Global Temperture Analysis

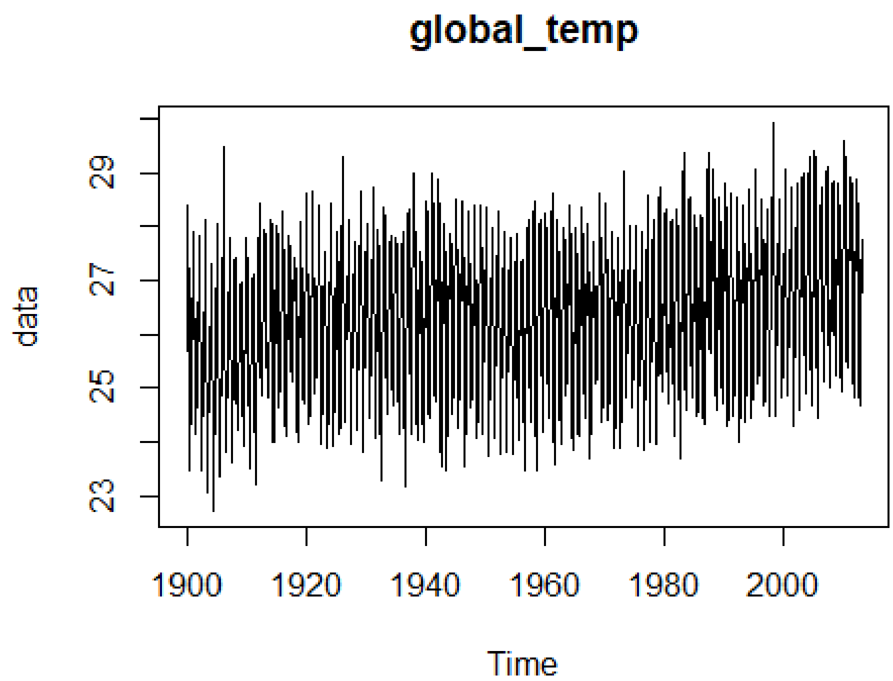
SHEMONA SWAIN 1948056

10/10/2020

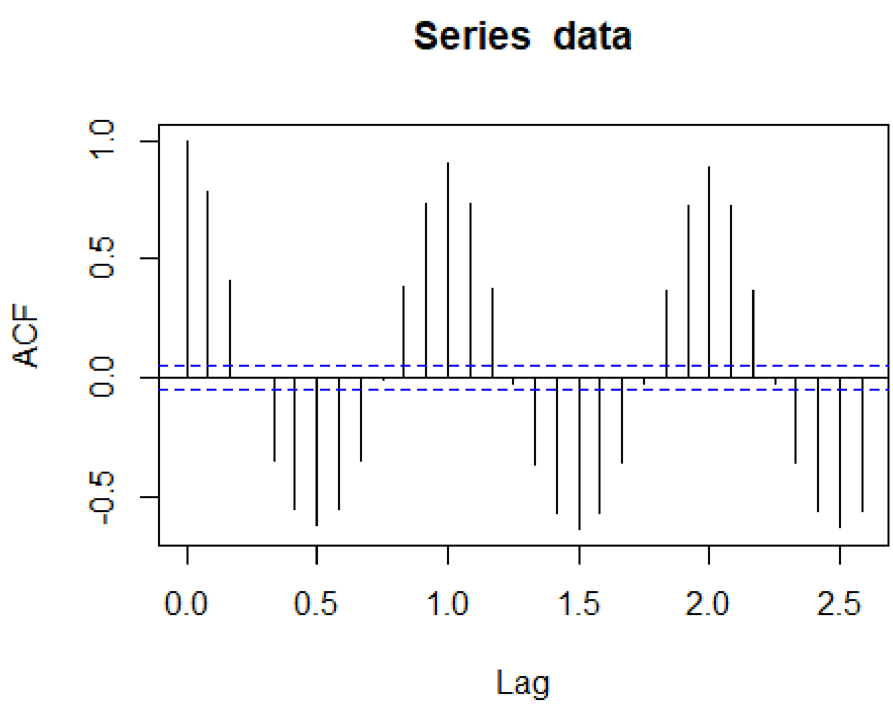
```r
# Installing all necessary libraries
library(tseries)

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

library(forecast)

## Warning: package 'forecast' was built under R version 4.0.2

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.2

## -- Attaching packages --------------------------------------------- tidyve
rse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## Warning: package 'ggplot2' was built under R version 4.0.2

## Warning: package 'tibble' was built under R version 4.0.2

## Warning: package 'tidyr' was built under R version 4.0.2

## Warning: package 'readr' was built under R version 4.0.2

## Warning: package 'dplyr' was built under R version 4.0.2

## Warning: package 'forcats' was built under R version 4.0.2

## -- Conflicts ------------------------------------------------ tidyverse_co
nflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(tidymodels)

## Warning: package 'tidymodels' was built under R version 4.0.2
```

```
## -- Attaching packages --------------------------------------------- tidymod
els 0.1.1 --

## v broom     0.7.1      v recipes   0.1.13
## v dials     0.0.9      v rsample   0.0.8
## v infer     0.5.3      v tune      0.1.1
## v modeldata 0.0.2      v workflows 0.2.1
## v parsnip   0.1.3      v yardstick 0.0.7

## Warning: package 'broom' was built under R version 4.0.2

## Warning: package 'dials' was built under R version 4.0.2

## Warning: package 'infer' was built under R version 4.0.2

## Warning: package 'modeldata' was built under R version 4.0.2

## Warning: package 'parsnip' was built under R version 4.0.2

## Warning: package 'recipes' was built under R version 4.0.2

## Warning: package 'rsample' was built under R version 4.0.2

## Warning: package 'tune' was built under R version 4.0.2

## Warning: package 'yardstick' was built under R version 4.0.2

## -- Conflicts ---------------------------------------------- tidymodels_co
nflicts() --
## x yardstick::accuracy() masks forecast::accuracy()
## x scales::discard()     masks purrr::discard()
## x dplyr::filter()       masks stats::filter()
## x recipes::fixed()      masks stringr::fixed()
## x dplyr::lag()          masks stats::lag()
## x yardstick::spec()     masks readr::spec()
## x recipes::step()       masks stats::step()

library(modeltime)
library(timetk)
library(lubridate)

## Warning: package 'lubridate' was built under R version 4.0.2

##
## Attaching package: 'lubridate'

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

library(lmtest)

## Warning: package 'lmtest' was built under R version 4.0.2
```

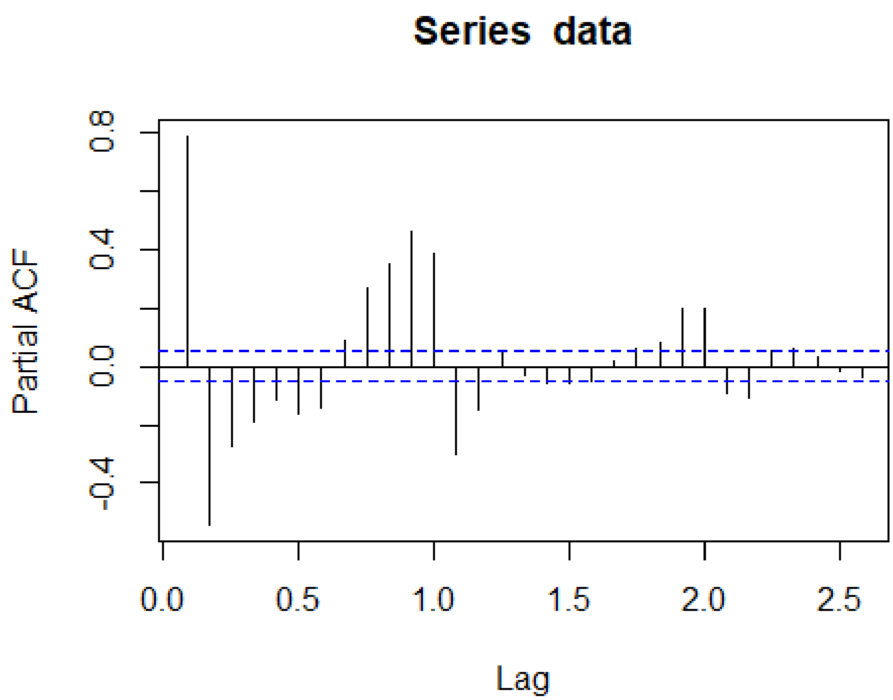
```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

library(FitAR)

## Warning: package 'FitAR' was built under R version 4.0.2

## Loading required package: lattice

## Loading required package: leaps

## Warning: package 'leaps' was built under R version 4.0.2

## Loading required package: ltsa

## Loading required package: bestglm

## Warning: package 'bestglm' was built under R version 4.0.2

##
## Attaching package: 'FitAR'

## The following object is masked from 'package:forecast':
##
##     BoxCox

library(TSstudio)

## Warning: package 'TSstudio' was built under R version 4.0.2




#Plotting the Time Series Data
global_temp <- read.csv("C:/Users/Lenovo/Dropbox/My PC (LAPTOP-T1F1GG8F)/Desk
top/global_temp.csv")
data<-ts(global_temp,start=c(1900,1),end=c(2013,1),frequency=12)
ts.plot(data,main="global_temp")
```




```
#Checking Auto-Correlation Function
#acf at diff lags
acf(data)
```



```r
#Plotting Partial auto correlation function
pacf(data)
```



Interpretation: So, from the above PACF plot we can say that it "cuts off" after the lag 2, while the ACF plot "tails off" to zero. So, we can say that it probably has something like an AR(2).

```r
adf.test(data)

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

##
##  Augmented Dickey-Fuller Test
##
## data:  data
## Dickey-Fuller = -4.2548, Lag order = 11, p-value = 0.01
## alternative hypothesis: stationary
```

Interpretation: So, when we observe the p- value we find that p-value < 0.05 i.e. 0.01 < 0.05. Hence, we reject the null hypothesis.So, we can say that the data is Stationary.

```r
#Fitting the best Model
fit_data=auto.arima(data,trace = TRUE)

##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2)(1,1,1)[12] with drift         : Inf
```

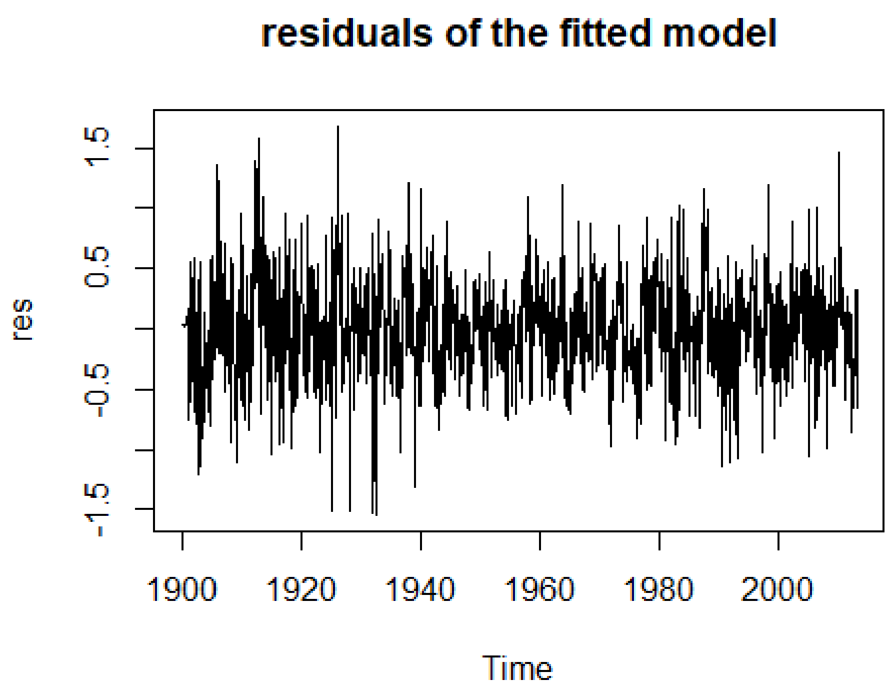
```
##  ARIMA(0,0,0)(0,1,0)[12] with drift         : 2336.053
##  ARIMA(1,0,0)(1,1,0)[12] with drift         : 1611.588
##  ARIMA(0,0,1)(0,1,1)[12] with drift         : 1378.817
##  ARIMA(0,0,0)(0,1,0)[12]                    : 2334.199
##  ARIMA(0,0,1)(0,1,0)[12] with drift         : 2052.972
##  ARIMA(0,0,1)(1,1,1)[12] with drift         : 1372.892
##  ARIMA(0,0,1)(1,1,0)[12] with drift         : 1753.133
##  ARIMA(0,0,1)(2,1,1)[12] with drift         : 1351.601
##  ARIMA(0,0,1)(2,1,0)[12] with drift         : 1572.136
##  ARIMA(0,0,1)(2,1,2)[12] with drift         : 1326.822
##  ARIMA(0,0,1)(1,1,2)[12] with drift         : 1361.307
##  ARIMA(0,0,0)(2,1,2)[12] with drift         : 1591.361
##  ARIMA(1,0,1)(2,1,2)[12] with drift         : Inf
##  ARIMA(0,0,2)(2,1,2)[12] with drift         : 1223.016
##  ARIMA(0,0,2)(1,1,2)[12] with drift         : Inf
##  ARIMA(0,0,2)(2,1,1)[12] with drift         : 1247.468
##  ARIMA(0,0,2)(1,1,1)[12] with drift         : 1246.184
##  ARIMA(1,0,2)(2,1,2)[12] with drift         : Inf
##  ARIMA(0,0,3)(2,1,2)[12] with drift         : 1186.21
##  ARIMA(0,0,3)(1,1,2)[12] with drift         : Inf
##  ARIMA(0,0,3)(2,1,1)[12] with drift         : 1217.201
##  ARIMA(0,0,3)(1,1,1)[12] with drift         : 1210.628
##  ARIMA(1,0,3)(2,1,2)[12] with drift         : Inf
##  ARIMA(0,0,4)(2,1,2)[12] with drift         : 1177.58
##  ARIMA(0,0,4)(1,1,2)[12] with drift         : Inf
##  ARIMA(0,0,4)(2,1,1)[12] with drift         : 1207.773
##  ARIMA(0,0,4)(1,1,1)[12] with drift         : 1203.665
##  ARIMA(1,0,4)(2,1,2)[12] with drift         : Inf
##  ARIMA(0,0,5)(2,1,2)[12] with drift         : 1175.239
##  ARIMA(0,0,5)(1,1,2)[12] with drift         : Inf
##  ARIMA(0,0,5)(2,1,1)[12] with drift         : 1206.21
##  ARIMA(0,0,5)(1,1,1)[12] with drift         : 1202.377
##  ARIMA(1,0,5)(2,1,2)[12] with drift         : Inf
##  ARIMA(0,0,5)(2,1,2)[12]                    : 1190.702
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(0,0,5)(2,1,2)[12] with drift         : Inf
##  ARIMA(0,0,4)(2,1,2)[12] with drift         : Inf
##  ARIMA(0,0,3)(2,1,2)[12] with drift         : Inf
##  ARIMA(0,0,5)(2,1,2)[12]                    : Inf
##  ARIMA(0,0,5)(1,1,1)[12] with drift         : Inf
##  ARIMA(0,0,4)(1,1,1)[12] with drift         : Inf
##  ARIMA(0,0,5)(2,1,1)[12] with drift         : Inf
##  ARIMA(0,0,4)(2,1,1)[12] with drift         : Inf
##  ARIMA(0,0,3)(1,1,1)[12] with drift         : Inf
##  ARIMA(0,0,3)(2,1,1)[12] with drift         : Inf
##  ARIMA(0,0,2)(2,1,2)[12] with drift         : Inf
##  ARIMA(0,0,2)(1,1,1)[12] with drift         : Inf
##  ARIMA(0,0,2)(2,1,1)[12] with drift         : Inf
```

```
##  ARIMA(0,0,1)(2,1,2)[12] with drift         : Inf
##  ARIMA(0,0,1)(2,1,1)[12] with drift         : Inf
##  ARIMA(0,0,1)(1,1,2)[12] with drift         : Inf
##  ARIMA(0,0,1)(1,1,1)[12] with drift         : Inf
##  ARIMA(0,0,1)(0,1,1)[12] with drift         : Inf
##  ARIMA(0,0,1)(2,1,0)[12] with drift         : 1578.865
## 
##  Best model: ARIMA(0,0,1)(2,1,0)[12] with drift
```

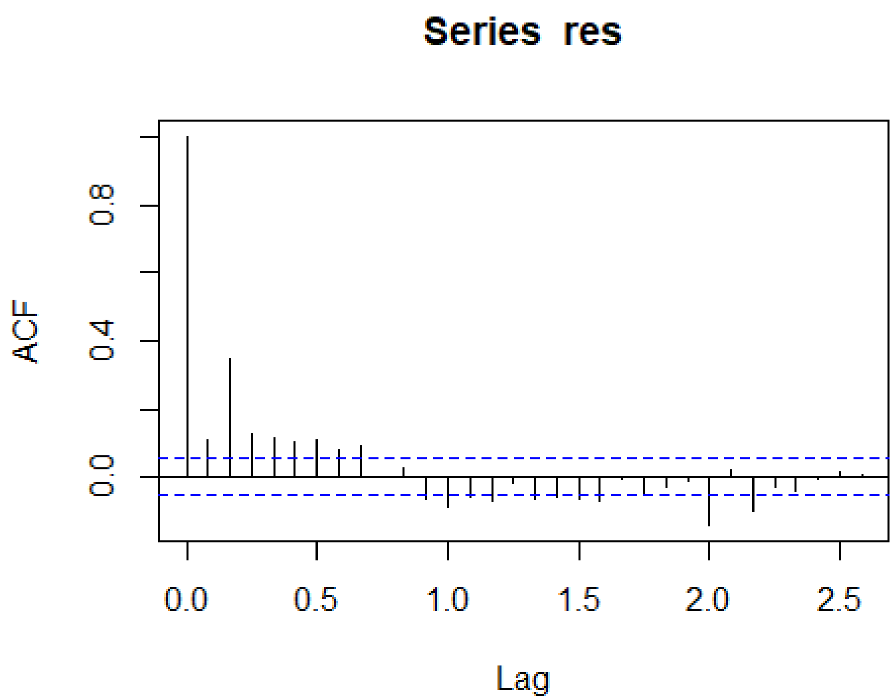
Interpretation:

1. As, auto.arima gives the best model that can be fitted to a given data in the form ARIMA(p,d,q) 2. By seeing the output we can say that d part is 0 because the data is already stationary otherwise d part would have been 1. 3. Also, There is neither trend nor seasonality. 4. So, the obtained model is ARIMA(0,0,1).

```
# Residual Analysis
#Checking whether the residuals of the best fitted model satisfying the assum
ptions
#checking for all 3 assumptions
res=residuals(fit_data)
#Plotting the Residuals
plot(res,main="residuals of the fitted model")
```




```
#Assumption_1:residuals are uncorrelated  Random Variable
#Finding acf of the residual series,this gives the dependency
acf(res)
```

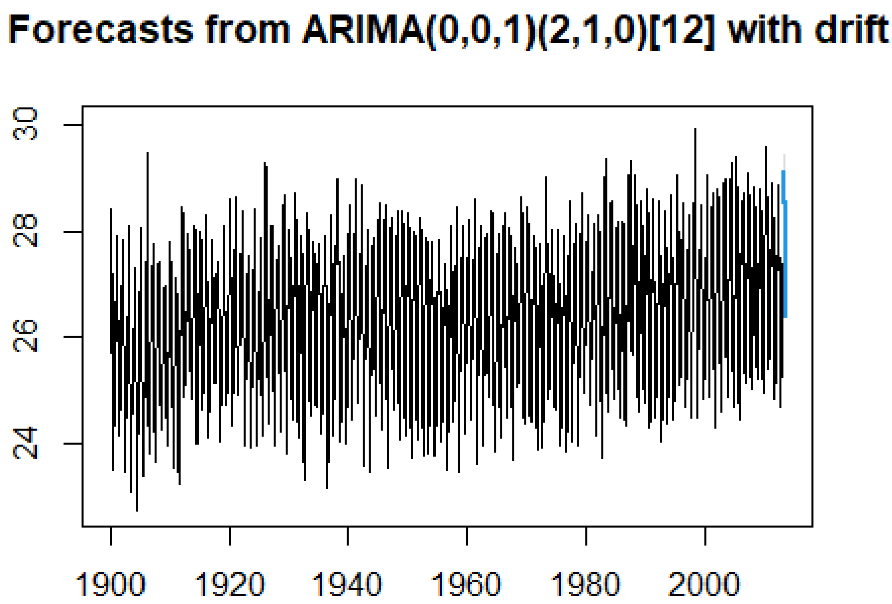


Interpretation:
Clearly all the values of acf are lying within the blue dotted line.i.e they are negligible.i.e there is independency.

```
#Assumption_2 :zero mean and const variance
#the avg of the obs are close to zero from the plot
#Assumption_3:normality of the plot
shapiro.test(res)

##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.99502, p-value = 0.0001819
```

Interpretation: 1. When we observe the p value we can say that that p-value 0.0001 <0.05.So, we reject the the null hypothesis. Hence, the distribution of the given data is different from normal distribution significantly.

```
#OUT-SAMPLE FORECAST
#Making predictions
#make the prediction for next 5 more observations based on ARIMA(0,0,1)
newfit=forecast(fit_data,h=5)
plot(newfit)
```

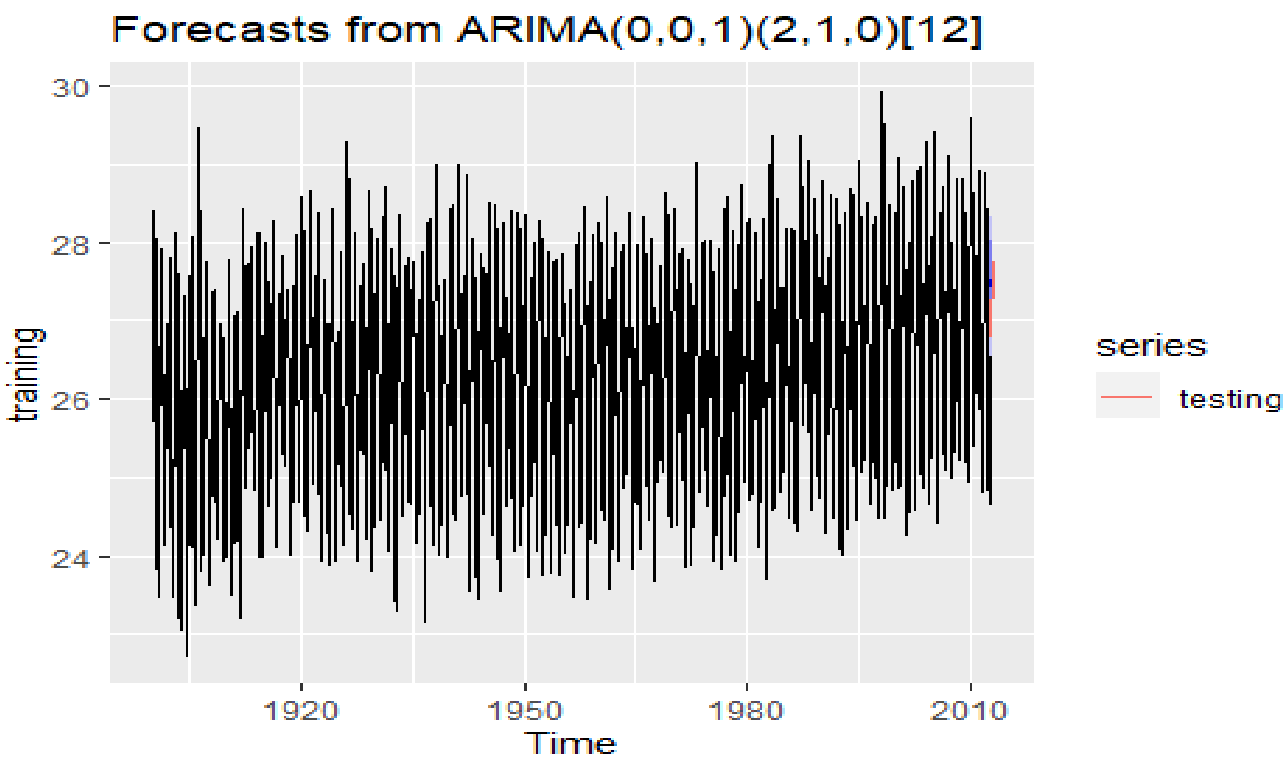


Interpretation:

1. h=5 means 5 step ahead we are predicting so the blue colour shows the forecast for 5 step ahead.

```
#IN-SAMPLE FORECAST
split_data <- ts_split(ts.obj = data, sample.out = 2)

training <- split_data$train
testing <- split_data$test

print("Length of original data:")

## [1] "Length of original data:"

length(data)

## [1] 1357

print("Length of Training data:")

## [1] "Length of Training data:"

length(training)

## [1] 1355

print("Length of Testing data:")
```

```
## [1] "Length of Testing data:"

length(testing)

## [1] 2

data.train <- Arima(training, order=c(0,0,1),
                    seasonal=c(2,1,0), lambda=0)
data.train %>%
  forecast(h=2) %>%
  autoplot() + autolayer(testing)
```



Interpretation: As we can see that the testing (red line) is overlapping the predicted forecast (blue line). So, we can say that it has forecasted the last 2 values accurately.