# Title of the topic

## Workshop on Topic

**Doe Jones** • 17-Jan-2022

NBIS

# Contents

**NBIS**

- RMarkdown
- Slides
- Layout
- Text formatting
- Code formatting
- Lists
- Images
- Math expressions
- Tables
- Static plots
- Interactive plots
- Maps
- Crosstalk
- General tips

# Introduction

This is a demo to showcase the **RaukR presentation design guide**, **usage** and capabilities of this presentation system. This presentation is created in RStudio using R package xaringan. This package in turn utilises the javascript library remarkjs as the underlying framework. Note that the markdown interpreter used by remarkjs (blackfriday) is different from that used by knitr (pandoc). Therefore, some features that you may be familiar with in regular RMarkdown may not work here. There are also many new features here not available in regular RMarkdown.

Note that many of the classes used in this presentation are custom built and not part of any standard package.

# RMarkdown

- This is an RMarkdown presentation.
- Markdown is a simple formatting syntax for authoring documents exported to HTML, PDF etc.
- In RStudio, you create a `.Rmd` text file.
- Add YAML matter to the top if not already there.

```
---
title: "This is a title"
output:
  xaringan::moon_reader
---
```

- Click the **Knit** button for interactive render.
- Or using `render()` as such:

```
rmarkdown::render("report.Rmd",output_format="html_document")
```

# Slides

**NB S**

**Slide separators**

Slides are separated by `---` .
Incremental content on same slide is separated by `--` like below.

**Slide properties**

Slides properties are created as below:

```
---
property: value
```

Slides can be named as such `name: intro` and then hyperlinked from elsewhere. For eg: `go to`
`[introduction](#intro)` displays as go to introduction.

Custom classes can be defined for each slide.

```
---
class: spaced
```

Few default options are *spaced* (increases line spacing), *middle* (center aligns content vertically)
and *center* (center aligns content horizontally).

Slides can be hidden using `exclude: true` .
Slides can be dropped from page count using `count: false` .

# Slides

**NB S**

**Slide notes**

Any content below `???` on a slide are notes. This is only visible in presenter mode (by pressing **P**).

**Keyboard shortcuts**

Press **H** to view keyboard shorcuts. Pressing **C** clones and creates two linked presentations.

# Layout

The slide content can be organised into columns which can be nested if needed. Classes for 30, 40, 50, 60 and 70 have been implemented for left and right. Note that the total width must sum to 100.

```
.pull-left-50[
<div style="background-color:#fdebd0">Left content</div>
]

.pull-right-50[
<div style="background-color:#eaf2f8">Right content</div>

.pull-left-60[
<div style="background-color:#d0ece7">Inner left content</div>
]
.pull-right-40[
<div style="background-color:#f2d7d5">Inner right content</div>
]
]
```

Left content

Right content

Inner left content

Inner right content

# Custom Divs

```
<div class="abstract">
This is a special box.
</div>
```

This is a special box.

# Text Formatting

NB S

Headings can be defined as shown below.

```
## Level 2 heading
### Level 3 heading
#### Level 4 heading
##### Level 5 heading
###### Level 6 heading
```

## Level 2 heading

### Level 3 heading

#### Level 4 heading

##### Level 5 heading

###### Level 6 heading

Level 1 usage is not recommended. Use level 2 for slide titles. Use level 3 and below for other titles.

Six custom classes are defined for text scaling.

```
.largest[Largest text.]
.larger[Larger text.]
.large[Large text.]
Normal text.
.small[Small text]
.smaller[Smaller text.]
.smallest[Smallest text.]
```

# Largest text.

## Larger text.

### Large text.

Normal text.

Small text.

Smaller text.

Smallest text.

# Text Formatting

Horizontal alignment of text can be adjusted as shown below.

```
.left[Left aligned text.]
.center[Center aligned text.]
.right[Right aligned text.]
```

Left aligned text.

<div align="center">Center aligned text.</div>

<div align="right">Right aligned text.</div>

Indented quotes using `>`

> This line is blockquoted

Icons from FontAwesome can be displayed using the HTML `<i>` tag. Note that not all icons may work.

```
Here is a <i class='fa fa-calendar'></i>
calendar and a <i class='fa fa-couch'></i>
couch.
```

Here is a 📅 calendar and a 🛋 couch.

A horizontal line can be created using `***`

---

`This is __Bold text__`  This is **Bold text**

`This is _Italic text_`  This is *Italic text*

`~~Strikethrough~~ text`  ~~Strikethrough~~ text

This is Subscript `H<sub>2</sub>O` displayed as $H_2O$

This is Superscript `2<sup>10</sup>` displayed as $2^{10}$

`This is a [link](r-project.org)`  This is a link

# Code formatting

**NB S**

Code can be defined inline where `` `this` `` looks like `this` . R code can be executed inline `` `r Sys.Date()` `` producing 2022-01-17. Code can also be defined inside code blocks.

```
```
This is code
```
```

```
This is code
```

R code is executed inside code blocks like this

```
```{r}
Sys.Date()
```
```

which shows the code and output.

```
Sys.Date()
```

```
## [1] "2022-01-17"
```

The code and results can be hidden by `` ```{r,echo=FALSE,results='hide'}` `` .

```
data(iris)
head(iris[,1:2])
```

```
##    Sepal.Length Sepal.Width
## 1           5.1         3.5
## 2           4.9         3.0
## 3           4.7         3.2
## 4           4.6         3.1
## 5           5.0         3.6
## 6           5.4         3.9
```

# Lists

**Bulleted List**

Bullet points are defined using the asterisk (*).

```
* Bullet 1
* Bullet 2
    + Sub-bullet 2.1
```

- Bullet 1
- Bullet 2

  - Sub-bullet 2.1

**Incremental Bullets**

```
* Incremental Bullet 1

--
* Incremental Bullet 2
```

Note that there is an empty line between the bullet point and the `--` below.

- Incremental Bullet 1
- Incremental Bullet 2

# Images • Markdown

## Using Markdown

Using regular markdown.

```
![](assets/landing.png)
```



The dimensions are based on image and/or fill up the entire available space. You have no control over the displayed dimensions.

Custom classes can be used to control size.

```
.size-20[![](assets/landing.png)]
.size-10[![](assets/landing.png)]
```





The number denotes % of the original dimension. The following size classes are implemented: 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95.

# Images • HTML

## Using Raw HTML

This image is 30% size.

```
<img src="assets/landing.png" style="width:30%;"/>
```



This image is 20% size, has shadow and corners rounded.

```
<img src="assets/landing.png" style="width:20%;" class="fancyimage"/>
```

# Images • R

**Using R**

R chunks in RMarkdown can be used to control image display size using the argument `out.width`.

This image is displayed at a size of 250 pixels.

```{r,out.width=250}
knitr::include_graphics('assets/landing.png')
```

# Math expressions

Some examples of rendering equations.

$$e^{i\pi} + 1 = 0$$

$$\frac{E \times X^2 \prod I}{2 + 7} = 432$$

$$\sum_{i=1}^{n} X_i$$

$$\int_0^{2\pi} \sin x \, dx$$

$$\left(\sum_{i=1}^{n} i\right)^2 = \left(\frac{n(n-1)}{2}\right)^2 = \frac{n^2(n-1)^2}{4}$$

$$P(|X - \mu| > k) = P(|X - \mu|^2 > k^2)$$

$$\leq \frac{\mathbb{E}\left[|X - \mu|^2\right]}{k^2}$$

$$\leq \frac{\mathrm{Var}[X]}{k^2}$$

$$X \sim \mathrm{N}(0, 1)$$

$$Y \sim \chi^2_{n-p}$$

$$R \equiv X/Y \sim t_{n-p}$$

# Tables • `kable`

The most simple table using `kable` from R package `knitr`.

```
knitr::kable(head(iris),'html')
```

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|---:|---:|---:|---:|---|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |

# Tables • `kableExtra`

More advanced table using `kableExtra` and `formattable` .

```
iris[c(1:2,51:52,105:106),] %>%
  mutate(Sepal.Length=color_bar("lightsteelblue")(Sepal.Length)) %>%
  mutate(Sepal.Width=color_tile("white","orange")(Sepal.Width)) %>%
  mutate(Species=cell_spec(Species,"html",color="white",bold=T,
    background=c("#8dd3c7","#fb8072","#bebada")[factor(.$Species)])) %>%
  kable("html",escape=F) %>%
  kable_styling(bootstrap_options=c("striped","hover","responsive"),
                full_width=F,position="left") %>%
  column_spec(5,width="3cm")
```

|     | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|-----|--------------|-------------|--------------|-------------|---------|
| 1   | 5.1          | 3.5         | 1.4          | 0.2         | setosa |
| 2   | 4.9          | 3.0         | 1.4          | 0.2         | setosa |
| 51  | 7.0          | 3.2         | 4.7          | 1.4         | versicolor |
| 52  | 6.4          | 3.2         | 4.5          | 1.5         | versicolor |
| 105 | 6.5          | 3.0         | 5.8          | 2.2         | virginica |
| 106 | 7.6          | 3.0         | 6.6          | 2.1         | virginica |

# Tables • DT

Interactive table using R package DT .

```r
library(DT)
DT::datatable(iris[1:20,],options=list(pageLength=7))
```

Show 7 entries                                                    Search: [          ]

| | Sepal.Length ⇕ | Sepal.Width ⇕ | Petal.Length ⇕ | Petal.Width ⇕ | Species ⇕ |
|---|---|---|---|---|---|
| 1 | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 2 | 4.9 | 3 | 1.4 | 0.2 | setosa |
| 3 | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4 | 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5 | 5 | 3.6 | 1.4 | 0.2 | setosa |
| 6 | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 7 | 4.6 | 3.4 | 1.4 | 0.3 | setosa |

Showing 1 to 7 of 20 entries                        Previous  1  2  3  Next

# Static plots • Base Plot

Plots using base R are widely used and may be good enough for most situations. But they lack a consistent coding framework.

```r
{par(mar=c(5,5,0,0))
  plot(x=iris$Sepal.Length,y=iris$Sepal.Width,
      col=c("coral","steelblue","forestgreen")[iris$Species],
      xlab="Sepal Length",ylab="Sepal Width",pch=19)
  legend(x=7,y=4.47,legend=c("setosa","versicolor","virginica"),
      col=c("coral","steelblue","forestgreen"),pch=19)}
```
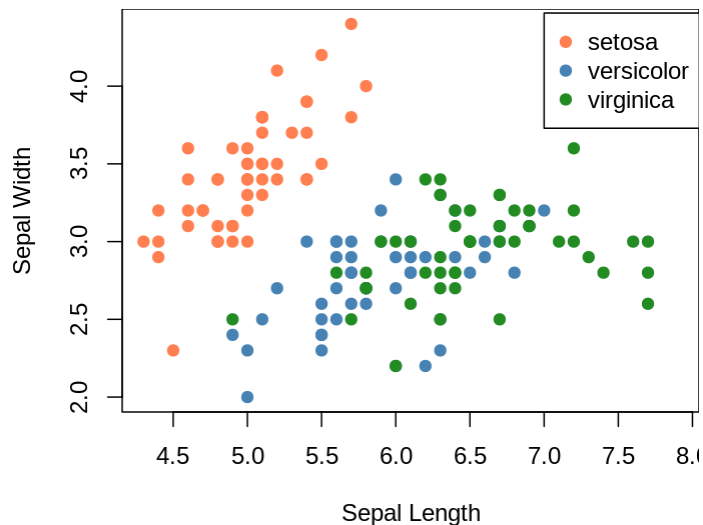
# Static plots • `ggplot2`

R package `ggplot2` is a versatile and almost complete plotting solution.

```r
iris %>%
  ggplot(aes(x=Sepal.Length,y=Sepal.Width,col=Species))+
  geom_point(size=2)+
  labs(x="Sepal Length",y="Sepal Width")+
  theme_bw(base_size=18)
```
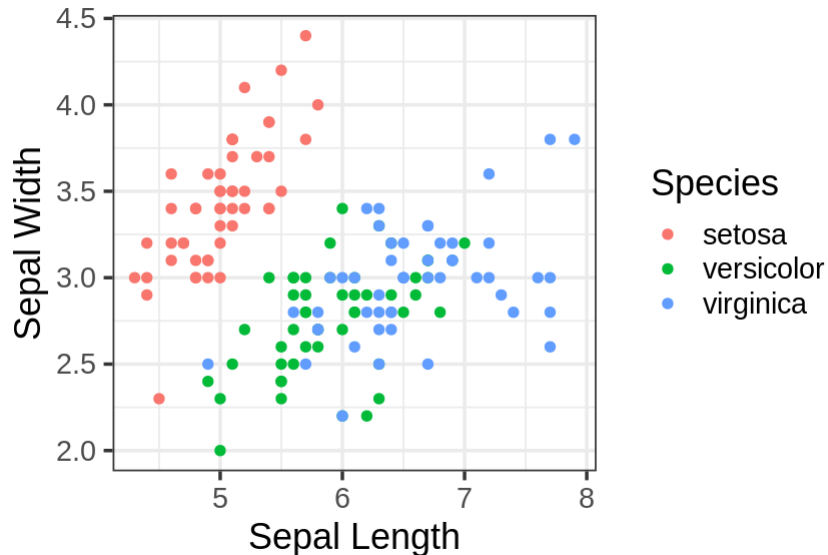
# Interactive plots • `highcharter`

R package `highcharter` is a wrapper around javascript library `highcharts` .

```r
library(highcharter)
h <- iris %>%
  hchart("scatter",hcaes(x="Sepal.Length",y="Sepal.Width",group="Species")) %>%
  hc_xAxis(title=list(text="Sepal Length"),crosshair=TRUE) %>%
  hc_yAxis(title=list(text="Sepal Width"),crosshair=TRUE) %>%
  hc_chart(zoomType="xy",inverted=FALSE) %>%
  hc_legend(verticalAlign="top",align="right") %>% hc_size(height=320)

htmltools::tagList(list(h))
```
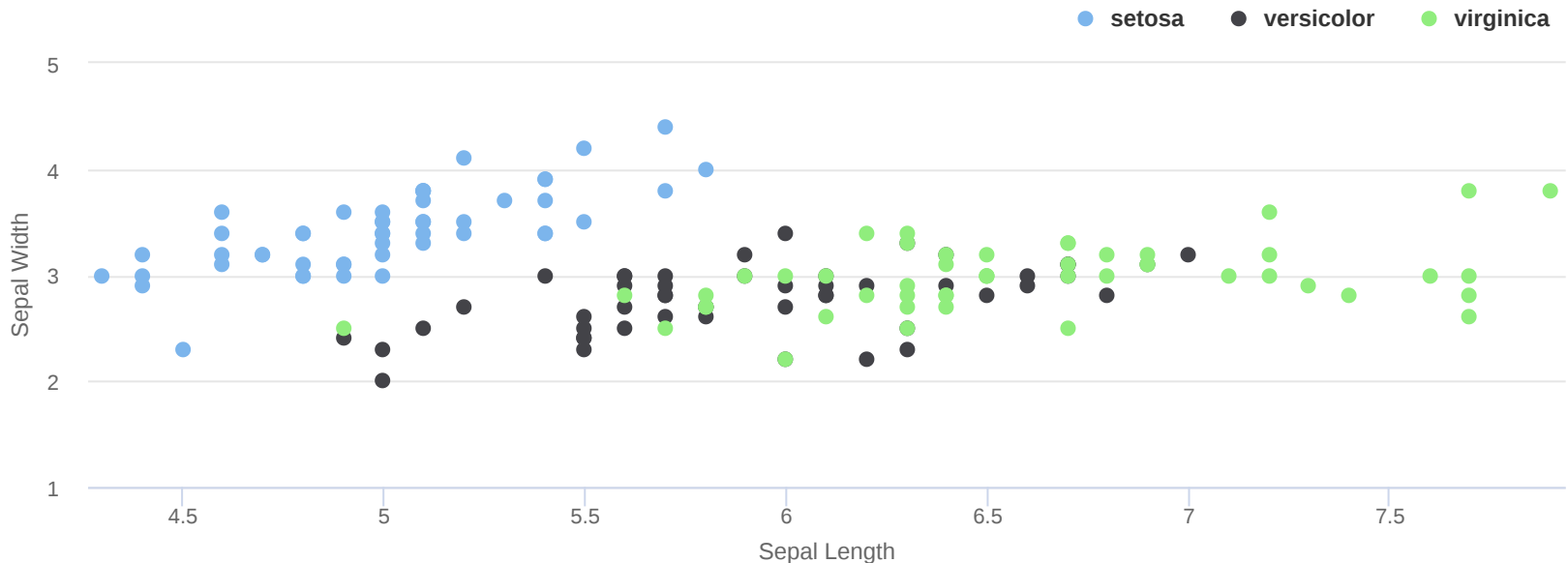
# Interactive plots • `plotly`

R package `plotly` provides R binding around javascript plotting library `plotly` .

```r
library(plotly)
p <- iris %>%
  plot_ly(x=~Sepal.Length,y=~Sepal.Width,color=~Species,width=500,height=350) %>%
  add_markers()
p
```
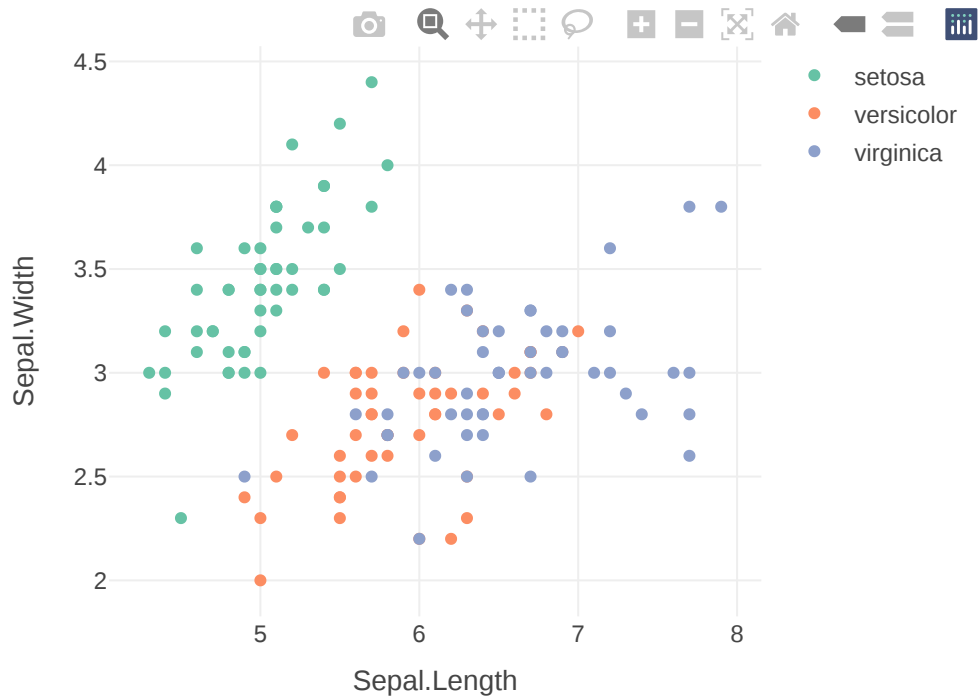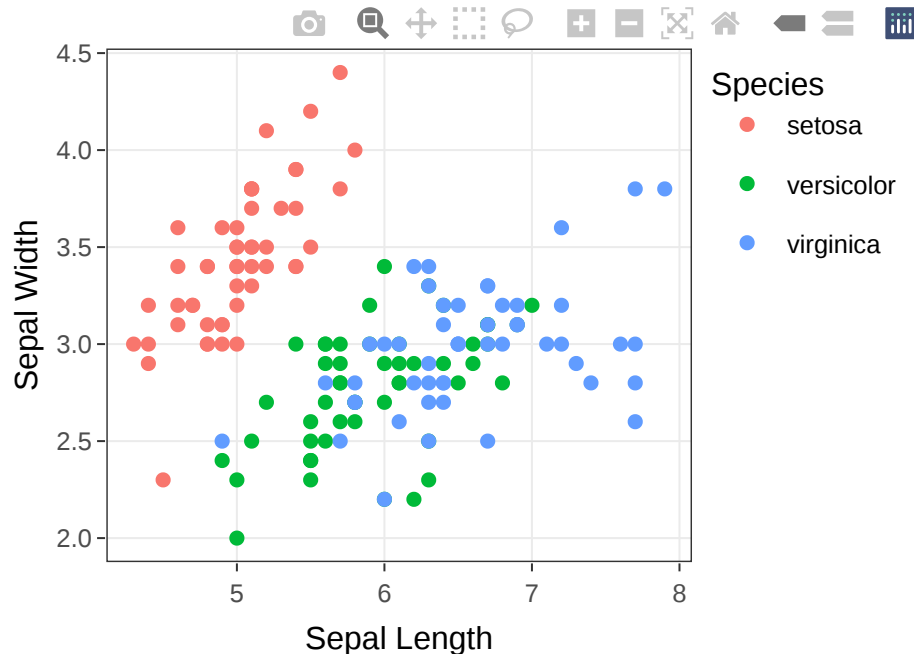
`plotly` also has a function called `ggplotly` which converts a static ggplot2 object into an interactive plot.

```r
library(plotly)
p <- ggplot(iris,aes(x=Sepal.Length,y=Sepal.Width,col=Species))+
  geom_point()+
  labs(x="Sepal Length",y="Sepal Width")+
  theme_bw(base_size=12)

plotly::ggplotly(p,width=460,height=330)
```

# Interactive plots • `ggiraph`

R package `ggiraph` converts a static ggplot2 object into an interactive plot.

```
library(ggiraph)
p <- ggplot(iris,aes(x=Sepal.Length,y=Petal.Length,colour=Species))+
      geom_point_interactive(aes(tooltip=paste0("<b>Petal Length:</b> ",Petal.Length,"\n<b>
  theme_bw()

tooltip_css <- "background-color:#f8f9f9;padding:10px;border-style:solid;border-width:2px;b

ggiraph(code=print(p),hover_css="cursor:pointer;stroke:black;fill-opacity:0.3",zoom_max=5,t
```
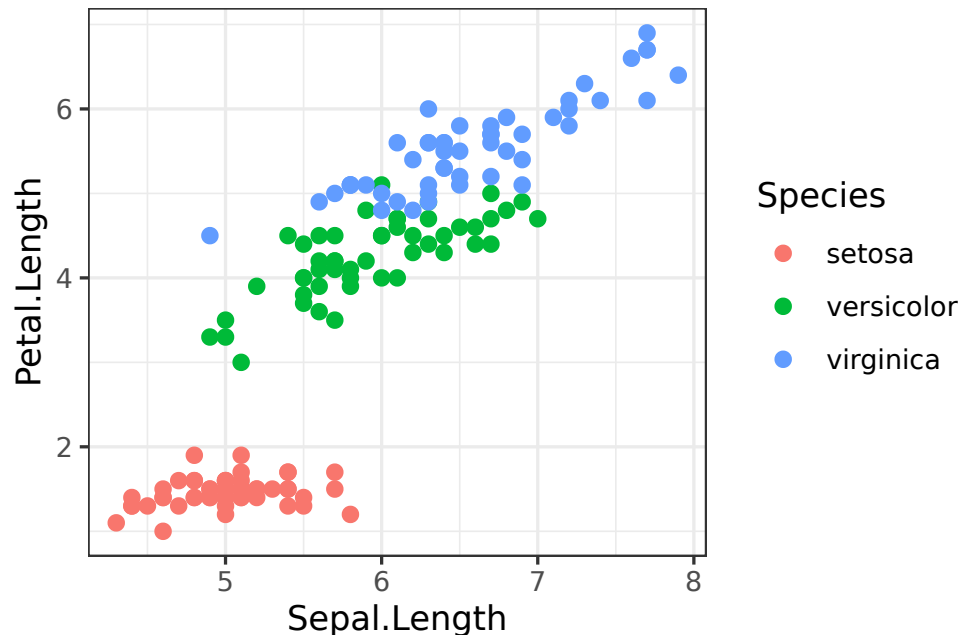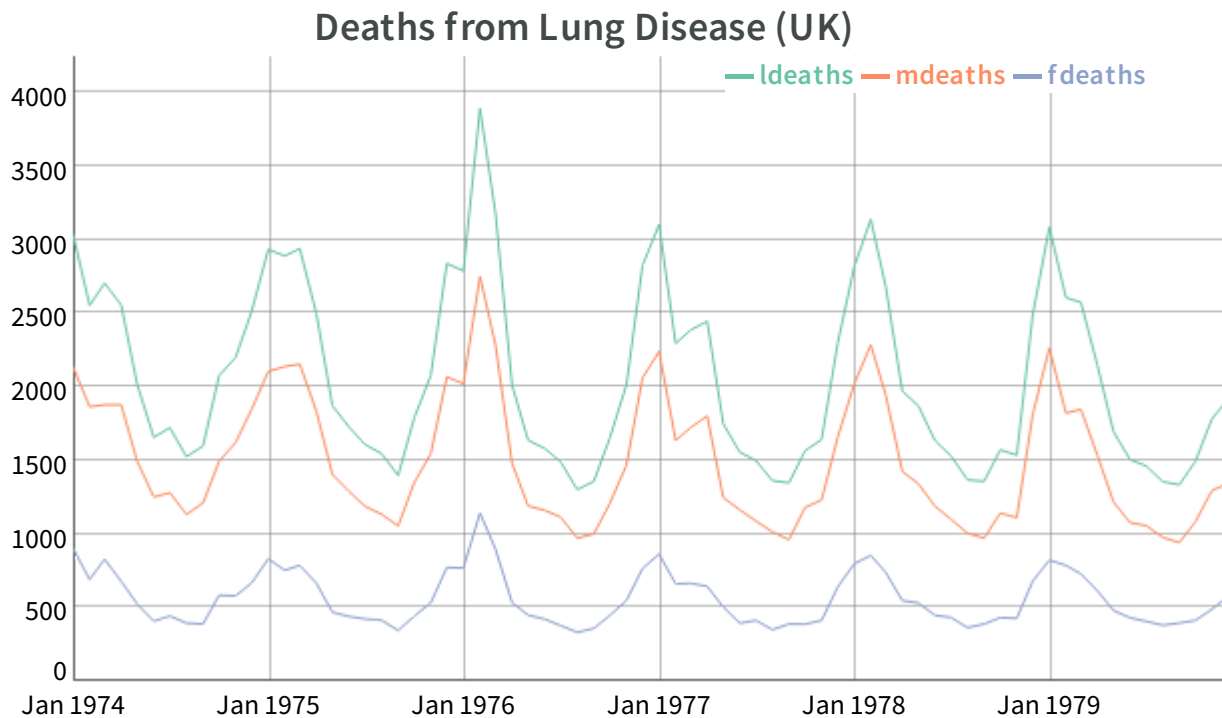
# Interactive time series • `dygraphs`

R package `dygraphs` provides R bindings for javascript library dygraphs for time series data.

```r
library(dygraphs)
lungDeaths <- cbind(ldeaths, mdeaths, fdeaths)
dygraph(lungDeaths,main="Deaths from Lung Disease (UK)") %>%
  dyOptions(colors=c("#66C2A5","#FC8D62","#8DA0CB"))
```



Deaths from Lung Disease (UK)

# Network graph

R package `networkD3` allows the use of interactive network graphs from the D3.js javascript library.

```r
library(networkD3)
data(MisLinks,MisNodes)
forceNetwork(Links=MisLinks,Nodes=MisNodes,Source="source",
             Target="target",Value="value",NodeID="name",
             Group="group",opacity=0.4,
             height=350,width=600)
```
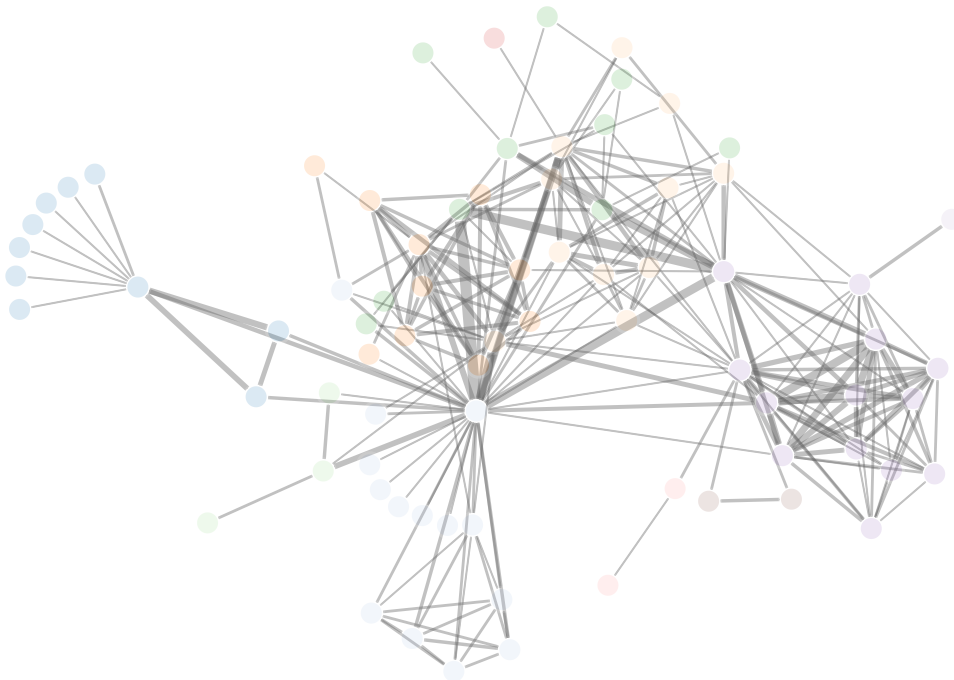
# Interactive maps • `leaflet`

R package `leaflet` provides R bindings for javascript mapping library; leafletjs.
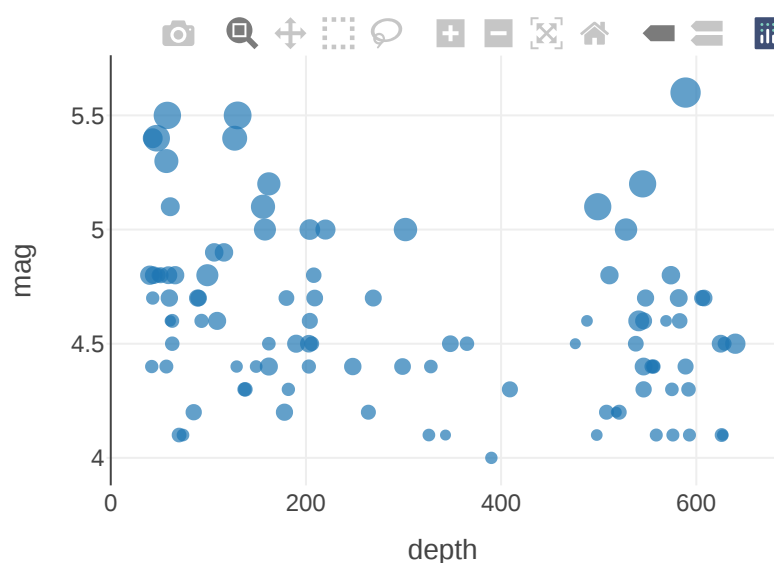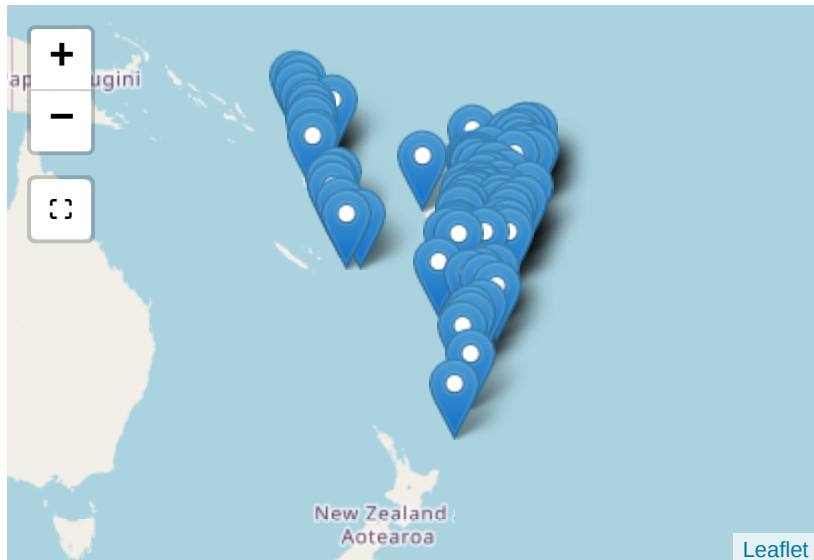
```r
library(leaflet)
leaflet(height=350,width=650) %>%
  addTiles(urlTemplate='http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png') %>%
  #addProviderTiles(providers$Esri.NatGeoWorldMap) %>%
  addMarkers(lat=57.639327,lng=18.288534,popup="RaukR") %>%
  setView(lat=57.639327,lng=18.288534,zoom=15)
```
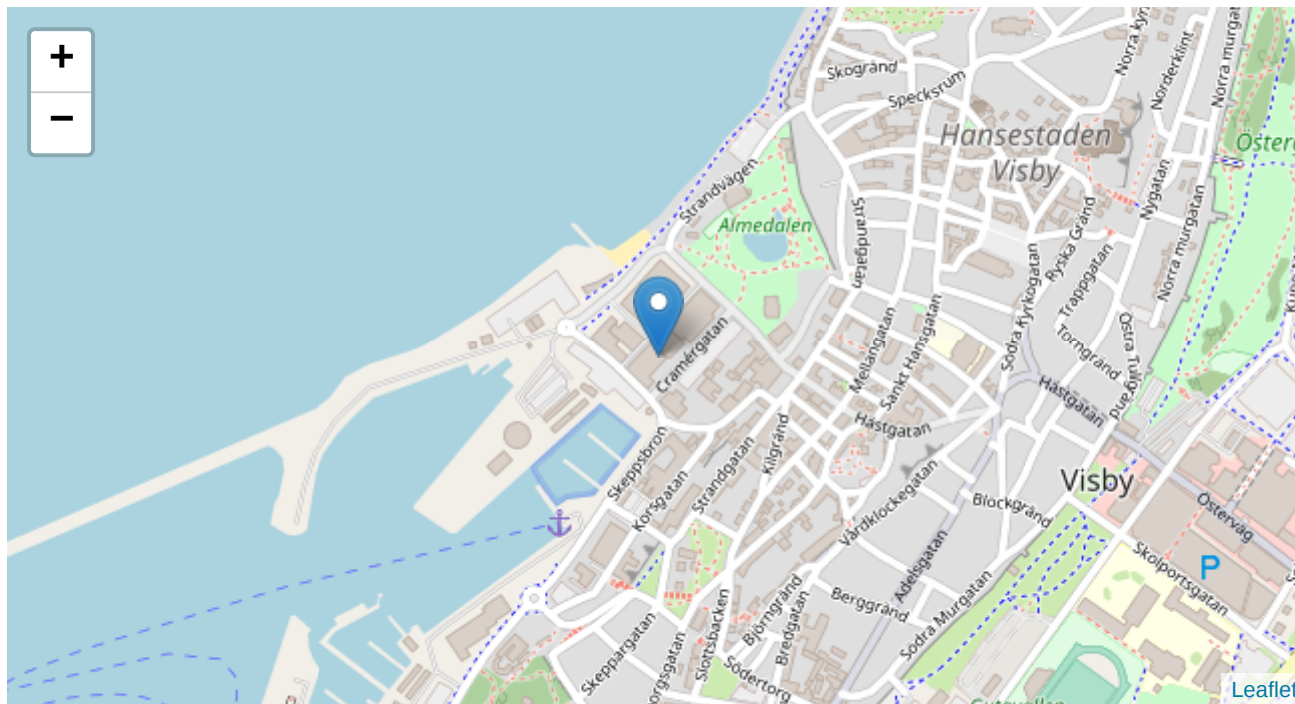
# Linking Plots • `crosstalk`

R package `crosstalk` allows crosstalk enabled plotting libraries to be linked. Through the shared 'key' variable, data points can be manipulated simultaneously on two independent plots.

```r
library(crosstalk)
shared_quakes <- SharedData$new(quakes[sample(nrow(quakes), 100),])
lf <- leaflet(shared_quakes,height=280) %>%
        addTiles(urlTemplate='http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png') %>%
        addMarkers()
py <- plot_ly(shared_quakes,x=~depth,y=~mag,size=~stations,height=280) %>%
        add_markers()
div(div(lf,style="float:left;width:49%"),div(py,style="float:right;width:49%"))
```

# Thank you. Questions?