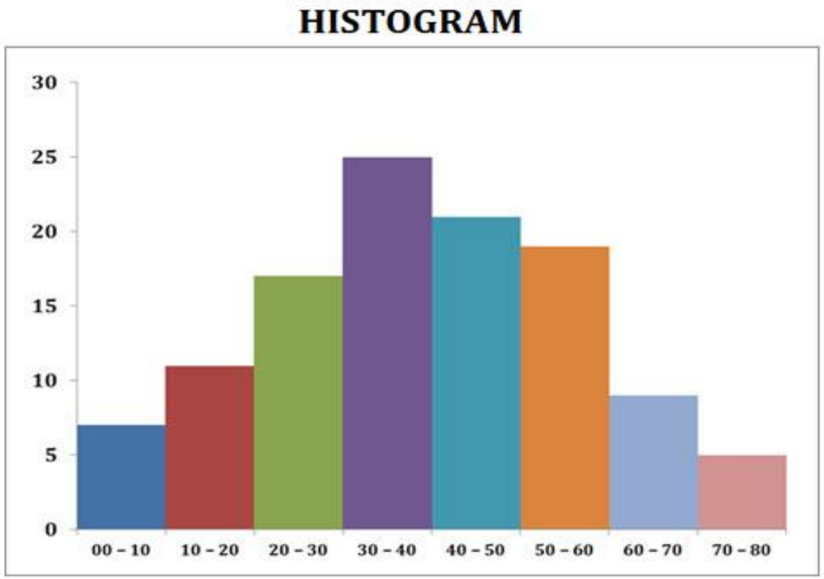
HISTOGRAM

By : Shreya Ganguly(MAT/19/50) & Anshika Gahlot (MAT/19/59)

What is a Histogram ?

A graphical display where the data is grouped into ranges (such as "100 to 149", "150 to 199", etc), and then plotted as bars. The height of each bar shows how many are in each range. Histogram is similar to bar chart but the difference is it groups the values into continuous ranges.



How to make a Histogram using Basic R ?

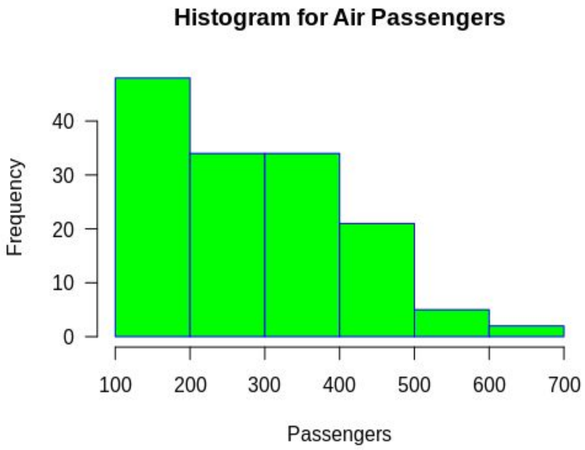
R creates histogram using **hist()** function. This function takes a vector as an input and uses some more parameters to plot histograms.

SYNTAX : The basic syntax for creating a histogram using R is :

```
hist(v, main, xlab, xlim, ylim, breaks, col, border)
```

Description of the Parameters Used –

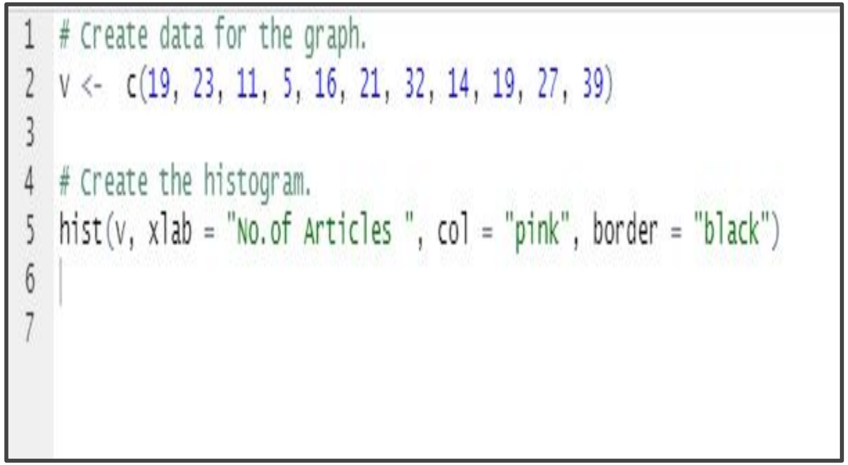
- **v** is a vector containing numeric values used in histogram.
- **main** indicates title of the chart.
- **col** is used to set color of the bars.
- **border** is used to set border color of each bar.
- **xlab** is used to give description of x-axis.
- **ylab** is used to give description of y-axis.
- **xlim** is used to specify the range of values on the x-axis.
- **ylim** is used to specify the range of values on the y-axis.
- **breaks** is used to mention the width of each bar.



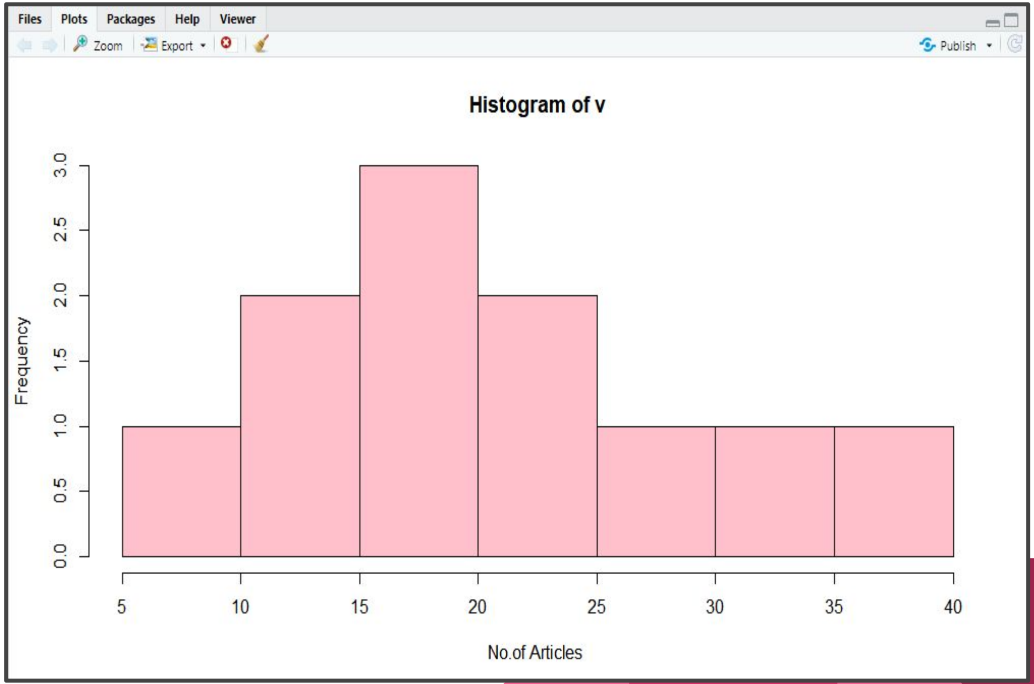
Examples :

1. Creating A Simple Histogram.

We create a vector ‘v’ and we plot the histogram using hist() and only some selected parameters(xlab , col, border).

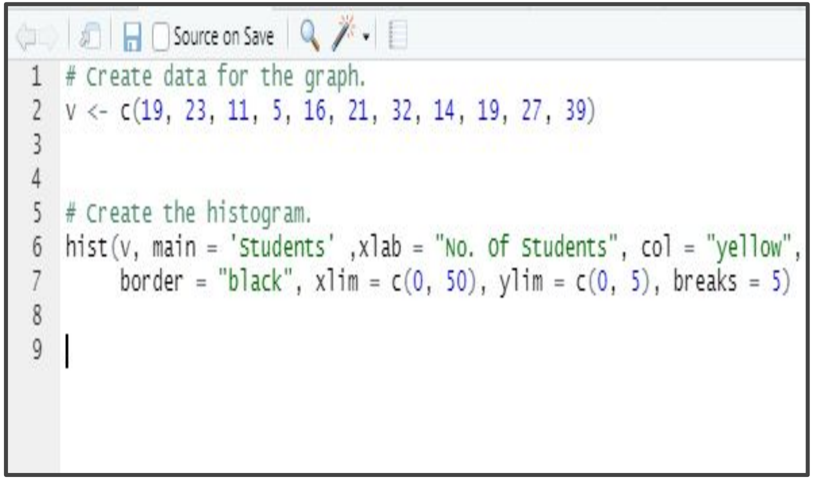


```
# Create data for the graph.
v <-  c(19, 23, 11, 5, 16, 21, 32, 14, 19, 27, 39)

# Create the histogram.
hist(v, xlab = "No.of Articles ", col = "pink", border = "black")

```

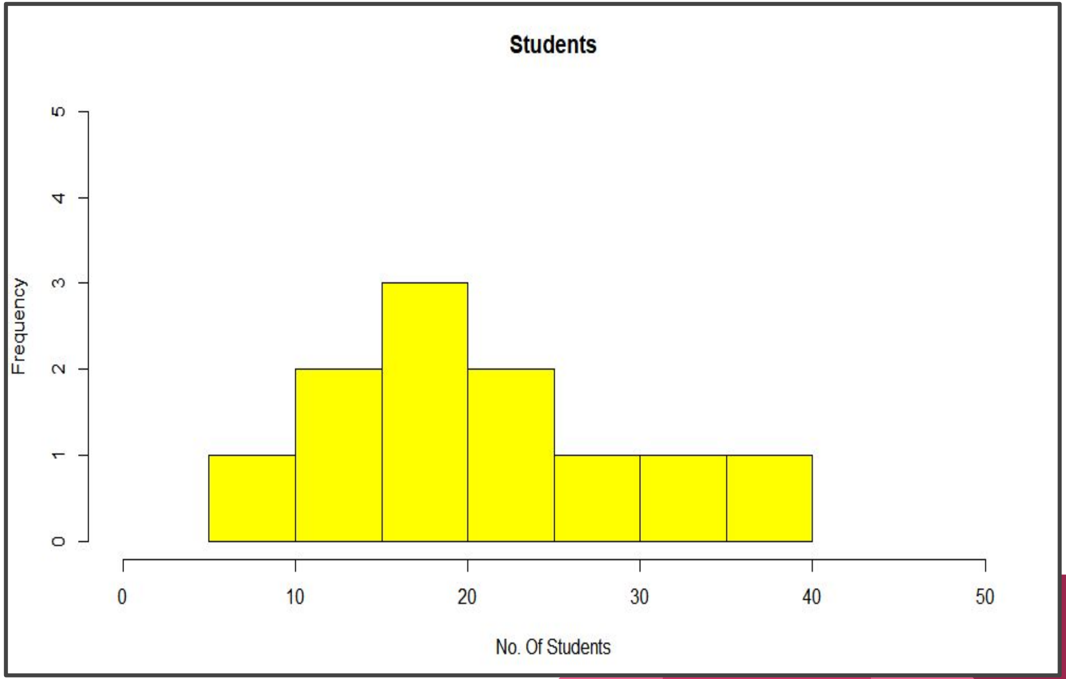


2. Specifying Range Of X & Y Values

We create a vector ‘v’ and we plot the histogram using hist() . Taking xlim and ylim for specifying the X & Y ranges .

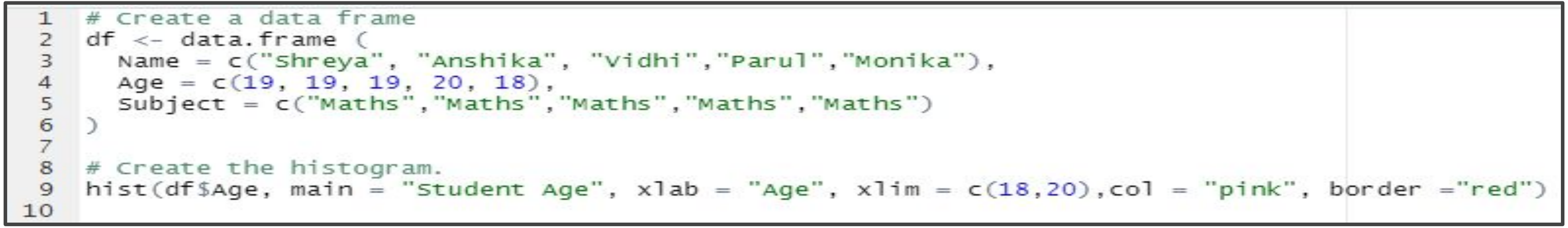


```r
# Create data for the graph.
v <- c(19, 23, 11, 5, 16, 21, 32, 14, 19, 27, 39)


# Create the histogram.
hist(v, main = 'Students' ,xlab = "No. Of Students", col = "yellow",
     border = "black", xlim = c(0, 50), ylim = c(0, 5), breaks = 5)

```

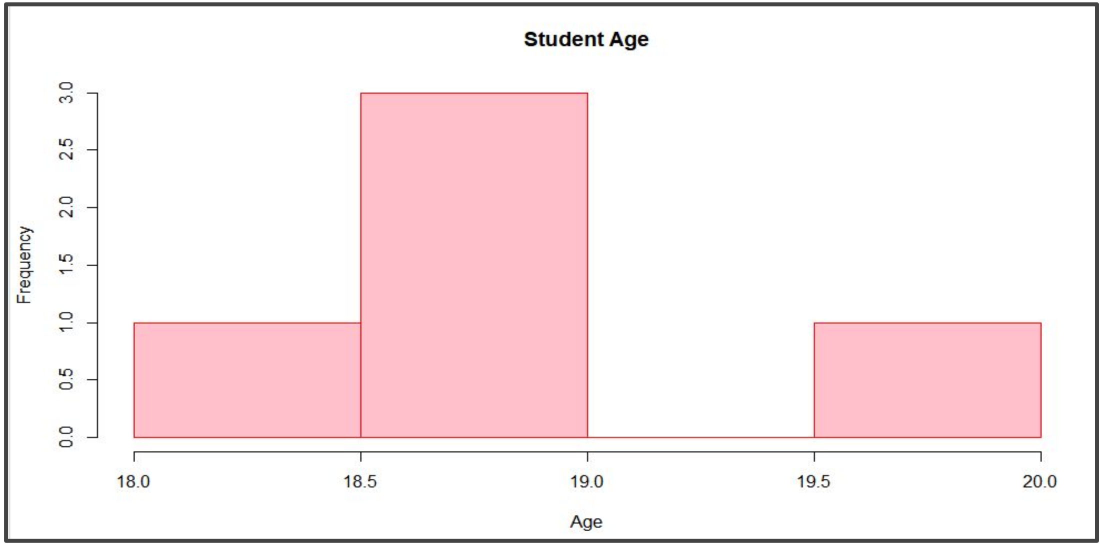


3. Histogram Using Demo Data.

We create a dataframe and then plot the histogram using hist().



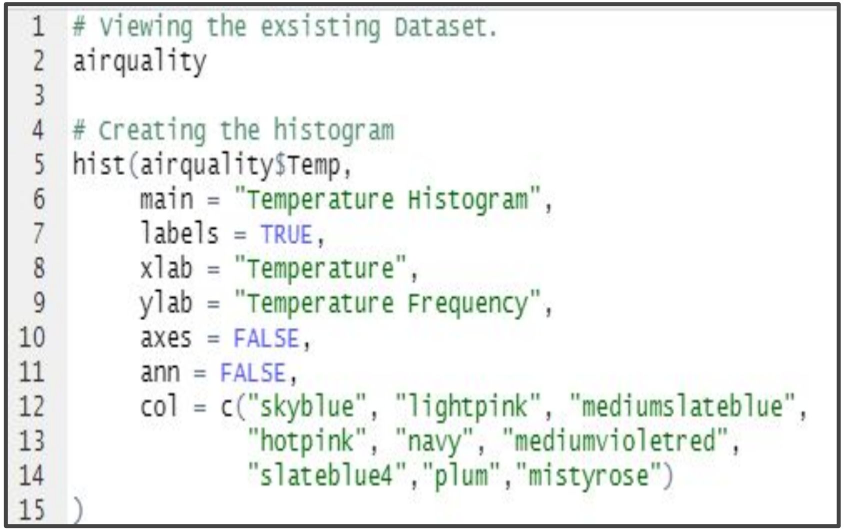
```r
# Create a data frame
df <- data.frame (
   Name = c("Shreya", "Anshika", "Vidhi","Parul","Monika"),
   Age = c(19, 19, 19, 20, 18),
   Subject = c("Maths","Maths","Maths","Maths","Maths")
)

# Create the histogram.
hist(df$Age, main = "Student Age", xlab = "Age", xlim = c(18,20),col = "pink", border ="red")

```



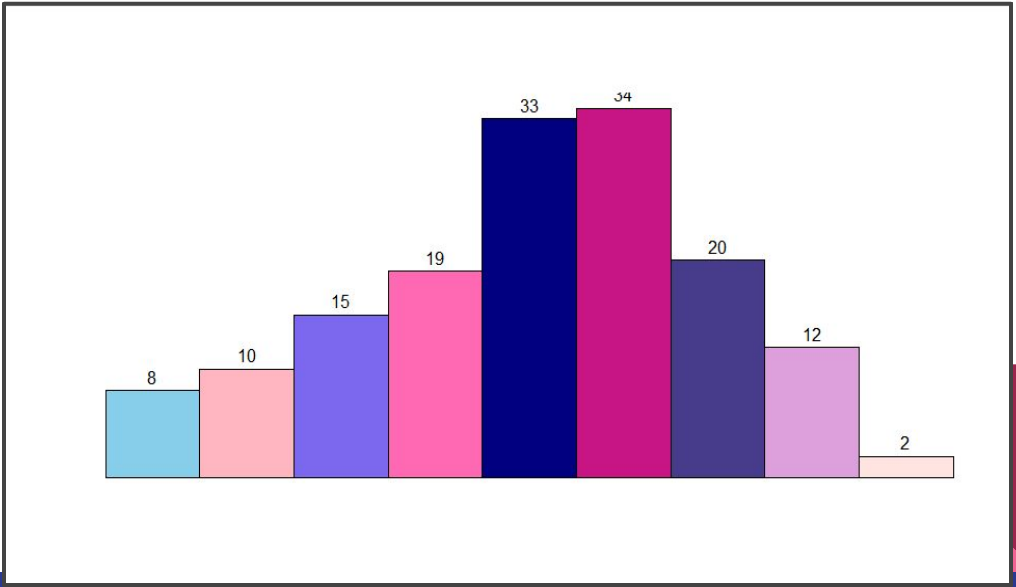
4. Colorful Bars, Adding Labels & Removing Axis.

We use an existing dataset and we plot the histogram using hist() .Inside the hist() we provide different colors to the ‘col’ parameter and we set ‘labels’ as TRUE and ‘axes’ , ‘ann’ as FALSE.

- **axes:** It is a Boolean argument. If it is TRUE, the axis is drawn.
- **labels:** It is a Boolean argument. If it is TRUE, Histogram returns the value on top of each bar.
- **ann:** It is a Boolean argument. If it is FALSE, Histogram removes the annotations from the plot area, which includes the Histogram name, Axis Names.

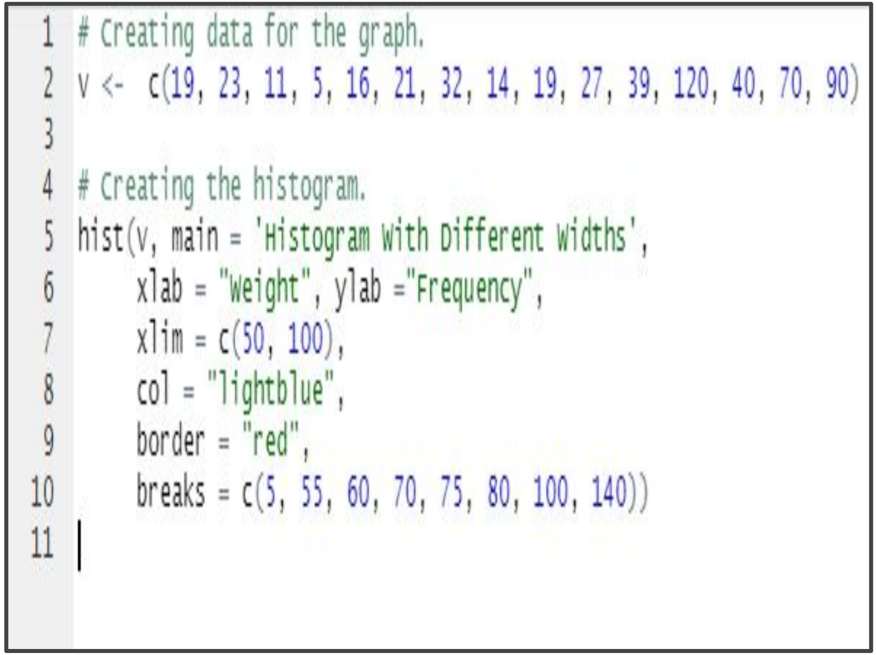


```r
# Viewing the exsisting Dataset.
airquality

# Creating the histogram
hist(airquality$Temp,
     main = "Temperature Histogram",
     labels = TRUE,
     xlab = "Temperature",
     ylab = "Temperature Frequency",
     axes = FALSE,
     ann = FALSE,
     col = c("skyblue", "lightpink", "mediumslateblue",
             "hotpink", "navy", "mediumvioletred",
             "slateblue4","plum","mistyrose")
)
```

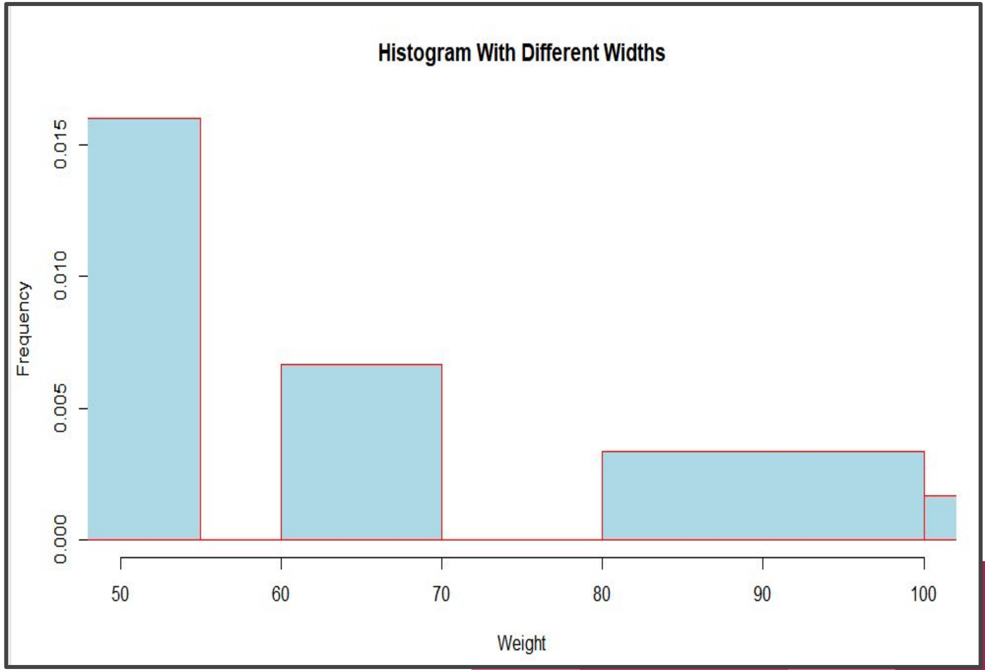


5. Histogram with Non-Uniform Width

We create a vector ‘v’ and we plot the histogram using hist() and we provide different widths to the break parameter.



```r
# Creating data for the graph.
v <-  c(19, 23, 11, 5, 16, 21, 32, 14, 19, 27, 39, 120, 40, 70, 90)

# Creating the histogram.
hist(v, main = 'Histogram With Different Widths',
     xlab = "Weight", ylab ="Frequency",
     xlim = c(50, 100),
     col = "lightblue",
     border = "red",
     breaks = c(5, 55, 60, 70, 75, 80, 100, 140))

```



Plotting Histograms Using ggplot2

We can also make histograms by using **ggplot2**.

ggplot2 is a plotting package that makes it simple to create complex plots from data in a data frame. It provides a more programmatic interface for specifying what variables to plot, how they are displayed, and general visual properties.

It is based on **The Grammar of Graphics**.

How To Install ggplot2

There are 3 ways to install ggplot2 :

1. We can install the whole tidyverse : install.packages("tidyverse")

2. We can install just ggplot2 : install.packages("ggplot2")

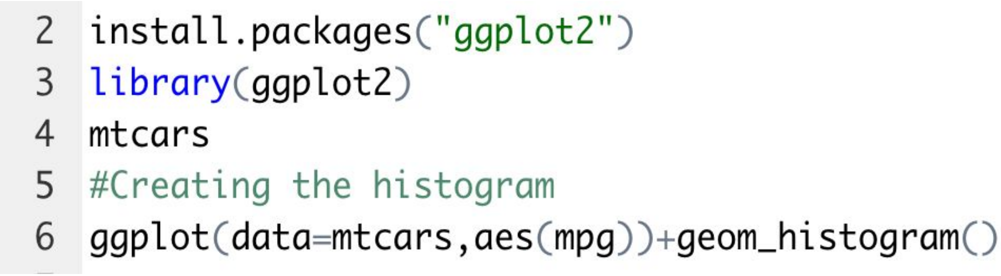
3. We can also use the developmental version from GitHub :

   ```
    install.packages("devtools")
   devtools::install_github("tidyverse/ggplot2")
   ```

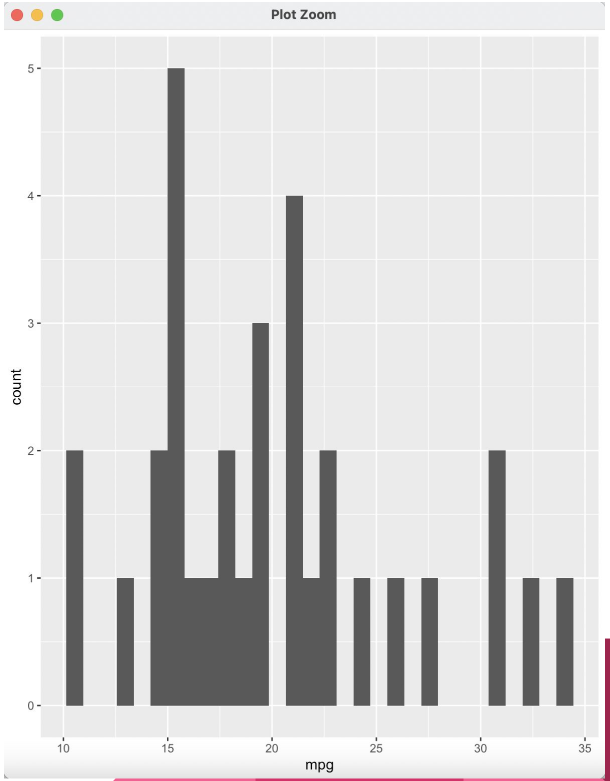
Using ggplot

We use data argument to specify the data set, we add aes to describe how variables in the data (such as mpg) are mapped to visual properties of geoms (geom_histogram).

Example 1 :



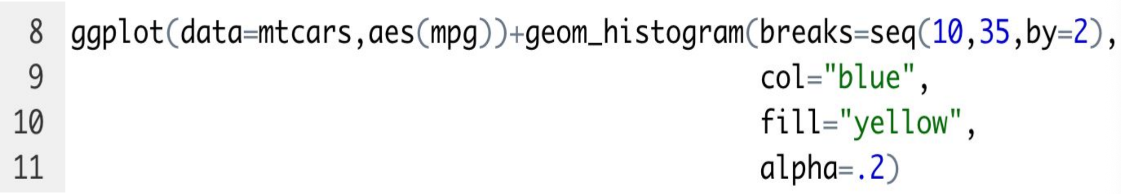
```
install.packages("ggplot2")
library(ggplot2)
mtcars
#Creating the histogram
ggplot(data=mtcars,aes(mpg))+geom_histogram()
```



Different Parameters of geom_histogram

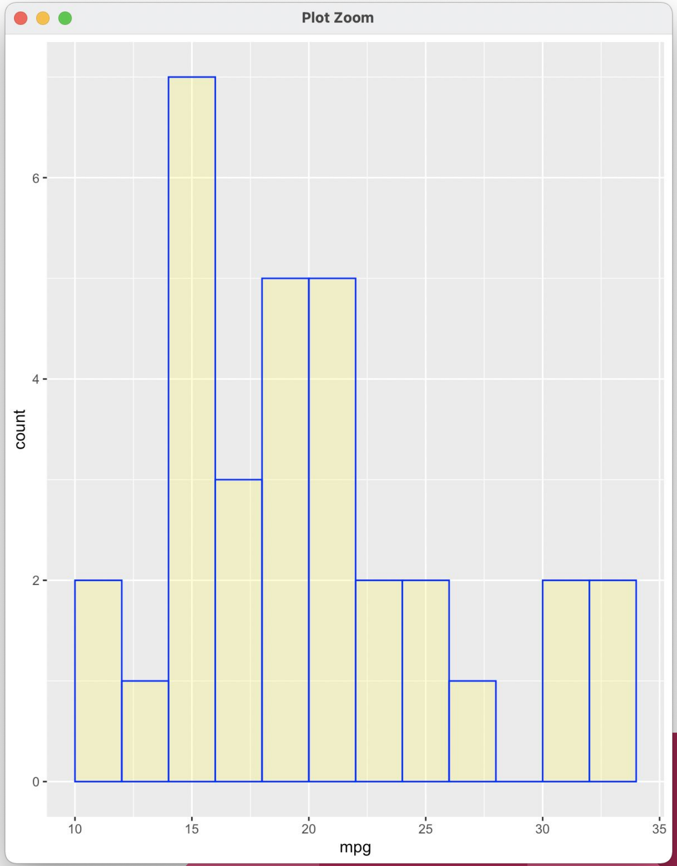
1. **breaks :** This is used to change the width of the columns on x-axis.
2. **col :** This is used to change colours of the border of the histograms.
3. **fill :** This is used to fill color in the histogram.
4. **alpha :** This is used to change the transparency of the graph. It ranges from 0 to 1.
5. **count :** number of points in bin**.**
6. **linetype :** This is used to define the type of line to be used.

Example 2 :

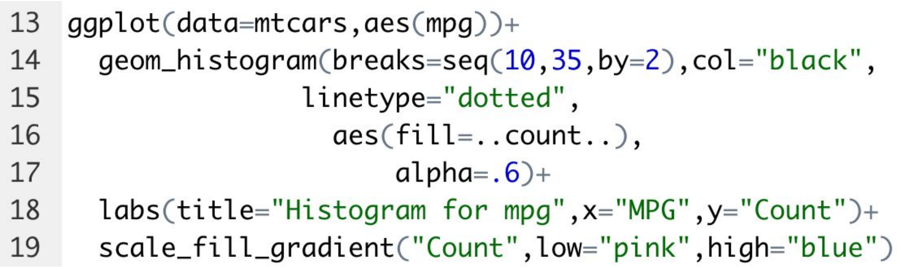


```
ggplot(data=mtcars,aes(mpg))+geom_histogram(breaks=seq(10,35,by=2),
                                            col="blue",
                                            fill="yellow",
                                            alpha=.2)
```

Here, we have used breaks in seq(10,35,by=2) to make the histogram easier to read and visualise.



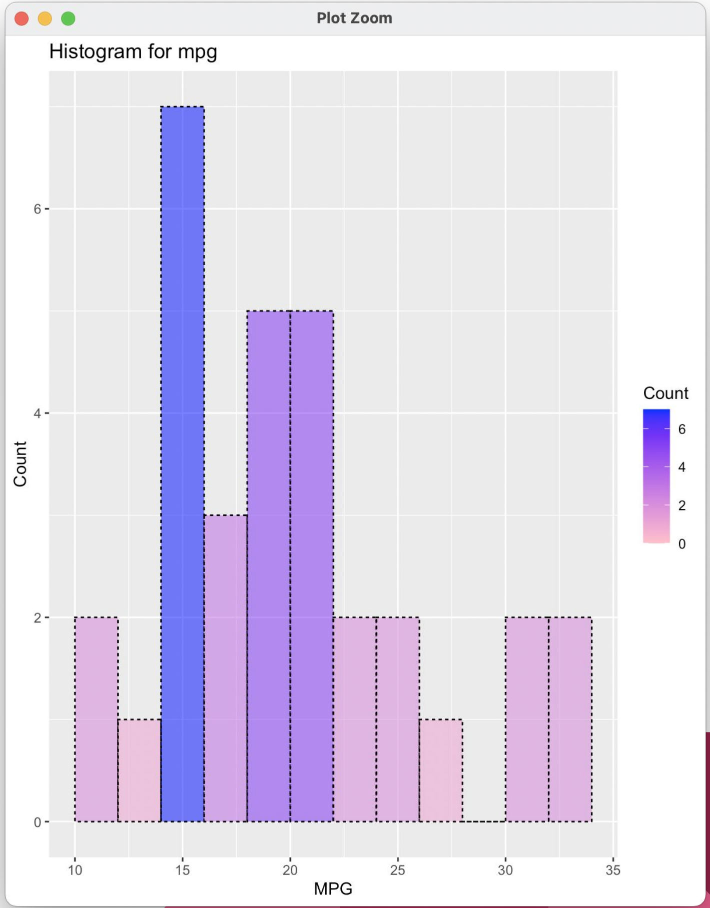
Example 3 :



```
ggplot(data=mtcars,aes(mpg))+
  geom_histogram(breaks=seq(10,35,by=2),col="black",
               linetype="dotted",
                 aes(fill=..count..),
                     alpha=.6)+
  labs(title="Histogram for mpg",x="MPG",y="Count")+
  scale_fill_gradient("Count",low="pink",high="blue")
```

Here , we have used aes(fill=..count..) and scale_fill_gradient to put color gradient in the graph. Low signifies low count and high signifies high count.

Labs() argument is used to give the title and name for x and y-axis.



Thank you !