Weather Forcasting

Mohit Abbi

3/29/2018

Installing necessary Libraries

```
library(forecast)
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
```

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

```
## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff
```

```
library(tseries)
```

Loading TempUSA data

```
Temp <- read.csv("C:/Users/Dell-pc/Desktop/Github/Projects/Weather Forecasting/TempUSA.csv", hea
der = TRUE)
#View(Temp)
str(Temp)
```

```
## 'data.frame':    154 obs. of  6 variables:
##  $ Date         : Factor w/ 154 levels "2000M01","2000M02",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ LosAngelesMin: num  10 10.1 10.1 12.5 14.2 16.6 16.8 18.2 16.7 13.8 ...
##  $ NYMin        : num  -5.6 -2.1 1.6 4.7 11.1 15.9 18.1 18.5 14.7 8.3 ...
##  $ HoustonMin   : num  7.4 9.3 12.5 12.6 20.4 21.7 22.2 22 19.1 15.2 ...
##  $ ChicagoMin   : num  -8.1 -2.6 0.8 2.7 10.4 13.8 16.4 16.7 12 7.6 ...
##  $ SeattleMin   : num  1.4 2.6 2.9 6.2 8 10.7 12.3 11.7 10.6 7.3 ...
```

```
summary(Temp)
```

```
##       Date     LosAngelesMin       NYMin          HoustonMin    
##  2000M01:  1   Min.   : 6.90   Min.   :-7.400   Min.   : 2.600  
##  2000M02:  1   1st Qu.:10.10   1st Qu.:-0.200   1st Qu.: 9.425  
##  2000M03:  1   Median :12.75   Median : 8.400   Median :15.200  
##  2000M04:  1   Mean   :13.04   Mean   : 8.112   Mean   :15.322  
##  2000M05:  1   3rd Qu.:16.07   3rd Qu.:16.400   3rd Qu.:21.775  
##  2000M06:  1   Max.   :19.60   Max.   :21.900   Max.   :25.400  
##  (Other):148                                                    
##    ChicagoMin        SeattleMin    
##  Min.   :-13.800   Min.   :-0.400  
##  1st Qu.: -3.150   1st Qu.: 2.950  
##  Median :  5.600   Median : 6.400  
##  Mean   :  5.131   Mean   : 6.842  
##  3rd Qu.: 13.700   3rd Qu.:10.900  
##  Max.   : 20.900   Max.   :14.200  
## 
```

Converting data into time series

```
temptrend<-ts(Temp, frequency=12, start =c(2000,1))
is.ts(temptrend)
```

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

```
str(temptrend)
```

```
##  Time-Series [1:154, 1:6] from 2000 to 2013: 1 2 3 4 5 6 7 8 9 10 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:6] "Date" "LosAngelesMin" "NYMin" "HoustonMin" ...
```

```
Temp1=Temp$LosAngelesMin
#View(Temp1)
str(Temp1)
```

```
##  num [1:154] 10 10.1 10.1 12.5 14.2 16.6 16.8 18.2 16.7 13.8 ...
```

For decomposition plot, I’ll use stl: seasonal trend decomposition

Make a stl object #TempLATS

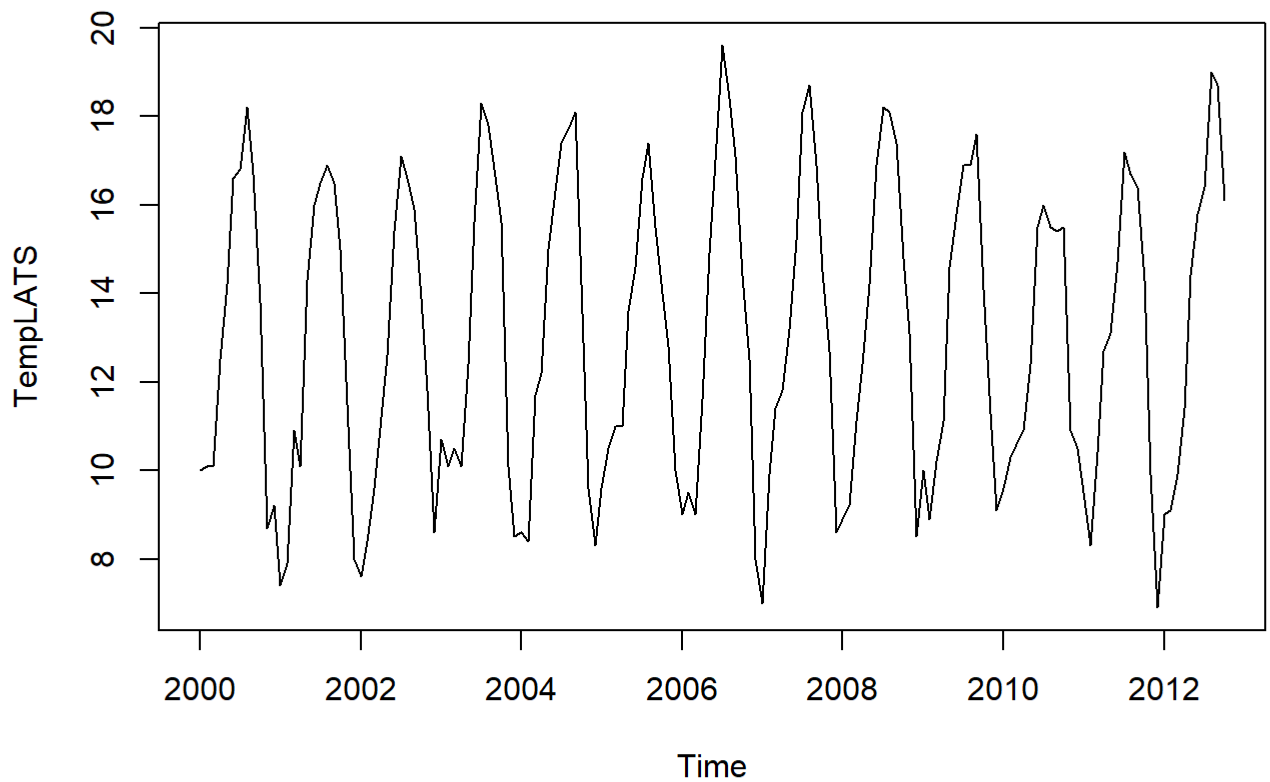
```
TempLATS<-ts(Temp1, frequency=12, start =c(2000,1) )
str(TempLATS)
```

```
##  Time-Series [1:154] from 2000 to 2013: 10 10.1 10.1 12.5 14.2 16.6 16.8 18.2 16.7 13.8 ...
```

```
is.ts(TempLATS)
```

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

```
plot(TempLATS)
```




```
class(TempLATS)
```

```
## [1] "ts"
```

```
adf.test(TempLATS)
```

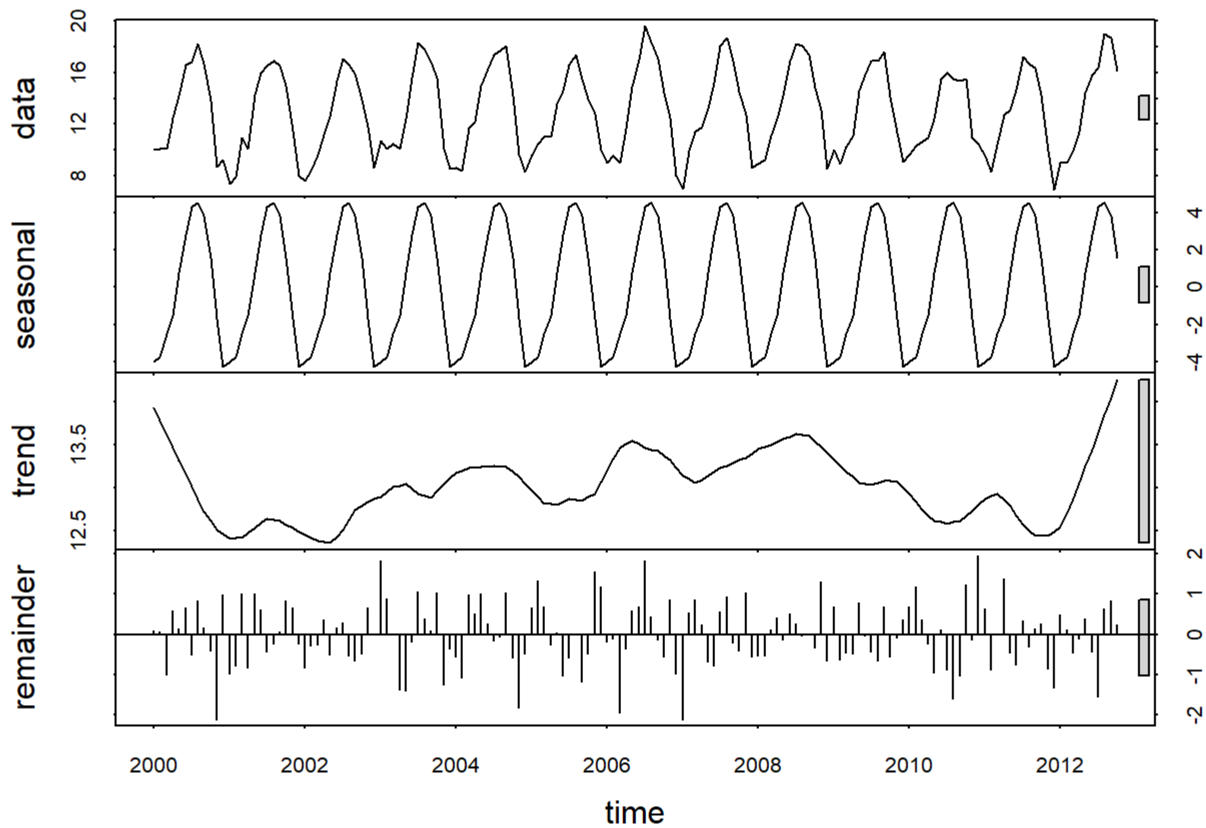
```
## Warning in adf.test(TempLATS): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  TempLATS
## Dickey-Fuller = -10.636, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```
is.ts(TempLATS)
```

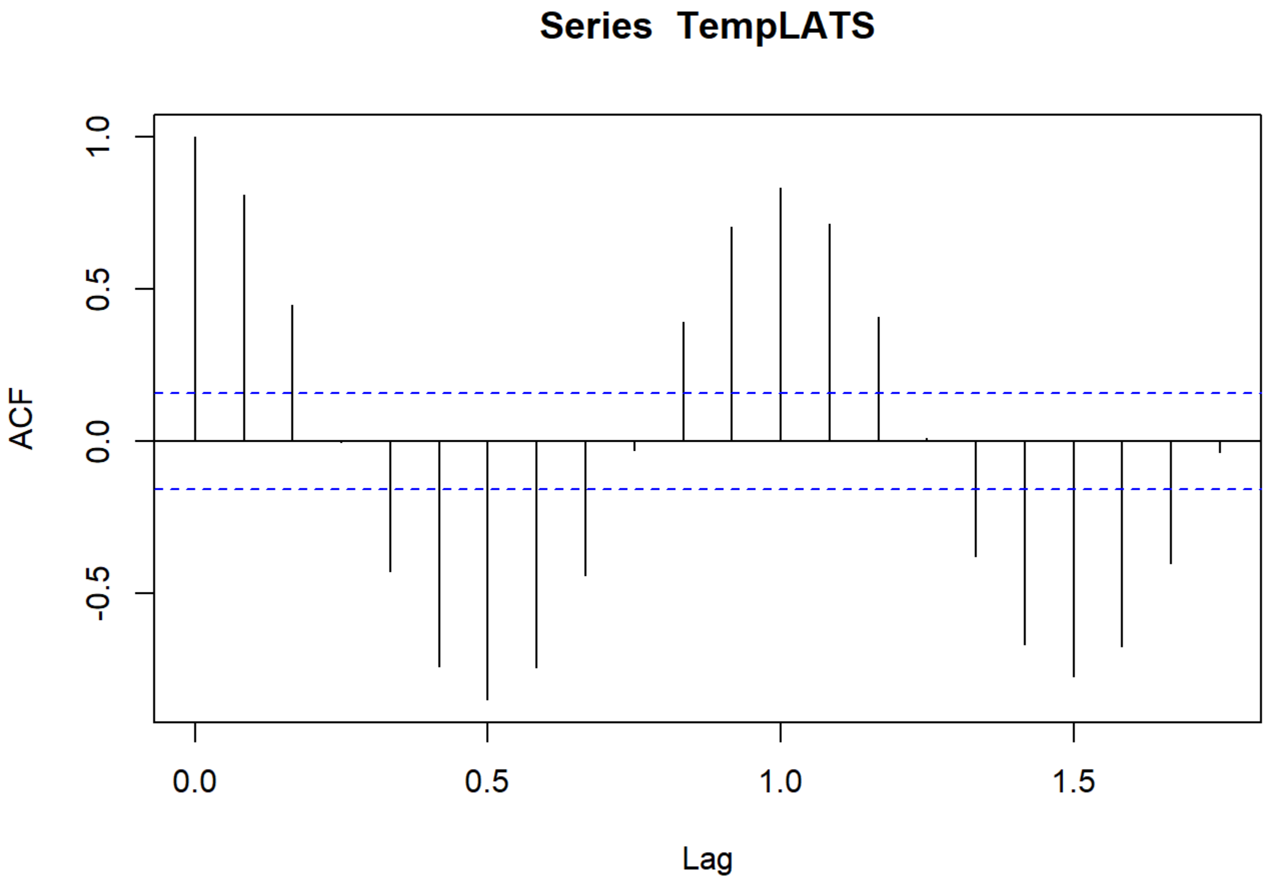
```
## [1] TRUE
```

```
LA=stl(TempLATS,s.window = "periodic")
#LA
plot(LA)
```

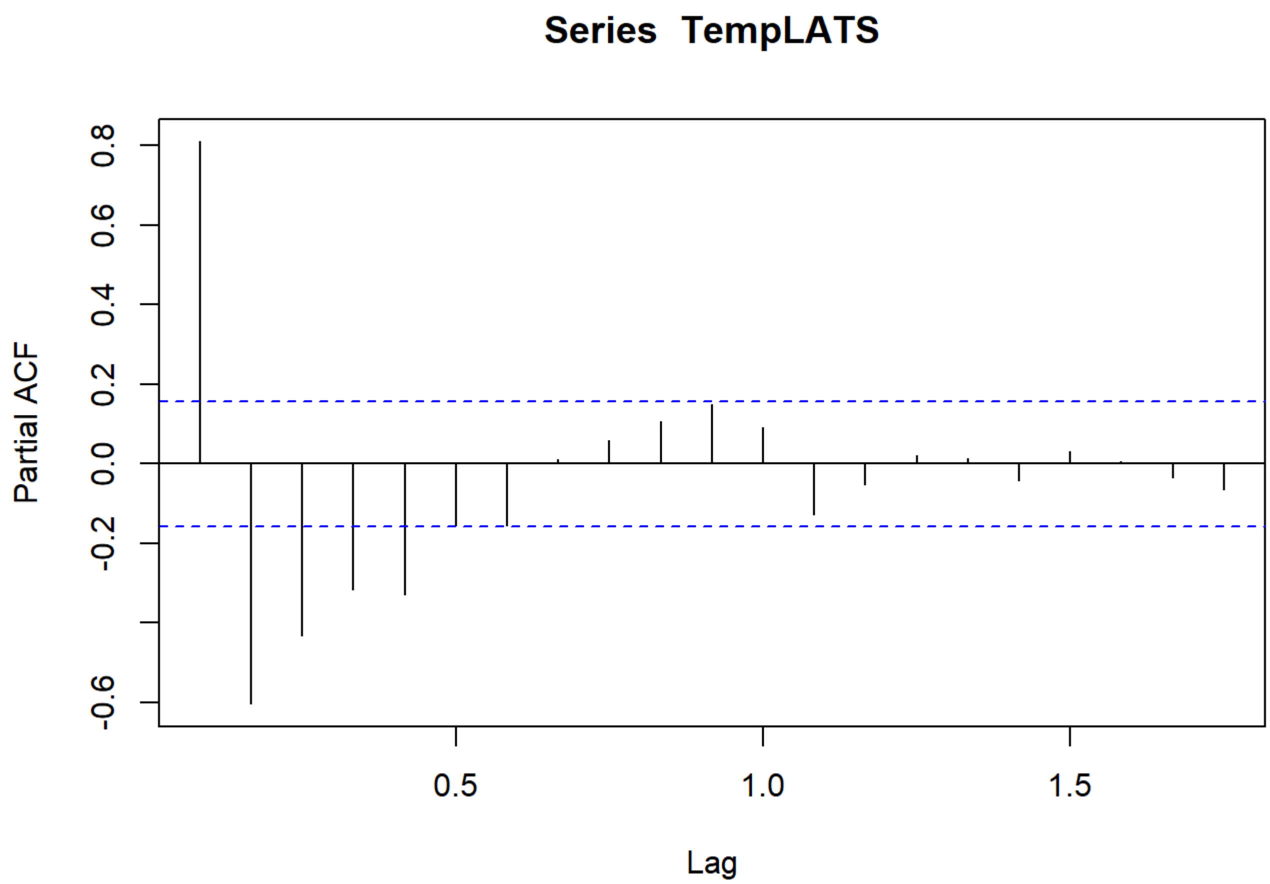


Plotting ACF and PACF

```
acf(TempLATS)
```




```
pacf(TempLATS)
```

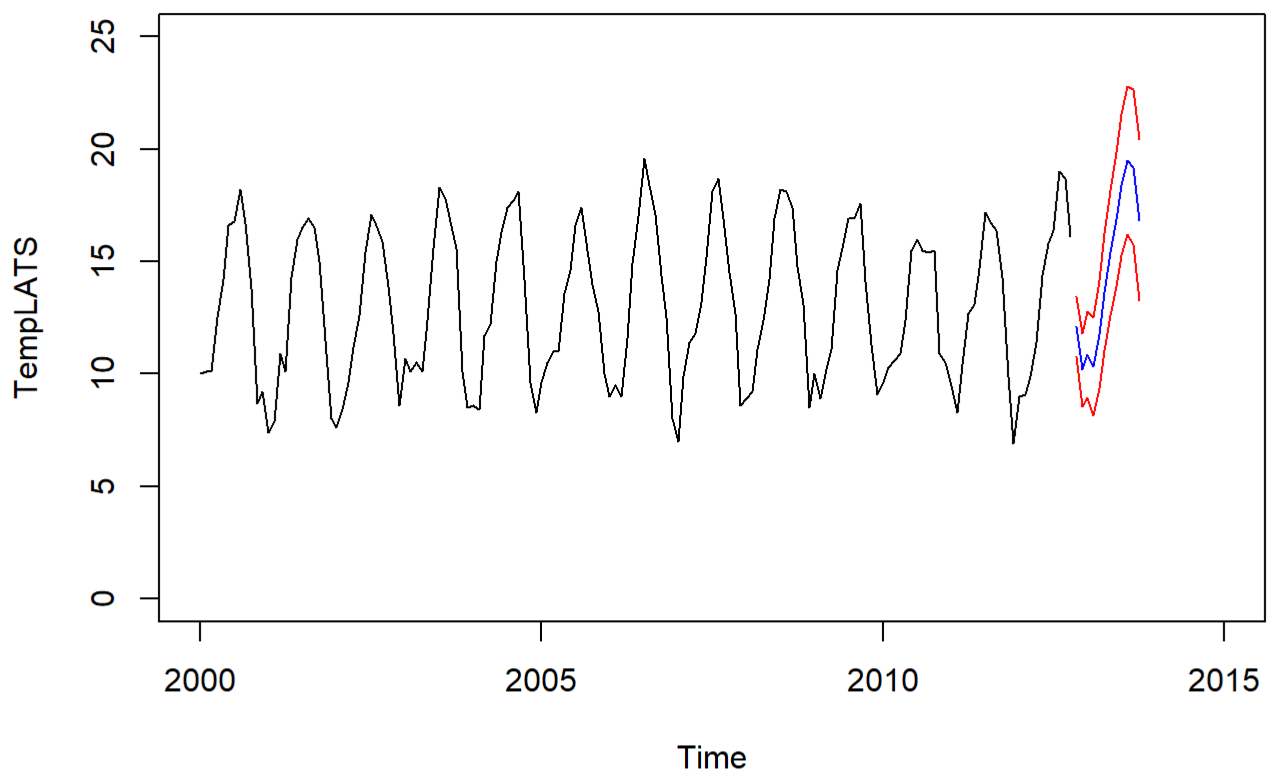


Generating Prediction(For LA)

```
LA.fit <- arima(TempLATS, order =c(1,1,0), seasonal = list(order = c(1,1,0), period =12), includ
e.mean = FALSE)
LA.fit
```

```
##
## Call:
## arima(x = TempLATS, order = c(1, 1, 0), seasonal = list(order = c(1, 1, 0),
##     period = 12), include.mean = FALSE)
##
## Coefficients:
##           ar1     sar1
##       -0.3273  -0.4778
## s.e.   0.0799   0.0788
##
## sigma^2 estimated as 1.804:  log likelihood = -243.27,  aic = 492.54
```

```
LA.pred <- predict(LA.fit, n.ahead = 12)
plot(TempLATS,xlim=c(2000,2015), ylim=c(0,25))
lines(LA.pred$pred, col="blue")
lines(LA.pred$pred+LA.pred$se, col="RED")
lines(LA.pred$pred-LA.pred$se, col="RED")
```

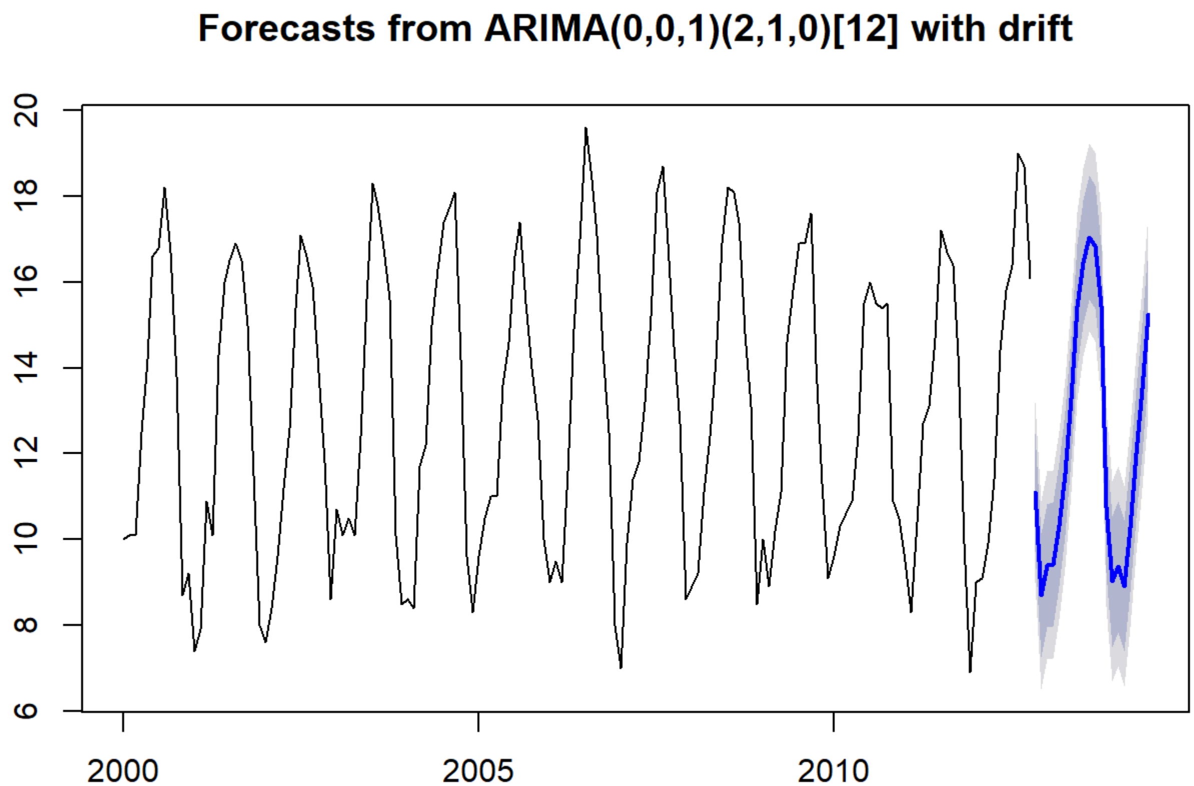


Forecasting using auto.arima

```
TemLA = auto.arima(TempLATS)
summary(TemLA)
```

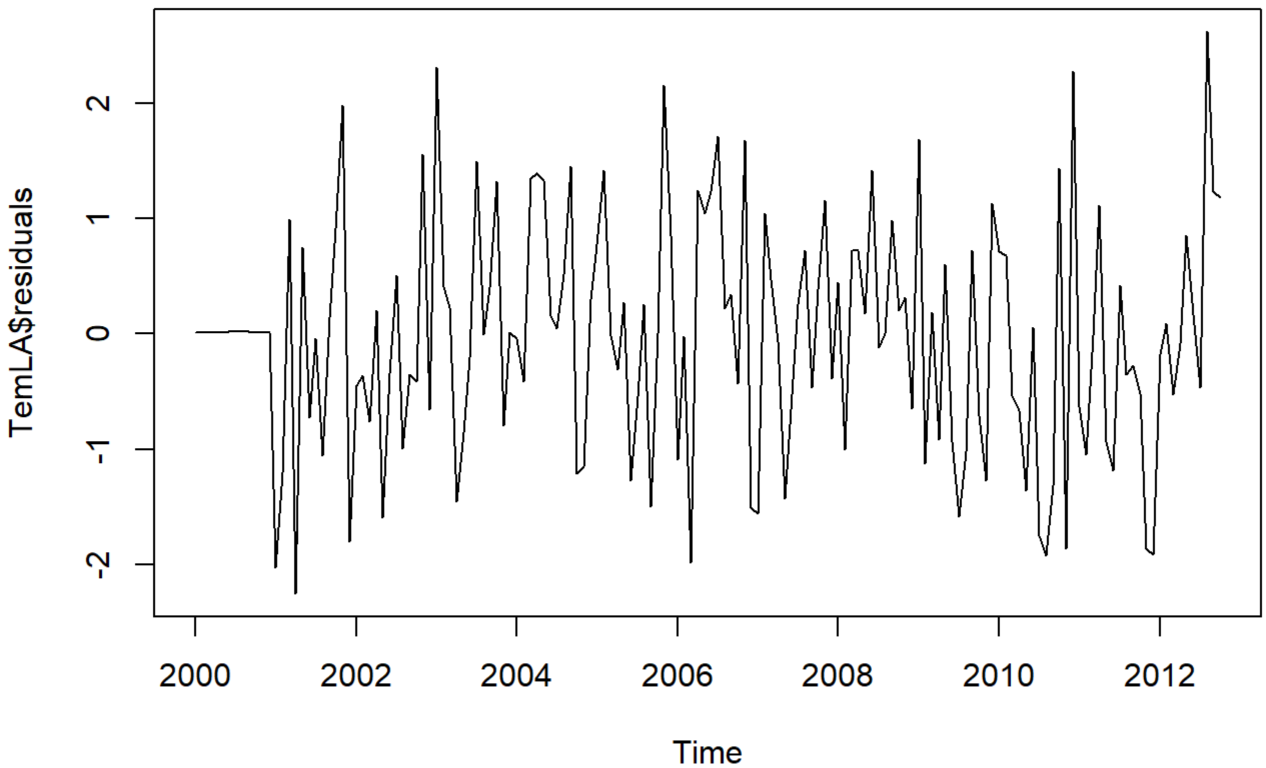
```
## Series: TempLATS
## ARIMA(0,0,1)(2,1,0)[12] with drift
##
## Coefficients:
##          ma1     sar1     sar2   drift
##       0.3009  -0.6499  -0.4057  0.0012
## s.e.  0.0761   0.0868   0.0846  0.0050
##
## sigma^2 estimated as 1.153:  log likelihood=-213.23
## AIC=436.45   AICc=436.89   BIC=451.23
##
## Training set error measures:
##                       ME     RMSE       MAE        MPE     MAPE      MASE
## Training set -0.02330697 1.016543 0.7951111 -0.8234918 6.713955 0.7186873
##                     ACF1
## Training set 0.01976779
```

```
forecLA = forecast(TemLA,h=20)
plot(forecast(forecLA))
```

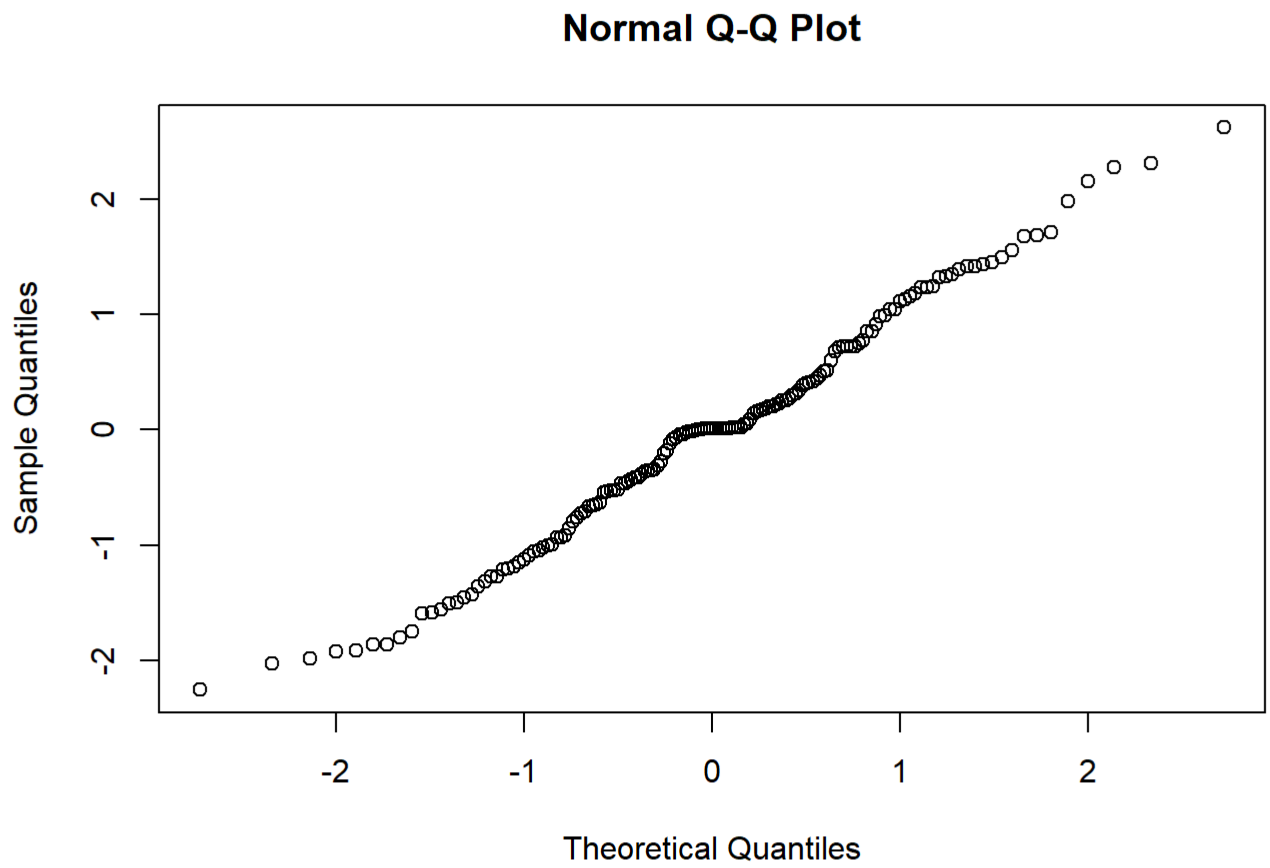


Autocorrelation

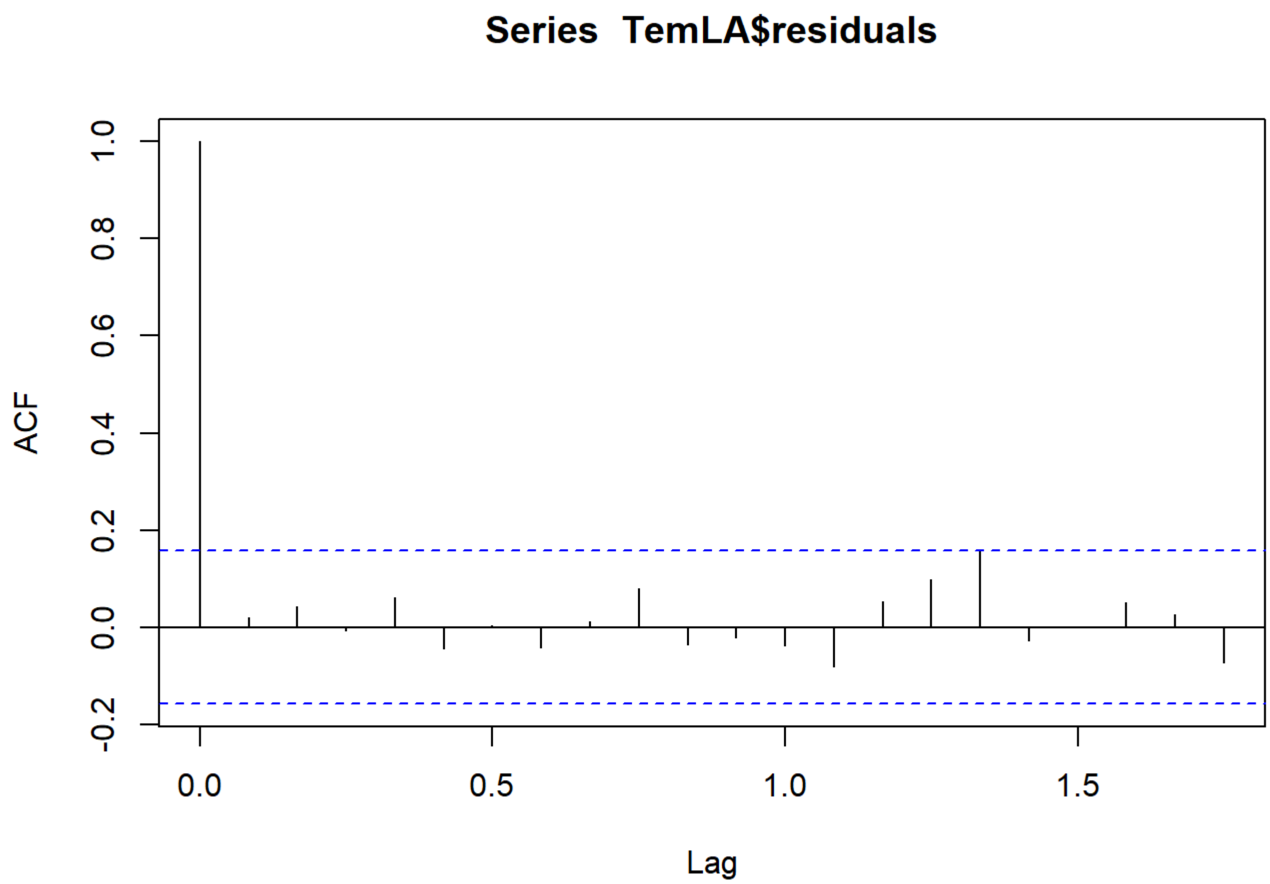
```
plot.ts(TemLA$residuals)
```




```
qqnorm(TemLA$residuals)
```

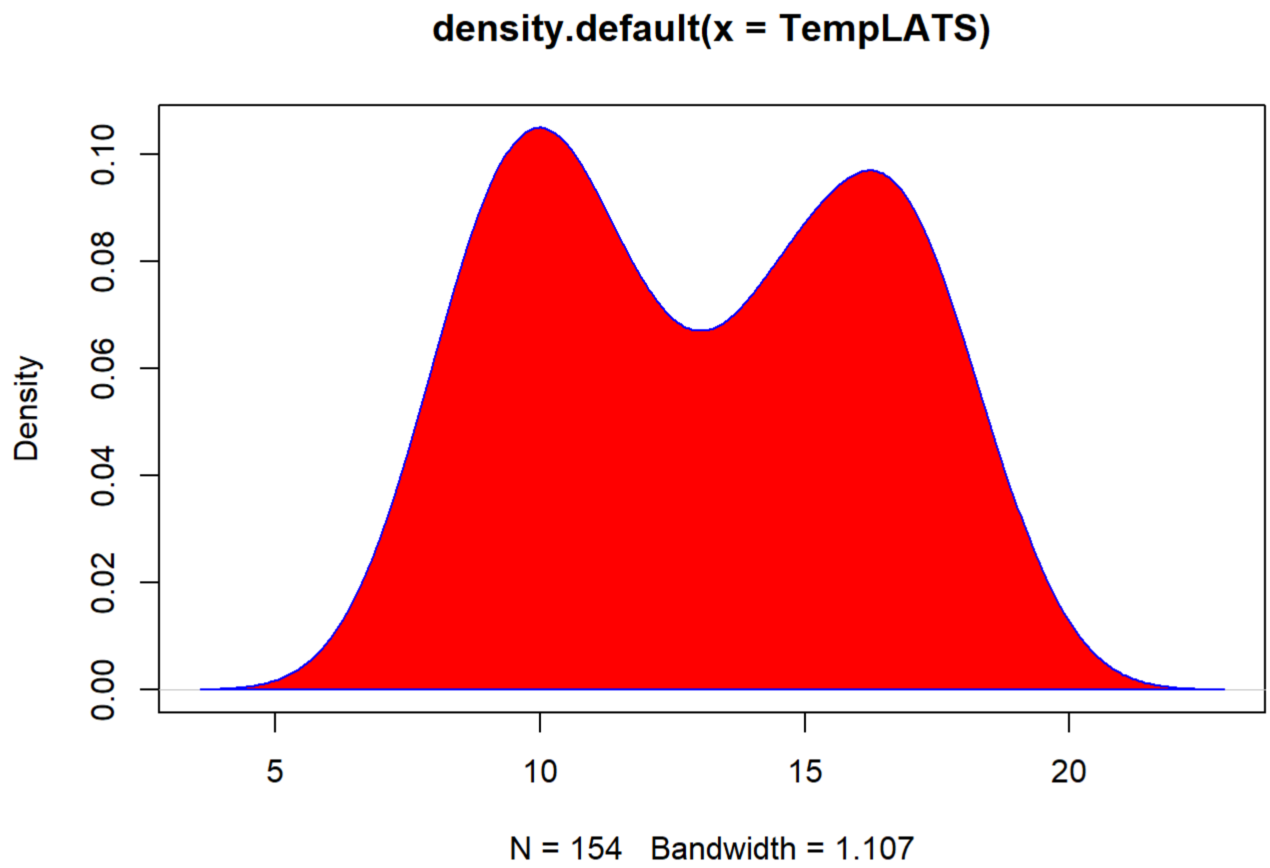



```
acf(TemLA$residuals)
```



Density Plot for LosAngeles

```
TLA<-density(TempLATS)
plot(TLA)
polygon(TLA,col = "red",border="blue")
```



Hypothesis testing

One sample T-test

```
# Ha: mu > 12
t.test(x=Temp$LosAngelesMin,mu=12, alternative="greater")
```

```
##
##  One Sample t-test
##
## data:  Temp$LosAngelesMin
## t = 3.8392, df = 153, p-value = 9.013e-05
## alternative hypothesis: true mean is greater than 12
## 95 percent confidence interval:
##  12.59297      Inf
## sample estimates:
## mean of x
##  13.04221
```

```
# Ha: mu < 12
t.test(Temp$LosAngelesMin, mu=12, alternative="less")
```

```
##
##  One Sample t-test
##
## data:  Temp$LosAngelesMin
## t = 3.8392, df = 153, p-value = 0.9999
## alternative hypothesis: true mean is less than 12
## 95 percent confidence interval:
##      -Inf 13.49145
## sample estimates:
## mean of x
##  13.04221
```

Proportion Test

```
prop.test(x=10, n=154, p = 0.3)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  10 out of 154, null probability 0.3
## X-squared = 39.409, df = 1, p-value = 3.437e-10
## alternative hypothesis: true p is not equal to 0.3
## 95 percent confidence interval:
##  0.03333629 0.11939350
## sample estimates:
##          p
## 0.06493506
```

```
prop.test(x=90, n=154, p=0.5)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  90 out of 154, null probability 0.5
## X-squared = 4.0584, df = 1, p-value = 0.04395
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.5022004 0.6623526
## sample estimates:
##         p
## 0.5844156
```

```
prop.test(x=90, n=154,p=0.5,alternative = "greater")
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  90 out of 154, null probability 0.5
## X-squared = 4.0584, df = 1, p-value = 0.02198
## alternative hypothesis: true p is greater than 0.5
## 95 percent confidence interval:
##  0.5149253 1.0000000
## sample estimates:
##         p
## 0.5844156
```