STAT 425 and STAT 625- Statistical Software

Dr. William Howell, Professor

Homework #4-Reading Raw Data; Manipulating Data

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-Reading SAS data sets

1. Reading a Space-Delimited Raw Data File

a. Write a DATA step to create a new data set named **work.qtrdonation**. Read the space-delimited raw data file, which can be named as follows:

Windows	**"&path\donation.dat"**
UNIX	**"&path/donation.dat"**
z/OS (OS/390)	**"&path..rawdata(donation)"**

Partial Raw Data File

```
120265 . . . 25
120267 15 15 15 15
120269 20 20 20 20
120270 20 10 5 .
120271 20 20 20 20
```

b. Read the following fields from the raw data file:

Name	Type	Length
IDNum	Character	6
Qtr1	Numeric	8
Qtr2	Numeric	8
Qtr3	Numeric	8
Qtr4	Numeric	8

c. Write a PROC PRINT step to create the report below. The results contain 124 observations.

Partial PROC PRINT Output

Obs	IDNum	Qtr1	Qtr2	Qtr3	Qtr4
1	120265	.	.	.	25
2	120267	15	15	15	15
3	120269	20	20	20	20
4	120270	20	10	5	.
5	120271	20	20	20	20

2. Reading a Delimited Raw Data File with Nonstandard Data Values

a. Write a DATA step to create a temporary data set, **prices**. Read the delimited raw data file named as follows:

Windows	**"&path\pricing.dat"**
UNIX	**"&path/ pricing.dat"**
z/OS (OS/390)	**"&path..rawdata(pricing)"**

All data fields are numeric.

Partial Raw Data File

```
210200100009*09JUN2011*31DEC9999*$15.50*$34.70
210200100017*24JAN2011*31DEC9999*$17.80*22.80
210200200023*04JUL2011*31DEC9999*$8.25*$19.80
210200600067*27OCT2011*31DEC9999*$28.90*47.00
210200600085*28AUG2011*31DEC9999*$17.85*$39.40
```

b. Generate the report below. The results should contain 16 observations.

Partial PROC PRINT Output

2011 Pricing

Obs	ProductID	StartDate	EndDate	Cost	Sales Price
1	210200100009	06/09/2011	12/31/9999	15.50	34.70
2	210200100017	01/24/2011	12/31/9999	17.80	22.80
3	210200200023	07/04/2011	12/31/9999	8.25	19.80
4	210200600067	10/27/2011	12/31/9999	28.90	47.00
5	210200600085	08/28/2011	12/31/9999	17.85	39.40

3. Reading a Delimited File with Missing Values

a. Write a DATA step to create a temporary data set, **prices**. Use the asterisk-delimited raw data file, which can be named as follows:

Windows	**"&path\prices.dat"**
UNIX	**"&path/prices.dat"**
z/OS (OS/390)	**"&path..rawdata(prices)"**

Partial Raw Data File

```
210200100009*09JUN2007*31DEC9999*$15.50*$34.70
210200100017*24JAN2007*31DEC9999*$17.80
210200200023*04JUL2007*31DEC9999*$8.25*$19.80
210200600067*27OCT2007*31DEC9999*$28.90
210200600085*28AUG2007*31DEC9999*$17.85*$39.40
```

There might be missing data at the end of some records. Read the following fields from the raw data file:

Name	Type	Length
ProductID	Numeric	8
StartDate	Numeric	8
EndDate	Numeric	8
UnitCostPrice	Numeric	8
UnitSalesPrice	Numeric	8

b. Define labels and formats in the DATA step to create a data set that generates the following output when they are used in the PROC PRINT step. The results should contain 259 observations.

Partial PROC PRINT Output

```
                                   2007 Prices

                                                              Sales
                        Start of      End of     Cost Price   Price per
 Obs    Product ID      Date Range    Date Range   per Unit     Unit

   1    210200100009    06/09/2007    12/31/9999      15.50      34.70
   2    210200100017    01/24/2007    12/31/9999      17.80        .
   3    210200200023    07/04/2007    12/31/9999       8.25      19.80
   4    210200600067    10/27/2007    12/31/9999      28.90        .
   5    210200600085    08/28/2007    12/31/9999      17.85      39.40
```

Part II-Manipulating data

4. Creating New Variables

a. Write a DATA step that reads **orion.customer** to create **work.birthday**.

b. In the DATA step, create three new variables: **Bday2012**, **BdayDOW2012**, and **Age2012**.

- **Bday2012** is the combination of the month of **Birth_Date**, the day of **Birth_Date**, and the constant of **2012** in the MDY function.
- **BdayDOW2012** is the day of the week of **Bday2012**.
- **Age2012** is the age of the customer in 2012. Subtract **Birth_Date** from **Bday2012** and divide the result by 365.25.

c. Include only the following variables in the new data set: **Customer_Name**, **Birth_Date**, **Bday2012**, **BdayDOW2012**, and **Age2012**.

d. Format **Bday2012** to appear in the form 01Jan2012. **Age2012** should be formatted to appear with no decimal places.

e. Write a PROC PRINT step to create the report below. The results should contain 77 observations.

Partial PROC PRINT Output

Obs	Customer_Name	Birth_ Date	Bday2012	Bday DOW2012	Age2012
1	James Kvarniq	27JUN1978	27JUN2012	4	34
2	Sandrina Stephano	09JUL1983	09JUL2012	2	29
3	Cornelia Krahl	27FEB1978	27FEB2012	2	34
4	Karen Ballinger	18OCT1988	18OCT2012	5	24
5	Elke Wallstab	16AUG1978	16AUG2012	5	34

5. Creating Multiple Variables in Conditional Processing

a. Write a DATA step that reads **orion.customer_dim** to create **work.season**.

b. Create two new variables: **Promo** and **Promo2**.

The value of **Promo** is based on the quarter in which the customer was born.

- If the customer was born in the first quarter, then **Promo** is equal to *Winter*.
- If the customer was born in the second quarter, then **Promo** is equal to *Spring*.
- If the customer was born in the third quarter, then **Promo** is equal to *Summer*.
- If the customer was born in the fourth quarter, then **Promo** is equal to *Fall*.

The value of **Promo2** is based on the customer's age.

- For young adults, whose age is between 18 and 25, set **Promo2** equal to *YA*.
- For seniors, aged 65 or older, set **Promo2** equal to *Senior*.

Promo2 should have a missing value for all other customers.

c. The new data set should include only **Customer_FirstName**, **Customer_LastName**, **Customer_BirthDate**, **Customer_Age**, **Promo**, and **Promo2**.

d. Create the report below. The results should include 77 observations.

Partial PROC PRINT Output

Obs	Customer_ FirstName	Customer_ LastName	Customer_ BirthDate	Promo	Customer_ Age	Promo2
1	James	Kvarniq	27JUN1978	Spring	33	
2	Sandrina	Stephano	09JUL1983	Summer	28	
3	Cornelia	Krahl	27FEB1978	Winter	33	
4	Karen	Ballinger	18OCT1988	Fall	23	YA
5	Elke	Wallstab	16AUG1978	Summer	33	
6	David	Black	12APR1973	Spring	38	
7	Markus	Sepke	21JUL1992	Summer	19	YA
8	Ulrich	Heyde	16JAN1943	Winter	68	Senior

6. Creating Variables Unconditionally and Conditionally

a. Write a DATA step that reads **orion.orders** to create **work.ordertype**.

b. Create a new variable, **DayOfWeek**, that is equal to the weekday of **Order_Date**.

c. Create the new variable **Type**, which is equal to the following:

- *Retail Sale* if **Order_Type** is equal to *1*
- *Catalog Sale* if **Order_Type** is equal to *2*
- *Internet Sale* if **Order_Type** is equal to *3*.

d. Create the new variable **SaleAds**, which is equal to the following:

- *Mail* if **Order_Type** is equal to *2*
- *Email* if **Order_Type** is equal to *3*.

e. Do not include **Order_Type**, **Employee_ID**, and **Customer_ID** in the new data set.

f. Create the report below. The results should contain 490 observations.

Partial PROC PRINT Output

Obs	Order_ID	Order_ Date	Delivery_ Date	Type	Sale Ads	Day Of Week
1	1230058123	11JAN2007	11JAN2007	Retail Sale		5
2	1230080101	15JAN2007	19JAN2007	Catalog Sale	Mail	2
3	1230106883	20JAN2007	22JAN2007	Catalog Sale	Mail	7
4	1230147441	28JAN2007	28JAN2007	Retail Sale		1
5	1230315085	27FEB2007	27FEB2007	Retail Sale		3

Part III- Supplemental exercises for STAT 625 and Honors credit

7. Reading a Tab-Delimited Raw Data File

a. Create a temporary data set, **managers2**. Use the tab-delimited raw data file, which can be named as follows:

Windows	**"&path\managers2.dat"**
UNIX	**"&path/ managers2.dat"**
z/OS (OS/390)	**"&path..rawdata(managers2)"**

Raw Data File

```
120102  Tom     Zhou    M       108255  Sales Manager
120103  Wilson  Dawes   M       87975   Sales Manager
120261  Harry   Highpoint       M       243190  Chief Sales Officer
121143  Louis   Favaron M       95090   Senior Sales Manager
121144  Renee   Capachietti     F       83505   Sales Manager
121145  Dennis  Lansberry       M       84260   Sales Manager
```

b. Read the following fields from the raw data file:

Name	Type
ID	Numeric
First	Character
Last	Character
Gender	Character
Salary	Numeric
Title	Character

c. The new data set should contain only **First**, **Last**, and **Title**.

d. Generate the report below. The results should contain six observations.

```
Obs     First     Last          Title

  1     Tom       Zhou          Sales Manager
  2     Wilson    Dawes         Sales Manager
  3     Harry     Highpoint     Chief Sales Officer
  4     Louis     Favaron       Senior Sales Manager
  5     Renee     Capachietti   Sales Manager
  6     Dennis    Lansberry     Sales Manager
```

8. Reading a Delimited File with Missing Values and Embedded Delimiters

a. Write a DATA step to create a temporary data set, **salesmgmt**. Use the raw data file, which can be named as follows:

Windows	**"&path\managers.dat"**

Partial Raw Data File

```
120102/Tom/Zhou/M//Sales Manager/AU/11AUG1969/'06/01/1989'
120103/Wilson/Dawes/M/87975/Sales Manager/AU/22JAN1949/'01/01/1974'
120261/Harry/Highpoint/M/243190//US/21FEB1969/'08/01/1987'
121143/Louis/Favaron/M/95090/Senior Sales Manager/US/26NOV1969/'07/01/1997'
121144/Renee/Capachietti/F/83505/Sales Manager/US/28JUN1964
121145/Dennis/Lansberry/M/84260/Sales Manager/US/22NOV1949/'04/01/1976'
```

b. **ID** is a numeric value. The **salesmgmt** data set should contain only the variables shown in the report below.

c. Write a PROC PRINT step to generate the report below. The results should contain six observations.

PROC PRINT Output

```
                                    Orion Star Managers

 Obs      ID      Last           Title                  HireDate    Salary

  1     120102    Zhou           Sales Manager          01JUN1989         .
  2     120103    Dawes          Sales Manager          01JAN1974     87975
  3     120261    Highpoint                             01AUG1987    243190
  4     121143    Favaron        Senior Sales Manager   01JUL1997     95090
  5     121144    Capachietti    Sales Manager                .      83505
  6     121145    Lansberry      Sales Manager          01APR1976     84260
```

9. Using the CATX and INTCK Functions to Create Variables

a. Write a DATA step that reads **orion.sales** to create **work.employees**.

In the DATA step, create a new variable, **FullName**. This variable is the combination of **First_Name**, a space, and **Last_Name**. Use the CATX function. You can find documentation about CATX in the SAS Help Facility or in the online documentation.

In the DATA step, create a new variable, **Yrs2012**. This variable is the number of years between January 1, 2012, and **Hire_Date**. Use the INTCK function. Documentation about INTCK can be found in the SAS Help Facility or in the online documentation.

b. Format **Hire_Date** to appear in the form 31/01/2012.

c. Give **Yrs2012** a label of **Years of Employment as of 2012**.

d. Create the report shown below. The results should contain 165 observations.

Partial PROC PRINT Output

```
                                      Years of
                                     Employment
Obs    FullName                Hire_Date    as of 2012

  1    Tom Zhou                01/06/1993        19
  2    Wilson Dawes            01/01/1978        34
  3    Irenie Elvish           01/01/1978        34
  4    Christina Ngan          01/07/1982        30
  5    Kimiko Hotstone         01/10/1989        23
```

10. Using WHEN Statements in a SELECT Group to Create Variables Conditionally

a. Write a DATA step that reads **orion.nonsales** to create **work.gifts**.

b. Create two new variables, **Gift1** and **Gift2**. Use a SELECT group with WHEN statements. You can find documentation about the SELECT group with WHEN statements in the SAS Help Facility or in the online documentation.

If **Gender** is equal to *F*, then the listed variables equal the following values:

Gift1 is equal to *Scarf*.

Gift2 is equal to *Pedometer*.

If **Gender** is equal to *M*, then the listed variables equal the following values:

Gift1 is equal to *Gloves*.

Gift2 is equal to *Money Clip*.

If **Gender** is not equal to *F* or *M*, then the listed variables equal the following values:

Gift1 is equal to *Coffee*.

Gift2 is equal to *Calendar*.

c. The new data set should include only **Employee_ID**, **First**, **Last**, **Gender**, **Gift1**, and **Gift2**.

d. Create the report below. The results should contain 235 observations.

Partial PROC PRINT Output

Employee_ID	First	Last	Gender	Gift1	Gift2
120101	Patrick	Lu	M	Gloves	Money Clip
120104	Kareen	Billington	F	Scarf	Pedometer
120105	Liz	Povey	F	Scarf	Pedometer
120106	John	Hornsey	M	Gloves	Money Clip
120107	Sherie	Sheedy	F	Scarf	Pedometer