



THE LONDON SCHOOL  
OF ECONOMICS AND  
POLITICAL SCIENCE ■

---

## MODULE 1 UNIT 2

### Data as the foundation for decision-making

---

## Table of contents

<b>1. Introduction</b>	<b>3</b>
<b>2. Different types of data</b>	<b>3</b>
2.1 Classes	4
2.1.1 Categorical data	4
2.1.2 Numerical data	6
2.2 Structure	6
<b>3. Data transformation</b>	<b>8</b>
<b>4. R programming language</b>	<b>9</b>
4.1 Importing data and packages into R	11
4.2 Transforming data	12
4.3 Fixing simple data issues	13
4.4 Plots	14
4.5 Generating an output	15
4.6 Practical example	15
<b>5. Conclusion</b>	<b>16</b>
<b>6. Bibliography</b>	<b>17</b>

**Learning outcomes:****LO3:** Recognise the different types of data.**LO4:** Interpret given data through suitable visualisations.**LO5:** Analyse data in R in preparation for machine learning applications.

## 1. Introduction

In modern society, almost all activities generate data. Every time people consult a search engine, use a navigation application, visit a website, post a picture on social media, order a meal, or watch a new movie, they generate data. By engaging with technology, personal data such as age, home address, workplace, employment status, gender, buying behaviour, buying power, personal circumstances, and interests is generated.

Due to this data surge, it can be extremely valuable for a business to use these data points to its advantage and drive revenue. However, this data is “like a raw diamond”, because it “must be processed in order to become valuable” to provide relevant insight (Tjønn, 2018).

This is where machine learning can be a powerful tool, as these data points are processed and used to drive business operations. The predictions calculated by machine learning models can be used to inform business decisions to ensure success. To do this, raw data must first be classified into a specific class or type of data. For that data to inform a business decision, it must be contextualised and interpreted (Tjønn, 2018). This contextualisation can be done by fitting a machine learning model onto the data set.

In this set of notes, a distinction is made between different types of data, which is essential to understanding which machine learning model to fit onto a given data set. This is followed by an introduction to R, the programming language that will be used throughout this course.

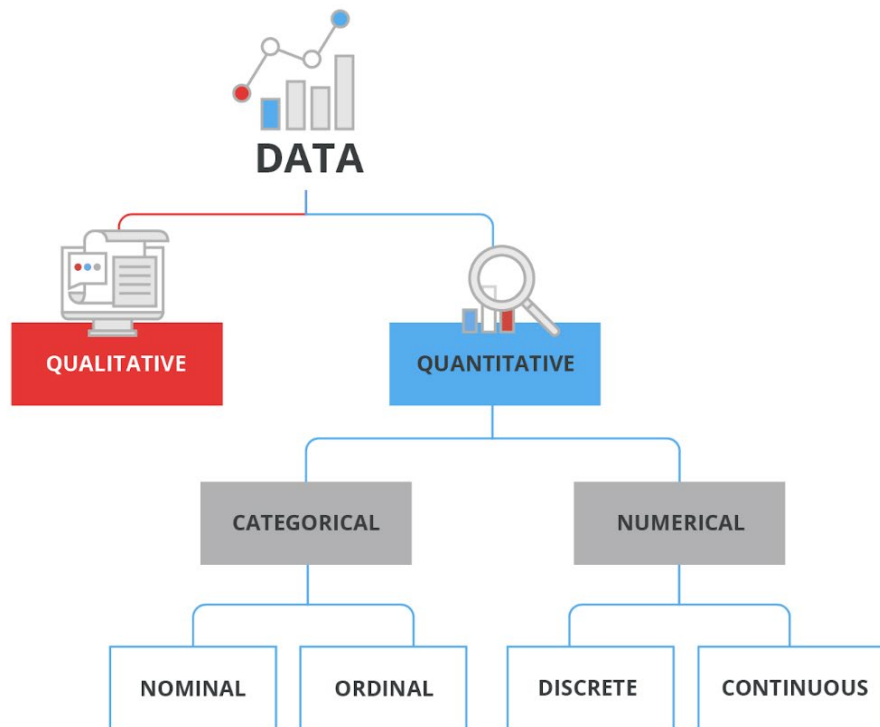
## 2. Different types of data

It is important to have a good understanding of the different types of data to apply machine learning successfully. From Unit 1, it should be clear that different machine learning techniques are only suitable when fitted onto specific types of data. Understanding the different types of data allows you to select the most suitable machine learning method (Donges, 2018).

Data can be divided into two main branches: qualitative and quantitative data. This course focuses on quantitative data and its subcategories: numerical and categorical data.

## 2.1 Classes

As seen in Figure 1, quantitative data can be divided into two subcategories: categorical and numerical data (Hale, 2018). Both of these subcategories are explored in the sections that follow.



**Figure 1:** Types of data.

### 2.1.1 Categorical data

Categorical data represents different categories, such as first language or age group. There are two types of categorical data, namely nominal data and ordinal data. Nominal data represents categorical units that are applied to label a value. The order of these units is not significant and therefore does not affect the meaning of the variable (Donges, 2018). Examples of nominal data include animal species or nationality, both of which are regarded as being equal (i.e. not ranked into levels of importance) (Hale, 2018). Ordinal data is categorical, ranked, and ordered in levels of importance. The order affects the meaning of such a variable (Donges, 2018), as indicated in Figure 2.

## NOMINAL CATEGORICAL DATA

**WHAT IS THE COLOUR OF YOUR EYES?**

---

☐ Blue

☐ Green

☐ Brown

☐ Other

**WHAT IS YOUR PREFERRED MODE OF TRANSPORT?**

---

☐ Train

☐ Bus

☐ Car

☐ Bicycle

## ORDINAL CATEGORICAL DATA

**WHAT IS YOUR HIGHEST LEVEL OF EDUCATION?**

---

☐ High school


☐ Bachelor's degree


☐ Master's degree


☐ Doctorate degree

**PLEASE RATE THE LEVEL OF CUSTOMER SERVICE YOU RECEIVED.**

---

☐  Good

☐  Neutral

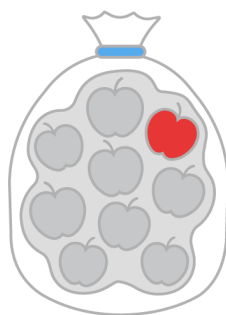
☐  Bad

**Figure 2:** Examples of categorical data.

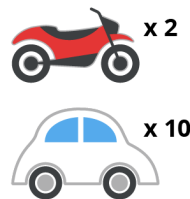
## 2.1.2 Numerical data

Numerical data consists of numbers and usually represents a measurement, such as house prices (Zhang, 2018.) Numerical data can further be divided into discrete or continuous data. Discrete data is distinct and separate in the sense that it can only be counted, not measured. This type of data can only be classified into specific classes, e.g. from one to infinity, but not divided into smaller parts. An example of discrete data is the number of tails achieved during 10 coin flips (Donges, 2018). There can never be 1.24 tails per 10 coin flips, but rather only a whole number (Zhang, 2018). In contrast to discrete data, continuous data can be measured and expressed in decimals, such as the size of a building. Figure 3 shows different examples of numerical data.

### DISCRETE NUMERICAL DATA

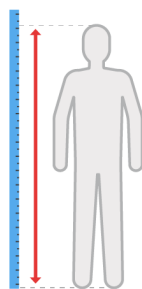


Number of apples  
in a 1kg bag



Number of vehicles  
versus motorcycles

### CONTINUOUS NUMERICAL DATA



Height



Weight

**Figure 3:** Examples of numerical data.

## 2.2 Structure

Data can either be structured or unstructured. Structured data has easily searchable patterns (Taylor, 2018), and can usually fit into fixed fields and specific columns, such as

a spreadsheet. These types of data are organised and can be easily processed by a machine learning model (Pickell, 2018).

Unstructured data does not have patterns that are easily searchable, such as video footage, audio files, and social media posts (Taylor, 2018). Unstructured data is difficult to deconstruct (Pickell, 2018).

**Table 1:** Comparison between structured and unstructured data. (Adapted from: Taylor, 2018)

	Structured data	Unstructured data
<b>Characteristics</b>	<ul style="list-style-type: none"> <li>• Easily searchable</li> <li>• Usually quantitative data</li> <li>• Pre-defined data</li> </ul>	<ul style="list-style-type: none"> <li>• Difficult to search</li> <li>• Usually qualitative data</li> <li>• No pre-defined data</li> </ul>
<b>Common applications</b>	<ul style="list-style-type: none"> <li>• Inventory-control systems</li> <li>• Airline ticketing</li> <li>• Customer relationship management (CRM) systems</li> <li>• Enterprise resource planning (ERP) systems</li> </ul>	<ul style="list-style-type: none"> <li>• Emails</li> <li>• Word processing</li> <li>• Presentation software</li> <li>• Media-viewing software</li> <li>• Media-editing software</li> </ul>
<b>Examples</b>	<ul style="list-style-type: none"> <li>• Personal identification number</li> <li>• Date</li> <li>• Telephone number</li> <li>• Credit card number</li> <li>• Customer name</li> <li>• Product name</li> <li>• Address</li> <li>• Product code</li> <li>• Transactional information</li> </ul>	<ul style="list-style-type: none"> <li>• Business report</li> <li>• Text file</li> <li>• Email</li> <li>• Audio file</li> <li>• Video file</li> <li>• Photo</li> <li>• Surveillance footage</li> <li>• Social media post</li> </ul>

**Pause and reflect:**

Consider your organisation's daily operations. What type of data does your organisation generate? Refer to what you learnt about matching data with appropriate machine learning techniques and explore what type of machine learning model you could fit onto the identified data within your organisation.

### 3. Data transformation

You have now explored the different types of data to determine the most appropriate machine learning technique to solve a given problem. The following interactive infographic illustrates the transformation of data. This infographic explores how different types of data are transformed from raw data points to visualisations, which can be used to inform business decisions.



**Interactive infographic 1: Transforming data.**



## 4. R programming language

R is a programming language that enables different applications of machine learning. Throughout this course, R is the programming language used in the integrated development environment (IDE) activities, including the Jupyter notebooks, embedded on the Online Campus. As a result, it is essential to have a basic understanding of R before continuing on this learning journey.

R is used for statistical computing and creating visualisations from data. R provides a wide variety of statistical techniques and offers an open source route to research in statistical methodology. One of the main strengths of R is that it enables the user to create high-quality plots from data with little effort (The R Foundation, n.d.).

### Explore further:

To explore the basics of R, navigate to the resource titled “An introduction to statistical learning” in this module’s downloads folder. Read Pages 42 to 51, which provides an introduction to R and covers the following topics:

- Basic commands
- Graphics
- Indexing data
- Loading data
- Additional graphical and numerical summaries

R is an integrated suite used to manipulate data to produce an output, and includes the following functionalities:

- **Data handling:** R is an effective data-handling tool, whereby data is gathered, recorded, and presented in a way that creates value for its users.
- **Data storage:** R stores the objects from a database into one file that can easily be imported and used to process the data to inform business decisions.
- **Calculations:** R consists of a suite of operators that can be used for calculations on arrays, or more specifically, matrices.
- **Integration:** R is an integrated suite of intermediate tools used for data analysis. It has a coherent system of tools and packages that can be used to analyse data in different ways.
- **Visualisation:** R includes graphical facilities used to visualise data, either on screen or in a hardcopy format.

- **Programming language:** R is an effective programming language that includes conditionals, loops, and user-defined recursive functions.

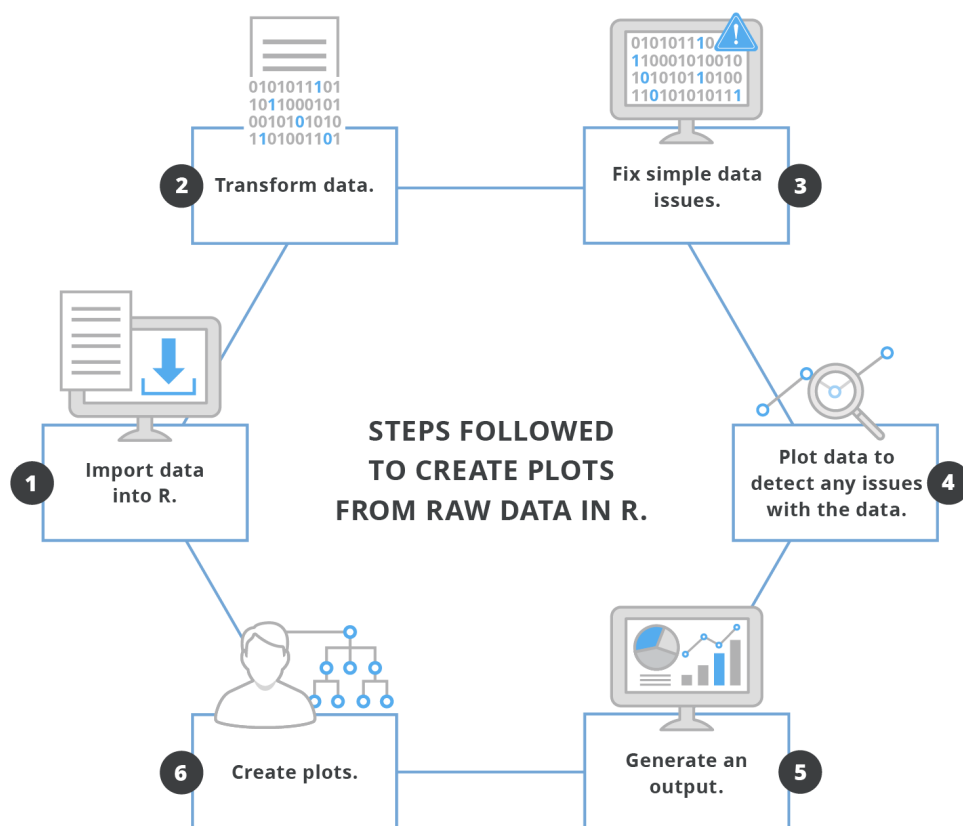
(The R Foundation, n.d.)

From the list of functionalities, it is clear that R is used to analyse data in a variety of ways, some of which will be discussed later in this course. Essentially, R is capable of analysing different data formats, including scalar, vector, and matrix formats. Scalar data is categorised as only containing a single value (Spector, 2002). A vector is a one-dimensional array that is a collection of one or more values of the same type (OverIQ, n.d.). A matrix is the most basic form of a collection of numbers that are arranged in a rectangular array; the matrix can be representative of images, a network, or an abstract structure (Sidhu, 2019).

The limitation of these data structures is that they should all be of the same data class (i.e. categorical, numerical, or a date). Trying to create a matrix that combines categorical and numerical data creates a structure that is only categorical in nature, as numeric values can be converted into characters, but characters cannot be converted into numeric values. To overcome this limitation, another data structure was created – the data frame. The `data.frame()` function in R creates a collection of variables with different properties, matrices, and lists (RDocumentation, n.d.a). This collection allows for a combination of numerical data, categorical data, and other data classes in one variable.

When viewing a data frame with multiple data classes, the numerical columns within the data frame can be part of either the “double” or “integer” classes. If the data forms part of the “double” class, it means that it creates a value with decimals (RDocumentation, n.d.b). This is in contrast to the “integer” class, which refers to a number without decimals. The categorical variables are part of the class “character”, indicating that these variables are either text or another character.

Although it is beneficial to understand the way a data frame is created, the practical application thereof falls outside the scope of this course. In most cases, it is not necessary to create new data; instead, you can use existing data sets. In this course, data sets will be provided when you are required to execute activities in R. To complete these activities, it is essential to understand the way in which data sets are imported into R.



**Figure 4:** Steps to create plots from raw data in R.

In the next section, each of the steps shown in Figure 4 is discussed in more detail.

## 4.1 Importing data and packages into R

When using existing data in R, the first step is to import the data. In most cases, the file that is imported will be a CSV file, a simple file format used to store data in tabular format. CSV refers to comma-separated values. As the name suggests, each cell in a row is separated by a comma. These files can be identified by the suffix “.csv” at the end of the file name. CSV files are similar to Microsoft Excel files and can be imported and exported to and from an Excel sheet. One of the key differences between an Excel sheet and a CSV file is that a CSV file only consists of one sheet, whereas Excel spreadsheets can contain multiple sheets in different tabs.

There are multiple ways to import CSV files into R:

- **`read.csv()` function:** This function is a base R function and is not very effective at analysing data as accurately as desired.

- **fread() function:** The `fread()` function, which is part of the `data.table` package, is used to import data from files directly into R and is much more effective at analysing data compared to the `read.csv()` function.
- **read\_csv() function:** The `read_csv()` function forms part of the `readr` package, which in turn forms part of the tidyverse universe of packages. The `read_csv()` function, like the `fread()` function, is more accurate at analysing data than the `read.csv()` function.

After the data is successfully imported into R, it is ready to be transformed appropriately, depending on the goal that should be achieved.

## 4.2 Transforming data

Once the data has been imported into R, it must be processed to the point where a model can be fitted onto the data to produce an output. All data sets imported into R are not suitable for this process immediately after they are loaded. Various R packages, that have to be imported, can process the data sets to make them suitable for model fitting. These packages are collections of functions and data sets that have been created to improve the functionality of R. These packages are stored in a directory, also referred to as a library. Some of the packages used throughout this course include the following:

- **caret:** caret is used for classification and regression training. It is a set of functions that aims to streamline the process of creating predictive models by splitting data, pre-processing data, selecting appropriate features, tuning and resampling, and estimating the importance of different variables.
- **CART:** This package is used to develop classification and regression trees (CART) to generate outputs.
- **data.table:** This is an extension of the `data.frame` package and is used to process large data sets to add, update, or remove columns.
- **dplyr:** Dplyr manipulates data by adding new variables, selecting variables and values, reducing multiple values, and shuffling rows.
- **glmnet:** The `glmnet` package is used to fit lasso or elastic-net regularisation when various forms of regression are applied.
- **Keras:** Keras is an application programming interface used when neural networks are trained. It enables fast experimentation within this field.
- **neuralnet:** This package is used to train neural networks. This package allows for flexible settings, depending on the data set.
- **TensorFlow:** TensorFlow is Google's open-source machine learning framework used for deep learning infrastructure.

- **Tidyverse:** This package is a collection of R packages used to prepare and visualise data.

This is, however, not an exhaustive list, and you will encounter more packages throughout your learning journey.

The terms used to describe these packages will become clear as they are applied in the respective modules. However, here is a practical example to illustrate how one of these packages would work. If the data set has a large array of variables, it may be necessary to reduce the number of variables to increase the model's accuracy. Selecting the most useful variables differs depending on the data set and the desired output.

To do this, the dplyr package is particularly useful. This package allows a user to select the most relevant variables from the data set in an easy-to-understand syntax. Not only can the number of variables from the imported data set be reduced, but rows that meet certain criteria can also be included, such as if only the male respondents' answers to a survey should be included. With a simple filter function in the package, the user can select multiple variables to be filtered on different criteria. So, instead of focusing on all male respondents, only male respondents between the ages of 30 and 40 years can be selected.

One of the greatest advantages of the dplyr package is that it allows the user to link multiple selections and filters at the same time. With this functionality, it is possible to select necessary columns or drop irrelevant columns, and filter the data by specific criteria, all in one flowing function.

The functionality of different packages will be illustrated in the practice IDE activity in this unit. After the data has been imported into R and the necessary variables have been selected, the data can be scanned for any issues that may affect the accuracy of the output.

## 4.3 Fixing simple data issues

The quality of the data can have a big impact on the accuracy of the output generated. Data cleaning is one of the most important aspects of data science, and by implication, machine learning (Sullivan, 2019). For this reason, it is essential that the data is analysed for any issues before it is used to generate the output. Different types of data require different levels of cleaning, depending on the issues that can be identified.

Some data cleaning issues include:

- **Unwanted observations:** The initial steps of cleaning data involve removing duplicate and irrelevant observations. Sometimes, an entry will appear twice in the same data set, but the duplicate must be removed. Irrelevant observations should also be removed if they do not contribute to solving the problem.
- **Structural errors:** Structural errors include typing errors, inconsistent text capitalisation, and mislabelled classes (i.e. two classes that should appear as one).
- **Unwanted outliers:** Outliers within the data set should be removed, but only for a legitimate reason. If an outlier is removed that should have been included to generate the output, the model prediction will be inaccurate.

- **Missing data:** Instances of missing data cannot be ignored in a data set. Missing data affects the accuracy of the output generated, because the model does not analyse the complete picture.

(Elite Data Science, n.d.)

As is evident in the previous list, two of the most common issues found in data sets are the presence of duplicate and missing values. All data sets must be analysed to ensure that duplicate and missing values are dealt with to ensure that an accurate output is generated. To do this in R, all values that could possibly be missing in the data set are converted into R's notation for missing data, NA. By acknowledging the missing values in the data set, the rows containing missing values can be omitted when generating the output.

The dplyr functions can also be applied to identify and remove missing values from a data set in R. To identify duplicate elements, the R function `duplicated()` can be used (Data Nova, n.d.). To only select unique data, the `unique()` function can be used. For example, to select the unique features from data frame `x`, the `unique(x)` function is used.

As stated previously, the functionality of different packages will be illustrated in the practice IDE activity later in this unit.

#### Explore further:

To learn more about cleaning data in R, consult this resource that sets out the process of [cleaning and preparing data for analysis](#).

As soon as the data has been cleaned, it is ready to be used to generate an output.

## 4.4 Plots

Depending on the type of variable and the intended output, different plots can be used to visualise the data. If, for example, the distribution of numerical values is visualised, a histogram may work best. For categorical values, a bar chart provides better insight into the different classes in the data set.

After exploring the data distribution, further steps can be taken to determine how variables relate to one another. In instances where there are two numerical variables, a scatter plot is useful to determine the relationship between the variables. A box plot is used to indicate whether a distribution is potentially skewed and whether there are any outliers present in a data set. A box plot can also be valuable when two or more data sets are compared with one another. Other useful graphs that can be generated in R include line charts, histograms, pie charts, dot charts, and miscs, to name a few (McCrown, n.d.).

Generating plots at this stage can be valuable if you are preparing the data and the machine learning model is still being optimised.

Explore the mechanics of [generating these plots in R](#) by engaging with the code and the resultant graphs.

**Explore further:**

Visit the R Graph Gallery to explore the [different graphs that can be generated in R](#).

## 4.5 Generating an output

After the data has been cleaned and plotted, the process of data analysis can begin. The problem and data in question can now be matched with an appropriate machine learning model to be fitted onto the data set.

There are a variety of data analyses that can be used, some of which are covered in more detail throughout this course, including linear regression, variable selection, shrinkage methods, logistic regression, generative models, tree-based methods, ensemble learning, neural networks, dimension reduction, and clustering.

Once an output is generated, it is valuable to use plots again to visualise the output. Visualisation transforms this data into digestible images that are easy to understand and ensures that the output can be used to solve the problem at hand.

## 4.6 Practical example

In Part 1 of Video 1, Professor Kostas Kalogeropoulos illustrates how data is imported and cleaned in R in preparation for machine learning applications.



**Video 1 Part 1:** Importing and cleaning data in R. (Access this set of notes on the Online Campus to engage with this video.)

In Part 2 of Video 1, Professor Kostas Kalogeropoulos demonstrates how data is visualised and transformed before being used to generate a variety of different plots in R.





**Video 1 Part 2:** Creating plots from data in R. (Access this set of notes on the Online Campus to engage with this video.)

## 5. Conclusion

Incredible amounts of data are generated every day, which creates new possibilities for businesses. However, in its raw form, data is not useful and should be cleaned and prepared before analysis.

Taking control of the data that is generated is an important step in using the data to an organisation's advantage. It is also important to determine what inferences should be made from the collected data. In doing this, it is essential to understand the type of data that is available, including the class, structure, and quality thereof. From there, you can choose the most suitable machine learning technique. To do this, refer to Unit 1, where the different types of machine learning techniques were introduced.

When the most accurate technique is chosen, the model can be fitted onto the data set to analyse the data and generate an output. To execute the processes of data analysis and applying machine learning to data, different programming languages, like R, can be used. Data can be analysed to drive decision-making by following the process explored in this set of notes.

Now that you have learned more about the different types of data and how input data is imported, transformed, and visualised in R, navigate to the practice IDE activity in the next component. Here, you will have the opportunity to practise loading data into R, cleaning the data, and creating different plots. You will then be tasked with replicating the process in the assessment IDE activity.



## 6. Bibliography

- Allaire, J.J. 2017. *Keras for R* [Blog, 5 September]. Available: <https://blog.rstudio.com/2017/09/05/keras-for-r/> [2019, December 5].
- Analytics Vidhya. 2019. *A beginner's guide to tidyverse – the most powerful collection of R packages for data science* [Blog, 13 May]. Available: <https://www.analyticsvidhya.com/blog/2019/05/beginner-guide-tidyverse-most-powerful-collection-r-packages-data-science/> [2019, December 5].
- Computer Hope. 2018. *How to create a CSV file*. Available: <https://www.computerhope.com/issues/ch001356.htm> [2019, November 19].
- DataMentor. n.d. *R plot function*. Available: <https://www.datamentor.io/r-programming/plot-function/> [2019, November 19].
- Data Novia. n.d. *Identify and remove duplicate data in R*. Available: <https://www.datanovia.com/en/lessons/identify-and-remove-duplicate-data-in-r/> [2019, November 19].
- Donges, N. 2018. *Data types in statistics*. Available: <https://towardsdatascience.com/data-types-in-statistics-347e152e8bee> [2019, November 15].
- Elite Data Science. n.d. *Data cleaning*. Available: <https://elitedatascience.com/data-cleaning> [2019, November 19].
- Friedman, J., Hastie, T., Tibshirani, R., Narasimhan, B., Simon, N. & Qian, J. 2019. *Glmnet: lasso and elastic-net regularized generalized linear models*. Available: <https://cran.r-project.org/web/packages/glmnet/index.html> [2020, January 10].
- Fritsch, S., Guenther, F., Wright, M.N., Suling, M. & Mueller, S.M. 2019. *Package 'neuralnet'*. Available: <https://cran.r-project.org/web/packages/neuralnet/neuralnet.pdf> [2020, January 10].
- Hale, J. 2018. *7 data types: a better way to think about data types for machine learning*. Available: <https://towardsdatascience.com/7-data-types-a-better-way-to-think-about-data-types-for-machine-learning-939fae99a689> [2019, November 15].
- Iconiq Inc. n.d. *R tutorial: data.table*. Available: <https://www.dezyre.com/data-science-in-r-programming-tutorial/r-data-table-tutorial> [2019, December 5].
- Kawaf, T. 2018. *TensorFlow for R* [Blog, 6 February]. Available: <https://blog.rstudio.com/2018/02/06/tensorflow-for-r/> [2019, December 5].
- Kuhn, M. 2019. *The caret package*. Available: <http://topepo.github.io/caret/index.html> [2019, December 5].
- McCrown, F. n.d. *Producing simple graphs with R*. Available: <https://sites.harding.edu/fmccrown/r/> [2020, January 30].

OverIQ. n.d. *One dimensional array in C*. Available: <https://overiq.com/c-programming-101/one-dimensional-array-in-c/> [2019, November 19].

Quick-R. n.d. *Tree-based models*. Available: <https://www.statmethods.net/advstats/cart.html> [2019, December 5].

RDocumentation. n.d.a. *Data.frame*. Available: <https://www.rdocumentation.org/packages/base/versions/3.6.1/topics/data.frame> [2019, November 19].

RDocumentation. n.d.b. *Double*. Available: <https://www.rdocumentation.org/packages/base/versions/3.6.1/topics/double> [2019, November 19].

Pickell, D. 2018. *Structured vs unstructured data – what's the difference?* Available: <https://learn.g2.com/structured-vs-unstructured-data> [2019, December 5].

Sidhu, R. 2019. *Beginner's introduction to matrices*. Available: <https://towardsdatascience.com/beginners-introduction-to-matrices-bd39289cc66a> [2019, November 19].

Spector, P. 2002. *Scalar data*. Available: <https://www.stat.berkeley.edu/~spector/extension/perl/notes/node21.html> [2019, November 19].

Sullivan, J. 2019. *Data cleaning with R and the tidyverse: detecting missing values*. Available: <https://towardsdatascience.com/data-cleaning-with-r-and-the-tidyverse-detecting-missing-values-ea23c519bc62> [2019, November 19].

Taylor, C. 2018. *Structured vs. unstructured data*. Available: <https://www.datamation.com/big-data/structured-vs-unstructured-data.html> [2019, November 15].

The R Foundation. n.d. *What is R?* Available: <https://www.r-project.org/about.html> [2019, November 18].

Tjønn, A.F. 2018. *When data becomes the new raw material for business, businesses should be valued based on their amount of data?* Available: <https://www.linkedin.com/pulse/when-data-becomes-new-raw-material-business-should-f%C3%B8llesdal-tj%C3%B8nn> [2019, November 19].

Wickham, H. n.d. *Dplyr*. Available: <https://www.rdocumentation.org/packages/dplyr/versions/0.7.8> [2019, December 5].

Zhang, A. 2018. *Data types from a machine learning perspective with examples*. Available: <https://towardsdatascience.com/data-types-from-a-machine-learning-perspective-with-examples-111ac679e8bc> [2019, November 15].