Buoy project

ziyi bai

2020/9/25

Introduction: What am I supposed to do in this project?

< p style="font-size:20px">

Buoy is a floating device that can be used for navigation, marker, rescue, research, military purposes. In this project, we will use 20 years weather data collected by one weather buoy near Boston to determine whether these data show us the increasing temperature pattern, which can be known as the problem of global warming. If my model shows me an increasing trend in temperature, then global warming might be an issue that encourage our concern.

Main Body: describe my approach of analysing

< p style="font-size:20px">

The first step in my project is to collect and reorganize data. We collected data from 1987 to 2016, one of my group mate, zijie huang, was responsible for organizing those data. We calculated the average value of ATMP and WTMP for each month and combine them into a new graph, MR_4.

```
## clean data
setwd("/Users/baiziyi/Desktop/MSSP/MA615/buoy/buoy-project")
MR <- read.csv(file="Buoy_MR.csv", header = TRUE)
MR_2 <- as_tibble(MR)
#Change type of Air_tmp and Water_tmp from chr to dbl
MR_2 <- MR_2 %>%
  mutate(ATMP = as.double(ATMP),
         WTMP = as.double(WTMP),
         YYYY_MM_DD = ymd(paste(YYYY,MM,DD,sep = "-"))) %>%
  relocate(YYYY_MM_DD)

#Get rid of tittle line
MR_2 <- filter(MR_2,hh != "hr")

#Get rid of abnormal data
MR_2 <- filter(MR_2,ATMP < 100, WTMP < 100)

#Get daily average Tmp
MR_3_1 <- select(MR_2,-c(2,3,4,7)) %>%
  group_by(YYYY_MM_DD) %>%
  summarize(Avg_ATMP = mean(ATMP))
```

```
## `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_WTMP = mean(WTMP))
```

```
## `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_ATMP = mean(Avg_ATMP))
```

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

```
MR_4_2 <- MR_3_2 %>%
  group_by(month = floor_date(YYYY_MM_DD,"month")) %>%
  summarize(Avg_WTMP = mean(Avg_WTMP))
```

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

```
MR_4 <- inner_join(MR_4_1,MR_4_2)
```

```
## Joining, by = "month"
```

```
print(MR_4)
```

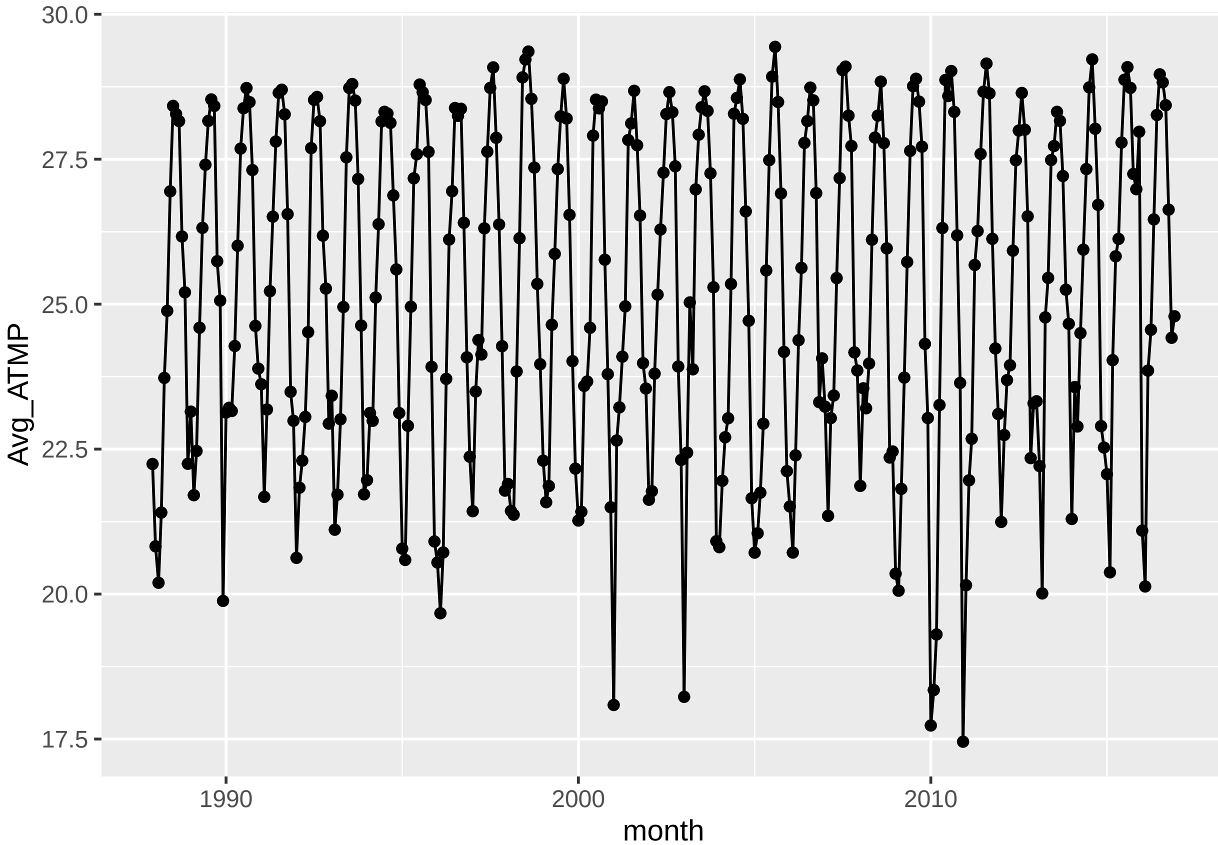
```
## # A tibble: 349 x 3
##    month      Avg_ATMP Avg_WTMP
##    <date>        <dbl>    <dbl>
##  1 1987-12-01     22.2     25.0
##  2 1988-01-01     20.8     23.8
##  3 1988-02-01     20.2     23.3
##  4 1988-03-01     21.4     22.9
##  5 1988-04-01     23.7     24.7
##  6 1988-05-01     24.9     25.9
##  7 1988-06-01     26.9     27.7
##  8 1988-07-01     28.4     29.6
##  9 1988-08-01     28.3     29.5
## 10 1988-09-01     28.2     29.0
## # ... with 339 more rows
```

```
#Export data
write_csv(MR_4,"MR_data_1987_2016.csv")
```

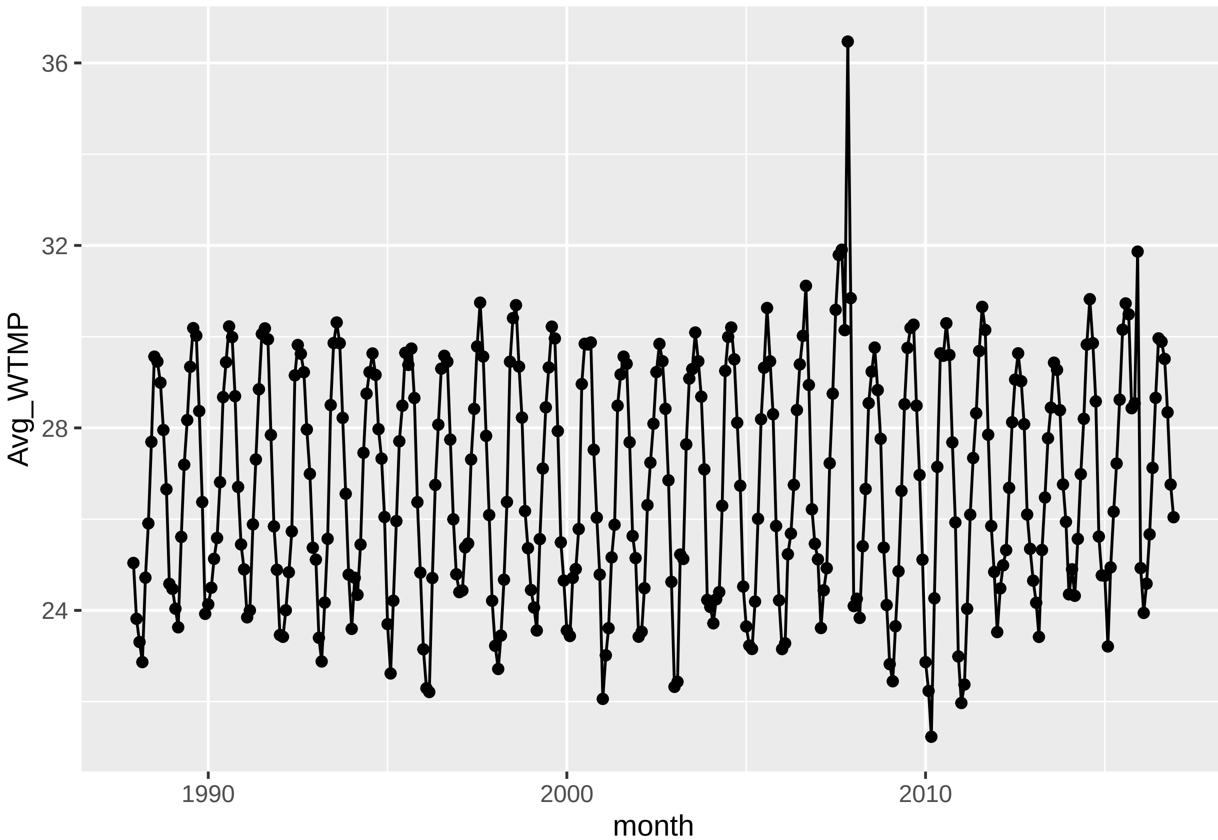
< p style=“font-size:20px”>

Then, I plotted the data. I plotted the average ATMP and WTMP as y-axis, use month as x-axis using ggplot. It seems that the pattern is messy and it is hard for us to clarify the trend of temperature. Also, we can see that for one year temperature trend, the highest temperature is in July and the lowest is in February.

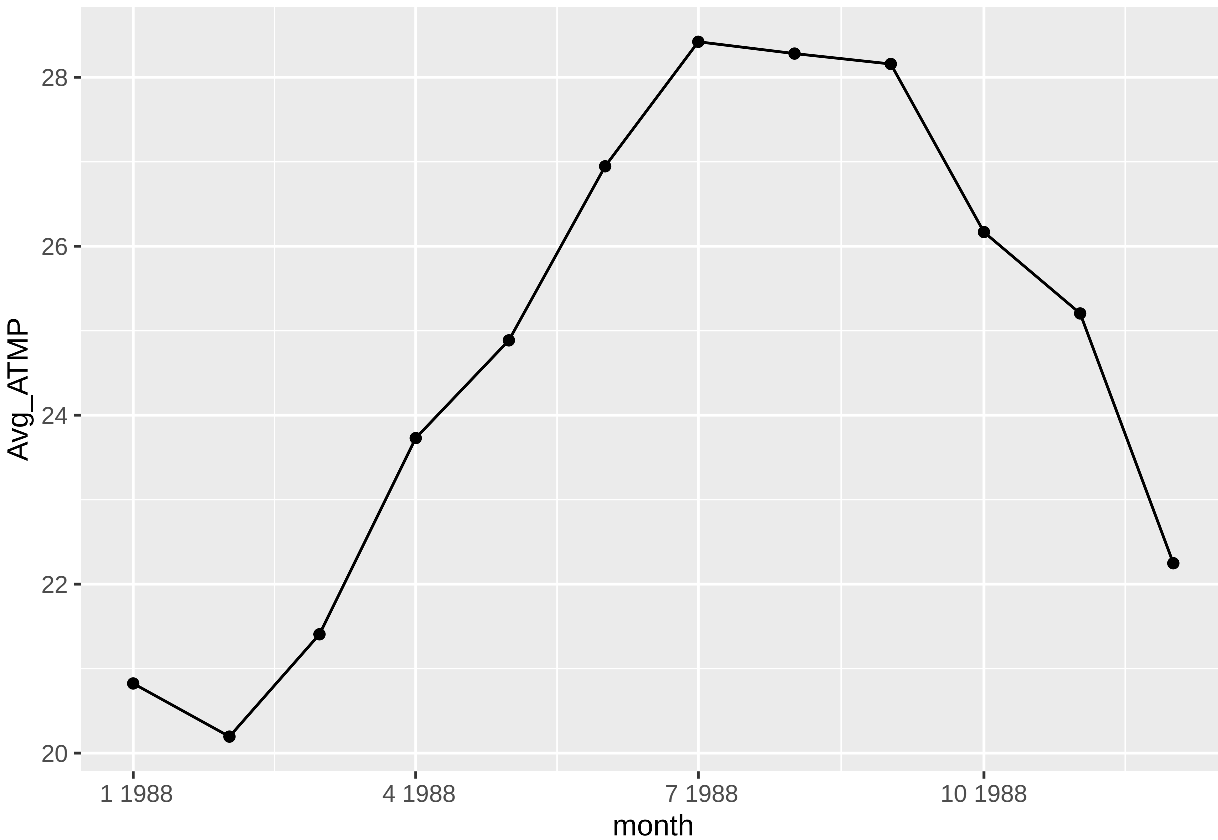
```
##Fit model
#Plot data
MR_4 %>%
  ggplot(aes(x = month,y = Avg_ATMP)) +
  geom_line() +
  geom_point()
```




```r
MR_4 %>%
  ggplot(aes(x = month,y = Avg_WTMP)) +
  geom_line() +
  geom_point()
```



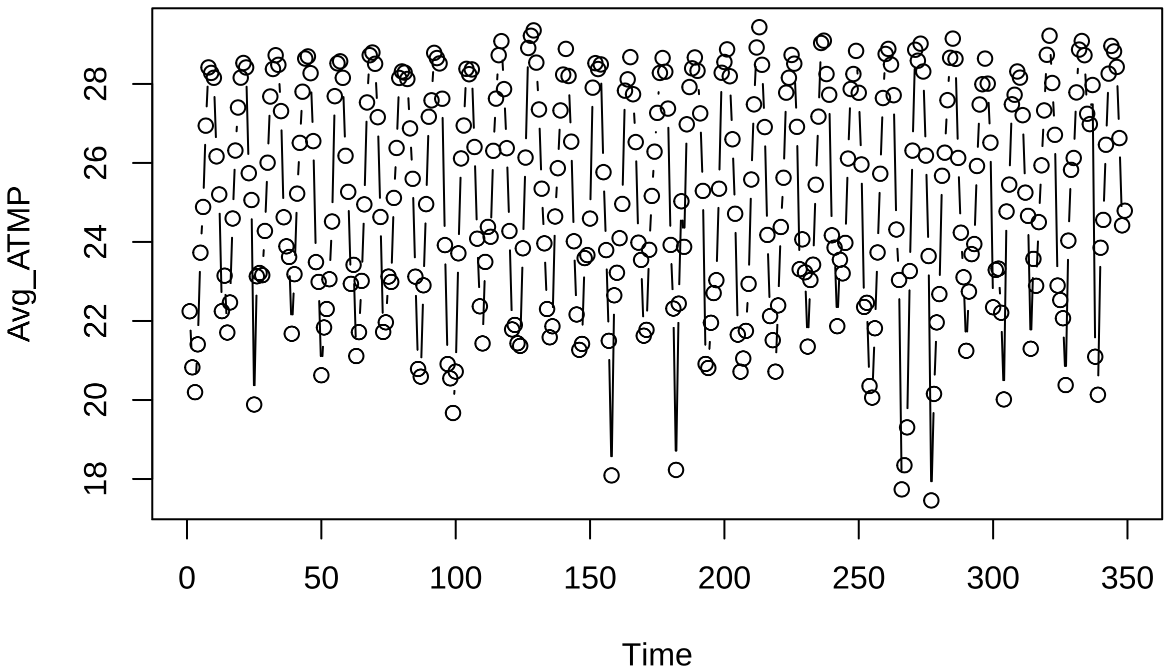

```r
#Check the trend within a year
MR_1988 <- filter(MR_4,year(month)==1988)
MR_1988 %>%
  ggplot(aes(x = month, y = Avg_ATMP)) +
  geom_line() +
  geom_point()
```



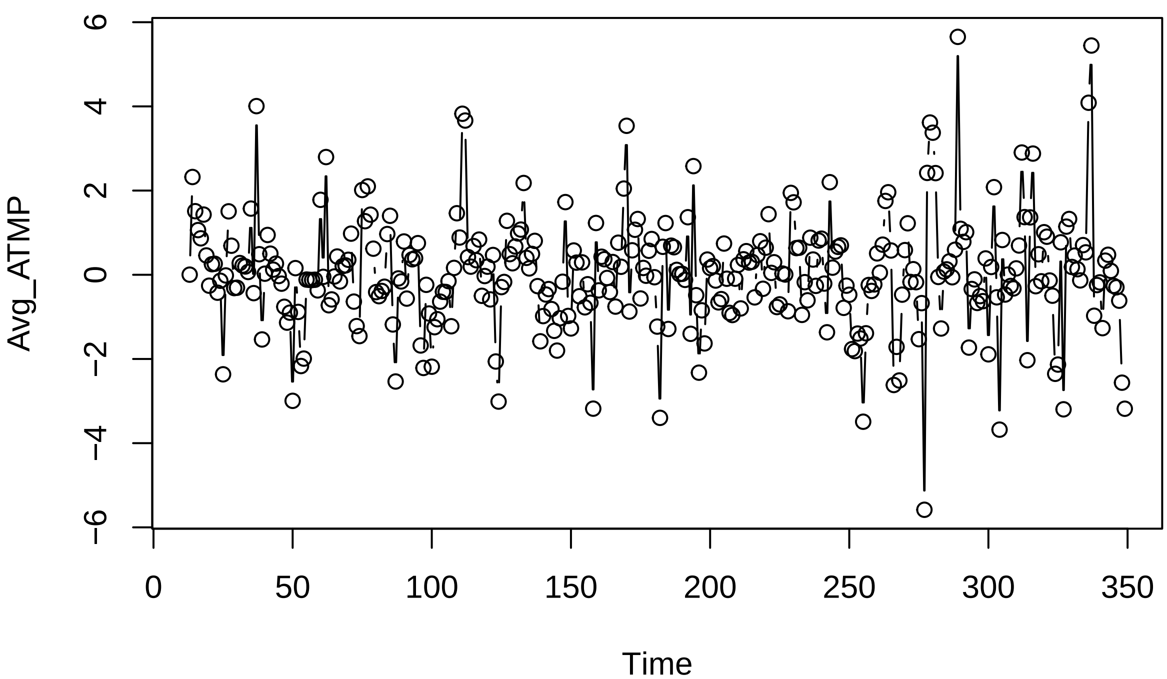
< p style="font-size:20px">

Cause these are years long data, so we suppose time series might be helpful. Still, this method just indicated little information about global warming.

```
###Fit time series model
#create TS
TS_MR_4_1 <- ts(select(MR_4_1,Avg_ATMP))

#check trend
plot(TS_MR_4_1,type="b")
```



```
#set seasonal index
diff12 = diff(TS_MR_4_1,12)
plot(diff12,type="b")
```

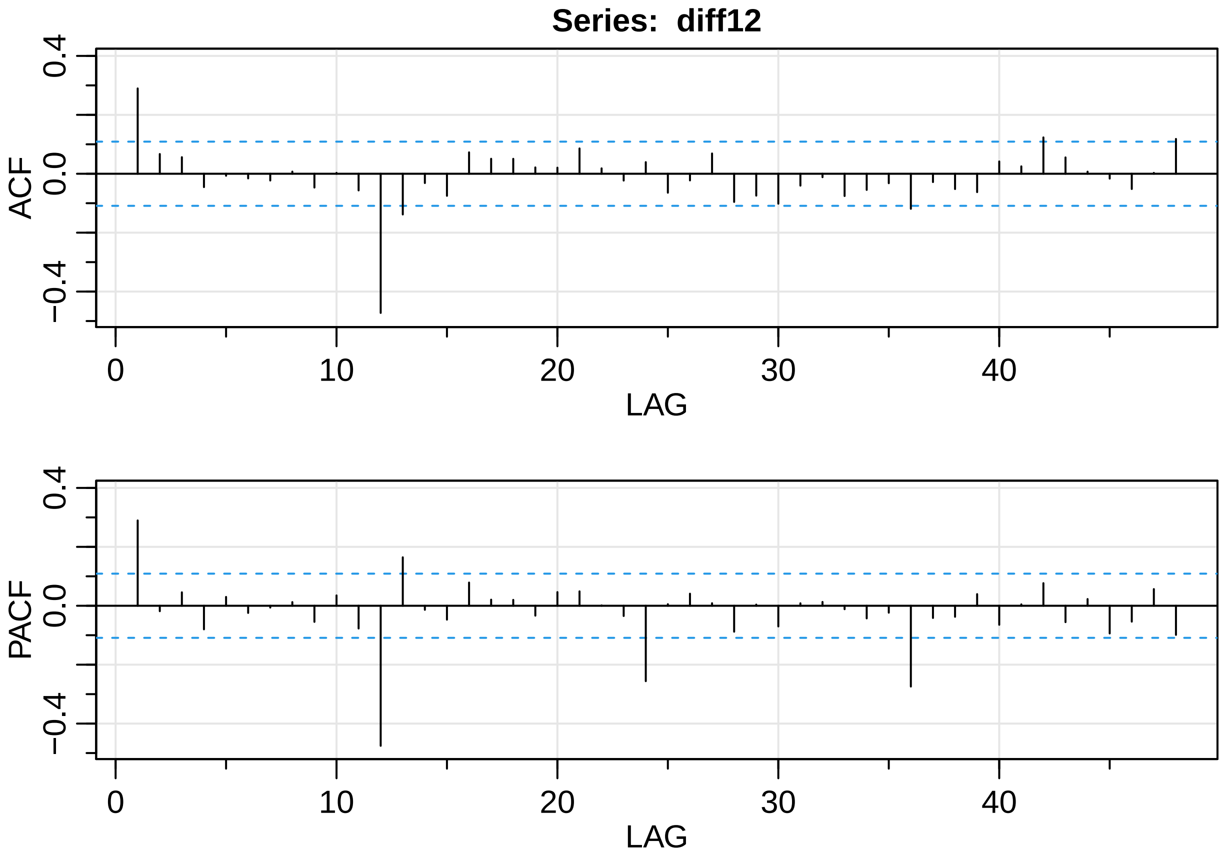


```
#check acf pacf
install.packages('astsa', repos = "http://cran.us.r-project.org")
```

```
## Warning: unable to access index for repository http://cran.us.r-project.org/src/contrib:
##   download from 'http://cran.us.r-project.org/src/contrib/PACKAGES' failed

## Warning: package 'astsa' is not available (for R version 4.0.2)

## Warning: unable to access index for repository http://cran.us.r-project.org/bin/macosx/contrib/4.0:
```

```
##   download from 'http://cran.us.r-project.org/bin/macosx/contrib/4.0/PACKAGES' failed
```

```r
library(astsa)
acf2(diff12,48)
```

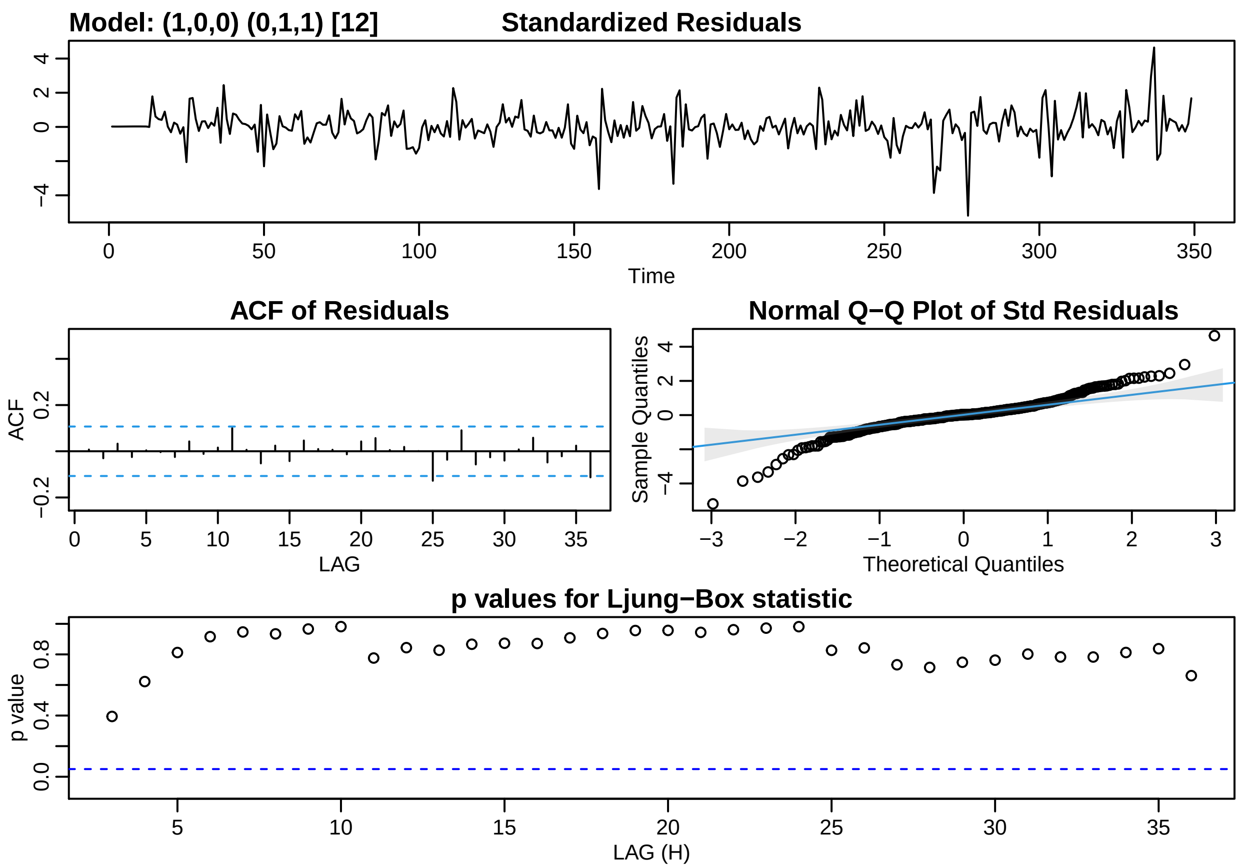


```
##      [,1]  [,2] [,3]  [,4]  [,5]  [,6]  [,7] [,8]  [,9] [,10] [,11] [,12] [,13]
## ACF  0.29  0.07 0.06 -0.05 -0.01 -0.02 -0.02 0.01 -0.05  0.00 -0.06 -0.47 -0.14
## PACF 0.29 -0.02 0.05 -0.08  0.03 -0.02 -0.01 0.01 -0.05  0.04 -0.08 -0.48  0.16
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF  -0.03 -0.08  0.07  0.05  0.05  0.02  0.02  0.09  0.02 -0.02  0.04 -0.06
## PACF -0.01 -0.05  0.08  0.02  0.02 -0.03  0.05  0.05  0.00 -0.03 -0.26  0.01
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF  -0.02  0.07 -0.10 -0.07 -0.10 -0.04 -0.01 -0.08 -0.05 -0.03 -0.12 -0.03
## PACF  0.04  0.01 -0.09  0.00 -0.07  0.01  0.01 -0.01 -0.04 -0.02 -0.27 -0.04
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF  -0.05 -0.06  0.04  0.03  0.12  0.06  0.01 -0.02 -0.05  0.00  0.12
## PACF -0.04  0.04 -0.06  0.00  0.08 -0.06  0.02 -0.09 -0.05  0.06 -0.10
```

```r
#fit model
sarima(TS_MR_4_1,1,0,0,0,1,1,12)
```

```
## initial  value 0.297411
## iter   2 value 0.073932
## iter   3 value 0.039620
## iter   4 value 0.003132
## iter   5 value -0.006321
## iter   6 value -0.007099
## iter   7 value -0.007488
## iter   8 value -0.007528
```

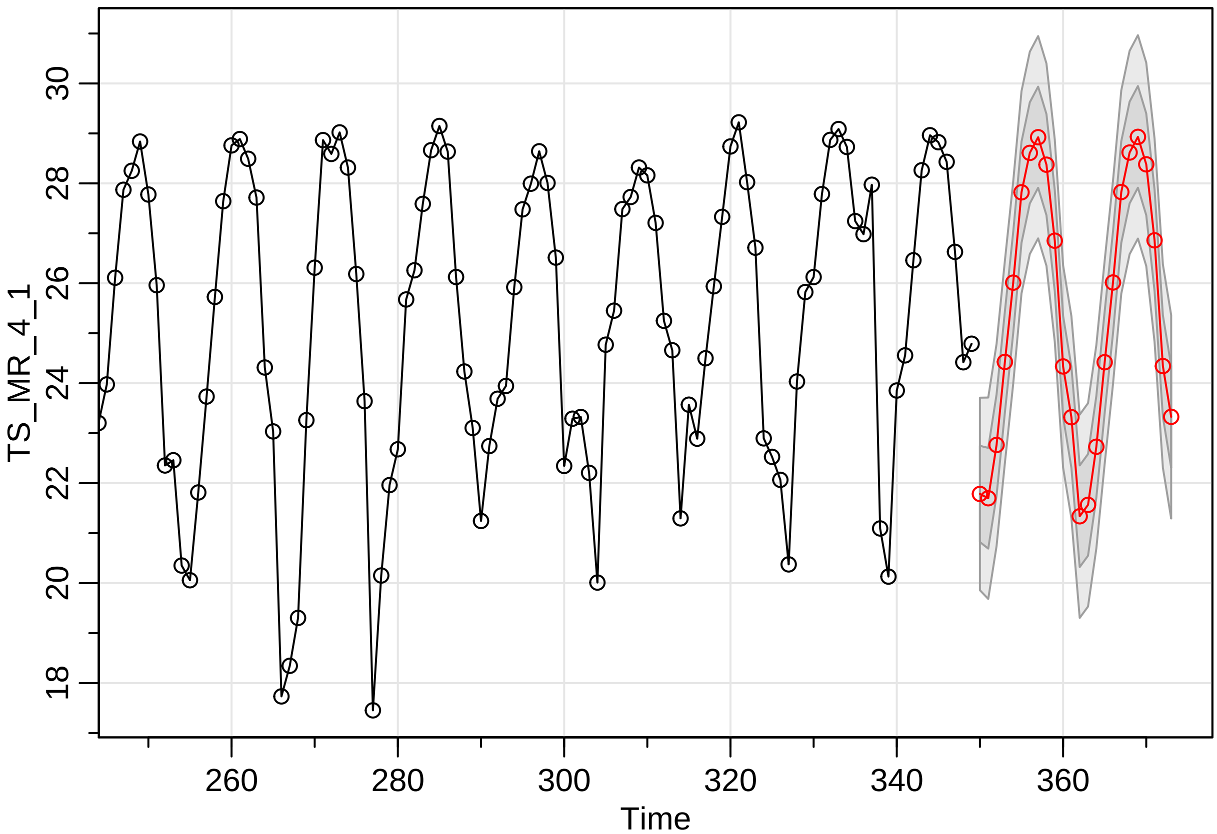
```
## iter   9 value -0.007531
## iter  10 value -0.007532
## iter  11 value -0.007532
## iter  11 value -0.007532
## iter  11 value -0.007532
## final  value -0.007532
## converged
## initial  value -0.006416
## iter   2 value -0.007419
## iter   3 value -0.007936
## iter   4 value -0.008382
## iter   5 value -0.008391
## iter   6 value -0.008393
## iter   7 value -0.008393
## iter   7 value -0.008393
## iter   7 value -0.008393
## final  value -0.008393
## 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
##       0.3072  -0.8973     6e-04
## s.e.  0.0522   0.0392     1e-03
##
## sigma^2 estimated as 0.9275:  log likelihood = -475.35,  aic = 958.71
##
## $degrees_of_freedom
## [1] 334
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        0.3072 0.0522   5.8818  0.0000
## sma1      -0.8973 0.0392 -22.9165  0.0000
## constant   0.0006 0.0010   0.6112  0.5415
##
## $AIC
## [1] 2.754908
##
## $AICc
## [1] 2.755108
##
## $BIC
## [1] 2.798817
```

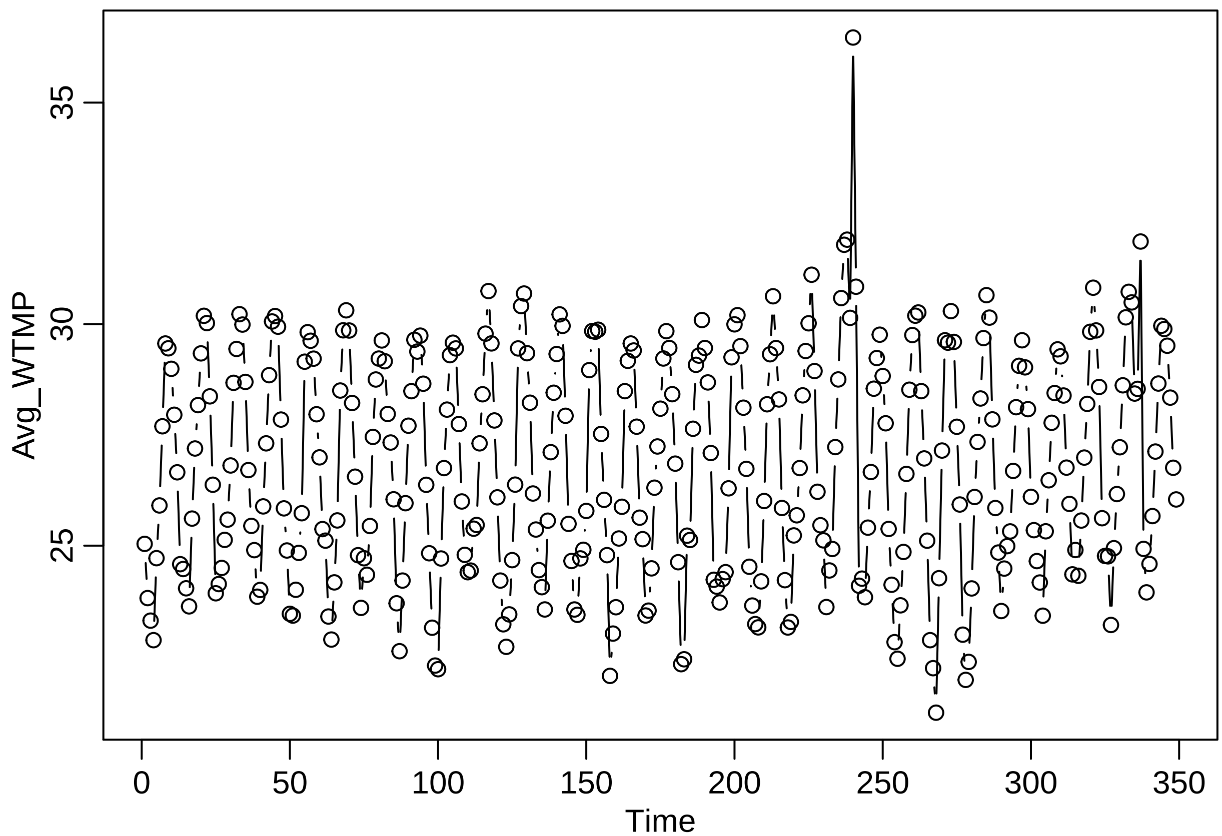
```
#predict
sarima.for(TS_MR_4_1,24,1,0,0,0,1,1,12)
```



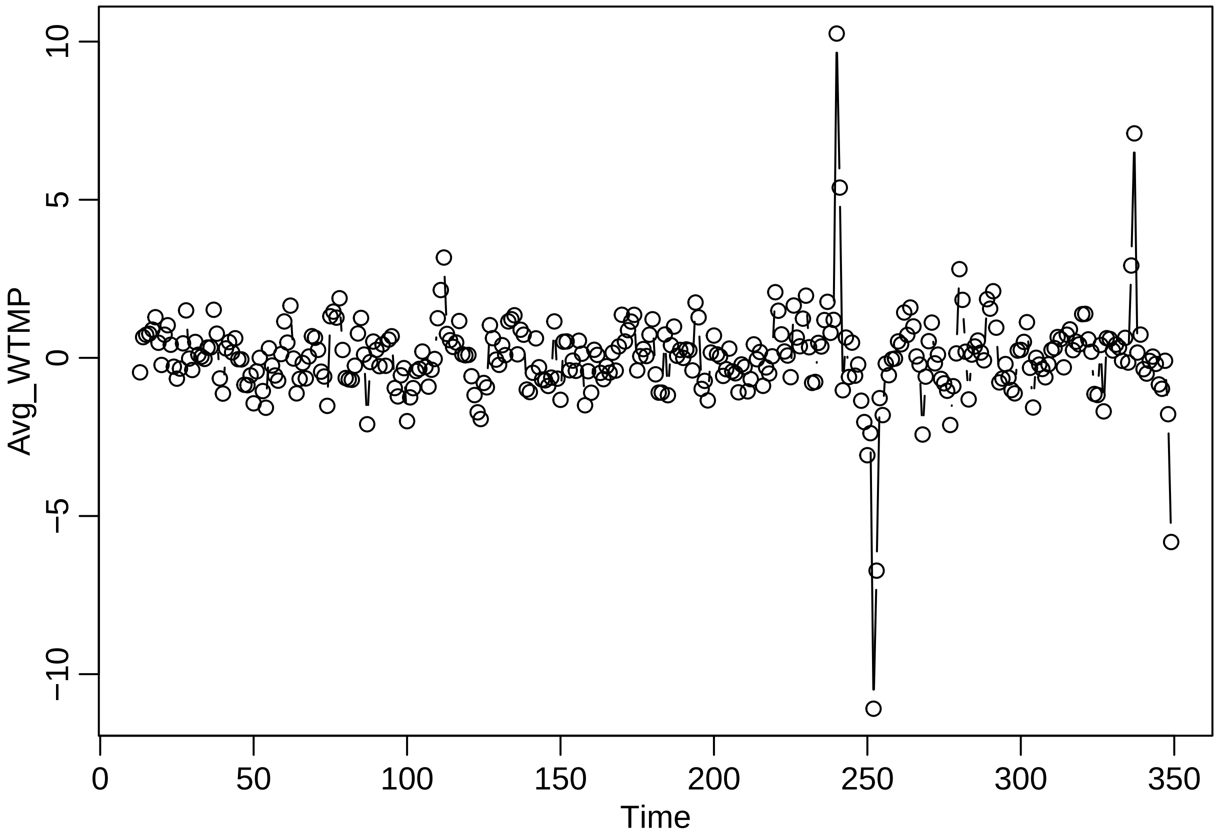

```
## $pred
```

```
## Time Series:
## Start = 350
## End = 373
## Frequency = 1
##  [1] 21.78475 21.69883 22.76530 24.42700 26.01375 27.82167 28.61075 28.92420
##  [9] 28.37455 26.85322 24.33688 23.32103 21.33811 21.56664 22.72972 24.42110
## [17] 26.01697 27.82770 28.61763 28.93135 28.38178 26.86048 24.34414 23.32829
##
## $se
## Time Series:
## Start = 350
## End = 373
## Frequency = 1
##  [1] 0.963251 1.007682 1.011775 1.012160 1.012197 1.012200 1.012200 1.012200
##  [9] 1.012200 1.012200 1.012197 1.012165 1.017020 1.017477 1.017520 1.017524
## [17] 1.017525 1.017525 1.017525 1.017525 1.017525 1.017524 1.017521 1.017489
```

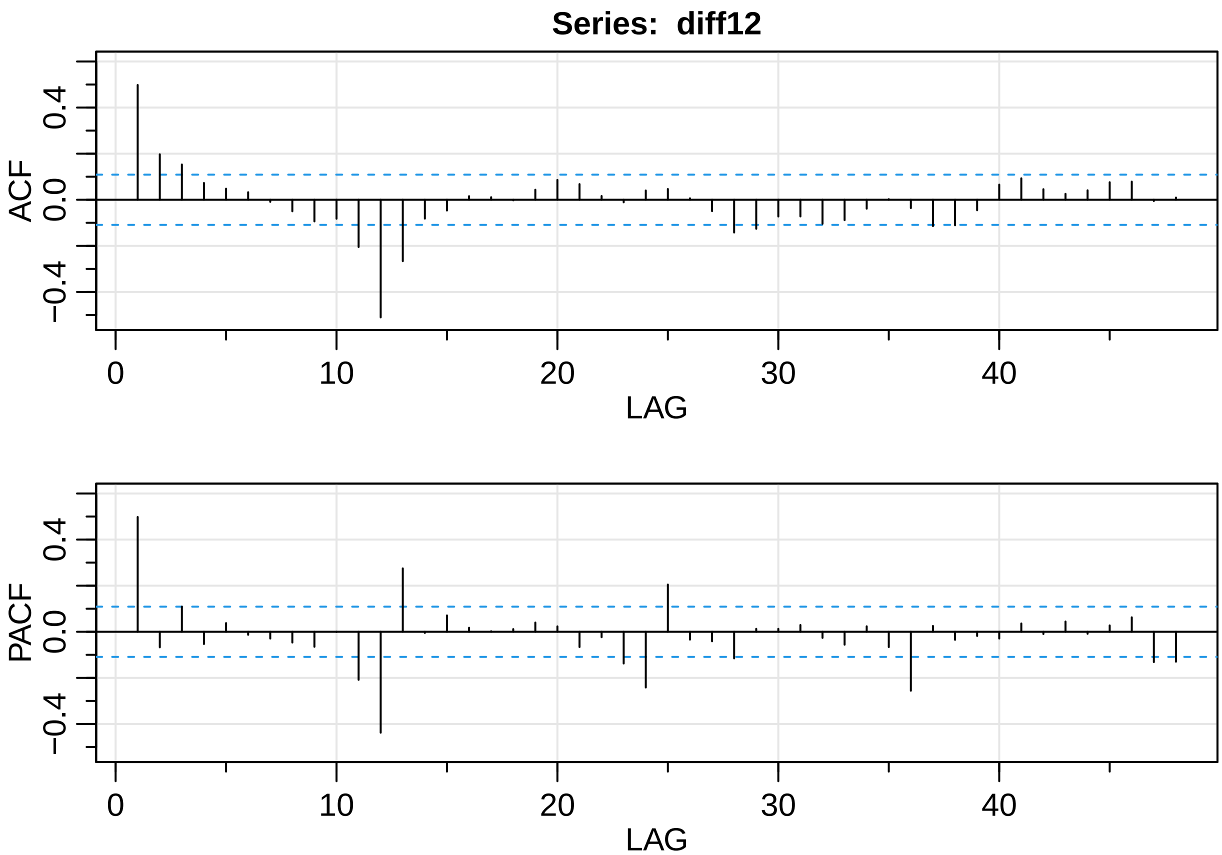
```r
#repeat process for water_tmp
TS_MR_4_2 <- ts(select(MR_4_2,Avg_WTMP))
plot(TS_MR_4_2,type="b")
```




```r
diff12 = diff(TS_MR_4_2,12)
plot(diff12,type="b")
```




```
acf2(diff12,48)
```

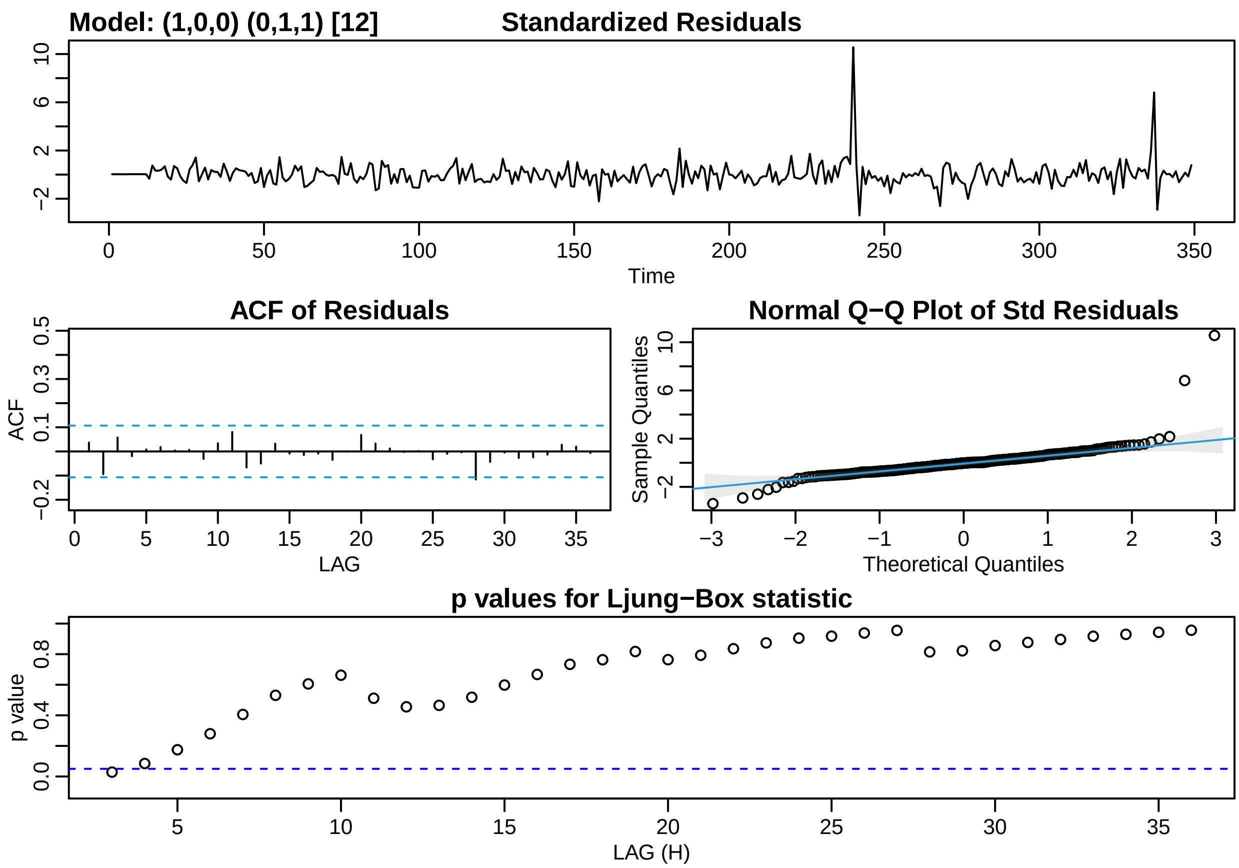



```
##      [,1]  [,2] [,3]  [,4] [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12] [,13]
## ACF   0.5  0.20 0.15  0.07 0.05  0.03 -0.01 -0.05 -0.09 -0.08 -0.20 -0.51 -0.27
## PACF  0.5 -0.07 0.11 -0.05 0.04 -0.01 -0.03 -0.05 -0.07  0.00 -0.21 -0.44  0.28
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF  -0.08 -0.05  0.02  0.01  0.00  0.04  0.09  0.07  0.02 -0.01  0.04  0.05
## PACF  0.00  0.07  0.02  0.00  0.01  0.04  0.02 -0.07 -0.02 -0.14 -0.24  0.20
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF   0.01 -0.05 -0.14 -0.13 -0.07 -0.07 -0.10 -0.09 -0.04  0.00 -0.04 -0.11
## PACF -0.03 -0.04 -0.12  0.01  0.01  0.03 -0.03 -0.06  0.02 -0.07 -0.26  0.03
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF  -0.11 -0.05  0.07  0.09  0.05  0.03  0.04  0.08  0.08 -0.01  0.01
## PACF -0.03 -0.02 -0.03  0.04 -0.01  0.04 -0.01  0.03  0.06 -0.13 -0.13
```

```r
sarima(TS_MR_4_2,1,0,0,0,1,1,12)
```

```
## initial  value 0.342105
## iter   2 value -0.028084
## iter   3 value -0.072888
## iter   4 value -0.123745
## iter   5 value -0.131540
## iter   6 value -0.131605
## iter   7 value -0.131777
## iter   8 value -0.131778
## iter   9 value -0.131779
## iter  10 value -0.131779
## iter  11 value -0.131779
## iter  11 value -0.131779
## iter  11 value -0.131779
```

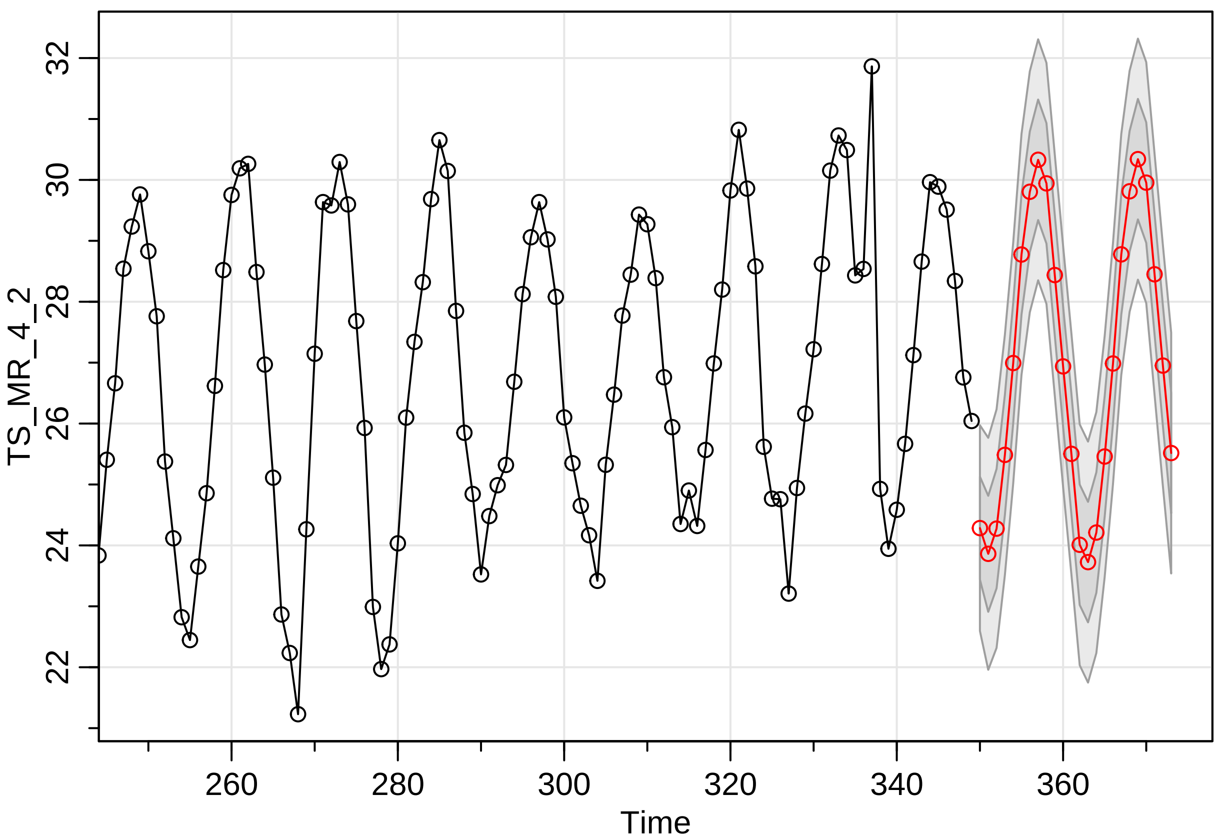
```
## final  value -0.131779
## converged
## initial  value -0.121733
## iter   2 value -0.123402
## iter   3 value -0.125236
## iter   4 value -0.125377
## iter   5 value -0.125393
## iter   6 value -0.125394
## iter   7 value -0.125395
## iter   8 value -0.125395
## iter   9 value -0.125395
## iter  10 value -0.125395
## iter  11 value -0.125396
## iter  12 value -0.125396
## iter  13 value -0.125396
## iter  14 value -0.125396
## iter  15 value -0.125396
## iter  15 value -0.125396
## final  value -0.125396
## 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
##       0.5204  -0.9999    0.0011
## s.e.  0.0464   0.2747    0.0009
##
## sigma^2 estimated as 0.6896:  log likelihood = -435.92,  aic = 879.85
##
## $degrees_of_freedom
## [1] 334
##
## $ttable
##          Estimate     SE t.value p.value
## ar1        0.5204 0.0464 11.2239  0.0000
## sma1      -0.9999 0.2747 -3.6402  0.0003
## constant   0.0011 0.0009  1.2354  0.2175
##
## $AIC
## [1] 2.528298
##
## $AICc
## [1] 2.528499
##
## $BIC
## [1] 2.572207
```

```
sarima.for(TS_MR_4_2,24,1,0,0,0,1,1,12)
```

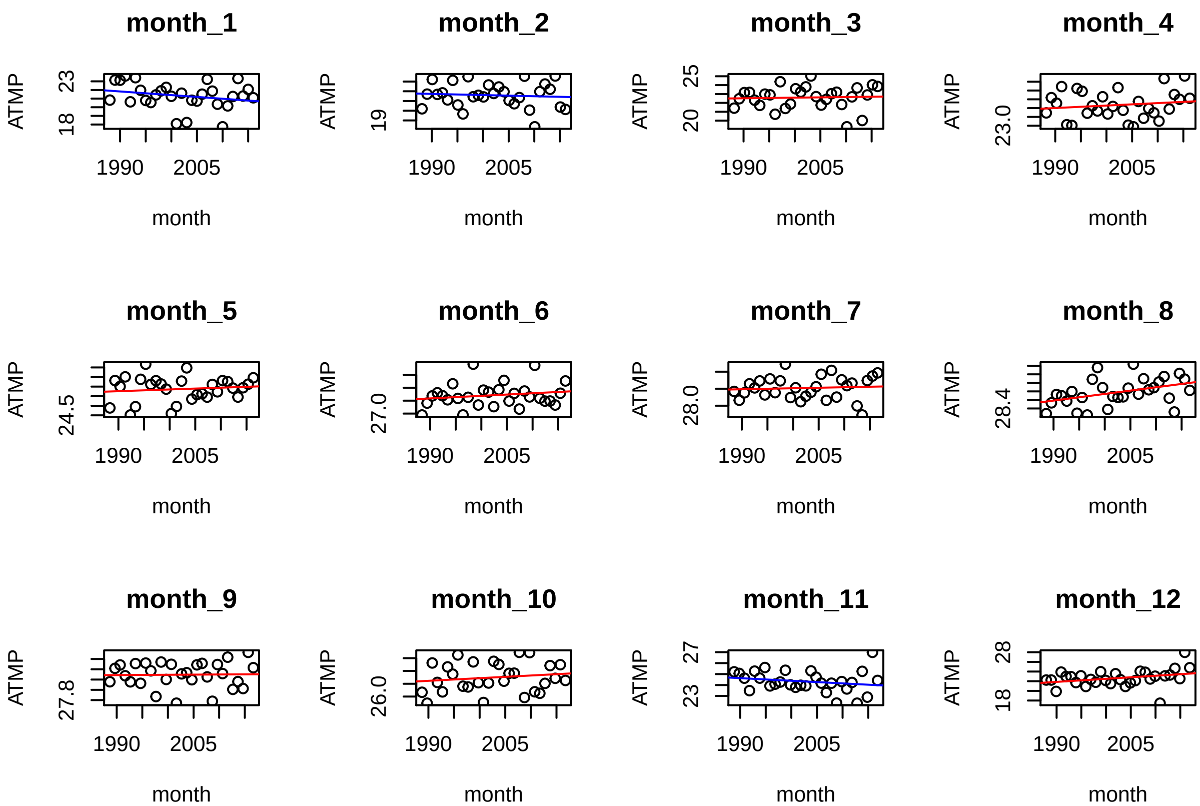


```
## $pred
## Time Series:
## Start = 350
## End = 373
## Frequency = 1
##  [1] 24.28422 23.86190 24.27552 25.48598 26.99359 28.77523 29.80509 30.33040
##  [9] 29.94304 28.43818 26.93859 25.50301 24.01008 23.72573 24.21114 25.45897
## [17] 26.98603 28.77778 29.81292 30.34096 29.95503 28.45092 26.95171 25.51633
##
## $se
## Time Series:
## Start = 350
## End = 373
## Frequency = 1
##  [1] 0.8445810 0.9521053 0.9791963 0.9864053 0.9883486 0.9888741 0.9890159
##  [8] 0.9890522 0.9890547 0.9890281 0.9889203 0.9885194 0.9889297 0.9890408
## [15] 0.9890709 0.9890790 0.9890812 0.9890817 0.9890812 0.9890791 0.9890712
## [22] 0.9890418 0.9889332 0.9885320
```

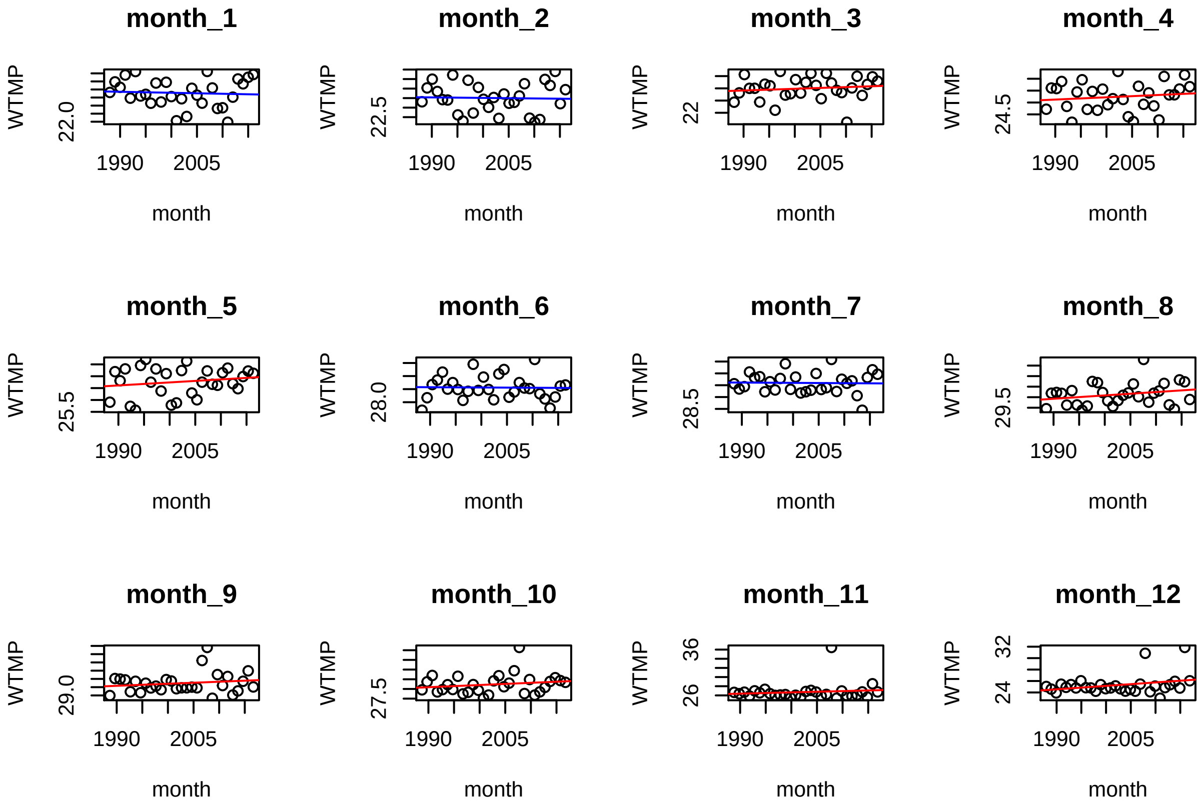
< p style="font-size:20px">

Finally, I decided to make individual graph of each month to show the trend of ATWP and WTMP. For both ATMP and WTMP, the temperature in winter tends to decreasing and the temperature in summer tends to increasing. The plot shows us a nearly linear relationship, so I will build linear regression model following.

```
###Fit lm model

#Red indicates an upward trend
#Blue indicates an downward trend

#plot atmp
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_ATMP~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_ATMP,xlab = "month", ylab = "ATMP",main = paste("month
    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_ATMP,xlab = "month", ylab = "ATMP",main = paste("month
    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_WTMP~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_WTMP,xlab = "month", ylab = "WTMP",main = paste("month
    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_WTMP,xlab = "month", ylab = "WTMP",main = paste("month
    abline(coef(LM_2)[1],coef(LM_2)[2], col="BLUE")

  }
}
```



Conclusion

< p style="font-size:20px">

To conclude, 20 or 30 years of data in Boston is still not enough to make more accurate prediction, But as I indicated in the previous chunk, red line means an upward trend of temperature. Those graphs show us that air and water temperatures in Boston tend to increasing in the summer. So, global warming maybe an issue that society needs to take into consideration.

Reference

2020. Knitr, Rstudio. Beijing.
2021. ggplot2, Rstudio. Beijing.
2022. rstanarm, Rstudio. Beijing.
2023. tidyverse, Rstudio. Beijing.
2024. lubridate, Rstudio. Beijing.
2025. zoo, Rstudio. Beijing.
2026. astsa, Rstudio. Beijing.