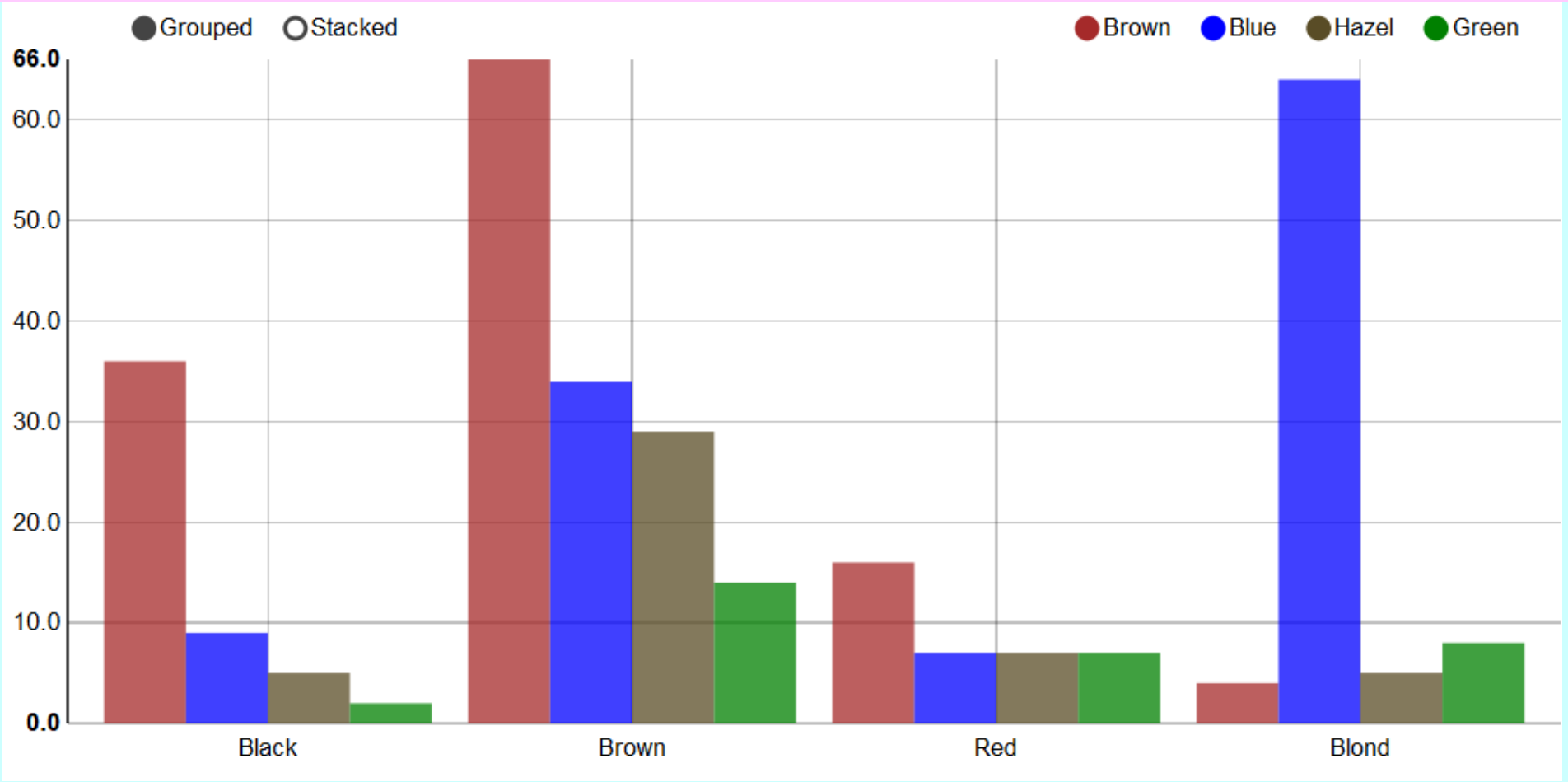
Interactive Graphics



1. Interactive Graphics with googleVis
2. Interactive Graphics with ggvis
3. Interactive Graphics with ggplot
4. Interactive Charts with rCharts
5. Interactive 3D Graphics

In order to interact with the report figures, one can create an interactive charts with googleVis.

`googleVis-package` *R Interface to Google Charts*

R interface to Google Charts API, allowing users to create interactive charts based on data frames. Charts are displayed locally via the R HTTP help server. A modern browser with Internet connection is required and for some charts Flash. The data remains local and is not uploaded to Google.

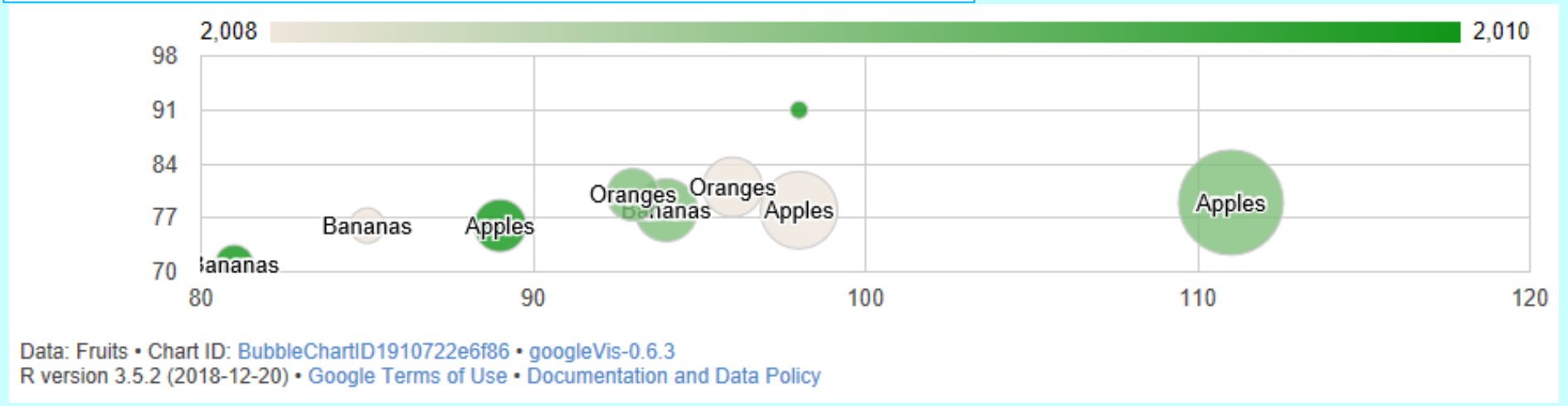
gvisAnnotatedTimeLine	gvisIntensityMap
gvisAnnotationChart	gvisLineChart
gvisAreaChart	gvisMap
gvisBarChart	gvisMerge
gvisBubbleChart	gvisMotionChart
gvisCalendar	gvisOrgChart
gvisCandlestickChart	gvisPieChart
gvisColumnChart	gvisSankey
gvisComboChart	gvisScatterChart
gvisGauge	gvisSteppedAreaChart
gvisGeoChart	gvisTable
gvisGeoMap	gvisTimeline
gvisHistogram	gvisTreeMap

gvisBubbleChart(data,idvar,xvar,yvar,colorvar,sizevar, ...) {googleVis}

The gvisBubbleChart function reads a data.frame and creates text output referring to the Google Visualisation API.

```
> library(googleVis)
> head(Fruits)
    Fruit Year Location Sales Expenses Profit       Date
1  Apples 2008     West    98       78     20 2008-12-31
2  Apples 2009     West   111       79     32 2009-12-31
3  Apples 2010     West    89       76     13 2010-12-31
4 Oranges 2008     East    96       81     15 2008-12-31
5 Bananas 2008     East    85       76      9 2008-12-31
6 Oranges 2009     East    93       80     13 2009-12-31
```

```
bubble <- gvisBubbleChart(Fruits, idvar='Fruit',
          xvar='Sales', yvar='Expenses',
          colorvar='Year', sizevar='Profit')
plot(bubble)
```

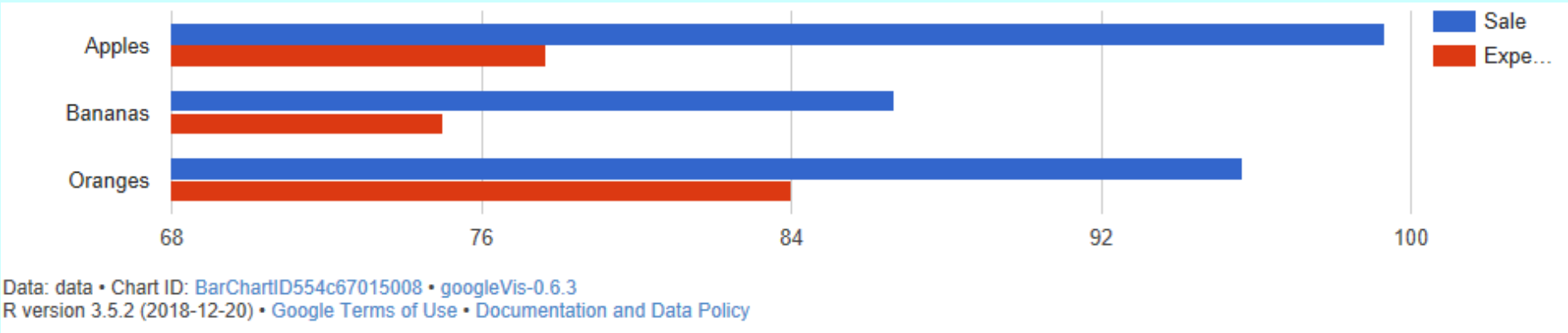


gvisBarChart(data, xvar, yvar, ...) {googleVis}

This function reads a data.frame and creates text output referring to the Google Visualisation API, which can be included into a web page, or as a stand-alone page.

```
> library(dplyr)
> dat <- Fruits %>% group_by(Fruit) %>%
+   summarize(Sale=mean(Sales),Expense=mean(Expenses))
> dat
# A tibble: 3 x 3
  Fruit    Sale Expense
  <fct>   <dbl>   <dbl>
1 Apples   99.3    77.7
2 Bananas  86.7    75
3 Oranges  95.7    84
```

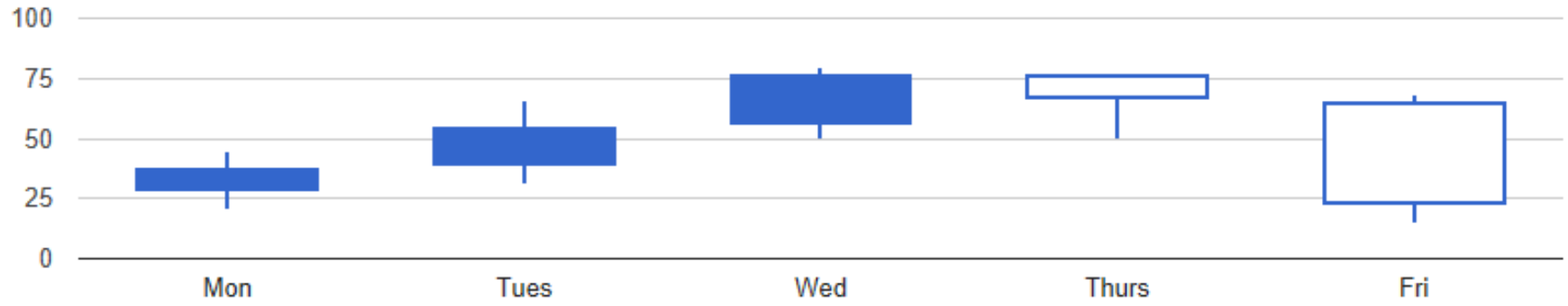
```
bar <- gvisBarChart(dat, xvar="Fruit", yvar=c("Sale","Expense"))
plot(bar)
```



gvisCandlestickChart(data, xvar, low, open, close, high, options = list(), ...) {googleVis}
Google interactive candlestick chart with R

```
> OpenClose
  Weekday Low Open Close High
1     Mon  20   28    38   45
2    Tues  31   38    55   66
3     Wed  50   55    77   80
4   Thurs  50   77    66   77
5     Fri  15   66    22   68
```

```
ck <- gvisCandlestickChart(OpenClose, xvar="Weekday",
        low="Low", open="Open", close="Close",
        high="High", options=list(legend='none'))
plot(ck)
```



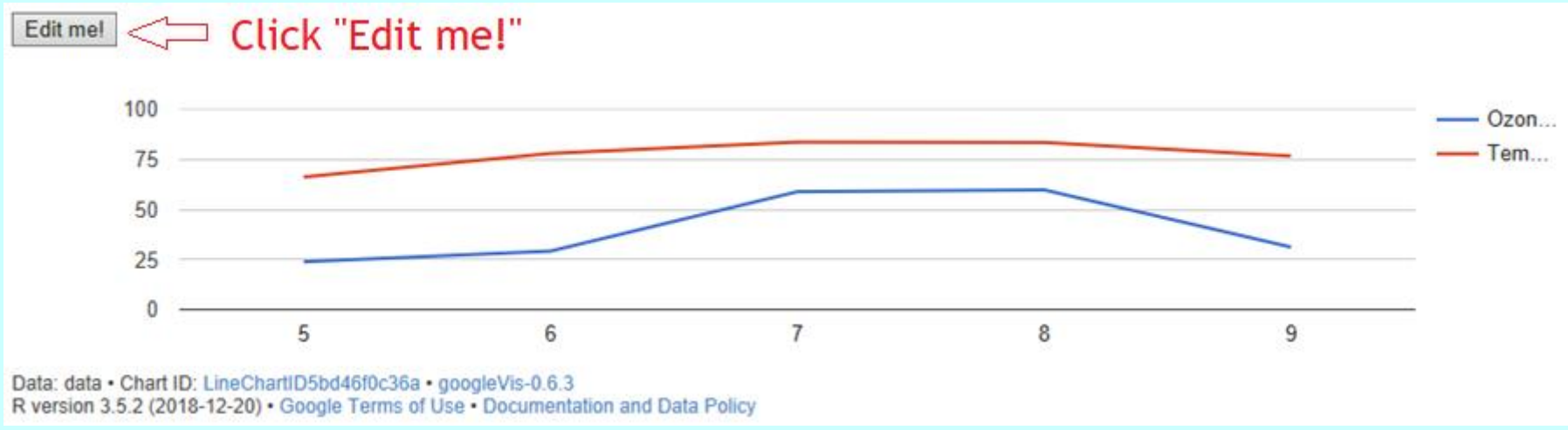
Data: data • Chart ID: CandlestickChartID54b47edc3fe1 • googleVis-0.6.3
R version 3.5.2 (2018-12-20) • Google Terms of Use • Documentation and Data Policy

gvisLineChart(data, xvar, yvar, options, ...) {googleVis}

This function reads a data.frame and creates text output referring to the Google Visualisation API, which can be included into a web page, or as a stand-alone page.

```
> library(dplyr)
> aq <- na.omit(airquality) %>% select(Ozone,Temp,Month) %>%
+   group_by(Month) %>%
+   summarize(OzoneMean=mean(Ozone),TempMean=mean(Temp))
> aq <- data.frame(aq); head(aq)
  Month OzoneMean TempMean
1     5  24.12500 66.45833
2     6  29.44444 78.22222
3     7  59.11538 83.88462
4     8  60.00000 83.69565
5     9  31.44828 76.89655
```

```
Line <- gvisLineChart(aq, xvar="Month", yvar=c("OzoneMean","TempMean"),
                      options=list(gvis.editor="Edit me!"))
plot(Line)
```



2. Interactive Graphics with ggvis

In order to interact with the report figures, one can create an interactive graphic with ggvis.

Package 'ggvis'

Title Interactive Grammar of Graphics

Description An implementation of an interactive grammar of graphics, taking the best parts of 'ggplot2', combining them with the reactive framework of 'shiny' and drawing web graphics using 'vega'.

ggvis(data, ...,) {ggvis}
ggvis is used to turn a dataset into a visualisation, setting up default mappings between variables in the dataset and visual properties.

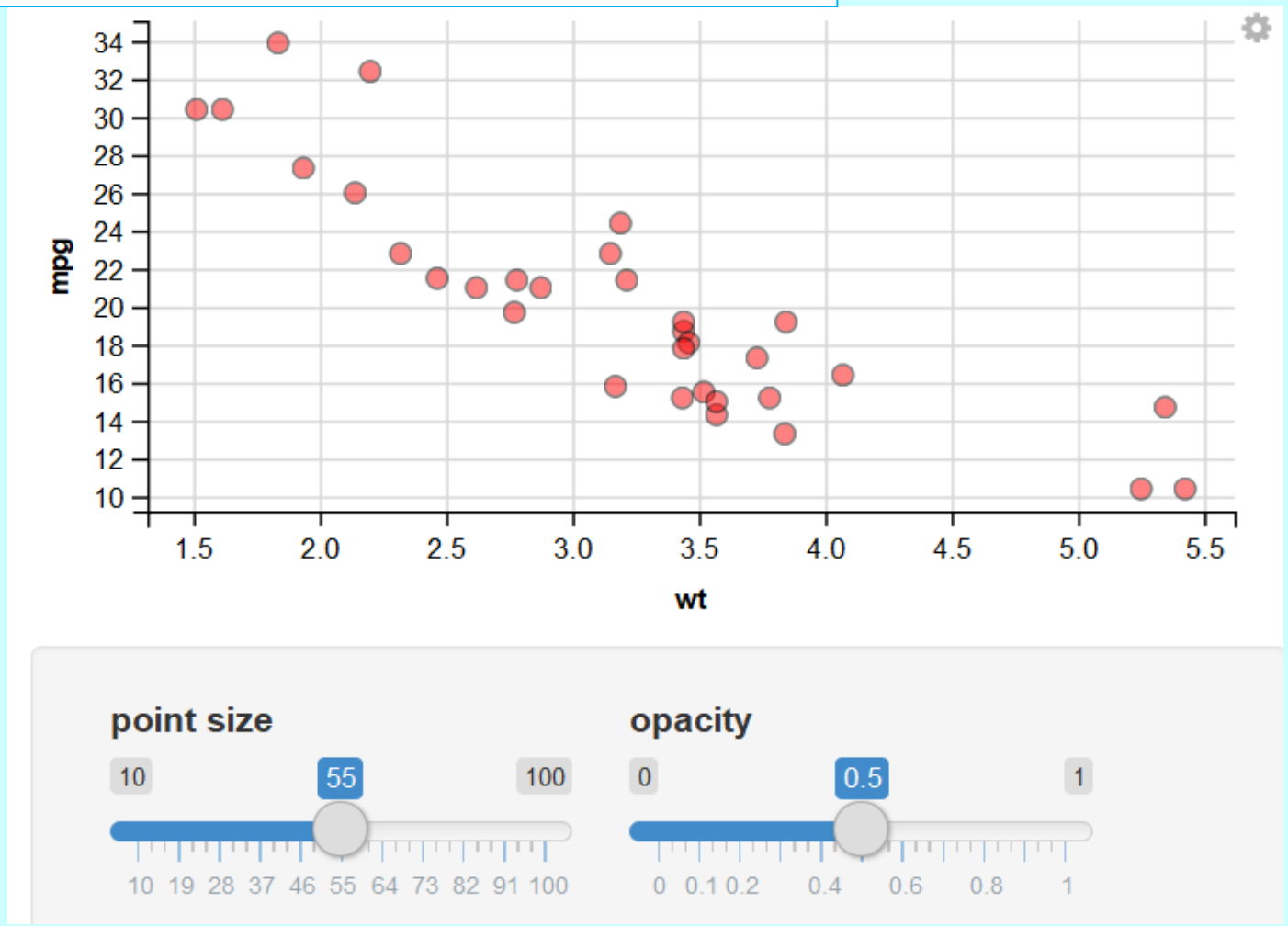
Basic interactive controls

- input_slider() : create an interactive slider
- `input_checkbox()` : a check-box
- `input_checkboxgroup()` : a group of check boxes
- `input_numeric()` : a spin box
- `input_radiobuttons()` : pick one from a set options
- `input_select()` : create a drop-down text box
- `input_text()` : arbitrary text input

[Reference] Chiu Yu-Wei, *R for Data Science Cookbook* (Packt Pub., Birmingham, 2016) pp.190-207.

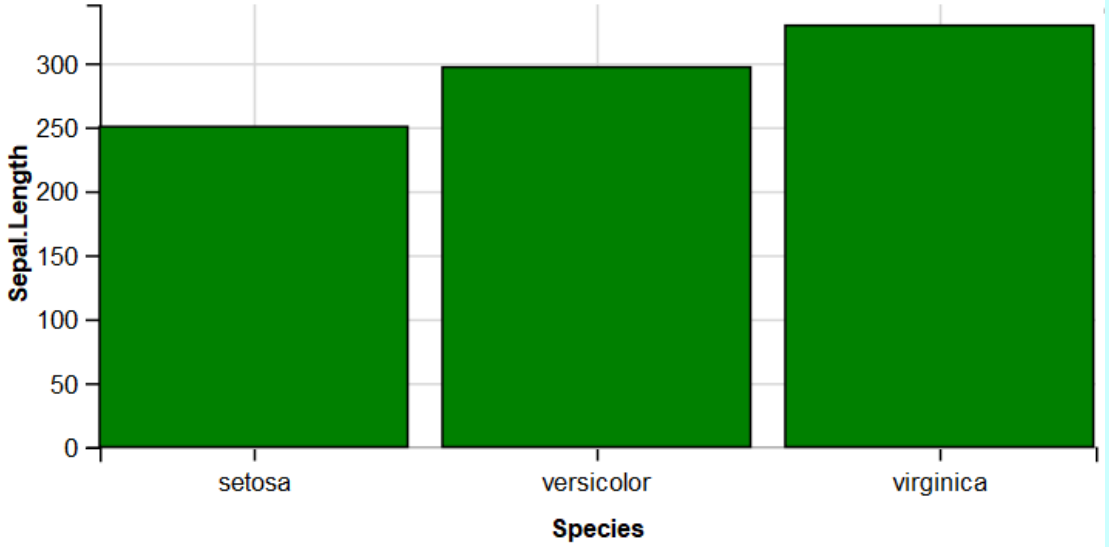
Slider interactive controls

```
#Fig. 1
library(ggvis)
mtcars %>%
  ggvis(~wt, ~mpg, fill:="red", stroke:="black",
        size:=input_slider(10,100,label="point size"),
        opacity:=input_slider(0,1,label="opacity")) %>%
  layer_points()
```

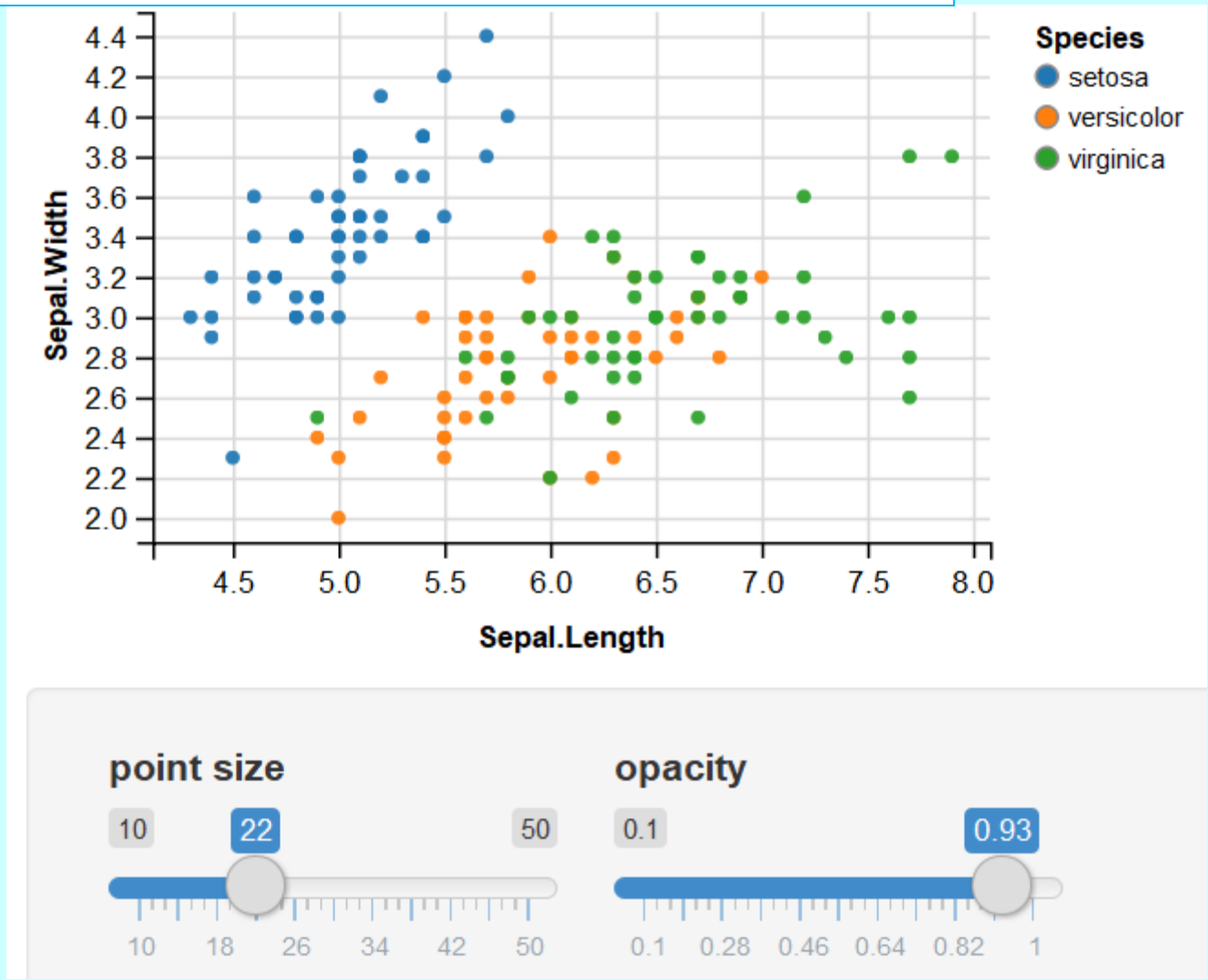


```
> library(dplyr)
> iris  %>% group_by(Species) %>%
+   summarize(SubTotal=sum(Sepal.Length))
# A tibble: 3 x 2
   Species    SubTotal
   <fct>         <dbl>
1 setosa         250.
2 versicolor     297.
3 virginica      329.
```

```
library(ggvis)
iris %>%
  ggvis(~Species, ~Sepal.Length, fill:=input_select(c("red","green","blue"),
        label="Fill Color")) %>% layer_bars()
```



```
#Fig. 3
iris %>%
  ggvis(~Sepal.Length,~Sepal.Width,fill=~Species,
        size:=input_slider(10,50,label="point size"),
        opacity:=input_slider(0.1,1,label="opacity")) %>%
  layer_points()
```



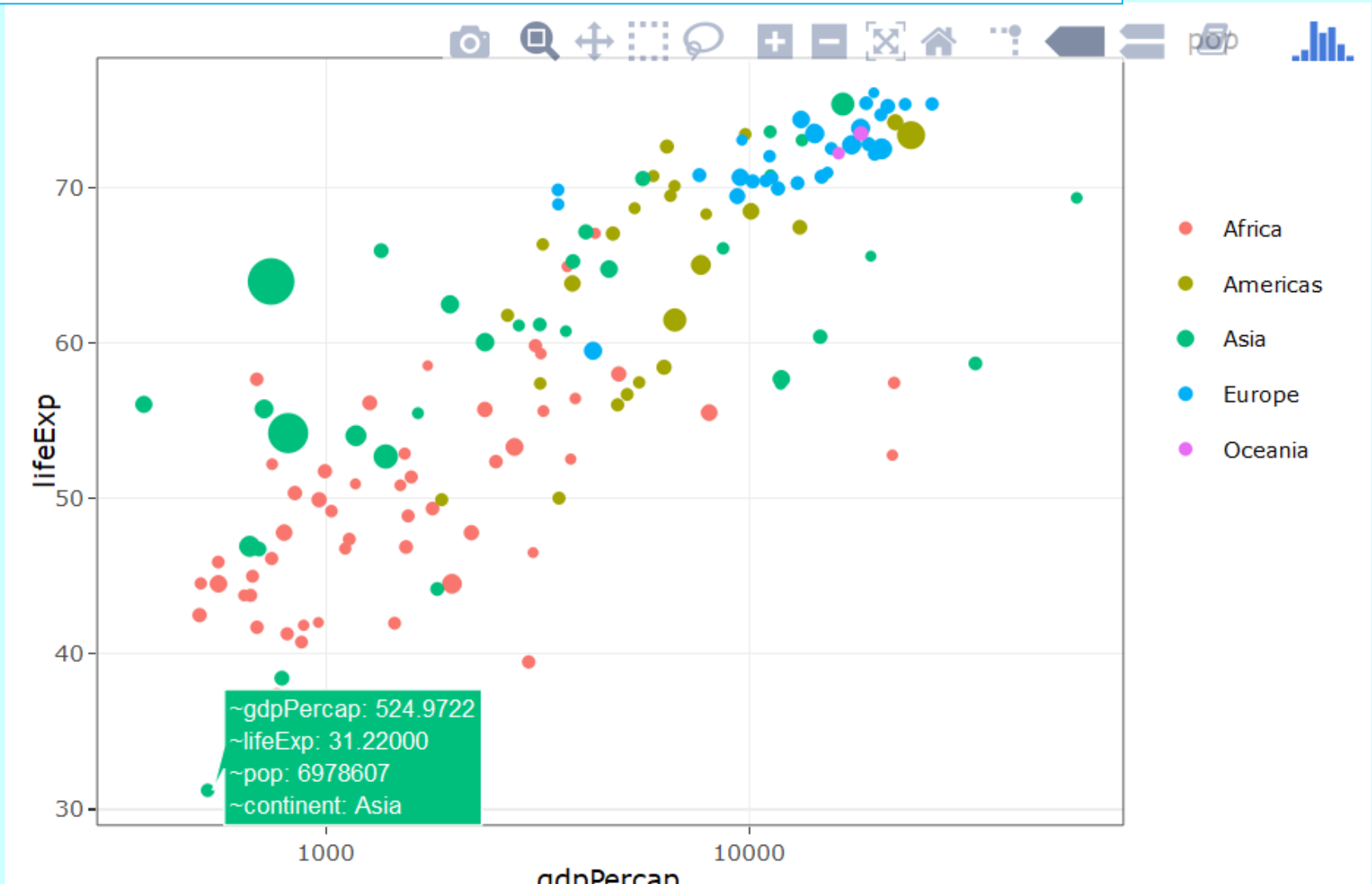
Sample data: gapminder

gapminder {gapminder}

Excerpt of the Gapminder data on life expectancy, GDP per capita, and population by country. The main data frame gapminder has 1704 rows and 6 variables.

```
> library(gapminder)
> str(gapminder)
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':       1704 obs. of  6 variables:
 $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3
 $ year     : int  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
 $ lifeExp  : num  28.8 30.3 32 34 36.1 ...
 $ pop      : int  8425333 9240934 10267083 11537966 13079460 14880372 128818
67957 16317921 22227415 ...
 $ gdpPercap: num  779 821 853 836 740 ...
> head(gapminder)
# A tibble: 6 x 6
  country     continent  year lifeExp      pop gdpPercap
  <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
1 Afghanistan Asia       1952    28.8  8425333      779.
2 Afghanistan Asia       1957    30.3  9240934      821.
3 Afghanistan Asia       1962    32.0 10267083      853.
4 Afghanistan Asia       1967    34.0 11537966      836.
5 Afghanistan Asia       1972    36.1 13079460      740.
6 Afghanistan Asia       1977    38.4 14880372      786.
```

```
library(ggplot2); library(plotly)
gap <- gapminder %>%
  filter(year==1977) %>%
  ggplot(aes(x=gdpPercap, y=lifeExp, size=pop, color=continent)) +
  geom_point() + scale_x_log10() + theme_bw()
ggplotly(gap)
```



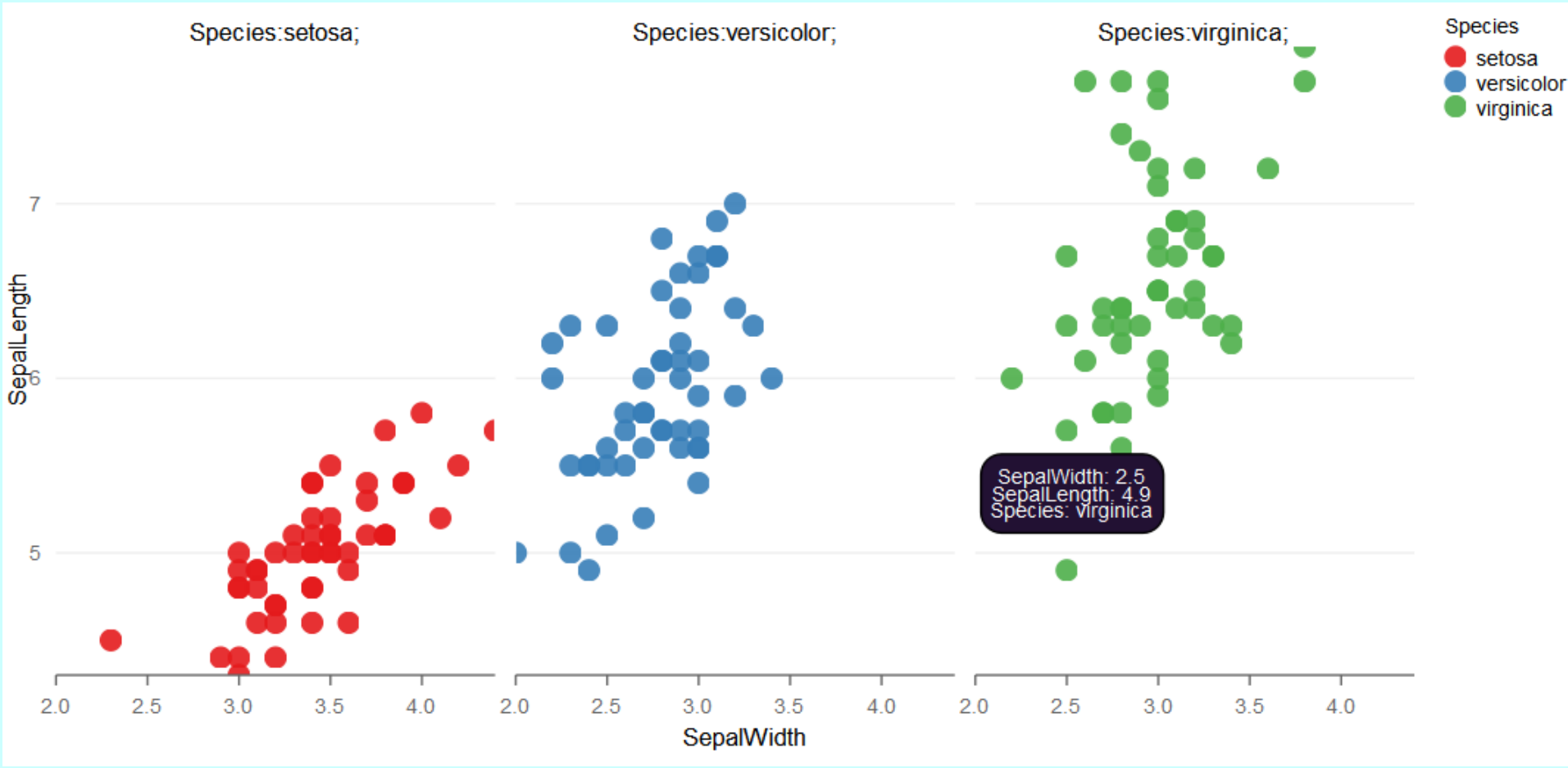
rCharts is an R package to create, customize and publish interactive javascript visualizations from R.

```
> install.packages("rCharts")
Installing package into 'C:/Users/user/Documents/R/win-library/3.5'
(as 'lib' is unspecified)
Warning in install.packages :
  package 'rCharts' is not available (for R version 3.5.2)
```

```
## How to install rCharts package?
library(devtools)
library(Rcpp)
install_github('ramnathv/rCharts', force= TRUE)
```

```
# Example 1: scatterplot
library(rCharts)
names(iris) = gsub("\\.","",names(iris))
rPlot(SepalLength~SepalWidth | Species,data=iris,type='point',color='Species')
```

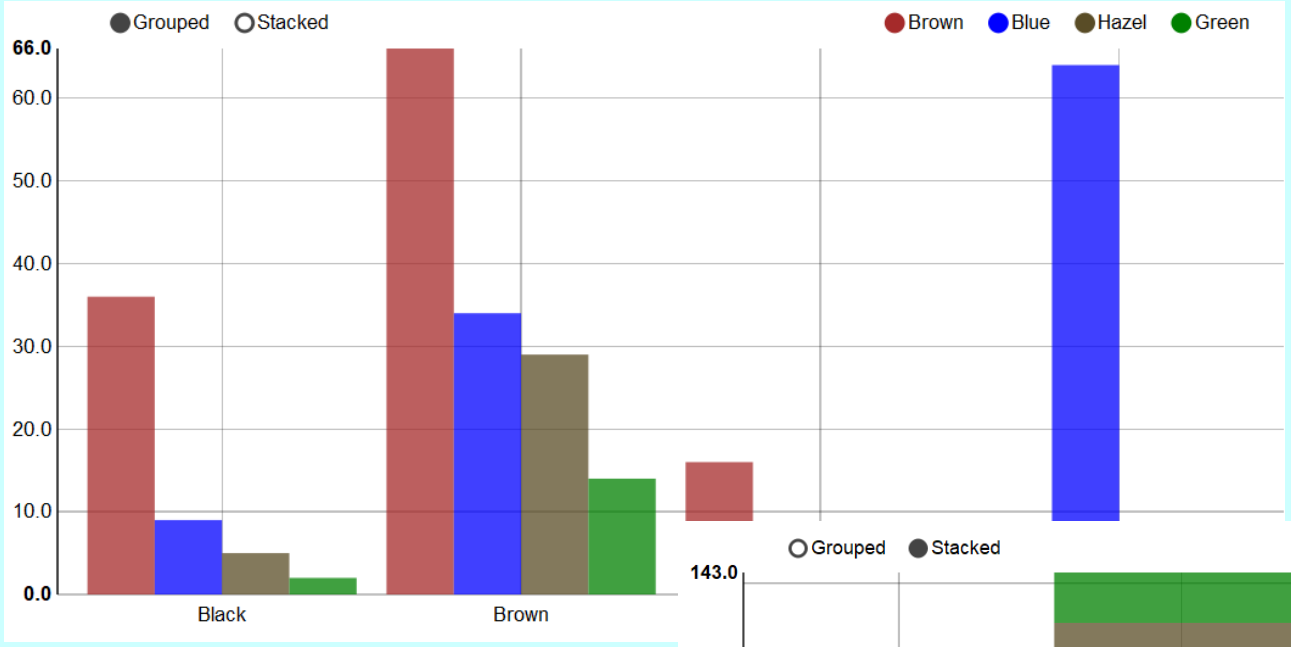
rPlot(x, ...) {rCharts} Main plotting function

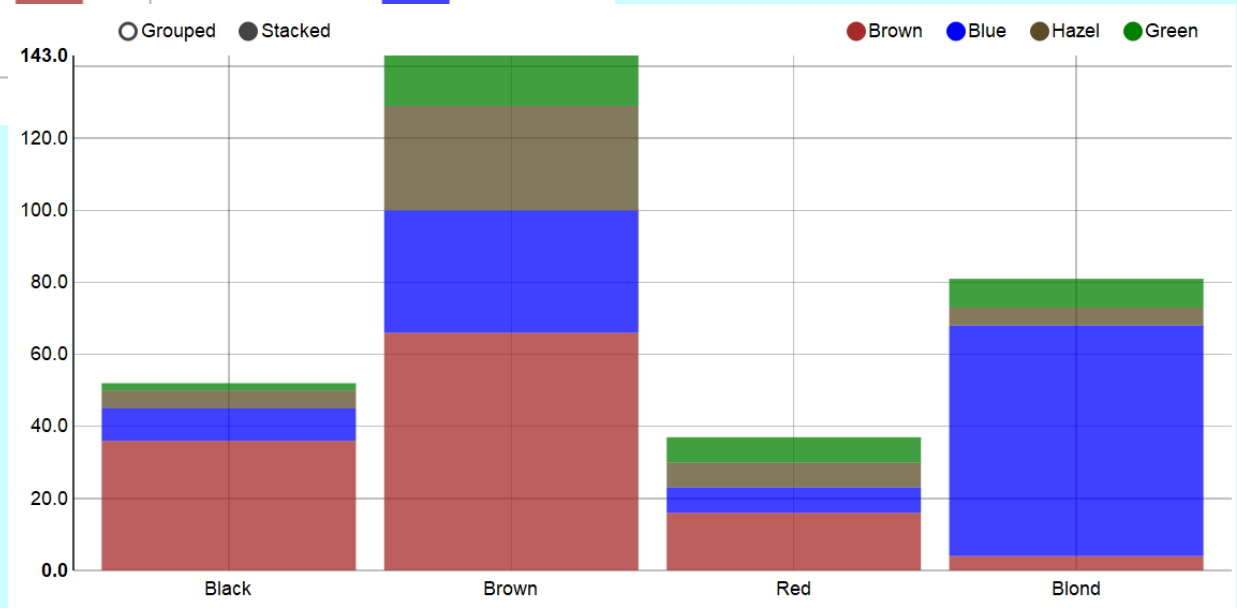


Example 2: interactive multibar chart

```
> #[Sample dataset] HairEyeColor
> # HairEyeColor is a three way table defined by categorical variables:
> # Hair, Eye, and Sex
> str(HairEyeColor)
 'table' num [1:4, 1:4, 1:2] 32 53 10 3 11 50 10 30 10 25 ...
 - attr(*, "dimnames")=List of 3
  ..$ Hair: chr [1:4] "Black" "Brown" "Red" "Blond"
  ..$ Eye : chr [1:4] "Brown" "Blue" "Hazel" "Green"
  ..$ Sex : chr [1:2] "Male" "Female"
> hec <- as.data.frame(HairEyeColor)
> str(hec)
'data.frame':   32 obs. of  4 variables:
 $ Hair: Factor w/ 4 levels "Black","Brown",..: 1 2 3 4 1 2 3 4 1 2 ..
 $ Eye : Factor w/ 4 levels "Brown","Blue",..: 1 1 1 1 2 2 2 2 3 3 ...
 $ Sex : Factor w/ 2 levels "Male","Female": 1 1 1 1 1 1 1 1 1 1 ...
 $ Freq: num  32 53 10 3 11 50 10 30 10 25 ...
> head(hec)
   Hair   Eye  Sex Freq
1 Black Brown Male   32
2 Brown Brown Male   53
3   Red Brown Male   10
4 Blond Brown Male    3
5 Black  Blue Male   11
6 Brown  Blue Male   50
```

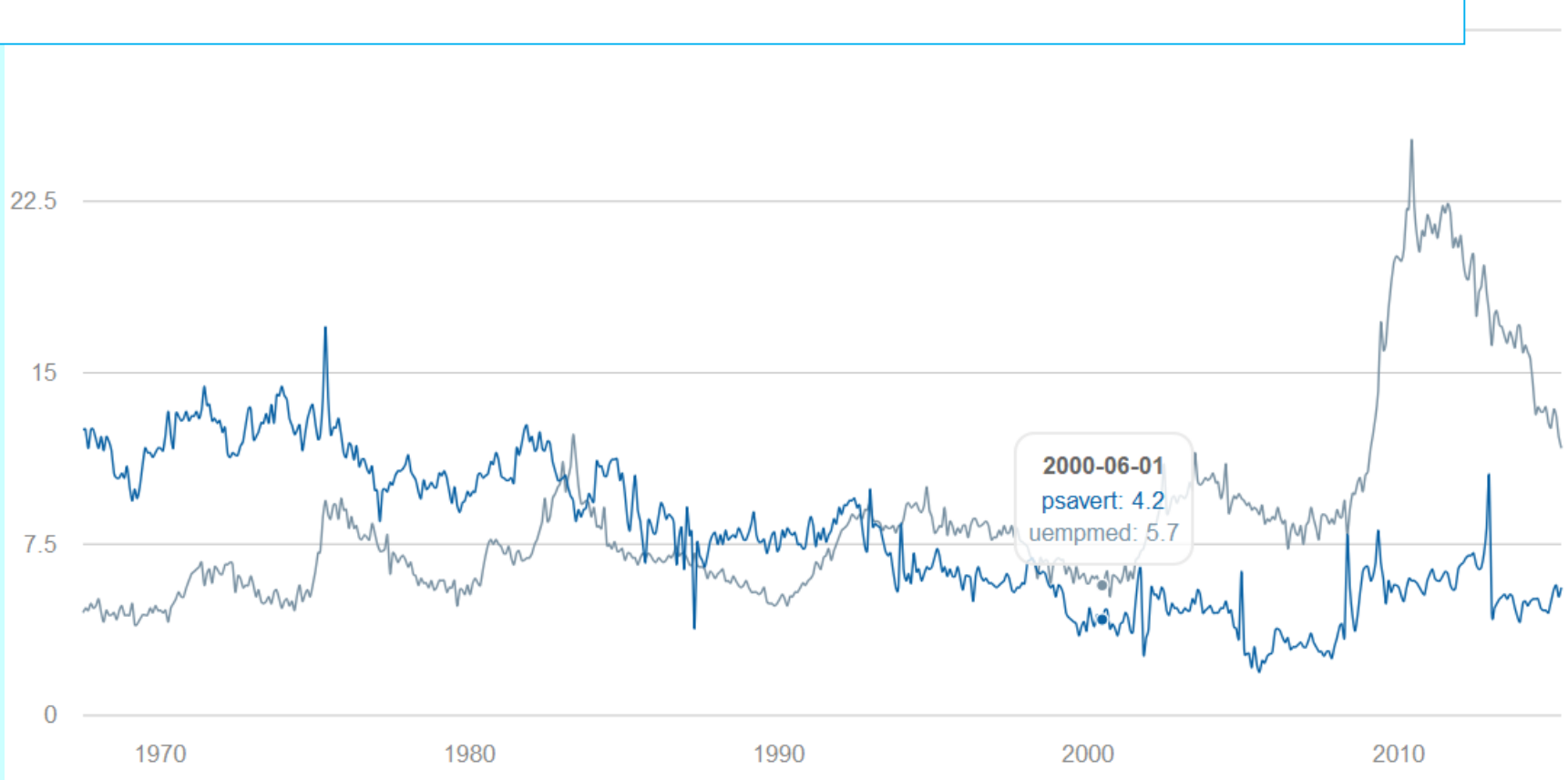
```
# Example 2: interactive multibar chart
p2 <- nPlot(Freq~Hair, group="Eye",
            data=subset(hec, Sex=="Female"),
            type="multiBarChart")
p2$chart(color=c("brown","blue","#594c26","green"))
p2
```





```
> ## Example 3: Multi Line chart
> data(economics, package = 'ggplot2')
> head(economics,2)
# A tibble: 2 x 6
  date         pce    pop psavert uempmed unemploy
  <date>     <dbl>  <int>   <dbl>   <dbl>    <int>
1 1967-07-01  507. 198712    12.5     4.5     2944
2 1967-08-01  510. 198911    12.5     4.7     2945
> econ = transform(economics, date = as.character(date))
```

```
m1 = mPlot(x='date', y=c('psavert','uempmed'), type='Line', data=econ)
m1$set(pointSize = 0, lineWidth = 1)
#m1$print(include_assets=TRUE)
m1
```



```
> ## Example 4: Bubble Chart
> data(survey,package="MASS")
> head(survey,2)
     Sex Wr.Hnd NW.Hnd W.Hnd   Fold Pulse Clap Exer Smoke Height       M.I    Age
1 Female   18.5   18.0 Right R on L    92 Left Some Never  173.0    Metric 18.250
2   Male   19.5   20.5  Left R on L   104 Left None Regul  177.8 Imperial 17.583
```

```
h2 = hPlot(Pulse~Height, data=survey, type="bubble", title="Zoom demo",
           subtitle="bubble chart", size="Age", group="Exer")
h2$chart(zoomType = "xy")
h2$exporting(enabled = FALSE)
#h2$print(include_assets=TRUE)
h2
```