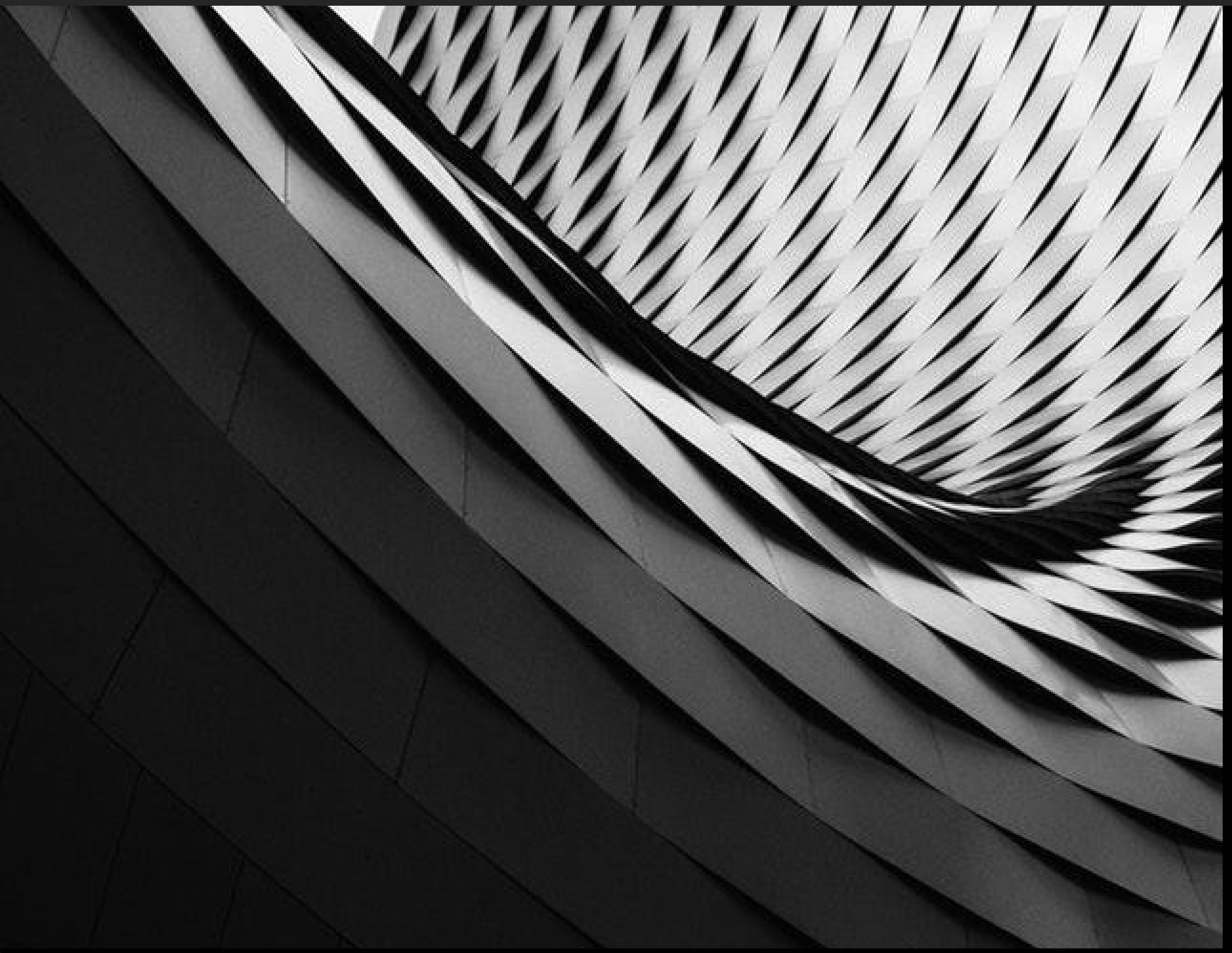
R | Basic Text Analysis

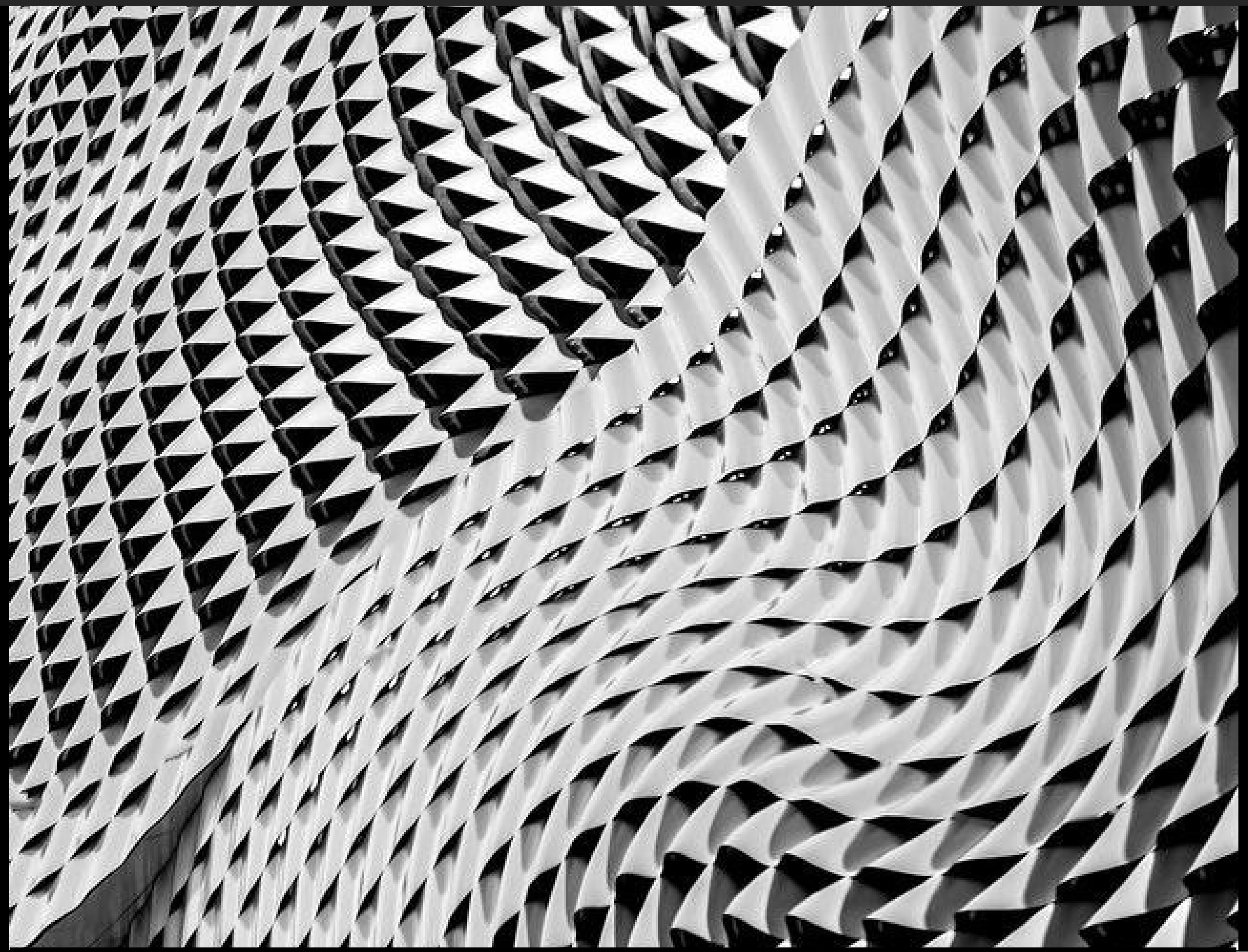
Presented by Aviv Lo

Table of Contents



Topic 2

AI/ML/NLP Explained

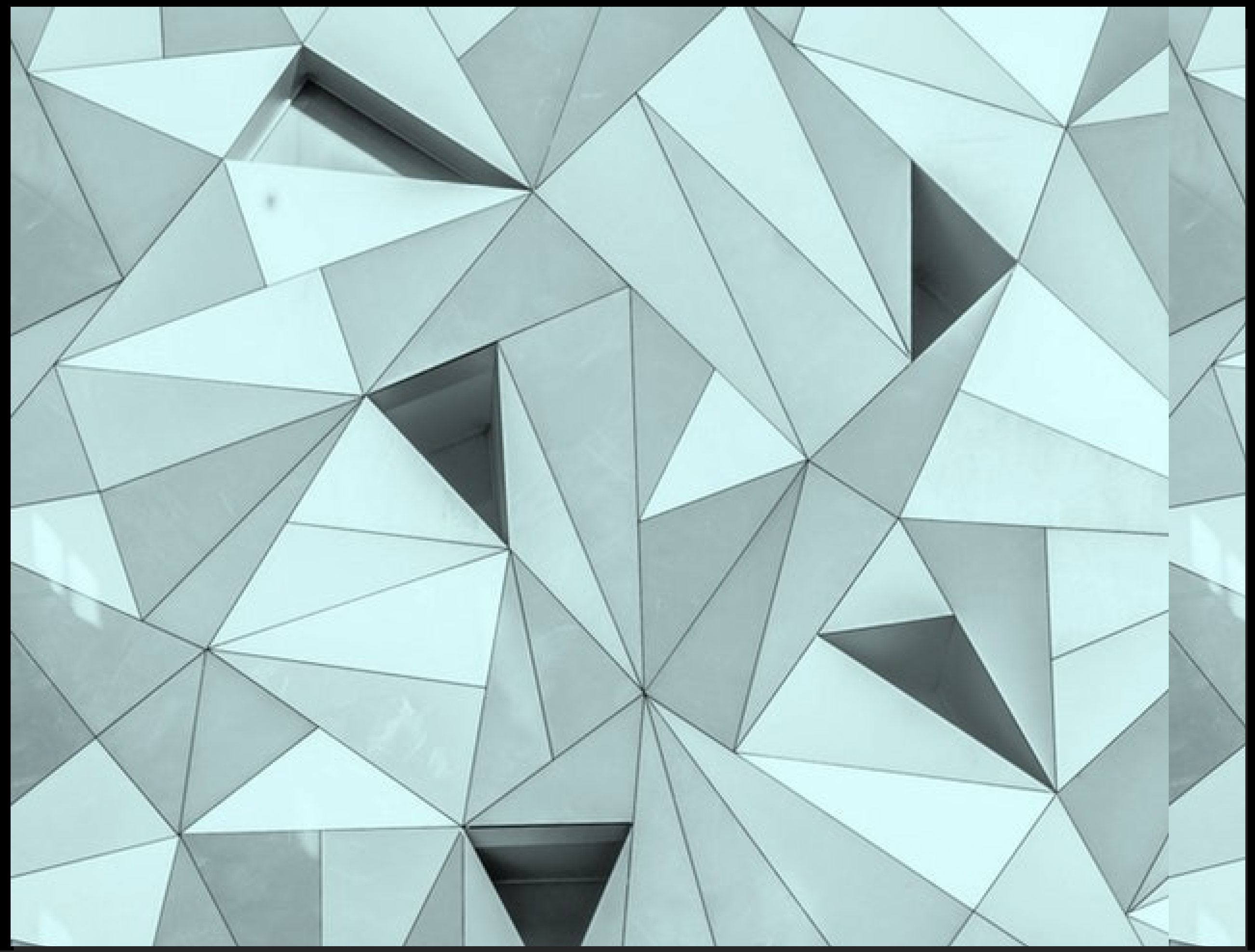


Topic 4

R Basics

Topic 1

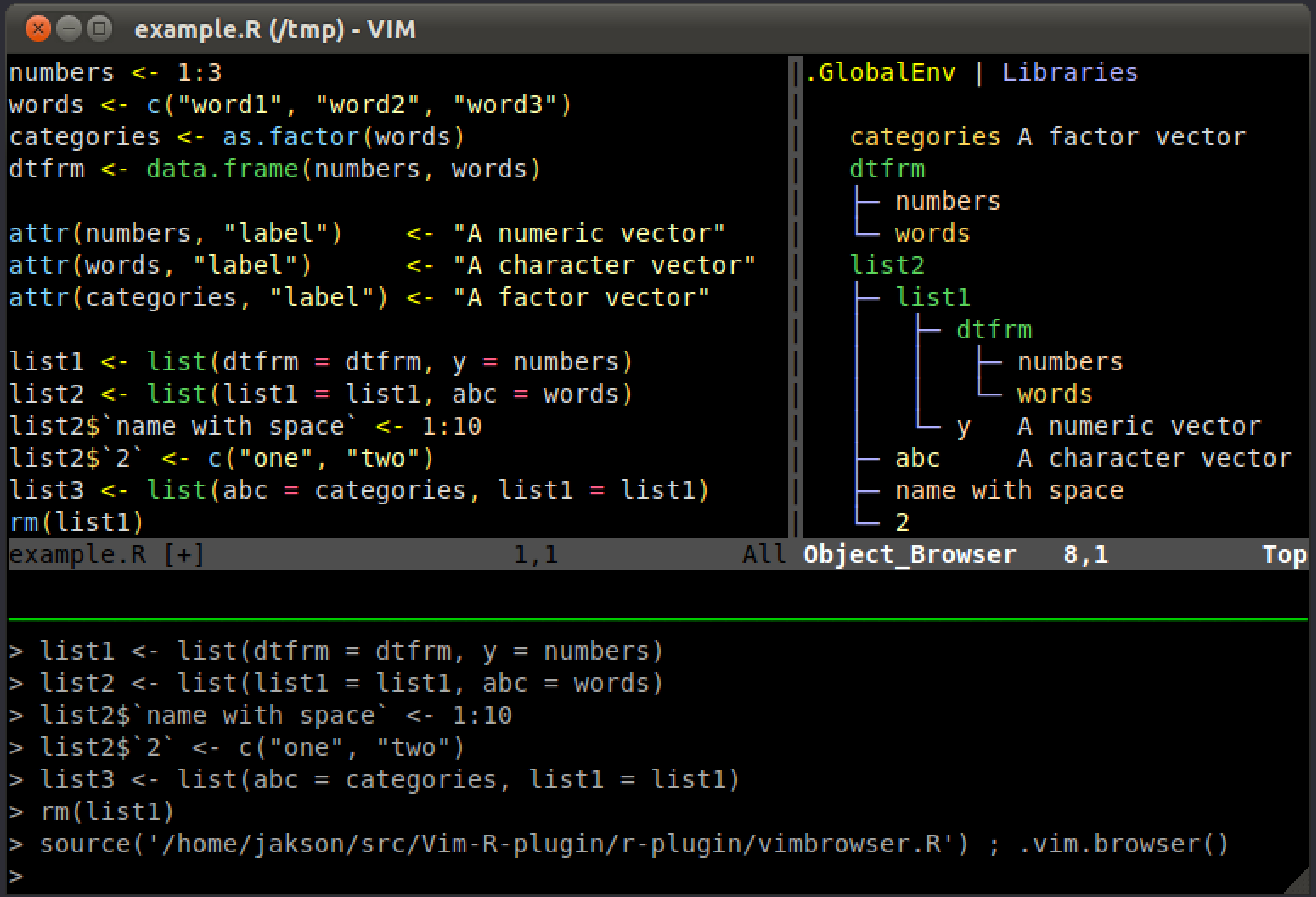
What is R?



Topic 3

Text Analysis

1 What exactly is R?

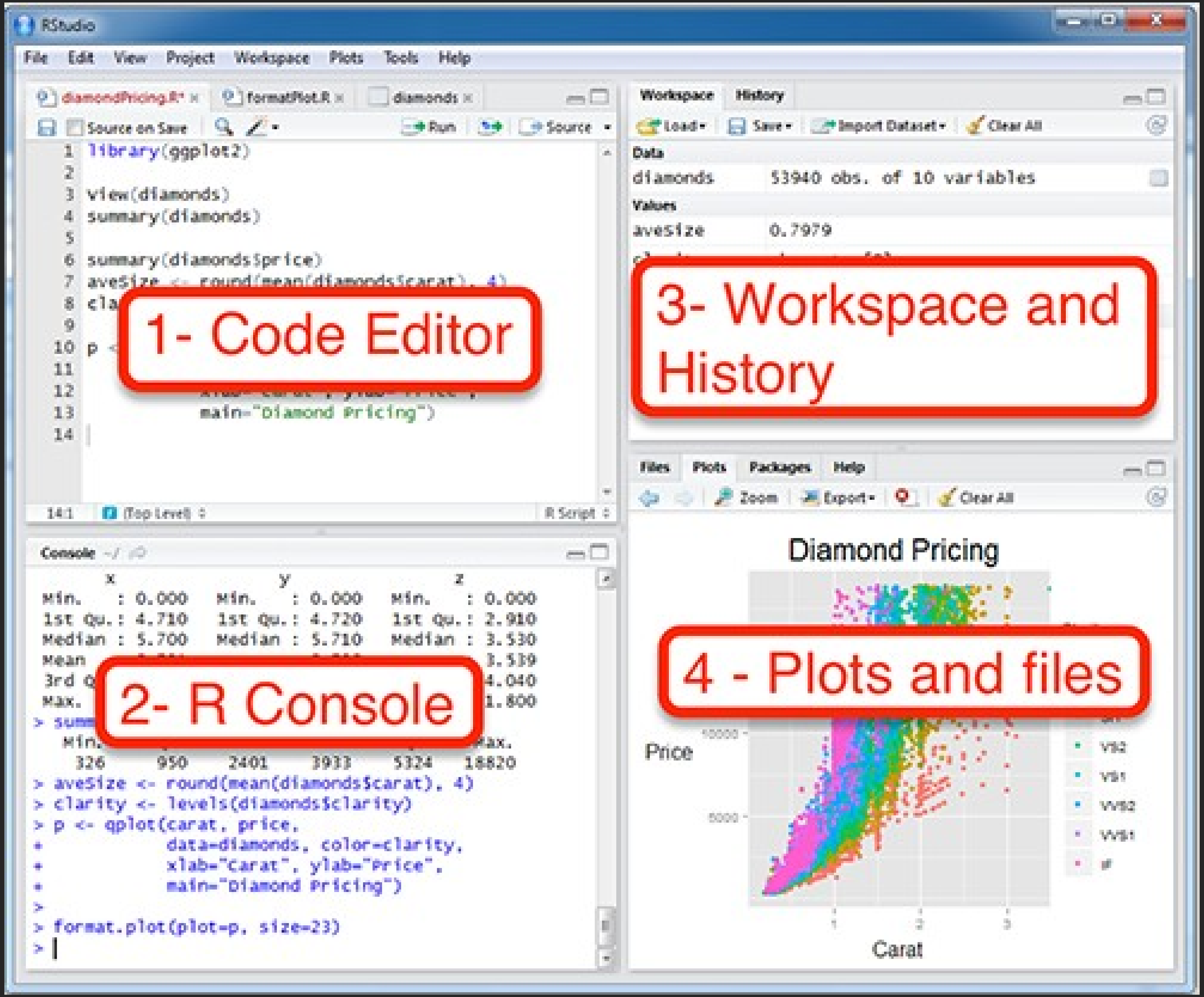


```
numbers <- 1:3                                      .GlobalEnv | Libraries
words <- c("word1", "word2", "word3")
categories <- as.factor(words)                         categories A factor vector
dtfrm <- data.frame(numbers, words)                    dtfrm
                                                       ├ numbers
attr(numbers, "label")    <- "A numeric vector"        └ words
attr(words, "label")      <- "A character vector"      list2
attr(categories, "label") <- "A factor vector"         ├ list1
                                                       │   ├ dtfrm
list1 <- list(dtfrm = dtfrm, y = numbers)              │   │   ├ numbers
list2 <- list(list1 = list1, abc = words)              │   │   └ words
list2$`name with space` <- 1:10                        │   └ y   A numeric vector
list2$`2` <- c("one", "two")                           ├ abc     A character vector
list3 <- list(abc = categories, list1 = list1)         ├ name with space
rm(list1)                                              └ 2
example.R [+]                        1,1          All Object_Browser    8,1          Top
```

```
> list1 <- list(dtfrm = dtfrm, y = numbers)
> list2 <- list(list1 = list1, abc = words)
> list2$`name with space` <- 1:10
> list2$`2` <- c("one", "two")
> list3 <- list(abc = categories, list1 = list1)
> rm(list1)
> source('/home/jakson/src/Vim-R-plugin/r-plugin/vimbrowser.R') ; .vim.browser()
>
```

1. Developed by Ross Ihaka and Robert Gentleman in 1991
2. Hence the name “R”
3. Maintained by The R Foundation for Statistical Computing
4. Difficult for beginners + Steep learning curve
5. Rich and powerful visualization libraries
6. Support matrix / vectorized operations
7. It’s born for stats

RStudio



1- Code Editor

```
library(ggplot2)

View(diamonds)
summary(diamonds)

summary(diamonds$price)
aveSize <- round(mean(diamonds$carat), 4)
cla

p <

            xlab="Carat", ylab="Price",
            main="Diamond Pricing")

```

2- R Console

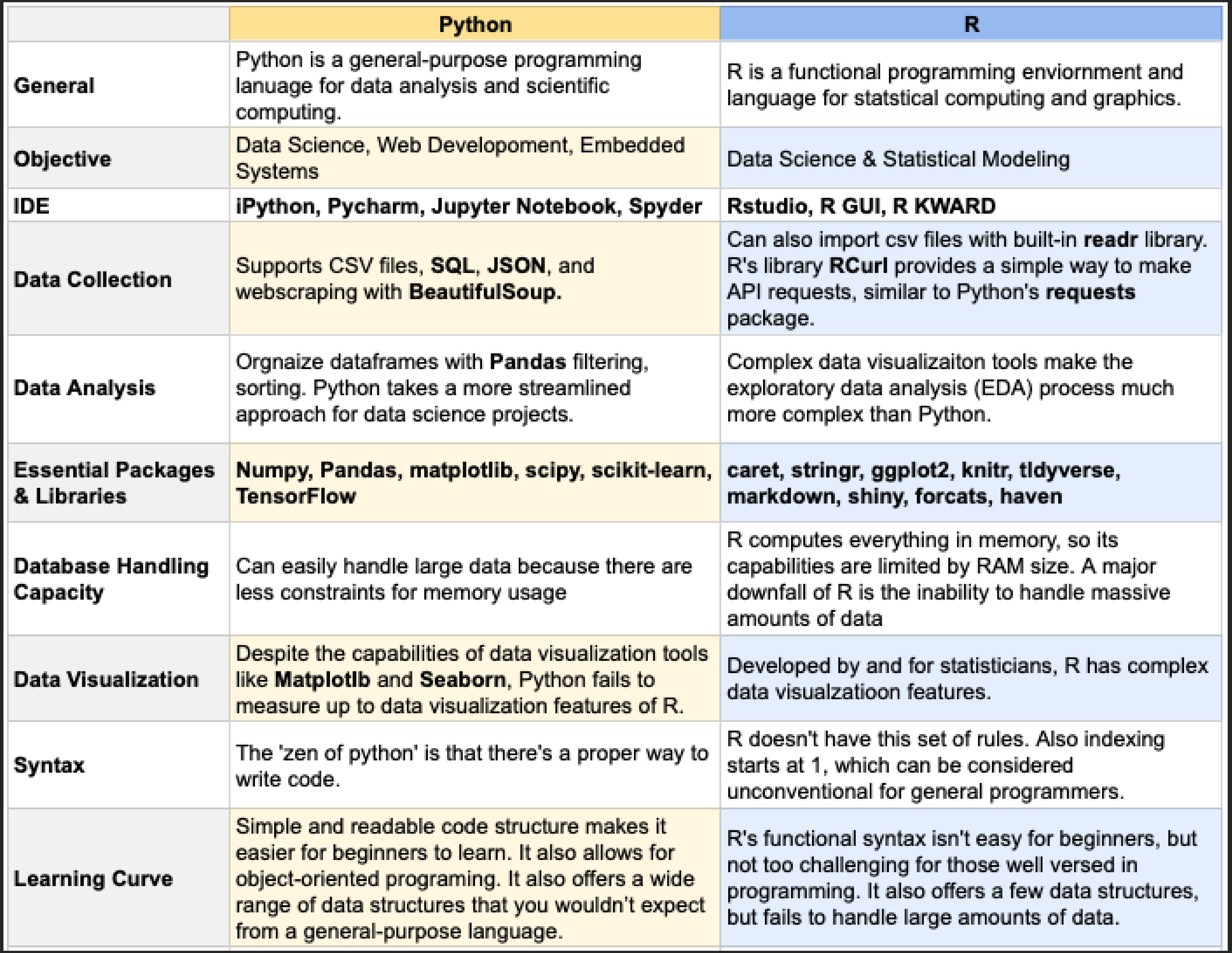
```
       x                 y                 z
 Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
 1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
 Median : 5.700   Median : 5.710   Median : 3.530
 Mean                                      : 3.539
 3rd Q                                       4.040
 Max.                                        1.800
> summ
   Min.                                      Max.
    326      950     2401     3933     5324    18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+            data=diamonds, color=clarity,
+            xlab="Carat", ylab="Price",
+            main="Diamond Pricing")
>
> format.plot(plot=p, size=23)
> 
```

3- Workspace and History

Data	
diamonds	53940 obs. of 10 variables
Values	
aveSize	0.7979

4 - Plots and files

R vs. Python

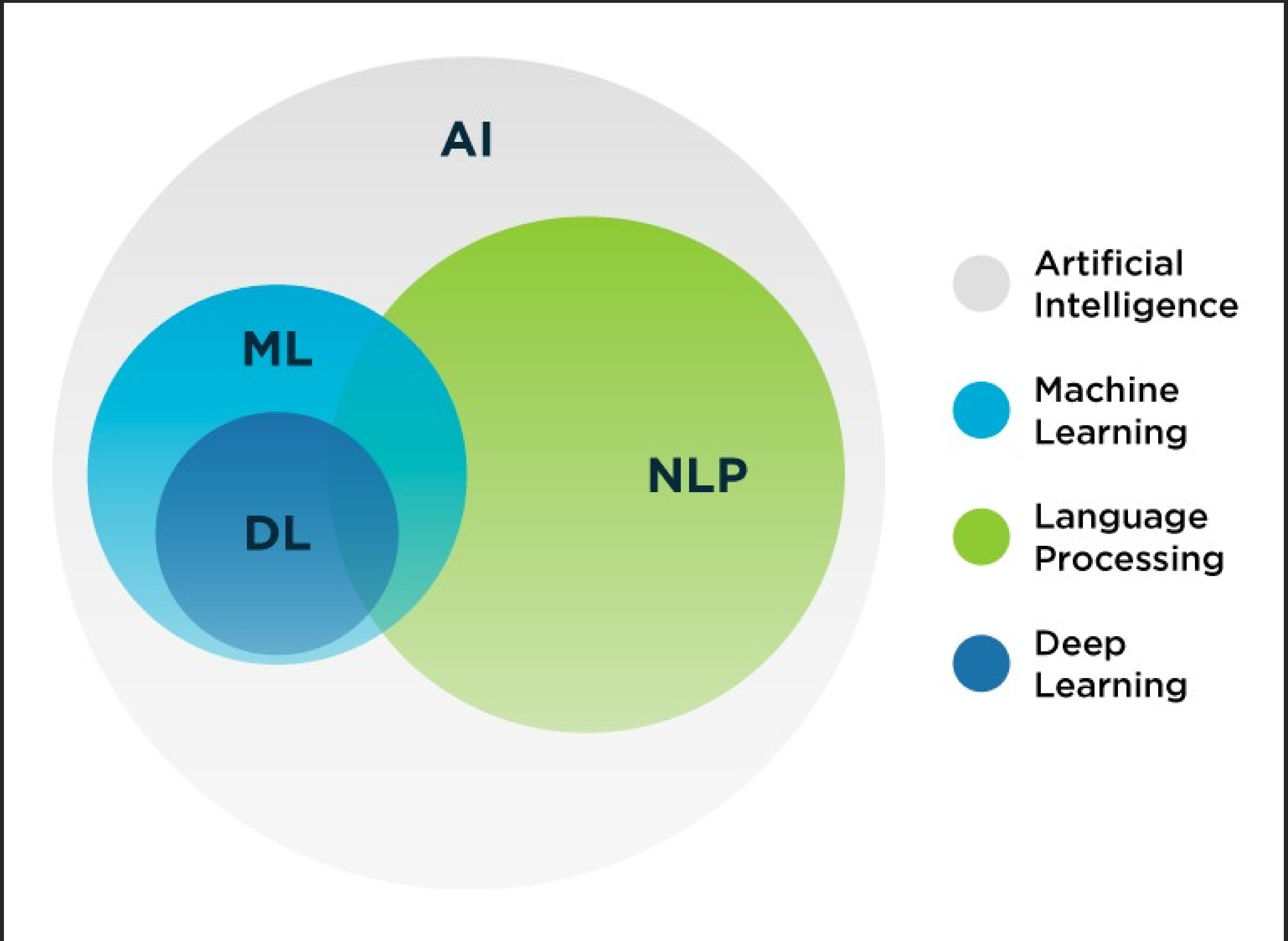


	Python	R
General	Python is a general-purpose programming lanuage for data analysis and scientific computing.	R is a functional programming enviornment and language for statstical computing and graphics.
Objective	Data Science, Web Developoment, Embedded Systems	Data Science & Statistical Modeling
IDE	**iPython, Pycharm, Jupyter Notebook, Spyder**	**Rstudio, R GUI, R KWARD**
Data Collection	Supports CSV files, **SQL**, **JSON**, and webscraping with **BeautifulSoup.**	Can also import csv files with built-in **readr** library. R's library **RCurl** provides a simple way to make API requests, similar to Python's **requests** package.
Data Analysis	Orgnaize dataframes with **Pandas** filtering, sorting. Python takes a more streamlined approach for data science projects.	Complex data visualizaiton tools make the exploratory data analysis (EDA) process much more complex than Python.
Essential Packages & Libraries	**Numpy, Pandas, matplotlib, scipy, scikit-learn, TensorFlow**	**caret, stringr, ggplot2, knitr, tldyverse, markdown, shiny, forcats, haven**
Database Handling Capacity	Can easily handle large data because there are less constraints for memory usage	R computes everything in memory, so its capabilities are limited by RAM size. A major downfall of R is the inability to handle massive amounts of data
Data Visualization	Despite the capabilities of data visualization tools like **Matplotlb** and **Seaborn**, Python fails to measure up to data visualization features of R.	Developed by and for statisticians, R has complex data visualzatioon features.
Syntax	The 'zen of python' is that there's a proper way to write code.	R doesn't have this set of rules. Also indexing starts at 1, which can be considered unconventional for general programmers.
Learning Curve	Simple and readable code structure makes it easier for beginners to learn. It also allows for object-oriented programing. It also offers a wide range of data structures that you wouldn't expect from a general-purpose language.	R's functional syntax isn't easy for beginners, but not too challenging for those well versed in programming. It also offers a few data structures, but fails to handle large amounts of data.

2

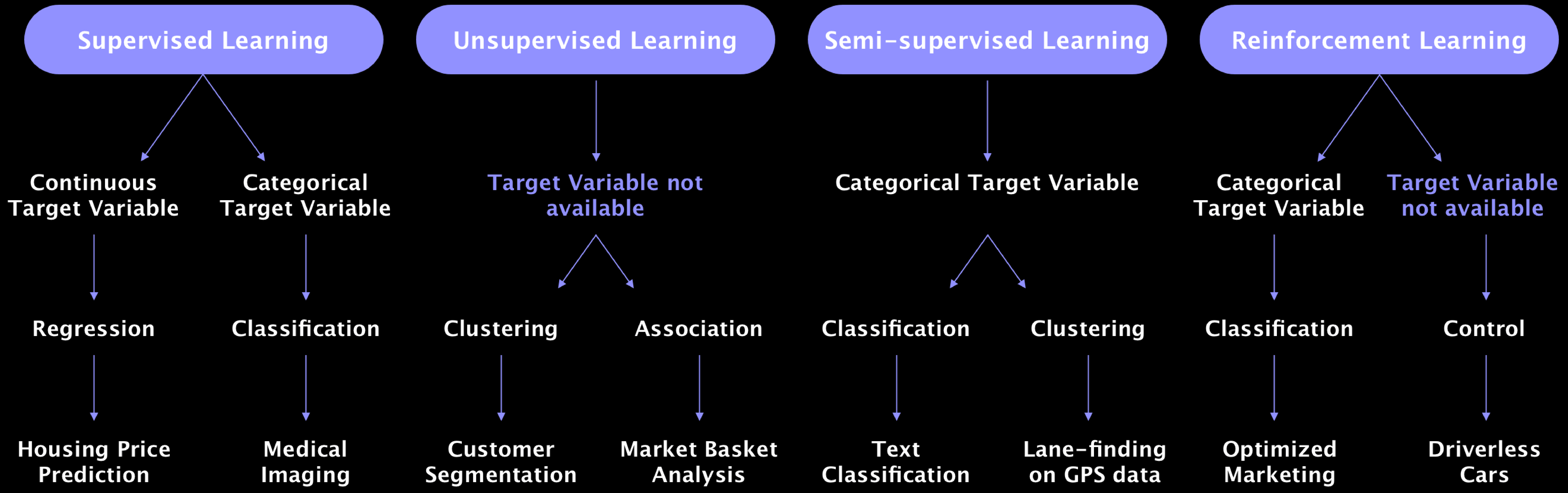
AI/ML/NLP Explained

AI Domains


AI
ML
DL
NLP
Artificial Intelligence
Machine Learning
Language Processing
Deep Learning

Machine Learning Family Tree

Machine Learning Types



Supervised Learning

Continuous Target Variable

Regression

Housing Price Prediction

Categorical Target Variable

Classification

Medical Imaging

Unsupervised Learning

Target Variable not available

Clustering

Customer Segmentation

Association

Market Basket Analysis

Semi-supervised Learning

Categorical Target Variable

Classification

Text Classification

Clustering

Lane-finding on GPS data

Reinforcement Learning

Categorical Target Variable

Classification

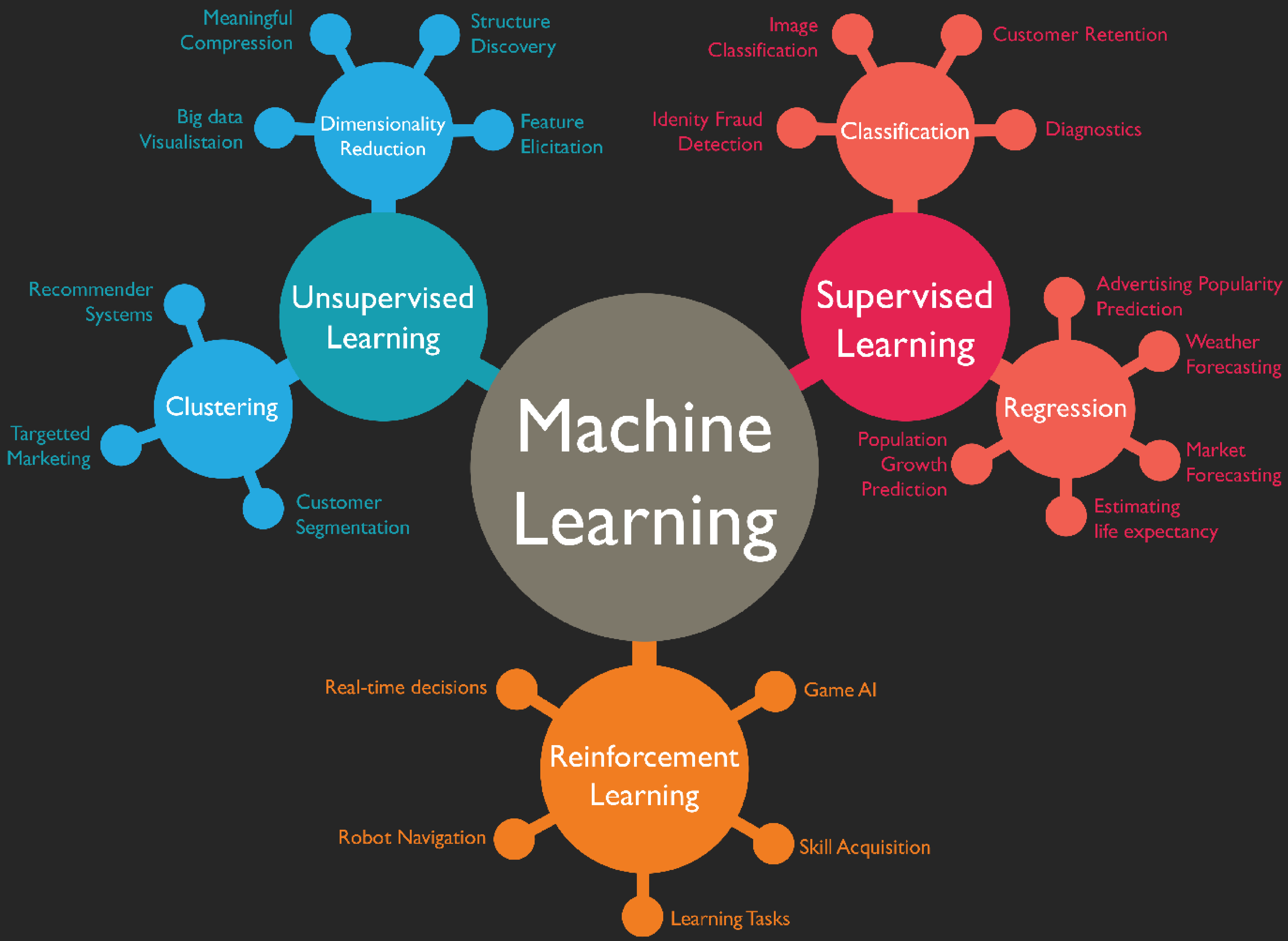
Optimized Marketing

Target Variable not available

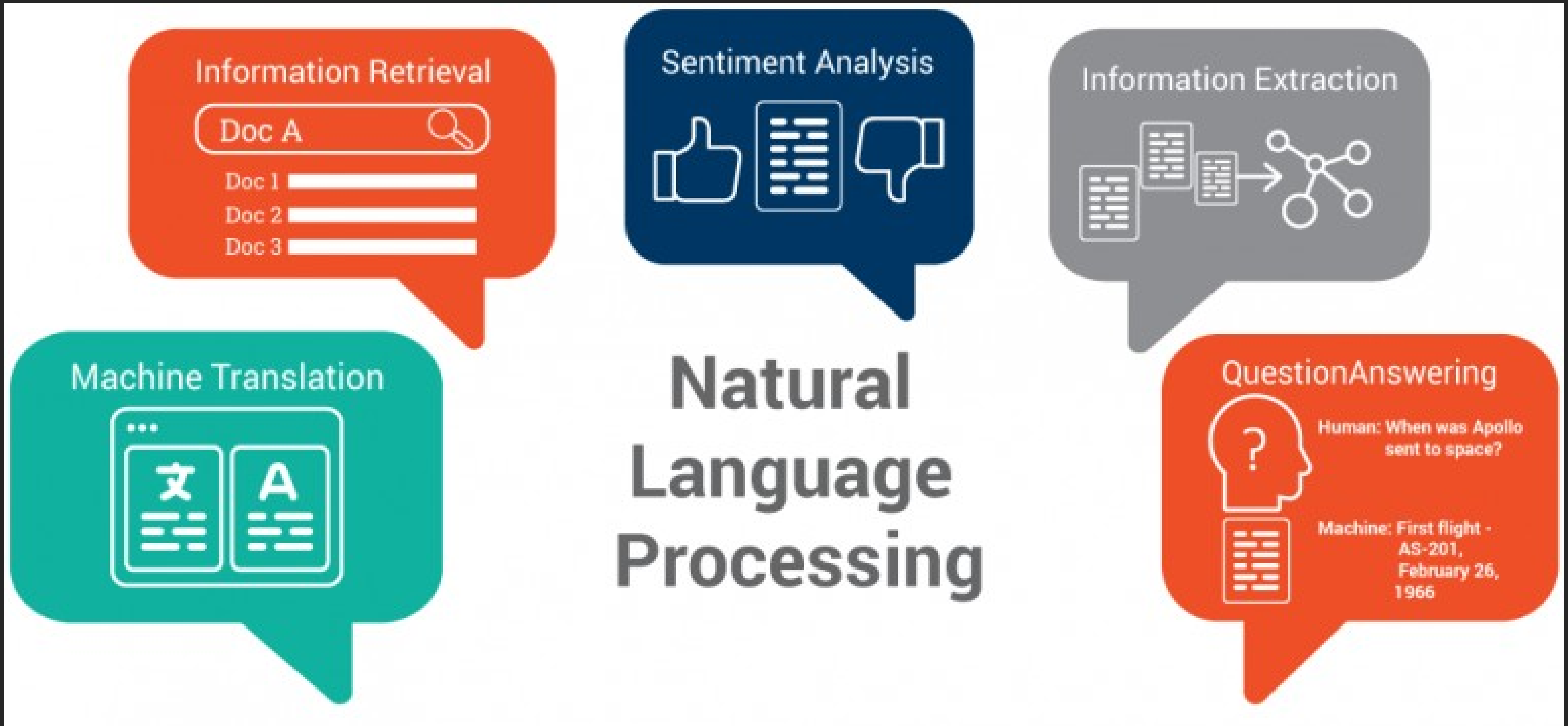
Control

Driverless Cars

Machine Learning Family Tree



Types of NLP


Information Retrieval
Doc A
Doc 1
Doc 2
Doc 3
Sentiment Analysis
Information Extraction
Machine Translation
文
A
Natural Language Processing
QuestionAnswering
?
Human: When was Apollo sent to space?
Machine: First flight - AS-201, February 26, 1966

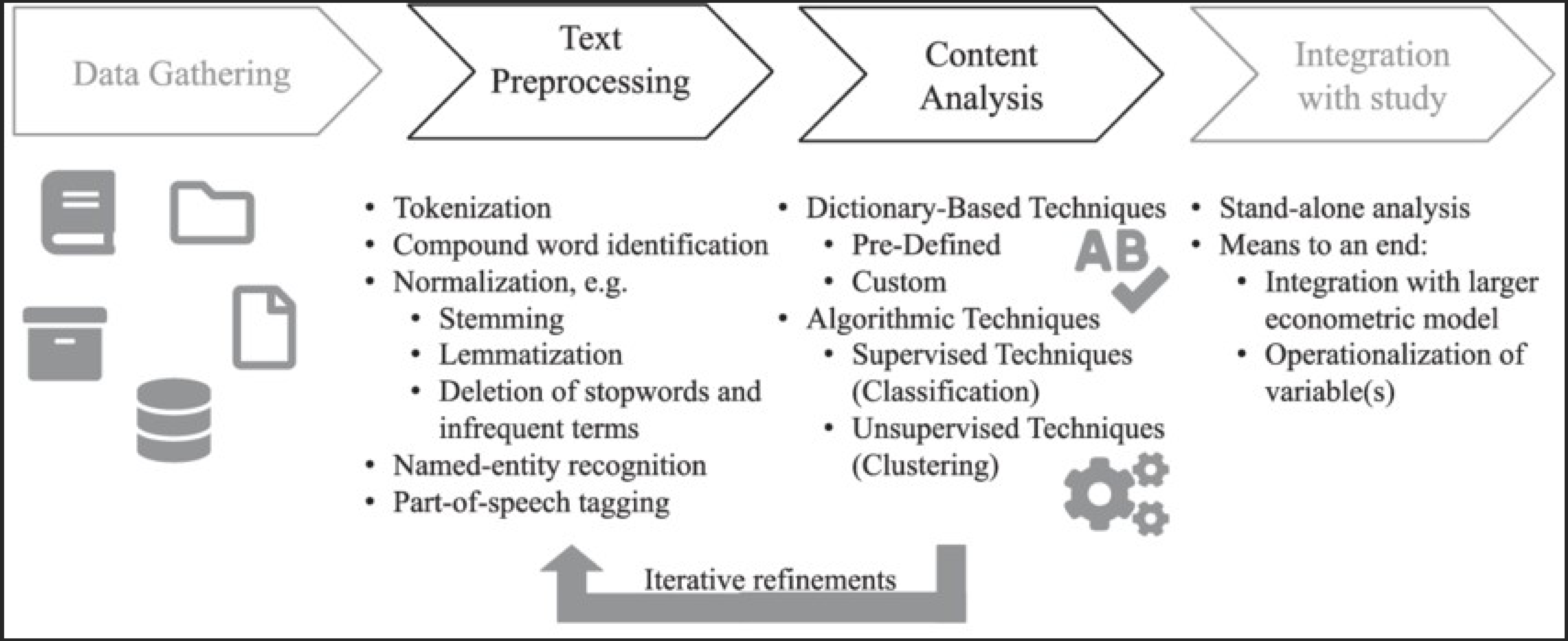
Recent Example: ChatGPT

Version	Parameters Count	Dataset
GPT-1	0.12 Billion	BookCorpus
GPT-2	1.5 Billion	WebText
GPT-3	175 Billion	CommonCrawl, WebText, English Wikipedia, Books 1 and Books 2 Corpora

3

Text Analysis

Process & Techniques



Data Gathering

Text Preprocessing

- Tokenization
- Compound word identification
- Normalization, e.g.
  - Stemming
  - Lemmatization
  - Deletion of stopwords and infrequent terms
- Named-entity recognition
- Part-of-speech tagging

Content Analysis

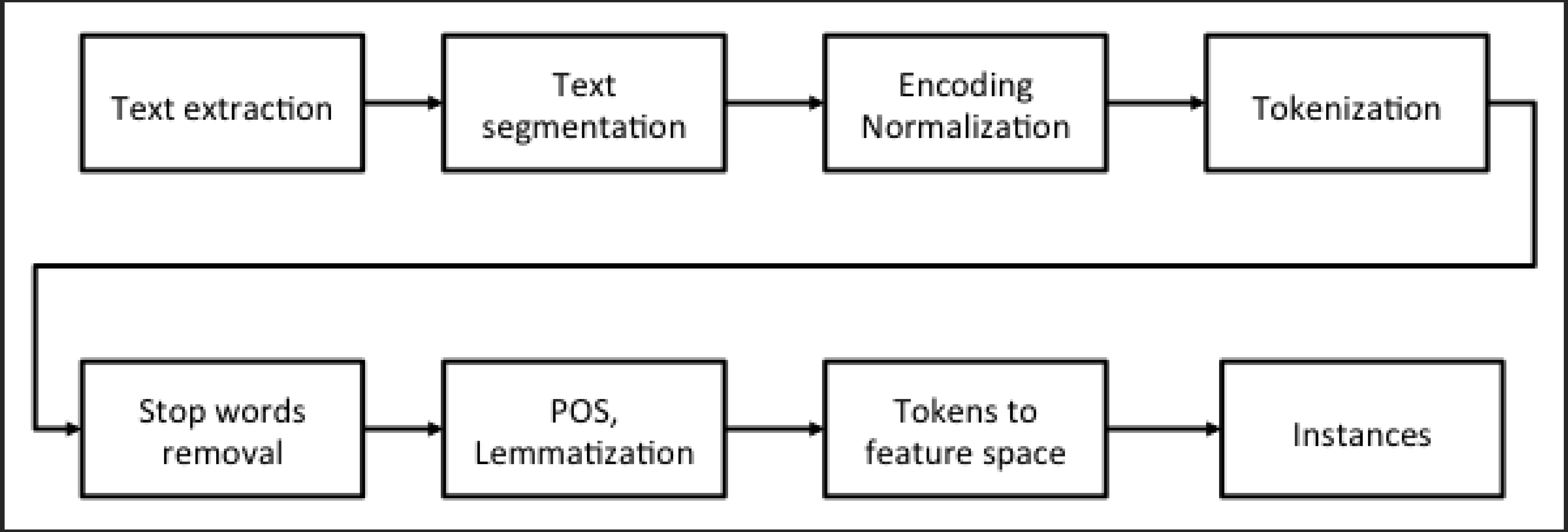
- Dictionary-Based Techniques
  - Pre-Defined
  - Custom
- Algorithmic Techniques
  - Supervised Techniques (Classification)
  - Unsupervised Techniques (Clustering)

Integration with study

- Stand-alone analysis
- Means to an end:
  - Integration with larger econometric model
  - Operationalization of variable(s)

Iterative refinements

Process & Techniques



Text extraction

Text segmentation

Encoding Normalization

Tokenization

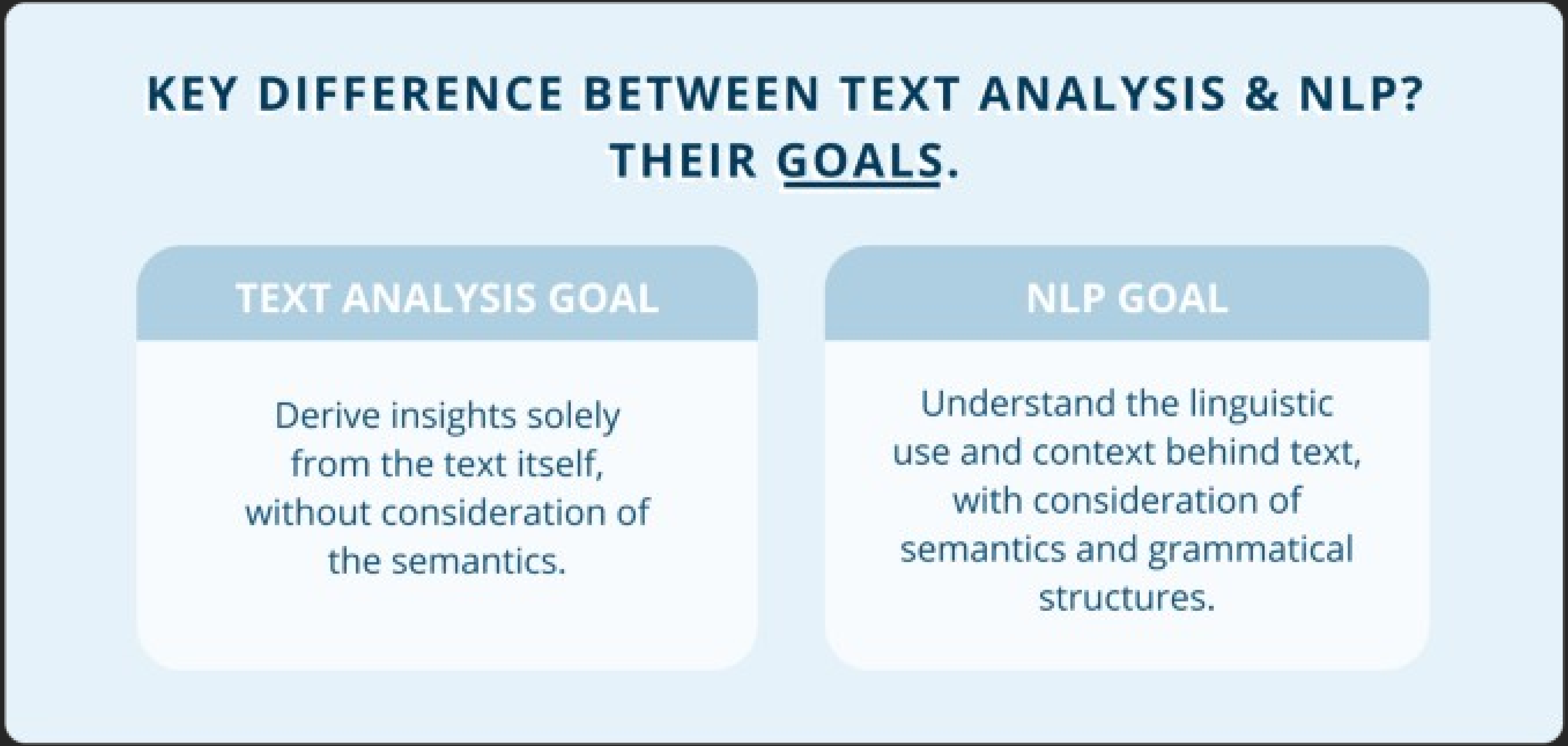
Stop words removal

POS, Lemmatization

Tokens to feature space

Instances

Goals



KEY DIFFERENCE BETWEEN TEXT ANALYSIS & NLP?
THEIR GOALS.

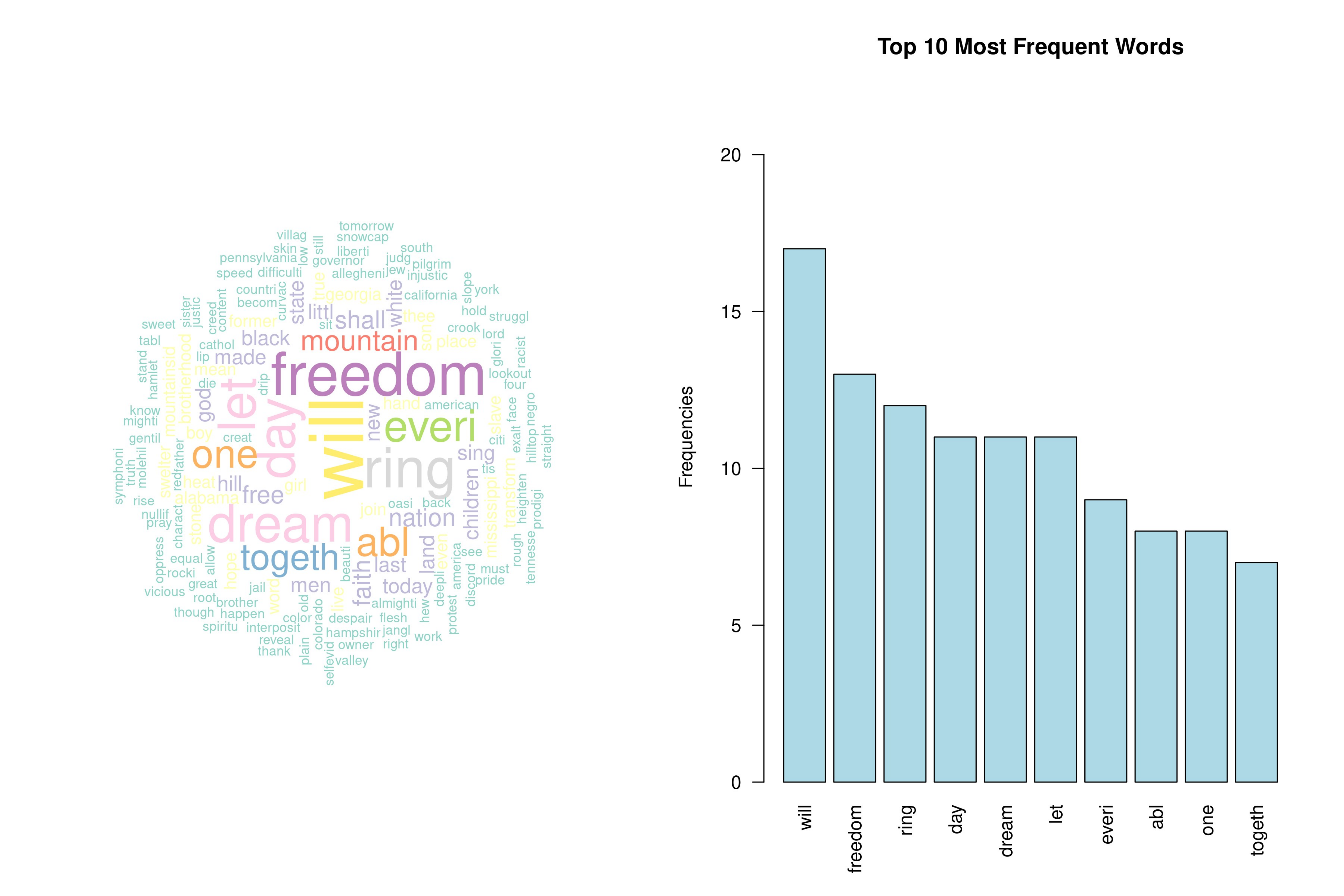
TEXT ANALYSIS GOAL

Derive insights solely from the text itself, without consideration of the semantics.

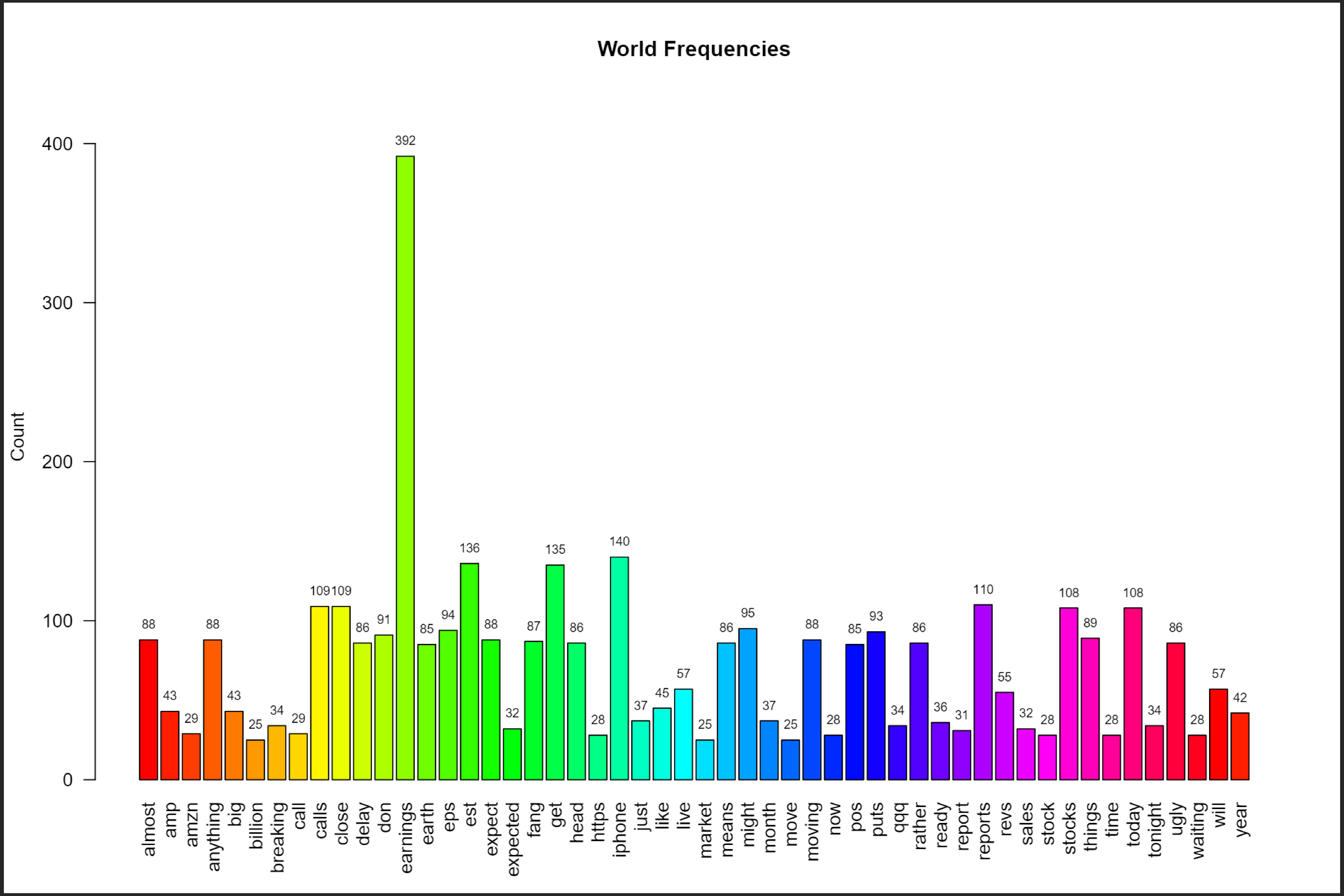
NLP GOAL

Understand the linguistic use and context behind text, with consideration of semantics and grammatical structures.

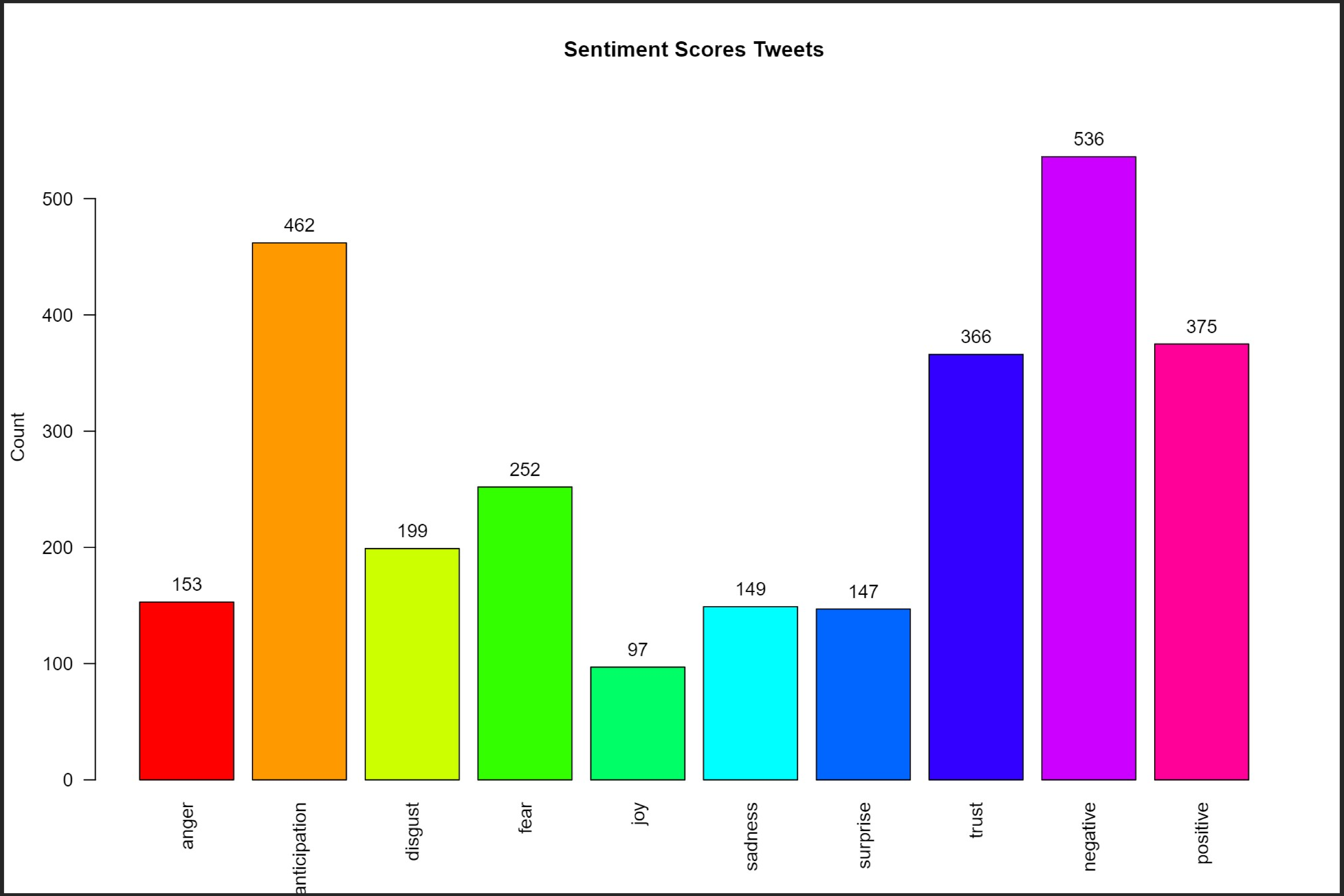
Example: Word Cloud & Word Frequency



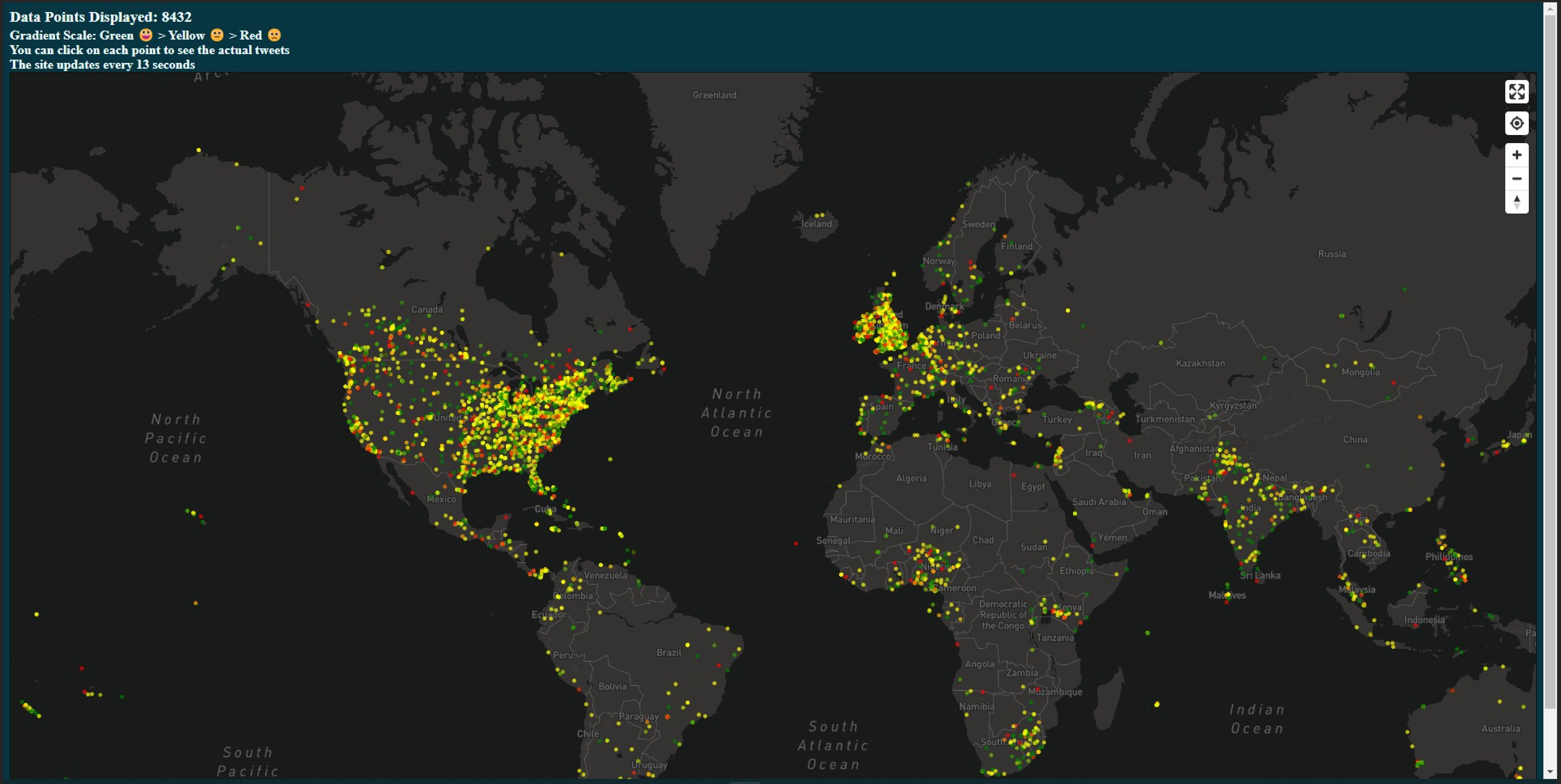
Example: Word Frequency



Example: Tweet Sentiment



Examples: Covid Sentiment Map



Data Points Displayed: 8432

Gradient Scale: Green 😀 > Yellow 😐 > Red ☹️

You can click on each point to see the actual tweets

The site updates every 13 seconds

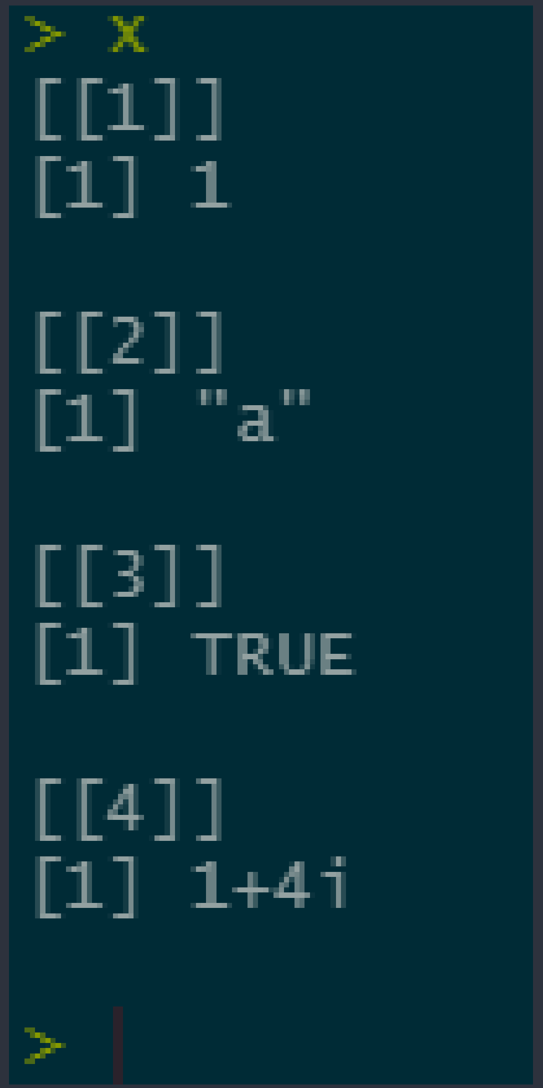
4 R Basics

R Basics

- **Data Types**
  - Character & Double & Integer & Logical & Complex
- **Data Structures**
  - Matrix & Data Frame
  - List & Vector
  - Factor
- **Variable Assignment**
  - x □ 1
  - y □ c(“apple”, ”orange”)
  - car_speeds <- read.csv(file = ‘./car-speeds.csv')
- **Accessing Data Frame**
  - Data_Frame[row, col]
  - Data_Frame$Column_Name

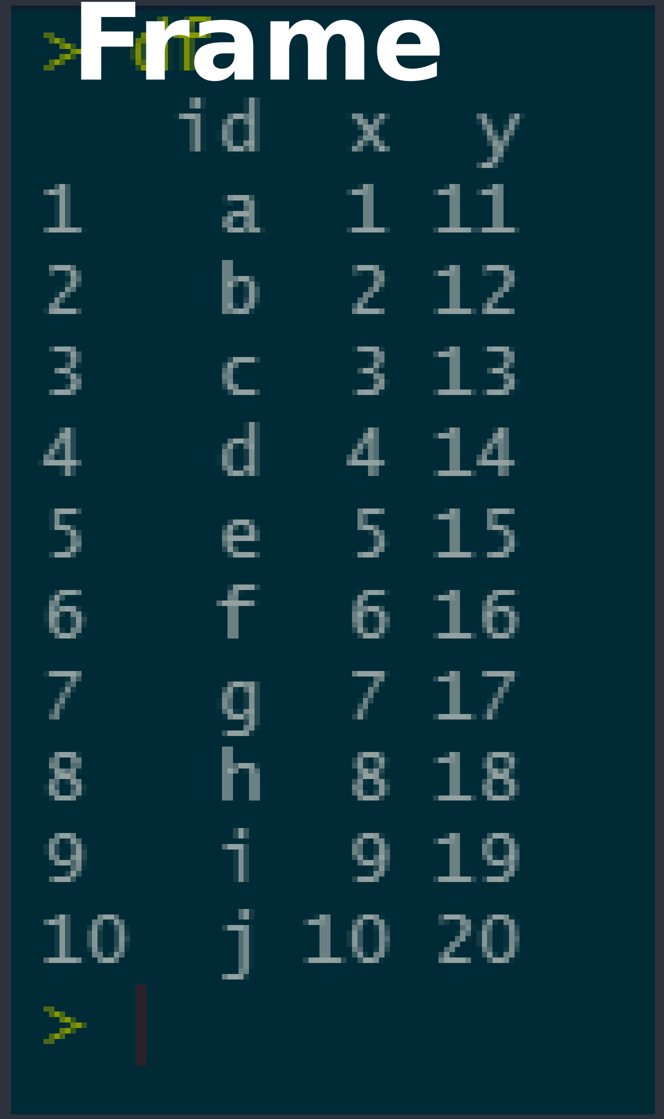
R Basics: Data Structure Examples

List



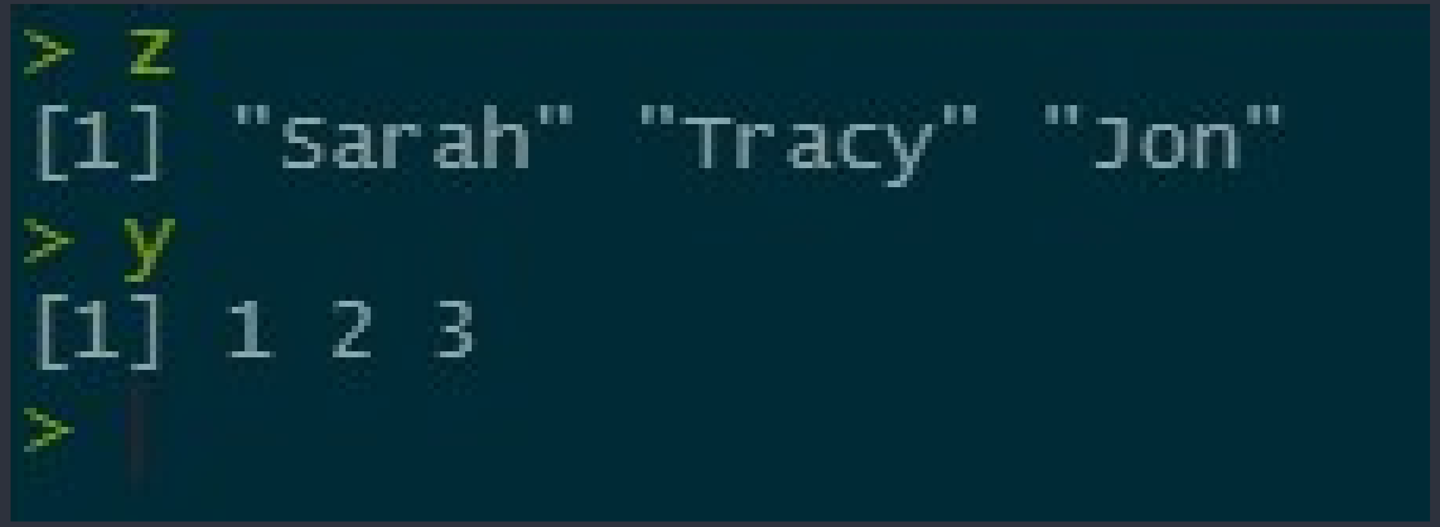
```
> x
[[1]]
[1] 1

[[2]]
[1] "a"

[[3]]
[1] TRUE

[[4]]
[1] 1+4i

> 
```

Data Frame



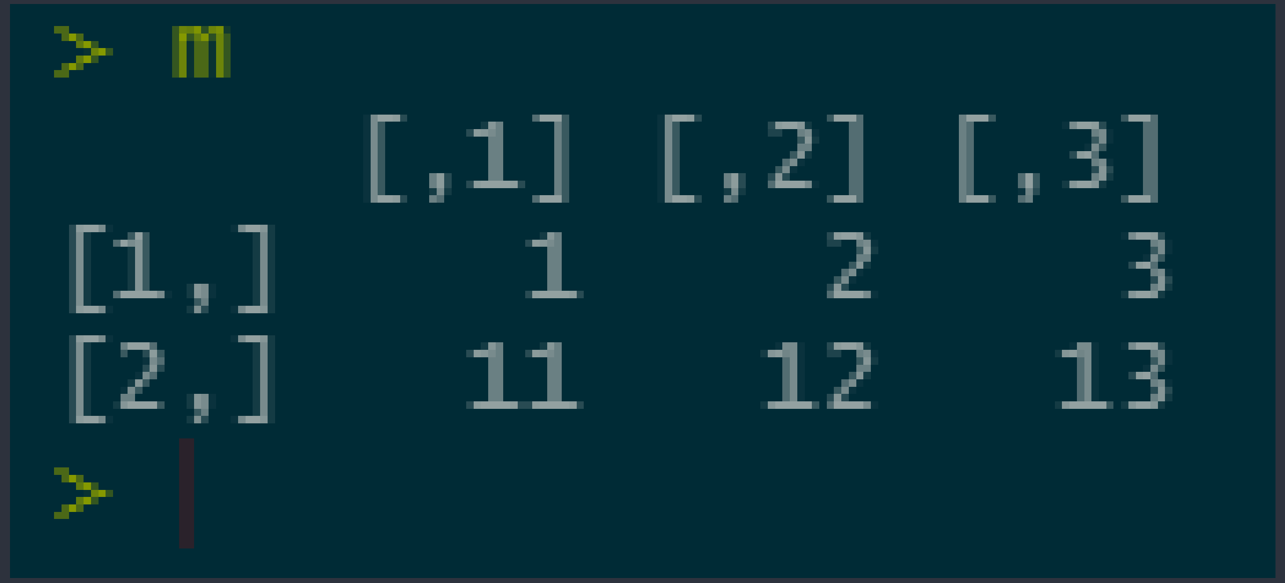
```
> df
    id  x  y
1    a  1 11
2    b  2 12
3    c  3 13
4    d  4 14
5    e  5 15
6    f  6 16
7    g  7 17
8    h  8 18
9    i  9 19
10   j 10 20
> 
```

Vector



```
> z
[1] "Sarah" "Tracy" "Jon"
> y
[1] 1 2 3
> 
```

Matrix



```
> m
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]   11   12   13
> 
```

R Basics: Exploring Your Data

- **nrow() & ncol()**
  - Gives you the number of rows of the dataset
  - Gives you the number of columns of the data set
- **head() & tail()**
  - Gives you the top 6 rows of the data set
  - Gives you the bottom 6 rows of the data set
- **str()**
  - Gives you the structure of the data set
- **dim()**
  - Gives you the dimension (row, col) of the data set
- **summary()**
  - Gives you a summary of your data set
  - Ex: min, max, mean, IQR, number of categorical variables …etc.

R

Let’s Code

Thank You