# Project Time Series - forecasting

Agata Dratwa, Agata Jankowska, Piotr Buczek

Dataset that we are going to use for conducting forecast represents monthly average temperature in Lagos since 1900 to 2012. Capitol of Nigeria was selected as a main concern of our work since the city is one of the most vulnerable places on the Earth to climate change consequences. **The goal of the project is to show that average temperature in Lagos has risen significantly over the years.**

# 1. Analysis of dataset

There is total of 1356 observations.

Average temperature is presented in Celsius degrees.

```
library(astsa)
library(xts)
library(zoo)
library(forecast)
library(Quandl)
library(tseries)
```

```
dat <- read.csv("out.csv")
```

Using imported dataset, we change data into time series object for further analysis.

```
data <- ts(dat$Temp, frequency=12, start=c(1900,1))
```

### Summary of data

Temperature varies from 24.5 degrees up to 29 degrees.

```
summary(data)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   24.55   26.17   26.79   26.76   27.31   29.03
```

First we will split data on train and test set. Train data will be from 1900 to 2012 and test set with year 2012.

```r
fstT <- 1900+0/12
lstT <- 2011+11/12
data.train = window(data, end=lstT)
data.test = window(data, start=lstT+1/12)
```

From the plot of data we can see evolution of the time series over time.

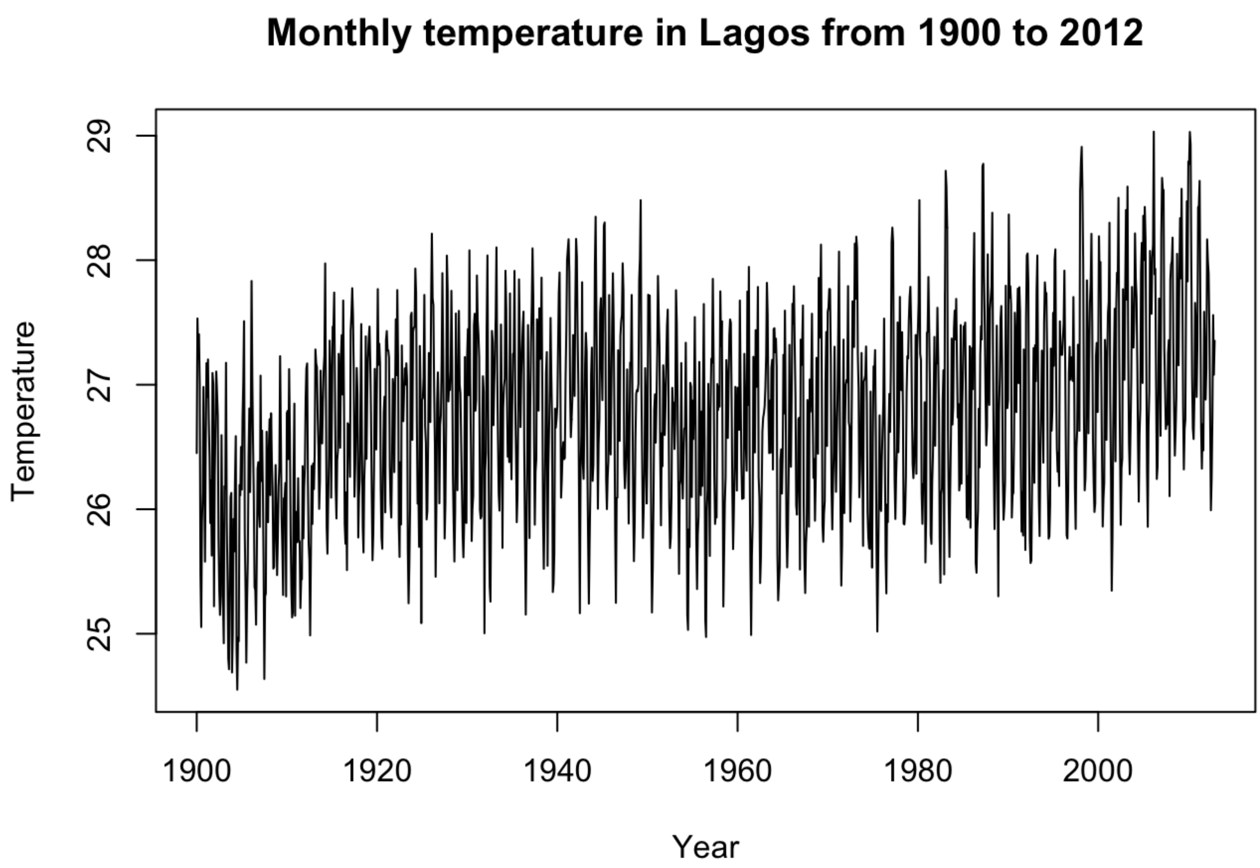
First analysis shows that temperature, during hundred of years, raised - so we can assume a trend.

We can clearly see that yearly data may have seasonality (so each year the data can vary in the same seasonal way through the months)

Amplitude in cycles seem stable over time so there is no heterocedasticy in data.

There is no visible outliers.

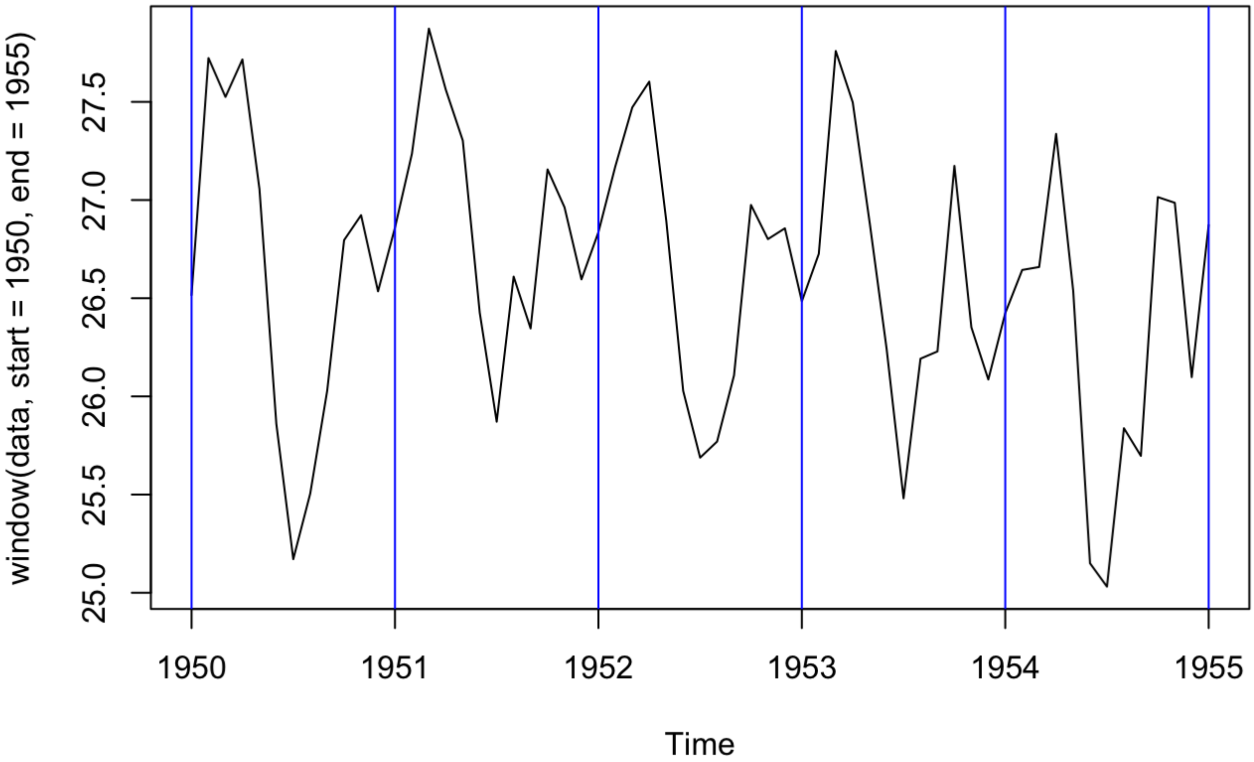
```r
plot.ts(data, xlab='Year', ylab='Temperature', main='Monthly temperature in Lagos
 from 1900 to 2012')
```




Closer look at cycles during years 1950-1955 shows yearly seasonality.

```r
plot.ts(window(data,start=1950,end=1955))
abline(v=1950, col="blue")
abline(v=1951, col="blue")
abline(v=1952, col="blue")
abline(v=1953, col="blue")
```

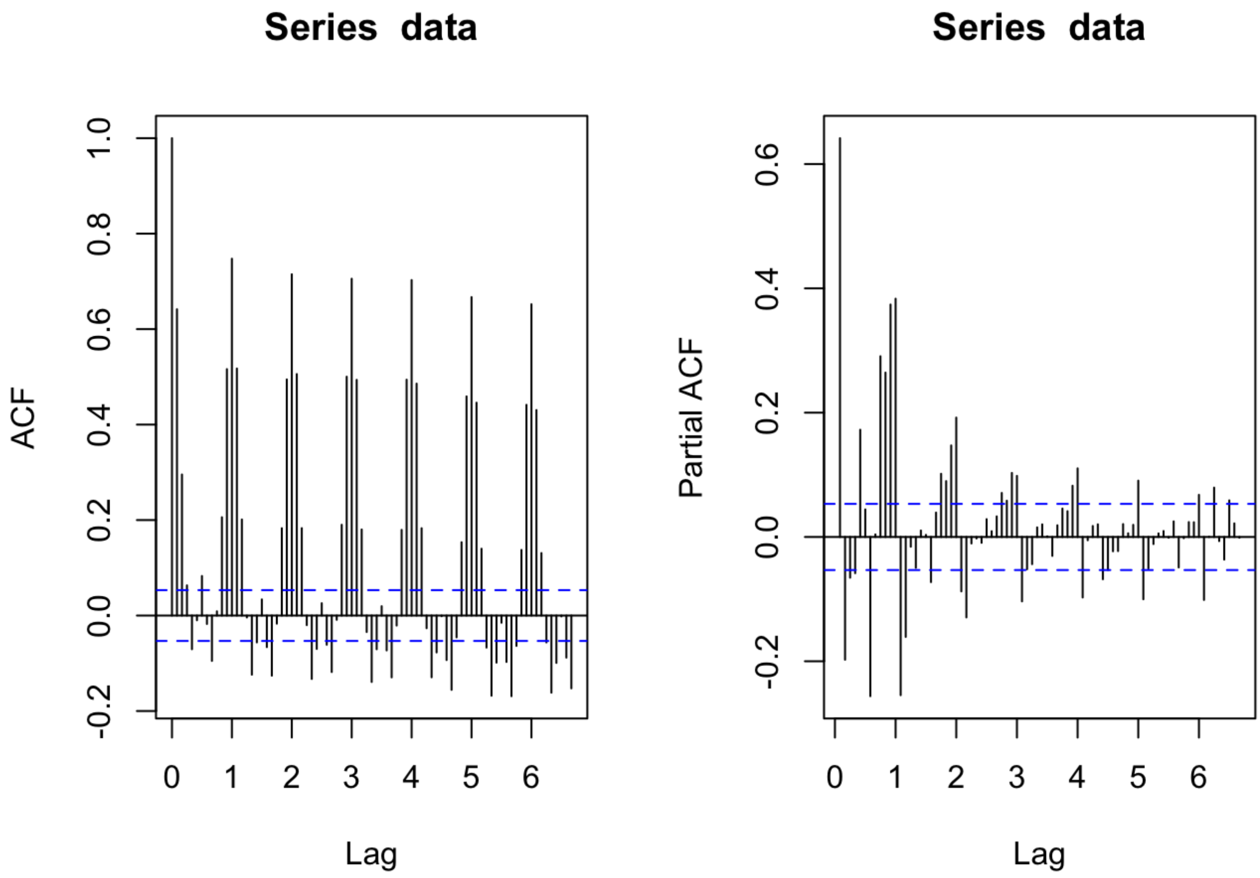
```
abline(v=1954, col="blue")
abline(v=1955, col="blue")
```




Both trend and seasonality features are present in ACF plot. There is visible trend because ACF decreases very slowly and there is a seasonality presented by cycles of 12 months.

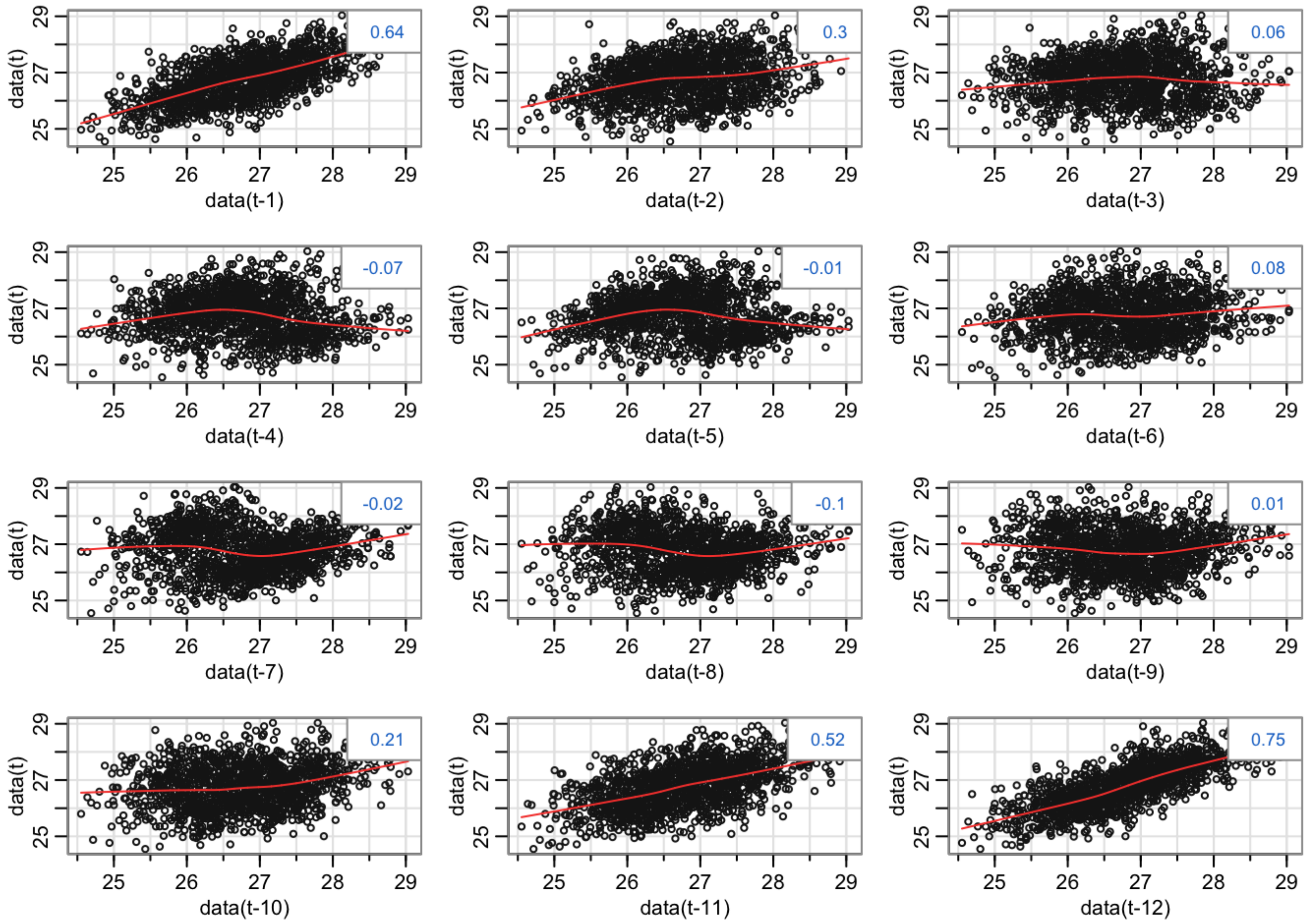
PACF plot decays to zero.

```
par(mfrow=c(1,2))
acf(data,80)
pacf(data,80)
```




Correlation in data separated for 1 year is positive and strong: 0.75.

```
lag1.plot(data,12)
```



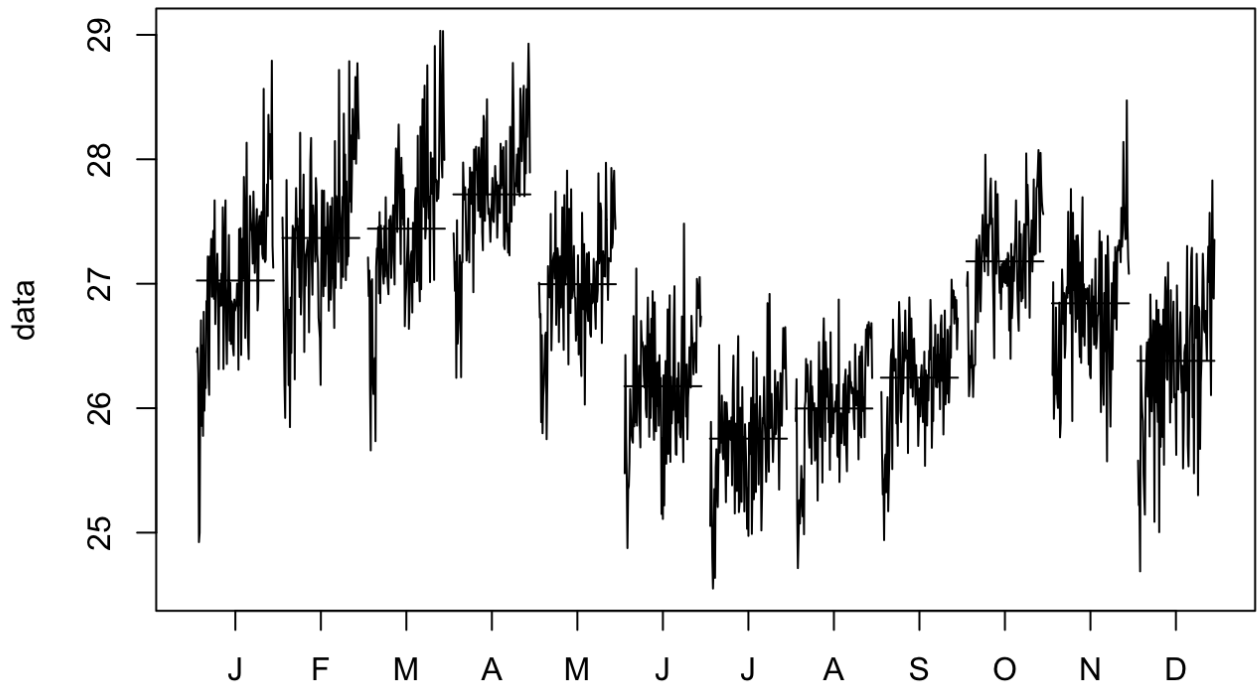

Monthplot shows seasonality over year.

Temperature changes during year.

The smallest temperature is in June to September and the biggest around March and April.

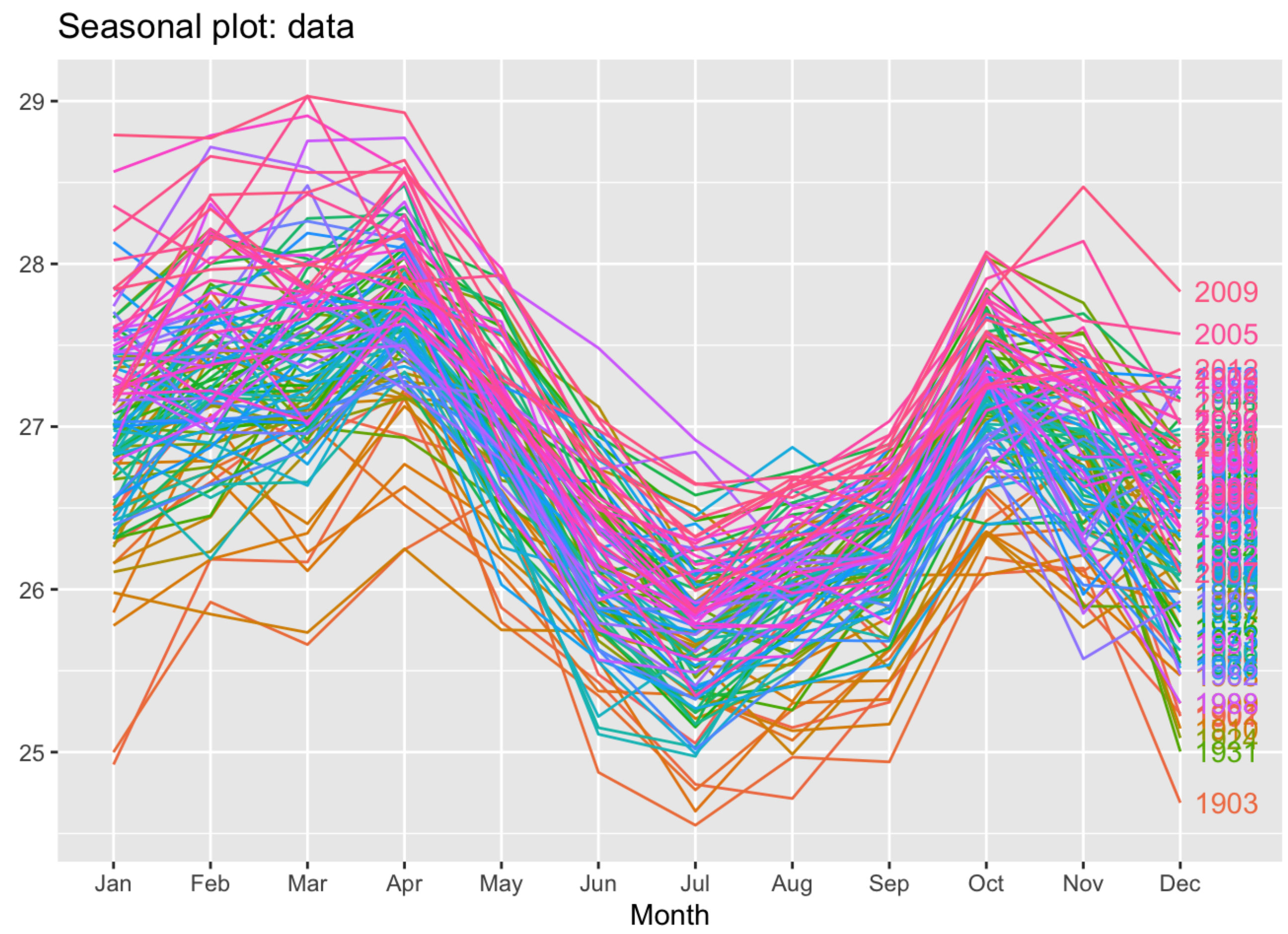
This plot shows clearly seasonal pattern and also shows the changes in seasonality over time. We can see increasing values in each month for each year.

```
monthplot(data)
```




In ggseasonplot it is visible delicatly increasing trend and yearly seasonality.

```
ggseasonplot(data, year.labels=TRUE)
```

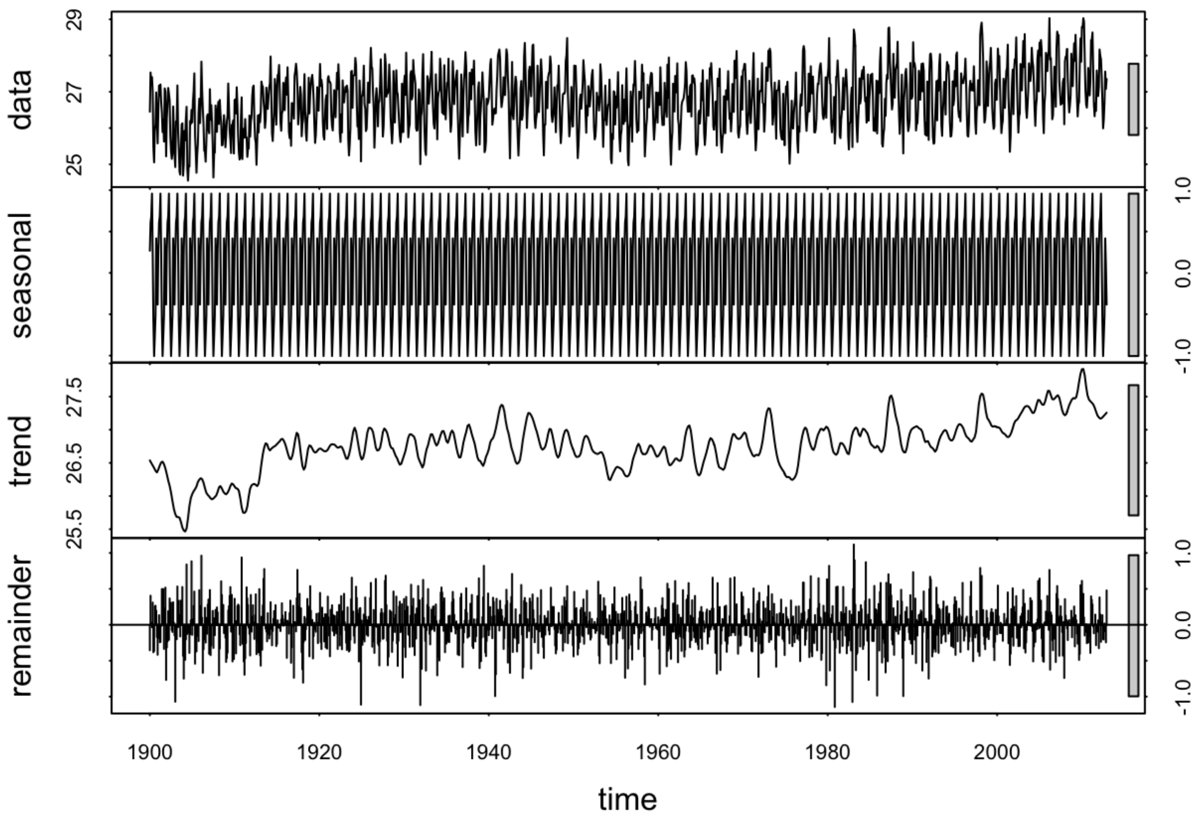



After decomposition of data we can see three components: Seasonal, trend, reminder.

Trend has increasing tendancy.

Seasonality is present through all data.

```
data.stlper=stl(data, s.window="periodic")
plot(data.stlper)
```




**Main conclusions**

Data is not stationary because it contains trend and seasonality.

There is no outliers so there is no need to clean the data.

# 2. Preparing stationary data - deseasonality and detrending

From previous analysis we assume that data is not stationary.

Next step is deciding which type of differences we need to stationarise the series. In other words we have to find how many unit roots we should consider in our model.

First of all, using these two functions we want to check if we need both trend and seasonal differencing. We can see that both are needed

```
ndiffs(data.train)
```

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

```
nsdiffs(data.train)
```

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

To detrend data we use filtering with the difference operator:

$$\nabla x_t = (1 - B)x_t = x_t - Bx_t = x_t - x_{t-1}$$

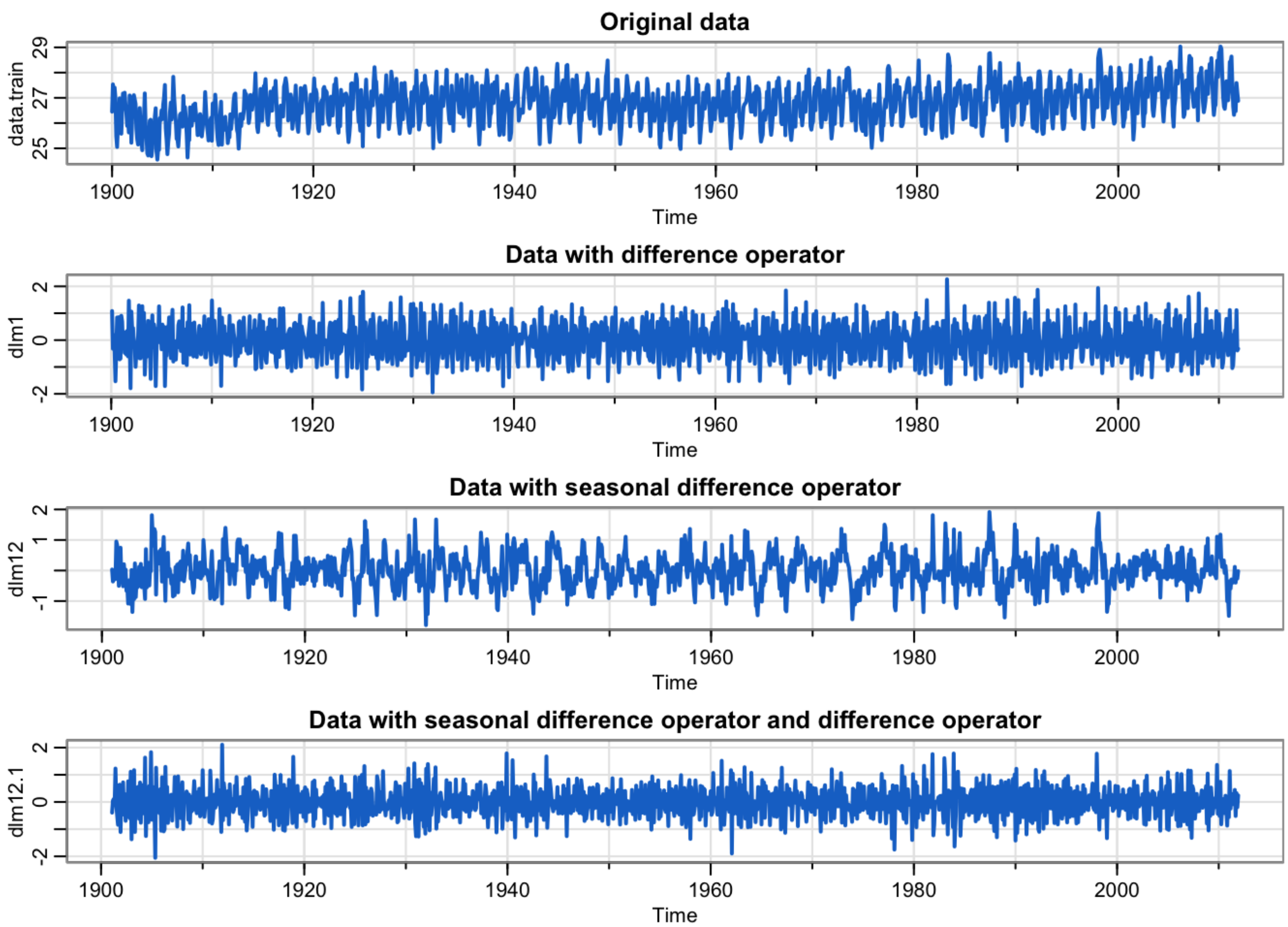
To remove seasonality from data we use seasonal difference operator:

$$\nabla^S x_t = x_t - x_{t-S}$$

### Plots with original data and with differenced data.

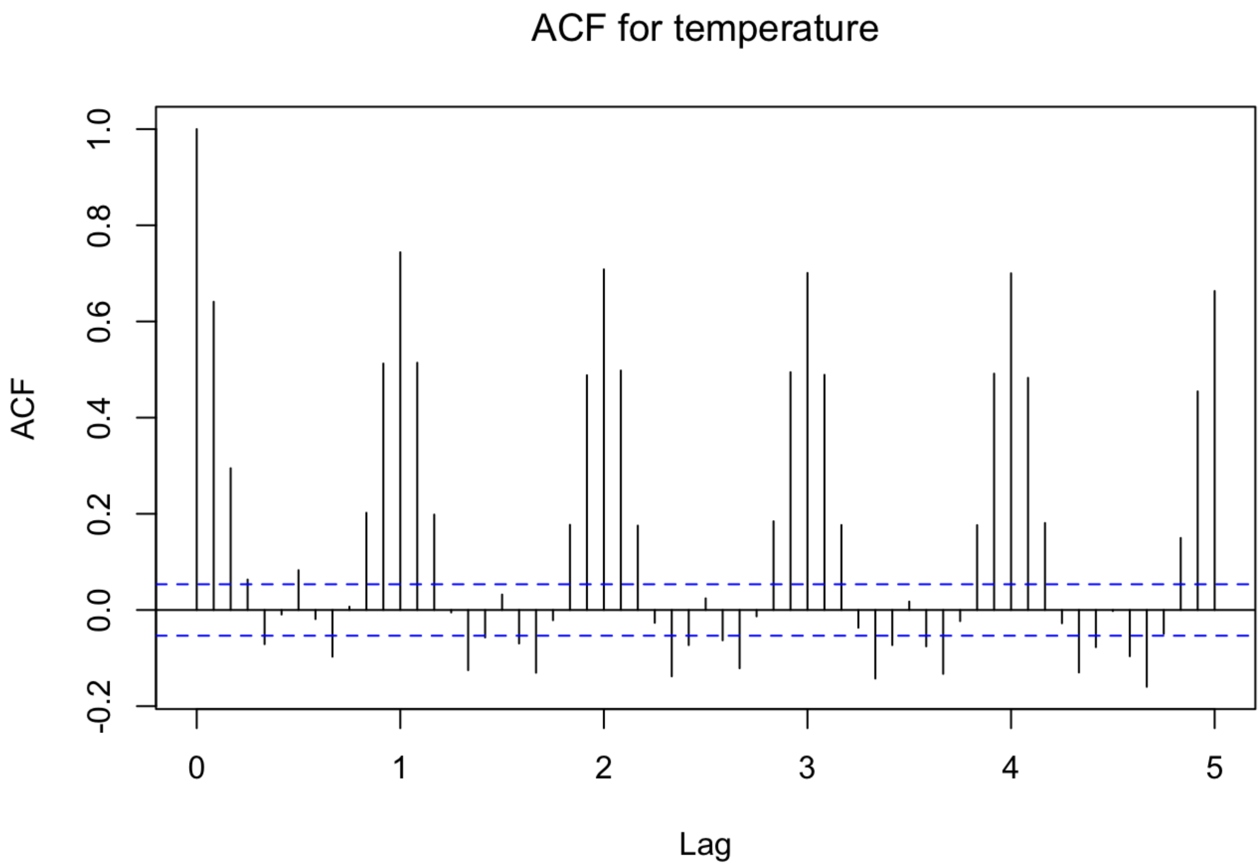
```
dlm1 = diff(data.train,1)
dlm12 = diff(data.train,12)
dlm12.1 = diff(diff(data.train,12),1)

par(mfrow=c(4,1))
tsplot(data.train, col=4, lwd=2, main="Original data")
tsplot(dlm1, col=4, lwd=2, main="Data with difference operator")
tsplot(dlm12, col=4, lwd=2, main="Data with seasonal difference operator")
tsplot(dlm12.1, col=4, lwd=2, main="Data with seasonal difference operator and dif
ference operator")
```

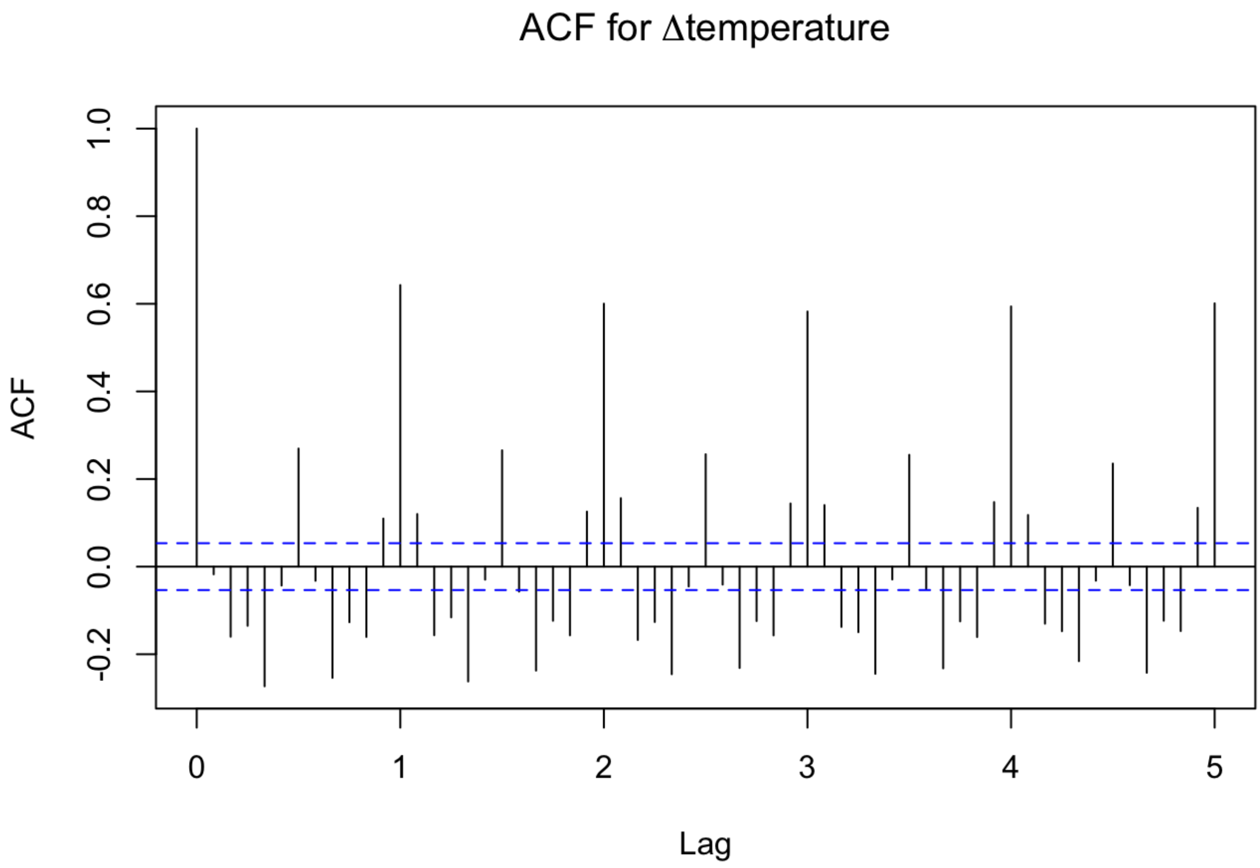



Plotting ACF of each time series in order to find the one which is stationary.

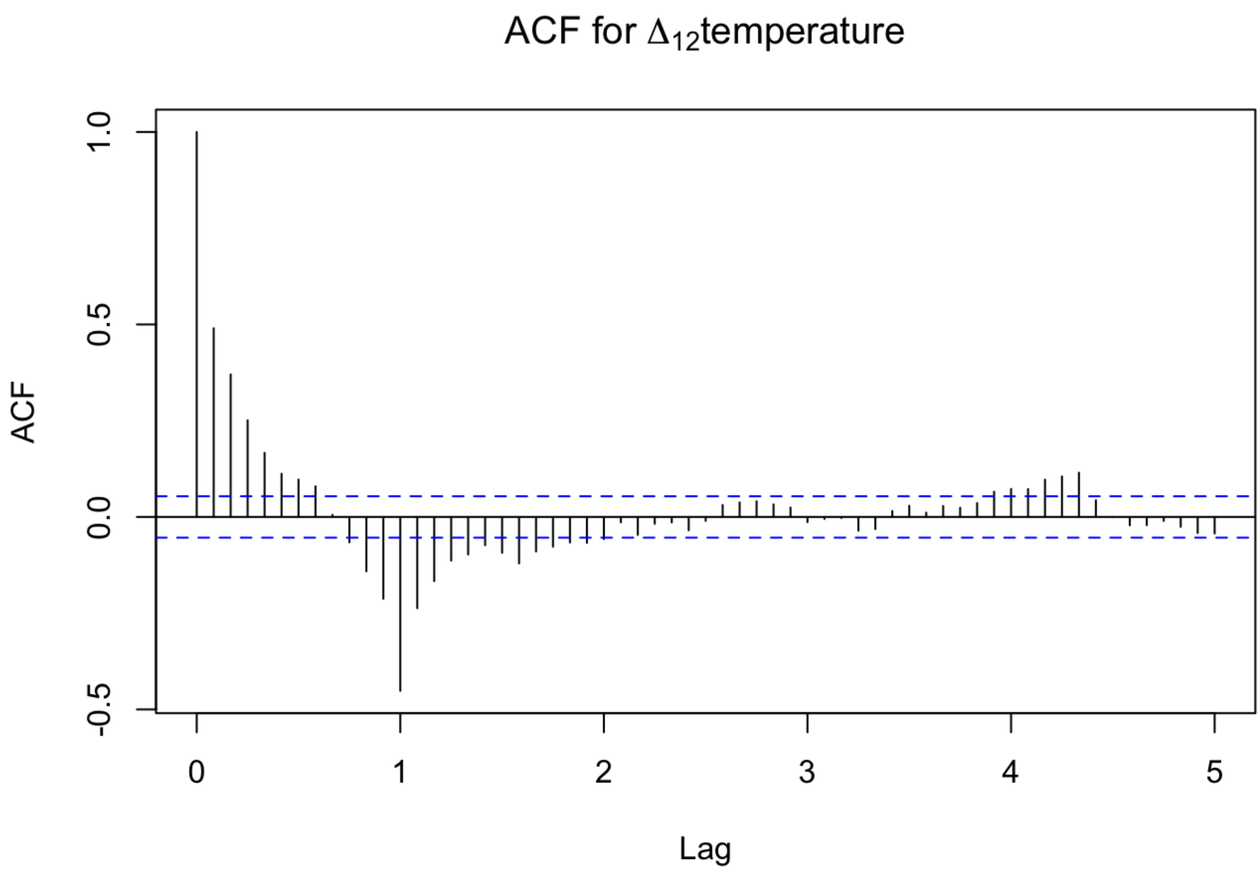
```r
acf(data.train, 60, main=expression(paste("ACF for temperature")))
```



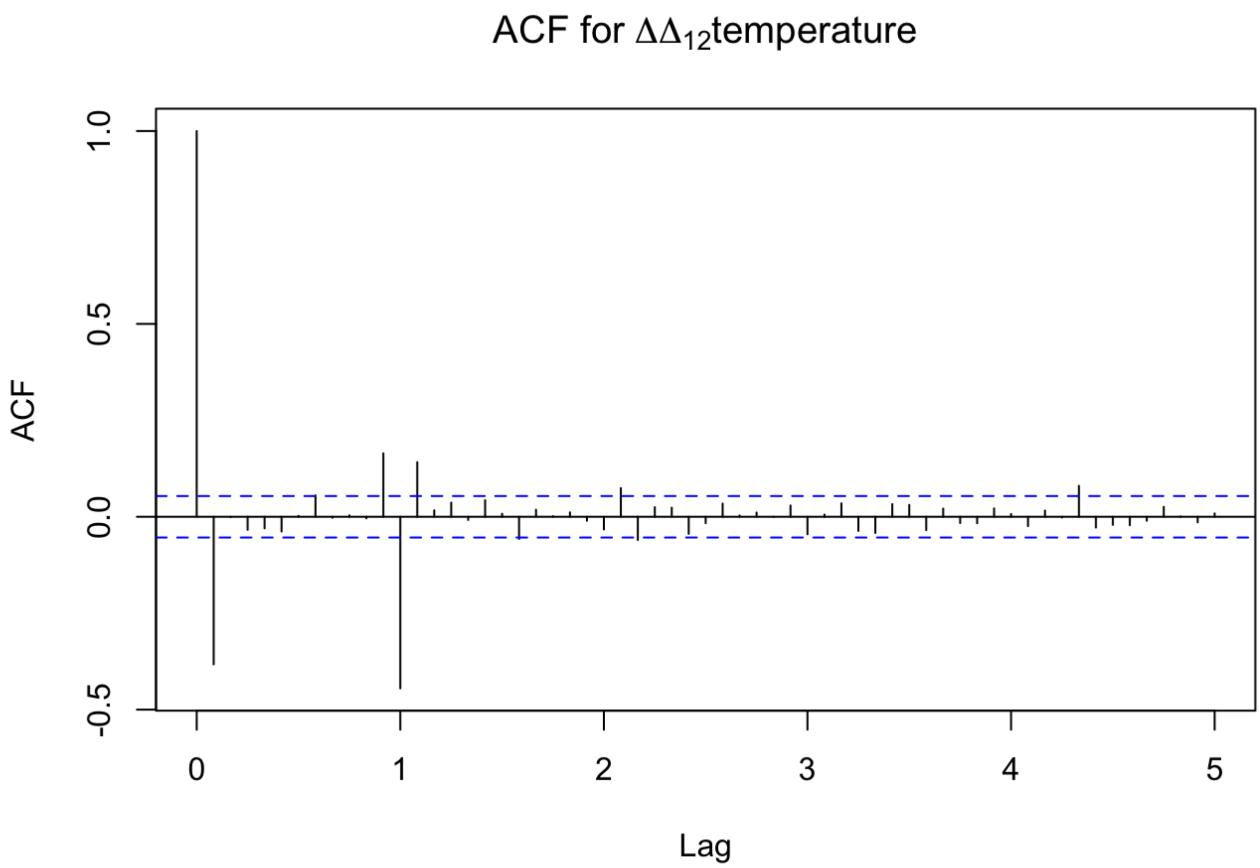

```r
acf(dlm1, 60, main=expression(paste("ACF for ", Delta,"temperature")))
```




```r
acf(dlm12, 60, main=expression(paste("ACF for ", Delta[12], "temperature")))
```

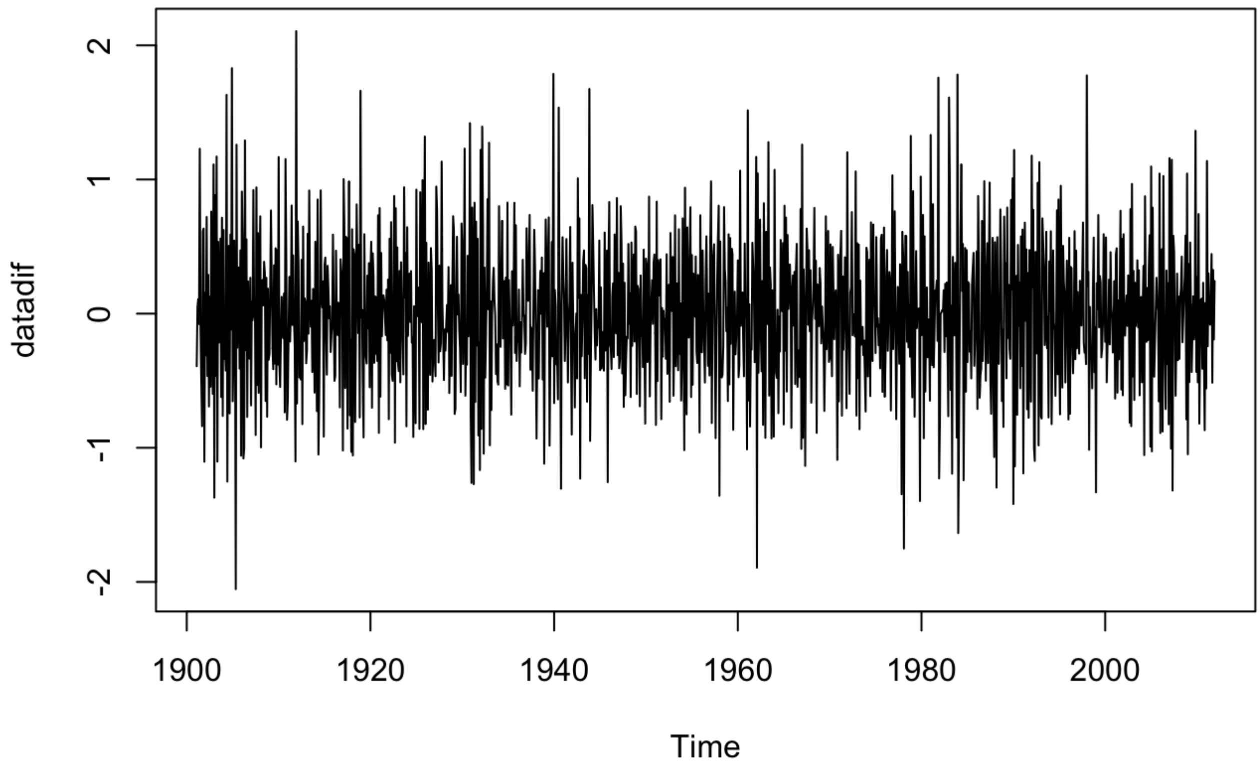


```
acf(dlm12.1, 60, main=expression(paste("ACF for ", Delta, Delta[12], "temperature"
)))
```




From the plots above we can see that seasonally (period = 12) and regularly differenced data seems more stationary that others because it drops quickly to zero. Therefore, this time series will be chosen for further analysis.

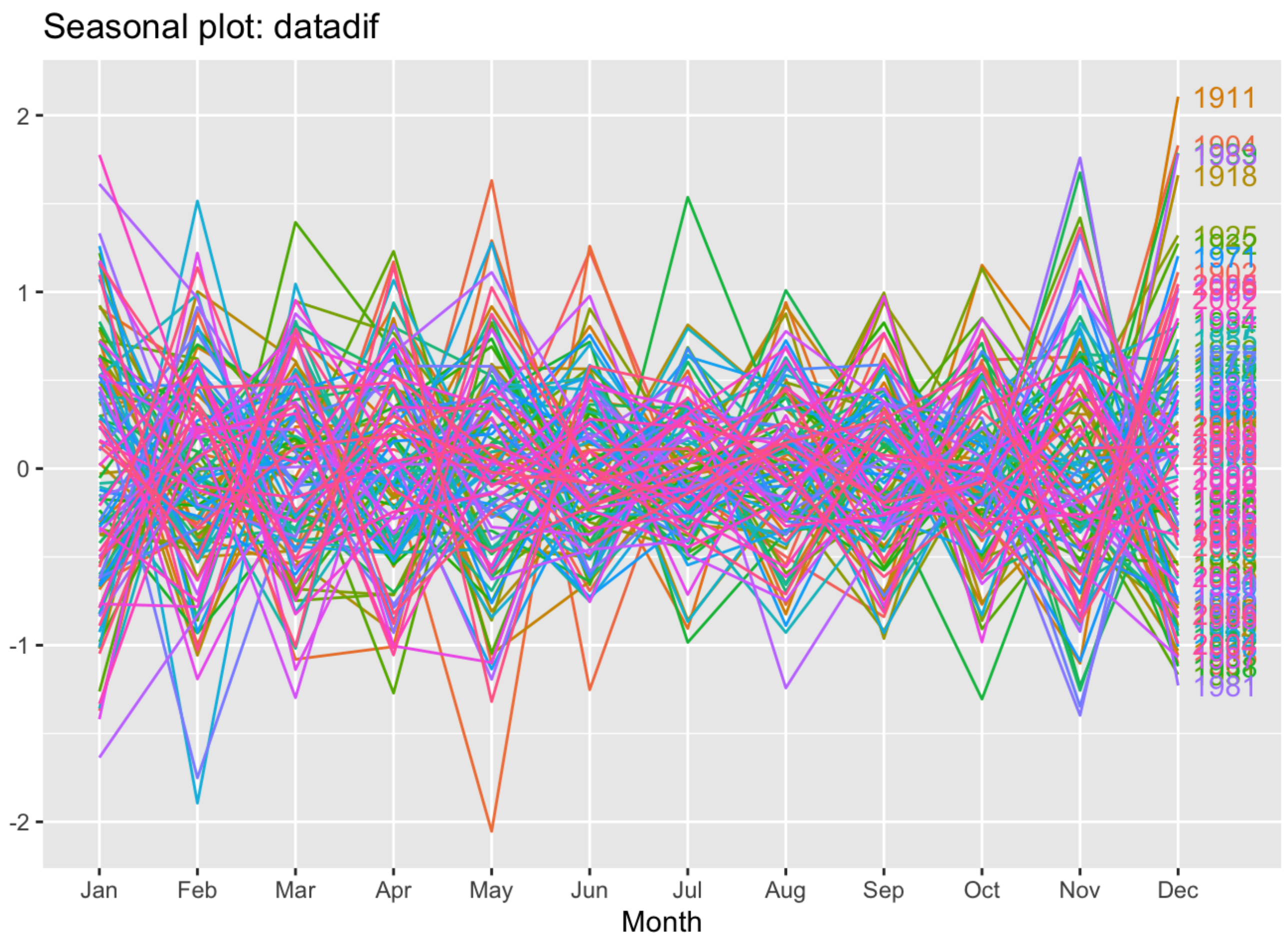
```
datadif<-dlm12.1
plot(datadif)
```



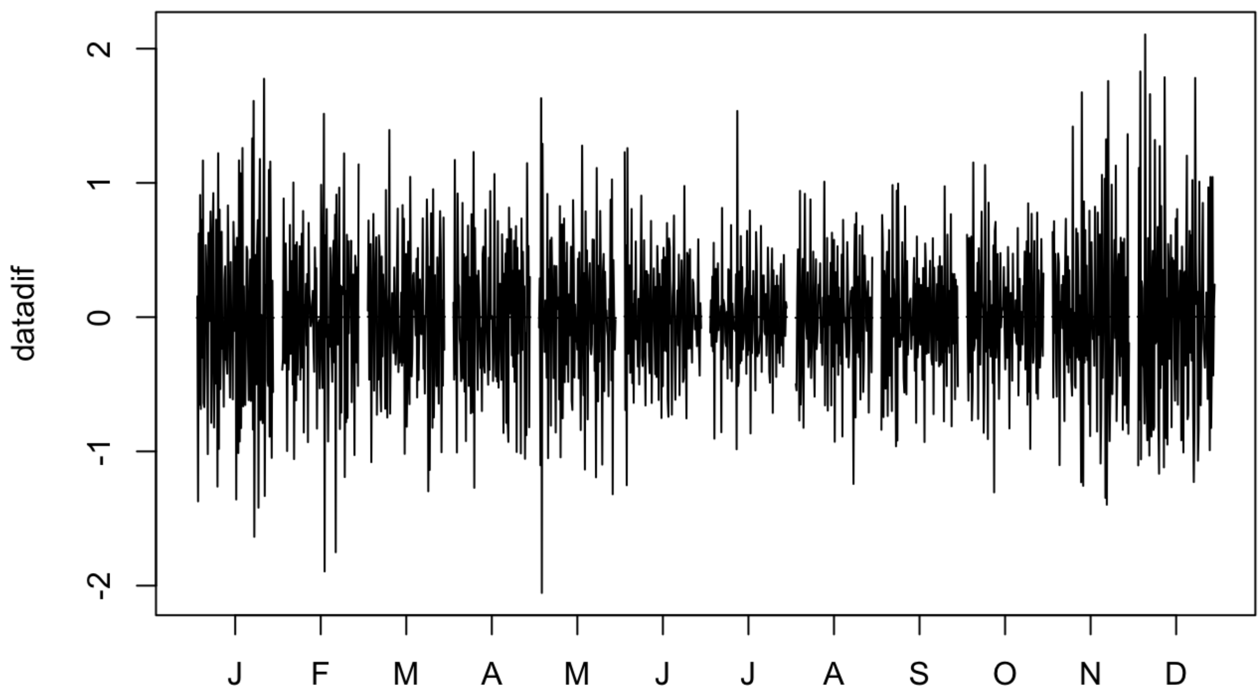

Now we can check if there is no more seasonality/trend using ggseasonplot and monthplot.

From those two plots we can conclude that there is no more seasonality and trend in our data.

```
ggseasonplot(datadif, year.labels=TRUE)
```




```
monthplot(datadif)
```



### Stationarity check

To check stationarity of our data we use Augmented Dickey Fuller Test and Kwiatkowski-Phillips-Schmidt-Shin Test.

```
adf.test(datadif)
```

```
## Warning in adf.test(datadif): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  datadif
## Dickey-Fuller = -10.358, Lag order = 10, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(datadif)
```

```
## Warning in kpss.test(datadif): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  datadif
## KPSS Level = 0.003026, Truncation lag parameter = 7, p-value = 0.1
```

### Results

ADF test: p-value:0.01 is smaller than significance level of 0.05. Due to that we have reason to reject null hypothesis so data is stationary.

KPSS test: p-value:0.1 is bigger than 0.05 so we dont reject null hypothesis and we conclude that data is stationary.

From both tests we conclude that data is stationary.

# 3. Modeling the data

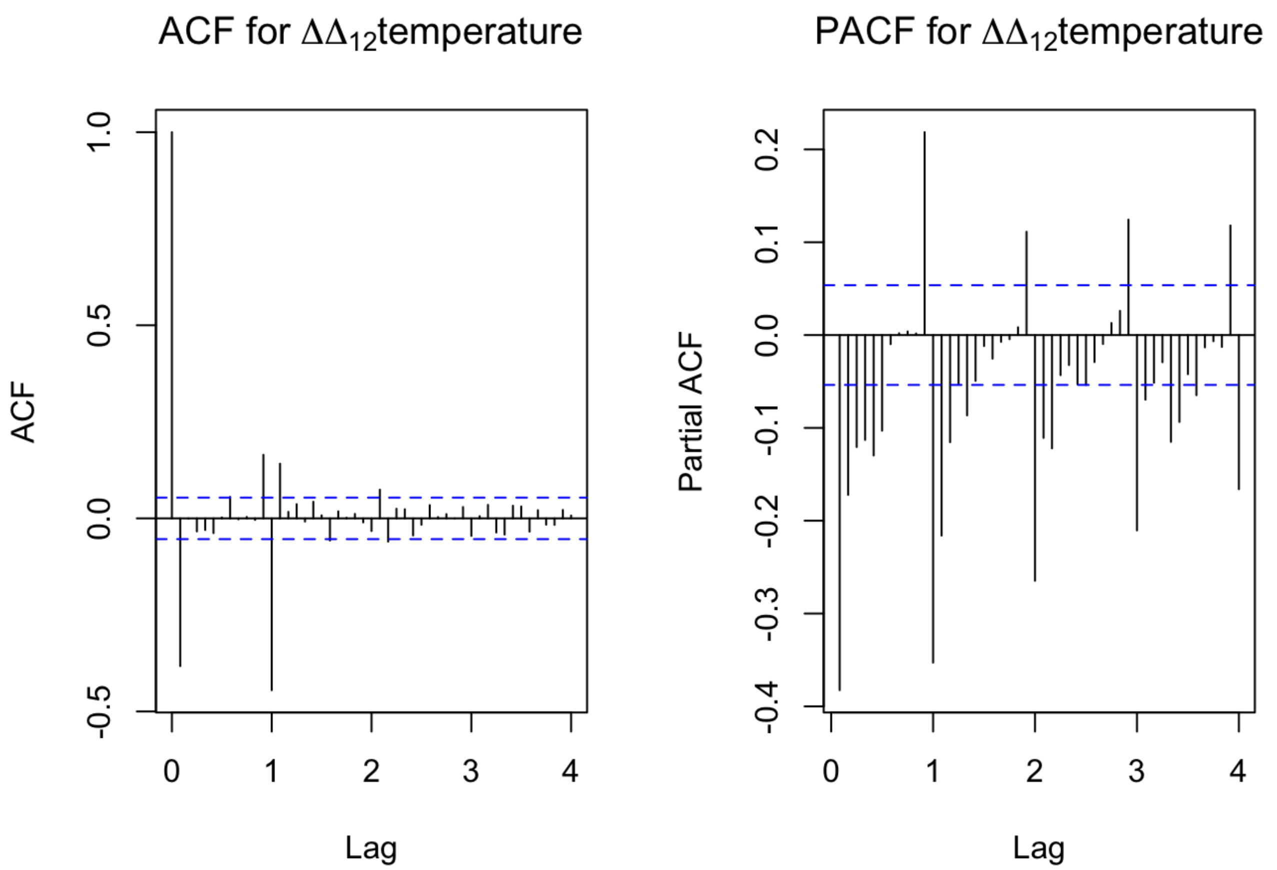
From previous analysis we know that in order to make the data stationary we had to make seasonal differencing and a regular one.

This means that a model will include a unit root d=1 as well as a seasonal unit root D=1.

In order to find adequate values for AR and MA components (regular and seasonal) now we have to analyse ACF and PACF plot.

```
par(mfrow=c(1,2))
acf(datadif, 48, main=expression(paste("ACF for ", Delta, Delta[12], "temperature"
```

```
)))
pacf(datadif,48, main=expression(paste("PACF for ", Delta, Delta[12], "temperatur
e")))
```




From the above plots we can observe that ACF decays to zero quicker than PACF what suggest MA model. In addition ACF plot shows significant correlation at lag 1 and 11. Also we can observe a significant correlation at lag 12 and 24 which indicates seasonal MA component in our model. In the next step we will try a couple of different models.

For every model we check its adequacy by analyzing their residuals and significance of each of estimated parameters.

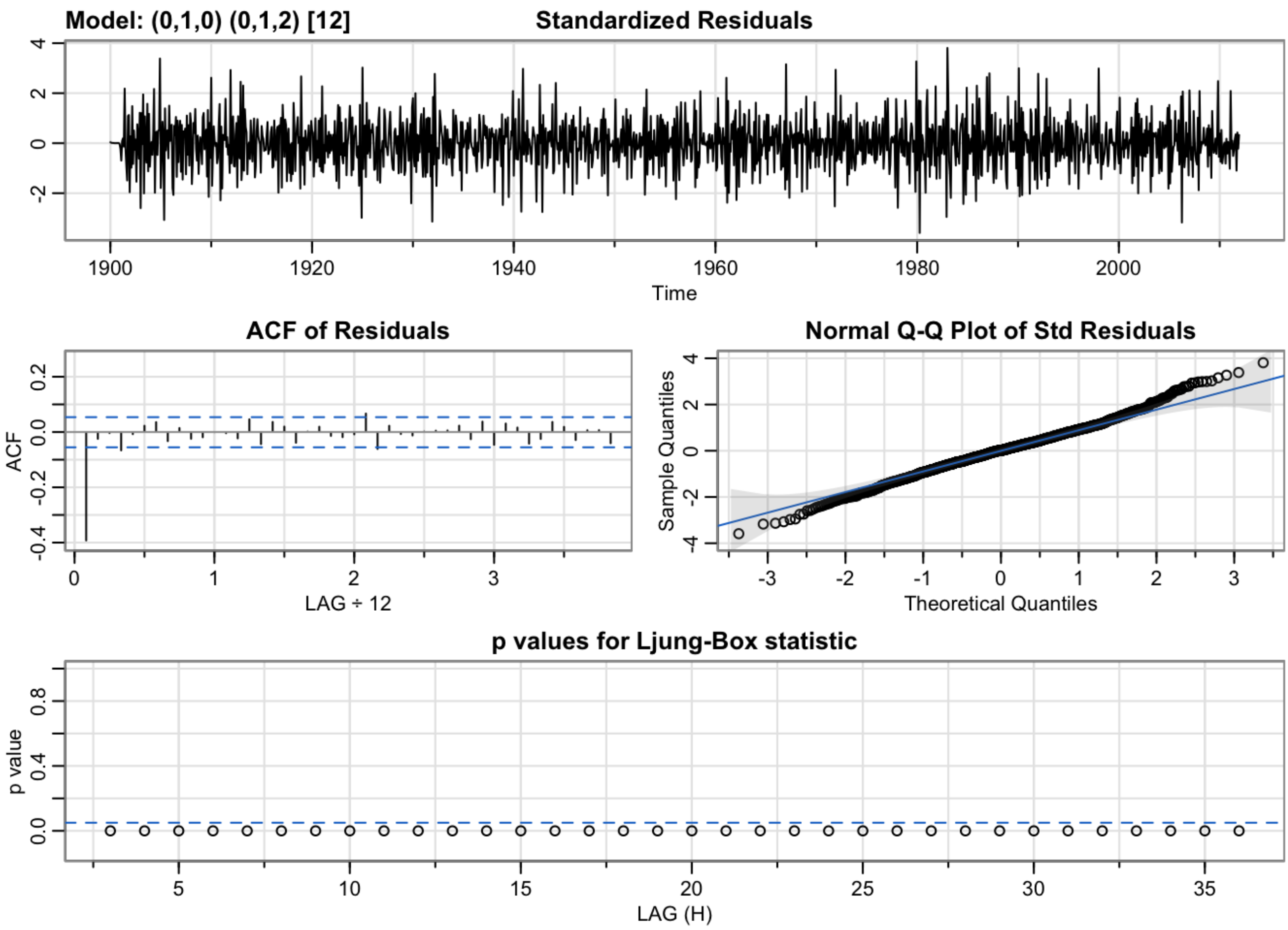
## Model 1

Significant spike at lag 24 suggests a seasonal MA(2) component. Consequently, taking into account regular and seasonal differencing, we will build a **SARIMA (0,1,0)x(0,1,2)12**

```
model_1<- sarima(data.train,0,1,0,0,1,2,12)
```

```
## initial  value -0.572584
## iter   2 value -0.738334
## iter   3 value -0.738560
## iter   4 value -0.746692
## iter   5 value -0.752184
## iter   6 value -0.818855
## iter   7 value -0.822590
```

```
## iter   8 value -0.835439
## iter   9 value -0.836092
## iter  10 value -0.836097
## iter  10 value -0.836097
## iter  10 value -0.836097
## final  value -0.836097
## converged
## initial  value -0.845732
## iter   2 value -0.850836
## iter   3 value -0.850957
## iter   4 value -0.850985
## iter   5 value -0.850998
## iter   6 value -0.850998
## iter   6 value -0.850998
## final  value -0.850998
## converged
```




```
model_1
```

```
## $fit
##
## Call:
## 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.c
ontrol = list(trace = trc,
##         REPORT = 1, reltol = tol))
##
```

```
## Coefficients:
##          sma1     sma2
##       -0.8784  -0.0789
## s.e.   0.0277   0.0279
##
## sigma^2 estimated as 0.1784:  log likelihood = -755.93,  aic = 1517.86
##
## $degrees_of_freedom
## [1] 1329
##
## $ttable
##      Estimate     SE  t.value p.value
## sma1  -0.8784 0.0277 -31.7474  0.0000
## sma2  -0.0789 0.0279  -2.8242  0.0048
##
## $AIC
## [1] 1.140389
##
## $AICc
## [1] 1.140396
##
## $BIC
## [1] 1.152096
```

Coefficients of the model are statistically significant.

From above plots we can conjecture that residuals are correleted which is undesirable phenomenon. ACF shows autocorrelation at few lags. However, we can also see that this correlation isn't very strong. What's more, for 35 lags in Ljung-Box statistic we receive p-values lower than significance level of 0.05. Therefore, we can reject hypothesis 0 stating that model doesn't exhibit lack of fit.

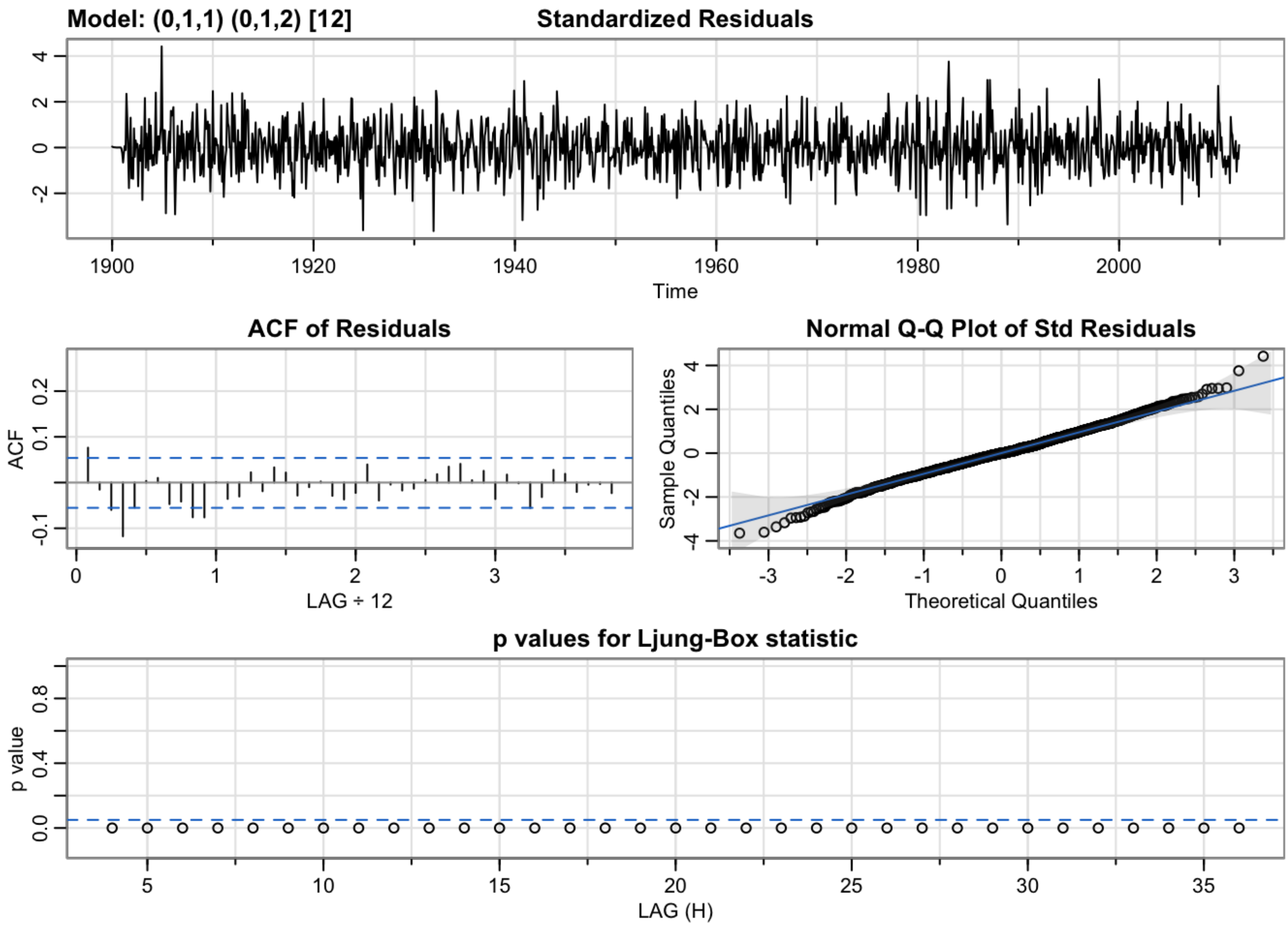
**Model is not suitable.**

# Model 2

```
model_2<- sarima(data.train,0,1,1,0,1,2,12)
```

```
## initial  value -0.572584
## iter   2 value -0.842610
## iter   3 value -0.932451
## iter   4 value -0.947904
## iter   5 value -0.956727
## iter   6 value -0.961830
## iter   7 value -0.963724
## iter   8 value -0.964218
## iter   9 value -0.964251
## iter  10 value -0.964271
## iter  11 value -0.964272
## iter  12 value -0.964272
## iter  12 value -0.964272
## iter  12 value -0.964272
## final  value -0.964272
## converged
## initial  value -0.977109
```

```
## iter   2 value -0.982026
## iter   3 value -0.982267
## iter   4 value -0.982481
## iter   5 value -0.982514
## iter   6 value -0.982517
## iter   7 value -0.982517
## iter   7 value -0.982517
## iter   7 value -0.982517
## final  value -0.982517
## converged
```




```
model_2
```

```
## $fit
##
## Call:
## 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.c
ontrol = list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1     sma1     sma2
##       -0.6358  -0.9327  -0.0107
## s.e.   0.0284   0.0289   0.0292
##
## sigma^2 estimated as 0.1373:  log likelihood = -580.88,  aic = 1169.75
```

```
##
## $degrees_of_freedom
## [1] 1328
##
## $ttable
##       Estimate     SE  t.value p.value
## ma1    -0.6358 0.0284 -22.4138  0.0000
## sma1   -0.9327 0.0289 -32.2389  0.0000
## sma2   -0.0107 0.0292  -0.3664  0.7141
##
## $AIC
## [1] 0.8788537
##
## $AICc
## [1] 0.8788673
##
## $BIC
## [1] 0.894462
```

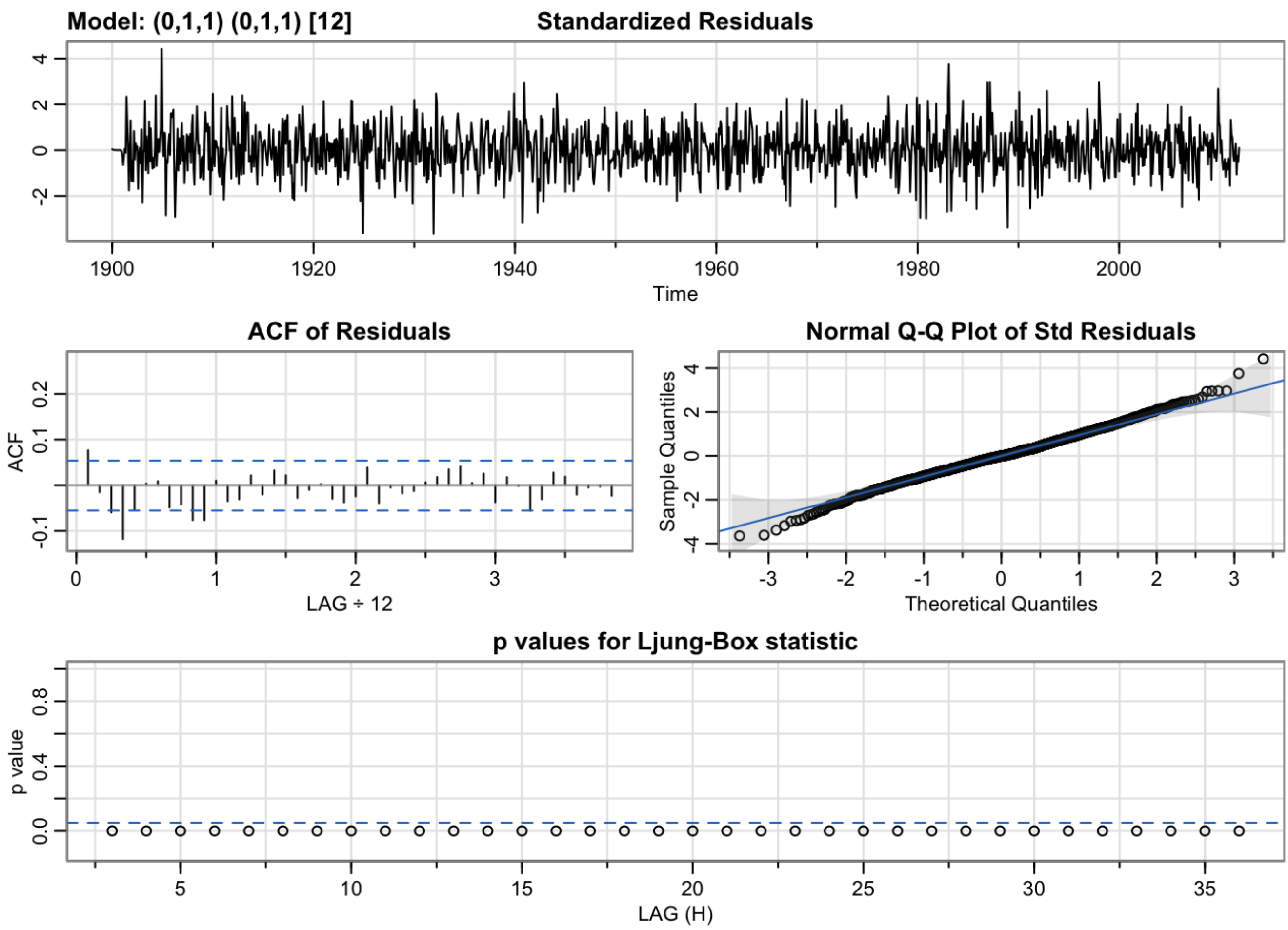
We can observe the same situation as in previous case.

**Model is not suitable.**

# Model 3

```
model_3 <- sarima(data.train, 0,1,1,0,1,1,12)
```

```
## initial  value -0.572584
## iter   2 value -0.836506
## iter   3 value -0.893434
## iter   4 value -0.959318
## iter   5 value -0.962050
## iter   6 value -0.964207
## iter   7 value -0.964226
## iter   8 value -0.964231
## iter   8 value -0.964231
## iter   8 value -0.964231
## final  value -0.964231
## converged
## initial  value -0.977075
## iter   2 value -0.982262
## iter   3 value -0.982397
## iter   4 value -0.982465
## iter   5 value -0.982467
## iter   5 value -0.982467
## iter   5 value -0.982467
## final  value -0.982467
## converged
```




```
model_3
```

```
## $fit
##
## Call:
## 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.c
ontrol = list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1     sma1
##       -0.6380  -0.9426
## s.e.   0.0277   0.0111
##
## sigma^2 estimated as 0.1374:  log likelihood = -580.94,  aic = 1167.89
##
## $degrees_of_freedom
## [1] 1329
##
## $ttable
##      Estimate     SE  t.value p.value
## ma1   -0.6380 0.0277 -22.9956       0
## sma1  -0.9426 0.0111 -85.2076       0
##
## $AIC
## [1] 0.8774517
```

```
##
## $AICc
## [1] 0.8774585
##
## $BIC
## [1] 0.889158
```

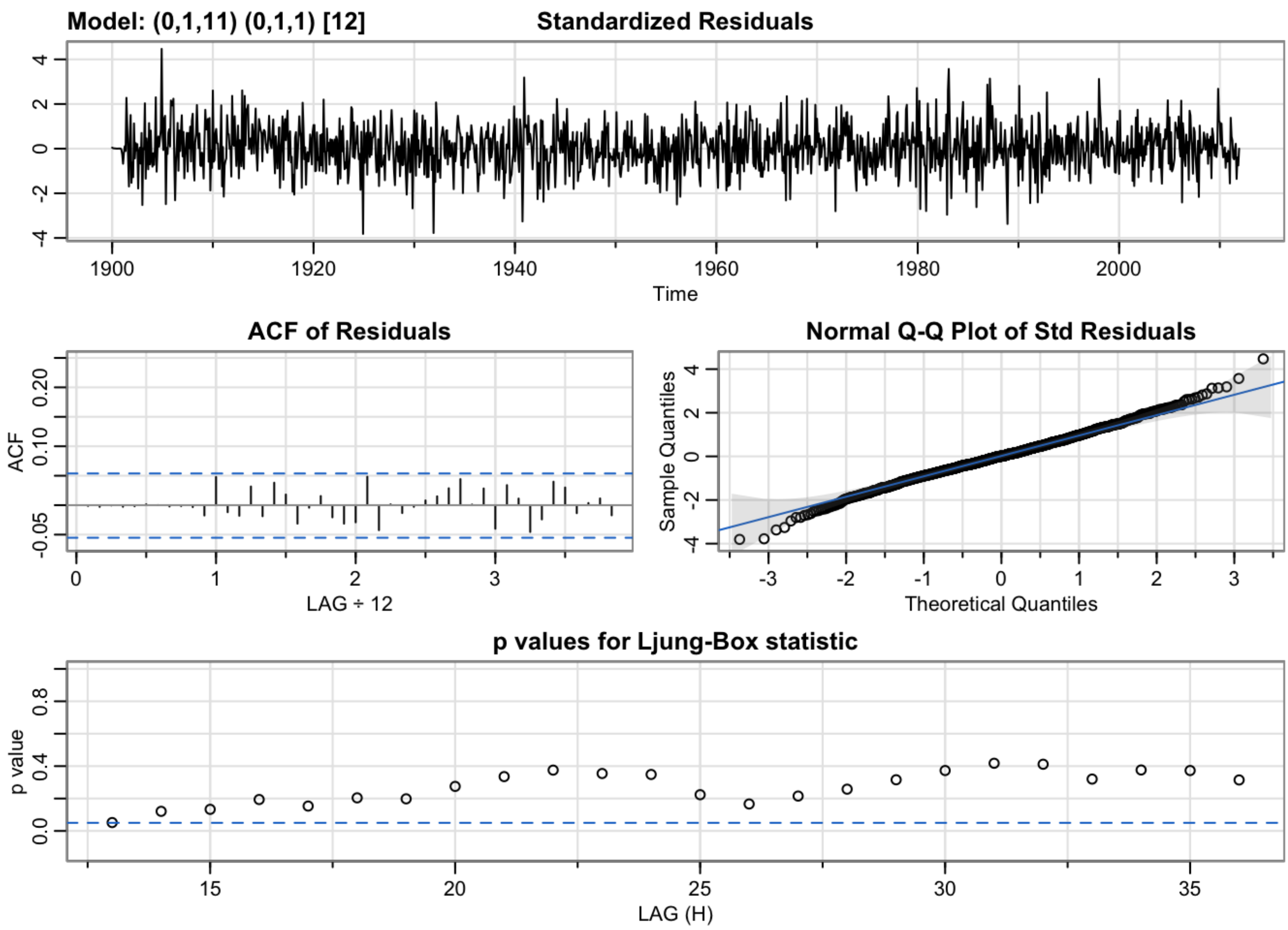
We can observe the same situation as in previous case.

**Model is not suitable.**

# Model 4

```
model_4<-sarima(data.train,0,1,11,0,1,1,12)
```

```
## initial  value -0.572584
## iter   2 value -0.814284
## iter   3 value -0.901694
## iter   4 value -0.924596
## iter   5 value -0.975133
## iter   6 value -0.980909
## iter   7 value -0.983857
## iter   8 value -0.992067
## iter   9 value -0.998418
## iter  10 value -1.000631
## iter  11 value -1.001965
## iter  12 value -1.003945
## iter  13 value -1.004215
## iter  14 value -1.004323
## iter  15 value -1.004346
## iter  16 value -1.004349
## iter  17 value -1.004350
## iter  18 value -1.004350
## iter  18 value -1.004350
## iter  18 value -1.004350
## final  value -1.004350
## converged
## initial  value -1.013118
## iter   2 value -1.015126
## iter   3 value -1.016451
## iter   4 value -1.017046
## iter   5 value -1.017331
## iter   6 value -1.017474
## iter   7 value -1.017508
## iter   8 value -1.017510
## iter   9 value -1.017511
## iter  10 value -1.017511
## iter  10 value -1.017511
## iter  10 value -1.017511
## final  value -1.017511
## converged
```




Since several parameters are not statistically significant we will make them zero and re-estimate the model.

```
model_4_2=Arima(data.train, order = c(0,1,11), seasonal=list(order=c(0,1,1), perio
d=12), fixed=c(NA, NA, NA, NA, 0, 0, 0, NA, 0, 0, 0, NA))
model_4_2
```

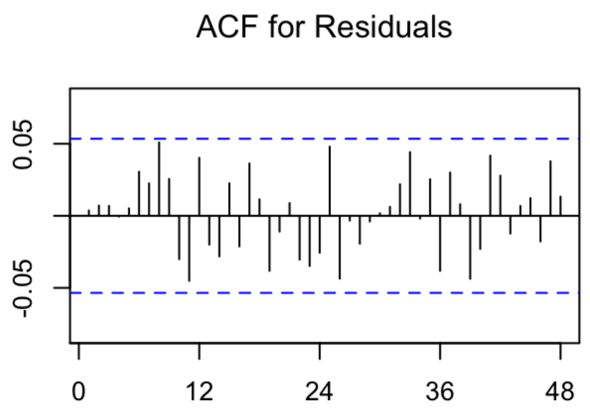
```
## Series: data.train
## ARIMA(0,1,11)(0,1,1)[12]
##
## Coefficients:
##            ma1      ma2      ma3      ma4  ma5  ma6  ma7      ma8  ma9  ma10
##        -0.6176  -0.0832  -0.0601  -0.0902    0    0    0  -0.0966    0     0
## s.e.    0.0273   0.0319   0.0317   0.0298    0    0    0   0.0200    0     0
##        ma11     sma1
##           0  -0.9438
## s.e.      0   0.0120
##
## sigma^2 estimated as 0.1292:  log likelihood=-539.37
## AIC=1092.74   AICc=1092.82   BIC=1129.09
```

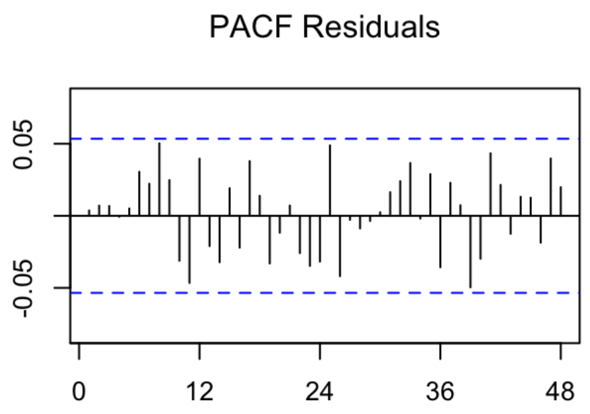
Now we have to recalculate the Ljung-Box statistic.

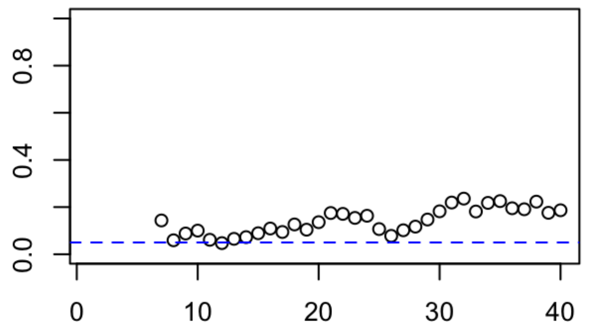
```
myLB= function(x.fit){
  res=NULL
  npar= dim(x.fit$var.coef)[1]
for (i in (npar+1):40){
  q=Box.test(x.fit$residuals,lag=i,type="Ljung-Box",fitdf=npar)
```

```
  res[i]=q$p.value}
  return(res)}

par(mfrow=c(2,2), mar=c(3,3,4,2))
Acf(model_4_2$residuals, type='correlation', lag=48, na.action=na.omit, ylab="", m
ain=expression(paste("ACF for Residuals")))
Acf(model_4_2$residuals, type='partial', lag=48, na.action=na.omit, ylab="", main=
expression(paste("PACF Residuals")))
plot(myLB(model_4_2),ylim=c(0,1))
abline(h=0.05,col="blue",lty=2)
```



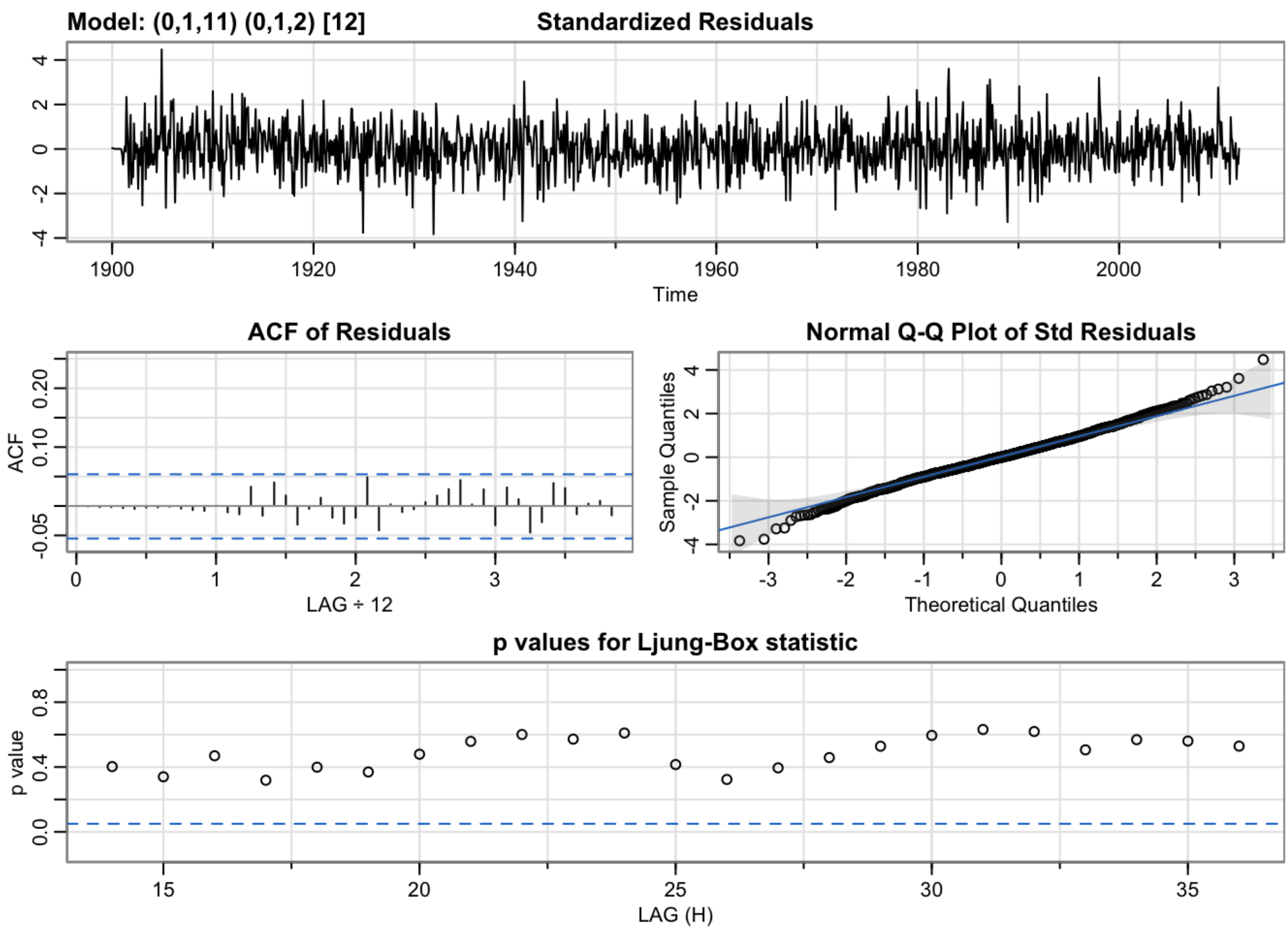

The residuals are mostly uncorrelated with each other and all parameters are statistically signicant. So far this model is the best one.

# Model 5

```
model_5<-sarima(data.train,0,1,11,0,1,2,12)
```

```
## initial  value -0.572584
## iter   2 value -0.819737
## iter   3 value -0.923778
## iter   4 value -0.950910
## iter   5 value -0.968656
## iter   6 value -0.981425
## iter   7 value -0.988148
## iter   8 value -0.995046
## iter   9 value -0.998589
## iter  10 value -1.002163
```

```
## iter  11 value -1.003598
## iter  12 value -1.004655
## iter  13 value -1.005638
## iter  14 value -1.005834
## iter  15 value -1.005895
## iter  16 value -1.005901
## iter  17 value -1.005902
## iter  18 value -1.005902
## iter  19 value -1.005902
## iter  19 value -1.005902
## iter  19 value -1.005902
## final  value -1.005902
## converged
## initial  value -1.014470
## iter   2 value -1.017747
## iter   3 value -1.018014
## iter   4 value -1.018631
## iter   5 value -1.018820
## iter   6 value -1.018833
## iter   7 value -1.018893
## iter   8 value -1.018894
## iter   9 value -1.018896
## iter  10 value -1.018896
## iter  10 value -1.018896
## iter  10 value -1.018896
## final  value -1.018896
## converged
```

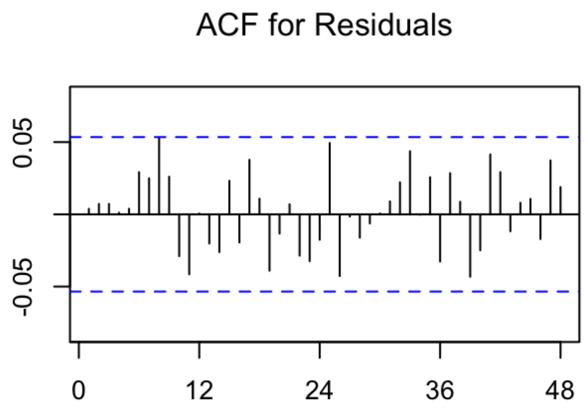


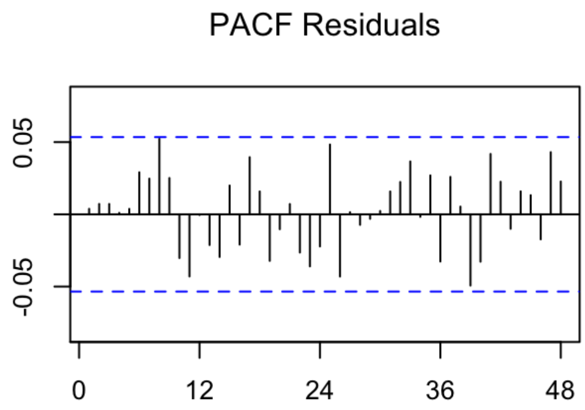

As like in an example above some of the parameters are not statistically significant.

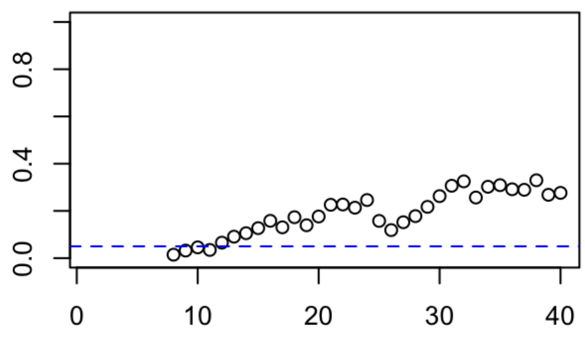
```
model_5_2=Arima(data.train, order = c(0,1,11), seasonal=list(order=c(0,1,2), perio
d=12), fixed=c(NA, NA, NA, NA, 0, 0, 0, NA, 0, 0, 0, NA, NA))
model_5_2
```

```
## Series: data.train
## ARIMA(0,1,11)(0,1,2)[12]
##
## Coefficients:
##            ma1      ma2      ma3      ma4  ma5  ma6  ma7      ma8  ma9  ma10
##        -0.6139  -0.0841  -0.0642  -0.0919    0    0    0  -0.0972    0     0
## s.e.    0.0274   0.0319   0.0318   0.0299    0    0    0   0.0197    0     0
##        ma11     sma1     sma2
##           0  -0.9014  -0.0463
## s.e.      0   0.0289   0.0290
##
## sigma^2 estimated as 0.129:  log likelihood=-538.11
## AIC=1092.21   AICc=1092.32   BIC=1133.76
```

```
par(mfrow=c(2,2), mar=c(3,3,4,2))
Acf(model_5_2$residuals, type='correlation', lag=48, na.action=na.omit, ylab="", m
ain=expression(paste("ACF for Residuals")))
Acf(model_5_2$residuals, type='partial', lag=48, na.action=na.omit, ylab="", main=
expression(paste("PACF Residuals")))
plot(myLB(model_5_2),ylim=c(0,1))
abline(h=0.05,col="blue",lty=2)
```



The residuals remain mostly uncorrelated and all the parameters are statistically significant. This is also a good model.

### Main conclusions

From five estimated models **we choose** two of them - **MODEL 4** and **MODEL 5**.

Those two models have mostly uncorrelated residuals and statistically significant parameters.

# 4. FORECASTING

### Information criteria

Using AIC, AICc and BIC criterions we can compare both models.

The smallest the value the better model.

Results:

| Model | Order | AIC | AICc | BIC |
|---|---|---|---|---|
| MODEL 4 | $(0,1,11) \times (0,1,1)_{12}$ | 1092.74 | 1092.82 | 1129.09 |
| MODEL 5 | $(0,1,11) \times (0,1,2)_{12}$ | 1092.21 | 1092.32 | 1133.76 |

MODEL 4 is slightly worse in AICc and AIC criterion.

MODEL 5 is worse in BIC criterion.

Models ended up really similar so we will test forecasting for both of them.

We forecast first 2 years (24months) using both models.

```
model_4_2.f.h <- forecast::forecast(model_4_2, h=24)
model_5_2.f.h <- forecast::forecast(model_5_2, h=24)
```

## Forecast accuracy evaluation.

Testing accuracy of forecasts can be made by using several measures. This time we will base our results on four of them: RMSE, MAE, MAPE, MASE.

RMSE (Root Mean Square Error) the closer to 0 the better.

MAE (Mean Absolute Error) the closer to 0 the better.

MAPE (Mean Absolute Percentage Error) the lower the better.

MASE (Mean Absolute Scaled Error) the lower the better.

Accuracy evaluation of MODEL 4.

```
model_4_2.f.h.acc=accuracy(model_4_2.f.h, data.test,d=1,D=1)
```

```
model_4_2.f.h.acc
```

```
##                        ME      RMSE       MAE         MPE     MAPE      MASE
## Training set  0.009762457 0.3569449 0.2768275  0.02066173 1.034494 0.6215900
## Test set     -0.288904472 0.3788153 0.3250461 -1.06834155 1.200472 0.7298603
##                      ACF1 Theil's U
## Training set  0.003717534        NA
## Test set     -0.358226954 0.5797837
```

Accuracy evaluation of MODEL 5.

```
model_5_2.f.h.acc=accuracy(model_5_2.f.h, data.test,d=1,D=1)

model_5_2.f.h.acc
```

```
##                        ME      RMSE       MAE         MPE     MAPE      MASE
## Training set  0.009111282 0.3565458 0.2766698  0.01819309 1.033926 0.6212359
## Test set     -0.275502460 0.3666839 0.3150646 -1.01889621 1.163478 0.7074479
##                      ACF1 Theil's U
## Training set  0.003818115        NA
## Test set     -0.342488491 0.5667416
```

## Conclusions

Training set:

| Model | RMSE | MAE | MAPE | MASE |
|---|---|---|---|---|
| MODEL 4 | 0.3569 | 0.2768 | 1.0344 | 0.6215 |
| MODEL 5 | 0.3565 | 0.2766 | 1.0339 | 0.6212 |

RMSE is slightly better in MODEL 5.

MAE is slightly better in MODEL 5.

MAPE is slightly better in MODEL 5.

MASE is slightly better in MODEL 5.

Test set:

| Model | RMSE | MAE | MAPE | MASE |
|---|---|---|---|---|
| MODEL 4 | 0.3788 | 0.3250 | 1.2004 | 0.7298 |
| MODEL 5 | 0.3666 | 0.3150 | 1.1634 | 0.7074 |

RMSE is slightly better in MODEL 5.

MAE is slightly better in MODEL 5.

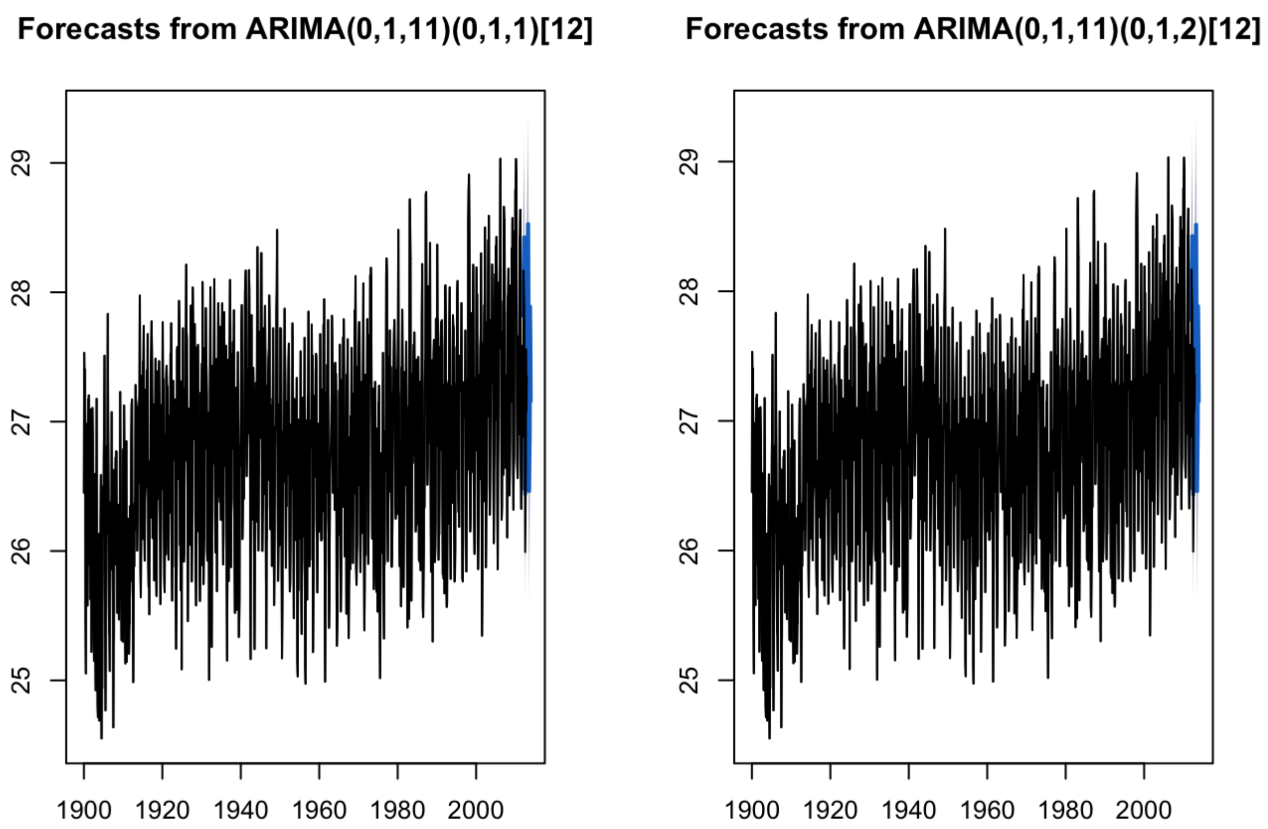
MAPE is slightly better in MODEL 5.

MASE is slightly better in MODEL 5.

In the training and test set, models perform similarly. There is no big differences between them. MODEL 5 seems to present a little better performance.

**Plots with forecasted data.**

```
par(mfrow=c(1,2), cex=0.8)
plot(model_4_2.f.h, xlim=c(1900,2013))
lines(data)
plot(model_5_2.f.h, xlim=c(1900,2013))
lines(data)
```

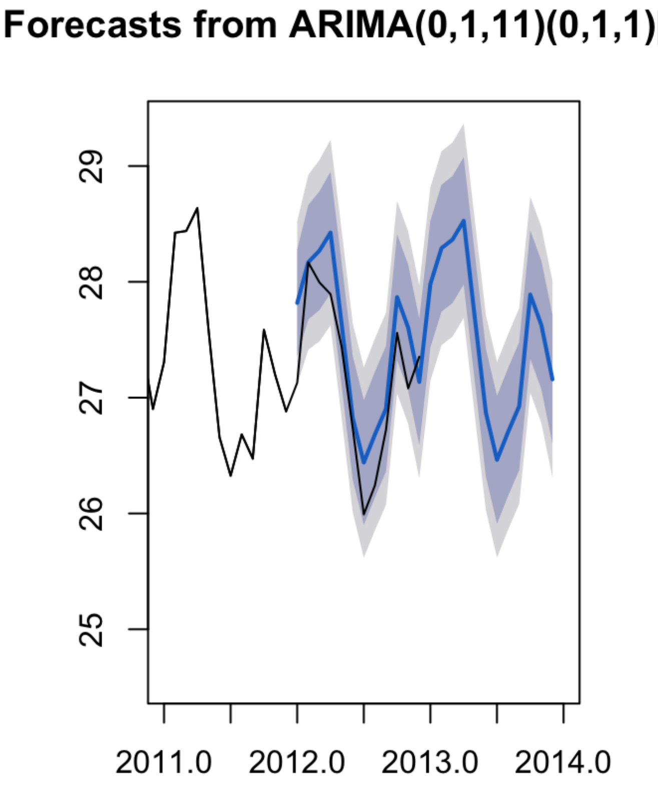


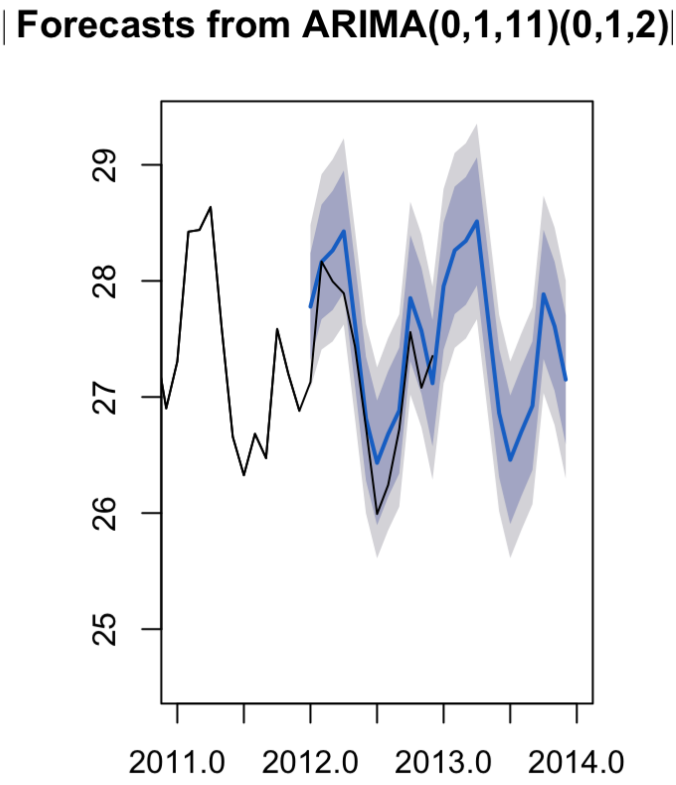

Closer look on forecasted data.

Both models present similar forecasts.

They doesn’t perfectly cover data for year 2012 (but its really close), anyway they seems to give decent forecasts.

```
par(mfrow=c(1,2))
plot(model_4_2.f.h, xlim=c(2011,2014))
lines(data)
plot(model_5_2.f.h, xlim=c(2011,2014))
lines(data)
```




# Comparing forecast with actual numbers

Model 4

```
predicted.values <- c(27.81774,28.16934,28.26876,28.42515,27.10722,26.29314,25.90272, 26.14023, 26.36268,27.32520,27.06099,26.59249)
months <- c('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec')
table <-data.frame(months)
table[2]<- data.test
table[3]<- predicted.values
table[4]<- round(abs(data.test - predicted.values)/data.test,4)
colnames(table)[1] = "Months"
colnames(table)[2] = "Observed value"
colnames(table)[3] = "Forecast"
colnames(table)[4] = "Error"
table
```

```
##   Months Observed value Forecast  Error
## 1    Jan         27.129 27.81774 0.0254
## 2    Feb         28.167 28.16934 0.0001
## 3    Mar         27.995 28.26876 0.0098
## 4    Apr         27.894 28.42515 0.0190
## 5    May         27.440 27.10722 0.0121
## 6    Jun         26.737 26.29314 0.0166
## 7    Jul         25.992 25.90272 0.0034
## 8    Aug         26.242 26.14023 0.0039
## 9    Sep         26.722 26.36268 0.0134
```

```
## 10     Oct          27.559 27.32520 0.0085
## 11     Nov          27.080 27.06099 0.0007
## 12     Dec          27.353 26.59249 0.0278
```

When we compare predicted value for 2012 with the actual ones the error varies from 0 to 0,028 and we can't observe any pattern in error as the number of steps ahead grows.. Generally, errors are low so the predictions are accurate.

Model 5

```
predicted.values <- c(27.77817,28.16395,28.26290,28.42638,27.63339,26.81359,26.43275,26.68086,26.88366,27.85191,27.56979,27.11868)
table <-data.frame(months)
table[2]<- data.test
table[3]<- predicted.values
table[4]<- round(abs(data.test - predicted.values)/data.test,3)
colnames(table)[1] = "Months"
colnames(table)[2] = "Observed value"
colnames(table)[3] = "Forecast"
colnames(table)[4] = "Error"
table
```

```
##    Months Observed value Forecast Error
## 1     Jan         27.129 27.77817 0.024
## 2     Feb         28.167 28.16395 0.000
## 3     Mar         27.995 28.26290 0.010
## 4     Apr         27.894 28.42638 0.019
## 5     May         27.440 27.63339 0.007
## 6     Jun         26.737 26.81359 0.003
## 7     Jul         25.992 26.43275 0.017
## 8     Aug         26.242 26.68086 0.017
## 9     Sep         26.722 26.88366 0.006
## 10    Oct         27.559 27.85191 0.011
## 11    Nov         27.080 27.56979 0.018
## 12    Dec         27.353 27.11868 0.009
```

As in the previous case we can't observe any pattern in error as the number of steps ahead grows. Also in this case the error in each month is low so predictions are accurate.

# 95 % prediction intervals for each model

Model 4

```
lower <- model_4_2.f.h$lower[13:24,2]
upper <- model_4_2.f.h$upper[13:24,2]
inter <- data.frame(months,lower, upper)
colnames(inter)[1] = "Months"
colnames(inter)[2] = "Low"
colnames(inter)[3] = "High"
inter
```

```
##    Months      Low     High
## 1     Jan 27.14632 28.81622
## 2     Feb 27.45374 29.12688
```

```
## 3      Mar 27.52777 29.20373
## 4      Apr 27.68783 29.36635
## 5      May 26.85094 28.53163
## 6      Jun 26.02879 27.71164
## 7      Jul 25.62049 27.30551
## 8      Aug 25.86015 27.54734
## 9      Sep 26.08225 27.77123
## 10     Oct 27.04441 28.73518
## 11     Nov 26.77984 28.47239
## 12     Dec 26.31098 28.00532
```

### Model 5

```
lower <- model_5_2.f.h$lower[13:24,2]
upper <- model_5_2.f.h$upper[13:24,2]
months <- c('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct',
'Nov', 'Dec')
inter <- data.frame(months,lower, upper)
colnames(inter)[1] = "Months"
colnames(inter)[2] = "Low"
colnames(inter)[3] = "High"
print("95 % Confidence interval for 2013 year for model 4")
```

```
## [1] "95 % Confidence interval for 2013 year for model 4"
```

```
inter
```

```
##     Months      Low     High
## 1      Jan 27.11873 28.79279
## 2      Feb 27.42419 29.10271
## 3      Mar 27.50380 29.18596
## 4      Apr 27.67012 29.35534
## 5      May 26.84160 28.52917
## 6      Jun 26.01657 27.70648
## 7      Jul 25.61268 27.30493
## 8      Aug 25.85288 27.54746
## 9      Sep 26.07619 27.77245
## 10     Oct 27.03575 28.73368
## 11     Nov 26.75815 28.45776
## 12     Dec 26.29925 28.00054
```

# Conclusions

From the above analysis we can see that we obtained model with high accuracy score which can be reliable predictor for temperature in Lagos. From our forecasting it can be concluded that the temperature in Lagos is significantly increasing during the previous years and have the same tendency for the future years. Since Lagos is in intertropical zone, which is the most vulnerable part of the earth for greenhouse effect, it can be used as a general measure for rising temperature on our planet. This forecast is an evidence that the greenhouse effect exists and can be used by organisations connected with combating the global warming. The rising tendency of temperature is alarming because if this trend maintains the consequences will be damaging for all planet earth and mankind. By showing this forecast we can convince and raise awareness both among people and companies to implement activities that could reduce their contribution to such effects. It can be achieved by using more renewable energy, using public communication instead of cars and preventing deforestation.