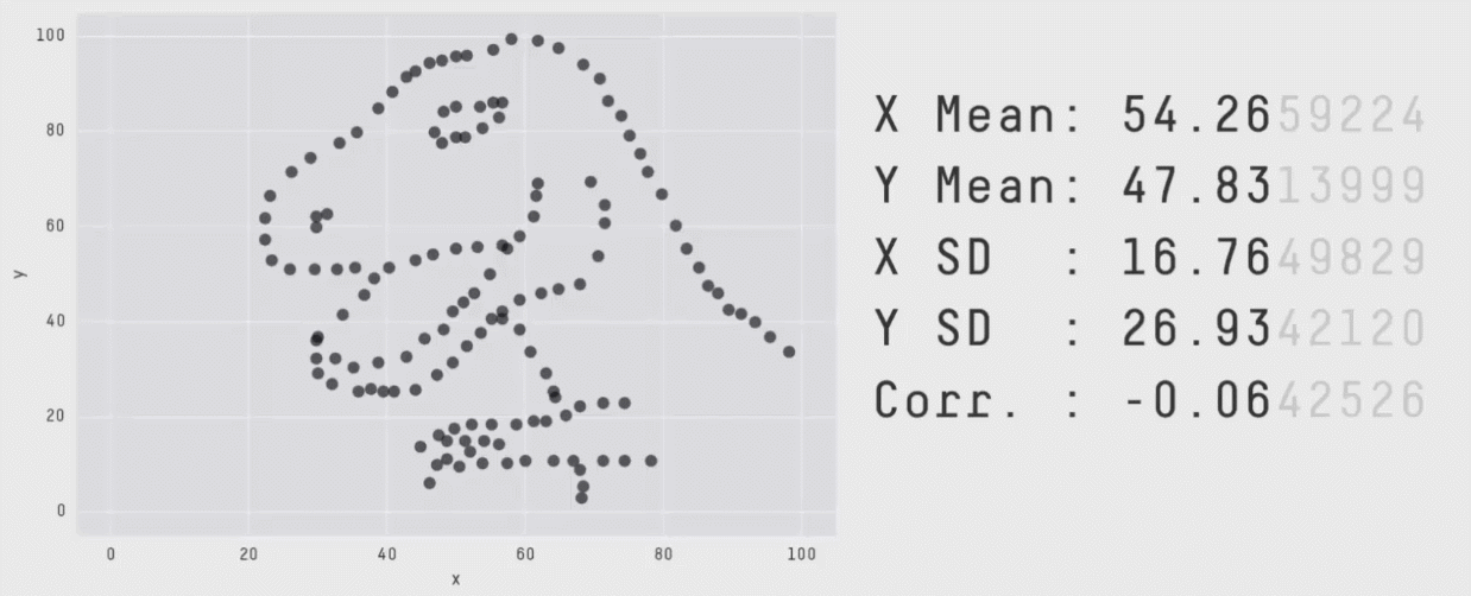
# Data visualisation with ggplot2

Francisco Rodríguez-Sánchez

https://frodriguezsanchez.net

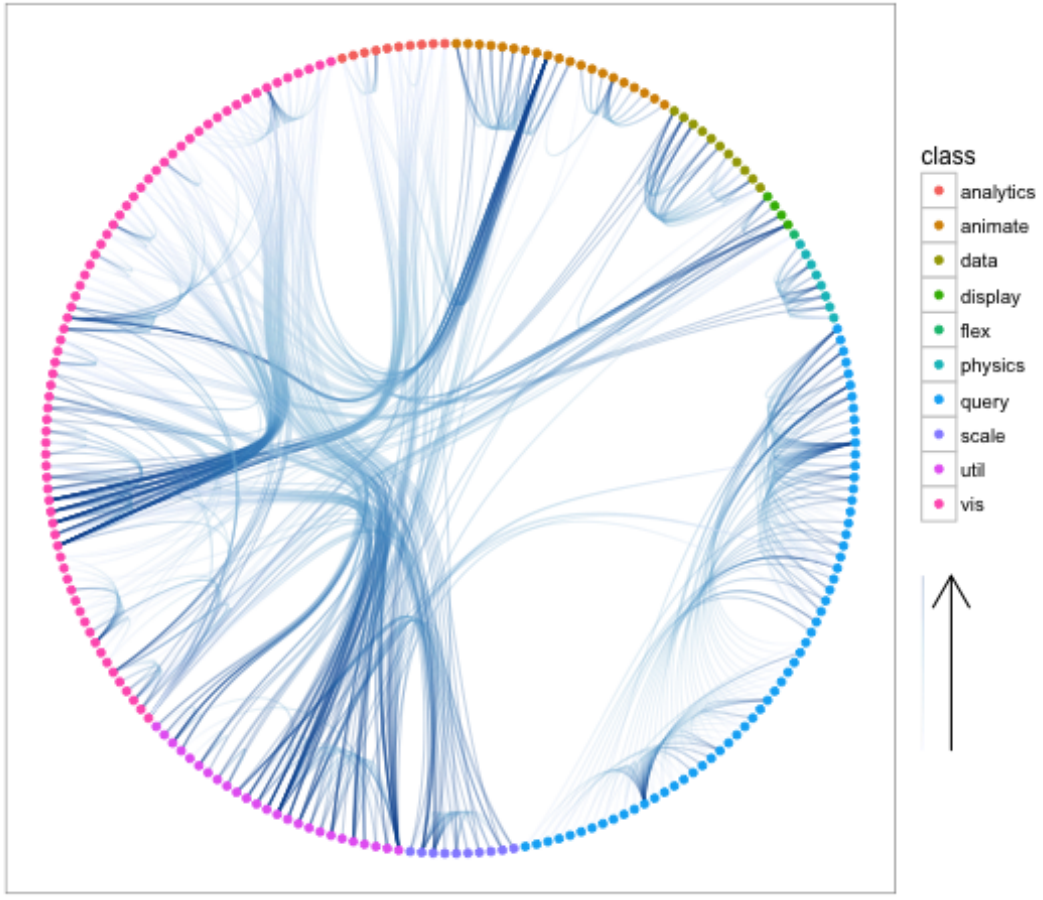
@frod_san

# Always plot data!




https://github.com/stephlocke/datasauRus

# Made with ggplot

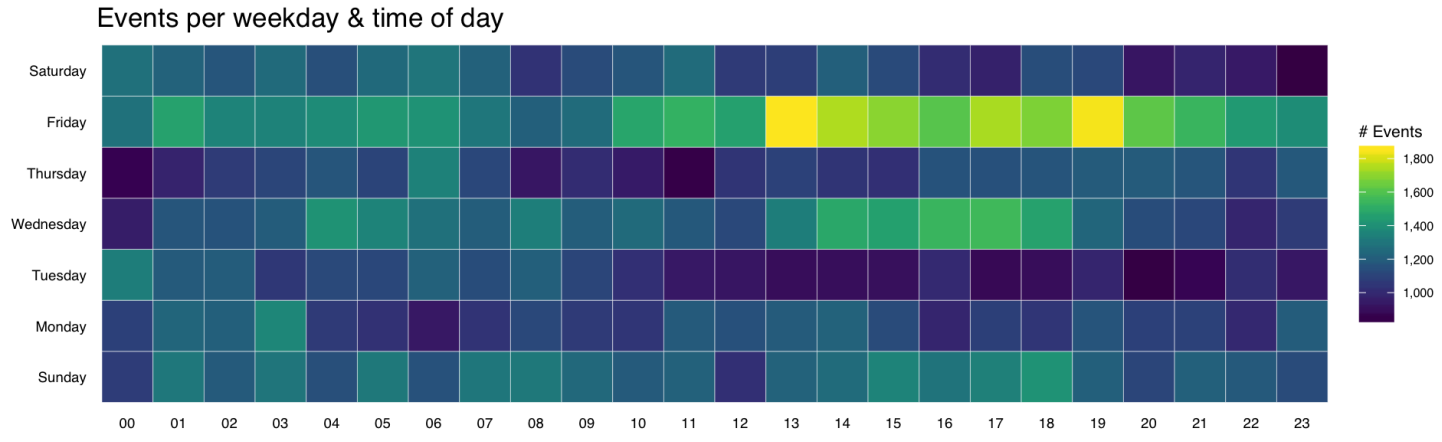



https://github.com/thomasp85/ggraph

# Made with ggplot

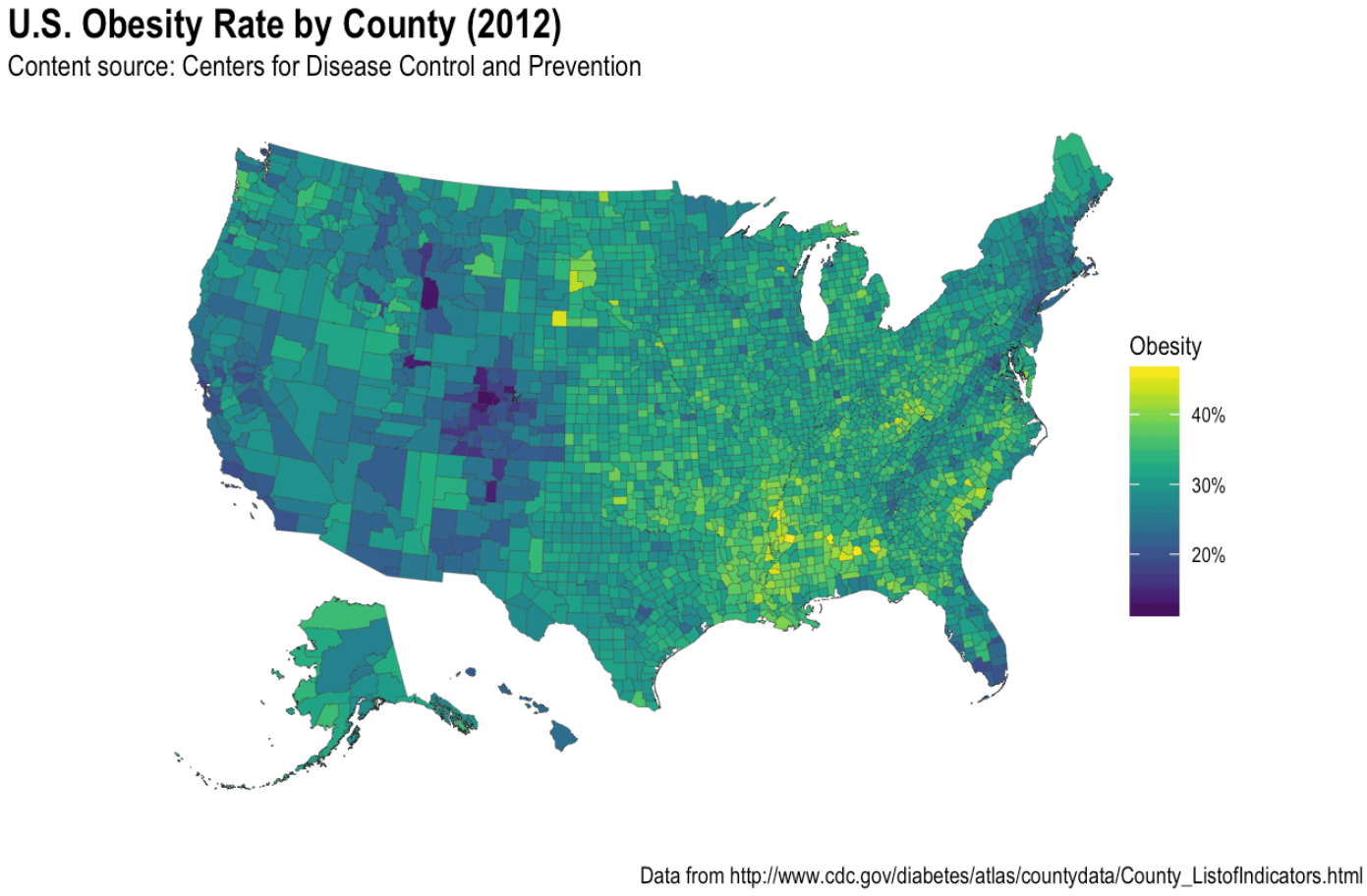

http://spatial.ly/2012/02/great-maps-ggplot2/

# Made with ggplot



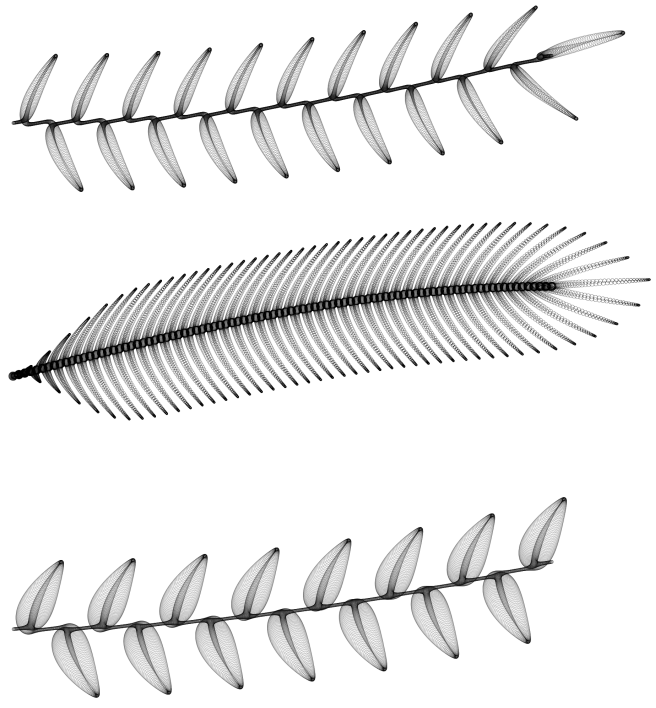

https://rud.is/b/2016/02/14/making-faceted-heatmaps-with-ggplot2/

# Made with ggplot




https://rud.is/b/2016/03/29/easier-composite-u-s-choropleths-with-albersusa/

# Made with ggplot



https://github.com/marcusvolz/mathart

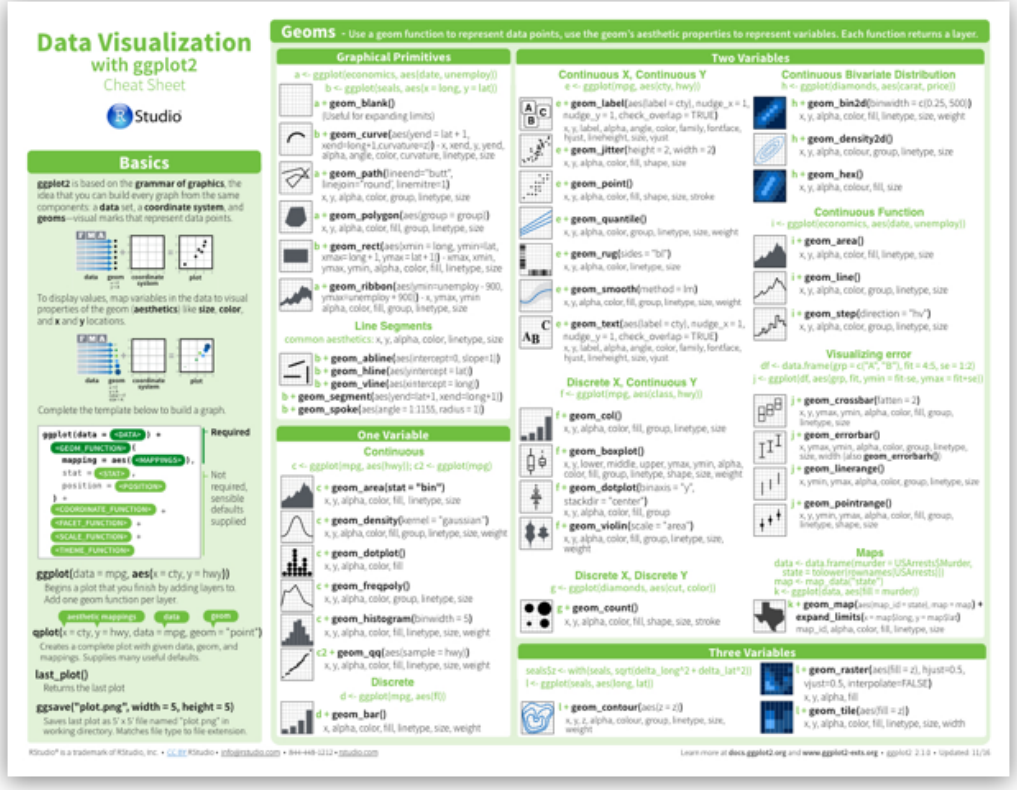
# Why ggplot

- Extremely powerful and flexible
- Consistent (grammar of graphics)
- Very powerful user base and active development

# Very good documentation and tutorials

- Official ggplot2 documentation
- ggplot2 book
- Data visualisation chapter in R for Data Science
- R graphics cookbook and Cookbook for R
- Data visualization: a practical introduction (K. Healy)
- Fundamentals of data visualization (C. Wilke)

# Cheatsheet

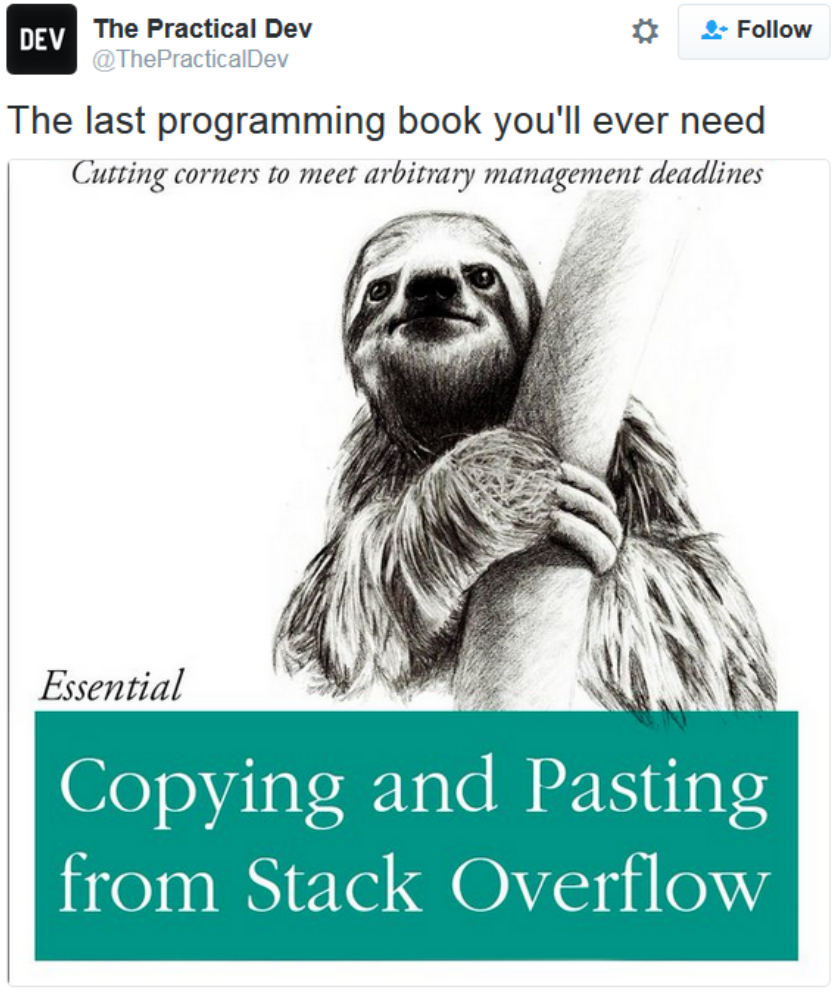



https://www.rstudio.com/resources/cheatsheets/

# Repos of figures + code

- From Data to Viz
- The R graph gallery
- R graphics cookbook
- Cookbook for R: Graphs
- Graphical data analysis with R
- R graph catalog

# Find answers in Stack Overflow, Rstudio Community, R4DS...



# Building a ggplot figure

# Our example dataset: paper planes flying experiment

```
library(paperplanes)
data(paperplanes)
head(paperplanes)
```

```
# A tibble: 6 × 8
     id hour    person   gender   age plane       paper distance
  <int> <fct>   <chr>    <fct>  <dbl> <chr>       <int>    <dbl>
1     1 [17,18) Roland   male      30 Standard80     80      7.8
2     2 [17,18) Astrid   female    30 Concorde120   120      2.7
3     3 [17,18) Roland   male      30 Standard120   120      9.2
4     4 [17,18) Isabella female    48 Standard120   120      6
5     5 [17,18) Fabienne female    17 Standard120   120      7.3
6     6 [17,18) Fabienne female    17 Standard120   120      7.8
```

# Ensuring 'paper' is factor, not numeric

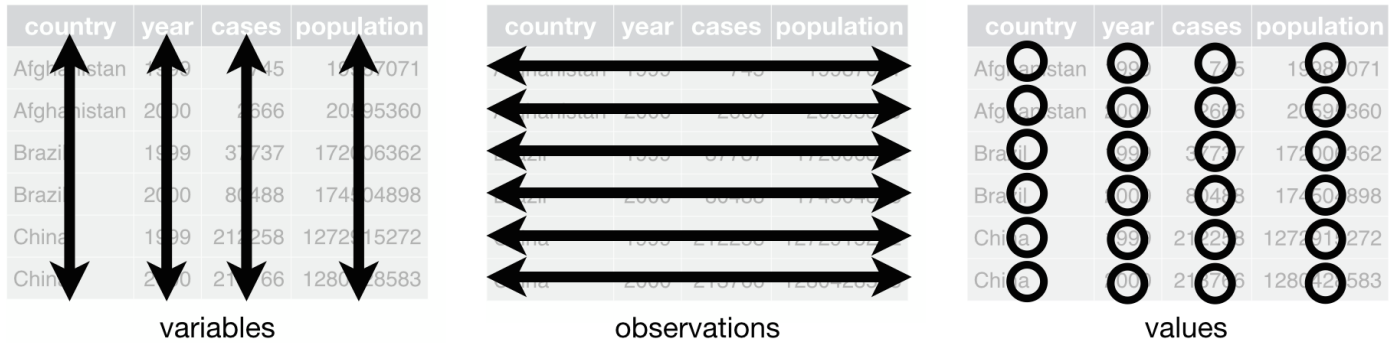
Using dplyr:

```
paperplanes <- paperplanes %>%
  mutate(paper = as.factor(paper))
```

R base:

```
paperplanes$paper <- as.factor(paperplanes$paper)
```

# Data must be a tidy data frame




**tidyr::gather(table4, key = "year", value = "cases", "1999", "2000")**

| country | year | cases |
|---|---|---|
| Afghanistan | 1999 | 745 |
| Afghanistan | 2000 | 2666 |
| Brazil | 1999 | 37737 |
| Brazil | 2000 | 80488 |
| China | 1999 | 212258 |
| China | 2000 | 213766 |

| country | 1999 | 2000 |
|---|---|---|
| Afghanistan | 745 | 2666 |
| Brazil | 37737 | 80488 |
| China | 212258 | 213766 |

table4

http://r4ds.had.co.nz/tidy-data.html

# Calling ggplot

```
library(ggplot2)
ggplot(paperplanes)
```

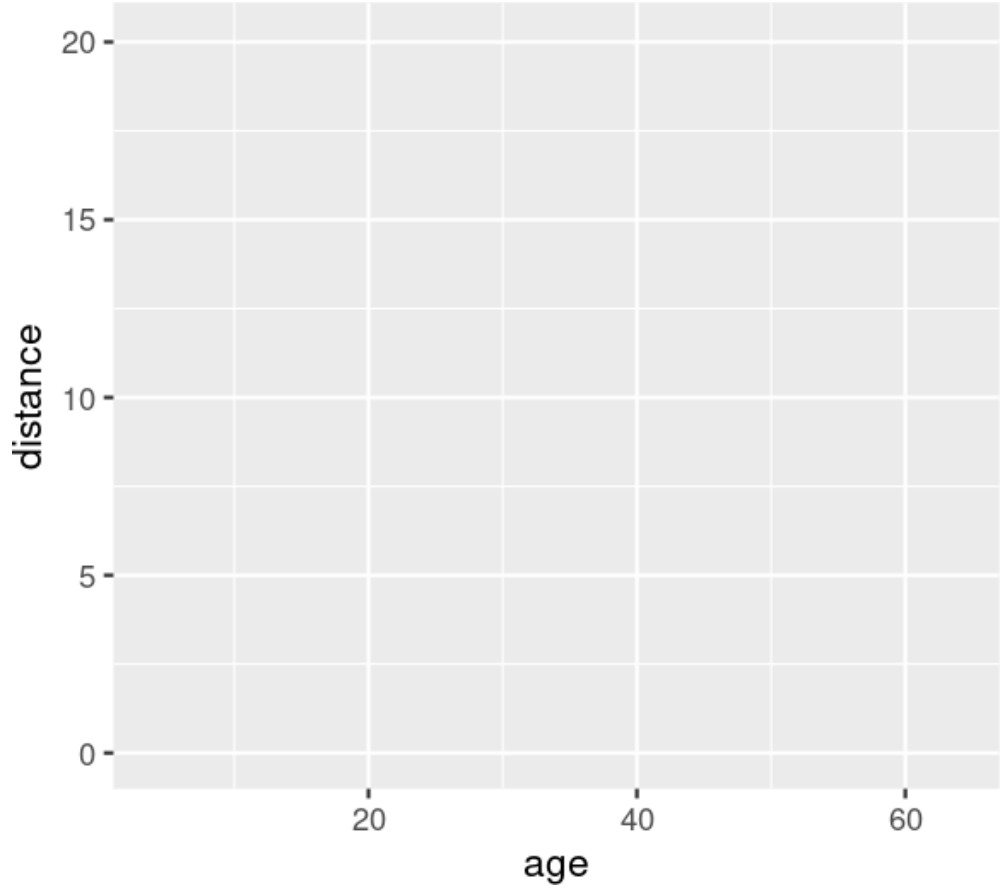
# First argument is a tidy data frame

```
ggplot(paperplanes)
```

# What variables as axes?

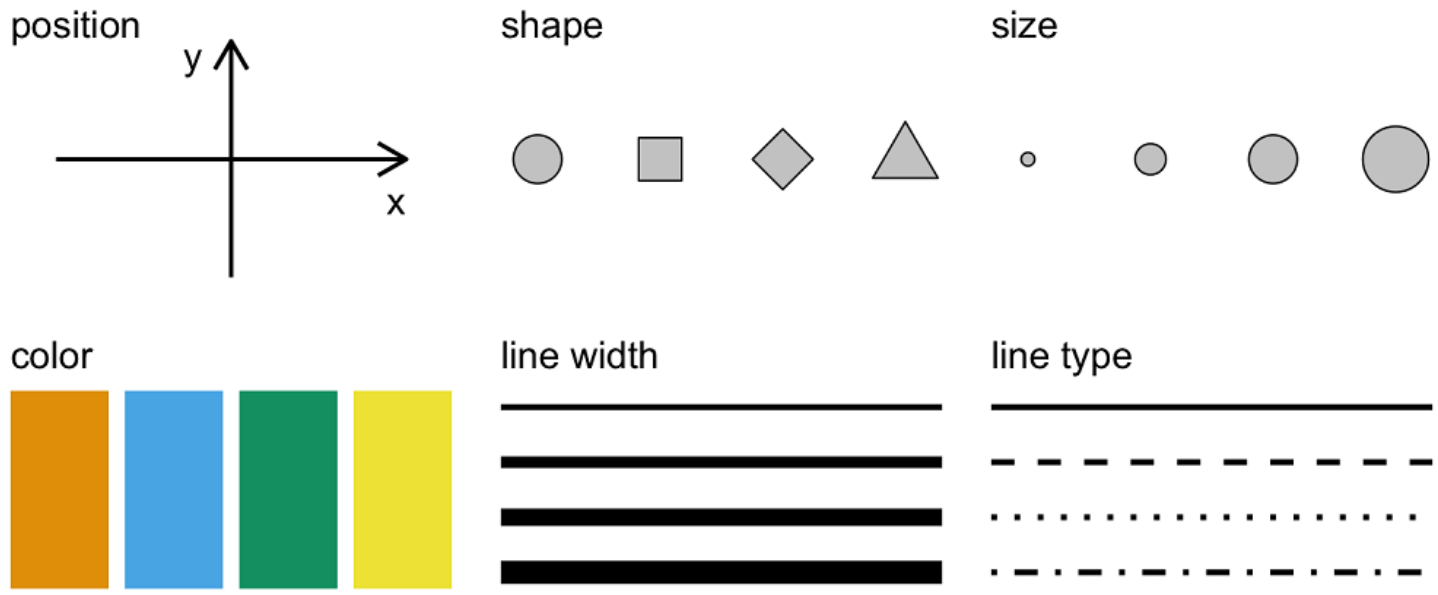
Note syntax: + followed by new line

```
ggplot(paperplanes) +
  aes(x = age, y = distance)
```



# Aesthetics (*aes*) map data variables (*age*, *distance*) to graphic elements (*axes*)

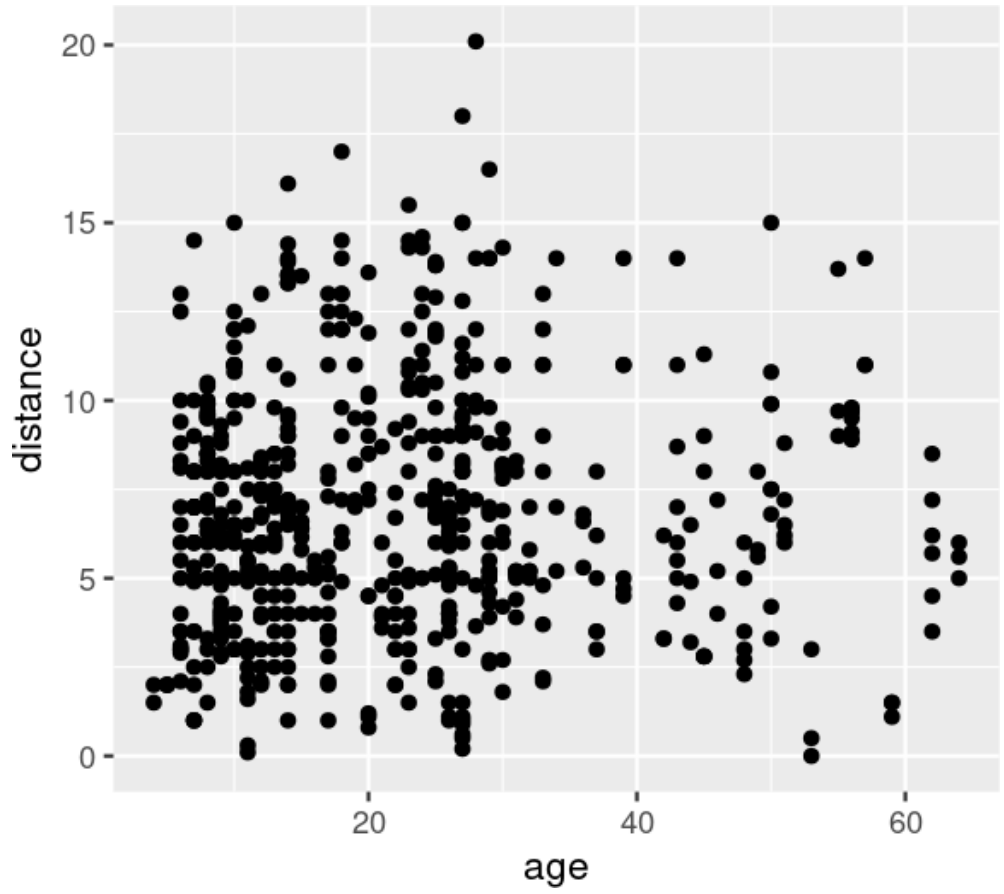
```
ggplot(paperplanes) +
  aes(x = age, y = distance)
```




http://serialmentor.com/dataviz/aesthetic-mapping.html

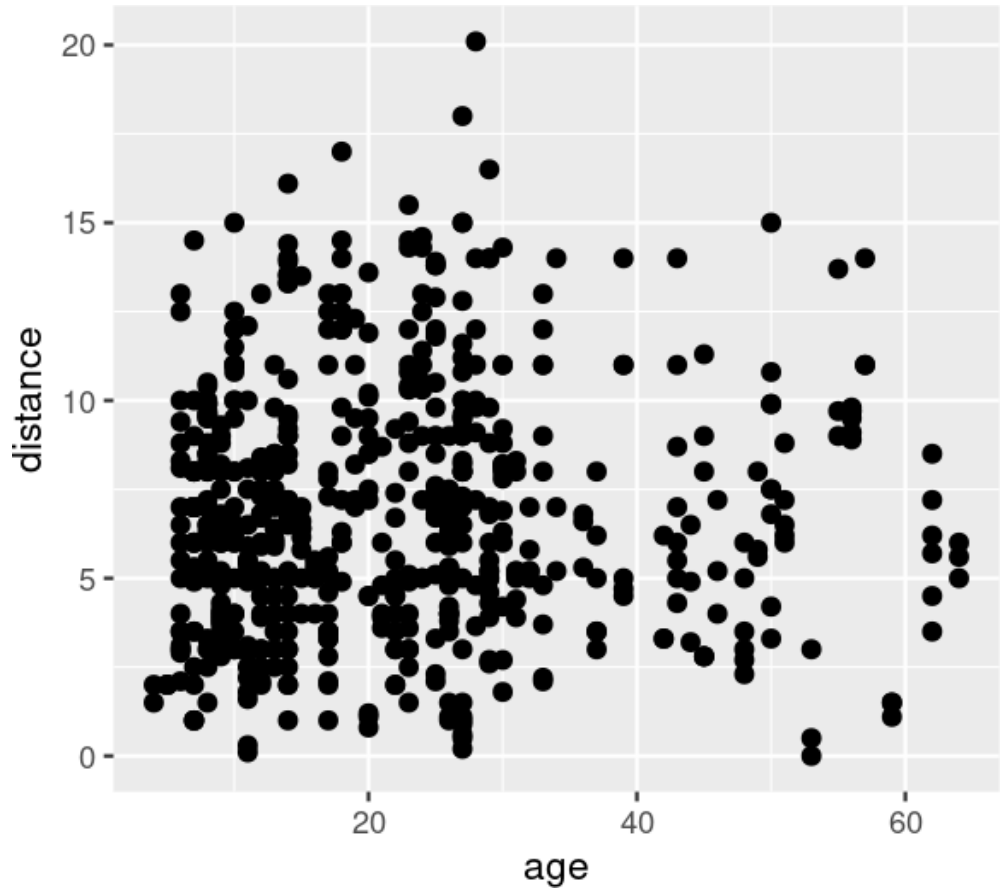
# Add layers (geoms)

```
ggplot(paperplanes) +
  aes(x = age, y = distance) +
  geom_point()
```



# Change point size and type

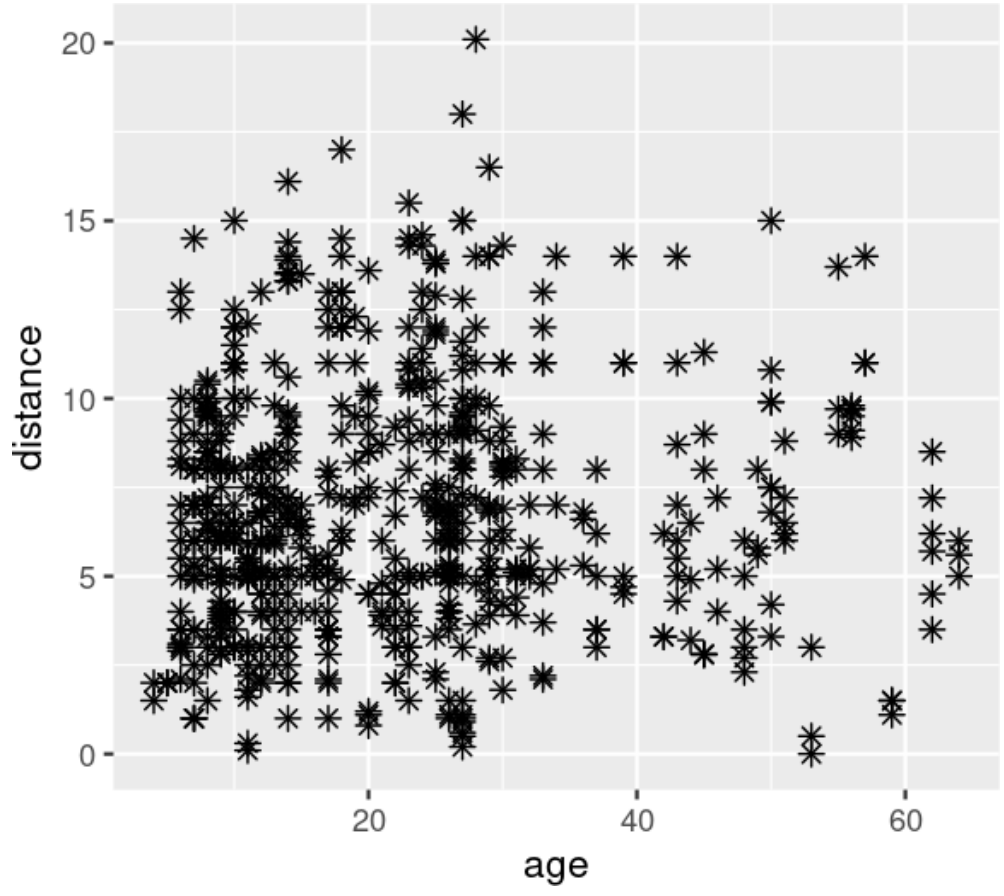
```
ggplot(paperplanes) +
  aes(x = age, y = distance) +
  geom_point(size = 2)
```




Check out `geom_point` help here

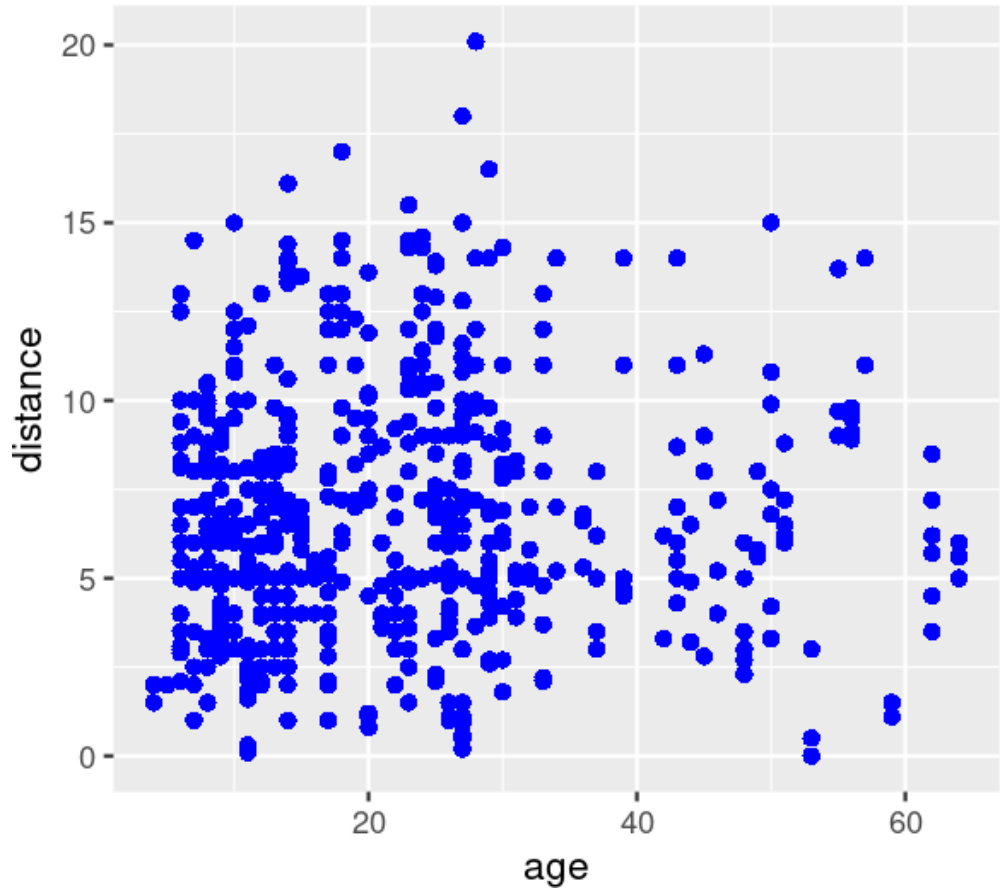
# Change point size and type

```
ggplot(paperplanes) +
  aes(x = age, y = distance) +
  geom_point(size = 2, shape = 8)
```



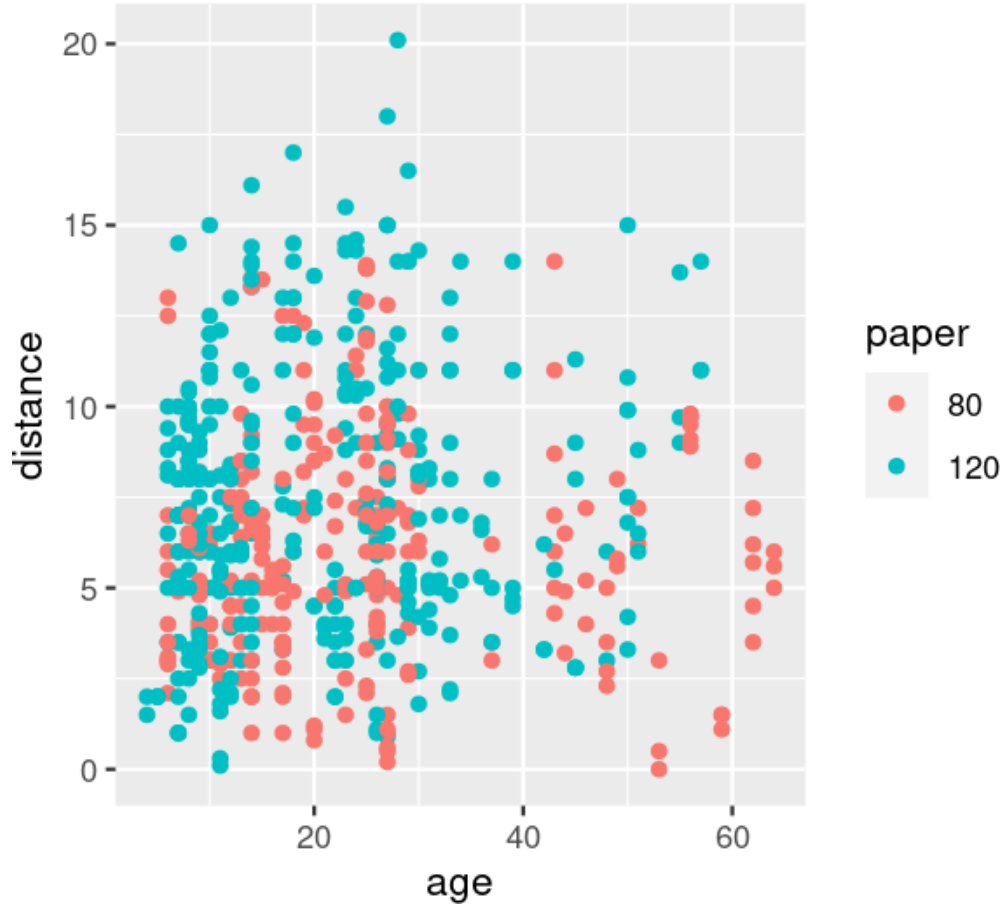
# Change point size and type

```
ggplot(paperplanes) +
  aes(x = age, y = distance) +
  geom_point(size = 2, shape = 16,
             colour = "blue")
```



# Map geom aesthetics (e.g. colour) to variable

```
ggplot(paperplanes) +
  aes(x = age, y = distance) +
  geom_point(aes(colour = paper))
```



Remember:

'aes' relates some graphical characteristic

(colour, size, shape...)

to a variable in the data

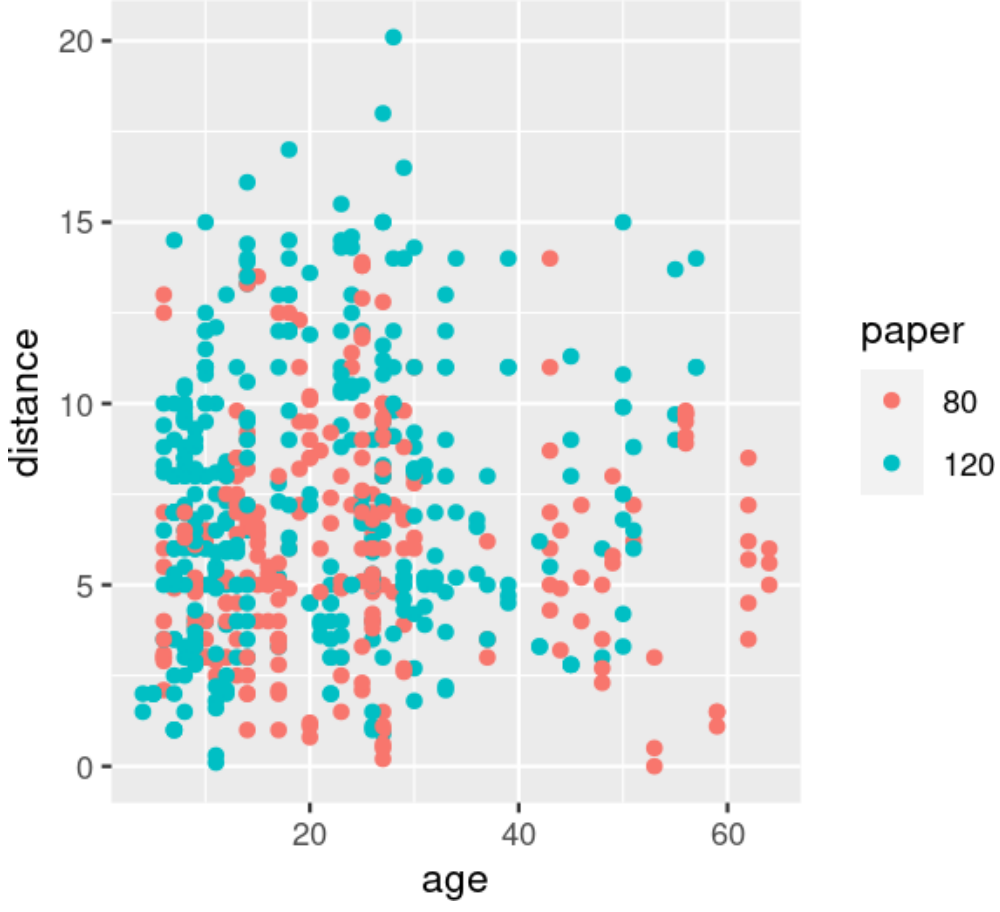
Note difference between

```
geom_point(colour = "blue")
# colour is given a concrete value ('blue')
```

```
geom_point(aes(colour = gender))
# colour maps a *variable* in the data (gender) USING `aes`
```

**This works:**

```
ggplot(paperplanes) +
  aes(x = age, y = distance) +
  geom_point(aes(colour = paper))
```




**This doesn't work:**

```
ggplot(paperplanes) +
  aes(x = age, y = distance) +
  geom_point(colour = paper)
```

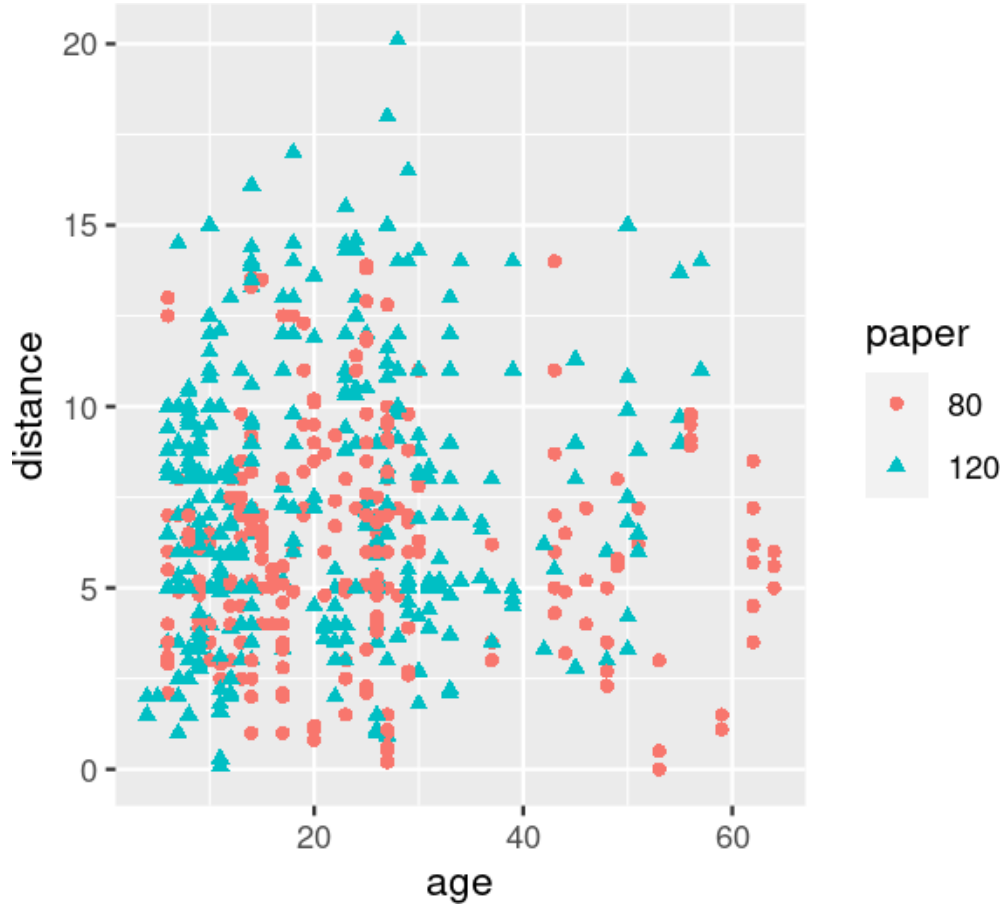
*Error in layer(data = data, mapping = mapping, stat = stat, geom = GeomPoint, : object 'paper' not found*

'paper' is a variable in dataframe, hence

**must use `aes`**

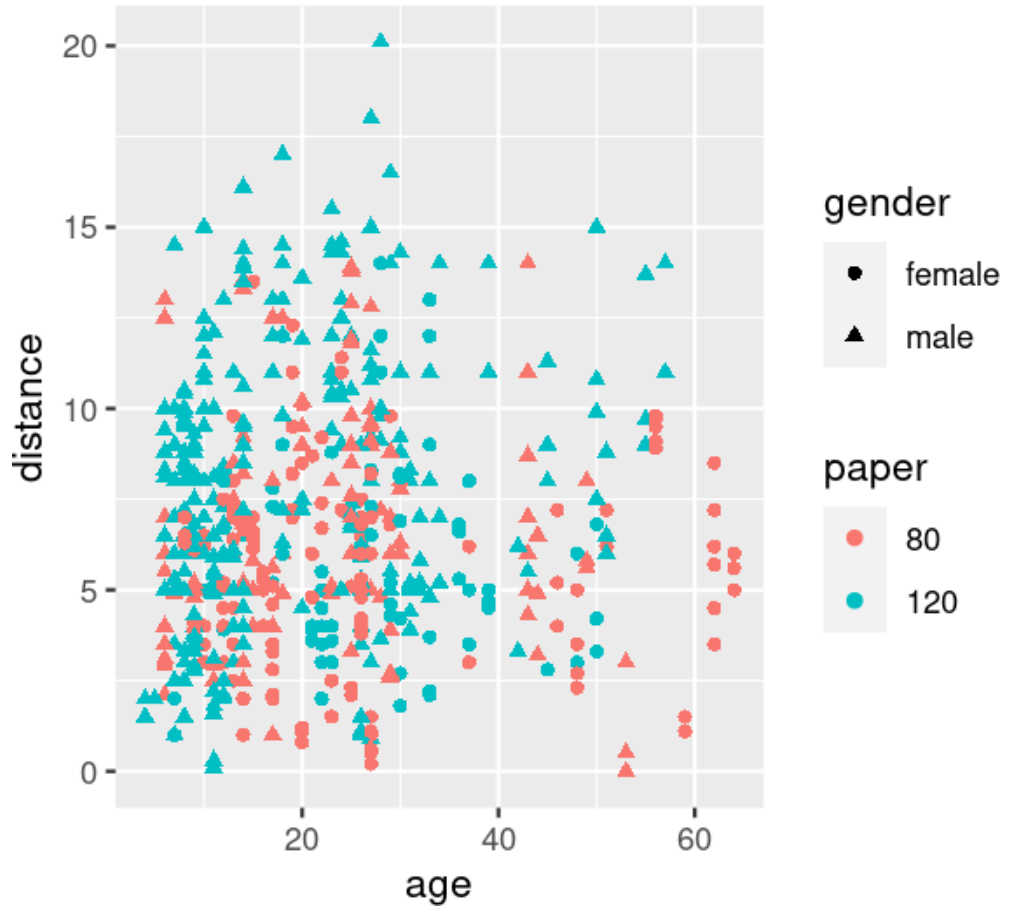
# Map geom aesthetics (colour, shape) to variable

```
ggplot(paperplanes) +
  aes(x = age, y = distance) +
  geom_point(aes(colour = paper,
                 shape = paper))
```



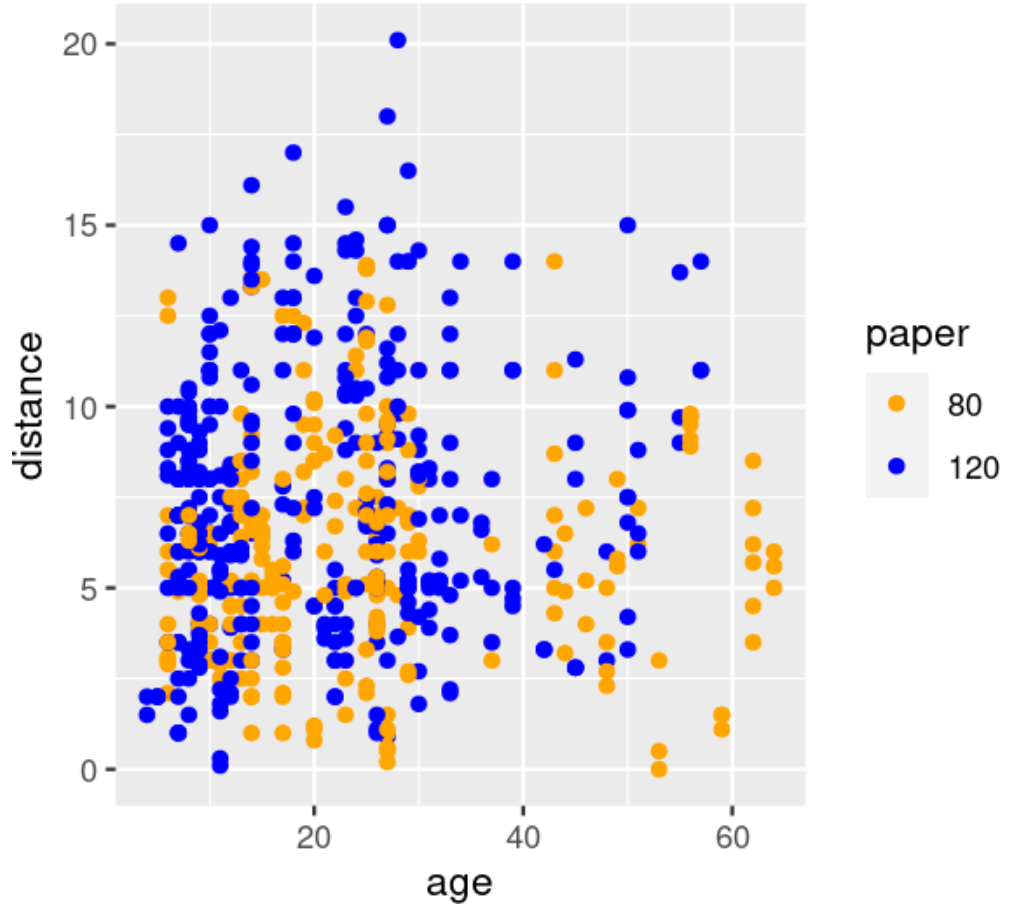
# Map geom aesthetics (colour, shape) to variable

```
ggplot(paperplanes) +
  aes(x = age, y = distance) +
  geom_point(aes(colour = paper,
                 shape = gender))
```



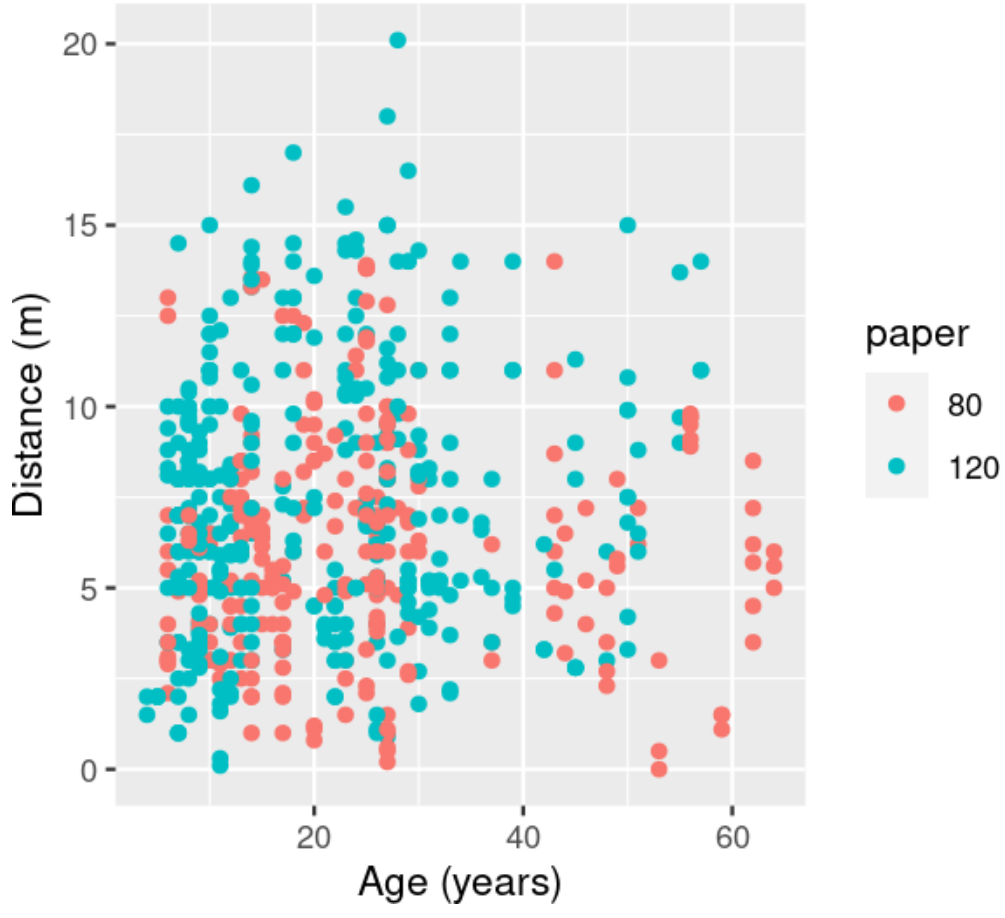
# Change colour scale

```
ggplot(paperplanes) +
  aes(x = age, y = distance) +
  geom_point(aes(colour = paper)) +
  scale_colour_manual(values = c("orange", "b
```



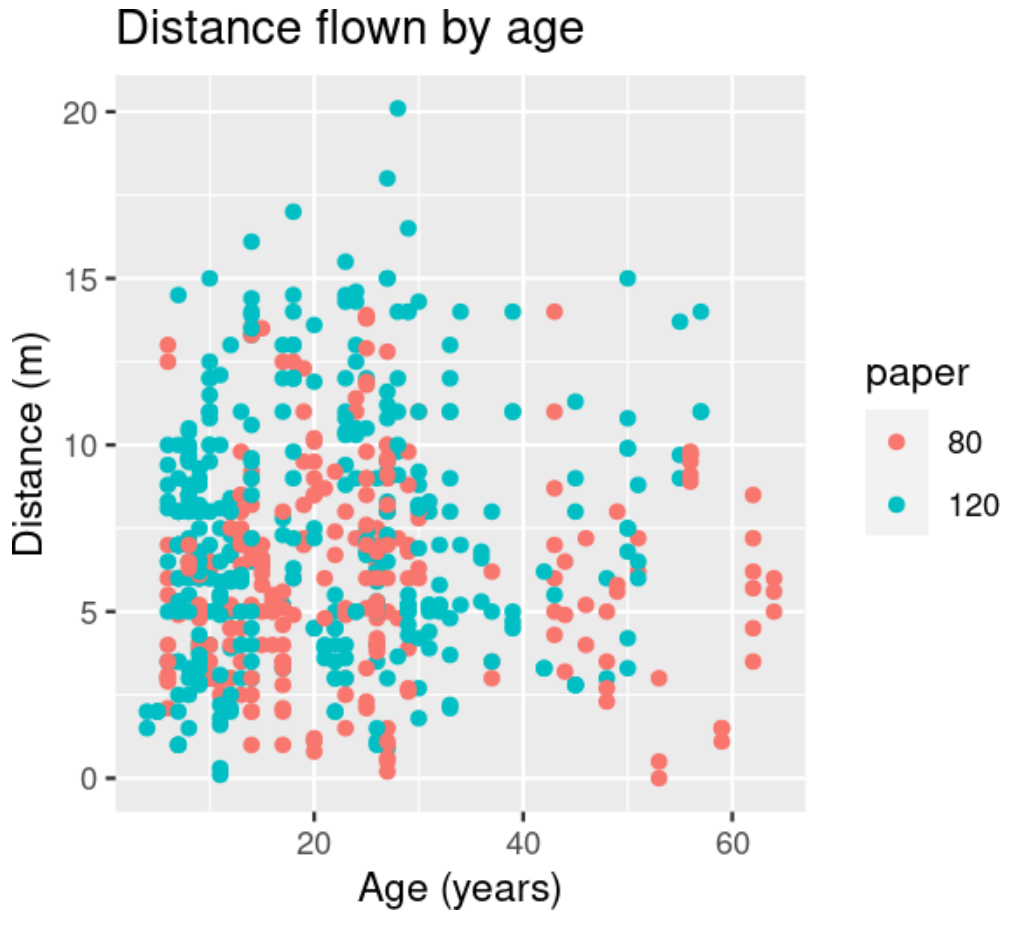
# Change axis labels: xlab & ylab

```
ggplot(paperplanes) +
  aes(x = age, y = distance) +
  geom_point(aes(colour = paper)) +
  labs(x = "Age (years)",
       y = "Distance (m)")
```



# Set title

```
ggplot(paperplanes) +
  aes(x = age, y = distance) +
  geom_point(aes(colour = paper)) +
  labs(x = "Age (years)",
       y = "Distance (m)") +
  labs(title = "Distance flown by age")
```

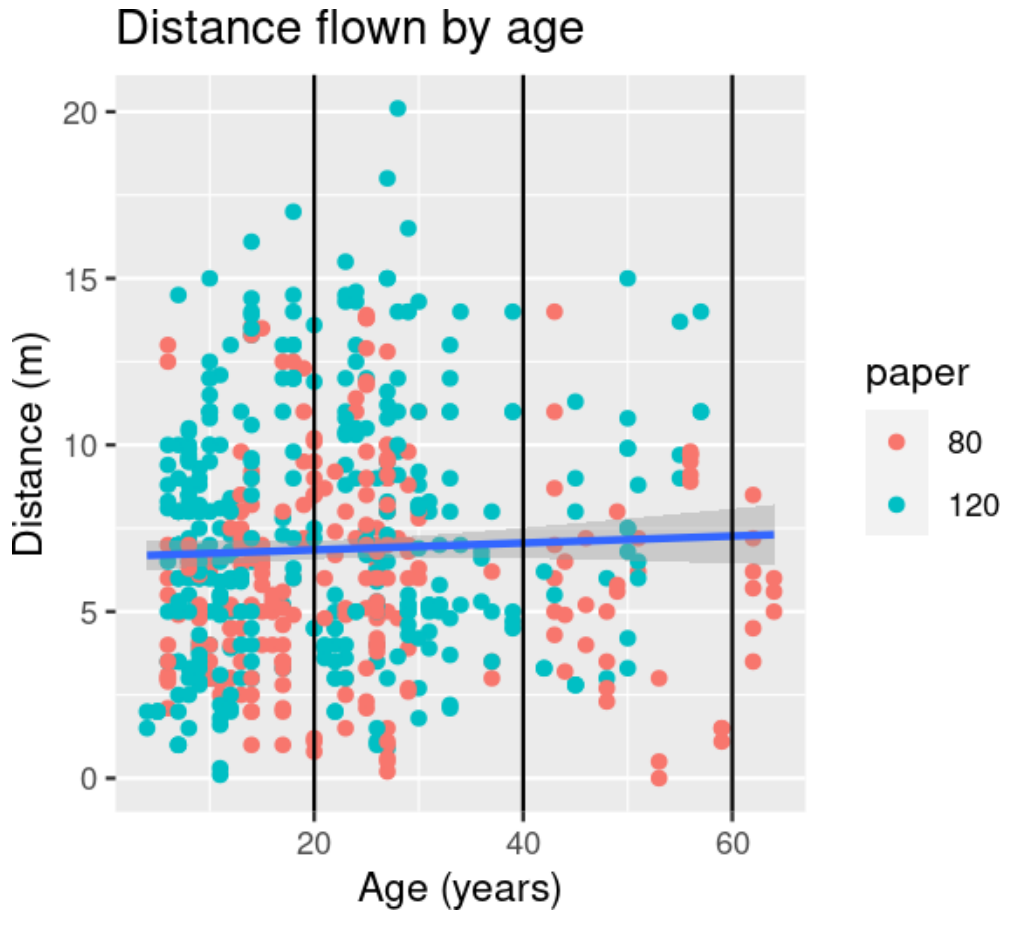


# Adding more layers

```
ggplot(paperplanes) +
  aes(x = age, y = distance) +
  geom_point(aes(colour = paper)) +
  labs(x = "Age (years)",
       y = "Distance (m)") +
  labs(title = "Distance flown by age") +
  geom_smooth(method = "lm")
```

# Adding more layers

```
ggplot(paperplanes) +
  aes(x = age, y = distance) +
  geom_point(aes(colour = paper)) +
  labs(x = "Age (years)",
       y = "Distance (m)") +
  labs(title = "Distance flown by age") +
  geom_smooth(method = "lm") +
  geom_vline(xintercept = c(20, 40, 60))
```



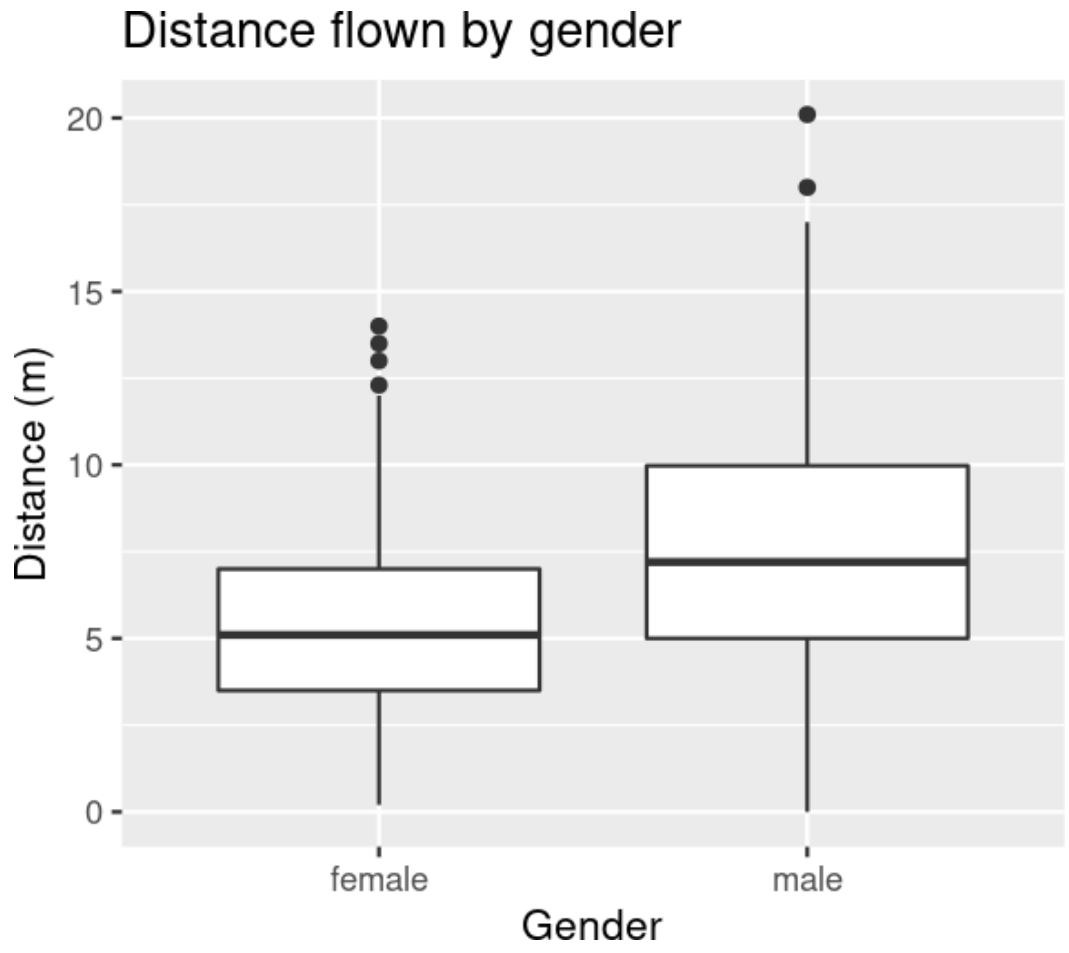
# Adding more layers

```
ggplot(paperplanes) +
  aes(x = age, y = distance) +
  geom_point(aes(colour = paper)) +
  labs(x = "Age (years)",
       y = "Distance (m)") +
  labs(title = "Distance flown by age") +
  geom_smooth(method = "lm") +
  geom_vline(xintercept = c(20, 40, 60)) +
  geom_hline(yintercept = 10)
```

# Summary

```
ggplot(paperplanes) +           # Name of (tidy) data frame
  aes(x = age, y = distance) + # Aesthetics (variables to map in axes)
  geom_point()                  # Geoms: geometric objects
```

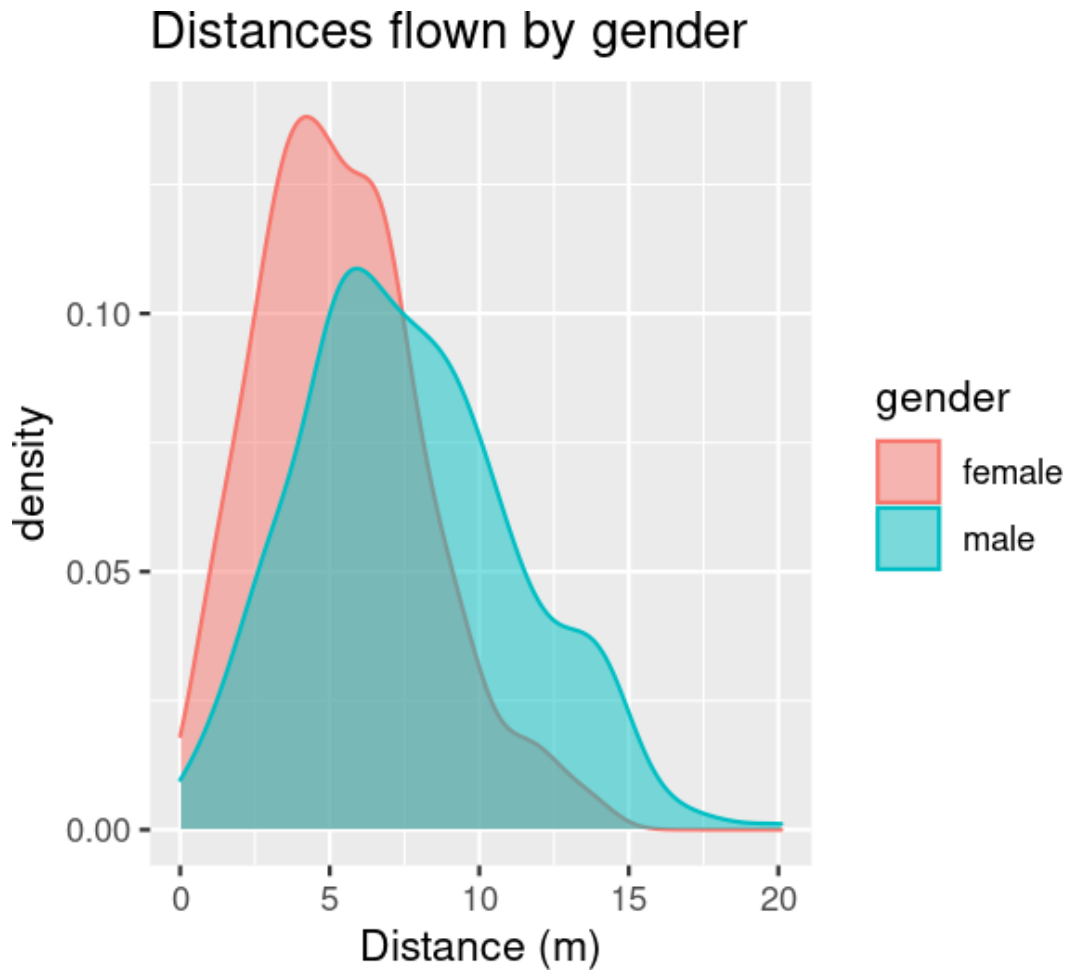
# Exercise: Make a plot like this one



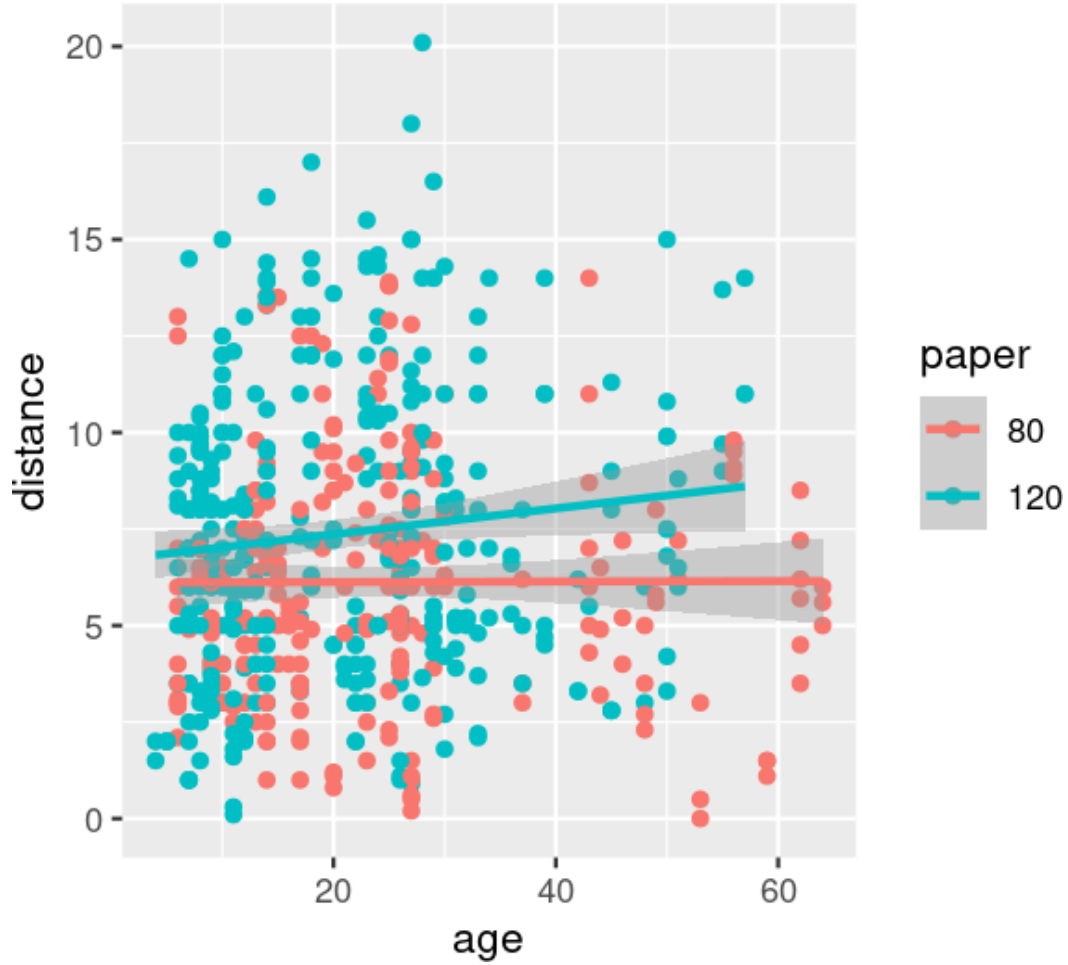
# Exercise: Make a plot like this one

# Exercise: Make a plot like this one

# Exercise: Make a plot like this one



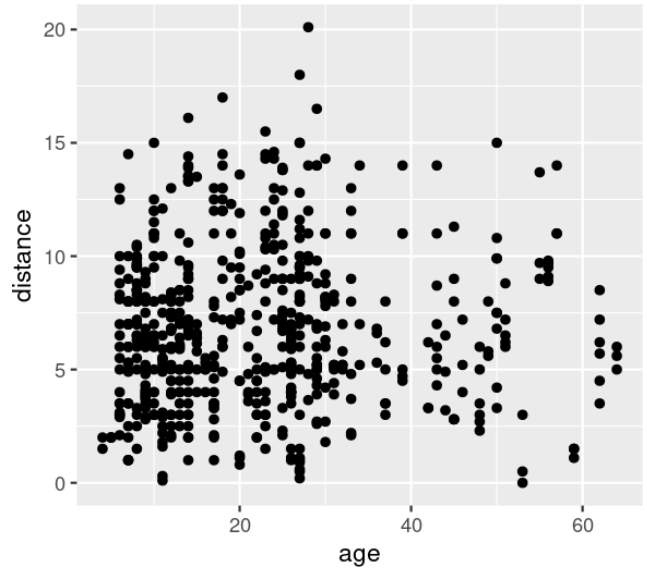
# Exercise: Make a plot like this one



# ggplot2 figures can be assigned to R objects

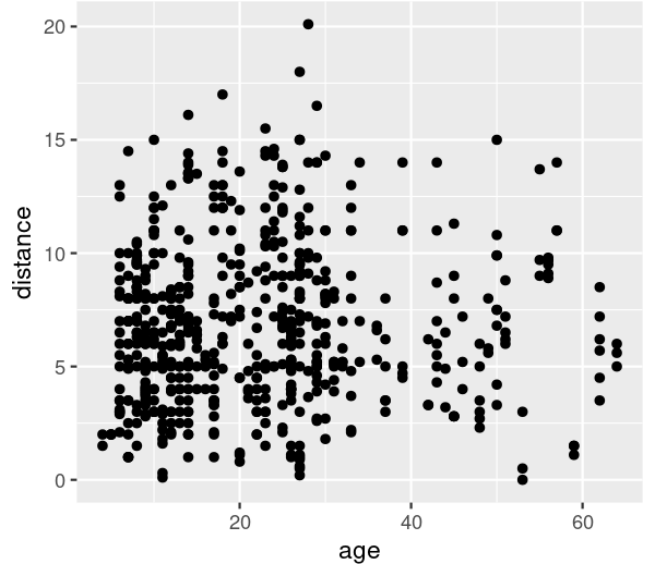
# Assigning ggplot objects

```
myplot <- ggplot(paperplanes) +
  aes(x = age, y = distance)
myplot + geom_point()
```



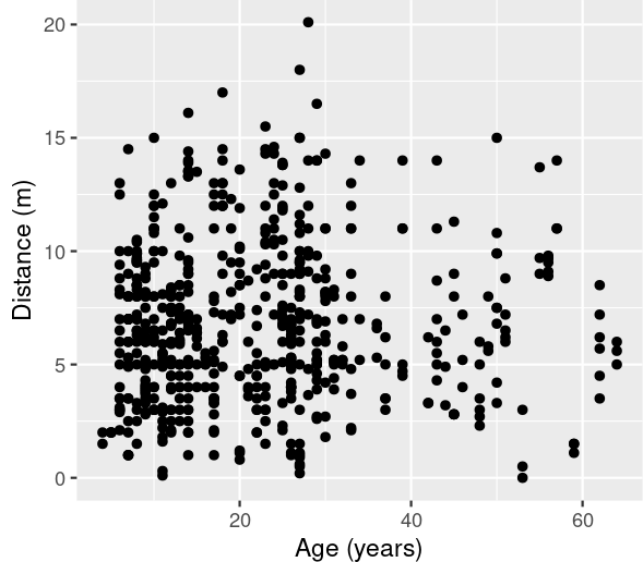
# Assigning ggplot objects

```
myplot <- ggplot(paperplanes) +
  aes(x = age, y = distance)
myplot <- myplot + geom_point()
myplot
```



# Assigning ggplot objects

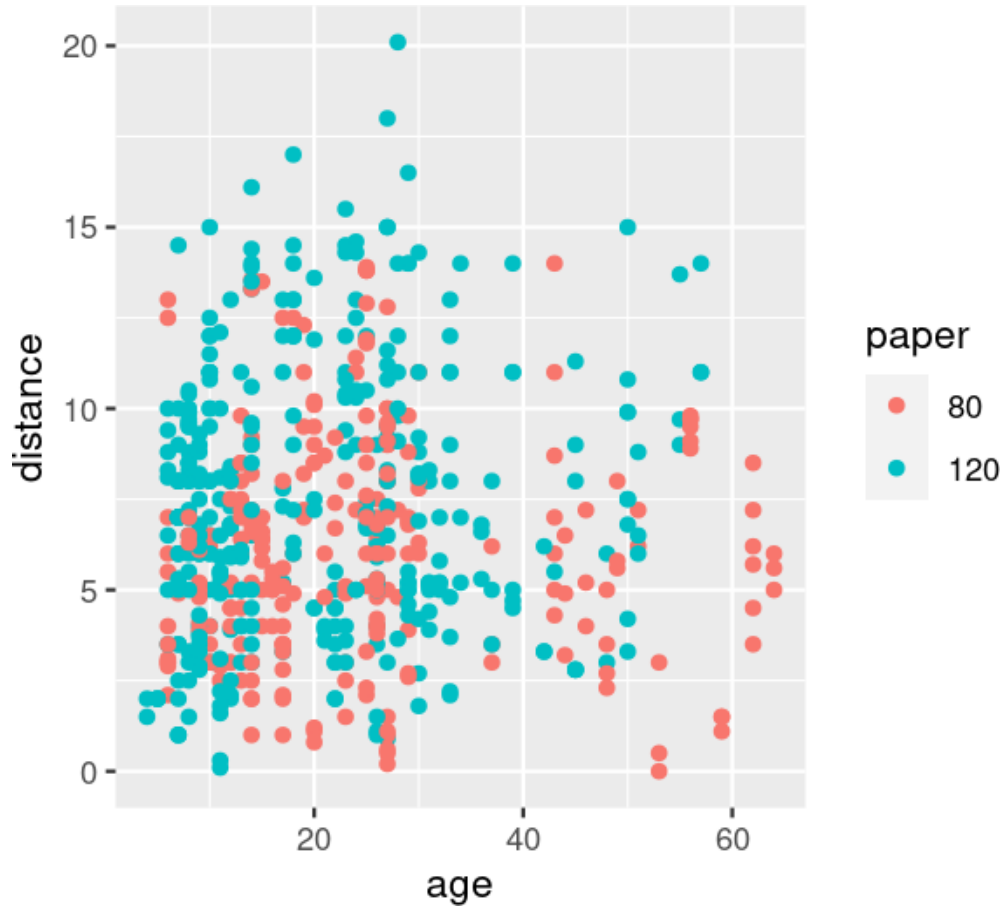
```
baseplot <- ggplot(paperplanes) +
  aes(x = age, y = distance)
scatterplot <- baseplot + geom_point()
labelled <- scatterplot + labs(x = "Age (years)", y = "Distance (m)")
labelled
```



# Themes: changing plot appearance

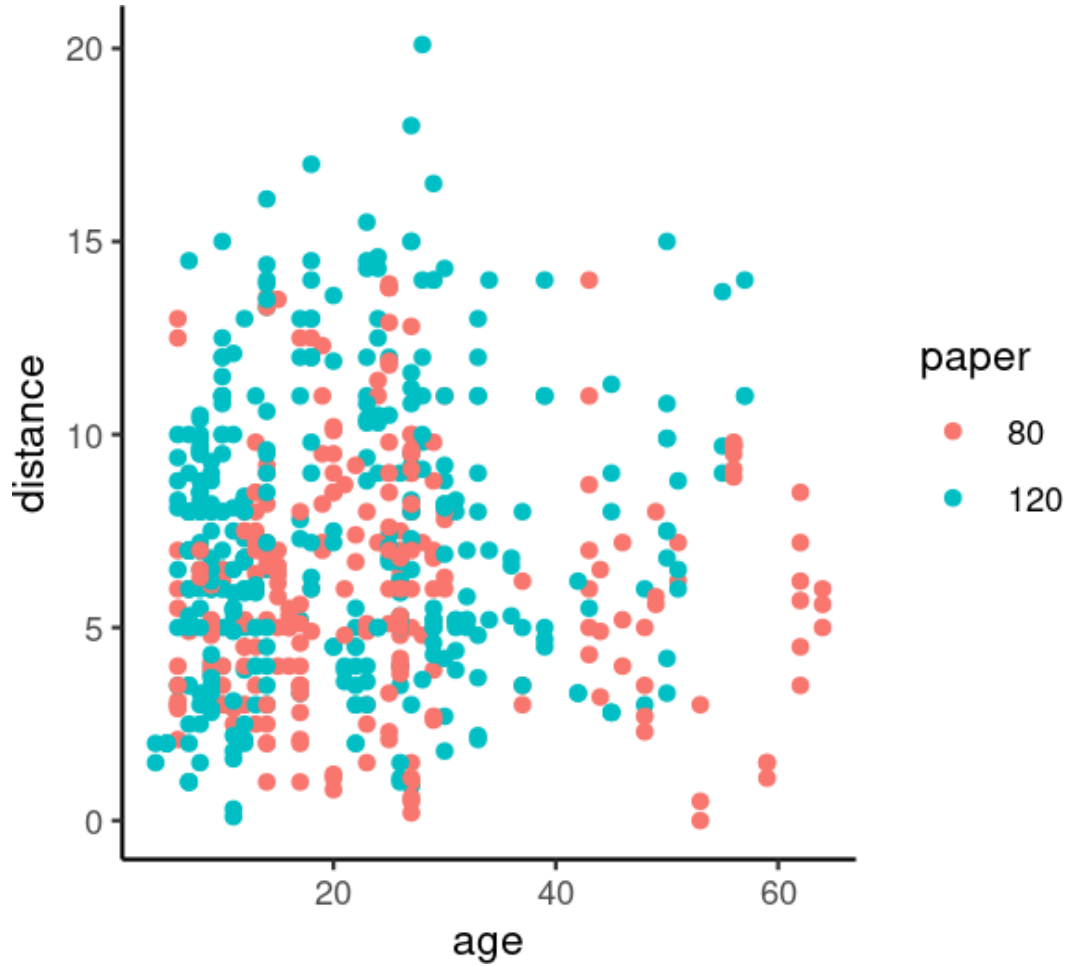
# Create 'myplot'

```
myplot <- ggplot(paperplanes) +
  aes(x = age,
      y = distance,
      colour = paper) +
  geom_point()

myplot
```



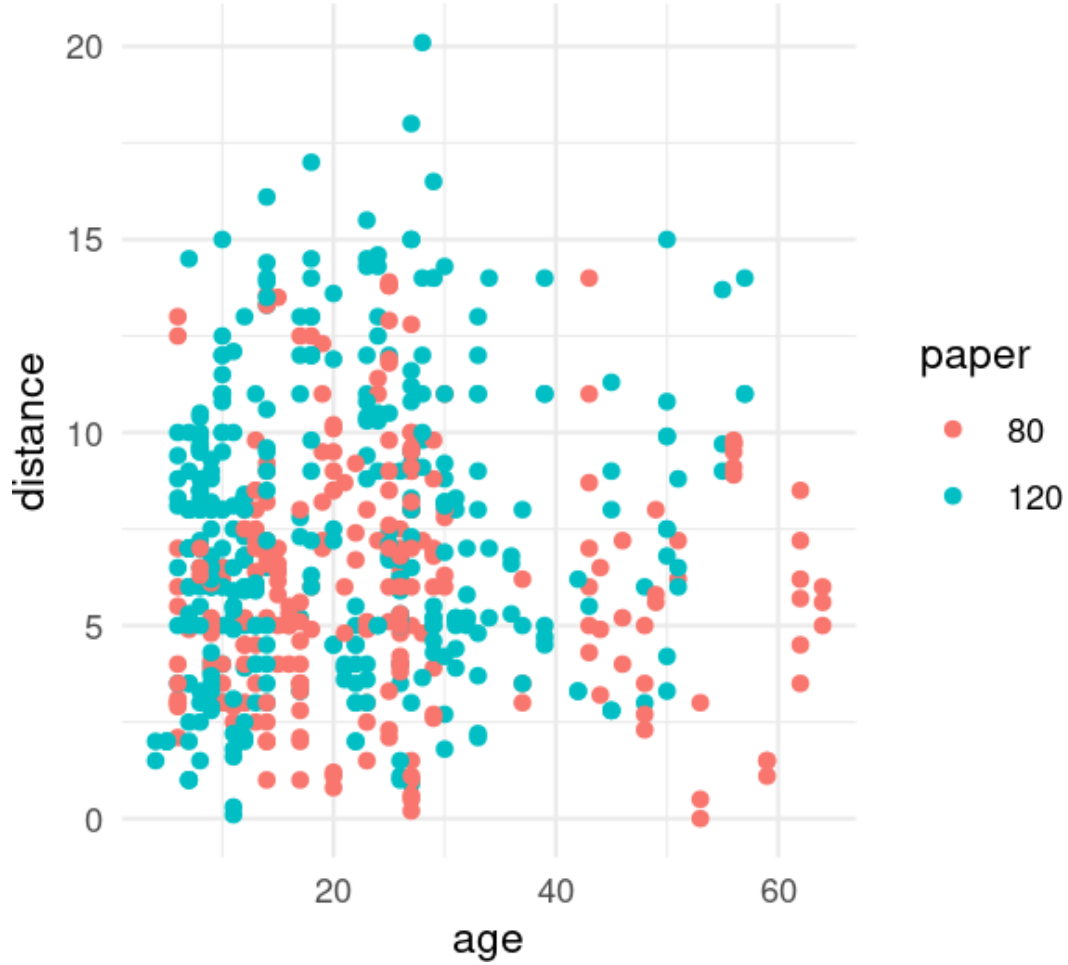
# Use theme_classic

```
myplot + theme_classic()
```



# theme_minimal

```
myplot + theme_minimal()
```

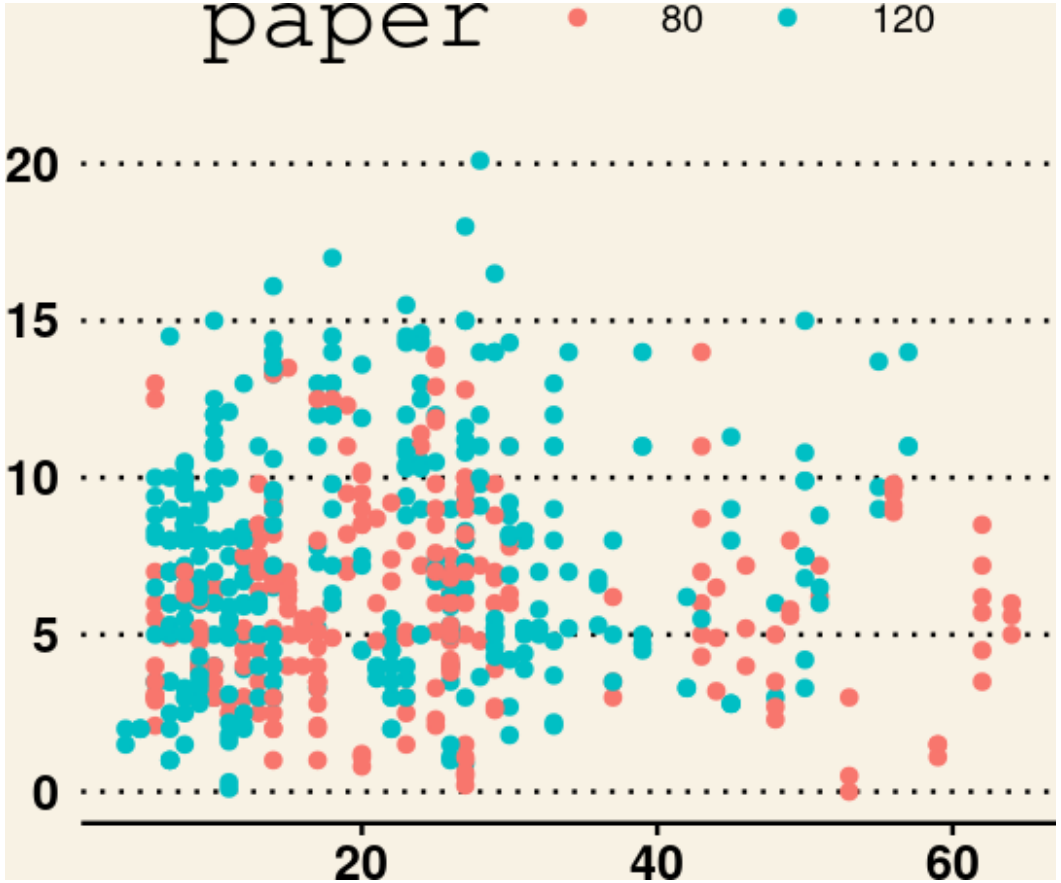


# Lots of themes out there

```
library(ggthemes)
myplot + theme_economist()
```

# Lots of themes out there

```
myplot + theme_wsj()
```

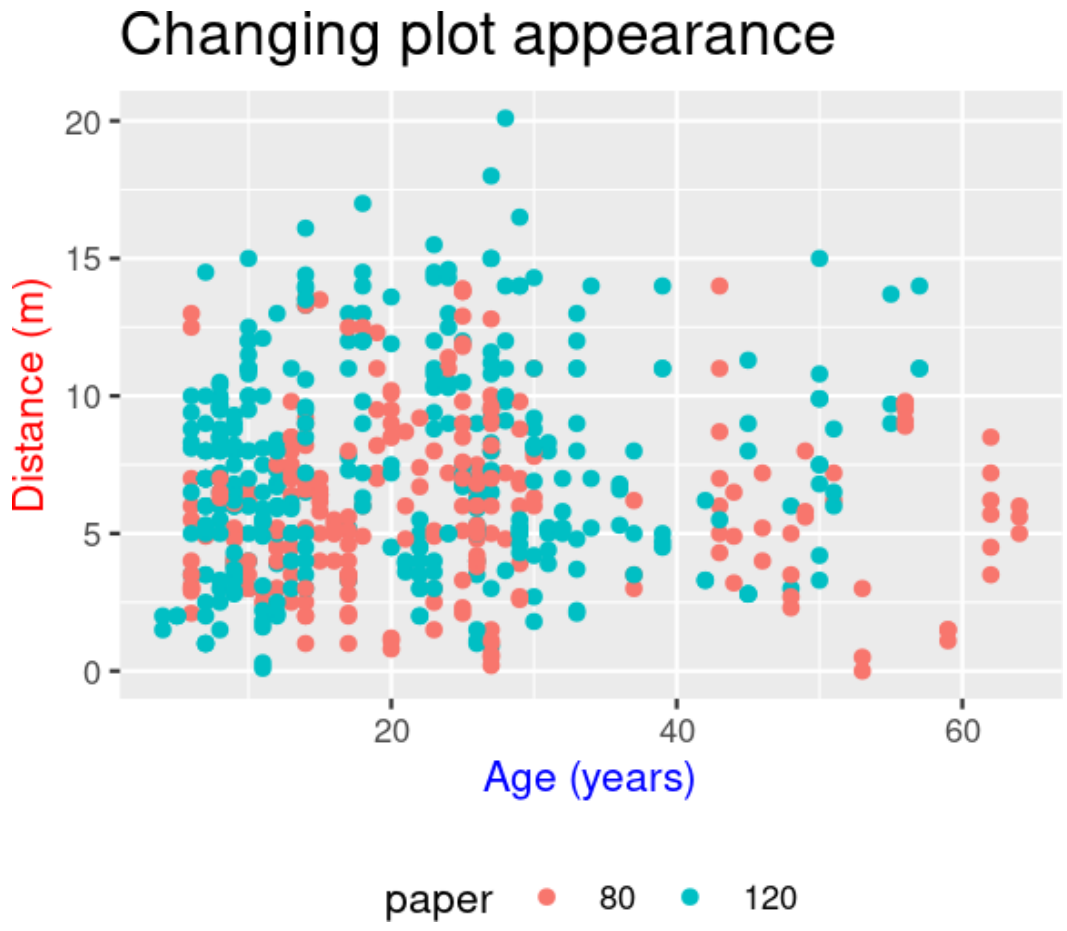


# Editing themes

```
?theme
```

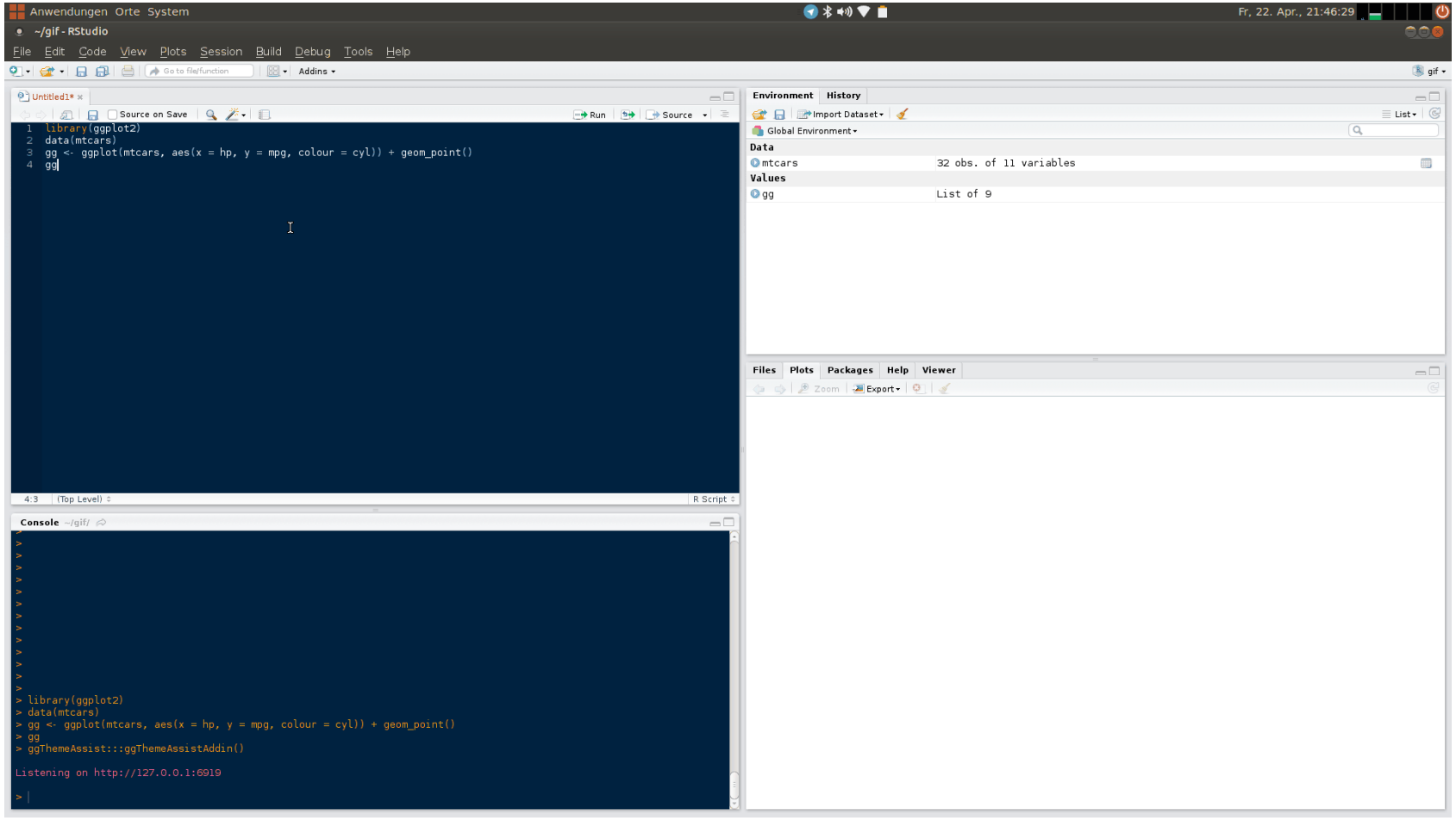
- `element_blank`
- `element_text`
- `element_line`
- `element_rect` (borders & backgrounds)

# Exercise: make a plot like this one



# Easily changing appearance with ggthemeassist (Rstudio addin)

https://github.com/calligross/ggthemeassist

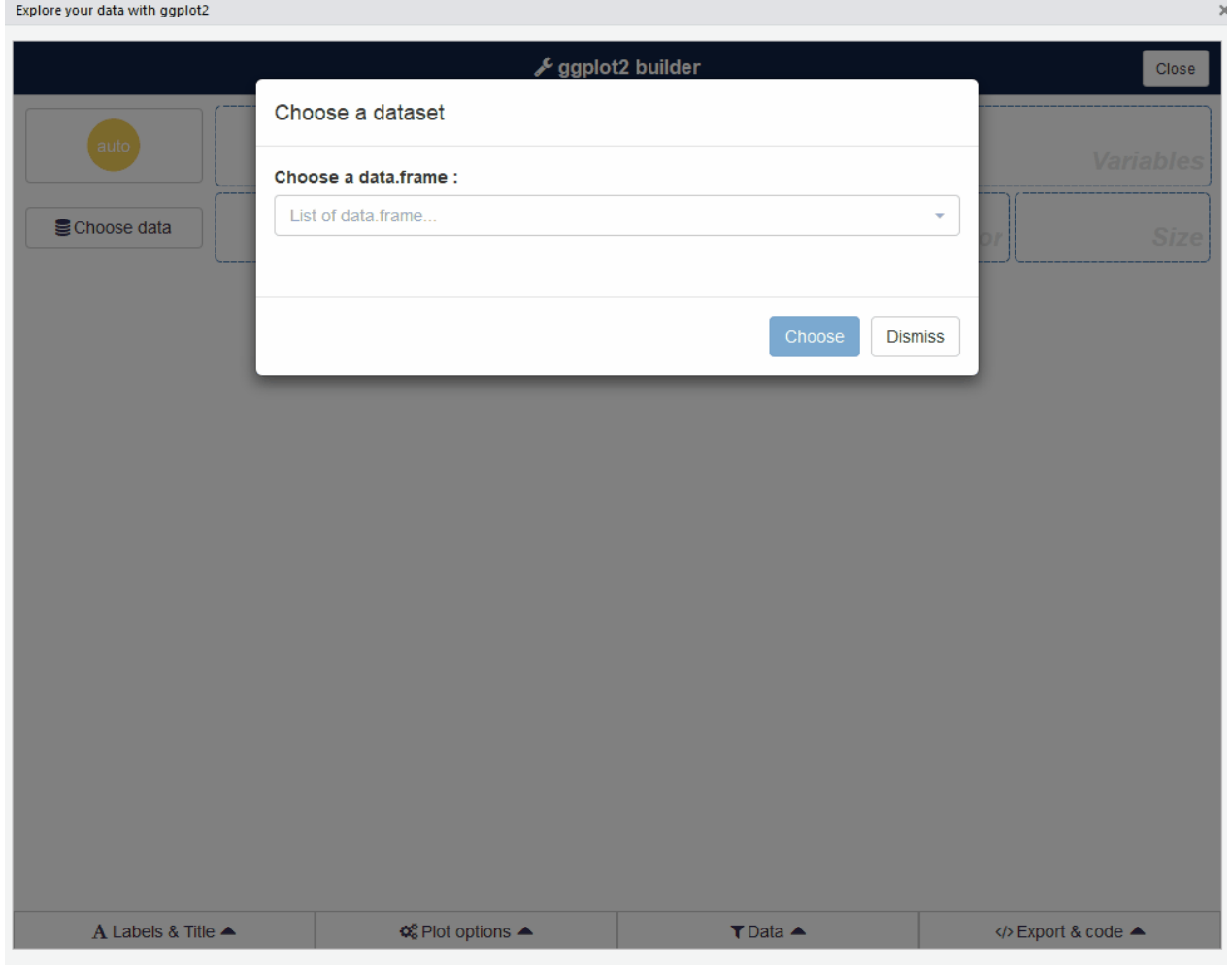


# Easily changing appearance with ggedit

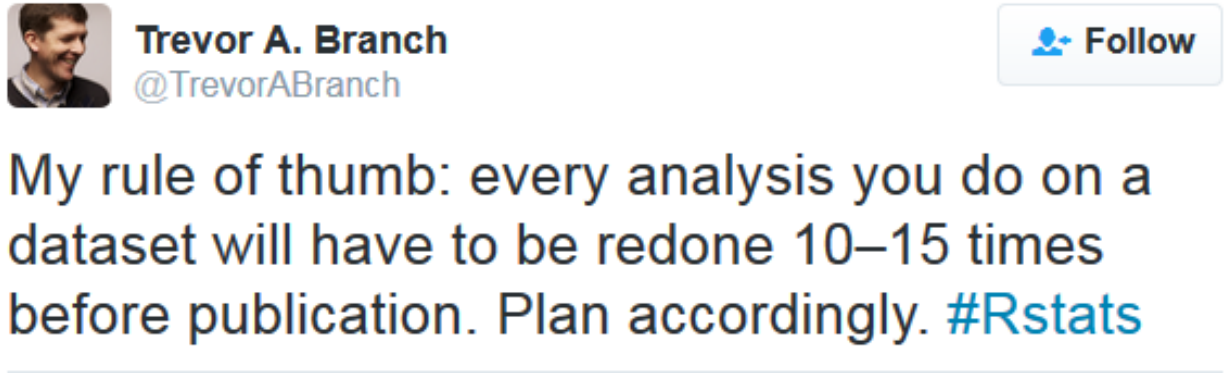
https://github.com/yonicd/ggedit

# esquisse: ggplot2 builder addin

https://github.com/dreamRs/esquisse



# Think twice before editing plots out of R




Why I think twice before editing plots out of R

Choosing the right visualization software

# Think twice before editing plots out of R

Referee #3: "Please increase font size in all figures"

```
myplot +
  theme(axis.title = element_text(size = 18))
```

# Publication-quality plots

```
library(cowplot)
myplot + theme_cowplot()
```

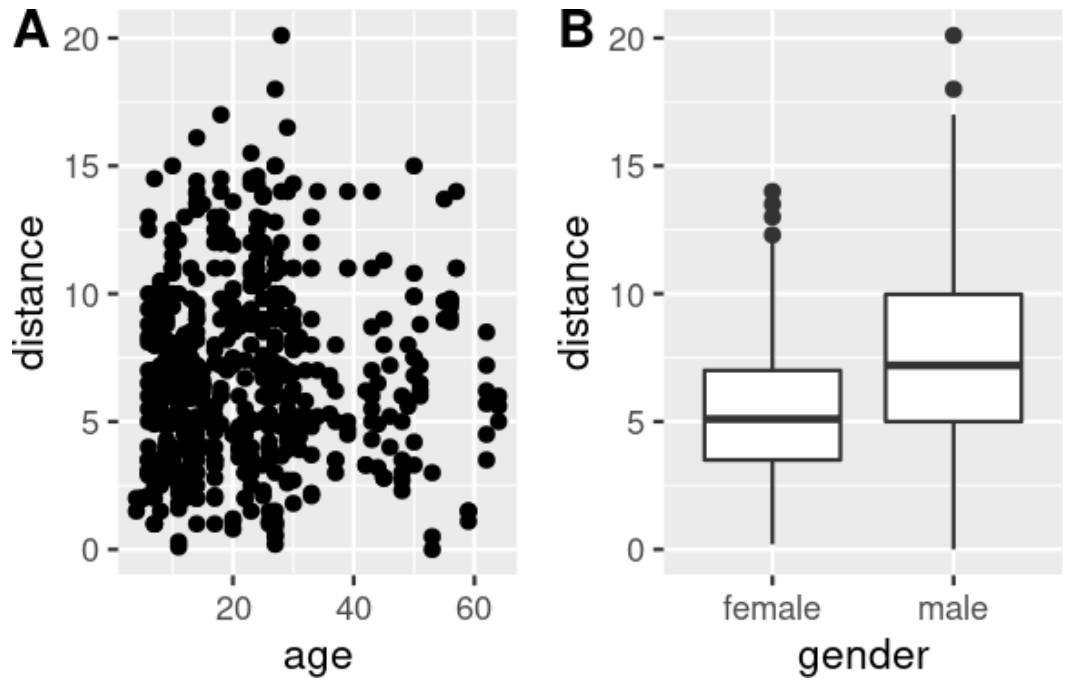
Some publication themes:

https://gist.github.com/Pakillo/c2c7ea11c528cc2ee20f#themes

# Composite figures

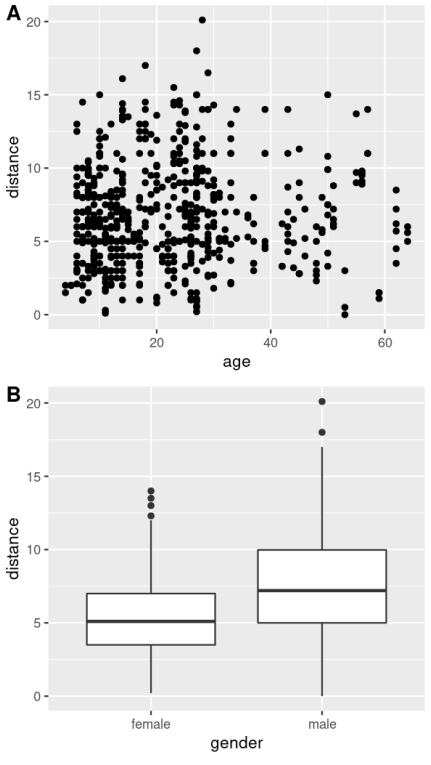
# Composite figures: cowplot

```
library(cowplot)
plot1 <- ggplot(paperplanes) + aes(age, distance) + geom_point()
plot2 <- ggplot(paperplanes) + aes(gender, distance) + geom_boxplot()
plot_grid(plot1, plot2, labels = "AUTO")
```



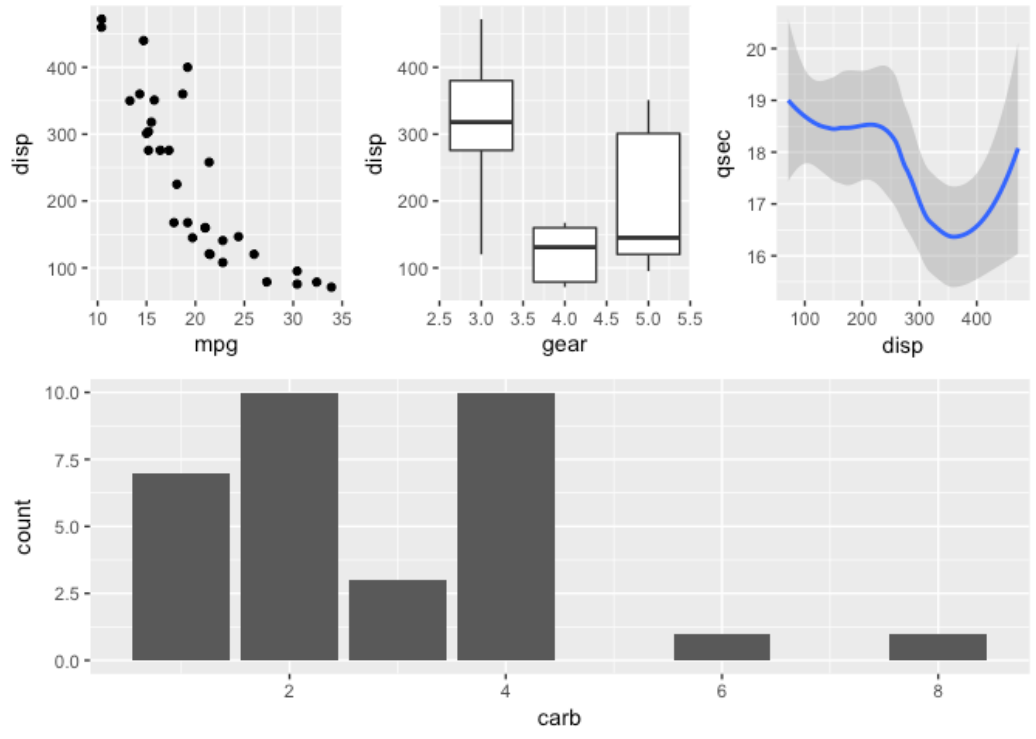
# Composite figures

```
plot_grid(plot1, plot2, labels = "AUTO", ncol = 1)
```



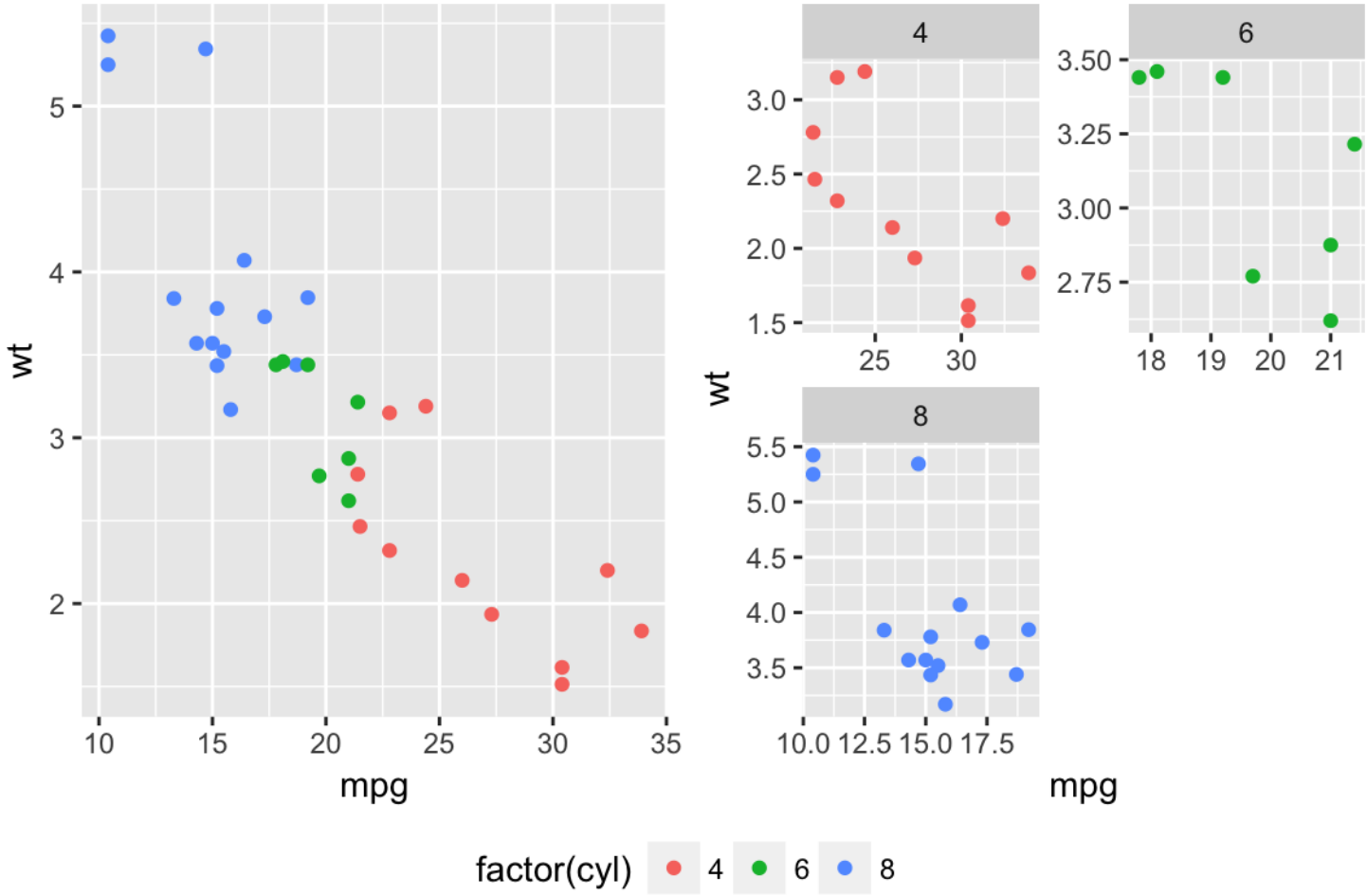
# Composite figures: patchwork

```
(p1 | p2 | p3) /
      p4
```




https://github.com/thomasp85/patchwork

# Composite figures: egg




https://cran.r-project.org/web/packages/egg/index.html

# Saving plot

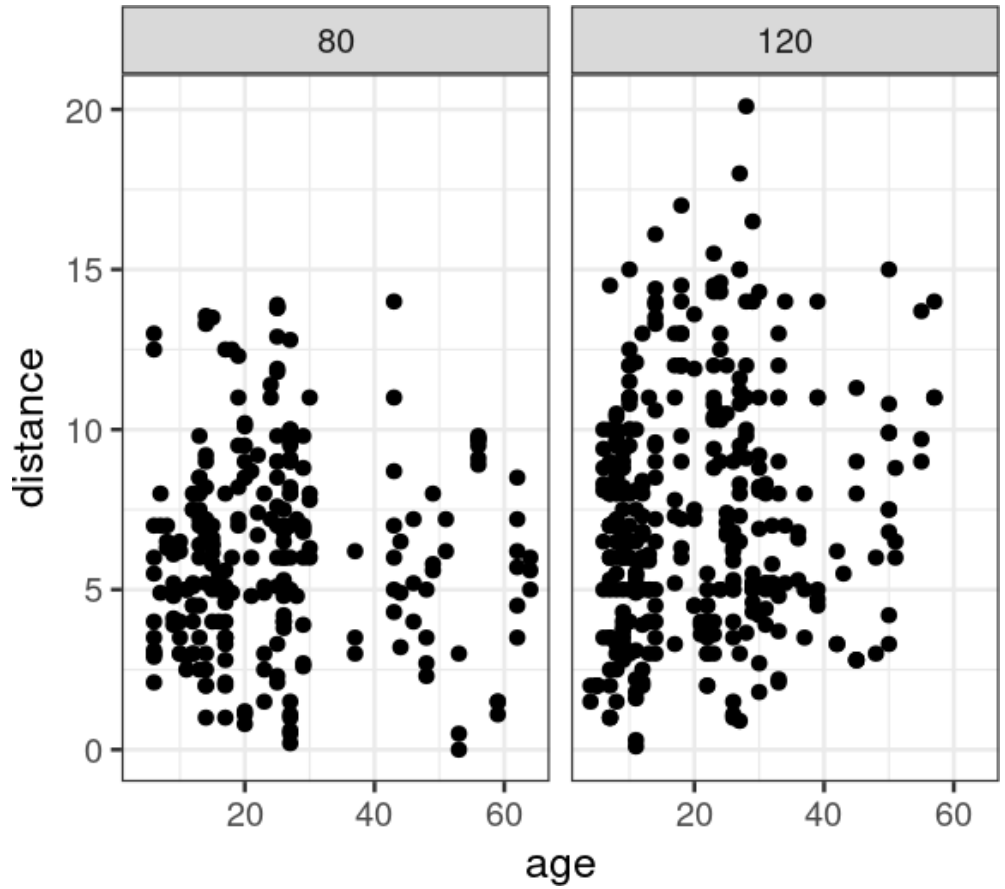
```
ggsave("myplot.pdf")
```

```
save_plot("myplot.pdf")
```

# Facetting (small multiples)

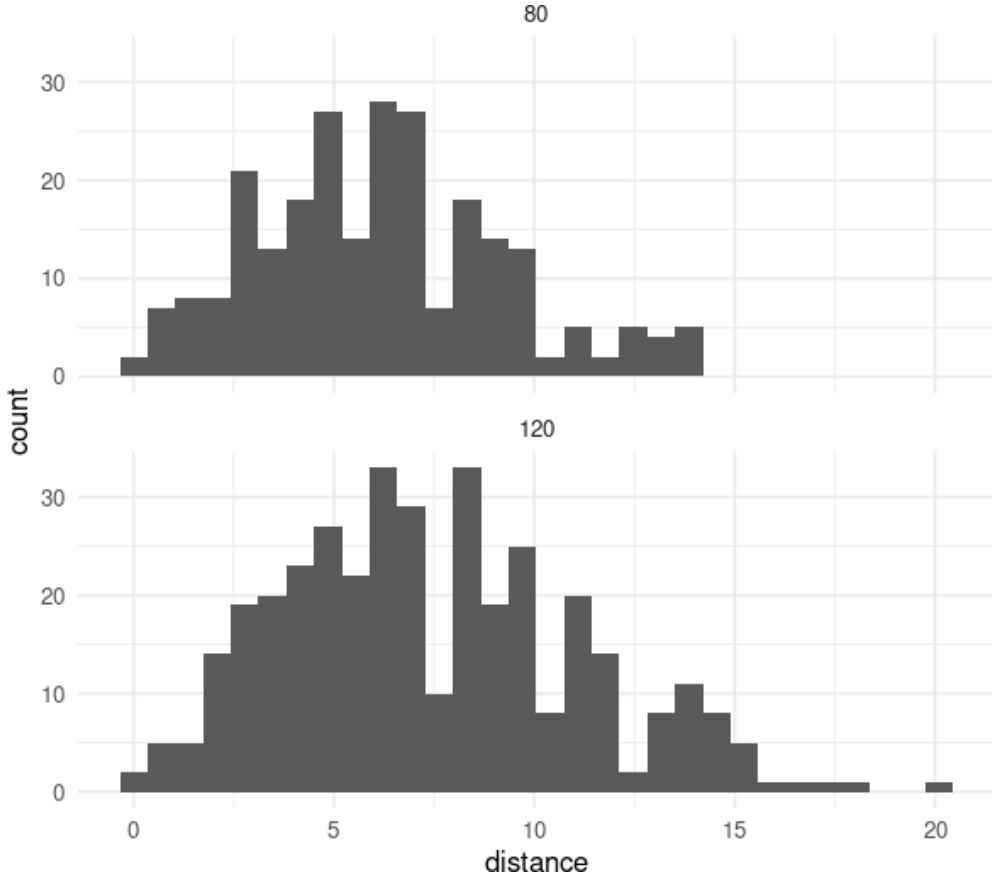
# Facetting

```
ggplot(paperplanes) +
  aes(x = age,
      y = distance) +
    geom_point() +
  theme_bw(base_size = 12) +
  facet_wrap(~paper)
```



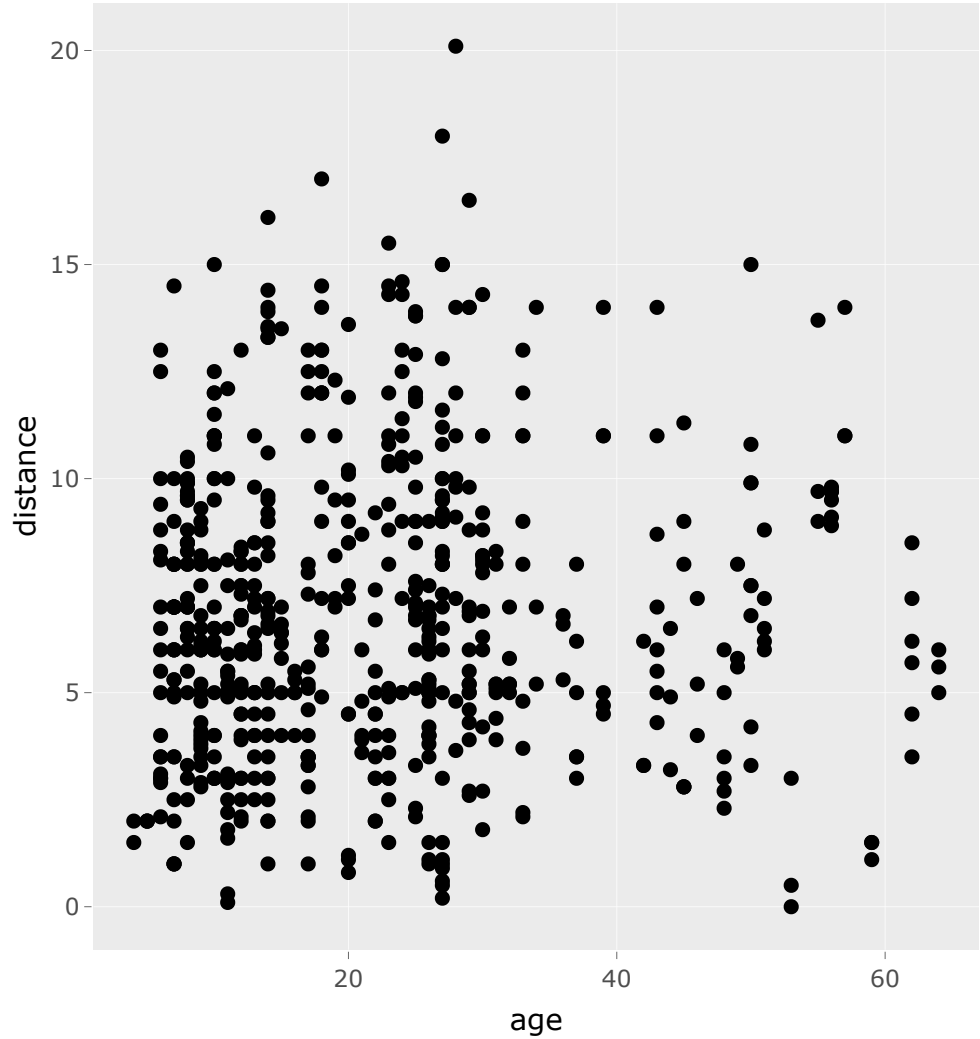
# Facetting

```
ggplot(paperplanes) +
  geom_histogram(aes(distance)) +
  theme_minimal(base_size = 8) +
  facet_wrap(~paper, nrow = 2)
```



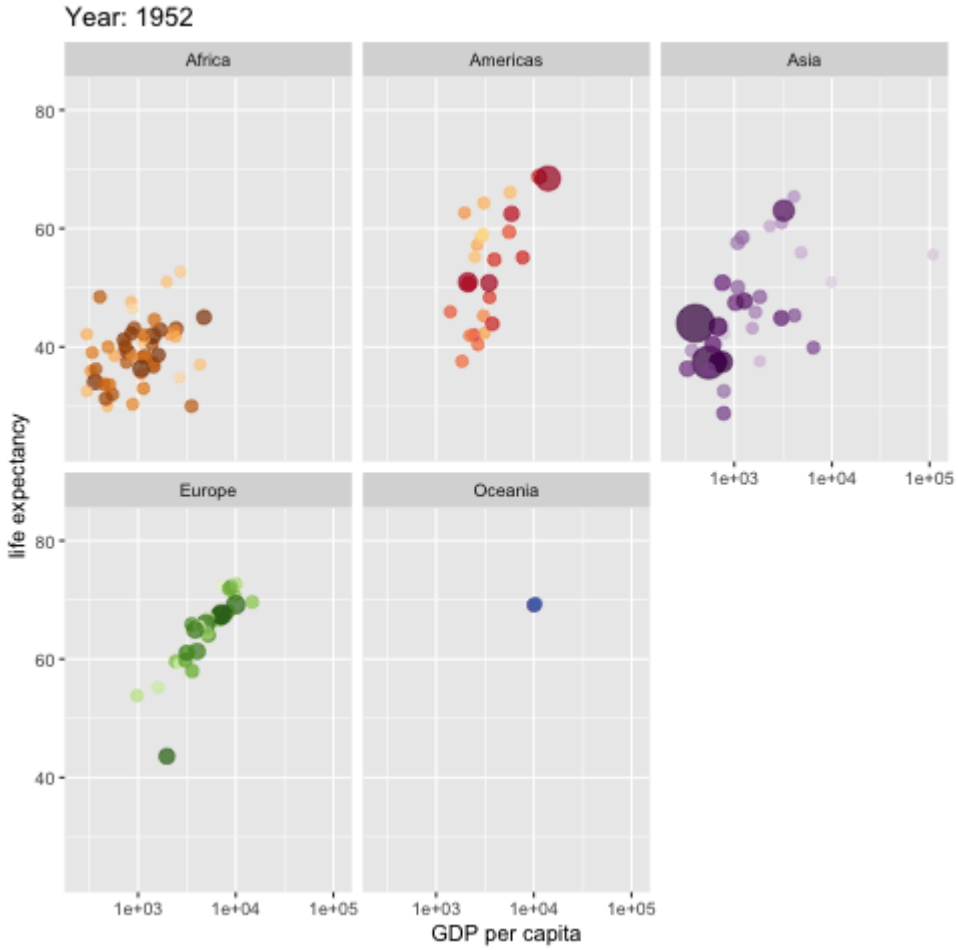
# Interactivity: plotly

```
library(plotly)

myplot <- ggplot(paperplanes) +
  aes(age, distance) +
  geom_point()

ggplotly(myplot)
```

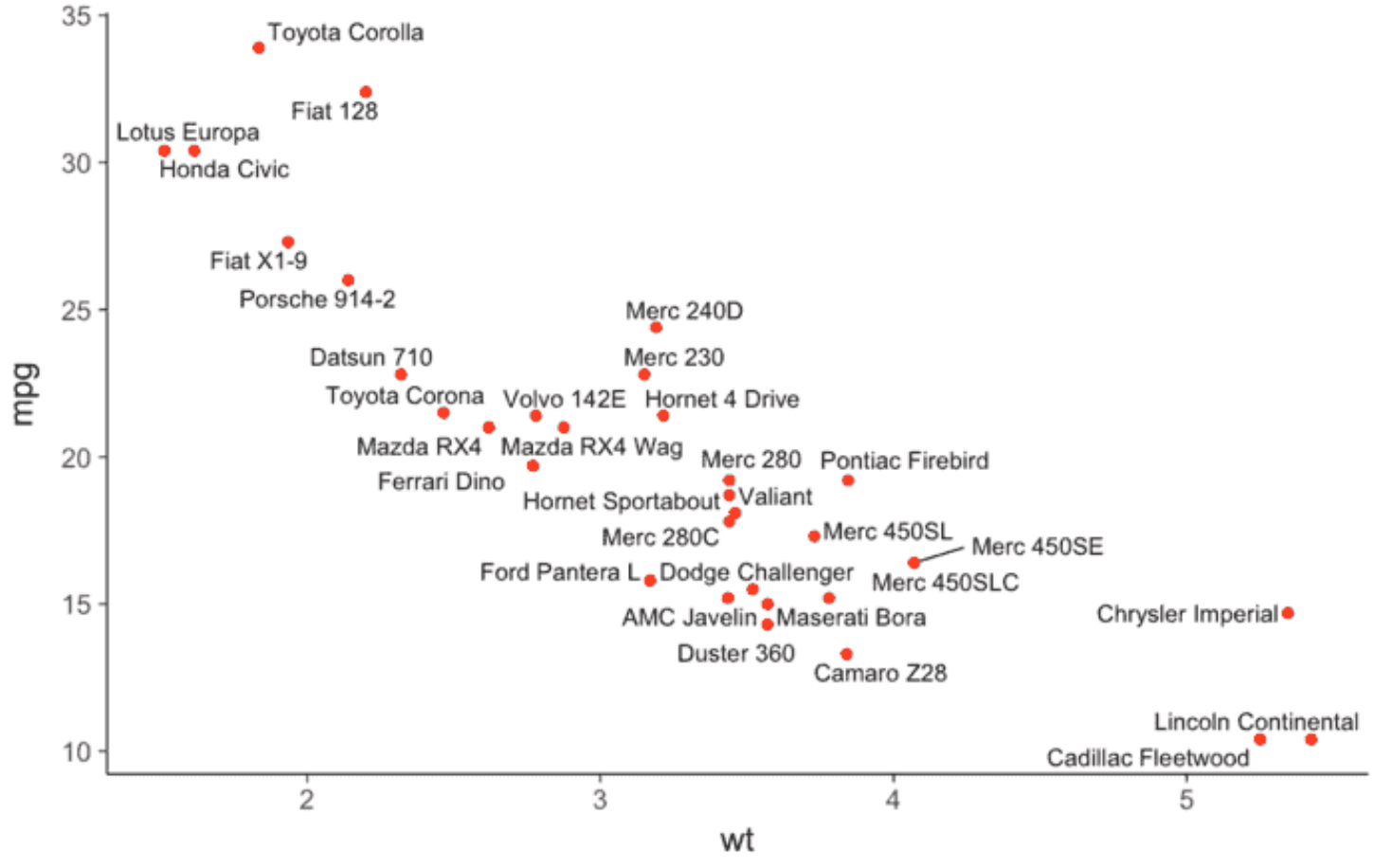


# Animated graphs

https://github.com/thomasp85/gganimate



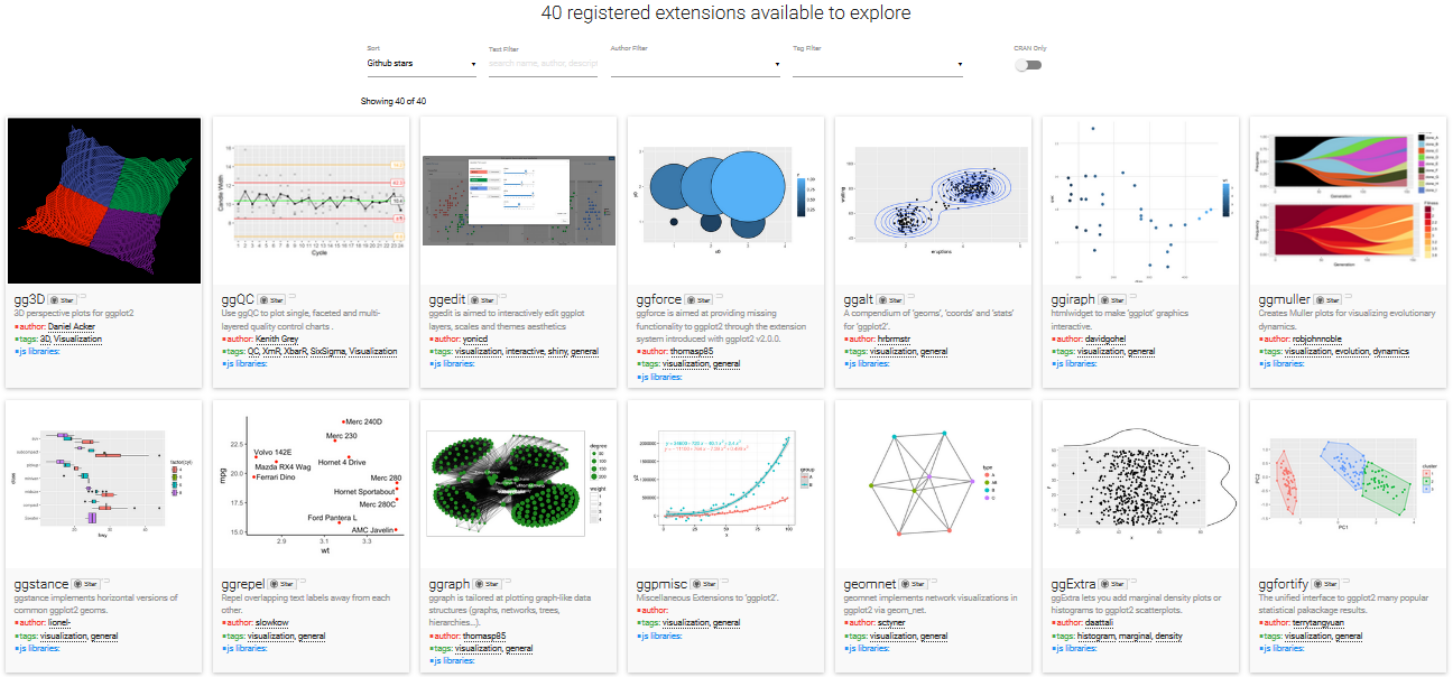
# Automatic label placement




https://cran.r-project.org/package=ggrepel

# Many extensions!

https://exts.ggplot2.tidyverse.org/gallery/

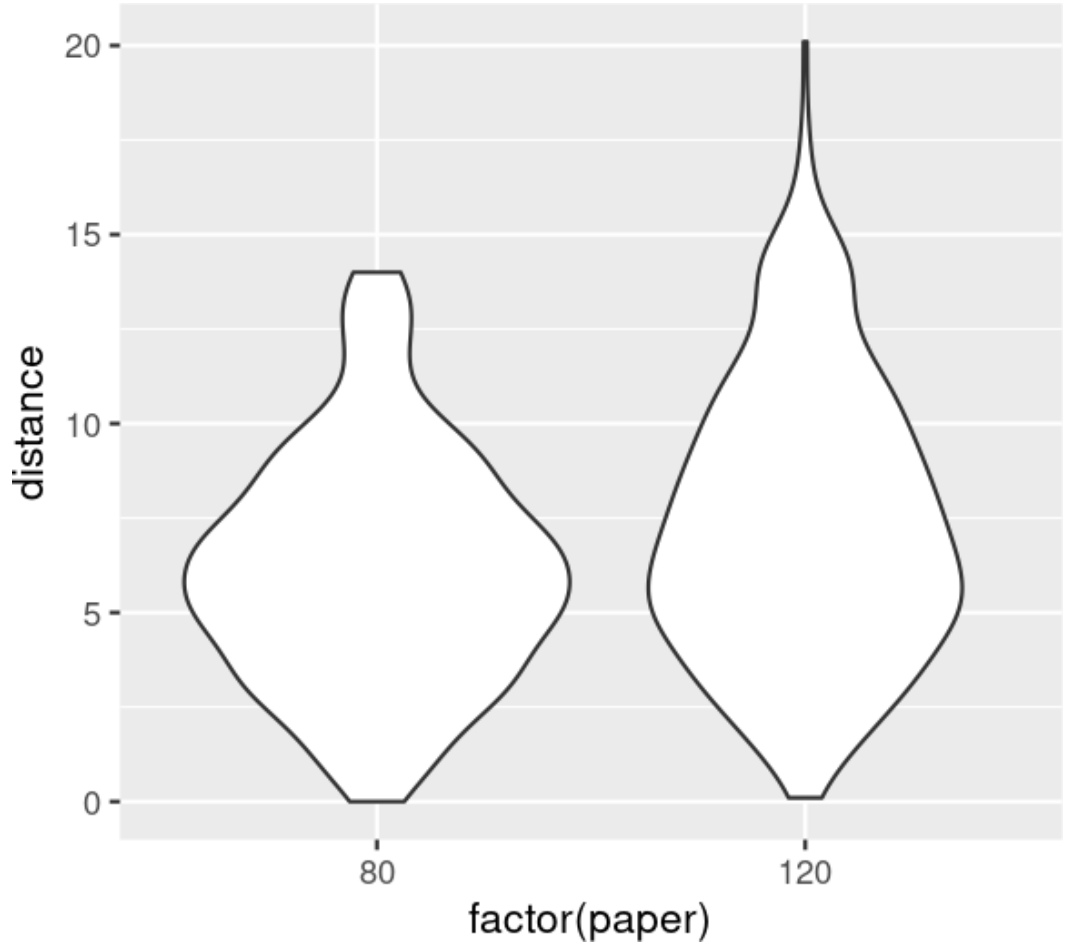


# Summary

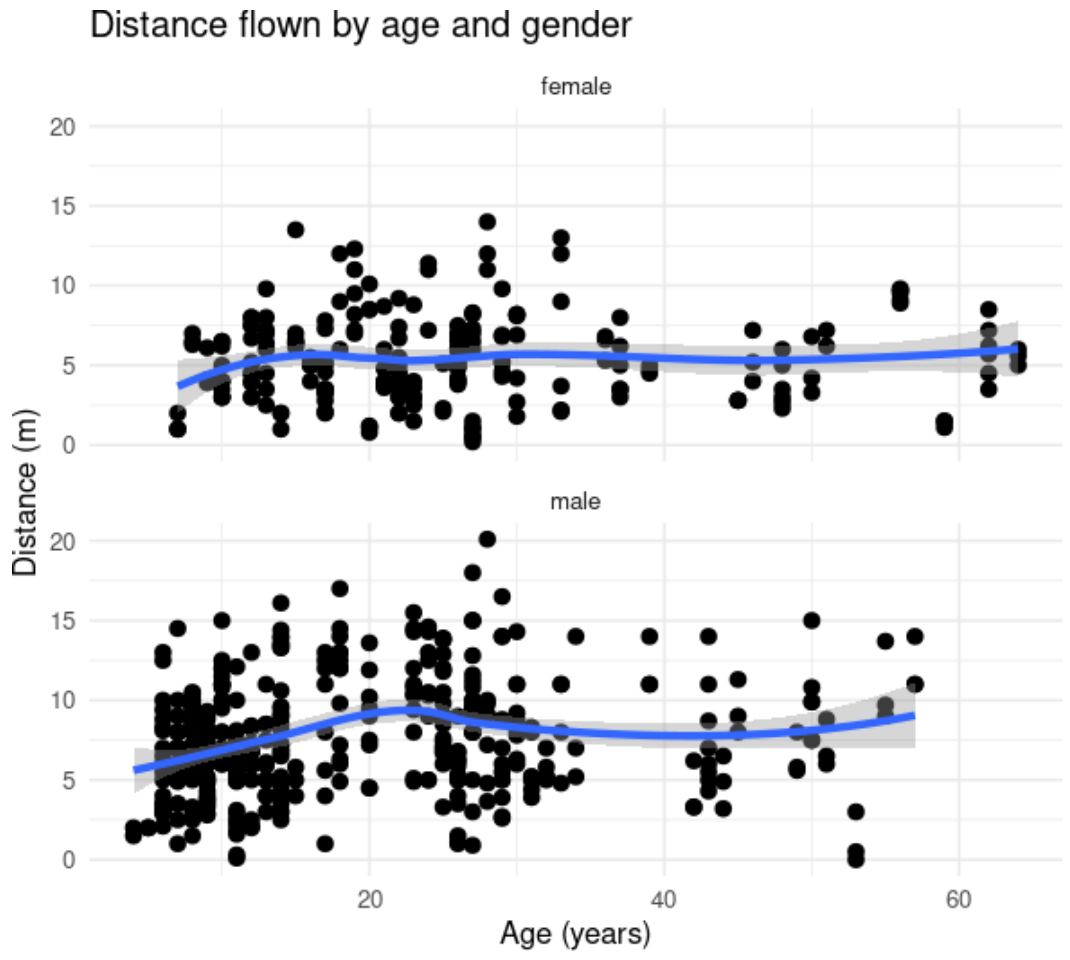
# Grammar of graphics

- **Data** (tidy data frame)
- **Layers** (*geoms*: points, lines, polygons…)
- **Aesthetics** mappings (x, y, size, colour…)
- **Scales** (colour, size, shape…)
- **Facets** (small multiples)
- **Themes** (appearance)
- **Coordinate system** (Cartesian, polar, map projections…)

# Exercise: make a plot like this one

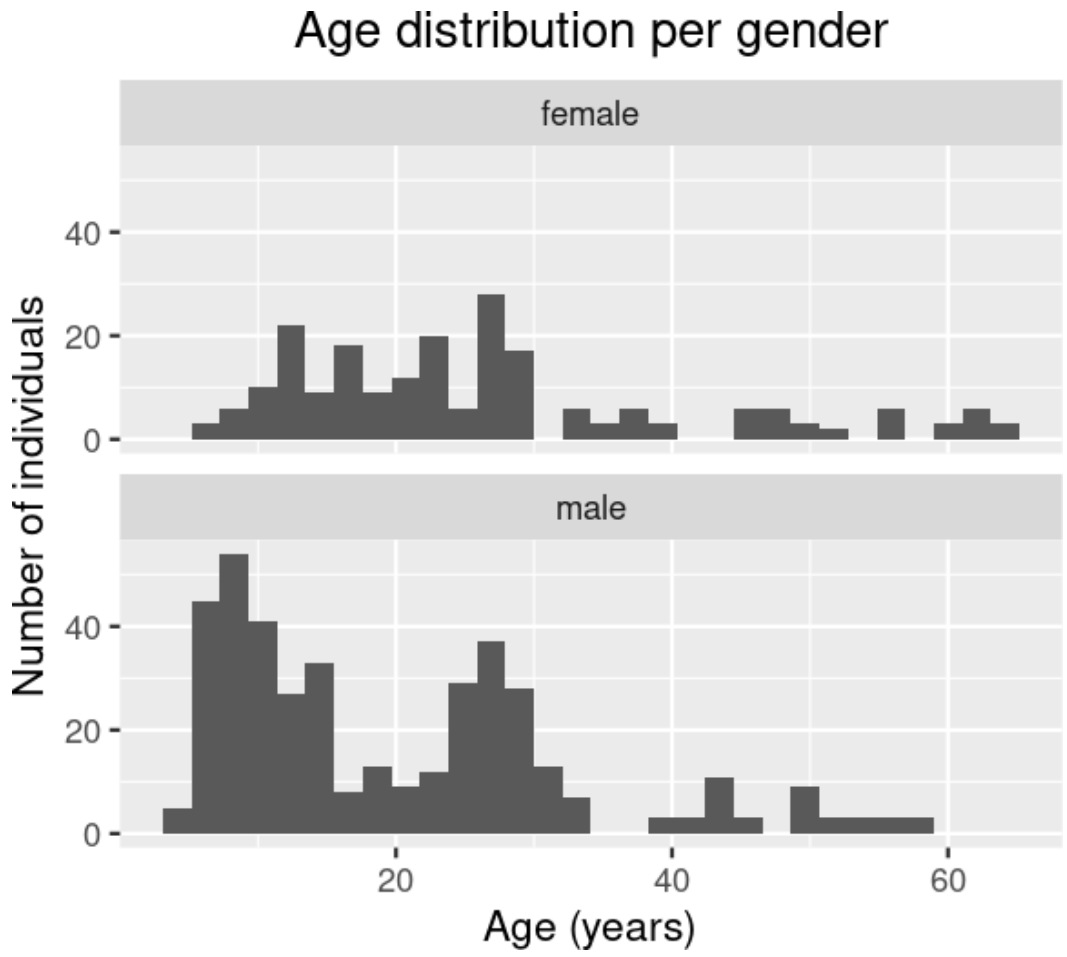


# Exercise: make a plot like this one



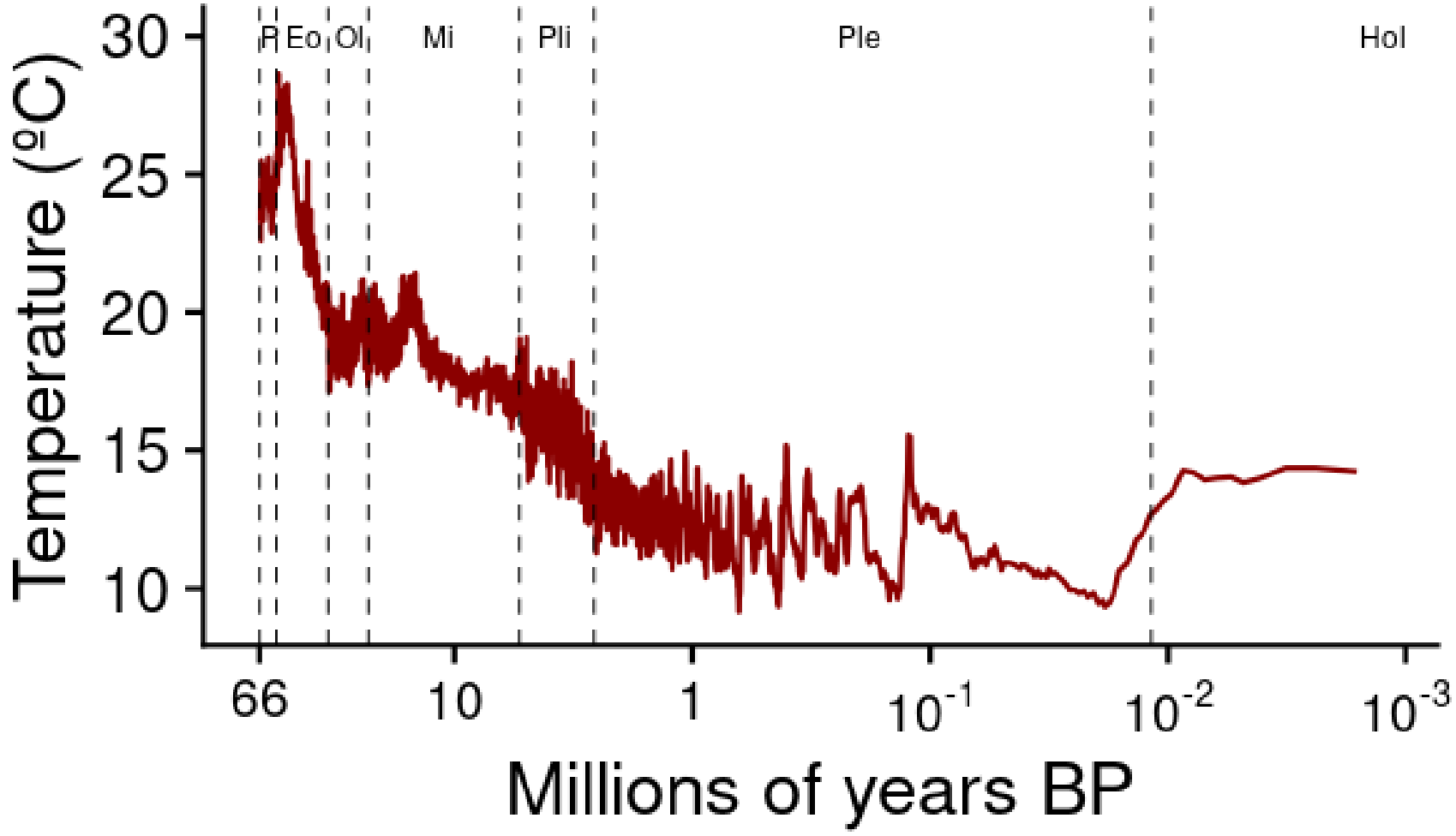
# Exercise: make a plot like this one

# Exercise: make a plot like this one



# Exercise: make a plot like this one

Data from http://www.columbia.edu/~mhs119/Sensitivity+SL+CO2/Table.txt



# Exercise: make a plot like this one

# END

Slides and source code available at https://github.com/Pakillo/ggplot-intro