**Deeza-Mae Smith  
Math 510 – Homework 7**

1. Exercises: Chapter 3.2 Levels 1, 2, and Challenge

**Level 1: Accessing a permanent data set**

**Number of observations in orion.country:** 7 (rows)

**Number of variables in orion.country:** 6 (columns)

**Name of the last country:** South Africa (ZA)

b) **Proc Contents Step:**  
proc contents data=orion.\_all\_nods;  
run;

**Name of the last member listed:** US\_SUPPLIERS (#37)

**Level 2: Viewing General Data Set Propoerties**

a) **Examine general data set properties:** Right click Staff and select Properties or run code:  
proc contents data=orion.staff;  
run;

b) **Sort information stored for orion.staff:** The ORION.STAFF data is not sorted

**The CONTENTS Procedure**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Set Name** | ORION.STAFF | **Observations** | 424 |
| **Member Type** | DATA | **Variables** | 10 |
| **Engine** | V9 | **Indexes** | 2 |
| **Created** | 11/17/2015 16:07:26 | **Integrity Constraints** | 1 |
| **Last Modified** | 11/17/2015 16:25:17 | **Observation Length** | 96 |
| **Protection** |  | **Deleted Observations** | 0 |
| **Data Set Type** |  | **Compressed** | NO |
| **Label** |  | **Sorted** | NO |
| **Data Representation** | SOLARIS\_X86\_64, LINUX\_X86\_64, ALPHA\_TRU64, LINUX\_IA64 |  |  |
| **Encoding** | utf-8 Unicode (UTF-8) |  |  |

**Challenge: SAS Autoexec File**

**Name of file:** AUTOEXEC.SAS

**Purpose of file:** The autoexec file contains SAS statements that are executed automatically when you invoke SAS or when you start another SAS process. The autoexec file can contain any valid SAS statements.

**How it is created:** We can use a SAS text editor to create your autoexec file. The recommended method is to create the file by using a SAS text editor (such as the Enhanced Editor window) and save it using the Save As dialog box.

**How it could be useful:** The autoexec file is a convenient way to execute a standard set of SAS program statements each time that you invoke SAS. You can include OPTIONS, LIBNAME, or FILENAME statements, or any other SAS statements and system options that you want the system to execute each time you invoke a SAS session.

3. Exercises: Chapter 4.1,2, and 3 Levels 1 and 2 (pages 4-25,26,38,39,48,49)

Chapter 4.1

**Level 1: Displaying orion.order\_fact with the PRINT procedure**

b) **Add sum statement to PRINT procedure:**  
proc print data=orion.order\_fact;  
sum Total\_Retail\_Price;  
run;

Total sum is $100,077.46

c) **Add WHERE statement to select Total\_Retail\_Price observations greater than 500:**proc print data=orion.order\_fact;  
sum Total\_Retail\_Price;  
where Total\_Retail\_Price>500;  
run;

The Obs column looks like it is sorted.

Yes, the sum changed and it is now $32,696.60

d) **Add option to suppress Obs column:**

proc print data=orion.order\_fact noobs;  
sum Total\_Retail\_Price;  
where Total\_Retail\_Price>500;  
run;

We can verify the number of observations by looking at the Log.

e) **Add an ID statement:**   
proc print data=orion.order\_fact noobs;  
sum Total\_Retail\_Price;  
where Total\_Retail\_Price>500;  
id Customer\_ID;  
run;

Output changed by Customer ID being the first column

f) **Add a VAR statement:**  
proc print data=orion.order\_fact noobs;  
sum Total\_Retail\_Price;  
where Total\_Retail\_Price>500;  
id Customer\_ID;  
var Customer\_ID Order\_ID Order\_Type Quantity Total\_Retail\_Price;  
run;

We notice that the column Customer\_ID appears twice in the output

g) **Modified VAR statement:** remove Customer\_ID from VAR statement  
proc print data=orion.order\_fact noobs;  
sum Total\_Retail\_Price;  
where Total\_Retail\_Price>500;  
id Customer\_ID;  
var Order\_ID Order\_Type Quantity Total\_Retail\_Price;  
run;

Level 2: Displaying orion.customer\_dim with the PRINT procedure

a) **PRINT step to display:**   
proc print data=orion.customer\_dim;  
run;

b) **Modified program to select customers aged 30-40 and remove Obs:**  
proc print data=orion.customer\_dim noobs;  
where Customer\_Age between 30 and 40;  
run;

c) **Add statement to use Customer\_ID as identifying column:**  
proc print data=orion.customer\_dim noobs;  
where Customer\_Age between 30 and 40;  
id Customer\_ID;  
run;

d) **Add statement to limit variables:**  
proc print data=orion.customer\_dim noobs;  
where Customer\_Age between 30 and 40;  
id Customer\_ID;  
var Customer\_Name Customer\_Age Customer\_Type;  
run;

Chapter 4.2

**Level 1:**

**5. Sorting orion.employee\_payroll and displaying the new data set**

a) **Sort orion.employee\_payroll by Salary and place into temporary data set:**   
proc sort data=orion.employee\_payroll out=work.sort\_salary;  
by Salary;  
proc print data=orion.employee\_payroll;  
run;

b) **Modified PROC PRINT step:**  
proc print data=work.sort\_salary;  
run;

**6.** **Sorting orion.employee\_payroll and displaying group observations**

a) Sort by gender then descending salary:  
proc sort data=orion.employee\_payroll out=work.sort\_salary2;  
by Employee\_Gender descending Salary;  
proc print data=orion.employee\_payroll;  
run;

b) Modified PROC PRINT step:  
proc print data= work.sort\_salary2;  
run;

**Level 2**

**7. Sorting orion.employee\_payroll and displaying a subset of the new data set**

a) **Sort by Employee\_Gender and then descending Salary**:  
proc sort data=orion.employee\_payroll out=work.sort\_sal;  
by Employee\_Gender descending Salary;  
proc print data=orion.employee\_payroll;  
run;

b) **Print subset of data set:**  
proc print data=orion.employee\_payroll noobs;  
by Employee\_Gender;  
sum Salary;  
where Employee\_Term\_Date is missing and Salary>65000;  
var Employee\_ID Salary Marital\_Status;  
run;

Chapter 4.3

**Level 1:**

**9. Displaying Titles and Footnotes in a Detail Report**

a) Display all observations for Australian Sales Reps:  
proc print data=orion.sales noobs;  
where Country='AU' and Job\_Title contains 'Rep. IV';  
run;

b) **Adding VAR statement:**  
proc print data=orion.sales noobs;  
where Country='AU' and Job\_Title contains 'Rep. IV';  
var Employee\_ID First\_Name Last\_Name Gender Salary;  
run;

c) **Add TITLE and FOOTNOTE statements:**title1'Australian Sales Employees';  
title2'Senior Sales Representatives';  
footnote1'Job\_Title: Sales Rep. IV';  
proc print data=orion.sales noobs;  
where Country='AU' and Job\_Title contains 'Rep. IV';  
var Employee\_ID First\_Name Last\_Name Gender Salary;  
run;  
title;  
footnote;

e) **Clear title and footnote statements:**  
title1;  
title2;  
footnote1;  
proc print data=orion.sales noobs;  
where Country='AU' and Job\_Title contains 'Rep. IV';  
var Employee\_ID First\_Name Last\_Name Gender Salary;  
run;  
title;  
footnote;

**10. Display Column Headings in a Detail Report**

a) **Modify using labels:**title 'Entry-level Sales Representatives';  
footnote 'Job\_Title: Sales Rep. I';

proc print data=orion.sales noobs label;

where Country='US' and Job\_Title='Sales Rep. I';  
var Employee\_ID First\_Name Last\_Name Gender Salary;  
label Employee\_ID='Employee ID'  
First\_Name='First Name'  
Last\_Name='Last Name'  
Salary='Annual Salary';

run;

title;  
footnote;

b) **Modify program to use blank space as SPLIT character:**  
title 'Entry-level Sales Representatives';  
footnote 'Job\_Title: Sales Rep. I';

proc print data=orion.sales noobs split=’ ‘;

where Country='US' and Job\_Title='Sales Rep. I';  
var Employee\_ID First\_Name Last\_Name Gender Salary;  
label Employee\_ID='Employee ID'  
First\_Name='First Name'  
Last\_Name='Last Name'  
Salary='Annual Salary';

run;

title;  
footnote;

**Level 2:**

**11. Writing an Enhanced Detail Report**

a) **Write program to display subset:**  
proc sort data=orion.employee\_addresses out=work.addresses;  
where Country='US';  
by State City Employee\_Name;  
run;

title 'US Employees by State';  
proc print data= work.addresses noobs split=' ';  
var Employee\_ID Employee\_Name City Postal\_Code;  
label Employee\_ID= 'Employee ID'   
Employee\_Name= 'Name'  
Postal\_Code= 'Zip Code';  
by State;  
run;

title;

5. Exercise: Chapter 5.1,2 Levels 1 and 2 (pages 5-10,11,24,25)

**Level 1**

**1. Displaying Formatted Values in a Detail Report**

a) **PROC PRINT of data set:**proc print data=orion.employee\_payroll;  
run;

b) **Modify PROC PRINT:**proc print data=orion.employee\_payroll;  
var Employee\_ID Salary Birth\_Date Employee\_Hire\_Date;  
run;

c) **Adding FORMAT statement**:  
proc print data=orion.employee\_payroll;  
format Salary dollar12.2 Birth\_Date mmddyy10. Employee\_Hire\_Date Date9.;  
var Employee\_ID Salary Birth\_Date Employee\_Hire\_Date;  
run;

**Level 2**

**2. Displaying Formatted Values in a Detail Report**

a) **Creating report:**title1 'US Sales Employees';  
title2 'Earning Under $26,000';  
proc print data=orion.sales noobs split=' ';  
label First\_Name='First Name' Last\_Name='Last Name' Hire\_Date='Date Hired';  
format Salary dollar12.2 Hire\_Date monyy7.;  
var Employee\_ID First\_Name Last\_Name Job\_Title Salary Hire\_Date;  
where Country= 'US' and Salary<26000;  
run;  
title;

Chapter 5.2

**Level 1**

**4. Creating User-Defined Formats**

b) **Creating character format named $GENDER:**  
proc format;  
value $gender 'F'='Female' 'M'='Male';  
run;

c) **Creating numeric format named MNAME:**  
proc format;  
value $gender 'F'='Female' 'M'='Male';  
value mname 1='January' 2='February' 3='March';  
run;

d) **Adding PROC PRINT step with user-defined formats:**  
proc print data=Q1Birthdays;  
format Employee\_Gender $gender.  
BirthMonth mname.;  
var Employee\_ID Employee\_Gender BirthMonth;  
run;

e) **Creating report:**data Q1Birthdays;

set orion.employee\_payroll;  
BirthMonth=month(Birth\_Date);  
if BirthMonth le 3;

run;

proc format;  
value $gender 'F'='Female' 'M'='Male';  
value mname 1='January' 2='February' 3='March';  
run;

title 'Employees with Birthdays in Q1';  
proc print data=Q1Birthdays;  
format Employee\_Gender $gender.  
BirthMonth mname.;  
var Employee\_ID Employee\_Gender BirthMonth;  
run;

**Level 2**

**5. Defining Ranges in User Defined Formats**

proc format;  
value $gender 'F'='Female' 'M'='Male' other='Invalid Code';  
value salrange 20000-<100000='Below $100,000'   
100000-500000='$100,000 or more'   
.='Missing salary'   
other='Invalid salary';  
run;

proc print data=orion.nonsales;

var Employee\_ID Job\_Title Salary Gender;  
title1 'Salary and Gender Values';  
 title2 'for Non-Sales Employees';  
 format Gender $gender.  
Salary salrange.;

run;

Exercises: Chapter 6.2, Level 2

**Level 2: Subsetting Observations based on Three Conditions**

**All steps:**  
data work.delays;  
set orion.orders;  
Order\_Month=month(Order\_Date);  
where Delivery\_Date>Order\_Date+4 and Employee\_ID=99999999;  
if Order\_Month=8;  
keep Employee\_ID Customer\_ID Order\_Date Delivery\_Date Order\_Month;  
label Order\_Date='Date Ordered'  
Delivery\_Date='Date Delivered'  
Order\_Month='Month Ordered';  
format Order\_Date Delivery\_Date mmddyy10.;  
run;

proc contents data=work.delays;  
run;

proc print data=work.delays;  
run;

9. Exercises: Chapter 9.1 Level 2

**Level 2: Creating New Variables**

**All steps:**data work.birthday;  
set orion.customer;  
Bday2012=mdy(month(Birth\_Date),day(Birth\_Date),2012);  
BdayDOW2012=weekday(Bday2012);  
Age2012=(Bday2012-Birth\_Date)/365.25;  
keep Customer\_Name Birth\_Date Bday2012 BdayDOW2012 Age2012;  
format Bday2012 date9.  
Age2012 3.;  
run;

proc print data=work.birthday;  
run;

10. Exercises: Chapter 9.2 Level 2

**Level 2**

**6. Creating Multiple Variables in Conditional Processing**

**All steps:**data work.season;  
set orion.customer\_dim;  
if qtr(Customer\_BirthDate)=1 then Promo='Winter';  
else if qtr(Customer\_BirthDate)=2 then Promo='Spring';  
else if qtr(Customer\_BirthDate)=3 then Promo='Summer';  
else if qtr(Customer\_BirthDate)=4 then Promo='Fall';  
if Customer\_Age>=18 and Customer\_Age<=25 then Promo2='YA';  
else if Customer\_Age>=65 then Promo2='Senior';  
keep Customer\_FirstName Customer\_LastName Customer\_BirthDate Customer\_Age Promo Promo2;  
run;

proc print data=work.season;  
run;

**7. Creating Variables Unconditionally and Conditionally**

**All steps:**data work.ordertype;  
set orion.orders;  
DayOfWeek=weekday(Order\_Date);  
if Order\_Type=1 then Type='Retail Sale';  
else if Order\_Type=2 then do;  
Type='Catalog Sale';SaleAds='Mail';  
end;  
else if Order\_Type=3 then do;  
Type='Internet Sale'; SaleAds='Email';  
end;  
drop Order\_Type Employee\_ID Customer\_ID;  
run;

proc print data=work.ordertype;  
run;

12. Exercises: Chapter 10.1 Level 2

**Level 2: Concatenating Data Sets with Variables of Different Lengths and Types**

a) **Use PROC CONTENTS to fill in table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 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 |

b) **Write data step:**  
data contacts;  
set orion.charities orion.us\_suppliers;  
run;

c) **Run PROC CONTENTS**:  
proc contents data=work.contacts;  
run;

Variable attributes were assigned to orion.charities

d) **Concatenate data step:**data contacts2;  
set orion.us\_suppliers orion.charities;  
run;

e) **PROC CONTENTS step:**  
proc contents data=work.contacts2;  
run;

Variable attributes were assigned to orion.us\_suppliers.

f) **Code:**data contacts3;  
set orion.us\_suppliers orion.consultants;  
run;

proc contents data=work.contacts3;  
run;

The data step failed because orion.consultants is numeric for ContactType while orion.us\_suppliers is a character for ContactType.

13. Exercises: Chapter 10.3 Level 2

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

**All steps:**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;

14. Exercises: Chapter 10.4 Level 2

**Level 2: Merging Using the IN= and RENAME= Options**

**All steps:**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;  
run;

proc print data=work.allcustomer;  
run;