

## STAT 517: Takehome 2

1. Student names, their scores, and the self-assigned weights for these scores, appear in the attached worksheet. (35 points)
  - (a) Read the data into SAS as the data set **Gradebook**.
  - (b) Use an **ARRAY** statement and **DO** loop to compute a course average. Undergrads should compute an unweighted course average, while grad students should compute a weighted course average using the weights for each assignment.
  - (c) Use an **IF** statement to create a character variable representing the letter grade corresponding to the course average (you should use the STAT 517 grading scale).
  - (d) In **PROC PRINT**, print the student's name, their course average, and their letter grade; use an attractive output format.
  - (e) Graduate students should print the students' names in alphabetical order.
2. The Excel file contains two separate measurements on global surface temperatures. (35 points)
  - (a) Import the data from the Excel file into SAS, and save it in a SAS library other than **WORK**, then read it into a datafile in your **WORK** library. How did SAS handle the missing value code "X" for the variable **Temperature 1**?
  - (b) What is the minimum temperature for the variable **Temperature 1**? What is the maximum temperature for the variable **Temperature 1**?
  - (c) Create a new variable called **AvTemp**, which is the average of the two temperature measurements for each year. Don't worry about missing data.
  - (d) Sort the resulting data set by the values of **AvTemp**, from greatest to least.
  - (e) After removing the records with missing observations, print the resulting sorted values of **Year** and **AvTemp** in the output window with a descriptive title. Do the variables have attractive names and readable formats? If not, modify the formats and names and print the variables in the output window again.
  - (f) In addition to the above steps, graduate students should compute averages for complete data records only, then write **Year** and **AvTemp** from those complete records to an external file. Be sure **AvTemp** has a readable format in the external file.
3. A waiter was assigned the following schedule for the month of October. (30 points)

10/01/09 10/06/09 10/07/09 10/10/09 10/16/09 10/20/09 10/22/09 10/25/09 10/27/09  
10/29/09

  - (a) Read these dates into SAS using **INFILE** or **DATALINES** statements; the dates should all be listed on a single line, though you can format them any way you would like within that line.

- (b) Find the week and weekday of each date. Using `PROC FREQ`, report how many days each week the waiter worked, and then how many of each weekday (i.e., how many Mondays, etc) that the waiter worked. Use `PROC FORMAT` to print better labels for the weekdays.
- (c) *Grad students*: The waiter receives a paycheck on the third Friday and the fifth Friday of the month. For each workday, compute the appropriate payday.
- (d) Print an output file; each record should include the work day, the day of the week (Monday, etc), and the associated pay day. You can use whatever formats or combinations of variables you like, but they should be reasonably attractive.