

Create a Multiplication Table with SAS, R and Python

Hengwei Liu, Blue Hyacinth Consulting, LLC

Abstract

In this short paper, we will create a multiplication table with three tools: SAS, R and Python.

In SAS we will use a SAS macro to do it. In R we will do it through a matrix. In Python we will use a matrix through the numpy module. In all three programs we will need to do some loops and some numeric to character variable conversion. This is a good topic to illustrate some basic functions of the three programming languages.

Introduction

We want to create the multiplication table shown in table 1. This is not a complicated task. We will do it using three tools: SAS, R and Python. In all three programs we need to do some looping. Also, we will need to convert some numeric variables to character variables.

```
1X1=1
2X1=2  2X2=4
3X1=3  3X2=6  3X3=9
4X1=4  4X2=8  4X3=12  4X4=16
5X1=5  5X2=10  5X3=15  5X4=20  5X5=25
6X1=6  6X2=12  6X3=18  6X4=24  6X5=30  6X6=36
7X1=7  7X2=14  7X3=21  7X4=28  7X5=35  7X6=42  7X7=49
8X1=8  8X2=16  8X3=24  8X4=32  8X5=40  8X6=48  8X7=56  8X8=64
9X1=9  9X2=18  9X3=27  9X4=36  9X5=45  9X6=54  9X7=63  9X8=72  9X9=81
```

Table 1: a multiplication table

Details

First we show how to do it in a SAS macro. We use macro command %do to handle the outer loop, but use the do command in data step to handle the inner loop. We create nine datasets and merge them into one dataset.

```
%macro multi_table;

%do i=1 %to 9;
data a&i;
do k=&i to 9;
```

```

z=k*&i;
col&i=compress(put(k, best.)||"x&i="||put(z,best.));
output;
end;

%if &i=1 %then %do;
data final; set a1;
%end;
%else %do;
data final;
merge final a&i;
by k;
run;
%end;

%end;

data final(keep=col:); set final;
run;

%mend;

%multi_table;

proc print data=final;
run;

```

The output is shown in table 2.

Obs	col1	col2	col3	col4	col5	col6	col7	col8	col9
1	1x1=1								
2	2x1=2	2x2=4							
3	3x1=3	3x2=6	3x3=9						
4	4x1=4	4x2=8	4x3=12	4x4=16					
5	5x1=5	5x2=10	5x3=15	5x4=20	5x5=25				
6	6x1=6	6x2=12	6x3=18	6x4=24	6x5=30	6x6=36			
7	7x1=7	7x2=14	7x3=21	7x4=28	7x5=35	7x6=42	7x7=49		
8	8x1=8	8x2=16	8x3=24	8x4=32	8x5=40	8x6=48	8x7=56	8x8=64	
9	9x1=9	9x2=18	9x3=27	9x4=36	9x5=45	9x6=54	9x7=63	9x8=72	9x9=81

Table 2: a multiplication table created with SAS

Now we show how to do it in R programming. We use the `as.character` to convert numeric value to character, use `paste` function to combine multiple character variables and use `gsub` to remove white space.

```
col <- matrix(nrow=9,ncol=9)

for (k in 1:9) {
  for (i in k:9) {
    z[i] <- as.character(i*k)
    str1 <- as.character(i)
    str2 <- 'X'
    str3 <- as.character(k)
    str4 <- '='
    col[i,k] <- paste(str1, str2, str3, str4, z[i])
    col[i,k] <- gsub(" ", "", col[i,k])
  }
}

print(col, quote=FALSE, na.print = "")
```

The output is shown in table 3.

```
> print(col, quote=FALSE, na.print = "")
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
[1,] 1X1=1
[2,] 2X1=2 2X2=4
[3,] 3X1=3 3X2=6 3X3=9
[4,] 4X1=4 4X2=8 4X3=12 4X4=16
[5,] 5X1=5 5X2=10 5X3=15 5X4=20 5X5=25
[6,] 6X1=6 6X2=12 6X3=18 6X4=24 6X5=30 6X6=36
[7,] 7X1=7 7X2=14 7X3=21 7X4=28 7X5=35 7X6=42 7X7=49
[8,] 8X1=8 8X2=16 8X3=24 8X4=32 8X5=40 8X6=48 8X7=56 8X8=64
[9,] 9X1=9 9X2=18 9X3=27 9X4=36 9X5=45 9X6=54 9X7=63 9X8=72 9X9=81
```

Table 3: a multiplication table created with R

Finally, we show how to do it in Python programming. We need the numpy module to handle matrix. First we define a 9 by 9 empty matrix. We use str function to convert numeric value to character. We use the tabulate module to print out the matrix nicely.

```
from tabulate import tabulate
import numpy as np

mystr=np.empty((9,9), dtype='S6')

for i in range(1,10):
    for n in range(i,10):
        z=n*(i)
        mystr[n-1,i-1]=str(n)+'X'+str(i)+'=' + str(z)

    if i != n:
        mystr[i-1, n-1]=" "

print(tabulate(mystr))
```

The output is shown in table 4.

```
C:\Users\hengw\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\hengw\PycharmProjects\pythonProject\main.py
-----
1X1=1
2X1=2  2X2=4
3X1=3  3X2=6  3X3=9
4X1=4  4X2=8  4X3=12  4X4=16
5X1=5  5X2=10  5X3=15  5X4=20  5X5=25
6X1=6  6X2=12  6X3=18  6X4=24  6X5=30  6X6=36
7X1=7  7X2=14  7X3=21  7X4=28  7X5=35  7X6=42  7X7=49
8X1=8  8X2=16  8X3=24  8X4=32  8X5=40  8X6=48  8X7=56  8X8=64
9X1=9  9X2=18  9X3=27  9X4=36  9X5=45  9X6=54  9X7=63  9X8=72  9X9=81
-----
Process finished with exit code 0
```

Table 4: a multiplication table created with Python

Conclusion

In this paper we have shown how to use SAS, R and Python to create a multiplication table. They all have the capability to get it done in a fairly short program.

Contact

Hengwei Liu

Hengwei_liu@yahoo.com

www.bluehyacinthconsulting.com