Inputting Data Sets

We have several data sets on our web page that we will be using multiple times over the course of the semester. This exercise ensures that you have enough familiarity with SAS and R's capabilities for reading large data sets into SAS's default WORK directory and R's default workspace.

We will be working with a tab-delimited text file, `Fall8.txt`, that includes coded information on the Fall 2008 first-year cohort of students at University of South Carolina. Download the file from the web and inspect it; make sure that the file is saved with its original extension. Note that it lacks variable names, and includes six fields, most of which are character variables. The six variables are Major Code/Major Text Description, Degree, Class, GPA, Gender and Student ID.

SAS

We could use an INFILE statement here, but I find it has largely been superceded by `PROC IMPORT`. I saved the text file on the department's new web server, which SAS identified as drive
W: during the import session. I selected `Import Data...` from the `File menu`, then selected Tab-delimited Text as my data source. After browsing to select my file, I clicked Options to make sure that the option to get variable names from the first row was de-selected (since the data set doesn't include variable names). After calling the data set `FALL08`, I saved the import wizard code in the SAS file `Fall08import`, then added a little more code to assign variable names; here is the final code, in case I don't want to step through the Import Wizard each time I use this data set.

```
PROC IMPORT OUT=WORK.FALL08
            DATAFILE="W:\courses\stat704\Fall8.txt"
            DBMS=TAB REPLACE;
      GETNAMES=NO;
      DATAROW=1;
RUN;
DATA FALL08; SET FALL08;
RENAME VAR1=Major VAR2=Degree VAR3=Class VAR4=GPA VAR5=Gender VAR6=ID;
RUN;
```

Note that I could save `FALL08` as a permanent SAS data set for later use, but chose not to do so here. The inconvenience of assigning new names each time the data set is loaded seems minor.

R

R's workhorse command for text files has always been `read.table`. There are related versions of this command that work well for other formats (e.g., `read.csv` for comma-delimited data sets), and `read.delim` works well for reading tab-delimited data sets. The following commands will read in the data set then assign column names. Rather than use

a lengthy directory prefix, I always select `Change dir...` under the File menu (or `Change Working Directory...` under the Misc menu to simplify the file name. If `na.strings` is omitted, GPA is treated as a factor, not a numeric variable.

```
Fall08.df <- read.delim("Fall8.txt",header=F,na.strings=".")
names(Fall08.df) <- c("Major","Degree","Class","GPA","Gender","ID")
```

Assignment

Use SAS and R to input the comma-delimited text file on the website containing SCDOT traffic count data, `traffic_count_data_2012.txt`. We use these data sets and others to develop sampling plans for the statewide safety belt survey, which the Stat Lab has supervised since 1992.

As with our previous data set, this data set does not have a header for variable names. The variables are described below (you can select your own variable names):

Variable Name	Information
County ID	1=Abbeville, 2=Aiken, ...
Station	Monitoring Station
Road Type	1=Interstate, 2=US Primary,...
Road Number	E.g, 20 for I-20, 76 for US 76
Road Class	0=Main, ... 5=Spur, ..., 8=Bypass
AADT	Average Annual Daily Traffic Count
Survey Year	
Station Description	

The input methods will be similar to those described above, though adjustments will need to be made here and there. Be sure to report on your efforts–I expect these assignments to be "process-oriented", and it is important for me to read how you respond to these assignments as you work your way through them.