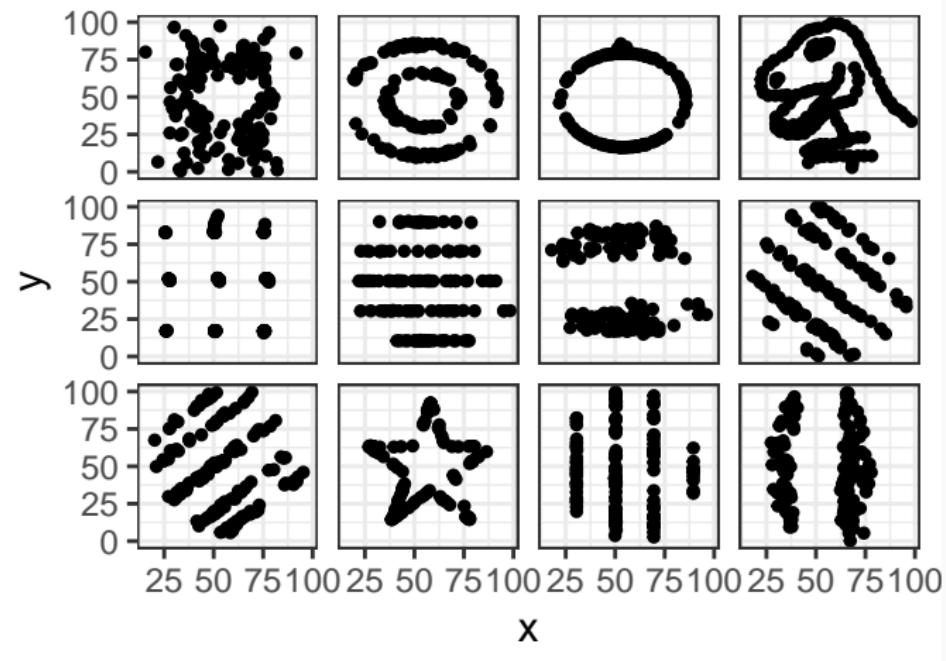


Data visualisation with ggplot2

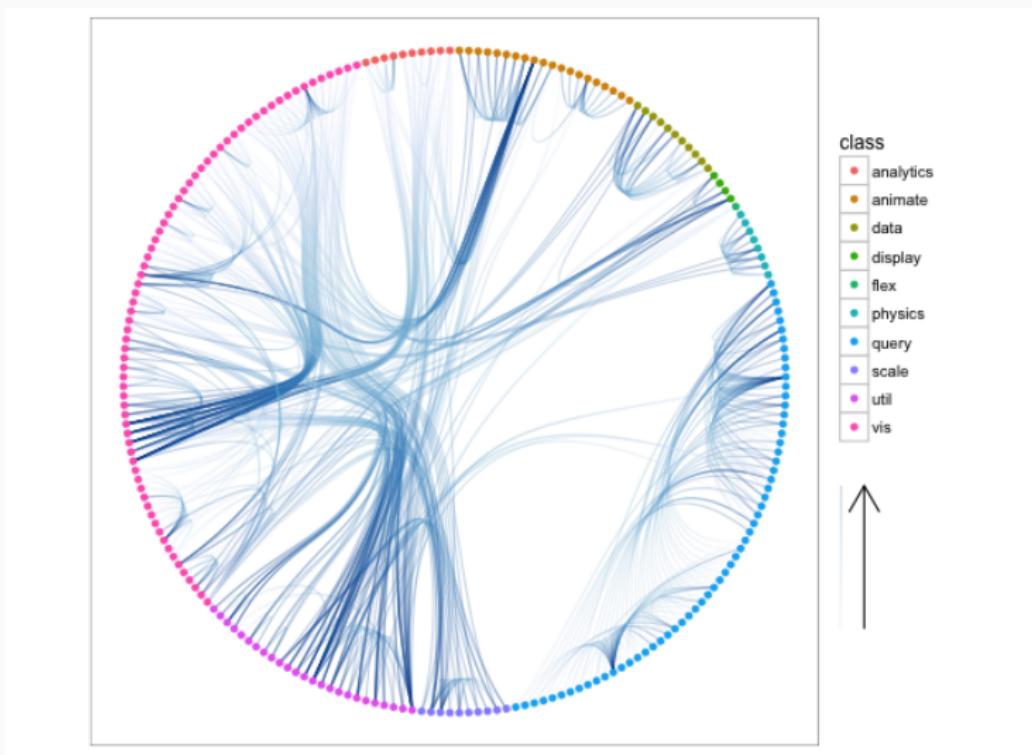
Francisco Rodriguez-Sanchez (@frod_san)

Always plot data!



<https://github.com/stephlocke/datasauRus>

Made with ggplot



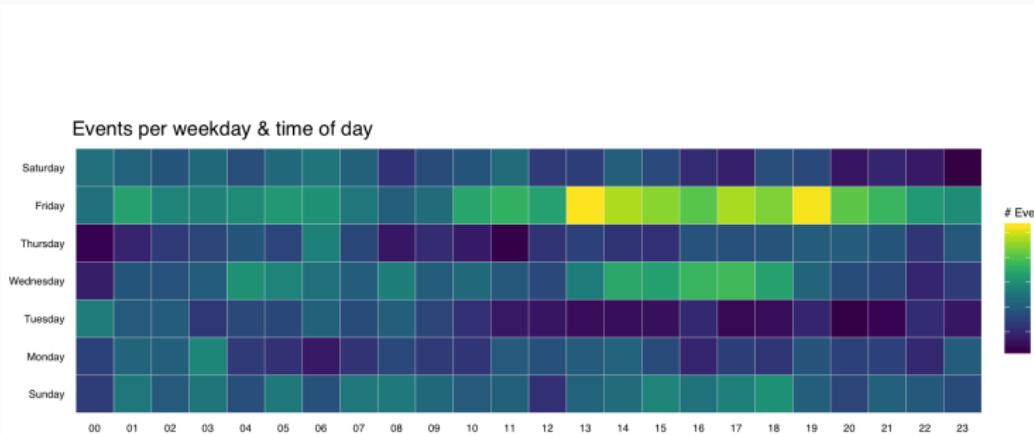
<https://github.com/thomasp85/ggraph>

Made with ggplot



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

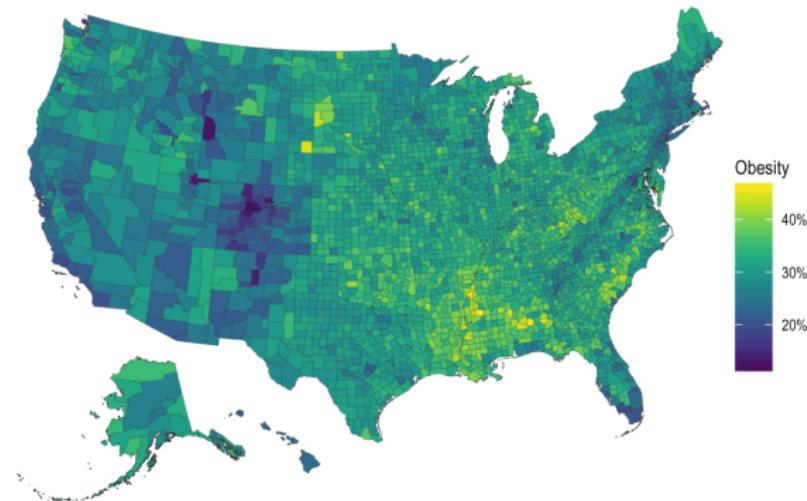
Made with ggplot



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

U.S. Obesity Rate by County (2012)

Content source: Centers for Disease Control and Prevention

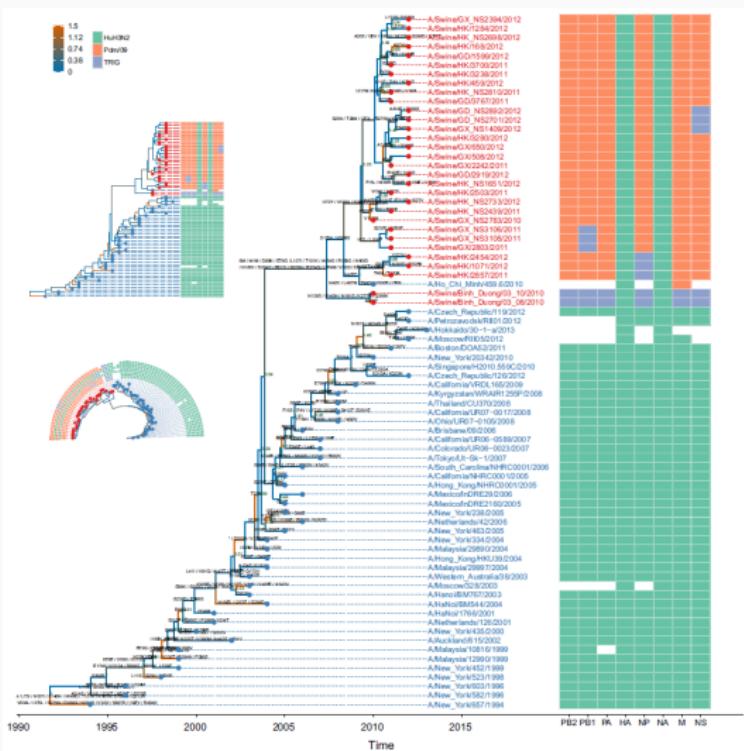


Data from http://www.cdc.gov/diabetes/atlas/countydata/County_ListofIndicators.html

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

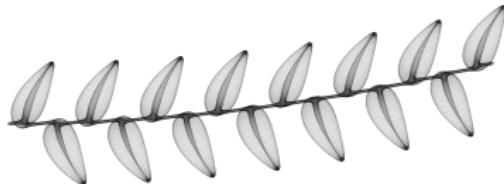
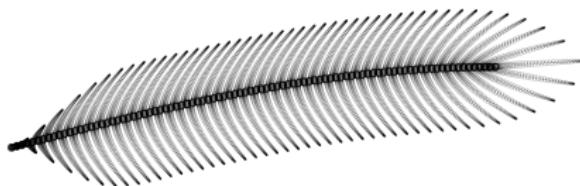
//rud.is/b/2016/03/29/easier-composite-u-s-choropleths-with-albersusa/

Made with ggplot



<https://guangchuangyu.github.io/ggtree/>

Made with ggplot



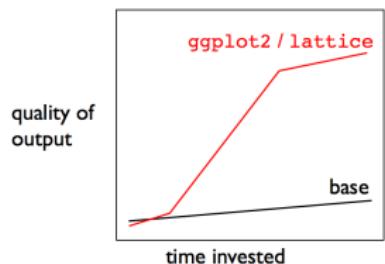
<https://github.com/marcusvolz/mathart>

Why ggplot

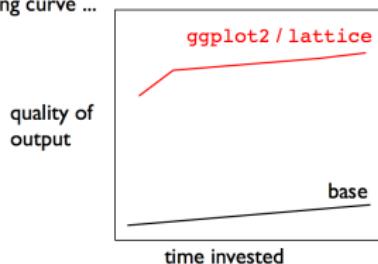
- Extremely powerful and flexible
- Consistent (grammar of graphics)
- Very powerful user base and active development

At the beginning it's hard, but then it pays off

week one



after you've climbed the steepest part of the learning curve ...



* figure is totally fabricated but, I claim, still true

* figure is totally fabricated but, I claim, still true

Source: <https://github.com/jennybc/ggplot2-tutorial>

Very good documentation and tutorials

- Official `ggplot2` documentation
- `ggplot2` book
- `R graphics cookbook` and `Cookbook for R`
- Beautiful plotting in R: A `ggplot2` cheatsheet
- Introduction to `ggplot2`
- Tutorial: `ggplot2`
- How to format plots for publication using `ggplot2`
- Visualising data with `ggplot2`
- Data Visualization with R and `ggplot2`
- `ggplot2` tutorial
- Data visualisation chapter in R for Data Science
- The complete `ggplot2` tutorial
- Data visualization: a practical introduction (K. Healy)
- Fundamentals of data visualization (C. Wilke)

Cheatsheet

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

Repos of figures + code

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

Find answers for all your questions in Stack Overflow



Search

ggplot2

36,854 results



The Practical Dev
@ThePracticalDev



The last programming book you'll ever need

Cutting corners to meet arbitrary management deadlines



Essential

Copying and Pasting
from Stack Overflow

Building a ggplot figure

Our example dataset: paper planes flying experiment

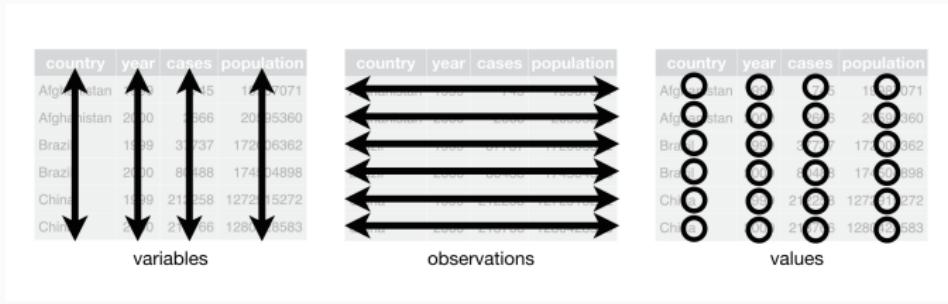
```
library(paperplanes)  
head(paperplanes)
```

id	hour	person	gender	age	plane	paper	distance
1	[17,18)	Roland	male	30	Standard80	80	7.8
2	[17,18)	Astrid	female	30	Concorde120	120	2.7
3	[17,18)	Roland	male	30	Standard120	120	9.2
4	[17,18)	Isabella	female	48	Standard120	120	6.0
5	[17,18)	Fabienne	female	17	Standard120	120	7.3
6	[17,18)	Fabienne	female	17	Standard120	120	7.8

Ensuring paper is factor, not numeric

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

Data must be a tidy data frame



```
tidy::gather(table4, key = "year", value = "cases", "1999", "2000")
```

The diagram shows the transformation of a wide data frame (`table4`) into a long/tidy data frame using the `tidy::gather` function.

`table4` (wide format):

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

`tidy::gather(table4, key = "year", value = "cases", "1999", "2000")` (tidy format):

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

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

Calling ggplot

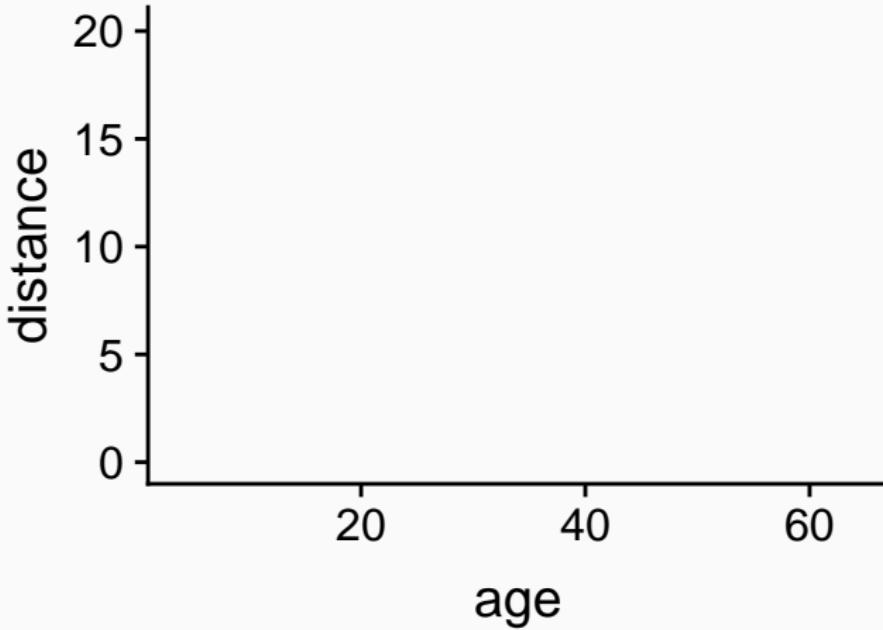
```
library(ggplot2)  
ggplot(paperplanes)
```

```
ggplot(paperplanes)
```

First argument is a tidy data frame

What variables as axes?

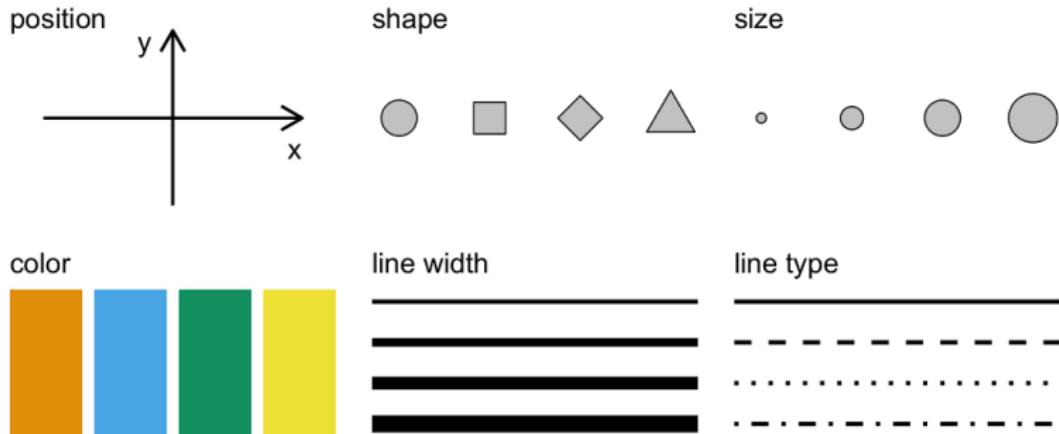
```
ggplot(paperplanes) +  
  aes(x = age, y = distance)
```



Note syntax: + followed by new line

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

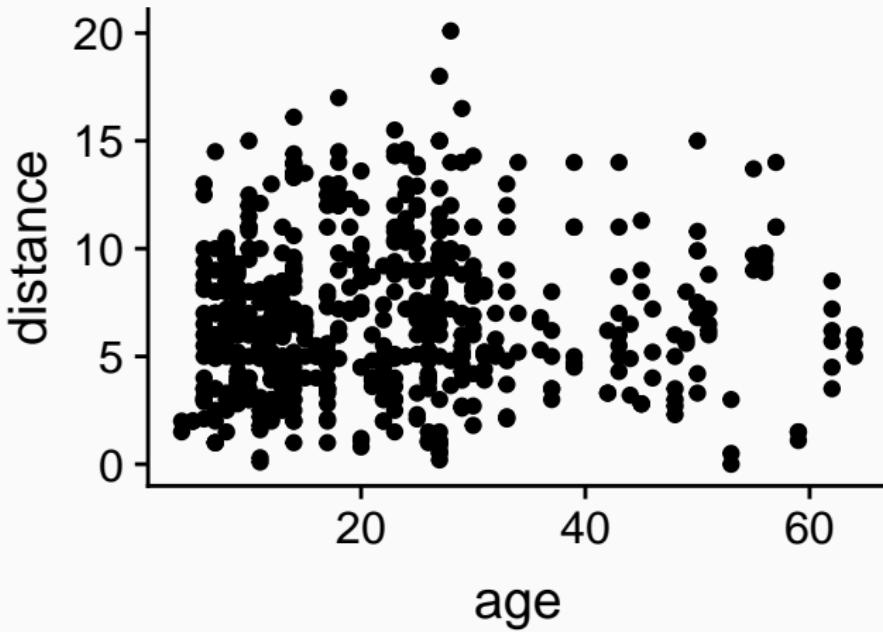
Aesthetics (`aes`) map data variables (`age`, `distance`) to graphic elements (`axes`)



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

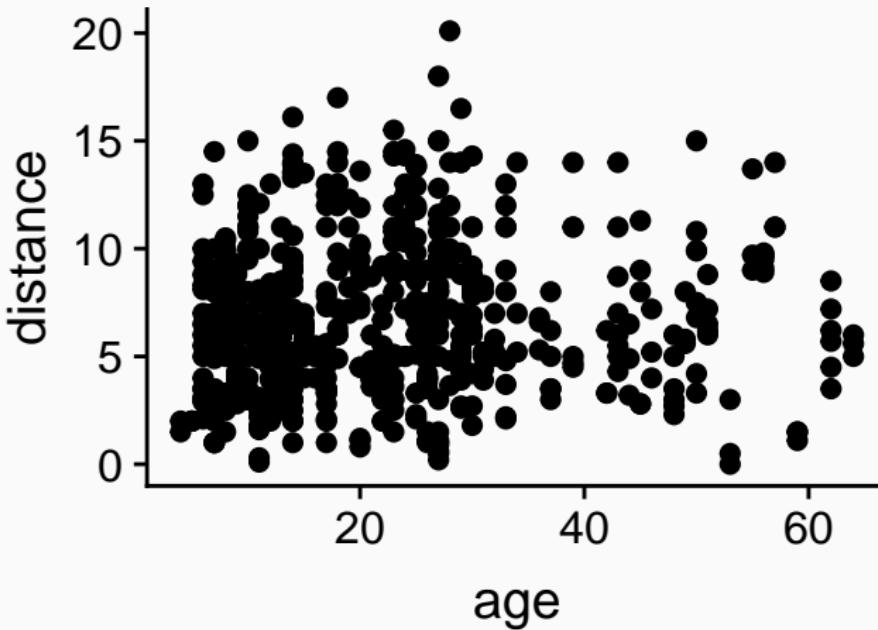
Adding layers (geoms)

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



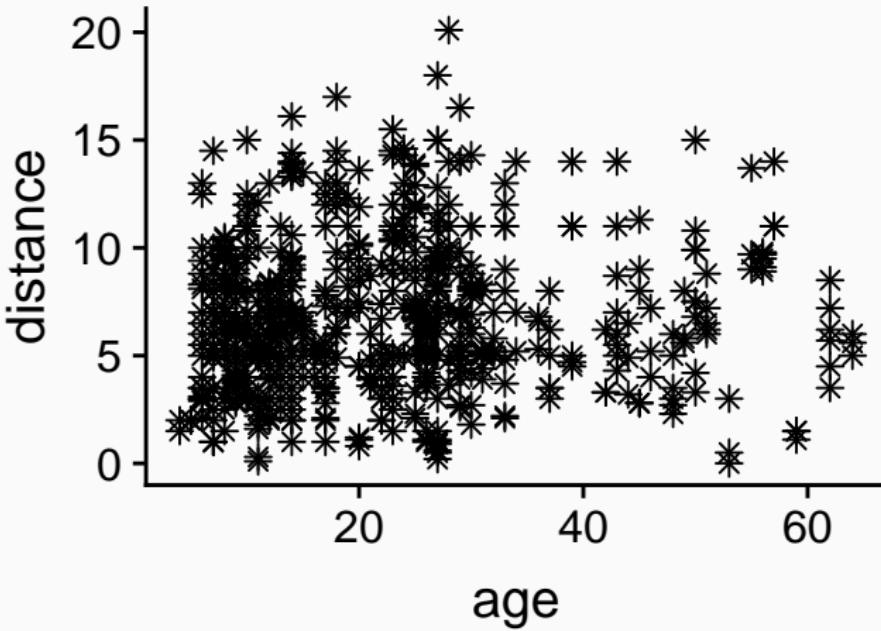
Changing point size and type

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



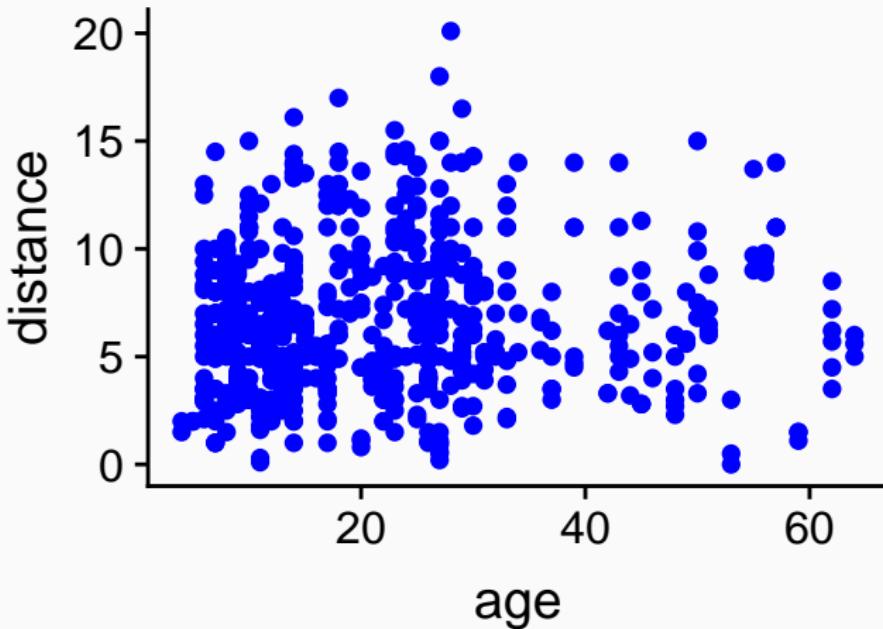
Changing point size and type

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



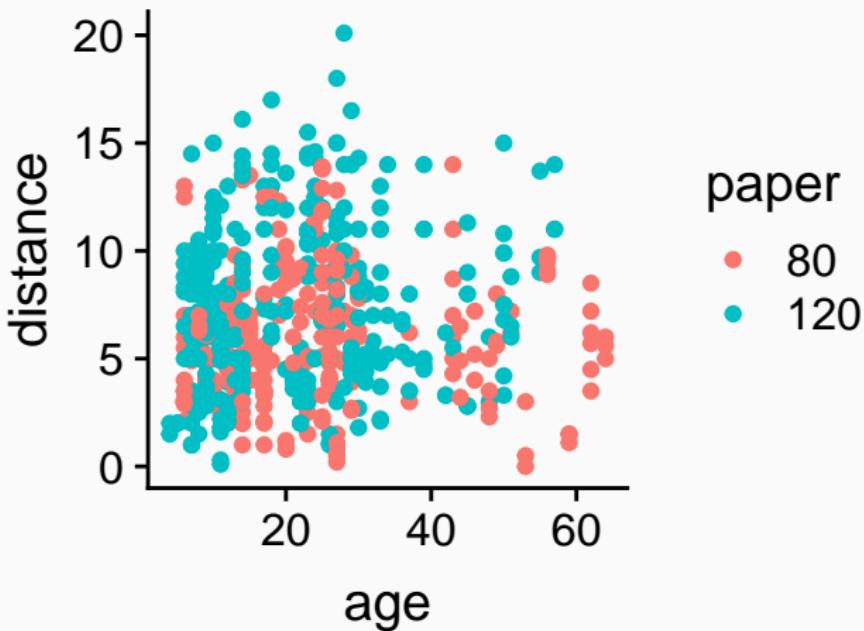
Changing point size and type

```
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))
```



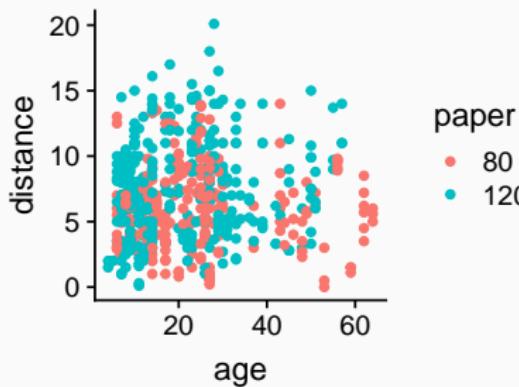
Note difference between

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

```
geom_point(aes(colour = gender))
# colour maps a *variable* (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)
```

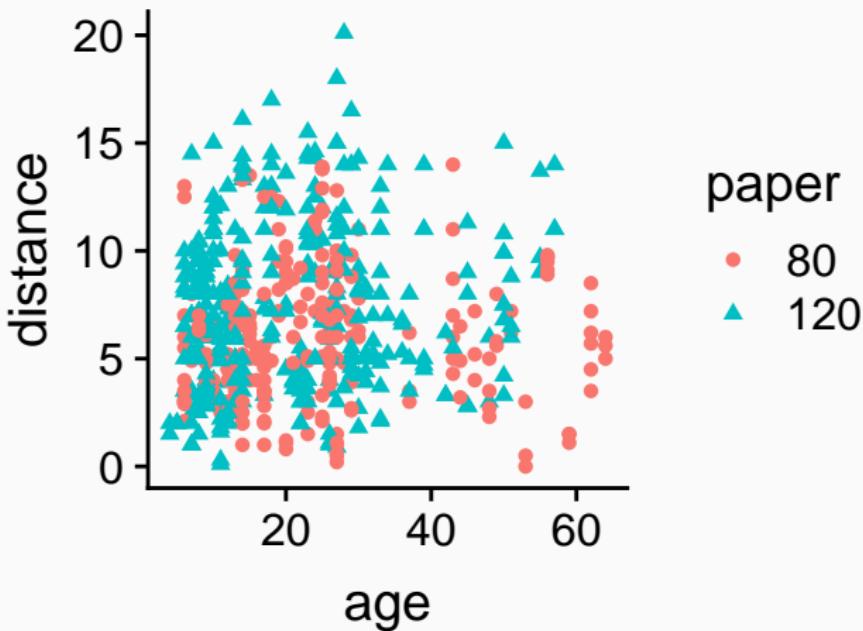
*Error in layer(data = data, mapping =
mapping, stat = stat, geom =
GeomPoint, : object 'paper' not found*

'paper' is a variable in dataframe

Must use aes

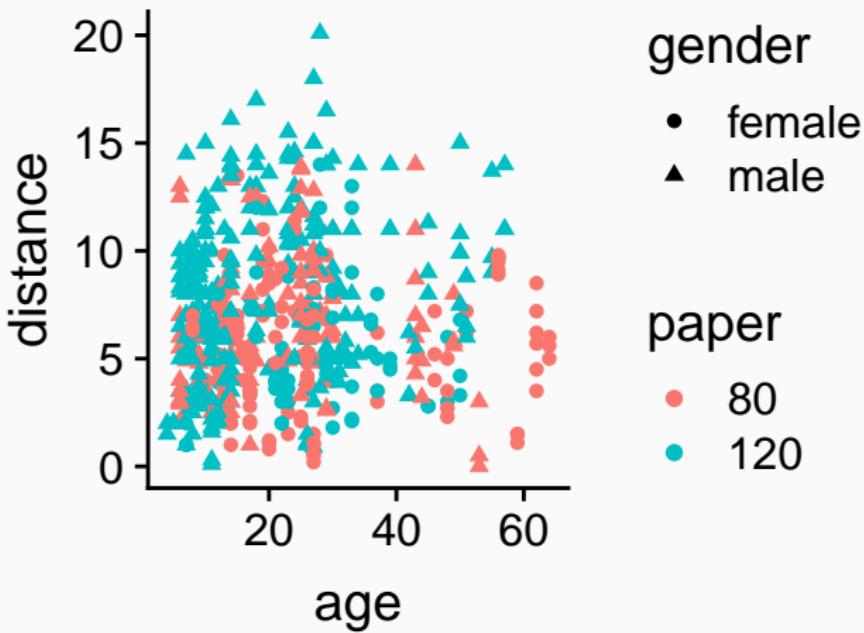
Map geom aesthetics (colour, shape) to variable

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



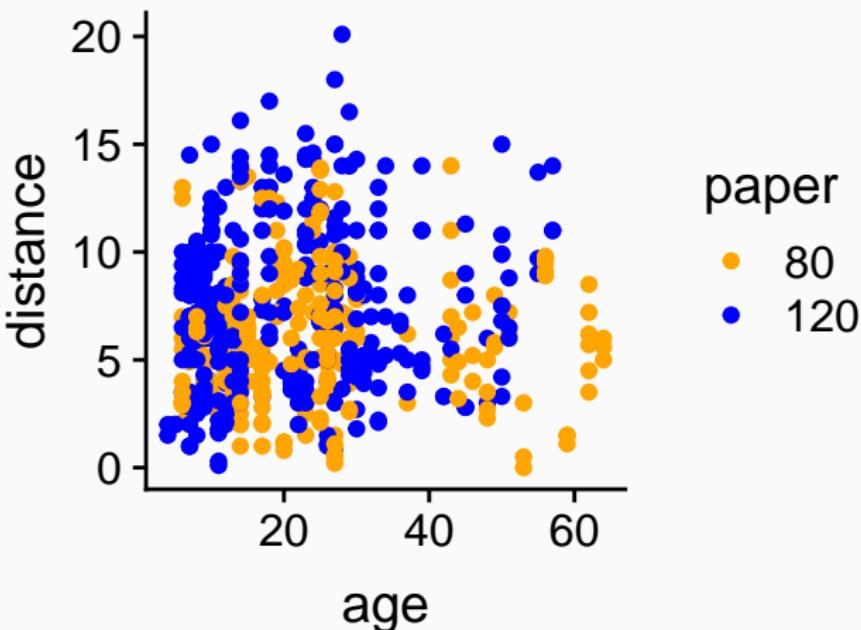
Map geom aesthetics (colour, shape) to variable

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



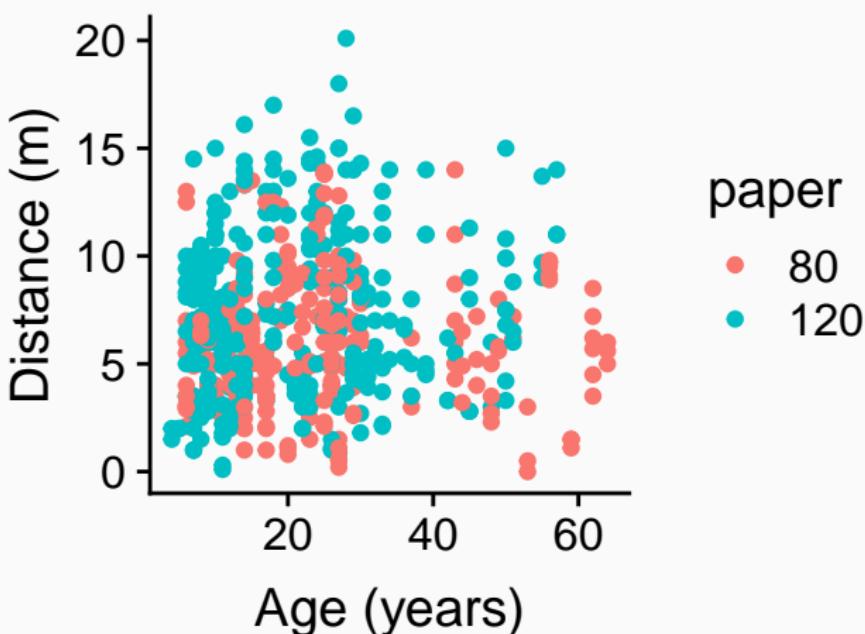
Change colour scale

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



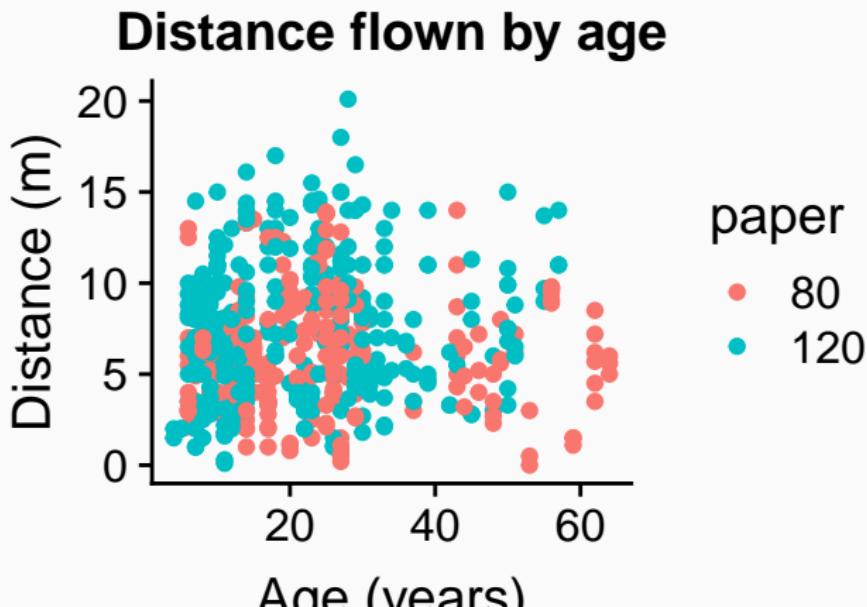
Change axis labels: `xlab` & `ylab`

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



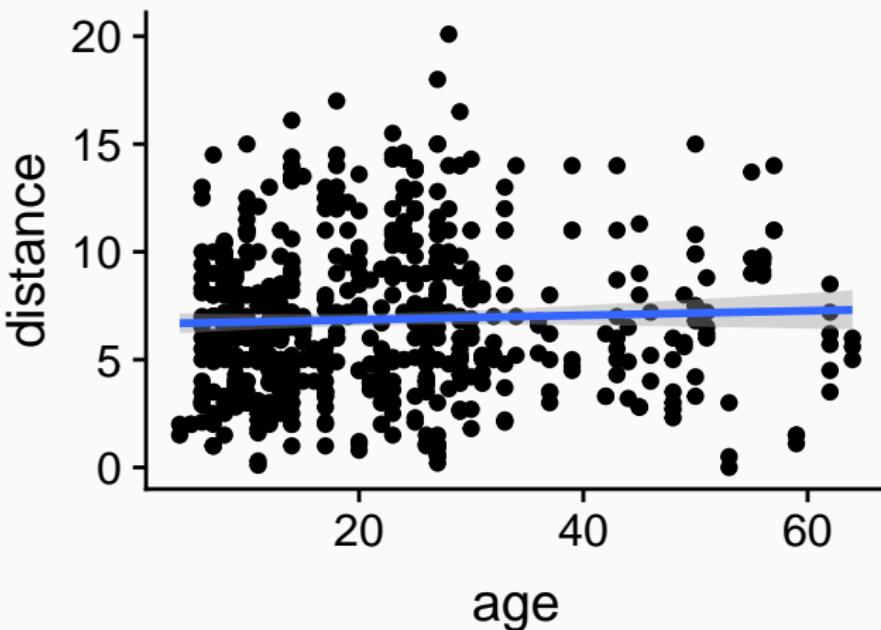
Set title

```
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")
```



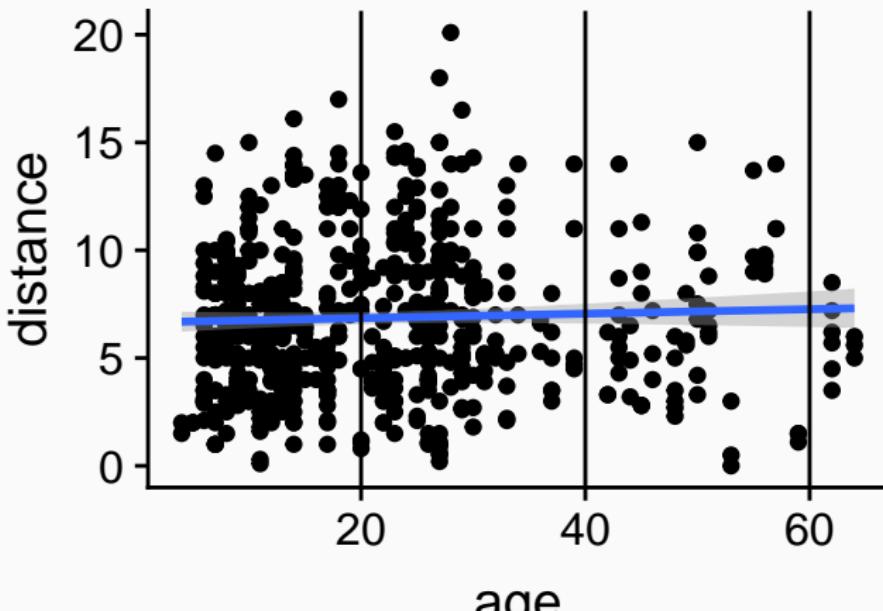
Adding more layers

```
ggplot(paperplanes) +  
  aes(x = age, y = distance) +  
  geom_point() +  
  geom_smooth(method = "lm")
```



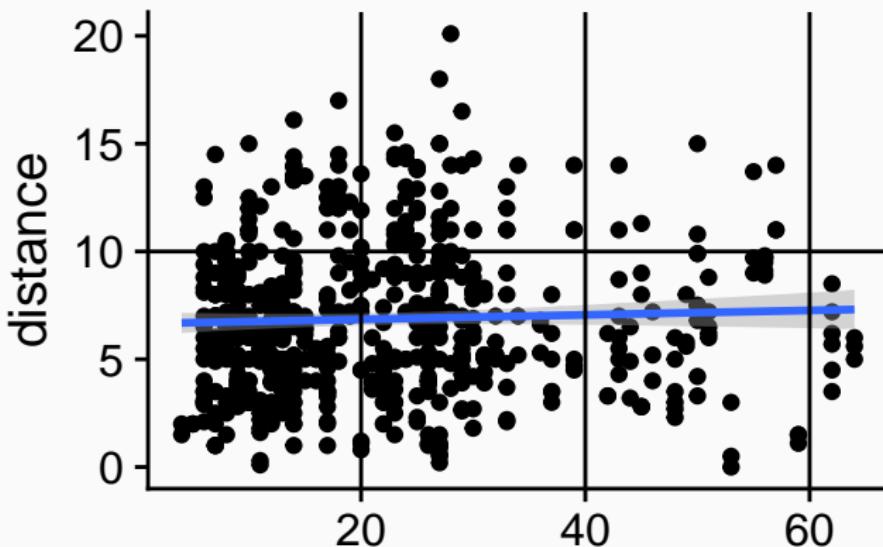
Adding more layers

```
ggplot(paperplanes) +  
  aes(x = age, y = distance) +  
  geom_point() +  
  geom_smooth(method = "lm") +  
  geom_vline(xintercept = c(20, 40, 60))
```



Adding more layers

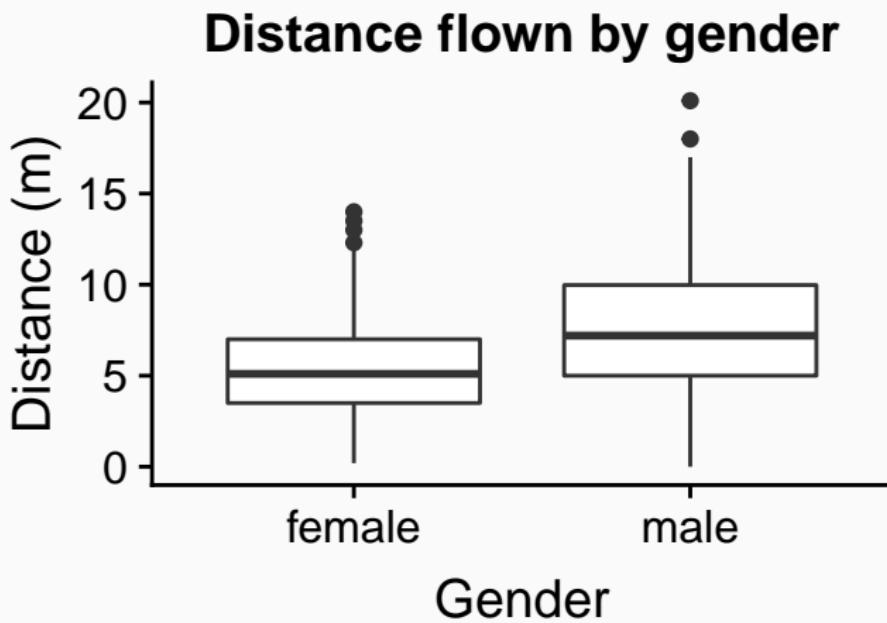
```
ggplot(paperplanes) +  
  aes(x = age, y = distance) +  
  geom_point() +  
  geom_smooth(method = "lm") +  
  geom_vline(xintercept = c(20, 40, 60)) +  
  geom_hline(yintercept = 10)
```



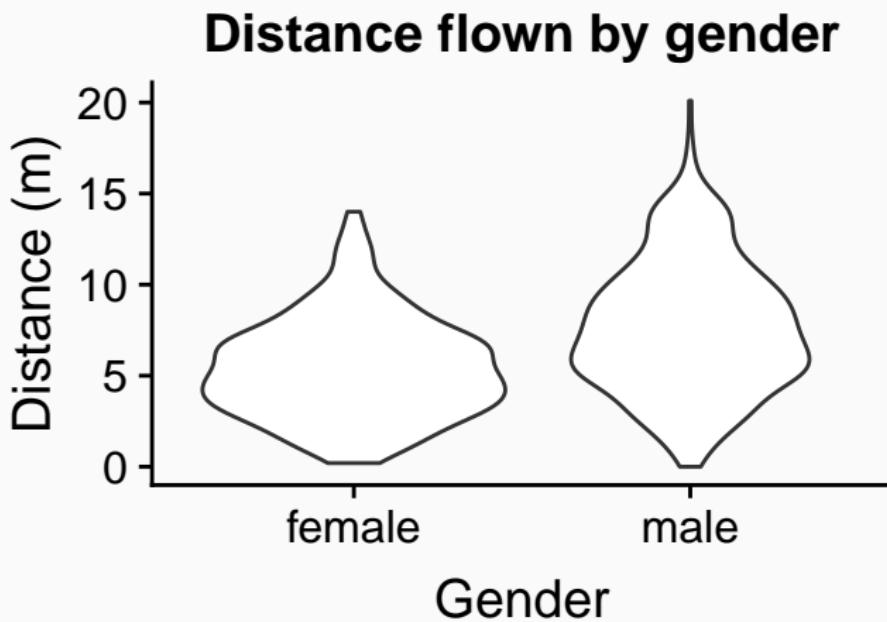
Summary

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

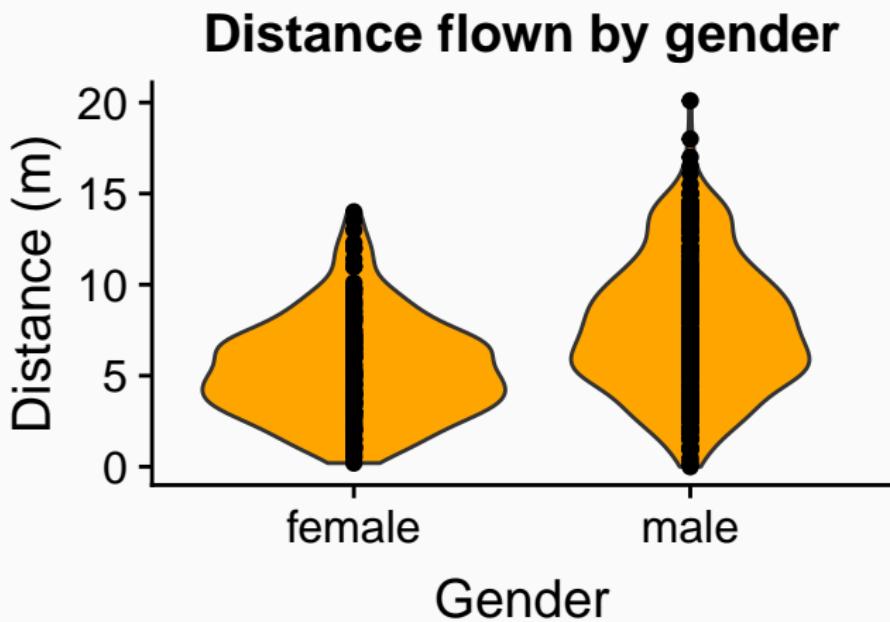
Exercise: Make a plot like this one



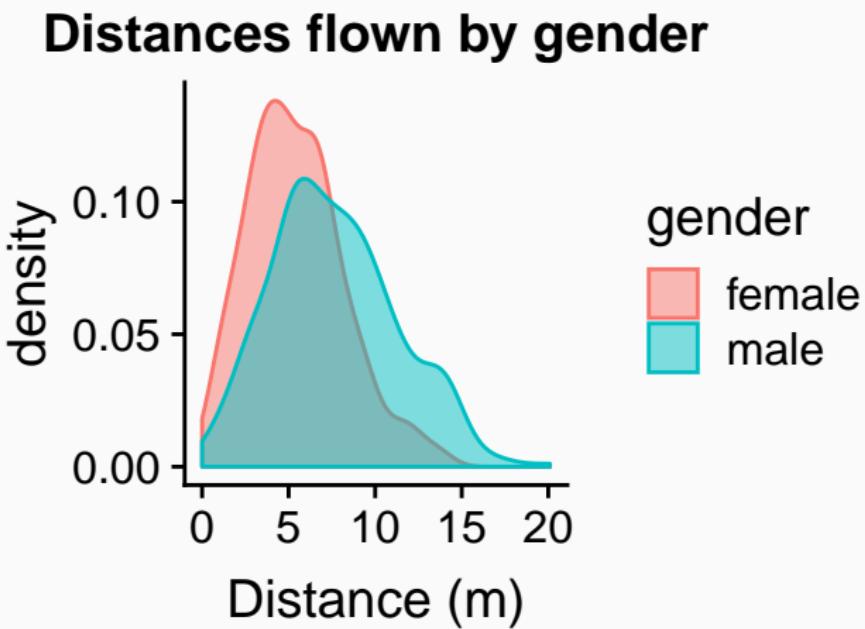
Exercise: Make a plot like this one



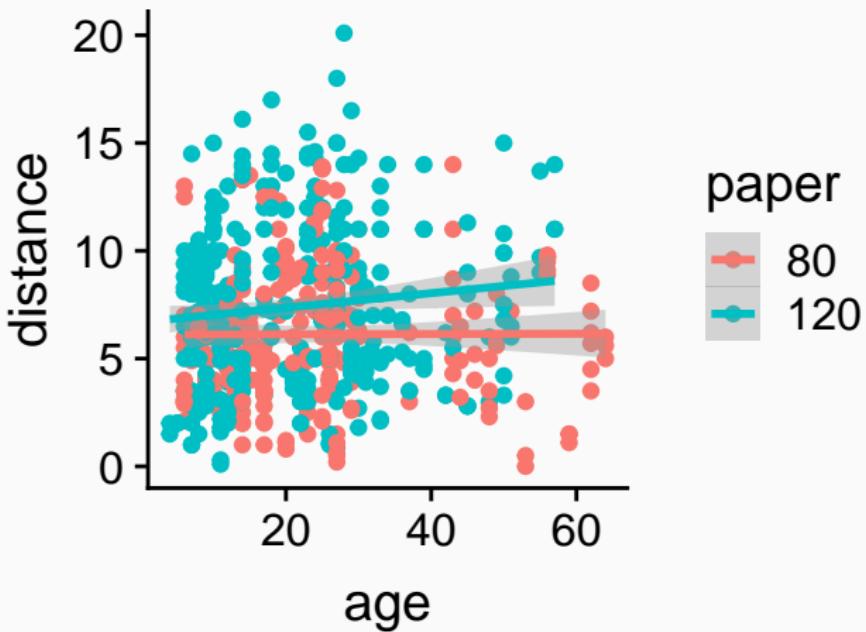
Exercise: Make a plot like this one



Exercise: Make a plot like this one



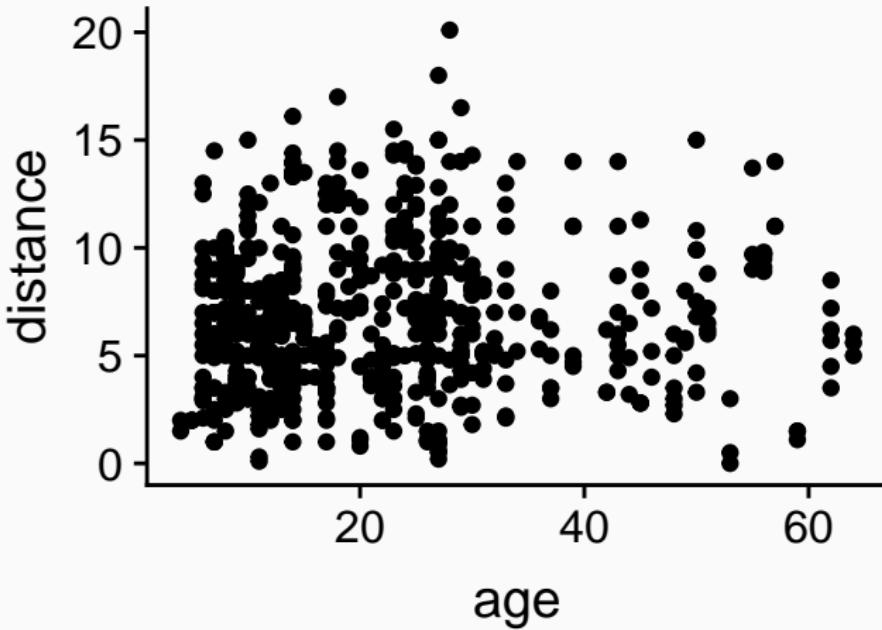
Exercise: Make a plot like this one



**ggplot2 figures can be assigned to R
objects**

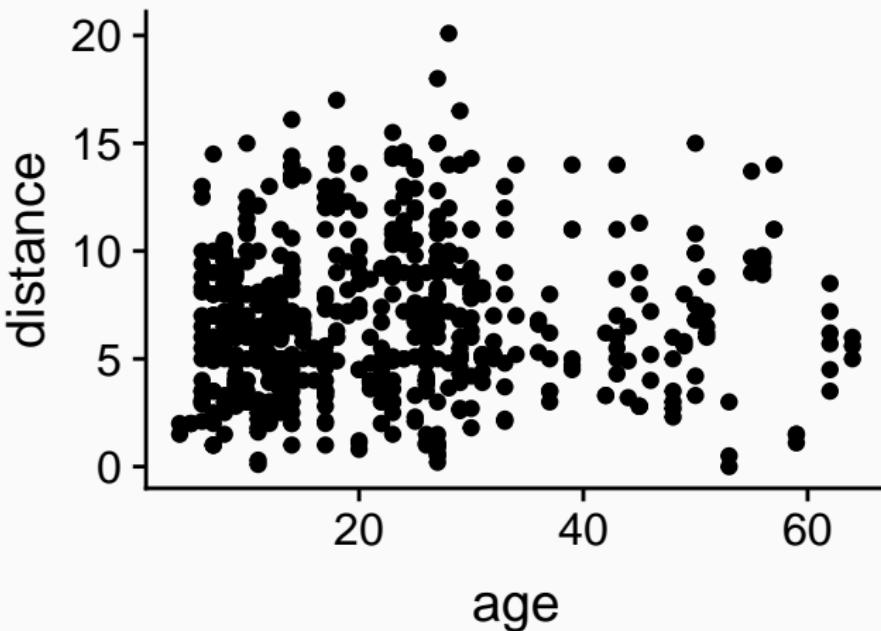
Assigning ggplot objects

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



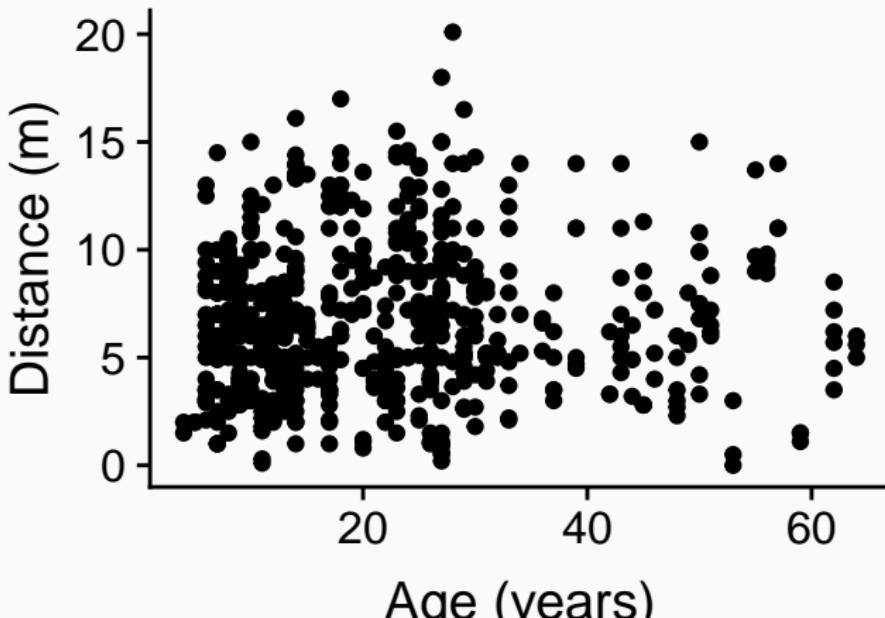
Assigning ggplot objects

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



Assigning ggplot objects

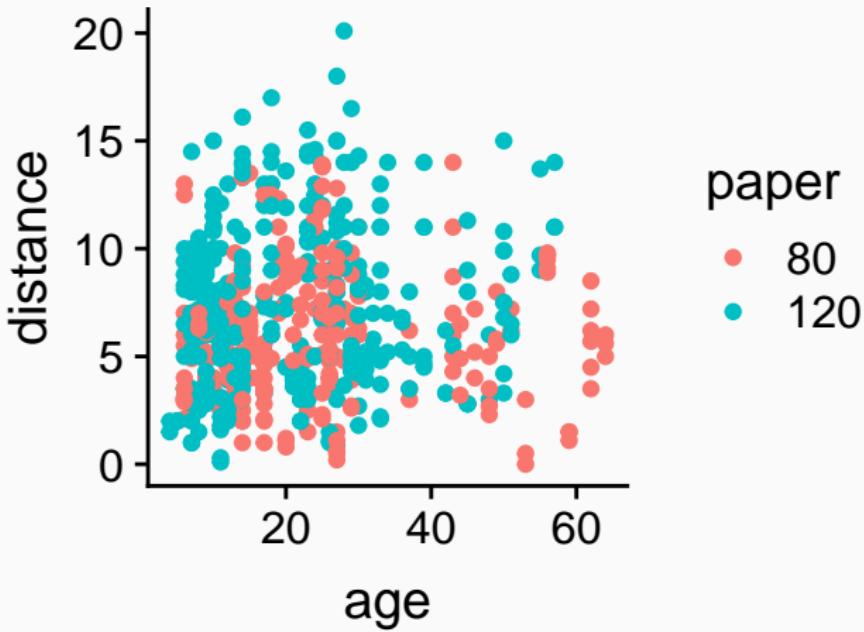
```
baseplot <- ggplot(paperplanes) +
  aes(x = age, y = distance)
scatterplot <- baseplot + geom_point()
labelled <- scatterplot + labs(x = "Age (years)", y = "Distance (m)")
labelled
```



Themes: changing plot appearance

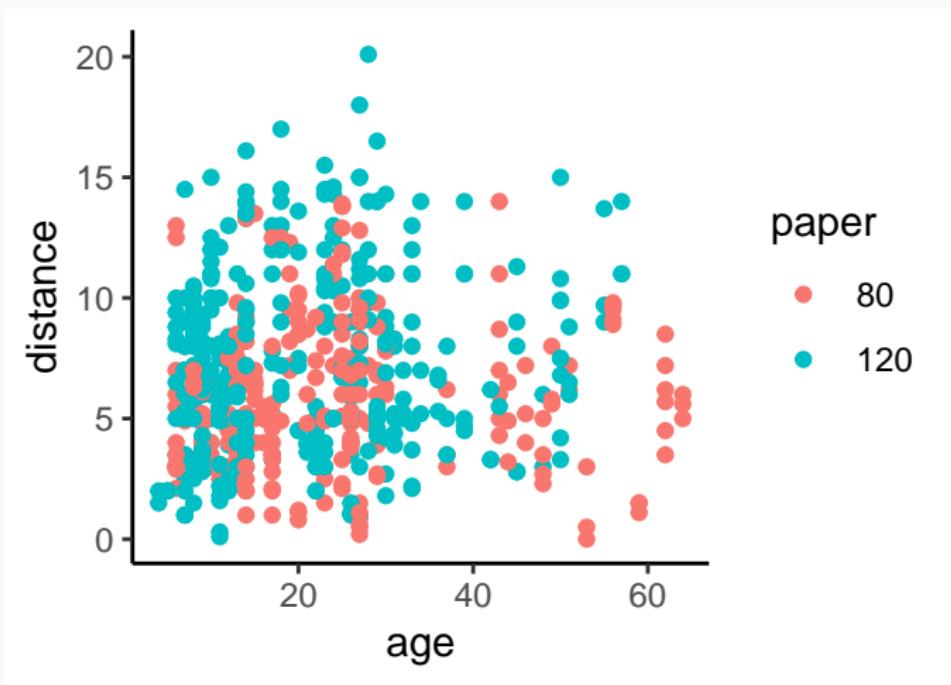
myplot

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



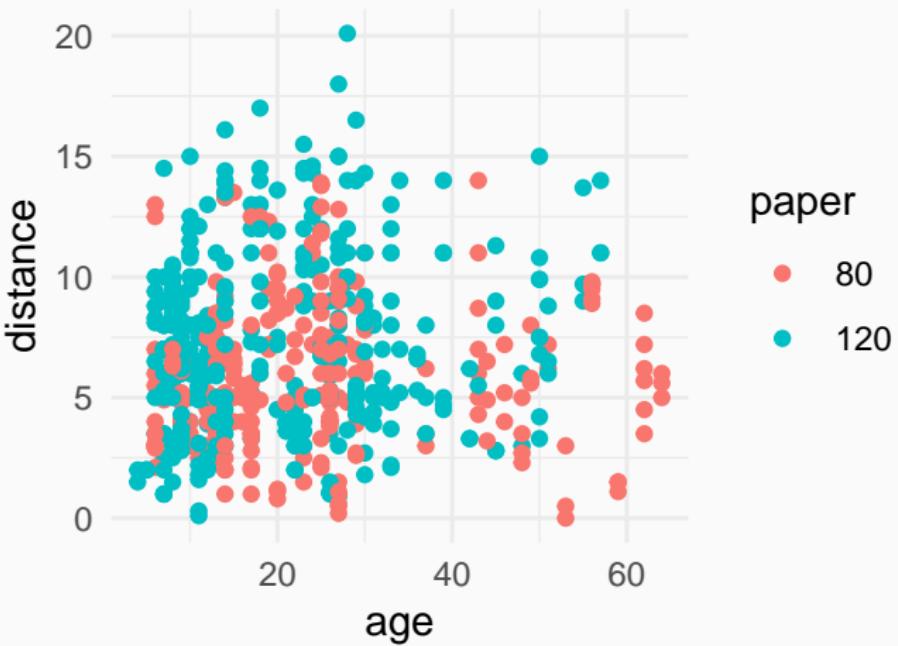
theme_classic

```
myplot + theme_classic()
```



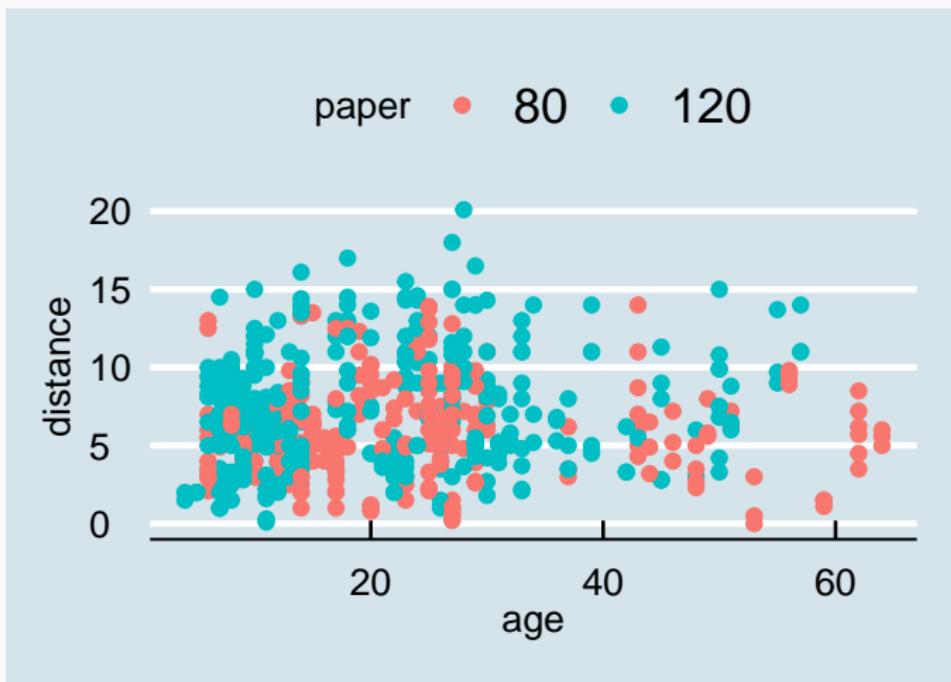
theme_minimal

```
myplot + theme_minimal()
```



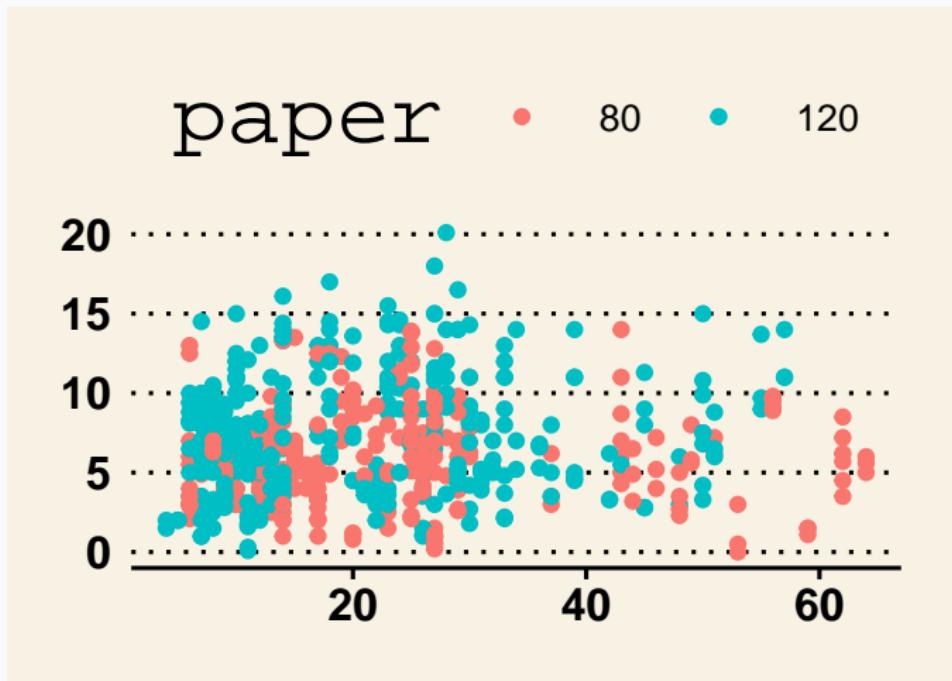
Lots of themes out there

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



Lots of themes out there

myplot + theme_wsj()

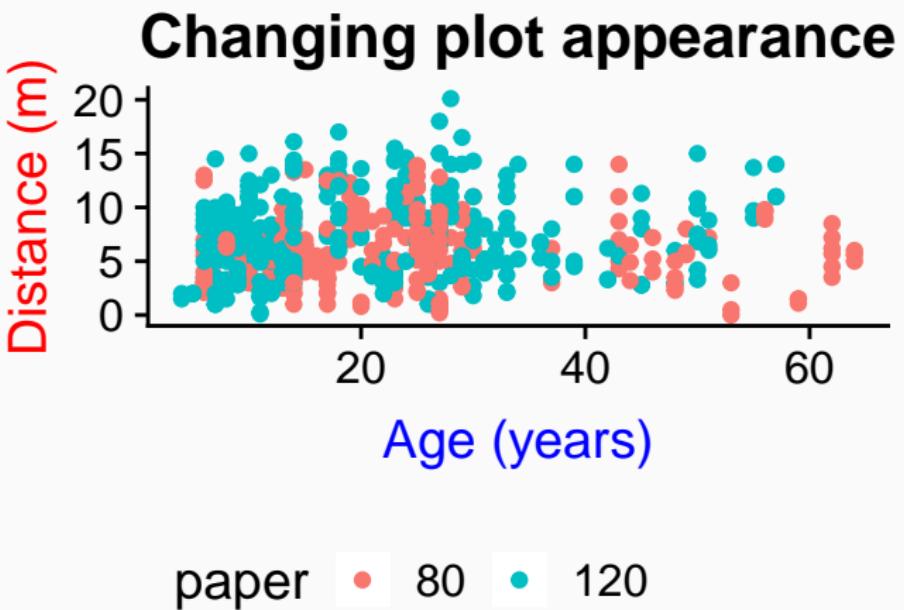


Editing themes

?theme

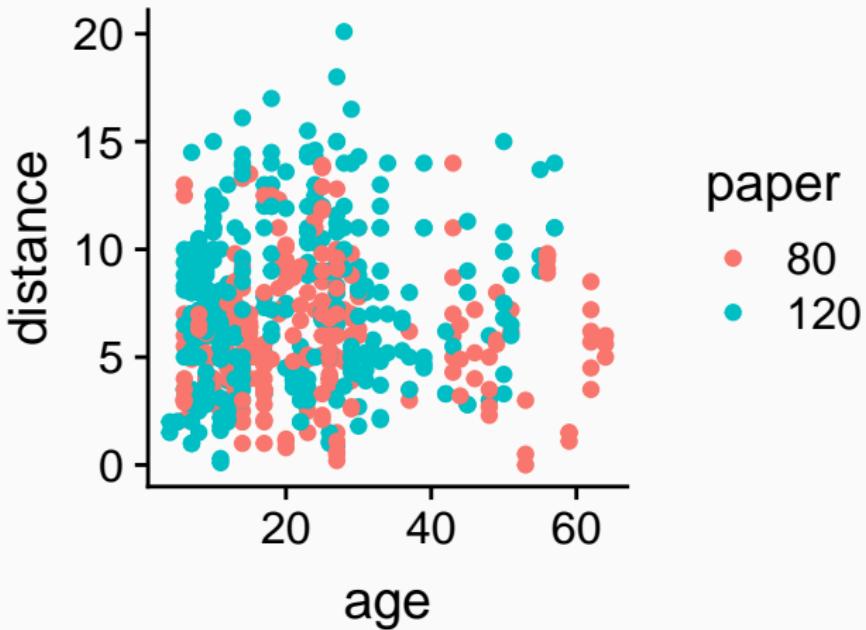
- element_blank
- element_text
- element_line
- element_rect (borders & backgrounds)

Exercise: make a plot like this one



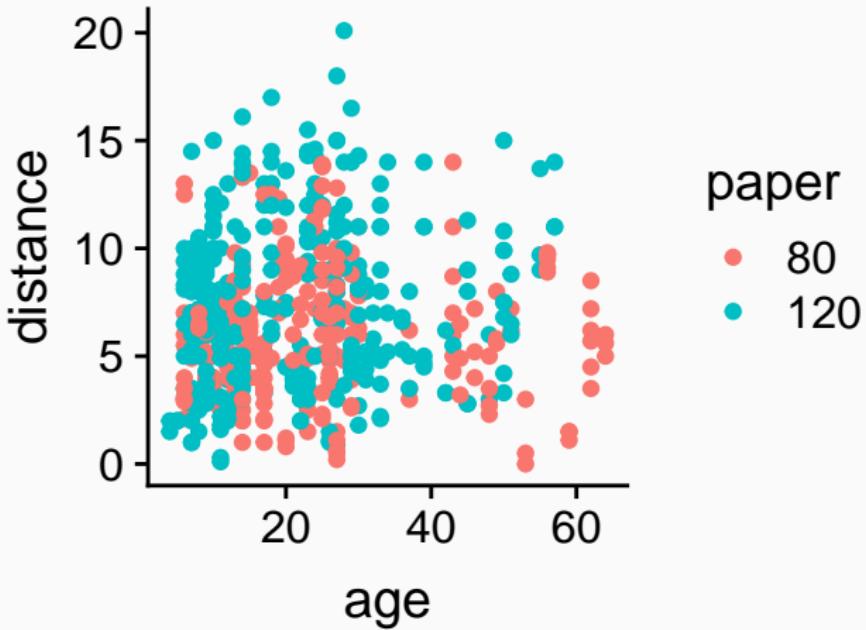
Easily changing appearance with ggthemeassist (Rstudio addin)

<https://github.com/calligross/ggthemeassist>



Easily changing appearance with ggedit

<https://github.com/metrumresearchgroup/ggedit>





Trevor A. Branch
@TrevorABranch

Follow

My rule of thumb: every analysis you do on a dataset will have to be redone 10–15 times before publication. Plan accordingly. #Rstats

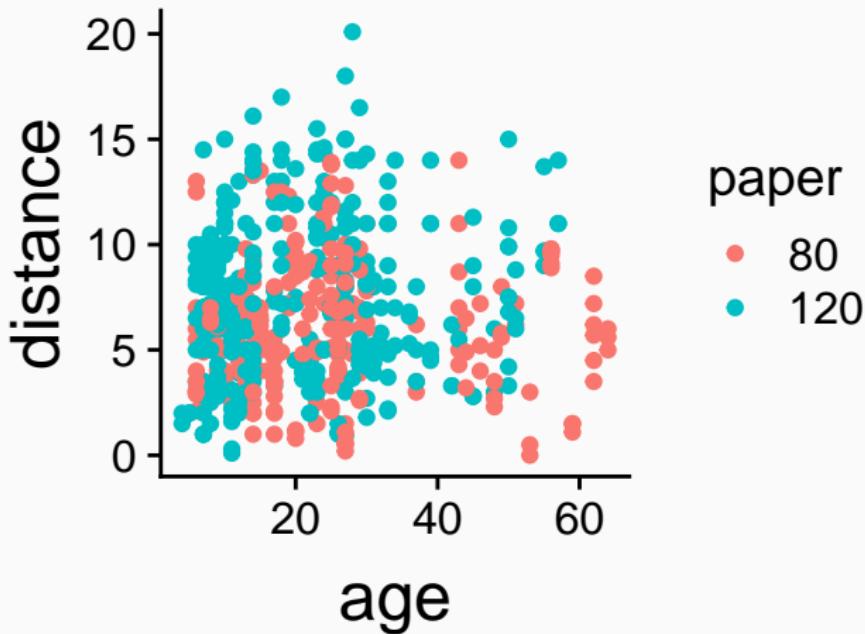
<http://mbjoseph.github.io/2015/02/26/plotting.html>

serialmentor.com/dataviz/choosing-the-right-visualization-software.html

Think twice before editing plots out of R

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

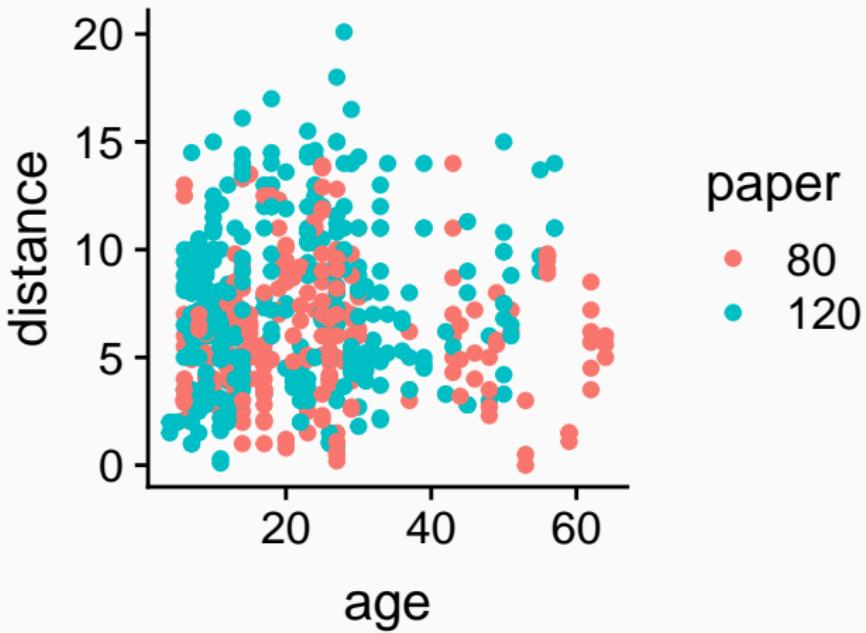
```
myplot +  
  theme(axis.title = element_text(size = 18))
```



Publication-quality plots

```
library(cowplot)
```

```
myplot
```



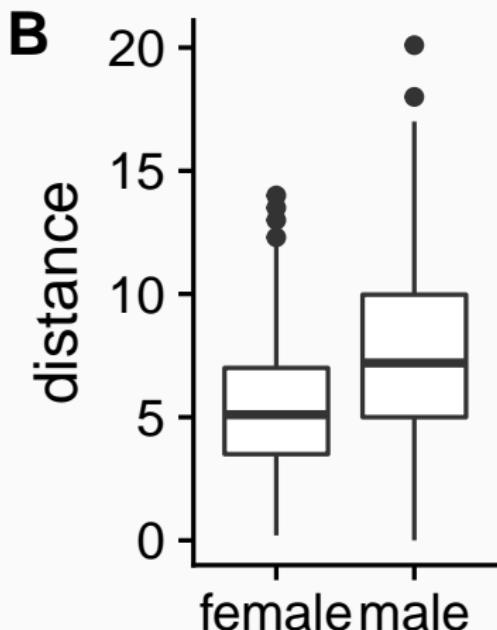
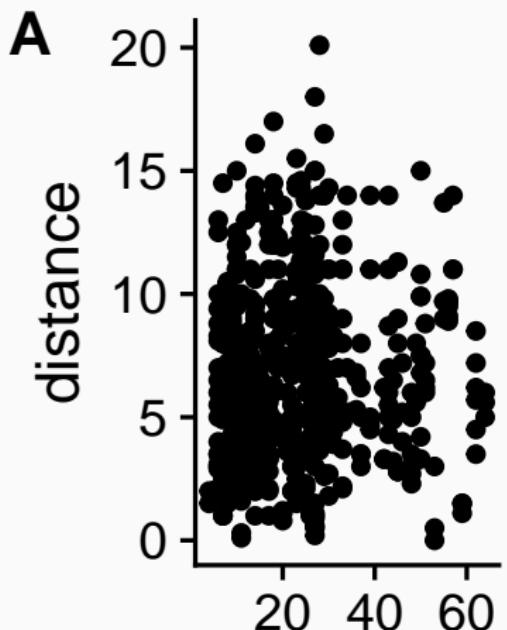
Some publication themes:

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

Composite figures

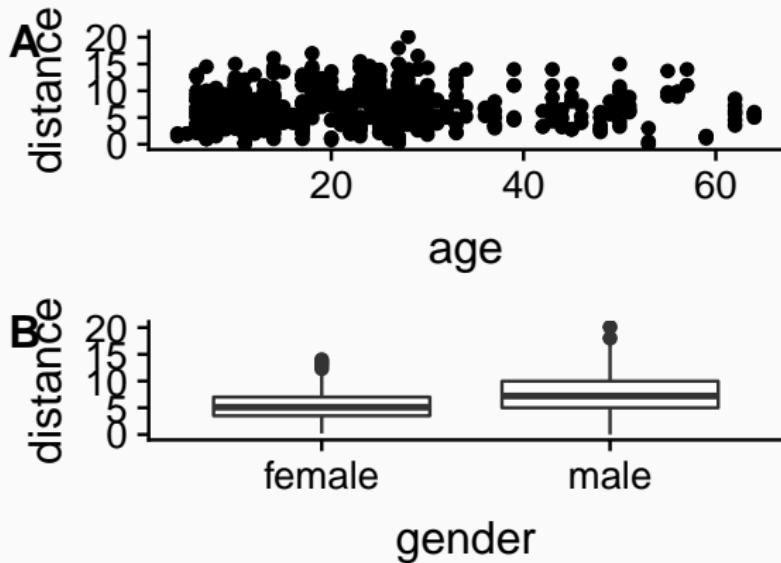
Composite figures: cowplot

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

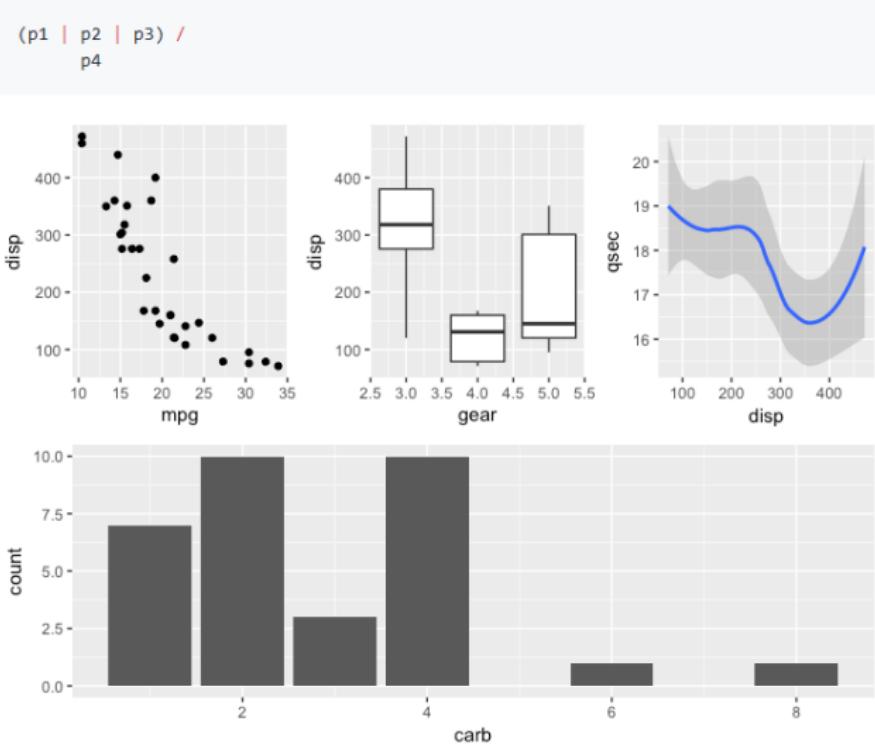


Composite figures

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

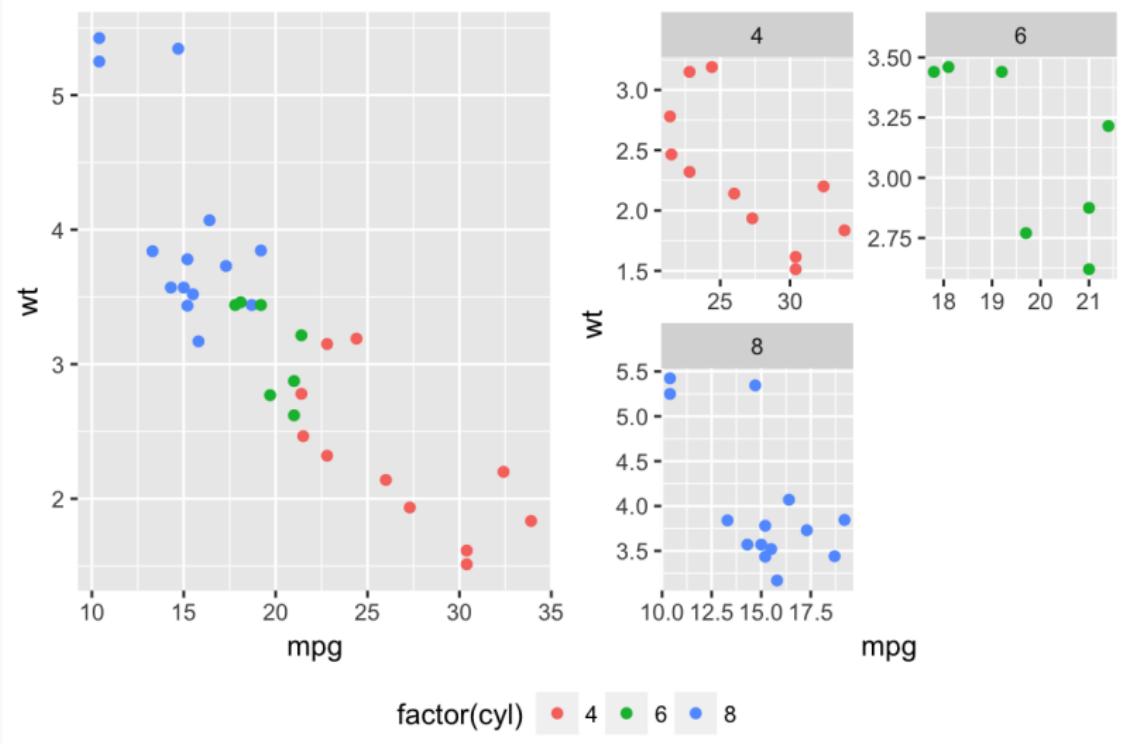


Composite figures: patchwork



<https://github.com/thomasp85/patchwork>

Composite figures: egg



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

Saving plot

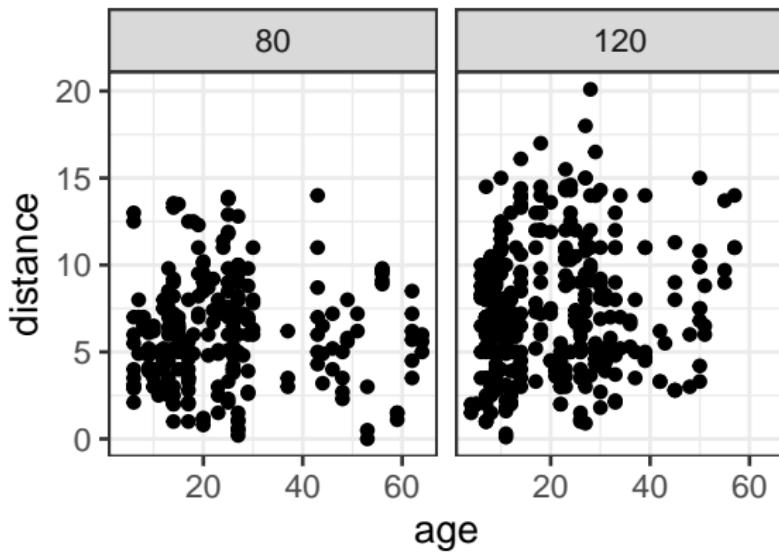
```
ggsave("myplot.pdf")
```

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

Facetting (small multiples)

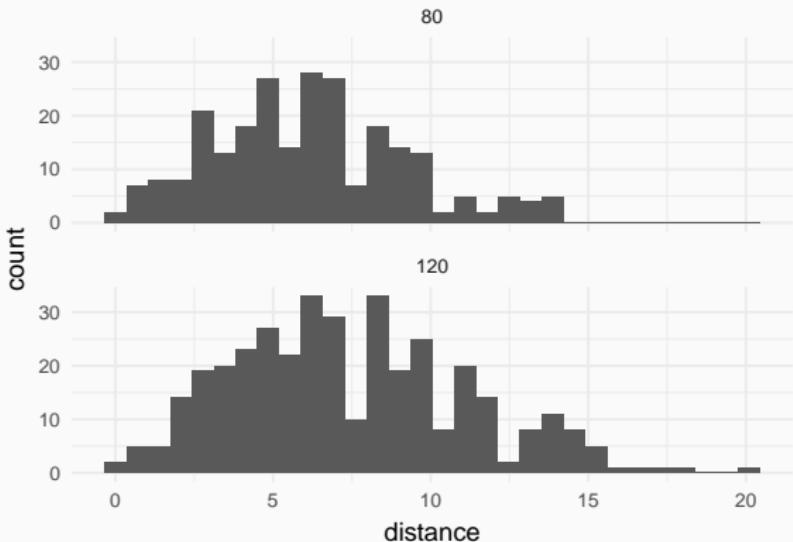
Facetting

```
ggplot(paperplanes) + aes(age, 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)
```

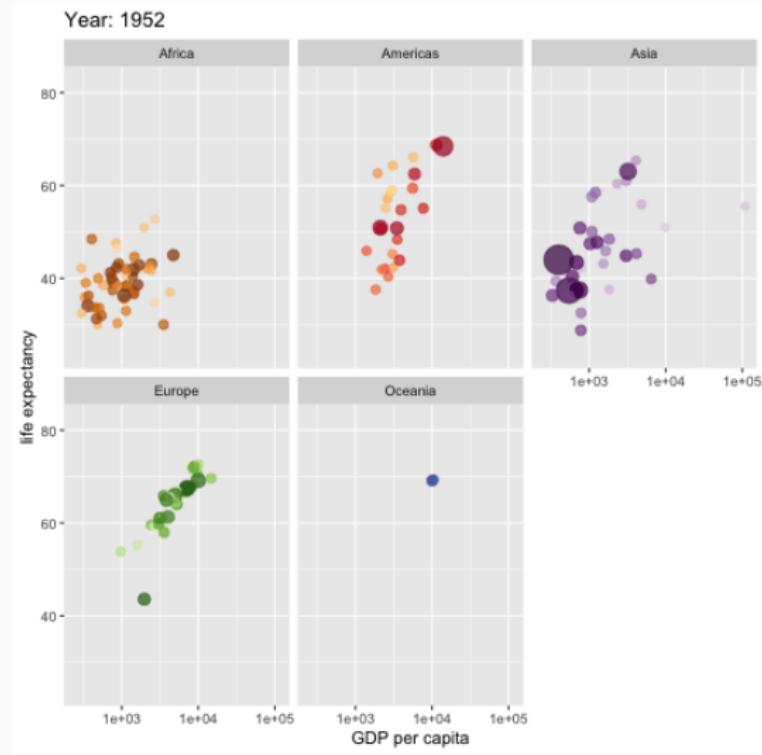


Interactivity: plotly

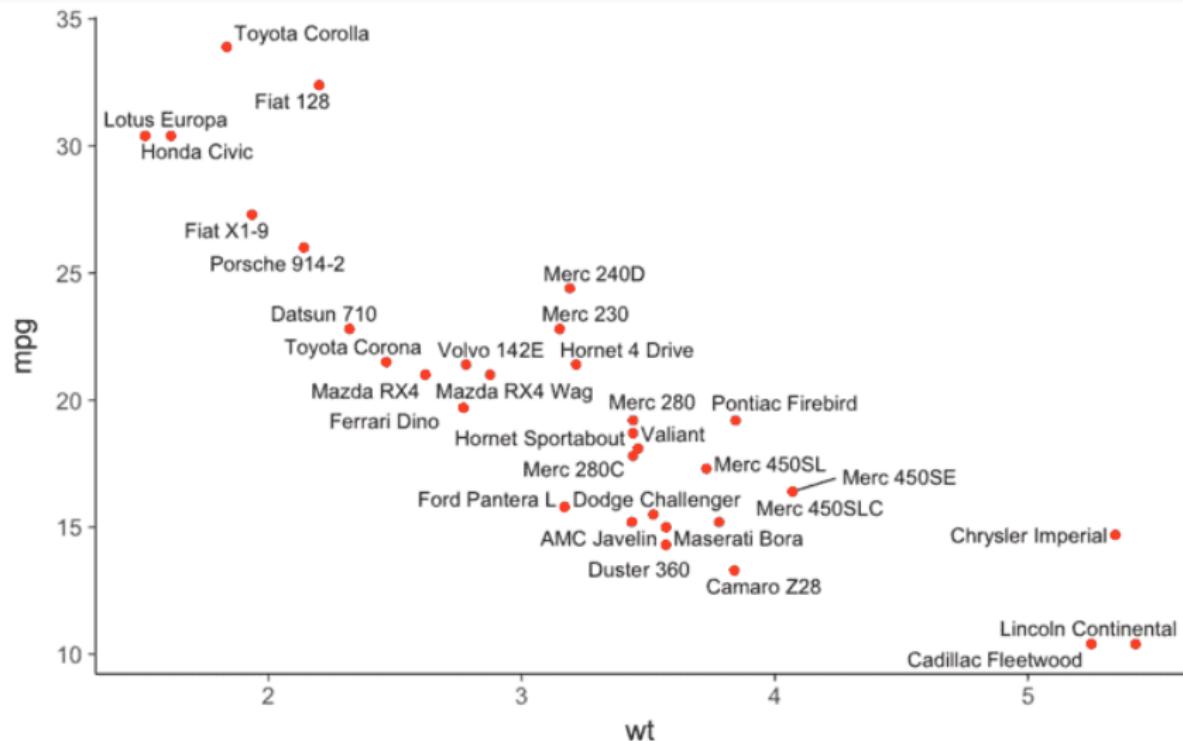
```
library(plotly)
myplot <- ggplot(paperplanes) +
  aes(age, distance)) +
  geom_point()
ggplotly(myplot)
```

Animated graphs

<https://github.com/thomasp85/gganimate>



Automatic label placement



Many extensions!

<https://www.ggplot2-exts.org/>

40 registered extensions available to explore

Sort: Github stars Filter: search name, author, desc... Author Filter Tag Filter CRAN Only

Showing 40 of 40

gg3D
3D perspective plots for ggplot2.
• **author:** Daniel Adler
• **tags:** 3D, Visualization
• **js libraries:**

ggQC
Use ggQC to plot single, faceted and multi-layered quality control charts.
• **author:** Kenneth Gray
• **tags:** QC, XMR, XbarR, SixSigma, Visualization
• **js libraries:**

ggedit
ggedit is aimed to interactively edit ggplot layers, scales and themes aesthetics.
• **author:** jordic
• **tags:** visualization, interactive, shiny, general
• **js libraries:**

ggforce
ggforce is aimed at providing missing functionality to ggplot2 through the extension system introduced with ggplot2 v2.0.0.
• **author:** thomasp85
• **tags:** visualization, general
• **js libraries:**

ggalt
A compendium of 'geom's, 'coords' and 'stats' for 'ggplot2'.
• **author:** hrbrmstr
• **tags:** visualization, general
• **js libraries:**

ggiplot
ggimplot is a ggplot2 extension for making interactive plots.
• **author:** denisghosh
• **tags:** visualization, general
• **js libraries:**

ggstance
ggstance implements horizontal versions of common ggplot2 geoms.
• **author:** forbesl
• **tags:** visualization, general
• **js libraries:**

ggrepel
ggrepel over-lapping text labels away from each other.
• **author:** slowflow
• **tags:** visualization, general
• **js libraries:**

ggraph
ggraph is tailored at plotting graph-like data structures (graphs, networks, trees, hierarchies...).
• **author:** thomasp85
• **tags:** visualization, general
• **js libraries:**

ggpmisc
Miscellaneous Extensions to 'ggplot2'.
• **author:**
• **tags:** visualization, general
• **js libraries:**

geomnet
geomnet implements network visualizations in ggplot2 via geom.net.
• **author:** schmitt
• **tags:** visualization, general
• **js libraries:**

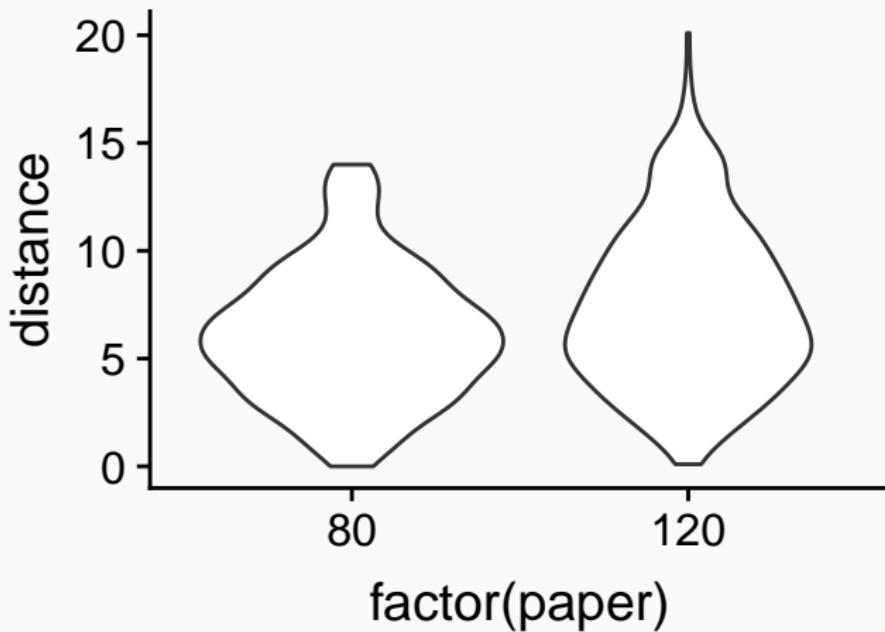
ggExtra
ggExtra lets you add histograms to ggplot2.
• **author:** damian
• **tags:** visualization, general
• **js libraries:**

Summary

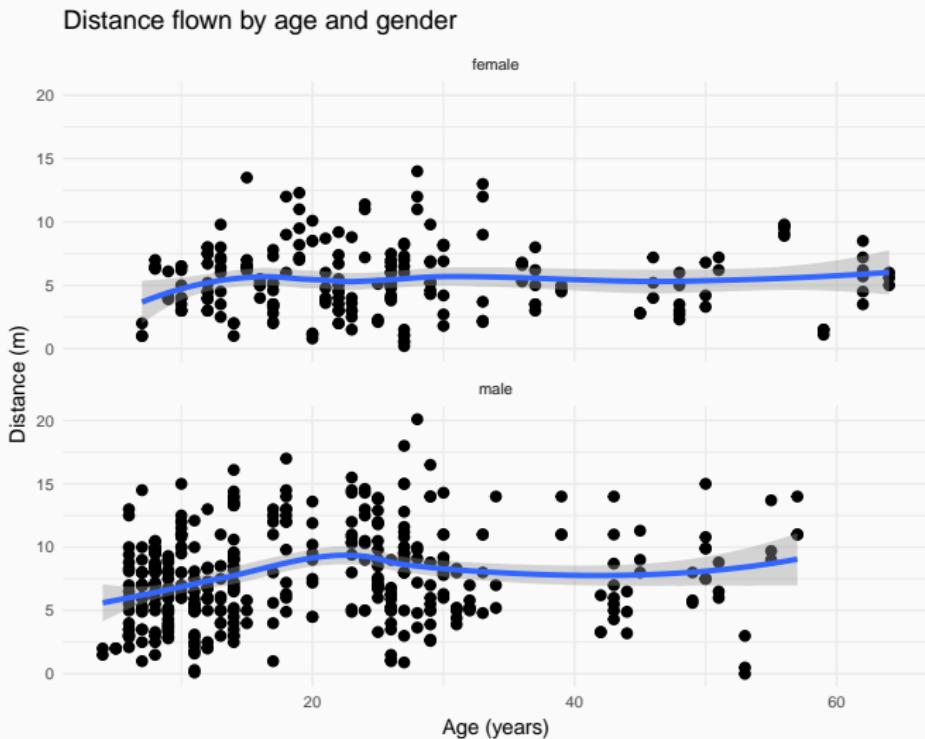
Grammar of graphics

- **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...)

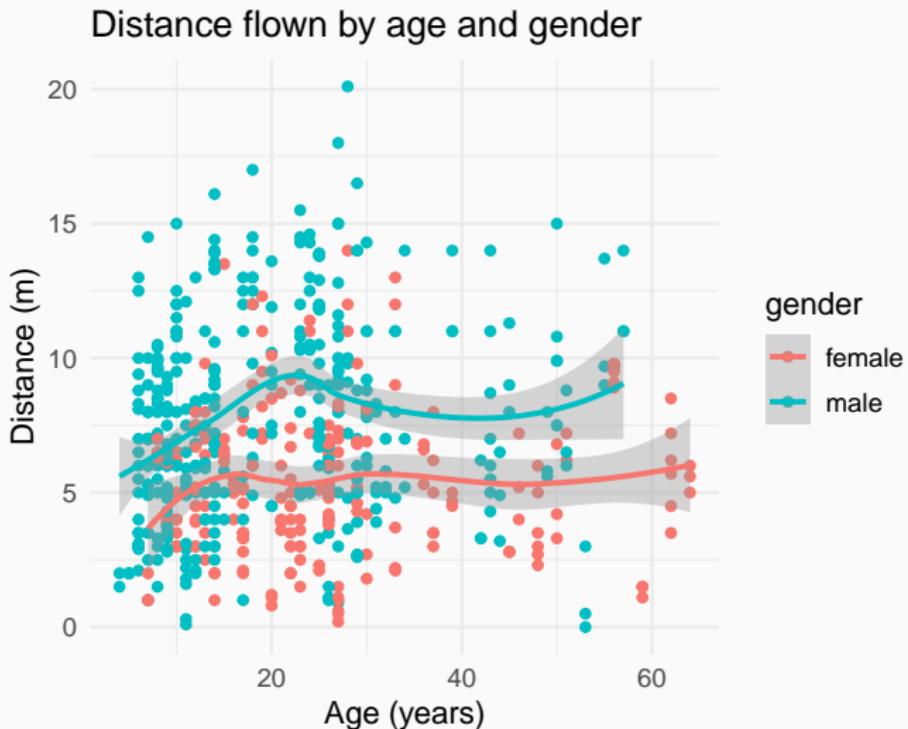
Exercise: make a plot like this one



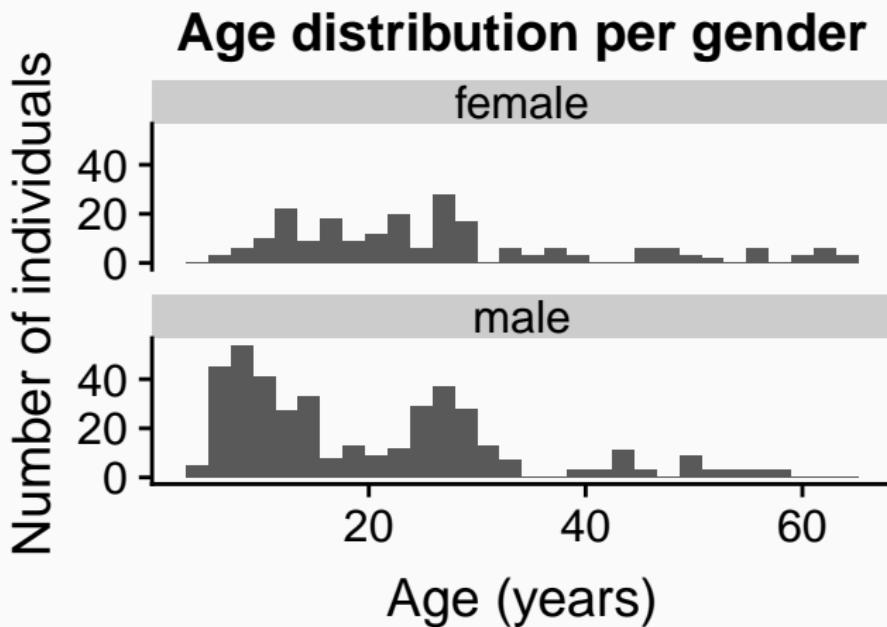
Exercise: make a plot like this one



Exercise: make a plot like this one

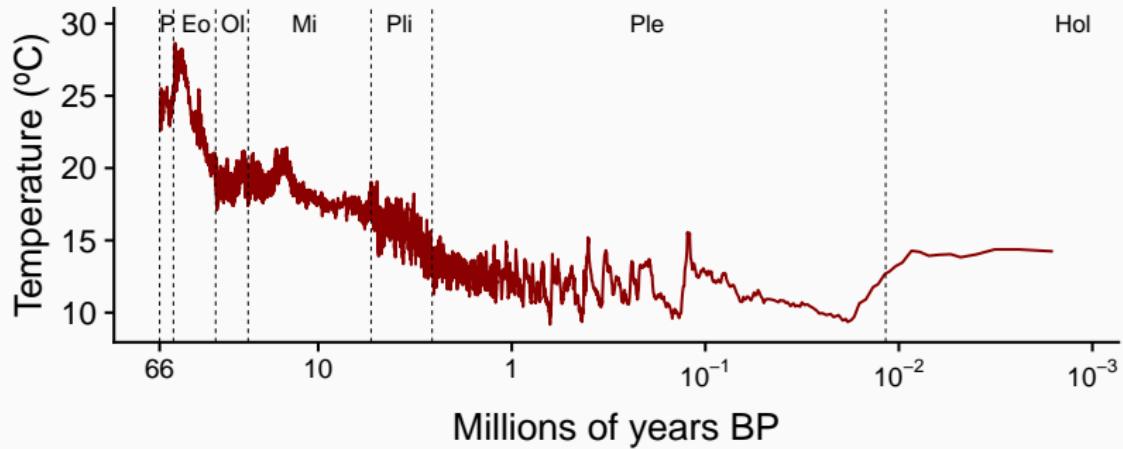


Exercise: make a plot like this one

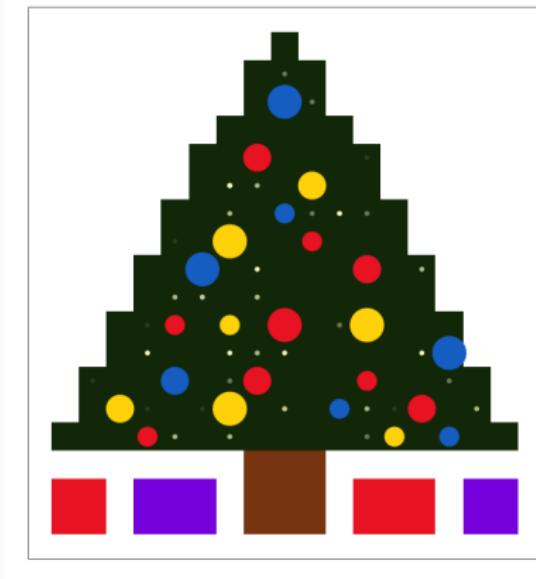


Exercise: make a plot like this one

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>