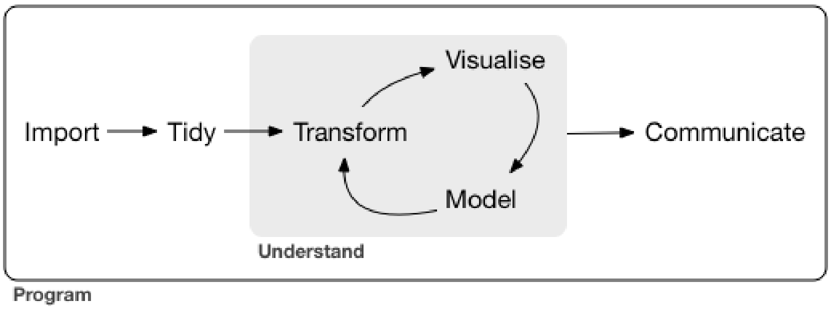
# Minilab 2b Worksheet

## Importing Data

A typical data science project looks something like the following.



Typically you will *import* (load) datasets from some external source rather than typing out the data into your R script. Useful functions for doing this come from the *readr* package in the tidyverse which are described in Chapter 11 Data Import of:

> Wickham, H. and Grolemund, G. (2017) *R for Data Science*, O'Reilly.

### 1. Importing data from a CSV file

For tiny datasets, we can just type (or copy-and-paste) the data directly into our R script. For larger datasets, we can import data from a text file, an Excel spreadsheet, a database or from many other sources. A CSV file is just a text file that follows a particular format.

(1) Open Microsoft Excel and enter a simple dataset involving one column with text (strings) and one column with numbers. Save your spreadsheet in "CSV (Comma delimited)" format.

(2) Open File Explorer, find your saved CSV file, click on it with the right mouse button and use "Open with ..." to open your file in a text editor (e.g. Notepad).
*Why do you think it is called a CSV format?*

(3) Suppose you have your data in a CSV file called data01.csv. In the Files pane (bottom right quadrant of the RStudio IDE), navigate your way to the folder that contains data01.csv. In the Files pane, use the More menu to select "Set As Working Directory". Now you can try the following R code.

```
library(tidyverse)
mydata = read_csv("data01.csv")
mydata
```

Check that the data in your CSV file has been successfully imported into R as a tibble.

*Warning!* — The function read_csv() is part of the readr package in the tidyverse. There is another (similar) function provided by R called read.csv(). Please stick with the tidyverse version read_csv() with an underscore as it provides a tibble and it is claimed to be faster for large CSV files.

*Exercise*. Go to the Kaggle website (https://www.kaggle.com/datasets) and search for a dataset on some topic that interests you.

- Download that dataset and open it in Excel. What structure does it have?
- See if you can import the dataset into RStudio. If the data is not in CSV format, have a look at https://readr.tidyverse.org/ (and especially the "Cheatsheet").
- See if you can make a boxplot using some of this data. What do you observe?

## 2. Importing data from a URL

Have a brief look at the article "The Most Common Unisex Names In America: Is Yours One Of Them?". The article has a link (just under the author's name) to the data it uses on GitHub.

(1) The link takes you to https://github.com/fivethirtyeight/data/tree/master/unisex-names. From that GitHub page, click on unisex_names_table.csv and you will be shown a nicely formatted table.

(2) Click on the "Raw" button (above and to the right of the nicely formatted table) and you will see a CSV file as raw text in your browser. The URL of this webpage can then be used in the code below to import this dataset directly from the web.

```
library(tidyverse)
url = "https://raw.githubusercontent.com/fivethirtyeight/data/master/unisex-names/unisex_names_table.csv"
namesdata = read_csv(url)
namesdata
```

*Warning!* — You might have trouble with copy-and-paste from the R code above. Check that the minus sign (-) in the url string appears correctly in your code.

(3) We can then use the arrange() function in dplyr to see which names have the male and female share almost the same.

```
View(arrange(namesdata,gap))
```

So the top two are Bless and Camdyn. *What are the next few? Do your results match the table in the original article?*

*Exercise.* Try importing another CSV dataset from the fivethiryeight GitHub https://github.com/fivethirtyeight/data/ pages.

## 3. Importing data from other file types

Other similar file formats are easy to import into R. For more information, you can read https://readr.tidyverse.org/ and https://r4ds.had.co.nz/data-import.html.

The RStudio *Data Import Cheatsheet* (from https://rstudio.com/resources/cheatsheets/) provides a useful summary of tidyverse functions for reading data into R.

*Exercise.*

(a) Take a CSV file and replace all the commas with semicolons and resave. Then use read_csv2() to load the data from the file. Check that the dataset loads correctly using View().

(b) Replace all the semicolons with tabs and save as a ".tsv" file. Then use read_tsv() to load the data from the .tsv file. Check that the dataset loads correctly using View().

(c) Create a small table in Excel with column names and save as an ".xlsx" file. Install the *readxl* package in R. Then use library(readxl) and read_excel() to load the data from the .xlsx file. Check that the dataset loads correctly using View(). For more information about reading from Excel files, see https://readxl.tidyverse.org/.

## Summary

In this minilab, we have looked at importing data from a CSV file, either in the local filesystem or stored on the web. We have also briefly looked at reading data from other file formats. It is also possible to read and write data directly from a database.