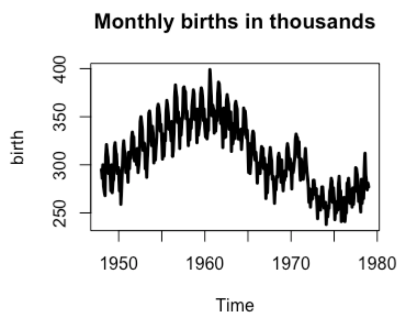
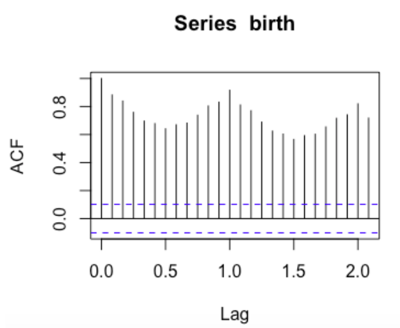
Group 31
A0102680R Ethan Koh
A0127222U Chan Yan Jia
A0124817E Lim Wei Qi
A0131386H Abigail Teo Si Min
A0105533R Wang Jiabao

Exercise 1 (Can one trust confidence intervals?)

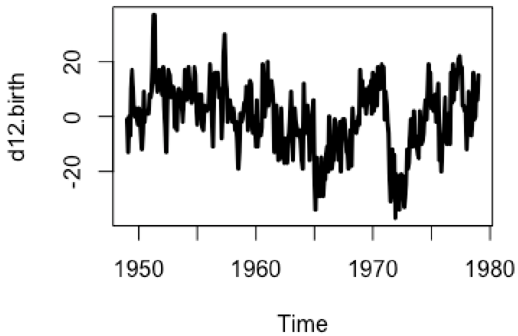
```
> setwd("/Users/yanjia/Desktop")
> load("tsa3.rda")
> par(mfrow=c(2,1))
> plot(birth, lwd=3, main="Monthly births in thousands")
> acf(birth)
```



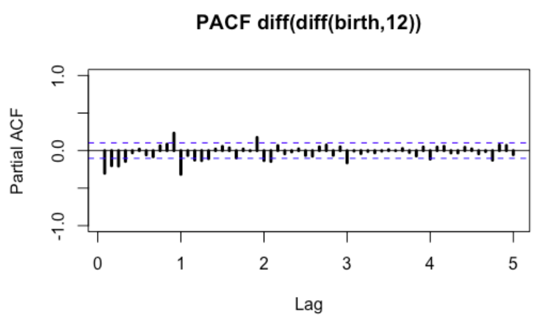


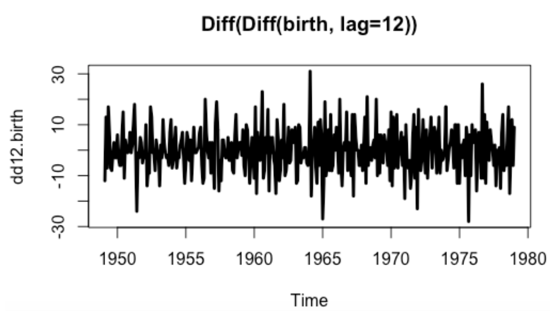
A seasonal pattern of 12 can be observed from the plot.

```
> d12.birth <- diff(birth, lag=12)
> plot(d12.birth, lwd=3)
```

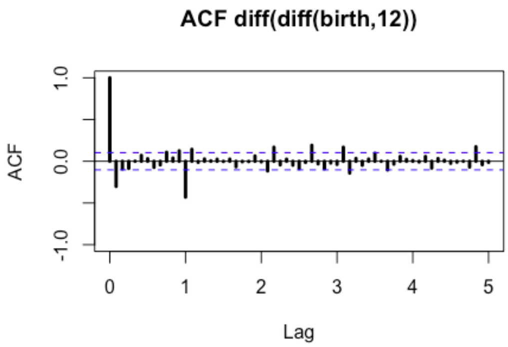


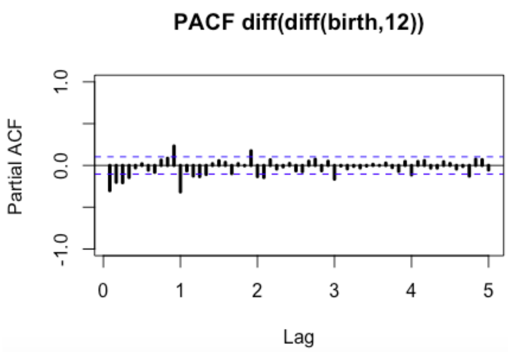

```
> # The data is not stationary. To remove this, we will have to differentiate the data.
> dd12.birth <- diff(d12.birth,lag=1)
> plot(dd12.birth, lwd=3, main="Diff(Diff(birth, lag=12))")
> #The data appears to be stationary.
> #As a result, we will let d and D be 1.
```





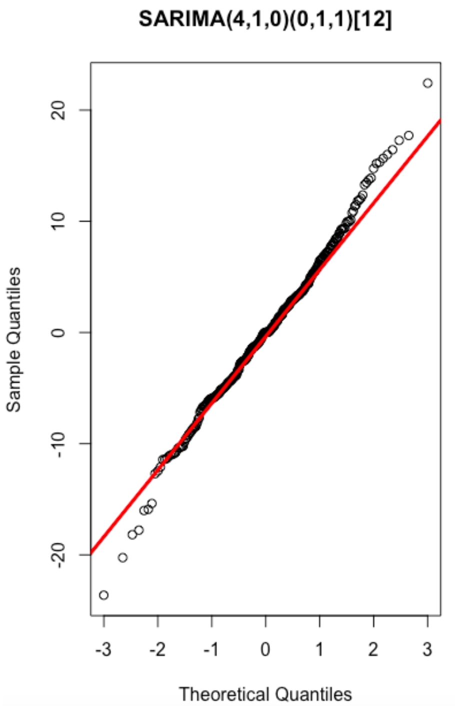

```
> acf(dd12.birth, lwd=3,main="ACF diff(diff(birth,12))", ylim=c(-1,1), lag.max=12*5)
> pacf(dd12.birth,lwd=3,main="PACF diff(diff(birth,12))", ylim=c(-1,1), lag.max=12*5)
```



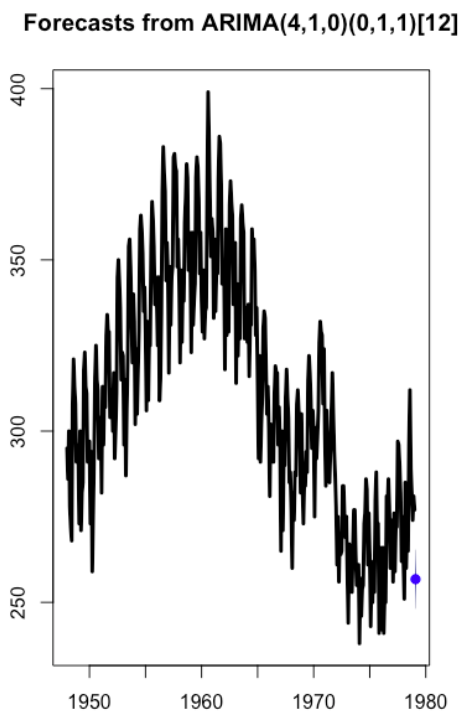



```
> #fit SARIMA, restrict models to p<=4, q<=4, P<=1, Q<=1
> AIC_best = 10**6
> for(p in 0:4){
+ for(q in 0:4){
+ for(P in c(0,1)){
+ for(Q in c(0,1)){
+ fit_sarima <- Arima(birth, order= c(p,1,q), seasonal = c(P,1,Q))
+ aic = fit_sarima$aic
+ if(aic<AIC_best){
+ AIC_best = aic
+ cat("p=",p,"q=",q,"P=",P,"Q=",Q,"\t AIC=",fit_sarima$aic,
+ "\t Number of parameters=",p+q+P+Q,"\n")
+ }
+ }
+ }
+ }
+ }
p= 0 q= 0 P= 0 Q= 0      AIC= 2621.434   Number of parameters= 0
p= 0 q= 0 P= 0 Q= 1      AIC= 2472.199   Number of parameters= 1
p= 0 q= 1 P= 0 Q= 1      AIC= 2428.557   Number of parameters= 2
p= 0 q= 2 P= 0 Q= 1      AIC= 2419.874   Number of parameters= 3
p= 0 q= 2 P= 1 Q= 1      AIC= 2419.863   Number of parameters= 4
p= 1 q= 1 P= 0 Q= 1      AIC= 2419.855   Number of parameters= 3
p= 1 q= 1 P= 1 Q= 1      AIC= 2419.66    Number of parameters= 4
p= 2 q= 3 P= 0 Q= 1      AIC= 2418.553   Number of parameters= 6
p= 2 q= 3 P= 1 Q= 1      AIC= 2418.216   Number of parameters= 7
p= 4 q= 0 P= 0 Q= 1      AIC= 2417.468   Number of parameters= 5
```

```
> #As seen in the results, we will choose the (4,1,0)(0,1,1)[12] model since it has the lowest
AIC.
> # The number of parameters are acceptable as well.
> sarima_fit <- Arima(birth, order=c(4,1,0), seasonal=c(0,1,1))
> tsdisplay(resid(sarima_fit), lwd=3, main=" Residual plot for model SARIMA(4,1,0)(0,1,1)[12]")
> qqnorm(resid(sarima_fit), main="SARIMA(4,1,0)(0,1,1)[12]")
> qqline(resid(sarima_fit), col="red", lwd=3)
```




```
> # The residual plot seems to show that the residuals are normally distributed.
> # This model appears to be a good fit to the data.
> #part b)
> plot(forecast(sarima_fit, h=1, level=c(0:80)), lwd=3)
```

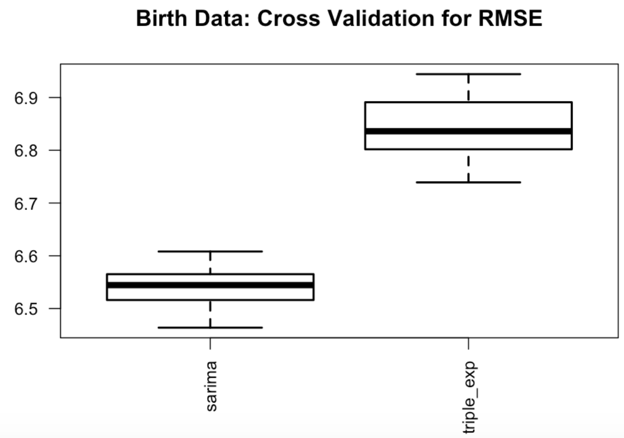



```
> # Holt's winter algorithm (Triple exponential smoothing) can be used as well.
> # Thus we will compare this model to the TES.
> cross_validation = function(time_series, start, forecast_length, ts_model){
+ #INPUT:
+ #======
+ #time_series: a time series
+ #start: minimum amount of data used for fitting a model
+ #forecast_length: number of forecast in the future
+ #ts_model: a function that takes a time_series as input and output a fitted model
+ ts_length = length(time_series)
+ accuracy_list = c()
+ for(k in c(start:(ts_length - forecast_length))){
+ #fit the model on data from 0 to "k"
+ fitted_model = ts_model(ts(time_series[0:k], frequency=12))
+ #extract Root Mean Square Error of prediction on the next "h" values
+ RMSE = accuracy(forecast(fitted_model, h = forecast_length))[2]
+ accuracy_list = c(accuracy_list, RMSE)
+ }
+ return( accuracy_list )
+ }
```

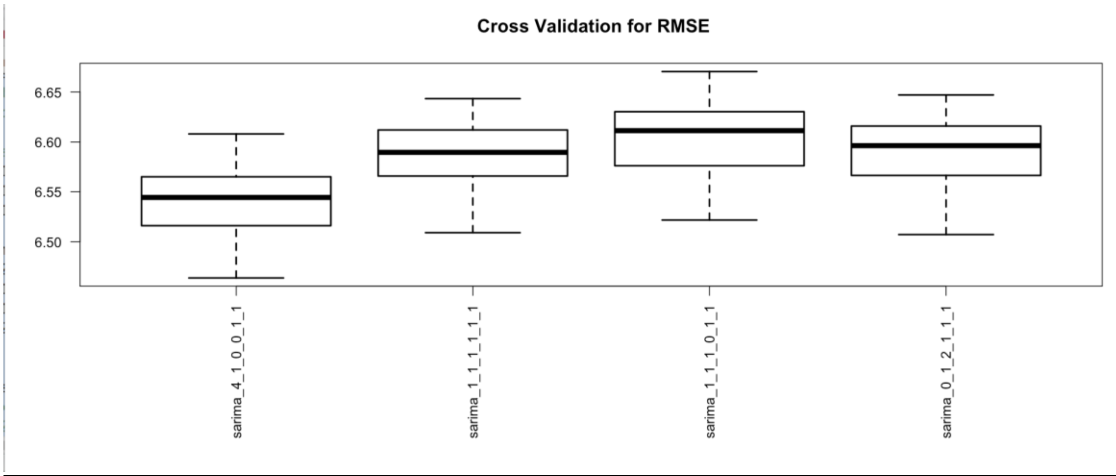
```
> ts_model_sarima = function(tseries)
+ return(Arima(tseries, order = c(4,1,0), seasonal=c(0,1,1), include.drift=F))
> ts_model_triple = function(tseries)
+ return( hw(tseries, initial="optimal", seasonal="additive") )
> start = 300
> forecast_length = 1
> CV_results = data.frame(
+ sarima = cross_validation(birth, start, forecast_length, ts_model_sarima),
+ triple_exp = cross_validation(birth, start, forecast_length, ts_model_triple)
+ )
```


```
> ts_model_sarima = function(tseries)
+ return(Arima(tseries, order = c(4,1,0), seasonal=c(0,1,1), include.drift=F))
> ts_model_triple = function(tseries)
+ return( hw(tseries, initial="optimal", seasonal="additive") )
> start = 300
> forecast_length = 1
> CV_results = data.frame(
+ sarima = cross_validation(birth, start, forecast_length, ts_model_sarima),
+ triple_exp = cross_validation(birth, start, forecast_length, ts_model_triple)
+ )
```


```
> boxplot(CV_results,las=2,cex.axis=0.9,main = "Birth Data: Cross Validation for RMSE", lwd=2)
```



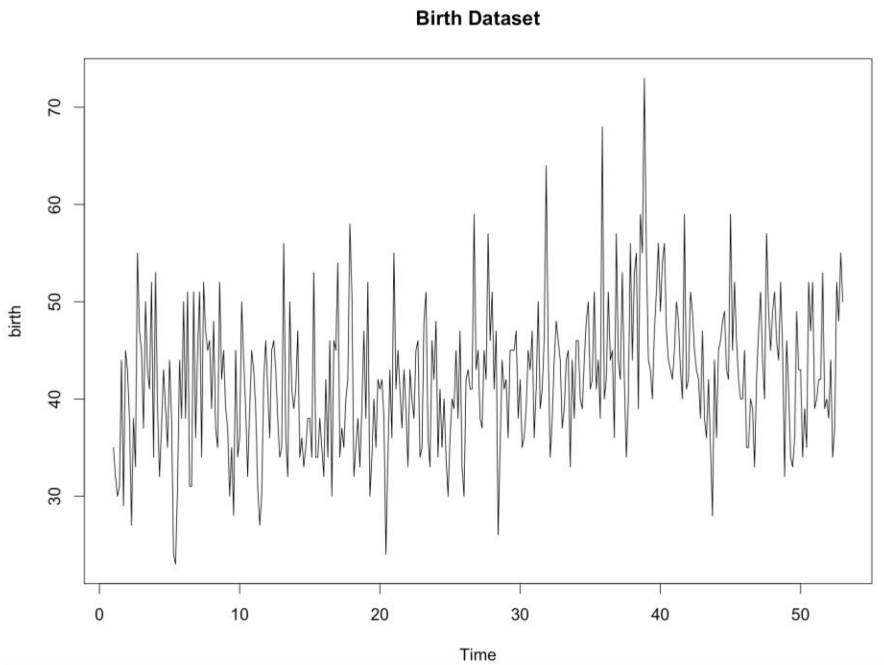

```
> # As seen in the box plot diagram, Our SARIMA model is better than the Triple exponential smoothing.
> # This is because the SARIMA model has a lower RMSE.
> # Just to consider the rest of the SARIMA models to see which one could possibly be a better fit.
> sarima_111_111 <- function(tseries) return(Arima(tseries, order=c(1,1,1),seasonal=c(1,1,1), include.drift=F))
> sarima_111_011 <- function(tseries) return(Arima(tseries, order=c(1,1,1),seasonal=c(0,1,1), include.drift=F))
> sarima_012_111 <- function(tseries) return(Arima(tseries, order=c(0,1,2),seasonal=c(1,1,1), include.drift=F))
> CV = data.frame(
+ sarima_4_1_0_0_1_1 = cross_validation(birth, start, forecast_length, ts_model_sarima),
+ sarima_1_1_1_1_1_1 = cross_validation(birth, start, forecast_length, sarima_111_111),
+ sarima_1_1_1_0_1_1 = cross_validation(birth, start, forecast_length, sarima_111_011),
+ sarima_0_1_2_1_1_1 = cross_validation(birth, start, forecast_length, sarima_012_111)
+ )

> boxplot(CV,las=2,cex.axis=0.9,main = " Cross Validation for RMSE", lwd=2)
```



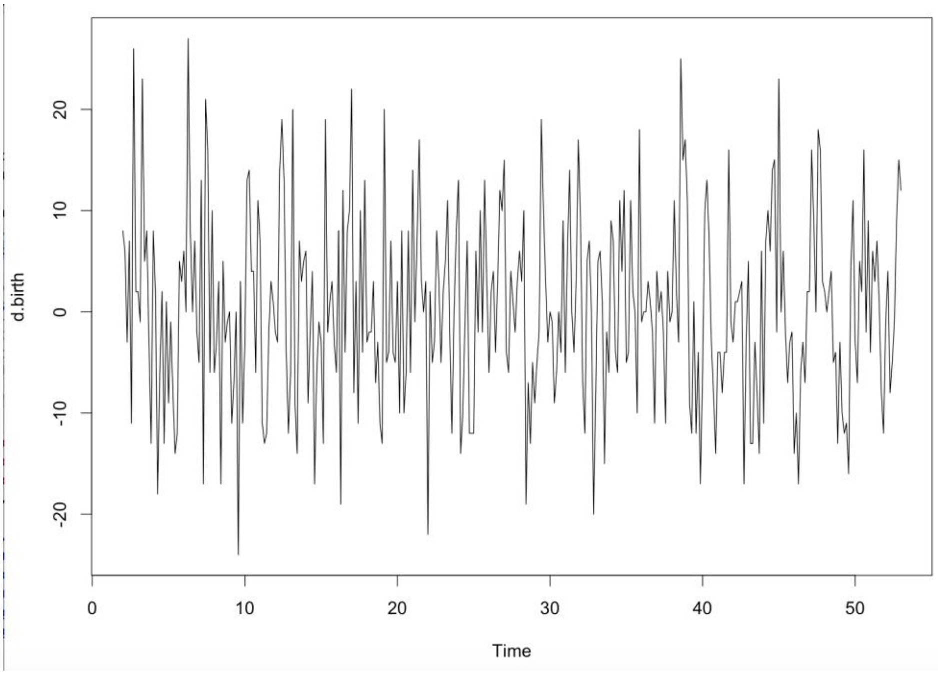
> # From the box plots we can see that the SAIMA (4,1,0)(0,1,1) [12] model is still a better model based of RMSE.
> # Hence the 80& confidence interval generated from the previous plot is reliable.

Exercise 2 (Number of Births in California?)

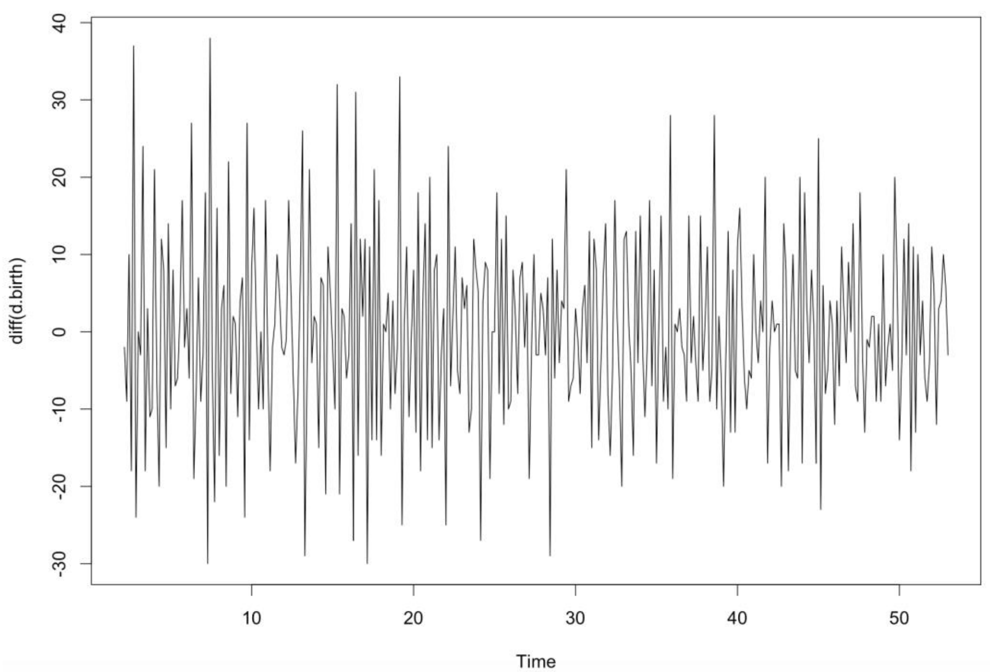
```
> birth=ts(birth_data$Daily.total.female.births.in.California,frequency=7)
> ts.plot(birth,main="Birth Dataset")
# plot the time series, a trend can be observed, a seasonal pattern may not be present.
```



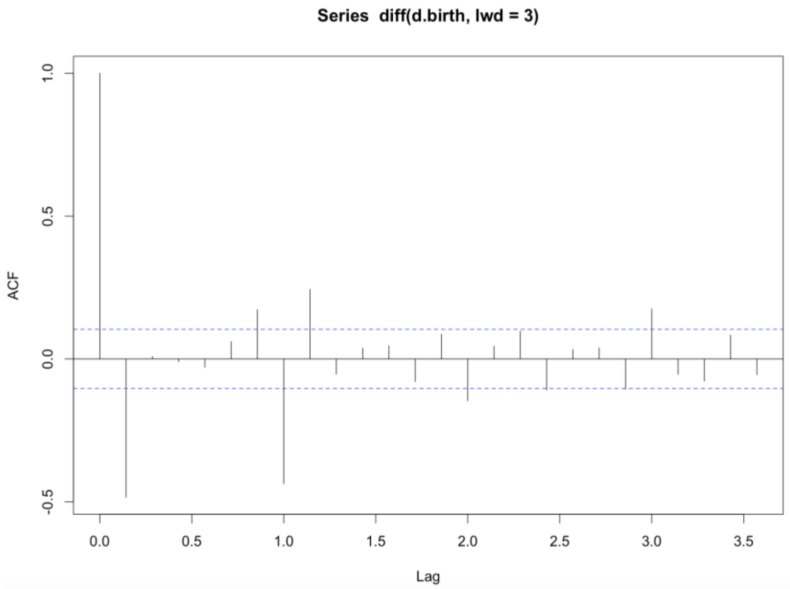

```
> d.birth<-diff(birth,lag=7)
> ts.plot(d.birth)
# differentiate once to remove the trend
```

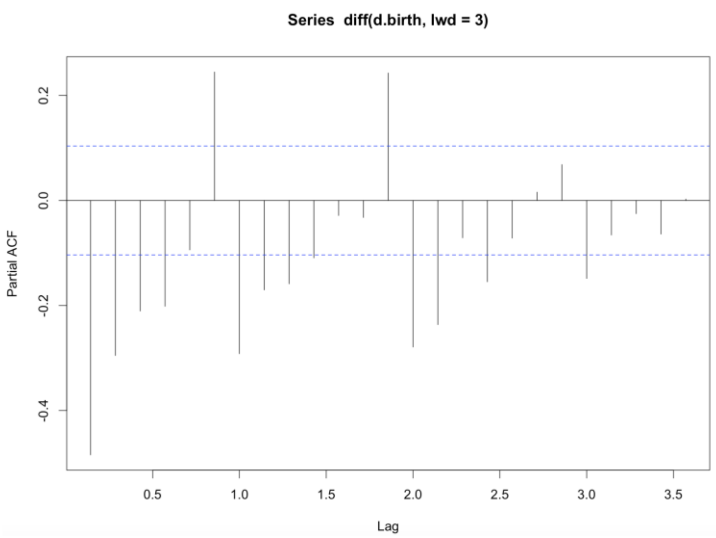


#Seems to have some seasonal pattern, so differentiate once more
> ts.plot(diff(d.birth))



> acf(diff(d.birth,lwd=3))
> pacf(diff(d.birth,lwd=3))
#plot the ACF and PACF.
#ACF goes to 0; first 2 coefficient is high. Possible model: MA(2)
#PACF goes to 0; first 2 coefficients are high. Possible model: AR(2)





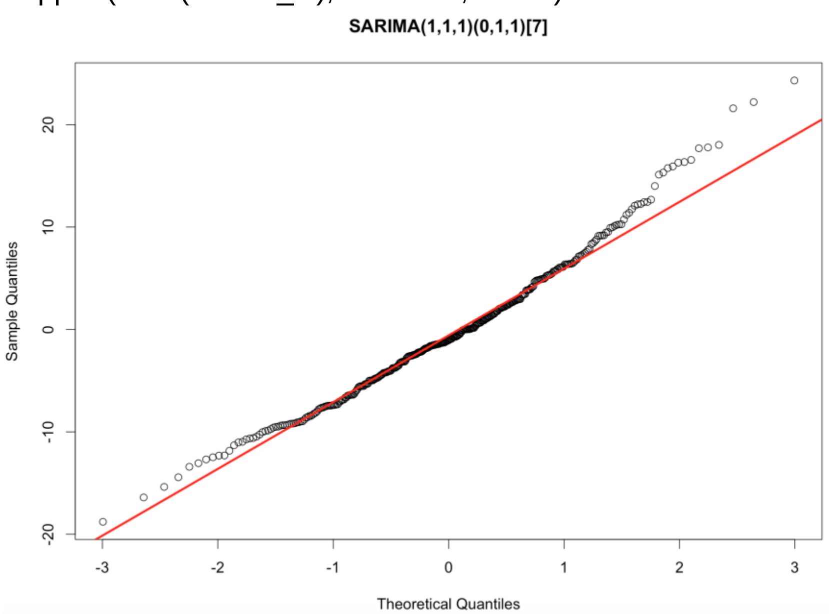

```
# Using AIC to fit a SARIMA model
# we restrict p,q<= 2, P,Q<= 1
>AIC_best = 10**6
>for(p in 0:2){
>  for(q in 0:2){
>    for(P in c(0,1)){
>      for(Q in c(0,1)){
        fit_sarima = Arima(birth, order = c(p,1,q), seasonal = c(P,1,Q))
        aic = fit_sarima$aic
        if(aic < AIC_best){
          AIC_best = aic
```

```
        cat("p=",p,"q=",q,"P=",P,"Q=",Q,"\t AIC=",fit_sarima$aic, "\t Number of parameters=",p+q+P+q,"\n")
      }
    }
  }
 }
}
```

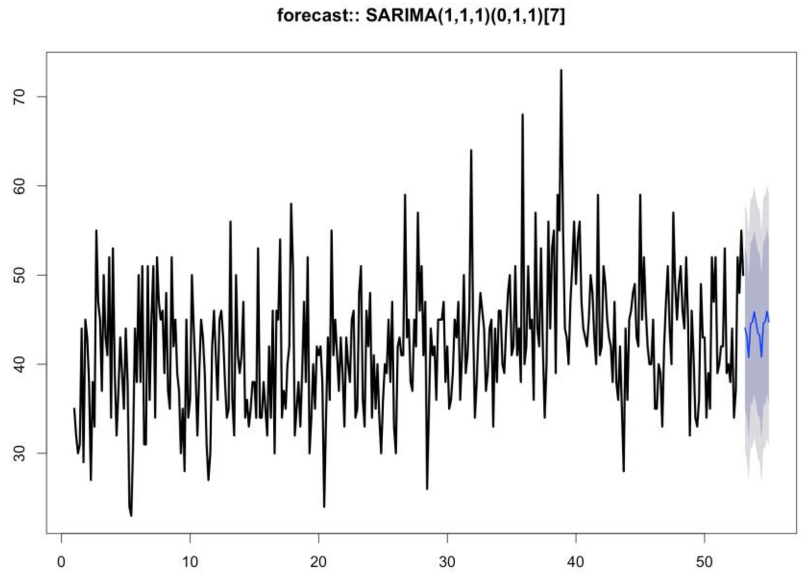
```
p= 0 q= 0 P= 0 Q= 0      AIC= 2806.298   Number of parameters= 0
p= 0 q= 0 P= 0 Q= 1      AIC= 2619.012   Number of parameters= 0
p= 0 q= 1 P= 0 Q= 0      AIC= 2611.651   Number of parameters= 2
p= 0 q= 1 P= 0 Q= 1      AIC= 2432.035   Number of parameters= 2
p= 0 q= 2 P= 0 Q= 1      AIC= 2431.208   Number of parameters= 4
p= 1 q= 1 P= 0 Q= 1      AIC= 2430.795   Number of parameters= 3
p= 1 q= 2 P= 0 Q= 1      AIC= 2430.387   Number of parameters= 5
```

We choose SARIMA(1,1,1)(0,1,1) as a model since it has the lowest AIC.

Checking the residuals
> sarima_fit = Arima(birth, order = c(1,1,1), seasonal = c(0,1,1))
> qqnorm(resid(sarima_fit), main="SARIMA(1,1,1)(0,1,1)[7]")
> qqline(resid(sarima_fit), col="red", lwd=3)

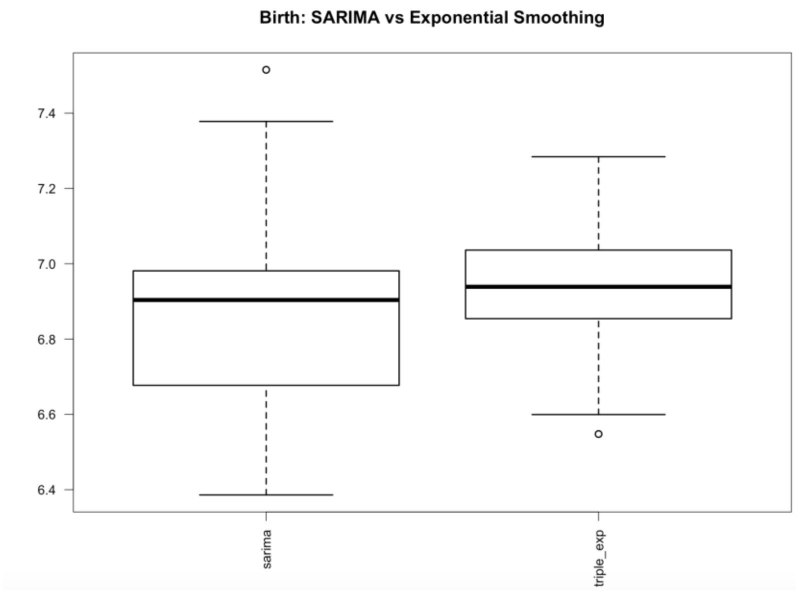


#does not look too far from Gaussian, so we can trust the forecast
> plot(forecast(sarima_fit, h=14), main="forecast:: SARIMA(1,1,1)(0,1,1)[7]", lwd=3)



#Comparing with triple exponential smoothing using cross validation

```
>cross_validation = function(time_series, start, forecast_length, ts_model){
  ts_length = length(time_series)
>  accuracy_list = c()
>  for(k in c(start:(ts_length - forecast_length))){
   fitted_model = ts_model(ts(time_series[0:k], frequency=7))
    RMSE = accuracy(forecast(fitted_model, h = forecast_length))[2]
   accuracy_list = c(accuracy_list, RMSE)
  }
>  return( accuracy_list )

> ts_model_sarima = function(tseries)
  return( Arima(tseries, order = c(1,1,1),seasonal = c(0,1,1), include.drift = F) )
> ts_model_triple = function(tseries)
  return( hw(tseries, initial = "optimal", seasonal = "additive") )
> start = 5*7
> forecast_length = 14
> CV_results = data.frame(sarima= cross_validation(birth, start, forecast_length,
ts_model_sarima),triple_exp = cross_validation(birth, start, forecast_length, ts_model_triple))
> boxplot(CV_results,las = 2, cex.axis = 0.9,main = "Birth: SARIMA vs Exponential Smoothing",
lwd=2)
```

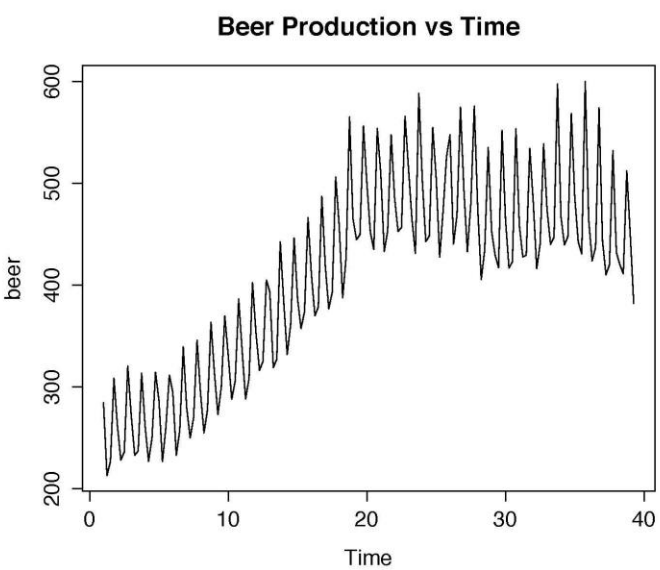


From the boxplot, we can see that the RMSE for the SARIMA model is smaller than the triple exponential smoothing, hence the forecast from a SARIMA model can be trusted.

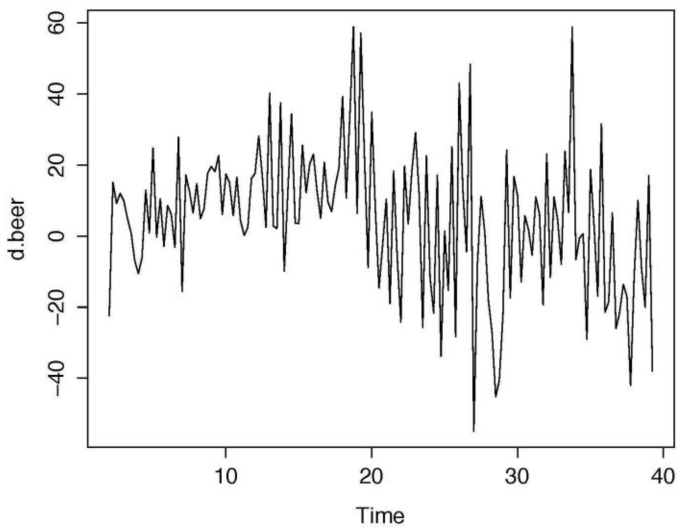
<u>Exercise 3 (How much beer?)</u>

Using AIC approach to fit a SARIMA model:

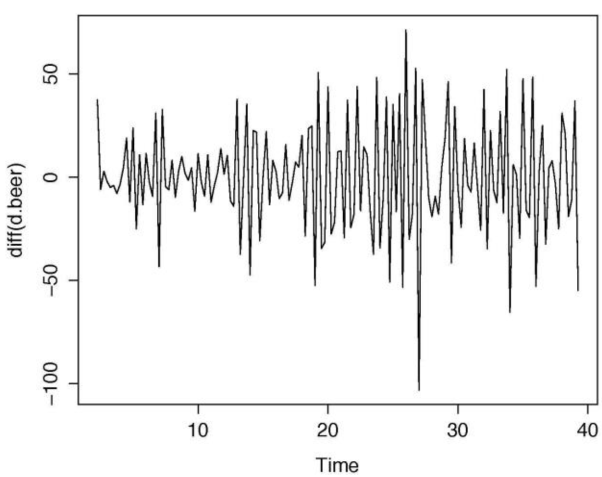
```
> library(forecast)
> beer_prod = read.csv("/Users/wjbpamela/Downloads/quarterly-beer-production-in-aus.csv")
> beer=ts(beer_prod[,2],frequency=4)
> ts.plot(beer,main = "Beer Production vs Time")
```



#differentiate the data to remove trend
```
> d.beer = diff(beer,lag=4)
> ts.plot(d.beer)
```

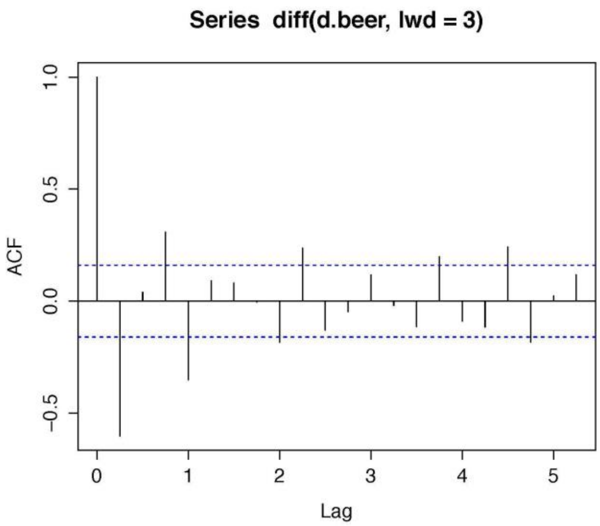


take the 2nd difference to check further whether there’s a seasonal pattern
```
> ts.plot(diff(d.beer))
```

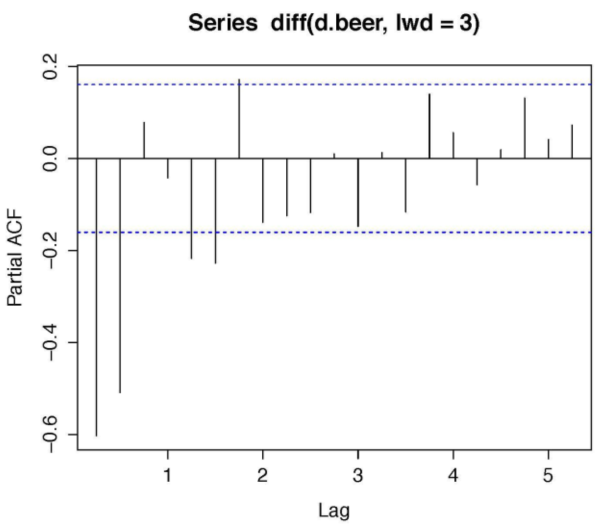


#We can see some seasonal pattern from the above graph

```
> acf(diff(d.beer,lwd=3))
```



> pacf(diff(d.beer,lwd=3))



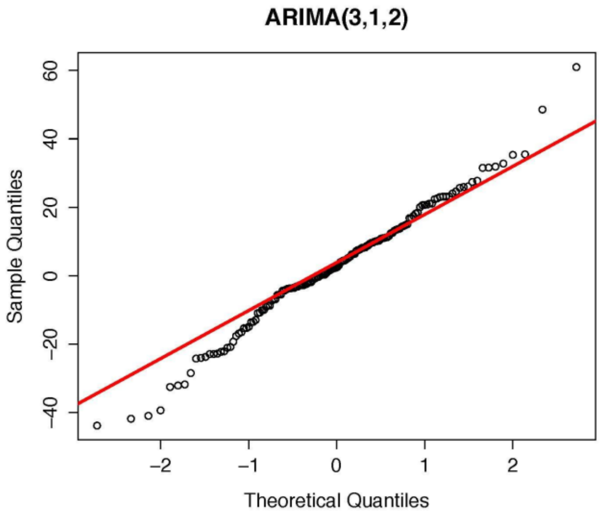
#from the acf and pacf graphs, we can try to fit an SARIMA model and restrict p<=5, q<=4, P<=1, Q<=1

```
> AIC_best = 10**6
> for(p in 0:5){
+       for(q in 0:2){
+       for(P in 0:1){
+       for(Q in 0:1){
+             fit_arima =
Arima(beer_prod[,2],order=c(p,1,q),seasonal=c(P,1,Q),method="ML")
+             aic = fit_arima$aic
+             if(aic < AIC_best){
+             AIC_best = fit_arima$aic
```

```
+                cat("p=",p,"q=",q,"P=",P,"Q=",Q,"\t AIC=",fit_arima$aic,
+                  "\t Number of parameters=",p+q+P+q,"\n")
+                  }
+        }
+        }
+        }
+ }
p= 0 q= 0 P= 0 Q= 0    AIC= 1734.215        Number of parameters= 0
p= 0 q= 1 P= 0 Q= 0    AIC= 1653.471        Number of parameters= 2
p= 1 q= 2 P= 0 Q= 0    AIC= 1653.15         Number of parameters= 5
p= 2 q= 0 P= 0 Q= 0    AIC= 1648.07         Number of parameters= 2
p= 2 q= 1 P= 0 Q= 0    AIC= 1560.387        Number of parameters= 4
p= 2 q= 2 P= 0 Q= 0    AIC= 1466.474        Number of parameters= 6
p= 3 q= 0 P= 0 Q= 0    AIC= 1369.22         Number of parameters= 3
p= 3 q= 1 P= 0 Q= 0    AIC= 1368.317        Number of parameters= 5
p= 3 q= 2 P= 0 Q= 0    AIC= 1335.621        Number of parameters= 7
# We choose the ARIMA(3,1,2)model with the smallest AIC
> Arima (beer_prod[,2], order = c(3,1,2), method="ML")
Series: beer_prod[, 2]
ARIMA(3,1,2)

Coefficients:
          ar1      ar2      ar3      ma1     ma2
      -0.8769  -0.9994  -0.8722  -0.1109  0.7112
s.e.   0.0396   0.0060   0.0389   0.0654  0.1095

sigma^2 estimated as 316.5:  log likelihood=-661.81
AIC=1335.62   AICc=1336.2   BIC=1353.8
```

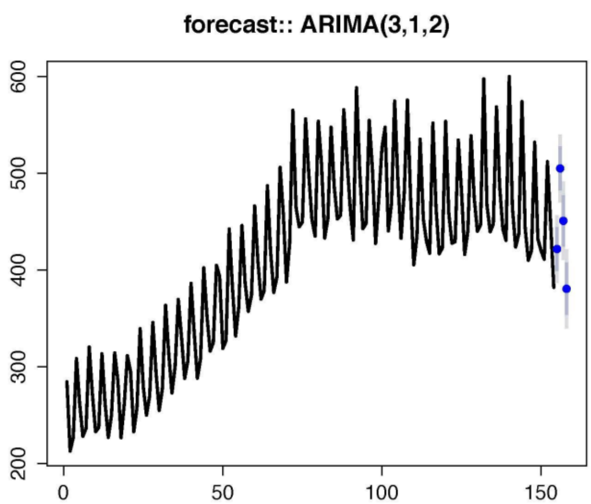
#Hence the fitted sarima model is: $\mathbf{X_k + 0.88X_{k-1} + X_{K-2} + 0.87X_{k-3} = W_k - 0.11W_{k-1} + 0.71W_{k-2}}$

```
#Check the residuals
> arima_fit =  Arima(beer_prod[,2], order = c(3,1,2),method="ML")
> qqnorm(resid(arima_fit), main="ARIMA(3,1,2)")
> qqline(resid(arima_fit), col="red", lwd=3)
```



#the residuals locates generally linearly, suggesting a Gaussian model, so the model can be trusted

```
> arima_beer = Arima (beer_prod[,2], order = c(3,1,2),method=”ML”)
> plot(forecast(arima_fit, h=4), main="forecast:: ARIMA(3,1,2)", lwd=3)
```



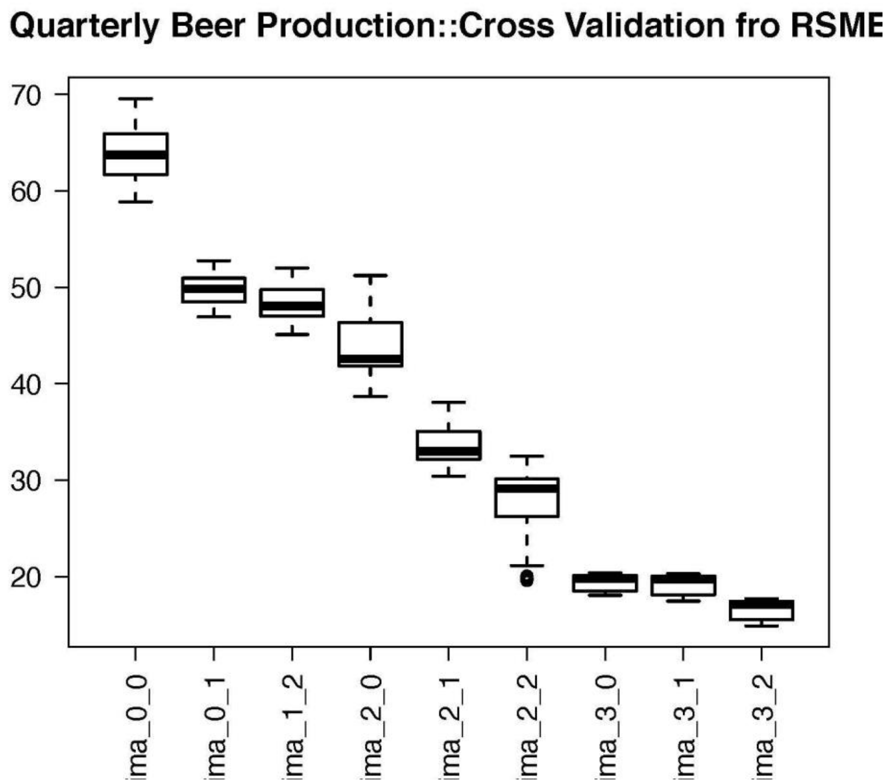

```
> forecast(arima_beer, h=4)
```

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
155	421.5602	398.7593	444.3610	386.6893	456.4311
156	504.8725	482.0700	527.6751	469.9990	539.7460
157	450.7136	424.2368	477.1904	410.2208	491.2064
158	380.4361	353.5754	407.2967	339.3562	421.5159

Hence the forecasted quarterly beer production for 1994Q3, 1994Q4, 1995Q1 and 1995Q2 are (421.56, 504.87, 450.71, 380.44)

Using Cross Validation to justify the answer:

```
> cross_validation = function (time_series, start,forecast_length,ts_model){
+     ts_length=length(time_series)
+     accuracy_list=c()
+         for (k in c(start:(ts_length-forecast_length))){
+         fitted_model = ts_model(time_series[0:k])
+        RSME=accuracy(forecast(fitted_model,h=forecast_length))[2]
+        accuracy_list=c(accuracy_list,RSME)
+         }
+     return(accuracy_list)
+ }
> start=80
> forecast_length=4
> ts_model_0_1_0 = function(ts) return( Arima(ts, order = c(0,1,0),method="ML") )
> ts_model_0_1 = function(ts) return( Arima(ts, order = c(0,1,1),method="ML") )
> ts_model_1_1_2 = function(ts) return( Arima(ts, order = c(1,1,2),method="ML") )
> ts_model_2_1_0 = function(ts) return( Arima(ts, order = c(2,1,0),method="ML") )
> ts_model_2_1_1 = function(ts) return( Arima(ts, order = c(2,1,1),method="ML") )
> ts_model_2_1_2 = function(ts) return( Arima(ts, order = c(2,1,2),method="ML") )
> ts_model_3_1_0 = function(ts) return( Arima(ts, order = c(3,1,0),method="ML") )
> ts_model_3_1_1 = function(ts) return( Arima(ts, order = c(3,1,1),method="ML") )
> ts_model_3_1_2 = function(ts) return( Arima(ts, order = c(3,1,2),method="ML") )

> CV_results=data.frame(
+         arima_0_1_0 = cross_validation(beer_prod[,2],start,forecast_length,ts_model_0_1_0),
+         arima_0_1_1 = cross_validation(beer_prod[,2],start,forecast_length,ts_model_0_1_1),
+         arima_1_1_2 = cross_validation(beer_prod[,2],start,forecast_length,ts_model_1_1_2),
+         arima_2_1_0 = cross_validation(beer_prod[,2],start,forecast_length,ts_model_2_1_0),
+         arima_2_1_1 = cross_validation(beer_prod[,2],start,forecast_length,ts_model_2_1_1),
+         arima_2_1_2 = cross_validation(beer_prod[,2],start,forecast_length,ts_model_2_1_2),
+         arima_3_1_0 = cross_validation(beer_prod[,2],start,forecast_length,ts_model_3_1_0),
+         arima_3_1_1 = cross_validation(beer_prod[,2],start,forecast_length,ts_model_3_1_1),
+         arima_3_1_2 = cross_validation(beer_prod[,2],start,forecast_length,ts_model_3_1_2)
+         )

> boxplot(CV_results, las=2, main = "Quarterly Beer Production::Cross Validation for
RSME",lwd=2)
```

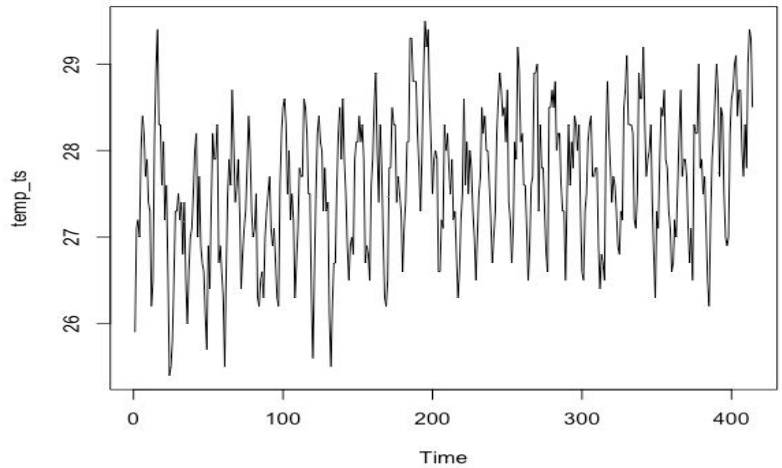


From the boxplots of RSME of different proposed models using Cross Validation approach, we can observe the average RSME for model ARIMA(3,1,2) is the smallest without any outlier, so the forecasted data computed using this fitted model can be trusted.

<u>Exercise 4 (Temperature in Singapore?)</u>

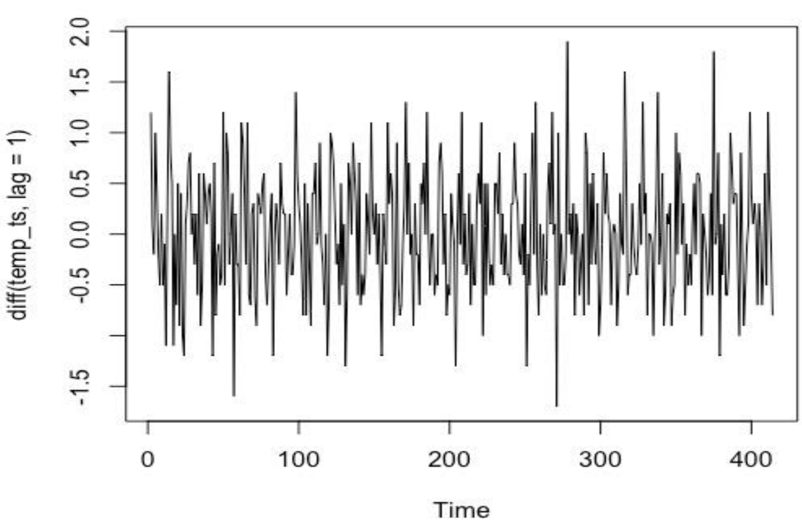
By plotting the mean_temp time series, we will attain the plot below. By observation, there seems to be an increasing trend with no seasonal pattern.

> temp = read.csv("/Users/Downloads/temperature_in_singapore.csv")
> temp_ts = ts(temp$mean_temp)
> plot(temp_ts)



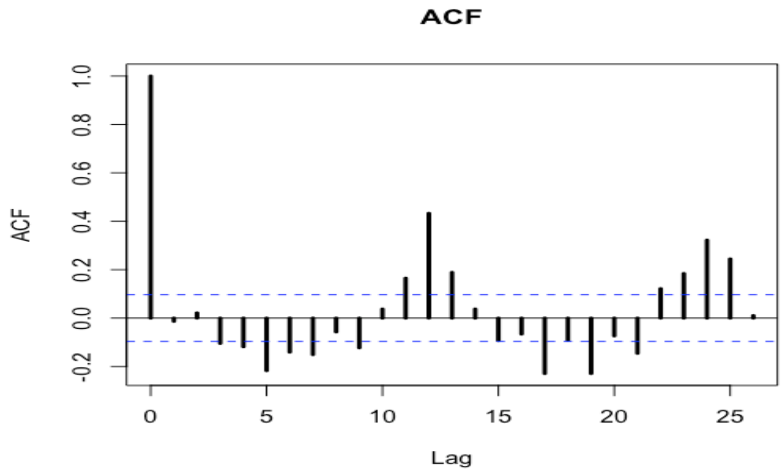
Next, differentiate the time series once to remove the trend.

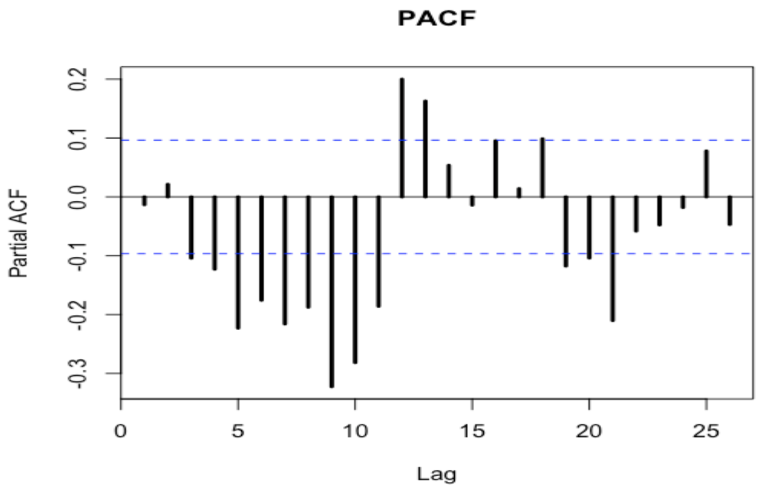
```
> diff_ts = diff(temp_ts,lag=1)
> plot(diff_ts)
```



Plot the ACF and PACF

```
> acf(diff_ts, lwd = 3, main = "ACF")
> pacf(diff_ts, lwd = 3, main = "PACF")
```





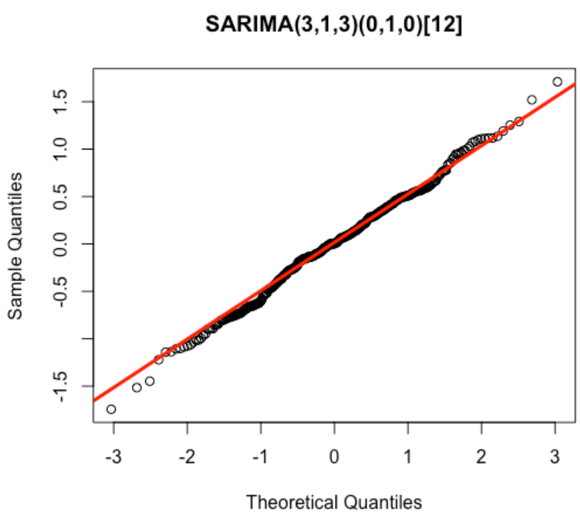
From the ACF plot, we noticed a seasonal pattern of length 12. This time series follows a SARIMA (p,d,q) (P,D,Q) [12]. By restricting the models to p<=4, q<=4, P<=1, Q<=1, we have the following code and result.

```
AIC_best = 10^3
for (p in 0:4){
  for (q in 0:4){
    for (P in c(0,1)){
      for (Q in c(0,1)){
        fit_sarima = Arima(temp_ts, order = c(p,1,q), seasonal = c(P,1,Q))
        aic = fit_sarima$aic
        if (aic< AIC_best){
          AIC_best = fit_sarima$aic
          cat("p=",p,"q=",q,"P=",P,"Q=",Q,"\t AIC=", fit_sarima$aic,"\t Number
              of parameters=", p+q+P+Q, "\n")
        }
      }
    }
  }
}
p= 0 q= 0 P= 0 Q= 0      AIC= 760.8958   Number of parameters= 0
p= 0 q= 3 P= 0 Q= 0      AIC= 712.5027   Number of parameters= 3
p= 0 q= 4 P= 0 Q= 0      AIC= 695.8122   Number of parameters= 4
p= 1 q= 3 P= 0 Q= 0      AIC= 691.588    Number of parameters= 4
p= 1 q= 4 P= 0 Q= 0      AIC= 690.2723   Number of parameters= 5
p= 2 q= 3 P= 0 Q= 0      AIC= 592.3867   Number of parameters= 5
p= 3 q= 3 P= 0 Q= 0      AIC= 563.7208   Number of parameters= 6
```

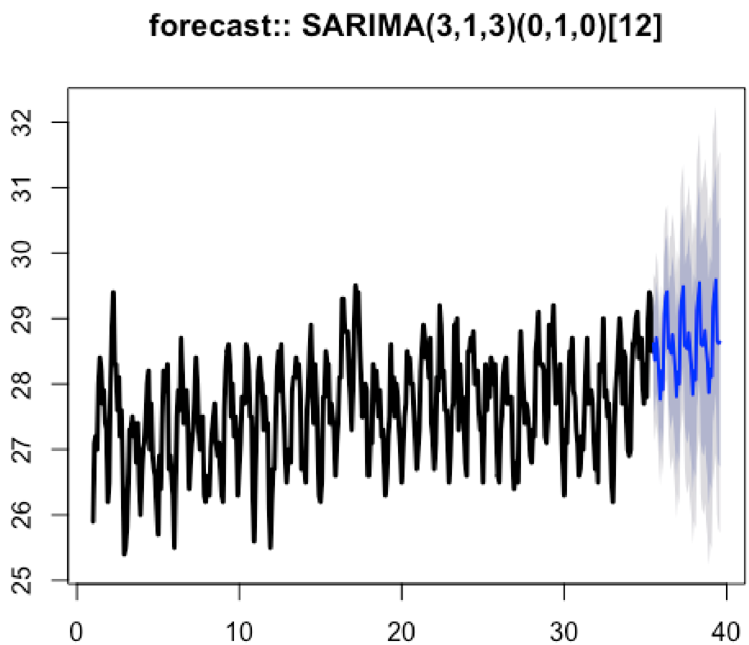
A reasonable model is SARIMA (3,1,3) (0,1,0) [12].

```
> sarima_fit=Arima(temp_ts, order=c(3,1,3), seasonal= c(0,1,0))
> qqnorm(resid(sarima_fit), main="SARIMA(3,1,3)(0,1,0)[12]")
> qqline(resid(sarima_fit), col="red", lwd=3)
```

The residuals are quite normal.



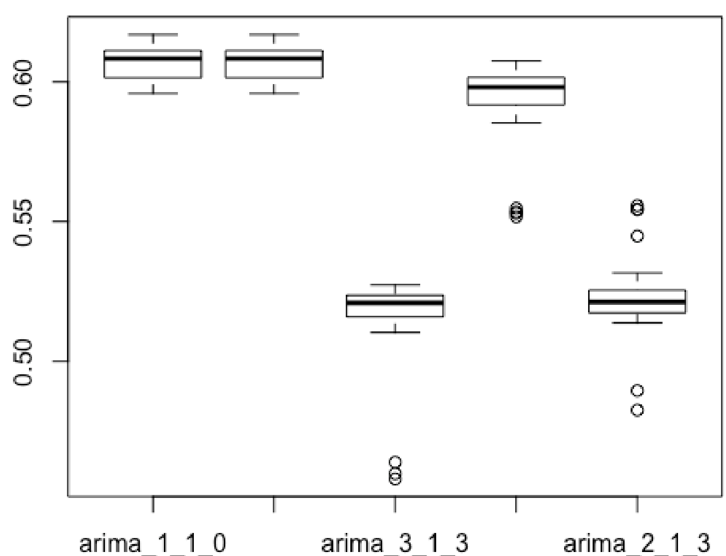

```
>plot(forecast(sarima_fit, h=50), main="forecast:: SARIMA(3,1,3)(0,1,0)[12]",lwd = 3)
```



Using cross validation to check the model:

```
library(fpp)
cross_validation = function(time_series, start, forecast_length, ts_model){
  ts_length = length(time_series)
  accuracy_list = c()
  for(k in c(start:(ts_length - forecast_length))){
    fitted_model = ts_model(time_series[0:k])
    RMSE = accuracy(forecast(fitted_model, h = forecast_length))[2]
    accuracy_list = c(accuracy_list, RMSE)
  }
  return( accuracy_list ) }

ts_1_1_0 = function(ts) return( Arima(ts, order = c(1,1,0), include.drift = F) )
ts_0_1_1 = function(ts) return( Arima(ts, order = c(0,1,1), include.drift = F) )
ts_3_1_3 = function(ts) return( Arima(ts, order = c(3,1,3), include.drift = F) )
ts_2_1_1 = function(ts) return( Arima(ts, order = c(2,1,1), include.drift = F) )
ts_2_1_3 = function(ts) return( Arima(ts, order = c(2,1,3), include.drift = F) )
```

```r
d = ts(temp[,2], frequency = 12)
start = 300
forecast_length = 200

arima_1_1_0 = cross_validation(d, start, forecast_length, ts_1_1_0)
arima_0_1_1 = cross_validation(d, start, forecast_length, ts_0_1_1)
arima_3_1_3 = cross_validation(d, start, forecast_length, ts_3_1_3)
arima_2_1_1 = cross_validation(d, start, forecast_length, ts_2_1_1)
arima_2_1_3 = cross_validation(d, start, forecast_length, ts_2_1_3)

CV_results = data.frame(arima_1_1_0,arima_0_1_1,arima_3_1_3,arima_2_1_1,arima_2_1_3)

boxplot(CV_results)
```



From the box plot diagram, the RMSE for the SARIMA (3,1,3) (0,1,0) [12] model is the smallest as compared to the rest of the other proposed model. Therefore, the SARIMA (3,1,3) (0,1,0) [12] model can provide forecasts that can be trusted.

Exercise 5 (Monthly Car Sales in Quebec?)

Abstract

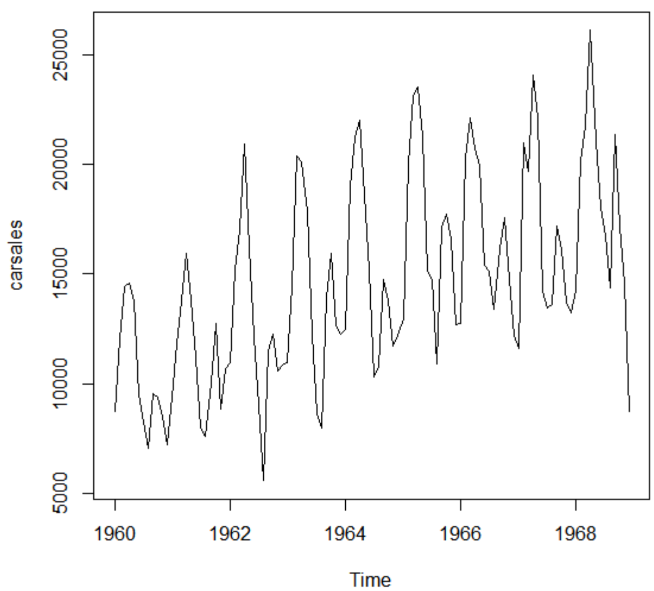
Goal: forecast number of car sales in Quebec during the next two years following the data collection

Time period: 1960 to 1968

Forecast: 1969 and 1970

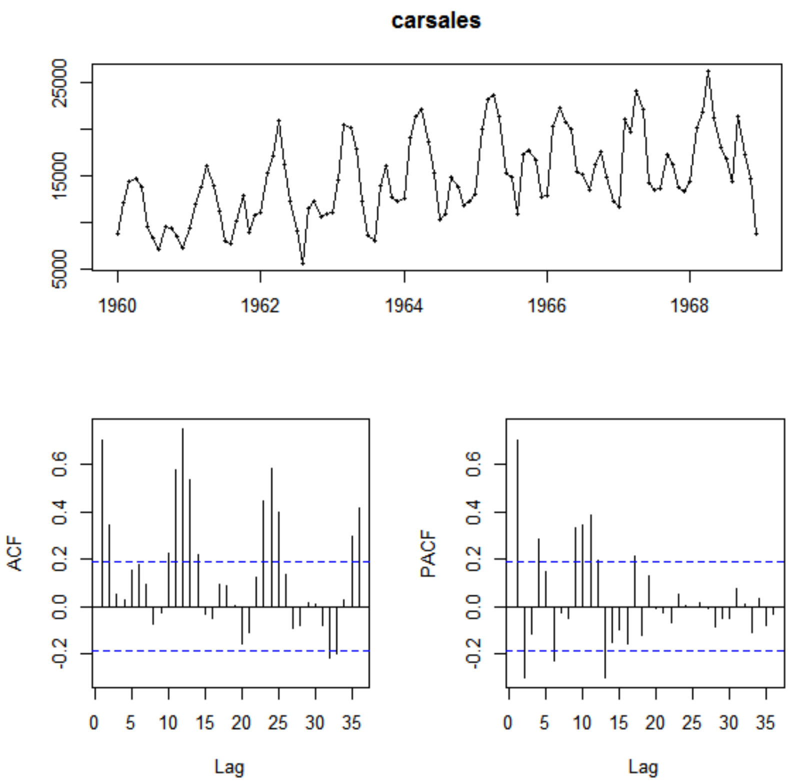
Model: SARIMA(0,1,1)(0,1,1)[12] for lower mean RMSE or TES for lower confidence interval for RMSE

```
data=read.csv("D:/NUS/st3233/monthly-car-sales-in-quebec-1960.csv",skip=2,header=FALSE)
sales=data$V2
carsales=ts(sales,start=c(1960,1),end=c(1968,12),frequency=12)
> carsales
       Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
1960  8728 12026 14395 14587 13791  9498  8251  7049  9545  9364  8456  7237
1961  9374 11837 13784 15926 13821 11143  7975  7610 10015 12759  8816 10677
1962 10947 15200 17010 20900 16205 12143  8997  5568 11474 12256 10583 10862
1963 10965 14405 20379 20128 17816 12268  8642  7962 13932 15936 12628 12267
1964 12470 18944 21259 22015 18581 15175 10306 10792 14752 13754 11738 12181
1965 12965 19990 23125 23541 21247 15189 14767 10895 17130 17697 16611 12674
1966 12760 20249 22135 20677 19933 15388 15113 13401 16135 17562 14720 12225
1967 11608 20985 19692 24081 22114 14220 13434 13598 17187 16119 13713 13210
1968 14251 20139 21725 26099 21084 18024 16722 14385 21342 17180 14577  8728
plot(carsales)
```

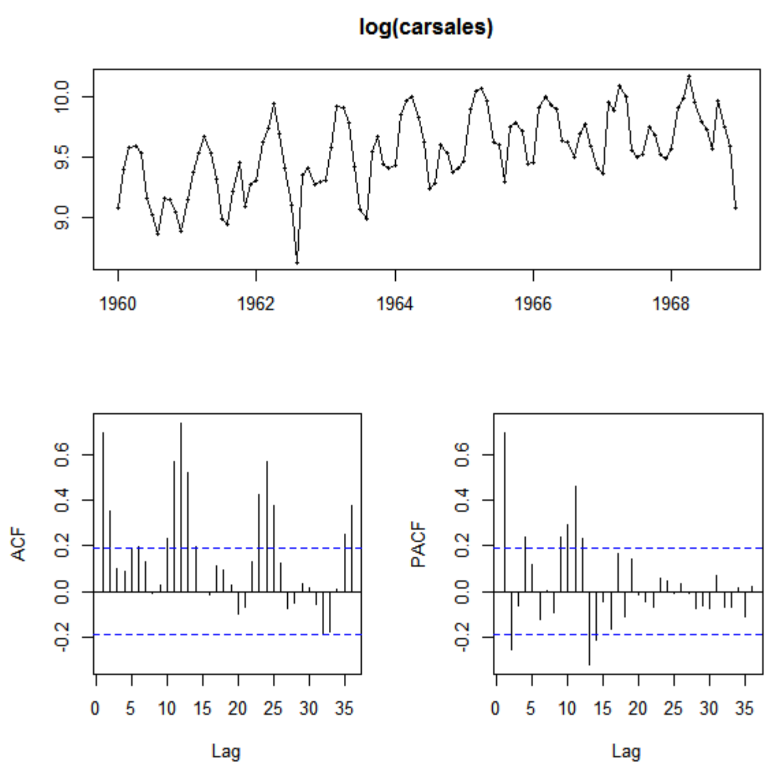


Looks like there is seasonality and upwards trend. Let's plot acf and pacf.

```
tsdisplay(carsales)
```

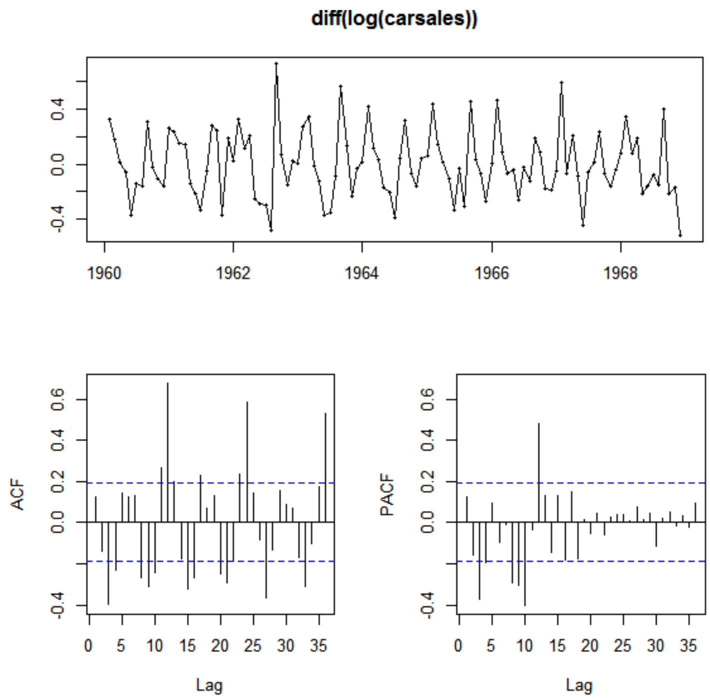


tsdisplay(log(carsales))



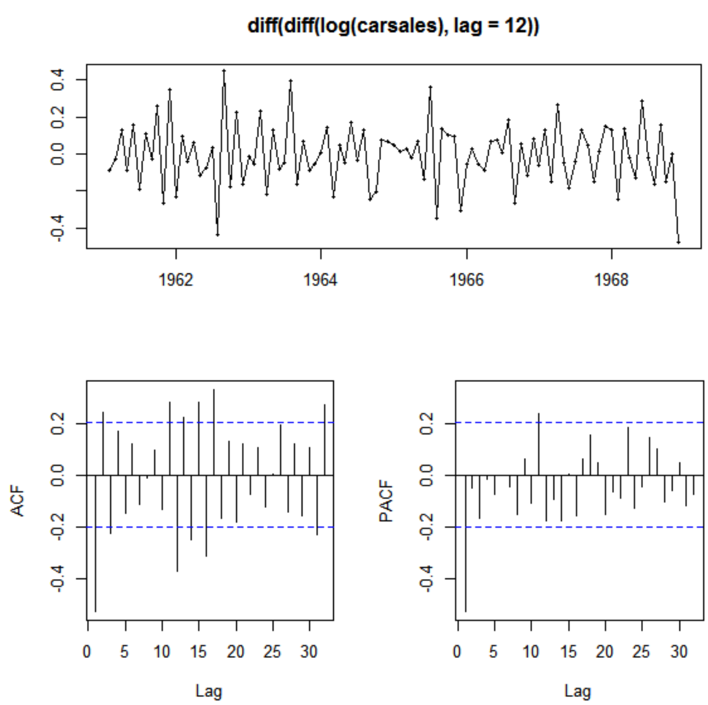
Seems like there is lag 2 from acf and seasonality. But try taking difference of log(carsales)

tsdisplay(diff(log(carsales)))

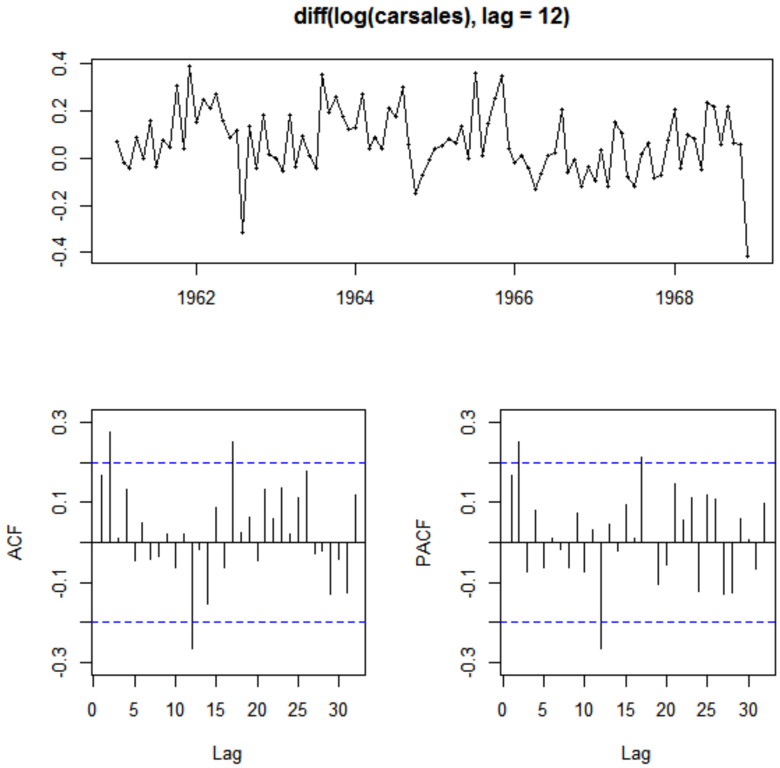


Time series plot looks more stationary. But seems to have seasonality. Trying again.

tsdisplay(diff(diff(log(carsales),lag=12)))



```
tsdisplay(diff(log(carsales),lag=12))
```



Using the latter, seems like there is significant at lag 2 for acf and pacf.

Try SARIMA modelling with AR/MA less than 2

Keeping parameters p and q <=2, P and Q <=1 since data size is not very big, and difference as 1,

```
logcarsales=log(carsales)
AICbest=10^6
for(p in 0:2){
for(q in 0:2){
for(P in 0:1){
for(Q in 0:1){
fit=arima(logcarsales,order=c(p,1,q),seasonal=c(P,1,Q))
aic=fit$aic
if(aic<AICbest){
AICbest=aic
cat("p=",p,"q=",q,"P=",P,"Q=",Q,"\t AIC=",fit$aic,"number of parameter \t=",p+q+P+Q,"\n")
}
}
}
}
}
```

```
p= 0 q= 0 P= 0 Q= 0      AIC= -63.09228 number of parameter     = 0
p= 0 q= 0 P= 0 Q= 1      AIC= -90.38152 number of parameter     = 1
p= 0 q= 1 P= 0 Q= 0      AIC= -100.0331 number of parameter     = 1
p= 0 q= 1 P= 0 Q= 1      AIC= -114.7018 number of parameter     = 2
p= 0 q= 2 P= 0 Q= 1      AIC= -114.7581 number of parameter     = 3
p= 1 q= 1 P= 0 Q= 1      AIC= -115.8669 number of parameter     = 3
p= 2 q= 1 P= 0 Q= 1      AIC= -116.3408 number of parameter     = 4
```

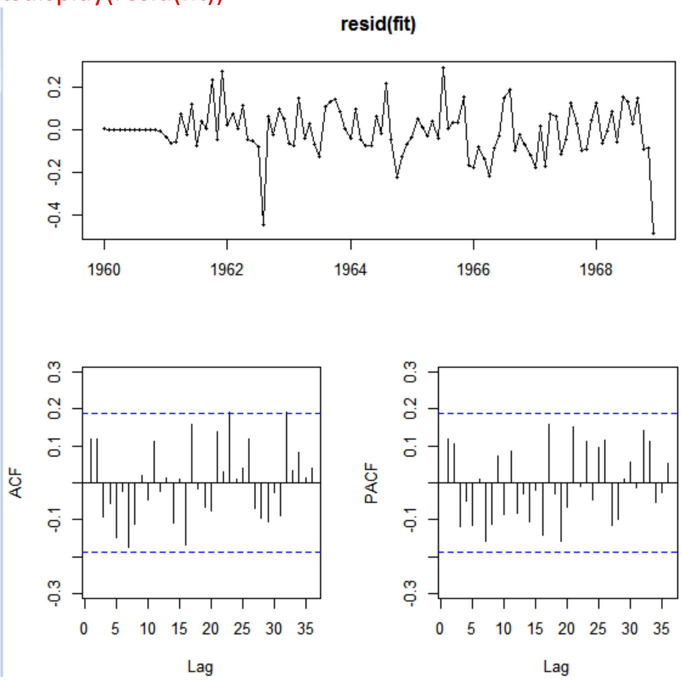
Since AIC=-114 is the lowest, let's pick the simplest model. p=0,q=1,P=0,Q=1

In between have tried with Arima with Drift included, arima without difference and a few combinations (not included here due to space) but results were no better or much worse. Hence arima without drift is used as the final.

Using model with p=0,q=1,P=0,Q=1,difference =1,

fit=arima(logcarsales,order=c(0,1,1),seasonal=c(0,1,1))

tsdisplay(resid(fit))

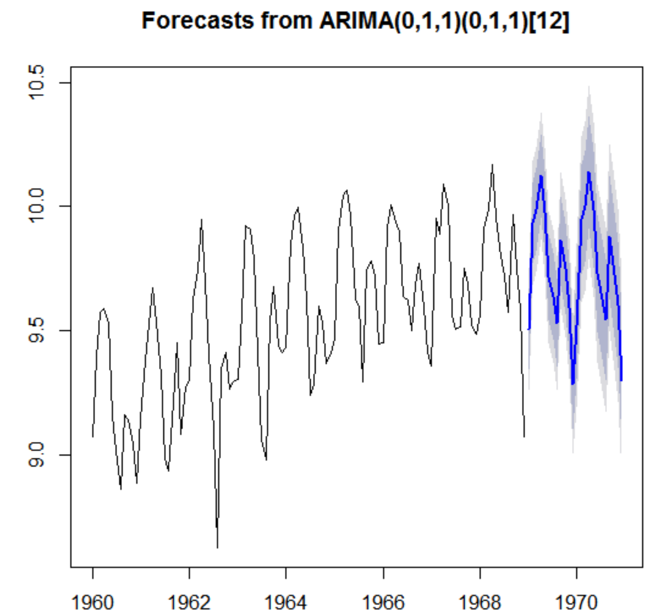


s

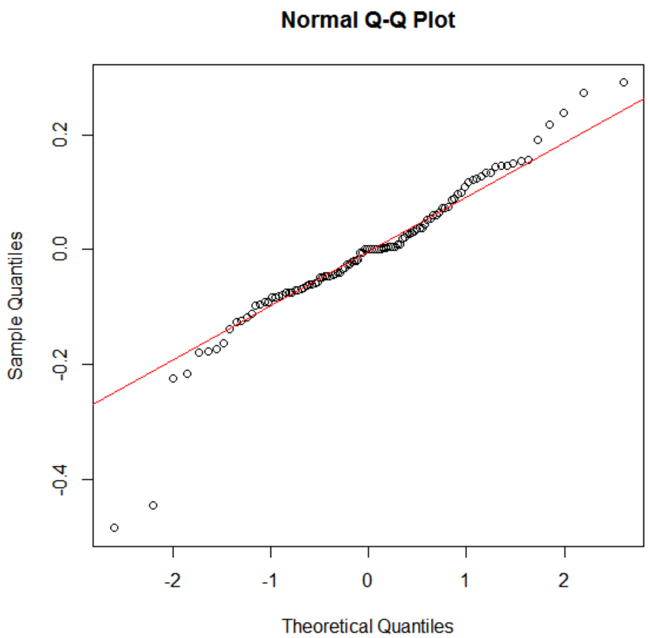
Looks good.

forecast=forecast(fit,h=24)

plot(forecast)



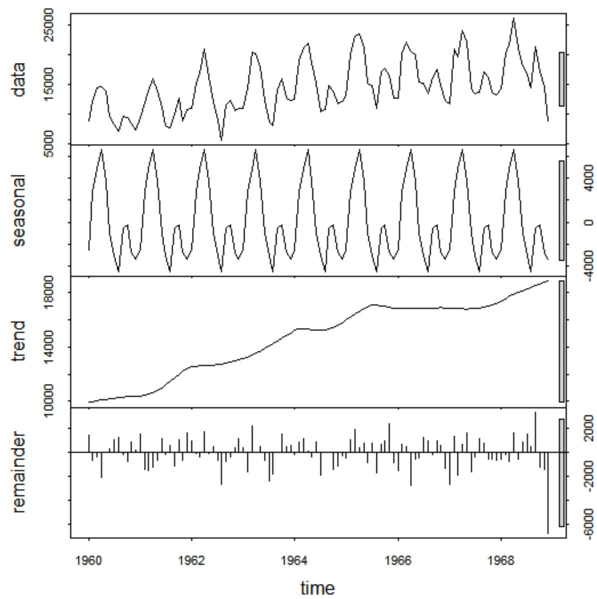
qqnorm(resid(fit))
qqline(resid(fit),col="red")



Somewhat Gaussian. Good.
The forecast I think was not bad.

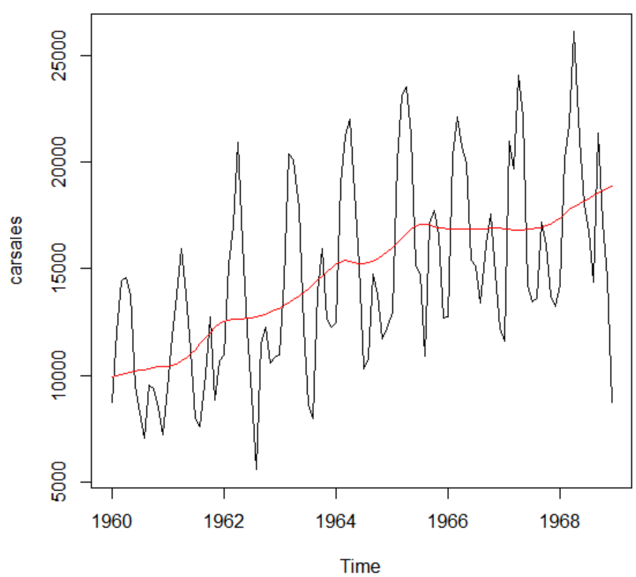
Let’s proceed to use exponential model and do a cross validation to compare.
Using Exponential model:

```
carsalesdecompose=stl(carsales,s.window="periodic",robust=TRUE)
plot(carsalesdecompose)
```

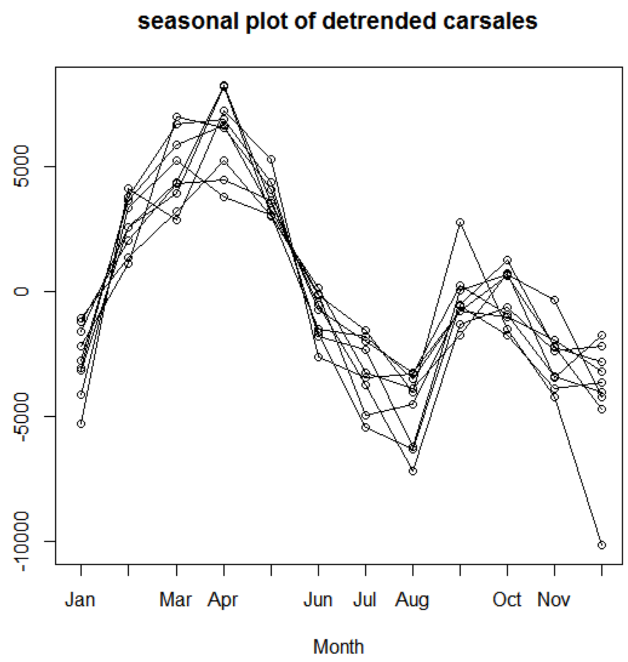


There is seasonlity and trend.

```
trend=carsalesdecompose$time.series[,"trend"]
plot(carsales)
lines(trend,col="red")
```

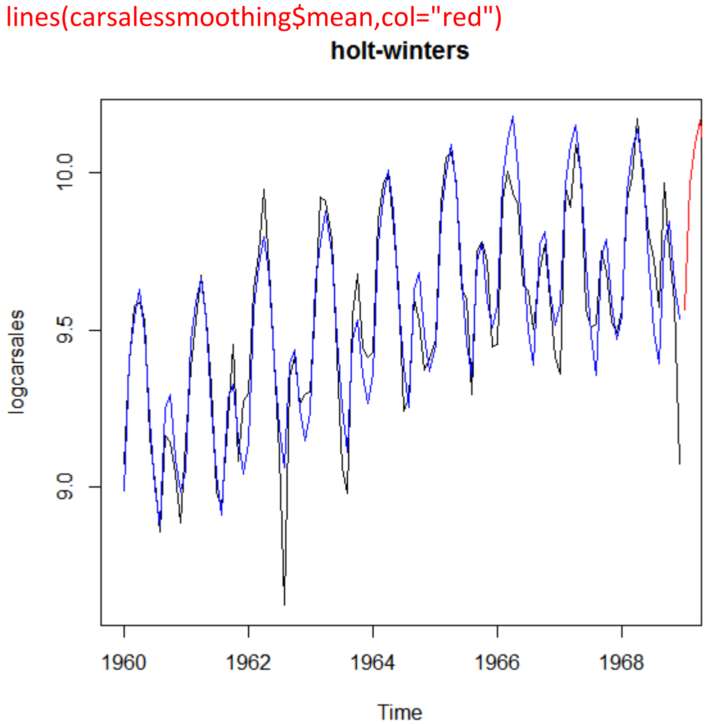



```
seasonplot(carsales-trend,s=12,main="seasonal plot of detrended carsales")
```



Obvious seasonal component.
Let’s apply Holt’s winter algorithm

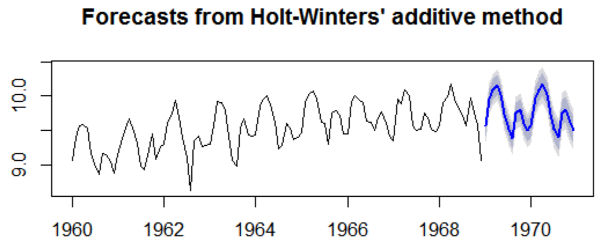
```
carsalessmoothing=hw(logcarsales,h=24,initial="optimal",seasonal="additive")
plot(logcarsales,main="holt-winters")
lines(fitted(carsalessmoothing),col="blue")
lines(carsalessmoothing$mean,col="red")
```

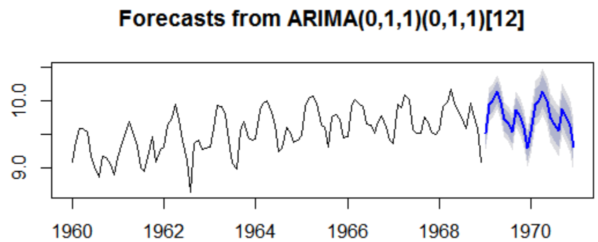


Comparing with earlier forecast side by side

```
fit=arima(logcarsales,order=c(0,1,1),seasonal=c(0,1,1))
forecast=forecast(fit,h=24)
```

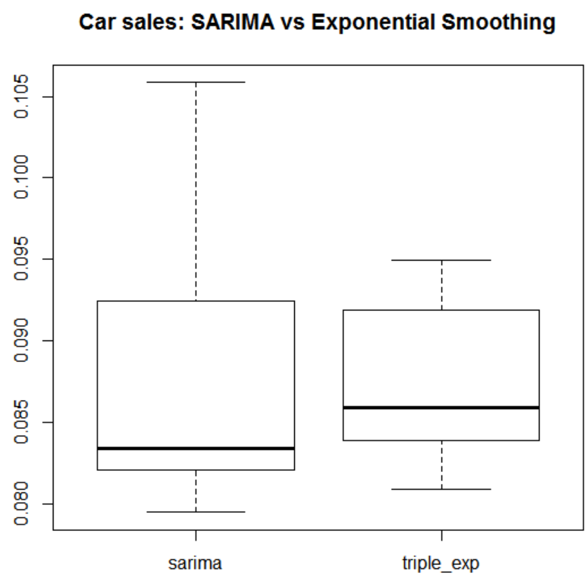
```
par(mfrow=c(2,1))
plot(carsalessmoothing)
plot(forecast)
```





Looks similar. Performing a cross validation to decide which is better:

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

ts_model_sarima = function(tseries) return( Arima(tseries,order=c(0,1,1),seasonal=c(0,1,1),include.drift
= F))
ts_model_triple = function(tseries) return( hw(tseries,h=24,initial="optimal",seasonal="additive"))
start = 5*12
forecast_length = 24
CV_results = data.frame(
sarima = cross_validation(logcarsales, start, forecast_length, ts_model_sarima),
triple_exp = cross_validation(logcarsales, start, forecast_length, ts_model_triple)
)

boxplot(CV_results,main = "Car sales: SARIMA vs Exponential Smoothing")
```



Difference between RMSE of both models is very small. Generally will pick the one with smaller RMSE which is Sarima. But if want better forecast confidence interval, will pick triple exponential smoothing model

Forecast from SARIME:

```
> fit=arima(logcarsales,order=c(0,1,1),seasonal=c(0,1,1))
> forecast=forecast(fit,h=24)
> forecast
         Point Forecast    Lo 80     Hi 80    Lo 95     Hi 95
Jan 1969       9.506714 9.346405  9.667022 9.261542  9.751885
Feb 1969       9.934379 9.771607 10.097151 9.685441 10.183318
Mar 1969       9.987660 9.822461 10.152859 9.735011 10.240310
Apr 1969      10.125825 9.958235 10.293415 9.869518 10.382132
May 1969       9.976578 9.806630 10.146526 9.716665 10.236491
Jun 1969       9.718106 9.545832  9.890379 9.454636  9.981576
Jul 1969       9.640650 9.466082  9.815218 9.373672  9.907629
Aug 1969       9.530000 9.353167  9.706833 9.259557  9.800442
Sep 1969       9.863863 9.684794 10.042932 9.590001 10.137726
Oct 1969       9.752940 9.571663  9.934218 9.475700 10.030181
Nov 1969       9.592598 9.409139  9.776058 9.312021  9.873176
Dec 1969       9.287187 9.101571  9.472803 9.003312  9.571062
Jan 1970       9.520263 9.305175  9.735350 9.191314  9.849211
Feb 1970       9.947928 9.728707 10.167149 9.612658 10.283198
Mar 1970      10.001209 9.777931 10.224487 9.659735 10.342684
Apr 1970      10.139374 9.912111 10.366637 9.791806 10.486942
May 1970       9.990127 9.758949 10.221306 9.636570 10.343685
Jun 1970       9.731655 9.496626  9.966684 9.372209 10.091101
Jul 1970       9.654200 9.415382  9.893017 9.288959 10.019440
Aug 1970       9.543549 9.301001  9.786096 9.172605  9.914493
Sep 1970       9.877412 9.631192 10.123633 9.500851 10.253974
Oct 1970       9.766490 9.516650 10.016329 9.384393 10.148586
Nov 1970       9.606147 9.352741  9.859554 9.218596  9.993699
Dec 1970       9.300736 9.043812  9.557661 8.907805  9.693668
```

From TES:

```
> carsalessmoothing
         Point Forecast     Lo 80     Hi 80    Lo 95     Hi 95
Jan 1969       9.563279  9.417522  9.709037 9.340362  9.786196
Feb 1969       9.961946  9.815327 10.108564 9.737712 10.186179
Mar 1969      10.089141  9.941563 10.236720 9.863440 10.314843
Apr 1969      10.166790 10.018150 10.315429 9.939465 10.394114
May 1969      10.024147  9.874340 10.173953 9.795038 10.253256
Jun 1969       9.737912  9.586830  9.888995 9.506852  9.968973
Jul 1969       9.517959  9.365488  9.670429 9.284775  9.751142
Aug 1969       9.389948  9.235975  9.543921 9.154467  9.625429
Sep 1969       9.754741  9.599149  9.910333 9.516783  9.992698
Oct 1969       9.801194  9.643864  9.958524 9.560578 10.041809
Nov 1969       9.608331  9.449143  9.767519 9.364874  9.851788
Dec 1969       9.500420  9.339251  9.661590 9.253932  9.746909
Jan 1970       9.571654  9.408382  9.734925 9.321951  9.821356
Feb 1970       9.970320  9.804823 10.135817 9.717215 10.223425
Mar 1970      10.097516  9.929671 10.265361 9.840820 10.354212
Apr 1970      10.175164 10.004848 10.345480 9.914688 10.435640
May 1970      10.032521  9.859611 10.205432 9.768078 10.296965
Jun 1970       9.746287  9.570660  9.921914 9.477689 10.014885
Jul 1970       9.526333  9.347868  9.704799 9.253394  9.799272
Aug 1970       9.398322  9.216898  9.579747 9.120858  9.675787
Sep 1970       9.763115  9.578612  9.947619 9.480942 10.045289
Oct 1970       9.809568  9.621868  9.997269 9.522505 10.096632
Nov 1970       9.616705  9.425690  9.807720 9.324573  9.908838
Dec 1970       9.508795  9.314348  9.703242 9.211413  9.806177
```