# 615 HW4

Jordan Stout

9/27/2022

Include:

```
library(data.table)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
library(tibble)
library(rstanarm)
```

```
## Loading required package: Rcpp
## This is rstanarm version 2.32.1

## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!

## - Default priors may change, so it's safest to specify priors, even if equivalent to the defaults.

## - For execution on a local, multicore CPU with excess RAM we recommend calling

##   options(mc.cores = parallel::detectCores())
```

READ IN DATA

```
get_url <- function(year) {
  paste0("https://www.ndbc.noaa.gov/view_text_file.php?filename=44013h", year, ".txt.gz&dir=data/histori
}

#========================================================
#INITIATE DATAFRAME WITH 1985 DATA
year1<-"1985"
buoy<-read.table(get_url(year1), header = TRUE, sep = "", na.strings = "MM", fill = TRUE)

header=scan(get_url(year1),what= 'character',nlines=1)
colnames(buoy)<-header
buoy = buoy %>%
  add_column(mm = NA, .after = "hh") %>%
  add_column(TIDE = NA, .after = "VIS")
#=========================================================
#HANDLE YEARS 1986-2006
years <- 1986:2006
for (i in years) {
  url <- get_url(i)
  temp_data <- read.table(get_url(i), header = TRUE, sep = "",fill = TRUE)
  temp_data = temp_data %>%
    add_column(mm = NA, .after = "hh") %>%
    add_column(TIDE = NA, .after = "VIS")
  buoy <- bind_rows(buoy, temp_data)
}
buoy$YY<-na.omit(c(buoy$YY, buoy$YYYY))
buoy <- buoy %>%
  select(-YYYY, -TIDE.1, -mm.1)
colnames(buoy)[colnames(buoy) == "YY"] <- "YYYY"
#==========================================================
#HANDLE YEARS 2007-2023
years<-2007:2023
for (i in years) {
  url <- get_url(i)
  temp_data <- (read.table(get_url(i), header = FALSE, sep = "",fill = TRUE, skip=1))
  header=scan(get_url(i),what= 'character',nlines=1)
  colnames(temp_data)<-header
  buoy <- bind_rows(buoy, temp_data)
}

buoy$YYYY<-na.omit(c(buoy$YYYY, buoy$`#YY`))
```

```r
buoy$BAR<-na.omit(c(buoy$BAR, buoy$PRES))
buoy <- buoy %>%
  select(-`#YY`, -PRES)

buoy$WD<-na.omit(c(buoy$WD, buoy$WDIR))
buoy <- buoy %>%
  select(-WDIR)
```

Your next exercise is to identify and deal with the null data in the dataset.Recall from class that for WDIR and some other variables these showed up as 999 in the dataset. Convert them to NA's. Is it always appropriate to convert missing/null data to NA's? When might it not be? Analyze the pattern of NA's. Do you spot any patterns in the way/dates that these are distributed?

It is not always appropriate to convert missing/null data to NA. First, it could be an indicator of a larger issue with your dataset, either the reading in process or the data collection. Second, if you plan to do operations on the data, you may want to replace missing data points with various other metrics like the mean or the mode assuming this wouldn't effect the integrity of your study.

CLEAN DATA

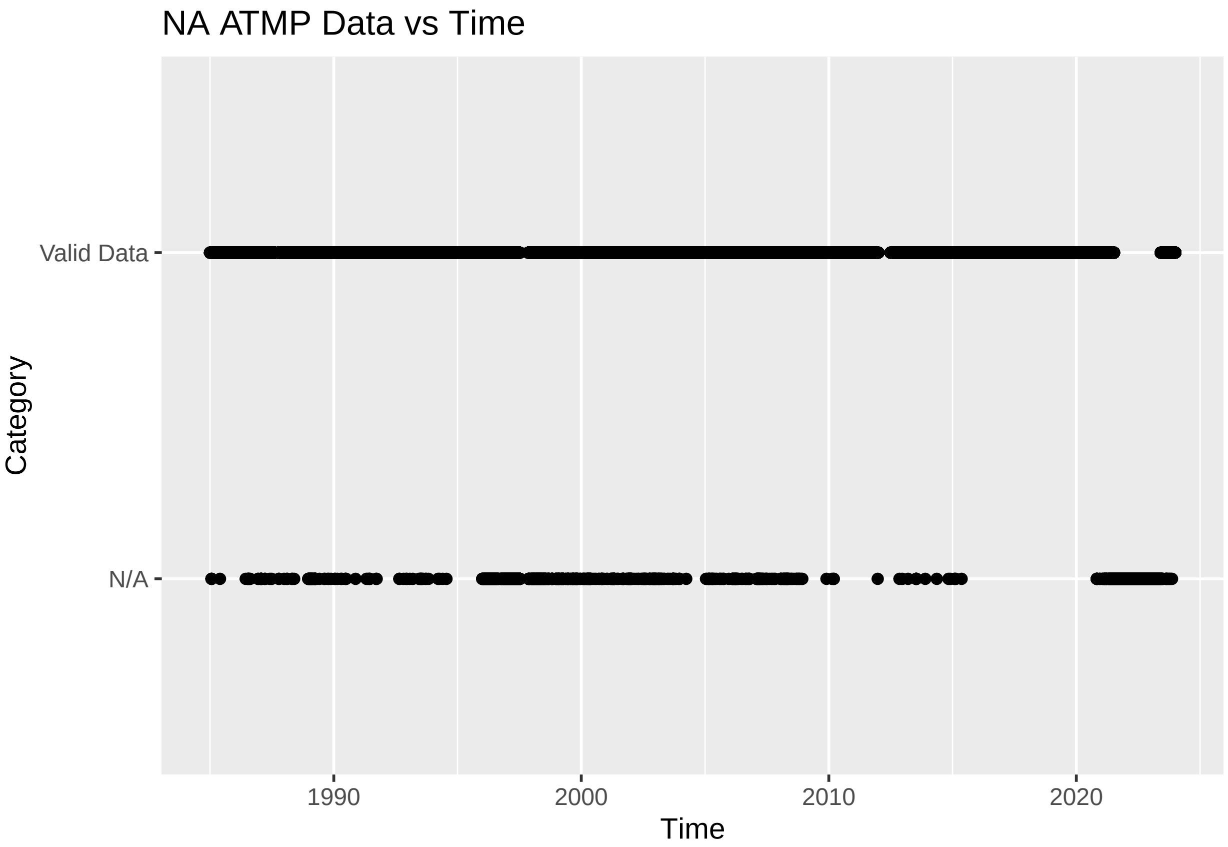
```r
#Remove TIDE column because it has nothing except 99 and NA
buoy=buoy %>%
  select(-TIDE)
#set all trash data to NA
buoy$VIS<-ifelse(buoy$VIS==99, NA, buoy$VIS)
buoy$DEWP<-ifelse(buoy$DEWP==999, NA, buoy$DEWP)
buoy$MWD<-ifelse(buoy$MWD==999, NA, buoy$MWD)
buoy$APD<-ifelse(buoy$APD==99, NA, buoy$APD)
buoy$DPD<-ifelse(buoy$DPD==99, NA, buoy$DPD)
buoy$WVHT<-ifelse(buoy$WVHT==99, NA, buoy$WVHT)
buoy$BAR<-ifelse(buoy$BAR==9999, NA, buoy$BAR)
buoy$ATMP<-ifelse(buoy$ATMP==999, NA, buoy$ATMP)
buoy$WTMP<-ifelse(buoy$WTMP==999, NA, buoy$WTMP)

#Add new datetime column
buoy$datetime <- ymd_h(paste(buoy$YYYY, buoy$MM, buoy$DD, buoy$hh, sep = "-"))
```

NA DISTRIBUTION ANALYSIS

```r
#create vector over time that specifies whether the ATMP index has a data reading or and NA
ATMP_NA<-ifelse(is.na(buoy$ATMP), "N/A", "Valid Data")
na_df<-data.frame(buoy$datetime, ATMP_NA)

#plot this vector versus time
ggplot(na_df, aes(x = buoy.datetime, y = ATMP_NA)) +
  geom_point() +
  labs(x = "Time", y = "Category", title = "NA ATMP Data vs Time")
```



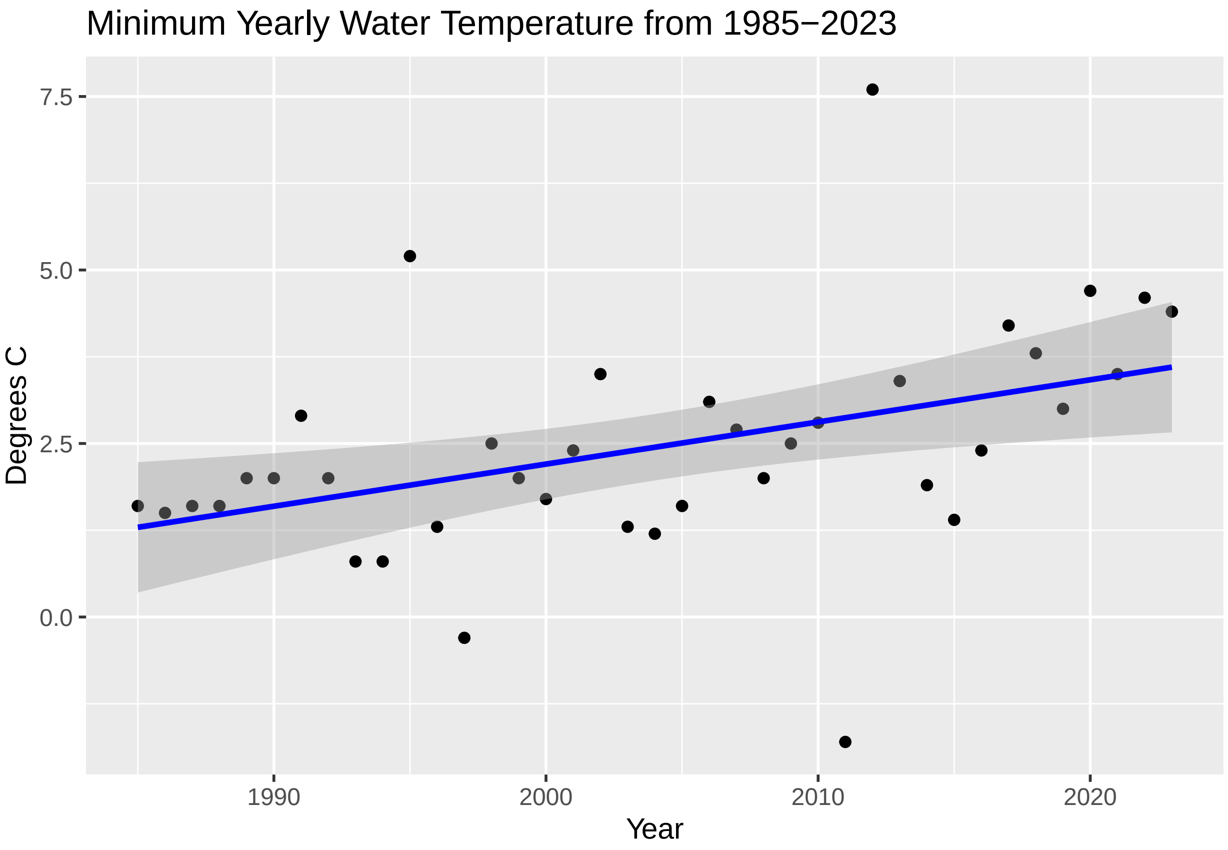
The graph above shows the validity of ATMP recordings from the buoy from 1985-2023. As you can see, beginning in the mid 2015s there was a period of perfect data collection without any null values. Online records state that around this time, as the importance of marine research improved, the NDBC began receiving significantly more funding. It can be inferred that this allowed them to either improve technology on the buoys themselves or increase the allowed bandwidth of data transfer which is stated to be a reason for null data in the past. Public records also state that the NDBC was negatively affected by the COVID-19 pandemic, which may explain the resurgence of null values shortly following 2020.

ATTEMPT TO PROVE CLIMATE CHANGE (IMPOSSIBLE BECAUSE ITS A LIBERAL CONSPIRACY)

```
#============================================
#YEARLY MINIMUM TEMPERATURES
yearly_min_temps <- buoy %>%
  group_by(year = year(datetime)) %>%
  summarise(min = min(WTMP, na.rm = TRUE))

ggplot(yearly_min_temps, aes(x = year, y = min)) +
  geom_point() +                           # Scatter plot
  geom_smooth(method = "lm", col = "blue")+
  labs(title = "Minimum Yearly Water Temperature from 1985-2023", x = "Year", y = "Degrees C")
```

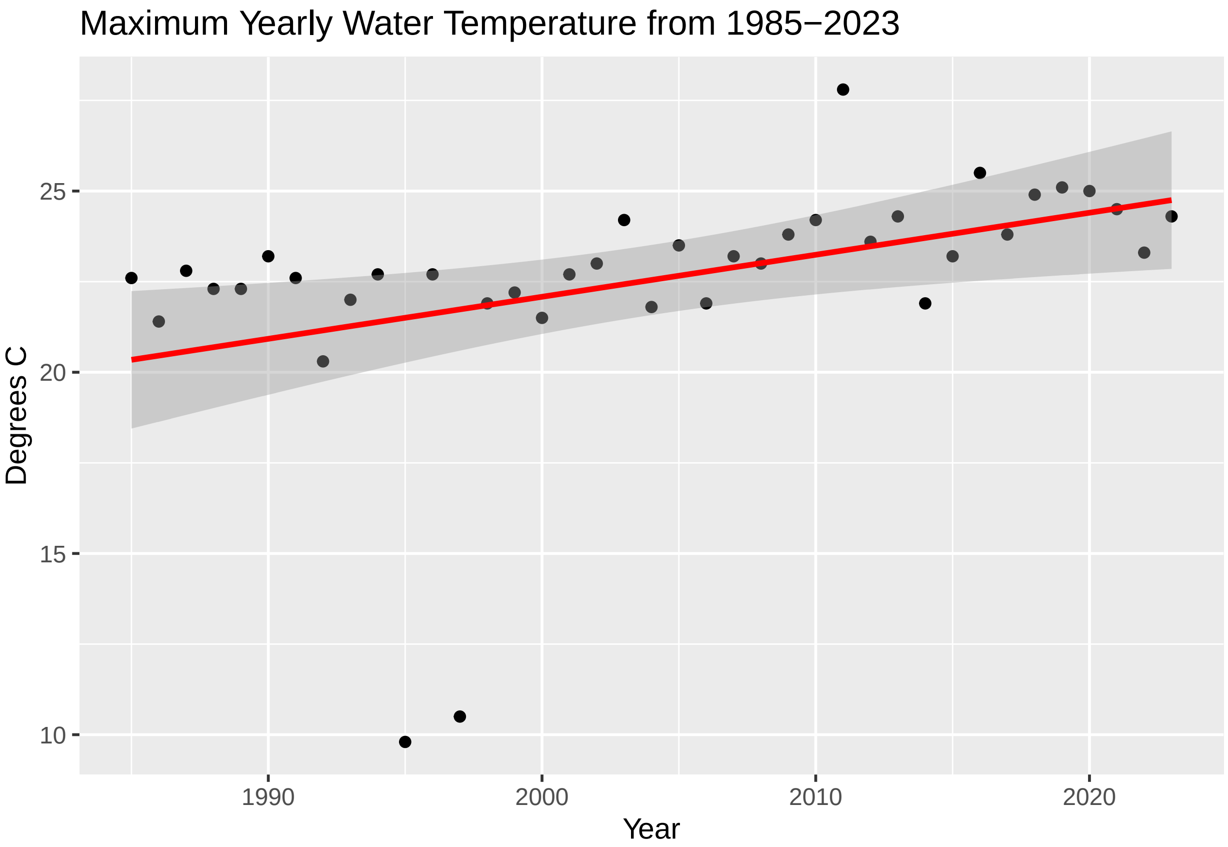
```
## `geom_smooth()` using formula = 'y ~ x'
```



```r
#===========================================
#YEARLY MAX TEMPERATURES
yearly_max_temps <- buoy %>%
  group_by(year = year(datetime)) %>%
  summarise(max = max(WTMP, na.rm = TRUE))

ggplot(yearly_max_temps, aes(x = year, y = max)) +
  geom_point() +                          # Scatter plot
  geom_smooth(method = "lm", col = "red")+
  labs(title = "Maximum Yearly Water Temperature from 1985-2023", x = "Year", y = "Degrees C")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



As you can see in both the yearly maximum and minimum water temperatures, the regression lines are significantly increasing from 1985-2023.

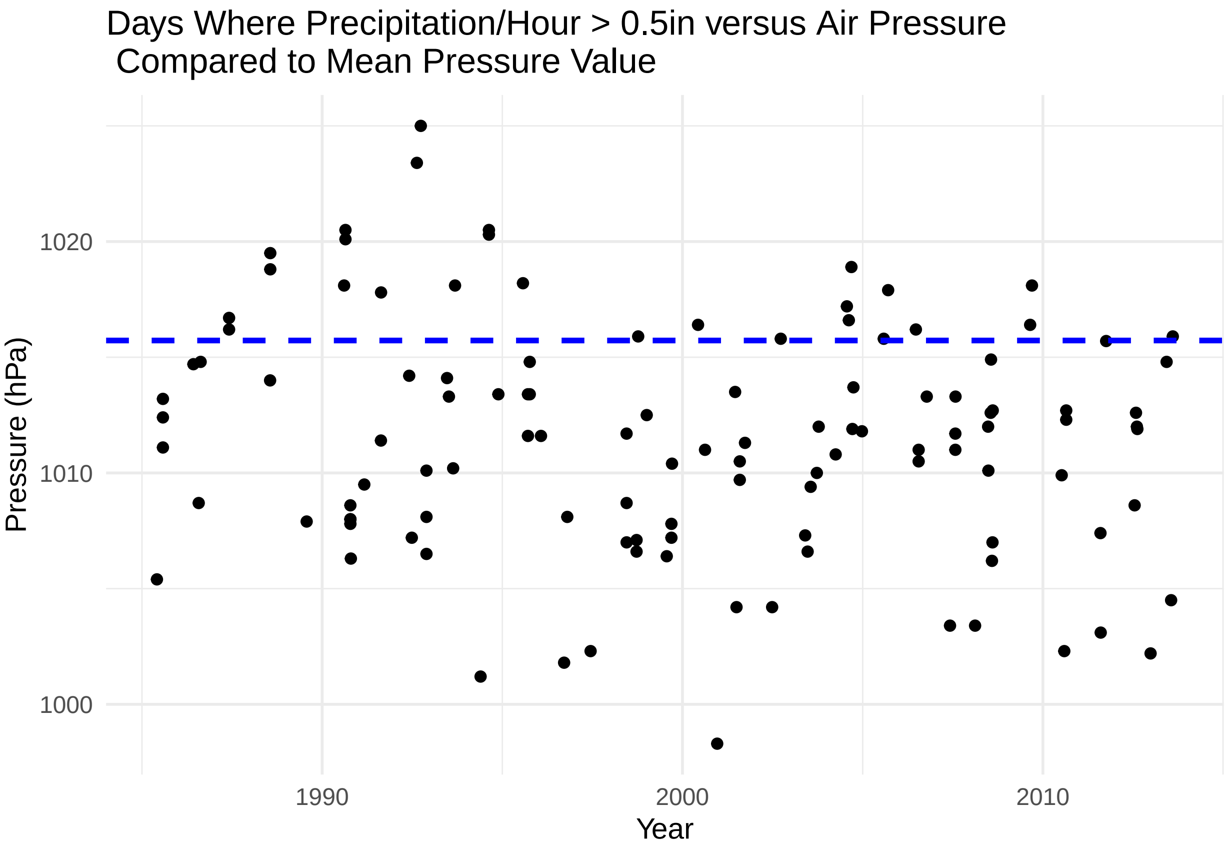
COMPARING RAIN DATA FROM BUOYS AND RAINFALL.CSV

```
rain<-read.csv("Rainfall.csv")
rain$datetime<-ymd_hm(rain$DATE)
rain_subset <- rain[, c("datetime", "HPCP")]
rain_subset <- rain_subset%>%
  left_join(select(buoy, datetime, BAR), by = "datetime")

rain_predict<-stan_glm(HPCP~BAR, data = rain_subset, refresh = 0)
lots_o_rain <- rain_subset %>%
  filter(HPCP > 0.5) %>%
  filter(BAR < 2000)
mean_BAR<-mean(na.omit(buoy$BAR))

ggplot(lots_o_rain, aes(x = datetime, y = BAR)) +
  geom_point() +
  labs(title = "Days Where Precipitation/Hour > 0.5in versus Air Pressure \n Compared to Mean Pressure
  geom_hline(yintercept = mean_BAR, color = "blue", linetype = "dashed", size = 1) +
  theme_minimal()
```

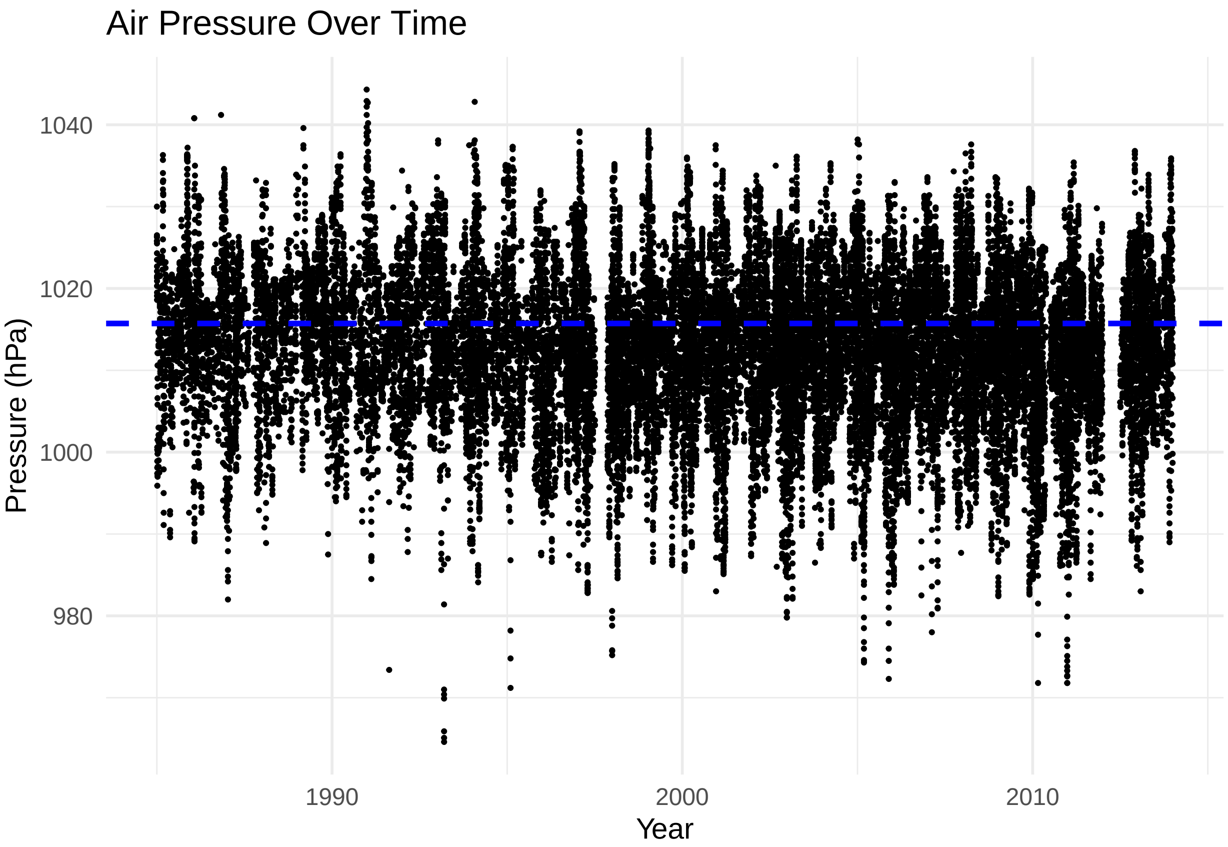
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



A low-pressure system forms when air in a particular area rises. As air rises, it cools and can hold less moisture. This cooling leads to condensation, which forms clouds. As the clouds develop and moisture condenses further, it can lead to the formation of precipitation, such as rain or snow. This is visualized above by graphing the air pressure on days where there was a lot of rain and seeing the majority of the points are below the vertical line which represents the average air temperature.

```
ggplot(rain_subset, aes(x = datetime, y = BAR)) +
  geom_point(size = 0.4) +
  labs(title = "Air Pressure Over Time", x = "Year", y = "Pressure (hPa)") +
  geom_hline(yintercept = mean_BAR, color = "blue", linetype = "dashed", size = 1) +
  theme_minimal()
```

```
## Warning: Removed 1735 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```

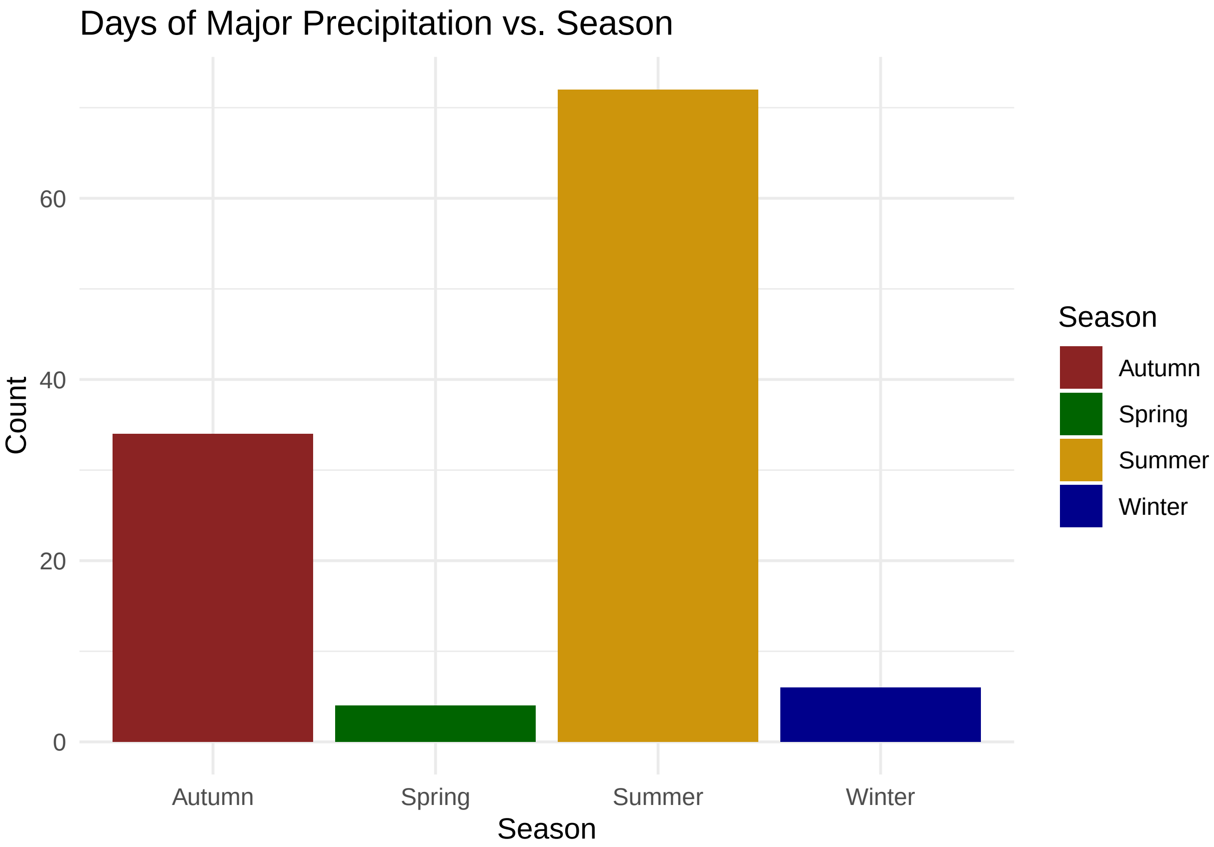


This graphic shows us that air pressure varies a lot per year. Notably, using the ideal gas law PV=nRT, we can prove that the lower winter temperatures bring season-long low air pressure leading to precipitation in the form of snowfall. On the other hand, high summer temperatures will bring many temporary low pressure system bringing precipitation in the form of thunderstorms. We can conclude that summer will have more days with a lot of precipitation, but winter snow will bring fewer but more extreme days of precipitation.

```
get_season <- function(date) {
  month <- month(date)
  if (month %in% c(3, 4, 5)) {
    return("Spring")
  } else if (month %in% c(6, 7, 8)) {
    return("Summer")
  } else if (month %in% c(9, 10, 11)) {
    return("Autumn")
  } else {
    return("Winter")
  }
}

lots_o_rain<-lots_o_rain %>%
  mutate(season = sapply(datetime, get_season))

ggplot(lots_o_rain, aes(x = season, fill = season)) +
  geom_bar(position = "dodge") +
  scale_fill_manual(values = c("Spring" = "darkgreen", "Summer" = "darkgoldenrod3", "Autumn" = "brown4"
  labs(title = "Days of Major Precipitation vs. Season", x = "Season", y = "Count", fill = "Season") +
  theme_minimal()
```



Using the above graphic, we can prove our hypothesis by seeing there are many more days of extreme precipitation during the summer with lots of rainfall. While in the winter there are fewer days but the numbers will be much more extreme on the scale of feet of snow.

While I didn't have any specific idea I wanted to prove using this data, we have formed a story of how, why and how much precipitation occurs in the various seasons.