Datasets and handling data in R

Objective: The objective is to you acquainted with the exercise format of the course. Additionally, the aim is to familiarize yourself with the standard dataformat used in the course. Upon completing this exercise it is expected that you:

- Understand the format of the exercises and how the exercises are related to the reports.
- Can import data into R and represent the data in course $\mathbf{X}$, $\mathbf{y}$ format.
- Can do common preprocessing steps for datasets.
- Have selected a proper dataset for use in the your project work (for the reports).

R Help: R help for a function is obtained by typing `?<function name>` in the R command prompt. In practice, the fastest and easiest way to get help in R is often to simply Google your problem. For instance: `"How to add legends to a plot in R"` or the content of an error message. In the later case, it is often helpful to find the *simplest* script or input to script which will raise the error.

Piazza discussion forum: You can get help by asking questions on Piazza: piazza.com/dtu.dk/fall2019/october2019

1.1 How to do the exercises

During exercises in the course, you will go through an exercise document like this one. The exercises are centred around running and understanding a series of scripts provided in the 02450 Toolbox. The exercise descriptions guide you through the scripts and note that you won't have time to code everything from scratch. The scripts are also the basis for your work in the reports, where you will be able to re-use large parts of the code. However, for the reports, you will tailor the scripts to your dataset and problem.

The exercises are structured as smaller numbered sections. When a certain section concerns a particular script, it will be stated and their number will match. For instance, the first script you will run (in a little while) is called `ex1_5_1.R` and corresponds to the section 1.5.1 in this document.

1.2 Getting started with R

We assume that you have a working R IDE set up. If that is not the case, complete the pre-exercise (Exercise 0) before proceeding. If you have already done the optional Exercise 0, you can skip the next section ("Installing the 02450 Toolbox").

In the following, we will assume RStudio is installed and used to run commands and edit files. In addition to RStudio several additional packages need to be installed.

These can be installed from RStudio by going to `Tools -> Install packages..` and input the package name. The following packages need to be installed: `R.matlab, mltools, data.table, readxl, FNN, gplots, cvTools, neuralnet, randomForest, mclust, mixtools, sm, SnowballC, scatterplot3d, rgl, tm, sos, lsa, glmnet`.

1.3 Installing the 02450 Toolbox

The course will make use of several specialized scripts and toolboxes not included with R. These are distributed as a toolbox which need to be installed.

1.3.1 Download and unzip the 02450 Toolbox for R, `02450Toolbox_R.zip` . It will be assumed the toolbox is unpacked to create the directories:

```
<base-dir>/02450Toolbox_R/Tools/     # Misc. tools and packages
<base-dir>/02450Toolbox_R/Data/      # Datasets directory
<base-dir>/02450Toolbox_R/Scripts/   # Scripts for exercises
```

For the exercises, you should work on the example scripts in `<base-dir>/02450Toolbox_R/Scripts/` (notice the scripts are labelled according to exercise number) and not try to write the scripts from the bottom up.

1.3.2 To finalize the installation you need to update your path. To do this run the file `<base-dir>/02450Toolbox_R/setup.R` and ensure you do not get any errors.

1.4 Representation of data in R

We will use a standard data representation throughout the course. Using this representation makes it easy to apply the various tools in the 02450 Toolbox on a new dataset. Once you have a given dataset in the standard format, the scripts will all be set up to work with it correctly.

An overview of the format is presented for R in this table:

	R var.	Type	Size	Description
	`X`	Numeric	$N \times M$	Data matrix: The rows correspond to N data objects, each of which contains M attributes.
	`attributeNames`	Cell array	$M \times 1$	Attribute names: Name (string) for each of the M attributes.
	`N`	Numeric	Scalar	Number of data objects.
	`M`	Numeric	Scalar	Number of attributes.
Classification	`y`	Numeric	$N \times 1$	Class index: For each data object, `y` contains a class index, $\texttt{y}_n \in \{0, 1, \ldots, C-1\}$, where C is the total number of classes.
Classification	`classNames`	Cell array	$C \times 1$	Class names: Name (string) for each of the C classes.
Classification	`C`	Numeric	Scalar	Number of classes.

1.5 Loading data

Before we can begin to do machine learning, we need to load the data. Datasets are distributed as various types of files, and a few common ones will be shown here for future reference to be used once you have to load your own dataset.

Once we have loaded a dataset, we often need to process the data before the format fits our needs. For this course, in particular, this mostly means putting the dataset in the $\mathbf{X}$, $\mathbf{y}$-format shown above. The machine learning algorithms you will use needs the data to be in a numerical format, so we will also go through how to convert data which is in a text format into a numerical format.

Lastly, we will also go through a few tasks that often need to be handled before we can load some dataset.

For illustrating loading data, we will consider the Iris flower dataset, which we will also return to later on. The Iris flower dataset or Fisher's Iris dataset is a multivariate dataset introduced by Sir Ronald Aylmer Fisher (1936) for the problem of classifying Iris flower types. It is sometimes called Anderson's Iris dataset because Edgar Anderson collected the data to quantify the geographic variation of Iris flowers in the Gaspé Peninsula. The dataset consists of 50 samples from each of three species of Iris flowers (Iris setosa, Iris virginica and Iris versicolor). Four variables were measured from each sample, they are the length and the width of sepal and petal, in centimetres. Based on the combination of the four variables, Fisher developed a linear discriminant model to distinguish the species from each other. It is used as a typical test for many other classification techniques (see also `http://en.wikipedia.org/wiki/Iris_flower_data_set`). The data has been downloaded from `http://archive.ics.uci.edu/ml/datasets/Iris`.

The perhaps most common and simple format of storing data is the comma-separated values-file format (or CSV). In such files, the data is stored such that a sample or an observation is a line in a text document, and the document then has as many lines (or rows) as there are samples. The attribute values for an observation is written within one line, separated by (usually) a comma or a tab-character in a consistent order. This order is usually defined in a header (the first line of the file), which has a designation of the variable name in some format.

1.5.1 Inspect the file

`<base-dir>/02450Toolbox_R/Data/iris.csv`

using a simple text editor (e.g. for Windows "Notepad" or for MacOS "TextEdit"). Afterwards, inspect the script `ex1_5_1.R` to see how to load the Iris data from a CSV-file and put it into the standard format. Since the class label (the flower species) are stored as text (or strings), we convert them into a numerical value.

1.5.2 Sometimes datasets are distributed as Excel-files (`.xls(x)`). Inspect the script `ex1_5_2.R` to see how to load the same Iris data, when it has been stored as an Excel-file (open

`<base-dir>/02450Toolbox_R/Data/iris.xls`

to have a look at the file).

1.5.3 Other times, and especially in this course, data is stored as MATLAB files (`.mat`). Inspect `ex1_5_3.R` to load the Iris data from

`<base-dir>/02450Toolbox_R/Data/iris.mat` .

1.5.4 In the examples up until now, we have handled the data in the Iris dataset as if to solve a classification problem. We could say that the *primary* machine learning modelling aim is to classify the species of Iris flower based on the petal and sepal dimensions. However, we could also use the dataset to illustrate how to do regression without needing to use a whole different dataset. We would achieve this by e.g. trying to predict either of the petal (or sepal) dimensions based on the remaining dimensions, for instance. This changes how we define our $\mathbf{X}$,$\mathbf{y}$-format. Inspect `ex1_5_4.R` to see how to cast the Iris dataset into a regression problem. In the script, we will set up the $\mathbf{X}$,$\mathbf{y}$-format such that we are predicting the petal lengths form the other available information. Notice that we change how we use the information of the class label from before (the species information). Instead of storing it as a single variable, we have now used a "one-out-of-K encoding", since it is a categorical variable—we will return to various types of variables when we go through chapter 2 in the book (where one-out-of-K-encodings are described in section 2.4.1).

1.5.5 While the Iris dataset is a real dataset, it is a very clean and easy to work with dataset. Usually, data is a bit messier, and we will consider a toy dataset that has some common issues. Often, the description of "real-world" data is stored along with the data in some form of a text file. Have a look at the folder

`<base-dir>/02450Toolbox_R/Data/messy_data/`,

and read more about the toy dataset—notice that there is a `README.txt` file.

Inspect the data in `messy_data.data` and try to identify some issues (use e.g. simple text editor as before). Afterwards, inspect `ex1_5_5.R` to see how the dataset can be stored in the desired representation.

1.6 Select a dataset for the report

In the course you will work on a dataset of your own choosing for the report to be handed in shortly after the course. Please talk to the instructor regarding the dataset to be used for the report.

1.7 Tasks for the report

You are now able to address the following tasks for the report

1. **A description of your data set.**
   Explain

   - What the problem of interest is (i.e. what is your data about),
   - Where you obtained the data,
   - What has previously been done to the data. (i.e. if available go through some of the original source papers and read what they did to the data and summarize what were their results).
   - What the primary machine learning modeling aim is for the data, i.e. which attributes you feel are relevant when carrying out a classification, a regression, a clustering, an association mining, and an anomaly detection in the later reports and what you hope to accomplish using these techniques. For instance, which attribute do you wish to explain in the regression based on which other attributes? Which class label will you predict based on which other attributes in the classification task? If you need to transform the data to admit these tasks, explain roughly how you might do this (but don't transform the data now!).

References