# Summary

*Shan-I Lee*

*10/5/2017*

**Somewhere in the introduction:**

---

Key findings with regards to the use of interactive techniques in data analysis, or the teaching of interactive data visualisation, will be noted explicitly as they are demonstrated through worked examples.

---

## Conclusion

The two stages of this research project reflects a learning process that students may undertake for a semester-long graduate level course using code-based, open source software for interactive data visualisation. The first stage of surveying interactive techniques and the current software available, reflects the scope and depth of knowledge and skills that could be covered, before application to novel situations. An in-depth survey would require students to put into practice the interactive techniques introduced using the open-source software discussed. The choice of which open-source software to focus on, should be guided by the scope of interactive techniques to be examined. Identification, different types of linked brushing, subset selection and tours, were established as key techniques to introduce. Subsequently this led to a more in-depth survey of the `plotly` and `crosstalk R` packages, as well as `Shiny` and `ggobi`. Familiar "small" datasets can be of use during this stage of the learning process since the focus is on mastering the software tools to enable the interactive techniques, rather than its use for data analysis. However, the educational value of interactive visualisation towards developing a deeper understanding of multivariate data analysis methods, such as principal component analysis and projection pursuits, may naturally arise as students attempt to apply the interactive techniques encountered in literature. Awareness towards the strengths, weaknesses, limitations and code efficiency of the open-source software tools, equips students to make informed decisions about which tool to use, as the focus shifts towards the role of interactive techniques in the data analysis cycle.

The second stage of the learning process focuses on using interactive visualisation within the data analysis cycle. Applying interactive techniques to exploratory data analysis highlighted how filtering and linked brushing helped to address problems with overplotting. However, the feasibility of applying interactive visualisation to "large" datasets, with the open-source tools examined (at their current stage of development), was influenced by the same computational and rendering challenges faced by static plots. The insights into multidimensional data structures gained from individual static plots were leveraged and further explored when interactive techniques were applied. Tooltip identification of outliers and linked brushing of individual and groups of observations, capitalised on the multivariate views offered by parallel coordinates plots. Furthermore, linked brushing between a visual representation of principal component analysis and the parallel coordinates plot enabled further insights into the multivariate structures underlying the data. The flexibility of transferring interactive techniques and software tools previously acquired in the first stage of the learning process, was also demonstrated. The skills learnt to implement interactive sliders for visualising dynamic tours, were utilised for a different purpose of testing the sensitivity of the analysis to sample size. The ease of extending and modifying existing code to interactively explore the data further, reflected the practicality of using interactive techniques in exploratory data analysis. Utilising code-based, open source software for teaching interactive data visualisation enables students to transfer the interactive techniques learnt to novel situations beyond the required coursework. Furthermore, the experiences gained from utilising interactive techniques within the data analysis cycle reflects the educational value and contribution of interactive data visualisation towards providing deeper insights into data.