MA615 Buoy Project

Xiaozhou Lu (Leo)

2020/9/25

1 Introduction

Nowdays, global warming has become a trend that arouse the attention of global concerns. We want to use statistical analysis to figure out the evidence of global warming and try to see how the trend is actually going. Therefore, we collect the data about the temperature both in air and seas in Boston ranging from 1990-2010 and analysis the result. First we import the data from NDBC website, and get it cleaned.Then we use linear regression to fit the model an try to find out if the slope is going up or going down. After that we use time series to draw a temperature charts as an assistance.

### make URLs  
### make URLs  
  
url1 <- "http://www.ndbc.noaa.gov/view\_text\_file.php?filename=mlrf1h"  
url2 <- ".txt.gz&dir=data/historical/stdmet/"  
  
years <- c(1987:2016)  
  
urls <- str\_c(url1, years, url2, sep = "")  
  
filenames <- str\_c("ma", years, sep = "")  
  
### Read the data from the website  
  
N <- length(urls)  
  
for (i in 1:N){  
 suppressMessages( ### This stops the annoying messages on your screen. Do this last.  
 assign(filenames[i], read\_table(urls[i], col\_names = TRUE))  
 )  
   
 file <- get(filenames[i])  
   
 x <- ncol(file)  
  
   
#Since data from different year has different columns, we seperate them and only take the columns we want   
 if(x %in% c(15,16)){  
 colnames(file)[1] <-"YYYY"  
 colnames(file)[2] <-"MM"  
 colnames(file)[3] <-"DD"  
 colnames(file)[4] <-"hh"  
 colnames(file)[12] <-"Air\_tmp"  
 colnames(file)[13] <-"Water\_tmp"  
 file <- file[,c(1,2,3,4,12,13)]  
 }  
 if(x==17){  
 colnames(file)[1] <-"YYYY"  
 colnames(file)[2] <-"MM"  
 colnames(file)[3] <-"DD"  
 colnames(file)[4] <-"hh"  
 colnames(file)[13] <-"Air\_tmp"  
 colnames(file)[14] <-"Water\_tmp"  
 file <- file[,c(1,2,3,4,13,14)]  
 }  
  
#Combine all dataframes  
 if(i == 1){  
 MR <- file  
 }  
  
 else{  
 MR <- rbind.data.frame(MR, file)  
 }  
}

2 Data Cleaning

We get thousands of data imported from the website. And many of them are not useful for our analysis. And we try to reduce the amount of the observations, since they are too many, without leaving out essential information contained. So we try to get daily average temperature and use it to get monthly average temperature. Our model will then be based on monthly average temperature. We decide to build our model monthly, so that we can ignore the influence of season changes on the temperature.

## Clean data

MR\_2 <- as\_tibble(MR)  
#Change type of Air\_tmp and Water\_tmp from chr to dbl  
  
MR\_2 <- MR\_2 %>%  
 mutate(Air\_tmp = as.double(Air\_tmp),  
 Water\_tmp = as.double(Water\_tmp),  
 YYYY\_MM\_DD = ymd(paste(YYYY,MM,DD,sep = "-"))) %>%  
 relocate(YYYY\_MM\_DD)

## Warning: Problem with `mutate()` input `Air\_tmp`.  
## ℹ 强制改变过程中产生了NA  
## ℹ Input `Air\_tmp` is `as.double(Air\_tmp)`.

## Warning in mask$eval\_all\_mutate(dots[[i]]): 强制改变过程中产生了NA

## Warning: Problem with `mutate()` input `Water\_tmp`.  
## ℹ 强制改变过程中产生了NA  
## ℹ Input `Water\_tmp` is `as.double(Water\_tmp)`.

## Warning in mask$eval\_all\_mutate(dots[[i]]): 强制改变过程中产生了NA

## Warning: Problem with `mutate()` input `YYYY\_MM\_DD`.  
## ℹ 10 failed to parse.  
## ℹ Input `YYYY\_MM\_DD` is `ymd(paste(YYYY, MM, DD, sep = "-"))`.

## Warning: 10 failed to parse.

#Get rid of tittle line  
MR\_2 <- filter(MR\_2,hh != "hr")  
  
#Get rid of abnormal data  
MR\_2 <- filter(MR\_2,Air\_tmp < 100, Water\_tmp < 100)  
  
#Get daily average Tmp  
MR\_3\_1 <- select(MR\_2,-c(2,3,4,7)) %>%  
 group\_by(YYYY\_MM\_DD) %>%  
 summarize(Avg\_Air\_tmp = mean(Air\_tmp))

## `summarise()` ungrouping output (override with `.groups` argument)

MR\_3\_2 <- select(MR\_2,-c(2,3,4,6)) %>%  
 group\_by(YYYY\_MM\_DD) %>%  
 summarize(Avg\_Water\_tmp = mean(Water\_tmp))

## `summarise()` ungrouping output (override with `.groups` argument)

#Get monthly average Tmp  
MR\_4\_1 <- MR\_3\_1 %>%  
 group\_by(month = floor\_date(YYYY\_MM\_DD,"month")) %>%  
 summarize(Avg\_Air\_tmp = mean(Avg\_Air\_tmp))

## `summarise()` ungrouping output (override with `.groups` argument)

MR\_4\_2 <- MR\_3\_2 %>%  
 group\_by(month = floor\_date(YYYY\_MM\_DD,"month")) %>%  
 summarize(Avg\_Water\_tmp = mean(Avg\_Water\_tmp))

## `summarise()` ungrouping output (override with `.groups` argument)

MR\_4 <- inner\_join(MR\_4\_1,MR\_4\_2)

## Joining, by = "month"

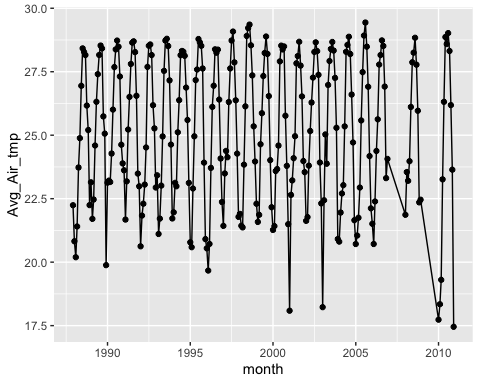
#Export data  
write\_csv(MR\_4,"MR\_data\_1987\_2016.csv")

3 First Interpretation

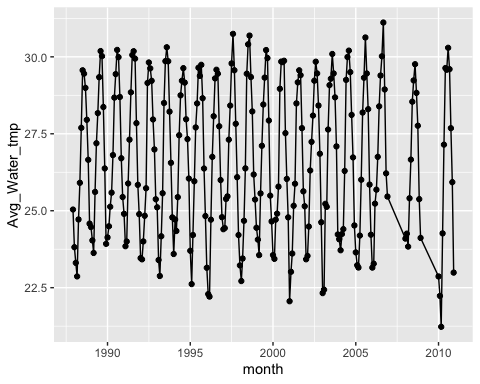
With the data set we need, we first want to have a brief understanding about the condition. So we draw a line chart from both air temperature and water temperature versus the date. And we get the graphs as follows.

##Fit model

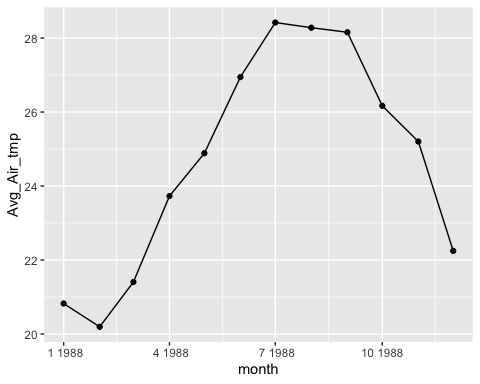
#Plot data  
MR\_4 %>%  
 ggplot(aes(x = month,y = Avg\_Air\_tmp)) +   
 geom\_line() +  
 geom\_point()



MR\_4 %>%  
 ggplot(aes(x = month,y = Avg\_Water\_tmp)) +   
 geom\_line() +  
 geom\_point()



#Check the trend within a year  
MR\_1988 <- filter(MR\_4,year(month)==1988)  
MR\_1988 %>%  
 ggplot(aes(x = month, y = Avg\_Air\_tmp)) +  
 geom\_line() +  
 geom\_point()



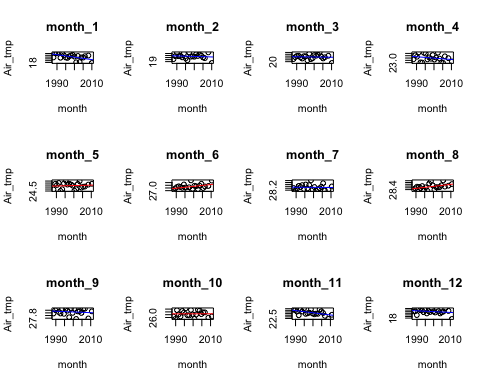
From the chart above, we get a normal feeling how the temperature is supposed to change seasonally. And we can see a slight hint that the temperature is actuallly going higher. But these are just our first impression in our eyes. We need more rigorous statistcal analysis to make our conclusion more convincing.

4 Linear Regression

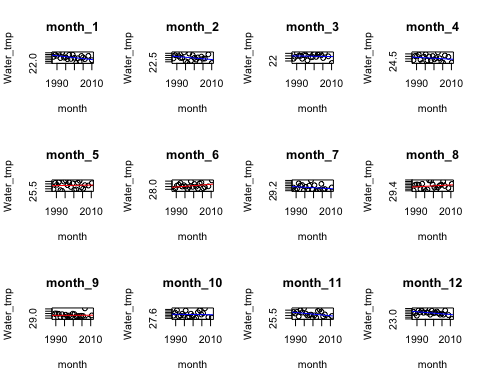
We now use linear regression to find a fit between temperature and date, then we get to analysis the coefficient to see whether their trend is going upwards or downwards.

###Fit time series model

#Red indicates an upward trend  
#Blue indicates an downward trend  
  
#plot air\_tmp  
par(mfrow=c(3,4))  
for(i in 1:12){  
 LM\_MR\_4\_1 <- filter(MR\_4\_1,month(month)==i)  
 LM\_1 <- lm(Avg\_Air\_tmp~month,data = LM\_MR\_4\_1)  
  
 if(coef(LM\_1)[2]>0){  
 plot(x = LM\_MR\_4\_1$month, y = LM\_MR\_4\_1$Avg\_Air\_tmp,xlab = "month", ylab = "Air\_tmp",main = paste("month",i,sep = "\_"))  
 abline(coef(LM\_1)[1],coef(LM\_1)[2], col="RED")  
   
 }  
   
 if(coef(LM\_1)[2]<0){  
 plot(x = LM\_MR\_4\_1$month, y = LM\_MR\_4\_1$Avg\_Air\_tmp,xlab = "month", ylab = "Air\_tmp",main = paste("month",i,sep = "\_"))  
 abline(coef(LM\_1)[1],coef(LM\_1)[2], col="BLUE")  
   
 }  
}



#plot water\_tmp  
par(mfrow=c(3,4))  
for(i in 1:12){  
 LM\_MR\_4\_2 <- filter(MR\_4\_2,month(month)==i)  
 LM\_2 <- lm(Avg\_Water\_tmp~month,data = LM\_MR\_4\_2)  
  
 if(coef(LM\_2)[2]>0){  
 plot(x = LM\_MR\_4\_2$month, y = LM\_MR\_4\_2$Avg\_Water\_tmp,xlab = "month", ylab = "Water\_tmp",main = paste("month",i,sep = "\_"))  
 abline(coef(LM\_2)[1],coef(LM\_2)[2], col="RED")  
   
 }  
   
 if(coef(LM\_2)[2]<0){  
 plot(x = LM\_MR\_4\_2$month, y = LM\_MR\_4\_2$Avg\_Water\_tmp,xlab = "month", ylab = "Water\_tmp",main = paste("month",i,sep = "\_"))  
 abline(coef(LM\_2)[1],coef(LM\_2)[2], col="BLUE")  
   
 }  
}



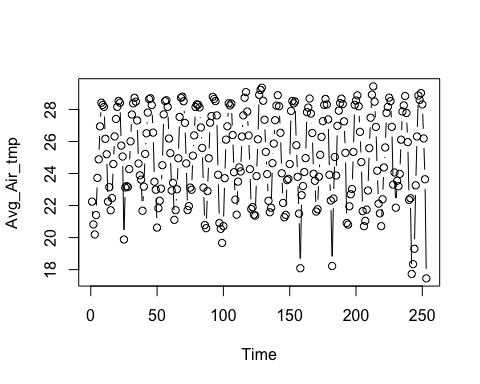
From the output above, for air temperature versus date, the coefficient is negative in 8 months and the positive in 4 months. Besides, for water temperature versus date, the coefficient is negative in 8 months and positive in 4 months too. So we have a reason to predict that the overall temperature trend is going downwards.

5 Time Series

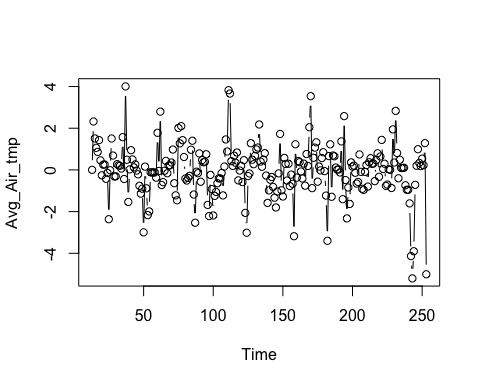
We use ARIMA model in time series as a support to our analysis. The graphs are shown as follows.

###Fit lm model

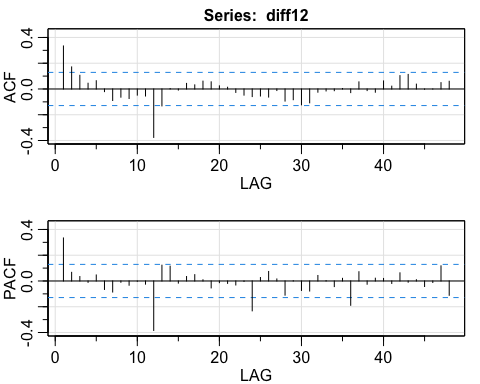
#create TS  
TS\_MR\_4\_1 <- ts(select(MR\_4\_1,Avg\_Air\_tmp))  
  
#check trend  
plot(TS\_MR\_4\_1,type="b")



#set seasonal index  
diff12 = diff(TS\_MR\_4\_1,12)  
plot(diff12,type="b")



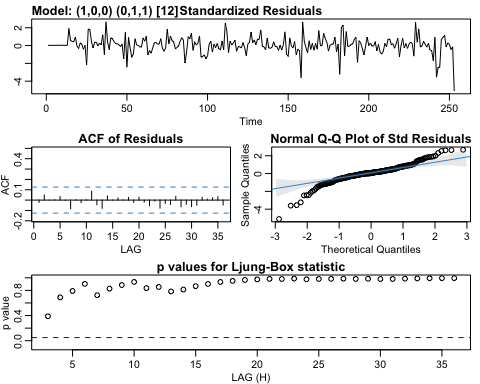
#check acf pacf  
acf2(diff12,48)



## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]  
## ACF 0.33 0.17 0.11 0.04 0.06 -0.02 -0.09 -0.06 -0.07 -0.05 -0.05 -0.38 -0.13  
## PACF 0.33 0.07 0.03 -0.01 0.05 -0.07 -0.09 -0.01 -0.03 0.00 -0.02 -0.38 0.12  
## [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]  
## ACF 0.00 -0.01 0.04 0.03 0.06 0.06 0.02 0.01 -0.03 -0.05 -0.06 -0.05  
## PACF 0.11 -0.02 0.03 0.05 0.01 -0.05 -0.01 -0.02 -0.03 0.00 -0.23 0.03  
## [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]  
## ACF -0.06 -0.01 -0.09 -0.08 -0.12 -0.11 -0.02 -0.02 -0.01 0.00 -0.03 0.06  
## PACF 0.07 0.02 -0.11 0.00 -0.07 -0.08 0.04 0.00 -0.04 0.02 -0.19 0.07  
## [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]  
## ACF -0.01 -0.03 0.06 0.02 0.10 0.11 0.04 0.00 0.00 0.05 0.06  
## PACF -0.03 0.02 0.02 -0.02 0.06 -0.01 0.01 -0.04 -0.01 0.12 -0.11

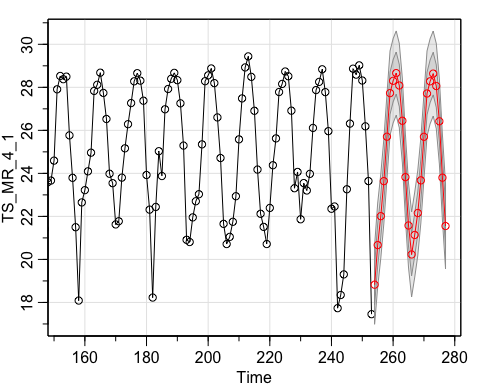
#fit model  
sarima(TS\_MR\_4\_1,1,0,0,0,1,1,12)

## initial value 0.243137   
## iter 2 value 0.056952  
## iter 3 value 0.018759  
## iter 4 value -0.041199  
## iter 5 value -0.043005  
## iter 6 value -0.043077  
## iter 7 value -0.043197  
## iter 8 value -0.043204  
## iter 9 value -0.043204  
## iter 10 value -0.043204  
## iter 11 value -0.043204  
## iter 11 value -0.043204  
## iter 11 value -0.043204  
## final value -0.043204   
## converged  
## initial value -0.044892   
## iter 2 value -0.047348  
## iter 3 value -0.047801  
## iter 4 value -0.048541  
## iter 5 value -0.048542  
## iter 6 value -0.048542  
## iter 7 value -0.048542  
## iter 7 value -0.048542  
## iter 7 value -0.048542  
## final value -0.048542   
## converged



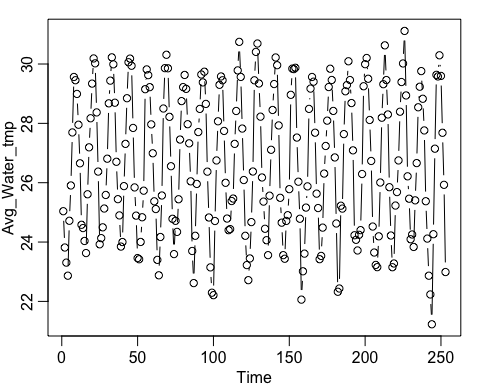
## $fit  
##   
## Call:  
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,   
## Q), period = S), xreg = constant, transform.pars = trans, fixed = fixed,   
## optim.control = list(trace = trc, REPORT = 1, reltol = tol))  
##   
## Coefficients:  
## ar1 sma1 constant  
## 0.3447 -0.8686 -0.0020  
## s.e. 0.0643 0.0462 0.0015  
##   
## sigma^2 estimated as 0.8459: log likelihood = -330.27, aic = 668.53  
##   
## $degrees\_of\_freedom  
## [1] 238  
##   
## $ttable  
## Estimate SE t.value p.value  
## ar1 0.3447 0.0643 5.3623 0.0000  
## sma1 -0.8686 0.0462 -18.8105 0.0000  
## constant -0.0020 0.0015 -1.3184 0.1886  
##   
## $AIC  
## [1] 2.652902  
##   
## $AICc  
## [1] 2.653286  
##   
## $BIC  
## [1] 2.708216

#predict  
sarima.for(TS\_MR\_4\_1,24,1,0,0,0,1,1,12)

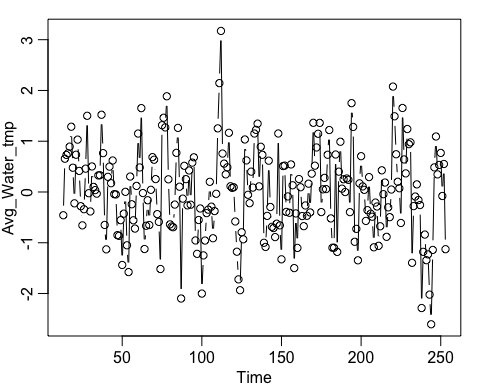


## $pred  
## Time Series:  
## Start = 254   
## End = 277   
## Frequency = 1   
## [1] 18.82520 20.66768 22.01178 23.64041 25.70421 27.73084 28.30159 28.66092  
## [9] 28.08484 26.44531 23.82433 21.57744 20.23048 21.13607 22.15724 23.67456  
## [17] 25.70000 27.71340 28.27959 28.63735 28.06073 26.42102 23.79997 21.55306  
##   
## $se  
## Time Series:  
## Start = 254   
## End = 277   
## Frequency = 1   
## [1] 0.9200156 0.9731337 0.9792530 0.9799774 0.9800635 0.9800737 0.9800749  
## [8] 0.9800750 0.9800749 0.9800739 0.9800656 0.9799952 0.9874917 0.9883785  
## [15] 0.9884838 0.9884964 0.9884978 0.9884980 0.9884980 0.9884980 0.9884979  
## [22] 0.9884969 0.9884886 0.9884189

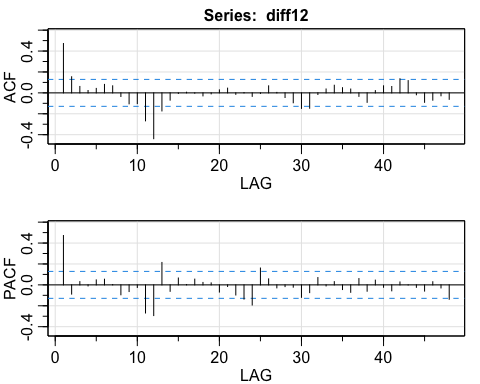
#repeat process for water\_tmp  
TS\_MR\_4\_2 <- ts(select(MR\_4\_2,Avg\_Water\_tmp))  
plot(TS\_MR\_4\_2,type="b")



diff12 = diff(TS\_MR\_4\_2,12)  
plot(diff12,type="b")



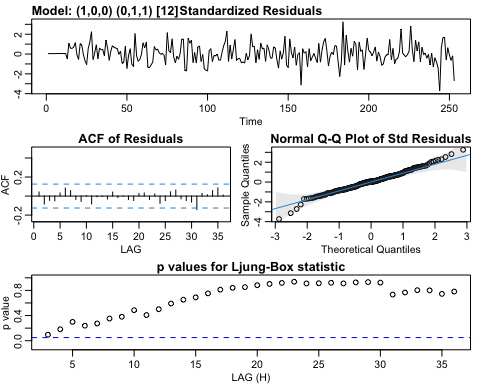
acf2(diff12,48)



## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]  
## ACF 0.47 0.15 0.06 0.02 0.04 0.08 0.07 -0.03 -0.10 -0.10 -0.27 -0.44 -0.17  
## PACF 0.47 -0.09 0.03 -0.01 0.05 0.05 0.00 -0.09 -0.06 -0.02 -0.27 -0.29 0.21  
## [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]  
## ACF -0.07 -0.01 0.01 0.00 -0.03 -0.01 0.03 0.05 -0.01 0.00 -0.03 -0.01  
## PACF -0.06 0.07 0.00 0.05 0.02 0.02 -0.07 -0.02 -0.10 -0.14 -0.19 0.16  
## [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]  
## ACF 0.07 0.00 -0.04 -0.09 -0.15 -0.15 -0.02 0.04 0.07 0.05 0.04 -0.03  
## PACF 0.06 -0.03 -0.02 -0.02 -0.12 -0.07 0.07 -0.01 0.03 -0.04 -0.07 0.06  
## [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]  
## ACF -0.09 0.02 0.07 0.06 0.13 0.12 -0.02 -0.09 -0.07 -0.03 -0.06  
## PACF -0.06 0.05 -0.02 -0.06 0.03 0.00 -0.02 -0.06 0.03 -0.03 -0.14

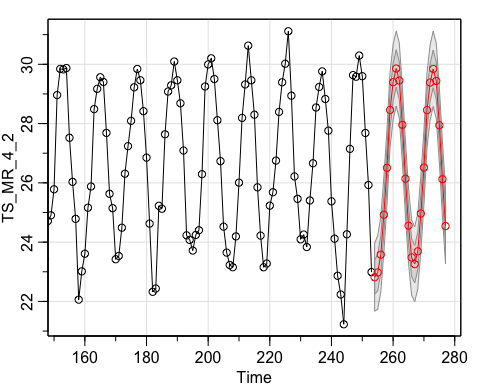
sarima(TS\_MR\_4\_2,1,0,0,0,1,1,12)

## initial value -0.128349   
## iter 2 value -0.405534  
## iter 3 value -0.456947  
## iter 4 value -0.483567  
## iter 5 value -0.487400  
## iter 6 value -0.487470  
## iter 7 value -0.487606  
## iter 8 value -0.487642  
## iter 9 value -0.487642  
## iter 10 value -0.487642  
## iter 10 value -0.487642  
## iter 10 value -0.487642  
## final value -0.487642   
## converged  
## initial value -0.493820   
## iter 2 value -0.499807  
## iter 3 value -0.505435  
## iter 4 value -0.505713  
## iter 5 value -0.505948  
## iter 6 value -0.505961  
## iter 7 value -0.505970  
## iter 8 value -0.505972  
## iter 9 value -0.505972  
## iter 10 value -0.505972  
## iter 11 value -0.505973  
## iter 12 value -0.505973  
## iter 13 value -0.505973  
## iter 13 value -0.505973  
## iter 13 value -0.505973  
## final value -0.505973   
## converged



## $fit  
##   
## Call:  
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,   
## Q), period = S), xreg = constant, transform.pars = trans, fixed = fixed,   
## optim.control = list(trace = trc, REPORT = 1, reltol = tol))  
##   
## Coefficients:  
## ar1 sma1 constant  
## 0.4323 -0.9952 -0.0012  
## s.e. 0.0589 0.8520 0.0008  
##   
## sigma^2 estimated as 0.3135: log likelihood = -220.02, aic = 448.05  
##   
## $degrees\_of\_freedom  
## [1] 238  
##   
## $ttable  
## Estimate SE t.value p.value  
## ar1 0.4323 0.0589 7.3364 0.0000  
## sma1 -0.9952 0.8520 -1.1681 0.2439  
## constant -0.0012 0.0008 -1.4709 0.1426  
##   
## $AIC  
## [1] 1.777974  
##   
## $AICc  
## [1] 1.778358  
##   
## $BIC  
## [1] 1.833288

sarima.for(TS\_MR\_4\_2,24,1,0,0,0,1,1,12)



## $pred  
## Time Series:  
## Start = 254   
## End = 277   
## Frequency = 1   
## [1] 22.81450 22.97754 23.57772 24.92438 26.50741 28.45900 29.39443 29.85687  
## [9] 29.45273 27.95678 26.13833 24.55676 23.48347 23.25827 23.69062 24.96474  
## [17] 26.51640 28.45444 29.38401 29.84391 29.43868 27.94225 26.12360 24.54194  
##   
## $se  
## Time Series:  
## Start = 254   
## End = 277   
## Frequency = 1   
## [1] 0.5717835 0.6229199 0.6320169 0.6337023 0.6340167 0.6340754 0.6340863  
## [8] 0.6340877 0.6340846 0.6340662 0.6339670 0.6334360 0.6339737 0.6340741  
## [15] 0.6340929 0.6340964 0.6340971 0.6340972 0.6340971 0.6340964 0.6340930  
## [22] 0.6340744 0.6339753 0.6334442

6 Conclusion

From the statistical analysis above, we can predict that the overall temperature trend in Boston is going downwards, which may have a little conflict with our assumption at first. But the data has its own limitations. First, the data collected is only from Boston area, it cannot represent the trend in global. Besides there are many other factors that may contribute the result, such as ocean current. Even though the result here doesn’t cater to the public belief that the world is become warmer and warmer, we still have to be respectful to our nature and live a environmental friendly life.