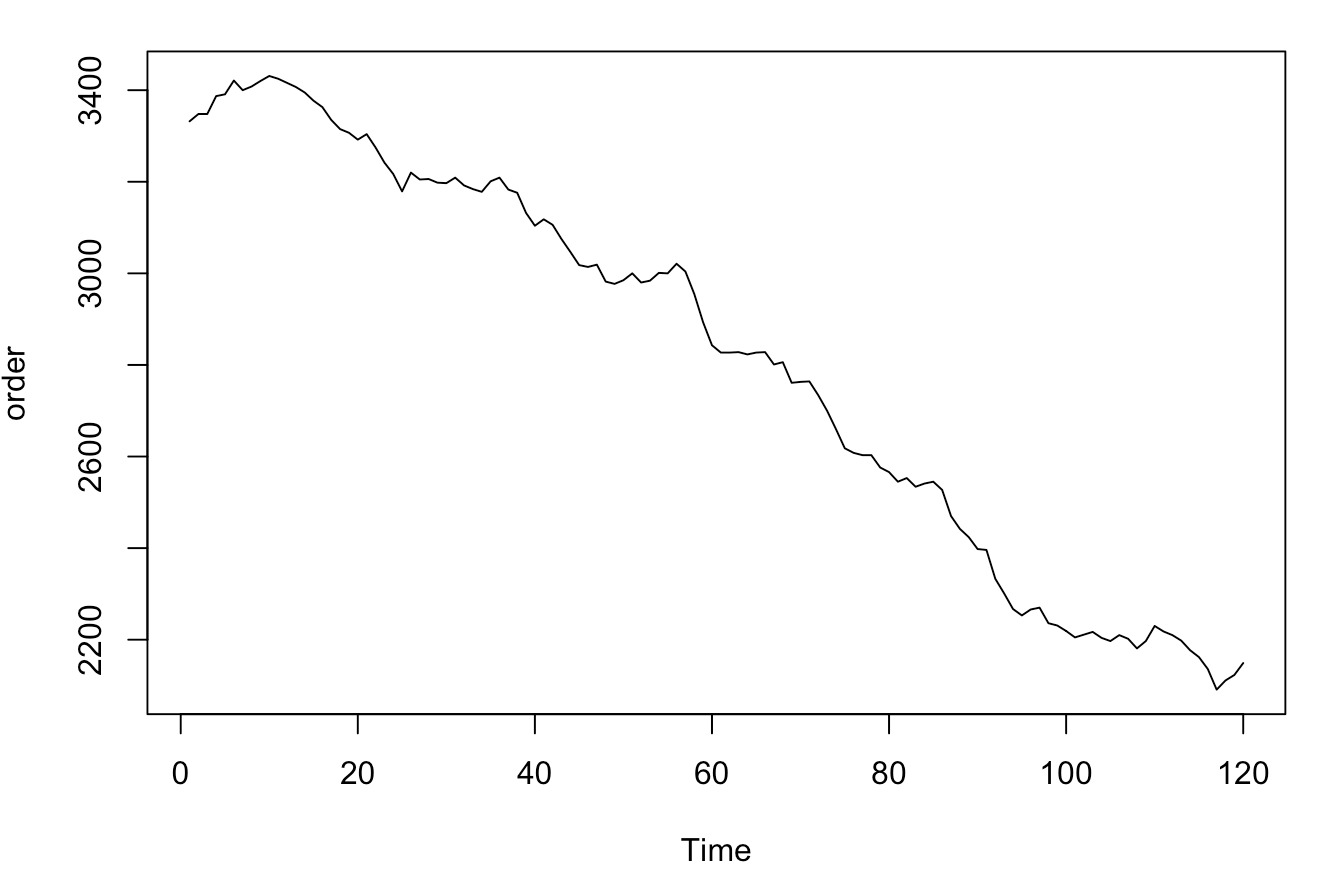
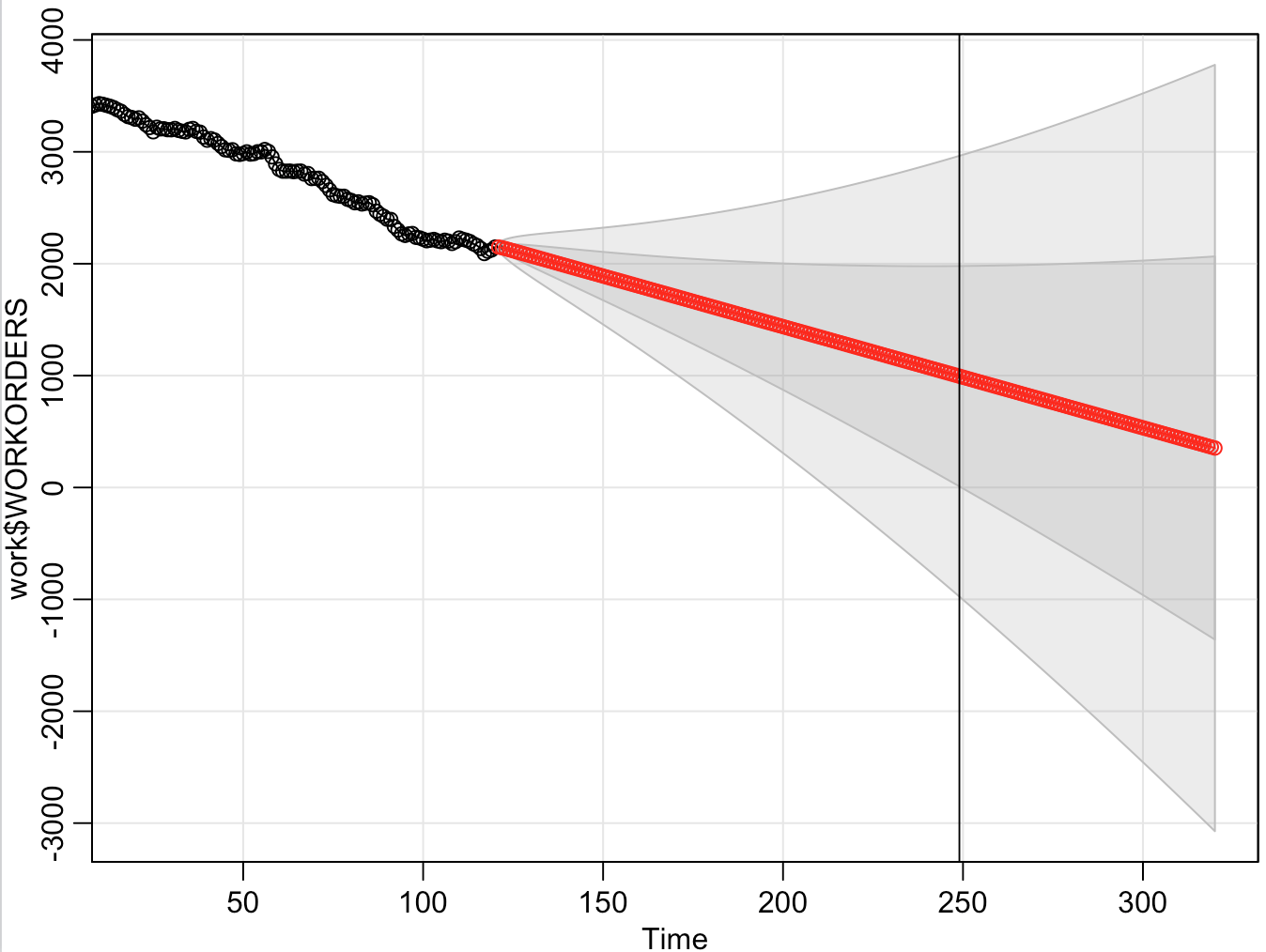
Report

1. Executive Summary

* Objective—This report presents an analysis on the number of corrective work orders during the construction phase of a nuclear plant. And it provides an estimate for how many days it will take to reach the operational level of 1000 work orders.
* Methods and Data—Here we mainly use the technology of time series, fitting an ARIMA model. The data include the work orders in the first 120 days of the construction phase of a nuclear plant.



* Conclusion—We fit an ARIMA(1,2,1) to the 120 days data, it gradually declines finally reach 1000 work orders after 129 days(exclude the first 120 days).



1. Summary

* The provided dataset includes 120 days data, and the order begins at 3332, ends at 2149. The max data is 3431, the min data is 2091.
* We separate the data into two parts, 90% of the data used for training, and the rest data are used for testing.
* The first 120 data declines rather stably, so at first we fit some linear regression models (Linear Trend Model, Quadratic Trend Model, Cubic Trend Model). Although the model’s R square is really high, the residuals act badly which means it would not do well in future forecasting.
* Then, we fit an ARIMA model, after checking the ACF&PACF graph and the residuals plot, we finally decide to choose ARIMA(1,2,1) and do the forecasting.

1. Other

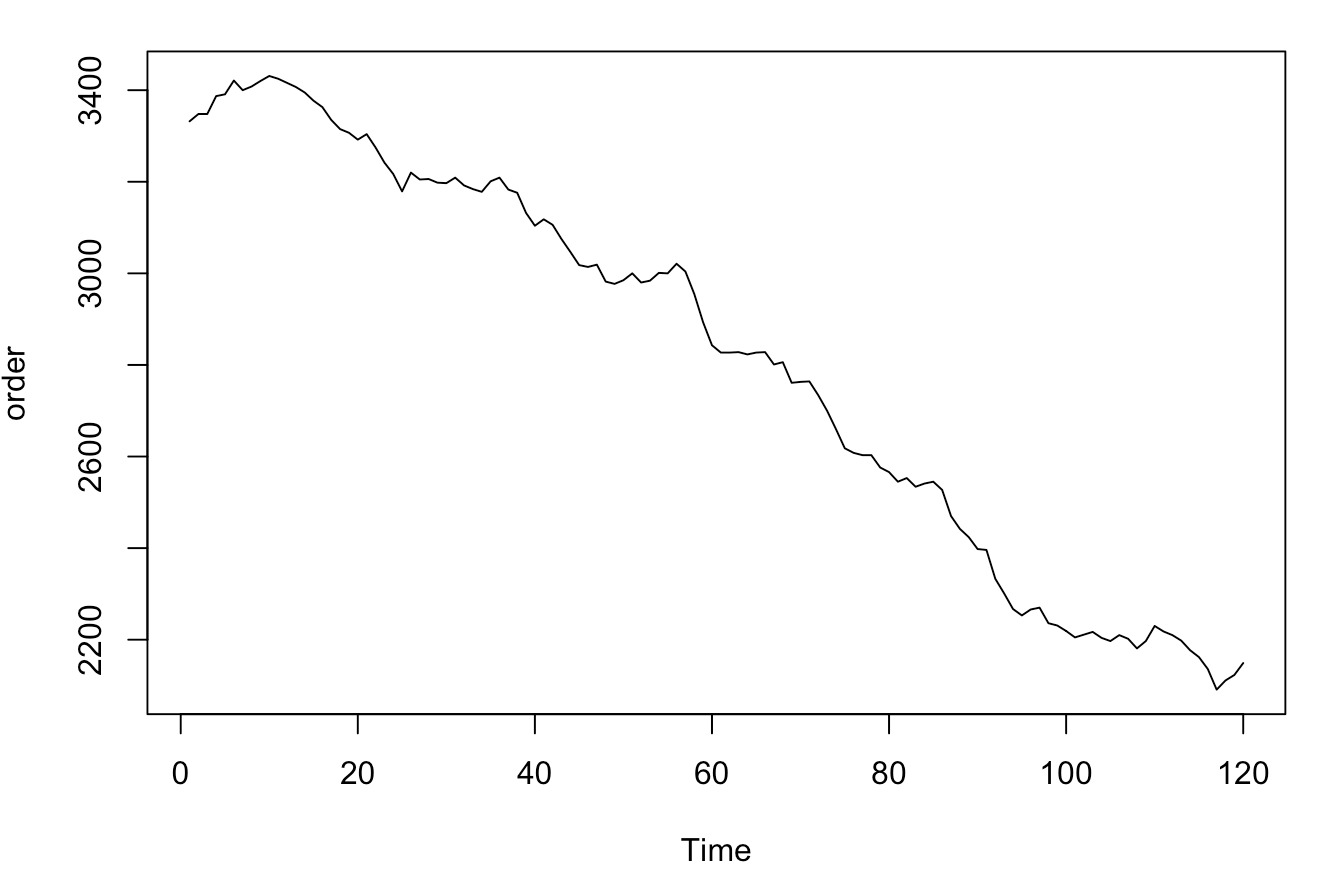
All the update and document are available at <https://github.com/yysyzz/hw5>

Total time is 06 h 26 min, rate is 50$/h and total fee is 346.13$.

Details:

Step 1:

At first, we plot the whole dataset and have a look.



Step 2:

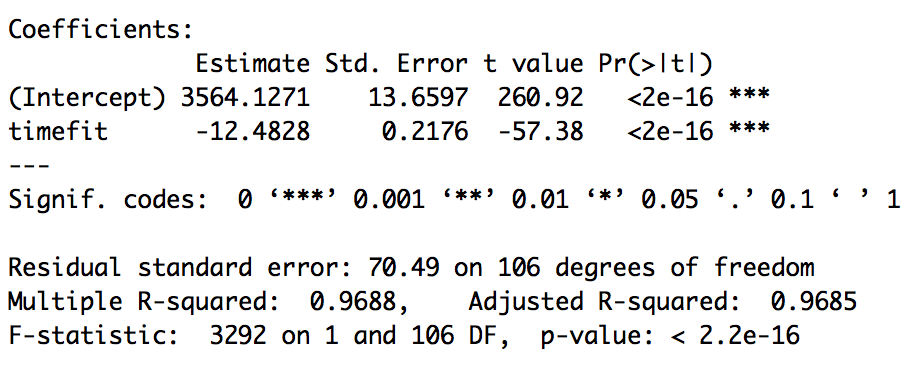
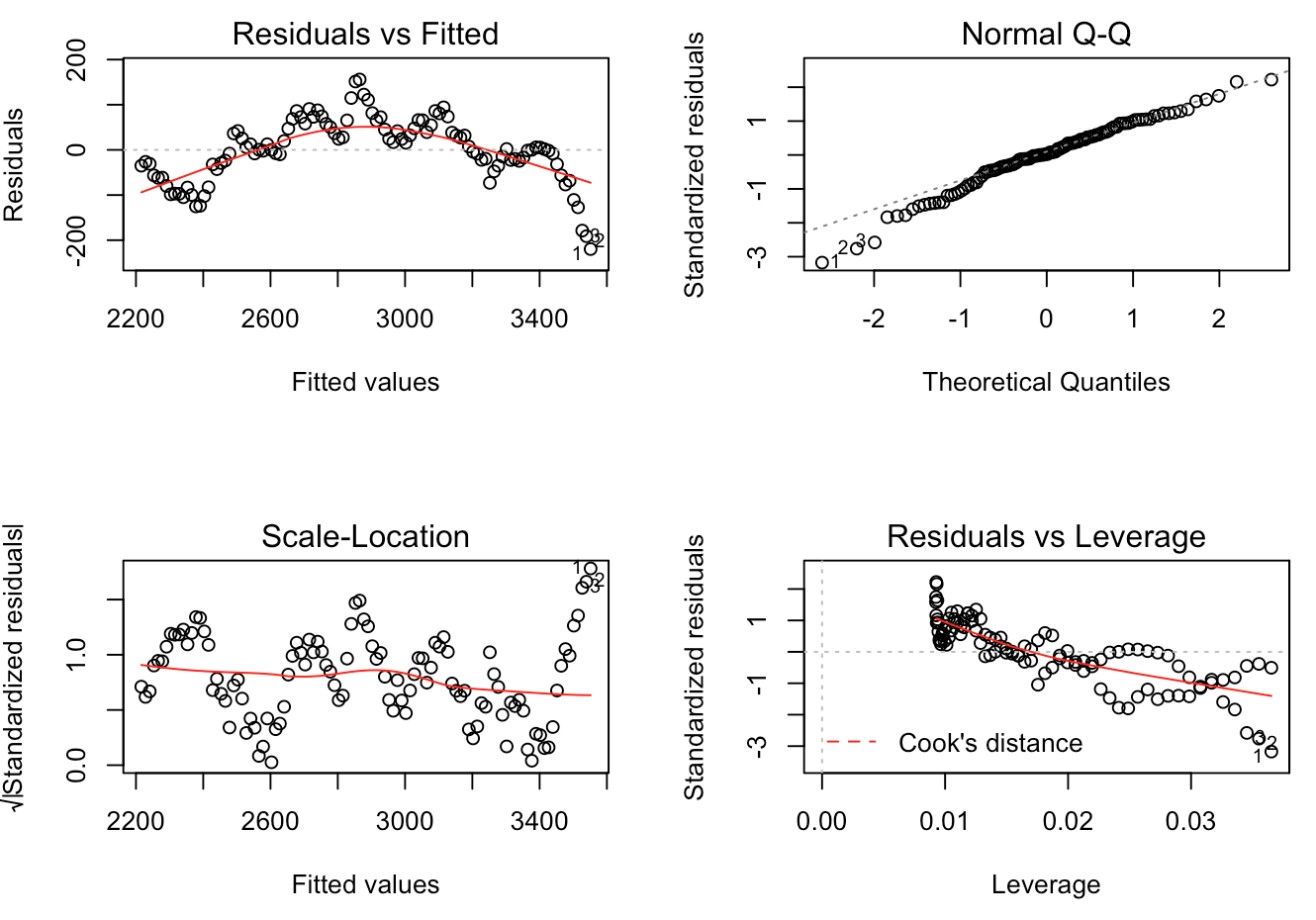
Split the dataset into two parts, the train and the test. We use 90%of the whole dataset to build the train dataset, which contains 108 data. The remain data belong to the test dataset.

Step 3:

We firstly fit Linear Trend Regression model, .

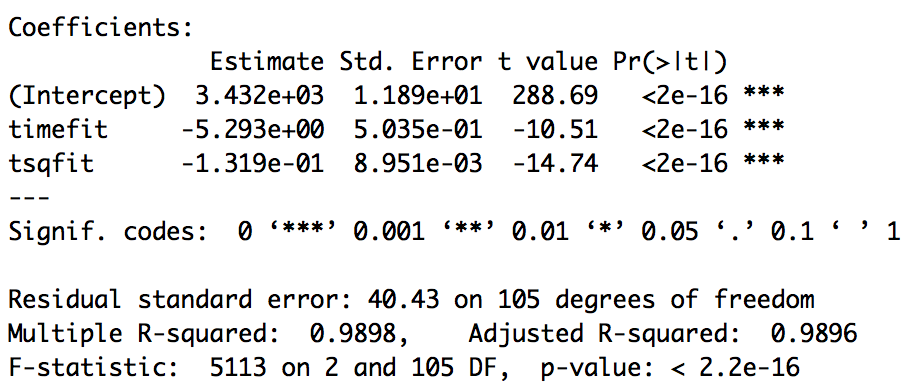
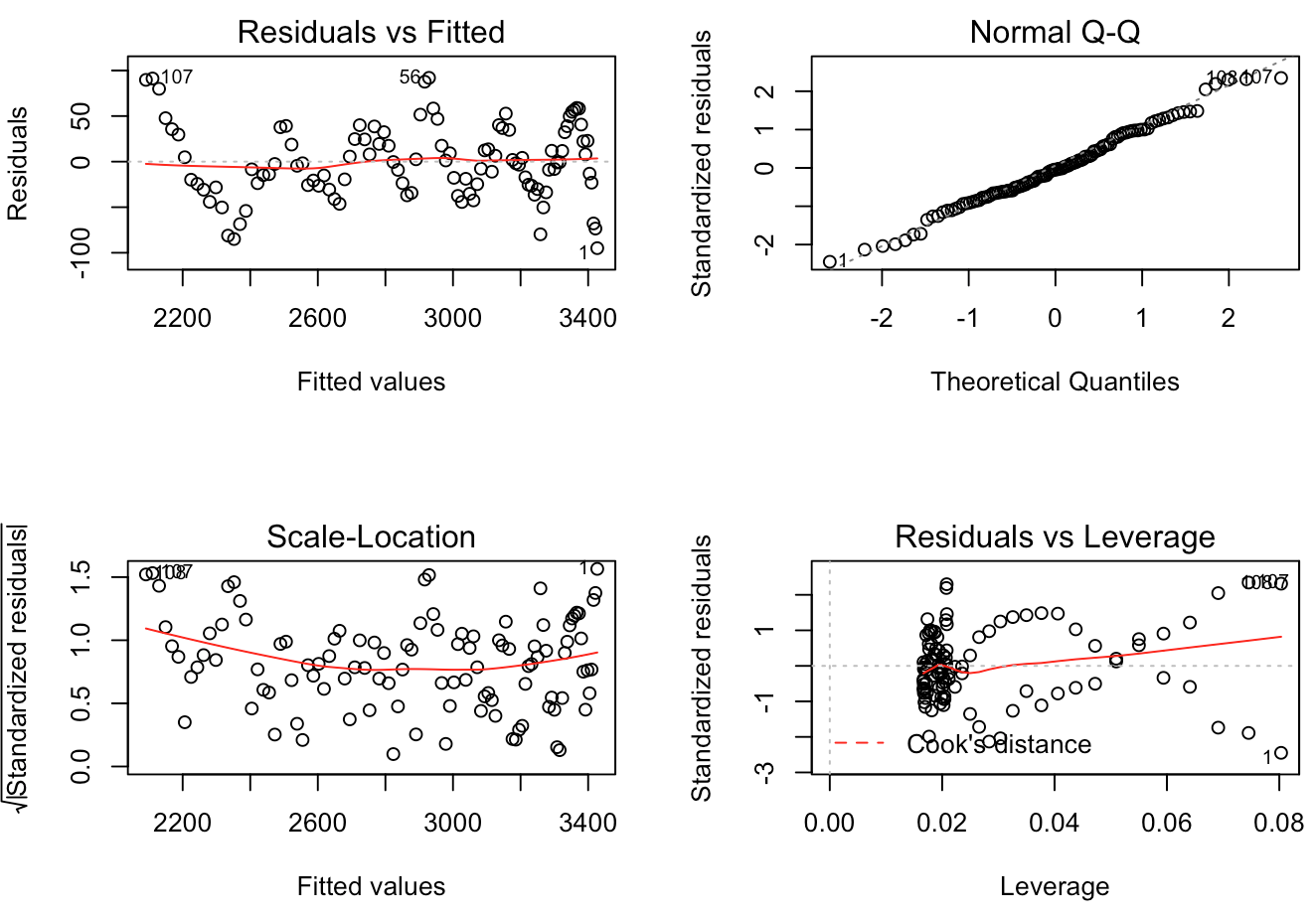
K = 1,

The fitted model is:

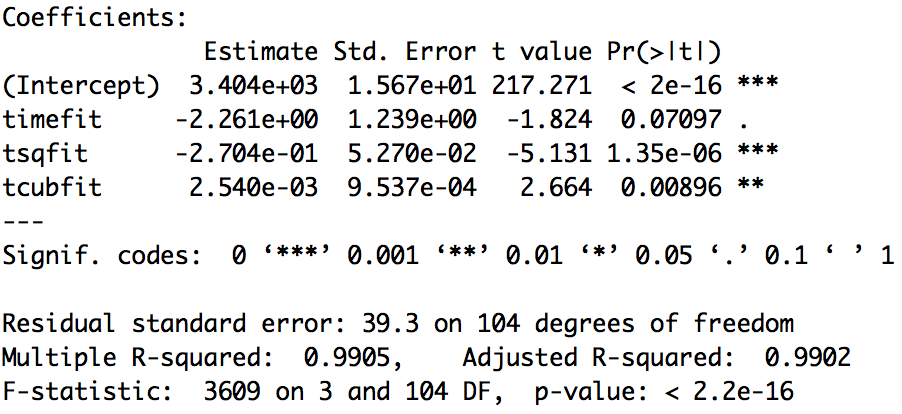
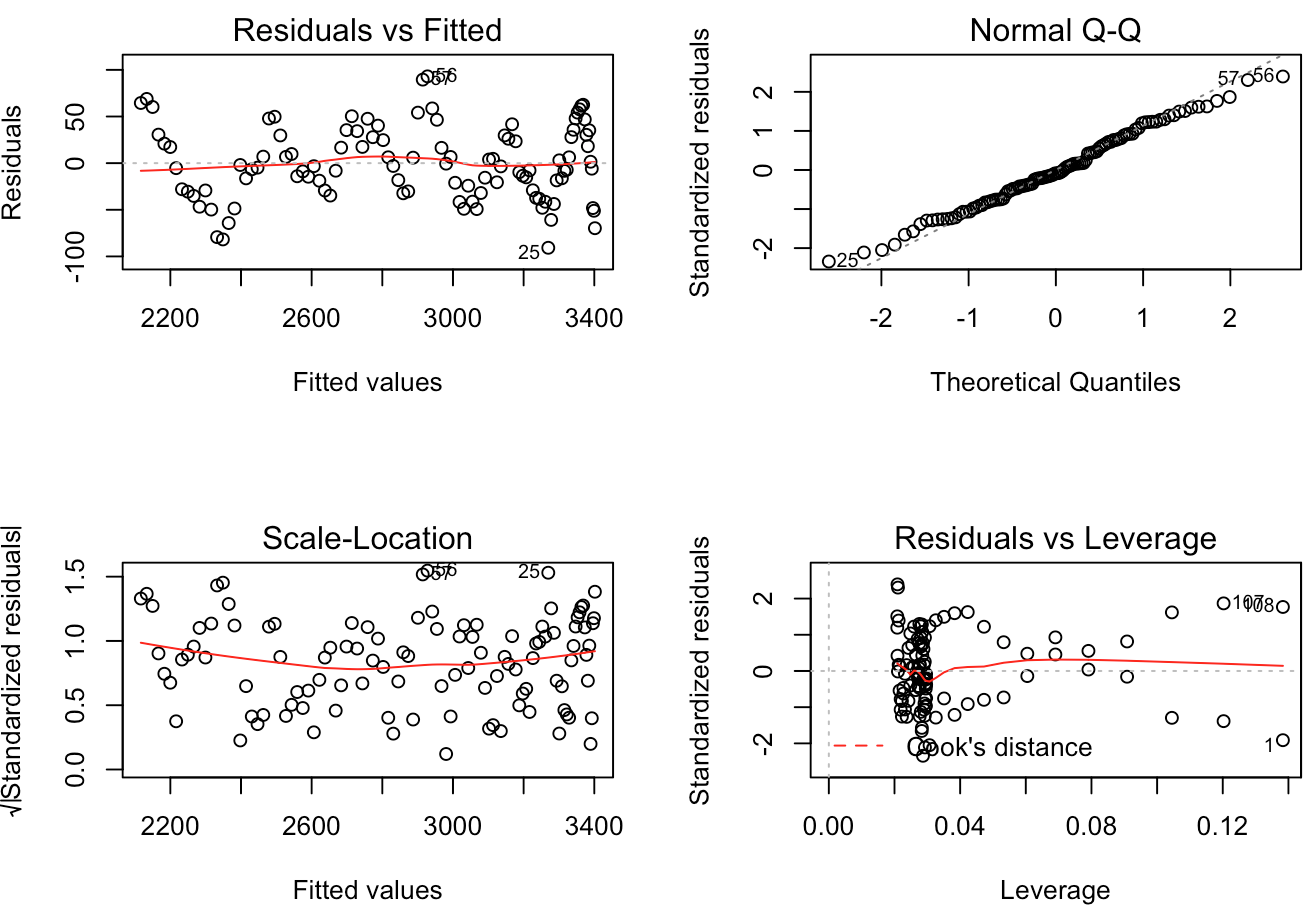
K = 2,

The fitted model is:

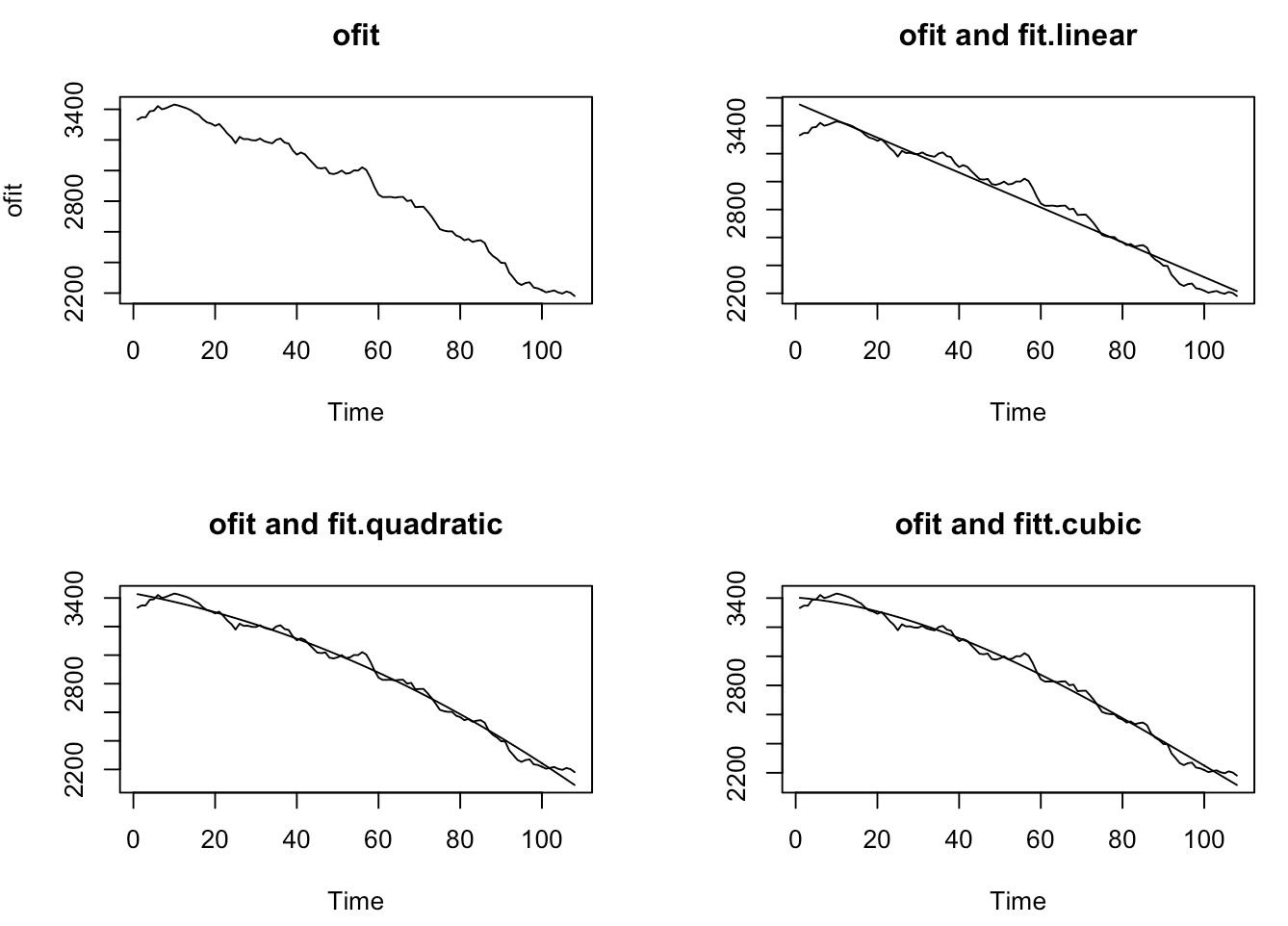
 

K = 3,

The fitted model is:

Plot time series with Fits from k=1,2,3 Models



Compare these three models(Linear Trend Model, Quadratic Trend Model, Cubic Trend Model, which one is preferred)

(1) Compare models using the R2 criterion

0.9688, 0.9898,0.9905,Cubic Trend model.

(2) Compare models using Significance Tests

we use the Extra Sum of Squares F-test.

Cubic Trend Model or Linear Trend Model ?

Extra SS F-test= 118.4447, qf(0.95,2,104)= 3.083706, so reject H0.

Compare the Quadratic Trend Model to the Cubic Trend Model, ,

Extra SS F-test= 7.0938, qf(0.95,1,104)=3.932438, so reject H0.

The significance tests prefer the Cubic Trend model.

(3) AIC

11.38555, 10.2827, 10.23522. The AIC prefer the Cubic Trend model.

(4) BIC

11.46006, 10.38204, 10.35939. The BIC prefer the Cubic Trend model.

(5) AICc

9.548908, 8.456612, 8.420159. The AICc prefer the Cubic Trend model.

(6)ME

31.98648, 206.1312, 158.2064. Linear Trend Model

(7)MPE

1.470221, 9.531554, 7.308646. Linear Trend Model

(8)MSE

1599.984, 44171.91, 25983.9. Linear Trend Model

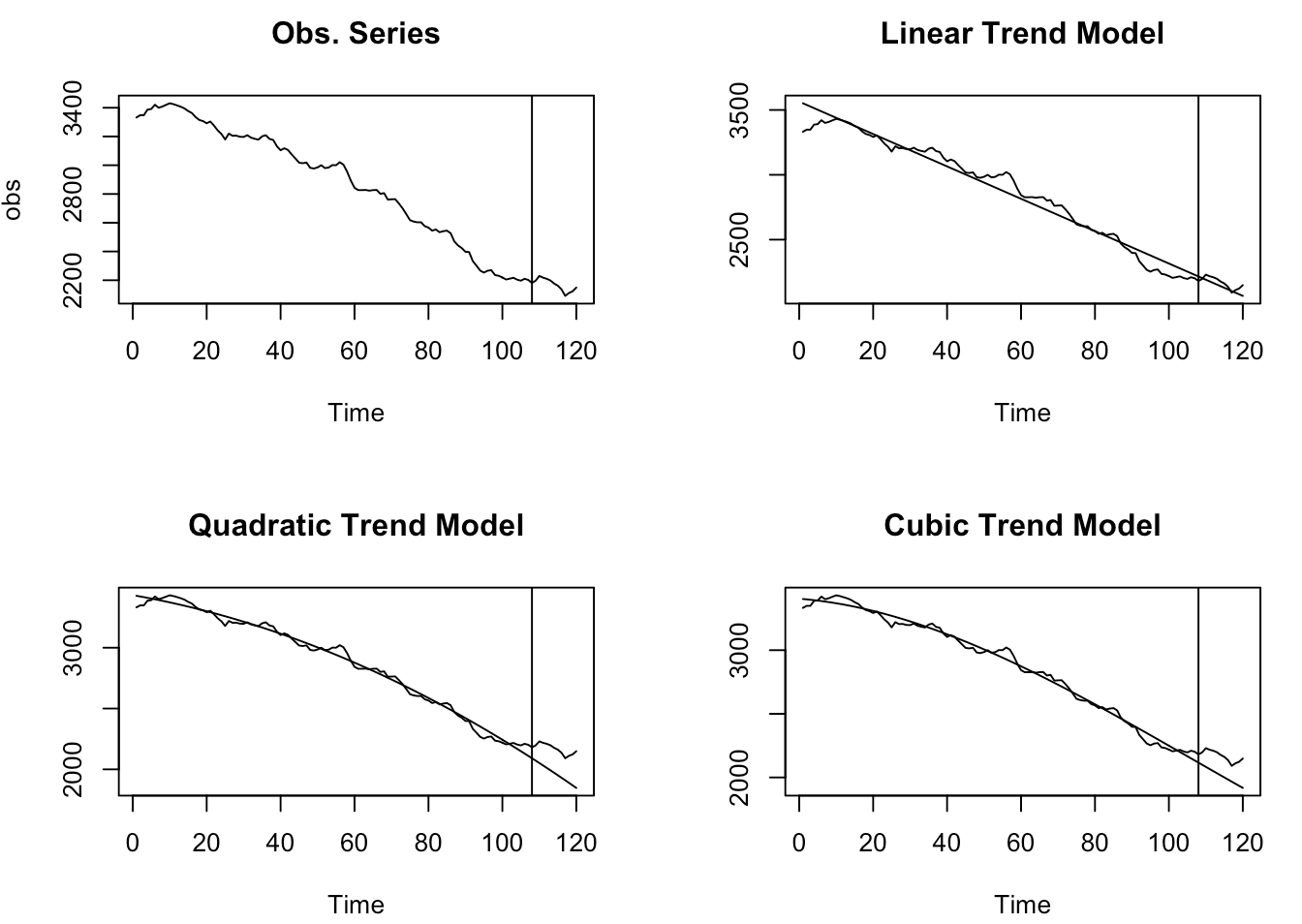
(9)MAE

35.17683, 206.1312, 158.2064. Linear Trend Model

(10)MAPE

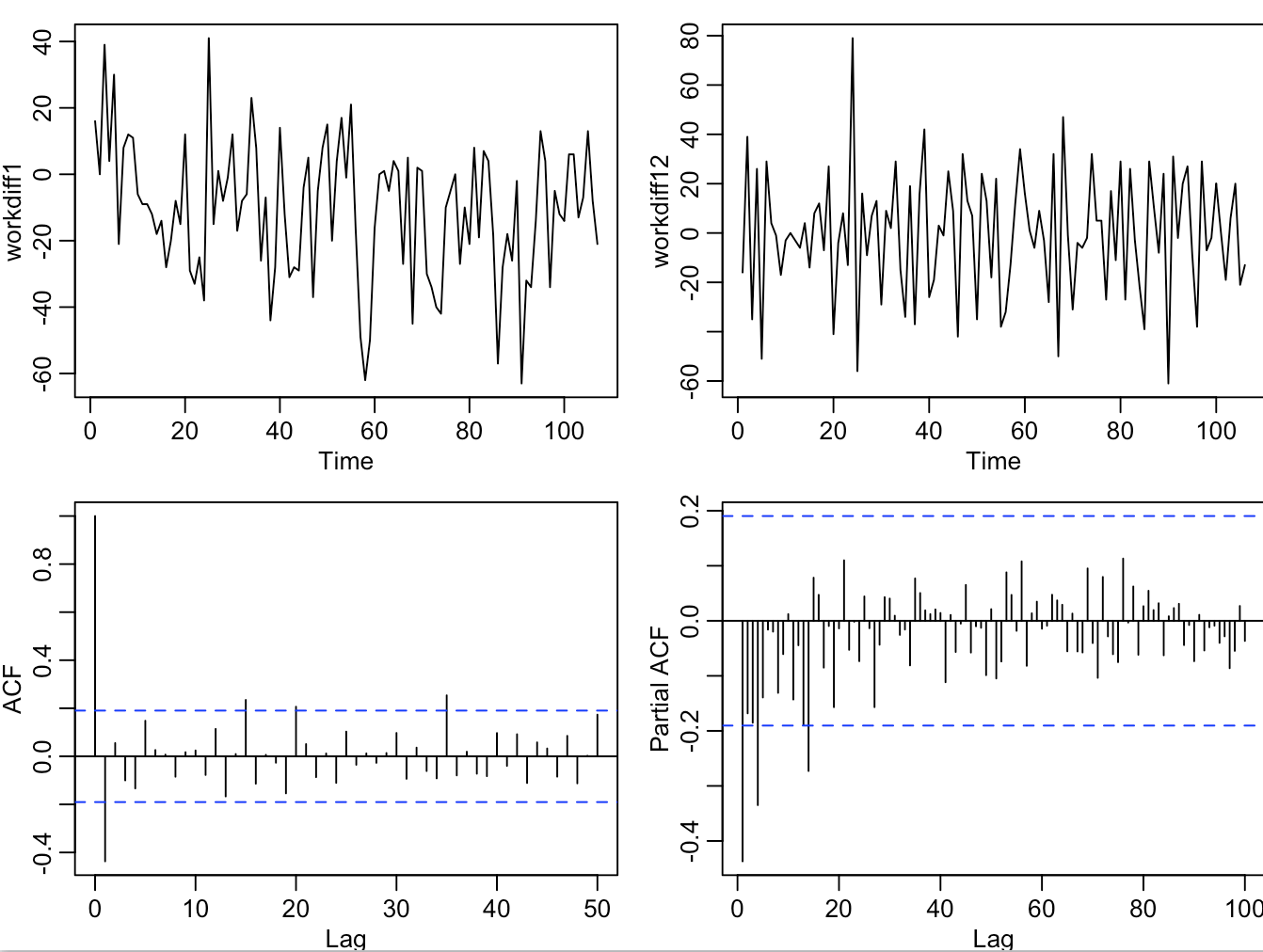
1.620296, 9.531554, 7.308646. Linear Trend Model

Finally, we choose the Linear Trend Model. Predict day is 86. (85 day: 1005.153268, 86 day: 992.670468)

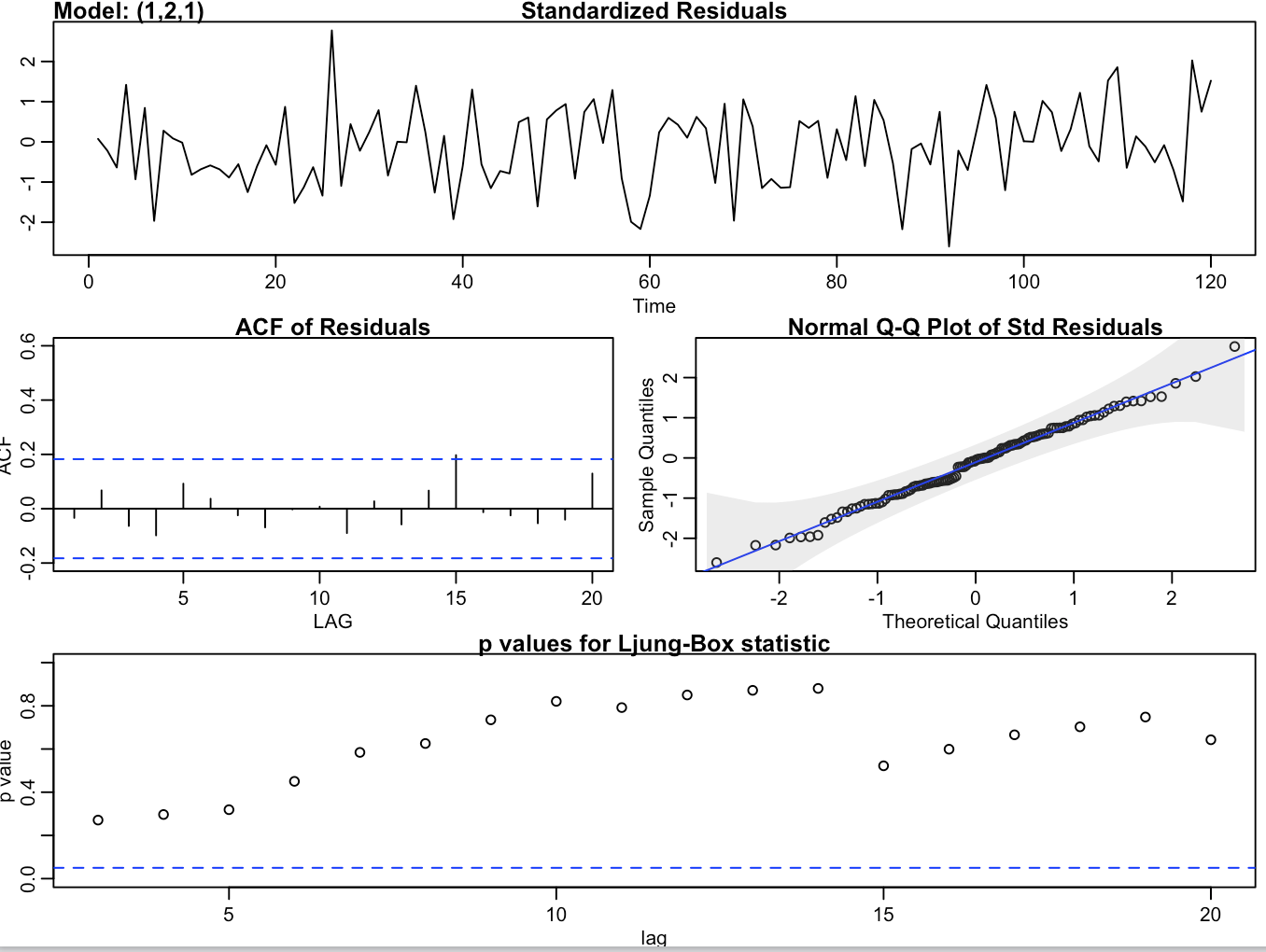
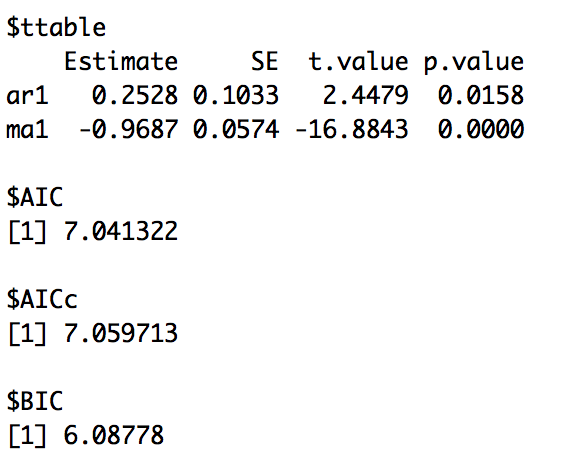


Step 4: Fit a time series model.

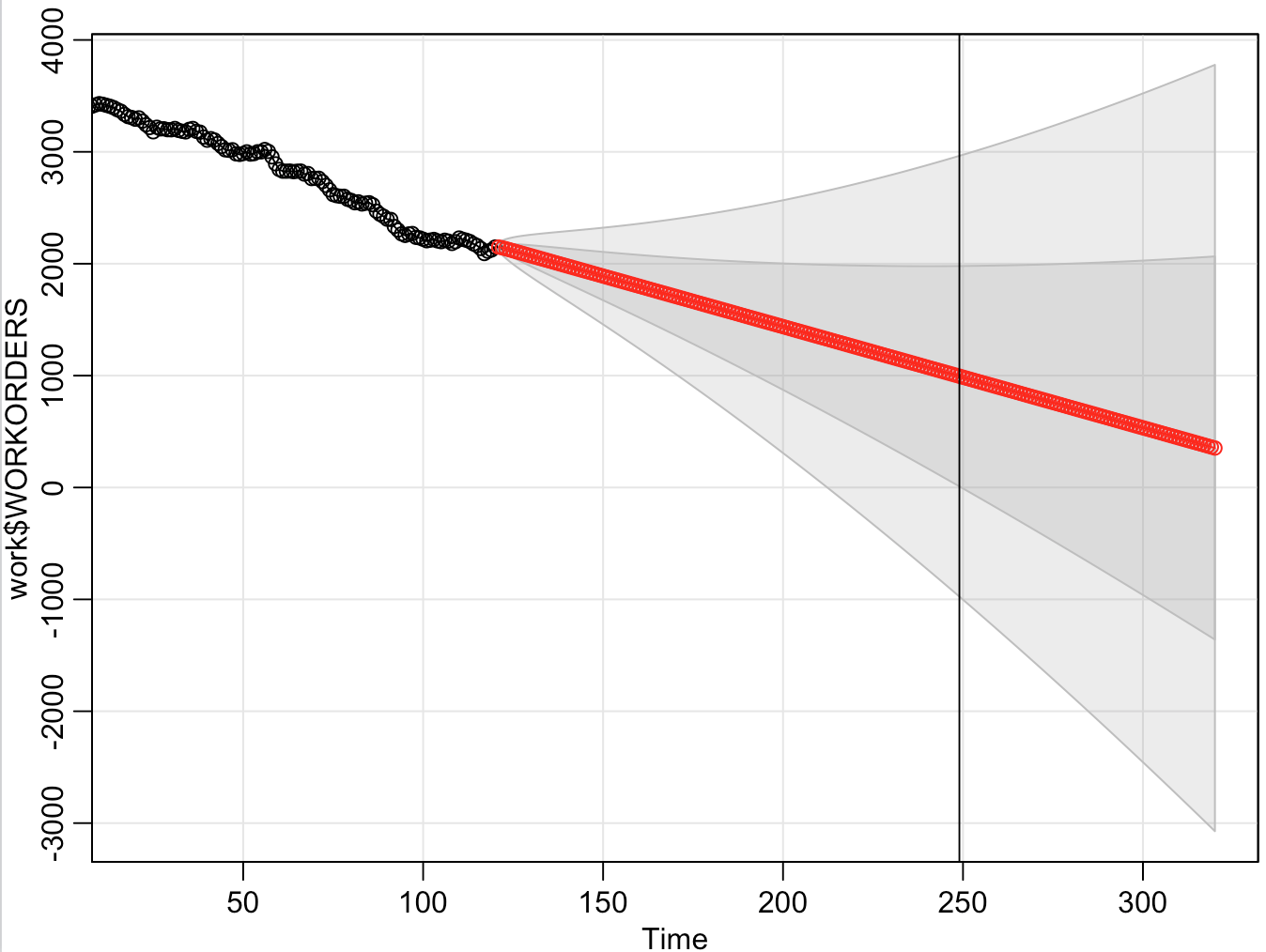
We firstly plot the data, since the variance doesn’t change much, we don’t need to do log to the data. Obviously, there exists a trend, so we do two times differencing (df2) and the trend become stationary. Then we draw the ACF and PACF plot. The ACF cuts off at lag 1, and the PACF tails off, so we try to fit ARIMA(0,2,1) and similar ARIMA(1,2,1),ARIMA(2,2,1), ARIMA(3,2,1), ARIMA(4,2,1). Consider theAIC,AICc,BIC, we choose ARIMA(1,2,1). Compare with the linear model, the arima model is better.



Consider that the last 12 day data might have a large influence to the model, we decide to fit the ARIMA model with the whole dataset. And the final model is ARIMA(1,2,1)



Use the final model to estimate the day reaching 1000 order, the result is 129.(128 day:1003.6528 129 day: 994.6121)



Limitation:

Since se of the predicted value is large, so the predict interval actually is really large. There is a big possibility that it would reach 1000 work orders much earlier or later. Also from previous experience, the series should reach a steady state which can’t not be shown with our ARIMA model.

Appendix:

1. code

#read the data in cvs

work <- read.csv("workorders.csv",header = T)

#print the whole dataset

day <- work$DAY

order <- work$WORKORDERS

order <- ts(order)

ts.plot(order)

order

#split the data into train and test part.

ofit <- order[1:108]

nfit <- length(ofit)

timefit <- time(ofit)

otest <- order[109:120]

ntest <- length(otest)

timetest <- time(otest)

#fit linear regression model

#k = 1

mlr.lin <- lm(ofit~timefit)

summary(mlr.lin)

plot(mlr.lin)

anova(mlr.lin)

#k = 2

tsqfit <- timefit^2/factorial(2)

mlr.quad <- lm(ofit~timefit+tsqfit)

summary(mlr.quad)

plot(mlr.quad)

anova(mlr.quad)

#k = 3

tcubfit <- timefit^3/factorial(3)

mlr.cub <- lm(ofit~timefit+tsqfit+tcubfit)

summary(mlr.cub)

plot(mlr.cub)

anova(mlr.cub)

#plot fitted model

par(mfrow=c(2,2))

ts.plot(ofit,main="ofit") # Time Series Plot

# Plot of xfit vs mlr.lin$fitted

plin=cbind(ofit,mlr.lin$fitted)

ts.plot(plin,main="ofit and fit.linear")

pquad=cbind(ofit,mlr.quad$fitted)

ts.plot(pquad,main="ofit and fit.quadratic")

pcub=cbind(ofit,mlr.cub$fitted)

ts.plot(pcub,main="ofit and fitt.cubic")

#compare

sigsq.lin=anova(mlr.lin)[["Mean Sq"]][2]

sigsq.quad=anova(mlr.quad)[["Mean Sq"]][3]

sigsq.cub=anova(mlr.cub)[["Mean Sq"]][4]

# Akaike Information Criterion,AIC

(AIC.lin = AIC(mlr.lin)/nfit )

(AIC.quad = AIC(mlr.quad)/nfit)

(AIC.cub = AIC(mlr.cub)/nfit)

# Bayesian Information Criterion,BIC

(BIC.lin = BIC(mlr.lin)/nfit )

(BIC.quad = BIC(mlr.quad)/nfit)

(BIC.cub = BIC(mlr.cub)/nfit)

# Corrected AIC, i.e., AICc using formula

k = 1

(AICc.lin=log(sigsq.lin)+(nfit+k)/(nfit-k-2) )

k = 2

(AICc.quad=log(sigsq.quad)+(nfit+k)/(nfit-k-2))

k = 3

(AICc.cub=log(sigsq.cub)+(nfit+k)/(nfit-k-2))

#test

new <- data.frame(timefit=c(109:120))

pfore.lin <- predict(mlr.lin,new,se.fit = TRUE)

pfore.lin$fit

efore.lin=otest-pfore.lin$fit

(me.lin=mean(efore.lin) )#31.98648

(mpe.lin=100\*(mean(efore.lin/otest)) )#1.470221

(mse.lin=sum(efore.lin\*\*2)/ntest)#1599.984

(mae.lin=mean(abs(efore.lin)))#35.17683

(mape.lin=100\*(mean(abs((efore.lin)/otest))))#1.620296

timefit=c(109:120)

tsqfit=timefit^2/factorial(2)

matq=matrix(c(timefit,tsqfit),ncol=2,dimnames = list(c(),c("timefit","tsqfit")))

matq

newnq <- data.frame(matq)

pfore.quad=predict(mlr.quad,newnq,se.fit = TRUE)

pfore.quad$fit # point predictions

(efore.quad=otest-pfore.quad$fit)

(me.quad=mean(efore.quad))#206.1312

(mpe.quad=100\*(mean(efore.quad/otest)))#9.531554

(mse.quad=sum(efore.quad\*\*2)/ntest)#44171.91

(mae.quad=mean(abs(efore.quad)))#206.1312

(mape.quad=100\*(mean(abs((efore.quad)/otest))))# 9.531554

timefit=c(109:120)

tsqfit=tfit^2/factorial(2)

tcubfit=tfit^3/factorial(3)

matc=matrix(c(timefit,tsqfit,tcubfit),ncol=3,dimnames = list(c(),c("timefit","tsqfit","tcubfit")))

newnc=data.frame(matc)

pfore.cub=predict(mlr.cub,newnc,se.fit = TRUE)

pfore.cub$fit

efore.cub=otest-pfore.cub$fit

(me.cub=mean(efore.cub))#158.2064

(mpe.cub=100\*(mean(efore.cub/otest)))#7.308646

(mse.cub=sum(efore.cub\*\*2)/ntest)#25983.9

(mae.cub=mean(abs(efore.cub)))#158.2064

(mape.cub=100\*(mean(abs((efore.cub)/otest))))#7.308646

linff=c(mlr.lin$fitted,pfore.lin$fit)

quadff=c(mlr.quad$fitted,pfore.quad$fit)

cubff=c(mlr.cub$fitted,pfore.cub$fit)

obs=c(ofit,otest)

obslin=cbind(obs,linff)

obsquad=cbind(obs,quadff)

obscub=cbind(obs,cubff)

time=c(1:length(obs))

ts.plot(obs,main="Obs. Series")

abline(v=108)

ts.plot(obslin,main="Linear Trend Model")

abline(v=108)

ts.plot(obsquad,main="Quadratic Trend Model")

abline(v=108)

ts.plot(obscub,main="Cubic Trend Model")

abline(v=108)

#fit the linear model to find 1000

goal <- data.frame(timefit=c(121:300))

pfore.lin <- predict(mlr.lin,goal,se.fit = TRUE)

pfore.lin$fit

#fit the cub model to find 1000

timefit=c(121:300)

tsqfit=timefit^2/factorial(2)

tcubfit=timefit^3/factorial(3)

matc=matrix(c(timefit,tsqfit,tcubfit),ncol=3,dimnames = list(c(),c("timefit","tsqfit","tcubfit")))

goalc=data.frame(matc)

pfore.cub=predict(mlr.cub,goalc,se.fit = TRUE)

pfore.cub$fit

#ARIMA model

workdiff1 <- diff(work$WORKORDERS[1:108])

ts.plot(workdiff1)

workdiff12 <- diff(workdiff1)

ts.plot(workdiff12)

acf(workdiff12,max(50))

pacf(workdiff12,max(100))

sarima(work$WORKORDERS[1:108],0,2,1)

foredata <- sarima.for(work$WORKORDERS[1:108], 12, 0, 1, 1)

tresidual <- work$WORKORDERS[109:120]-foredata$pred

sum(tresidual)/12#ME 57.50273 55.82046

100\*mean(tresidual/work$WORKORDERS[109:120])#MPE 2.643544 2.565986

mean(tresidual^2)#3842.004 3651.492

par(mfrow=c(1,1))

sarima.for(work$WORKORDERS[1:108], 200, 0, 2, 1)

#use the whole dataset to bulid ARIMA model

finaldiff1 <- diff(work$WORKORDERS)

ts.plot(finaldiff1)

finaldiff2 <- diff(finaldiff1)

ts.plot(finaldiff2)

mean(finaldiff2)

acf(finaldiff2,max(100))

pacf(finaldiff2,max(50))

sarima(work$WORKORDERS,1,2,1)#6.104306 6.08778 6.191969

sarima.for(work$WORKORDERS, 200, 1, 2, 1)

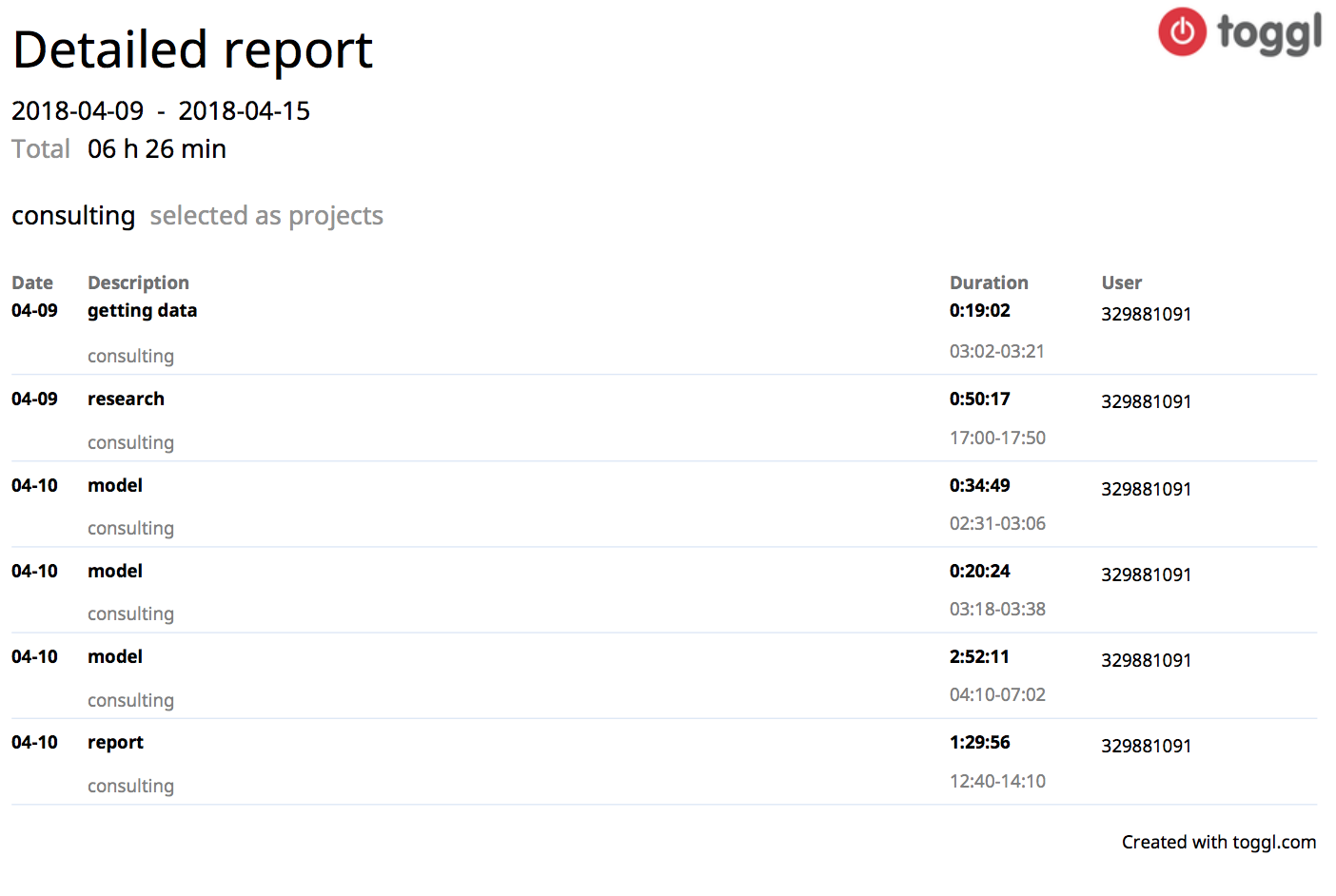
abline(v=249)

summary(work$WORKORDERS)

1. Reference

Time Series Analysis , by R.H. Shumway & D.S. Sto er

**Invoice**



Rate: 50 $/h

Subtotal: 325$

Tax( 6.5%): 21.13$

Total: 346.13$