STAT 342 - Introduction to Statistical Computing and Exploratory Data Analysis

Name: Jackson Yuan Student ID: 301387501

Introduction

This report is aimed at introducing how to write **PROC TABULATE** and **PROC SGPLOT** with some samples and complete code in SAS.

PROC TABULATE:

Introduction:

This function is a procedure that can describe, summary, add and delete some statistics data from too complex dataset to make it understanded. It will save time and reduce mistakes when we process a large dataset. `PROC TABULATE` is similar as `PROC MEAN`, `PROC FREQ` which is compute the statistics from the dataset and organize them by creating a new table.

Now, the full syntax of **PROC TABULATE** contains **BY, CLASS, CLASSLEV, FREQ, KEYLABEL, KEYWORD, TABLE, VAR, WEIGHT**. The most important part is **CLASS, VAR, TABLE**.

The full structure of `PROC TABULATE`

```
PROC TABULATE data = my_data <option>
    BY <Descending> variable 1, 2, 3;
    CLASS class-variables 1, 2, 3;
    CLASSLEV Variables 1, 2, 3;
    FREQ variable;
    KEYLABEL Variables 1, 2, 3;
    KEYWORD Variables 1, 2, 3;
    TABLE <page-expression>, <row-expression>, <column-expression>;
    VAR analysis-variables 1, 2, 3;
    Weight variable;
```

At first, the table 1.1 contains ..... It will show ..... after we process the data. Now we need some specific values from the dataset.

Before introducing Tabulate detailly, we need to know some concept of its keywords: Firstly, **Figure 1.1** shows Row, Row heading, Column, Column heading and cell when we process the dataset by PROC TABULATE and see its result. It is important for users to identify which variables users should put them into somewhere and what users want to generate. Secondly, **Figure 1.2**show Table Dimensions and Category.

The most important components

1. **CLASS statement:** Variables following **CLASS** named `class variables`, which determine how many categories it will have. For example, we have `3` variables in `CLASS` statement, then we will have 3 categories in result.
2. **Var statement:** Variables after **VAR** are `analysis variables` which are what users want to analyze. Note: Var variables only can accept numeric value, so if users want to select character values, it must be change into numeric value at first
3. **Table statement:** determine the structure of the result table and only column parameter is necessary, but it will show more dimensions if users add more variables for contracts and all of these variables need to be mentioned in **CLASS** or **VAR statement**. **TABLE** will only support at most 3 dimensions and all of them can be expressed by dimension expression, which is consist of operators and elements.

- If has `three expression` then the result will have three dimensions(page, row, column)
- If has `two expression` then the result will have two dimensions(row, column)
- If has `one expression` then the result will have one dimensions(column)

The real data in Practice with comments

Note: The data is retrieved from the UCL Student performance data (https://archive.ics.uci.edu/ml/datasets/Student+Performance)

```
/*Read the data and follow the format variables*/
/*See the result table on Table 1.1*/
proc import out=bank datafile="/home/u63591328/bank-full.csv"
dbms=csv replace;
    getnames=YES;
    delimiter=';';
    format contact $9. default $5. education $11. housing $5.
job $14. loan $5. marital $10.
        month $5. poutcome $9. y $4.;
run;
/*Generate the information of the dataset and identify the
types of variables*/
/*See the result table on Table 1.2*/
PROC Contents data = bank;
run;

/*Example 1 */
/*See the result table on Table 1.3*/
PROC TABULATE data=bank;
CLASS marital age;
TABLE marital * age, N; /*This is two dimension, row and
column*/
RUN;
/** According to Table 1.3, we can see each age on different
type of marital condition from the table. This is just first
step, users could use other statement to make a range of the
age to reduce the rows(observations)
*/

/*Example 2 */
```

```
/*See the result table on Table 1.3*/
PROC TABULATE data=bank;
VAR age balance;/*Note: age and balance is numeric value*/
CLASS marital;
TABLE (age balance)*(N MAX MIN MEAN), marital; /*'*'and '()' is operator, N MAX MIN MEAN and age balance is elements*/
/** According to Table 1.4, we can easily to see the result about the N MAX MIN MEAN of age and balance in the different marital condition.
We found VAR variables can imply the row dimension(2*4)
And CLASS variables can imply column dimension(1*3) which is how many category we have
*/

/*Example 3*/
/*See the result table on Table 1.4*/
PROC TABULATE data=bank;
VAR age balance;
CLASS marital education;
TABLE (age balance)*(N MAX MIN MEAN), marital,education;/*This is three dimension*/
/** According to Table 1.4, we can see N MAX MIN MEAN of age and balance(analysis variable) in different marital condition depending on different level of education.
The table is 2*4 because the page dimension is depend on '(age balance)*(N MAX MIN MEAN), so we can easily change the order to get a set of different tables
*/
/*Now let us change the order of TABLE statement to see what happened here on Table 1.5*/
/*We have three type of marital condition: single, divorce, married*/
PROC TABULATE data=bank;
VAR age balance;
CLASS marital education;
TABLE marital,education,(age balance)*(N MAX MIN MEAN);/*Note: The TABLE variables should be mentioned at CLASS or VAR statement*/
/**
We can see a less number of tables but get same result from the dataset, so it is smart to identify which variables can make less, simplier and easy to understand tables.
```

```
51  */
```

Other Optional Statement:

1. **BY:** it will sort the observations by `BY` variables in descending order
2. **CLASSLEV:** It will give a specific style for variables in CLASS statement
3. **FREQ:** It will give the frequency of the variables
4. **KEYLABEL:** a keyword for the duration of the PROC TABULATE step. It only process the last one in the step
5. **KEYWORD:** only affect HTML, RTF, Printer output
6. **Weight:** It will calculate the weight of value for analysis variables

PROC SGPLOT

Introduction

`PROC SGPLOT` is a SAS procedure which can draw many analysis graphics, which is a part of `ODS Graphic System`. `PROC SGPLOT` provides a simple method to create many graphics, including histograms, scatter plots, box plots, line graphs, pie charts, etc. It is easy to write that users just only give the variables as x-axis and y-axis and the plot type, then SAS will generate which plot users desire.

> ODS Graphic System

- ODS represents for `Output Delivery System`
- It is default to turn `On` when users are in Microsoft/Unix environment.

Four types of plots that users can create

1. **Basic Plots :** scatter, series, step, band, needle, and vector plots
2. **Fit and confidence plots:** loess, regression, and penalized B-spline curves, and ellipses
3. **Distribution plots:** box plots, histograms, and normal and kernel density estimates
4. **Categorization plots:** dot plots, bar charts, and line plots

The full structure of `PROC SGPLOT`

```
PROC SGPLOT < option(s)>;
BAND X= variable | Y= variable
UPPER= numeric-value | numeric-variable LOWER= numeric-value |
numeric-variable
</option(s)>;
DENSITY response-variable </option(s)>;
DOT category-variable </option(s)>;
ELLIPSE X= numeric-variable Y= numeric-variable </option(s)>;
HBAR category-variable </option(s)>;
HBOX response-variable </option(s)>;
HISTOGRAM response-variable </option(s)>;
HLINE category-variable </option(s)>;
INSET "text-string-1" <... "text-string-n"> | (label-list);
KEYLEGEND <"name-1" ... "name-n"> </option(s)>;
LOESS X= numeric-variable Y= numeric-variable </option(s)>;
NEEDLE X= variable Y= numeric-variable </option(s)>;
PBSPLINE X= numeric-variable Y= numeric-variable </option(s)>;
REFLINE value(s) </option(s)>;
REG X= numeric-variable Y= numeric-variable </option(s)>;
SCATTER X= variable Y= variable </option(s)>;
SERIES X= variable Y= variable </option(s)>;
STEP X= variable Y= variable </option(s)>;
VBAR category-variable </option(s)>;
VBOX response-variable </option(s)>;
VECTOR X= numeric-variable Y= numeric-variable </option(s)>;
VLINE category-variable </option(s)>;
XAXIS <option(s)>;
X2AXIS <option(s)>;
YAXIS <option(s)>;
Y2AXIS <option(s)>;
```

Bar Chart

```
PROC SGPLOT data = my_data<option>
    vbar <VARIABLE_NAME>
run;
```

- Users can make a vertical `vbar` or a horizontal `hbar` bar chart
- Can create a stack or cluster bar chart

- SAS assume `frequency` as default, but can change it into `MEAN` and `SUM`, etc.

The real data in Practice with comments

Note: The data is retrieved from the UCL Student performance data (https://archive.ics.uci.edu/ml/datasets/Student+Performance)

```
/*Example 1 HISTAGRAM PLOT*/
/*See the Figure 2.1*/
PROC SGPLOT data=bank;
    Histogram age/scale=count
    nbins=30;
    Density age;/*It will give a normal density curve*/
    Density age / type=kernel;/*It will give a kernel density curve*/
run;
/*Example 2 SCATTER PLOT*/
/*See the Figure 2.2*/
PROC SGPLOT data=bank;
    scatter x = age y = marital;
    ellipse x = age y = marital;/*It will give a ellipse where points are gathering at*/
run;
/*Example 3 BOX PLOT*/
/*See the Figure 2.3*/
PROC SGPLOT data=bank;
    vbox balance/ category=marital;
run;
/*Example 4 Bar chart*/
/*See the Figure 2.4*/
PROC SGPLOT data=bank;
    vbar age/group=marital groupdisplay = cluster;/*Stack and Cluster for the bar chart*/
run;
```

REFERENCES

1. PROC TABULATE: PROC TABULATE Statement (sas.com)
2. PROC SGPLOT: The SGPLOT Procedure (sas.com)
3. (40) PROC TABULATE: SAS for Beginners (Lesson 12) - YouTube
4. https://blog.csdn.net/weixin_49282401/article/details/118093238
5. https://archive.ics.uci.edu/ml/datasets/Student+Performance