Using read\_delim vs read\_csv

Rob Carver

March 2017

This is a quick note about reading a text file that uses a character *other* than a comma to separate variables in a text file.

We are well-acquainted with read.csv {base} and read\_csv {readr} for comma-separated-values files. Our Wine files use semi-colons to delimit the columns.

Here is code to handle that issue. We just need to specify the particular character that separates columns.

#read\_lines demo  
library(readr)  
red <- read\_delim("Data/winequality-red.csv", delim = ";")

## Parsed with column specification:  
## cols(  
## `fixed acidity` = col\_double(),  
## `volatile acidity` = col\_double(),  
## `citric acid` = col\_double(),  
## `residual sugar` = col\_double(),  
## chlorides = col\_double(),  
## `free sulfur dioxide` = col\_double(),  
## `total sulfur dioxide` = col\_integer(),  
## density = col\_double(),  
## pH = col\_double(),  
## sulphates = col\_double(),  
## alcohol = col\_double(),  
## quality = col\_integer()  
## )

## Warning: 2 parsing failures.  
## row col expected actual  
## 1296 total sulfur dioxide no trailing characters .5  
## 1297 total sulfur dioxide no trailing characters .5

head(red)

## # A tibble: 6 × 12  
## `fixed acidity` `volatile acidity` `citric acid` `residual sugar`  
## <dbl> <dbl> <dbl> <dbl>  
## 1 7.4 0.70 0.00 1.9  
## 2 7.8 0.88 0.00 2.6  
## 3 7.8 0.76 0.04 2.3  
## 4 11.2 0.28 0.56 1.9  
## 5 7.4 0.70 0.00 1.9  
## 6 7.4 0.66 0.00 1.8  
## # ... with 8 more variables: chlorides <dbl>, `free sulfur dioxide` <dbl>,  
## # `total sulfur dioxide` <int>, density <dbl>, pH <dbl>,  
## # sulphates <dbl>, alcohol <dbl>, quality <int>