STAT 425 and STAT 625- Statistical Software

Dr. Monica Jackson, Instructor

Homework #5-SOLUTIONS

Part I

1. Concatenating Data Sets with Variables of Different Lengths and Types

a. Open **p110e03**. Submit the PROC CONTENTS steps or explore the data sets interactively to complete the table below. Fill in attribute information for each variable in each data set.

	Code		Company		ContactType	
	Type	**Length**	**Type**	**Length**	**Type**	**Length**
orion.charities	Char	6	Char	40	Char	10
orion.us_suppliers	Char	6	Char	30	Char	1
orion.consultants	Char	6	Char	30	Num	8

```
proc contents data=orion.charities;
run;
proc contents data=orion.us_suppliers;
run;
proc contents data=orion.consultants;
run;

data work.contacts;
   set orion.charities orion.us_suppliers;
run;
proc contents data=work.contacts;
run;

data work.contacts2;
   set orion.us_suppliers orion.charities;
run;
proc contents data=work.contacts2;
run;

data work.contacts3;
   set orion.us_suppliers orion.consultants;
run;
```

c. Submit a PROC CONTENTS step to examine **work.contacts**. From which input data set were the variable attributes assigned? **the first data set in the set statement, orion.charities**

e. Submit a PROC CONTENTS step to examine **work.contacts2**. From which input data set were the variable attributes assigned? **the first data set in the set statement, orion.us_suppliers**

f. Write a DATA step to concatenate **orion.us_suppliers** and **orion.consultants**, and creating a temporary data set, **contacts3**.

Why did the DATA step fail? **ContactType was defined as both character and numeric.**

2. **Merging a Sorted Data Set and an Unsorted Data Set in a One-to-Many Merge**

```
proc sort data=orion.product_list
          out=work.product_list;
   by Product_Level;
run;

data work.listlevel;
   merge orion.product_level work.product_list ;
   by Product_Level;
   keep Product_ID Product_Name Product_Level Product_Level_Name;
run;

proc print data=work.listlevel noobs;
   where Product_Level=3;
run;
```

3. **Merging Using the IN= and RENAME= Options**

```
proc sort data=orion.customer
          out=work.customer;
   by Country;
run;

data work.allcustomer;
   merge work.customer(in=Cust)
         orion.lookup_country(rename=(Start=Country
                                      Label=Country_Name) in=Ctry);
   by Country;
   keep Customer_ID Country Customer_Name Country_Name;
   if Cust=1 and Ctry=1;
run;

proc print data=work.allcustomer;
run;
```

Part II

1. **Producing Frequency Reports with PROC FREQ**

```
proc format;
   value ordertypes
         1='Retail'
         2='Catalog'
         3='Internet';
run;

title 'Order Summary by Year and Type';
proc freq data=orion.orders;
   tables Order_Date;
   tables Order_Type / nocum;
   tables Order_Date*Order_Type / nopercent norow nocol;
   format Order_Date year4. Order_Type ordertypes.;
run;
title;
```

2. **Validating orion.qtr2_2011 with PROC FREQ**

```
proc freq data=orion.qtr2_2011 nlevels;
   tables Order_ID Order_Type;
run;
```

What invalid data exists for **Order_ID** and **Order_Type**?

- **two observations with missing values for Order_ID**
- **one observation with a value of *0* for Order_Type**
- **one observation with a value of *4* for Order_Type**

3. **Analyzing Missing Numeric Values with PROC MEANS**

```
title 'Number of Missing and Non-Missing Date Values';
proc means data=orion.staff nmiss n nonobs;
   var Birth_Date Emp_Hire_Date Emp_Term_Date;
   class Gender;
run;
title;
```

4. **Validating orion.shoes_tracker with the UNIVARIATE Procedure**

```
proc univariate data=orion.shoes_tracker;
   var Product_ID;
run;
```

How many values of **Product_ID** are too small? **one (*2.20200E+10*)**

How many values of **Product_ID** are too large? **one (*2.2020E+12*)**

5. **Directing Output to EXCELXP Destination**

```
ods tagsets.excelxp file="&path\p111e13.xls" style=Listing;

title 'Customer Type Definitions';
proc print data=orion.customer_type;
run;

title 'Country Definitions';
proc print data=orion.country;
run;

ods tagsets.excelxp close;
```

Part III

1. **Merging and Creating Output in Multiple Data Sets**

```
proc sort data=orion.orders
          out=work.orders;
   by Employee_ID;
run;

data work.allorders work.noorders;
   merge orion.staff(in=Staff) work.orders(in=Ord);
   by Employee_ID;
   if Ord=1 then output work.allorders;
   else if Staff=1 and Ord=0 then output work.noorders;
  /* alternate statement */
  /* else output work.noorders; */
   keep Employee_ID Job_Title Gender Order_ID Order_Type Order_Date;
run;

proc print data=work.allorders;
run;

proc print data=work.noorders;
run;
```

2. **Creating an Output Data Set with PROC FREQ**

```
proc freq data=orion.order_fact noprint;
   tables Product_ID / out=product_orders;
run;

data product_names;
   merge product_orders orion.product_list;
   by Product_ID;
   keep Product_ID Product_Name Count;
run;

proc sort data=product_names;
   by descending Count;
run;

title 'Top Five Products by Number of Orders';
proc print data=product_names(obs=5) label noobs;
   var Count Product_ID Product_Name;
   label Product_ID='Product Number'
         Product_Name='Product'
         Count='Orders';
run;
title;
```

3. **Adding Options to the EXCELXP Destination**

```
ods tagsets.excelxp file="&path\p111e14.xls"
                    style=Listing
                    options(doc='help'
                            embedded_titles='yes'
                            sheet_name='First Report');

title 'Customer Type Definitions';
proc print data=orion.customer_type;
run;

ods tagsets.excelxp options(sheet_name='Second Report');

title 'Country Definitions';
proc print data=orion.country;
run;

ods tagsets.excelxp close;
```