PSP1 Homework

Before starting the homework, check to confirm that Program 3 represents PSP1.

Develop the program specified on pages 3-10 as Program 3, using the PSP1 process. To follow the PSP1 process, make use of the process scripts available on Process dashboard. Before starting Program 3, review the top-level PSP1 process script.

Use your Program 2 (from PSP0.1 Homework) as the base code to which you may add, delete from or modify code to create Program 3. The program should work correctly in a linux environment.

Record the following information on the Process Dashboard, as described in the process scripts (refer to process scripts for more information):

- A conceptual design of the program to identify the parts to be added to the base program
- Use the Size Estimating Template and the PROBE wizard to estimate the added and modified size as well as the estimated time for Program 3 (in the Planning phase).
- Time taken for each PSP1 phase.
- Details of all the defects found/fixed in each phase.
- Completed Test Report template (in the Test phase).
- Actual Program 3 size data using the LOC Counter available on Process Dashboard (in Postmortem phase).
- Process problems and solutions in the PIP form (in Postmortem phase).

Remember to mark the completion of each phase and the Project overall on the Dashboard before saving your data for submission.

Submit the following in a zip file (using the submission page on Blackboard) by the due date and time:

- PSP data from Process Dashboard retrieved using the save data backup function.
- Conceptual design of the program
- Completed Test Report
- Completed PIP form
- Your well documented C program as a text file with .c extension.
- Input data file used for testing the program.
- All LOC counter outputs used.
- A document showing the program test results including inputs/outputs and run time messages as evidence that the program works correctly.

An important note:

Programming environments include a wide variety of mathematical and other embedded functions. While using embedded functions for some program actions is appropriate, they should not be used to write the principal functions of the exercise program. When students use such functions, they can write many of the PSP exercises in fewer than ten LOC. This takes too little time and is of no value in a PSP exercise which is focused on the process of program development. If students use these functions, they will not be fulfilling the purpose of the exercise and so will not receive credit for doing the homework.

Program 3 Specification

Program 3 requirements

Program 3 requirements

Using PSP1, write a program in C to calculate relative size ranges for very small, small, medium, large, and very large ranges using standard deviation. The program should accept input data from a file.

Thoroughly test the program. Test the program using the data provided in tables 1 and 2. Expected values are included in table 3.

Class Name	Class LOC	Number of Methods
each_char	18	3
string_read	18	3
single_character	25	3
each_line	31	3
single_char	37	3
string_builder	82	5
string_manager	82	4
list_clump	87	4
list_clip	89	4
string_decrementer	230	10
Char	85	3
Character	87	3
Converter	558	10

Table 1. LOC/Method Data

Chapter	Pages
Preface	7
Chapter 1	12
Chapter 2	10
Chapter 3	12
Chapter 4	10
Chapter 5	12
Chapter 6	12
Chapter 7	12
Chapter 8	12
Chapter 9	8
Appendix A	8
Appendix B	8
Appendix C	20
Appendix D	14
Appendix E	18
Appendix F	12

Table 2. Pgs/Chapter

Continued on next page

Program 3 requirements, continued

	VS	S	M	L	VL
LOC/Method	4.3953	8.5081	16.4696	31.8811	61.7137
Pgs/Chapter	6.3375	8.4393	11.2381	14.9650	19.9280

Table 3. Expected Values

Relative Size Table

Using relative size tables in the PSP

In the PSP, relative size tables are used to give you a framework for judging the size of new parts in your planned products. For example, if you know the sizes of all previously developed parts of a certain type, you can then better judge the likely size of a new part of that type. The standard deviation procedure described in the following section allows you to balance your esimates so they more or less conform to the normal distribution.

The medium range (M) is the area from -0.5 standard deviations to +0.5 standard deviations from the mean, as shown in Figure 1. Assuming that the data approximates a normal distribution, the likely number of parts that are within plus or minus 0.5 standard deviation of the average value is 38.3 percent. Following similar logic, the range percentages area are as follows:

- 6.68 % should be very small
- 24.17% should be small
- 38.2% should be medium
- 24.17% should be large
- 6.68% should be very large

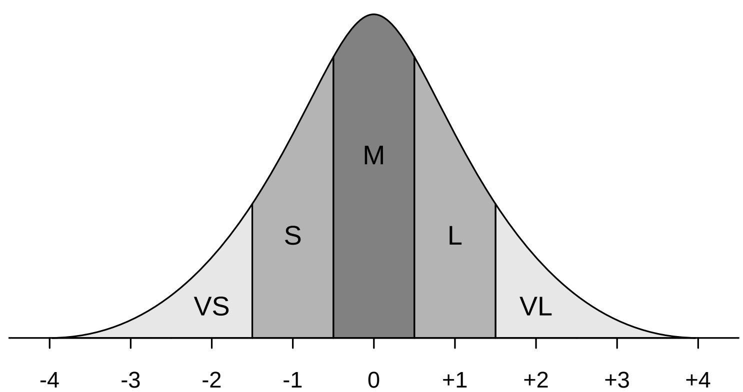

Figure 1. Ranges of standard deviations

Calculating a relative size table using standard deviation

The PROBE estimating method divides historical size data into categories that represent your kind of work. One way of doing this is based on standard deviation. First, divide your historical data into functional categories that each have at least 6 to 8 members (calculation, text, and data, for example). For each category you can then calculate the relative size ranges for VS, S, M, L, and VL following the below procedure.

1. Divide the part sizes by the number of items in each part to determine size per item, if applicable. For instance, you may not have enough data on classes to develop a relative size table, but you do have sufficient data on methods. Instead of using total LOC per class, you can use LOC/method.

Continued on next page

Calculating a relative size table using standard deviation, continued

2. Next, you'll need to log-normally transform the data. This is necessary because you cannot have negative sizes and the smaller values tend to bunch up. Log-normally transforming the data allows you plot the data around a mean of zero. For each size value, x_i, take the natural logarithm, ln, to give ln (x_i).
3. Calculate the average of these *n* logarithmic values: $avg = \frac{\sum_{i=1}^{n} \ln(x_i)}{n}$.
4. Calculate the variance of these values: $\text{var} = \sigma^2 = \frac{\sum_{i=1}^{n} \left(\ln(x_i) - avg\right)^2}{(n-1)}$.
5. Calculate the standard deviation: $\sigma = \sqrt{\text{var}}$.
6. Calculate the logarithmic ranges:

   ln(VS) = $avg - 2\sigma$

   ln(S) = $avg - \sigma$

   ln(M) = avg

   ln(L) = $avg + \sigma$

   ln(VL) = $avg + 2\sigma$
7. Last, convert the natural log values back to their original form by calculating the anti-logarithm (calculate *e* to the power of the log value) to get the midpoints of the size ranges

   VS = $e^{\ln(VS)}$
   S = $e^{\ln(S)}$
   M = $e^{\ln(M)}$
   L = $e^{\ln(L)}$
   VL = $e^{\ln(VL)}$

Example of calculating a relative size table

Example of calculating a relative size table

In this example, we'll calculate the relative size ranges for very small, small, medium, large, and very large ranges using standard deviation for the data in the table below.

Class Name	Class LOC	Number of Methods	LOC/method
each_char	18	3	6.0000
string_read	18	3	6.0000
single_character	25	3	8.3333
each_line	31	3	10.3333
single_char	37	3	12.3333
string_builder	82	5	16.4000
string_manager	82	4	20.5000
list_clump	87	4	21.7500
list_clip	89	4	22.2500
string_decrementer	230	10	23.0000
Char	85	3	28.3333
Character	87	3	29.0000
Converter	558	10	55.8000

1. Divide the part sizes by the number of items in each part to determine size per item, if applicable. In this instance, the LOC/method is calculated for each class.
2. For each size value, x_i, calculate the natural logarithm, ln, to give ln(x_i).

Class Name	LOC/method	ln(x_i)
each_char	6.0000	1.7918
string_read	6.0000	1.7918
single_character	8.3333	2.1203
each_line	10.3333	2.3354
single_char	12.3333	2.5123
string_builder	16.4000	2.7973
string_manager	20.5000	3.0204
list_clump	21.7500	3.0796
list_clip	22.2500	3.1023
string_decrementer	23.0000	3.1355
Char	28.3333	3.3440
Character	29.0000	3.3673
Converter	55.8000	4.0218
Total		36.4197

Continued on next page

Example of calculating a relative size table, continued

3. Calculate the average of these n logarithmic values:

$$avg = \frac{\sum_{i=1}^{n} \ln(x_i)}{n} = \frac{36.4197}{13} = 2.8015 .$$

4. Calculate the variance of these values:

$$\mathrm{var} = \sigma^2 = \frac{\sum_{i=1}^{n} \left(\ln(x_i) - avg\right)^2}{(n-1)} = \frac{5.2350}{12} = 0.4363$$

Class Name	LOC/method	*ln*(x_i)	(*ln*(x_i)-*avg*)^2
each_char	6.0000	1.7918	1.0196
string_read	6.0000	1.7918	1.0196
single_character	8.3333	2.1203	0.4641
each_line	10.3333	2.3354	0.2173
single_char	12.3333	2.5123	0.0836
string_builder	16.4000	2.7973	0.0000
string_manager	20.5000	3.0204	0.0479
list_clump	21.7500	3.0796	0.0773
list_clip	22.2500	3.1023	0.0905
string_decrementer	23.0000	3.1355	0.1115
Char	28.3333	3.3440	0.2943
Character	29.0000	3.3673	0.3201
Converter	55.8000	4.0218	1.4890
Total		36.4197	5.2350

5. Calculate the standard deviation: $\sigma = \sqrt{\mathrm{var}} = \sqrt{0.4363} = 0.6605$

6. Calculate the logarithmic ranges:

   ln(VS) = $avg - 2\sigma = 2.8015 - 1.3210 = 1.4805$

   ln(S) = $avg - \sigma = 2.8015 - 0.6605 = 2.1410$

   ln(M) = $avg = 2.8015$

   ln(L) = $avg + \sigma = 2.8015 - 0.6605 = 3.4620$

   ln(VL) = $avg + 2\sigma = 2.8015 - 1.3210 = 4.1225$

Continued on next page

Example of calculating a relative size table, continued

7. Convert the natural log values back to their original form by calculating the anti-logarithm by calculating e to the power of the log value to determine the midpoints of the size ranges:
   VS = $e^{\ln(VS)} = e^{1.4805} = 4.3953$

   S = $e^{\ln(S)} = e^{2.1410} = 8.5081$

   M = $e^{\ln(M)} = e^{2.8015} = 16.4696$

   L = $e^{\ln(L)} = e^{3.4620} = 31.8811$

   VL = $e^{\ln(VL)} = e^{4.1225} = 61.7137$