**Lab sheet 1 (Week 2: Wed. 7th October 2020):**

**Presenting, understanding and analysing data**

**QUESTION:** What is the purpose of this lab sheet? Where does it fit into the general scheme of the module and why is it useful for you?

**ANSWER:** In the CS3003 coursework distributed later in the term, you will be asked to carry out analysis of data from a system or number of systems. The techniques you use in this lab sheet and others are **exactly** the types of analysis that would be useful for that. So, by completing this worksheet you will be preparing yourself for the coursework challenge as well as hopefully learning something about software which may be useful in an exam setting too.

Aside from the analysis in a CS3003 coursework context, the techniques described may also be useful in your FYP in the same way; finally, every Software Engineer must be able to abstract information from data and understand what characterises their code. We can gain great insights into our code if we can collect data from it and then plot that data in some way. As Lord Kelvin (1824-1907) said*: “If you cannot measure it, you cannot improve it.”*

AI techniques will tell you clever things about data, but standard data analyses, as presented in this sheet, are still highly useful for exploring data.

**NOTE: We have used Excel in this example. You’re free to use any other statistics tool if you want (e.g., R or SPSS or any tool you want).**

**1. Summarising data**

To start visualising data, you need first to identify the purpose of the graph and then determine which types of graphs to use. After drawing the graphs, you need to be sure that the graph is and well-labelled. Different types of charts can be used to visualize data, including but not limited to bar charts, pie charts, histograms and boxplots.

**EXERCISE 1.** Use the data from Table 1 (next page) which describes some extracted data from a system and copy it into Excel. Produce any chart which shows the number of bugs found in each version. You can copy each column one at a time into Excel.

**2. Boxplots**

A boxplot is a tool used to describe a group of numerical data visually through a set of five numbers: minimum, lower quartile, median, upper quartile and maximum. A boxplot may also indicate outliers, but more of this later. Here’s what a boxplot shows: <https://www.khanacademy.org/math/statistics-probability/summarizing-quantitative-data/box-whisker-plots/a/box-plot-review>

Follow the link for Excel: <https://www.youtube.com/watch?v=9yQELTgLWAM>

**EXERCISE 2:** Using the data in the table below, produce boxplots to show the distribution of the “No of bugs” and “size of the class (LOC)”. Label your figure as per the video. You can copy a column into Excel. What is the median number of bugs and median number size of class?

Answer: median bugs 7 and median size of class 350.

**EXERCISE 3:** Calculate the median number of bugs per version? What do you notice about the median values? Do you get the same result if you calculate the averages?

Answer: version 1 median is 3, version 2 median is 5, version 3 median is 7, version 4 median is 9.5 and version 5 median is 11. The median number of bugs increases with the versions (as the system ages).

If you use averages you get version 1 average of 3.6, version 2 average of 5.2, version 3 average of 7, version 4 average of 9.75 and version 5 average of 9. So – the result is not the same (the final average value is lower). This is important since it shows the difference in the power of conclusions we can make based on the statistical measure we use (in this case median versus average). We would get a different conclusion if we used just one measure. For drawing conclusions in Software Engineering, this is important - we need to provide all the evidence.

|  |  |  |  |
| --- | --- | --- | --- |
| Class | No of bugs | Version | Size of the class (LOC) |
| A | 2 | 1 | 100 |
| A | 4 | 2 | 200 |
| A | 6 | 3 | 310 |
| A | 9 | 4 | 400 |
| A | 10 | 5 | 600 |
| B | 3 | 1 | 500 |
| B | 5 | 2 | 230 |
| B | 8 | 3 | 375 |
| B | 8 | 4 | 730 |
| B | 12 | 5 | 731 |
| C | 3 | 1 | 800 |
| C | 5 | 2 | 150 |
| C | 7 | 3 | 334 |
| C | 10 | 4 | 900 |
| C | 13 | 5 | 207 |
| D | 9 | 1 | 760 |
| D | 11 | 2 | 100 |
| D | 13 | 3 | 2000 |
| D | 12 | 4 | 350 |
| D | 1 | 5 | 65 |
| E | 1 | 1 | 70 |
| E | 1 | 2 | 72 |
| E | 1 | 3 | 1378 |

Table 1: Bug data

**3. Correlation**

In statistical analysis, we often look for relationships between variables. Correlation analysis is one statistical technique used to determine whether values of two variables are associated. Correlation analysis is also used to quantify the strength of the relationship between variables. To carry out correlation analysis, we can start with scatterplot to visualize the relationships between variables. So, correlation coefficients are a statistical measure of the strength of the relationship between two variables. It can also identify the direction of the correlation (either negative or positive) between the variables.

**EXERCISE 4:** Use the data in Table 1 to:

1. Draw a scatter plot showing “No of bugs” vs. “Size of the class (LOC)”. Visually, does there appear to be any correlation between the two variables? Although there are some strange values, the trend seems to be one where bugs have a positive relationship with size of class. In other words, as a class gets bigger, more bugs will arise.
2. Calculate the Pearson correlation coefficient between “No of bugs” and “Size of the class (LOC)”. Use the ‘correl’ function in Excel. What do you notice about the correlation coefficients and what it says about the significance? Use the following link to help: <https://www.youtube.com/watch?v=sGlsdHD-lcA>. For argument’s sake, let’s say that a correlation value 0.0 to 0.24 is weak, 0.25-0.49 medium, 0.50-.0.74 strong and 0.75-1.00 very strong. The same grading applies for negative values (0.0 to -0.24 is weak, etc). Correlation value is: 0.32 (rounded) which according to our scale makes it a medium strength correlation.

**EXERCISE 5:** Repeat part b) above for every combination of variables (bugs, version and size). What do you notice about the values?

Correlation values are:

Bugs versus version correlation value is: 0.53 (strong).

Version versus size correlation value is: 0.13 (weak).

This implies that bugs increase the later the version of the system, but class size doesn’t necessarily do the same. Have a think about this. This suggests that complexity of code rather than length per se might be a strong contributing factor to bugs.

**EXERCISE 6:**

In next week’s lab, we will be dealing with data from large systems. Here is the link to a NASA bug dataset: <http://promise.site.uottawa.ca/SERepository/datasets/kc1-class-level-numericdefect.arff> to give you a taster.

If you have time, see if you can import the data into excel and analyse the last two attributes called: @attribute sumLOC\_TOTAL numeric and @attribute NUMDEFECTS numeric. These two attributes represent size of class in LOC and number of bugs. Repeat EXERCISE 4 with this data.

Don’t worry about versions in this data. Sometimes, you get that data, sometimes you don’t.