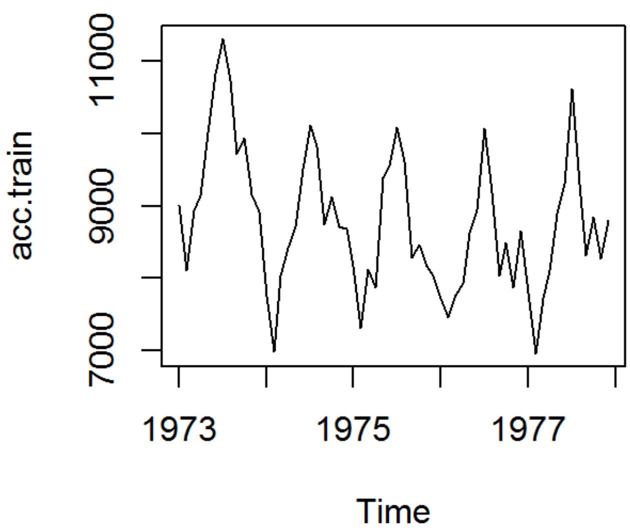
Question 3

Part a

```
# read in the data
acc.all<-USAccDeaths
acc.train<-ts(acc.all[1:60],start=c(1973,1),frequency=12) # convert the training set to time series
acc.test<-ts(acc.all[61:72],start=c(1978,1),frequency=12) # convert the test set to time series
plot(acc.train)
```



Comments: 1. a clear seasonality (frequency= 12 months) can be observed in the plot. 2. There is a slightly quadratic term in the plot.

Part b

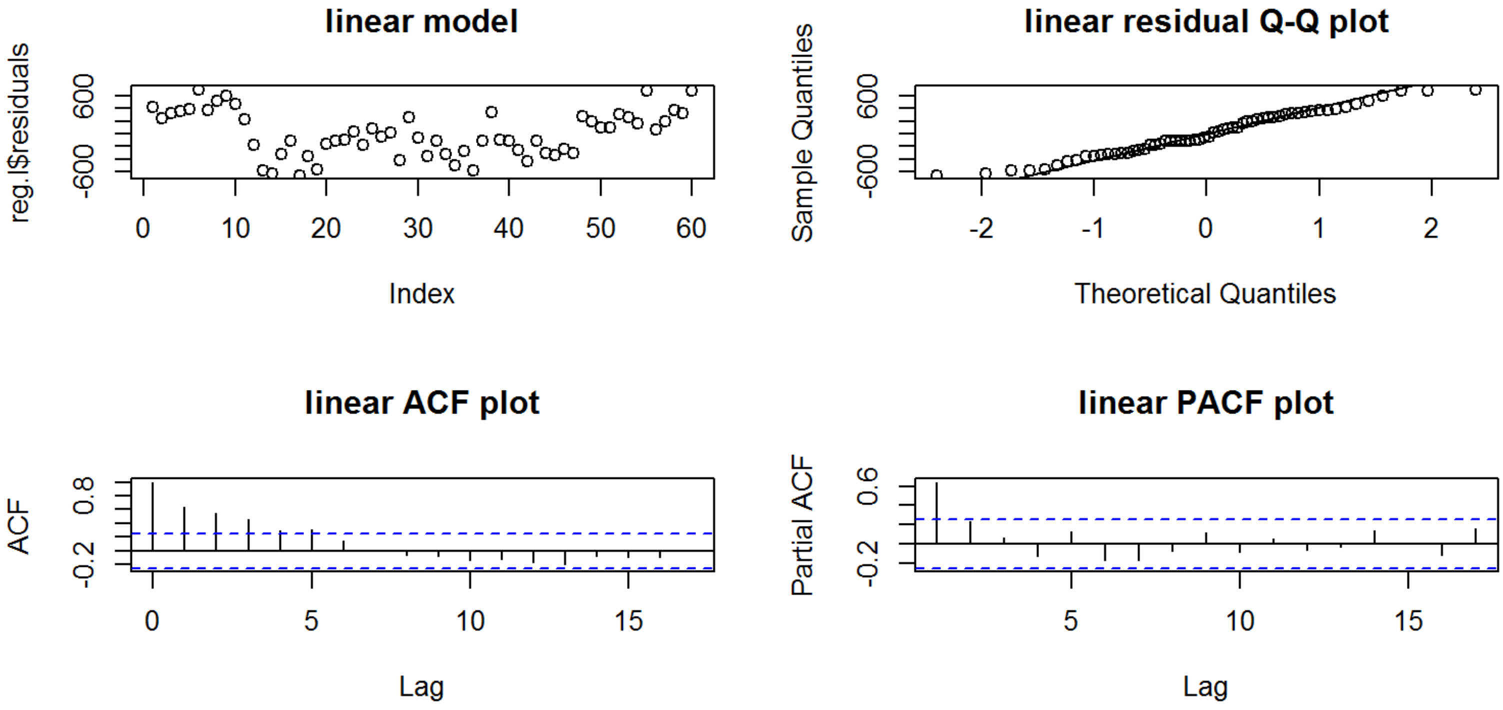
```
month<-factor(cycle(acc.train))
t<-c(1:60)/12
## first, fit the data with linear function and seasonality
reg.l<-lm(acc.train~t+month)
summary(reg.l)
```

```
##
## Call:
## lm(formula = acc.train ~ t + month)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -659.5 -282.3  -63.9  285.3  703.8
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8600.86     194.36  44.253  < 2e-16 ***
## t            -247.32      36.51  -6.774 1.80e-08 ***
## month2       -702.79     252.96  -2.778  0.00783 **
## month3         72.22     253.01   0.285  0.77656
## month4        268.23     253.10   1.060  0.29467
## month5       1123.04     253.23   4.435 5.52e-05 ***
## month6       1645.05     253.40   6.492 4.83e-08 ***
## month7       2484.66     253.60   9.798 6.17e-13 ***
## month8       1792.27     253.83   7.061 6.61e-09 ***
## month9        697.68     254.11   2.746  0.00853 **
## month10      1074.09     254.42   4.222  0.00011 ***
## month11       554.50     254.76   2.177  0.03457 *
## month12       757.91     255.14   2.971  0.00467 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 399.9 on 47 degrees of freedom
## Multiple R-squared:  0.8618, Adjusted R-squared:  0.8265
## F-statistic: 24.42 on 12 and 47 DF,  p-value: 3.382e-16
```

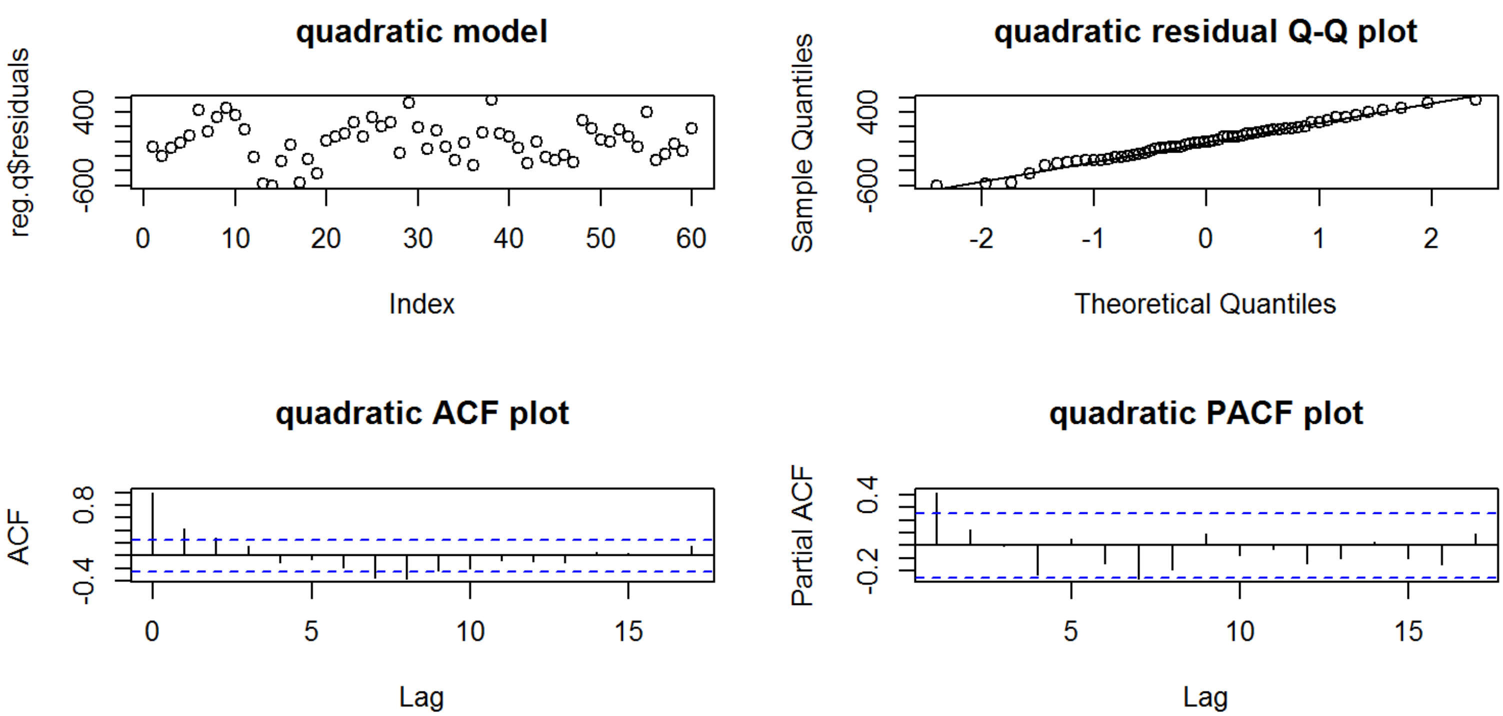
```
## Then, fit the data with quadratic function and seasonality
t2<-t^2
reg.q<-lm(acc.train~t+t2+month)
summary(reg.q)
```

```
##
## Call:
## lm(formula = acc.train ~ t + t2 + month)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -596.29 -189.41   -3.24  170.78  571.80
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9148.93     168.94  54.155  < 2e-16 ***
## t            -902.86     108.28  -8.338 9.42e-11 ***
## t2            128.96      20.62   6.254 1.20e-07 ***
## month2       -693.83     187.98  -3.691 0.000590 ***
## month3         88.34     188.04   0.470 0.640714
## month4        289.72     188.12   1.540 0.130384
## month5       1148.12     188.22   6.100 2.05e-07 ***
## month6       1671.92     188.35   8.876 1.56e-11 ***
## month7       2511.53     188.50  13.324  < 2e-16 ***
## month8       1817.35     188.67   9.632 1.32e-12 ***
## month9        719.18     188.86   3.808 0.000413 ***
## month10      1090.21     189.08   5.766 6.49e-07 ***
## month11       563.46     189.33   2.976 0.004641 **
## month12       757.91     189.60   3.997 0.000230 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 297.2 on 46 degrees of freedom
## Multiple R-squared:  0.9253, Adjusted R-squared:  0.9042
## F-statistic: 43.83 on 13 and 46 DF,  p-value: < 2.2e-16
```

```
## Check the adequacy of model by residual plots
par(mfrow=c(2,2))
plot(reg.l$residuals,main="linear model")
qqnorm(reg.l$residuals, main="linear residual Q-Q plot")
qqline(reg.l$residuals)
acf(reg.l$residuals,main = "linear ACF plot")
acf(reg.l$residuals,type="partial",main="linear PACF plot")
```




```
plot(reg.q$residuals, main="quadratic model")
qqnorm(reg.q$residuals,main="quadratic residual Q-Q plot")
qqline(reg.q$residuals)
acf(reg.q$residuals,main="quadratic ACF plot")
acf(reg.q$residuals,type="partial",main="quadratic PACF plot")
```



Comment: The residual plot for the linear model seems to have a quadratic term, and the Q-Q plot of the quadratic model looks more like a straight line compared to the linear model. Comparing ACF plots, they both have exponential decay in the first few spike, but the quadratic model also has spikes for lag=7&8. And the PACF is similar, with both linear and quadratic model having a spike at lag=1 and the quadratic model having a slightly significant value at lag=7.

```
PI1<-predict.lm(reg.l,newdata=data.frame(t=c(61:72)/12,month=factor(cycle(acc.test))),interval="pre
diction")
PI2<-predict.lm(reg.q,newdata=data.frame(t=c(61:72)/12,t2=(c(61:72)/12)^2,month=factor(cycle(acc.te
st))),interval="prediction")
PI1
```

```
##         fit      lwr       upr
## 1  7343.625 6435.147  8252.103
## 2  6620.225 5711.747  7528.703
## 3  7374.625 6466.147  8283.103
## 4  7550.025 6641.547  8458.503
## 5  8384.225 7475.747  9292.703
## 6  8885.625 7977.147  9794.103
## 7  9704.625 8796.147 10613.103
## 8  8991.625 8083.147  9900.103
## 9  7876.425 6967.947  8784.903
## 10 8232.225 7323.747  9140.703
## 11 7692.025 6783.547  8600.503
## 12 7874.825 6966.347  8783.303
```

```
PI2
```

```
##          fit      lwr       upr
## 1   7891.696 7193.545  8589.847
## 2   7232.775 6529.094  7936.456
## 3   8051.654 7341.878  8761.430
## 4   8291.533 7575.113  9007.953
## 5   9190.212 8466.613  9913.810
## 6   9756.091 9024.795 10487.386
## 7  10639.570 9900.074 11379.065
## 8   9991.049 9242.868 10739.230
## 9   8940.327 8182.991  9697.663
## 10  9360.606 8593.663 10127.550
## 11  8884.885 8107.899  9661.872
## 12  9132.164 8344.716  9919.613
```

```
(acc.test>PI1[,"lwr"] & acc.test<PI1[,"upr"])
```

```
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov
## 1978  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
##        Dec
## 1978 FALSE
```

```
(acc.test>PI2[,"lwr"] & acc.test<PI2[,"upr"])
```

```
##       Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 1978 TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

Comment: The linear model does not predit so well as there are some values failed to fall in the predicted interval. The quadratic model is relatively better.

```
press.l<-sum((PI1[,1]-acc.test)^2)
press.q<-sum((PI2[,1]-acc.test)^2)
press.l
```

```
## [1] 8014530
```

```
press.q
```

```
## [1] 545905.4
```

The PRESS value also indicates that quadratic model is the better fit.

Part c

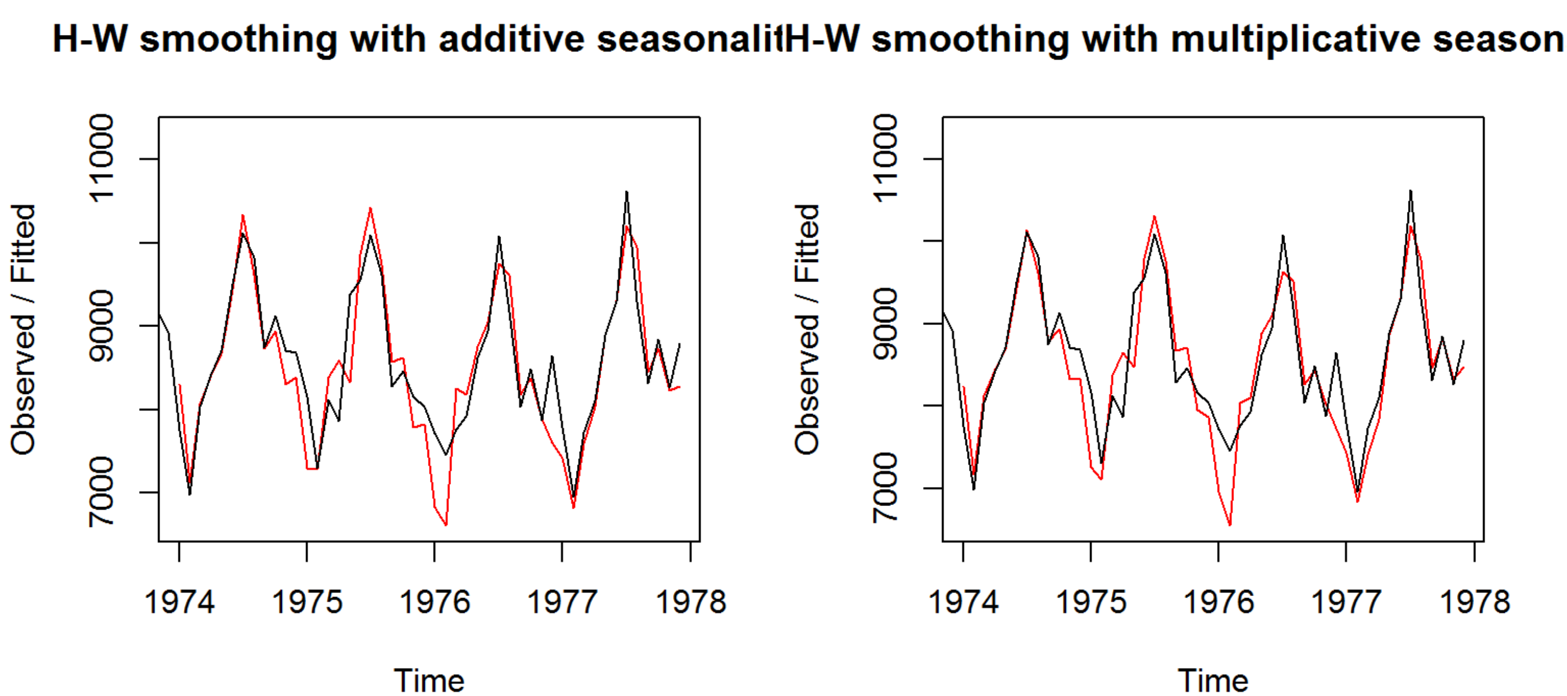
```
acc.add<-HoltWinters(acc.train,seasonal="add")
acc.mult<-HoltWinters(acc.train,seasonal="mult")
acc.add
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = acc.train, seasonal = "add")
##
## Smoothing parameters:
##  alpha: 0.7309863
##  beta : 0.01149582
##  gamma: 1
##
## Coefficients:
##             [,1]
## a     8449.64745
## b      -56.24708
## s1    -857.13900
## s2   -1729.13010
## s3   -1000.10708
## s4    -608.37577
## s5     236.25736
## s6     709.12215
## s7    1791.15986
## s8     996.58684
## s9     169.22640
## s10    681.17476
## s11    131.78116
## s12    346.35255
```

```
acc.mult
```

```
## Holt-Winters exponential smoothing with trend and multiplicative seasonal component.
##
## Call:
## HoltWinters(x = acc.train, seasonal = "mult")
##
## Smoothing parameters:
##  alpha: 0.5708687
##  beta : 0.01454266
##  gamma: 0.8343467
##
## Coefficients:
##            [,1]
## a   8320.5057273
## b    -52.4558162
## s1     0.9234349
## s2     0.8237593
## s3     0.8977443
## s4     0.9304070
## s5     1.0268687
## s6     1.0814210
## s7     1.2101157
## s8     1.1032012
## s9     1.0003856
## s10    1.0683017
## s11    1.0088240
## s12    1.0543607
```

```
par(mfcol=c(1,2))
plot(acc.add,main="H-W smoothing with additive seasonality")
plot(acc.mult,main="H-W smoothing with multiplicative seasonality")
```



```
acc.add$SSE
```

```
## [1] 7988469
```

```
acc.mult$SSE
```

```
## [1] 7094263
```

```
PI.add<-predict(acc.add,12,prediction.interval=TRUE)
PI.mult<-predict(acc.mult,12,prediction.interval=TRUE)
PI.add
```

```
##                fit       upr      lwr
## Jan 1978 7536.261  8333.281 6739.242
## Feb 1978 6608.023  7599.246 5616.800
## Mar 1978 7280.799  8437.409 6124.189
## Apr 1978 7616.283  8920.506 6312.061
## May 1978 8404.669  9844.241 6965.098
## Jun 1978 8821.287 10387.155 7255.419
## Jul 1978 9847.078 11532.253 8161.902
## Aug 1978 8996.258 10795.168 7197.347
## Sep 1978 8112.650 10020.743 6204.557
## Oct 1978 8568.351 10581.837 6554.866
## Nov 1978 7962.711 10078.387 5847.035
## Dec 1978 8121.035 10336.163 5905.907
```

```
PI.mult
```

```
##                fit       upr      lwr
## Jan 1978 7635.006  8331.618 6938.394
## Feb 1978 6767.672  7564.783 5970.560
## Mar 1978 7328.411  8273.273 6383.549
## Apr 1978 7546.235  8613.069 6479.402
## May 1978 8274.741  9512.866 7036.616
## Jun 1978 8657.609 10032.114 7283.103
## Jul 1978 9624.431 11214.639 8034.223
## Aug 1978 8716.237 10261.410 7171.065
## Sep 1978 7851.430  9350.646 6352.213
## Oct 1978 8328.424  9991.263 6665.584
## Nov 1978 7811.820  9467.590 6156.050
## Dec 1978 8109.126  9763.559 6454.693
```

```
(acc.test>PI.add[,"lwr"] & acc.test<PI.add[,"upr"])
```

```
##       Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 1978 TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
(acc.test>PI.mult[,"lwr"] & acc.test<PI.mult[,"upr"])
```

```
##       Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 1978 TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

Comment: The two models both predit the values well, as the forecast values all lie in the prediction interval.

```
press.add<-sum((PI.add[,1]-acc.test)^2)
press.mult<-sum((PI.mult[,1]-acc.test)^2)
press.add
```

```
## [1] 5685759
```

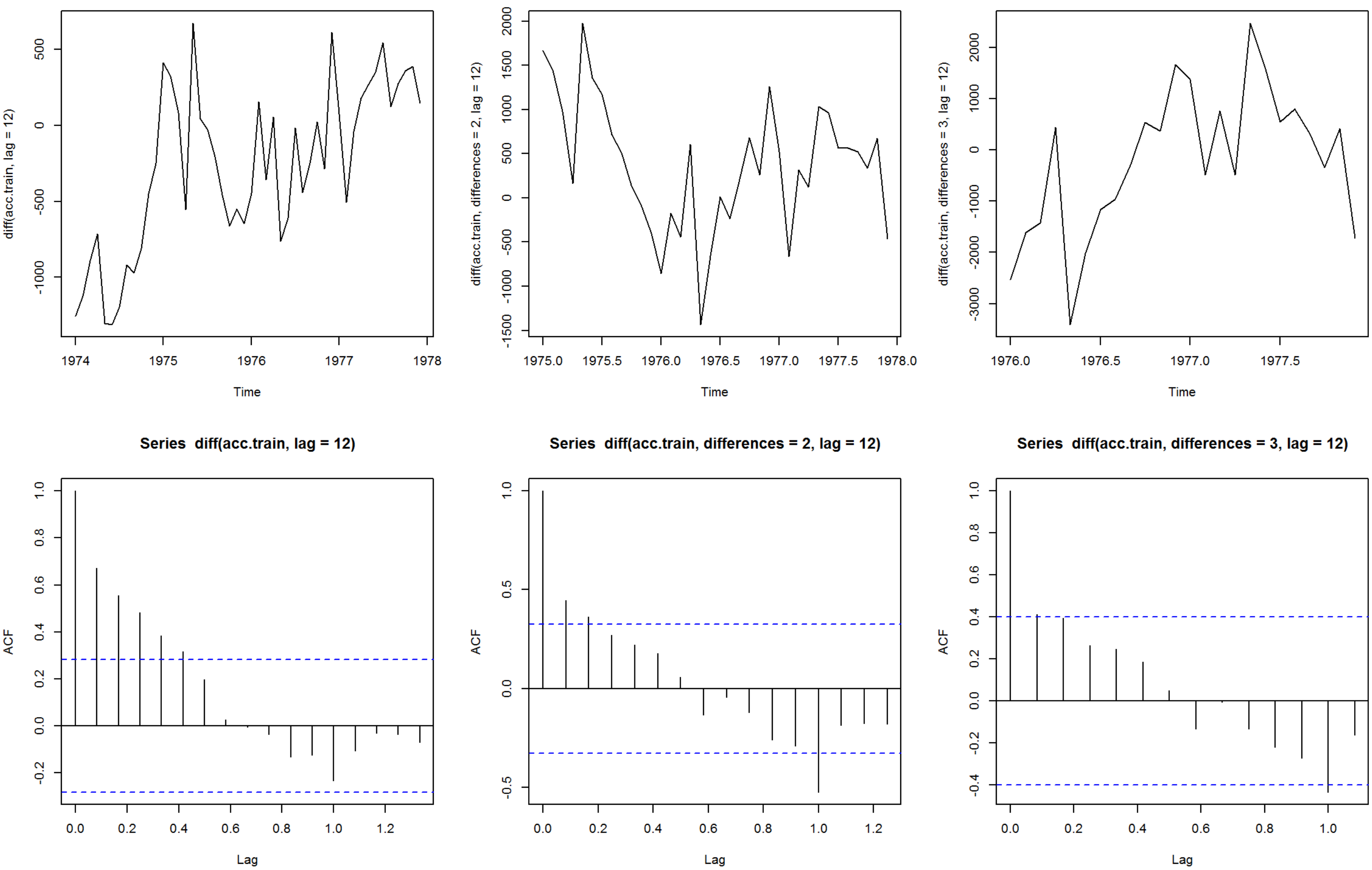
```
press.mult
```

```
## [1] 8055476
```

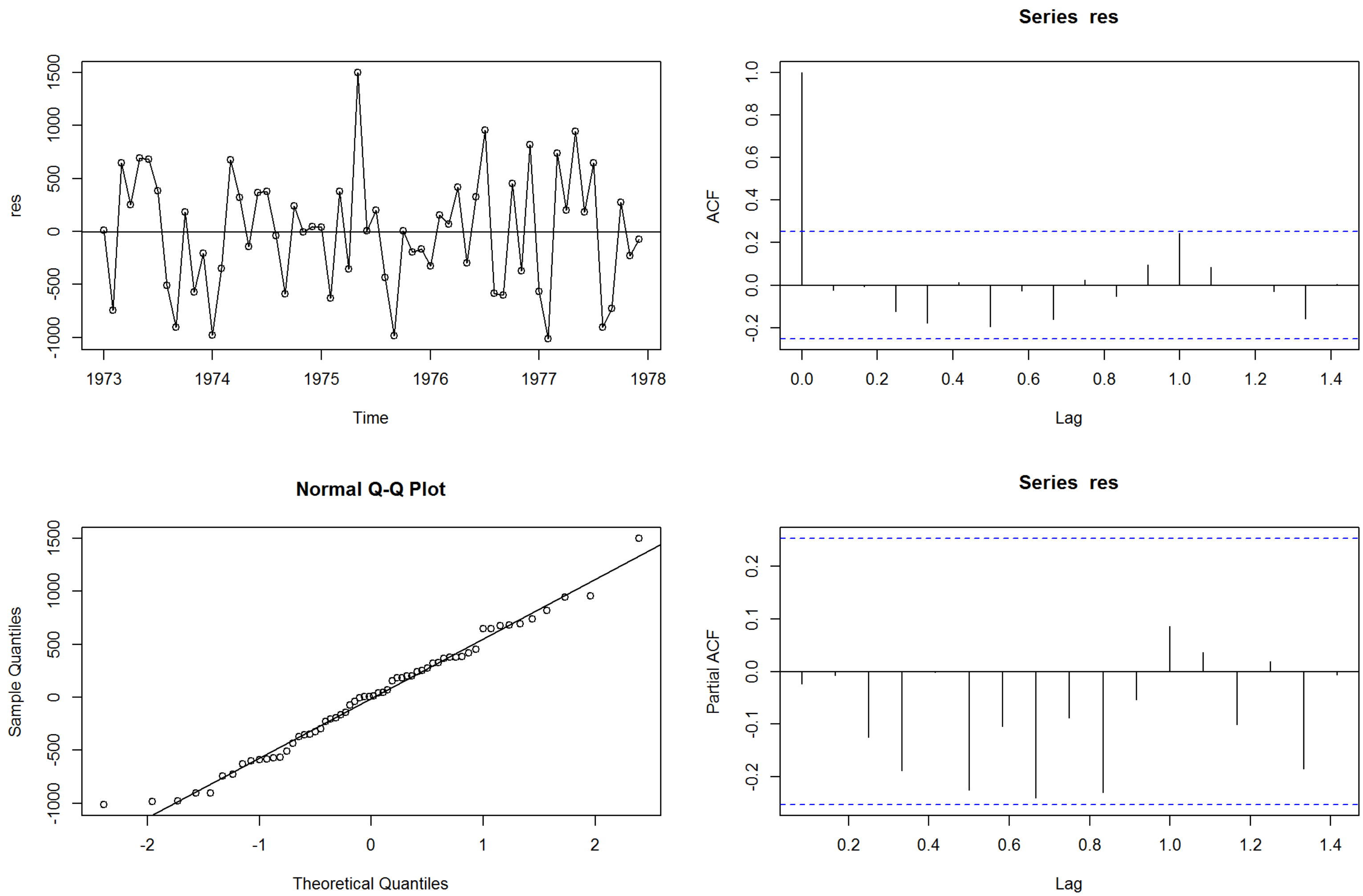
Comment: Since the PRESS value for additive model is smaller, it seems to be a better fit.

Part d

```
par(mfcol=c(2,3))
plot(diff(acc.train,lag=12))
acf(diff(acc.train,lag=12))
plot(diff(acc.train,differences=2,lag=12))
acf(diff(acc.train,differences=2,lag=12))
plot(diff(acc.train,differences=3,lag=12))
acf(diff(acc.train,differences=3,lag=12))
```




```
# try differencing once with MA model
df2ma2 = arima(acc.train, order=c(0,1,2),seasonal=list(order=c(0,0,1), period=12),method="ML")
par(mfcol=c(2,2))
res = df2ma2$residuals
ts.plot(res)
points(res)
abline(h=mean(res))
qqnorm(res)
qqline(res)
acf(res)
acf(res, type="partial")
```



Comment: The residual plots look random. Besides, it has the smallest aic value compared to any other models. So we pick this one to forecast the data.

```
sarimaPI<-predict(df2ma2,n.head=12,newdata=data.frame(t=c(61:72)/12,month=factor(cycle(acc.test))),
interval="prediction",nahead=12)
sarimaPI
```

```
## $pred
##           Jan
## 1978 8363.964
##
## $se
##           Jan
## 1978 556.0492
```

```
#(acc.test>sarimaPI[,"lwr"] & acc.test<sarimaPI[,"upr"])
```

There is something wrong with the code which we don't really know… It is supposed to be 12 values but we only got 2.

part e

The data should be fit into the quadratic model with seasonalities. It has the relatively low value of PRESS compared to any other models. Therefore, the model and prediction interval for the forecasting data is shown in the following r output:

```r
summary(reg.q)
```

```
##
## Call:
## lm(formula = acc.train ~ t + t2 + month)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -596.29 -189.41   -3.24  170.78  571.80
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9148.93     168.94  54.155  < 2e-16 ***
## t            -902.86     108.28  -8.338 9.42e-11 ***
## t2            128.96      20.62   6.254 1.20e-07 ***
## month2       -693.83     187.98  -3.691 0.000590 ***
## month3         88.34     188.04   0.470 0.640714
## month4        289.72     188.12   1.540 0.130384
## month5       1148.12     188.22   6.100 2.05e-07 ***
## month6       1671.92     188.35   8.876 1.56e-11 ***
## month7       2511.53     188.50  13.324  < 2e-16 ***
## month8       1817.35     188.67   9.632 1.32e-12 ***
## month9        719.18     188.86   3.808 0.000413 ***
## month10      1090.21     189.08   5.766 6.49e-07 ***
## month11       563.46     189.33   2.976 0.004641 **
## month12       757.91     189.60   3.997 0.000230 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 297.2 on 46 degrees of freedom
## Multiple R-squared:  0.9253, Adjusted R-squared:  0.9042
## F-statistic: 43.83 on 13 and 46 DF,  p-value: < 2.2e-16
```

```r
PI2
```

```
##          fit      lwr       upr
## 1   7891.696 7193.545  8589.847
## 2   7232.775 6529.094  7936.456
## 3   8051.654 7341.878  8761.430
## 4   8291.533 7575.113  9007.953
## 5   9190.212 8466.613  9913.810
## 6   9756.091 9024.795 10487.386
## 7  10639.570 9900.074 11379.065
## 8   9991.049 9242.868 10739.230
## 9   8940.327 8182.991  9697.663
## 10  9360.606 8593.663 10127.550
## 11  8884.885 8107.899  9661.872
## 12  9132.164 8344.716  9919.613
```

```r
(acc.test>PI2[,"lwr"] & acc.test<PI2[,"upr"])
```

```
##       Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 1978 TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```