

### Working directory:

```
getwd()
find the current working directory
setwd("file/path")
change the current working directory
rm(list = ls())
removes all items from the environment
```

### Stay organized

```
remember to write a header with
a short title and
date of the thing you work on e.g.:
# Lutz_Tutorial
# How to: R
#
# I/O: data.txt
#
#last mod: 04/27/23
#
#setwd("~/Desktop/Learning-stuff")
```

## Get started

### Accessing documentation pages

```
?sd
gets you a description of the function and examples
try ?data.frame ?ggplot ?plot
```

### Download and use packages

```
install.packages(name-of-package)
to install the package
library(name-of-package)
to use data
```

## Reading and writing data in R

### Reading

```
data <- read.table("data.txt")
to read and save a text file in your working directory
```

you need to make adjustments if the file does not have the required form which includes:

- the first line should have a name for each variable
- each additional line has its first item a row label and values for each variable

**Tipp:** check `?read.table` for examples

### Writing

```
write.table(x, "data.txt")
Write a text file based on x, which can either be a matrix or a data frame
```

writing csv files:

```
write.csv("<filename>", row.names = 1)
```

**Tipp:** `?write.table`

## Data types

### Vectors

creating

```
c(2, 3, 4)           → 2 3 4
2:6                  → 2 3 4 5 6
seq(2, 3, by = 0.5)  → 2.0 2.5 3.0
rep(1:2, times = 3)   → 1 2 1 2 1 2
rep(1:2, each = 3)    → 1 1 1 2 2 2
```

selecting

```
x <- c(1, 2, 3, 4)
x[4]                  → 4
x[-4]                 → 1 2 3
x[2:4]                → 2 3 4
x[c(1, 3)]            → 1 3
x[x < 4]              → 1 2 3
```

### Variables

```
names <- c("Marazzi", "Tukey", "Cox")
age <- 40
named storage our program can manipulate
```

Logical (boolean)  
Numeric (integers or floats)  
Character (character strings)  
Factor (character strings with preset levels)

### Converting types

```
as.logical
as.numeric
as.character
as.factor
```

### Data frames

```
df <- data.frame( x = 1:3, y = c("a", "b", "c") )
a case of list where all elements are of the same length
```

View(df)

	x	y
1	1	a
2	2	b
3	3	c

### Matrices

```
matrix(c(1,2,3, 11,12,13), nrow = 2, ncol = 3, byrow)
```

### Arrays

```
array(data = NA, dim = length(data), dimnames = NULL)
```

### Alternative to loops

```
for(i in seq_along(1:10000)){x1[i]
  <- mean(rnorm(25, 0, 15))}
```

```
replicate(1000, {mean(rnorm(25, 0, 15))})
two ways to replicate a sample of means
```

```
sapply(data, mean, na.rm = TRUE)
The sapply function in R applies a function to a vector or list and
returns a vector, a matrix or an array
e.g. sapply(data.frame(x = 1:3, y = 4:6), mean) -> x
2 y 5
```

# Data Management

## Sorting Data

`mtcars[order(mpg, cyl), ]`  
`mtcars[order(mpg, -cyl), ]`  
default in the `order()` function is ascending

## Merging data

`merge(dfA, dfB, by = "ID")`  
you can merge two data frames by one or a vector of variables  
`rbind(dfA, dfB)`  
merges dataframes vertically (they must have the same variables)  
`cbind(dfA, dfB)`  
same but horizontally

## Aggregating Data

`aggregate(x, ...)`  
Splits data into subsets, computes summary statistics for each

eg. `aggregate(var1 ~ var2, data, mean)` computes the mean of each value of `var1` dependent on `var2`.

## Reshaping Data

`t(data)`  
transposes a matrix or data frame  
for more check the reshape package

## Subsetting (List output)

`df$x` → x column  
`df$[[2]]` → y column

## Subsetting (Matrix output)

`df[, 2]` → y column (a b c)  
`df[2, ]` → second row (2 b)  
`df[2, 2]` → second element in second row (b)  
`nrow(df)` → Number of rows  
`ncol(df)` → Number of columns  
`dim(df)` → Number of columns and rows

`cbind (bind columns)`

`rbind (bind rows)`

# Graphics

Bar charts `barplot(height, ...)`

Box plots `boxplot(x, ...)`

Dot plots `dotchart(x, ...)`

Scatterplots `plot(x, ...)`

Line charts `plot(x, type = "l")`

Histograms `hist(x, ...)`

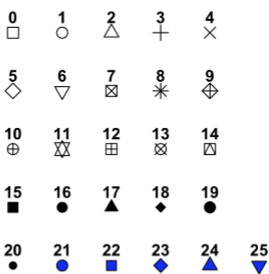
## Add text

axis labels `xlab = , ylab = ,`  
subtitle `sub`  
title `main`  
  
font face `font = 1 (plain) ...`  
font family `family = "serif"`

## Add to existing plot

## Add new plot (any)

`lines(x, ...)`  
`axis(side, ...)`  
`points(x, ...)`  
`symbol pch=`  
`mtext(text, ...)`



# Descriptive Statistics

## Correlations

`cov()` `cor()`

## T- Tests

`t.test(x ~ y) / t.test(y1, y2)`  
2-group t-test  
(check if the measurements are paired and decide between `paired = TRUE` or the default `FALSE`)

## Regression Diagnostics

`lm(x ~ y, data)`  
interprets the regression line

## ANOVA

`fit <- aov(y ~ A, data=mydataframe)`  
example Anova design (one-way Anova)

## Other functions

`mean()` `sd()` `median()` `var()` `min()` `max()` `quantile()`