National University of Singapore

ST3233: Applied Time Series Analysis
Assignment 2

Final Version

Group Number: 07

Oct.29.2016

Group Members	
He Guanzhou	A0153091M
Hu Hao	A0152988L
Hu Jiongyi	A0153190M
Jiang Zeyu	A0152953B
Ye Rong	A0153626E

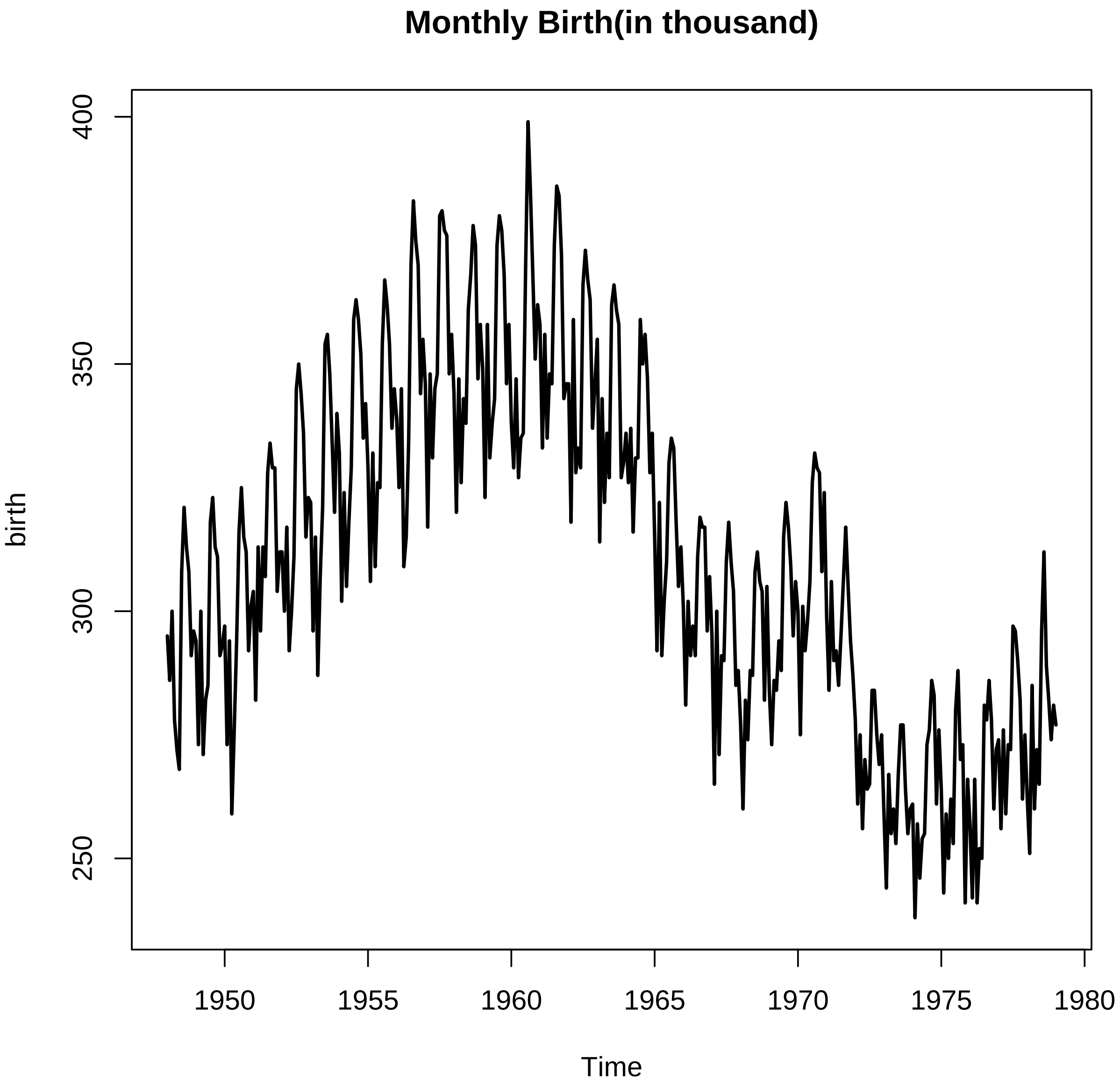
Contents

1 Exercise 1 (Can one trust confidence intervals?)

```
library(forecast)

## Loading required package:  zoo
##
## Attaching package:  'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
## Loading required package:  timeDate
## This is forecast 7.1

library(fpp)

## Loading required package:  fma
## Loading required package:  tseries
## Loading required package:  expsmooth
## Loading required package:  lmtest
```

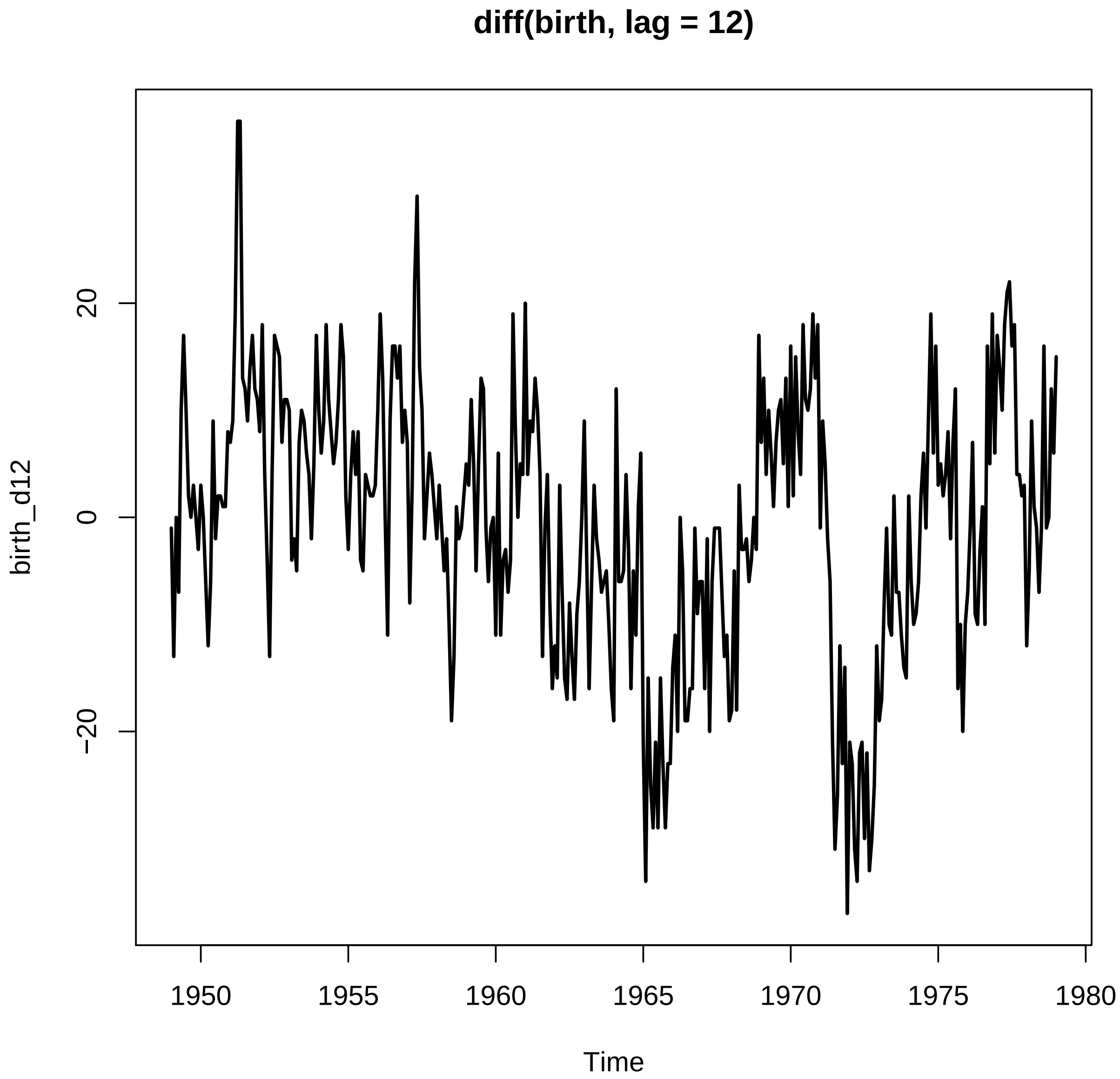
1. Fit a SARIMA model

```
load("E:/ST3233/Assignment2/Datasets/tsa3.rda")
plot(birth,lwd = 2 ,main = "Monthly Birth(in thousand)")
```

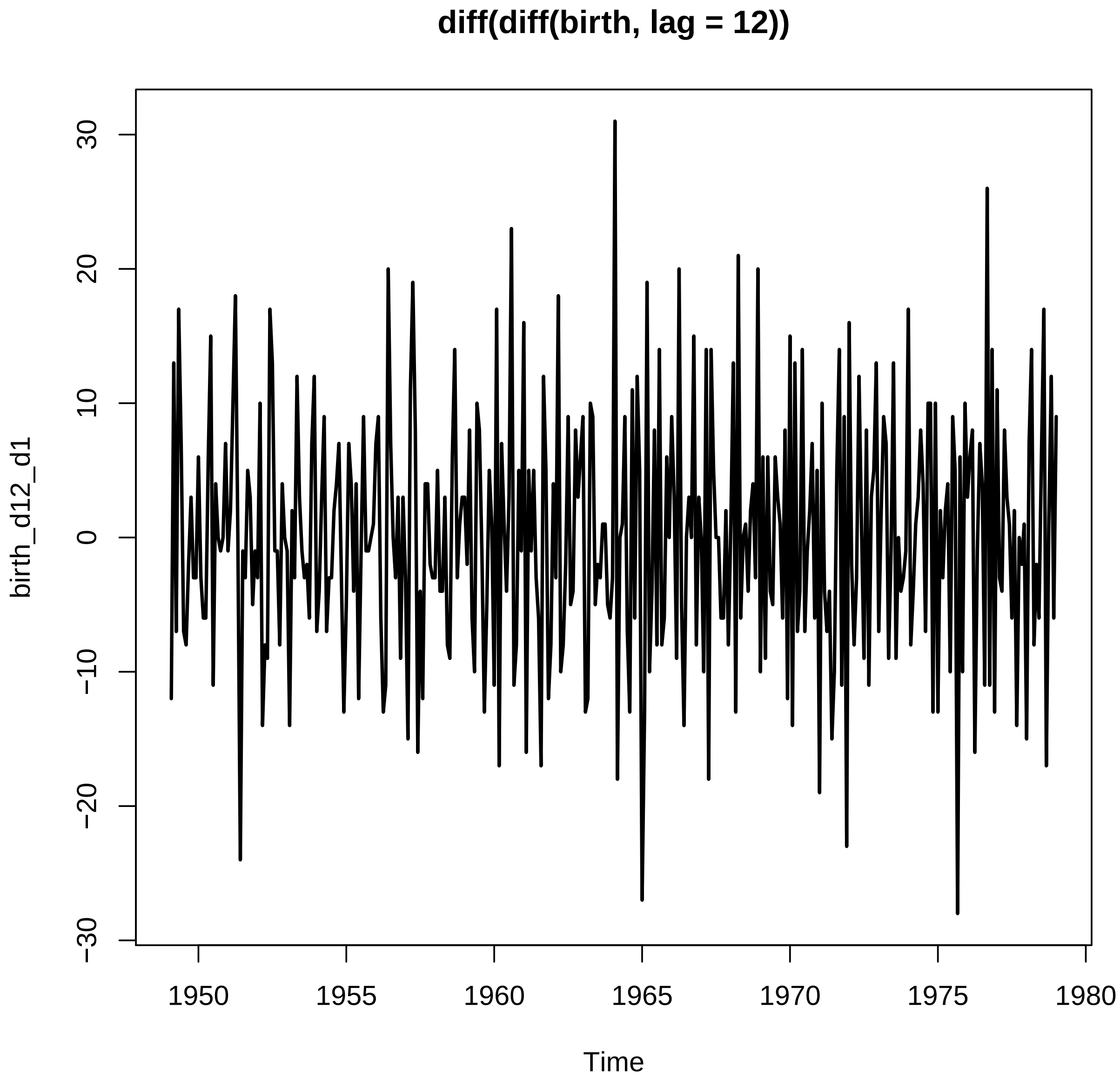


From the plot, seasonal component = 12, so we differentiate it twice: lag = 12, lag = 1, and apply SARIMA model.

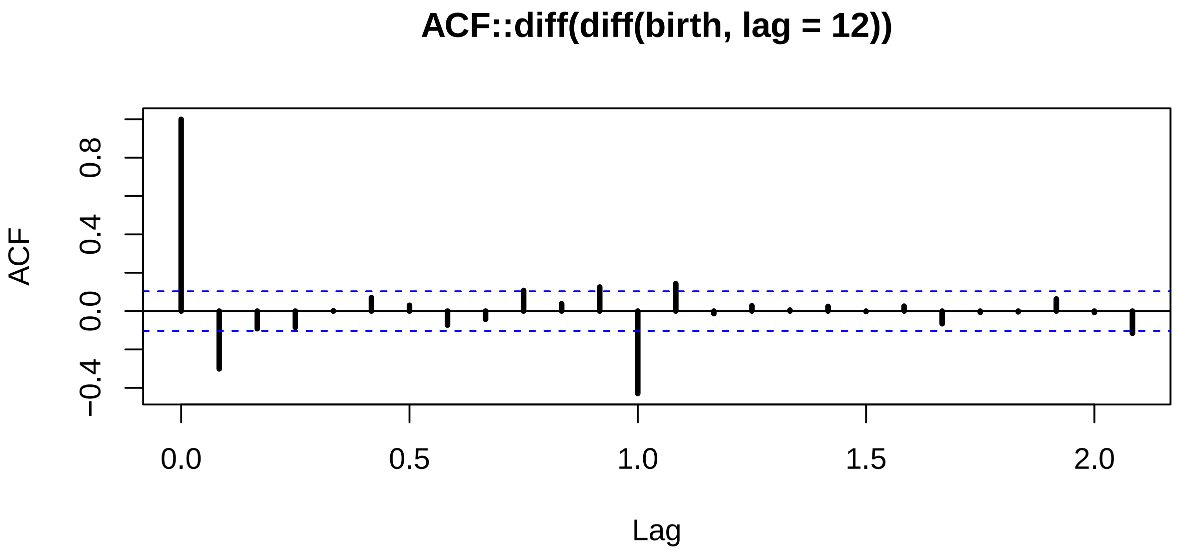
```
birth_d12 <- diff(birth, lag = 12)
plot(birth_d12,lwd = 2,main = "diff(birth, lag = 12)")
```

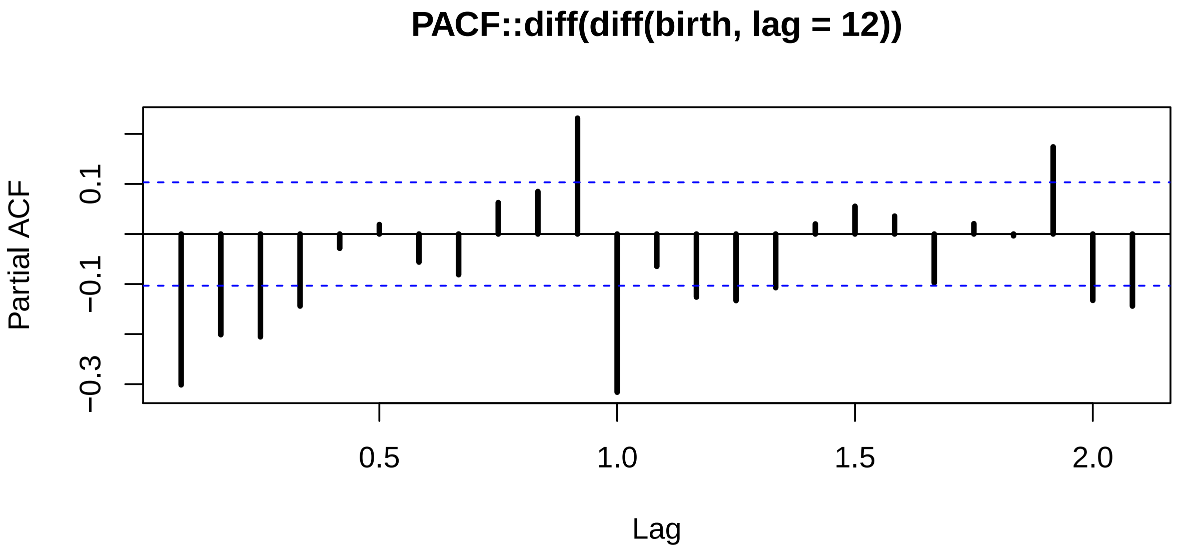


```
birth_d12_d1 <- diff(birth_d12,lag = 1)
plot(birth_d12_d1,lwd = 2,main = "diff(diff(birth, lag = 12))")
```



```r
par(mfrow=c(2,1))
acf(birth_d12_d1,lwd = 3, main = "ACF::diff(diff(birth, lag = 12))")
pacf(birth_d12_d1,lwd = 3, main = "PACF::diff(diff(birth, lag = 12))")
```





From acf plot ,q $\leq 1, and from partial - acf plot, p \leq 4. For SARIMA model, generally, P, Q \leq 1$

```
AIC_best = 10**6
k <-  0
for(p in 0:4){
  for(q in 0:1){
    for(P in 0:1){
      for(Q in 0:1){
        fit_sarima = Arima(birth, order = c(p,1,q), seasonal = c(P,1,Q))
        if (fit_sarima$aic < AIC_best){
          k = k + 1
          AIC_best <- fit_sarima$aic
          cat("model",k,"\t p=",p,"q=",q,"P=",P,"Q=",Q,"\t AIC=",
              AIC_best,"\t Number of parameters=", p+q+P+Q, "\n")
        }
      }
    }
  }
}

## model 1   p= 0 q= 0 P= 0 Q= 0   AIC= 2621.434   Number of parameters= 0
## model 2   p= 0 q= 0 P= 0 Q= 1   AIC= 2472.199   Number of parameters= 1
## model 3   p= 0 q= 1 P= 0 Q= 1   AIC= 2428.557   Number of parameters= 2
## model 4   p= 1 q= 1 P= 0 Q= 1   AIC= 2419.855   Number of parameters= 3
## model 5   p= 1 q= 1 P= 1 Q= 1   AIC= 2419.66   Number of parameters= 4
## model 6   p= 4 q= 0 P= 0 Q= 1   AIC= 2417.468   Number of parameters= 5
```

From the output, since SARIMA((4,1,0)(0,1,1)[12]) gives lowest AIC, we choose it.(number of parameters = 4+0+0+1 = 5)

```
birth_fit <- Arima(birth, order = c(4,1,0), seasonal = c(0,1,1))
```

Conclusion: The SARIMA model is : SARIMA((4,1,0)(0,1,1)[12]) 2. Use your model to get a 80% confidence interval for the number of births in Feb 1979.

```
forecast(fit_sarima, h=1)

##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Feb 1979        256.856 248.1997 265.5123 243.6174 270.0946
```

Thus,the 80% confidence interval for the number of births in Feb 1979 is [250.0562,267.3686] 3. Use an approach similar to cross validation to estimate whether you can trust the 80% confidence interval.

```
ts_length <-  length(birth)
forecast_length <- 1
start <- 250
lower_bounds <- c()
upper_bounds<- c()
correct_num <- 0
wrong_num <- 0
#Correct_num means the number of birth[i] in the 80% confidence interval
for(i in start:(ts_length - forecast_length)){

  fitted_sarima<-  Arima(birth[0:i], order = c(4,1,0), seasonal = c(0,1,1))
  forecast_result <-  forecast(fitted_sarima, h = forecast_length)
  lower_bounds[i] <- forecast_result$lower[1]
  upper_bounds[i] <- forecast_result$upper[1]
  if (birth[i+1] > lower_bounds[i] & birth[i+1] < upper_bounds[i]){
    correct_num = correct_num + 1
  }else{
    wrong_num= wrong_num + 1
  }
}
```

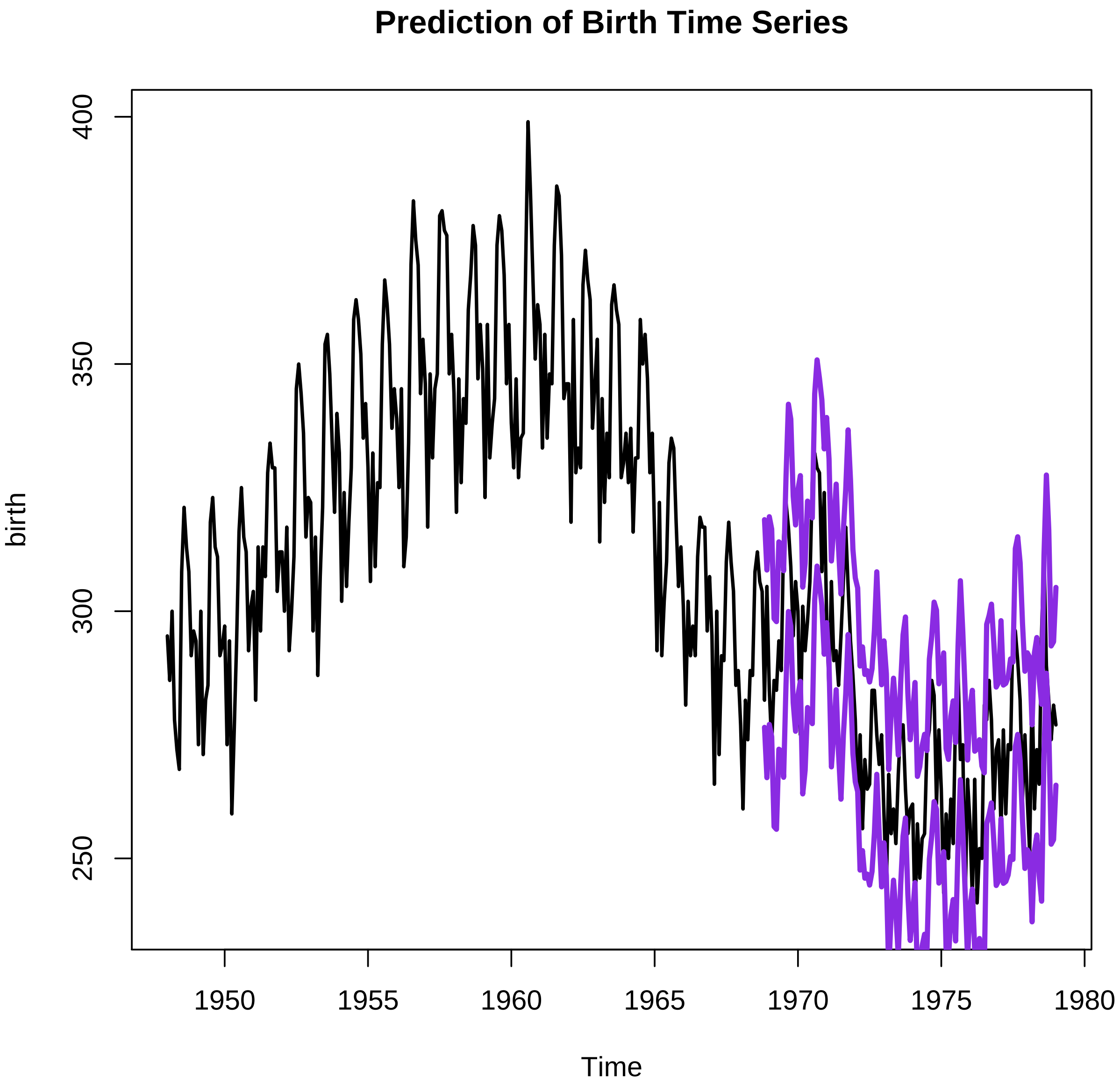
```
correct_num
```

```
## [1] 104
```

```
wrong_num
```

```
## [1] 19
```

```
#plot the bounds of confidence interval and the time series.
upper_bounds_ts<-ts(upper_bounds, start = c(1948,2), frequency = 12)
lower_bounds_ts<-ts(lower_bounds, start = c(1948,2), frequency = 12)

par(mfrow=c(1,1))
plot(birth, lwd =2, main="Prediction of Birth Time Series")
lines(upper_bounds_ts, lwd = 3, col = "blueviolet")
lines(lower_bounds_ts, lwd = 3, col = "blueviolet")
```



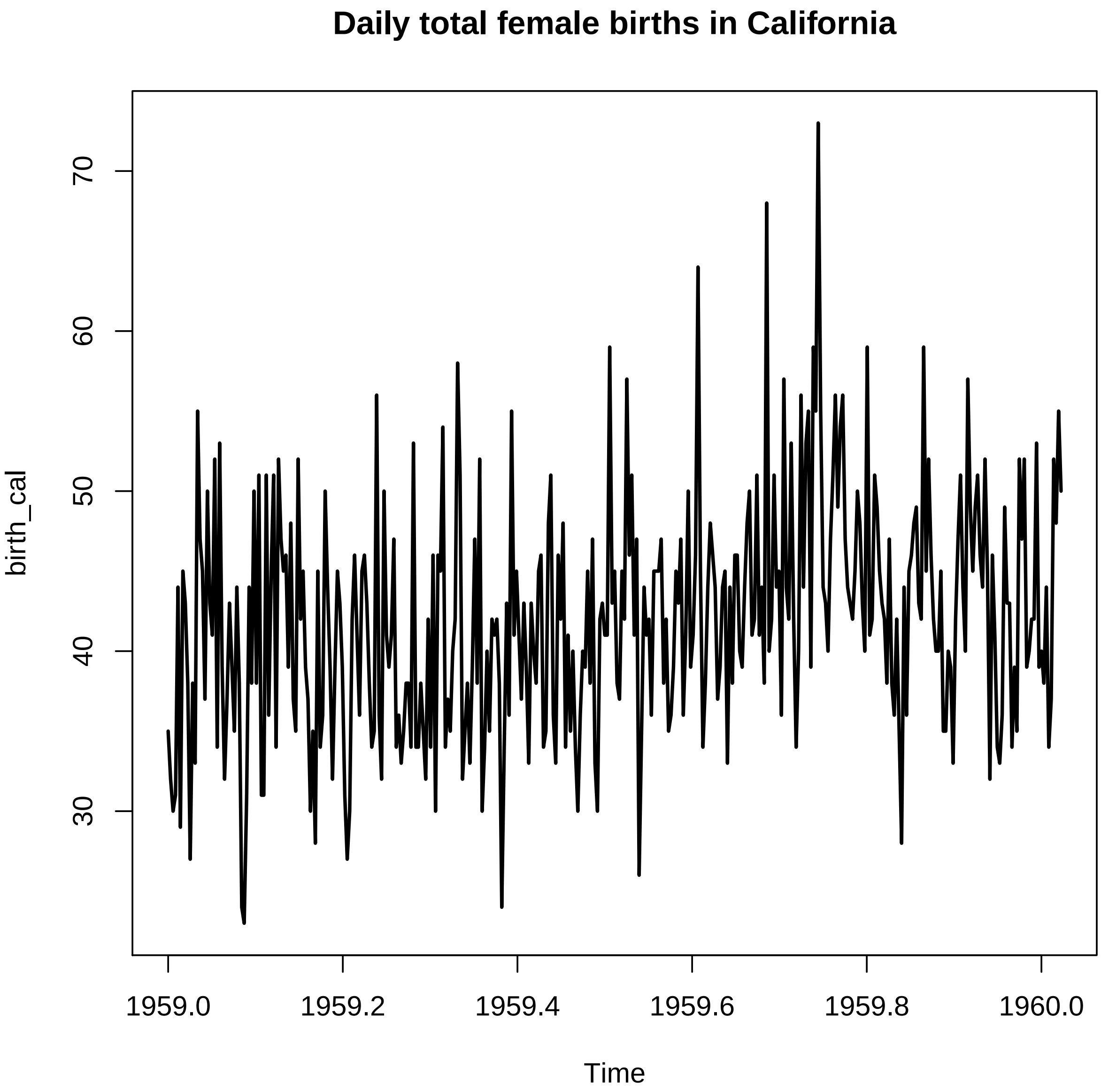
We did 123 forecasts to examine whether the true value lies in the 80% confidence interval of prediction, There are 104/123 in the confidence interval, and 19/123 of the forecasts doesn't. Also, from the plot, we can trust 80% confidence interval.

2 Exercise 2 (Number of Birth in California?)

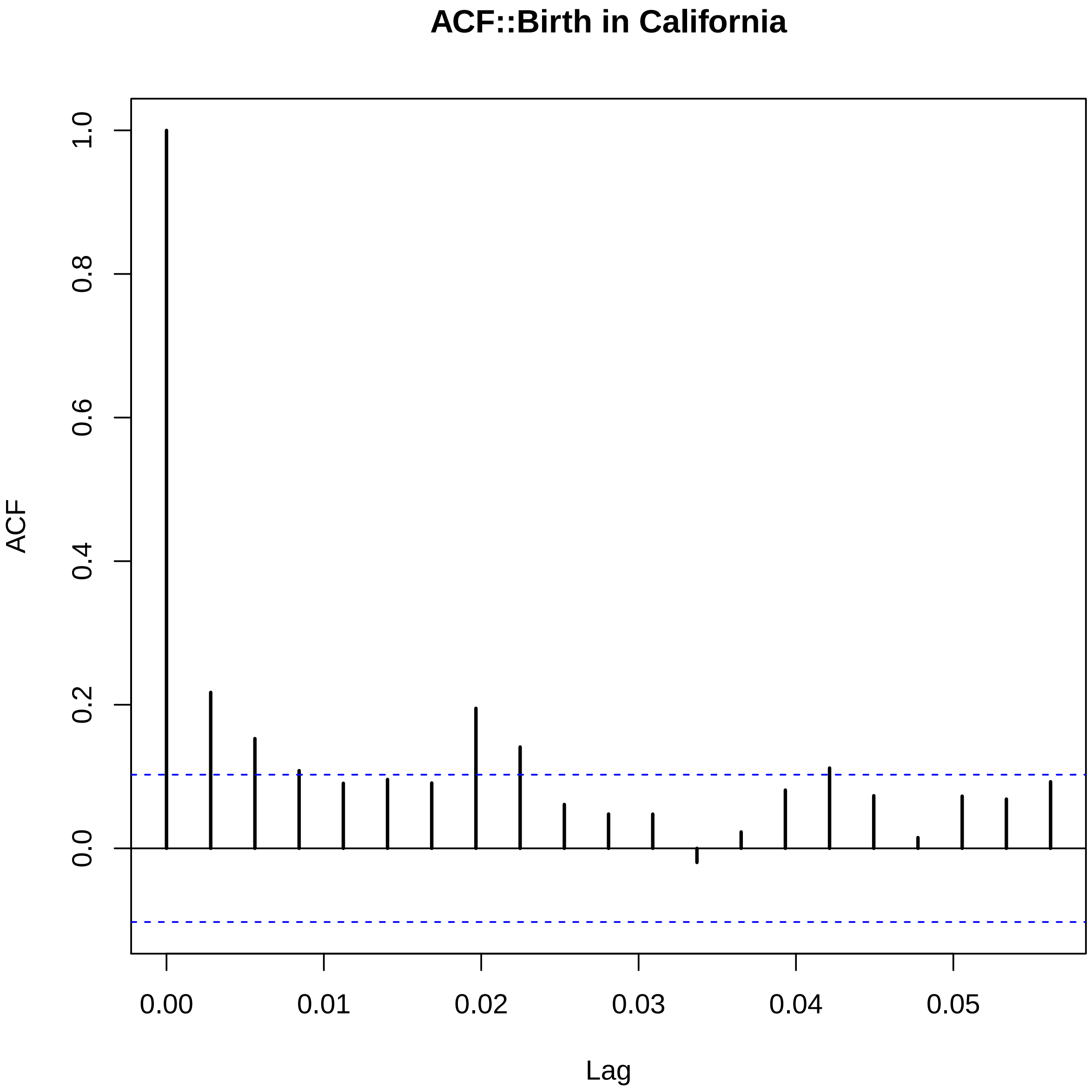
1. Load the data and plot.

```
birth_data <- read.csv("E:/ST3233/Assignment2/Datasets/daily-total-female-births-in-cal.csv",
                       header= TRUE,sep=",")
birth_cal <- ts(birth_data$Daily.total.female.births.in.California,
                frequency = 356,start = c(1959))
```

```
par(mfrow=c(1,1))
plot(birth_cal,lwd = 2, main = "Daily total female births in California")
```

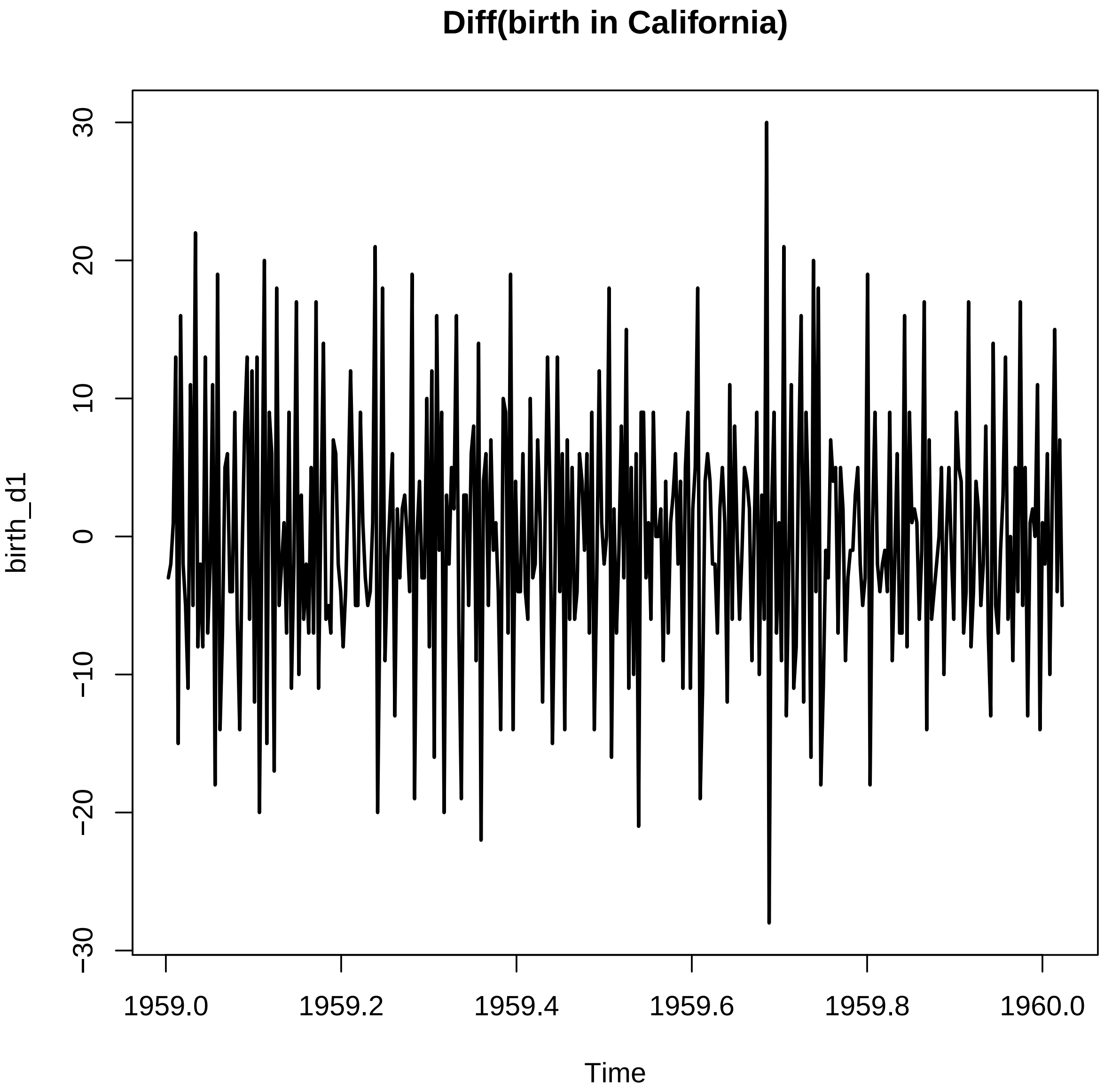


```
#Plot ACF to see whether the time series is stationary or not.
acf(birth_cal,lwd = 2, main ="ACF::Birth in California",lag.max = 20)
```

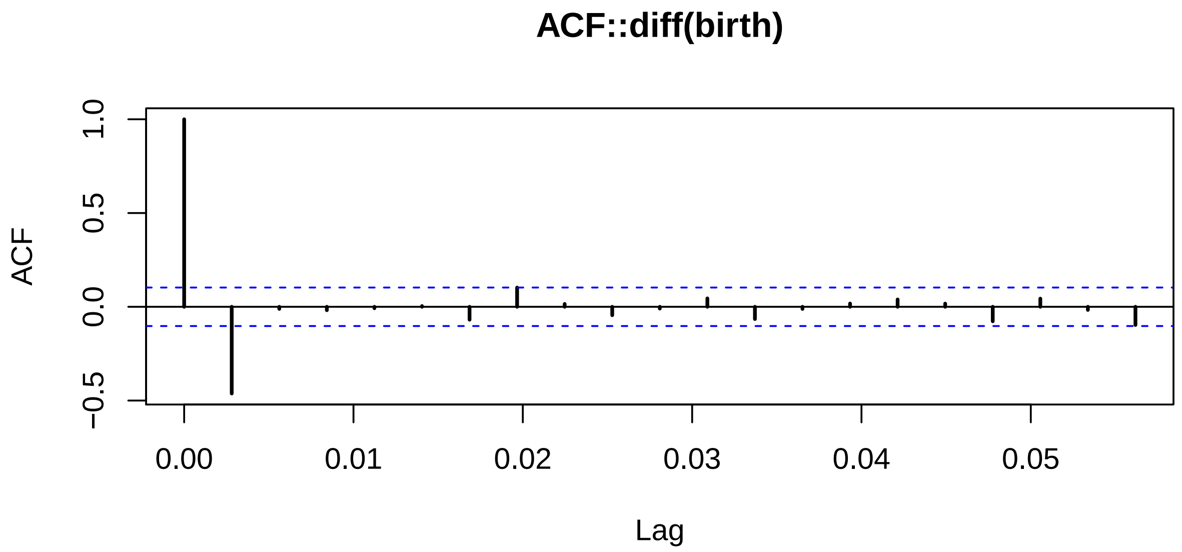


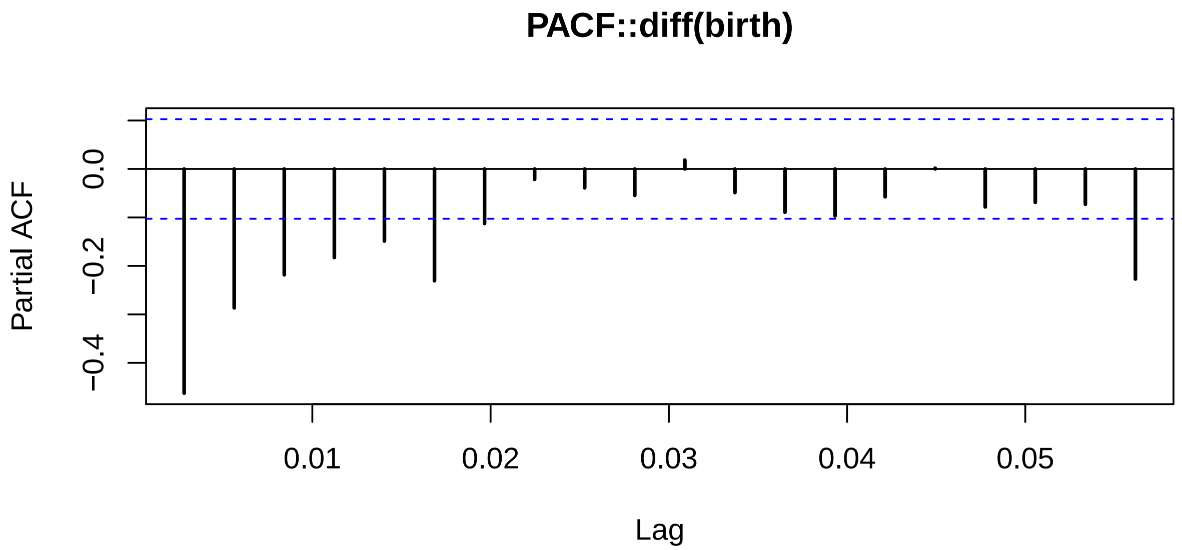
From the plot, the time series is not stationary, thus we cannot use ARMA model. Besides, there is no seasonal component in this time series, so we first choose ARIMA model. 2. Fit an ARIMA model.

```
#Consider a new time series: diff(birth_cal)
birth_d1 = diff(birth_cal)
plot(birth_d1,lwd = 2, main = "Diff(birth in California)")
```



```
#Diff(birth_cal) is stationary, and then consider the acf and pacf.

par(mfrow=c(2,1))
acf(birth_d1,lwd = 2, main =  "ACF::diff(birth)",lag.max = 20)
pacf(birth_d1,lwd = 2, main =  "PACF::diff(birth)",lag.max = 20)
```





From acf plot ,q $\leq 1, andfrompartial - acfplot, p \leq 6withd = 1$

```
AIC_best <- 10**6
for (p in 0:6){
  for (q in 0:1){
    fit_arima <- Arima(birth_cal, order = c(p,1,q))
    if (fit_arima$aic < AIC_best){
      AIC_best <- fit_arima$aic
      cat("p = ",p,",d = 1, q = ",q,", AIC = ",AIC_best,"\n")
    }
  }
}
```

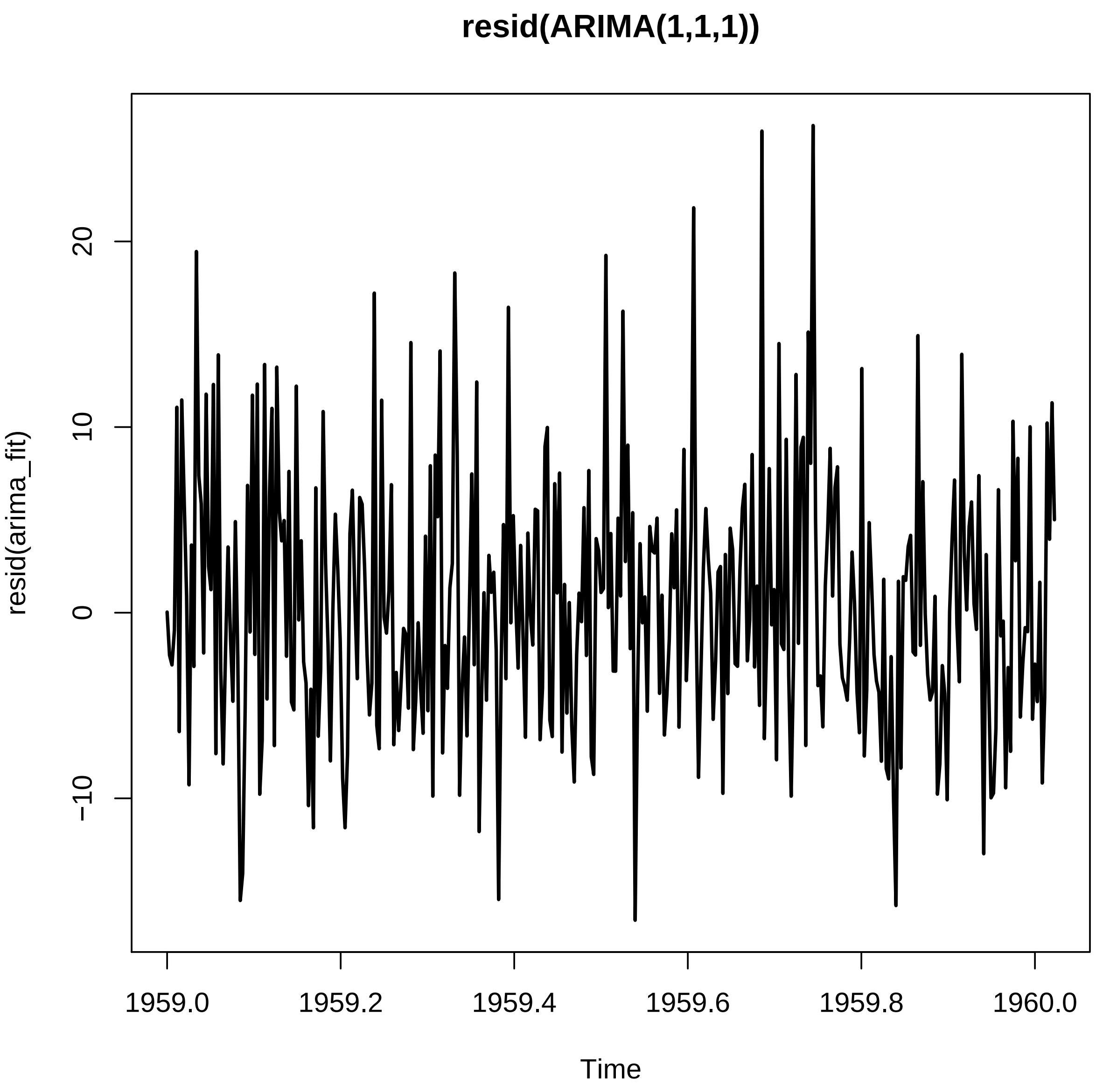
```
## p =  0 ,d = 1, q =  0 , AIC =  2648.768
## p =  0 ,d = 1, q =  1 , AIC =  2462.221
## p =  1 ,d = 1, q =  1 , AIC =  2459.074
```

ARIMA(1,1,1) has the lowest AIC, so we choose ARIMA(1,1,1)

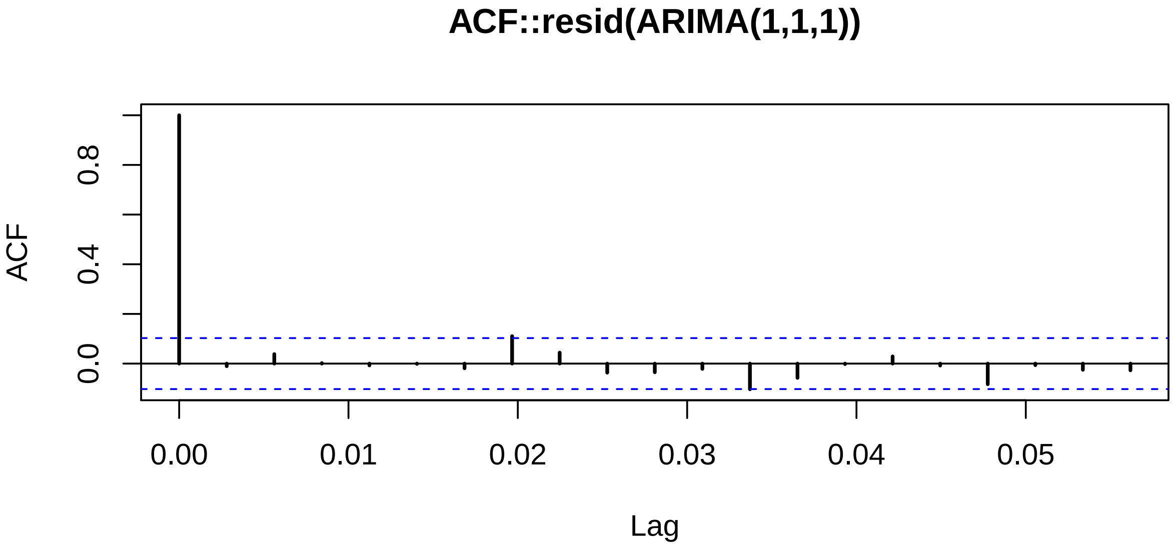
```
arima_fit <- Arima(birth_cal,order = c(1,1,1))
```

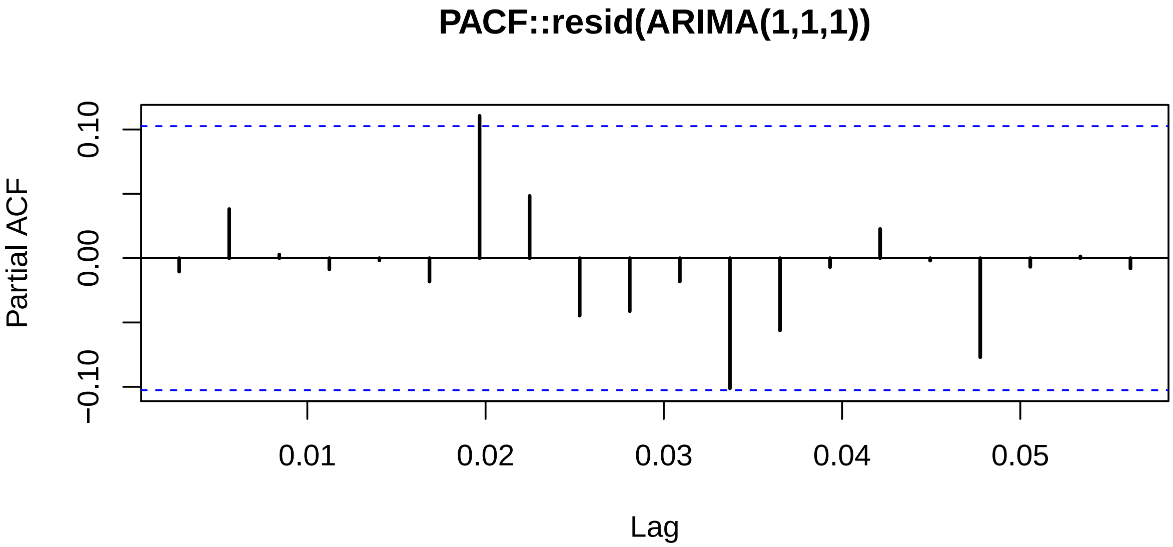
3. Examining the normality of the residuals to test the ARIMA(1,1,1) model

```
par(mfrow=c(1,1))
plot(resid(arima_fit),lwd=2, main="resid(ARIMA(1,1,1))")
```

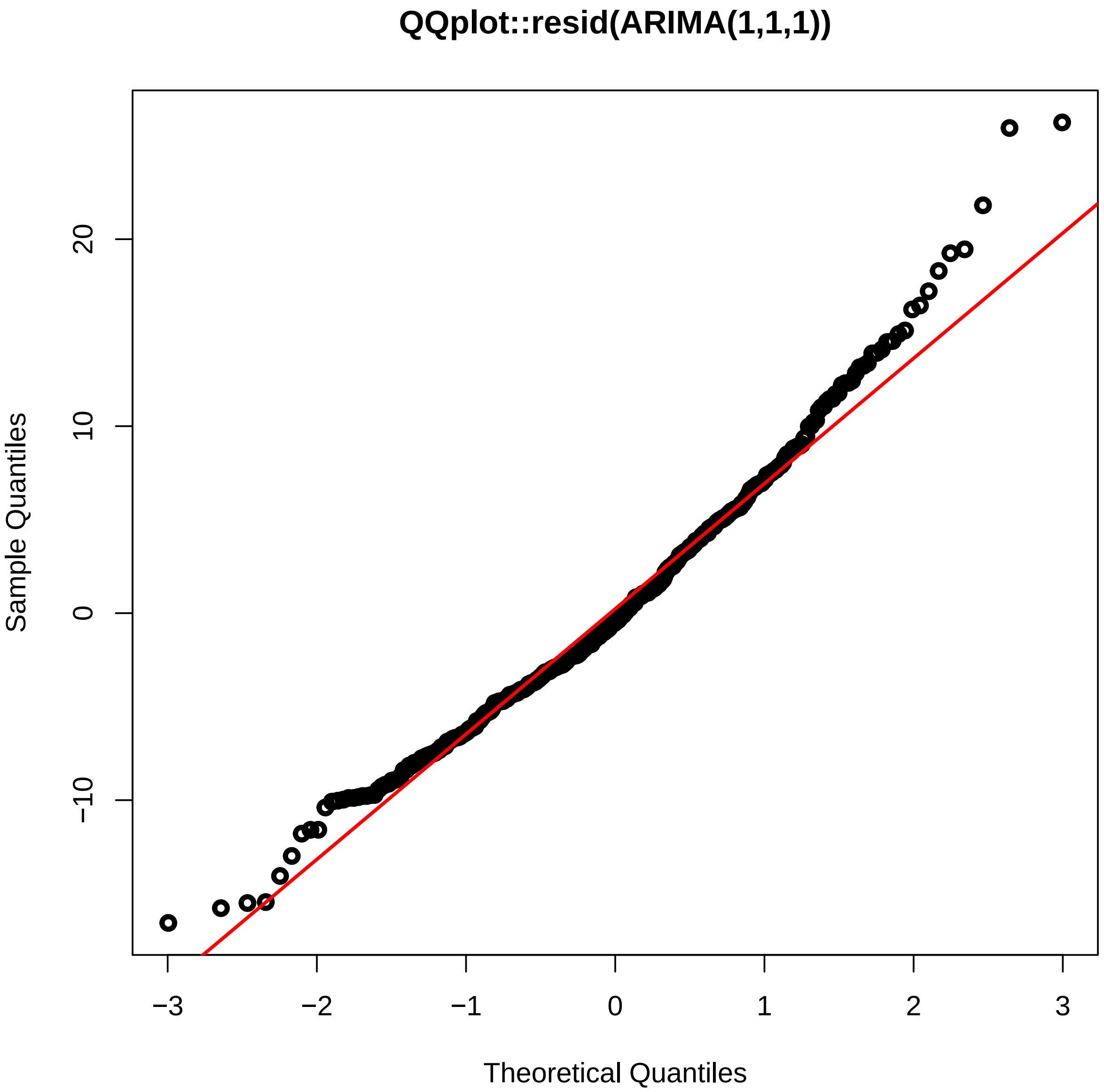


```
par(mfrow=c(2,1))
acf(resid(arima_fit),lwd=2, main="ACF::resid(ARIMA(1,1,1))",lag.max = 20)
pacf(resid(arima_fit),lwd=2, main="PACF::resid(ARIMA(1,1,1))",lag.max = 20)
```



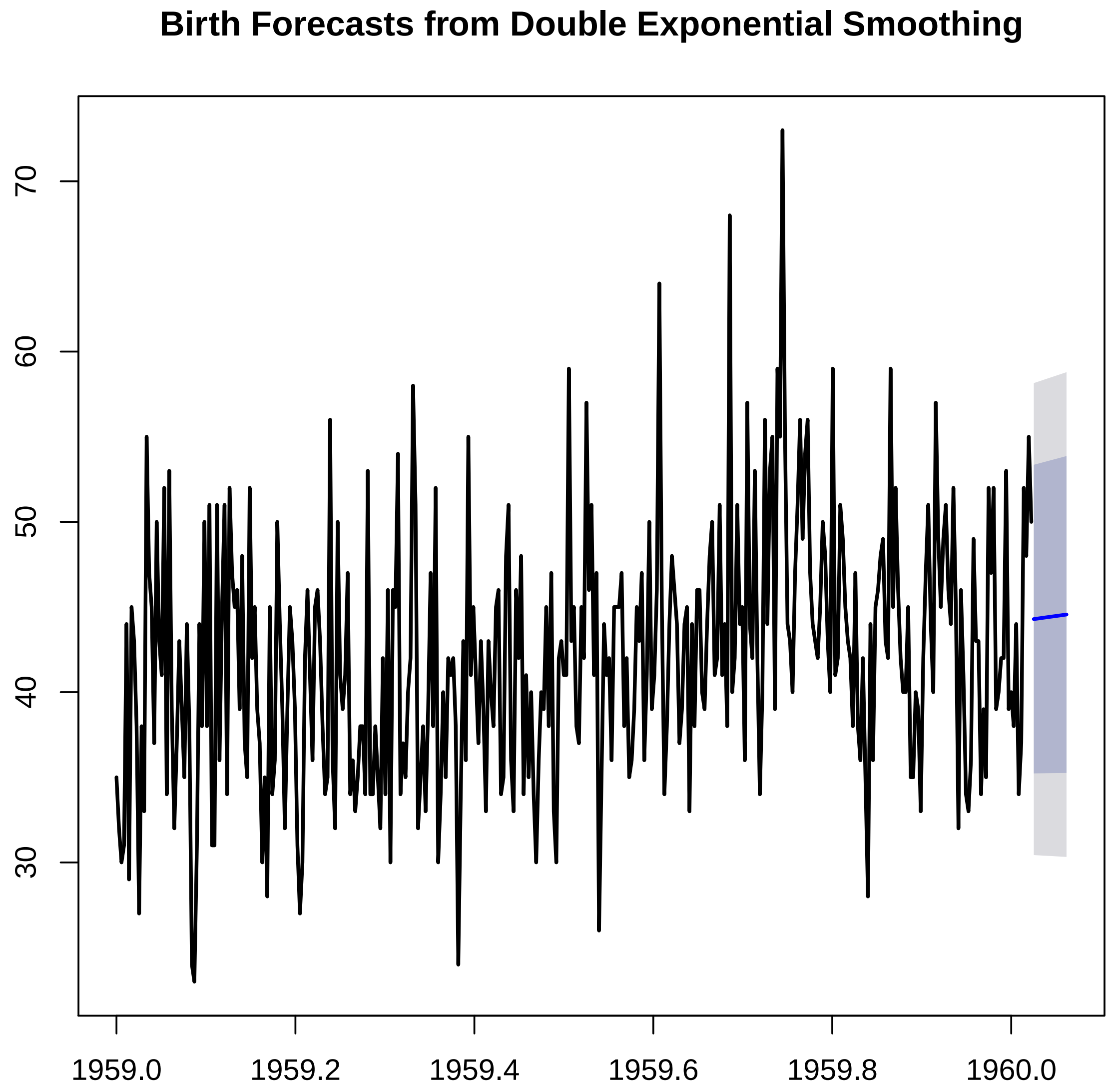


```
par(mfrow=c(1,1))
qqnorm(resid(arima_fit), main="QQplot::resid(ARIMA(1,1,1))", lwd=3)
qqline(resid(arima_fit), lwd=2, col="red")
```



Thus, the residuals are follows a Gaussian Distribution. 4. Another model is Double exponential smooting

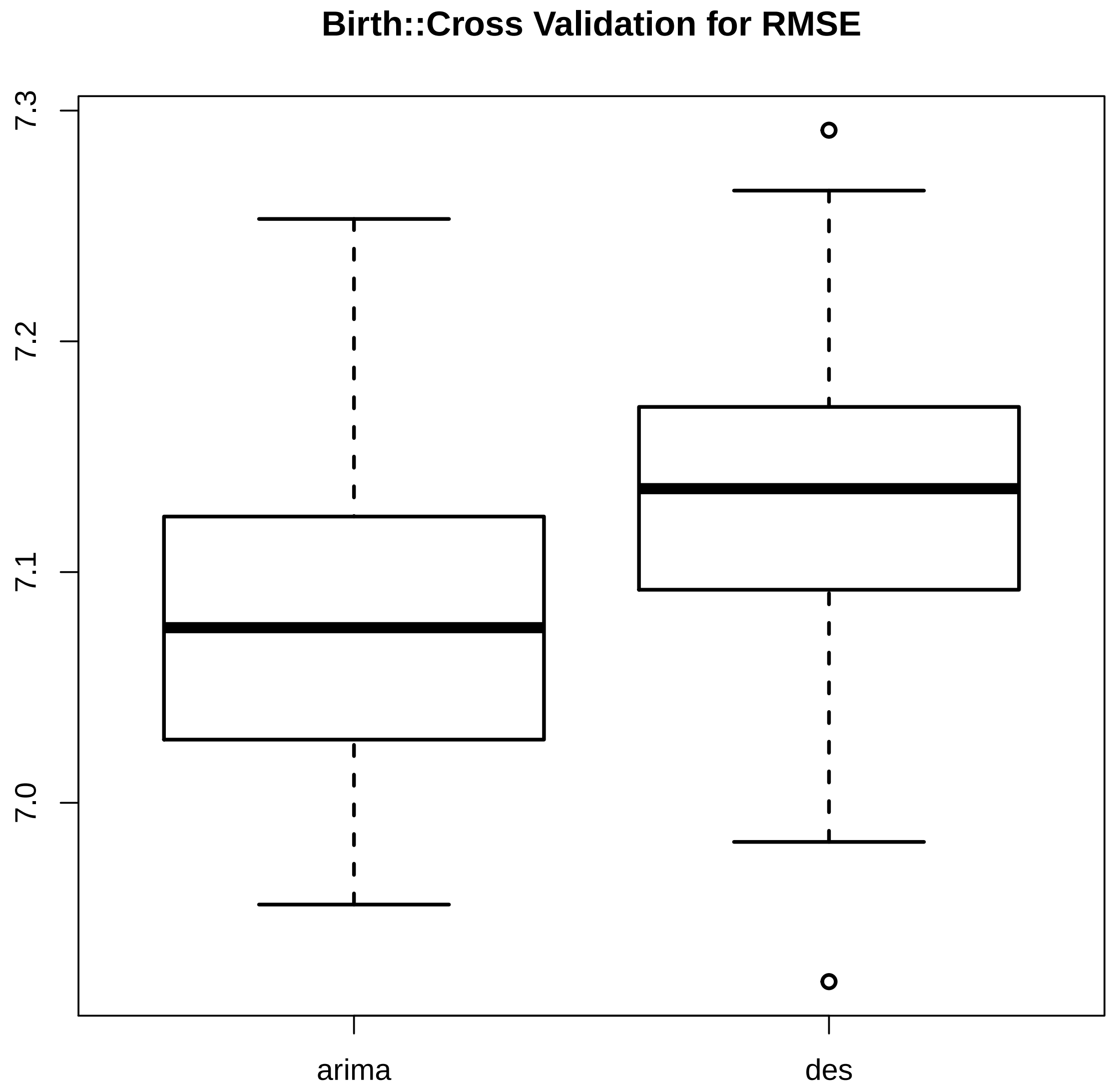
```
DES_fit <- holt(birth_cal, initial = "optimal", h =2*7)
DES_fit

##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 1960.025       44.29105 35.22863 53.35346 30.43128 58.15081
## 1960.028       44.31156 35.23049 53.39262 30.42326 58.19985
## 1960.031       44.33206 35.23232 53.43181 30.41521 58.24892
## 1960.034       44.35257 35.23414 53.47101 30.40713 58.29802
## 1960.037       44.37308 35.23593 53.51024 30.39902 58.34715
## 1960.039       44.39359 35.23771 53.54948 30.39088 58.39631
## 1960.042       44.41410 35.23947 53.58874 30.38270 58.44550
## 1960.045       44.43461 35.24120 53.62803 30.37450 58.49472
## 1960.048       44.45512 35.24292 53.66733 30.36627 58.54397
## 1960.051       44.47563 35.24462 53.70665 30.35801 58.59325
## 1960.053       44.49614 35.24630 53.74599 30.34972 58.64256
## 1960.056       44.51665 35.24795 53.78535 30.34140 58.69190
## 1960.059       44.53716 35.24959 53.82473 30.33305 58.74127
## 1960.062       44.55767 35.25121 53.86413 30.32467 58.79067

plot(DES_fit,main = "Birth Forecasts from Double Exponential Smoothing", lwd = 2)
```



5. Compare ARIMA(1,1,1) and DES by using cross-validation

```
#Define a function CV to do cross-validation
CV <-  function(time_series, start, forecast_length,ts_model){
  ts_length <-  length(time_series)
  accuracy_list = c()
  for(k in start:(ts_length - forecast_length)){
    fitted_model <- ts_model(ts(time_series[0:k]))
    RMSE <-  accuracy(forecast(fitted_model, h = forecast_length))[2]
    accuracy_list = c(accuracy_list, RMSE)
  }
  return(accuracy_list)
```

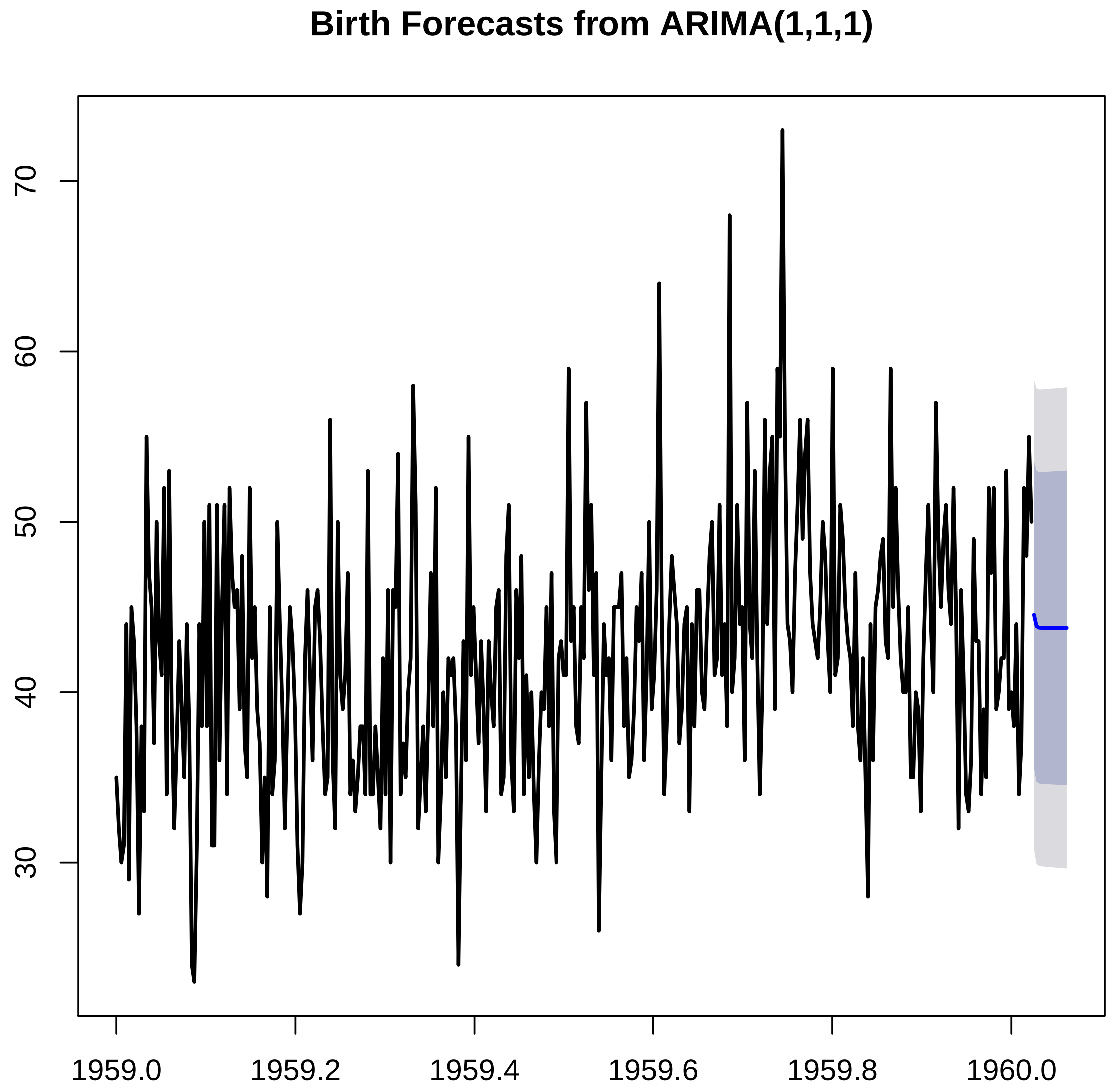
```r
}

#Define two models
model_ARIMA <- function(ts) return(Arima(ts,order = c(1,1,1)))
model_DES <- function(ts) return(holt(ts,initial = "optimal"))

start <- 250
forecast_length <- 7
CV_birth_Cal <- data.frame(
  arima = CV(birth_cal, start, forecast_length, model_ARIMA),
  des = CV(birth_cal, start, forecast_length, model_DES)
)

boxplot(CV_birth_Cal,main = "Birth::Cross Validation for RMSE", lwd=2)
```



From boxplot, ARIMA(1,1,1) has a lower RMSE, which is a better model. That's why we choose ARIMA(1,1,1) 6. Forecast the number of birth during the two weeks by using ARIMA(1,1,1)

```
arima_forecast <- forecast(arima_fit, h = 2*7)
arima_forecast

##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 1960.025       44.55357 35.54440 53.56275 30.77523 58.33191
## 1960.028       43.87142 34.74366 52.99918 29.91172 57.83113
## 1960.031       43.78599 34.64332 52.92865 29.80348 57.76849
## 1960.034       43.77528 34.62372 52.92685 29.77917 57.77140
## 1960.037       43.77394 34.61412 52.93376 29.76521 57.78268
## 1960.039       43.77378 34.60579 52.94176 29.75255 57.79500
## 1960.042       43.77376 34.59762 52.94989 29.74007 57.80744
## 1960.045       43.77375 34.58948 52.95803 29.72761 57.81989
## 1960.048       43.77375 34.58134 52.96616 29.71517 57.83233
## 1960.051       43.77375 34.57322 52.97429 29.70275 57.84476
## 1960.053       43.77375 34.56510 52.98241 29.69033 57.85718
## 1960.056       43.77375 34.55699 52.99052 29.67792 57.86958
## 1960.059       43.77375 34.54888 52.99862 29.66553 57.88198
## 1960.062       43.77375 34.54078 53.00672 29.65314 57.89436
```

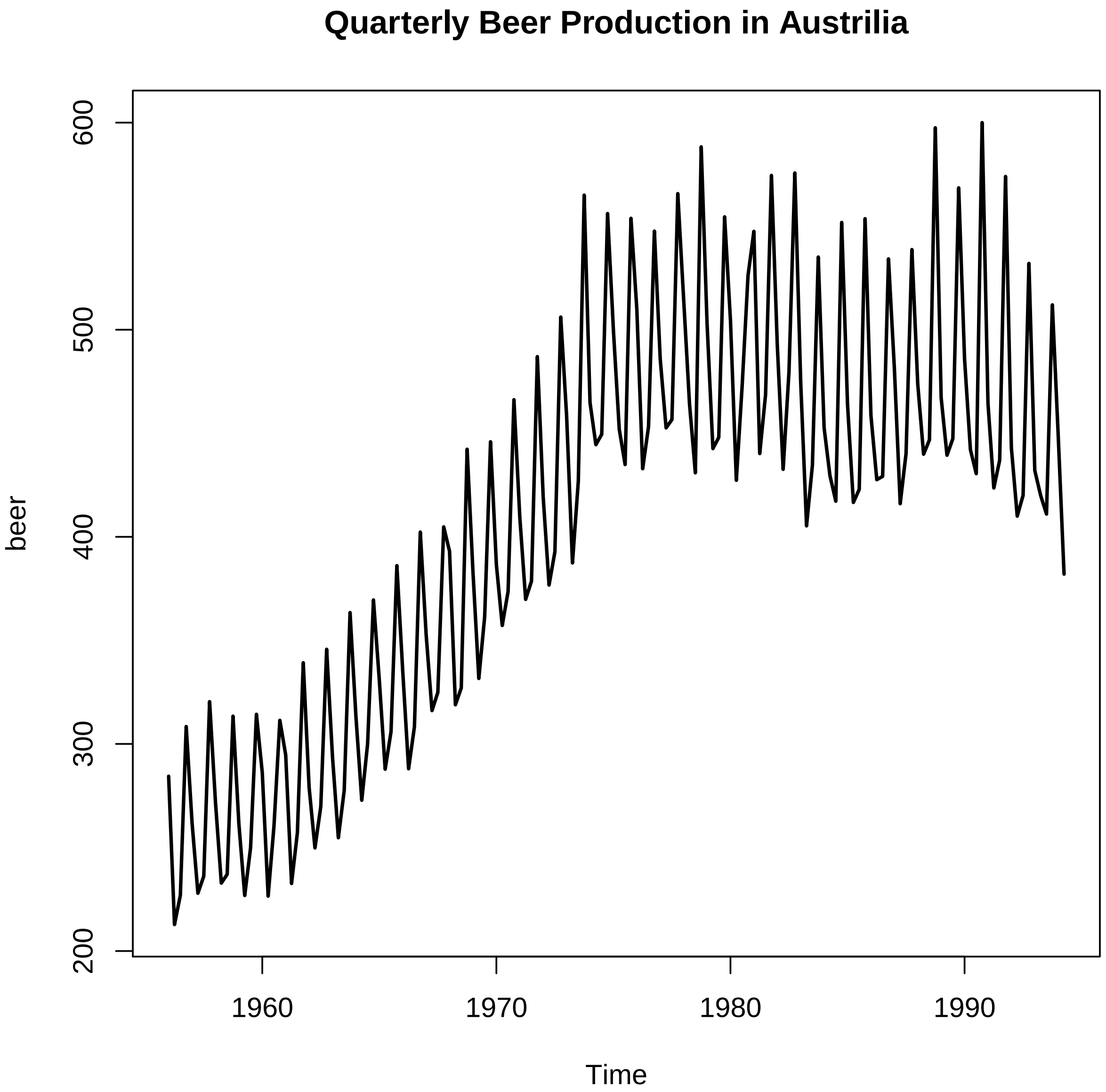
```
plot(arima_forecast,main = "Birth Forecasts from ARIMA(1,1,1)",lwd = 2)
```



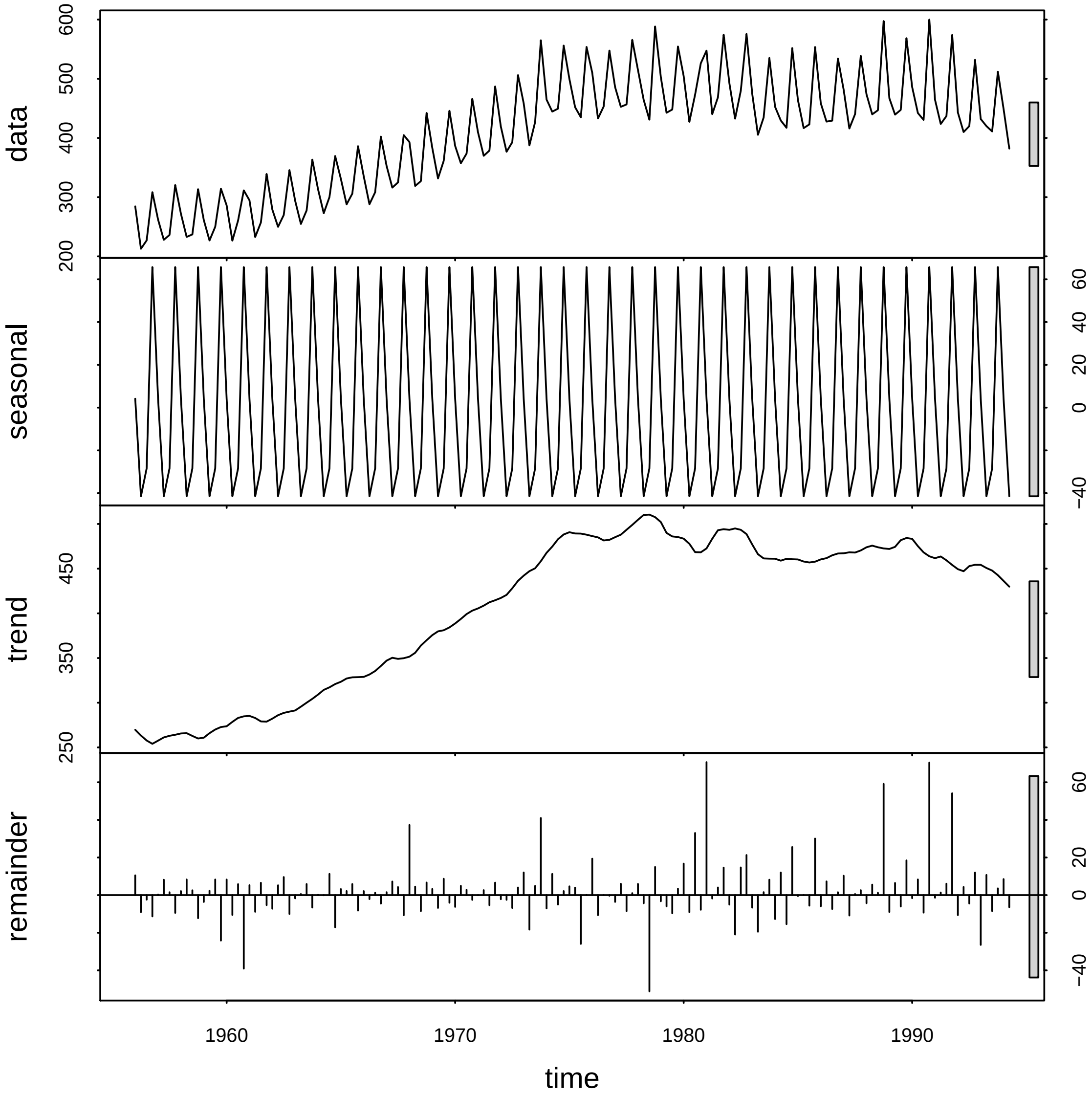
3 Exercise 3 (How much beer?)

1. Load the data and plot.

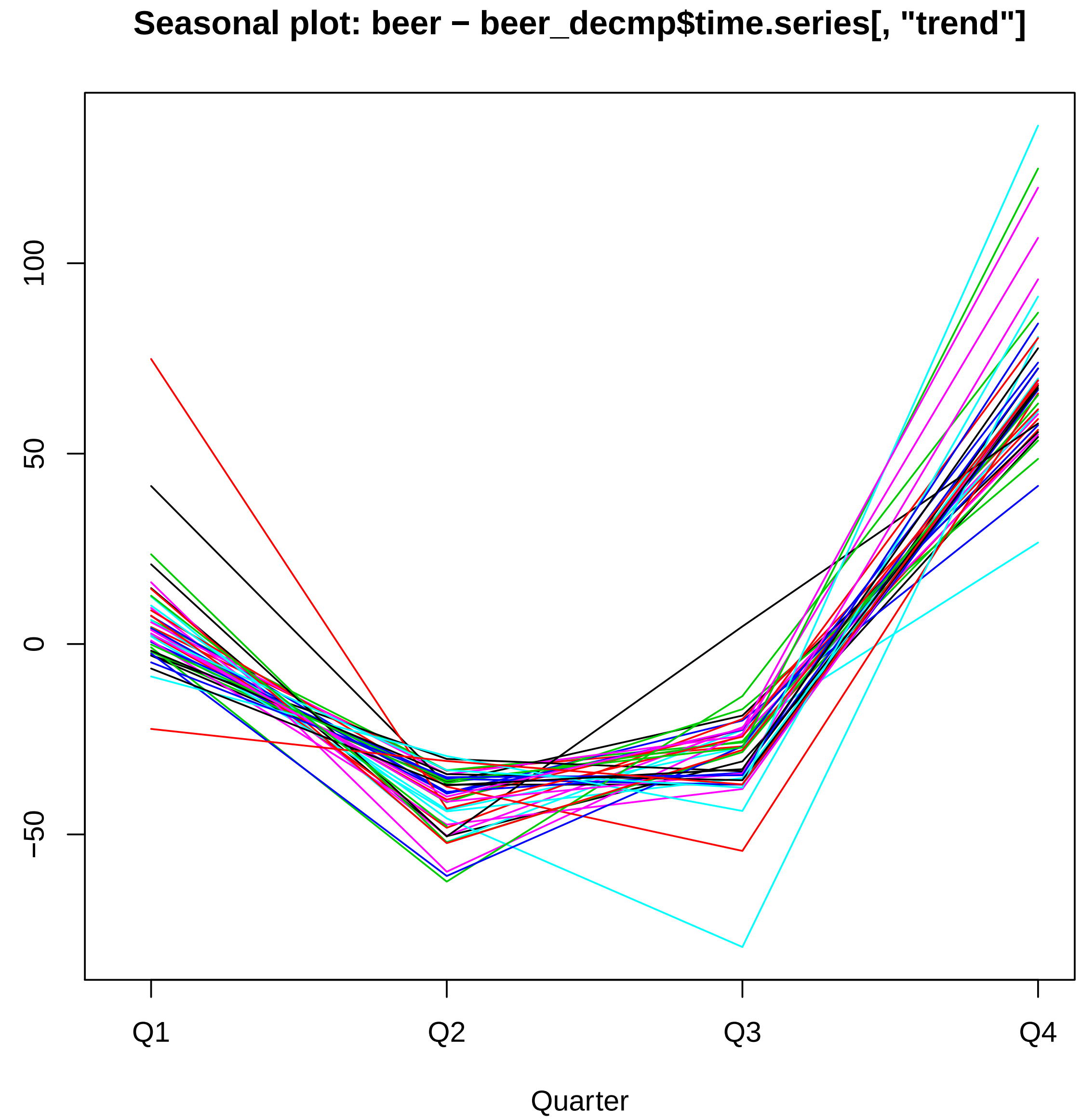
```
beer_data <- read.csv("E:/ST3233/Assignment2/Datasets/quarterly-beer-production-in-aus.csv",
                      header= TRUE,sep=",")
beer <- ts(beer_data$beer_production,frequency = 4,start=c(1956))
par(mfrow=c(1,1))
plot(beer,lwd = 2, main = "Quarterly Beer Production in Austrilia")
```



```
#From the plot, we can clearly see that there is periodicity and trend, thus we decompose it.
beer_decmp <-stl(beer,s.window = "periodic", robust = T)
plot(beer_decmp)
```

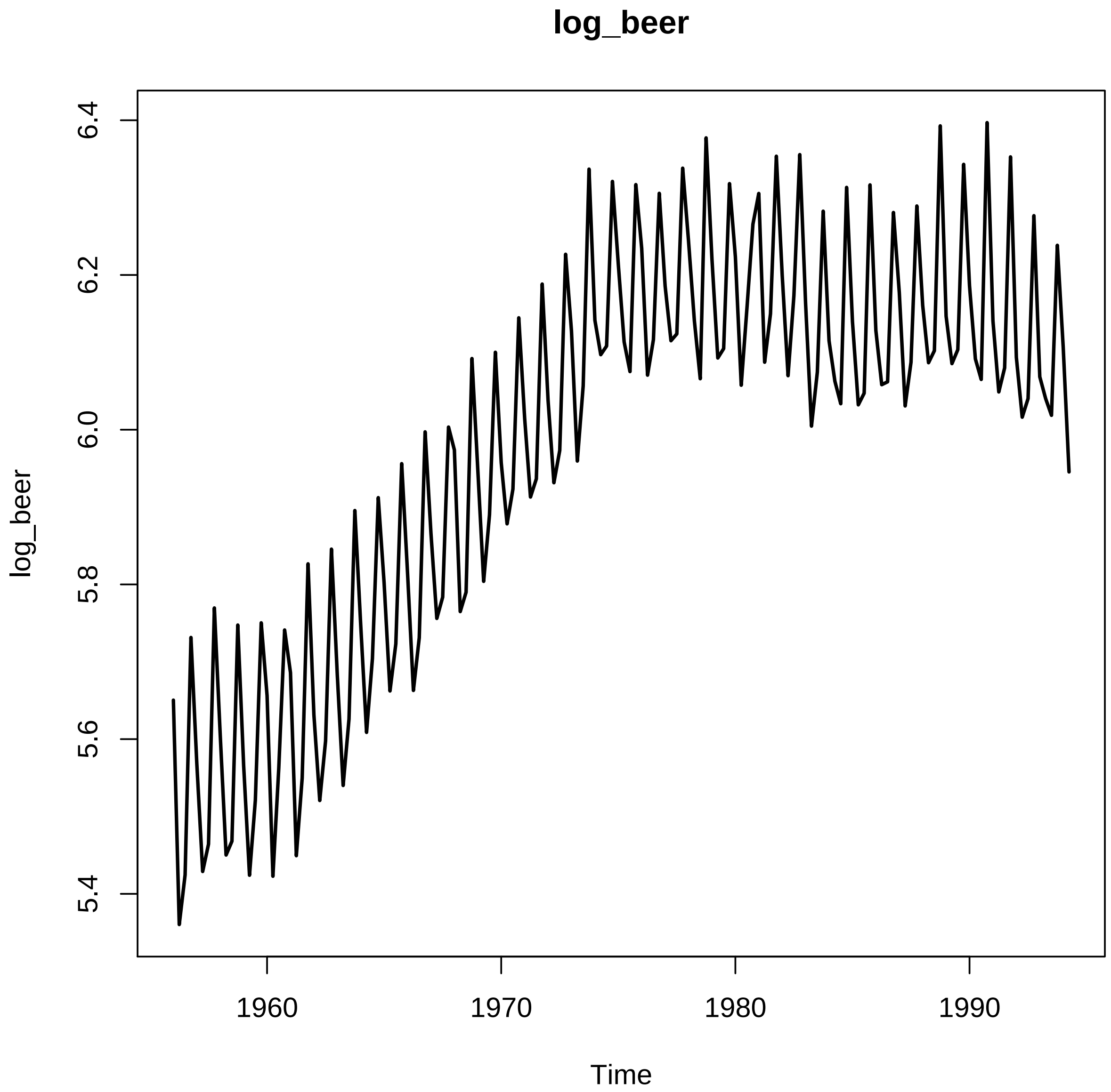


```
#seasonal plot
seasonplot(beer - beer_decmp$time.series[,"trend"], s = 4,col = 1:6, type = "l")
```

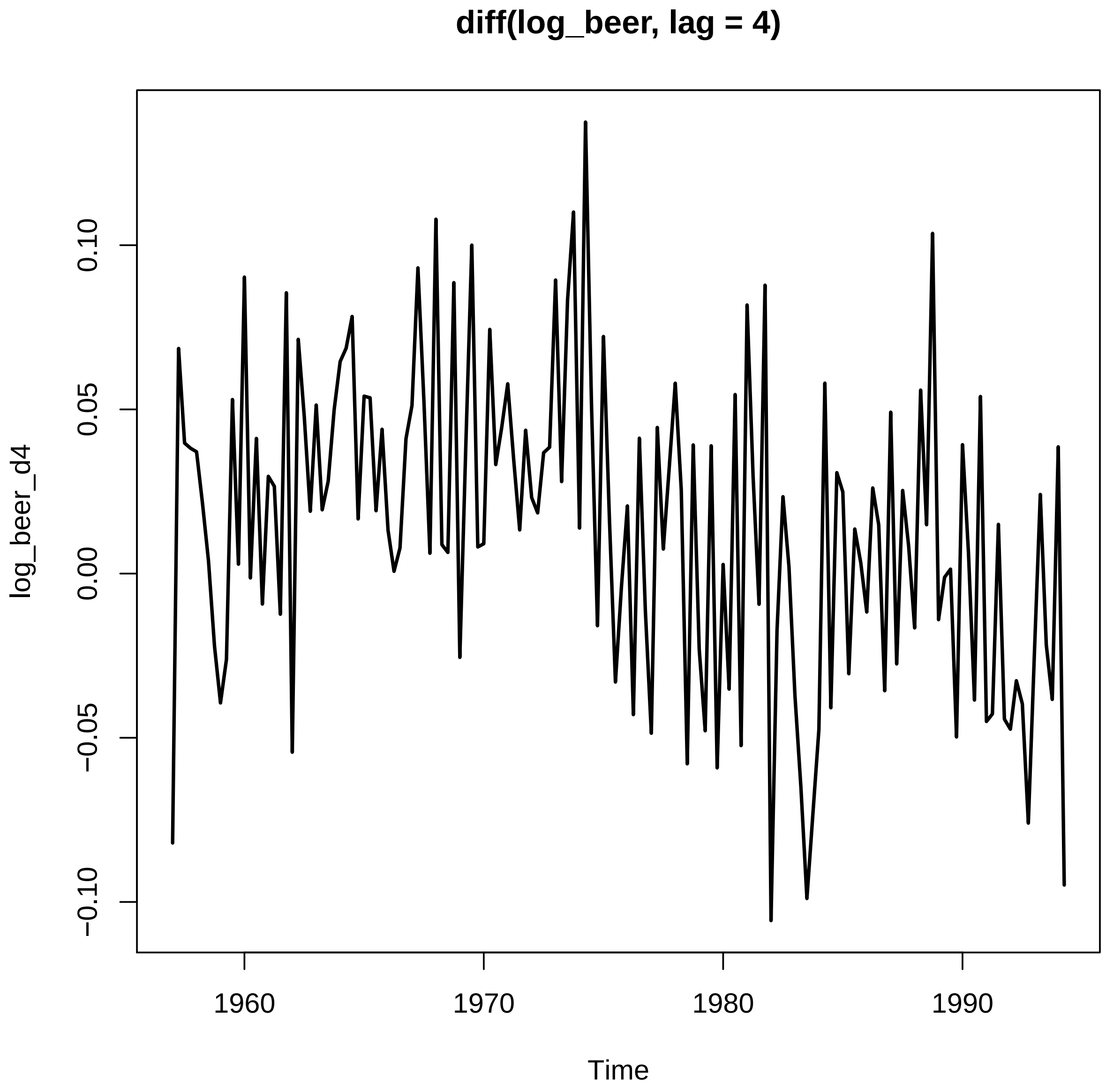


There is seasonal behavior, thus we first use SARIMA model 2. Fit the model. Notice that the fluctation of the time series becomes larger as the time changes, so we consider a new time series: log(beer)

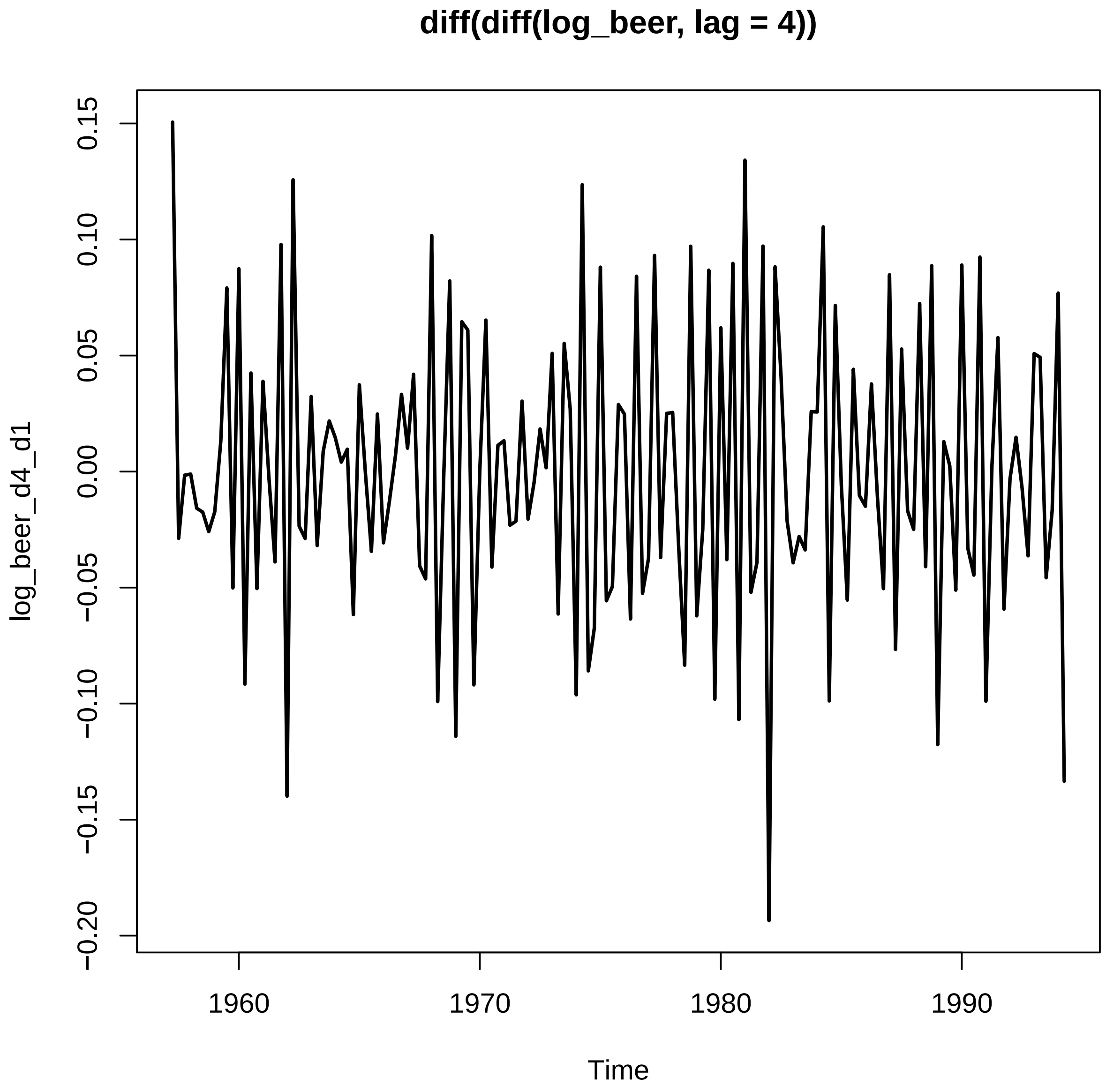
```
log_beer = log(beer)
plot(log_beer, lwd=2, main="log_beer")
```



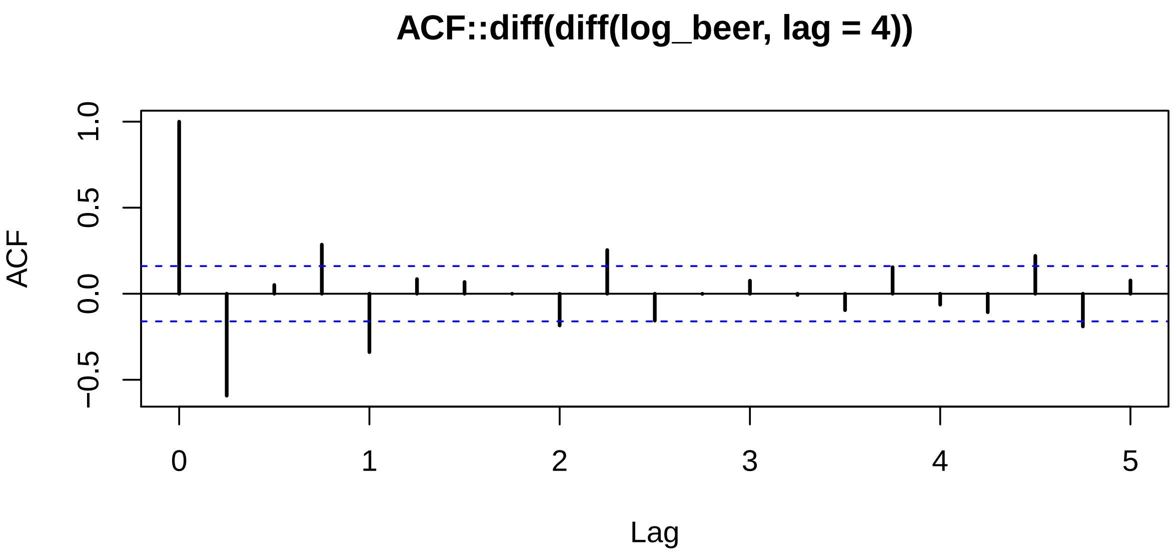
```
log_beer_d4 = diff(log_beer, lag = 4)
plot(log_beer_d4,lwd = 2, main = "diff(log_beer, lag = 4)")
```

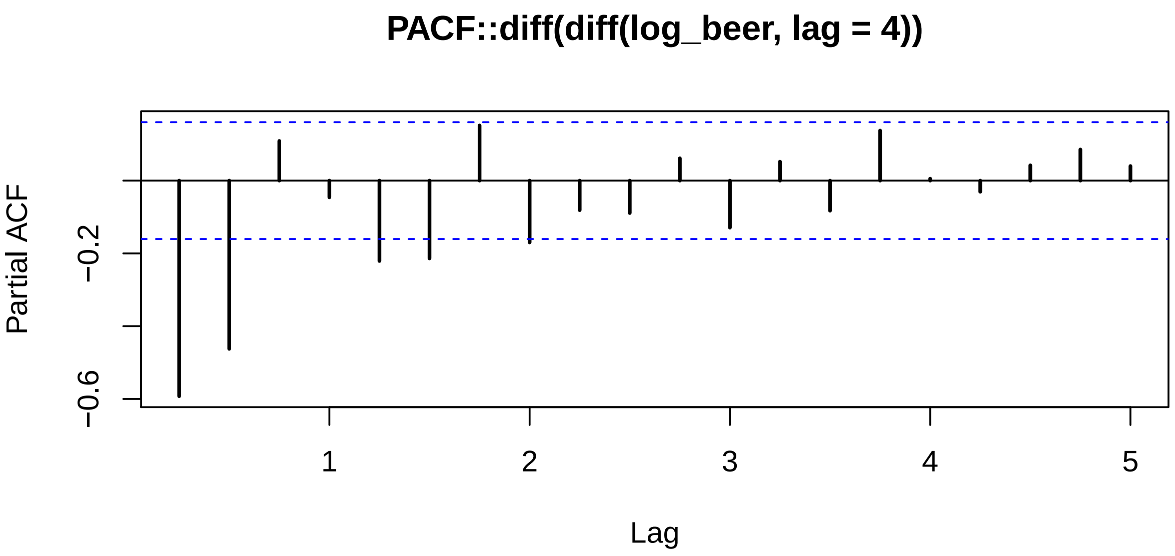


```
log_beer_d4_d1 = diff(log_beer_d4, lag = 1)
plot(log_beer_d4_d1,lwd = 2, main = "diff(diff(log_beer, lag = 4))")
```



```
par(mfrow=c(2,1))
acf(log_beer_d4_d1,lwd = 2, main = "ACF::diff(diff(log_beer, lag = 4))",lag.max = 20)
pacf(log_beer_d4_d1,lwd = 2, main = "PACF::diff(diff(log_beer, lag = 4))",lag.max = 20)
```





From acf plot ,q $\leq 4, andfrompartial - acfplot, p \leq 2withd = D = 1andP \leq 1, Q \leq 1$

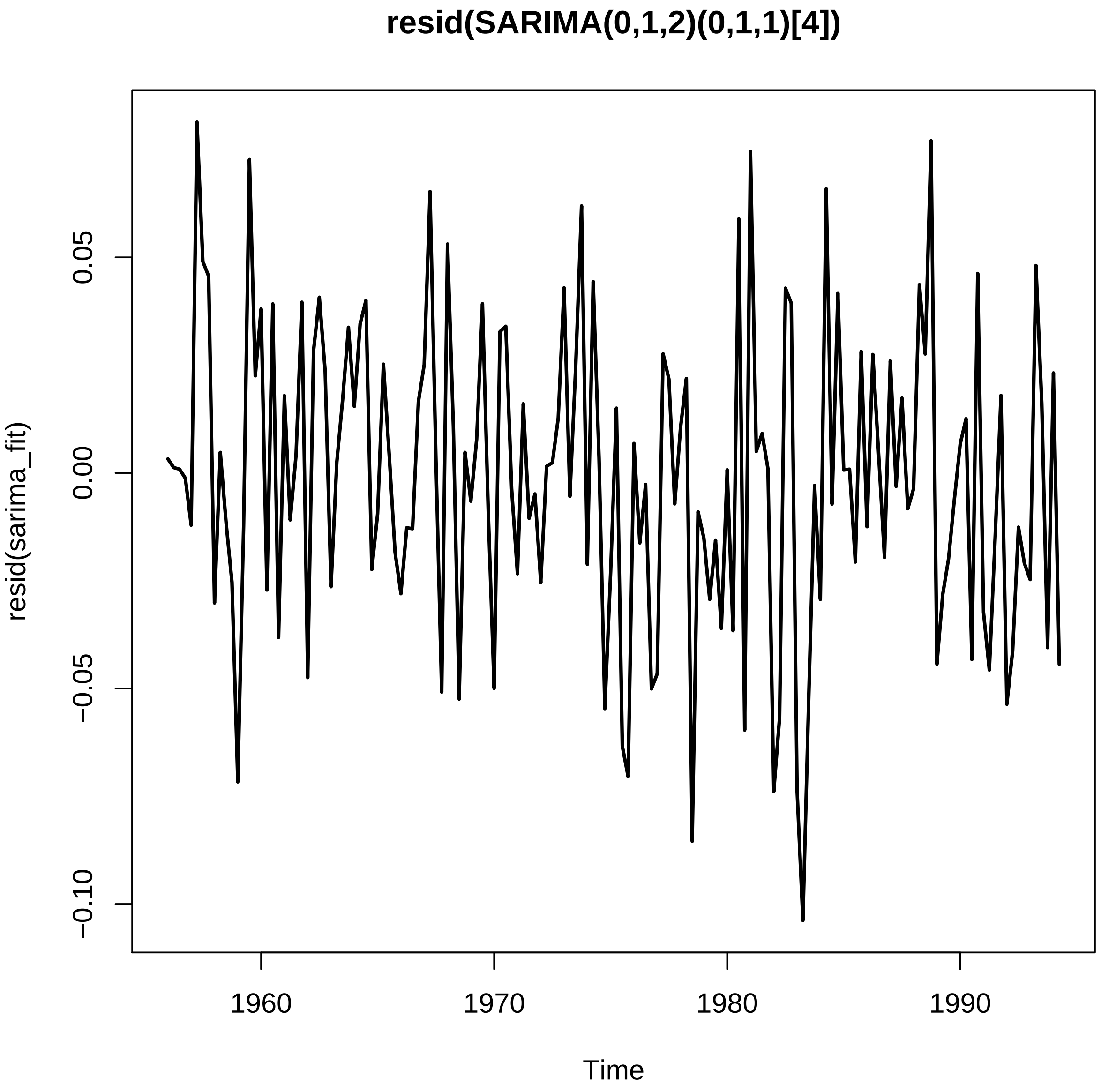
```
AIC_best <- 10**6
for (p in 0:2){
  for (q in 0:4){
    for (P in 0:1){
      for (Q in 0:1){
        fit_sarima <- Arima(log_beer, order = c(p,1,q), seasonal = c(P,1,Q))
        if (fit_sarima$aic < AIC_best){
          AIC_best <- fit_sarima$aic
          cat("p = ",p,", q = ",q,",P = ",P,",Q = ",Q,"\t AIC = ",AIC_best,"\n")
        }
      }
    }
  }
}

## p =  0 , q =  0 ,P =  0 ,Q =  0   AIC =  -403.1997
## p =  0 , q =  0 ,P =  0 ,Q =  1   AIC =  -456.3742
## p =  0 , q =  1 ,P =  0 ,Q =  0   AIC =  -505.6392
## p =  0 , q =  1 ,P =  0 ,Q =  1   AIC =  -537.6958
## p =  0 , q =  2 ,P =  0 ,Q =  1   AIC =  -556.2169
```

The lowest AIC gives the best fitted model of $\log_beer, whichisSARIMA(0, 1, 2)(0, 1, 1)[4]$

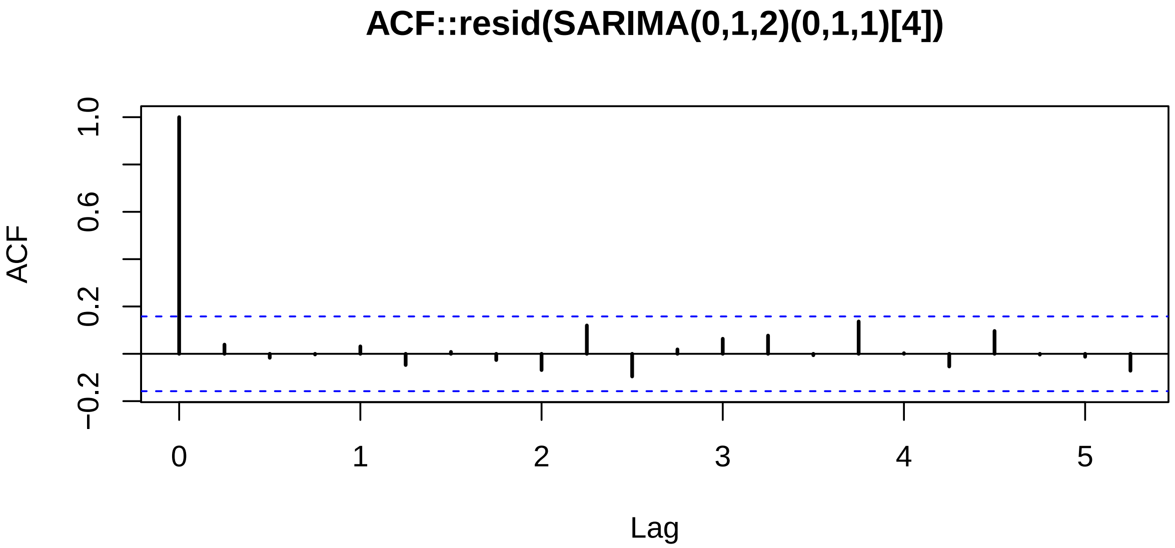
```
sarima_fit <- Arima(log_beer, order = c(0,1,2), seasonal = c(0,1,1))
```

3. Then consider the residuals of the SARIMA model.

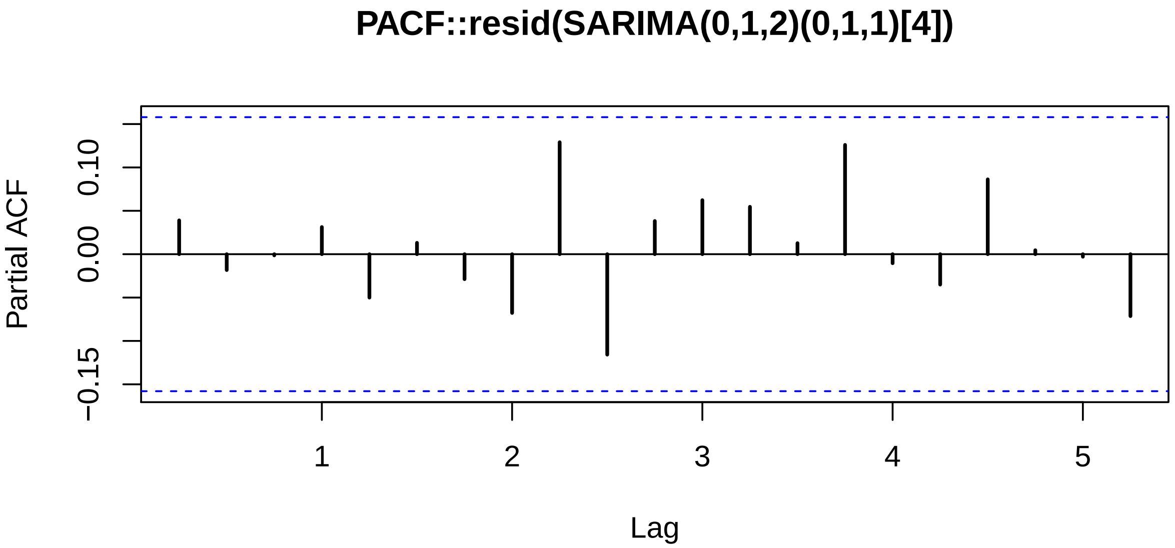
```
par(mfrow=c(1,1))
plot(resid(sarima_fit),lwd=2, main="resid(SARIMA(0,1,2)(0,1,1)[4])")
```



```
par(mfrow=c(2,1))
acf(resid(sarima_fit),lwd=2, main="ACF::resid(SARIMA(0,1,2)(0,1,1)[4])")
pacf(resid(sarima_fit),lwd=2, main="PACF::resid(SARIMA(0,1,2)(0,1,1)[4])")
```

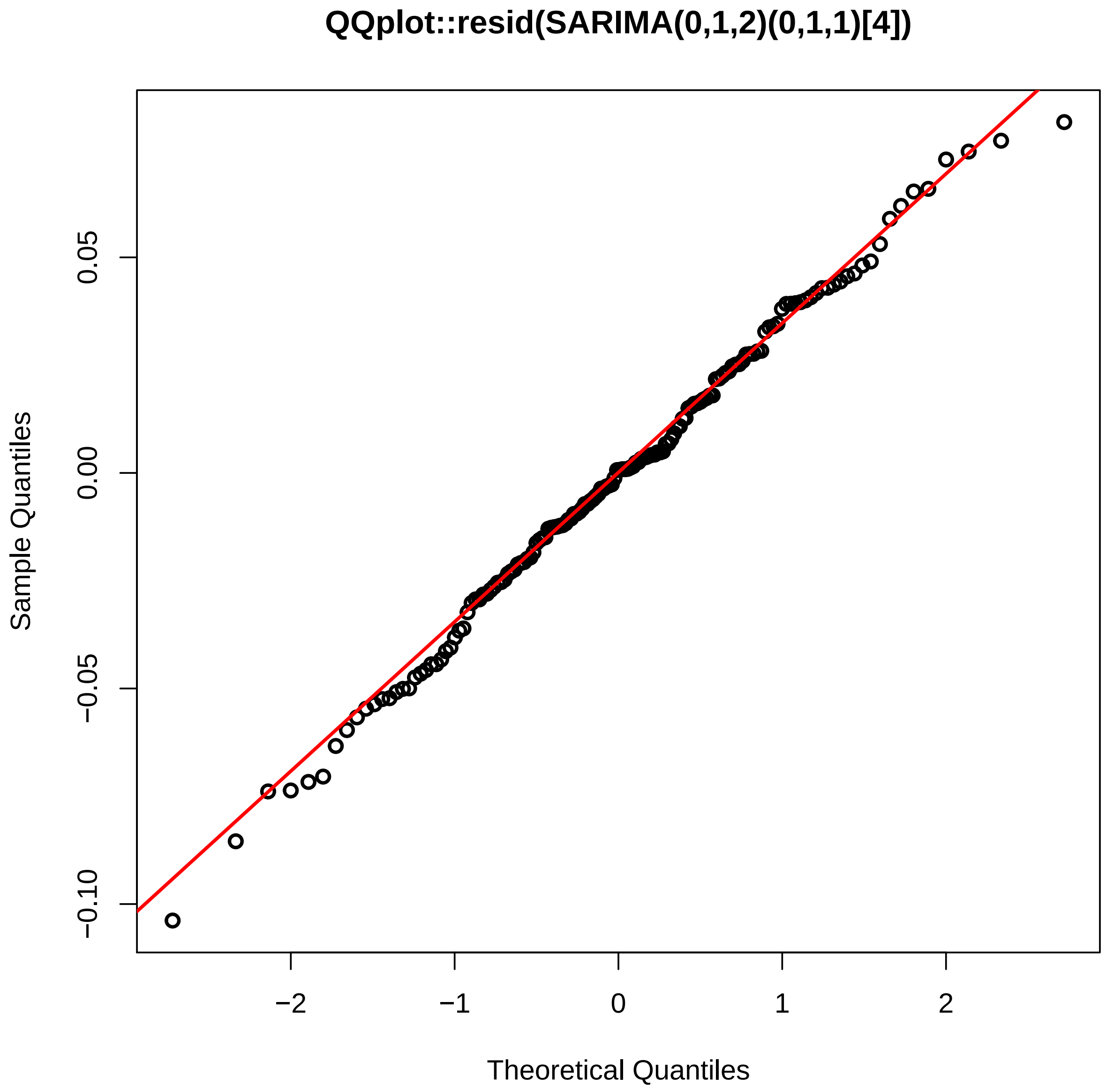


ACF::resid(SARIMA(0,1,2)(0,1,1)[4])



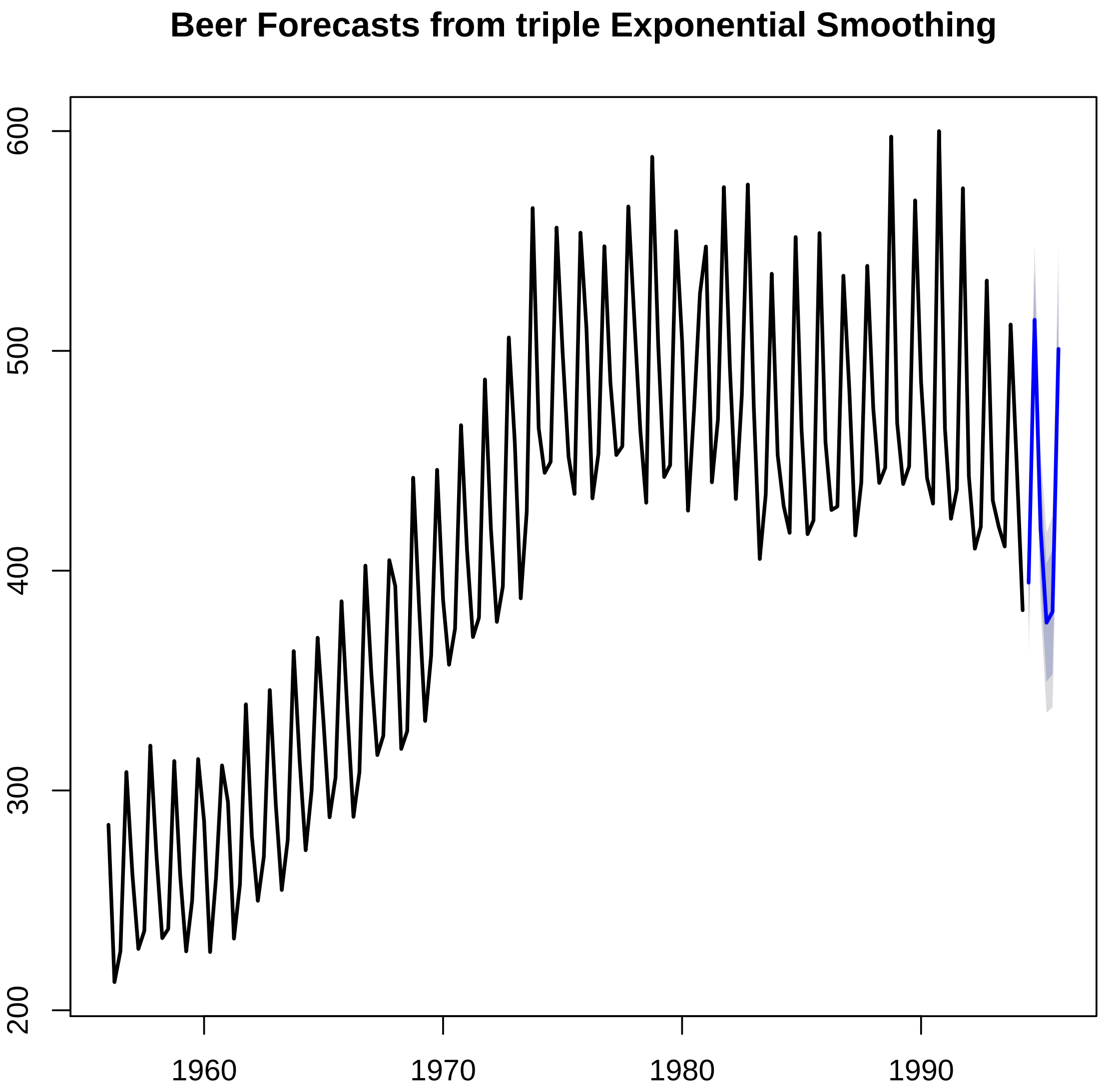
PACF::resid(SARIMA(0,1,2)(0,1,1)[4])

```
#The residual is stationary.

par(mfrow=c(1,1))
qqnorm(resid(sarima_fit),lwd=2, main="QQplot::resid(SARIMA(0,1,2)(0,1,1)[4])")
qqline(resid(sarima_fit), lwd=2, col="red")
```



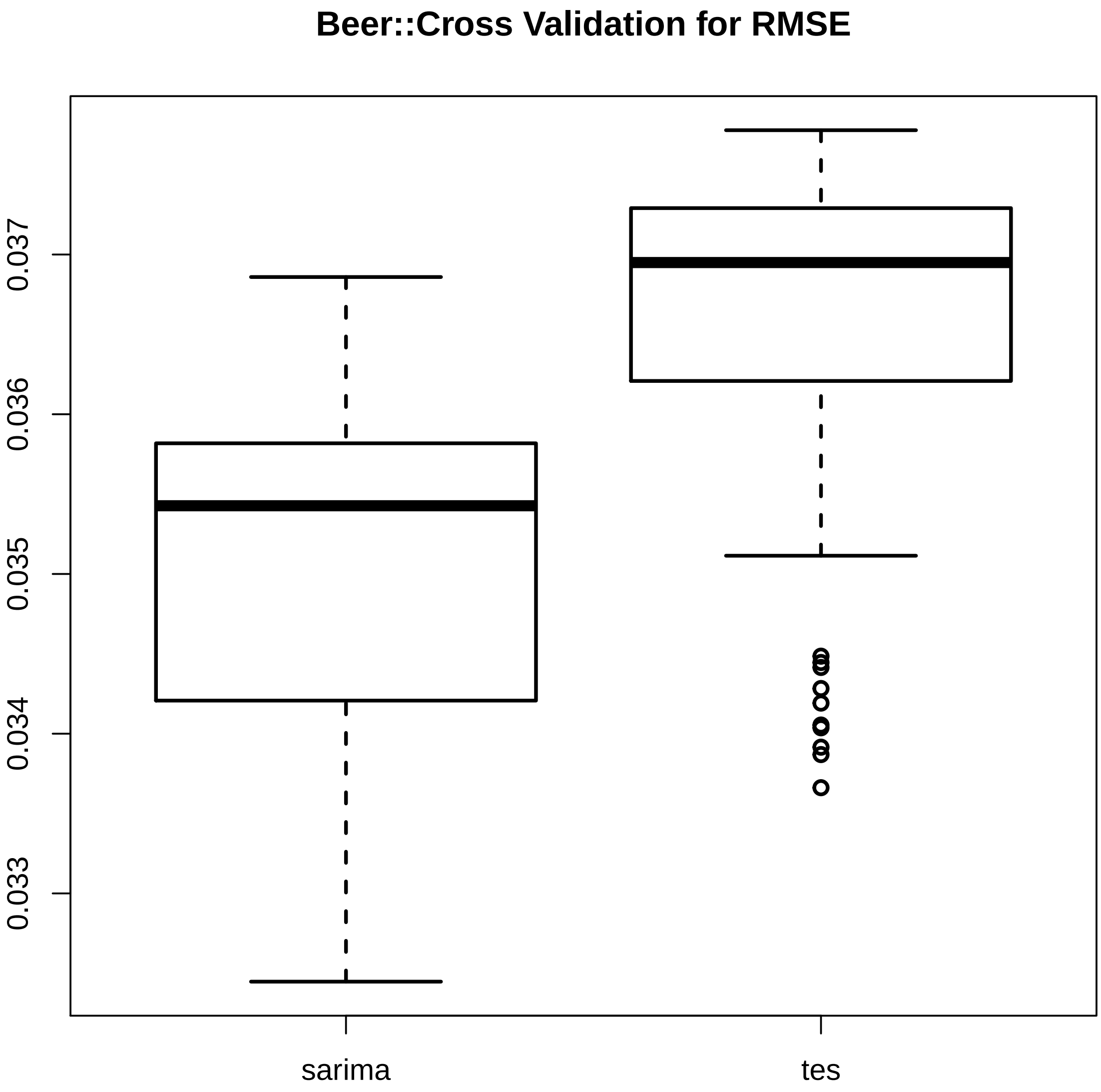
The distribution of residuals can be regard as a gaussian distribution.
So, SARIMA(0,1,2)(0,1,1)[4] is a good model. 4. Another method is to use Triple exponential smoothing

```
DES_fit <- hw(beer, initial = "optimal", seasonal = "additive", h = 6)
DES_fit

##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 1994 Q3       394.5326 373.2603 415.8050 361.9994 427.0659
## 1994 Q4       514.1839 491.8614 536.5065 480.0445 548.3234
## 1995 Q1       419.3392 395.7355 442.9429 383.2404 455.4379
## 1995 Q2       376.3166 349.5231 403.1101 335.3394 417.2937
## 1995 Q3       381.2753 352.8547 409.6960 337.8097 424.7410
## 1995 Q4       500.9267 470.6690 531.1844 454.6515 547.2018

plot(DES_fit,main = "Beer Forecasts from triple Exponential Smoothing", lwd = 2)
```



5. Use cross-validation to compare these two models

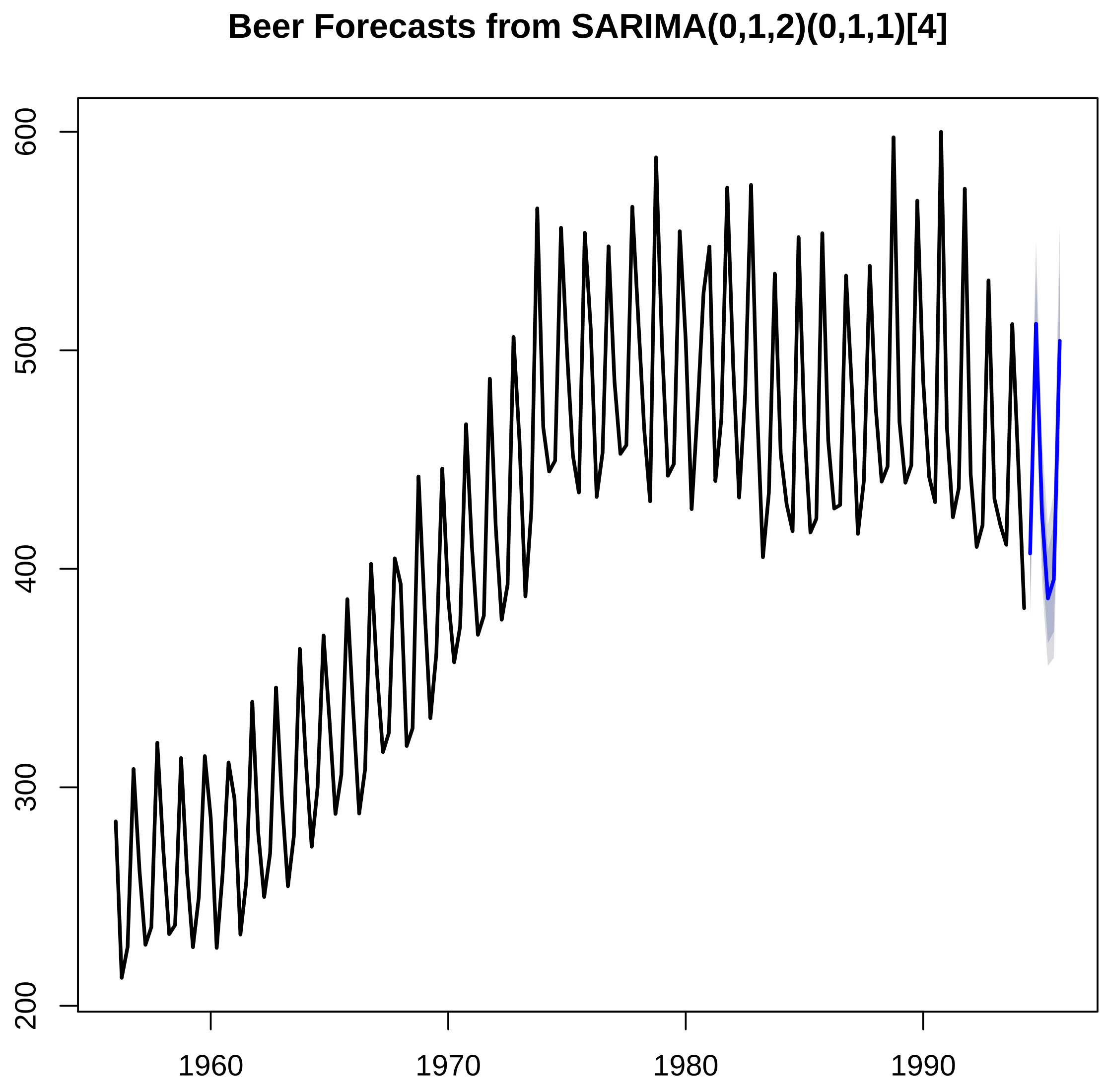
```
CV <-  function(time_series, start, forecast_length,ts_model){
  ts_length <-  length(time_series)
  accuracy_list = c()
  for(k in start:(ts_length - forecast_length)){
    fitted_model <- ts_model(ts(time_series[0:k],frequency = 4))
    RMSE <-  accuracy(forecast(fitted_model, h = forecast_length))[2]
    accuracy_list = c(accuracy_list, RMSE)
  }
  return(accuracy_list)
}

#Define two models
model_SARIMA <- function(ts)
  return(Arima(ts, order = c(0,1,2), seasonal = c(0,1,1)))
model_TES <- function(ts)
  return(hw(ts,initial = "optimal", seasonal = "additive"))

start <- 90
forecast_length <- 6
CV_beer <- data.frame(
  sarima = CV(log_beer, start, forecast_length, model_SARIMA),
  tes = CV(log_beer, start, forecast_length, model_TES)
)

boxplot(CV_beer,main = "Beer::Cross Validation for RMSE", lwd=2)
```



From boxplot, SARIMA(0,1,2)(0,1,1)[4] gives a better prediction, because it gives a lower RMSE. 6.

Forecast the number of birth during the two weeks by using SARIMA(0,1,2)(0,1,1)[4].

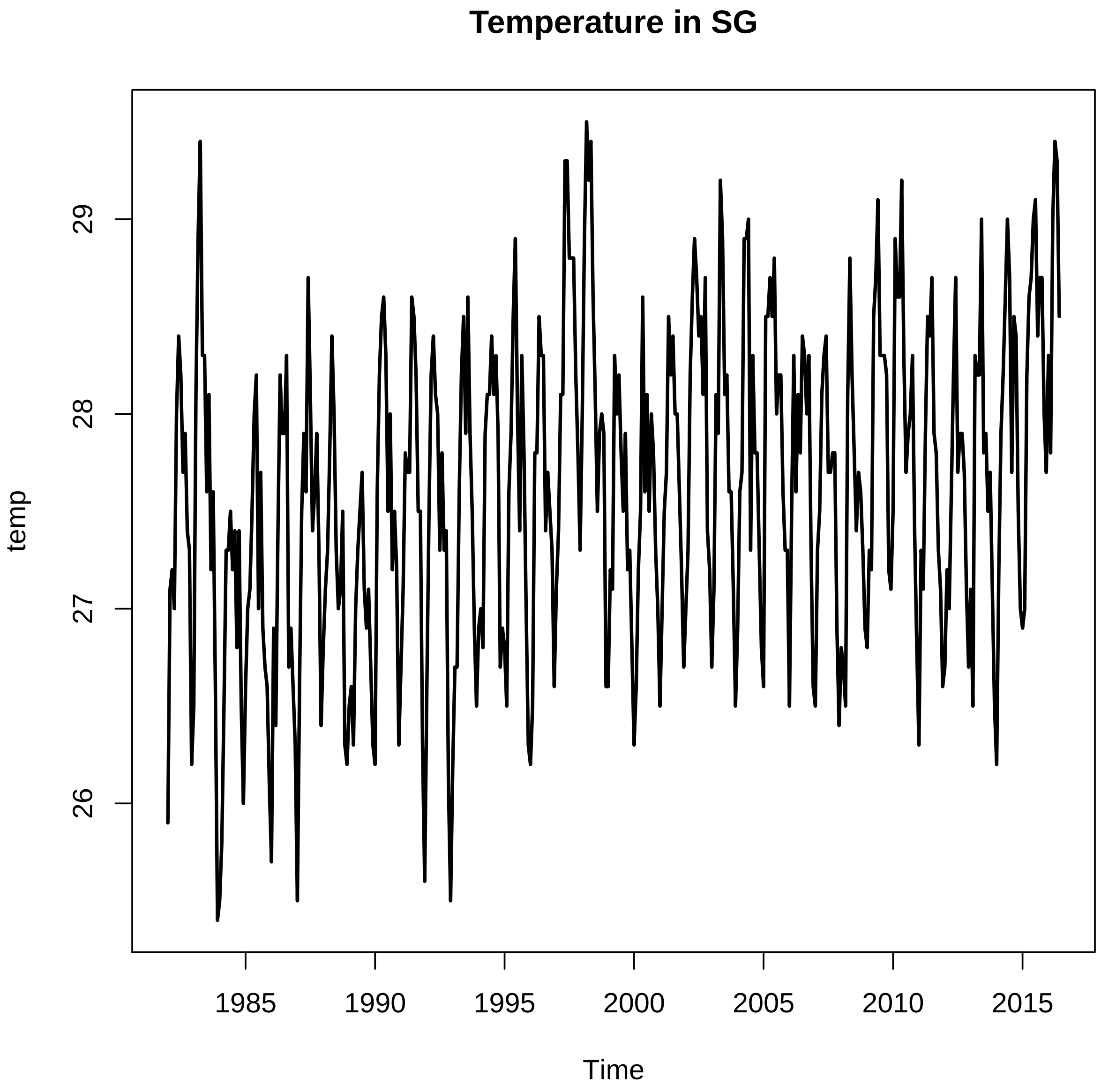
```
sarima_forecast <- forecast(sarima_fit, h = 6)
sarima_forecast$x<-exp(sarima_forecast$x)
sarima_forecast$lower<-exp(sarima_forecast$lower)
sarima_forecast$upper<-exp(sarima_forecast$upper)
sarima_forecast$mean<-exp(sarima_forecast$mean)
sarima_forecast

##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 1994 Q3       407.0242 388.5254 426.4037 379.0755 437.0334
## 1994 Q4       512.2461 488.9607 536.6404 477.0658 550.0208
## 1995 Q1       425.5336 404.5116 447.6481 393.8069 459.8163
## 1995 Q2       386.4387 365.9450 408.0801 355.5399 420.0228
## 1995 Q3       395.0946 371.2026 420.5244 359.1455 434.6421
## 1995 Q4       504.3541 472.3371 538.5413 456.2195 557.5673

plot(sarima_forecast, main = "Beer Forecasts from SARIMA(0,1,2)(0,1,1)[4]",lwd = 2)
```



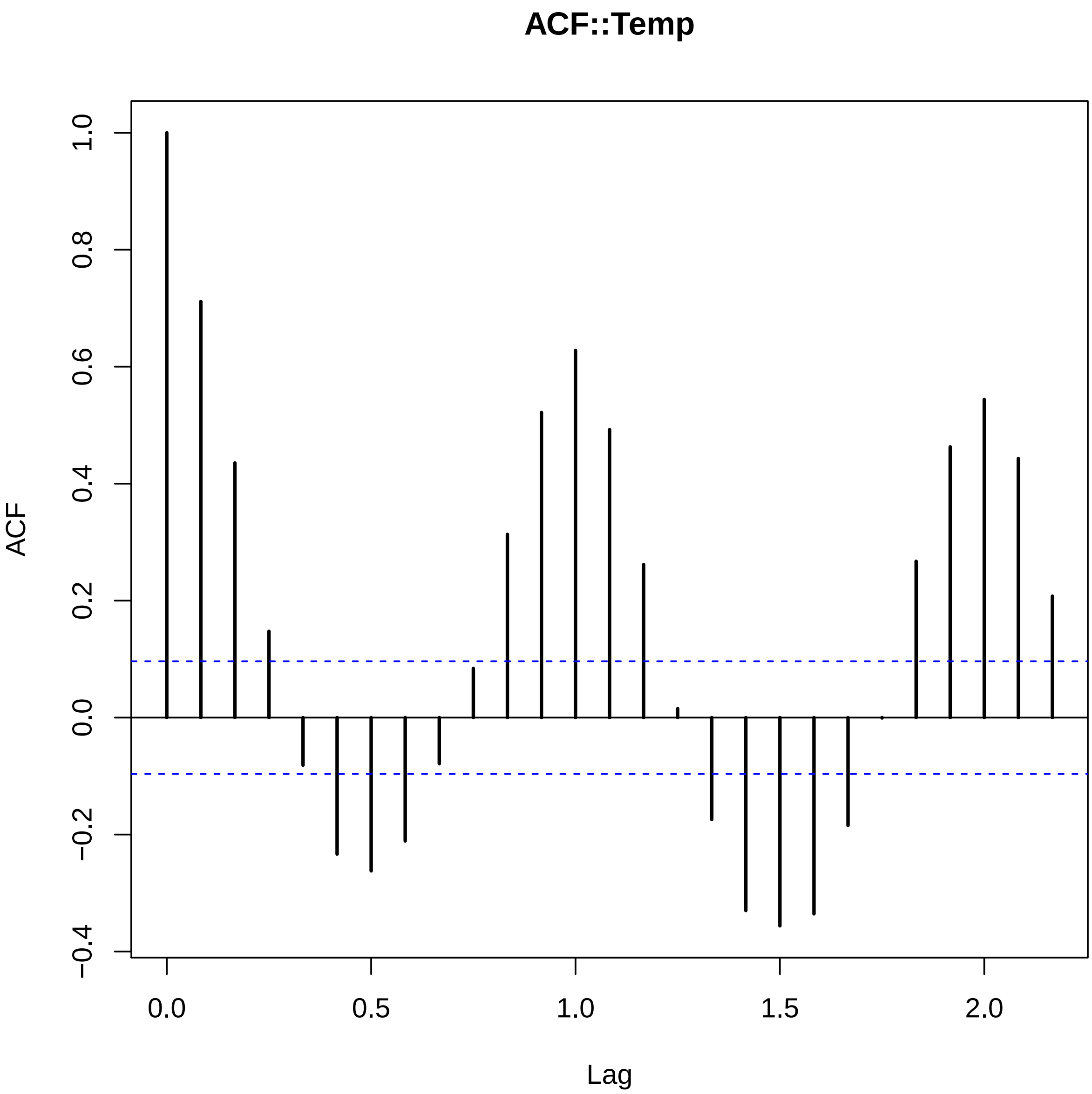
4 Exercise 4 (Temperature in Singapore?)

1. Load the data and plot.

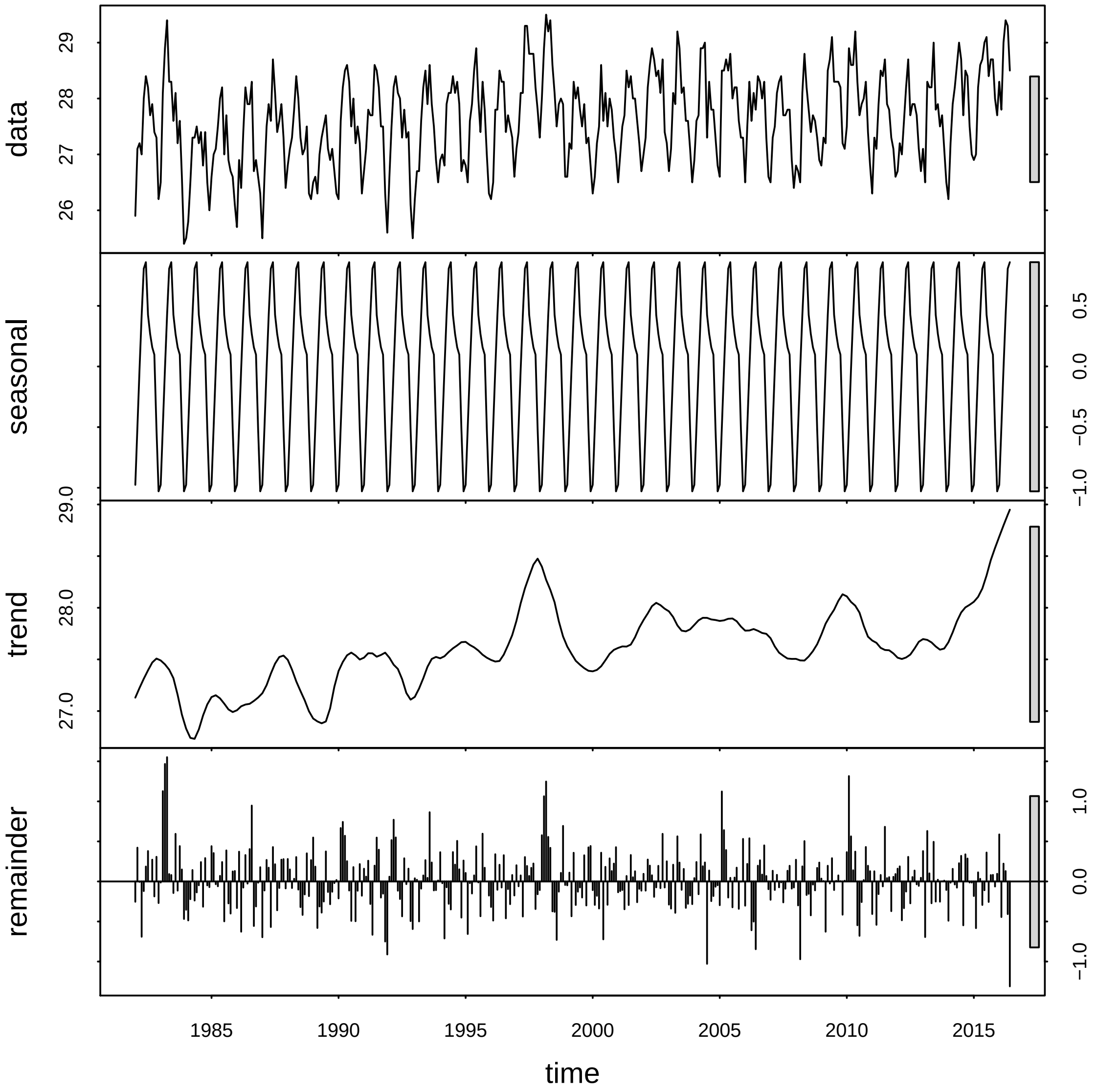
```
temp_data <- read.csv("E:/ST3233/Assignment2/Datasets/temperature_in_singapore.csv",
                      header= TRUE,sep=",")
temp <- ts(temp_data$mean_temp,frequency = 12,
           start=c(1982,1))
par(mfrow=c(1,1))
plot(temp,lwd = 2, main = "Temperature in SG")
```



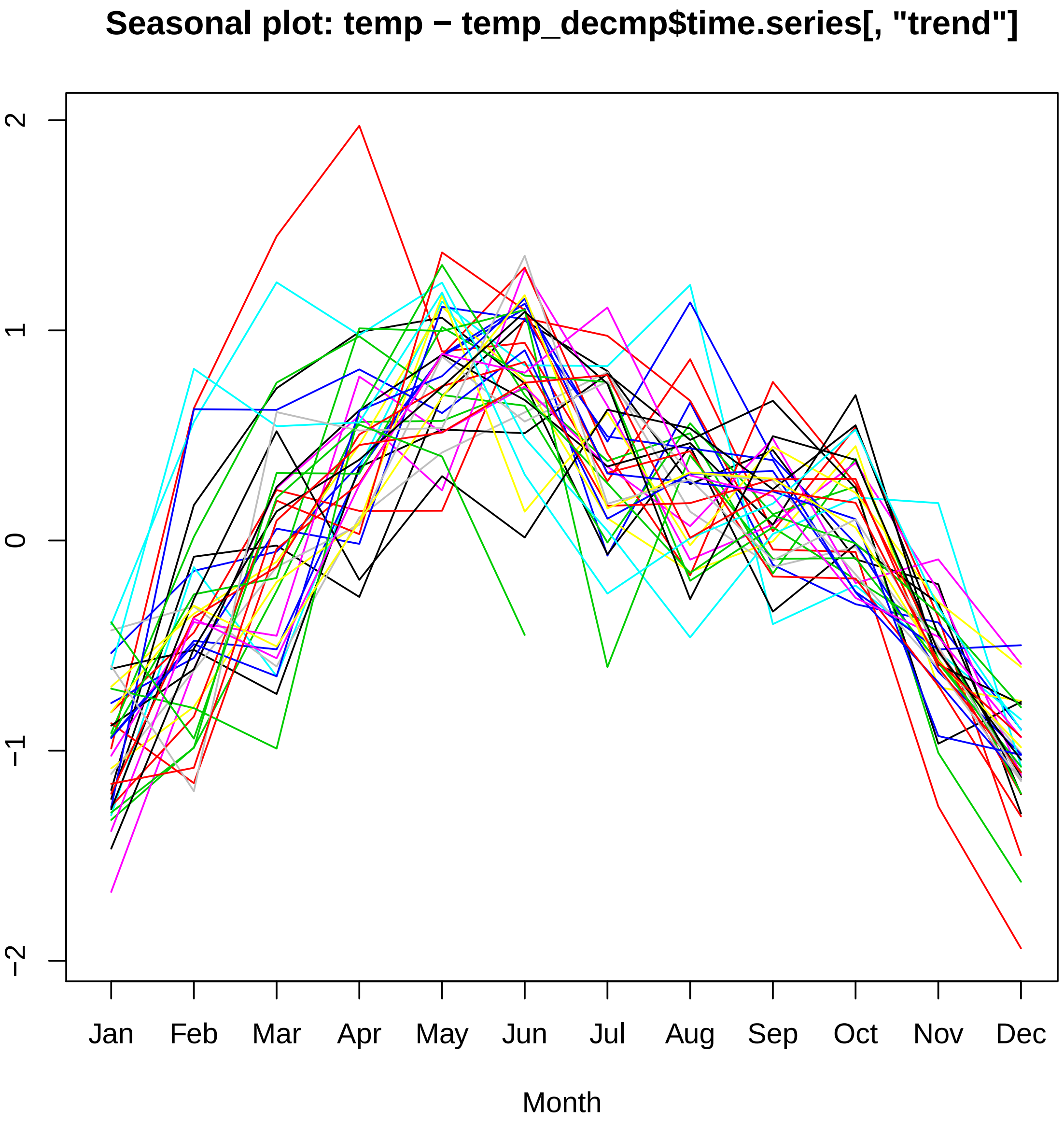
```
acf(temp,lwd = 2 , main = "ACF::Temp")
```



```
#The time series {temp} is not stationary and has seasonal behavior, decompose it.
temp_decmp <-stl(temp,s.window = "periodic", robust = T)
plot(temp_decmp)
```

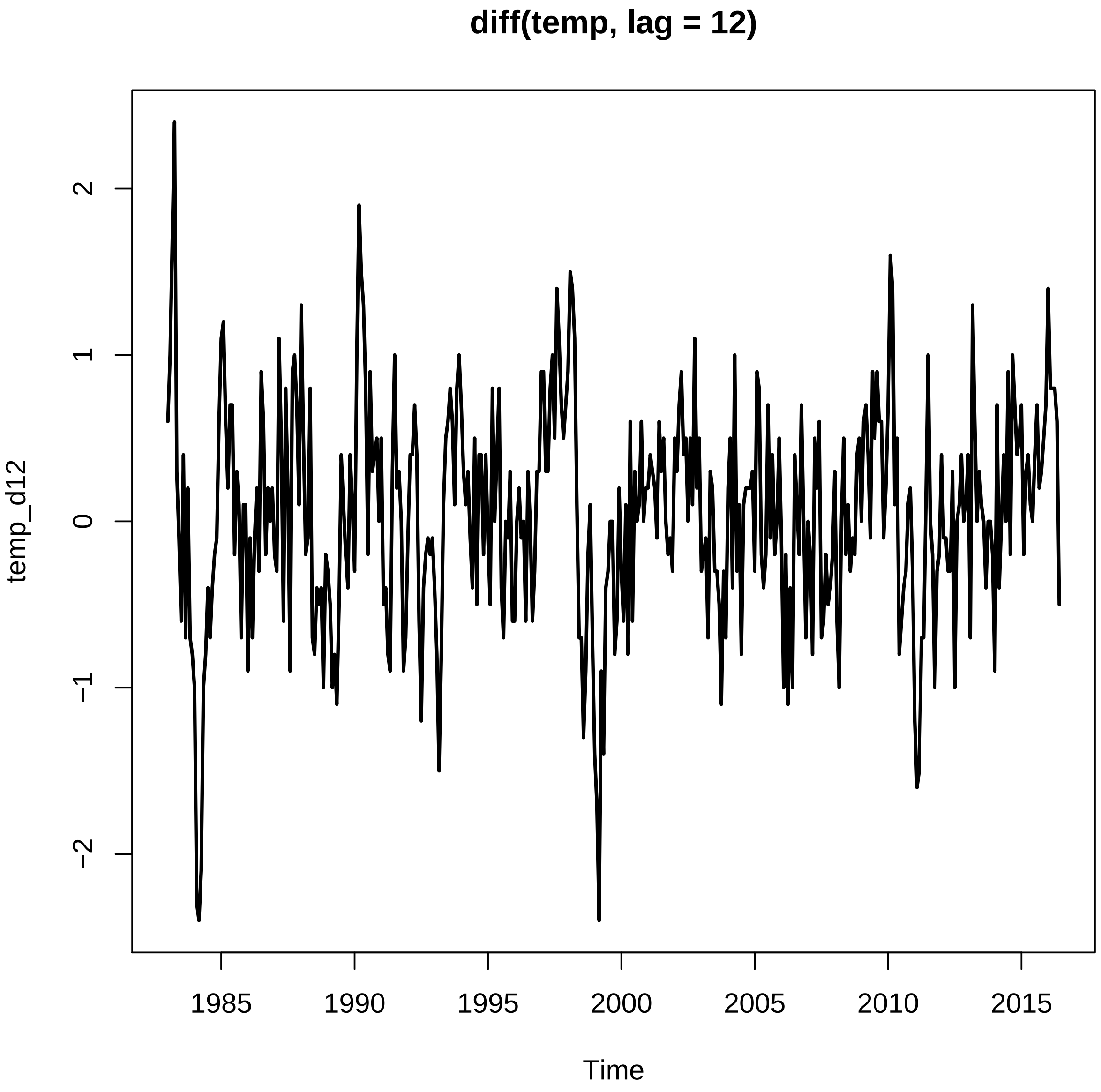


```
seasonplot(temp-temp_decmp$time.series[,"trend"],s = 12, col = 1:12, type = "l")
```

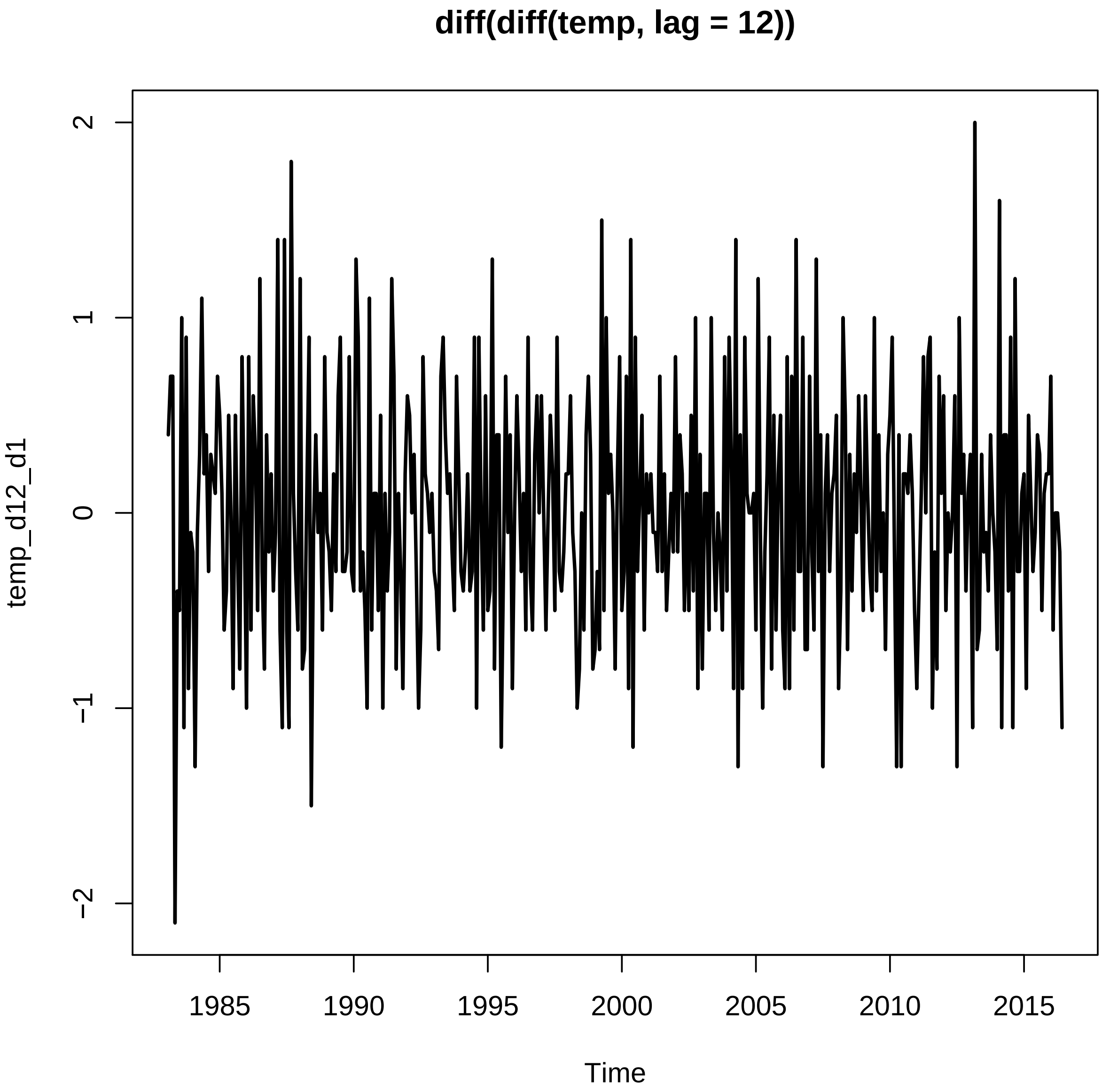


There is seasonal component and trend, thus, use SARIMA model 2. Fit the SARIMA model.

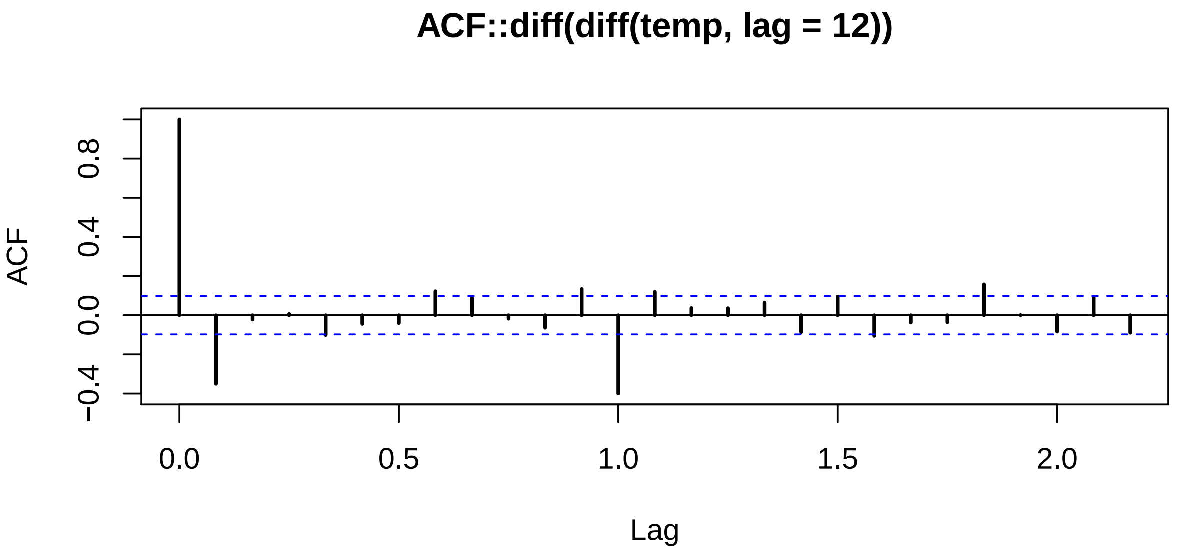
```
temp_d12 = diff(temp, lag = 12)
plot(temp_d12,lwd = 2, main = "diff(temp, lag = 12)")
```

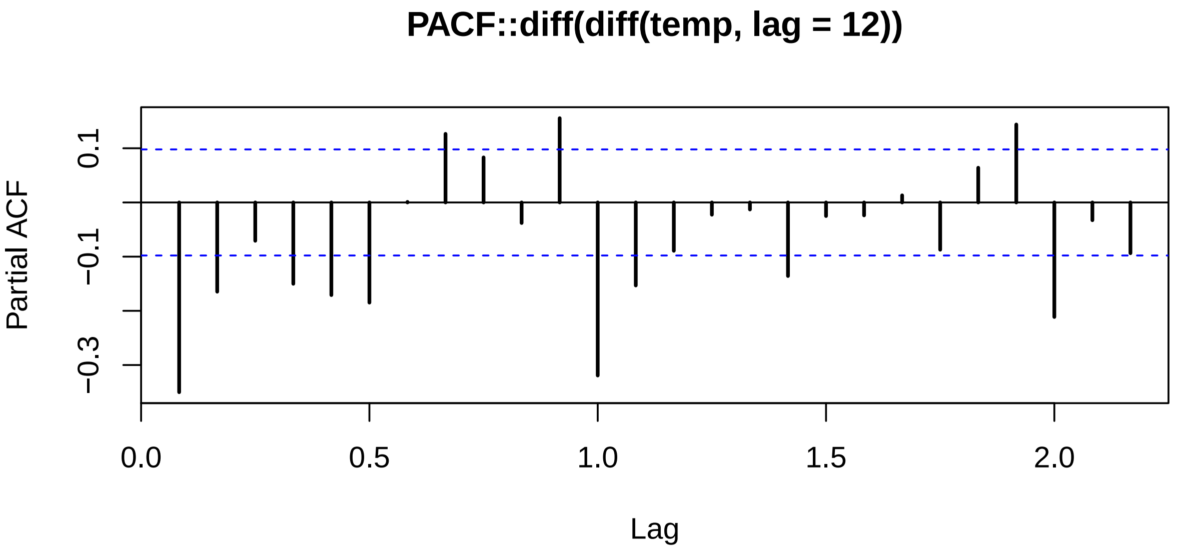


```
temp_d12_d1 = diff(temp_d12, lag = 1)
plot(temp_d12_d1,lwd = 2, main = "diff(diff(temp, lag = 12))")
```



```
par(mfrow=c(2,1))
acf(temp_d12_d1,lwd = 2, main = "ACF::diff(diff(temp, lag = 12))")
pacf(temp_d12_d1,lwd = 2, main = "PACF::diff(diff(temp, lag = 12))")
```





From acf plot ,q $\leq 1, and from partial - acf plot, p \leq 5 with d = D = 1 and P \leq 1, Q \leq 1.$

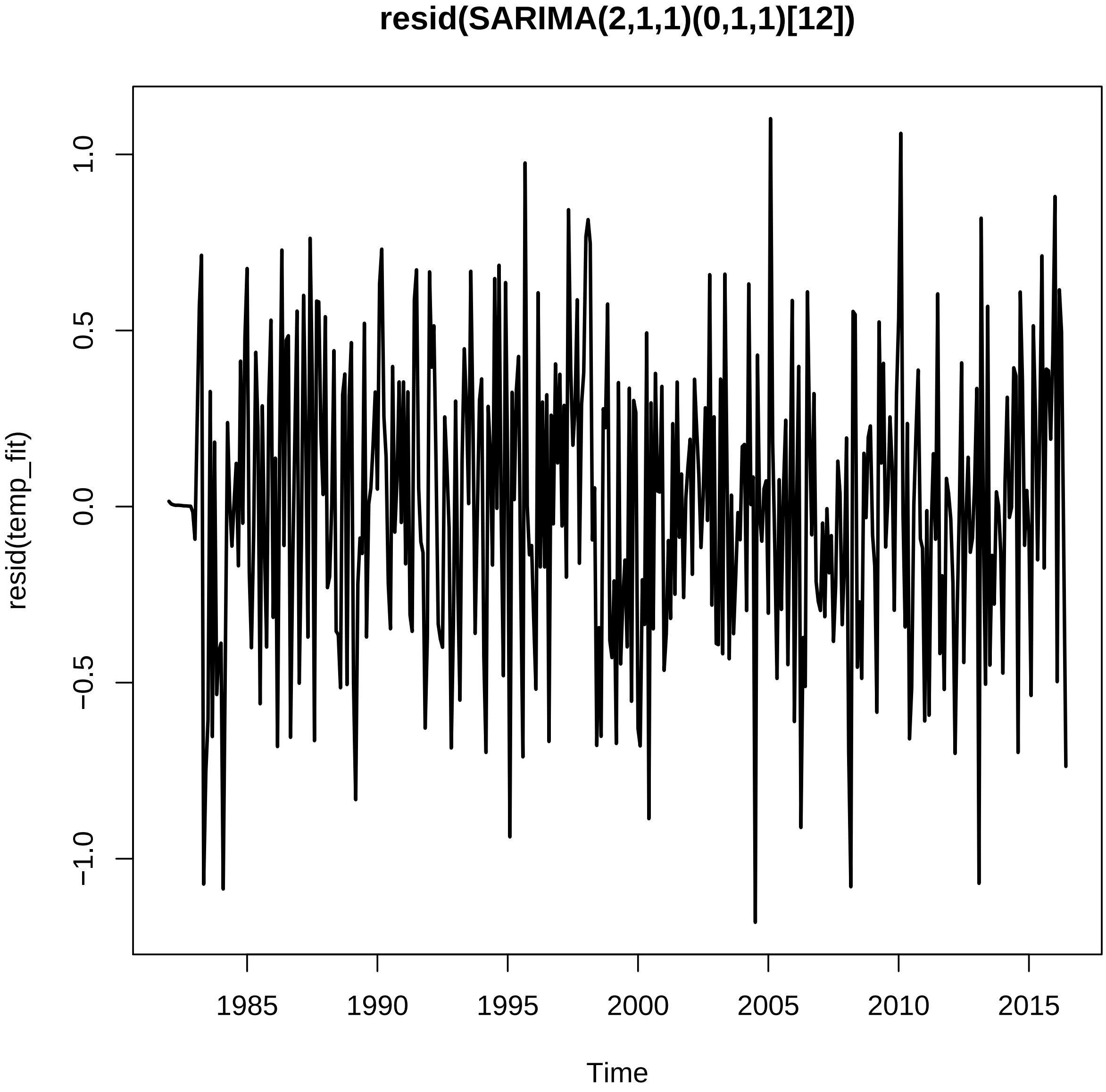
```
AIC_best <- 10**6
for (p in 0:5){
  for (q in 0:1){
    for (P in 0:1){
      for (Q in 0:1){
        fit_sarima <- Arima(temp, order = c(p,1,q),seasonal = c(P,1,Q))
        if (fit_sarima$aic < AIC_best){
          AIC_best <- fit_sarima$aic
          cat("p = ",p,", q = ",q,",P = ",P,",Q = ",Q,"\t AIC = ",AIC_best,"\n")
        }
      }
    }
  }
}

## p =  0 , q =  0 ,P =  0 ,Q =  0   AIC =  780.5999
## p =  0 , q =  0 ,P =  0 ,Q =  1   AIC =  596.2691
## p =  0 , q =  0 ,P =  1 ,Q =  1   AIC =  593.1758
## p =  0 , q =  1 ,P =  0 ,Q =  1   AIC =  493.1153
## p =  1 , q =  1 ,P =  0 ,Q =  1   AIC =  488.5129
## p =  2 , q =  1 ,P =  0 ,Q =  1   AIC =  483.6007
## p =  2 , q =  1 ,P =  1 ,Q =  1   AIC =  483.2574
## p =  3 , q =  1 ,P =  0 ,Q =  1   AIC =  483.1996
## p =  3 , q =  1 ,P =  1 ,Q =  1   AIC =  483.1188
```

From the results, SARIMA(2,1,1)(0,1,1)[12] gives a lower AIC, with number of parameters = 4

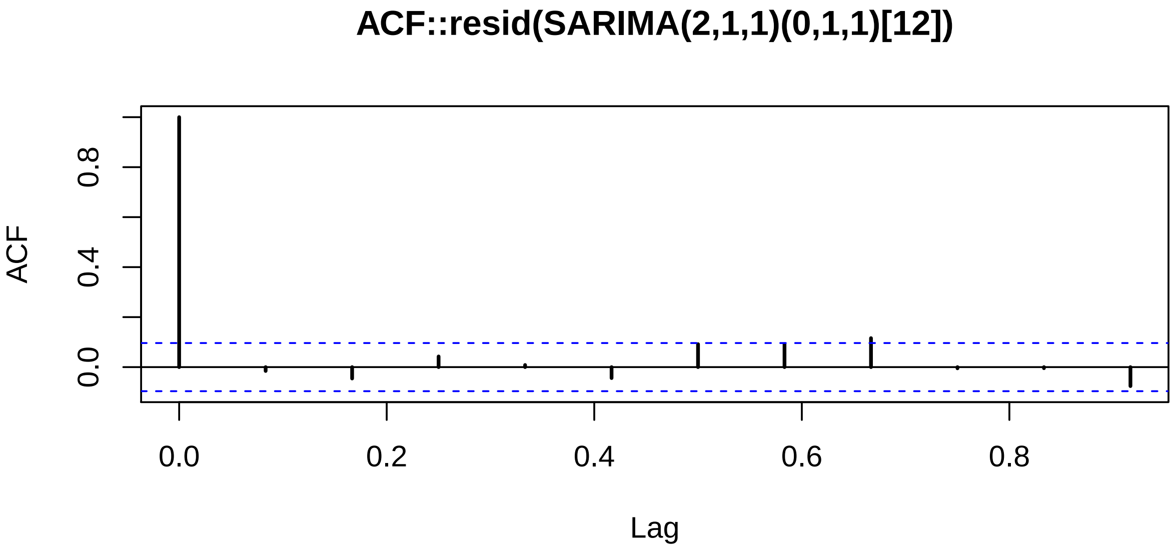
```
temp_fit <- Arima(temp, order = c(2,1,1), seasonal = c(0,1,1))
```

3. Then consider the residuals of the SARIMA model.

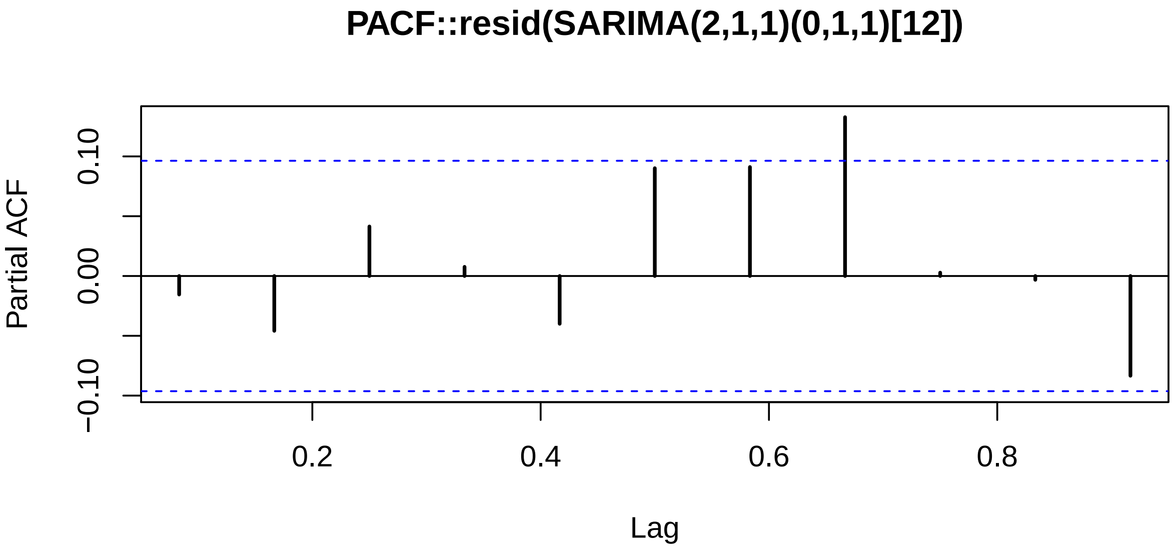
```
par(mfrow=c(1,1))
plot(resid(temp_fit),lwd=2, main="resid(SARIMA(2,1,1)(0,1,1)[12])")
```



```
par(mfrow=c(2,1))
acf(resid(temp_fit),lwd=2, main="ACF::resid(SARIMA(2,1,1)(0,1,1)[12])",lag.max = 11)
pacf(resid(temp_fit),lwd=2, main="PACF::resid(SARIMA(2,1,1)(0,1,1)[12])",lag.max = 11)
```

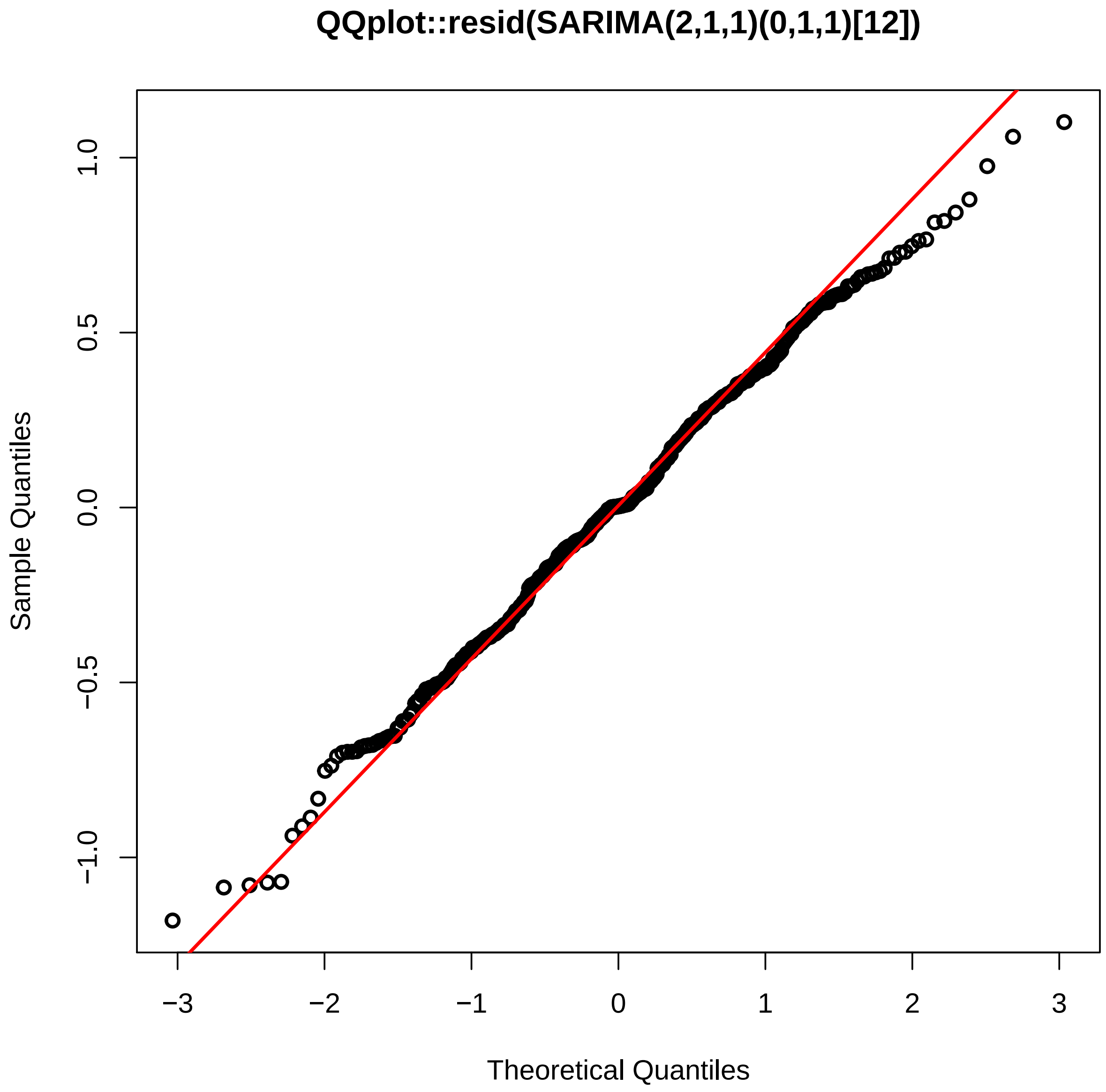


ACF::resid(SARIMA(2,1,1)(0,1,1)[12])



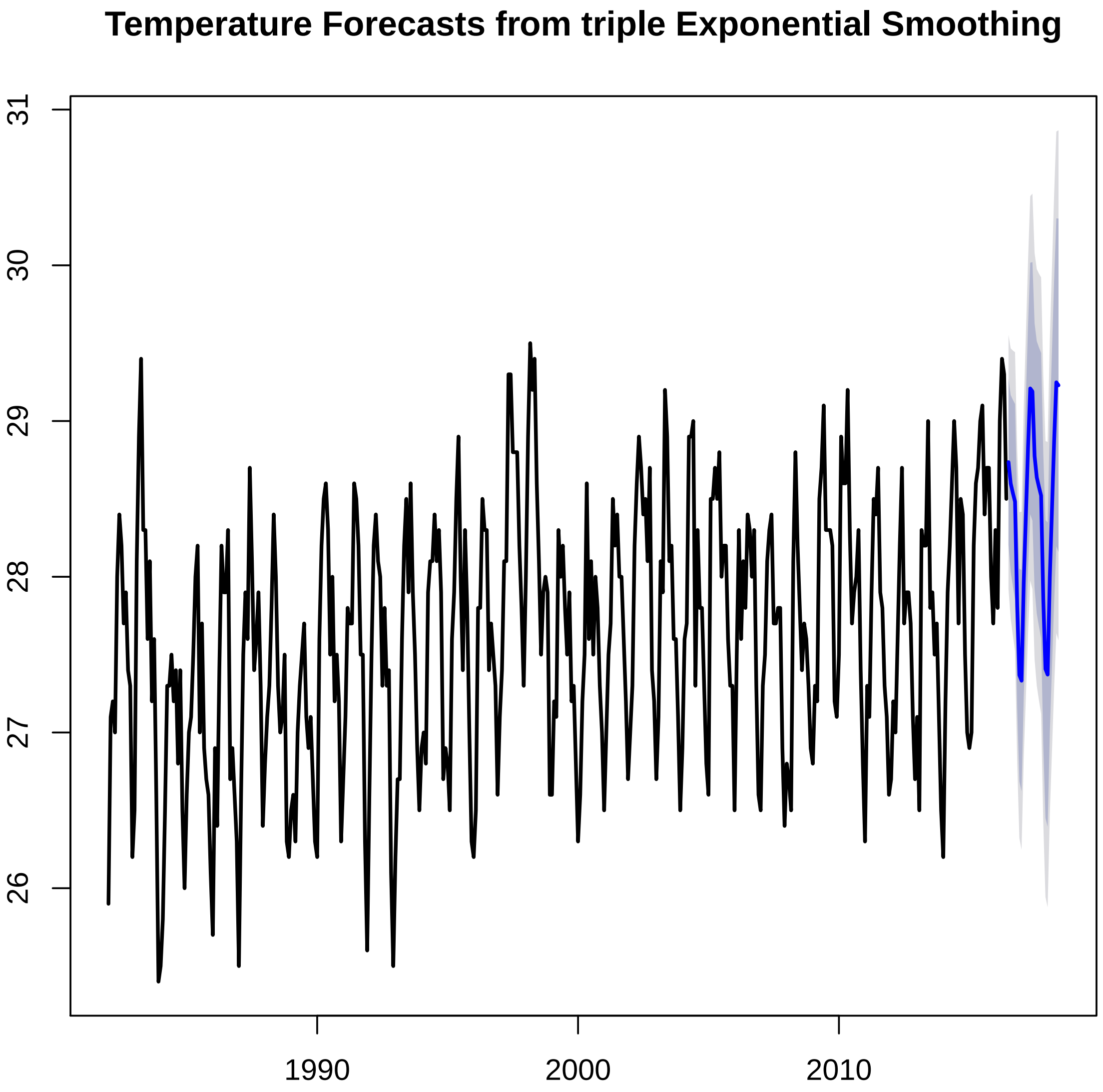
PACF::resid(SARIMA(2,1,1)(0,1,1)[12])

```r
#The residual is stationary.
par(mfrow=c(1,1))
qqnorm(resid(temp_fit),lwd=2, main="QQplot::resid(SARIMA(2,1,1)(0,1,1)[12])")
qqline(resid(temp_fit), lwd=2, col="red")
```



The residual follows a Gaussian Distribution. So, SARIMA(2,1,1)(0,1,1)[12] is a good smodel. 4. Another method is to use Triple exponential smoothing

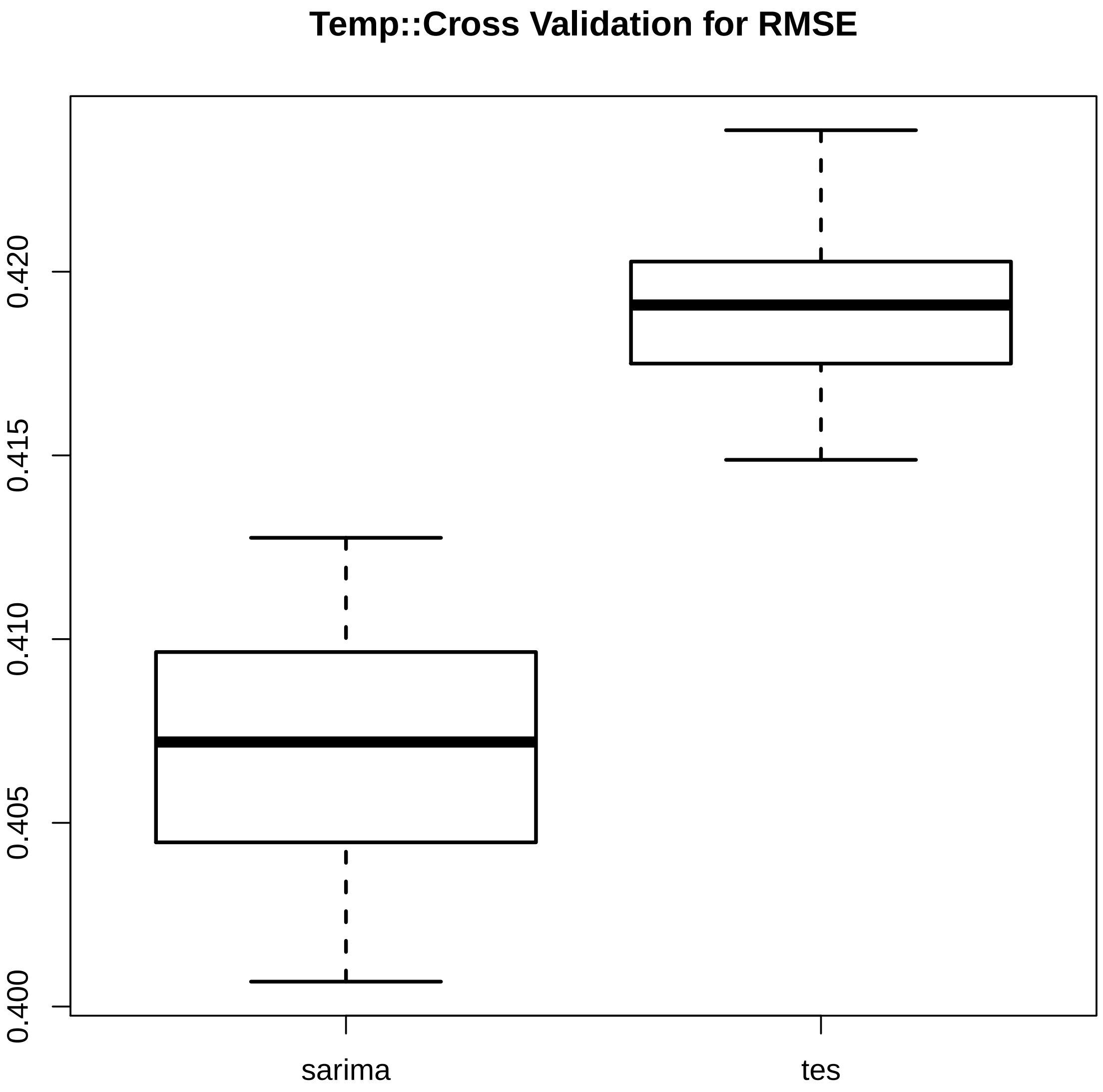
```
DES_fit <- hw(temp, initial = "optimal", seasonal = "additive", h = 2*12)
DES_fit

##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jul 2016       28.73622 28.20193 29.27051 27.91910 29.55334
## Aug 2016       28.59954 28.03249 29.16659 27.73232 29.46677
## Sep 2016       28.53805 27.93993 29.13618 27.62331 29.45280
## Oct 2016       28.48181 27.85405 29.10957 27.52174 29.44188
## Nov 2016       27.81602 27.15986 28.47217 26.81252 28.81952
## Dec 2016       27.36870 26.68524 28.05216 26.32344 28.41397
## Jan 2017       27.33246 26.62266 28.04227 26.24691 28.41802
## Feb 2017       27.96003 27.22474 28.69533 26.83549 29.08457
## Mar 2017       28.39468 27.63466 29.15470 27.23234 29.55702
## Apr 2017       28.83468 28.05065 29.61872 27.63560 30.03376
## May 2017       29.20939 28.40197 30.01681 27.97455 30.44423
## Jun 2017       29.19020 28.35997 30.02044 27.92047 30.45994
## Jul 2017       28.77542 27.92291 29.62792 27.47163 30.07921
## Aug 2017       28.63874 27.76446 29.51301 27.30165 29.97582
## Sep 2017       28.57725 27.68166 29.47284 27.20757 29.94694
## Oct 2017       28.52101 27.60453 29.43749 27.11938 29.92264
## Nov 2017       27.85521 26.91824 28.79218 26.42224 29.28818
## Dec 2017       27.40790 26.45081 28.36499 25.94416 28.87164
## Jan 2018       27.37166 26.39480 28.34852 25.87768 28.86564
## Feb 2018       27.99923 27.00293 28.99554 26.47551 29.52295
## Mar 2018       28.43388 27.41844 29.44932 26.88089 29.98686
## Apr 2018       28.87388 27.83959 29.90816 27.29208 30.45568
## May 2018       29.24859 28.19574 30.30144 27.63839 30.85878
## Jun 2018       29.22940 28.15823 30.30057 27.59119 30.86761

plot(DES_fit,main = "Temperature Forecasts from triple Exponential Smoothing", lwd = 2)
```



5. Use cross-validation to compare these two models

```r
CV <-  function(time_series, start, forecast_length,ts_model){
  ts_length <-  length(time_series)
  accuracy_list = c()
  for(k in start:(ts_length - forecast_length)){
    fitted_model <- ts_model(ts(time_series[0:k],frequency = 12))
    RMSE <-  accuracy(forecast(fitted_model, h = forecast_length))[2]
    accuracy_list = c(accuracy_list, RMSE)
  }
  return(accuracy_list)
}
```

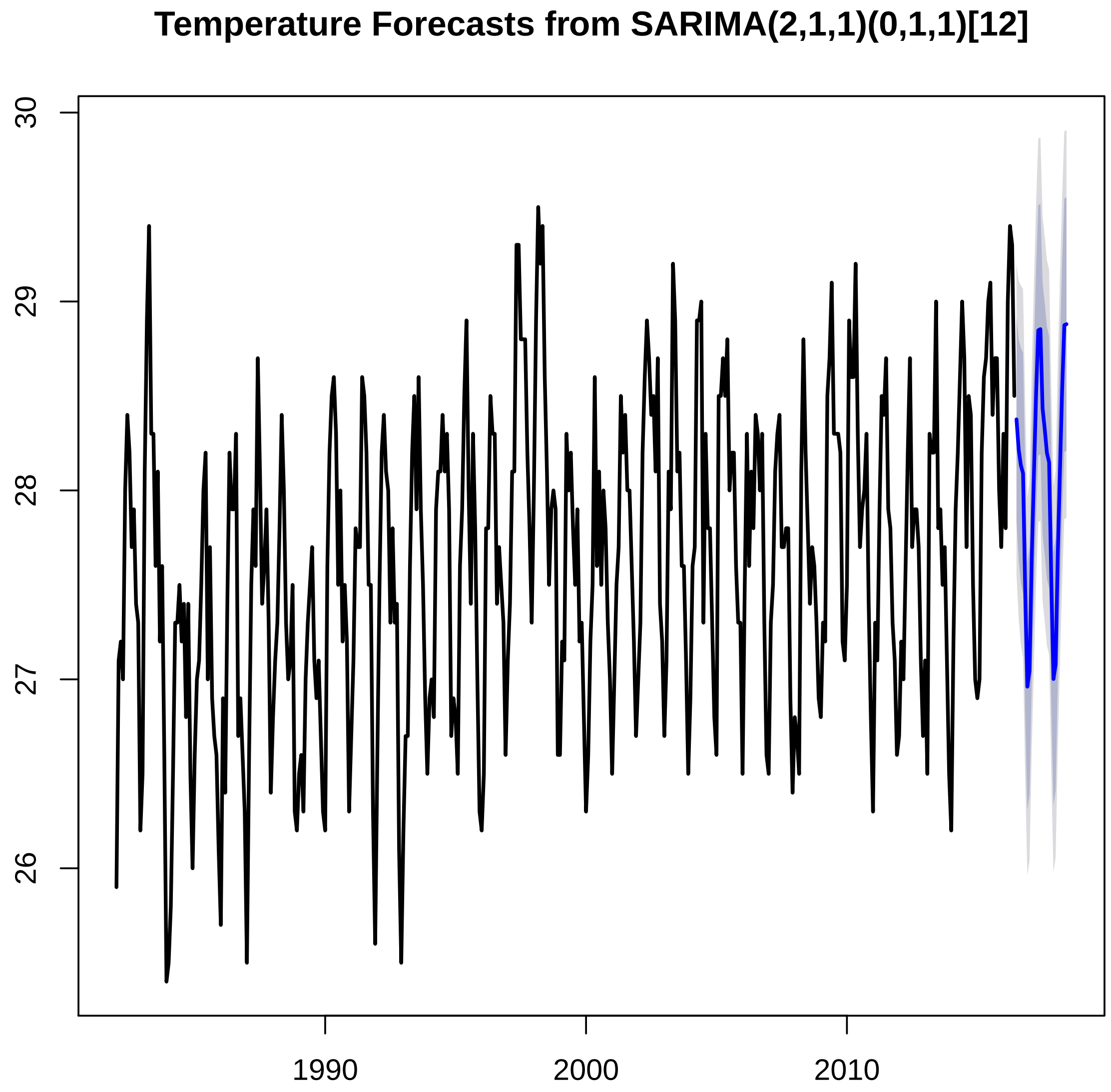
```r
model_SARIMA <- function(ts)
  return(Arima(ts, order = c(2,1,1), seasonal = c(0,1,1)))
model_TES <- function(ts)
  return(hw(ts,initial = "optimal", seasonal = "additive"))

start <- 300
forecast_length <- 24
CV_temp <- data.frame(
  sarima = CV(temp, start, forecast_length, model_SARIMA),
  tes = CV(temp, start, forecast_length, model_TES)
)

boxplot(CV_temp,main = "Temp::Cross Validation for RMSE", lwd=2)
```



From boxplot, SARIMA(2,1,1)(0,1,1)[12] gives a better prediction, due to the lower RMSE. 6. Forecast the number of birth during the two weeks by using SARIMA(2,1,1)(0,1,1)[12] model.

```
temp_forecast <- forecast(temp_fit, h = 2*12)
temp_forecast

##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jul 2016       28.37633 27.83817 28.91450 27.55328 29.19939
## Aug 2016       28.21659 27.63264 28.80053 27.32352 29.10966
## Sep 2016       28.13355 27.51165 28.75546 27.18243 29.08468
## Oct 2016       28.09030 27.45209 28.72852 27.11424 29.06637
## Nov 2016       27.47766 26.82966 28.12567 26.48662 28.46870
## Dec 2016       26.96136 26.30782 27.61489 25.96186 27.96085
## Jan 2017       27.04456 26.38762 27.70149 26.03986 28.04926
## Feb 2017       27.63363 26.97449 28.29277 26.62556 28.64170
## Mar 2017       28.07603 27.41536 28.73670 27.06562 29.08644
## Apr 2017       28.50624 27.84445 29.16803 27.49412 29.51836
## May 2017       28.84737 28.18470 29.51003 27.83391 29.86082
## Jun 2017       28.85382 28.19044 29.51720 27.83926 29.86837
## Jul 2017       28.43358 27.76872 29.09844 27.41676 29.45040
## Aug 2017       28.32352 27.65779 28.98925 27.30538 29.34166
## Sep 2017       28.19824 27.53172 28.86476 27.17888 29.21759
## Oct 2017       28.14836 27.48117 28.81556 27.12798 29.16875
## Nov 2017       27.52417 26.85636 28.19198 26.50284 28.54550
## Dec 2017       27.00181 26.33343 27.67020 25.97961 28.02402
## Jan 2018       27.08014 26.41124 27.74904 26.05715 28.10313
## Feb 2018       27.66597 26.99660 28.33534 26.64226 28.68968
## Mar 2018       28.10604 27.43622 28.77586 27.08164 29.13044
## Apr 2018       28.53463 27.86437 29.20489 27.50955 29.55971
## May 2018       28.87461 28.20390 29.54531 27.84885 29.90036
## Jun 2018       28.88025 28.20911 29.55139 27.85383 29.90667
```

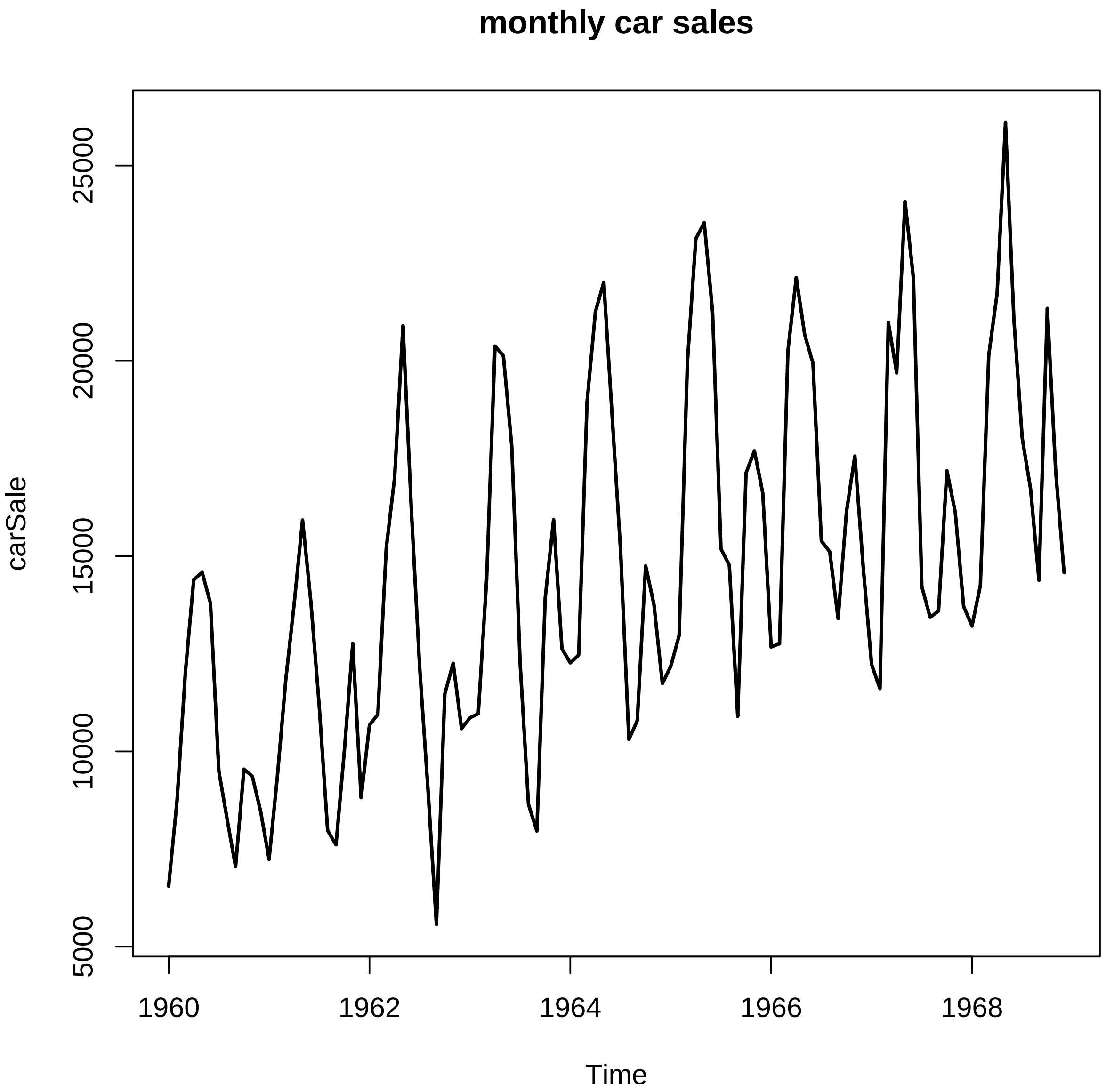
```
plot(temp_forecast, main = "Temperature Forecasts from SARIMA(2,1,1)(0,1,1)[12]",lwd = 2)
```



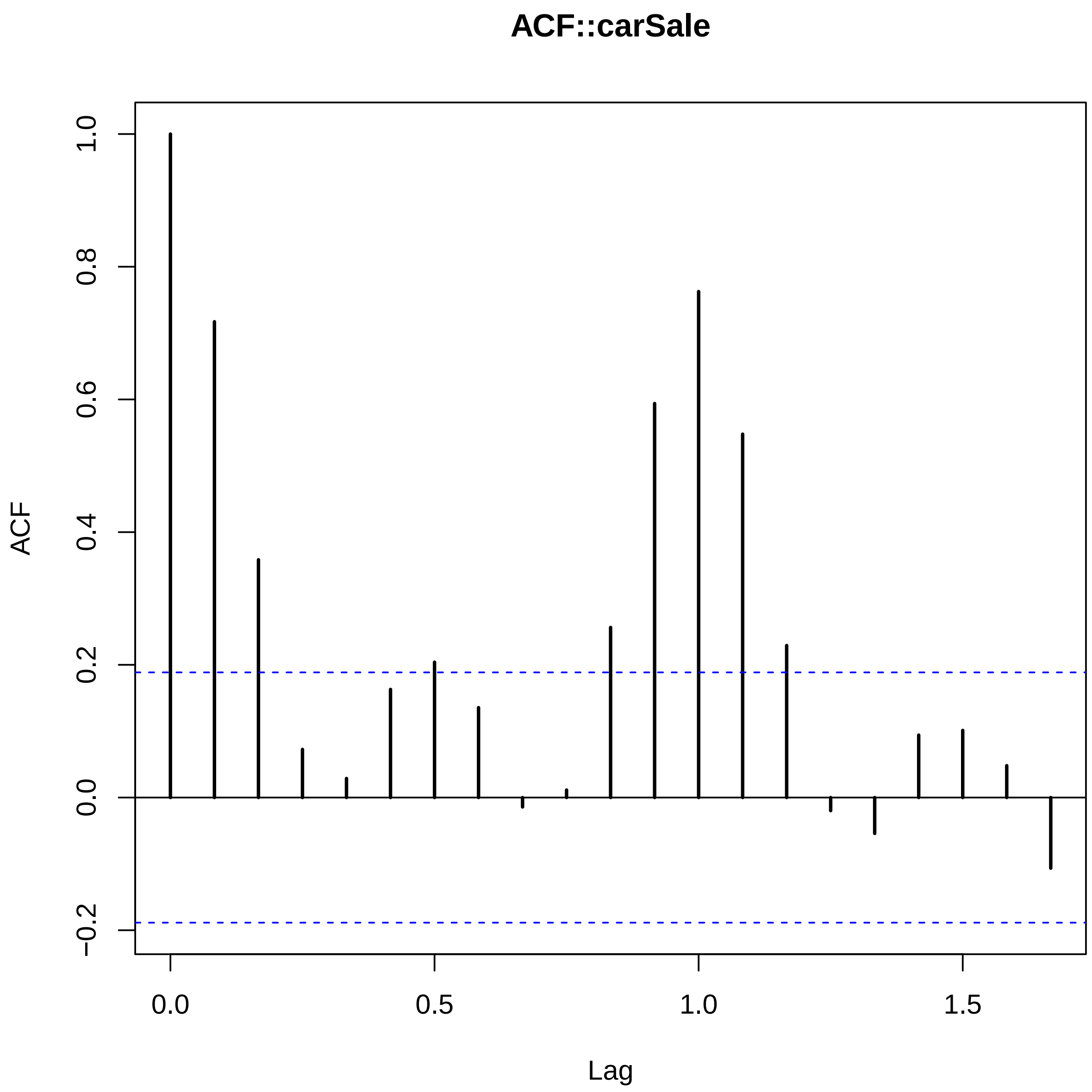
5 Exercise 5 (Monthly Car Sales in Quebuc?)

1. Load the data and plot.

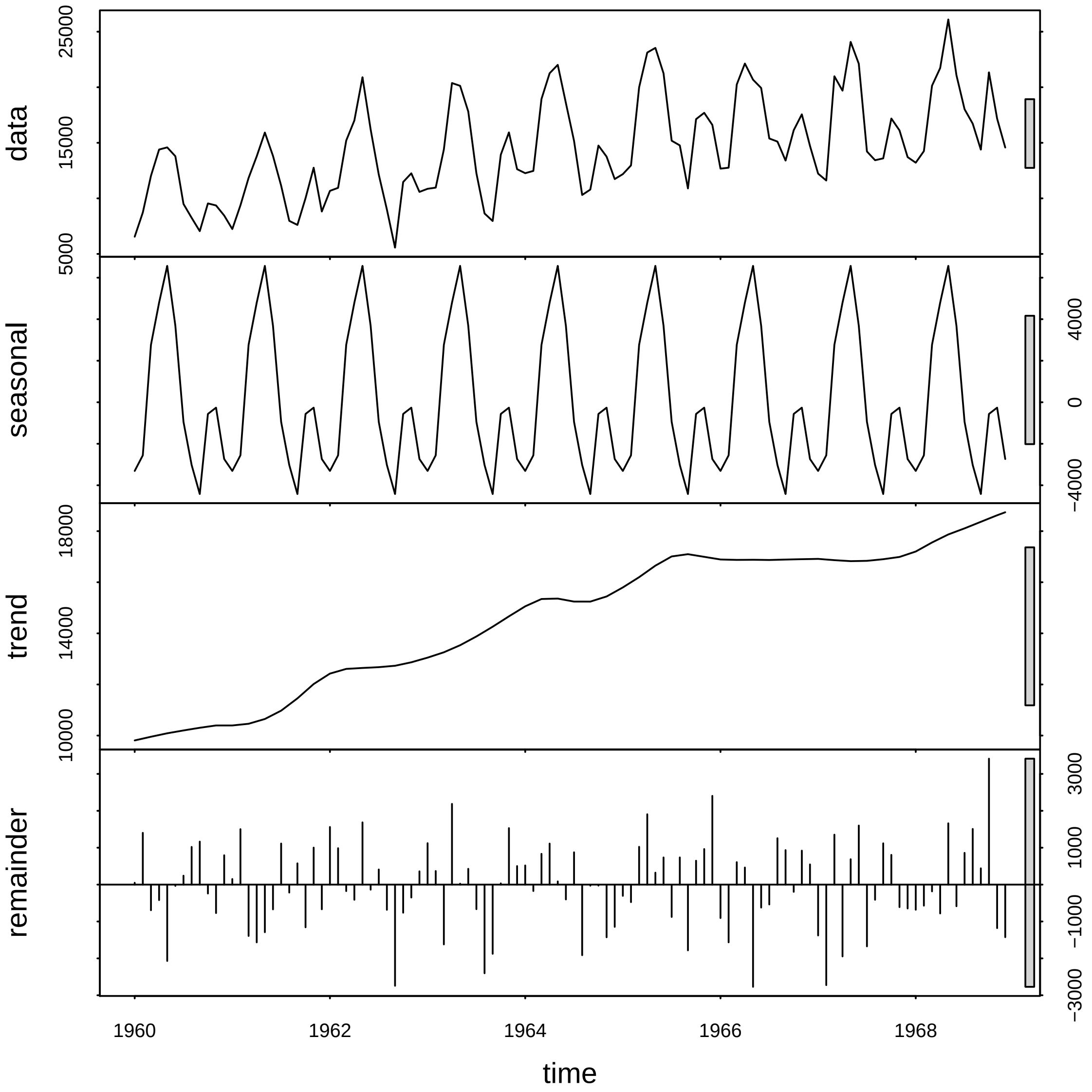
```
carSale_data <- read.csv("E:/ST3233/Assignment2/Datasets/monthly-car-sales-in-quebec-1960.csv",
                         header= TRUE,sep=",")
carSale <- ts(carSale_data$Monthly.car.sales.in.Quebec.1960.1968[1:108],
              frequency = 12,start=c(1960,1))
par(mfrow=c(1,1))
plot(carSale,lwd = 2, main = "monthly car sales")
```



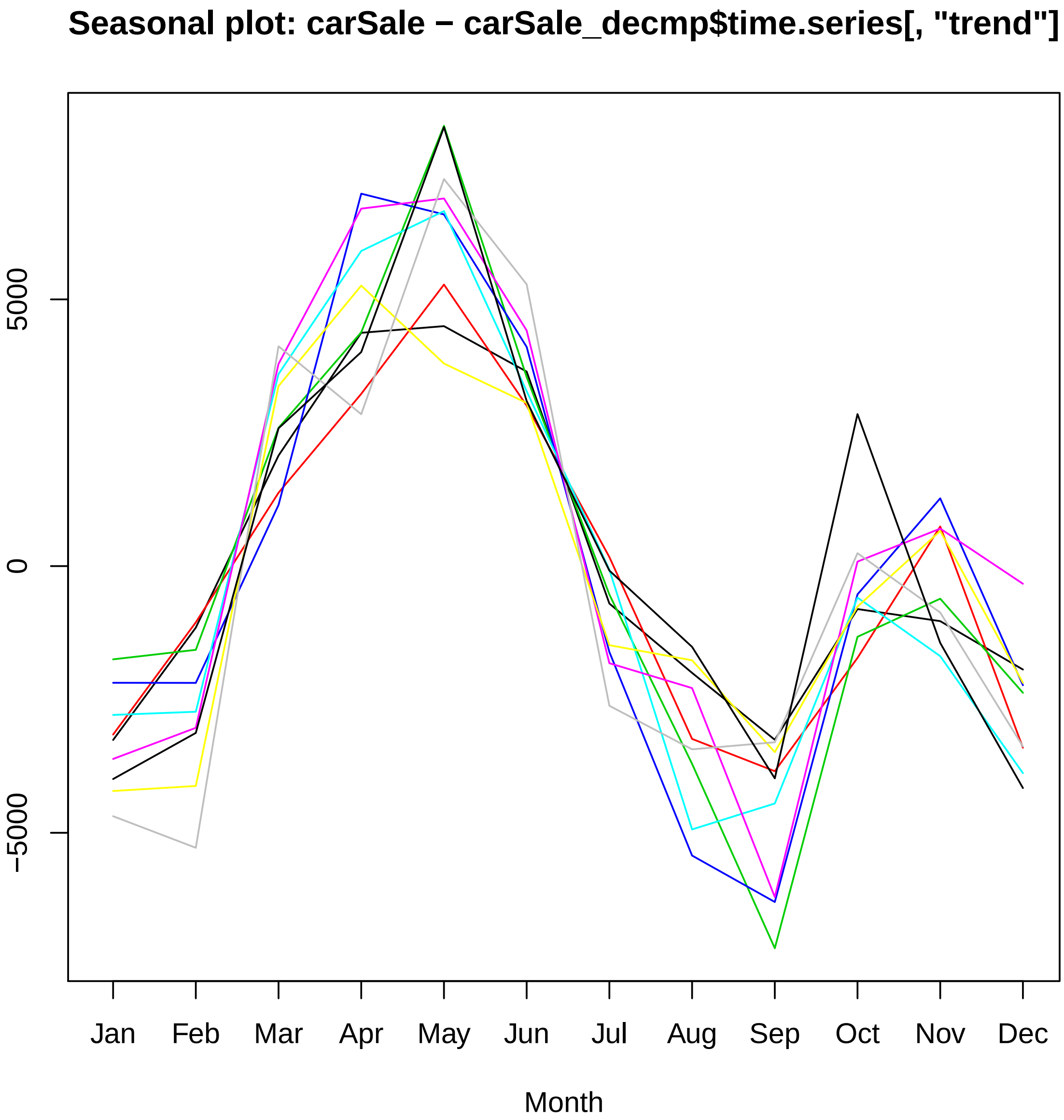
```
acf(carSale,lwd = 2 , main = "ACF::carSale")
```



```
#The time series is not stationary and has periodicity and trend.

carSale_decmp <-stl(carSale, s.window = "periodic", robust = T)
plot(carSale_decmp)
```

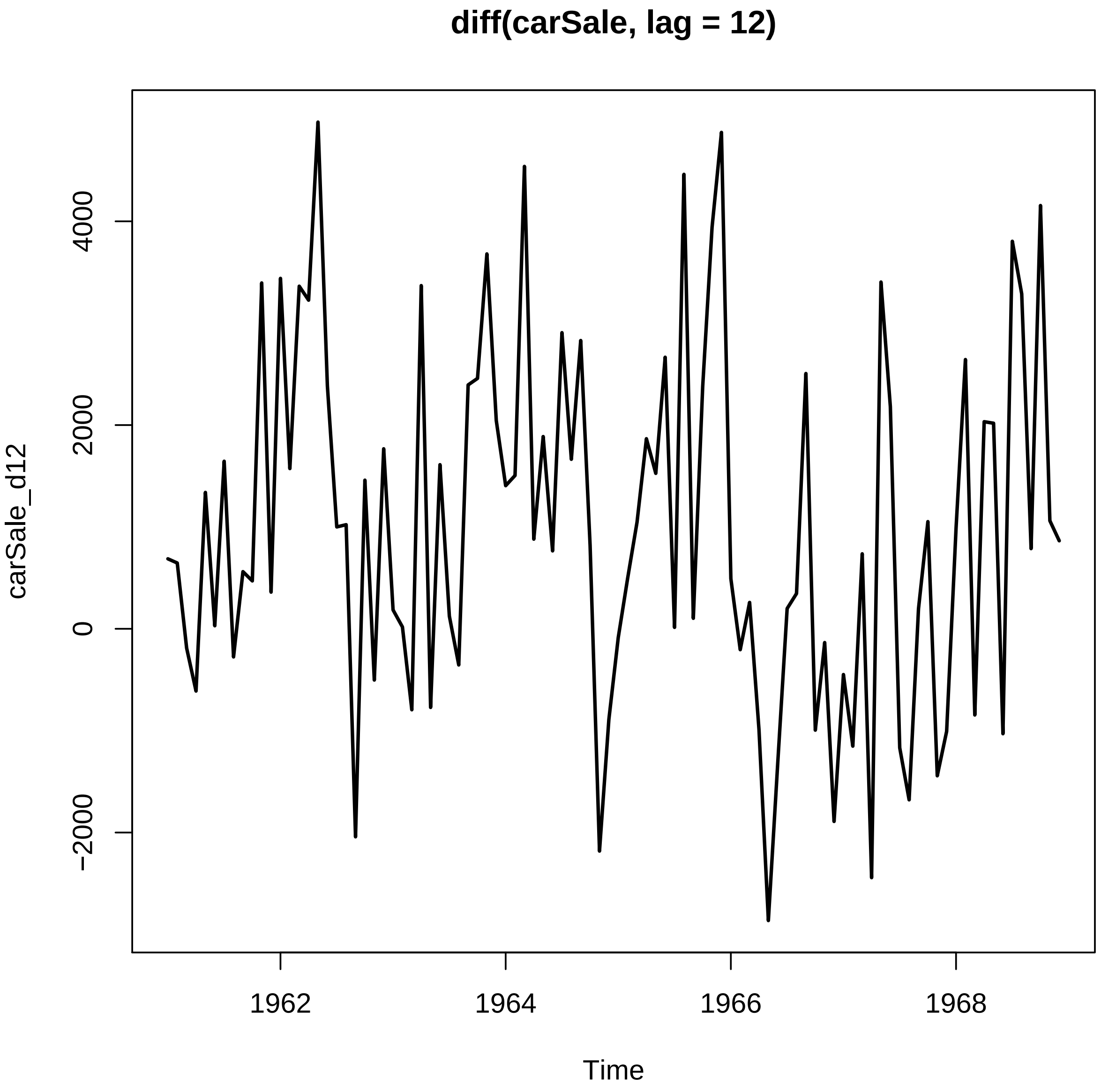


```
seasonplot(carSale-carSale_decmp$time.series[,"trend"],s = 12, col = 1:12, type = "l")
```

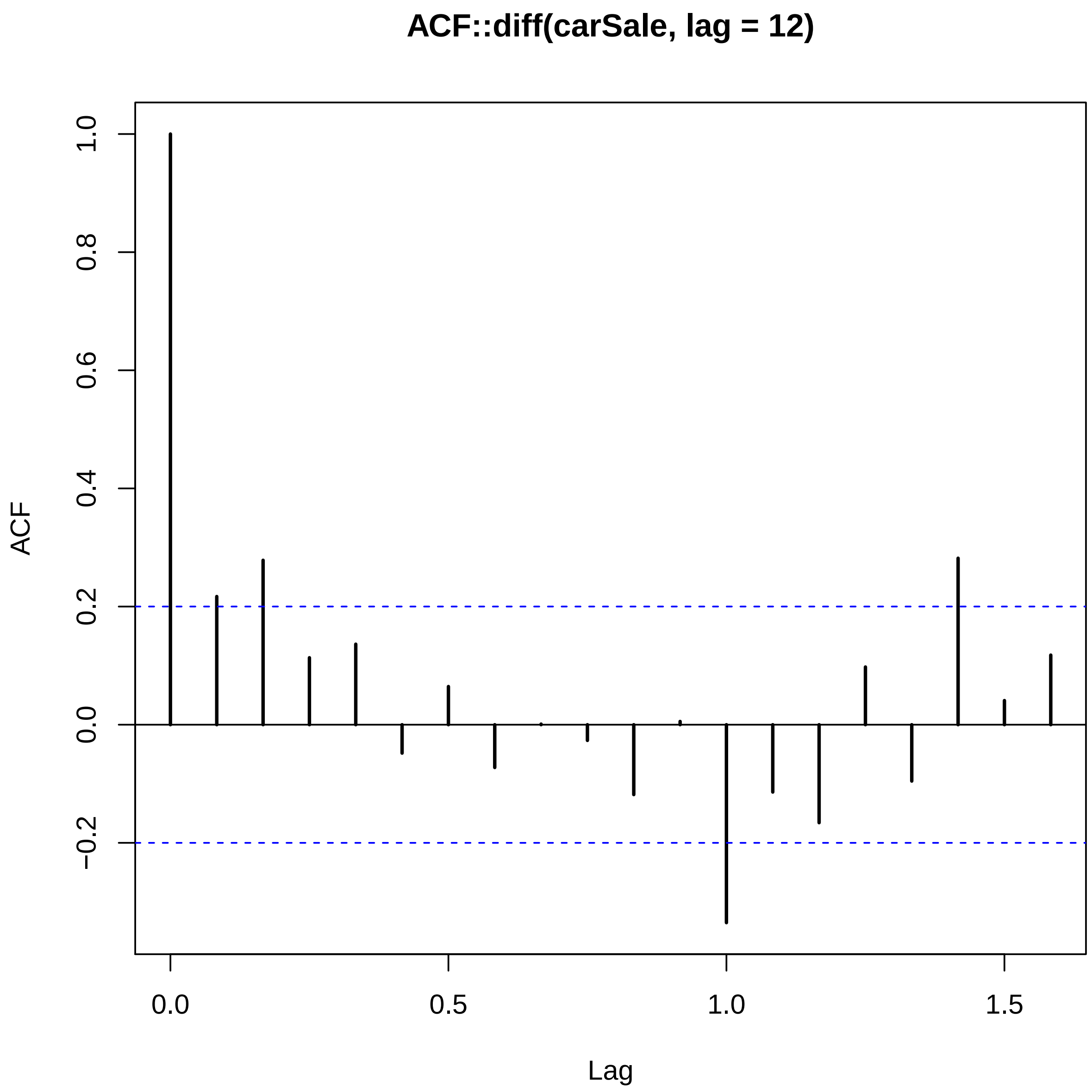


There is seasonal component and trend, thus, use SARIMA model 2.First fit the SARIMA model.

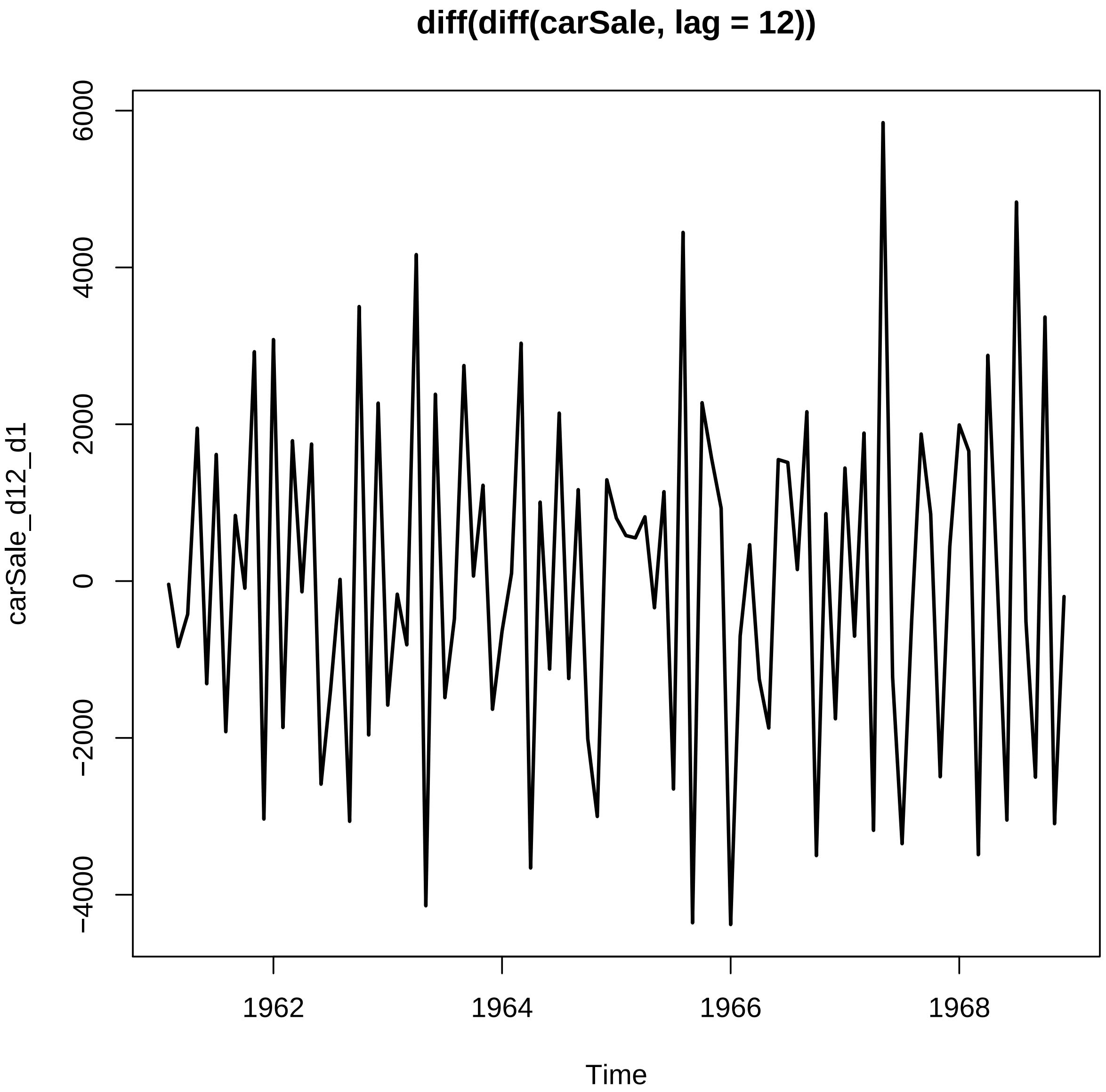
```
carSale_d12 = diff(carSale, lag = 12)
plot(carSale_d12,lwd = 2, main = "diff(carSale, lag = 12)")
```



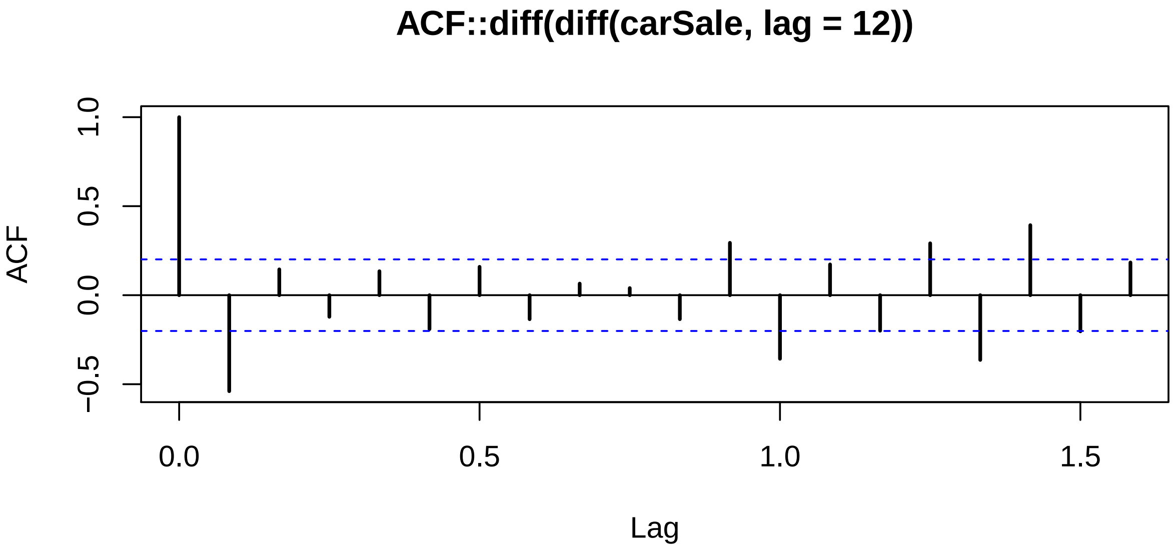
```
acf(carSale_d12,lwd = 2, main ="ACF::diff(carSale, lag = 12)")
```

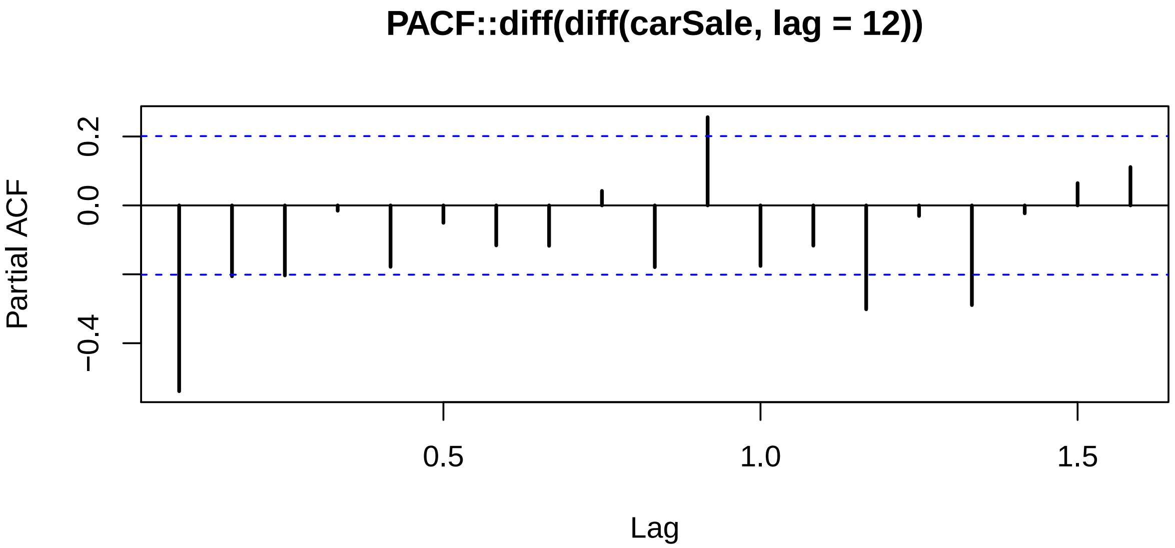


```
#carSale_d12 is not stationary, consider diff(carSale_d12)

carSale_d12_d1 = diff(carSale_d12, lag = 1)
plot(carSale_d12_d1,lwd = 2, main = "diff(diff(carSale, lag = 12))")
```



```
par(mfrow=c(2,1))
acf(carSale_d12_d1,lwd = 2, main = "ACF::diff(diff(carSale, lag = 12))")
pacf(carSale_d12_d1,lwd = 2, main = "PACF::diff(diff(carSale, lag = 12))")
```





From acf plot ,q $\leq 1, andfrompartial - acfplot, p \leq 2withd = D = 1andP \leq 1, Q \leq 1$

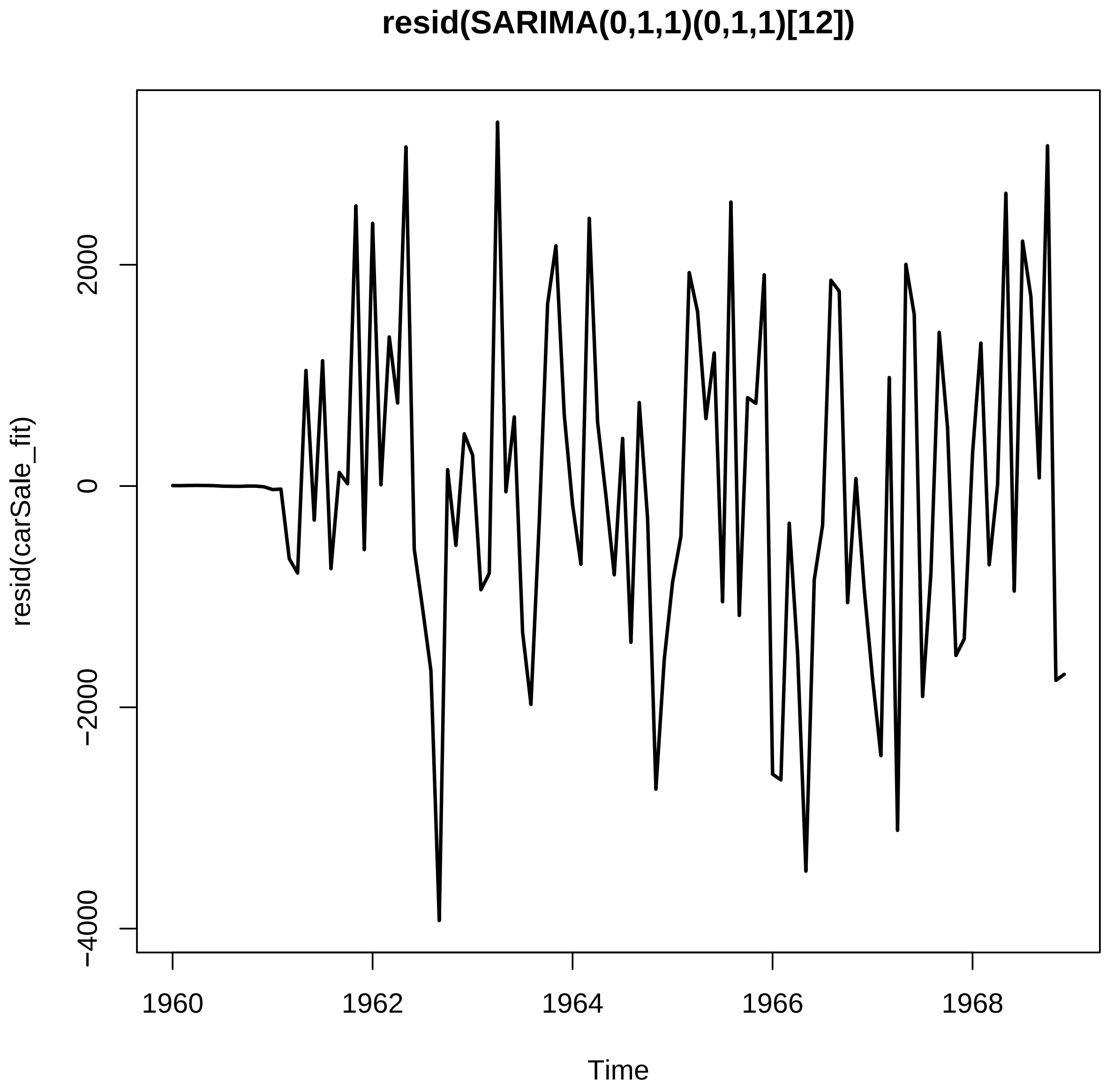
```
AIC_best <- 10**6
for (p in 0:2){
  for (q in 0:1){
    for (P in 0:1){
      for (Q in 0:1){
        fit_sarima <- Arima(carSale, order = c(p,1,q),seasonal = c(P,1,Q))
        if (fit_sarima$aic < AIC_best){
          AIC_best <- fit_sarima$aic
          cat("p = ",p,", q = ",q,",P = ",P,",Q = ",Q,"\t AIC = ",AIC_best,"\n")
        }
      }
    }
  }
}

## p =  0 , q =  0 ,P =  0 ,Q =  0   AIC =  1734.779
## p =  0 , q =  0 ,P =  0 ,Q =  1   AIC =  1712.686
## p =  0 , q =  1 ,P =  0 ,Q =  0   AIC =  1694.382
## p =  0 , q =  1 ,P =  0 ,Q =  1   AIC =  1676.588
```

The lowest AIC gives the best fitted model, which is SARIMA(0,1,1)(0,1,1)[12]

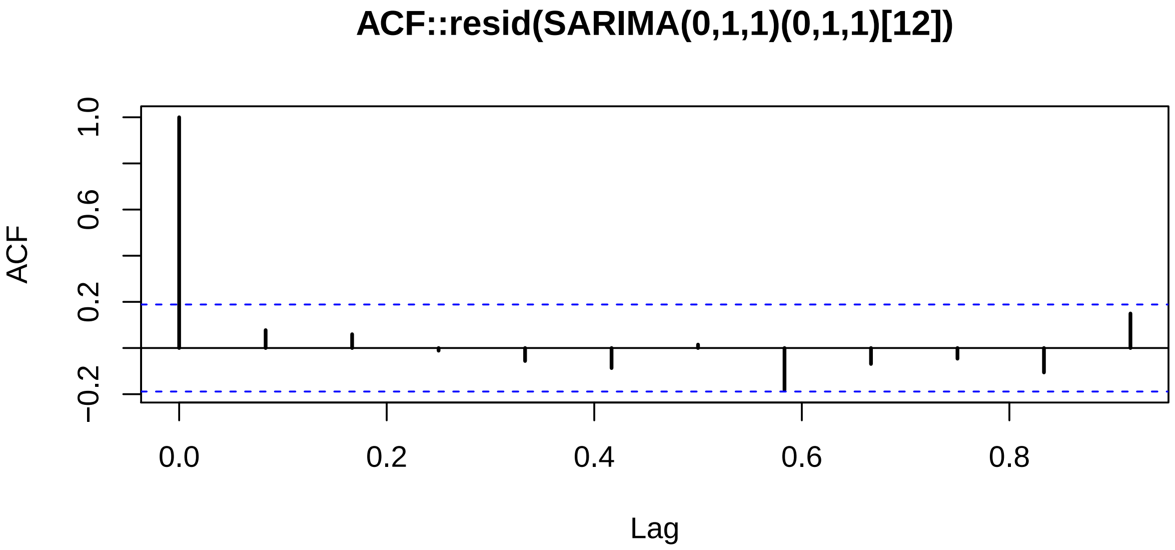
```
carSale_fit <- Arima(carSale, order = c(0,1,1), seasonal = c(0,1,1))
```

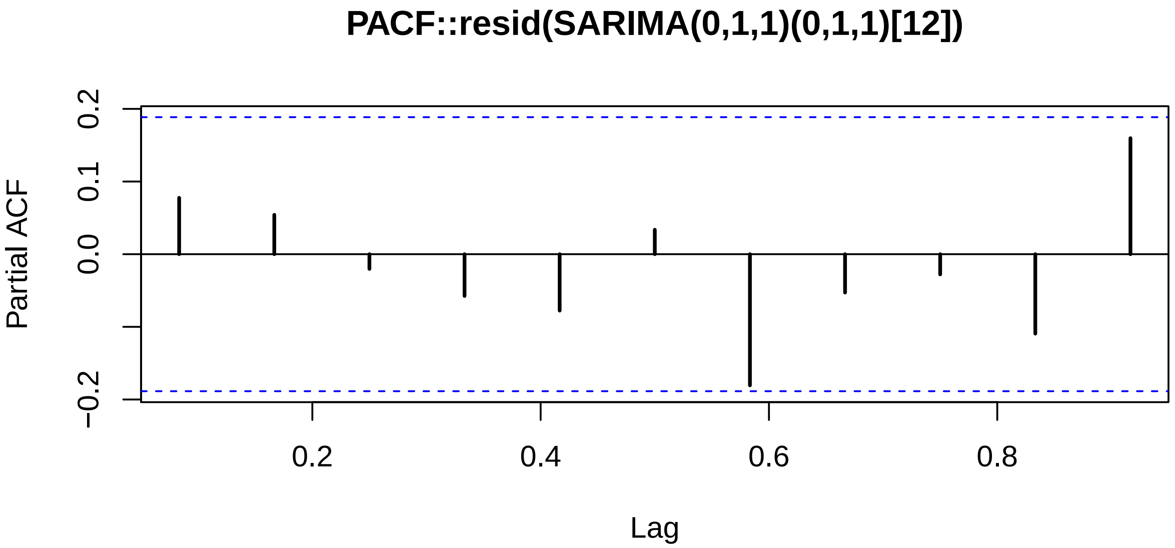
3. Then consider the residuals of the SARIMA model.

```
par(mfrow=c(1,1))
plot(resid(carSale_fit),lwd=2, main="resid(SARIMA(0,1,1)(0,1,1)[12])")
```

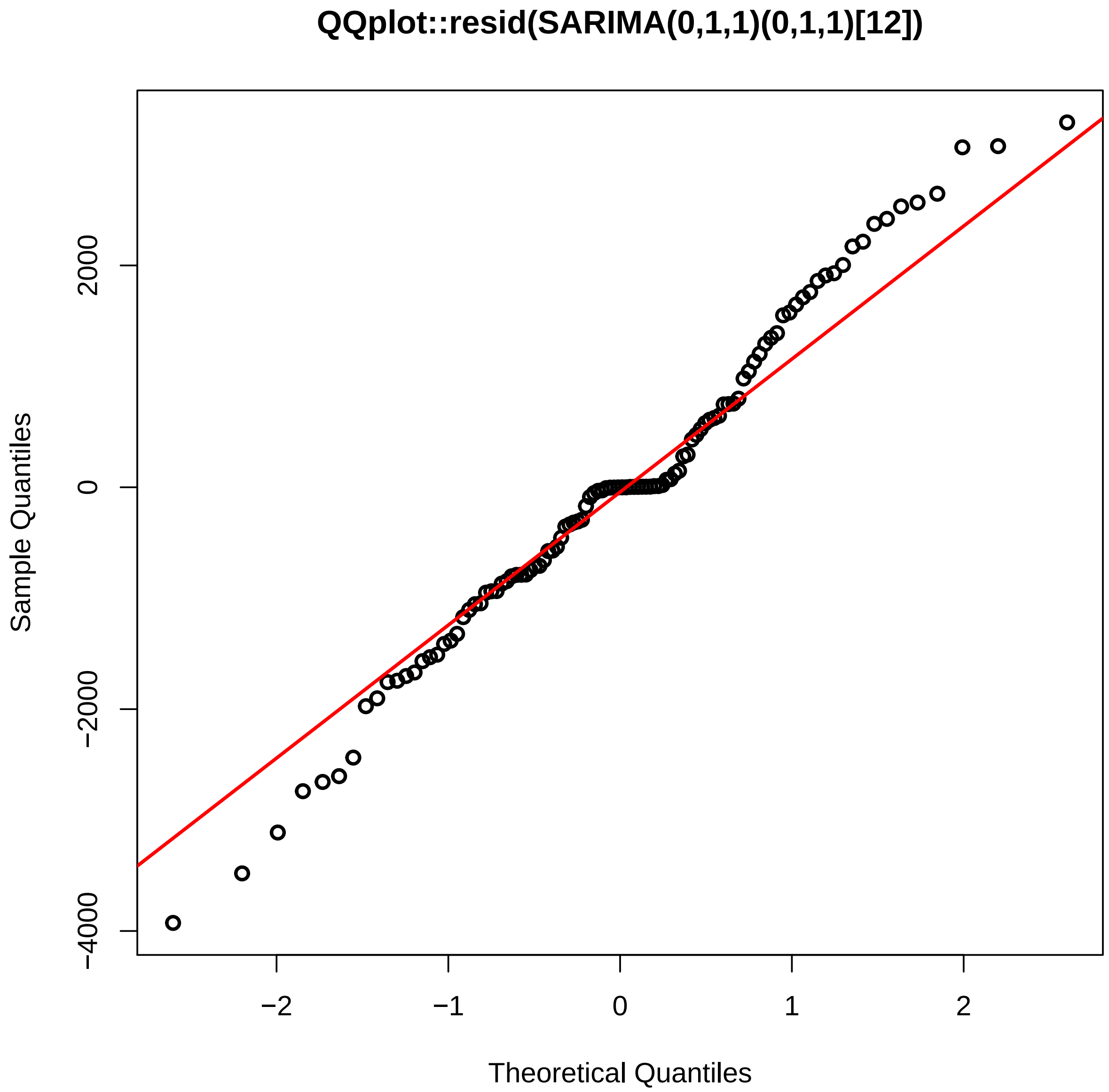


```r
par(mfrow=c(2,1))
acf(resid(carSale_fit),lwd=2, main="ACF::resid(SARIMA(0,1,1)(0,1,1)[12])",lag.max = 11)
pacf(resid(carSale_fit),lwd=2, main="PACF::resid(SARIMA(0,1,1)(0,1,1)[12])",lag.max = 11)
```





```
#The residual is stationary.
par(mfrow=c(1,1))
qqnorm(resid(carSale_fit),lwd=2, main="QQplot::resid(SARIMA(0,1,1)(0,1,1)[12])")
qqline(resid(carSale_fit), lwd=2, col="red")
```



```
shapiro.test(resid(carSale_fit))

##
##  Shapiro-Wilk normality test
##
## data:  resid(carSale_fit)
## W = 0.98601, p-value = 0.3211
```

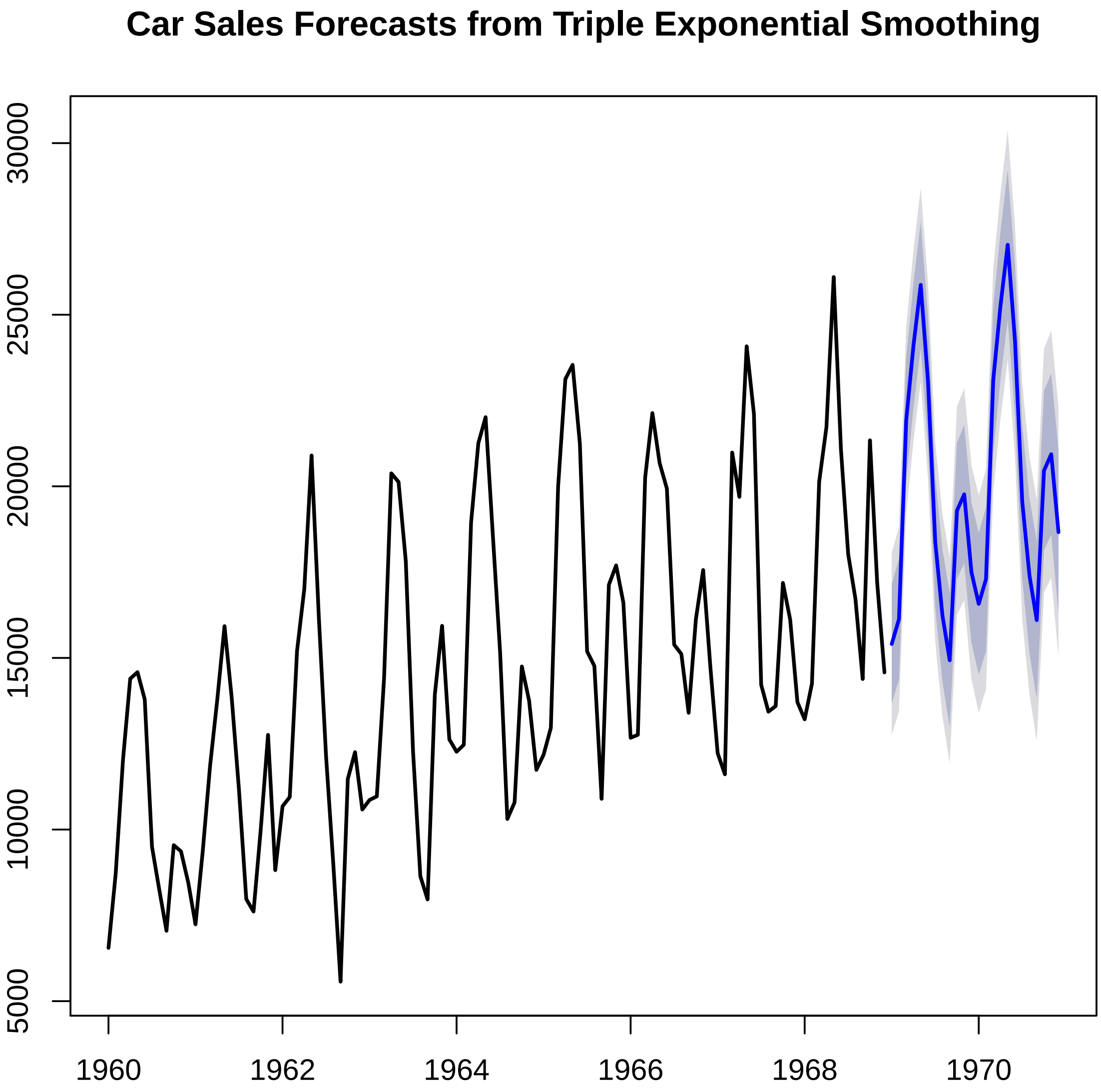
From the qq-plot and Shapiro test(p-value = 0.3211¿0.05), the residual can be regard as a gaussian distribution. So, SARIMA(0,1,1)(0,1,1)[12] is a good model. 4. Another method is using Triple exponential

smoothing

```r
TES_fit <- hw(carSale, initial = "optimal", seasonal = "additive", h = 2*12)
TES_fit
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 1969       15404.76 13672.55 17136.96 12755.58 18053.94
## Feb 1969       16126.19 14365.75 17886.62 13433.83 18818.54
## Mar 1969       21926.35 20137.72 23714.98 19190.87 24661.83
## Apr 1969       24093.96 22277.16 25910.76 21315.40 26872.52
## May 1969       25871.41 24026.46 27716.35 23049.81 28693.00
## Jun 1969       23075.58 21202.51 24948.64 20210.97 25940.19
## Jul 1969       18388.86 16487.69 20290.03 15481.28 21296.45
## Aug 1969       16231.14 14301.88 18160.40 13280.60 19181.69
## Sep 1969       14931.16 12973.82 16888.50 11937.67 17924.65
## Oct 1969       19290.16 17304.74 21275.57 16253.73 22326.59
## Nov 1969       19768.70 17755.22 21782.19 16689.35 22848.06
## Dec 1969       17494.59 15453.01 19536.17 14372.26 20616.91
## Jan 1970       16572.93 14503.28 18642.58 13407.67 19738.18
## Feb 1970       17294.36 15196.64 19392.08 14086.17 20502.54
## Mar 1970       23094.52 20968.72 25220.32 19843.39 26345.65
## Apr 1970       25262.13 23108.24 27416.02 21968.04 28556.22
## May 1970       27039.58 24857.59 29221.56 23702.52 30376.64
## Jun 1970       24243.75 22033.65 26453.85 20863.70 27623.80
## Jul 1970       19557.03 17318.81 21795.26 16133.97 22980.10
## Aug 1970       17399.31 15132.95 19665.68 13933.21 20865.42
## Sep 1970       16099.33 13804.80 18393.86 12590.15 19608.51
## Oct 1970       20458.33 18135.62 22781.04 16906.05 24010.61
## Nov 1970       20936.87 18585.96 23287.79 17341.47 24532.28
## Dec 1970       18662.76 16283.59 21041.92 15024.14 22301.37
```

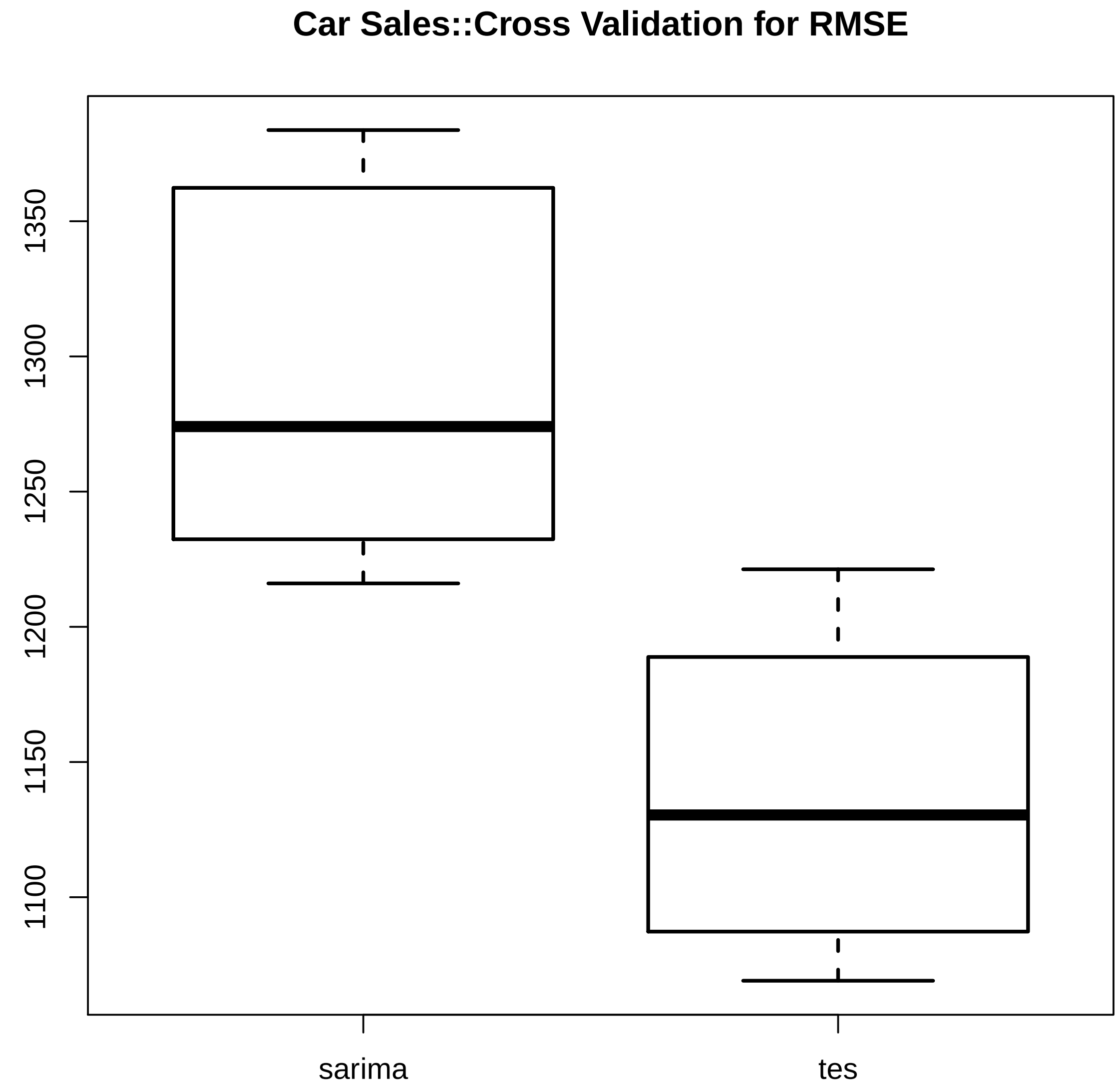
```r
plot(TES_fit,main = "Car Sales Forecasts from Triple Exponential Smoothing", lwd = 2)
```



5. Use cross-validation to compare these two models

```
CV <-  function(time_series, start, forecast_length,ts_model){
  ts_length <-  length(time_series)
  accuracy_list = c()
  for(k in start:(ts_length - forecast_length)){
    fitted_model <- ts_model(ts(time_series[0:k],frequency = 12))
    RMSE <-  accuracy(forecast(fitted_model, h = forecast_length))[2]
    accuracy_list = c(accuracy_list, RMSE)
  }
  return(accuracy_list)
}
```

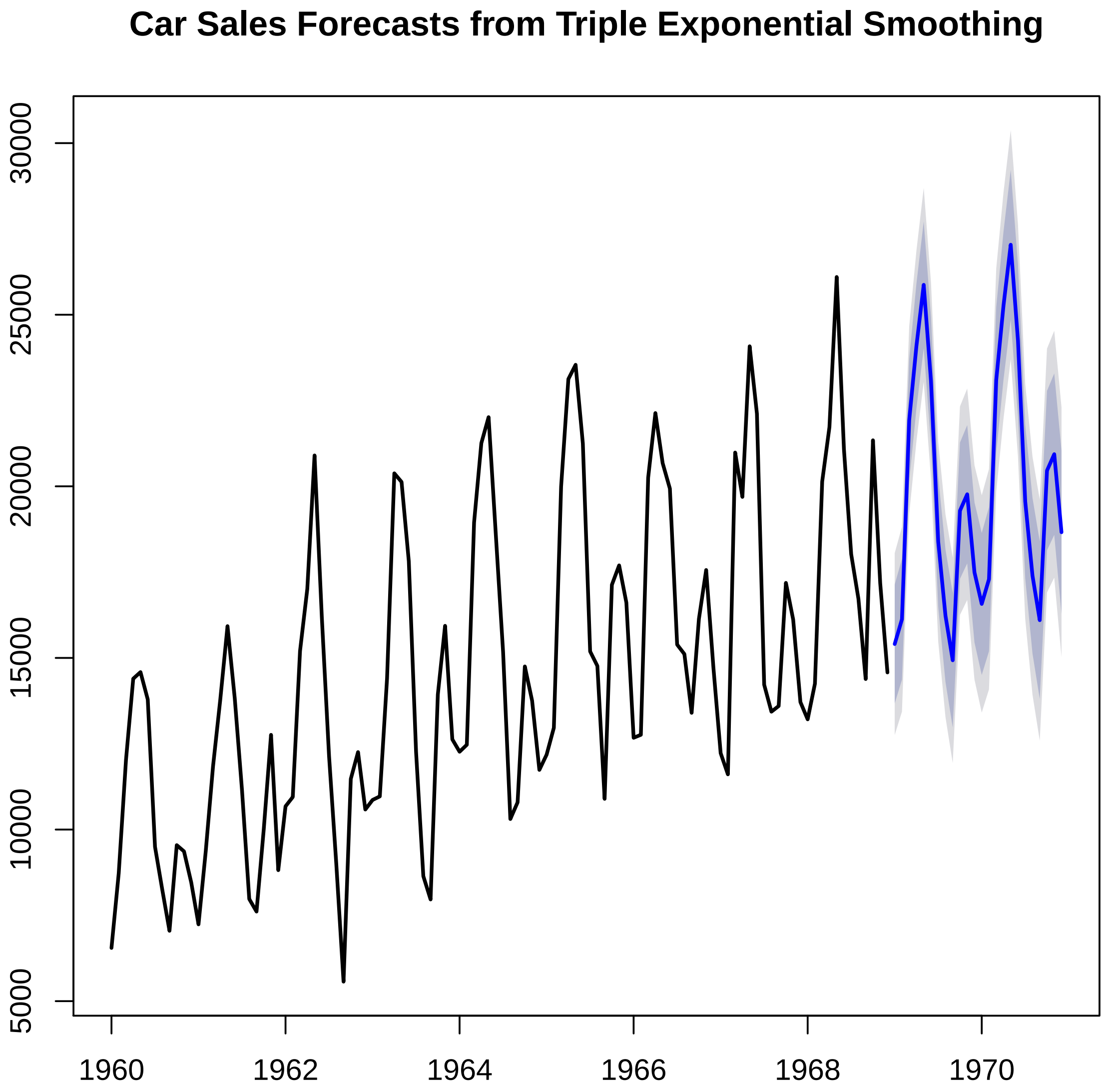
```r
model_SARIMA <- function(ts)
  return(Arima(ts, order = c(0,1,1), seasonal = c(0,1,1)))
model_TES <- function(ts)
  return(hw(ts,initial = "optimal", seasonal = "additive"))

start <- 60
forecast_length <- 2*12
CV_carSale <- data.frame(
  sarima = CV(carSale, start, forecast_length, model_SARIMA),
  tes = CV(carSale, start, forecast_length, model_TES)
)

boxplot(CV_carSale,main = "Car Sales::Cross Validation for RMSE", lwd=2)
```



From boxplot, TES gives a better prediction due to the lower RMSE. 6. Forecast the number of birth during the two weeks by using triple exponential smoothing.

```
TES_fit <- hw(carSale, initial = "optimal", seasonal = "additive", h = 2*12)
TES_fit

##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 1969       15404.76 13672.55 17136.96 12755.58 18053.94
## Feb 1969       16126.19 14365.75 17886.62 13433.83 18818.54
## Mar 1969       21926.35 20137.72 23714.98 19190.87 24661.83
## Apr 1969       24093.96 22277.16 25910.76 21315.40 26872.52
## May 1969       25871.41 24026.46 27716.35 23049.81 28693.00
## Jun 1969       23075.58 21202.51 24948.64 20210.97 25940.19
## Jul 1969       18388.86 16487.69 20290.03 15481.28 21296.45
## Aug 1969       16231.14 14301.88 18160.40 13280.60 19181.69
## Sep 1969       14931.16 12973.82 16888.50 11937.67 17924.65
## Oct 1969       19290.16 17304.74 21275.57 16253.73 22326.59
## Nov 1969       19768.70 17755.22 21782.19 16689.35 22848.06
## Dec 1969       17494.59 15453.01 19536.17 14372.26 20616.91
## Jan 1970       16572.93 14503.28 18642.58 13407.67 19738.18
## Feb 1970       17294.36 15196.64 19392.08 14086.17 20502.54
## Mar 1970       23094.52 20968.72 25220.32 19843.39 26345.65
## Apr 1970       25262.13 23108.24 27416.02 21968.04 28556.22
## May 1970       27039.58 24857.59 29221.56 23702.52 30376.64
## Jun 1970       24243.75 22033.65 26453.85 20863.70 27623.80
## Jul 1970       19557.03 17318.81 21795.26 16133.97 22980.10
## Aug 1970       17399.31 15132.95 19665.68 13933.21 20865.42
## Sep 1970       16099.33 13804.80 18393.86 12590.15 19608.51
## Oct 1970       20458.33 18135.62 22781.04 16906.05 24010.61
## Nov 1970       20936.87 18585.96 23287.79 17341.47 24532.28
## Dec 1970       18662.76 16283.59 21041.92 15024.14 22301.37
```

```
plot(TES_fit,main = "Car Sales Forecasts from Triple Exponential Smoothing", lwd = 2)
```


Car Sales Forecasts from Triple Exponential Smoothing
30000
25000
20000
15000
10000
5000
1960
1962
1964
1966
1968
1970