Time Series Modeling for Google Stock Price

Deng Qisheng

2021/05/30

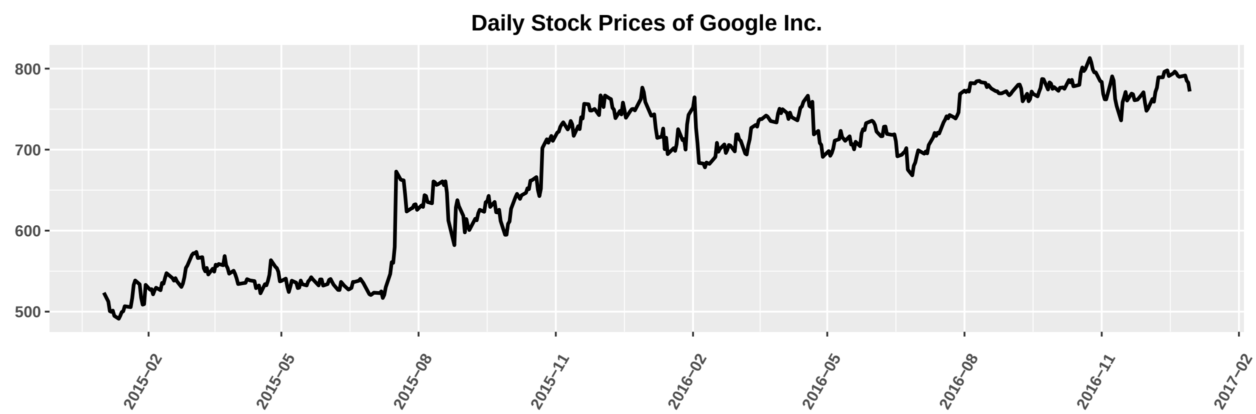
1 Introduction

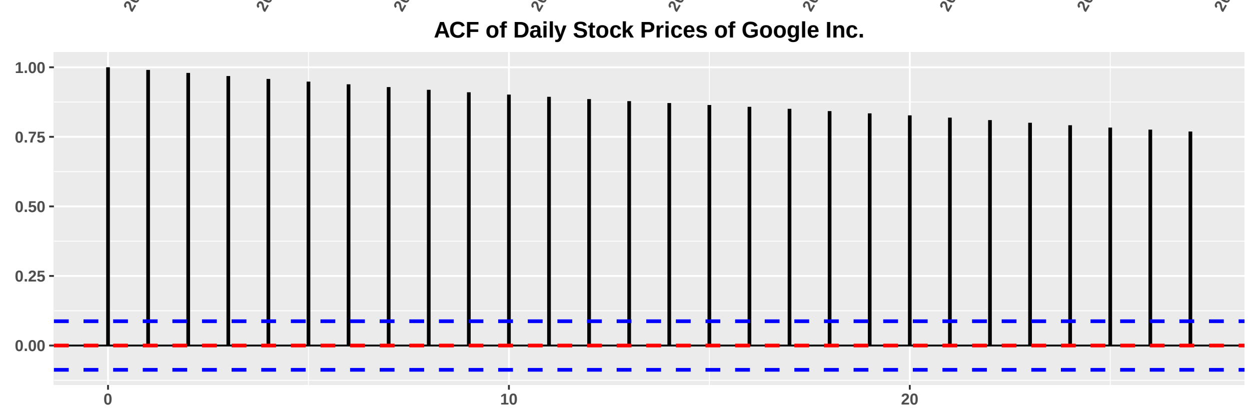
This paper aims to model stock price of Google Inc. using ARIMA(P, I, Q) + GARCH(p, q) models. Stock price data of Google Inc. from 2015/01/01 to 2016/12/31, downloaded from Yahoo! Finance are used to analyze in this project. Before we analyze the price data, environment should be set up in R.

2 Descriptive Data Analysis

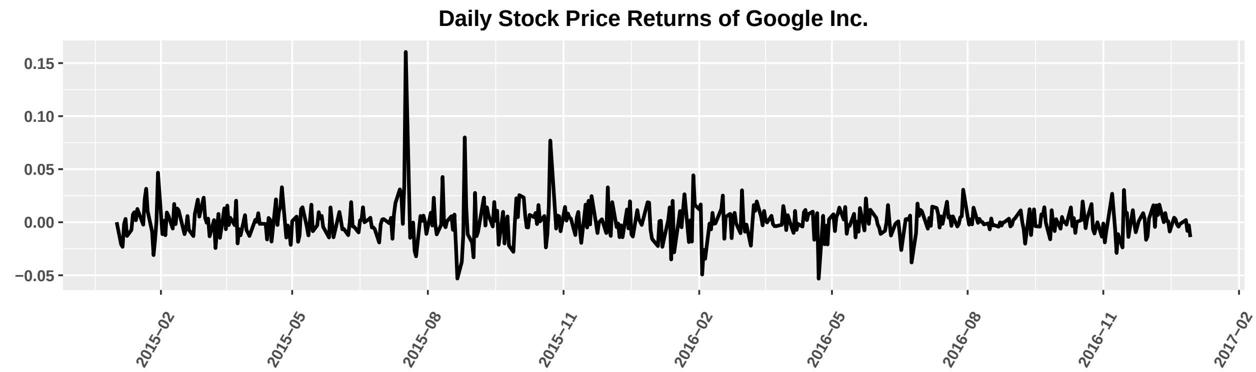
Data preparation is the process of cleaning and transforming raw data prior to processing, analysis and modeling. In this project, stock price of Google Inc. are key variables to concern about. Load in the dataset from *GOOG.csv*.

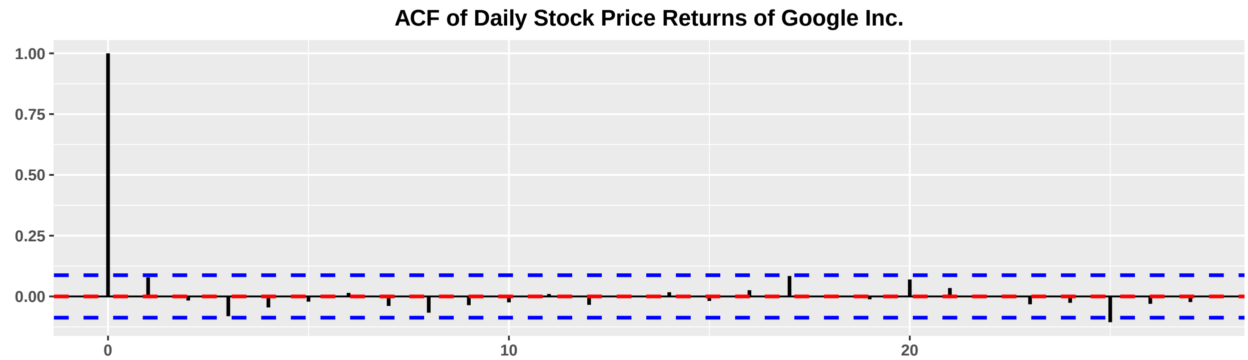
Visualize the price trend and ACF of Google Inc.. The stock price trend seems not to be a stationary process. The ACF plot shows a slow decay which imply that this trend is non-stationary.



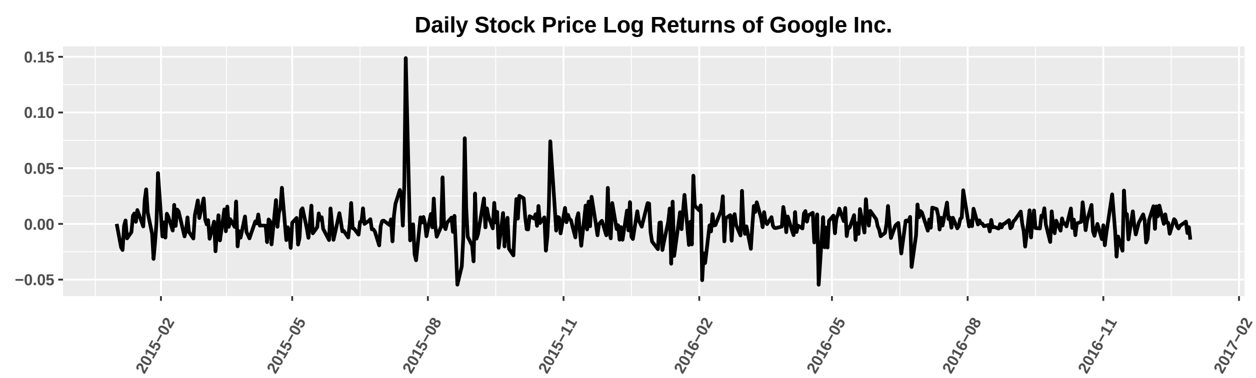


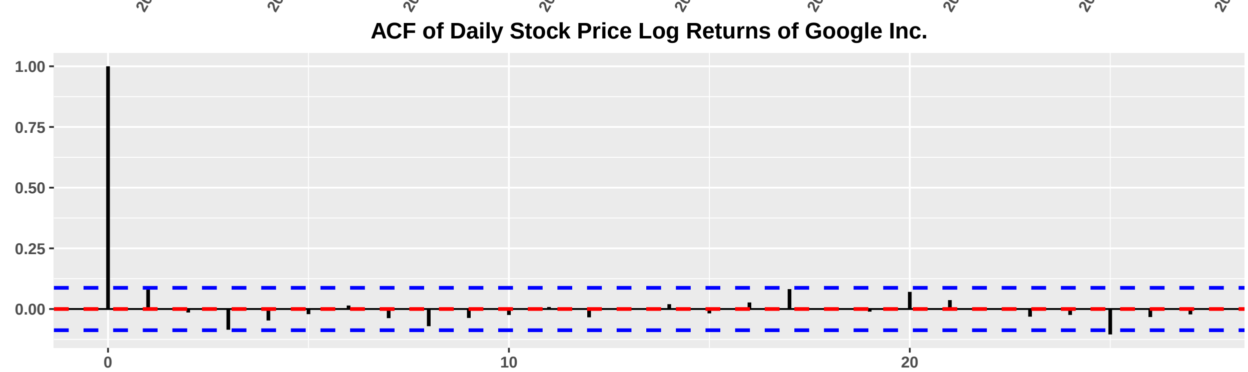
Next step, we are going to verify the stationarity of price return. The stock price return trend seems to be a stationary process. The ACF plot decays to zero quickly which imply that this trend is stationary.





Finally, we are interested in the log returns trend of stock price. The stock price log return trend seems to be a stationary process. The ACF plot decays to zero quickly which imply that this trend is stationary.





3 Modeling

We are going to apply time series model to stock price data. From the descriptive data analysis, we find that the trend of price returns are stationary similarly. In ARIMA(P, I, Q) + GARCH(p, q) models, return data means $I = 1$ for stock price data. Hence, we consider the ARMA(P, Q) + GARCH(p, q) models on returns data, which means that ARIMA(P, 1, Q) + GARCH(p, q) models for stock price data.

3.1 ARIMA(1, 1, 1) + GARCH(1, 1) model

Residual with normal distribution

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(1, 1) + garch(1, 1), data = Price$Return,
##     cond.dist = "QMLE", trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ arma(1, 1) + garch(1, 1)
## <environment: 0x000000001eea3340>
##  [data = Price$Return]
##
## Conditional Distribution:
##  QMLE
##
## Coefficient(s):
##          mu          ar1          ma1        omega       alpha1        beta1
##  1.6338e-04   7.7941e-01  -8.2510e-01   4.2513e-05   3.8524e-01   5.1307e-01
##
## Std. Errors:
##  robust
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## mu      1.634e-04   1.258e-04    1.299   0.1941
## ar1     7.794e-01   8.285e-02    9.408  < 2e-16 ***
## ma1    -8.251e-01   6.441e-02  -12.810  < 2e-16 ***
## omega   4.251e-05   1.789e-05    2.376   0.0175 *
## alpha1  3.852e-01   1.961e-01    1.965   0.0494 *
## beta1   5.131e-01   1.288e-01    3.984 6.78e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  1425.851    normalized:  2.82907
##
## Description:
##  Sun May 30 23:59:06 2021 by user: DengQisheng
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  439.9687  0
```

```
##  Shapiro-Wilk Test  R    W      0.952816  1.321038e-11
##  Ljung-Box Test     R    Q(10)  11.15704  0.3454151
##  Ljung-Box Test     R    Q(15)  13.91114  0.5322788
##  Ljung-Box Test     R    Q(20)  19.98453  0.458898
##  Ljung-Box Test     R^2  Q(10)  4.999668  0.8912002
##  Ljung-Box Test     R^2  Q(15)  6.971155  0.9584452
##  Ljung-Box Test     R^2  Q(20)  16.20246  0.7039875
##  LM Arch Test       R    TR^2   5.594313  0.9351374
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -5.634329 -5.584061 -5.634609 -5.614611
```

Residual with t distribution

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(1, 1) + garch(1, 1), data = Price$Return,
##     cond.dist = "std", trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ arma(1, 1) + garch(1, 1)
## <environment: 0x00000000130467f8>
##  [data = Price$Return]
##
## Conditional Distribution:
##  std
##
## Coefficient(s):
##          mu          ar1          ma1        omega       alpha1        beta1
##  4.4543e-04  -2.9637e-02   5.7523e-02   1.3978e-06   1.2804e-02   9.7797e-01
##       shape
##  3.9517e+00
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## mu      4.454e-04   5.972e-04    0.746    0.456
## ar1    -2.964e-02   5.962e-01   -0.050    0.960
## ma1     5.752e-02   5.954e-01    0.097    0.923
## omega   1.398e-06   1.122e-06    1.246    0.213
## alpha1  1.280e-02   6.006e-03    2.132    0.033 *
## beta1   9.780e-01   9.246e-03  105.777  < 2e-16 ***
## shape   3.952e+00   6.376e-01    6.198 5.73e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  1463.404    normalized:  2.90358
##
```

```
## Description:
##  Sun May 30 23:59:06 2021 by user: DengQisheng
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  22386.9   0
##  Shapiro-Wilk Test  R    W      0.8359226 0
##  Ljung-Box Test     R    Q(10)  8.108578  0.618232
##  Ljung-Box Test     R    Q(15)  9.4146    0.8548613
##  Ljung-Box Test     R    Q(20)  13.7032   0.8452201
##  Ljung-Box Test     R^2  Q(10)  2.507814  0.9907614
##  Ljung-Box Test     R^2  Q(15)  2.901152  0.9996735
##  Ljung-Box Test     R^2  Q(20)  3.362135  0.9999891
##  LM Arch Test       R    TR^2   2.491675  0.9981921
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -5.779381 -5.720734 -5.779760 -5.756376
```

From the result above, we find that ARMA(1, 1) model with normal residuals is suitable and ARMA(1, 1) model with t residuals is not suitable. We find that beta1 is significant but alpha is not. Hence, we should consider more GARCH models.

3.2 ARIMA(1, 1, 1) + GARCH(1, 2) model

Residual with normal distribution

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(1, 1) + garch(1, 2), data = Price$Return,
##     cond.dist = "QMLE", trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ arma(1, 1) + garch(1, 2)
## <environment: 0x000000001fa24418>
##  [data = Price$Return]
##
## Conditional Distribution:
##  QMLE
##
## Coefficient(s):
##         mu         ar1         ma1       omega      alpha1       beta1
## 8.1088e-04  1.0385e-02  2.5805e-02  2.9171e-05  4.1908e-01  6.2135e-03
##      beta2
## 5.4070e-01
##
## Std. Errors:
##  robust
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     8.109e-04   5.584e-04    1.452  0.14648
## ar1    1.039e-02   2.085e-01    0.050  0.96028
## ma1    2.580e-02   2.028e-01    0.127  0.89873
## omega  2.917e-05   1.576e-05    1.851  0.06422 .
## alpha1 4.191e-01   1.611e-01    2.601  0.00929 **
## beta1  6.214e-03   4.337e-02    0.143  0.88607
## beta2  5.407e-01   1.380e-01    3.918 8.91e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  1430.521    normalized:  2.838335
##
## Description:
##  Sun May 30 23:59:06 2021 by user: DengQisheng
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  260.0308  0
##  Shapiro-Wilk Test  R    W      0.9637233 8.041766e-10
##  Ljung-Box Test     R    Q(10)  10.55774  0.3929946
##  Ljung-Box Test     R    Q(15)  13.00085  0.6022322
##  Ljung-Box Test     R    Q(20)  18.5939   0.5483465
```

```
##  Ljung-Box Test     R^2  Q(10)  5.765848  0.834537
##  Ljung-Box Test     R^2  Q(15)  7.864265  0.929109
##  Ljung-Box Test     R^2  Q(20)  15.73327  0.7330278
##  LM Arch Test       R    TR^2   7.209486  0.8434649
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -5.648893 -5.590246 -5.649272 -5.625888
```

Residual with t distribution

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(1, 1) + garch(1, 2), data = Price$Return,
##     cond.dist = "std", trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ arma(1, 1) + garch(1, 2)
## <environment: 0x0000000012390d58>
##  [data = Price$Return]
##
## Conditional Distribution:
##  std
##
## Coefficient(s):
##          mu          ar1          ma1        omega       alpha1        beta1
##  4.4597e-04  -3.1040e-02   6.0773e-02   2.4256e-06   2.3528e-02   1.6115e-01
##       beta2        shape
##  7.9920e-01   3.9473e+00
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## mu      4.460e-04   5.948e-04    0.750   0.4534
## ar1    -3.104e-02   5.671e-01   -0.055   0.9563
## ma1     6.077e-02   5.655e-01    0.107   0.9144
## omega   2.426e-06   2.293e-06    1.058   0.2901
## alpha1  2.353e-02   1.264e-02    1.861   0.0628 .
## beta1   1.612e-01   4.520e-01    0.357   0.7214
## beta2   7.992e-01   4.486e-01    1.782   0.0748 .
## shape   3.947e+00   6.446e-01    6.124 9.12e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  1463.744    normalized:  2.904253
##
## Description:
##  Sun May 30 23:59:07 2021 by user: DengQisheng
##
```

```
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  17959.81  0
##  Shapiro-Wilk Test  R    W      0.8472148 0
##  Ljung-Box Test     R    Q(10)  8.433056  0.5866149
##  Ljung-Box Test     R    Q(15)  9.780355  0.8333239
##  Ljung-Box Test     R    Q(20)  14.13546  0.8235617
##  Ljung-Box Test     R^2  Q(10)  2.962298  0.9822974
##  Ljung-Box Test     R^2  Q(15)  3.409009  0.9991207
##  Ljung-Box Test     R^2  Q(20)  3.929392  0.9999598
##  LM Arch Test       R    TR^2   2.927314  0.9960358
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -5.776761 -5.709736 -5.777254 -5.750469
```

From the result above, we find that GARCH(1, 2) models are not appropriate choices.

3.3 ARIMA(1, 1, 1) + GARCH(2, 1) model

Residual with normal distribution

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(1, 1) + garch(2, 1), data = Price$Return,
##     cond.dist = "QMLE", trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ arma(1, 1) + garch(2, 1)
## <environment: 0x0000000020954b40>
##  [data = Price$Return]
##
## Conditional Distribution:
##  QMLE
##
## Coefficient(s):
##          mu          ar1          ma1        omega       alpha1       alpha2
##  0.00016513   0.77789417  -0.82418239   0.00004316   0.38818749   0.00000001
##       beta1
##  0.50748505
##
## Std. Errors:
##  robust
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## mu      1.651e-04   1.295e-04    1.275  0.20221
## ar1     7.779e-01   8.233e-02    9.448  < 2e-16 ***
## ma1    -8.242e-01   6.397e-02  -12.885  < 2e-16 ***
## omega   4.316e-05   1.627e-05    2.653  0.00799 **
## alpha1  3.882e-01   2.447e-01    1.587  0.11261
## alpha2  1.000e-08   1.582e-01    0.000  1.00000
## beta1   5.075e-01   8.706e-02    5.829 5.58e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  1426.059    normalized:  2.829483
##
## Description:
##  Sun May 30 23:59:07 2021 by user: DengQisheng
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  440.6234  0
##  Shapiro-Wilk Test  R    W      0.9527424 1.287747e-11
##  Ljung-Box Test     R    Q(10)  11.06716  0.3523096
##  Ljung-Box Test     R    Q(15)  13.78627  0.5417968
##  Ljung-Box Test     R    Q(20)  19.97254  0.4596485
```

```
##  Ljung-Box Test     R^2  Q(10)  4.981405  0.8924167
##  Ljung-Box Test     R^2  Q(15)  6.991961  0.9578724
##  Ljung-Box Test     R^2  Q(20)  16.23585  0.701889
##  LM Arch Test       R    TR^2   5.648541  0.9327528
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -5.631187 -5.572540 -5.631566 -5.608182
```

Residual with t distribution

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(1, 1) + garch(2, 1), data = Price$Return,
##     cond.dist = "std", trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ arma(1, 1) + garch(2, 1)
## <environment: 0x00000000203ea888>
##  [data = Price$Return]
##
## Conditional Distribution:
##  std
##
## Coefficient(s):
##         mu         ar1         ma1       omega      alpha1      alpha2
## 5.8443e-04  3.1094e-03  1.5423e-02  4.0606e-05  9.1223e-02  9.3233e-02
##      beta1       shape
## 6.5184e-01  3.9259e+00
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     5.844e-04   7.277e-04    0.803 0.421924
## ar1    3.109e-03   9.298e-01    0.003 0.997332
## ma1    1.542e-02   9.359e-01    0.016 0.986853
## omega  4.061e-05   2.671e-05    1.520 0.128445
## alpha1 9.122e-02   7.742e-02    1.178 0.238709
## alpha2 9.323e-02   1.072e-01    0.870 0.384572
## beta1  6.518e-01   1.726e-01    3.776 0.000159 ***
## shape  3.926e+00   7.247e-01    5.417 6.05e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  1462.557    normalized:  2.901899
##
## Description:
##  Sun May 30 23:59:07 2021 by user: DengQisheng
##
```

```
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  2345.948  0
##  Shapiro-Wilk Test  R    W      0.9163588 4.422749e-16
##  Ljung-Box Test     R    Q(10)  10.39008  0.4069621
##  Ljung-Box Test     R    Q(15)  12.83274  0.6152155
##  Ljung-Box Test     R    Q(20)  18.25106  0.5708739
##  Ljung-Box Test     R^2  Q(10)  3.152463  0.9776048
##  Ljung-Box Test     R^2  Q(15)  4.298024  0.9965857
##  Ljung-Box Test     R^2  Q(20)  8.463946  0.9883258
##  LM Arch Test       R    TR^2   3.412378  0.991866
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -5.772052 -5.705027 -5.772545 -5.745760
```

From the result above, we find that GARCH(2, 1) models also are not appropriate choices.

4 Conclusion

In this project, we build time series models for stock price data of Google Inc. and we find the best model as follow.

$$\begin{cases} \Delta X_t = 0.000163 + 0.779(\Delta X_{t-1} - 0.000163) + a_t + 0.825 a_{t-1} \\ a_t = \sigma_t \epsilon_t, \epsilon_t \sim \mathcal{N}(0,1) \\ \sigma_t^2 = 0.0000425 + 0.385 a_{t-1}^2 + 0.513 \epsilon_{t-1}^2 \end{cases}$$

5 Appendix

The appendix shows the entire codes for this project.

5.1 Setting

```
model.1.QMLE <- garchFit(formula=~arma(1, 1)+garch(1, 1), data=Price$Return,
                         cond.dist="QMLE", trace=FALSE)

summary(model.1.QMLE)
```

5.2 Descriptive Data Analysis

```
Price.trend <- ggplot(data=Price) +
    geom_line(aes(x=Date, y=Close), size=1, group=0) +
    labs(title="Daily Stock Prices of Google Inc.", x=NULL, y=NULL) +
    theme(plot.title=element_text(hjust=0.5, face="bold"),
          axis.text=element_text(face="bold"),
          axis.text.x=element_text(angle=60, vjust=0.5, hjust=0.7),
          legend.position="bottom", legend.title=element_blank(),
          legend.text=element_text(face="bold")) +
    scale_x_date(date_labels="%Y-%m", date_breaks="3 month")
```

```r
Price.acf.ci.line <- qnorm((1 - 0.95) / 2) / sqrt(length(Price$Close))
Price.acf <- ggplot(data=with(acf(Price$Close, plot = FALSE), data.frame(lag, acf)),
                    mapping=aes(x=lag, y=acf)) +
    geom_hline(aes(yintercept = 0)) +
    geom_segment(mapping=aes(xend=lag, yend=0), size=1) +
    geom_hline(yintercept=-Price.acf.ci.line, color="blue", linetype="dashed", size=1) +
    geom_hline(yintercept=Price.acf.ci.line, color="blue", linetype="dashed", size=1) +
    geom_hline(yintercept=0, color="red", linetype="dashed", size=1) +
    labs(title="ACF of Daily Stock Prices of Google Inc.", x=NULL, y=NULL) +
    theme(plot.title=element_text(hjust=0.5, face="bold"),
          axis.text=element_text(face="bold"),
          legend.position="bottom", legend.title=element_blank(),
          legend.text=element_text(face="bold"))

grid.arrange(Price.trend, Price.acf, nrow=2)
```

```r
Return.trend <- ggplot(data=Price) +
    geom_line(aes(x=Date, y=Return), size=1, group=0) +
    labs(title="Daily Stock Price Returns of Google Inc.", x=NULL, y=NULL) +
    theme(plot.title=element_text(hjust=0.5, face="bold"),
          axis.text=element_text(face="bold"),
          axis.text.x=element_text(angle=60, vjust=0.5, hjust=0.7),
          legend.position="bottom", legend.title=element_blank(),
          legend.text=element_text(face="bold")) +
    scale_x_date(date_labels="%Y-%m", date_breaks="3 month")

Return.acf.ci.line <- qnorm((1 - 0.95) / 2) / sqrt(length(Price$Return))
Return.acf <- ggplot(data=with(acf(Price$Return, plot = FALSE), data.frame(lag, acf)),
                     mapping=aes(x=lag, y=acf)) +
    geom_hline(aes(yintercept = 0)) +
    geom_segment(mapping=aes(xend=lag, yend=0), size=1) +
    geom_hline(yintercept=-Return.acf.ci.line, color="blue", linetype="dashed", size=1) +
    geom_hline(yintercept=Return.acf.ci.line, color="blue", linetype="dashed", size=1) +
    geom_hline(yintercept=0, color="red", linetype="dashed", size=1) +
    labs(title="ACF of Daily Stock Price Returns of Google Inc.", x=NULL, y=NULL) +
    theme(plot.title=element_text(hjust=0.5, face="bold"),
          axis.text=element_text(face="bold"),
          legend.position="bottom", legend.title=element_blank(),
          legend.text=element_text(face="bold"))

grid.arrange(Return.trend, Return.acf, nrow=2)
```

```r
Log.Return.trend <- ggplot(data=Price) +
    geom_line(aes(x=Date, y=Log.Return), size=1, group=0) +
    labs(title="Daily Stock Price Log Returns of Google Inc.", x=NULL, y=NULL) +
    theme(plot.title=element_text(hjust=0.5, face="bold"),
          axis.text=element_text(face="bold"),
          axis.text.x=element_text(angle=60, vjust=0.5, hjust=0.7),
          legend.position="bottom", legend.title=element_blank(),
          legend.text=element_text(face="bold")) +
    scale_x_date(date_labels="%Y-%m", date_breaks="3 month")

Log.Return.acf.ci.line <- qnorm((1 - 0.95) / 2) / sqrt(length(Price$Log.Return))
Log.Return.acf <- ggplot(data=with(acf(Price$Log.Return, plot = FALSE), data.frame(lag, acf)),
```

```r
                    mapping=aes(x=lag, y=acf)) +
    geom_hline(aes(yintercept = 0)) +
    geom_segment(mapping=aes(xend=lag, yend=0), size=1) +
    geom_hline(yintercept=-Log.Return.acf.ci.line, color="blue", linetype="dashed", size=1) +
    geom_hline(yintercept=Log.Return.acf.ci.line, color="blue", linetype="dashed", size=1) +
    geom_hline(yintercept=0, color="red", linetype="dashed", size=1) +
    labs(title="ACF of Daily Stock Price Log Returns of Google Inc.", x=NULL, y=NULL) +
    theme(plot.title=element_text(hjust=0.5, face="bold"),
          axis.text=element_text(face="bold"),
          legend.position="bottom", legend.title=element_blank(),
          legend.text=element_text(face="bold"))

grid.arrange(Log.Return.trend, Log.Return.acf, nrow=2)
```

5.3 Modeling

```r
model.1.QMLE <- garchFit(formula=~arma(1, 1)+garch(1, 1), data=Price$Return,
                         cond.dist="QMLE", trace=FALSE)

summary(model.1.QMLE)
```

```r
model.1.std <- garchFit(formula=~arma(1, 1)+garch(1, 1), data=Price$Return,
                        cond.dist="std", trace=FALSE)

summary(model.1.std)
```

```r
model.2.QMLE <- garchFit(formula=~arma(1, 1)+garch(1, 2), data=Price$Return,
                         cond.dist="QMLE", trace=FALSE)

summary(model.2.QMLE)
```

```r
model.2.std <- garchFit(formula=~arma(1, 1)+garch(1, 2), data=Price$Return,
                        cond.dist="std", trace=FALSE)

summary(model.2.std)
```

```r
model.3.QMLE <- garchFit(formula=~arma(1, 1)+garch(2, 1), data=Price$Return,
                         cond.dist="QMLE", trace=FALSE)

summary(model.3.QMLE)
```

```r
model.3.std <- garchFit(formula=~arma(1, 1)+garch(2, 1), data=Price$Return,
                        cond.dist="std", trace=FALSE)

summary(model.3.std)
```