

Statistical Computing with R

Masters in Data Science 503 (S13)

First Batch, SMS, TU, 2021

Shital Bhandary

Associate Professor

Statistics/Bio-statistics, Demography and Public Health Informatics

Patan Academy of Health Sciences, Lalitpur, Nepal

Faculty, Data Analysis and Decision Modeling, MBA, Pokhara University, Nepal

Faculty, FAIMER Fellowship in Health Professions Education, India/USA.

Review Preview

- Basic graphics/plots:
 - Bar chart
 - Histogram
 - Q-Q plot
 - Density plot
 - Pie chart
 - Line chart
 - Scatterplot
 - Boxplot etc.

Graphs/Plots in R:

- Base R: “graphics” & “grDevices” packages, loads automatically, we need to learn it in this course
- “lattice” package # we need to install it to use it, not covered in this course
- **“ggplot2” package #we need to install it to use it in this course**
- **GG = Grammar of Graphics**

Basic plots: Bar Diagram (s)

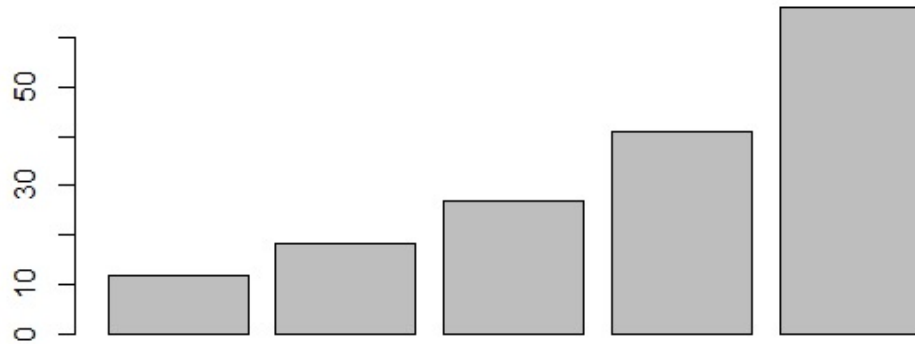
- It is used to represent the distribution of categorical variable graphically
- It can be:
 - Simple
 - Sub-divided/stacked
 - Multiple/Grouped

Bar diagram in R: base “graphics” packages

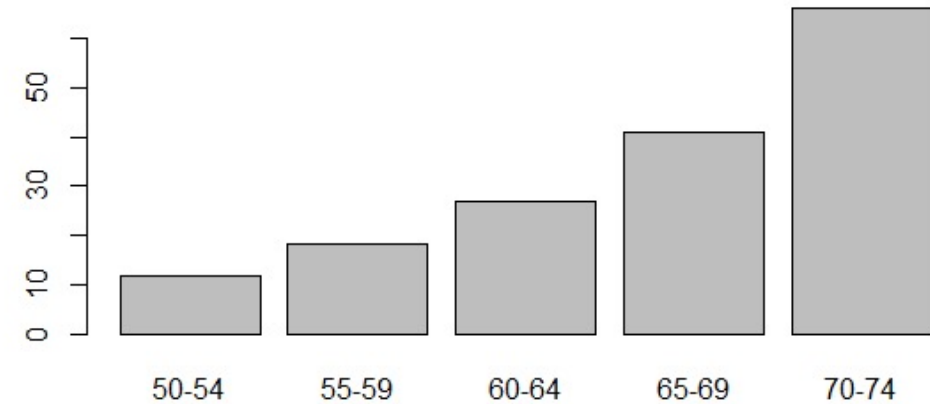
Type: ?barplot to see the full syntax

- We can use the R’s built-in dataset: VADeaths
- `gd <- as.data.frame(VADeaths)`
- `View(gd)` **#You may need to transform your raw data like this one using chapter 5 of your text book!**
- `barplot(gd$`Rural Male`)`
- `barplot(gd$`Rural Male`, names.arg = c("50-54", "55-59", "60-64", "65-69", "70-74"))`

Outputs:



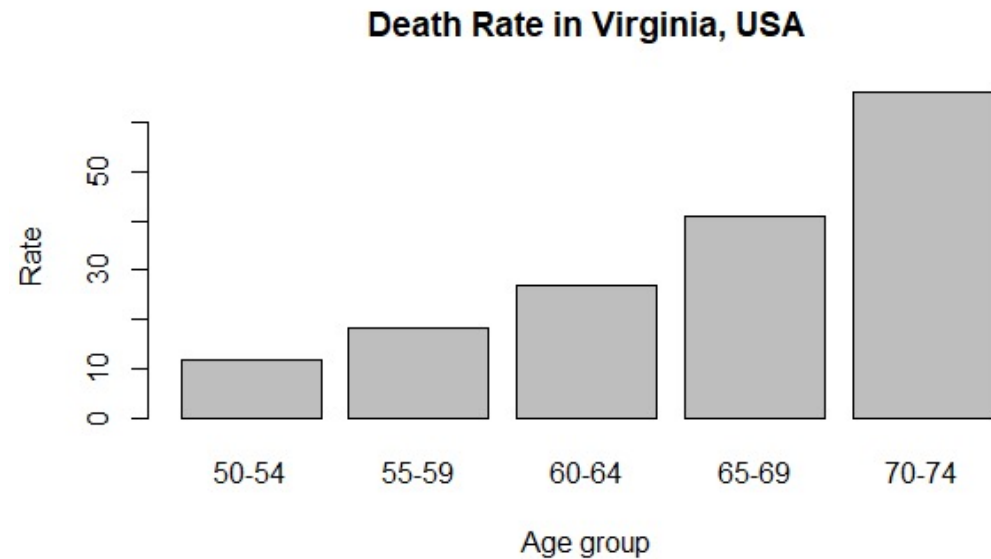
`barplot(gd$`Rural Male`)`



`barplot(gd$`Rural Male`, names.arg = c("50-54", "55-59", "60-64", "65-69", "70-74"))`

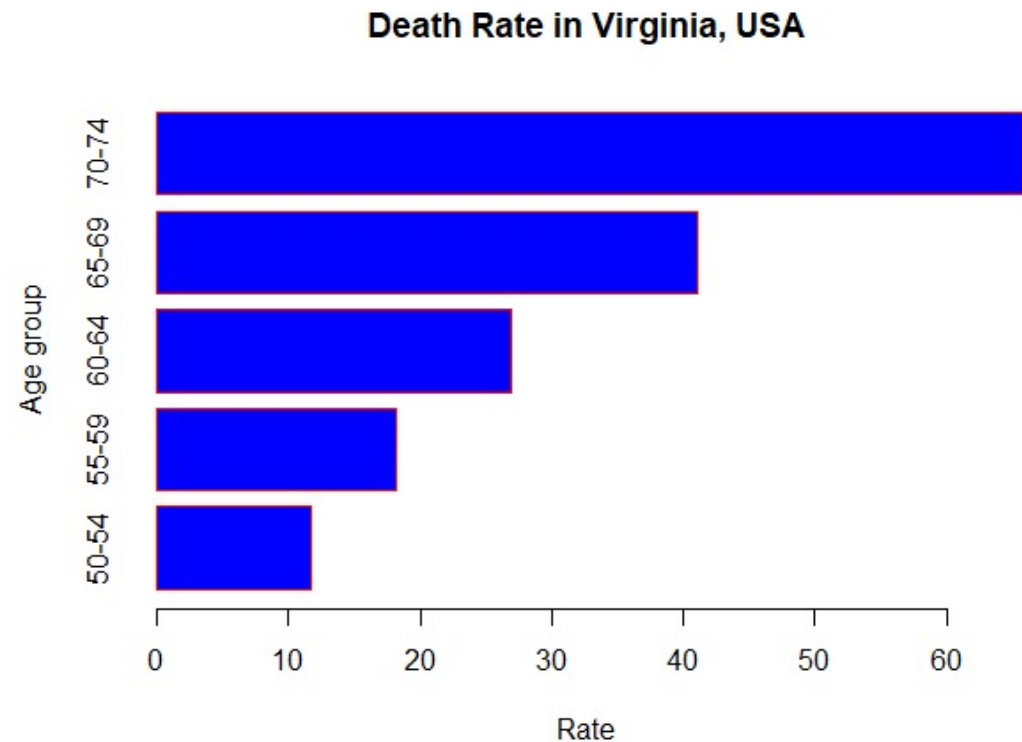
Adding title and labels for x & y axis:

```
barplot(gd$`Rural Male`,  
names.arg = c("50-54", "55-59",  
"60-64", "65-69", "70-74"),  
main = "Death Rate in Virginia,  
USA", xlab = "Age group", ylab =  
"Rate")
```



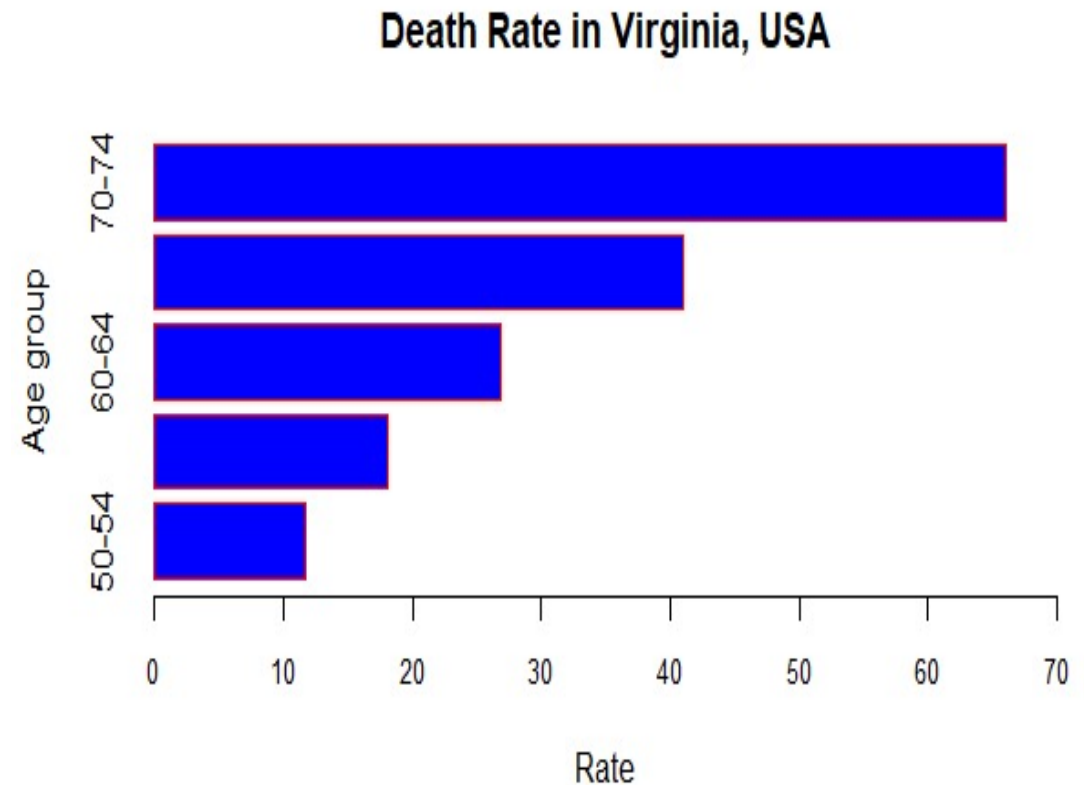
Changing orientation and adding colors:

```
barplot(gd$`Rural Male`, horiz = T,  
names.arg = c("50-54", "55-59",  
"60-64", "65-69", "70-74"),  
main = "Death Rate in Virginia,  
USA", xlab = "Rate", ylab = "Age  
group",  
col = "blue", border = "red")
```



Changing axis length and font size:

- `barplot(gd$`Rural Male`, horiz = T, names.arg = c("50-54", "55-59", "60-64", "65-69", "70-74"), main = "Death Rate in Virginia, USA", xlab = "Rate", ylab = "Age group", col = "blue", border = "red", xlim = c(0,70), cex.axis = 0.80)`



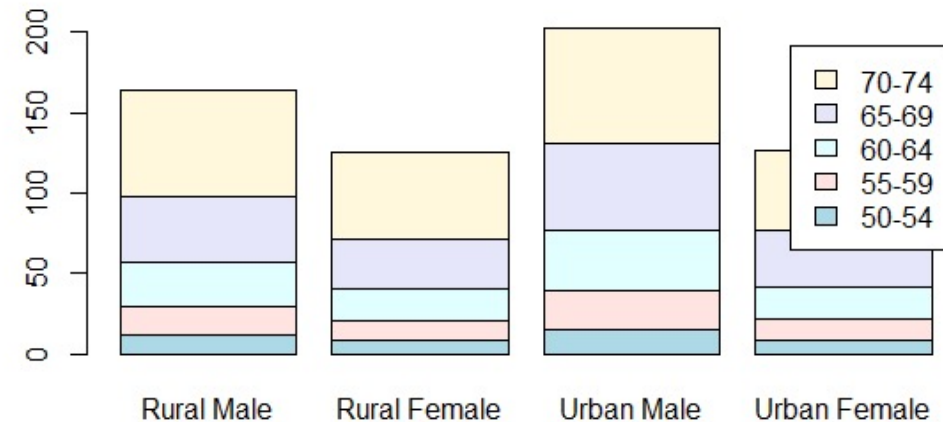
Sub-divided/Staked Bar diagram:

```
barplot(gd,  
        col = c("lightblue",  
                "mistyrose", "lightcyan",  
                "lavender", "cornsilk"),  
        legend =  
        rownames(VADeaths))
```

- Error in barplot.default(gd, col = c("lightblue", "mistyrose", "lightcyan", :
• 'height' must be a vector or a matrix

Sub-divided/Staked Bar diagram:

```
gdm <- as.matrix(gd)
barplot(gdm,
        col = c("lightblue",
                "mistyrose", "lightcyan",
                "lavender", "cornsilk"),
        legend = rownames(gd))
```



Note: The gdm is a matrix!

Sub-divided bar diagram with placement, size and box of the legend:

- **# Define a set of colors**

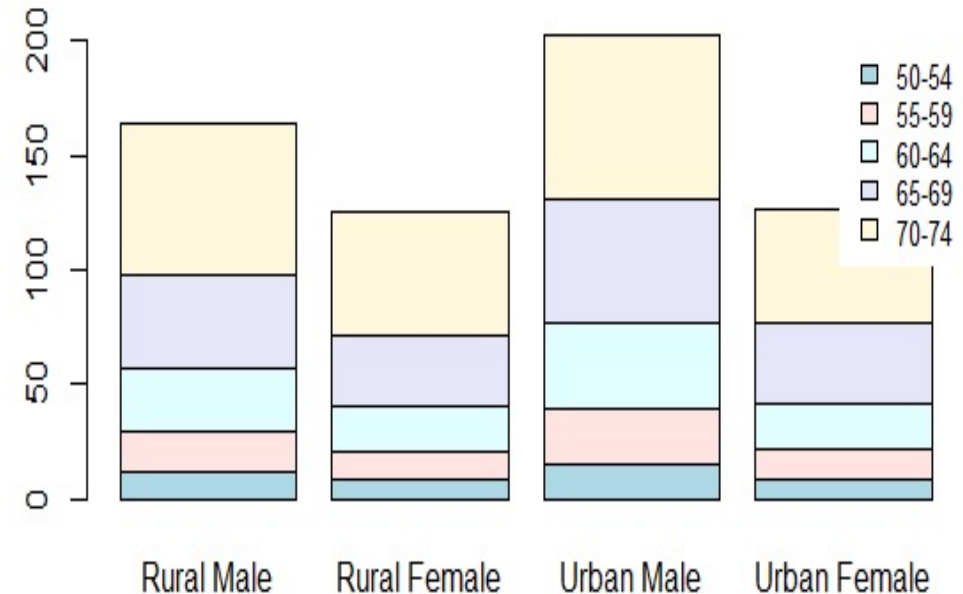
```
my_colors <- c("lightblue",  
"mistyrose", "lightcyan",  
              "lavender", "cornsilk")
```

- **# Bar plot**

```
barplot(gdm, col = my_colors)
```

- **# Add legend**

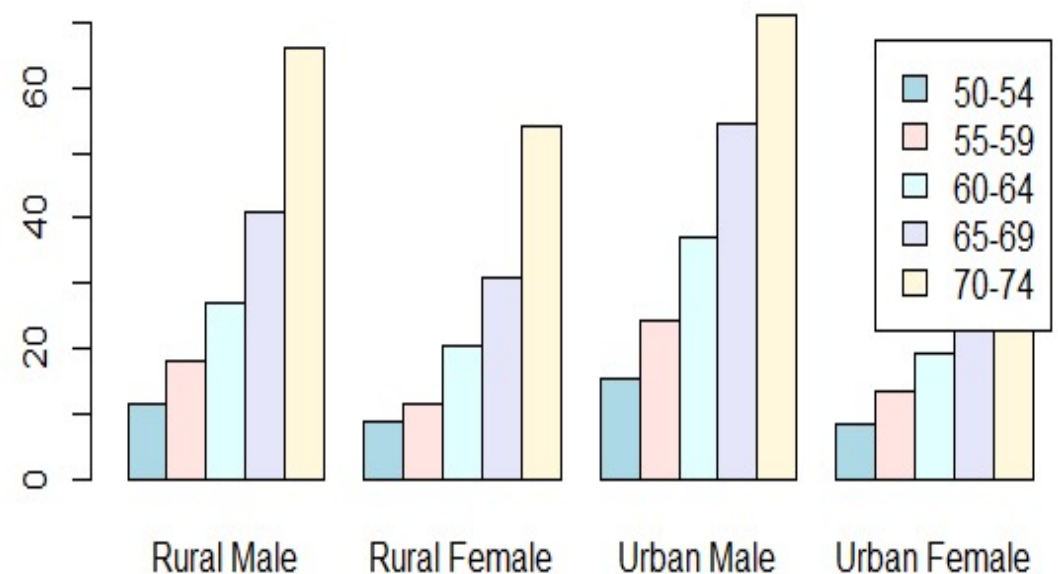
```
legend("topleft", legend =  
rownames(gdm),  
fill = my_colors, box.lty = 0, cex = 0.8)
```



Multiple/Grouped Bar Diagram:

- `barplot(gdm,`
 `col = c("lightblue",`
 `"mistyrose", "lightcyan",`
 `"lavender", "cornsilk"),`
 `legend = rownames(gdm),`
 `beside = T)`

Note: Adding beside – TRUE will produce the multiple bar chart!



Multiple/Group Bar Diagram with change in legend values:

Define a set of colors

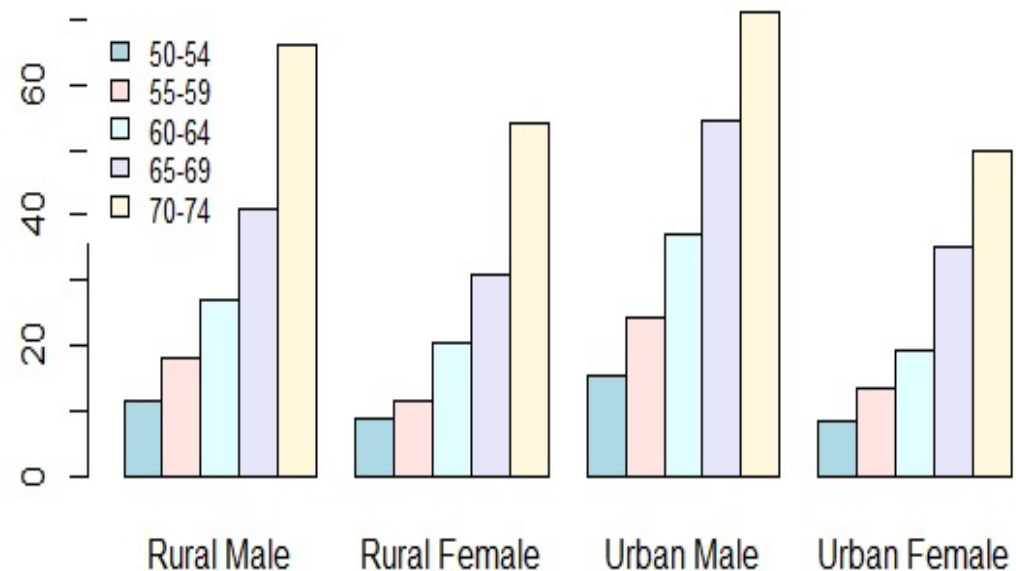
```
my_colors <- c("lightblue",  
"mistyrose", "lightcyan",  
"lavender", "cornsilk")
```

Bar plot

```
barplot(gdm, col = my_colors, beside  
= TRUE)
```

Add legend

```
legend("topleft", legend =  
rownames(gdm), fill = my_colors,  
box.lty = 0, cex = 0.8)
```

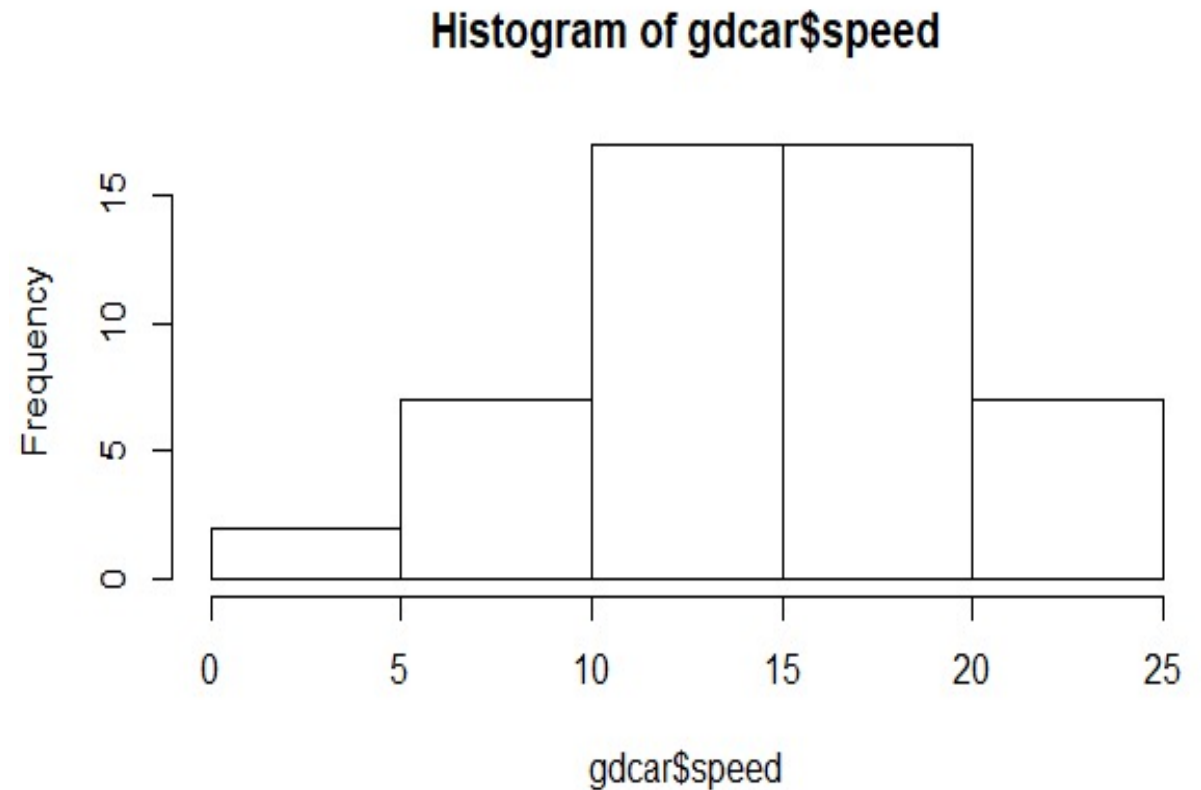


Histogram, Q-Q and Density plots:

- This is use to represent “continuous” variable graphically
- Histogram is created after converting the continuous data into class intervals, which R calls “breaks”
- Histogram is based on the “density” rather than frequencies
- Histogram, Q-Q plot and density plots are effective to check the “distribution” of the data, which in turn provides cues to use correct descriptive statistics and tests!

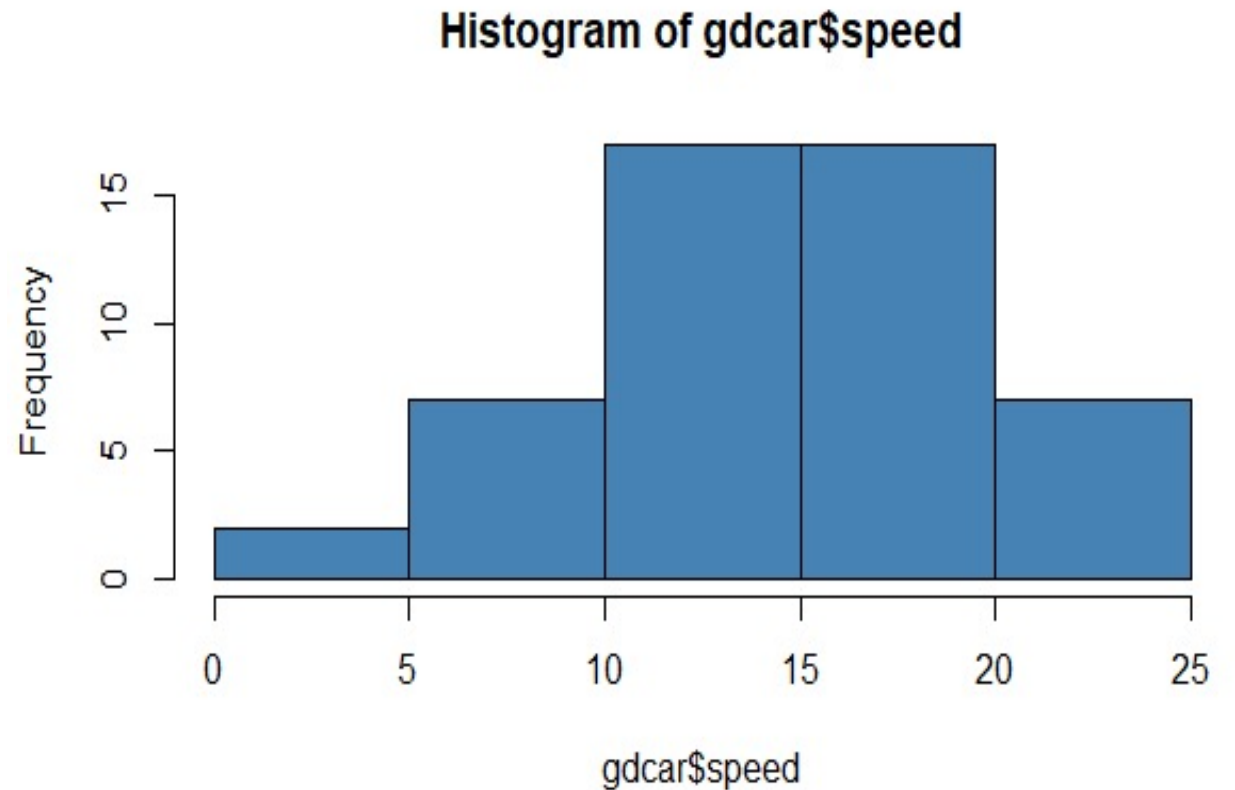
Histogram using built-in “cars” data:

- `gdcar <- as.data.frame(cars)`
- `hist(gdcar$speed)`



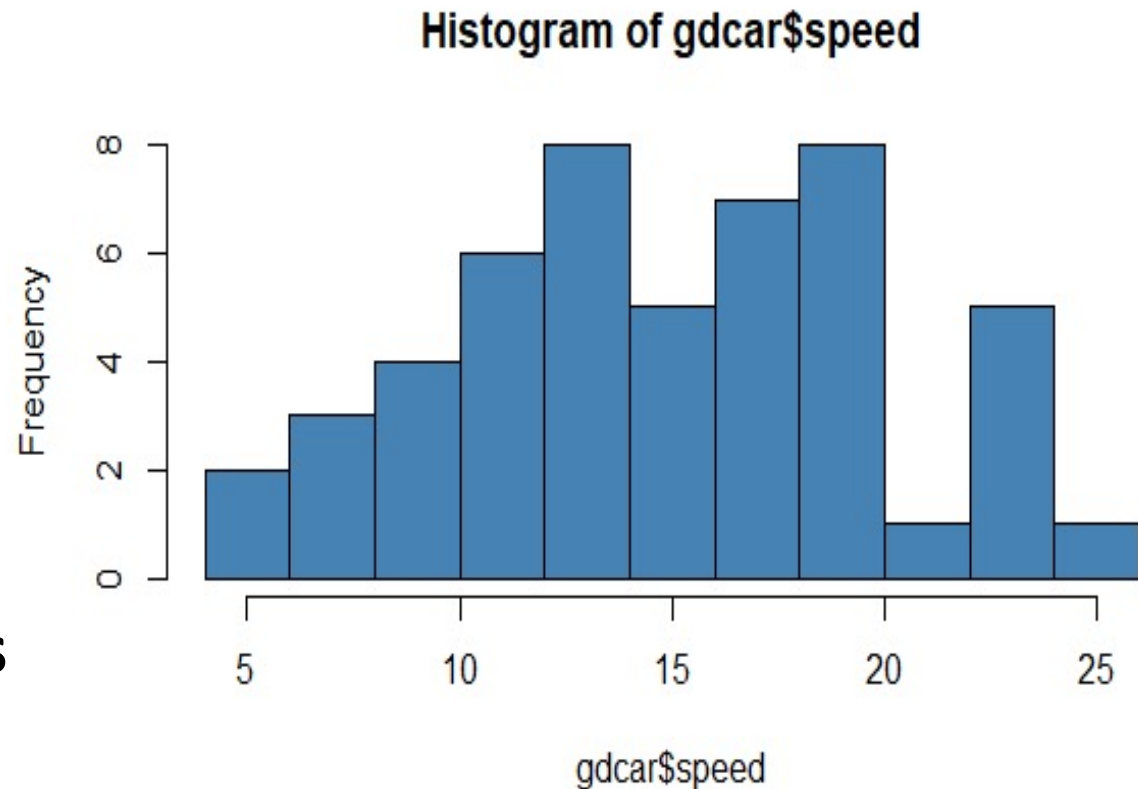
Histogram using built-in “cars” data:

- `gdcar <- as.data.frame(cars)`
- `hist(gdcar$speed, col = "steelblue")`



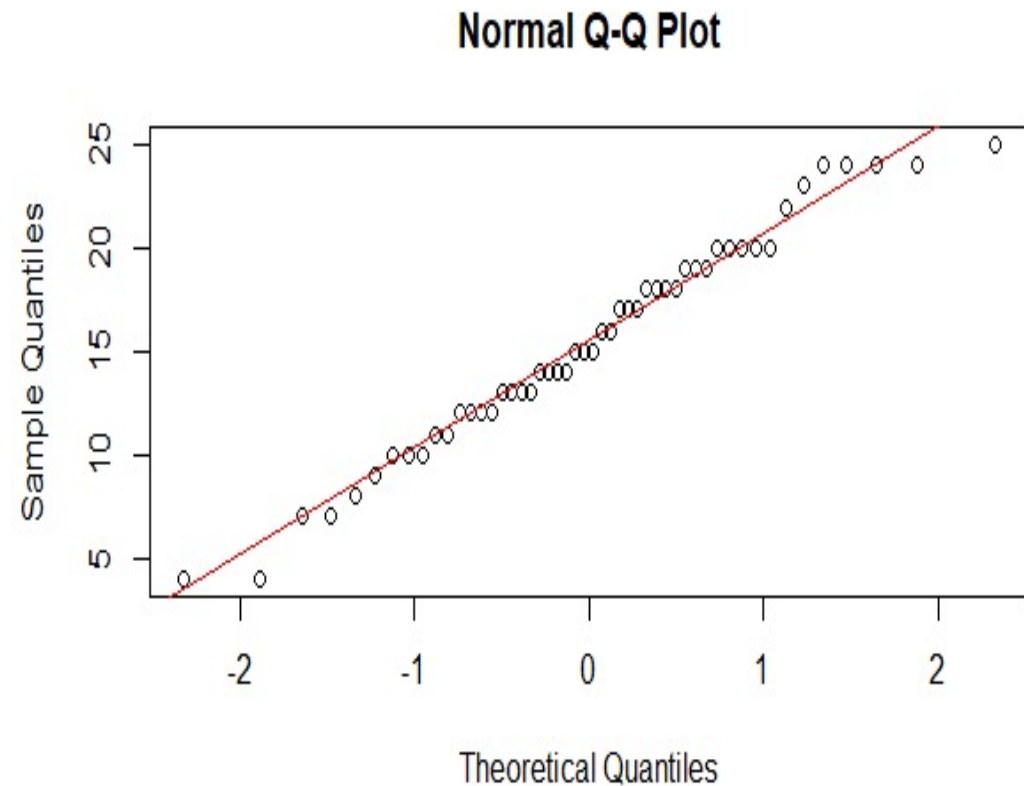
Histogram using built-in “cars” data:

- `gdcar <- as.data.frame(cars)`
- `hist(gdcar$speed, col = "steelblue", breaks = 10)`
- Is there any rule on how many “breaks” should be used in a histogram?
- How to change title and x-axis label in this diagram?



Q-Q plot with Q-Q line:

- `#Q-Q plot`
- `qqnorm(gdcars$speed)`
- `#Q-Q line`
- `qqline(gdcars$speed, col="red")`
- Note: The observed values do not lie in the theoretical normal distribution quintiles. It can be considered as “robust” though!



Density plot (freq. polygon): Speed variable

#Density plot using cars dataset

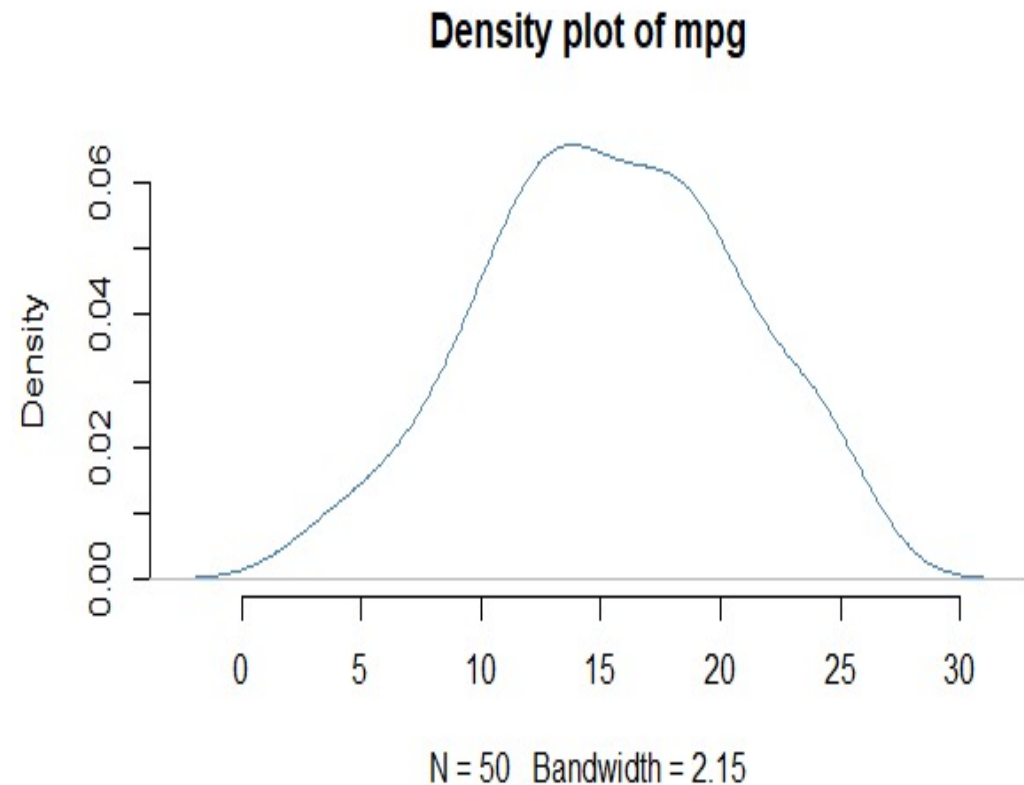
Compute the density data

- `dens <- density(cars$speed)`

What is the density?

plot density

- `plot(dens, frame = FALSE, col = "steelblue", main = "Density plot of mpg")`



Density plot with polygon fill: Speed variable

#Density plot using cars dataset

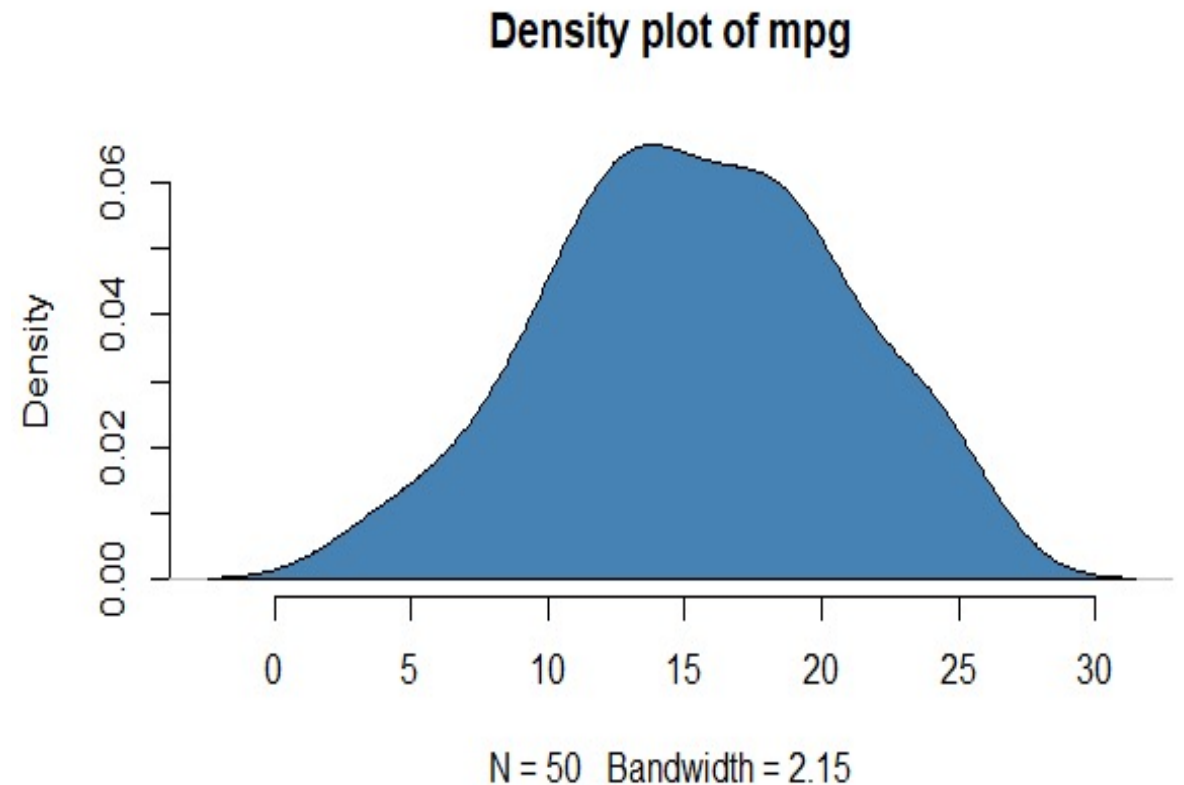
Compute the density data

- `dens <- density(cars$speed)`

plot density

- `plot(dens, frame = FALSE, col = "steelblue", main = "Density plot of mpg")`

`polygon(dens, col = "steelblue")`

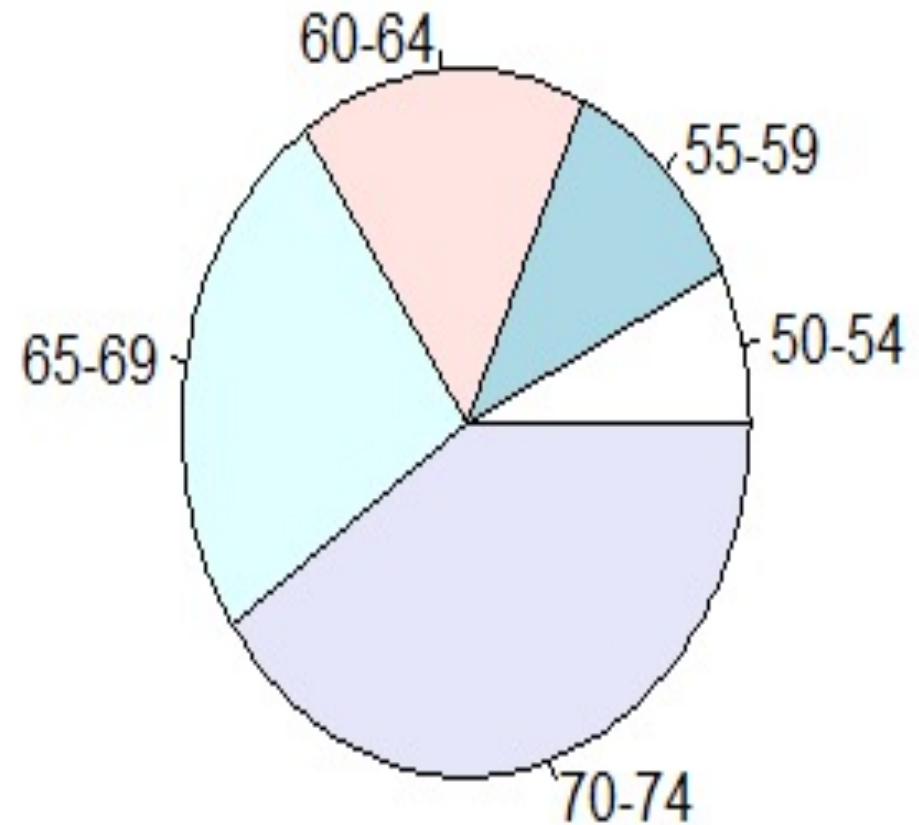


Pie chart:

- It is used to represent “categorical” variable in a circular diagram
- Sometime pie chart is more meaningful than the bar diagram for the categorical variable
- We can create Pie chart using “pie” command in R
- We can change the color of the Pie chart sections

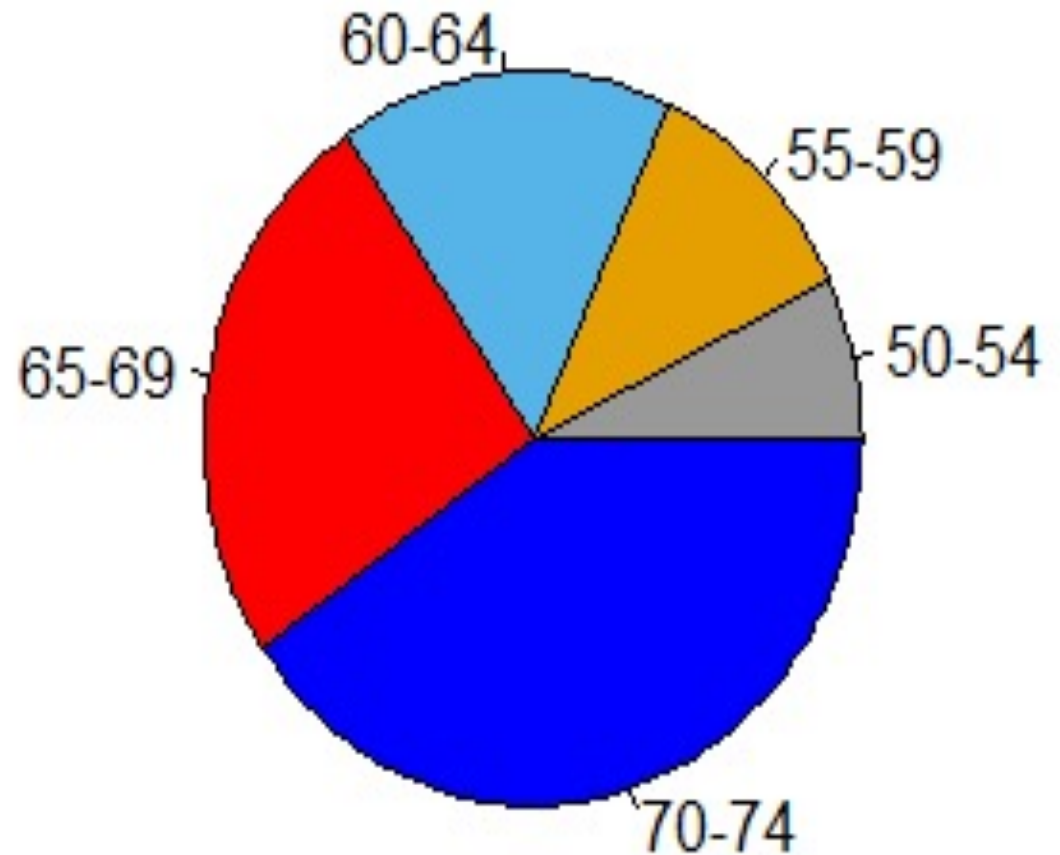
Pie chart: In-built “VADeaths” data

- `gd <- as.data.frame(VADeaths)`
- `pie(gd$`Rural Male`, labels = rownames(gd), radius = 1)`



Pie chart: Changing colors

- `gd <- as.data.frame(VADeaths)`
- `pie(gd$`Rural Male`, labels = rownames(gd), radius = 1, col = c("#999999", "#E69F00", "#56B4E9", "red", "blue"))`
- **How to show value or % inside the Pie chart slices?**
- **How to place a legend on the "topright" of this graph?**



Pie chart with

#Adding % and legend

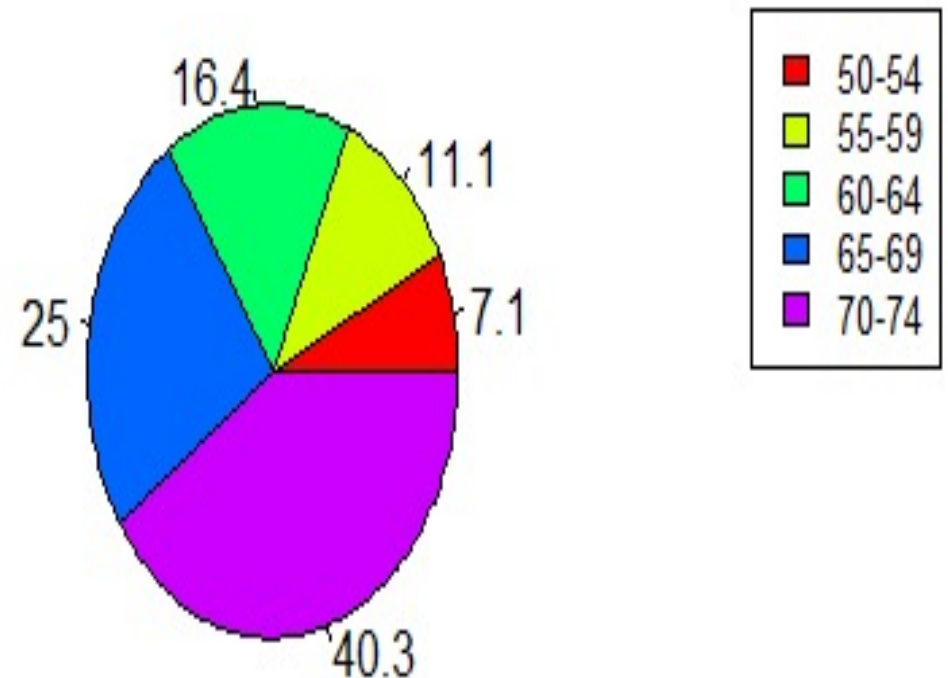
```
gd$piepercent<- round(100*gd$`Rural  
Male`/sum(gd$`Rural Male`), 1)
```

#Pie chart

```
pie(gd$`Rural Male`, labels =  
gd$piepercent, main = "% Deaths by  
Age groups for Rural Male",col =  
rainbow(length(gd$`Rural Male`)))
```

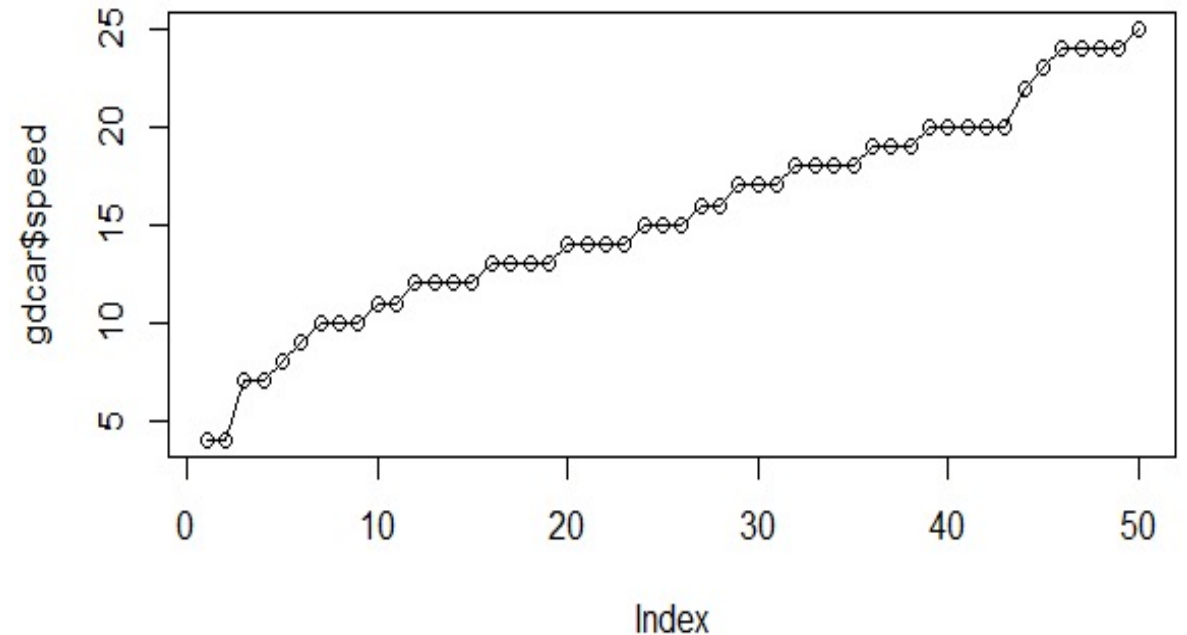
```
legend("topright", c("50-54","55-  
59","60-64","65-69", "70-74"), cex =  
0.8, fill = rainbow(length(gd$`Rural  
Male`)))
```

% Deaths by Age groups for Rural Male



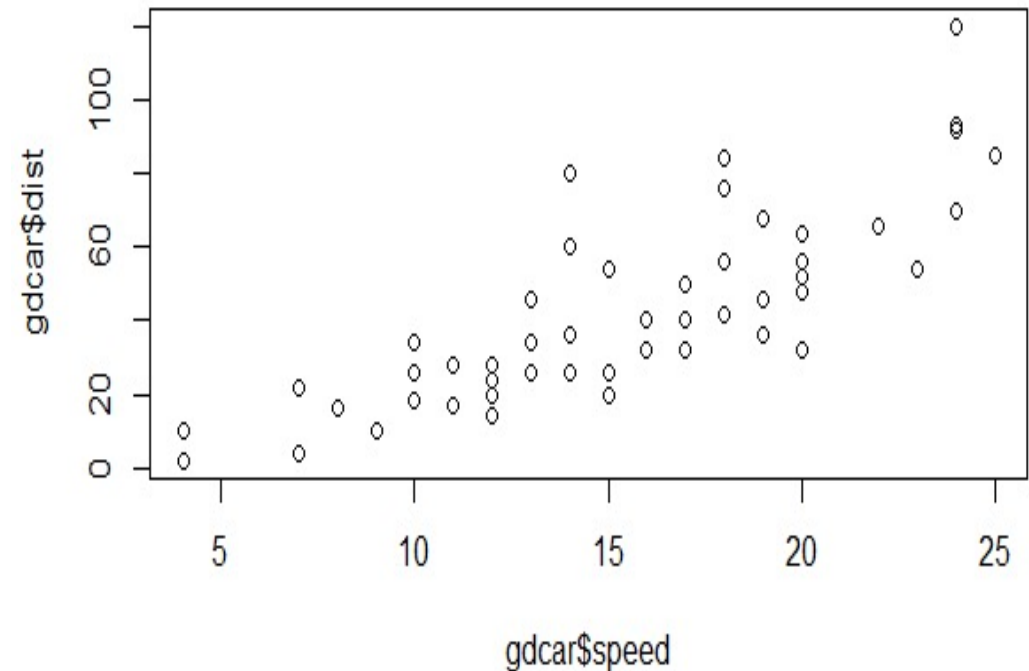
Line chart: We have already discussed it!

- Useful for time series data
- `plot(gdcars$speed)`
- `plot(gdcars$speed, type = "o")`
- Here R automatically created an index variable for x-axis starting from 1 and ending in 50!



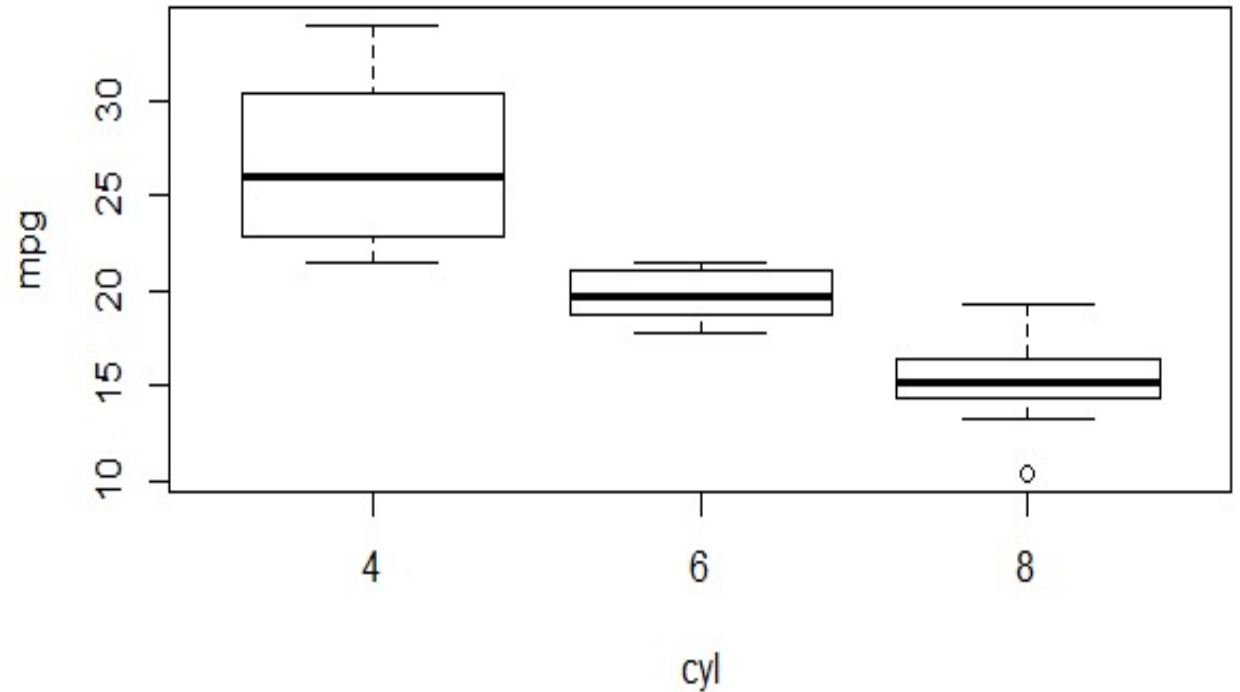
Scatterplot: We have already discussed it!

- `plot(gdcar$speed, gdcar$dist)`
- How to add title?
- How to change x-axis label?
- How to change y-axis label?
- **Which correlation coefficient is appropriate here?**



Boxplot: We have already discussed it!

- `boxplot(mpg ~ cyl, data = mtcars)`
- `boxplot(mpg ~ gear, data = mtcars, xlab = "Number of cylinders, ylab = "Miles Per Gallon", main "Mileage Data")`



Question/Queries?

Thank you!

@shitalbhandary