


Total Points: 30/30

Term Grade: A+

```
/*
Name: Orysyia Stus
Due Date: August 22 2016
*/

/*
Exercise 7.1. Consider the following data set, GROUP_SCORE.SAS7BDAT. Use
one-to-one reading to add one
variable, MAXSCORE, that contains the maximum score of the SCORE variable.
*/
data GROUP_SCORE;
    input ID Group $ Score;
datalines;
1 A 6
2 A 9
3 B 2
4 A 8
5 A 8
6 A 9
7 A 6
8 A 6
9 B 1
10 B 2
;
data ex_1;
    set GROUP_SCORE;
    maxscore = max(Score);
run;

/*
Exercise 10.2. Based on the GRADE.SAS7BDAT data set, create three
variables, MATH_POINT, ENGLISH_POINT, and
PE_GRADE, using user-defined formats. The instructions for creating these
three variables are described in
Exercise 2.1 in Chapter 2.
*/
proc format;
    value $MATH
        'A'=4
        'B'=3
        'C'=2
        'D'=1
        'F'=0;
    /*value ENGLISH
        90-high = 4
        80-90 = 3
        70-80 = 2
        60-70 = 1
        low-60 = 0,*/
    value PE
        1 = 'pass'
        0 = 'no pass';
run;
proc format fmtlib;
```



```

run;
data ex_2;
    set grade;
    if MATH = 'A' then MATH_POINT = 4;
    else if MATH = 'B' then MATH_POINT = 3;
    else if MATH = 'C' then MATH_POINT = 2;
    else if MATH = 'D' then MATH_POINT = 1;
    else if MATH = 'F' then MATH_POINT = 0;
    if ENGLISH >= 90 then ENGLISH_POINT = 4;
    else if 80 <= ENGLISH < 90 then ENGLISH_POINT = 3;
    else if 70 <= ENGLISH < 80 then ENGLISH_POINT = 2;
    else if 60 <= ENGLISH < 70 then ENGLISH_POINT = 1;
    else if ENGLISH < 60 AND not missing(ENGLISH) then ENGLISH_POINT =
0;
    if PE = 1 then PE_GRADE = 'pass';
    else if PE = 0 then PE_GRADE = 'no pass';
run;

```



```

/*
Problem 1
The inventory.sas7dbat data set contains the number of boxes of cookies
for each person that is planning
to sell
The sales.sas7dbat data set contains the number of boxes of cookies that
some of the people have already
sold
Create a data set that contains the number of cookies remaining by using
match-merging. Note, you only
need two SORT procedures and one DATA step to solve this problem. The

```

nal data should have three variables: ID, SOLD, and LEFT. The listing output should be look like the one below

```
*/  
proc sort data= inventory out= inventory_sort;  
    by id;  
run;  
proc sort data= sales out= sales_sort;  
    by id;  
run;  
data Problem_1;  
    merge inventory_sort sales_sort;  
    by id;  
    left = ABS(sold-quantity);  
run;
```

/*

Problem 2

You will use two data sets: geocode.sas7bdat and households.sas7bdat.

These data sets were originally

downloaded from the US Census Bureau. The description of these two data sets is listed below:

geocode.sas7bdat

households.sas7bdat

The data set geocode.sas7bdat contains 51 observations and the data set households contains 52 observa-

tions. For this problem, you will need to create one single data set that contains the variables STATE,

ID (in 1- or 2-digits), TOTHOUSE, and UNMARRIED, and only contains observations that occur from

both data sets. Notice that the last two digits from the 9-digit

geography code are the same as the 2-digit

geography codes. When you combine these two data sets, be careful about the variable type. The

rst

ve observations of your

nal data set should look similar to the one below

*/

```
data geocode1;  
    set GEOCODE;  
    geoid = substrn(geoid, max(1,length(geoid)-1),2);  
    geoid1 = input(geoid, 8.);  
    drop geoid;  
    rename geoid1 = geoid;
```

run;

```
data households1;  
    set HOUSEHOLDS;
```

run;

```
data Problem_2;  
    merge geocode1 households1;  
    by geoid;  
    rename geoid = id;  
    if cmiss(of _all_) then delete;
```

run;

/*

Problem 3

You will work with the sbp.sas7bdat dataset. Here are the



rst and last 10 observations of the data set.

This dataset contains SBP (Systolic Blood Pressure) measurements for each patient. Some patients were measured once and some were measured more than once. The description of each variable is described below

The gender of each patient can be identi

ed from the last

eld of the ID variable with `M' for Male or
`F' for Female. The gender

eld can be in either upper or lower cases. Some of the IDs start with an 'x' and some don't. For example, 'x13260M' and '13260M' refer to the same person. Based on this dataset, create a new dataset that contains only one observation for each patient with the following variables

```
*/  
data Problem_3;  
    set SBP;  
    newid = compress(id, 'x, F, M, m, f');  
    drop id;  
    sex = UPCASE(substr(id, max(1, length(id),1)));  
    if SBP > 300 then SBP = .;  
run;  
proc sort data= Problem_3 out=one;  
    by newid descending sbp;  
run;  
proc sort data= one nodupkey;  
    by newid;  
run;  
data Problem3_final;  
    set one;  
    maxsbp = sbp;  
    drop sbp;  
    maxvisit = visit;  
    drop visit;  
run;
```

