Homework 10-SOLUTIONS

Part I

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

```
data web_converted(drop=nProduct_ID);
   length Product_ID $ 12;
   set orion.web_products2(rename=(Product_ID=nProduct_ID));
   Product_ID=put(nProduct_ID,12.);
run;

data revenue
   NotSold(keep=Price Product_ID Product_Name)
   InValidCode(Keep=Product_ID Quantity Customer);
   merge web_converted(in=InConv rename=(Name=Product_Name))
         orion.web_orders2(in=InOrders rename=(Name=Customer));
   by Product_ID;
   if InConv and InOrders then do;
      Revenue=Quantity * Price;
      output revenue;
   end;
   else if InConv and not InOrders then output notsold;
   else if not InConv and InOrders then output invalidcode;
run;

title 'Revenue from Orders';
proc print data=revenue noobs;
run;

title 'Products Not Ordered';
proc print data=notsold noobs;
run;

title 'Invalid Orders';
proc print data=invalidcode noobs;
run;
title;
```

Part II.

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

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.

p210s02

```
data ages;
   set orion.ages (rename=(First_Age=Start Last_Age=End
                             Description=Label));
   retain FmtName 'ages';
run;

proc format library=orion.MyFmts fmtlib cntlin=ages;
   select ages;
run;
```

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 AGE format.

p210s02

```
data sales;
   set orion.sales(keep=Employee_ID Birth_Date);
   Age=int(yrdif(Birth_Date, today(), 'AGE'));
   Age_Cat=put(Age, ages.);
run;
```

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

p210s02

```
proc print data=sales(obs=5);
   format Birth_Date date9.;
   title 'Sales Data Set';
run;
```

Part III

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

   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.

   p210s03

```
data ages_mod;
   set orion.ages_mod(rename=(First_Age=Start Last_Age=End
                                Description=Label));
   retain fmtname 'ages_mod';
   EEXCL='Y';
run;

proc format library=orion.MyFmts fmtlib cntlin=ages_mod;
   select ages_mod;
run;
```

   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.

   p210s03

```
options fmtsearch=(orion.MyFmts);

data sales;
   set orion.sales(keep=Employee_ID Birth_Date);
   Age=int(yrdif(Birth_Date, today(), 'AGE'));
   Age_Cat=put(Age, ages_mod.);
run;
```

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

   p210s03

```
proc print data=sales(obs=5);
   format birth_date date9.;
   title 'Sales Data Set';
run;
```