Guest: Dr. Dennis Jennings

Today we hopefully will welcome Dr. Dennis Jennings, a 1982 graduate of the department. He has extensive experience in the pharmaceutical industry, including leadership of clinical operations, statistics and data management functions, in companies such as Takeda, TAP, and Abbott. If he's able to come, he'll come in about 2:50; until then we'll talk about computing workflow during an project.

Workflow in R

by Rob J. Hyndman, Professor of Statistics, Monash University, Australia,
18 September 2009, `http://robjhyndman.com/hyndsight/workflow-in-r/`

This came up recently on Stack Overflow (`www.stackoverflow.com`). One of the answers was particularly helpful and I thought it might be worth mentioning here. The idea presented there is to break the code into four files, all stored in your project directory. These four files are to be processed in the following order.

- `load.R` This file includes all code associated with loading the data. Usually, it will be a short file reading in data from files.
- `clean.R` This is where you do all the pre-processing of data, such as taking care of missing values, merging data frames, handling outliers. By the end of this file, the data should be in a clean state, ready to use. It is much better to do this here rather than clean the data on the original file as this enables you to have a complete record of everything done to the data.
- `functions.R` All of the functions needed to perform the actual analysis are stored here. This file should do nothing other than define the functions you need for analysis. (If you require your own functions for loading or cleaning the data, include them at the top of either `load.R` or `clean.R`.) In particular, `functions.R` should not do anything to the data. This means that you can modify this file and reload it without having to go back and repeat steps 1 & 2 which can take a long time to run for large data sets.
- `do.R` Here is the code to actually do the analysis. This file will use the functions defined in `functions.R` to do the calculations, produce figures and tables, etc. All figures and tables that end up in your report, paper or thesis should be coded here. Never create figures and tables manually (i.e., with the mouse and menus) as then you can't easily reproduce.

There are many advantages to this setup. First, you don't have to reload the data each time you make a change in a subsequent step. Second, if you come back to an old project, you will be able to work out what was done relatively quickly. It also forces a certain amount of structured thinking in what you are doing, which is helpful.

Often there will be bits and pieces of code that you write, but don't end up using, yet don't want to delete. These should either be commented out or saved in files with other names. All analysis from reading data to producing the final results should be reproducible by simply `source()`ing these four files in order with no further user intervention.

Example 1

```
## in file "todayswork.R"
data <- read.csv("mydata.csv")
attach(data)
mymodel=lm(V1 ~V2 + V3+V4)
summary(mymodel)
```

```
## in file "cars/analyze_mpg-2014-04-23.R"
d <- read.csv("cars.csv")
model.mpg <- lm(mpg ~ cyl + disp + hp, data=d)
summary(model.mpg)
```

Example 2

```
average<-mean(feet/12+inches,na.rm=TRUE)

average <- mean(feet / 12 + inches, na.rm = TRUE)
```

Example 3

```
if (y < 0 && debug) {
  message("Y is negative")
}

if (y == 0) {
log(x)
  } else {
  y ^ x
}
```

```
if (y < 0 && debug)
message("Y is negative")

if (y == 0) {
  log(x)
}
else {
  y ^ x
}
```

STAT8801, April 23, 2014
Dr. Jennings/Computing Workflow

Name: _____

Tell me something interesting you learned about code and workflow.

What's a question you'd like to ask Dr. Jennings?

Tell me one interesting thing you heard from Dr. Jennings.