Delhi temperature timeseries data SARIMA model fitting

Here we are doing a time-series analysis of the daily temperature in the city of Delhi for the period of Jan 01, 1995 to Dec 31, 2019.

Reading the data

```
DelhiTemp <- read.csv(file = '../../../Data/Delhi_temperature_1995_2020.csv')
```

Viewing the data

```
head(DelhiTemp)
```

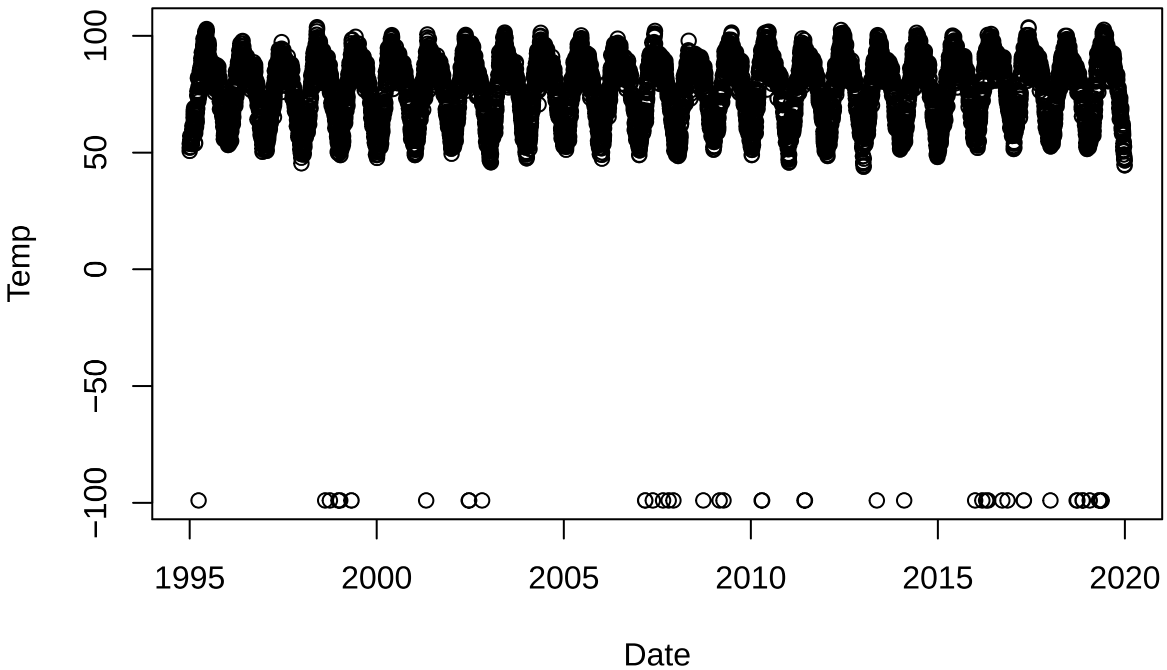
```
##   Month Day Year Temperature
## 1     1   1 1995        50.7
## 2     1   2 1995        52.1
## 3     1   3 1995        53.8
## 4     1   4 1995        53.7
## 5     1   5 1995        54.5
## 6     1   6 1995        54.3
```

Creating a date column and a YearMon column from Month, Day and Year columns.

```
DelhiTemp$Date <- as.Date(with(DelhiTemp, paste(Year, Month, Day,sep="-")), "%Y-%m-%d")
DelhiTemp$YearMon <- as.yearmon(paste(DelhiTemp$Year, DelhiTemp$Month), "%Y %m")
```

Temperature vs. Date plot

```
Temp <- DelhiTemp$Temperature
Date <- DelhiTemp$Date
plot(Temp~Date)
```



Our data shows clear yearly seasonality and their are some very large outliers (temperature~-100) possibly because of missing data for those dates.

Outlier detection and missing value treatment

Identifying outliers and replacing them with backfilled values:

```
DelhiTemp$Temperature[DelhiTemp$Temperature < 0] <- NA
which(is.na(DelhiTemp$Temperature))
```

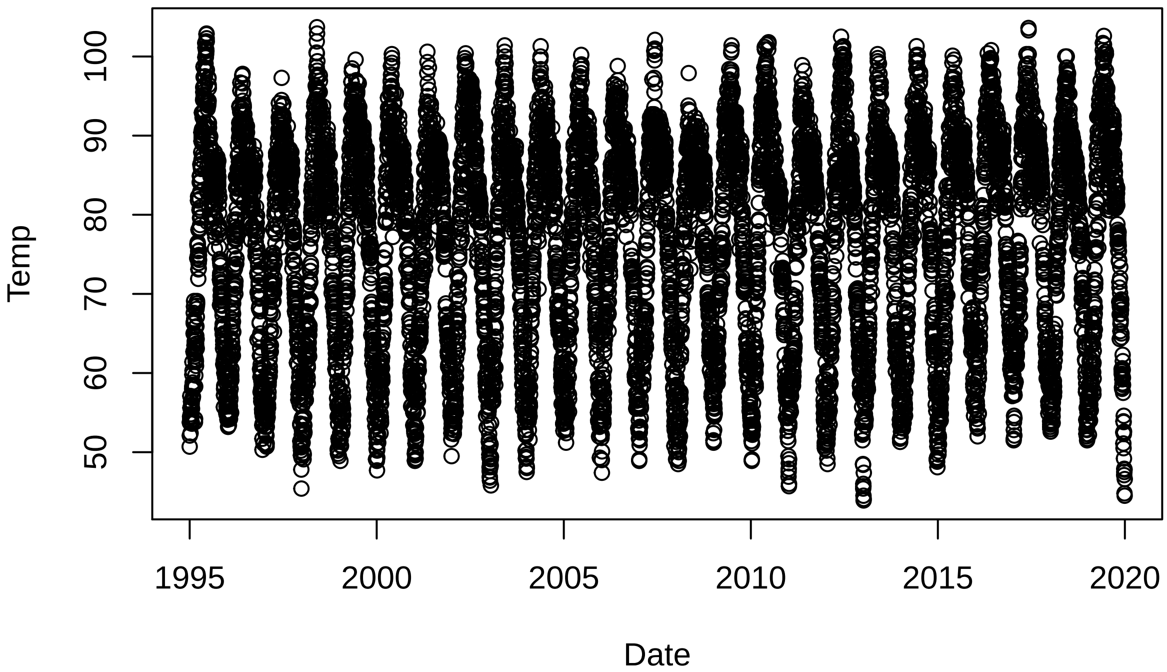
```
##  [1]   88 1324 1370 1454 1455 1460 1461 1471 1580 1581 2310 2726 2727 2728 2729
## [16] 2856 4448 4449 4523 4623 4678 4724 5016 5175 5213 5586 5587 5588 5590 6004
## [31] 6005 6006 6007 6008 6710 6977 7670 7740 7777 7796 7935 7986 8145 8146 8405
## [46] 8661 8668 8719 8720 8722 8789 8880 8881 8883 8886 8902 8903 8904 8905
```

```
DelhiTemp$Temperature <- na.locf(DelhiTemp$Temperature, fromLast = TRUE)
which(is.na(DelhiTemp$Temperature))
```

```
## integer(0)
```

Updated Temperature vs. Date plot

```
Temp <- DelhiTemp$Temperature
Date <- DelhiTemp$Date
plot(Temp~Date)
```



Binning

Binning temperatures to their mean monthly values

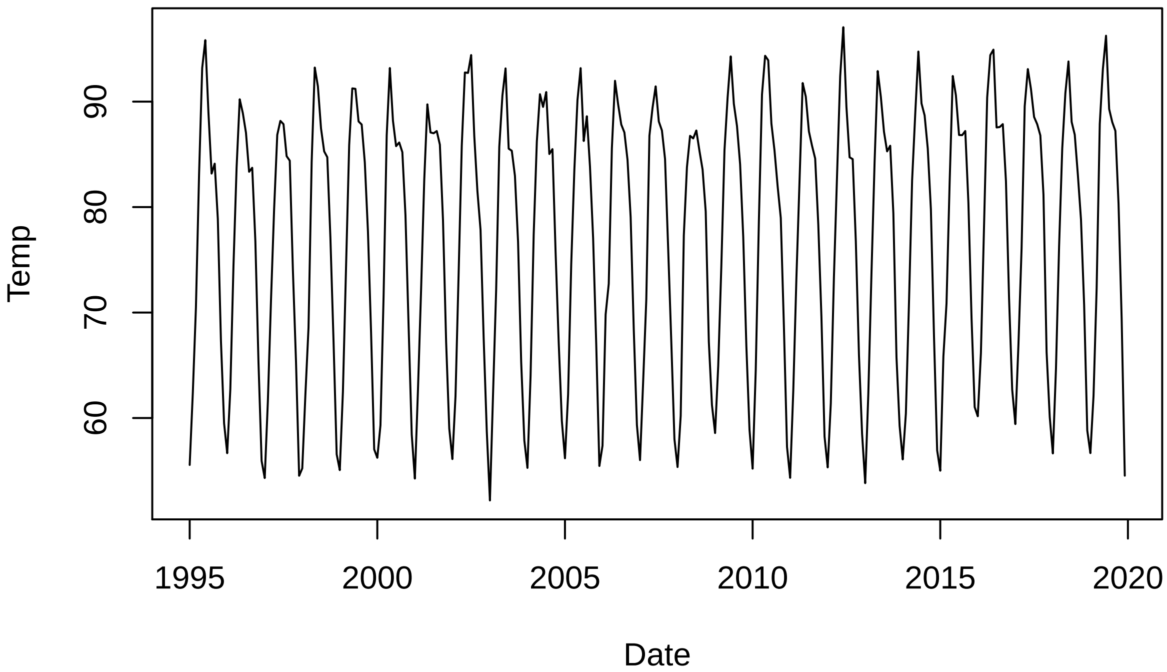
```
DelhiMonthlyTemp <- DelhiTemp %>% group_by("YearMon"=DelhiTemp$YearMon) %>% summarize(Temperature = mear

head(DelhiMonthlyTemp);
```

```
## # A tibble: 6 x 2
##   YearMon   Temperature
##   <yearmon>       <dbl>
## 1 Jan 1995         55.5
## 2 Feb 1995         62.4
## 3 Mar 1995         70.5
## 4 Apr 1995         83.1
## 5 May 1995         93.2
## 6 Jun 1995         95.8
```

Monthly mean temperature vs. Date (Month) plot

```
Temp <- DelhiMonthlyTemp$Temperature
Date <- DelhiMonthlyTemp$YearMon
plot(Temp~Date,type="l")
```



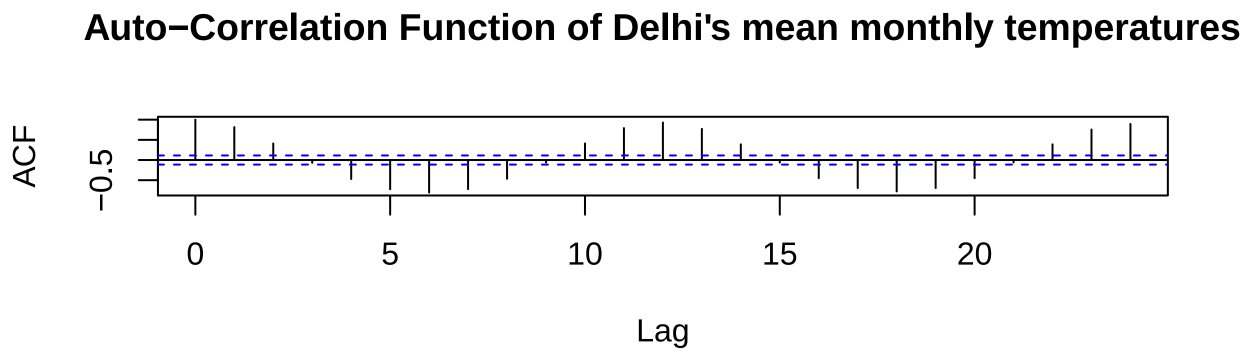
Once again we see that the data shows clear seasonality but no variation in variance so differencing should be enough for taking care of the trend.

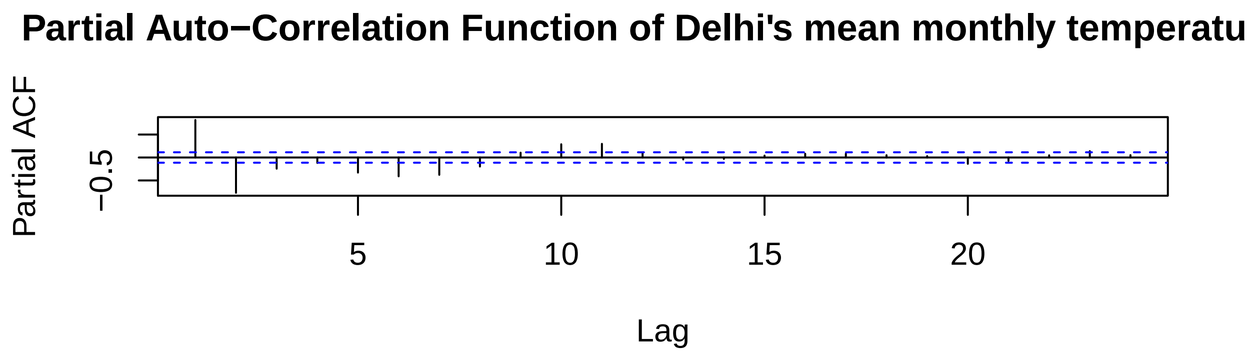
ACF and PACF plots

Let's first look at the auto-correlation function and the partial auto-correlation function plots.

```
Temp <- DelhiMonthlyTemp$Temperature

par(mfrow=c(2,1))
acf(Temp,main="Auto-Correlation Function of Delhi's mean monthly temperatures")
pacf(Temp,main="Partial Auto-Correlation Function of Delhi's mean monthly temperatures");
```





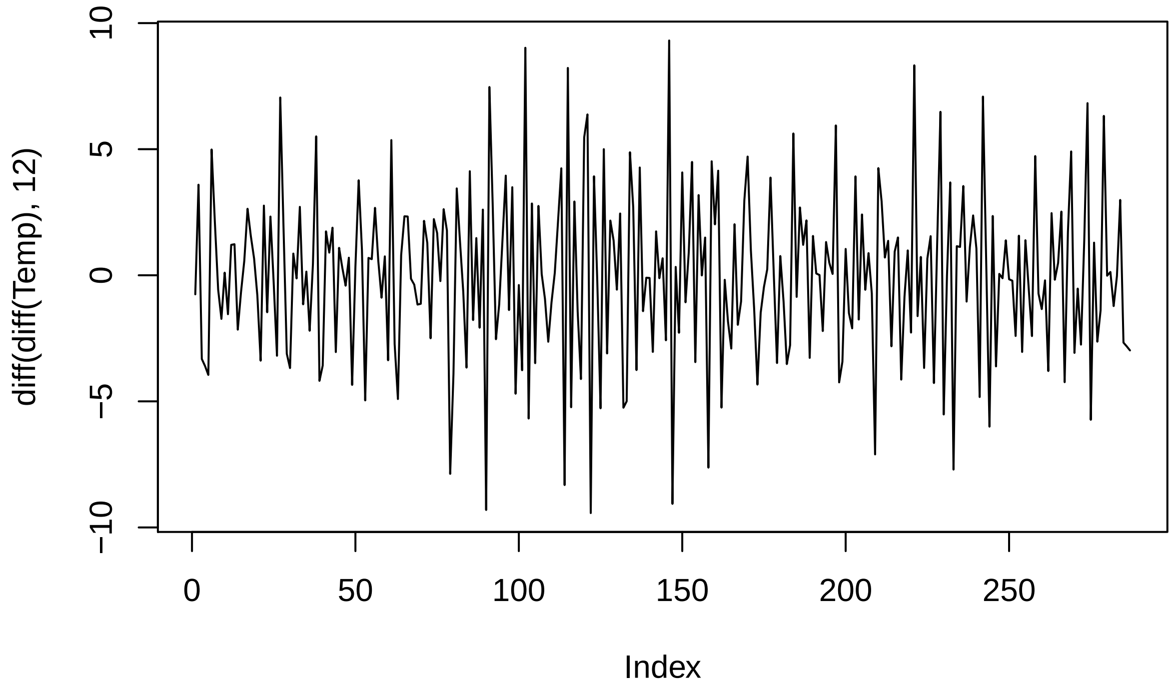
We see that the ACF plot also shows lots of correlation and clear seasonality.

Guessing the right orders for (S)ARIMA model fitting

1. Differncing orders (d, D)

Non-seasonal differencing -> *diff(data)* Seasonal differencing -> *diff(data,12)* Together -> *diff(diff(data),12)*

```
plot(diff(diff(Temp),12),type="l")
```



the plot shows almost no-trends except for a few large peaks at the center which may be outliers => d=1, D=1. We can also use the ADF test for checking the stationarity

```
diff_data <- diff(diff(Temp),12)
adf.test(diff_data)
```

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

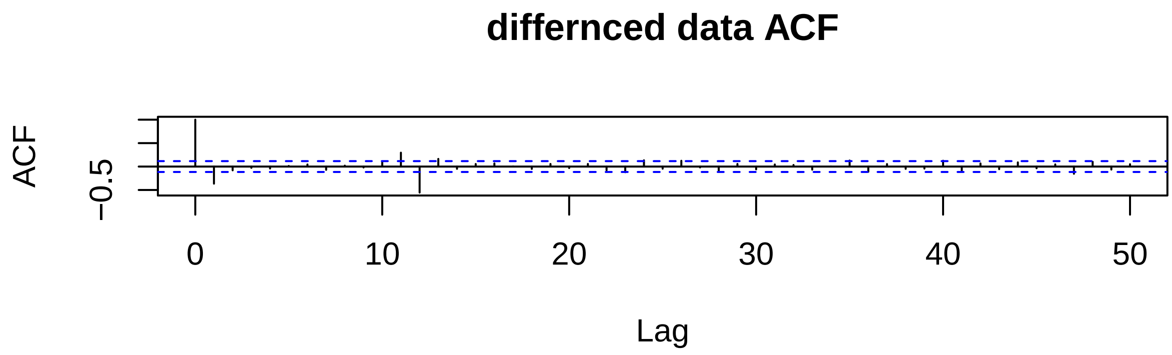
```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff_data
## Dickey-Fuller = -10.489, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

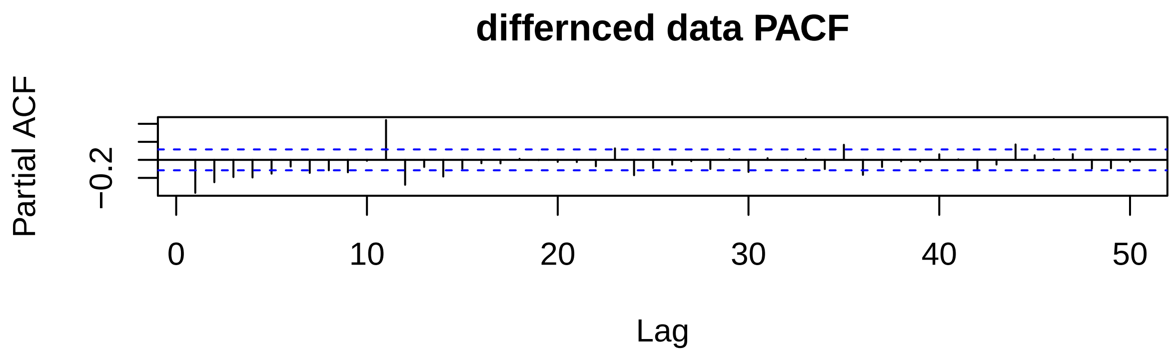
The test confirms that the differenced data is stationary.

2. orders for the auro-regressive (AR) and Moving Average (MA) terms i.e. p and q

ACF and PACF for differenced data

```
par(mfrow=c(2,1))
acf(diff(diff(Temp),12),main='differnced data ACF',50)
pacf(diff(diff(Temp),12),main='differnced data PACF',50);
```





The ACF plot shows significant correlations at lag=1,11 and 12 while the PACF shows significant correlation for lag=1,11 and 12 The correlations at later parts may be due to seasonality

Finding best parameters using

1. Grid Search

Trying for different values of p,q,P,Q and note down AIC, SSE and p-value (for Ljun-box-test). We want high p-values and small AIC and SSE using parsimony principle (simpler the better) while searching

```
d=1; DD=1; per=12

for(p in 1:2){
  for(q in 1:2){
    for(i in 1:6){
      for(j in 1:3){
        if(p+d+q+i+DD+j<=10){

          model<-arima(x=Temp, order = c((p-1),d,(q-1)), seasonal = list(order=c((i-1),DD,(j-1)), perio

          pval<-Box.test(model$residuals, lag=log(length(model$residuals)))

          sse<-sum(model$residuals^2)

          cat(p-1,d,q-1,i-1,DD,j-1,per, 'AIC=', model$aic, ' SSE=',sse,' p-VALUE=', pval$p.value,'\n')

        }
      }
    }
```

```
  }
}
```

```
## 0 1 0 0 1 0 12 AIC= 1499.338  SSE= 3099.001  p-VALUE= 8.952575e-08
## 0 1 0 0 1 1 12 AIC= 1320.604  SSE= 1476.127  p-VALUE= 8.894564e-07
## 0 1 0 0 1 2 12 AIC= 1320.109  SSE= 1502.587  p-VALUE= 1.212609e-06
## 0 1 0 1 1 0 12 AIC= 1392.531  SSE= 2087.502  p-VALUE= 1.215459e-06
## 0 1 0 1 1 1 12 AIC= 1319.835  SSE= 1500.886  p-VALUE= 1.226718e-06
## 0 1 0 1 1 2 12 AIC= 1320.854  SSE= 1488.524  p-VALUE= 1.356503e-06
## 0 1 0 2 1 0 12 AIC= 1370.082  SSE= 1901.339  p-VALUE= 7.147622e-07
## 0 1 0 2 1 1 12 AIC= 1321.25  SSE= 1485.317  p-VALUE= 1.148136e-06
## 0 1 0 2 1 2 12 AIC= 1322.845  SSE= 1486.992  p-VALUE= 1.336722e-06
## 0 1 0 3 1 0 12 AIC= 1349.824  SSE= 1740.855  p-VALUE= 4.246554e-06
## 0 1 0 3 1 1 12 AIC= 1322.842  SSE= 1494.211  p-VALUE= 1.265004e-06
## 0 1 0 4 1 0 12 AIC= 1344.303  SSE= 1686.293  p-VALUE= 3.323896e-06
## 0 1 1 0 1 0 12 AIC= 1409.009  SSE= 2237.481  p-VALUE= 0.02513918
## 0 1 1 0 1 1 12 AIC= 1238.17  SSE= 1070.666  p-VALUE= 0.01793177
## 0 1 1 0 1 2 12 AIC= 1239.052  SSE= 1060.471  p-VALUE= 0.01667548
## 0 1 1 1 1 0 12 AIC= 1319.848  SSE= 1607.407  p-VALUE= 0.01328627
## 0 1 1 1 1 1 12 AIC= 1239.027  SSE= 1060.696  p-VALUE= 0.01664074
## 0 1 1 1 1 2 12 AIC= 1240.96  SSE= 1061.698  p-VALUE= 0.01650034
## 0 1 1 2 1 0 12 AIC= 1295.577  SSE= 1453.315  p-VALUE= 0.007799349
## 0 1 1 2 1 1 12 AIC= 1240.994  SSE= 1061.651  p-VALUE= 0.01664093
## 0 1 1 3 1 0 12 AIC= 1282.807  SSE= 1369.994  p-VALUE= 0.001596375
## 1 1 0 0 1 0 12 AIC= 1460.192  SSE= 2683.762  p-VALUE= 5.570336e-05
## 1 1 0 0 1 1 12 AIC= 1288.522  SSE= 1281.156  p-VALUE= 2.515287e-05
## 1 1 0 0 1 2 12 AIC= 1289.366  SSE= 1269.078  p-VALUE= 2.049262e-05
## 1 1 0 1 1 0 12 AIC= 1361.389  SSE= 1860.666  p-VALUE= 7.996062e-05
## 1 1 0 1 1 1 12 AIC= 1289.222  SSE= 1272.065  p-VALUE= 2.021527e-05
## 1 1 0 1 1 2 12 AIC= 1290.516  SSE= 1267.385  p-VALUE= 2.03264e-05
## 1 1 0 2 1 0 12 AIC= 1338.392  SSE= 1691.143  p-VALUE= 9.26347e-05
## 1 1 0 2 1 1 12 AIC= 1290.295  SSE= 1270.33  p-VALUE= 2.401602e-05
## 1 1 0 3 1 0 12 AIC= 1324.173  SSE= 1585.588  p-VALUE= 6.74795e-05
## 1 1 1 0 1 0 12 AIC= 1385.017  SSE= 2016.254  p-VALUE= 0.9964005
## 1 1 1 0 1 1 12 AIC= 1221.998  SSE= 978.4572  p-VALUE= 0.9779703
## 1 1 1 0 1 2 12 AIC= 1223.462  SSE= 972.9903  p-VALUE= 0.9678169
## 1 1 1 1 1 0 12 AIC= 1298.865  SSE= 1459.958  p-VALUE= 0.9675623
## 1 1 1 1 1 1 12 AIC= 1223.441  SSE= 972.6935  p-VALUE= 0.9671796
## 1 1 1 2 1 0 12 AIC= 1277.296  SSE= 1331.71  p-VALUE= 0.9371032
```

2. Using auto.arima()

```
y <- msts(Temp, seasonal.periods=c(12))
auto.arima( y, d = 1, D = 1,  max.p = 5,  max.q = 5,  max.P = 5,  max.Q = 5, max.order = 10,  start.p =
```

```
## Series: y
## ARIMA(5,1,0)(5,1,0)[12]
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5     sar1     sar2     sar3
##       -0.5142  -0.4444  -0.3373  -0.2261  -0.1488  -0.8628  -0.6184  -0.4755
## s.e.   0.0606   0.0653   0.0695   0.0676   0.0603   0.0625   0.0823   0.0867
##          sar4     sar5
##       -0.2962  -0.1132
## s.e.   0.0850   0.0654
```

```
##
## sigma^2 estimated as 4.656:  log likelihood=-628.94
## AIC=1279.88   AICc=1280.84   BIC=1320.13
```

Best-model

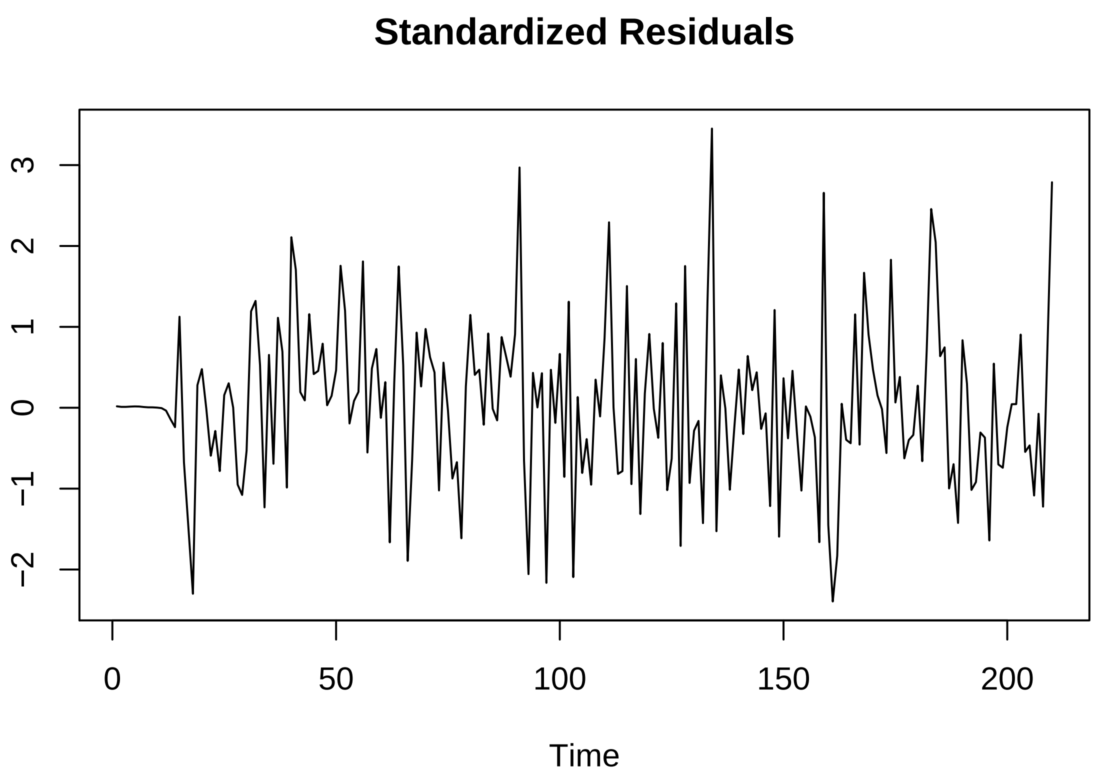
For some reason auto-arima is unable to reproduce the minimum value of AIC which was found in the grid-search method. From the grid-search the lowest AIC of 1221.998 is found for a 1,1,1,0,1,1,12 SARIMA model which also has a large enough p-value.

Train-test split

```
N = length(Temp)
n = 0.7*N
train = Temp[1:n]
test  = Temp[(n+1):N]
```

SARIMA(1,1,1,0,1,1,12) fitting results

```
model<-arima(x=train, order = c(1,1,1), seasonal = list(order=c(0,1,1), period=per))
standard_residuals<- model$residuals/sd(model$residuals)
plot(standard_residuals,ylab='',main='Standardized Residuals')
```



We see that the residuals look almost stationary which we also confirmed with the ADF test

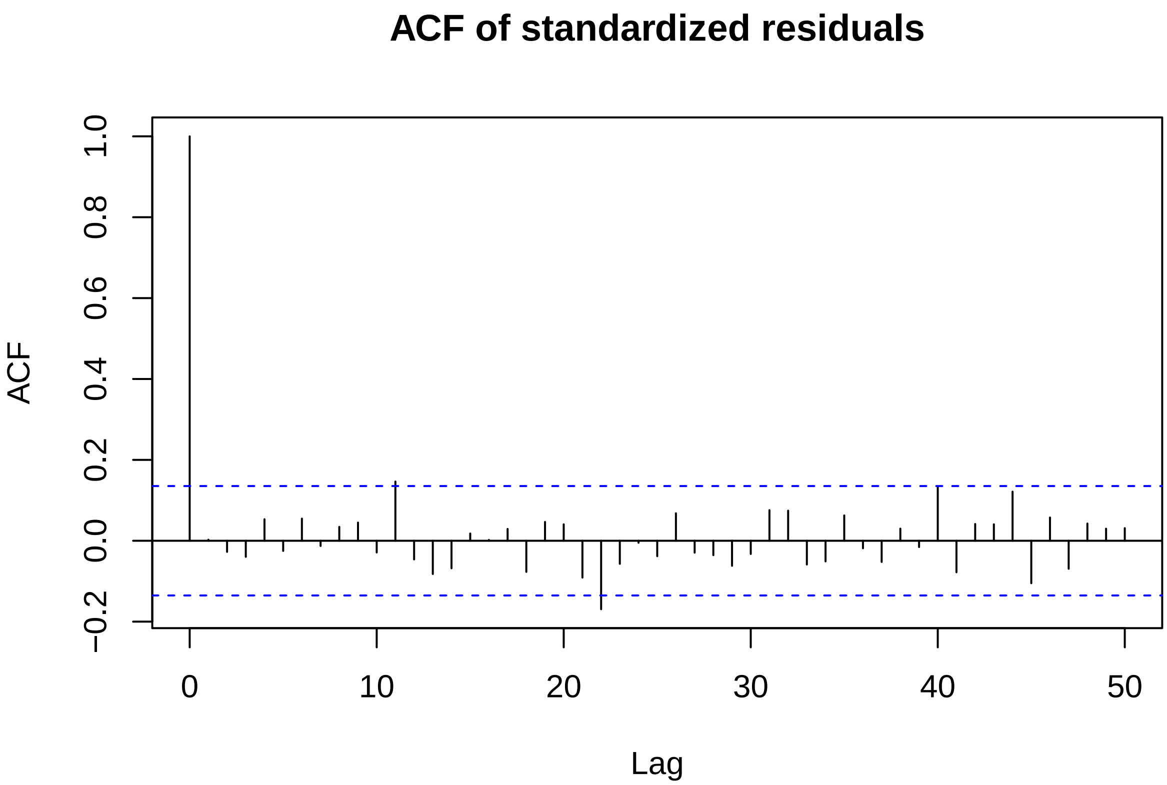
```
print(adf.test(standard_residuals))
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  standard_residuals
## Dickey-Fuller = -5.4662, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

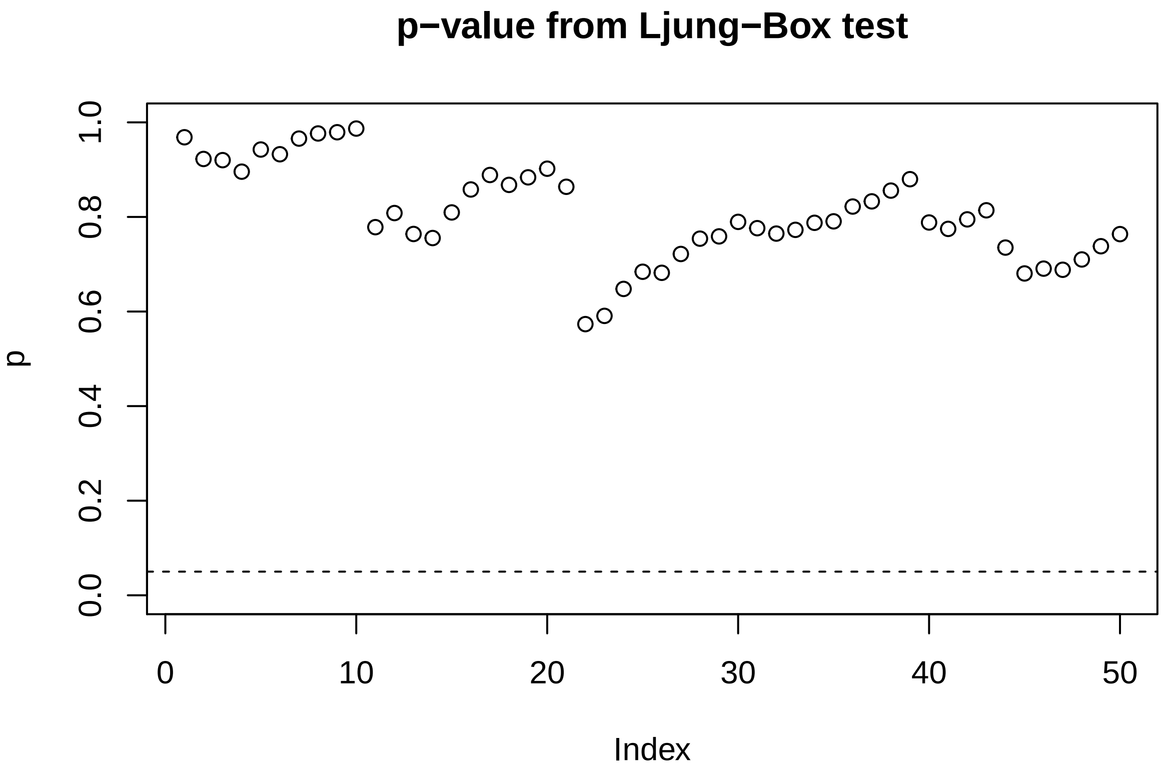
Let's check for correlations in the residual using the ACF plot

```
acf(standard_residuals,50,main='ACF of standardized residuals');
```



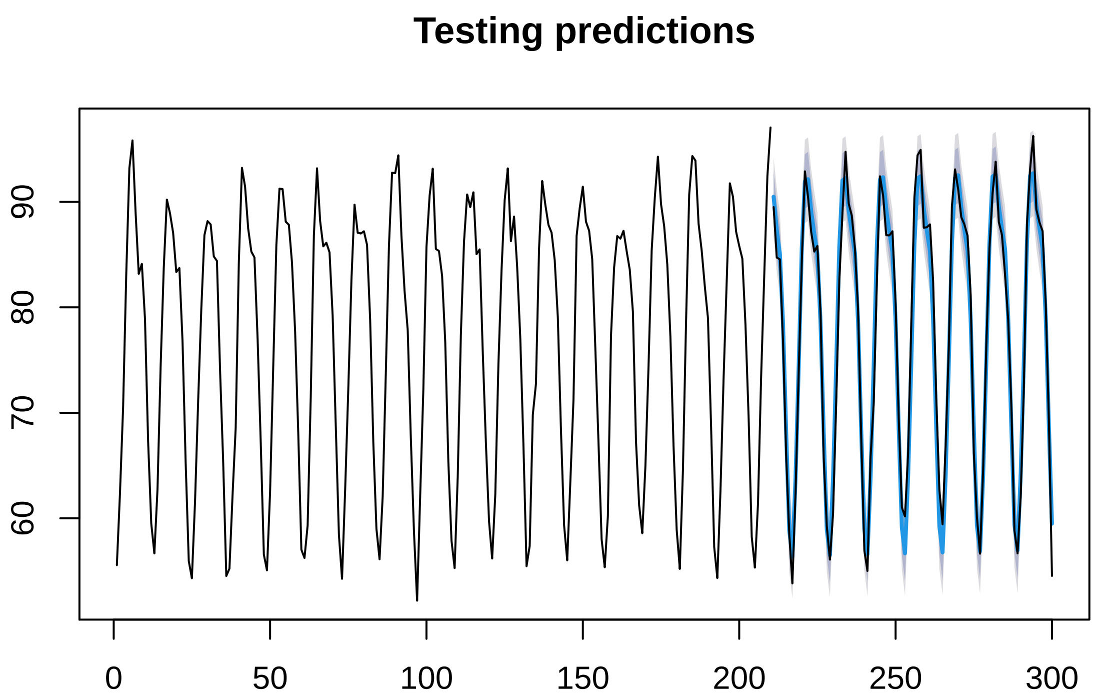
Next, we will perform a Ljung-Box test on the residuals. The null hypotheis for the test is:
H0: The dataset points are independently distributed (not correlated).
where a p-value of greater than 0.05 will be insifficient to reject the null hypothesis.

```
for (lag in seq(1:50)){
  pval<-Box.test(model$residuals, lag=lag)
  p[lag]=pval$p.value
}
plot(p,ylim = (0.0:1), main='p-value from Ljung-Box test')
abline(h=0.05,lty=2)
```



Any value above the dashed line (at y=0.05) is significant. We see that the p-values of the Ljung-Box test at all the lags are significant and therefore the hypothesis that the residuals are not correlated cannot be rejected.

```
model<-arima(x=train, order = c(1,1,1), seasonal = list(order=c(0,1,1), period=per))
pred_len=length(test)
plot(forecast(model, h=pred_len),main='Testing predictions')
train_x = seq(length(train)+1,length(train)+length(test))
lines(train_x,test)
```



Here the black lines in the first part (left) shows the training data and those in the second part shows the test data which alos has blue lines overlaid on it showing the predictions from our model which seem to match the test data pretty well. The small shaded region on the blue lines shows the confidence interval (difficult to resolve here but it actually consists of two different dark and light shaded regions showing the 80% and 95% confidence regions).

Forecasting using the best-model

```
model<-arima(x=Temp, order = c(1,1,1), seasonal = list(order=c(0,1,1), period=per))
par(mfrow=c(1,1))
h=12 # forecasting for the 12 months after the end of the dataset
plot(forecast(model,h), main='Forecasts for next 12 months');
```

Forecasts for next 12 months