



UNIVERSITY OF  
PORTSMOUTH

# R for Data Analysis

## Reading and Preparing Data in R

(TB2 - Week 8)

Atefeh Khazaei

[atefeh.khazaei@port.ac.uk](mailto:atefeh.khazaei@port.ac.uk)



# What we will learn this week?

- ❑ Reading Dataset
- ❑ Basic Statistics about Dataset
- ❑ Handling Missing Values

# Reading Datasets

## 1. Using of “datasets” package:

- ❑ This package contains a variety of datasets.
- ❑ <https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/00Index.html>

## 2. Reading data using “read” command:

- ❑ `Data <- read.csv(file = "DataFileName.csv")`
  - ❑ E.g: `census <- read.csv(file = "Census.csv", stringsAsFactors = FALSE)`

# Reading Datasets (cont.)

- ❑ Some functions to check dataset details:

- ❑ Example:

  - ❑ **head**(census)

  - ❑ **nrow**(census)

  - ❑ **ncol**(census)

  - ❑ **describe**(census)

# Required Packages

- ❑ “**funModeling**” contains a set of functions related to exploratory data analysis, data preparation, and model performance.
  - ❑ `install.packages("funModeling")`
- ❑ “**tidyverse**” is an opinionated collection of R packages designed for data science. It is useful for data analysis, high-level graphics, utility operations, functions for computing sample size and power, importing and annotating datasets, imputing missing values, advanced table making, variable clustering, and character string manipulation.
  - ❑ `install.packages("tidyverse")`
- ❑ Dependent packages to these packages are also installing.

# Required Packages (cont.)

- ❑ After installing the packages, we should import the necessary libraries.
  - ❑ **library(funModeling)**
  - ❑ **library(tidyverse)**
  - ❑ **library(Hmisc)** # it is a dependent installed library

# Important Notes

## If you have problem with install packages

- ❑ Install R and Rstudio without using Anaconda
- ❑ R language: <https://cran.r-project.org/>
- ❑ Rstudio: <https://www.rstudio.com/products/rstudio/download/>
- ❑ To add R to Jupyter in Rstudio
  - ❑ Execute `install.packages('IRkernel')` command
  - ❑ Execute `IRkernel::installspec(user=FALSE)` command

# More Data Understanding

- ❑ “**df\_status**” function to have metrics about data types, zeros, infinite numbers, and missing values.
  - ❑ `df_status(census)`
- ❑ “**glimpse**” function to explore the number of observations (rows) and variables, and a head (10 first records).
  - ❑ `glimpse(census)`
- ❑ “**freq**” function to see the basic statistical details about categorical features
  - ❑ `freq(census)`



# More Data Understanding (cont.)

- ❑ “**polt\_num**” and “**profiling\_num**” commands, to see the basic statistical details and plotting of numerical variables.
  - ❑ **plot\_num**(datasetName)
  - ❑ **profiling\_num**(datasetName)
- ❑ There are more parameter for these command that you can try them.

# More Data Understanding (cont.)

- ❑ Sorting the data frame based on any numerical column with decreasing or increasing option.
  - ❑ `NameofDataFrame[ order(NameofDataFrame$NameofColumn) , columns]`
- ❑ Example:
  - ❑ `census[ order(census$Approximated.Social.Grade, decreasing = TRUE) , ]`
    - ❑ `NameofDataFrame$NameofColumn`
    - ❑ The last coma (,) means showing all columns
  - ❑ `census[order(census$Age), c(7,17)]`
    - ❑ `c(7,17)` means only showing the 7th and 17th columns

# Handling Missing Values

- ❑ “**is\_na**” function to check if there is any missing values in the dataframe.
  - ❑ `is_na(datasetName)`
- ❑ Check the missing values for only one column.
  - ❑ `is_na(datasetName$columnName)`
- ❑ Check which rows have missing values
  - ❑ `which(is_na(datasetName))`
- ❑ Count number of missing values of each column
  - ❑ `colSums(is_na(datasetName))`

# Handling Missing Values (cont.)

□ Another example:

```
df <- data.frame(col1 = c(1:3, NA),  
                 col2 = c("this", NA, "is", "text"),  
                 col3 = c(TRUE, FALSE, TRUE, TRUE),  
                 col4 = c(2.5, 4.2, 3.2, NA),  
                 stringsAsFactors = FALSE)
```

df

A data.frame: 4 × 4

col1	col2	col3	col4
<int>	<chr>	<lgl>	<dbl>
1	this	TRUE	2.5
2	NA	FALSE	4.2
3	is	TRUE	3.2
NA	text	TRUE	NA

```
df$col2[is.na(df$col2)]='text'
```

Filling with default value.

```
df$col4[is.na(df$col4)] <- mean(df$col4, na.rm = TRUE)  
df
```

A data.frame: 4 × 4

col1	col2	col3	col4
<int>	<chr>	<lgl>	<dbl>
1	this	TRUE	2.5
2	text	FALSE	4.2
3	is	TRUE	3.2
NA	text	TRUE	3.3

Filling with mean value

# Handling Missing Values (cont.)

- ❑ Delete the rows which have missing values using
- ❑ **na.omit()** OR **na.exclude()**

```
na.omit(df)
```

A data.frame: 3 × 4

	col1	col2	col3	col4
	<int>	<chr>	<lgl>	<dbl>
1	1	this	TRUE	2.5
2	2	text	FALSE	4.2
3	3	is	TRUE	3.2

```
na.exclude(df)
```

A data.frame: 3 × 4

	col1	col2	col3	col4
	<int>	<chr>	<lgl>	<dbl>
1	1	this	TRUE	2.5
2	2	text	FALSE	4.2
3	3	is	TRUE	3.2

# References & More Resources

## □ References:

### □ Learning R:

<https://www.linkedin.com/learning/learning-r-2/>



### □ R Programming in Data Science: Setup and Start

<https://www.linkedin.com/learning/r-programming-in-data-science-setup-and-start/>

### □ To use LinkedInLearning, you can log in with your university account:

<https://myport.port.ac.uk/study-skills/linkedin-learning>



# Practical Session

- ❑ Try these slides' examples on "Titanic" dataset.
- ❑ "Titanic" dataset is available on Moodle.