Boxplot

You can use the geometric object geom_boxplot() from ggplot2 library to draw a box plot. Box plot helps to visualize the distribution of the data by quartile and detect the presence of outliers.
We will use the airquality dataset to introduce box plot with ggplot. This dataset measures the airquality of New York from May to September 1973. The dataset contains 154 observations. We will use the following variables:

- Ozone: Numerical variable
- Wind: Numerical variable
- Month: May to September. Numerical variable

Create Box Plot

Before you start to create your first box plot, you need to manipulate the data as follow:

- Step 1: Import the data
- Step 2: Drop unnecessary variables
- Step 3: Convert Month in factor level
- Step 4: Create a new categorical variable dividing the month with three level: begin, middle and end.
- Step 5: Remove missing observations

All these steps are done with dplyr and the pipeline operator %>%.

```
library(dplyr)
library(ggplot2)
# Step 1
data_air <- airquality %>%

#Step 2
select(-c(Solar.R, Temp)) %>%
```

```
#Step 3
mutate(Month = factor(Month, order = TRUE, labels = c("May",
"June", "July", "August", "September")),

#Step 4
day_cat = factor(ifelse(Day < 10, "Begin", ifelse(Day < 20,
"Middle", "End"))))
```

A good practice is to check the structure of the data with the function glimpse().

```
glimpse(data_air)
```

Output:

```
## Observations: 153
## Variables: 5
## $ Ozone   <int> 41, 36, 12, 18, NA, 28, 23, 19, 8, NA, 7,
16, 11, 14, ...
## $ Wind    <dbl> 7.4, 8.0, 12.6, 11.5, 14.3, 14.9, 8.6,
13.8, 20.1, 8.6...
## $ Month   <ord> May, May, May, May, May, May, May, May,
May, May, May,...
## $ Day     <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
14, 15, 16,...
## $ day_cat <fctr> Begin, Begin, Begin, Begin, Begin, Begin,
Begin, Begi...
```

There are NA's in the dataset. Removing them is wise.

```
# Step 5
data_air_nona <-data_air %>% na.omit()
```

Basic box plot

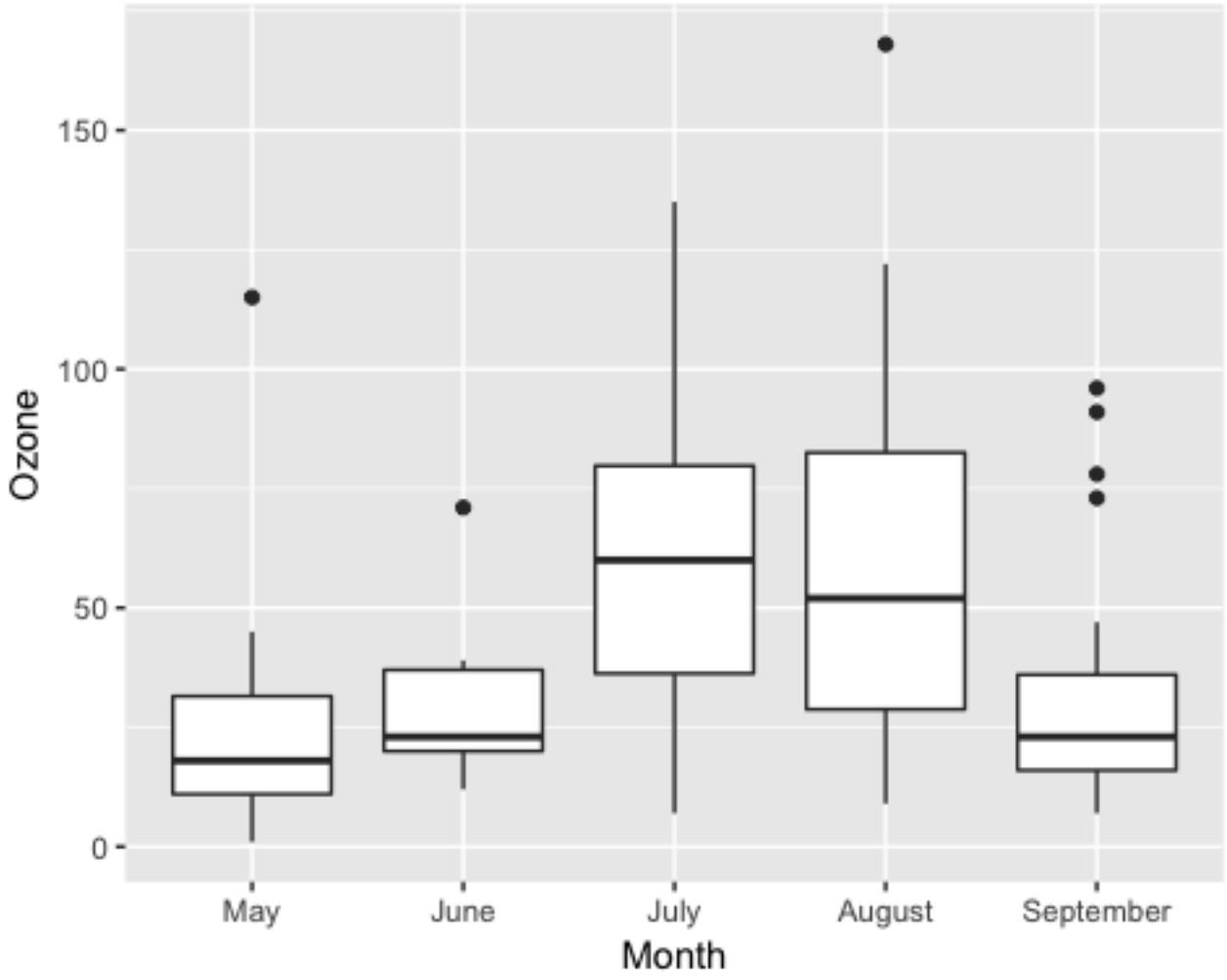
Let's plot the basic box plot with the distribution of ozone by month.

```
# Store the graph
box_plot <- ggplot(data_air_nona, aes(x = Month, y = Ozone,
fill= ))
# Add the geometric object box plot
box_plot +
    geom_boxplot()
```

Code Explanation

- Store the graph for further use
    - box_plot: You store the graph into the variable box_plot It is helpful for further use or avoid too complex line of codes
- Add the geometric object box plot
    - You pass the dataset data_air_nona to ggplot.
    - Inside the aes() argument, you add the x-axis and y-axis.
    - The + sign means you want R to keep reading the code. It makes the code more readable by breaking it.
    - Use geom_boxplot() to create a box plot

Output:



Change side of the graph

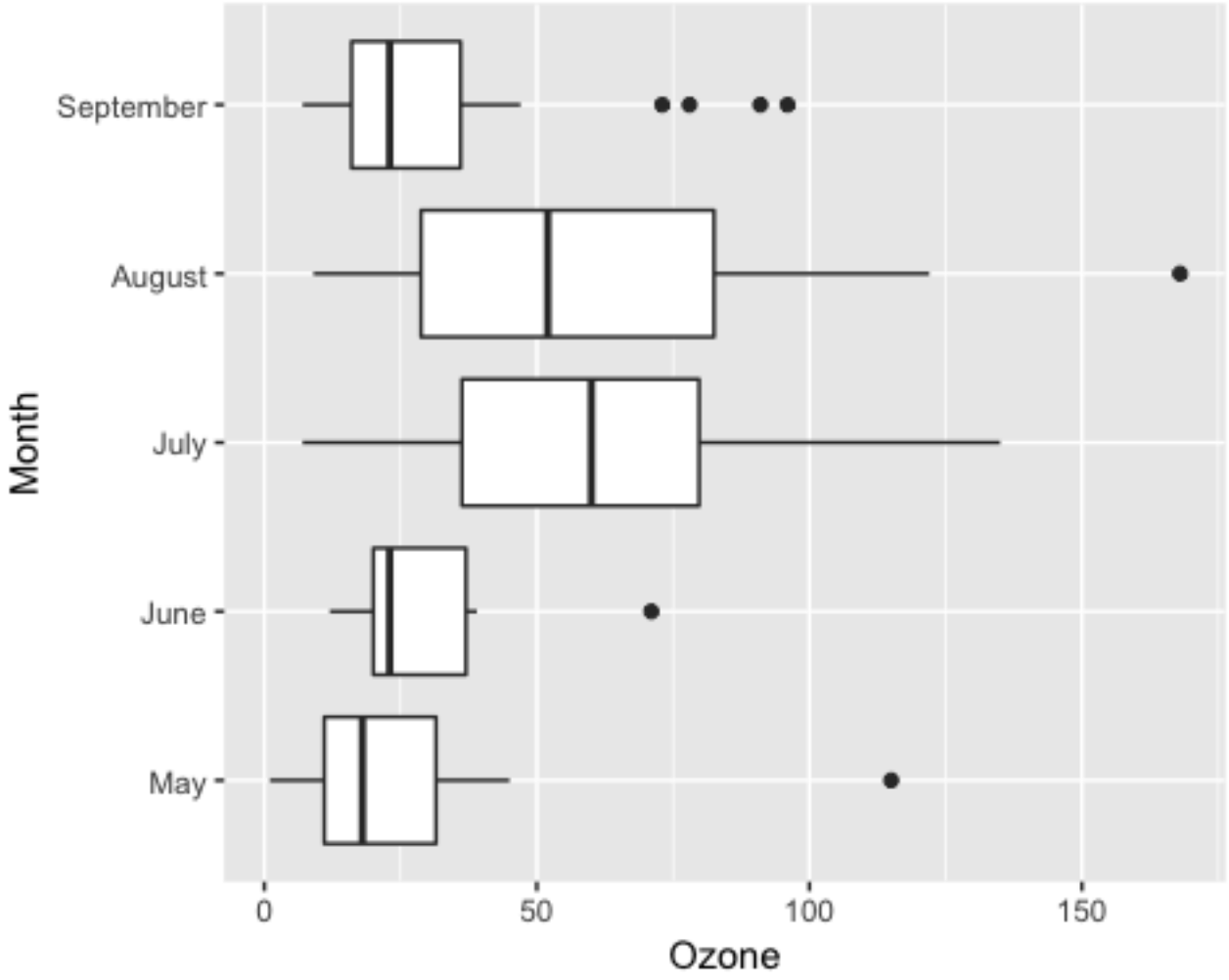
You can flip the side of the graph.

```
box_plot +
  geom_boxplot()+
  coord_flip()
```

Code Explanation

- box_plot: You use the graph you stored. It avoids rewriting all the codes each time you add new information to the graph.
- geom_boxplot(): Create the box plot
- coord_flip(): Flip the side of the graph

Output:



Change color of outlier

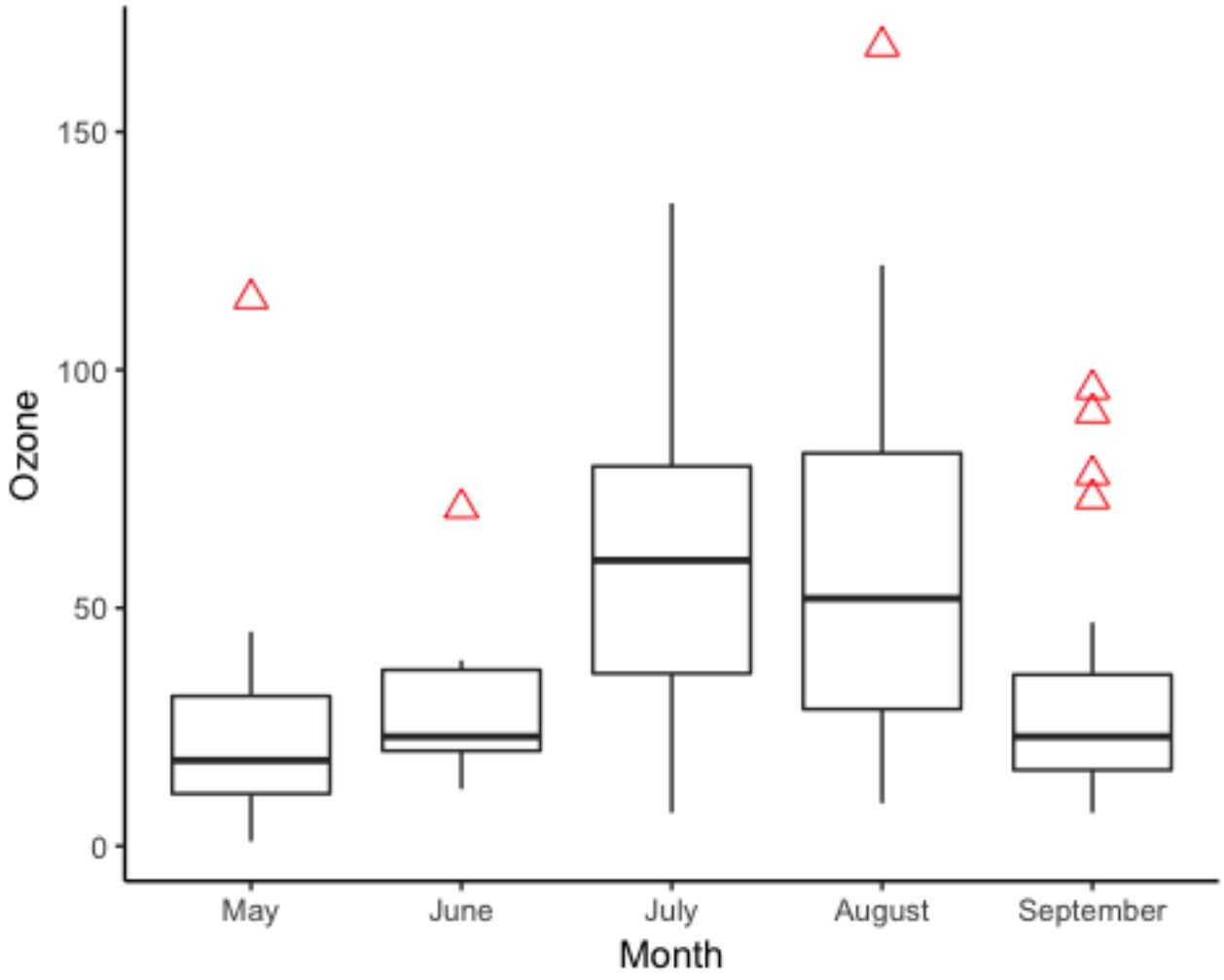
You can change the color, shape and size of the outliers.

```
box_plot +
    geom_boxplot(outlier.colour = "red",
        outlier.shape = 2,
        outlier.size = 3) +
    theme_classic()
```

Code Explanation

- outlier.colour="red": Control the color of the outliers
- outlier.shape=2: Change the shape of the outlier. 2 refers to triangle
- outlier.size=3: Change the size of the triangle. The size is proportional to the number.

Output:



Add a summary statistic

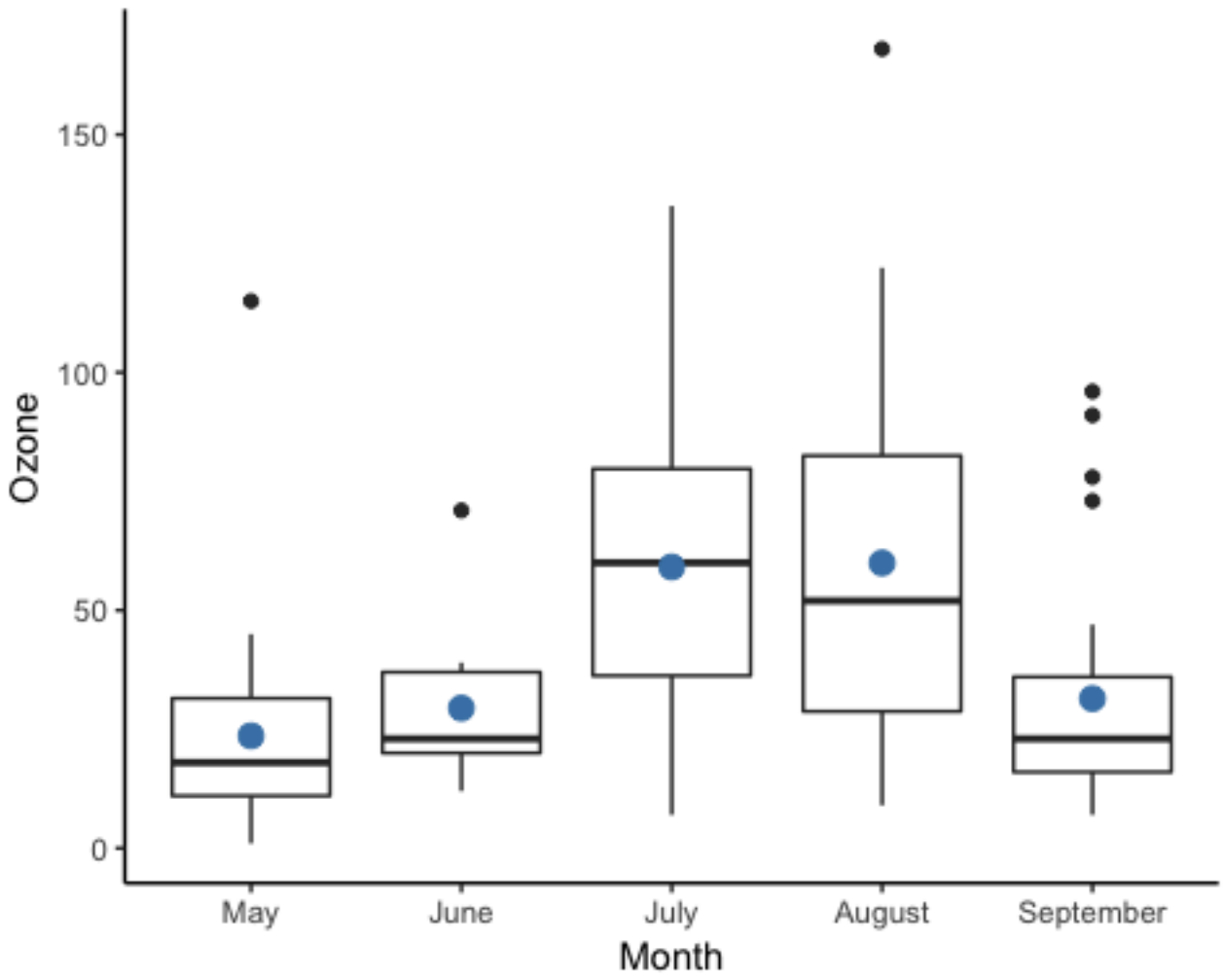
You can add a summary statistic to the box plot.

```
box_plot +
    geom_boxplot() +
    stat_summary(fun.y = mean,
        geom = "point",
        size = 3,
        color = "steelblue") +
    theme_classic()
```

Code Explanation

- stat_summary() allows adding a summary to the box plot
- The argument fun.y controls the statistics returned. You will use mean
- Note: Other statistics are available such as min and max. More than one statistics can be exhibited in the same graph
- geom = "point": Plot the average with a point
- size=3: Size of the point
- color ="steelblue": Color of the points

Output:



Box Plot with Dots

In the next plot, you add the dot plot layers. Each dot represents an observation.

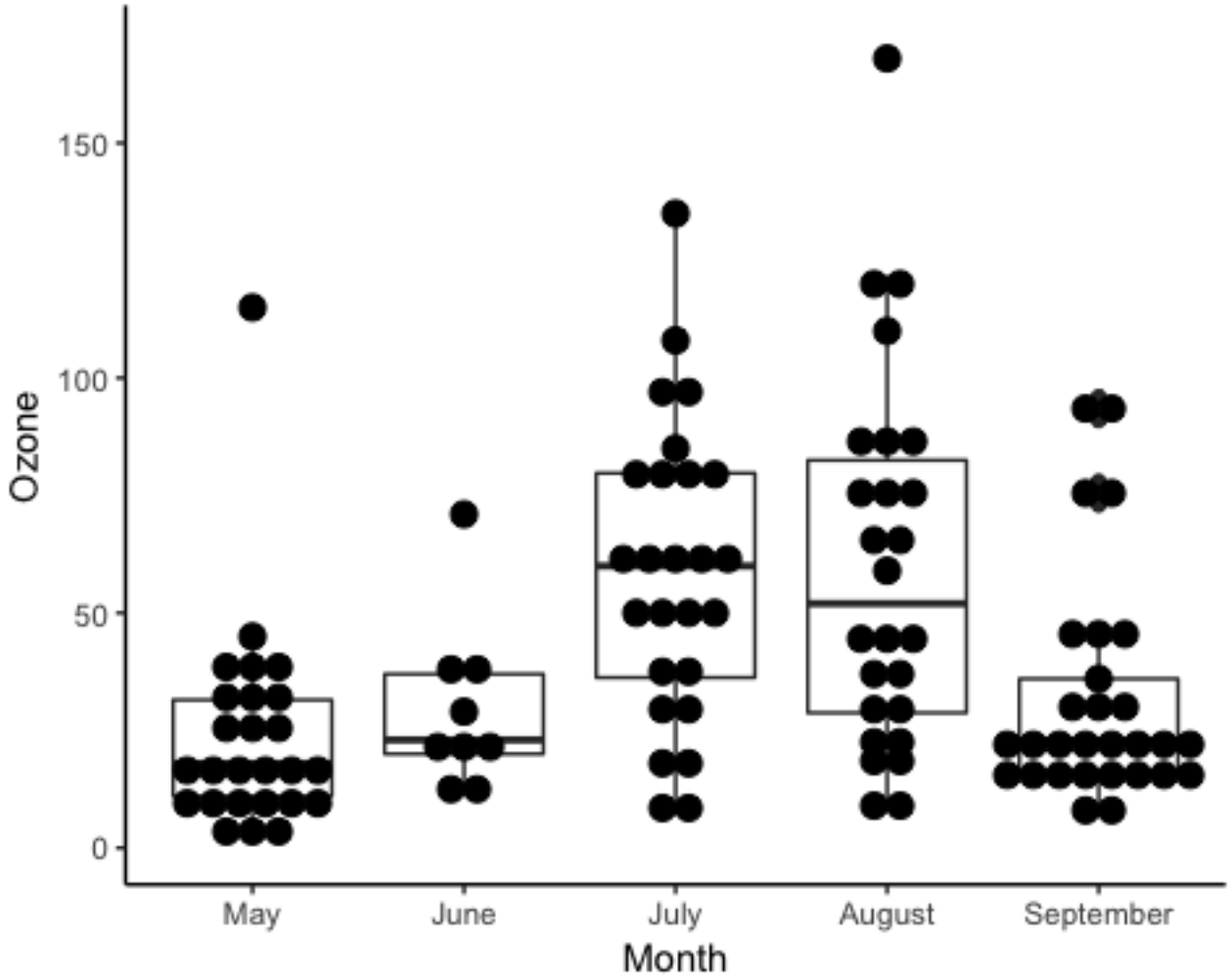
```
box_plot +
    geom_boxplot() +
    geom_dotplot(binaxis = 'y',
        dotsize = 1,
        stackdir = 'center') +
    theme_classic()
```

Code Explanation

- geom_dotplot() allows adding dot to the bin width
- binaxis='y': Change the position of the dots along the y-axis. By default, x-axis
- dotsize=1: Size of the dots
- stackdir='center': Way to stack the dots: Four values:

- "up" (default),
- "down"
- "center"
- "centerwhole"

Output:



Control Aesthetic of the Box Plot

Change the color of the box

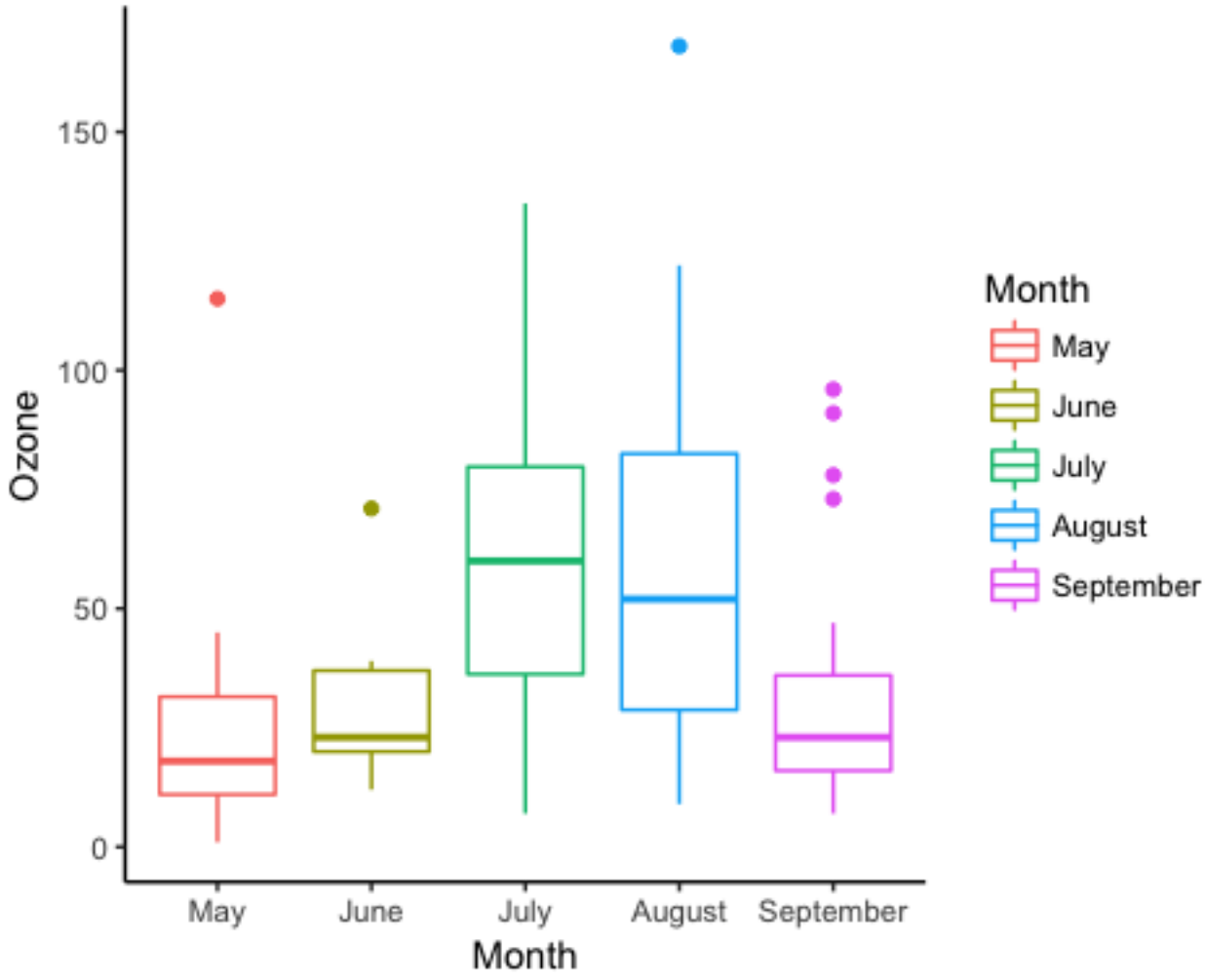
You can change the colors of the group.

```
ggplot(data_air_nona, aes(x = Month, y = Ozone, color = Month)) +
    geom_boxplot() +
    theme_classic()
```

Code Explanation

- The colors of the groups are controlled in the aes() mapping. You can use color= Month to change the color of the box according to the months

Output:



Box plot with multiple groups

It is also possible to add multiple groups. You can visualize the difference in the air quality according to the day of the measure.

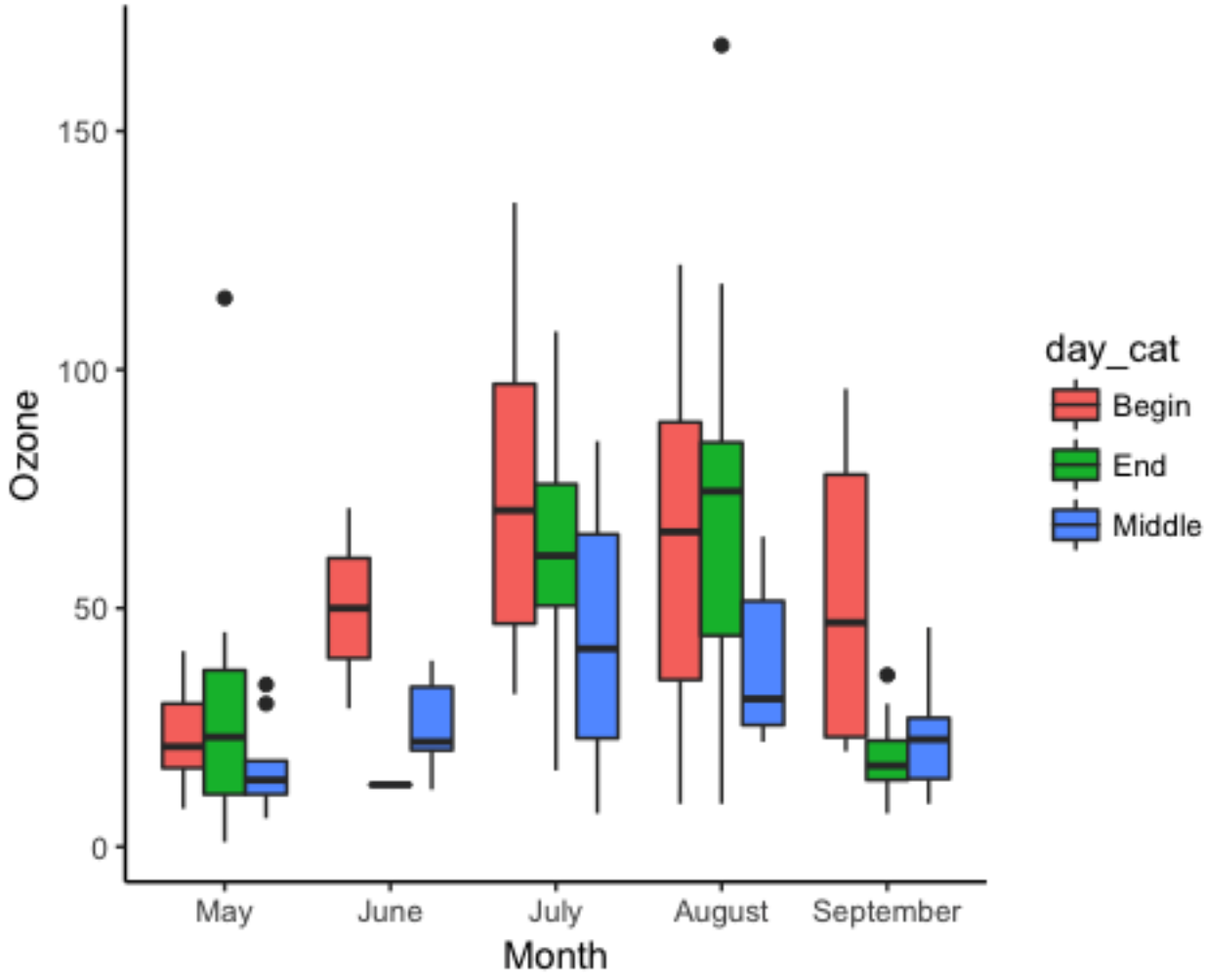
```
ggplot(data_air_nona, aes(Month, Ozone)) +
    geom_boxplot(aes(fill = day_cat)) +
    theme_classic()
```

Code Explanation

- The aes() mapping of the geometric object controls the groups to display (this variable has to be a factor)

- aes(fill= day_cat) allows creating three boxes for each month in the x-axis

Output:



Box Plot with Jittered Dots

Another way to show the dot is with jittered points. It is a convenient way to visualize points with a categorical variable.

This method avoids the overlapping of the discrete data.

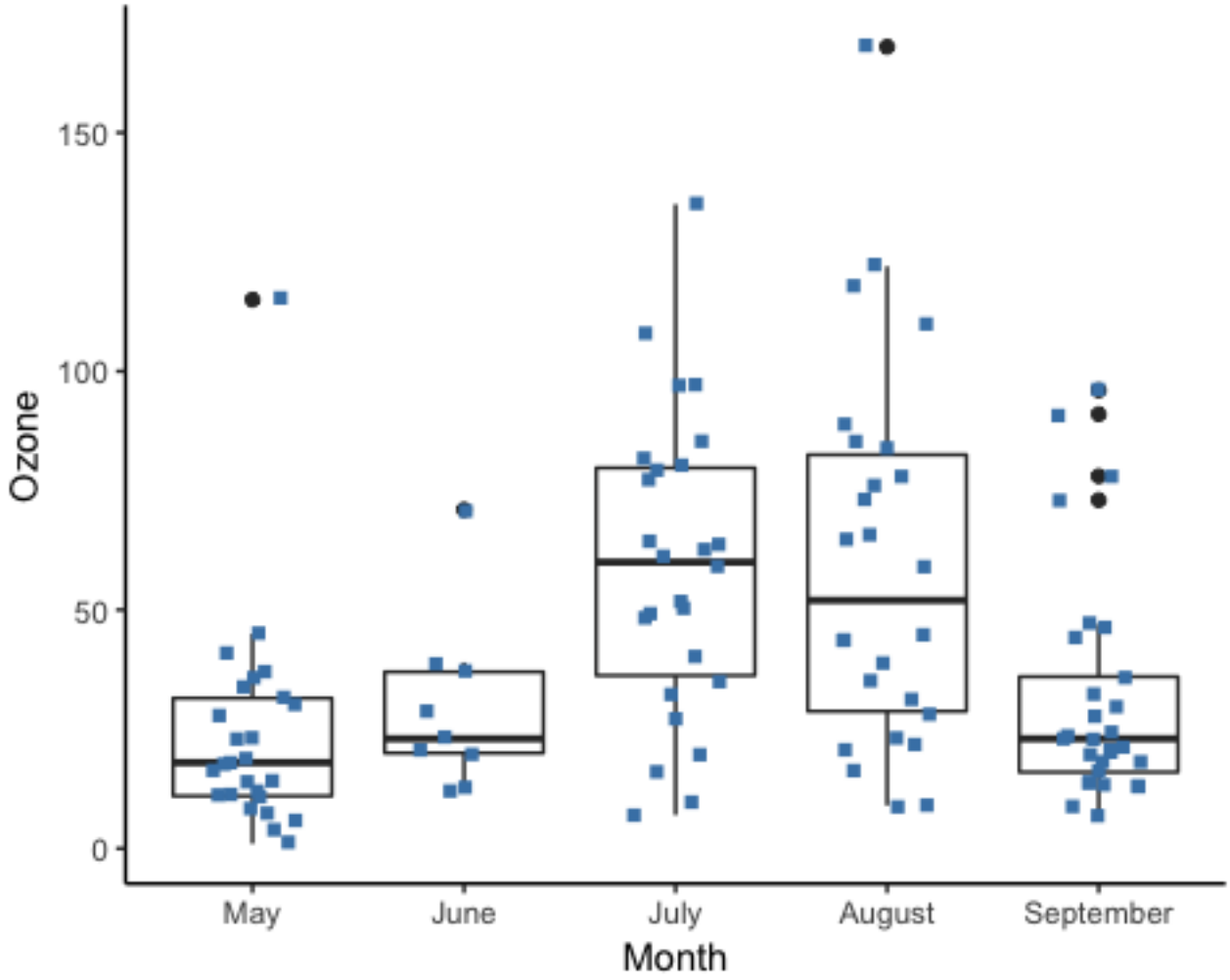
```
box_plot +
    geom_boxplot() +
    geom_jitter(shape = 15,
        color = "steelblue",
        position = position_jitter(width = 0.21)) +
```

```
    theme_classic()
```

Code Explanation

- geom_jitter() adds a little decay to each point.
- shape=15 changes the shape of the points. 15 represents the squares
- color = "steelblue": Change the color of the point
- position=position_jitter(width = 0.21): Way to place the overlapping points. position_jitter(width = 0.21) means you move the points by 20 percent from the x-axis. By default, 40 percent.

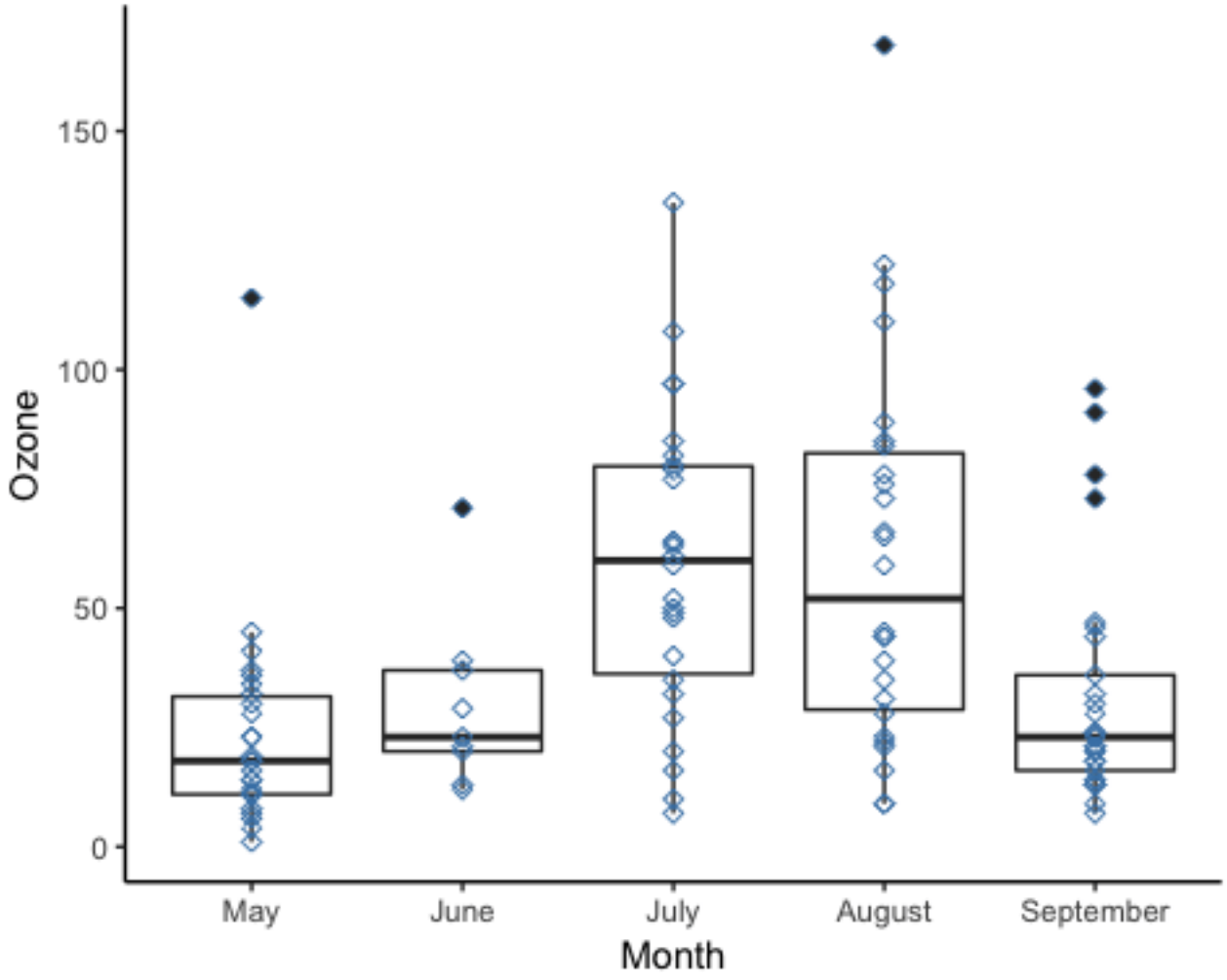
Output:



You can see the difference between the first graph with the jitter method and the second with the point method.

```
box_plot +
    geom_boxplot() +
    geom_point(shape = 5,
```

```
        color = "steelblue") +
    theme_classic()
```



Notched Box Plot

An interesting feature of geom_boxplot(), is a notched box plot. The notch plot narrows the box around the median. The main purpose of a notched box plot is to compare the significance of the median between groups. There is strong evidence two groups have different medians when the notches do not overlap.

```
box_plot +
    geom_boxplot(notch = TRUE) +
    theme_classic()
```

Code Explanation

- geom_boxplot(notch=TRUE): Create a notched box plot

Output: