Chapter 2 Data Summary in SAS

In this chapter, we will learn four procedures for summarizing variables – PROC MEANS, PROC UNIVARIATE, PROC FREQ, and PROC STANDARD.

2.1 PROC MEANS

The MEANS procedure serves as a data summarization tool to compute descriptive statistics.

- Mean, standard deviation, confidential interval for mean
- Quantiles, including median
- Identify extreme values

Use dataset *Blood.txt* to see the procedure. Here is the information about the variables.

Variable	Label
Subject	Subject ID
Gender	Gender (F or M)
BloodType	Blood type (A, B, O, or AB)
AgeGroup	Age group (Young or Old)
WBC	White blood cells
RBC	Red blood cells
Chol	Cholesterol

First, we import the data using a DATA step.

```
*Import data using INFILE*;
data sasdata.blood;
  infile "/folders/myfolders/Datasets/blood.txt";
  input Subject Gender$ BloodType$ AgeGroup$ WBC RBC Chol;
  label BloodType = "Blood type"
        AgeGroup = "Age group"
        WBC = "White blood cells"
        RBC = "Red blood cells"
        Chol = "Cholesterol";
run;
```

Second, we get descriptive statistics using the PROC MEANS.

```
*PROC MEANS procedure -- descriptive statistics*;
proc means data=sasdata.blood;
run;
```

SAS output table

Jue Wang EPS 704 Chapter 2

The MEANS Procedure

Variable	Label	N	Mean	Std Dev	Minimum	Maximum
Subject		1000	500.5000000	288.8194361	1.0000000	1000.00
WBC	White blood cells	908	7042.97	1003.37	4070.00	10550.00
RBC	Red blood cells	916	5.4835262	0.9841158	1.7100000	8.7500000
Chol	Cholesterol	795	201.4352201	49.8867157	17.0000000	331.0000000

a. PROC MEANS options and VAR statement

By default, it provides the number of valid responses (N), mean, standard deviation, minimum, and maximum for all of the numeric variables.

- We can compute the statistics for specific variables using statement VAR.
- We can also specify the statistics that we want in particular.

Here is a list of commonly used PROC MEANS options

PROC MEANS option	**Statistic produced**
N	Number of non-missing values
NMISS	Number of missing values
MEAN	Arithmetic mean
SUM	Sum of the values
MIN	Minimum value
MAX	Maximum value
MEDIAN	Median value
STD	Standard deviation
VAR	Variance
CLM	95% confidence interval for the mean
Q1	Value of the first quartile (25th percentile)
Q3	Value of the third quartile (75th percentile)
QRANGE	Interquartile range ($IQR = Q3 - Q1$)

Note: Besides these statistics, the option MAXDEC=*value* is often used to specify decimal places to be printed in the output table. For example,

```
*PROC MEANS procedure -- use VAR statement and request specific statistics*;
proc means data=sasdata.blood n nmiss clm mean median Q1 Q3 maxdec=2;
   var RBC WBC;
run;
```

SAS output table (only printed RBC and WBC; 2 decimal places)

The MEANS Procedure

Variable	Label	N	N Miss	Lower 95% CL for Mean	Upper 95% CL for Mean	Mean	Median	Lower Quartile	Upper Quartile
RBC	Red blood cells	916	84	5.42	5.55	5.48	5.52	4.84	6.11
WBC	White blood cells	908	92	6977.62	7108.32	7042.97	7040.00	6375.00	7710.00

Note: If you simply want to calculate the descriptive statistics but do not want to print them, use NOPRINT option can suppress the printing.

b. CLASS and BY statements

These two statements work in a similar way – specifies a grouping variable for which summary statistics are produced separately for the subjects in different groups. Let's look at CLASS first.

```
*PROC MEANS procedure -- use VAR statement and request specific statistics*;
proc means data=sasdata.blood n nmiss clm mean median Q1 Q3 maxdec=2;
   class gender;
   var RBC WBC;
run;
```

SAS output table

The MEANS Procedure

Gender	N Obs	Variable	Label	N	N Miss	Lower 95% CL for Mean	Upper 95% CL for Mean	Mean	Median	Lower Quartile	Upper Quartile
Female	440	RBC	Red blood cells	409	31	5.40	5.59	5.50	5.55	4.89	6.14
		WBC	White blood cells	403	37	7014.72	7210.15	7112.43	7150.00	6460.00	7800.00
Male	560	RBC	Red blood cells	507	53	5.39	5.56	5.47	5.48	4.79	6.09
		WBC	White blood cells	505	55	6899.65	7075.44	6987.54	6930.00	6350.00	7680.00

If using the BY statement, the output tables are organized in a different way. Also, the data must be sorted first by Gender (the BY variable) using PROC SORT. Data do not need to be sorted if using CLASS statement.

```
*Sort the data by Gender*;
proc sort data=sasdata.blood out=blood_sort;
  by gender;
run;

*PROC MEANS procedure -- BY statement*;
proc means data=blood_sort n nmiss clm mean median Q1 Q3 maxdec=2;
   var RBC WBC;
   by gender;
run;
```

Note: The order of statements in PROC MEANS does not matter.

SAS output tables (produced separate tables for female and male)

The MEANS Procedure

Gender=Female

Variable	Label	N	N Miss	Lower 95% CL for Mean	Upper 95% CL for Mean	Mean	Median	Lower Quartile	Upper Quartile
RBC	Red blood cells	409	31	5.40	5.59	5.50	5.55	4.89	6.14
WBC	White blood cells	403	37	7014.72	7210.15	7112.43	7150.00	6460.00	7800.00

Gender=Male

Variable	Label	N	N Miss	Lower 95% CL for Mean	Upper 95% CL for Mean	Mean	Median	Lower Quartile	Upper Quartile
RBC	Red blood cells	507	53	5.39	5.56	5.47	5.48	4.79	6.09
WBC	White blood cells	505	55	6899.65	7075.44	6987.54	6930.00	6350.00	7680.00

c. OUTPUT statement

The OUTPUT statement puts the computed summary statistics in another dataset. For example

```
*PROC MEANS procedure -- OUTPUT statement*;
proc means data=sasdata.blood n nmiss clm mean median Q1 Q3 maxdec=2;
   class gender;
   var RBC;
   output out=out_RBC mean=mean_RBC std=sd_RBC;
run;
```

Now, check the OUTPUT DATA (not the RESULTS) to see the out_RBC dataset. This dataset is stored in the WORK library (temporary).

2.2 PROC UNIVARIATE

This procedure provides a variety of summary statistics for each variable. For example,

```
*PROC UNIVARIATE procedure*;
proc univariate data=sasdata.blood;
  var RBC WBC Chol;
run;
```

Partial SAS output tables

The UNIVARIATE Procedure
Variable: RBC (Red blood cells)

Moments			
N	916	Sum Weights	916
Mean	5.4835262	Sum Observations	5022.91
Std Deviation	0.98411576	Variance	0.96848384
Skewness	-0.0221357	Kurtosis	0.01809726
Uncorrected SS	28429.4213	Corrected SS	886.16271
Coeff Variation	17.9467687	Std Error Mean	0.0325161

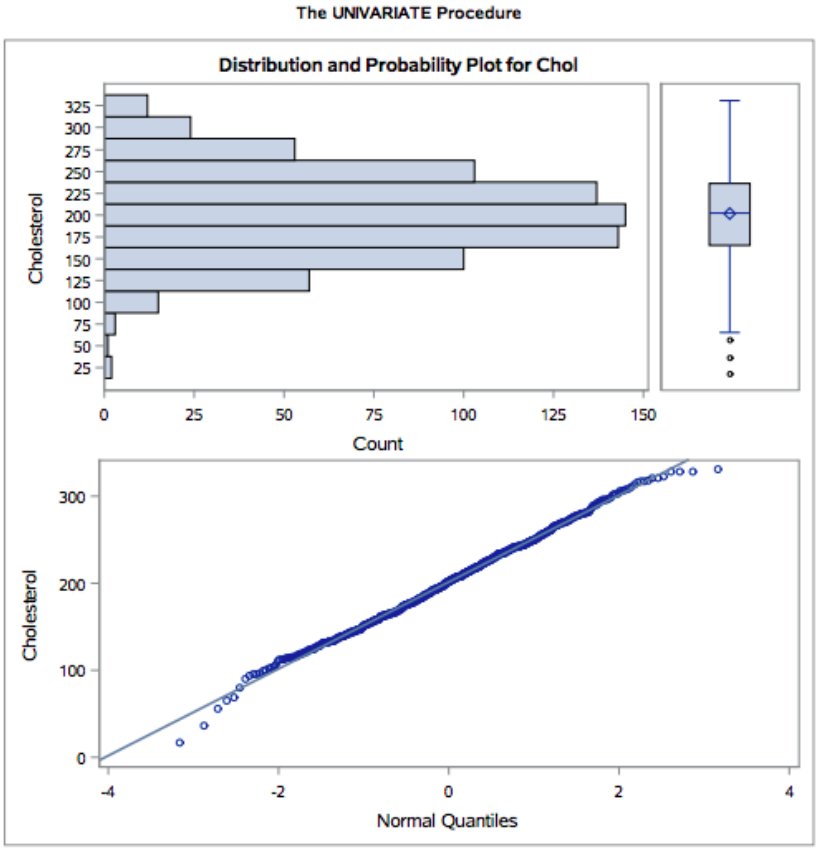
Basic Statistical Measures			
Location		Variability	
Mean	5.483526	Std Deviation	0.98412
Median	5.520000	Variance	0.96848
Mode	5.410000	Range	7.04000
		Interquartile Range	1.27000

Note: The complete list of output tables is not shown here to save space. Please check them out in your SAS. The CLASS statement works the same way in the UNIVARIATE procedure.

A nice feature of this procedure is that we can generate some plots, such as histogram, boxplot, and normal probability plot. To do so, we simply add the PLOTS option to PROC UNIVARIATE.

```
*PROC UNIVARIATE procedure -- plots*;
proc univariate data=sasdata.blood plots;
  var Chol;
run;
```

SAS output figures



2.3 PROC FREQ

This procedure can be used to count frequency, percent, cumulative frequency, and cumulative percent in one-way, two-way, and three-way tables.

a. The TABLES statement: specify variables that will be summarized

- One-way table: provides frequency measures for each variable separately. For example

```
*FREQ procedure -- simple use showing proportions*;
proc freq data=sasdata.blood;
  tables Gender BloodType AgeGroup;
run;
```

SAS output tables

The FREQ Procedure

Gender	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Female	440	44.00	440	44.00
Male	560	56.00	1000	100.00

Blood type

BloodType	Frequency	Percent	Cumulative Frequency	Cumulative Percent
A	412	41.20	412	41.20
AB	44	4.40	456	45.60
B	96	9.60	552	55.20
O	448	44.80	1000	100.00

Age group

AgeGroup	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Old	598	59.80	598	59.80
Young	402	40.20	1000	100.00

- Create a two-way table using * between two variables, e.g., Gender by Blood Type.

```
*FREQ procedure -- 2-way table*;
proc freq data=sasdata.blood;
  tables Gender*BloodType;
run;
```

SAS output table

The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of Gender by BloodType

Gender	BloodType(Blood type) A	AB	B	O	Total
Female	178 17.80 40.45 43.20	20 2.00 4.55 45.45	34 3.40 7.73 35.42	208 20.80 47.27 46.43	440 44.00
Male	234 23.40 41.79 56.80	24 2.40 4.29 54.55	62 6.20 11.07 64.58	240 24.00 42.86 53.57	560 56.00
Total	412 41.20	44 4.40	96 9.60	448 44.80	1000 100.00

Note: SAS reads Row variable (Gender) * Column variable (BloodType). You can transpose the 2-way table by specifying BloodType*Gender.

- Extension I: Create a three-way table Gender by Blood Type by Age Group.

```
*FREQ procedure -- 3-way table*;
proc freq data=sasdata.blood;
  tables Gender*BloodType*AgeGroup;
run;
```

SAS output tables

The FREQ Procedure

Frequency Percent Row Pct Col Pct	Table 1 of BloodType by AgeGroup Controlling for Gender=Female		
BloodType(Blood type)	AgeGroup(Age group)		
	Old	Young	Total
A	110 25.00 61.80 42.64	68 15.45 38.20 37.36	178 40.45
AB	11 2.50 55.00 4.26	9 2.05 45.00 4.95	20 4.55
B	18 4.09 52.94 6.98	16 3.64 47.06 8.79	34 7.73
O	119 27.05 57.21 46.12	89 20.23 42.79 48.90	208 47.27
Total	258 58.64	182 41.36	440 100.00

Frequency Percent Row Pct Col Pct	Table 2 of BloodType by AgeGroup Controlling for Gender=Male		
BloodType(Blood type)	AgeGroup(Age group)		
	Old	Young	Total
A	143 25.54 61.11 42.06	91 16.25 38.89 41.36	234 41.79
AB	15 2.68 62.50 4.41	9 1.61 37.50 4.09	24 4.29
B	41 7.32 66.13 12.06	21 3.75 33.87 9.55	62 11.07
O	141 25.18 58.75 41.47	99 17.68 41.25 45.00	240 42.86
Total	340 60.71	220 39.29	560 100.00

Note: 1st variable (separate tables)*2nd variable (rows)*3rd variable (columns).

- Extension II: Can create multiple tables

```
*FREQ procedure -- Multiple 2-way tables*;
proc freq data=sasdata.blood;
  tables Gender*BloodType Gender*AgeGroup;
run;
```

SAS output tables

The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of Gender by BloodType

Gender	BloodType(Blood type) A	AB	B	O	Total
Female	178 17.80 40.45 43.20	20 2.00 4.55 45.45	34 3.40 7.73 35.42	208 20.80 47.27 46.43	440 44.00
Male	234 23.40 41.79 56.80	24 2.40 4.29 54.55	62 6.20 11.07 64.58	240 24.00 42.86 53.57	560 56.00
Total	412 41.20	44 4.40	96 9.60	448 44.80	1000 100.00

Frequency
Percent
Row Pct
Col Pct

Table of Gender by AgeGroup

Gender	AgeGroup(Age group) Old	Young	Total
Female	258 25.80 58.64 43.14	182 18.20 41.36 45.27	440 44.00
Male	340 34.00 60.71 56.86	220 22.00 39.29 54.73	560 56.00
Total	598 59.80	402 40.20	1000 100.00

- Options in PROC FREQ line – NOPRINT (function the same as in PROC MEANS: suppress the results being printed)

- Options in TABLES statement
  - **NOFREQ**: do not display frequency
  - **NOPERCENT**: do not display percent
  - **NOROW**: do not display row percentage
  - **NOCOL**: do no display column percentage
  - **CHISQ**: performs Pearson chi-square test (See below)
    - **EXPECTED**: expected counts in a cell
  - **FISHER**: conducts Fisher's exact test (https://en.wikipedia.org/wiki/Fisher%27s_exact_test)

Supplemental material: Pearson's Chi-Square Test

- Purpose: Examine independence between two categorical variables
- Hypothesis
  - Null hypothesis: two variables are independent.
  - Alternative hypothesis: two variables are dependent/related.
- Test statistic:

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{\left(O_{ij} - E_{ij}\right)^2}{E_{ij}}$$

where
i refers to row number, r denotes total number of rows;
j refers to column number, c denotes total number of columns;
O_{ij} = observed counts in cell (i, j);
E_{ij} = expected counts in cell $(i, j) = \frac{(ith\ row\ total)*(jth\ column\ total)}{sample\ size}$;

- Under null hypothesis, the test statistic follows a chi-square distribution with $(r-1)*(c-1)$ degrees of freedom. For more information, consider taking EPS700 Quantitative Methods I.

Example

```
*FREQ procedure -- 2-way table and Pearson chi-square test*;
proc freq data=sasdata.blood;
  tables Gender*BloodType /norow nocol nopercent chisq expected;
run;
```

SAS output table

The FREQ Procedure

Frequency
Expected

Table of Gender by BloodType

Gender	BloodType(Blood type) A	AB	B	O	Total
Female	178 181.28	20 19.36	34 42.24	208 197.12	440
Male	234 230.72	24 24.64	62 53.76	240 250.88	560
Total	412	44	96	448	1000

Statistics for Table of Gender by BloodType

Statistic	DF	Value	Prob
Chi-Square	3	4.0865	0.2523
Likelihood Ratio Chi-Square	3	4.1389	0.2469
Mantel-Haenszel Chi-Square	1	0.5828	0.4452
Phi Coefficient		0.0639	
Contingency Coefficient		0.0638	
Cramer's V		0.0639	

Sample Size = 1000

b. The OUTPUT statement (function the same as in PROC MEANS)

It puts the computed statistics in another dataset.

- OUT= defines the name and path of the dataset
- CHISQ option stores the chi-square test results in the output dataset. Run the following codes and see what you get.

```
*FREQ procedure -- OUTPUT statement*;
proc freq data=sasdata.blood noprint;
  tables Gender*BloodType / chisq;
  output out=blood_chisq chisq;
run;
```

Note: Here are what you can expect. NOPRINT – no output table will be printed; OUTPUT – an output dataset will be generated. Check the label of each column in the output dataset.

c. The WEIGHT statement – if your dataset is a frequency table instead of raw data responses.

```
*FREQ procedure -- WEIGHT statement*;
data blood_freq;
  input Gender$ BloodType$ frequency;
  datalines;
  Female A  178
  Female AB 20
  Female B  34
  Female O  208
  Male   A  234
  Male   AB 24
  Male   B  62
  Male   O  240
  ;
run;

proc freq data=blood_freq;
  tables Gender*BloodType / chisq;
  weight frequency;
run;
```

Note: This will give you the exact output tables as above. Try it without using the WEIGHT statement and see the differences.

Exercise: Let's do an exercise together. Import dataset *survey.txt*.

Variable	Label	Variable	Label
ID	Subject ID	Q1	The governor doing a good job
Gender	Gender (M or F)	Q2	The property tax should be lowered
Age	Age as of 01/01/2006	Q3	Guns should be banned
Salary	Yearly salary	Q4	Expand the Green Acre program
		Q5	The school needs to be expanded

Note: For Q1-Q5, Likert scale was used – 1, strongly disagree; 2, disagree; 3, neutral; 4, agree; 5, strongly agree.

DATA step: First, import data from .txt file.

```
data sasdata.survey;
  infile "/folders/myfolders/Datasets/survey.txt";
  input ID Gender$ Age Salary Q1-Q5;
run;
```

PROC FREQ: Next, conduct Pearson's chi-square test to examine the independence between Gender and Q3.

```
proc freq data=sasdata.survey;
  tables Gender*Q3 / norow nocum nopercent chisq;
run;
```

SAS output tables

The FREQ Procedure

Table of Gender by Q3

Frequency Col Pct	Q3					
Gender	1	2	3	4	5	Total
F	0 0.00	2 66.67	0 0.00	1 100.00	0 0.00	3
M	1 100.00	1 33.33	1 100.00	0 0.00	1 100.00	4
Total	1	3	1	1	1	7

Statistics for Table of Gender by Q3

Statistic	DF	Value	Prob
Chi-Square	4	4.2778	0.3697
Likelihood Ratio Chi-Square	4	5.7416	0.2193
Mantel-Haenszel Chi-Square	1	0.0062	0.9370
Phi Coefficient		0.7817	
Contingency Coefficient		0.6159	
Cramer's V		0.7817	

WARNING: 100% of the cells have expected counts less than 5. Chi-Square may not be a valid test.

Sample Size = 7

Conclusion: Preference towards whether or not guns should be banned is independent of gender.

2.4 PROC STANDARD

This procedure is used to standardize the variables.

Supplement material: Standardization

Purpose: Standardization is the process of putting different variables on the same scale. This process allows you to compare scores between different types of variables.

Procedure: Let X_i denotes the i_{th} observation of variable X in a random sample. Sample mean of this variable is $\bar{X}$, and let s denote sample standard deviation. The standardization procedure can be described as

$$Z_i = \frac{X_i - \bar{X}}{s}$$

where $i = 1, 2, \dots, n$ and n is the sample size.

After standardization, the new variable Z_i has a mean of 0 and a standard deviation of 1.

- No output will be created. Therefore, we use OUT= to specify a dataset for saving the standardized variables.
- We can define a theoretical mean (other than zero) for centering and any meaning unit (instead of 1) as the new standard deviation. Therefore, in PROC STANDARD, we need to define the mean and standard deviation that we want for the standardized/new variable.

Example (Create standardized RBC and WBC values to Z scores)

```
*STANDARD procedure*;
proc standard data=sasdata.blood out=standard_blood mean=0 std=1;
   var RBC WBC;
run;
```

Check the output dataset standard_blood in the OUTPUT DATA window. Also, let's use PROC MEANS to check the mean and standard deviation of the new RBC and WBC variables.

Before standardization	After standardization
Before using PROC STANDARD; title1 "Before using PROC STANDARD"; proc means data=sasdata.blood mean std; var RBC WBC; run;	*After using PROC STANDARD*; title1 "After using PROC STANDARD"; proc means data=standard_blood mean std; var RBC WBC; run;

SAS output tables

Before using PROC STANDARD

The MEANS Procedure

Variable	Label	Mean	Std Dev
RBC	Red blood cells	5.4835262	0.9841158
WBC	White blood cells	7042.97	1003.37

After using PROC STANDARD

The MEANS Procedure

Variable	Label	Mean	Std Dev
RBC	Red blood cells	5.098542E-15	1.0000000
WBC	White blood cells	9.32624E-17	1.0000000