**Time-Series Analysis**

**20161342 KeonWooKim**

1. **Get your own time series data**

**Data Source**

-Yahoo Finance

<https://finance.yahoo.com/quote/TSLA/history?period1=1546300800&period2=1606262400&interval=1d&filter=history&frequency=1d&includeAdjustedClose=true>

**Data Explanation**

-Tesla Inc (TSLA)’s daily Adj Close price

-There are 495 daily data (19-01-01~20-12-16)

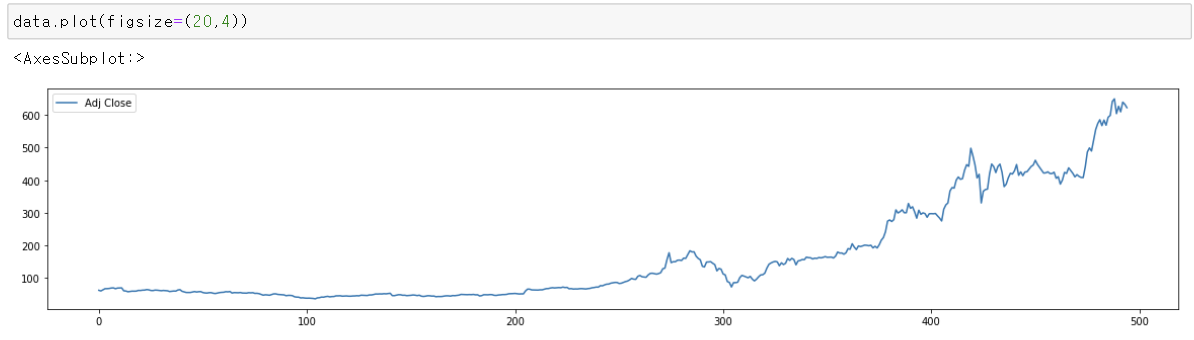
-To ensure continuity of data, ‘Adj Close’ price is used when analyzing time series data not (‘Open’, ‘High’, ‘Low’, ‘Close’) price.

1. **Set a relevant problem**

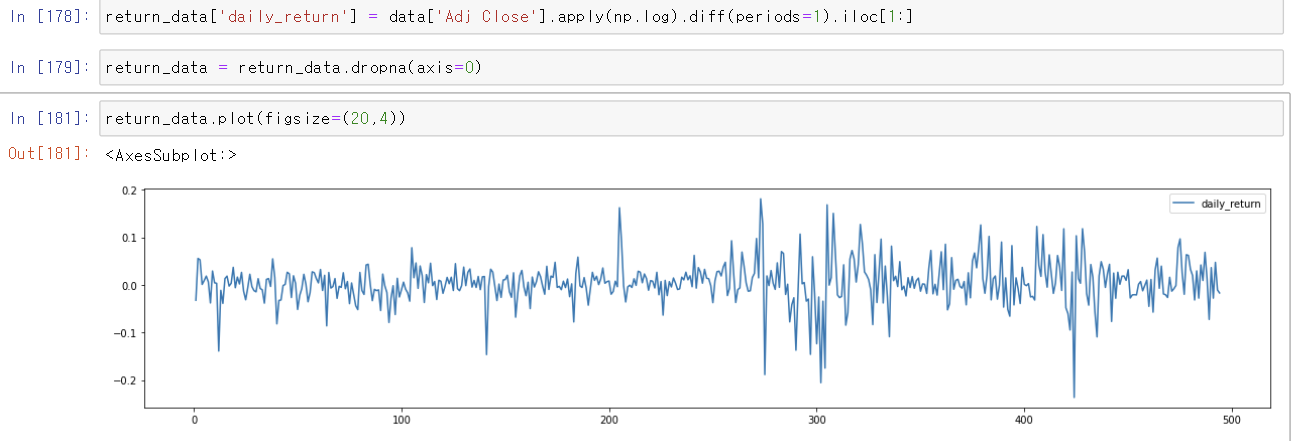
Tesla is an one of attractive stocks with people’s interest in the stock market because it is highly profitable and has a very bright corporate outlook and expectation. In fact, a lot of people in Korea are also investing in Tesla stocks these days, which is so called ‘Seo-Hak-Gae-Me(서학개미)’. However, I read multiple articles depicting Tesla’s stock as a risk bubble that may burst at any time. Thus, it gave me a chance to look at aggregated financial information from ‘www.investing.com’. I found that Beta value, a coefficient that measures the sensitivity of individual’s stocks’ share price return when changes occurred in the stock market, was very high about ‘2.16’. Moreover, the figure for ‘PER’, the number of shares divided by EPS, was ridiculously higher than that of its same industry group. When looking for ‘PER’ in Toyota, Honda, and BMW, the average ‘PER’ figure in the auto industry is about 7 to 10 but Tesla’s ‘PER’ is recorded as 1137.26 which is over 100 times higher than others. With these circumstances, Tesla can be judged to be a bubble risky stock, and as soon as Tesla’s stock price falls sharply, many people will lose their money, which could be directly related to social chaos. Through this project, I would like to use the time series data analysis methodology learned in several lecture to find out stochastically when will Tesla’s stock price rise or fall in the future in order to establish a strategy. I strongly believe that if the data analysis in this project yields a significant probability, it will alleviate some of the confusion which can be occurred soon in the stock market.



1. **Plot the time series data**

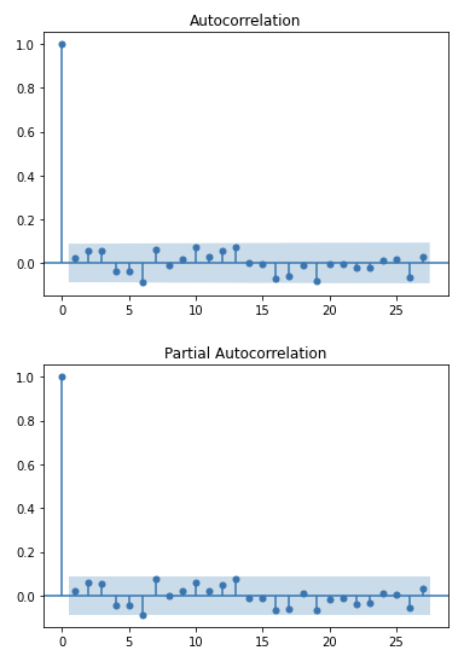


As I depicted Tesla’s stock price (2019.01.01~2020.12.16), it shows an extreme right-upward trend in which price rises over time. Before analyzing this data, it should be assumed that the distribution of stock prices follows the Random-Walk. Also, we can know that its graph does not show stationary series but nonstationary. Thus, we should make it as stationary by transforming data.

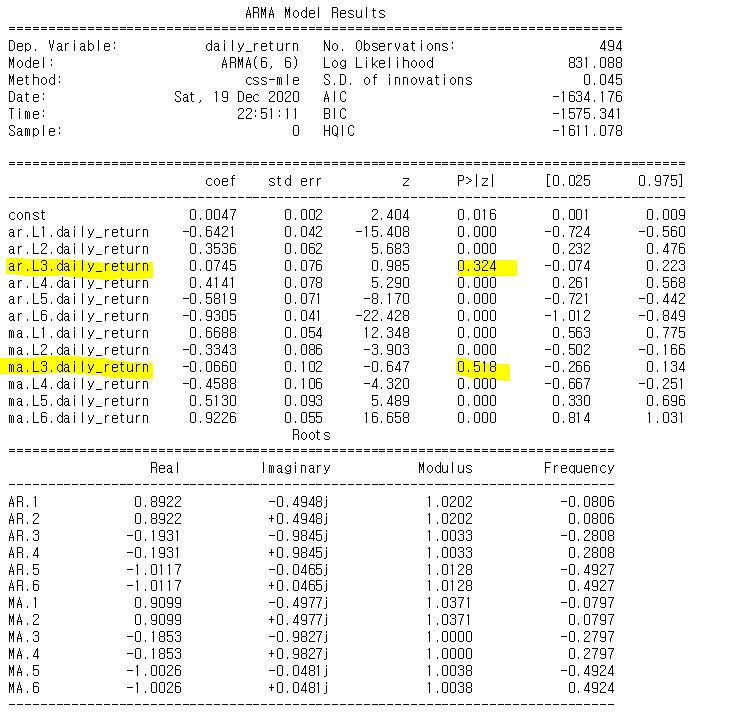


Since it is based on financial time series data I applied log transformation then, also applied one difference to make the series more stationary. Thus, its plot seems to have constant mean and variance compared to original plot.

1. **ARIMA(6,1,6) model**



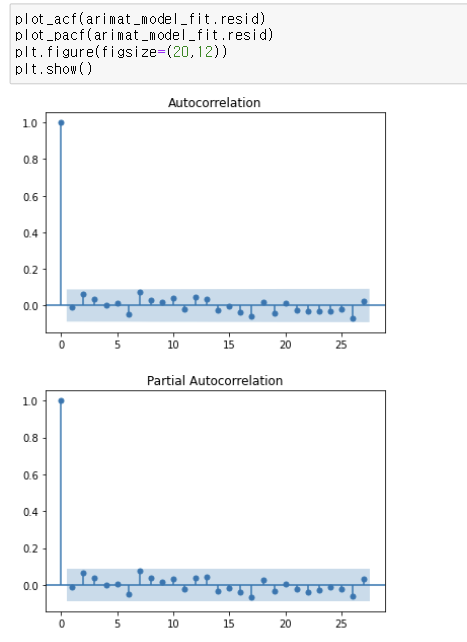
As we see the ACF plot first, we can find out that it cuts off after lag6. Then, as we see PACF plot, it shows a similar distribution to the ACF plot, and it also cuts off after lag6. Thus, I decided to use MA(6) due to ACF, AR(6) due to PACF and one difference from transformation, overall, ARIMA(6,16) model is used.

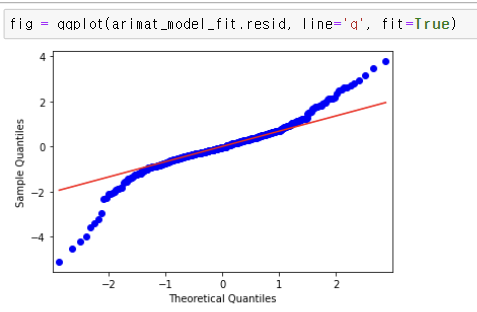


It is summary of ARIMA(6,16) model, and it shows the overall statistics of its model. AIC = -1634.176, BIC = -1575.341, and coefficients of ARIMA(6,16) models etc. are shown here. One thing to note is that coefficient values in AR3 and MA3 are statistically insignificant due to its high p-values.



Through plot of residuals in ARIMA(6,16) model, we can identify a rectangular scatter around a zero horizontal level with no trends which means its model is adequate.





Also, since it can be concluded as it doesn’t show statistically evidence of non-zero ACF in residuals I concluded that its model is appropriate to use. Although one thing I’m worried about is that when I look at the QQ-plot, the two ends are not linear, since the overall data is close to a red line we can assess its normality is fine. To sum up, ARIMA(6,16) is appropriate model to use.

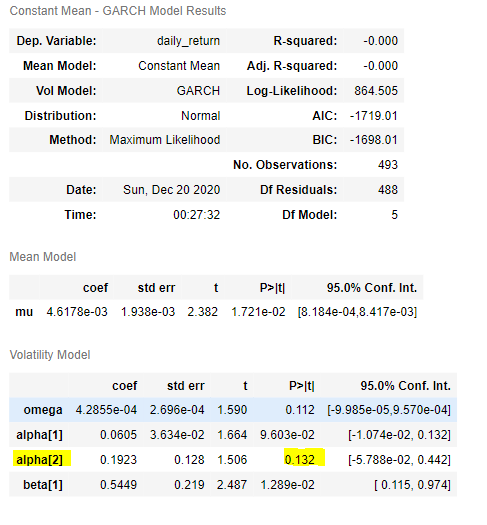
1. **GARCH(2,1) model**

Since stock price data is financial time series data, I applied a heteroscedastic model to get better result compared to first model I used ‘ARIMA(6,1,6)’. This is because ARIMA model is based on conditional mean structure of time series data but assumption of a constant conditional variance is violated. So heteroscedastic model is appropriate to use since it can model the conditional variance structure of time series data, especially, GARCH model.

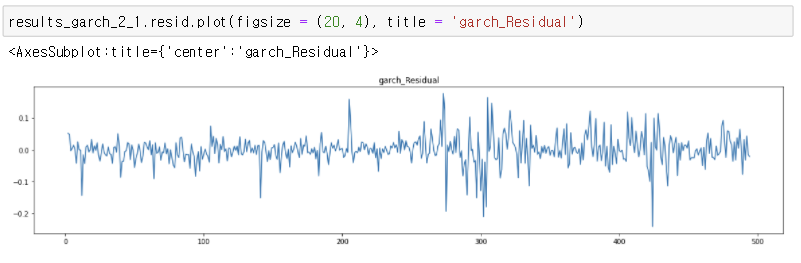
To determine the parameter of GARCH(p,q), I compared AIC and BIC of the GARCH model with different parameters in the ‘built-in’ method.

 **GARCH(1,1) GARCH(2,1)**  **GARCH(1,2) GARCH(2,2) GARCH(3,2)**

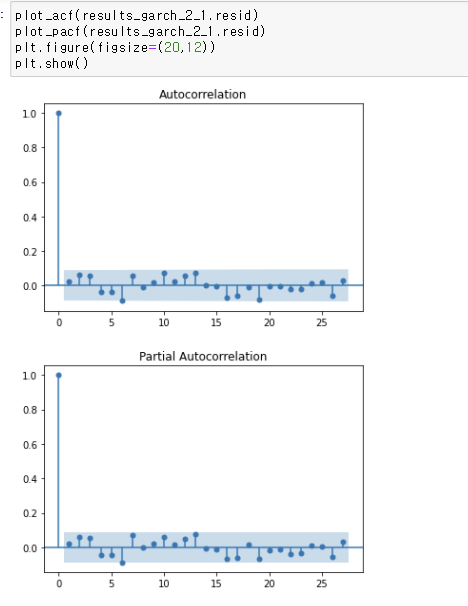
Increasing the value of the parameter (p,q) one by one from (1,1), I printed out value of each model’s AIC and BIC values sequentially. I stopped built-in process on GARCH(3,2), since it seems to have higher values on ‘BIC’. Among them, GARCH(2,1) model shows most statisticallly appropriate values which mean having the lowest AIC=-1703.62, BIC=-1682.62 compared to others. Its value of AIC and BIC are even lower than previous ARIMA(6,1,6) model so its heteroscedastic model seems to be more appropriate model since it has financial time series data.

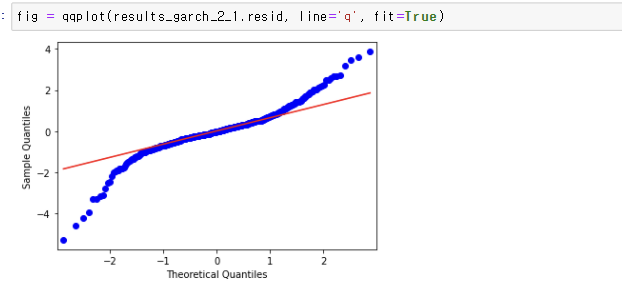


I selected GARCH(2,1) model, and it is a summary of its model’s statistics value. Through this table, it shows that the value of Alpha(2) is not statistically significant due to its high p-value. So, it would be better to use only omega, Alpha(1) and Beta(1) on formula.



Through plot of residuals in GARCH(2,1) model, we can identify a rectangular scatter around a zero horizontal level with no trends which means its model is adequate.





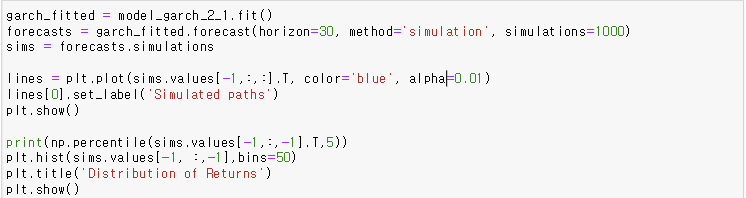
Also, since it can be concluded as it doesn’t show statistically evidence of non-zero ACF in residuals except on lag6 for small amount I concluded that its model is appropriate to use. Although one thing I’m worried about is that when I look at the QQ-plot, the two ends are not linear, since the overall data is close to a red line we can assess its normality is fine. To sum up, GARCH(2,1) is appropriate model to use.

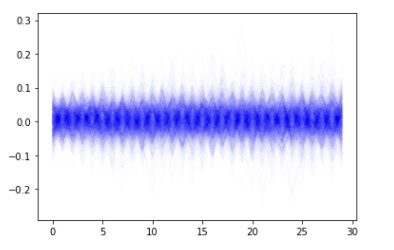
1. **Selected model prediction-GARCH(2,1) model**

As I mentioned above, since my data follows financial time series data, GARCH(2,1) model shows more statistically signifcant results than ARIMA(6,1,6) model. As we comapre AIC and BIC values, GARCH has much lower values than ARIMA’s.

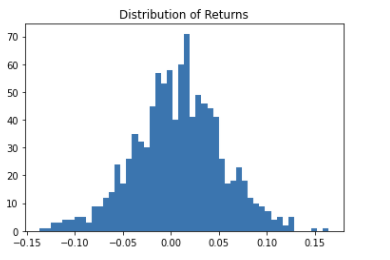
ARIMA(6,1,6)\_AIC = -1634.18, BIC = -1575.34, GARCH(2,1)\_ AIC=-1703.62, BIC=-1682.62

Thus, I finally chose GARCH(2,1) model.





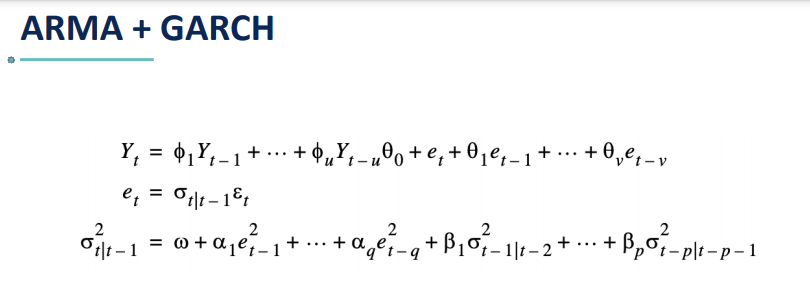
I simulated 30 days from the end of TESLA’s stock price (2020-12-17~30days). Its simulation results shows that darker blue parts are more likely path than lighter parts. In this graph we can infer days having lower/higher volatility. The code does not really care about the inter-path but only how the end of 30 day period look like.



Its histogram shows the end of 30 day’s distribution which means that by looking at first sparse figure swinging around at 90 degree angle. (darker side: taller part, lighter side: lower part).

1. **Inference**

From the previous analysis, as I learned on Time-Series Analysis lecture this semester, GARCH model shows better result compared to ARIMA model based on AIC, BIC since my data has finance characteristics. However, on each model’s residuals autocorrelation, GARCH(2,1) model shows statistically significant value on lag6 while ARIMA(6,1,6) does not. I’d better to combine these two models in order to utilize each model’s advantages. Thus, next time it would better to predict future price by using combination of ARIMA and GARCH model that I made above. Since ARIMA model estimates the conditional mean and GARCH model estimates the conditional variance present in the residuals of ARIMA estimation. While making ARIMA-GARCH model, it would be really important problem choosing parameters ARIMA(p1,d,q1)-GARCH(p2,q2) based on aic or bic.



To sum up, since the results of my time-series analysis through ARIMA and GARCH model do not show statistically significant and clear values, I believe that predicting the future prices is not appropriate with theses methods. Otherwise, if I chose analyzing TESLA’s stock price volatility not price as project’s topic, it would have better results and be utilized much better.

Problems to be overcome:

1. More days of stock prices are needed (more data)
2. Better data processing
3. Use a high-level model (ex.ARIMA-GARCH, State Space, Neural Network)

Next time if I solved out these three problems I thought while doing this project, it would have much better results then could be helpful to people I mentioned on **(2.Set a relevant problem).**

1. **Python Code**
2. **import** **pandas** **as** **pd**
3. **import** **numpy** **as** **np**
4. **import** **matplotlib.pyplot** **as** **plt**
5. **from** **arch** **import** arch\_model
6. **from** **statsmodels.graphics.tsaplots** **import** plot\_acf, plot\_pacf
7. **from** **statsmodels.tsa.arima\_model** **import** ARIMA
8. **from** **statsmodels.graphics.api** **import** qqplot
9. data = pd.read\_csv('C:/Users/김건우/Desktop/UNIST/3학년 2학기/시계열분석(MGE)/tsla\_price\_data.csv')
10. data.plot(figsize=(20,4))
11. return\_data = pd.DataFrame()
12. return\_data['daily\_return'] = data['Adj Close'].apply(np.log).diff(periods=1).iloc[1:]
13. return\_data = return\_data.dropna(axis=0)
14. return\_data.plot(figsize=(20,4))
15. plot\_acf(return\_data)
16. plot\_pacf(return\_data)
17. plt.figure(figsize=(20,12))
18. plt.show()
19. arima\_model = ARIMA(return\_data, order=(6,0,6))
20. arimat\_model\_fit = arima\_model.fit(trend='c',full\_output=**True**, disp=1)
21. print(arimat\_model\_fit.summary())
22. arimat\_model\_fit.resid.plot(figsize = (20, 4), title = 'arima\_Residual')
23. plot\_acf(arimat\_model\_fit.resid)
24. plot\_pacf(arimat\_model\_fit.resid)
25. plt.figure(figsize=(20,12))
26. plt.show()
27. fig = qqplot(arimat\_model\_fit.resid, line='q', fit=**True**)
28. model\_garch\_1\_1 = arch\_model(return\_data.daily\_return[1:],mean='Constant',vol='GARCH',p=1,q=1)
29. results\_garch\_1\_1 = model\_garch\_1\_1.fit()
30. results\_garch\_1\_1.summary()
31. results\_garch\_1\_1.resid.plot(figsize = (20, 4), title = 'garch\_Residual')
32. plot\_acf(results\_garch\_1\_1.resid)
33. plot\_pacf(results\_garch\_1\_1.resid)
34. plt.figure(figsize=(20,12))
35. plt.show()
36. fig = qqplot(results\_garch\_1\_1.resid, line='q', fit=**True**)
37. print(results\_garch\_1\_1.aic, results\_garch\_1\_1.bic)
38. model\_garch\_2\_1 = arch\_model(return\_data.daily\_return[1:],vol='GARCH',p=2,q=1)
39. results\_garch\_2\_1.resid.plot(figsize = (20, 4), title = 'garch\_Residual')
40. plot\_acf(results\_garch\_2\_1.resid)
41. plot\_pacf(results\_garch\_2\_1.resid)
42. plt.figure(figsize=(20,12))
43. fig = qqplot(results\_garch\_2\_1.resid, line='q', fit=**True**)
44. print(results\_garch\_2\_1.aic, results\_garch\_2\_1.bic)
45. model\_garch\_1\_2 = arch\_model(return\_data.daily\_return[1:],mean='Constant',vol='GARCH',p=1,q=2)
46. plt.show()
47. results\_garch\_1\_2 = model\_garch\_1\_2.fit()
48. results\_garch\_1\_2.summary()
49. results\_garch\_1\_2.resid.plot(figsize = (20, 4), title = 'garch\_Residual')
50. plot\_acf(results\_garch\_1\_2.resid)
51. plot\_pacf(results\_garch\_1\_2.resid)
52. plt.figure(figsize=(20,12))
53. plt.show()
54. fig = qqplot(results\_garch\_1\_2.resid, line='q', fit=**True**)
55. print(results\_garch\_1\_2.aic, results\_garch\_1\_2.bic)
56. model\_garch\_2\_2 = arch\_model(return\_data.daily\_return[1:],mean='Constant',vol='GARCH',p=2,q=2)
57. results\_garch\_2\_2 = model\_garch\_2\_2.fit()
58. results\_garch\_2\_2.summary()
59. results\_garch\_2\_2.resid.plot(figsize = (20, 4), title = 'garch\_Residual')
60. plot\_acf(results\_garch\_2\_2.resid)
61. plot\_pacf(results\_garch\_2\_2.resid)
62. plt.figure(figsize=(20,12))
63. plt.show()
64. fig = qqplot(results\_garch\_2\_2.resid, line='q', fit=**True**)
65. print(results\_garch\_2\_2.aic, results\_garch\_2\_2.bic)
66. model\_garch\_3\_2 = arch\_model(return\_data.daily\_return[1:],mean='Constant',vol='GARCH',p=3,q=2)
67. results\_garch\_3\_2 = model\_garch\_3\_2.fit()
68. results\_garch\_3\_2.summary()
69. results\_garch\_3\_2.resid.plot(figsize = (20, 4), title = 'garch\_Residual')
70. plot\_acf(results\_garch\_3\_2.resid)
71. plot\_pacf(results\_garch\_3\_2.resid)
72. plt.figure(figsize=(20,12))
73. plt.show()
74. fig = qqplot(results\_garch\_3\_2.resid, line='q', fit=**True**)
75. print(results\_garch\_3\_2.aic, results\_garch\_3\_2.bic)
76. best\_garch = arch\_model(return\_data.daily\_return[1:], vol= 'Garch').fit()
77. **for** p\_num **in** range(1,4):
78. **for** q\_num **in** range(1,4):
79. temp\_garch = arch\_model(return\_data.daily\_return[1:], vol= 'Garch', p = p\_num , q = q\_num).fit()
80. **if** temp\_garch.bic < best\_garch.bic:
81. best\_garch = temp\_garch
82. print('best\_p,q',best\_garch)
83. garch\_fitted = model\_garch\_2\_1.fit()
84. forecasts = garch\_fitted.forecast(horizon=30, method='simulation', simulations=1000)
85. sims = forecasts.simulations
86. lines = plt.plot(sims.values[-1,:,:].T, color='blue', alpha=0.01)
87. lines[0].set\_label('Simulated paths')
88. plt.show()
89. print(np.percentile(sims.values[-1,:,-1].T,5))
90. plt.hist(sims.values[-1, :,-1],bins=50)
91. plt.title('Distribution of Returns')
92. plt.show()