Boston Buoy Project Report

Maggie Sha

2020/9/25

Project Question

Global warming has became a popular topic. It is trending on TV, books, advertisements on the roadside, almost everywhere, to alarm people that our earth is experiencing quick change to a dreadful direction due to human activities. Our question for this project is, are we able to find the evidence of global warming using the collected data by a single weather buoy at Boston in the NOAA National Data Buoy Center?

Data Source

We are using a publically available dataset from NDBC, National Data Buoy Center.

- The data is from https://www.ndbc.noaa.gov/station_page.php?station=44013
- In this report, I only focus on data from year 1998 to 2018.

Approach and Organization of the work

In this dataset, there are many irrelevant data such as wind direction, pressure, etc, and data without meaning, such as TIDE, which is filled only by number 99. For this project, I do not use data other than time (year, month, day, hour) and ATMP, the air temperature.

The buoy data is collected from 1984 until now, I choose the most recent 21 years from 1998 to 2018 to do the analysis. To find the most typical result, I focus on the most hottest moment of a day, which is 2pm in the afternoon. Because the ATMP of the same month in the same year is very close, I use the average air temperature of each month instead of analyzing on every day in each month. By doing so, if I can get the scatterplots with clear positive slope on the average temperature of every month over the year 1998 to 2018, the result would be very convincing on showing the global warming trend.

After I go over the dataset in detail, I find there are many places that need to be organized. My task includes renaming the columns, unifying the column number and type, substituting the NA data, and filtering the air temperature data with extreme valus such as 99 celcius and 999 celcius, which are apparently fake data. Finally, I put all the data into one data frame called "MR".

Then, I narrow the scope to only see data at 2pm for each day. Here is my result:

```
two_pm <- MR[MR$hh %in% c("14"), ]
print(two_pm)
```

```
## # A tibble: 7,354 x 18
##     YYYY    MM DD    hh     WDIR  WSPD   GST  WVHT   DPD   APD   MWD  PRES  ATMP
##    <dbl> <dbl> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  1998     1 01    14      288   9.7  12.4 1     4     3.99   999 1029. -11.5
##  2  1998     1 02    14      223   8.9  10.5 0.61  3.12  3.16   999 1021.   2.7
##  3  1998     1 03    14      220   3.9   4.8 0.35 10     3.94   999 1020.   7.1
##  4  1998     1 04    14      247   4.3   4.6 0.24  8.33  4.44   999 1024.   7.5
##  5  1998     1 05    14      159   1.7   2.3 1.15  6.25  5.36   999 1030.   5.2
##  6  1998     1 06    14      186   2.6   3.9 0.49  6.67  4.95   999 1019    7.7
##  7  1998     1 07    14       47   8.2   9.9 1.35  6.67  4.8    999 1023.   4.4
##  8  1998     1 08    14       18   5.4   6.8 1.94  8.33  6.29   999 1010.   5.8
##  9  1998     1 09    14       35   8.7  10.2 1.69  7.69  5.26   999 1008.   3.8
## 10  1998     1 10    14      225   5.5   6.4 1.02  7.14  6.34   999 1016.   4.3
## # ... with 7,344 more rows, and 5 more variables: WTMP <dbl>, DEWP <dbl>,
## #   VIS <dbl>, TIDE <dbl>, mm <chr>
```

The next step is to collect the mean of air temperature of each month into one dataframe. I start from one year, the year 1998, and one month, which is January, to do the calculation. After I getting the average ATMP of January, I put it into a set called "ave", and then I calculate the average ATMP of February in 1998, putting the result into "ave" again. Following the same process, in the end, "ave" contains 12 average ATMP number corresponds to 12 months in 1998.

Next, I build a dataframe called "data_frame" with 13 rows and 21 columns to collect year number and its corresponding 12 average temperature numbers in each column. For example, the first column will be "1998, (following by 12 averages.)". After year 1998 is finished, I keep doing the same work for year 1999, until I have all 21 years been included in the "data_frame". Finally, I switch the columns and rows to make the columns be "year,Jan,Feb…Dec", and give the new dataframe a name "final.df"

Here is how "final.df" looks like, notice there are six NA data because of the missing values of ATMP from January to June in 2012 in the dataset.

```
final.df <- as.data.frame(t(data_frame))
colnames(final.df) = c("year","Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct
print(final.df)
```

```
##       year         Jan         Feb        Mar      Apr       May      Jun
## V1    1998  1.27666667  2.06428571  2.4466667 6.558621 11.403333 13.93793
## V1.1  1999  0.01935484  0.64444444  1.9833333 6.560000 11.080000 16.53333
## V1.2  2000 -2.82580645 -0.57241379  3.5193548 6.100000  9.932258 15.08333
## V1.3  2001 -1.14193548 -0.87142857  0.9033333 5.303448 11.783871 17.18966
## V1.4  2002  2.13000000  1.32500000  3.1290323 7.136667 10.222581 14.97586
## V1.5  2003 -4.06666667 -3.06071429  0.5516129 4.313333  9.693548 14.66000
## V1.6  2004 -6.49677419 -1.63448276  1.7903226 5.753333 10.566667 14.98333
## V1.7  2005 -2.22580645 -0.54444444  0.4838710 6.486667  8.689655 15.40333
## V1.8  2006  2.31290323 -0.60000000  1.5793103 7.055556 10.493548 15.95667
## V1.9  2007  0.53870968 -3.56296296  1.2433333 5.086667 11.319355 15.83103
## V1.10 2008  0.56896552  0.40714286  1.8566667 7.103448 10.958065 17.11667
## V1.11 2009 -3.11935484 -0.67857143  1.4161290 7.120000 11.577419 14.36207
## V1.12 2010 -0.90645161 -0.06428571  4.5741935 8.666667 11.300000 20.74000
## V1.13 2011 -1.48709677 -0.69642857  2.4258065 7.273333 11.009677 16.03667
## V1.14 2012         NaN         NaN        NaN      NaN       NaN      NaN
## V1.15 2013  0.67419355 -0.16071429  2.2483871 6.356667 12.051613 15.70500
## V1.16 2014 -2.14838710 -1.53571429 -0.5000000 5.766667 11.261290 16.43667
## V1.17 2015 -2.30967742 -5.61428571 -0.2677419 5.953333 12.325806 14.82667
## V1.18 2016  0.86129032  2.28965517  3.9290323 6.143333 11.464516 16.69000
## V1.19 2017  2.58064516  2.02500000  1.3032258 7.716667 11.313333 16.37333
## V1.20 2018 -1.01935484  2.60000000  2.3548387 5.593333 12.390323 16.60667
##            Jul      Aug      Sep      Oct      Nov        Dec
## V1    18.76129 17.78788 16.87000 11.38667 7.006667  4.3645161
## V1.1  19.13667 17.90968 16.86667 10.58065 7.830769  2.3354839
## V1.2  17.34194 18.37742 15.20333 10.47419 5.876667 -1.0419355
## V1.3  17.29355 19.68387 15.99667 11.79355 8.170000  4.7866667
## V1.4  19.54516 19.81613 17.59000 10.89000 6.092593  1.7178571
## V1.5  18.88710 19.53548 16.93667 10.72258 6.980000  2.3645161
## V1.6  17.99032 18.49032 16.27667 11.36452 6.663333  1.7580645
## V1.7  18.90968 19.82667 17.03333 11.17097 7.460000  0.6967742
## V1.8  19.18387 18.62258 16.47333 11.14516 8.696667  4.7548387
## V1.9  18.54667 19.00000 16.71333 13.29355 6.196667  1.0100000
## V1.10 20.05161 19.27742 17.05667 11.30667 6.440000  3.3500000
## V1.11 18.43548 20.12258 16.25333 11.10323 9.365517  1.5900000
## V1.12 20.74194 19.37097 17.91000 12.52258 7.390000  1.1677419
## V1.13 20.81290 19.87097 17.16000 13.30645 9.873333  5.9500000
## V1.14 20.72581 20.55484 16.47333 12.97419 6.540000  4.6677419
## V1.15 21.13889 19.15161 15.97333 13.00968 6.173333  1.8451613
## V1.16 18.69677 19.05484 16.87667 13.23871 6.100000  4.1129032
## V1.17 19.84194 20.62581 18.74333 11.47742 9.083333  7.2096774
## V1.18 20.39355 21.56552 18.21379 12.85484 8.383333  4.0225806
## V1.19 19.36774 19.45484 17.15667 15.21935 7.666667  2.3258065
## V1.20 20.40323 21.87097 18.82667 11.61290 6.356667  2.8500000
```

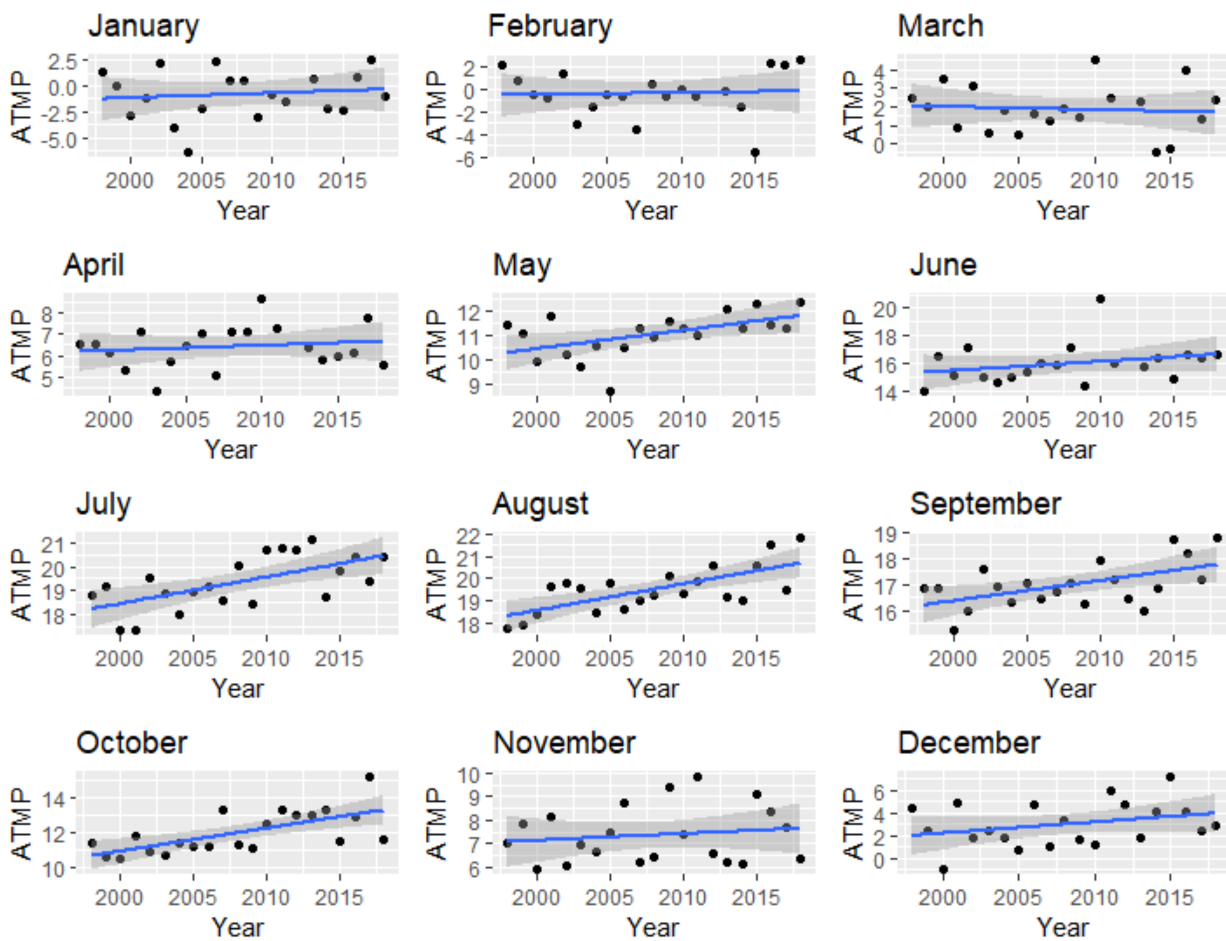
Plot

Based on the data in "final.df", I can apply ggplot function to get the scatterplot of average temperature of each month over 21 years. There are 12 plots, standing for the change of average air temperature at 2pm in every month from 1998 to 2018.

Visialization

```r
library(png)
library(grid)

img = readPNG("Rplot.png")
grid.raster(img)
```



Analysis and Conclusion

I fit a linear model with se = TRUE to the change in average air temperature in each month over 21 years to look at the slopes. From the plots, we can see that while in the summer season the slopes are intuitively positive, they are relatively flat in the winter season. From May to October, the gray zone which stands for the 95% confidence interval is thinner than that in other months, this means the uncertainty in these months is small. From November to April, the uncertainty is larger and the average temperature from year to year fluctuates more.

Based on the plots, we can conclude that there is evidence of obvious temperature increasing in the warm seasons (temperature greater than 10 celcius). To be specific, May to October from year 1998 to 2018. This can be viewed as a strong evidence of global warming.

However, from November to April, when the temperature is low (less than 10 celcius), the evidence is not too strong, and in March the slope is even negative. Can we use this as the evidence to say global warming is fake? My answer is no.

The first reason is related to climatology. Since the buoy locates in Boston, it stands for mid-latitude region in the Northern hemisphere. According to climatology, when global warming happens, the melting of glaciers in the Arctic region will cause the cold air moves towards south from high latitudes, thus in the winter season we still feel cold or colder in Boston.

The second reason is, the decreasing temperature in cold seasons only happends in some region, such as the east coast of the United States and Europe. In many other region the temperature in winter seasons is increasing. According to NOAA's global climate report, the global temperature in winter seasons increased tremendously from 1910 to 2020.

References

NOAA.(2020).Global Climate Report[online]. Available from:https://www.ncdc.noaa.gov/sotc/global/202006 [accessed 25 September 2020].

```
citation("ggplot2")
```

```
##
## To cite ggplot2 in publications, please use:
##
##   H. Wickham. ggplot2: Elegant Graphics for Data Analysis.
##   Springer-Verlag New York, 2016.
##
## A BibTeX entry for LaTeX users is
##
##   @Book{,
##     author = {Hadley Wickham},
##     title = {ggplot2: Elegant Graphics for Data Analysis},
##     publisher = {Springer-Verlag New York},
##     year = {2016},
##     isbn = {978-3-319-24277-4},
##     url = {https://ggplot2.tidyverse.org},
##   }
```

```r
citation("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
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {tidyr: Tidy Messy Data},
##     author = {Hadley Wickham},
##     year = {2020},
##     note = {R package version 1.1.2},
##     url = {https://CRAN.R-project.org/package=tidyr},
##   }
```

```r
citation("citation")
```

```
##
## Dietrich J (2020). _citation: Software Citation Tools_. R package
## version 0.4.1.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {citation: Software Citation Tools},
##     author = {Jan Philipp Dietrich},
##     year = {2020},
##     note = {R package version 0.4.1},
##   }
```

```r
citation("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
##
```

```
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {ggpubr: 'ggplot2' Based Publication Ready Plots},
##     author = {Alboukadel Kassambara},
##     year = {2020},
##     note = {R package version 0.4.0},
##     url = {https://CRAN.R-project.org/package=ggpubr},
##   }
```

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