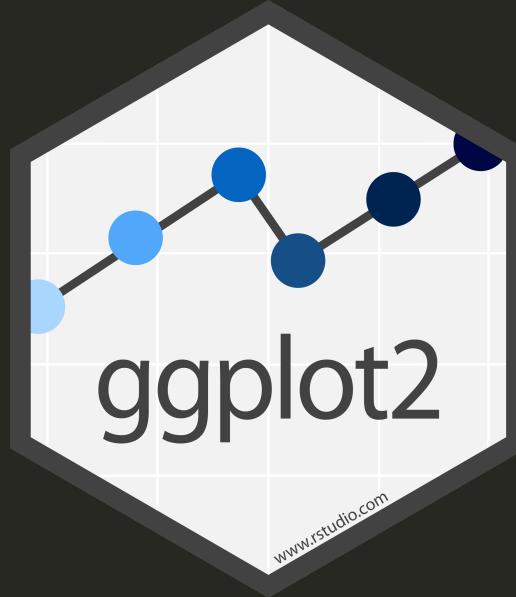


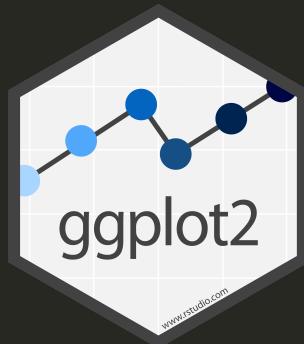
# ggploting your way to beautiful figures



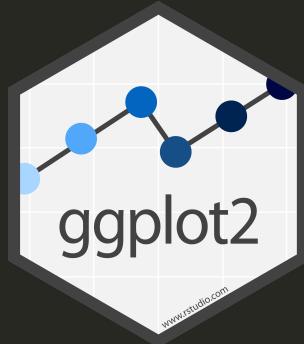
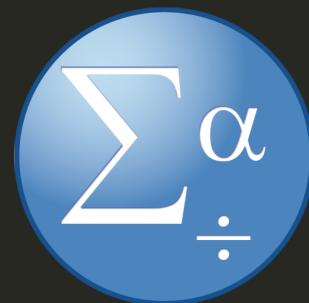
Daniel Roelfs (@dthroelfs)  
Adriano Winterton (@fuyu00)  
Linn Sofie Saether (@linnssaether)

# How to improve our figures

# When it comes to figure making



>



>

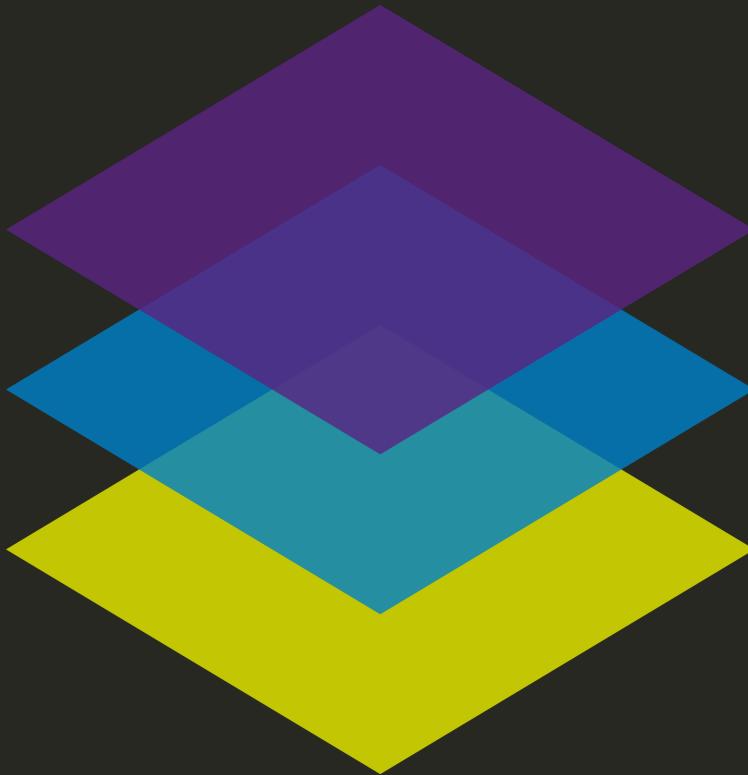


# The anatomy of a ggplot

Geometries

Aesthetics

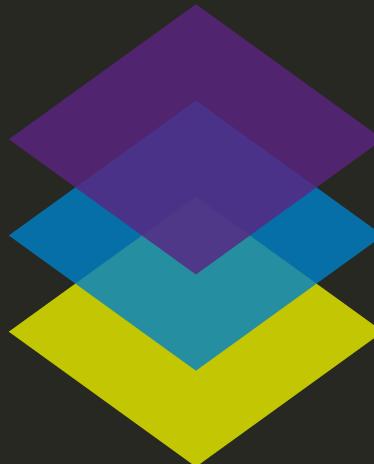
Data



Geometries

Aesthetics

Data



**Geometries** = boxplot, line, scatter, etc.

**Aesthetics** = x-axis, y-axis, color, fill, shape

**Data** = your awesome TOP data 😊

# Installing the packages we need

```
install.packages("tidyverse") # <- includes ggplot2  
devtools::install_github("norment/normentR") # <- themes and colors
```

# Loading the packages

```
library(tidyverse)
library(normentR)
```

# The skeleton



# The code

```
ggplot(data = mtcars, aes(x = wt, y = mpg)) +  
  geom_point()
```

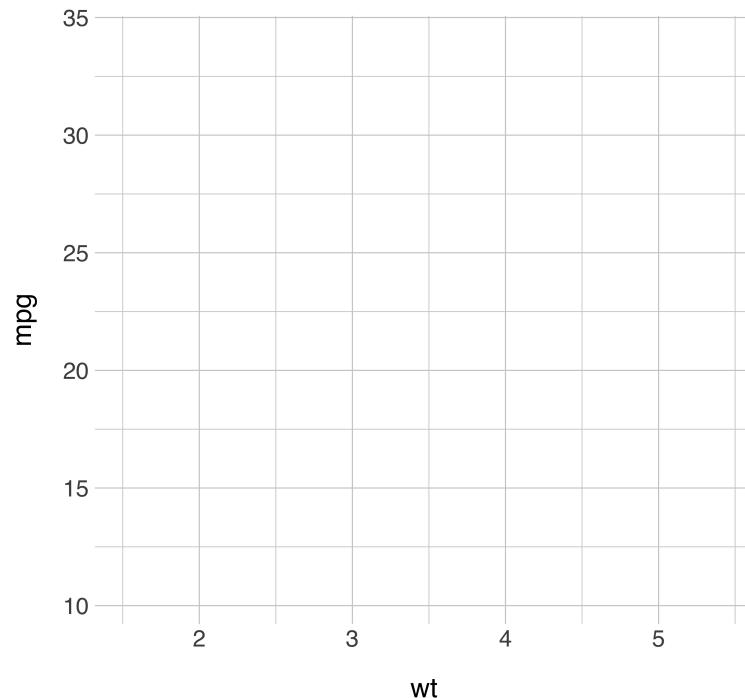
Data: mtcars

Aesthetics: aes(x = wt, y = mpg)

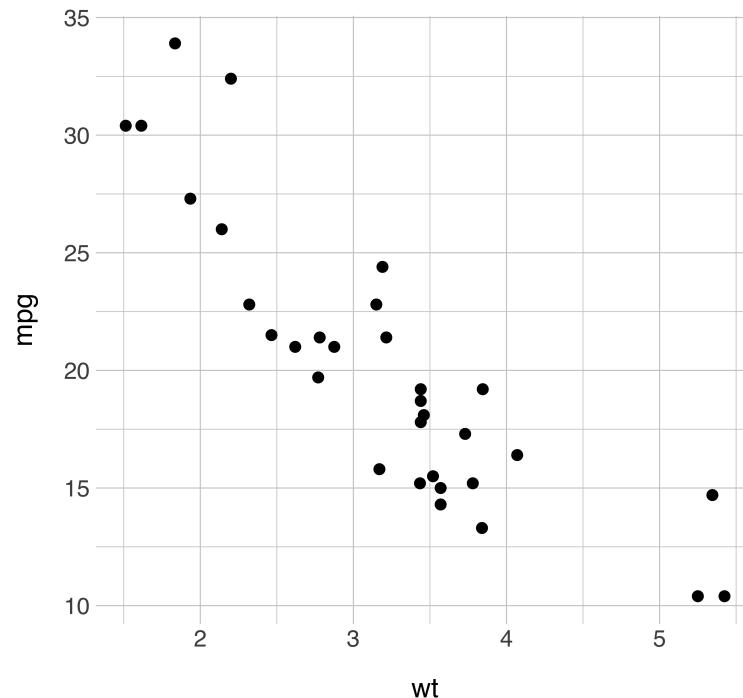
Geometry: geom\_point()

ggplot, the layer cake 🍰

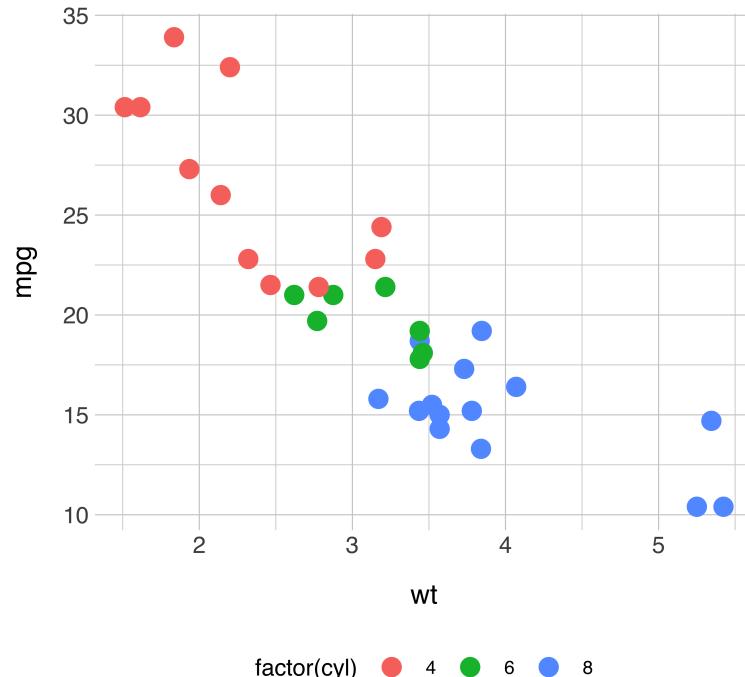
```
ggplot(mtcars, aes(x = wt, y = mpg))
```



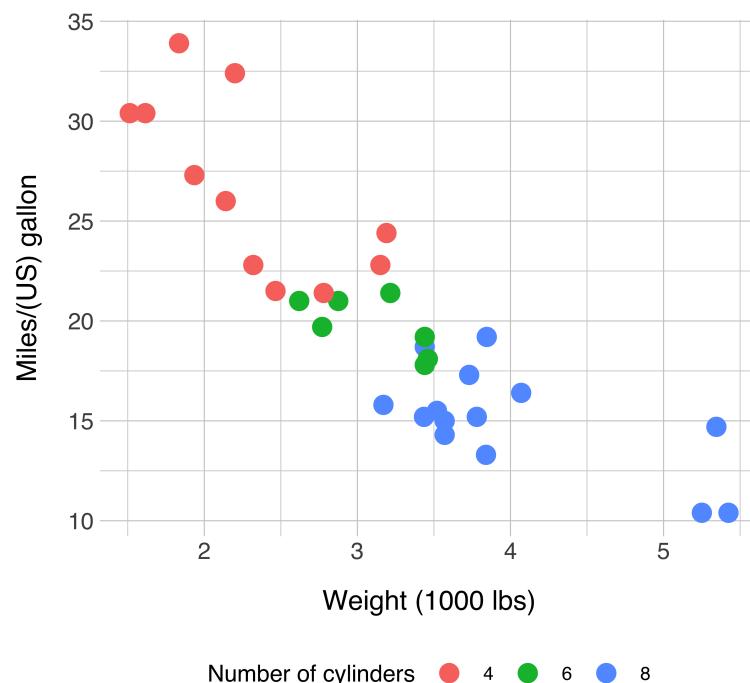
```
ggplot(mtcars, aes(x = wt, y = mpg)) +  
  geom_point()
```



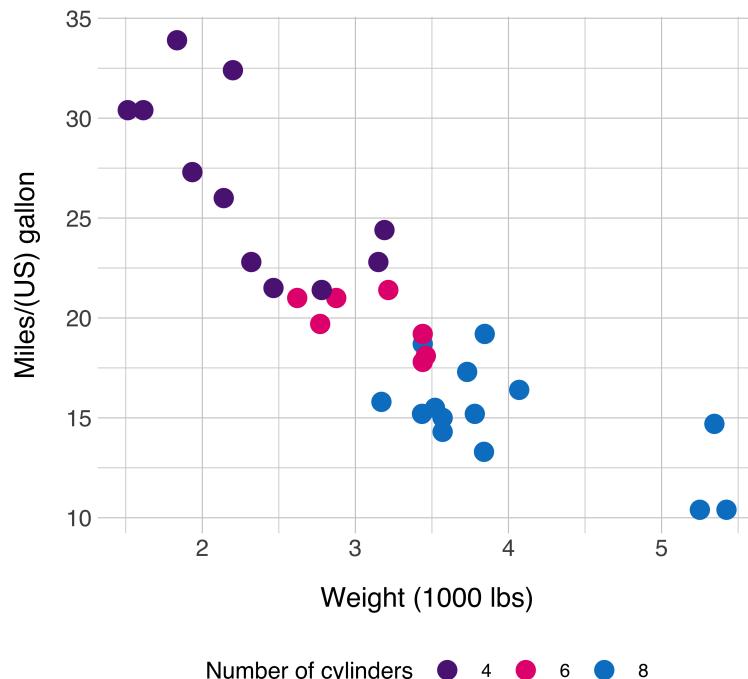
```
ggplot(mtcars, aes(x = wt, y = mpg, color = factor(cyl))) +  
  geom_point(size = 3)
```



```
ggplot(mtcars, aes(x = wt, y = mpg, color = factor(cyl))) +  
  geom_point(size = 3) +  
  labs(x = "Weight (1000 lbs)",  
       y = "Miles/(US) gallon",  
       color = "Number of cylinders")
```



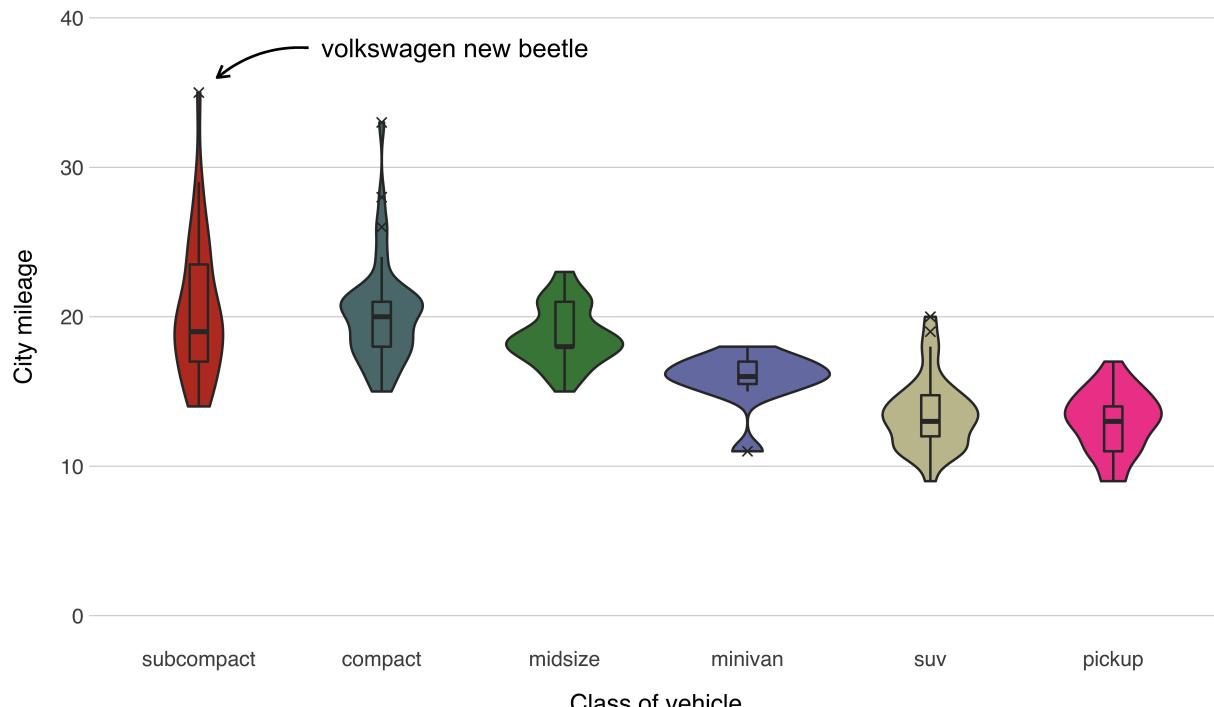
```
ggplot(mtcars, aes(x = wt, y = mpg, color = factor(cyl))) +  
  geom_point(size = 3) +  
  labs(x = "Weight (1000 lbs)",  
       y = "Miles/(US) gallon",  
       color = "Number of cylinders") +  
  scale_color_norment(discrete = TRUE, palette = "logo")
```



```
p <- ggplot(mpg %>% filter(class != "2seater"),
             aes(x = reorder(class,-cty), y = cty,
                 fill = reorder(class,-cty))) +
  geom_violin() +
  geom_boxplot(width = 0.1, outlier.shape = 4) +
  geom_curve(aes(xend = 1.1, yend = 36, x = 1.6, y = 38),
             curvature = 0.2,
             arrow = arrow(length = unit(2,"mm"))) +
  geom_text(data = . %>% filter(cty == max(cty)),
            aes(x = 2.4, y = 38,
                label = paste(manufacturer, model, sep = " "))) +
  labs(x = "Class of vehicle",
       y = "City mileage",
       fill = "Class",
       title = "What class of car has the\nbest mileage in the city?",
       caption = "Source: mpg") +
  scale_x_discrete() +
  scale_y_continuous(limits = c(0,40)) +
  scale_fill_norment(discrete = TRUE, palette = "nejm") +
  theme_norment(legend = FALSE, grid = "Y") +
  theme(
    plot.title = element_text(hjust = -0.1)
  )
```

```
print(p)
```

## What class of car has the best mileage in the city?



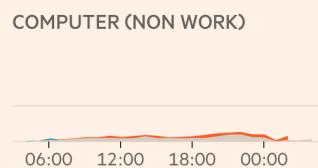
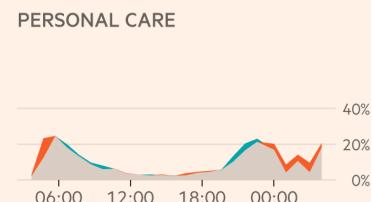
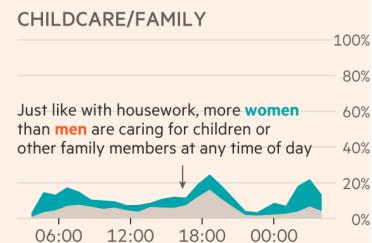
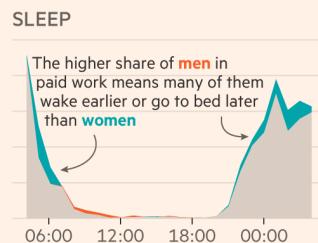
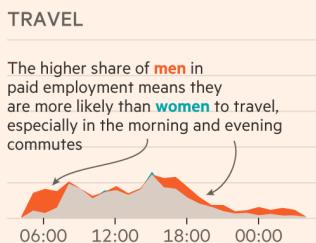
Source: mpg

# What's possible with ggplot

## How women and men spend their time differently in the UK

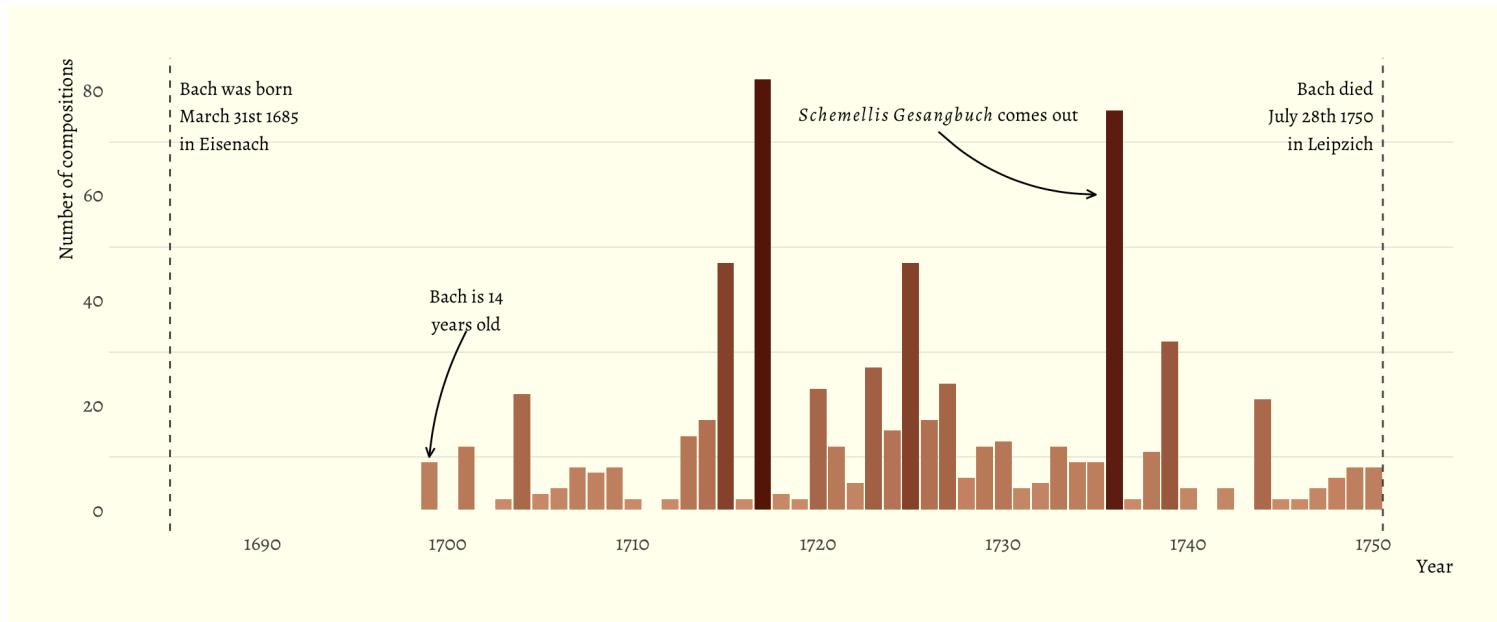
Share of women and men doing specific activities by time of day (%)

Data shown are for adults in households with dependent children

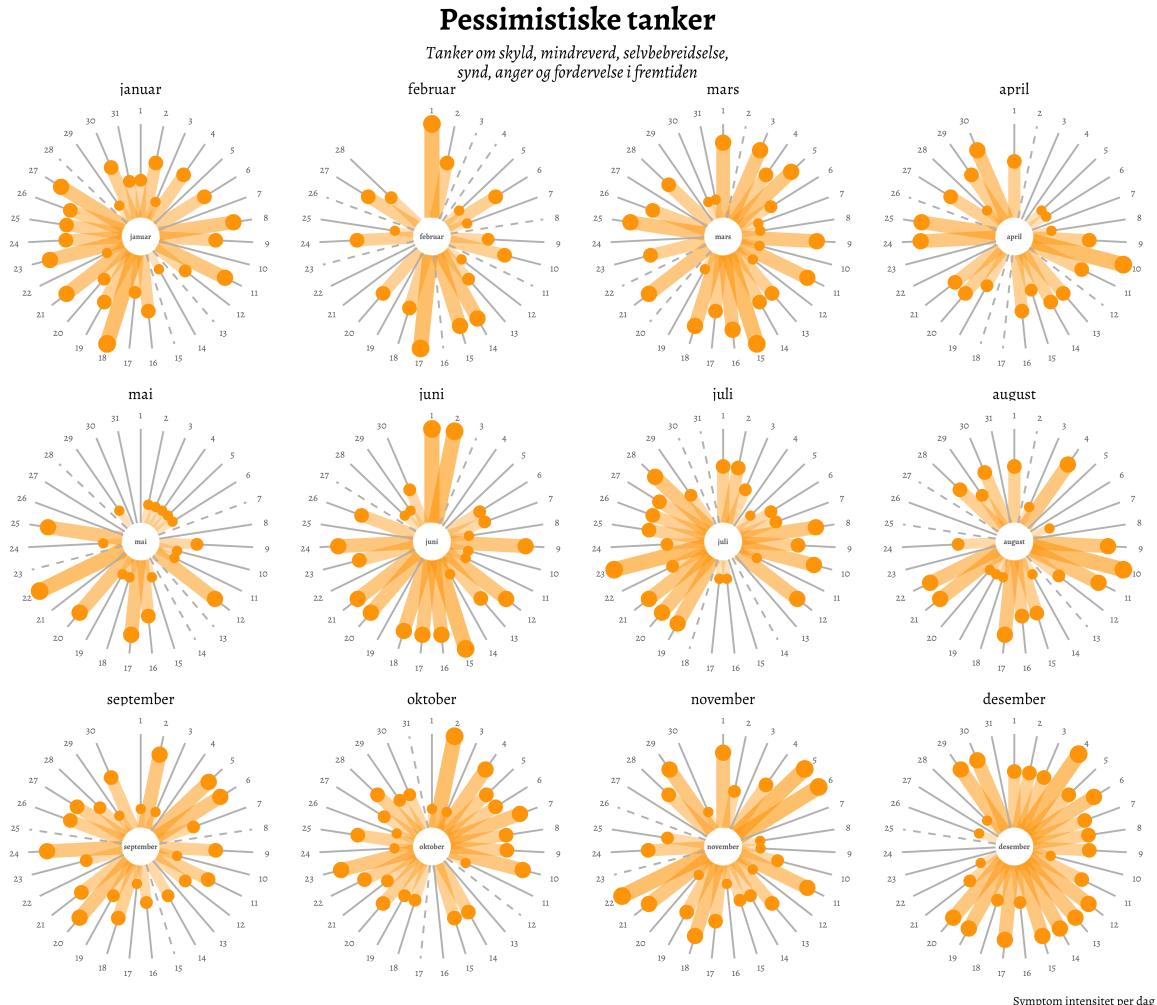


Source: FT analysis of UK Time Use Survey 2015  
© FT

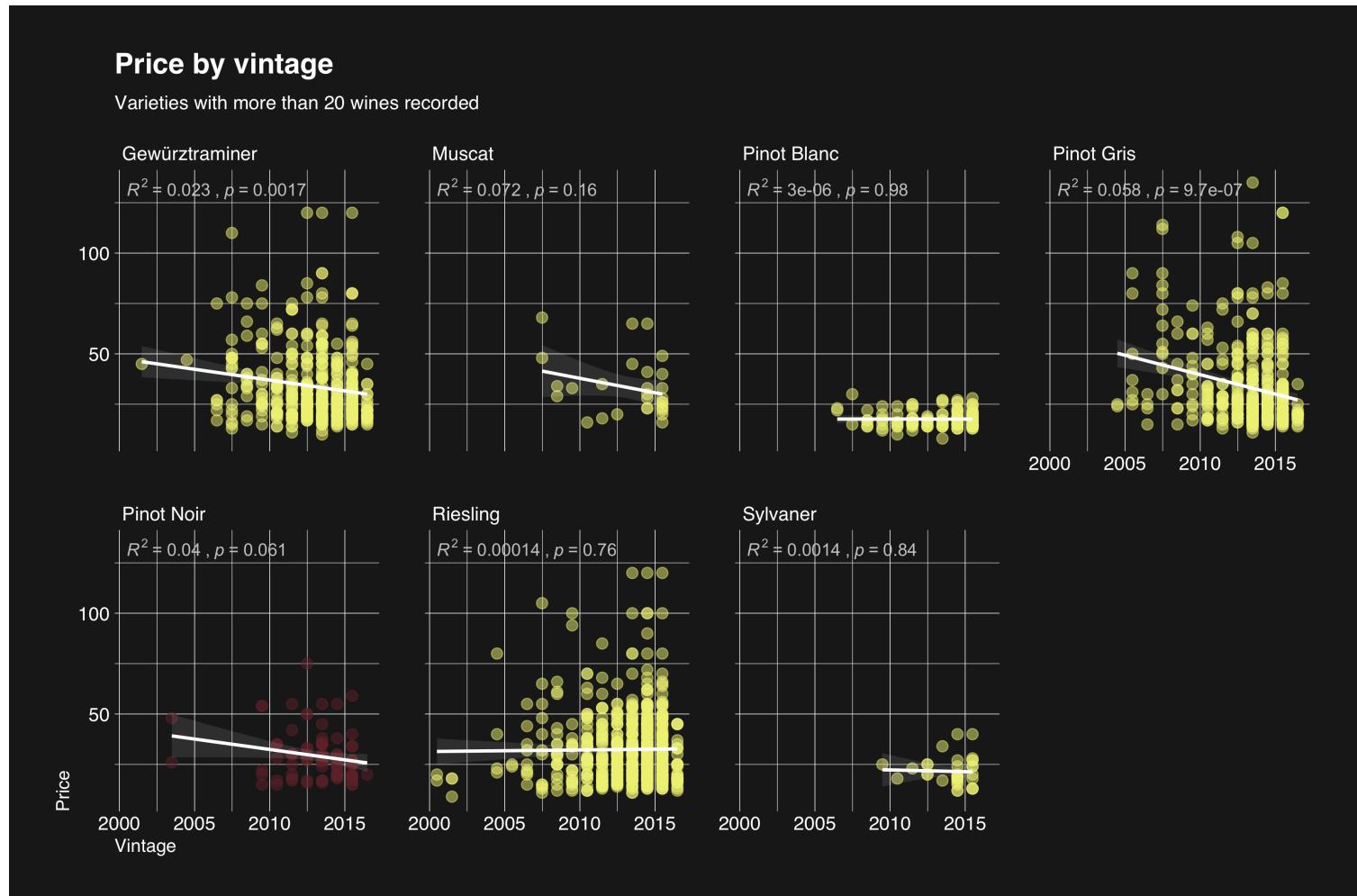
# What's possible with ggplot



# What's possible with ggplot



# What's possible with ggplot



One more (very) important concept:

Long vs wide format 😊

# Long vs wide format

**ggplot wants a single column for x-axis, y-axis, color, etc**

This is often not a problem, e.g.:

```
str(your_data)
```

```
## 'data.frame':    30 obs. of  3 variables:  
##   $ id      : Factor w/ 30 levels "HC_0102","HC_0110",...: 17 3 29 12 7 25 1  
##   $ age     : int  50 49 33 21 36 47 60 58 64 44 ...  
##   $ measure: num  0.5 2.12 1.26 0.82 1.1 0.72 2.16 1.66 2.64 2.74 ...
```

```
print(your_data)
```

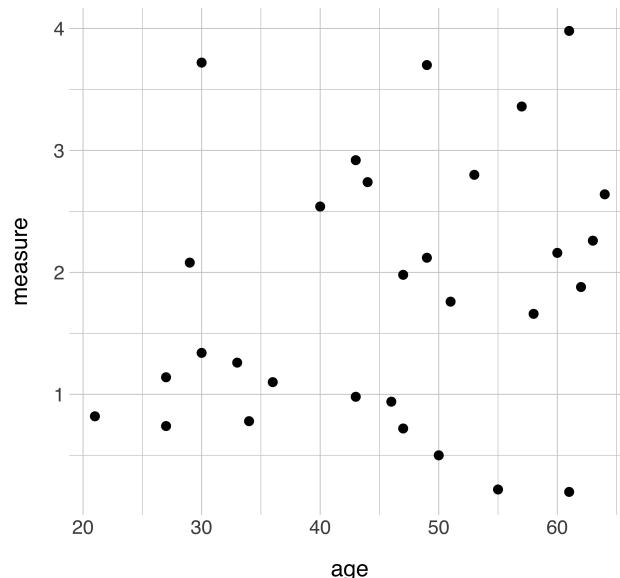
```
##          id age measure  
## 1  HC_0492  50    0.50  
## 2  HC_0119  49    2.12  
## 3  HC_0862  33    1.26  
## 4  HC_0375  21    0.82  
## 5  HC_0263  36    1.10  
## 6  HC_0707  47    0.72  
## 7  HC_0102  60    2.16  
## 8  HC_0471  58    1.66  
## 9  HC_0734  64    2.64
```

# Long vs wide format

**ggplot** wants a single column for x-axis, y-axis, color, etc

This is often not a problem, e.g.:

```
ggplot(your_data, aes(x = age, y = measure)) +  
  geom_point()
```



# Long vs wide format

## Example of wide format

View(wide)

▲	id	sex	diag	iq	workmem	procspeed	cpt
1	1	M	HC	101	251	262	289
2	2	M	BD	99	255	230	286
3	3	F	SCZ	114	210	284	288
4	4	F	SCZ	105	231	205	232
5	5	M	HC	92	286	212	253

# Long vs wide format

## Conversion to long format

```
long <- pivot_longer(data = wide,
                      cols = c(iq,workmem,procspeed,cpt))
#View(long)
```

# Long vs wide format

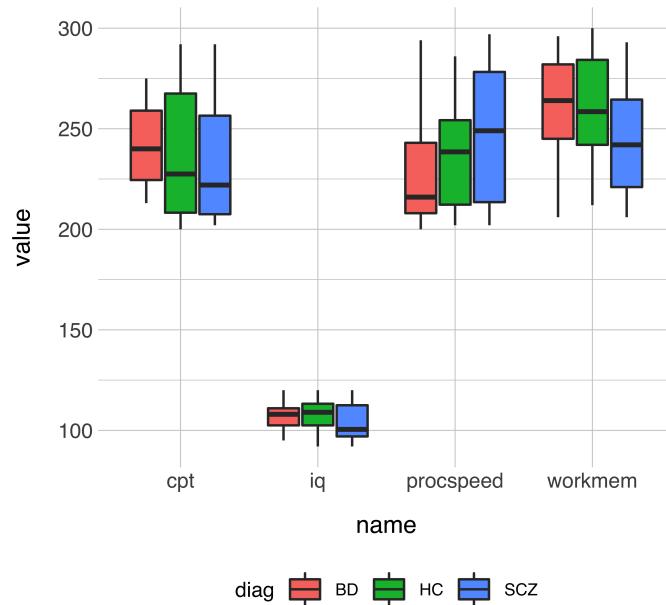
## Conversion to long format

```
long <- pivot_longer(data = wide,  
                      cols = c(iq,workmem,procspeed,cpt))  
View(long)
```

# Long vs wide format

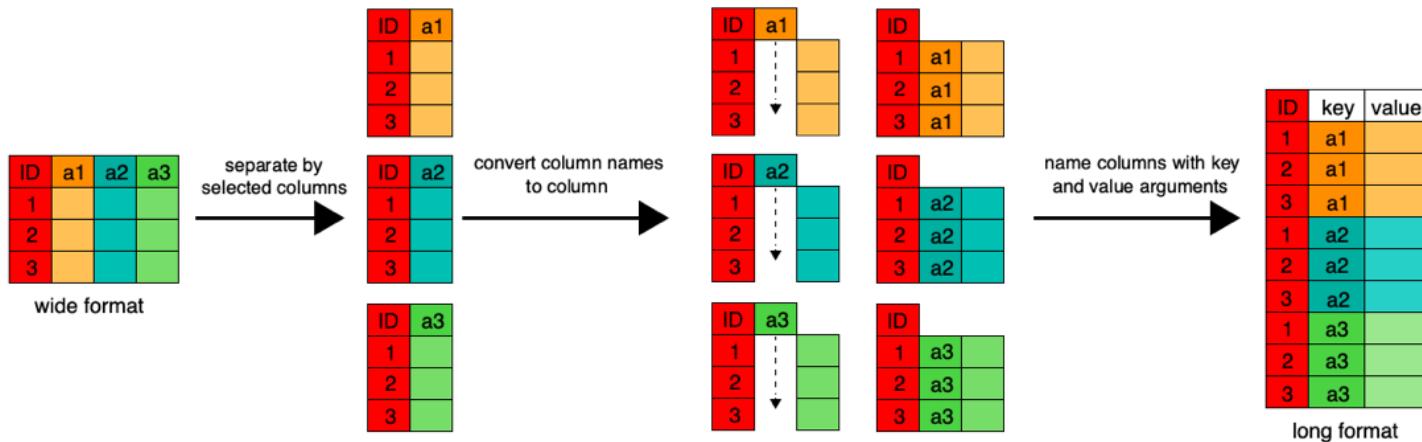
## Conversion to long format

```
ggplot(long, aes(x = name, y = value, fill = diag)) +  
  geom_boxplot()
```



# Long vs wide format

## Conversion to long format



# Introduction to the workshop

Download the data here:

[bit.ly/2vXsJPJ](https://bit.ly/2vXsJPJ)