# Lab 6: Exporting Data and Output from R

Stat 159, Fall 2016, Prof. Sanchez

*October 4, 2016*

## Data Set `mtcars`

To illustrate the different data exporting possibilities, as well as writing output features in R, we are going to use the data frame `mtcars` that comes in R:

```
head(mtcars)
```

```
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

## Writing tables

One common task in most data analysis projects involves exporting data tables. For this purposes you can use `write.table()` or `write.csv()`. Here are a couple of examples that export `mtcars` to text files using different types of field separators. Note that these examples assume you want the files in your working directory.

```
# blank separated (default)
write.table(mtcars, file = 'mtcars.txt', row.names = FALSE)

# tab-separated value
write.table(mtcars, file = 'mtcars.tsv', sep = "\t", row.names = FALSE)

# comma-separated value
write.csv(mtcars, file = 'mtcars.csv', row.names = FALSE)
```

## Sending output with `cat()`

You can use `cat()` to concatenate and print information to a file. For instance, say you are interested in some descriptive statistics about the column `mpg` (miles-per-gallon):

```r
# summary statistics of mpg
min(mtcars$mpg)
max(mtcars$mpg)
median(mtcars$mpg)
mean(mtcars$mpg)
sd(mtcars$mpg)
```

The goal is to generate a file `mpg-statistics.txt` with the following contents:

```
Miles per gallon summary statistics

Minimum: 10.4
Maximum: 33.9
Median : 19.2
Mean   : 20.09
Std Dev: 6.02
```

Here's one way to start:

```r
# summary statistics of mpg
mpg_min <- min(mtcars$mpg)
mpg_max <- max(mtcars$mpg)
mpg_med <- median(mtcars$mpg)
mpg_avg <- mean(mtcars$mpg)
mpg_sd <- sd(mtcars$mpg)

# name of output file
outfile <- "mpg-statistics.txt"

# first line of the file
cat("Miles per gallon summary statistics\n\n", file = outfile)
# subsequent lines appended to the output file
cat("Minimum:", mpg_min, "\n", file = outfile, append = TRUE)
cat("Maximum:", mpg_max, "\n", file = outfile, append = TRUE)
cat("Median :", mpg_med, "\n", file = outfile, append = TRUE)
cat("Mean   :", mpg_avg, "\n", file = outfile, append = TRUE)
cat("Std Dev:", mpg_sd, "\n", file = outfile, append = TRUE)
```

To make it "prettier" you may consider using `sprintf()`

```r
sprintf('Minimum: %s', mpg_min)
```

Now let's re-export the lines:

```r
cat("Miles per gallon summary statistics\n\n", file = outfile)
cat(sprintf('Minimum: %0.2f', mpg_min), "\n", file = outfile, append = TRUE)
cat(sprintf('Maximum: %0.2f', mpg_max), "\n", file = outfile, append = TRUE)
cat(sprintf('Median : %0.2f', mpg_med), "\n", file = outfile, append = TRUE)
cat(sprintf('Mean   : %0.2f', mpg_avg), "\n", file = outfile, append = TRUE)
cat(sprintf('Std Dev: %0.2f', mpg_sd), "\n", file = outfile, append = TRUE)
```

**Your turn:** How would you avoid writing that many calls to `cat()`?

## Sending R output to a file with `sink()`

Another interesting function is `sink()`. This function is very useful when you want to export R output as is displayed in the R console. For example, consider the output from `summary()`

```r
summary(mtcars$mpg)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  10.40   15.42   19.20   20.09   22.80   33.90
```

You could assign the output of `summary(mtcars$mpg)` to an object `mpg_summary` and then try `writeLines()` to export the results to a file `mpg-summary.txt`, but you won't keep the same format of R:

```r
mpg_summary <- summary(mtcars$mpg)
writeLines(mpg_summary, con = "mpg-summary.txt")
```

To be able to keep the same output display of R, you must use `sink()`. This function will **divert** or redirect R output to the specified file:

```r
# sink output
sink(file = "mpg-stats2.txt")
# summary statistics of mpg
summary(mtcars$mpg)
# stops diverting output
sink()
```

**Your turn:** Use `sink()` to send the output from running a linear regression of `hp` on `mph` with the function `lm()`. Also export the results from using `summary()` on the regression object. And/or try running a t-test between `mpg` and `hp` with `t.test()`.

## Exporting tables with `xtable()`

Another interesting tool to export tables in LaTeX or HTML formats is provided by the R package `"xtable"` and its main function `xtable()`.

```
library(xtable)

# linear regression
reg <- lm(hp ~ mpg, data = mtcars)

# create xtable and export it
reg_table <- xtable(reg)
print(reg_table, type = "latex", file = "reg-table.tex")
print(reg_table, type = "html", file = "reg-table.html")
```

## R's Binary Data

R also allows you to save objects in R's binary format with the functions `save()` and `save.image()`. It is customary to use the `.RData` extension for the files created by `save()` and `save.image()`. You may also encounter users specifying the old extension `.rda` or some other variation.

You can use `save()` to save specific objects from your current session. For example, here is how to save the data frame `mtcars` to your working dir:

```
save(mtcars, file = 'mtcars.RData')
```

The difference between `save()` and `save.image()` is that the latter saves all the objects in your current session. This is actually the function that is run behind the scenes everytime you quit R and accept to save the so-called *workspace image*.

You can share `mtcars.RData` with any other R user, regardless of the operating system that they use. To read in binary R files, use `load()`.

**Your turn:** Subset the data set `mtcars` for cars with automatic transmission, `am == 1`, and export the resulting data using both `write.table()` and `save()`. Compare the size of the produced files.