Q1. I conducted the following codes to assign MACRO variables, would you please tell me what is wrong in the section marked yellow?

27 data WORK.e ;

28 %let \_EFIERR\_ = 0; /\* set the ERROR detection macro variable \*/

29 infile 'M:\Public\eclskw.csv' delimiter = ',' MISSOVER DSD lrecl=32767 firstobs=2 ;

30 informat variable $8. ;

31 format variable $8. ;

32 input

33 variable $

34 ;

35 if \_ERROR\_ then call symputx('\_EFIERR\_',1); /\* set ERROR detection macro variable \*/

36 run;

NOTE: The infile 'M:\Public\eclskw.csv' is:

Filename=M:\Public\eclskw.csv,

RECFM=V,LRECL=32767,File Size (bytes)=209,

Last Modified=20Jun2018:15:53:48,

Create Time=20Jun2018:11:29:09

NOTE: 21 records were read from the infile 'M:\Public\eclskw.csv'.

The minimum record length was 5.

The maximum record length was 8.

NOTE: The data set WORK.E has 21 observations and 1 variables.

NOTE: DATA statement used (Total process time):

real time 0.03 seconds

cpu time 0.00 seconds

21 rows created in WORK.e from M:\Public\eclskw.csv.

NOTE: WORK.E data set was successfully created.

NOTE: The data set WORK.E has 21 observations and 1 variables.

37 data \_null\_;

38 set e;

39 call symput(n , variable);

40 run;

NOTE: Numeric values have been converted to character values at the places given by:

(Line):(Column).

39:17

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','CHILDID ') at line 39 column 10.

variable=CHILDID n=. \_ERROR\_=1 \_N\_=1

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','PARENTID') at line 39 column 10.

variable=PARENTID n=. \_ERROR\_=1 \_N\_=2

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','S1\_ID ') at line 39 column 10.

variable=S1\_ID n=. \_ERROR\_=1 \_N\_=3

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','T1\_ID ') at line 39 column 10.

variable=T1\_ID n=. \_ERROR\_=1 \_N\_=4

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','CS\_TYPE2') at line 39 column 10.

variable=CS\_TYPE2 n=. \_ERROR\_=1 \_N\_=5

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','CREGION ') at line 39 column 10.

variable=CREGION n=. \_ERROR\_=1 \_N\_=6

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','KURBAN\_R') at line 39 column 10.

variable=KURBAN\_R n=. \_ERROR\_=1 \_N\_=7

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','R3REGION') at line 39 column 10.

variable=R3REGION n=. \_ERROR\_=1 \_N\_=8

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','F2SPECS ') at line 39 column 10.

variable=F2SPECS n=. \_ERROR\_=1 \_N\_=9

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','E2SPEDC ') at line 39 column 10.

variable=E2SPEDC n=. \_ERROR\_=1 \_N\_=10

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','P2OFTEN5') at line 39 column 10.

variable=P2OFTEN5 n=. \_ERROR\_=1 \_N\_=11

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','P2ATTENS') at line 39 column 10.

variable=P2ATTENS n=. \_ERROR\_=1 \_N\_=12

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','P2PARGRP') at line 39 column 10.

variable=P2PARGRP n=. \_ERROR\_=1 \_N\_=13

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','P2OFTEN2') at line 39 column 10.

variable=P2OFTEN2 n=. \_ERROR\_=1 \_N\_=14

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','C1R4RTSC') at line 39 column 10.

variable=C1R4RTSC n=. \_ERROR\_=1 \_N\_=15

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','C2R4RTSC') at line 39 column 10.

variable=C2R4RTSC n=. \_ERROR\_=1 \_N\_=16

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','P2HOWCHD') at line 39 column 10.

variable=P2HOWCHD n=. \_ERROR\_=1 \_N\_=17

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','P2CHILDR') at line 39 column 10.

variable=P2CHILDR n=. \_ERROR\_=1 \_N\_=18

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','P2WRKSHP') at line 39 column 10.

variable=P2WRKSHP n=. \_ERROR\_=1 \_N\_=19

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: Invalid argument to function SYMPUT(' .','P2TEACHI') at line 39 column 10.

WARNING: Limit set by ERRORS= option reached. Further errors of this type will not be printed.

variable=P2TEACHI n=. \_ERROR\_=1 \_N\_=20

ERROR: Symbolic variable name . must begin with a letter or underscore.

NOTE: The SAS System stopped processing this step because of errors.

NOTE: There were 21 observations read from the data set WORK.E.

NOTE: DATA statement used (Total process time):

real time 0.03 seconds

cpu time 0.03 seconds

WARNING: Apparent symbolic reference N not resolved.

41 %put &n. ;

&n.

**In callsymput statement put n in quotes. Like this call symput(‘n’,variable). Again it will try to store all the variable values in a single macro variable that is in n and it has only final value that is P2TEACHI.**

Q2. I write a MACRRO to create frequency distribution, and the assigned positon of MACRO variable is the path where the data is, but I do not know why it stops. Would you please correct the following errors? Thank you.

322 %SYSMSTORECLEAR;

323

324 %let path='C:\Users\um175569\Desktop\New folder';

325 %put &path.;

'C:\Users\um175569\Desktop\New folder'

326 proc export data = WORK.output outfile = "&path.\output.xlsx"

327 dbms = xlsx replace;

328 run;

Error creating XLSX file -> ('C:\Users\um175569\Desktop\New folder' 'D:\\output.xlsx') . It is

either not an Excel spreadsheet or it is damaged. Error code=80001002

Requested Output File is Invalid

ERROR: Export unsuccessful. See SAS Log for details.

NOTE: The SAS System stopped processing this step because of errors.

NOTE: PROCEDURE EXPORT used (Total process time):

real time 0.00 seconds

cpu time 0.00 seconds

/\*\*BUT CREATED DATA CAN BE SUCCESSFULLY EXPROTED INTO TEXT FILE\*\*/

NOTE: The file 'C:\Users\um175569\Desktop\t.txt' is:

Filename=C:\Users\um175569\Desktop\t.txt,

RECFM=V,LRECL=32767,File Size (bytes)=0,

Last Modified=21Jun2018:10:23:37,

Create Time=21Jun2018:10:23:35

NOTE: 9225 records were written to the file 'C:\Users\um175569\Desktop\t.txt'.

The minimum record length was 20.

The maximum record length was 70.

**The problem is with the path which you are exporting the xlsx file. In the path there is New folder which is having space between New and folder. So avoid it and instead give New\_folder or Newfolder without any spaces in the path location. While exporting to text file you wont find any spaces and there is no New folder.**

Q3.

What is difference between SAS/ACCESS and SAS/CONNECT? Would you please provide me real world examples? How do you use SAS to check data accuracy/validity? Would you please provide real world example of using SAS for QC? Thank you.

**SAS/ACCESS is used to connect to an external database and read and write to that database through SAS environment. Suppose if you have an Oracle database which is holding the data in Oracle tables so your aim is to read that data and do some analysis. And after reading the database you can get them and save into SAS as SAS datasets.In this case you will use SAS/ACCESS package which is to be installed while doing SAS installation by admin. There are two ways you can access data in Oracle database one is through Implicit Pass through and other is explicit pass through.**

**SAS/CONNECT is a different package when compared to SAS/ACCESS often people get confused between both of them. SAS/CONNECT is used to connect one SAS Server to another to download and upload the data. Suppose if there is huge amount of data in Server1 on which SAS is not installed and you have SAS installed on Server2 and you want to access some data of Server1 with SAS on Server2 so you will use SAS/CONNECT package to connect to Server1 and download data to SAS Server2.**

**SAS QC is for datacleaning, quality checks, data validation using various techniques. The Key features are Data cleasing, Data Profiling,Business Rule validation, Entity resolution, Data Integration. Suppose you are maintaining customer database of a Global retain market which is located across various countries. So in order to consolidate all the customers at one place we need to clean the data and remove the duplicates, remove the redundant customer details, remove the missing values and validate with some business rules like customer should have both first and last names.**

Q6. The following codes come from an on-line technical paper. I appreciate that author of sharing his copy right. I am eager to learn. Would you please explain meaning of codes line by line for me? Thank you.

DATA alldata0;

IF \_n\_=0 THEN SET patdata;

IF \_n\_=1 THEN DO;

DECLARE HASH \_h1

(dataset: "PATDATA");

rc=\_h1.definekey("SUBJECT");

rc=\_h1.definedata("TRT\_CODE");

rc=\_h1.definedone();

call missing(SUBJECT,TRT\_CODE);

END;

SET adverse;

rc=\_h1.find();

IF rc^=0 THEN trt\_code=" ";

DROP rc;;

RUN;

DATA \_null\_;

SET sashelp.vtable;

WHERE libname='WORK';

WHERE ALSO memname in('PATDATA','ADVERSE');

CALL SYMPUT('X'||memname,put(nobs,8.));

RUN;

DATA alldata0;

LENGTH trt\_code $1;

ARRAY f{&xpatdata.,2} $6 \_TEMPORARY\_;

DO i=1 TO &xpatdata.;

SET patdata (RENAME=(trt\_code=trt\_code\_dict));

f{i,1}=PUT(subject,6.);

f{i,2}=trt\_code\_dict;

END;

DO i=1 TO &xadverse.;

SET adverse;

trt\_code='';

DO j=1 TO &xpatdata.;

IF subject=INPUT(f(j,1),best.) THEN DO;

trt\_code=f{j,2};

OUTPUT;

END;

IF ^MISSING(trt\_code) THEN LEAVE;

END;

IF MISSING(trt\_code) THEN OUTPUT;

END;

DROP i j trt\_code\_dict;

RUN;