R course

Data visualization

Daniel Vaulot

2025-01-16

R - Session 03

- Graph types
- Grammar of graphics
- Playing with ggplot2
- Multiple graphs
- ggplot2 syntax

Intro to Data vizualisation

Installation and Resources

Packages

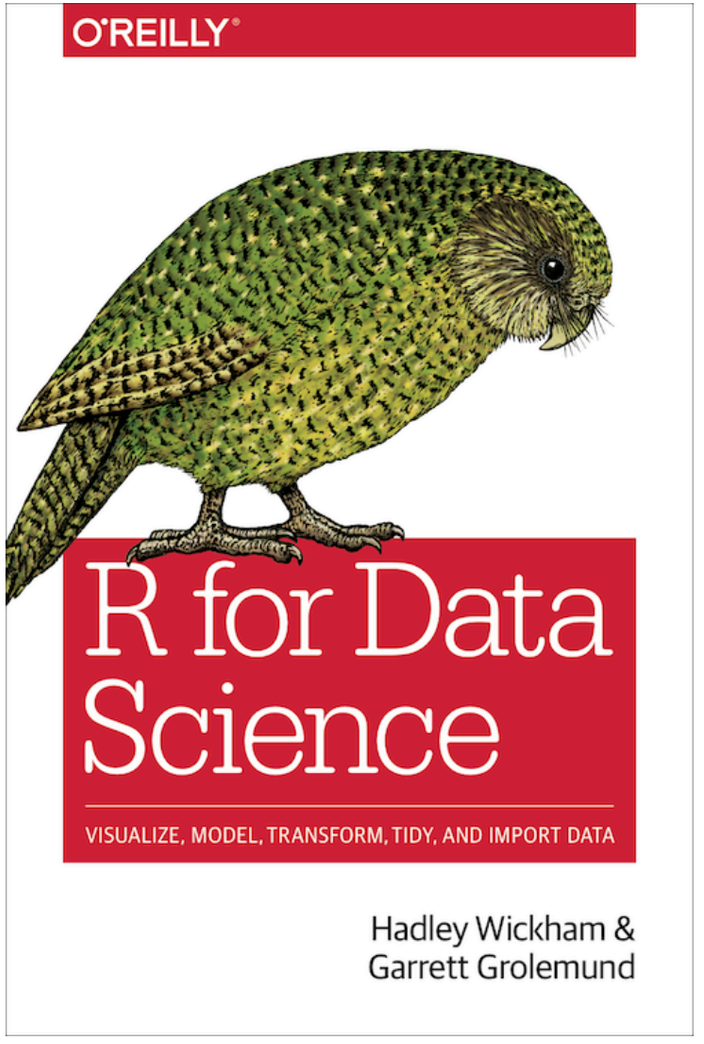
- ggplot2
- patchwork

Reading

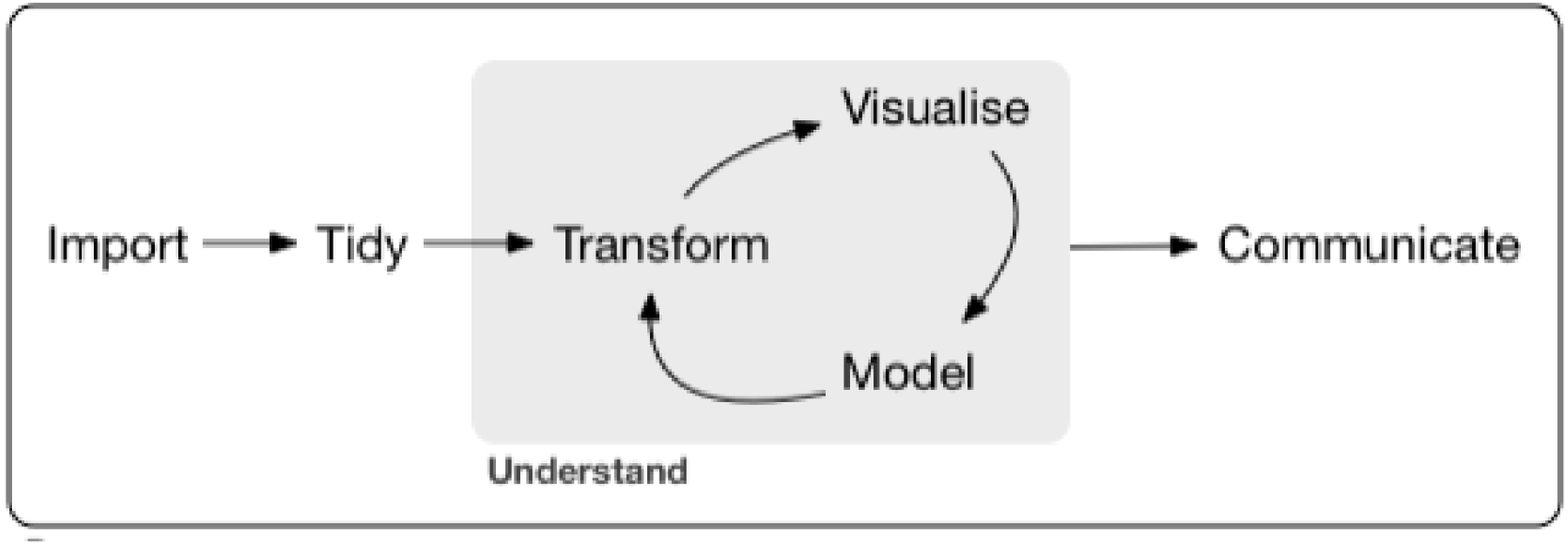
- Chapter 28 of R for data science

Resources

- Fundamental of data visualization
- Data visualization: practical introduction



Workflow



Graph purposes

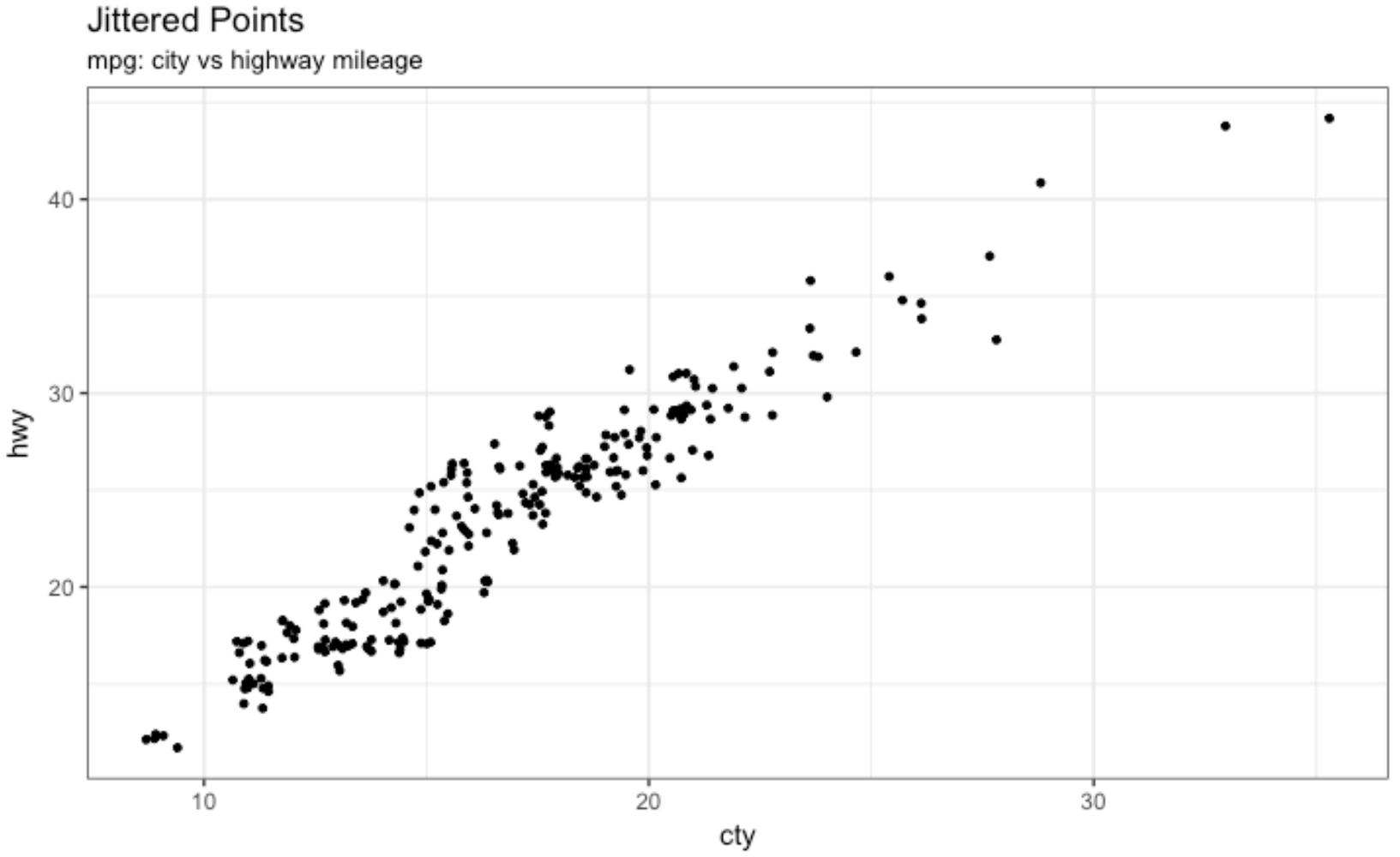
- **Analysis graphs**
  - design to see patterns, trends
  - aid the process of data description
  - interpretation
- **Presentation graphs**
  - design to attract attention
  - make a point
  - illustrate a conclusion

Source: Michael Friendly

Graph types

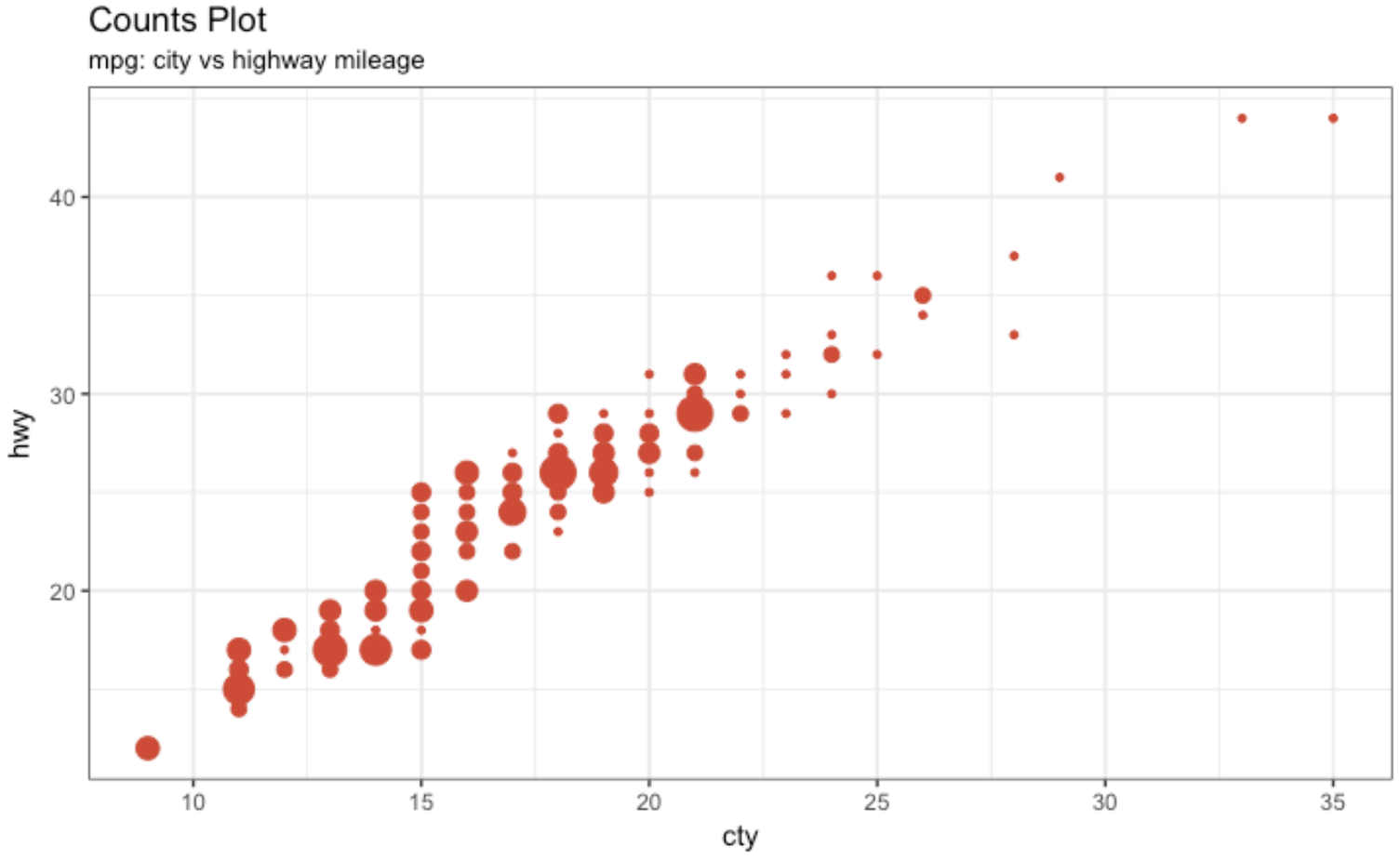
Jitter

- Two variables numerical



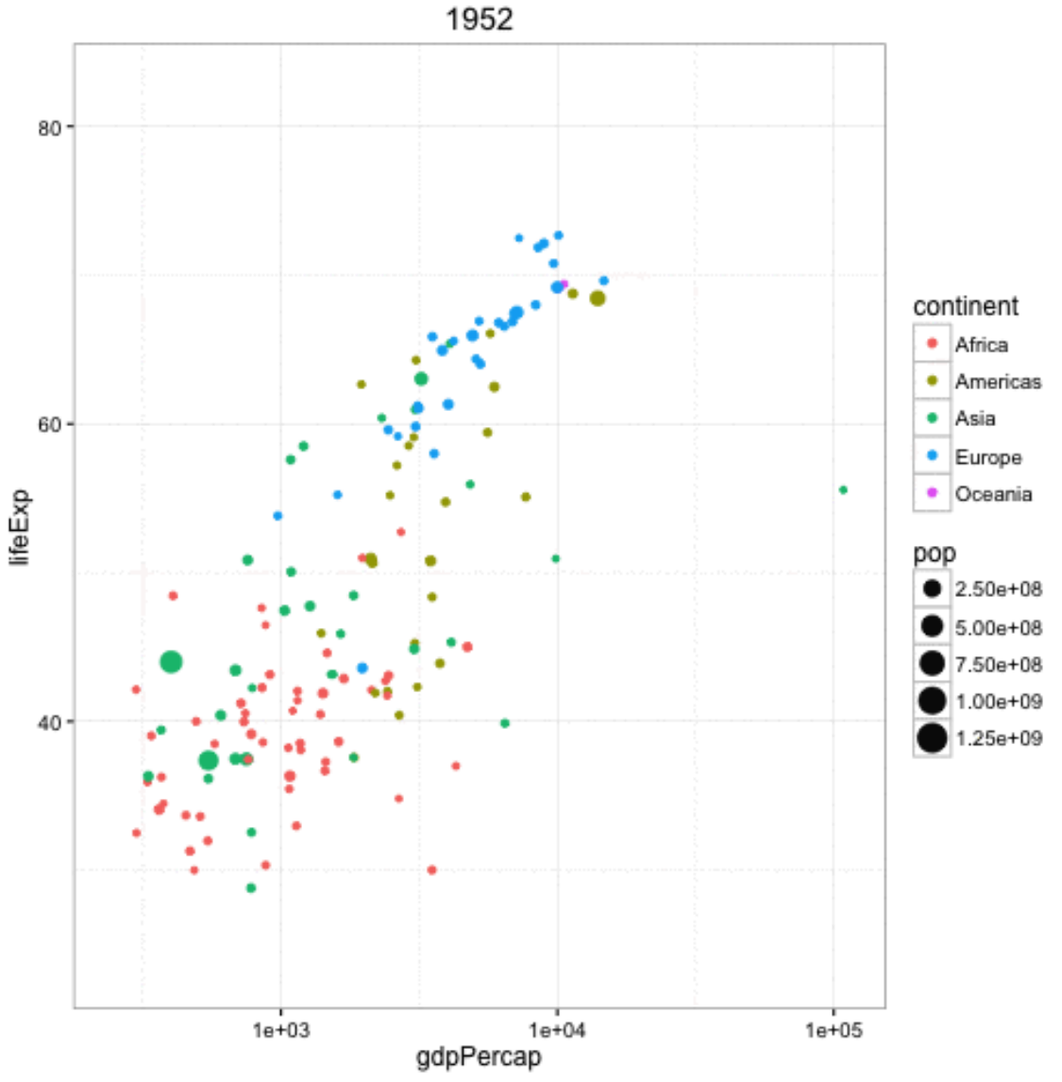
Bubble

- Two variables numerical
- **Add another variable numerical**



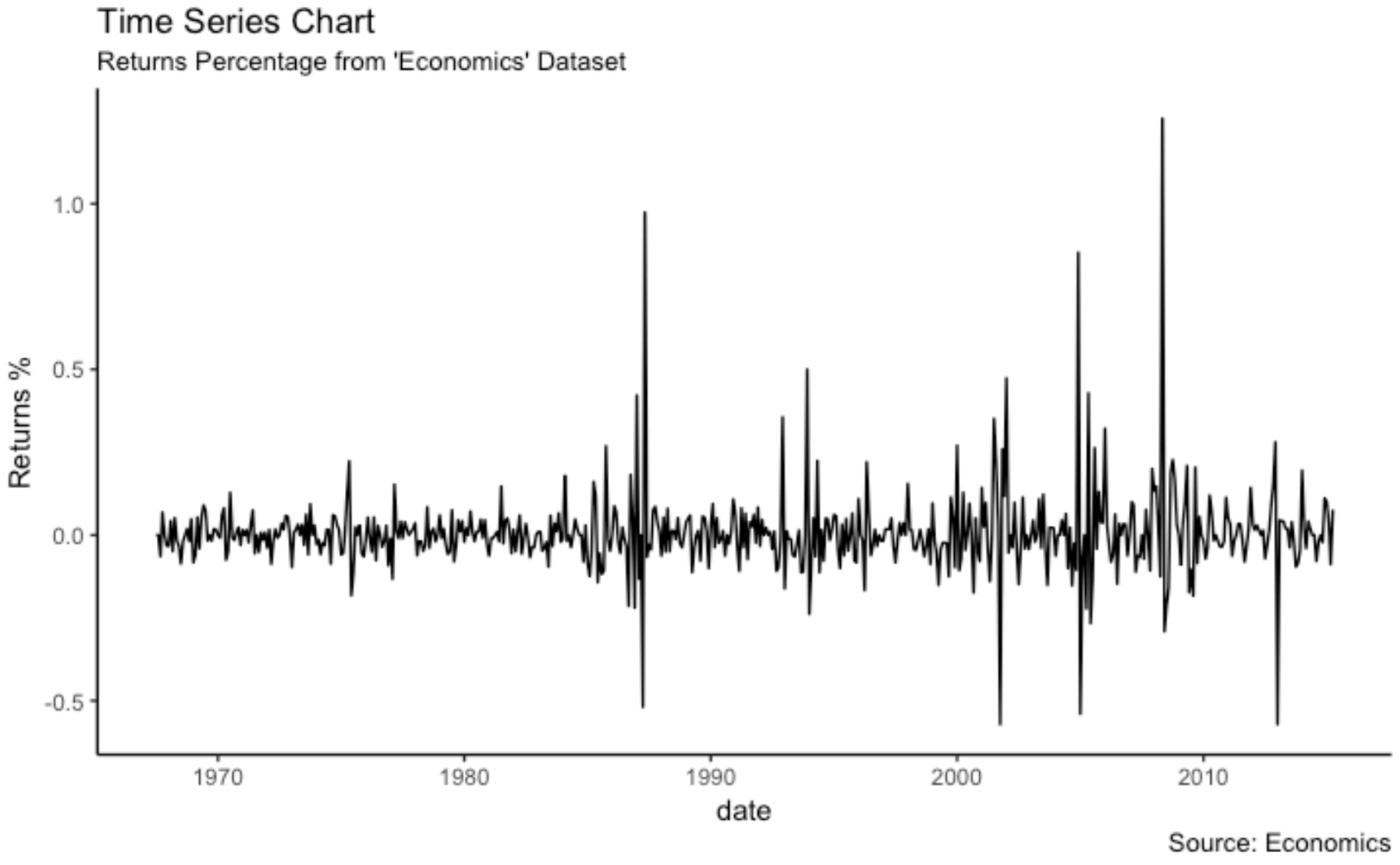
Animate

- Two variables numerical
- One variable numerical
- One variable categorical
- **Animate another variable**



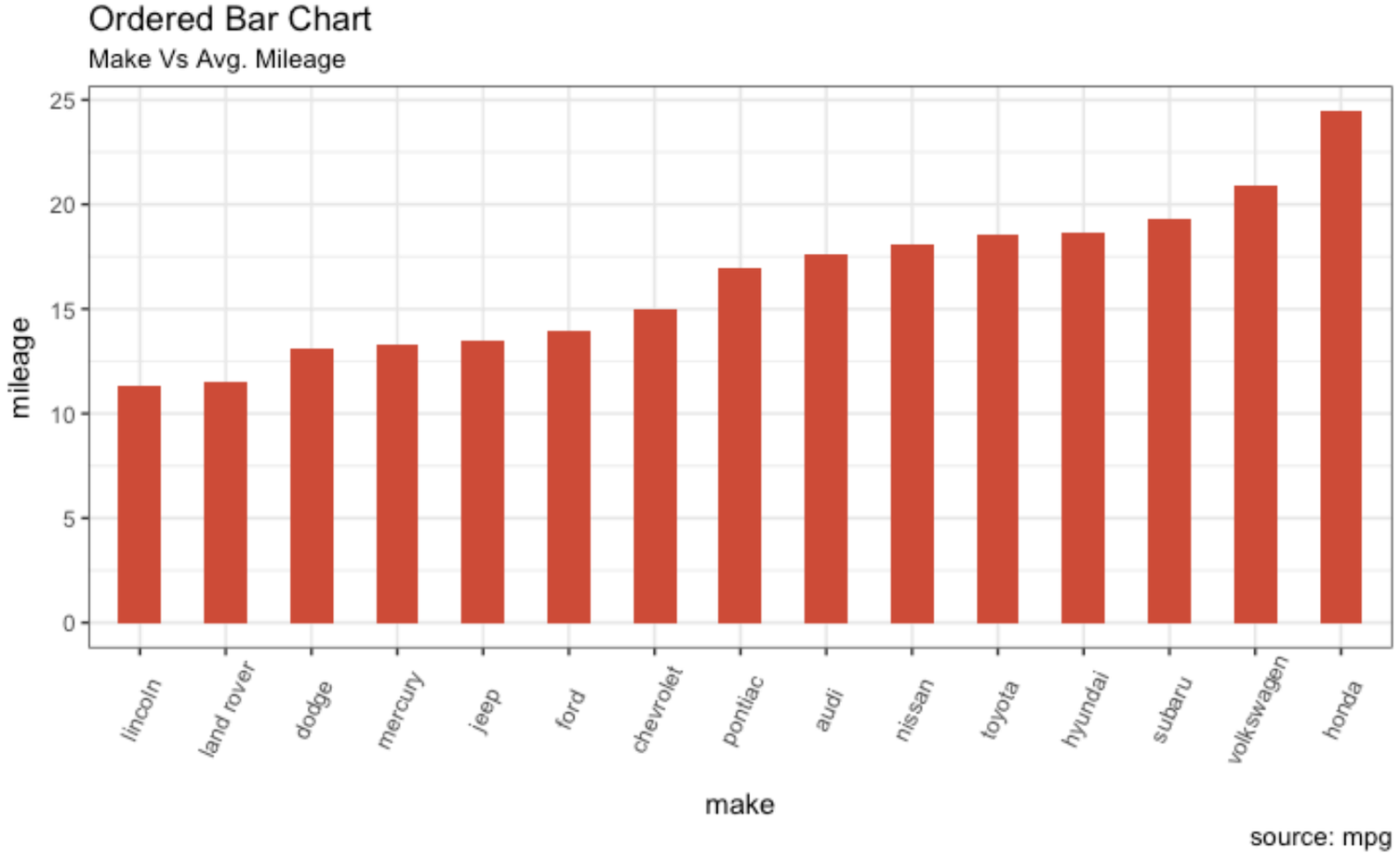
Times series

- Line graph



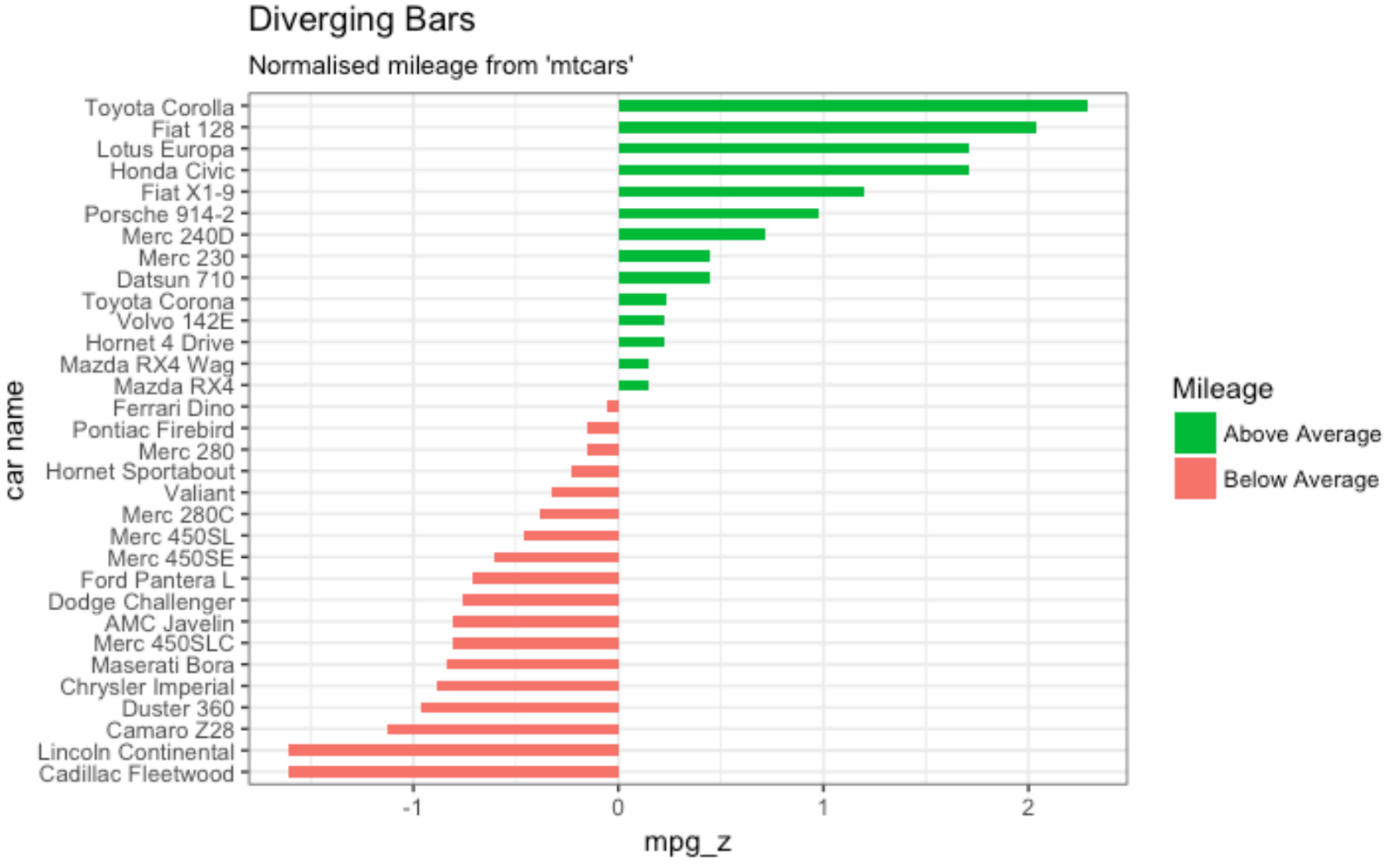
Bargraphs

- One variable categorical
- One variable numerical



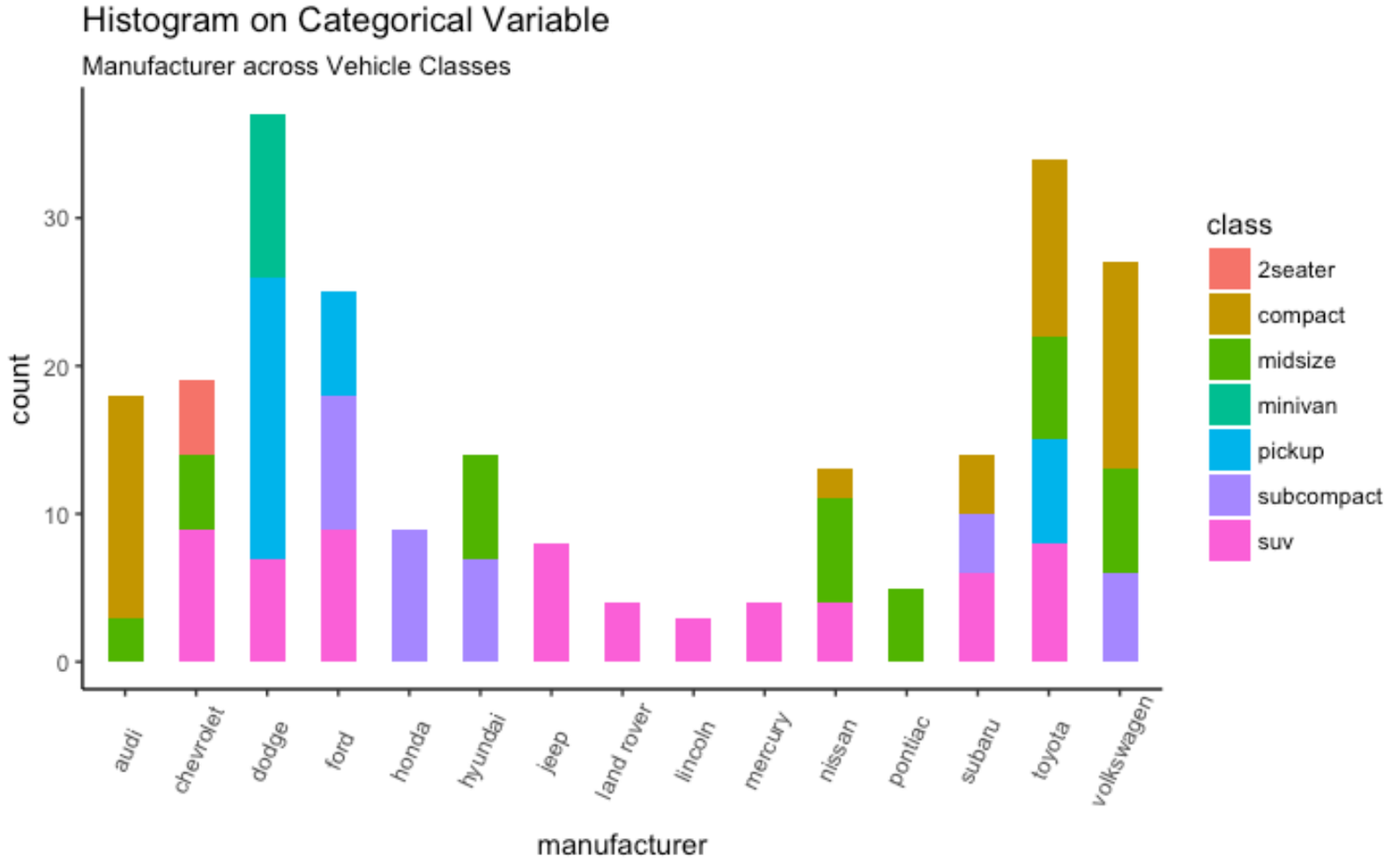
Bargraphs

- Rotate



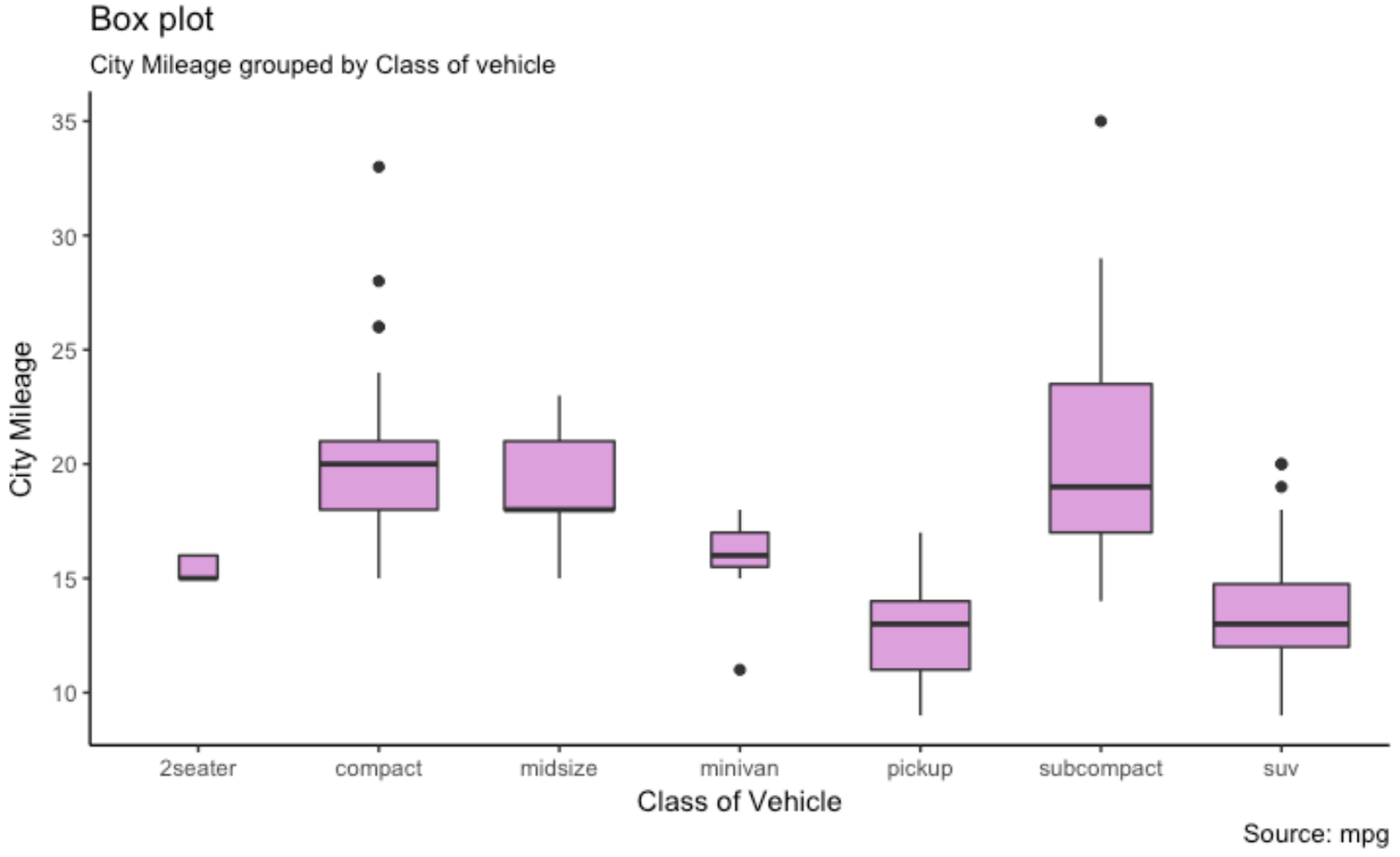
Bargraphs

- Two variable categorical
- One variable numerical



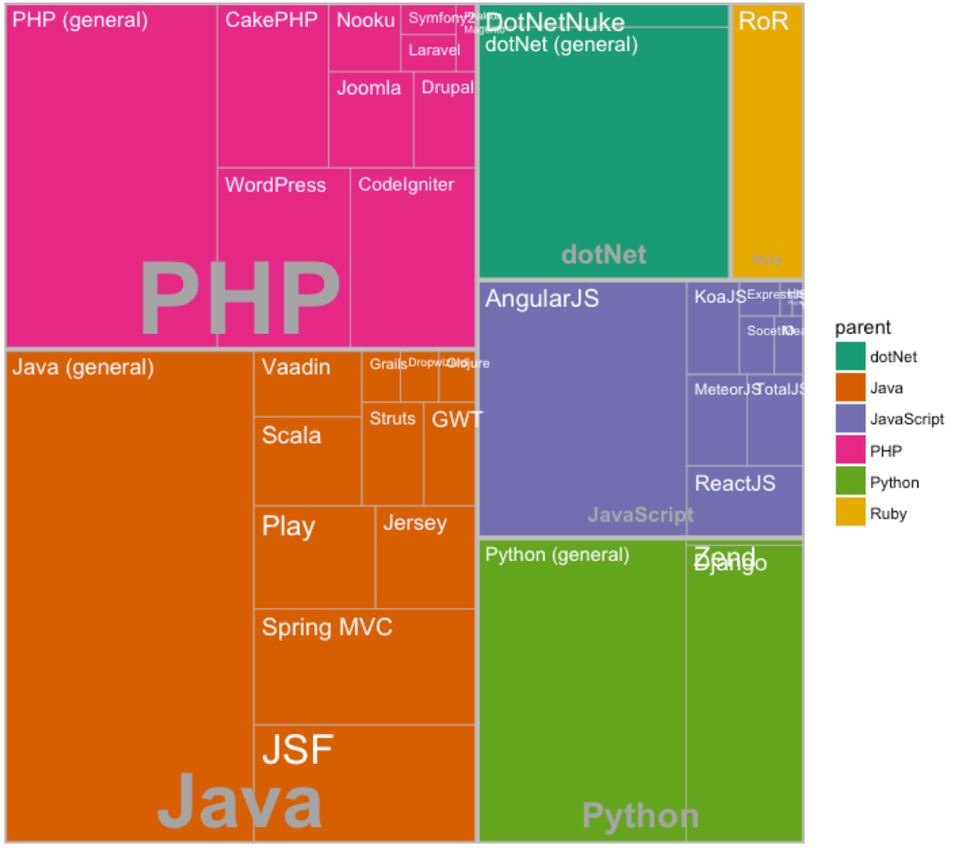
Boxplots

- One variable categorical
- One variable numerical but with many values



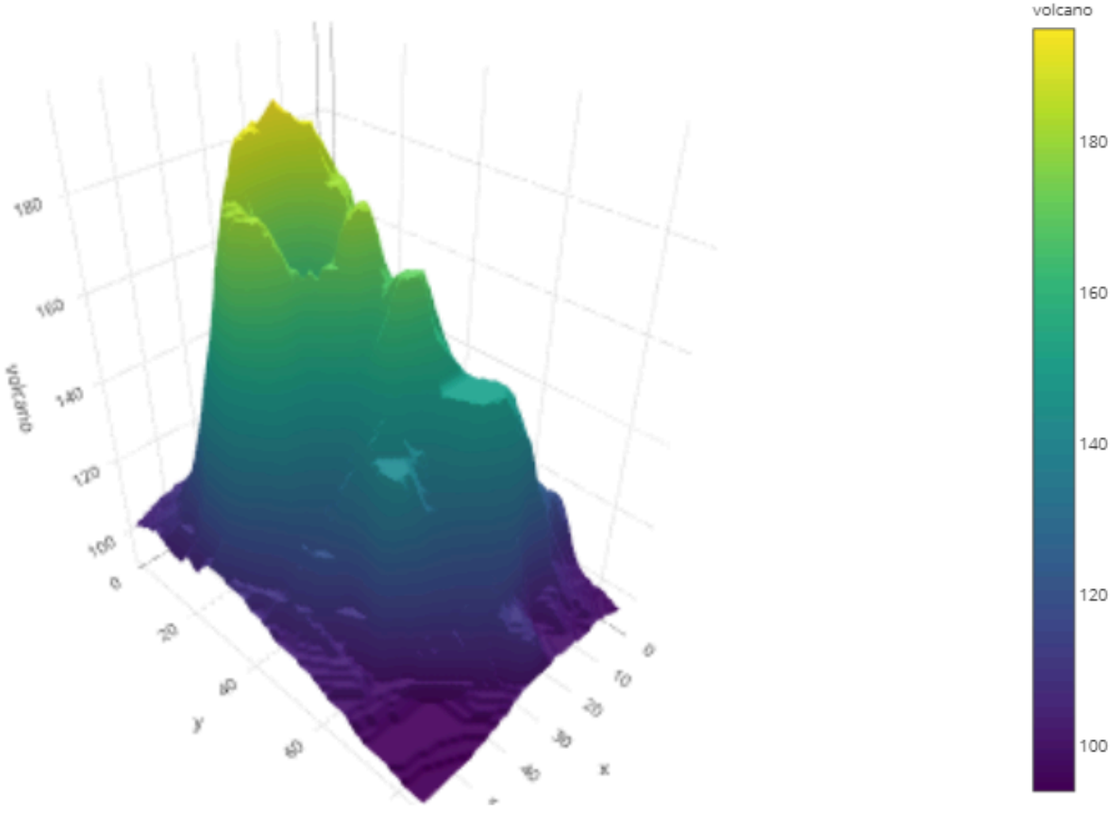
Treemaps

- One variable categorical
- One variable numerical
- Much better than pie charts



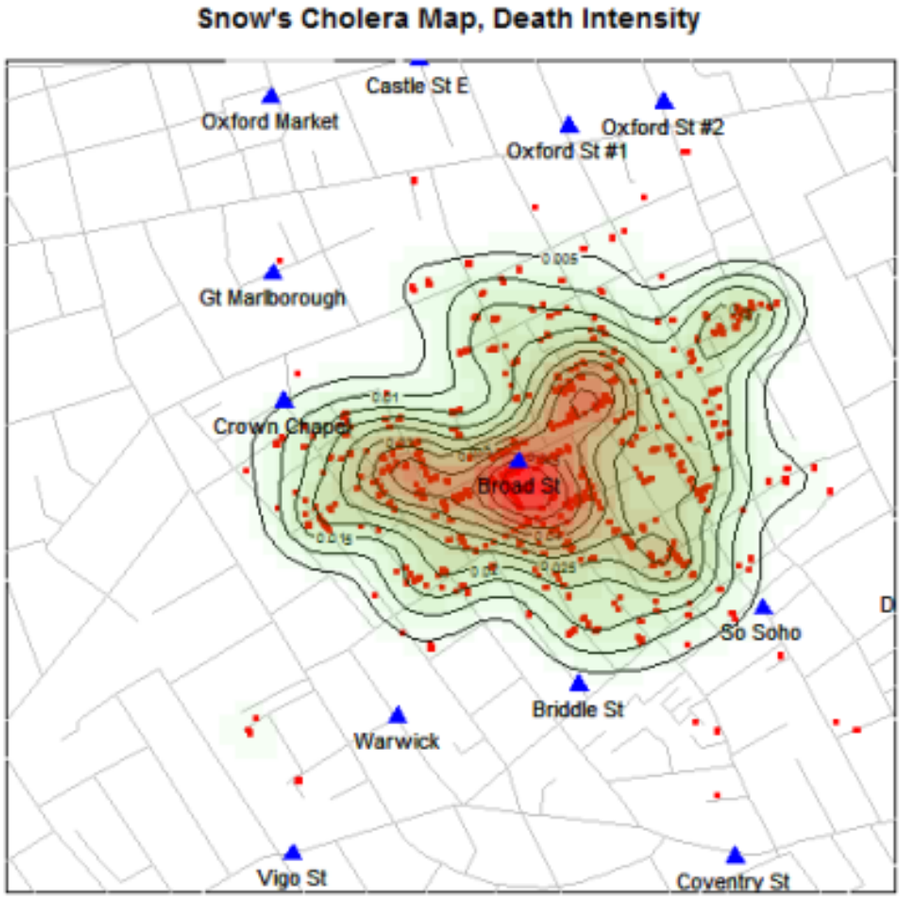
3D

- Three variable numerical
- Avoid unless it is a simple shape



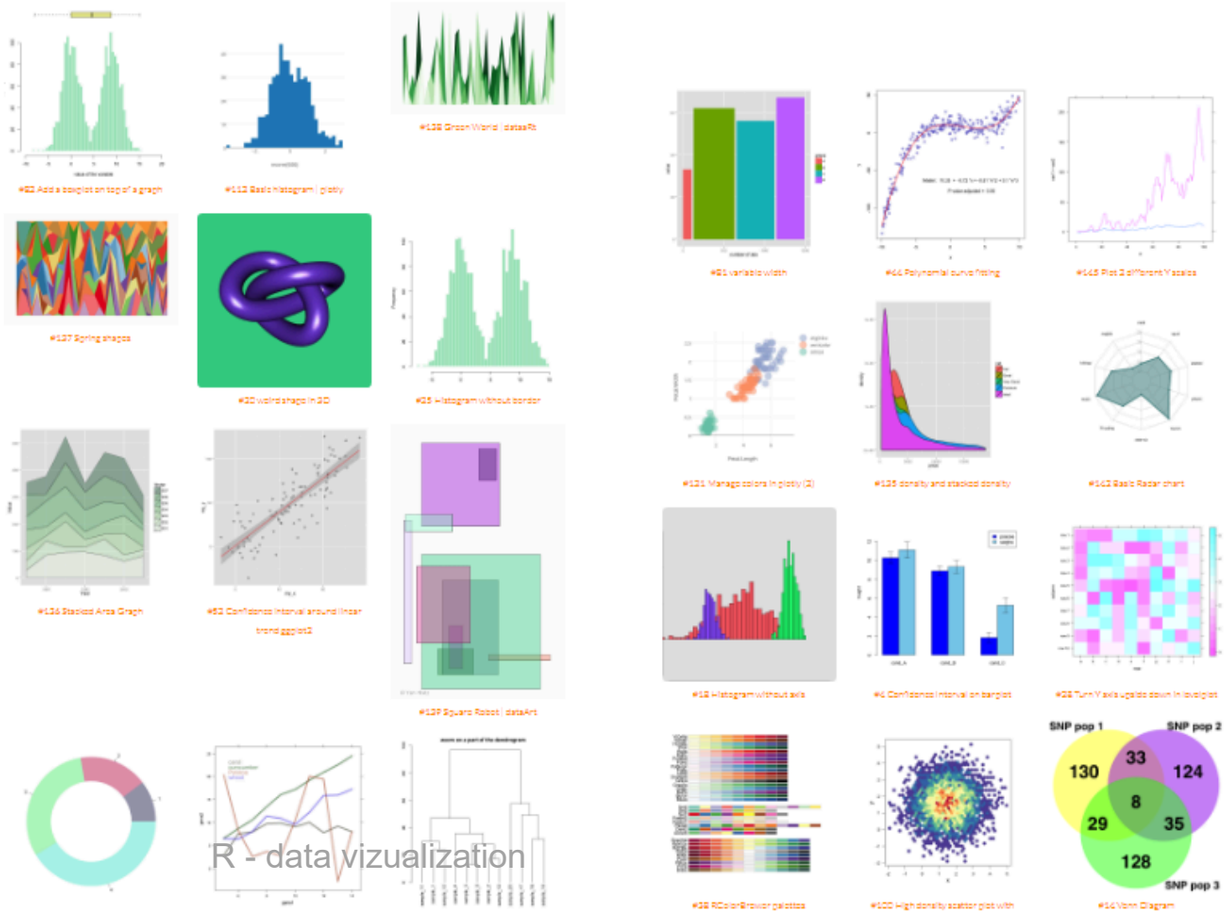
Contours

- Three variable numerical
- Better than 3D



Many…

- Choose as a function of what you want to analyze or the story you want to tell
- https://www.r-graph-gallery.com/all-graphs/



ggplot2


ggplot2:
Build a data
MASTERPIECE
THEMES
GEOMS
aes
fill
color
ggAnimate
plotly
HORST '18

Initialize

Load necessary libraries

```
library("readxl") # Import the data from Excel file
library("dplyr")  # filter and reformat data frames
library("ggplot2") # graphics
library("patchwork") # arrange graphics
```

Read the data

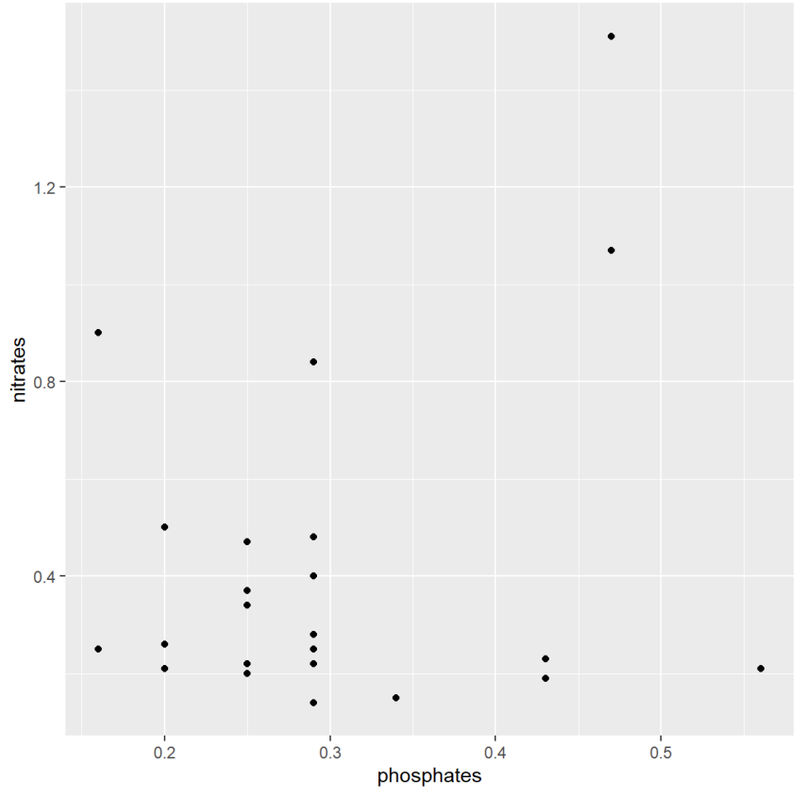
```
samples <- readxl::read_excel("data/CARBOM data.xlsx",
                              sheet = "Samples_boat") %>%
           tidyr::fill(station)
```

sample number	transect	station	date	time	depth	level	latitude	longitude	picoeuks	nanoeuks	phosphates	nitrates	temperature	salinity
10	1	81	2013-11-13	1899-12-31 01:00:00	140	Deep	-27.42	-44.72	3278	1232	0.20	0.26	17.3	35.9
11	1	85	2013-11-13	1899-12-31 13:30:00	110	Deep	-26.80	-45.30	16312	1615	0.29	0.22	21.3	36.5
120	2	96	2013-11-18	1899-12-31 23:50:00	5	Surf	-27.39	-47.82	1150	75	0.43	0.19	23.1	33.5
121	2	96	2013-11-18	1899-12-31 23:50:00	30	Deep	-27.39	-47.82	1737	218	0.43	0.23	22.6	33.7
122	2	96	2013-11-18	1899-12-31 23:50:00	50	Deep	-27.39	-47.82	853	234	0.56	0.21	20.3	35.9
125	2	98	2013-11-18	1899-12-31 05:00:00	5	Surf	-27.59	-47.39	3086	1300	0.29	0.25	23.1	35.7
126	2	98	2013-11-18	1899-12-31 05:00:00	50	Deep	-27.59	-47.39	1217	782	0.25	0.20	23.7	37.2
127	2	98	2013-11-18	1899-12-31 05:00:00	85	Deep	-27.59	-47.39	3420	226	0.25	0.47	22.9	37.0
13	1	86	2013-11-13	1899-12-31 17:00:00	105	Deep	-26.33	-45.41	6366	1007	0.34	0.15	20.9	36.3
140	2	101	2013-11-18	1899-12-31 12:00:00	5	Surf	-27.79	-46.96	500	366	0.29	0.14	23.5	36.5

A simple plot

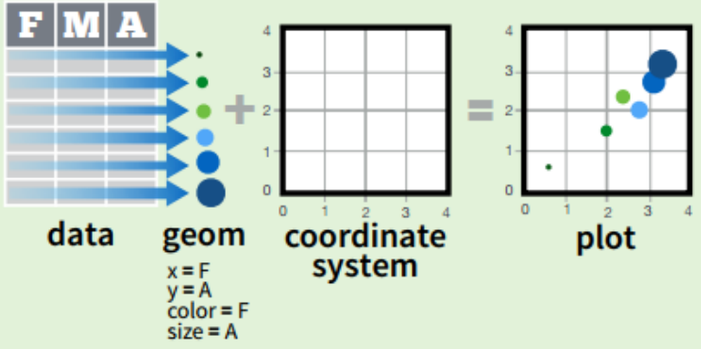
- Choose the data set
- Choose the geometric representation
- Choose the **aesthetics** : x,y, color, shape etc…

```
# All functions are from ggplot2
# package unless specified

ggplot(data=samples) +
    geom_point(mapping = aes(x=phosphates,
                             y=nitrates))
```



The grammar of graphics

To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.




```
ggplot(data=samples) +
 geom_point(mapping = aes(x=phosphates,
                          y=nitrates))
```

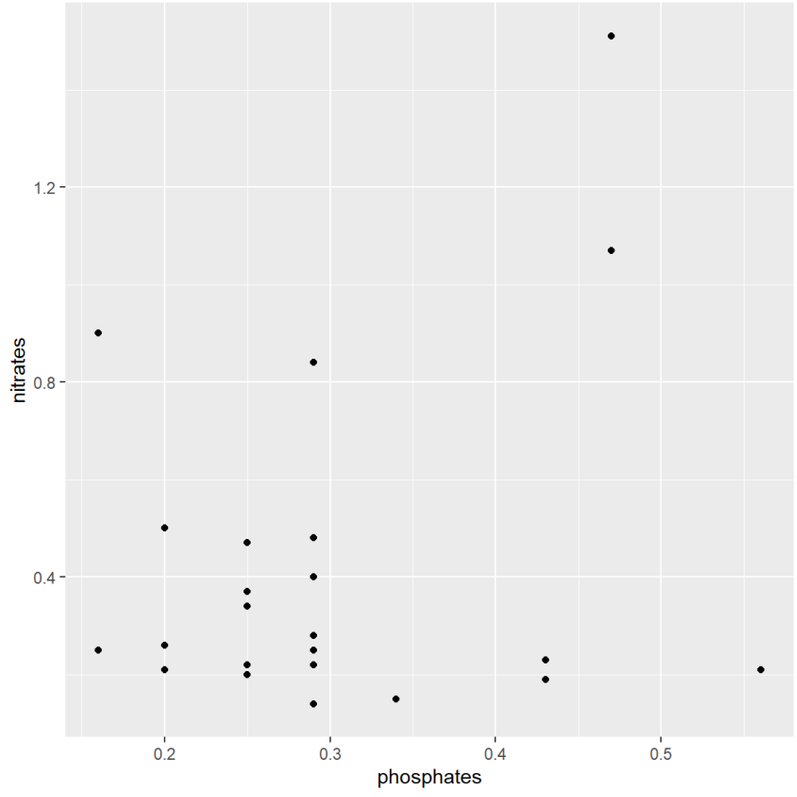
Every graph can be described as a combination of independent building blocks:

- **data**: a data frame: quantitative, categorical; local or data base query
- **aes**thetic mapping of variables into visual properties: size, color, x, y
- **geom**etric objects (“geom”): points, lines, areas, arrows, …
- **coord**inate system (“coord”): Cartesian, log, polar, map

Alternatively

- Move mapping into ggplot function

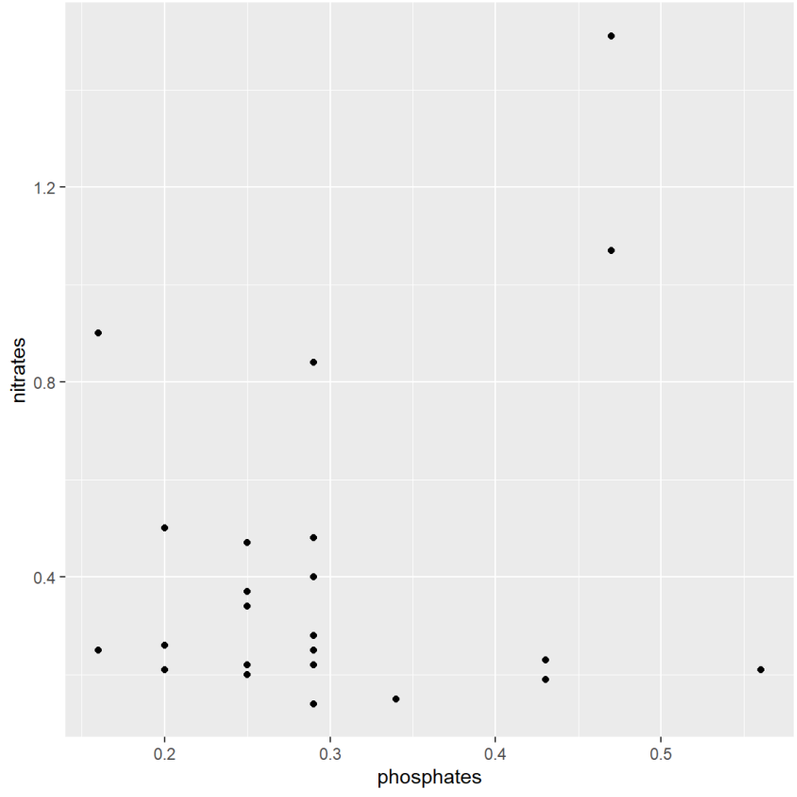
```
ggplot(data=samples,
       mapping = aes(x=phosphates,
                     y=nitrates)) +
 geom_point()
```



Alternatively

- Remove function arguments

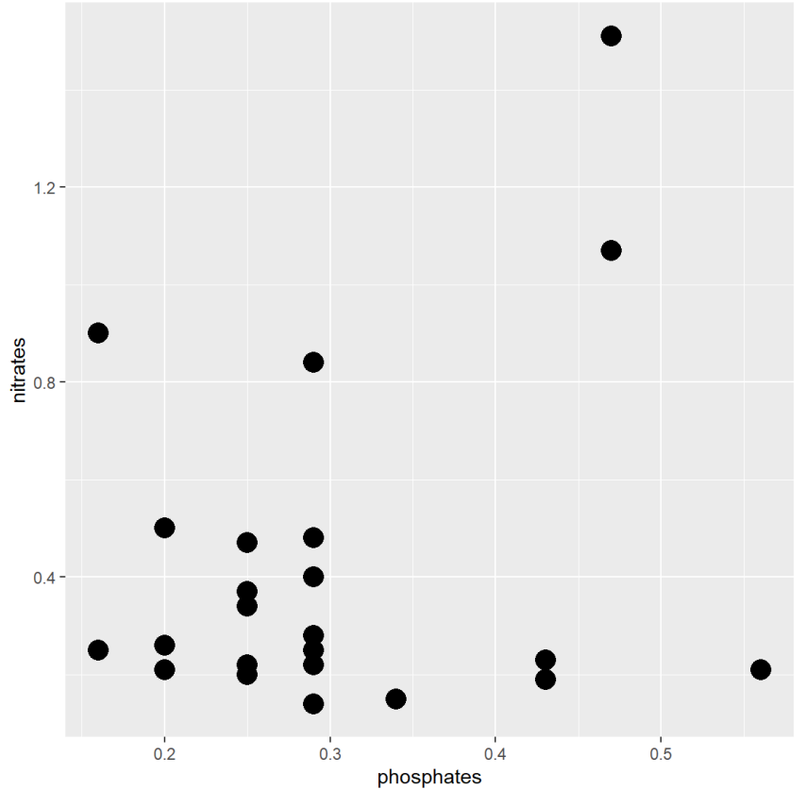
```
ggplot(samples,
       aes(x=phosphates,
           y=nitrates)) +
 geom_point()
```



Makes dots bigger

- Add: **size=5** outside of the aesthetics function

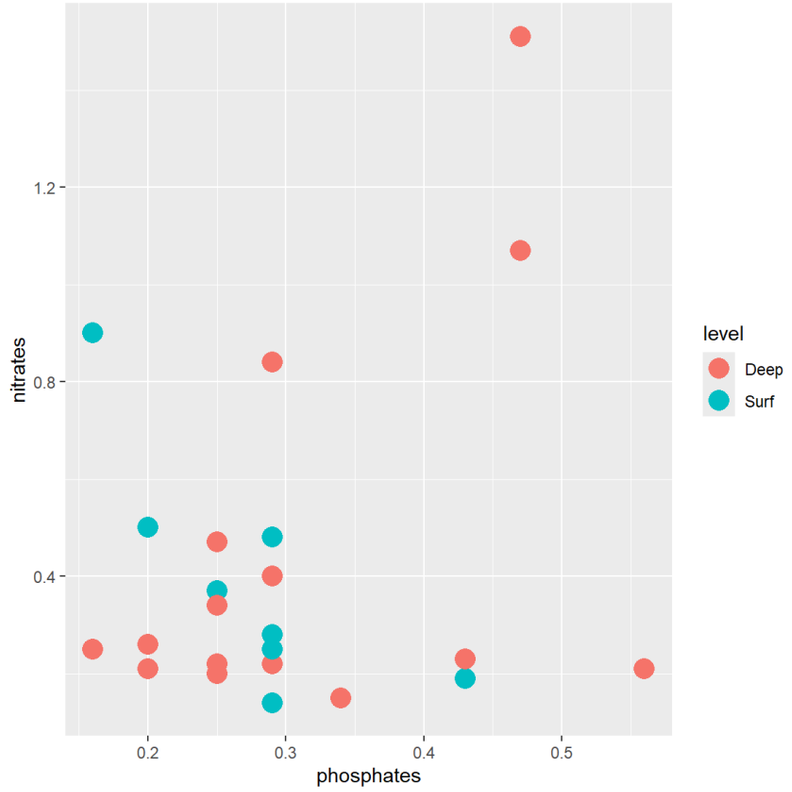
```
ggplot(samples,
       aes(x=phosphates,
           y=nitrates)) +
 geom_point(size=5)
```



Color according to depth level (discrete)

- The mapping aesthetics must be an argument of the aes function
- geom_point(**color=level**, size=5) will generate an error…

```
ggplot(samples,
       aes(x=phosphates,
           y=nitrates,
           color=level)) +
 geom_point(size=5)
```



Color according to depth level (discrete)

- The mapping aesthetics must be an argument of the aes function
- geom_point(**color=level**, size=5) will generate an error…

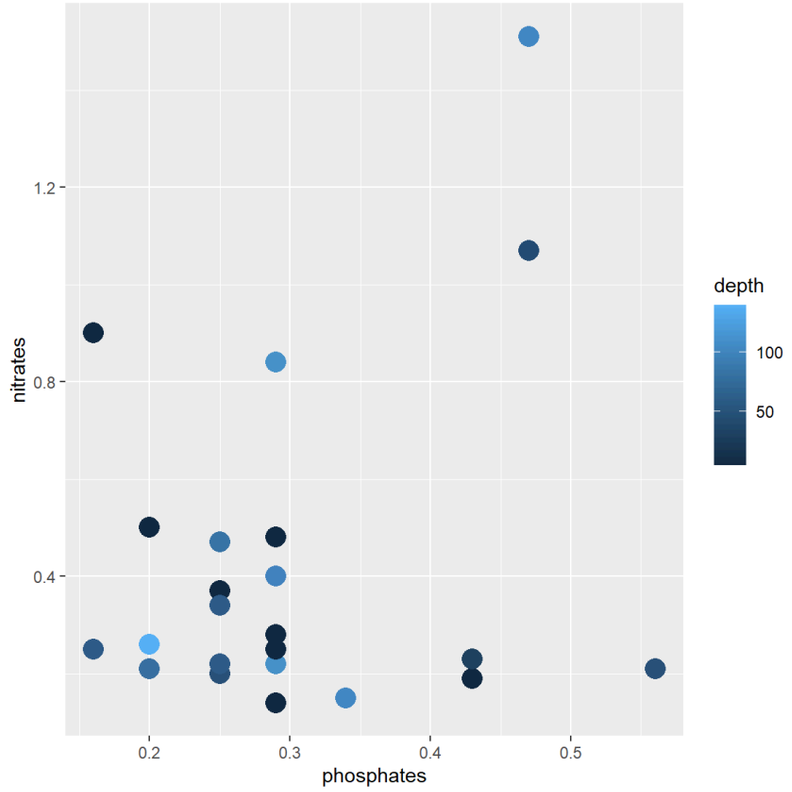
```
ggplot(samples,
       aes(x=phosphates,
           y=nitrates)) +
 geom_point(color=level, size=5)
```

```
Error in eval(expr, envir, enclos): object 'level'
not found
```

Color according to depth (continuous)

- The mapping aesthetics must be an argument of the aes function
- Add: **color=depth**

```
ggplot(samples,
       aes(x=phosphates,
           y=nitrates,
           color=depth)) +
 geom_point(size=5)
```



Symbol according to transect (continuous)

- Add: **shape=transect**

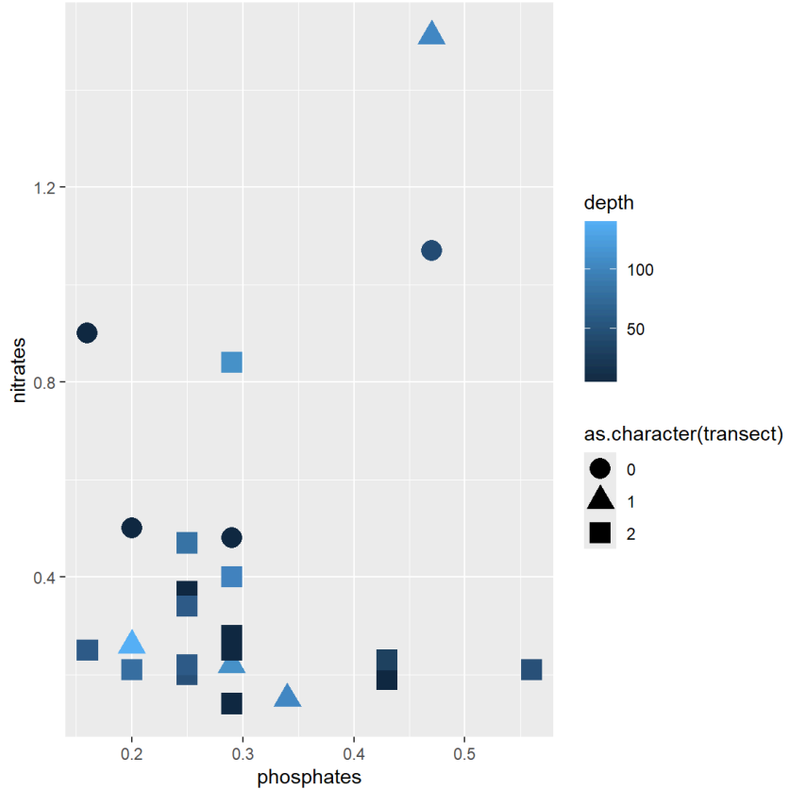
```
ggplot(samples,
       aes(x=phosphates,
           y=nitrates,
           color=depth,
           shape=transect)) +
 geom_point(size=5)
```

```
Error in `geom_point()`:
! Problem while computing aesthetics.
ℹ Error occurred in the 1st layer.
Caused by error in `scale_f()`:
! A continuous variable cannot be mapped to the
shape aesthetic.
ℹ Choose a different aesthetic or use
`scale_shape_binned()`.
```

Symbol according to transect (continuous)

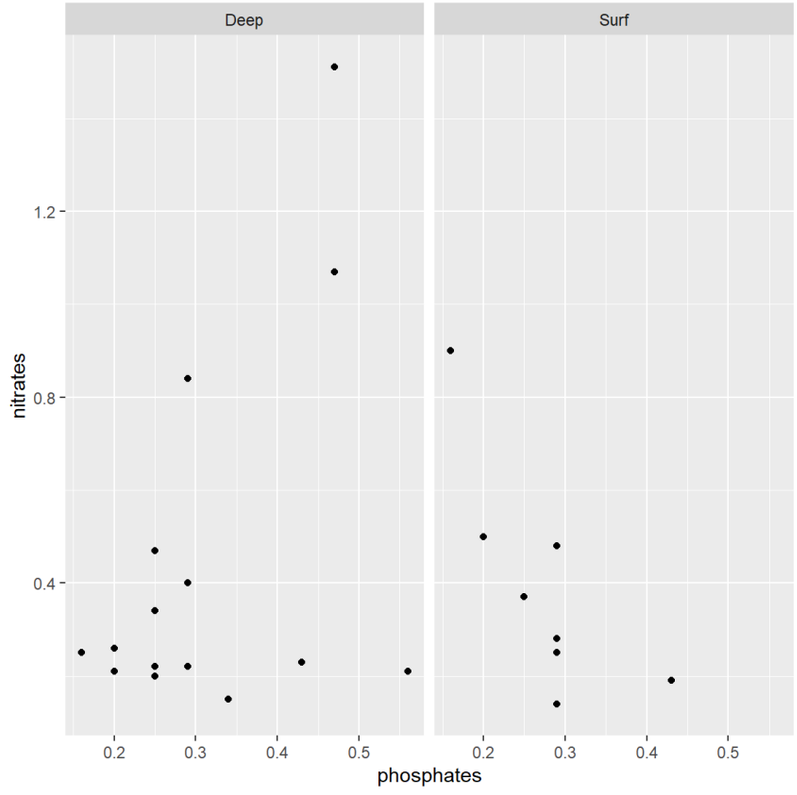
- Add: **shape=as.character(transect)**

```
ggplot(samples,
       aes(x=phosphates,
           y=nitrates,
           color=depth,
           shape=as.character(transect))) +
 geom_point(size=5)
```



Panels depending on one variable

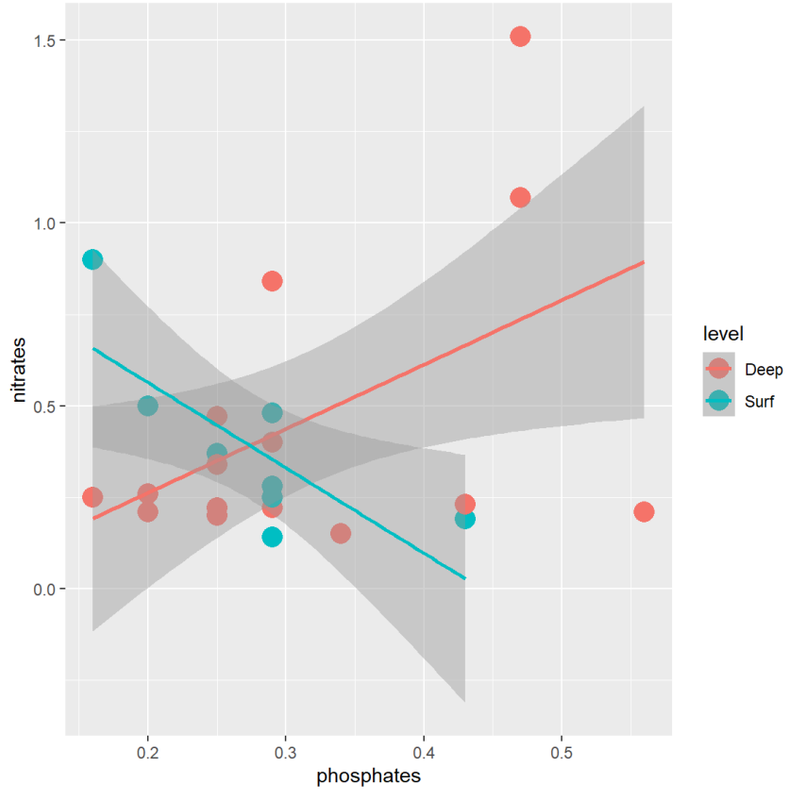
```
ggplot(samples,
       aes(x=phosphates,
           y=nitrates)) +
 geom_point() +
 facet_wrap(~ level)
```



Adding a regression line

- Add: **geom_smooth()**
- You can choose the type of smoothing “lm” is for linear model

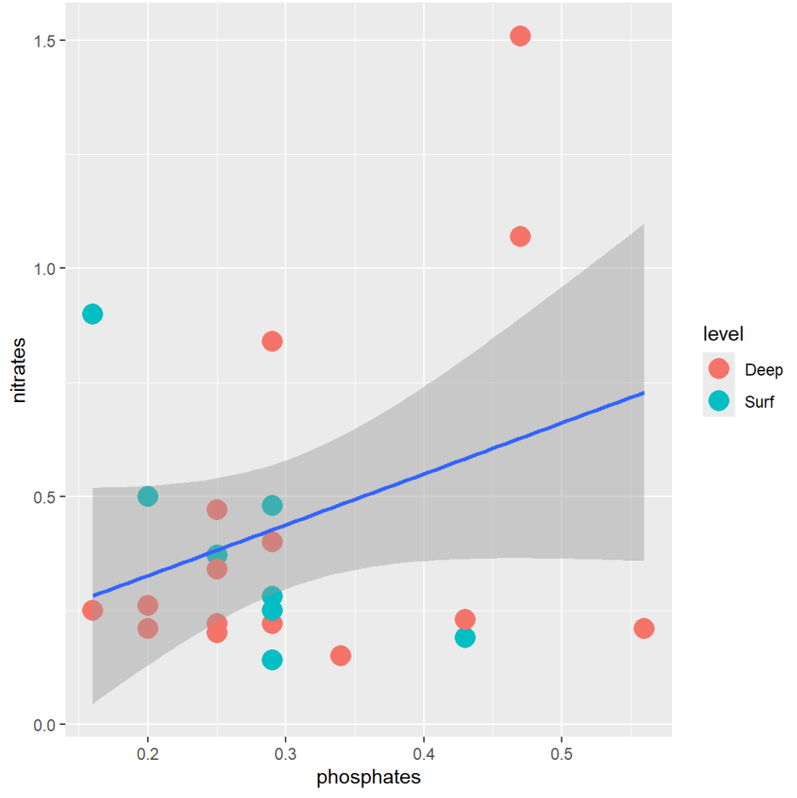
```
ggplot(samples,
       aes(x=phosphates,
           y=nitrates,
           color=level)) +
 geom_point(size=5) +
 geom_smooth(mapping = aes(x=phosphates,
                           y=nitrates),
             method="lm")
```



Adding a regression line

- If the mapping is in the ggplot function is for all the geom….

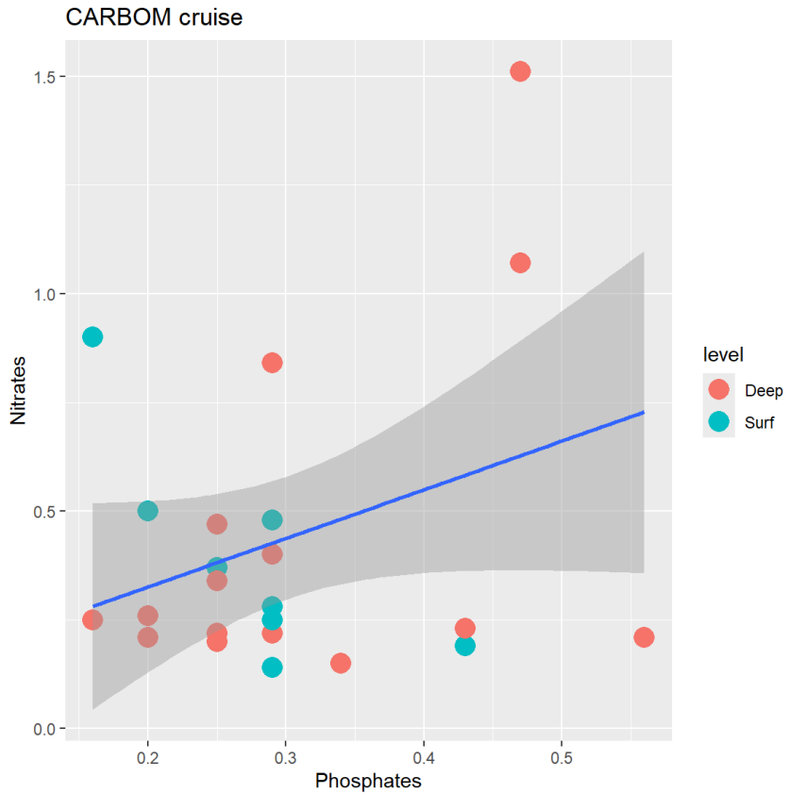
```
ggplot(samples,
       aes(x=phosphates,
           y=nitrates)) +
 geom_point(aes(color=level),
            size=5) +
 geom_smooth(mapping = aes(x=phosphates,
                           y=nitrates),
             method="lm")
```



Finalizing the graph

- Adding labels and legends

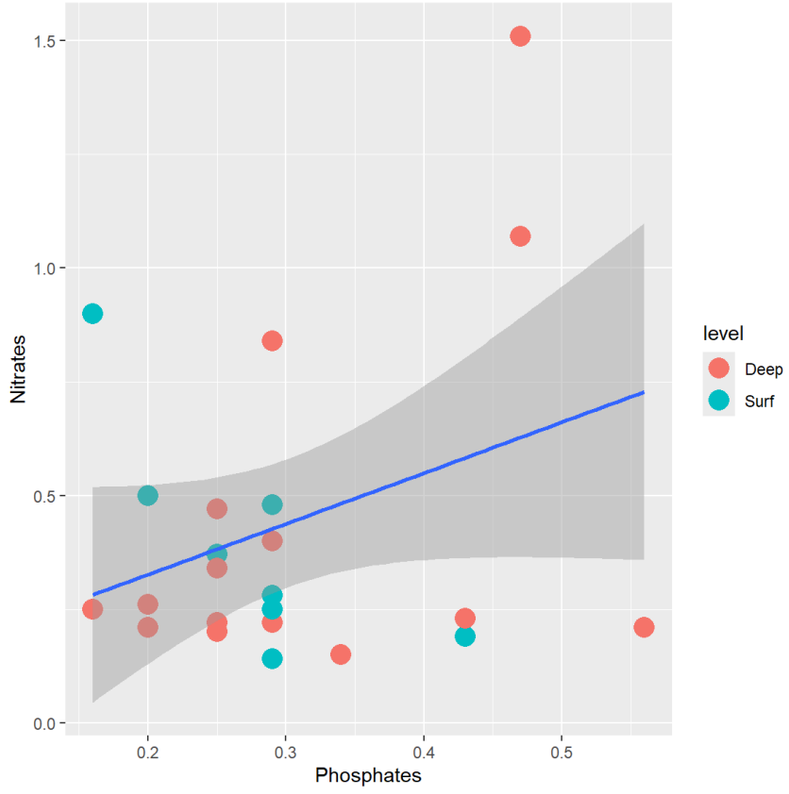
```
ggplot(samples) +
 geom_point(mapping = aes(x=phosphates,
                          y=nitrates,
                          color=level),
            size=5) +
 geom_smooth(mapping = aes(x=phosphates,
                           y=nitrates),
             method="lm") +
 xlab("Phosphates") +
 ylab("Nitrates") +
 ggtitle("CARBOM cruise")
```



Multigraphs (patchwork package)

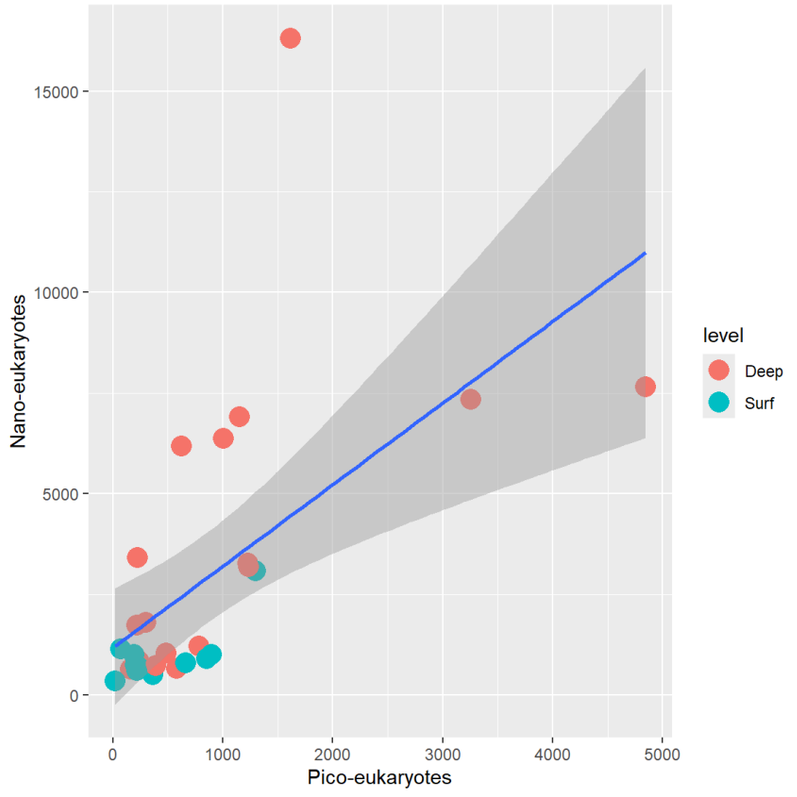
First graph

```
g1 <- ggplot(samples) +
 geom_point(mapping = aes(x=phosphates,
                          y=nitrates,color=
                            level), size=5)
 geom_smooth(mapping = aes(x=phosphates,
                           y=nitrates),
             method="lm") +
 xlab("Phosphates") +
 ylab("Nitrates")

g1
```



Second graph

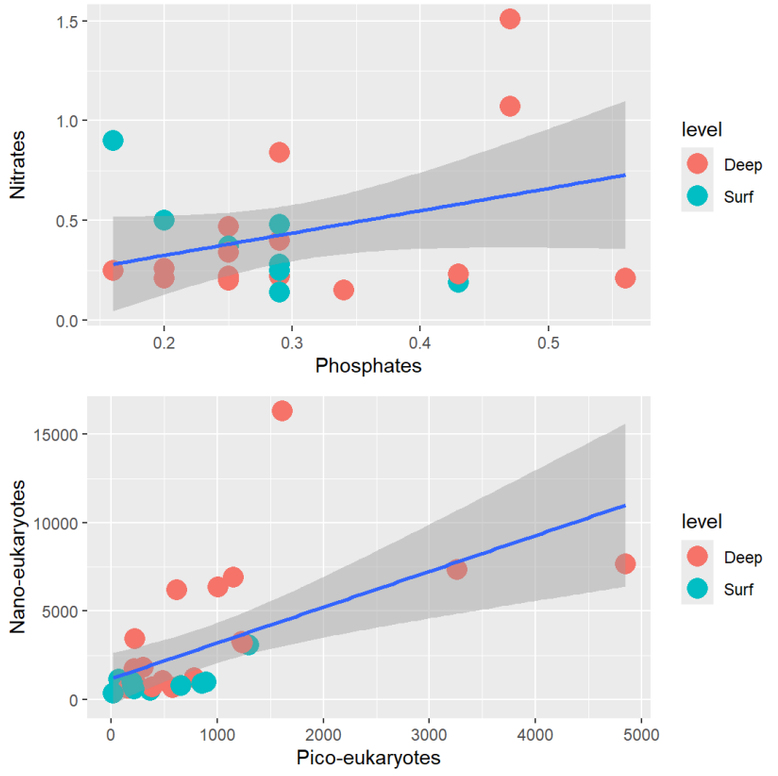
```
g2<- ggplot(samples) +
 geom_point(mapping = aes(x=nanoeuks,
                          y=picoeuks,
                          color=level),
            size=5) +
 geom_smooth(mapping = aes(nanoeuks,
                           y=picoeuks),
             method="lm") +
 xlab("Pico-eukaryotes") +
 ylab("Nano-eukaryotes")

g2
```



Package patchwork

- Other packages:
  - `gridExtra`
  - `cowplot`

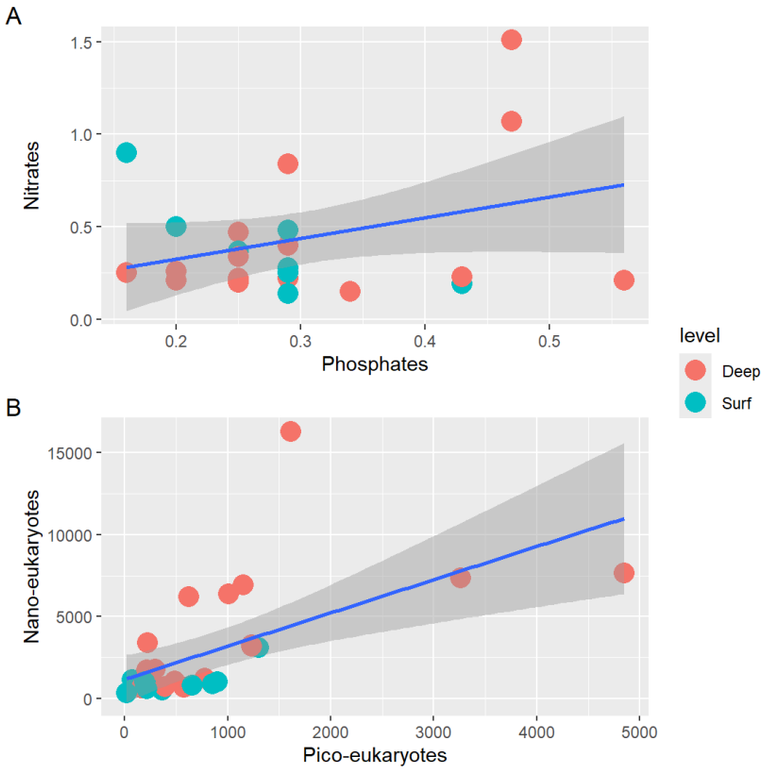
```
library(patchwork)
(g1 / g2)
```



Package patchwork

- Adding annotation
- Collecting legends

```
g1 / g2 +
  plot_annotation(tag_levels = 'A') +
  plot_layout(guides = 'collect')
```



Package esquisse

esquisse 2.0.1.9000 Reference Articles Changelog Search for

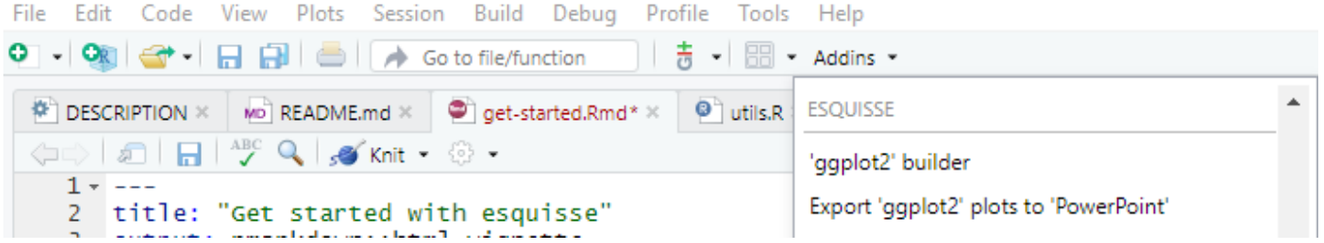
Get started with esquisse

Source: vignettes/get-started.Rmd

```
library(esquisse)
```

Launch the addin

In RStudio, you can use the *Addins* menu :



Or in the R console :

```
esquisser()
```

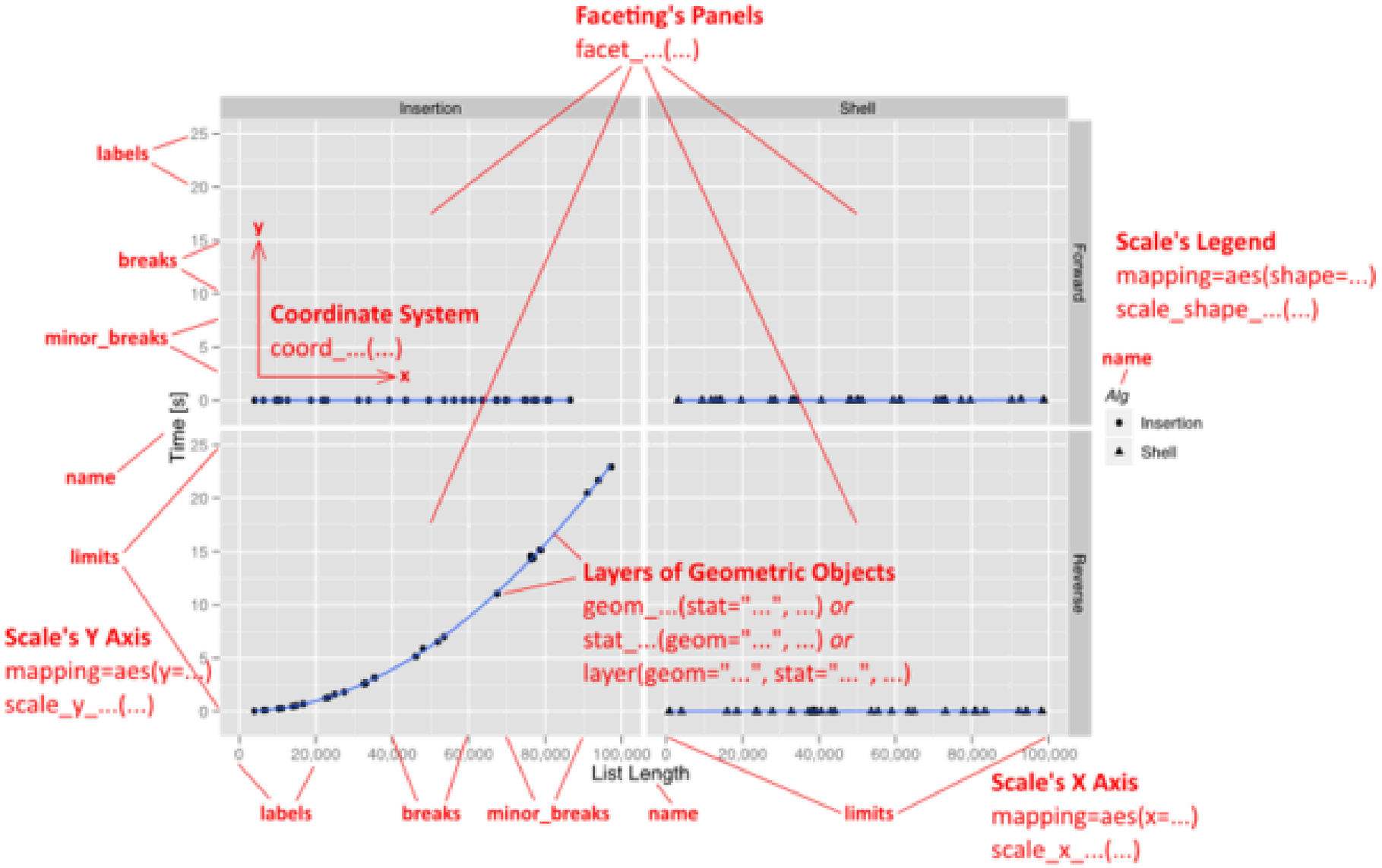
To use a `data.frame` by default, if using the *Addins* menu highlight with the cursor a

On this page

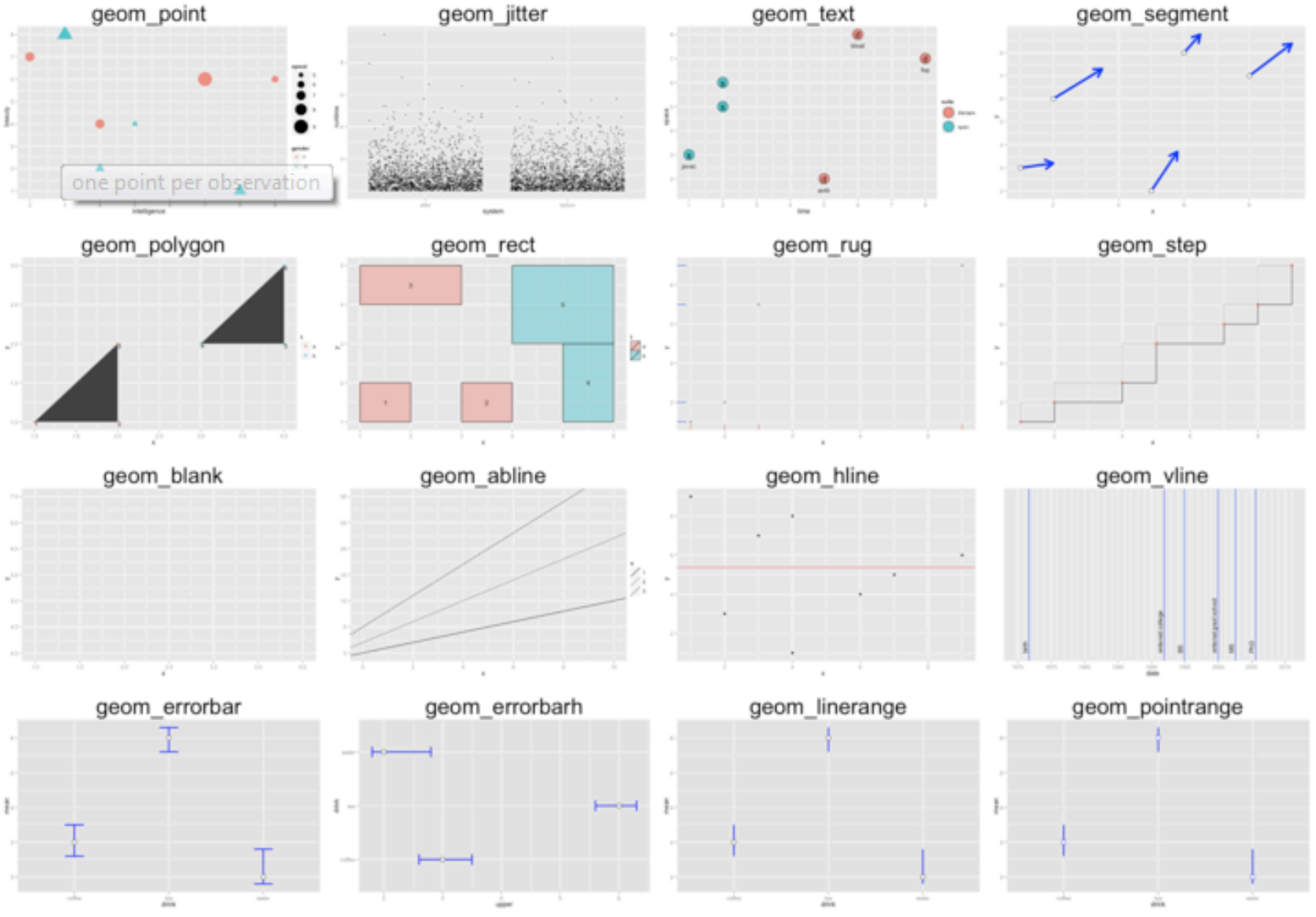
- Launch the addin
- Import data into {esquisse}
- Create a plot
- Controls
- Export
- Addin options

ggplot2 syntax

Anatomy of a plot



Geometries

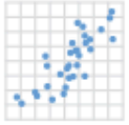

geom_point
one point per observation
geom_jitter
geom_text
geom_segment
geom_polygon
geom_rect
geom_rug
geom_step
geom_blank
geom_abline
geom_hline
geom_vline
geom_errorbar
geom_errorbarh
geom_linerange
geom_pointrange

Continuous x and y

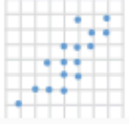
continuous x , continuous y

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

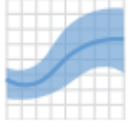
e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE**)** x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust



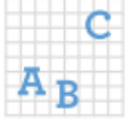
e + geom_jitter(height = 2, width = 2**)**
x, y, alpha, color, fill, shape, size



e + geom_point(), x, y, alpha, color, fill, shape, size, stroke



e + geom_smooth(method = lm**)**, x, y, alpha, color, fill, group, linetype, size, weight

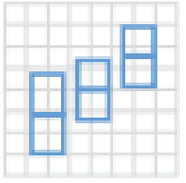


e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE**)**, x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

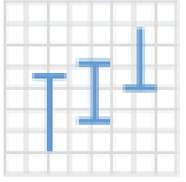
Plotting error

visualizing error

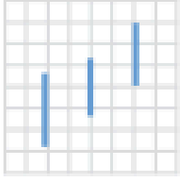
```
df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))
```



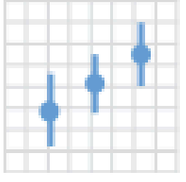
j + geom_crossbar(fatten = 2**)**
x, y, ymax, ymin, alpha, color, fill, group, linetype, size



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



j + geom_linerange()
x, ymin, ymax, alpha, color, group, linetype, size

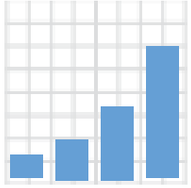


j + geom_pointrange()
x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

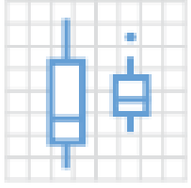
Discrete x - Continuous y

discrete x , continuous y

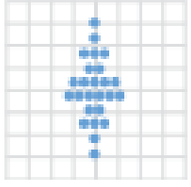
f <- ggplot(mpg, aes(class, hwy))



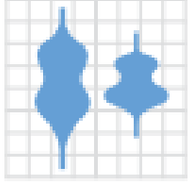
f + geom_col(), x, y, alpha, color, fill, group, linetype, size



f + geom_boxplot(), x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight



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



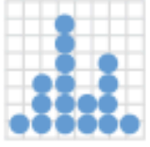
f + geom_violin(scale = "area"**)**, x, y, alpha, color, fill, group, linetype, size, weight

Continuous x

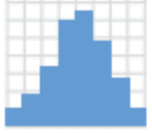
ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

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



c + geom_dotplot()
x, y, alpha, color, fill



c + geom_histogram(binwidth = 5**)** x, y, alpha, color, fill, linetype, size, weight

3D

THREE VARIABLES

seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))l <- ggplot(seals, aes(long, lat))

l + geom_contour(aes(z = z))
x, y, z, alpha, colour, group, linetype, size, weight

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

l + geom_tile(aes(fill = z)), x, y, alpha, color, fill, linetype, size, width

Modifying axis and scales

GENERAL PURPOSE SCALES

Use with most aesthetics

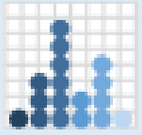
scale_*_continuous() - map cont' values to visual ones
scale_*_discrete() - map discrete values to visual ones
scale_*_identity() - use data values **as** visual ones
scale_*_manual(values = c()**)** - map discrete values to manually chosen visual ones
scale_*_date(date_labels = "%m/%d"), date_breaks = "2 weeks"**)** - treat data values as dates.
scale_*_datetime() - treat data x values as date times. Use same arguments as scale_x_date(). See ?strptime for label formats.

X & Y LOCATION SCALES

Use with x or y aesthetics (x shown here)

scale_x_log10() - Plot x on log10 scale
scale_x_reverse() - Reverse direction of x axis
scale_x_sqrt() - Plot x on square root scale

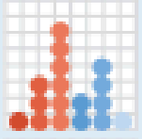
COLOR AND FILL SCALES (CONTINUOUS)



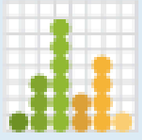
o <- c + geom_dotplot(aes(fill = ..x..))

o + scale_fill_distiller(palette = "Blues"**)**

o + scale_fill_gradient(low="red", high="yellow"**)**

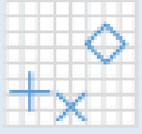


o + scale_fill_gradient2(low="red", high="blue", mid = "white", midpoint = 25**)**

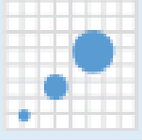


o + scale_fill_gradientn(colours=topo.colors(6)**)**
Also: rainbow(), heat.colors(), terrain.colors(), cm.colors(), RColorBrewer::brewer.pal()

SHAPE AND SIZE SCALES



p <- e + geom_point(aes(shape = fl, size = cyl))
p + scale_shape() + scale_size()
p + scale_shape_manual(values = c(3:7)**)**



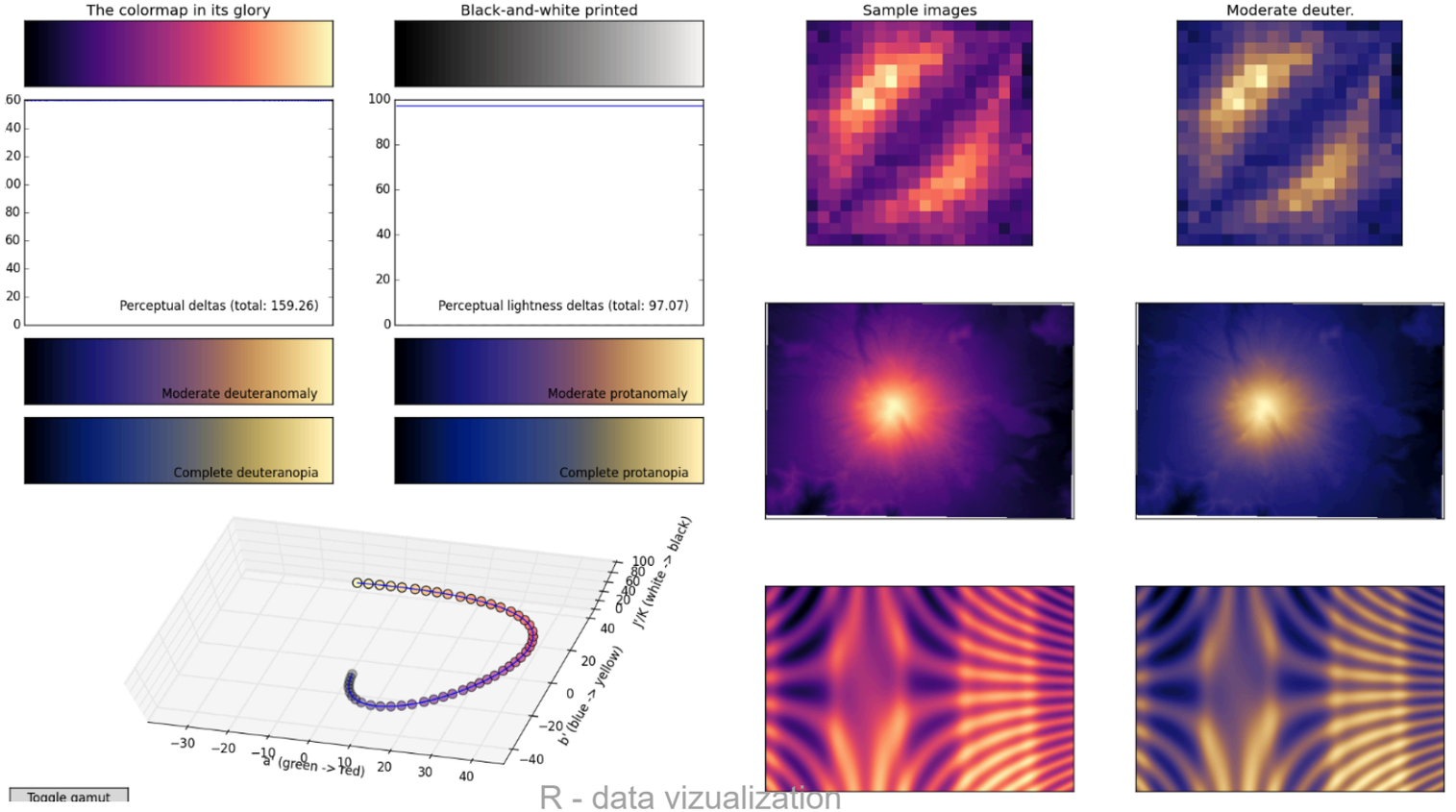
p + scale_radius(range = c(1,6)**)**
p + scale_size_area(max_size = 6**)**

Palettes

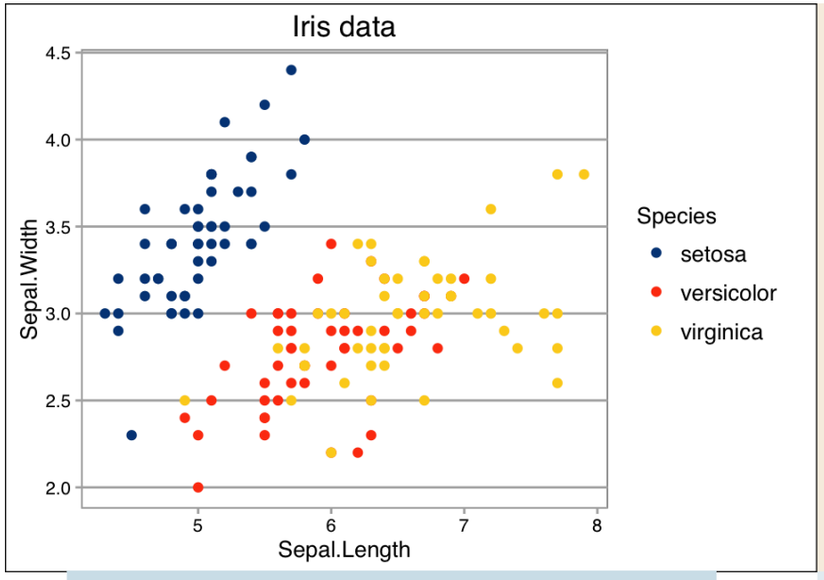
- Package tmaptools
  - Function : `palette_explorer()`
- Package paletteer
  - More than 1000 palettes

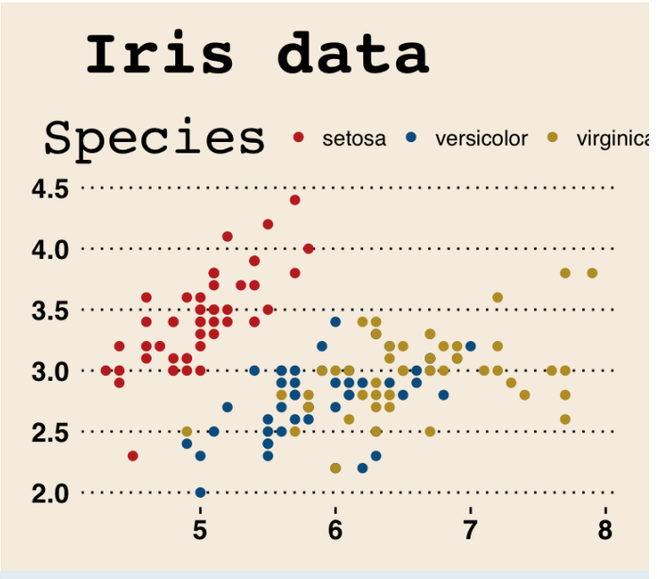
Palettes

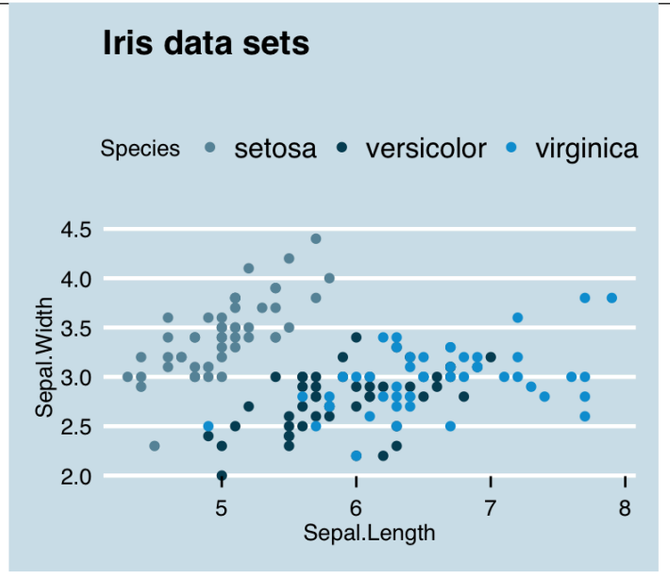
- Use color blind friendly palettes
  - viridis (e.g. `scale_colour_viridis_c()`)

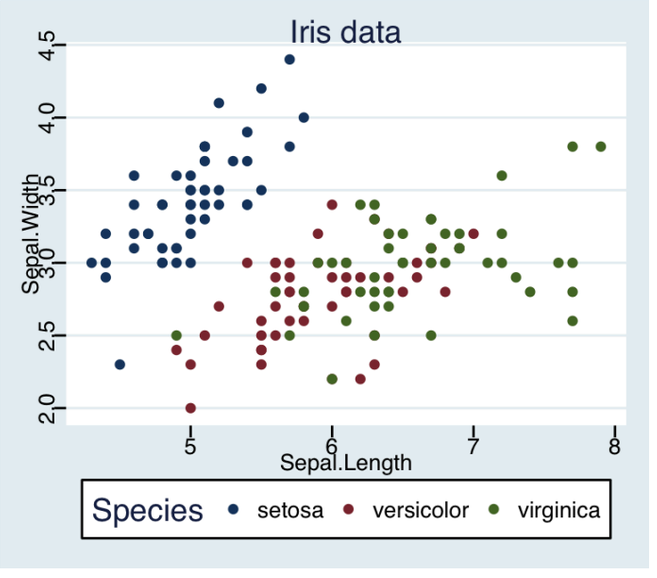


Themes


Iris data
Sepal.Width
Sepal.Length
Species
setosa
versicolor
virginica


Iris data
Species
setosa
versicolor
virginica


Iris data sets
Species
setosa
versicolor
virginica
Sepal.Width
Sepal.Length


Iris data
Sepal.Width
Sepal.Length
Species
setosa
versicolor
virginica

Extensions

ggplot2 extensions - gallery

- Add Your Extension!
- exts.ggplot2.tidyverse.org

- Navbar Link

142 registered extensions available to explore

▼ Github stars

- Name
- Author
- Github stars

Github stars
Sort
search name, author, descrip Text Filter
▼

-
- Alex Zanidean (1)
- AllanCameron (1)
- almeidaxan (1)
- aphalo (4)
- arcresu (1)

Let’s do a graph

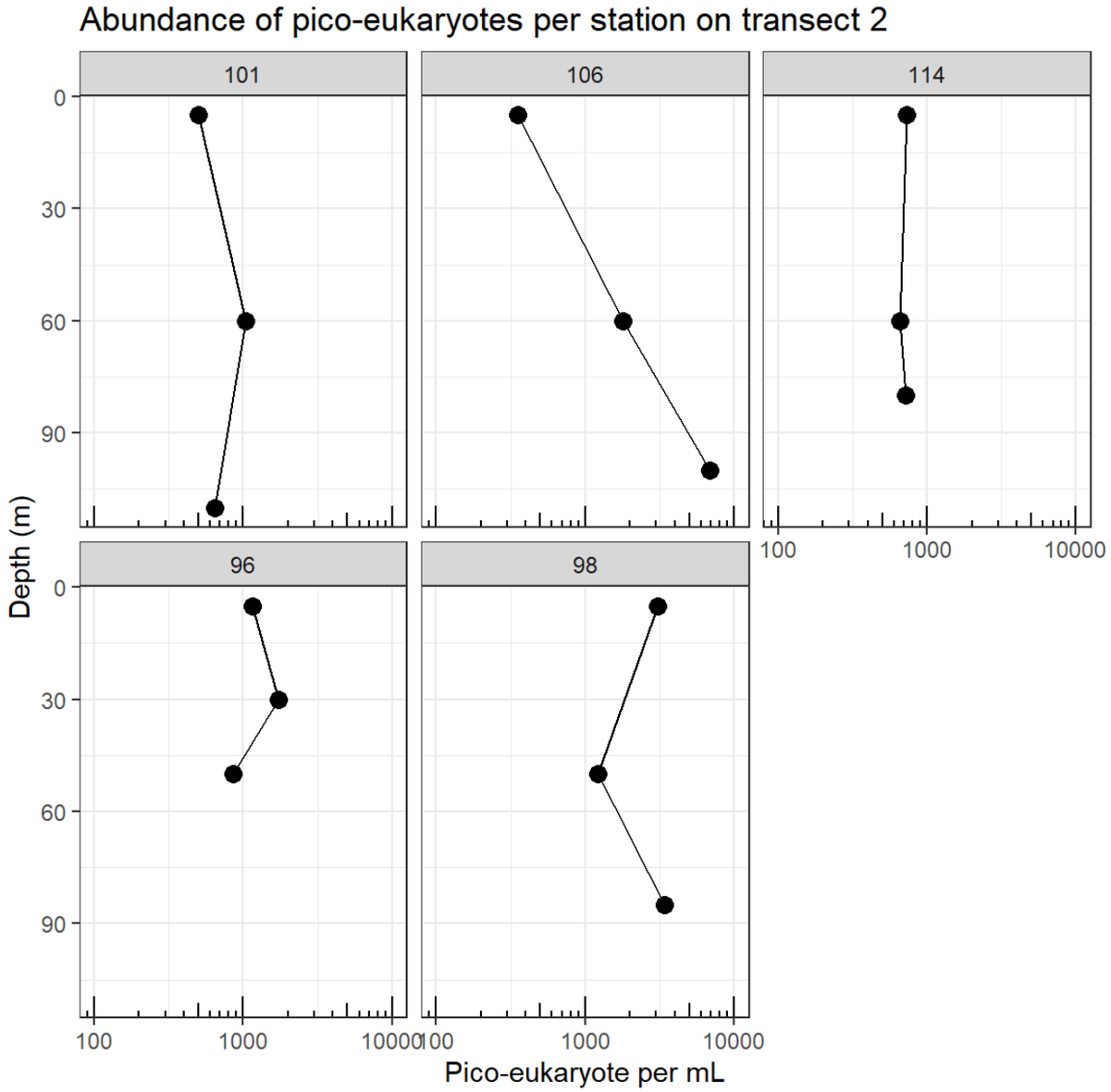
Your mission

Reproduce graph on right

- Only transect 2
- One panel per station
- Increasing depth
- Log scale for x
- White background

Instructions

- Work by group of 2 (1 expert, 1 less expert)
- We will correct later in the week



Recap

- Conceptualize your graph before coding
- Decide what element is fixed and what varies
- It takes time to get what you want…
- Exploratory vs. final

Next time: Markdown and Quarto