Lab Worksheet - Exploratory Data Analysis in airquality Dataset

```
# load libraries:
library(tidyverse) # Recall that tidyverse contains dplyr, ggplot2, and many
other useful packages.

# Load `airquality` dataset:
data("airquality")
airquality |>
    head()
```

```
  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4   67     5   1
2    36     118  8.0   72     5   2
3    12     149 12.6   74     5   3
4    18     313 11.5   62     5   4
5    NA      NA 14.3   56     5   5
6    28      NA 14.9   66     5   6
```

Q1: Render this quarto file to a pdf. To do so, you can use the hotkey `Ctrl + Shift + K` or `Cmd + Shift + K`. Then, you will see a command appear in the Positron Terminal. Once the command finishes running, you will see a pdf file appear in the same directory as this file and a **VIEWER** window will display the pdf on the right pane of Positron. Here you should see the rendered contents of this file.

Q2: Convert the `Month` column to a factor with it's levels labeled as month names (e.g. "May", "June", etc.).

```
# your code here
airquality <- airquality |>
  mutate(
    Month = factor(
      Month,
      levels = 5:9,  # original numeric codes
      labels = c("May", "June", "July", "August", "September")  # names to use
    )
  )
```

Q3: Create a new column `Temp_C` that converts the temperature from Fahrenheit to Celsius.

```
# your code here
airquality <- airquality |>
  mutate(
    Temp_C = (Temp - 32) * 5/9
  )
```

Q4: Rename the `Wind` column to `Wind_mph` and the Temp column to `Temp_F`.

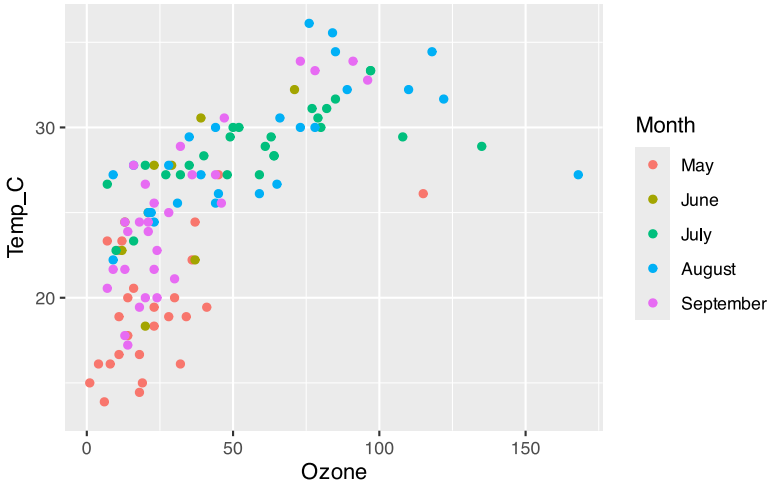
```
# your code here
airquality <- airquality |>
    rename(
        Wind_mph = Wind,
       Temp_F = Temp
    )

# airquality <- airquality |>
#    mutate(
#        Wind_mph = Wind,
#       Temp_F = Temp,
#    select(-wind, -Temp)
#    )
```

Q5: Create a scatter plot of `Ozone` vs `Temp_C`, colored by `Month`. Add appropriate axis labels and a title.

```
# your code here
airquality |>
    ggplot(aes(x = Ozone,
               y = Temp_C,
               color = Month)) +
    geom_point()
```

```
Warning: Removed 37 rows containing missing values or values outside the scale
range
(`geom_point()`).
```



Q6: Create a violin plot of `Ozone` levels for each month. Add appropriate axis labels and a title.

```
# your code here
```

Q7: Create a segmented bar chart of the proportion of days with `Ozone` levels above 50 by `Month`. Remove NA values in the `Ozone` column. Add appropriate axis labels and a title.

```
# your code here
```

Q8: Create a stacked bar chart showing the count of days available for each month. Fill bar colors by `Ozone` levels above or below 50 for each month. Remove NA values in the `Ozone` column. Add appropriate axis labels and a title.

```
# your code here
```

Q9: What is the utility of using a segmented bar chart vs a regular bar chart in this context? Which do you think is more informative and why?

your answer here

Q10: Create a summary table of the number and proportion of NA values in each column.

```
# your code here
airquality |>
```

```
    summarize(
        across(everything(), ~ sum(is.na( . )) # ~ style of writing
( . )=Placeholder where cols go
        )
    )  |>
    pivot_longer(everything(), names_to = "column_name", values_to =
"number_of_na_vals")
```

```
# A tibble: 7 × 2
  column_name number_of_na_vals
  <chr>                   <int>
1 Ozone                      37
2 Solar.R                     7
3 Wind_mph                    0
4 Temp_F                      0
5 Month                       0
6 Day                         0
7 Temp_C                      0
```

```
        # The Hard Way Manualy:
        # sum_na_vals_Ozone = sum(is.na(Ozone)
        # sum_na_vals_Solar.R = sum(is.na(Solar.R)
```