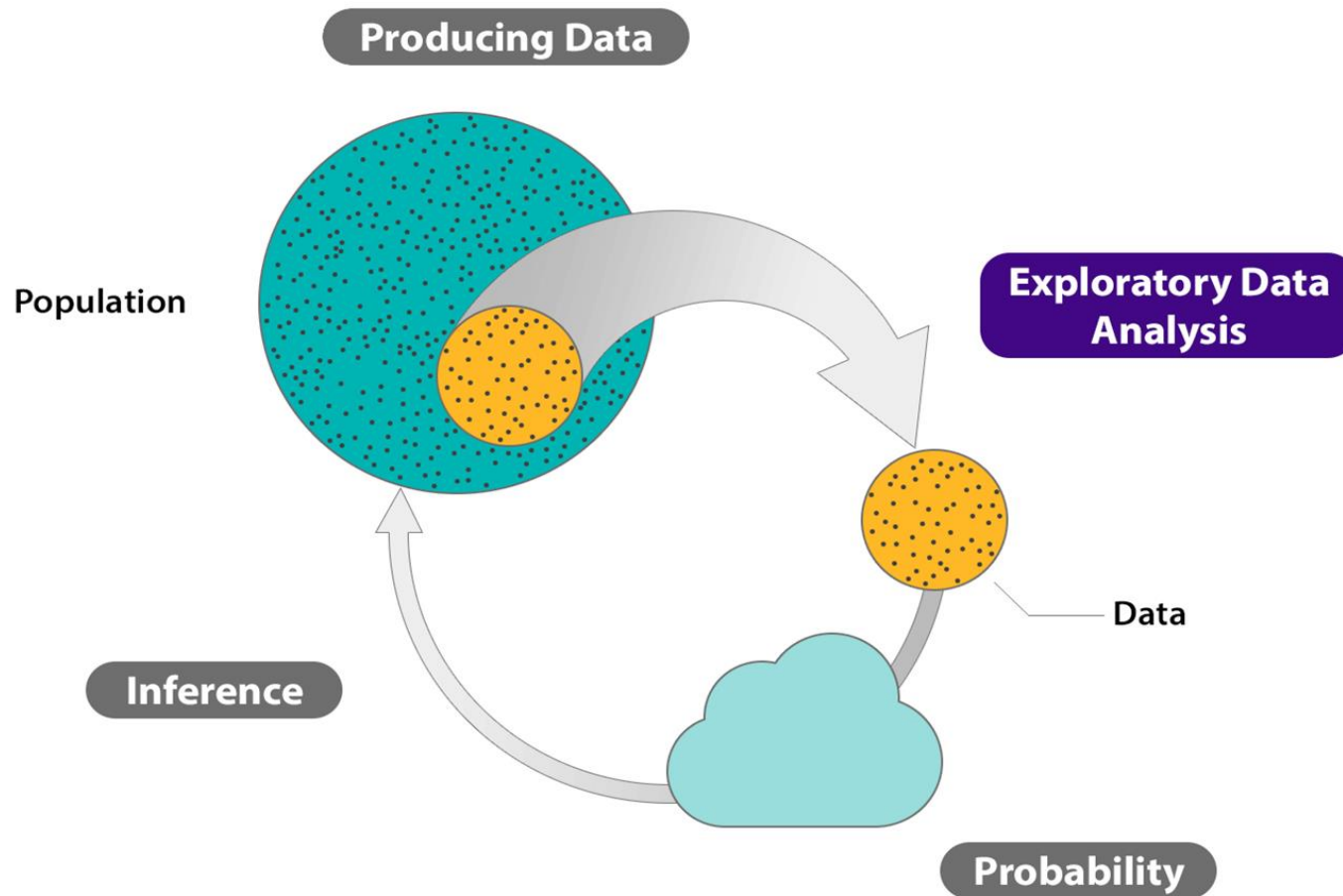




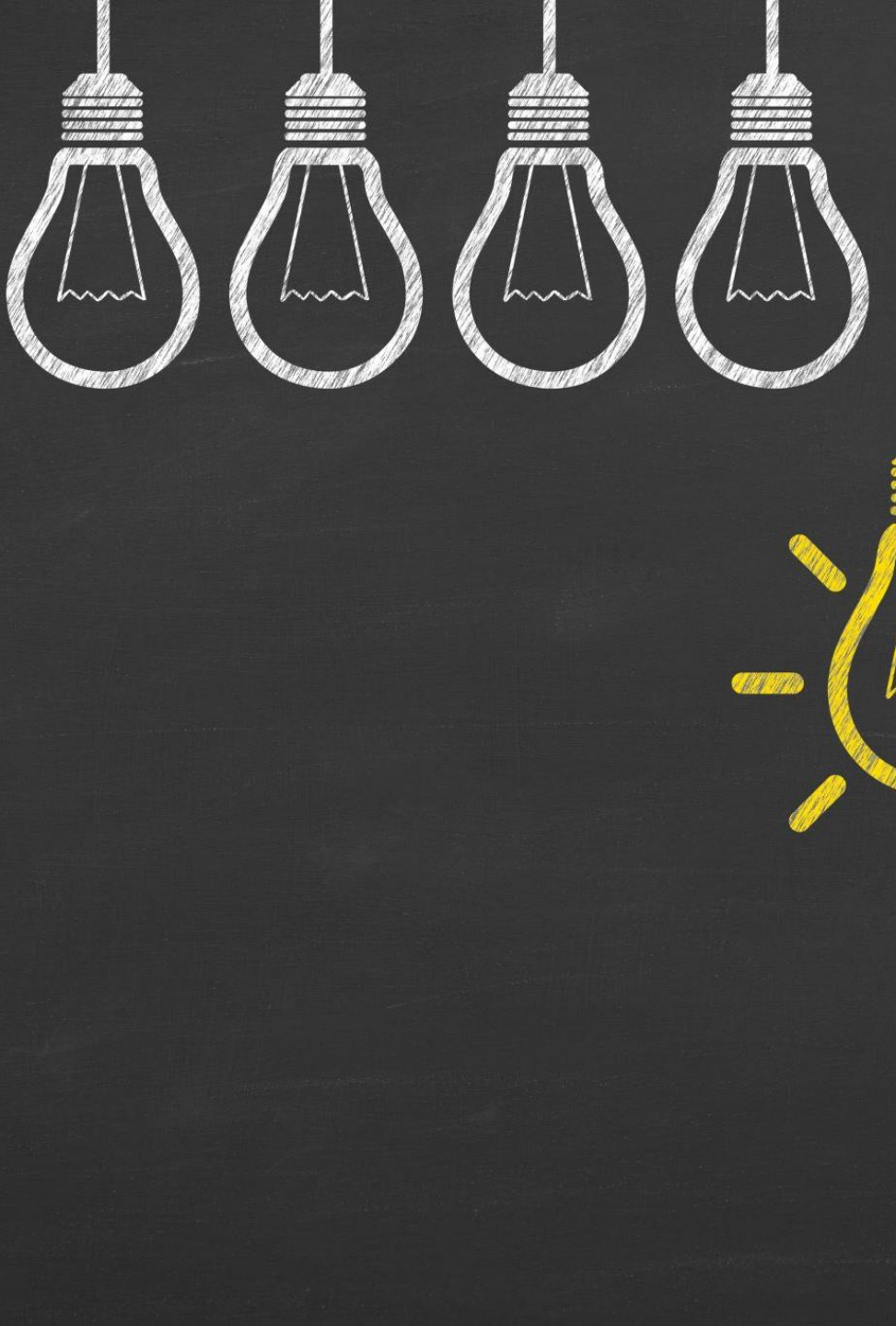
# Data Analysis

“The data scientist is concerned primarily with the data, the insights which can be extracted from it and the stories that we can tell.”\*

\*This reference was provided in an article “Data sciences, data engineer and other data careers explained” by Matthew Mayo



**Exploratory Data Analysis:** an approach to analyzing data sets to summarize their main characteristics, often with visual methods. - Wikipedia



# Learning objectives:

---

- Explain the significance of Exploratory Data Analysis and Data Science
- Demonstrate in what way Exploratory Data Analysis involves the following tasks:
  - Organizing and summarizing the raw data
  - Gaining valuable hints for data cleaning
  - Determining important features and patterns in the data and any unusual nonconformities from those patterns
  - Interpreting results in the context of the problem
- Learn how to invoke some basic EDA methods effectively, in order to understand datasets and prepare for more advanced analysis.

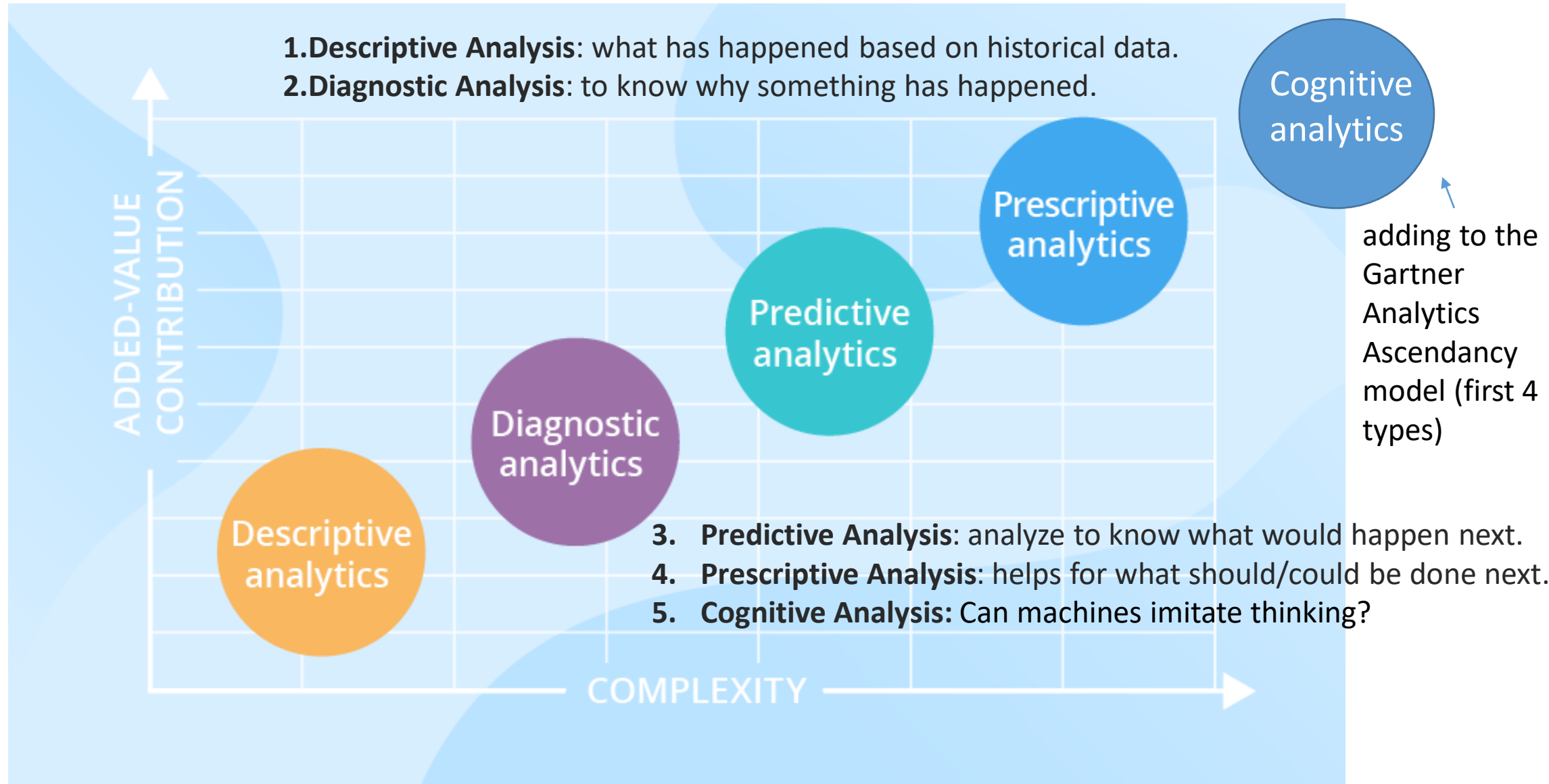
Article read: click [here](#)

## How Visualization is Transforming Exploratory Data Analysis



By [Todd Mostak](#), CEO at OmniSci

Let's recall from the Introduction to Data Science lecture: The Data Analysis categories



# Two Branches of Statistics

## Descriptive Statistics

- as a science, involves the collection, organization, summarization, and presentation of data
- involves raw data, as well as graphs, tables, and numerical summaries
- “Just the facts”
- Refer to sample without making any assumptions about the population

## Inferential Statistics

- as a science, involves using descriptive statistics to estimate population parameters
- deals with interpretation of the information collected
- usually used in conjunction with descriptive statistics within a statistical study





# “Get to Know” the dataset

- Doing so upfront will make the rest of the project much smoother, in 3 main ways:
  1. You’ll gain valuable hints for [Data Cleaning](#).
  2. You’ll think of ideas for [Feature Engineering](#).
  3. You’ll get a "feel" for the dataset, which will help you communicate results and deliver greater impact.
- EDA should be **quick, efficient, and decisive**... not long and drawn out!
- You see, there are infinite possible plots, charts, and tables, but you only need a **handful** to "get to know" the data well enough to work with it.



# What is EDA?

An approach for data analysis that employs a variety of techniques

1. Maximize insight into a data set
2. Uncover underlying structure
3. Extract important variables
4. Detect outliers and anomalies
5. Test underlying assumptions
6. Develop parsimonious models and
7. Determine optimal factor settings



# EDA is a data approach.

The EDA sequence is:

Problem => Data => Analysis=> Model=> Conclusions

As opposed for a classical approach:

Problem => Data => Model=> Analysis=> Conclusions

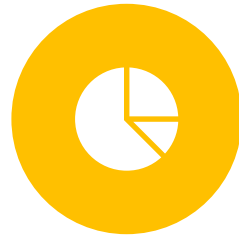
# Steps involved in EDA



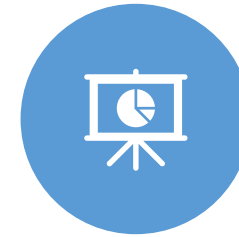
ORGANIZE AND  
SUMMARIZE  
YOUR DATA ON A  
DATA CODING  
SHEET.



IF DESIRED,  
ORGANIZE DATA  
FOR COMPUTER  
ENTRY.



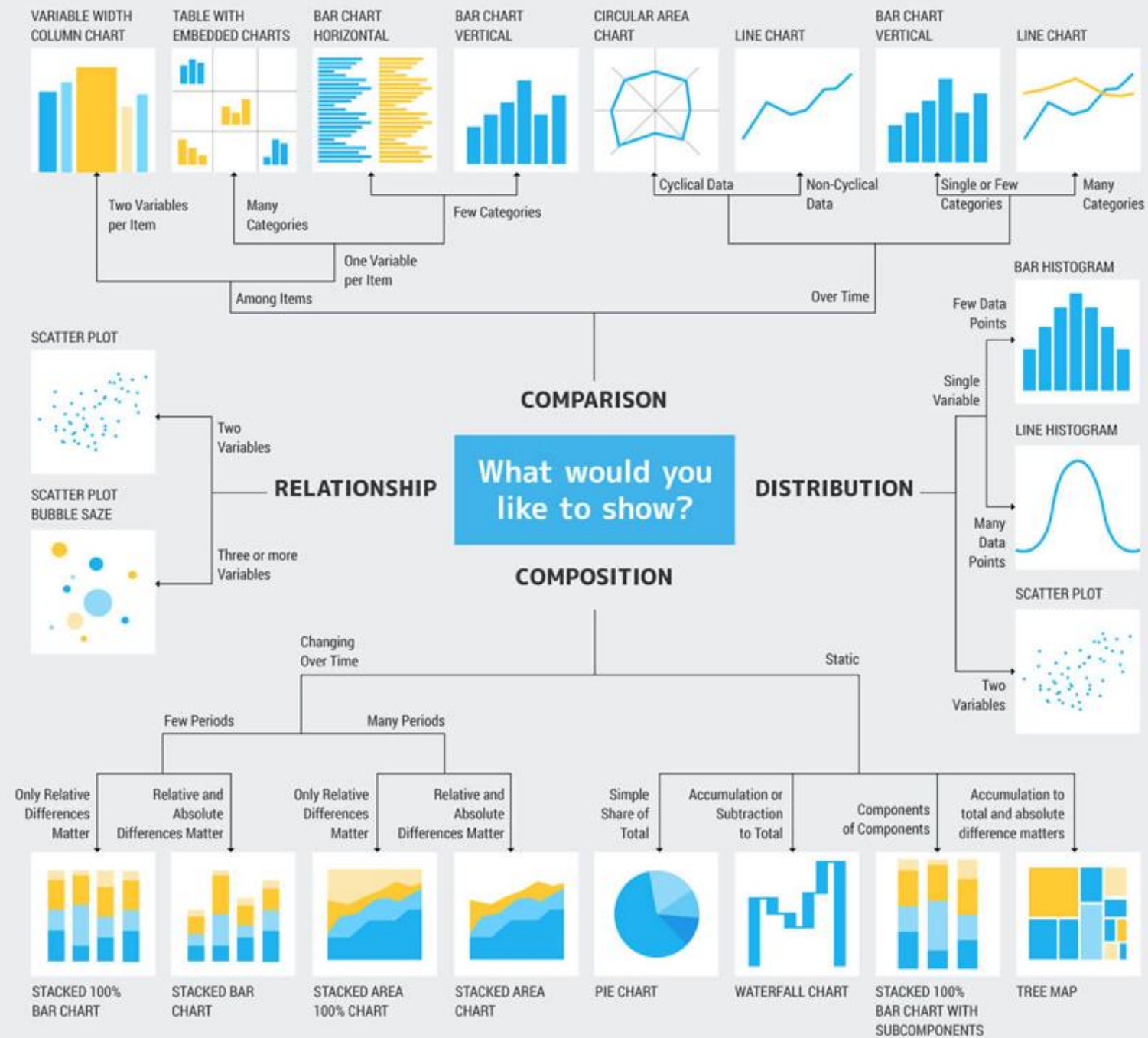
GRAPH DATA  
(BAR GRAPH,  
HISTOGRAM,  
LINE GRAPH, OR  
SCATTERPLOT)  
SO THAT YOU  
CAN VISUALLY  
INSPECT  
DISTRIBUTIONS.



DISPLAY  
FREQUENCY  
DISTRIBUTIONS  
ON A  
HISTOGRAM,  
AND CREATE A  
STEMPLOT  
(STEM & LEAF)



EXAMINE YOUR  
GRAPHS FOR  
NORMALITY OR  
SKEWNESS IN  
YOUR  
DISTRIBUTIONS.



# 5 Main Usages for Distinction of Chart Types

- **Presenting Distribution.**

Distribution charts lay out how items are distributed to different parts. The best chart to use for this type of data are line charts, histogram chart and scatter charts which illustrating items correlation, among others.

- **Visualizing the compositions.**

Three types of charts benefit for visualizing the composition of an issue. It is obvious that pie charts are designed to show the compositions as different parts of a pie can represent one composition and whole pie is the completion of an item. Area charts and stacked bar charts can also be visualized by different color areas for visualizing compositions.

- **Showing the relationship.**

Among all the data, it is of vital importance to find the relationships toward data. Spider charts and bubble charts are the perfect choices of analyzing the relationship as they contain relationship of one data variable to the whole group or other variables.

- **Indicating the trend.**

When you need a chart for indicating the trend of a series data in a fixed period, there are two basic charts proper for you – Column Chart and Line Chart. Both two charts show the changing trend with differences of data. You can also combine them into one chart named Pareto chart which is better for knowing relevant trends.

- **Comparing.**

Most charts are created aiming to data comparison. Comparing function helps you visualize mass data as data always appear with a large quantity. Divided into two parts, different items could be compared by bar chart, column chart, six sigma chart and spider chart while one fixed item can be compared through different time like line chart and column chart.

# EDA is majorly performed using the following methods:

Univariate visualization – provides summary statistics for each field in the raw data set

Bivariate visualization – is performed to find the relationship between each variable in the dataset and the target variable of interest

Multivariate visualization – is performed to understand interactions between different fields in the dataset

Dimensionality reduction – helps to understand the fields in the data that account for the most variance between observations and allow for the processing of a reduced volume of data.



# Characteristics of Effective Graphics

Induce the viewer to think about the substance rather than: methodology, graphic design, the technology of graphic production or something else.

- Avoid distorting what the data has to say
- Allow data sets to be coherent, in a condensed space
- Serves a reasonably clear purpose: description, exploration, tabulation
- Intimately integrated with the statistical and verbal descriptions of a data set.
- Encourage the brain to compare different pieces of data.
- Can reveal the data at several levels of detail, from a broad overview to the fine structure.



Descriptive Statistics is the default process in Data analysis. Exploratory Data Analysis (EDA) is not complete without a Descriptive Statistic analysis.

## Common Forms of Descriptive Statistics

1. Summary statistics: find the pattern in data, data discovery, data cleaning using
  - a. Measures of central tendency
  - b. Measures of dispersion
2. Tables
  - a. Frequency tables
3. Graphs
  - a. Visualize data with Boxplots, histograms, stem and leaf plots and scatter plots

# Data questions that should be answered

- After data has been loaded into any software tool for analysis, it is readily apparent that for all but the smallest data sets, there is far too much data to be able to make sense of it through visual inspection of the values.
- At the core of the Data Understanding stage is using summary statistics and data visualization to gain insight into what data you have available for modeling.

✓ Is the data any good?

✓ Is it clean?

✓ Is it representative of what it is supposed to measure?

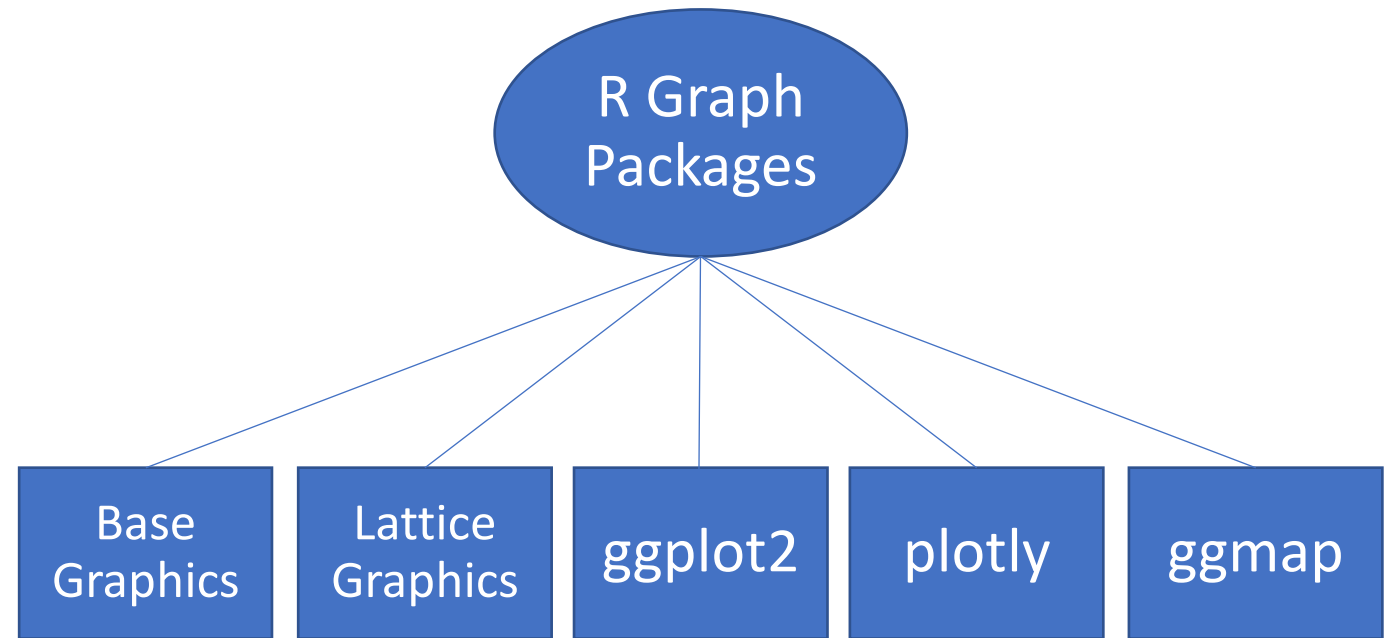
✓ Is it populated?

✓ Is it distributed as you expect?

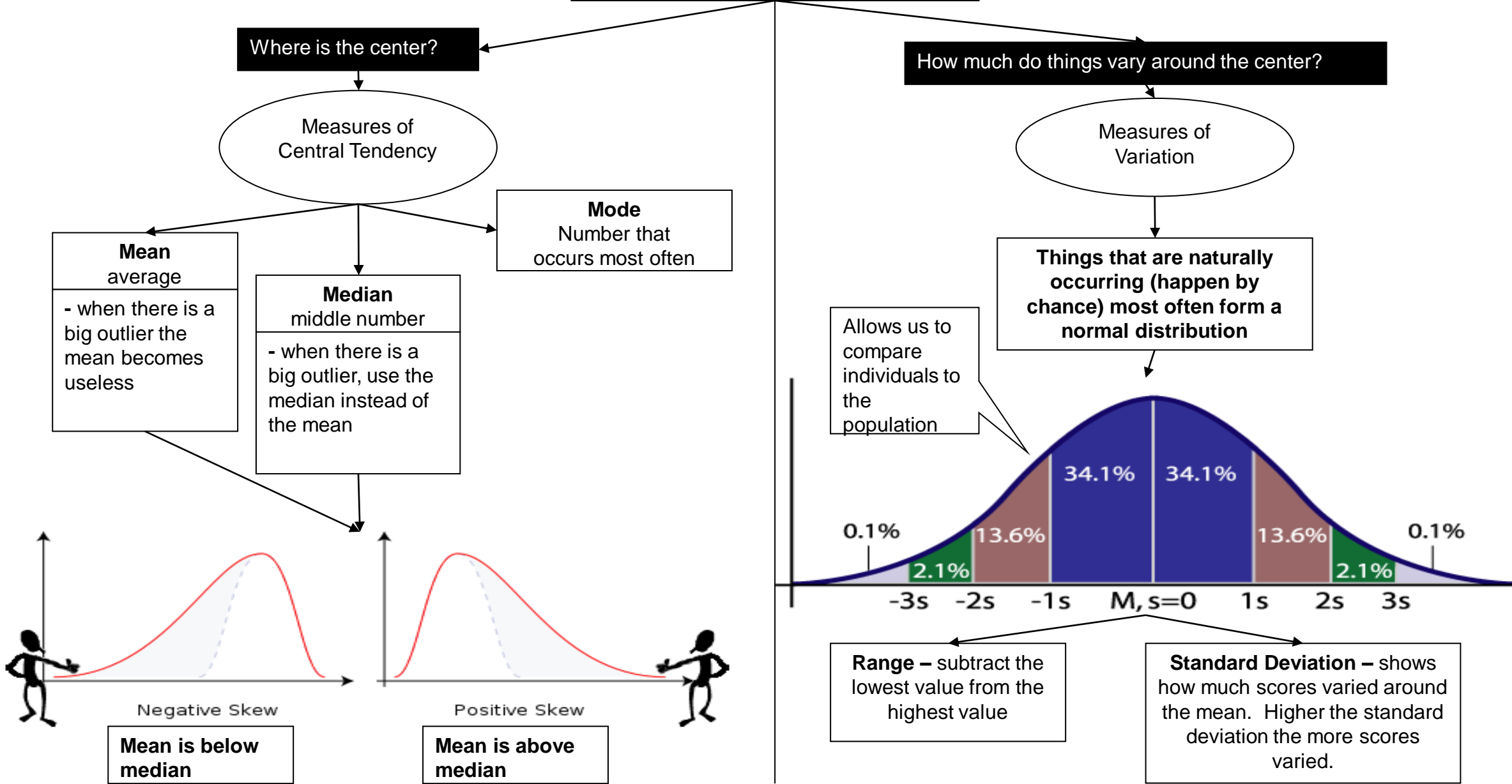
✓ Will it be useful

for building predictive models? These are all questions that should be answered before you begin to build predictive models.

**Data  
visualization  
in R can be  
performed in  
the following  
ways:**



# Descriptive Statistics



Here are some  
simple statistics  
functions you  
will likely use  
often:

Function	Description
<code>range()</code>	Range (minimum and maximum) of vector
<code>min()</code> , <code>max()</code>	Minimum or maximum of vector
<code>mean()</code> , <code>median()</code>	Mean or median of vector
<code>sd()</code>	Standard deviation of vector
<code>table()</code>	Number of observations per level for a factor vector
<code>cor()</code>	Determine correlation(s) between two or more vectors
<code>summary()</code>	Summary statistics, depends on class



# Measure of Central Tendency

When to use mean, median and mode?

- Mean – Average value, when your data is not skewed i.e., normally distributed. In other words, there are no extreme values present in the data set.
- Median –Middle value, when your data is skewed or you are dealing with ordinal (ordered categories) data (e.g., Likert scale 1. Strongly dislike 2. Dislike 3. Neutral 4. Like 5. Strongly like)
- Mode – Most frequent value, when dealing with nominal (unordered categories) data.

## Measures of Dispersion:

Measures of variability gives how “spread out” the data

---

Range	Difference between max and min in a distribution
Interquartile range	Correspondes to the difference between the first and third quartiles
Standard Deviation	Average distance of scores in a distribution from their mean
Variance	Square of the standard deviation
Skewness	Degree to which scores in a distribution are spread out
Kurtosis	Flatness of peakness of the curve

## Descriptive Statistics

Variable	<u>Obs</u>	Mean	<u>Std.Dev.</u>	Min	Max
price	74	6165.257	2949.496	3291	15906
mpg	74	21.297	5.786	12	41
rep78	69	3.406	.99	1	5
headroom	74	2.993	.846	.846	5
trunk	74	13.757	4.277	5	23
weight	74	3019.459	777.194	1760	4840
length	74	187.932	22.266	142	233
turn	74	39.649	4.399	31	51
displacement	74	197.297	91.837	79	425
<u>gear_ratio</u>	74	3.015	.456	2.19	3.89
foreign	74	.297	.46	0	1



# R Special Summary Commands

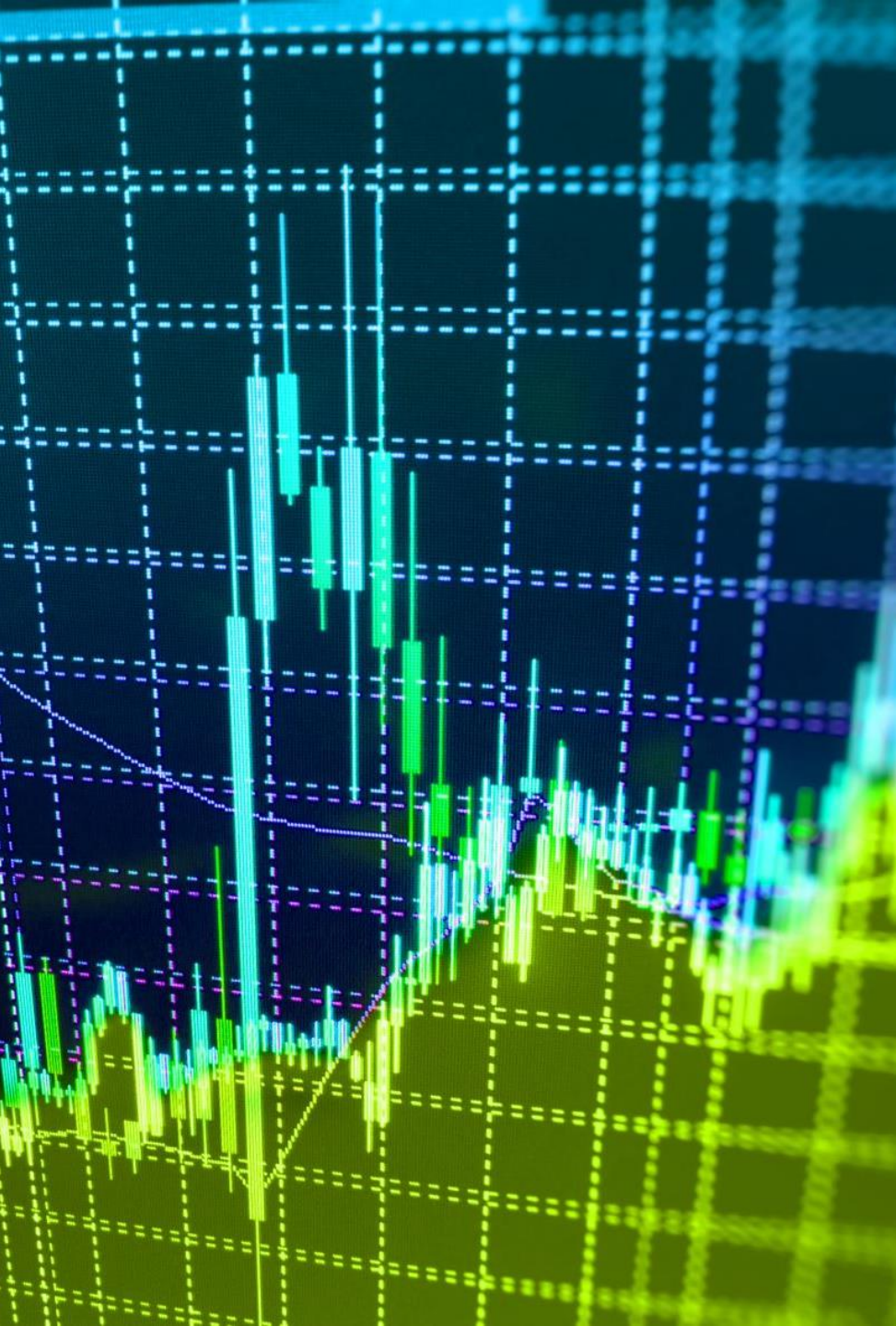
There are two types of special summary commands:

- Row Summary Commands – Applied to work with row data. Two commands here are *rowMeans()* and *rowSums()*.
- Column Summary Commands – Also, applied to work with row data but the two commands here are *colMeans()* and *colSums()*.

Q1	Q2	Q3	Q4	Q5
0	0	0	1	1
0	1	1	1	0
1	0	0	1	1
0	0	1	1	0
1	1	1	1	1

```
patientsurvey <- data.frame("q1" = c(0, 0, 1, 0, 1),
                             "q2" = c(0, 1, 0, 0, 1),
                             "q3" = c(0, 1, 0, 1, 1),
                             "q4" = c(1, 1, 1, 1, 1),
                             "q5" = c(1, 0, 1, 0, 1))
```

```
rowMeans(patientsurvey)
rowSums(patientsurvey)
colMeans(patientsurvey)
colSums(patientsurvey)
```



## Frequency tables: Used to describe categorical variables

---

You can generate contingency (frequency) tables using:

- the `table()` function
  - tables of proportions using the `proportions()` function (formerly `prop.table`), and
  - marginal frequencies using `marginSums()` (formerly `margin.table`)
- 
- Compute table margins and relative frequency
    - `table(x)`
    - `marginSums(x, margin = NULL)`
    - `proportions(x, margin = NULL)`



# Frequency tables: Used to describe categorical variables

Weight (Kg)	Frequency	Cumulative Frequency
0 up to 20	2	2
20 up to 40	7	9
40 up to 60	12	21
60 up to 80	6	27
80 up to 100	3	30

- For demonstration, see R program FrequencyTables.R

- `install.packages("Hmisc")`
- `library("Hmisc")`
- Similar to the `summary` function is the `describe` function.

No	variable	stats / values	Freqs (% of valid)	Text Graph	valid	Missing
1	type [factor]	1. A 2. B 3. C 4. D 5. E	7 (11.7%) 15 (25.0%) 11 (18.3%) 8 (13.3%) 19 (31.7%)	IIIIII IIIIIIIIIIIIII IIIIIIIIII IIIIII IIIIIIIIIIIIIIIIII	60 (93.75%)	4 (6.25%)
2	score [integer]	mean (sd) : 48.29 (31.71) min < med < max : 0 < 45 < 124 IQR (cv) : 54 (0.66)	47 distinct val.	: .	58 (90.62%)	6 (9.38%)
3	category [factor]	1. X 2. Y 3. Z	17 (26.6%) 10 (15.6%) 37 (57.8%)	IIIIIIII IIII IIIIIIIIIIIIIIIIII	64 (100%)	0 (0%)
4	rating [integer]	mean (sd) : 1389025.28 (12598684.85) min < med < max : -11102847 < 5 < 1e+08 IQR (cv) : 7.25 (9.07)	22 distinct val.	: : : : : :	64 (100%)	0 (0%)

# Base R

## SET GRAPHICAL PARAMETERS

*the following can only be set with par()*

**par (...)**

<i>multiple plots</i>	<code>mfc</code> = <code>c(nrow, ncol)</code>	<i>plot margins (outer)</i>	<code>oma</code> = <code>c(bottom, left, top, right)</code> default: <code>c(0, 0, 0, 0)</code> lines
	<code>mfrow</code> = <code>c(nrow, ncol)</code>		
<i>plot margins</i>	<code>mar</code> = <code>c(bottom, left, top, right)</code> default: <code>c(5.1, 4.1, 4.1, 2.1)</code> lines	<i>query x &amp; y limits</i>	<code>par ("usr")</code>

## CREATE A NEW PLOT

<b>Bar charts</b>	<code>barplot(height, ...)</code>	<b>Histograms</b>	<code>hist(X, ...)</code>
<i>bar labels</i>	<code>names.arg</code> =	<i>breakpts</i>	<code>breaks</code> =
<i>border</i>	<code>border</code> =		
<i>fill color</i>	<code>col</code> =		
<i>horizontal</i>	<code>horiz</code> = TRUE		
<b>Box plots</b>	<code>boxplot(X, ...)</code>	<b>Line charts</b>	<code>plot(X, type = "l")</code>
<i>horizontal</i>	<code>horizontal</code> = TRUE	<i>line type</i>	<code>lty</code> = "blank" 0 "solid" 1 "dashed" 2 "dotted" 3
<i>box labels</i>	<code>names</code> =	<i>line width</i>	<code>lwd</code> =
<b>Dot plots</b>	<code>dotchart(X, ...)</code>	<b>Scatterplots</b>	<code>plot(X, ...)</code>
<i>dot labels</i>	<code>labels</code> =	<i>symbol</i>	<code>pch</code> =

## REMOVE

<i>axis labels</i>	<code>ann</code> = FALSE
<i>axis, tickmarks, and labels</i>	<code>xaxt</code> = "n" <code>yaxt</code> = "n"
<i>plot box</i>	<code>bty</code> = "n"

NOTE: Many of the parameters here can be also be set in `par()`. See R help for more options.

## ADJUST

<i>allow plotting out of plot region</i>	<code>xpd</code> = TRUE
<i>aspect ratio</i>	<code>asp</code> =
<i>axis limits</i>	<code>xlim</code> =, <code>ylim</code> =
<i>axis lines to match axis limits</i>	<code>xaxs</code> = "i", <code>yaxs</code> = "i" (internal axis calculation)

## ADD TEXT

<b>location</b>	<code>xlab</code> =, <code>ylab</code> =	<b>size</b> (magnification factor)	<code>cex</code> =
<i>axis labels</i>	<code>sub</code> =	<i>all elements</i>	<code>cex.lab</code> =
<i>subtitle</i>	<code>main</code> =	<i>axis labels</i>	<code>cex.sub</code> =
<i>title</i>		<i>subtitle</i>	<code>cex.axis</code> =
		<i>tick mark labels</i>	<code>cex.main</code> =
<b>font face</b>	<code>font</code> = 1 (plain) 2 (bold) 3 (italic) 4 (bold italic)	<b>position</b>	
<i>font family</i>	<code>family</code> = "serif" "sans" "mono"	<i>text direction</i>	<code>las</code> = 1 (horizontal)
		<i>justification</i>	<code>adj</code> = 0 .5 1 (left, center, right)

## ADD TO AN EXISTING PLOT

<b>Add new plot</b> [any plot function]	<code>(..., add = TRUE)</code> ex <code>barplot(x, add = TRUE)</code>	<b>Lines</b>	<code>lines(X, ...)</code>
		<i>line style</i>	<code>lty</code> =
		<i>line width</i>	<code>lwd</code> =
		<i>color</i>	<code>col</code> =
<b>Axes</b>	<code>axis(side, ...)</code>	<b>Points</b>	<code>points(X, ...)</code>
<i>location</i>	<code>side</code> = 1 2 3 4 (bottom, left, top, right)	<i>symbol</i>	<code>pch</code> = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
<i>tick mark:</i>		<i>color</i>	<code>col</code> =
<i>labels</i>	<code>labels</code> =	<i>fill color</i>	<code>bg</code> = (pch: 21-25 only)
<i>location</i>	<code>at</code> =		
<i>remove</i>	<code>tick</code> = FALSE	<b>Text</b>	<code>text(x, y, text, ...)</code>
<i>rotate text</i>	<code>las</code> = 1 (horizontal)	<i>position (rel. to x,y)</i>	<code>pos</code> = 1 2 3 4 (below, left, above, right) (default=center)
<b>Axis labels</b>	<code>mtext(text, ...)</code>	<b>Title</b>	<code>title(main, ...)</code>
<i>location</i>	<code>side</code> = 1 2 3 4 (bottom, left, top, right)	<i>axis labels</i>	<code>xlab</code> =, <code>ylab</code> =
<i>lines to skip</i>	<code>line</code> = (from plot region, default = 0)	<i>subtitle</i>	<code>sub</code> =
<i>position</i>	<code>at</code> = x or y-coord (depending on side)	<i>title</i>	<code>main</code> =
<i>justification</i>	<code>adj</code> = 0 .5 1 (left, center, right)		

# The main arguments in R graphics:

legend : names to display

bty: type of box around the legend.

horiz : legend in column or in row

col : symbol color

pch: symbol type

pt.cex : symbol size

cex : text size

text.col: text color

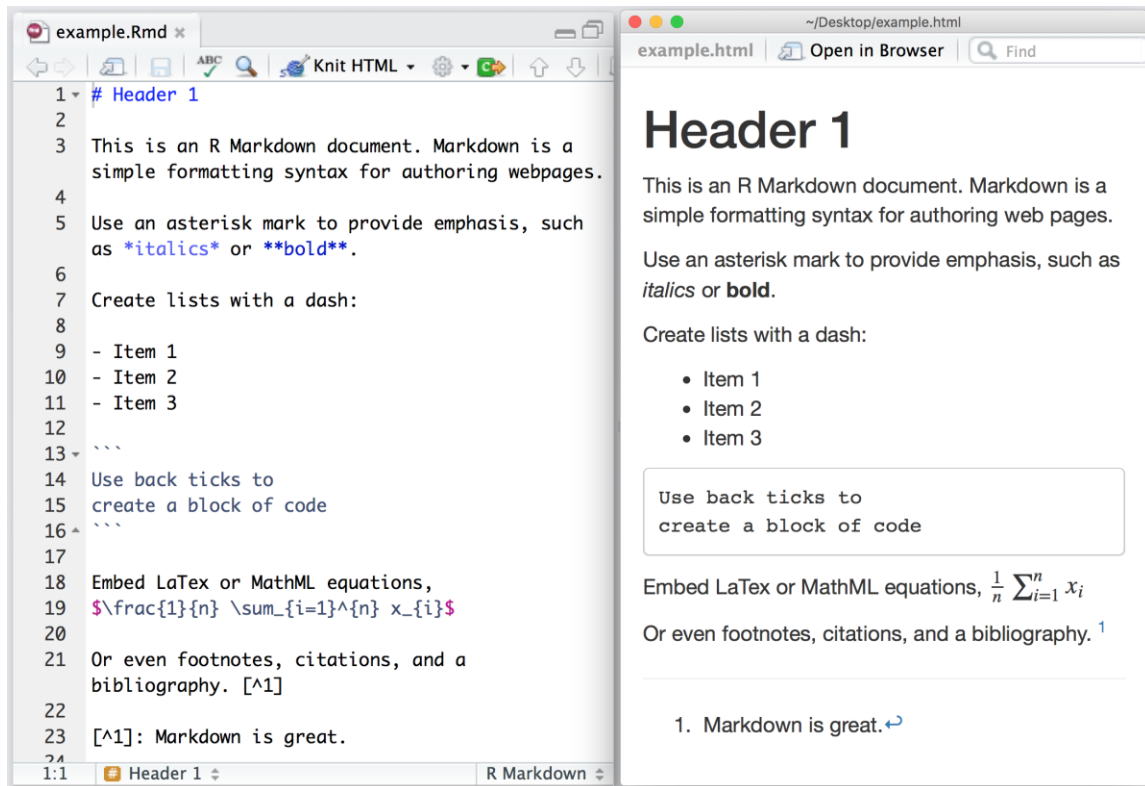
topright : legend position: bottomright, bottom, bottomleft, left, topleft, top, topright, right, center.

inset : % (from 0 to 1) to draw the legend away from x and y axis

You can also give the X and Y coordinate of the legend:

legend(3, 5, ...)

# RMarkdown

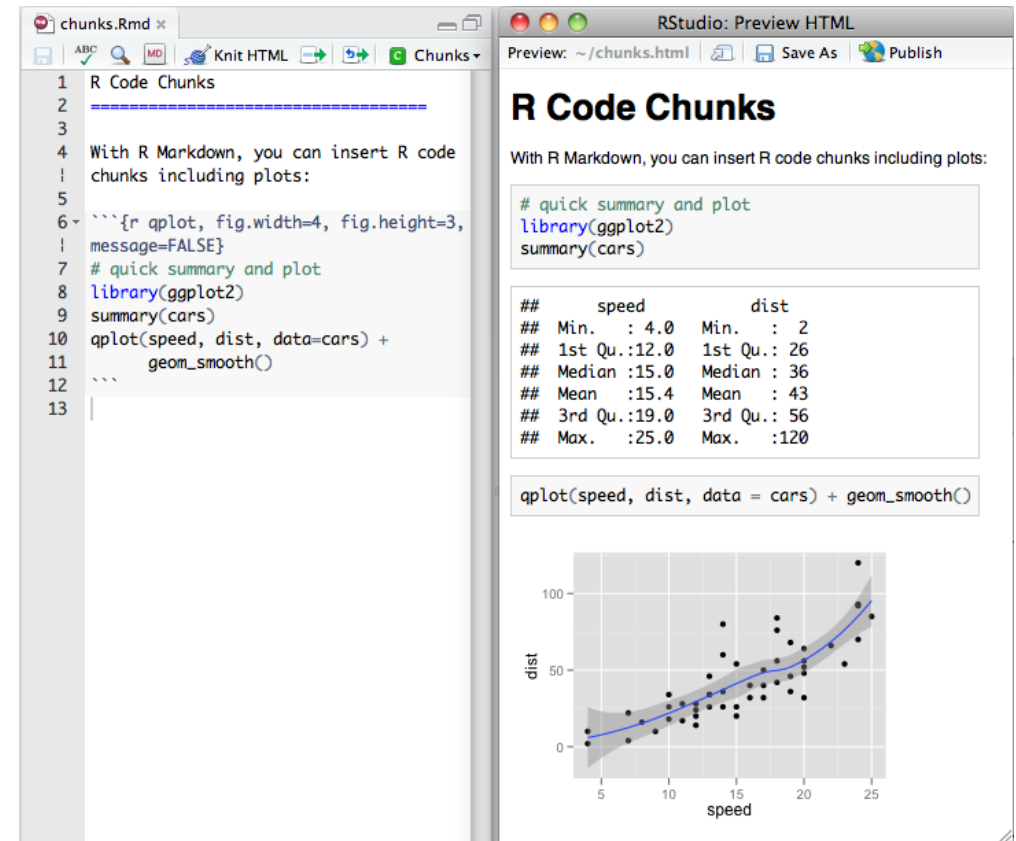


The screenshot shows the RStudio editor with a file named 'example.Rmd'. The document content is as follows:

```
1 # Header 1
2
3 This is an R Markdown document. Markdown is a
4 simple formatting syntax for authoring webpages.
5 Use an asterisk mark to provide emphasis, such
6 as italics or bold.
7
8 Create lists with a dash:
9 - Item 1
10 - Item 2
11 - Item 3
12
13 ```
14 Use back ticks to
15 create a block of code
16 ```
17
18 Embed LaTeX or MathML equations,
19 
$$\frac{1}{n} \sum_{i=1}^n x_i$$

20
21 Or even footnotes, citations, and a
22 bibliography. [^1]
23
24 [^1]: Markdown is great.
```

The right pane shows the rendered HTML output 'example.html'. The document is titled 'Header 1' and contains the same text as the R Markdown source, with formatting applied: italics, bold, a bulleted list, a code block, a LaTeX equation, and a footnote.



The screenshot shows the RStudio editor with a file named 'chunks.Rmd'. The document content is as follows:

```
1 R Code Chunks
2 =====
3
4 With R Markdown, you can insert R code
5 chunks including plots:
6
7 ```{r qplot, fig.width=4, fig.height=3,
8   message=FALSE}
9 # quick summary and plot
10 library(ggplot2)
11 summary(cars)
12 aplot(speed, dist, data=cars) +
13   geom_smooth()
```

The right pane shows the rendered HTML output 'Preview: ~/chunks.html'. The document is titled 'R Code Chunks' and contains the same text as the R Markdown source, with the R code chunk executed and the resulting summary and plot displayed.

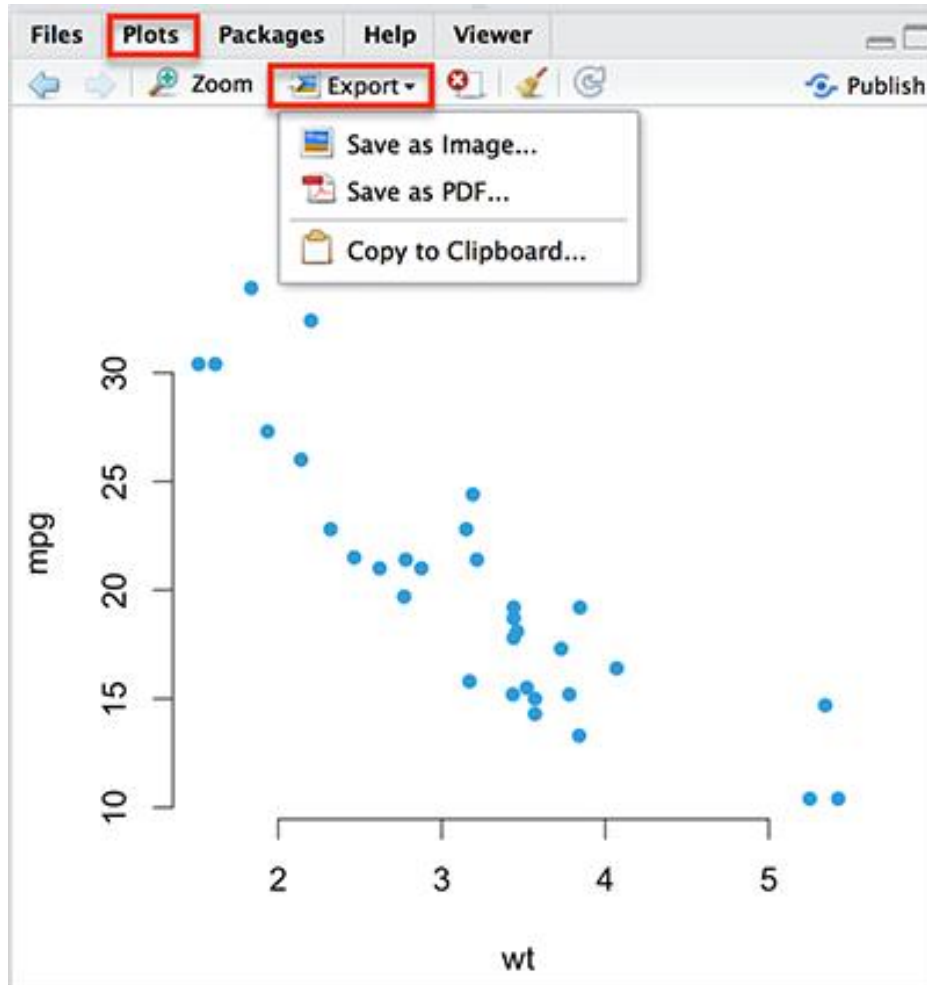
Summary of cars data:

	speed	dist
## Min.	4.0	2
## 1st Qu.	12.0	26
## Median	15.0	36
## Mean	15.4	43
## 3rd Qu.	19.0	56
## Max.	25.0	120

The plot shows 'dist' on the y-axis (0 to 100) and 'speed' on the x-axis (5 to 25). It displays a scatter plot of data points with a blue smoothed regression line and a grey shaded confidence interval.

Great reference on RMarkdown can be found [here](#).

Plots panel → Export → Save as Image or Save as PDF



It's also possible to save the graph using R codes as follow:

1. Specify files to save your image using a function such as **jpeg()**, **png()**, **svg()** or **pdf()**. Additional argument indicating the width and the height of the image can be also used.
2. Create the plot
3. Close the file with **dev.off()**

```
# Customizing the output
pdf("my_plot.pdf",           # File name
    width = 8, height = 7,   # Width and height in inches
    bg = "white",            # Background color
    colormodel = "cmyk",     # Color model (cmyk is required
                             # for most publications)
    paper = "a4")            # Paper size

# Creating a plot
plot(rnorm(20))

# Closing the graphical device
dev.off()
```



Barplot	A barplot (or barchart; bargraph) illustrates the association between a numeric and a categorical variable. The barplot represents each category as a bar and reflects the corresponding numeric value with the bar's size.	<code>barplot(x)</code>
Boxplot	Displays the distribution of a numerical variable based on five summary statistics: minimum non-outlier; first quartile; median; third quartile; and maximum non-outlier. Furthermore, boxplots show the positioning of outliers and whether the data is skewed.	<code>boxplot(x)</code>
Density Plot	A density plot (or kernel density plot; density trace graph) shows the distribution of a numerical variable over a continuous interval. Peaks of a density plot visualize where the values of numerical variables are concentrated.	<code>plot(density(x))</code>
Heatmap	A heatmap (or shading matrix) visualizes individual values of a matrix with colors. More common values are typically indicated by brighter reddish colors and less common values are typically indicated by darker colors.	<code>heatmap(cbind(x, y))</code>
Histogram	A histogram groups continuous data into ranges and plots this data as bars. The height of each bar shows the number of observations within each range.	<code>hist(x)</code>

Line Plot	A line plot, visualizes values along a sequence (e.g., over time). Line plots consist of an x-axis and a y-axis. The x-axis usually displays the sequence and the y-axis the values corresponding to each point of the sequence.	<code>plot(1:length(y), y, type = "l")</code>
Pairs Plot	A pairs plot is a plot matrix, consisting of scatterplots for each variable-combination of a data frame.	<code>pairs(data.frame(x, y))</code>
Polygon Plot	A polygon plot displays a plane geometric figure (i.e., a polygon) within the plot	<code>plot(1,1 col = "white", xlab="X", ylab = "Y") polygon(x= c(0.7, 1.3, 1.3, 0.8), y=c(0.6, 1.0, 1.4, 1.3), col = "#353436")</code>
QQplot	Quantile-Quantile plot; Quantile-Quantile diagram) determines whether two data sources come from a common distribution. QQ plots draw the quantiles of the two numerical data sources against each other. If both data sources come from the same distribution, the points fall on a 45-degree angle.	<code>qqplot(x,y)</code>
Scatterplot	A scatterplot (or scatter plot; scatter graph; scatter chart; scattergram; scatter diagram) displays two numerical variables with points, whereby each point represents the value of one variable on the x-axis and the value of the other variable on the y-axis.	<code>plot(x,y)</code>
Venn Diagram	A venn diagram (or primary diagram; set diagram; logic diagram) illustrates all possible logical relations between certain data characteristics. Each characteristic is represented as a circle, whereby overlapping parts of the circles illustrate elements that have both characteristics at the same time.	<code>Install.packages("VennDiagram") library("VennDiagram") plot.new() draw.single.venn(area = 10)</code>

# R Graphical Parameters Cheat Sheet

Example:

```
par(mfrow=c(1,2))
```

# set the plotting area into a 1\*2 array

```
barplot(max.temp, main="Barplot")
pie(max.temp, main="Piechart",
radius=1)
```

## par () Graphical Parameters

Visual cheat sheet for some plot parameters in R. See `?par` for more information.

### Symbol Styles

pch   Point Types	lty   Line Types
○ 1	— 1
△ 2	- - - 2
+ 3	... 3
× 4	- . - . 4
◇ 5	- - - 5
▽ 6	- . . . 6
⊠ 7	
* 8	lwd   Line Width
⊕ 9	— .1
⊗ 10	— .25
⊛ 11	— .5
⊞ 12	— 1
⊟ 13	— 3
⊠ 14	— 6
■ 15	
● 16	
▲ 17	
◆ 18	
● 19	
● 20	
○ 21	
□ 22	
◇ 23	
△ 24	
▽ 25	

you can also use any character

### Figures Arrangement

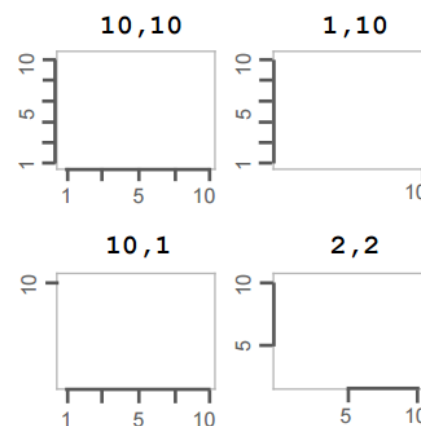
**mfrow** | Multiple Figures by Row  
2, 3



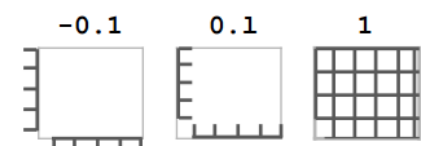
Also available **mfc** for multiple figures by column

### Axes

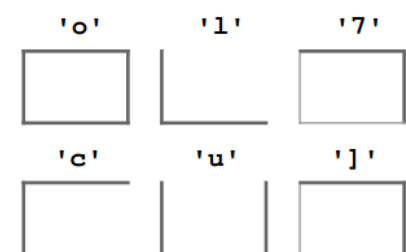
**lab** | Tick Placement



**tck** | Tick Length



**bty** | Box Type



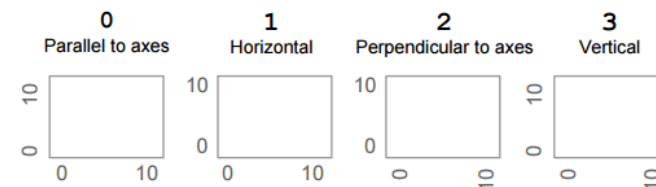
### Text and Labels

**family, font** | Typeface and Font Style

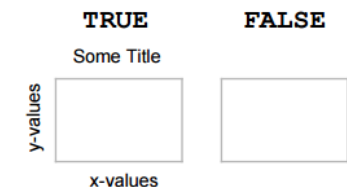
family: mono font: 1	family: serif font: 1	family: sans font: 1
family: mono font: 2	family: serif font: 2	family: sans font: 2
family: mono font: 3	family: serif font: 3	family: sans font: 3
family: mono font: 4	family: serif font: 4	family: sans font: 4

Also available: **font.main** (main title), **font.lab** (axis labels), **font.sub** (subtitle)

**las** | Label Orientation



**ann** | Plot Annotation

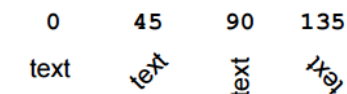


**lheight** | Line Height

1  
The quick brown fox jumps over the lazy dog and runs away with all the food

1.5  
The quick brown fox jumps over the lazy dog and runs away with all the food

**srt** | String Rotation



Based on Flowing Data's cheat sheet