Homework 2

Name: Anutida Sangkla ID: 002602236

Problem 1

part (a)

Note that

$$MSE(A) = E[(x_{t+\ell} - Ax_t)^2]$$

$$MSE(A) = E(x_{t+\ell}^2) - 2AE(x_{t+\ell}\, x_t) + A^2E(x_t^2)$$

From the the autocovariance, we will get

$$MSE(A) = \gamma(0) \; - 2A\gamma(\ell) \; + A^2\gamma(0)$$

Finding the derivative respect to A

$$\frac{dMSE(A)}{dA} = -2\gamma(\ell) + 2A\gamma(0)$$

Setting the derivative to zero and solve for A

$$-2\gamma(\ell) + 2A\gamma(0) = 0$$

$$A = \frac{2\gamma(\ell)}{2\gamma(0)}$$

$$A = \frac{\gamma(\ell)}{\gamma(0)} = \rho(\ell)$$

Therefore, it can be seen that the minimum is obtained at $A = \rho(\ell)$

part (b) Show that the minimum mean-square prediction error is

$$MSE(A) = \gamma(0)[1 - \rho(\ell)^2]$$

From

$$MSE(A) = \gamma(0) \; - 2A\gamma(\ell) \; + A^2\gamma(0)$$

$$MSE(A) = \gamma(0) \; - 2\rho(\ell)\gamma(\ell) \; + \rho(\ell)^2\gamma(0)$$

$$\gamma(0)\left[1 - \frac{2\rho(\ell)\gamma(\ell)}{\gamma(0)} \; + \rho(\ell)^2\right]$$

$$\gamma(0)[1 - 2\rho(\ell)^2 + \rho(\ell)^2]$$

Therefore,

$$MSE(A) = \gamma(0)[1 - \rho(\ell)^2]$$

part (c)

If $x_{t+\ell} = Ax_t$, then

$$E(x_{t+\ell} - Ax_t)^2 = \gamma(0)[1 - \rho(\ell)^2]$$

Since

$$A = \rho(\ell)$$

If

$$A > 0$$

implies that

$$\rho(\ell) = 1$$

and if

$$A < 0$$

implies that

$$\rho(\ell) = -1$$

Problem 2

part (a)

$$x_t = 0.5x_{t-1} - 0.25x_{t-3} + w_t + 0.8w_{t-1} - 0.2w_{t-2}$$

Then

$$x_t - 0.5x_{t-1} + 0.25x_{t-3} = w_t + 0.8w_{t-1} - 0.2w_{t-2}$$

$$(1 - 0.5B + 0.25B^3)x_t = (1 + 0.8B - 0.2B^2)w_t$$

The AR polynomial is

$$\phi(B) = 1 - 0.5B + 0.25B^3$$

The MA polynomial is

$$\theta(B) = 1 + 0.8B - 0.2B^2$$

Check causality and invertibility:

```
AR.poly<-c(1,-0.5,0,0.25)
(a = polyroot(AR.poly))

## [1]  1+1i -2+0i  1-1i

MA.poly<-c(1,0.8,-0.2)
(b = polyroot(MA.poly))

## [1] -1-0i  5+0i
```

From the result, It turns out that there is no common root of the AR and MA polynomials.

Then, the root z of AR polynomial are

```
a[1]

## [1] 1+1i

Re(a[1])

## [1] 1

Im(a[1])

## [1] 1

z1 = sqrt(Re(a[1])^2+Im(a[1])^2)
sprintf('%s = %3.4f', '|z1|',z1)

## [1] "|z1| = 1.4142"

a[2]

## [1] -2+0i

Re(a[2])

## [1] -2

Im(a[2])

## [1] 7.034092e-18

z2 = sqrt(Re(a[2])^2+Im(a[2])^2)
sprintf('%s = %3.0f', '|z2|',z2)

## [1] "|z2| =   2"

a[3]

## [1] 1-1i

Re(a[3])

## [1] 1
```

```r
Im(a[3])
```

```
## [1] -1
```

```r
z3 = sqrt(Re(a[3])^2+Im(a[3])^2)
sprintf('%s = %3.5f', '|z3|',z3)
```

```
## [1] "|z3| = 1.41421"
```

$$|z_1| = \ 1.4142 > 1$$

$$|z_2| = \ 2 > 1$$

$$|z_3| = \ 1.4142 > 1$$

Therefore, this process is causal.

The root z of MA polynomial are

```r
b[1]
```

```
## [1] -1-0i
```

```r
Re(b[1])
```

```
## [1] -1
```

```r
Im(b[1])
```

```
## [1] -2.524355e-29
```

```r
z1 = sqrt(Re(b[1])^2+Im(b[1])^2)
sprintf('%s = %3.0f', '|z1|',z1)
```

```
## [1] "|z1| =   1"
```

```r
b[2]
```

```
## [1] 5+0i
```

```r
Re(b[2])
```

```
## [1] 5
```

```r
Im(b[2])
```

```
## [1] 2.524355e-29
```

```r
z2 = sqrt(Re(b[2])^2+Im(b[2])^2)
sprintf('%s = %3.0f', '|z2|',z2)
```

```
## [1] "|z2| =   5"
```

$$|z_1| = \ 1 = 1$$

$$|z_2| = \ 5 > 1$$

Therefore, this process is not invertible since the MA polynomial has root z = 1 which is not larger than 1.

As a result, they are causal but not invertible.

part (b)

$$x_t = 2.6x_{t-1} + 1.2x_{t-2} + w_t - 2.9w_{t-1} - 0.3w_{t-2}$$

Then

$$x_t - 2.6x_{t-1} - 1.2x_{t-2} = w_t - 2.9w_{t-1} - 0.3w_{t-2}$$

$$(1 - 2.6B - 1.2B^2)x_t = (1 - 2.9B - 0.3B^2)w_t$$

The AR polynomial is

$$\phi(B) = 1 - 2.6B - 1.2B^2$$

The MA polynomial is

$$\theta(B) = 1 - 2.9B - 0.3B^2$$

Check causality and invertibility:

```
AR.poly<-c(1,-2.6,-1.2)
(a = polyroot(AR.poly))

## [1]  0.3333333+0i -2.5000000-0i

MA.poly<-c(1,-2.9,-0.3)
(b = polyroot(MA.poly))

## [1]   0.3333333+0i -10.0000000-0i
```

From the result, It turns out that 0.33 is a common root of the AR and MA polynomials. Thus, it can be canceled out.

Then, the remaining AR has

```
phi_1 = 1/ abs(Re(a[2]))
sprintf('%s = %3.1f', 'phi(1)',phi_1)

## [1] "phi(1) = 0.4"
```

$$\phi(1) = 0.4$$

the remaining MA has

```
theta_1 = 1/ abs(Re(b[2]))
```

```r
sprintf('%s = %3.1f', 'theta(1)',theta_1)

## [1] "theta(1) = 0.1"
```

$$\theta(1) = 0.1$$

Note that the root z of AR polynomial is

```r
a[2]

## [1] -2.5-0i

Re(a[2])

## [1] -2.5

Im(a[2])

## [1] -1.032321e-21

z = sqrt(Re(a[2])^2+Im(a[2])^2)
sprintf('%s = %3.1f', '|z|',z)

## [1] "|z| = 2.5"
```

$$|z| = \ 2.5 > 1$$

This process is causal.

The root z of MA polynomial is

```r
b[2]

## [1] -10-0i

Re(b[2])

## [1] -10

Im(b[2])

## [1] -4.038968e-28

z = sqrt(Re(b[2])^2+Im(b[2])^2)
sprintf('%s = %3.0f', '|z|',z)

## [1] "|z| =  10"
```

Also,

$$|z| = \ 10 > 1$$

Therefore, this process is invertible.

As a result, they are causal and invertible.

We can get the MA representation of this process by

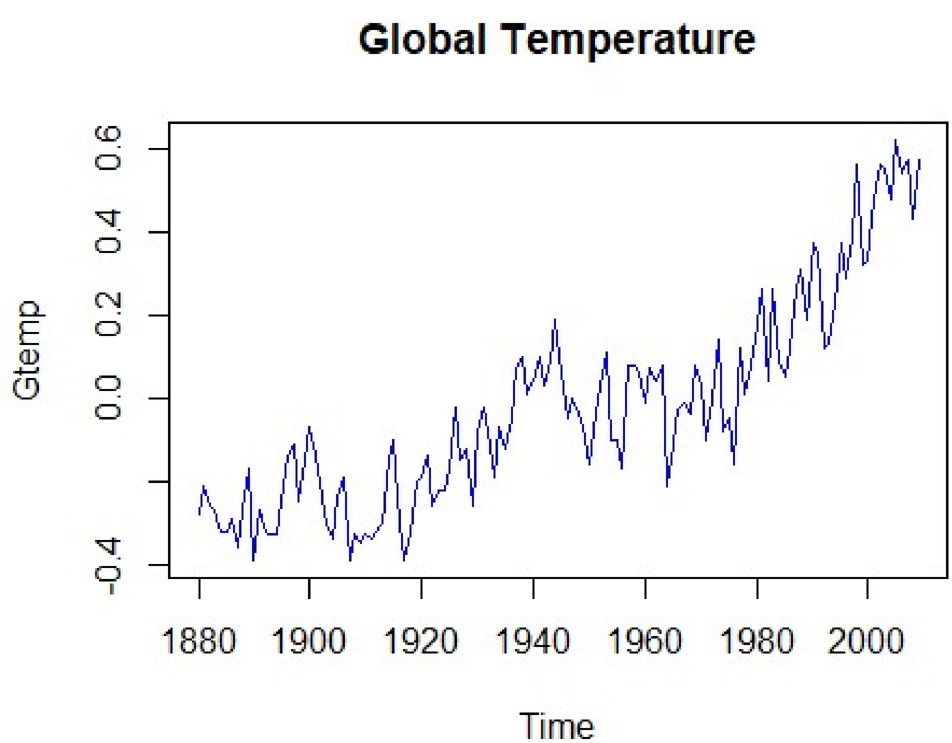
```
ARMAtoMA(ar = 0.4, ma = 0.1, 10)

##  [1] 0.500000000 0.200000000 0.080000000 0.032000000 0.012800000 0.005120000
##  [7] 0.002048000 0.000819200 0.000327680 0.000131072
```

We can get the AR representation of this process by reversing

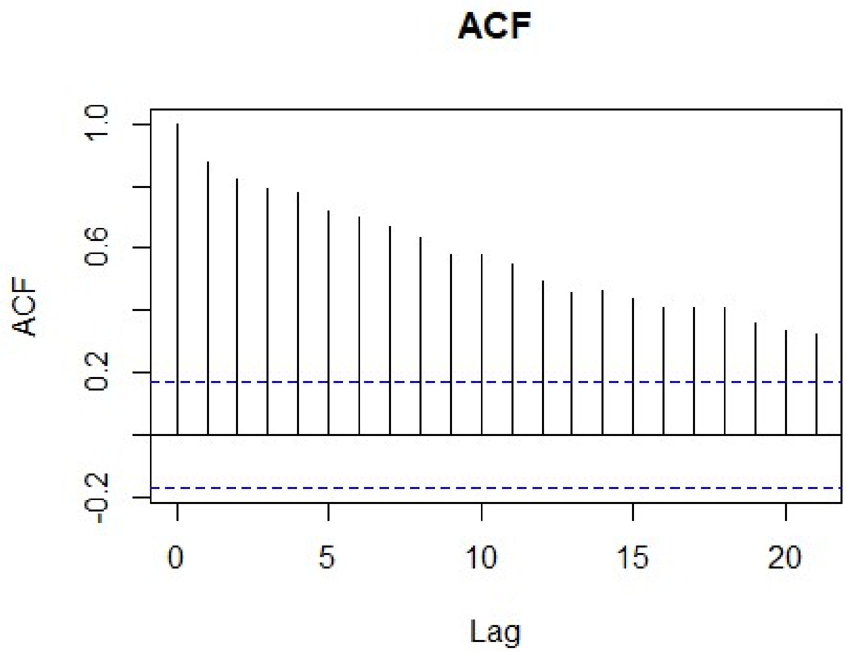
```
ARMAtoMA(ar = -0.4, ma = -0.1, 10)

##  [1] -0.500000000  0.200000000 -0.080000000  0.032000000 -0.012800000
##  [6]  0.005120000 -0.002048000  0.000819200 -0.000327680  0.000131072
```

Problem 3

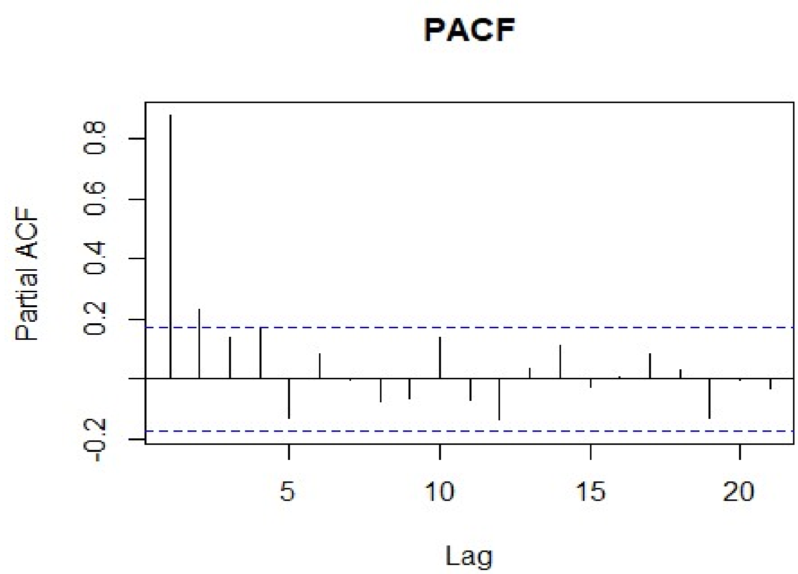
```
library(astsa)

## Warning: package 'astsa' was built under R version 4.0.3

par(mfrow=c(1,1))
plot(gtemp, main="Global Temperature", col="blue", ylab = "Gtemp")
```



```r
acf(gtemp, main = 'ACF')
```



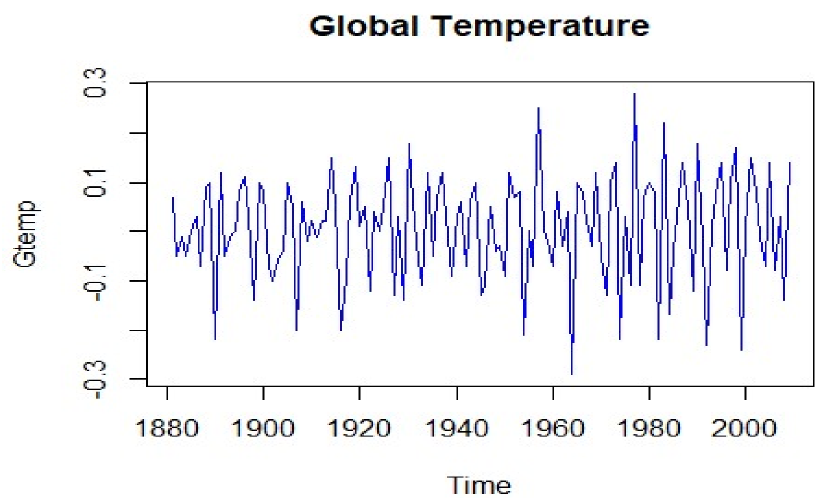

```r
pacf(gtemp, main = 'PACF')
```



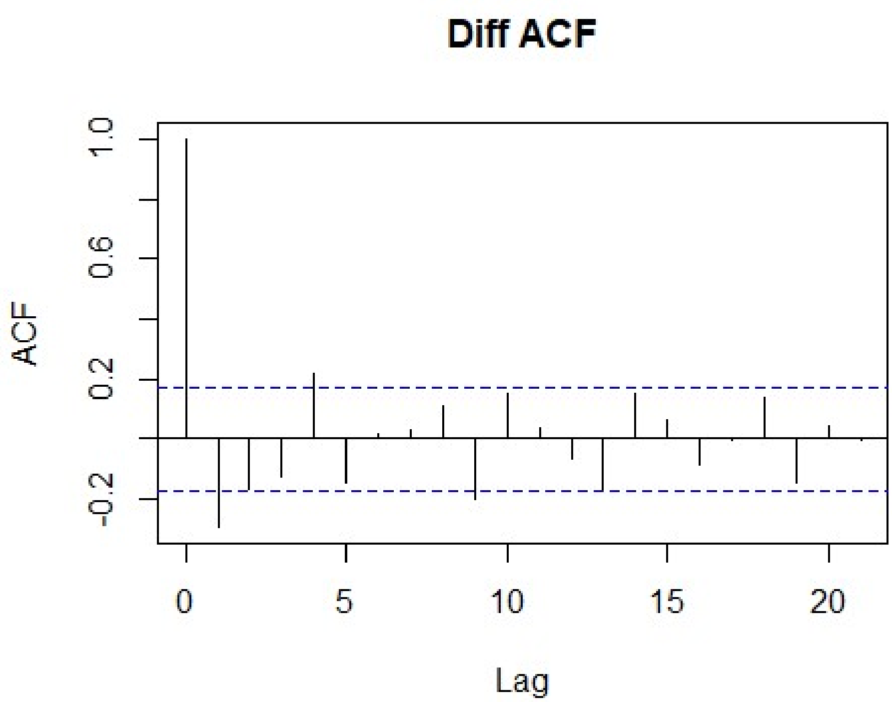
It can be observed from the time series plot of global temperature that the mean of the global temperature seems not constant over time. Therefore, the series are not stationary. The ACF plot indicates an AR(1) model since it gradually tapers to zero and the PACF plot also indicates that lag 1 is highly significant. As a result, AR(1) model should be included to the model.

Data transformation

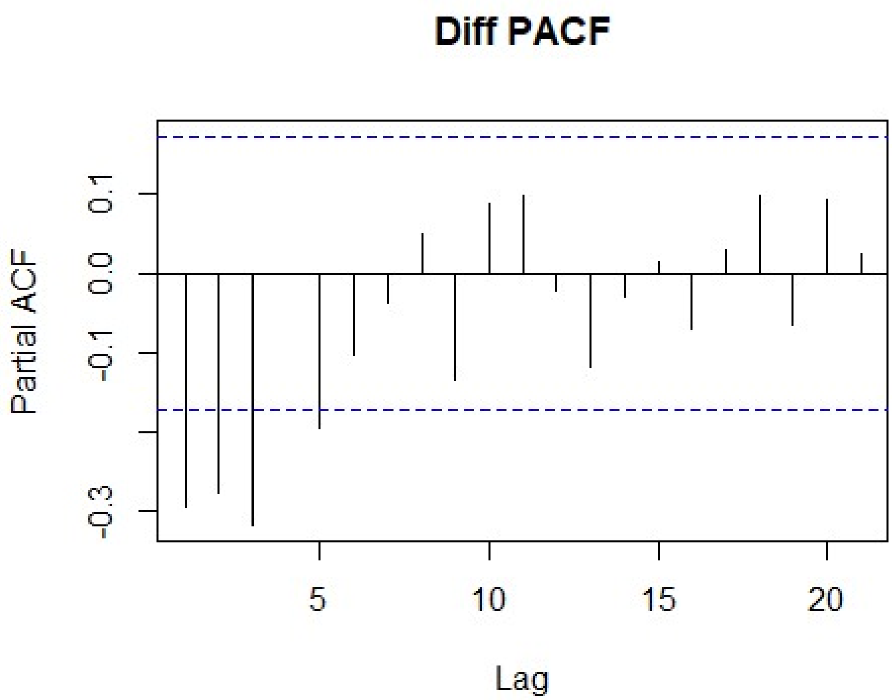
```
par(mfrow=c(1,1))
plot(diff(gtemp), main="Global Temperature", col="blue", ylab = "Gtemp")
```




```
acf(diff(gtemp), main = 'Diff ACF' )
```




```
pacf(diff(gtemp), main = 'Diff PACF')
```



Since the data seem not to be stationary, we transformed data. From the plot, we can observe that the mean of differenced global temperature seems constant and the logistic transformation is not necessary since the variance seems constant. From ACF and PACF plots, the MA(1) should be included in this model

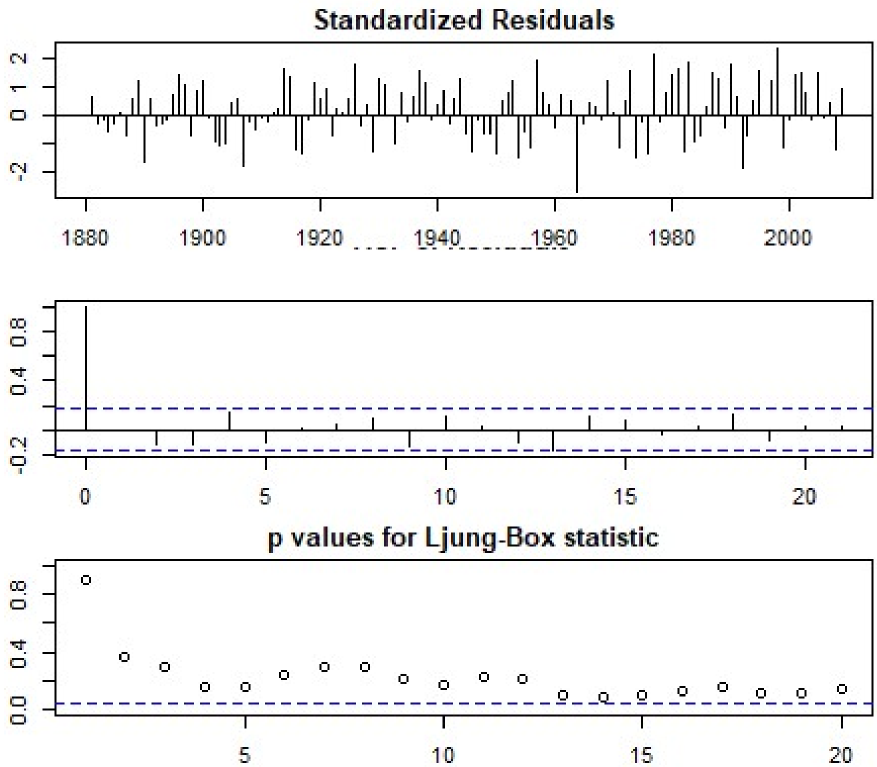
Fitting model

The appropriate model is ARIMA(1,1,1).

```
fit.model = arima(gtemp, order = c(1,1,1))
fit.model

##
## Call:
## arima(x = gtemp, order = c(1, 1, 1))
##
## Coefficients:
##          ar1      ma1
##       0.2256  -0.7158
## s.e.  0.1235   0.0792
##
## sigma^2 estimated as 0.009539:  log likelihood = 116.83,  aic = -227.65
```

Model diagnosis

```
par(mfrow=c(1,1),mar=c(2,2,2,4))
tsdiag(fit.model, gof.lag = 20)
```



From the standardized residuals, the plot indicates white noise process with a few outliers around 1960. The ACF of residuals plot indicates that all the lags are significant. Most of the p-values are above 0, indicating model fits and seems normally distributed.

Model Forecast

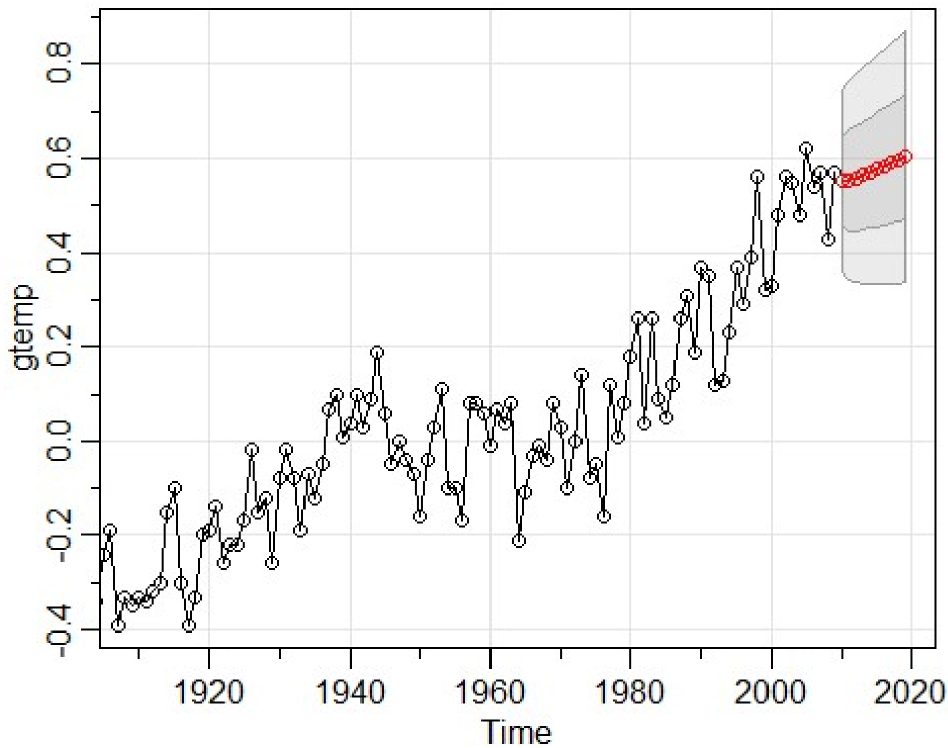
We can predict next 10 years global temperature by

```
predict.model = predict(fit.model, n.ahead = 10)
print(predict.model)
```

```
## $pred
## Time Series:
## Start = 2010
## End = 2019
## Frequency = 1
##  [1] 0.5382877 0.5311345 0.5295210 0.5291570 0.5290749 0.5290564 0.5290522
##  [8] 0.5290513 0.5290511 0.5290510
##
## $se
## Time Series:
## Start = 2010
## End = 2019
## Frequency = 1
##  [1] 0.09766745 0.10962454 0.11635025 0.12195574 0.12715800 0.13212192
##  [7] 0.13689865 0.14151265 0.14598053 0.15031559
```

Or an alternative approach is

```
predict = sarima.for(gtemp, 10,1,1,1)
```




```
print(predict)

## $pred
## Time Series:
## Start = 2010
## End = 2019
## Frequency = 1
##  [1] 0.5524482 0.5527277 0.5575903 0.5636309 0.5699743 0.5763955 0.5828367
##  [8] 0.5892831 0.5957307 0.6021788
##
## $se
## Time Series:
## Start = 2010
## End = 2019
## Frequency = 1
##  [1] 0.09572513 0.10583855 0.11061397 0.11430409 0.11767181 0.12089498
##  [7] 0.12402174 0.12706842 0.13004294 0.13295073
```

From the prediction, the global temperature seems to be upward trend which means that the global temperature increases as time increase.

Problem 4

```r
library(readxl)
options(warn=-1)
sales <- read_excel("sales.xls")
str(sales)
```

```
## tibble [9,994 x 21] (S3: tbl_df/tbl/data.frame)
##  $ Row ID       : num [1:9994] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Order ID     : chr [1:9994] "CA-2016-152156" "CA-2016-152156" "CA-2016-13868
8" "US-2015-108966" ...
##  $ Order Date   : POSIXct[1:9994], format: "2016-11-08" "2016-11-08" ...
##  $ Ship Date    : POSIXct[1:9994], format: "2016-11-11" "2016-11-11" ...
##  $ Ship Mode    : chr [1:9994] "Second Class" "Second Class" "Second Class" "St
andard Class" ...
##  $ Customer ID  : chr [1:9994] "CG-12520" "CG-12520" "DV-13045" "SO-20335" ...
##  $ Customer Name: chr [1:9994] "Claire Gute" "Claire Gute" "Darrin Van Huff" "S
ean O'Donnell" ...
##  $ Segment      : chr [1:9994] "Consumer" "Consumer" "Corporate" "Consumer" ...
##  $ Country      : chr [1:9994] "United States" "United States" "United States"
"United States" ...
##  $ City         : chr [1:9994] "Henderson" "Henderson" "Los Angeles" "Fort Laud
erdale" ...
##  $ State        : chr [1:9994] "Kentucky" "Kentucky" "California" "Florida" ...
##  $ Postal Code  : num [1:9994] 42420 42420 90036 33311 33311 ...
##  $ Region       : chr [1:9994] "South" "South" "West" "South" ...
##  $ Product ID   : chr [1:9994] "FUR-BO-10001798" "FUR-CH-10000454" "OFF-LA-1000
0240" "FUR-TA-10000577" ...
##  $ Category     : chr [1:9994] "Furniture" "Furniture" "Office Supplies" "Furni
ture" ...
##  $ Sub-Category : chr [1:9994] "Bookcases" "Chairs" "Labels" "Tables" ...
##  $ Product Name : chr [1:9994] "Bush Somerset Collection Bookcase" "Hon Deluxe
Fabric Upholstered Stacking Chairs, Rounded Back" "Self-Adhesive Address Labels fo
r Typewriters by Universal" "Bretford CR4500 Series Slim Rectangular Table" ...
##  $ Sales        : num [1:9994] 262 731.9 14.6 957.6 22.4 ...
##  $ Quantity     : num [1:9994] 2 3 2 5 2 7 4 6 3 5 ...
##  $ Discount     : num [1:9994] 0 0 0 0.45 0.2 0 0 0.2 0.2 0 ...
##  $ Profit       : num [1:9994] 41.91 219.58 6.87 -383.03 2.52 ...
```

1. Data exploration:

Testing for missing values

```r
null<-sapply(sales, function(x) sum(is.na(x)))
print(null)
```

```
##        Row ID      Order ID    Order Date     Ship Date     Ship Mode
##             0             0             0             0             0
##   Customer ID Customer Name       Segment       Country          City
##             0             0             0             0             0
##         State   Postal Code        Region    Product ID      Category
##             0             0             0             0             0
```

```
##  Sub-Category  Product Name         Sales      Quantity      Discount
##             0             0             0             0             0
##        Profit
##             0
```

Converting order date to date type

```
Order.date <- sales$`Order Date`
dates <- as.Date(Order.date, "%m/%d/%Y")
head(dates)

## [1] "2016-11-08" "2016-11-08" "2016-06-12" "2015-10-11" "2015-10-11"
## [6] "2014-06-09"
```

Grouping data by monthly sales

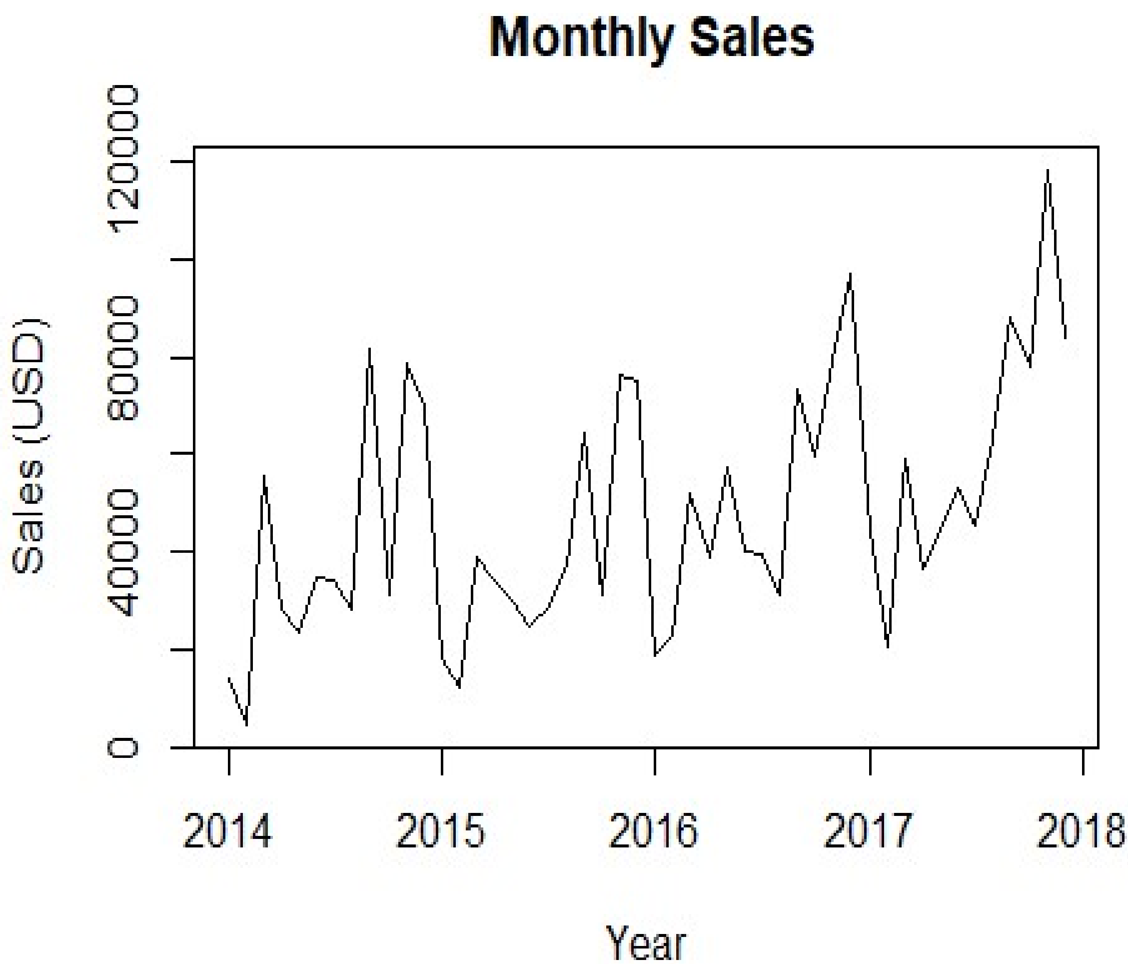
```
sale <- data.frame(dates, sales = sales$Sales)
head(sale)

##        dates    sales
## 1 2016-11-08 261.9600
## 2 2016-11-08 731.9400
## 3 2016-06-12  14.6200
## 4 2015-10-11 957.5775
## 5 2015-10-11  22.3680
## 6 2014-06-09  48.8600

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(lubridate)

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

total_sales <- sale %>%
    group_by(dates) %>%
    summarize(Total_Sale = sum(sales))

## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
print(total_sales)

## # A tibble: 1,237 x 2
##    dates      Total_Sale
##    <date>          <dbl>
##  1 2014-01-03      16.4
##  2 2014-01-04     288.
##  3 2014-01-05      19.5
##  4 2014-01-06    4407.
##  5 2014-01-07      87.2
##  6 2014-01-09      40.5
##  7 2014-01-10      54.8
##  8 2014-01-11       9.94
##  9 2014-01-13    3554.
## 10 2014-01-14      62.0
## # ... with 1,227 more rows

total_sales$Order_Date <- floor_date(total_sales$dates, 'month')

monthly_sales <- total_sales %>%
  group_by(Order_Date) %>%
  summarize(Monthly_Sales = sum(Total_Sale))

## `summarise()` ungrouping output (override with `.groups` argument)

print(monthly_sales)

## # A tibble: 48 x 2
##    Order_Date Monthly_Sales
##    <date>             <dbl>
##  1 2014-01-01        14237.
##  2 2014-02-01         4520.
##  3 2014-03-01        55691.
##  4 2014-04-01        28295.
##  5 2014-05-01        23648.
##  6 2014-06-01        34595.
##  7 2014-07-01        33946.
##  8 2014-08-01        27909.
##  9 2014-09-01        81777.
## 10 2014-10-01        31453.
## # ... with 38 more rows

monthly_sales<-ts(monthly_sales[-1], frequency = 12,start=c(2014,01),end=c(2017,12
))
print(monthly_sales)

##            Jan       Feb       Mar       Apr       May       Jun
## 2014  14236.895  4519.892 55691.009 28295.345 23648.287 34595.128
## 2015  18174.076 11951.411 38726.252 34195.209 30131.686 24797.292
## 2016  18542.491 22978.815 51715.875 38750.039 56987.728 40344.534
## 2017  43971.374 20301.133 58872.353 36521.536 44261.110 52981.726
```

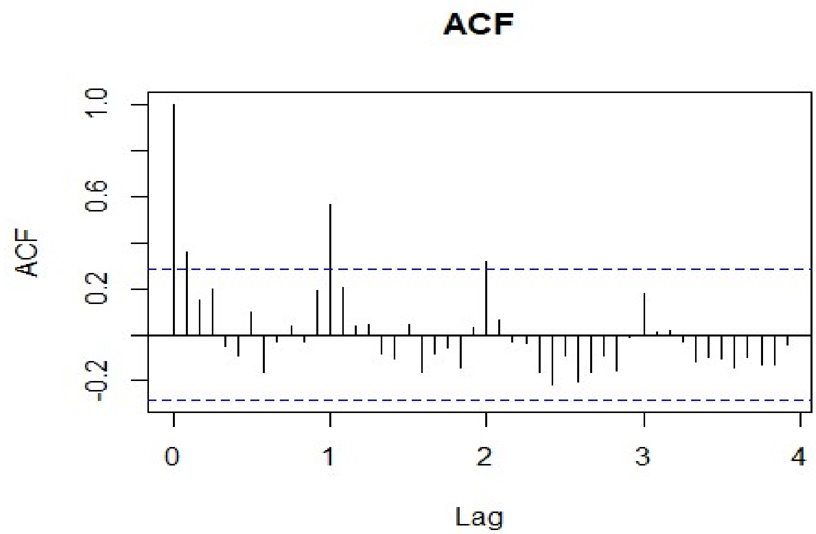
```
##              Jul        Aug        Sep        Oct        Nov        Dec
## 2014  33946.393  27909.468  81777.351  31453.393  78628.717  69545.621
## 2015  28765.325  36898.332  64595.918  31404.924  75972.564  74919.521
## 2016  39261.963  31115.374  73410.025  59687.745  79411.966  96999.043
## 2017  45264.416  63120.888  87866.652  77776.923 118447.825  83829.319
```

Check data

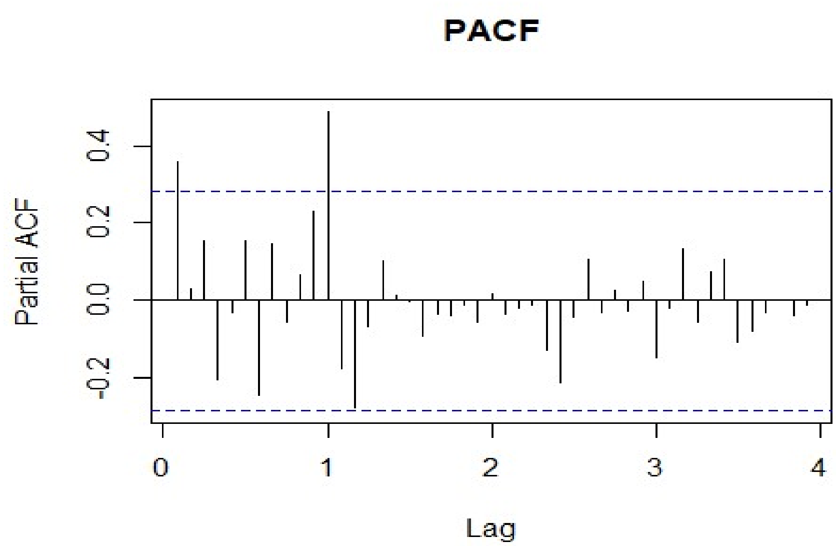
```
plot(monthly_sales,xlab="Year",ylab="Sales (USD)",main=" Monthly Sales")
```




```
acf(monthly_sales, 50,main = 'ACF')
```



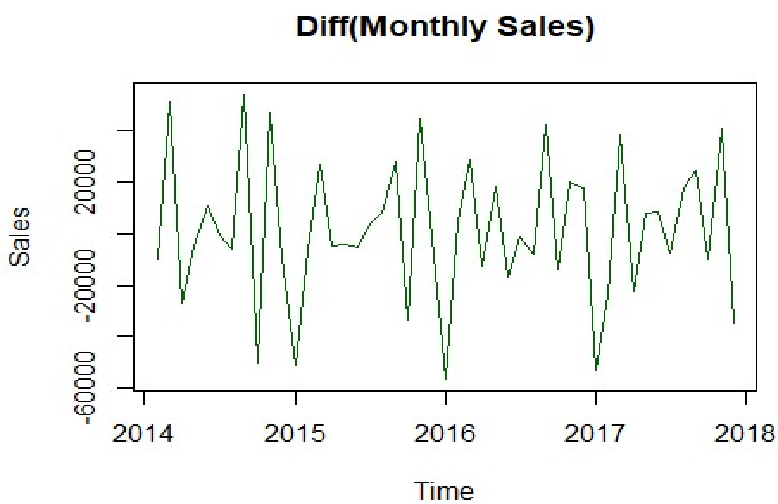

```
pacf(monthly_sales, 50,main = 'PACF')
```



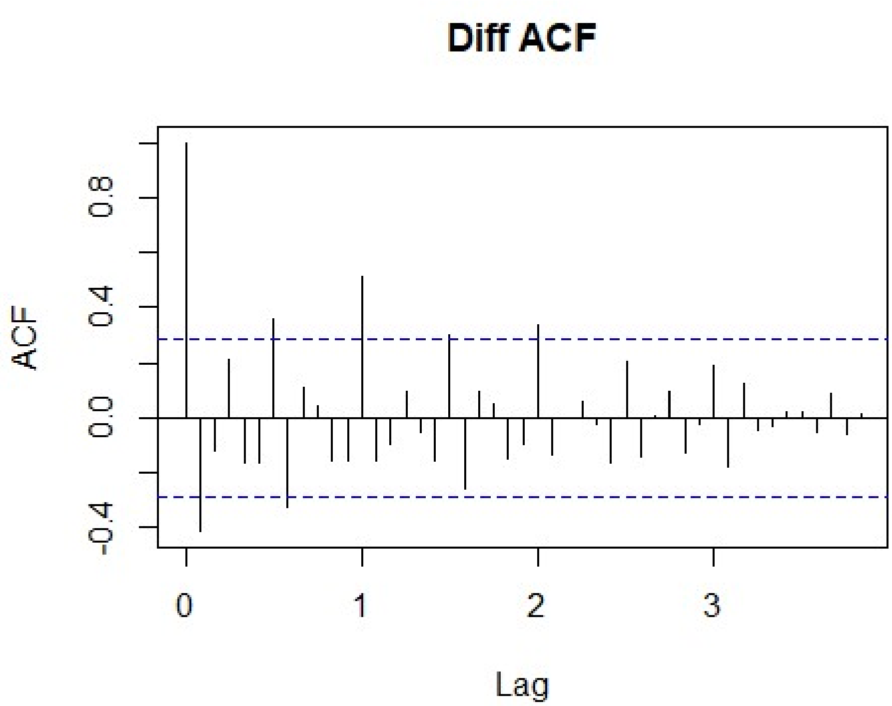
From the plot, we can observe that the monthly sales is not constant with seasonality component since sales are low at the beginning of the year and high at the end of the year. As we know that a stationary time series is one whose properties do not depend on time, time series with trends, or with seasonality, are not stationary. As a result, it can be observed that the series are not stationary. From the PACF plot, it appears large spike at lag 1 and then followed by a wave that alternates between positive and negative correlations. There is no AR process indicated.

Data transformation

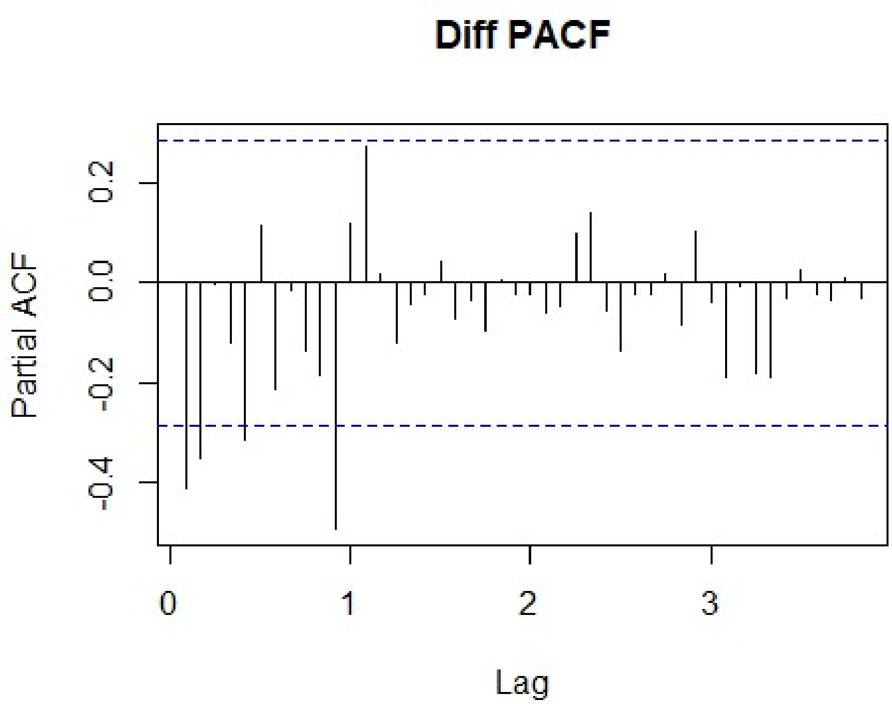
```
par(mfrow=c(1,1))
plot(diff(monthly_sales), main="Diff(Monthly Sales)", col="darkgreen", ylab = "Sal
es")
```




```
acf(diff(monthly_sales),50, main = 'Diff ACF' )
```



```r
pacf(diff(monthly_sales),50, main = 'Diff PACF')
```



By looking at the plot of the differenced monthly sales, it appears more stationarized since there is no upward trend. From the ACF plot above, it appears large spike at lag 1. Thus MA(1) model should be included.

2. Model fitting

The appropriate model is ARIMA(0,1,1)(0,1,1).

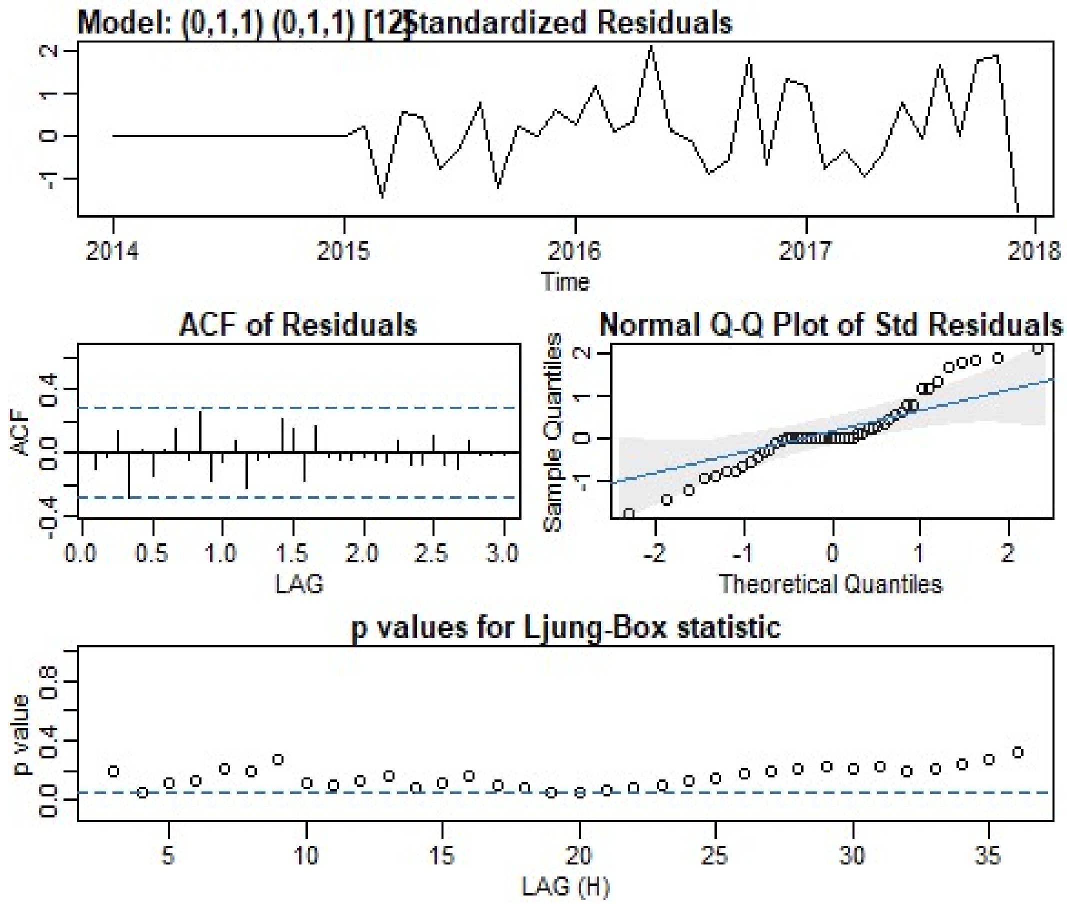
```r
library(forecast)

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

##
## Attaching package: 'forecast'

## The following object is masked from 'package:astsa':
##
##     gas

fit.model = arima(monthly_sales, order = c(0,1,1), seasonal = list(order = c(0,1,1
)))
summary(fit.model)
```

```
##
## Call:
## arima(x = monthly_sales, order = c(0, 1, 1), seasonal = list(order = c(0, 1,
##     1)))
##
## Coefficients:
##           ma1     sma1
##       -0.8212  -0.7559
## s.e.   0.1049   0.6452
##
## sigma^2 estimated as 121190440:  log likelihood = -380.37,  aic = 766.74
##
## Training set error measures:
##                    ME     RMSE      MAE      MPE     MAPE      MASE       ACF1
## Training set 1530.637 9400.457 6383.278 2.152934 13.86926 0.2948091 -0.1090972
```

3. Model diagnosis

```
par(mar=c(2,2,2,4))
sarima(monthly_sales, 0,1,1,0,1,1,12)
```

```
## initial  value 9.796300
## iter   2 value 9.551243
## iter   3 value 9.461603
## iter   4 value 9.423985
## iter   5 value 9.420004
## iter   6 value 9.418815
## iter   7 value 9.417964
## iter   8 value 9.417280
## iter   9 value 9.417278
## iter   9 value 9.417278
## iter   9 value 9.417278
## final  value 9.417278
## converged
## initial  value 9.451890
## iter   2 value 9.450261
## iter   3 value 9.449544
## iter   4 value 9.449102
## iter   5 value 9.448847
## iter   6 value 9.448734
## iter   7 value 9.448728
## iter   8 value 9.448724
## iter   9 value 9.448723
## iter  10 value 9.448718
## iter  11 value 9.448717
## iter  12 value 9.448717
## iter  12 value 9.448717
## iter  12 value 9.448717
## final  value 9.448717
## 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, transform.pars = trans, fixed
= fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1     sma1
##       -0.8212  -0.7559
## s.e.   0.1049   0.6452
##
## sigma^2 estimated as 121190440:  log likelihood = -380.37,  aic = 766.74
##
## $degrees_of_freedom
## [1] 33
##
## $ttable
##      Estimate     SE t.value p.value
## ma1   -0.8212 0.1049 -7.8258  0.0000
## sma1  -0.7559 0.6452 -1.1715  0.2498
##
```

```
## $AIC
## [1] 16.66817
##
## $AICc
## [1] 16.67424
##
## $BIC
## [1] 16.76961
```

The plots suggest that the model residuals seem normally distributed.

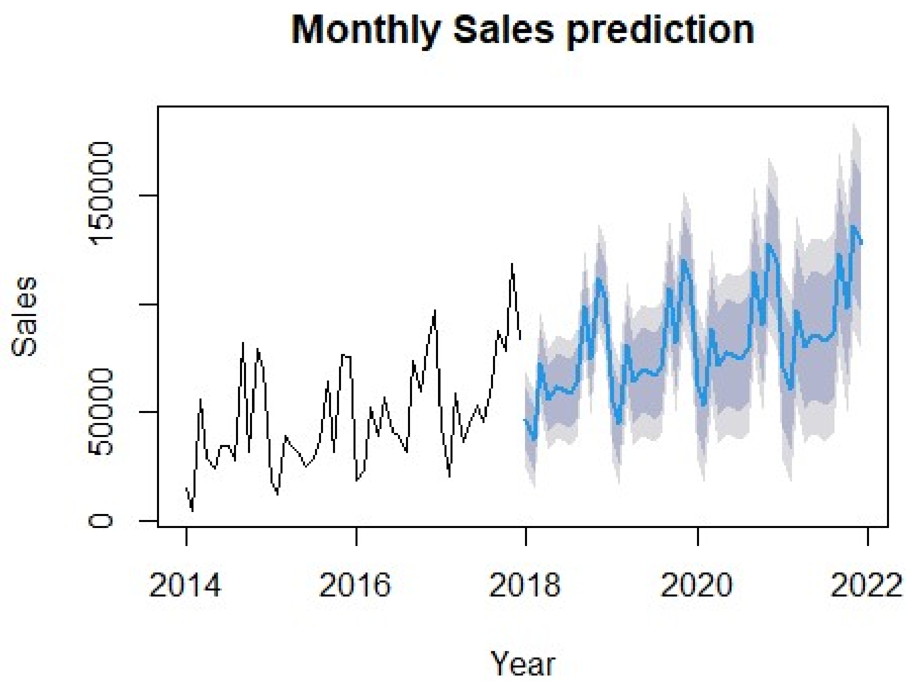
4. Model Forecast

We can predict next 4 years monthly sales by

```
predict.model = predict(fit.model, n.ahead = 48)
print(predict.model)
```

```
## $pred
##           Jan       Feb       Mar       Apr       May       Jun       Jul
## 2018  46565.13  36923.66  72936.38  55940.29  61178.12  60687.03  58754.91
## 2019  54496.63  44855.16  80867.88  63871.79  69109.61  68618.52  66686.41
## 2020  62428.12  52786.65  88799.37  71803.28  77041.11  76550.02  74617.90
## 2021  70359.62  60718.15  96730.87  79734.78  84972.60  84481.51  82549.40
##           Aug       Sep       Oct       Nov       Dec
## 2018  62791.28  98749.89  74224.10 111720.07 103338.62
## 2019  70722.77 106681.39  82155.60 119651.56 111270.12
## 2020  78654.27 114612.88  90087.09 127583.06 119201.61
## 2021  86585.76 122544.37  98018.58 135514.55 127133.11
##
## $se
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2018 11396.64 11574.03 11748.74 11920.89 12090.58 12257.93 12423.03 12585.96
## 2019 14330.98 14559.25 14784.00 15005.38 15223.54 15438.61 15650.74 15860.02
## 2020 17826.84 18093.45 18356.20 18615.23 18870.71 19122.77 19371.56 19617.19
## 2021 21741.33 22038.75 22332.20 22621.85 22907.84 23190.30 23469.37 23745.15
##           Sep      Oct      Nov      Dec
## 2018 12746.80 12905.64 13062.56 13217.60
## 2019 16066.58 16270.52 16471.93 16670.91
## 2020 19859.78 20099.45 20336.29 20570.40
## 2021 24017.77 24287.32 24553.92 24817.65
```

```
plot(forecast(fit.model,48), main = 'Monthly Sales prediction', ylab = 'Sales', xl
ab = 'Year')
```



From the prediction, it appears that the monthly sales are not stable with seasonality component since the sales are generally low at the beginning of every year and high at the end of every year.