

# Getting Data In and Out of R



David Li

# Principal functions reading data

---

- ❑ `read.table`, `read.csv`, for reading tabular data
- ❑ `readLines`, for reading lines of a text file
- ❑ `source`, for reading in R code files (inverse of `dump`)
- ❑ `dget`, for reading in R code files (inverse of `dput`)
- ❑ `load`, for reading in saved workspaces
- ❑ `unserialize`, for reading single R objects in binary form

# Principal functions writing data

---

- ❑ `write.table`, for writing tabular data to text files (i.e. CSV) or connections
- ❑ `writeLines`, for writing character data line-by-line to a file or connection
- ❑ `dump`, for dumping a textual representation of multiple R objects
- ❑ `dput`, for outputting a textual representation of an R object
- ❑ `save`, for saving an arbitrary number of R objects in binary format (possibly compressed) to a file.
- ❑ `serialize`, for converting an R object into a binary format for outputting to a connection (or file).

# Video

---

▣ [https://youtu.be/Z\\_dc\\_FADyi4](https://youtu.be/Z_dc_FADyi4)

▣ ?read.table

```
read.table(file="http://statweb.stanford.edu/~rag/stat141/exs/whale.txt"),header=T) #  
read from internet
```

▣ ?readLines

```
readLines("http://statweb.stanford.edu/~rag/stat141/exs/whale.txt")
```

# Using dput() and dump()

---

## □ dput()/dget()

- `y <- data.frame(a = 1, b = "a")`
- `dput(y)`
- `dput(y, file = "y.R")`

## □ dump()/source()

- `x <- "foo"; y <- data.frame(a = 1L, b = "a")`
- `dump(c("x", "y"), file = "data.R")`
- `rm(x, y)`
- `source("data.R")`
- `str(y)`

# Difference between dput() and dump()

---

- ❑ dump can be used to output multiple objects
- ❑ dump adds the object name and can be source()'d

# Binary Formats `save()/load()`

---

- ❑ `a <- data.frame(x = rnorm(100), y = runif(100))`
- ❑ `b <- c(3, 4.4, 1 / 3)`
- ❑ *## Save 'a' and 'b' to a file*
- ❑ `save(a, b, file = "mydata.rda")`
- ❑ *## Load 'a' and 'b' into your workspace*
- ❑ `load("mydata.rda")`
- ❑ *## Save everything to a file*
- ❑ `save.image(file = "mydata.RData")`
- ❑ *## load all objects in this file*
- ❑ `load("mydata.RData")`

# Reading in Larger Datasets with read.table

---

□ Video <https://youtu.be/BJYYIJO3UFI>

□ A tip

```
> initial <- read.table("datatable.txt", nrow = 100)
> classes <- sapply(initial, class)
> tabAll <- read.table("datatable.txt", colClasses = classes)
```



# Summary

---

- ❑ `write.csv()` and `write.table()` are used when you want to exchange data in tabular text format.
- ❑ `dput()` saves single data object in R code
- ❑ `dump()` saves multiple data objects and their metadata in R code
- ❑ `save()` is similar to `dump()` but saves in binary format or ASCII
- ❑ `save.image()` saves workspace in binary format

# Calculating Memory Requirements for R Objects

---

- An example: a data frame with 1,500,000 rows and 120 columns, all of which are numeric data.
  - $1,500,000 \times 120 \times 8$  bytes/numeric

# Computing Lab Ex.

---

## ▣ Lab 2

# Next week

---

- ▣ Control Structures