MET MA 603: SAS Programming and Applications

Exam 2 Part II

80 points

Exam Rules

- No collaboration, notes, or other outside resources are allowed, except for a hand-written 3 by 5 inch index card.
- Save solutions to all problems in a single SAS file and upload it to Blackboard. Grading will be based on the submitted code. Do not upload any datasets or other files.
- Include your name in the name of the SAS file.
- Multiple attempts are allowed as long as they are submitted before the deadline. The most recent attempt submitted before the deadline will be the one that is graded.
- Points will be deducted from late submissions.

Question 6 (12 points)

You are an official at a diving competition.

Twenty divers each had 5 dives which were scored by the judges. The dataset Diving.sas7bdat contains the 100 scores.

According to the rules of the competition, the Final Score for each diver is based on the average of their dives, with the highest and lowest score omitted (thus each diver has three dives that are used to calculate their Final Score). For example if a diver had scores of 8.6, 9.4, 8.8, 8.9, and 9.5, the overall score would be (9.4 + 8.8 + 8.9) / 3 = 9.03.

Create a SAS dataset that contains only the name of each diver and their Final Score.

Question 7 (20 points)

You work in the accounting department of a car dealership.

Customers order cars by choosing the model of car as well as any special features they want. Special features cost extra and are indicated in the customer's order with a 1.

The dataset Car_Orders.sas7bdat contains the information for all of the cars ordered today.

The dataset Base_Model_Price_List.sas7bdat contains the prices for each model (before adding any special features), and the dataset Special_Features_Price_List.sas7bdat contains the prices for the special features (note that the order of the special feature columns is the same as it is in the Car_Orders dataset).

Create a dataset which contains the total price for each order, which is the price of the model + the price of all special features included in the order.

Question 8 (10 points)

Using the dataset base_model_price_list.sas7bdat, create a text file on your Desktop called “ModelPrices” that matches the requirements below:

1. Columns 1-9 contains the name of your car dealership, which is “SAS-Honda”.
2. Columns 11-20 should contains the name of the model. The model name must be formatted in uppercase letters (this is done using the UPCASE format).
3. Columns 21-28 should contain the base price of the model. The price should be formatted using the DOLLAR format with 0 decimal places shown.

Your text file does not need to contain column headers.

Question 9 (20 points)

You are a stockbroker.

Your colleague has the following investing strategy: when the market (measured here by the Dow Jones) goes up, he buys because he thinks it will go up again the next day. When the market goes down, he sells because he thinks the market will go down the next day.

The dataset Dow_Jones.sas7bdat has the daily closing price of the Dow Jones from 1900 to 1999.

Use Proc Corr (you will need to do some manipulation using Data Steps first) to measure the correlation between price changes on consecutive days (when the market was open for trading).

Indicate using a comment why you think your colleague has a good or bad strategy.

Question 10 (18 points)

Use the Diving.sas7bdat dataset.

Create a macro that takes as an input the name of a diver, and creates a dataset that contains the dive records for that diver. The name of the dataset that is created should be based on the name of the diver + "_Dives", e.g., Viger_Dives. The macro must work regardless of which diver's name is used as the input to the macro.

For example, if the macro is executed with an input of Viger, it should create a dataset like the table below.

VIEWTABLE: Work.Viger_dives

	Diver	Dive	Score
1	Viger	1	10
2	Viger	2	10
3	Viger	3	9.8
4	Viger	4	9.8
5	Viger	5	10

Formats

Format	Definition	Example
$UPCASE*w*.	Converts character data to uppercase	HELLO
Date*w*.	Writes SAS date values in form *ddmmmyy* or *ddmmmyyyy* where mmm represents the first three letters of the month name	10OCT17
MMDDYY*w*.	Writes SAS date values in form *mm/dd/yy* or *mm/dd/yyyy*	10OCT2017
TIME*w*.	Writes SAS times values in form *hh:mm:ss.ss*	11:25:45.03
COMMA*w.d*	Writes numbers with commas	100,000
DOLLAR*w.d*	Writes numbers with a leading $ and commas	$25.10
PERCENT*w.d*	Writes numeric data as percentages. Negative numbers indicated with parentheses	(25%)
PERCENTN*w.d*	Writes numeric data as percentages. Negative numbers indicated with the minus sign	12.02%