

MA213 Basic Statistics and Probability - Lab3 Guide

Lab 3: Plotting and Summaries

Learning Objectives

- Classify and Analyze Variables: Categorize variables based on their types (e.g., numerical/categorical, continuous/discrete, ordinal), assess their association (positive, negative, or independent), and determine which make sense as explanatory vs. response variables.
 - Use R for Data Management and Exploration: Utilize R to load, pre-process, and explore data through visualization and summarization techniques.
-

Graphics with `ggplot2` package

Plotting with `ggplot2` package begins with

```
ggplot(data = df, aes(x=x_variable, y=y_variable))
```

where

`df` : your dataframe name,

`aes()` : Aesthetics to define -> specifying which variables are mapped to the x and y axes.

and then you add `geoms` functions – geometrical objects as a graphical representation of the data in the plot (points, lines, bars). `ggplot2` offers many different geoms; We will use a few common ones today, including:

- `geom_point()` : scatter plots, dot plots, etc.
- `geom_line()` : trend lines, time-series, etc.
- `geom_histogram()` : histograms

In short,

Complete the template below to build a graph.

```
ggplot (data = <DATA>) +  
  <GEOM_FUNCTION> (mapping = aes(<MAPPINGS>),  
    stat = <STAT>, position = <POSITION>) +  
  <COORDINATE_FUNCTION> +  
  <FACET_FUNCTION> +  
  <SCALE_FUNCTION> +  
  <THEME_FUNCTION>
```

required

Not required, sensible defaults supplied

Plots to show New York Air Quality Data

We want to examine the shape of `Ozone` to see whether the data is symmetric, skewed or how the mean is centered and where...? Let's see how this `Ozone` data is shaped by using several different methods.

Import the Data

```
df = read.csv("Lab3/airquality.csv")
```

1. Summary of `Ozone`.

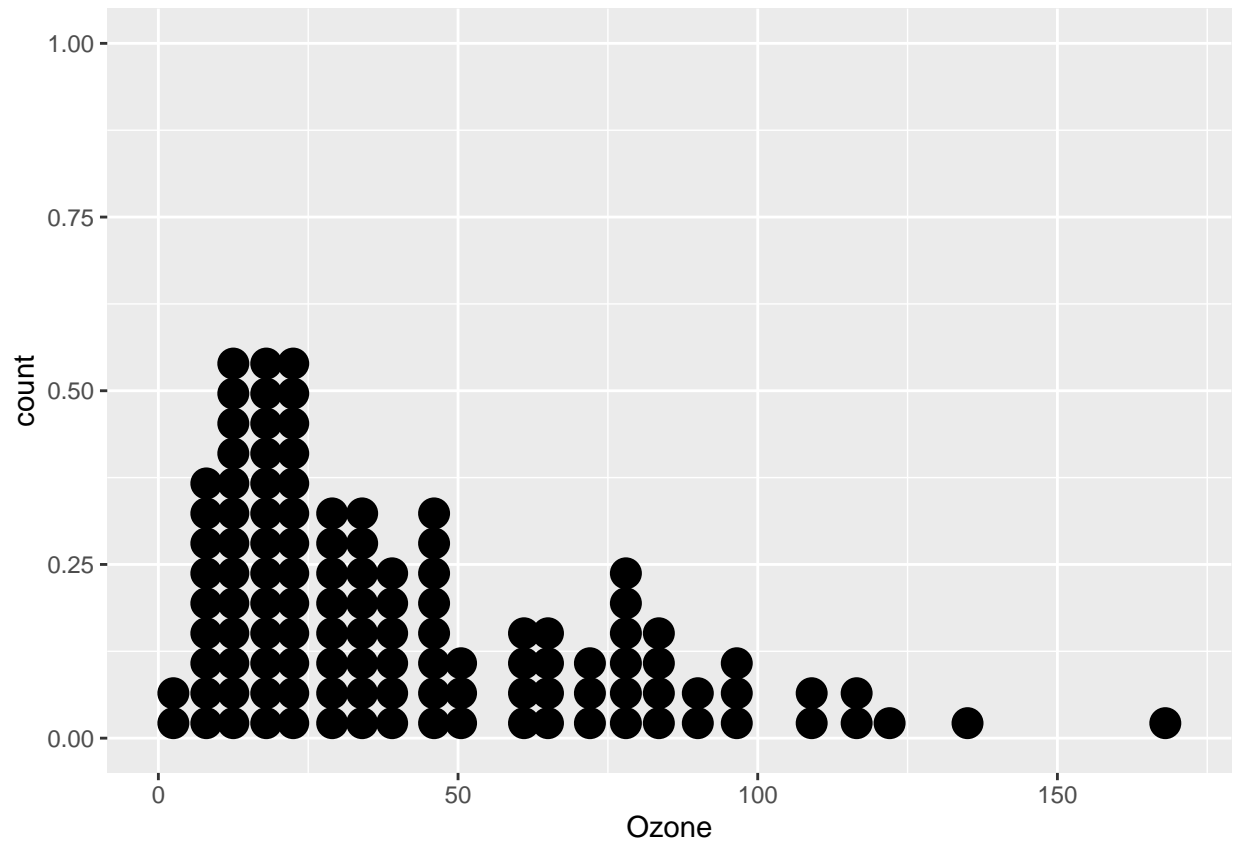
```
summary(df$Ozone)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's  
##      1.00   18.00   31.50   42.13   63.25   168.00       37
```

2. Stacked dot plot of `Ozone` .

```
ggplot(data = df, aes(Ozone)) +  
  geom_dotplot(binwidth = 5) # bins of width 5
```

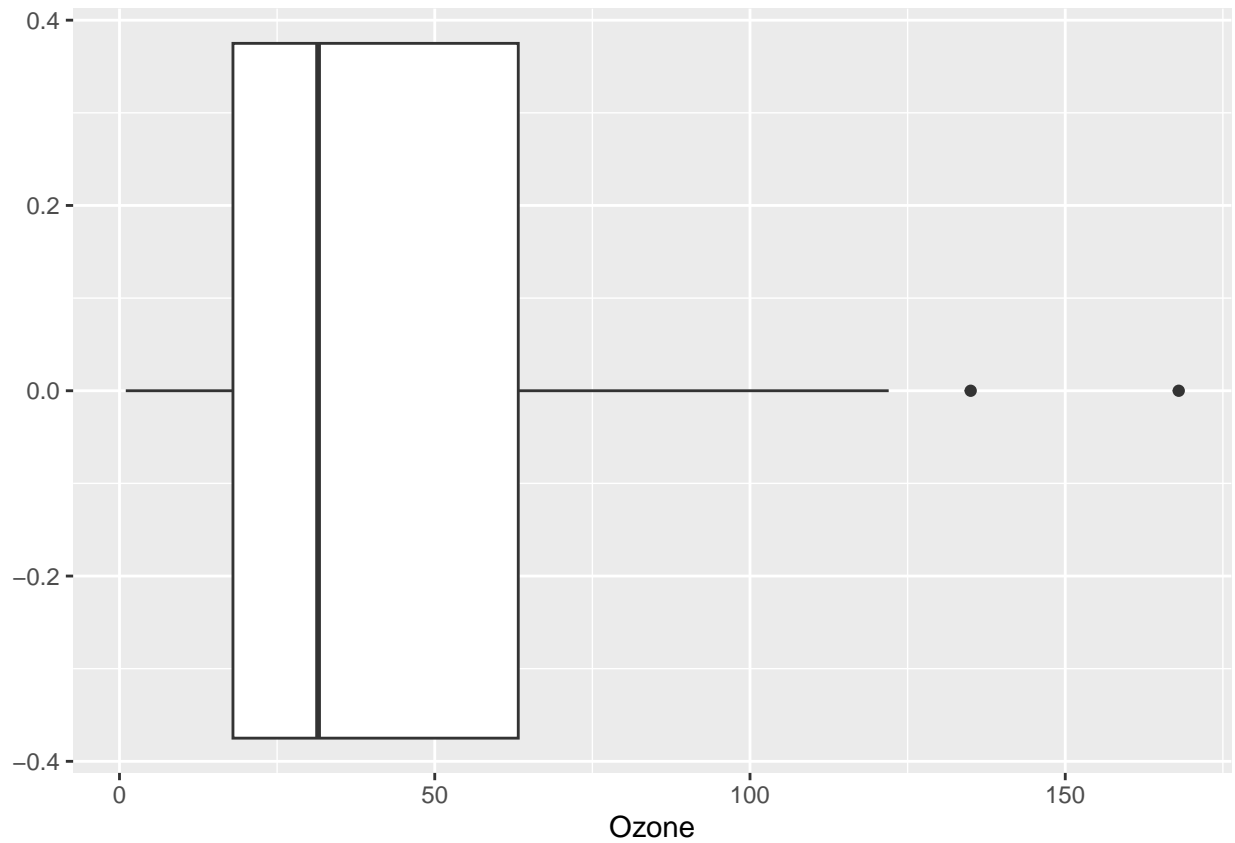
```
## Warning: Removed 37 rows containing missing values  
## or values outside the scale range  
## (`stat_bindot()`).
```



3. Box plot of Ozone.

```
ggplot(data=df, aes(Ozone)) +  
  geom_boxplot()
```

```
## Warning: Removed 37 rows containing non-finite  
## outside the scale range  
## (`stat_boxplot()`).
```



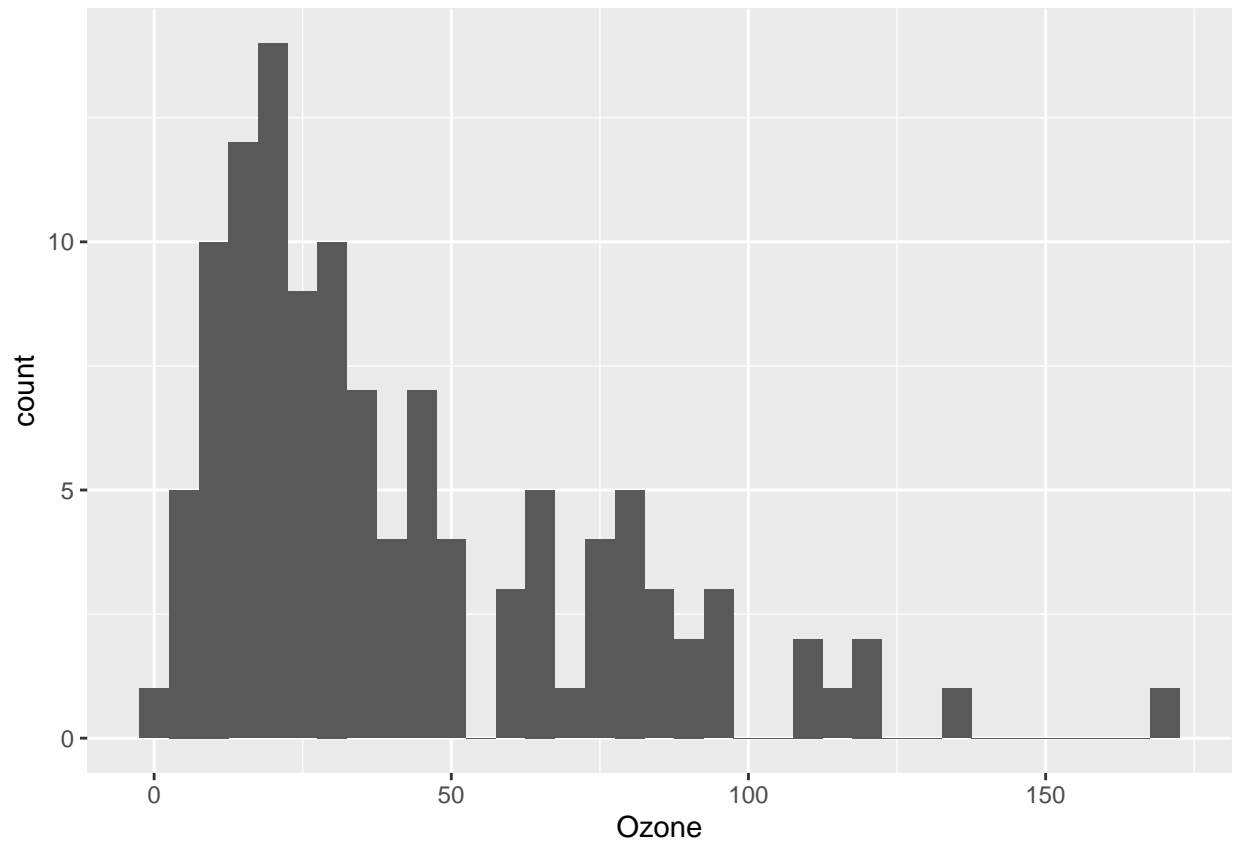
This shows that there are two suspected outliers.

4. Histogram of Ozone .

```
# ggplot version

ggplot(data=df ) +
  geom_histogram(mapping=aes(Ozone), binwidth = 5)

## Warning: Removed 37 rows containing non-finite
## outside the scale range (`stat_bin()`).
```

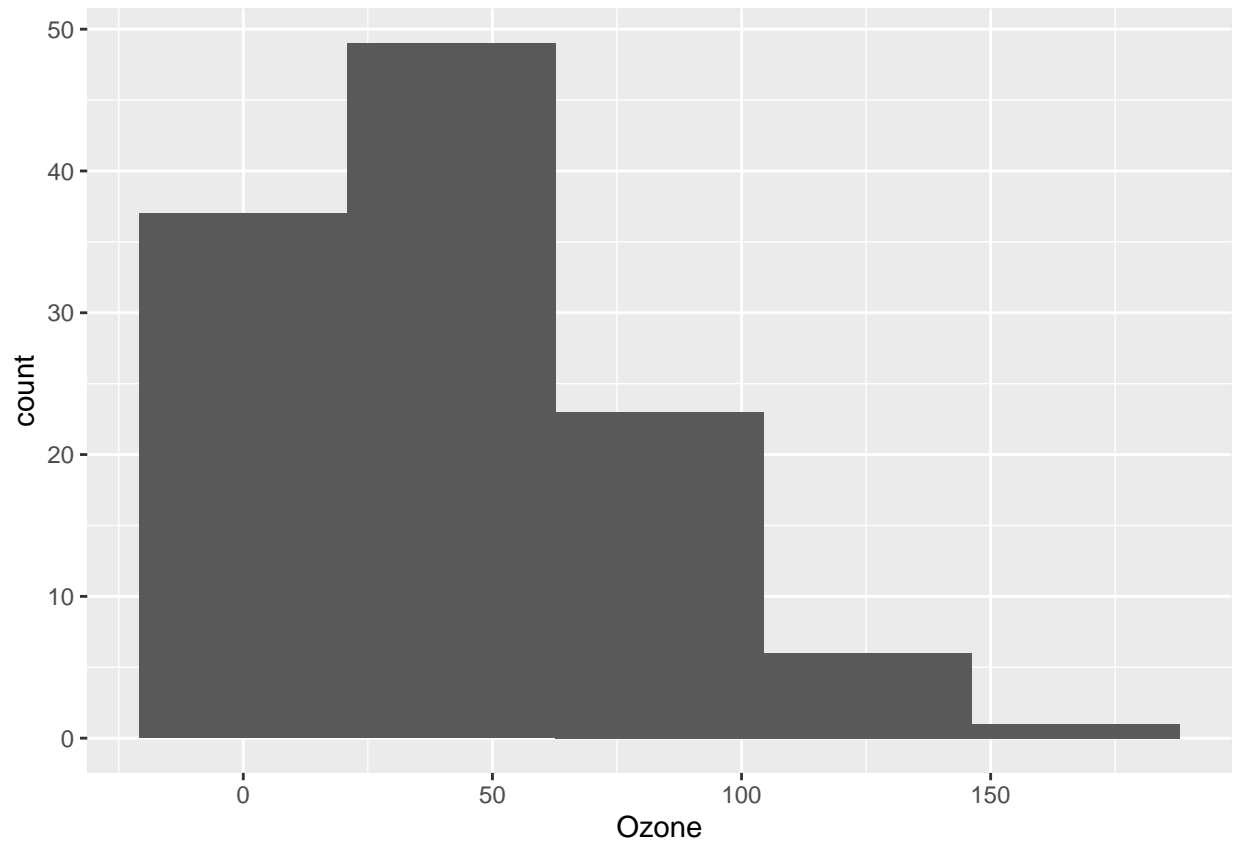


5. Histogram with different bin option

by choosing how many observations in each bin (e.g. `bins=5` means each bin contains 5 observations).

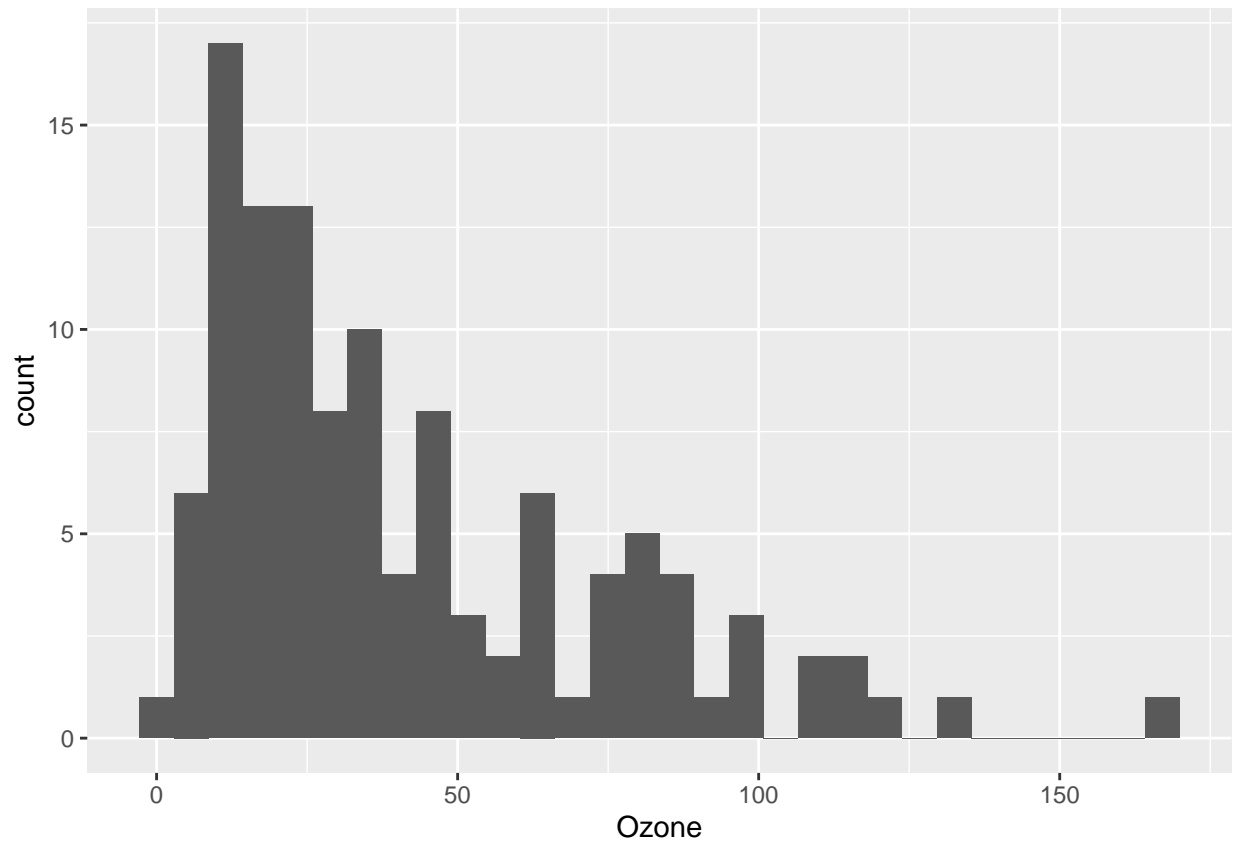
```
# bins = 5  
ggplot(data=df ) + geom_histogram(aes(Ozone), bins = 5)
```

```
## Warning: Removed 37 rows containing non-finite  
## outside the scale range (`stat_bin()`).
```



```
# bins = 30
ggplot(data=df, aes(Ozone)) +
  geom_histogram(bins = 30)
```

```
## Warning: Removed 37 rows containing non-finite
## outside the scale range (`stat_bin()`).
```



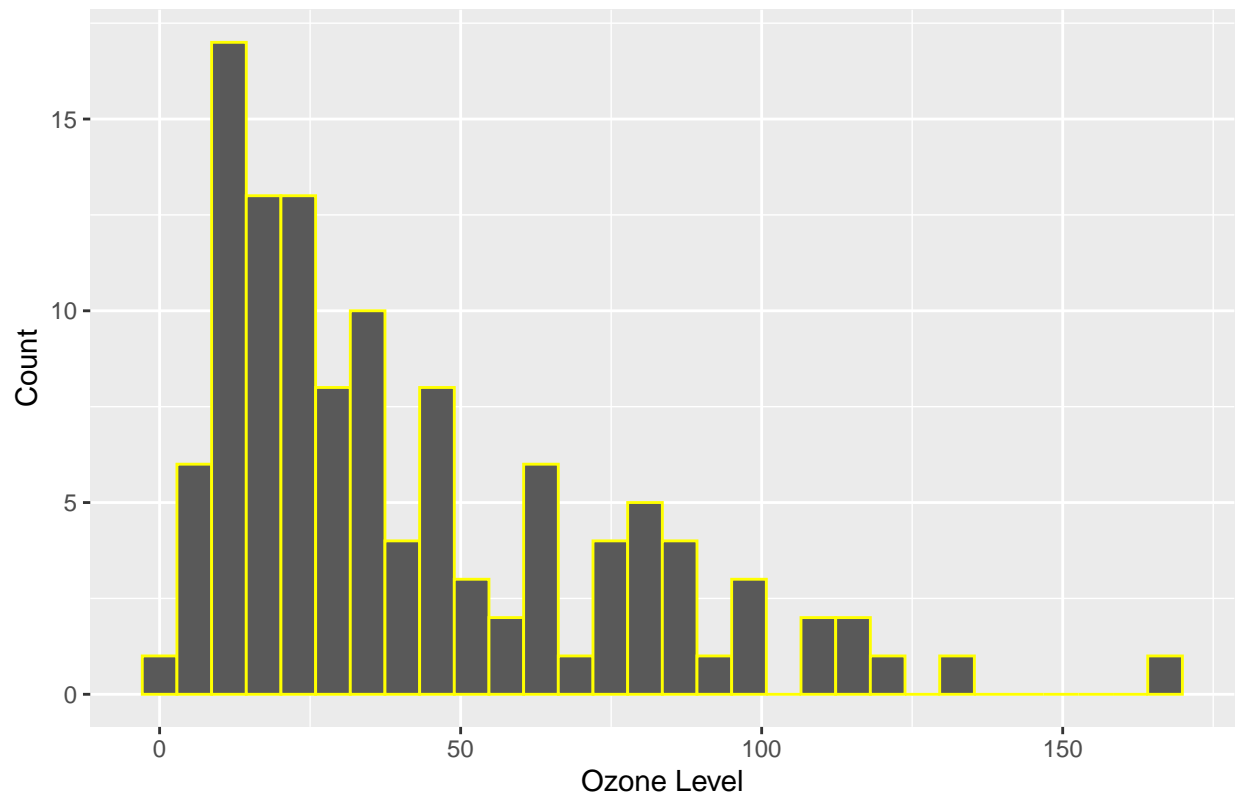
6. Labels in ggplot

With `labs(x='x lable', y='y lable', title = 'name for your plot')` you can name your plot.

```
ggplot(data=df, aes(Ozone)) +  
  geom_histogram(bins = 30, color='yellow') +  
  labs(x='Ozone Level', y='Count', title = 'Histogram with 30 bins')
```

```
## Warning: Removed 37 rows containing non-finite  
## outside the scale range (`stat_bin()`).
```

Histogram with 30 bins



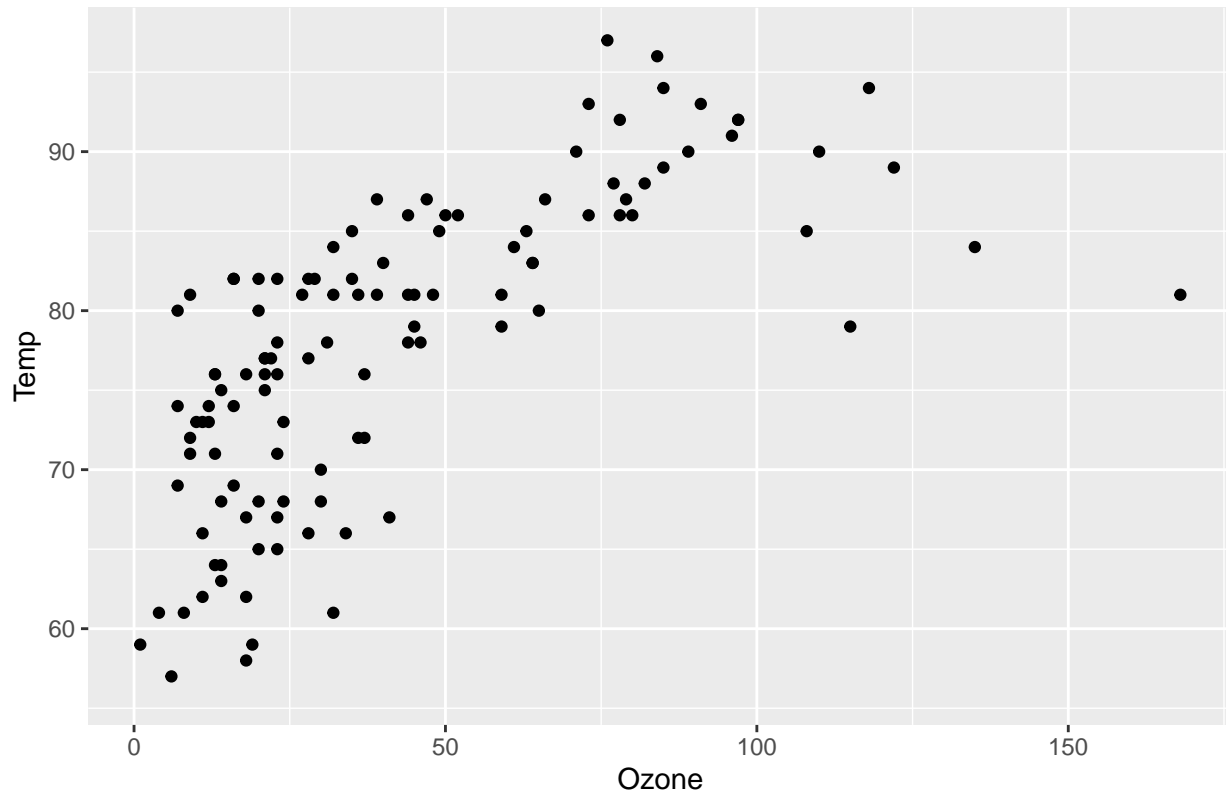
7. Scatter plot of Ozone and Temp .

We want to examine whether there is a relationship between ozone and temperature (both numerical). How do we want to approach this?

```
ggplot(df) + geom_point(aes(x=Ozone, y=Temp)) +  
  labs(title='Scatter plot of Ozone vs Temperature')
```

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


Scatter plot of Ozone vs Temperature



Titanic Data

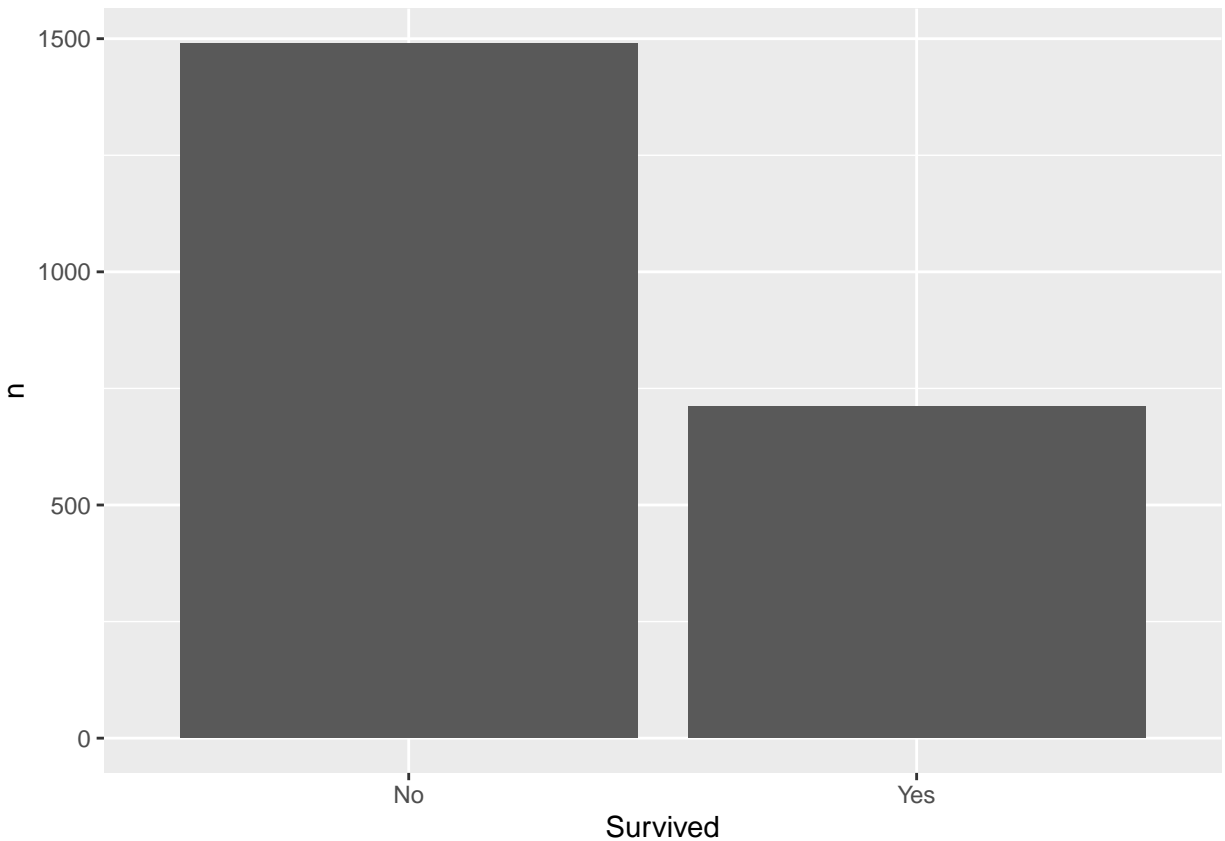
We are going to work with titanic data `titanic.csv`. Let's import the data.

```
# loading from csv file
df <- read.csv("Lab3/titanic.csv")
glimpse(df)

## Rows: 2,201
## Columns: 5
## $ X      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18~
## $ Class  <chr> "3rd", "3rd", "3rd", "3rd", "3rd", "3rd", "3rd", "3rd", "3rd"~
## $ Sex    <chr> "Male", "Male", "Male", "Male", "Male", "Male", "Male", "Male", "Male"~
## $ Age    <chr> "Child", "Child", "Child", "Child", "Child", "Child", "Child", "Child"~
## $ Survived <chr> "No", "No", "No", "No", "No", "No", "No", "No", "No", "No", "~
```

8. Barplot for Survived

```
new_df <- df %>% count(Survived)
ggplot(new_df, aes(x = Survived, y=n)) +
  geom_col()
```



9. Barplot of Survived and Class

This shows summary table of Survived and Class

```
df %>%
  group_by(Survived, Class) %>%
  summarise(count=n())

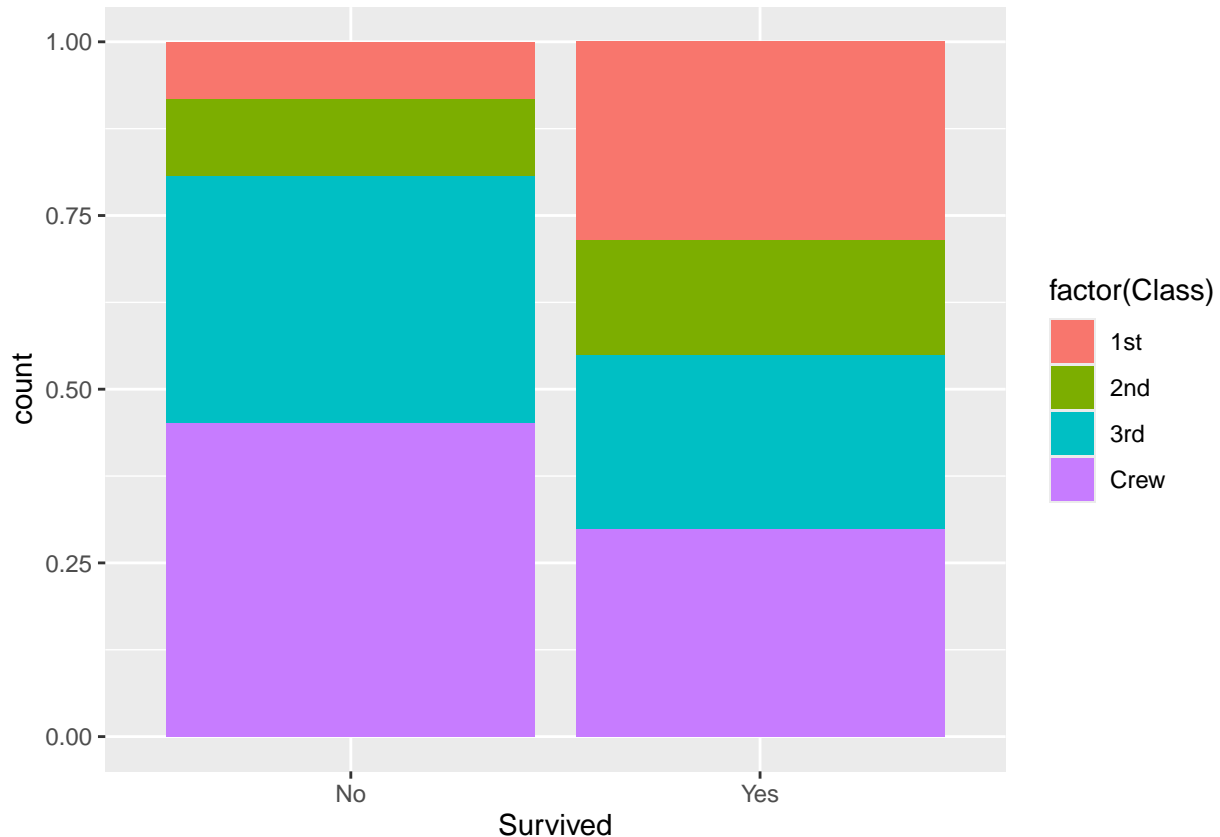
## `summarise()` has grouped output by
## 'Survived'. You can override using the
## `.groups` argument.

## # A tibble: 8 x 3
## # Groups:   Survived [2]
##   Survived Class count
##   <chr>    <chr> <int>
## 1 No      1st     122
## 2 No      2nd     167
## 3 No      3rd     528
## 4 No      Crew    673
## 5 Yes     1st     203
## 6 Yes     2nd     118
## 7 Yes     3rd     178
## 8 Yes     Crew    212
```

Adding `ggplot()` will turn it into barplot using `geom_col()`

```
df %>%
  group_by(Survived, Class) %>%
  summarise(count=n()) %>%
  ggplot(aes(x = factor(Survived), y=count, fill=factor(Class))) +
  geom_col(position = "fill") + # position can be changed to "dodge", "stack", "jitter", "fill"
  labs(x="Survived")

## `summarise()` has grouped output by
## 'Survived'. You can override using the
## `.groups` argument.
```



```
# for positioning in geom functions
# https://ggplot2.tidyverse.org/reference/layer\_positions.html
```

10. Pie chart of Class

We need table class data first in order to show a pie chart

```
# dplyr + ggplot

table_data <- df %>%
  count(Class) %>%
  mutate(Class = factor(Class))
```

```
table_data
```

```
##   Class    n
```

```
## 1 1st 325
## 2 2nd 285
## 3 3rd 706
## 4 Crew 885
```

```
# alternate way using group_by() + summarise()
```

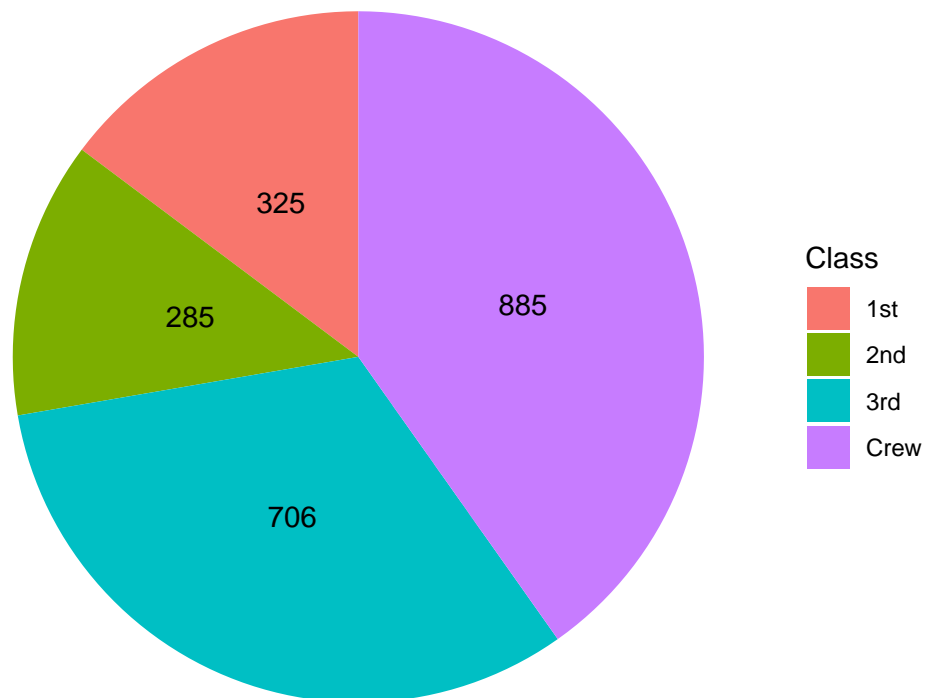
```
table_data2 <- df %>%
  group_by(Class) %>%
  summarise(N = n())
```

```
table_data2
```

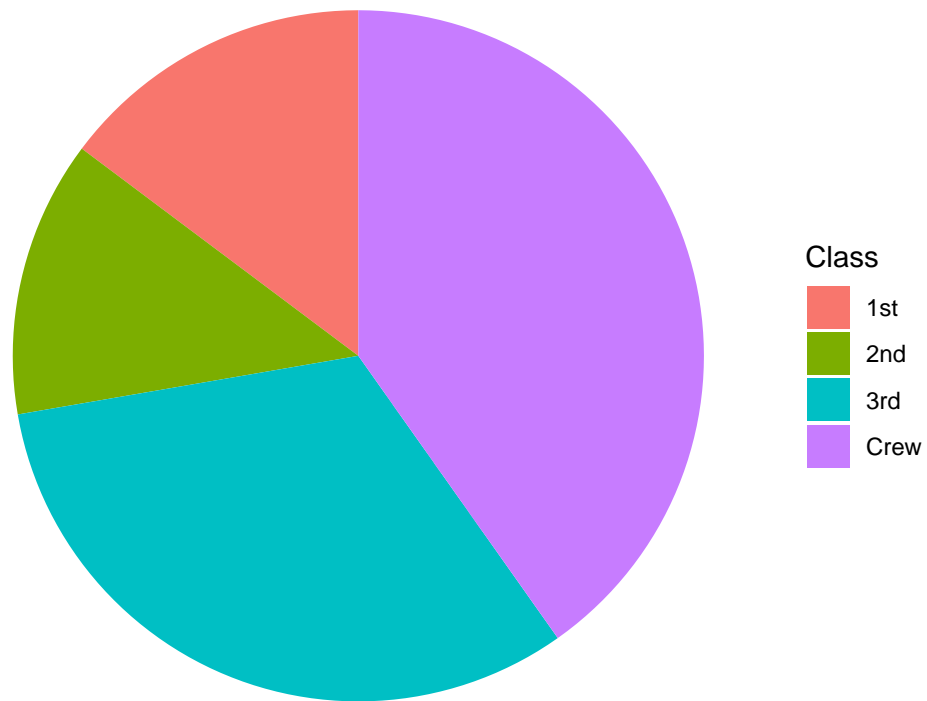
```
## # A tibble: 4 x 2
##   Class      N
##   <chr> <int>
## 1 1st     325
## 2 2nd     285
## 3 3rd     706
## 4 Crew    885
```

Using this table class object `table_data`, we obtain a pie chart.

```
ggplot(data = table_data, aes(x="", y=n, fill=Class)) +
  geom_col() +
  geom_text(aes(label = n), position = position_stack(vjust = 0.5)) +
  coord_polar(theta="y") +
  theme_void()
```



```
# or simply
ggplot(data = table_data, aes(x="", y=n, fill=Class)) +
  geom_col() +
  coord_polar(theta="y") +
  theme_void()
```



```
# using table_data2
ggplot(data = table_data2, aes(x="", y=N, fill=Class)) +
  geom_col() +
  coord_polar(theta="y") +
  theme_void()
```

