Unit 1-8 Exercises

1. **Validating `orion.shoes_tracker` with the PRINT and FREQ Procedures**

   a. Retrieve the starter program **p108e01**.

   b. The data in **`orion.shoes_tracker`** should meet the following requirements:

   - **`Product_Category`** must not be missing.
   - **`Supplier_Country`** must have a value of `GB` or `US`.

   Add a WHERE statement to the PROC PRINT step to find any observations that do **not** meet the above requirements.

   c. Add a VAR statement to create the following PROC PRINT report:

```
                  Product_                                  Supplier_
           Obs    Category    Supplier_Name                  Country       Supplier_ID

             1     Shoes      3Top Sports                      us                    .
             2                3Top Sports                      US                 2963
             5     Shoes      3Top Sports                      UT                 2963
            10     Shoes      Greenline Sports Ltd             gB                14682
```

   How many observations have missing **`Product_Category`**? ____________________________

   How many observations have invalid values of **`Supplier_Country`**? ____________________

   d. Add a PROC FREQ step with a TABLES statement to create frequency tables for **`Supplier_Name`** and **`Supplier_ID`** of **`orion.shoes_tracker`**. Include the NLEVELS option.

   The data in **`orion.shoes_tracker`** should meet the following requirements:

   - **`Supplier_Name`** must be `3Top Sports` or `Greenline Sports Ltd`.
   - **`Supplier_ID`** must be `2963` or `14682`.

   What invalid data exist for **`Supplier_Name`** and **`Supplier_ID`**? ______________________

2. Validating `orion.qtr2_2007` with the PRINT and FREQ Procedures

a. Write a PROC PRINT step with a WHERE statement to validate the data in **`orion.qtr2_2007`**.

The data in **`orion.qtr2_2007`** should meet the following requirements:

- **`Delivery_Date`** values must be equal to or greater than **`Order_Date`** values.
- **`Order_Date`** values must be in the range of April 1, 2007 – June 30, 2007.

The WHERE statement should find any observations that do **not** meet the above requirements.

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

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

    5     1242012259         1          121040             10       18APR2007    12APR2007
   22     1242449327         3        99999999             27       26JUL2007    26JUL2007
```

How many observations have **`Delivery_Date`** values occurring before **`Order_Date`** values?
____

How many observations have **`Order_Date`** values out of the range of April 1, 2007 – June 30, 2007? ____

c. Add a PROC FREQ step with a TABLES statement to create frequency tables for **`Order_ID`** and **`Order_Type`** of **`orion.qtr2_2007`**. Include the NLEVELS option.

d. Submit the PROC FREQ step.

The data in **`orion.qtr2_2007`** should meet the following requirements:

- **`Order_ID`** must be unique (36 distinct values) and not missing.
- **`Order_Type`** must have a value of `1`, `2`, or `3`.

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

3. Using the PROPCASE Function, Two-Way Frequency Table, and MISSING Option

a. Write a PROC PRINT step with a WHERE statement to validate the data in `orion.shoes_tracker`. All `Product_Name` values should be written in proper case.

Documentation on the PROPCASE 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** ⇨ **PROPCASE Function**).

b. Add a VAR statement to create the following PROC PRINT report:

```
                 Obs       Product_ID              Product_Name

                   3     220200300015    men's running shoes piedmont
                   6     220200300096    Mns.raptor Precision Sg Football
```

c. Add a PROC FREQ step with a TABLES statement to create the following two-way frequency table with `Supplier_Name` and `Supplier_ID` of `orion.shoes_tracker`:

```
                              The FREQ Procedure

                      Table of Supplier_Name by Supplier_ID

               Supplier_Name(Supplier Name)     Supplier_ID(Supplier ID)

               Frequency        |
               Percent          |
               Row Pct          |
               Col Pct          |       .|    2963|   14682|  Total
               -----------------+--------+--------+--------+
               3Top Sports      |       1|       5|       1|      7
                                |   10.00|   50.00|   10.00|  70.00
                                |   14.29|   71.43|   14.29|
                                |  100.00|   71.43|   50.00|
               -----------------+--------+--------+--------+
               3op Sports       |       0|       2|       0|      2
                                |    0.00|   20.00|    0.00|  20.00
                                |    0.00|  100.00|    0.00|
                                |    0.00|   28.57|    0.00|
               -----------------+--------+--------+--------+
               Greenline Sports |       0|       0|       1|      1
                Ltd             |    0.00|    0.00|   10.00|  10.00
                                |    0.00|    0.00|  100.00|
                                |    0.00|    0.00|   50.00|
               -----------------+--------+--------+--------+
               Total                    1        7        2       10
                                    10.00    70.00    20.00   100.00
```

Documentation on two-way frequency tables and the MISSING option can be found in the SAS Help and Documentation from the Contents tab (**SAS Products** ⇨ **Base SAS** ⇨ **Base SAS Procedures Guide: Statistical Procedures** ⇨ **The FREQ Procedure** ⇨ **Syntax: FREQ Procedure** ⇨ **TABLES Statement**).

The data in `orion.shoes_tracker` should meet the following requirements:

- A `Supplier_Name` of `3Top Sports` must have a `Supplier_ID` of `2963`.

- A **Supplier_Name** of Greenline Sports Ltd must have a **Supplier_ID** of 14682.

What invalid data exists for **Supplier_Name** and **Supplier_ID**? ____________________

4. Validating orion.price_current with the MEANS and UNIVARIATE Procedures

a. Retrieve the starter program **p108e04**.

b. Add a VAR statement to the PROC MEANS step to validate **Unit_Cost_Price**, **Unit_Sales_Price**, and **Factor**.

c. Add statistics to the PROC MEANS statement to create the following PROC MEANS report:

```
                                  The MEANS Procedure

    Variable            Label                          N         Minimum         Maximum
    ------------------------------------------------------------------------------------

    Unit_Cost_Price     Unit Cost Price              171       2.3000000     315.1500000
    Unit_Sales_Price    Unit Sales Price             170       6.5000000         5730.00
    Factor              Yearly increase in Price     171       0.0100000     100.0000000
    ------------------------------------------------------------------------------------
```

The data in **orion.price_current** should meet the following requirements:

- **Unit_Cost_Price** must be in the numeric range of 1 – 400.
- **Unit_Sales_Price** must be in the numeric range of 3 – 800.
- **Factor** must be in the numeric range of 1 – 1.05.

What variables have invalid data? ____

d. Add a PROC UNIVARIATE step with a VAR statement to validate **Unit_Sales_Price** and **Factor**.

e. Submit the PROC UNIVARIATE step and find the Extreme Observations output.

How many values of **Unit_Sales_Price** are over the maximum of 800? _______________

How many values of **Factor** are under the minimum of 1? ____________________________

How many values of **Factor** are over the maximum of 1.05? __________________________

5. **Validating `orion.shoes_tracker` with the MEANS and UNIVARIATE Procedures**

   a. Write a PROC MEANS step with a VAR statement to validate **`Product_ID`** of **`orion.shoes_tracker`**.

   b. Add the MIN, MAX, and RANGE statistics to the PROC MEANS statement.

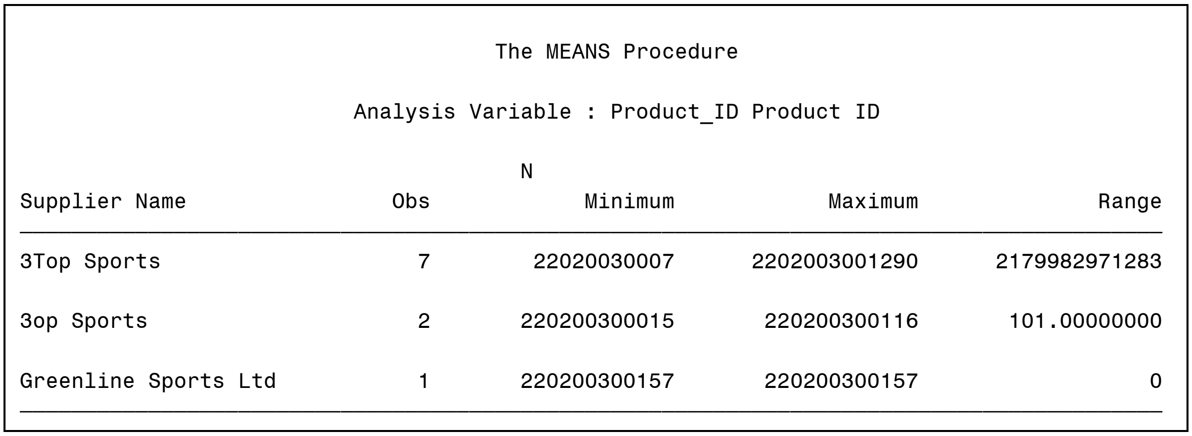
   c. Add **`FW=15`** to the PROC MEANS statement. The FW= option specifies the field width to display the statistics in printed or displayed output.

   d. Add the following CLASS statement to group the data by **`Supplier_Name`**:

```
class Supplier_Name;
```

Documentation on the FW= option and the CLASS statement can be found in the SAS Help and Documentation from the Contents tab (**SAS Products** ⇨ **Base SAS** ⇨ **Base SAS 9.3 Procedures Guide** ⇨ **Procedures** ⇨ **The MEANS Procedure**).

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



```
                              The MEANS Procedure

                    Analysis Variable : Product_ID Product ID

                                    N
Supplier Name                     Obs          Minimum          Maximum            Range
------------------------------------------------------------------------------------------
3Top Sports                         7      22020030007    2202003001290    2179982971283

3op Sports                          2     220200300015     220200300116    101.00000000

Greenline Sports Ltd                1     220200300157     220200300157                0
------------------------------------------------------------------------------------------
```

Which **`Supplier_Name`** has invalid **`Product_ID`** values assuming **`Product_ID`** must have only twelve digits? ____

   f. Add a PROC UNIVARIATE step with a VAR statement to validate **`Product_ID`** of **`orion.shoes_tracker`**.

   g. Submit the PROC UNIVARIATE step and find the Extreme Observations output.

How many values of **`Product_ID`** are too small? ______________________________

How many values of **`Product_ID`** are too large? ______________________________

6. Selecting Only the Extreme Observations Output from the UNIVARIATE Procedure

a. Write a PROC UNIVARIATE step with a VAR statement to validate **`Product_ID`** of **`orion.shoes_tracker`**.

b. Before the PROC UNIVARIATE step, add the following ODS statement:

```
ods trace on;
```

c. After the PROC UNIVARIATE step add, the following ODS statement:

```
ods trace off;
```

d. Submit the program and notice the trace information in the SAS log.

What is the name of the last Output Added in the SAS log? ________________________________

e. Add an ODS SELECT statement immediately before the PROC UNIVARIATE step to select only the Extreme Observation output object.

✎ Documentation on the ODS TRACE and ODS SELECT statements can be found in the SAS Help and Documentation from the Contents tab (**SAS Products** ⇨ **Base SAS** ⇨ **SAS 9.3 Output Delivery System User's Guide** ⇨ **ODS Language Statements** ⇨ **Dictionary of ODS Language Statements**).

f. Submit the program to create the following PROC UNIVARIATE report:

```
                        The UNIVARIATE Procedure
                   Variable:  Product_ID  (Product ID)

                          Extreme Observations

           --------Lowest-------        -------Highest------

                  Value      Obs             Value      Obs

            2.20200E+10        4        2.2020E+11        6
            2.20200E+11        1        2.2020E+11        7
            2.20200E+11        2        2.2020E+11        9
            2.20200E+11        3        2.2020E+11       10
            2.20200E+11        5        2.2020E+12        8
```

7. Cleaning Data from `orion.qtr2_2007`

a. Retrieve the starter program **p108e07**.

b. Add two conditional statements to the DATA step to correct the following invalid data:

Variable	Obs	Invalid Value	Correct Value	Reference Variable
`Delivery_Date`	5	`12APR2007`	`12MAY2007`	**`Order_ID=1242012259`**
`Order_Date`	22	`26JUL2007`	`26JUN2007`	**`Order_ID=1242449327`**

c. Submit the program. Verify that zero observations were returned from the PROC PRINT step.

8. Cleaning Data from `orion.price_current`

a. Retrieve the starter program **p108e08**.

b. Add a DATA step prior to the PROC steps to read **`orion.price_current`** to create **`Work.price_current`**. In the DATA step, include two conditional IF-THEN statements to correct the following invalid data:

Variable	Obs	Invalid Value	Correct Value	Reference Variable
`Unit_Sales_Price`	41	`5730`	`57.30`	**`Product_ID=220200200022`**
`Unit_Sales_Price`	103	`.`	`41.20`	**`Product_ID=240200100056`**

c. Submit the program. Verify that **`Unit_Sales_Price`** is in the numeric range of 3 – 800.