Data source:

http://finance.yahoo.com/q/hp?s=GOOG&a=00&b=1&c=2013&d=03&e=1&f=2016&g=m

> project<-read.csv(file.choose())

> data = project[2]

> par(mfrow=c(2,3))

> ts.plot(diff(data), main = "Google stock data after one difference")

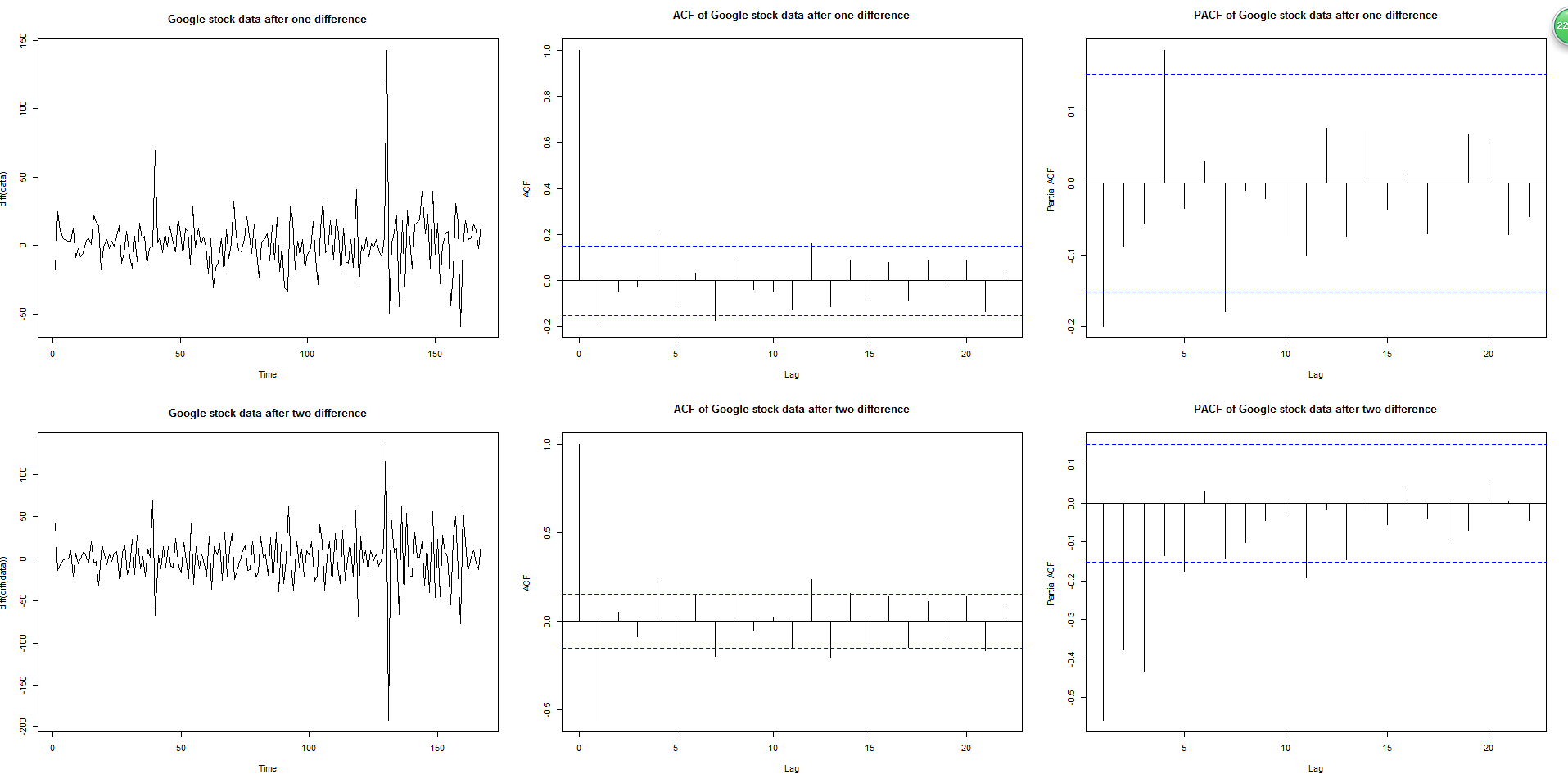
> acf(diff(data), main = "ACF of Google stock data after one difference")

> pacf(diff(data), main = "PACF of Google stock data after one difference")

> ts.plot(diff(diff(data)), main = "Google stock data after two difference")

> acf(diff(diff(data)), main = "ACF of Google stock data after two difference")

> pacf(diff(diff(data)), main = "PACF of Google stock data after two difference")



> train = data[2:160,]

> test = data[161,170,]

> par(mfrow=c(2,3))

> ts.plot(diff(train), main = "First 160 data after one difference")

> par(mfrow=c(2,3))

> ts.plot(diff(train), main = "First 160 data after one difference")

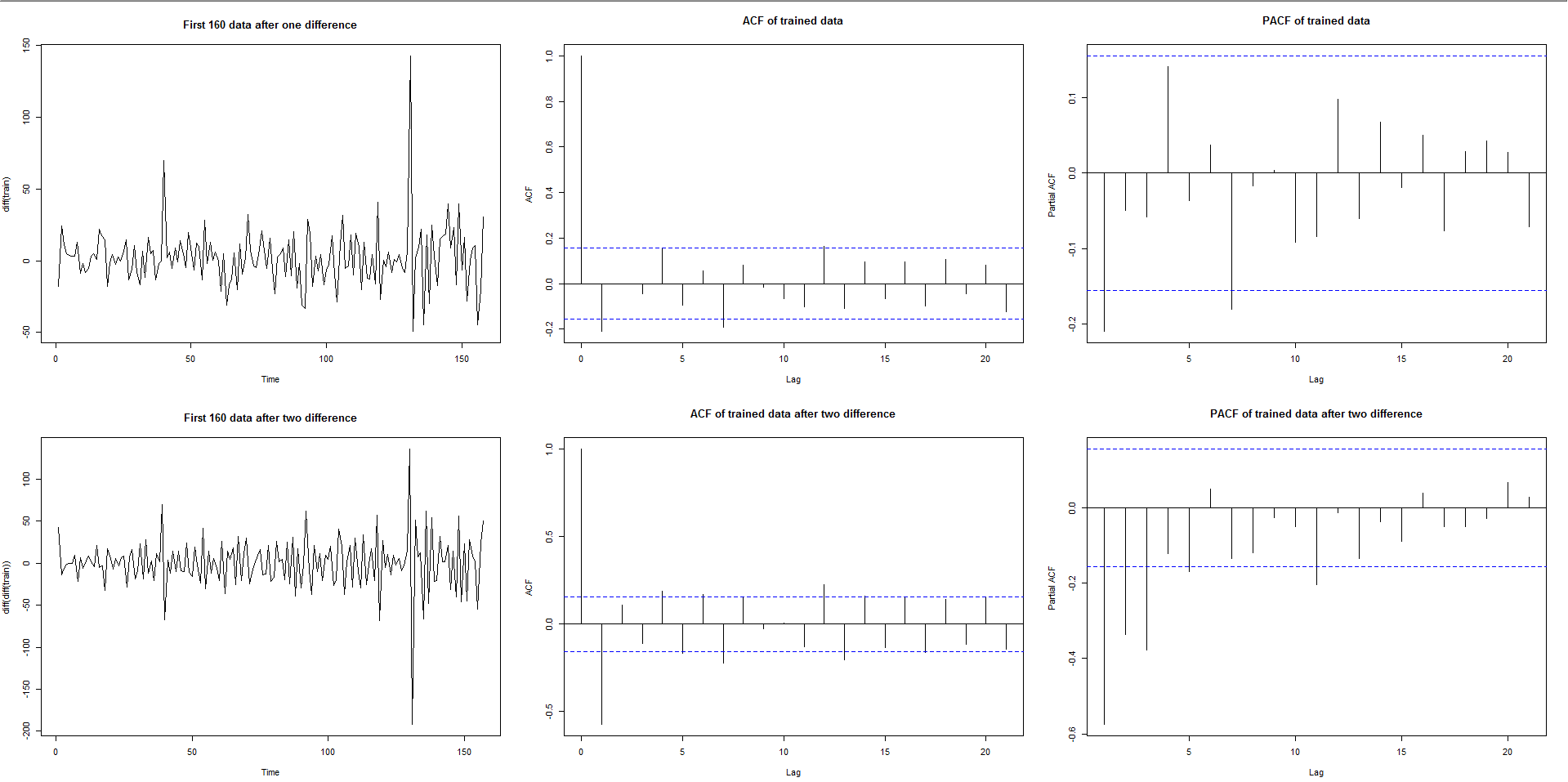
> acf(diff(train), main = "ACF of trained data")

> pacf(diff(train), main = "PACF of trained data")

> ts.plot(diff(diff(train)), main = "First 160 data after two difference")

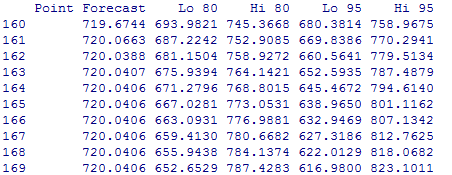
> acf(diff(diff(train)), main = "ACF of trained data after two difference")

> pacf(diff(diff(train)), main = "PACF of trained data after two difference")

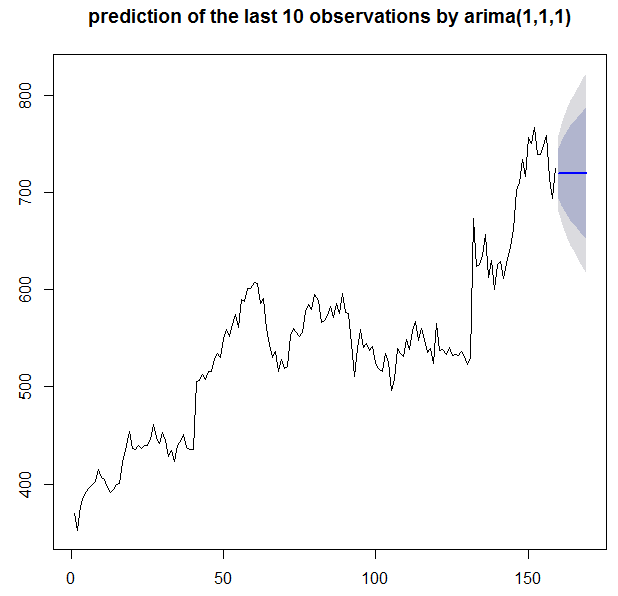


> fit1 = arima(train, order = c(1,1,1))

> forecast(fit1, h = 10)

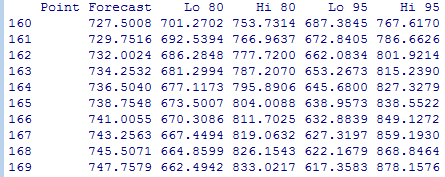


> plot(forecast(fit1),main = "prediction of the last 10 observations by arima(1,1,1)")

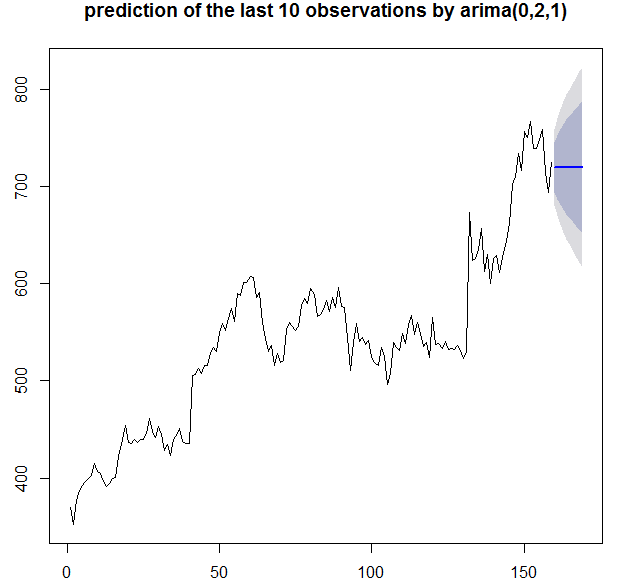


> fit2 = arima(train, order = c(0,2,1))

> forecast(fit2, h = 10)



plot(forecast(fit1),main = "prediction of the last 10 observations by arima(0,2,1)")





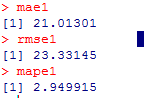
> forecast1 = forecast(fit1, h = 10)

> err1 = test - forecast1$mean

> mae1 = mean(abs(err1))

> rmse1 = sqrt(mean(err1^2))

> mape1 = mean(abs((err1\*100)/test))



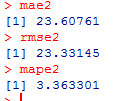
> forecast2 = forecast(fit2, h = 10)

> err2 = test - forecast2$mean

> mae2 = mean(abs(err2))

> rmse2 = sqrt(mean(err1^2))

> mape2 = mean(abs((err2\*100)/test))



> fit3 = HoltWinters(ts(train, frequency = 4))

> forecast3 = forecast(fit3, h = 10)

> forecast3

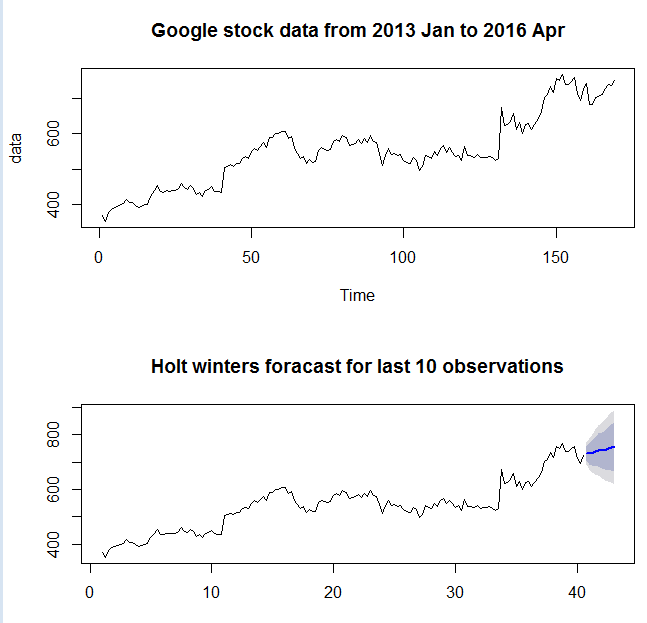
> par(mfrow=c(2,1))

> ts.plot(data)

> par(mfrow=c(2,1))

> ts.plot(data, main = "Google stock data from 2013 Jan to 2016 Apr")

> plot(forecast3, main = "Holt winters foracast for last 10 observations")

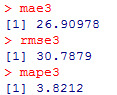


> err3 = test - forecast3$mean

> mae3 = mean(abs(err3))

> mape3 = mean(abs((err3\*100)/test))

> rmse3 = sqrt(mean(err3^2))

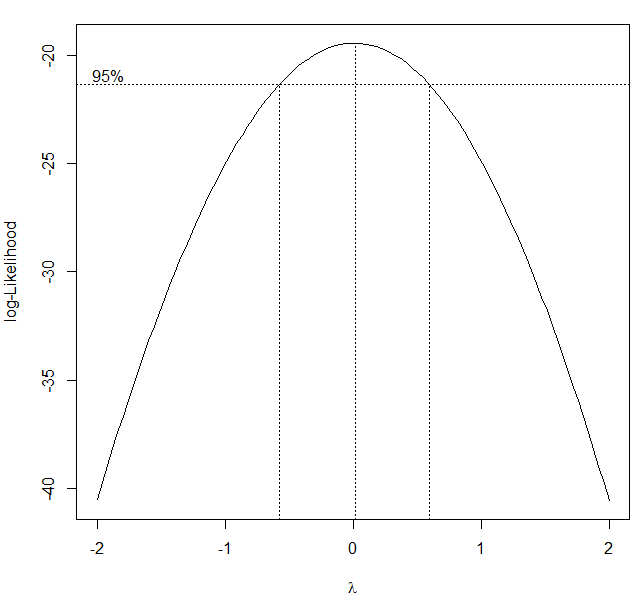


Use classical decomposition approach:

1. Box-cox transformation:

>library(MASS)

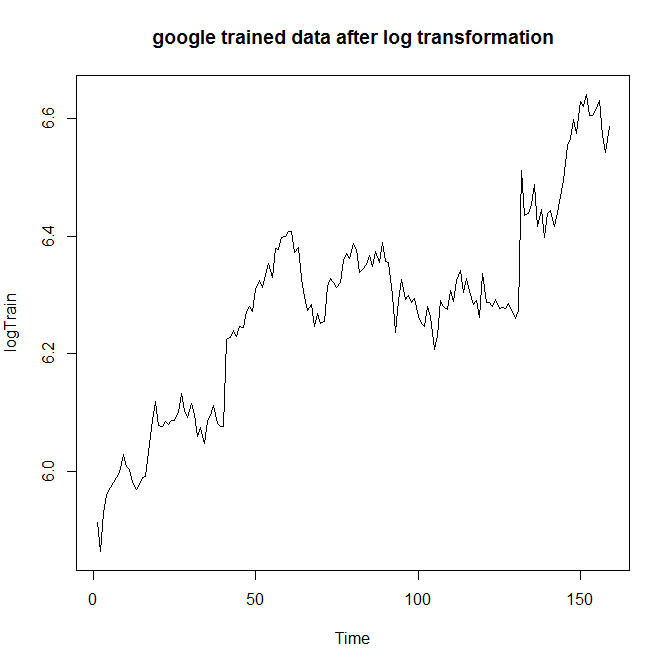
>boxcox(train~c(1:length(train)))



So we can use log transformation to stablize variance.

> logTrain = log(train)

> ts.plot(logTrain, main = "google trained data after log transformation")



> par(mfrow=c(2,2))

> ts.plot(train, main = "google stock trained data")

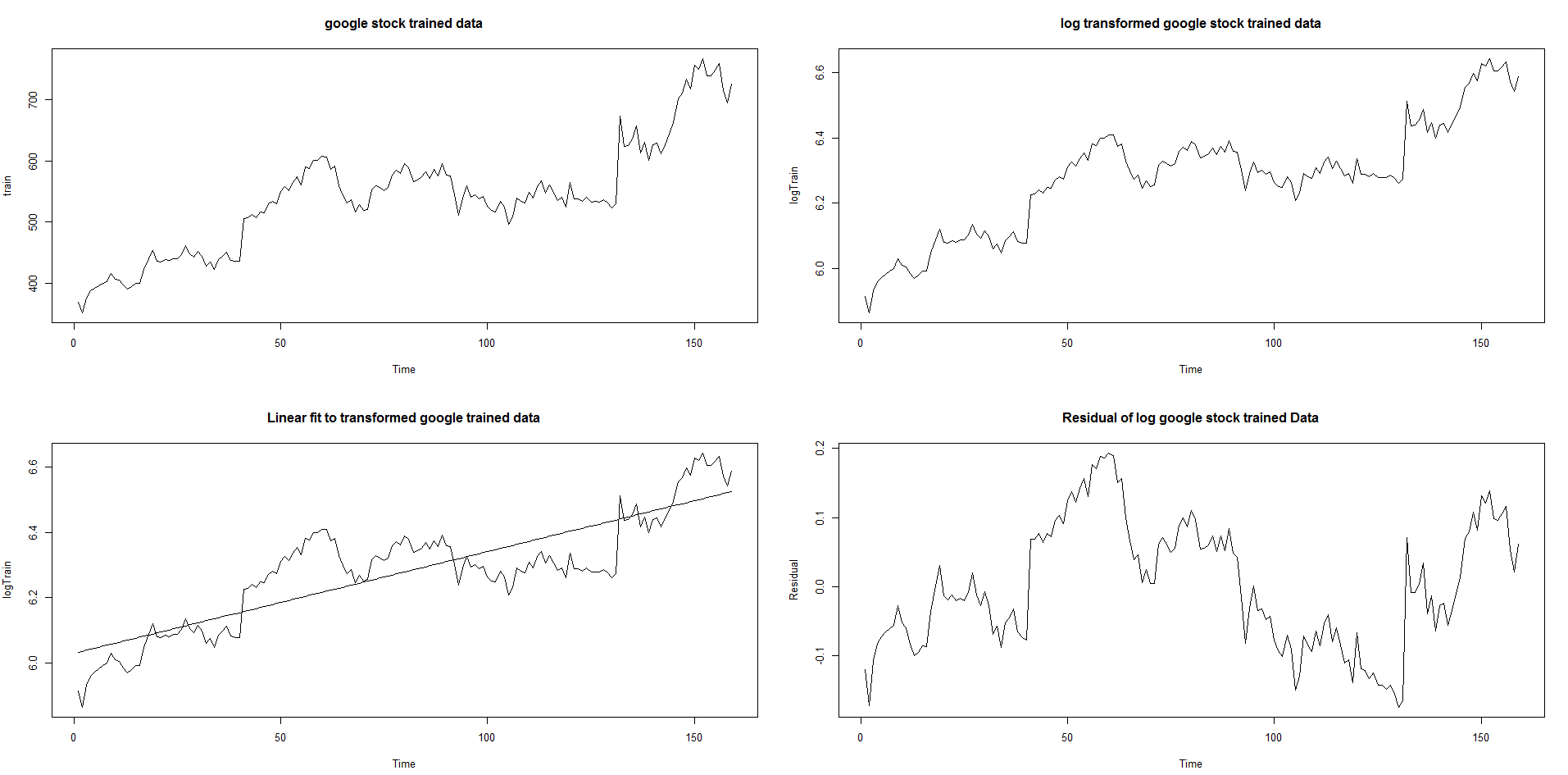
> ts.plot(logTrain, main = "log transformed google stock trained data")

> ts.plot(logTrain, main = "Linear fit to transformed google trained data")

> linearFit = lm(logTrain~c(1:length(logTrain)))

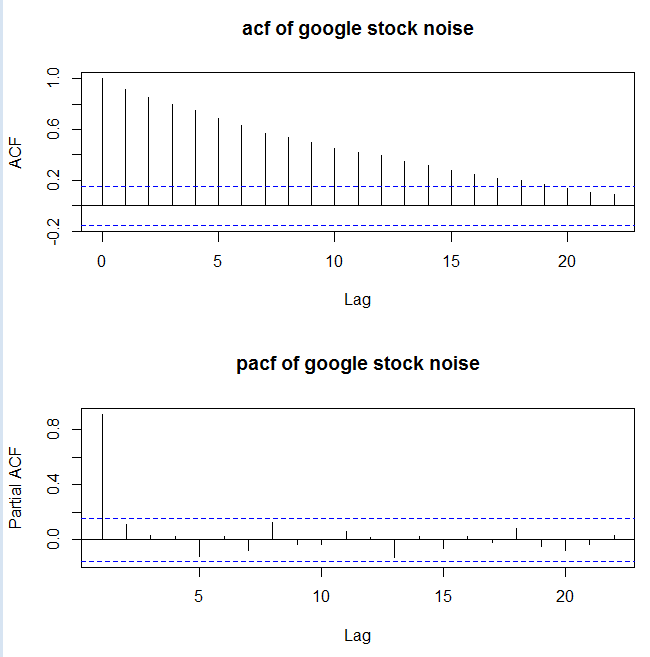
> lines(ts(linearFit$fitted.values))

> plot(ts(linearFit$residuals),ylab="Residual", main = "Residual of log google stock trained Data")



> acf(ts(linearFit$residuals), main = "acf of google stock noise")

> pacf(ts(linearFit$residuals), main = "pacf of google stock noise")



Try high order trend-fit:

> t1 = c(1:length(logTrain))

> t2 = t1^2

> t3 = t1^3

> t4 = t1^4

> t5 = t1^5

> Fit2 = lm(logTrain~t1 + t2)

> Fit3 = lm(logTrain~t1 + t2 + t3)

> Fit4 = lm(logTrain~t1 + t2 + t3 + t4)

> Fit5 = lm(logTrain~t1 + t2 + t3 + t4 + t5)

> par(mfrow=c(2,2))

> ts.plot(logTrain, main = "square fit to transformed google trained data")

> lines(ts(Fit2$fitted.values))

> ts.plot(logTrain, main = "cube fit to transformed google trained data")

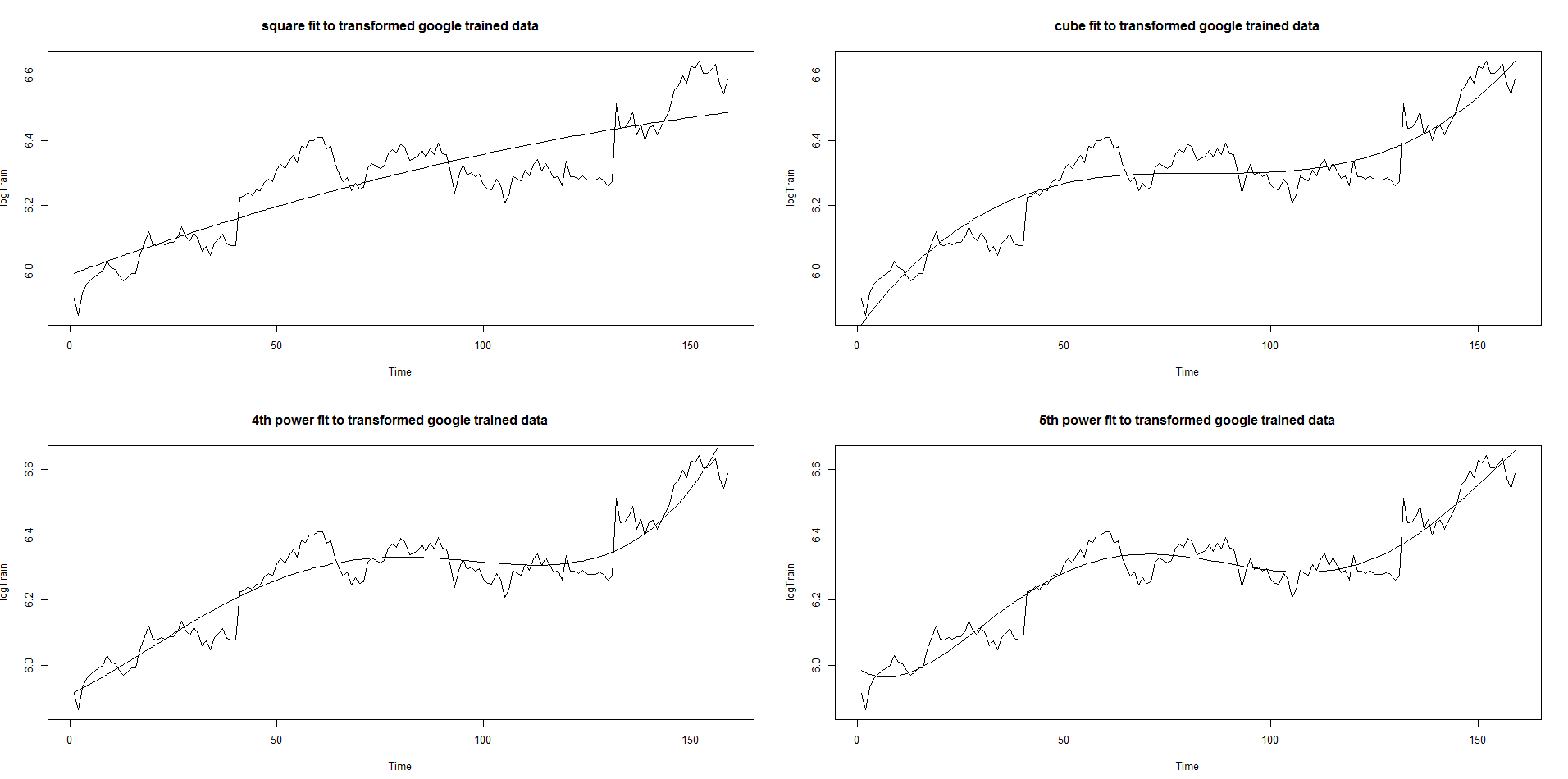
> lines(ts(Fit3$fitted.values))

> ts.plot(logTrain, main = "4th power fit to transformed google trained data")

> lines(ts(Fit4$fitted.values))

> ts.plot(logTrain, main = "5th power fit to transformed google trained data")

> lines(ts(Fit5$fitted.values))



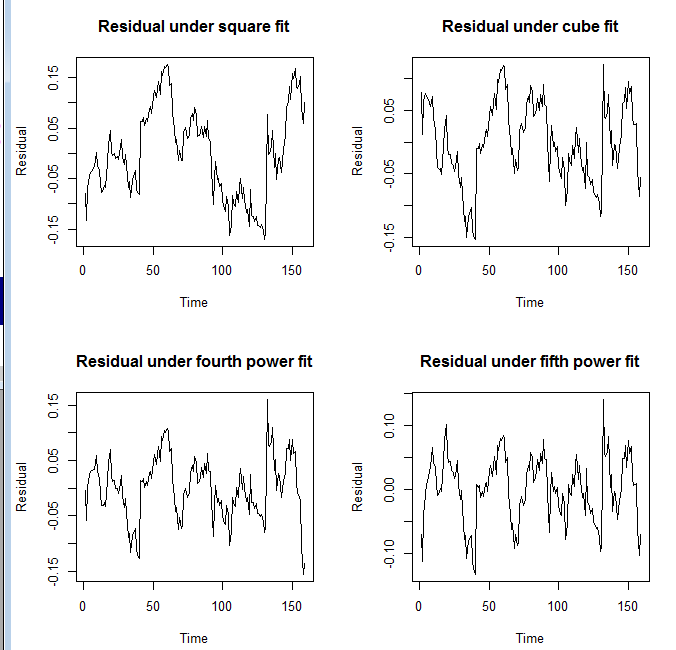
> par(mfrow=c(2,2))

> plot(ts(Fit2$residuals),ylab="Residual", main = "Residual under square fit")

> plot(ts(Fit3$residuals),ylab="Residual", main = "Residual under cube fit")

> plot(ts(Fit4$residuals),ylab="Residual", main = "Residual under fourth power fit")

> plot(ts(Fit5$residuals),ylab="Residual", main = "Residual under fifth power fit")



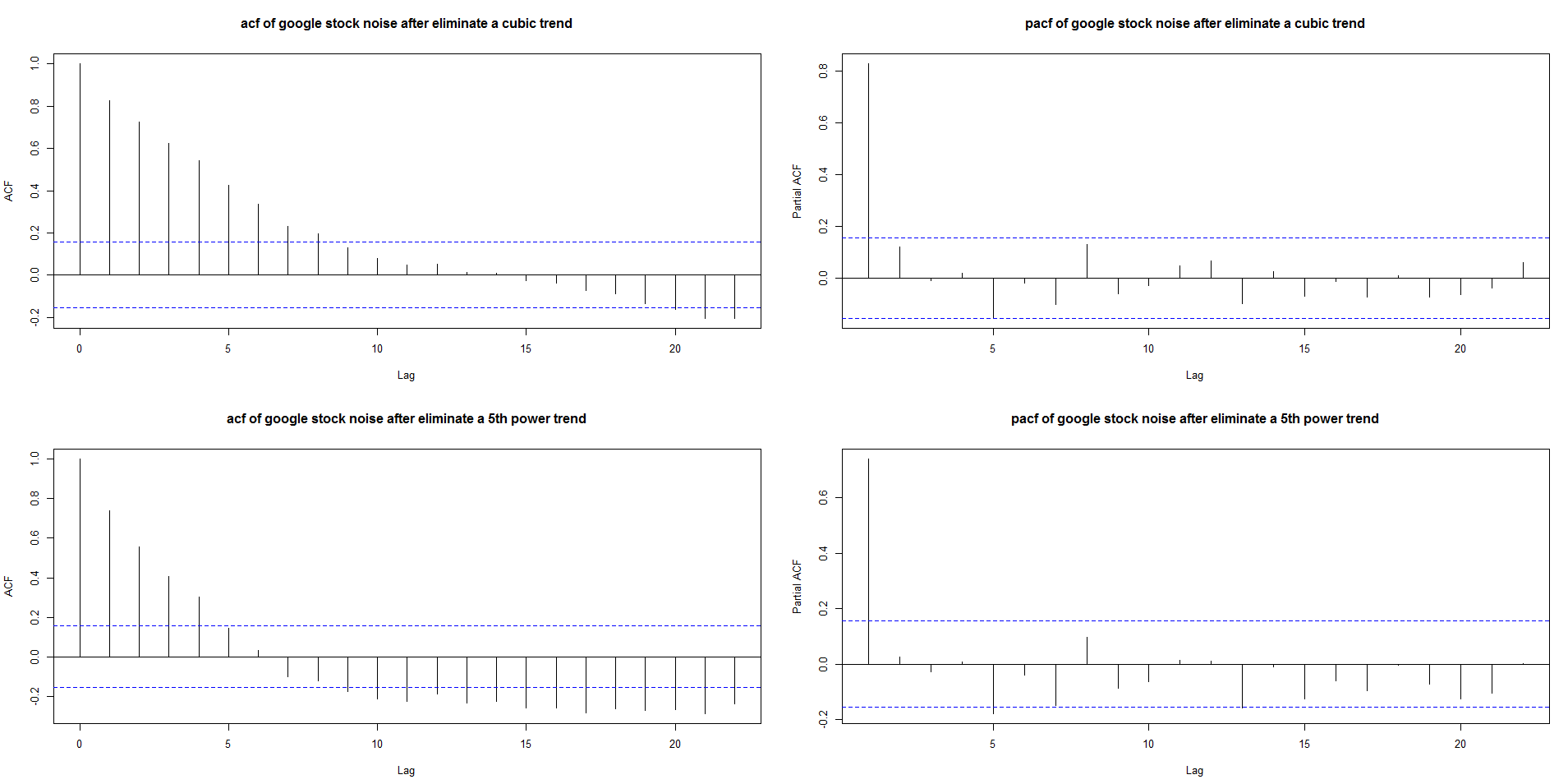
> par(mfrow=c(2,2))

> acf(ts(Fit3$residuals), main = "acf of google stock noise after eliminate a cubic trend")

> pacf(ts(Fit3$residuals), main = "pacf of google stock noise after eliminate a cubic trend")

> acf(ts(Fit5$residuals), main = "acf of google stock noise after eliminate a 5th power trend")

> pacf(ts(Fit5$residuals), main = "pacf of google stock noise after eliminate a 5th power trend")



**This is interesting: as the order of polunomial goes up, the acf of residuals decay faster!**