Unit 1-9 Exercises

1. Creating Two New Variables

a. Retrieve the starter program **p109e01**.

b. In the DATA step, create two new variables, `Increase` and `NewSalary`.

- `Increase` is the `Salary` multiplied by 0.10.
- `NewSalary` is `Salary` added with `Increase`.

c. Include only the following variables: `Employee_ID`, `Salary`, `Increase`, and `NewSalary`.

d. Store formats displaying commas for `Salary`, `Increase`, and `NewSalary`.

e. Submit the program to create the following PROC PRINT report:

Partial PROC PRINT Output (First 10 of 424 Observations)

```
                                 Employee
                                  Annual
         Obs    Employee_ID       Salary     Increase    NewSalary

           1       120101        163,040       16,304      179,344
           2       120102        108,255       10,826      119,081
           3       120103         87,975        8,798       96,773
           4       120104         46,230        4,623       50,853
           5       120105         27,110        2,711       29,821
           6       120106         26,960        2,696       29,656
           7       120107         30,475        3,048       33,523
           8       120108         27,660        2,766       30,426
           9       120109         26,495        2,650       29,145
          10       120110         28,615        2,862       31,477
```

2. Creating Three New Variables

a. Write a DATA step to read `orion.customer` to create `Work.birthday`.

b. In the DATA step, create three new variables, `Bday2009`, `BdayDOW2009`, and `Age2009`.

- `Bday2009` is the combination of the month of `Birth_Date`, the day of `Birth_Date`, and the constant of `2009` in the MDY function.
- `BdayDOW2009` is the day of the week of `Bday2009`.
- `Age2009` is the age of the customer in 2009. Subtract `Birth_Date` from `Bday2009` and then divide by `365.25`.

c. Include only the following variables: `Customer_Name`, `Birth_Date`, `Bday2009`, `BdayDOW2009`, and `Age2009`.

d. Format `Bday2009` to resemble a two-digit day, a three-letter month, and a four-digit year. `Age2009` should be formatted to appear with no digits after the decimal point.

e. Write a PROC PRINT step to create the following report:

Partial PROC PRINT Output (First 10 of 77 Observations)

```
                                  Birth_                     Bday
   Obs    Customer_Name             Date      Bday2009    DOW2009    Age2009

     1    James Kvarniq          27JUN1974    27JUN2009       7         35
     2    Sandrina Stephano      09JUL1979    09JUL2009       5         30
     3    Cornelia Krahl         27FEB1974    27FEB2009       6         35
     4    Karen Ballinger        18OCT1984    18OCT2009       1         25
     5    Elke Wallstab          16AUG1974    16AUG2009       1         35
     6    David Black            12APR1969    12APR2009       1         40
     7    Markus Sepke           21JUL1988    21JUL2009       3         21
     8    Ulrich Heyde           16JAN1939    16JAN2009       6         70
     9    Jimmie Evans           17AUG1954    17AUG2009       2         55
    10    Tonie Asmussen         02FEB1954    02FEB2009       2         55
```

3. Using the CATX and INTCK Functions to Create Variables

a. Write a DATA step to read **`orion.sales`** to create **`Work.employees`**.

b. In the DATA step, create the new variable **`FullName`**, which is the combination of **`First_Name`**, a space, and **`Last_Name`**. Use the CATX function.

Documentation on the CATX function can be found in the SAS Help and Documentation from the Contents tab (**SAS Products** ⇨ **Base SAS** ⇨ **SAS 9.3 Language Reference: Dictionary** ⇨ **Dictionary of Language Elements** ⇨ **Functions and CALL Routines** ⇨ **CATX Function**).

c. In the DATA step, create the new variable **`Yrs2012`**, which is the number of years between January 1, 2012, and **`Hire_Date`**. Use the INTCK function.

Documentation on the INTCK function can be found in the SAS Help and Documentation from the Contents tab (**SAS Products** ⇨ **Base SAS** ⇨ **SAS 9.3 Language Reference: Dictionary** ⇨ **Dictionary of Language Elements** ⇨ **Functions and CALL Routines** ⇨ **INTCK Function**).

d. Format **`Hire_Date`** to resemble a two-digit day, a two-digit month, and a four-digit year.

e. Give **`Yrs2012`** a label of **`Years of Employment in 2012`**.

f. Write a PROC PRINT step with a VAR statement to create the following report:

Partial PROC PRINT Output (First 10 of 165 Observations)

```
                                                    Years of
                                                   Employment
                Obs    FullName             Hire_Date     in 2012

                  1    Tom Zhou            01/06/1989        23
                  2    Wilson Dawes        01/01/1974        38
                  3    Irenie Elvish       01/01/1974        38
                  4    Christina Ngan      01/07/1978        34
                  5    Kimiko Hotstone     01/10/1985        27
                  6    Lucian Daymond      01/03/1979        33
                  7    Fong Hofmeister     01/03/1979        33
                  8    Satyakam Denny      01/08/2006         6
                  9    Sharryn Clarkson    01/11/1998        14
                 10    Monica Kletschkus   01/11/2006         6
```

4. **Creating Variables Conditionally**

a. Retrieve the starter program **p109e04**.

b. In the DATA step, create three new variables, **`Discount`**, **`DiscountType`**, and **`Region`**.

If **`Country`** is equal to `CA` or `US`,

- **`Discount`** is equal to `0.10`
- **`DiscountType`** is equal to `Required`
- **`Region`** is equal to `North America`.

If **`Country`** is equal to any other value,

- **`Discount`** is equal to `0.05`
- **`DiscountType`** is equal to `Optional`
- **`Region`** is equal to `Not North America`.

c. Include only the following variables: **`Supplier_Name`**, **`Country`**, **`Discount`**, **`DiscountType`**, and **`Region`**.

d. Submit the program to create the following PROC PRINT report:

Partial PROC PRINT Output (First 10 of 52 Observations)

```
                                                                          Discount
 Obs    Supplier_Name                Country        Region          Discount    Type

   1    Scandinavian Clothing A/S      NO      Not North America      0.05     Optional
   2    Petterson AB                   SE      Not North America      0.05     Optional
   3    Prime Sports Ltd               GB      Not North America      0.05     Optional
   4    Top Sports                     DK      Not North America      0.05     Optional
   5    AllSeasons Outdoor Clothing    US      North America          0.10     Required
   6    Sportico                       ES      Not North America      0.05     Optional
   7    British Sports Ltd             GB      Not North America      0.05     Optional
   8    Eclipse Inc                    US      North America          0.10     Required
   9    Magnifico Sports               PT      Not North America      0.05     Optional
  10    Pro Sportswear Inc             US      North America          0.10     Required
```

5. **Creating Variables Unconditionally and Conditionally**

   a. Write a DATA step to read **`orion.orders`** to create **`Work.ordertype`**.

   b. Create the new variable **`DayOfWeek`**, which is equal to the week day of **`Order_Date`**.

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

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

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

   - `Mail` if **`Order_Type`** is equal to `1`
   - `Email` if **`Order_Type`** is equal to `2`.

   e. Do not include **`Order_Type`, `Employee_ID`**, and **`Customer_ID`**.

   f. Write a PROC PRINT step to create the following report:

   Partial PROC PRINT Output (First 20 of 490 Observations)

```
                                                                         Day
                             Order_     Delivery_                 Sale    Of
 Obs        Order_ID           Date       Date          Type        Ads    Week

   1      1230058123    11JAN2003    11JAN2003    Catalog Sale     Mail      7
   2      1230080101    15JAN2003    19JAN2003    Internet Sale    Email     4
   3      1230106883    20JAN2003    22JAN2003    Internet Sale    Email     2
   4      1230147441    28JAN2003    28JAN2003    Catalog Sale     Mail      3
   5      1230315085    27FEB2003    27FEB2003    Catalog Sale     Mail      5
   6      1230333319    02MAR2003    03MAR2003    Internet Sale    Email     1
   7      1230338566    03MAR2003    08MAR2003    Internet Sale    Email     2
   8      1230371142    09MAR2003    11MAR2003    Internet Sale    Email     1
   9      1230404278    15MAR2003    15MAR2003    Catalog Sale     Mail      7
  10      1230440481    22MAR2003    22MAR2003    Catalog Sale     Mail      7
  11      1230450371    24MAR2003    26MAR2003    Internet Sale    Email     2
  12      1230453723    24MAR2003    25MAR2003    Internet Sale    Email     2
  13      1230455630    25MAR2003    25MAR2003    Catalog Sale     Mail      3
  14      1230478006    28MAR2003    30MAR2003    Internet Sale    Email     6
  15      1230498538    01APR2003    01APR2003    Catalog Sale     Mail      3
  16      1230500669    02APR2003    03APR2003    Retail Sale                4
  17      1230503155    02APR2003    03APR2003    Internet Sale    Email     4
  18      1230591673    18APR2003    23APR2003    Internet Sale    Email     6
  19      1230591675    18APR2003    20APR2003    Retail Sale                6
  20      1230591684    18APR2003    18APR2003    Catalog Sale     Mail      6
```

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

   a. Write a DATA step to read **`orion.nonsales`** to create **`Work.gifts`**.

   b. Create two new variables, **`Gift1`** and **`Gift2`**, using a SELECT group with WHEN statements.

   If **`Gender`** is equal to `F`,

   - **`Gift1`** is equal to `Perfume`
   - **`Gift2`** is equal to `Cookware`.

   If **`Gender`** is equal to `M`,

   - **`Gift1`** is equal to `Cologne`
   - **`Gift2`** is equal to `Lawn Equipment`.

   If **`Gender`** is not equal to `F` or `M`,

   - **`Gift1`** is equal to `Coffee`
   - **`Gift2`** is equal to `Calendar`.

   Documentation on the SELECT group with WHEN statements can be found in the SAS Help and Documentation from the Contents tab (**SAS Products** ⇨ **Base SAS** ⇨ **SAS 9.3 Language Reference: Dictionary** ⇨ **Dictionary of Language Elements** ⇨ **Statements** ⇨ **SELECT Statement**).

   c. Include only the following variables: **`Employee_ID`**, **`First`**, **`Last`**, **`Gift1`**, and **`Gift2`**.

   d. Write a PROC PRINT step to create the following report:

   Partial PROC PRINT Output (First 15 of 235 Observations)

```
      Obs    Employee_ID    First        Last           Gift1     Gift2

        1       120101      Patrick      Lu            Cologne    Lawn Equipment
        2       120104      Kareen       Billington    Perfume    Cookware
        3       120105      Liz          Povey         Perfume    Cookware
        4       120106      John         Hornsey       Cologne    Lawn Equipment
        5       120107      Sherie       Sheedy        Perfume    Cookware
        6       120108      Gladys       Gromek        Perfume    Cookware
        7       120108      Gabriele     Baker         Perfume    Cookware
        8       120110      Dennis       Entwisle      Cologne    Lawn Equipment
        9       120111      Ubaldo       Spillane      Cologne    Lawn Equipment
       10       120112      Ellis        Glattback     Perfume    Cookware
       11       120113      Riu          Horsey        Perfume    Cookware
       12       120114      Jeannette    Buddery       Coffee     Calendar
       13       120115      Hugh         Nichollas     Cologne    Lawn Equipment
       14            .      Austen       Ralston       Cologne    Lawn Equipment
       15       120117      Bill         Mccleary      Cologne    Lawn Equipment
```

7. **Subsetting Observations Based on Two Conditions**

   a. Retrieve the starter program **p109e07**.

   b. In the DATA step, write a statement to select only the observations that have **`Emp_Hire_Date`** greater than or equal to July 1, 2006. Subset the observations as they are being read into the program data vector.

   c. In the DATA step, write another statement to select only the observations that have an increase greater than 3000.

   d. Submit the program to create the following PROC PRINT report:

```
                                  Employee
                                   Annual    Employee
        Obs     Employee ID        Salary    Hire Date       Increase     NewSalary

          1          120128        30,890    01NOV2006          3,089        33,979
          2          120144        30,265    01OCT2006          3,027        33,292
          3          120161        30,785    01OCT2006          3,079        33,864
          4          120264        37,510    01DEC2006          3,751        41,261
          5          120761        30,960    01JUL2006          3,096        34,056
          6          120995        34,850    01AUG2006          3,485        38,335
          7          121055        30,185    01AUG2006          3,019        33,204
          8          121062        30,305    01AUG2006          3,031        33,336
          9          121085        32,235    01JAN2007          3,224        35,459
         10          121107        31,380    01JUL2006          3,138        34,518
```

8. **Subsetting Observations Based on Three Conditions**

   a. Write a DATA step to read **`orion.orders`** to create **`Work.delays`**.

   b. Create the new variable **`Order_Month`**, which is equal to the month of **`Order_Date`**.

   c. Use a WHERE statement and a subsetting IF statement to select only the observations that meet all of the following conditions:

   - **`Delivery_Date`** values that are more than four days beyond **`Order_Date`**
   - **`Employee_ID`** values that are equal to 99999999
   - **`Order_Month`** values occurring in August

   d. Write a PROC PRINT step to create the following report:

```
                      Order_                                  Order_     Delivery_  Order_
Obs       Order_ID     Type    Employee_ID   Customer_ID        Date        Date      Month

  1     1231227910       2        99999999         70187   13AUG2003   18AUG2003       8
  2     1231270767       3        99999999            52   20AUG2003   26AUG2003       8
  3     1231305521       2        99999999            16   27AUG2003   04SEP2003       8
  4     1231317443       2        99999999            61   29AUG2003   03SEP2003       8
  5     1233484749       3        99999999          2550   10AUG2004   15AUG2004       8
  6     1233514453       3        99999999         70201   15AUG2004   20AUG2004       8
  7     1236673732       3        99999999             9   10AUG2005   15AUG2005       8
  8     1240051245       3        99999999            71   30AUG2006   05SEP2006       8
  9     1243165497       3        99999999         70201   24AUG2007   29AUG2007       8
```

9. Using an IF-THEN DELETE Statement to Subset Observations

a. Write a DATA step to read `orion.employee_donations` to create `Work.bigdonations`.

b. Create the new variable `Total`, which is equal to the sum of `Qtr1`, `Qtr2`, `Qtr3`, and `Qtr4`.

c. Create the new variable `NoDonation`, which is equal to the count of missing values in `Qtr1`, `Qtr2`, `Qtr3`, and `Qtr4`. Use the NMISS function.

Documentation on the NMISS function can be found in the SAS Help and Documentation from the Contents tab (**SAS Products** ⇨ **Base SAS** ⇨ **SAS 9.3 Language Reference: Dictionary** ⇨ **Dictionary of Language Elements** ⇨ **Functions and CALL Routines** ⇨ **NMISS Function**).

d. The final data set should contain only observations meeting the following two conditions:

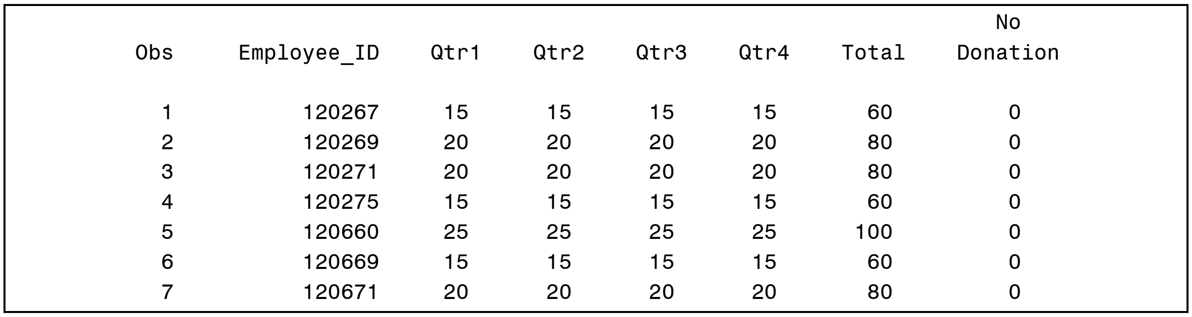
- `Total` values greater than or equal to 50
- `NoDonation` values equal to 0.

Use an IF-THEN DELETE statement to eliminate the observations where the conditions are not met.

The IF-THEN DELETE statement is mentioned at the end of this section in a self-study section.

e. Write a PROC PRINT step with a VAR statement to create the following report:

Partial PROC PRINT Output (First 7 of 50 Observations)



```
                                                                          No
   Obs    Employee_ID    Qtr1    Qtr2    Qtr3    Qtr4    Total    Donation

     1       120267       15      15      15      15       60         0
     2       120269       20      20      20      20       80         0
     3       120271       20      20      20      20       80         0
     4       120275       15      15      15      15       60         0
     5       120660       25      25      25      25      100         0
     6       120669       15      15      15      15       60         0
     7       120671       20      20      20      20       80         0
```