Unit 2-5 Exercises

1. Extracting Characters Based on Position

The data set **`orion.au_salesforce`** has employee data for the Sales branch in Australia.

Partial Listing of **`orion.au_salesforce`** (63 Total Observations, 9 Total Variables)

```
                   First_
     Employee_ID   Name         Last_Name      Job_Title

          120102   Tom          Zhou          Sales Manager
          120103   Wilson       Dawes         Sales Manager
          120121   Irenie       Elvish        Sales Rep. II
          120122   Christina    Ngan          Sales Rep. II
          120123   Kimiko       Hotstone      Sales Rep. I
          120124   Lucian       Daymond       Sales Rep. I
```

a. Orion Star wants to create user ID codes for logging onto the Australian Sales intranet site.

- Each user ID will consist of the first letter of the first name, the final letter of the first name, and the first four letters of the last name.
- All these letters should be lowercase.
- As a first step to doing this, extract these letters and change their case.
- Create a new data set named **`work.codes`**.
- Create three new variables named **`FCode1`**, **`FCode2`**, and **`LCode`**. As described above, these variables should contain the following:

Variable Name	Value
FCode1	First letter of **`First_Name`**
FCode2	Final letter of **`First_Name`**
LCode	First four letters of **`Last_Name`**

Remember to make these new values lowercase too.

There are several ways to approach getting the final letter of **`First_Name`**. For one of those ways, you need to know that the length of **`First_Name`** is 12 characters.

b. Print the resulting data set.

- Include only the variables **`First_Name`**, **`FCode1`**, **`FCode2`**, **`Last_Name`**, and **`LCode`**.
- Add an appropriate title.
- Verify your output.

Partial PROC PRINT output (63 Total Observations)

```
                               Extracted Letters for User IDs

                    First_
             Obs    Name         FCode1    FCode2    Last_Name       LCode

               1    Tom            t         m       Zhou            zhou
               2    Wilson         w         n       Dawes           dawe
               3    Irenie         i         e       Elvish          elvi
               4    Christina      c         a       Ngan            ngan
               5    Kimiko         k         o       Hotstone        hots
               6    Lucian         l         n       Daymond         daym
               7    Fong           f         g       Hofmeister      hofm
               8    Satyakam       s         m       Denny           denn
               9    Sharryn        s         n       Clarkson        clar
```

Later you see techniques that can be used to combine these letters into a single character value.

2. Extracting Characters Based on Position

The data set **`orion.newcompetitors`** has data on competing retail stores that recently opened near existing Orion Star locations.

Listing of **`orion.newcompetitors`**

```
                                              Postal_
                           ID       City          Code

                        AU15301W    PERTH        6002
                        AU12217E    SYDNEY       2000
                        CA    150   Toronto      M5V 3C6
                        CA    238   Edmonton     T5T 2B2
                        US 356NC    charlotte    28203
                        US1013CO    denver       80201
                        US   12CA   San diego    92139
```

a. Orion Star would like a data set containing only the small retail stores from these observations.

- Create a new variable, **`Country`**, that contains the first two characters of **`ID`**.
- Create a new variable, **`Store_Code`**, that contains the other characters from the value in **`ID`**. Left justify the value so that there are no leading blanks.
- The first character in the value of **`Store_Code`** indicates the size of the store, and **`1`** is the code for a small retail store.
- Write a program to output only the small retail store observations.

  Hint: You might need to use a SUBSTR functions as part of a subsetting IF statement
- Make sure that the **`City`** values appear in proper case (as displayed below).

b. Print your results with an appropriate title.

Only show the columns **Store_Code**, **Country**, **City**, and **Postal_Code**.

PROC PRINT output (5 Total Observations)

```
                          New Small-Store Competitors

                   Store_                              Postal_
                    Code     Country    City            Code

                   15301W      AU       Perth          6002
                   12217E      AU       Sydney         2000
                   150         CA       Toronto        M5V 3C6
                   1013CO      US       Denver         80201
                   12CA        US       San Diego      92139
```

3. Converting U.S. Postal Codes to State Names

The data set **orion.contacts** contains a list of contacts for the U.S. charities that Orion Star donates to.

Listing of **orion.contacts**

```
  ID     Title    Name                        Address1                    Address2

 AQI     Ms.      Farr,Sue                    15 Harvey Rd.               Macon, GA  31298
 CCI     Dr.      Cox,Kay B.                  163 McNeil Pl.              Kern, CA  93280
 CNI     Mr.      Mason,Ron                   442 Glen Ave.               Miami, FL  33054
 CS      Ms.      Ruth,G. H.                  2491 Brady St.              Munger, MI  48747
 CU      Prof.    Florentino,Helen-Ashe H.    PO Box 2253                 Washington, DC 20018
 DAI     Ms.      Van Allsburg,Jan F.         25 Chesire Pl.              Short Hills, NJ  07078
 ES      Mr.      Laff,Stanley X.             1725 Airport Rd.            Springfield, IL  62707
 FFC     Mr.      Rizen,George Q.             11234 W Hoyt St.            Chicago, IL  60612
 MI      Dr.      Mitchell,Marc J.            922 Mitchell Circle         Chicago, IL  60603
 SBA     Ms.      Mills,Dorothy E.            34 Clear Sky Rd.            Butte, MT  59750
 V2      Dr.      Webb,Jonathan W.            1012 Hwy 54                 Morrisville, NC  27560
 YYCR    Mr.      Keenan,Maynard J.           1315 Green Valley Ln.       Sedona, AZ  86351
```

a. Create a new data set named **states** that includes the variables **ID** and **Name** as well as a new variable named **Location** that shows the full name in proper case for the state that the contact is based in.

Hint: **Address2** is 24 characters long and the last item in **Address2** is always the ZIP code. Look in the online Help for character functions that use ZIP codes as arguments.

b. Print your results.

PROC PRINT output (12 Total Observations)

```
                     ID      Name                            Location

                    AQI      Farr,Sue                        Georgia
                    CCI      Cox,Kay B.                      California
                    CNI      Mason,Ron                       Florida
                    CS       Ruth,G. H.                      Michigan
                    CU       Florentino,Helen-Ashe H.        District of Columbia
                    DAI      Van Allsburg,Jan F.             New Jersey
                    ES       Laff,Stanley X.                 Illinois
                    FFC      Rizen,George Q.                 Illinois
                    MI       Mitchell,Marc J.                Illinois
                    SBA      Mills,Dorothy E.                Montana
                    V2       Webb,Jonathan W.                North Carolina
                    YYCR     Keenan,Maynard J.               Arizona
```

4. Cleaning Text Data

Customer names are available in a data set named **`orion.customers_ex5`**:

Customer_ID	Name	Country	Gender	Birth_Date
000-000-00-0004	KVARNIQ, James	US	M	27Jun1974
Silver000-000-00-0005	STEPHANO, Sandrina	US	F	9-Jul1979
000-000-00-0009	KRAHL, Cornelia	DE	F	27Feb1974
platinum000-000-00-0010	BALLINGER, Karen	US	F	18Oct1984
000-000-00-0011	WALLSTAB, Elke	DE	F	16Aug1974
Silver000-000-00-0012	BLACK, David	US	M	12Apr1969

Use this data set to create a new data set named **`names`** that contains each customer's name in this format:

Mr. John B. Smith

Ms. Jane Doe

a. Write a program to create the **`names`** data set.

- The **`names`** data set should contain only three variables: **`New_Name`**, **`Name`**, and **`Gender`**.
- The **`New_Name`** variable should contain the customer's name in the new format.
  - Female names should be preceded by the honorific title Ms.
  - Male names by the title Mr.

b. Print the **`names`** data set.

c. Verify that your conversion efforts were successful.

Partial PROC PRINT Output (77 Total Observations)

```
        Obs    New_Name                    Name                     Gender
          1    Mr. James Kvarniq           KVARNIQ, James             M
          2    Ms. Sandrina Stephano       STEPHANO, Sandrina         F
          3    Ms. Cornelia Krahl          KRAHL, Cornelia            F
          4    Ms. Karen Ballinger         BALLINGER, Karen           F
          5    Ms. Elke Wallstab           WALLSTAB, Elke             F
          6    Mr. David Black             BLACK, David               M
          7    Mr. Markus Sepke            SEPKE, Markus              M
```

5. (Optional) Searching for and Replacing Character Values

- As in the previous exercise, the data set **`orion.customers_ex5`** contains information about Orion Star customers.
- Customers who are frequent purchasers are tagged as `Silver`, `Gold`, or `Platinum`, which appears at the beginning of their **`Customer_ID`** value.
- Due to updates in the way that Orion Star designates **`Customer_ID`** values, the existing values need to be modified. Any four-digit string, for example, `-00-`, in **`Customer_ID`** should be replaced by `-15-` in the output data sets.

  a. Create three output data sets: **`work.silver`**, **`work.gold`**, and **`work.platinum`**.

  - Search **`Customer_ID`** for the values `Silver`, `Gold`, and `Platinum` and output them to the respective data set when they are found.
  - You should get 17 observations in **`work.silver`**, 2 in **`work.gold`**, and 5 in **`work.platinum`**.
  - Keep the variables **`Customer_ID`**, **`Name`**, and **`Country`** in all data sets.

  b. Print the data sets with appropriate titles.

  c. Confirm that any `-00-` was replaced by `-15-`.

  Hint: Make sure that your searches are not case sensitive!

Partial PROC PRINT Output (17 Total Observations)

```
                                Silver-Level Customers

                      Customer_ID         Name                  Country

                 Silver000-000-15-0005    STEPHANO, Sandrina      US
                 Silver000-000-15-0012    BLACK, David            US
                 Silver000-000-15-0024    KLEM, Robyn             US
```

PROC PRINT Output (2 Total Observations)

```
                               Gold-Level Customers

                    Customer_ID              Name            Country

                 Gold000-000-15-0027    MCCLUNEY, Cynthia      US
                 Gold000-000-07-0201    BORWICK, Angel         CA
```

PROC PRINT Output (5 Total Observations)

```
                                     Platinum-Level Customers

                          Customer_ID          Name                    Country

                    platinum000-000-15-0010    BALLINGER, Karen          US
                    platinum000-000-15-0031    MARTINEZ, Cynthia         US
                    platinum000-000-15-0171    BOWERMAN, Robert          AU
                    platinum000-000-15-2806    VAN DEN BERG, Raedene     ZA
                    platinum000-000-07-0100    YEARGAN, Wilma            CA
```

6. Searching Character Values and Explicit Output

- The data set **`orion.employee_donations`** contains information on charity contributions from Orion Star employees.
- Each employee is allowed to list either one or two charities, which are shown in the **`Recipients`** variable.

Partial Listing of **`orion.employee_donations`** (124 Total Observations, 7 Total Variables)

```
                Employee_ID    Recipients

                     120265    Mitleid International 90%, Save the Baby Animals 10%
                     120267    Disaster Assist, Inc. 80%, Cancer Cures, Inc. 20%
                     120269    Cancer Cures, Inc. 10%, Cuidadores Ltd. 90%
                     120270    AquaMissions International 10%, Child Survivors 90%
                     120271    Cuidadores Ltd. 80%, Mitleid International 20%
                     120272    AquaMissions International 10%, Child Survivors 90%
                     120275    AquaMissions International 60%, Child Survivors 40%
                     120660    Disaster Assist, Inc.
                     120662    Cancer Cures, Inc.
                     120663    EarthSalvors 30%, Vox Victimas 70%
```

Some charity names have a comma in them.

a. Use explicit output to create a data set named **`work.split`**.

- The data set will have one observation for each combination of employee and charity to which he donated.
- Some employees made two contributions; therefore, they will have two observations in the output data set. These employees contain a % character in the value of **`Recipients`**.

Store the position where the % character was found in a variable named **`PctLoc`**. This can make subsequent coding easier.

- Create a variable named **`Charity`** with the name and percent contribution of the appropriate charity.
- Read only the first 10 observations from **`orion.employee_donations`** to test your program.

b. Modify the program to read the entire **`orion.employee_donations`** data set.

- Print only the columns **`Employee_ID`** and **`Charity`**.

- Add an appropriate title.

Partial PROC PRINT Output (212 Total Observations)

```
                              Charity Contributions for each Employee

                           Employee_ID    Charity

                                120265    Mitleid International 90%
                                120265    Save the Baby Animals 10%
                                120267    Disaster Assist, Inc. 80%
                                120267    Cancer Cures, Inc. 20%
                                120269    Cancer Cures, Inc. 10%
                                120269    Cuidadores Ltd. 90%
                                120270    AquaMissions International 10%
                                120270    Child Survivors 90%
                                120271    Cuidadores Ltd. 80%
                                120271    Mitleid International 20%
                                120272    AquaMissions International 10%
                                120272    Child Survivors 90%
                                120275    AquaMissions International 60%
                                120275    Child Survivors 40%
                                120660    Disaster Assist, Inc. 100%
                                120662    Cancer Cures, Inc. 100%
                                120663    EarthSalvors 30%
                                120663    Vox Victimas 70%
```

7. Using Character Functions with the Input Buffer

- The raw data file **supply.dat** contains information on supplier IDs (up to five characters), supplier names (up to 30 characters), and the country from which that supplier ships (two characters).

Raw Data File **supply.dat** (52 rows total)

```
                                 50 Scandinavian Clothing A/S NO
109 Petterson AB SE
316 Prime Sports Ltd GB
755 Top Sports DK
772 AllSeasons Outdoor Clothing US
798 Sportico ES
1280 British Sports Ltd GB
1303 Eclipse Inc US
1684 Magnifico Sports PT
1747 Pro Sportswear Inc US
2963 3Top Sports US
2995 Van Dammeren International NL
```

- The keyword _INFILE_, when SAS reads from a raw data file, enables you to treat the contents of the input buffer as one long character string. This can sometimes be helpful, given the wide variety of character functions in SAS.
- Blanks appear both as delimiters and inside supplier names.

Remember that both the SCAN and FIND functions can process backward through strings. See SAS Help and Documentation for more details on how to do this.

a. Create a data set named **`work.supplier`**.

- Use list input to obtain values for **`Supplier_ID`**.
- Utilize character functions and the _INFILE_ statement to get values for **`Supplier_Name`** and **`Country`**.

b. Print the data set with an appropriate title.

Partial PROC PRINT Output (52 Total Observations)

```
                                  Supplier Information

                 Supplier_
                    ID       Supplier_Name                     Country

                   50        Scandinavian Clothing A/S           NO
                   109       Petterson AB                        SE
                   316       Prime Sports Ltd                    GB
                   755       Top Sports                          DK
                   772       AllSeasons Outdoor Clothing         US
                   798       Sportico                            ES
                   1280      British Sports Ltd                  GB
                   1303      Eclipse Inc                         US
```

8. Calculating Statistics and Rounding

The data set **`orion.orders_midyear`** contains an observation for each customer, with the total retail value of the customer's monthly orders for the first half of the year.

Partial Listing of **`orion.orders_midyear`** (24 Total Observations)

```
          Customer_ID    month1    month2    month3     month4     month5     month6

                    5     213.1       .       478.0     525.80     394.35     191.79
                   10     188.1    414.09    2876.9    3164.59    2373.44     169.29
                   11      78.2       .         .          .          .        70.38
                   12     135.6       .       117.6     129.36      97.02     122.04
                   18        .        .        29.4      32.34      24.26        .
                   24      93.0    265.80       .          .          .        83.70
                   27     310.7    782.90       .          .          .       279.63
                   31    1484.3    293.30       .          .          .      1335.87
                   34     642.5       .        86.3      94.93      71.20     578.25
```

a. Create a data set named **`work.sale_stats`** with three new variables for all months in which the customer placed an order.

- The variable **`MonthAvg`** should contain the average.
- The variable **`MonthMax`** should contain the maximum.
- The variable **`MonthSum`** should contain the sum of values.
- Round **`MonthAvg`** to the nearest integer.

Most SAS descriptive statistics functions automatically ignore missing values.

b. Print the variables **`Customer_ID`**, **`MonthAvg`**, **`MonthMax`**, and **`MonthSum`**. Add an appropriate title.

Partial PROC PRINT Output (24 Total Observations)

```
                 Statistics on Months in which the Customer Placed an Order

                                    Month      Month        Month
                    Customer_ID      Avg        Max          Sum

                              5      361      525.80     1803.04
                             10     1531     3164.59     9186.41
                             11       74       78.20      148.58
                             12      120      135.60      601.62
                             18       29       32.34       86.00
                             24      148      265.80      442.50
                             27      458      782.90     1373.23
                             31     1038     1484.30     3113.47
                             34      295      642.50     1473.18
```

9. Calculating Statistics for Missing, Median, and Highest Values

The data set **`orion.orders_midyear`** contains an observation for each customer, with the total retail value of the customer's monthly orders for the first half of the year.

Partial Listing of **`orion.orders_midyear`** (24 Total Observations)

```
           Customer_ID    month1     month2     month3      month4      month5      month6

                     5     213.1        .        478.0      525.80      394.35      191.79
                    10     188.1     414.09     2876.9     3164.59     2373.44      169.29
                    11      78.2        .          .           .           .         70.38
                    12     135.6        .        117.6      129.36       97.02      122.04
                    18        .         .         29.4       32.34       24.26         .
                    24      93.0     265.80        .           .           .         83.70
                    27     310.7     782.90        .           .           .        279.63
                    31    1484.3     293.30        .           .           .       1335.87
                    34     642.5        .         86.3       94.93       71.20      578.25
```

a. Orion Star wants to look at information on the median order and the top two months' orders, but only for frequent customers.

- Create a data set named **`work.freqcustomers`** that contains the requested statistics.
- Frequent customers are defined to be those who placed an order in at least five of the six months.

b. Print your results with an appropriate title.

✎ The *median* of a set of values is the middle or central value. For example, the median
of {1, 200, 3} is the value 3. If the set has an even number of values, the median is the midpoint between the two middle values.

✎ Consult the SAS documentation, as needed, to learn more about functions that can generate the desired results. It might be particularly useful to look at "Functions and CALL Routines by Category" in the *SAS Language Dictionary*.

PROC PRINT Output

```
                         Month Statistics on Frequent Customers

                                                                                 Month_
                                                                Month_   Month_   2nd
Customer_ID  month1  month2  month3   month4   month5   month6   Median  Highest  Highest

          5  213.10     .     478.0   525.80   394.35  191.790   394.35   525.80   478.00
         10  188.10  414.09  2876.9  3164.59  2373.44  169.290  1393.77  3164.59  2876.90
         12  135.60     .     117.6   129.36    97.02  122.040   122.04   135.60   129.36
         34  642.50     .      86.3    94.93    71.20  578.250    94.93   642.50   578.25
         41  134.00  119.20   313.0   344.30   258.23  120.600   196.11   344.30   313.00
         45  443.88  216.20    40.2    44.22    33.17  399.492   130.21   443.88   399.49
         90   33.60  110.20   396.9   436.59   327.44   30.240   218.82   436.59   396.90
         92   16.90     .     160.5   176.55   132.41   15.210   132.41   176.55   160.50
        171   73.99  534.60  1241.4  1365.54  1024.16   66.591   779.38  1365.54  1241.40
```

10. Using the PUT and INPUT Functions

The data set **`orion.shipped`** contains details about each product shipped to one of Orion Star's retail outlets in 2007.

Partial Listing of **`orion.shipped`**

```
 Product_ID      Ship_Date    Quantity      Price

240800200021     05JAN2007        2        $42.45
240800200035     04JAN2007        6        $12.15
240200100225     04JAN2007        2        $77.85
210200500002     09JAN2007        3         $5.70
```

Partial PROC CONTENTS Output for **`orion.shipped`**

```
      Variables in Creation Order

#    Variable      Type    Len    Format

1    Product_ID    Num       8
2    Ship_Date     Num       8    DATE9.
3    Quantity      Num       8
4    Price         Char      7
```

An analyst at Orion Star has written a SAS program to calculate the total price of the items shipped and create a comment that includes the ship date. Unfortunately, the SAS program is giving unexpected results.

a. Open and submit the program, **p205e10.sas**.

b. View the unexpected results.

Partial PROC PRINT of Unexpected Results

```
   Product_ID      Ship_Date    Quantity     Price        Comment           Total

  240800200021     05JAN2007        2        $42.45    Shipped on 17171        .
  240800200035     04JAN2007        6        $12.15    Shipped on 17170        .
  240200100225     04JAN2007        2        $77.85    Shipped on 17170        .
  210200500002     09JAN2007        3         $5.70    Shipped on 17175        .
```

c. Modify the program to generate the expected results.

Partial PROC PRINT of Desired Results

```
  Product_ID      Ship_Date    Quantity     Price            Comment              Total

  240800200021     05JAN2007        2        $42.45    Shipped on 01/05/2007     $84.90
  240800200035     04JAN2007        6        $12.15    Shipped on 01/04/2007     $72.90
  240200100225     04JAN2007        2        $77.85    Shipped on 01/04/2007    $155.70
  210200500002     09JAN2007        3         $5.70    Shipped on 01/09/2007     $17.10
```

- Look above at the PROC CONTENTS output for **`orion.shipped`**.
- Notice that **`Ship_Date`** is numeric with a permanently assigned DATE9. format. It needs to be converted into a character value using the MMDDYY10. format.
- Notice that **`Price`** is character. It needs to be converted into a numeric value using the COMMA7.2 or DOLLAR7.2 informat.
- Use functions to convert the values of **`Ship_Date`** and **`Price`** to get the desired results.

11. Changing a Variable's Data Type

The data set **`orion.US_newhire`** contains information about newly hired employees.

Partial Listing of **`orion.US_newhire`**

```
       ID           Telephone    Birthday

 120-012-40-4928      5467887     05DEC1968
 120-012-83-3816      6888321     03MAY1965
 120-341-44-0781      9418123     23NOV1972
 120-423-01-7721      7839191     28JUN1967
```

Partial PROC CONTENTS Output of **`orion.US_newhire`**

```
      Variables in Creation Order

   #    Variable     Type    Len

   1    ID           Char     15
   2    Telephone    Num       8
   3    Birthday     Char      9
```

a. Create a new data set from **`orion.US_newhire`**.

- Name the new data set **`US_converted`**.
- Remove the embedded dashes in **`ID`**.
- Convert **`ID`** to a numeric value.
- Convert **`Telephone`** to character and place a – (hyphen/dash) between the third and forth digits.
- Convert **`Birthday`** to a SAS date value.

b. Print **`US_converted`** with an appropriate title and use PROC CONTENTS to check the variables types.

Partial PROC PRINT of **`US_converted`** (10 Total Observations)

```
                    US New Hires

             ID          Telephone    Birthday

        120012404928    546-7887       3261
        120012833816    688-8321       1949
        120341440781    941-8123       4710
        120423017721    783-9191       2735
```

Partial PROC CONTENTS of **`US_converted`**

```
        Variables in Creation Order

       #    Variable     Type    Len

       1    ID           Num       8
       2    Telephone    Char      8
       3    Birthday     Num       8
```