This course was developed as a part of the VLIR-UOS Cross-Cutting projects:

- Statistics: 2011-2016, 2017.
- Statistics: 2017.
- Statistics for development : 2018-2022.
- The >eR-BioStat ITP: 2024-2026.

The >eR-Biostat initiative

Introduction to Visualization using the R package `ggplot2`

Developed by
Thi Huyen Nguyen and Ziv Shkedy
(Hasselt University, Belgium)

LAST UPDATE: 05/2025

Visit us on Facebook ER-BioStat

GitHub https://github.com/eR-Biostat

twitter @erbiostat

Topics

1. Main focus of the course: Introduction to data visualization and EDA using `ggplot2`.
2. EDA and visualization for location, spread & shape at an introduction level.
3. Introduction to the R package `ggplot2`.

4. Many examples for illustration on real datasets:
   1. Different datasets.
   2. How to do it in R.

Datasets

- Data are given as a part of R programs for the course.
- Some datasets are a part of R packages that are needed to be installed.
- For this course we use:
  - The `airquality` data.
  - The `NHANES` data.
  - The `singer` data.
  - The `mtcars` data.
  - The `boston` data.
  - The `fatihful` data.

- All datasets are a part of R packages or R datasets.
- To access some of the datasets, R packages should be installed.
- All datasets are a part of the two Rmd programs related o the course.

Software

- Data analysis using R:
  - R studio.
  - R markdown.
  - Specific R packages.

We will cover these topics tomorrow.

Software

- R functions for visualization:
  - Basic R function for visualization.
  - The `lattice` package.
  - Mainly: `ggplot2`.

- HTML file (online):
  - **`eR_Biostat_Kampala_VD1_2025.html`**.
  - **`eR_biostat_Kampala_VD2_2025.html`**.
- R program for the examples is available online:
  - **`Visualization_intro.Rmd`.**
  - **`er.prog4c_2_VT_2025V1.Rmd`.**

The HTML file

er_prog4c_2_VT_2025-V1.knit

File C:/Ziv_Temp_2023/Workshop_Vietnam_2025/ShortCourse/OnlineBook/er_prog4c_2_VT_2025-V1.html

uhasselt.be bookmarks

All Bookmarks

content

Code

26-04-2025 >eR-BioStat

Visualizing Data and Exploratory Data analysis using `ggplot2` in R

Ziv Shkedy and Thi Huyen Nguyen

Show

1. Introduction

> "Exploratory data analysis can never be the whole story, but nothing else can serve as the foundation stone - as the first step"
>
> — John W. Tukey (1977)

Location, Spread and Shape in univariate data

In this course, we focus on descriptive measures, numerical and graphical, to characterize and visualize the features of a particular univariate distribution. The following three main factors are usually used to specify a particular distribution:

- Location
- Spread
- Shape

Each of these control different characteristics of a distribution.

R datasets for illustraions

In order to simplify the usage of slides, the data we used for illustrations are R datasets. We give a short description of each data in the relevant slides. * More details can be found with `help(dataset)` or (for datasets of the first part) in

- The singers data: singers.
- The airquality data: airquality.
- The cars data: mtcars.
- The Old Faithful Geyser Data: oldfaithful.
- The Boston data: boston.

Type here to search

ENG 15:21 7/05/2025

The HTML file

er_prog4c_2_VT_2025-V1.knit

File C:/Ziv_Temp_2023/Workshop_Vietnam_2025/ShortCourse/OnlineBook/er_prog4c_2_VT_2025-V1.html

uhasselt.be bookmarks

All Bookmarks

For our example, we focus on `mpg`, `hp` and `cyl`.

The `mtcars` data

Bivariate data (miles/(US) per gallon, horsepower). We wish to produce a basic scatterplot: mgp Vs. h using the R function `plot()`.

Show

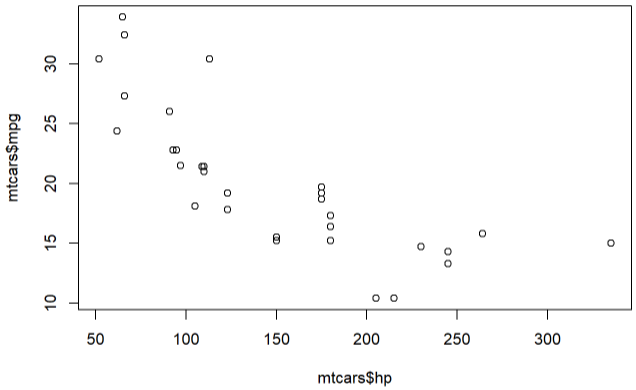


Figure 1: Miles/(US) per gallon vs. Horsepower

Linear regression

We consider a simple linear regression model of the form

$$\mathrm{mpg}_i = \beta_0 + \beta_1 \times \mathrm{hp}_i + \varepsilon_i$$

. In R, we can fit the model using the `lm()` function.

Show

The fitted model

Show

Example of a plot in the HTML

Type here to search

ENG 15:22 7/05/2025

The HTML file

er_prog4c_2_VT_2025-V1.knit

File C:/Ziv_Temp_2023/Workshop_Vietnam_2025/ShortCourse/OnlineBook/er_prog4c_2_VT_2025-V1.html

uhasselt.be bookmarks

All Bookmarks

For our example, we focus on mpg, hp and cyl.

The mtcars data

Bivariate data (miles/(US) per gallon, horsepower). We wish to produce a basic scatterplot: mgp Vs. h using the R function plot().

Hide

```
plot(mtcars$hp,mtcars$mpg)
```

The R code to produce the example.

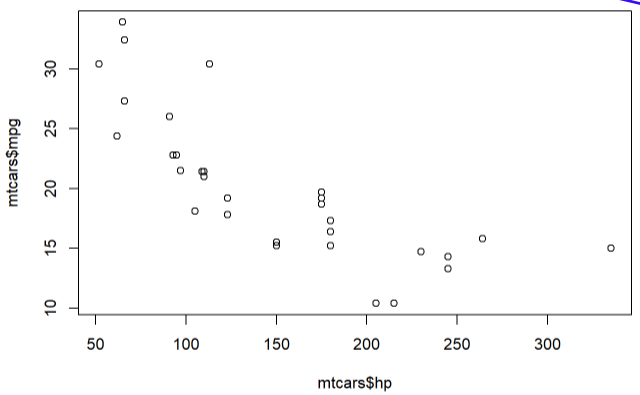


Figure 1: Miles/(US) per gallon vs. Horsepower

Linear regression

We consider a simple linear regression model of the form

$$\mathrm{mpg}_i = \beta_0 + \beta_1 \times \mathrm{hp}_i + \varepsilon_i$$

. In R, we can fit the model using the lm() function.

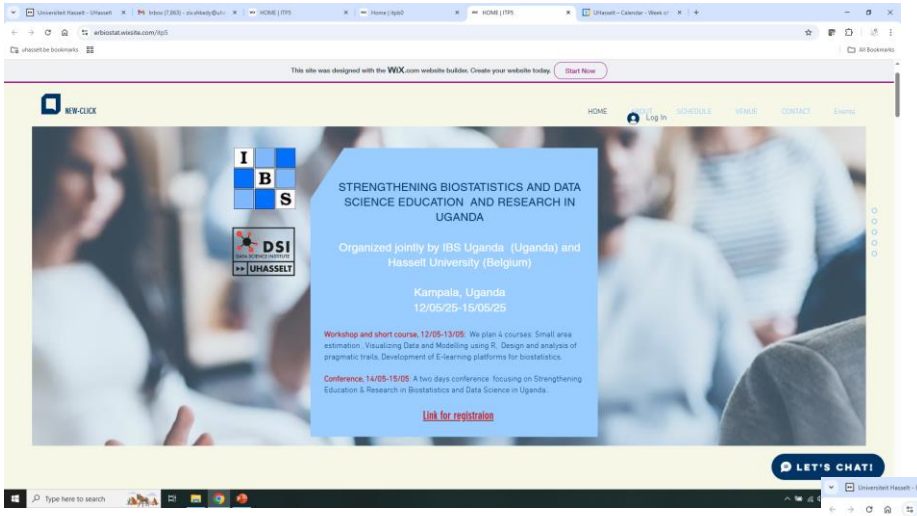
Show

The fitted model

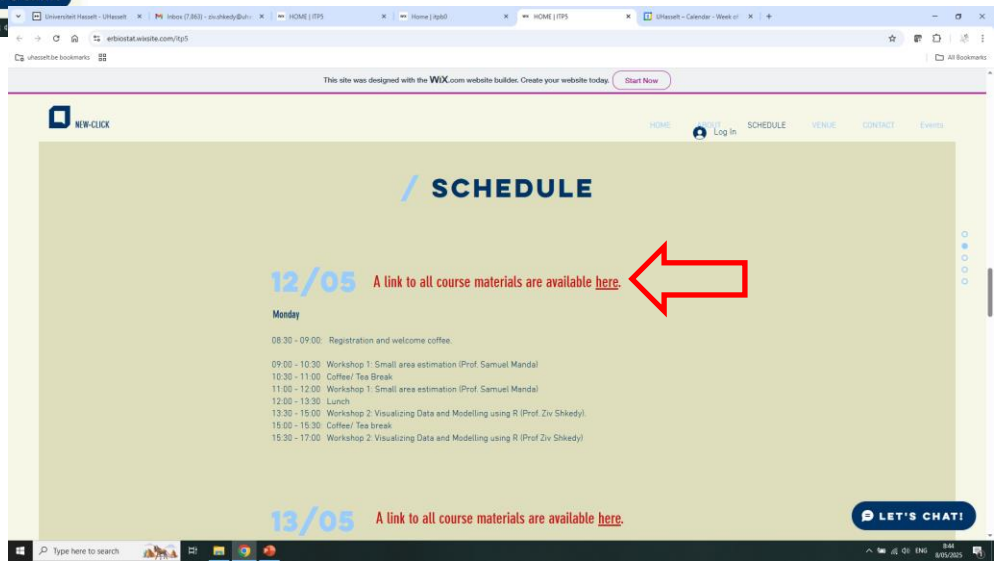
Type here to search

ENG 15:23 7/05/2025

Conference’s website



https://erbiostat.wixsite.com/itp5



Course materials online

https://erbiostat.wixsite.com/itpb0

Course materials online

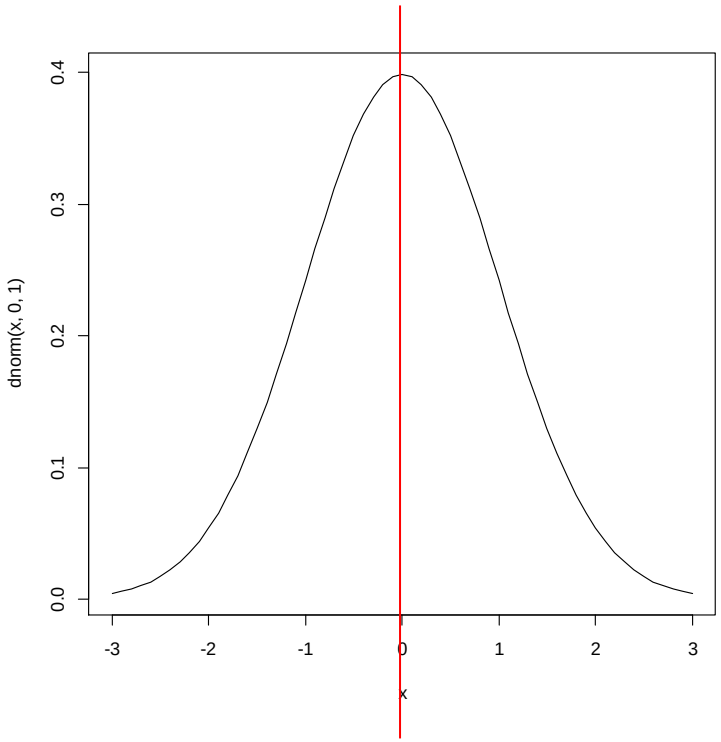
https://erbiostat.wixsite.com/itpb0

Part 1

Location, spread & shape

Location

Density of standard normal, N(0,1), distribution

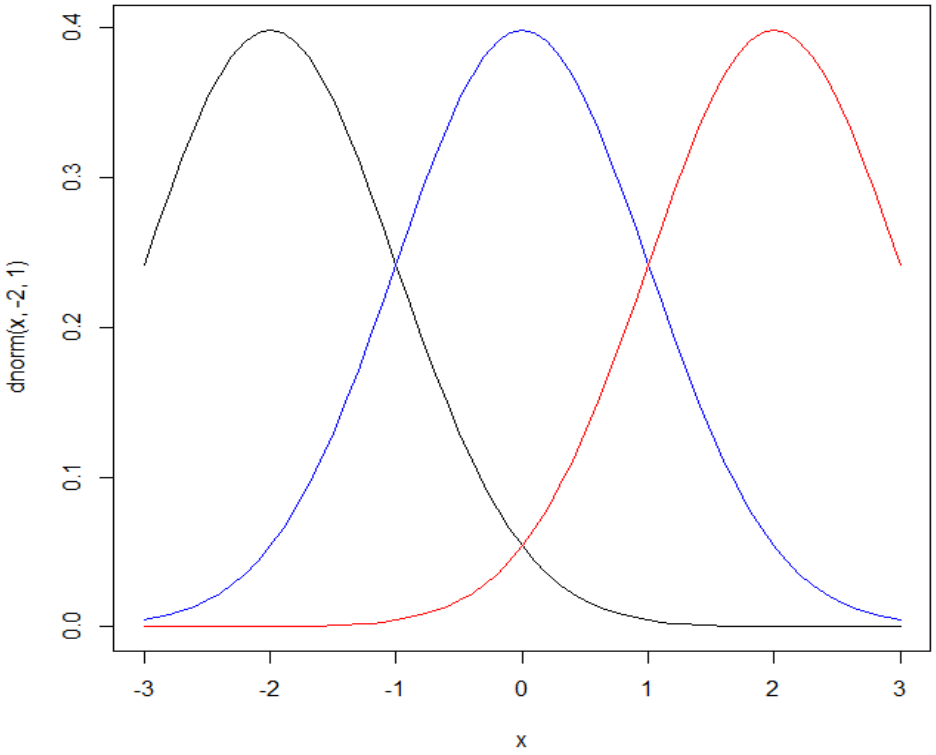


The center of the distribution

- We focus on the center of the distribution where the main part of the data is located.
- How can we visualize and summarize the location of the distribution ?

Densities of N(μ,1)

- Example: three density functions for μ = -2, 0 and 2 (black , blue and red). The distributions are shifted relative to each other and the value of μ determines the shift.



- The three distribution have the same variability but different center.

- R code:

```
x<-seq(-3,3,0.1)
plot(x,dnorm(x, -2, 1),type="l")
lines(x,dnorm(x, 0, 1),col="blue")
lines(x,dnorm(x, 2, 1),col="red")
```

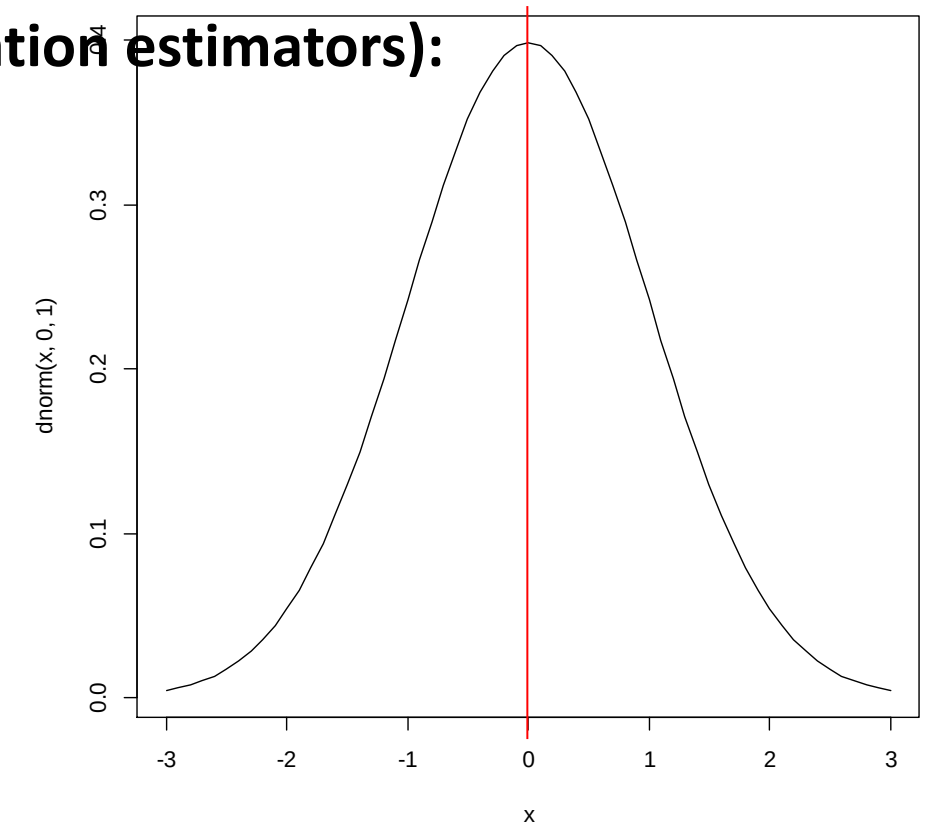
Numerical summaries for location

In real life μ is unknown and need to be estimated from the data.

The estimator for μ is called a location estimator.

Numerical summaries (location estimators):

- Mean.
- Median.
- Trimmed mean.



Numerical summaries for location

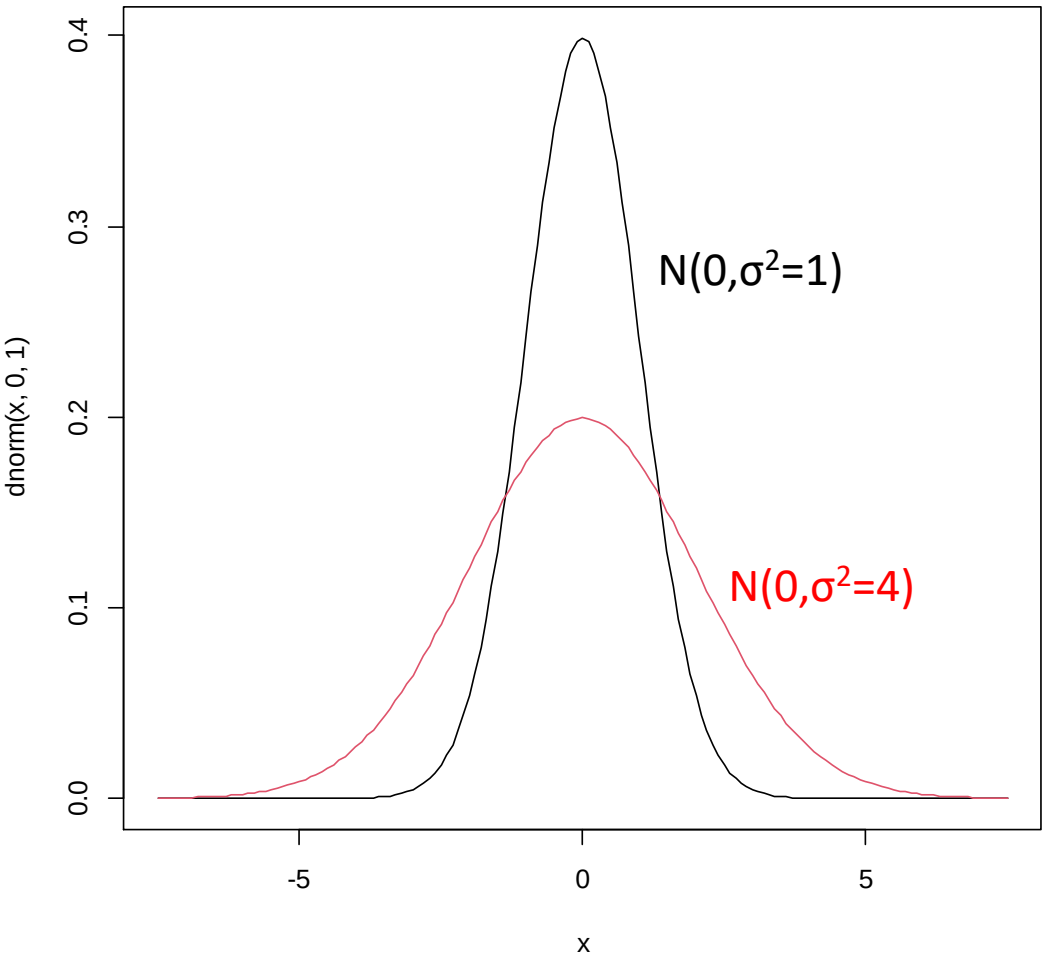
- Most common summary statistics: **sample mean**
- Other estimators: the **median** and the **trimmed mean**

- If the data comes from symmetric distribution the mean gives an estimate for the location of the center of the distribution.

- What if the data comes from non symmetric distribution ?
- How should we choose an estimator among the three?
- What is the difference between the mean, median and trimmed mean ?

Spread

Spread



- Spread of a distribution measures how close the data are to each other.
- How concentrated are the data around the location of the distribution.

- Two densities with the same location but different variability.

Example: spread in two samples

- Consider the following hypothetical samples:
  - Sample 1: -1, 0 , 1
  - Sample 2: -50, 0, 50
- Both samples are symmetric around 0.
- The location estimators for both samples are the same (0).
- The data in the first sample range from -1 to 1, in the second sample the data range from -50 to 50.
- The variability in the second sample is higher.

Variance and forth speard

- Spread Estimators:
- **Standard deviation**
- The most simple measure for spread is the sample variance given by:

$$S_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2$$

- **Fourth-spared**
- A more robust estimator for the spread of the distribution is the fourth-spread (the **interquartile range**) given by

Fourth-spread = upper fourth – lower fourth

Standard deviation and Four-spread

- The fourth-spread is the difference between the 75% and the 25% quantiles of the data.
- It is the range of 50% of the data in the center of the distribution
- It is more robust estimator than the variance since it is not influenced from outliers at the tails as the variance (see later).

- Consider a sample of 5 observations:

$$24, \underbrace{35, 39, 50}_{50-35=15}, 60$$

- The fourth-spread is 15 and the sample variance 192.3.

Standard deviation and Four-spread

- Now, suppose that we change the sample to

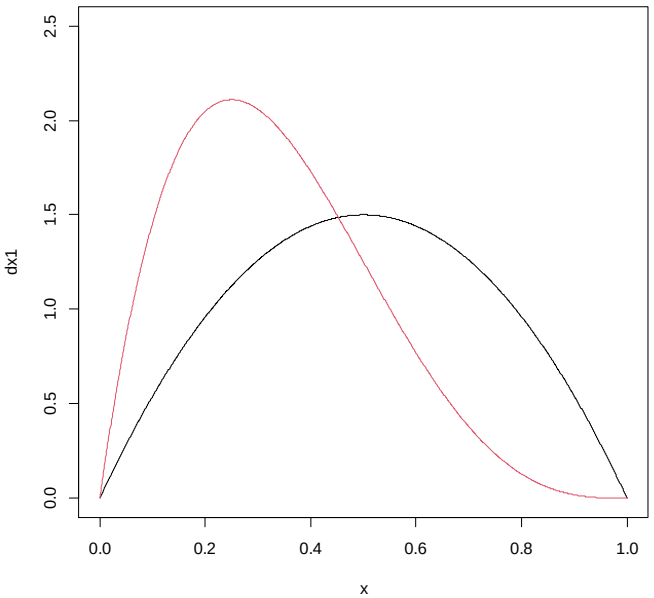
$$24, \underbrace{35, 39, 50}_{50-35=15}, 800$$

- The fourth-spread remains the same
- The sample variance now is equal to 116,520.3.
- The sample variance is sensitive to change, but four-spread is not.

Shape

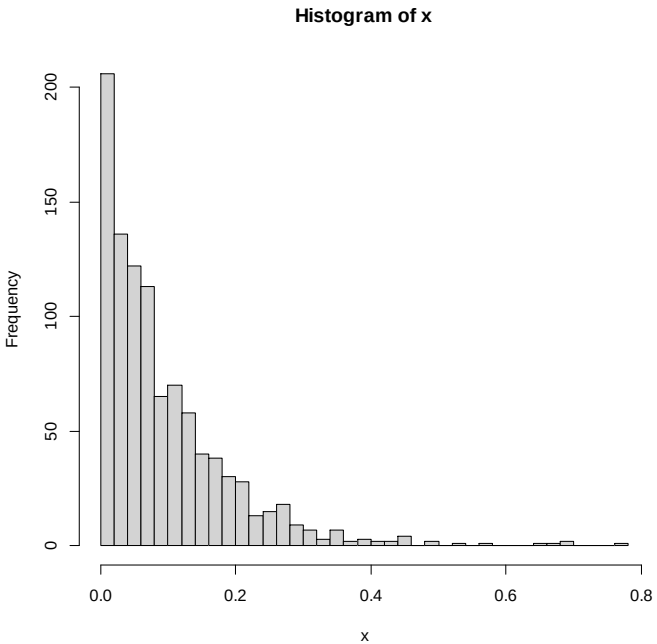
Shape

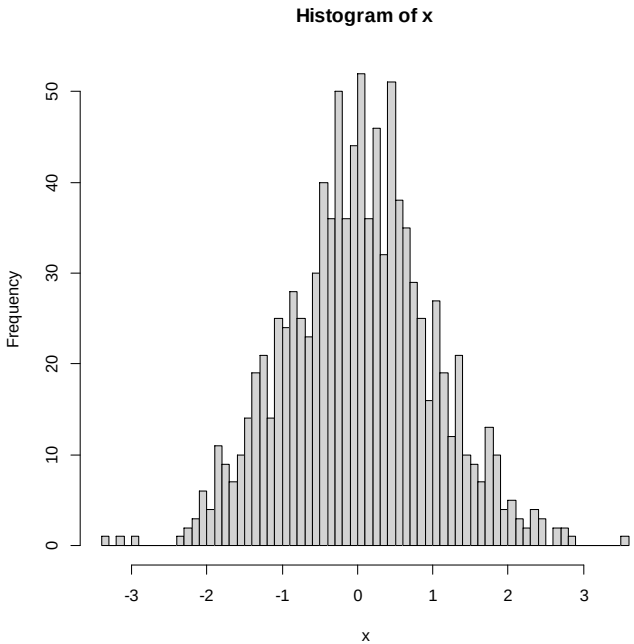
- How does the distribution look like ?
  - Symmetric ?
  - Asymmetric ?
  - Skewed ?
  - One mode or more ?
  - Outliers ?



- Are the data following a normal distribution ?
- How “close” the data are to a normal distribution ?

Random samples





$X \sim exp(\mu = 10)$

Random sample (n=1000).

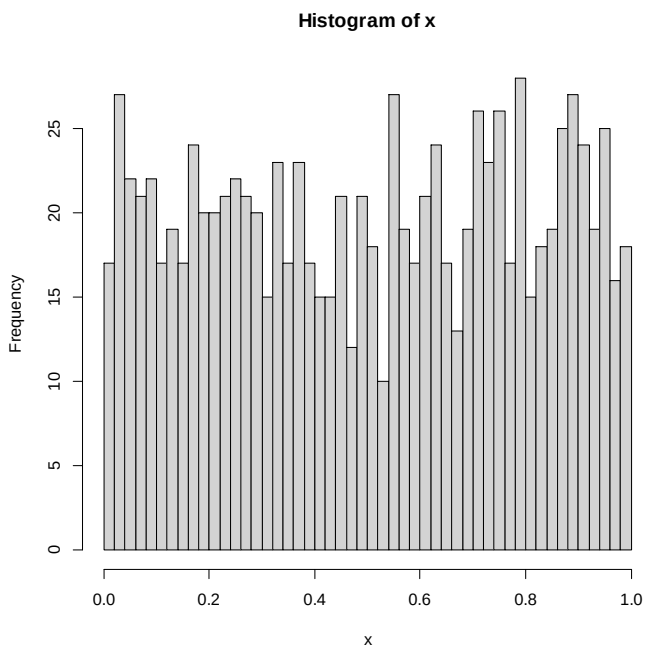
```
x<-rexp(1000,10)
hist(x,nclass=50)
```

$X \sim N(0,1)$

Random sample (n=1000).

```
x<-rnorm(1000,0,1)
hist(x,nclass=50)
```

Random samples



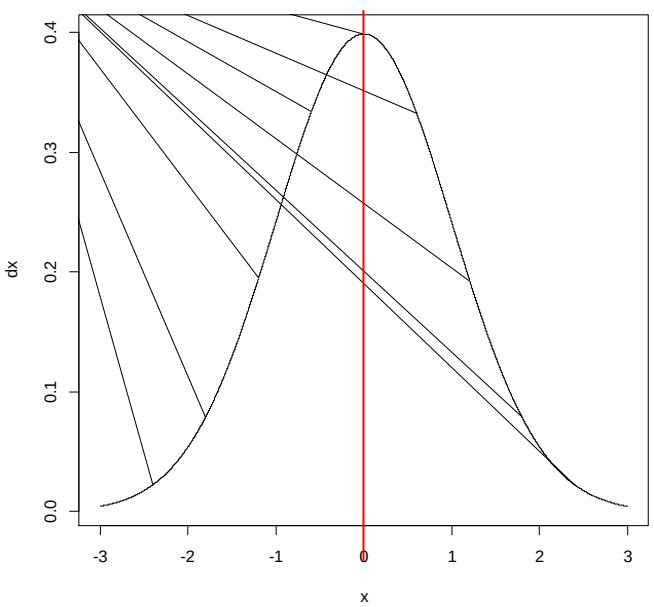
$X \sim U(0,1)$

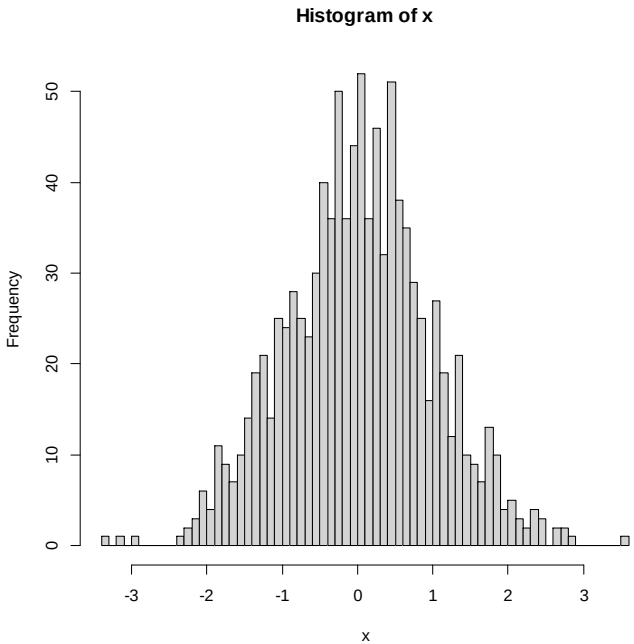
Random sample (n=1000).

```
x<-runif(1000,0,1)
hist(x,nclass=50)
```

- Shape:
  - How does the distribution look like ?
  - Symmetric, skewed ?
  - Uni model, bi model ?

Density and random samples





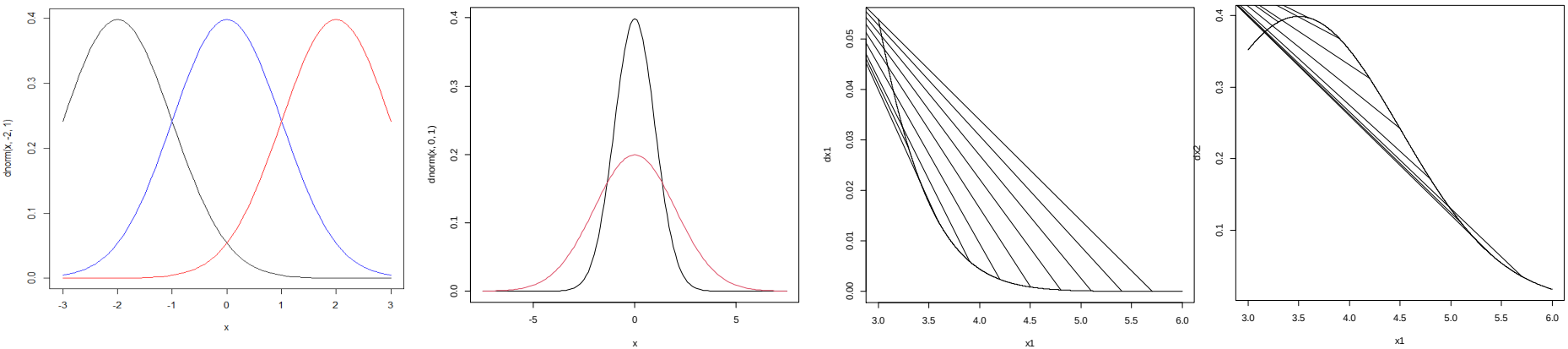

```
x<-seq(from=-3,to=3,length=10000)
dx<-dnorm(x,0,1)
plot(x,dx,type="l")
```

```
x<-rnorm(1000,0,1)
hist(x,nclass=50)
```

The population

A random sample from the population.

Location, spread, shape



- In this course, we focus on descriptive measures, numerical and graphical, to characterize and visualize the features of a particular distribution.
- We focus on:
  - Location.
  - Spread.
  - Shape.
- Each of these control different characteristics of a distribution.

R code for the example

```
x1<-seq(from=3,to=6,length=10000)
dx1<-dnorm(x1,0,1)
dx2<-dnorm(x1,3.5,1)
plot(x1,dx1,type="l")
plot(x1,dx2,type="l")
```

Part 2

The R package `ggplot2`

The R package ggplot2

- `ggplot2` is a plotting R package.
- It provides helpful commands to create complex plots.
- It provides a program interface for specifying:
  - what variables to plot.
  - how they are displayed.
  - general visual properties.

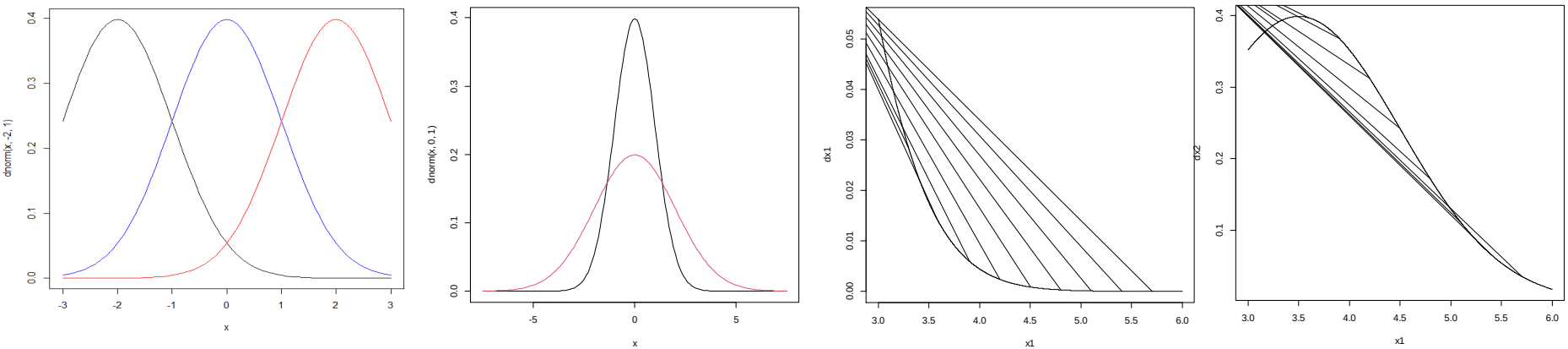
ggplot2 Layers

- `ggplots2` graphics are built layer by layer in order to add new elements to the figure.
- Adding layers in this fashion allows for extensive flexibility and customization of plots.

`ggplot2` Layers

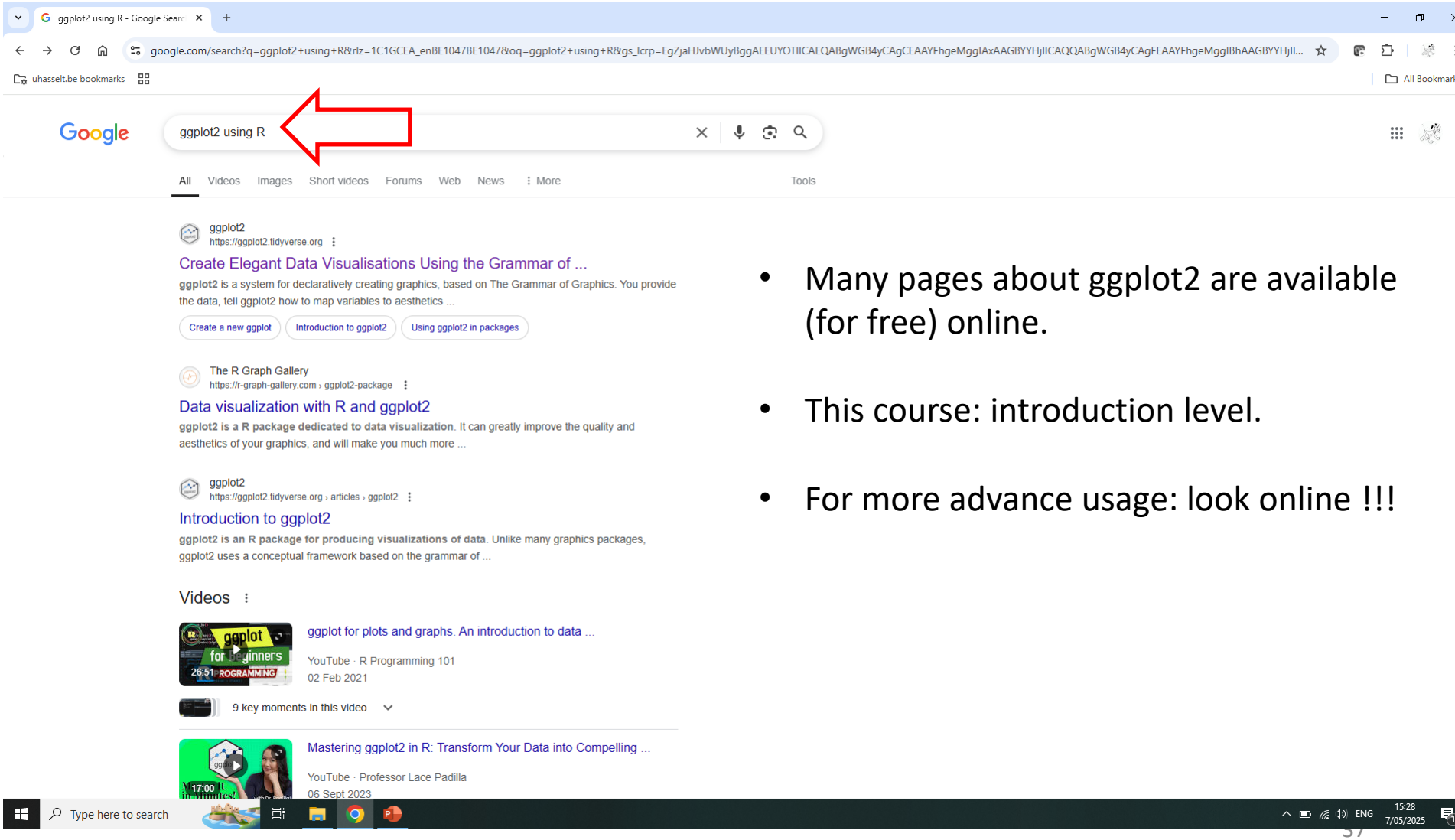
- Layers in `ggplots2` graphics are related to:
  - Data.
  - Variables to be used.
  - Type of plots.
  - Setting of the figure.

Usage of the `ggplot2` package for visualization



- Which graphical tools in **`ggplot2`** are available to characterize and visualize the features of a particular distribution:
    - Location.
    - Spread.
    - Shape.

`ggplot2` online



- Many pages about ggplot2 are available (for free) online.
- This course: introduction level.
- For more advance usage: look online !!!

Part 3

Visualization of one numerical variable in one sample

Rmd program: `Vizualization_intro_Rmd`
HTML file : `eR_biostat_Kampala_VD1_2025.html`

Example 3.1

The `airquality` data

Daily average of wind speed

The HTML file

C:/Ziv_Temp_2023/Workshop_Vietnam_2025/ShortCourse/OnlineBook/Slides/Visualization_intro.html

Visualization_intro.html | Open in Browser | Find | Publish

Code

The wind speed in the airquality dataset

The NHANES dataset

07-05-2025 >eR-BioStat

Introduction to Visualization using `ggplot2` R

Ziv Shkedy et al

Show

The wind speed in the airquality dataset

The `airquality` dataset gives information about 153 daily air quality measurements in New York, May to September 1973.

Show

```
## [1] 153   6
```

Show

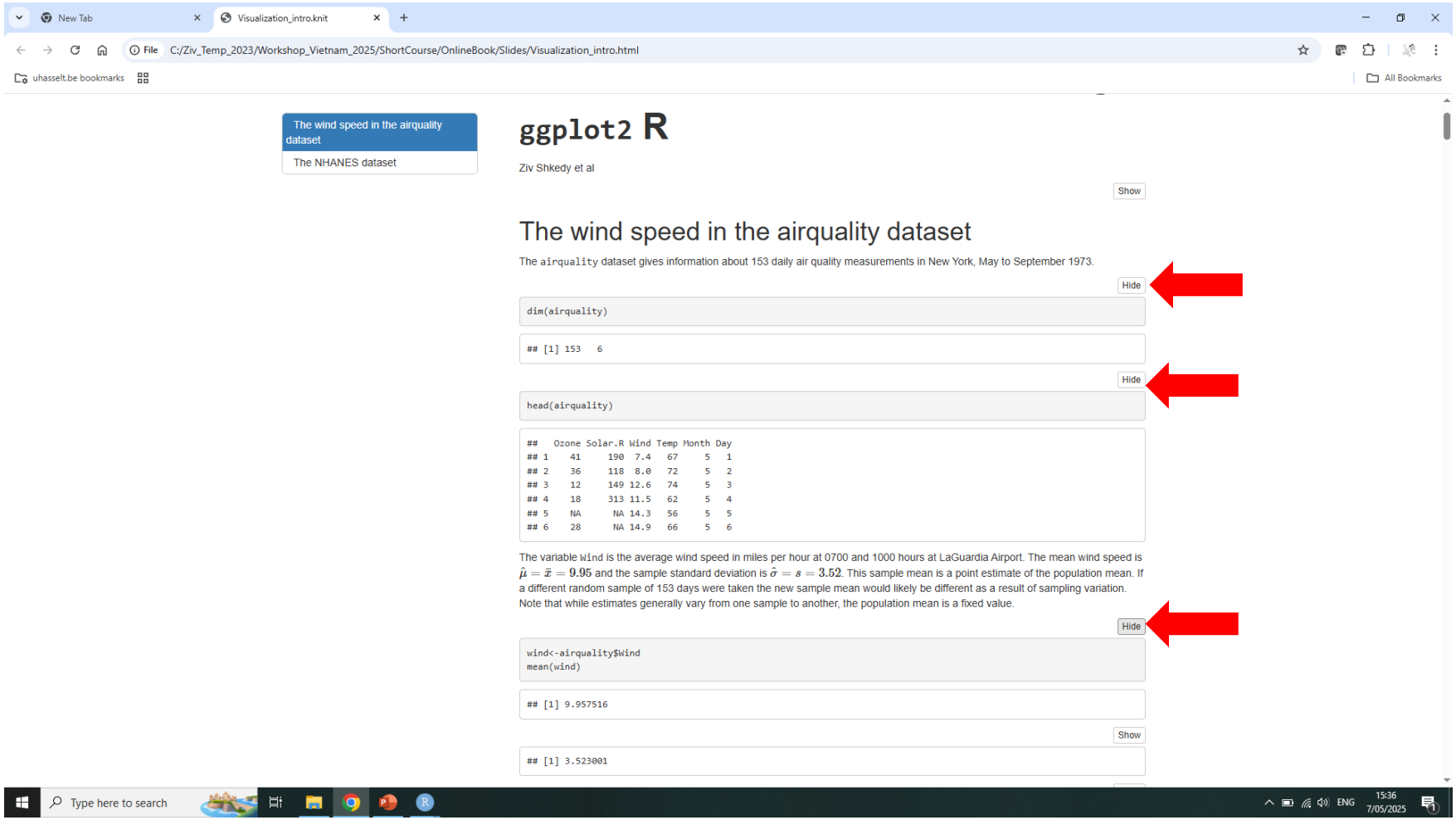
```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
```

Type here to search | ENG | 15:34 7/05/2025

Covers the examples in slides: 43-89

HTML file: **`Kampala_VD1_2025.html`**

The HTML file: R code



The average wind speed per day

- The `airquality` dataset gives information about 153 daily air quality measurements in New York, May to September 1973.
- The variable `Wind` is the average wind speed in miles per hour at 0700 and 1000 hours at LaGuardia Airport.

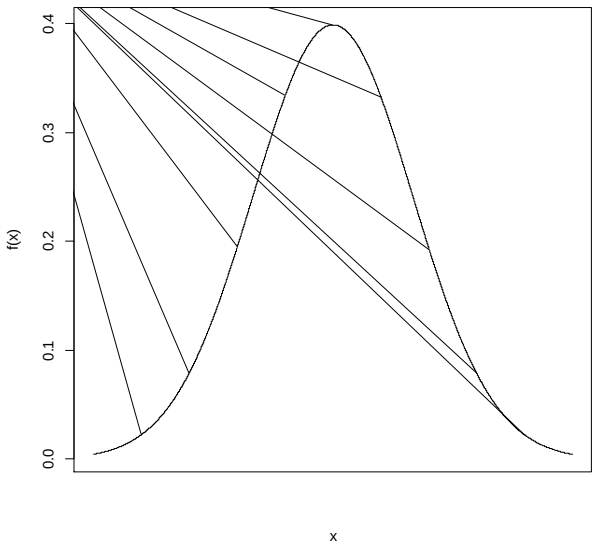
The wind speed in the `airquality` dataset

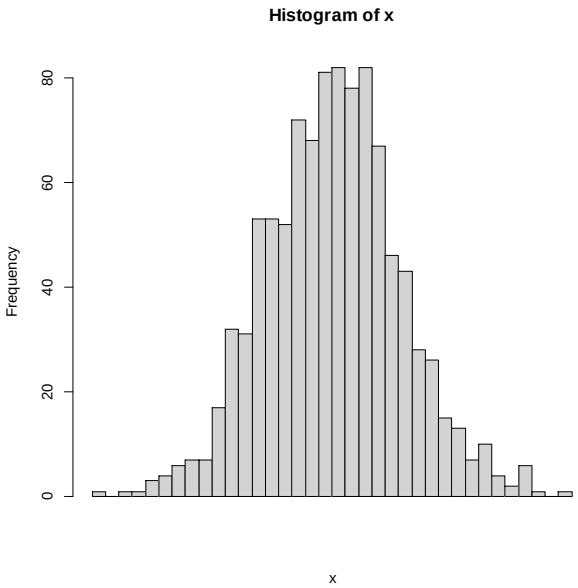
- Daily air quality measurements in New York, May to September 1973.

```
> help("airquality")
> airquality$Wind
  [1]  7.4  8.0 12.6 11.5 14.3 14.9  8.6 13.8 20.1  8.6  6.9  9.7  9.2 10.9 13.2 11.5 12.0 18.4
 [19] 11.5  9.7  9.7 16.6  9.7 12.0 16.6 14.9  8.0 12.0 14.9  5.7  7.4  8.6  9.7 16.1  9.2  8.6
 [37] 14.3  9.7  6.9 13.8 11.5 10.9  9.2  8.0 13.8 11.5 14.9 20.7  9.2 11.5 10.3  6.3  1.7  4.6
 [55]  6.3  8.0  8.0 10.3 11.5 14.9  8.0  4.1  9.2  9.2 10.9  4.6 10.9  5.1  6.3  5.7  7.4  8.6
 [73] 14.3 14.9 14.9 14.3  6.9 10.3  6.3  5.1 11.5  6.9  9.7 11.5  8.6  8.0  8.6 12.0  7.4  7.4
 [91]  7.4  9.2  6.9 13.8  7.4  6.9  7.4  4.6  4.0 10.3  8.0  8.6 11.5 11.5 11.5  9.7 11.5 10.3
[109]  6.3  7.4 10.9 10.3 15.5 14.3 12.6  9.7  3.4  8.0  5.7  9.7  2.3  6.3  6.3  6.9  5.1  2.8
[127]  4.6  7.4 15.5 10.9 10.3 10.9  9.7 14.9 15.5  6.3 10.9 11.5  6.9 13.8 10.3 10.3  8.0 12.6
[145]  9.2 10.3 10.3 16.6  6.9 13.2 14.3  8.0 11.5
```

- 153 observations of the daily average of wind speed.
- A numerical variable.
- EDA:
  - Center of the distribution ?
  - How the distribution look like?

What do we want to visualize ?





How dose the density (in the population) of the wind speed look like ?

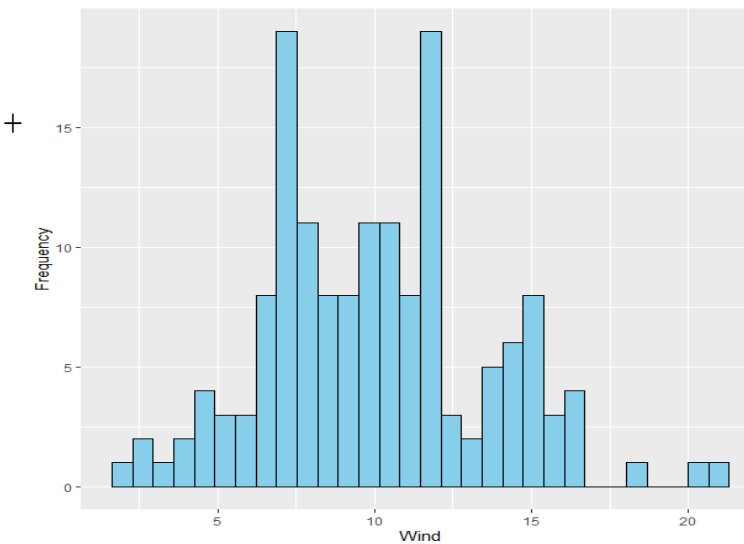
The sample that we have.

R code for the example

```
x1<-seq(from=-3,to=3,length=10000)
dx1<-dnorm(x1,0,1)
plot(x1,dx1,type="l",xaxt="n",yxat="n",xlab="x",ylab="f(x)")
x<-rnorm(1000,0,1)
hist(x,nclass=50,xaxt="n",yxat="n")
```

Histogram of wind speed using `ggplot2`

```
ggplot(airquality, aes(x = Wind)) +
geom_histogram(fill = "skyblue", color = "black")+
ylab("Frequency")
```



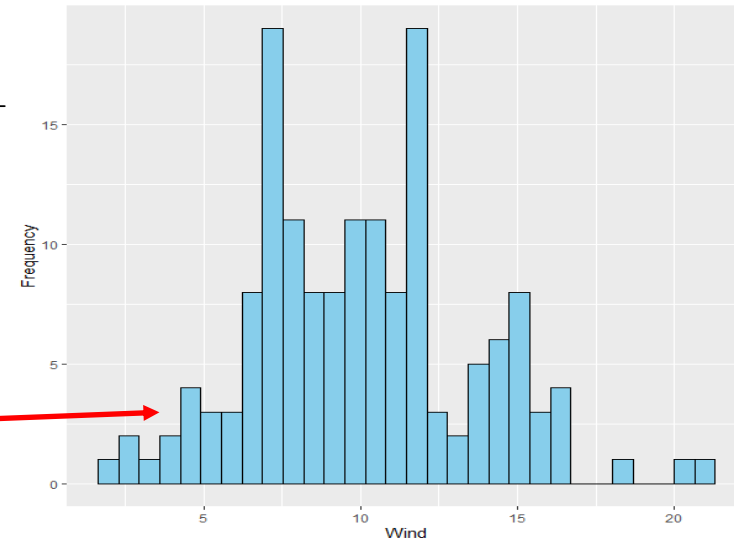
Layer 1: data and variable to be used

```
ggplot(airquality, aes(x = Wind))
```

- We define an **aes**thetic mapping (using the **`aes()`** function:
  - Select the variable(s) to be plotted.
  - Specify how to present them in the graph, e.g., as x/y positions.

Histogram of wind speed

```
ggplot(airquality, aes(x = Wind)) +
geom_histogram(fill = "skyblue", color = "black")+
ylab("Frequency")
```



Layer 2: the plot type to be used

```
geom_histogram(fill = "skyblue", color = "black")
```

- **`geom_histogram()`:** plot a histogram of the data.
  - Selecting the color of the bars: `fill=…`.
  - Selecting the color of the lines separate the bars: `color=…`

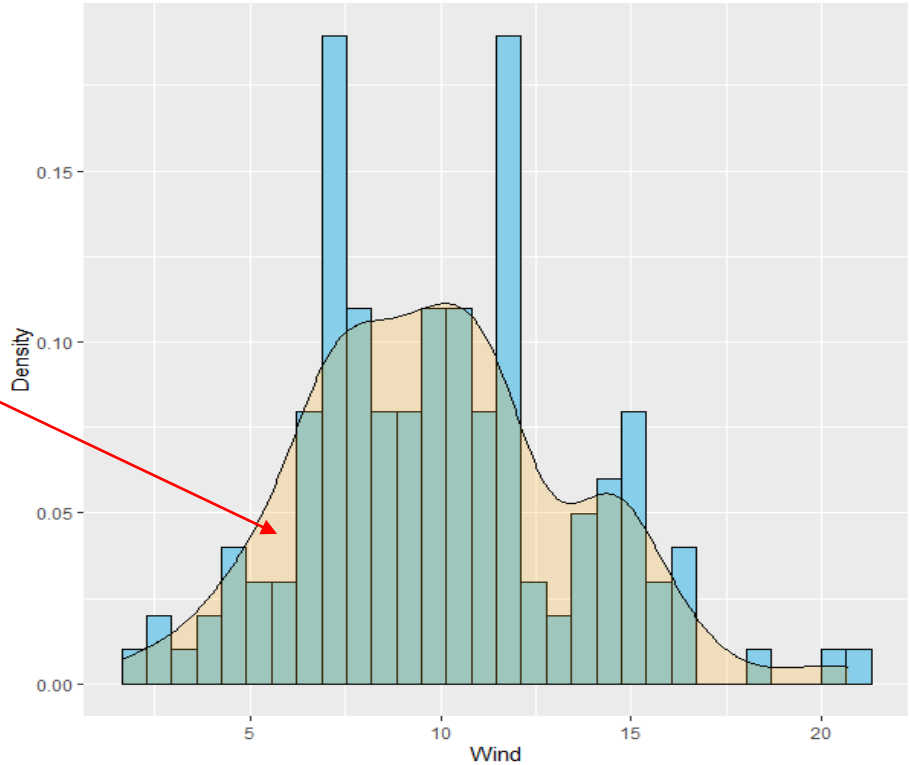
Histogram with density plot of wind speed

```
ggplot(airquality, aes(x = Wind)) +
geom_histogram(aes(y = ..density..), fill = "skyblue", color = "black") +
geom_density(alpha = 0.2, fill = "orange")+  ylab("Density")
```

Layer 3: adding the density plot:

```
geom_density(alpha = 0.2,
             fill = "orange")
```

- The color of the density plot: `fill=…`
- The opacity of the density plot: `alpha=…`



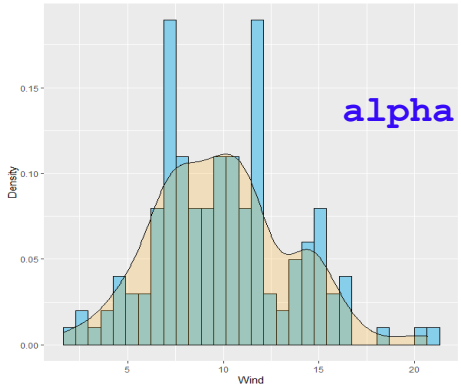
Histogram with density plot of wind speed

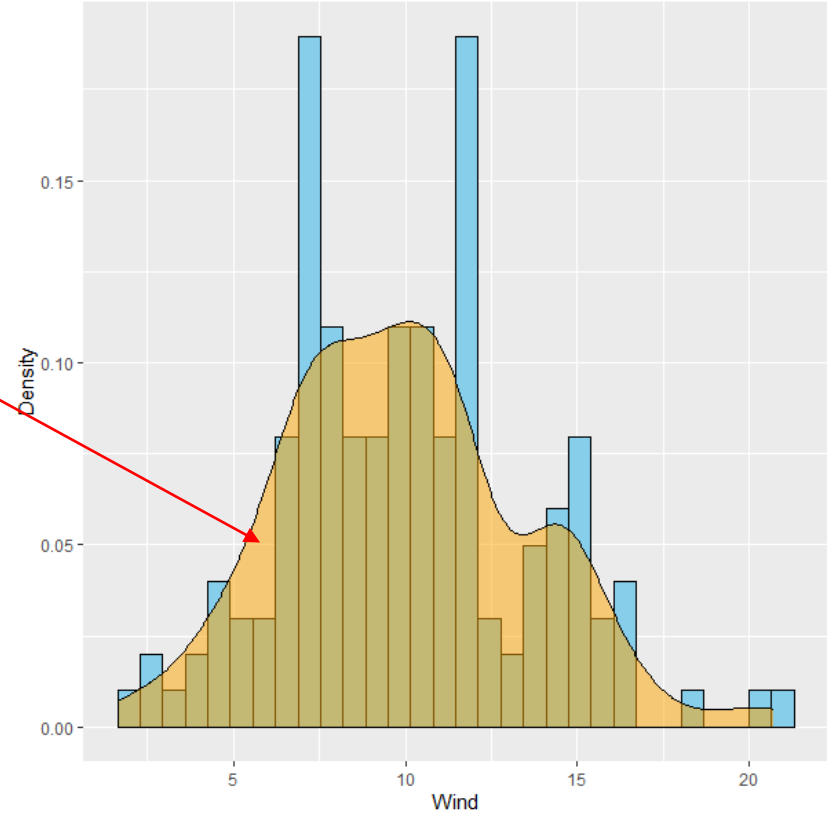
```
ggplot(airquality, aes(x = Wind)) +
geom_histogram(aes(y = ..density..), fill = "skyblue", color = "black") +
geom_density(alpha = 0.5, fill = "orange")+  ylab("Density")
```

Layer 3: adding the density plot:

```
geom_density(alpha = 0.5,
             fill = "orange")
```

- Changing the value of alpha:





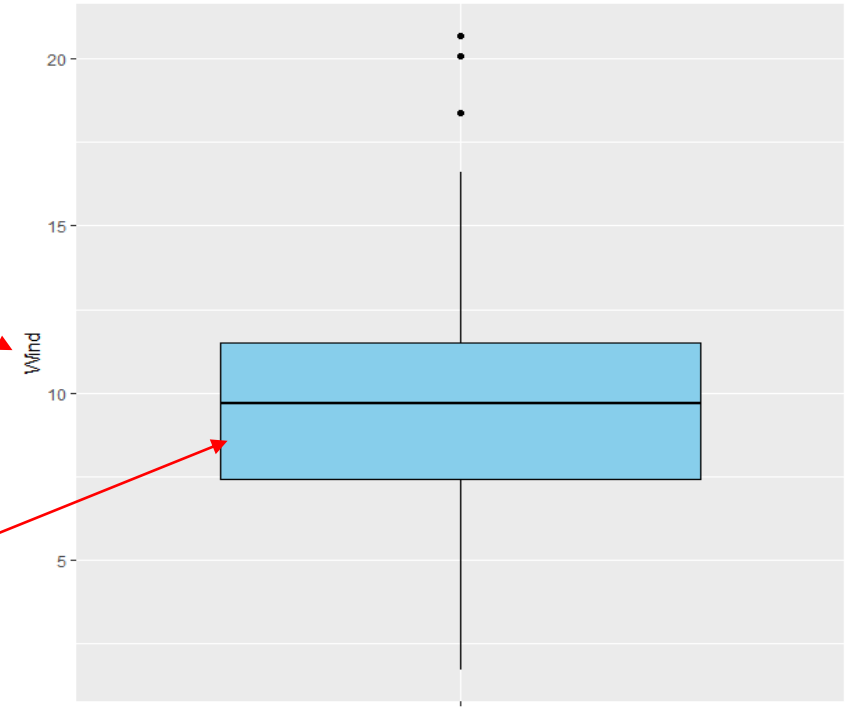
Boxplot of wind speed

```
ggplot(airquality, aes(x = "", y = Wind)) +
geom_boxplot(fill = "skyblue", color = "black")+  xlab("")
```

Layer 1: data and variable to be used:

```
ggplot(airquality, aes(x = "", y = Wind))
```

The variable Wind is plotted on the Y-axis.



Layer 2: type of the plot and setting:

```
geom_boxplot(fill = "skyblue", color = "black")
```

`geom_boxplot`: plot a boxplot.

The colors of the lines.

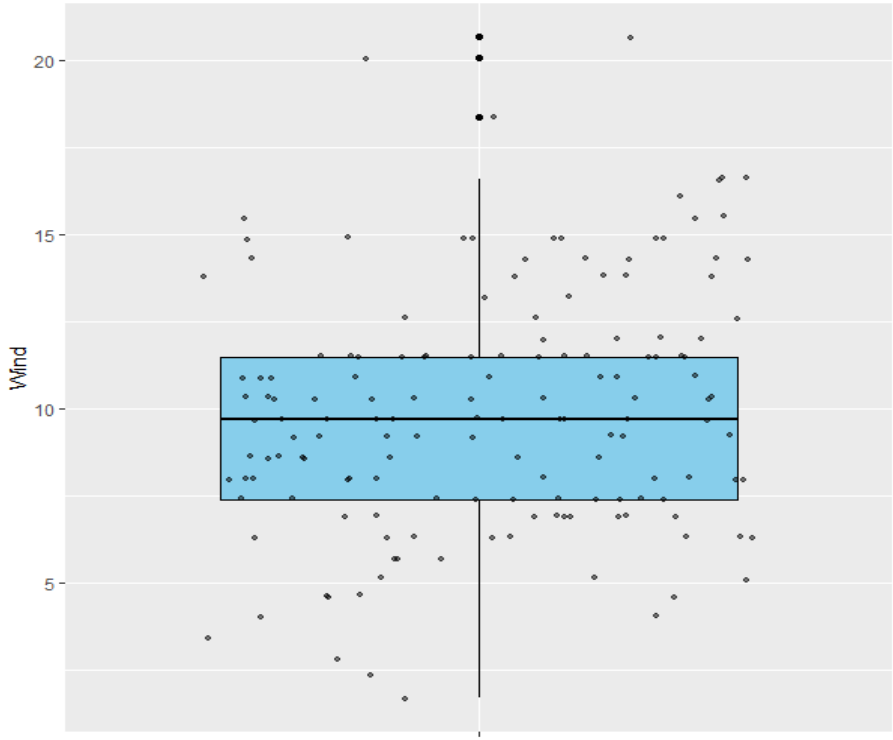
Boxplot of wind speed with data points

```
ggplot(airquality, aes(x = "", y = Wind)) +
geom_boxplot(fill = "skyblue", color = "black") +
geom_jitter(aes(x = "", y = Wind), color = "black", size = 1, alpha = 0.5) +
xlab("")
```

Layer 3: add the data to the boxplot:

```
geom_jitter(aes(x = "", y = Wind),
color = "black", size = 1, alpha = 0.5)
```

- `geom_jitter()`: add the data points to the boxplot.
- `alpha=0.5`: control the spread of the data.



Boxplot of wind speed with data points

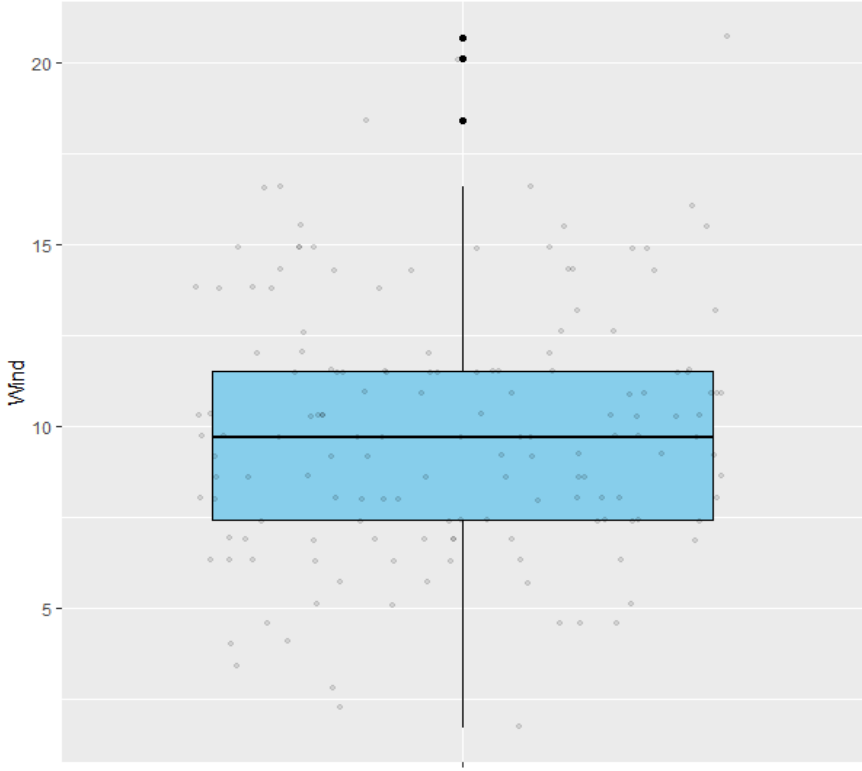
```
ggplot(airquality, aes(x = "", y = Wind)) +
geom_boxplot(fill = "skyblue", color = "black") +
geom_jitter(aes(x = "", y = Wind), color = "black", size = 1, alpha = 0.1) +
xlab("")
```

Layer 3: add the data to the boxplot:

```
geom_jitter(aes(x = "", y = Wind),
color = "black", size = 1, alpha = 0.1
```

- `alpha=0.5` VS. `alpha=0.1`

See next slide



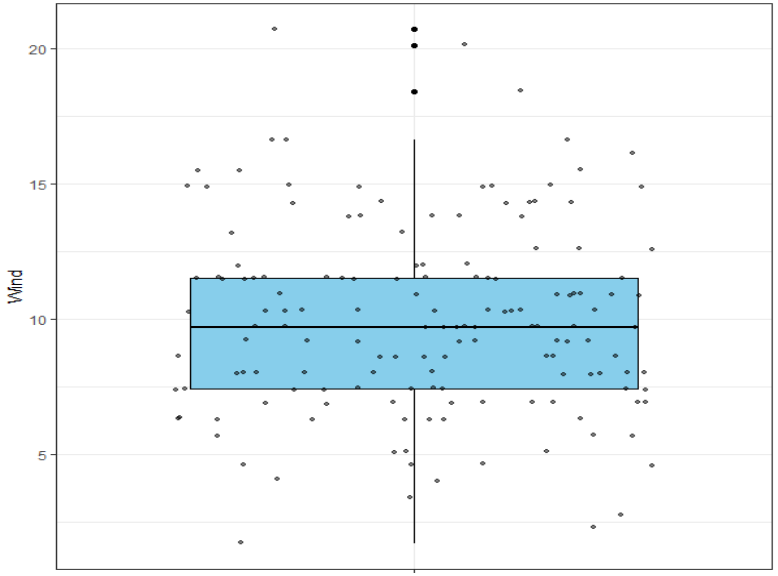
Boxplot of wind speed with data points

```
ggplot(airquality, aes(x = "", y = Wind)) +
geom_boxplot(fill = "skyblue", color = "black") +
geom_jitter(aes(x = "", y = Wind), color = "black", size = 1, alpha = 0.5) +
xlab("") + theme_bw()
```

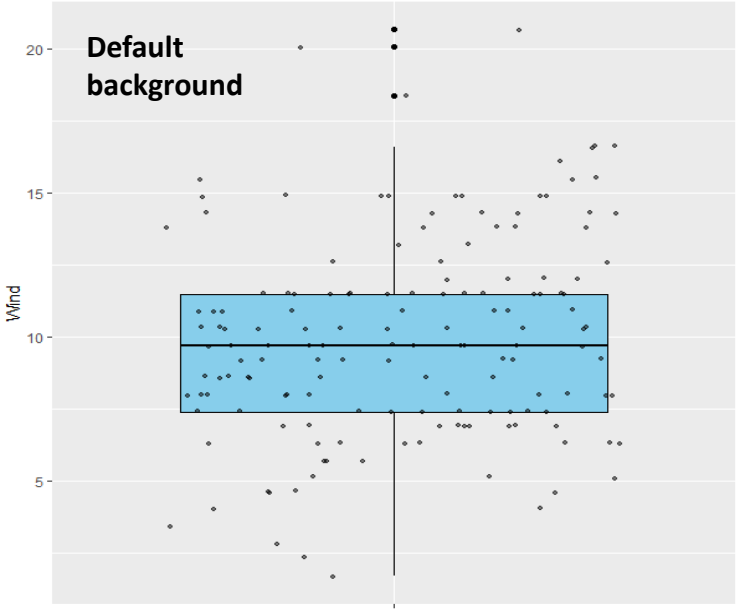
Layer 4: change the backgroup color:

```
theme_bw()
```

bw="black & white"



Point size with alpha=0.5



Violin plot of wind speed

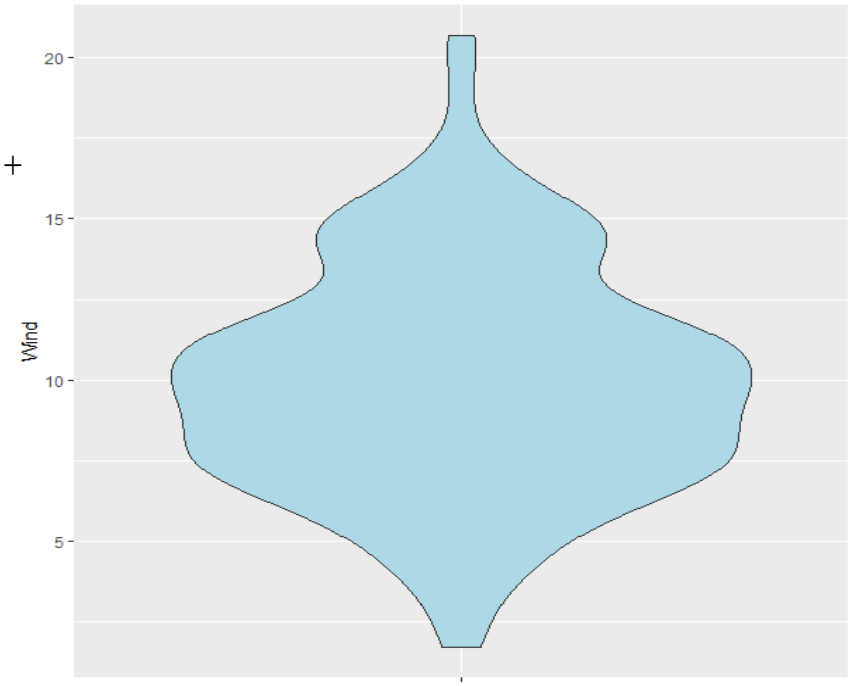
```
ggplot(airquality, aes(x = "", y = Wind)) +
geom_violin(fill = "lightblue") +  xlab("")
```

Layer 1: data and variable to be used:

```
ggplot(airquality, aes(x = "", y = Wind)) +
```

Layer 2: make a violin plot:

```
geom_violin(fill = "lightblue")+  xlab("")
```

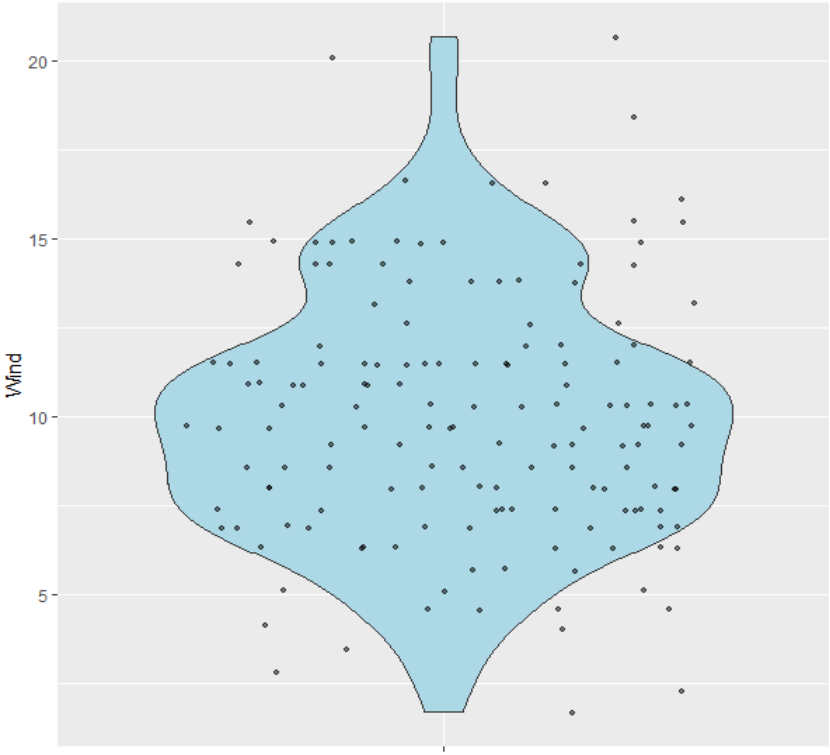


Violin plot of wind speed with data points

```
ggplot(airquality, aes(x = "", y = Wind)) +
geom_violin(fill = "lightblue") +
geom_jitter(aes(x = "", y = Wind), color = "black", size = 1, alpha = 0.5) +
xlab("")
```

Layer 3: add the data to the plot:

```
geom_jitter(aes(x = "", y = Wind),
color = "black",
size = 1, alpha = 0.5)
```



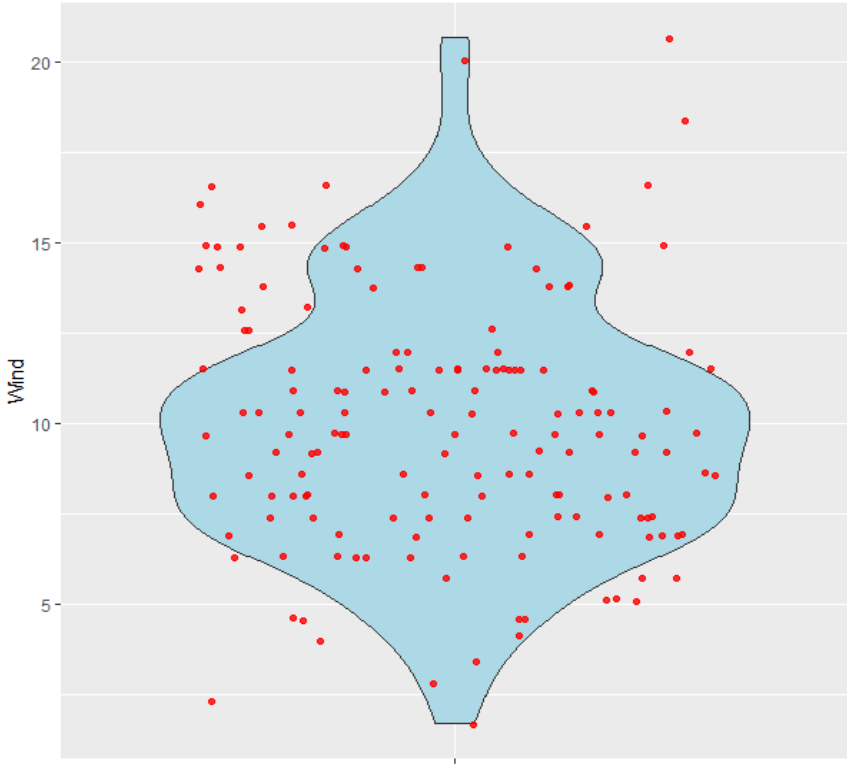
Violin plot of wind speed with data points

```
ggplot(airquality, aes(x = "", y = Wind)) +
geom_violin(fill = "lightblue") +
geom_jitter(aes(x = "", y = Wind), color = "red", size = 1.5, alpha = 0.8) +
xlab("")
```

Layer 3: add the data to the plot:

```
geom_jitter(aes(x = "", y = Wind),
color = "red",
size = 1.5, alpha = 0.8)
```

- `Color`: color of the points.
- `Size`: size of the points.
- `Alpha`: the opacity of the points.



Example 3.2

The NHANES dataset

BMI

The `NHANES` dataset

- The `NHANES` dataset consists of data from the US National Health and Nutrition Examination Study.
- Information about 76 variables is available for 10000 subjects included in the study.

- Three variables:
  - BMI.
  - Number of sleep hours per night.
  - Total cholesterol level.

The BMI variable

The variable BMI measures the body mass index.

```
> NHANES$BMI
   [1] 32.22 32.22 32.22 15.30 30.57 16.82 20.64 27.24 27.24 27.24 23.67 23.69
  [13] 26.03 19.20 26.22 26.60 27.40 28.54 25.84 24.74 19.73 19.73 20.66 36.32
  [25] 36.32 35.84 24.32 25.95 31.43 31.43 27.18 21.00 25.79 25.79 29.13 30.60
  [37] 30.60 23.34 22.85 22.85 26.46 26.46 26.46 26.46 25.45 21.16 46.69 20.15
  [49] 27.06 37.33 37.33 15.59 15.59 25.54 24.98 22.63 14.35 37.92 37.92 37.92
  [61]    NA 18.16 25.52 28.96 28.96 32.49 32.49 32.49 18.35 16.24 16.24 28.48
  [73] 28.48 19.41 36.28 25.87 25.87 25.87 28.60 21.03 21.03 21.03 30.90 30.90
  [85] 30.90 30.90 31.51 31.51 27.74 27.25 27.25 24.53 29.83 22.81 29.27 17.87
   .........
```

- 10000 observations.
- Numerical variable with missing values (NA).

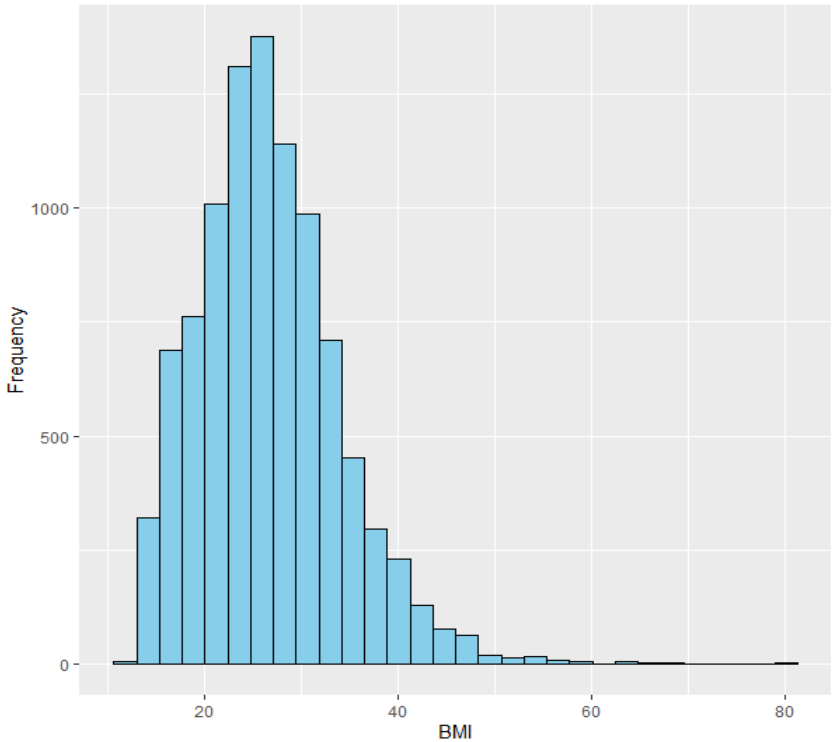
Histogram of BMI

```
ggplot(NHANES, aes(x = BMI)) +
geom_histogram(fill = "skyblue", color = "black") +
ylab("Frequency")
```

Layer 1: data and variable to be used:

```
ggplot(NHANES, aes(x = BMI)) +
```

- We define an **aes**thetic mapping (using the aes() function:
  - Selecting the variable(s) to be plotted.
  - Specifying how to present them in the graph, e.g., as x/y positions.



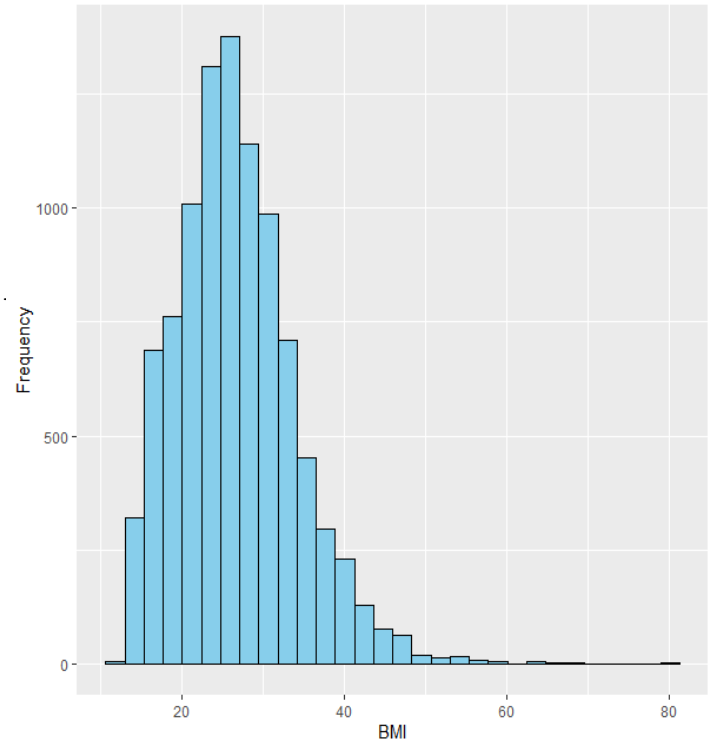
Histogram with density plot of BMI

```
ggplot(NHANES, aes(x = BMI)) +
geom_histogram(fill = "skyblue", color = "black") +
ylab("Frequency")
```

Layer 2: the plot type to be used:

```
geom_histogram(fill = "skyblue", color = "black")
```

- `geom_histogram()`: plot a histogram of the data.
  - Selecting the color of the bars: `fill=…`.
  - Selecting the color of the lines separate the bars: `colors=…`



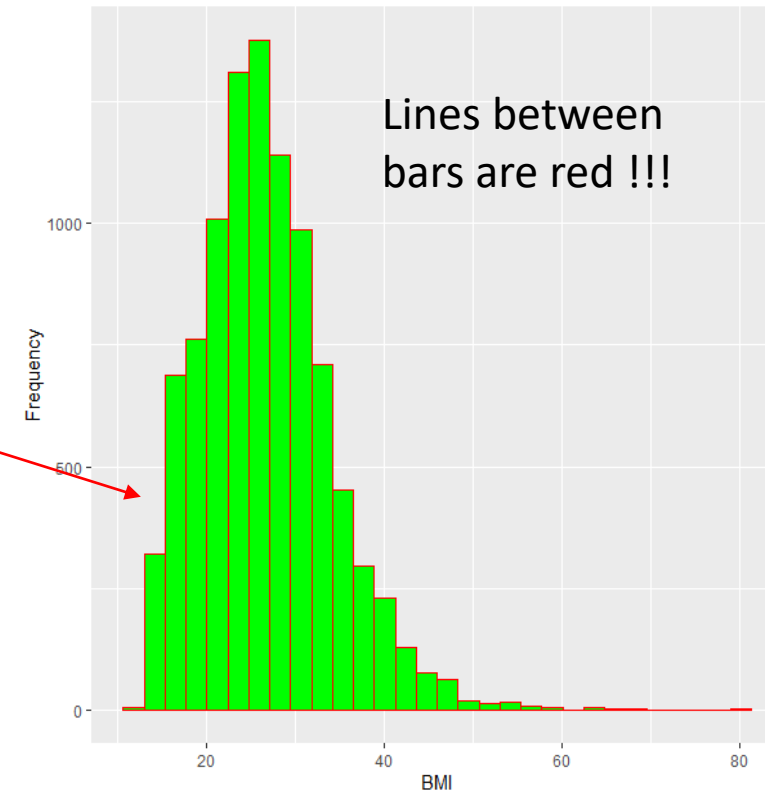
Histogram with density plot of BMI

```
ggplot(NHANES, aes(x = BMI)) +
geom_histogram(fill = "green", color = "red") +  ylab("Frequency")
```

Layer 2: the plot type to be used:

```
geom_histogram(fill = "green", color = "red")
```

- fill=….
- colors=…



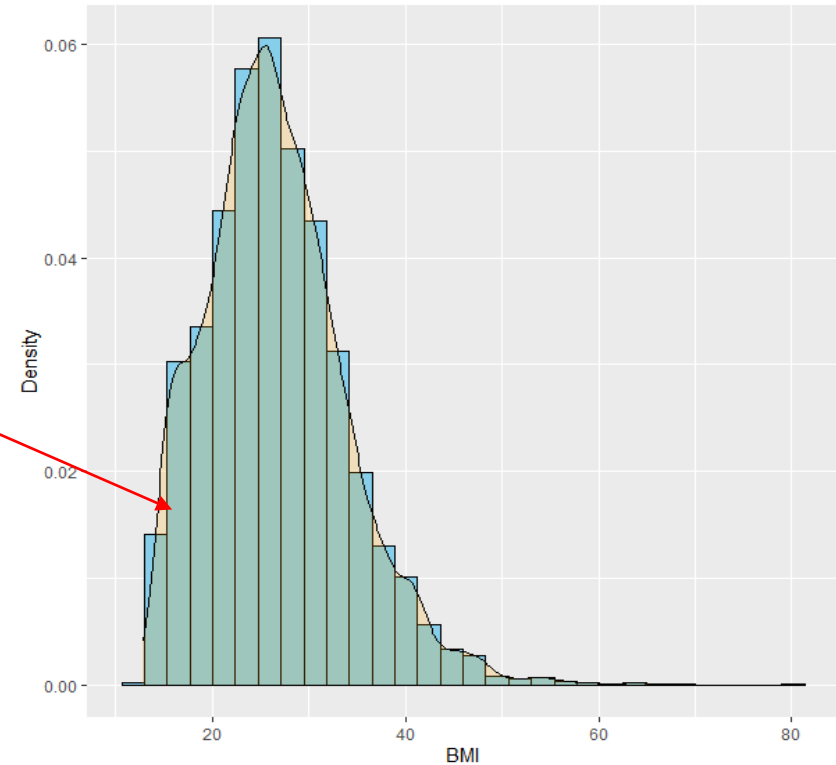
Histogram with density plot of BMI

```
ggplot(NHANES, aes(x = BMI)) +
geom_histogram(aes(y = ..density..), fill = "skyblue", color = "black") +
geom_density(alpha = 0.2, fill = "orange") +  ylab("Density")
```

Layer 3: adding the density plot:

```
geom_density(alpha = 0.2,
             fill = "orange")
```

- The color of the density plot: `fill=…`
- The opacity of the density plot: `alpha=…`



Boxplot of BMI

```
ggplot(NHANES, aes(x = "", y = BMI)) +
geom_boxplot(fill = "skyblue", color = "black") +  xlab("")
```

Layer 1: data and variable to be used:

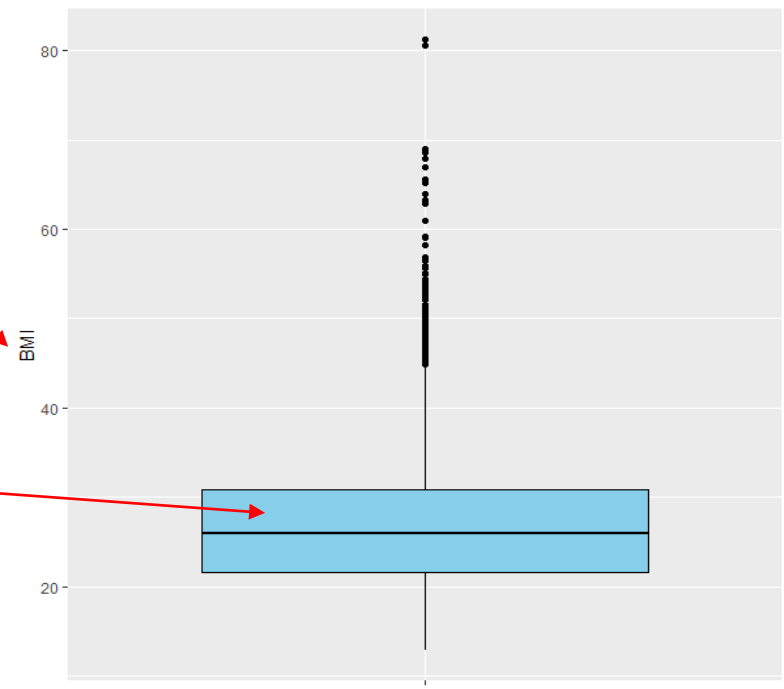
```
ggplot(NHANES, aes(x = "", y = BMI))
```

The variable BMI is plotted on the Y-axis.

Layer 2: type of the plot and setting:

```
geom_boxplot(fill = "skyblue",
             color = "black")
```

The color of the lines.



`geom_boxplot`: plot a boxplot.

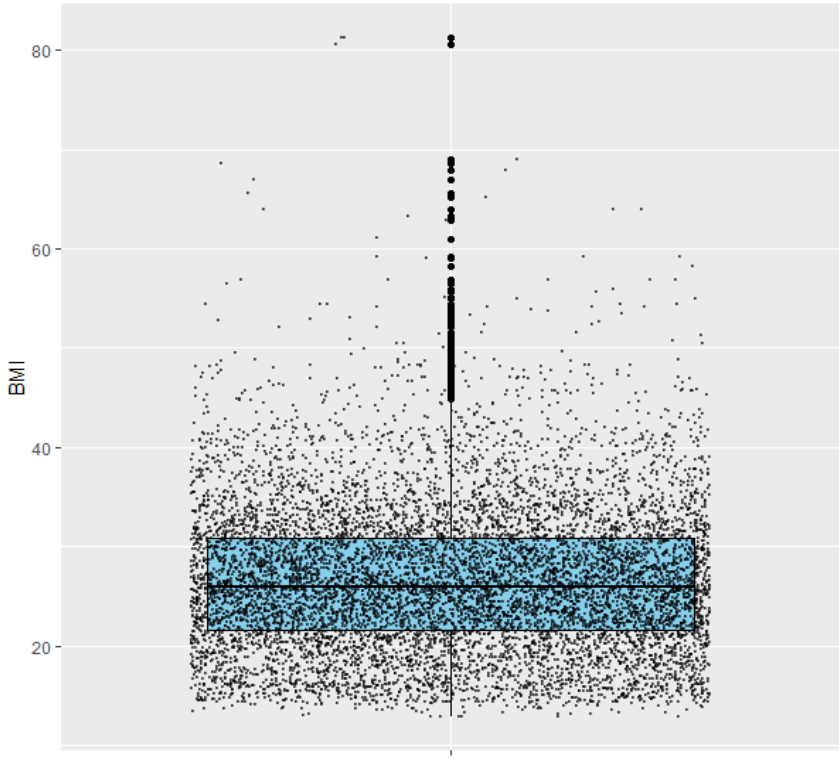
Boxplot of BMI with data points

```
ggplot(NHANES, aes(x = "", y = BMI)) +
geom_boxplot(fill = "skyblue", color = "black") +
geom_jitter(aes(x = "", y = BMI), color = "black",
size = 0.1, alpha = 0.5) +  xlab("")
```

Layer 3: add the data to the boxplot:

```
geom_jitter(aes(x = "", y = BMI), color
= "black", size = 0.1, alpha = 0.5)
```

- `geom_jitter()`: add the data points to the boxplot.
- `alpha=0.5`: control the spread of the data.



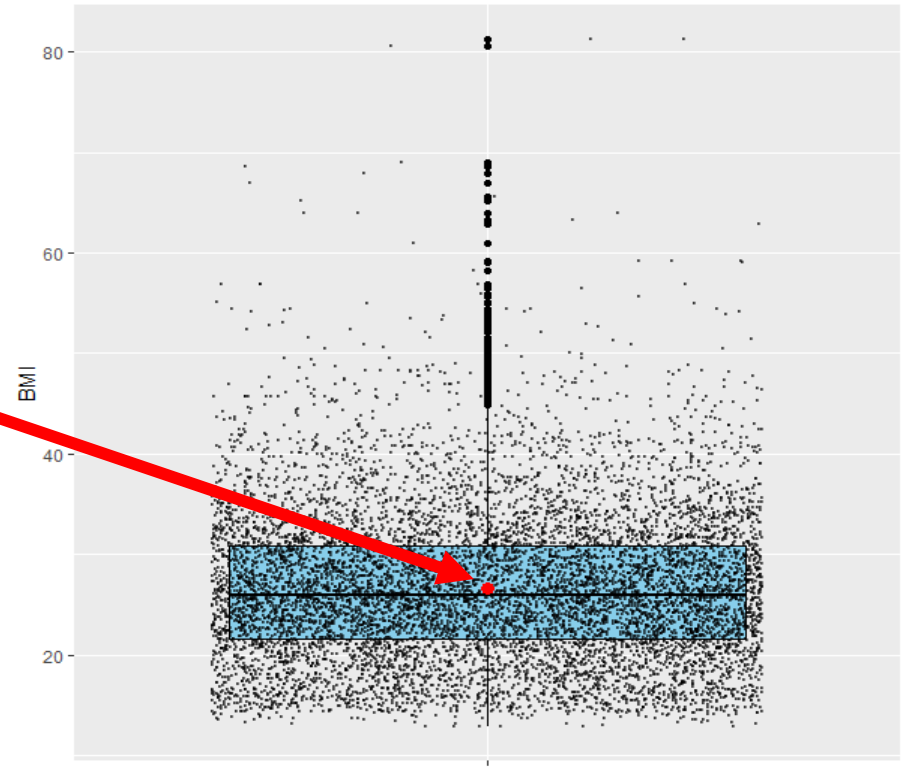
Boxplot of BMI with data points

```
ggplot(NHANES, aes(x = "", y = BMI)) +
geom_boxplot(fill = "skyblue", color = "black") +
geom_jitter(aes(x = "", y = BMI), color = "black", size = 0.1, alpha = 0.5) +
stat_summary(fun = mean, size = 0.5, color = "red") + xlab("")
```

Layer 4: add the mean

```
stat_summary(fun = mean,
size = 0.5, color = "red")
```

The function `stat_summary()` calculate summary stats of the data.

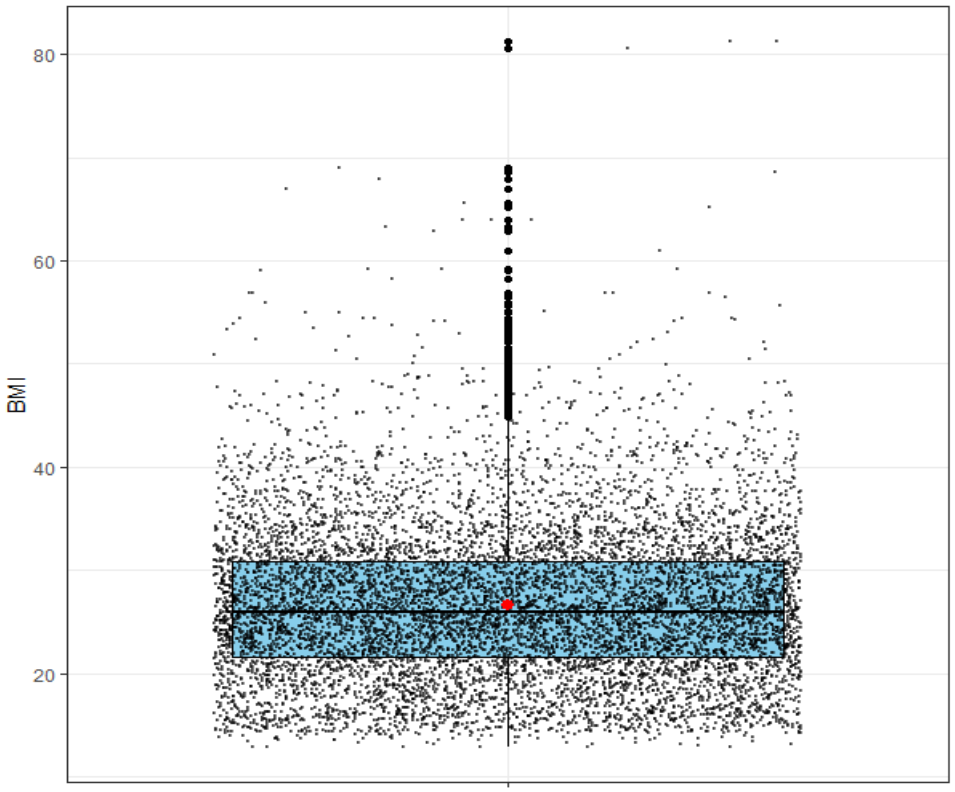


Boxplot of BMI with data points

```
ggplot(NHANES, aes(x = "", y = BMI)) +
geom_boxplot(fill = "skyblue", color = "black") +
geom_jitter(aes(x = "", y = BMI), color = "black", size = 0.1, alpha = 0.5) +
stat_summary(fun = mean, size = 0.5, color = "red") + xlab("") +
theme_bw()
```

Layer 5: Changing background to black and white.

```
theme_bw()
```



Violin plot of BMI

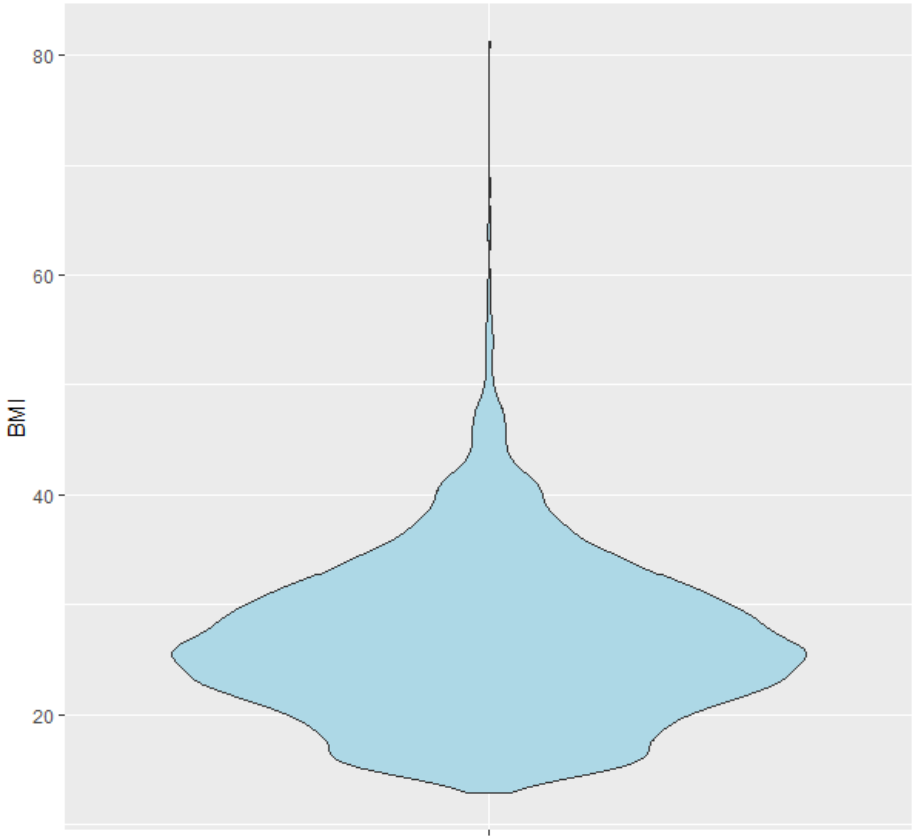
```
ggplot(NHANES, aes(x = "", y = BMI)) +
geom_violin(fill = "lightblue") +  xlab("")
```

Layer 1: data and variable to be used:

```
ggplot(NHANES, aes(x = "", y = BMI))
```

Layer 2: make a violin plot:

```
geom_violin(fill = "lightblue")
```



Violin plot of BMI with mean and SD

```
ggplot(NHANES, aes(x = "", y = BMI)) +
geom_violin(fill = "lightblue") +
stat_summary(fun = mean,  size = 0.5, color = "red") +
geom_errorbar(aes(ymin = NHANES_summary$mean_BMI - NHANES_summary$sd_BMI,
ymax = NHANES_summary$mean_BMI + NHANES_summary$sd_BMI), width = 0.2,
color = "blue") +  xlab("") +  ylab("BMI")
```

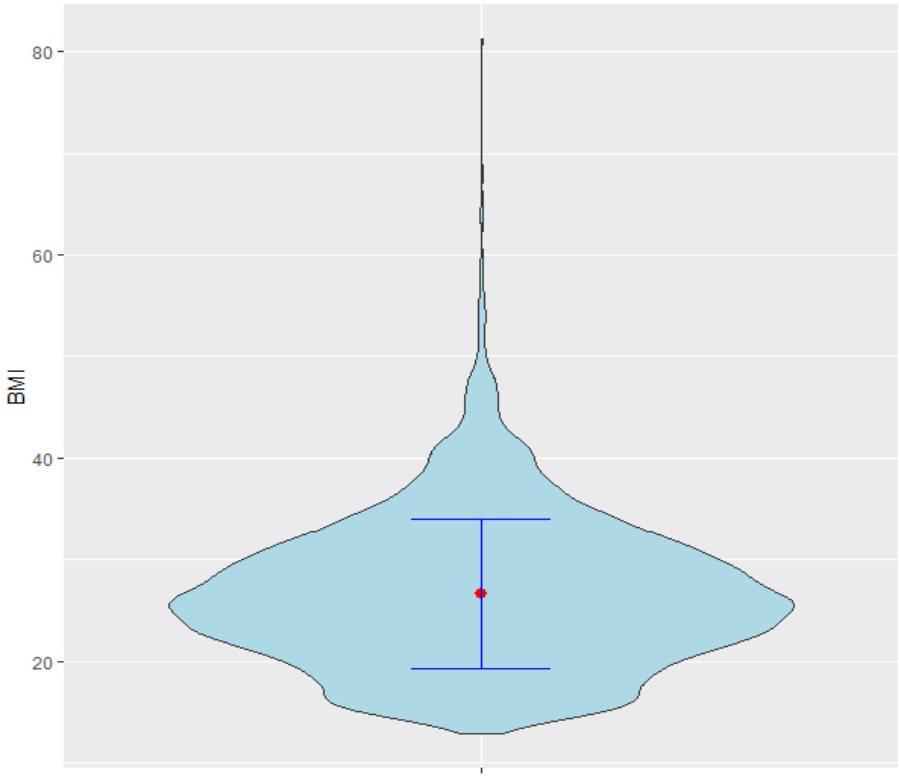
Layer 3: add the mean and SD to the plot:

- Calculate the mean and SD:

```
NHANES_summary <- NHANES %>%
summarize(mean_BMI = mean(BMI,
na.rm = TRUE), sd_BMI = sd(BMI,
na.rm = TRUE))
```

- Add to the plot:

```
stat_summary(fun = mean,  size = 0.5,
color = "red") +
geom_errorbar(aes(ymin =
NHANES_summary$mean_BMI -
NHANES_summary$sd_BMI,
ymax = NHANES_summary$mean_BMI +
NHANES_summary$sd_BMI),
width = 0.2, color = "blue")
```



Example 3.3

The NHANES dataset

The number of sleep hour per night

The number of sleep hours per night variable

The variable SleepHrsNight measures the number of sleep hours per night.

```
> NHANES$SleepHrsNight
   [1]  4  4  4 NA  8 NA NA  8  8  8  7  5  4 NA  5  7 NA  6  6  6  7  7  8  6  6  5
  [27] NA  6  4  4  5  7  5  5  6  7  7  7 NA NA  8  8  8  8  6  6  6  6  8  4  4 NA
  [53] NA  6  8  9 NA  6  6  6 NA NA  6  7  7  9  9  9 NA NA NA  8  8  8  8  6  6  6
  [79]  6  6  6  6  8  8  8  8  6  6 NA  8  8 NA  7  7  5  7  8 NA NA NA  8  6  6  6
 [105]  6  6  8  8  8 NA  6  8  8  6  8  8  7  7  7  7  7 NA  6  6  7  7  8  7 10  7
 [131]  6  6  6  6  6  6  5 NA  6  6  4  5  7  7  6  6  7  7  6  7  7 12 NA NA  6  6
 [157]  6  6  8  8 NA  7  7  6  7 NA  7  6  6  8  6  8  8 NA  4  4  6 NA  8  8  6  5
  ……
```

- 10000 observations.
- Numerical variable with missing values (NA).
- How the distribution look like?

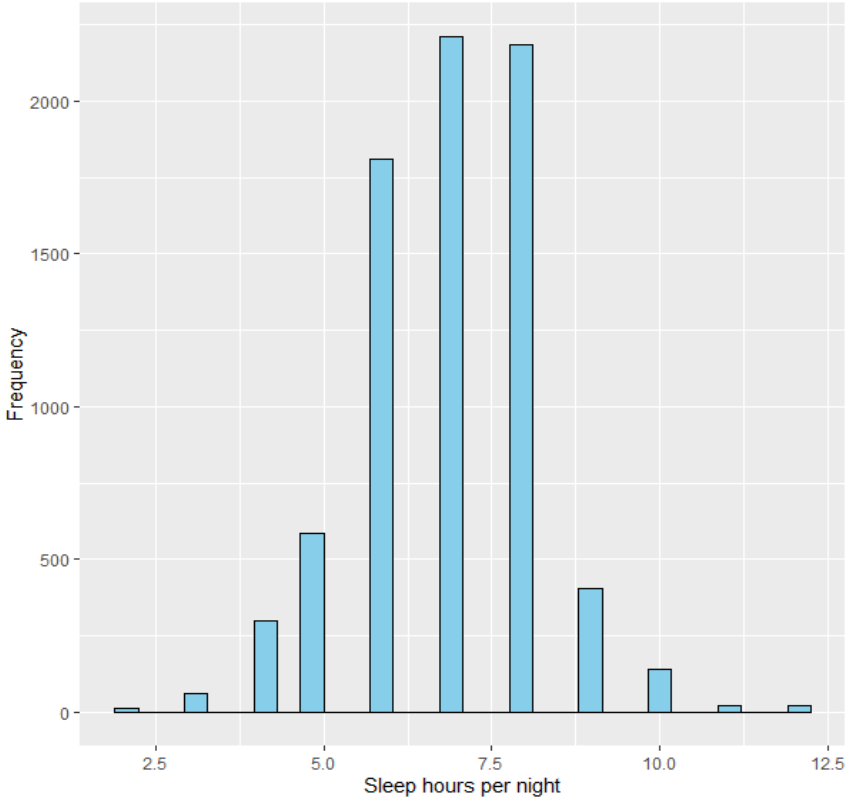
Histogram of number of sleep hours per night

```
ggplot(NHANES, aes(x = SleepHrsNight)) +
geom_histogram(fill = "skyblue", color = "black") +
ylab("Frequency") +  xlab("Sleep hours per night")
```

Layer 1: data and variable to be used:

```
ggplot(NHANES, aes(x = SleepHrsNight))
```

- We define an **aes**thetic mapping (using the aes() function:
  - Selecting the variable(s) to be plotted.
  - Specifying how to present them in the graph, e.g., as x/y positions.



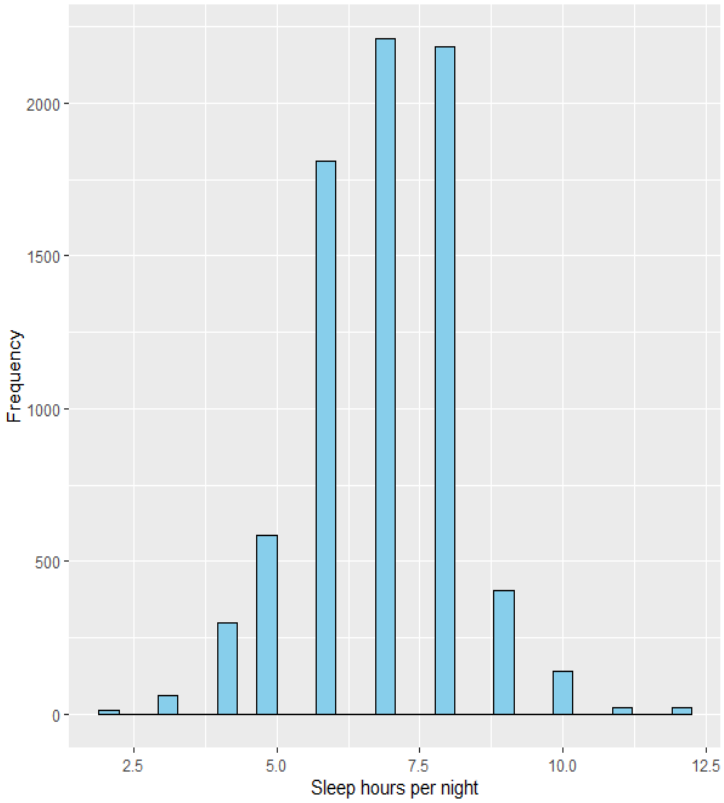
Histogram of number of sleep hours per night

```
ggplot(NHANES, aes(x = SleepHrsNight)) +
geom_histogram(fill = "skyblue", color = "black") +
ylab("Frequency") +  xlab("Sleep hours per night")
```

Layer 2: the plot type to be used:

```
geom_histogram(fill = "skyblue", color = "black")+
```

- `geom_histogram()`: plot a histogram of the data.
  - Selecting the color of the bars: `fill=…`.
  - Selecting the color of the lines separate the bars: `color=…`



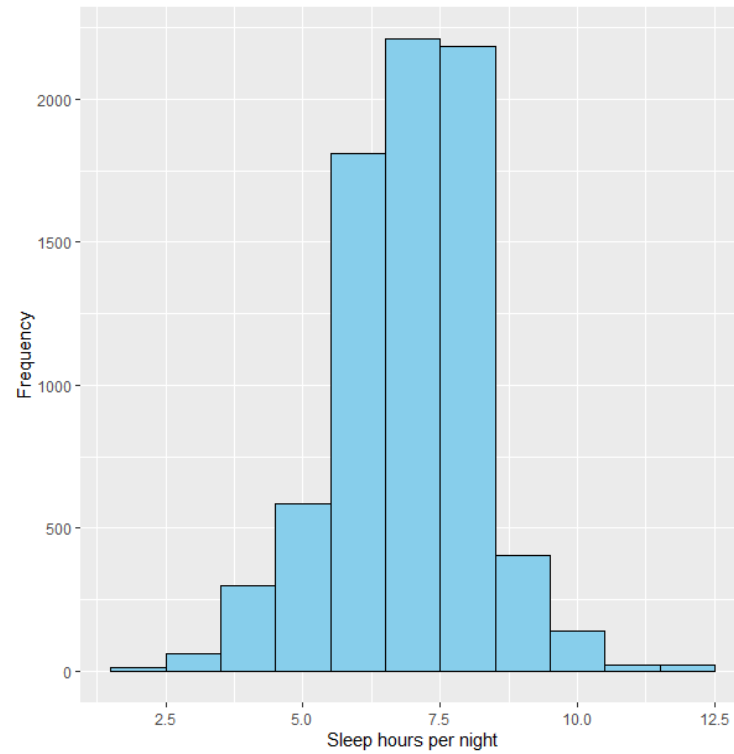
Histogram of number of sleep hours per night

```
ggplot(NHANES, aes(x = SleepHrsNight)) +
geom_histogram(fill = "skyblue", color = "black", binwidth = 1) +
ylab("Frequency") +  xlab("Sleep hours per night")
```

Layer 2: Adjust the width of the bars:

```
geom_histogram(fill = "skyblue", color = "black",
               binwidth = 1)
```

- Adjusting the width of the bars: `binwidth =…`



Histogram of number of sleep hours per night

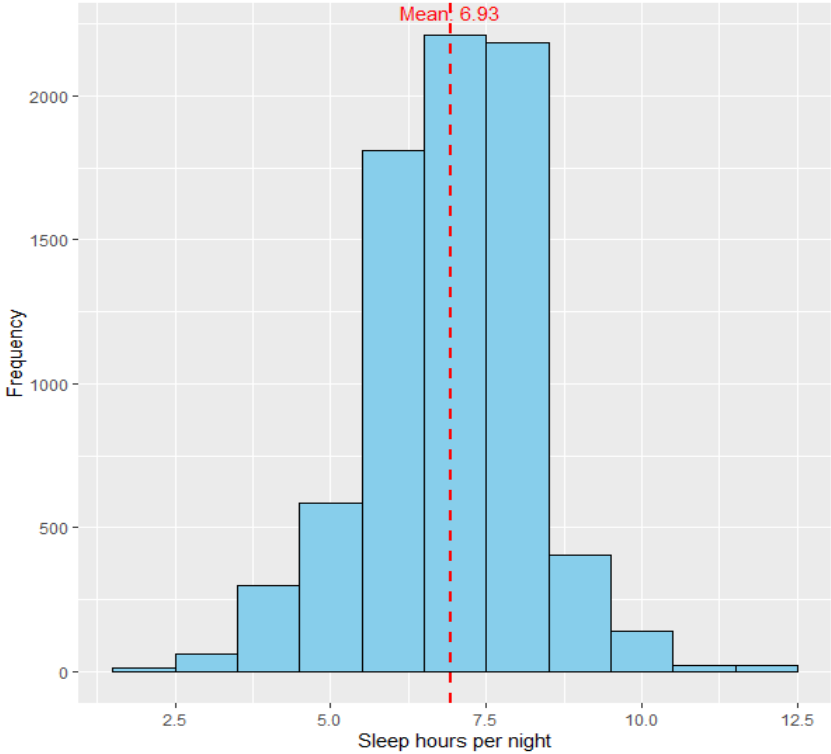
```
ggplot(NHANES, aes(x = SleepHrsNight)) +
geom_histogram(fill = "skyblue", color = "black") +  ylab("Frequency") +
xlab("Sleep hours per night") +
geom_vline(aes(xintercept = mean_sleep), color = "red", linetype =
"dashed", size = 1) +
annotate("text", x = mean_sleep, y = max(table(NHANES$SleepHrsNight)),
label = paste("Mean:", round(mean_sleep, 2)), color = "red", vjust = -1)
```

Layer 3: Calculate the mean sleep hours per night

```
mean_sleep <- NHANES %>%
summarize(mean_SleepHrsNight =
mean(SleepHrsNight, na.rm = TRUE))
%>%  pull(mean_SleepHrsNight)
```

Layer 3: add the mean line, and mean text annotation

```
geom_vline(aes(xintercept = mean_sleep),
color = "red", linetype = "dashed", size
= 1) +
annotate("text", x = mean_sleep, y =
max(table(NHANES$SleepHrsNight)),
label = paste("Mean:", round(mean_sleep,
2)), color = "red", vjust = -1)
```



Boxplot of number of sleep hours per night

```
ggplot(NHANES, aes(x = "", y = SleepHrsNight)) +
geom_boxplot(fill = "skyblue", color = "black")+
ylab("Sleep hours per night")+  xlab("")
```

Layer 1: data and variable to be used:

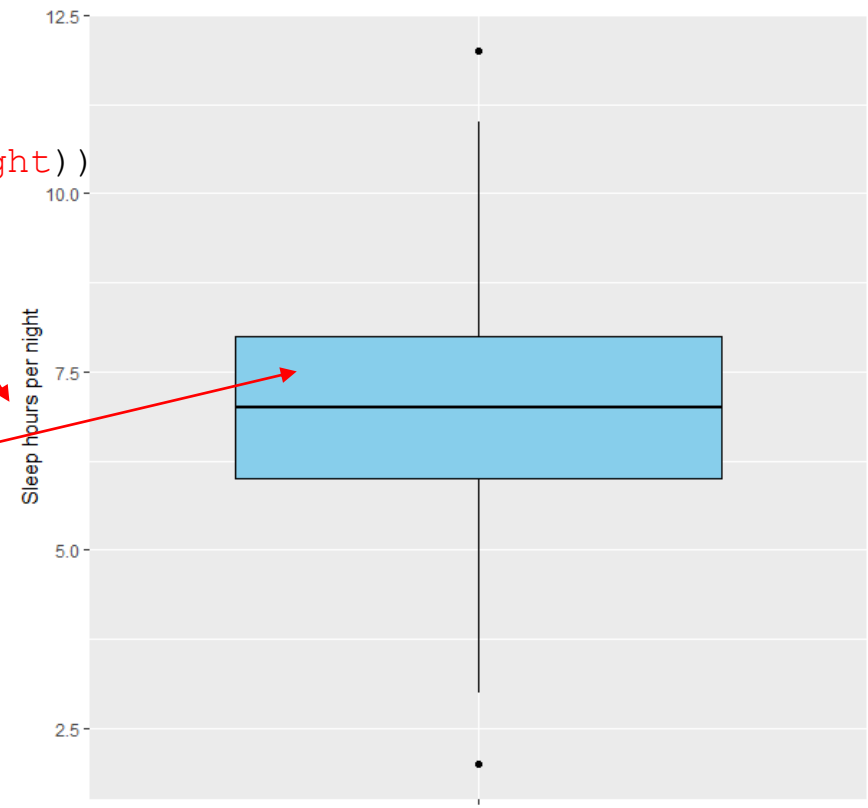
```
ggplot(NHANES, aes(x = "", y = SleepHrsNight))
```

The variable `SleepHrsNight` is plotted on the Y-axis.

Layer 2: type of the plot and setting:

```
geom_boxplot(fill = "skyblue",
             color = "black")
```

The colors of the lines.



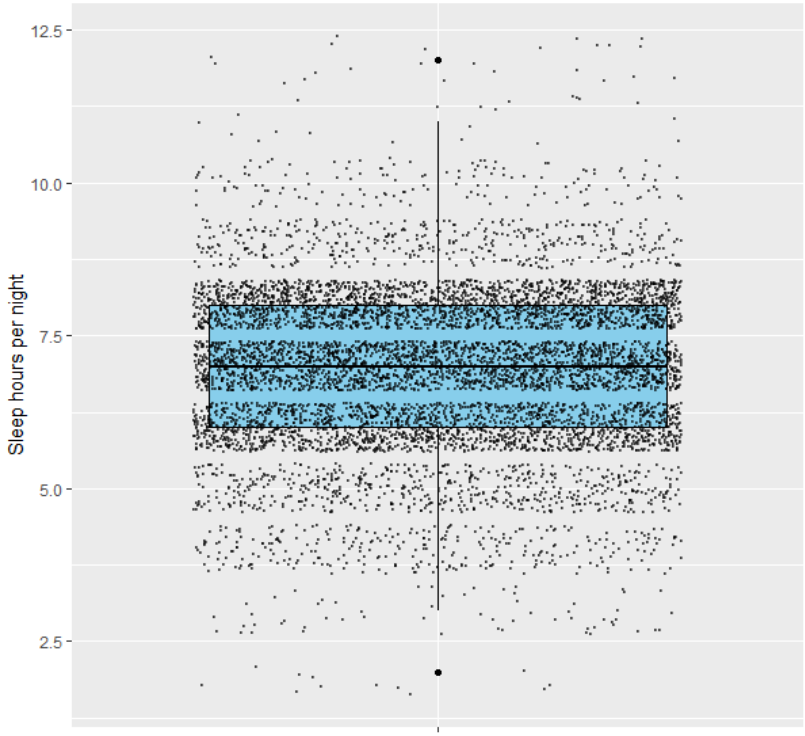
Boxplot of number of sleep hours per night with data points

```
ggplot(NHANES, aes(x = "", y = SleepHrsNight)) +
geom_boxplot(fill = "skyblue", color = "black")+
geom_jitter(aes(x = "", y = SleepHrsNight), color = "black",
size = 0.1, alpha = 0.5)+
ylab("Sleep hours per night")+  xlab("")
```

Layer 3: add the data to the boxplot:

```
geom_jitter(aes(x = "", y = SleepHrsNight),
color = "black", size = 0.1, alpha = 0.5)
```

- `geom_jitter()`: add the data points to the boxplot.
- `alpha=0.5`: control the spread of the data.



Violin plot of number of sleep hours per night

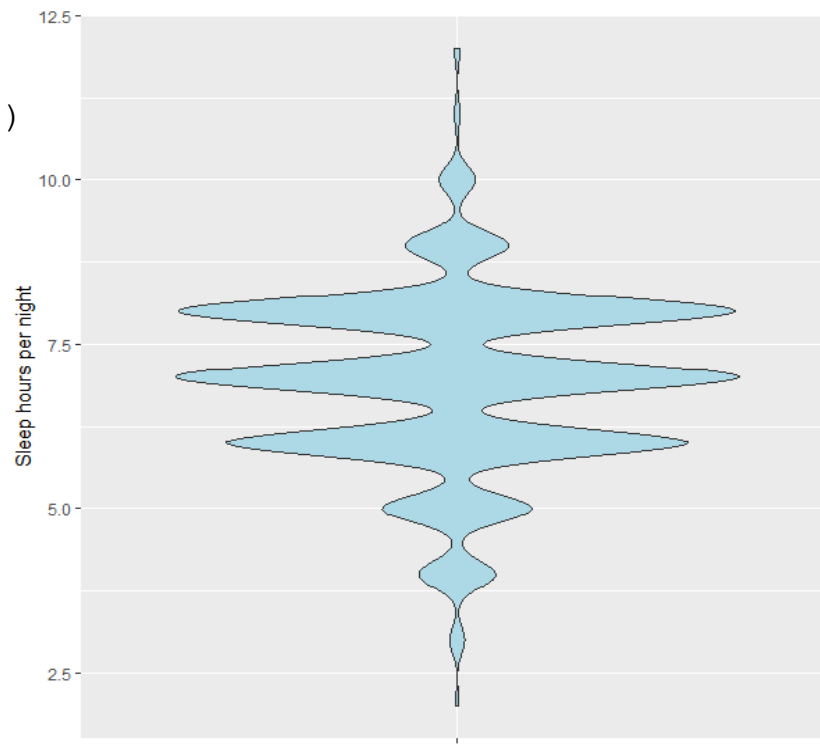
```
ggplot(NHANES, aes(x = "", y = SleepHrsNight)) +
geom_violin(fill = "lightblue")+
xlab("")+  ylab("Sleep hours per night")
```

Layer 1: data and variable to be used:

```
ggplot(NHANES, aes(x = "", y = SleepHrsNight))
```

Layer 2: make a violin plot:

```
geom_violin(fill = "lightblue")+  xlab("")
```

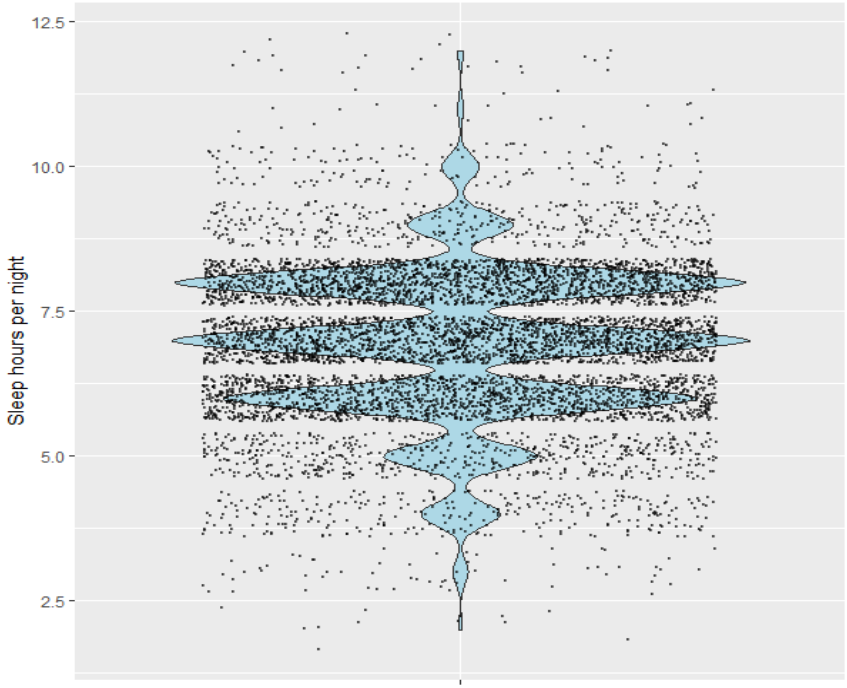


Violin plot of number of sleep hours per night with data points

```
ggplot(NHANES, aes(x = "", y = SleepHrsNight)) +
geom_violin(fill = "lightblue")+
geom_jitter(aes(x = "", y = SleepHrsNight), color = "black",
size = 0.1, alpha = 0.5)+
xlab("")+  ylab("Sleep hours per night")
```

Layer 3: add the data to the plot:

```
geom_jitter(aes(x = "", y = SleepHrsNight),
color = "black", size = 0.1, alpha = 0.5)
```

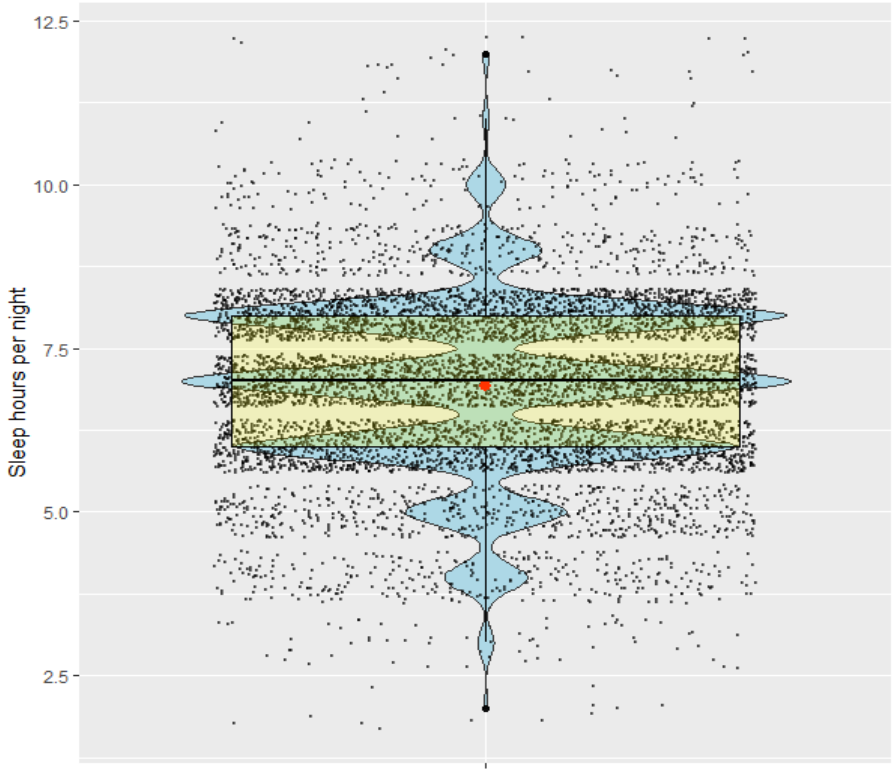


Violin plot of number of sleep hours per night with data points

```
ggplot(NHANES, aes(x = "", y = SleepHrsNight)) +
geom_violin(fill = "lightblue") +
geom_jitter(aes(x = "", y = SleepHrsNight), color = "black"
size = 0.1, alpha = 0.5) +
stat_summary(fun = mean, size = 0.5, color = "red") +
geom_boxplot(fill = "yellow", color = "black", alpha = 0.2) +
xlab("") + ylab("Sleep hours per night")
```

Layer 4: add the mean and boxplot to the plot

```
stat_summary(fun = mean, size = 0.5,
color = "red") +
geom_boxplot(fill = "yellow",
color = "black", alpha = 0.2)
```



Example 3.4

The NHANES dataset

Total cholesterol level

The total cholesterol level

The variable `TotChol` measures total cholesterol level.

```
> NHANES$TotChol
   [1] 3.49 3.49 3.49   NA 6.70 4.86 4.09 5.82 5.82 5.82 4.99 4.24 6.41   NA 4.78
  [16] 5.22 4.86 5.59 6.39 3.00 5.79 5.79 5.04 4.81 4.81 4.68 4.14 5.12 5.61 5.61
  [31] 4.16 5.95 4.16 4.16 4.97 4.53 4.53 2.61 4.27 4.27 3.62 3.62 3.62 3.62 5.74
  [46] 4.32 3.36 4.03 5.30 4.24 4.24 3.85 3.85 4.42 4.60 4.37   NA 4.63 4.63 4.63
  [61]   NA 2.66 4.09   NA   NA 5.33 5.33 5.33 4.03   NA   NA 7.32 7.32 4.32 4.45
  [76] 4.29 4.29 4.29 3.83 5.79 5.79 5.79 4.84 4.84 4.84 4.84 3.15 3.15 4.65 7.03
  [91] 7.03 3.90 8.09 4.97 6.03 4.81 4.01 4.55 4.22 3.90 5.69   NA   NA 3.72 3.72
  ……
```

- 10000 observations.
- Numerical variable with missing values (NA).

Histogram of the total cholesterol level

```
ggplot(NHANES, aes(x = TotChol)) +
geom_histogram(fill = "skyblue", color = "black") +
ylab("Frequency") + xlab("The total cholesterol level")
```

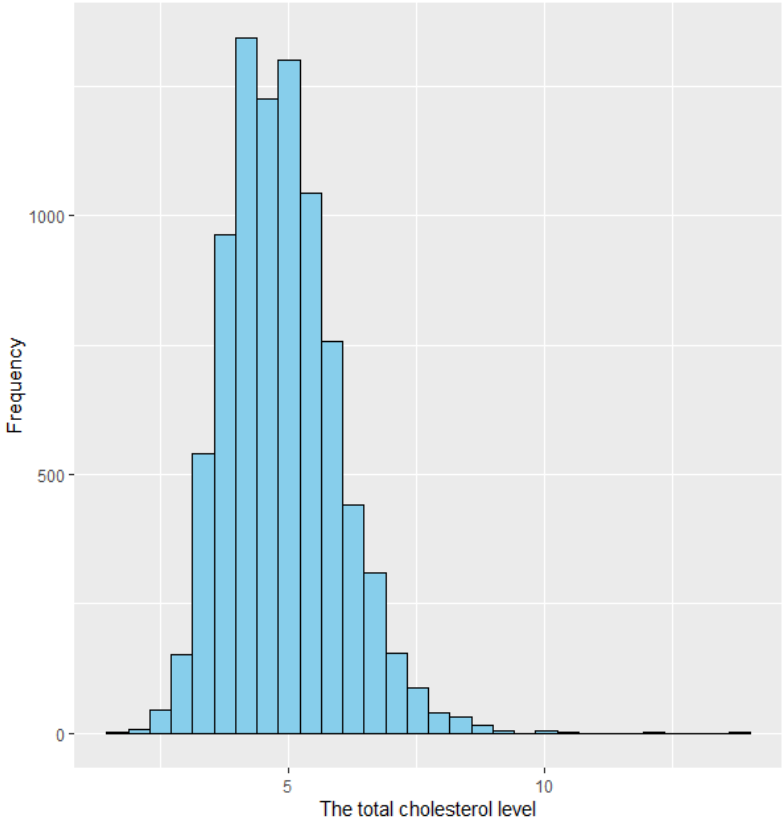
Layer 1: data and variable to be used:

```
ggplot(NHANES, aes(x = TotChol))
```

Layer 2: the plot type to be used:

```
geom_histogram(fill = "skyblue",
               color = "black")
```

- `geom_histogram()`: plot a histogram of the data.
  - Selecting the color of the bars: `fill=…`.
  - Selecting the color of the lines separate the bars: `colors=…`

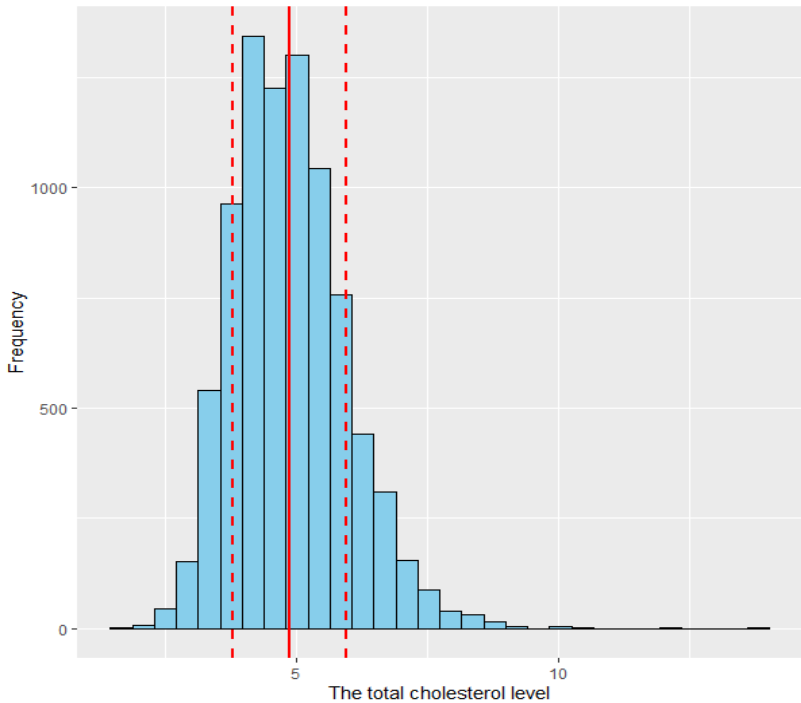


Histogram of the total cholesterol level

```
ggplot(NHANES, aes(x = TotChol)) +
geom_histogram(fill = "skyblue", color = "black") +
geom_vline(aes(xintercept = TotChol_summary$mean_TotChol), color = "red",
linetype = "solid", size = 1) +
geom_vline(aes(xintercept = (TotChol_summary$mean_TotChol -
TotChol_summary$sd_TotChol)), color = "red", linetype = "dashed", size = 1) +
geom_vline(aes(xintercept = (TotChol_summary$mean_TotChol +
TotChol_summary$sd_TotChol)), color = "red", linetype = "dashed", size = 1) +
ylab("Frequency") +    xlab("The total cholesterol level")
```

Layer 3: add the lines of the mean and +/- SD

```
geom_vline(aes(xintercept =
TotChol_summary$mean_TotChol), color =
"red", linetype = "solid", size = 1) +
geom_vline(aes(xintercept =
(TotChol_summary$mean_TotChol -
TotChol_summary$sd_TotChol)), color =
"red", linetype = "dashed", size = 1) +
geom_vline(aes(xintercept =
(TotChol_summary$mean_TotChol +
TotChol_summary$sd_TotChol)), color =
"red", linetype = "dashed", size = 1)
```



Boxplot of the total cholesterol level

```
ggplot(NHANES, aes(x = "", y = TotChol)) +
geom_boxplot(fill = "skyblue", color = "black")+
ylab("The total cholesterol level") + xlab("")
```

Layer 1: data and variable to be used:

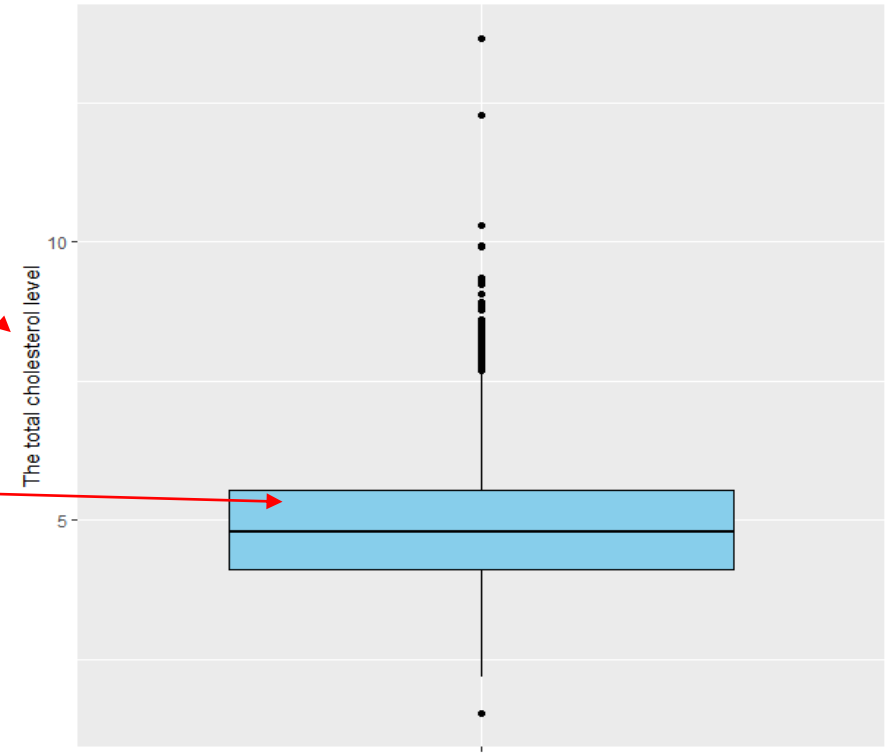
```
ggplot(NHANES, aes(x = "", y = TotChol))
```

The variable total cholesterol level is plotted on the Y-axis.

Layer 2: type of the plot and setting:

```
geom_boxplot(fill = "skyblue",
             color = "black)
```

The colors of the lines.

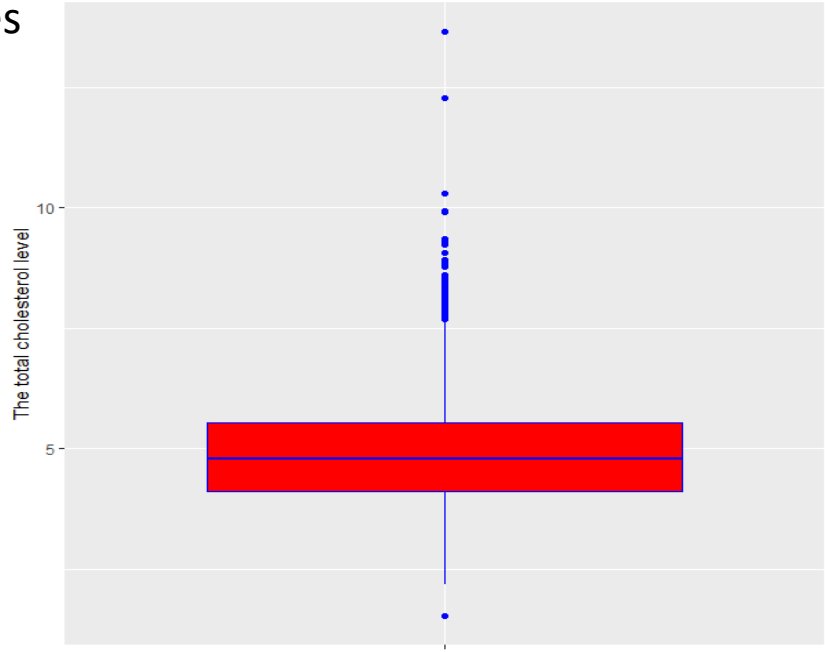


Boxplot of the total cholesterol level

```
ggplot(NHANES, aes(x = "", y = TotChol)) +
geom_boxplot(fill = "red", color = "blue")+
ylab("The total cholesterol level") + xlab("")
```

Layer 2: Changing colors of the box and the lines

```
geom_boxplot(fill = "red",
             color = "blue")
```

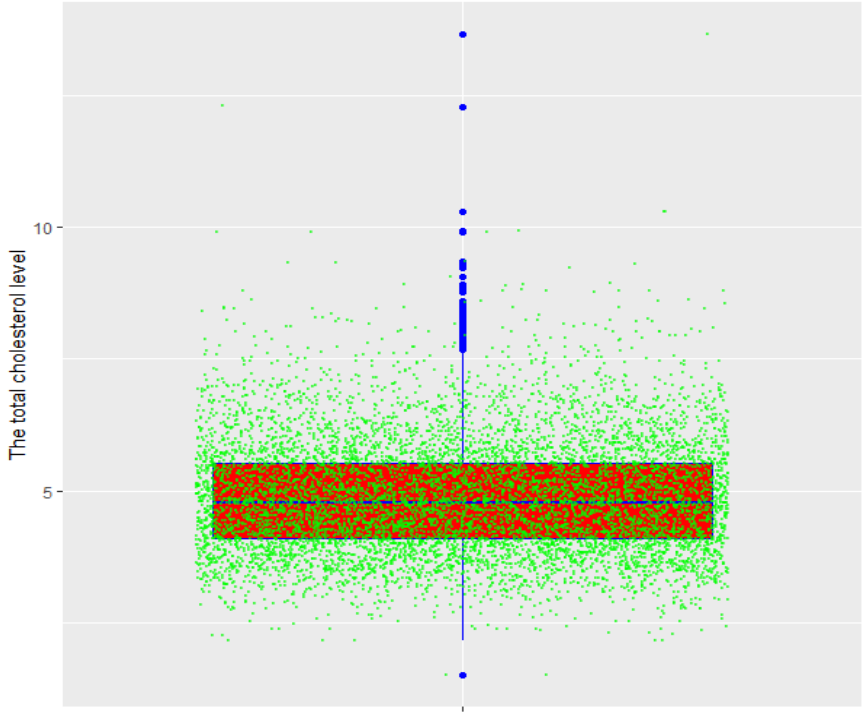


Boxplot of the total cholesterol level with data points

```
ggplot(NHANES, aes(x = "", y = TotChol)) +
geom_boxplot(fill = "red", color = "blue") +
geom_jitter(aes(x = "", y = TotChol), color = "green",
size = 0.1, alpha = 0.5) +
ylab("The total cholesterol level")+ xlab("")
```

Layer 3: Changing colors of the box, lines, ar points.

```
geom_jitter(aes(x = "", y = TotChol),
color = "green",
size = 0.1, alpha = 0.5) +
```



Violin plot of the total cholesterol level

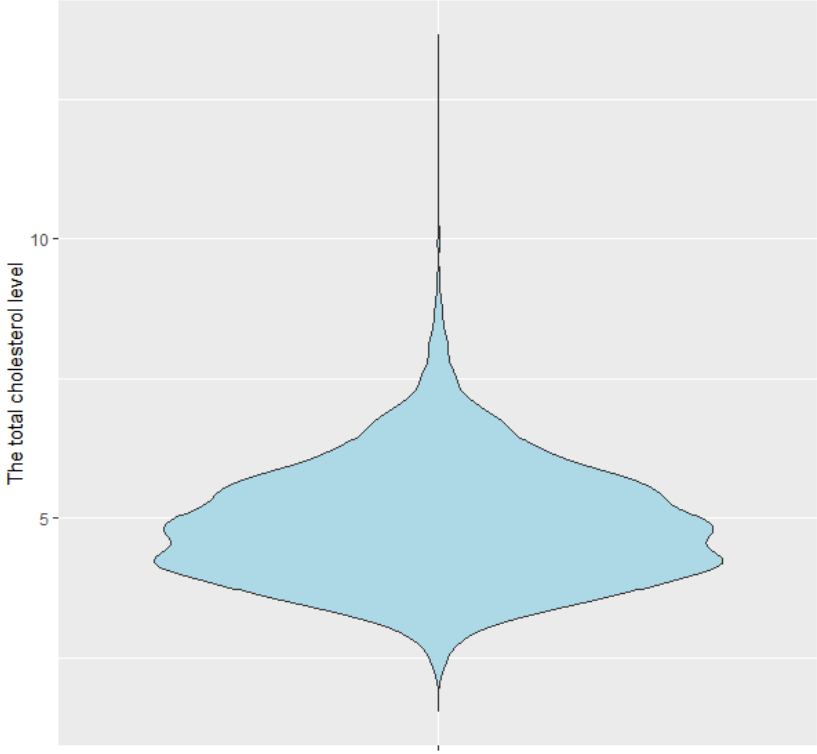
```
ggplot(NHANES, aes(x = "", y = TotChol)) +
geom_violin(fill = "lightblue")+
xlab("")+ ylab("The total cholestrol level")
```

Layer 1: data and variable to be used:

```
ggplot(NHANES, aes(x = "", y = TotChol))+
```

Layer 2: make a violin plot:

```
geom_violin(fill = "lightblue")
```



Violin plot of the total cholesterol level with data points

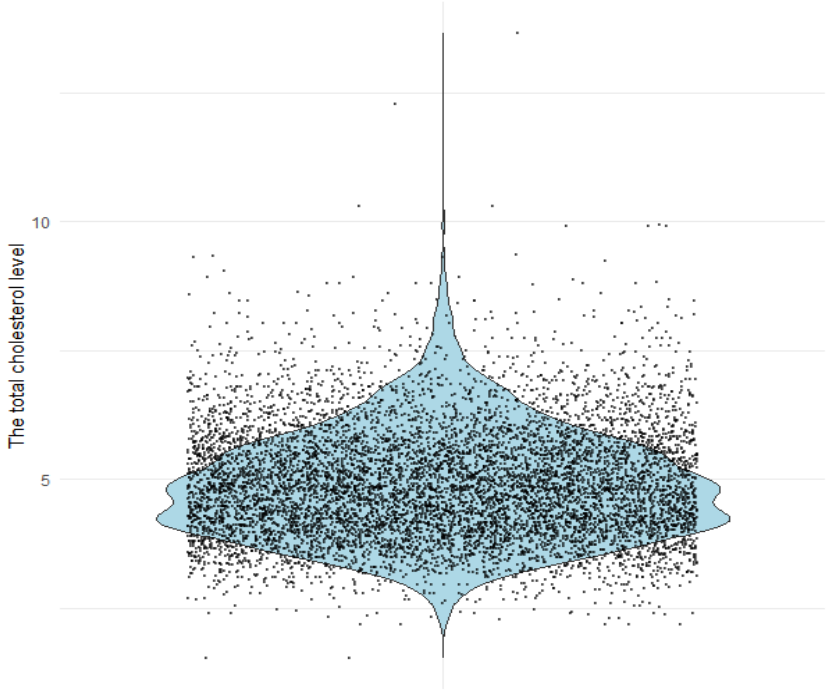
```
ggplot(NHANES, aes(x = "", y = TotChol)) +
geom_violin(fill = "lightblue")+
geom_jitter(aes(x = "", y = TotChol), color = "black",
size = 0.1, alpha = 0.5) +
xlab("")+  ylab("The total cholesterol level") +
theme_minimal()
```

Layer 3: add the data to the plot:

```
geom_jitter(aes(x = "", y = TotChol),
color = "black",
size = 0.1, alpha = 0.5)
```

Layer 4: Changing background:

```
theme_minimal()
```



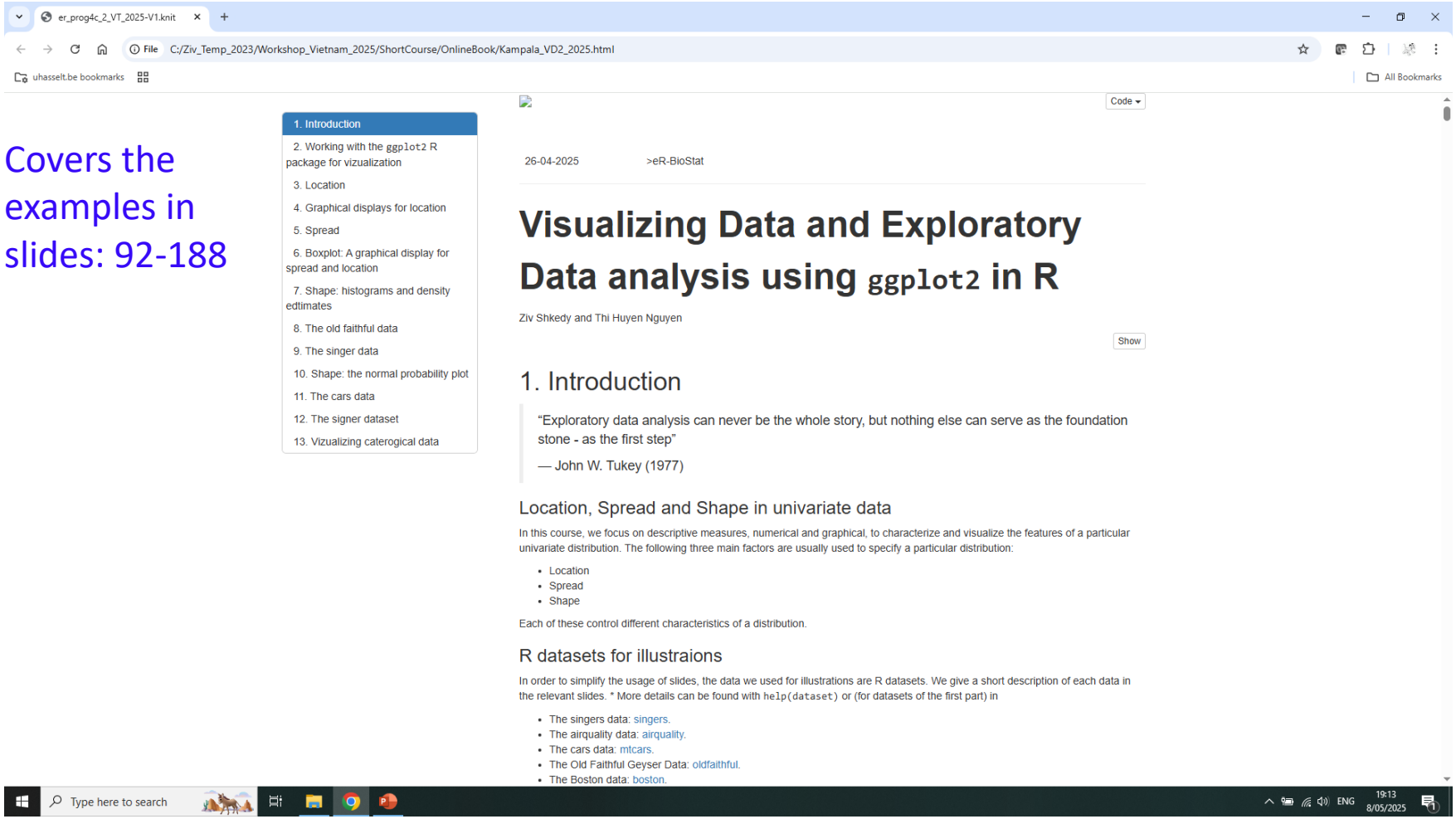
Part 4

Visualization of two numerical variables

Rmd program: `er_prog4c_2_VT_2025_V1.Rmd`
HTML file: `eR_Biostat_Kampala_VD2_2025.html`

The HTML file

Covers the examples in slides: 92-188



HTML file: **`Kampala_VD2_2025.html`**

Example 4.1

The `mtcars` data

Exploring the correlation between two numerical variables across a factor

The `mtcars` data

- The data was extracted from the 1974 *Motor Trend* US magazine.
- The data gives information about fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).
- For our analysis we focus on miles per gallon and horse power of the cars.

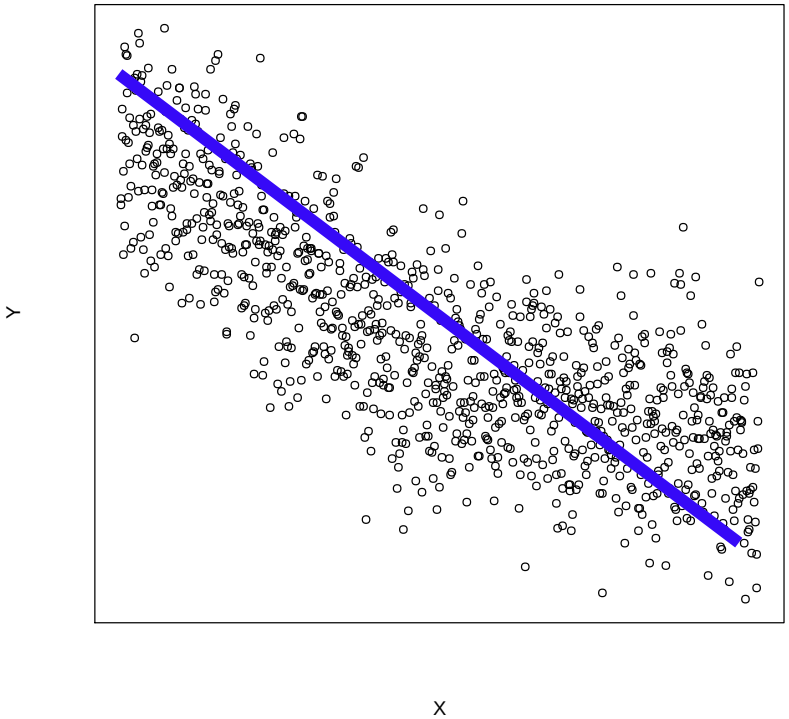
The `mtcars` data

```{r,echo=TRUE}
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

- Numerical variables:
  - `hp`: Horse power.
  - `mpg`: Mile per gallon.
- `cyl`: A factor with three levels.
- Represent the number of cylinders.

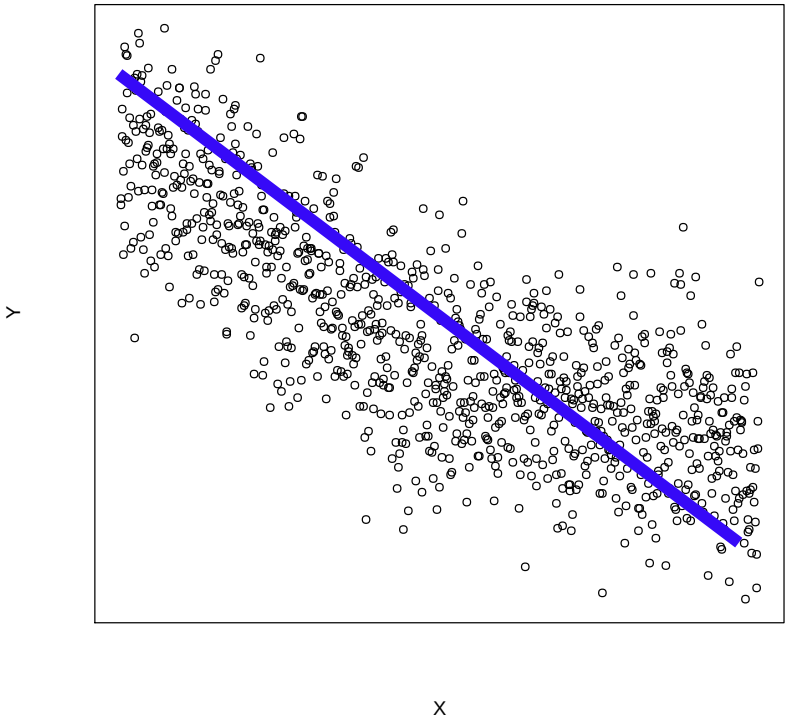
What do we want to visualize ?

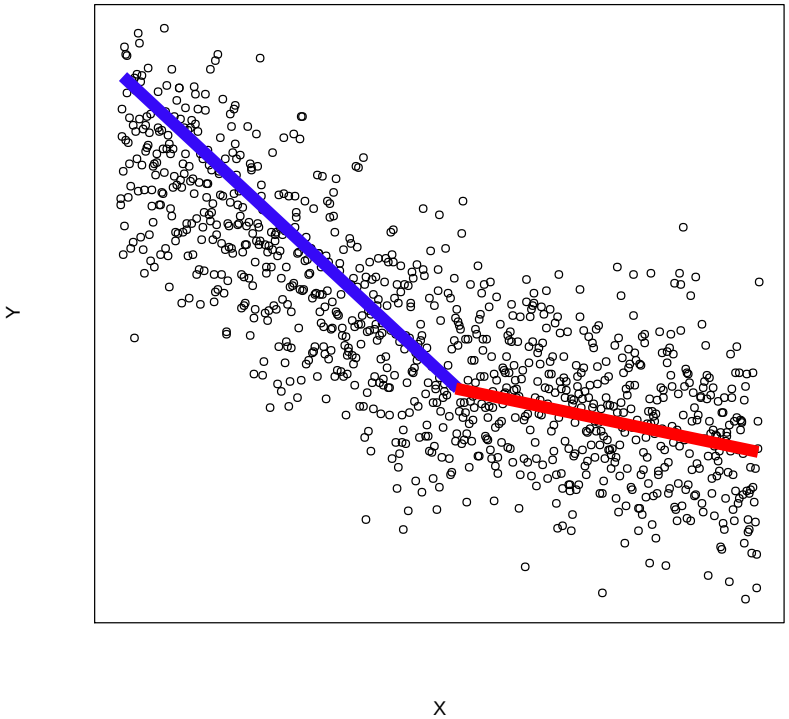


- Our aim: explore the correlation between X and Y.
- A scatterplot of two numerical variables X and Y.

- Can we summarize the relationship with a straight line ?

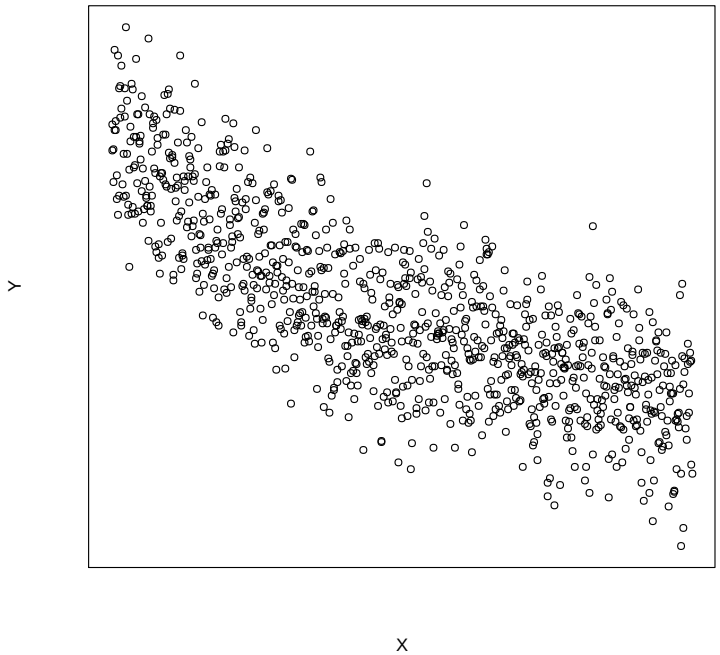
What do we want to visualize ?

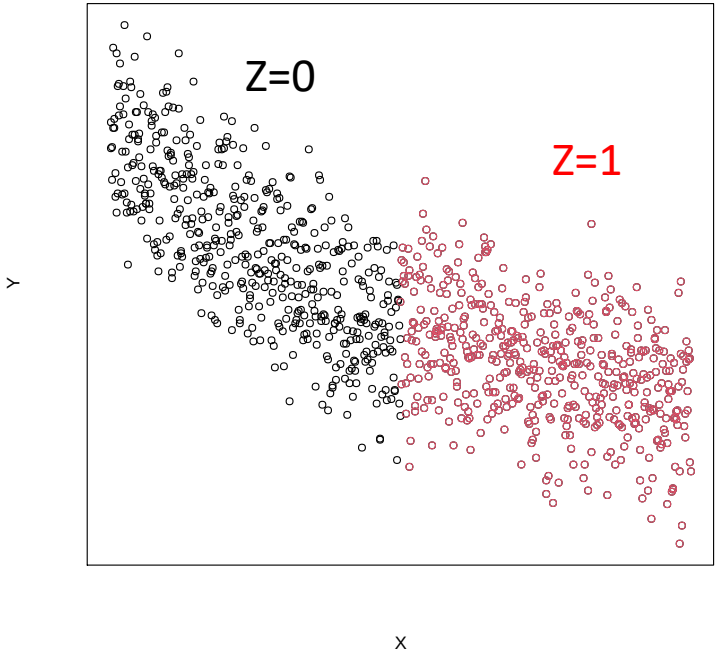




- Can we summarize the relationship with a straight line ?

What do we want to visualize ?





- Suppose that we have in the data a factor (Z) with two levels, what is the influence of this factor to the relationship between X and Y ?

R code for the example

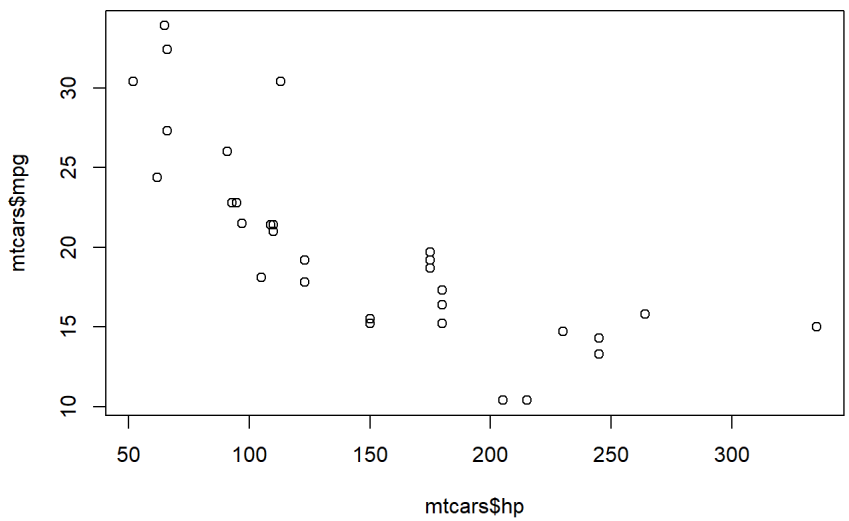
```
x1<-seq(from=0,to=0.5,length=500)
x2<-seq(from=0.5,to=1,length=500)
x3<-c(x1,x2)
y1<-2+(-2)*x1+rnorm(500,0,0.25)
y2<-1.5+(-0.5)*x2+rnorm(500,0,0.25)
y3<-c(y1,y2)
plot(x3,y3,xaxt="n",yaxt="n",xlab="X",ylab="Y")
```

The `mtcars` data: scatterplot of `hp` vs. `mpg`

Basic scaterplot in R: hp VS. mpg.

```
plot(mtcars$hp,mtcars$mpg)
```

X Y



Scatterplot: `hp` vs. `mpg` (`ggplot2`)

Layer 1: Basic scaterplot using `ggplot2` R, `hp` VS. `mpg`.

```
gg <- ggplot(mtcars, aes(hp,mpg)) +
      geom_point() +
      labs(x = "Horsepower",y= "Miles Per Gallon")
gg
```

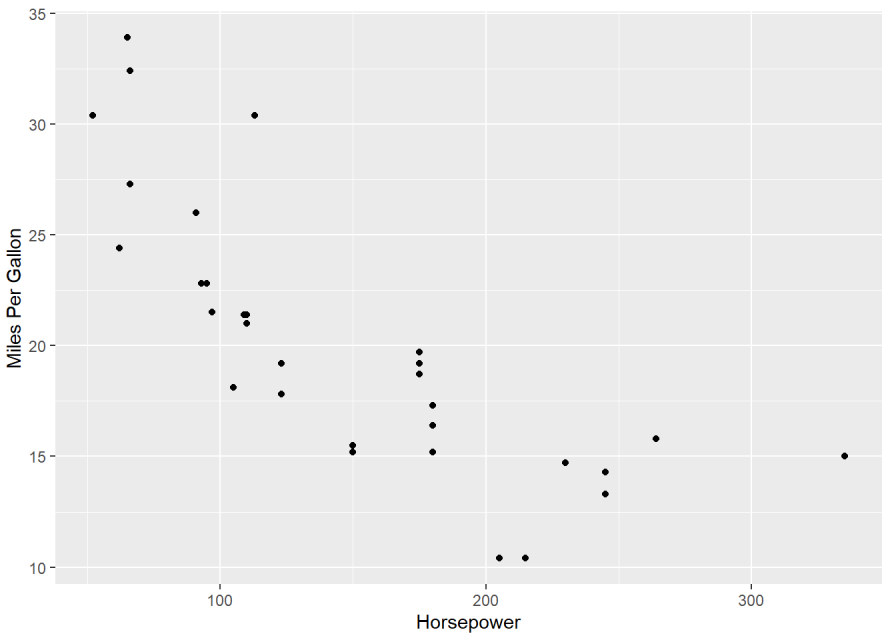
`aes(hp,mpg)`

X Y

Which variable to use in the plot (x and y).

`geom_point()`

Which plot to produce. **`geom_point`**=scatterplot.



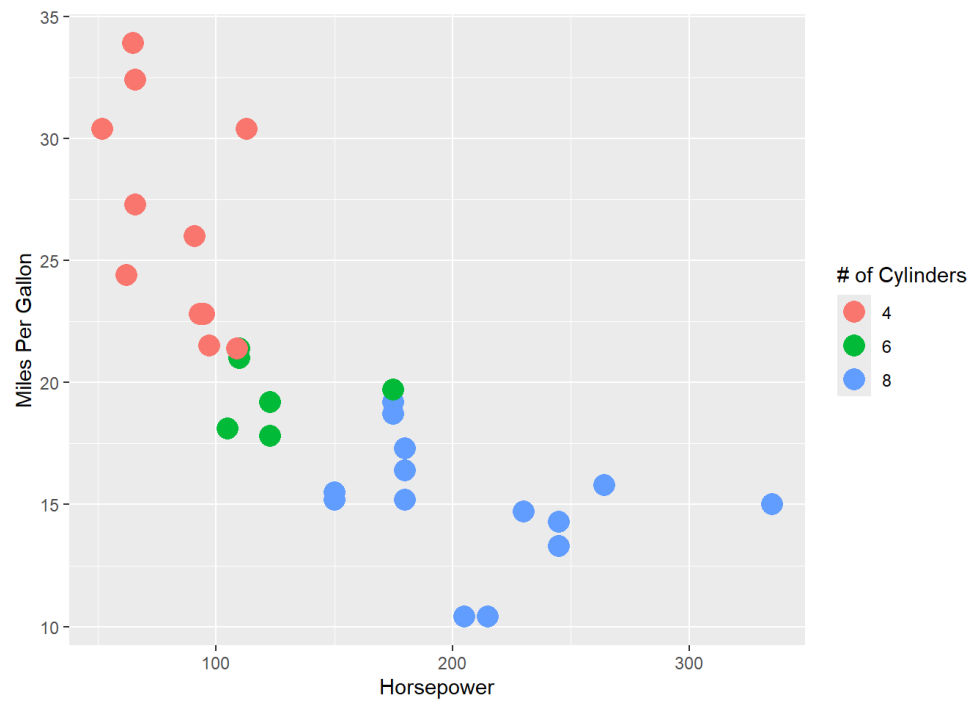
Scatterplot: `hp` vs. `mpg` (ggplot2)

Layer 2: vizualize the effect of number of cylinders on mpg.

```
gg <- ggplot(mtcars, aes(hp, mpg)) +
      geom_point(aes(color=as.factor(cyl)),
                 size=5) +
      labs(x = "Horsepower",y= "Miles Per
           Gallon",color= "# of Cylinders")
gg
```

```
aes(color=as.factor(cyl))
```

Use different colors by the level of the number of cylinders (=factor).



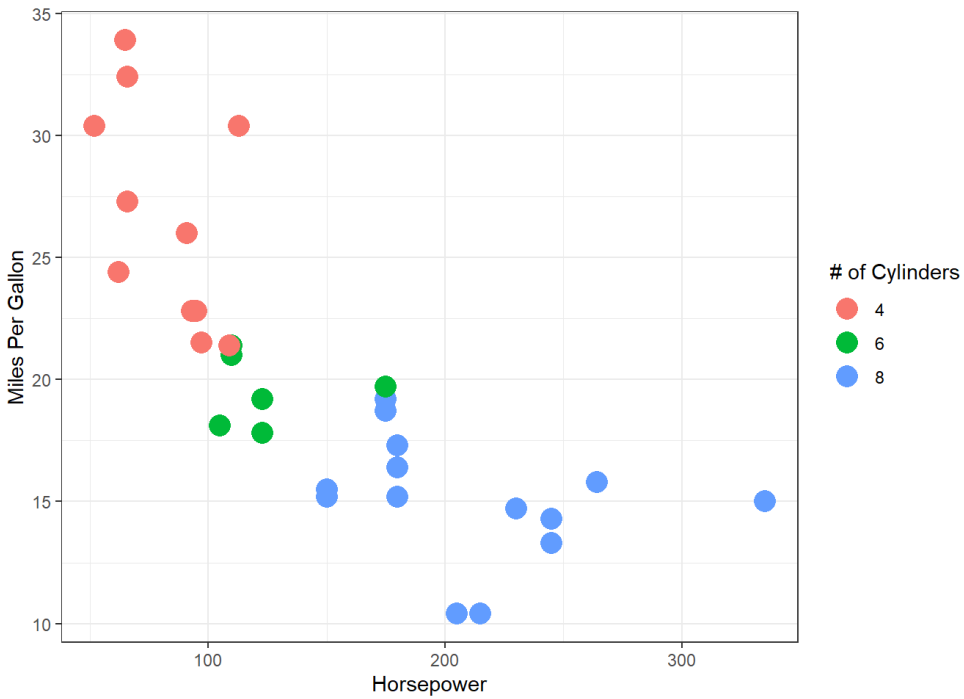
Scatterplot: hp vs. mpg: `ggplot2`

Layer 3: change background color.

```
gg <- ggplot(mtcars, aes(hp, mpg)) +
      geom_point(aes(color=as.factor(cyl)), size=5) +
      labs(x = "Horsepower",y= "Miles Per Gallon",color= "# of Cylinders")+
      theme_bw()
gg
```

```
theme_bw()
```

Black & white



Scatterplot: hp vs. mpg using `ggplot2`

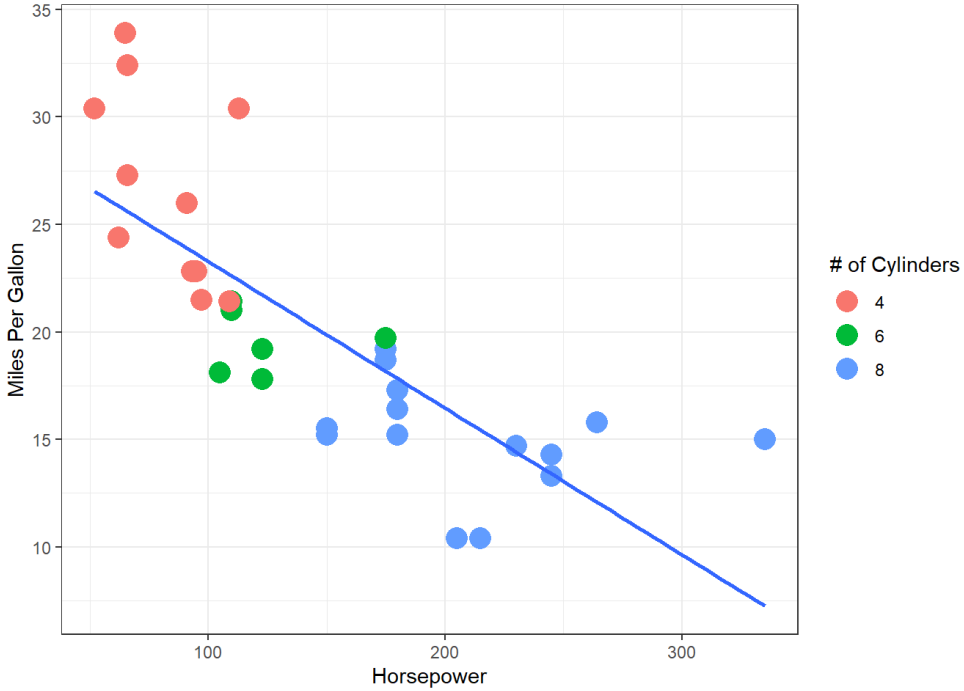
Layer 4: add a regression line to the figure.

```
gg <- ggplot(mtcars, aes(hp, mpg)) +
      geom_point(aes(color=as.factor(cyl)), size=5) +
      geom_smooth(method="lm", se=FALSE) +
      labs(x = "Horsepower",y= "Miles Per Gallon", color= "# of Cylinders") +
      theme_bw()
gg
```

```
geom_smooth(method="lm", se=FALSE)
```

Add a regression line to the plot.

$$mpg_i = \beta_0 + \beta_1 \times hp_i + \varepsilon_i$$



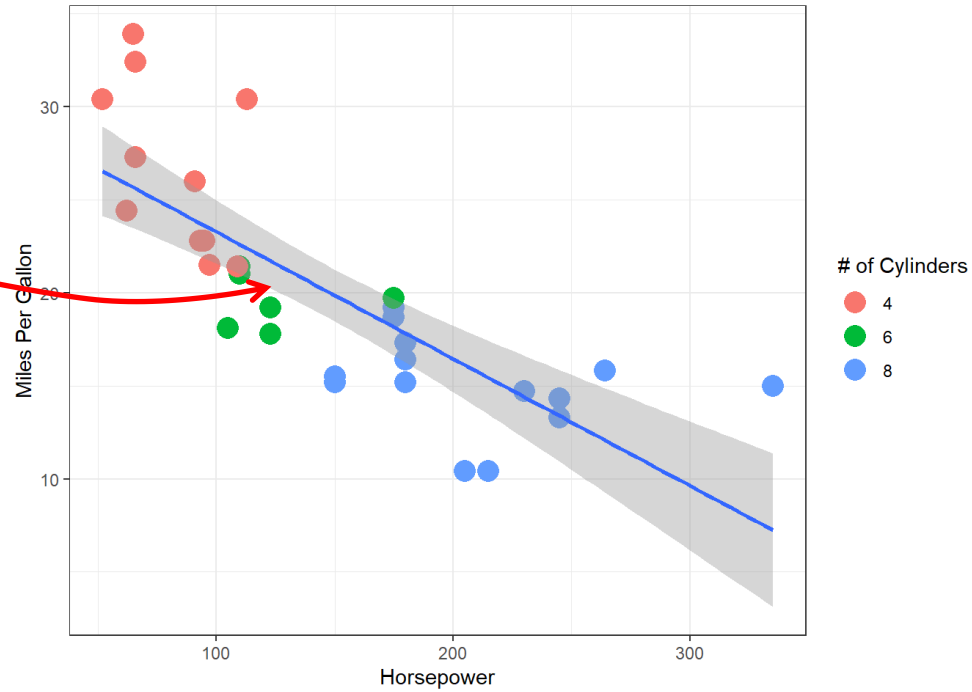
Scatterplot: hp vs. mpg using `ggplot2`

Layer 5: add a C.I around the regression line.

```
gg <- ggplot(mtcars, aes(hp, mpg)) +
      geom_point(aes(color=as.factor(cyl)), size=5) +
      geom_smooth(method="lm", se=TRUE) +
      labs(x = "Horsepower",y= "Miles Per Gallon",color= "# of Cylinders")+
      theme_bw()
gg
```

```
geom_smooth(method="lm", se=TRUE)
```

$$mpg_i = \beta_0 + \beta_1 \times hp_i + \varepsilon_i$$



Scatterplot: hp vs. mpg using `ggplot2`

Layer 5: add a smoother + C.I.

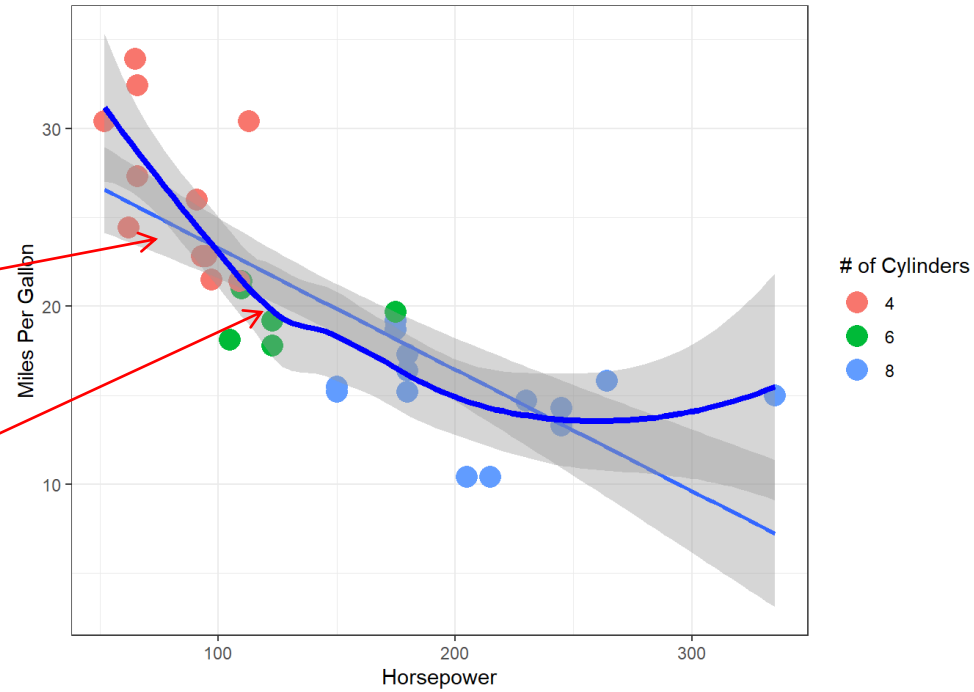
```
gg <- ggplot(mtcars, aes(hp, mpg)) +
      geom_point(aes(color=as.factor(cyl)), size=5) +
      geom_smooth(method="lm", se=TRUE) +
      geom_smooth(method="loess",colour = "blue", size = 1.5)+
      labs(x = "Horsepower",y= "Miles Per Gallon",color= "# of Cylinders") +
      theme_bw()

gg
```

```
geom_smooth(method="lm", se=TRUE)
```

Add a loess smoother to the plot:

```
geom_smooth(method="loess",
            colour = "blue",
            size = 1.5)
```



Scatterplot: hp vs. mpg using `ggplot2`

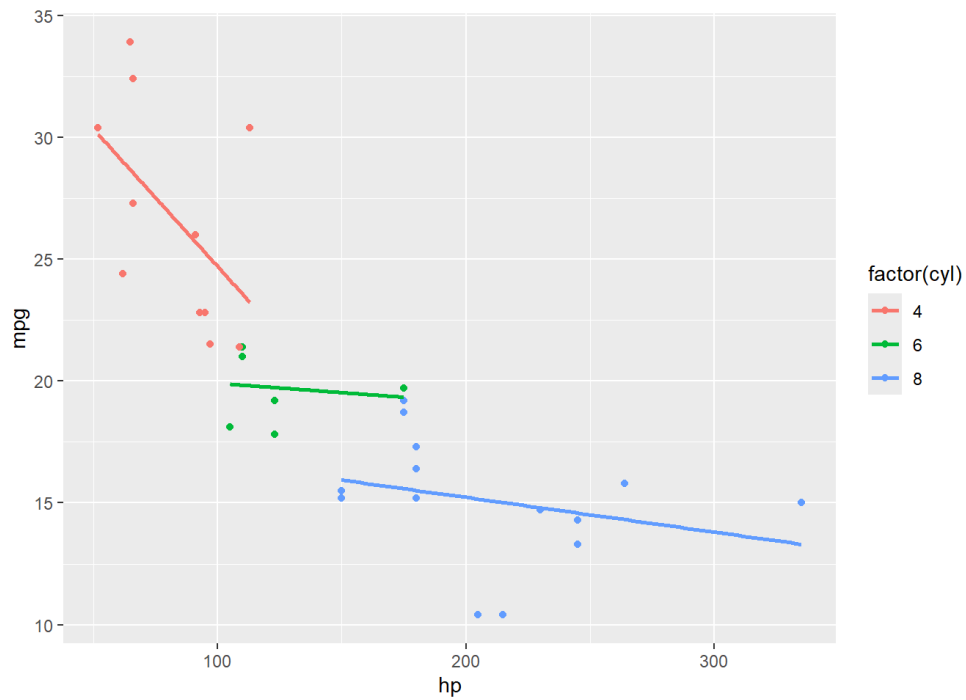
New scatterplot: Visualizing correlation patterns across cylinders

```
qplot(hp,mpg,data = mtcars, colour = factor(cyl))+
geom_smooth(method = "lm",se = F)
```

Produce a scatterplot with colored by number of cylinders.

```
geom_smooth(method = "lm",se = F)
```

Add a regression line (per group).



Part 5

Visualization of numerical variables across a factor variable in one sample

Rmd program: `er_prog4c_2_VT_2025_V1.Rmd`
HTML file: `Kampala_VD2_2025.html`

Example 5.1

The `singer` dataset

Heights of singers

The `singer` dataset

- The `singer` data set gives information about Heights of New York Choral Society singers .
- The `singer` data set is found in the `lattice` R package.

The `singer` dataset

- Information about:
  - Heights in inches of the singers in the New York Choral Society in 1979.
  - The data are grouped according to voice part (a factor).
  - The vocal range for each voice part increases in pitch according to the following order: Bass 2, Bass 1, Tenor 2, Tenor 1, Alto 2, Alto 1, Soprano 2, Soprano 1.

The `singer` dataset

```
head(singer)
```

```
##   height voice.part
## 1     64  Soprano 1
## 2     62  Soprano 1
## 3     66  Soprano 1
## 4     65  Soprano 1
## 5     60  Soprano 1
## 6     61  Soprano 1
```

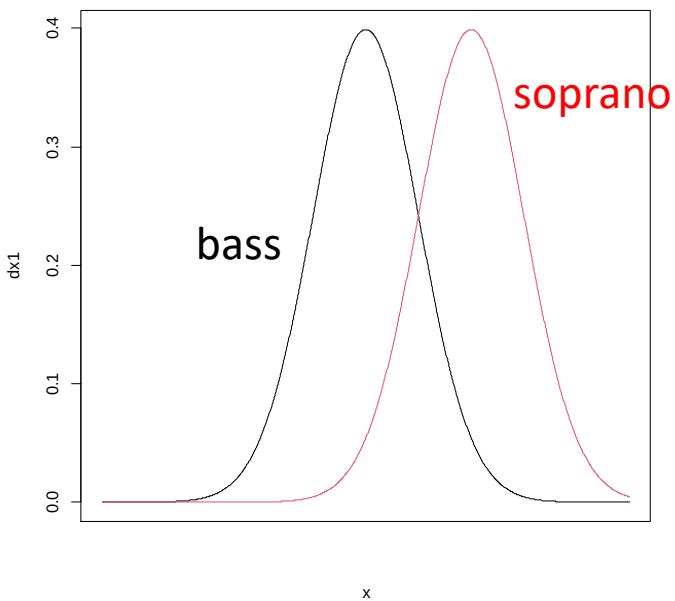
Factor with 8 levels.

Height of the singers: numerical variable.

- Male singers: Bass 2, Bass 1, Tenor 2, Tenor 1.
- Female singers: Alto 2, Alto 1, Soprano 2, Soprano 1.

What do we want to visualize ?

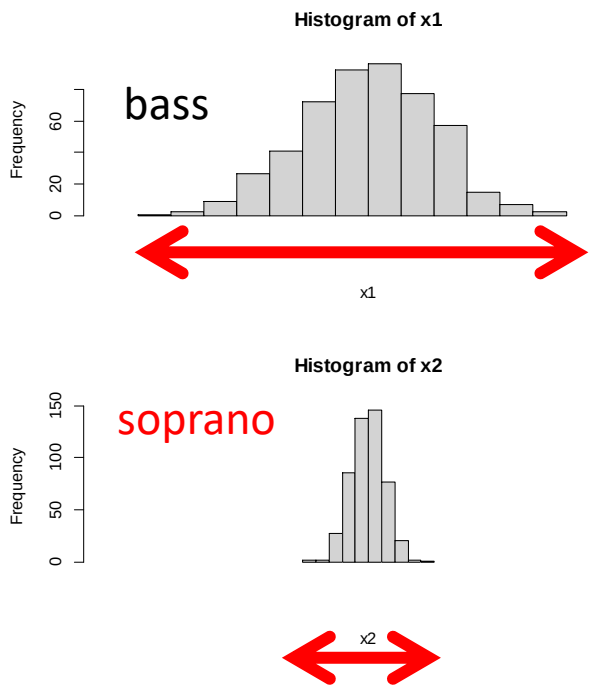
- Main focus on of the analysis (1):
  - Location: shift in singers' height across voice part groups.
  - For Example:



- How can we visualize the shift in height across voice groups?
- Can we say that, for example, bass singers are taller than soprano singers ?

What do we want to visualize ?

- Main focus on of the analysis (2):
  - Spread: comparison of variability ?
  - For Example:



- How can we visualize the difference in variability across voice groups?
- Can we say that, for example, that the height of bass singers has larger variability than the height of soprano singers ?

R code for the example

```
par(mfrow=c(2,1))
x1<-rnorm(500,0,1)
hist(x1,xaxt="n",yxat="n",xlim=c(-4,4))
x2<-rnorm(500,0,0.25)
hist(x2,xaxt="n",yxat="n",xlim=c(-4,4))
```

The `singer` dataset: dotplot (`lattice`)

```
dotplot(singer$voice.part~singer$height,
        aspect=1,
         xlab="Mean Height (inches)")
```

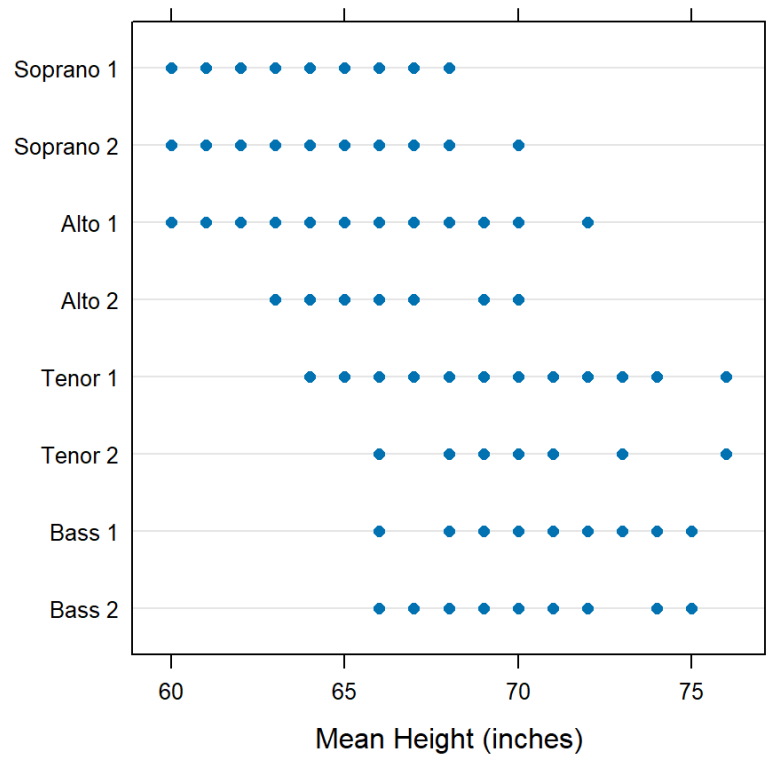
A `lattice` function to produce a dotplot.

The `lattice` R package: an "old" graphical R package.

`singer$voice.part~singer$height`

A factor.

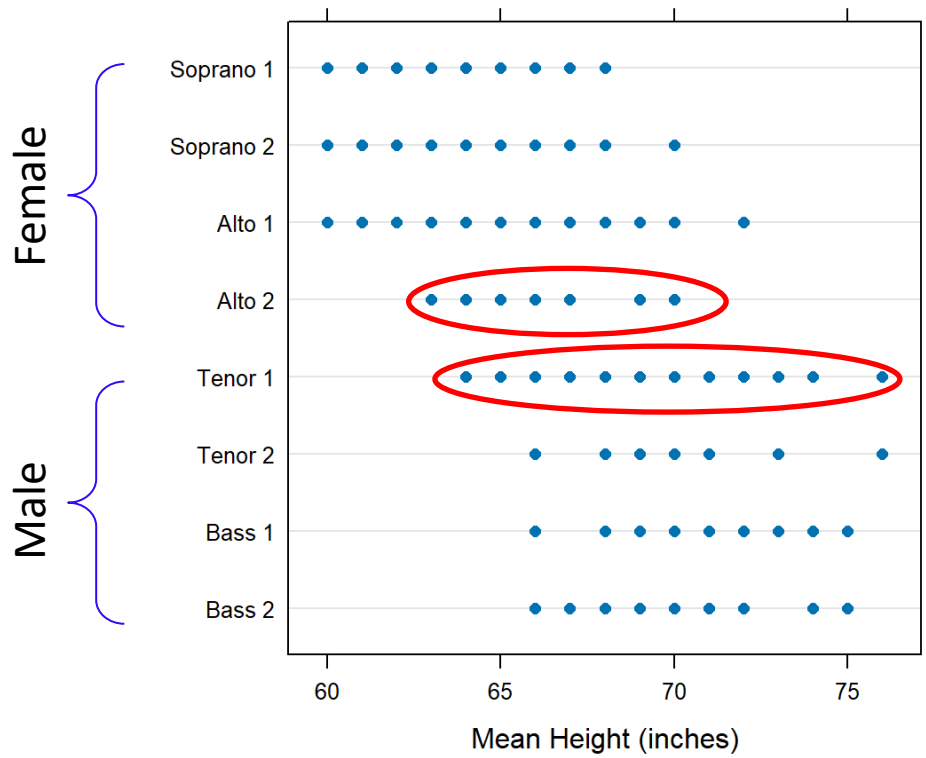
A continuous variable.



The `singer` dataset: dotplot (`lattice`)

- Graphical display for location.
- In general: female are shorter than male.

- Variability: Tenor1 compared to Alto 2.



The `singer` dataset: dotplot using `ggplot2`

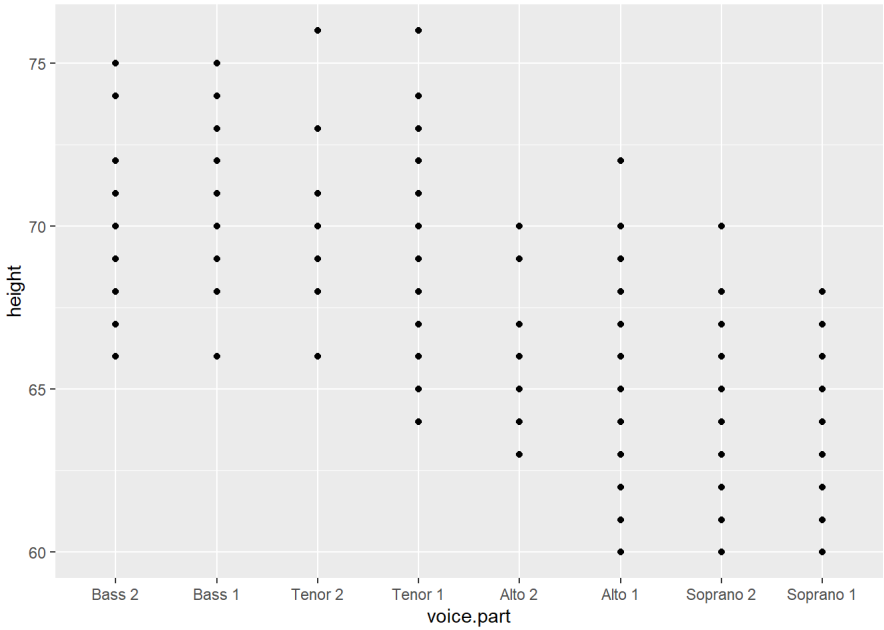
Layer 1: produce a dotplot.

```
ggplot(singer,aes(voice.part,height)) +
       geom_point()
```

`aes(voice.part,height)`

A factor Continuous variable

`geom_point()`: Which plot to produce.
`geom_point`=scatterplot.



Main problem: two subject with the same height will have the same "point" in the figure.

The `singer` dataset: dotplot using `ggplot2`

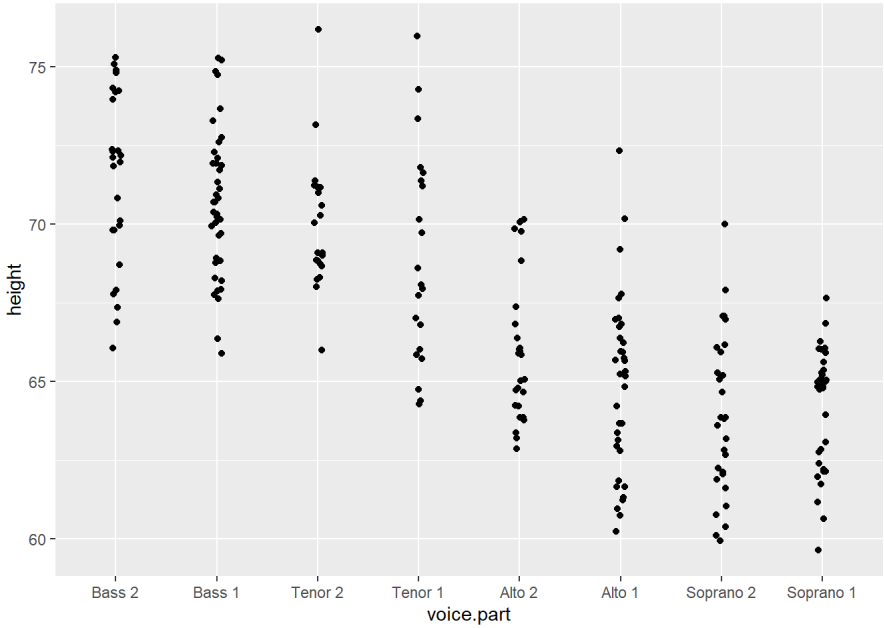
Layer 1: dotplot with jitter.

```
gplot(singer,aes(voice.part,height)) +
geom_jitter(position = position_jitter(width = .05))
```

```
geom_jitter(position =
            position_jitter(width = .05))
```

Create a variability around the "center" so a group will be represented with a cloud of points.

Main advantage: we can get an idea about the sample size (per group).



The `singer` dataset: dotplot using `ggplot2`

Layer 2: dadding the information about the groups means to the dotplot.

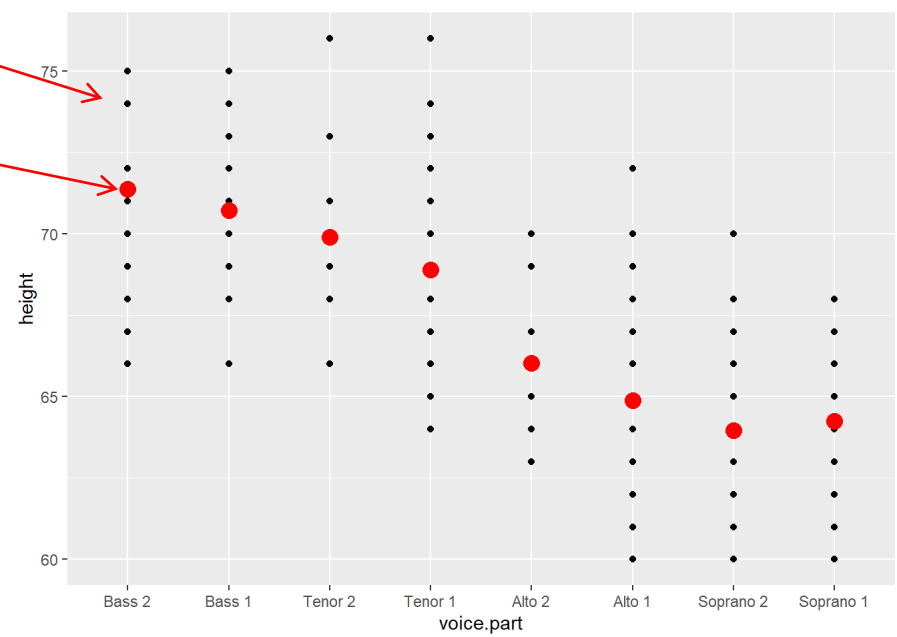
```
ggplot(singer, aes(voice.part,height)) +
geom_point() +
stat_summary(geom = "point", fun.y = "mean", colour = "red", size = 4)
```

Produce a dotplot without jitter

```
stat_summary(geom = "point",
             fun.y = "mean",
             colour = "red", size = 4)
```

Calculate the group mean (=summary for location).

Mean heights of male vs. female.



The `singer` dataset: boxplot using `lattice`

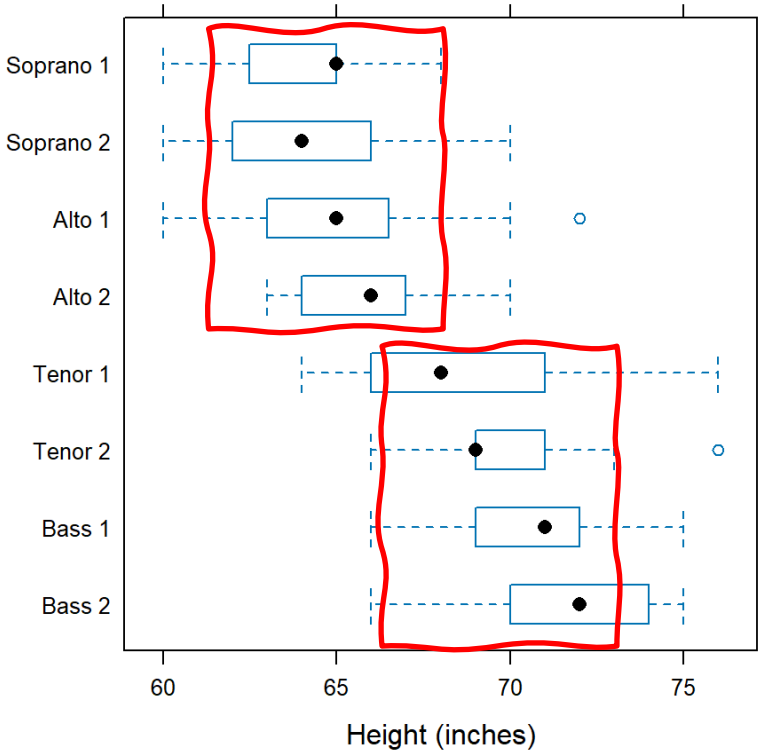
```
bwplot(as.factor(singer$voice.part)~ singer$height,
        data=singer,
         aspect=1,
         xlab="Height (inches)”)
```

The center is represented by the median.

Variability: the length of the box.

Outliers.

Male vs. Female: shift in location.

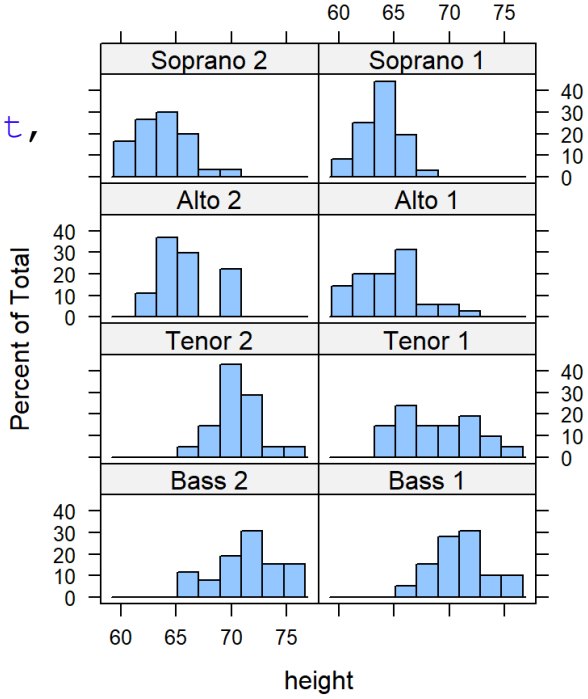


The `singer` dataset: multiway histogram using `lattice`

```
histogram(~ singer$height | singer$voice.part,
          data=singer,
          layout = c(2, 4),
         aspect = 0.5,
          xlab = "height")
```

```
histogram(~ singer$height | singer$voice.part,
```

```
histogram(~ continuous variable | factor
```



The `singer` dataset: multiway histogram using `ggplot2`

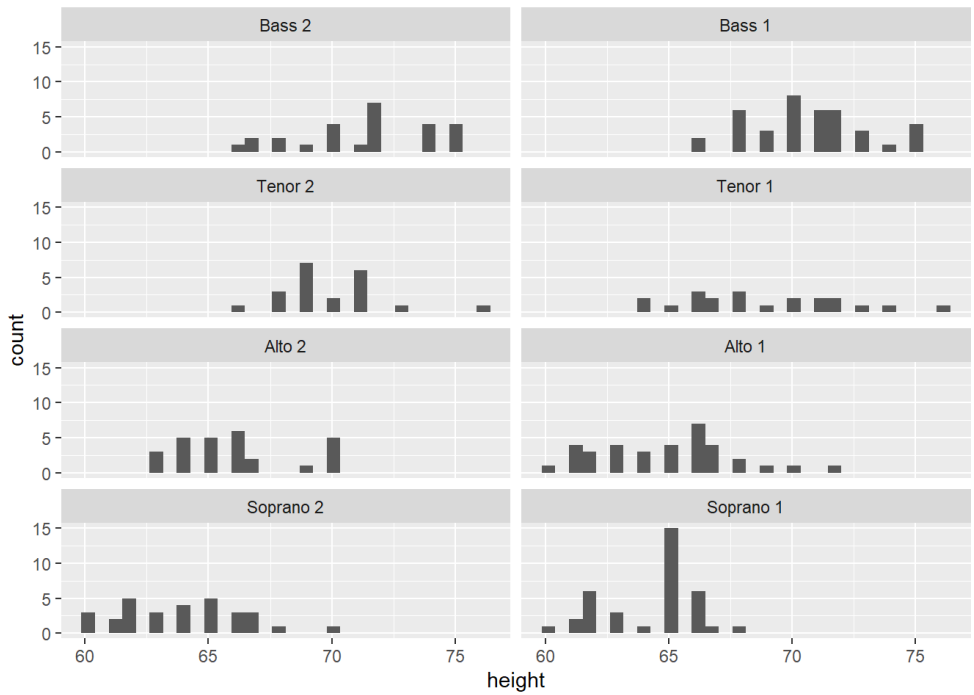
Layer 1: A multiway histogram of the height by voice group.

```
ggplot(singer, aes(height)) +
geom_histogram() +
facet_wrap(~voice.part,ncol = 2)
```

```
ggplot(singer, aes(height)) +
geom_histogram() +
```

Plot histogram of the variable.

Use the variable height.



The `singer` dataset: multiway histogram using `ggplot2`

How to create a multi-panel figure ?

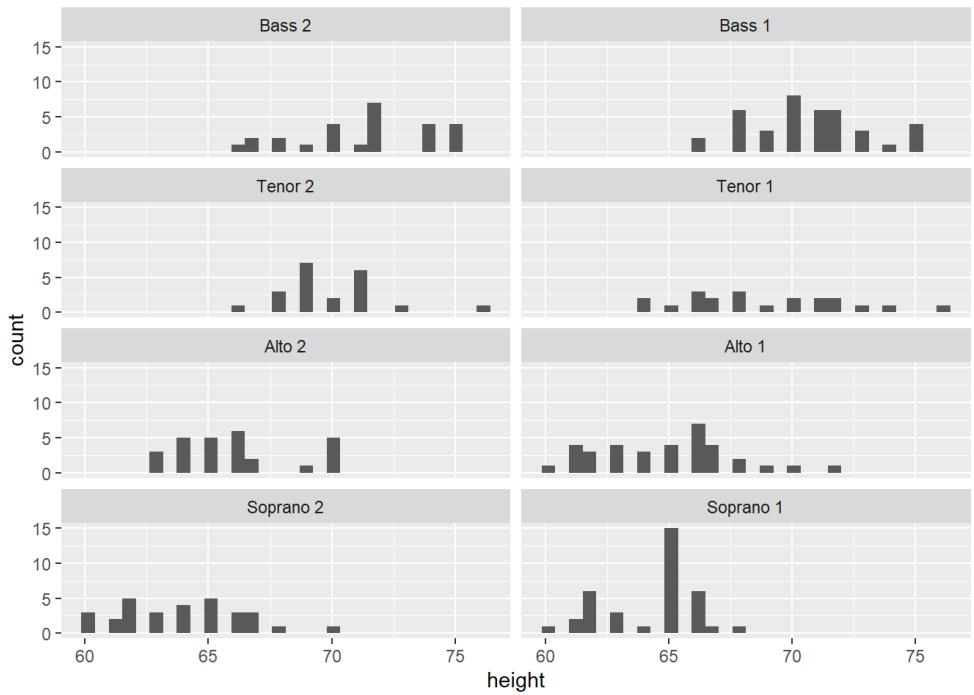
```
ggplot(singer, aes(height)) +
geom_histogram() +
facet_wrap(~voice.part,ncol = 2)
```

```
facet_wrap(~voice.part,ncol = 2)
```

Produce the histogram across the factor level.

The factor.

Produce the the plot in two columns.



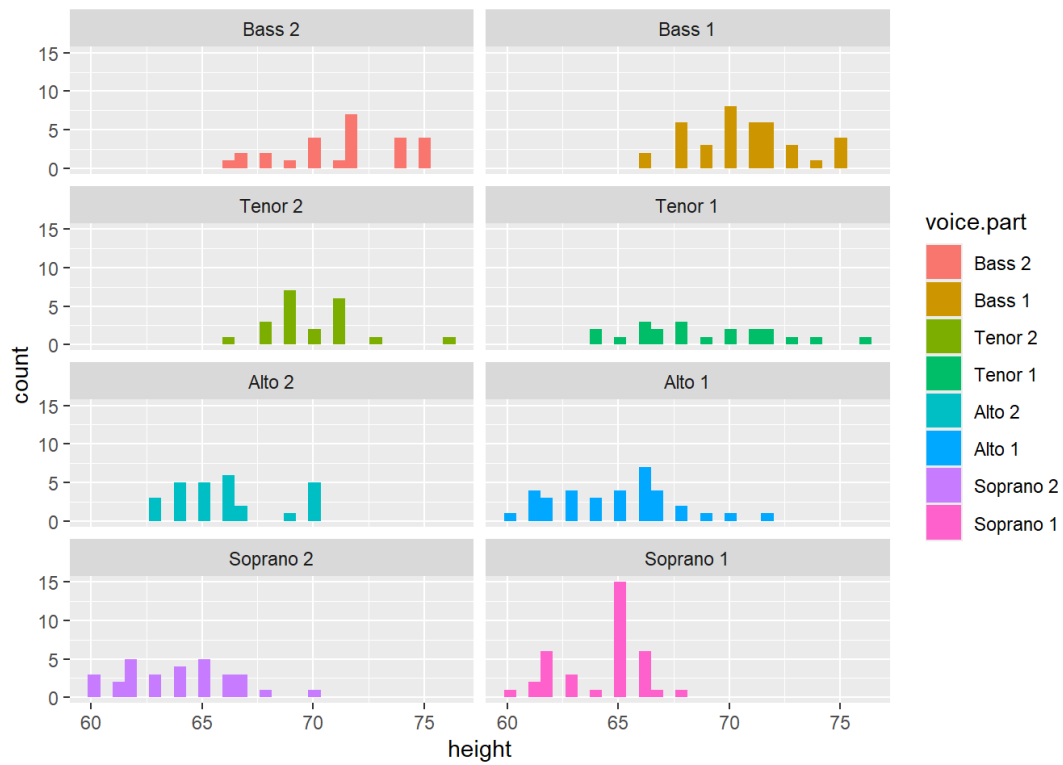
The `singer` dataset: multiway histogram using `ggplot2`

Layer 2: A histogram of the height by voice group, add colors by group.

```
ggplot(singer, aes(height,fill = voice.part)) +
geom_histogram() +
facet_wrap(~voice.part,ncol = 2)
```

```
aes(height,fill = voice.part))
```

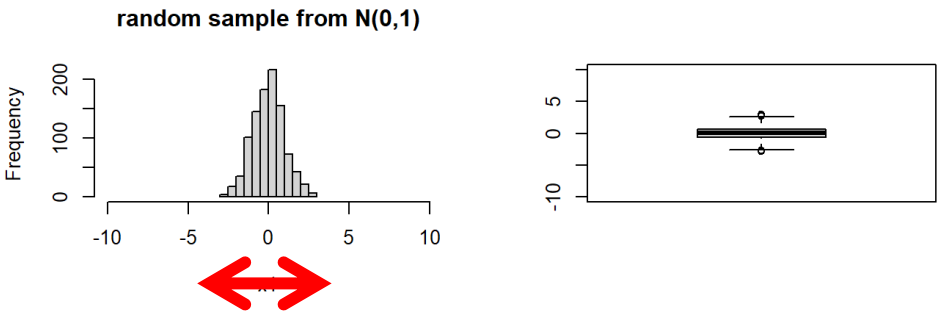
Produce histogram in different colors by the factor level.

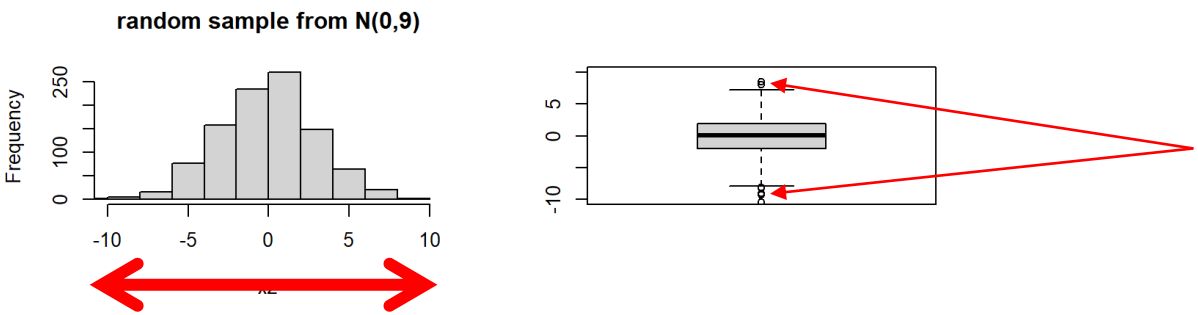


Part 6

Visualization of spread

Boxplot: a graphical display for spread





Two random samples with different spread.

- In the boxplot:
  - The length of the box represents the inter quartile range (variability in the center).
  - Outliers observations.

R code for the example

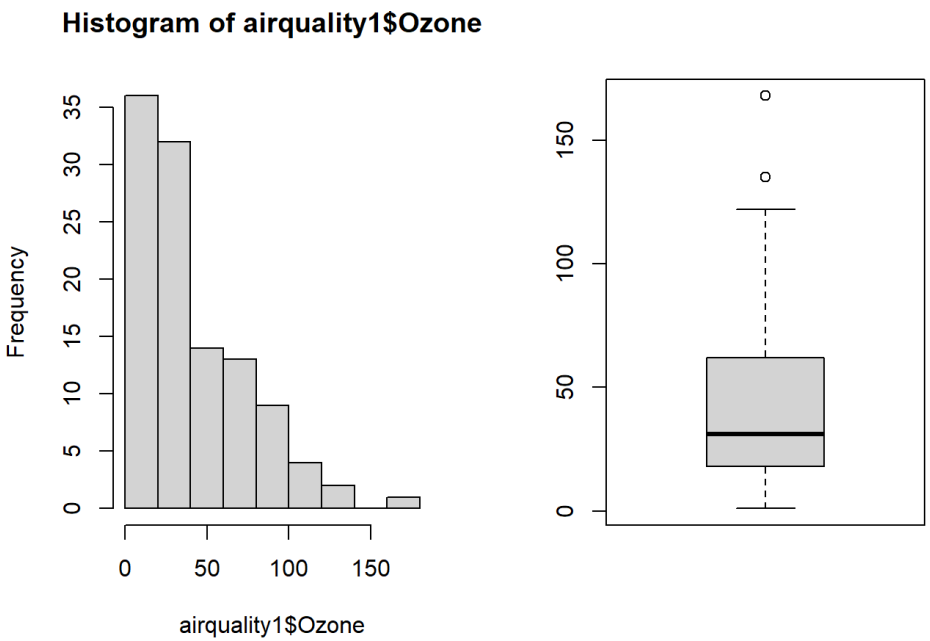
```
x1<-rnorm(1000,0,1)
par(mfrow=c(2,2))
hist(x1,main="random sample from N(0,1)",xlim=c(-10,10))
boxplot(x1,ylim=c(-10,10))
x2<-rnorm(1000,0,3)
hist(x2,main="random sample from N(0,9)",xlim=c(-10,10))
boxplot(x2,ylim=c(-10,10))
```

Boxplot: a graphical display for spread

- Boxplot for the `Ozone` level in the airquality dataset.

- Information in the boxplot:
  - Shape of the distribution (skewed to the right).

  - Outliers.




```
par(mfrow=c(1,2))
airquality1<na.omit(airquality)hist(airquality1$Ozone)
boxplot(airquality1$Ozone)
```

Example 6.1

The `singers` dataset

Heights of singers

The `singer` dataset: boxplot using `lattice`

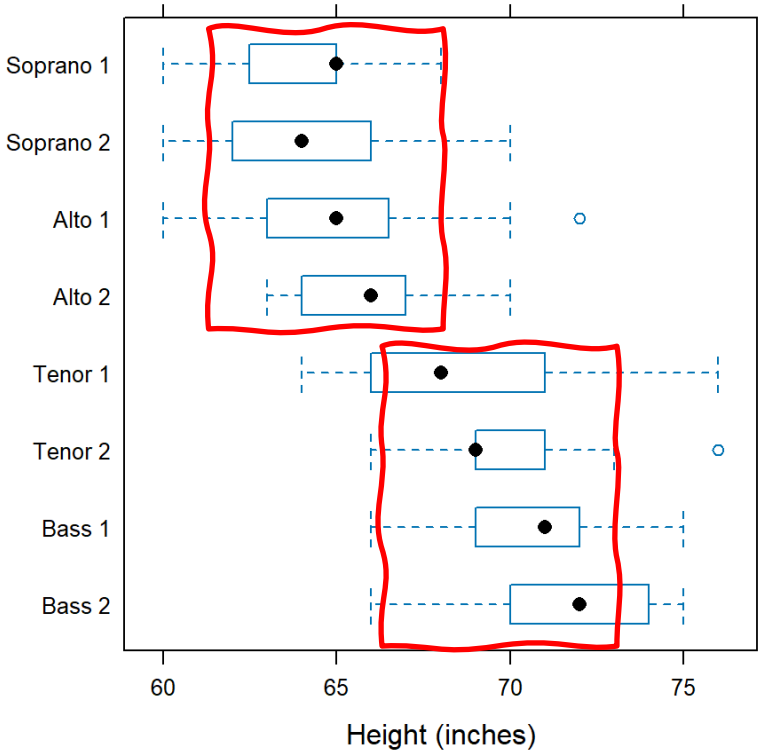
```
bwplot(as.factor(singer$voice.part)~ singer$height,
        data=singer,
         aspect=1,
         xlab="Height (inches)”)
```

The center is represented by the median.

Variability: the length of the box.

Outliers.

Male vs. Female: shift in location.



The `singer` dataset: boxplot using `ggplot2`

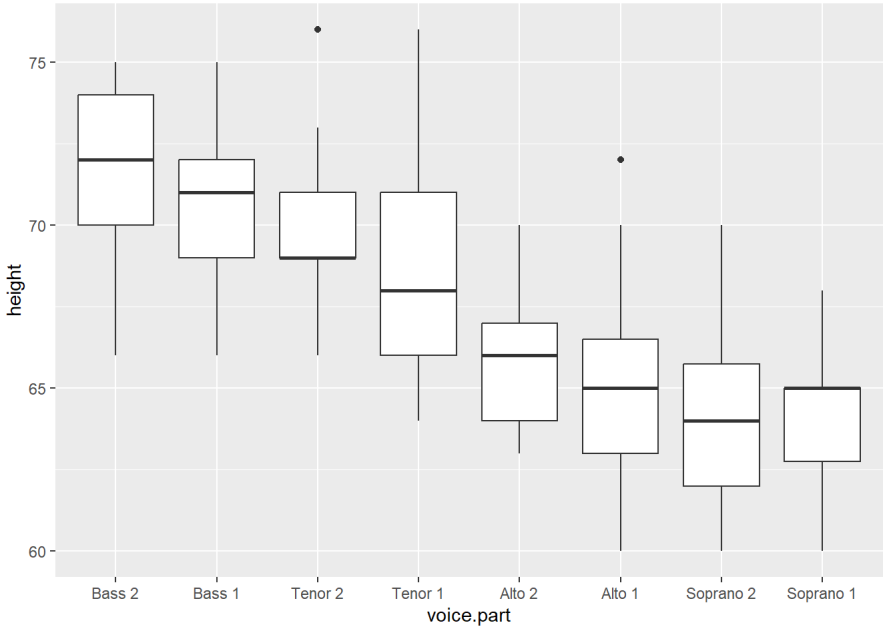
Layer 1: Basic boxplot of height by voice group.

```
ggplot(singer, aes(voice.part,height)) +
geom_boxplot()
```

`aes(voice.part,height)`

X: factor

`geom_boxplot()` Produce a boxplot (by group).



The `singer` dataset: boxplot using `lggplot2`

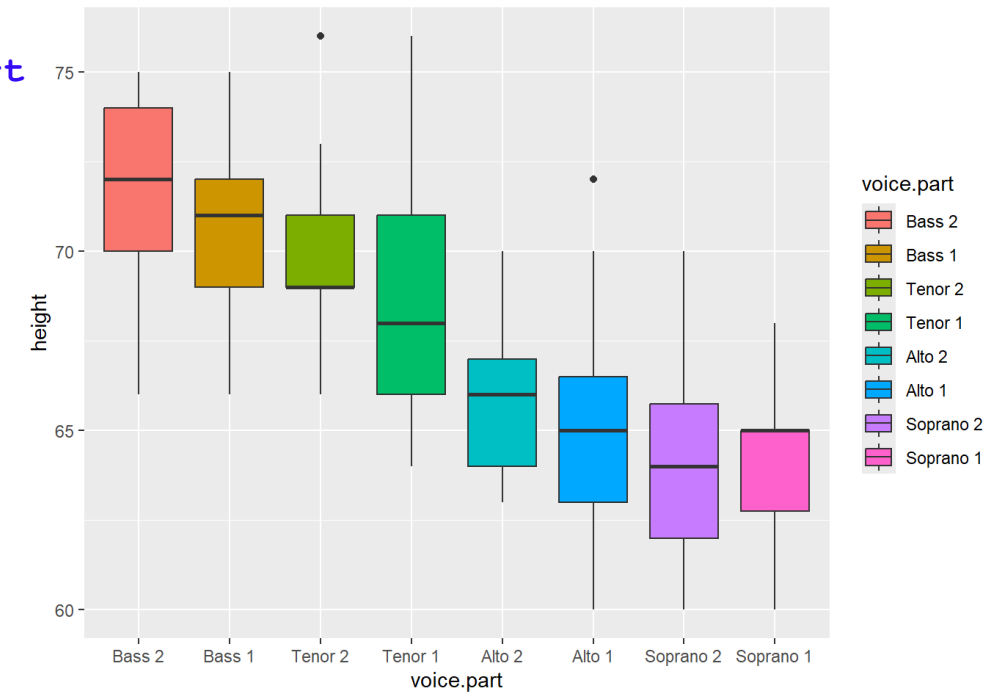
Layer 2: A Boxplot of heights by voice group in different colors.

```
ggplot(singer, aes(voice.part,height,fill=voice.part)) +
geom_boxplot()
```

```
aes(voice.part,height,fill=voice.part
```

Use different color by the factor level.

The factor



The `singer` dataset: boxplot using `ggplot2`

New plot: A Boxplot of heights by voice group with jitter points.

```
qplot(voice.part, height, data = singer, geom = c("boxplot", "jitter"))
```

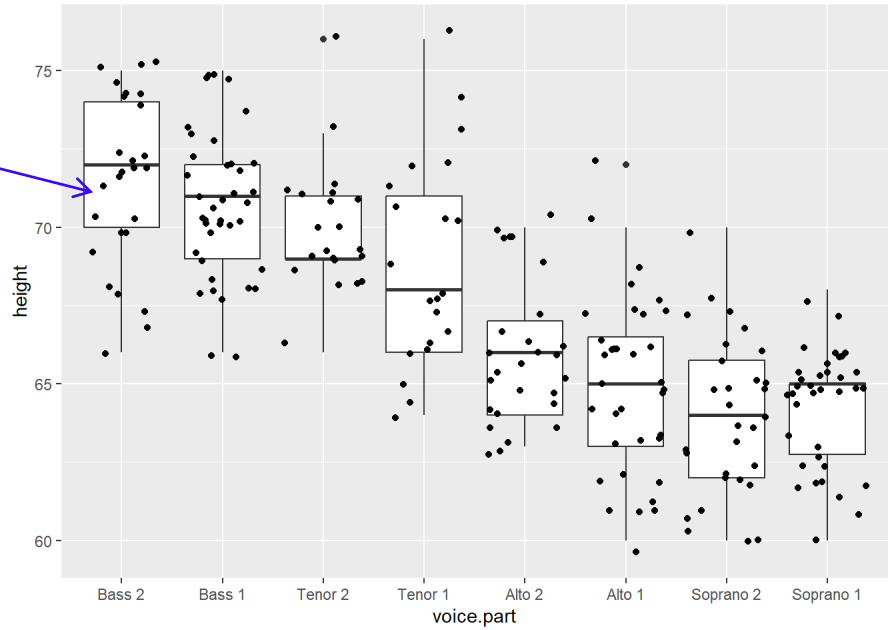
X:factor Y:numeric

```
geom = c("boxplot", "jitter"))
```

Adding the points on the box

Produce a boxplot with jitter.

Indication about the variability in the center of the distribution (the box length).



The `singer` dataset: violin plot using `ggplot2`

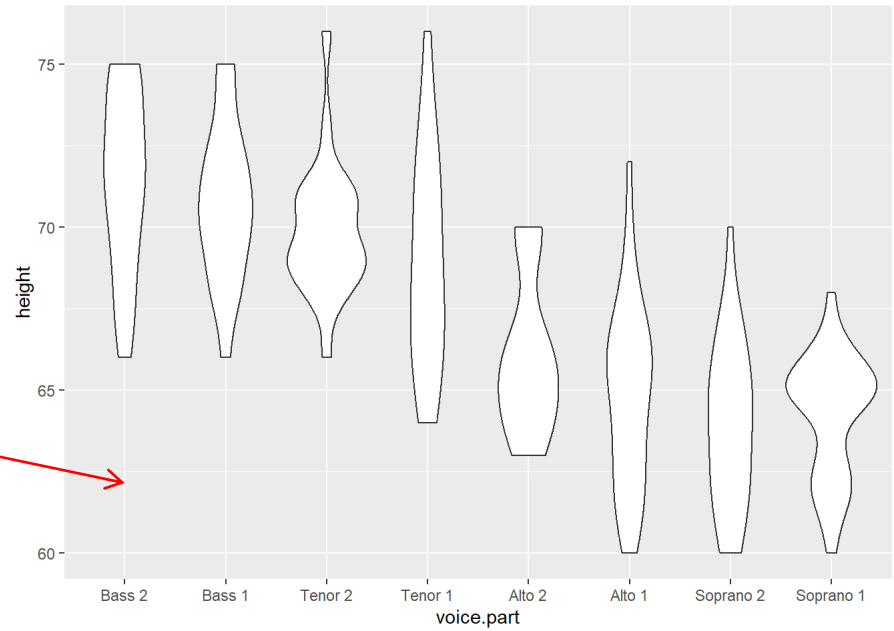
Layer 1: Basic violin plot.

```
ggplot(singer, aes(voice.part,height)) +
geom_violin()
```

```
geom_violin()
```

Produce a violin plot.

Default background color.



The `singer` dataset: violin plot using `ggplot2`

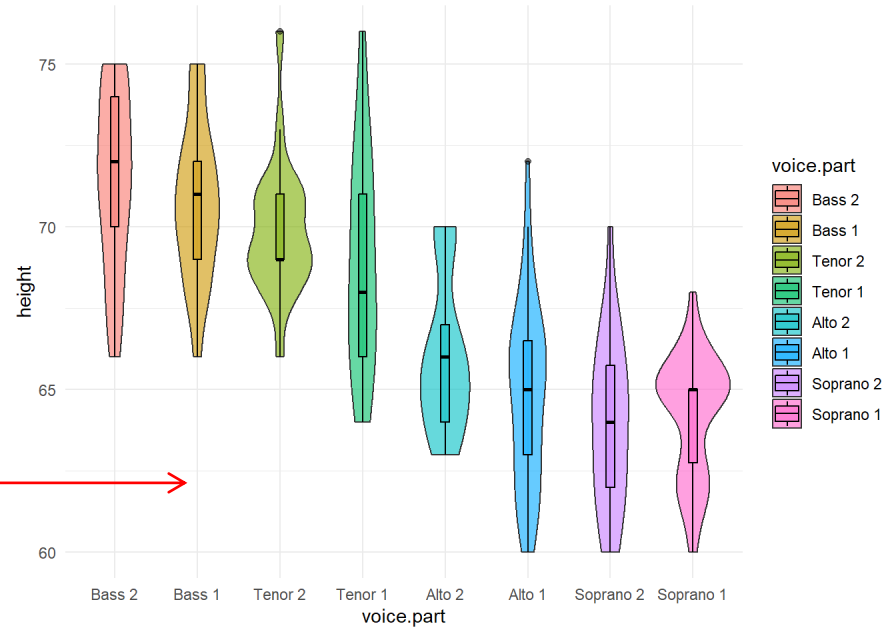
Layer 2: violin plot and a boxplot in the same figure.

```
ggplot(singer, aes(x = voice.part, y = height, fill = voice.part)) +
geom_violin(alpha = 0.6) +
geom_boxplot(width = 0.1, color = "black", alpha = 0.5) +
theme_minimal()
```

```
geom_violin(alpha = 0.6) +
geom_boxplot(width = 0.1,
             color = "black",
             alpha = 0.5) +
```

```
theme_minimal()
```

Use the "minimal" setting to the background.



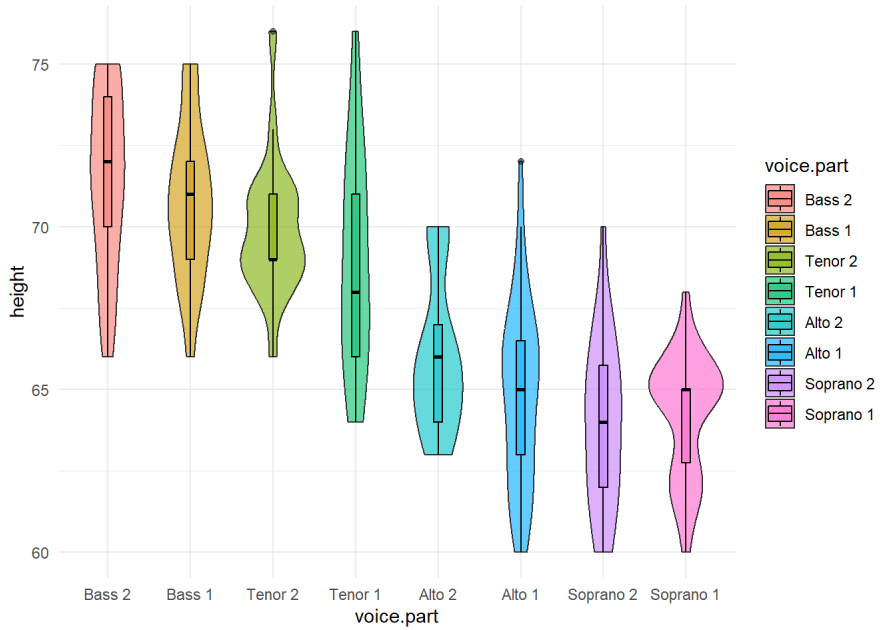
The `singer` dataset: violin plot using `ggplot2`

Layer 2: violin plot and a boxplot in the same figure.

```
ggplot(singer, aes(x = voice.part, y = height, fill = voice.part)) +
geom_violin(alpha = 0.6) +
geom_boxplot(width = 0.1, color = "black", alpha = 0.5) +
theme_minimal()
```

```
geom_violin(alpha = 0.6) +
geom_boxplot(width = 0.1,
             color = "black",
             alpha = 0.5) +
```

`alpha=` The brightness of the area inside the plot.



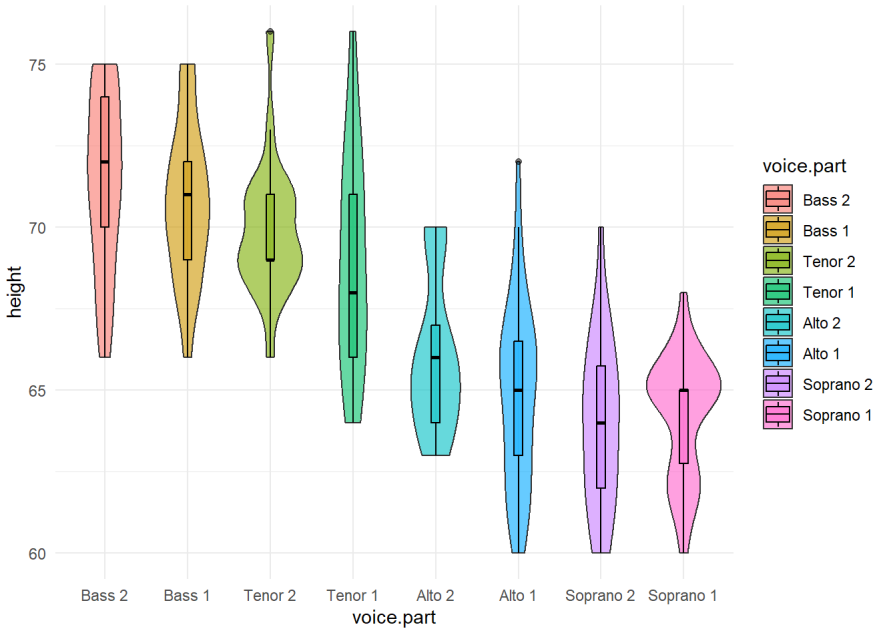
The `singer` dataset: violin plot using `ggplot2`

Layer 2: violin plot and a boxplot in the same figure.

```
ggplot(singer, aes(x = voice.part, y = height, fill = voice.part)) +
geom_violin(alpha = 0.6) +
geom_boxplot(width = 0.1, color = "black", alpha = 0.5) +
theme_minimal()
```

```
ggplot(singer,
aes(x = voice.part,
    y = height,
    fill = voice.part))
```

Use different colors by group.

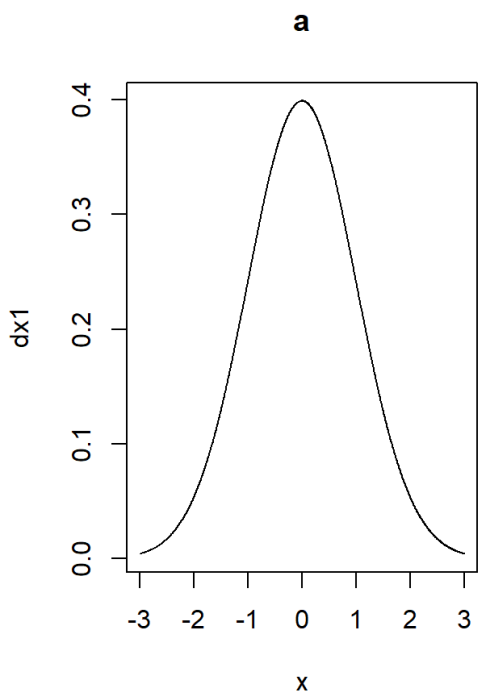


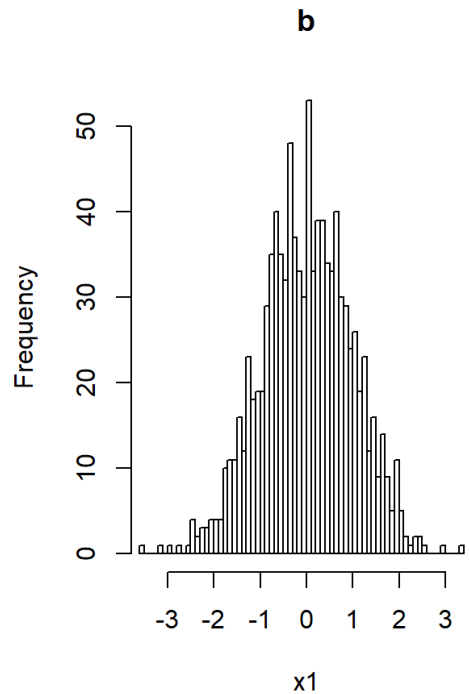
Part 7

Exploring the shape of a distribution

Density estimate

Density (in the population) and histogram

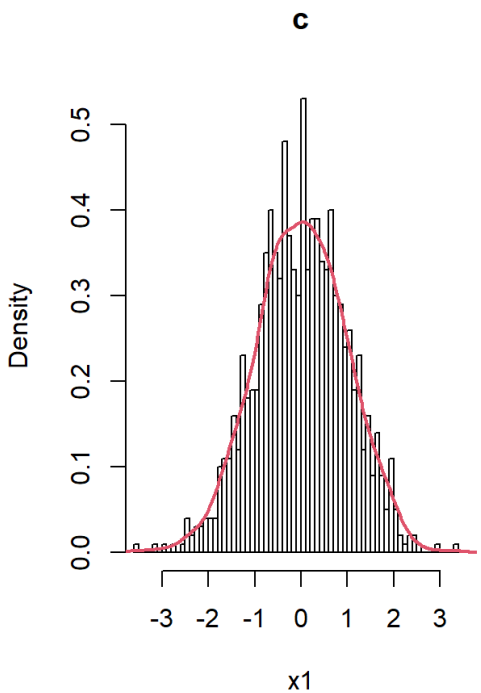




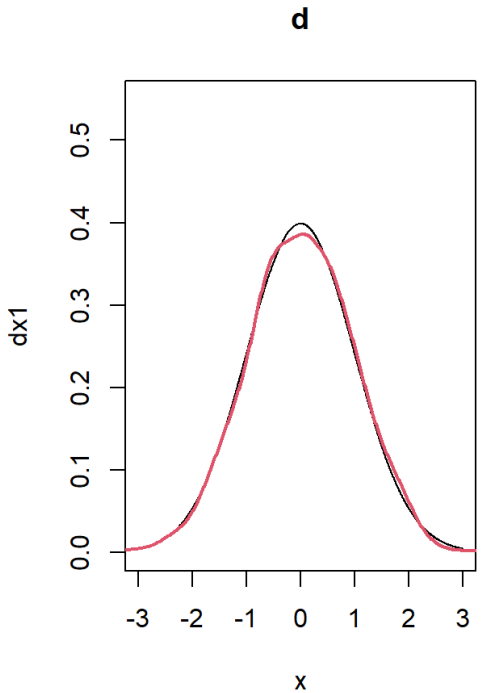
The population

A random sample from the population.

Density estimate



A histogram and a density estimate based on the sample.



The true density (of the population) and density estimate based on the sample.

Example 7.1

The `faithful` dataset

Distribution of eruptions time

The old faithful data

- The data gives information about waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA.
- We focus on two variables:
  - Eruption time in mins.
  - Waiting time to next eruption (in mins).

The `faithful` data

```
head(faithful)
```

```
##   eruptions waiting
## 1     3.600      79
## 2     1.800      54
## 3     3.333      74
## 4     2.283      62
## 5     4.533      85
## 6     2.883      55
```

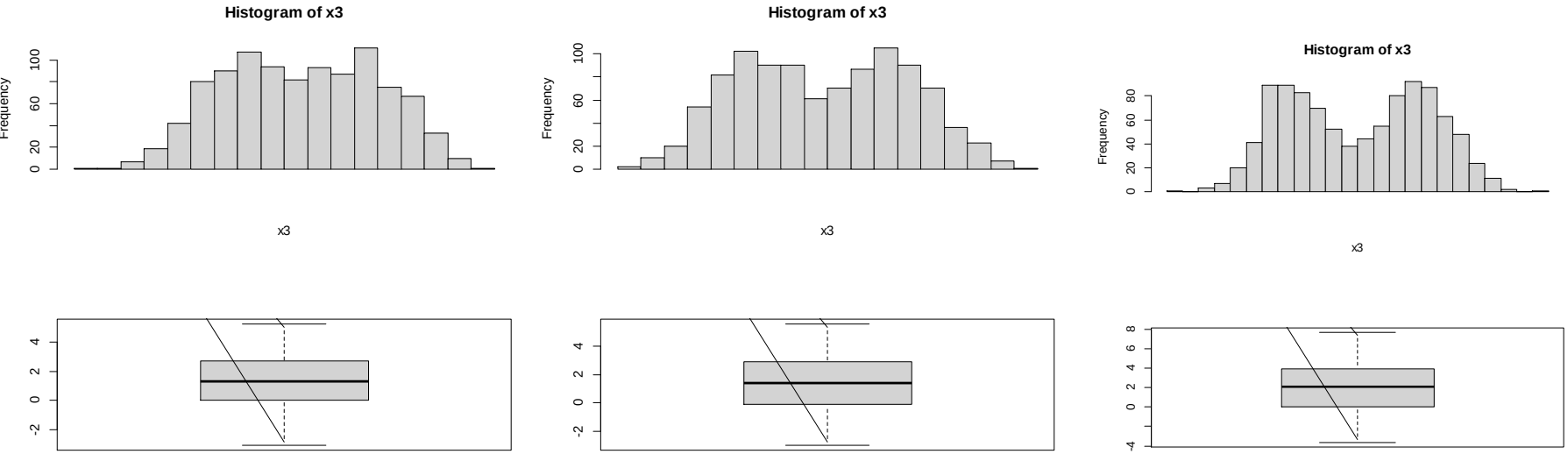
Numerical variable

The `faithful` data

```
 [1]  3.600 1.800 3.333 2.283 4.533 2.883 4.700 3.600 1.950 4.350 1.833 3.917
 [13] 4.200 1.750 4.700 2.167 1.750 4.800 1.600 4.250 1.800 1.750 3.450 3.067
 [25] 4.533 3.600 1.967 4.083 3.850 4.433 4.300 4.467 3.367 4.033 3.833 2.017
 [37] 1.867 4.833 1.833 4.783 4.350 1.883 4.567 1.750 4.533 3.317 3.833 2.100
 [49] 4.633 2.000 4.800 4.716 1.833 4.833 1.733 4.883 3.717 1.667 4.567 4.317
 [61] 2.233 4.500 1.750 4.800 1.817 4.400 4.167 4.700 2.067 4.700 4.033 1.967
 [73] 4.500 4.000 1.983 5.067 2.017 4.567 3.883 3.600 4.133 4.333 4.100 2.633
 [85] 4.067 4.933 3.950 4.517 2.167 4.000 2.200 4.333 1.867 4.817 1.833 4.300
 [97] 4.667 3.750 1.867 4.900 2.483 4.367 2.100 4.500 4.050 1.867 4.700 1.783
[109] 4.850 3.683 4.733 2.300 4.900 4.417 1.700 4.633 2.317 4.600 1.817 4.417
[121] 2.617 4.067 4.250 1.967 4.600 3.767 1.917 4.500 2.267 4.650 1.867 4.167
[133] 2.800 4.333 1.833 4.383 1.883 4.933 2.033 3.733 4.233 2.233 4.533 4.817
[145] 4.333 1.983 4.633 2.017 5.100 1.800 5.033 4.000 2.400 4.600 3.567 4.000
[157] 4.500 4.083 1.800 3.967 2.200 4.150 2.000 3.833 3.500 4.583 2.367 5.000
[169] 1.933 4.617 1.917 2.083 4.583 3.333 4.167 4.333 4.500 2.417 4.000 4.167
[181] 1.883 4.583 4.250 3.767 2.033 4.433 4.083 1.833 4.417 2.183 4.800 1.833
[193] 4.800 4.100 3.966 4.233 3.500 4.366 2.250 4.667 2.100 4.350 4.133 1.867
[205] 4.600 1.783 4.367 3.850 1.933 4.500 2.383 4.700 1.867 3.833 3.417 4.233
[217] 2.400 4.800 2.000 4.150 1.867 4.267 1.750 4.483 4.000 4.117 4.083 4.267
[229] 3.917 4.550 4.083 2.417 4.183 2.217 4.450 1.883 1.850 4.283 3.950 2.333
[241] 4.150 2.350 4.933 2.900 4.583 3.833 2.083 4.367 2.133 4.350 2.200 4.450
[253] 3.567 4.500 4.150 3.817 3.917 4.450 2.000 4.283 4.767 4.533 1.850 4.250
[265] 1.983 2.250 4.750 4.117 2.150 4.417 1.817 4.467
```

- How does the distribution of `eruptions` time look like ?

What do we want to visualize ?



- Example of three samples of 1000 observations.
- Which pattern we see ?
- Which figure is better to explore the shape ?

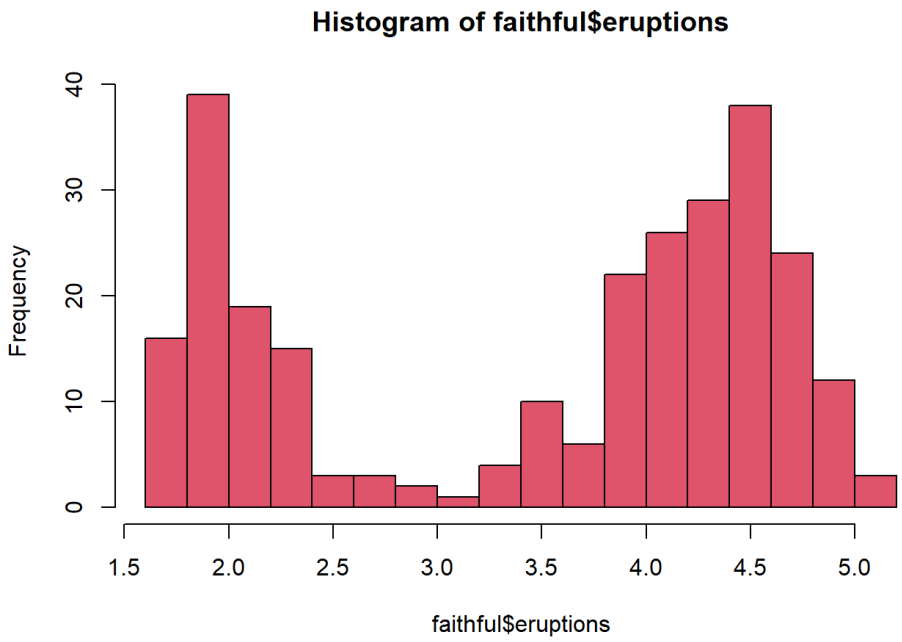
R code for the example

```
x1<-rnorm(500,0,1)
x2<-rnorm(500,2.9,1)
x3<-c(x1,x2)
par(mfrow=c(2,1))
hist(x3,nclass=25,xaxt="n",yxat="n")
boxplot(x3)
```

The `faithful` data: histogram of the eruptions time (basic plot)

Basic histogram in R.

```
hist(faithful$eruptions,nclass=20,col=2)
```



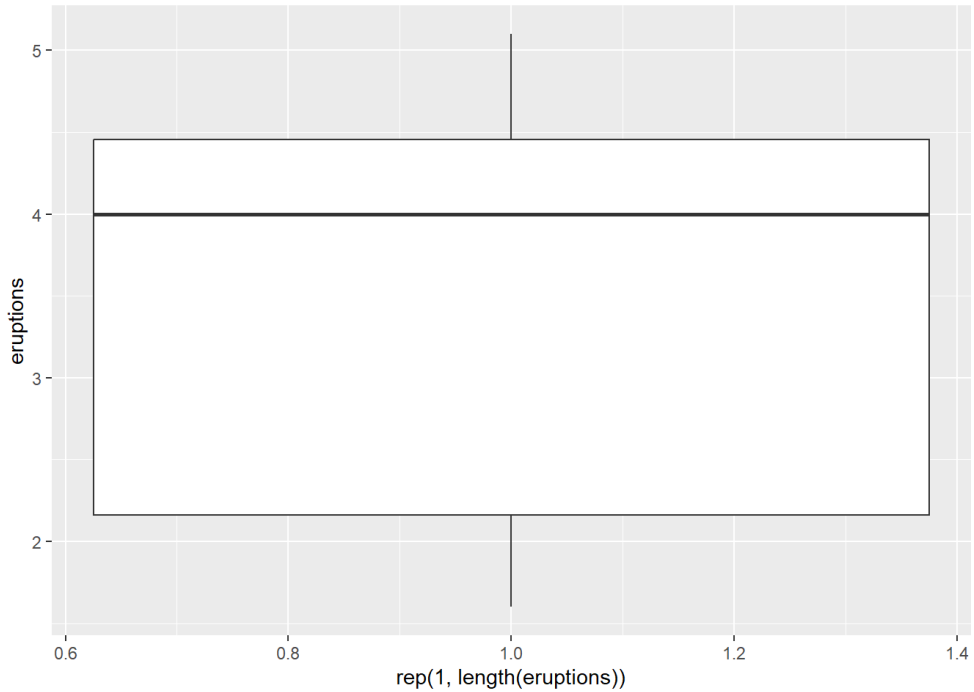
The `faithful` data: boxplot of the eruptions time (`ggplot2`)

Layer 1: Basic boxplot.

```
qplot(rep(1,length(eruptions)),eruptions, data=faithful,
geom = c("boxplot"))
```

```
geom = c("boxplot"))
```

Do we see the pattern in the data ?



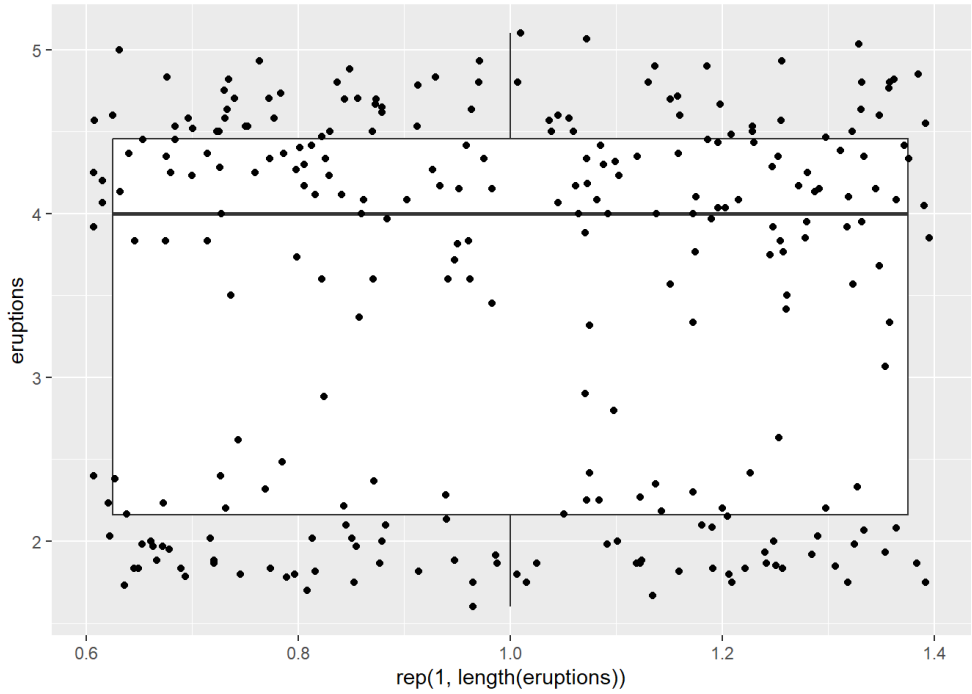
The `faithful` data: boxplot of the eruptions time (`ggplot2`)

Layer 2: Basic boxplot + jitter points.

```
qplot(rep(1,length(eruptions)),eruptions, data=faithful,
geom = c("boxplot", "jitter"))
```

```
geom = c("boxplot", "jitter"))
```

Do we see the pattern in the data ?

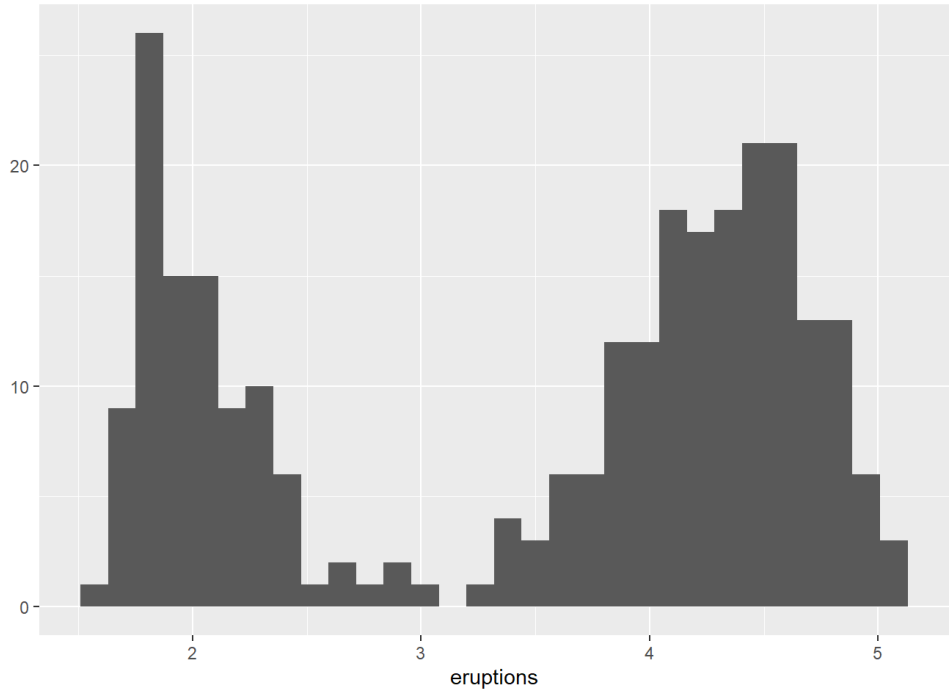


The `faithful` data: histogram of the eruptions time (`ggplot2`)

```
qplot(eruptions, data=faithful, geom="histogram")
```

`geom="histogram"`

Produce a histogram.

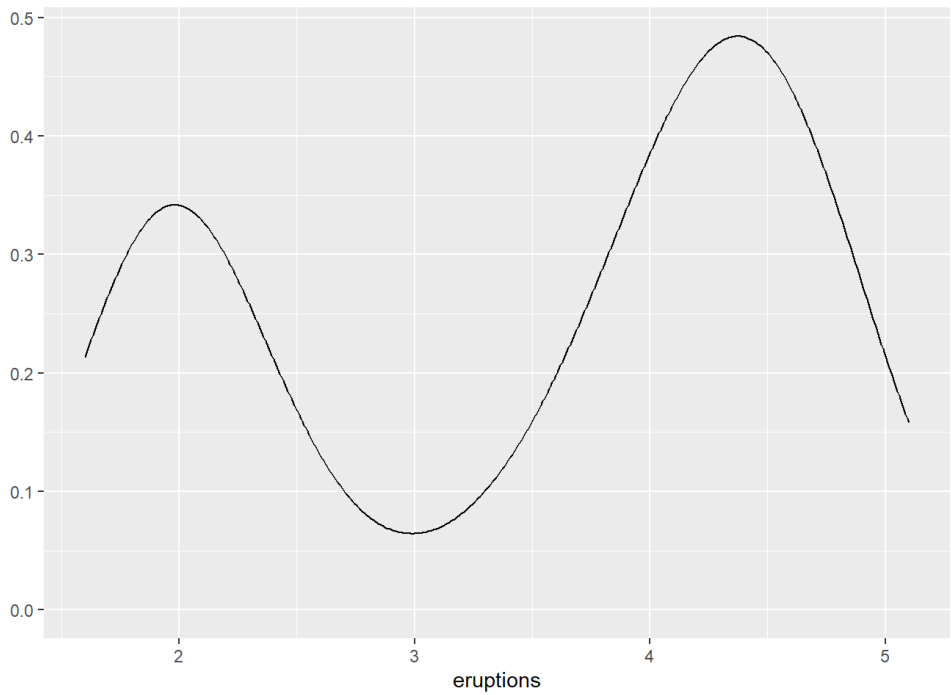


The `faithful` data: density plot of the eruptions time (`ggplot2`)

```
qplot(eruptions, data=faithful, geom="density")
```

geom="density"

Produce a density.



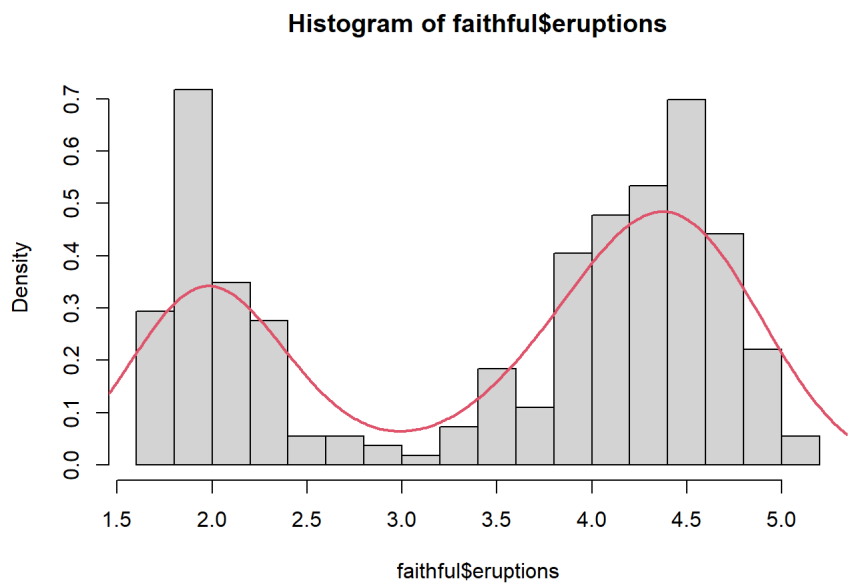
The `faithful` data: density plot and histogram of the eruptions time (basic plot)

```
hist(faithful$eruptions,nclass=15,probability = TRUE)
dx<-density(faithful$eruptions)
lines(dx$x,dx$y,lwd=2,col=2)
```

Produce a histogram and density plot on the same figure.

Basic R plot.

`ggplot2` ?



Example 7.2

The `singers` dataset

Heights of singers

The `singer` dataset: dotplot using `ggplot2`

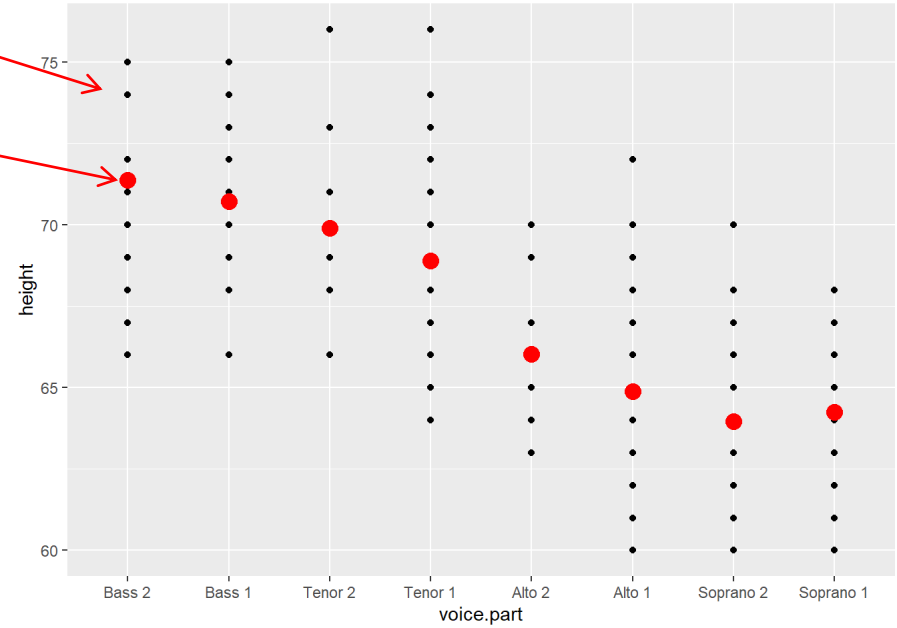
Layer 2: dadding the information about the groups means to the dotplot.

```
ggplot(singer, aes(voice.part,height)) +
geom_point() +
stat_summary(geom = "point", fun.y = "mean", colour = "red", size = 4)
```

Produce a dotplot without jitter

```
stat_summary(geom = "point",
             fun.y = "mean",
             colour = "red", size = 4)
```

Calculate the group mean.



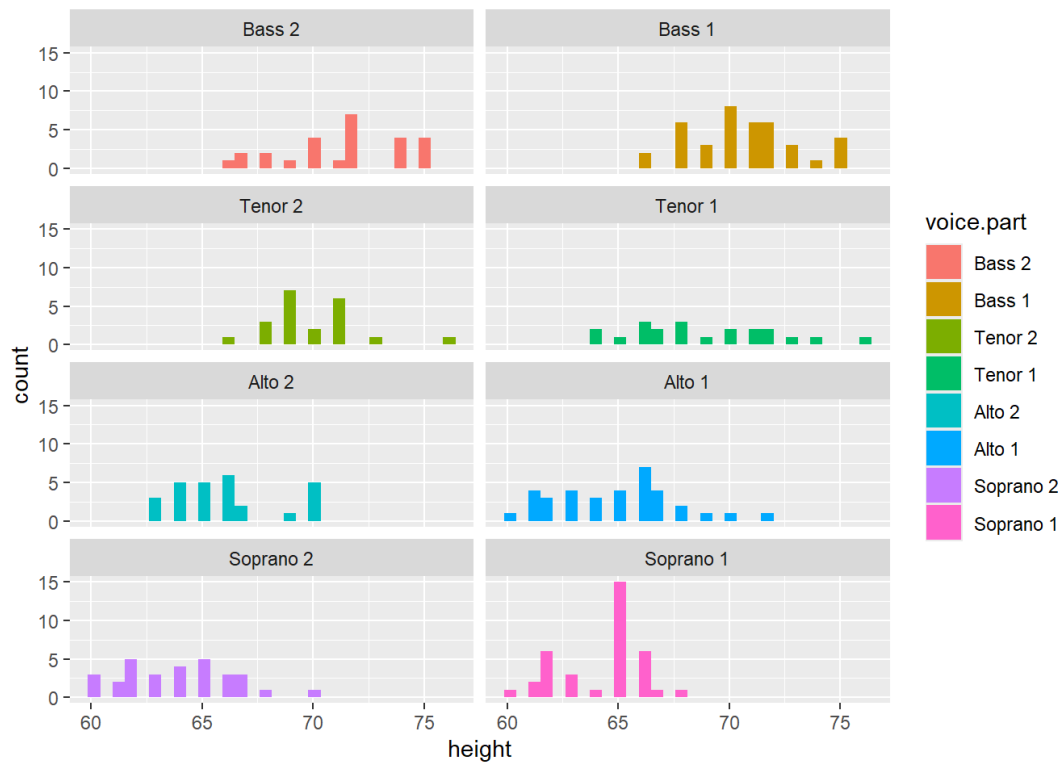
The `singer` dataset: multiway histogram using `ggplot2`

Layer 2: A histogram of the height by voice group, add colors by group.

```
ggplot(singer, aes(height,fill = voice.part)) +
geom_histogram() +
facet_wrap(~voice.part,ncol = 2)
```

```
aes(height,fill = voice.part))
```

Produce histogram in different colors by the factor level.



The `singer` dataset: density plot using `ggplot2`

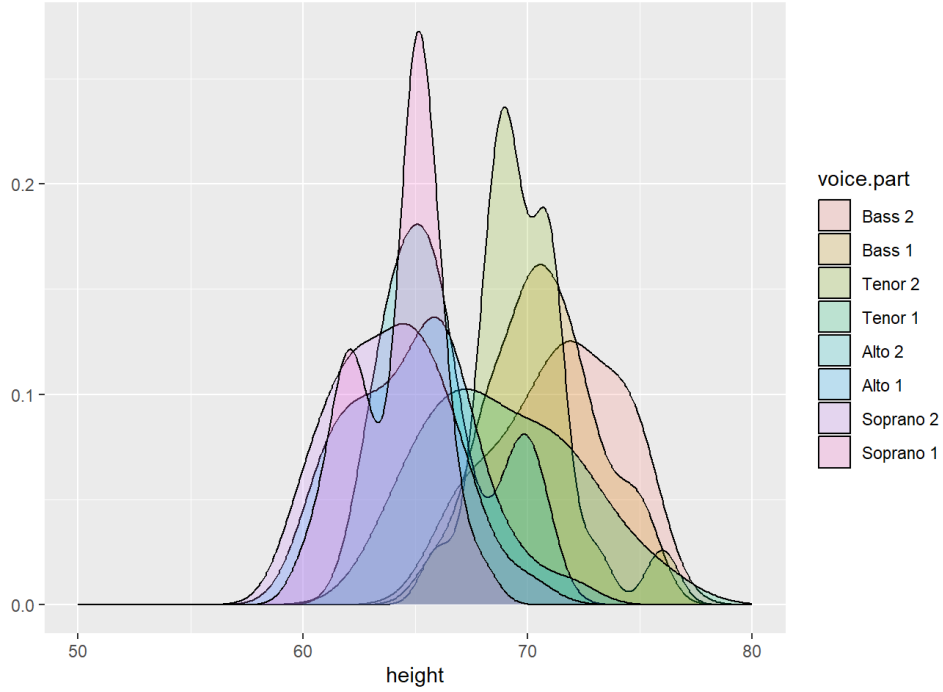
Layer 1: A density plot of the height by voice group, add colors by group.

```
qplot(height, data=singer, geom="density",
      xlim = c(50,80),
      fill = voice.part, alpha = I(0.2))
```

Different color by group

Main problem: the figure is too "crowded".

Difficult to see patterns in the data.



The `singer` dataset: density plot using `ggplot2 + ggridges`

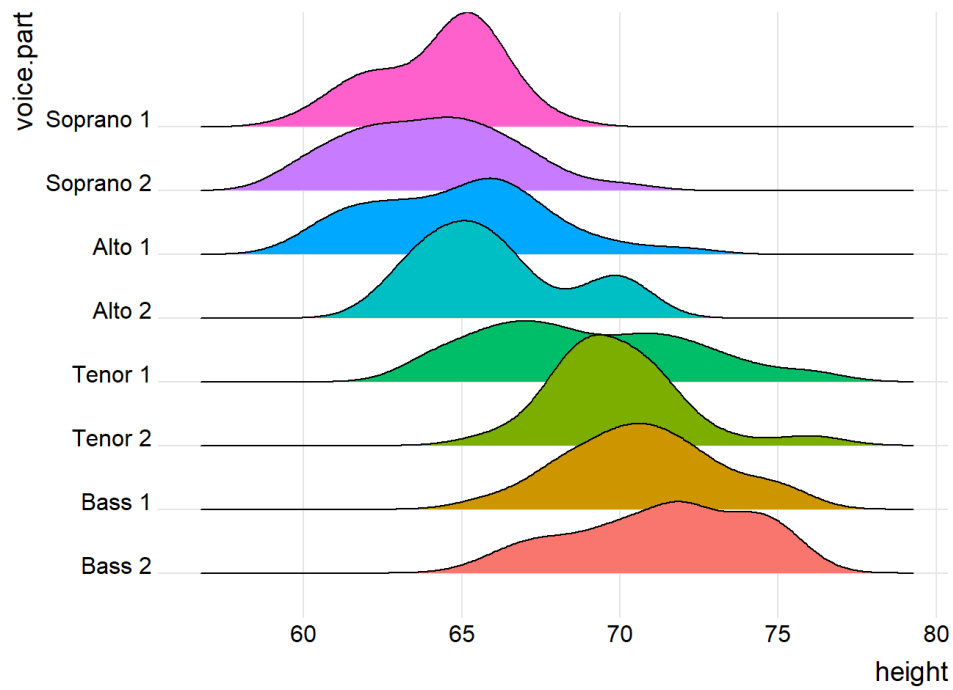
```
library(ggridges)
ggplot(singer, aes(x=height,y=voice.part,fill = voice.part)) +
geom_density_ridges() +
theme_ridges() +
theme(legend.position = "none")
```

```
library(ggridges)

geom_density_ridges()
theme_ridges()
```

Produce a density plot using the package `ggridges`.

Clear visualization of the data.



Part 8

Exploring the shape of a distribution using a qq normal plot

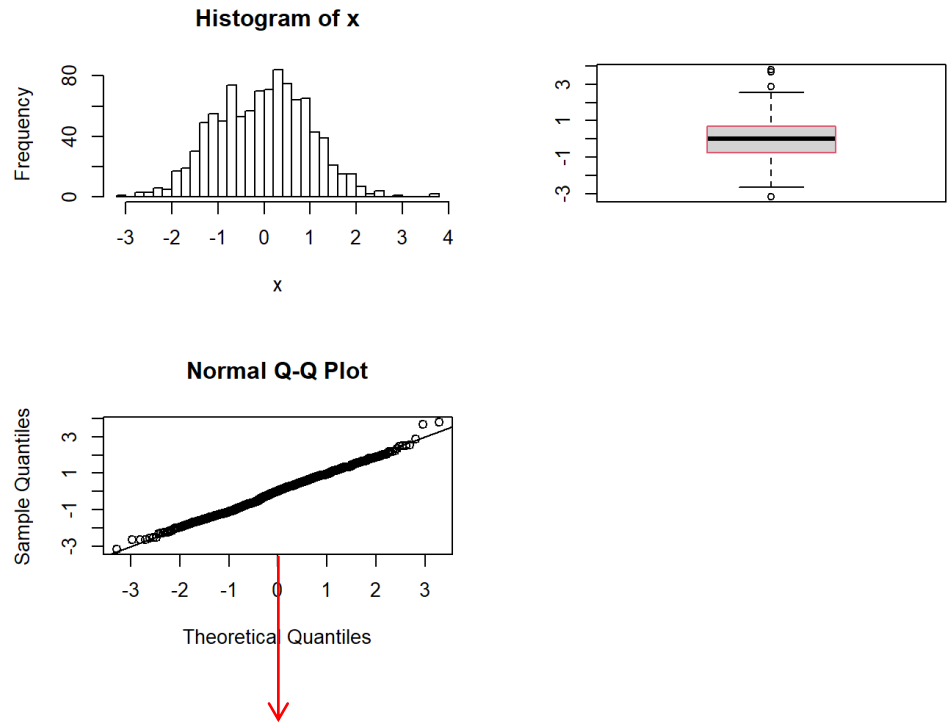
Normal probability plot

- How close our sample is to a normal distribution ?
- Symmetry ?
- Location ?
- Variability ?

qq normal plot

$X \sim N(0,1)$

- A sample of 1000 observations.
- Boxplot and qq-normal plot.



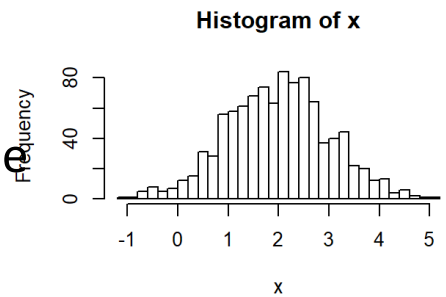
The mean of N(0,1)

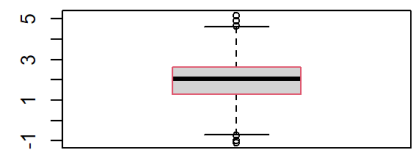
```
x <- rnorm(1000, 0, 1)
par(mfrow = c(2, 2))
hist(x, nclass = 25, col = 0)
boxplot(x, boxcol = 2, medcol = 1)
qqnorm(x)
abline(0, 1)
```

qq normal plot

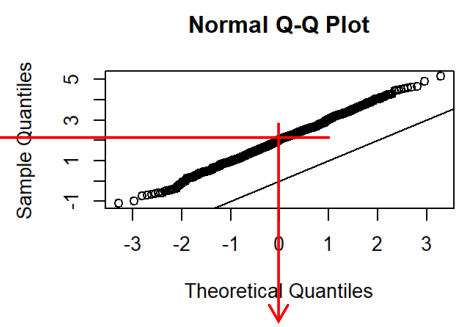
$X \sim N(2,1)$

- The data are symmetric around the mean.





The mean of the data



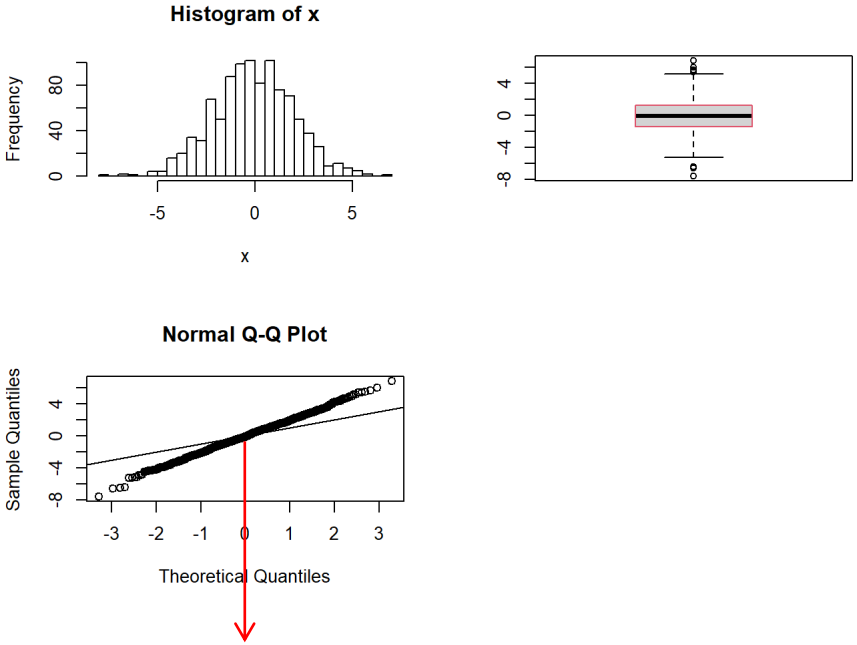
The data are parallel to the 45° line.

The mean of N(0,1)

qq normal plot

$X \sim N(0,2)$

- A sample of 1000 observations.
- The mean is the same as N(0,1) but higher variability.

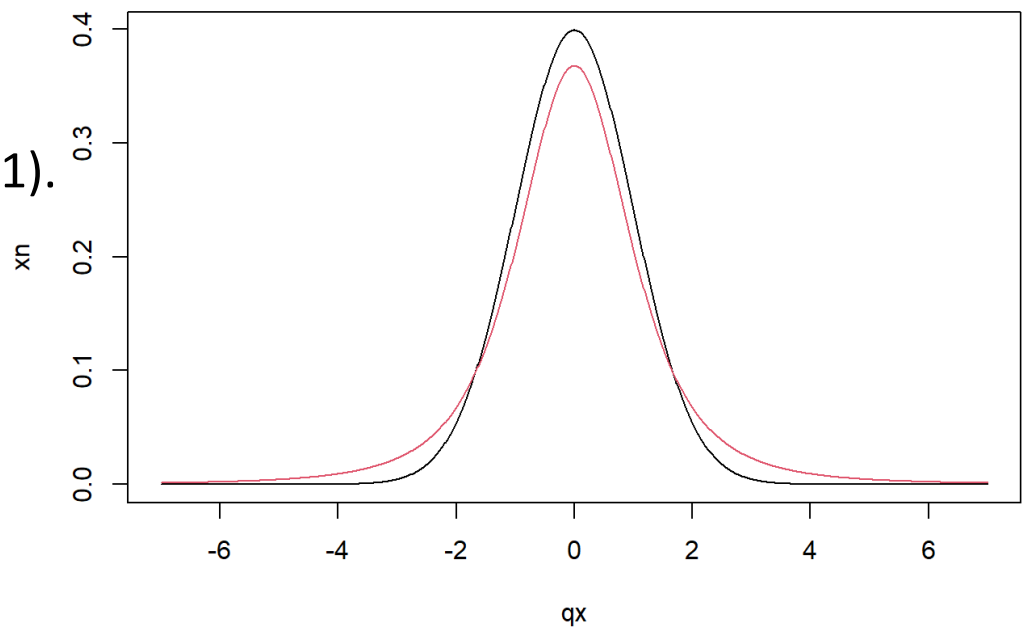


The mean of N(0,1)

qq normal plot

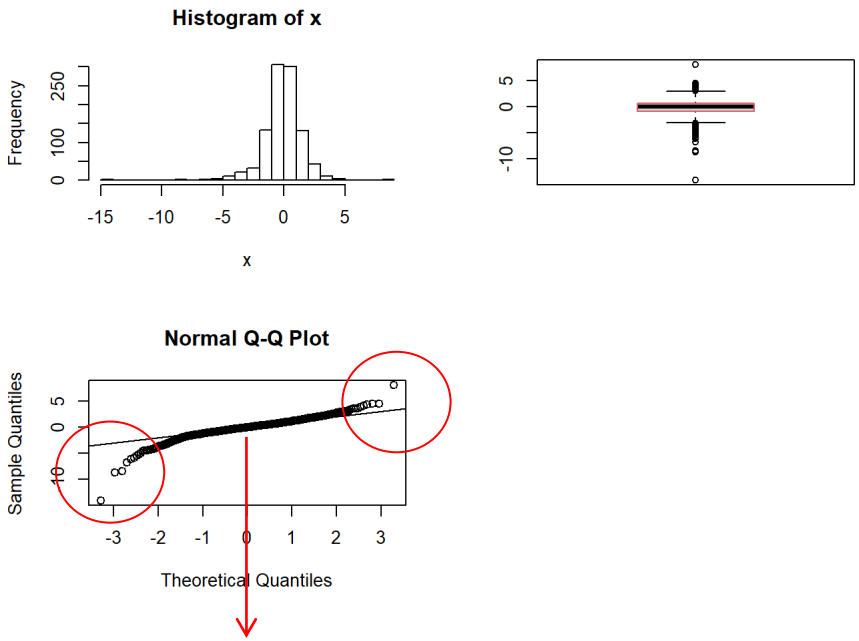
$X \sim t_{(3)}$

- Longer tails compared with N(0,1).



qq normal plot

- A sample of 1000 observations.
- The mean is the same as N(0,1) but higher variability.
- Boxplot: outliers.
- qqnormal plot: the same center but longer tails.

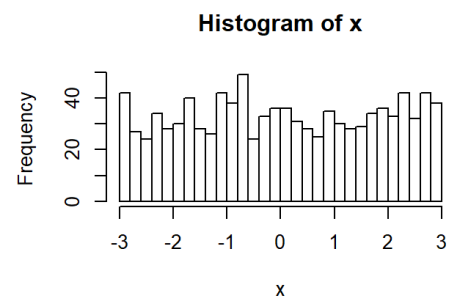


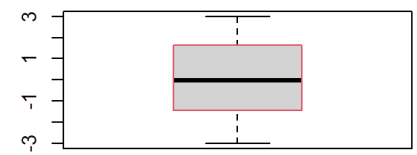
The mean of N(0,1)

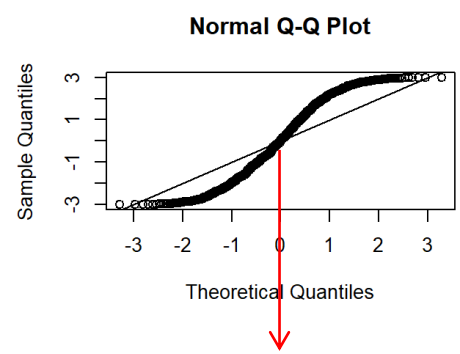
qq normal plot

$X \sim U(-3,3)$

- A sample of 1000 observations.
- The same center as N(0,1) but higher variability.
- Boxplot: no outliers.
- qqnormal plot: the same center but longer tails.







The mean of N(0,1)

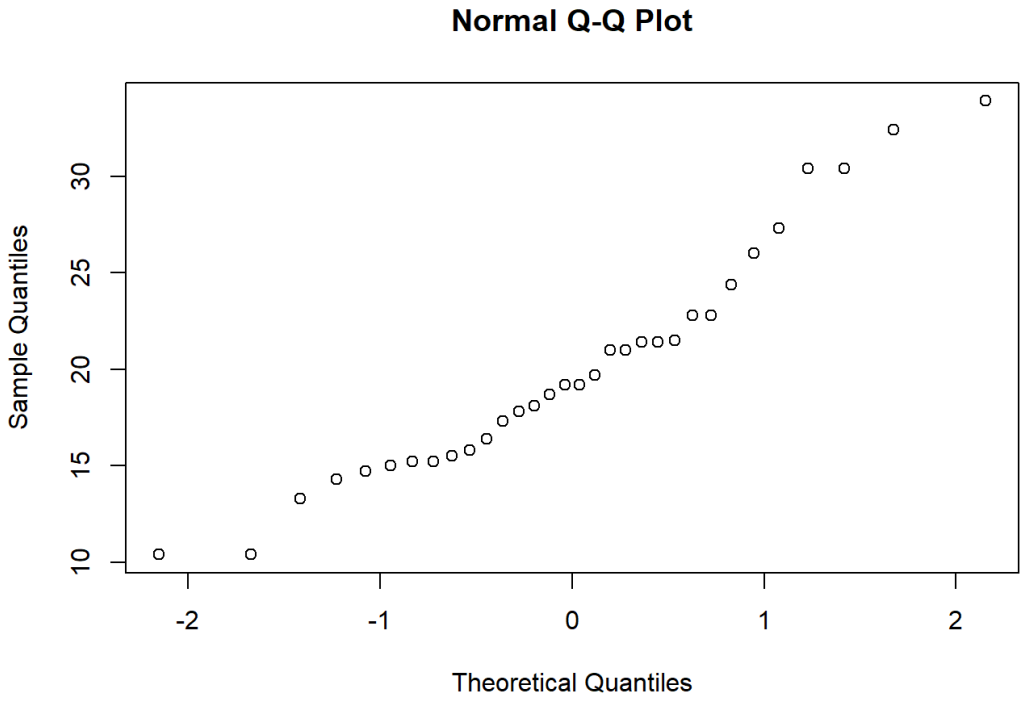
Example 8.1

The `mtcars` dataset

Distribution of mpg

qq-normal plot for `mpg`

```
qqnorm(mtcars$mpg)
```

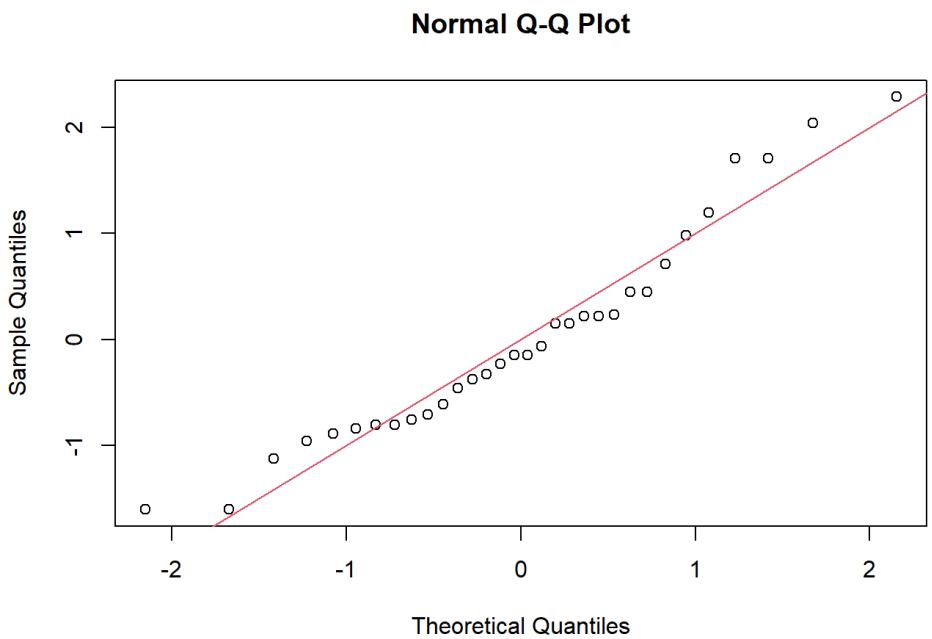


qq-normal plot for the z scores of `mpg`

We define a standardized variable:

$$Z_i = \frac{X_i - \bar{X}}{SD_X}$$

```
m.mpg<-mean(mtcars$mpg)
sd.mpg<-sqrt(var(mtcars$mpg))
z<-(mtcars$mpg-m.mpg)/sd.mpg
qqqnorm(z)
abline(0,1,col=2)
```

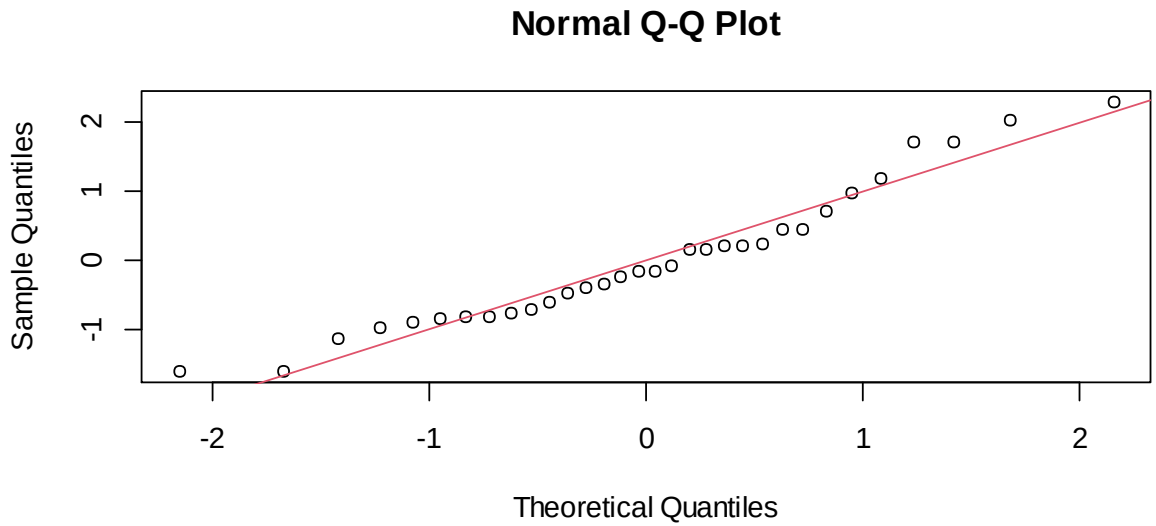


Close to a standard normal distribution ?

qq-normal plot for the z scores of `mpg`

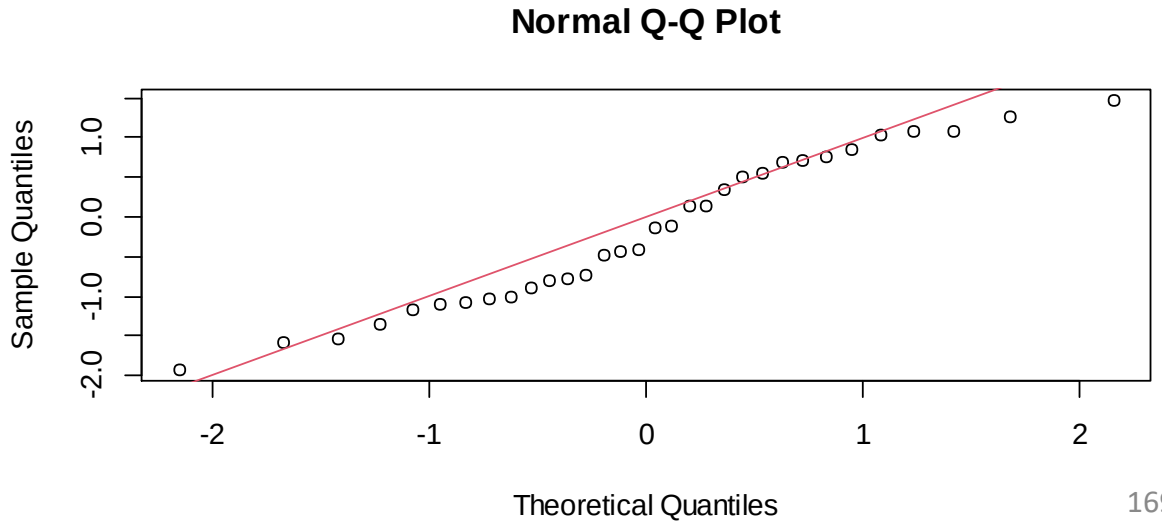
The `mtcars` dataset:

$$Z_i = \frac{X_i - \bar{X}}{SD_X}$$



Random sample, n=32

$$Z_i \sim N(0,1)$$



R code for the example

```
par(mfrow=c(2,1))
m.mpg<-mean(mtcars$mpg)
sd.mpg<-sqrt(var(mtcars$mpg))
z<-(mtcars$mpg-m.mpg)/sd.mpg
qqnorm(z)
abline(0,1,col=2)
length(z)
z1<-rnorm(32,0,1)
qqnorm(z1)
abline(0,1,col=2)
```

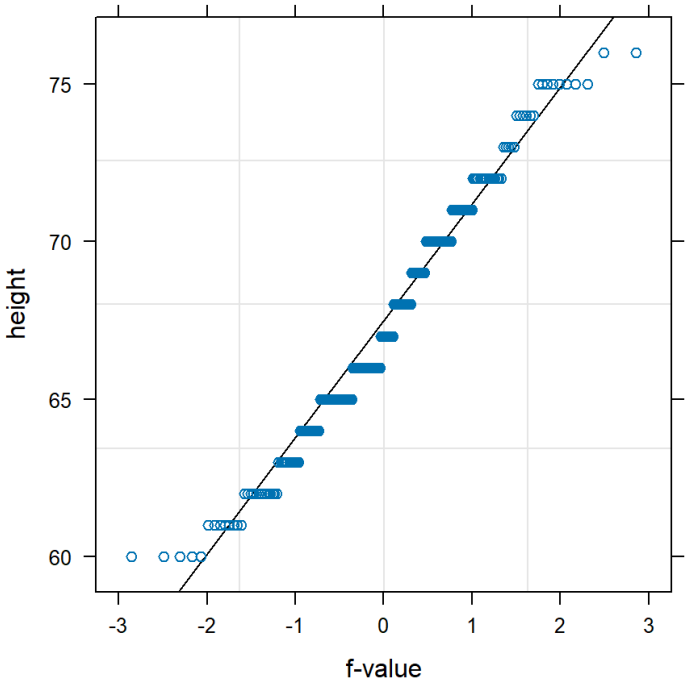
Example 8.2

The `singer` dataset

Distribution of the singers' heights

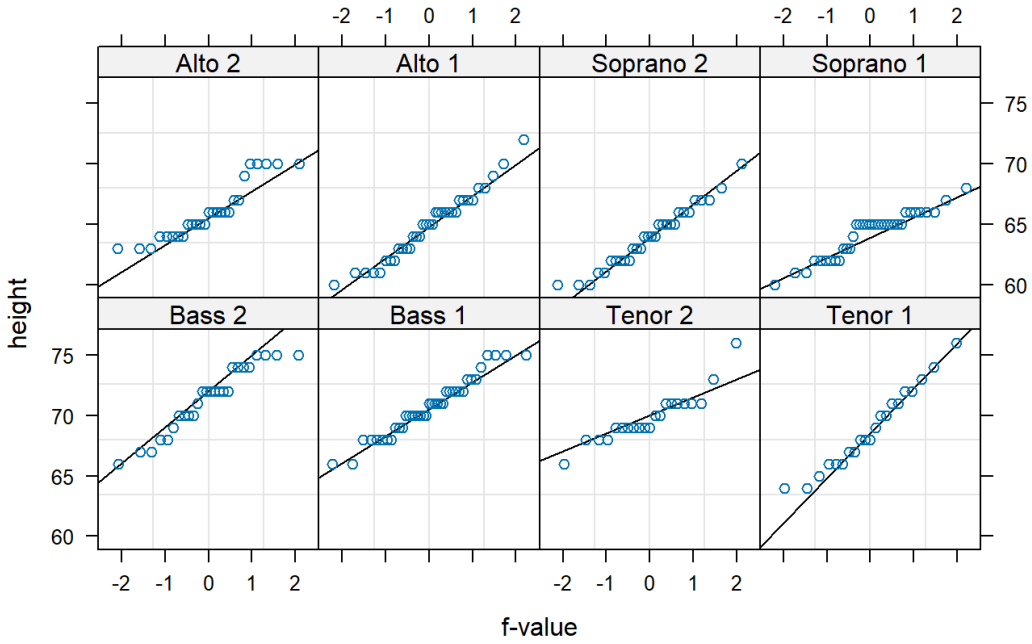
The `singer` dataset: qq-normal plot for the heights (`lattice`)

```
qqmath(~ height,
       disistribution = qnorm,
       data=singer,
       layout=c(1,1),
       prepanel = prepanel.qqmathline,
       panel = function(x, ...)
       {panel.grid()
        panel.qqmathline(x, ...)
        panel.qqmath(x, ...)
        },
        aspect=1,
        xlab = "f-value",
        ylab="height")
```



The `singer` dataset: qq-normal plot for the heights (`lattice`)

```
qqmath(~ height | voice.part,
       distribution = qnorm,
       data=singer,
       layout=c(4,2),
        prepanel = prepanel.qqmat
                   hline,
        panel = function(x, ...)
        {
        panel.grid()
        panel.qqmathline(x, ...)
        panel.qqmath(x, ...)
        },
        aspect=1,
        xlab = "f-value",
        ylab="height")
```



Part 9

Visualization of categorical variables in one sample

Example 9.1

The `boston` dataset

The `boston` data

- The `Boston` dataset contains information about various attributes for suburbs in Boston, Massachusetts.

```
library(MASS)
data(Boston)
names(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

The `boston` data

- For the analysis in this part of the course, we use three variables:
  - `age`: proportion of owner-occupied units built prior to 1940.
  - `chas`: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).
  - `crim`: per capita crime rate by town.

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

Crime rate

- The variable `crim` is a numerical variable that gives information about the per capita crime rate by town.
- We define a new categorical variable `crim_cat` by re-coding the variable `crim` into three categories:
  - Crime rate less than 5 (low).
  - Crime rate between 5 and 15 (medium).
  - Crime rate higher than 15 (high).

Categorical crime rate

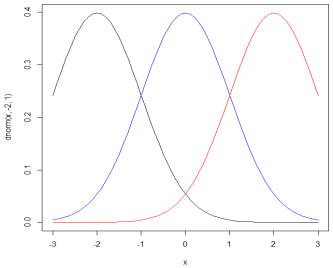
- Distribution of crime rate:

```
##
##     Low Medium   High
##     400     76     30
```

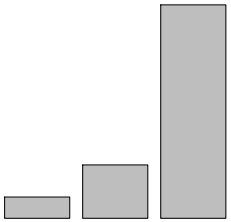
- What do we want to visualize ?
  - Distribution of `age` across the crime rate categories.
  - Distribution of the crime rate categories.
  - Distribution of crime rate by `chas` categories.

Categorical crime rate

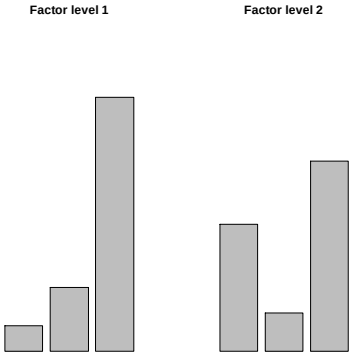
- What do we want to visualize ?
  - Distribution of age across the crime rate categories.
  - Distribution of the crime rate categories.
  - Distribution of crime rate by `chas` categories.



Distribution of a continuous variable across a factor.



Distribution of a categorical variable across a factor.



Distribution of a categorical variable across a factor.

Distribution of `age` by crime categories

Layer 1: distribution of `age` by crime rate.

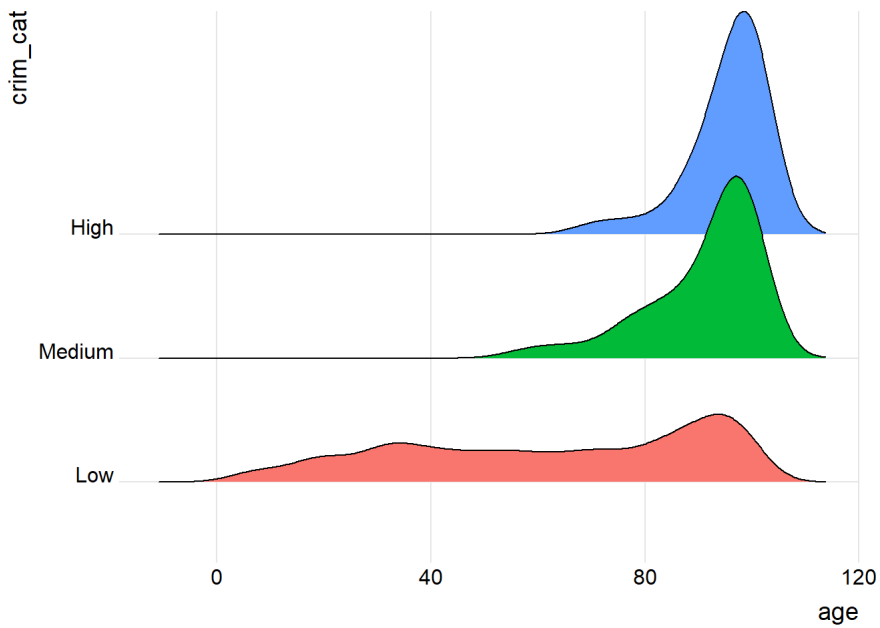
```
library(ggridges)
ggplot(Boston3, aes(x=age,y=crim_cat,fill = crim_cat)) +
  geom_density_ridges() +
  theme_ridges() +
  theme(legend.position = "none")
```

```
library(ggridges)
aes(x=age,y=crim_cat,fill = crim_cat))
```

Color by the crime rate categories.

```
geom_density_ridges()
```

Density plot using the `ggridges` package.



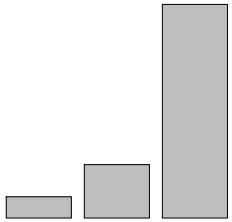
Distribution of crime rate categories

Frequency table for crime rate.

```
counttab=as.data.frame(table(Boston3$crim_cat))
colnames(counttab)=c("Category", "Freq")
counttab
```

```
##   Category Freq
## 1      Low  400
## 2   Medium   76
## 3     High   30
```

What do we want to visualize ?



The distribution of the crime rate.

Distribution of crime rate categories: pie chart (ggplot2)

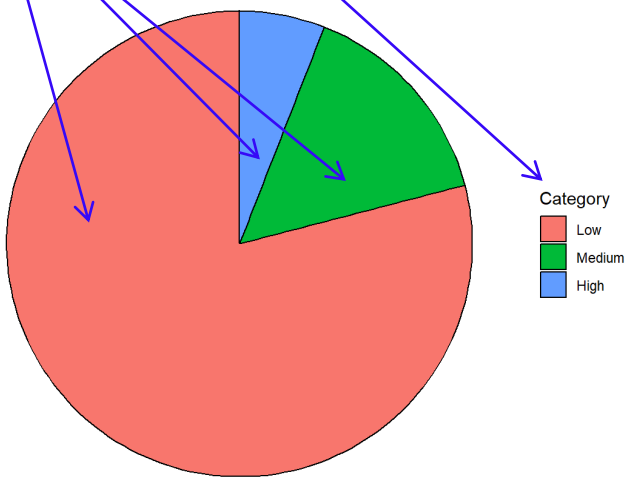
Layer 1: Basic pie chart.

```
plot1=ggplot(counttab, aes(x="", y=Freq, fill=Category)) +
  geom_bar(stat="identity", width=1, color="black") +
  coord_polar("y", start=0)+
  theme_void()
Plot1
```

```
aes(x="", y=Freq, fill=Category))
```

```
geom_bar(stat="identity", width=1,
         color="black")
```

Use the frequency.




```
coord_polar("y", start=0)+
```

Produce a pie chart.

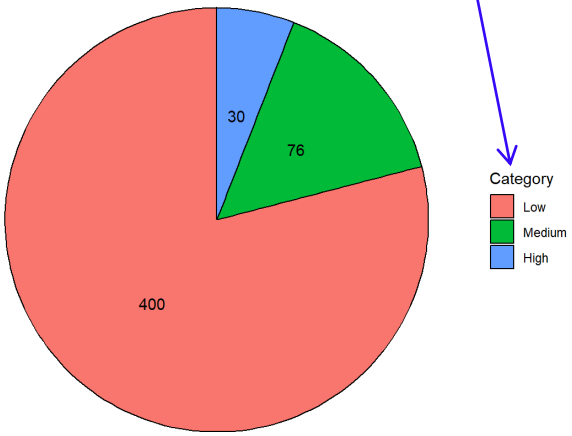
Distribution of crime rate categories: pie chart (ggplot2)

Layer 2: dadding the information about the groups counts on the pie chart.

```
counttab=as.data.frame(table(Boston3$crim_cat))
colnames(counttab)=c("Category", "Freq")

plot2=ggplot(counttab, aes(x="", y=Freq, fill=Category)) +
  geom_bar(stat="identity", width=1, color="black") +
  coord_polar("y", start=0)+
  theme_void()+
  geom_text(aes(label = Freq),
            position = position_stack(vjust =
plot2
```

```
##   Category Freq
## 1      Low  400
## 2   Medium   76
## 3     High   30
```



Distribution of crime rate categories: pie chart (ggplot2)

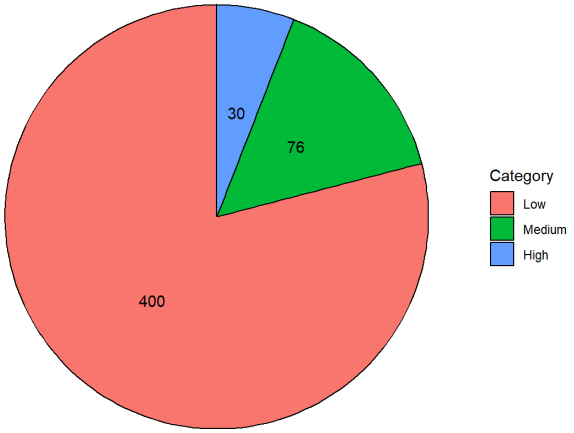
Layer 2: dadding the information about the groups counts on the pie chart.

```
plot2=ggplot(counttab, aes(x="", y=Freq, fill=Category)) +
  geom_bar(stat="identity", width=1, color="black") +
  coord_polar("y", start=0)+
  theme_void()+
  geom_text(aes(label = Freq),
            position = position_stack(vjust = 0.5))
plot2
```

Different colors for the categories

```
aes(x="", y=Freq, fill=Category))
geom_bar(stat="identity", width=1,
         color="black")
```

Use the counts



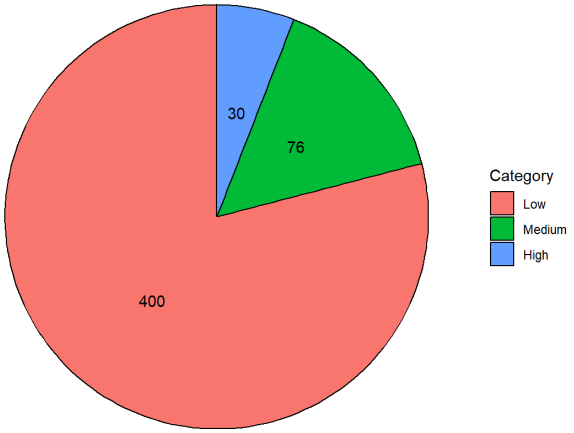
Distribution of crime rate categories: pie chart (ggplot2)

Layer 2: dadding the information about the groups counts on the pie chart.

```
plot2=ggplot(counttab, aes(x="", y=Freq, fill=Category)) +
  geom_bar(stat="identity", width=1, color="black") +
  coord_polar("y", start=0)+
  theme_void()+
  geom_text(aes(label = Freq),
            position = position_stack(vjust = 0.5))
plot2
```

```
coord_polar("y", start=0)
theme_void()
geom_text(aes(label = Freq),
          position = position_stack(vjust = 0.5))
```

Add the variable `Freq` as text.



Distribution of crime rate categories: barplot (ggplot2)

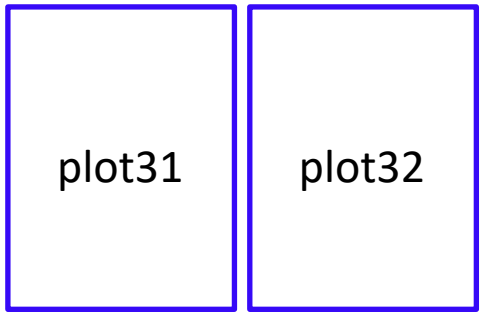
Layer 1: barplot without/with frequencies.

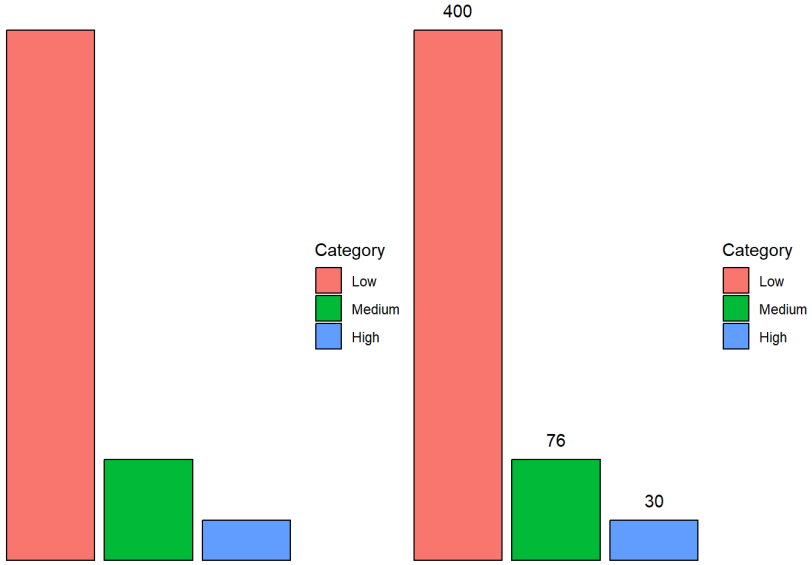
```
plot31=ggplot(counttab, aes(x=Category, y=Freq, fill=Category)) +
  geom_bar(stat = "identity", color="black")+
  theme_void()

plot32=ggplot(counttab, aes(x=Category, y=Freq, fill=Category)) +
  geom_bar(stat = "identity", color="black")+
  theme_void()+
  geom_text(aes(label = Freq),vjust=-1)

library(gridExtra)
grid.arrange(plot31, plot32, ncol=2)
```

Compare this to the pie chart





Distribution of crime rate by chas

```
counttab1=as.data.frame(table(Boston3$crim_cat,Boston3$chas))
colnames(counttab1)=c("Category","Chas1", "Freq")
counttab1
```

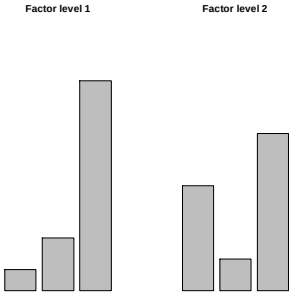
What do we want to visualize ?

```
##   Category Chas1 Freq
## 1      Low     0  370
## 2   Medium     0   71
## 3     High     0   30
## 4      Low     1   30
## 5   Medium     1    5
## 6     High     1    0
```

A factor with three levels.

A factor with two levels.

	Chas1=0	Chas1=1
Low	370	30
Medium	71	5
High	30	0



The distribution of crime rate by Chas category.

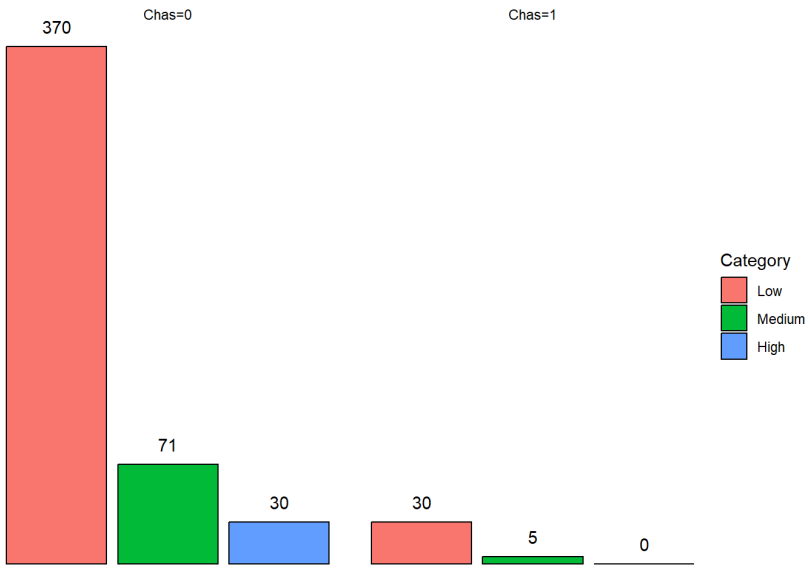
Distribution of crime rateby chas categories: barplot (`ggplot2`)

Layer 2: dadding the information about the groups means to the dotplot.

```
plot31a=ggplot(counttab1, aes(x=Category, y=Freq, fill=Category)) +
   geom_bar(stat = "identity", color="black")+
   geom_text(aes(label = Freq),vjust=-1)+
   facet_wrap(~as.factor(Chas1),labeller = as_labeller(c("1" = "Chas=1", "0" = "Chas=0")))+
   theme_void()
Plot31a
```

```
aes(x=Category, y=Freq, fill=Category))
```

```
geom_bar(stat = "identity", color="black")
geom_text(aes(label = Freq),vjust=-1)
```



Distribution of crime rateby chas categories: barplot (ggplot2)

Layer 2: dadding the information about the groups means to the dotplot.

```
plot31a=ggplot(counttab1, aes(x=Category, y=Freq, fill=Category)) +
   geom_bar(stat = "identity", color="black")+
   geom_text(aes(label = Freq),vjust=-1)+
   facet_wrap(~as.factor(Chas1),labeller = as_labeller(c("1" = "Chas=1", "0" = "Chas=0")))+
   theme_void()
Plot31a
```

```
facet_wrap(~as.factor(Chas1),
           labeller = as_labeller
           (c("1" = "Chas=1", "0" = "Chas=0")))
```