Introduction to R: Writing and Reading files

Reading data into your R environment is the first step in conducting data analysis. Data comes in many different forms and although R is equipped to deal with most data formats, this lesson will focus on reading common data formats like comma separated values files (CSV) and Microsoft Excel files.

R Working Directory and File Paths

Before we can jump in and starting loading data, we need to learn a little bit about R's working directory and file paths. When you run R, it starts in a default location in your computer's file system called the working directory. You can check your working directory with the getwd() function:

```
getwd()                                  # Get the current working directory
```

```
## [1] "C:/Users/elias/Dropbox/Problem Solving Environments for Data Science 2018/R introduction"
```

The working directory acts as your starting point for accessing other files on your computer. To load data into R from your hard disk, you either need to put the data file in your working directory, change your working directory to the folder containing the data or supply the data's file path to the data reading function. You can change your working directory by supplying a new file path in quotes to the setwd() function:

```
setwd("C:/Users/elias/Documents")          # Set a new working directory

getwd()                                   # Check the working directory again
```

```
## [1] "C:/Users/elias/Documents"
```

*Note: you can use forward slashes for your file path even in Windows which normally uses backslashes. If you want to use backslashes for file paths in Windows you should use double backslashes (\\) Instead of worrying about slashes in filepaths, you can have R construct file paths for you using the file.path() function. It takes a comma separated sequence of character strings and then uses them to construct a file path string for you:

```
my_path <- file.path("C:","Users","elias","","Documents")  # Construct path

print (my_path )                # Check the path
```

```
## [1] "C:/Users/elias//Documents"
```

```
setwd(my_path)                  # Set the working directory to the path

getwd()                         # Check the working directory again
```

```
## [1] "C:/Users/elias/Documents"
```

In RStudio you can also change the working directory under the "Session" dropdown menu. Under session select "Set working directory", "Choose Directory", navigate to the folder you want to set as your working directory and click "Select folder."

You can list the files and folders in the current working directory using the list.files() function:

```
list.files()       # A list of files and folders in my Documents
```

```
##  [1] "control flow.nb.html"
##  [2] "control flow.Rmd"
##  [3] "control_flow.pdf"
##  [4] "datasets.xlsx"
```

```
##  [5] "frames.nb.html"
##  [6] "frames.Rmd"
##  [7] "functions.nb.html"
##  [8] "functions.pdf"
##  [9] "functions.Rmd"
## [10] "initial data exploration and preparation.nb.html"
## [11] "initial data exploration and preparation.Rmd"
## [12] "Intr_R_vectors_matrices_lists_functions_Reading_wri_files"
## [13] "Intr_R_vectors_matrices_lists_functions_Writing_Reading_files"
## [14] "Introduction to R - frequency tables.ipynb"
## [15] "Introduction to R - Merging Data.ipynb"
## [16] "Lists.nb.html"
## [17] "Lists.pdf"
## [18] "Lists.Rmd"
## [19] "matrices.nb.html"
## [20] "matrices.pdf"
## [21] "matrices.Rmd"
## [22] "preparing dates.Rmd"
## [23] "preparing numerical data.nb.html"
## [24] "preparing numerical data.Rmd"
## [25] "readxl_documentation.pdf"
## [26] "vectors.nb.html"
## [27] "vectors.pdf"
## [28] "vectors.Rmd"
## [29] "working with character data.nb.html"
## [30] "working with character data.Rmd"
## [31] "Writing_and_Reading_files.nb.html"
## [32] "Writing_and_Reading_files.Rmd"
```

Read CSV and TSV Files

Data is commonly stored in simple text files consisting of values delimited (separated) by a special character. For instance, CSV files use commas as the delimiter and tab separated value files (TSV) use tabs as the delimiter. You can use the read.csv() function to read CSV files into R:

```
draft <- read.csv(file ='C:\\Users\\elias\\Downloads\\risk_factors_cervical_cancer.csv',     # Path to
                  stringsAsFactors = FALSE)   # Encode characters as factors?

print(head(draft,15))
```

```
##    Age Number.of.sexual.partners First.sexual.intercourse
## 1   18                        4.0                    15.0
## 2   15                        1.0                    14.0
## 3   34                        1.0                       ?
## 4   52                        5.0                    16.0
## 5   46                        3.0                    21.0
## 6   42                        3.0                    23.0
## 7   51                        3.0                    17.0
## 8   26                        1.0                    26.0
## 9   45                        1.0                    20.0
## 10  44                        3.0                    15.0
## 11  44                        3.0                    26.0
## 12  27                        1.0                    17.0
```

```
## 13  45                     4.0                   14.0
## 14  44                     2.0                   25.0
## 15  43                     2.0                   18.0
##    Num.of.pregnancies Smokes Smokes..years. Smokes..packs.year.
## 1                 1.0    0.0           0.0                 0.0
## 2                 1.0    0.0           0.0                 0.0
## 3                 1.0    0.0           0.0                 0.0
## 4                 4.0    1.0          37.0                37.0
## 5                 4.0    0.0           0.0                 0.0
## 6                 2.0    0.0           0.0                 0.0
## 7                 6.0    1.0          34.0                 3.4
## 8                 3.0    0.0           0.0                 0.0
## 9                 5.0    0.0           0.0                 0.0
## 10                  ?    1.0   1.266972909                 2.8
## 11                4.0    0.0           0.0                 0.0
## 12                3.0    0.0           0.0                 0.0
## 13                6.0    0.0           0.0                 0.0
## 14                2.0    0.0           0.0                 0.0
## 15                5.0    0.0           0.0                 0.0
##    Hormonal.Contraceptives Hormonal.Contraceptives..years. IUD IUD..years.
## 1                      0.0                             0.0 0.0         0.0
## 2                      0.0                             0.0 0.0         0.0
## 3                      0.0                             0.0 0.0         0.0
## 4                      1.0                             3.0 0.0         0.0
## 5                      1.0                            15.0 0.0         0.0
## 6                      0.0                             0.0 0.0         0.0
## 7                      0.0                             0.0 1.0         7.0
## 8                      1.0                             2.0 1.0         7.0
## 9                      0.0                             0.0 0.0         0.0
## 10                     0.0                             0.0   ?           ?
## 11                     1.0                             2.0 0.0         0.0
## 12                     1.0                             8.0 0.0         0.0
## 13                     1.0                            10.0 1.0         5.0
## 14                     1.0                             5.0 0.0         0.0
## 15                     0.0                             0.0 1.0         8.0
##    STDs STDs..number. STDs.condylomatosis STDs.cervical.condylomatosis
## 1   0.0           0.0                 0.0                          0.0
## 2   0.0           0.0                 0.0                          0.0
## 3   0.0           0.0                 0.0                          0.0
## 4   0.0           0.0                 0.0                          0.0
## 5   0.0           0.0                 0.0                          0.0
## 6   0.0           0.0                 0.0                          0.0
## 7   0.0           0.0                 0.0                          0.0
## 8   0.0           0.0                 0.0                          0.0
## 9   0.0           0.0                 0.0                          0.0
## 10  0.0           0.0                 0.0                          0.0
## 11  0.0           0.0                 0.0                          0.0
## 12  0.0           0.0                 0.0                          0.0
## 13  0.0           0.0                 0.0                          0.0
## 14  0.0           0.0                 0.0                          0.0
## 15  0.0           0.0                 0.0                          0.0
##    STDs.vaginal.condylomatosis STDs.vulvo.perineal.condylomatosis
## 1                          0.0                                0.0
## 2                          0.0                                0.0
```

```
## 3                          0.0                                0.0
## 4                          0.0                                0.0
## 5                          0.0                                0.0
## 6                          0.0                                0.0
## 7                          0.0                                0.0
## 8                          0.0                                0.0
## 9                          0.0                                0.0
## 10                         0.0                                0.0
## 11                         0.0                                0.0
## 12                         0.0                                0.0
## 13                         0.0                                0.0
## 14                         0.0                                0.0
## 15                         0.0                                0.0
##    STDs.syphilis STDs.pelvic.inflammatory.disease STDs.genital.herpes
## 1            0.0                              0.0                 0.0
## 2            0.0                              0.0                 0.0
## 3            0.0                              0.0                 0.0
## 4            0.0                              0.0                 0.0
## 5            0.0                              0.0                 0.0
## 6            0.0                              0.0                 0.0
## 7            0.0                              0.0                 0.0
## 8            0.0                              0.0                 0.0
## 9            0.0                              0.0                 0.0
## 10           0.0                              0.0                 0.0
## 11           0.0                              0.0                 0.0
## 12           0.0                              0.0                 0.0
## 13           0.0                              0.0                 0.0
## 14           0.0                              0.0                 0.0
## 15           0.0                              0.0                 0.0
##    STDs.molluscum.contagiosum STDs.AIDS STDs.HIV STDs.Hepatitis.B STDs.HPV
## 1                         0.0       0.0      0.0              0.0      0.0
## 2                         0.0       0.0      0.0              0.0      0.0
## 3                         0.0       0.0      0.0              0.0      0.0
## 4                         0.0       0.0      0.0              0.0      0.0
## 5                         0.0       0.0      0.0              0.0      0.0
## 6                         0.0       0.0      0.0              0.0      0.0
## 7                         0.0       0.0      0.0              0.0      0.0
## 8                         0.0       0.0      0.0              0.0      0.0
## 9                         0.0       0.0      0.0              0.0      0.0
## 10                        0.0       0.0      0.0              0.0      0.0
## 11                        0.0       0.0      0.0              0.0      0.0
## 12                        0.0       0.0      0.0              0.0      0.0
## 13                        0.0       0.0      0.0              0.0      0.0
## 14                        0.0       0.0      0.0              0.0      0.0
## 15                        0.0       0.0      0.0              0.0      0.0
##    STDs..Number.of.diagnosis STDs..Time.since.first.diagnosis
## 1                          0                               ?
## 2                          0                               ?
## 3                          0                               ?
## 4                          0                               ?
## 5                          0                               ?
## 6                          0                               ?
## 7                          0                               ?
## 8                          0                               ?
```

```
## 9                           0                                ?
## 10                          0                                ?
## 11                          0                                ?
## 12                          0                                ?
## 13                          0                                ?
## 14                          0                                ?
## 15                          0                                ?
##    STDs..Time.since.last.diagnosis Dx.Cancer Dx.CIN Dx.HPV Dx Hinselmann
## 1                                ?         0      0      0  0          0
## 2                                ?         0      0      0  0          0
## 3                                ?         0      0      0  0          0
## 4                                ?         1      0      1  0          0
## 5                                ?         0      0      0  0          0
## 6                                ?         0      0      0  0          0
## 7                                ?         0      0      0  0          1
## 8                                ?         0      0      0  0          0
## 9                                ?         1      0      1  1          0
## 10                               ?         0      0      0  0          0
## 11                               ?         0      0      0  0          0
## 12                               ?         0      0      0  0          0
## 13                               ?         0      0      0  0          0
## 14                               ?         0      0      0  0          0
## 15                               ?         0      0      0  0          0
##    Schiller Citology Biopsy
## 1         0        0      0
## 2         0        0      0
## 3         0        0      0
## 4         0        0      0
## 5         0        0      0
## 6         0        0      0
## 7         1        0      1
## 8         0        0      0
## 9         0        0      0
## 10        0        0      0
## 11        0        0      0
## 12        0        0      0
## 13        0        0      0
## 14        0        0      0
## 15        0        0      0
```

Data loaded into R via read.csv() becomes data frame. To load tab separated values, include the sep argument and set it to the tab character "$\widehat{\ }$":

```
draft2 <- read.csv(file="C:\\Users\\elias\\Downloads\\risk_factors_cervical_cancer.csv",   # Path to th
          sep = "\t",                       # Use tabs as the delimiting character
          stringsAsFactors = FALSE)

print(head(draft2,15))
```

```
##    Age.Number.of.sexual.partners.First.sexual.intercourse.Num.of.pregnancies.Smokes.Smokes..years..Sm
## 1
## 2
## 3
## 4
## 5
```

```
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
```

The read.csv() function is an extension of a more general data reading function called read.table(). read.csv() just sets a few arguments of read.table() to values suitable for reading CSV and TSV files. The read.table() function has numerous additional arguments that have various effects on reading data; there are too many arguments to cover them all in detail here but you can always get more information by checking the function documents with ?read.table or help(read.table).

Read Excel Files

Microsoft Excel is a ubiquitous enterprise spreadsheet program that stores data in its own format with the extension .xls or .xlsx. One simple way to read Excel data into R is to open an Excel workbook using Excel, save the data in CSV format or as a tab-delimited text file and then use the read.csv() function to load the data into R.

The readxl package makes it easy to get data out of Excel and into R. Compared to many of the existing packages (e.g. gdata, xlsx, xlsReadWrite) readxl has no external dependencies, so it's easy to install and use on all operating systems. It is designed to work with tabular data.

readxl supports both the legacy .xls format and the modern xml-based .xlsx format. The libxls C library is used to support .xls, which abstracts away many of the complexities of the underlying binary format. To parse .xlsx, we use the RapidXML C++ library.

Installation

The easiest way to install the latest released version from CRAN is to install the whole tidyverse.

install.packages("tidyverse")

NOTE: you will still need to load readxl explicitly, because it is not a core tidyverse package loaded via library(tidyverse).

Alternatively, install just readxl from CRAN:

install.packages("readxl")

Or install the development version from GitHub:

install.packages("devtools") devtools::install_github("tidyverse/readxl")

Usage

library(readxl)

readxl includes several example files, which we use throughout the documentation. Use the helper readxl_example() with no arguments to list them or call it with an example filename to get the path.

readxl_example()

```
#>  [1] "clippy.xls"    "clippy.xlsx"   "datasets.xls"  "datasets.xlsx"
#>  [5] "deaths.xls"    "deaths.xlsx"   "geometry.xls"  "geometry.xlsx"
#>  [9] "type-me.xls"   "type-me.xlsx"
readxl_example("clippy.xls")
#> [1] "/Users/jenny/resources/R/library/readxl/extdata/clippy.xls"
```

read_excel() reads both xls and xlsx files and detects the format from the extension.

```
xlsx_example <- readxl_example("datasets.xlsx")
read_excel(xlsx_example)
#> # A tibble: 150 x 5
#>   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
#>          <dbl>       <dbl>        <dbl>       <dbl> <chr>
#> 1         5.10        3.50         1.40       0.200 setosa
#> 2         4.90        3.00         1.40       0.200 setosa
#> 3         4.70        3.20         1.30       0.200 setosa
#> # ... with 147 more rows

xls_example <- readxl_example("datasets.xls")
read_excel(xls_example)
#> # A tibble: 150 x 5
#>   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
#>          <dbl>       <dbl>        <dbl>       <dbl> <chr>
#> 1         5.10        3.50         1.40       0.200 setosa
#> 2         4.90        3.00         1.40       0.200 setosa
#> 3         4.70        3.20         1.30       0.200 setosa
#> # ... with 147 more rows
```

List the sheet names with excel_sheets().

```
excel_sheets(xlsx_example)
#> [1] "iris"     "mtcars"   "chickwts" "quakes"
```

Specify a worksheet by name or number.

```
read_excel(xlsx_example, sheet = "chickwts")
#> # A tibble: 71 x 2
#>   weight feed
#>    <dbl> <chr>
#> 1   179. horsebean
#> 2   160. horsebean
#> 3   136. horsebean
#> # ... with 68 more rows
read_excel(xls_example, sheet = 4)
#> # A tibble: 1,000 x 5
#>     lat  long depth   mag stations
#>   <dbl> <dbl> <dbl> <dbl>    <dbl>
#> 1 -20.4  182.  562.  4.80      41.
#> 2 -20.6  181.  650.  4.20      15.
#> 3 -26.0  184.   42.  5.40      43.
```

```
#> # ... with 997 more rows
```

There are various ways to control which cells are read. You can even specify the sheet here, if providi

```
read_excel(xlsx_example, n_max = 3)
#> # A tibble: 3 x 5
#>   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
#>          <dbl>       <dbl>        <dbl>       <dbl> <chr>
#> 1         5.10        3.50         1.40       0.200 setosa
#> 2         4.90        3.00         1.40       0.200 setosa
#> 3         4.70        3.20         1.30       0.200 setosa
read_excel(xlsx_example, range = "C1:E4")
#> # A tibble: 3 x 3
#>   Petal.Length Petal.Width Species
#>          <dbl>       <dbl> <chr>
#> 1         1.40       0.200 setosa
#> 2         1.40       0.200 setosa
#> 3         1.30       0.200 setosa
read_excel(xlsx_example, range = cell_rows(1:4))
#> # A tibble: 3 x 5
#>   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
#>          <dbl>       <dbl>        <dbl>       <dbl> <chr>
#> 1         5.10        3.50         1.40       0.200 setosa
#> 2         4.90        3.00         1.40       0.200 setosa
#> 3         4.70        3.20         1.30       0.200 setosa
read_excel(xlsx_example, range = cell_cols("B:D"))
#> # A tibble: 150 x 3
#>   Sepal.Width Petal.Length Petal.Width
#>         <dbl>        <dbl>       <dbl>
#> 1        3.50         1.40       0.200
#> 2        3.00         1.40       0.200
#> 3        3.20         1.30       0.200
#> # ... with 147 more rows
read_excel(xlsx_example, range = "mtcars!B1:D5")
#> # A tibble: 4 x 3
#>     cyl  disp    hp
#>   <dbl> <dbl> <dbl>
#> 1    6.  160.  110.
#> 2    6.  160.  110.
#> 3    4.  108.   93.
#> # ... with 1 more row
```

If NAs are represented by something other than blank cells, set the na argument.

```
read_excel(xlsx_example, na = "setosa")
#> # A tibble: 150 x 5
#>   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
#>          <dbl>       <dbl>        <dbl>       <dbl> <chr>
#> 1         5.10        3.50         1.40       0.200 <NA>
#> 2         4.90        3.00         1.40       0.200 <NA>
#> 3         4.70        3.20         1.30       0.200 <NA>
#> # ... with 147 more rows
```

If you are new to the tidyverse conventions for data import, you may want to consult the data import

chapter in R for Data Science. readxl will become increasingly consistent with other packages, such as readr. Reference:https://readxl.tidyverse.org/

```
library(readxl)
draft3 <- readxl_example("datasets.xlsx")  # Reads the first worksheet in the file
read_excel(draft3)
```

```
## # A tibble: 150 x 5
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           <dbl>       <dbl>        <dbl>       <dbl> <chr>
##  1         5.10        3.50         1.40       0.200 setosa
##  2         4.90        3.00         1.40       0.200 setosa
##  3         4.70        3.20         1.30       0.200 setosa
##  4         4.60        3.10         1.50       0.200 setosa
##  5         5.00        3.60         1.40       0.200 setosa
##  6         5.40        3.90         1.70       0.400 setosa
##  7         4.60        3.40         1.40       0.300 setosa
##  8         5.00        3.40         1.50       0.200 setosa
##  9         4.40        2.90         1.40       0.200 setosa
## 10         4.90        3.10         1.50       0.100 setosa
## # ... with 140 more rows
```

```
head(draft3)
```

```
## [1] "C:/Users/elias/Documents/R/win-library/3.4/readxl/extdata/datasets.xlsx"
```

Reading Web Data

The Internet gives you access to more data than you could ever hope to analyze. Data analysis often begins with getting data from the web and loading it into R. Websites that offer data for download usually let you download data as CSV, TSV or excel files. The easiest way to use web data in R, is to simply download data to your hard drive in CSV, TSV or an excel file format and then use the functions we discussed earlier to load the data into R. You can supply a url to read.csv() or read.table() to read data directly from the web, but doing so can be problematic since web data isn't always formatted nicely. It can be helpful to do a little data cleaning, such as deleting unnecessary titles, images or other oddities in excel or a text editor to prepare data for use in R. In addition, large data sets often come in compressed formats like .zip and need to be decompressed before loading them into R so they aren't always easy loaded directly from the web. Reading from the clipboard is another option for reading web data and other tabular data. To read in data from the clipboard, highlight the data you want to copy and use the appropriate copy function as if you were going to copy and paste the data. Next, use the read.csv() or read.table() function with the the first argument set to "clipboard":

```
# Go to http://www.basketball-reference.com/leagues/NBA_2017_totals.html
# click the CSV button to format data and then copy some data to the clipboard

BB_reference_data <- read.csv("clipboard")       # Read data from the clipboard
```

```
## Warning in read.table(file = file, header = header, sep = sep, quote =
## quote, : incomplete final line found by readTableHeader on 'clipboard'
```

```
print ( head(BB_reference_data, 10) )            # Check the data
```

```
## [1] http...hamelg.blogspot.com
## <0 rows> (or 0-length row.names)
```

Data comes in all sorts of formats other than the friendly ones we've discussed thus far. R has functions and

packages for working with data in other common data formats like SAS, SPSS and Stata files, json, xml, html and databases. We won't cover how to deal with all the different data sources you might encounter in this lesson, but rest assured that there is probably a way to work with your data in R if you do some digging. If you encounter a data source you don't know how to work with, a little bit of Googling will usually reveal how to convert it into a more familiar format or use an R package to deal with it directly.

Writing Data To CSV

In the course of cleaning data, data analysis and predictive modeling, you'll generate new data. You can write data in an R data frame to CSV using the write.csv() function:

```
write.csv(BB_reference_data,      # Name of variable assigned to the data
          "BB_data.csv",          # Name of the file to create to store the data
           row.names = FALSE,)    # Whether to include row names in the file
```

Data is written to your current working directory. It's a good idea to save data after long, computationally expensive operations so that you don't lose progress or results. Now that we know the basics of reading and writing data, we are almost ready to start exploring data, but before diving in we will spend a couple lessons learning basic R programming constructs. Reference:http://hamelg.blogspot.com