Import Section Quiz Questions

Q.1: When you declare your variables in the INPUT statement, some variables have a $ beside them. What does this indicate?

A: To declare it a numeric variable.

B: To indicate that you don't want to display it in the output.

C: To declare it a character variable.

D: To declare that you want a capital letter for your variable.

Q. 2: Did we use the data step or proc step to import our xlsx file?

A: Data step

B: Proc step

C: Other

D: None of the above

Q. 3: Did we use the data step or proc step to import our .txt file?

A: Data step

B: Proc step

C: Other

D: None of the above

Q. 4: What purpose does the INFILE option FIRSTOBS serve?

A: Tells SAS to begin reading the data from the input SAS dataset at the line number specified

B: Tells SAS to begin the reading the data from the input SAS dataset, and to only read a certain amount (that is, 2 lines) of data

C: Tells SAS to begin reading the data from the input SAS dataset, and skip a certain number of lines that you specify (that is, skip 2 lines)

D: None of the above

Working with Data Quiz Questions

Q. 1: What is the default delimiter for .txt files for list input?

A: A comma

B: Blank space

C: A dot

D: The delete button

Q. 2: If you wanted to add a period (.) as a new delimiter, what line of code of code makes that possible?

A: DLM=.

B: DLM='.'

C: DLM="."

D: DLM=(.)

Q. 3: What do we use the CARDS statement for?

A: It tells SAS to add a random value to our data.

B: It tells SAS to add a random variable to our data.

C: It tells SAS that data lines will follow.

D: It tells SAS that we want to organize our data in a specific manner.

Q. 4: What is one statement that helps you ‘reduce’ your data?

A: reduce statement

B: input statement

C: keep statement

D: null statement

Q. 5: I talked about ‘Cleaning’ your data; in what ways did I suggest that you can clean your data?

A: Renaming variables

B: Putting labels on variables

C: Finding errors

D: The DSD infile option

E: Missover file option

F: Both a and b

G: Both a and d

Q. 6: We talked about free formatted and fixed formatted data, I mentioned in that lesson one advantage of fixed formatted. What was it?

A: It makes thing look tidier

B: You can fit more data in a line

C: It’s easier to locate your data

D: No real benefit

Q. 7: Why are column definitions usually needed for fixed formatted data?

A: They aren't needed usually

B: It makes your data look cleaner and tidier

C: It ensures that your external file reads the data properly

D: Since there are no delimiters

Assignment 0.5 Integration Assignment Instructions

1. Import Balance_Bank.xlsx.

Hint 1a: Make sure to import both sheets.

Hint 1b: Import each sheet individually.

2. Sort both sheets by Acc_number variable.

3. Finally, using the data step, merge the two sheets by Acc_number, create a Balance variable based on Credit minus Debit, and retain the Balance variable.

Hint 1a: You will need to include first. somewhere in your data step.

#3 will require you to do a bit of research if you are not familiar with the first. and last. indicator variables. While I use first. in the lecture before this assignment (counting/enumeration) Variable lecture), I do not explain it in detail.

Question 1:

What is the balance for Acc_number 12346 for 03/01/2018?

Please also provide the code that produced the answer for the question.

Question 2:

How would you explain the role of first. in your code?

Solution:

Question 1: What is the balance for Acc_number 12346 for 03/01/2018?

Please also provide the code that produced the answer for the question.

Instructor answer

```
proc import datafile='/home/ermin0/Balance_Bank.xlsx' out=sheet1
dbms=xlsx;

sheet=sheet1;

run;

proc import datafile='/home/ermin0/Balance_Bank.xlsx' out=sheet2
dbms=xlsx;

sheet=sheet2;

run;

proc contents data=sheet1;

run;

proc sort data=sheet1 out=sheet1_sorted;

by Acc_number;

run;

proc sort data=sheet2 out=sheet2_sorted;
```

```
by Acc_number;
run;
data my_balance;
```

```
merge sheet1_sorted sheet2_sorted;
by Acc_number;
retain Balance;
if first.Acc_number then Balance=Credit-Debit;
else Balance=Balance+Credit-Debit;
run;
```

Question 2:

How would you explain the role of first. in your code?

Instructor answer

When we use the BY statement, we are telling SAS to group by a particular variable. In our case, we told it to group by Acc_number. We also sorted our Acc_number to be in order. Using first.Acc_number, this indicator variable will identify if a particular row/observation should be included into a group. For example, most customers will have many transactions, therefore, their Acc_number will be in many rows/observations to reflect all their transactions.

We need first to tell us if a transaction on a particular row should be grouped with a particular Acc_number. first. will identify if a particular row/observation will be included for a particular customer/Acc_number by assigning = 1 for an observation when an observation is the first observation in each group of values (the first time it sees a particular Acc_number) and assign = 0 when a row/obs is not the first observation in each group of values. (the 2nd, 3rd, 4th… times it sees that

Acc_number). So, when it moves on to a new Acc_number, it once again assigns = 1, then 0 and so it goes on.

Informats/Formats/SAS Input Quiz

Q. 1: Which is an example of standard numeric data?

A: $24,000

B: -34.245

C: 50%

D: ¼

Q. 2: Formatted input can be used to read…

A: both standard and non-standard data in fixed field

B: standard free format data

C: standard data fixed in fields

D: non-standard data in fixes fields

Q. 3: Formatted input can be used to read…

A: both standard and non-standard data in fixed field

B: standard free format data

C: standard data in fixed fields

D: non-standard data in fixed fields

Q. 4: The commaw.d informat can be used to read which of the following? (Hint: As we didn't cover this specifically, this informat strips out special characters!)

A: 15,085

B: $235.57

C: 18%

D: All of the above

Functions Quiz

Q. 1: If you want to ensure that a certain variable only has certain types of characters, what function is super useful?

A: Compress function

B: Coalesce function

C: Substr function

D: Verify function

Q. 2: In cases where there is certain numeric data missing (that is, maybe you have a client's home phone number, but not their cell, or the other way round), what function could help you locate the FIRST number that comes up for them?

A: Substr function
B: Coalesce function
C: Put function

Q. 3: If you want to merge two strings (let's say a last and first name) but ensure that there is some form of delimiter (a comma, dot, and so on) separating the names. What function is useful?

A: Coalesce function

B: Scan function
C: Catx function
D: Compress function
E: None of the above

Q. 4: If you wanted to return the length of a character string, EXCLUDING trailing blanks, which function would you use?

A: LENGTHC
B: LENGTH
C: Neither

Q. 5: If you want to convert a character variable to a numeric one, so you can do analysis on the variable, what function would accomplish this?

A: PUT FUNCTION

B: INPUT FUNCTION

C: INPUT STATEMENT

D: LENGTH STATEMENT

E: COMPRESS FUNCTION

Integration Assignments

The upcoming assignments are meant to help you learn how to integrate

many of the skills you have learned in the course thus far. The solutions for both assignments are available in the lecture following Assignment 1 and 2.

Assignment 1: Integration of Skills 1

1. Import mpi_national.csv

2. Change the order of the ISO and Country variables. Meaning, make country the first column and ISO the second column (note: this does not require you to edit the actual csv file)

3.Allocate the correct number of bytes for all variables.

4.Correctly denote if a variable is character or numeric.

5.Create a new variable that finds the difference between MPI Urban and MPI Rural.

6.I want you to replace the 'NIG' ISO variable value for Nigeria and change it to: 'NGA'.

7.I want the ISO/Country values to be brought together and be separated by a comma. (that is, Serbia, SRB). You can call the variable that stores this 'together'.

8. I only want you to print/display a subset of the data when Intensity of Deprivation Rural is equal to or greater than 40.0.

Assignment 2: Integration of Skills 2

1. Import Current_Employee_Names__Salaries__and_Position_Titles.csv. This file can be found on google drive. Check the first section/chapter of the

course for the link. You can name the dataset assignment2solution.

2. You can give the variables these names (below). However, check the dataset and make sure to declare the variables (that is, numeric or

character).name jobtitles dep full or part salorhour typicalh annuals hourly.

3. Convert annuals and hourly variables to numeric data types.

4. Please verify that you have successfully converted these variables to numeric data types.

5. This question is linked to #3. You should have new variable names for the variables written in question #3 (annuals and hourlyr) as you had to store the conversion of those variables in a new variable). I want you to use these new variable names for this question. If you have checked the dataset, you will notice that most employees earn a salary but there is a significant group that gets paid hourly. The idea here is that each employee gets a salary or gets paid an hourly rate, NOT BOTH. I'd like you to create a variable (you can call it salorhourly) that will return the first not missing value. (that is, it will return an individual's salary or hourly rate).

6. I want you to proc print the dataset, and only print out the employees that earn $50 or more per hour, print out only the name variable and whatever you named your hourly variable when you converted it to numeric from character, and then format that variable so the dollar sign is utilized for the hourly rate.

Solutions for Both Assignments

dataassignment2solution;

lengthname$35 jobtitles$20 dep$20 fullorpart$14 salorhour$10 typicalh

lengthname$35 jobtitles$20 dept$20 fulorpart$14 salorhour$10 typeam
8annuals$20 hourlyr$25;

infile"E:\Workspace\index\Current_Employee_Names__Salaries__and_
Position_Titles.csv"DSDMISSOVER FIRSTOBS=2;

```
input name$ jobtitles$ dep$ fullorpart$ salorhour$ typicalh annuals$
hourlyr$;
annualsnum=input(annuals,comma11.);
hourlyrnum=input(hourlyr,comma9.);
salorhourly=coalesce(annualsnum,hourlyrnum);
run;
proc contents data=assignment2solution;
run;
proc print data=assignment2solution(where=(hourlyrnum=>50));
varname hourlyrnum;
format hourlyrnum dollar11.2;
run;
```

Important Note Before You Start

Welcome to the Case Study portion!

Just one important note.

Please remember to change the first line of code.

Those of you using SAS software need to add this at the top:

libname mydata '/folders/myfolders/';

instead of libname perm 'F:\';

instead of fbname perm "L:\";

Furthermore, **you don't need to put anything before your data names. (that is, perm.MemberConditions).** So, get rid of the perm part in the rest of the code.

Case Study Code

Here is the code I used. If you didn't read the note at the beginning, please remember to change the first line of code.

```
informat DOB ddmmyy10.;

format DOB ddmmyy10.;
```

Those of you using SAS software need to add this at the top:

```
libname mydata '/folders/myfolders/';
```

instead of

```
libname perm 'E:\';
```

Furthermore, you don't need to put anything before your data names. (that is, perm.MemberConditions). So get rid of the perm part in the rest of the code.

```
libname perm 'E:\';data perm.identifyconditions;

infile "C:\Users\ededic84\Documents\WPS Workspaces\Workspace1\sas project (erm)\casestudy.csv" DSD MISSOVER FIRSTOBS=2;

input ID$ SEX$ DOB Pdx Dx_2 Dx_3 Dx_4;

informat DOB ddmmyy10.;
```

run;

```
proc sort data = perm.identifyconditions out =
perm.identifyconditionstwo;

by ID;

run;

proc print data=perm.identifyconditionstwo;

format DOB ddmmyy10.;

run;

data perm.MemberConditions;

set perm.identifyconditionstwo;

by ID;

length Diabetes Depression COPD Asthma

CKD HIV Schizophrenia Hypertension Migraine $14 Conditions $50;

retain Diabetes Depression COPD Asthma CKD HIV Schizophrenia
Hypertension Migraine

Conditions;

if first.ID then

do;

Diabetes ="";
```

Depression ="";

COPD ="";

Asthma ="";

```
CKD="";
HIV="";
Schizophrenia="";
Hypertension="";
Migraine="";
end;
array diag(4) Pdx Dx_2 Dx_3 Dx_4;
do i= 1 to 4;
if diag(i) in: ("25000")
then Diabetes="Diabetes";
if diag(i) in: ("29620")
then Depression="Depression";
if diag(i) in: ("4912")
then COPD="COPD";
if diag(i) in: ("493")
then Asthma="Asthma";
if diag(i) in: ("40300")
then CKD="CKD";
```

then CRD="CRD";

if diag(i) in: ("042")

then HIV="HIV";

```
if diag(i) in: ("3310")
then Schizophrenia="Schizophrenia";
if diag(i) in: ("5723")
then Hypertension="Hypertension";
if diag(i) in: ("34631")
then Migraine="Migraine";
end;
if last.ID then do;
TotConditions = sum(( Diabetes ne ""),( Depression ne ""), (COPD ne
""),
( Asthma ne ""), (CKD ne ""),(HIV ne ""),(Schizophrenia ne ""),
(Hypertension ne ""), (Migraine ne "") );

Conditions = catx(", ", Diabetes, Depression,
COPD, Asthma,CKD,HIV,Schizophrenia,Hypertension,Migraine);
output;
 end;
```

```
keep ID Sex DOB ToTConditions Conditions;

run;
```

```
proc print data=perm.MemberConditions;
format DOB date9.;
run;
```

SAS SQL Assignment 1

Use the following dataset for the question:

```
data statesstats;
length state $ 25 ;
input State Zone$ Unemp Wage Crime Income;
cards;
Alabama S 6.0 10.75 780 27196
Arizona S 6.4 11.17 715 31293
Arkansas S 5.3 9.65 593 25565
California W 8.6 12.44 1078 35331
Colorado W 4.2 12.27 567 37833
Connecticut E 5.6 13.53 456 41097
Delaware E 4.9 13.90 686 35873
Florida S 6.6 9.97 1206 29294
Georgia S 5.2 10.35 723 31467
```

Idaho N 5.6 11.88 282 31536

Illinois N 5.7 12.26 960 35081

Indiana S 4.9 13.56 489 27858

Iowa N 3.7 12.47 326 33079

Kansas S 5.3 12.14 469 28322

Kentucky S 5.4 11.82 463 26595

Louisiana S 8.0 13.13 1062 25676

Maine E 7.4 11.68 126 30316

Maryland E 5.1 13.15 998 39198

Massachusetts E 6.0 12.59 805 40500

Michigan N 5.9 16.13 792 35284

Minnesota N 4.0 12.60 327 33644

Mississippi S 6.6 9.40 434 25400

Missouri N 4.9 11.78 744 30190

Montana N 5.1 12.50 178 27631

Nebraska N 2.9 10.94 339 31794

Nevada W 6.2 11.83 875 35871

New_Hampshire E 4.6 11.73 138 35245

New_Jersey E 6.8 13.38 627 42280

New_Mexico W 6.3 10.14 930 26905

New_York E 6.9 12.19 1074 31899

North_Carolina S 4.4 10.19 679 30114

North_Dakota N 3.9 10.19 82 28278

Ohio N 5.5 14.38 504 31855

Oklahoma S 5.8 11.41 635 26991

Oregon W 5.4 12.31 503 31456

Pennsylvania N 6.2 12.49 418 32066

Rhode_Island E 7.1 10.35 402 31928

South_Carolina S 6.3 9.99 1023 29846

South_Dakota N 3.3 9.19 208 29733

Tennessee S 4.8 10.51 766 28639

Texas S 6.4 11.14 762 30775

Utah W 3.7 11.26 301 35716

Vermont E 4.7 11.54 114 35802

Virginia E 4.9 11.25 372 37647

Washington E 6.4 14.42 515 33533

West_Virginia S 8.9 12.60 208 23564

Wisconsin N 4.7 12.41 264 35388

Wyoming N 5.3 11.81 286 33140;

run;

Question:

I want you to produce output that includes **all columns**, **ordered by Avg**income with the highest incomes on top/first. I also want some **summary statistics (average income)** based on **Zone,** while only showing **AvgIncome** of 30,000 or more.

HINT: It can be done in one step.

Solution

First (the right solution, but SAS must remerge the data):

You will receive a note a**bout the query requiring remerging** if you do the solution as the first one. Remember, this is a NOTE, not an error. There is nothing that you have to do. Nevertheless, it is essential to understand.

The problem is that many columns/variables in the SELECT statement are not involved in the comparing (AvgIncome >) or groupby statement. This means that SAS has to do a second pass-by to retrieve the values of these columns/variables, despite not being involved. This is why remerging may be required.

```
select State,Zone,Unemp,Wage,Crime,Income, Avg (Income)

AS AvgIncome

from statesstats

group by zone

having AvgIncome > 30000

Order by AvgIncome DESC;
```

If we look at the following example, **I've taken out all the columns/variables that are not involved**, and as a result, you should not get this note.

Second (no remerging required):

```
proc sql;
```

```
proc sql;

select Zone, Avg (Income) AS AvgIncome

from statesstats
```

group by Zone

having AvgIncome > 30000

Order by AvgIncome DESC;

The other situation in which you may get this problem is when you use the HAVING clause, and you use a variable/column in a comparison that is not in the group-by.

If we had:

having salary > AvgSalary;

SAS SQL Assignment 2

Use the following datasets to answer the question:

data statesstats;

length state $ 25 ;

input State Zone$ Unemp;

cards;

Alabama S 6.0

Arizona S 6.4

Arkansas S 5.3

California W 8.6

Colorado W 4.2

Connecticut E 5.6

Delaware E 4.9

Florida S 6.6

Georgia S 5.2

Idaho N 5.6

run;

data statesstats2;

length state $ 25 ;

input State Wage Crime Income;

cards;

Alabama 10.75 780 27196

Arizona 11.17 715 31293

Arkansas 9.65 593 25565

California 12.44 1078 35331

Colorado 12.27 567 37833

Connecticut 13.53 456 41097

Delaware 13.90 686 35873

Florida 9.97 1206 29294

Georgia 10.35 723 31467

Idaho 11.88 282 31536

Question:

I'd like you to join ON the State column and ensure that all the columns show as they should. Ensure that State does not show up twice in the output.

Solution

```
proc sql;

select *from statesstats as ss,statesstats2 as ss2

where ss.state=ss2.state;

Quit;
```

or

```
proc sql;

select *from statesstats as ss inner join statesstats2 as ss2

ON ss.state=ss2.state;

Quit;
```

As you will notice, t**he State column is there twice**. A simple solution is this:

```
proc sql;

SELECT statesstats.*, Wage,Crime,Income

FROM statesstats as s inner join statesstats2 as ss

ON s.State=ss.State;
```

Quit;

SAS SQL Assignment 3

1. I want you to **create a table** out of the statesstats dateset and give it the name **finalstats**. I also want you to use Case expression to come up with a rating system based on the **Unemployment column**. (that is, if some state has a particular unemployment rate, they would get a specific numeric rating). You can come up with any rating system, as long as it's based on the **Unemp column**. Name the column **UnemploymentR**.

2. Next, I want you to a**dd a new column**. I want you to call it **OverallCrimeRisk**. A label and format would be nice as well.

3. Finally, I want you to update that Column by adding rows. I want the OverallCrimeRisk column to be based on the UnemploymentR * crime/100.

```
proc sql;


create table finalstats as


SELECT State,Zone,Unemp,Wage,Crime,Income,


CASE


WHEN Unemp BETWEEN 0.1 and 4.0 THEN 1
```

When Unemp BETWEEN 4.1 and 6.0 THEN 2

```
WHEN unemp between 6.1 and 8.2 THEN 3

ELSE 4

end as UnemploymentR

from statesstats;

quit;

proc sql;

alter table finalstats

add OverallCrimeRisk num label='Overall Crime Risk' format=best8.;

proc sql;

update finalstats
```

```
set OverallCrimeRisk=UnemploymentR* Crime/100;


select *
```

from finalstats;

quit;

SAS SQL Assignment 4

Use this data:

Alabama S 6.0 10.75 780 27196

Arizona S 6.4 11.17 715 31293

Arkansas S 5.3 9.65 593 25565

California W 8.6 12.44 1078 35331

Colorado W 4.2 12.27 567 37833

Connecticut E 5.6 13.53 456 41097

Delaware E 4.9 13.90 686 35873

Florida S 6.6 9.97 1206 29294

Georgia S 5.2 10.35 723 31467

Idaho N 5.6 11.88 282 31536

Illinois N 5.7 12.26 960 35081

Indiana S 4.9 13.56 489 27858

Colorado W 4.2 12.27 567 37833

Connecticut E 5.6 13.53 456 41097

Question(s):

1. Use the preceding dataset, which is just a partial table of the original statestats dataset.

2. Find the duplicates.

3. Show only the unique observations (without the duplicates) in the final output.

4. Add summary statistics, the average, minimum, and maximum for the income column.

Solution

```
proc sql;

select *, count(*) as Count

from statesstats

group by State,Zone,Unemp,Wage,Crime,Income
```

```
having count(*) > 1;

quit;
```

```
Proc SQL;

select distinct State,Zone,Unemp,Wage,Crime,Avg(income) as
AvgIncome, Min(Income) as MinIncome, Max(Income) as MaxIncome

from statesstats;

quit;
```

index Assignment

1. Use the following code to import the assignment dataset.

Of course, make sure that you ***first*** download the file (***brooklynsales.csv***) (**it is attached as a resource**) and then import it into whichever software you are using.

PS: Don't forget to c***hange the location of file in the infile statement*** (as the following code is the location of where I have the file on my computer).

```
data brooklysales;

infile "E:\Workspace\index\brooklynsales.csv" DSD MISSOVER
FIRSTOBS=2;
```

```
input borough neighborhood$ buildingclass$ tax_class$ b_class$
address$ zip$ salesp$;
```

run;

2. Answer ***all*** the questions posed. ***Assume*** that it makes sense to make an index in this scenario.

Questions for This Assignment

1. Which *variables* in the dataset would be poor candidates as index variables, and *how* did you confirm this?

ANSWER:

tax_class & buildingclass are not great candidates. You can confirm this by using proc freq and seeing if it's a discriminant variable by looking at percent column in the proc frequency output.

2. Create a *simple index*, *on the zip variable*, using *proc sql* for the brooklynsales *existing* dataset.

ANSWER:

proc sql;

create index zip on brooklynsales;

quit;

quit;

3. How can you *confirm* that you successfully created your simple index from question 2? *Hint: What is the SAS code to find out?*

ANSWER:

proc contents data=brooklynsales;

run;

Look at the output tables that proc contents produces. The bottom table is called “Alphabetic List of Indexes and Attributes” and should list the index you created in question 2.

4. Use the following code and tell me if the simple index you created (*in question 2*) was *utilized?* If it was utilized, how do you know? If it wasn’t utilized, how do you know?

ANSWER:

data useindex;

set brooklynsales;

where zip=11201 or zip=11249;

run;

5. Before you answer this question, here is a little background to help you out.

There are *many reasons* why SAS might *not* utilize your Index, even if you created one. Some reasons are *broad/general*. An example of a *broad/general* reason is *poor selection of the discriminant variable*.

If you make a *poor selection* in terms of the discriminant variable, *it may actually take longer to use the index than not. So SAS decides not to use your Index.*

More *specific reasons* have to do with using *certain statements, expressions, or forms of statements/expressions* that will ensure that your Index is not used. For example, did you know that *you can't* use an if statement if you want to exploit your Index?

Question:

With that in mind, how can you change where expression from question #4 so that your index is utilized?

ANSWER:

Instead of using the ***OR*** operator, use the ***AND operator***. Run the code again and *check the log.* Check the resource that spells out the forms of WHERE expression that you can use.

```
options msglevel=I;

data useindex;

set brooklynsales;

where zip=11201 AND zip=11249;

run;
```