STAT 425 and STAT 625- Statistical Software

Dr. William Howell, Professor

Homework #10- Combining SAS Data Sets; Creating and Maintaing Permanent Formats

Directions: Please submit one program file, one output file, and one log file for the entire assignment. Use comment statements to separate your answers. For questions that do not require a SAS program use comment statements. For example:

/*

Question #1d: my answer

Question #2a: my answer

***/**

/*Question #4b: */

--*SAS program*—

/*Question #5*/

Please make sure the log and output file contain only one run. For example, clear the screen for the log and output file and submit your program one last time before you upload your solutions to ***Blackboard***. See lab 1 for the instructions on how to clear your output and log files.

Part I: Combining SAS Data Sets

1. Handling Same-Named Variables and Different Data Types for BY Variables

The data set **orion.web_products2** contains an observation for every product available for sale on Orion Star's wholesale website.

Partial **orion.web_products2** (20 Total Observations)

```
  Product_ID      Price    Name

 120400304333    114.36    Smasher Super Rq Ti 350 Tennis Racket
 120400305288     53.26    Knife
 120400305846    107.74    Big Guy Men's Air Deschutz Viii Shoes
 120400308766     40.96    Big Guy Men's Packable Hiking Shorts
 120400308849     12.23    Wood Box for 6 Balls
```

The data set **orion.web_orders2** contains a list of orders made in a single day from the website. Each observation contains the product ID, the quantity ordered, and the customer's name.

Partial **orion.web_orders2** (43 Total Observations)

```
  Product_ID     Quantity    Name

 120400305288       16       Carglar Aydemir
 120400305288       19       Sanelisiwe Collier
 120400305846       13       Candy Kinsey
 120400305846       13       Cynthia Martinez
 120400305846       10       Rolf Robak
```

The two data sets are sorted by **Product_ID**. **Product_ID** is a numeric variable in **orion.web_products2** and a character variable with a length of 12 in **orion.web_orders2**.

a. Create a new data set, **web_converted**, from the **orion.web_products2** data set. Change the type of **Product_ID** to character. (Use the data set **web_converted** to merge with **orion.web_orders2** in the next step.)

Hint: Use the RENAME= data set option to change **Product_ID** to some other name, such as **nProduct_ID**, the LENGTH statement to declare a new character variable named **Product_ID**, and an assignment statement with a PUT function to explicitly convert the numeric value in **nProduct_ID** into a character value in **Product_ID**.

b. Create ***three*** new data sets:

- A data set named **revenue** contains the product code, the price, the quantity sold, the product name, the customer name and the revenue generated from each sale. **Revenue** is calculated as **Price*Quantity**.

  The **Name** variable in **web_converted** refers to the product name and the **Name** variable in **web_orders2** refers to the customer name. Give each variable an appropriate name in the **revenue** data set.

- A data set named **notsold** contains the product code, price, and product name for each product that was not sold.
- A data set named **invalidcode** contains the product code, quantity, and customer name for each observation in the **web_orders2** data set that does not have a corresponding product code in the **web_products2** data set.

c. Print the three data sets with appropriate titles. The data sets should contain 39, 7, and 4 observations, respectively. (The data sets you create might have different variable names than the ones shown here.)

Partial **work.revenue** (39 Observations)

```
                                                  Revenue from Orders

  Product_ID   Price Product_Name                             Quantity Customer            Revenue

 120400305288  53.26 Knife                                       16    Carglar Aydemir      852.16
 120400305288  53.26 Knife                                       19    Sanelisiwe Collier  1011.94
 120400305846 107.74 Big Guy Men's Air Deschutz Viii Shoes       13    Candy Kinsey        1400.62
 120400305846 107.74 Big Guy Men's Air Deschutz Viii Shoes       13    Cynthia Martinez    1400.62
 120400305846 107.74 Big Guy Men's Air Deschutz Viii Shoes       10    Rolf Robak          1077.40
```

Partial **work.notsold** (7 Observations)

```
                         Products Not Ordered

    Product_ID      Price    Name

   120400304333    114.36    Smasher Super Rq Ti 350 Tennis Racket
   120400308849     12.23    Wood Box for 6 Balls
   120400311211     69.16    Tipee Summer Sleeping Bag
   120400317183    164.82    Smasher Rd Ti 70 Tennis Racket
   120400329978    114.47    Tipee Twin Blue/Orange
```

Listing of **invalidcode** (4 Observations)

```
                   Invalid Orders

      Product_ID     Quantity    Name

     120400311465       13       Thomas Leitmann
     120400312556        7       Robyn Klem
     120400315078       23       Tonie Asmussen
     120400326278       10       Theunis Brazier
```

Part II-Creating and Maintaining Permanent Formats

1. Creating Formats with Inclusive Ranges from a SAS Data Set

The data set **orion.ages** contains three variables: **First_Age**, **Last_Age**, and **Description**.

Partial **orion.ages**

```
            First_
     Obs     Age      Last_Age    Description

      1       15         30       15-30 years
      2       30         45       31-45 years
      3       45         60       46-60 years
      4       60         75       61-75 years
```

a. Create a format from the **orion.ages** data set and store it permanently in the **orion.MyFmts** catalog. Use the appropriate option to view the values in the format.

a. Write a DATA step to create a data set named **sales** that reads the **Employee_ID** and **Birth_Date** variables from the **orion.sales** data set. Create a new variable named **Age** that is the employee's age as of the current date, and another new variable named **Age_Cat** that is the value of the variable **Age** using the AGE format. The YRDIF function with the AGE basis calculates the exact age of the employee with decimal places.

b. Print the first five observations of the **sales** data set to confirm that the new variables were created correctly.

PROC PRINT Output (as of July 18, 2012)

```
                                 Sales Data Set

                                  Birth_
       Obs     Employee_ID          Date     Age      Age_Cat

         1          120102     11AUG1973      38    31-45 years
         2          120103     22JAN1953      59    46-60 years
         3          120121     02AUG1948      63    61-75 years
         4          120122     27JUL1958      53    46-60 years
         5          120123     28SEP1968      43    31-45 years
```

Part III- Supplemental exercises for STAT 625 and Honors credit

1. Creating Formats with Exclusive Ranges from a SAS Data Set

The data set **orion.ages_mod** contains three variables: **First_Age**, **Last_Age**, and **Description**.

Partial **orion.ages_mod**

```
                         First_
                  Obs      Age       Last_Age     Description

                   1        15          30        15-29 years
                   2        30          45        30-44 years
                   3        45          60        45-59 years
                   4        60          75        60-75 years
```

a. Create a format named AGES_MOD from the **orion.ages_mod** data set and store it permanently in the **orion.MyFmts** catalog. Use the appropriate option to view the values in the format.

The value of the **Last_Age** variable is not to be included in the **Description** variable. Use SAS Help or SAS OnlineDoc to investigate the **EEXCL** variable that is required to get the correct results for this exercise.

b. Write a DATA step to create a data set named **sales** that reads the **Employee_ID** and **Birth_Date** variables from the **orion.sales** data set. Create a new variable named **Age** that is the employee's age as of the current date, and another new variable named **Age_Cat** that is the value of the variable **Age** using the AGES_MOD format. One possible solution for calculating **Age** is to use the YRDIF function with the AGE basis.

c. Print the first five observations of the **sales** data set to confirm that the new variables were created correctly.

PROC PRINT Output (as of July 18, 2012)

```
                                    Sales Data Set

                                           Birth_
             Obs      Employee_ID            Date     Age       Age_Cat

               1           120102       11AUG1973      38     30-44 years
               2           120103       22JAN1953      59     45-59 years
               3           120121       02AUG1948      63     60-75 years
               4           120122       27JUL1958      53     45-59 years
               5           120123       28SEP1968      43     30-44 years
```