ECO 274 LAB: Descriptive Statistics and Graphics

Learning Objectives

By the end of this lecture students will be able to learn about the:

- Measures of variability in data
- Range, Interquartile Range, Variance, and Standard Deviation
- Applications of Interquartile Range (IQR)
- Interpret the values of IQR and Standard Deviation
- Generate graphical display of data: histograms, empirical cumulative distribution, QQ-plots, box plots, bar plots, dot charts and pie charts

Measures of variability gives how "spread out" the data are.

Range: minimum & maximum

Range corresponds to biggest value minus the smallest value. It gives you the full spread of the data.

Quartiles divide the data into 4 parts. Note that, the interquartile range (IQR) - corresponding to the difference between the first and third quartiles - is sometimes used as a robust alternative to the standard deviation.

The interquartile range (IQR) measures the spread of the middle half of your data. It is the range for the middle 50% of your sample. Use the IQR to assess the variability where most of your values lie. Larger values indicate that the central portion of your data spread out further. Conversely, smaller values show that the middle values cluster more tightly.

IQR = Q3 – Q1

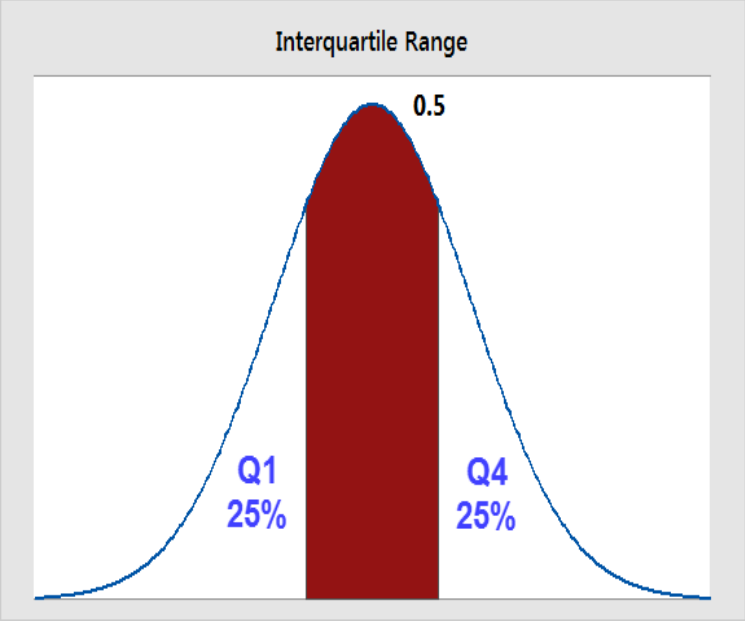
Equivalently, the interquartile range is the region between the 75th and 25th percentile (75 – 25 = 50% of the data).

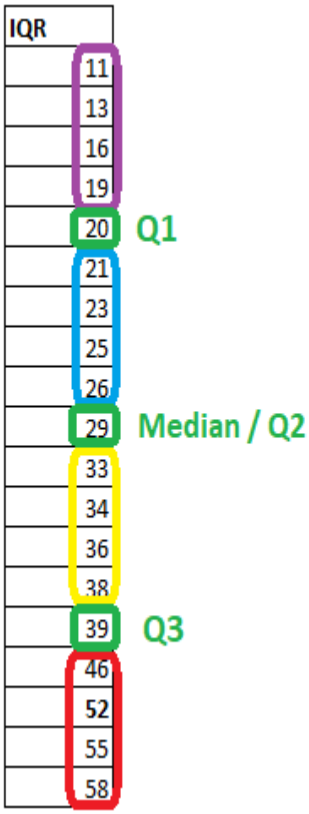
##Applications of IQR

- to measure variability,
- to assess distribution properties in boxplots graph
- to identify outliers, and
- to test whether the data is normally distributed.

```
data(airquality)
summary(Wind)
quantile(airquality$Temp)
range=max(airquality$Temp)- min(airquality$Temp)
```

```
To compute the interquartile range, type this:

IQR(airquality$Temp)

# It is also possible to obtain other quantiles; this is done by adding an
argument containing the desired percentage cut points. To get the deciles,
use the sequence function:

 pvec <- seq(0,1,0.1) #sequence of digits from 0 to 1, by 0.1

 pvec

quantile(airquality$Temp, pvec)
```

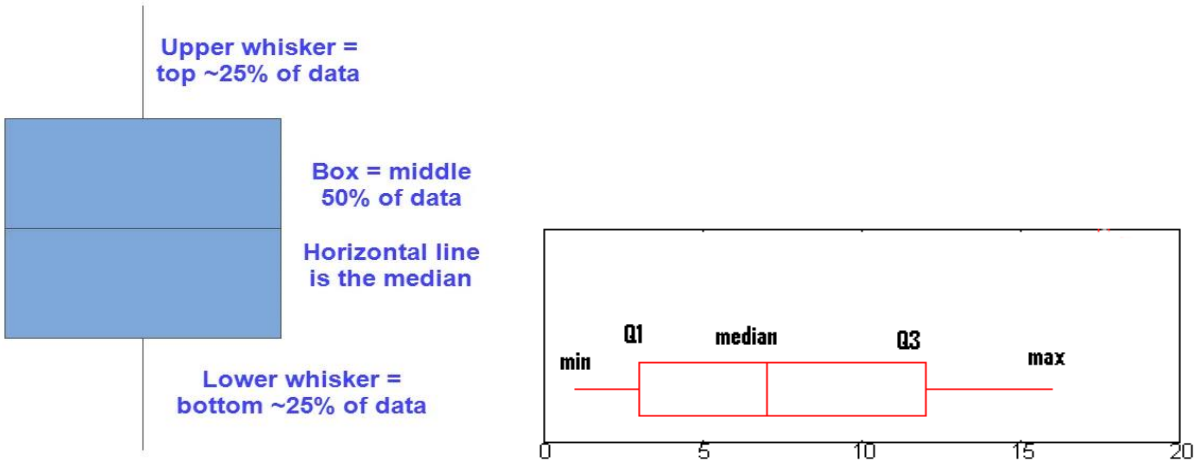




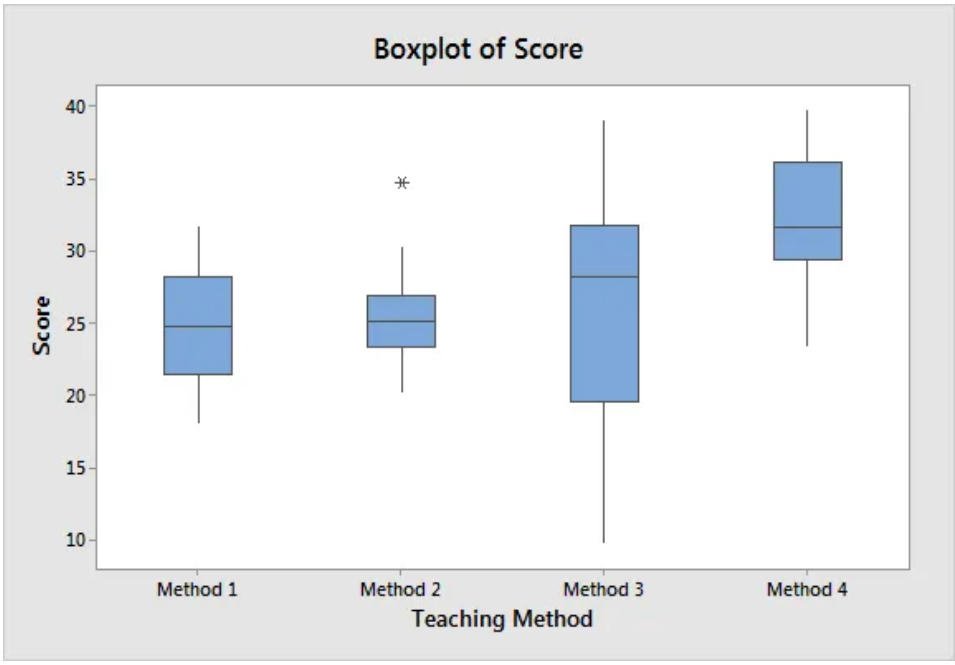
To visualize the interquartile range, imagine dividing your data into quarters. Statisticians refer to these quarters as quartiles and label them from low to high as Q1, Q2, Q3, and Q4. The lowest quartile (Q1) covers the smallest quarter of values in your dataset. The upper quartile (Q4) comprises the highest quarter of values. The IQR is the red area. We prefer using the interquartile range instead of the full data range because extreme values and outliers affect it less. Unlike the more familiar mean and standard deviation, the interquartile range and the median are robust measures. For normal distributions, you can use the standard deviation to determine the percentage of observations that fall specific distances from the mean. However, that doesn't work for skewed distributions, and the IQR is an excellent alternative.

##Using Boxplots to Graph the Interquartile Range

A box plot tells us, more or less, about the distribution of the data. It gives a sense of how much the data is actually spread about, what's its range, and about its skewness. As in the below, boxplots visualize interquartile ranges and their relation to the median and the overall distribution.



The box in the boxplot shows interquartile range! The box contains 50% of data. If the median is closer to one side or the other of the box, it's a skewed distribution. When the median is near the center of the interquartile range, your distribution is symmetric.



In the boxplot above, method 3 has the highest variability in scores and is left-skewed. Conversely, method 2 has a tighter distribution that is symmetrical, it also has an outlier (display asterisks symbol). Method 1 also has a tighter distribution that is symmetrical.

```
##Box plot
data(iris)
View(iris)
str(iris) # structure of dataset
boxplot(iris$Sepal.Length)

# Boxplots are even more informative when presented side-by-side for
comparing and contrasting distributions from two or more groups. For
instance, we compare the length of the sepal across the different species:

boxplot(iris$Sepal.Length ~ iris$Species)

library(ggplot2)
ggplot(iris) +
  aes(x = Species, y = Sepal.Length) + geom_boxplot()
```

Using the IQR to Find Outliers

To find outliers, we need to know your data's IQR, Q1, and Q3 values, and then need to calculate outlier gates.

Q1 − 1.5 * IQR: Lower outlier gate.

Q3 + 1.5 * IQR: Upper outlier gate.

Then look for values in the dataset that are below the lower gate or above the upper gate. All the values in dataset lie outside of these two benchmark gates are outliers.

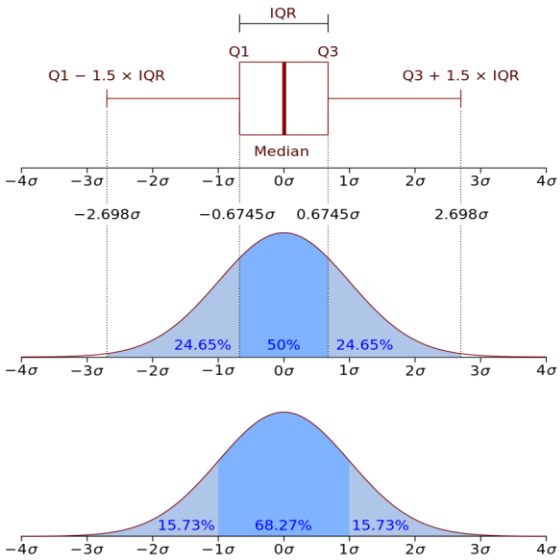
Using the Interquartile Range to Test Normality

We can use the interquartile range as a simple test to determine whether your data are normally distributed. To perform this test, we need to know the sample standard deviation (s) and sample mean ($\overline{x}$). Input these values into the formulas for Q1 and Q3 below.

$Q1 = \overline{x} - (s * 0.675)$

$Q3 = \overline{x} + (s * 0.675)$

Compare these calculated values to data's actual Q1 and Q3 values. If they are notably different, the data might not follow the normal distribution. However, this test is not an alternative to several other normality tests, e.g., a formal normality hypothesis test.

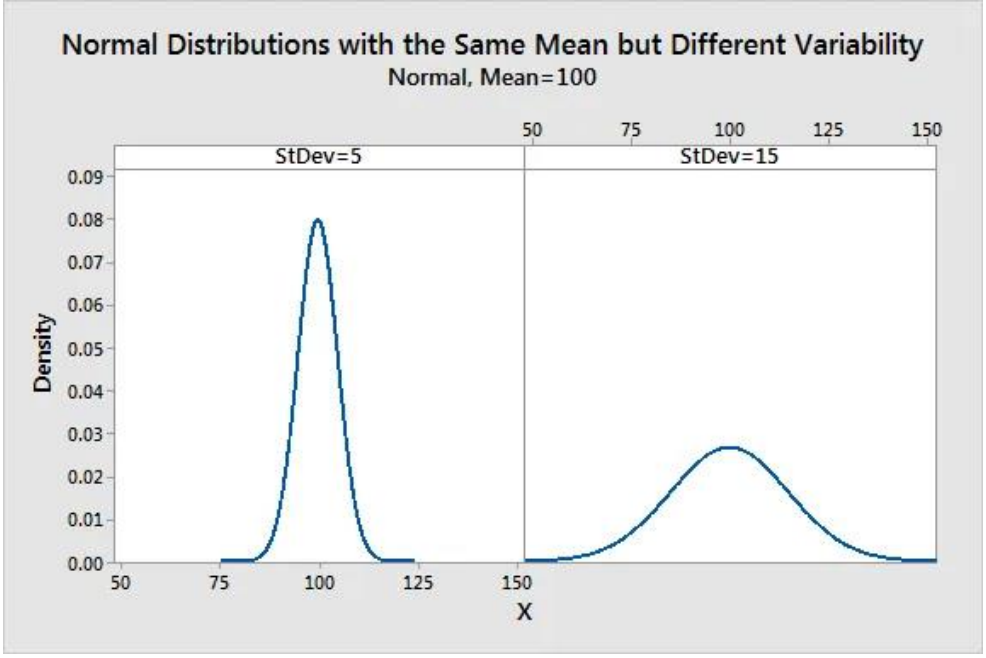


Source: https://commons.wikimedia.org/w/index.php?curid=14524285

For the values of 1.5 and 0.675, please read through the https://towardsdatascience.com/why-1-5-in-iqr-method-of-outlier-detection-5d07fdc82097. And https://math.stackexchange.com/questions/966331/why-john-tukey-set-1-5-iqr-to-detect-outliers-instead-of-1-or-2

##How the IQR compares with the other measures of variability

A measure of variability is a summary statistic that represents the amount of dispersion in a dataset. A low dispersion indicates that the data points tend to be clustered tightly around the center. High dispersion signifies that they tend to fall further away. When we have normally distributed data, or approximately so, the standard deviation becomes particularly valuable. The standard deviation is the standard or typical difference between each data point and the mean.



In statistics, variability, dispersion, and spread are synonyms that denote the width of the distribution. When a distribution has lower variability, the values in a dataset are more consistent. However, when the variability is higher, the data points are more dissimilar and extreme values become more likely.

When we are comparing samples that are the same size, consider using the range as the measure of variability. It's a reasonably intuitive statistic. Just be aware that a single outlier can throw the range off. The range is particularly suitable for small samples when we don't have enough data to calculate the other measures reliably, and the likelihood of obtaining an outlier is also lower.

When we have a skewed distribution, the median is a better measure of central tendency, and it makes sense to pair it with either the interquartile range or other percentile-based ranges because all of these statistics divide the dataset into groups with specific proportions. For normally distributed data, or even data that aren't very skewed, reporting the mean and the standard deviation is the way to go.

The main descriptive statistics in R and their graphical representation.

Histogram

A histogram gives an idea about the distribution of a quantitative variable. The idea is to break the range of values into intervals and count how many observations fall into each interval. Histograms are a bit similar to barplots, but histograms are used for quantitative variables whereas barplots are used for qualitative variables. The major difference between histograms and bar plots is that histograms are used to plot the frequency distribution of quantitative variables while bar plots are used for categorical variables.

```
hist(iris$Sepal.Length)

# Add the arguments breaks = inside the hist() function if you want to change
the number of bins. A rule of thumb (known as Sturges' law) is that the
number of bins should be the rounded value of the square root of the number
of observations. The dataset includes 150 observations so in this case the
number of bins can be set to 12.

ggplot(iris) +
  aes(x = Sepal.Length) +
  geom_histogram(color="blue", fill="white")

# By default, the number of bins is 30. You can change this value with
geom_histogram(bins = 12) for instance.

highchart() %>%
  hc_chart(type = "column") %>%
  hc_title(text = "A highcharter chart") %>%
  hc_xAxis(categories = 2012:2016) %>%
  hc_add_series(data = c(3900,  4200,  5700,  8500, 11900),
                name = "Downloads")

hchart(diamonds$cut, colorByPoint = TRUE, name = "Cut")
hchart(diamonds$price, color = "#B71C1C", name = "Price") %>%
  hc_title(text = "You can zoom me")

## Bar plot using mpg data

ggplot(mpg, aes(x=factor(cyl)))+ geom_bar(col="red",fill="green", alpha =
.2,stat="count")
```

 # Dotplot

#A dotplot is more or less similar than a boxplot, except that observations are represented as points and there is no summary statistics presented on the plot:

```
library(lattice)

dotplot(iris$Sepal.Length ~ iris$Species)
```

#Scatterplot

Scatterplots allow to check whether there is a potential link between two quantitative variables. For this reason, scatterplots are often used to visualize a potential correlation between two variables. For instance, we can draw a scatterplot of the length of the sepal and the length of the petal:

```


plot(iris$Sepal.Length, iris$Petal.Length)
```

```
ggplot(iris) +
  aes(x = Sepal.Length, y = Petal.Length) +
  geom_point()

##scatterplots are even more informative when differentiating the points
according to a factor, in this case the species:

ggplot(iris) +
  aes(x = Sepal.Length, y = Petal.Length, colour = Species) +
  geom_point() +
  scale_color_hue()
library("highcharter")
data(diamonds, mpg, package = "ggplot2")

View(diamonds)
View(mpg)
hchart(mpg, "scatter", hcaes(x = displ, y = hwy, group = class))
```

QQ-plot

For a single variable

In order to check the normality assumption of a variable (normality means that the data follow a normal distribution, also known as a Gaussian distribution), we usually use histograms and/or QQ-plots.

```
# Draw points on the qq-plot:
qqnorm(dat$Sepal.Length)

# Draw the reference line:
qqline(dat$Sepal.Length)

Or a QQ-plot with confidence bands with the qqPlot() function from the {car}
package:

library(car) # package must be installed first
qqPlot(iris$Sepal.Length)

library(ggpubr)
ggqqplot(iris$Sepal.Length)

# If points are close to the reference line (sometimes referred as Henry's
line) and within the confidence bands, the normality assumption can be
considered as met. The bigger the deviation between the points and the
reference line and the more they lie outside the confidence bands, the less
likely that the normality condition is met. When facing a non-normal
distribution, the first step is usually to apply the logarithm transformation
on the data and recheck to see whether the log-transformed data are normally
distributed.
```

Density plot

Density plot is a smoothed version of the histogram and is used in the same concept, that is, to represent the distribution of a numeric variable. The functions plot() and density() are used together to draw a density plot:

```
plot(density(dat$Sepal.Length))
ggplot(dat) +
  aes(x = Sepal.Length) +
  geom_density()
```

Normal distribution

The **normal distribution which is also called Gaussian distribution** is defined by the following probability density function, where μ is the population mean and σ^2 is the variance.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

If a random variable X follows the normal distribution, then we write:

$$X \sim N(\mu, \sigma^2)$$

In particular, the normal distribution with $\mu = 0$ and $\sigma = 1$ is called the ***standard normal distribution***, and is denoted as $N(0,1)$.

The normal distribution is important because of the **Central Limit Theorem**, which states that the population of all possible samples of size n from a population with mean μ and variance σ^2 approaches a normal distribution with mean μ and σ^2/n when n approaches infinity.

To characterize the distribution of a continuous random variable, we can use the probability density function (pdf) .

```
# To get a better understanding on the shape of the normal pdf, let's
visualize the pdf of N(0,1).

df <- seq(from = -5, to = 5, by = 0.05)
norm_dat <- data.frame(df, pdf = dnorm(df))
ggplot(norm_dat) + geom_line(aes(x = df, y = pdf))

#To see the shape for the defaults (mean 0, standard deviation 1)

x <- rnorm(100)
hist(x,probability=TRUE,col="green",main="normal distribuition with
mean=0,sigma=1")
```

```
curve(dnorm(x),add=T, col="red", lwd=4)

# Next, you can take a step further to visualize three different normal
distributions in the same plot, N(0,1), N(1,4), and N(−1,0.25)

norm_dat_1 <- data.frame(dist = "N(0,1)", x = df, pdf = dnorm(df))
norm_dat_2 <- data.frame(dist = "N(1,4)", x = df, pdf = dnorm(df, mean = 1,
sd = 2))
norm_dat_3 <- data.frame(dist = "N(-1, 0.25)", x = df, pdf = dnorm(df, mean =
-1, sd = 0.5))
norm_dat <- rbind(norm_dat_1, norm_dat_2, norm_dat_3)
ggplot(norm_dat) + geom_line(aes(x = x, y = pdf, color = dist))


curve(dnorm(x,0,1),-10,10,lwd=1,lty=1,col="red")
curve(dnorm(x,0,2),add=T,lwd=2,lty=2,col="blue")
curve(dnorm(x,0,3),add=T,lwd=3,lty=3,col="green")
legend("topright",c("sigma1","sigma2","sigma3"),lwd=1:3,lty=1:3)
legend("topright",c("sigma1","sigma2","sigma3"),lwd=1:3,lty=1:3,col =
c("red","blue","green"))


curve(dnorm(x,0,1),-10,10,lwd=1,lty=1,col="red")
curve(dnorm(x,0,2),add=T,lwd=2,lty=2,col="blue")
curve(dnorm(x,0,3),add=T,lwd=3,lty=3,col="green")
legend("topleft",expression(sigma==1,sigma==2,sigma==3),lwd=1:3,lty=1:3,col =
c("red","blue","green"))
text(6,0.3, expression(f(x)==frac(1,sqrt(2*pi)*sigma)*e^{-
frac(x^2,2*sigma^2)}))
```

CDF

In addition to pdf, you can compute the cumulative distribution function (cdf) of the normal distribution using the function. The CDF of a variable X, or just distribution function of X, is essentially just a representation of the probability that X will take a value less than or equal to x.

$$F_x(x) = P(X <= x)$$

The unique thing about a CDF is that it is monotonic. More specifically, monotonic increasing. # read more about cdf on : https://towardsdatascience.com/what-is-a-cumulative-distribution-function-2e0540ec2a60.

```
q <- seq(from = -5, to = 5, by = 0.1)
norm_dat <- data.frame(q = q, cdf = pnorm(q))
ggplot(norm_dat) + geom_line(aes(x = q, y = cdf))
```

Central Limit Theorem

The normal distribution is important because of the Central Limit Theorem, which states that the population of all possible samples of size n from a population with mean μ and variance σ2 approaches a normal distribution with mean μ and σ2/n when n approaches infinity.

```
#Importing the  Clt-data csv data
data<-read.csv(file.choose()) # Import Clt-data set


#Count of Rows and columns
dim(data)

#View top 10 rows of the dataset
head(data,10)

tail(data,15)


#Step 3 - Calculate the population mean and plot the observations


#Calculate the population mean
mean(data$ Wall.Thickness)

#Plot all the observations in the data
hist(data$Wall.Thickness,col = "slategray2",main = "Histogram for Wall
Thickness",xlab = "wall thickness")
abline(v=12.8,col="red",lty=12)


#Now we take sample size=10, samples=9000
#Calculate the arithmetice mean and plot the mean of sample 9000 times

s10<-c()
n=9000
for (i in 1:n) {
  s10[i] = mean(sample(data$Wall.Thickness,10, replace = TRUE))}
hist(s10, col ="lightgreen", main="Sample size =10",xlab = "wall thickness")
abline(v = mean(s10), col = "Red")
abline(v = 12.8, col = "blue")


#Now, we know that we can get a very nice bell-shaped curve as the sample
sizes increase.

#We will take sample size=30, 50 & 500 samples=9000
#Calculate the arithmetice mean and plot the mean of sample 9000 times

s30 <- c()
s50 <- c()
s500 <- c()
n =9000
```

```
for ( i in 1:n){
  s30[i] = mean(sample(data$Wall.Thickness,30, replace = TRUE))
  s50[i] = mean(sample(data$Wall.Thickness,50, replace = TRUE))
  s500[i] = mean(sample(data$Wall.Thickness,500, replace = TRUE))
}

#create plot matrix

par(mfrow=c(1,3))


hist(s30, col ="lightblue",main="Sample size=30",xlab ="wall thickness")
abline(v = mean(s30), col = "red")

hist(s50, col ="lightgreen", main="Sample size=50",xlab ="wall thickness")
abline(v = mean(s50), col = "red")

hist(s500, col ="orange",main="Sample size=500",xlab ="wall thickness")
abline(v = mean(s500), col = "red")

#Here, we get a good bell-shaped curve and the sampling distribution
#approaches normal distribution as the sample sizes increase.
```

Sources Acknowledged:

1. https://statisticsglobe.com; towardsdatascience.com;

2. https://statisticsbyjim.com/basics

3. Harvard Chan Bioinformatics Core, R for data science lab, Medium write up.