# Reading Source Data in R

R is useful for analyzing data, once there is data to be analyzed. Real datasets come from numerous places in various formats, many of which actually can be leveraged by the savvy R programmer with the right set of packages and practices.

> **** Note: It is possible to stream data in R, but it's probably outside of the scope of most educational uses. We will only be worried about static data. ****

## Understanding the flow of data

How many times have you accessed some information in a browser or software package and wished you had that data as a spreadsheet or csv instead? R, like most programming languages, can serve as the bridge between the information you have and the information you want. A significant amount of programming is an exercise in changing file formats.


To use R for data manipulation and analysis, source data must be read in, analyzed or manipulated as necessary, and the results are output in some meaningful format. This document focuses on the red arrow: How is source data read into R? The ability to access data is fundamental to building useful, productive systems, and is sometimes the biggest challenge

Immediately preceding the scope of this document is the source data itself. It is left to the reader to understand the source data: How and where it is stored, in what file formats, file ownership and permissions, etc. Immediately beyond the scope of this document is actual programming. It is left to the reader to determine which tools and methods are best applied to the source data to yield the desired results.

## Common Data Sourcing Methods and How to Access them

### Local Storage and File Connections

The easiest way to store and access files in R is from local storage. Common methods for reading in data (based on the built-in base and utils libraries) are usually happy to take a file path as an argument, including `source()` and `read.csv()`. The file path can be relative or absolute and can include mapped network storage.

### Terminal

The `readline()` function reads input data from the terminal as the user types.

### HTTP and FTP

Internet protocols are used to connect to certain types of remote servers (FTP servers in particular) and REST APIs. Internet protocols can be handled using either the RCurl or httr packages depending on your specific needs. RCurl offers the most control and flexibility over your internet connection, but it is also more difficult to use: it works well for retrieving and scraping websites (even if you have to spoof the browser to do so). It is also preferable if you want the most granular level of control possible. httr is newer and more straightforward: it works well for FTP transfers and calling REST APIs. Note that you may have to read in data in a binary format and convert it into the file type that most makes sense for your needs.

### Databases

You can access databases using the RODBC package, the RODBCDBI package, or RStudio's odbc package. The odbc package tends to be the fastest according to RStudio's benchmark data. All of these packages are available on CRAN.

Using database connections allows you to read from SQL based databases (MySQL, SQL Server, Oracle, etc.) as well as Microsoft Access databases.

## Common Data Formats and How to Read them

Below is a list of common data formats and a well-documented method for reading that data. This list is not meant to be comprehensive. Packages are specified where necessary and are all available from CRAN.

- CSV
  - read.csv()
- TXT
  - read.table() for tabular data
  - source() for source files
  - open(), open.connection(), open.srcfile(), etc. for general file connections
- XLSX
  - read.xlsx() or readworkbook() from openxlsx package
- ZIP
  - unzip() to unzip the folder, then use the appropriate read method for the content
- JSON
  - fromJSON() or read_json() from jsonlite package
- XML
  - read_xml() from xml2 package
- HTML
  - read_html() from xml2 package
- PDF
  - See pdftools package
- Google Sheets
  - gs_read() from googlesheets package
- Images
  - image_read() from magick package

## Links, Sources, and Further Reading

- R Data Import/Export
  https://cran.r-project.org/doc/manuals/r-release/R-data.html
- Packages
  - googledrive
    https://cran.r-project.org/package=googledrive
  - googlesheets
    https://cran.r-project.org/package=googlesheets
  - httr
    https://cran.r-project.org/package=httr
  - jsonlite
    https://cran.r-project.org/package=jsonlite
  - magick
    https://cran.r-project.org/package=magick
  - odbc
    https://cran.r-project.org/package=odbc
  - openxlsx
    https://cran.r-project.org/package=openxlsx
  - pdftools
    https://cran.r-project.org/package=pdftools
  - RCurl
    https://cran.r-project.org/package=RCurl
  - RODBC
    https://cran.r-project.org/package=RODBC
  - RODBCDBI
    https://cran.r-project.org/package=RODBCDBI
  - xml2
    https://cran.r-project.org/package=xml2