

Phases of Data Analysis

Ask	Define the problem you are trying to solve.
Prepare	What data do I need to solve this problem? Do I have access to obtain it?
Process	Clean the data of errors and inaccuracies.
Analyze	Perform calculations to tell a data story. Exploratory Analysis, Statistical modelling
Share	Clear visuals of the data and solution. This includes the reproducible code.
Act	Provide recommendations based on data.

File Manipulation

Get Working Directory	<code>getwd()</code>
Set Working Directory	<code>setwd()</code>
See Directory Contents	<code>dir()</code>
Create Folder	<code>dir.create("tFolder")</code>
Create File	<code>file.create("test.csv")</code>
Copy File	<code>file.copy("test.csv", "-tFolder")</code>
Edit File	<code>myedit(test.R)</code>
Delete File	<code>unlink("test.csv")</code>

Structure & Dimensions

Structure	<code>str(data)</code>
Get # of Rows & Columns	<code>dim(data)</code>
Return # of Rows	<code>nrow(data)</code>
Return # of Cols	<code>ncol(data)</code>
Return 1st 6 Rows	<code>head(data)</code>
Get Class Type	<code>class(data)</code>

Importing Data

Web Scraping	<code>con = url("http://google.com")</code> <code>htmlCode = readlines(con)</code> <code>close(con)</code>
Remote File	<code>fileUrl <- "https://website.com/data.csv"</code> <code>download.file(fileUrl, destfile = ".myData.csv", method = "curl")</code>
Import Data as Table	<code>inData <- read.table("data.csv", sep = " ", header = TRUE)</code>

Applying Functions

Apply a function over an array	<code>apply(data, Margin, Function)</code> #1=Rows 2=Cols
Apply a function to each element of list, vector, or DF and return a list	<code>lapply(data, Function)</code>
Same as lapply, but returns a vector instead	<code>sapply(data, Function)</code>
Apply a function to a subset specified by the FactorList	<code>taapply(vector, factorList, Function)</code>

Clean & Test Data

Check for NAs	<code>colSums(is.na(data))</code>
Logical NA Test	<code>all(colSums(is.na(data)) == 0)</code>
Trim Whitespace	<code>trimws(charVector)</code>
Verify Data Type	<code>class(data)</code> or <code>str(data)</code>
Find Specific	<code>test[test\$someCol %in% c("abcdefg", "hello"),]</code>

String Manipulation

Uppercase	<code>toupper(names(charVector))</code>
Lowercase	<code>tolower(names(charVector))</code>

String Manipulation (cont)

String Split	<code>strsplit(names(charVector), "\\.")</code>
Find & Replace 1st	<code>sub("_", "", names(charVector))</code>
Find & Replace All	<code>gsub("_", "", names(charVector))</code>
Get Location of Value	<code>grep("F", LETTERS)</code>
Get Value from location	<code>grep("F", LETTERS, value=TRUE)</code>
Table Count Instances	<code>table(grepl("F", LETTERS))</code>
Get Substring	<code>substr(charData, 1, 7)</code>
Paste with Space	<code>paste("Test", "Message")</code>
Paste Without Space	<code>paste0("Test", "Message")</code>

Statistics

Statistical Summary	<code>summary(data)</code>
Mean	<code>mean(data)</code>
Standard Deviation	<code>sd(vector)</code>
Variance	<code>var(vector)</code>
Range	<code>range(vector)</code>

Normal Distribution	<code>rnorm(n, mean, sd)</code>
Binomial Distribution	<code>rbinom(n, size, prob)</code>
Poisson Distribution	<code>rpois(n, size)</code>
Uniform Distribution	<code>runif(n, min=0, max=10)</code>
Exponential Distribution	<code>rexp(n)</code>

K-Means Clustering	<code>kmeans(data, centers = 3)</code>
Hierarchical Clustering	<code>hclust(dist(data))</code>