FAANG Analysis

Yu-Hao, Hsu
January 17, 2021

Contents

Chapter 1

Risks of FAANG

1.1 Introduction

Risk is the uncertainty that an event happens or not. There are many factors that can affect the risks of an investment: market risk, interest risk, liquidity risk, credit risk,...etc. This chapter, however, will not going to talk about these factors. Instead, 'how' to evaluate them quantitatively will be shown.

Before discussion begins, knowing the data that will be used is important. First, the stock that have been chosen were Facebook, Apple, Amazon, Netflix, and Google (the top five popular company in technology). The data were collected by yfinance and save in a dataframe. Moreover, rate of returns were included as well.

```python
fb = yf.Ticker('FB').history(period='max').Close
az = yf.Ticker('AMZN').history(period='max').Close
ap = yf.Ticker('AAPL').history(period='max').Close
nf = yf.Ticker('NFLX').history(period='max').Close
gg = yf.Ticker('GOOGL').history(period='max').Close
data = pd.DataFrame({'FB':fb,'AMZN':az,'AAPL':ap,'NFLX':nf,'GOOGL':gg}).dropna()
data['FB_Ret'] = data['FB'].pct_change()
data['AMZN_Ret'] = data['AMZN'].pct_change()
data['AAPL_Ret'] = data['AAPL'].pct_change()
data['NFLX_Ret'] = data['NFLX'].pct_change()
data['GOOGL_Ret'] = data['GOOGL'].pct_change()
data
```

	FB	AMZN	AAPL	NFLX	GOOGL	FB_Ret	AMZN_Ret	AAPL_Ret	NFLX_Ret	GOOGL_Ret
Date										
2012-05-18	38.23	213.85	16.37	9.99	300.50	NaN	NaN	NaN	NaN	NaN
2012-05-21	34.03	218.11	17.32	10.25	307.36	-0.109861	0.019921	0.058033	0.026026	0.022829
2012-05-22	31.00	215.33	17.19	9.67	300.70	-0.089039	-0.012746	-0.007506	-0.056585	-0.021668
2012-05-23	32.00	217.28	17.61	10.27	305.04	0.032258	0.009056	0.024433	0.062048	0.014433
2012-05-24	33.03	215.24	17.45	10.04	302.13	0.032188	-0.009389	-0.009086	-0.022395	-0.009540
...	...	...	...	...	...	...	...	...	...	...
2020-10-05	264.65	3199.20	116.50	520.65	1482.83	0.018120	0.023744	0.030791	0.034966	0.018707
2020-10-06	258.66	3099.96	113.16	505.87	1451.02	-0.022634	-0.031020	-0.028670	-0.028388	-0.021452
2020-10-07	258.12	3195.69	115.08	534.66	1459.14	-0.002088	0.030881	0.016967	0.056912	0.005596
2020-10-08	263.76	3190.55	114.97	531.79	1483.43	0.021850	-0.001608	-0.000956	-0.005368	0.016647
2020-10-09	264.45	3286.65	116.97	539.44	1510.45	0.002616	0.030120	0.017396	0.014385	0.018215

2113 rows × 10 columns

Figure 1.1: Close Price & Rate of Return

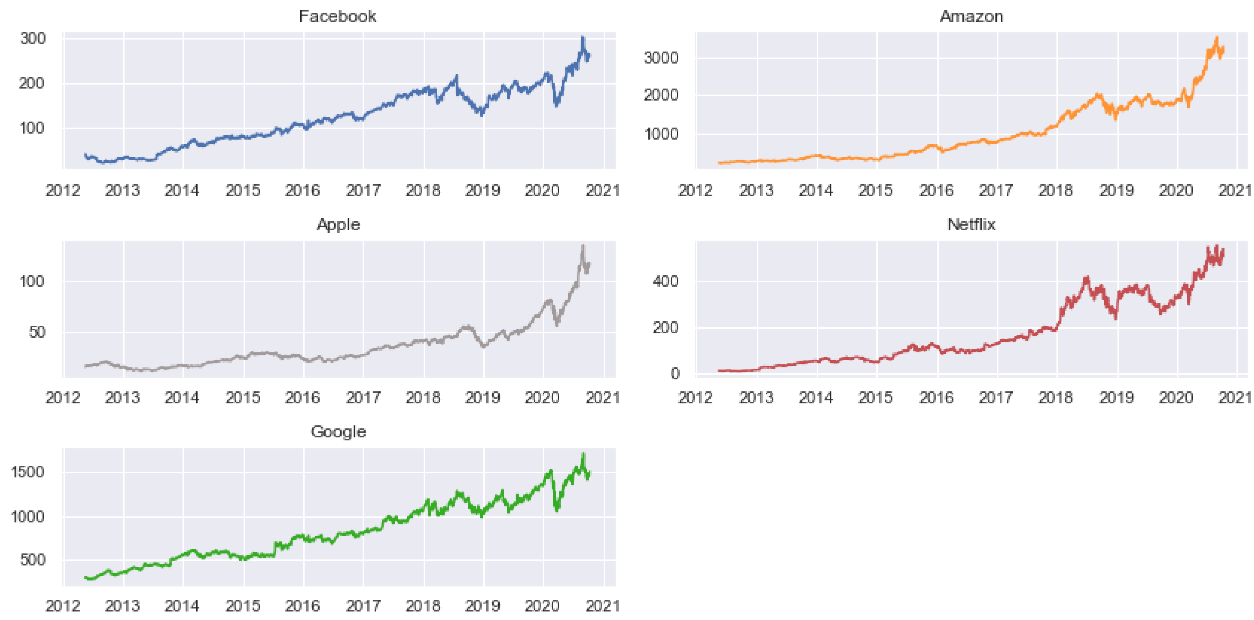

Figure 1.2: Stock Trends

One Term Simple Rate of Return:

$$R_t(1) = \frac{P_t - P_{t-1}}{P_{t-1}} = \frac{P_t}{P_{t-1}} - 1 \tag{1.1}$$

Multiple Term Simple Rate of Return:

$$R_t(k) = \frac{P_t - P_{t-k}}{P_{t-k}} = \frac{P_t}{P_{t-k}} - 1 \tag{1.2}$$

Relation Between One Term & Multiple Term:

$$R_t(n) = \prod_{j-0}^{k-1} (1 + R_{t-j}) - 1 \tag{1.3}$$

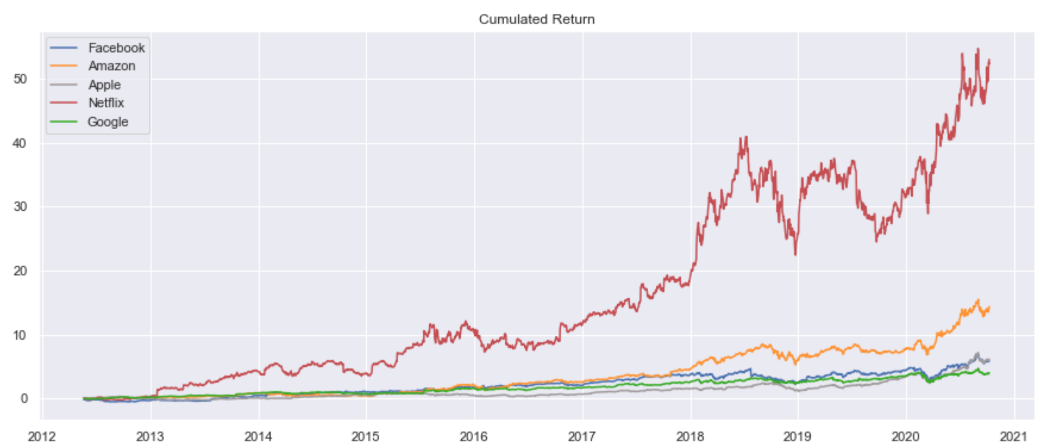

Figure 1.3: Cumulative Returns

1.1.1 Summary

Facebook has grown 6.86 times higher
Amazon has grown 14.83 times higher
Apple has grown 7.05 times higher
Netflix has grown 51.77 times higher
Google has grown 4.92 times higher
$\Rightarrow$ Rank growth from high to low: Netflix > Amazon > Apple > Facebook > Google

1.2 Standard Deviation

1.2.1 STD

$$\sigma^2 = E\{[R - E(R)]^2\} \tag{1.4}$$

Strength:

1. Easy to understand $\rightarrow$ The distribution of the data

Weakness:

1. Not perfectly match with the meaning of "Risk": Risk consider only the downward movement of the returns; upward movement makes profits so should be excluded.

2. Most of the financial instruments are not normally distributed. Wrong decisions may be done if only base on STD.

1.2.2 Summary

Generally speaking, we can first look at STD to have a basic understanding on the data.
NFLX_Ret$\rightarrow$0.030264
FB_Ret$\rightarrow$ 0.023560
AMZN_Ret$\rightarrow$0.019097
AAPL_Ret$\rightarrow$0.017917
GOOGL_Ret$\rightarrow$0.015897
$\Rightarrow$ We can make a hypothesis that the risk of Netflix > Facebook > Amazon > Apple > Google

1.3 Downside Deviation

Downside deviation is similar to the concept of STD. However, downside deviation evaluates only the distributions that are below the Minimum Acceptable Rate of Return (MARR), where MARR can be Risk free return, zero, or average returns.

$$\sigma(R, MARR) = \sqrt{E\{[min(R - MARR, 0)]\}} \tag{1.5}$$

$$\Rightarrow \sigma(R, MARR) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\{[min(R - MARR, 0)]\}} \text{ ,where T=\#of samples} \tag{1.6}$$

```python
#Choose Avg.Return as MARR(Minimum Acceptable Rate of Return)
def downsideRisk(stock_return):
    mu = stock_return.mean()
    temp = stock_return[stock_ return<mu]
    downsideRisk = ((sum((mu-temp)**2))/len(stock_return))**0.5
    return(downsideRisk)
```

Downside deviation is better than STD in the way that downside deviation better match the concept of risk, and it can support decisions when the data is not normally distributed.

1.4 Value at Risk (VaR)

Generally speaking: Var = The expected maximum loss under a level of confidence

$$Pr\{X < -VaR(\alpha, \Delta t)\} = \alpha\% \tag{1.7}$$

ex) Under 95% confidence, the maximum loss of next month will be 10%.

However, the weakness of VaR is that the meaning is misleading: maximum possible loss can't be evaluate by simply looking at VaR. 95% percent VAR really means that in 5% of cases (ex. 2-3 days in a year with daily VaR) the loss is expected to be greater than the VaR amount.

There are three ways to calculate VaR:

1. Historical Stimulation Approach: Easily calculated, just find the lowest return that is under the confidence interval(no need to find the distributions or other statistical estimators).
   However, weakness is that history may not repeat. Also, amount of data is an important factor. If number of data $<$ 1500, it may be considered as too low. On the other hand, too much data may include outdated information, which can cause error.

```
            # Take 95% confidence level as example
            VaR = data_history_return.quantile(0.05)

```

2. Variance-Covariance Approach: Has similar concept as Historical Stimulation Approach but different in the way that Variance-Covariance includes the calculation of data distributions. However, the weakness is same as Historical Stimulation Approach, where history may not recur. Also, it assumes that the data is normally distributed, where most financial instruments doesn't.

```
            # Take 95% confidence level as example
            from scipy.stats import norm
            VaR = norm.ppf(0.05,data_history_return.mean(),data_history_return.std())

```

3. Monte Carlo Simulation: A method of taking large amount of randomized sampling to simulate different situations. Monte Carlo will be a better choice if the model can be construct accurate and precisely. However, the simulations take large calculations, and the simulations were from pseudo random number, which may cause larger error.

1.5 Expected Shortfall (ES)

Expected Shortfall is simply an extension of VaR: It takes the average value under VaR (or literally the expected shortfall).

$$ES = E[X|_{X \leq -VaR}] \tag{1.8}$$

```
    data[data<=data.quantile(0.05)].mean()
```

1.6 Maximum Drawdown (MDD)

1.6.1 MDD

Drawdown is the loss from the maximum P_t back to P_T at time T during (0,T).

$$D(T) = max\{0, max_{t \in (0,T)} P_t - P_T\} \tag{1.9}$$

corresponding Drawdown Percentage will be:

$$d(T) = \frac{D(T)}{max_{t \in (0,T)} P_t} \tag{1.10}$$

Maximum Drawdown is the highest drawdown during (0,T). For example, when an investor buys a stock at price X at t=0 but the price begins to drop afterward. At time t, price once reached its lowest at price Y and never goes back to price X. The MDD will then be X-Y, or the maximum loss. In addition, MDD is especially useful when evaluating chasing stocks.

1.6.2 Caluclate MDD With Rate of Returns

If we invest 1 dollar on a stock and hold till time T, our 1 dollar will become $(1+R_1)$, $(1+R_1)(1+R_2)$, $(1+R_1)(1+R_2)(1+R_3)$,...,$\Pi_{k=1}^{T}(1+R_k)$[1]:

$$D(T) = max\{0, max_{t\in(0,T)}\Pi_{k=1}^{t}(1+R_k) - \Pi_{k=1}^{T}(1+R_k)\} \tag{1.11}$$

Corresponding Drawdown Percentage will be:

$$d(T) = \frac{D(T)}{max_{t\in(0,T)}\Pi_{k=1}^{t}(1+R_k)} \tag{1.12}$$

```
    def maxDrawdown(stock_return):
        value = (1+stock_return).cumprod()
        downValue = value.cummax()-value
        downPercentage = downValue/(downValue+value)
        MDD = downValue.max()
        mddPercentage = downPercentage.max()
        return(MDD,mddPercentage)
```

1.7 Conclusion

Figure 1.4: FAANG Risks

When we first look at the STD of each stock, we made a hypothesis that the risk of Netflix > Facebook > Amazon > Apple > Google. To support the hypothesis, Downside Deviation, Value at Risk (VaR), Expected Shortfall (ES), and Maximum Drawdown (MDD) were examined. By looking at the graphs, we can tell that the Downside Risk, VaR, and ES have all given out same conclusion as the hypothesis: risk of Netflix > Facebook > Amazon > Apple > Google. On the other hand, when we examine the MDD, we find out that the MDD of Facebook > Netflix > Apple > Amazon > Google. This gives us an understanding that, when extreme event happens, Facebook can brings a maximum loss to investor (while Netflix has the highest risks). In addition, since Google has the lowest risks and minimum MDD, Google will be a stabler investment in FAANG.

[1]Review Section 3:Trend and Return

Chapter 2

Capital Asset Pricing Model (CAPM)

2.1 Introduction

Risks can be categorized as systematic, and unsystematic risk. Systematic risk is the one that can affect the entire market, which is beyond the control of an individual and can't be diversified. For example, macroeconomic risks such as inflation, political crisis, and natural disasters are consider as unsystematic risks. On the other hand, systematic risk is the risk that can be prevented, or can be diversified through asset allocation. For example, risks that only affect individual firm or sector, such as failure of product development, and the drop of real estates during recession, respectively.

Since systematic risk can't be diversified, it is required for a risk premium. Therefore, there should be a relation between the risk premium and the return of an asset. CAPM is the model that build under this concept. In this chapter, CAPM will be discussed and application on FAANG will be shown.

2.2 CAPM

Commonly seen CAPM:

$$E(R_i) = R_f + \beta[E(R_m) - R_f] \tag{2.1}$$

where:

$$\begin{aligned}
&E(R_i) &&= \text{Expected return of asset i}\\
&R_f &&= \text{Risk free rate}\\
&E(R_m) &&= \text{Expected return of market}\\
&E(R_m) - R_f &&= \text{the market risk premium}
\end{aligned}$$

βmeasures the move of an asset with respect the the market:

$$\beta_i = \frac{\sigma(R_i, R_m)}{\sigma^2(R_m)} \tag{2.2}$$

- When R_i=R_m, $|\beta| = 1$, which means that the movement of the asset return is larger that the market return.
- When $|\beta| < 1$, it means that the movement of the asset return is smaller that the market return.
- When $|\beta| > 1$, it means that the movement of the asset return is larger that the market return.

2.3 Jensen's Alpha

In 1968, Michael Jensen raised am evaluation method of mutual fund manager: Jensen's Alpha[1].

$$\alpha_i = R_i - E[R_i] \tag{2.3}$$

$$\rightarrow \alpha_i = R_i - [R_f + \beta(E(R_m) - R_f)] \tag{2.4}$$

$$\Rightarrow R_i - R_f = \alpha_i + \beta_i(E(R_m) - R_f) \tag{2.5}$$

where α is the abnormal return that an asset compare to the expected return computed by CAPM.

[1]Jensen Michael C. "The Performance of Mutual Funds in the Period 1945-1964." The Journal of Finance 23.2 (1968): 389-416

2.4 FAANG CAPM

```python

import statsmodels.api as sm
#Add NASDAQ Data for Market Return
nasdaq = yf.Ticker('^IXIC').history(period='max').Close
data['NASDAQ'] = nasdaq
data['NASDAQ_Ret'] = nasdaq.pct_change()
def capm(stock_name,color):
    #risk-free rate = 1 YearTreasury Yield Curve Rates (at 2020/10/9)=  0.15%
    # Convert 1 Year Rate to Daily Rate
    rf = (1+0.0015)**(1/360)-1

    #Prepare DataFrame
    stockRet = data[stock_name] - rf
    marketRet = data['NASDAQ_Ret'] - rf
    excessReturn = pd.concat([stockRet,marketRet],axis=1)
    excessReturn.dropna(inplace=True)

    #OLS
    model = sm.OLS(excessReturn.iloc[:,0],sm.add_constant(excessReturn.iloc[:,1]))
    result = model.fit()
    result.summary()
```

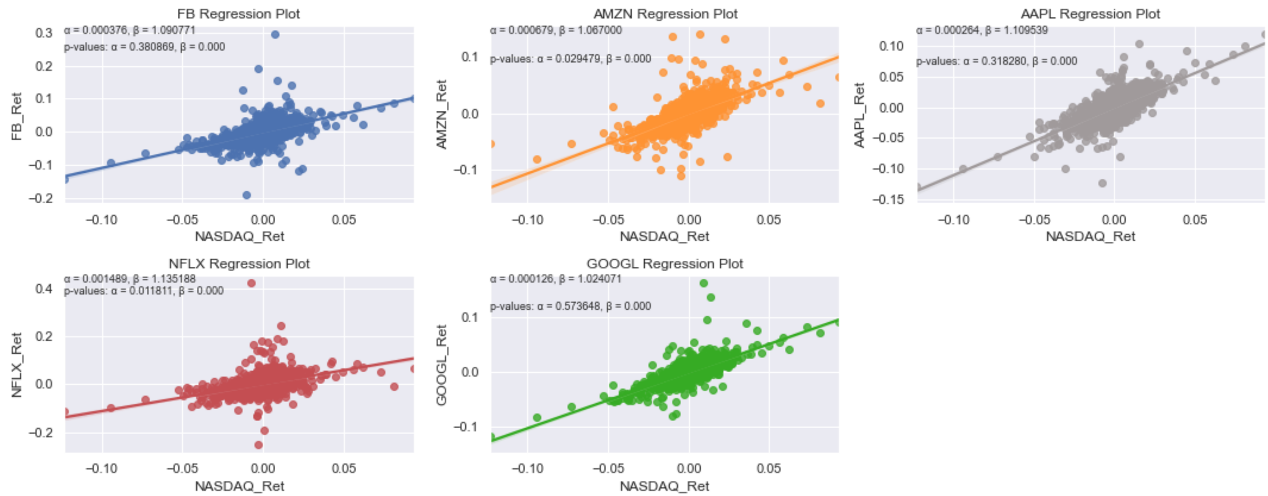

Figure 2.1: FAANG CAPM

- Facebook

$$R_{FB} - R_f = 0.000376 + 1.090771(R_{NASDAQ} - R_f) \tag{2.6}$$

α_{FB} have a p-value[2] of 0.38 and β_{FB} have a p-value of 0. As a result, under 95% level of confidence, there is no excess return for Facebook. In addition, the β for Facebook is 1.09, which tells that Facebook moves with the NASDAQ market.

- Amazon

$$R_{AMZN} - R_f = 0.000679 + 1.067(R_{NASDAQ} - R_f) \tag{2.7}$$

α_{AMZN} have a p-value of 0.029 and β_{FB} have a p-value of 0. As a result, under 95% level of confidence, there is an excess return of 0.068% for Amazon. In addition, the β for Amazon is 1.067, which tells that Amazon moves with the NASDAQ market.

- Apple

$$R_{AAPL} - R_f = 0.000264 + 1.109539(R_{NASDAQ} - R_f) \tag{2.8}$$

α_{AAPL} have a p-value of 0.32 and β_{AAPL} have a p-value of 0. As a result, under 95% level of confidence, there is no excess return for Apple. In addition, the β for Apple is 1.109, which tells that Apple moves with the NASDAQ market.

[2]When null hypothesis is true, p-value is the maximum probability that an observed result(or more extreme) happens.

- Netflix

$$R_{NFLX} - R_f = 0.001489 + 1.135188(R_{NASDAQ} - R_f) \tag{2.9}$$

α_{NFLX} have a p-value of 0.012 and β_{NFLX} have a p-value of 0. As a result, under 95% level of confidence, there is an excess return of 0.15% for Netflix. In addition, the β for Netflix is 1.135, which tells that Netflix moves with the NASDAQ market.

- Google

$$R_{GOOGL} - R_f = 0.000126 + 1.024071(R_{NASDAQ} - R_f) \tag{2.10}$$

α_{GOOGL} have a p-value of 0.57 and β_{FB} have a p-value of 0. As a result, under 95% level of confidence, there is no excess return for Google. In addition, the β for Google is 1.024, which tells that Google moves with the NASDAQ market.

Chapter 3

Fama-French Three Factor Model

3.1 Introduction

Even though CAPM gives an nice relation between risk premium and expected return of an asset, but risk-premium by itself can't explain the expected return completely. In 1993, Fama and French adds market equity and book-to-market-equity ratio (B/M) as additional two factors for the explanation of expected return. The reason for market equity as a factor can refer to the research of Banz's in 1981, where he found out that a firm with smaller market equity will have better performance than the market, and the reason for book-to-market-equity ratio as a factor can refer to the research of Stattman, where he found out that firm with higher B/M ratio tends to have higher peroformance than the market.

3.2 Fama-French Three Factor Model

The Fama-French Three Factor Model:

$$E(R_{it}) - R_{ft} = \beta_i[E(R_{mt} - R_{ft})] + s_i E[SMB_t] + h_i E[HML_t] \tag{3.1}$$

where:

SMB_t = the excess return of small firms (the return of small firms - the return of large firms)
s_i = the corresponding movement of asset i with respect to the SMB factor
HML_t = the excess return of High B/M ratio (the return of high B/M ratio firms - the return of low B/M ratio firms)
h_i = the corresponding movement of asset i with respect to the HML factor

Like Jensen's Alpha, its also useful to write as:

$$E(R_{it}) - R_{ft} = \alpha_i + \beta_i(R_{mt} - R_{ft}) + s_i SMB_t + h_i HML_t \tag{3.2}$$

3.3 SMB & HML

1. Find market value of the firms in the market at time t

$$MV_{it} = P_{it} \times Q_{it} \tag{3.3}$$

where:

P_{it} = Price of firm i at time t
Q_{it} = Outstanding share of firm i at time t

2. Calculate the median of the market firms
⇒ Firms that have MV larger than the median will be big firms and the one that's lower will be the small firms

3. Find Book-to-Market Ratio of the firms in the market at time t

$$BM_{it} = \frac{BV_{it}}{MV_{it}} \tag{3.4}$$

where:

BV_{it} = Book Value of firm i at time t
MV_{it} = Market Equity of firm i at time t

4. Group the firms low, medium, and high BM groups[1]
$\Rightarrow$ Firms that have low BM (the lowest 30% of the market) will be Low; Firms that have high BM (highest 30% of the market) will be High; Firms that have medium BM (rest of the market(40%)) will be Medium.
5. There will be 6 combinations of the firms: SL,SM,SH,BL,BM,BH. For example, SL will be the small firms with low BM.
6. Small Minus Big Factor (the excess return of small firms from the big firms)
$$SMB = \frac{1}{3}(SL_t + SM_t + SH_t) - \frac{1}{3}(BL_t + BM_t + BH_t) \tag{3.5}$$
7. High Minus Low Factor (the excess return of high BM firms from the low BM firms)
$$HML = \frac{1}{2}(SH_t + BH_t) - \frac{1}{2}(SL_t + BL_t) \tag{3.6}$$

Note: Even though the SMB and HML is complicate to calculate, but it can be obtained from the data companies.

3.4 FAANG Fama-French

Date	FB_Ret	AMZN_Ret	AAPL_Ret	NFLX_Ret	GOOGL_Ret	NASDAQ_Ret	SMB	HML	RF
2012-05-21	-0.109861	0.019921	0.058033	0.026026	0.022829	0.024622	0.67	-1.18	0.0
2012-05-22	-0.089039	-0.012746	-0.007506	-0.056585	-0.021668	-0.002855	-0.87	0.28	0.0
2012-05-23	0.032258	0.009056	0.024433	0.062048	0.014433	0.003889	0.49	-0.27	0.0
2012-05-24	0.032188	-0.009389	-0.009086	-0.022395	-0.009540	-0.003768	-0.01	0.21	0.0
2012-05-25	-0.033909	-0.010918	-0.005731	-0.000996	-0.020091	-0.000652	0.23	-0.10	0.0
...	...	...	...	...	...	...	...	...	...
2020-07-27	0.012093	0.015388	0.023688	0.031637	0.014070	0.016702	0.75	-1.93	0.0
2020-07-28	-0.014475	-0.017963	-0.016484	-0.014405	-0.016856	-0.012735	-0.72	1.01	0.0
2020-07-29	0.013775	0.011065	0.019231	-0.008250	0.013208	0.013541	0.54	0.84	0.0
2020-07-30	0.005187	0.006049	0.012122	0.002725	0.009754	0.004256	0.47	-1.81	0.0
2020-07-31	0.081748	0.036961	0.104666	0.006340	-0.032775	0.014872	-1.54	-0.65	0.0

2063 rows × 9 columns

Figure 3.1: FAANG French Fama Data

```python
def famafrench(stock_name):
    #Prepare DataFrame
    stockRet = data[stock_name]-data['RF']
    stockRet.name = 'stockRet'
    marketRet = data['NASDAQ_Ret']-data['RF']
    marketRet.name = 'marketRet'
    SMB = data['SMB']
    SMB.name='SMB'
    HML = data['HML']
    HML.name='HML'
    regressData= pd.concat([stockRet,marketRet,SMB,HML],axis=1)
    regressData.dropna(inplace=True)
    #Regress
    regress = sm.OLS(regressData['stockRet'],sm.add_constant(regressData.iloc[:,1:]))
    result = regress.fit()
```

[1] Low BM groups is also known as growth stock; High BM groups is also known as value stock

1. FB:

$$R_{FB} - R_f = 0.000569 + 1.093153(R_{NASDAQ} - R_f) - 0.000410SMB - 0.002983HML \tag{3.7}$$

- α=0.000569, β=1.093153, SMB coef.=-0.000410, HML coef.=-0.002983
- Corresponding p-value: α=0.193163, β=0.000000, SMB coef.=0.614319, HML coef.=0.000004

2. AMZN:

$$R_{AMZN} - R_f = 0.000753 + 1.069962(R_{NASDAQ} - R_f) - 0.002449SMB - 0.005416HML \tag{3.8}$$

- α=0.000753, β=1.069962, SMB coef.=-0.002449, HML coef.=-0.005416
- Corresponding p-value: α=0.014592, β=0.000000, SMB coef.=0.000020, HML coef.=0.000000

3. AAPL:

$$R_{AAPL} - R_f = 0.000494 + 1.111724(R_{NASDAQ} - R_f) - 0.003837SMB - 0.001192HML \tag{3.9}$$

- α=0.000494, β=1.111724, SMB coef.=-0.003837, HML coef.=-0.001192
- Corresponding p-value: α=0.061862, β=0.000000, SMB coef.=0.000000, HML coef.=0.002337

4. NFLX:

$$R_{NFLX} - R_f = 0.001685 + 1.109416(R_{NASDAQ} - R_f) + 0.001501SMB - 0.005936HML \tag{3.10}$$

- α=0.001685, β=1.109416, SMB coef.=0.001501, HML coef.=-0.005936
- Corresponding p-value: α=0.004992, β=0.000000, SMB coef.=0.178676, HML coef.=0.000000

5. GOOGL:

$$R_{GOOGL} - R_f = 0.000207 + 1.044771(R_{NASDAQ} - R_f) - 0.002353SMB - 0.001649HML \tag{3.11}$$

- α=0.000207, β=1.044771, SMB coef.=-0.002353, HML coef.=-0.001649
- Corresponding p-value: α=0.359083, β=0.000000, SMB coef.=0.000000, HML coef.=0.000000

In conclusion, it can be seen that, under 95% confidence, Amazon and Netflix will have excess return. Moreover, all of the FAANG are moving positively with the NASDAQ market. In addition, since FAANG stocks were all base on technology[2], they all have low BM; therefore, they move in an inverse direction with the HML factor. Also, the SMB also affects Amazon, Apple, and Google in an negative way, since they were all large firms. Possible reason that the SMB didn't explain well on Facebook and Netflix may be caused by the big drops during 2019[3], where other three firms didn't plunged severely in 2019.

[2]Technology sectors are usually growth stocks.
[3]Look at the trends in CH1.1:Introduction

Chapter 4

Characteristics of Time Series

4.1 Introduction

If historical data is used for forecasting, there are two important characteristics that the time series need to fulfill: Autocorrelation and Stationary. In one hand, Autocorrelation checks for the relation between the past and future. On the other hand, Stationary checks for the continuous of the relation trend. In addition, for the test of autocorrealtion, ADF will be discussed and for the test of stationary, Ljung-Bbox will be introduced.

4.2 Autocorrelation

When using historical data to infer future events, its important that the historical data have some kind of connection with the future. If there's no connection, then there's no evident to prove that the history will reappear. The way to evaluate the "connection" between past and future is through Autocorrelation, where two different time periods in a time series will be evaluate. Further more, Autocorrelation can be present in three different functions: Autocovariance Function, Autocorrelation Coefficient Function (ACF), and Partial Autocorrelation Coefficient Function (PACF).

4.2.1 Autocovariance

Just like Covariance, but the variables are from one time series, where one variable will be X_k and one will be X_{k-l} (l=time lag):

$$\gamma_l = E[(X_k - \mu_k)(X_{k-l} - \mu_{k-l})] = Cov(X_k, X_{k-l}), l = 0, 1, 2... \tag{4.1}$$

Notice when $l = 0$, $\gamma_0 = \mathrm{Var}(X_k)$:

$$\gamma_0 = E[(X_k - \mu_k)(X_k - \mu_k)] = Var(X_k) \tag{4.2}$$

4.2.2 Autocorrelation Coefficient Function (ACF)

Autocorrelation Coefficient acts like β in CAPM, which measures the movement of the future data (lag data) with respect to the past data (lag=0 data):

$$\rho_l = \frac{Cov(X_k, X_{k-l})}{Var(X_k)} = \frac{\gamma_l}{\gamma_0} \tag{4.3}$$

4.2.3 Partial Autocorrelation Function (PACF)

When autocorrelation is calculated and $\rho_l > 0$, all of the influences of the lags will be sum up. For example, assume an asset has price $p_1, p_2, ..., p_t$, when ρ_{t-1} is calculated, ρ_{t-1} captures all of the effect of $\rho_{t-2}, \rho_{t-3}...$ because p_1 affects p_2, and p_2 affects p_3... In order to capture the effect of single term l on future, Partial Autocorrelation Function[1] is introduced, where the effect of other terms will be eliminated:

$$x_t = \rho_{k1}x_{t-1} + \rho_{k2}x_{t-2} + \cdots + \rho_{kk}x_{t-k} + \epsilon_t \tag{4.4}$$

$$\rightarrow x_t - \rho_{k1}x_{t-1} - \rho_{k2}x_{t-2} + \cdots - \rho_{kk-1}x_{t-k+1} = \rho_{kk}x_{t-k} + \epsilon_t \tag{4.5}$$

$$\Rightarrow \rho_{ll} = Corr(X_k, X_{k-l}|X_{k-1}, X_{k-2}, ..., X_{k-l+1}) \tag{4.6}$$

[1]Calculation of PACF involves linear algebra and is pretty complicated. Therefore, is not going to be presented.

4.3 Stationary

Knowing the autocorrelation of time series is the first step. Next step is to understand the stationary of the data. Stationary evaluates that whether or not a historical trend will carry on in the future. If a time series is not stationary, it means that the characteristic of the data keeps changing, in other words, the historical trends won't pass on to the future. Therefore, its important for a time series to be stationary if its used for prediction.

4.3.1 Strictly Stationary

For a time series X_t, for any positive integer n, $t_1, t_2, ..., t_n \in T$, for any integer τ:

$$F_X(x_{t_1}, ..., x_{t_n}) = F_X(x_{t_1+\tau}, ..., x_{t_n+\tau}) \tag{4.7}$$

This implies that the joint probability distribution function $F_X(x_{t_1}, ..., x_{t_n})$ is the same as $F_X(x_{t_1+\tau}, ..., x_{t_n+\tau})$ and is called Strictly Stationary. If a time series fulfills the property of Strictly Stationary, all of its statistics characteristics (e.g. Var, mean,...) will not change.

4.3.2 Weakly Stationary

It's not easy to prove for a Strictly Stationary conditions, so Weakly Stationary is introduced[2]. For Weakly Stationary, only three conditions needs to be fulfilled:

1. Mean is constant

$$E(X_t) = \mu \tag{4.8}$$

2. Variance is finite

$$E(X_t^2) < \infty \tag{4.9}$$

3. Autocovariance γ_l of X_t is independent from t, where γ_l depends only on l

$$\gamma_l(t) = \gamma_l(t+h) \tag{4.10}$$

4.4 Stationary Test

4.4.1 Trend Plot

By the property of the Weakly Stationary, the expectation value of a time series should be constant and variance should be finite at the same time. Therefore, the trend plot of the time series should looked like a oscillation moving up and down around an constant value.

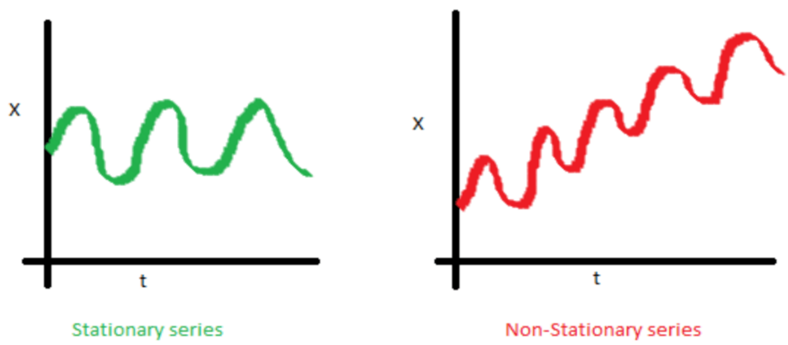

Figure 4.1: Stationary Trends

[2]Compared to Strictly Stationary, Weak Stationary is more practical. Therefore, most of the time, Weakly Stationary is been referred when talking about stationary

4.4.2 ACF & PACF Graph

For a stationary time series, ACF and PACD drops to 0 quickly. On the other hand, if a time series is not stationary, its ACF and PACF will decline slowly instead of dropping quickly.
From:

$$x_t = \rho_{k1}x_{t-1} + \rho_{k2}x_{t-2} + \cdots + \rho_{kk}x_{t-k} + \epsilon_t \tag{4.11}$$

where[3]: $\epsilon_t \sim IID(0, \sigma_\epsilon^2)$ It can be seen that if all the ρ are equal to zero, then $x_t = \epsilon_t$. Therefore, $E(X_t) = 0$ & $Var(X_t) = \sigma_\epsilon^2$ and by the stationary property, x_t is stationary.

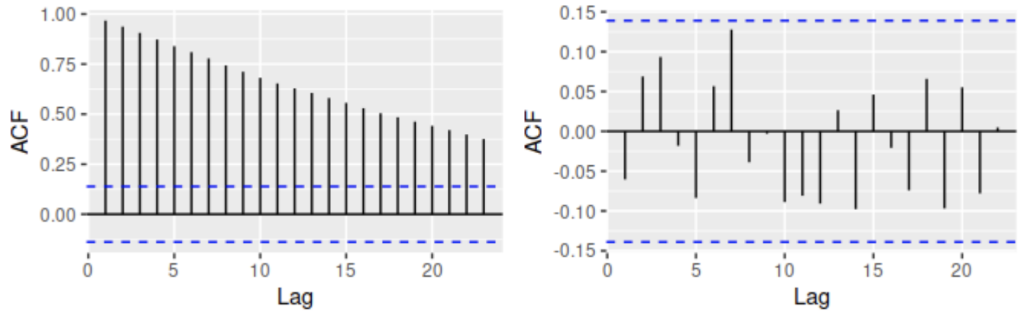

Figure 4.2: Non-Stationary (Left) & Stationary (Right) ACF

4.4.3 Unit Root

While there are stationary trends that can be tell visually, there exists other stationary trends that aren't easy tot tell just by observing at the graph. Therefore, unit root test is introduced.

For a Non-stationary time series, if it can be transformed to a stationary time series after d finite difference, it is called I(d) (I means Integrated).
For example, Random Walk Process:

$$x_t = x_{t-1} + \epsilon_t, x_0 = 0, \quad where\, \epsilon_t \sim IID(0, \sigma_\epsilon^2) \tag{4.12}$$

where

$$\begin{aligned} E(x_t) &= \mathrm{E}(\mathrm{x}_{t-1}) = ... = E(x_1) = E(x_0 + \epsilon_1) = E(0+0) = 0 \\ Var(x_t) &= \mathrm{Var}(\mathrm{x}_{t-1}) + \sigma_\epsilon^2 = ... = Var(x_2) = Var(x_1) + \sigma_\epsilon^2 = Var(x_0) + \sigma_\epsilon^2 + \sigma_\epsilon^2 = t\sigma_\epsilon^2 \end{aligned}$$

However, after 1 finite difference:

$$\Delta x_t = x_t - x_{t-1} = \epsilon_t \tag{4.13}$$

The time series become stationary. Therefore, x_t is a non-stationary I(1) time series.

By using a Lag Operator (L)[4], $Lx_t = x_{t-1}, LLx_t = L^2x_t = Lx_{t-1} = x_{t-2}$, stationary of a time series can be tell with the unit root test:

$$x_t = x_{t-1} + \epsilon_t = Lx_t + \epsilon_t \tag{4.14}$$

If L = 1, it means that $x_t = x_t + \epsilon_t$, which infers that the time series itself x_t contains a random part ϵ_t.

In order to check whether L=1, x_t can be rewrite:

$$x_t = Lx_t + \epsilon_t \rightarrow (1-L)x_t = \epsilon_t \tag{4.15}$$

Therefore, if 1-L=0, or L=1, it means that the time series is Non-stationary (contains the random part ϵ)[5].

To check for unit root, Argumented Dickey-Fuller Test (ADF) can be used:

$$y_t = \alpha + \rho_1 y_{t-1} + \rho_2 y_{t-2} + \cdots + \rho_p y_{t-p} + \epsilon_t,\, where\, \epsilon_t \sim IID(0, \sigma_\epsilon^2) \tag{4.16}$$

$$\Delta y_t = y_t - y_{t-1} = \alpha + \rho_1 y_{t-1} + \rho_2 y_{t-2} + \cdots + \rho_p y_{t-p} + \epsilon_t - y_{t-1} \tag{4.17}$$

[3]IID means Independent and Identically Distributed
[4]The function of L is to "lag" the previous time series and transform it into current time series.
[5]The process of getting L=1 is called the unit test

$$\rightarrow \Delta y_t = \alpha + (\rho_1 - 1)y_{t-1} + \rho_2 y_{t-2} + \cdots + \rho_p y_{t-p} + \epsilon_t \tag{4.18}$$
$$\Rightarrow \Delta y_t = \alpha + \gamma y_{t-1} + \delta y_{t-1} + \epsilon_t \tag{4.19}$$

where:

α = corresponding intercept (acts like the α in the CAPM)
γ = (ρ-1); if γ= 0, it means $\rho = 1$ and unit root exists, so times series is Non-stationary[6].
δy_{t-1} = argumented part of the equation

Last, corresponding time trend affect will be added to the equation, and the the result is the Argumented Dickey-Fuller Test Equation:

$$\Rightarrow \Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta y_{t-1} + \epsilon_t \tag{4.20}$$

4.5 White Noise

White noise is an important type of random process. If a time series is a white noise, it means that there is no autocorrelation and therefore, no predictions could be made from the historical data.
White Noise satisfies three conditions:

1. E(X_t)=0
2. Var(X_t)=σ^2
3. γ_l=0, where $l > 0$

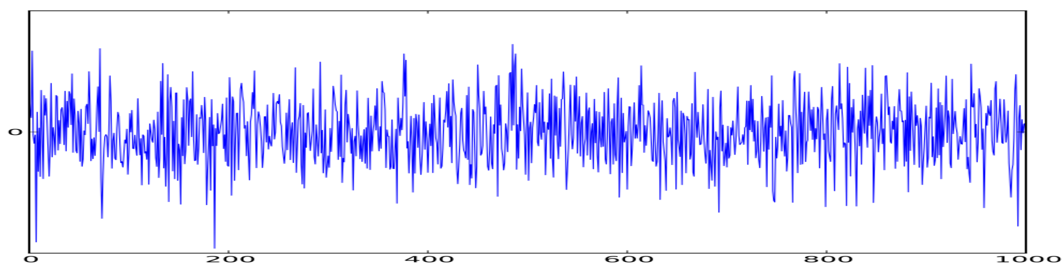

Figure 4.3: White Noise. Not Predictable.

4.5.1 Ljung-Box Test

The method to test for White Noise is the Ljung-Box Test.
Ljung-Box Test begins with the calculation of Q statistic of each lag terms l:

$$Q(l) = n(n+2)\sum_{k=1}^{l}\frac{\rho_k^2}{n-k} \sim \chi_l^2 \tag{4.21}$$

where:

$Q(l)$ = Q statistic of each lag term l; acts like the t-score in a t-test
n = number of sample size
l = lag term
ρ_k = autocorrelation of lag k
χ_l^2 = Chi Square Distribution

After each lag term Q(l) has been calculated, next step is to compare each term to the critical value of the χ_l^2. It can be seen that Q(l) increase proportional with the ρ; therefore, larger the autocorrelation, larger $Q(l)$:

1. H_0: White Noise
2. H_1: Not Random, exists autocorrelation

- If Q(l) $> \chi_l^2$, reject H_0. The time series exists autocorrelation.
- If Q(l) $< \chi_l^2$, accept H_0. The time series is a white noise.

[6]Can be check by setting a H_0:γ=0,H_1:γ<0, and do a hypothesis test DF=$\frac{\hat{\gamma}}{SE(\hat{\gamma})}$ In addition, the DF = Dickey Fuller Distribution

4.6 FAANG Characteristics

```python
from statsmodels.tsa import stattools
import statsmodels.api as sm
from arch.unitroot import ADF
stattools.acf(data['Stock_Return'],nlags=30) # Gives each acf of the corresponding lag term
sm.graphics.tsa.plot_acf(data['Stock_Return'], lags=30); #Plot the acf with 30 term lags
stattools.pacf(data['Stock_Returnt'],nlags=30)
sm.graphics.tsa.plot_pacf(data['Stock_Return'], lags=30);
adf = ADF(data['Stock_Return'])
adf.summary() #Can also use adf.stat to get the DF stat
lb = stattools.q_stat(stattools.acf(data['Stock_Return'])[1:10],len(data['Stock_Return'])) # Begins with 1. If not, acf is definitely correlated because there is no lag term.
lb[1][-1]# The output will be the p-value of the overall Q(1). If output <0.05, reject H_0 and is not white noise.
```

1. Facebook:
   ADF Test: DF = -15.55 < -2.86 = Critical Value of 95%DF
   → Facebook is Stationary
   Ljung-Box Test: p-value of LB Stat = 0.00563 < 0.05 = 95% confidence level
   → Facebook is not white noise

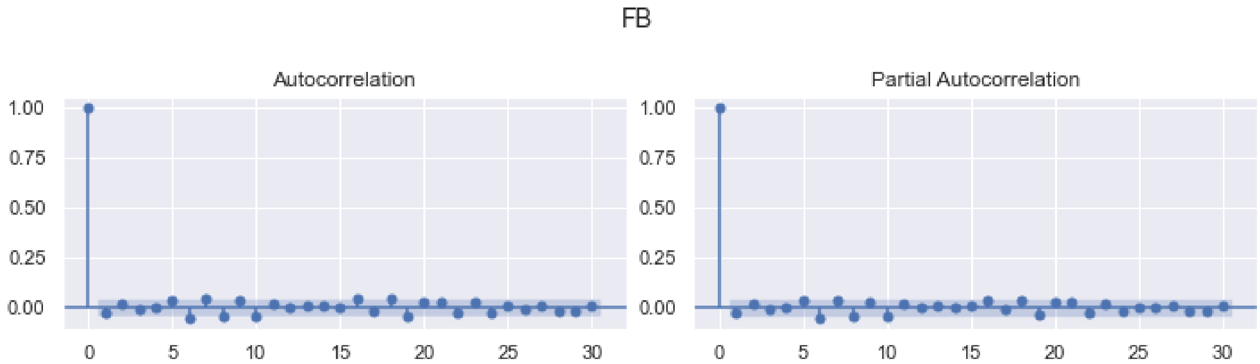

Figure 4.4: Facebook ACF & PACF

2. Amazon:
   ADF Test: DF = -15.36 < -2.86 = Critical Value of 95%DF
   → Amazon is Stationary
   Ljung-Box Test: p-value of LB Stat = 0.00479 < 0.05 = 95% confidence level
   → Amazon is not white noise

Figure 4.5: Amazon ACF & PACF

3. Apple:
   ADF Test: DF = -11.59 < -2.86 = Critical Value of 95%DF
   → Apple is Stationary
   Ljung-Box Test: p-value of LB Stat = 0.000 < 0.05 = 95% confidence level
   → Apple is not white noise

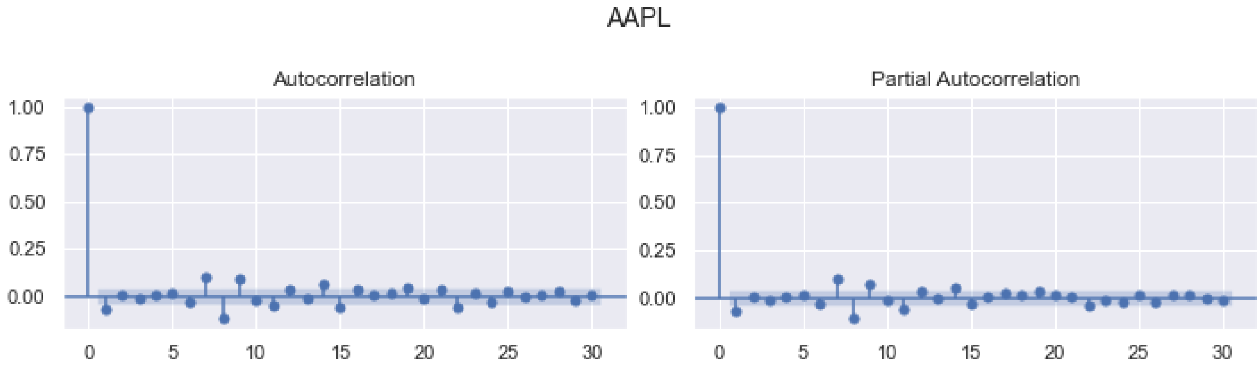

Figure 4.6: Apple ACF & PACF

4. Netflix:
   ADF Test: DF = -44.29 < -2.86 = Critical Value of 95%DF
   → Apple is Stationary
   Ljung-Box Test: p-value of LB Stat = 0.237> 0.05 = 95% confidence level
   → Netflix is not autocorrelated, it is a white noise

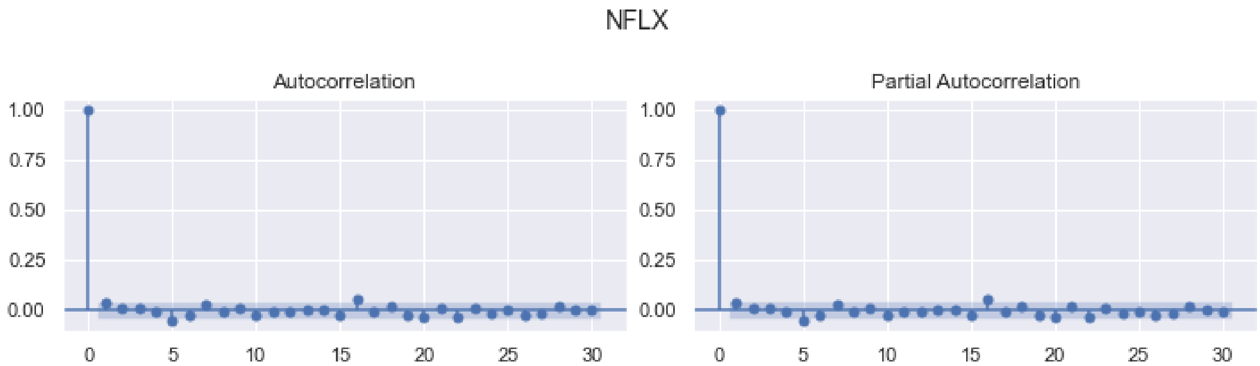

Figure 4.7: Netflix ACF & PACF

5. Google:
   ADF Test: DF = -15.82 < -2.86 = Critical Value of 95%DF
   → Google is Stationary
   Ljung-Box Test: p-value of LB Stat = 0.000 < 0.05 = 95% confidence level
   → Google is not white noise

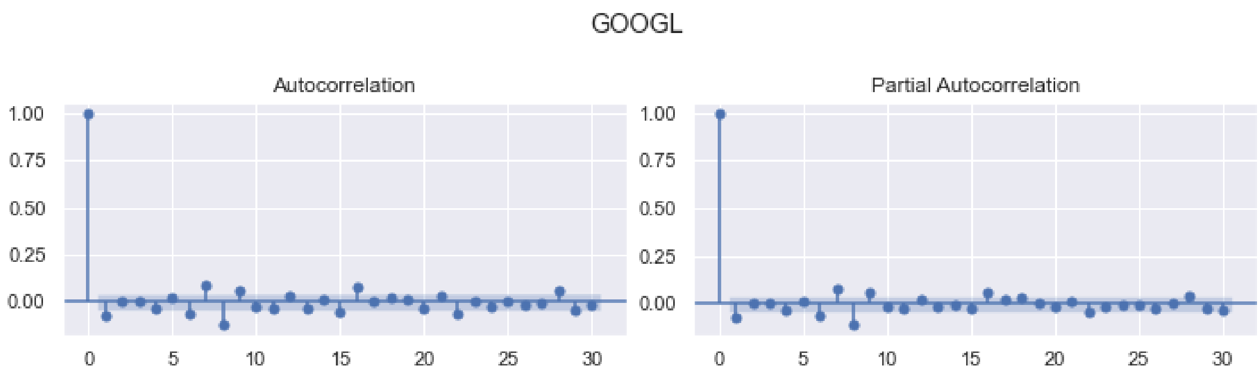

Figure 4.8: Google ACF & PACF

In conclusion, after running through the ADF and Ljung-Box test, it can be tell that all the stocks of FAANG are stationary. However, Netflix is the only one that is a stationary white noise.

Chapter 5

MA & AR & ARMA

5.1 Introduction

After knowing the characteristics of the time series, if the data are stationary and not White Noise, forecast can be the next step for the analysis. To do a forecast, building a model is necessary. In this chapter, Moving Average (MA), Autoregression (AR), Autoregression Moving Average (ARMA) will be introduced.

5.2 Moving Average (MA)

Moving average takes the past l term's average to be the expected value of the next term. It is the easiest way to make predictions, where only averages will be evaluate. In addition,there are three main types of moving average:simple MA, weighted MA, and Exponential MA.

5.2.1 Simple Moving Average

Simple Moving Average simply takes the past l term's average as the expected value of next term:

$$\hat{x}_{t+1} = \frac{x_t + x_{t-1} + x_{t-2} + \cdots + x_{t-n+1}}{n} \tag{5.1}$$

where:

$\hat{x}_{t+1}$ = predicted value of next term
$_t$ = value of current term
$_{t-i}$ = value of t-i term
n = number of observations

5.2.2 Weighted Moving Average

It can be tell that Simple MA gives every term an equal weight. However, in reality, the weight of each term may not be the same. Therefore, in order to make better prediction, Weighted MA gives out different weight to each term:

$$\hat{x}_{t+1} = w_0 x_t + w_1 x_{t-1} + x_{t-2} + \cdots + w_n x_{t-n+1} \tag{5.2}$$

where:

w$_i$ = weight of each term

5.2.3 Exponential MA

Exponential MA is an extension of Weighted MA, where each term's weight is given as an exponential decay (the further the data, the smaller the effect):

$$\hat{x}_{t+1} = \alpha x_t + \alpha(1-\alpha) w_1 x_{t-1} + \alpha(1-\alpha)^2 x_{t-2} + \cdots + \alpha(1-\alpha)^n x_{t-n+1} \tag{5.3}$$

where:

α = a constant smoothing factor between 0 and 1 (practical choice would be 0.05-0.3)

5.3 Autoregressive Moving Average Model (ARMA)

5.3.1 Autoregressive Model (AR)

Autoregression Model is a linear regression model that takes the past values as the explanatory variables and the current value as the dependent valuable.

$$x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \epsilon_t \tag{5.4}$$

where:

ϵ = error term; is a white noise (E[ϵ]=0,Var(ϵ_t)=σ_ϵ^2)and has no correlation with x_i terms
ϕ[1] = $|\phi| < 1$

Correlation of p term can be evaluated:

1. Subtract mean[2] μ from both side of eq. 5.4:

$$x_t - \mu = \phi_1(x_{t-1} - \mu) + \phi_2(x_{t-2} - \mu) + \cdots + \phi_p(x_{t-p} - \mu) + \epsilon_t \tag{5.5}$$

2. Multiply each side with (x_t-μ) to get Autocovariance[3]:

$$(x_t - \mu)(x_t - \mu) = \gamma_0 = \phi_1(x_{t-1} - \mu)(x_t - \mu) + \phi_2(x_{t-2} - \mu)(x_t - \mu) + \cdots + \phi_p(x_{t-p} - \mu)(x_t - \mu) \tag{5.6}$$

$$\rightarrow \gamma_0 = \phi_1\gamma_1 + \phi_2\gamma_2 + \cdots + \phi_p\gamma_p \tag{5.7}$$

3. Divide each side with γ_0

$$\frac{\gamma_0}{\gamma_0} = 1 = \phi_1\frac{\gamma_1}{\gamma_0} + \phi_2\frac{\gamma_2}{\gamma_0} + \cdots + \phi_p\frac{\gamma_p}{\gamma_0} = \phi_1\rho_1 + \phi_2\rho_2 + \cdots + \phi_p\rho_1 \tag{5.8}$$

4. By step 1 - 3, general correlation can be evaluate:

$$\rho_1 = \phi_1 + \phi_2\rho_2 + \phi_3\rho_2 + \cdots + \phi_p\rho_{p-1} \tag{5.9}$$

$$\rho_2 = \phi_1\rho_1 + \phi_2 + \phi_3\rho_1 + \cdots + \phi_p\rho_{p-2} \tag{5.10}$$

$$\rightarrow \rho_p = \phi_1\rho_{p-1} + \phi_2\rho_{p-2} + \phi_3\rho_{p-3} + \cdots + \phi_p\rho_p \tag{5.11}$$

5. From above, conclusion for stationary time series can be drawn: as p increases, the correlation decreases but may not be zero.

5.3.2 Moving Average Model (MA)

Moving Average Model is a linear regression model that takes the error term of each term ϵ_t as the explanatory variables and the current value as the dependent valuable.

$$x_t = \mu + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \cdots + \theta_q\epsilon_{t-q} \tag{5.12}$$

where:

$\{\epsilon_t\}$ = white noise
$E(x_t)$ = μ
$Var(x_t)$[4] = E[(X$_t-\mu$)(X$_t-\mu$))]=γ_0=$(1+\theta_1^2 + \theta_2^2 + \cdots + \theta_q^2)\sigma_\epsilon^2$

Corresponding Correlations:

1. $l = 0$

$$\rho = \frac{Cov(X_k, X_{k-l})}{Var(X_k)} = \frac{Cov(X_k, X_k)}{Var(X_k)} = \frac{Var(X_k)}{Var(X_k)} = 1 \tag{5.13}$$

[1] If $|\phi| \geq 1$, series will not be stationary and nothing can be inferred.
[2] μ=E[x_t]=$\phi_0 + \phi_1\mu_1 + \phi_2\mu_2 + \cdots + \phi_p\mu_p$. Since time series is stationary,$\mu_1 = \mu_2 = \cdots = \mu_p$
[3] $\gamma_l = E[(X_k - \mu_k)(X_{k-l} - \mu_{k-l})]$
[4] Var(x_t)=Var($\epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \cdots + \theta_q\epsilon_{t-q}$)$\rightarrow$apply variance property Var[θX]=$\theta^2 Var[X]$

2. l = 1,2,...,q

$$Cov(X_k, X_{k-l}) = (x_k - \mu)(x_{k-l}) \tag{5.14}$$

$$\rightarrow (\epsilon_k + \theta_1\epsilon_{k-q} + \cdots + \theta_q\epsilon_{k-l})(\epsilon_{k-l} + \theta_1\epsilon_{k-1} + \cdots + \theta_q\epsilon_{k-l-q}) \tag{5.15}$$

   Write in Tabular Form[5]:
   $(\epsilon_k + \theta_1\epsilon_{k-1} + \cdots + \theta_l\epsilon_{k-l} + \cdots + \theta_q\epsilon_{k-l})$
   $\times \qquad (\epsilon_{k-l} + \cdots + \theta_{q-l}\epsilon_{k-l} + \cdots + \theta_q\epsilon_{k-l-q})$

   $Cov(X_k, X_{k-l})$ will be:

$$\sigma^2(\theta_l + \theta_1\theta_{l+1} + \theta_2\theta_{l+2} + \cdots + \theta_q\theta_{q-l}) \tag{5.16}$$

   ρ will then be:

$$\rho = \frac{Cov(X_k, X_{k-l})}{\gamma_0} = \frac{\sigma^2(\theta_l + \theta_1\theta_{l+1} + \theta_2\theta_{l+2} + \cdots + \theta_q\theta_{q-l})}{(1+\theta_1^2+\theta_2^2+\cdots+\theta_q^2)\sigma_\epsilon^2} = \frac{(\theta_l + \theta_1\theta_{l+1} + \theta_2\theta_{l+2} + \cdots + \theta_q\theta_{q-l})}{(1+\theta_1^2+\theta_2^2+\cdots+\theta_q^2)} \tag{5.17}$$

3. l > q:
   Take MA(2) as example:

$$X_t = \mu + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2}$$
$$X_{t-3} = \mu + \epsilon_{t-3} + \theta_1\epsilon_{t-4} + \theta_2\epsilon_{t-5}$$

   Since there are no common ϵ terms, ρ=0.

5.4 Autoregression Moving Average Model (ARMA)

The concept of ARMA is that a time series x_t may not only depends on its past terms, but also some random errors. Therefore, ARMA[6] combines the AR and MA to combine both effects.
ARMA(p,q) is represent as:

$$x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \cdots + \theta_q\epsilon_{t-q} \tag{5.18}$$

Steps to Construct ARMA Model:

1. Identify the characteristic of the time series data → Stationary (ADF test) and White Noise (Ljung-Box test).
   If Non-stationary, transfer to stationary.
   If the data is a White Noise, stop analysis.

2. Choose optimal p, q value for the ARMA Model.

Model	ACF	PACF
AR(p)	Tails Off	0 After p
MA(q)	0 After q	Tails Off
ARMA(p,q)	Tails Off	Tails Off

   For ARMA, that's not easy to choose the right p, q. Therefore, one way is to generate few ARMA with different combinations of (p,q) and choose the ARMA model with the minimum AIC[7] score.

3. Check if the residue of the model is White Noise. When the residue is White Noise, it means that complete information has been included in the model.

[5] Remember that only Var[ϵ_i]=σ_ϵ^2. E[$\epsilon_t\epsilon_s$]=0 for any s $\neq t$
[6] Notice that ARMA(p,0)=AR(p), ARMA(0,q)=MA(q)
[7] Akaike information criterion → AIC = 2k-2L, where k = # of parameters and L = log likelihood = how well the model is fitted. The concept of AIC is to have the best fitting model with the lowest parameters (overfitting is prevented)

5.5 FAANG ARMA

From last chapter, stationary and White Noise of FAANG returns have been checked and only Netflix will be excluded from the following analysis.

```python
from statsmodels.tsa import arima_model
stockTrain = data[stock]['2012':'2019']
stockTest = data[stock]['2020']
norders = [(1,0,1),(2,0,1),(2,0,2),(3,0,1),(3,0,2),(3,0,3)]
aics = []
for norder in norders:
    model = arima_model.ARIMA(stockTrain,order=norder).fit()
    aics.append(model.aic)
nindex = aics.index(min(aics))
bestmodel = arima_model.ARIMA(stockTrain,order=norders[nindex]).fit()
stock_forecast = pd.DataFrame(bestmodel.forecast(len(stockTest))[0]
```

1. FB: ARMA(2,2) & Residue's Ljung-Box p-value: 0.83 → residue is White Noise

$$x_t = 0.00114 - 0.965x_{t-1} - 0.991x_{t-2} + 0.965\epsilon_{t-1} + 0.999\epsilon_{t-2} \tag{5.19}$$

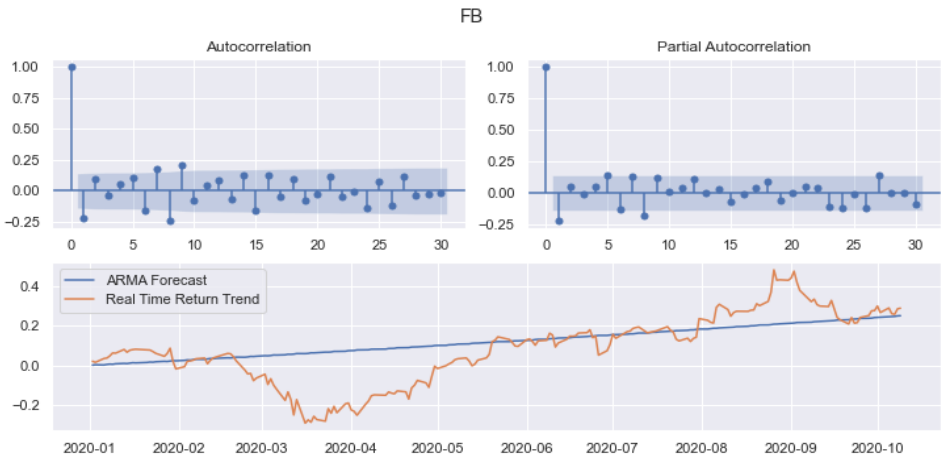

Figure 5.1: Top is Facebook ACF & PACF. Bottom is Facebook ARMA forecast compared to real time trend.

2. AMZN: ARMA(2,1) & Residue's Ljung-Box p-value: 0.68 → residue is White Noise

$$x_t = 0.00129 + 0.629x_{t-1} - 0.0416x_{t-2} - 0.635\epsilon_{t-1} \tag{5.20}$$

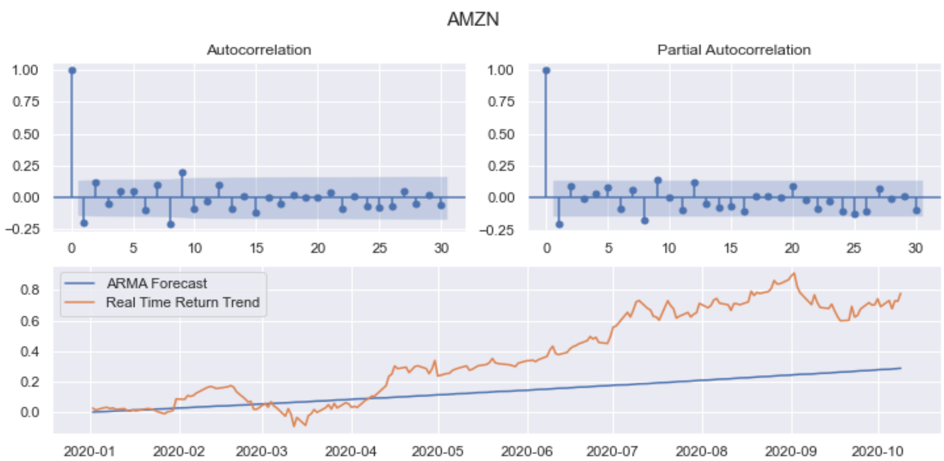

Figure 5.2: Top is Amazon ACF & PACF. Bottom is Amazon ARMA forecast compared to real time trend.

3. AAPL: ARMA(2,2) & Residue's Ljung-Box p-value: 0.19

$$x_t = 0.000908 - 0.789x_{t-1} - 0.989x_{t-2} + 0.799\epsilon_{t-1} + 0.986\epsilon_{t-2} \tag{5.21}$$

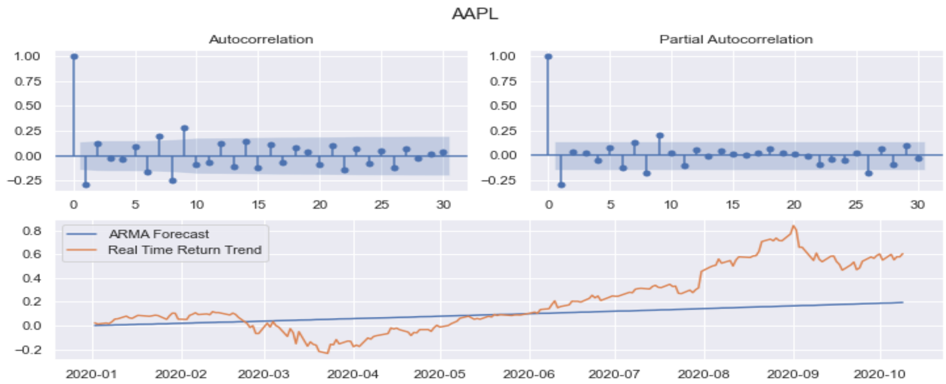

Figure 5.3: Top is Apple ACF & PACF. Bottom is Apple ARMA forecast compared to real time trend.

4. GOOGL: ARMA(3,3) & Residue's Ljung-Box p-value: 0.45

$$x_t = 0.000839 + 0.0109x_{t-1} + 0.0823x_{t-2} + 0.873x_{t-3} + 0.005\epsilon_{t-1} - 0.124\epsilon_{t-2} - 0.882\epsilon_{t-3} \tag{5.22}$$

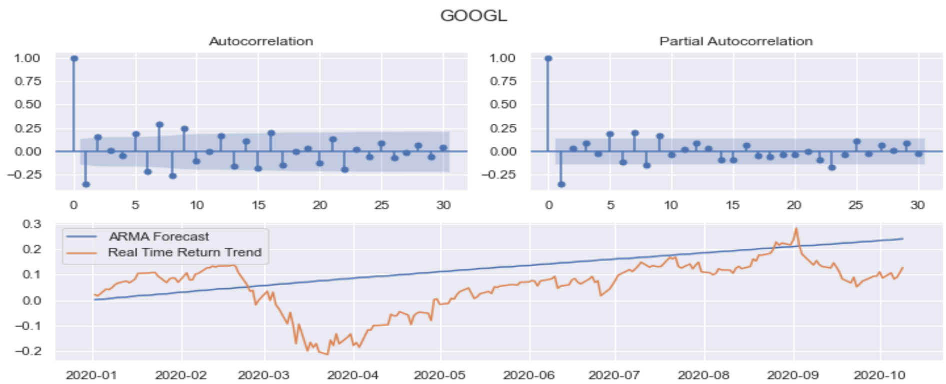

Figure 5.4: Top is Google ACF & PACF. Bottom is Google ARMA forecast compared to real time trend.

In conclusion, AR captures the historical lag term's effect on current rate of return and MA captures the unexpected events of the White Noise. Compare to Amazon and Apple, it can be seen that ARMA has done a better forecast on Facebook and Google. The reason behind this is the ARCH effect, which will be discussed on next chapter.

Chapter 6

ARCH and GARCH

6.1 Introduction

After building an ARMA model for the FAANG, it's found that ARMA didn't have a good performance on Amazon and Google. The reason for the poor performance is the ARCH effect, where the volatility of the returns are not constant over time (homoskedasticity); instead, it's changing over time (heteroskedasticity), even more, the volatility is autocorrelated. In this chapter, ARCH effect will be examine.

6.2 ARCH

ARCH is used for capture the volatility of a time series. It first assumes that the model satisfies one of the AR, MA, or ARMA model. For example, AR(1) model:

$$y_t = a_0 + a_1 y_{t-1} + \epsilon_t \tag{6.1}$$

Then, the volatility can be represent with ϵ_t as:

$$Var(y_t|y_{t-1}) = Var(\epsilon_t|y_{t-1}) \tag{6.2}$$

Next, to capture volatility clustering, the volatility can be infer as correlated with the square of the ϵ:

$$\epsilon_t = \sigma_t u_t \sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \cdots + \alpha_p \epsilon_{t-p}^2 \tag{6.3}$$

To verify the ARCH effect:
Step 1) Check the stationary of the time series and construct a model for the time series (e.g. ARMA). Step 2) Use Ljung-Box to check the autocorrelation of ϵ_t^2. Step 3) If the ϵ_t^2 show's clustering, construct the model with estimators.

```
def arch(stock,pq):
    ############### Split Data ########################
    train = data[stock]['2012':'2019']
    test = data[stock]['2020']
    stock_name = stock.replace('_Ret','')
    ############### Check Residue^2  #################
    model = arima_model.ARIMA(train,order = pq).fit()
    stdresid = model.resid/math.sqrt(model.sigma2)
    lbt = stattools.q_stat(stattools.acf(stdresid**2)[1:30],len(stdresid))
    wnNumber = lbt[1][-1]
    print('Ljung-Box p-value = %.5f'%(wnNumber))
    if wnNumber > 0.05:
        print('%s is a White Noise'%(stock_name))
    else:
        print('ARCH Effect Exists')
    ###############  ################################
```

6.3 GARCH

GARCH is an derivation of ARCH, where the effect of volatility itself is also included for the prediction of volatility. The concept is similar to the ARMA, where AR captures the time series itself and MA captures the ϵ:

$$\epsilon_t = \sigma_t u_t \sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \epsilon_{t-j}^2 \tag{6.4}$$

Note that both ARCH and GARCH only captures the statistic characteristic of a time series but not used for interpret the cause of the characteristic.

```
import datetime as dt
import pandas_datareader.data as web
from arch import arch_model
data_train = data['data_Ret']['Train Start_Yr':'Train End_Yr']
data_test = data['data_Ret']['Test Yr']
data = data['data_Ret']
am = arch_model(data)
split_date = dt.datetime(2020,1,2) #Here takes 2020/1/2 as example
result = am.fit(last_obs=split_date)
result.summary() #Print out the GARCH result
forecasts = res.forecast(horizon=1, start=split_date) #Forecast the volatility
forecasts_variance = forecasts.variance
forecasts.variance[split_date:].plot() #Plot the volatility
```

6.4 FAANG GARCH

1. AMZN: ARMA(2,1) & Square Root of Residue's Ljung-Box p-value: 0 $\rightarrow$ residue has ARCH effect

$$r_t = 0.00176 + \epsilon_t \tag{6.5}$$

where:

$$\begin{aligned}\epsilon_t &= \sigma_t * u_t\\ \sigma_t^2 &= 0.0000337 + 0.1\epsilon_{t-1}^2 + 0.8\sigma_{t-1}^2\end{aligned}$$

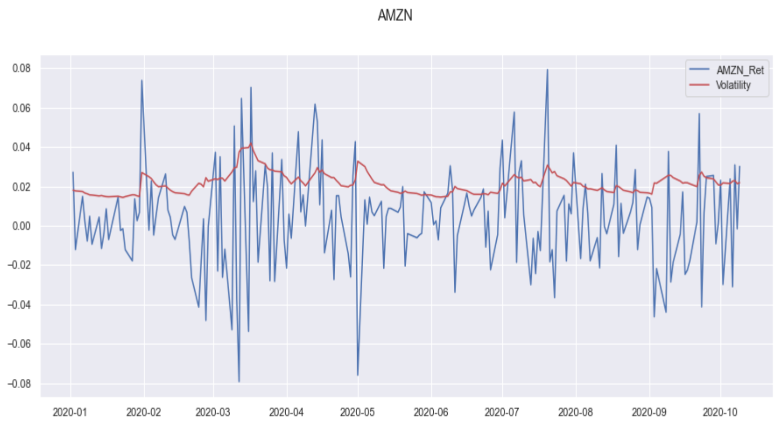

Figure 6.1: Forecast volatility on Amazon

2. AAPL: ARMA(2,2) & Square Root of Residue's Ljung-Box p-value: 0 $\rightarrow$ residue has ARCH effect

$$r_t = 0.00149 + \epsilon_t \tag{6.6}$$

where:

$$\begin{aligned}\epsilon_t &= \sigma_t * u_t\\ \sigma_t^2 &= 0.0000253 + 0.1\epsilon_{t-1}^2 + 0.8\sigma_{t-1}^2\end{aligned}$$

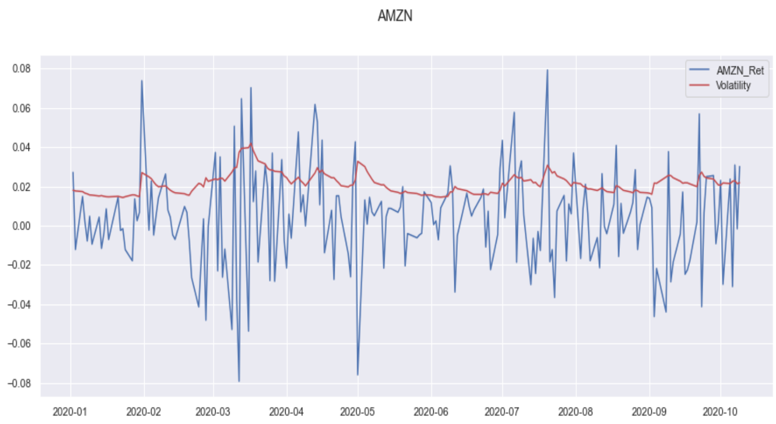

Figure 6.2: Forecast volatility on Apple