Chapter 3. Data Manipulation

3.1. Sort Datasets: PROC SORT

- A dataset can be sorted by one or more variables.
- Overwrite the existing dataset unless using out= option.
- By default, sort in ascending order.
- Sort with respect to the order of variable list.

General Syntax

```
proc sort data=dataset out=new-data;
     * var1: ascending, var2: descending;
     by var1 descending var2;
run;
```

3.2. Subset Datasets: IF or WHERE

- Select observations from one dataset by defining selection criteria.

General Syntax

```
data new-dataset;
     set dataset;
     where condition;
     if condition;
run;
```

3.3. IF-THEN/ELSE Statement

- Useful when grouping observations based on multiple conditions

General Syntax

```
* Multiple criteria;
if condition then action1;
else if condition then action2;
else action3;

* DO & END: Execute multiple actions;
if condition then do;
action1; action2; action3;
end;
```

Example

Raw Data

Data1

Obs	ID	TREAT	INITWT	WT3MOS	AGE
1	1	Other1	166.28	146.98	35
2	2	Other2	214.42	210.22	30
3	3	Other2	172.46	159.42	33
4	5	Other2	175.41	160.66	30
5	6	Other2	173.13	169.40	20
6	7	Other1	181.25	170.94	30
7	10	Other1	239.83	214.48	48
8	11	Other1	175.32	162.66	51
9	12	Other2	227.01	211.06	29
10	13	Other2	274.82	251.82	31

SAS Code

```
proc sort data=data1;
     by ID;
run;

proc sort data=data1 out=data1_sort;
     by age descending initwt;
run;

data subset1;
     set data1;
     where TREAT = "Other1";
run;

data subset2;
     set data1;
     if AGE > 30;
run;

data subset3;
     set data1;
     if AGE <= 30 then delete;
run;

data ifelse;
     set data1;
     length agegroup $5.;
     if age >= 50 then agegroup = "50+";
     else if age >= 30 & age < 50 then agegroup = "30-50";
     else agegroup = "-30";
run;
```

3.4. Combine Datasets

- SET statement
  - Concatenate (stack) datasets.
  - If one of the datasets has a variable not contained in the other, missing values will be added instead.
  - Add BY statement after sorting datasets to interleave datasets.

General Syntax

```
data new-dataset;
     set dataset1 ... datasetn;
run;
```

- PROC APPEND
  - Useful when the two datasets contain exactly same variables (If not, ERROR).

General Syntax

```
proc append base=dataset1 data=dataset2;
run;
```

- MERGE statement
  - Useful when combining datasets from different sources
  - All datasets must be *sorted* first by the matching variables.
  - If you merge two datasets that have other variables in common, then the variables from the second dataset will overwrite the variables with the same name in the first dataset.
  - One-to-one: Only one observation for each value of the BY variable in all datasets.
  - One-to-many: One dataset has one observation for each value of the BY variable, while the other has multiple observations.
  - Many-to-many: More than one observation with a given BY variable in each dataset.

General Syntax

```
proc sort data=dataset1;
     by ID-Variable; run;
...
proc sort data=datasetn;
     by ID-Variable; run;

data new-dataset;
     merge dataset1 ... datasetn;
     by ID-Variable;
run;
```

- Divide a dataset into multiple datasets

General Syntax

```
data new-dataset1 new-dataset2; * Create 2 datasets;
     set from-dataset;
     if condition then output new-dataset1;
     else output new-dataset2;
run;
```

- MERGE (IN= Option)
  - Helpful to know which dataset an observation comes from
  - Create an indicator variable (0 / 1) that indicates whether the current observation comes from the input dataset or not.
  - Make sure that only complete records are collected in one dataset, and create another dataset with partially missing observations.

General Syntax

```
data compete missing;
merge dataset1(in=in1) dataset2(in=in2);
by id-variable;
if in1 and in2 then output complete; * Check for complete observations;
else output missing;
run;
```

Example

Raw Data

```
data data1;
input ID TREAT $
      INITWT WT3MOS AGE;
cards;
1 Other1 166.28 146.98 35
2 Other2 214.42 210.22 30
3 Other2 172.46 159.42 33
5 Other2 175.41 160.66 30
6 Other2 173.13 169.40 20
7 Other1 181.25 170.94 30
10 Other1 239.83 214.48 48
11 Other1 175.32 162.66 51
12 Other2 227.01 211.06 29
13 Other2 274.82 251.82 31
;
run;
```

```
data data2;
input ID TREAT $ INITWT
      WT3MOS AGE;
cards;
14 Surgery 203.60 169.78 38
17 Surgery 171.52 150.33 42
18 Surgery 207.46 155.22 41
;
run;
```

```
data data3;
input ID GENDER $ AREA
      $ @@;
cards;
1 F NY 1 F NJ
6 F CA 8 M PA
11 M CT 12 M AZ
14 F GA 16 M IL
17 M NC 18 F OH
;
run;
```

SAS Code

```
* Case1) Stacking;
data set1;
     set data1 data2;
*    by ID;
run;

* Case2) PROC APPEND;
proc append base=data1 data=data2;
run;

* Case3) Merging;
data merge1;
     merge set1 data3;
     by ID;
run;
```

```
data complete missing;
     merge set1(in=in1) data3(in=in2);
     by id;
     if in1 and in2 then output complete;
     else output missing;
run;

data heavy light;
     set set1;
     if initwt > 180 then output heavy;
     else output light;
run;
```

3.5. Operators in SAS

Operator	Definition	Operator		Definition
		Symbolic	Mnemonic	
*	Multiplication	=	EQ	Equal to
+	Addition	^=	NE	Not equal to
-	Subtraction	>	GT	Greater than
**	Exponentiation	>=	GE	Greater than or equal to
/	Division	<	LT	Less than
		<=	LE	Less than or equal to
			IN	Equal to one of the list
		&	AND	All comparisons must be true.
		|, ¦, !	OR	At least one comparison must be true.

3.6. Modify, Delete and Rename Variables

- The assignment statement can be used to create/modify/delete variables in the DATA step.
- KEEP = list-of-variables: Tell SAS which variables to keep.
- DROP = list-of-variables: Tell SAS which variables to drop.
- RENAME (old-var = new-var): Tell SAS to rename certain variables.

General Syntax

```
* Case 1) DATA step: RENAME, DROP, KEEP statements;
data new-dataset;
     set dataset;
     rename old-var=new-var;
     drop list-of-variables;
     keep list-of-variables;
run;

* Case 2) DATA step: Either next to dataset name or on SET statement;
data new-dataset (keep=list-of-variables drop=list-of-variables rename=(var=new-var);
     set dataset (keep=list-of-variables drop=list-of-variables rename=(var=new-var);
run;

* Case 3) PROC step;
proc print data=dataset (keep=list-of-variables drop=list-of-variables
                         rename=(var=new-var));
run;
```

Example

Raw Data

Obs	ID	TREAT	INITWT	WT3MOS	AGE
1	1	Other1	166.28	146.98	35
2	2	Other2	214.42	210.22	30
3	3	Other2	172.46	159.42	33
4	5	Other2	175.41	160.66	30
5	6	Other2	173.13	169.40	20
6	7	Other1	181.25	170.94	30
7	10	Other1	239.83	214.48	48
8	11	Other1	175.32	162.66	51
9	12	Other2	227.01	211.06	29
10	13	Other2	274.82	251.82	31
11	14	Surgery	171.52	150.33	42
12	17	Surgery	203.60	169.78	38
13	18	Surgery	207.46	155.22	41

SAS Code

```
data data4;
    set set1;
    rename INITWT = InitialWeight
           WT3MOS = Weight3Months;

    * Define new variables;
    WTdiff = WT3MOS - INITWT;
    INITWT_kg = INITWT * 0.4536;
    INITWT_kg2 = INITWT / 2.2046;

    length agegroup $10.;
    if age >= 50 then agegroup="50+";
    else if age >= 30 then agegroup="30-50";
    else agegroup="-30";
```

```
        if agegroup in ("50+", "-30") then delete;

        drop agegroup;
        keep ID INITWT WT3MOS WTdiff INITWT_kg INITWT_kg2 Age;
        format INITWT_kg 6.2 INITWT_kg2 8.4;

run;
```

Output

Obs	ID	InitialWeight	Weight3Months	AGE	WTdiff	INITWT_kg	INITWT_kg2
1	1	166.28	146.98	35	-19.30	75.42	75.4241
2	2	214.42	210.22	30	-4.20	97.26	97.2603
3	3	172.46	159.42	33	-13.04	78.23	78.2273
4	5	175.41	160.66	30	-14.75	79.57	79.5655
5	7	181.25	170.94	30	-10.31	82.22	82.2145
6	10	239.83	214.48	48	-25.35	108.79	108.7862
7	13	274.82	251.82	31	-23.00	124.66	124.6575
8	14	171.52	150.33	42	-21.19	77.80	77.8010
9	17	203.60	169.78	38	-33.82	92.35	92.3524
10	18	207.46	155.22	41	-52.24	94.10	94.1032

3.7. Labels

- Make the output more readable and informative.
- How the *variables* appear changes, not the variable names.
- (DATA step) LABEL statement: Labels remain associated with the respective variables.
- (PROC step) LABEL statement: Only used for that procedure

3.8. Formats

- Specify how we want the data *values* to look.
- Use either 1) SAS built-in formats or 2) user-defined formats
- FORMAT statement specified in a DATA step sets the variable format *permanently*.
- FORMAT statement specified in a PROC is only used in that *specific procedure*.
- PROC FORMAT: Define your own formats.
- "format-name" is the name of the format that is used in a FORMAT statement.
- Formats for character start with a $.

- NO semicolon (;) in the VALUE statement until you have covered all possible values.
- Regrouping values using FORMAT: Specify range of values
- For non-integer values, make sure there are no cracks in your ranges.
- For convenience, you can specify user-defined permanent formats under your library.

General Syntax

```
data new-dataset;
     set dataset;
     label var-name-1 = "Label-1"
           ...
           var-name-k = "Label-k";
run;

proc format;
     value format-name  category-1 = "Formatted text-1"
                        ...
                        category-k = "Formatted text-k";
run;

proc procedurename data=dataset;
     format var-name format-name.;
run;
```

Example

SAS Code

```
/* Labeling */
data label;
set set1;
label ID = "Patient ID"
      TREAT = "Treatment"
      INITWT = "Initial Weight"
      WT3MOS = "Weight after 3 Months"
      AGE = "Age";
run;

/* Formatting */
proc format;
value agegroup  0-<30 = "Less than 30"
               30-<50 = "Between 30 and 50"
               50- HIGH = "Greater than or
                           equal to 50";
value $treatment "Surgery" = "Surgical Treatment"
                 "Other1" = "Other Treatment"
                 "Other2" = "Other Treatment";
* $ for character variable;
run;

proc print data=set1;
     format age agegroup. treat $treatment.;
run;
```

Output

Obs	ID	TREAT	INITWT	WT3MOS	AGE
1	1	Other Treatment	166.28	146.98	Between 30 and 50
2	2	Other Treatment	214.42	210.22	Greater than or equal to 50
3	3	Other Treatment	172.46	159.42	Between 30 and 50
4	5	Other Treatment	175.41	160.66	Between 30 and 50
5	6	Other Treatment	173.13	169.40	Less than 30
6	7	Other Treatment	181.25	170.94	Between 30 and 50
7	10	Other Treatment	239.83	214.48	Between 30 and 50
8	11	Other Treatment	175.32	162.66	Greater than or equal to 50
9	12	Other Treatment	227.01	211.06	Less than 30
10	13	Other Treatment	274.82	251.82	Between 30 and 50
11	14	Surgical Treatment	203.60	169.78	Between 30 and 50
12	17	Surgical Treatment	171.52	150.33	Between 30 and 50
13	18	Surgical Treatment	207.46	155.22	Between 30 and 50