153project

```r
library(astsa)
library(TSA)
```

```
## Loading required package: leaps

## Loading required package: locfit

## locfit 1.5-9.1    2013-03-22

## Loading required package: mgcv

## Loading required package: nlme

## This is mgcv 1.8-22. For overview type 'help("mgcv-package")'.

## Loading required package: tseries

##
## Attaching package: 'TSA'

## The following objects are masked from 'package:stats':
##
##     acf, arima

## The following object is masked from 'package:utils':
##
##     tar
```

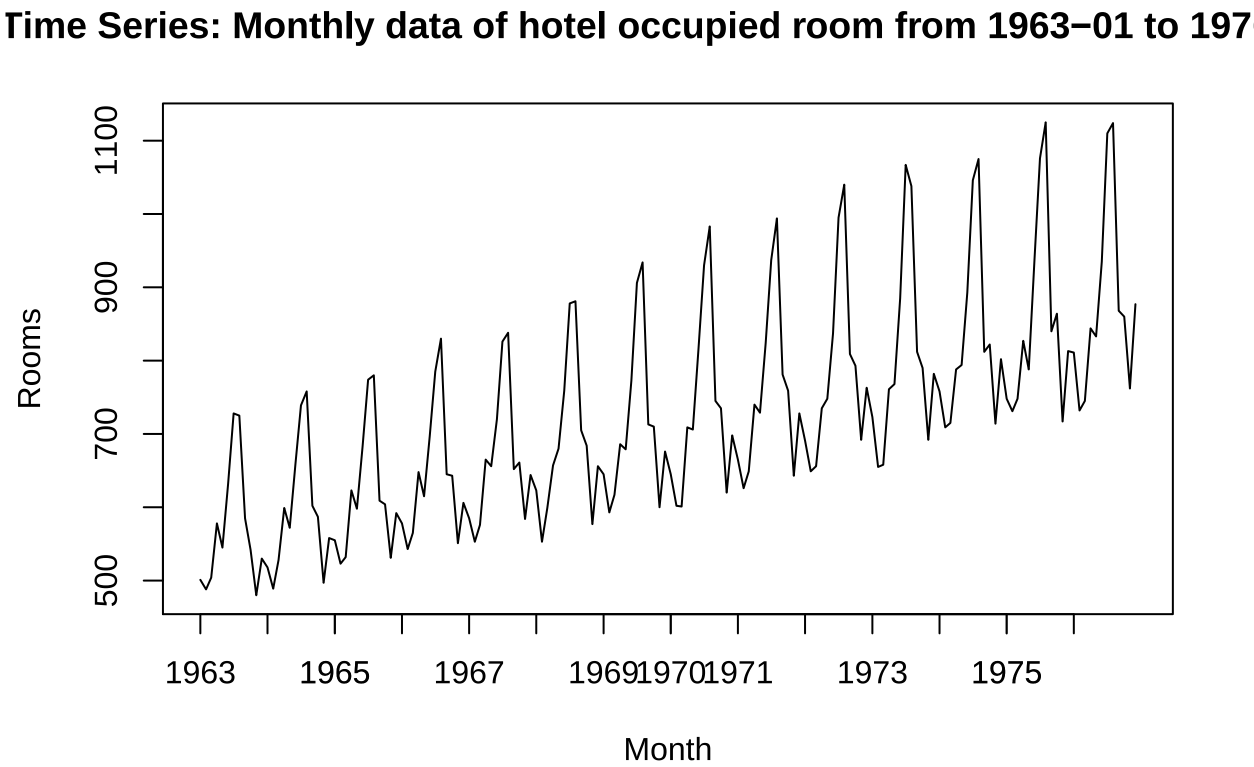
```r
library(tsoutliers)
library(randtests)
```

```
##
## Attaching package: 'randtests'

## The following object is masked from 'package:tseries':
##
##     runs.test
```

2.1 Set up and EDA

```r
setwd("/Users/furonghuang/Documents/Study materials/Statistics/Time Series/Project")
hotel.raw = read.csv("monthly-hotel-occupied-room-av-6.csv")
colnames(hotel.raw)=c("Month", "Rooms")
hotel = hotel.raw[-169,]
hotel.ts = ts(hotel[,2], start=c(1963, 1), end = c(1976,12), frequency = 12)
hotel.train = ts(hotel[,2], start=c(1963, 1), end = c(1974,12), frequency = 12) # select 1963-1974 as t
hotel.train.df = hotel[1:144, ]
hotel.test = ts(hotel[145:168, 2], start=c(1975, 1), end = c(1976,12), frequency = 12) # leave last 2 y
hotel.test.df = hotel[145:168, ]
hotel$Month = as.Date(paste(as.character(hotel$Month), "-01", sep=""),
                      format = "%Y-%m-%d")
# in order to show yearly tick on the axis, have to use data.frame for plotting
plot(Rooms~Month, data= hotel, type="l",
     main="Time Series: Monthly data of hotel occupied room from 1963-01 to 1976-12")
axis.Date(side=1, at=seq.Date(min(hotel$Month),max(hotel$Month), by="1 year"))
```




```
# add tick every year
```

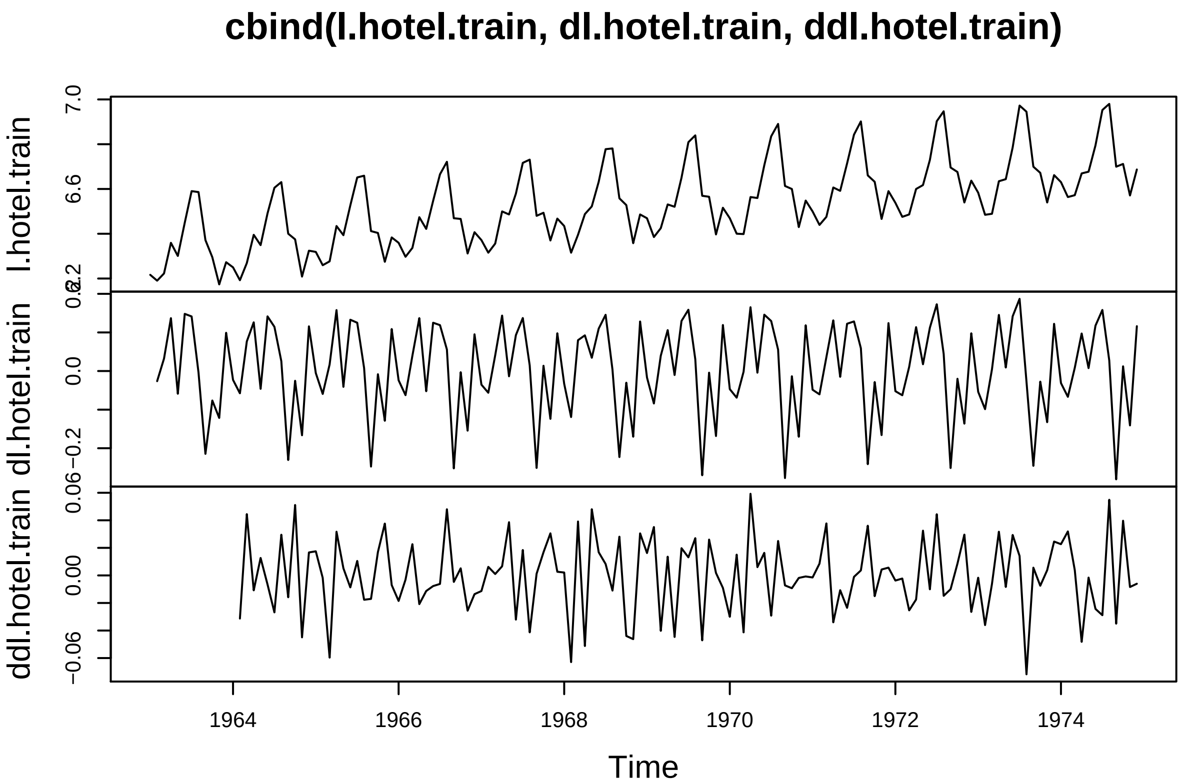
Test for outliner

```
tso(hotel.train, types = c("TC", "AO", "LS", "IO", "SLS"))
```

```
## Series: hotel.train
## Regression with ARIMA(2,0,0)(1,1,0)[12] errors
##
## Coefficients:
##          ar1     ar2     sar1    SLS80   SLS115
##       0.5685  0.3598  -0.4857  41.6283  48.3424
## s.e.  0.0811  0.0815   0.0822  11.3906  11.4903
##
## sigma^2 estimated as 240.6:  log likelihood=-548.92
## AIC=1109.84   AICc=1110.51   BIC=1127.14
##
## Outliers:
##   type ind    time coefhat tstat
## 1  SLS  80 1969:08   41.63 3.655
## 2  SLS 115 1972:07   48.34 4.207
```

Chasing stationarity

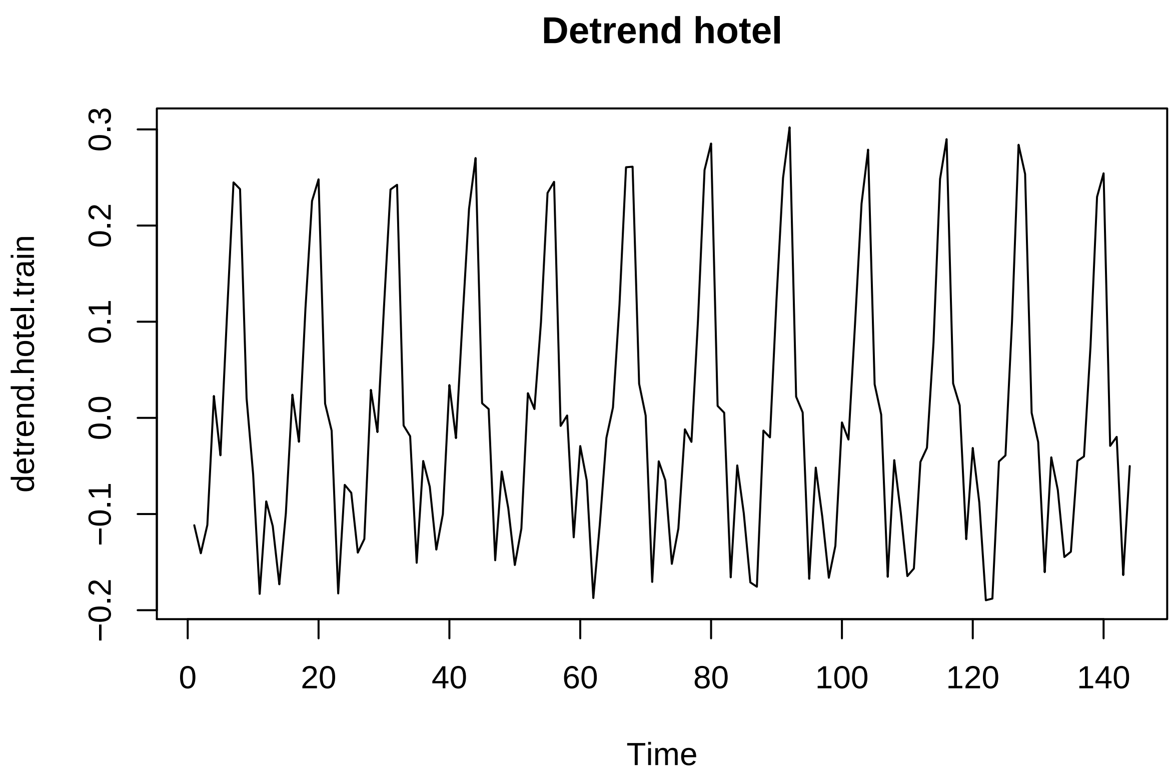
```
l.hotel = log(hotel[,2])
l.hotel.train = log(hotel.train)
dl.hotel.train = diff(l.hotel.train)
ddl.hotel.train = diff(dl.hotel.train,12)
plot.ts(cbind(l.hotel.train, dl.hotel.train, ddl.hotel.train))
```



2.2 Spectral analysis

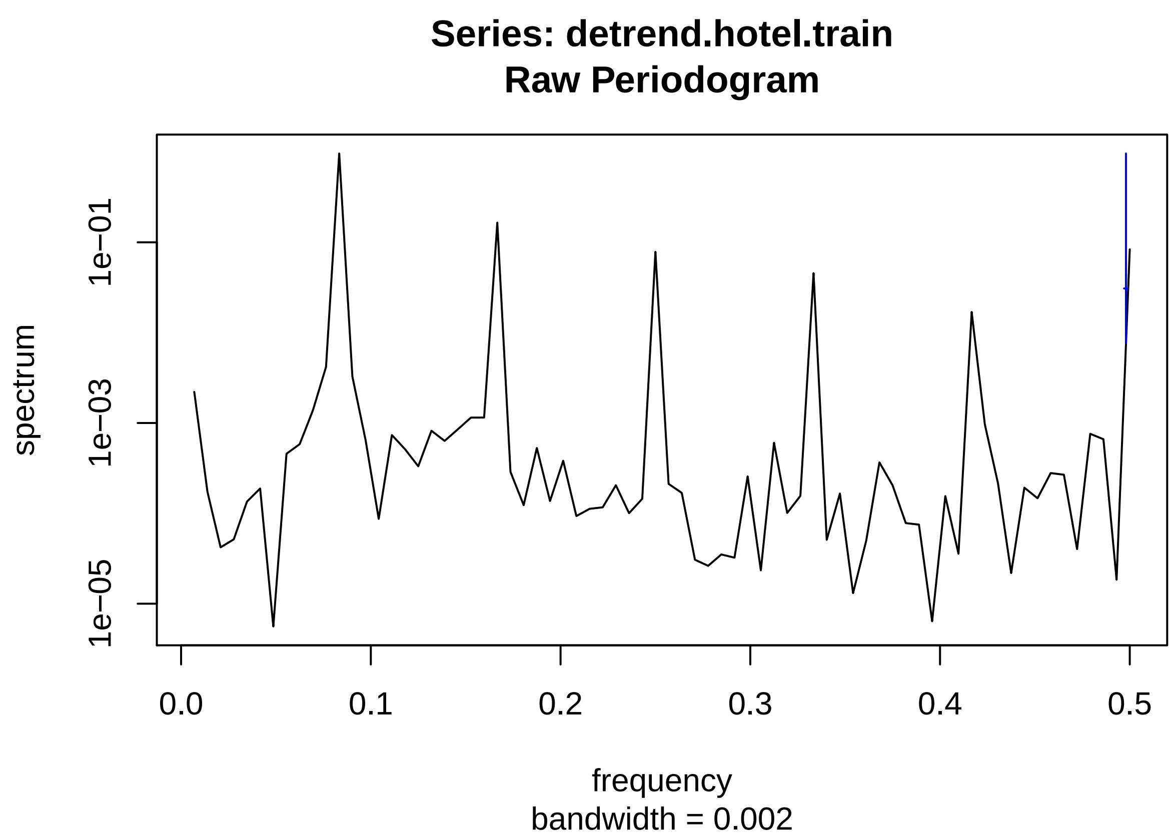
detrend by linear regression, residual plot

```
t = 1:length(l.hotel.train)
fit = lm(l.hotel.train ~ t)
detrend.hotel.train = fit$residuals
# to use spec.pgram() for periodogram, we need stationary, so use the detrend data. However, for spec.p
plot.ts(detrend.hotel.train, main="Detrend hotel")
```



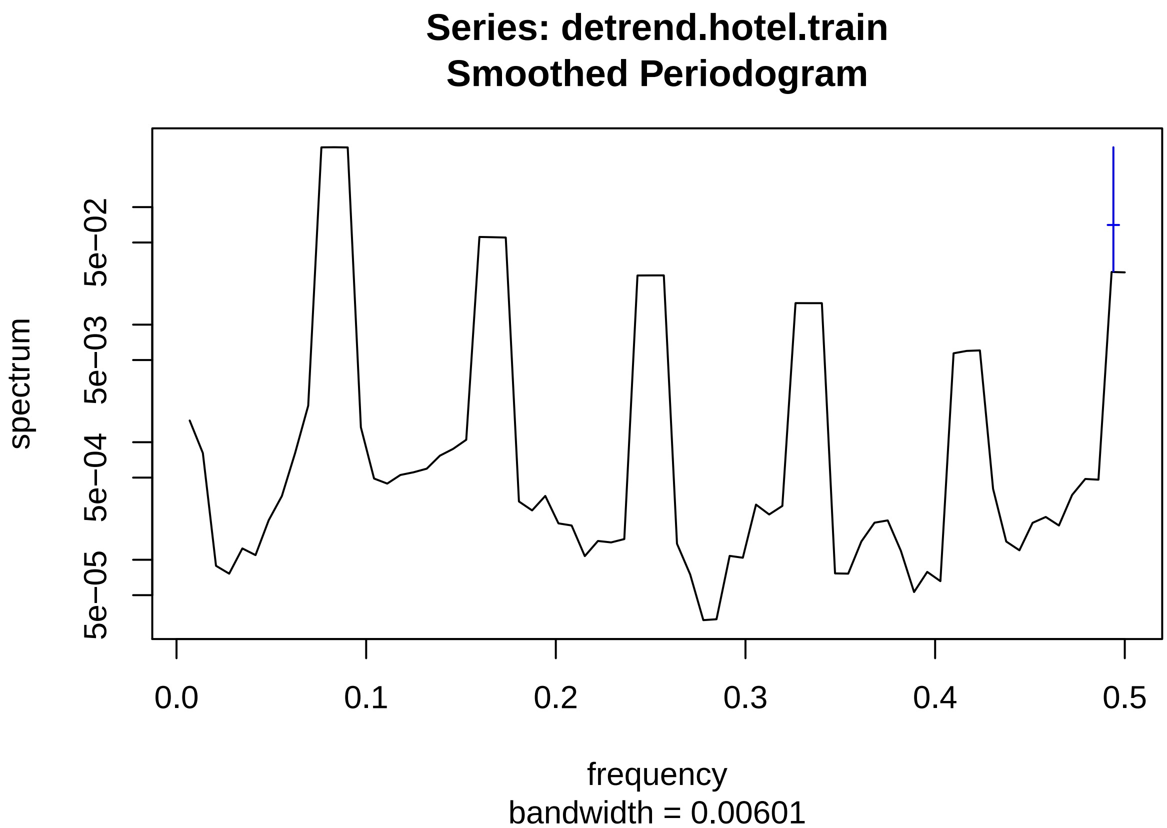
Periodogram without smoothing and tapering

```
spec.pgram(detrend.hotel.train, taper = 0)
```



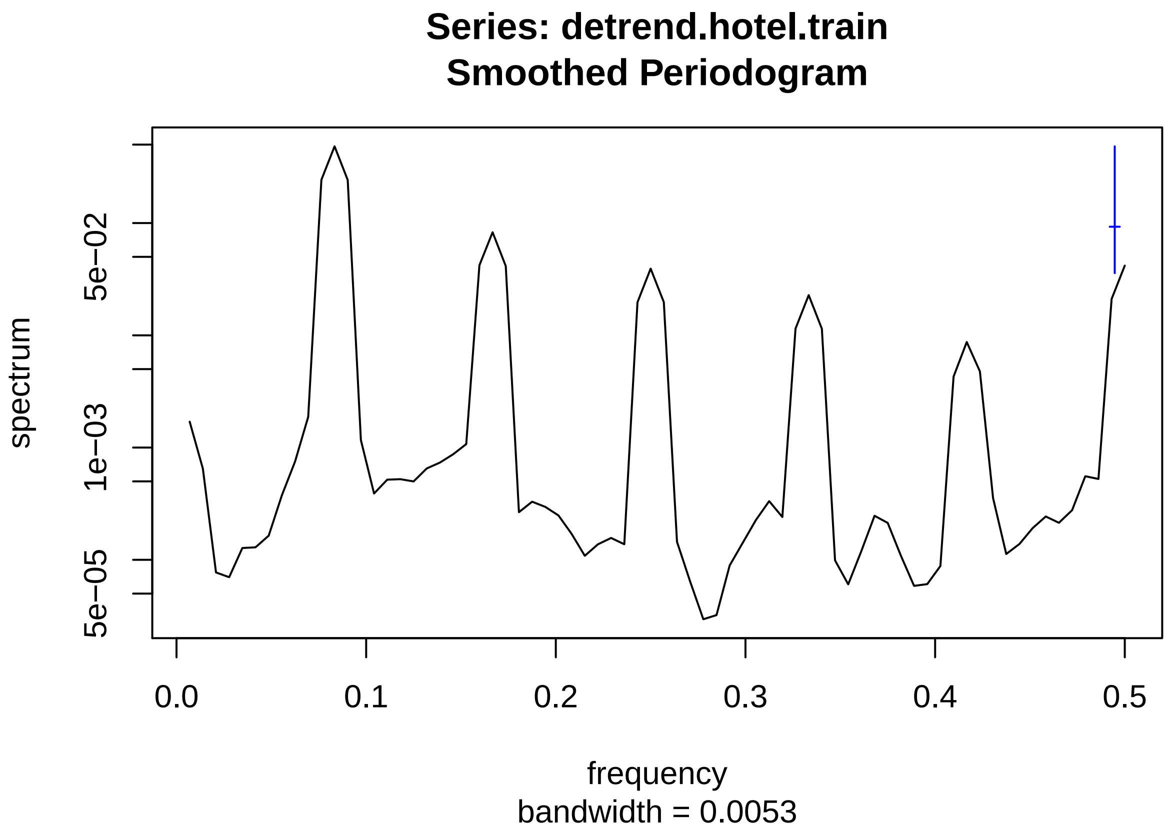
Use daniell kernel to smooth the periodogram

```
spec.pgram(detrend.hotel.train, kernel("daniell", 1), taper=0)
```



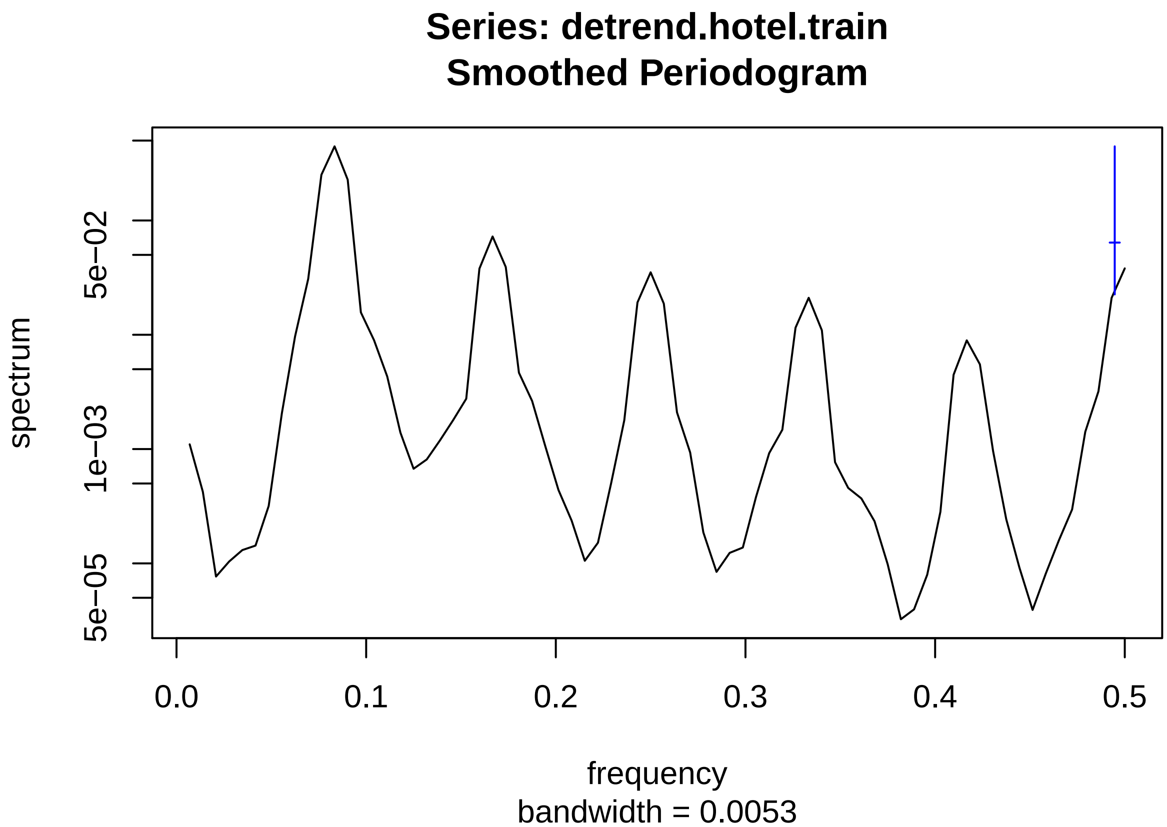
To avoid the flat region at the peaks, use modified.daniell kernel

```
spec.pgram(detrend.hotel.train, kernel("modified.daniell", 1), taper=0)
```



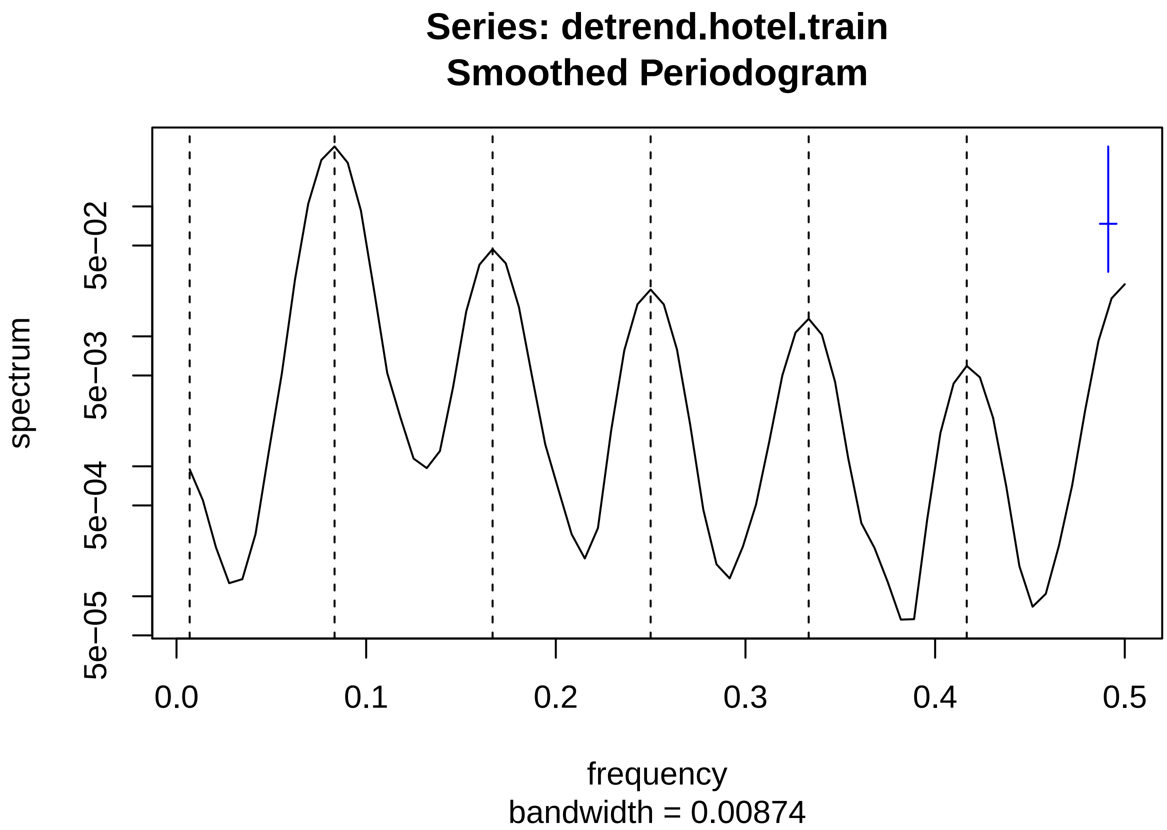
To reduce the side lobes, use tapering

```
spec.pgram(detrend.hotel.train, kernel("modified.daniell", 1), taper=0.2)
```



Find the key frequencies

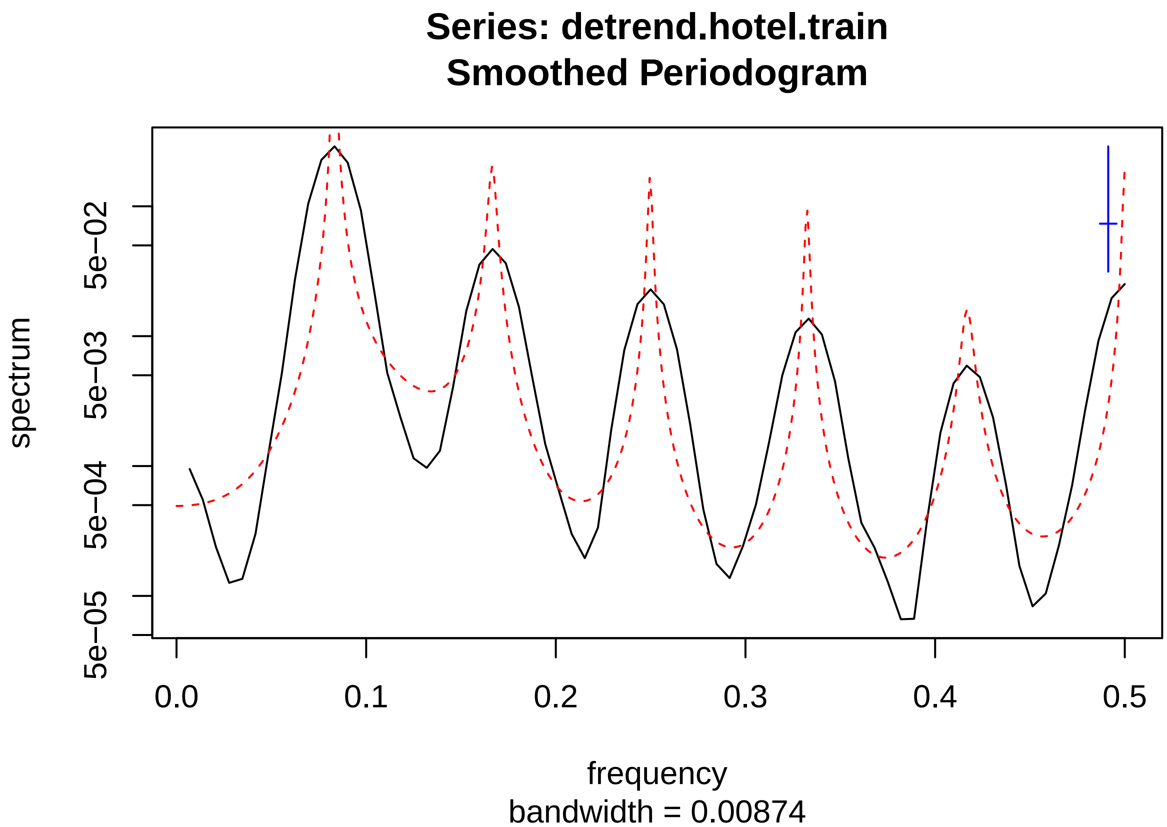
```
pgram = spec.pgram(detrend.hotel.train, kernel("modified.daniell", c(1, 1, 1)), taper=0.2)
key_freq_ind = c(1, which(diff(sign(diff(pgram$spec)))==-2) + 1)
key_freq = pgram$freq[key_freq_ind]
abline(v=key_freq, lty=2)
```




```
# notice that the periodogram should range from [0, 1/2], sometimes if you use the log data(which retai
```

Parametric way to find the periodogram and comparision to Nonparametric way.

```
spec.pgram(detrend.hotel.train, kernel("modified.daniell", c(1, 1, 1)), taper=0.2)
pgram_ar = spec.ar(detrend.hotel.train, plot=F) # plot the parametric spectral estimation as red, it ha
lines(pgram_ar$freq, pgram_ar$spec, lty=2, col="red")
```

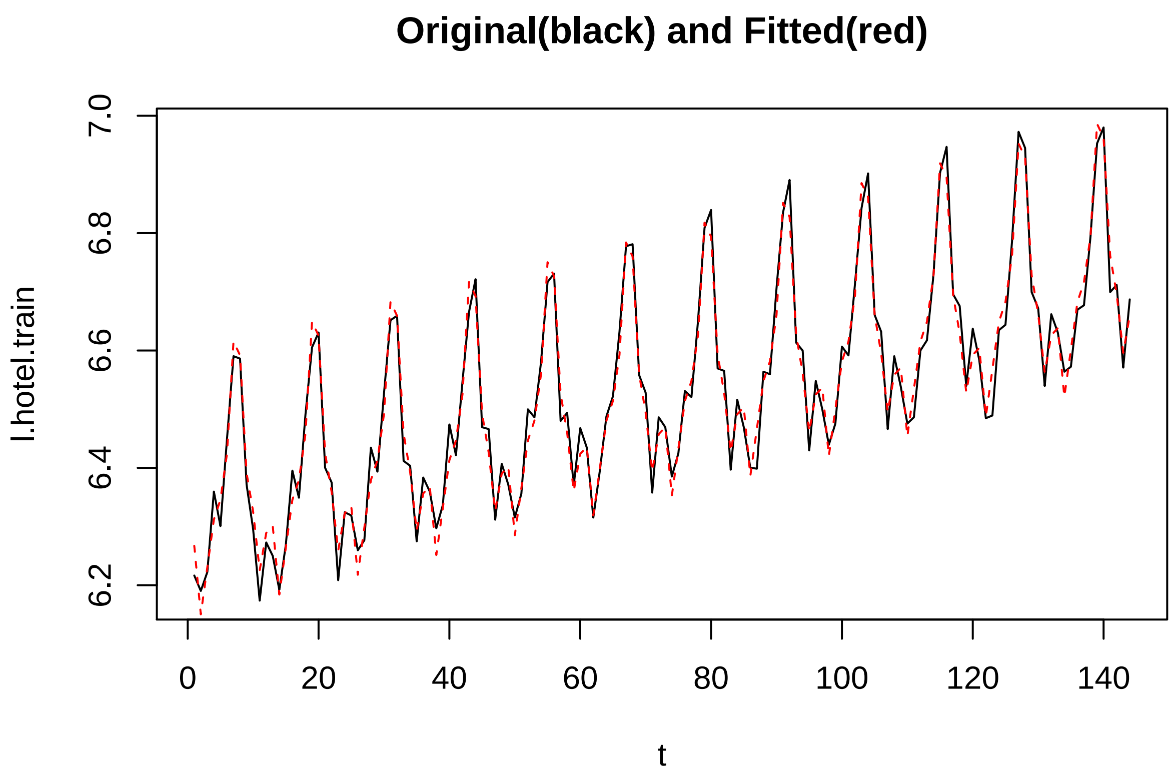


Parametric and Nonparametric yeild the same key frequencies

Check the model

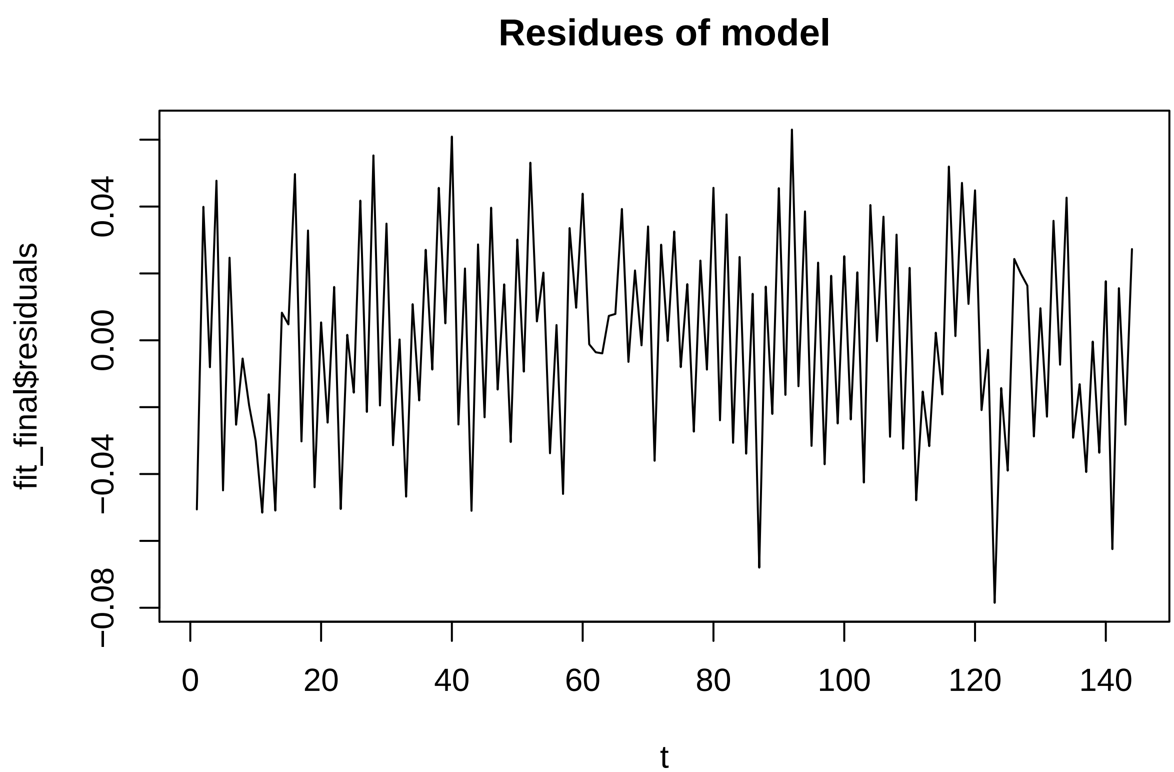
```
top_freq = key_freq[order(pgram$spec[key_freq_ind], decreasing = T)][1:5]
periodic_terms = do.call(cbind, lapply(top_freq, function(freq) {
  cbind(cos(2 * pi * freq * t), sin(2 * pi * freq * t))
})) # no need to add columns of 1 and t because the fn lm() will accomodate the trend
df = data.frame(l.hotel.train, t, periodic_terms) # change: fit the original data instead of the log da
fit_final = lm(l.hotel.train ~ ., df) # first 2 cols are for linear trend, next cols are cos&sin terms,

plot(t, l.hotel.train, type="l", main ="Original(black) and Fitted(red)") # plot the original time seri
lines(t, fit_final$fitted.values, lty=2, col="red")
```



residual of the actual data and the model data

```r
plot(t, fit_final$residuals, type = "l", main="Residues of model")
```

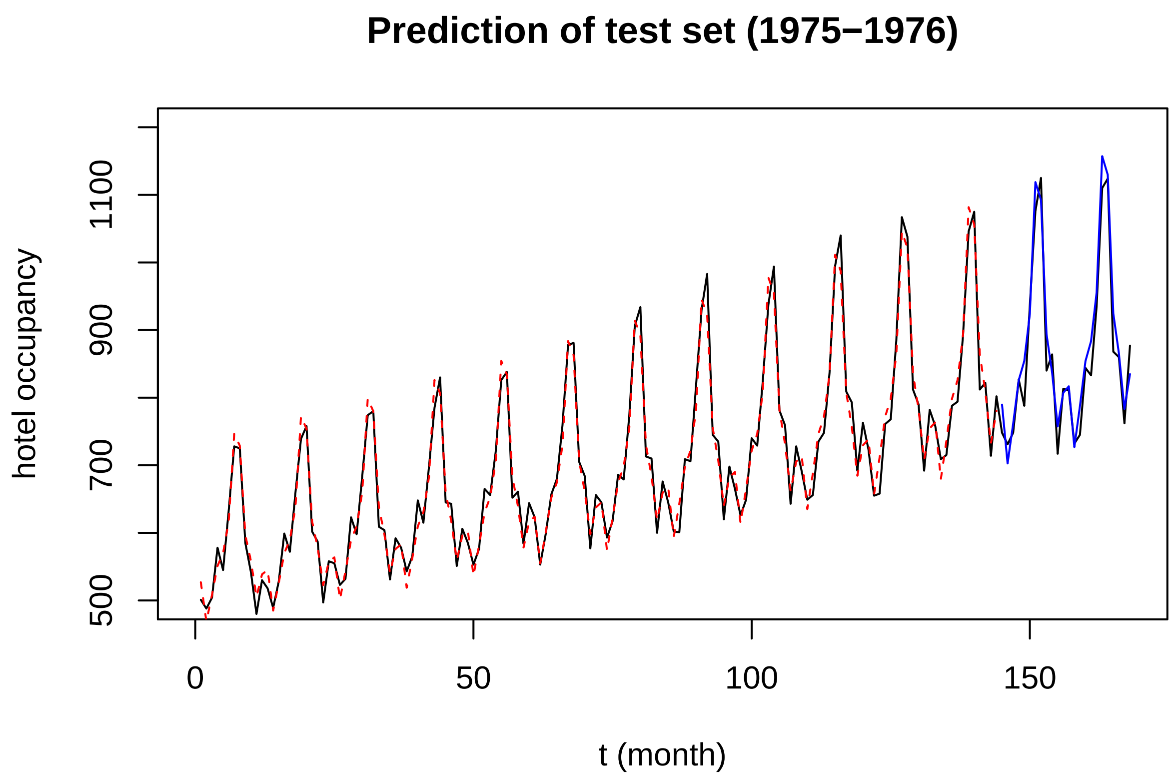


The residuals are nearly equally spread on two sides of y=0 line and have a constant mean.

Predict the test set

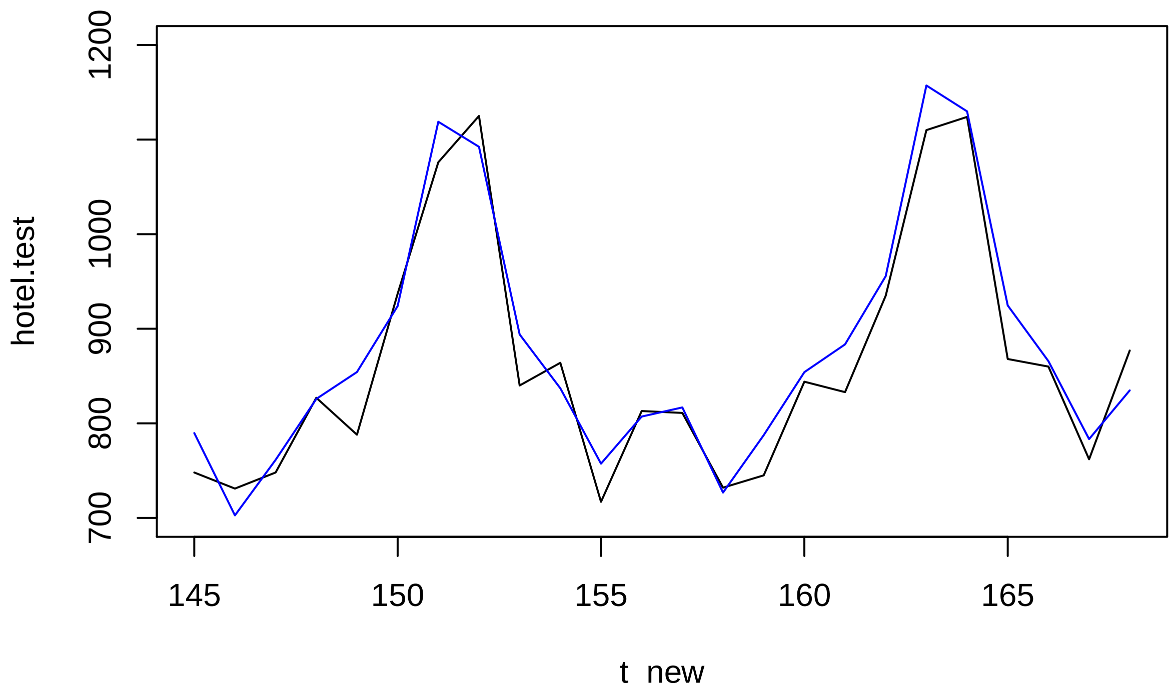
```r
t_new = (tail(t, 1) + 1):(tail(t, 1) + 24)
periodic_terms_new = do.call(cbind, lapply(top_freq, function(freq) {
  cbind(cos(2 * pi * freq * t_new), sin(2 * pi * freq * t_new)) # key freqs are the same for the whole
}))
df_new = data.frame(t_new, periodic_terms_new)
colnames(df_new) = colnames(df)[-1]
hotel.pred.periogram = predict.lm(fit_final, newdata=df_new, interval="prediction", level=.95) # calc p

plot(c(t, t_new), hotel$Rooms, type="l", xlim=c(0, tail(t_new, 1)), ylim=c(500, 1200), xlab = "t (month
lines(t, exp(fit_final$fitted.values), lty=2, col="red")
lines(t_new, exp(hotel.pred.periogram[, "fit"]), col="blue") # blue is prediction for test set
```



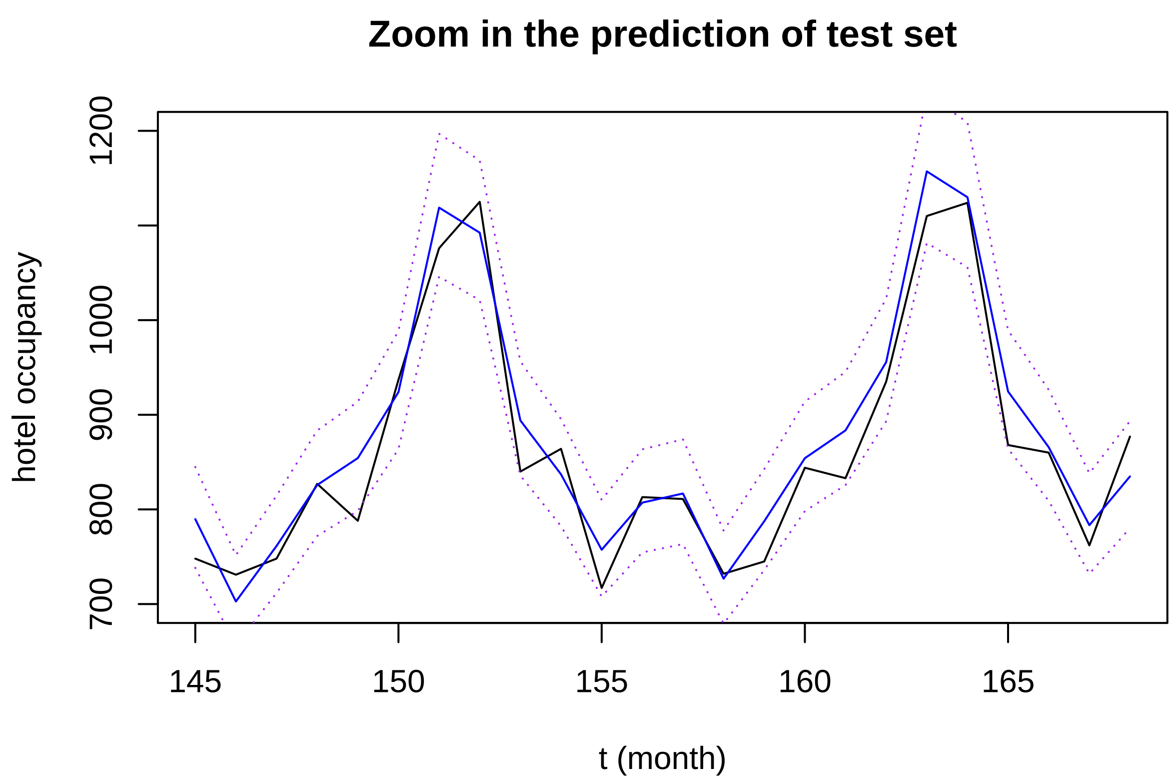
Real test value compared to predicted value

```r
plot(t_new, hotel.test, type = "l", ylim = c(700,1200))
lines(t_new, exp(hotel.pred.periogram[, "fit"]), col="blue") # blue is prediction for test set
```



add CI

```
plot(t_new, hotel.test, type = "l", ylim = c(700,1200), main="Zoom in the prediction of test set", xlab
lines(t_new, exp(hotel.pred.periogram[, "fit"]), col="blue") # blue is prediction for test set
matlines(t_new, exp(hotel.pred.periogram[, 2:3]), col = "purple", lty=3) #purple is CI
```



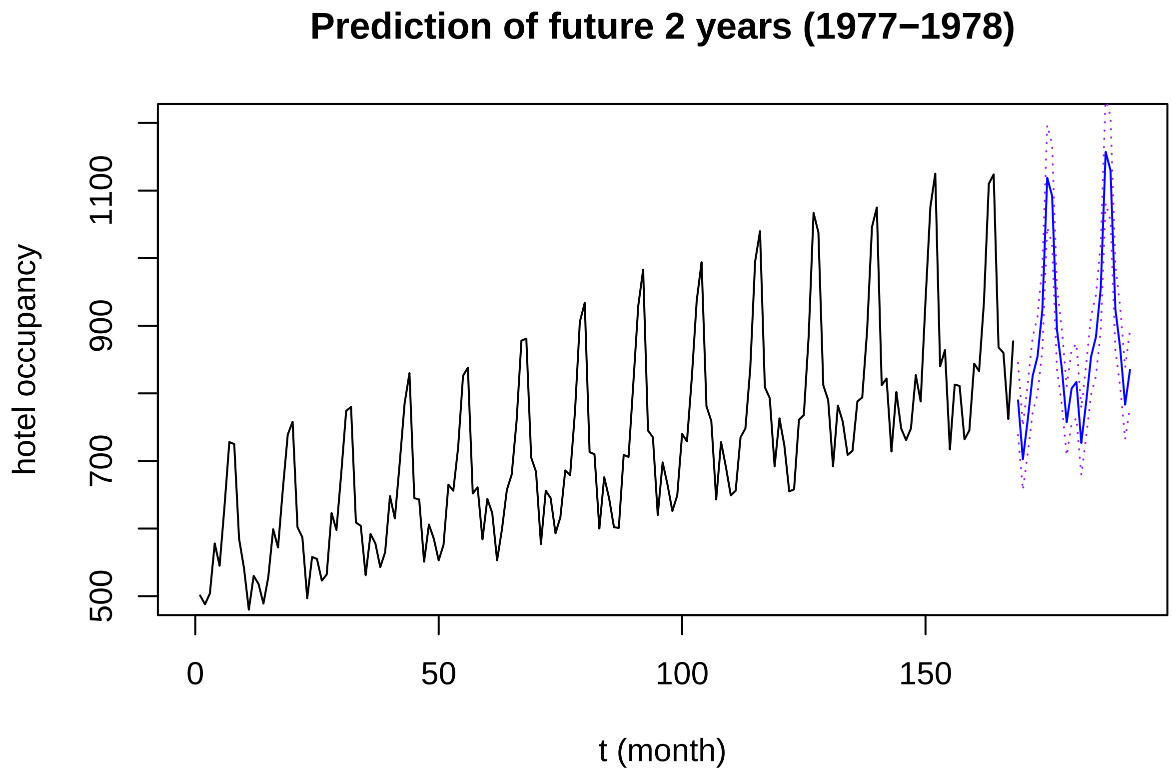
MSE of the test set

```r
mean((exp(hotel.pred.periogram[, "fit"])-hotel.test)^2)
```

```
## [1] 1173.298
```

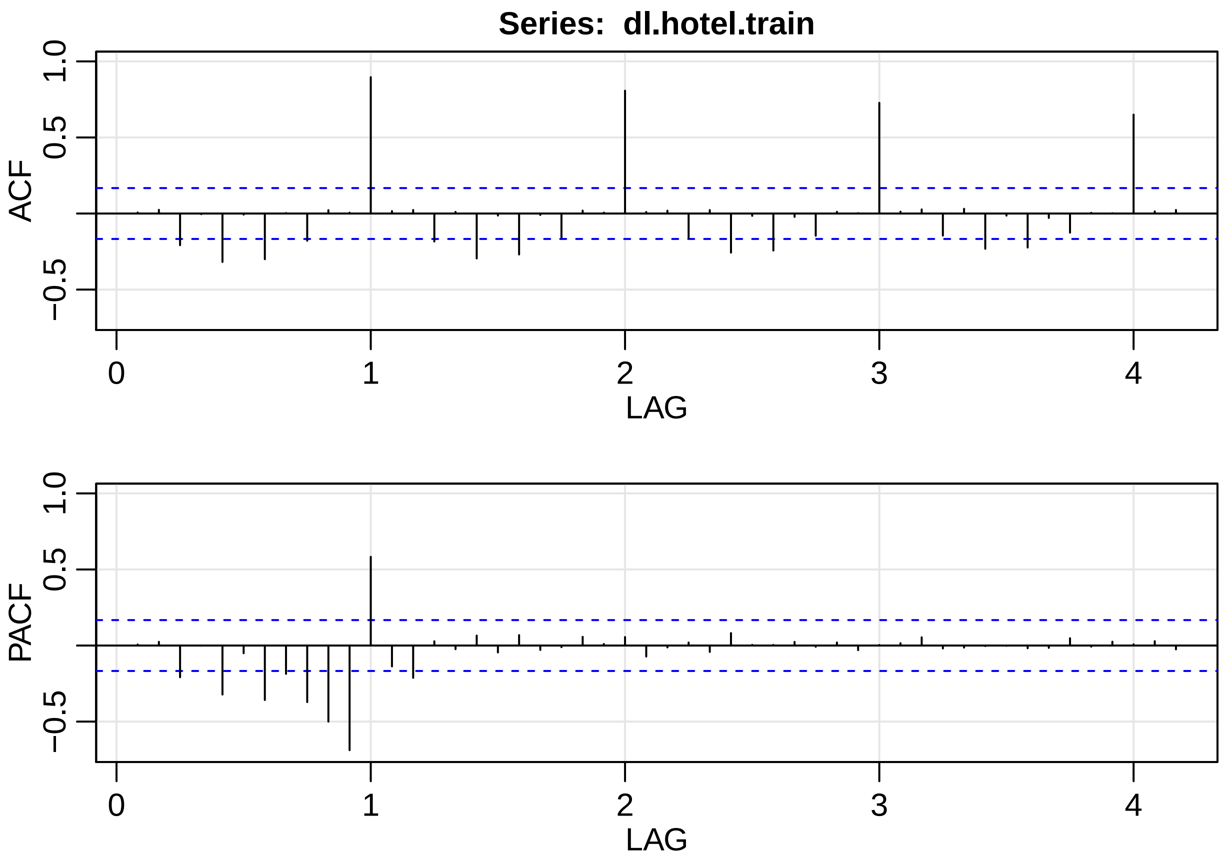
Predict the future 2 years (1977-1978)

```r
t_pred = (tail(t_new, 1) + 1):(tail(t_new, 1) + 24)
plot(c(t, t_new), hotel$Rooms, type = "l", xlim = c(0, tail(t_pred,1)), ylim = c(500,1200), xlab = "t (m
lines(t_pred, exp(hotel.pred.periogram[,"fit"]), col="blue")
matlines(t_pred, exp(hotel.pred.periogram[,2:3]), col='purple', lty = 3)
```



2.3 Fit SARIMA ACF

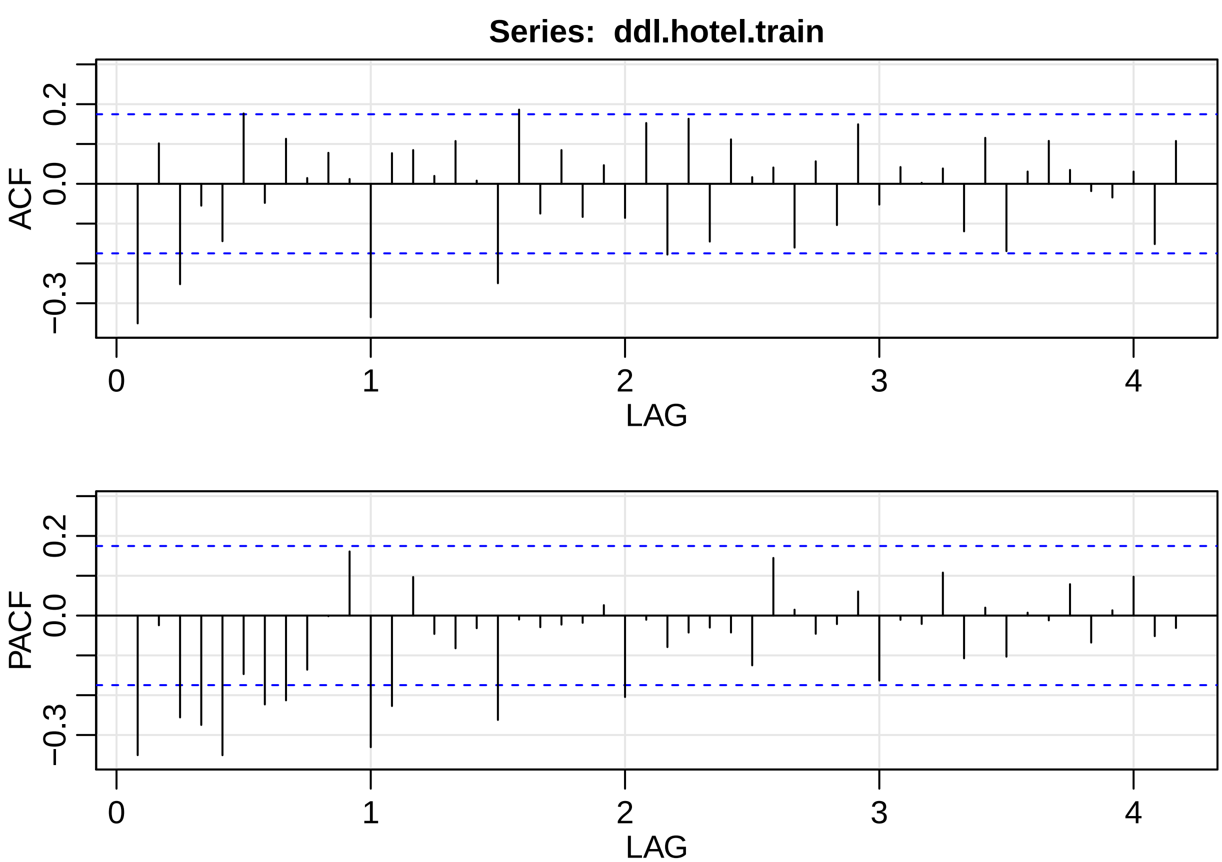
```r
acf2(dl.hotel.train,50) # see a seasonal pattern every 12 months
```




```
##        ACF  PACF
##  [1,]  0.01  0.01
##  [2,]  0.02  0.02
##  [3,] -0.21 -0.21
##  [4,]  0.00  0.00
##  [5,] -0.32 -0.32
##  [6,] -0.01 -0.05
##  [7,] -0.30 -0.36
##  [8,]  0.00 -0.19
##  [9,] -0.18 -0.37
## [10,]  0.02 -0.50
## [11,]  0.01 -0.69
## [12,]  0.90  0.58
## [13,]  0.02 -0.14
## [14,]  0.02 -0.21
## [15,] -0.18  0.03
## [16,]  0.01 -0.02
## [17,] -0.30  0.07
## [18,] -0.01 -0.05
## [19,] -0.27  0.07
## [20,] -0.01 -0.03
## [21,] -0.16 -0.01
## [22,]  0.02  0.06
## [23,]  0.01  0.01
## [24,]  0.81  0.06
## [25,]  0.01 -0.07
## [26,]  0.02 -0.01
```

```
## [27,] -0.16  0.02
## [28,]  0.02 -0.04
## [29,] -0.26  0.08
## [30,] -0.02  0.00
## [31,] -0.24  0.00
## [32,] -0.02  0.02
## [33,] -0.15 -0.01
## [34,]  0.01  0.02
## [35,]  0.00 -0.03
## [36,]  0.73  0.00
## [37,]  0.01  0.02
## [38,]  0.03  0.05
## [39,] -0.15 -0.02
## [40,]  0.03 -0.01
## [41,] -0.23  0.00
## [42,] -0.01  0.00
## [43,] -0.22 -0.02
## [44,] -0.03 -0.02
## [45,] -0.13  0.05
## [46,]  0.01 -0.01
## [47,]  0.00  0.03
## [48,]  0.65  0.01
## [49,]  0.01  0.03
## [50,]  0.02 -0.03
```

```r
acf2(ddl.hotel.train, 50)
```




```
##         ACF  PACF
```

```
##  [1,] -0.35 -0.35
##  [2,]  0.10 -0.02
##  [3,] -0.25 -0.26
##  [4,] -0.05 -0.27
##  [5,] -0.14 -0.35
##  [6,]  0.18 -0.15
##  [7,] -0.05 -0.22
##  [8,]  0.11 -0.21
##  [9,]  0.01 -0.14
## [10,]  0.08  0.00
## [11,]  0.01  0.16
## [12,] -0.34 -0.33
## [13,]  0.08 -0.23
## [14,]  0.08  0.10
## [15,]  0.02 -0.05
## [16,]  0.11 -0.08
## [17,]  0.01 -0.03
## [18,] -0.25 -0.26
## [19,]  0.19 -0.01
## [20,] -0.07 -0.03
## [21,]  0.08 -0.02
## [22,] -0.08 -0.02
## [23,]  0.05  0.03
## [24,] -0.09 -0.20
## [25,]  0.15 -0.01
## [26,] -0.18 -0.08
## [27,]  0.16 -0.04
## [28,] -0.15 -0.03
## [29,]  0.11 -0.04
## [30,]  0.02 -0.13
## [31,]  0.04  0.14
## [32,] -0.16  0.01
## [33,]  0.06 -0.05
## [34,] -0.10 -0.02
## [35,]  0.15  0.06
## [36,] -0.05 -0.16
## [37,]  0.04 -0.01
## [38,]  0.00 -0.02
## [39,]  0.04  0.11
## [40,] -0.12 -0.11
## [41,]  0.12  0.02
## [42,] -0.17 -0.10
## [43,]  0.03  0.01
## [44,]  0.11 -0.01
## [45,]  0.03  0.08
## [46,] -0.02 -0.07
## [47,] -0.03  0.01
## [48,]  0.03  0.10
## [49,] -0.15 -0.05
## [50,]  0.11 -0.03
```

From the ACF of the first differencing data, see a yearly seasonal pattern

Estimate parameter

```r
eacf(dl.hotel.train)
```

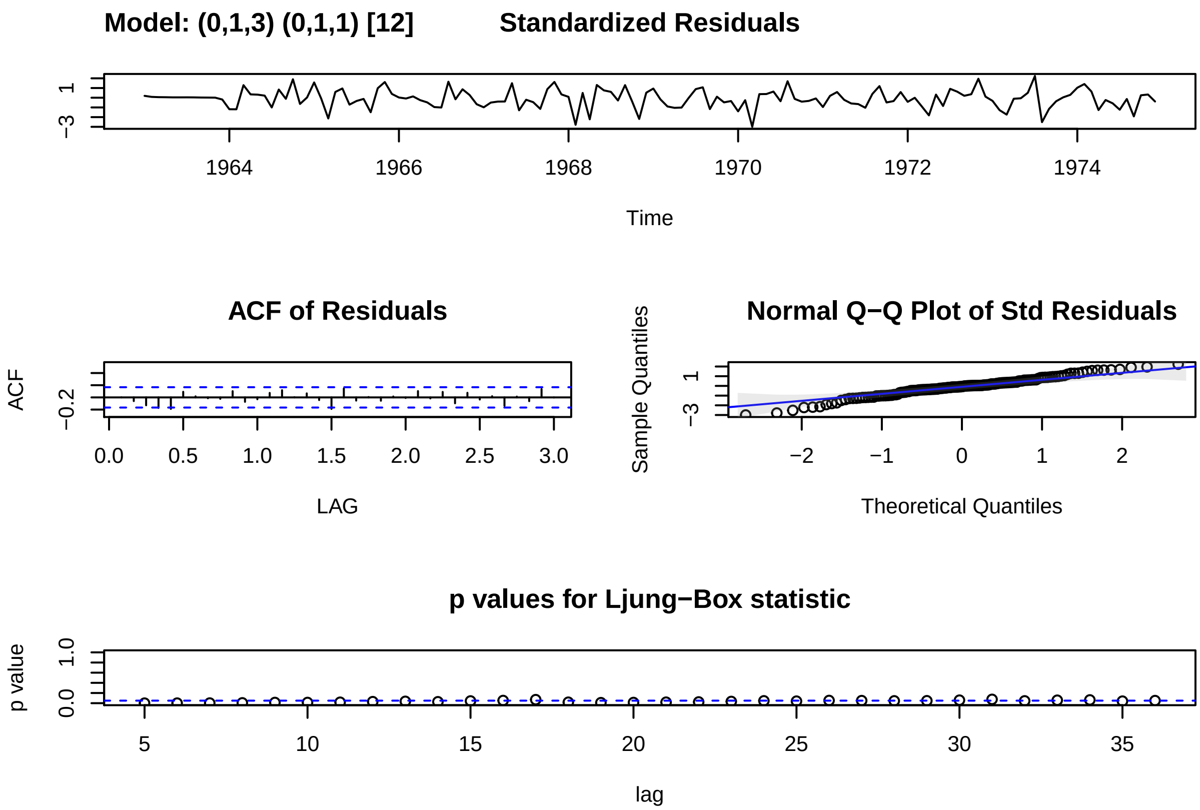
```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 o o x o x o x o x o o  x  o  o
## 1 x o x o x o x o x o o  x  x  o
## 2 o o o o o o o o x o o  x  o  o
## 3 o x o o o o o o o o o  x  o  x
## 4 o x o x o o o o o o o  x  o  x
## 5 o x o x o o o o o o o  x  o  x
## 6 o x o x o o o o o o o  x  o  x
## 7 x x x x o o o o o o o  x  x  x
```

eacf of the first differencing data gives no information, check the eacf of the seasonal differencing data to get further guess of the model parameter(seasonal)

Fit model Tests several possible parameter

```r
sarima(l.hotel.train, 0,1,3, 0,1,1, 12) # AIC -6.92
```

```
## initial  value -3.618183
## iter   2 value -3.846594
## iter   3 value -3.934272
## iter   4 value -3.961719
## iter   5 value -3.965279
## iter   6 value -3.967273
## iter   7 value -3.967794
## iter   8 value -3.968131
## iter   9 value -3.969699
## iter  10 value -3.969757
## iter  11 value -3.969955
## iter  12 value -3.970799
## iter  13 value -3.970998
## iter  14 value -3.971009
## iter  15 value -3.971015
## iter  16 value -3.971015
## iter  16 value -3.971015
## iter  16 value -3.971015
## final  value -3.971015
## converged
## initial  value -3.953875
## iter   2 value -3.956691
## iter   3 value -3.957362
## iter   4 value -3.958505
## iter   5 value -3.958568
## iter   6 value -3.958578
## iter   7 value -3.958581
## iter   7 value -3.958581
## iter   7 value -3.958581
## final  value -3.958581
## converged
```

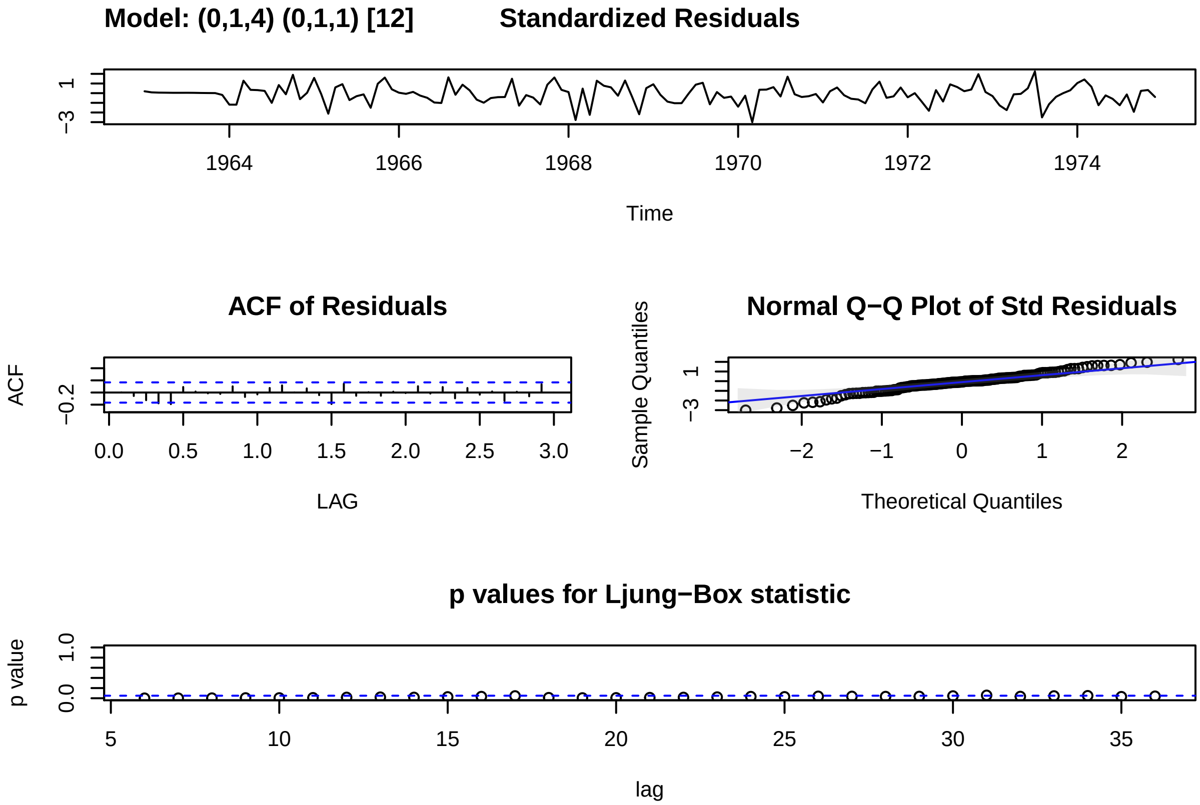



```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, optim.control = list(trace = trc,
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1     ma2      ma3     sma1
##       -0.6436  0.0277  -0.3306  -0.5699
## s.e.   0.0840  0.1109   0.0821   0.0869
##
## sigma^2 estimated as 0.0003435:  log likelihood = 332.69,  aic = -655.39
##
## $degrees_of_freedom
## [1] 127
##
## $ttable
##      Estimate     SE t.value p.value
## ma1   -0.6436 0.0840 -7.6581  0.0000
## ma2    0.0277 0.1109  0.2499  0.8031
## ma3   -0.3306 0.0821 -4.0268  0.0001
## sma1  -0.5699 0.0869 -6.5567  0.0000
##
## $AIC
## [1] -6.920911
##
## $AICc
```

```
## [1] -6.904003
##
## $BIC
## [1] -7.838416
```

```r
sarima(l.hotel.train, 0,1,4, 0,1,1, 12) # AIC -6.9
```

```
## initial  value -3.618183
## iter   2 value -3.856307
## iter   3 value -3.952109
## iter   4 value -3.956633
## iter   5 value -3.960929
## iter   6 value -3.963759
## iter   7 value -3.965321
## iter   8 value -3.966128
## iter   9 value -3.966845
## iter  10 value -3.970863
## iter  11 value -3.970890
## iter  12 value -3.971197
## iter  13 value -3.971208
## iter  14 value -3.971233
## iter  15 value -3.971234
## iter  16 value -3.971234
## iter  16 value -3.971234
## iter  16 value -3.971234
## final  value -3.971234
## converged
## initial  value -3.954142
## iter   2 value -3.957055
## iter   3 value -3.957412
## iter   4 value -3.958481
## iter   5 value -3.958570
## iter   6 value -3.958598
## iter   7 value -3.958606
## iter   8 value -3.958606
## iter   9 value -3.958606
## iter   9 value -3.958606
## iter   9 value -3.958606
## final  value -3.958606
## converged
```

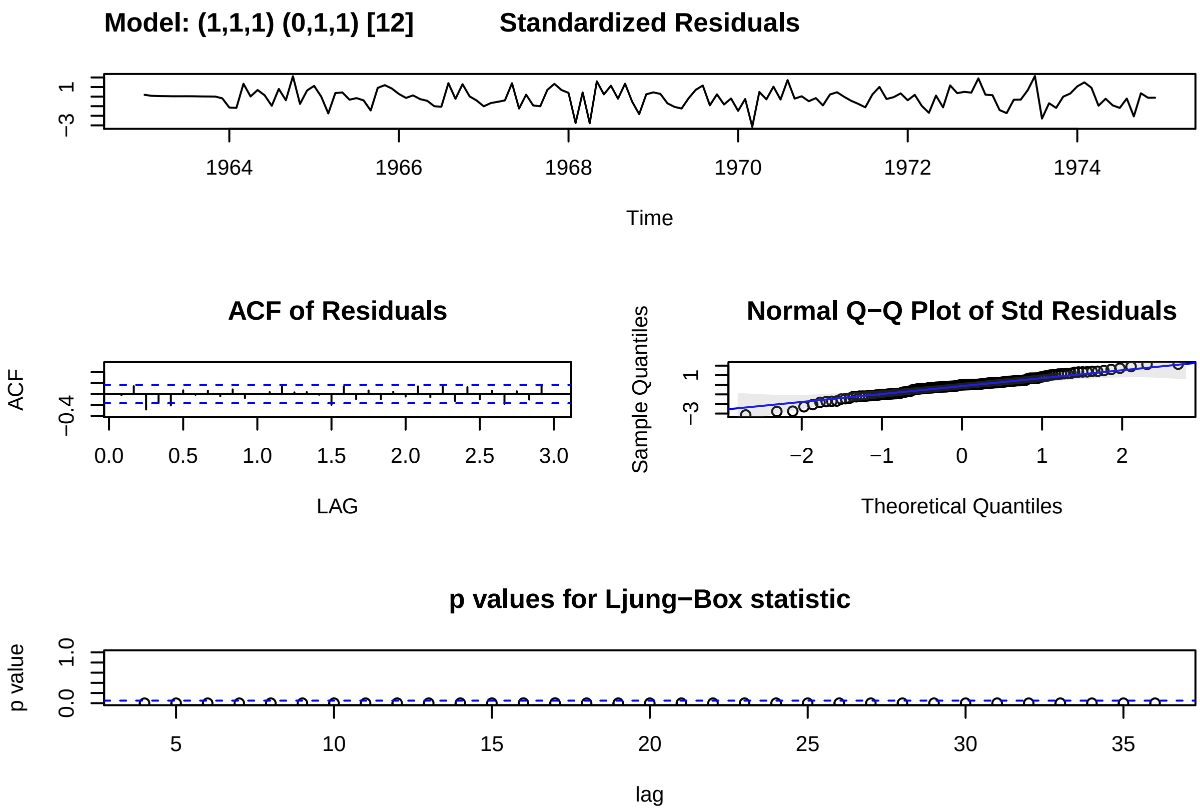



```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, optim.control = list(trace = trc,
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1     ma2      ma3     ma4     sma1
##       -0.6473  0.0231  -0.3321  0.0101  -0.5684
## s.e.   0.0959  0.1262   0.0844  0.1265   0.0889
##
## sigma^2 estimated as 0.0003435:  log likelihood = 332.7,  aic = -653.39
##
## $degrees_of_freedom
## [1] 126
##
## $ttable
##      Estimate     SE t.value p.value
## ma1   -0.6473 0.0959 -6.7527  0.0000
## ma2    0.0231 0.1262  0.1828  0.8553
## ma3   -0.3321 0.0844 -3.9338  0.0001
## ma4    0.0101 0.1265  0.0798  0.9365
## sma1  -0.5684 0.0889 -6.3910  0.0000
##
## $AIC
## [1] -6.906753
##
```

```
## $AICc
## [1] -6.888606
##
## $BIC
## [1] -7.803634
```

```r
sarima(l.hotel.train, 1,1,1, 0,1,1, 12) #AIC -6.8
```

```
## initial  value -3.619580
## iter   2 value -3.761664
## iter   3 value -3.818141
## iter   4 value -3.821599
## iter   5 value -3.823133
## iter   6 value -3.823140
## iter   7 value -3.823149
## iter   8 value -3.823158
## iter   9 value -3.823187
## iter  10 value -3.823210
## iter  11 value -3.823250
## iter  12 value -3.823287
## iter  13 value -3.823411
## iter  14 value -3.823514
## iter  15 value -3.823721
## iter  16 value -3.823919
## iter  17 value -3.824582
## iter  18 value -3.825201
## iter  19 value -3.826905
## iter  20 value -3.828765
## iter  21 value -3.835236
## iter  22 value -3.842605
## iter  23 value -3.848161
## iter  24 value -3.851174
## iter  25 value -3.853932
## iter  26 value -3.854899
## iter  27 value -3.855000
## iter  28 value -3.855039
## iter  29 value -3.855039
## iter  29 value -3.855039
## iter  29 value -3.855039
## final  value -3.855039
## converged
## initial  value -3.880827
## iter   2 value -3.918116
## iter   3 value -3.919192
## iter   4 value -3.921234
## iter   5 value -3.921778
## iter   6 value -3.921779
## iter   6 value -3.921779
## iter   6 value -3.921779
## final  value -3.921779
## converged
```

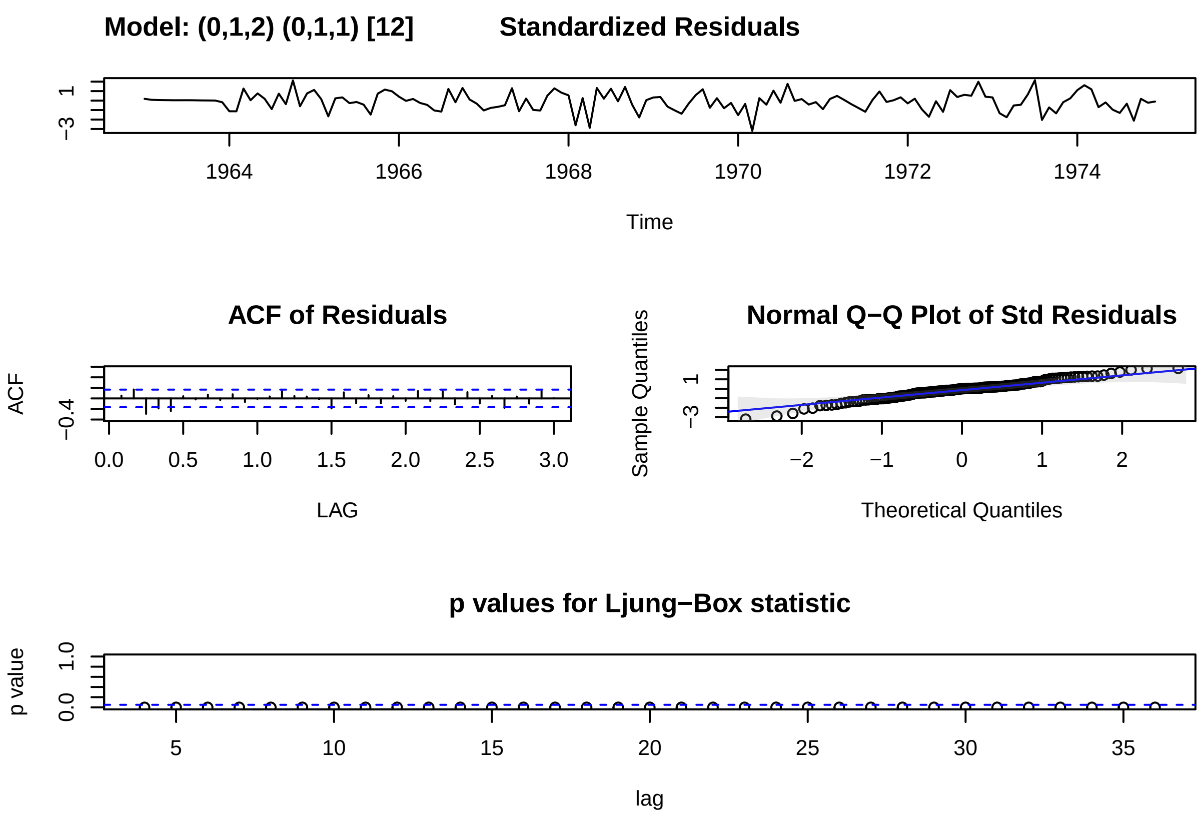



```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, optim.control = list(trace = trc,
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ma1     sma1
##       0.3075  -0.9640  -0.5431
## s.e.  0.0875   0.0252   0.0909
##
## sigma^2 estimated as 0.0003716:  log likelihood = 327.87,  aic = -647.74
##
## $degrees_of_freedom
## [1] 128
##
## $ttable
##      Estimate     SE  t.value p.value
## ar1    0.3075 0.0875   3.5119   6e-04
## ma1   -0.9640 0.0252 -38.2915   0e+00
## sma1  -0.5431 0.0909  -5.9734   0e+00
##
## $AIC
## [1] -6.856096
##
## $AICc
## [1] -6.840209
```

```
##
## $BIC
## [1] -7.794225
```

```r
sarima(l.hotel.train, 0,1,2, 0,1,1, 12) # AIC -6.8
```

```
## initial  value -3.618183
## iter   2 value -3.775789
## iter   3 value -3.784697
## iter   4 value -3.868252
## iter   5 value -3.903339
## iter   6 value -3.913558
## iter   7 value -3.916894
## iter   8 value -3.916901
## iter   9 value -3.916986
## iter   9 value -3.916987
## iter   9 value -3.916987
## final  value -3.916987
## converged
## initial  value -3.907874
## iter   2 value -3.908885
## iter   3 value -3.909396
## iter   4 value -3.909413
## iter   5 value -3.909413
## iter   5 value -3.909413
## iter   5 value -3.909413
## final  value -3.909413
## converged
```

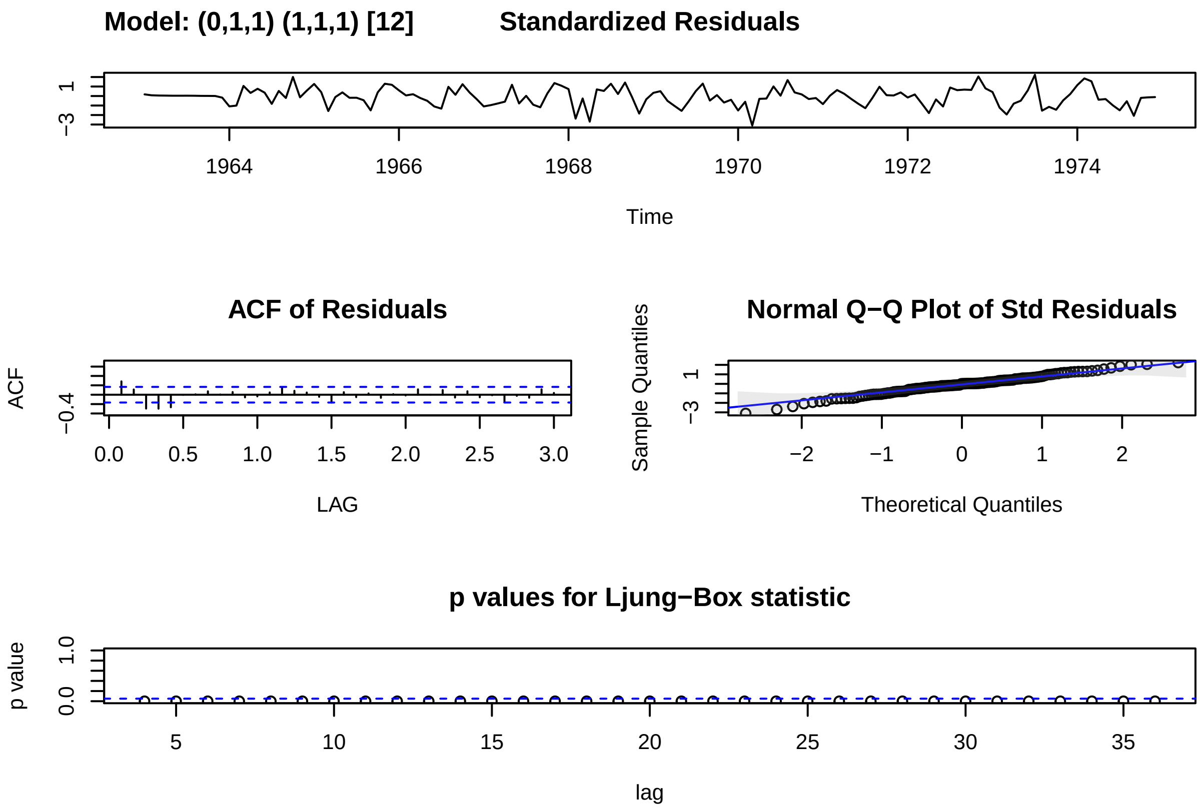


```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, optim.control = list(trace = trc,
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1      ma2     sma1
##       -0.7387  -0.2083  -0.5236
## s.e.   0.0706   0.0666   0.0906
##
## sigma^2 estimated as 0.000382:  log likelihood = 326.25,  aic = -644.5
##
## $degrees_of_freedom
## [1] 128
##
## $ttable
##      Estimate     SE  t.value p.value
## ma1   -0.7387 0.0706 -10.4643  0.0000
## ma2   -0.2083 0.0666  -3.1273  0.0022
## sma1  -0.5236 0.0906  -5.7790  0.0000
##
## $AIC
## [1] -6.8284
##
## $AICc
## [1] -6.812513
##
## $BIC
## [1] -7.766529
```

```r
sarima(l.hotel.train, 0,1,1, 1,1,1, 12)# AIC -6.75
```

```
## initial  value -3.628213
## iter   2 value -3.794209
## iter   3 value -3.815477
## iter   4 value -3.823057
## iter   5 value -3.825543
## iter   6 value -3.826248
## iter   7 value -3.826633
## iter   8 value -3.826934
## iter   9 value -3.827712
## iter  10 value -3.829404
## iter  11 value -3.829525
## iter  12 value -3.829645
## iter  13 value -3.829654
## iter  14 value -3.829655
## iter  14 value -3.829655
## iter  14 value -3.829655
## final  value -3.829655
## converged
## initial  value -3.851850
## iter   2 value -3.865842
## iter   3 value -3.872573
```

```
## iter   4 value -3.873454
## iter   5 value -3.874285
## iter   6 value -3.875736
## iter   7 value -3.875854
## iter   8 value -3.875924
## iter   9 value -3.875927
## iter  10 value -3.875931
## iter  11 value -3.875932
## iter  12 value -3.875933
## iter  13 value -3.875934
## iter  14 value -3.875934
## iter  14 value -3.875934
## iter  14 value -3.875934
## final  value -3.875934
## converged
```



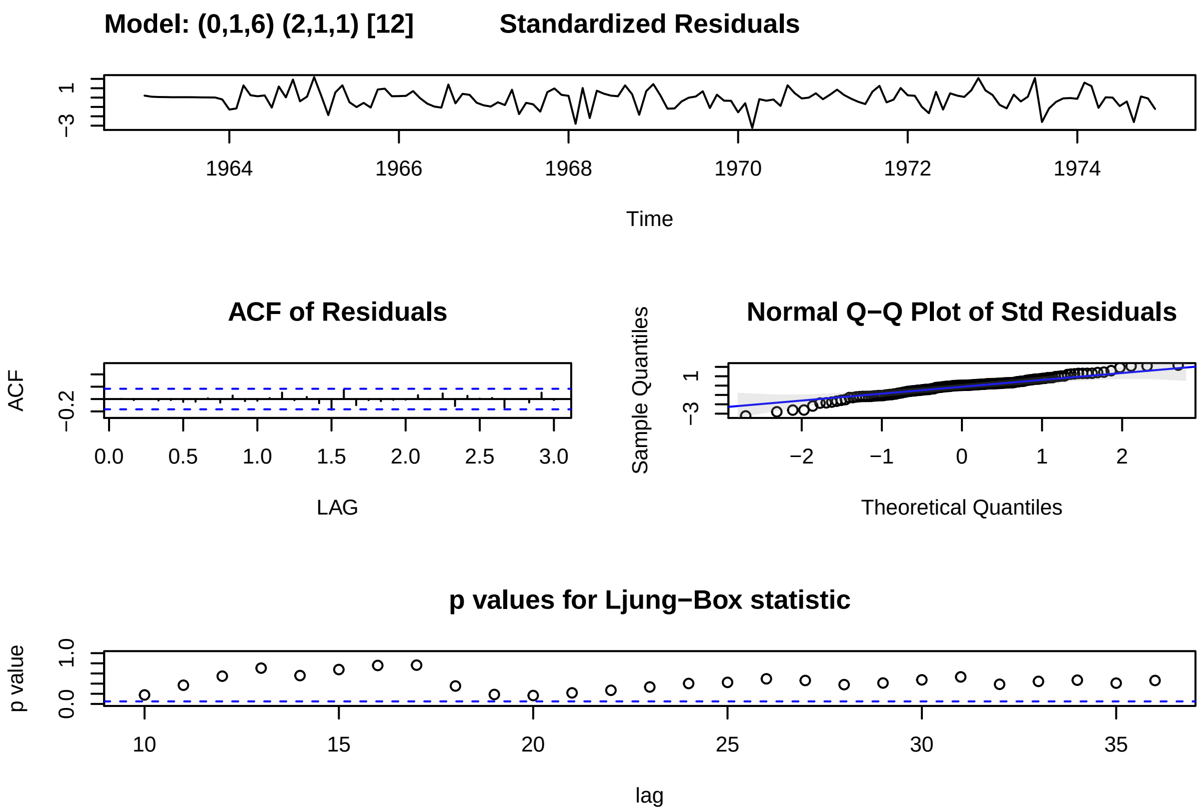

```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, optim.control = list(trace = trc,
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1     sar1     sma1
##       -0.9465   0.0012  -0.5021
## s.e.   0.0294   0.1905   0.1737
##
## sigma^2 estimated as 0.0004096:  log likelihood = 321.87,  aic = -635.73
```

```
##
## $degrees_of_freedom
## [1] 128
##
## $ttable
##      Estimate     SE  t.value p.value
## ma1   -0.9465 0.0294 -32.1514  0.0000
## sar1   0.0012 0.1905   0.0060  0.9952
## sma1  -0.5021 0.1737  -2.8900  0.0045
##
## $AIC
## [1] -6.75855
##
## $AICc
## [1] -6.742662
##
## $BIC
## [1] -7.696678
```

```
sarima(l.hotel.train, 0,1,6, 2,1,1, 12) # AIC -7.00
```

```
## initial  value -3.621738
## iter   2 value -3.941921
## iter   3 value -3.973425
## iter   4 value -3.992559
## iter   5 value -4.003881
## iter   6 value -4.032396
## iter   7 value -4.045952
## iter   8 value -4.048727
## iter   9 value -4.053736
## iter  10 value -4.057005
## iter  11 value -4.061510
## iter  12 value -4.064213
## iter  13 value -4.067672
## iter  14 value -4.068399
## iter  15 value -4.069049
## iter  16 value -4.071590
## iter  17 value -4.075770
## iter  18 value -4.076852
## iter  19 value -4.082695
## iter  20 value -4.083603
## iter  21 value -4.084407
## iter  22 value -4.086668
## iter  23 value -4.089393
## iter  24 value -4.090503
## iter  25 value -4.092870
## iter  26 value -4.094454
## iter  27 value -4.094492
## iter  28 value -4.094503
## iter  29 value -4.095508
## iter  29 value -4.095508
## iter  30 value -4.095513
## iter  31 value -4.097573
## iter  31 value -4.097573
## iter  32 value -4.097841
```

```
## iter  32 value -4.097841
## iter  33 value -4.097865
## iter  33 value -4.097865
## iter  33 value -4.097865
## final  value -4.097865
## converged
## initial  value -3.998589
## iter   2 value -4.022125
## iter   3 value -4.026332
## iter   4 value -4.026408
## iter   5 value -4.029059
## iter   6 value -4.029449
## iter   7 value -4.029593
## iter   8 value -4.029684
## iter   9 value -4.030067
## iter  10 value -4.030201
## iter  11 value -4.030385
## iter  12 value -4.030411
## iter  13 value -4.030415
## iter  14 value -4.030416
## iter  15 value -4.030416
## iter  15 value -4.030416
## iter  15 value -4.030416
## final  value -4.030416
## converged
```



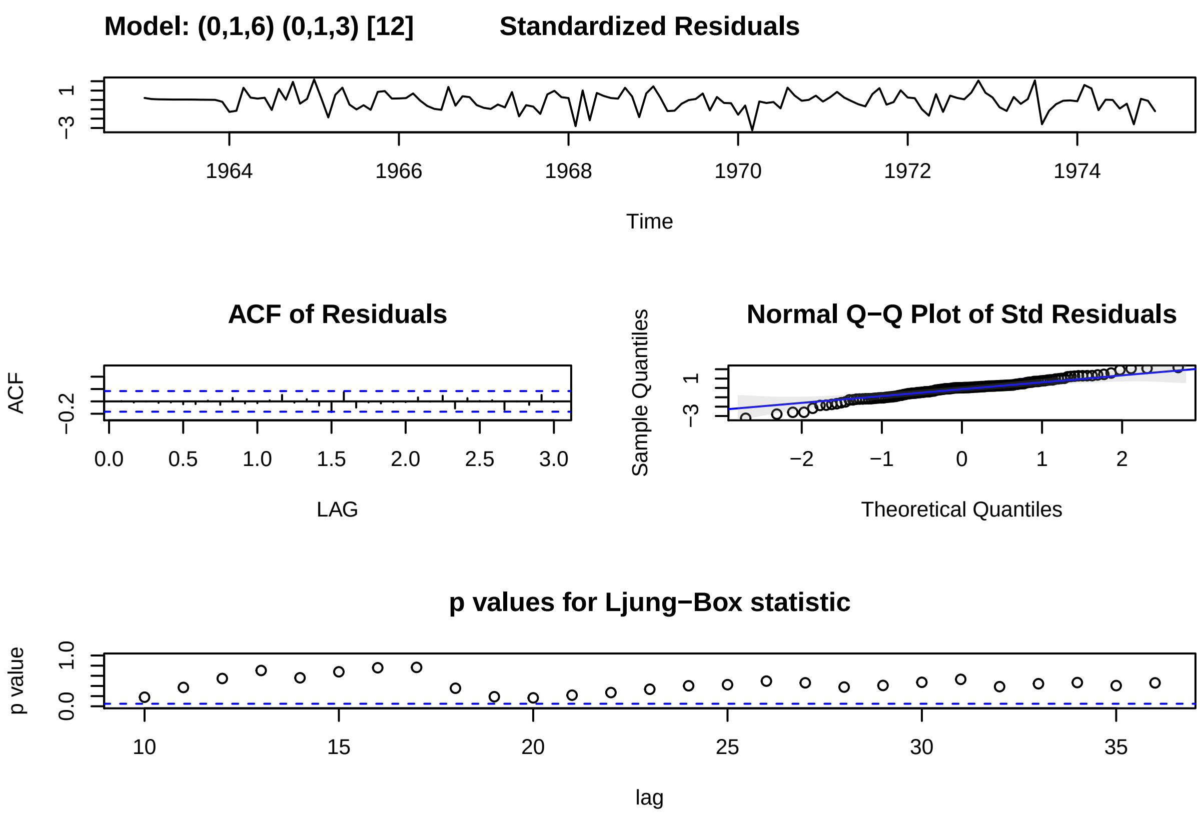

```
## $fit
##
## Call:
```

```
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, optim.control = list(trace = trc,
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1     ma2      ma3      ma4      ma5     ma6    sar1    sar2
##       -0.7351  0.0005  -0.4504  -0.0730  -0.0264  0.3719  0.0607  0.0580
## s.e.   0.0847  0.1084   0.1033   0.1147   0.1096  0.0942  0.2986  0.1875
##          sma1
##       -0.6163
## s.e.   0.2848
##
## sigma^2 estimated as 0.000296:  log likelihood = 342.1,  aic = -664.21
##
## $degrees_of_freedom
## [1] 122
##
## $ttable
##      Estimate     SE t.value p.value
## ma1   -0.7351 0.0847 -8.6808  0.0000
## ma2    0.0005 0.1084  0.0050  0.9960
## ma3   -0.4504 0.1033 -4.3584  0.0000
## ma4   -0.0730 0.1147 -0.6363  0.5258
## ma5   -0.0264 0.1096 -0.2414  0.8097
## ma6    0.3719 0.0942  3.9482  0.0001
## sar1   0.0607 0.2986  0.2034  0.8391
## sar2   0.0580 0.1875  0.3093  0.7576
## sma1  -0.6163 0.2848 -2.1639  0.0324
##
## $AIC
## [1] -7.000033
##
## $AICc
## [1] -6.974657
##
## $BIC
## [1] -7.81442
```

```
sarima(l.hotel.train, 0,1,6, 0,1,3, 12) #AIC -7.00
```

```
## initial  value -3.618183
## iter   2 value -3.877024
## iter   3 value -3.953588
## iter   4 value -3.977262
## iter   5 value -3.996886
## iter   6 value -4.017744
## iter   7 value -4.020510
## iter   8 value -4.020695
## iter   9 value -4.022566
## iter  10 value -4.022699
## iter  11 value -4.022723
## iter  12 value -4.022724
## iter  13 value -4.022725
## iter  14 value -4.022725
## iter  15 value -4.022725
```

```
## iter  15 value -4.022725
## iter  15 value -4.022725
## final  value -4.022725
## converged
## initial  value -4.026609
## iter   2 value -4.026976
## iter   3 value -4.029252
## iter   4 value -4.030194
## iter   5 value -4.030455
## iter   6 value -4.030500
## iter   7 value -4.030509
## iter   8 value -4.030513
## iter   9 value -4.030513
## iter  10 value -4.030513
## iter  10 value -4.030513
## iter  10 value -4.030513
## final  value -4.030513
## converged
```




```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, optim.control = list(trace = trc,
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1     ma2      ma3      ma4      ma5    ma6     sma1    sma2
##       -0.7342  0.0006  -0.4509  -0.0737  -0.0262  0.372  -0.5543  0.0259
```

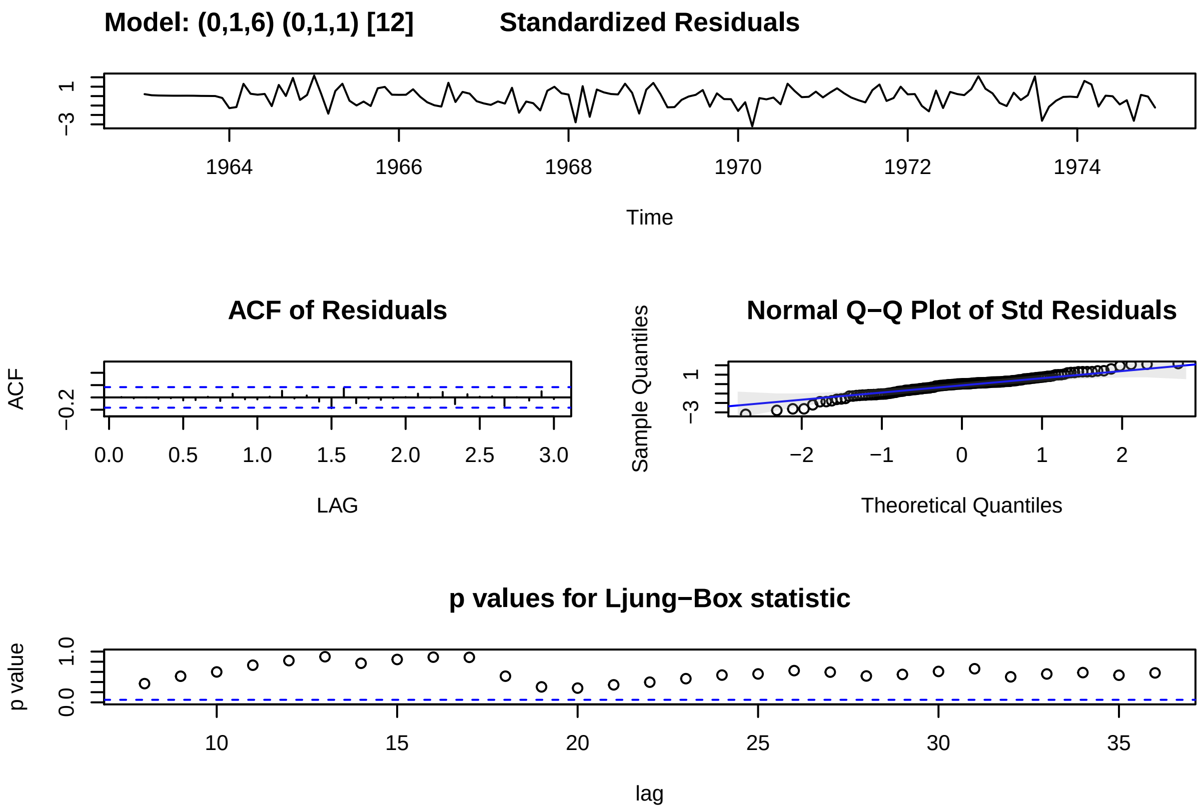
```
## s.e.    0.0850  0.1084   0.1033   0.1151   0.1098  0.094   0.1027  0.1172
##           sma3
##        -0.0403
## s.e.    0.1150
##
## sigma^2 estimated as 0.0002959:  log likelihood = 342.12,  aic = -664.23
##
## $degrees_of_freedom
## [1] 122
##
## $ttable
##      Estimate     SE t.value p.value
## ma1   -0.7342 0.0850 -8.6387  0.0000
## ma2    0.0006 0.1084  0.0053  0.9958
## ma3   -0.4509 0.1033 -4.3646  0.0000
## ma4   -0.0737 0.1151 -0.6409  0.5228
## ma5   -0.0262 0.1098 -0.2389  0.8116
## ma6    0.3720 0.0940  3.9557  0.0001
## sma1  -0.5543 0.1027 -5.3982  0.0000
## sma2   0.0259 0.1172  0.2208  0.8256
## sma3  -0.0403 0.1150 -0.3506  0.7265
##
## $AIC
## [1] -7.000506
##
## $AICc
## [1] -6.97513
##
## $BIC
## [1] -7.814892
```

Get the best model

```r
sarima(l.hotel.train, 0,1,6, 0,1,1, 12)
```

```
## initial  value -3.618183
## iter   2 value -3.879886
## iter   3 value -3.966292
## iter   4 value -3.982463
## iter   5 value -4.000071
## iter   6 value -4.017342
## iter   7 value -4.021022
## iter   8 value -4.022382
## iter   9 value -4.022683
## iter  10 value -4.022703
## iter  11 value -4.022716
## iter  12 value -4.022716
## iter  13 value -4.022716
## iter  13 value -4.022716
## iter  13 value -4.022716
## final  value -4.022716
## converged
## initial  value -4.026552
## iter   2 value -4.026884
## iter   3 value -4.028991
```

```
## iter   4 value -4.029885
## iter   5 value -4.030013
## iter   6 value -4.030027
## iter   7 value -4.030027
## iter   8 value -4.030028
## iter   8 value -4.030028
## iter   8 value -4.030028
## final  value -4.030028
## converged
```




```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, optim.control = list(trace = trc,
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1      ma2      ma3      ma4      ma5     ma6     sma1
##       -0.7372  -0.0013  -0.4477  -0.0745  -0.0210  0.3685  -0.5510
## s.e.   0.0840   0.1083   0.1034   0.1111   0.1051  0.0940   0.0917
##
## sigma^2 estimated as 0.0002964:  log likelihood = 342.05,  aic = -668.11
##
## $degrees_of_freedom
## [1] 124
##
## $ttable
##      Estimate     SE t.value p.value
```

```
## ma1   -0.7372 0.0840 -8.7759  0.0000
## ma2   -0.0013 0.1083 -0.0120  0.9905
## ma3   -0.4477 0.1034 -4.3289  0.0000
## ma4   -0.0745 0.1111 -0.6706  0.5037
## ma5   -0.0210 0.1051 -0.1994  0.8422
## ma6    0.3685 0.0940  3.9200  0.0001
## sma1  -0.5510 0.0917 -6.0102  0.0000
##
## $AIC
## [1] -7.026553
##
## $AICc
## [1] -7.005257
##
## $BIC
## [1] -7.882187
```

```
# by several trials of parameters, will write more choices of parameters for comparison purpose
```

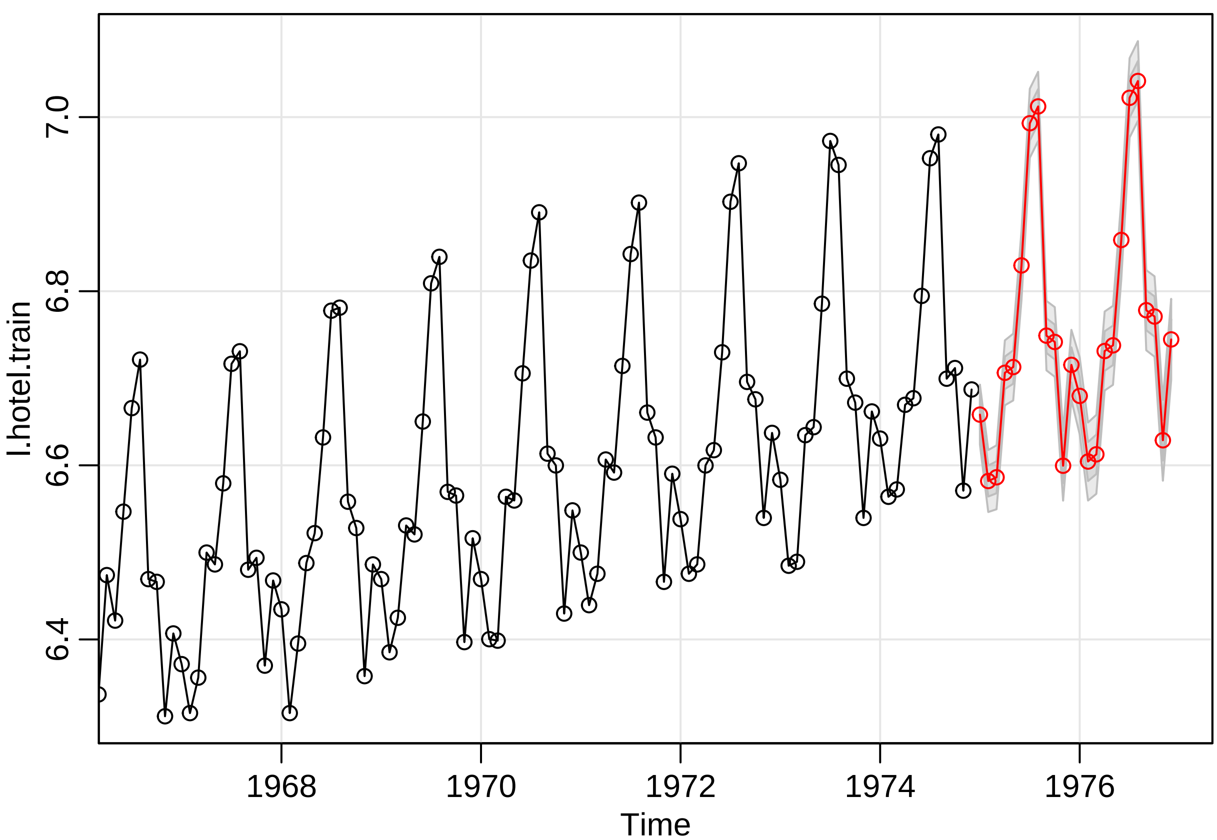
Verify the model:

Take the last two years as the test set

```
#hotel.train = ts(hotel.ts[1:144], start=c(1963, 1), end = c(1974,12), frequency = 12 ) # the train set
l.hotel.train = log(hotel.train)
#hotel.test = hotel[145:168,] # test set
```

predict the "unseen" test data for model verify purpose

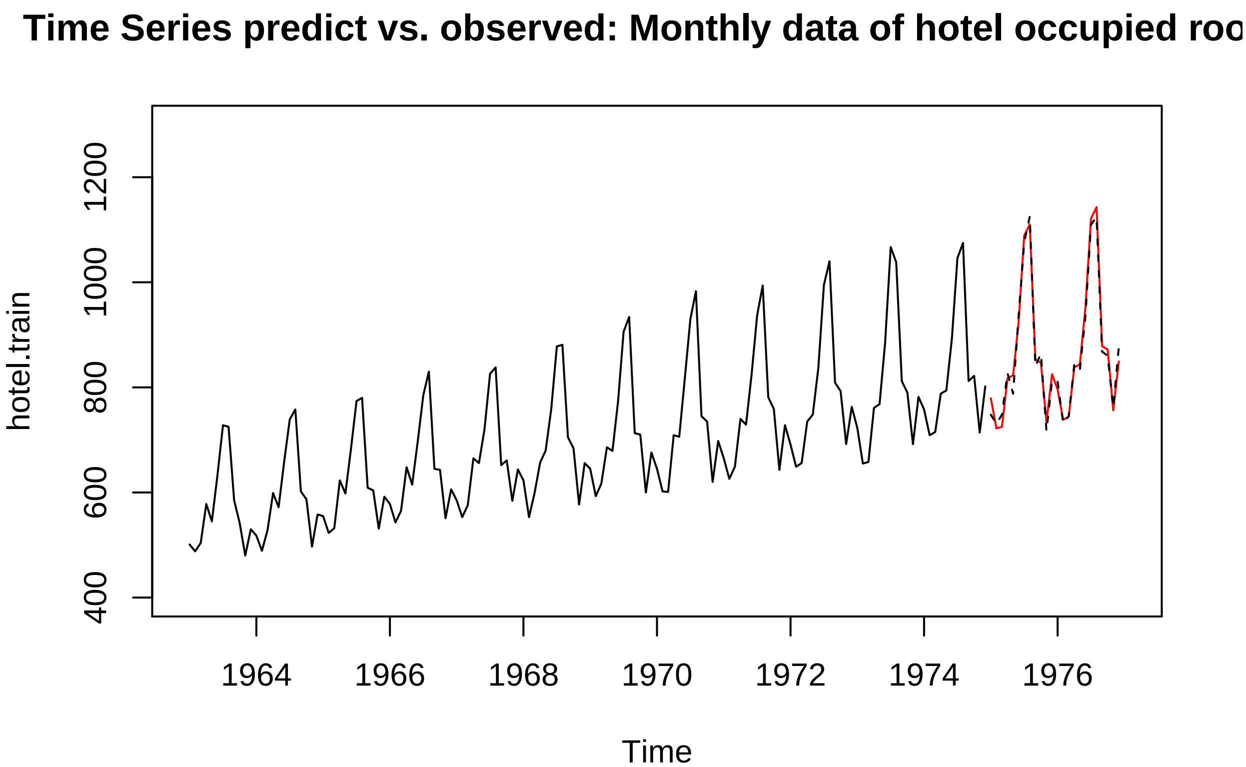
```
test.arima = sarima.for(l.hotel.train, 24, 0,1,6, 0,1,1, 12)
```



Plot the predicted value of the test data along with the test data

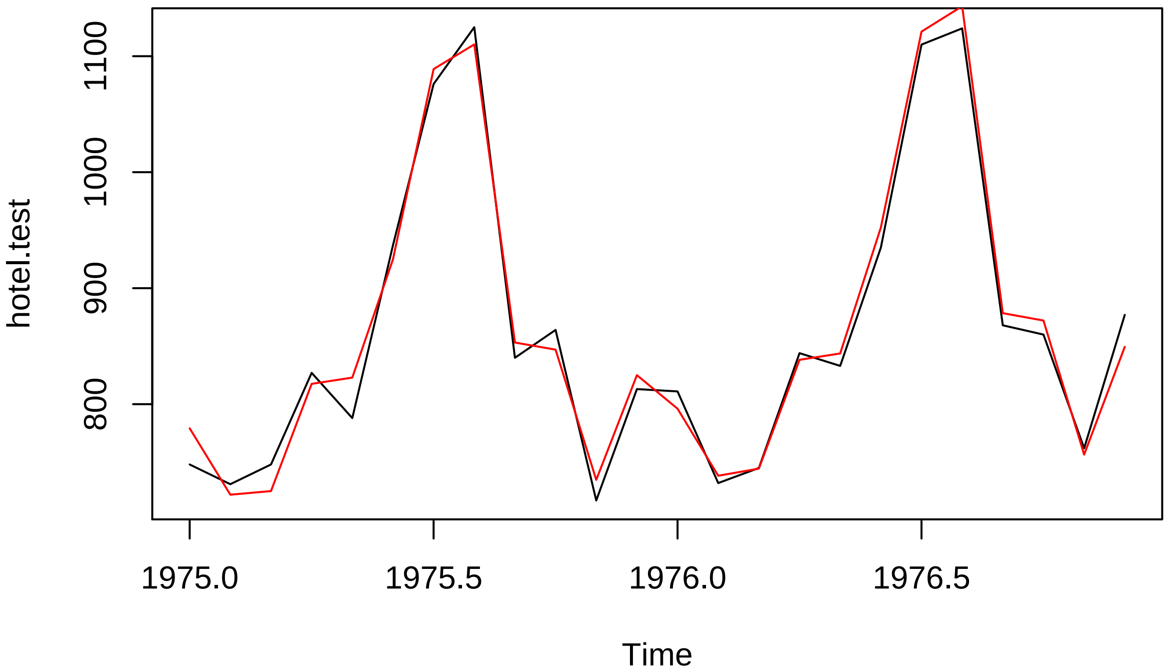
```
exp.test.arima = exp(test.arima$pred)

plot(hotel.train, xlim=c(1963, 1977), ylim=c(400,1300),
     main="Time Series predict vs. observed: Monthly data of hotel occupied room ")
lines(exp.test.arima, col="red") # red line denotes the predicted value
lines(hotel.test, lty = 2)
```



Comparison of test set and observed set

```
plot(hotel.test)
lines(exp.test.arima, col="red") # red line denotes the predicted value
```



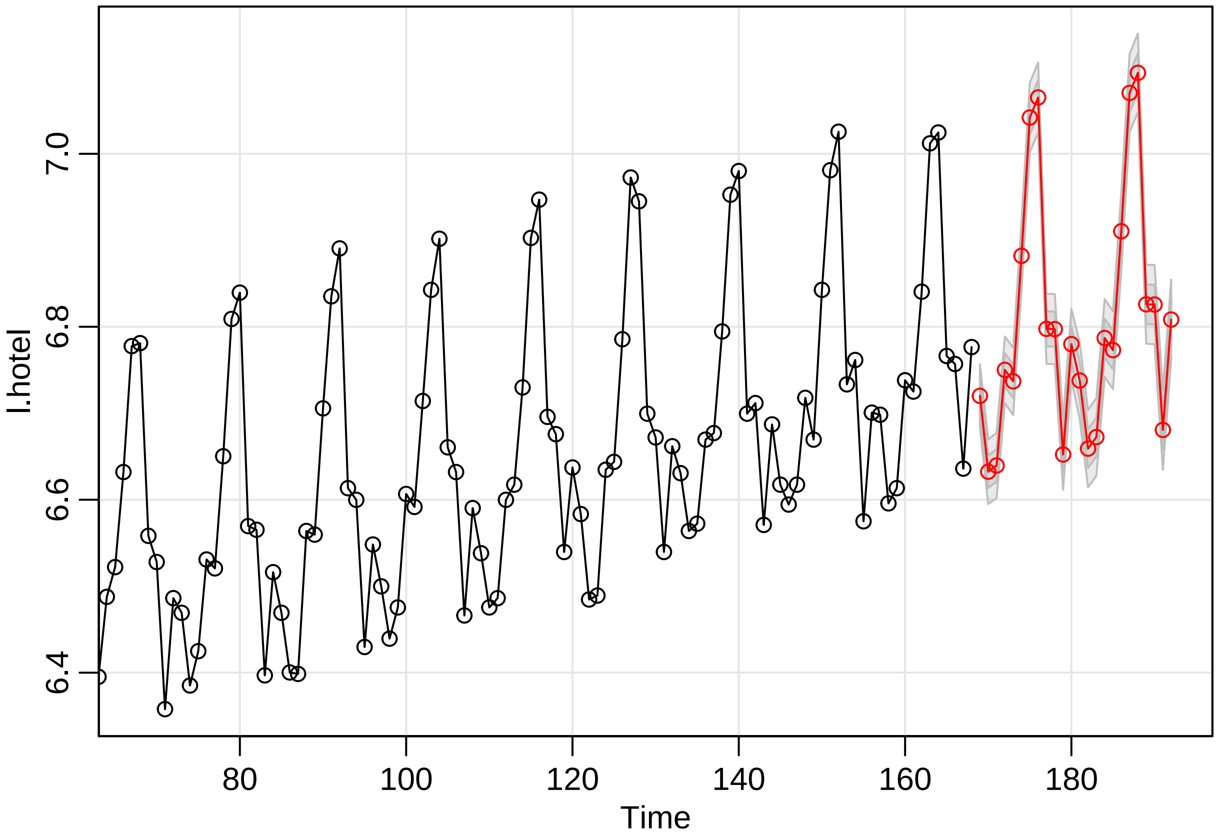
Mean square error

```
mean((exp.test.arima - hotel.test)^2) # 274.4558
```

```
## [1] 274.4558
```

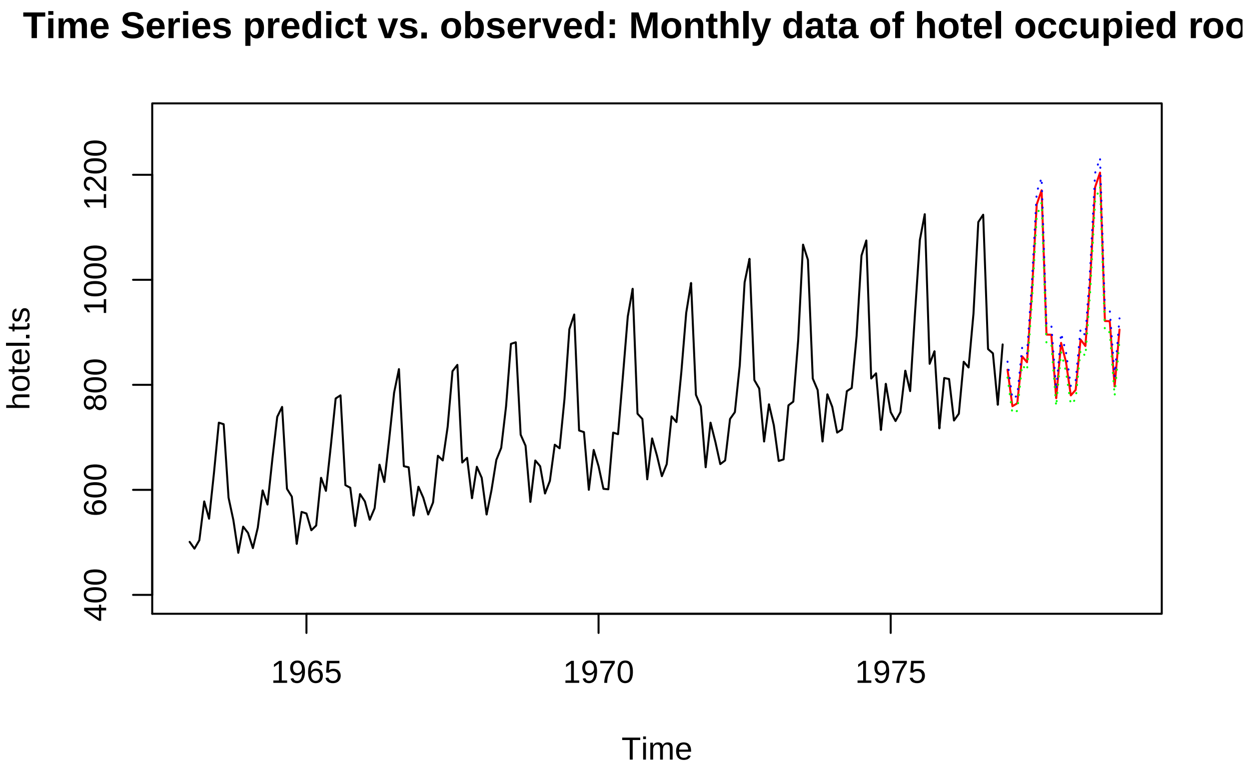
Prediction

```
pred.hotel = sarima.for(l.hotel, 24,  0,1,6, 0,1,1, 12)
```



Plot back to the original scale

```
plot(hotel.ts, xlim=c(1963, 1979), ylim=c(400,1300),
     main="Time Series predict vs. observed: Monthly data of hotel occupied room ")
arimapre = ts(exp(pred.hotel$pred), start=c(1977, 1), end = c(1978,12), frequency = 12)
lines(arimapre, col ="red")
# add CI for predicted value
lines(ts(exp(pred.hotel$pred + pred.hotel$se), start=c(1977, 1), end = c(1978,12), frequency = 12),
      col="blue", lty =3) # upper CI
lines(ts(exp(pred.hotel$pred - pred.hotel$se), start=c(1977, 1), end = c(1978,12), frequency = 12), col=
```


Time Series predict vs. observed: Monthly data of hotel occupied roo
hotel.ts
1200
1000
800
600
400
1965
1970
1975
Time