



22 May 2018

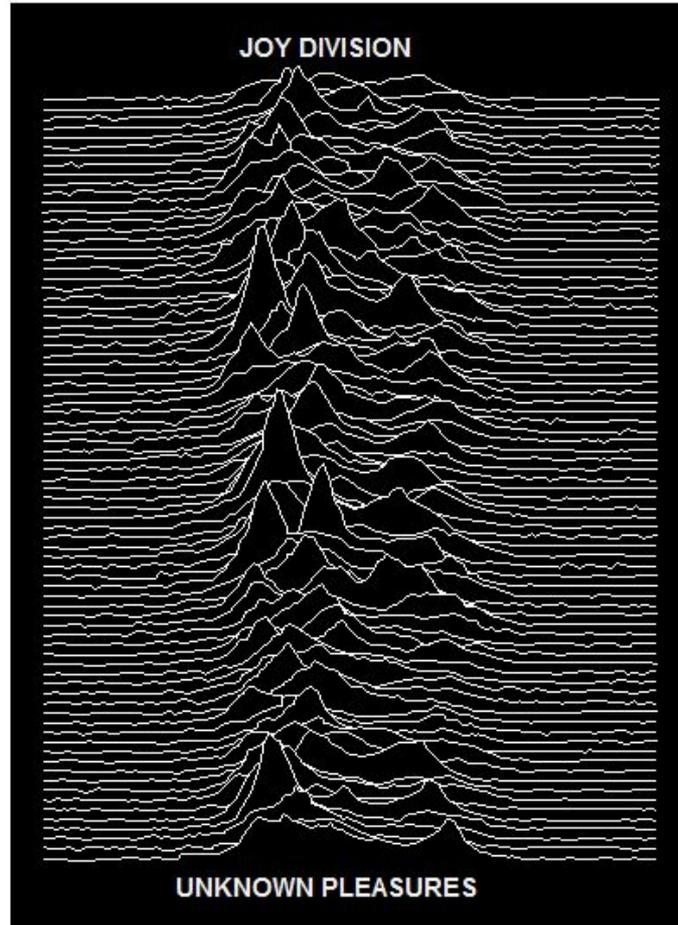
Herman Teirlinck,
01.05 - Isala Van Diest

What have I done?!?

```
library(tidyverse)
library(ggplot2)
library(ggribes)

pulsar <- read_csv("../data/20180522_pulsar_cp1919.csv")

ggplot(pulsar, aes(x, y, height = y, group = line)) +
  geom_ridgeline(fill = "black", colour = "white", size = 0.4) +
  coord_fixed(0.5) +
  annotate("text", x = median(pulsar$x), y = -50, label = "UNKNOWN PLEASURES",
         colour = "#F0F0F0", hjust = 0.5, fontface = 2) +
  annotate("text", x = median(pulsar$x), y = 1650, label = "JOY DIVISION",
         colour = "#F0F0F0", hjust = 0.5, fontface = 2) +
  theme(panel.grid = element_blank(),
        panel.background = element_rect(fill = "black"),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        axis.title = element_blank())
```





DATA

VISUALIZATION

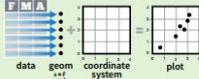
Data Visualization with ggplot2

Cheat Sheet

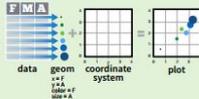


Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same few components: a **data set**, a set of **geoms**—visual marks that represent data points, and a **coordinate system**.



To display data values, map variables in the data set to aesthetic properties of the geom like **size**, **color**, and **x** and **y** locations.



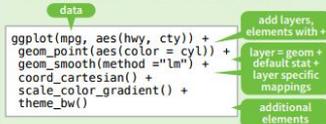
Build a graph with **qplot()** or **ggplot()**

aesthetic mappings + **data** + **geom**

qplot(x = cty, y = hwy, color = cyl, data = mpg, geom = "point")
Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

ggplot(data = mpg, aes(x = cty, y = hwy))

Begins a plot that you finish by adding layers to. No defaults, but provides more control than **qplot()**.



Add a new layer to a plot with a **geom_***() or **stat_***() function. Each provides a geom, a set of aesthetic mappings, and a default stat and position adjustment.

last_plot()

Returns the last plot

Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

One Variable

Continuous

a <- ggplot(mpg, aes(hwy))



a + **geom_area(stat = "bin")**
x, y, alpha, color, fill, linetype, size



a + **geom_density(kernel = "gaussian")**
x, y, alpha, color, fill, linetype, size, weight



a + **geom_dotplot()**
x, y, alpha, color, fill



a + **geom_freqpoly()**
x, y, alpha, color, linetype, size
b + **geom_freqpoly(aes(y = ..density..))**



a + **geom_histogram(binwidth = 5)**
x, y, alpha, color, fill, linetype, size, weight
b + **geom_histogram(aes(y = ..density..))**

Discrete

b <- ggplot(mpg, aes(fill))



b + **geom_bar()**
x, alpha, color, fill, linetype, size, weight

Graphical Primitives

c <- ggplot(map, aes(long, lat))



c + **geom_polygon(aes(group = group))**
x, y, alpha, color, fill, linetype, size

d <- ggplot(economics, aes(date, unemploy))



d + **geom_path(lineend = "butt", linejoin = "round", linemitre = 1)**
x, y, alpha, color, linetype, size



d + **geom_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900))**
x, ymax, ymin, alpha, color, fill, linetype, size

e <- ggplot(seals, aes(x = long, y = lat))



e + **geom_segment(aes(xend = long + delta_long, yend = lat + delta_lat))**
x, xend, y, yend, alpha, color, linetype, size



e + **geom_rect(aes(xmin = long, ymin = lat, xmax = long + delta_long,**

Two Variables

Continuous X, Continuous Y

f <- ggplot(mpg, aes(cty, hwy))



f + **geom_blank()**



f + **geom_jitter()**
x, y, alpha, color, fill, shape, size



f + **geom_point()**
x, y, alpha, color, fill, shape, size



f + **geom_quantile()**
x, y, alpha, color, linetype, size, weight



f + **geom_rug(sides = "bl")**
alpha, color, linetype, size



f + **geom_smooth(model = lm)**
x, y, alpha, color, fill, linetype, size, weight



f + **geom_text(aes(label = cty))**
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

Discrete X, Continuous Y

g <- ggplot(mpg, aes(class, hwy))



g + **geom_bar(stat = "identity")**
x, y, alpha, color, fill, linetype, size, weight



g + **geom_boxplot()**
lower, middle, upper, x, ymax, ymin, alpha, color, fill, linetype, shape, size, weight



g + **geom_dotplot(binaxis = "y", stackdir = "center")**
x, y, alpha, color, fill



g + **geom_violin(scale = "area")**
x, y, alpha, color, fill, linetype, size, weight

Discrete X, Discrete Y

h <- ggplot(diamonds, aes(cut, color))



h + **geom_jitter()**
x, y, alpha, color, fill, shape, size

Continuous Bivariate Distribution

i <- ggplot(movies, aes(year, rating))



i + **geom_bin2d(binwidth = c(5, 0.5))**
xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size, weight



i + **geom_density2d()**
x, y, alpha, colour, linetype, size



i + **geom_hex()**
x, y, alpha, colour, fill size

Continuous Function

j <- ggplot(economics, aes(date, unemploy))



j + **geom_area()**
x, y, alpha, color, fill, linetype, size



j + **geom_line()**
x, y, alpha, color, linetype, size



j + **geom_step(direction = "hv")**
x, y, alpha, color, linetype, size

Visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4.5, se = 1.2)
k <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))



k + **geom_crossbar(fatten = 2)**
x, y, ymax, ymin, alpha, color, fill, linetype, size



k + **geom_errorbar()**
x, ymax, ymin, alpha, color, linetype, size, width (also **geom_errorbarh()**)



k + **geom_linerange()**
x, ymin, ymax, alpha, color, linetype, size



k + **geom_pointrange()**
x, y, ymin, ymax, alpha, color, fill, linetype, shape, size

Maps

data <- data.frame(murder = USArrests\$Murder, state = tolower(rownames(USArrests)))
map <- map_data("state")
l <- ggplot(data, aes(fill = murder))



l + **geom_map(aes(map_id = state), map = map) + expand_limits(x = map\$long, y = map\$lat)**
map_id, alpha, color, fill, linetype, size

Three Variables

sealsSz <- with(seals, sqrt(delta_long^2 + delta_lat^2))
m <- ggplot(seals, aes(long, lat))



m + **geom_raster(aes(fill = h), hjust = 0.5, vjust = 0.5, interpolate = FALSE)**
x, y, alpha, fill

Install the package suite:

```
install.packages("tidyverse")
```

Load the package suite:

```
library(tidyverse)
```

Share your snippets during the coding session!

Go to <https://hackmd.io/CcRhOYJcScegKm8sF05Rqw> and post your code in between backticks:

For example:

```
```\n\nlibrary(lubridate)\n\nmy_data <- ... \n\n```
```

## DATA

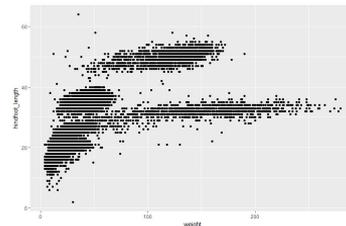
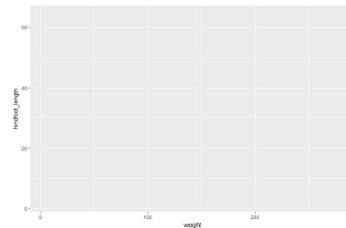
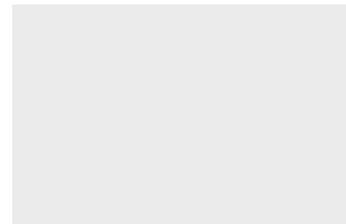
```
ggplot(data = surveys)
```

## MAPPING

```
ggplot(surveys, aes(x = weight,
 y = hindfoot_length))
```

## GEOM

```
ggplot(surveys, aes(x = weight,
 y = hindfoot_length)) +
 geom_point()
```



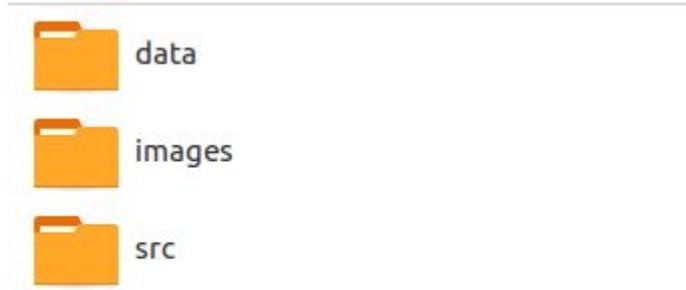
# The concept

We defined a number of challenges. If you were able to achieve a challenge, add a  to your laptop screen.

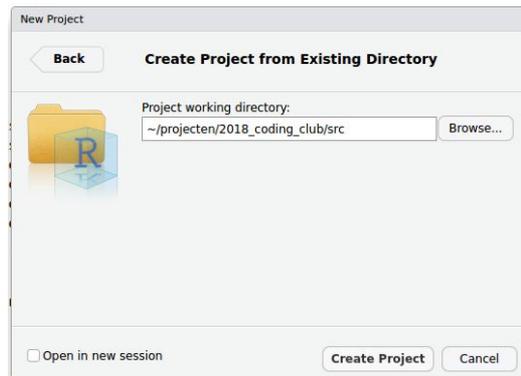
The objective is that **everyone** achieves !

- Someone has more  than you? **Ask for help!**
- Someone has less  than you? **Provide help!**

- Download coding club material and work locally, not in sync with the Google drive

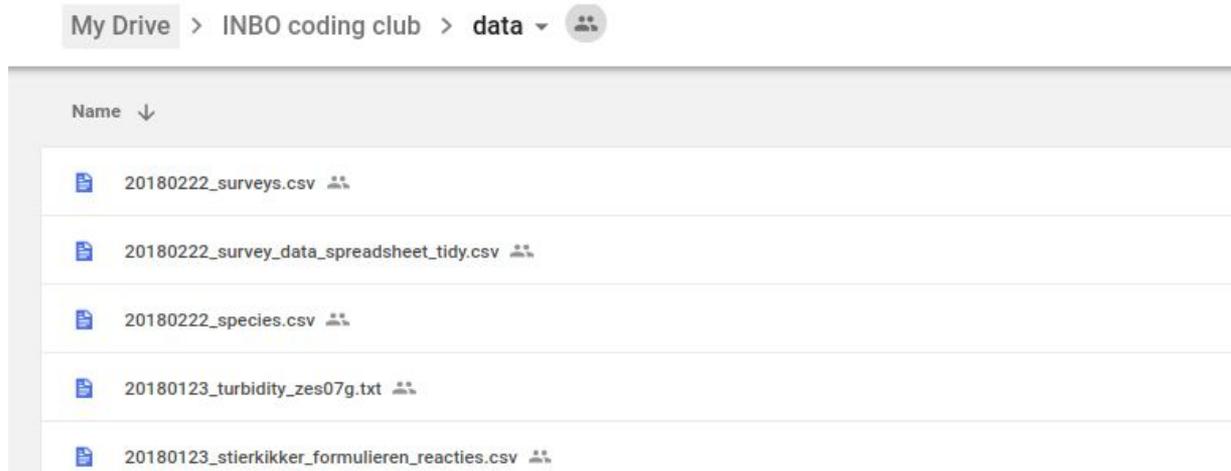


- Create new Rstudio project in the **/src** folder



- Download coding club material and work locally, not in sync with the Google drive
- Create new Rstudio project in the **src** folder...
- Use relative paths to data files!

```
> library(tidyverse)
> surveys <- read_csv("../data/20180222_surveys.csv") %>%
 filter(!is.na(weight), # remove missing weight
 !is.na(hindfoot_length), # remove missing hindfoot_length
 !is.na(sex)) # remove missing sex
```



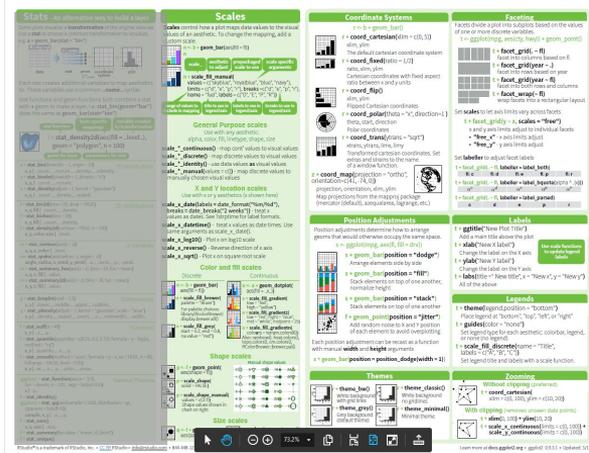


Starting from:

```

gplot(surveys, aes(x = weight,
 y = hindfoot_length)) +
 geom_point()

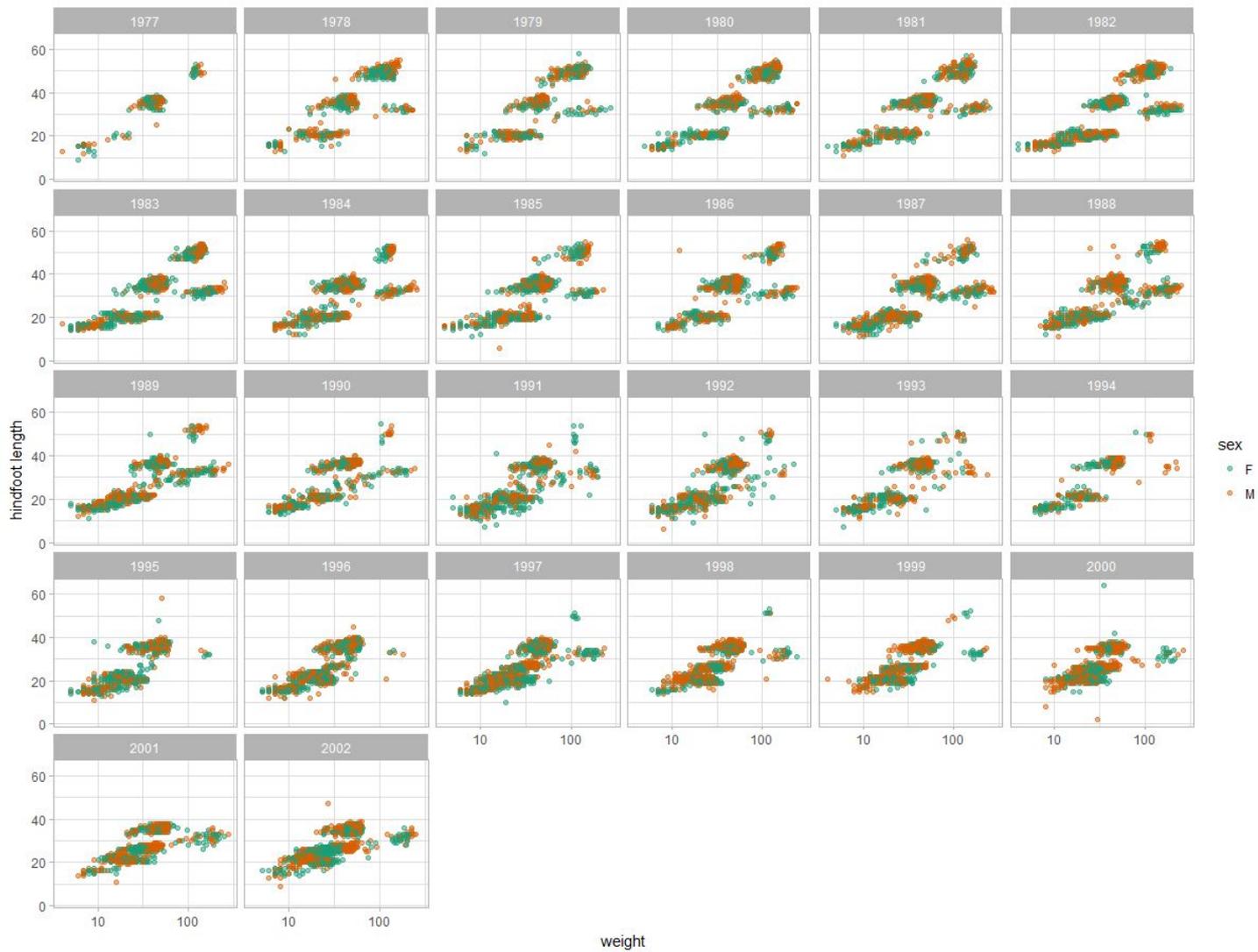
```

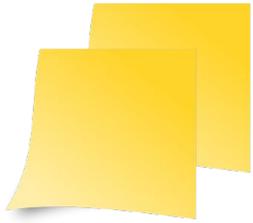


gAdd following adjustments:

- Map the color aesthetic to the `sex` variable
- Adjust the transparency (`alpha`) of the points to 0.5
- Change the y label (`ylabel`) to "hindfoot length"
- Use a logarithmic `scale_x` for the x-axis
- Change the color `scale` of the points to another scale (pick one yourself)
- Change the general `theme` (pick one yourself)
- Split the graph by each year (`facet_wrap`)

Other interesting elements to the plot? Add them to the [hackmd](#) code snippet!





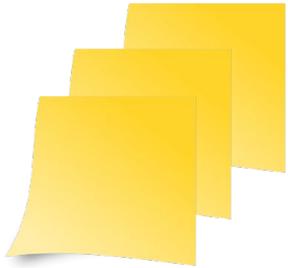
First,

```
> tidy_bevolking <- read_csv("../data/20180522_gent_groeiperwijk_tidy.csv")
```

1. Make a heatmap of the districts of Ghent evolution, with the years in the x-axis and the districts in the y-axis.
2. Use a *diverging* color map and remove the x and y labels.



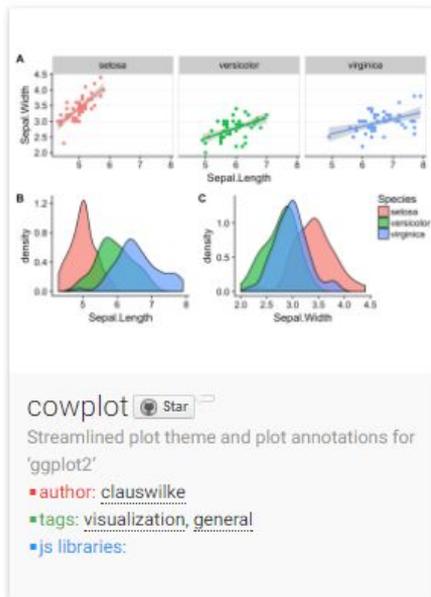


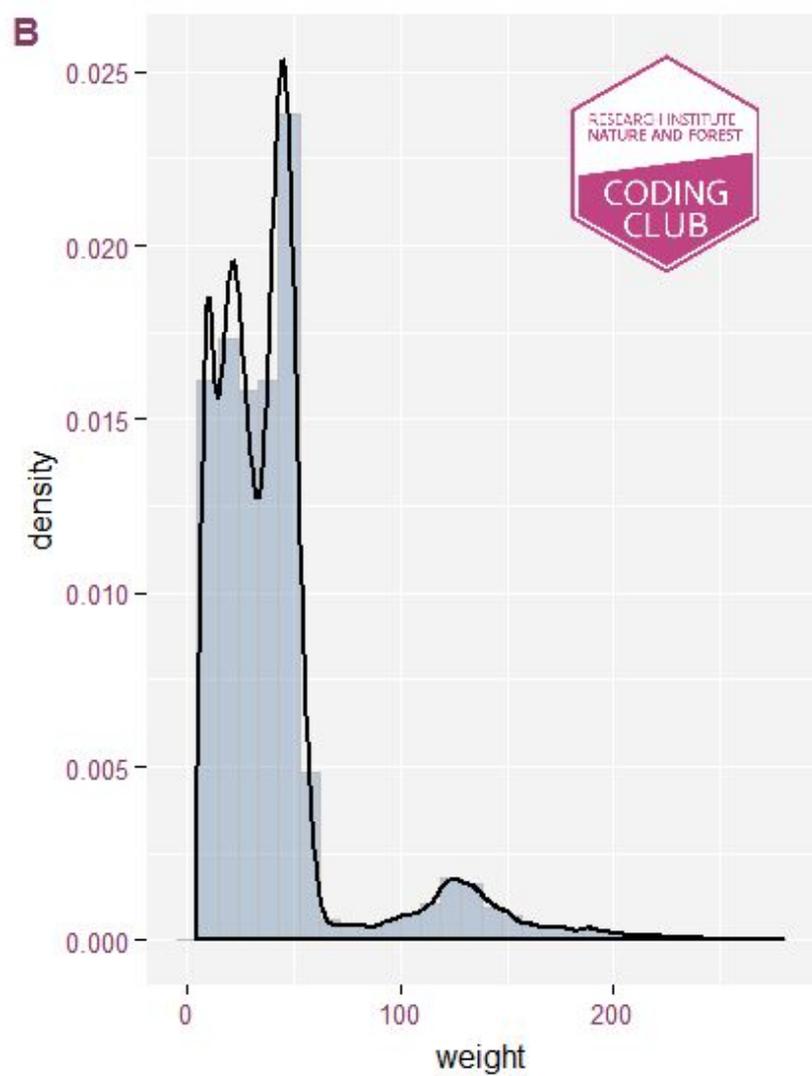
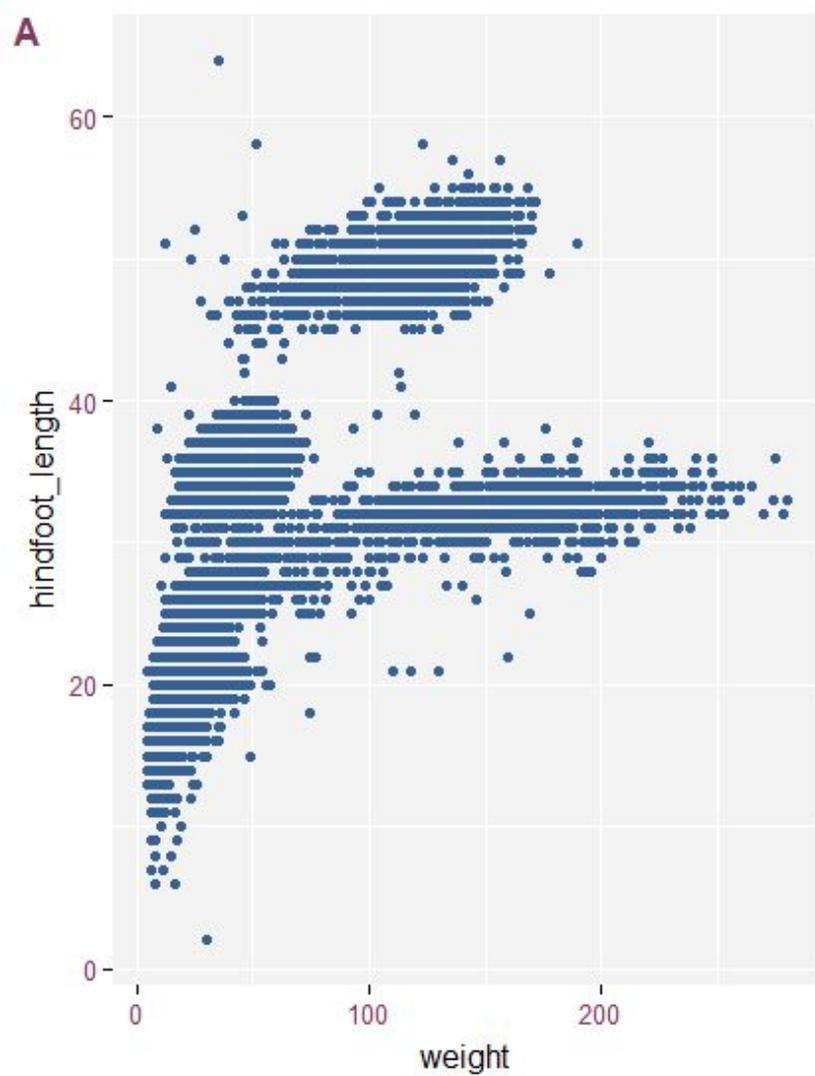


First,

```
> install.packages("cowplot")
> install.packages("INBOtheme") # install inbotheme
> library(cowplot)
> library(INBOtheme)
> weight_scatter <- ggplot(surveys,
 aes(x = weight, y = hindfoot_length)) +
 geom_point()
```

1. Make a second plot, called `weight_density`, showing a histogram of the weights in combination with a density estimate (style as you like). Use the INBO theme.
2. Check the [cowplot documentation](#) and place both graphs next to each other, labeling as subplot A and B.
3. (BONUS) add the [coding club logo](#) to the combine graph





ggplot2 extensions

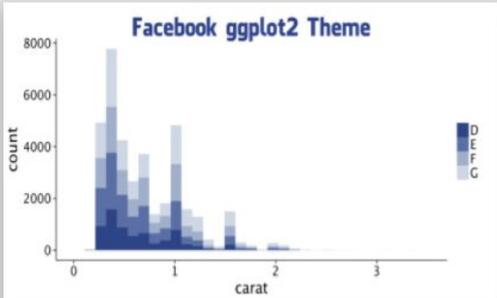
Secure | https://www.ggplot2-exts.org

Home Gallery Extensions GitHub

## A List of ggplot2 extensions

This site tracks and lists ggplot2 extensions developed by R users in the community.

The aim is to make it easy for R users to find developed extensions.



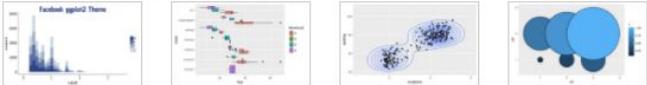
Facebook ggplot2 Theme

count

carat

New Geoms New Stats

See the Extensions



Go to the [Gallery](#) page to see a list of ggplot2 extensions.

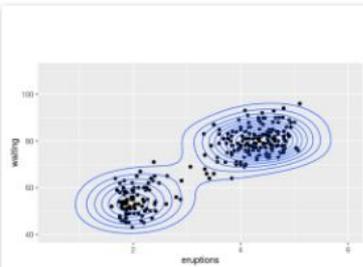
[See the Extensions »](#)

## Submit your extension

Submit your ggplot2 extensions so that other R users can easily find them. To do so, simply submit a pull request using these [simple instructions](#).

[Submit your Extension »](#)

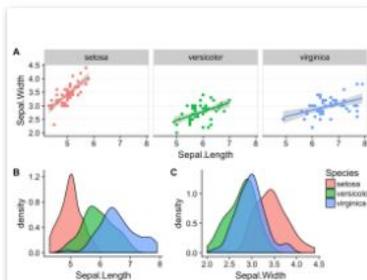
Maintained by Daniel Emaasit



### ggalt

A compendium of 'geoms', 'coords' and 'stats' for 'ggplot2'.

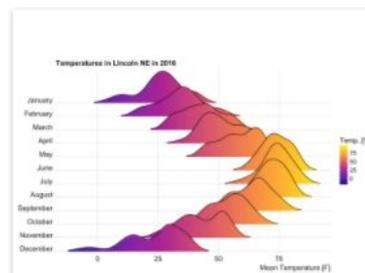
- **author:** [hrbrmstr](#)
- **tags:** [visualization](#), [general](#)
- **js libraries:**



### cowplot

Streamlined plot theme and plot annotations for 'ggplot2'

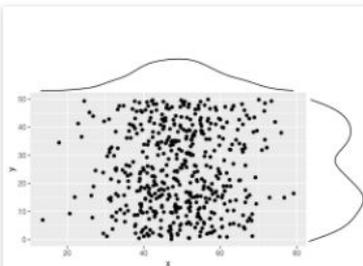
- **author:** [clauswilke](#)
- **tags:** [visualization](#), [general](#)
- **js libraries:**



### ggridges

Ridgeline plot geoms for 'ggplot2'

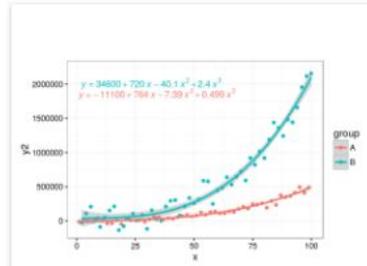
- **author:** [clauswilke](#)
- **tags:** [visualization](#), [general](#)
- **js libraries:**



### ggExtra

ggExtra lets you add marginal density plots or histograms to ggplot2 scatterplots.

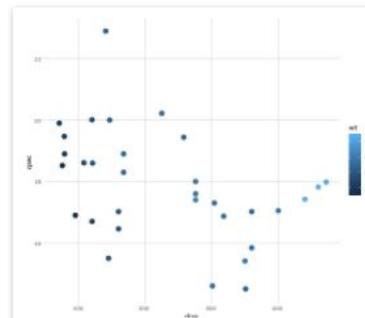
- **author:** [daattali](#)
- **tags:** [histogram](#), [marginal](#), [density](#)
- **js libraries:**



### ggpmisc

Miscellaneous Extensions to 'ggplot2'.

- **author:**
- **tags:** [visualization](#), [general](#)
- **js libraries:**



### ggiraph

htmlwidget to make 'ggplot' graphics interactive.

- **author:** [davidgohel](#)
- **tags:** [visualization](#), [general](#)
- **js libraries:**





Zaal: Herman Teirlinck - 01.05 - Isala Van Diest

Datum: 2018-06-14, van 10:00 tot 12:00

*(registratie aangekondigd via [DG\\_useR@inbo.be](mailto:DG_useR@inbo.be))*