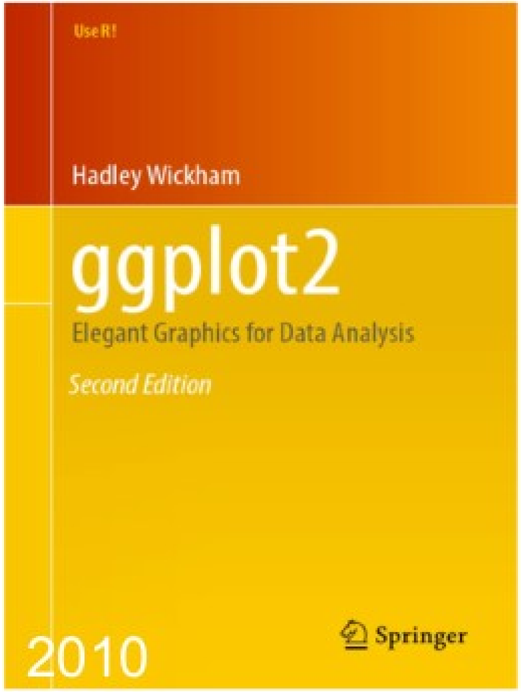
Visualization & ggplot2 Lab



Use R!

Hadley Wickham

ggplot2

Elegant Graphics for Data Analysis

Second Edition

Springer

2010

Goals for the session

1. Discuss the principles of good vs bad data viz
2. Understand the grammar of graphics
4. Introduce, explain and use ggplot()
5. Discuss how to select the most appropriate plots
6. Visualization and discovery of global trends

base R plotting

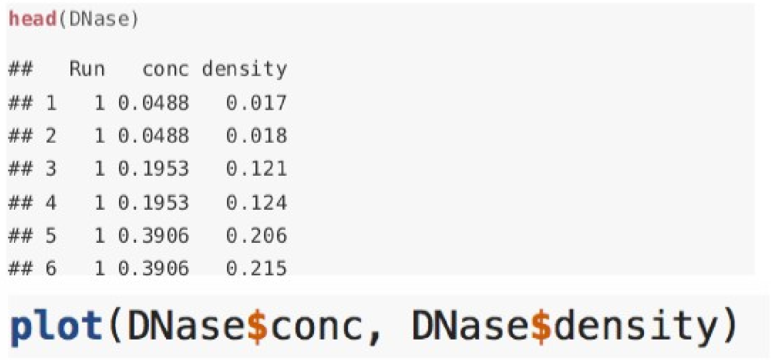
Drawbacks:

- **Layout choices have to be made at the beginning** with no overview over what may still be coming
- **Different functions for different plot types**, with different interfaces
- Routine tasks can require lots of **boilerplate code**
- **No concept of facets / lattices**
- Only a **single global coordinate** system allowed per plot
- **Poor default colours**
- **Resizing** often leads to unsatisfactory results

base R plotting

canvas model:
a series of instructions that **sequentially** fill the plotting canvas

Great for quick data exploration!



```
head(DNase)

##   Run   conc density
## 1   1 0.0488   0.017
## 2   1 0.0488   0.018
## 3   1 0.1953   0.121
## 4   1 0.1953   0.124
## 5   1 0.3906   0.206
## 6   1 0.3906   0.215
```

```
plot(DNase$conc, DNase$density)
```

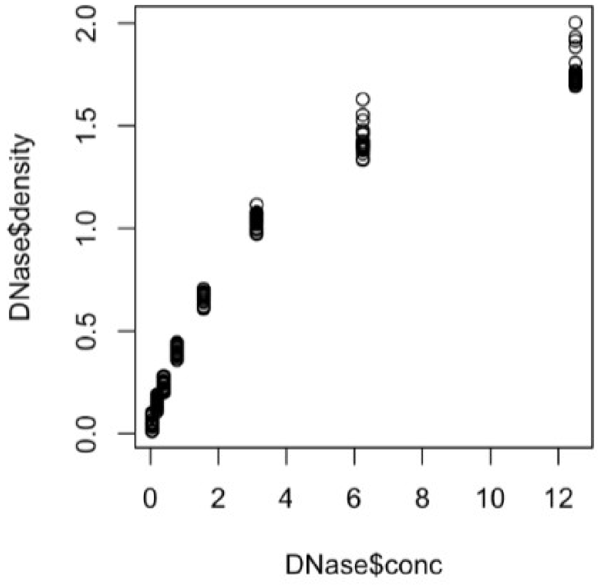
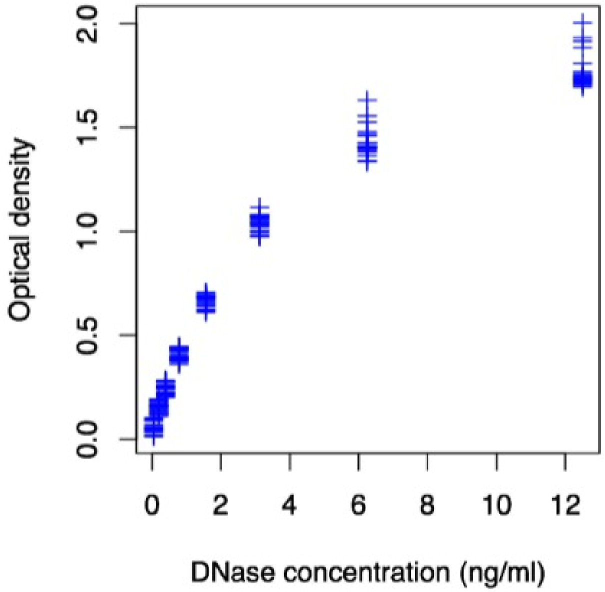


Figure 3.2: Plot of concentration vs. density for an ELISA assay of DNase.

base R plotting

canvas model:
a series of instructions that **sequentially** fill the plotting canvas

Inefficient for customization and generating complex plots.

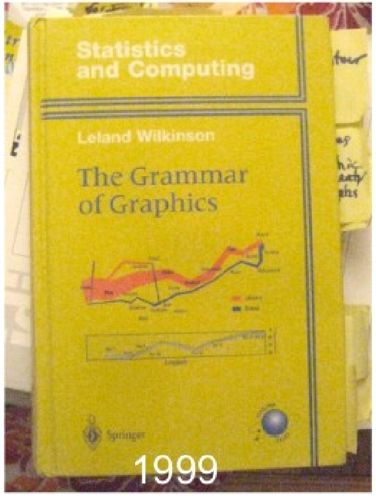


```r
plot(DNase$conc, DNase$density,
ylab = attr(DNase, "labels")$y,
xlab = paste(attr(DNase, "labels")$x, attr(DNase, "units")$x),
pch = 3, col = "blue")
```

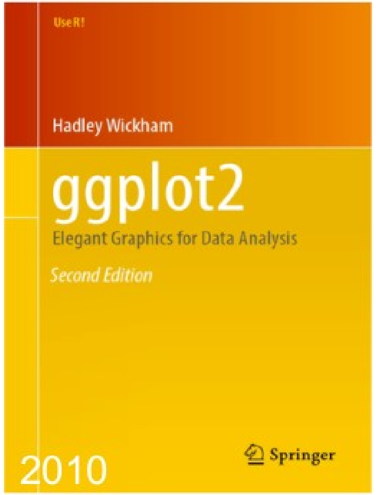
Goals for the session

1. Discuss the principles of good vs bad data viz
2. **Understand the grammar of graphics**
3. **Introduce, explain and use ggplot()**
4. Discuss how to select the most appropriate plots
5. Visualization and discovery of global trends

The Grammar of Graphics



1999



2010

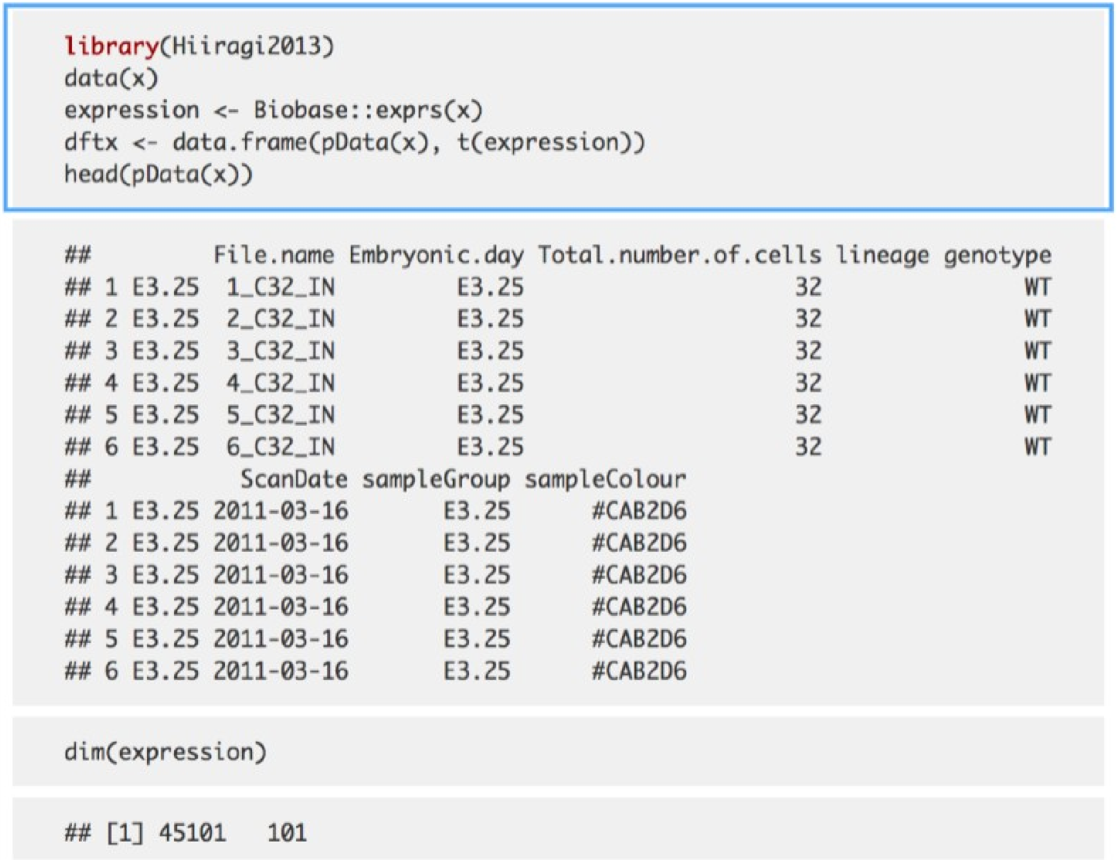
Concept **coined by Leland Wilkinson in 1999**.
An **abstraction** which facilitates reasoning and communicating graphics.

ggplot2 is an implementation of **a layered grammar of graphics** that enables users to independently specify the building blocks of a plot and combine them to create just about any kind of graphical display.

ggplot grammar of graphics

- **datasets** (*nouns*)
- **geometric objects** (*verbs*): visual presentations of the data (points, lines, rectangles, contours..)
- **aesthetics** (*adverbs*): instructions on how to map data variables to geometric objects
- stat. transformations/summaries (line fitting, binning)
- coordinate systems and scales (linear, log, rank)
- layers: overlay
- facets: separating data into multiple subplots
- settings (text size, font, alignment, angle, positioning)

Data must be in *dataframe* format



```r
library(Hiiragi2013)
data(x)
expression <- Biobase::exprs(x)
dftx <- data.frame(pData(x), t(expression))
head(pData(x))
```

```
##          File.name Embryonic.day Total.number.of.cells lineage genotype
## 1 E3.25  1_C32_IN         E3.25                    32               WT
## 2 E3.25  2_C32_IN         E3.25                    32               WT
## 3 E3.25  3_C32_IN         E3.25                    32               WT
## 4 E3.25  4_C32_IN         E3.25                    32               WT
## 5 E3.25  5_C32_IN         E3.25                    32               WT
## 6 E3.25  6_C32_IN         E3.25                    32               WT
##            ScanDate sampleGroup sampleColour
## 1 E3.25 2011-03-16       E3.25      #CAB2D6
## 2 E3.25 2011-03-16       E3.25      #CAB2D6
## 3 E3.25 2011-03-16       E3.25      #CAB2D6
## 4 E3.25 2011-03-16       E3.25      #CAB2D6
## 5 E3.25 2011-03-16       E3.25      #CAB2D6
## 6 E3.25 2011-03-16       E3.25      #CAB2D6
```

```r
dim(expression)
```

```
## [1] 45101   101
```

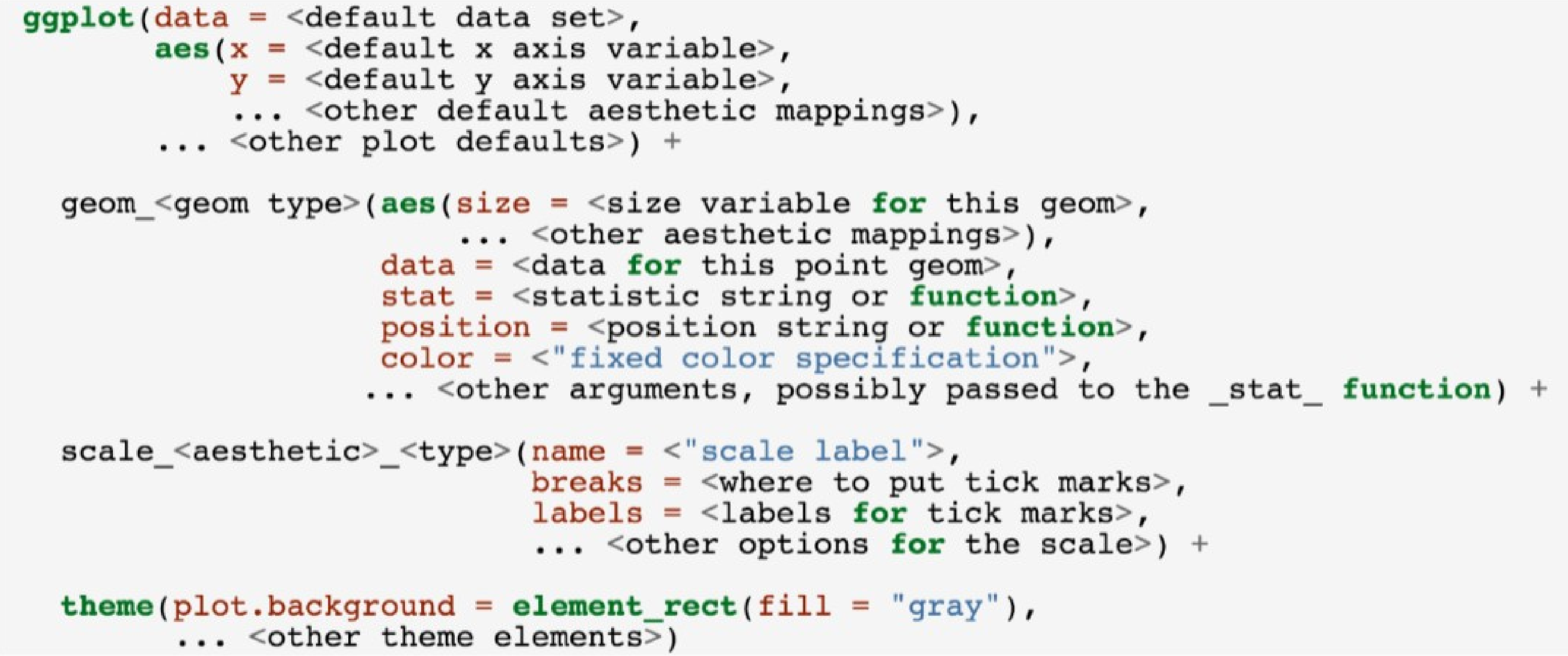
`ggplot()` requires input data in form of a dataframe

Gene expression **microarray dataset on early development of mouse embryos**

transcriptomes of ~100 individual cells at different time points in. [1]

[1] Cell-to-cell expression variability followed by signal reinforcement progressively segregates early mouse lineages by Ohnishi et al., Nature Cell Biology (2014) 16(1): 27-37. doi: 10.1038/ncb2881.

ggplot() template



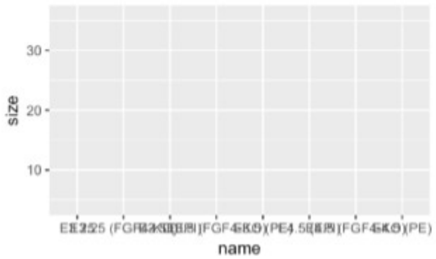
```
ggplot(data = <default data set>,
       aes(x = <default x axis variable>,
           y = <default y axis variable>,
           ... <other default aesthetic mappings>),
       ... <other plot defaults>) +

  geom_<geom type>(aes(size = <size variable for this geom>,
                       ... <other aesthetic mappings>),
                   data = <data for this point geom>,
                   stat = <statistic string or function>,
                   position = <position string or function>,
                   color = <"fixed color specification">,
                  ... <other arguments, possibly passed to the _stat_ function) +

  scale_<aesthetic>_<type>(name = <"scale label">,
                           breaks = <where to put tick marks>,
                           labels = <labels for tick marks>,
                           ... <other options for the scale>) +

  theme(plot.background = element_rect(fill = "gray"),
        ... <other theme elements>)
```

Using the same plot, we can easily change the coordinates

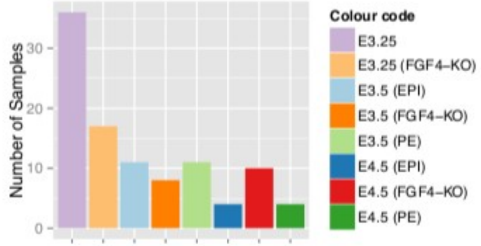
```
groupSize <- table(dftx$sampleGroup)
groupSize
```

```
pb <- ggplot(data.frame(
                 name = names(groupSize),
                 size = as.vector(groupSize)),
             aes(x = name, y = size))
```

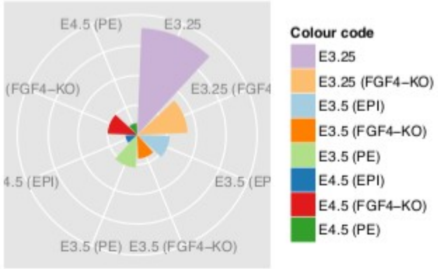
No geom defined yet!



```
pb <- pb + geom_bar(stat = "identity") +
   aes(fill = name) +
   scale_fill_manual(values = groupColour, name = "Colour code") +
   theme(axis.text.x = element_text(angle = 90, hjust = 1))  +
   xlab("Groups") + ylab("Number of Samples")
```

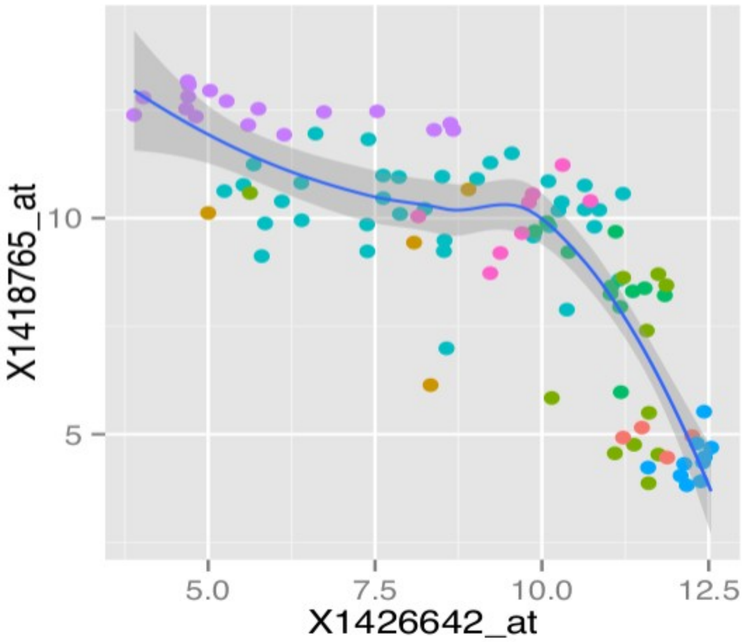


```
pb.polar <- pb + coord_polar() +
  theme(axis.text.x = element_text(angle = 0, hjust = 1),
        axis.text.y = element_blank(),
        axis.ticks = element_blank()) +
  xlab("") + ylab("")
pb.polar
```



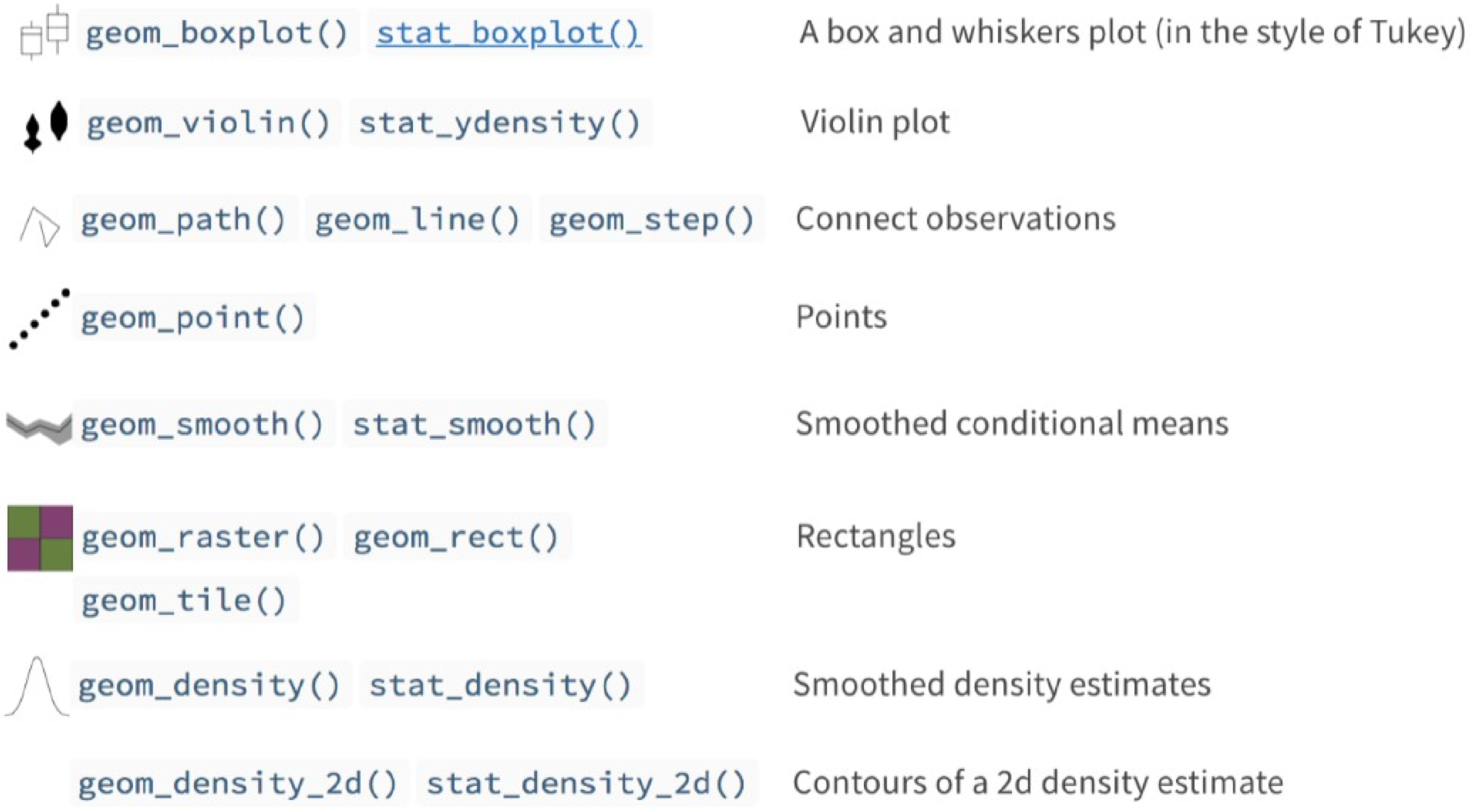
Multiple layers can be superposed

```
ggplot( dftx, aes( x = X1426642_at, y = X1418765_at ))  +
  geom_point( aes( colour = sampleColour), shape = 19 ) +
  geom_smooth( method = "loess" ) +
  scale_colour_discrete( guide = FALSE )
```



Here, the first layers holds the points, the second holds the smoothed average.

geometric objects

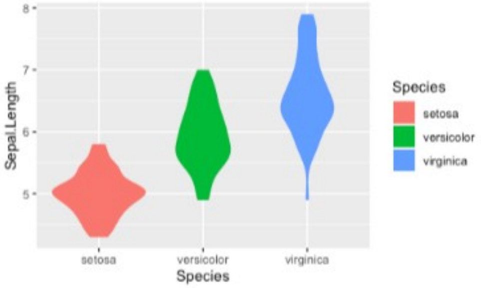


geom_boxplot() stat_boxplot()	A box and whiskers plot (in the style of Tukey)
geom_violin() stat_ydensity()	Violin plot
geom_path() geom_line() geom_step()	Connect observations
geom_point()	Points
geom_smooth() stat_smooth()	Smoothed conditional means
geom_raster() geom_rect() geom_tile()	Rectangles
geom_density() stat_density()	Smoothed density estimates
geom_density_2d() stat_density_2d()	Contours of a 2d density estimate

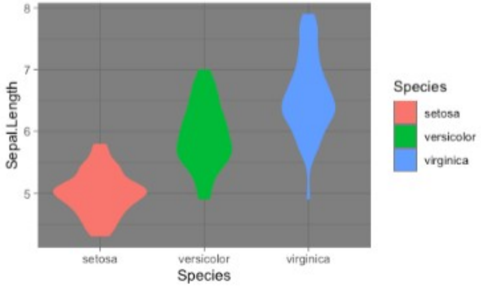
Cheat sheet: https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf

Themes can change the look

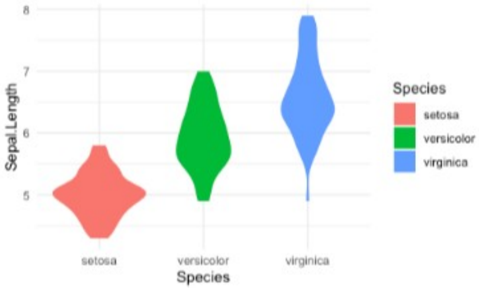
```
g = ggplot(iris,
           aes(x = Species,
               y = Sepal.Length,
               fill = Species))+
  geom_violin(col = NA)
g
```



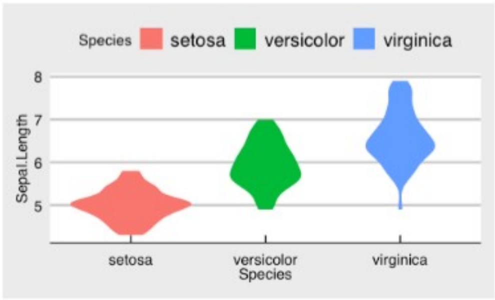
```
g + theme_dark()
```



```
g + theme_minimal()
```

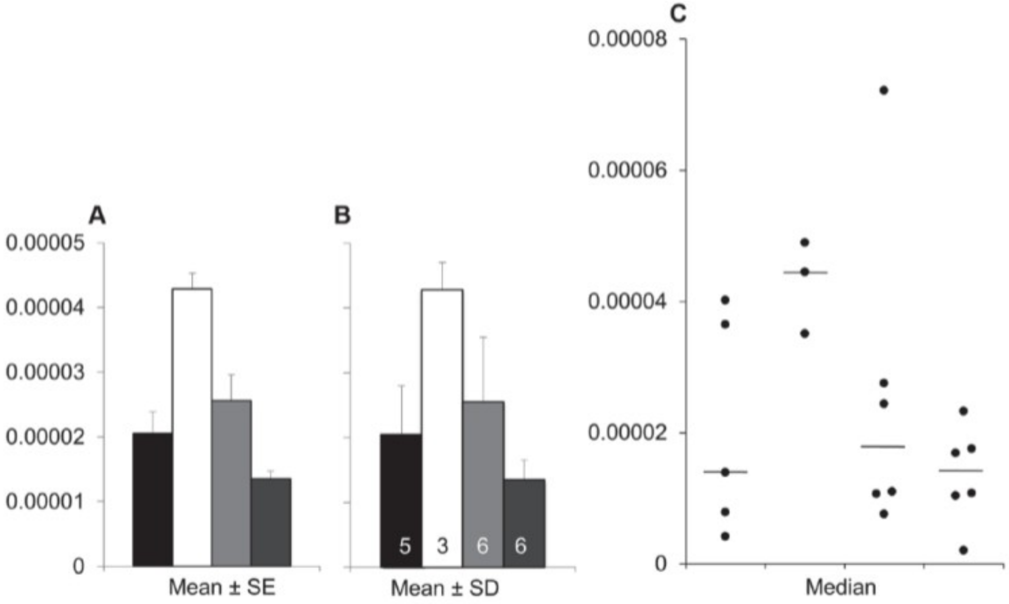


```
library(ggthemes)
g + theme_economist_white()
```



~~Bar charts with error bars~~

What is wrong with {bar charts + error bars} ?



Bar charts (with error bars) **not good for showing distributions**

Use bar charts **only to show class counts.**

Fig 3. Bar graphs and scatterplots convey very different information. While scatterplots prompt the reader to critically evaluate the statistical tests and the authors' interpretation of the data, bar graphs discourage the reader from thinking about these issues. Placental endothelin 1 (*EDN1*) mRNA data for four different groups of participants is presented in bar graphs showing mean ± SE (Panel A), or mean ± SD (Panel B), and in a univariate scatterplot (Panel C). Panel A (mean ± SE) suggests that the second group has higher values than the remaining groups; however, Panel B (mean ± SD) reveals that there is considerable overlap between groups. Showing SE rather than SD magnifies the apparent visual differences between groups, and this is exacerbated by the fact that SE obscures any effect of unequal sample size. The scatterplot (Panel C) clearly shows that the sample sizes are small, group one has a much larger variance than the other groups, and there is an outlier in group three. These problems are not apparent in the bar graphs shown in Panels A and B.

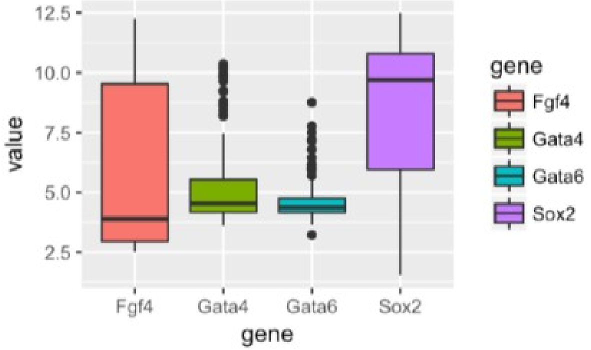
doi:10.1371/journal.pbio.1002128.g003

Weissgerber TL, et al. (2015) Beyond Bar and Line Graphs: Time for a New Data Presentation Paradigm. PLOS Biology 13(4): e1002128.

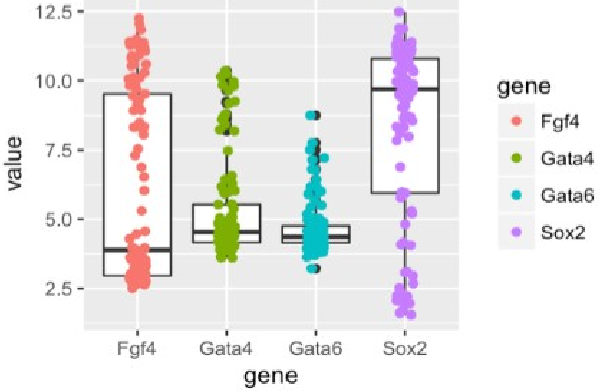
Boxplot

Boxplots are good for plotting summary of 1D continuous data; they allow you to **compare quantiles of data distributions.**

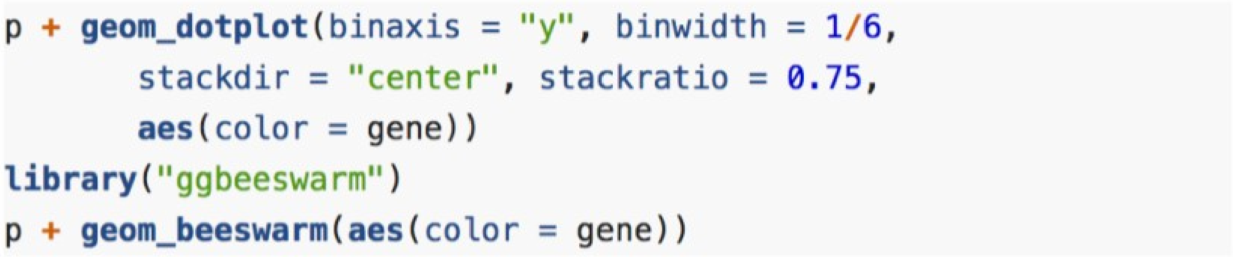
```
p = ggplot(genes, aes( x = gene, y = value))
p + geom_boxplot(aes(fill = gene))
```



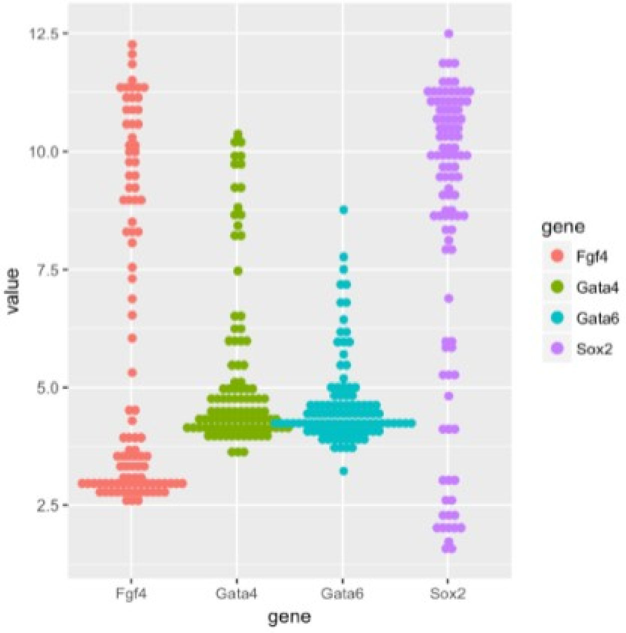
```
p + geom_boxplot() +
  geom_jitter(aes(color = gene), width = 0.1, height = 0)
```

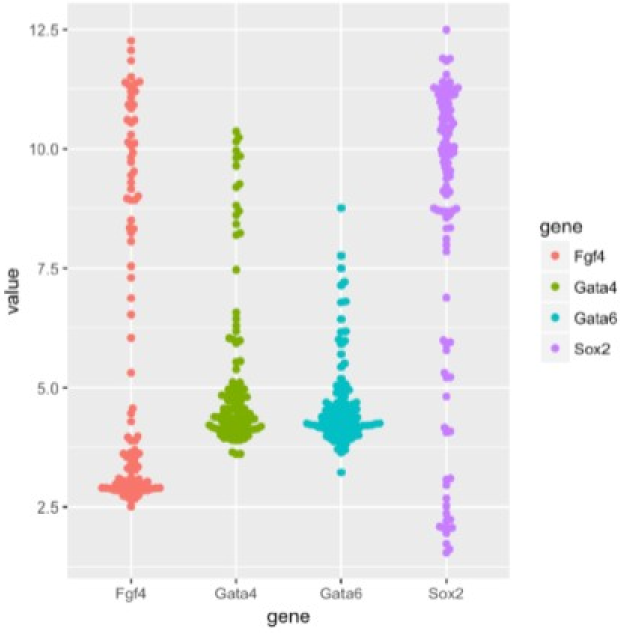


Dot & Beeswarm Plot



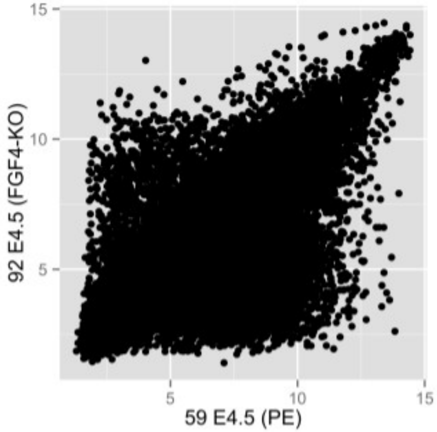
```
p + geom_dotplot(binaxis = "y", binwidth = 1/6,
       stackdir = "center", stackratio = 0.75,
       aes(color = gene))
library("ggbeeswarm")
p + geom_beeswarm(aes(color = gene))
```



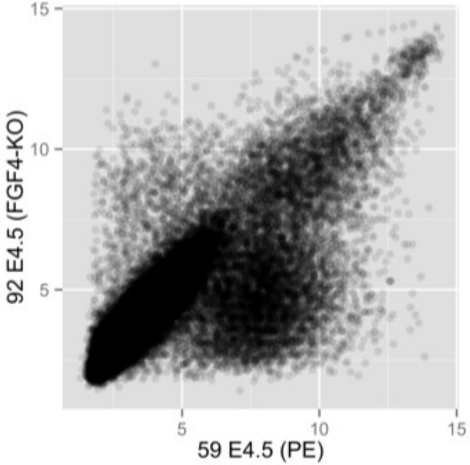


Showing distributions in 2D

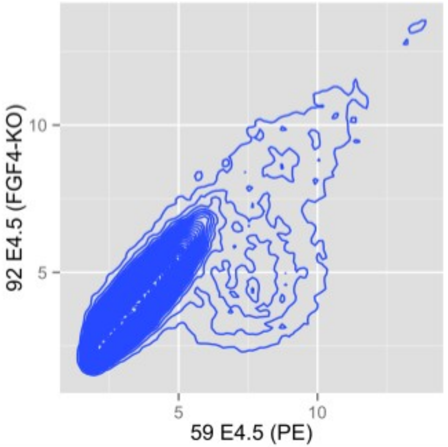
```
scp <- ggplot(dfx, aes( x = ‘59 E4.5 (PE)‘ ,
                        y = ‘92 E4.5 (FGF4-KO)‘))
scp + geom_point()
```



```
scp  + geom_point(alpha = 0.1)
```

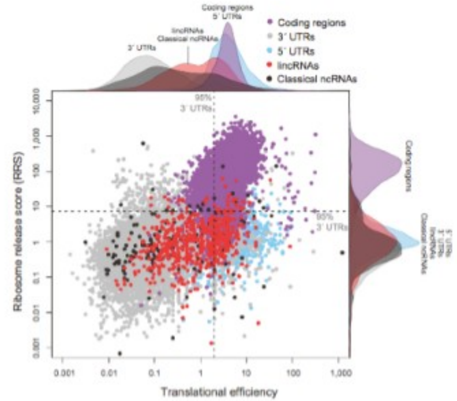


```
scp + geom_density2d(h = 0.5, bins = 60)
```

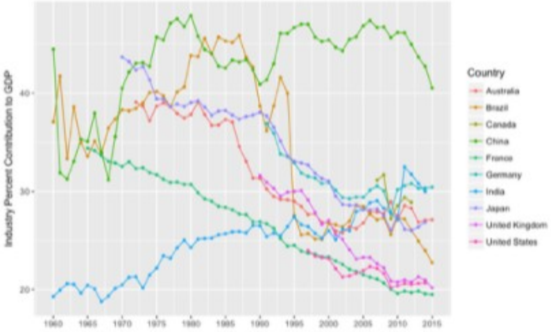


2D data plots

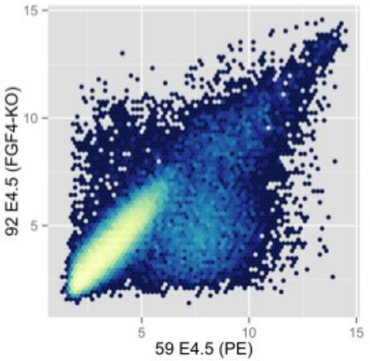
Scatterplots (x,y)-point plots



Line plots (x,y)-line plots



2D density requires the choice of bandwidth; obscures the sample size (i.e. the uncertainty of the estimate)



3-5D: **aesthetics** allow to show more than 2D

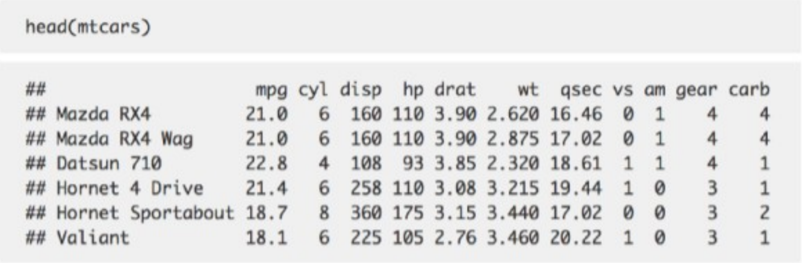
geom_point's

aesthetics

(apart from x and y):

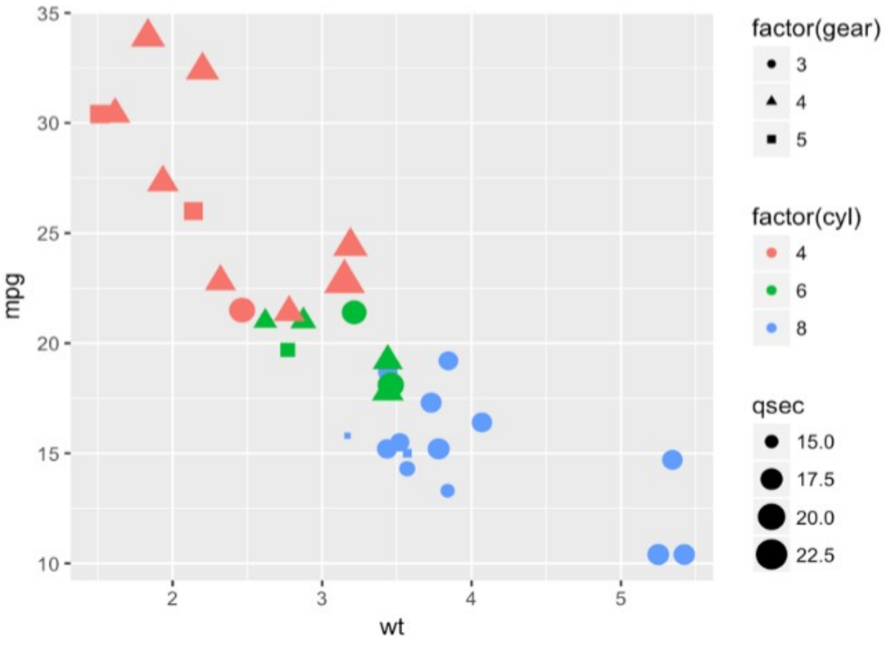
- fill / color
- shape
- size
- alpha

```
ggplot(data = mtcars) +
  geom_point(
    aes(x = wt, y = mpg,
        shape = factor(gear),
        color = factor(cyl),
        size = qsec))
```



```
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
```



3-5D: **aesthetics** allow to show more than 2D

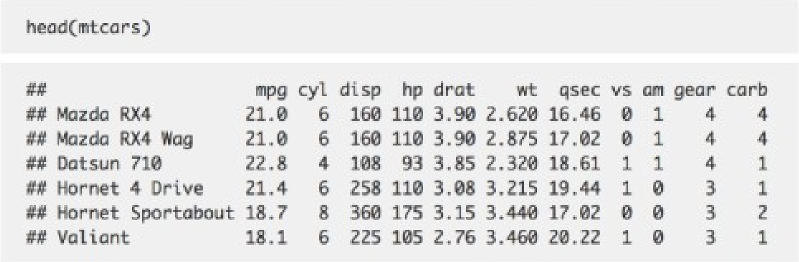
geom_point's

aesthetics

(apart from x and y):

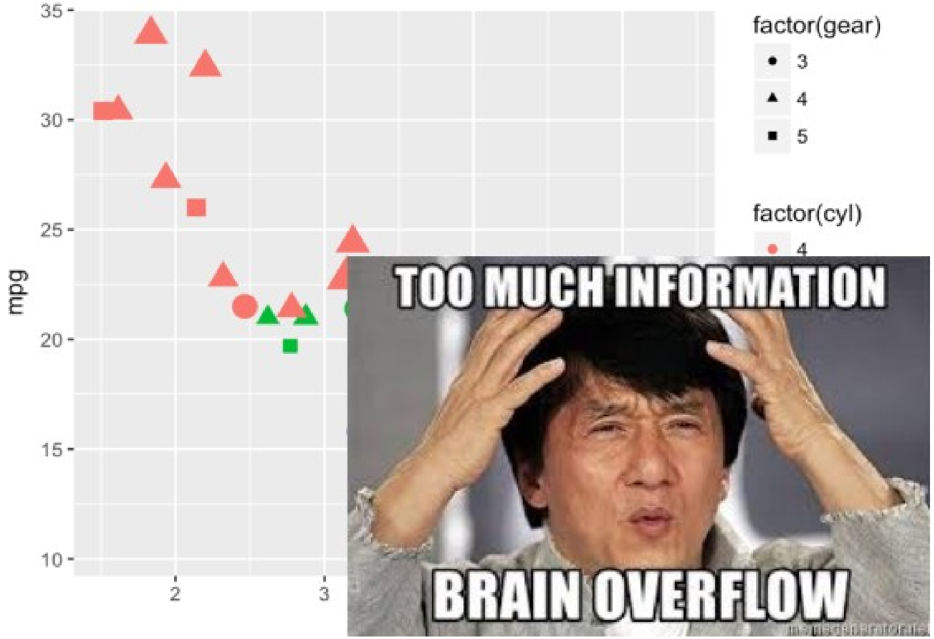
- fill / color
- shape
- size
- alpha

```
ggplot(data = mtcars) +
  geom_point(
    aes(x = wt, y = mpg,
        shape = factor(gear),
        color = factor(cyl),
        size = qsec))
```

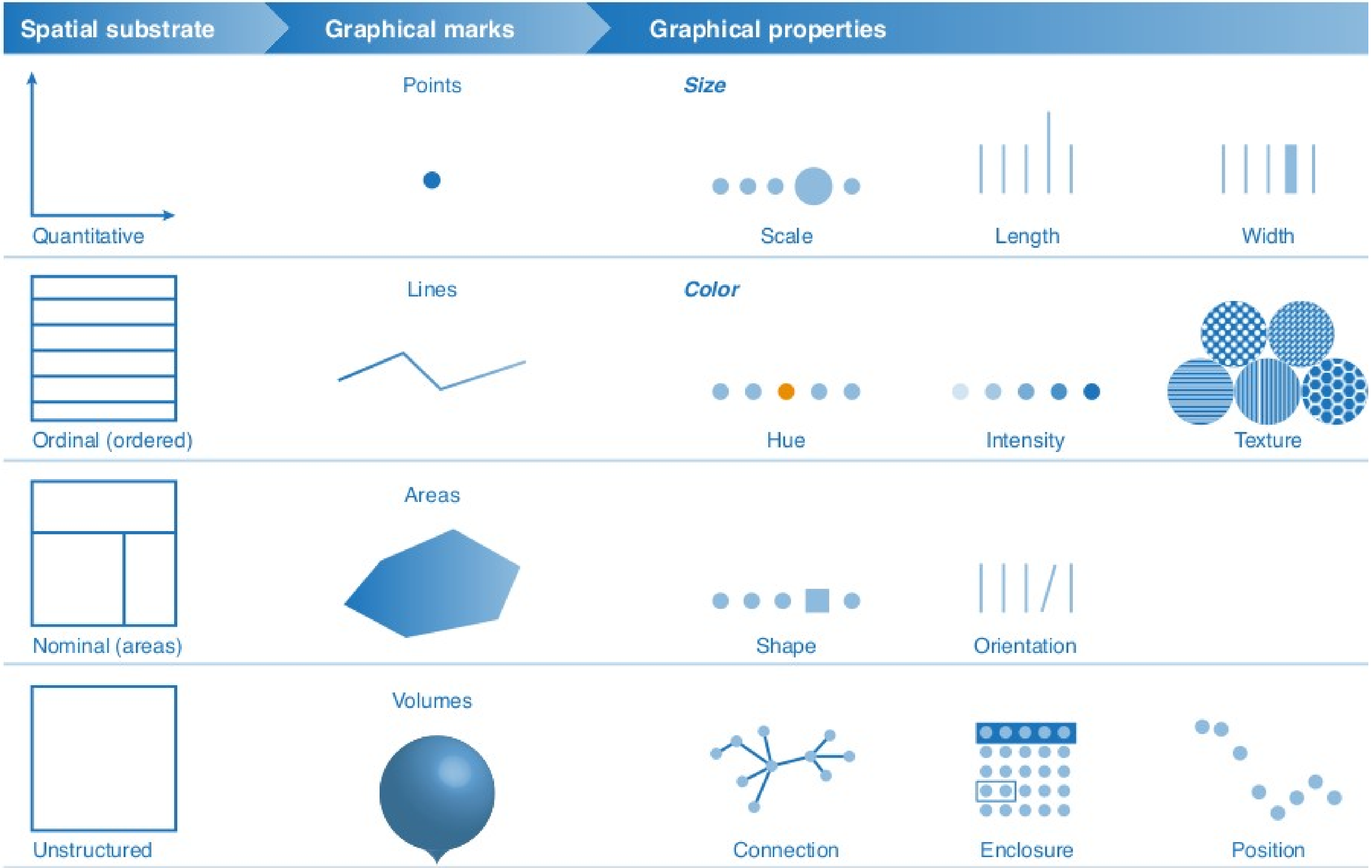


```
head(mtcars)
```

```
##                    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
```



A diversity of **graphical properties (aesthetics)** are available to show dimensions



Spatial substrate

Graphical marks

Graphical properties

Quantitative

Points

Size

Scale

Length

Width

Ordinal (ordered)

Lines

Color

Hue

Intensity

Texture

Nominal (areas)

Areas

Shape

Orientation

Unstructured

Volumes

Connection

Enclosure

Position

Faceting is useful to show more dimensions without overcrowding the graph

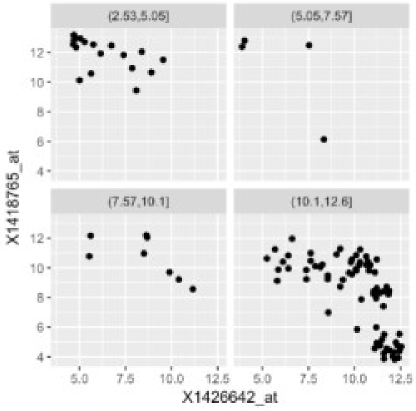
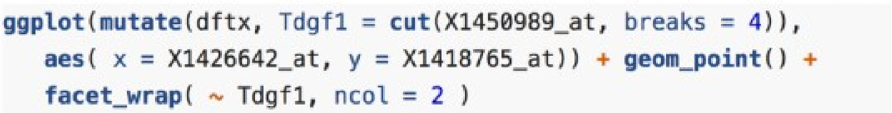


Figure 3.33: Faceting: the same data as in Figure 3.9, split by the continuous variable X1450989_at and arranged by facet_wrap.

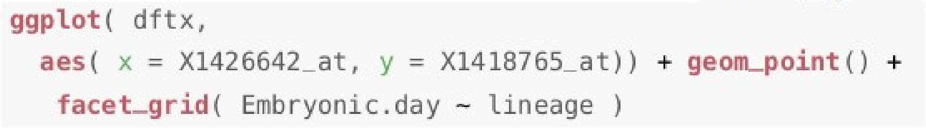
Trellis — chart that uses multiple instances of the same chart

facet_wrap

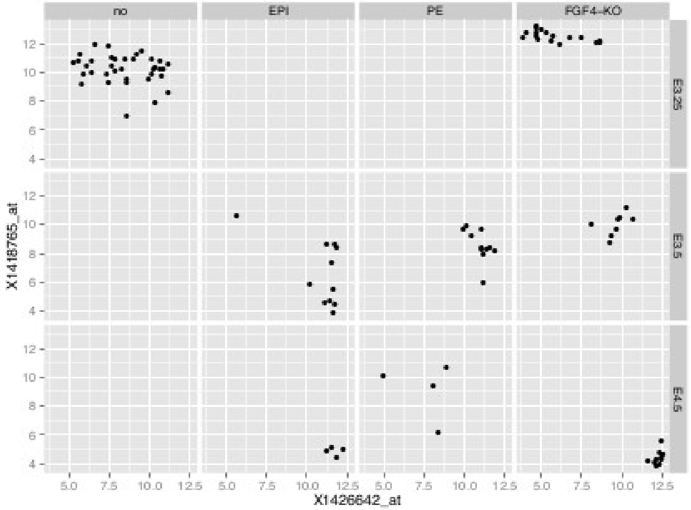


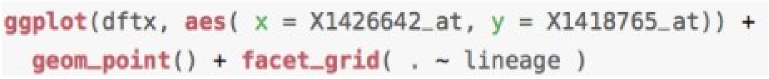
```
ggplot(mutate(dftx, Tdgf1 = cut(X1450989_at, breaks = 4)),
   aes( x = X1426642_at, y = X1418765_at)) + geom_point() +
   facet_wrap( ~ Tdgf1, ncol = 2 )
```

facet_grid

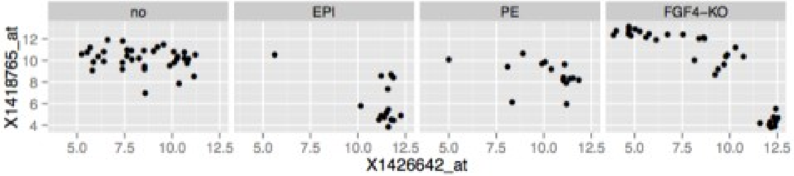


```
ggplot( dftx,
  aes( x = X1426642_at, y = X1418765_at)) + geom_point() +
   facet_grid( Embryonic.day ~ lineage )
```

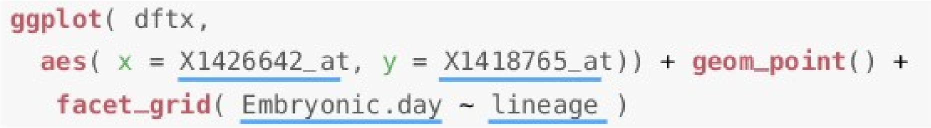




```
ggplot(dftx, aes( x = X1426642_at, y = X1418765_at)) +
  geom_point() + facet_grid( . ~ lineage )
```



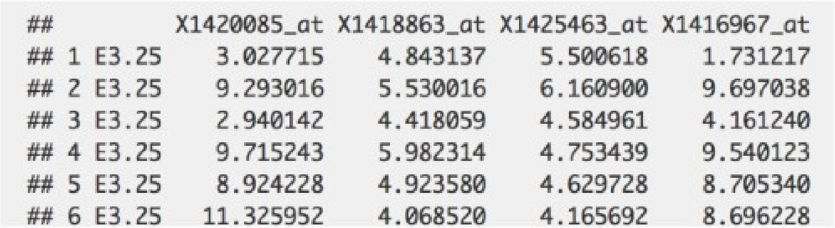
Tidying data to use columns as aesthetics



```
ggplot( dftx,
  aes( x = X1426642_at, y = X1418765_at)) + geom_point() +
   facet_grid( Embryonic.day ~ lineage )
```

Data.frame in R can be in:

wide format

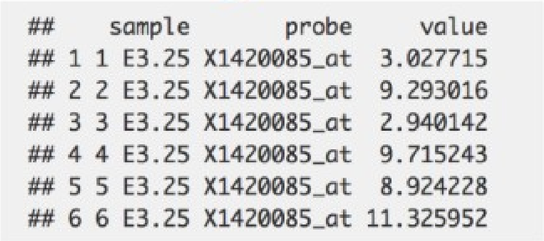


```
##          X1420085_at X1418863_at X1425463_at X1416967_at
## 1 E3.25     3.027715    4.843137    5.500618    1.731217
## 2 E3.25     9.293016    5.530016    6.160900    9.697038
## 3 E3.25     2.940142    4.418059    4.584961    4.161240
## 4 E3.25     9.715243    5.982314    4.753439    9.540123
## 5 E3.25     8.924228    4.923580    4.629728    8.705340
## 6 E3.25    11.325952    4.068520    4.165692    8.696228
```

e.g. a expression matrix with each raw containing a gene expression for all samples

Each **row** corresponds to a **sample** and each **column** to a **feature** (or vice versa).

long format



```
##    sample       probe     value
## 1 1 E3.25 X1420085_at  3.027715
## 2 2 E3.25 X1420085_at  9.293016
## 3 3 E3.25 X1420085_at  2.940142
## 4 4 E3.25 X1420085_at  9.715243
## 5 5 E3.25 X1420085_at  8.924228
## 6 6 E3.25 X1420085_at 11.325952
```

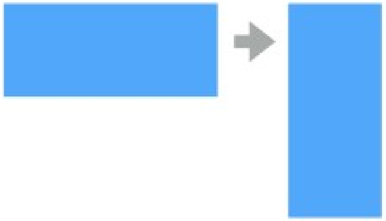
e.g. a collapsed expression data with each row corresponding to a gene-sample pair

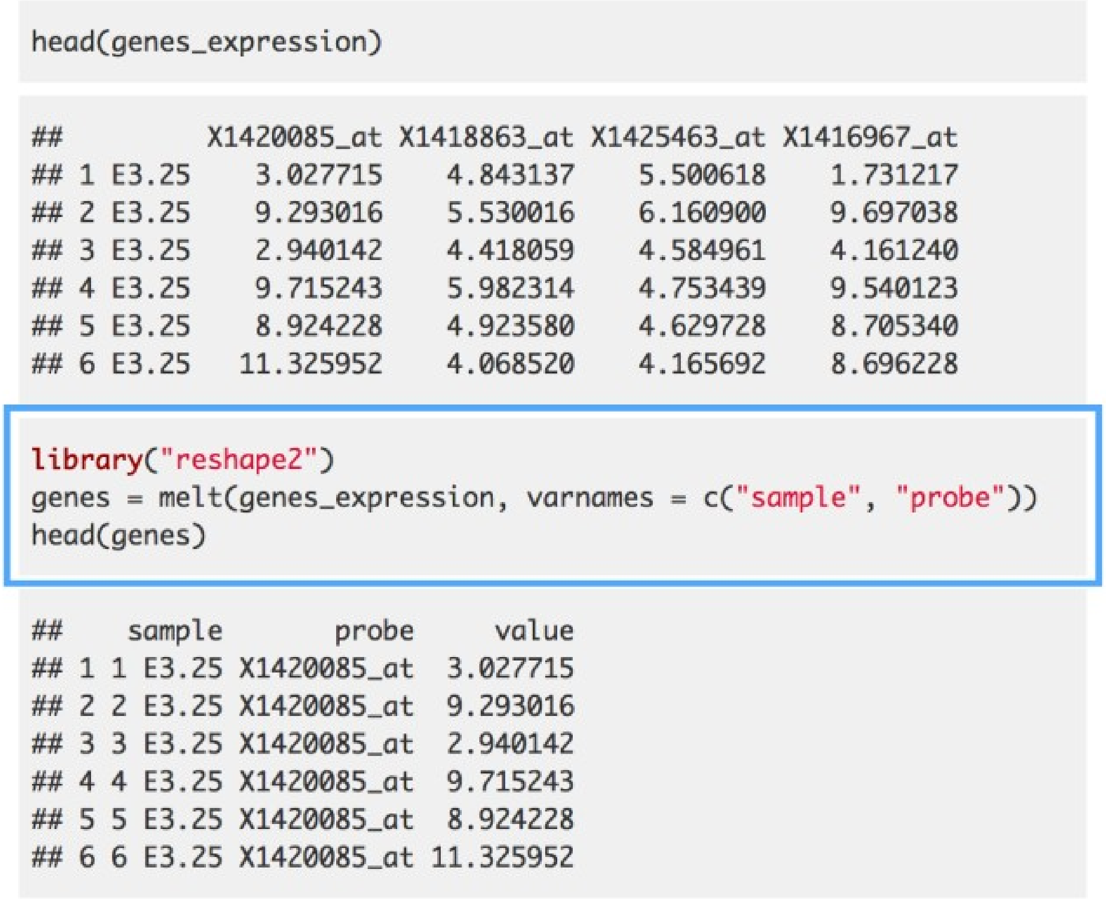
Feature and **sample** information is stored separately for each measurement in data columns.

How do you switch :wide: <> :long: ?

`melt()` from wide to long

To convert between two data formats we can use functions **`melt()`**, and **`dcast()`** from package **`reshape2`**





```
head(genes_expression)
```

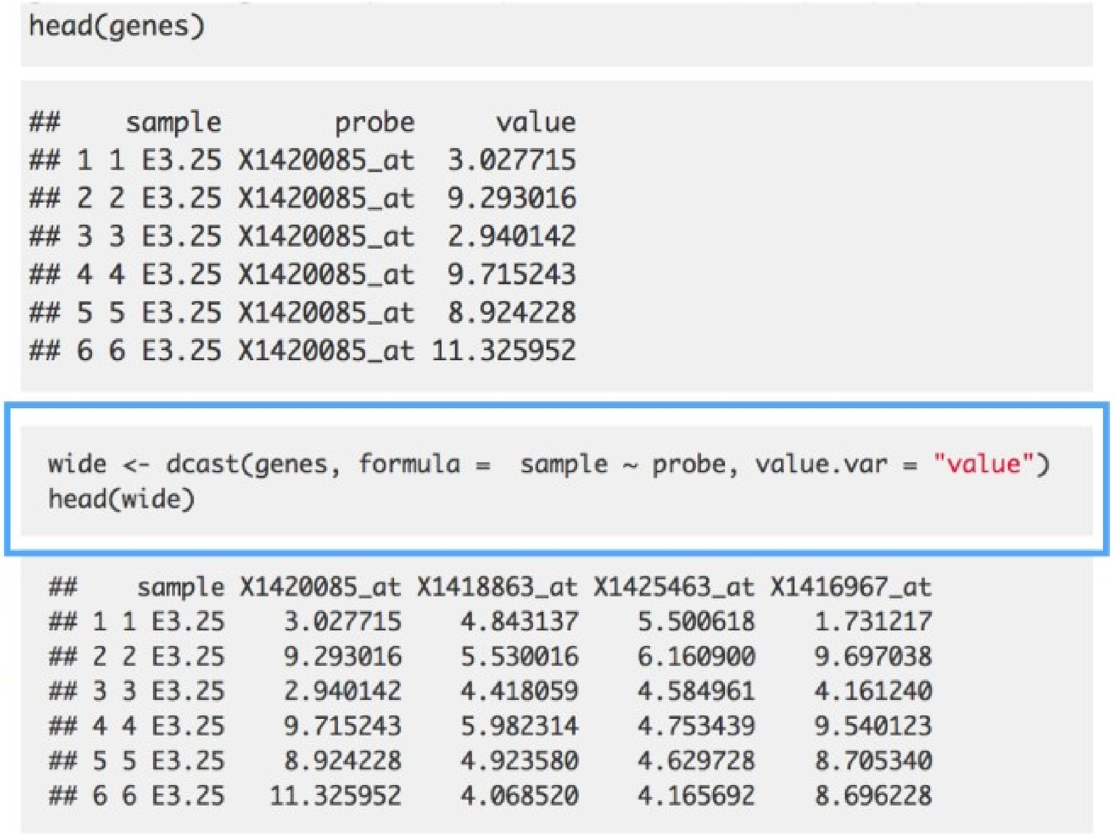
```
##           X1420085_at X1418863_at X1425463_at X1416967_at
## 1 E3.25      3.027715    4.843137    5.500618    1.731217
## 2 E3.25      9.293016    5.530016    6.160900    9.697038
## 3 E3.25      2.940142    4.418059    4.584961    4.161240
## 4 E3.25      9.715243    5.982314    4.753439    9.540123
## 5 E3.25      8.924228    4.923580    4.629728    8.705340
## 6 E3.25     11.325952    4.068520    4.165692    8.696228
```

```
library("reshape2")
genes = melt(genes_expression, varnames = c("sample", "probe"))
head(genes)
```

```
##     sample       probe     value
## 1 1 E3.25 X1420085_at  3.027715
## 2 2 E3.25 X1420085_at  9.293016
## 3 3 E3.25 X1420085_at  2.940142
## 4 4 E3.25 X1420085_at  9.715243
## 5 5 E3.25 X1420085_at  8.924228
## 6 6 E3.25 X1420085_at 11.325952
```

dcast() from long to wide

To convert between two data formats we can use functions **melt()**, and **dcast()** from package **reshape2**

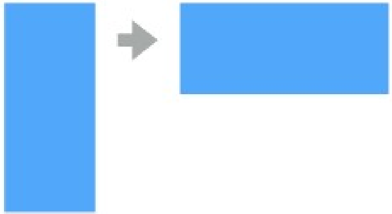


```
head(genes)
```

```
##    sample        probe     value
## 1 1 E3.25 X1420085_at  3.027715
## 2 2 E3.25 X1420085_at  9.293016
## 3 3 E3.25 X1420085_at  2.940142
## 4 4 E3.25 X1420085_at  9.715243
## 5 5 E3.25 X1420085_at  8.924228
## 6 6 E3.25 X1420085_at 11.325952
```

```
wide <- dcast(genes, formula =  sample ~ probe, value.var = "value")
head(wide)
```

```
##    sample X1420085_at X1418863_at X1425463_at X1416967_at
## 1 1 E3.25    3.027715    4.843137    5.500618    1.731217
## 2 2 E3.25    9.293016    5.530016    6.160900    9.697038
## 3 3 E3.25    2.940142    4.418059    4.584961    4.161240
## 4 4 E3.25    9.715243    5.982314    4.753439    9.540123
## 5 5 E3.25    8.924228    4.923580    4.629728    8.705340
## 6 6 E3.25   11.325952    4.068520    4.165692    8.696228
```



Interactivity

Use shiny or plotly

https://shiny.rstudio.com/gallery/genome-browser.html

Animations (time-dependent plots):
https://gganimate.com

Linked Charts
https://anders-biostat.github.io/linked-charts/

NB: ggvis is senescent

Acknowledgments

- Susan Holmes
- Wolfgang Huber
- Sudarshan Shetty
- Wisam Saleem