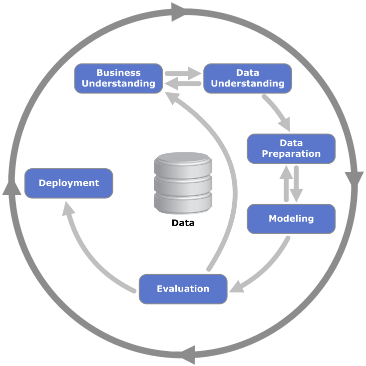
Machine learning



1. Problem Understanding lub Business Understanding
2. Data Understanding
3. Data Preparation
4. Modeling
5. Evaluation
6. Deployment

All **R** functions and datasets are stored in packages. Only when a package is loaded are its contents available.

The standard (or base) packages are considered part of the R source code. They contain the **basic functions** that allow R to work, and the **datasets** and **standard statistical and graphical functions**. They should be automatically available in any R installation.

To install a particular package use a command like

```
install.packages("name_of_package")
```

to load

```
library(name_of_package)
```

Packages - namespaces

- packages can have namespaces, and currently all of the base and recommended packages do except the datasets package.
- namespaces do three things:
  - allow the package writer to hide functions and data that are meant only for internal use;
  - they prevent functions from breaking when a user (or other package writer) picks a name that clashes with one in the package;
  - provide a way to refer to an object within a particular package.

For example, **t()** is the transpose function in R, but users might define their own function named t. Namespaces prevent the user's definition from taking precedence, and breaking every function that tries to transpose a matrix.

The double-colon operator :: selects definitions from a particular namespace.

In the example above, the transpose function will always be available as **base::t**, because it is defined in the base package.

- **The tidyverse** is a collection of **R** packages designed to facilitate the entire analytics process by offering a standardized format for exchanging data between packages.
- It includes packages designed to import, manipulate, visualize, and model data with a series of functions that easily work across different tidyverse packages.

tidyverse

```
install.packages("tidyverse")
library(tidyverse)
```

The following are the major packages that make up the tidyverse:

- **readr** for importing data into R from a variety of file formats
- **tibble** for storing data in a standardized format
- **dplyr** for manipulating data
- **ggplot2** for visualizing data
- **tidyr** for transforming data into "tidy" form
- **purrr** for functional programming
- **stringr** for manipulating strings
- **lubridate** for manipulating dates and times

- **The readr package** is the package that allows you to import data from a standard file format into R.
- **The readr functions** load a file that is stored on disk or at a URL and imports it into a tidyverse-friendly data structure known as a tibble (more on tibbles later on lab).

Comma-delimited files are the most common way to exchange data between different environments.

These files, which are also known as comma-separated value (CSV) files, store data in a simple, standardized format that may be imported or exported from almost any source.

Name	Age	Gender	ZIP
Mary	27	F	11579
Tom	32	M	07753
Beth	43	F	46556

```
Name,Age,Gender,ZIP
Mary,27,F,11579
Tom,32,M,07753
Beth,43,F,46556
```

We can read CSV files into R using the read_csv() function from the readr package.

A few of the most important arguments

- **file**
- **col_names** specifies where R should obtain the names of the variables used in the dataset.
  - The default value for col_names is TRUE, which indicates that R should use the values appearing in the first line of the CSV file as the variable names.
  - If this value is set to FALSE, R will generate its own column names using the sequentially numbered format X1, X2, X3, and so on.
- **col_types** specifies the data types for the columns. If you do not include this argument, R will guess the appropriate data types based on the values in the file.

l for logical, **n** for numeric, **i** for integers, **c** for characters,
f for factors, **D** for dates, **T** for datetimes

Tibbles

The **tibble** is a modern version of the **R** data frame implemented as part of the tidyverse.

The tibble frame

- enables efficient data processing
- clearly displays information about variables from the dataset
- makes it easier to work with and output large datasets to the screen without overwhelming system.

The read_csv() function from the readr package reads input data directly into a tibble.

the glimpse() function

The glimpse() function shows us

- the number of observations or rows in the data,
- the number of variables or columns in the data,
- the variable names, the data types, and a sample of the data stored in each variable.

Descriptive Statistics

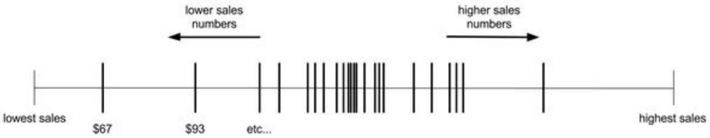
Data Mining Map. http://www.saedsayad.com/data_mining_map.htm

Statistical fundamentals and terminology

The field of **statistics** helps us to gain an understanding of our data, and to quantify what our data and results look like.
It also provides us with mechanisms to measure how well our application is performing and prevent certain machine learning pitfalls (such as under/overfitting).

Distribution

A **distribution** is a representation of how often values appear within a dataset.



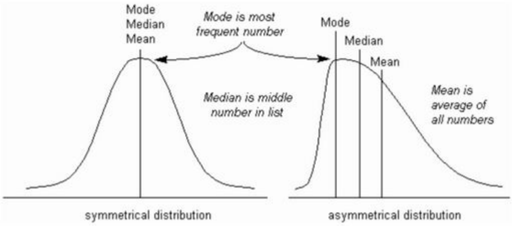
There are two types of these measures:

- **central tendency measures** - these measure where most of the values are located, or where the center of the distribution is located;
- **spread or dispersion measures** - these measure how the values of the distribution are spread across the distribution's range (from the lowest value to the highest value).

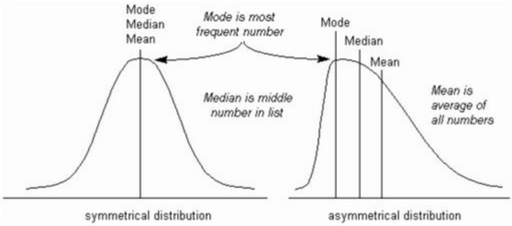
Measures of central tendency include the following:

- **Mean** - this is what you might commonly refer to as an average. We calculate this by summing all of the numbers in the distribution and then dividing by the count of the numbers. The mean of a sample $x_1, x_2, \ldots, x_n$, usually denoted by $\bar{x}$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

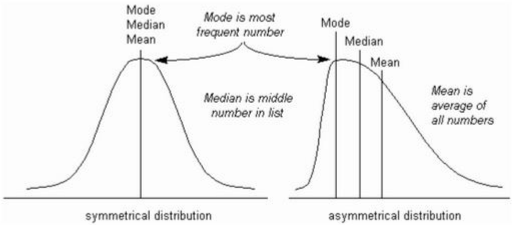


Measures of central tendency



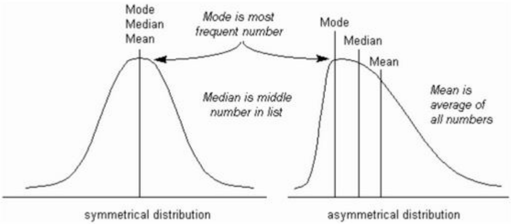
- **Median** - if we sort all of the numbers in our distribution from the lowest to highest, this is the number that separates the lowest half of the numbers from the highest half of the numbers. The median of a sample $x_1, x_2, \ldots, x_n$, where $x_1 \leq x_2 \leq \ldots \leq x_n$

$$m_e = \begin{cases} x_{(n+1)/2} & n \text{ is odd,} \\ \frac{1}{2}\left(x_{(n/2)} + x_{(n/2+1)}\right) & \end{cases}$$



- **Mode** - this is the most frequently occurring value in the distribution.

Measures of central tendency



- If the **mean** and **median** are significantly different, then we can expect that some observations for a given distribution are far from the mean. These are **outlier points**.

Example summary of distribution of variable **mileage:**

```
summary(usedcars[c("mileage")])
Min.   : 4867
1st Qu.: 27200
Median : 36385
Mean   : 44261
3rd Qu.: 55125
Max.   :151479
```

- let us note that the **mean= 44 261** is about 20 per cent higher than the **median = 36 385**;
- The mean is more sensitive than the median to extreme values, so in our case we can observe that there are cars with very high mileage in the dataset.

Measures of dispersion include the following:

- **Maximum** - the highest value of the distribution
- **Minimum** - the lowest value of the distribution
- **Range** - the difference between the maximum and minimum
- **Variance** - this measure is calculated by taking each of the values in the distribution, calculating each one's difference from the distribution's mean, squaring this difference, adding it to the other squared differences, and dividing by the number of values in the distribution
- **Standard deviation** - the square root of the variance
- **Quantiles/quartiles** - similar to the median, these measures define cut-off points in the distribution where a certain number of lower values are below the measure and a certain number of higher values are above the measure

Variance

The variance of a sample $x_1, x_2, \ldots, x_n$ is of the form

$$s^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}$$

where $\bar{x}$ is the mean value.

The standard deviation is calculating a sort of average distance of how far the data values are from the arithmetic mean.

- by taking $x - \bar{x}$, you are finding the literal difference between the value and the mean of the sample;
- by squaring the result, $(x_i - \bar{x})^2$, we are putting a **greater penalty on outliers** because squaring a large error only makes it much larger.
- by dividing by the number of items in the sample, we are taking (literally) the average squared distance between each point and the mean.

The standard deviation of a sample $x_1, x_2, \ldots, x_n$ is of the form

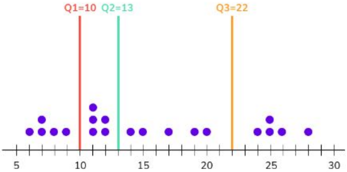
$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}}$$

where $\bar{x}$ is the mean value.

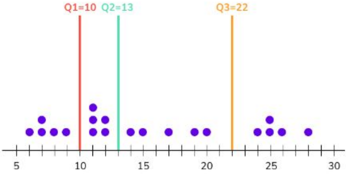
A low standard deviation indicates that the values tend to be close to the mean of the set, while a high standard deviation indicates that the values are spread out over a wider range.

A **quartile** divides the number of sorted data points into four parts, or quarters, of more-or-less equal size. The three main quartiles are as follows:

- **the first quartile** Q_1 is defined as the middle number between the smallest number (minimum) and the median of the data set. It is also known as the lower or 25th empirical quartile, as 25% of the data is below this point.
- **the second quartile** Q_2 is the median of a data set; thus 50% of the data lies below this point.
- **the third quartile** Q_3 is the middle value between the median and the highest value (maximum) of the data set. It is known as the upper or 75th empirical quartile, as 75% of the data lies below this point.



Quartile




```
x <- c(6,7,7,8,9,11,11,11,12,12,14,15,17,19,20,24,25,25,26,28)
mean(x, na.rm = TRUE)
median(x, na.rm = TRUE)
sd(x, na.rm = TRUE)
min(x, na.rm = TRUE)
max(x, na.rm = TRUE)
quantile(x, na.rm = TRUE)
```

Quantiles

q-quantiles are values that partition a finite set of values into q subsets of (nearly) equal sizes.

- In statistics and probability, quantiles are cut points dividing the range of a probability distribution into continuous intervals with equal probabilities, or dividing the observations in a sample in the same way.
- Common quantiles have special names, such as
  - quartiles (four groups),
  - quintiles (five groups),
  - deciles (ten groups),
  - percentiles (100 groups).

the summary() function

In R, we can get summary statistics for a dataset by using the summary() function.

The results of function show two different formats for the descriptive statistics:

- **for categorical features** - shows the feature values along with the frequency for each value.
- **for continuous features** - shows the mean, median, minimum, maximum, first and third quartile values and number of missing values (NAs) for the features.

dplyr package

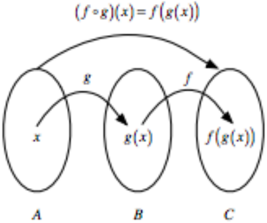
Recall that dplyr is a package in the tidyverse that is used for data exploration and manipulation.

It provides five main commands

- **select** for choosing the columns or variables
- **filter** for choosing rows or observations
- **arrange** for sorting rows
- **mutate** for modifying variables
- **group_by** for finding groups
- **summarize** for aggregating rows

If we had to perform a large number of operations where each successive function relied on the output of the previous one for its input, our code would quickly become difficult to read.

The **pipe operator** is is used to control **the logical flow of the code**. Pipes is written as $\% > \%$. It is provided by the magrittr package, which is loaded as part of the tidyverse.




```
pi %>% sin %>% cos
cos(sin(pi))
```

Description of the data

- (previous) **numerical summarization** of the data - allows us to better understand it
- (next) **data visualization** - allows us quickly understand way of describing data

A picture is worth a thousand words.

Visualizations serve as a great tool for asking and answering questions about data.

Depending on the type of question, there are four key objectives that inform the type of data visualization we use:

- comparison
- relationship
- distribution
- composition

In R, there are three main plotting systems:

- base graphics
- Lattice
- ggplot2

Visualizing the Data - ggplot2

- **ggplot2** is the most popular system among data scientists
- it is a part of the tidyverse
- the **gg** in ggplot2 stands for grammar of graphics, a school of thought that says any data graphic can be created by combining data with layers of plot components such as axes, tickmarks, gridlines, dots, bars, and lines.
- the + **operator** adds successive layers

- A comparison visualization is used to illustrate the difference between two or more items at a given point in time or over a period of time.
- A commonly used comparison chart is the **boxplot.**
- Boxplots are typically used to compare the distribution of a continuous feature against the values of a categorical feature.

Boxplot visualizes the five summary statistics (minimum, first quartile, median, third quartile, and maximum) and all outlying points individually.

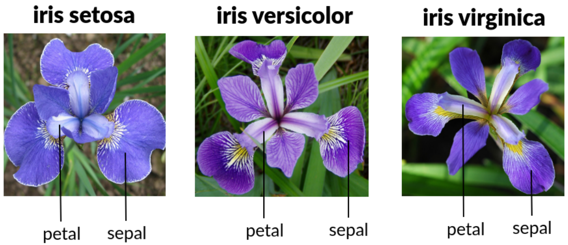
A common visualization of the five-number summary is a **boxplot**, also known as a **box-and-whisker** plot. The boxplot displays the center and spread of a numeric variable in a format that allows you to quickly obtain a sense of its range and compare it to other features.

Visualizing the Data - relationship

- Relationship visualizations are used to illustrate the correlation between two or more variables.
- **Scatterplots** are one of the most commonly used relationship visualizations.
- These are typically for both continuous features.

The **plot()** function is one of most powerful functions in base R.

```
summary(iris)
```




```
plot(iris$Sepal.Width)
plot(iris$Sepal.Width,type = "b")
plot(x = iris$Petal.Length, y = iris$Petal.Width)
```

Visualizing the Data - distribution

- Distribution visualizations show the statistical distribution of the values of a feature.
- One of the most commonly used distribution visualizations is the histogram.
- With a histogram you can show the **spread** and **skewness** of data for a particular feature

- Histogram divides the feature values into a predefined number of portions or bins that act as containers for values, with the same range.
- Histogram is composed of a series of bars with heights indicating the count, or frequency, of values falling within each of the equal-width bins partitioning the values.

- A composition visualization shows the component makeup of the data.
- Stacked bar charts and pie charts are two of the most commonly used composition visualizations.
- With a stacked bar chart, you can show how a total value can be divided into parts or highlight the significance of each part relative to the total value.

Thank you for your attention!!!