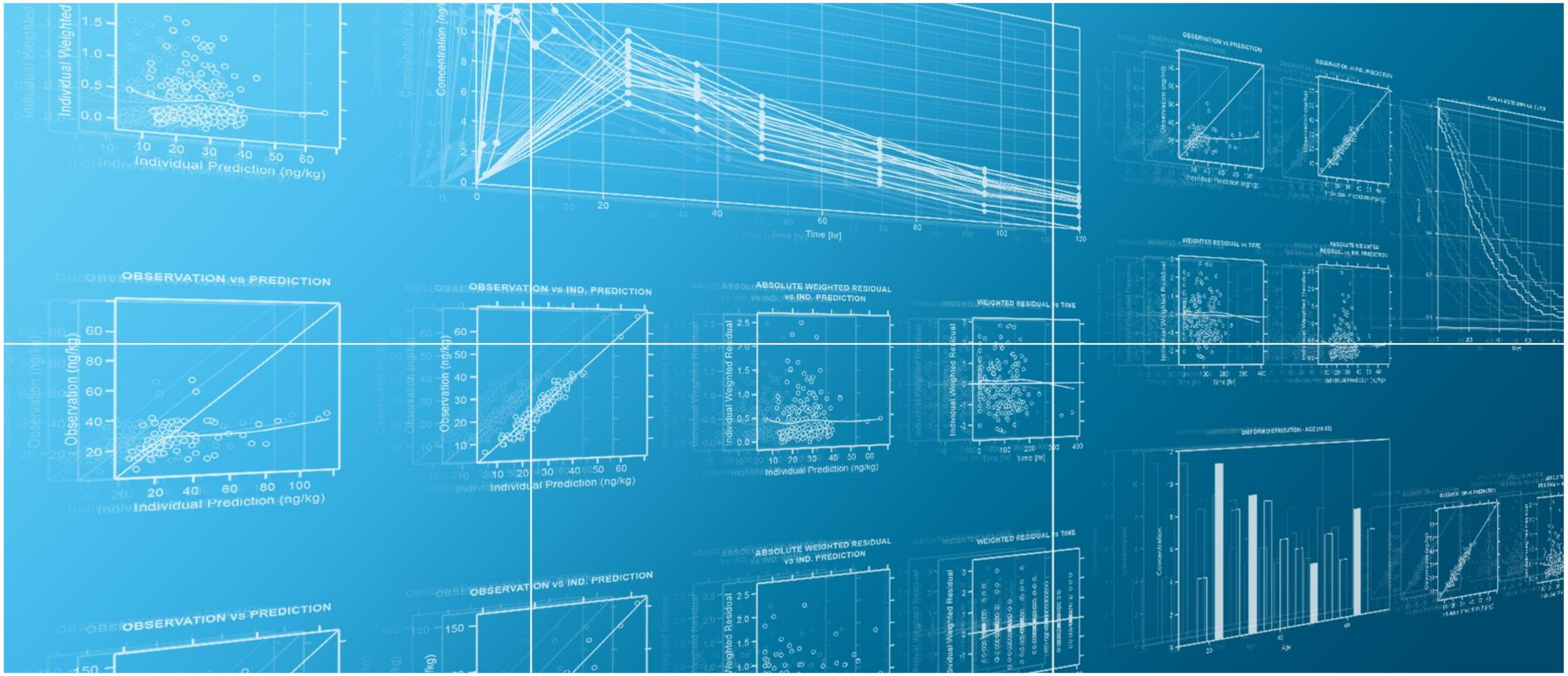
MANGOSOLUTIONS
data analysis that delivers

Introduction to `ggvis`



Overview

- Recap of the basics of ggplot2
- Getting started with ggvis
- The %>% operator
- Changing aesthetics
- Layers
- Interactivity

Aimee Gott – R Consultant
agott@mango-solutions.com

Resources for the Workshop

- R (version 3.1.2)
- RStudio
- ggvis (version 0.4)
- tubeData.csv

Aimee Gott – R Consultant
agott@mango-solutions.com

The Data

- All examples will be using tubeData
- London Tube performance Data from the TFL website
- The original data can be found on

  http://data.london.gov.uk/dataset/tube-network-performance-data-transport-committee-report

Aimee Gott – R Consultant
agott@mango-solutions.com

RECAP OF `ggplot2`

MANGOSOLUTIONS
data analysis that delivers

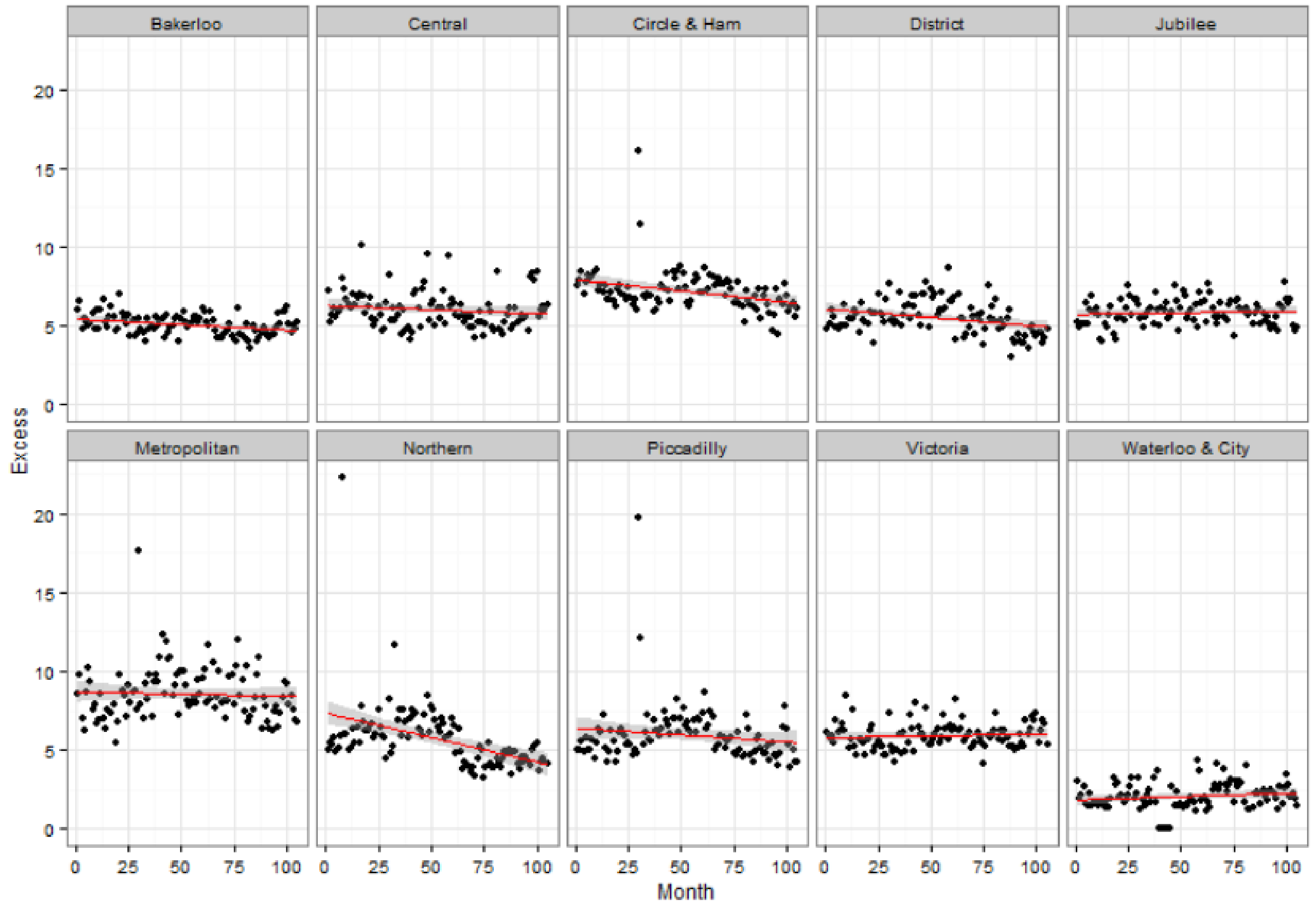
Aimee Gott – R Consultant
agott@mango-solutions.com

Main features of ggplot2

- Create graphics using `qplot` or `ggplot`
- Add layers to an existing plot using `+`
- Change aesthetics by variables in the data
- Control the type of plot using `geoms`
- Panel by variables using the `facet_*` functions

Aimee Gott – R Consultant
agott@mango-solutions.com

```
> qplot(Month, Excess, data = tubeData) +
+         geom_smooth(method = "lm", col = "red") +
+         facet_wrap(~Line) +
+         theme_bw()
```



The `geoms`

- ggplot2 includes a number of geoms for controlling the type of plot we create

```
> grep("^geom", objects("package:ggplot2"), value = TRUE)
 [1] "geom_abline"     "geom_area"       "geom_bar"        "geom_bin2d"
 [5] "geom_blank"      "geom_boxplot"    "geom_contour"    geom_crossbar"
 [9] "geom_density"    "geom_density2d" "geom_dotplot"    "geom_errorbar"
[13] "geom_errorbarh"  "geom_freqpoly"   "geom_hex"        "geom_histogram"
[17] "geom_hline"      "geom_jitter"     "geom_line"       "geom_linerange"
[21] "geom_map"        "geom_path"       "geom_point"      "geom_pointrange"
[25] "geom_polygon"    "geom_quantile"   "geom_raster"     "geom_rect"
[29] "geom_ribbon"     "geom_rug"        "geom_segment"    "geom_smooth"
[33] "geom_step"       "geom_text"       "geom_tile"       "geom_violin"
[37] "geom_vline"
```

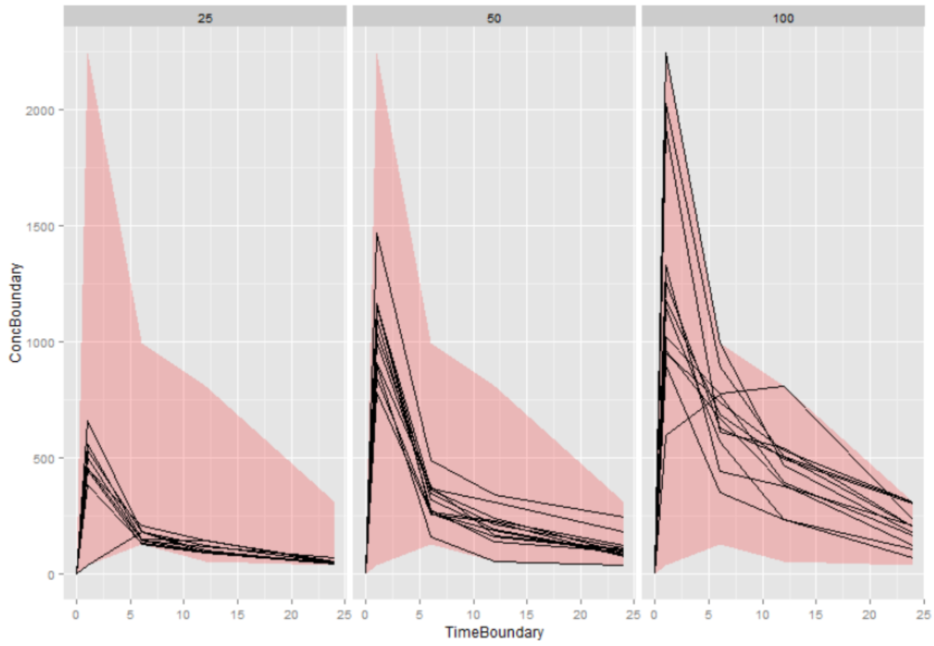
MANGOSOLUTIONS
data analysis that delivers

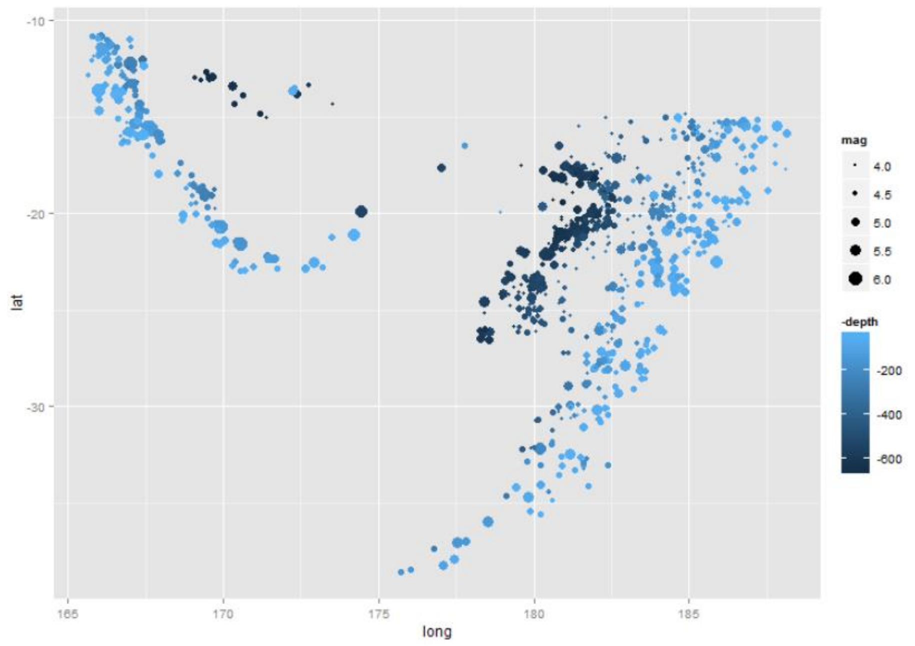
Aimee Gott – R Consultant
agott@mango-solutions.com

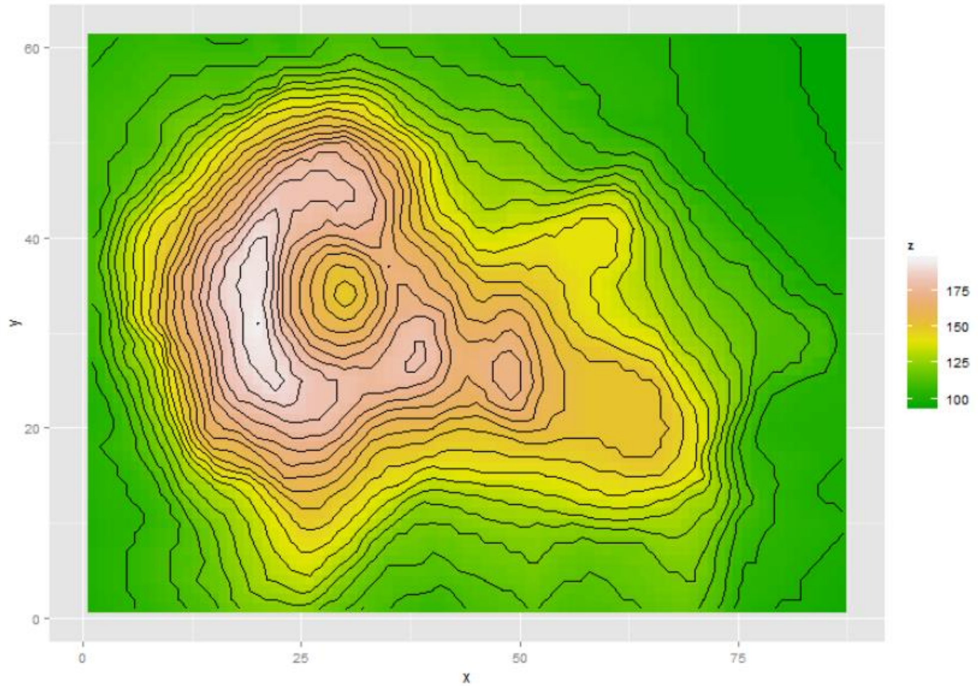
Facetting

- We can panel graphics based on variables in the data using facets
- facet_wrap and facet_grid add panels as layers

Aimee Gott – R Consultant
agott@mango-solutions.com


25
50
100
ConcBoundary
TimeBoundary
0
500
1000
1500
2000
0
5
10
15
20
25


lat
long
-10
-20
-30
165
170
175
180
185
mag
4.0
4.5
5.0
5.5
6.0
-depth
-200
-400
-600


y
x
0
20
40
60
0
25
50
75
z
175
150
125
100

Scales and themes

- ggplot2 provides a large number of scale functions to control aspects of a graphic including axes and legends
- theme functions allow us to control the overall style of the graphic

Aimee Gott – R Consultant
agott@mango-solutions.com

GETTING STARTED WITH `ggvis`

MANGOSOLUTIONS
data analysis that delivers

Aimee Gott – R Consultant
agott@mango-solutions.com

Using `ggvis` - a word of warning!

```
> require(ggvis)
Loading required package: ggvis
The ggvis API is currently rapidly evolving. We strongly
recommend that you do not rely on this for production,
but feel free to explore. If you encounter a clear bug,
please file a minimal reproducible example at
https://github.com/rstudio/ggvis/issues. For questions
and other discussion, please use
https://groups.google.com/group/ggvis.
```

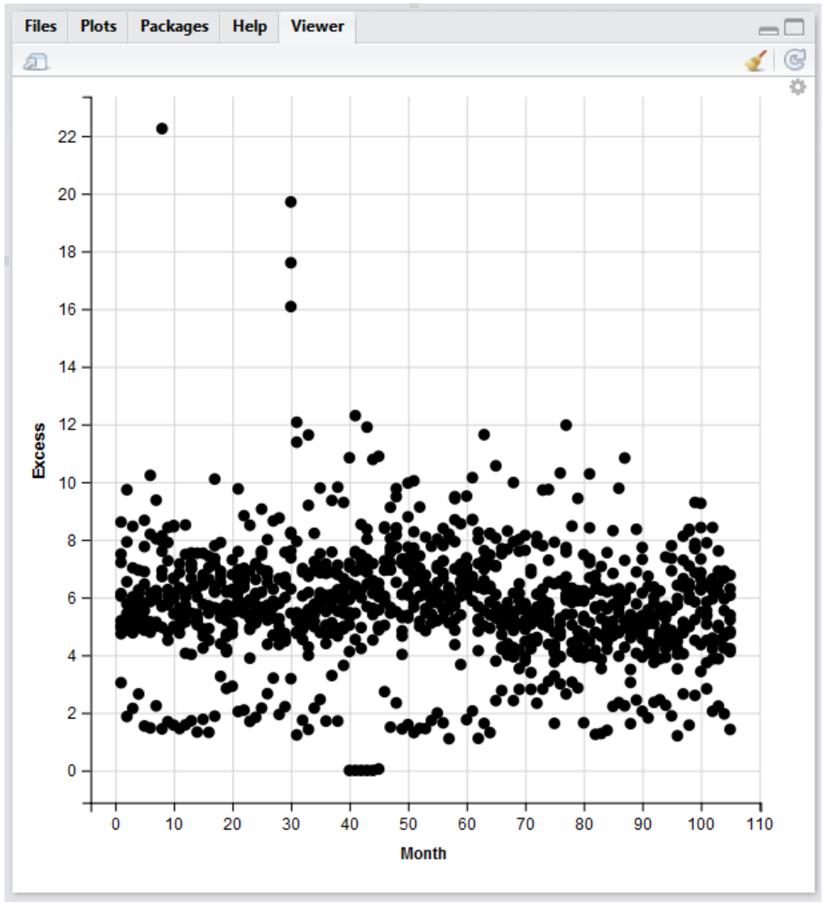
Aimee Gott – R Consultant
agott@mango-solutions.com

Creating a first plot

- To create a plot object we use the function `ggvis`
- When we refer to variables in the data we use the `~` symbol before the name, i.e. `~Ozone`
- We need to use a layer function, such as `layer_points`, to plot the object

Aimee Gott – R Consultant
agott@mango-solutions.com

```
> myPlot <- ggvis(tubeData, x = ~Month, y = ~Excess)
> layer_points(myPlot)
```



Viewing `ggvis` graphics

- ggvis uses Vega to render graphics in a web browser
- In RStudio the default it to use the "Viewer" pane

- From the web browser we can download SVG or png version of our graphics

Aimee Gott – R Consultant
agott@mango-solutions.com

THE %>% OPERATOR

MANGOSOLUTIONS
data analysis that delivers

Aimee Gott – R Consultant
agott@mango-solutions.com

The %>% Operator

- `ggvis` makes use of the %>% operator from the package `magrittr`
- This allows us to layer up graphics in the same way we would with `ggplot2`

MANGOSOLUTIONS
data analysis that delivers

Aimee Gott – R Consultant
agott@mango-solutions.com

The %>% Operator

- The %>% operator passes the left hand object to the first argument of the right hand expression

```
> tubeData$Excess %>% tapply(tubeData$Line, mean)
       Bakerloo          Central    Circle & HamDistrict
       5.047714         5.998667                7.166095
```

- We can pass data or objects to functions in this way

Aimee Gott – R Consultant
agott@mango-solutions.com

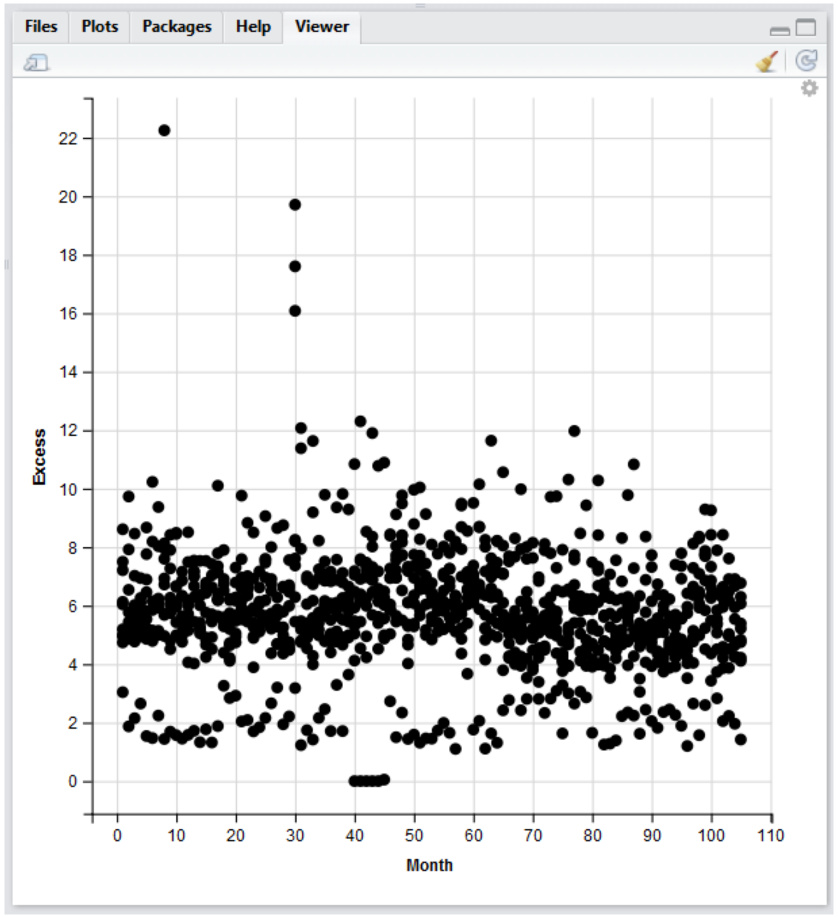
%>% in ggvis

- With ggvis we pass "ggvis" objects

- We create the initial object by passing data to ggvis()
- All other functions expect a ggvis object as the first argument and return a ggvis object

Aimee Gott – R Consultant
agott@mango-solutions.com

```
> tubeData %>%
+     ggvis(x = ~Month, y = ~Excess) %>%
+     layer_points()
```



CHANGING PROPERTIES

MANGOSOLUTIONS
data analysis that delivers

Aimee Gott – R Consultant
agott@mango-solutions.com

Aesthetics

- As with all graphics there are a number of aesthetics we can set
  - stroke
  - fill
  - size
  - opacity

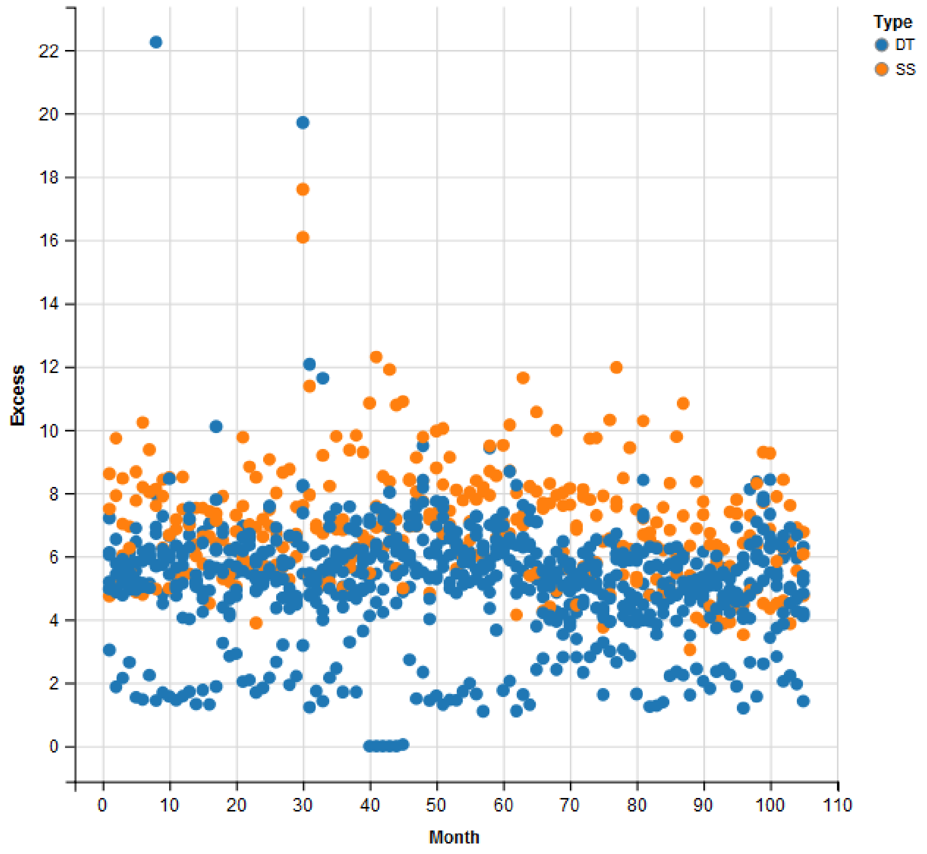
Aimee Gott – R Consultant
agott@mango-solutions.com

Changing based on variables

- In ggvis we map a variable to a property using "="
- We have to remember to use the "~" with all variable names
- fill = ~Line would set the fill based on the Line variable

Aimee Gott – R Consultant
agott@mango-solutions.com

```
> tubeData %>%
+     ggvis(x = ~Month, y = ~Excess, fill = ~Type) %>%
+     layer_points()
```

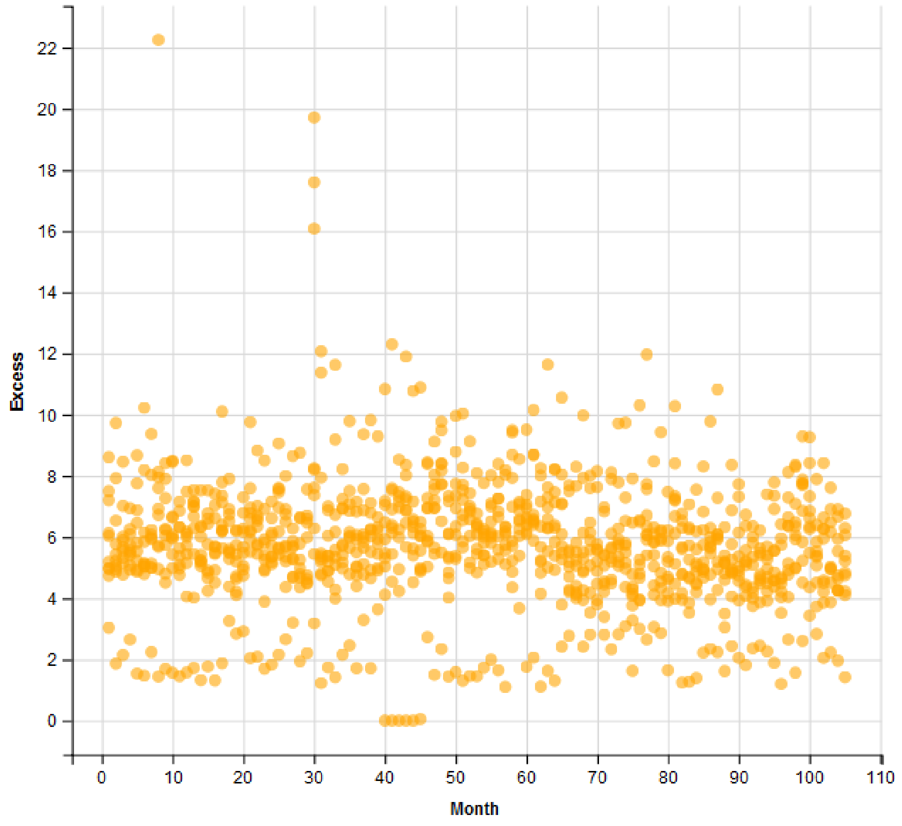


Setting property values

- When we set a property based on a value we use ":="
- fill := "red" would set the fill to red

Aimee Gott – R Consultant
agott@mango-solutions.com

```
> tubeData %>%
+     ggvis(x = ~Month, y = ~Excess, fill := "orange",
+           opacity := 0.6) %>%
+     layer_points()
```



Exercise

- Create a plot of mpg against wt using the mtcars data
- Update the plot to colour by the cylinder variable, ensure that the points are coloured by distinct colours rather than on a scale
- Update the plotting symbol to be triangles

Aimee Gott – R Consultant
agott@mango-solutions.com

ADDING LAYERS

Aimee Gott – R Consultant
agott@mango-solutions.com

Changing the plot type

- In ggplot2 we use geom functions to determine the type of plot we create
- In ggvis we use `layer` functions
- Not all geoms are currently available as layers

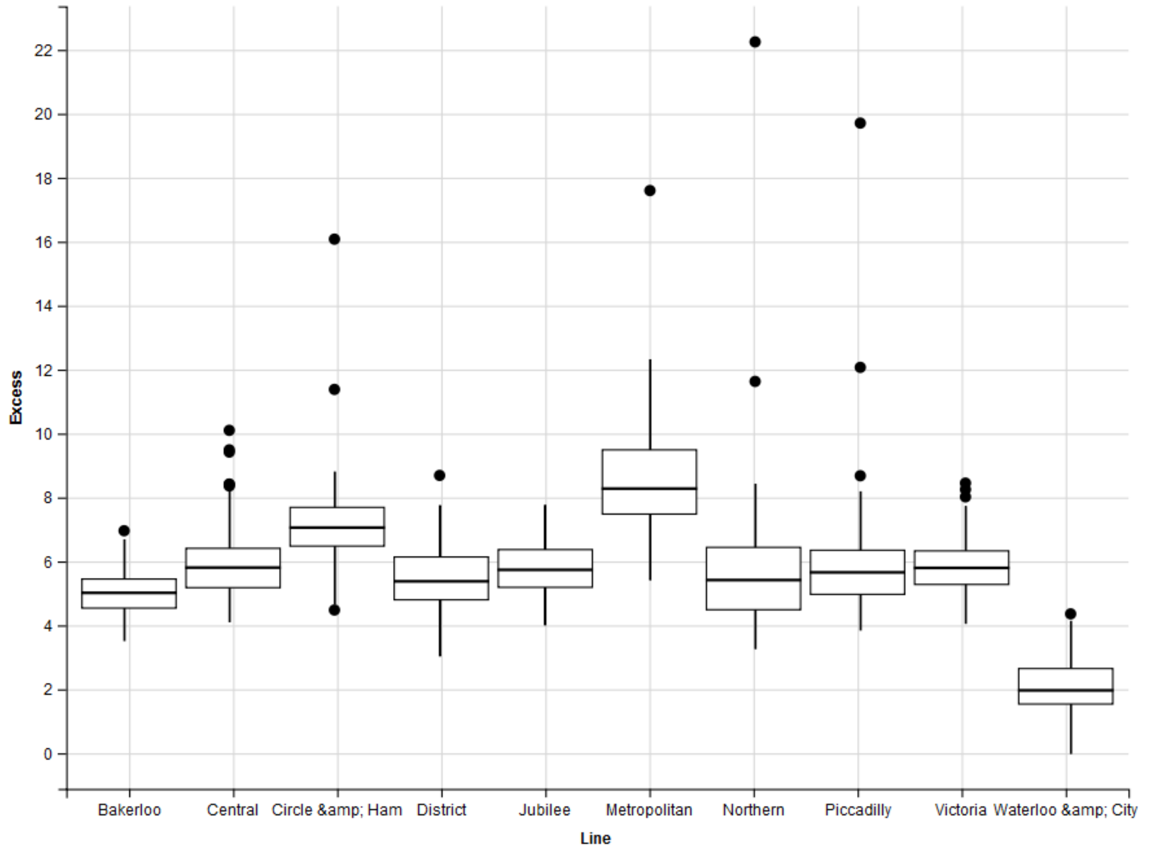
Aimee Gott – R Consultant
agott@mango-solutions.com

Layers

Function	Description
layer_points	Adds data as points
layer_histograms	Adds data as a histogram
layer_boxplots	Draws as a boxplot
layer_lines	Adds data as lines
layer_smooths	Adds a smoothing line
layer_paths	Joins data as a single path
layer_text	Adds text
layer_model_predictions	Adds lines for model predictions, such as lm lines

Aimee Gott – R Consultant
agott@mango-solutions.com

```
> tubeData %>%
+     ggvis(x = ~Line, y = ~Excess) %>%
+     layer_boxplots()
```



Exercise

- Update the plot of mpg against wt to include a smooth line of the data
- Add a confidence interval to the smooth line and colour in red
- Add a regression line and colour it blue

- Create a boxplot of mpg split by cylinder (*hint: the cylinder variable will need to be a factor*)

Aimee Gott – R Consultant
agott@mango-solutions.com

MAKING PLOTS INTERACTIVE

MANGOSOLUTIONS
data analysis that delivers

Aimee Gott – R Consultant
agott@mango-solutions.com

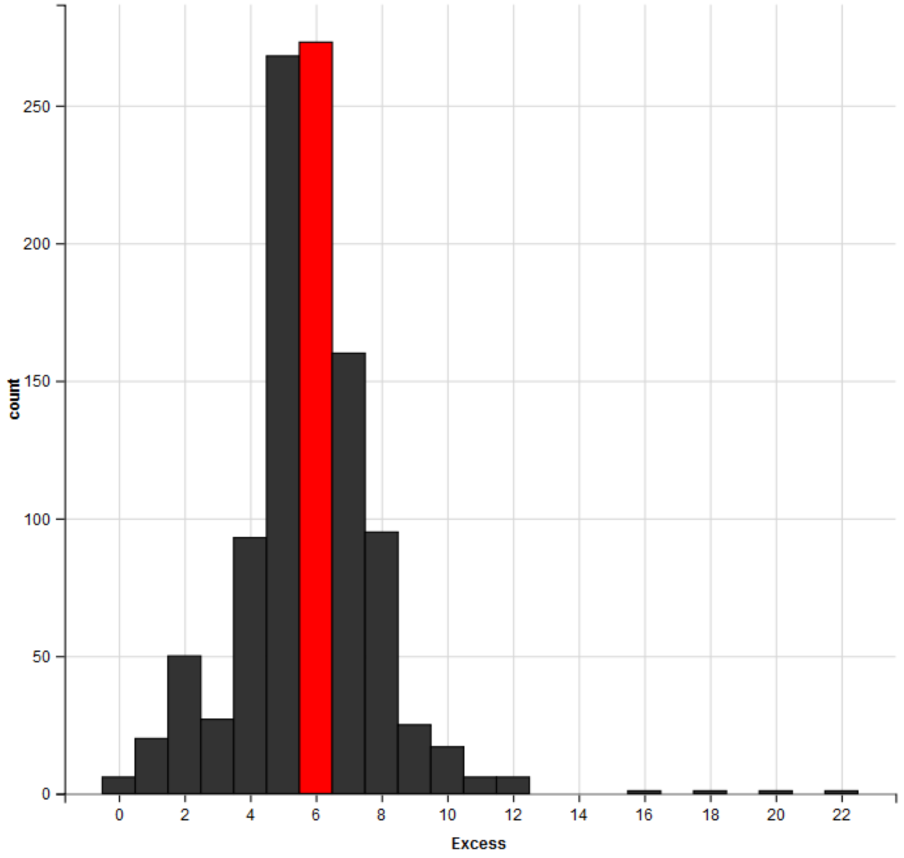
Basic interactivity

- The most basic interactivity we can add is "hover over" changes
- We can change properties by using `property.hover` arguments

```
fill.hover := "red"
```

Aimee Gott – R Consultant
agott@mango-solutions.com

```
> tubeData %>%
+     ggvis(~Excess) %>%
+     layer_histograms(fill.hover = "red")
```



Interactive Input

- We can also set properties to be the output of an interactive control

```
opacity := input_slider(0, 1, label =
"Opacity")
```

- We use the setting ":=" for this input
- We can optionally set labels next to the control

Aimee Gott – R Consultant
agott@mango-solutions.com

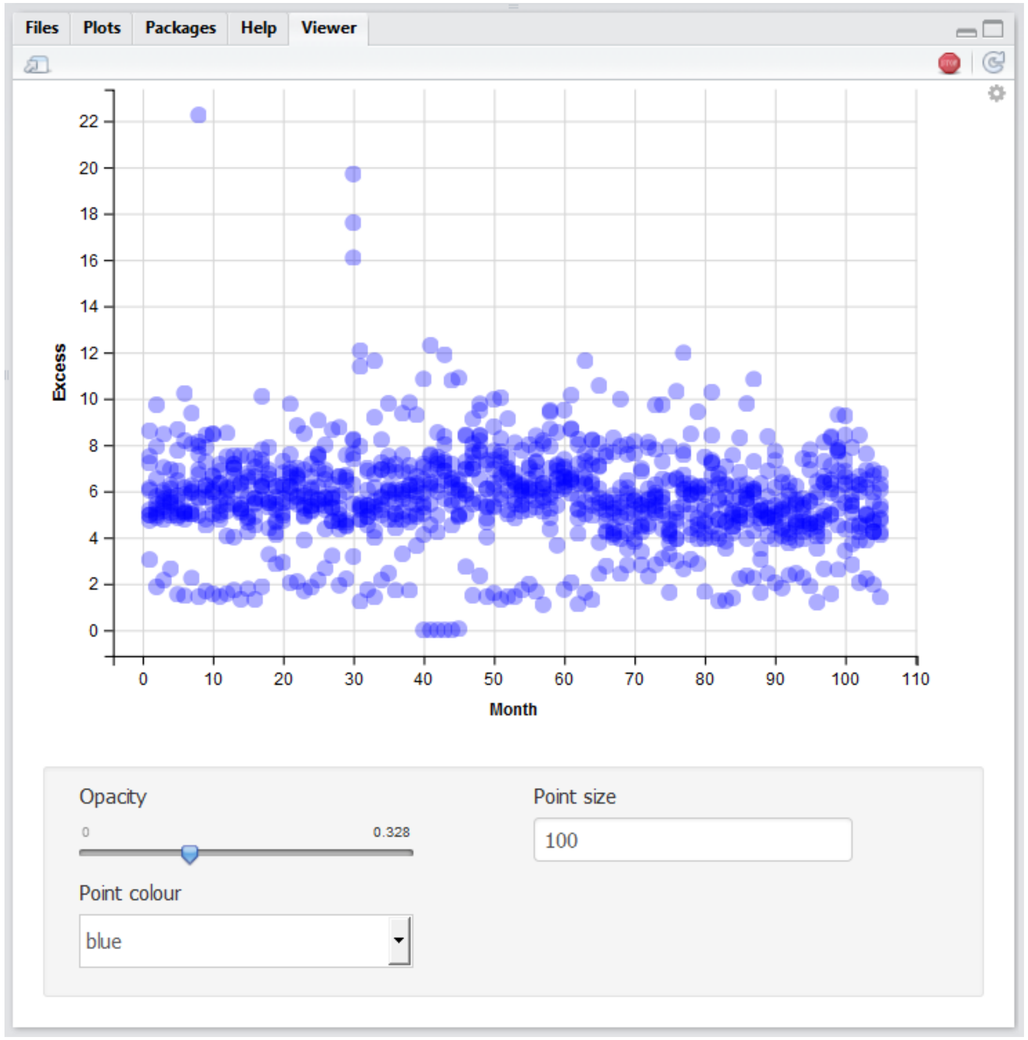
Interactive Input Functions

Function	Description
input_slider	Slider to select values or ranges of values
input_checkbox	A single check box
input_checkboxgroup	A group of check boxes
input_numeric	A spin box
input_radiobuttons	Selection of a single value from a set of options
input_select	A drop down text selection
input_text	Text input

Aimee Gott – R Consultant
agott@mango-solutions.com

```
> tubeData %>%
+   ggvis(x = ~Month, y = ~Excess,
+         opacity := input_slider(0, 1,
+                                 value = 0.7, label = "Opacity"),
+         size := input_numeric(30, label = "Point size"),
+         fill := input_select(c("red", "orange", "blue"),
+                                 label = "Point colour")) %>%
+   layer_points()
```

Aimee Gott – R Consultant
agott@mango-solutions.com



Opacity
0 0.328

Point colour
blue

Point size
100

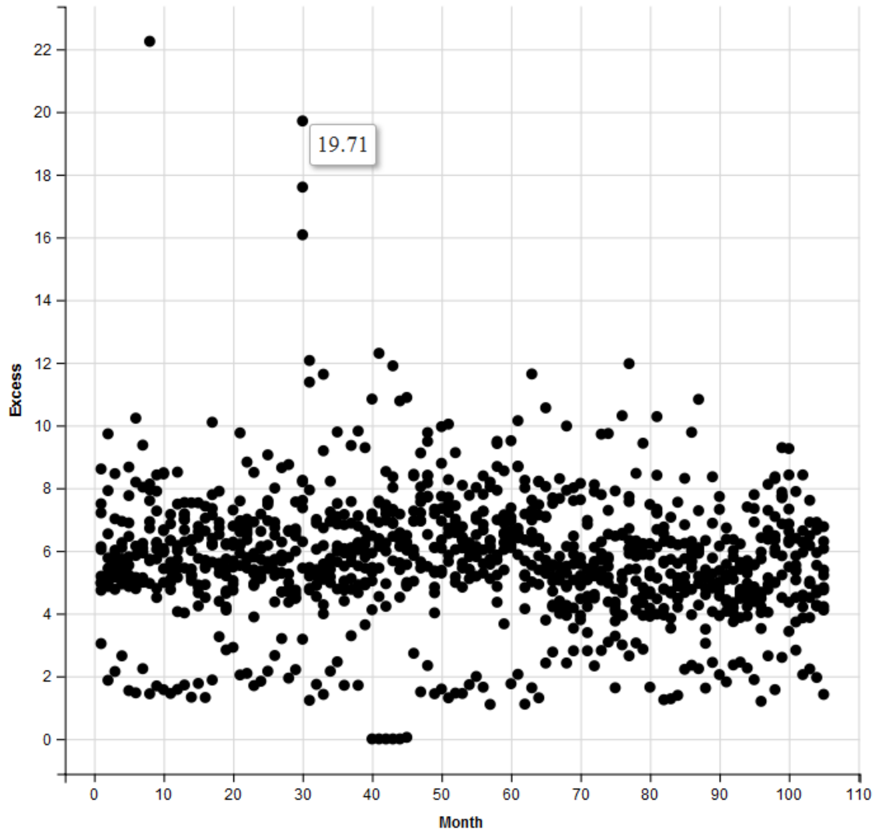
Aimee Gott – R Consultant
agott@mango-solutions.com

Tooltips

- add_tooltip allows us to include other behaviour when we hover or click on a point
- We can provide a single function that takes as input a list of the data stored in a given point

Aimee Gott – R Consultant
agott@mango-solutions.com

```
> tubeData %>%
+     ggvis(x = ~Month, y = ~Excess) %>%
+     layer_points()  %>%
+     add_tooltip(function(data) data$Excess)
```



Exercise

- Update the previous plot of mpg against wt so points change colour when they hover over
- Add a tooltip that shows the value of mpg when the point is hovered over
- Add a slider for the span of the smooth line so that values can be set between 0 and 1

Aimee Gott – R Consultant
agott@mango-solutions.com

COMMON PLOT FUNCTIONS

Aimee Gott – R Consultant
agott@mango-solutions.com

Controlling axis and legends

- We can control the axes using the add_axis function
- This controls axis labels, tick marks and even grid lines

```
add_axis("x", title = "Month")
```

Aimee Gott – R Consultant
agott@mango-solutions.com

Controlling axis and legends

- The add_legend and hide_legend functions allow us to control if we see a legend and where it appears

```
hide_legend("fill")
add_legend(c("fill", shape))
```

Aimee Gott – R Consultant
agott@mango-solutions.com

Scales

- ggvis has fewer scale functions than in ggplot2 but control much more

```
> grep("^scale", objects("package:ggvis"), value = TRUE)
[1] "scale_datetime" "scale_logical"  "scale_nominal"  "scale_numeric"
[5] "scale_ordinal"  "scale_singular" "scaled_value"
```

Aimee Gott – R Consultant
agott@mango-solutions.com

ggvis VS ggplot2

Aimee Gott – R Consultant
agott@mango-solutions.com

How are they similar?

- We can layer graphics in a similar fashion
- Aesthetics can be set based on variables in the data
- We can control the type of plot with specific functions

Aimee Gott – R Consultant
agott@mango-solutions.com

How are they different?

- Only one main plot function to work with as opposed to two
- Layering is done using %>% rather than +
- Fewer scale functions
- Much functionality is not yet available in ggvis e.g. facetting

Aimee Gott – R Consultant
agott@mango-solutions.com

Which should I use?

- For static graphics: ggplot2
- For interactive graphics: ggvis**

**If you are using ggvis remember it's still being actively developed and may change in structure and functionality

Aimee Gott – R Consultant
agott@mango-solutions.com

Finding out more

- Rstudio are maintaining documentation on their webpage:

  http://ggvis.rstudio.com/

- Or come have a chat after the workshop!

Aimee Gott – R Consultant
agott@mango-solutions.com