DataSci 306, Homework 4

Max Han, maxhan

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

You all may be watching the news lately on 'Helene' and it may be interesting to look into storms data again! So let us do some more EDA on the storms dataset.

Note: Full points are only given to charts that are perfect in all aspects (the title is appropriate, x/y labels make sense, label values not overlapping, etc.)

Problem 1 (1.5 points)

Identifying and imputing missing values

Identify NA values (0.5 point)

Are there any missing values in the dataset? Which are the columns that have missing values?

```
sum(is.na(storms))
```

```
## [1] 33758
```

```
colSums(is.na(storms))
```

```
##                        name                        year
##                           0                           0
##                       month                         day
##                           0                           0
##                        hour                         lat
```

```
##                         0                         0
##                      long                    status
##                         0                         0
##                  category                      wind
##                     14734                         0
##                  pressure tropicalstorm_force_diameter
##                         0                      9512
##  hurricane_force_diameter
##                      9512
```

Imputing Missing Values (a) (0.5 points)

If you have to decide on some value for imputing the NA values in `hurricane_force_diameter` column, what value would you choose? I will replace NA value with mean of `hurricane_force_diameter`

```
storms$hurricane_force_diameter[is.na(storms$hurricane_force_diameter)] <- mean(storms$hurricane_force_
colSums(is.na(storms))
```

```
##                         name                         year
##                            0                            0
##                        month                          day
##                            0                            0
##                         hour                          lat
##                            0                            0
##                         long                       status
##                            0                            0
##                     category                         wind
##                        14734                            0
##                     pressure tropicalstorm_force_diameter
##                            0                         9512
##  hurricane_force_diameter
##                            0
```

Imputing Missing Values (b) (0.5 points)

If you now analyze the `category` of the storm and its corresponding `hurricane_force_diameter`, what would be the value you would choose to impute for `hurricane_force_diameter` if the category of that storm is 'NA'? Justify your answer. Then write a statement to impute the missing values for `hurricane_force_diameter` with your chosen value.

Hint: Look into the documentation (`?storms`) to find out what it means if the category is 'NA' and then also analyze the data to finally decide on the correct value to impute.

```
storms$hurricane_force_diameter[is.na(storms$category)] <- 0
colSums(is.na(storms))
```

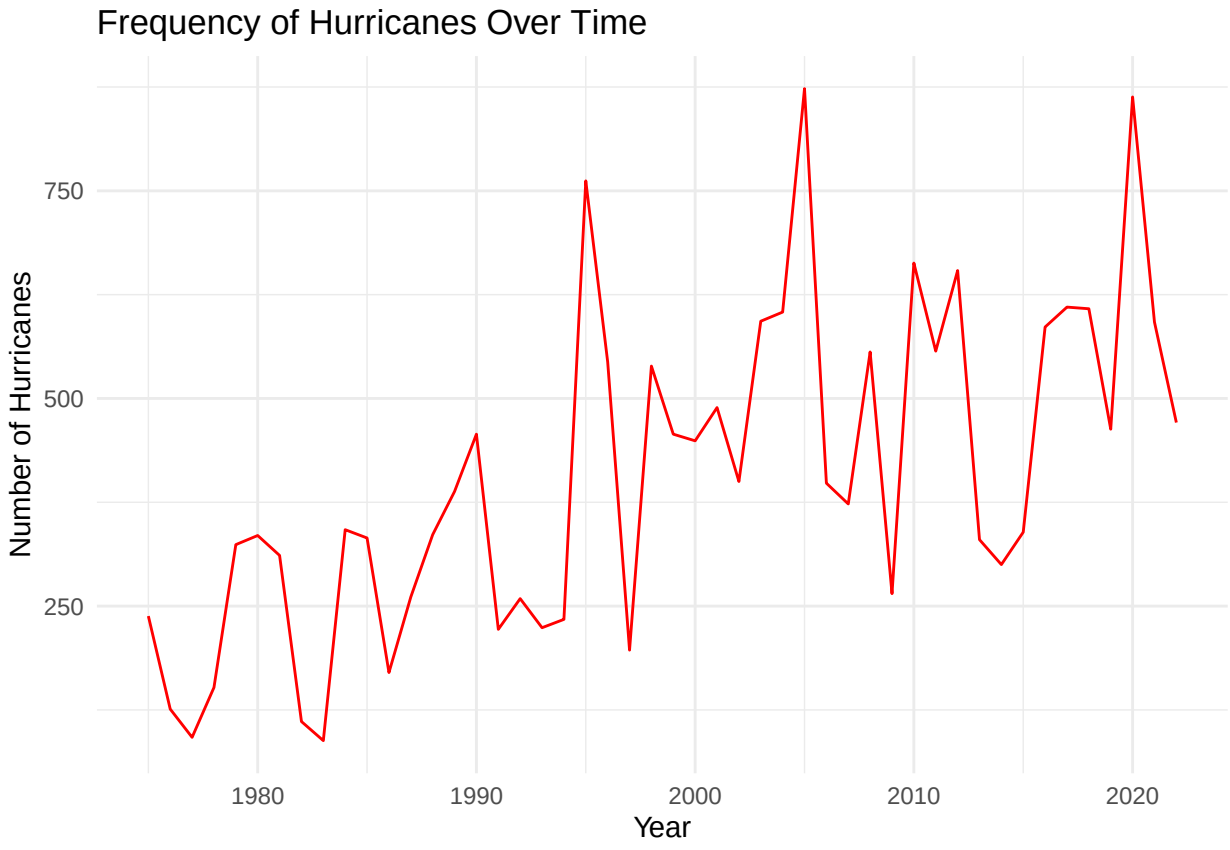
```
##                         name                         year
##                            0                            0
##                        month                          day
##                            0                            0
##                         hour                          lat
##                            0                            0
##                         long                       status
##                            0                            0
##                     category                         wind
##                        14734                            0
##                     pressure tropicalstorm_force_diameter
##                            0                         9512
```

```
##     hurricane_force_diameter
##                            0
```

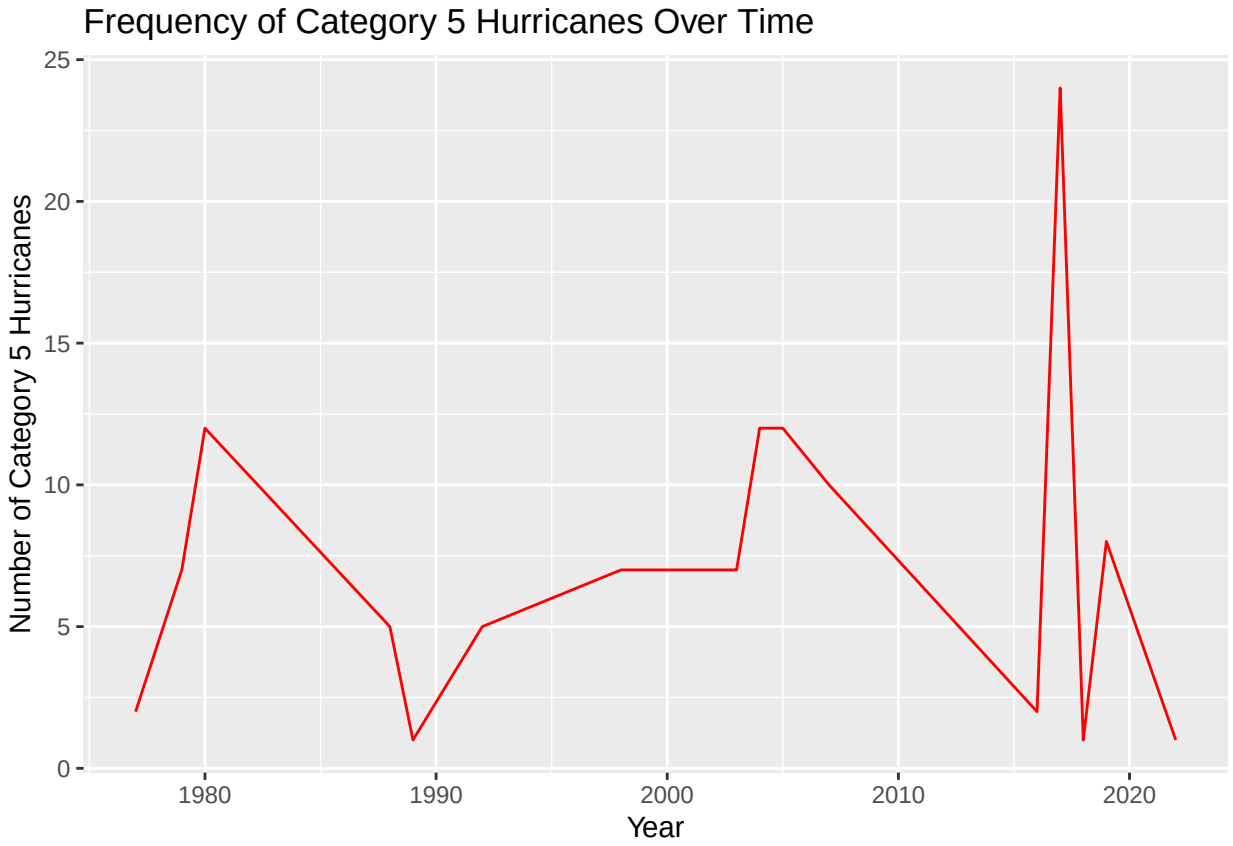
Problem 2 (1 point)

How does the frequency of hurricanes vary over time? Is the category 5 hurricanes increasing over the years? Are there any noticeable trends or patterns in storm occurrence over the years? Please provide your insights with a suitable chart(s)

```
storm_counts <- storms |>
  group_by(year) |>
  summarize(total_storms = n())

storm_counts |> ggplot(aes(x = year, y = total_storms)) +
  geom_line(color = "red") +
  labs(title = "Frequency of Hurricanes Over Time", x = "Year", y = "Number of Hurricanes") +
  theme_minimal()
```




```
category_5_counts <- storms |>
  filter(category == 5) |>
  group_by(year) |>
  summarize(category_5_storms = n())

ggplot(category_5_counts, aes(x = year, y = category_5_storms)) +
  geom_line(color = "red") +
  labs(title = "Frequency of Category 5 Hurricanes Over Time", x = "Year", y = "Number of Category 5 Hu
```



Problem 3 (1 point)

Using the other pertinent columns, create a new column called `date`. Using this `date` column, write code to find the days that have the highest number of unique storms (i.e., unique storm names) recorded.

Hint: look into the *make_datetime* function

```
storms1 <- storms |>
  mutate(date = make_datetime(year, month, day))

storms_date <- storms1 |>
  group_by(date) |>
  summarize(unique_frequency = n_distinct(name)) |>
  arrange(desc(unique_frequency))

storms_date
```

```
## # A tibble: 3,678 x 2
##    date                unique_frequency
##    <dttm>                         <int>
##  1 2020-09-17 00:00:00                6
##  2 2020-09-18 00:00:00                6
##  3 1992-09-26 00:00:00                5
##  4 1995-08-28 00:00:00                5
##  5 2020-09-14 00:00:00                5
##  6 2020-09-19 00:00:00                5
##  7 2022-09-26 00:00:00                5
##  8 1980-09-06 00:00:00                4
```

```
##  9 1980-09-07 00:00:00                4
## 10 1981-09-11 00:00:00                4
## # i 3,668 more rows
```

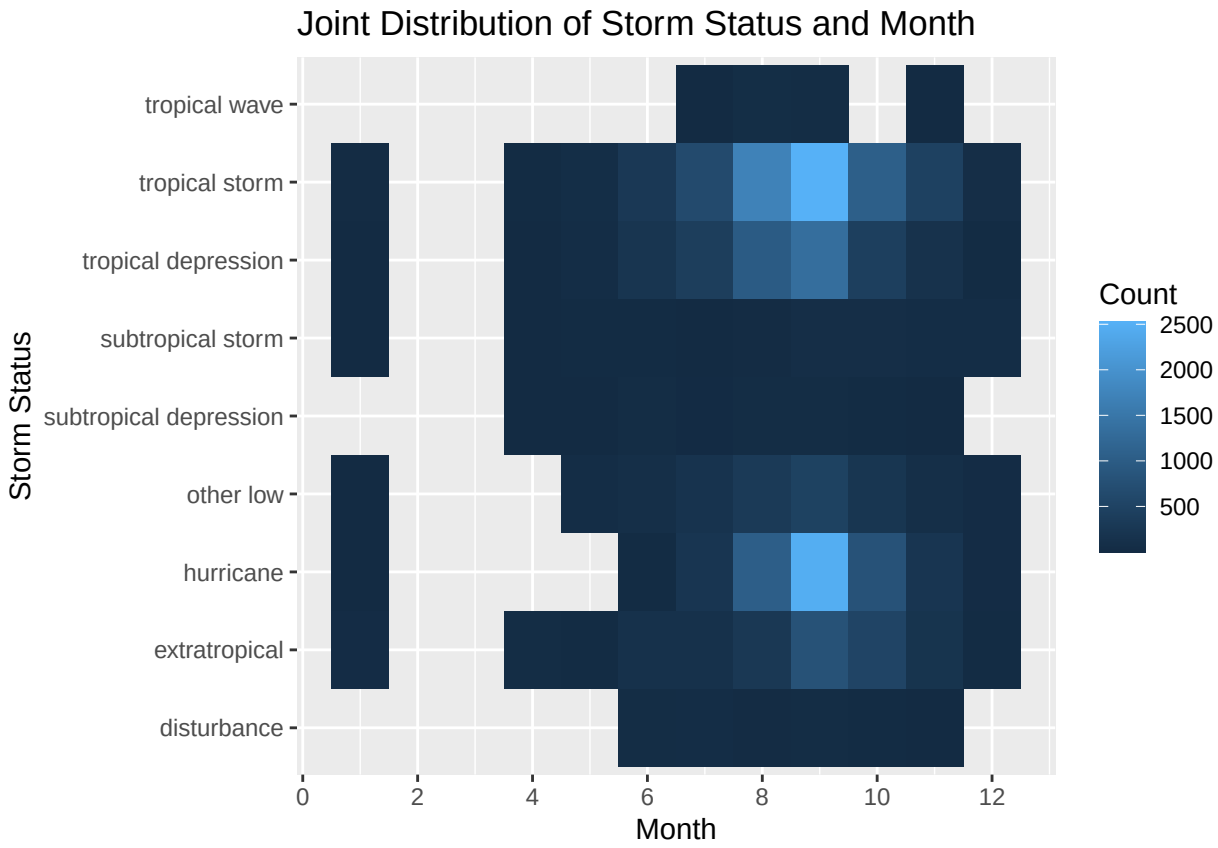
Problem 4 (1 points)

Plot a joint distribution of storm status and month. Based on this visualization, identify the months that are not ideal to travel

```
storm_status_by_month <- storms |>
  group_by(status, month) |>
  summarize(count = n(), .groups = "drop") |>
  ungroup()

storm_status_by_month
```

```
## # A tibble: 75 x 3
##    status        month count
##    <fct>         <dbl> <int>
##  1 disturbance       6    35
##  2 disturbance       7    46
##  3 disturbance       8    25
##  4 disturbance       9    41
##  5 disturbance      10    16
##  6 disturbance      11     8
##  7 extratropical     1    29
##  8 extratropical     4    40
##  9 extratropical     5    18
## 10 extratropical     6   130
## # i 65 more rows
```

```
ggplot(storm_status_by_month, aes(x = month, y = status, fill = count)) +
  geom_tile() +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 6)) +
  labs(title = "Joint Distribution of Storm Status and Month",
       x = "Month",
       y = "Storm Status",
       fill = "Count")
```

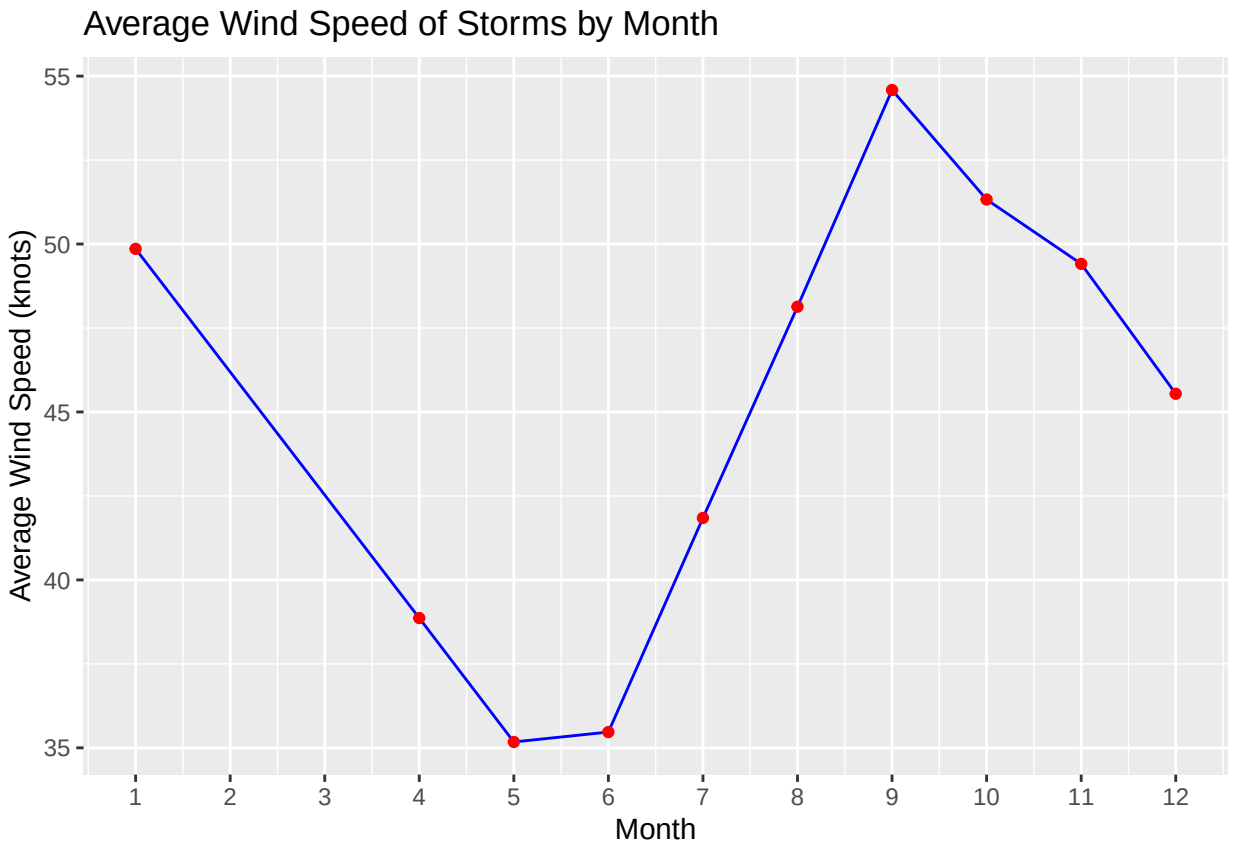


Based on the plot, it is not idea to travel from July to October.

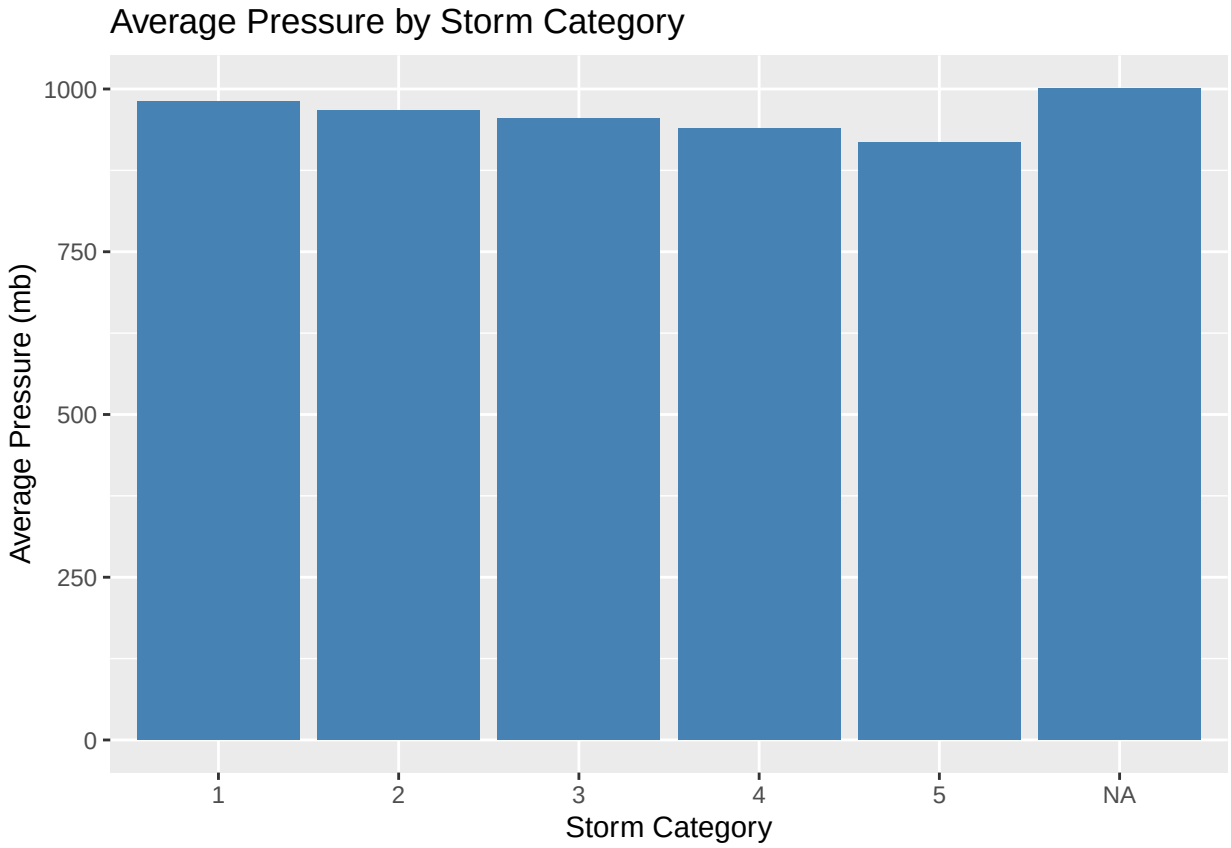
Problem 5 (1 points)

As an analyst, it's crucial to develop proficiency in posing good questions and answering them. Pose two additional questions that could pique your curiosity, then plot meaningful charts to answer these questions and share your perspectives.

```
# How does the average wind speed of storms vary by month?

average_wind_speed_by_month <- storms |>
  group_by(month) |>
  summarize(avg_wind_speed = mean(wind, na.rm = TRUE))

ggplot(average_wind_speed_by_month, aes(x = month, y = avg_wind_speed)) +
  geom_line(color = "blue") +
  geom_point(color = "red") +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 12)) +
  labs(title = "Average Wind Speed of Storms by Month", x = "Month", y = "Average Wind Speed (knots)")
```




```r
# Which storm category has the highest average pressure, and how does it compare across categories?

average_pressure_by_category <- storms |>
  group_by(category) |>
  summarize(avg_pressure = mean(pressure, na.rm = TRUE))

ggplot(average_pressure_by_category, aes(x = factor(category), y = avg_pressure)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Average Pressure by Storm Category", x = "Storm Category", y = "Average Pressure (mb)")
```



Problem 6 (2 points)

Find correlation coefficients between the various attributes of the storm and plot a heatmap (geom_tile) that shows the negative, positive, and no correlation with different colors.

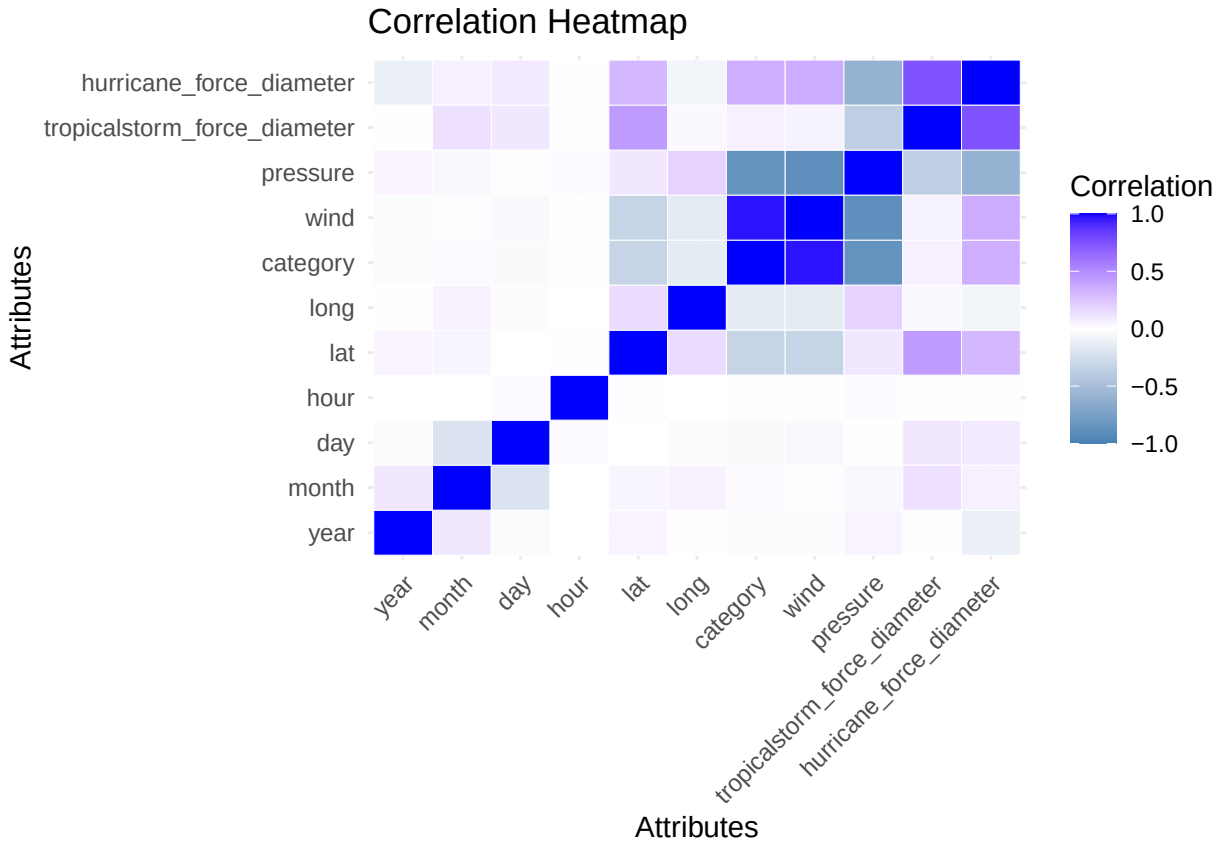
Hint: Once you find the correlation coefficients, you may use the `melt` method to reshape the values

```
numeric_columns <- storms |>
  select(is.numeric)
```

```
## Warning: Use of bare predicate functions was deprecated in tidyselect 1.1.0.
## i Please use wrap predicates in `where()` instead.
##   # Was:
##   data %>% select(is.numeric)
##
##   # Now:
##   data %>% select(where(is.numeric))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
correlation_matrix <- cor(numeric_columns, use = "complete.obs")
correlation_melted <- melt(correlation_matrix)

ggplot(data = correlation_melted, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "steelblue", high = "blue", mid = "white",
                       midpoint = 0, limit = c(-1, 1), space = "Lab",
```

```
                      name = "Correlation") +
  theme_minimal() +
  labs(title = "Correlation Heatmap", x = "Attributes", y = "Attributes") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

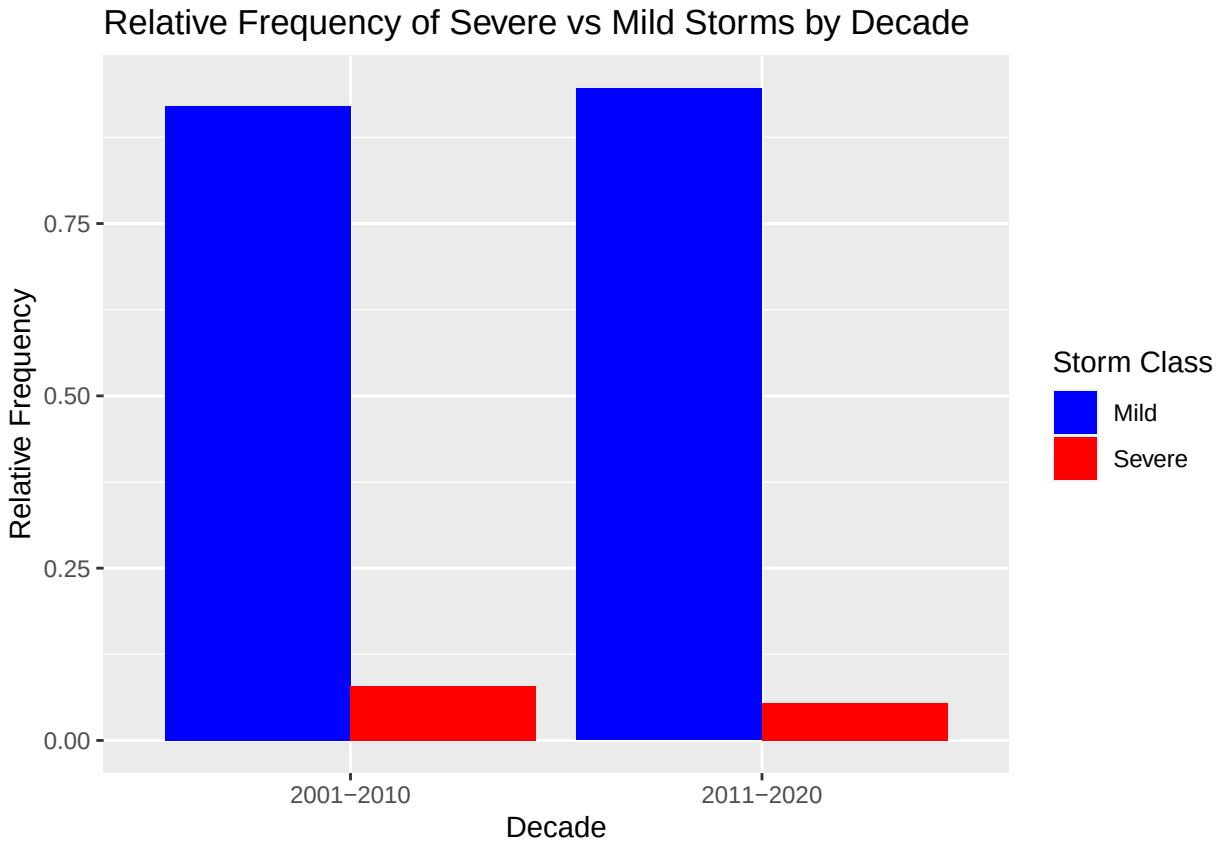


Problem 7 (2.5 points)

Create a bar chart that shows the relative frequency of 'severe storms' (i.e., category 3, 4 and 5 are considered severe for our exercise) vs 'mild storms' (everything else is classified as mild for this exercise) for the last two decades. Consider 2001 to 2010 as the first decade and 2011 to 2020 as the second decade for this measurement. We need to see the relative frequency chart with decades on the 'x' axis. Refer eBook link here: https://r4ds.hadley.nz/data-visualize.html#two-categorical-variables (1.5 points)

```
storms <- storms |>
  mutate(storm_class = ifelse(category %in% c(3, 4, 5), "Severe", "Mild"),
         decade = ifelse(year >= 2001 & year <= 2010, "2001-2010",
                         ifelse(year >= 2011 & year <= 2020, "2011-2020", NA))) %>%
  filter(!is.na(decade))

storm_counts <- storms |>
  group_by(decade, storm_class) |>
  summarize(count = n(), .groups = "drop") |>
  ungroup() |>
  group_by(decade) |>
  mutate(relative_frequency = count / sum(count))

ggplot(storm_counts, aes(x = decade, y = relative_frequency, fill = storm_class)) +
```

```
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Relative Frequency of Severe vs Mild Storms by Decade",
       x = "Decade", y = "Relative Frequency", fill = "Storm Class") +
  scale_fill_manual(values = c("Severe" = "red", "Mild" = "blue"))
```



Challenge - Hypothesis testing (1 pt)

Then answer, is the category 3, 4 and 5 storms (i.e, severe storms) occurring more often in the last decade? In other words, is the proportion of category 3, 4 and 5 storms from 2001 to 2010 statistically different from proportion of category 3, 4 and 5 storms from 2011 to 2021?

Hint: In STATS 250 you learned how to test this. Use chi-squared test for this analysis

```
storm_table <- storms |>
  group_by(decade, storm_class) |>
  summarize(count = n(), .groups = "drop") |>
  spread(storm_class, count, fill = 0)

contingency_table <- as.matrix(storm_table[, -1])
rownames(contingency_table) <- storm_table$decade

chi_squared_test <- chisq.test(contingency_table)

print(chi_squared_test)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  contingency_table
```

```
## X-squared = 26.791, df = 1, p-value = 2.266e-07
```

The p-value tells use that we have enough evidence there is enough evidence to claim that the proportion of category 3, 4 and 5 storms from 2001 to 2010 statistically different from proportion of category 3, 4 and 5 storms from 2011 to 2021.