Code-based, open-source software for teaching interactive data visualisation

Shan-I Lee, BSc (Hons)
Supervisor: Paul Murrell

Department of Statistics
The University of Auckland

November 16, 2017

Problem

Tukey (1965, p. 25)

Today, software and hardware together provide far more powerful factories than most statisticians realize, factories that many of today's most able young people find exciting and worth learning about

Problem

Tukey (1965, p. 25)

Today, software and hardware together provide far more powerful factories than most statisticians realize, factories that many of today's most able young people find exciting and worth learning about

- How does interactivity benefit data analysis?
- Which interactive techniques are 'worth learning'?
- Which code-based, open-source software to use?

Method

- Literature review of interactive techniques.
  - Interactive data visualisation using **GGobi** graphical user interface (Cook and Swayne, 2007)

Method

- Literature review of interactive techniques.
  - Interactive data visualisation using **GGobi** graphical user interface (Cook and Swayne, 2007)
- Survey of current code-based, open-source software.

Method

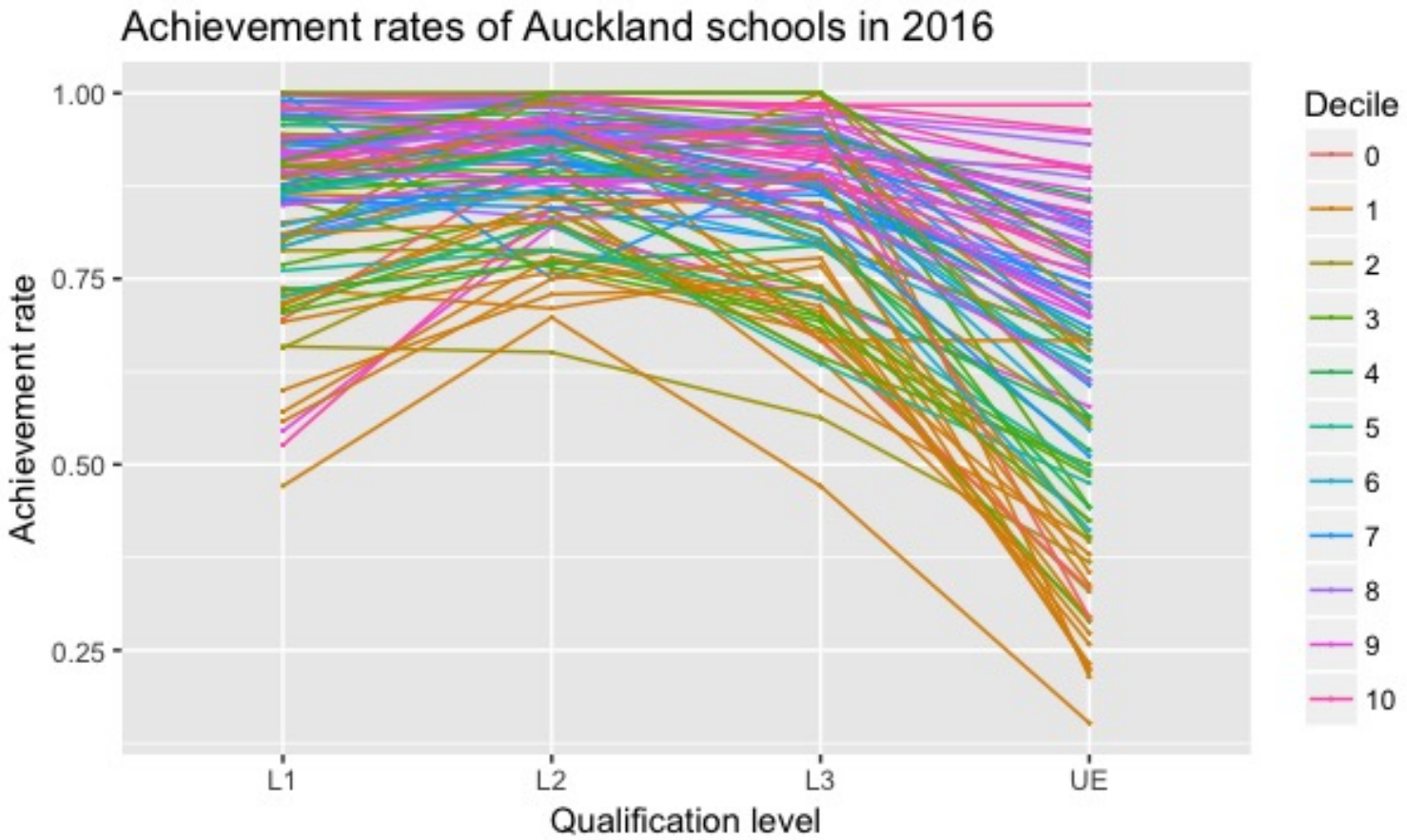
- Literature review of interactive techniques.
  - Interactive data visualisation using **GGobi** graphical user interface (Cook and Swayne, 2007)
- Survey of current code-based, open-source software.
- Explore how interactive techniques further insight into data.
  - Application to exploratory data analysis (EDA) of the 2016 National Certificate of Educational Achievement (NCEA) results for Auckland schools.

Findings

- Key interactive techniques that enrich data analysis:
  - Linked brushing
  - Identification
  - Subset selection
  - Scaling
  - Tours
- A focal set of **R** packages for applying interactive data visualisation: **plotly**, **crosstalk** & **shiny**.
  - Coverage of key interactive techniques
  - Ease of installation and application
- The benefits of interactivity justify the effort of teaching interactive tools.

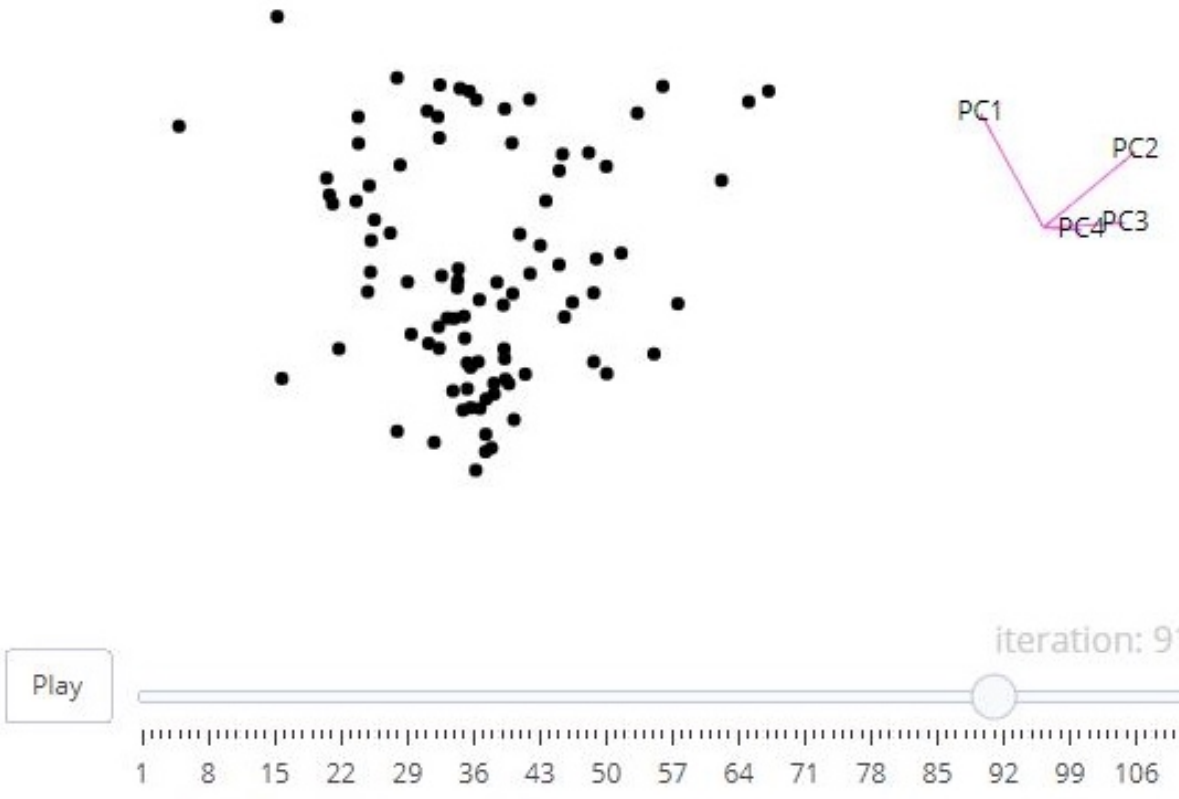
Leveraging static plots

Parallel coordinates plot (PCP) ▸ Demo

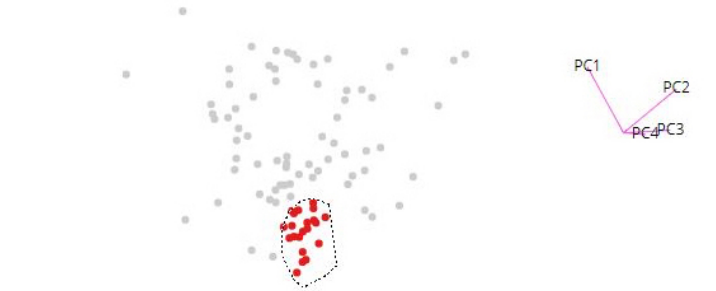


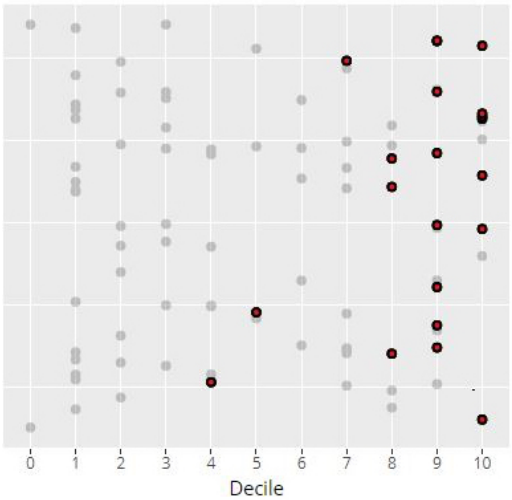
Relating multiple views

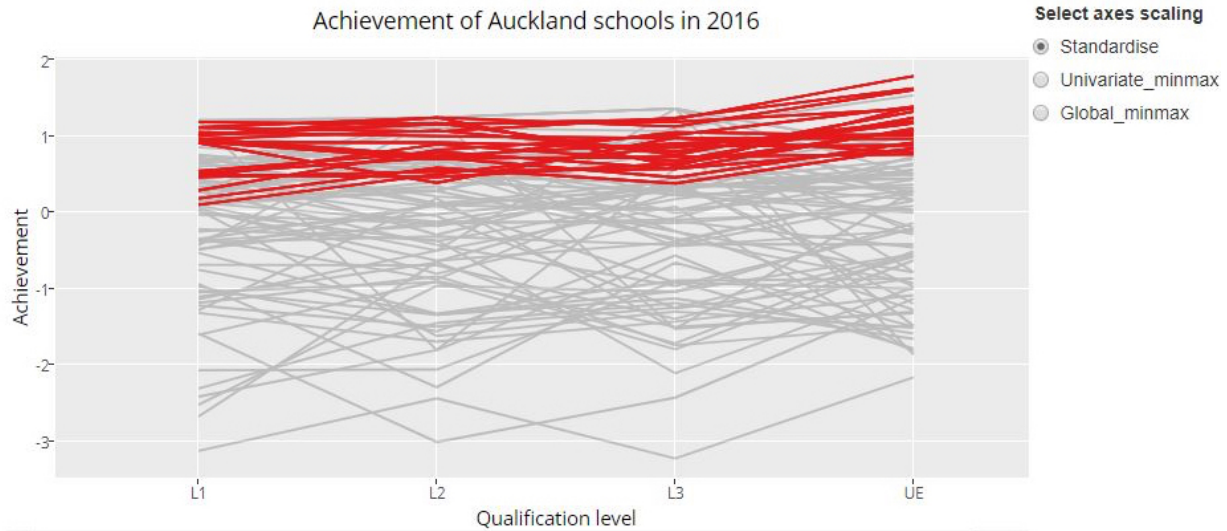
Tours



Relating multiple views







Benefits to EDA

- **Linked brushing** and **identification** allowed fast querying of unusual patterns, groups and/or individuals.
- **Subset selection** via filtering views alleviated issues with overplotting and colour schemes.
- Interactive **scaling** revealed different structures.
- **Linked brushing** related multiple views together and helped with interpretation.
- **Tours** allowed multivariate structures to be explored.
- Questions were quickly addressed and more questions arose from probing the data with interactive techniques.

A focal set of software

Coverage of interactive techniques by **shiny**, **plotly** and **crosstalk**.

Package	Linked brushing	Tooltip Identification	Subset selection	Scaling	Animation (for tours)	Active R session
Crosstalk	Link by case easiest		Filtering views only		Yes	
Plotly		Yes	Filtering views only	Zoom in or out	Yes	
Shiny	Aggregate brush possible		Analysis & filtering views		Yes	Yes

Ease of application

Code for **linked brushing** using **crosstalk** & **plotly**

```
# Function to transform data frame and produce a static PCP
ggpcp <- function(df) {
  # Transform df to a long data frame
  long <- gather(df, "Qualification", "Achievement", c("L1", "L2", "L3", "UE")) %>%
    SharedData$new(key=~School, group="ncea.pcp")
  # Static PCP
  pcp.static <- ggplot(long, aes(x=Qualification, y=Achievement, colour=Decile, group=School)) +
    geom_line() +
    geom_point(size=0.01) +
    labs(x="Qualification level", y="Achievement rate")
  return(pcp.static)
}

ggpcp(akl) %>%
  ggplotly() %>%
    highlight(on="plotly_select", off="plotly_deselect", persistent=T, dynamic=T)
```

A focal set of software

Coverage of interactive techniques by **shiny**, **plotly** and **crosstalk**.

Package	Linked brushing	Tooltip Identification	Subset selection	Scaling	Animation (for tours)	Active R session
Crosstalk	Link by case easiest		Filtering views only		Yes	
Plotly		Yes	Filtering views only	Zoom in or out	Yes	
Shiny	Aggregate brush possible		Analysis & filtering views		Yes	Yes

Conclusion

- Interactive techniques benefit data analysis.
  - Insights beyond static plots
  - Utilises and relates multiple views
  - Further exploration of the data

Conclusion

- Interactive techniques benefit data analysis.
  - Insights beyond static plots
  - Utilises and relates multiple views
  - Further exploration of the data
- The **R** packages **shiny**, **plotly** and **crosstalk** enable interactive data visualisation with code-based, open-source software.

Conclusion

- Interactive techniques benefit data analysis.
  - Insights beyond static plots
  - Utilises and relates multiple views
  - Further exploration of the data
- The **R** packages **shiny**, **plotly** and **crosstalk** enable interactive data visualisation with code-based, open-source software.
- The benefits of applying interactive techniques to data analysis warrant teaching interactive data visualisation to future statisticians.

References I

Chang, W., Cheng, J., Allaire, J., Xie, Y., and McPherson, J. (2017). *shiny: Web Application Framework for R*. R package version 1.0.3.

Cheng, J. (2017). *crosstalk: Inter-Widget Interactivity for HTML Widgets*. R package version 1.0.1.

Cook, D. and Swayne, D. F. (2007). *Interactive and Dynamic Graphics for Data Analysis With R and GGobi*. Springer Publishing Company, Incorporated, 1st edition.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

References II

Sievert, C., Parmer, C., Hocking, T., Chamberlain, S., Ram, K., Corvellec, M., and Despouy, P. (2017). *plotly: Create Interactive Web Graphics via 'plotly.js'*. R package version 4.7.0.

Tukey, J. W. (1965). The technical tools of statistics. *The American Statistician*, 19(2):23–28.