Entering / cleaning data 1

Reading in flat files

Reading in flat files

R can read any of the types of files we just looked at by using one of the functions from the `readr` package:

File type	General function
Delimited	`read_delim`
Fixed width	`read_fwf`

You will just need to be able to clearly tell R *how* to read the file in, including what type of flat file it is and what delimiter it uses.

Reading in flat files

For example, the file "AWOIS_Wrecks_KnownYear.tab" is a flat delimited file with tabs as delimiters containing the subset of the Office of the Coast Survey's Automated Wreck and Obstruction Information System (AWOIS) for which the year the vessel sank is known.

You can download this file by going to this link and using the "Raw" button in the top right hand corner (right click and select "Download Linked File").

Reading in flat files

If save this file in your working directory, to read it in and assign it the name `shipwrecks`, you can run:

```r
library("readr")
shipwrecks <- read_delim("AWOIS_Wrecks_KnownYear.tab",
                         delim = "\t")
```

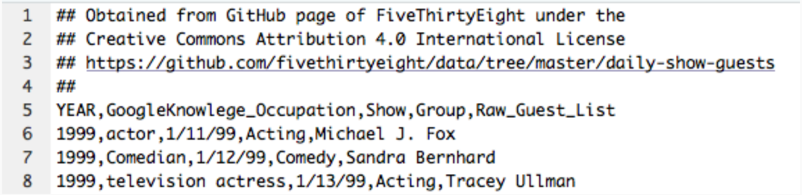
`readr` family of functions

Some of the interesting options with the `readr` family of functions are:

Option	Description
`skip`	How many lines of the start of the file should you skip?
`col_names`	What would you like to use as the column names?
`col_types`	What would you like to use as the column types?
`n_max`	How many rows do you want to read in?
`na`	How are missing values coded?

Reading in flat files

The "daily_show_guests.csv" file you worked with in the previous In-Course Exercise is a delimited flat file with commas as the delimiters. **It also has four lines of information about the data, before the actual data begins.**



```
## Obtained from GitHub page of FiveThirtyEight under the
## Creative Commons Attribution 4.0 International License
## https://github.com/fivethirtyeight/data/tree/master/daily-show-guests
##
YEAR,GoogleKnowlege_Occupation,Show,Group,Raw_Guest_List
1999,actor,1/11/99,Acting,Michael J. Fox
1999,Comedian,1/12/99,Comedy,Sandra Bernhard
1999,television actress,1/13/99,Acting,Tracey Ullman
```

Reading in flat files

You can handle this by using the `skip` option to tell R to skip the first four lines:

```
read_delim("daily_show_guests.csv", delim = ",", skip = 4)
```

```
## # A tibble: 2,693 x 5
##     YEAR GoogleKnowlege_~ Show  Group
##    <dbl> <chr>            <chr> <chr>
##  1  1999 actor            1/11~ Acti~
##  2  1999 Comedian         1/12~ Come~
##  3  1999 television actr~ 1/13~ Acti~
##  4  1999 film actress     1/14~ Acti~
##  5  1999 actor            1/18~ Acti~
##  6  1999 actor            1/19~ Acti~
##  7  1999 Singer-lyricist  1/20~ Musi~
##  8  1999 model            1/21~ Media
##  9  1999 actor            1/25~ Acti~
## 10  1999 stand-up comedi~ 1/26~ Come~
## # ... with 2,683 more rows, and 1 more variable:
## #   Raw_Guest_List <chr>
```

`readr` family of functions

Many members of the `readr` package that read delimited files are doing the same basic thing. The only difference is what defaults they have for the delimiter (`delim`).

Some key members of the `readr` family for delimited data:

Function	Delimiter
`read_csv`	comma
`read_csv2`	semi-colon
`read_table2`	whitespace
`read_tsv`	tab

`readr` family of functions

For any type of delimited flat files, you can also use the more general `read_delim` function to read in the file. However, you will have to specify yourself what the delimiter is (e.g., `delim = ","` for a comma-separated file).

For example, the following two calls do the same thing:

```r
read_delim("daily_show_guests.csv", delim = ",", skip = 4)
read_csv("daily_show_guests.csv", skip = 4)
```

`readr` family of functions

The `readr` package also includes some functions for reading in fixed width files:

- `read_fwf`
- `read_table`

These allow you to specify field widths for each fixed width field, but they will also try to determine the field-widths automatically.

Reading data from other files types

You can also read data in from a variety of other file formats, including:

File type	Function	Package
Excel	`read_excel`	`readxl`
SAS	`read_sas`	`haven`
SPSS	`read_spss`	`haven`
Stata	`read_stata`	`haven`

Reading in flat files

Once you read the data in, you should investigate it to make sure it looks like it was read in without bugs.

For example, you may want to look at a subset of the data using the tools you learned last week:

```
# Check out a subset of the data
library("dplyr")
slice(.data = select(.data = shipwrecks, c(2, 4, 5, 9)), 1:4)
```

```
## # A tibble: 4 x 4
##   VESSLTERMS    LATDEC LONDEC YEARSUNK
##   <chr>          <dbl>  <dbl>    <dbl>
## 1 SUBCHASER 187   37.3  -75.5     1918
## 2 BIRCH LAKE      37.3  -75.6     1943
## 3 PACIFIC         37.3  -75.6     1925
## 4 UNKNOWN         37.3  -75.7     1916
```

Reading in flat files

You can also use the head function to look at the first few rows:

```
head(x = shipwrecks, n = 3)
```

```
## # A tibble: 3 x 10
##   RECRD VESSLTERMS FEATURE_TYPE LATDEC LONDEC
##   <dbl> <chr>      <chr>         <dbl>  <dbl>
## 1   959 SUBCHASER~ Wreck - Sub~   37.3  -75.5
## 2   960 BIRCH LAKE Wreck - Sub~   37.3  -75.6
## 3   968 PACIFIC    Wreck - Sub~   37.3  -75.6
## # ... with 5 more variables: GP_QUALITY <chr>,
## #   DEPTH <dbl>, SOUNDING_TYPE <chr>,
## #   YEARSUNK <dbl>, HISTORY <chr>
```

Reading in flat files

Or you can also use the `tail` function to look at the last few rows:

```r
tail(x = shipwrecks, n = 3)
```

```
## # A tibble: 3 x 10
##   RECRD VESSLTERMS FEATURE_TYPE LATDEC LONDEC
##   <dbl> <chr>      <chr>         <dbl>  <dbl>
## 1  2764 EVENING S~ Not Charted    38.9  -74.6
## 2  2600 ACARA      Wreck - Sub~   40.6  -73.6
## 3  7722 HOWARD     Wreck - Sub~   40.6  -73.5
## # ... with 5 more variables: GP_QUALITY <chr>,
## #   DEPTH <dbl>, SOUNDING_TYPE <chr>,
## #   YEARSUNK <dbl>, HISTORY <chr>
```

Reading in flat files

Also, check that you have the number of rows and columns that you expect:

```r
dim(shipwrecks)
```

```
## [1] 197  10
```