

R Data Import/Export and Descriptive Statistics

Edited by: Hsiao-Yun Huang
Department of Statistics and Information Science,
Fu-Jen Catholic University





Imports

- Keyboard Input
- Reading data from text (ASCII)
- Reading data from Excel (csv)
- Importing data saved by other softwares.
 - SPSS
 - SAS
- Accessing data stored in a database management system



Keyboard Input

- Entering data into R with keyboard can be very efficient in some tasks.
- Some useful tips:
 - The combine function `c()`
 - `scan()`
 - `seq()`
 - `rep()`
 - `paste()`
 - `round()`
 - `data.frame()`
 - `edit()`
 - `fix()`



Data sets in R packages

- A lot of R packages have their own data sets and served as a good source of data sets for learning data analysis.
- `data()` # View the Data sets in basic package 'datasets'.
- `data(CO2)`
- `View(CO2)`
- `data(package="car")`
- `data(Davis,package="car")`
- `View(Davis)`



Reading data from text

- `get.wd()`
- `set.wd()`
- `read.table()`
 - `header`
 - `sep`
 - `skip`
- Data with no variable names
 - `colnames()`
-



Reading data from excel

- `getwd()`
- `setwd()`
- `read.csv()`
 - `header()`
 - `skip()`
- Data with no variable names
 - `colnames()`



Data from other Statistical Systems

- RStudio can import data from: Minitab, S-PLUS, SAS, SPSS, Stata, Systat....
- The package `foreign` provides facilities for files produced by these statistical systems.
- SAS
 - Function `read.xport()` reads a file in SAS Transport (XPORT) format and return a list of data frames.
 - If SAS is available on your system, function `read.ssd` can be used to create and run a SAS script that saves a SAS permanent dataset (`.ssd` or `.sas7bdat`) in Transport format. It then calls `read.xport` to read the resulting file.
 - For those without access to SAS but running on Windows, the SAS System Viewer (a zero-cost download) can be used to open SAS datasets and export them to e.g. `.csv` format.
 - For more details, please see <https://cran.r-project.org/doc/manuals/r-release/R-data.html>
- Spss
 - Function `read.spss()` can read files created by the 'save' and 'export' commands in SPSS. It returns a list with one component for each variable in the saved data set. SPSS variables with value labels are optionally converted to R factors.



Importing data from web (url)

- Example 1: Go to <https://data.montgomerycountymd.gov/>
 - Select a file you like.
 - Copy the url
 - Paste the url in `read.csv()`



The screenshot shows a web browser window displaying the 'dataMontgomery' website. The page title is 'Alcoholic Beverage License Violations'. The URL in the address bar is 'https://data.montgomerycountymd.gov/Business/units/Alcoholic-Beverage-License-Violations/431-104'. The page features a search bar, navigation links, and a main content area with a description of the dataset, update frequency, and 'About this Dataset' section. A dropdown menu is open over the 'Export Data' button, showing options like 'CSV', 'JSON', 'Excel', and 'PDF'. The Windows taskbar is visible at the bottom.

Alcoholic Beverage License Violations

The dataset includes alcohol violations as a result of sale to minor compliance efforts, routine inspections and enforcement efforts.

Update Frequency: Monthly

About this Dataset

Updated	Dataset Information
July 29, 2017	Department: Liquor Control, Department of
Data last updated: Jul 28, 2017	Update Frequency: Monthly
Metadata last updated: January 07, 2017	
Data issued: December 9, 2015	Topics



Importing data from web (url)

- Example 2: Go to <http://socserv.mcmaster.ca/jfox/Books/Companion/data.html>
- Select a file you like.
 - Copy the url
 - Paste the url in `read.csv()`



R Companion Second Edition

socserv.mcmaster.ca/fox/Books/Companion/data.html

An R Companion to Applied Regression, Second Edition

John Fox and Sanford Weisberg

Data Files Used in the Book

Disclaimer: These data files are provided for readers of Fox and Weisberg, *An R Companion to Applied Regression, 2nd Edition* (Sage, 2011). Others are welcome to use them, and we are of course always interested in learning about errors in the text or data files, but if you are not a reader of the book, please do not ask us questions about the data files.

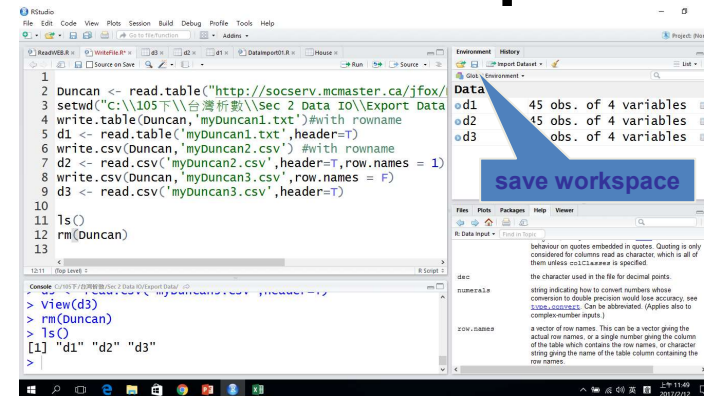
- Duncan's occupational-prestige data: [Duncan.txt](#)
- The to-be-or-not-to-be soliloquy from Hamlet: [Hamlet.txt](#)
- Canadian occupational-prestige data: [Prestige.txt](#)
- Canadian occupational-prestige data with errors: [PrestigeErrors.txt](#)
- Canadian occupational-prestige data in fixed format: [PrestigeFixed.txt](#)
- Excel file with worksheets for the Duncan and Canadian occupational-prestige data: [Assets.xls](#)

Note: Most of the examples in the *R Companion* use data in the [car package](#) for R.

下午 02:11 2017/10/2

Export data and workspace

- `setwd()`
- `write.table()`
- `write.csv`
- `row.names`
- Manage workspace
 - `ls()` # Lists the objects in the workspace
 - `rm()` #clearing space
 - `search()` # Shows the loaded packages
 - `library()` # Shows the installed packages
 - `dir()` # show files in the working directory



The screenshot shows the RStudio interface with the following code in the editor:

```
1 Duncan <- read.table("http://socserv.mcmaster.ca/jfox/
2 setwd("c:\\105下\\台灣折數\\Sec 2 Data IO\\Export Data
3 write.table(Duncan, 'myDuncan1.txt') #with rowname
4 d1 <- read.table('myDuncan1.txt', header=T)
5 write.csv(Duncan, 'myDuncan2.csv') #with rowname
6 d2 <- read.csv('myDuncan2.csv', header=T, row.names = 1)
7 write.csv(Duncan, 'myDuncan3.csv', row.names = F)
8 d3 <- read.csv('myDuncan3.csv', header=T)
9
10
11 ls()
12 rm(Duncan)
13
```

The Environment pane on the right shows three data objects:

- d1: 45 obs. of 4 variables
- d2: 45 obs. of 4 variables
- d3: 45 obs. of 4 variables

A blue arrow points to the Environment pane with the text "save workspace".





Missing Data

- `is.na()` # detect missing element in the dataframe in logical matrix
- `rowSums(is.na())` # Number of missing per row
- `colSums(is.na())` # Number of missing per column/variable
- `complete.cases()` #returns a logical vector indicating which cases are complete.
- `na.omit()` #returns the object with listwise deletion of missing values.



Categorical data: Frequencies

- `table()` #Count factor variable frequency; K way table
- `addmargins()` # Adding row/col margins

- `prop.table(,1)` # Row proportions
- `round(prop.table(,1), 2)` # Round row prop to 2 digits
- `round(100*prop.table(,1), 2)` # Round row prop to 2 digits (percents)
- `addmargins(round(prop.table(,1), 2),2)` # Round row prop to 2 digits

- `prop.table(,2)` # Column proportions
- `round(prop.table(,2), 2)` # Round column prop to 2 digits
- `round(100*prop.table(,2), 2)` # Round column prop to 2 digits (percents)
- `addmargins(round(prop.table(,2), 2),1)` # Round col prop to 2 digits

- `prop.table()` # Tototal proportions
- `round(prop.table(),2)` # Tototal proportions rounded
- `round(100*prop.table(),2)` # Tototal proportions rounded



Categorical data: Frequencies

- `chisq.test()` # Do chisq test Ho: no relationship
- `fisher.test()` # Do fisher'exact test Ho: no relationship
- `assocstats()` #association measures and degree of association.
- X²(chi-square) tests for relationships between nominal variables. The null hypothesis (H₀) is that there is no relationship.
- Cramer's V is a measure of association between two nominal variables. It goes from 0 to 1 where 1 indicates strong
- Fisher's exact test is used when there are very few cases in the cells (usually less than 5). It tests the relationship between two nominal variables.
- The null is that variables are independent.
- Source: <http://dss.princeton.edu/training/StataTutorial.pdf>



Descriptive Statistics For Continuous Variable

- `mean()` # Mean of all numeric variables
- `median()` #Median
- `var()` # Variance
- `sd()` # Standard deviation
- `max()` # Max value
- `min()` # Min value
- `range()` # Range
- `quantile()` # Quantiles 25%
- `quantile(mydata$SAT, c(.3,.6,.9))` # Customized quantiles
- `length(mydata$SAT)` # Num of observations when a variable is specify
- `length(mydata)` # Number of variables when a dataset is specify
- `which.max(mydata$SAT)` # Determines the location, i.e., index of the (first) minimum or maximum of a numeric vector"
- `which.min(mydata$SAT)` # From help: "Determines the location, i.e., index of the (first) minimum or maximum of a numeric vector"

Thank you for your attention!

THE END

