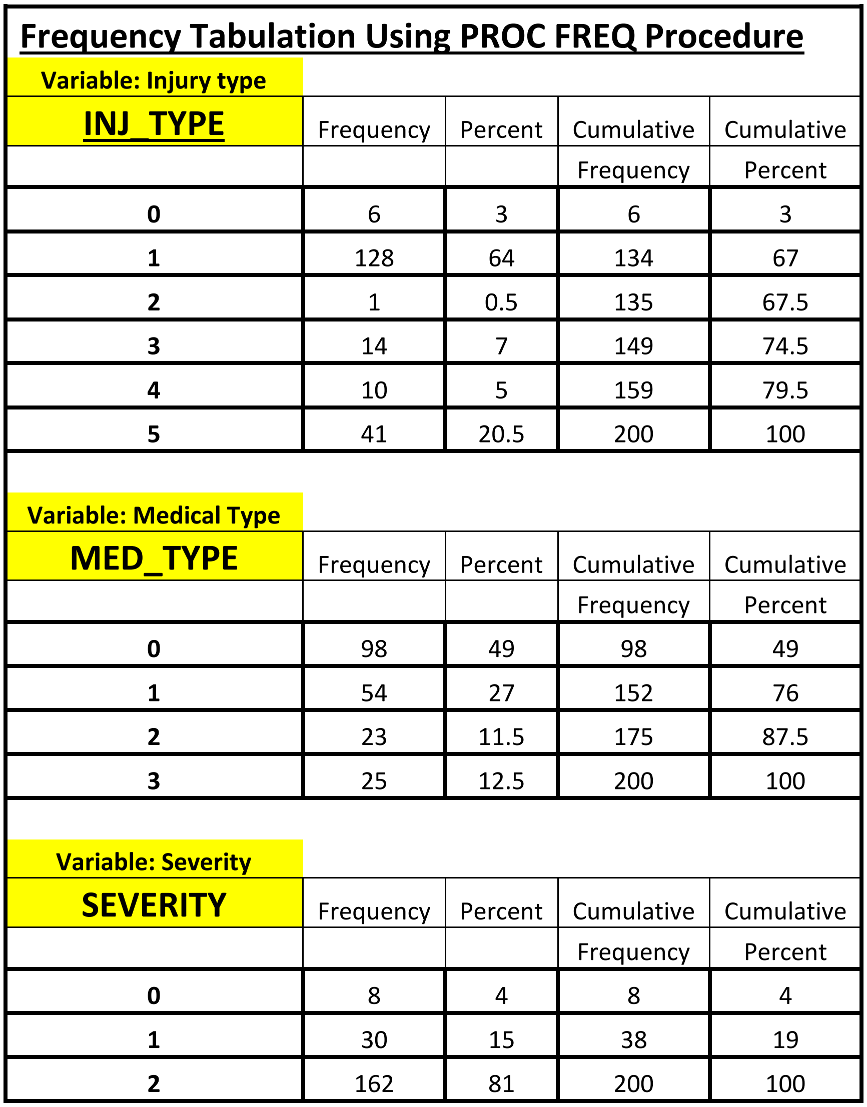
“Value Labels” (FORMATS) and Numerical Functions

Suppose the frequency distribution of 3 variables produced by SAS Proc FREQ is displayed as shown below.

While reading at this output, it is impossible to know what does, for example, the **code 1** stand for in the case of the variable, **INJ_TYPE** (or **MED_TYPE** or **SEVERITY**).

To make this display of the output readable, we add what is called the “Value Labels”.



Frequency Tabulation Using PROC FREQ Procedure

Variable: Injury type

INJ_TYPE	Frequency	Percent	Cumulative	Cumulative
			Frequency	Percent
0	6	3	6	3
1	128	64	134	67
2	1	0.5	135	67.5
3	14	7	149	74.5
4	10	5	159	79.5
5	41	20.5	200	100

Variable: Medical Type

MED_TYPE	Frequency	Percent	Cumulative	Cumulative
			Frequency	Percent
0	98	49	98	49
1	54	27	152	76
2	23	11.5	175	87.5
3	25	12.5	200	100

Variable: Severity

SEVERITY	Frequency	Percent	Cumulative	Cumulative
			Frequency	Percent
0	8	4	8	4
1	30	15	38	19
2	162	81	200	100

“Value Labels” (FORMATS) and Numerical Functions

SYNTAX for adding the “Value Lables” so that the numeric codes used to code the variables are clear before analyzing the data and printing out the output.

Begins with PROC FORMAT; statement

The general format of a VALUE statement

```
VALUE     format name
                         Range1 = ‘label1’
                         Range2 = ‘label2’
                  …………;
```

format name

- **Must begin with a $ sign if the format applies to character data**
- **Cannot be longer than eight characters (In SAS 9, numeric format can be up to 32 characters while the character format can be up to 31 characters)**
- **Cannot be name of an existing SAS format**
- **Cannot end with a number**
- **Does not end in a period when specified in the VALUE statement**

Range **One or more variable values such as: 1, 0 - <25, ‘F’, etc.**

Label **Text string enclosed in quotation marks**

Where do these statements go in the SAS program?

These statements go at the beginning of the program or before the DATA STEP.

CREATION OF “VALUE LABELS”

For the above 3 variables, following are the SAS statements to create VALUE labels

```
Proc format;
     Value Injuryf
          0= "Assult (0)"
          1= "Fall (1)"
          2= "Gun Shot Injury (2)"
          3= "Others (Fall of heavy object, Machine injuries(3)"
          4= "Playing & Sports Related Injuries (4)"
          5= "RTA (5)";

     value Medicalf
          0= "Neurosurgery Related (0)"
          1= "Orthopedics Related (1)"
          2= "Polytrauma (2)"
          3= "Surgically Related (3)";

     value Sevf
          0= "Green (Ambulatory Patients) (0)"
          1= "Red (Resuscitation Bay) (1) "
          2= "Yellow (Patients who require observation) (2)";
run;
```

Note the following:

SAS Variable Name	SAS Format Name	SAS Label for the Variable
INJ_TYPE	INJURYF	Type of Injury
MED_TYPE	MEDICALF	Medical Type
SEVERITY	SEVF	Severity

IMPLEMENTATION OF “VALUE LABELS”

In order to implement these “value labels”, we will need the following SAS statement before the DATALINES statement or between the LABEL statement and DATALINES statement if LABELS are included or within each SAS Procedure.

```
format inj_type injury. Med_type medical. Severity sevf.;
```

The SAS Program would NOW look like the following:

```
Proc format;
      Value Injuryf
                  0= "Assult (0)"
                  1= "Fall (1)"
                  2= "Gun Shot Injury (2)"
                  3= "Others (Fall of heavy object, Machine injuries) (3)"
                  4= "Playing & Sports Related Injuries (4)"
                  5= "RTA (5)";

      value Medicalf
                  0= "Neurosurgery Related (0)"
                  1= "Orthopedics Related (1)"
                  2= "Polytrauma (2)"
                  3= "Surgically Related (3)";

      value Sevf
                  0= "Green (Ambulatory Patients) (0)"
                  1= "Red (Resuscitation Bay) (1)"
                  2= "Yellow (Patients who require observation) (2)";
run;
data example14;
      input @1 mrn 7.0
                  @9 age 2.0
                  @13 sex 1.0
                  @17 inj_type 1.0
                  @21 envr 1.0
                  @25 med_type 1.0
                  @29 GCS 2.0
                  @33 severity 1.0
                  @37 hosp_care 1.0
                  @41 length_stay 3.1;
label
      MRN = 'Medical Record Number'
      AGE = 'Age in Months'
      SEX = 'Gender'
      INJ_TYPE = 'Injury type'
      ENVR = 'Environment where trauma occurred'
      MED_TYPE = 'Medical Type'
      GCS = 'Glasgow Coma Score'
      SEVERITY = 'Severity'
      HOSP_CARE = 'Pre-Hospital Care Given?'
      LENGTH_STAY = 'Length of stay in hours';
datalines;
1748487     26  0 1     0     1     15     2     1     1.5
1748968     6     0     1     0     0      9     1     1     6
1749228     6     0     1     0     0      15    2     0     6.5
....... more data
;
proc freq data=example14;
      title 'Frequency tabulation';
      tables inj_type med_type severity;
      format inj_type injuryf. Med_type medicalf. severity sevf.;
run;
```

“Value Labels” (FORMATS) and Numerical Functions

After the creation and implementation of the “Value Labels”, the SAS output from PROC FREQ by running the SAS program shown on the previous page. Note that the SAS output is very clear to read and understand.

Frequency Tabulation Using PROC FREQ Procedure				
Variable: Injury type				
INJ_TYPE	Frequency	Percent	Cumulative	Cumulative
			Frequency	Percent
Assault (0)	6	3	6	3
Fall (1)	128	64	134	67
Gun Shot Injury (2)	1	0.5	135	67.5
Others (Fall of heavy object, Machine injuries) (3)	14	7	149	74.5
Playing & Sports Related Injuries (4)	10	5	159	79.5
RTA (5)	41	20.5	200	100
Variable: Medical Type				
MED_TYPE	Frequency	Percent	Cumulative	Cumulative
			Frequency	Percent
Neurosurgery Related (0)	98	49	98	49
Orthopedics Related (1)	54	27	152	76
Polytrauma (2)	23	11.5	175	87.5
Surgically Related (3)	25	12.5	200	100
Variable: Severity				
SEVERITY	Frequency	Percent	Cumulative	Cumulative
			Frequency	Percent
Green (Ambulatory Patients) (0)	8	4	8	4
Red (Resuscitation Bay) (1)	30	15	38	19
Yellow (Patients who require observation) (2)	162	81	200	100

Remark: The complete data for 200 subjects are not included in the above SAS program.

Note:

The label for each level of the variable is not only self-explanatory but also contains the numeric code attached to that level.

“Value Labels” (FORMATS) and Numerical Functions

An example of PROC FORMAT implementation in a SAS program

For example, you will be given the following information to create formats.

In a pediatric trauma study, the data for the following variables were collected:

MRN	
AGE	
SEX	0 = Male, 1 = Female
INJ_TYPE	0 = Assault, 1 = Fall, 2 = Gun shot injury, 3 = Others (Fall of heavy objects, machine injuries), 4 = Sports related injury, 5 = RTA
ENVR	0 = Home, 1 = Outdoor
MED_TYPE	0 = Neurosurgery related, 1 = Orthopedics related, 2 = Polytrauma, 3 = Surgically related)
GCS	
SEVERITY	0 = Green (Ambulatory patients), 1 = Red (Resuscitation bay), 2 = Yellow (Patients who require observation)
HOSP_CARE	0 = No, 1 = Yes
LENGTH_STAY	

Variable Names	***Variable Type***	**Column#**	***Label***
MRN	numeric	1-7	Medical Record Number
AGE	numeric	9-10	Age in Months
SEX	numeric	12	Gender
INJ_TYPE	numeric	14	Injury type
ENVR	numeric	16	Environment where trauma occurred
MED_TYPE	numeric	18	Medical Type
GCS	numeric	20-21	Glasgow Coma Score
SEVERITY	numeric	23	Severity
HOSP_CARE	numeric	25	Pre-Hospital Care Given?
LENGTH_STAY	numeric	27-29	Length of stay in hours

Note:

In EXAMPLE14.SAS below, I have included the FORMAT statement (to implement the formats) within PROC FREQ procedure.

“Value Labels” (FORMATS) and Numerical Functions

EXAMPLE14.SAS

```
 options nodate nonumber;

Proc format;
      Value Injuryf
                  0= "Assult (0)"
                  1= "Fall (1)"
                  2= "Gun Shot Injury (2)"
                  3= "Others (Fall of heavy object, Machine injuries)(3)"
                  4= "Playing & Sports Related Injuries (4)"
                  5= "RTA (5)";

      Value Envrf
                  0= "Home (0)"
                  1= "Outdoor (1)";

      Value Sexf
                  1= "Male (1)"
                  0= "Female (0)";

      value Prehospf
                  0= "No (0)"
                  1= "Yes (1)";
      value Medicalf
                  0= "Neurosurgery Related (0)"
                  1= "Orthopedics Related (1)"
                  2= "Polytrauma (2)"
                  3= "Surgically Related (3)";

      value Sevf
                  0= "Green (Ambulatory Patients) (0)"
                  1= "Red (Resuscitation Bay) (1)"
                  2= "Yellow (Patients who require observation) (2)";
run;
data example14;
      input @1 mrn 7.0
                  @9 age 2.0
                  @13 sex 1.0
                  @17 inj_type 1.0
                  @21 envr 1.0
                  @25 med_type 1.0
                  @29 GCS 2.0
                  @33 severity 1.0
                  @37 hosp_care 1.0
                  @41 length_stay 3.1;

label
      MRN = 'Medical Record Number'
      AGE = 'Age in Months'
      SEX = 'Gender'
      INJ_TYPE = 'Injury type'
      ENVR = 'Environment where trauma occurred'
      MED_TYPE = 'Medical Type'
      GCS = 'Glasgow Coma Score'
      SEVERITY = 'Severity'
      HOSP_CARE = 'Pre-Hospital Care Given?'
      LENGTH_STAY = 'Length of stay in hours';
```

“Value Labels” (FORMATS) and Numerical Functions

```
datalines;
1748487     26  0 1     0     1     15    2     1     1.5
1748968     6     0     1     0     0     9     1     1     6
1749228     6     0     1     0     0     15    2     0     6.5
1749449     30    1     1     0     0     14    2     0     3
1749540     24    0     4     1     3     15    2     0     2
1749553     16    0     1     0     0     15    0     0     2
1749631     12    0     3     1     2     15    2     0     1.5
1749644     6     0     3     1     3     15    0     0     0.5
1749696     12    0     1     0     0     10    2     0     3
1749722     16    1     5     1     1     15    2     0     2
1749735     16    0     1     1     0     15    2     0     3
1749774     24    0     3     1     0     15    2     0     3
1749969     12    0     1     0     0     15    2     0     3
1750047     28    0     1     1     3     15    2     1     1
1750073     28    0     5     1     3     15    2     0     0.75
1750229     26    0     1     1     0     13    1     1     5
1750294     6     0     1     0     0     15    2     0     3
1750398     16    0     1     0     2     15    2     0     3
1750424     8     1     1     0     0     15    2     0     2
1750437     10    1     3     1     3     15    2     0     2
1750450     18    1     1     0     0     15    2     0     6.5
1750632     16    1     1     0     0     15    2     0     6.5
1750918     6     0     3     1     0     15    2     0     2
1751178     26    0     1     0     0     12    1     0     2
1751347     6     1     3     1     3     15    2     0     2
1751412     20    0     5     1     2     3     1     0     0.5
1751529     8     0     5     1     2     3     1     0     0.5
1751672     16    0     1     1     0     15    2     0     3
1751971     8     1     4     1     0     15    2     0     5
1752114     30    0     4     1     1     15    2     0     0.5
1752257     12    0     5     1     0     15    2     0     2.5
1752348     30    0     2     1     0     15    2     0     4
1752504     22    0     1     0     0     15    2     0     4
1752517     4     1     1     0     0     15    2     0     6.5
1752920     28    0     3     0     0     12    2     0     3
1752972     18    0     1     0     2     15    2     0     3
1753102     8     0     5     1     1     15    2     0     1.5
1753128     18    0     3     1     3     15    1     0     3
1753219     8     1     1     1     1     15    2     0     1
1753284     26    0     1     0     0     15    2     1     2
1753336     16    0     1     0     1     15    2     0     2
1753349     16    1     1     0     1     15    2     0     3
1753414     8     1     5     1     0     15    2     1     2
1753440     26    0     1     0     0     9     1     1     6
1753466     24    0     1     0     0     15    2     1     3
1753609     26    0     1     0     0     5     1     0     3
1753674     22    0     1     0     0     15    2     0     3
1753739     22    0     5     1     0     9     1     0     3
1753856     14    0     1     0     1     15    2     1     2
1754168     20    0     1     1     1     15    2     0     8
1754402     24    0     1     0     0     15    2     0     3
1754467     28    1     1     0     1     15    2     0     8
1754493     10    0     5     1     3     15    2     0     2
1754636     26    0     0     1     3     15    2     0     2
1754714     12    1     1     0     2     15    1     1     1.5
1754883     10    0     1     0     0     3     1     0     0.5
```

```
1754974      12     0      5      1      0      15     2      0      4
1755052      20     0      5      1      0      3      1      0      0.5
1755091      12     0      1      0      1      15     2      1      2
1755182      22     0      1      0      0      3      1      1      5
1755234      8      0      1      0      0      10     2      1      6
1755247      28     0      1      0      0      15     1      1      4
1755273      8      0      1      0      2      15     2      0      6
1755416      22     0      1      0      1      15     2      0      8
1755494      4      0      1      1      0      15     0      0      6
1755507      24     0      5      1      1      15     2      1      2
1755546      30     0      5      1      2      15     1      1      1.5
1755676      8      1      1      0      3      15     2      1      1
1755715      10     0      5      1      3      15     2      1      1
1755806      24     0      5      1      2      15     1      0      3
1755832      24     0      1      0      1      15     2      1      2
1755845      16     0      1      0      3      15     1      0      3
1756001      10     0      5      1      2      15     1      1      1.5
1756014      8      0      1      0      0      15     2      0      3
1756157      24     0      5      1      0      10     2      1      6
1756300      24     0      1      0      0      15     2      0      2
1756339      22     0      1      0      0      15     2      1      6.5
1760603      20     0      1      0      0      15     2      0      6.5
1763827      8      0      1      0      0      15     2      0      6.5
1765309      12     1      5      1      1      15     2      0      6.5
1765478      4      0      1      0      3      15     2      0      1.5
1765504      24     1      1      0      1      15     2      1      1.5
1765517      12     1      1      0      1      15     0      0      0.5
1765712      12     1      1      0      0      15     2      0      6.5
1765920      10     1      4      1      3      15     2      0      1.5
1766076      28     0      5      1      0      5      1      0      12.5
1766102      20     0      4      0      1      15     2      1      0.5
1766583      22     1      5      1      3      15     2      0      1.5
1766739      12     0      1      0      0      15     2      0      1.5
1766921      30     0      5      1      1      15     2      0      1.5
1767116      14     0      1      0      0      15     2      0      6.5
1767129      18     1      1      0      0      15     2      0      1
1767246      28     0      1      1      3      15     2      0      1.5
1767415      14     1      5      1      1      15     2      0      6.5
1767519      6      0      1      0      0      15     2      0      1.5
1767623      22     0      0      1      3      15     2      0      1
1767688      10     0      1      0      0      15     2      0      6.5
1767727      26     1      1      0      1      15     2      1      0.5
1767779      26     1      5      1      1      15     2      1      0.5
1767857      14     1      1      0      0      15     2      0      6.5
1767883      10     1      1      0      1      15     0      0      0.5
1767896      16     0      1      0      0      15     2      0      2
1767935      24     1      1      0      0      15     2      0      0.5
1767948      2      1      1      0      1      15     2      0      2
1767987      4      0      1      0      0      15     2      0      6.5
1768000      4      0      3      0      3      15     0      1      0.5
1768169      6      1      0      1      3      15     2      0      11.5
1768273      20     1      5      1      2      15     2      0      6.5
1768390      22     0      5      1      1      15     2      0      1.5
1768481      20     0      5      0      2      15     2      0      2.5
1768780      24     0      4      0      1      15     2      0      3
1768910      28     0      0      1      3      15     2      0      1.5
1768975      24     0      1      0      1      15     2      0      2.5
```

“Value Labels” (FORMATS) and Numerical Functions

```
1769105    6    1    1    0    1    15   2    0    4
1769300    12   1    1    1    0    15   2    0    5
1769378    6    1    1    0    1    15   2    0    6
1769482    4    0    5    1    3    15   2    0    5.5
1769612    4    1    0    0    3    15   0    0    0.5
1769716    10   0    1    0    3    15   2    1    1
1769950    12   1    1    0    0    15   2    1    3
1770041    4    1    1    0    3    15   2    1    3.5
1770106    20   0    5    1    1    15   2    0    4
1770301    12   0    4    1    1    15   2    1    4
1770834    6    1    5    1    0    15   2    1    12
1770847    24   0    4    1    1    15   2    0    2
1770860    10   1    1    1    1    15   2    0    1
1770951    6    0    1    1    1    15   2    0    8
1770977    4    1    1    0    1    15   2    1    4
1771120    18   0    4    0    1    15   2    1    4
1771185    10   0    1    0    2    15   2    0    6
1771380    14   0    1    0    0    15   2    0    6.5
1771653    28   0    1    0    0    15   2    1    6.5
1771666    20   0    1    0    2    15   2    1    8
1771822    6    1    1    0    0    15   2    0    6.5
1771835    8    0    1    0    2    15   2    1    8
1771874    16   0    5    1    1    15   2    1    2
1771939    8    0    5    1    1    15   2    0    2
1772017    14   0    4    1    1    15   2    0    3
1772043    24   0    1    0    1    15   2    0    3
1772121    24   0    1    0    1    15   2    0    4
1772238    6    0    1    0    1    15   2    0    4
1772290    18   0    1    0    2    15   2    0    8
1772381    12   1    1    0    0    15   2    0    6.5
1772394    22   0    1    0    0    15   2    0    6.5
1772615    16   1    1    1    1    15   2    0    8
1772745    12   1    1    0    0    15   2    0    3
1772823    4    1    1    0    0    15   2    0    6.5
1773005    8    1    3    1    2    5    1    0    8
1773031    2    1    1    0    0    15   2    0    3
1773096    22   0    1    0    0    15   2    0    6.5
1773109    26   0    1    0    0    15   1    1    4
1773616    26   1    5    1    0    15   2    0    6.5
1773629    18   1    1    0    0    15   2    0    6.5
1773642    16   0    1    0    0    5    1    1    3
1773720    6    0    5    1    1    15   2    0    8
1773993    28   0    1    1    3    15   2    0    1.5
1774201    26   0    3    0    2    15   2    0    3
1774214    24   0    1    0    0    15   2    1    6.5
1774318    14   0    1    0    1    15   2    0    8
1774357    6    0    1    0    0    15   2    1    6
1774383    12   1    1    1    2    15   2    0    8
1774448    14   0    5    1    1    15   2    0    2
1774487    20   0    5    1    0    5    1    0    4
1774500    4    0    1    0    0    5    1    0    3
1774513    28   0    1    1    1    15   2    1    2
1774526    14   1    1    0    1    15   2    0    3
1774617    6    0    1    0    1    9    1    1    3
1774630    8    0    1    0    0    5    1    0    3
1774695    2    0    5    1    0    15   2    0    3.5
1774851    10   0    1    0    0    3    1    0    0.5
```

```
1774877     18    1     1     0     0     15    2     0     6.5
1774955     10    0     1     0     2     15    2     0     8
1774968     10    0     1     0     0     15    2     0     3
1775345     12    0     1     0     0     15    2     0     1.5
1775371     16    1     1     0     1     15    2     1     3
1775514     16    1     1     0     1     15    2     1     2
1775644     20    1     1     0     2     15    1     1     1.5
1775657     10    1     1     0     0     15    2     0     6.5
1775800     8     0     5     1     2     14    2     0     2
1775813     28    0     1     0     0     15    2     1     3
1775826     12    1     0     1     0     10    2     1     2
1775982     2     0     1     1     0     3     1     1     1.5
1776125     4     1     5     1     0     10    2     1     6
1776177     16    0     5     1     0     10    2     0     2
1776203     10    0     1     0     0     10    2     0     2
1776268     18    1     5     1     0     15    2     0     2
1776307     24    0     3     1     0     15    2     1     2
1776385     24    0     3     1     2     15    2     0     2
1776593     16    0     1     1     0     15    2     0     2
1776957     10    0     1     0     0     15    2     0     1.5
1776970     16    0     3     1     1     15    2     1     2
1777282     2     1     1     0     0     15    2     0     2
1777334     26    0     1     0     0     15    2     0     6.5
1777425     24    0     1     0     0     15    2     0     6.5
1777464     6     0     1     0     0     15    2     0     6.5
1777490     8     1     1     1     0     15    2     0     6.5
1777503     20    1     5     1     1     15    1     0     1
1782950     8     1     1     0     1     15    0     0     0.5
1783015     14    0     1     0     0     15    2     0     0.5
1783028     8     0     1     0     0     15    2     0     1.5
;
proc contents data=example14 position;
      title 'Output from the Proc Contents procedure';
run;

proc print data=example14;
      title 'Tabulation of the data from EXAMPL14 data set';
run;

proc freq data=example14;
      title 'Frequency tabulation';
      tables sex inj_type envr med_type severity hosp_care;
      format inj_type injuryf. Envr envrf. Sex sexf. Med_type medicalf.
           severity sevf. Hosp_care prehospf.;
run;
proc means data = example14 n mean stddev stderr maxdec=2;
      title 'Proc Means Output';
      var  age gcs length_stay;
run;
```

Note the use of three SAS procedures PRINT, FREQ and CONTENTS;

Use the **order = freq** option if you want to print values in descending order of frequency. As an example, `proc freq data=example14 order=freq;`

Another example of PROC FORMAT implementation in a SAS program

The following is extracted from the book, “*Applied Statistics and the SAS Programming Language*” by Ron P. Cody and J.K. Smith. Please refer to the textbook for more details.

In this *sample questionnaire*, the data for the following variables will be collected:

ID

AGE (in years)

GENDER (1 = Male, 2 = Female)

RACE (1 = White, 2 = African American, 3 = Hispanic, 4 = Other)

MARITAL STATUS (1 = Single, 2 = Married, 3 = Widowed, 4 = Divorced)

EDUCATION (1 = HS or less, 2 = Two-year college, 3 = Four-year college
4 = Post graduate degree)
Q6 (The President has been doing a good job)

Q7 (The budget should be increased)

Q8 (There should be more federal aid to big cities)
Q6 – Q8 have the following codes:
1 = Strongly disagree, 2 = Disagree, 3 = Neutral
4 = Agree, 5 = Strongly agree

Variable Names	*Variable Type*	*Column#*	*Label*
ID	numeric	1-3	Subject ID
AGE	numeric	5-7	Subjects age in years
GENDER	character	9	Subjects gender
RACE	character	11	Subjects race
MARITAL	character	13	Subjects marital status
EDUC	character	15	Subjects education level
Q6	numeric	17	President is doing a good job
Q7	numeric	19	Budget should be increased
Q8	numeric	21	Give more federal aid to big cities

“Value Labels” (FORMATS) and Numerical Functions

EXAMPLE14A.SAS

```
options nodate nonumber;
proc format;
      value $genderf    '1'= 'Male (1)'
                        '2' = 'Female (2)';

      value $racef      '1' = 'White (1)'
                        '2' = 'African American (2)'
                        '3' = 'Hispanics (3)'
                        '4' = 'Other (4)';

      value $marital    '1' = 'Single (1)'
                        '2' = 'Married (2)'
                        '3' = 'Widowed (3)'
                        '4' = 'Divorced (4)';

      value $educf      '1' = 'High School or Less (1)'
                        '2' = 'Two year college (2)'
                        '3' = 'Four year college (3)'
                        '4' = 'Post graduate degree (4)';

      value q6_q8f      1 = 'Strongly disagree (1)'
                        2 = 'Disagree (2)'
                        3 = 'Neutral (3)'
                        4 = 'Agree (4)'
                        5 = 'Strongly agree (5)';
data example14a;
      input @1 id 3.0
            @5 age 3.0
            @9 gender $1.
            @11 race $1.
            @13 marital $1.
            @15 educ $1.
            @17 q6 1.0
            @19 q7 1.0
            @21 q8 1.0;

      label
            ID = 'Subjects ID'
            AGE = 'Subjects age in years'
            GENDER = 'Subjects gender'
            RACE = 'Subjects race'
            MARITAL = 'Subjects marital status'
            EDUC = 'Subjects education level'
            Q6 = 'President is doing a good job'
            Q7 = 'Arms budget should be increased'
            Q8 = 'Give more federal aid to big cities';

      format
            gender $genderf. race $racef. marital $marital. educ $educf.
            q6 q7 q8 q6_q8f.;

datalines;
001 009 1 1 1 1 2 3 2
002 045 2 2 2 2 4 2 2
003 063 2 3 4 3 4 4 5
```

“Value Labels” (FORMATS) and Numerical Functions

```
004 056 1 3 2 2 5 3 3
005 061 1 1 3 2 1 4 5
;

proc print data=example14a;
      title 'Tabulation of the data from EXAMPL14 data set';
run;

proc freq data=example14a;
      title 'Frequency tabulation';
      tables gender race marital educ q6 q7 q8;
run;

proc contents data=example14a position;
      title 'Output from the Proc Contents procedure';
run;
```

**** Note the use of three SAS procedures PRINT, FREQ and CONTENTS.**

"Value Labels" (FORMATS) and Numerical Functions

Using Numerical Functions:

Some of the numerical functions that help to do mathematical calculations are **ROUND, LOG, LOG10, INT, MIN, MAX, MEAN, SUM, N, CEIL, FLOOR, RANNOR, RANUNI, LAG, DIF** and **SQRT**.

Brief Description of the Numerical Functions

- ✓ **ROUND** rounds the variable value to the nearest rounding unit.

  For example, **ROUND(12.345, 0.01) = 12.35** (rounds the value of 12.345 to the nearest hundredth).

- ✓ **LOG**(*var*) gives the natural logarithm of ***var***.

- ✓ **LOG10**(*var*) gives the common logarithm of ***var***.

- ✓ **INT**(*var*) gives the INTEGER part of ***var***.

  For example, **INT(-3.4) = -3**

- ✓ **MOD**(*var1, var2*) gives the REMAINDER when the first of two arguments is divided by the second.

  For example, **MOD(28, 5) = 3**

- ✓ **CEIL**(*var*) rounds ***var*** to the next largest integer.

  For example, **CEIL(8.4) = 9, CEIL(-3.2) = -3**

- ✓ **FLOOR**(*var*) rounds ***var*** to the next smallest integer.

  For example, **FLOOR(8.4) = 8, FLOOR(-3.2) = -4**

- ✓ **RANUNI**(*seed*) generates a random number uniformly distributed between 0 and 1.

  SEED is optional. If entered, it must be **0** or **any number of your choice**.

  SEED = 0 (or negative) specifies that the function RANUNI use the time clock to generate a random seed to initiate the random number sequence. It generates a different random series each time it is run unless the program is run exactly at the same time every day.

  If you want to generate the same series of random numbers every time you run the program, then use any other number as the seed.

- ✓ **RANNOR**(*seed*) generates a normally distributed random number. **SEED** is optional. If entered, it must be 0 **or** any number of your choice.

  Follow the same rule as in the case of RANUNI regarding the seed value.

“Value Labels” (FORMATS) and Numerical Functions

The following is a SAS program that uses a few of the above numerical functions on the variables ***X***, ***Y***, and ***Z.*** **EXAMPLE15.SAS**

```
options linesize=64 pagesize=55 nodate nonumber;
      LINESIZE(LS)sets the number of characters that will print across the
      width of the page. It can be any number from 64 through 256.

      PAGESIZE(PS) sets the number of lines that will print down the length
      of the page. It can be any number from 20 through 200.

      LS=64 and PS=55 works well for an 8.5 by 11 inch sheet of paper

      NOCENTER option aligns the output at the left;

data example15;
input x y z;                  * x, y, and z are the three variables;

round_x = round(x, .1);       * x rounded to the nearest tenth;

ln_x = log(x);                * Natural log of x;

b10_x = log10(x);             * Log of x to base 10;

intp_x = int(x);              * The integer part of x;

minimum = min(of x y z);      * Minimum value of x, y, & z;

maximum = max(of x y z);      * Maximum value of x, y, & z;

average = mean(of x y z);     * Average of NONMISSING values of x, y, & z;

sum = sum(of x y z);          * Sum of NONMISSING values of x, y, and z;

nonmiss = n(of x y z);        * Number of nonmissing values of x, y, and z;

***************************************************************************
* The following 2 functions are useful for computing LAGS and DIFFERENCES of
  series.;
            lagvalueofz = lag(z);
            difvalueofz = dif(z);
***************************************************************************;
datalines;
4 5 6
4.5 3.4 2.8
1 2 3
4 . 6
2.33 5 5
2.5 2.6 2.7
;
proc print data=example15;
      title 'Tabulation of data from EXAMPLE15 data set';
run;
```

SET Statement

We talked about this SET statement once before in my prior lectures

Note the use of **SET** statement in the next program.

Suppose we have an existing SAS data set. We want to use this existing SAS data set as a basis to create a new SAS data set. This can be accomplished by using the **SET** statement.

The *SET* statement is used in a *DATA* step to refer to another SAS data set in the program.

The observations in the original data set are evaluated by the new data set created using the **SET** statement (*Note* that the original data set still exists and can be used by SAS).

Syntax:

```
data new; * This statement creates a temporary SAS
                 data set by name NEW;
    set one; * ONE is the original SAS data set;

proc print data=new;
run;
```

Conversion of Character DATA to Numeric data

```
******************************************************************;
* The following program converts character data to numeric data;
******************************************************************;
```

EXAMPLE16.SAS

```
data one;
      input age $ height $ weight $;
datalines;
24 69 161
23 68 150
39 69 189
55 70 180
27 . 155
;
proc means data=one;
var age height weight;
run;        * These 3 statements do not produce the MEANS. WHY?;
******************************************************************;
data two;
      set one;
nage=age*1;
nheight=height*1;
nweight=weight*1;
run;

proc print data=two;
      title 'Tabulation of data from NUMERIC data set';
run;
******************************************************************;
proc means data= two;
      var nage nheight nweight;
run;          * These 3 statements DO produce the MEANS. WHY?;

*******************************************************************;
*    This part of the syntax will be discussed after introducing the
     concept of ARRAYs;

* Use of ARRAY statements to do the same thing as above;

data three;
      set one;

      array ex16a{3} age -- weight;
      array ex16b{3} nage nheight nweight;
            do i = 1 to 3;
                  ex16b{i} = 1*ex16a{i};
            end;
      drop age height weight i;

proc means data=three;
run;
*******************************************************************;
```

Conversion of Character DATE values to Numeric data

```
******************************************************************;
The following program converts character DATE values to numeric data
so that the data set can be SORTED by the DATE;
******************************************************************;
```

EXAMPLE16A.SAS

```
data one;
      input date $10.;
datalines;
12/10/2014
10/12/2006
02/03/2015
01/07/2007
;
proc sort data=one;
by date;
run;

Proc print data=one;
Run;
* The dates are entered as CHARACTER values. If the data set is sorted
  By the variable, DATE, it will not be sorted.

* Convert this character date values into numeric date values as follows;

data two;
      set one;
newdate = input (date, mmddyy10.);

proc sort data=two;
      by newdate;
run;

proc print data = two;
      var date;
run;
```