

Plotting with ggplot2

ESTP Use of R in Official Statistics

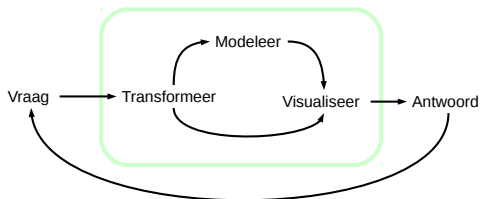


When communicating results to non-technical types there is nothing better than a clear visualization to make your point.

Numerical quantities focus on expected values, graphical summaries on unexpected values.

John Tukey

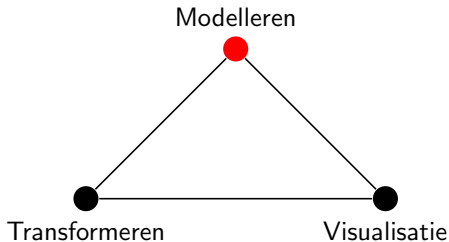
Role visualization in data-analysis





R has great plotting facilities:

- very fine-grained control over plot
- Most functionality is/was focused on static images.



- `base/graphics` offers extended possibilities for making and tweaking graphs
- `ggplot2` creates by default good graphs
- Lots of extra packages for making specialized graphs.
- E.g.: `highchartR`, `plotly`, `ggvis`, `dyGraph`.

Exporting figures/graph



By hand (not recommended):

- In Rstudio by default the plotting panel is used.
- You can export plot from RStudio using the menu: “Export”.

Setting programmatically a device:

1) Set as plotting device:

- pdf Creates a pdf device (document)
- png Creates a png device (document)
- ...

2) Use your plotting commands: plots will be writing to disk.

3) Close the device: `dev.off()`

ggplot also has `ggsave`.

Grammar of graphics





- Grammar of Graphics plotting library
- Hadley Wickham

```
install.packages("ggplot2")
```




```
head(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

dataset mtcars



```
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
## $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 1
## $ cyl : num   6 6 4 6 8 6 8 4 4 6 ...
## $ disp: num  160 160 108 258 360 ...
## $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
## $ drat: num   3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92
## $ wt  : num   2.62 2.88 2.32 3.21 3.44 ...
## $ qsec: num   16.5 17 18.6 19.4 17 ...
## $ vs  : num   0 0 1 1 0 1 0 1 1 1 ...
## $ am  : num   1 1 1 0 0 0 0 0 0 0 ...
## $ gear: num   4 4 4 3 3 3 3 4 4 4 ...
## $ carb: num   4 4 1 1 2 1 4 2 2 4 ...
```

Definition of statistical graphic



*A statistical graph is a **mapping** of **variables** on **aesthetics** of **geometric** objects in which variable values are **scaled** to **aesthetic values**.*

- geom__
- scale__

geometric objects?



- line, bar, area, polygon, etc.

In `ggplot` these are combined into a *geom_*etric object

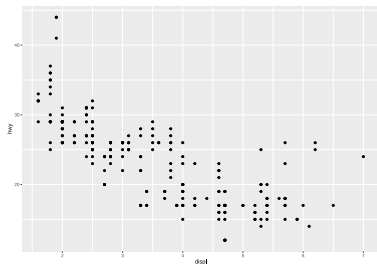


- Gebaseerd op *Grammar of Graphics*;
- Graph = Data + Mapping + Geometry + Coördinate system;
- Graph as one or more **geometries**.
- Geometry has **aesthetic** features (size, position, color etc.);
- Columns are mapped to aesthetics.

Example scatterplot



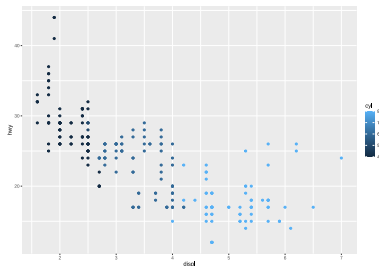
```
library(ggplot2)
ggplot( data = mpg
        , aes(x = displ, y = hwy)) +
  geom_point()
```



scatter plot (color)



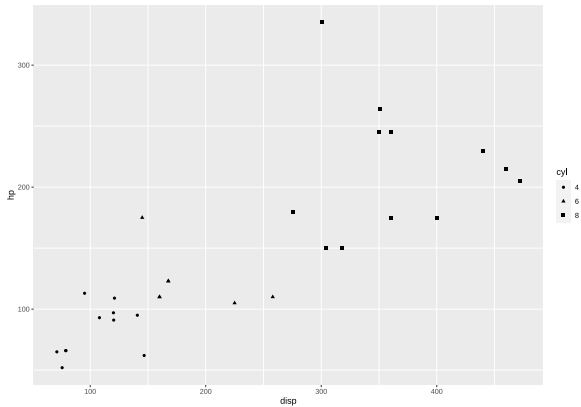
```
mtcars$cyl <- factor(mtcars$cyl)
library(ggplot2)
ggplot( data = mpg
        , aes( x = displ
                , y = hwy
                , color = cyl
                )) +
  geom_point()
```



scatter plot (shape)



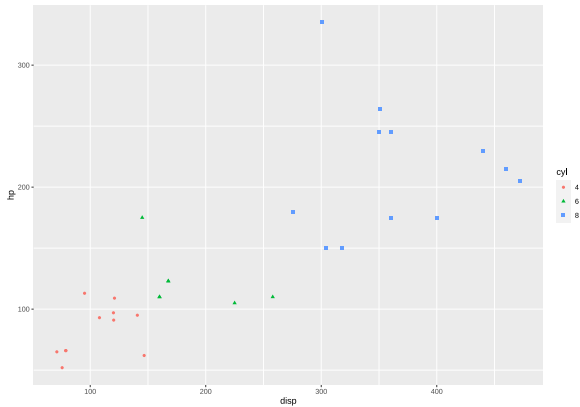
```
mtcars$cyl <- factor(mtcars$cyl)
ggplot(data = mtcars, aes(x=disp, y=hp, shape=cyl)) + geom_point()
```



scatter plot (shape)



```
mtcars$cyl <- factor(mtcars$cyl)
ggplot(data = mtcars, aes(x=disp, y=hp, shape=cyl, color = cyl)) + geom_point()
```





- Create a histogram from `carat` column in the `diamonds` dataset (with `geom_histogram`)
- Play around with the `binwidth`: what is the story of the data?



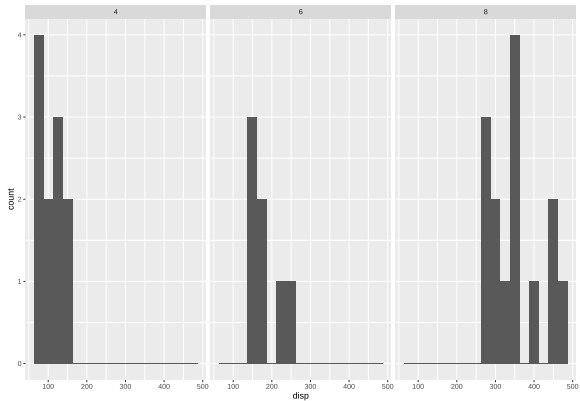
Split any ggplot plot on a categorical variable into small multiples

- very useful during analyses!
- by default same scales, to make comparison easy

facets



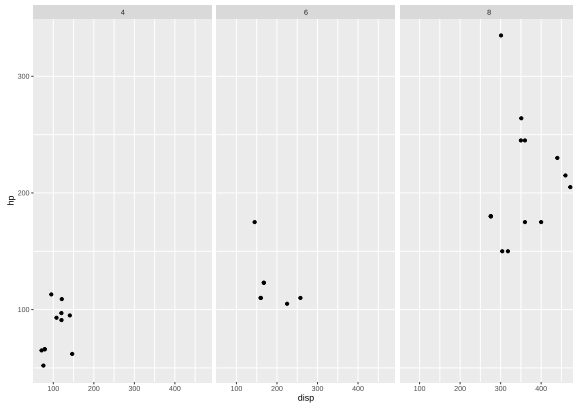
```
ggplot(mtcars) + geom_histogram(aes(x = disp), binwidth = 25) + facet_wrap(~cyl)
```



facets



```
mtcars$cyl <- factor(mtcars$cyl)
ggplot(data = mtcars, aes(x=disp, y=hp)) + geom_point() + facet_wrap(~cyl)
```



- Specify data with `ggplot`;
- add extra layers or properties using a `+`
- Specify a geometry with `geom_`.
- Map a column to an aesthetic with `aes`;
- Specify a geometry with `geom_`.
- Each aesthetic has a `scale_`
- See cheat sheet.



- each plot can be stored in a variable and be adjust

```
p <- ggplot(mtcars) + geom_point(aes(x = disp, y = hp))  
p + facet_wrap(~cyl)  
p + facet_wrap(~am)
```

- a plot can be save easily with ggsave

```
ggsave("test.pdf") # remembers last plot  
# or when assigned to variable  
ggsave("test.png", plot = p)
```

ggplot2 useful packages



- `ggforce`, annotation
- `gganimate` create an animation from a `ggplot`
- `ggrepel`, nice label positioning of annotated labels.