	0.007061	0.054067
10	-0.000026	0.007081	-0.060455	-0.002716	0.000241	0.003129	0.027419

open_to_close Data Analysis

	mean	std	min	25%	50%	75%	max
1	-0.000030	0.020229	-0.091961	-0.010746	0.000000	0.010019	0.142581
2	-0.000316	0.019153	-0.098559	-0.010513	-0.000863	0.009469	0.144606
3	-0.000047	0.012913	-0.044171	-0.007587	0.000000	0.007213	0.067538
4	0.000196	0.013107	-0.050644	-0.007186	0.000000	0.007959	0.075645
5	-0.000729	0.014495	-0.163320	-0.007139	-0.000760	0.005696	0.055296
6	-0.000052	0.001918	-0.007651	-0.001145	-0.000114	0.000919	0.010324
7	-0.000036	0.004660	-0.028346	-0.002359	0.000125	0.002164	0.036809
8	-0.000328	0.015448	-0.079224	-0.008106	0.000000	0.007737	0.072278
9	-0.000308	0.010298	-0.058522	-0.005366	-0.000389	0.004999	0.041409
10	-0.000448	0.005020	-0.024321	-0.002791	-0.000306	0.002032	0.027331

high_to_low Data Analysis

	mean	std	min	25%	50%	75%	max
1	0.025759	0.020300	0.000000	0.014125	0.020687	0.030595	0.283390
2	0.025692	0.017544	0.004545	0.015524	0.021487	0.030807	0.213284
3	0.016863	0.009697	0.000000	0.010256	0.014405	0.021243	0.069388
4	0.017240	0.009224	0.000000	0.011126	0.014922	0.020859	0.080043
5	0.017378	0.013078	0.002404	0.009188	0.013760	0.021350	0.171946
6	0.002927	0.001568	0.000576	0.001796	0.002572	0.003601	0.013109
7	0.006389	0.004222	0.001008	0.003768	0.005297	0.007694	0.055140
8	0.019692	0.012611	0.000000	0.010979	0.017196	0.024944	0.088850
9	0.014451	0.008961	0.000000	0.008296	0.012756	0.018758	0.063893
10	0.006672	0.004102	0.000000	0.003932	0.005673	0.008279	0.033183

The results show that stock no.1, 2 and 8 have the greatest fluctuation (risk), with the close price standard deviation 0.027, 0.028 and 0.014 correspondingly; Stock no.6, 7 and 10 have the smallest fluctuation (risk) with the close price standard deviation 0.003, 0.007 and 0.008 correspondingly.

In addition, stock no.1, no.2 and no.6 have gained positive avenues in the whole observed historical period. Stock no.4 and 8 have positive skewness. The kurt value of stock no.2 is significantly high, indicating that this stock price has experienced a very sharp rise and then a very sharp fall in a short period.