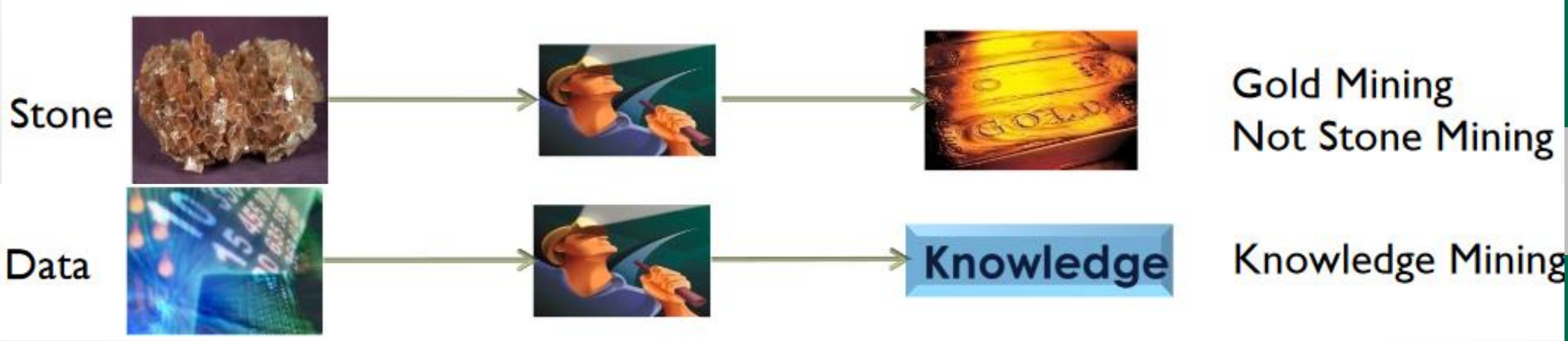
Welcome to

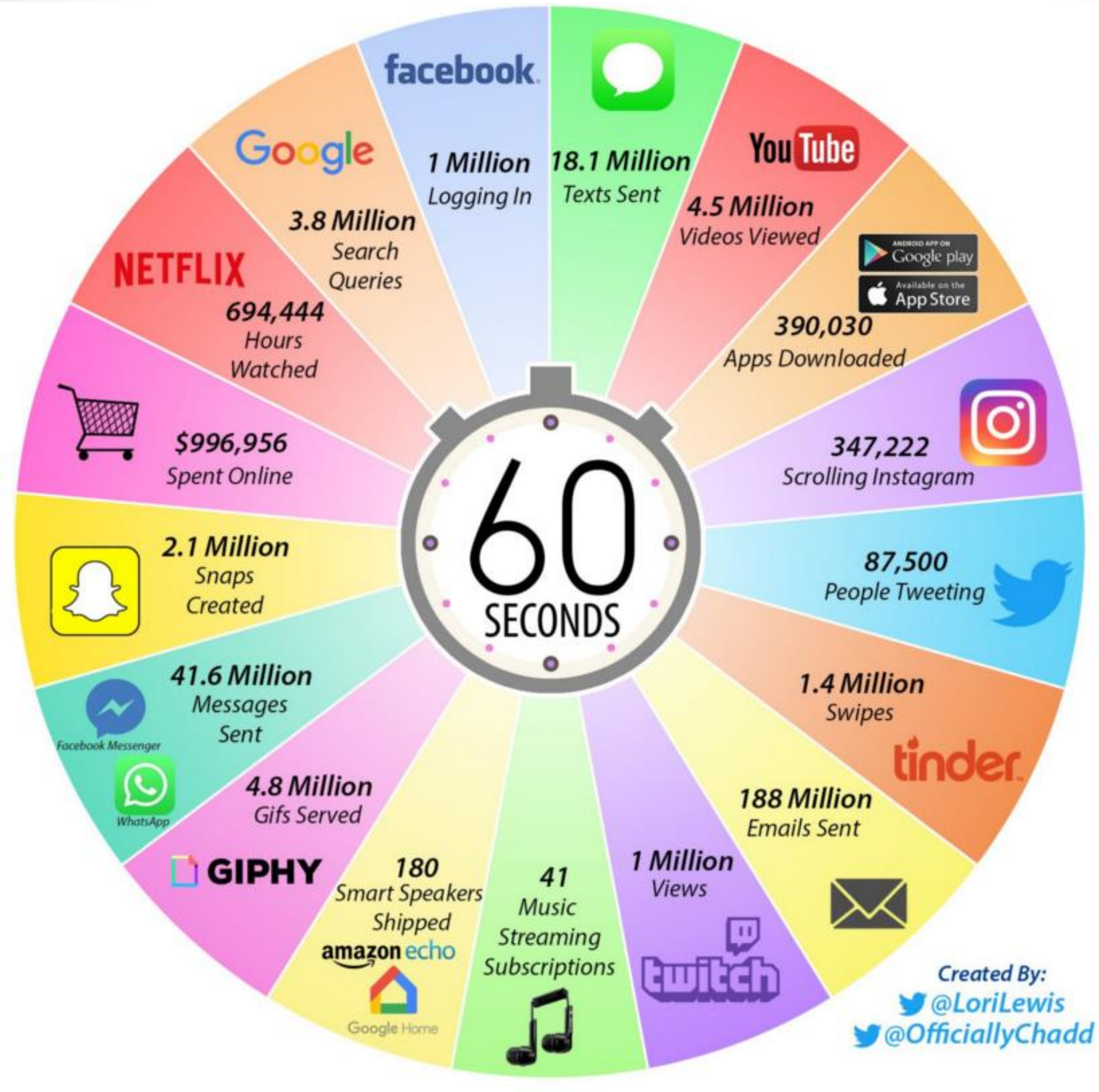
Data Mining

Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner.

* David Hand, Heikki Mannila, and Padhraic Smyth, *Principles of Data Mining*, MIT Press, Cambridge, MA, 2001

Alternative name: Knowledge Discovery in Databases (KDD).



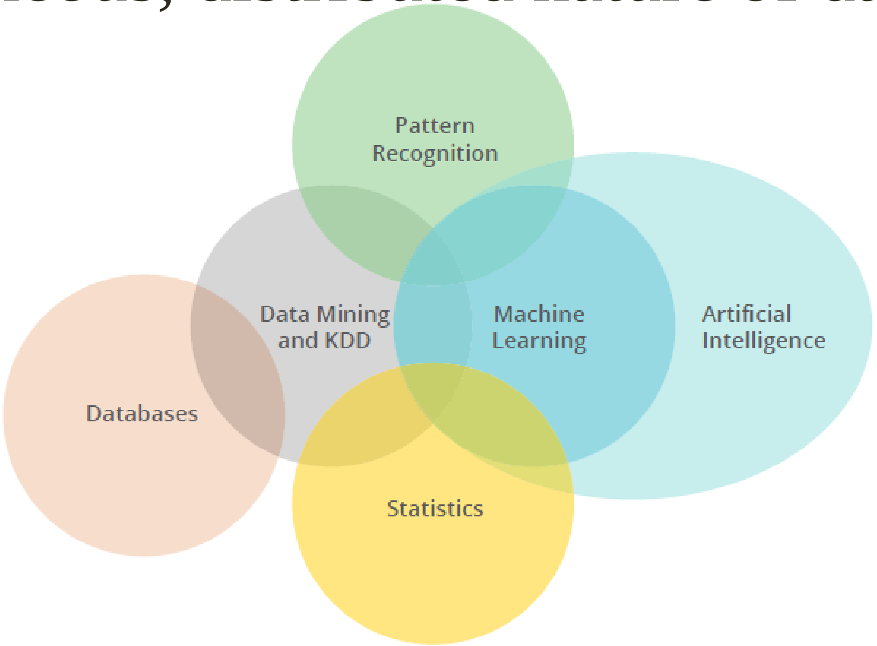


2019: What Happens in An Internet Minute

The Origins of Data Mining:

Draws ideas from machine learning, artificial intelligence (AI), pattern recognition, statistics, and database systems.

- Traditional Techniques may be unsuitable due to
  - Enormity of data
  - High dimensionality of data,
  - Heterogeneous, distributed nature of data.



Data Mining Applications:

Data mining should be applicable to any kind of information.

1. Finance:

Discovering hidden correlations between various financial indicators.

2. Marketing and Sales:

Understand the hidden patterns inside historical purchasing transaction data.

3. Education:

Predicting students’ future learning behavior, studying.

4. Biology:

Discovery of structural patterns and analysis of genetic networks and protein pathways.

5. Manufacturing Engineering:

Discovering patterns in the complex manufacturing process.

6. Energy:

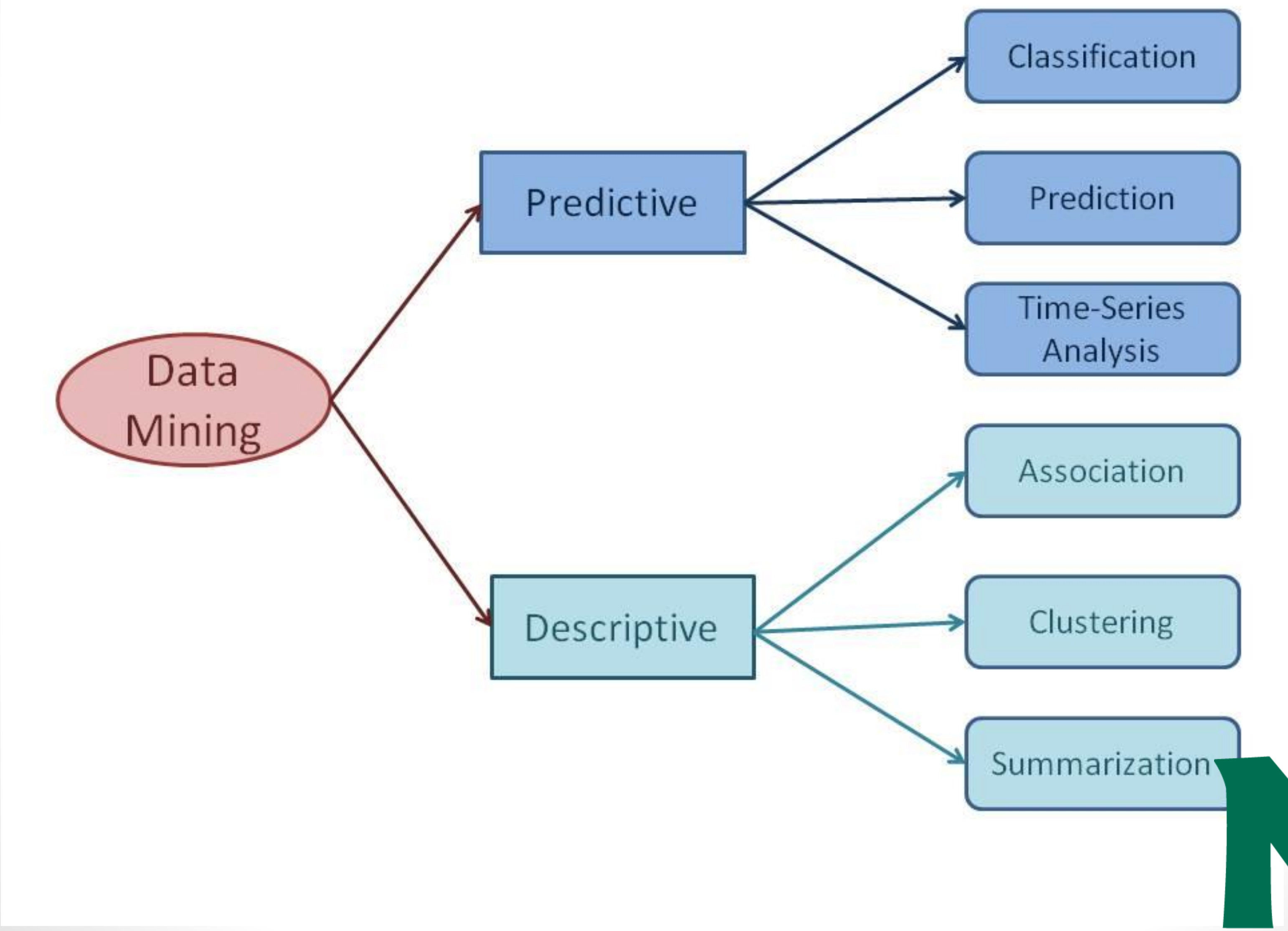
Capture weak signals of potentially threatening events.

7. Other Applications:

Healthcare, Retail, telecommunication, business, E-commerce, crime agencies, sports, etc.

Data Mining Tasks:

Data mining tasks are generally divided into two major categories:



1. Exploratory Data Analysis (EDA):

Goal:

Interact with data without clear objective.

Techniques:

- Graphical techniques (Visualization): Histogram, box-plot, scatter plot, Pareto chart, etc.
- Quantitative techniques: Summary statistics such as mean, median, correlation, etc.
- Dimensionality Reduction: Principal component analysis (PCA), Multidimensional scaling, etc.

Mean	Sum of all values / Total number of values
Median	Middle value(when data are arranged in order
Mode	Most common value

Central tendency of a distribution

Variance	**how far a set of numbers are spread out from mean**
Interquartile range	divides a data set into quartiles.
Standard deviation	dispersion of a set of data from mean

Measure of Variation

Skewness	**Measure of symmetry**
Kurtosis	**Kurtosis is a measure of "peakedness" relative to a Gaussian shape**

Skewness & Kurtosis

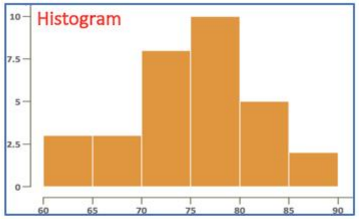
Descriptive statistics

EDA Methods

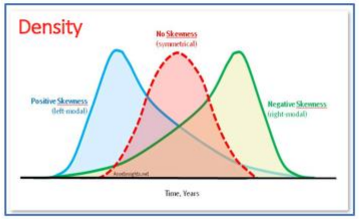
Visualizations

1-dimension

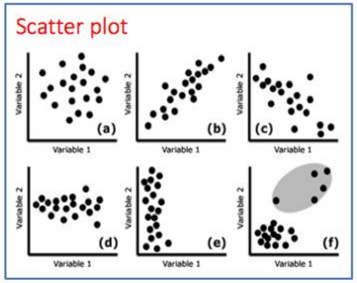
Few data points



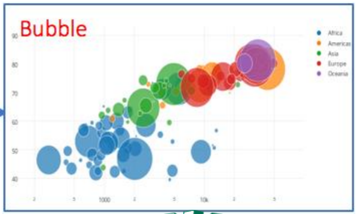
Many data points



2-dimension



3-dimension



2. Descriptive Modeling:

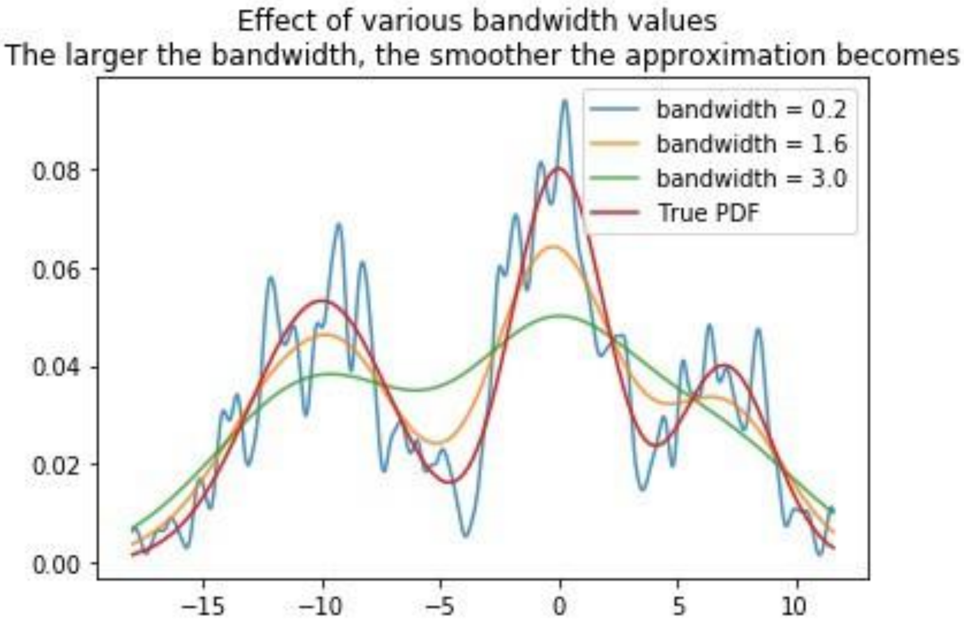
Goal:

Summarize the data or the underlying generative process. "What happened and why?"

Techniques:

We can use the Exploratory Data Analysis techniques.

- Cluster analysis, density estimation, etc.



3. Predictive Modeling:

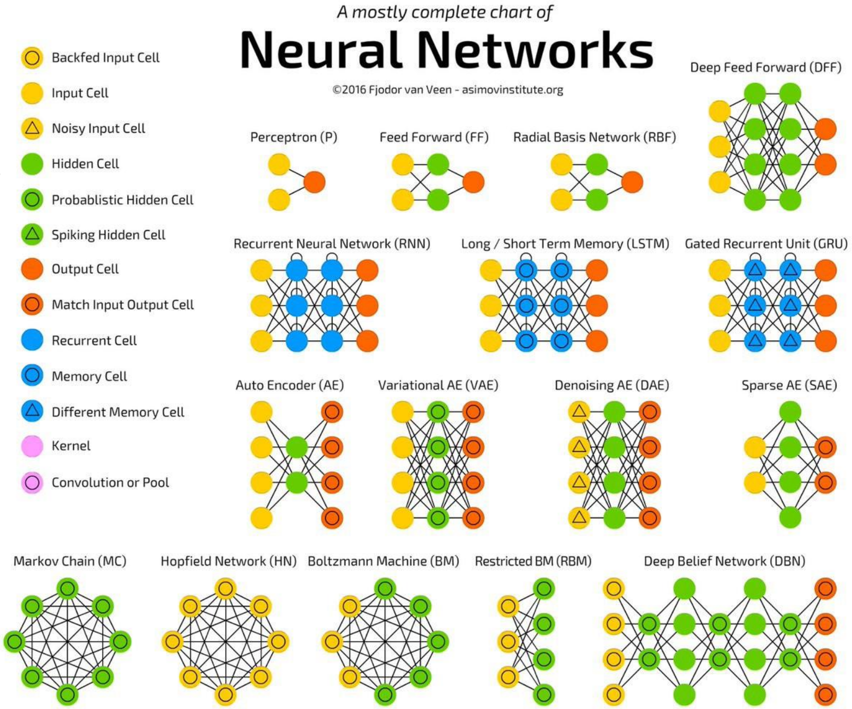
Goal:

using known results to create, process, and validate a model that can be used to forecast future outcomes.

“What will happen?”

Techniques:

- Regression analysis.
- Time series.
- Neural Networks.



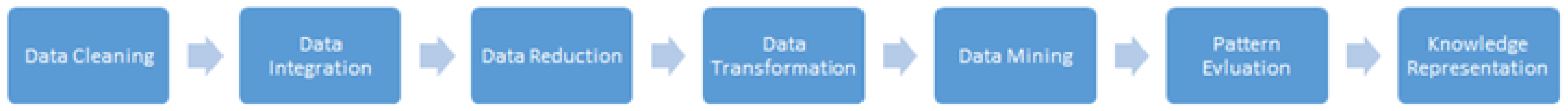
The Data Mining Process:

1. Data Cleaning (Cleansing)
2. Data Integration
3. Data Selection
4. Data Transformation

Pre-processing phase

5. Data Mining
6. Pattern Evaluation
7. Knowledge Presentation

Analytical phase



Data Cleaning → Data Integration → Data Reduction → Data Transformation → Data Mining → Pattern Evluation → Knowledge Representation

The Data Mining Process:

1. Data Cleaning (Cleansing):

Removing or modifying data that is incorrect, incomplete, irrelevant, inconsistent, duplicated, missing, or improperly formatted

2. Data Integration:

Combining data from several data sources.

3. Data Selection:

The data relevant to the analysis is decided on and retrieved from the data collection.

4. Data Transformation:

Transforming and consolidate the data into different forms that's suitable for mining.

5. Data Mining:

Applying methods to extract patterns from the data.

6. Pattern Evaluation:

Identifying the interesting patterns that represent knowledge.

7. Knowledge Presentation:

Using visualization and knowledge representation tools to present the mined data to the user.

Data Types

1. Nondependency (Independent) Oriented Data:

Refers to the simple data type. Data records do not have any specified dependencies between either the data items or the attributes.

Data can be structured so items are separate.

- Gender, ZIP code, text data.

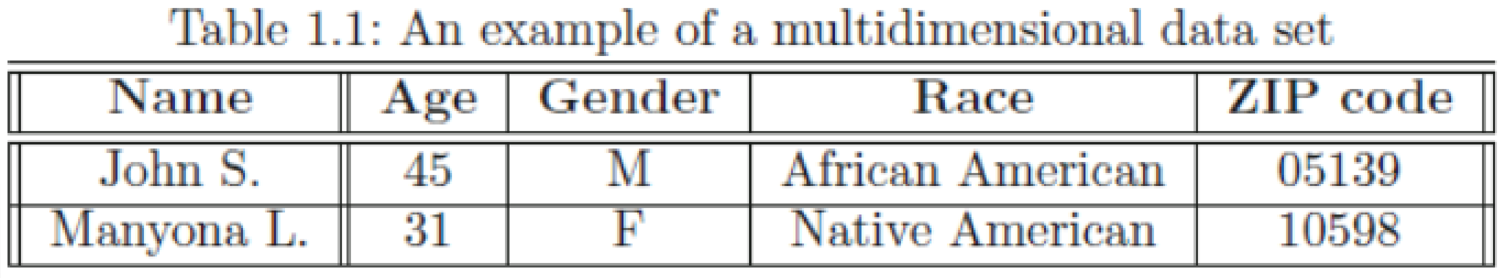


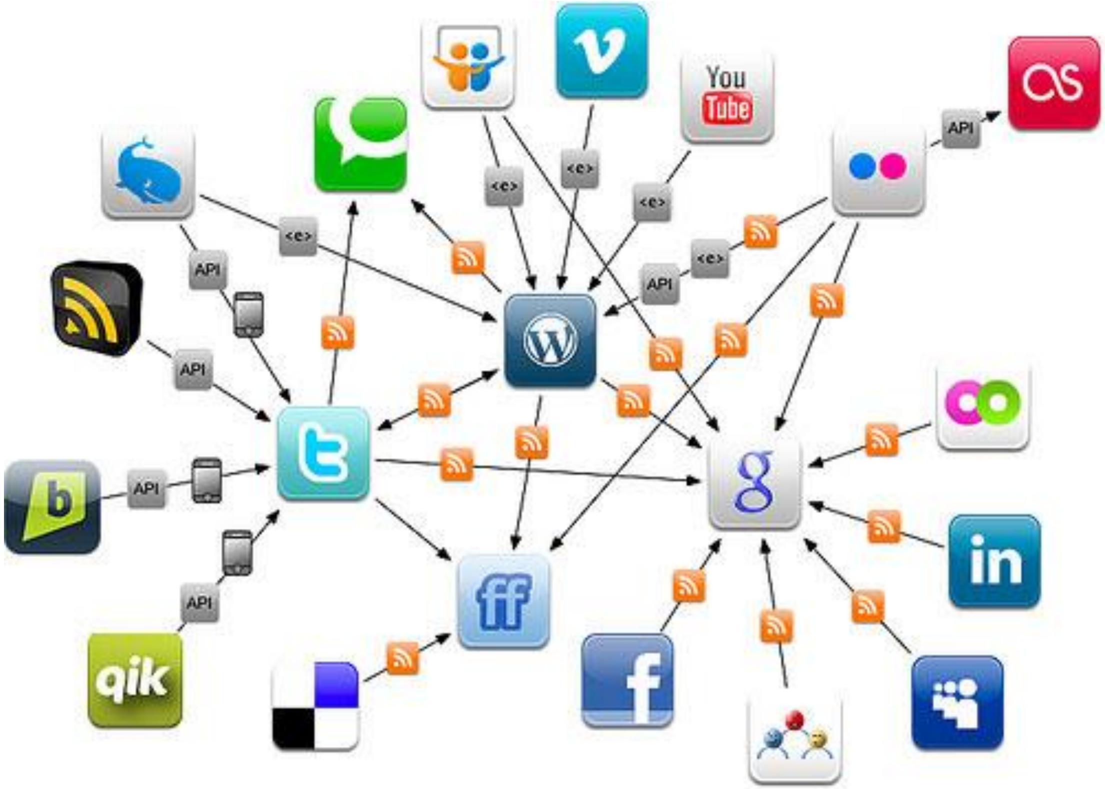
Table 1.1: An example of a multidimensional data set

Name	Age	Gender	Race	ZIP code
John S.	45	M	African American	05139
Manyona L.	31	F	Native American	10598

2. Dependency-Oriented Data:

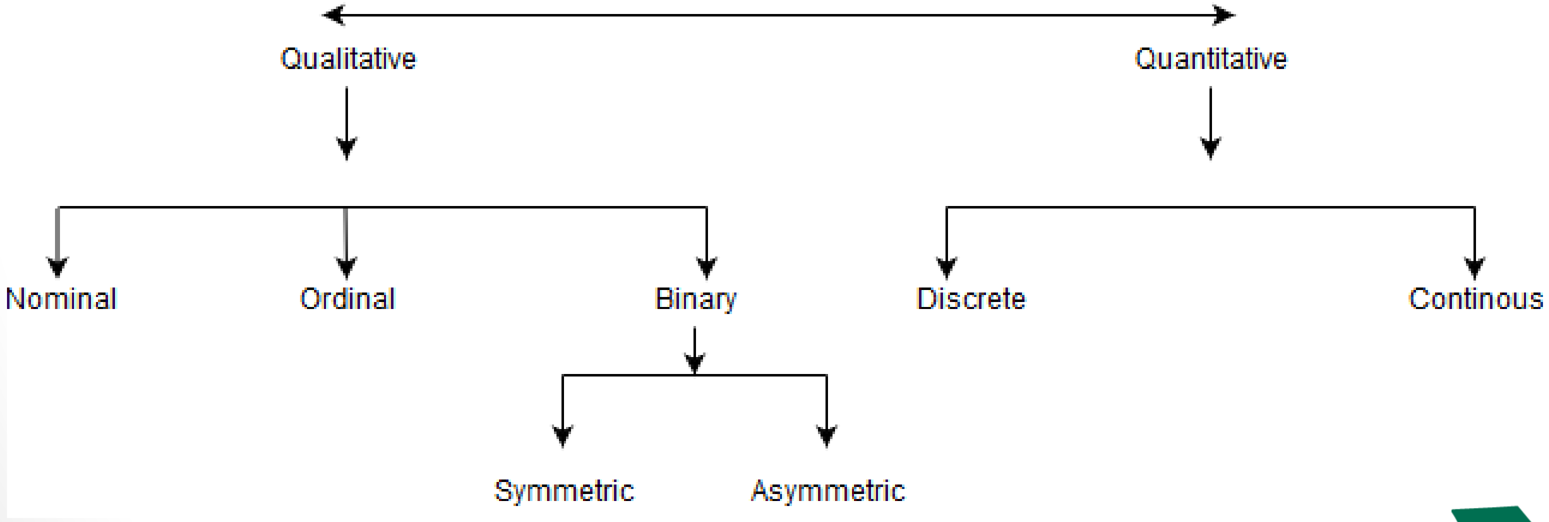
Implicit or explicit relationships may exist between data items.

- Social Network data, Time Series.



What Is an Attribute?

An **attribute** is a data field, representing a characteristic or feature of a data object. The nouns *attribute*, *dimension*, *feature*, and *variable* are often used interchangeably in the literature. The term ***dimension*** is commonly used in data warehousing.



Nominal Attributes:

Nominal means “relating to names.” The values of a nominal attribute are *symbols* or *names of things*. Each value represents some kind of category, code, or state, and so nominal attributes are also referred to as categorical. The values do not have any meaningful order. In computer science, the values are also known as enumerations.

Binary Attributes:

A **binary** attribute is a *nominal* attribute with only two categories or states: 0 or 1, where 0 typically means that the attribute is **absent**, and 1 means that it is **present**. Binary attributes are referred to as **Boolean** if the two states correspond to *true* and *false*.

- A binary attribute is **symmetric** if both of its states are equally valuable and carry the same weight; that is, there is no preference on which outcome should be coded as 0 or 1. (Gender)
- A binary attribute is **asymmetric** if the outcomes of the states are not equally important, such as the positive and negative outcomes of HIV test. We code the most important outcome, which is usually the rarest one, by 1 (e.g., HIV positive) and the other by 0 (e.g., HIV negative).

Ordinal Attributes:

An ordinal attribute is an attribute with possible values that have a meaningful order or ranking among them, but the magnitude between successive values is not known.

- Ordinal attributes may also be obtained from the discretization of numeric quantities by splitting the value range into a finite number of ordered categories.

Numeric Attributes:

A numeric attribute is quantitative; that is, it is a measurable quantity, represented in integer or real values. Numeric attributes can be *interval-scaled* or *ratio-scaled*.

Interval-Scaled Attributes:

Interval-scaled attributes are measured on a scale of equal-size units. The values of interval-scaled attributes have order and can be *positive*, *0*, or *negative*. Thus, in addition to providing a ranking of values, such attributes allow us to compare and quantify the difference between values.

- Temperatures in *Celsius* and *Fahrenheit* do not have a true zero-point, that is, neither $0^{o}C$ nor $0^{o}F$ indicates “no temperature.”

Ratio-Scaled Attributes:

A ratio-scaled attribute is a numeric attribute with an inherent zero-point. That is, if a measurement is ratio-scaled, we can speak of a value as being a multiple (or ratio) of another value.

- Temperatures in *Celsius* and *Fahrenheit* do not have a true zero-point, that is, neither $0^{o}C$ nor $0^{o}F$ indicates "no temperature."

Discrete versus Continuous Attributes:

- A discrete attribute has a finite or countably infinite set of values. The attributes hair color, smoker, medical test, and drink size each have a finite number of values, and so are discrete.
- If an attribute is not discrete, it is continuous. The terms numeric attribute and continuous attribute are often used interchangeably in the literature.

Introduction to

R Studio®

Installing R-Studio:

1. Install R : you need to install R first.

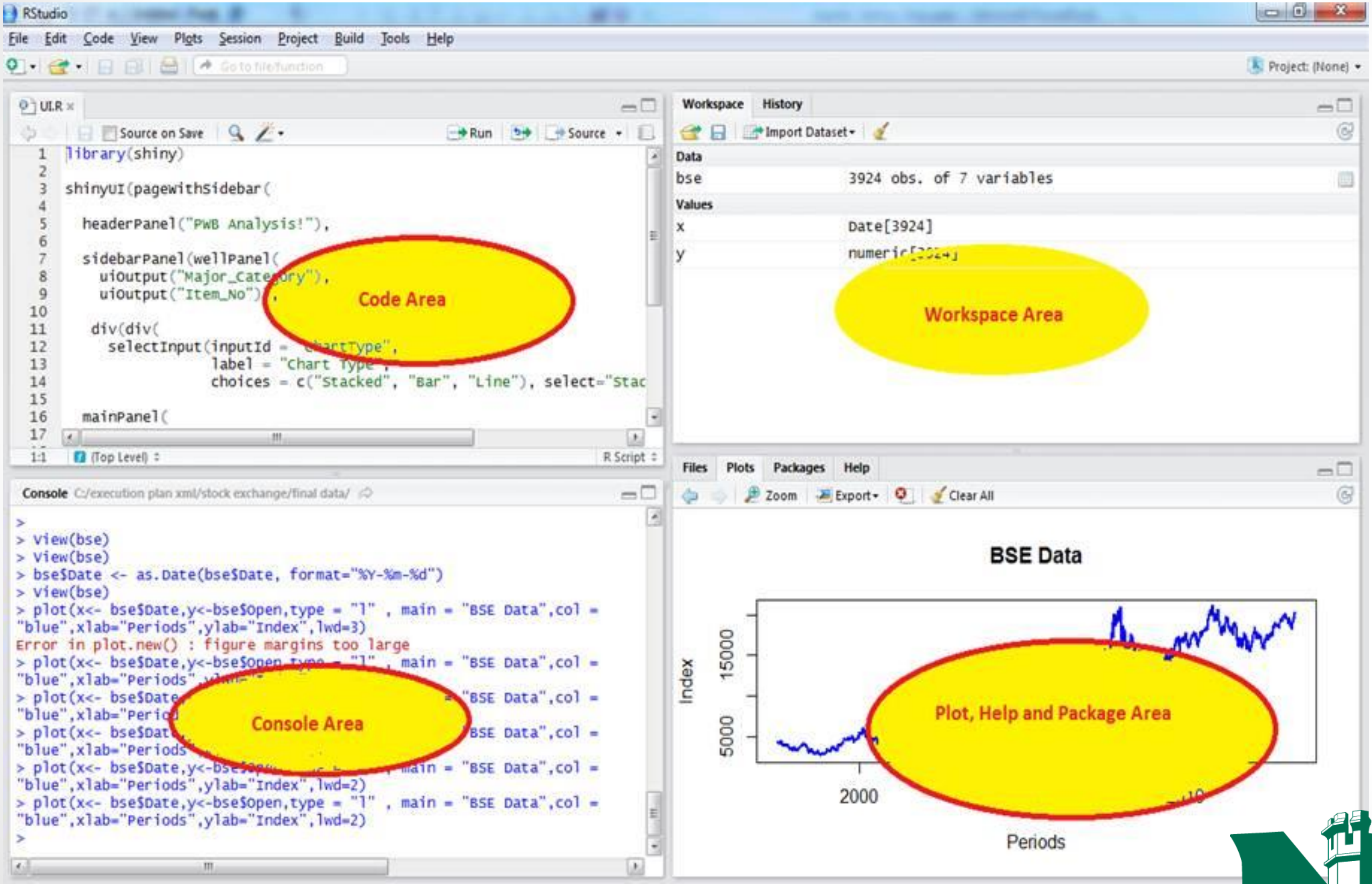
   https://cloud.r-project.org/

2. Install R-Studio:

   https://www.rstudio.com/products/rstudio/download/

Select a free version.

R-studio Screen:



```
library(shiny)

shinyUI(pageWithSidebar(

  headerPanel("PWB Analysis!"),

  sidebarPanel(wellPanel(
    uiOutput("Major_Category"),
    uiOutput("Item_No")),

   div(div(
     selectInput(inputId = "chartType",
                 label = "Chart Type",
                 choices = c("Stacked", "Bar", "Line"), select="Stac

  mainPanel(
```

Data

bse	3924 obs. of 7 variables

Values

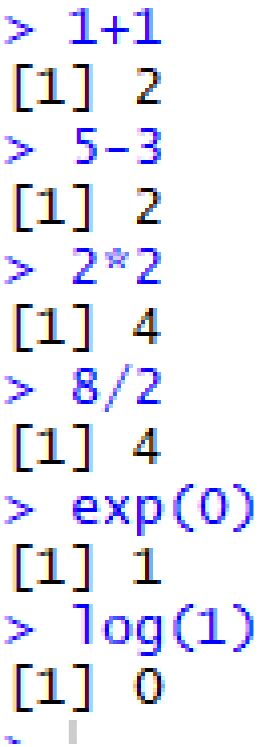
x	Date[3924]
y	numeric[3924]

```
>
> View(bse)
> View(bse)
> bse$Date <- as.Date(bse$Date, format="%Y-%m-%d")
> View(bse)
> plot(x<- bse$Date,y<-bse$Open,type = "l" , main = "BSE Data",col =
"blue",xlab="Periods",ylab="Index",lwd=3)
Error in plot.new() : figure margins too large
> plot(x<- bse$Date,y<-bse$Open,type = "l" , main = "BSE Data",col =
"blue",xlab="Periods",ylab=
> plot(x<- bse$Date                          "BSE Data",col =
"blue",xlab="Peri
> plot(x<- bse$Dat                           BSE Data",col =
"blue",xlab="Periods"
> plot(x<- bse$Date,y<-bse                   main = "BSE Data",col =
"blue",xlab="Periods",ylab="Index",lwd=2)
> plot(x<- bse$Date,y<-bse$Open,type = "l" , main = "BSE Data",col =
"blue",xlab="Periods",ylab="Index",lwd=2)
>
```

Calculator:

One of the simplest possible tasks in R is to enter arithmetic expression and receive a result.

Example:



```
> 1+1
[1] 2
> 5-3
[1] 2
> 2*2
[1] 4
> 8/2
[1] 4
> exp(0)
[1] 1
> log(1)
[1] 0
>
```

Note:

you need to click on run on the top right of R script window.

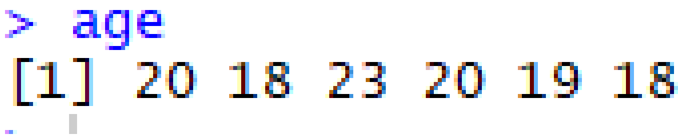
Enter raw data to R:

There are different ways to enter data to R. If we have a small sample, we can create “**vector**” of data.

Example:

```
> age = c(20, 18, 23, 20, 19, 18)
```

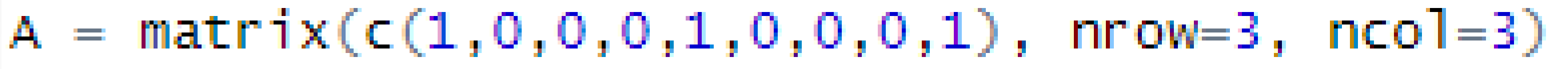
To present the data, write the variable name “age” and click on run.



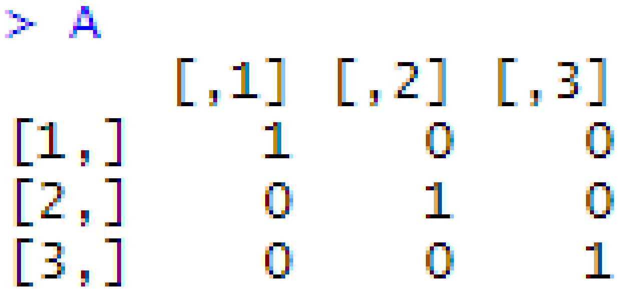
```
> age
[1] 20 18 23 20 19 18
```

Enter matrix to R:

Example:



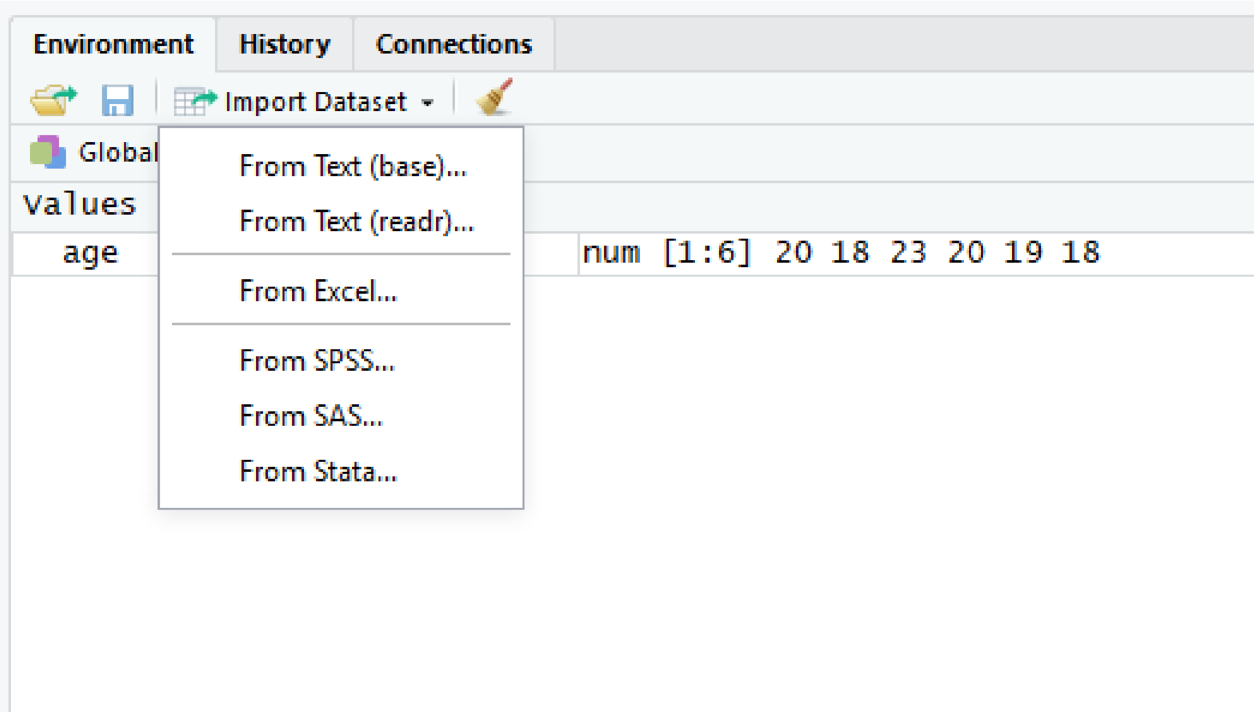
```
A = matrix(c(1,0,0,0,1,0,0,0,1), nrow=3, ncol=3)
```



```
> A
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
```

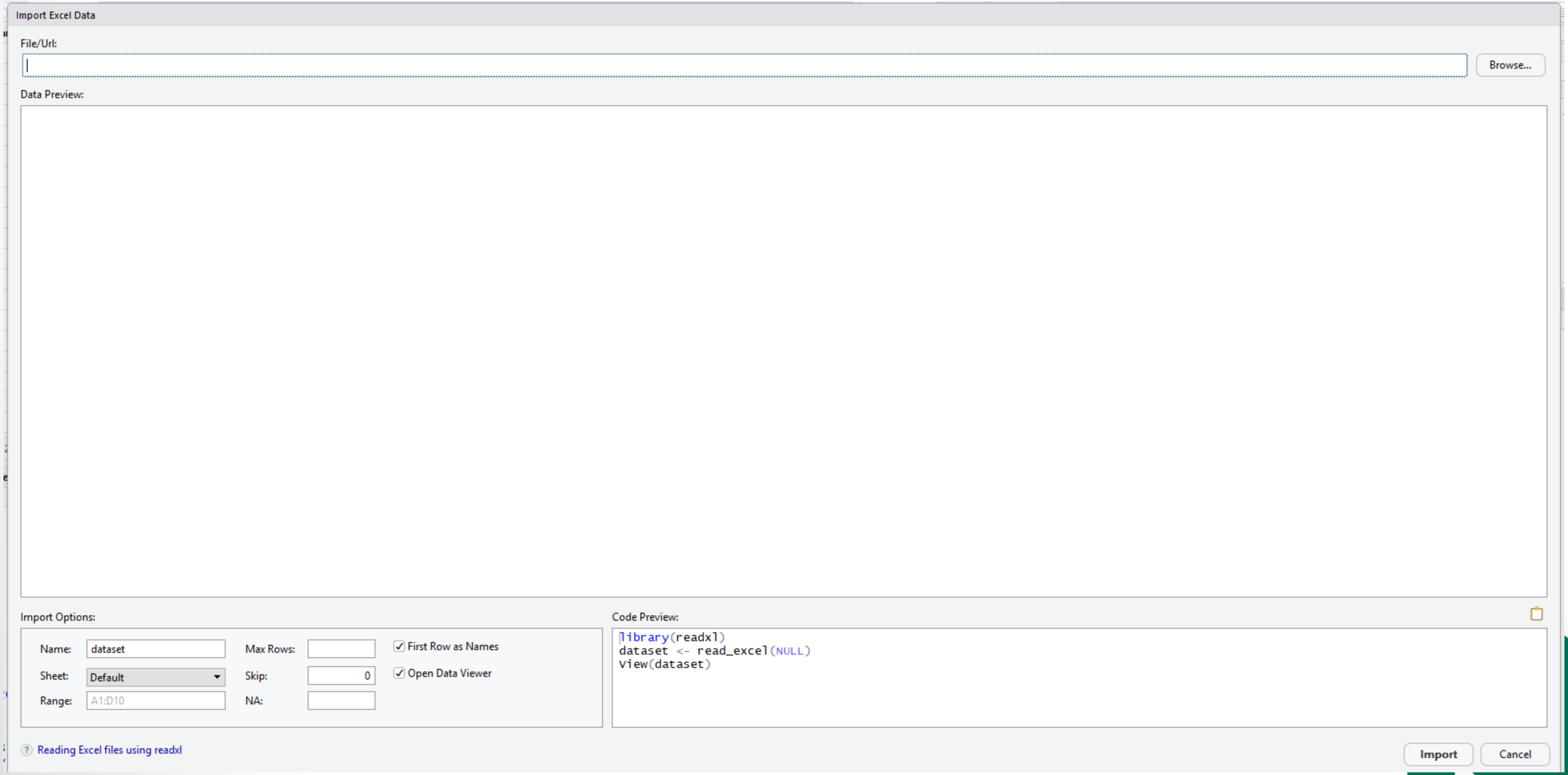
Imoprt data to R-studio:

If your data was saved in file: such as text, excel, etc. Then go the window in the top right and click on *import dataset*.



```
age    num [1:6] 20 18 23 20 19 18
```

Select the file type, then you will get the following window, click on *Browse* to select your file, and click on *Import*.



Name: dataset
Sheet: Default
Range: A1:D10
Max Rows:
Skip: 0
NA:
☑ First Row as Names
☑ Open Data Viewer

```
library(readxl)
dataset <- read_excel(NULL)
View(dataset)
```

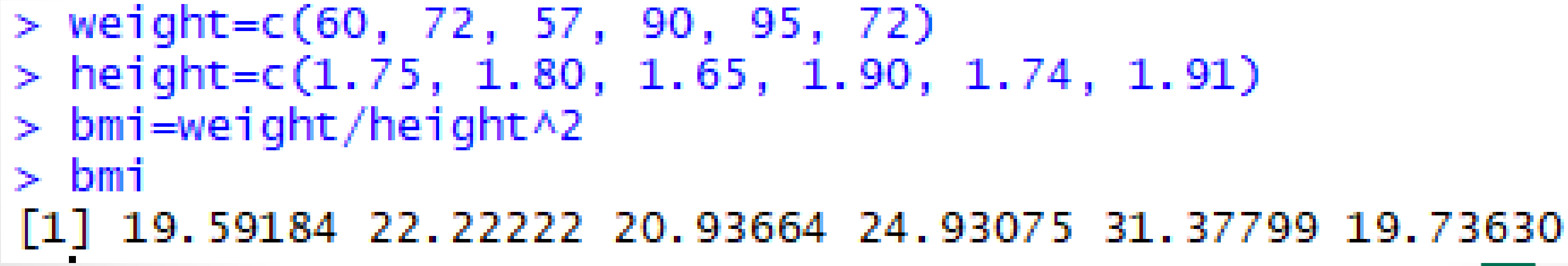
Create a new variable:

Example:

Suppose that we have a weight (kg) and height (meter) of a sample of 6 persons.

Find the body mass index which is defined for each person as the weight in kg divided by the square of the height in meter.

$$bmi = \frac{weight}{height^2}$$



```
> weight=c(60, 72, 57, 90, 95, 72)
> height=c(1.75, 1.80, 1.65, 1.90, 1.74, 1.91)
> bmi=weight/height^2
> bmi
[1] 19.59184 22.22222 20.93664 24.93075 31.37799 19.73630
```