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.

Interuniversity Institute for Biostatistics
and statistical Bioinformatics

The >eR-Biostat initiative

Making R based education materials in statistics accessible for all

Introduction to Visualization using R: Continuous variable in (one population)

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

LAST UPDATE: 05/2024

Visit us on Facebook ER-BioStat

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

twitter @erbiostat

Software

- R functions for visualization:
  - `ggplot2.`
- R program for the examples is available online:
  - `Visualization_intro.Rmd`.

Datasets

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

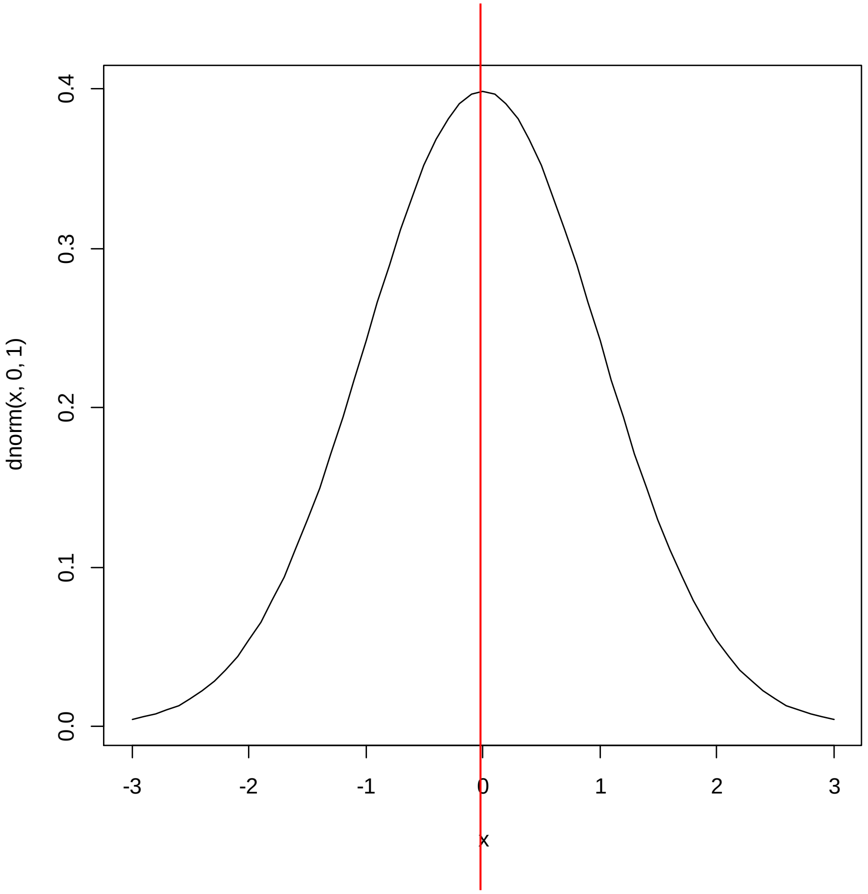
Topics

1. EDA and visualization for location and spread.
2. Introduction to the R package `ggplot2`.

Part 1: Location & spread

Location

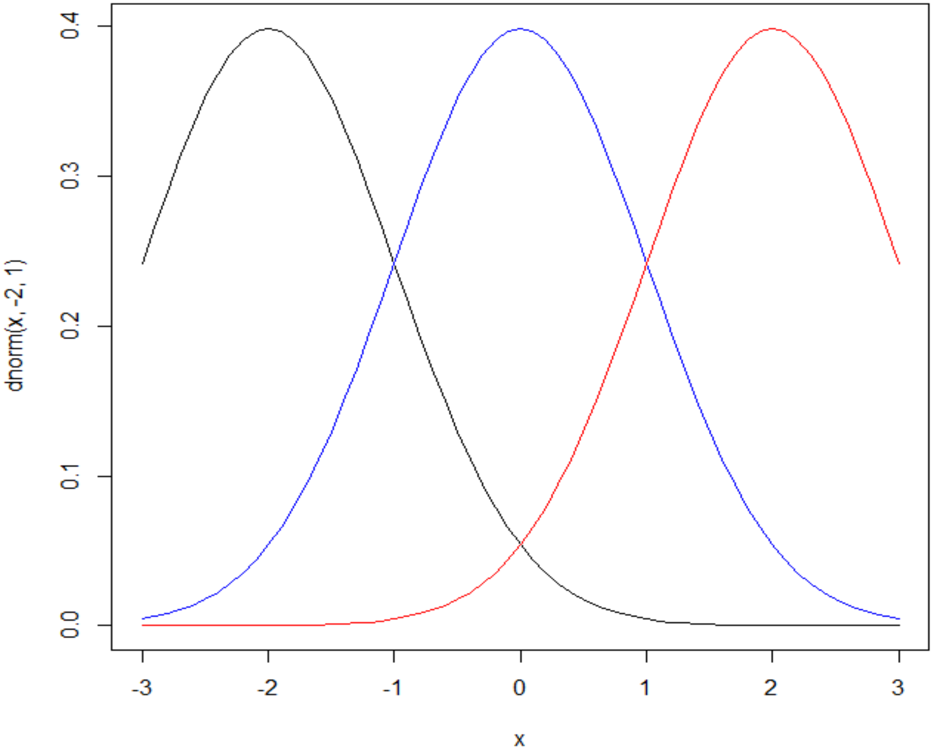
Density of standard normal, N(0,1), distribution



The center 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.

Introduction

```
> 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")
```

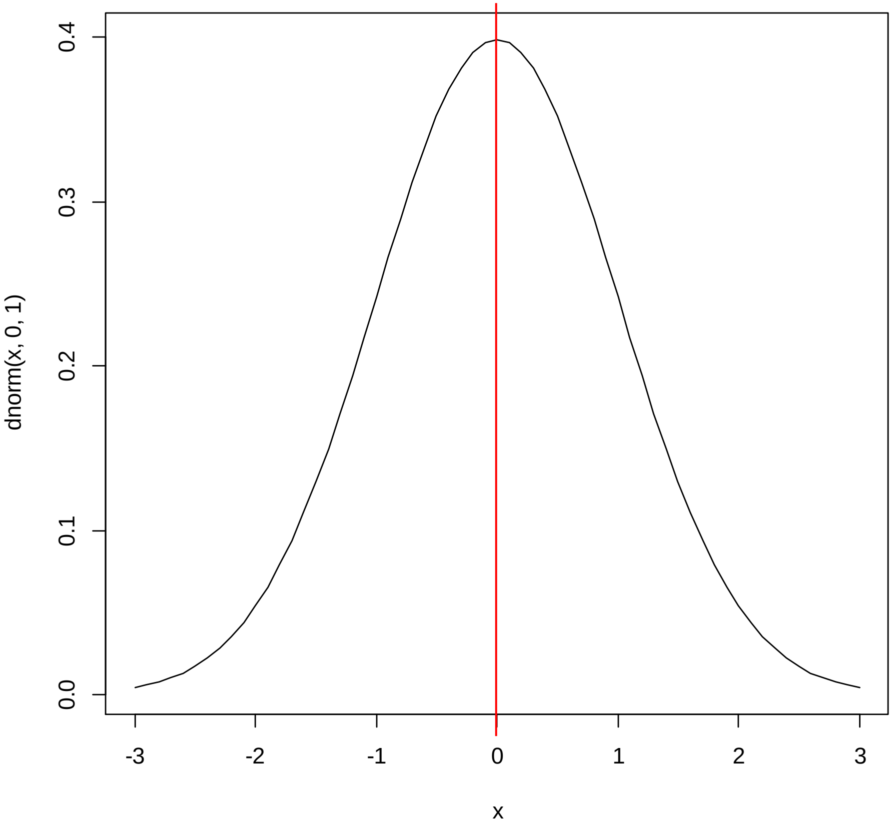
Numerical summaries for location

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

The estimator for μ is called location estimator.

Numerical summaries:

- 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 ?

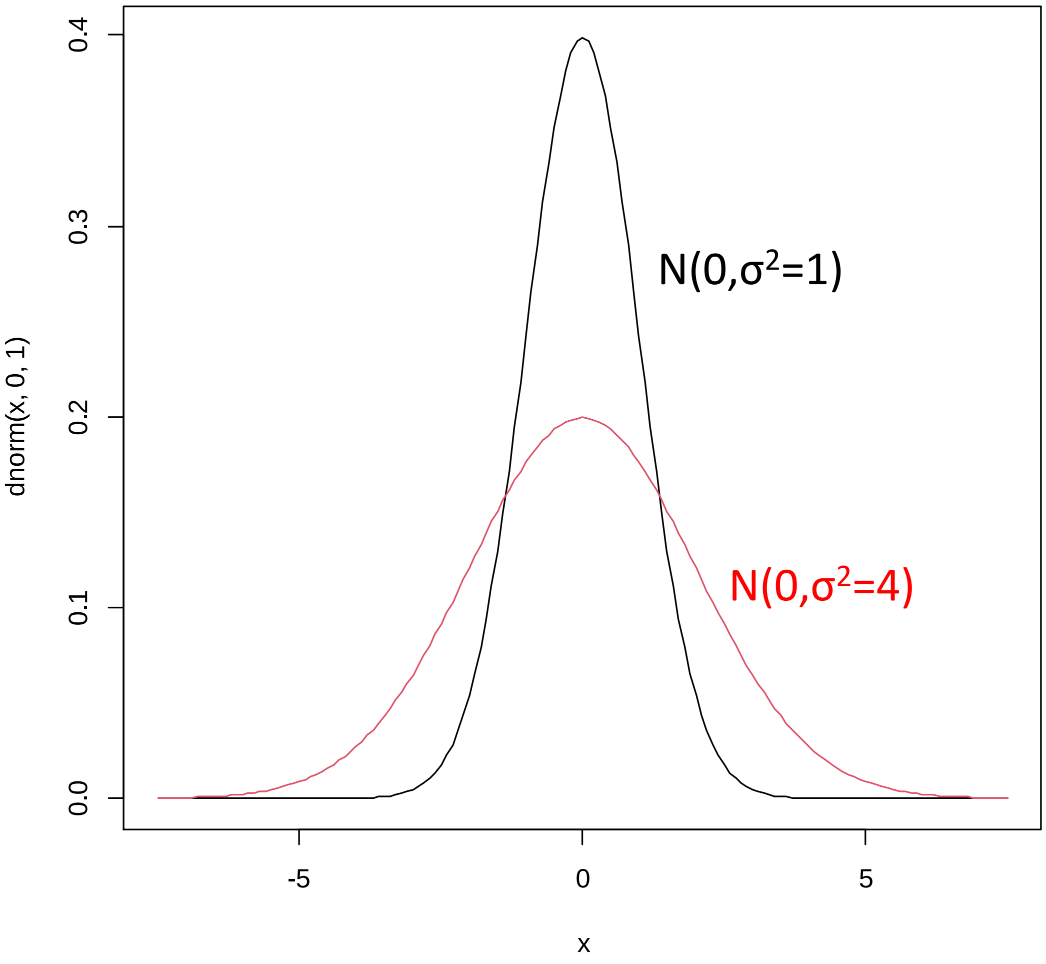
Spread

Spread

- Until now we summarized the distribution of the data with location estimators
- In this chapter we will focus on the **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.

Spread



- 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.

- Hence, sample variance is sensitive to change, but four-spread is not.

What next ?

- How to visualize the location and the spread of a distribution ?
- Which graphical display to use ?
- What can we learn from a figure about the location and pread of the destruction.

- All examples: one sample of numerical variable.

Part 2: the R package `ggplot2`

The R package ggplot2

- `ggplot2` is a plotting R package that 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

- ggplots graphics are built layer by layer by adding new elements.
- Adding layers in this fashion allows for extensive flexibility and customization of plots.

ggplot2 Layers

- Layers in ggplots graphics are related to:
  - Data.
  - Variables to be use.
  - Type of plots.
  - Setting of the figure.

Part 3: Examples

Example 1

The `airquality` data

Daily average of wind speed

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

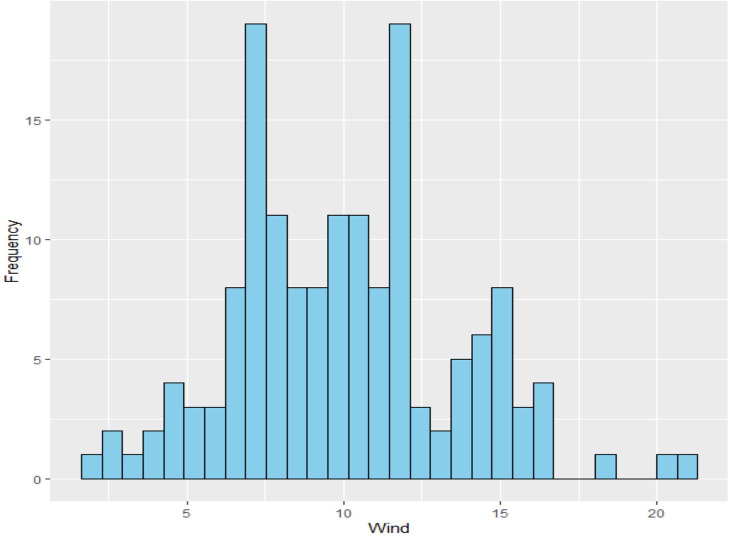
- Daily air quality measurements in New York, May to September 1973.

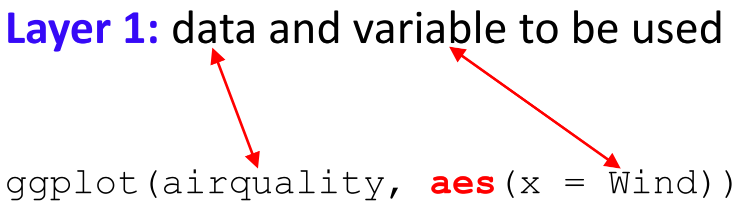
```
> help("airquality")
> airquality$Wind
  [1]  7.4  8.0 12.6 11.5 14.3 14.9  8.6 13.8 20.1  8.6  6.9  9.7  9.2 10.9 13.2 11.5 12.0 18.4
 [19] 11.5  9.7  9.7 16.6  9.7 12.0 16.6 14.9  8.0 12.0 14.9  5.7  7.4  8.6  9.7 16.1  9.2  8.6
 [37] 14.3  9.7  6.9 13.8 11.5 10.9  9.2  8.0 13.8 11.5 14.9 20.7  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.
- How the distribution look like?

Histogram of wind speed

```
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))
```

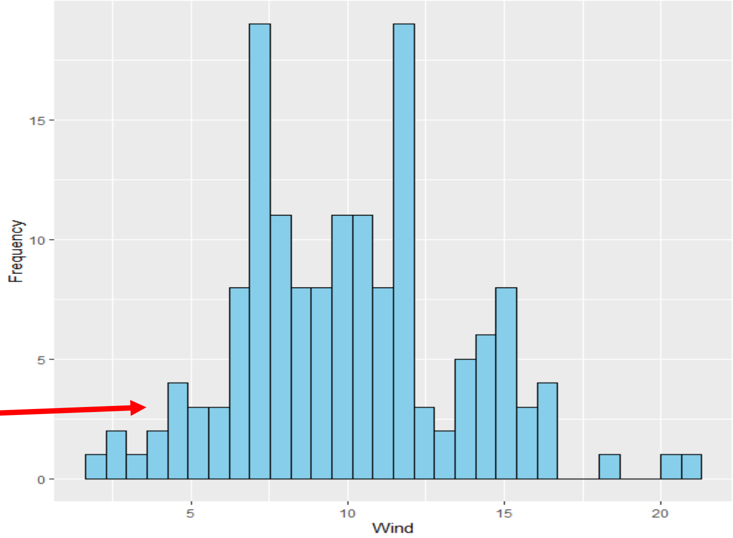
- We define an **aesthetic** mapping (using the `aes()` function:
  - Select the variable(s) to be plotted.
  - Specify how to present them in the graph, e.g., as x/y positions.

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=…`

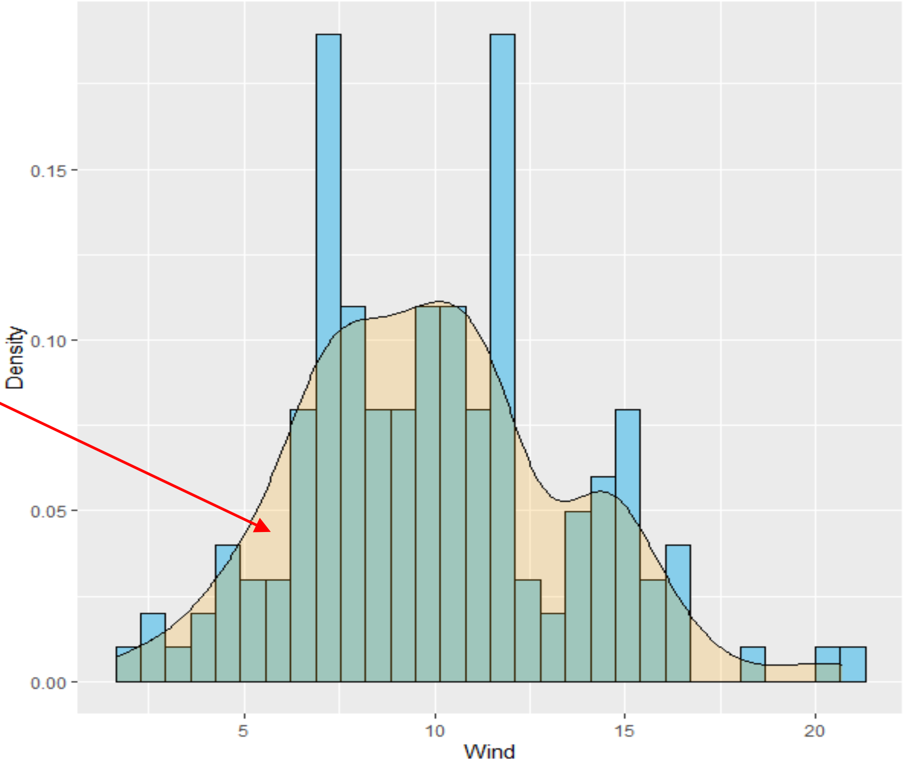
Histogram with density plot of wind speed

```
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=…`



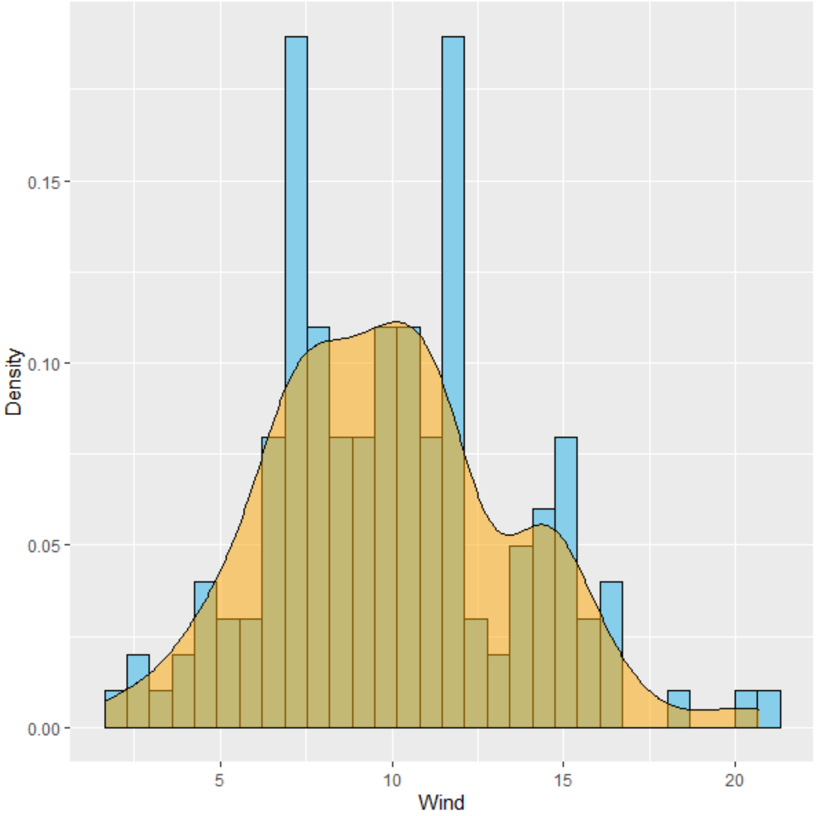
Histogram with density plot of wind speed

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

Layer 3: adding the density plot:

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

- Changing the value of alpha:



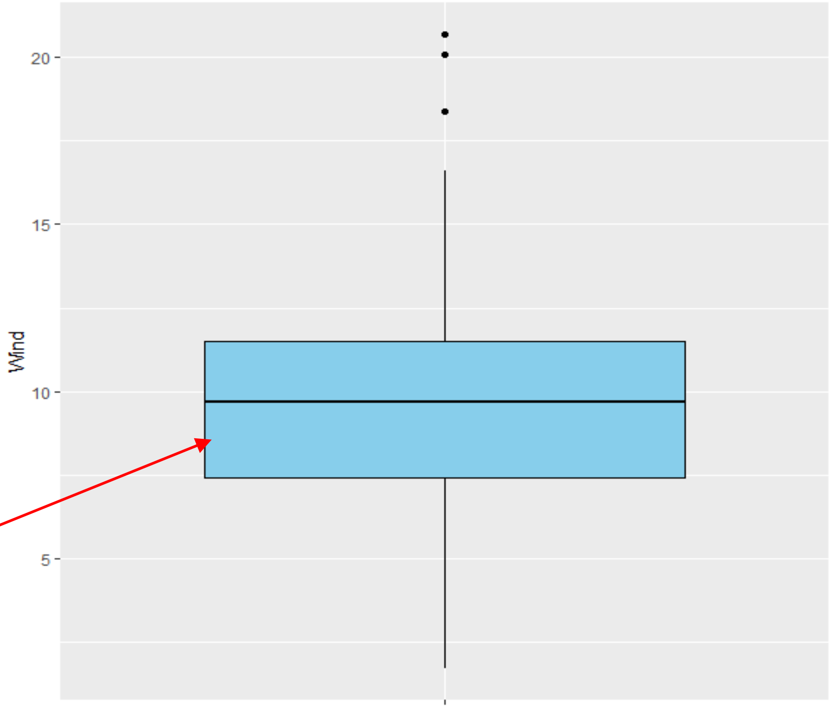
Boxplot of wind speed

```
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")
```

The colors of the lines.

`geom_boxplot`: plot a boxplot.

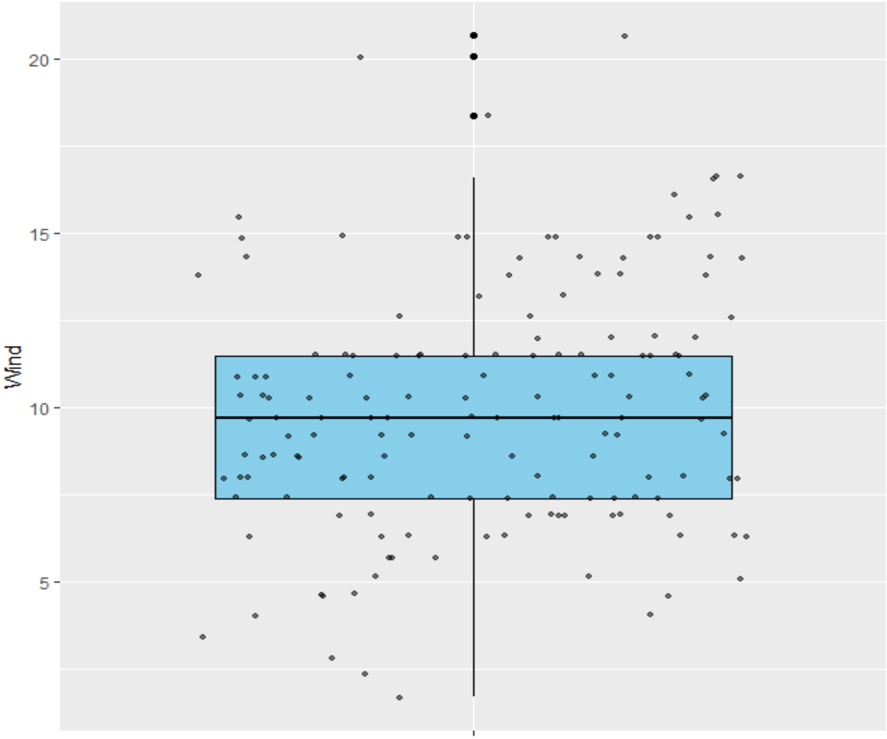
Boxplot of wind speed with data points

```
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

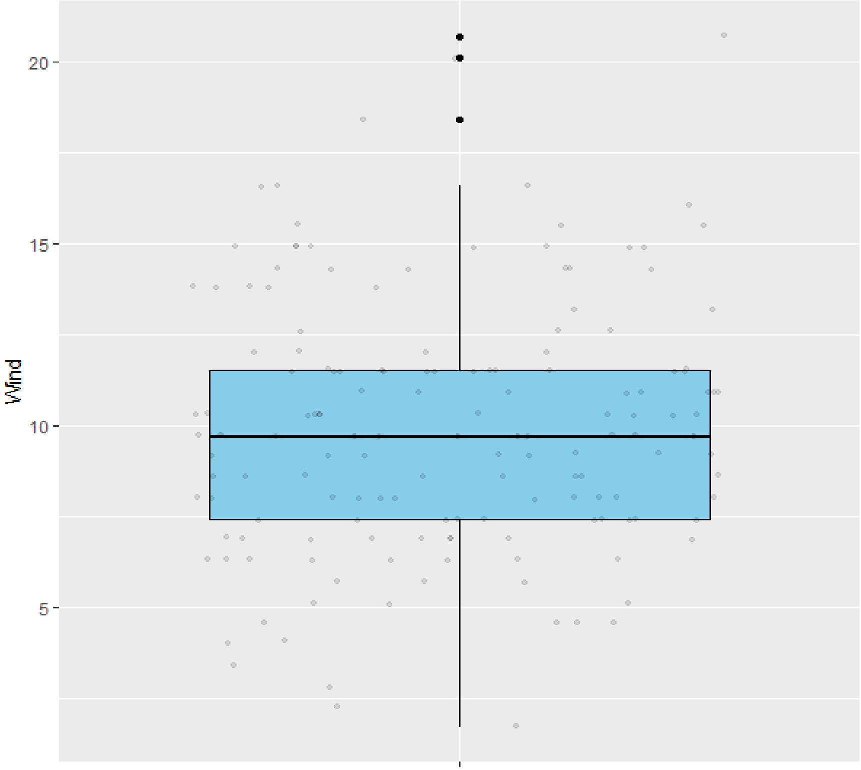
```
ggplot(airquality, aes(x = "", y = Wind)) +
geom_boxplot(fill = "skyblue", color = "black") +
geom_jitter(aes(x = "", y = Wind), color = "black", size = 1, alpha = 0.1) +
xlab("")
```

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



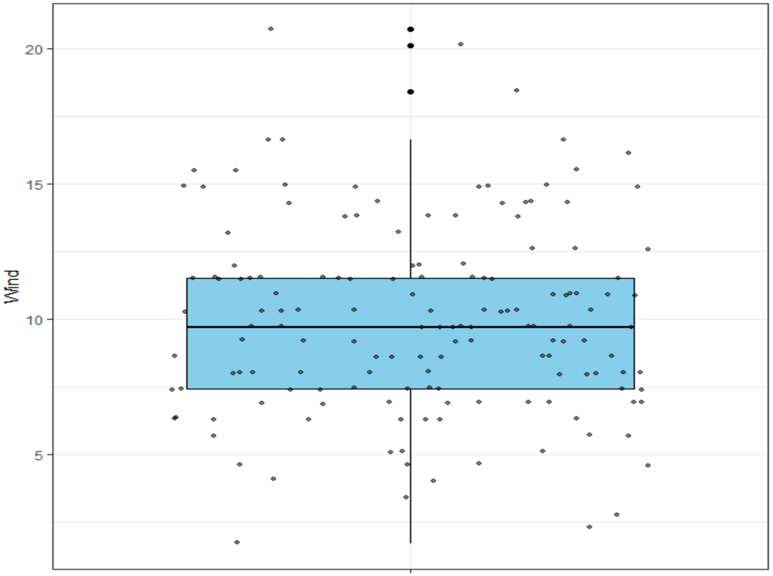
Boxplot of wind speed with data points

```
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()
```

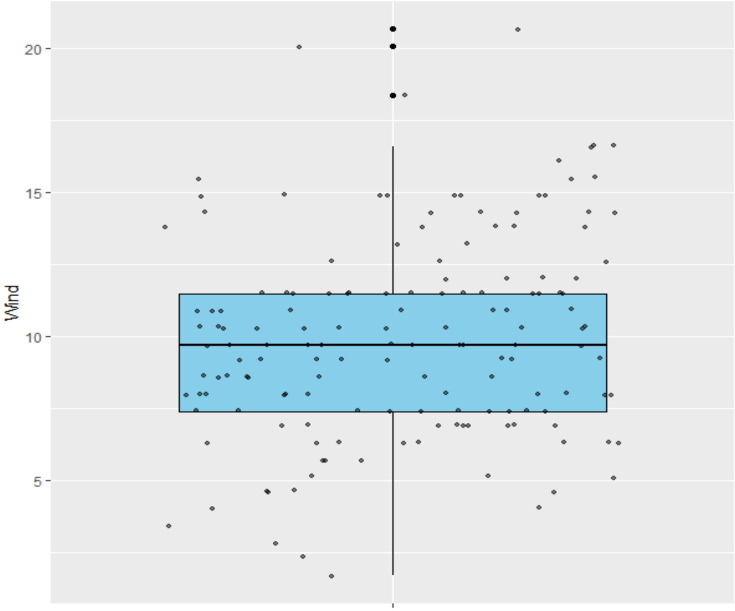
Layer 4: change the backgroup color:

`theme_bw()`

bw="black & white"



Point size with alpha=0.5



Violin plot of wind speed

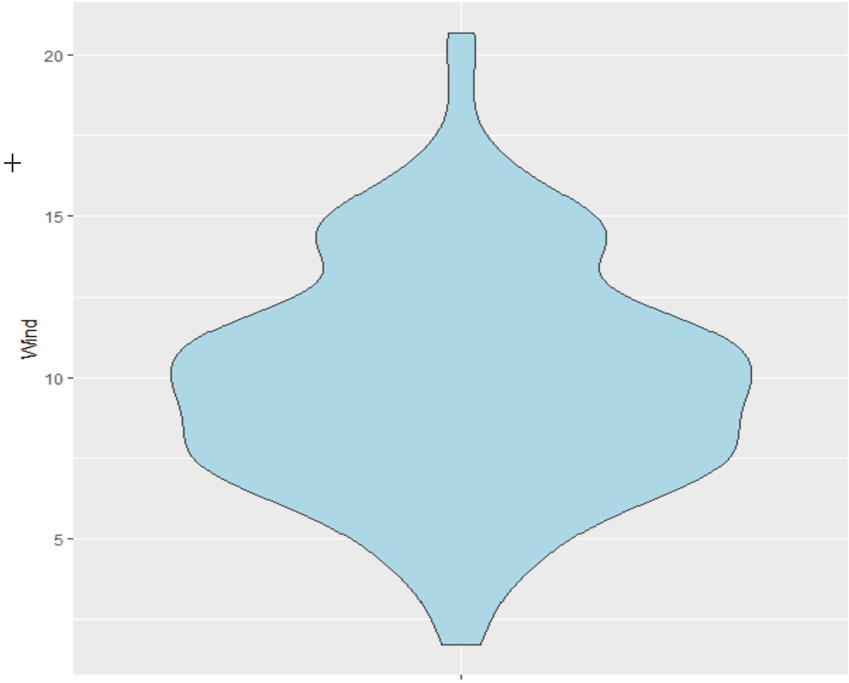
```
ggplot(airquality, aes(x = "", y = Wind)) +
geom_violin(fill = "lightblue") +  xlab("")
```

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("")
```

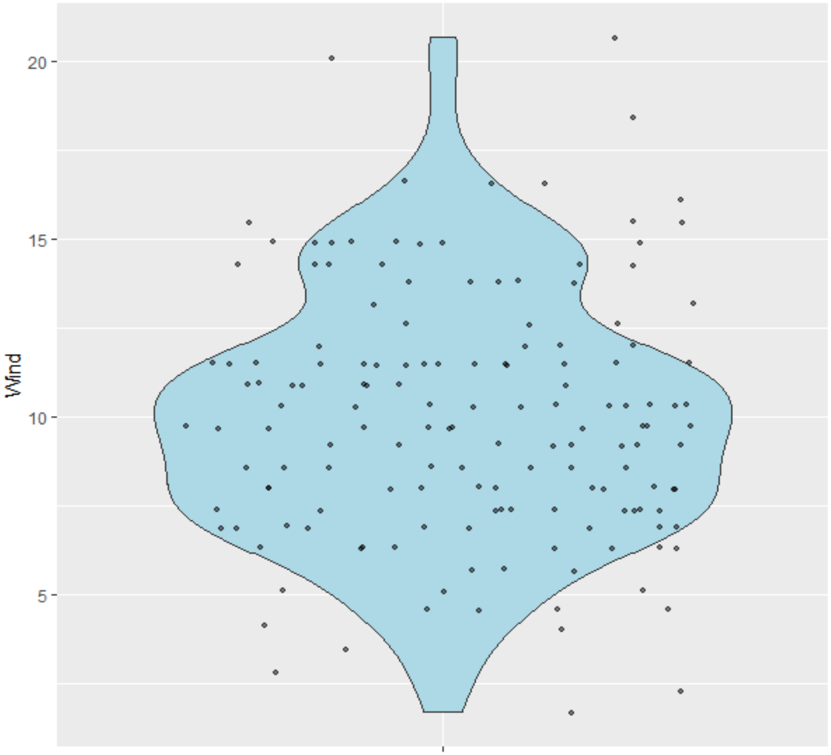


Violin plot of wind speed with data points

```
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)
```



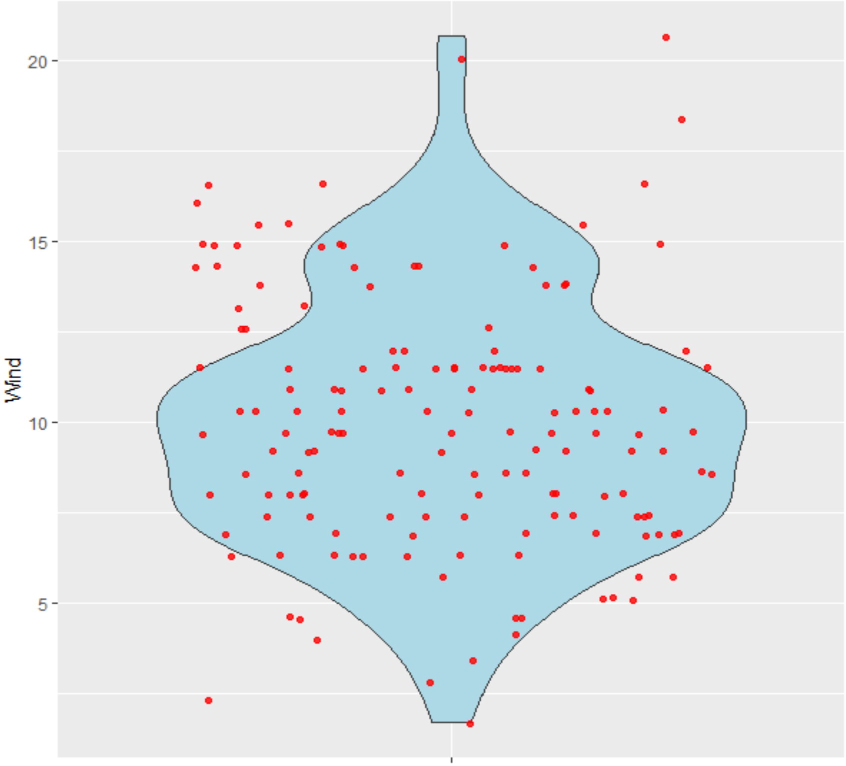
Violin plot of wind speed with data points

```
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 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).

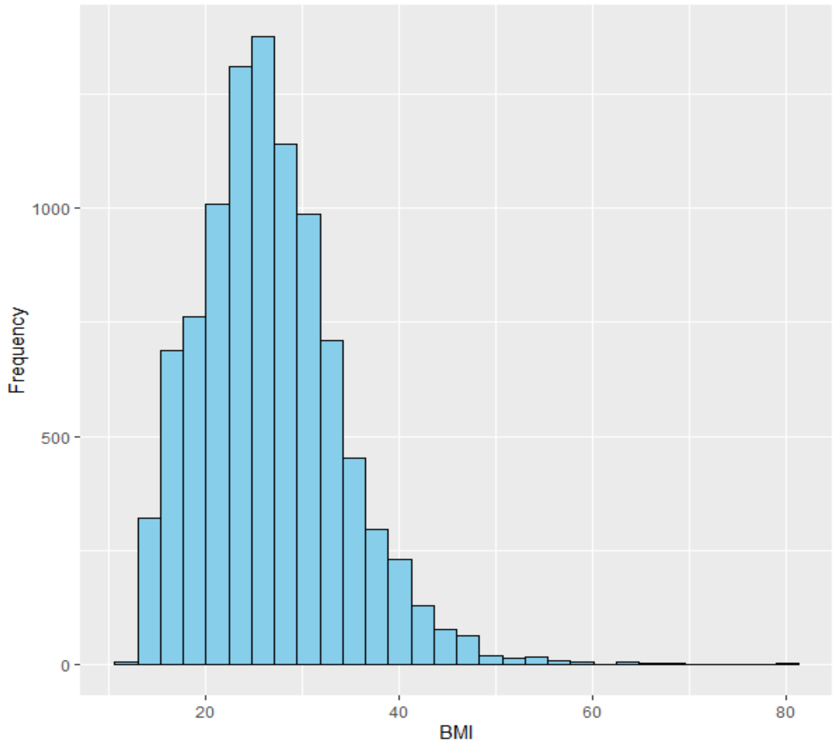
Histogram of BMI

```
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.



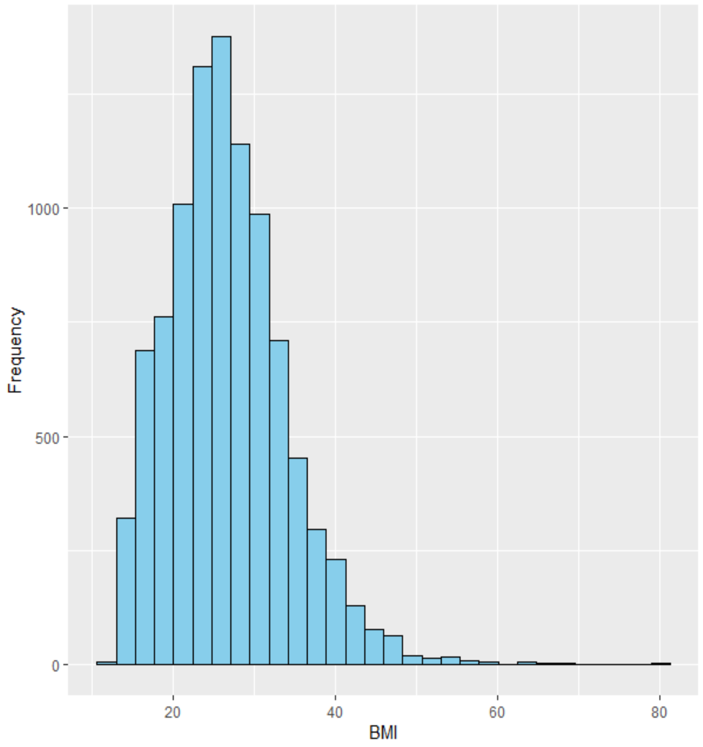
Histogram with density plot of BMI

```
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=…`



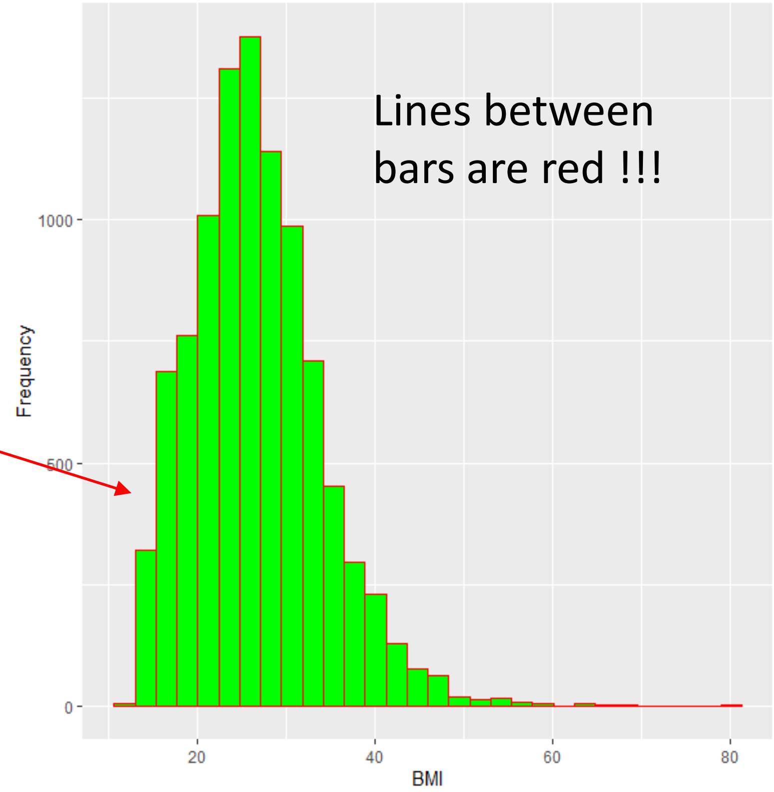
Histogram with density plot of BMI

```
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=…`



Lines between bars are red !!!

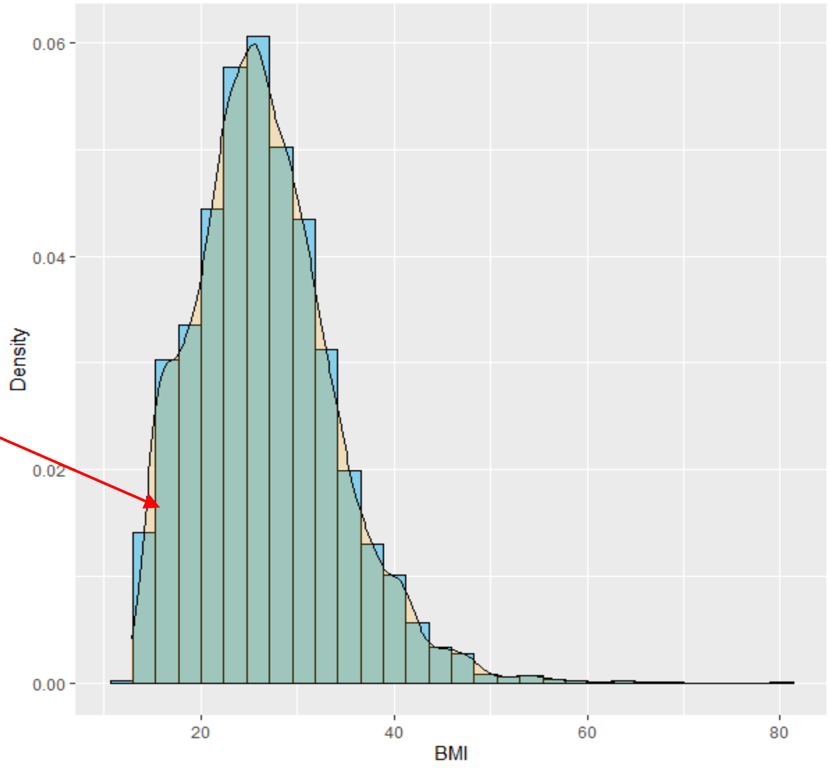
Histogram with density plot of BMI

```
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:

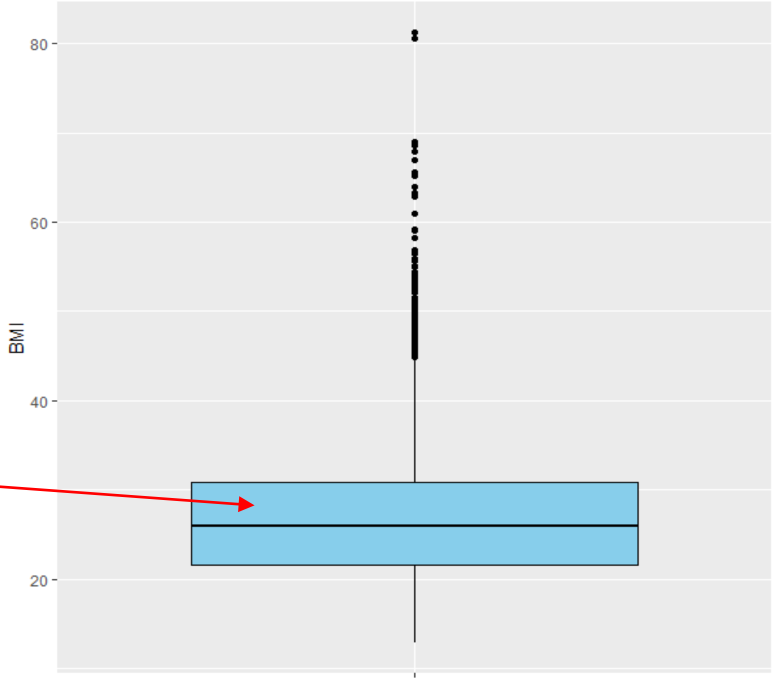
```
ggplot(NHANES, aes(x = "", y = BMI))
```

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.

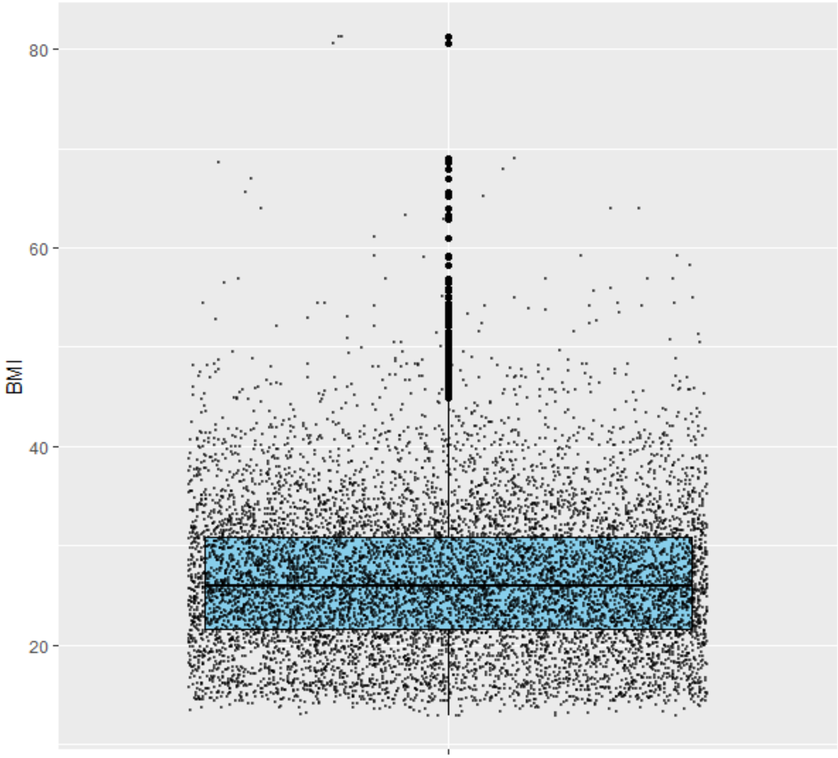
Boxplot of BMI with data points

```
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.

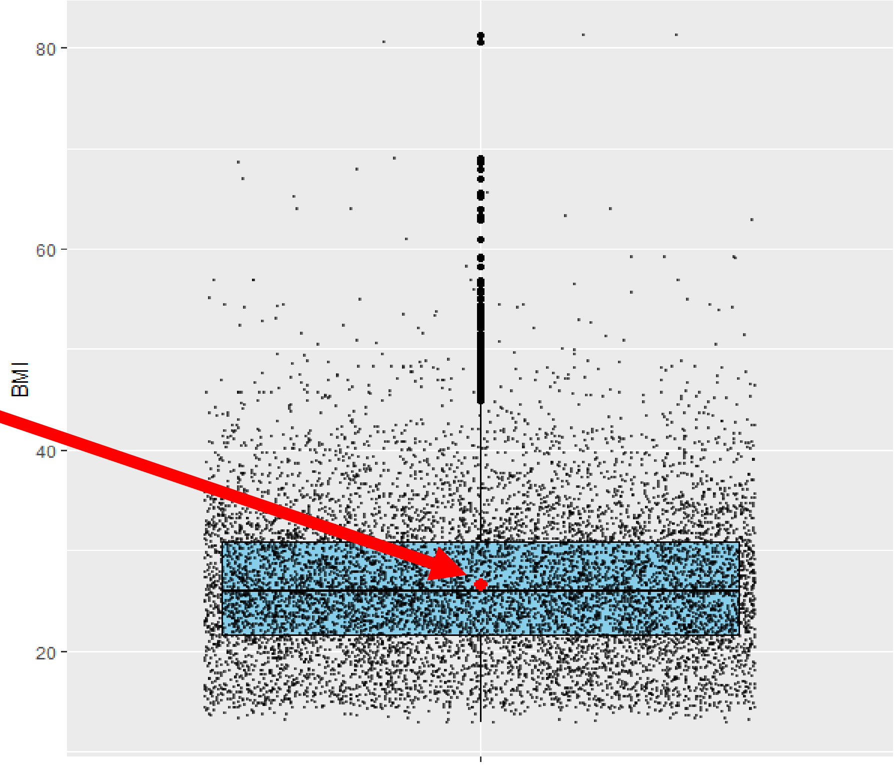


Boxplot of BMI with data points

```
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")
```



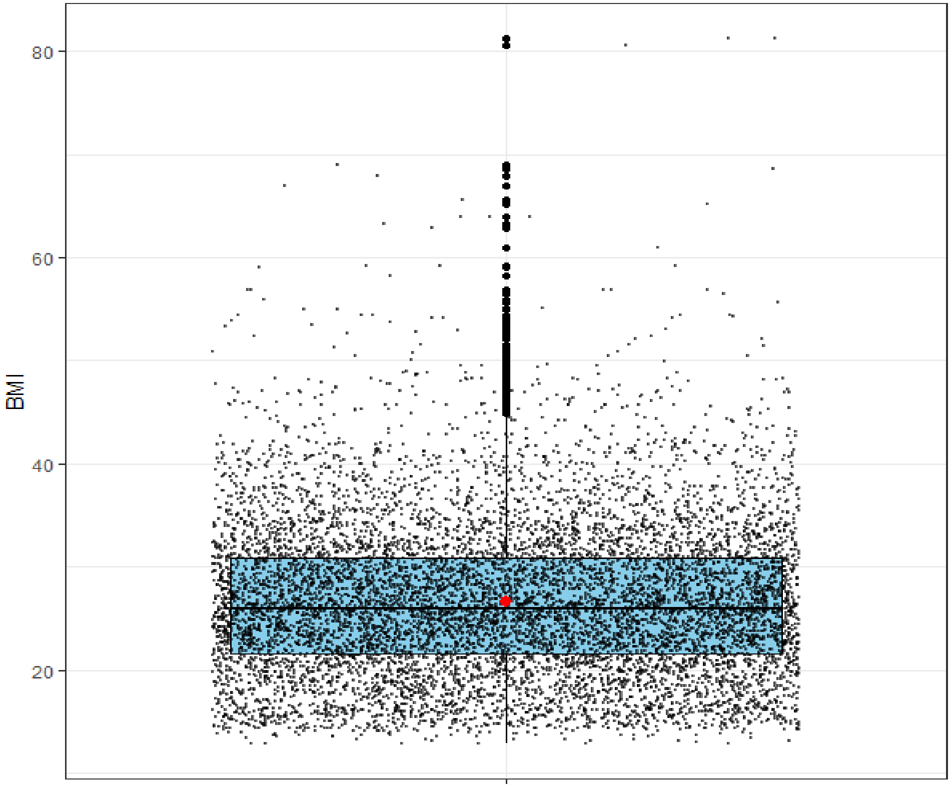
The function `stat_summary()` calculate summary stats of the data.

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

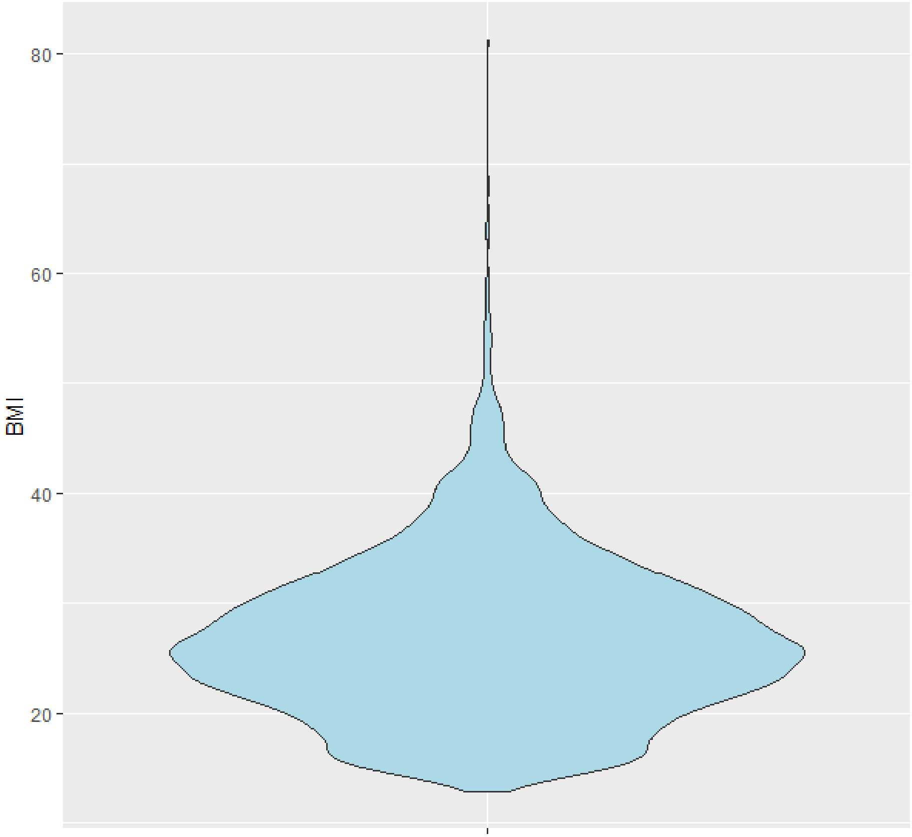
```
ggplot(NHANES, aes(x = "", y = BMI)) +
geom_violin(fill = "lightblue") +  xlab("")
```

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")
```

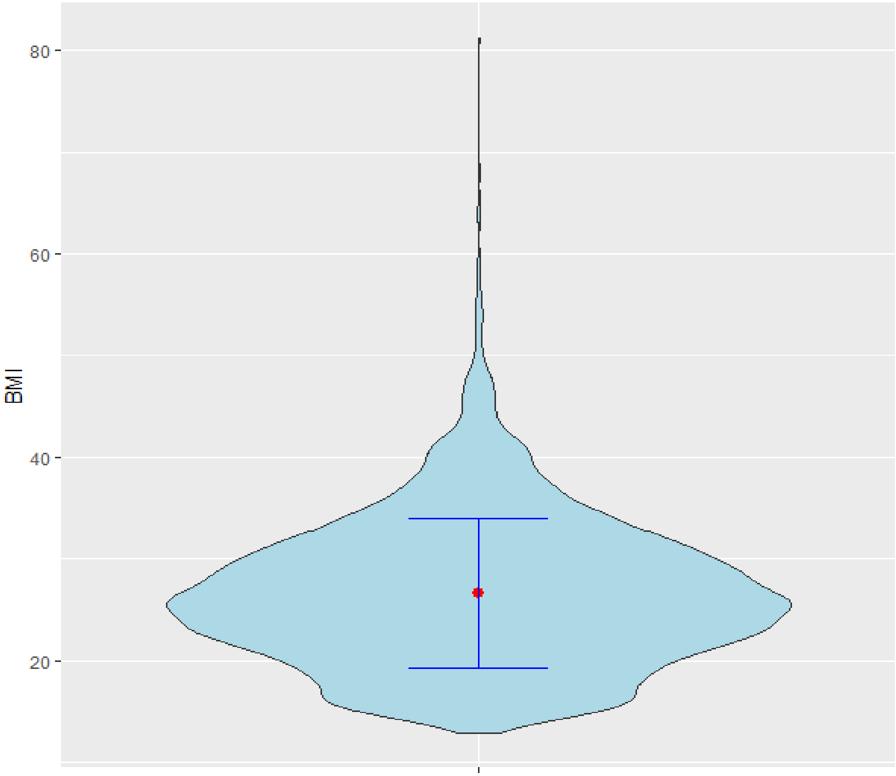
Layer 3: add the mean and SD to the plot:

- 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

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?

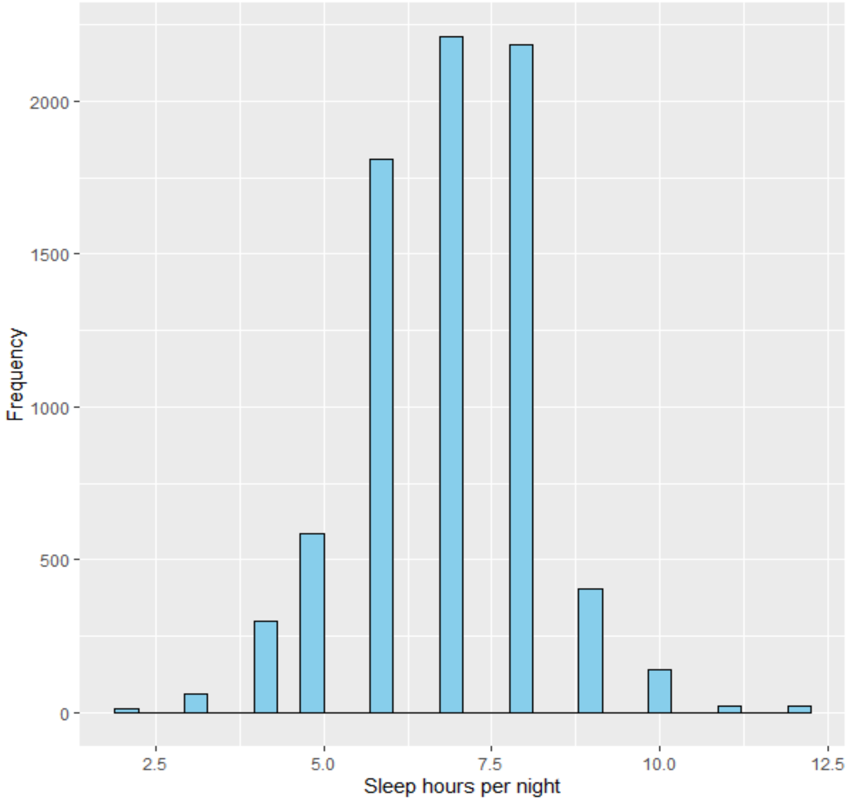
Histogram of number of sleep hours per night

```
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 **aesthetic** 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.



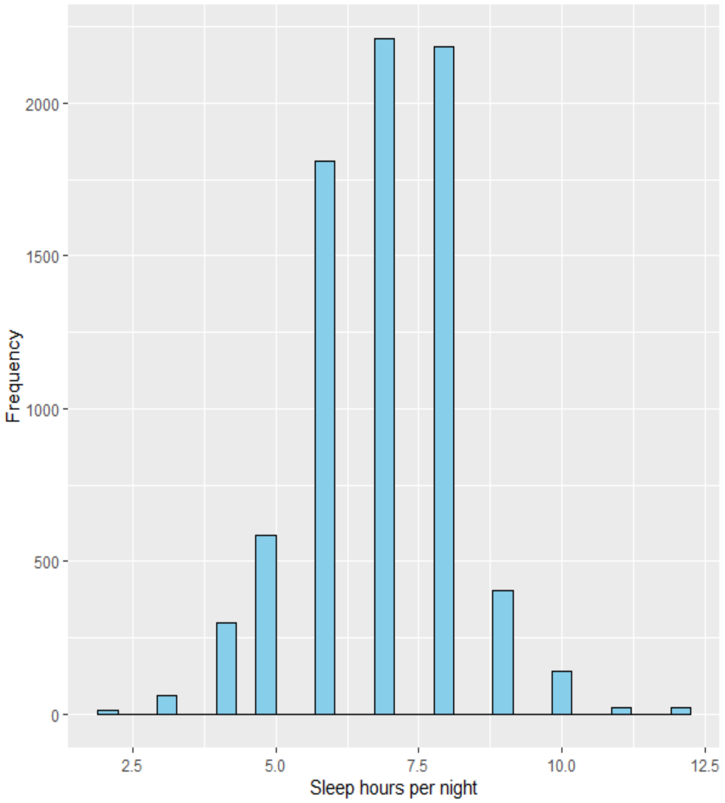
Histogram of number of sleep hours per night

```
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=…`



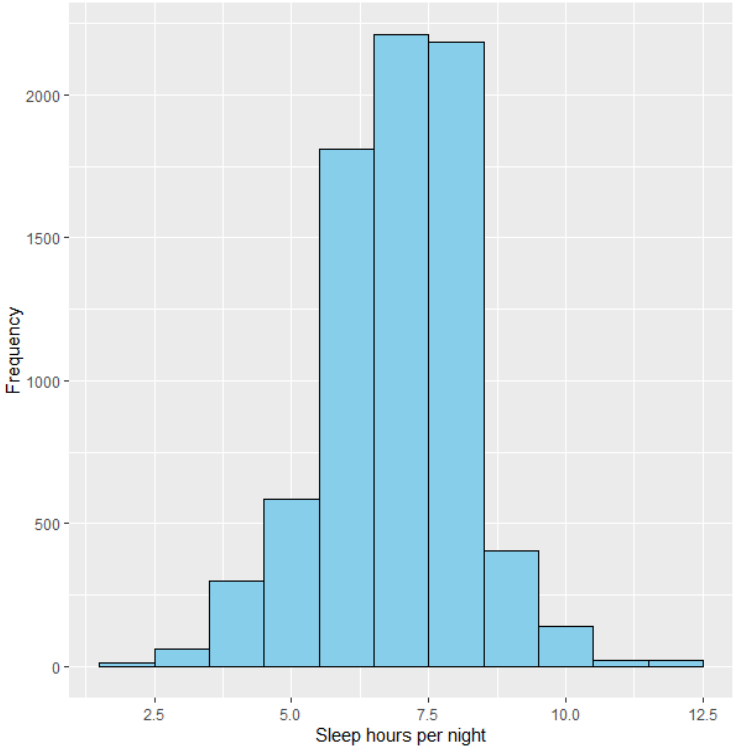
Histogram of number of sleep hours per night

```
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

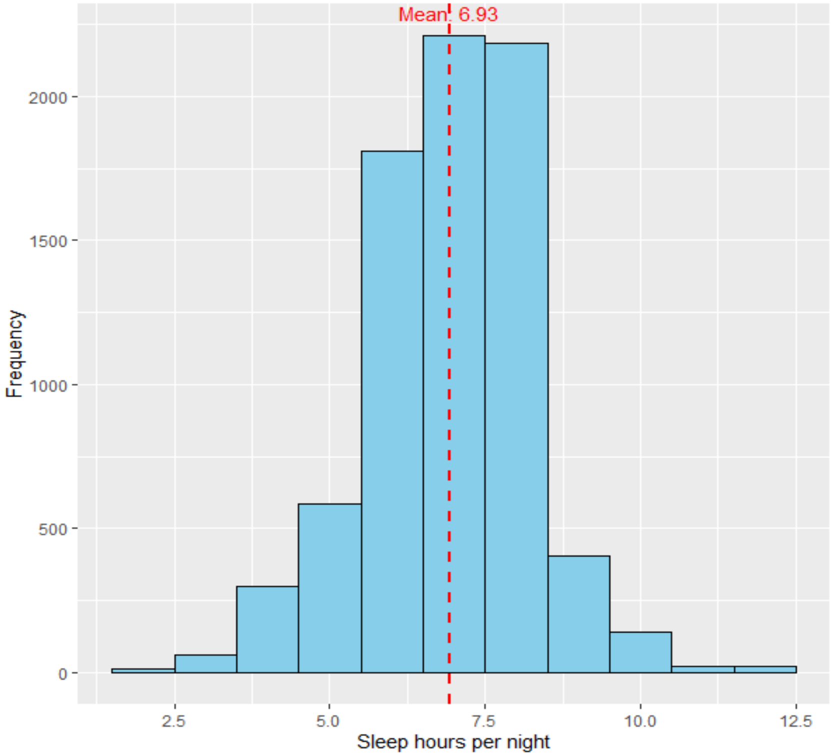
```
ggplot(NHANES, aes(x = SleepHrsNight)) +
geom_histogram(fill = "skyblue", color = "black") +  ylab("Frequency") +
xlab("Sleep hours per night") +
geom_vline(aes(xintercept = mean_sleep), color = "red", linetype =
"dashed", size = 1) +
annotate("text", x = mean_sleep, y = max(table(NHANES$SleepHrsNight)),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

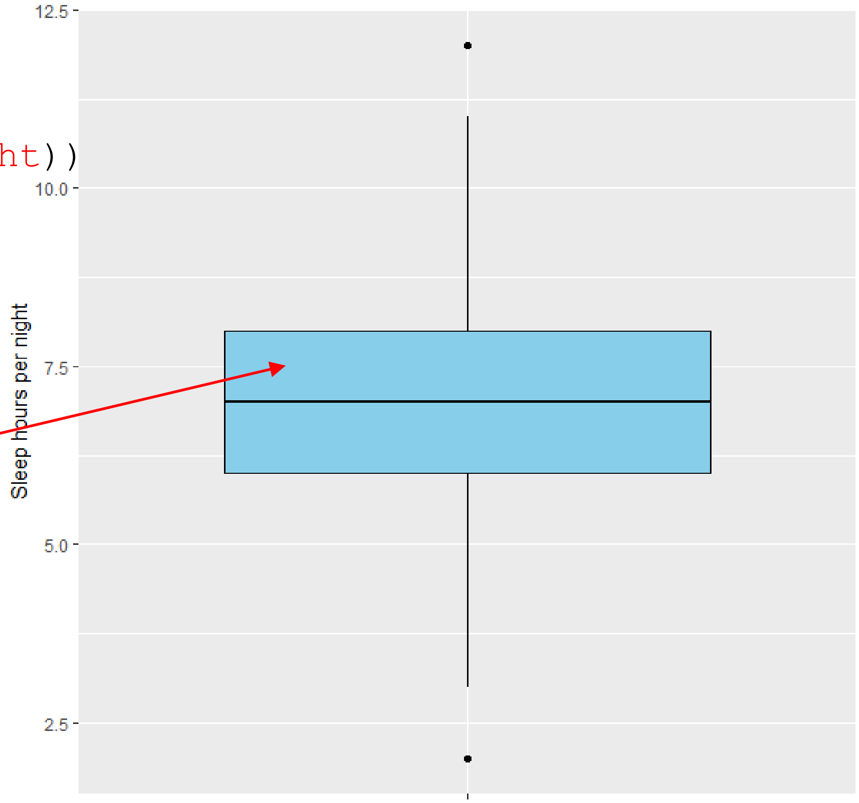
```
ggplot(NHANES, aes(x = "", y = SleepHrsNight))
```

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.



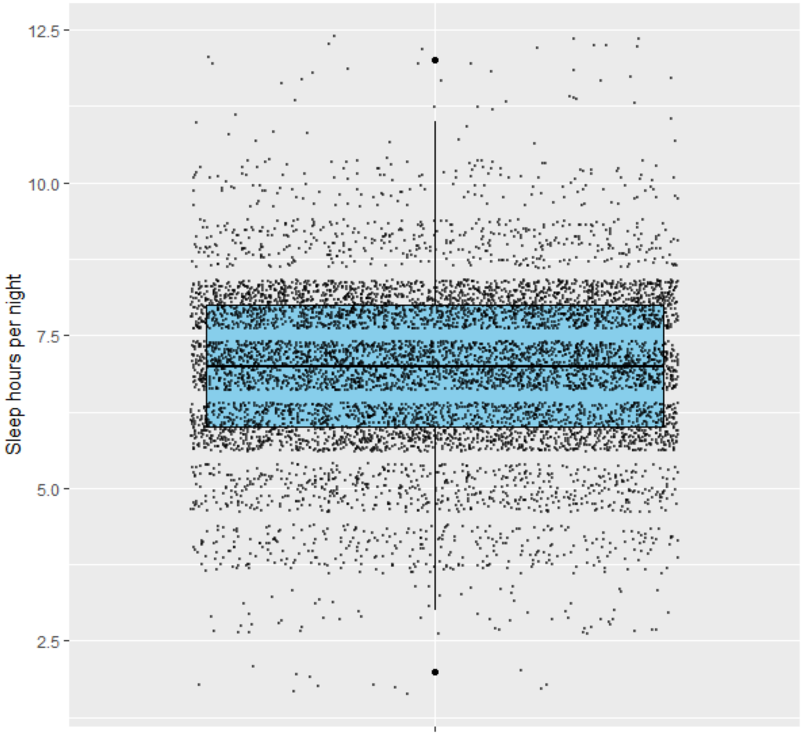
Boxplot of number of sleep hours per night with data points

```
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

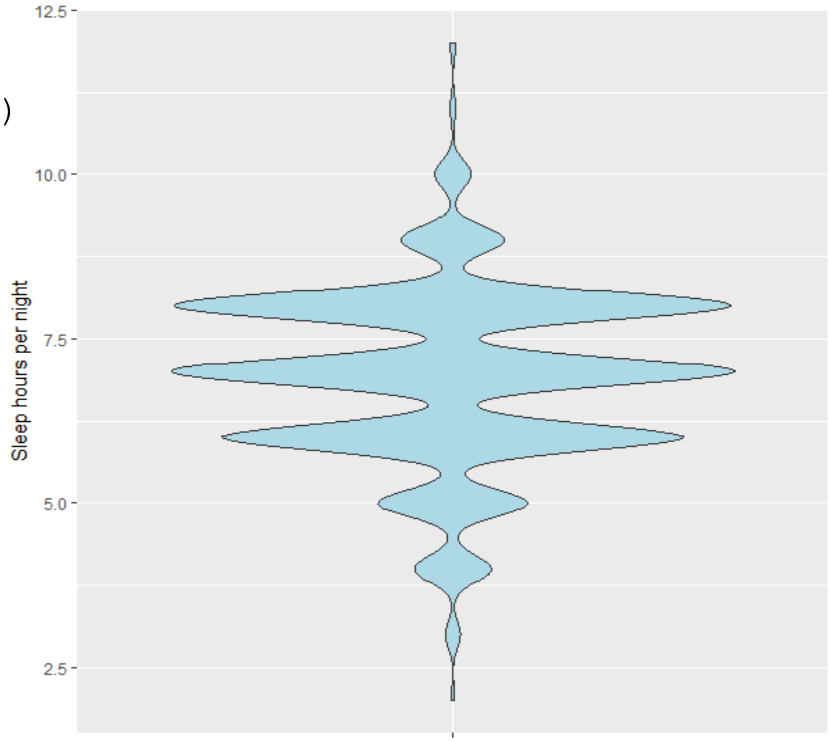
```
ggplot(NHANES, aes(x = "", y = SleepHrsNight)) +
geom_violin(fill = "lightblue")+
xlab("")+  ylab("Sleep hours per night")
```

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("")
```

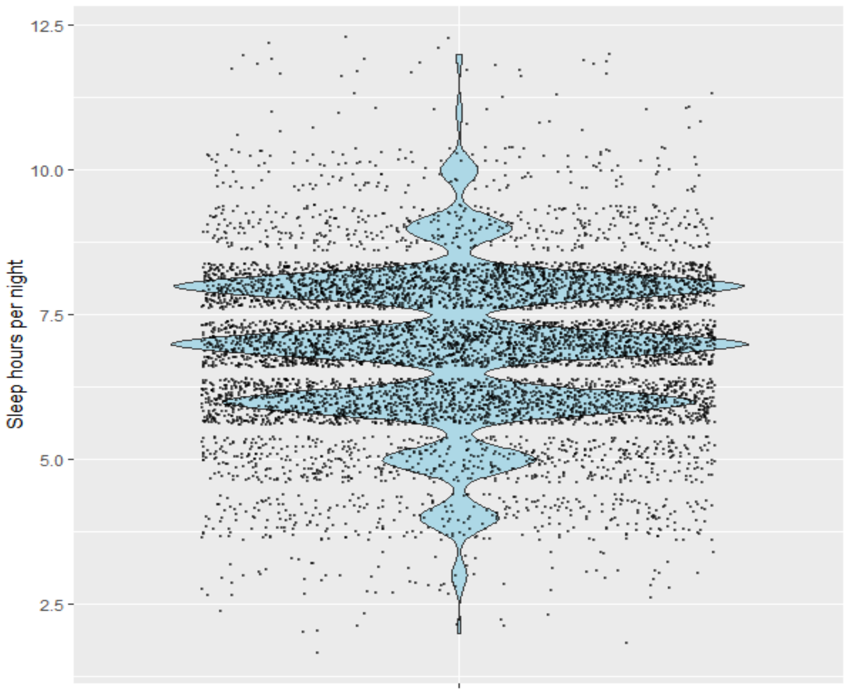


Violin plot of number of sleep hours per night with data points

```
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)
```

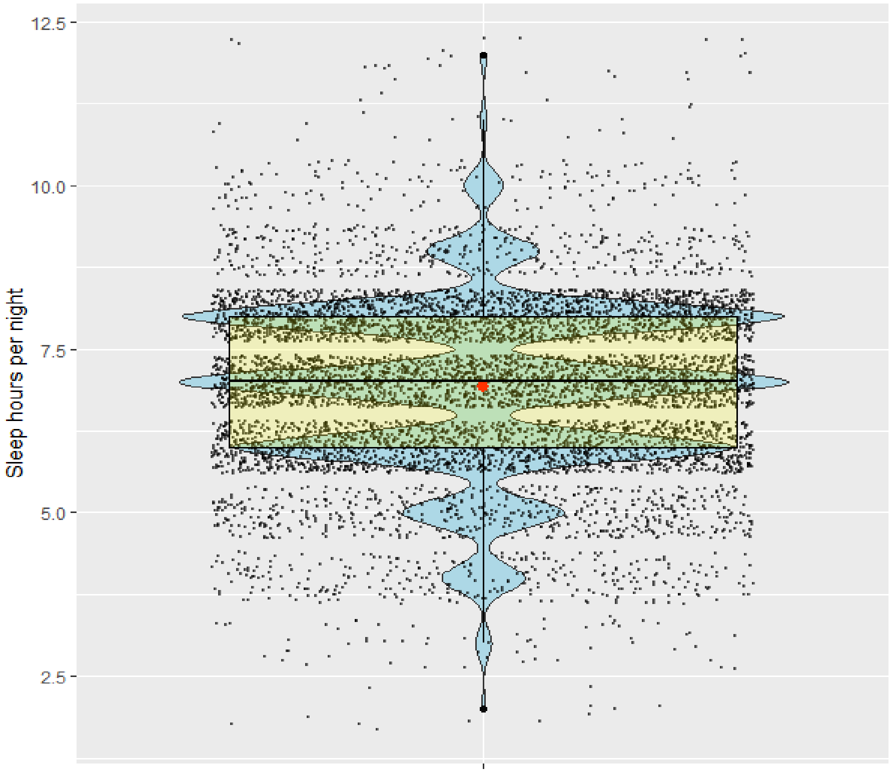


Violin plot of number of sleep hours per night with data points

```
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 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")
```

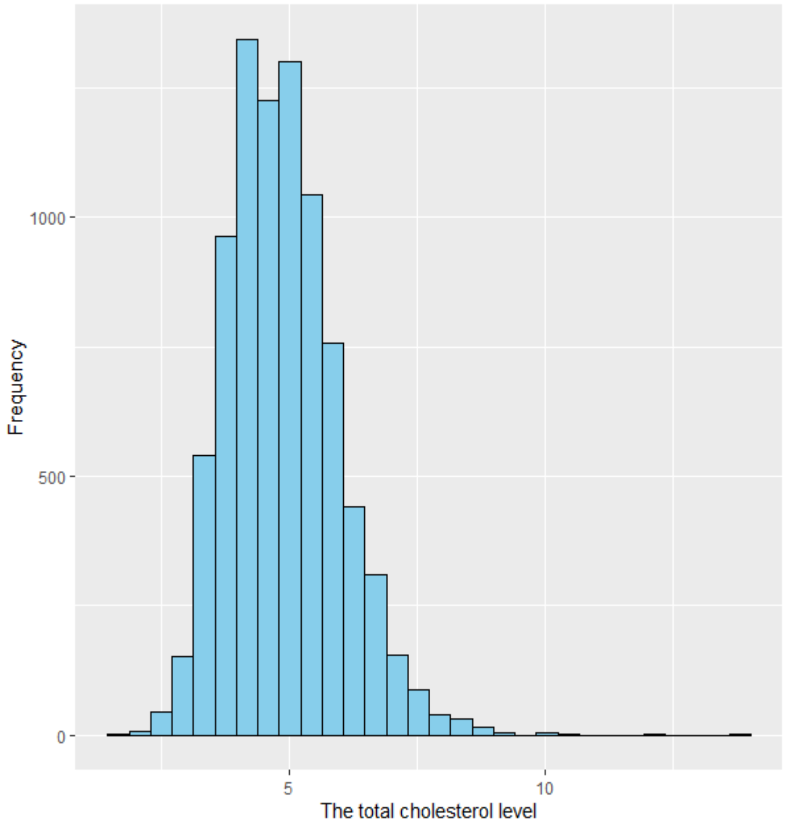
Layer 1: data and variable to be used:

```
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=`…

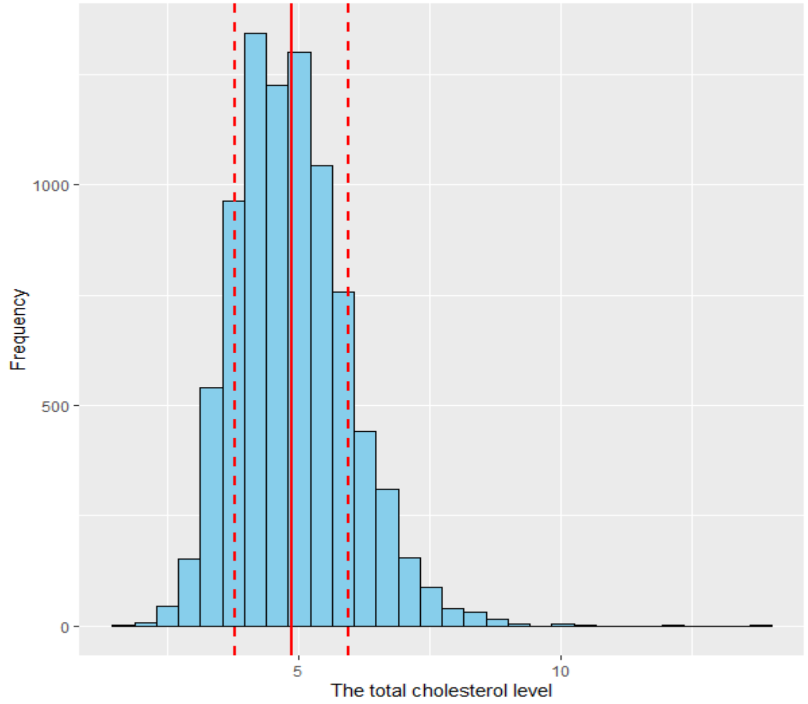


Histogram of the total cholesterol level

```
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:

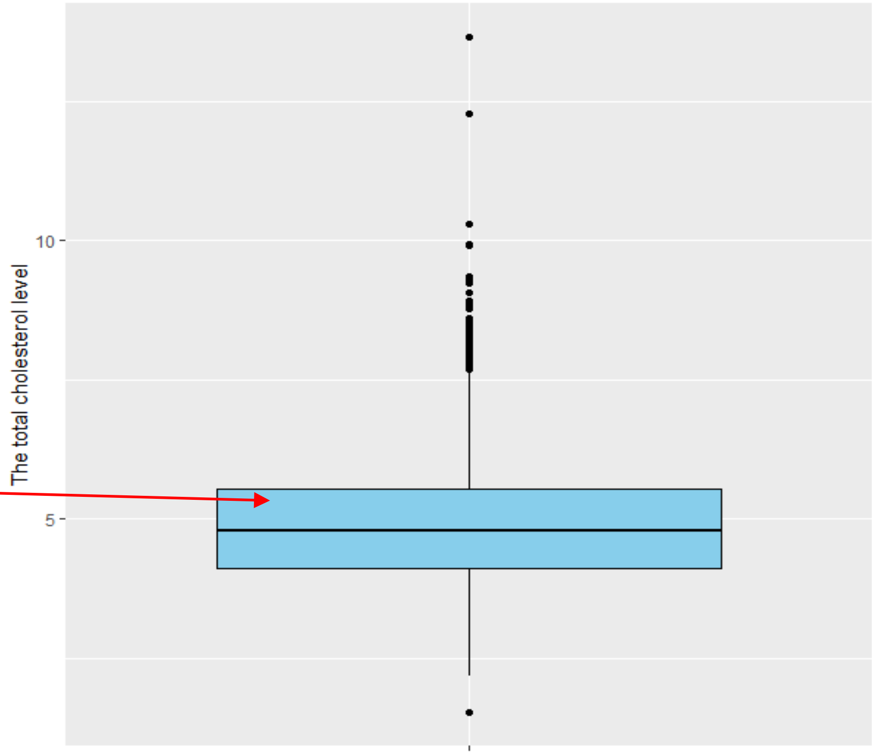
```
ggplot(NHANES, aes(x = "", y = TotChol))
```

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.

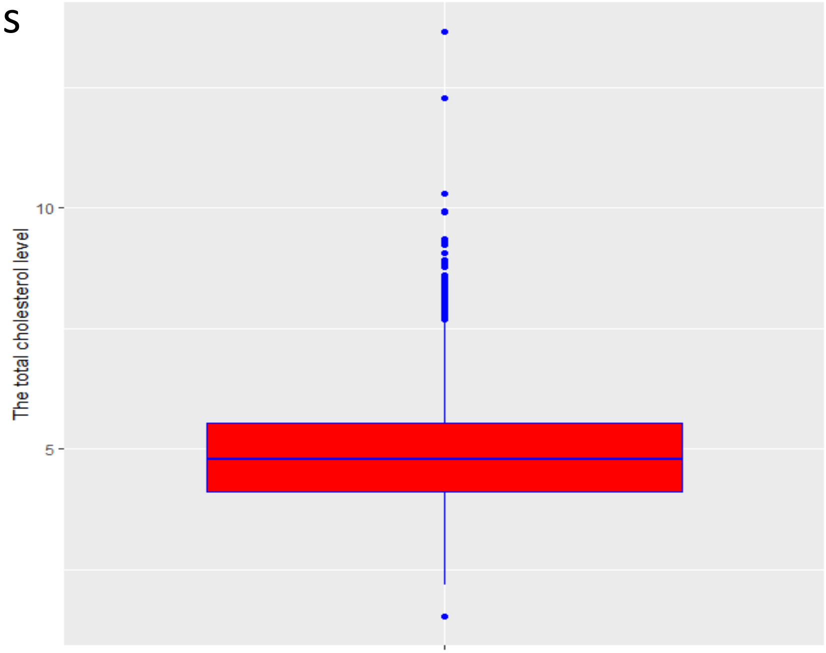


Boxplot of the total cholesterol level

```
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")
```

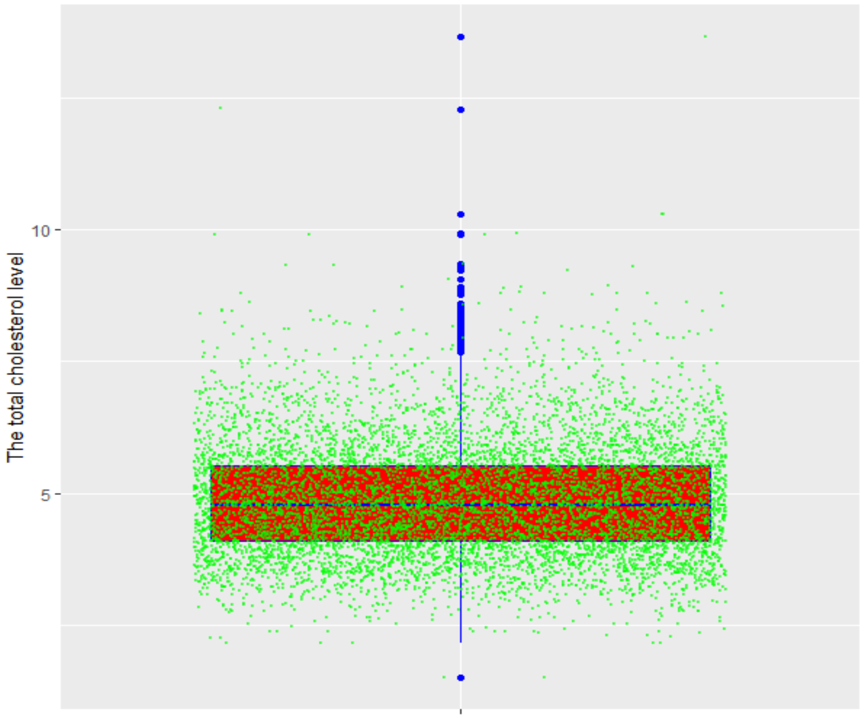


Boxplot of the total cholesterol level with data points

```
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

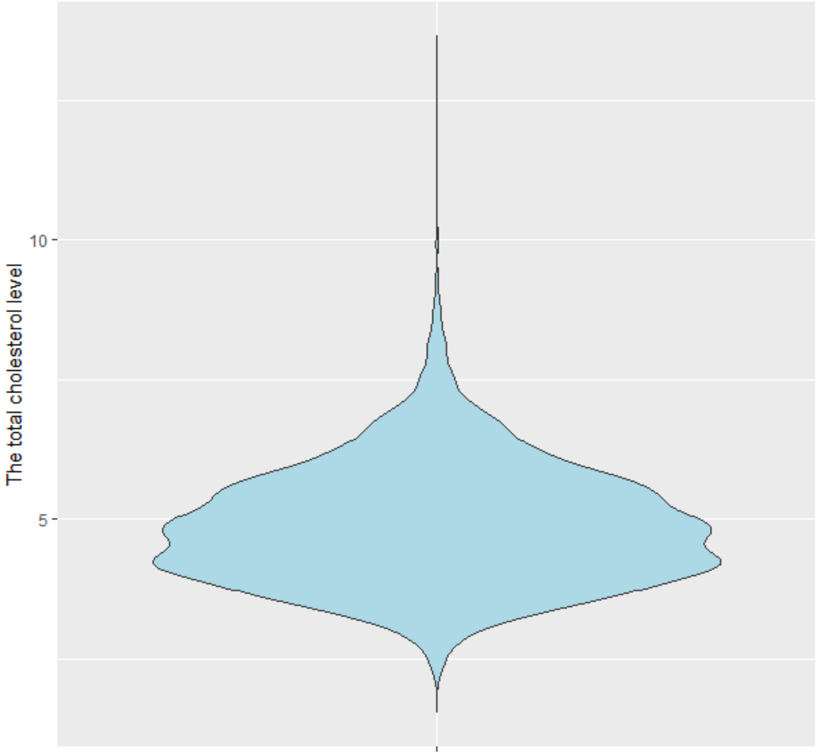
```
ggplot(NHANES, aes(x = "", y = TotChol)) +
geom_violin(fill = "lightblue")+
xlab("")+ ylab("The total cholestrol level")
```

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

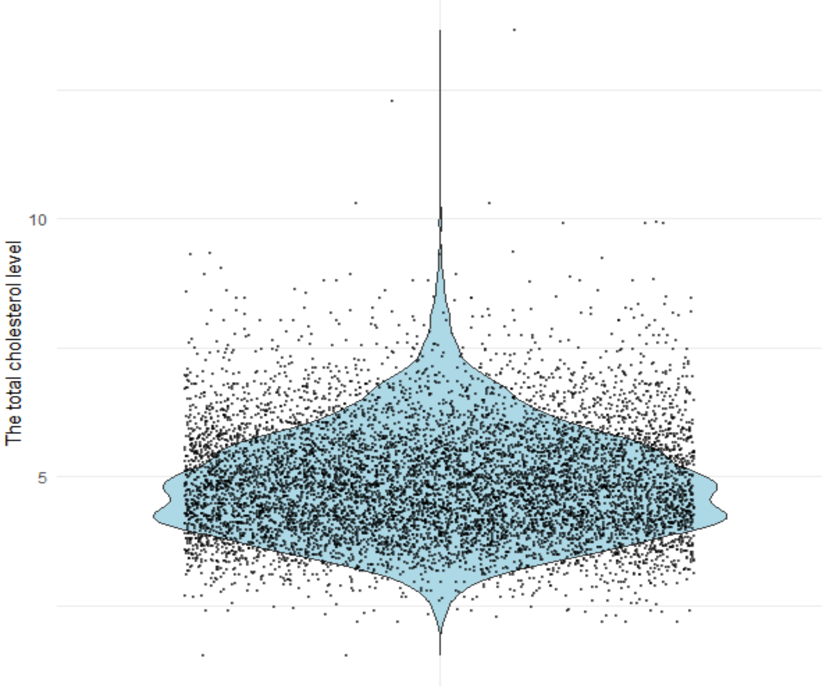
```
ggplot(NHANES, aes(x = "", y = TotChol)) +
geom_violin(fill = "lightblue")+
geom_jitter(aes(x = "", y = TotChol), color = "black",
size = 0.1, alpha = 0.5) +
xlab("")+  ylab("The total cholesterol level") +
theme_minimal()
```

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()
```


The total cholesterol level