Buoy Project: Analysis of Climate Change

Jung Hwa Yeom

MA615: Data Science in R

September 25, 2020

1. Introduction

To begin with, we need to find whether there is evidence of global warming in the weather buoy. We read 20 years of data NDBC Station 44013 in NOAA National Data Buoy Center. This project requires us to process from acquiring data to presenting results. Additionally, there are tasks to make substitutions for NA data and decide an appropriate sampling frequency. To determine sampling frequency, we reduce the number of observations in the dataset. The dataset consists of year, month, date, and time with other various variables. Our group focuses on monthly average air temperature to observe global warming trends. We compare time versus air temperature from 1988 to 2018. If there is a positive slope in the plots, we can conclude global warming has progressed. Furthermore, as the slope is steeper, it indicates global warming drastically proceeds.

2. Explanation of R codes

```
1  library(tidyverse)
2  library(ggplot2)
3  library(rstanarm)
4  library(ggpubr)
5  library(dplyr)
6  library(tidyr)
7  # library(stringr)
8
9
```

At the beginning of the codes, we show all the packages we use in this project. We used tidyverse, ggplot2, rstanarm, ggpubr, dplyr, and tidyr.

```r

### make URLs

url1 <- "https://www.ndbc.noaa.gov/view_text_file.php?filename=44013h"
url2 <- ".txt.gz&dir=data/historical/stdmet/"

years <- c(1998:2018)

urls <- str_c(url1, years, url2, sep = "")

filenames <- str_c("buoy", years, sep = "")

```

Before reading data, we need to bring data from NDBC Station 44013 in NOAA National Data Buoy Center. Both "url1" and "url2" enable reading data from NDBC Station 44013. We use the data from 1988 to 2018, so we extracted those data by setting "years." The "urls" shows an url in the formats of url1 and url2. If the data is from 1988, then the filename is "buoy1988." We include buoy and years in the filenames to easily indicate the data year.

```r
###  Read the data from the website

N <- length(urls)

for (i in 1:N){
  if (i <= 7) {
    suppressMessages(  ###  This stops the annoying messages on your screen.  Do this last.
      assign(filenames[i], read_table(urls[i], col_names = c("YYYY","MM","DD","hh",
                                                             "WDIR","WSPD", "GST","WVHT","DPD","APD",
                                                             "MWD","PRES","ATMP","WTMP","DEWP","VIS","TIDE"), skip = 1)
  ))}
  else {
  suppressMessages(  ###  This stops the annoying messages on your screen.  Do this last.
    assign(filenames[i], read_table(urls[i], col_names = c("YYYY","MM","DD","hh","mm",
                                    "WDIR","WSPD", "GST","WVHT","DPD","APD",
                                    "MWD","PRES","ATMP","WTMP","DEWP","VIS","TIDE"), skip = 2))

    )}

  file <- get(filenames[i])
  colnames(file)[1] <-"YYYY"

  # put '19' in front of 98
  if (i == 1) {
    file[i] = file[i] + 1900

  }

  if (i <= 3) {
    file[,'TIDE'] = 0
  }

  if (i <= 7) {
    file[,'mm'] = 0
  }

  if(i == 1){
    MR <- file
  }

  else{
    MR <- rbind.data.frame(MR, file)

  }
}
```

This part is reading the data from the website. First of all, we define “N” as the length of urls. We used a for loop to read tables and to put ‘19’ in front of years. To further illustrate, in the tables, years before 2000 show only two last digits of the year. For example, if the year is 1988, then only ‘88’ is shown in the table. After processing the for loop, then all the tables will store in “MR” by using rbind.data.frame function.

```
MR <- filter(MR, MR$ATMP < 50)
MR$MM <- as.numeric(MR$MM)
two_pm <- MR[MR$hh %in% c("14"), ]
```

One of the tasks in this project is to make substitutions for NA data. For instance, there is data that air temperature is 999 celsius degrees, which is impossible and meaningless to use. Therefore, we remove the data which has over 50 Celsius degrees in the air temperature (ATMP) column. In addition, as.numeric function allows us to make “MR$MM” a numeric variable. We select 2 pm because 2 pm is the hottest time in a day and we analyze the air temperature at 2 pm. “two_pm” chooses “14” which means 2 pm in the “hh” columns.

```
for (i in 1998:2018){
  year <- two_pm[two_pm$YYYY %in% c(i),]
  for (j in 1:12) {
    month <- year[year$MM %in% c(j),]
    if (j == 1){
      ave = mean(month$ATMP)
    } else {
      ave <- rbind(ave, mean(month$ATMP))
    }
  }
  if (i == 1998) {
    data_frame <- rbind.data.frame("year" = i, "ave_temp" <- ave)
  } else {
    data_frame <- cbind.data.frame(data_frame, rbind.data.frame("year" = i, "ave_temp" <- ave))
  }
}
final.df <- as.data.frame(t(data_frame))

colnames(final.df) = c("year","Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec")
```

In this part, we use another for loop to calculate the average of each month. The for loop has similar properties with the first our for loop. We take an average of the month in 20 years by writing a mean function and store the average by using rbind function. Basically, we are making a new data frame that only has the average of the month at 2 pm from 1988 to 2018. “final.df” is our final data frame before we start to plot a graph. Lastly, we set up new

column names and the column names are years and months to better indicate which month of the average air temperature is.

```
# do the plot from here

jj<-ggplot(data=final.df, aes(x=year, y=Jan)) +
  geom_point() +
  labs(x="Year", y="ATMP", title= "") + geom_smooth(method = lm)
ff<- ggplot(data=final.df, aes(x=year, y=Feb)) +
  geom_point()+
  labs(x="Year", y="ATMP", title= "") + geom_smooth(method = lm)
mm <-ggplot(data=final.df, aes(x=year, y=Mar)) +
  geom_point() +
  labs(x="Year", y="ATMP", title= "") + geom_smooth(method = lm)
aa<-ggplot(data=final.df, aes(x=year, y=Apr)) +
  geom_point() +
  labs(x="Year", y="ATMP", title= "") + geom_smooth(method = lm)
mayy<-ggplot(data=final.df, aes(x=year, y=May)) +
  geom_point() +
  labs(x="Year", y="ATMP", title= "") + geom_smooth(method = lm)
junn<-ggplot(data=final.df, aes(x=year, y=Jun)) +
  geom_point() +
  labs(x="Year", y="ATMP", title= "") + geom_smooth(method = lm)
jull<-ggplot(data=final.df, aes(x=year, y=Jul)) +
  geom_point() +
  labs(x="Year", y="ATMP", title= "") + geom_smooth(method = lm)
augg<-ggplot(data=final.df, aes(x=year, y=Aug)) +
  geom_point() +
  labs(x="Year", y="ATMP", title= "") + geom_smooth(method = lm)
ss<-ggplot(data=final.df, aes(x=year, y=Sep)) +
  geom_point() +
  labs(x="Year", y="ATMP", title= "") + geom_smooth(method = lm)
oo<-ggplot(data=final.df, aes(x=year, y=Oct)) +
  geom_point() +
  labs(x="Year", y="ATMP", title= "") + geom_smooth(method = lm)
novv<-ggplot(data=final.df, aes(x=year, y=Nov)) +
  geom_point() +
  labs(x="Year", y="ATMP", title= "") + geom_smooth(method = lm)
dd<-ggplot(data=final.df, aes(x=year, y=Dec)) +
  geom_point() +
  labs(x="Year", y="ATMP", title= "") + geom_smooth(method = lm)
```

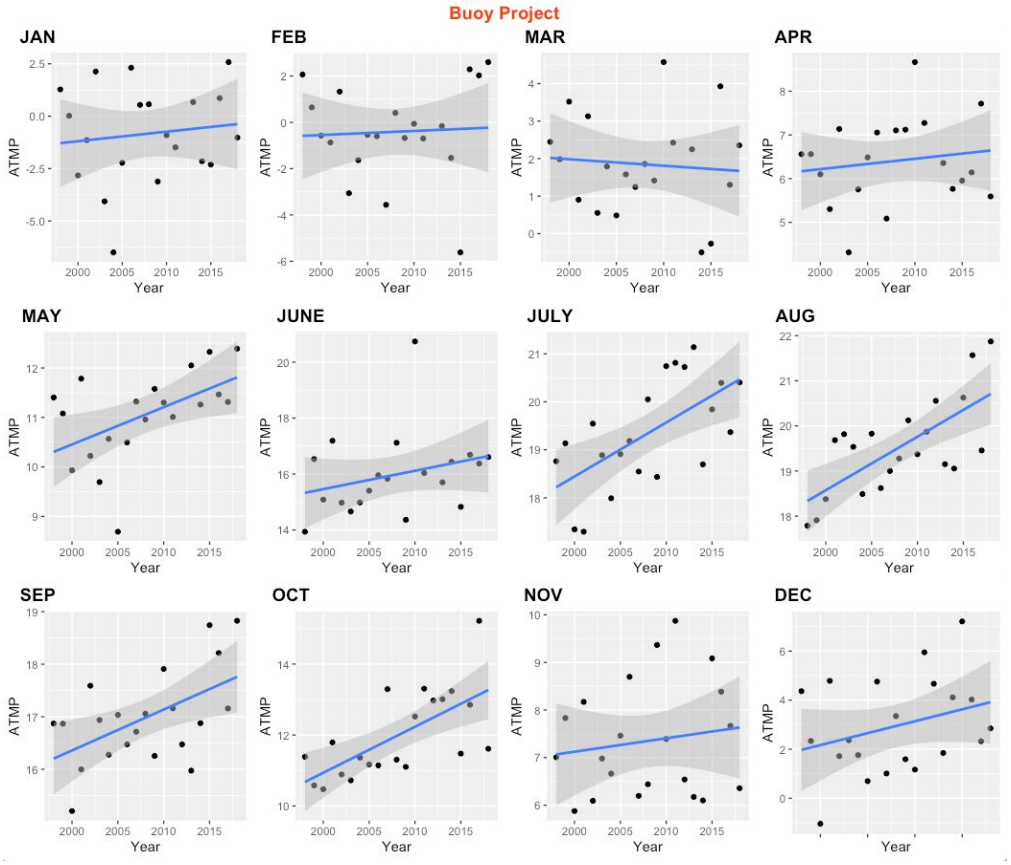
We make 12 plots because there are 12 months in a year; thus, it is effective to compare each month. “jj”, “ff”, “mm”, etc. mean each month. We use ggplot, geom_plot, and geom_smooth to plot the graph. The x-axis is “Year” and the y-axis is “ATMP” to compare time and air temperature. Furthermore, we do not remove shades in the plot because we want to observe confidence intervals.

```
figure <- ggarrange(jj, ff, mm, aa, mayy, junn, jull, augg, ss, oo, novv, dd + rremove("x.text"),
          labels = c("JAN", "FEB", "MAR", "APR", "MAY","JUNE","JULY","AUG","SEP","OCT","NOV","DEC"),
          ncol = 4, nrow = 3)
annotate_figure(figure,
                top = text_grob("Buoy Project", color = "red", face = "bold", size = 14))
```

Finally, we need to put 12 graphs on one page to view the graphs at once. To combine all graphs on one page, we use ggarange function and label each month on the top left corner.

Annotate_figure function allows us to format the graph such as labeling a title, assigning colors and font size.

3. Interpretation



The above graphs are the average air temperature of each month from 1998 to 2018. Grey shades around the lines are confidence intervals. If a graph has a wide range of the confidence interval, it means the global warming trend is not strong enough compared to a narrow one. The summer season has steeper positive slopes and narrow confidence intervals, so we can interpret air temperature is rising as time goes, showing global warming is rapidly progressing. To be specific, July and August are the months when temperatures had risen the most in 20 years. On the other hand, the winter season does not show a clear positive slope or narrow confidence intervals. However, it does not mean that global warming does not happen

in the winter season because not only air temperature is the factor that determines the level of global warming. Global warming leads to extreme weather both hotter summer and colder winter. To sum up, we can conclude global warming progressed from 1988 to 2018 and expect global warming will become more serious in the future.

Bibliography


```
> citation(package = "tidyverse")

  Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686,
  https://doi.org/10.21105/joss.01686


> citation(package = "ggplot2")

To cite ggplot2 in publications, please use:

  H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.


> citation(package= "rstanarm")

To cite rstanarm in publications please use the first citation entry. If you were using the
'stan_jm' modelling function then, where possible, please consider including the second citation
entry as well.

  Goodrich B, Gabry J, Ali I & Brilleman S. (2020). rstanarm: Bayesian applied regression modeling
  via Stan. R package version 2.21.1 https://mc-stan.org/rstanarm.

  Brilleman SL, Crowther MJ, Moreno-Betancur M, Buros Novik J & Wolfe R. Joint longitudinal and
  time-to-event models via Stan. StanCon 2018. 10-12 Jan 2018. Pacific Grove, CA, USA.
  https://github.com/stan-dev/stancon_talks/


> citation(package= "ggpubr")

To cite package 'ggpubr' in publications use:

  Alboukadel Kassambara (2020). ggpubr: 'ggplot2' Based Publication Ready Plots. R package version
  0.4.0. https://CRAN.R-project.org/package=ggpubr


> citation(package = "dplyr")

To cite package 'dplyr' in publications use:

  Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2020). dplyr: A Grammar of Data
  Manipulation. R package version 1.0.2. https://CRAN.R-project.org/package=dplyr


> citation(package = "tidyr")

To cite package 'tidyr' in publications use:

  Hadley Wickham (2020). tidyr: Tidy Messy Data. R package version 1.1.2.
  https://CRAN.R-project.org/package=tidyr
```