**UNIT-1**

**Explain about list operations in R**

In R lists act as containers.

They can store dissimilar data type elements.

A list can contain another list

Different ways to create list

1. using list(): example: x=list(1,"gopi",TRUE,1+4i)

2. using as.list: example:

x=1:10

y=as.list(x)

It doesnot print out like vector.

Prints a new line for each element

Elements are indexed by double brackets

Single brackets will still return another list

x=list(1,"gopi",TRUE,1+4i)

print(x)

[[1]]

[1] 1

[[2]]

[1] "gopi"

[[3]]

[1] TRUE

[[4]]

[1] 1+4i

Giving names to elements in list

names(x)=c("Roll","name","single","favorite")

$Roll

[1] 1

$name

[1] "gopi"

$single

[1] TRUE

$favorite

[1] 1+4i

printing second element in list

print(x[2])

(or)

print(x$name)

To insert an element 98 after 3rd index

append(x,list(98),3)

merging two lists

l1=list(1,2)

l2=list("a","b")

l3=c(l1,l2)

convert list to a vector

v1=unlist(l1)

**What is Data Science?**

Data Science is a blend of various tools, algorithms, and machine learning principles with the goal to discover hidden patterns from the raw data.

Data Science Life Cycle

*Step 1: Define Problem Statement:* It is a brief description of the problem that you are going to solve.

*Step 2: Data Collection:* You need to collect the data which can help to solve the problem. Depending on the problem statement, the data collection method is broadly classified into two categories.

* First, when you have some unique problem and no related research is done on the subject. Then, you need to collect new data. This method is called as primary data collection.
* Another method is to use the data which is readily available or collected by someone else. This method is called as secondary data collection. You can collect and aggregate the data from various open-source websites such as Github, Kaggle, and datahub.

*Step 3: Data Quality Check and Remediation* :After collecting the data, one need to do a sanity check on the data. If the data is of bad quality, it can give misleading information.

*Step 4: Exploratory Data analysis:* one can use descriptive statistics such as central value measures and variability measures. Also, visualization methods such as graphs and plots can be used for extracting useful insights from data.

*Step 5: Data Modeling:* Modelling means formulating every step and gather the techniques required to achieve the solution.

* Predictive causal analytics: build a model that can perform predictive analytics on the payment history of the customer to predict if the future payments will be on time or not.
* Prescriptive analytics: The best example for this is Google’s self-driving car. The data gathered by vehicles can be used to train self-driving cars. You can run algorithms on this data to bring intelligence to it. This will enable your car to take decisions like when to turn, which path to take, when to slow down or speed up.
* Supervised machine learning model: model trained using existing data and used to predict unknown data
* Unsupervised machine learning model: model is build based on the hidden patterns present in the data.

**Explaind different types of Sources of Data in Data Mining or databases**

1. ***Flat Files***
2. ***Relational Databases***
3. ***DataWarehouse***
4. ***Transactional Databases***
5. ***Multimedia Databases***
6. ***Spatial Databases***
7. ***Time Series Databases***
8. ***World Wide Web(WWW)***
9. ***Flat Files***
   * Flat files is defined as data files in text form or binary form with a structure
   * Data stored in flat files have no relations between them.
   * Eg: CSV file.
   * **Application**: Used in DataWarehousing to store data, Used in carrying data to and from server, etc.
10. ***Relational Databases***
    * A Relational database is defined as the collection of tables
    * Physical schema in Relational databases is a schema which defines the structure of tables.
    * Logical schema in Relational databases is a schema which defines the relationship among tables.
    * Ex: SQL.
    * **Application**: Data Mining etc.
11. ***DataWarehouse***
    * A datawarehouse is defined as the collection of data integrated from multiple sources that will queries and decision making.
    * **Application**: Business decision making, Data mining, etc.
12. ***Transactional Databases***
    * Transactional databases is a collection of data organized by time stamps, date, etc to represent transaction in databases.
    * This type of database has the capability to roll back or undo its operation when a transaction is not completed or committed.
    * Highly flexible system where users can modify information without changing any sensitive information.
    * **Application**: Bank Transaction Data etc.
13. ***Multimedia Databases***
    * Multimedia databases consists audio, video, images and text media.
    * They can be stored on Object-Oriented Databases.
    * They are used to store complex information in a pre-specified formats.
    * **Application**: Digital libraries, video-on demand, news-on demand, musical database, etc.
14. ***Spatial Database***
    * Store geographical information.
    * Stores data in the form of coordinates, topology, lines, polygons, etc.
    * **Application**: Maps, Global positioning, etc.
15. ***Time-series Databases***
    * Time series databases contains stock exchange data and user logged activities.
    * Handles array of numbers indexed by time, date, etc.
    * It requires real-time analysis.
    * **Application**: eXtremeDB, Graphite, InfluxDB, etc.
16. ***WWW***
    * WWW refers to World wide web is a collection of documents and resources like audio, video, text, etc which are identified by Uniform Resource Locators (URLs) through web browsers, linked by HTML pages, and accessible via the Internet network.
    * It is the most heterogeneous repository as it collects data from multiple resources.
    * It is dynamic in nature as Volume of data is continuously increasing and changing.
    * **Application**: Online shopping, Job search, Research, studying, etc.

**How to collect data for your analysis**

Data collection is the process of accumulating data that's required to solve a problem statement.

Depending on the problem statement, the data collection method is broadly classified into two categories- primary and secondary data collection.

**Primary Data Collection**

First, when you have some unique problem and no related research is done on the subject. Then, you need to collect new data. This method is called as primary data collection.

Advantages:

1. Features are included based on the purpose of the research question or task. Not vice versa. This helps to only use meaningful data.
2. It is traceable how the variables were created.

Disadvantages:

1. It can be challenging to find suitable sources.
2. It takes a lot of time to gather the data.
3. Transforming features into the right format can be a lot of effort.
4. Access to databases like Thomson Reuters is often restricted. If your university or employer does not have a license, this kind of information can get very costly.

The primary data collection method is further classified into two types. They are

1. Quantitative Data Collection Methods
2. Qualitative Data Collection Methods

Let us discuss the different methods performed to collect the data under these two data collection methods.

Quantitative Data Collection Methods

It is based on mathematical calculations using various formats like close-ended questions, correlation and regression methods, mean, median or mode measures. This method is cheaper than qualitative data collection methods and it can be applied in a short duration of time.

Qualitative Data Collection Methods

It does not involve any mathematical calculations. This method is closely associated with elements that are not quantifiable. This qualitative data collection method includes interviews, questionnaires, observations, case studies, etc. There are several methods to collect this type of data. They are

1. Observation Method: The different types of observations are:

* Structured and unstructured observation
* Controlled and uncontrolled observation
* Participant, non-participant and disguised observation

2. Interview Method: The method of collecting data in terms of oral or verbal responses. It is achieved in two ways, such as

Personal Interview – In this method, a person known as an interviewer is required to ask questions face to face to the other person. The personal interview can be structured or unstructured, direct investigation, focused conversation, etc.

Telephonic Interview – In this method, an interviewer obtains information by contacting people on the telephone to ask the questions or views orally.

3. Questionnaire Method: In this method, the set of questions are mailed to the respondent. They should read, reply and subsequently return the questionnaire. The questions are printed in the definite order on the form. A good survey should have the following features:

1. Short and simple
2. Should follow a logical sequence
3. Provide adequate space for answers
4. Avoid technical terms
5. Should have good physical appearance such as colour, quality of the paper to attract the attention of the respondent

**Secondary Data Collection**

Another method is to use the data which is readily available or collected by someone else. This method is called as secondary data collection. You can collect and aggregate the data from various open-source websites such as Github, Kaggle, and datahub.

**Unit 2:**

**Experimental Design**

<https://www.youtube.com/watch?v=10ikXret7Lk>

Data for statistical studies are obtained by conducting either experiments or surveys. Experimental design is the branch of statistics that deals with the design and analysis of experiments.

In an experimental study, variables of interest are identified.

One or more of these variables are referred to as the **factors** of the study or **independent variables**. These variables influence another variable called response variable or **dependent variable**.

Three of the more widely used experimental designs are the completely randomized design, the randomized block design, and the factorial design.

Some more terminology:

Experiment – A procedure for investigating the effect of an experimental condition on a response variable.

Experimental units – Individuals on whom an experiment is performed (usually called subjects or participants).

Extraneous factor – A variable that is not of interest in the current study but is thought to affect the response variable.

Treatment – The process, intervention, or other controlled circumstance applied to randomly assigned experimental units.

Confounding variable - associated in a noncausal way with a factor and affects the response (usually found in experiments). Because of the confounding, we find that we can’t tell whether any effect we see was caused by our factor or by the confounding variable – or even both working together.

Direct control – Holding extraneous factors constant so that their effects are not

confounded with those of the experimental conditions.

For example, suppose we test two laundry detergents and carefully control the

water temperature at 180F. This would reduce the variation in our results due to water

temperature,

Blocking –

For example, say we were allocating players to two 5-player basketball teams from a pool of 10 children. So wouldn’t it be better to assign one 14-yearold to each team (at random) and four of the 11-year-olds to each team (at random)? When we do this the variable age is called a blocking variable. The levels of age are called blocks

Randomized block design – When randomization occurs only within blocks.

Principals of experimental design –

Randomize subjects to treatments (or of treatments to trials) to even out effects

we cannot control and/or ensure that the experiment does not systematically favor

one experimental condition over another.

Control aspects of the experiment that we know may have an effect on the

response, but that are not the factors being studied.

Replicate over as many subjects as possible. Results for a single subject are just

anecdotes.

Block to reduce the effects of identifiable attributes of the subjects that cannot be

controlled.

The best experiments are usually randomized, comparative, double-blind,

and placebo-controlled.

References:

<https://www.youtube.com/watch?v=GMqrOdCx4Yg>

<https://www.youtube.com/watch?v=10ikXret7Lk>

<https://www.madison-lake.k12.oh.us/userfiles/473/Classes/8673/Notes%20for%20Experimental%20Design.pdf>

<https://www.youtube.com/watch?v=1-wc7CNeIz8&t=314s>

**Data Attributes**

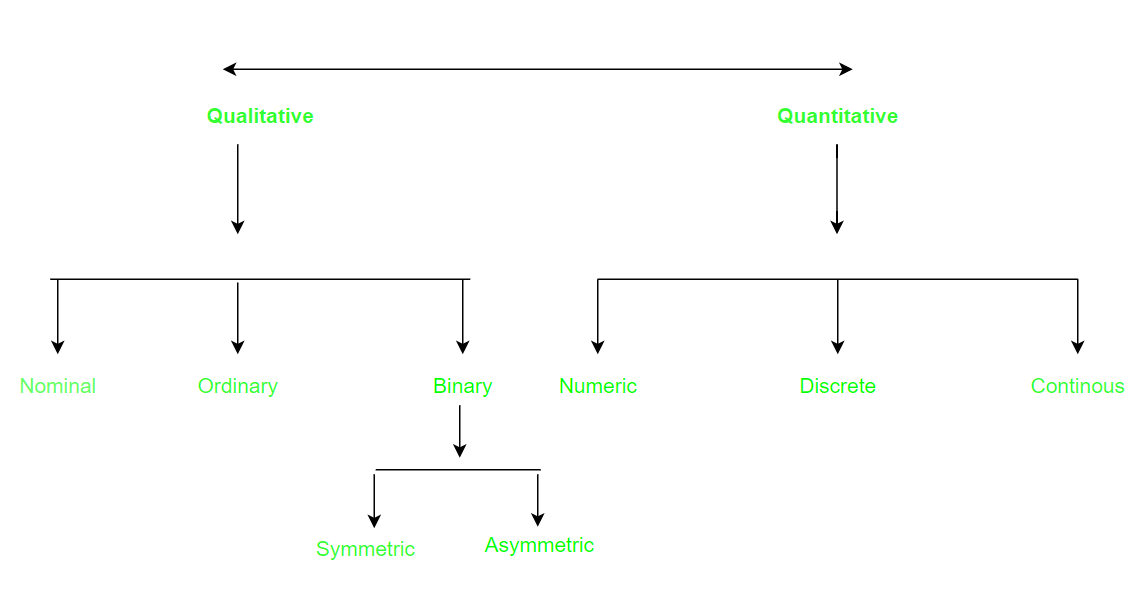
Attribute:

It can be seen as a data field that represents the characteristics or features of a data object. For a customer, object attributes can be customer Id, address, etc. We can say that a set of attributes used to describe a given object are known as attribute vector or feature vector.

Type of attributes :

This is the First step of Data Data-preprocessing. We differentiate between different types of attributes and then preprocess the data. So here is the description of attribute types.

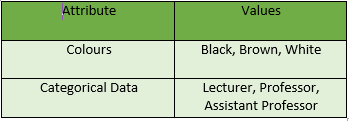
* Qualitative (Nominal (N), Ordinal (O), Binary(B)).
* Quantitative (Numeric, Discrete, Continuous)



**Qualitative Attributes:**

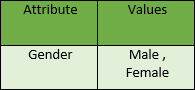
1. Nominal Attributes – related to names: The values of a Nominal attribute are names of things, some kind of symbols. Values of Nominal attributes represents some category or state and that’s why nominal attribute also referred as categorical attributes and there is no order (rank, position) among values of the nominal attribute.

Example :



2. Binary Attributes: Binary data has only 2 values/states. For Example yes or no, affected or unaffected, true or false.

* Symmetric: Both values are equally important (Gender).
* Asymmetric: Both values are not equally important (Result).



Table

Description automatically generated

3. Ordinal Attributes : The Ordinal Attributes contains values that have a meaningful sequence or ranking(order) between them, but the magnitude between values is not actually known, the order of values that shows what is important but don’t indicate how important it is.

Table

Description automatically generated

**Quantitative Attributes:**

1. Numeric: A numeric attribute is quantitative because, it is a measurable quantity, represented in integer or real values. Numerical attributes are of 2 types, interval, and ratio.

An interval-scaled attribute has values, whose differences are interpretable, but the numerical attributes do not have the correct reference point, or we can call zero points. Data can be added and subtracted at an interval scale but can not be multiplied or divided. Consider an example of temperature in degrees Centigrade. If a day’s temperature of one day is twice of the other day we cannot say that one day is twice as hot as another day.

A ratio-scaled attribute is a numeric attribute with a fix zero-point. If a measurement is ratio-scaled, we can say of a value as being a multiple (or ratio) of another value. The values are ordered, and we can also compute the difference between values, and the mean, median, mode, Quantile-range, and Five number summary can be given.

2. Discrete : Discrete data have finite values it can be numerical and can also be in categorical form. These attributes has finite or countably infinite set of values.

Example:

Table

Description automatically generated

3. Continuous: Continuous data have an infinite no of states. Continuous data is of float type. There can be many values between 2 and 3.

Example :

Table

Description automatically generated

Resources:

<https://www.geeksforgeeks.org/understanding-data-attribute-types-qualitative-and-quantitative/>

<https://www.geeksforgeeks.org/data-mining-data-attributes-and-quality/>

**Explain about Data Cleaning**

The success or failure of a project relies on proper data cleaning. **Steps involved in Data Cleaning**

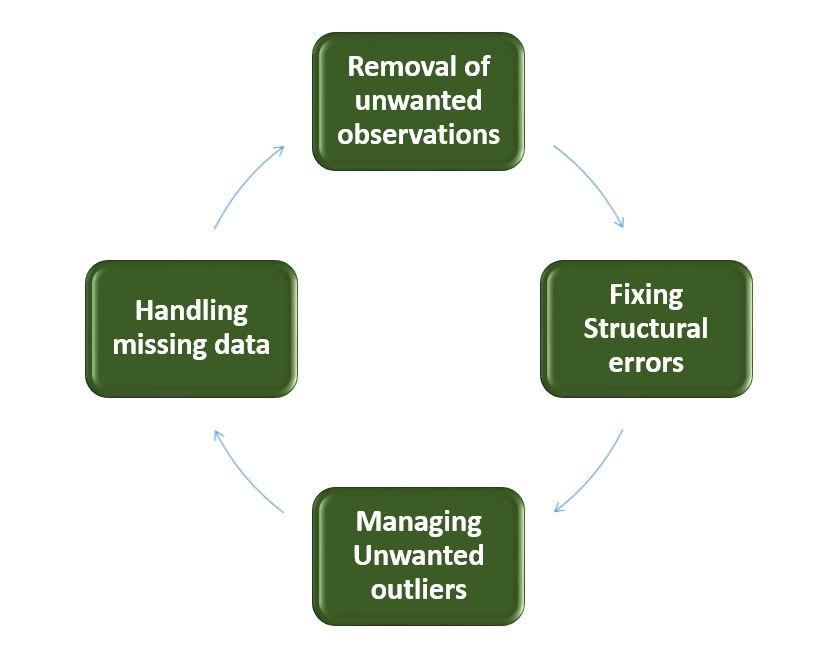
Data cleaning tasks attempts to

1.Removal of unwanted data

2. Fixing Structural Errors

3. Managing unwanted outliers

4. Handling missing data



1. **Removal of unwanted observations**

This includes deleting duplicate/ redundant or irrelevant values from your dataset.

Irrelevant observations are any type of data that is of no use to us and can be removed directly.

**2. Fixing Structural errors**

The errors that arise during measurement, transfer of data, or other similar situations are called structural errors.

For example, the model will treat red, yellow, and red-yellow as different classes or attributes, though one class can be included in the other two classes. So, these are some structural errors that make our model inefficient and give poor quality results.

3. **Managing Unwanted outliers**

Outliers can cause problems with certain types of models. For example, linear regression models are less robust to outliers than decision tree models. Generally, we should not remove outliers until we have a good reason to remove them. Sometimes, removing them improves performance, sometimes not.

**4. Handling missing data** The two most common ways to deal with missing data are:

1. Dropping observations with missing values.
2. **Fill in the missing value manually**
3. **Use a global constant to fill in the missing value:** If all missing values are replaced by “unknown”, then mining program may mistakenly think that they form an interesting concept.

So this method is simple and not foolproof.

4**. Use the attribute mean to fill in the missing value:** For example, Use average income value to replace the missing value for income.

**Some data cleansing tools**

* Openrefine
* Trifacta Wrangler
* TIBCO Clarity
* Cloudingo
* IBM Infosphere Quality Stage

**Data characterization and analysis**

Data characterization is summarization of the general characteristics of features of a target class of data.

We want to characterize the class or in other words, we can say that suppose we want to compare the classes. Now the confusing question is that What if we are not sure which attribute we should include for the class characterization or class comparison? If we specify too many attributes, then these attributes can be a solid reason to slow down the overall process of data mining.

We can solve this problem with the help of analytical characterization.

Analytical characterization is used to help and identifying the weakly relevant, or irrelevant attributes. We can exclude these unwanted irrelevant attributes when we preparing our data for the mining.

Due to the lack of an automated generalization, we must explicitly tell the system which attributes are irrelevant and must be removed, and similarly, we must explicitly tell the system which attributes are relevant and must be included in the class characterization.

Attribute generalization thresholds

Due to the lack of an automated generalization, we must explicitly tell the system how much deeper we need to generalize the attribute.

The process of generalization is totally dependent on the user who explicitly performs all these actions.

We can determine the classifying power of an attribute within a set of data with the help of a Quantitative relevance measure. Some competing methods of Relevance Measures are mentioned below;

1. Gini index
2. χ2 contingency table statistics
3. Gain ratio (C4.5)
4. Uncertainty coefficient
5. information gain (ID3)

<https://t4tutorials.com/analytical-characterization-in-data-mining/>

<https://www.geeksforgeeks.org/analysis-of-attribute-relevance-in-data-mining/>

**Unit-III**

**Data modelling and mining techniques**

<https://www.optimizationgroup.com/methods/data-modeling-data-mining/>

Model evaluation

<https://www.saedsayad.com/model_evaluation.htm>

Data visualization

<https://machinelearningmastery.com/data-visualization-in-r/>

application of data science

<https://www.edureka.co/blog/data-science-applications/>

introduction to r

<http://www.gvpcew.ac.in/Material/IT/2%20IT%20-%20UNIT-1%20Start%20Learning%20R.pdf>

<https://intellipaat.com/mediaFiles/2019/02/Data-structure-in-R-Cheat-Sheet-.png>

Graphical user interface, application

Description automatically generated

R also has many data structures. These include

* vector
* list
* matrix
* data frame
* factors (we will avoid these, but they have their uses)
* tables

**Vectors**

A vector is the most common and basic data structure in R and is pretty much the

workhorse of R. Vectors can be of two types:

* atomic vectors
* lists

Atomic Vectors A vector can be a vector of characters, logical, integers or numeric.

Create an empty vector with vector()

x <- vector()

# with a pre-defined length

x <- vector(length = 10)

# with a length and type

vector("character", length = 10)

vector("numeric", length = 10)

vector("integer", length = 10)

vector("logical", length = 10)

The general pattern is vector(class of object, length). You can also create

vectors by concatenating them using the c() function.

Various examples:

x <- c(1, 2, 3)

x is a numeric vector. These are the most common kind. They are numeric objects

and are treated as double precision real numbers.

To explicitly create integers, add a L at the end.

x1 <- c(1L, 2L, 3L)

You can also have logical vectors.

y <- c(TRUE, TRUE, FALSE, FALSE)

(Don't use T and F!)

Finally you can have character vectors:

z <- c("Alec", "Dan", "Rob", "Rich")

Examine your vector

typeof(z)

length(z)

class(z)

str(z)

Question: Do you see property that's common to all these vectors above?

Add element Annette to the existing strings in vector z

z <- c(z, "Annette")

More examples of vectors

x <- c(0.5, 0.7)

x <- c(TRUE, FALSE)

x <- c("a", "b", "c", "d", "e")

x <- 9:100

x <- c(i+0i, 2+4i)

You can also create vectors as sequence of numbers

series <- 1:10

seq(10)

seq(1, 10, by = 0.1)

Other objects

Inf is infinity. You can have positive or negative infinity.

1/0

# [1] Inf

1/Inf

# [1] 0

NaN means Not a number. it's an undefined value.

0/0

NaN.

Each object has an attribute. Attributes can be part of an object of R. These include

* names
* dimnames
* length
* class
* attributes (contain metadata)

For a vector, length(vector\_name) is just the total number of elements.

Vectors may only have one type

R will create a resulting vector that is the least common denominator.

.

xx <- c(1.7, "a")

xx <- c(TRUE, 2)

xx <- c("a", TRUE)

This is called implicit coercion.

The coersion rule goes logical -> integer -> numeric -> complex -> character.

You can also coerce vectors explicitly using the as.<class\_name>.

Example

as.numeric()

as.character()

When you coerce an existing numeric vector with as.numeric(), it does nothing.

x <- 0:6

as.numeric(x)

as.logical(x)

as.character(x)

as.complex(x)

Sometimes coercions, especially nonsensical ones won't work.

x <- c("a", "b", "c")

as.numeric(x)

as.logical(x)

# both don't work

Sometimes there is implicit conversion

1 < "2"

# TRUE

"1" > 2

# FALSE

1 < "a"

# TRUE

**Matrix**

Matrices are a special vector in R. They are not a separate class of object but simply a vector but now with dimensions added on to it. Matrices have rows andcolumns.

m <- matrix(nrow = 2, ncol = 2)

dim(m) same as attributes(m)

Matrices are constructed columnwise.

m <- matrix(1:6, nrow=2, ncol =3)

Other ways to construct a matrix

m <- 1:10

dim(m) <- c(2,5)

This takes a vector and transform into a matrix with 2 rows and 5 columns.

Another way is to bind columns or rows using cbind() and rbind().

x <- 1:3

y <- 10:12

cbind(x,y)

# or

rbind(x,y)

**List**

In R lists act as containers. Unlike atomic vectors, its contents are not restricted to a single mode and can encompass any data type. Lists are sometimes called recursive vectors, because a list can contain other lists. This makes them fundamentally different from atomic vectors.

List is a special vector. Each element can be a different class.

Create lists using list or coerce other objects using as.list()

x <- list(1, "a", TRUE, 1+4i)

x <- 1:10

x <- as.list(x)

length(x)

What is the class of x[1]? how about x[[1]]?

xlist <- list(a = "Rich FitzJohn", b = 1:10, data = head(iris))

what is the length of this object? what about its structure?

List can contain as many lists nested inside.

temp <- list(list(list(list())))

is.recursive(temp)

Lists are extremely useful inside functions. You can "staple" together lots of different kinds of results into a single object that a function can return.

It doesn't print out like a vector. Prints a new line for each element.

Elements are indexed by double brackets. Single brackets will still return another list.

**Data frame**

A data frame is a very important data type in R. It's pretty much the de facto data structure for most tabular data and what we use for statistics.

data frames can have additional attributes such as rownames(). This can be useful for annotating data, like subject\_id or sample\_id. But most of the time they are not used.

e.g. rownames() useful for annotating data. subject names. other times they are not useful.

* Data frames Usually created by read.csv and read.table.
* Can convert to matrix with data.matrix()
* Coercion will force and not always what you expect.
* Can also create with data.frame() function.

With and data frame, you can do nrow(df) and ncol(df) rownames are usually 1..n.

Combining data frames

df <- data.frame(id = letters[1:10], x = 1:10, y = rnorm(10))

> df

id x y

1 a 1 -0.3913992

2 b 2 -0.8607609

3 c 3 1.1234612

4 d 4 -0.8283688

5 e 5 -0.8785586

6 f 6 0.2116839

7 g 7 -0.3795995

8 h 8 -0.5992272

9 i 9 0.3203085

10 j 10 0.2901185

cbind(df, data.frame(z = 4))

When you combine column wise, only row numbers need to match. If you are

adding a vector, it will get repeated.

Useful functions head() - see first 5 rows tail() - see last 5 rows dim() - see

dimensions nrow() - number of rows ncol() - number of columns str() - structure

of each column names() - will list column names for a data.frame (or any object

really).

A data frame is a special type of list where every element of a list has same length.

See that it is actually a special list:

> is.list(iris)

[1] TRUE

> class(iris)

[1] "data.frame"

> --

Naming objects

Other R objects can also have names not just true for data.frames. Adding names is

helpful since it's useful for readable code and self describing objects.

x <- 1:3

names(x) <- c("rich", "daniel", "diego")

x

Lists can also have names.

x <- as.list(1:10)

names(x) <- letters[seq(x)]

x

Finally matrices can have names and these are called dimnames

m <- matrix(1:4, nrow = 2)

dimnames(m) <- list(c("a", "b"), c("c", "d"))

# first element = rownames

# second element = colnames

<https://www.javatpoint.com/verilog-scalar-and-vector>

recycling

<https://excelkingdom.blogspot.com/2018/01/what-recycling-of-vector-elements-in-r.html>

<https://www.youtube.com/c/Numyard/playlists>

Various data are usually collected for analysis purposes.

For example, demographic data (age, ethnicity, age,), employees salaries, job grade, height, student grades, curriculum. These observations can be classified as either categorical (also known as qualitative) or numeric (or quantitative) data.

Numerical data is any type of data where anything you are measuring can be characterised with a number.

These types of data can be divided into 2 subgroups: discrete and continuous.

Discrete numerical data are typically whole numbers. Discrete numerical data cannot be *measured, they are counted*. For example, since you measure your weight on a scale, it’s not discrete data. Examples include the number of people in a class, test questions answered correctly. There are two questions you can ask when deciding if data is discrete:

o Can we count the data?

o Can it divided into smaller and smaller parts?

Continuous numerical variables are variables that may contain any value within some range. Continuous data is measured. Examples of continuous data include a person’s height, weight, time in a race, temperature.

Categorical data describe a ‘characteristic’ of a data unit and are selected from a small group of categories.

Categorical data can be divided into 2 subgroups: ordinal and nominal.

With nominal data, the order does not matter, while ordinal data requires a specific order. Nominal (sometimes called labels) scales are used for labeling variables, without any quantitative value. Examples include gender (Male, Female), hair color (Brown, Black, Blonde, Gray), citizenship (American, Ivorian, French, Canadian).

Nominal data attributes can’t either be ordered or measured.

**Examples:**

* Gender (Women, Men)
* Eye color (Blue, Green, Brown)
* Hair color (Blonde, Brown, Brunette, Red, etc.)
* Marital status (Married, Single)
* Religion (Muslim, Hindu, Christian)

With ordinal scales the order of the values is the focus and most significant. Examples include assessing the temperatures: cold, cool, warm and hot.

