MET MA 603: SAS Programming and Applications

Exam 1 Part II

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 5

An actuary is doing a reserve analysis using the data contained in the SAS dataset LDFs.sas7bdat.

The actuary wants to calculate a statistic for each Maturity called "Mean x-hi/lo", which is based on the mean excluding the highest and lowest values.

For example, the Maturity of 120 has the factors 0.7620, 1.0292, 0.9251, 0.9292, 1.1735, 1.0220, and 1.0064. The highest and lowest value, 1.1735 and 0.7620, are eliminated, and the Mean x-hi/lo is calculated as:

$$(1.0292 + 0.9251 + 0.9292 + 1.0220 + 1.0064) / 5 = 0.9824.$$

Use SAS to calculate the Mean x-hi/lo statistic for each Maturity.

Question 6

Below is the definition of the mathematical constant *e*.

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = \frac{1}{1} + \frac{1}{1} + \frac{1}{1 \cdot 2} + \frac{1}{1 \cdot 2 \cdot 3} + \cdots$$

e can be estimated by calculating the first m terms of the sequence. The higher the value of m, the closer the estimate will be to the true value of *e*.

In SAS, the function exp(x) raises *e* to the power x, thus exp(1) = *e*.

Use a Do Loop to determine the value of m that will produce an estimate of *e* that is within 0.00001% of the true value.

Note: The function abs(x) returns the absolute value of x.

Note: The function fact(x) calculates x!

Question 7

Use Proc Import to create a SAS dataset based on the external file exit_poll.csv. The data contains a survey asking which candidate (Jones or Grant) each person voted for.

Use Proc Freq to identify these issues in the data (do not consider FirstName and LastName in the analysis):

- An observation has one or more missing values (the observation should be removed).
- An observation has illogical or contradictory data (the observation should be removed).
- An observation has a word with a misspelling or inconsistent case (the word should be corrected).
- Every observation has the same value for a variable (the variable should be removed).

Use a Data Step to clean the data: use IF-THEN statements and dataset options to make any necessary corrections to observations and variables.

Question 8

Use the Data Step to create a SAS dataset based on the external file life_insurance.txt. All variables except for “Cause” should be of the numeric datatype.

Format the resulting SAS dataset to look similar to the external file.

Question 9

The SAS dataset world_cup.sas7bdat was not created correctly. Use functions to create five separate variables.

The corrected dataset should look similar to the one below:

VIEWTABLE: W

	Year	Winner	Score	RunnerUp
1	1930	Uruguay	4–2	Argentina
2	1934	Italy	2–1	Czechoslovakia
3	1938	Italy	4–2	Hungary
4	1950	Uruguay	2–1	Brazil
5	1954	WestGermany	3–2	Hungary
6	1958	Brazil	5–2	Sweden
7	1962	Brazil	3–1	Czechoslovakia
8	1966	England	4–2	WestGermany
9	1970	Brazil	4–1	Italy
10	1974	WestGermany	2–1	Netherlands
11	1978	Argentina	3–1	Netherlands
12	1982	Italy	3–1	WestGermany
13	1986	Argentina	3–2	WestGermany
14	1990	WestGermany	1–0	Argentina
15	1994	Brazil	0–0	Italy
16	1998	France	3–0	Brazil
17	2002	Brazil	2–0	Germany
18	2006	Italy	1–1	France
19	2010	Spain	1–0	Netherlands
20	2014	Germany	1–0	Argentina

Appendix

Point Distribution

Question	Points
5	16
6	16
7	16
8	16
9	16
Part II Total	80

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%

Informats

Informat	Definition
$CHARw.	Reads character data - does not trim leading or trailing blanks
$w.	Reads character data - trims leading blanks
Datew.	Reads dates in form *ddmmmyy* or *ddmmmyyyy* where *mmm* represents the first three letters of the month name
MMDDYYw.	Reads dates in form *mmddyy* or *mmddyyyy* where a special character such as / or - separates the month, day and year
DDMMYYw.	Reads dates in form *ddmmyy* or *ddmmyyyy* where a special character such as / or - separates the day, month and year
COMMAw.d	Removes embedded commas and $, converts left parentheses to minus sign
PERCENTw.d	Converts percentages to numbers

Character Functions

Function	Inputs	Definition
ANYALNUM	*(arg)*	Returns position of first occurance of any alphabetic character or numeral
ANYALPHA	*(arg)*	Returns position of first occurance of any alphabetic character
ANYDIGIT	*(arg)*	Returns position of first occurance of any numeral
ANYSPACE	*(arg)*	Returns position of first occurrence of a white space character
CAT	*(arg1, arg2, etc.)*	Concatenates (combines) two or more character strings together leaving any blanks
COMPRESS	*(arg)*	Removes spaces from character data
LENGTH	*(arg)*	Returns the length of an argument (missing values have a length of 1)
FIND	*(arg1, arg2)*	Returns position of *arg2* within the *arg1*
PROPCASE	*(arg)*	Converts first character in word to uppercase and remaining characters to lowercase
SUBSTR	*(arg,position,n)*	Extracts a substring from an *argument* starting at *position* for *n* characters
TRIM	*(arg)*	Removes trailing blanks from character expression
UPCASE	*(arg)*	Converts all letters in *argument* to uppercase

Numeric Functions

Function	Inputs	Definition
INT	*(arg)*	Returns the integer portion of the argument
LOG	*(arg)*	Returns the natural logarithm
LOG10	*(arg)*	Returns the logarithm to the base 10
MAX	*(arg1, arg2, etc.)*	Returns the largest non-missing value
MEAN	*(arg1, arg2, etc.)*	Returns the arithmetic mean of non-missing values
MIN	*(arg1, arg2, etc.)*	Returns the smallest non-missing value
SUM	*(arg1, arg2, etc.)*	Returns the sum of non-missing values

Numeric (Date-related) Functions

Function	Inputs	Definition
DAY	*(date)*	Returns the day of the month from a SAS date value
MDY	*(month,day,year)*	Returns a SAS date value from *month*, *day* and *year* values
MONTH	*(date)*	Returns the month (1-12) from a SAS date value
TODAY	*()*	Returns the current date as a SAS date value
YEAR	*(date)*	Returns the year from a SAS date value