TSA_Assignment_2038

Rasika Bose, 2038

3rd Jan 2022

Data Description

The data set that we have used here is the daily mean temperature of Delhi,India from 1st January 2013 to 24th April 2017. We have collected the data from Kaggle. The total data was split into train and test files separately where the train data consists of data from 1st Jan 2013 to 31st December 2016 and the test data consists of data from 1st Jan 2017 to 24th April 2017. From the data we have observed there are 4 features namely mean temperature, humidity, wind speed and mean pressure. We have done uni-variate time series analysis on mean temperature.

Data link

https://www.kaggle.com/sumanthvrao/daily-climate-time-series-data

Loading required libraries

```
library(readr)
library(tseries)
library(forecast)
```

Loading the data

```
train = read_csv('DailyDelhiClimateTrain.csv')
test = read_csv('DailyDelhiClimateTest.csv')
train = train[,1:2]
test = test[,1:2]
```

```
str(train)
```

```
## tibble [1,462 x 2] (S3: tbl_df/tbl/data.frame)
##  $ date    : Date[1:1462], format: "2013-01-01" "2013-01-02" ...
##  $ meantemp: num [1:1462] 10 7.4 7.17 8.67 6 ...
```

The training data consists of a total of 1462 observations of date and mean temperature.

```
str(test)
```

```
## tibble [114 x 2] (S3: tbl_df/tbl/data.frame)
##  $ date    : Date[1:114], format: "2017-01-01" "2017-01-02" ...
##  $ meantemp: num [1:114] 15.9 18.5 17.1 18.7 18.4 ...
```

The test data consists of a total of 114 observations of date and mean temperature.

Summary

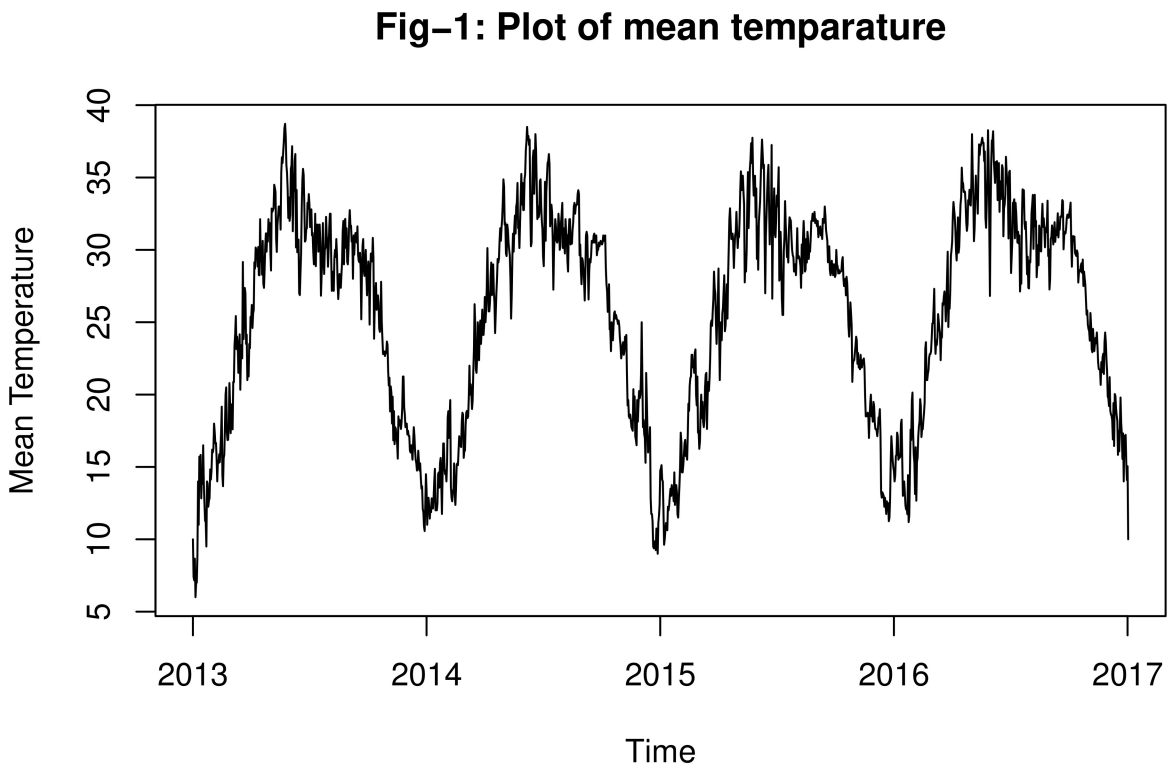
```
summary(train$meantemp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    6.00   18.86   27.71   25.50   31.31   38.71
```

Exploratory Data Analysis

Plot of the data

```
t = ts(train$meantemp,start=c(2013,1),deltat=1/365)
plot.ts(t,ylab = 'Mean Temperature',main='Fig-1: Plot of mean temparature')
```

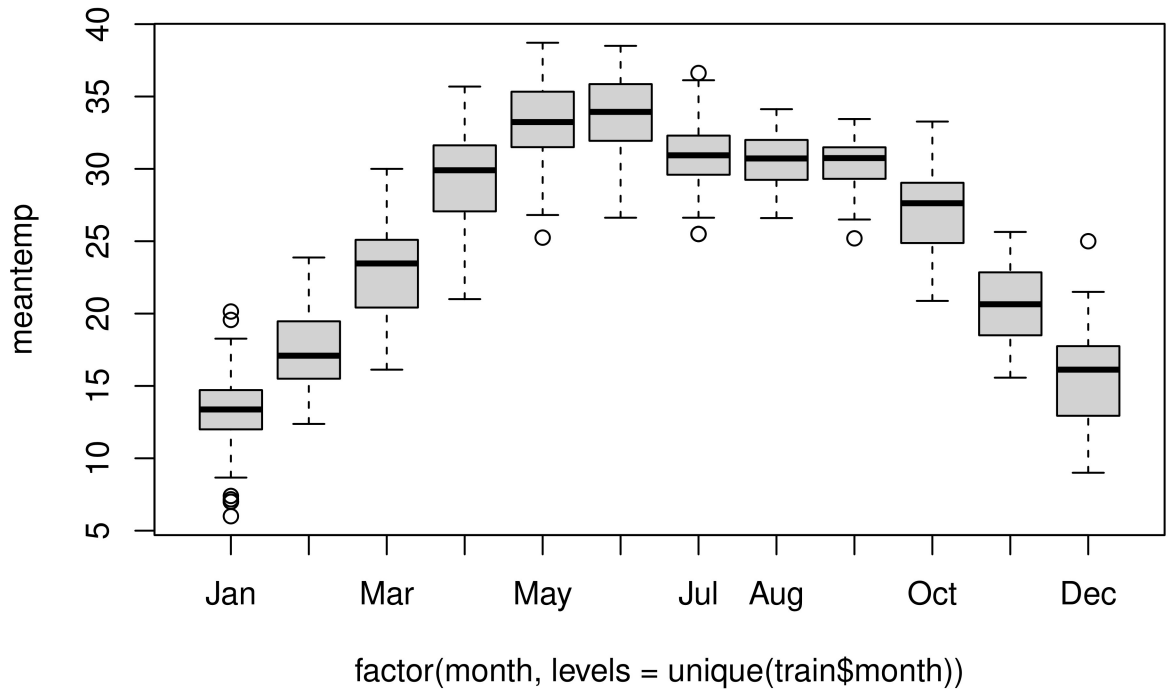


From the plot we can visually infer that seasonality is present in the data with no visual evidence of trend.

Box plot

```
train['month'] = format(train$date,"%b")
train['Year'] = format(train$date,"%Y")
boxplot(meantemp~factor(month,levels=unique(train$month)),data = train,main='Fig-2: Box plot')
```

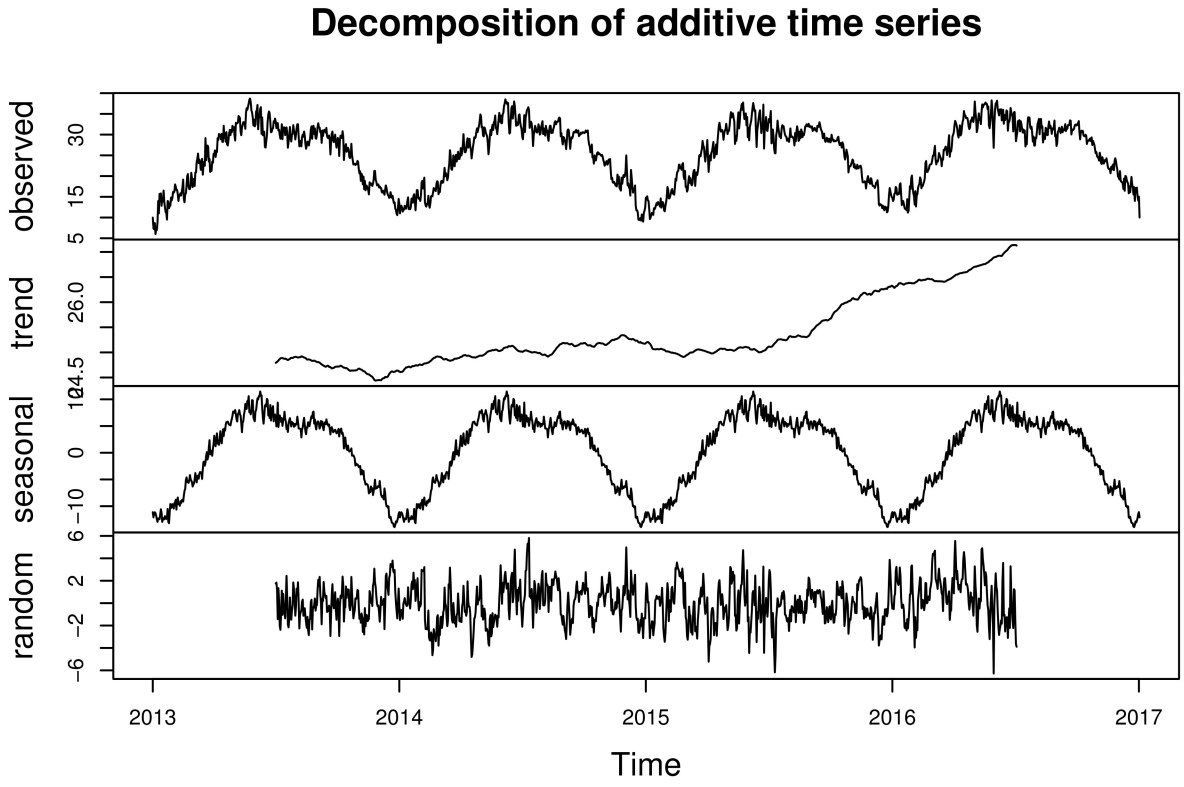
Fig–2: Box plot



From the month-wise box plot we can observe the behavior of the mean temperature distribution of every month for four years 2013-2016.

Decomposition of time series

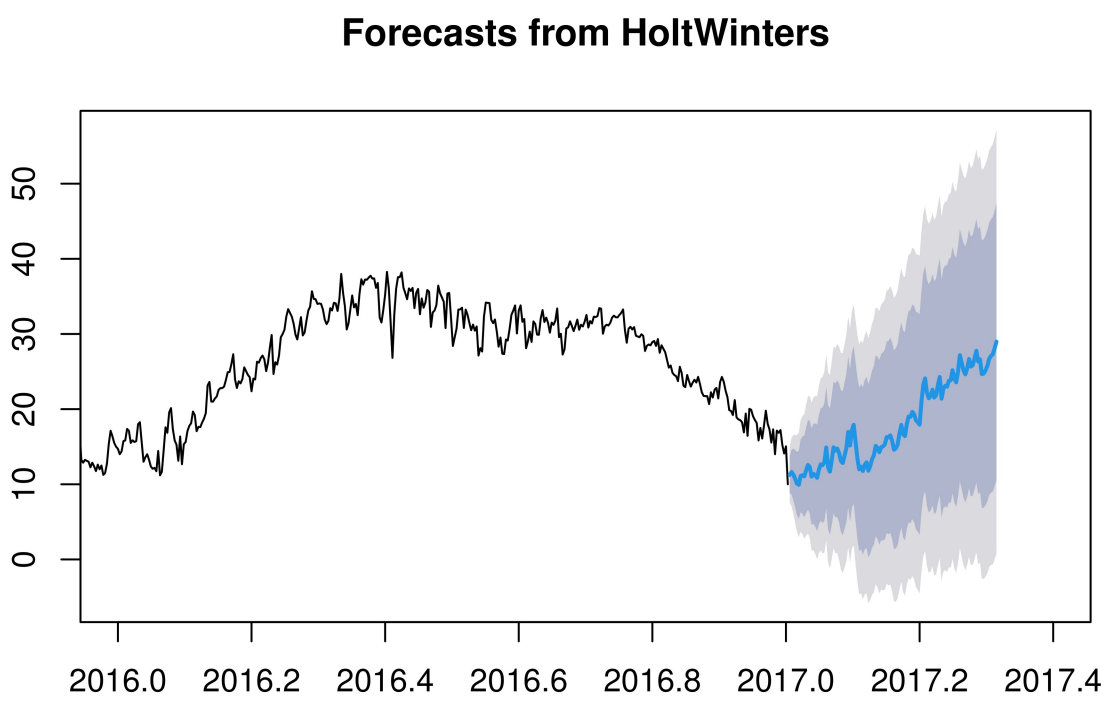
```
dec = decompose(t)
plot(dec)
```



From the decomposed plot we can infer the slight presence of trend and presence of seasonality in the data. The seasonality is present in the additive manner because from the original plot we could see there is no change in the width of seasonality period over time.

HoltWinter's forecasting using simple exponential smoothing

```
h =HoltWinters(t)
pred_h = forecast::forecast(h,h=114)
plot(pred_h,xlim=c(2016,2017.4))
```



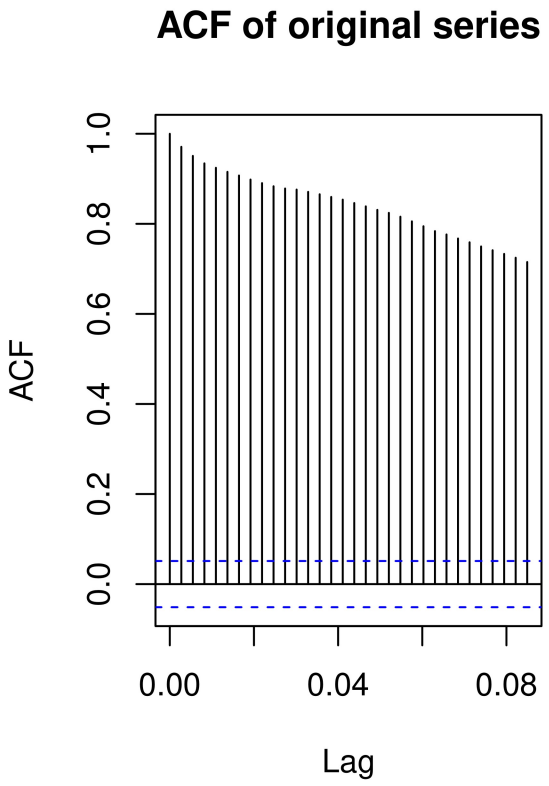

```
mse_h = sum((pred_h$mean-test$meantemp)^2)/114
mse_h
```

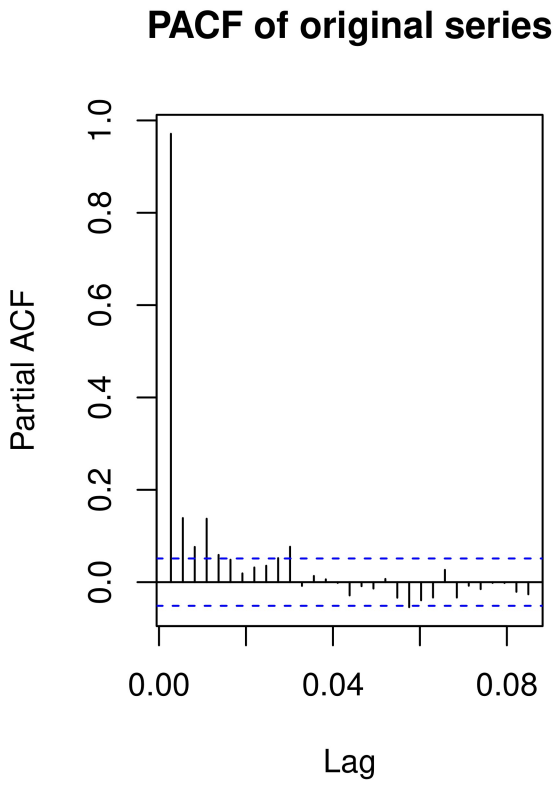
```
## [1] 21.17561
```

Since the data contains slight trend and seasonality so simple exponential smoothing will not give accurate predictions.

ACF and PACF of original data

```
par(mfrow=c(1,2))
acf(t,main='ACF of original series')
pacf(t,main='PACF of original series')
```





Observing the acf plot we can see that the data points are highly correlated which is an indication that the time series maybe non-stationary.

Checking for stationarity of data

```
tseries::adf.test(t)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -1.8526, Lag order = 11, p-value = 0.6407
## alternative hypothesis: stationary
```

The p-value is greater than 0.05 so we fail to reject null hypothesis and conclude that the data is non-stationary. Now we will apply first order differencing on the original data and check for stationarity.

```
t_diff = diff(t) # First order differencing
tseries::adf.test(t_diff)
```

```
##
##  Augmented Dickey-Fuller Test
##
```

```
## data:  t_diff
## Dickey-Fuller = -14.011, Lag order = 11, p-value = 0.01
## alternative hypothesis: stationary
```

```
tseries::pp.test(t_diff)
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  t_diff
## Dickey-Fuller Z(alpha) = -1374.7, Truncation lag parameter = 7, p-value
## = 0.01
## alternative hypothesis: stationary
```

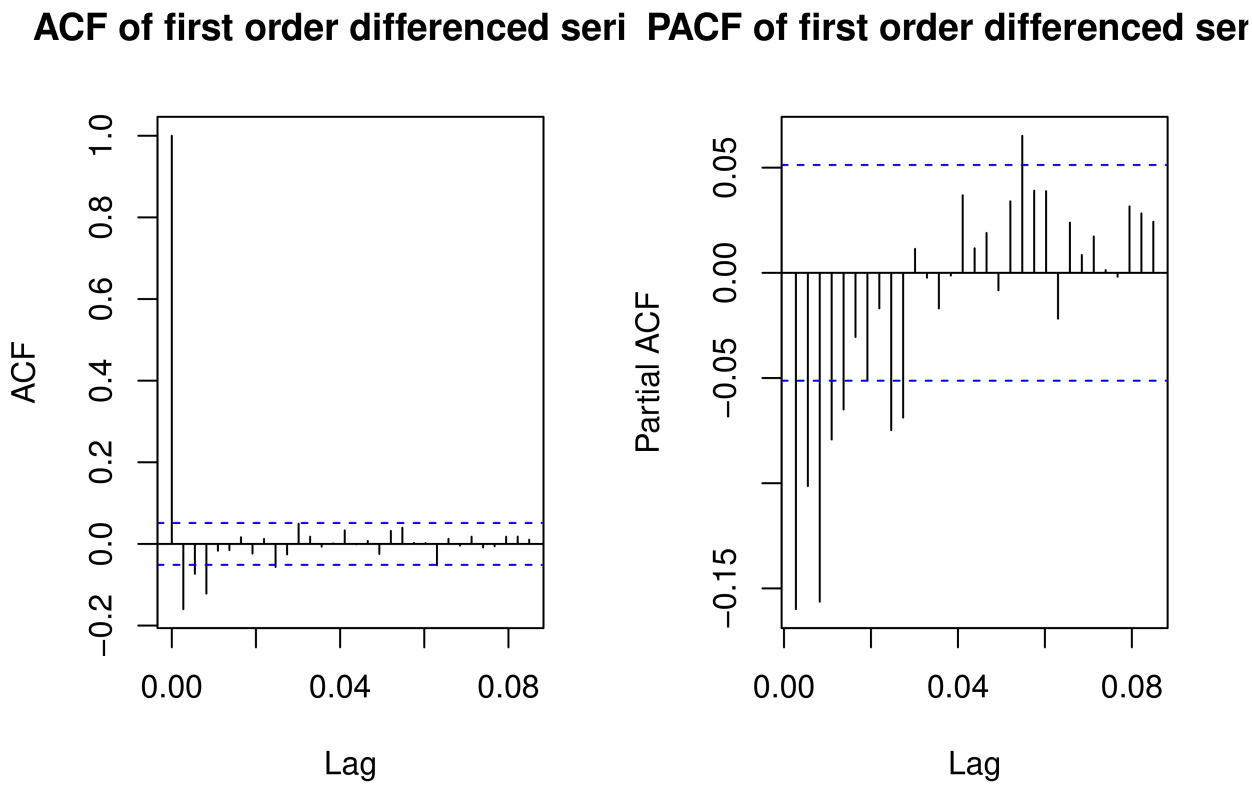
```
tseries::kpss.test(t_diff)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  t_diff
## KPSS Level = 0.16682, Truncation lag parameter = 7, p-value = 0.1
```

Applying ADF test, PP test and KPSS test we can observe that all the test conclude the first order differenced data is stationary.

ACF and PACF of first order differenced data

```
par(mfrow=c(1,2))
acf(t_diff,main='ACF of first order differenced series')
pacf(t_diff,main='PACF of first order differenced series')
```



From the acf and pacf graph we can guess that an appropriate model will be ARIMA(p = 3,d = 1,q = 1). But we also know that seasonal component is present in the data. So, we will try to implement a SARIMA model.

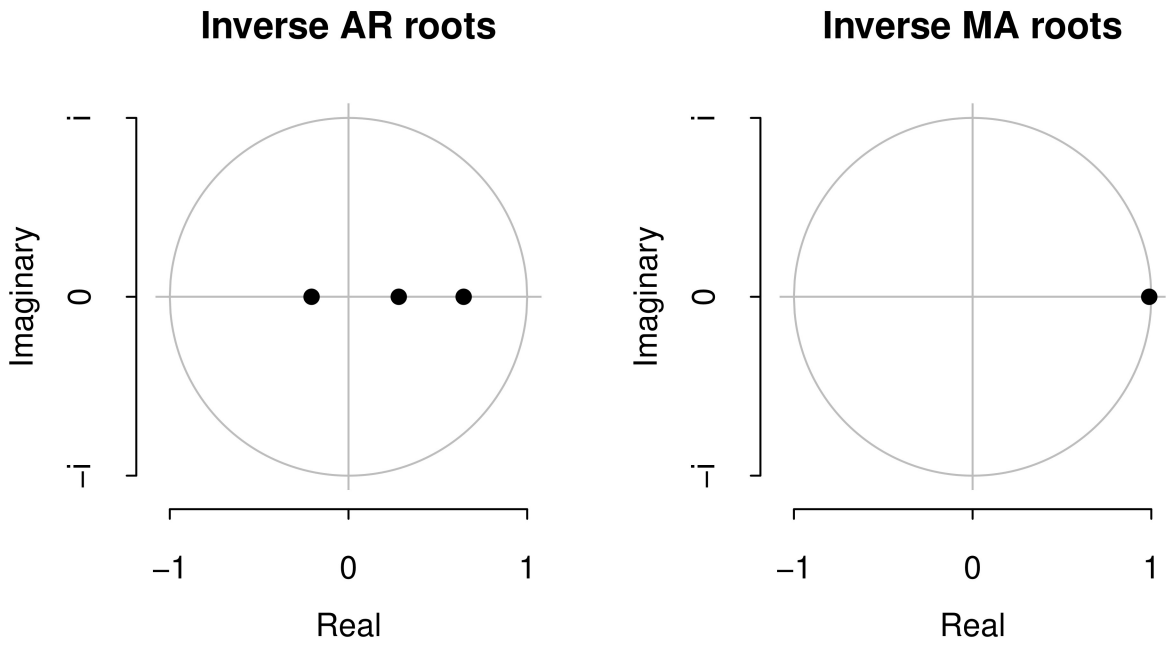
Modelling the data

```
model = auto.arima(t)
summary(model)
```

```
## Series: t
## ARIMA(3,1,1)(0,1,0)[365]
##
## Coefficients:
##          ar1     ar2      ar3      ma1
##       0.7188  0.0102  -0.0375  -0.9883
## s.e.  0.0311  0.0373   0.0311   0.0081
##
## sigma^2 estimated as 4.591:  log likelihood=-2392.92
## AIC=4795.84   AICc=4795.9   BIC=4820.84
##
## Training set error measures:
##                       ME     RMSE      MAE        MPE     MAPE     MASE
## Training set 0.01624948 1.851854 1.212622 -0.2772842 5.181115 0.474274
```

```
##                      ACF1
## Training set 0.002660008
```

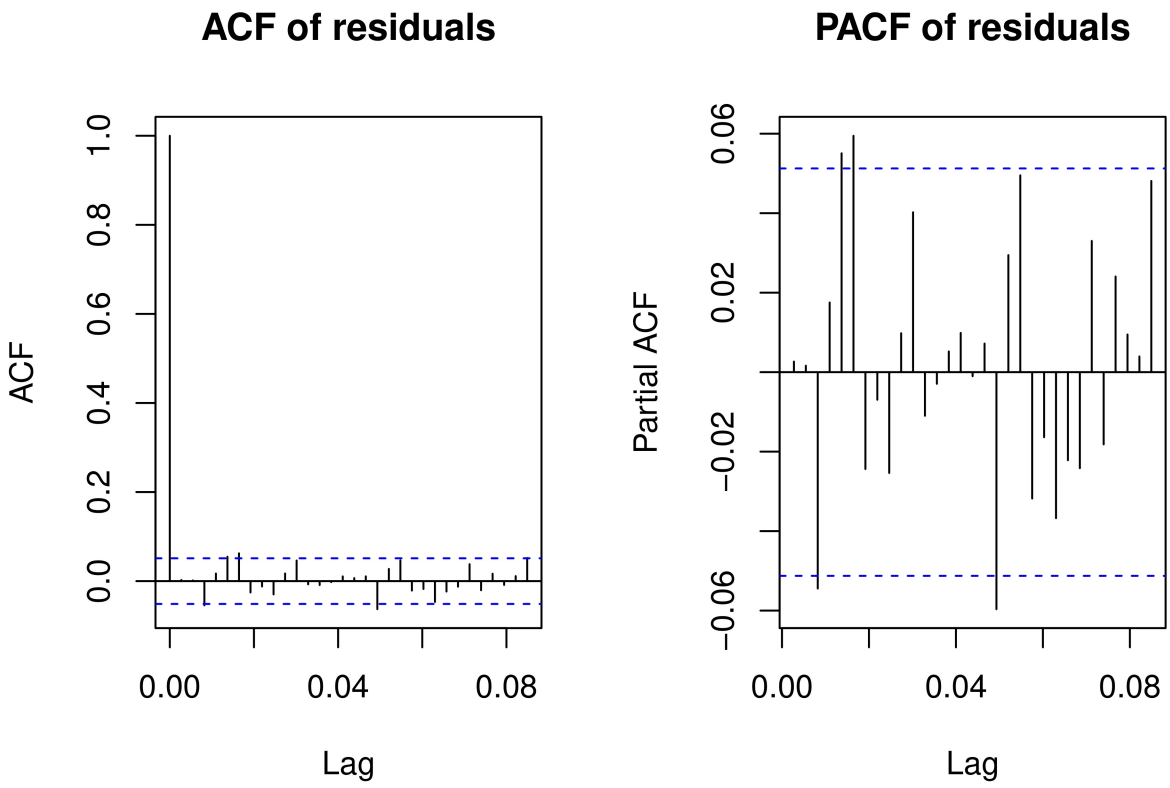
```
plot(model)
```



auto.arima() has calculated the appropriate model to be $SARIMA(3,1,1) \times (0,1,0)_{365}$. *s= 365 because the data is daily reported. Since inverse AR and MA roots lie inside the unit circle, the respective roots must lie outside the unit circle which is an evidence of stationarity.*

ACF and PACF of model residuals

```
res = model$residuals
par(mfrow=c(1,2))
acf(res,main='ACF of residuals')
pacf(res,main = 'PACF of residuals')
```



From the acf and pacf graphs we can conclude that there is no correlation among the residuals.

Test for residual mean t be zero

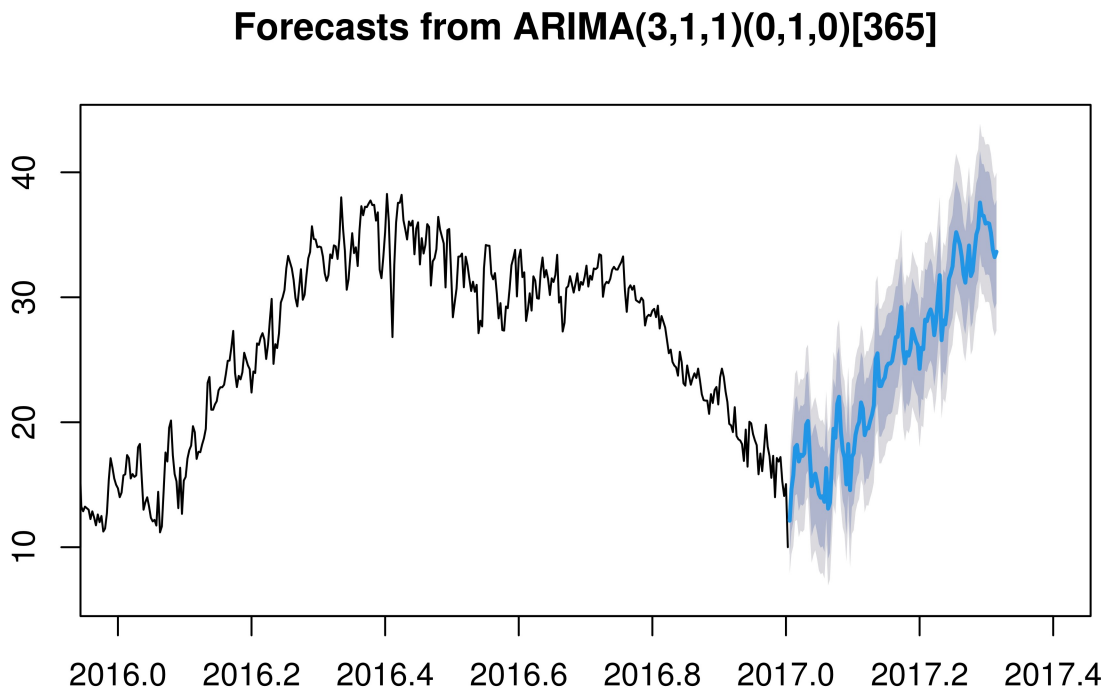
```
wilcox.test(res)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  res
## V = 560605, p-value = 0.109
## alternative hypothesis: true location is not equal to 0
```

Since the p-value is greater than 0.05 we accept the null hypothesis and conclude that the mean of the residuals is zero.

SARIMA forecast of the data

```
pred = forecast::forecast(model,h = 114)  # 114 steps ahead forecast because
                                          # test data has 114 observations
plot(pred,xlim = c(2016,2017.4))
```



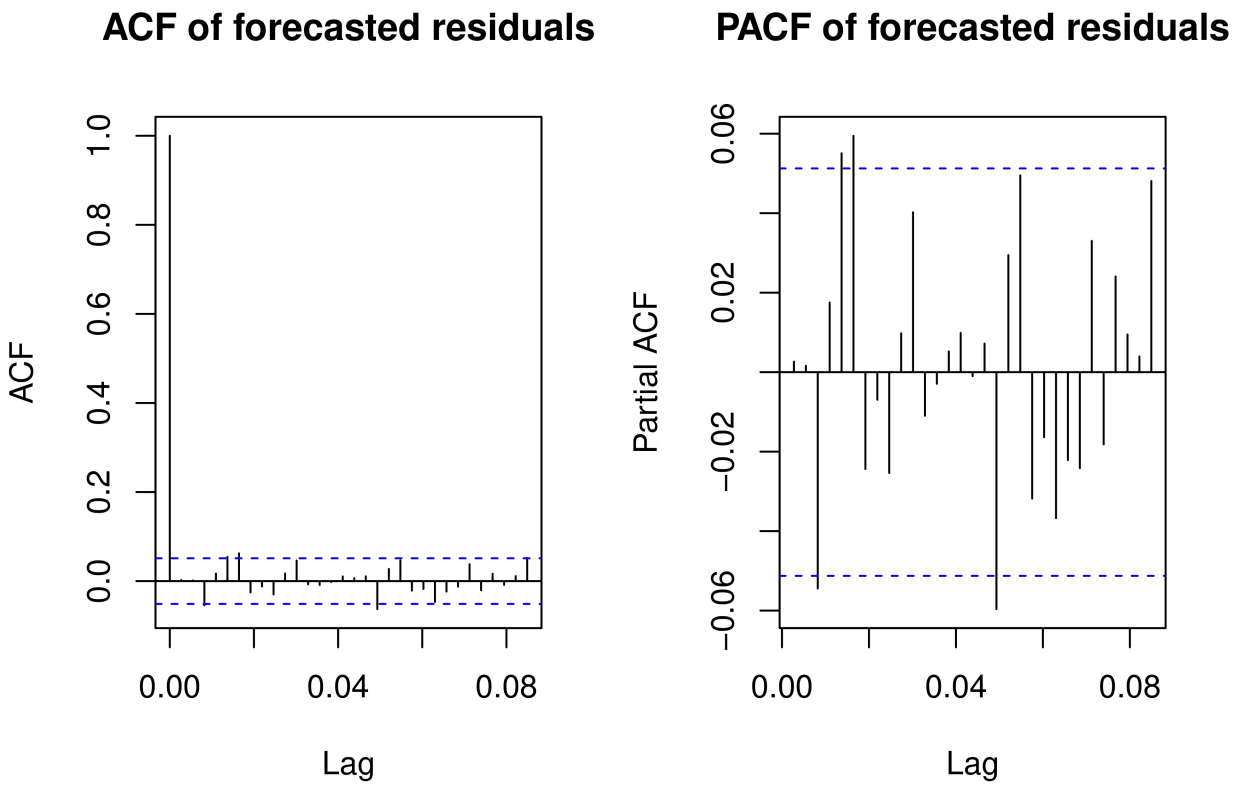

```
mse_s = sum((pred$mean-test$meantemp)^2)/114
c(mse_h,mse_s) # mse_h = HoltWinter's mse, mse_s = SARIMA mse
```

```
## [1] 21.17561 17.46407
```

We can see that mse_s is less than mse_h and conclude that SARIMA model gives a better prediction.

ACF and PACF of SARIMA forecasted data

```
par(mfrow=c(1,2))
acf(pred$residuals,main="ACF of forecasted residuals")
pacf(pred$residuals,main="PACF of forecasted residuals")
```



From the acf and pacf graph we can conclude that the residuals are uncorrelated

Test for SARIMA forecasted residuals mean to be zero

```
wilcox.test(pred$residuals)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  pred$residuals
## V = 560605, p-value = 0.109
## alternative hypothesis: true location is not equal to 0
```

The p-value is greater that 0.05 so we accept null hypothesis and conclude that the mean of residuals is zero