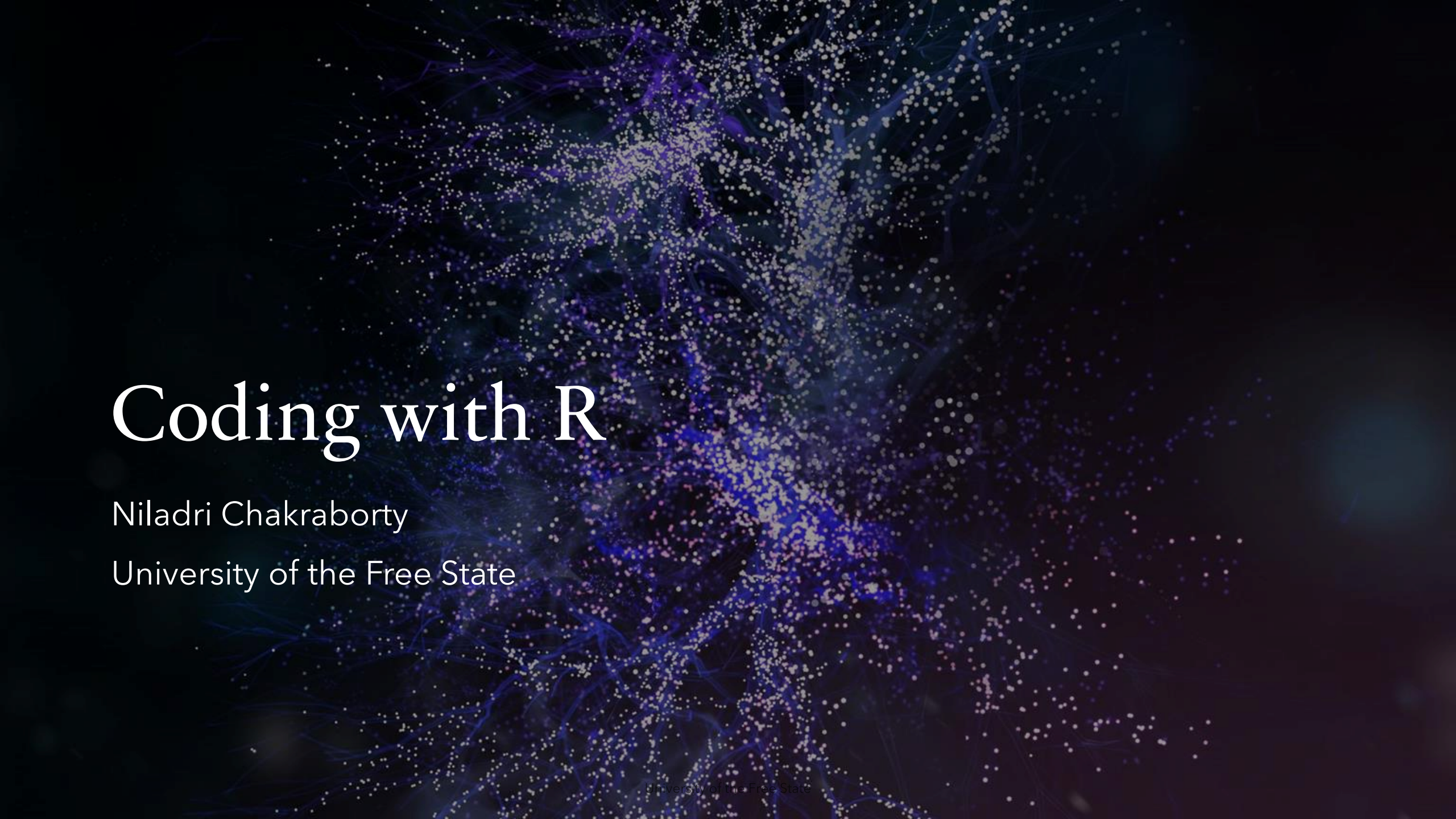


Coding with R

Niladri Chakraborty

University of the Free State

Data visualization

We shall be using one built-in dataset called the **air quality** dataset, which pertains to the daily air quality measurements in New York from May to September 1973.

This dataset consists of more than 100 observations on **6 variables**, i.e., **Ozone**(mean parts per billion), **Solar.R**(Solar Radiation), **Wind**(Average wind speed), **Temp**(maximum daily temperature in Fahrenheit), **Month**(month of observation) and **Day**(Day of the month).

Data visualization

Details:

Daily readings of the following air quality values for May 1, 1973 (a Tuesday) to September 30, 1973.

Ozone: Mean ozone in parts per billion from 1300 to 1500 hours at Roosevelt Island

Solar.R: Solar radiation in Langleys in the frequency band 4000--7700 Angstroms from 0800 to 1200 hours at Central Park

Wind: Average wind speed in miles per hour at 0700 and 1000 hours at LaGuardia Airport

Temp: Maximum daily temperature in degrees Fahrenheit at La Guardia Airport.

Data visualization

- data(airquality)
- We learnt head(airquality) shows first 6 observations.
- You can also use **head(airquality,n)** and **tail(airquality,n)** The head outputs the top **n** elements in the dataset while the tail method outputs the bottom **n**.

Data visualization

- #Example
- # suppose we need to print first 20 outputs from airquality data
- data(airquality)
- n = 20
- # for the first 20 observations
- head(airquality,n)
- # for the last 20 observations
- tail(airquality,n)

Data visualization

- **summary(**airquality**)**

- #save ozone data
- data.ozone = airquality$Ozone

- #save temp data
- data.temp = airquality$Temp

- #save wind data
- data.wind = airquality$Wind

Data visualization

- # if there is any missing value
- is.na(data.ozone)
- is.na(data.temp)
- is.na(data.wind)

- # If there is TRUE in the output, it means there is missing value. So there is missing value in ozone data.
- # Now we remove the missing value from the ozone data
- data.ozone = data.ozone[!is.na(data.ozone)]
- data.ozone

Data visualization

We start with loading the required packages.

Package ggplot2 is included in the tidyverse package.

Package ggplot2 is a plotting package that makes it simple to create complex plots from data in a data frame.

Data visualization

- To build a ggplot, we will use the following basic template that can be used for different types of plots:

- #ggplot(data = <DATA>, mapping = aes(<MAPPINGS>)) + <GEOM_FUNCTION>()

- #use the ggplot() function and bind the plot to a specific data frame using the data argument
- ggplot(data = airquality)

Data visualization

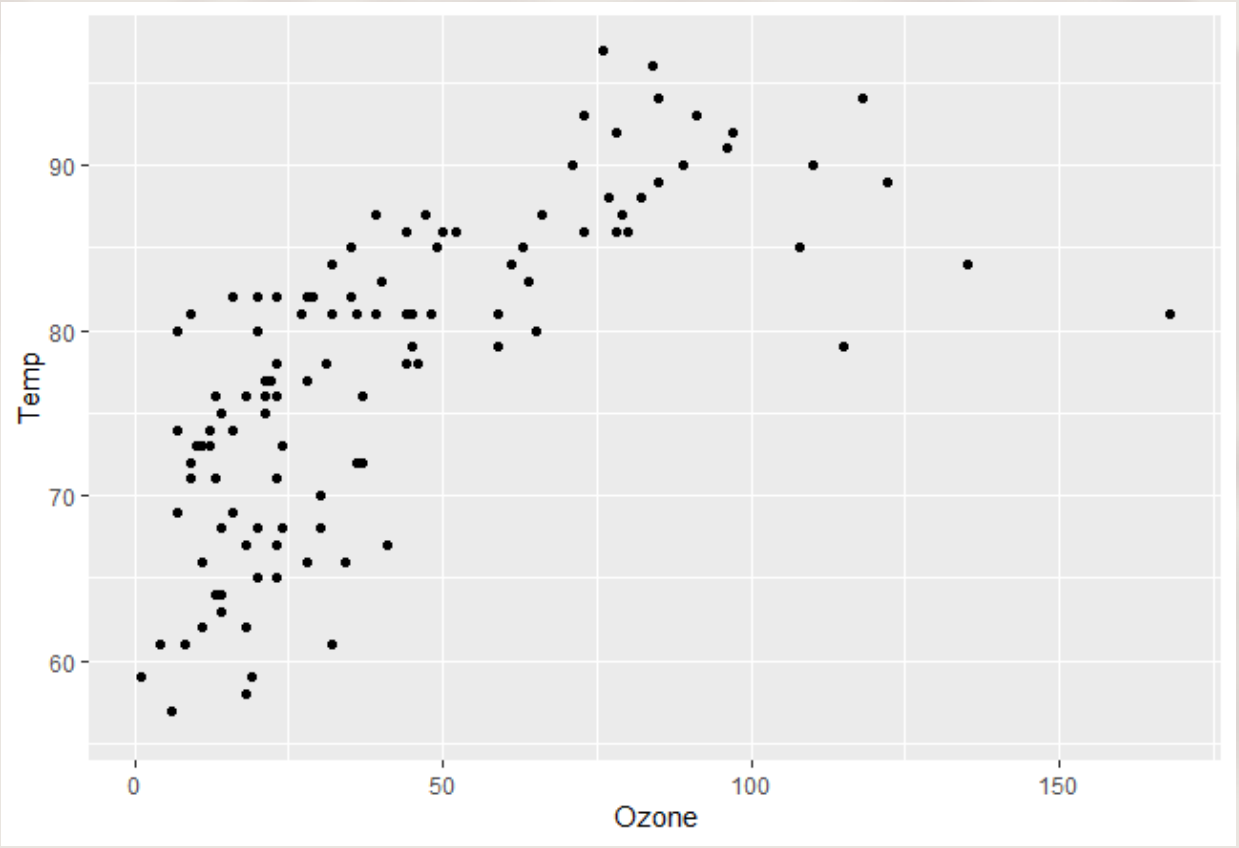
- Define a mapping using aes() function, by selecting variables and specifying how to present them in the graph.

- Add **'geoms'** – this function decides the graphical representations of the data in the plot (points, lines, bars).

- To add a geom to the plot use '+' operator.
- Let's use geom_point() first to create a scatter plot.

Data visualization

- ggplot(data = airquality, mapping = aes(x = Ozone, y = Temp))+geom_point()
- Temperature vs. Mean Ozone parts scatter plot looks like →
-



Data visualization

- What do we conclude by a quick look at the plot ?
- Average ozone parts and temperature correlated.

- # Let us use the Iris data.
- # The **iris** dataset is provided natively by R, we call it a built-in dataset.
- # So, we do not need to use a package to get this data.

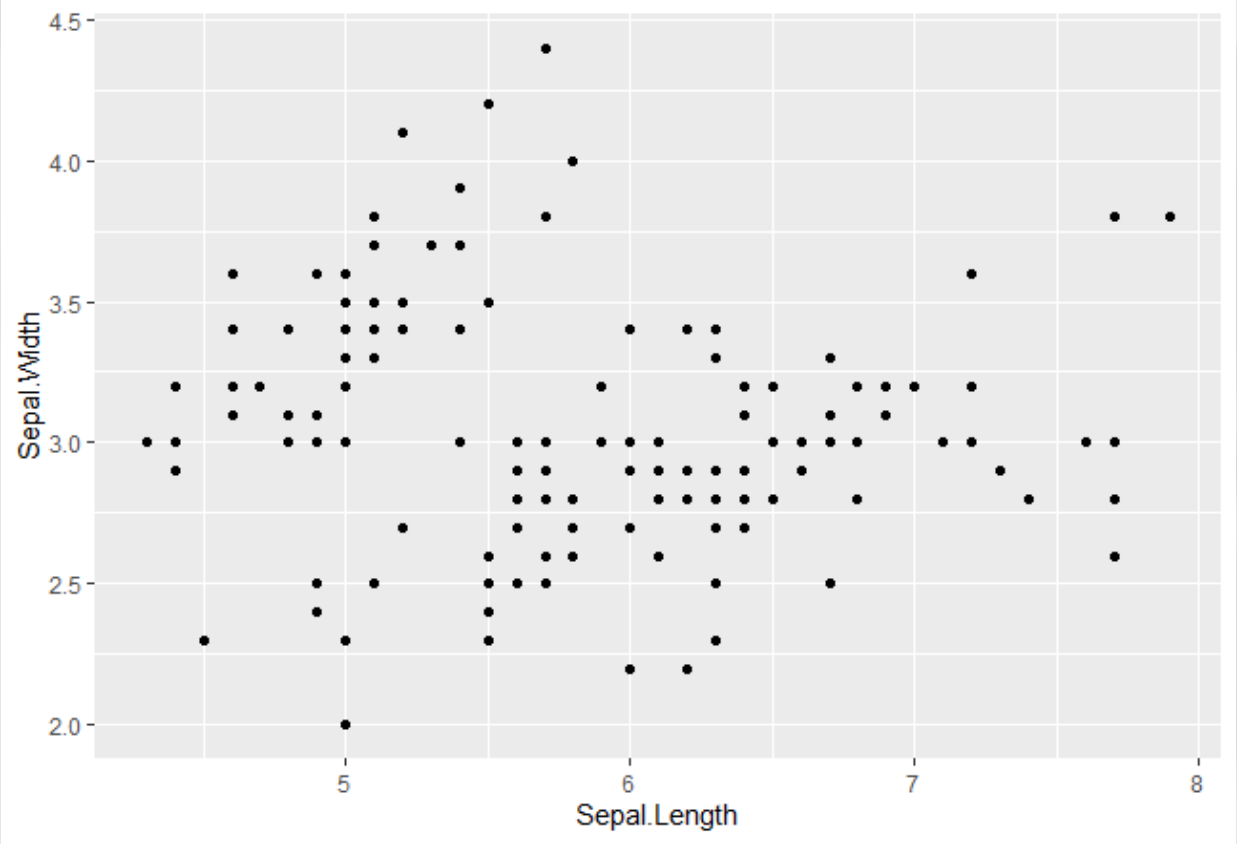
Data visualization

- head(iris)

- # basic scatterplot
- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) +
- geom_point()

Data visualization

- Sepal.width vs. Sepal.Length scatter plot looks like →

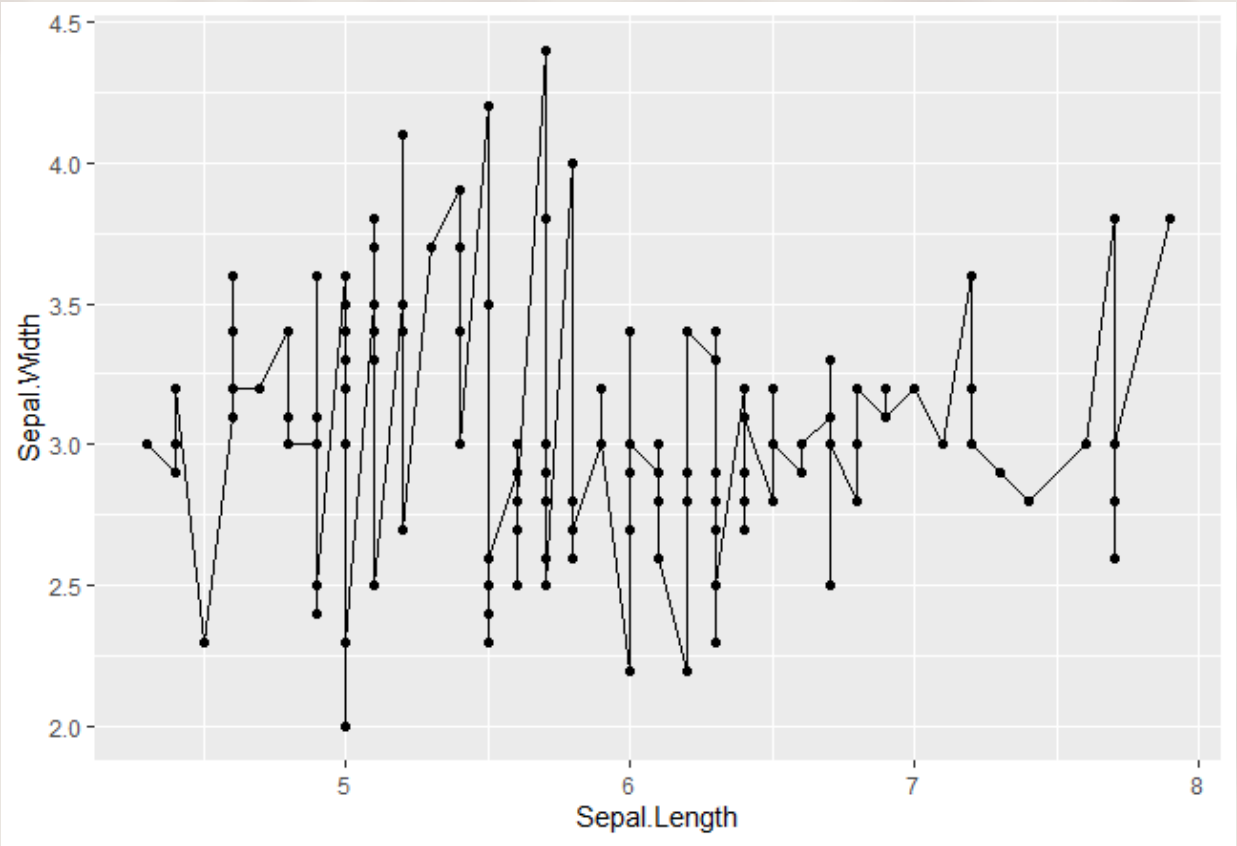


Data visualization

- #To create a connected scatter plot
- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) +
- geom_point()+geom_line()

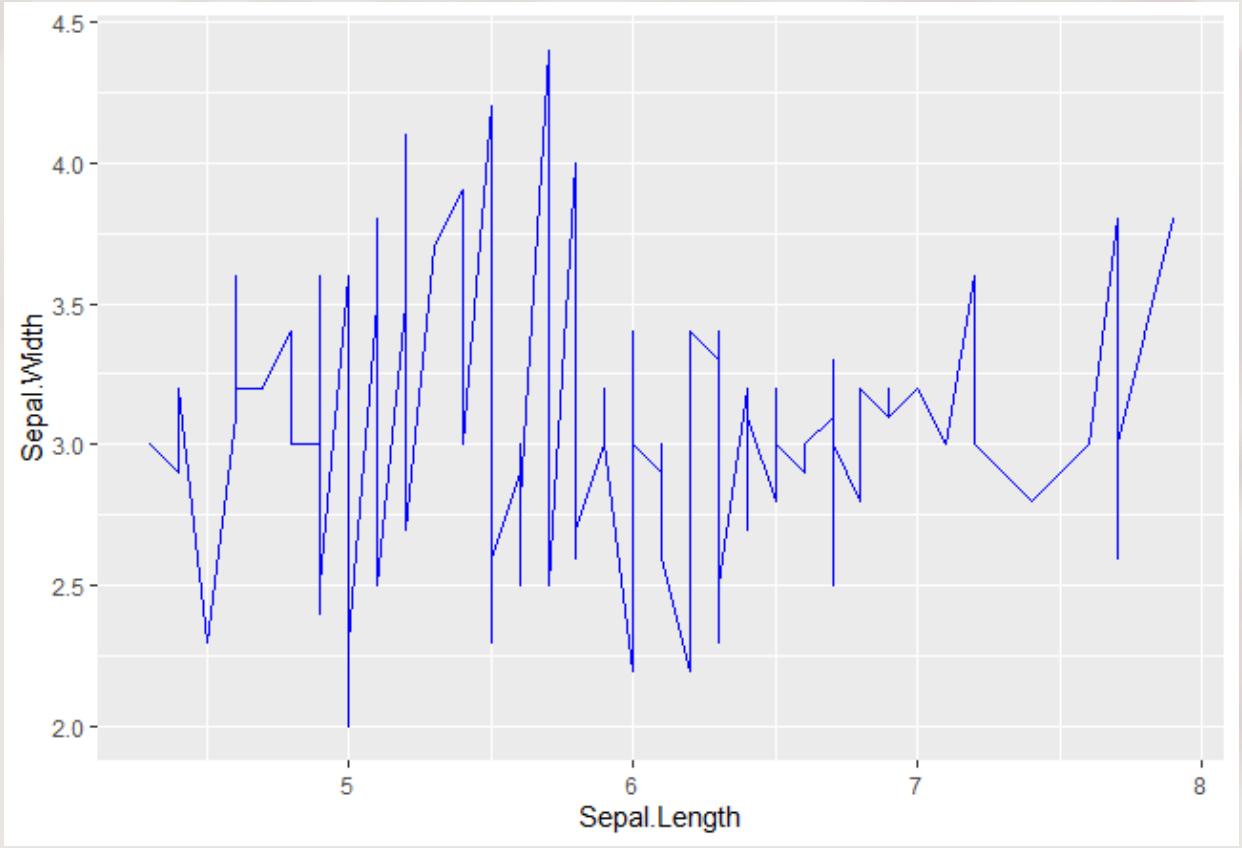
Data visualization

- Sepal.width vs. Sepal.Length connected scatter plot looks like →



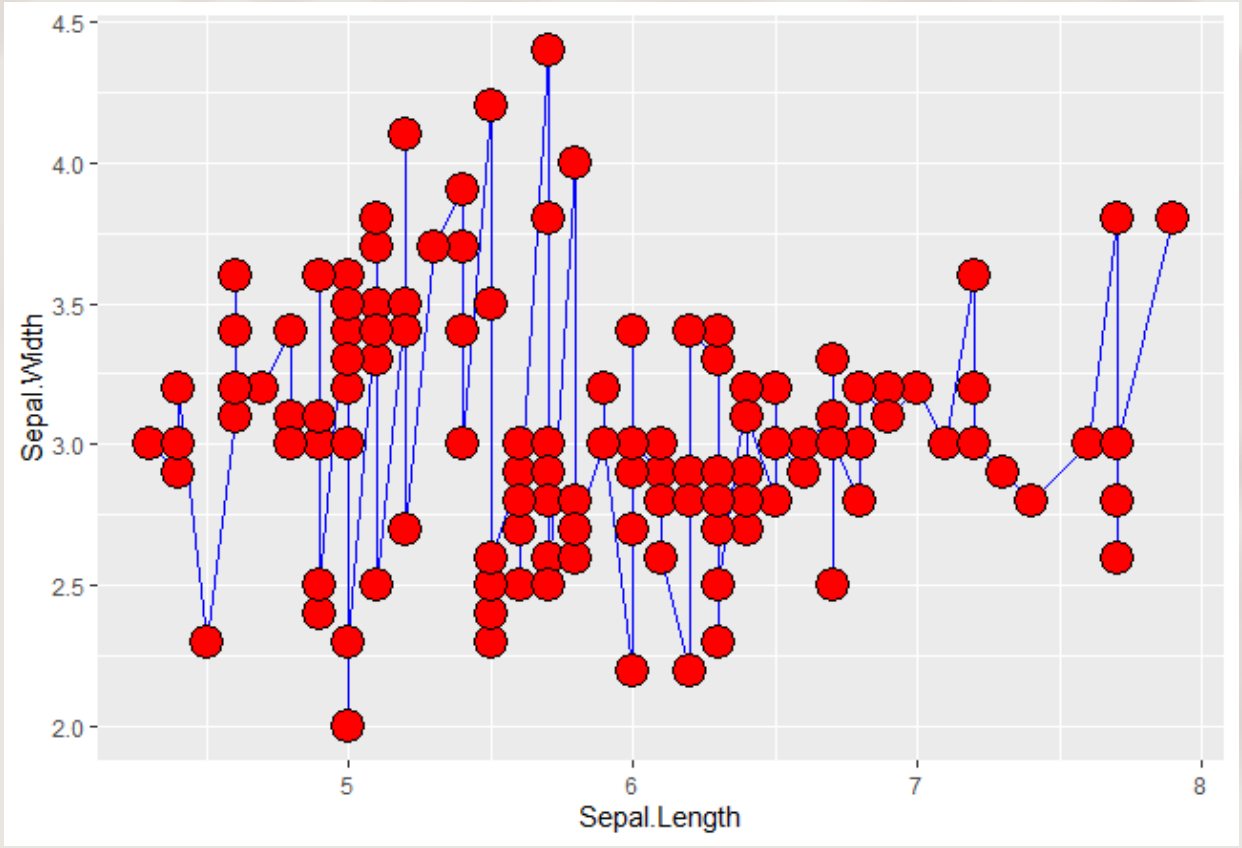
Data visualization

- Customize the connected scatterplot
- You can add a color:
- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) + geom_line(color="blue")



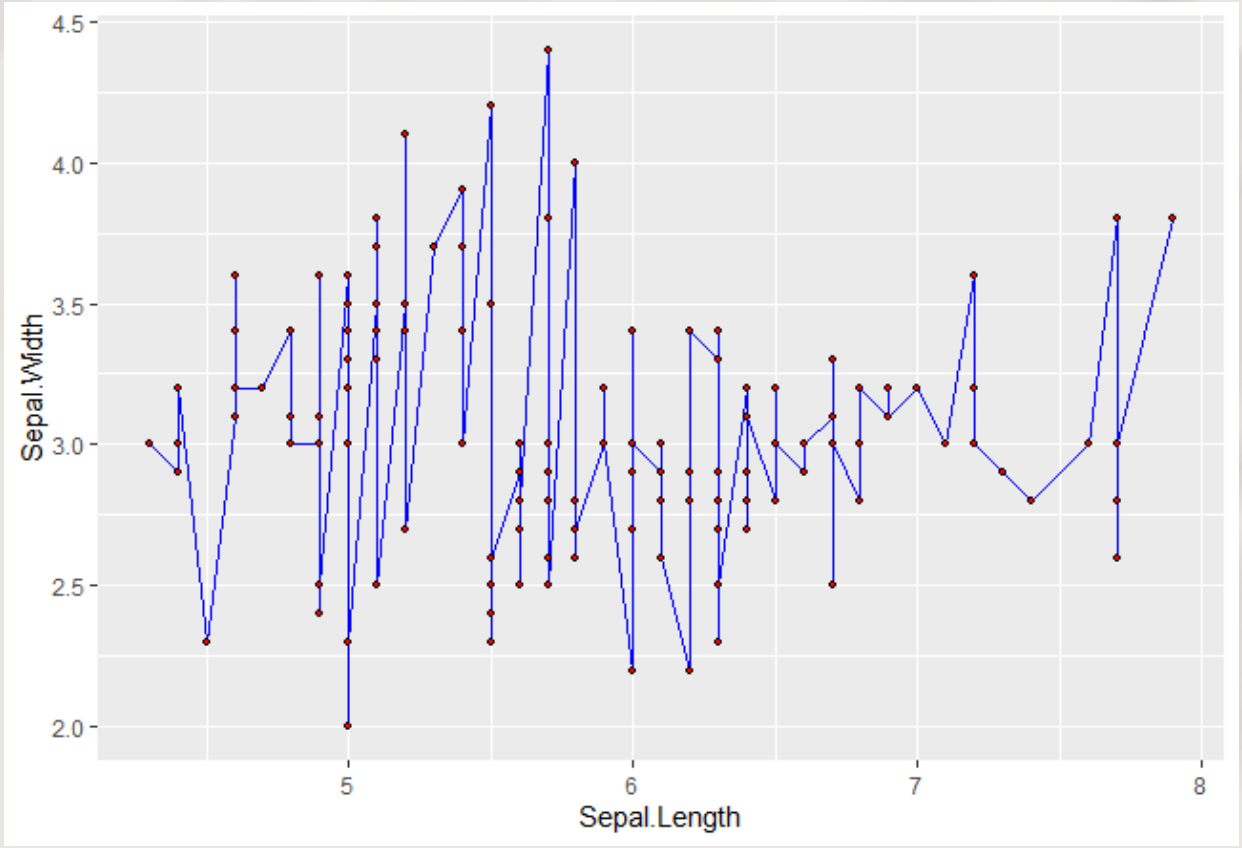
Data visualization

- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) + geom_line(color="blue") + geom_point(shape=21, color="black", fill="red", size=6)



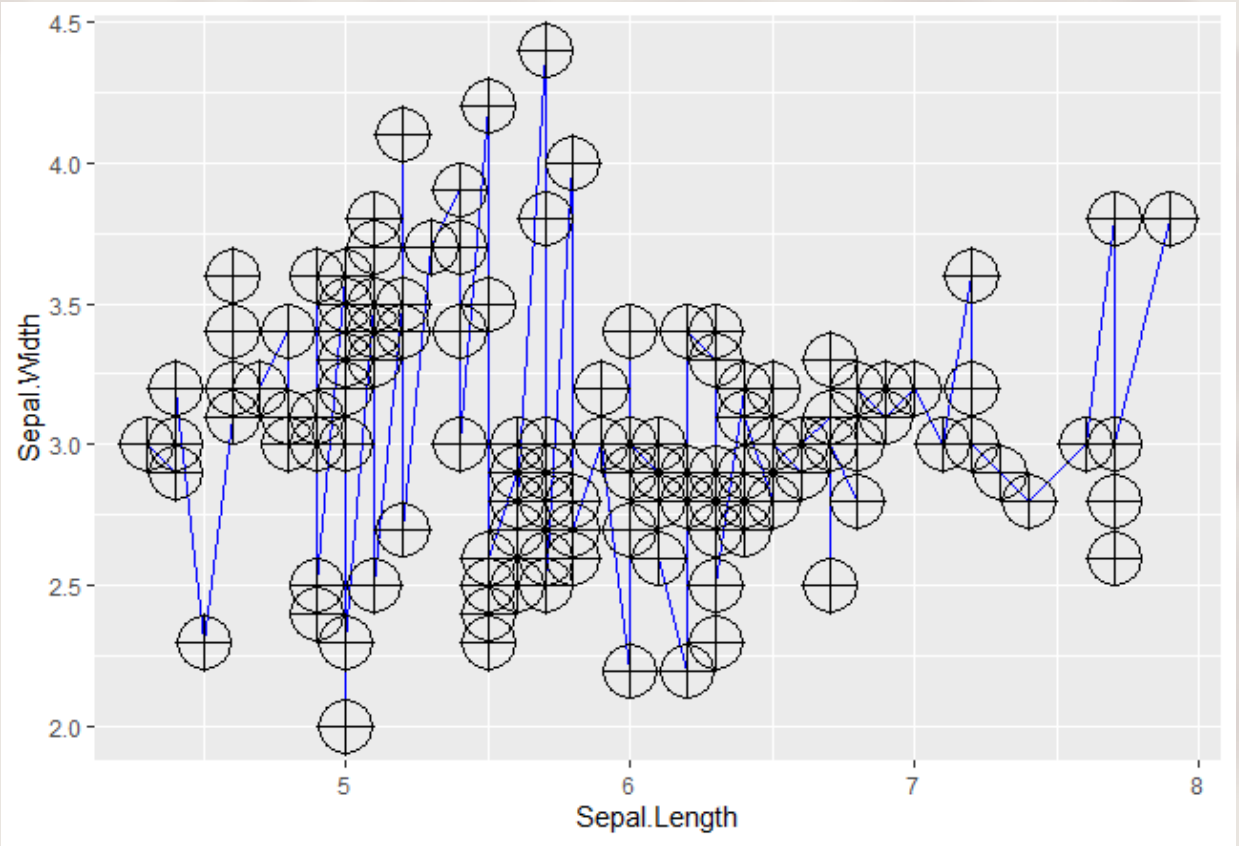
Data visualization

- Change the size of the points
- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) + geom_line(color="blue") + geom_point(shape=21, color="black", fill="red", size=1)



Data visualization

- Change the size of the points
- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) + geom_line(color="blue") + geom_point(shape=21, color="black", fill="red", size=1)



Data visualization

- **Custom** the **general theme** with the theme_ipsum() function of the **hrbrthemes package**.
- Add a title with ggtitle().
- Custom circle and line with arguments like shape, size, color and more.

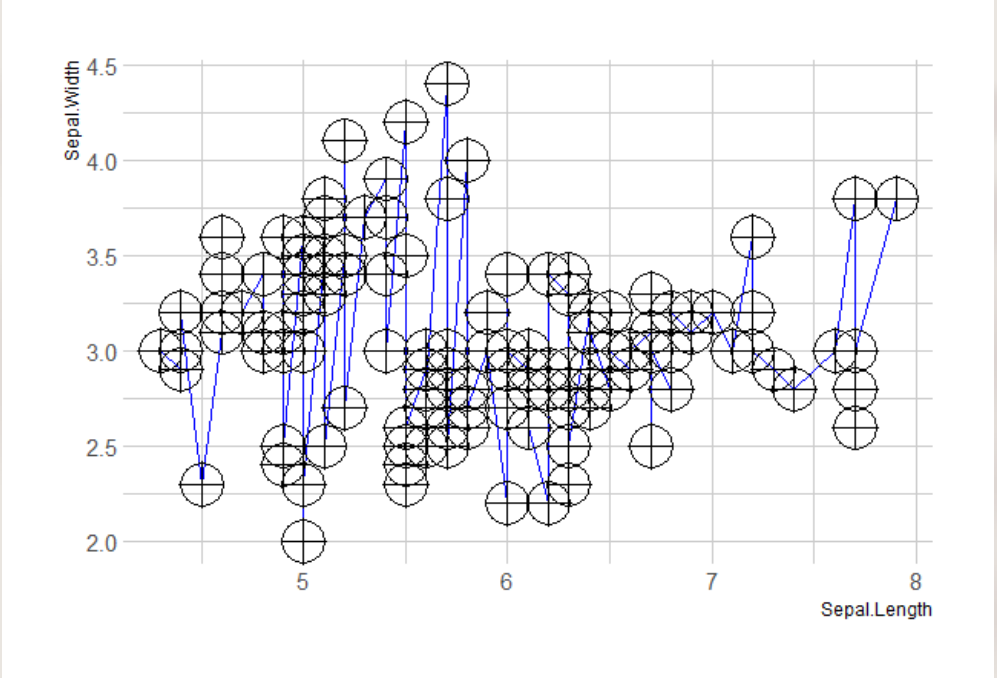
Data visualization

- # Libraries
- library(tidyverse)
- library(hrbrthemes)
- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) +

  geom_line(color="blue") +

  geom_point(shape=10, color="black", fill="red", size=10) +

  theme_ipsum()



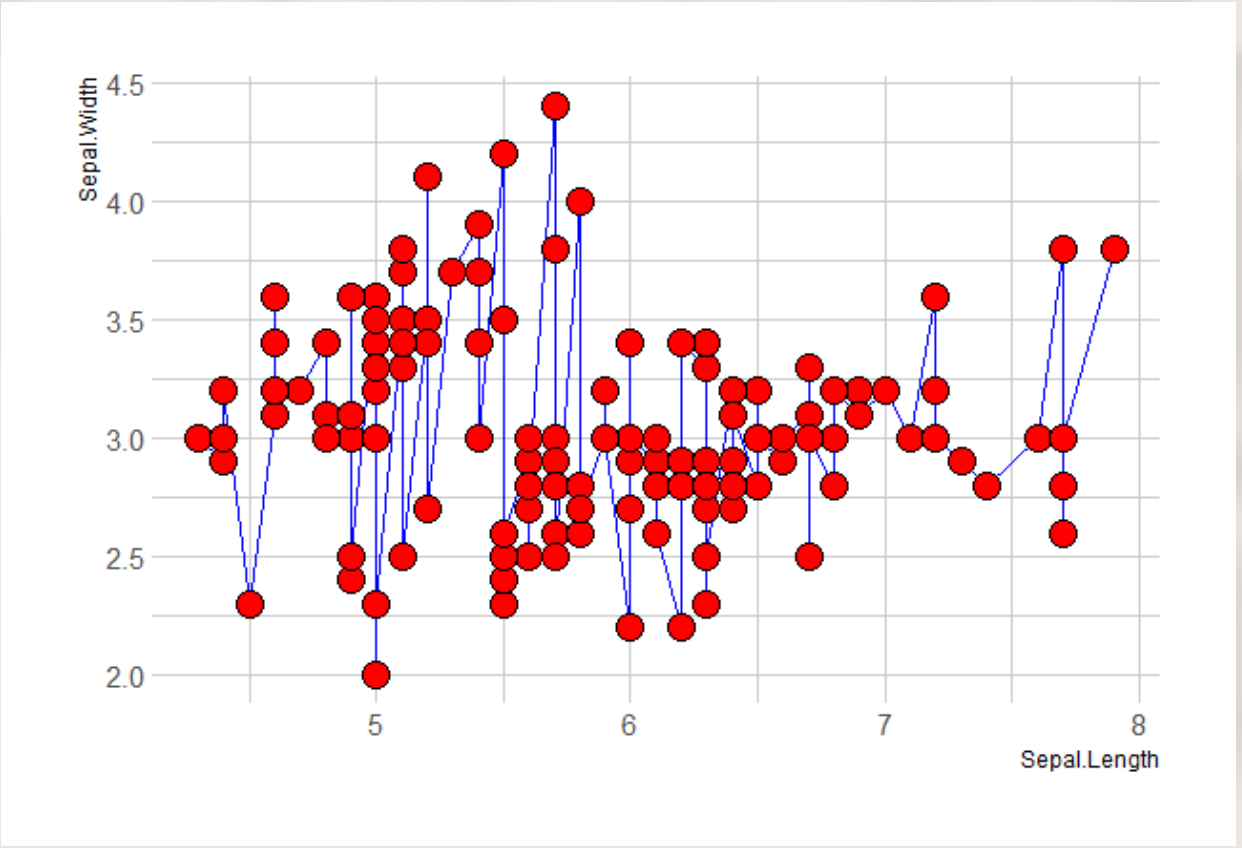
Data visualization

- Change size and shape with the general theme as the background:
- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) +

geom_line(color="blue") +

geom_point(shape=21, color="black", fill="red", size=5) +

theme_ipsum()



Data visualization

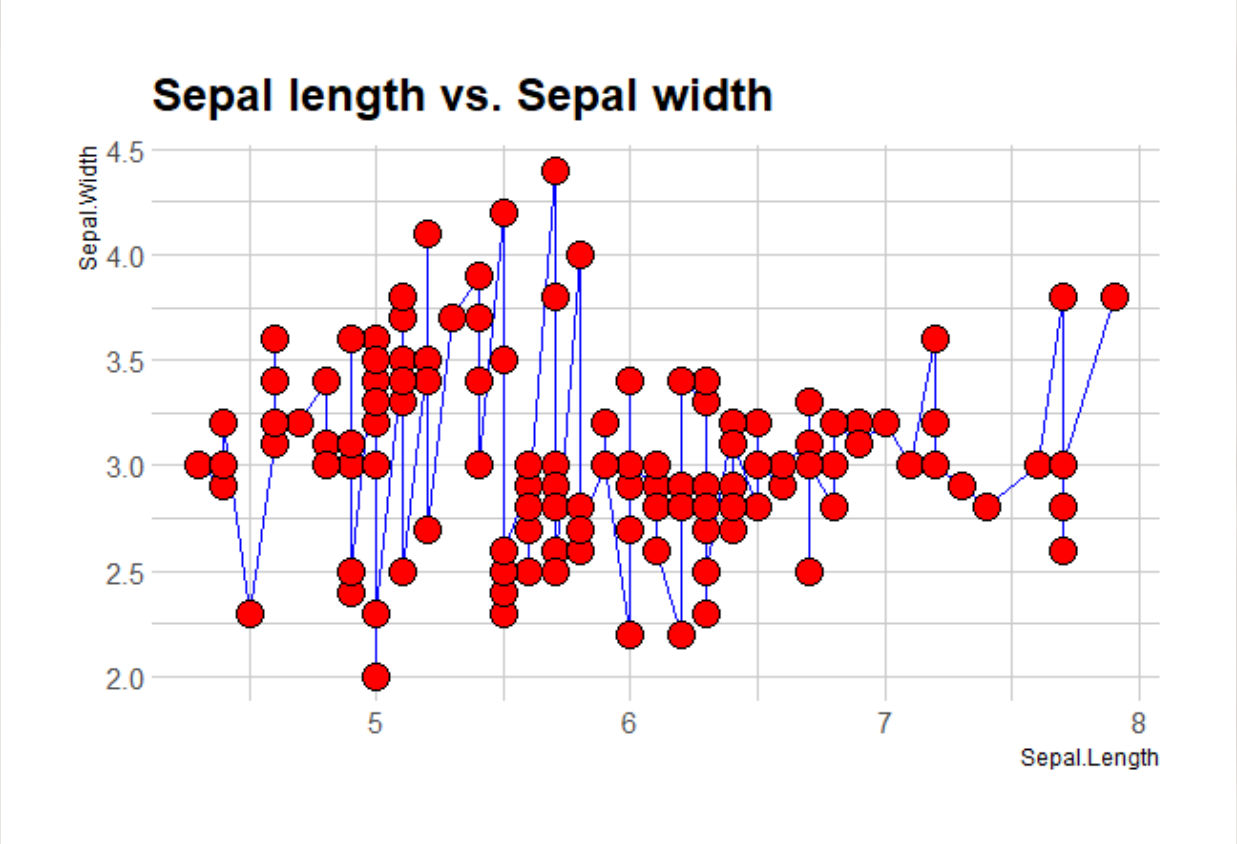
- Add title with the ggtitle() function
- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) +

  geom_line(color="blue") +

  geom_point(shape=21, color="black", fill="red", size=5) +

  theme_ipsum() +

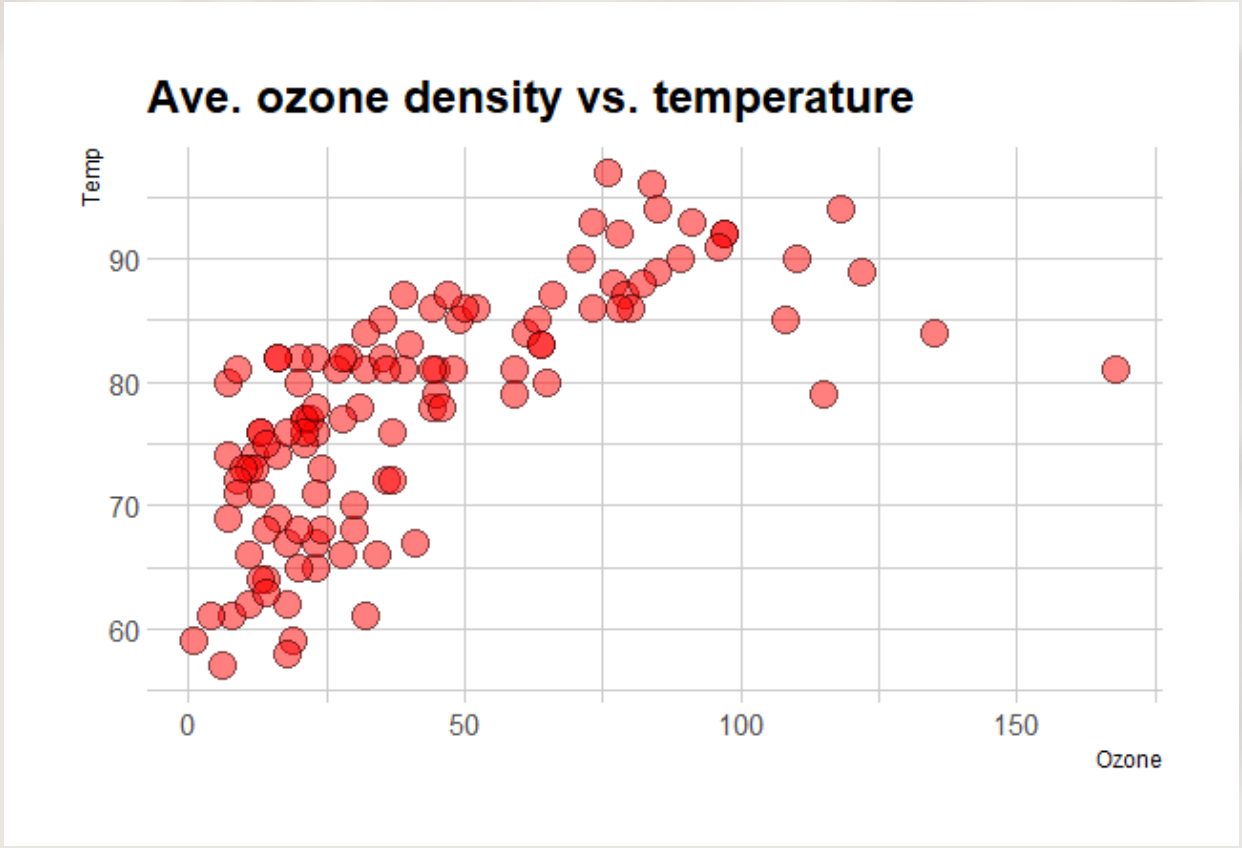
  ggtitle("Sepal length vs. Sepal width")



Data visualization

- ggplot(data = airquality, mapping = aes(x = Ozone, y = Temp))+
- geom_point(alpha=0.5,shape=21, color="black", fill="red", size=5) +

theme_ipsum() + ggtitle("Ave. ozone density vs. temperature")



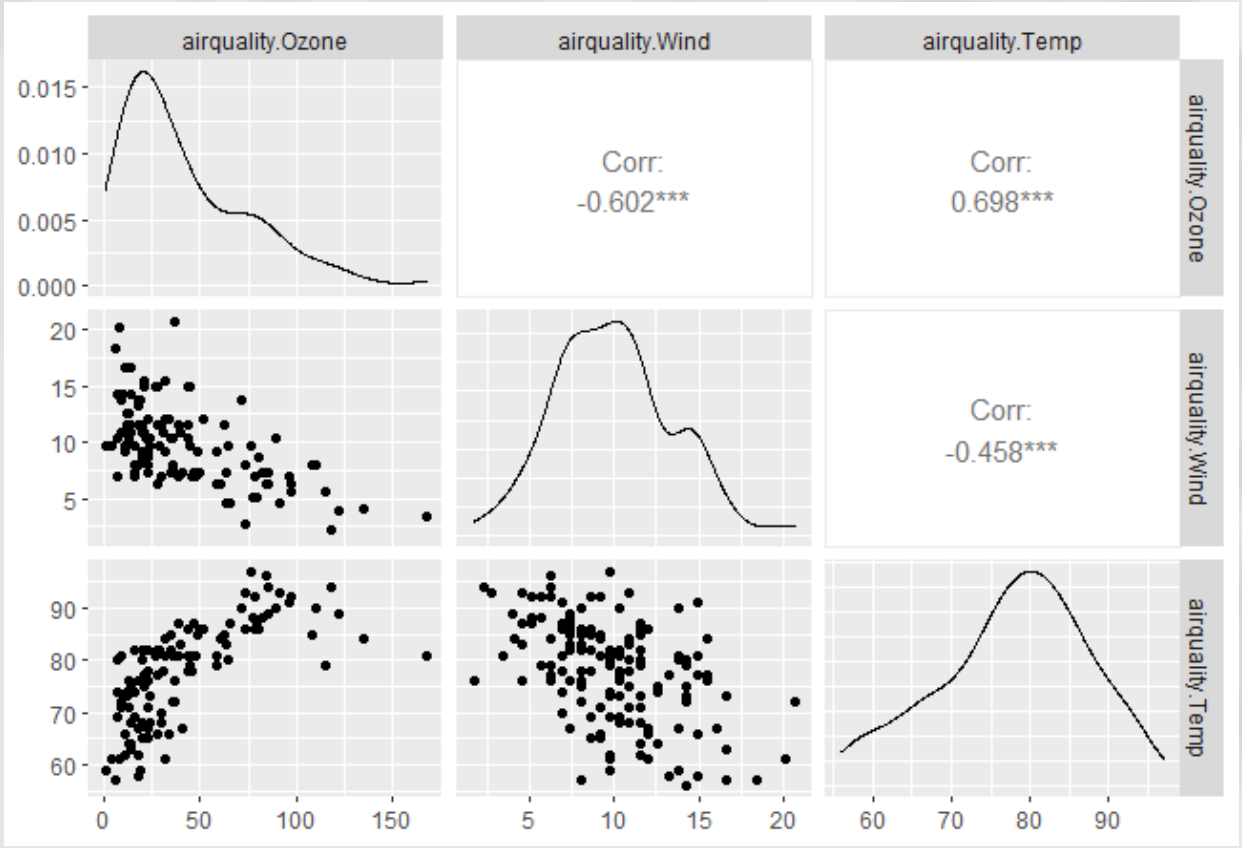
Data visualization

- ## package GGally is has R functions for great correlation plots

- library(GGally)
- data = data.frame(airquality$Ozone,airquality$Wind, airquality$Temp)
- ggpairs(data)

Data visualization

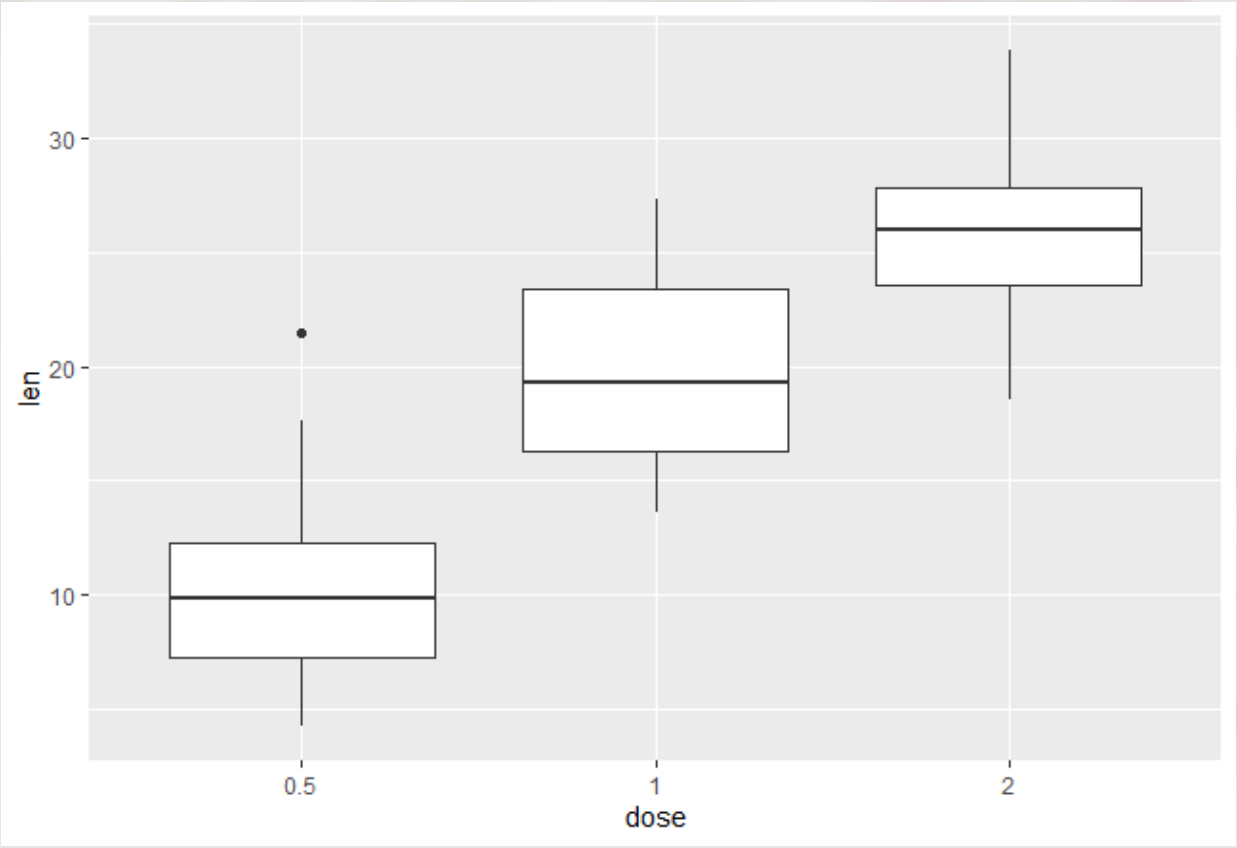
- Scatterplots of each pair of numeric variable are drawn on the left part of the figure.
- Pearson correlation is displayed on the right.
- Variable distribution is available on the diagonal.



- To create a boxplot we use a dataset called ToothGrowth.
- Convert the variable dose from a numeric to a factor variable
  ToothGrowth$dose <- as.factor(ToothGrowth$dose)
  head(ToothGrowth)

Data visualization

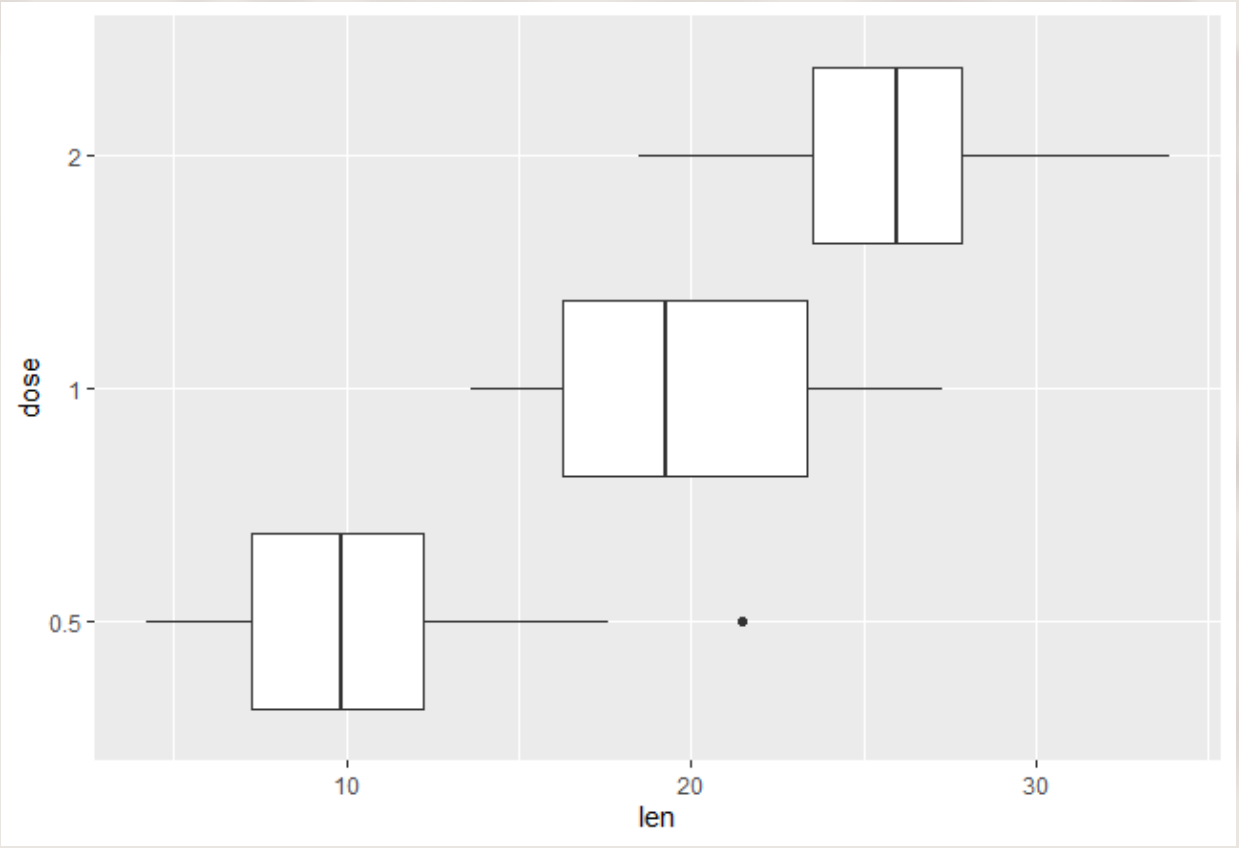
- ggplot(ToothGrowth, aes(x=dose, y=len)) + geom_boxplot()



Data visualization

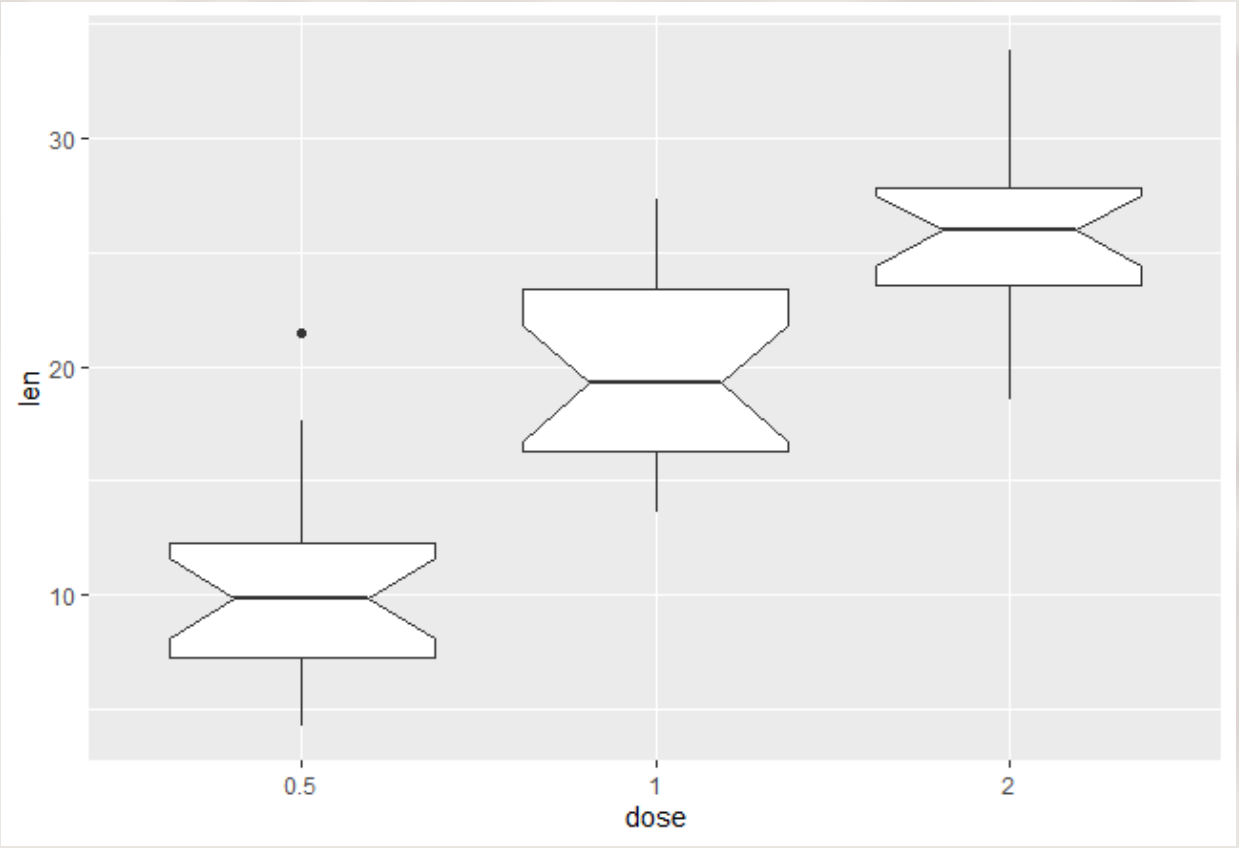
- # Rotate the box plot
- ggplot(ToothGrowth, aes(x=dose, y=len)) +

geom_boxplot() + coord_flip()



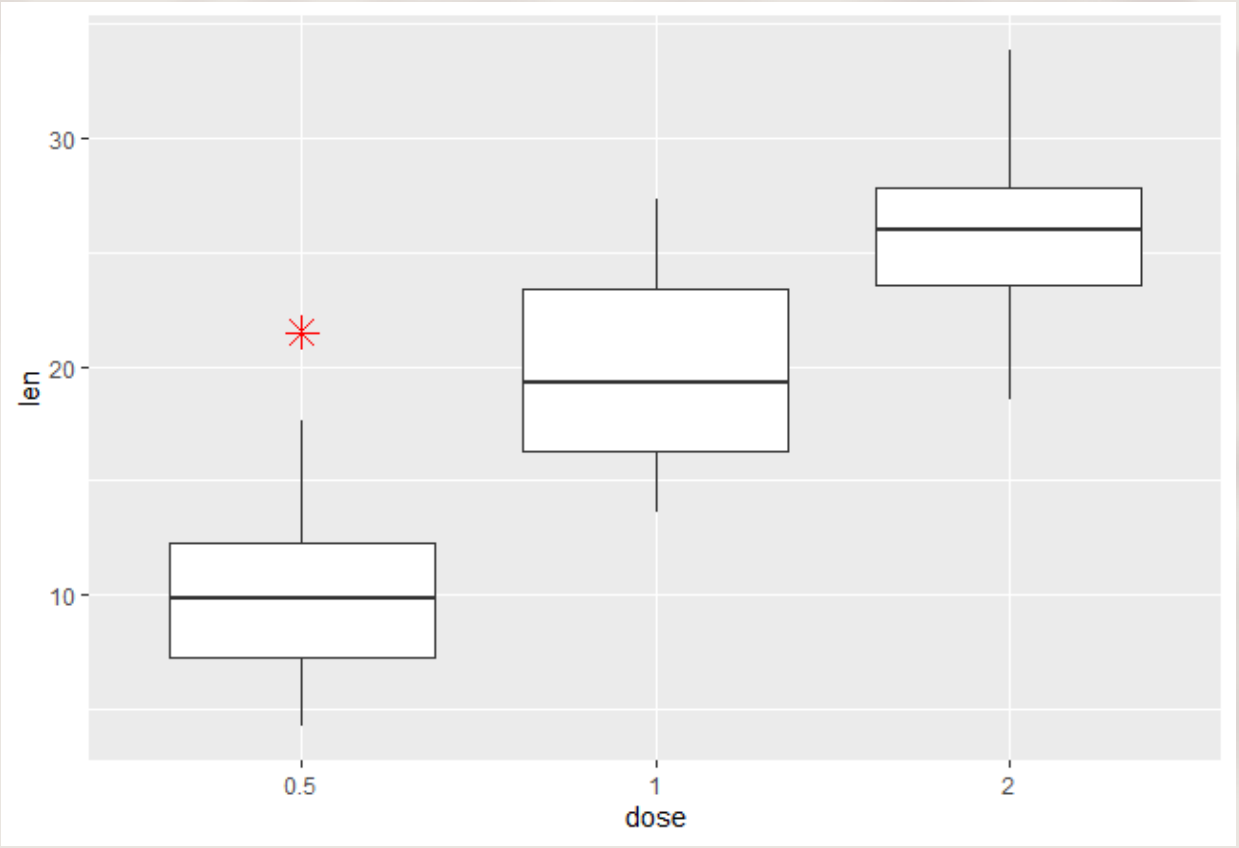
Data visualization

- # Notched box plot
- ggplot(ToothGrowth, aes(x=dose, y=len)) +
- geom_boxplot(notch=TRUE)



Data visualization

- # Change outlier, color, shape and size
- ggplot(ToothGrowth, aes(x=dose, y=len)) +
- geom_boxplot(outlier.colour="red", outlier.shape=8,
- outlier.size=4)



Data visualization

- # Change outlier, color, shape and size
- ggplot(ToothGrowth, aes(x=dose, y=len)) + geom_boxplot(outlier.colour="red", outlier.shape=11, outlier.size=9)