a3

Q1

```
data_pre<-read.csv(file="IBM.csv", header=TRUE, sep=",")
data_pre$Date <- as.Date(data_pre$Date, format = "%Y-%m-%d")
data<-data_pre[data_pre$Date >= "2015-01-01" & data_pre$Date <= "2018-12-31",]
dim(data)
```

```
## [1] 1006    2
```

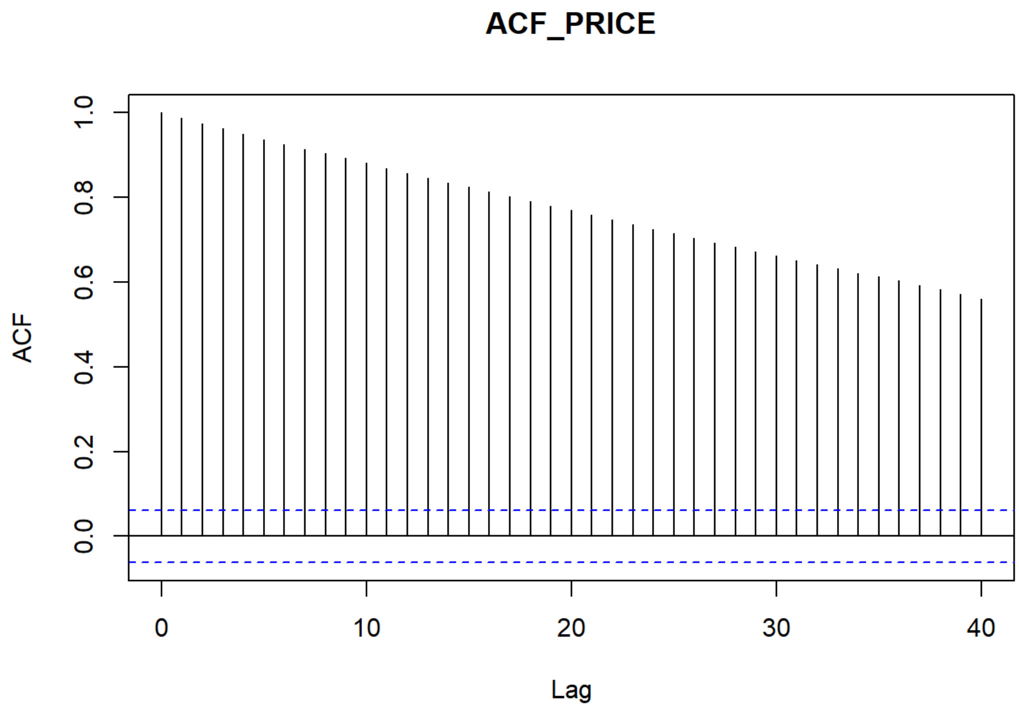
```
tail(data, n=30)
```

```
##            Date Adj.Close
## 1179 2015-02-13  131.6974
## 1180 2015-02-12  130.1539
## 1181 2015-02-11  129.8911
## 1182 2015-02-10  130.1867
## 1183 2015-02-09  127.8795
## 1184 2015-02-06  128.6759
## 1185 2015-02-05  128.7498
## 1186 2015-02-04  127.9753
## 1187 2015-02-03  129.2064
## 1188 2015-02-02  126.1000
## 1189 2015-01-30  124.9993
## 1190 2015-01-29  126.7686
## 1191 2015-01-28  123.5643
## 1192 2015-01-27  125.2928
## 1193 2015-01-26  127.4861
## 1194 2015-01-23  127.0865
## 1195 2015-01-22  126.6952
## 1196 2015-01-21  124.0046
## 1197 2015-01-20  127.9671
## 1198 2015-01-16  128.1220
## 1199 2015-01-15  126.0266
## 1200 2015-01-14  127.0295
## 1201 2015-01-13  127.8530
## 1202 2015-01-12  127.5513
## 1203 2015-01-09  129.7283
## 1204 2015-01-08  129.1657
## 1205 2015-01-07  126.4180
## 1206 2015-01-06  127.2496
## 1207 2015-01-05  130.0544
## 1208 2015-01-02  132.1335
```

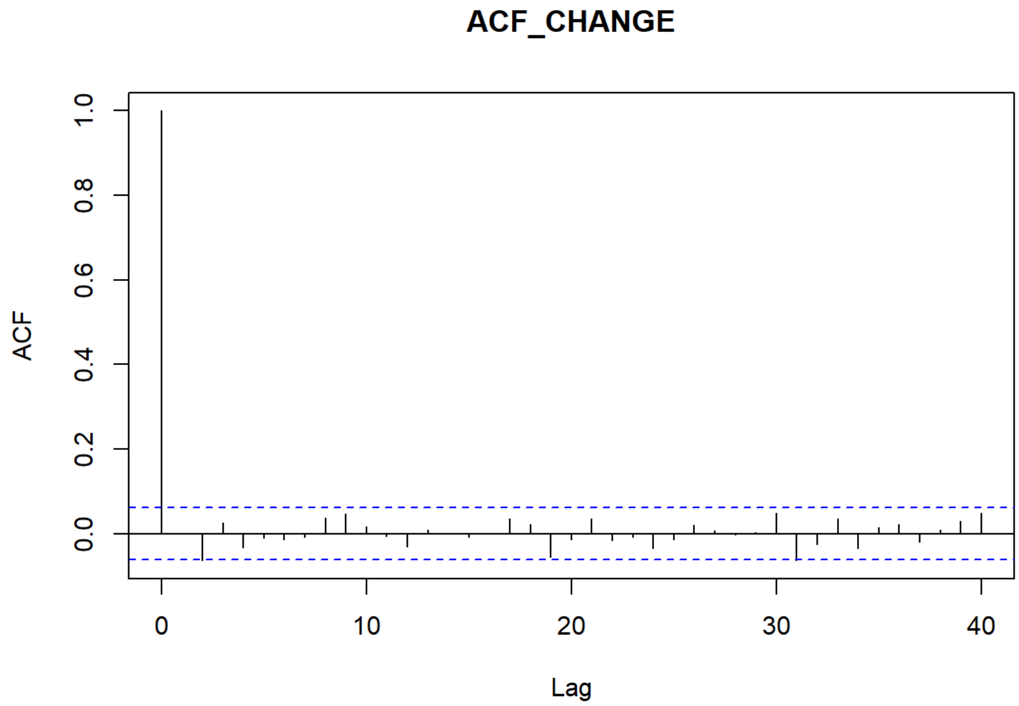
From the R output above, length of data is 1006.

Q2

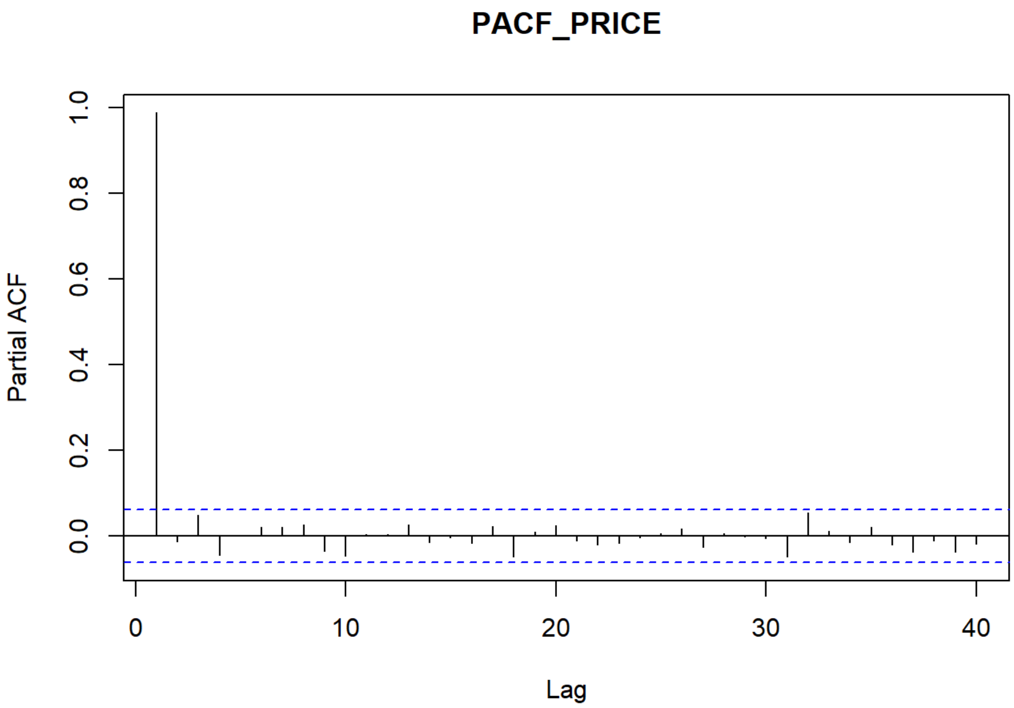
```
ts = as.ts(data$Adj.Close, start=2015, frequency=12)
acf(ts, lag.max=40, main="ACF_PRICE")
```



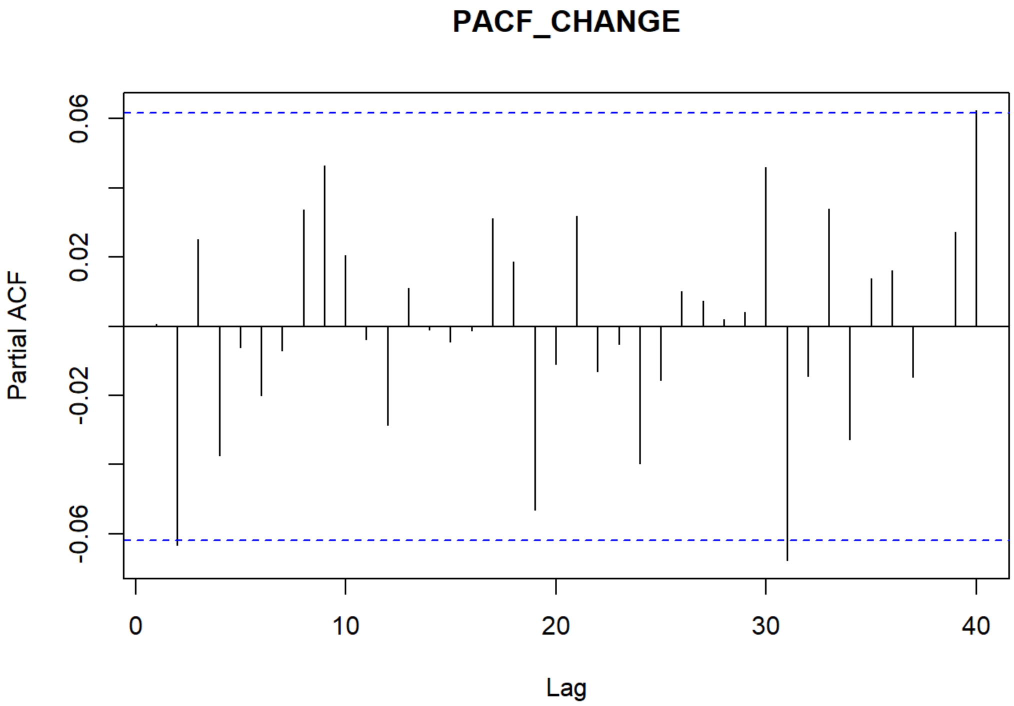

```
acf(diff(ts), lag.max=40, main="ACF_CHANGE")
```




```
pacf(ts, lag.max=40, main="PACF_PRICE")
```




```
pacf(diff(ts), lag.max=40, main="PACF_CHANGE")
```



I would pick p=1 since the ACF_PRICE figure shows the sample autocorrelation function appears to be decreasing exponentially.

Q3

```
Box.test(ts, lag=40, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  ts
## X-squared = 24776, df = 40, p-value < 2.2e-16
```

Null hypothesis of white noise is rejected because of the very small p-value of Box test. At least 1 of the first 40 autocorrelations is nonzero

Q4

```
AR1 = arima(data$Adj.Close, order = c(1,0,0))
print(AR1)
```

```
##
## Call:
## arima(x = data$Adj.Close, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.9905   132.6845
## s.e.  0.0043     5.1762
##
## sigma^2 estimated as 2.915:  log likelihood = -1967.57,  aic = 3941.14
```

```
Box.test(AR1$resid, type = "Ljung", lag = 40, fitdf = 1)
```

```
##
##  Box-Ljung test
##
## data:  AR1$resid
## X-squared = 33.772, df = 39, p-value = 0.7069
```

The large p-value suggests we cannot reject that the residuals are uncorrelated,. Therefore AR1 is a good fit. The model with estimated parameters is: \[Y_t - 132.6845 = 0.9905Y_{t-1} + error\]

Q5

```
adf.test(ts)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts
## Dickey-Fuller = -2.9457, Lag order = 10, p-value = 0.178
## alternative hypothesis: stationary
```

```
pp.test(ts)
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  ts
## Dickey-Fuller Z(alpha) = -13.637, Truncation lag parameter = 7,
## p-value = 0.349
## alternative hypothesis: stationary
```

```
kpss.test(ts)
```

```
## Warning in kpss.test(ts): p-value smaller than printed p-value
```

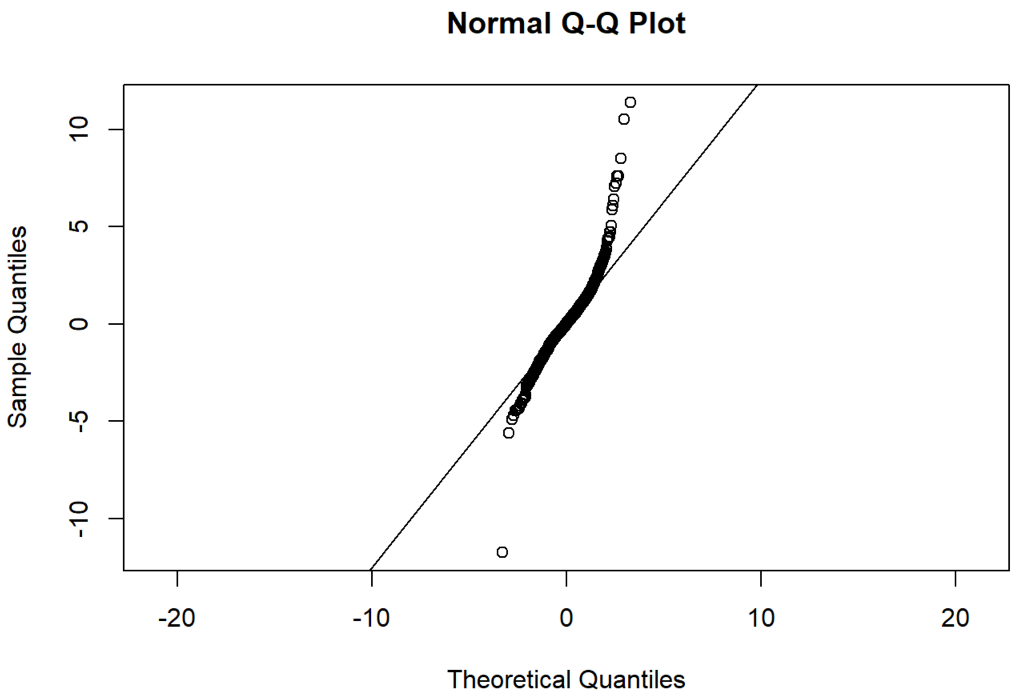
```
##
##  KPSS Test for Level Stationarity
##
## data:  ts
## KPSS Level = 1.9266, Truncation lag parameter = 7, p-value = 0.01
```

The large p-value in PP and ADF tests suggest we reject that data is stationary. The small p-value in KPSS test suggests we do not reject that data is non-stationary.

In conclusion the tests suggest data is non-stationary.

Q6

```
qqnorm(AR1$residuals, asp = 1)
qqline(AR1$residuals, asp=1)
```



From the Normal Q-Q plot we can see the residuals do not follow Gaussian. This concave-convex pattern suggests that it has light tails.

Q7

```
ARIMA = auto.arima(data$Adj.Close, max.p = 5, max.q = 5, ic = "aic")
print(ARIMA)
```

```
## Series: data$Adj.Close
## ARIMA(0,1,0)
##
## sigma^2 estimated as 2.928:  log likelihood=-1965.86
## AIC=3933.72   AICc=3933.73   BIC=3938.63
```

From the R output above, ARIMA(0, 1, 0) is the best model. The model is \[Y_t =Y_{t-1} + error\] It appeas that it is a random walk.

Q8

```
library(MASS)
tfit <- fitdistr(ARIMA$residuals, "t")
```

```
print(tfit)
```

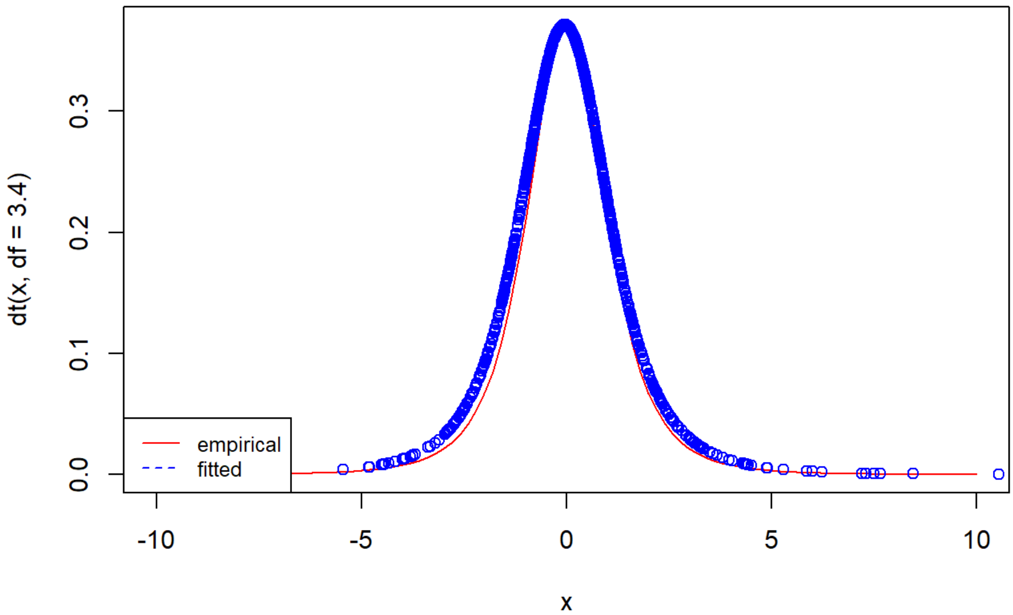
```
##          m              s              df
##   -0.03774174    1.12616528    3.39492443
##  ( 0.04282033) ( 0.04568404) ( 0.39427634)
```

Q9

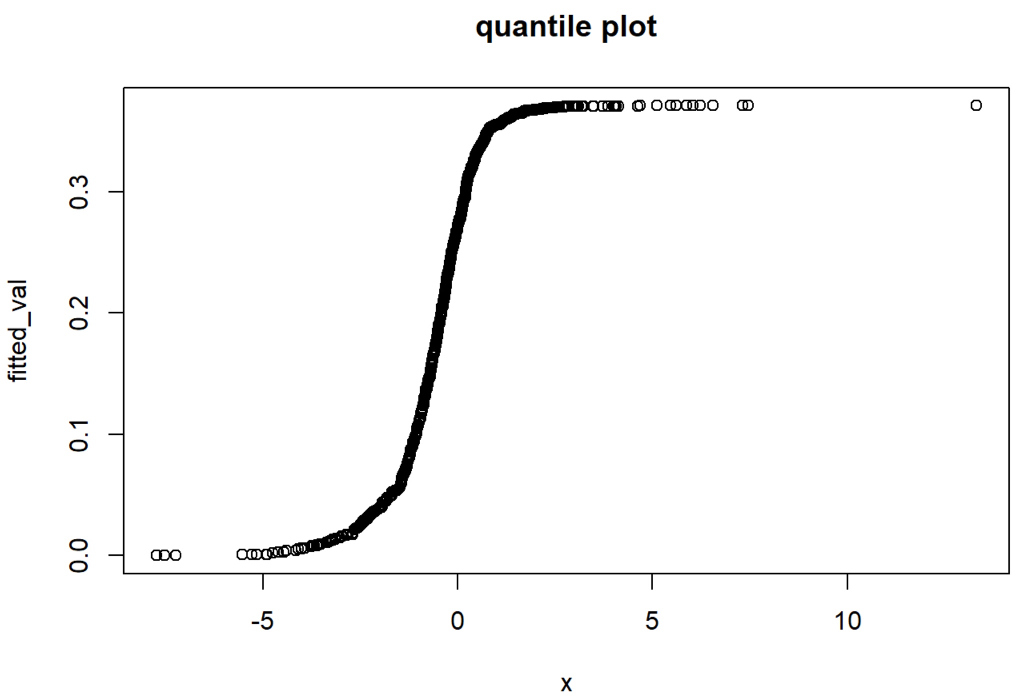
```
x<-rt(1006, df = 3.4)
curve(dt(x, df = 3.4), -10, 10, col = 'red')
res <- ARIMA$residuals

fitted_val <- dt((ARIMA$residuals + 0.03774) / 1.1262, df=3.4)

points(res, fitted_val, col = 'blue')
legend("bottomleft", legend=c("empirical", "fitted"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```




```
qqplot(x,fitted_val, main="quantile plot")
```



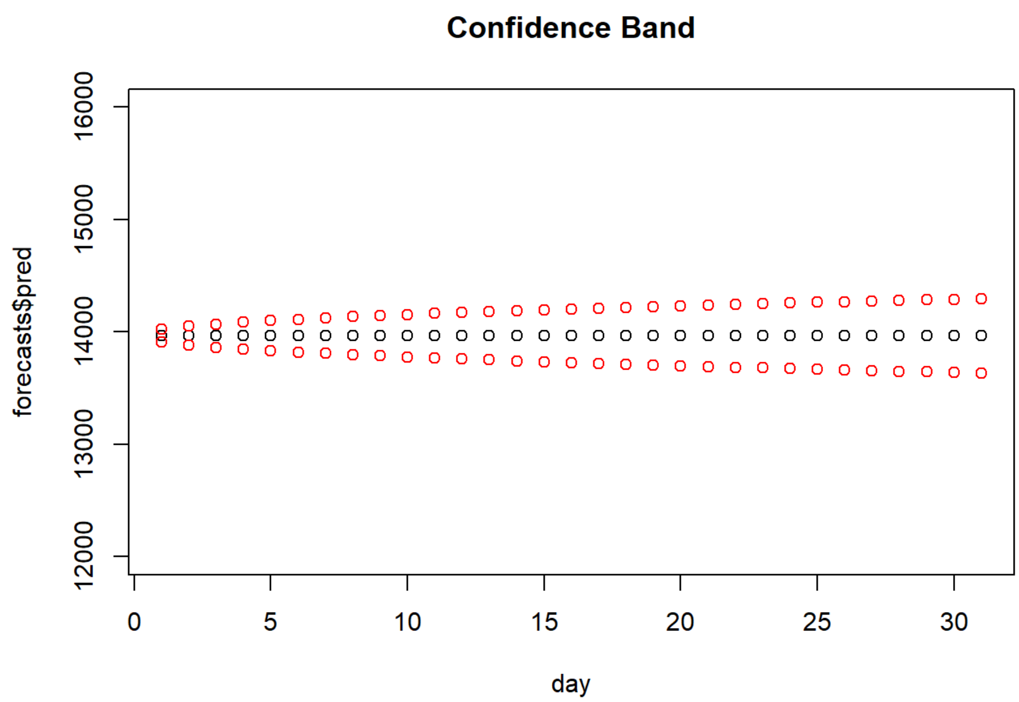
From the density plots and the quantile plot, t-distribution is a good fit.

Q10

```
#model besed resampling
n <- 1000
sim <- rep(0, 1000)
error <- rnorm(n)
sim[1] <- error[1]
for (i in 2:n) {
sim[i] <- 132.6845 + 0.9905*sim[i-1] + error[i]
}
```

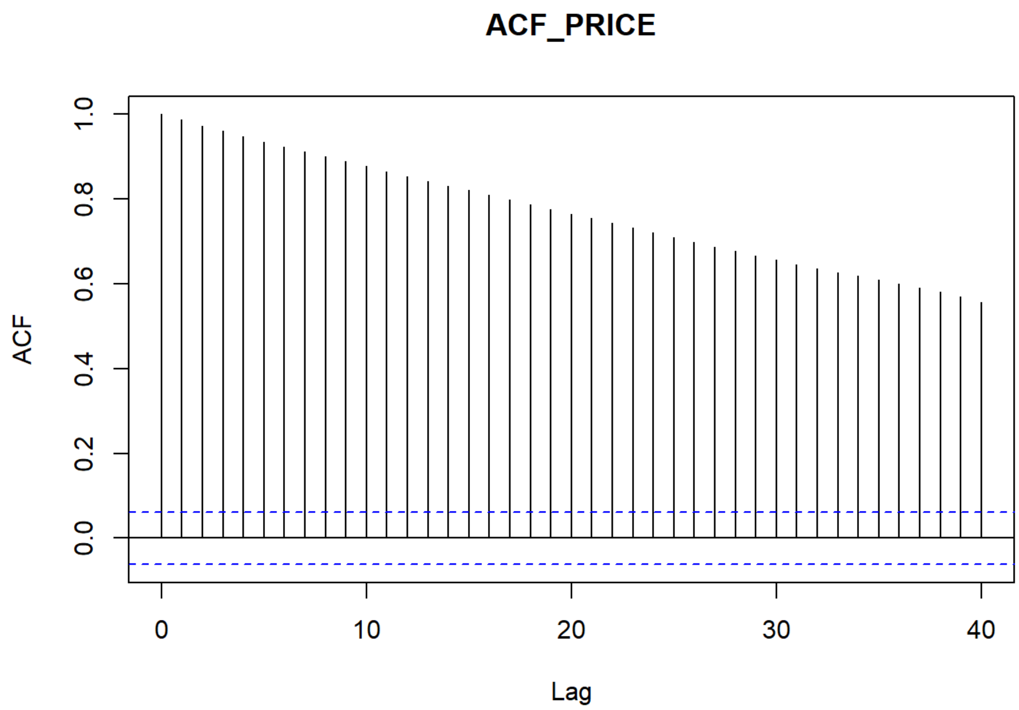
I used the AR1 model for resamping and forecasting. Above is a model based re-sampling of the AR1 model i got in Q4. There are 1000 simulations in total. Next I will forecast on the simulated time series.

```
simFit = arima(sim, order = c(1,0,0))
library(forecast)
forecasts = predict(simFit, 31)
#below is the prediction for 31 days in Jan 2019.
day = seq(1, 31, by=1)
upper <- forecasts$pred + 1.96*forecasts$se
lower <- forecasts$pred - 1.96*forecasts$se
plot(day, forecasts$pred, ylim = c(12000, 16000), main = "Confidence Band")
points(day, upper, col = "red")
points(day, lower, col = "red")
```

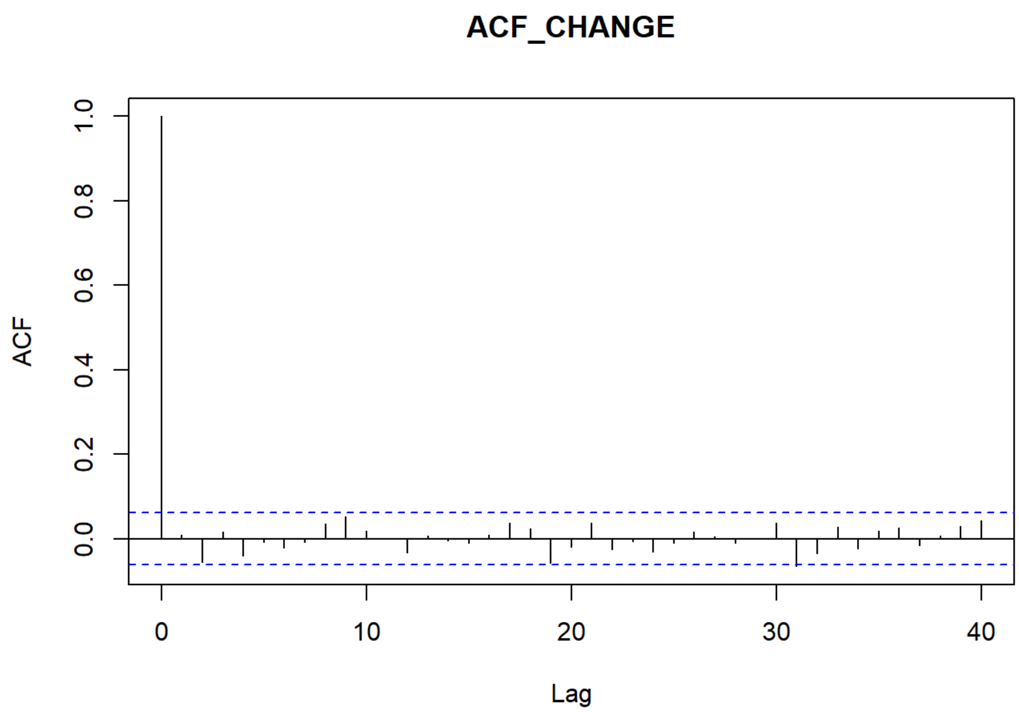


Q11

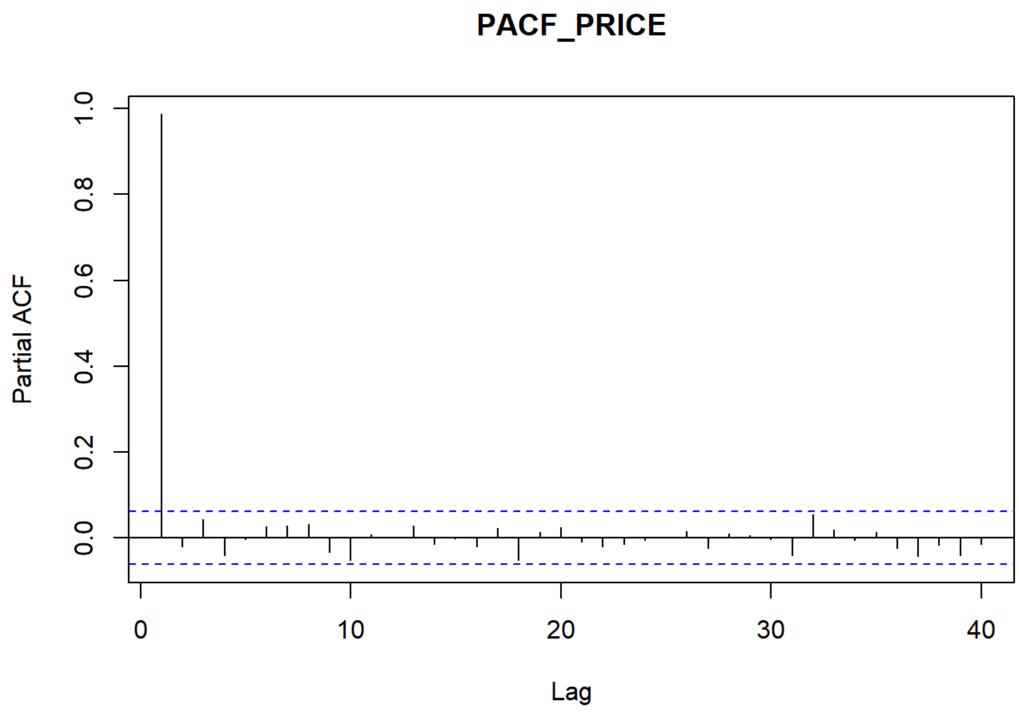
```
ts_log = as.ts(log(data$Adj.Close), start=2015, frequency=12)
acf(ts_log, lag.max=40, main="ACF_PRICE")
```



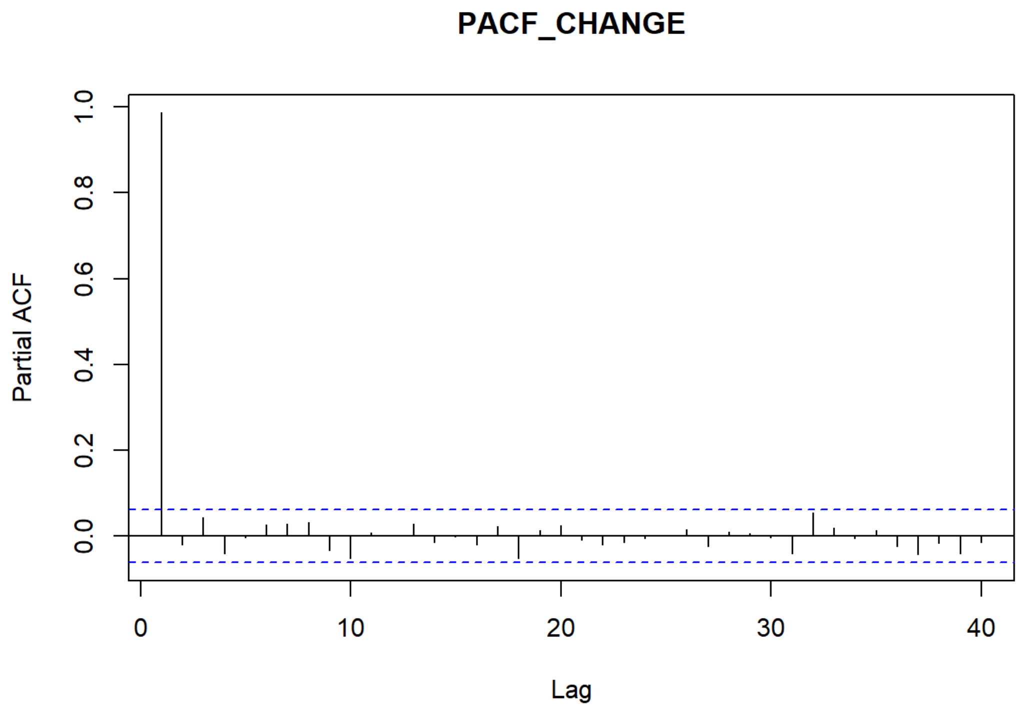
```
acf(diff(ts_log), lag.max=40, main="ACF_CHANGE")
```



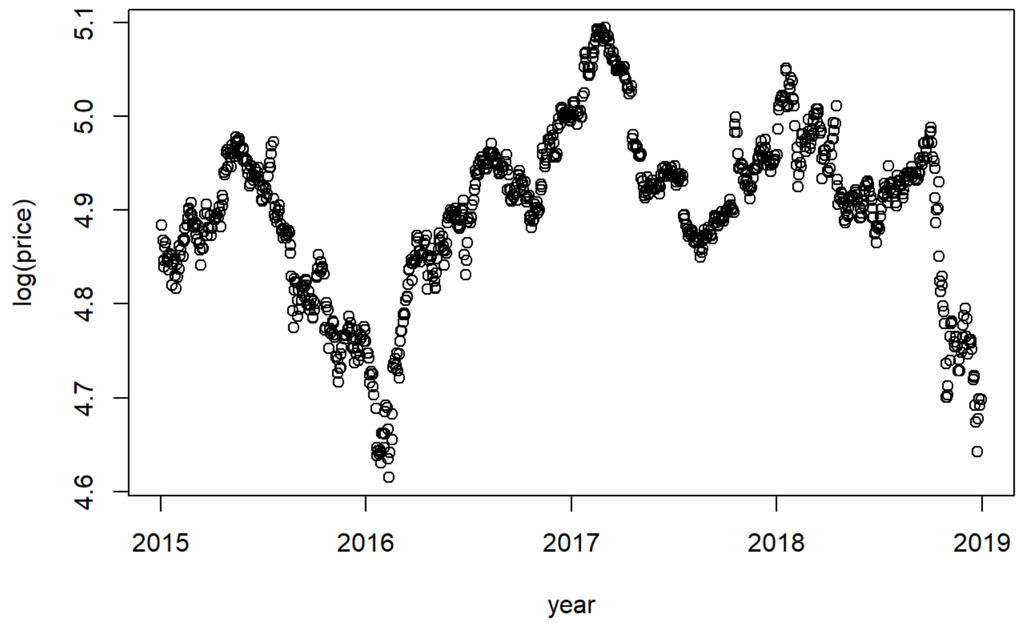

```
pacf(ts_log, lag.max=40, main="PACF_PRICE")
```



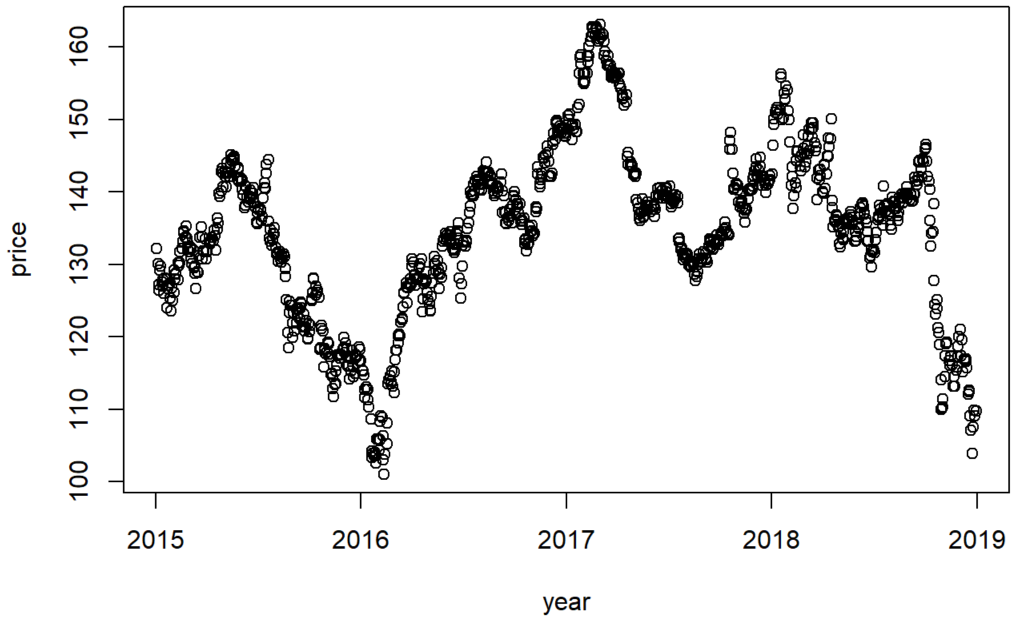

```
pacf(ts_log, lag.max=40, main="PACF_CHANGE")
```




```
plot(data$Date, log(data$Adj.Close), xlab = "year", ylab = "log(price)")
```




```
plot(data$Date, data$Adj.Close, xlab = "year",ylab = "price")
```



From conparision of the 2 graphs. After taking logrithis, the y-axis labels ranges from 4.6 to 5.1 instead of the original 100 to 160. More precisely, taking logarithms is very helpful in stabilizing the size of the oscillations.