# SAS for Beginners Exercises

**kamakshaiah Musunuru**
Associate Professor – Business Analytics
Dhruva College of Management
kamakshaiah.m@gmail.com

## I. Entering and printing a variable

### 1. printing a variable

```
input x;
cards;
1
2
3
4
; proc print data=work;
var x;
run;
```

### 2. data manipulation

```
data work;
input x y;
z=x+y;
cards;
1 2
2 3
3 4
4 5
; proc print data=work;
run;
```

## II. SAS constants, variables and expressions

**SAS variables are either *numeric* or *character* type.**

**Range list is special type of variable with characters and numbers.**

**Missing value are denoted with period (.) for numbers, blank () for characters. There are arthmetic operators, comparison operators, and logical operators.**

**There are variety of functions available in SAS, for example; sin, cos, log(base e), exp and etc. For non-negative square root of x, we can use `sqrt(x)`. For probability distributions probnorm(x) calculates N(0,1) distribution function at x; probit(x) calculates inverse of the same.**

**1.**

```
data ;
name='tom';
cards;
proc print;
run;
```

**2. data input from the program: Suppose each observation consists of three variables,y=weight,x1= height, andx2= average number of calories consumed daily. Suppose we have four observations given by**

```
160 66 400
152 70 500
180 72 4500
240 68 7000

data example;
input y x1 x2;
cards;
160 66  400
152 70  500
180 72 4500
240 68 7000
proc print;
run;
```

**3. Using if else in data input;**

```
data numbers;
input x 1-2 y 3-4 z 5-7;
if x=1 then y=0;
else z=0;
datalines;
1 2 3
2 3 4
3 4 5
4 5 6
5 6 7
6 7 8
7 8 9
8 9 10
9 10 11
10 11 12;
proc print data=numbers;
run;
```

4. How to change charater or number length;

```
data numbers;
input x 1-3 y $4-9 z $10-17;
if x=1 then y='stupid';
else z='silent';
datalines;
1  kama  3
2  rama  4
3  karma 5
4  lemma 6
5  gumma 7
6  sita  8
7  zita  9
8  tita  10
9  gita  11
10 gita  12;
proc print data=numbers;
run;
```

5. how to append data set;

```
data one;
infile 'E:\sas_work\mat.txt';
input x 1-2 y 3-4 country $ 5-17;
datalines;
data two;
input x 1-2 y 3-4 country $ 5-17;
datalines;
```

```
3 4 poland, 22
data three;
input x 1-2 y 3-4 country $ 5-17;
datalines;
5 6 swis, 34
data four;
set one two three;
proc print data=four;
run;
```

6. Subsetting a data set;

```
data one;
infile 'E:\sas_work\mat.txt';
input x 1-2 y 3-4 country $ 5-17;
datalines;
data two;
set one;
if y=2;
data three;
set one;
if y=4;
data four;
set two three;

proc print data=four;
run;
```

7. How to sort variables

```
data sort;
infile 'E:\sas_work\sort.txt';
input x 1-3 y 4-6 z 7;
datalines;
data sort1;
infile 'E:\sas_work\sort1.txt';
input x 1-3 y 4-6 z 7;
datalines;
data concat;
set sort sort1;
proc print data=concat;
by z;
run;
```

8. merge horizontally

```
data sort;
infile 'E:\sas_work\sort.txt';
input x 1-3 y 4-6 z 7;
datalines;
data add;
input w;
datalines;
11
12
13
14
data concat;
merge sort add;
proc print data=concat;
run;
```

9. how to use 'do' while creating a data set in SAS

```
data one;
do i=1 to 100;
x=3/(i-1)*10;
y=1/(x**2);
output one;
end;
drop i;
cards;

proc print data=one;
run;


data one;
do i=1 to 100;
x=i;
y=1/x;
z=x**2;
p=sqrt(x);
output one;
end;
drop i;
cards;

proc print data=one noobs;
run;
```

10. How to save and retrieve data sets from SAS editor.

```
libname storage 'E:\sas_work';
data storage.doexample;
input x y z;
cards;
1 2 3
4 5 6
7 8 9
run;

libname storage 'E:\sas_work';
proc print data=storage.doexample;
run;

libname storage 'E:\sas_work';
proc contents data=storage.doexample;
run;
```

11. IF, ELSE

```
data one;
input x $ y;
if x eq 'blue' then z=1;
else if x eq 'red' then z=1;
else z=0;
cards;
blue 21
red  32
yell 12
proc print data=one;
run;
```

12. Do & IF, ELSE

```
data one;
input x;
```

```
if x gt 5 then
do;
y=1/x;
z=x**2;
end;
else do;
y=0;
z=0;
end;
cards;
1
2
3
4
5
6
7
8
9
10
proc print data=one;
run;
```

## III. Descriptive Statistics

### 13. How to make tables

```
data one;
do i=1 to 20;
x=i;
y=1/x;
z=x**2;
p=sqrt(x);
output one;
end;
drop i;
cards;
proc print data=one;
run;
proc freq data=one;
tables x;
run;

proc freq data=one;
tables y;
run;

proc freq data=one;
tables z;
run;

proc freq data=one;
tables p;
run;
```

### 14. How to make cross tables

**Example – 1**

```
data one;
do i=1 to 20;
x=i;
y=1/x;
```

```
z=x**2;
p=sqrt(x);
output one;
end;
drop i;
cards;
proc print data=one;
run;
proc freq data=one;
tables x*y;
run;
```

**Example – 2**

```
data auto;
input make $1-7 price 8-12 MPG 13-15 rep78 16-17 foreign 18;
cards;
AMC     4099 22 3 0
AMC     4749 17 3 0
AMC     3799 22 3 0
Audi    9690 17 5 1
Audi    6295 23 3 1
BMW     9735 25 4 1
Buick   4816 20 3 0
Buick   7827 15 4 0
Buick   5788 18 3 0
Buick   4453 26 3 0
Buick   5189 20 3 0
Buick  10372 16 3 0
Buick   4082 19 3 0
Cad.   11385 14 3 0
Cad.   14500 14 2 0
Cad.   15906 21 3 0
Chev.   3299 29 3 0
Chev.   5705 16 4 0
Chev.   4504 22 3 0
Chev.   5104 22 2 0
Chev.   3667 24 2 0
Chev.   3955 19 3 0
Datsun 6229 23 4 1
Datsun 4589 35 5 1
Datsun 5079 24 4 1
Datsun 8129 21 4 1
;
proc print data=auto;
run;

proc freq data=auto;
tables rep78*foreign;
run;
```

### 14.1. To suppress the result (percentages)

```
data auto;
input make $1-7 price 8-12 MPG 13-15 rep78 16-17 foreign 18;
cards;
AMC     4099 22 3 0
AMC     4749 17 3 0
AMC     3799 22 3 0
Audi    9690 17 5 1
Audi    6295 23 3 1
BMW     9735 25 4 1
Buick   4816 20 3 0
Buick   7827 15 4 0
```

```
Buick  5788 18 3 0
Buick  4453 26 3 0
Buick  5189 20 3 0
Buick 10372 16 3 0
Buick  4082 19 3 0
Cad.  11385 14 3 0
Cad.  14500 14 2 0
Cad.  15906 21 3 0
Chev.  3299 29 3 0
Chev.  5705 16 4 0
Chev.  4504 22 3 0
Chev.  5104 22 2 0
Chev.  3667 24 2 0
Chev.  3955 19 3 0
Datsun 6229 23 4 1
Datsun 4589 35 5 1
Datsun 5079 24 4 1
Datsun 8129 21 4 1
;
proc print data=auto;
run;
proc freq data=auto;
tables rep78*foreign \ norow nocol nofreq;
run;
```

15. **Calculating** means

```
data auto;
input make $1-7 price 8-12 MPG 13-15 rep78 16-17 foreign 18;
cards;
AMC    4099 22 3 0
AMC    4749 17 3 0
AMC    3799 22 3 0
Audi   9690 17 5 1
Audi   6295 23 3 1
BMW    9735 25 4 1
Buick  4816 20 3 0
Buick  7827 15 4 0
Buick  5788 18 3 0
Buick  4453 26 3 0
Buick  5189 20 3 0
Buick 10372 16 3 0
Buick  4082 19 3 0
Cad.  11385 14 3 0
Cad.  14500 14 2 0
Cad.  15906 21 3 0
Chev.  3299 29 3 0
Chev.  5705 16 4 0
Chev.  4504 22 3 0
Chev.  5104 22 2 0
Chev.  3667 24 2 0
Chev.  3955 19 3 0
Datsun 6229 23 4 1
Datsun 4589 35 5 1
Datsun 5079 24 4 1
Datsun 8129 21 4 1
;
proc means data=auto;
var price rep78;
run;
```

16. data in classess

```
data auto;
```

```
input make $1-7 price 8-12 MPG 13-15 rep78 16-17 foreign 18;
cards;
AMC    4099 22 3 0
AMC    4749 17 3 0
AMC    3799 22 3 0
Audi   9690 17 5 1
Audi   6295 23 3 1
BMW    9735 25 4 1
Buick  4816 20 3 0
Buick  7827 15 4 0
Buick  5788 18 3 0
Buick  4453 26 3 0
Buick  5189 20 3 0
Buick 10372 16 3 0
Buick  4082 19 3 0
Cad.  11385 14 3 0
Cad.  14500 14 2 0
Cad.  15906 21 3 0
Chev.  3299 29 3 0
Chev.  5705 16 4 0
Chev.  4504 22 3 0
Chev.  5104 22 2 0
Chev.  3667 24 2 0
Chev.  3955 19 3 0
Datsun 6229 23 4 1
Datsun 4589 35 5 1
Datsun 5079 24 4 1
Datsun 8129 21 4 1
;
proc means data=auto;
class foreign;
var MPG;
run;
```

**UNIVARIATE ANALYSIS**

**17. To get** univariate **details**

```
data auto;
input make $1-7 price 8-12 MPG 13-15 rep78 16-17 foreign 18;
cards;
AMC    4099 22 3 0
AMC    4749 17 3 0
AMC    3799 22 3 0
Audi   9690 17 5 1
Audi   6295 23 3 1
BMW    9735 25 4 1
Buick  4816 20 3 0
Buick  7827 15 4 0
Buick  5788 18 3 0
Buick  4453 26 3 0
Buick  5189 20 3 0
Buick 10372 16 3 0
Buick  4082 19 3 0
Cad.  11385 14 3 0
Cad.  14500 14 2 0
Cad.  15906 21 3 0
Chev.  3299 29 3 0
Chev.  5705 16 4 0
Chev.  4504 22 3 0
Chev.  5104 22 2 0
Chev.  3667 24 2 0
Chev.  3955 19 3 0
```

```
Datsun 6229 23 4 1
Datsun 4589 35 5 1
Datsun 5079 24 4 1
Datsun 8129 21 4 1
;
proc univariate data=auto;
var MPG;
run;
```

### 17.1. To get univariate summary in classes (foreign)

```
data auto;
input make $1-7 price 8-12 MPG 13-15 rep78 16-17 foreign 18;
cards;
AMC    4099 22 3 0
AMC    4749 17 3 0
AMC    3799 22 3 0
Audi   9690 17 5 1
Audi   6295 23 3 1
BMW    9735 25 4 1
Buick  4816 20 3 0
Buick  7827 15 4 0
Buick  5788 18 3 0
Buick  4453 26 3 0
Buick  5189 20 3 0
Buick 10372 16 3 0
Buick  4082 19 3 0
Cad.  11385 14 3 0
Cad.  14500 14 2 0
Cad.  15906 21 3 0
Chev.  3299 29 3 0
Chev.  5705 16 4 0
Chev.  4504 22 3 0
Chev.  5104 22 2 0
Chev.  3667 24 2 0
Chev.  3955 19 3 0
Datsun 6229 23 4 1
Datsun 4589 35 5 1
Datsun 5079 24 4 1
Datsun 8129 21 4 1
;
proc univariate data=auto;
class foreign;
var MPG;
run;
```

## 18. T test

```
data auto;
input make $1-7 price 8-12 MPG 13-15 rep78 16-17 foreign 18;
cards;
AMC    4099 22 3 0
AMC    4749 17 3 0
AMC    3799 22 3 0
Audi   9690 17 5 1
Audi   6295 23 3 1
BMW    9735 25 4 1
Buick  4816 20 3 0
Buick  7827 15 4 0
Buick  5788 18 3 0
Buick  4453 26 3 0
Buick  5189 20 3 0
```

```
Buick 10372 16 3 0
Buick  4082 19 3 0
Cad.  11385 14 3 0
Cad.  14500 14 2 0
Cad.  15906 21 3 0
Chev.  3299 29 3 0
Chev.  5705 16 4 0
Chev.  4504 22 3 0
Chev.  5104 22 2 0
Chev.  3667 24 2 0
Chev.  3955 19 3 0
Datsun 6229 23 4 1
Datsun 4589 35 5 1
Datsun 5079 24 4 1
Datsun 8129 21 4 1
;
proc ttest data=auto;
class foreign;
var MPG;
run;
```

**OUTPUT**

**The TTEST Procedure**

Statistics

| Variable | foreign | N | Lower CL Mean | Mean | Upper CL Mean | Lower CL Std Dev | Std Dev | Upper CL Std Dev | Std Err |
|---|---|---|---|---|---|---|---|---|---|
| MPG | 0 | 19 | 17.844 | 19.789 | 21.735 | 3.0494 | 4.0357 | 5.968 | 0.9258 |
| MPG | 1 | 7 | 18.906 | 24 | 29.094 | 3.549 | 5.5076 | 12.128 | 2.0817 |
| MPG | Diff (1-2) | | -8.271 | -4.211 | -0.15 | 3.4743 | 4.4495 | 6.19 | 1.9673 |

T-Tests

| Variable | Method | Variances | DF | t Value | Pr > \|t\| |
|---|---|---|---|---|---|
| MPG | Pooled | Equal | 24 | -2.14 | 0.0427 |
| MPG | Satterthwaite | Unequal | 8.5 | -1.85 | 0.0996 |

Equality of Variances

| Variable | Method | Num DF | Den DF | F Value | Pr > F |
|---|---|---|---|---|---|
| MPG | Folded F | 6 | 18 | 1.86 | 0.2862 |

**From the above output, it is clear that p-value for pooled variance is not significant (Ho accepted). Hence, we accept the case of equal variances, i.e. The p-value is < 0.05, hence, null hypothesis is rejected.** The difference is significant.

## 19. Chi-square test

```
data auto;
input make $1-7 price 8-12 MPG 13-15 rep78 16-17 foreign 18;
cards;
AMC    4099 22 3 0
AMC    4749 17 3 0
AMC    3799 22 3 0
Audi   9690 17 5 1
Audi   6295 23 3 1
BMW    9735 25 4 1
Buick  4816 20 3 0
Buick  7827 15 4 0
Buick  5788 18 3 0
Buick  4453 26 3 0
```

```
Buick  5189 20 3 0
Buick 10372 16 3 0
Buick  4082 19 3 0
Cad.  11385 14 3 0
Cad.  14500 14 2 0
Cad.  15906 21 3 0
Chev.  3299 29 3 0
Chev.  5705 16 4 0
Chev.  4504 22 3 0
Chev.  5104 22 2 0
Chev.  3667 24 2 0
Chev.  3955 19 3 0
Datsun 6229 23 4 1
Datsun 4589 35 5 1
Datsun 5079 24 4 1
Datsun 8129 21 4 1
;
proc freq data=auto;
tables rep78*foreign / CHISQ EXACT ;
run;
```