Japanese tourists number

Suixin Jiang

3/7/2020

Import data

```r
jp <- read.csv('TSA.csv')
names(jp)
```

```
## [1] "Year"                "Month"               "Amounts.of.tourists"
## [4] "CNY.JPY"             "Average.temperature" "Shopping.month"
## [7] "Consumption.rate"
```

Time-series variables and then run a linear regression model

```r
tour=ts(jp$Amounts.of.tourists,frequency=12,start=c(2010,1),end=c(2018,12))
ex.rate=ts(jp$CNY.JPY,frequency=12,start=c(2010,1),end=c(2018,12))
temp=ts(jp$Average.temperature,frequency=12,start=c(2010,1),end=c(2018,12))
shopmon=ts(jp$Shopping.month,frequency=12,start=c(2010,1),end=c(2018,12))
con.rate=ts(jp$Consumption.rate,frequency=12,start=c(2010,1),end=c(2018,12))

lm.mod=lm(tour~ex.rate+temp+as.factor(shopmon)+con.rate,data=jp)
summary(lm.mod)
```

```
##
## Call:
## lm(formula = tour ~ ex.rate + temp + as.factor(shopmon) + con.rate,
##     data = jp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.171 -20.756  -5.683  20.761  73.642
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           56.8649    24.8116   2.292 0.023948 *
## ex.rate               -3.6781     1.9726  -1.865 0.065087 .
## temp                  -0.2287     0.2170  -1.054 0.294325
## as.factor(shopmon)1    0.2903     6.6953   0.043 0.965497
## con.rate            1047.4255   308.9525   3.390 0.000991 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.05 on 103 degrees of freedom
## Multiple R-squared:  0.114,  Adjusted R-squared:  0.07957
## F-statistic: 3.313 on 4 and 103 DF,  p-value: 0.0135
```

```r
dwtest(lm.mod)
```

```
##
```

```
##  Durbin-Watson test
##
## data:  lm.mod
## DW = 1.4229, p-value = 0.0005287
## alternative hypothesis: true autocorrelation is greater than 0
```

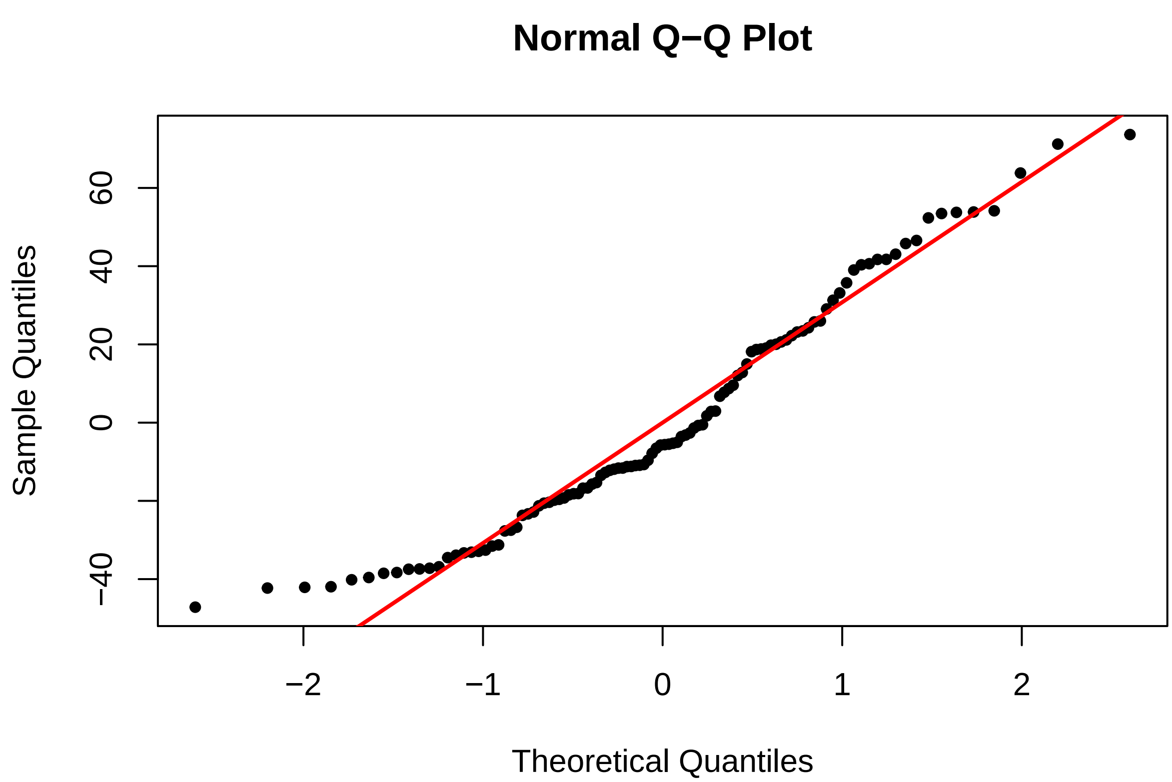
The dwtest p-value is very small which indicates autocorrelation problem

Residuals analysis

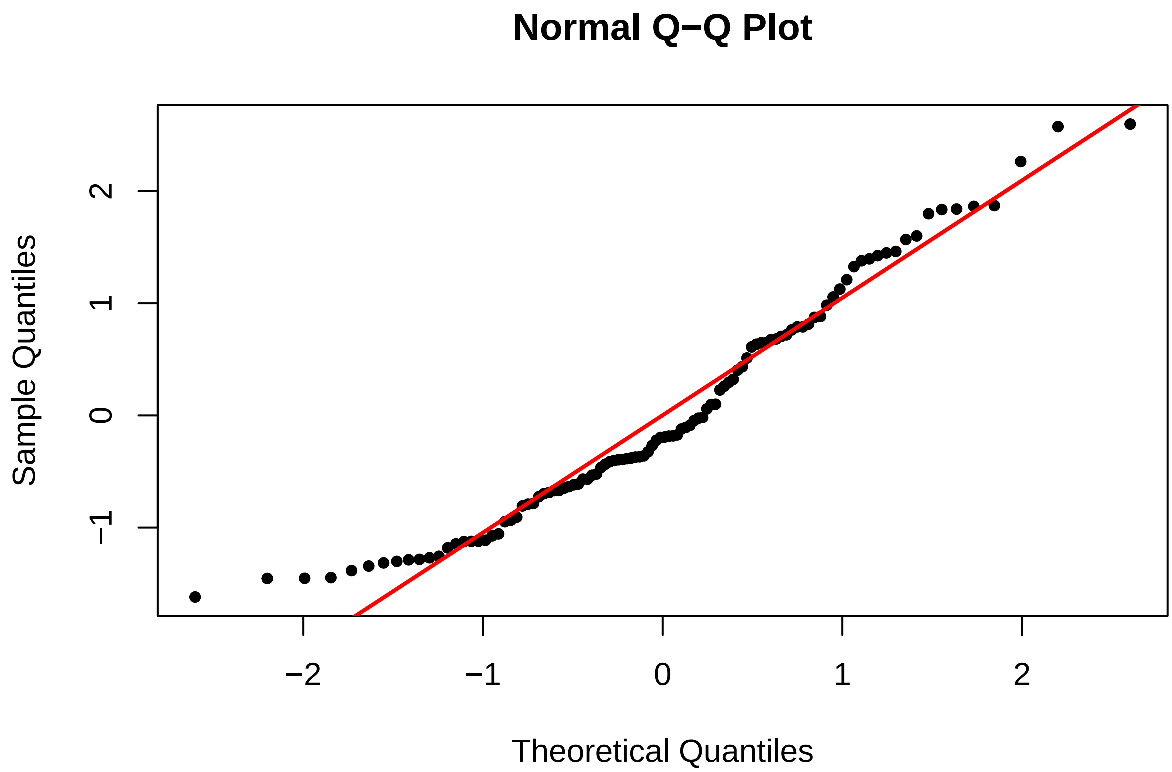
```r
reslm=lm.mod$residuals
studlm=studres(lm.mod)
fit=lm.mod$fitted.values
```

QQ plot

```r
qqnorm(reslm,pch=20)
qqline(reslm,col='red',lwd=2)
```

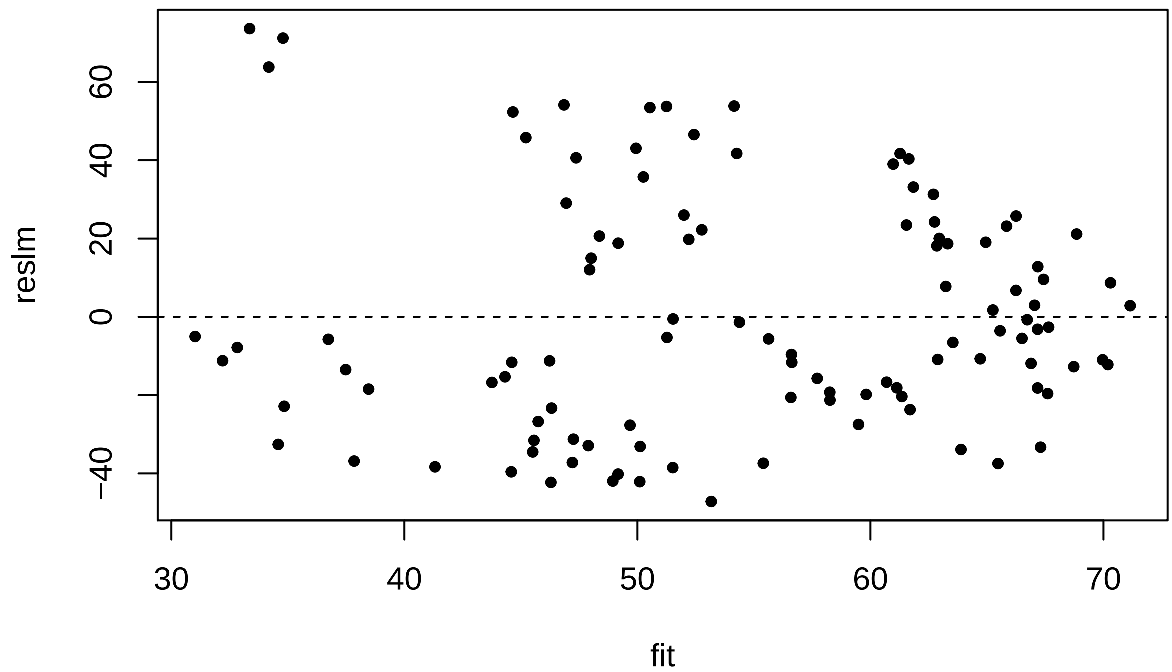



```r
qqnorm(studlm,pch=20)
qqline(studlm,col='red',lwd=2)
```

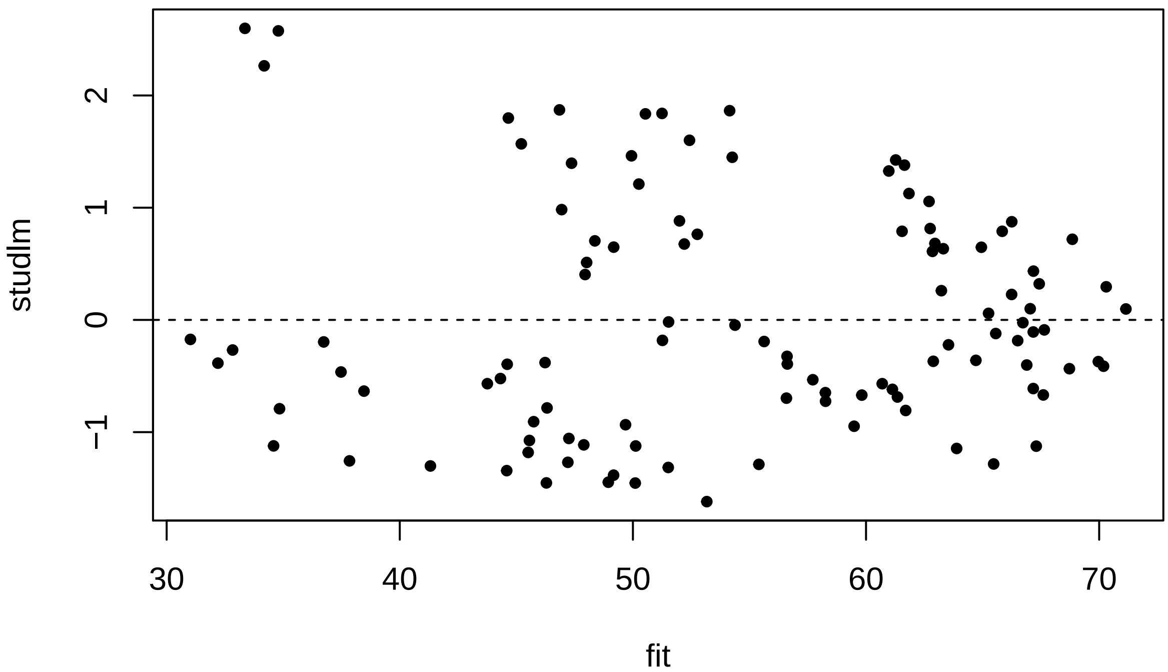


Fitted vs. residuals

```
plot(reslm~fit,pch=20)
abline(h=0,lty=2)
```

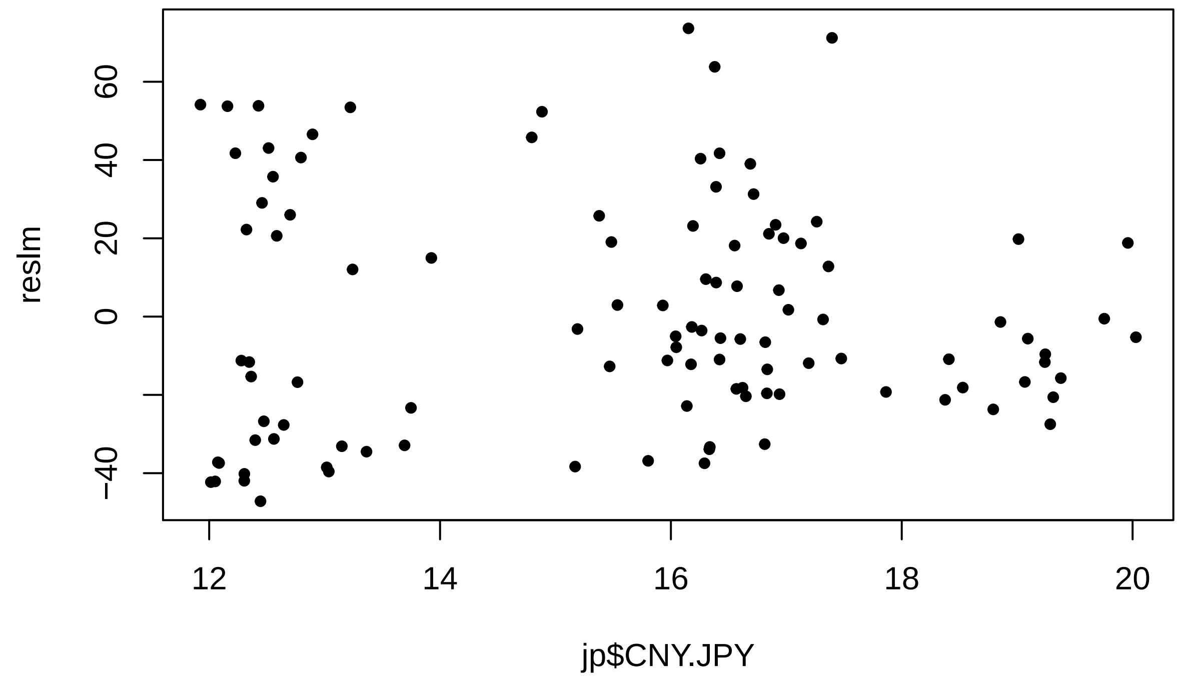



```
plot(studlm~fit,pch=20)
abline(h=0,lty=2)
```

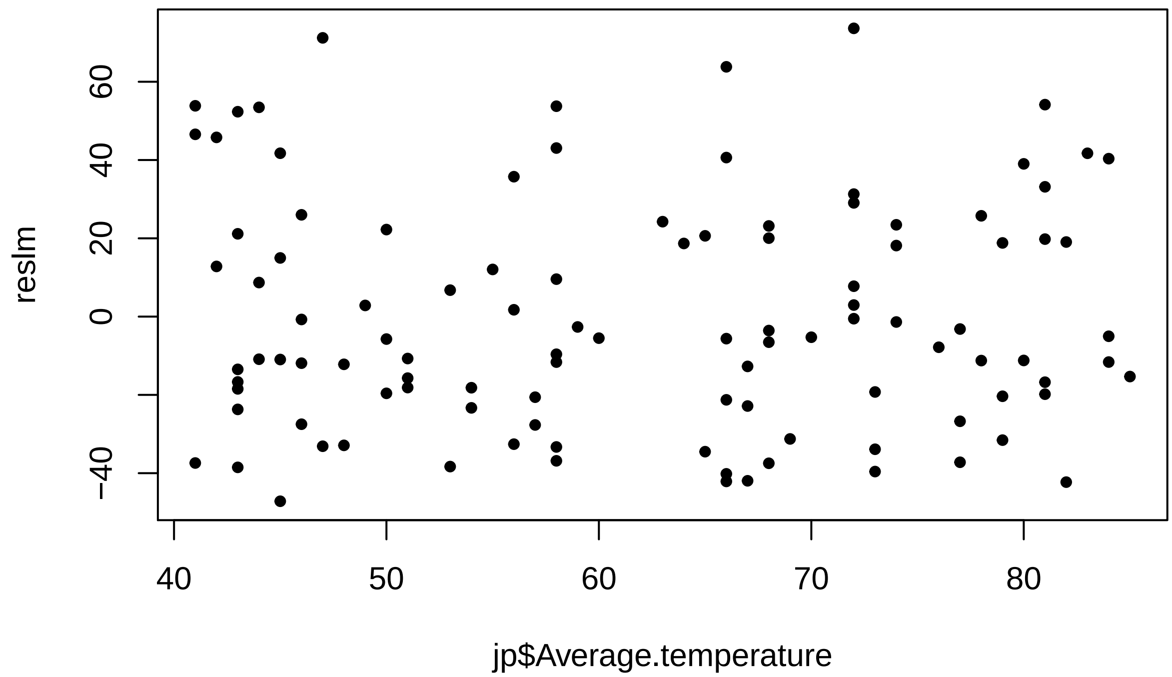


Non-constant variance check

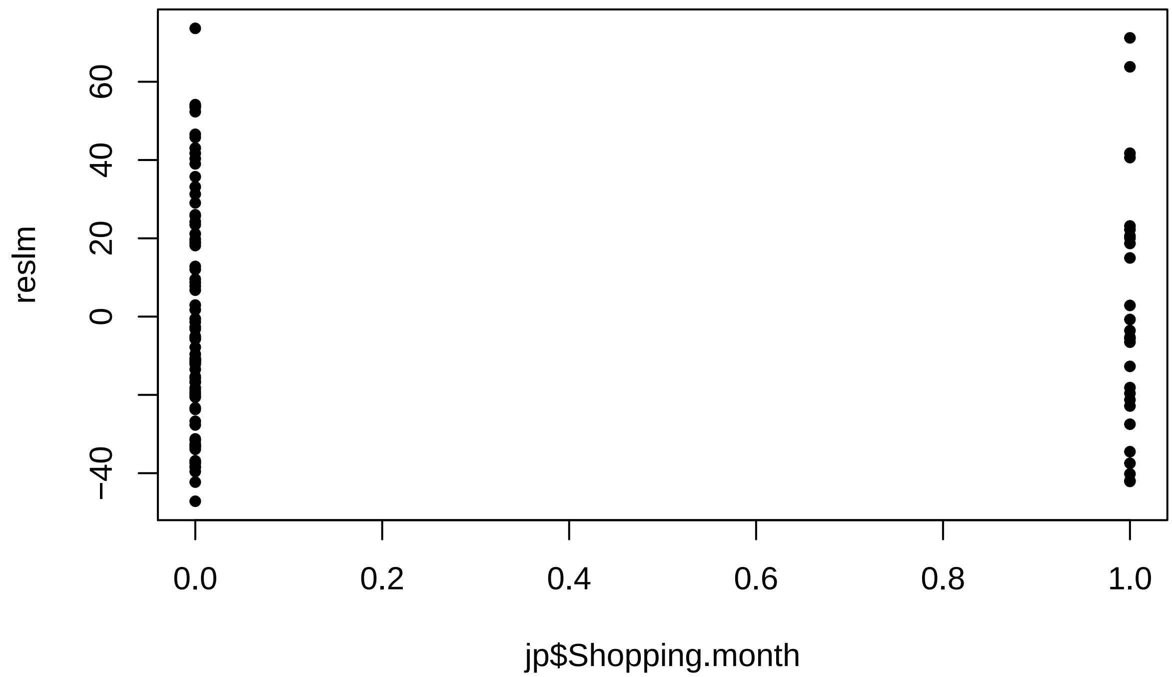
```
plot(reslm~jp$CNY.JPY,pch=20)
```



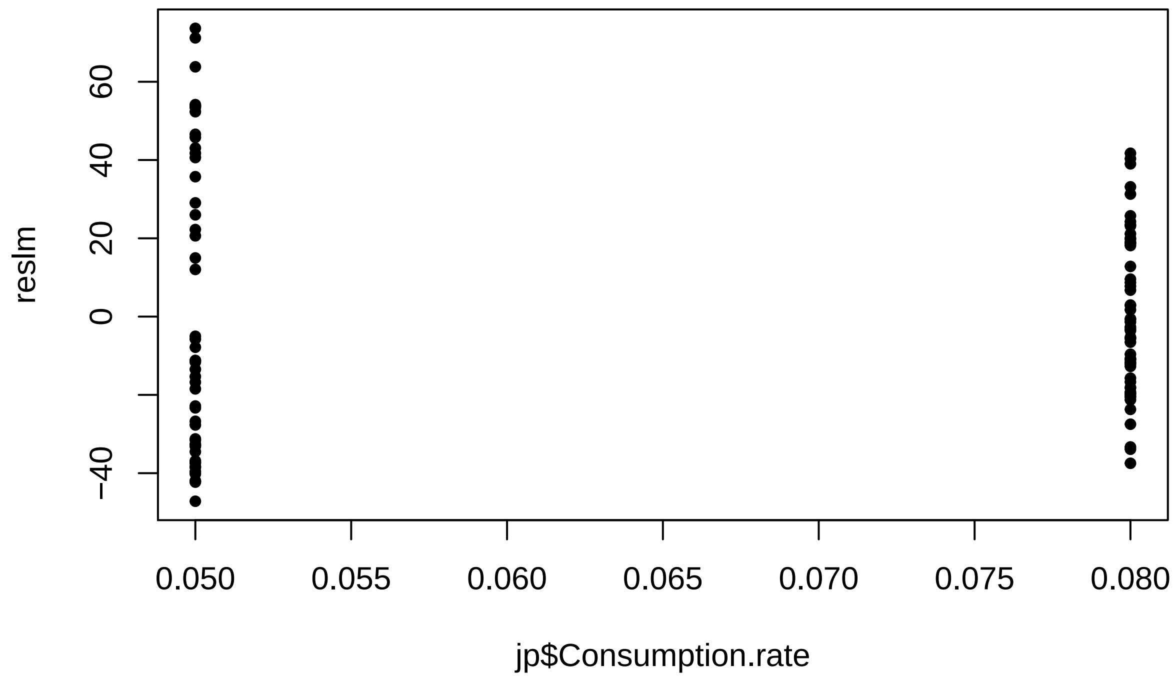

```
plot(reslm~jp$Average.temperature,pch=20)
```




```
plot(reslm~jp$Shopping.month,pch=20)
```



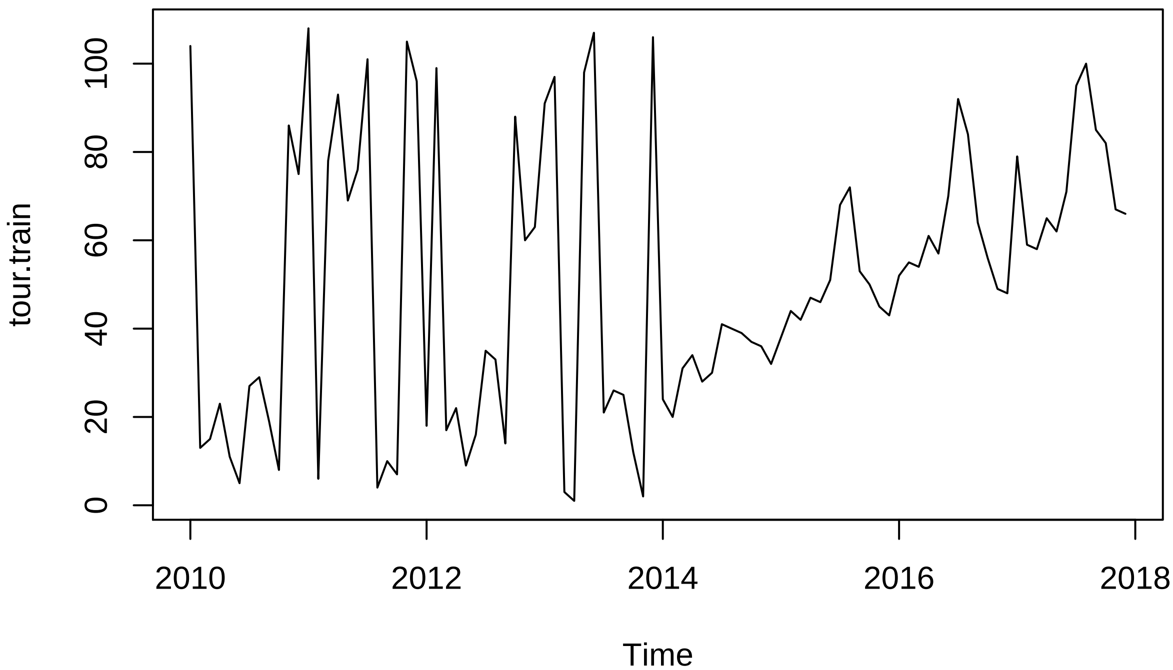

```
plot(reslm~jp$Consumption.rate,pch=20)
```



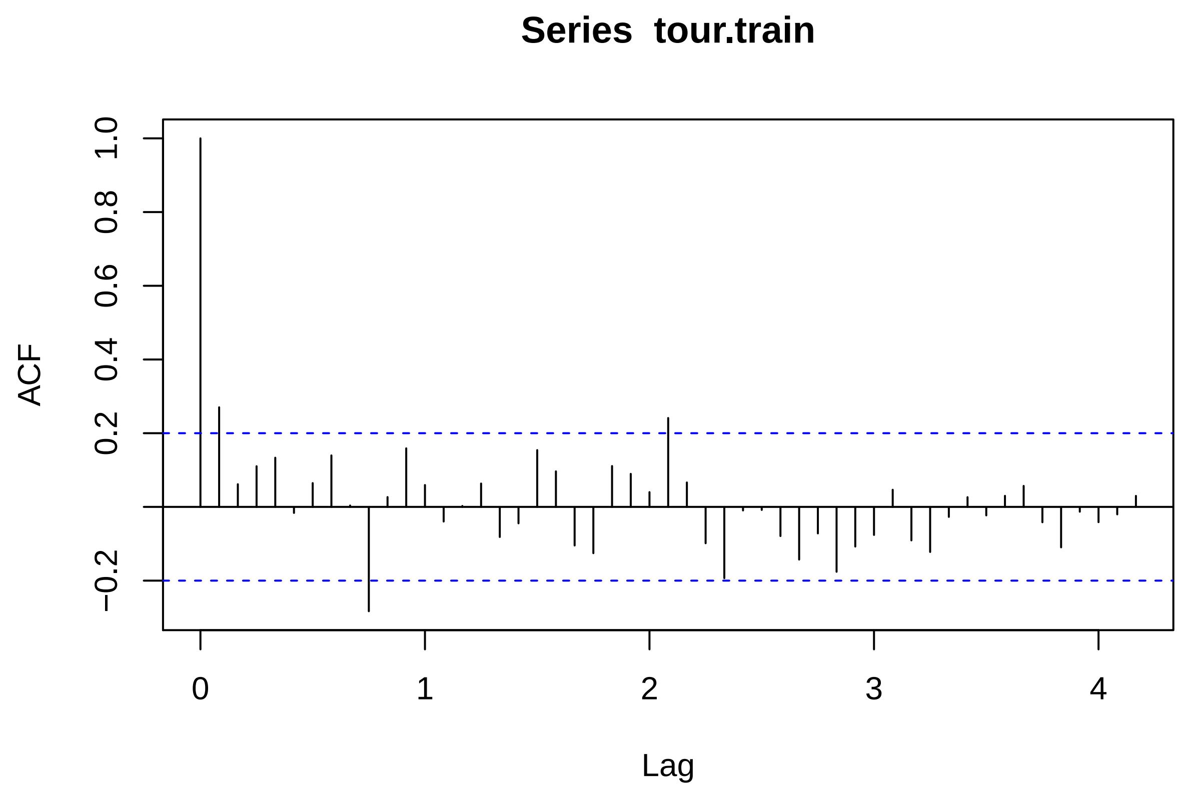
Since the R-squared is to small then considering ARIMA model

Split the data into training and testing sets

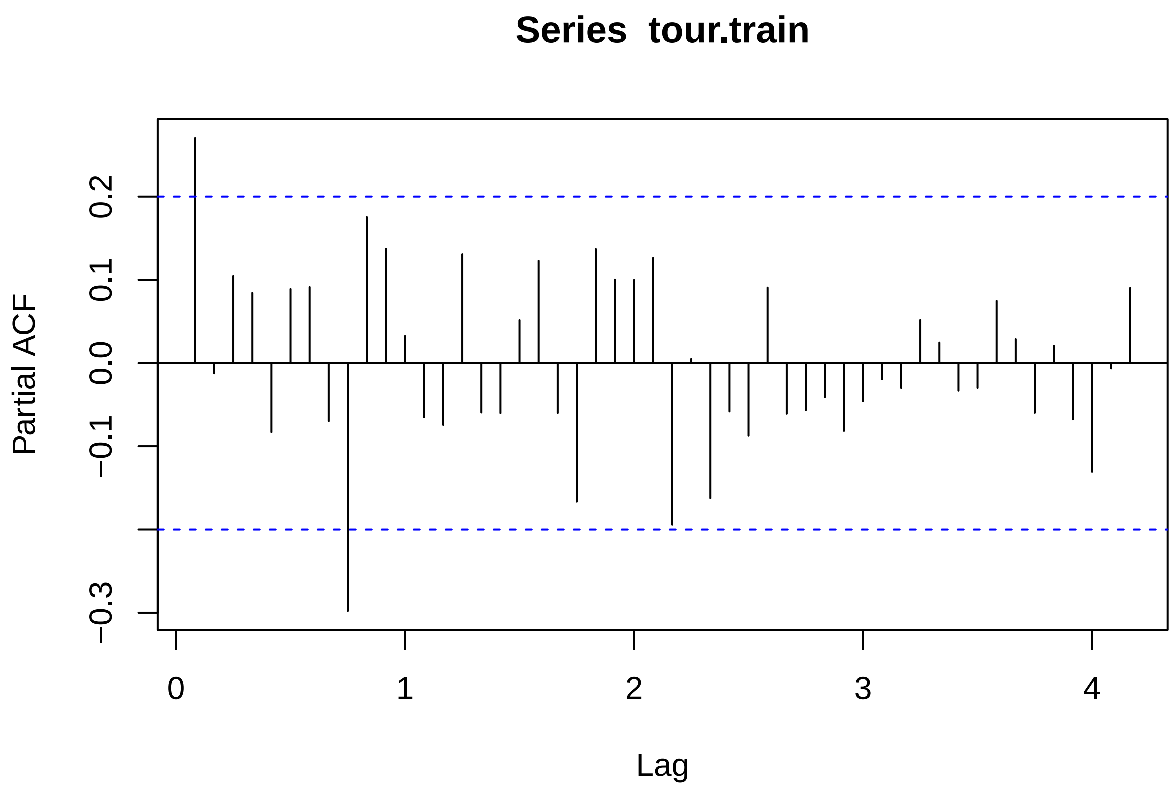
```
tour.train=ts(jp$Amounts.of.tourists[1:96],frequency=12,
    start=c(2010,1),end=c(2017,12))
tour.test=ts(jp$Amounts.of.tourists[97:108],frequency=12,
    start=c(2018,1),end=c(2018,12))
plot(tour.train)
```




```
acf(tour.train, lag.max=50)
```

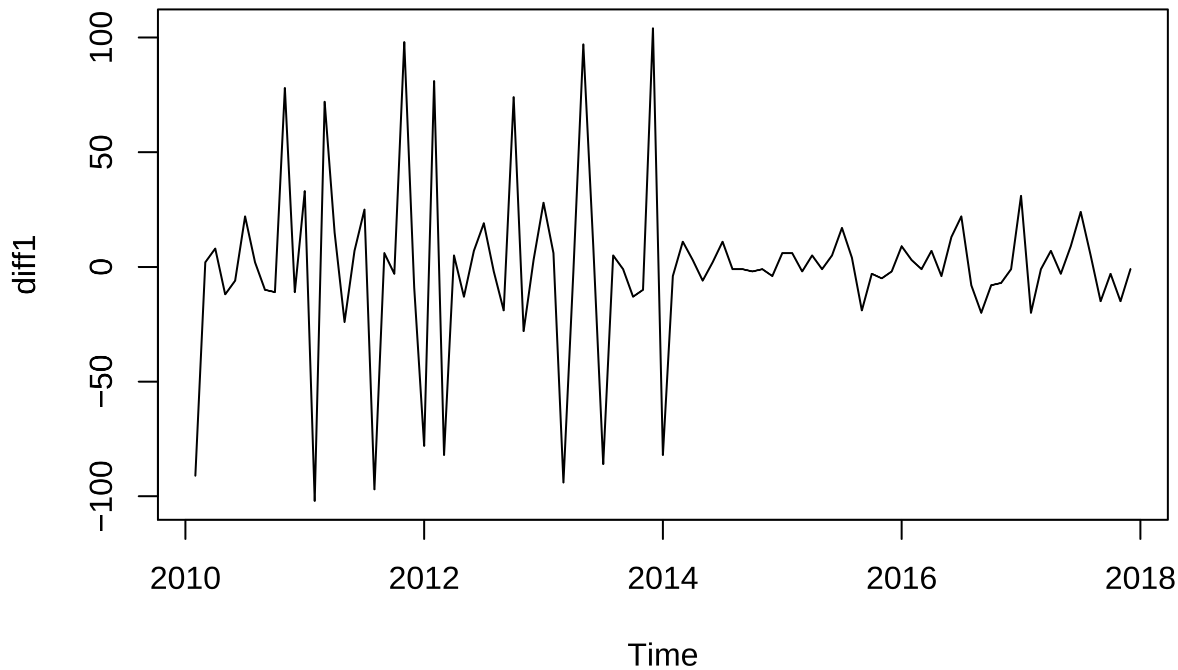



```
pacf(tour.train, lag.max=50)
```

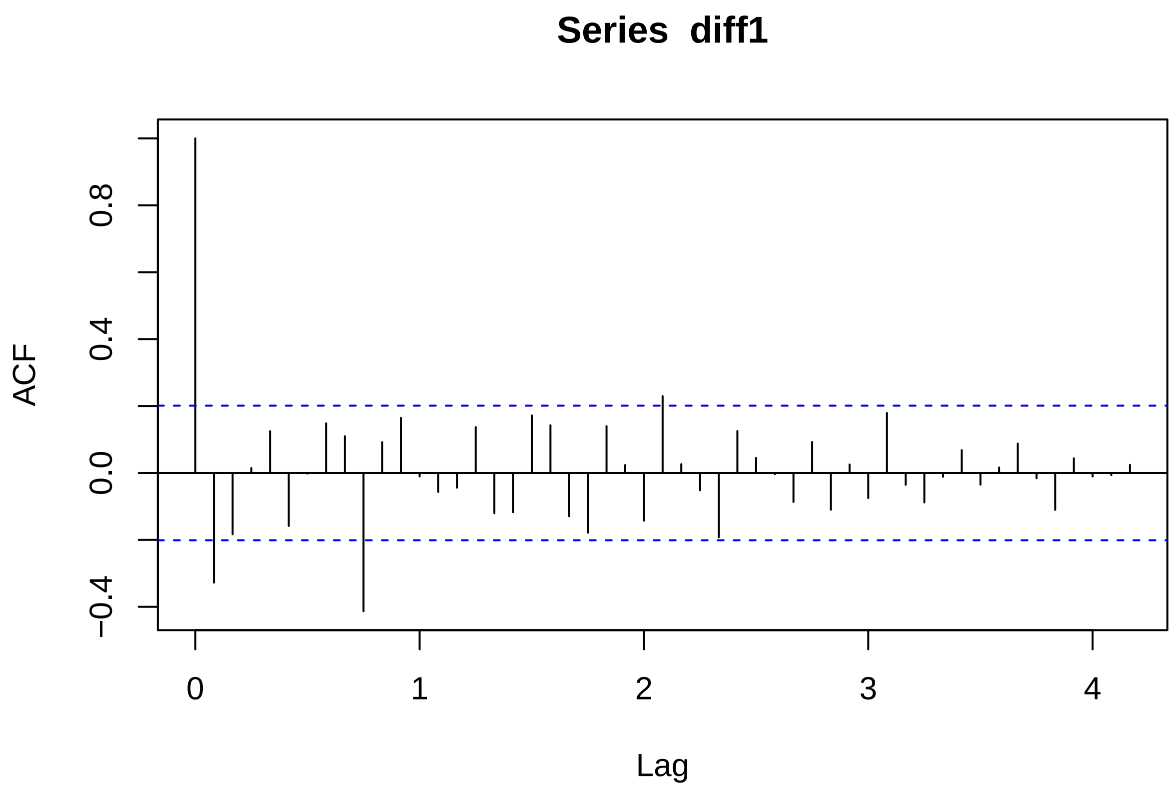


Plots show seasonality and trend, so we make the 1st differencing to remove trend

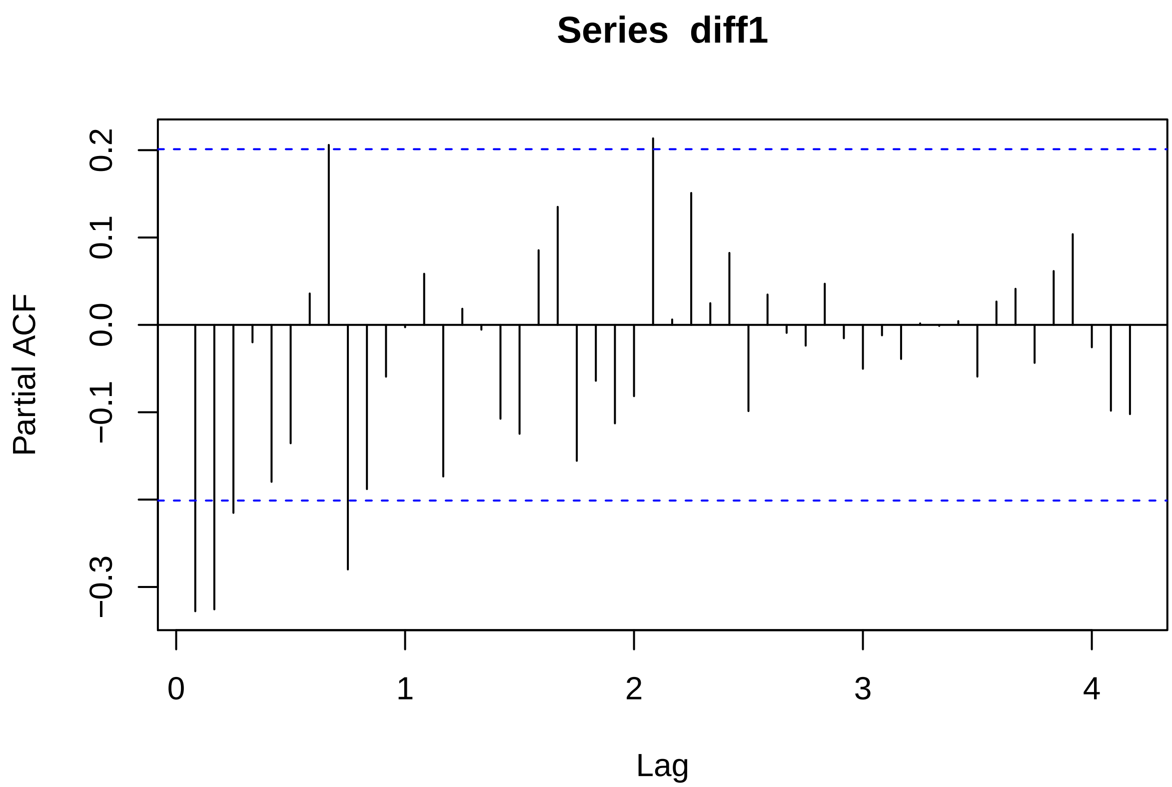
```
diff1=diff(tour.train)
plot(diff1)
```




```r
acf(diff1,lag.max=50)
```




```
pacf(diff1,lag.max=50)
```

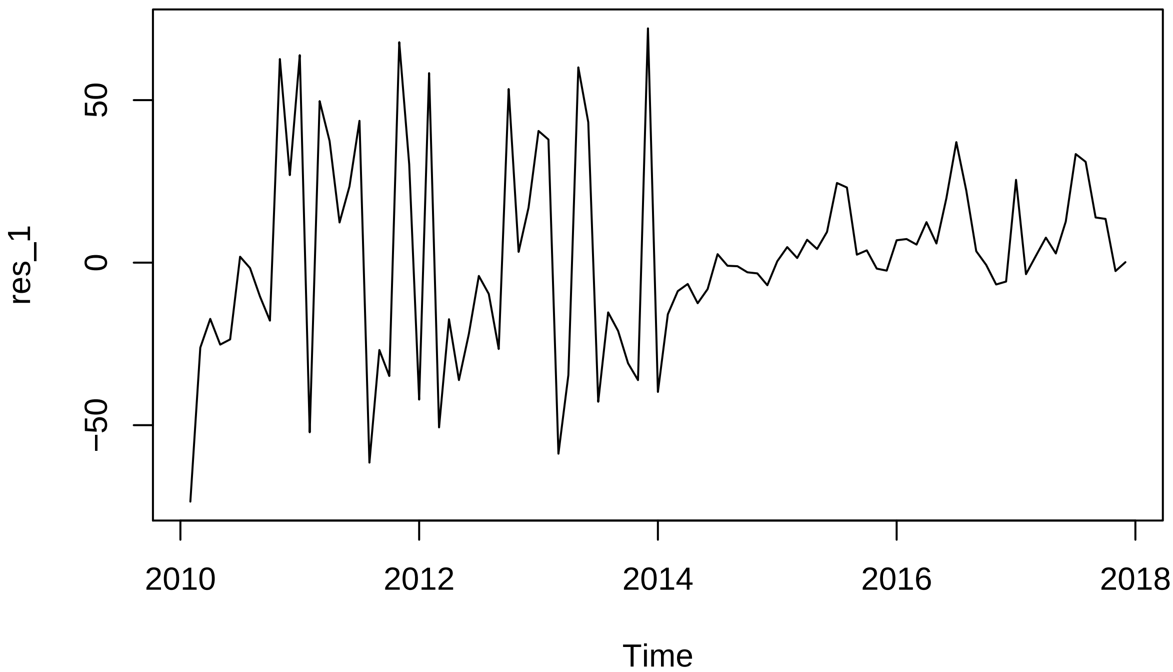



```r
model_1=auto.arima(diff1)
model_1
```

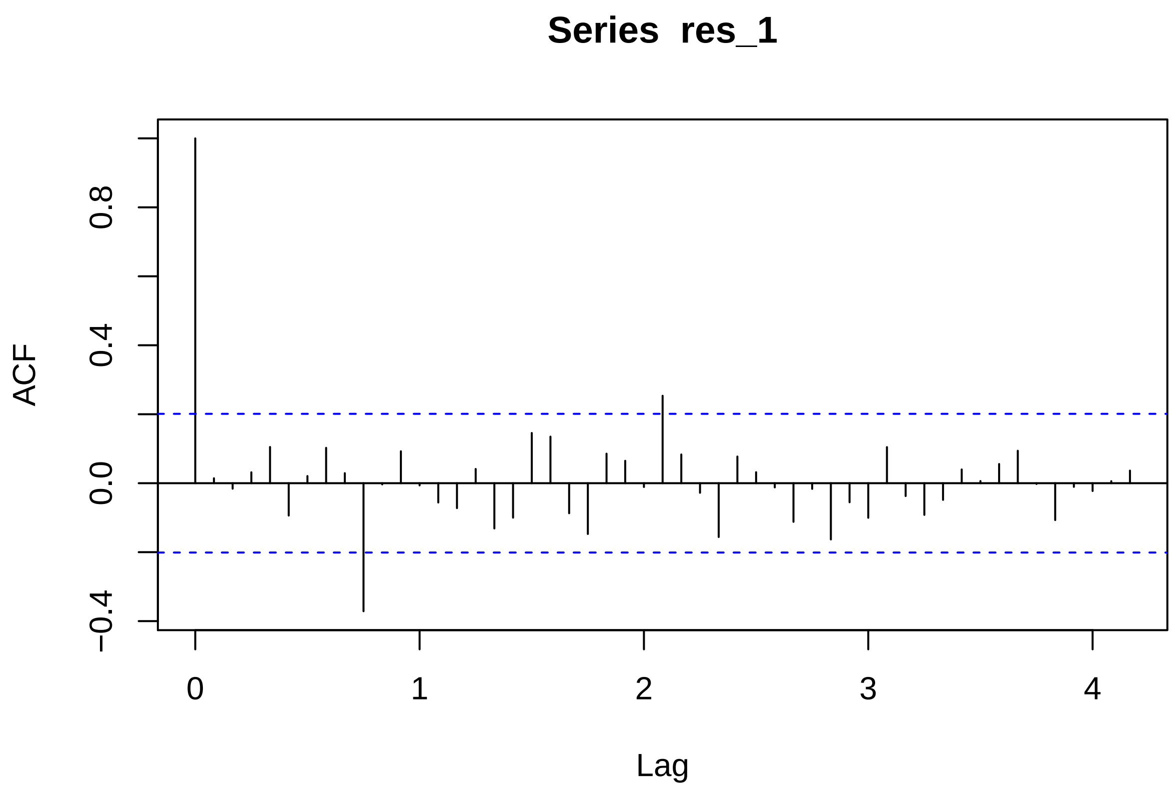
```
## Series: diff1
## ARIMA(0,0,2) with zero mean
##
## Coefficients:
##           ma1      ma2
##       -0.6958  -0.2190
## s.e.   0.1055   0.1078
##
## sigma^2 estimated as 900.7:  log likelihood=-457.77
## AIC=921.55   AICc=921.81   BIC=929.21
```

R suggests that model_1 is ARIMA(0,0,2)

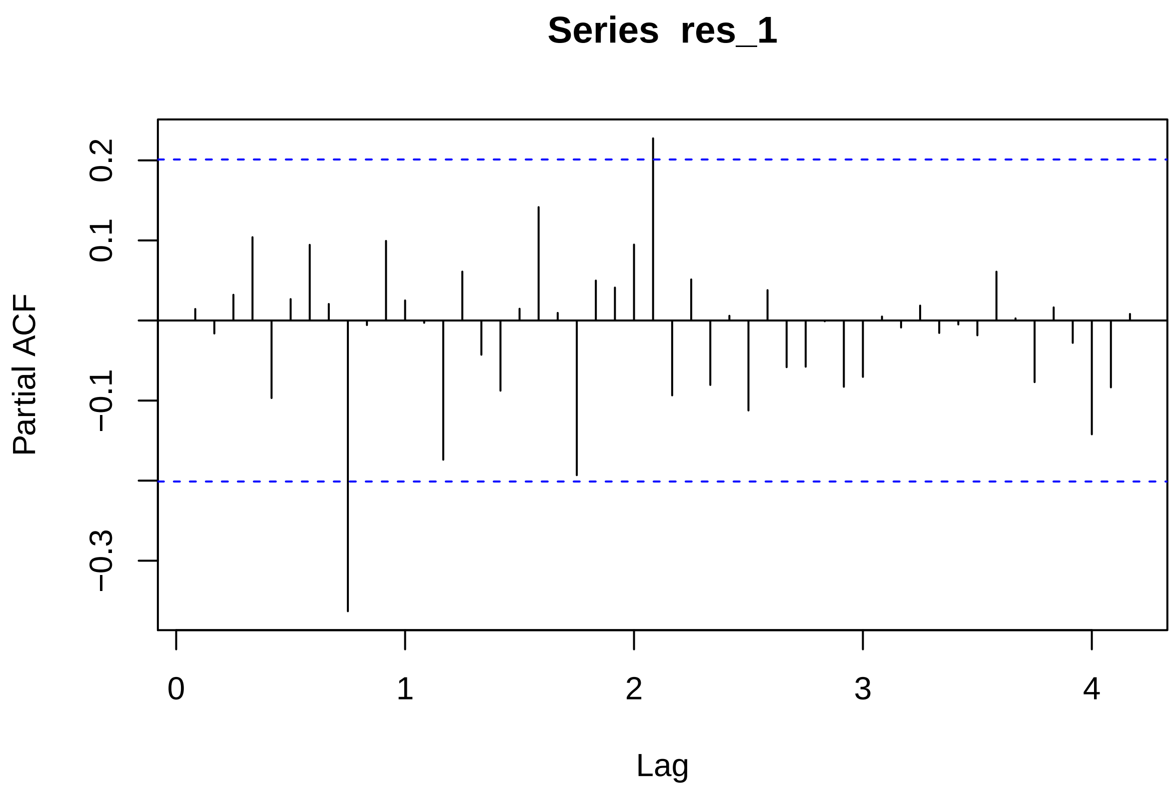
```r
res_1=model_1$residuals
plot(res_1)
```




```
acf(res_1,lag.max=50)
```




```
pacf(res_1,lag.max=50)
```




```r
Box.test(res_1,type='Ljung-Box',fitdf=2,lag=20)
```

```
##
##  Box-Ljung test
##
## data:  res_1
## X-squared = 29.152, df = 18, p-value = 0.04655
```

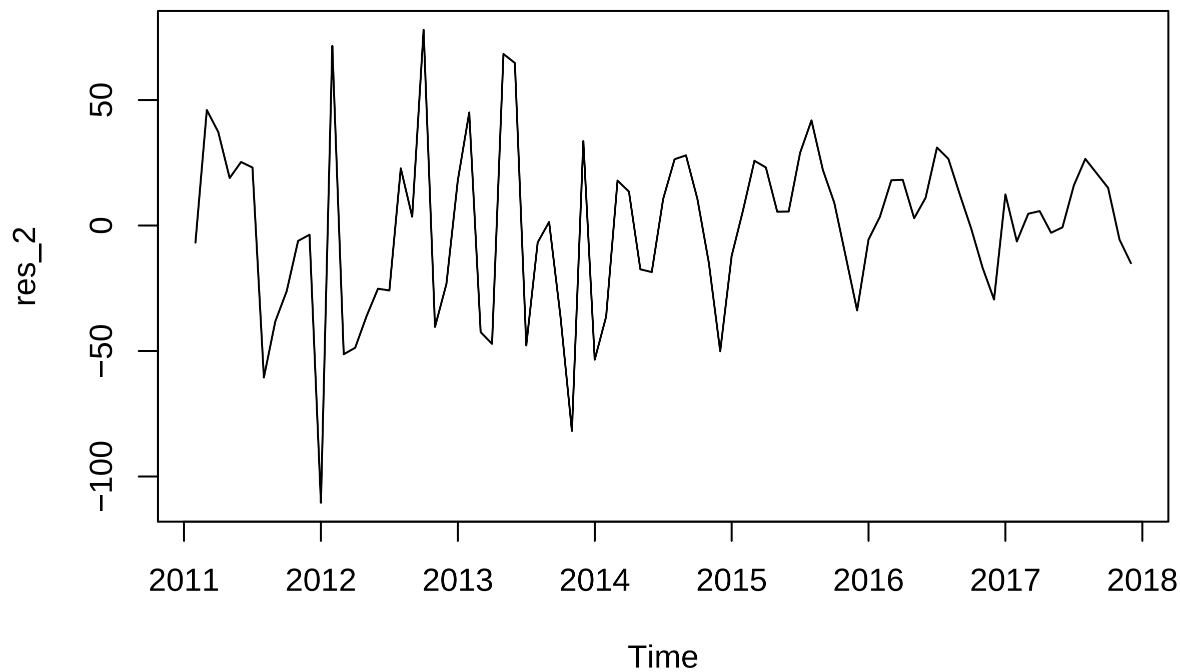
p-value is 0.04655. Then we further difference the training data to remove seasonality

```r
diff2=diff(diff1,differences=1,lag=12)
model_2=auto.arima(diff2)
model_2
```

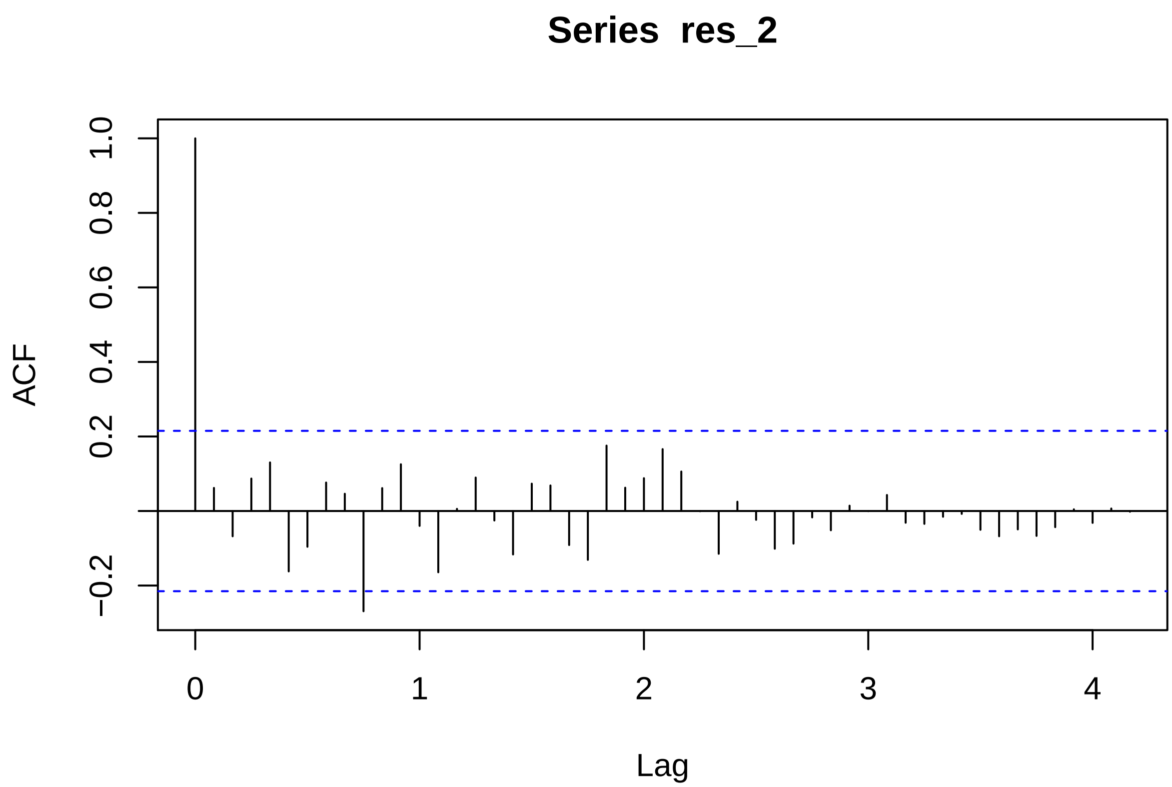
```
## Series: diff2
## ARIMA(0,0,1)(0,0,1)[12] with zero mean
##
## Coefficients:
##           ma1     sma1
##       -0.8782  -0.6877
## s.e.   0.0745   0.1237
##
## sigma^2 estimated as 1148:  log likelihood=-413.88
## AIC=833.76   AICc=834.06   BIC=841.02
```

R suggest that model_2 is ARIMA(0,0,1)(0,0,1)[12]

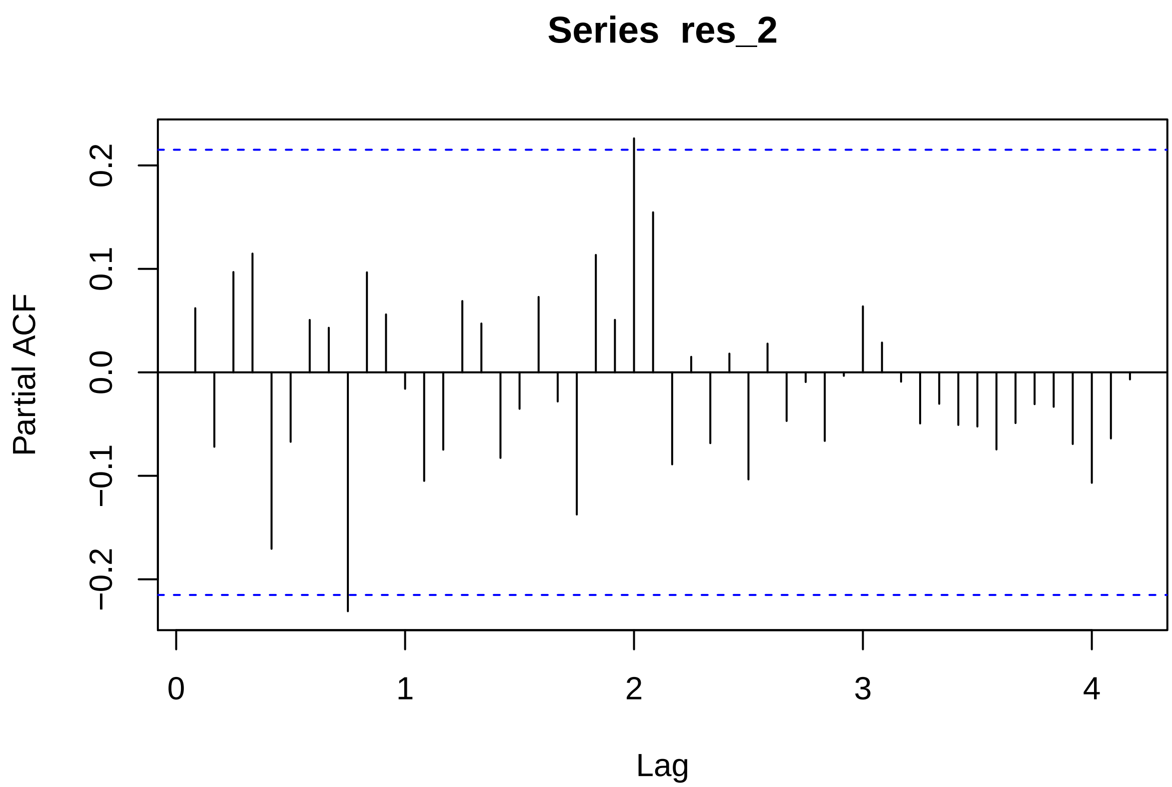
```
res_2=model_2$residuals
plot(res_2)
```




```
acf(res_2,lag.max=50)
```




```
pacf(res_2,lag.max=50)
```



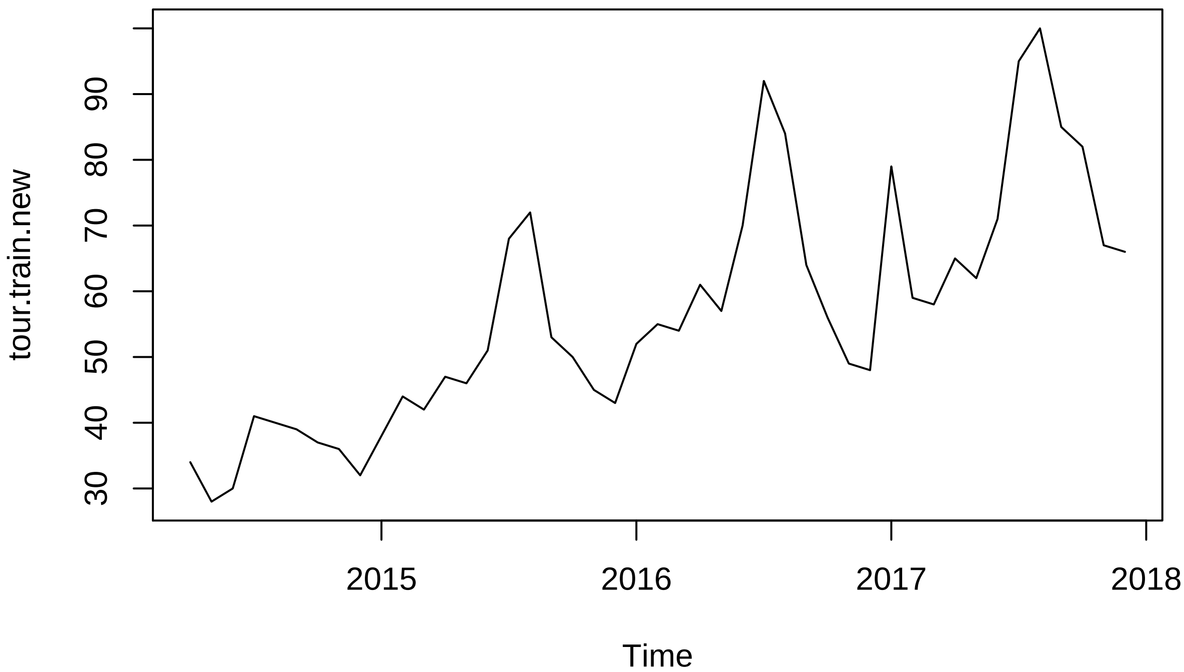

```
Box.test(res_2,type='Ljung-Box',fitdf=5,lag=20)
```

```
##
##  Box-Ljung test
##
## data:  res_2
## X-squared = 22.966, df = 15, p-value = 0.08488
```

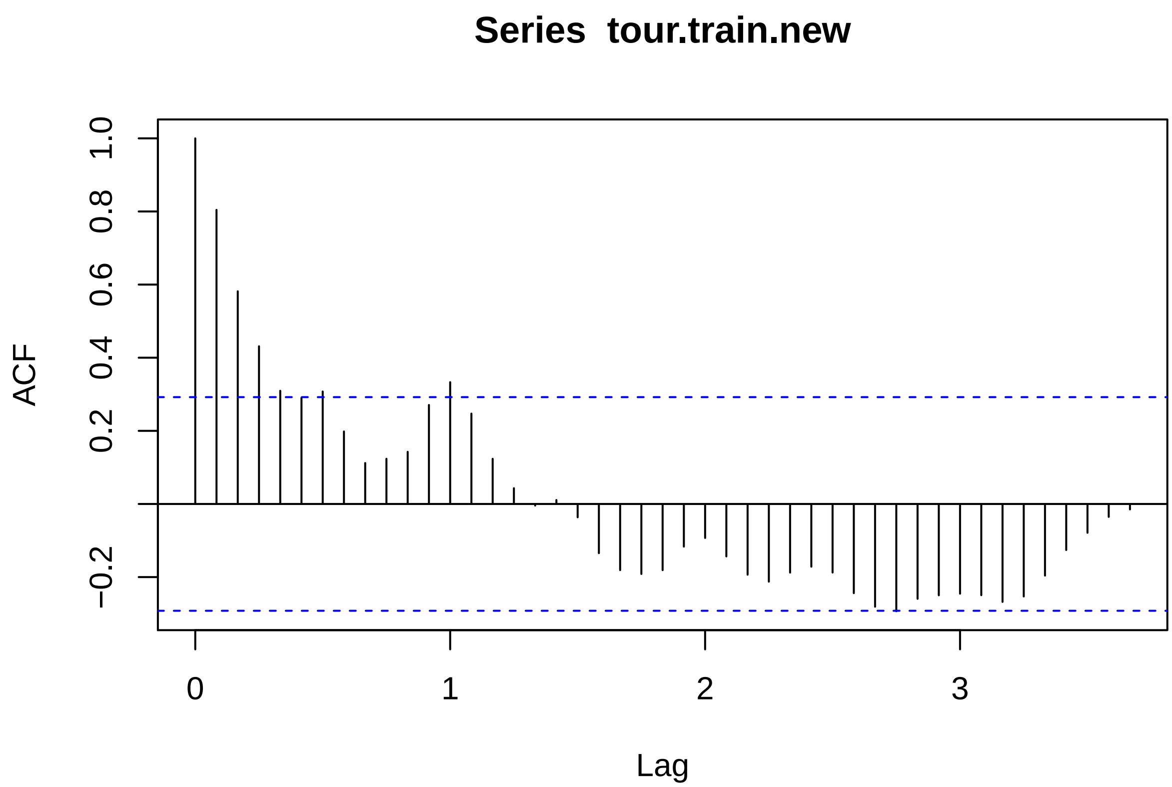
P-value is 0.08488, it is still small, so we step back to the beginning to

Consider further split the data, and only use part of them to train the model

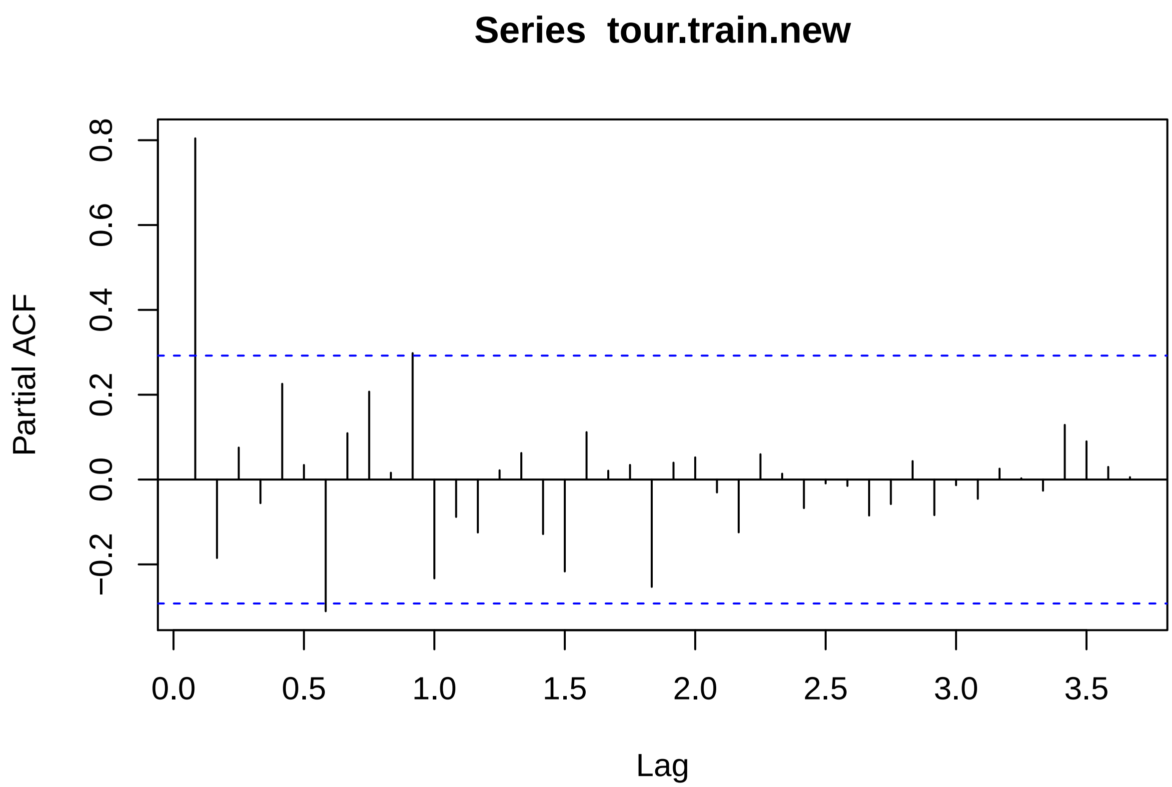
```
tour.train.new=ts(jp$Amounts.of.tourists[52:96],frequency=12,
    start=c(2014,4),end=c(2017,12))
plot(tour.train.new)
```




```
acf(tour.train.new,lag.max=50)
```




```
pacf(tour.train.new,lag.max=50)
```

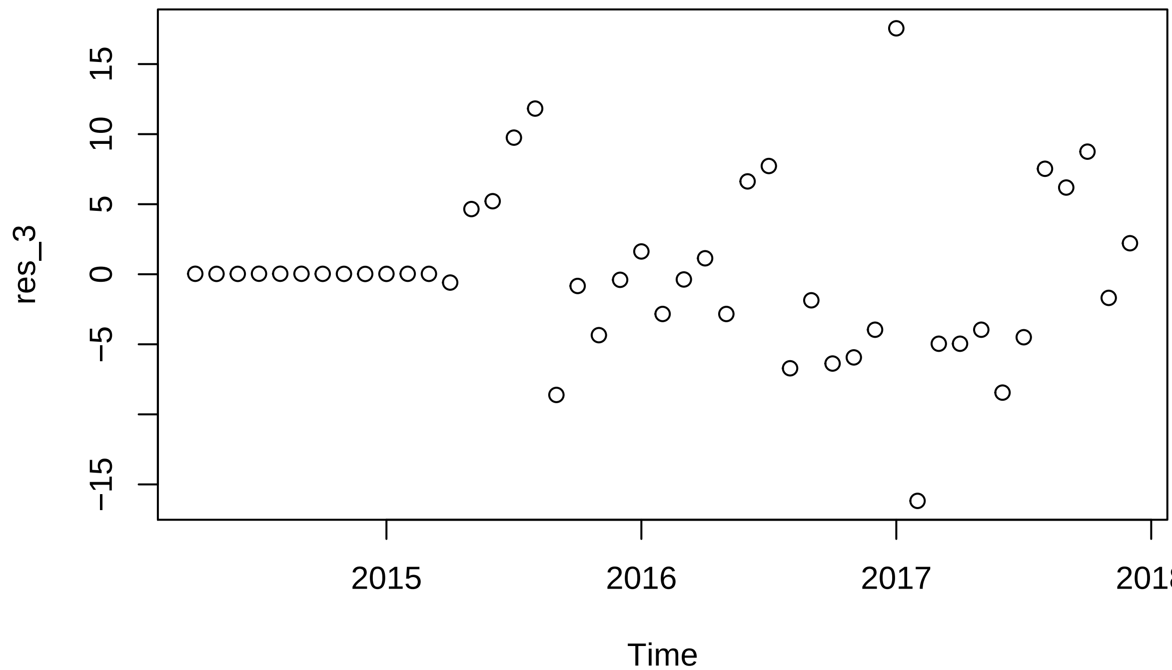



```r
model_3=auto.arima(tour.train.new)
model_3
```

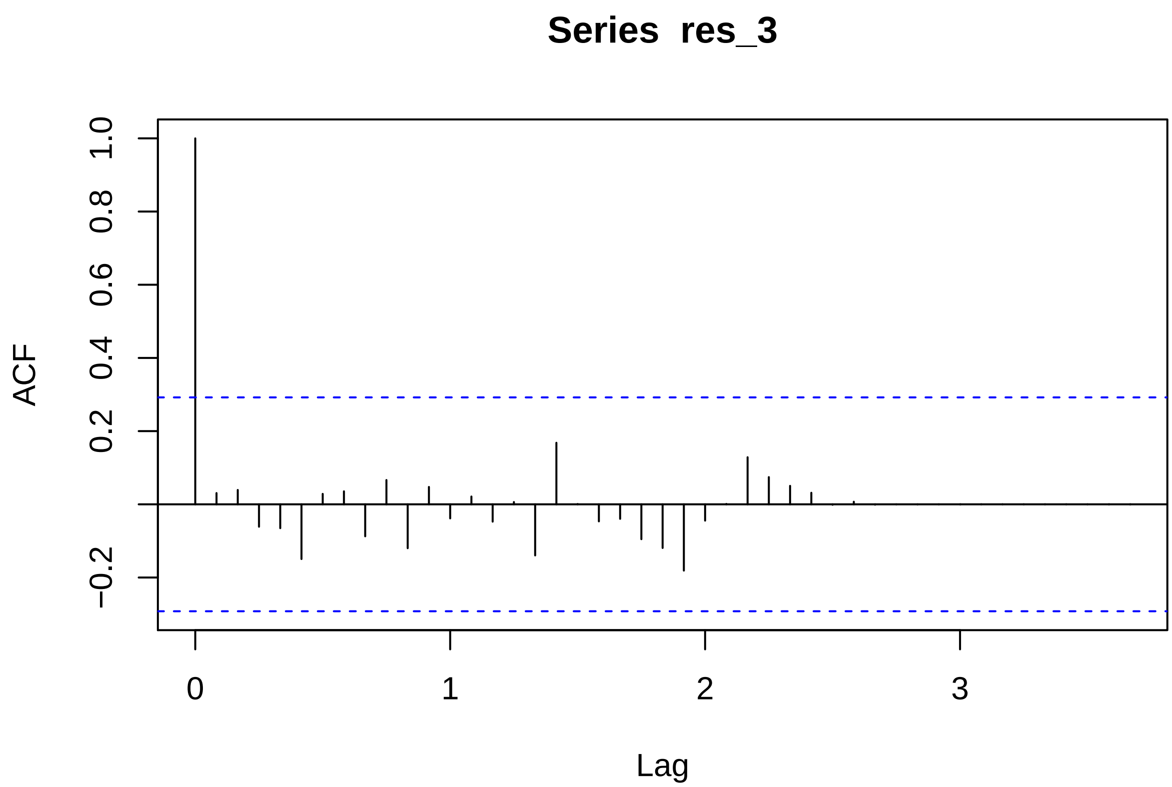
```
## Series: tour.train.new
## ARIMA(1,0,0)(0,1,0)[12] with drift
##
## Coefficients:
##          ar1   drift
##       0.4876  1.1400
## s.e.  0.1479  0.1886
##
## sigma^2 estimated as 49.91:  log likelihood=-110.45
## AIC=226.9   AICc=227.73   BIC=231.39
```

R suggests model_3 is ARIMA(1,0,0)(0,1,0)[12]

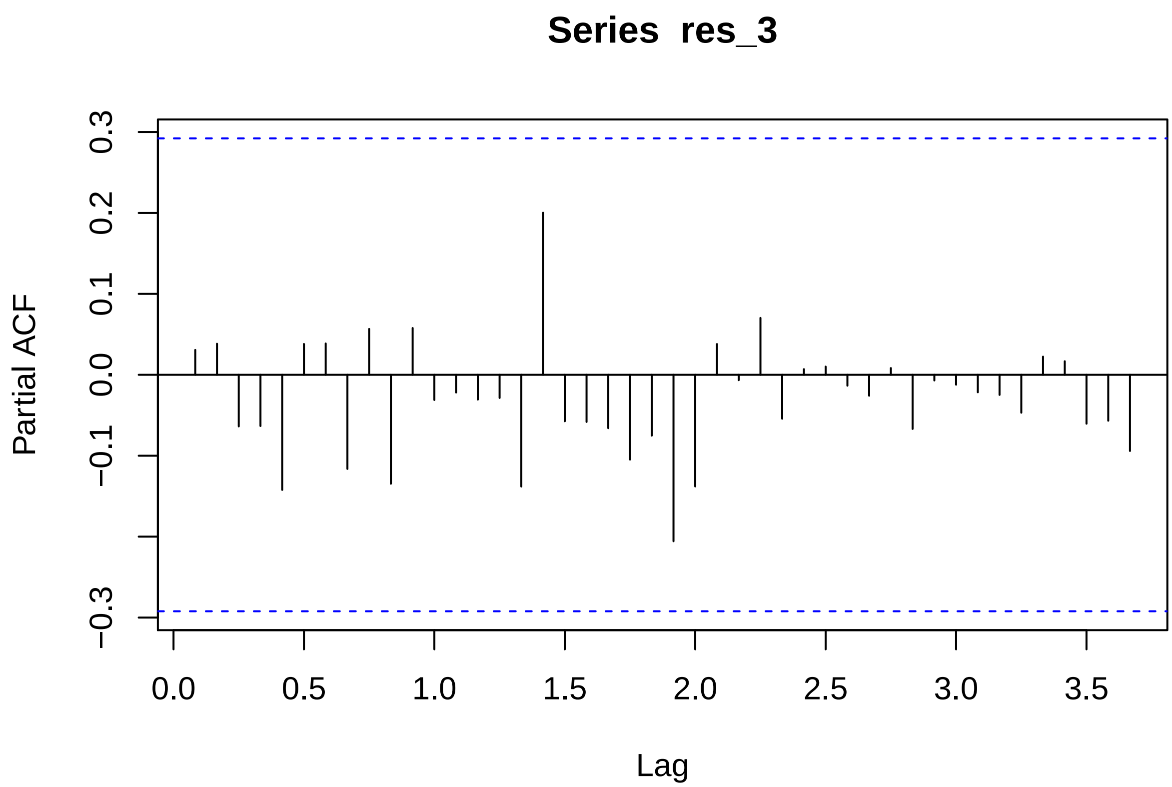
```r
res_3=model_3$residuals
plot(res_3,type='p')
```




```
acf(res_3,lag.max=50)
```




```
pacf(res_3,lag.max=50)
```



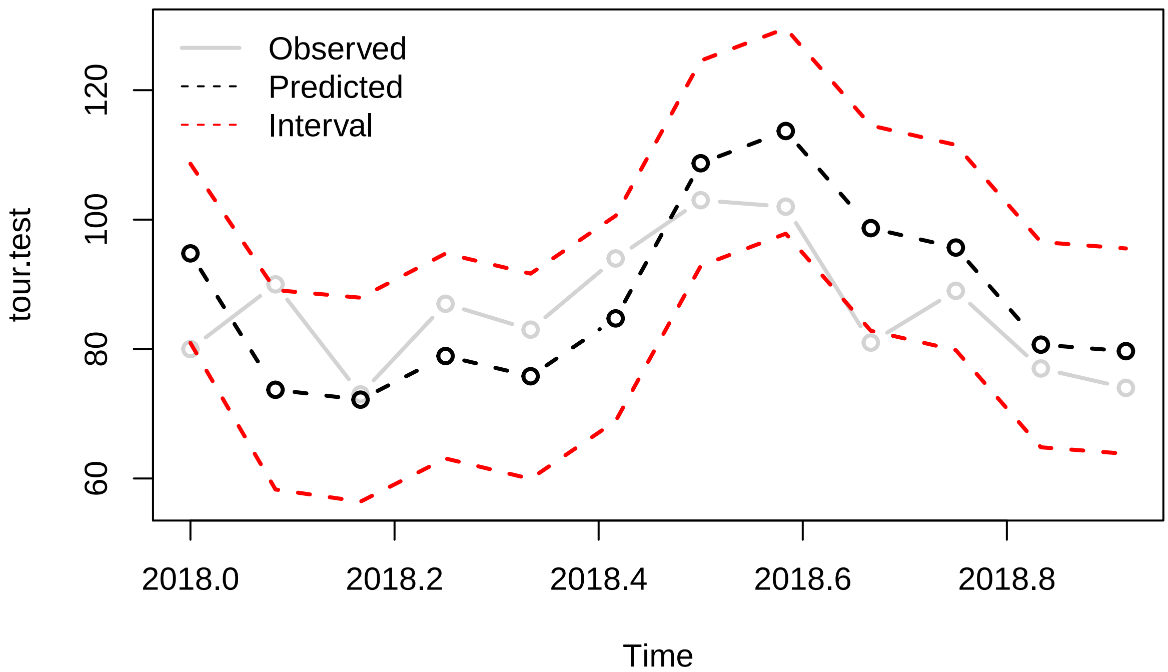

```r
Box.test(res_3,type='Ljung-Box',fitdf=1,lag=20)
```

```
##
##  Box-Ljung test
##
## data:  res_3
## X-squared = 7.6842, df = 19, p-value = 0.9896
```

p-value is 0.9896, model_3 is accepted

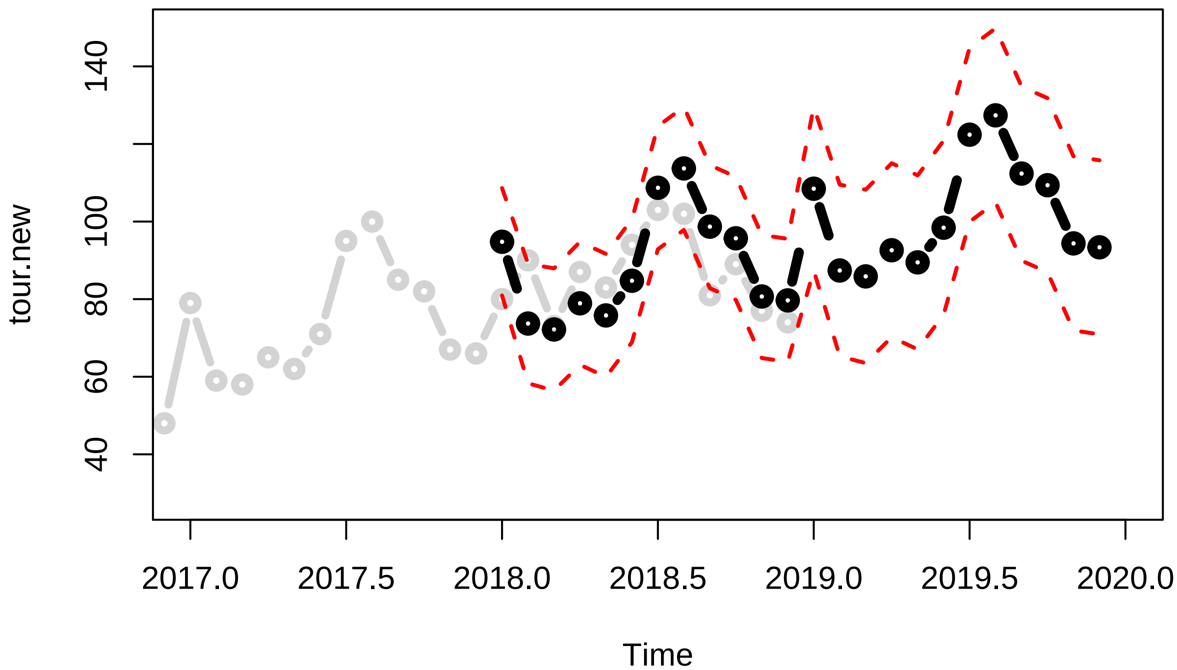
Then we use testing set to check its accuracy

```r
pred18=forecast(model_3,h=12,level=95)
PRED=pred18$mean
LB=pred18$lower
UB=pred18$upper
miny=min(tour.test,PRED,LB,UB)
maxy=max(tour.test,PRED,LB,UB)
plot(tour.test,col='lightgray',type='b',lwd=2,ylim=c(miny,maxy))
lines(PRED,type='b',lty=2,lwd=2)
lines(LB,lty=2,lwd=2,col='red')
lines(UB,lty=2,lwd=2,col='red')
legend('topleft',legend=c('Observed','Predicted','Interval'),lty=c(1,2,2),
    lwd=c(2,1,1),col=c('lightgray','black','red'),bty='n')
```



Further predict tourists in 2019

```
tour.new=ts(jp$Amounts.of.tourists[52:108],frequency=12,start=c(2014,4),end=c(2018,12))
pred1819=forecast(model_3, h=24, level=95)
PRED1819=pred1819$mean
LB1819=pred1819$lower
UB1819=pred1819$upper
miny1819=min(tour.new, PRED1819,LB1819,UB1819)
maxy1819=max(tour.new, PRED1819,LB1819,UB1819)
plot(tour.new,col='lightgray',type='b',lwd=4,xlim=c(2017,2020),ylim=c(miny1819,maxy1819))
lines(PRED1819,lty=2,lwd=5,type='b')
lines(LB1819,lty=2,lwd=2,col='red')
lines(UB1819,lty=2,lwd=2,col='red')
```



Predicted tourists number in 2019

```
pred1819
```

```
##          Point Forecast     Lo 95     Hi 95
## Jan 2018       94.78682  80.93975 108.63390
## Feb 2018       73.70762  58.30198  89.11326
## Mar 2018       72.18137  56.42781  87.93493
## Apr 2018       78.92476  63.08960  94.75992
## May 2018       75.79963  59.94512  91.65413
## Jun 2018       84.73861  68.87951 100.59771
## Jul 2018      108.70886  92.84866 124.56905
## Aug 2018      113.69435  97.83390 129.55480
## Sep 2018       98.68727  82.82676 114.54779
## Oct 2018       95.68382  79.82329 111.54436
## Nov 2018       80.68214  64.82161  96.54268
## Dec 2018       79.68132  63.82079  95.54186
## Jan 2019      108.46775  87.41145 129.52404
## Feb 2019       87.38835  65.27554 109.50115
## Mar 2019       85.86200  63.50533 108.21868
## Apr 2019       92.60535  70.19107 115.01962
## May 2019       89.48019  67.05225 111.90814
## Jun 2019       98.41916  75.98797 120.85036
## Jul 2019      122.38940  99.95744 144.82137
## Aug 2019      127.37489 104.94274 149.80705
```

```
## Sep 2019       112.36782  89.93562 134.80001
## Oct 2019       109.36437  86.93216 131.79657
## Nov 2019        94.36268  71.93047 116.79489
## Dec 2019        93.36186  70.92965 115.79407
```