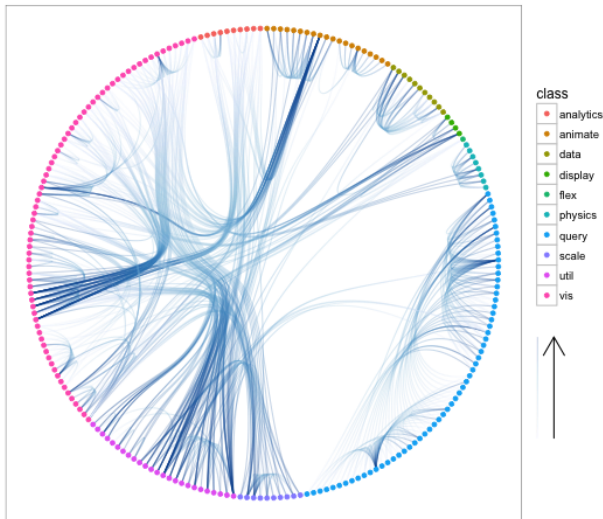


# Data visualisation with ggplot2

---

Francisco Rodriguez-Sanchez (@frod\_san)

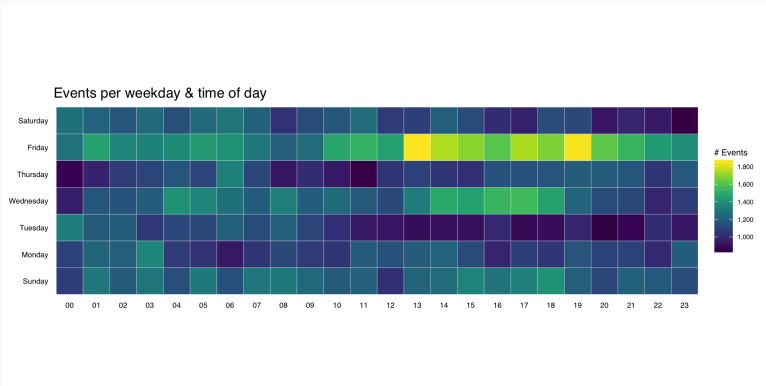
March 2017



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



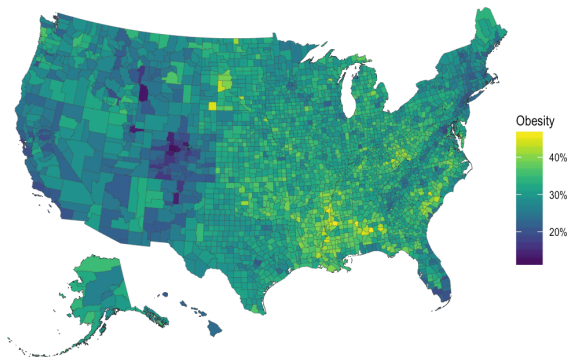
<http://spatial.ly/2012/02/great-maps-ggplot2/>



<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](http://www.cdc.gov/diabetes/atlas/countydata/County_ListofIndicators.html)

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

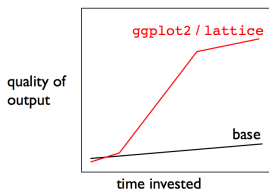
6

# Why ggplot

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

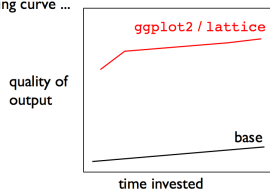
# At the beginning it's hard, but then it pays off

week one ....



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

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



\* 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](#)



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



Search

ggplot2

36,854 results



The Practical Dev  
@ThePracticalDev



Follow

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

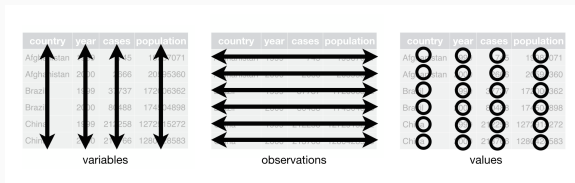
---

## 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

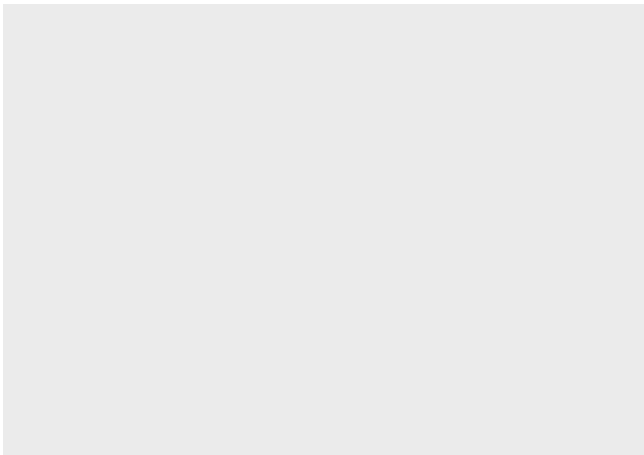
# Data must be a tidy data frame



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

## Calling ggplot

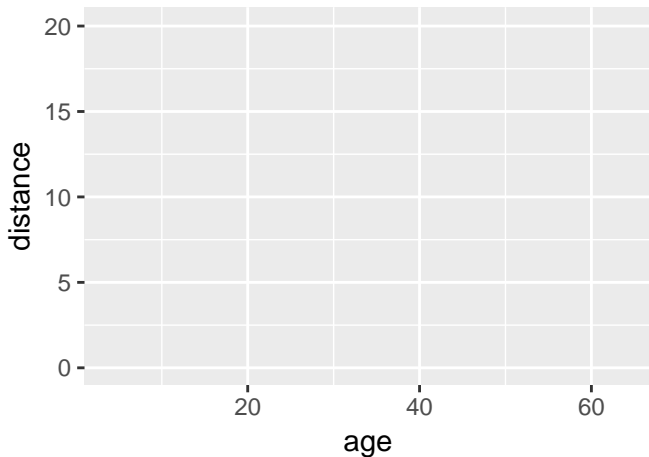
```
library(ggplot2)  
ggplot(paperplanes)
```





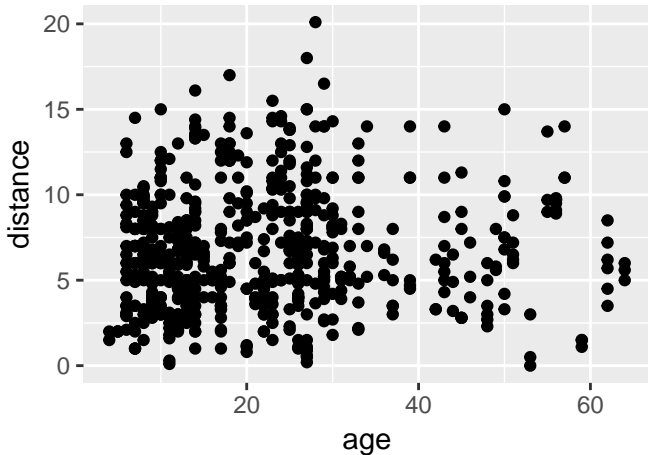
## What variables as axes?

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



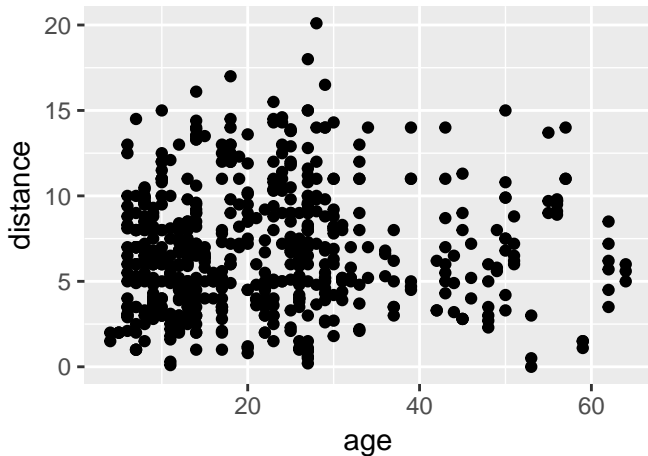
## Adding layers (geoms)

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



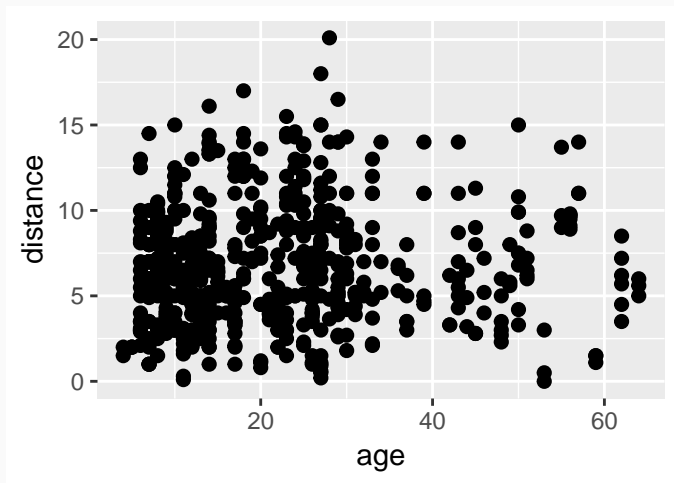
## Assigning ggplot objects

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



## Changing point size and type

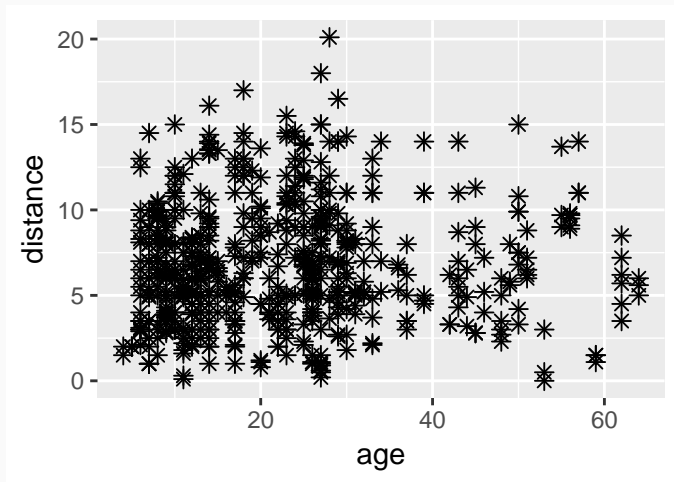
```
myplot + geom_point(size = 2)
```



Check out `geom_point` help [here](#)

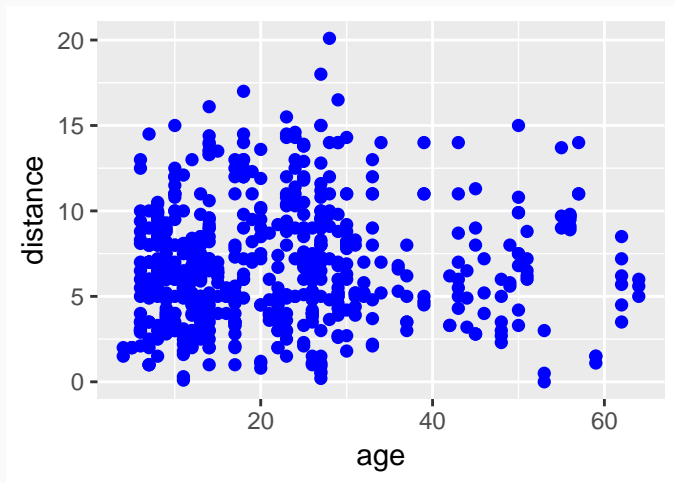
## Changing point size and type

```
myplot + geom_point(size = 2, shape = 8)
```



## Changing point size and type

```
myplot + geom_point(size = 2, shape = 16, colour = "blue")
```



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

```
myplot + geom_point(aes(colour = gender))
```



## Map geom aesthetics (colour, shape) to variable

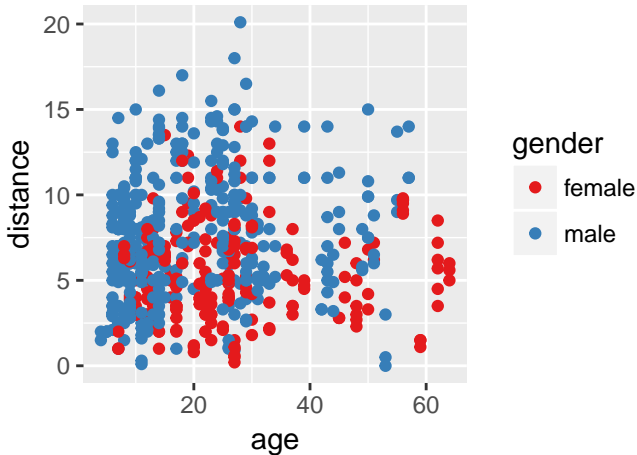
```
myplot + geom_point(aes(colour = gender, shape = gender))
```





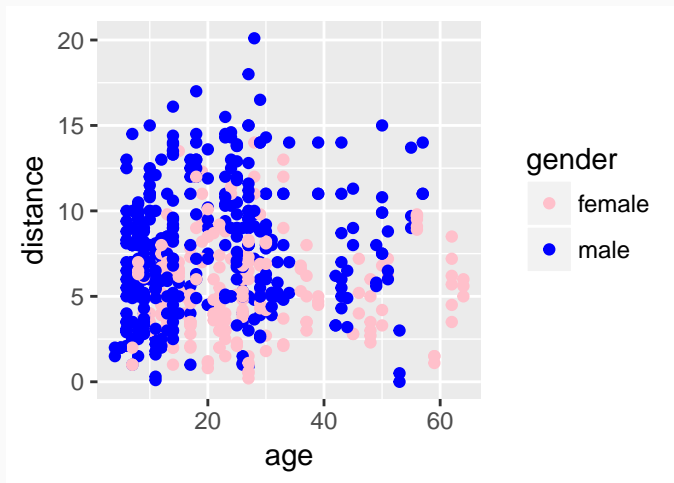
## Change colour scale

```
myplot + geom_point(aes(colour = gender)) +  
  scale_colour_brewer(type = "qual", palette = 6)
```



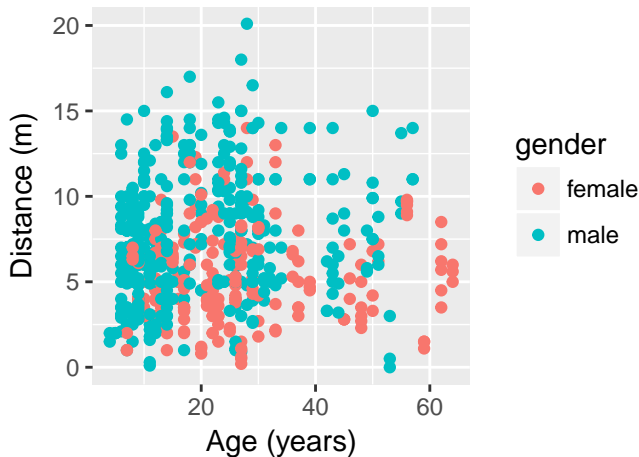
## Change colour scale

```
myplot + geom_point(aes(colour = gender)) +  
  scale_colour_manual(values = c("pink", "blue"))
```



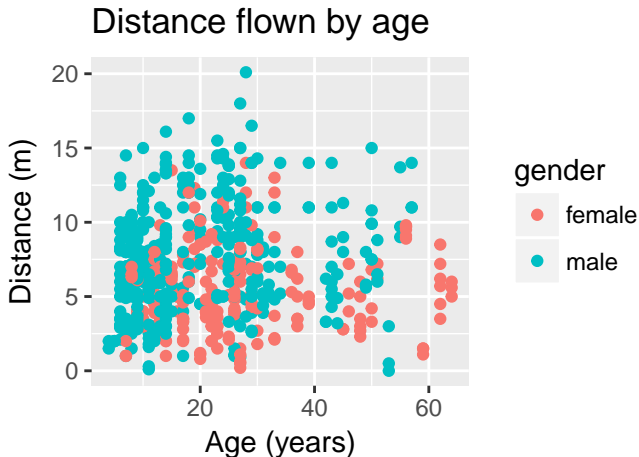
## Change axis labels: xlab & ylab

```
myplot <- myplot + geom_point(aes(colour = gender))  
myplot <- myplot +  
  labs(x = "Age (years)", y = "Distance (m)")
```



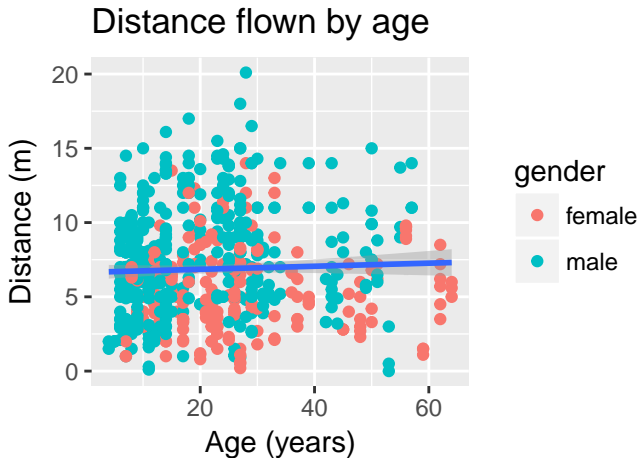
## Set title

```
myplot <- myplot +  
  labs(title = "Distance flown by age")
```



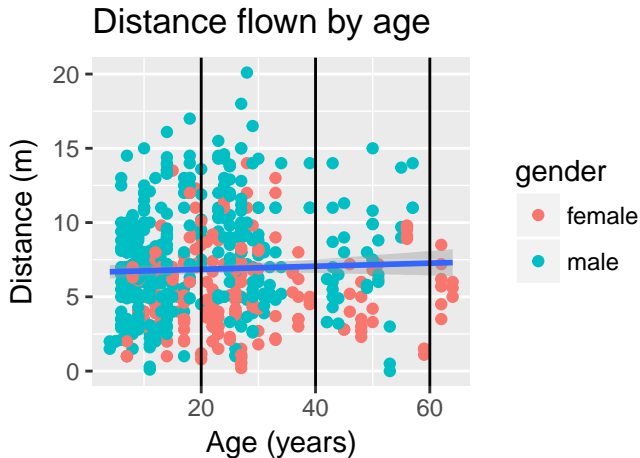
## Adding another layer

```
myplot <- myplot + geom_smooth(method = "lm")
```



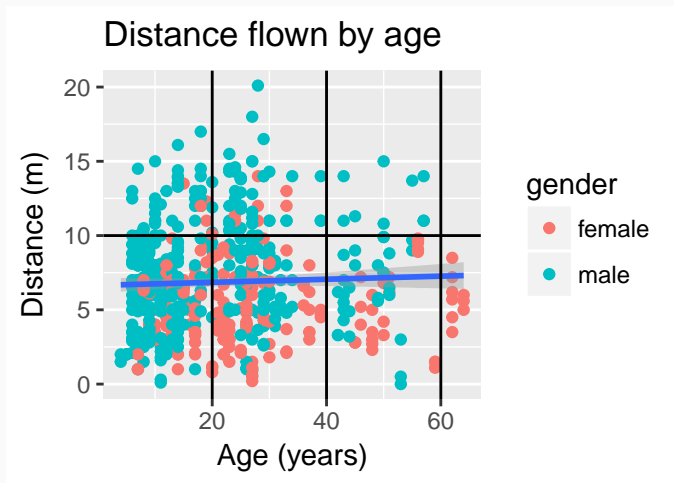
## Adding another layer

```
myplot <- myplot + geom_vline(xintercept = c(20, 40, 60))
```

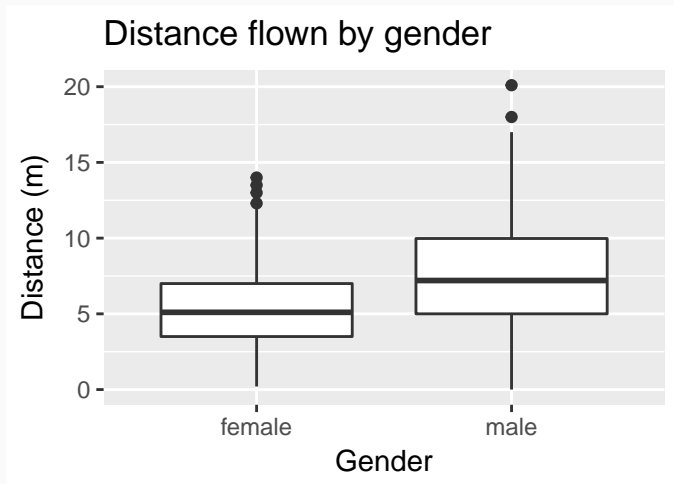


## Adding another layer

```
myplot <- myplot + geom_hline(yintercept = 10)
```

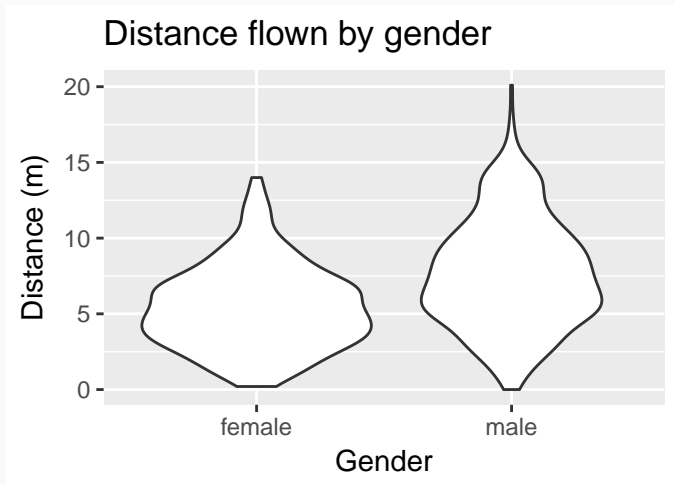


## Exercise: Make a plot like this one

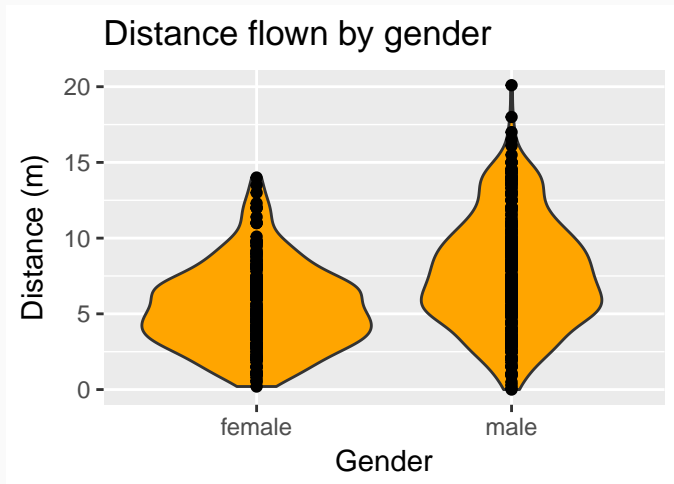




## Exercise: Make a plot like this one



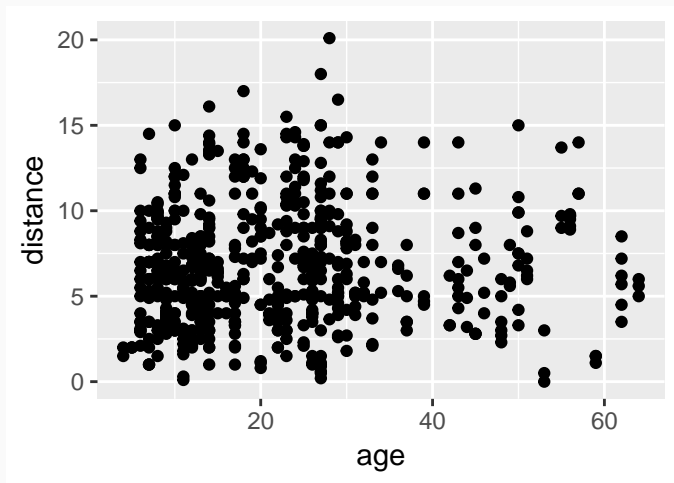
## Exercise: Make a plot like this one



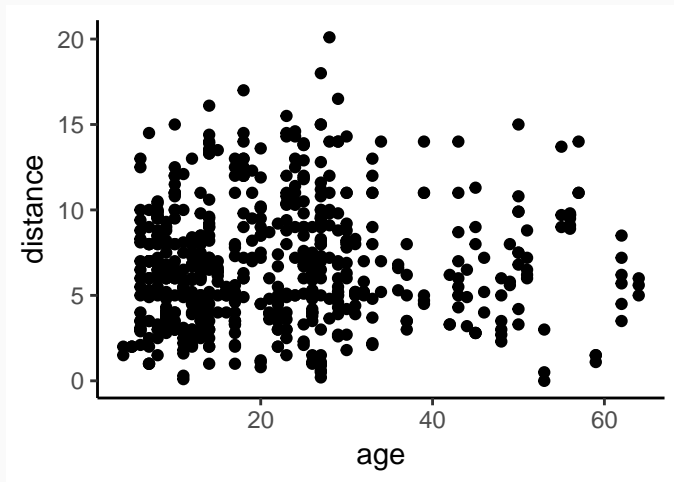
## Themes: changing plot appearance

---

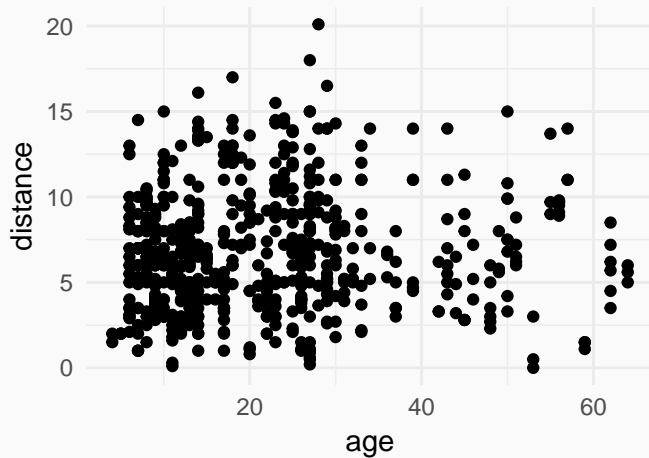
```
myplot <- ggplot(paperplanes, aes(x = age, y = distance)) +  
  geom_point()
```



```
myplot + theme_classic()
```

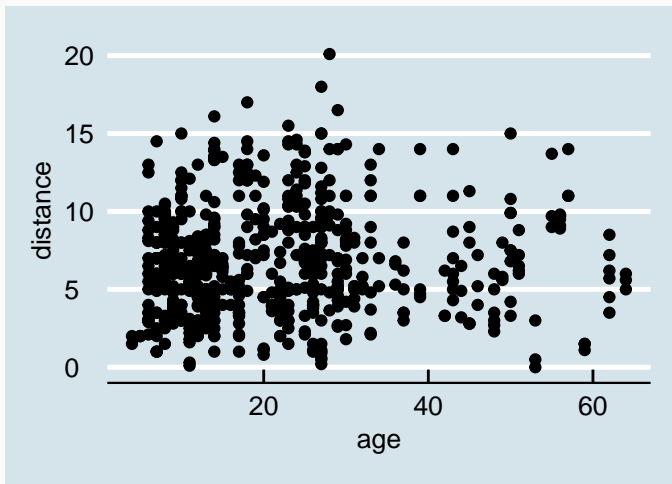


```
myplot + theme_minimal()
```



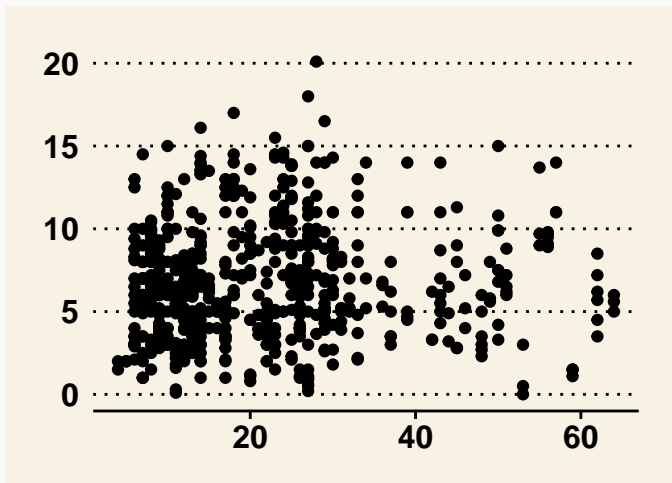
## Lots of themes out there

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



## Lots of themes out there

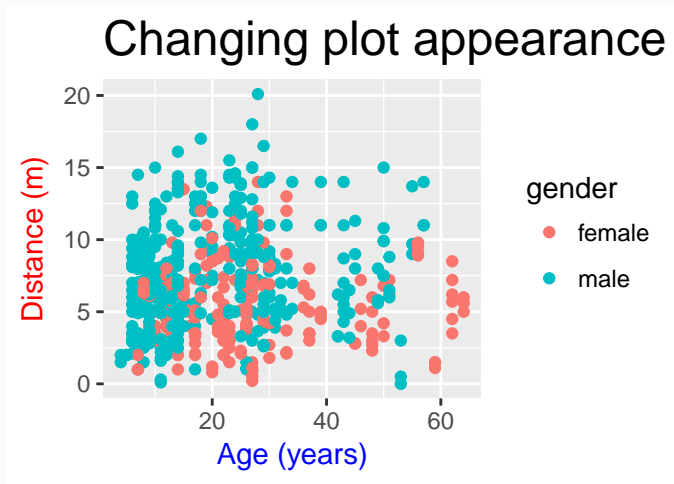
```
myplot + theme_ws()
```





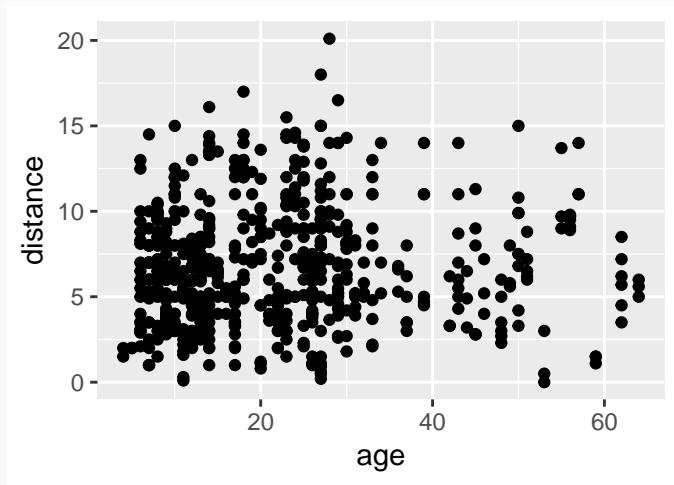
?theme

Exercise: make a plot like this one



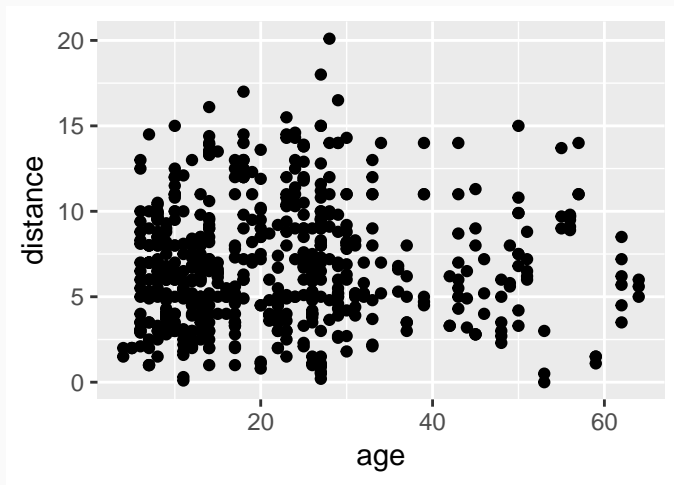
## Easily changing appearance with ggthemeassist (Rstudio addin)

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



## Easily changing appearance with ggedit

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



## Think twice before editing plots out of R



**Trevor A. Branch**

@TrevorABranch



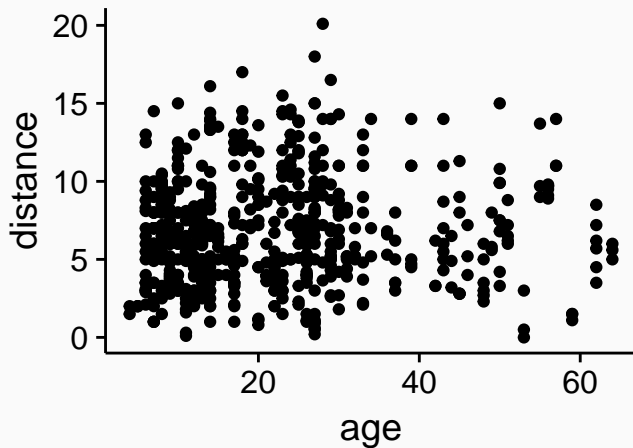
Follow

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>

## Publication-quality plots

```
library(cowplot)  
myplot
```



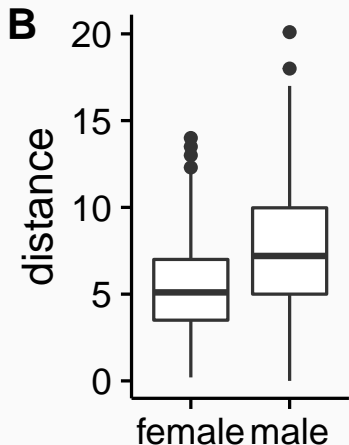
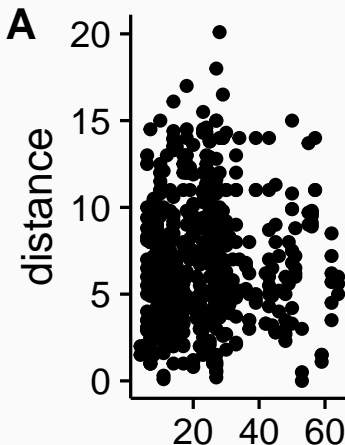
Publication themes:

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

## Composite figures

```
library(cowplot)

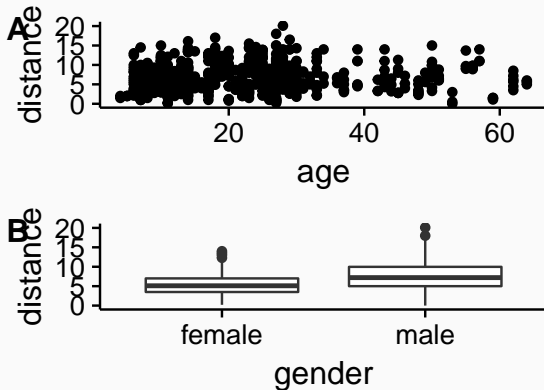
plot1 <- ggplot(paperplanes, aes(age, distance)) + geom_point()
plot2 <- ggplot(paperplanes, aes(gender, distance)) + geom_boxplot()
plot_grid(plot1, plot2, labels = "AUTO")
```





## Composite figures

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

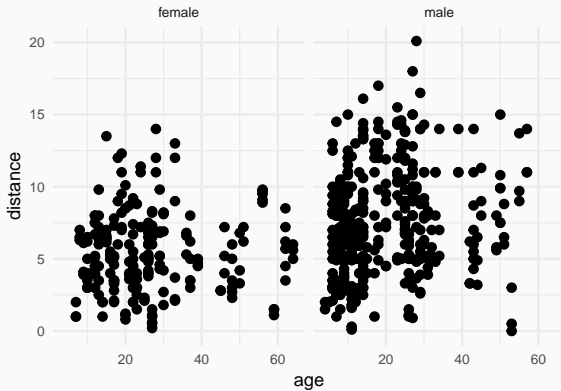


## Saving plot: ggsave

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

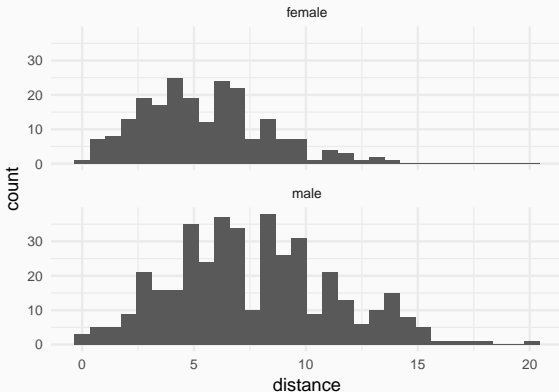
# Facetting

```
ggplot(paperplanes, aes(age, distance)) +  
  geom_point() + theme_minimal(base_size = 8) +  
  facet_wrap(~gender)
```



# Facetting

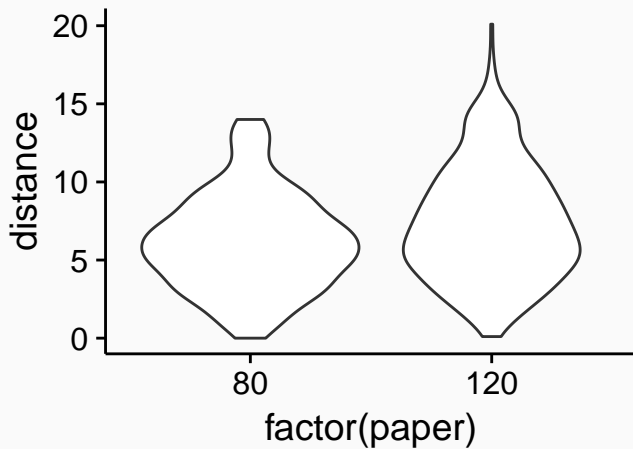
```
ggplot(paperplanes) +  
  geom_histogram(aes(distance)) + theme_minimal(base_size = 8) +  
  facet_wrap(~gender, nrow = 2)
```



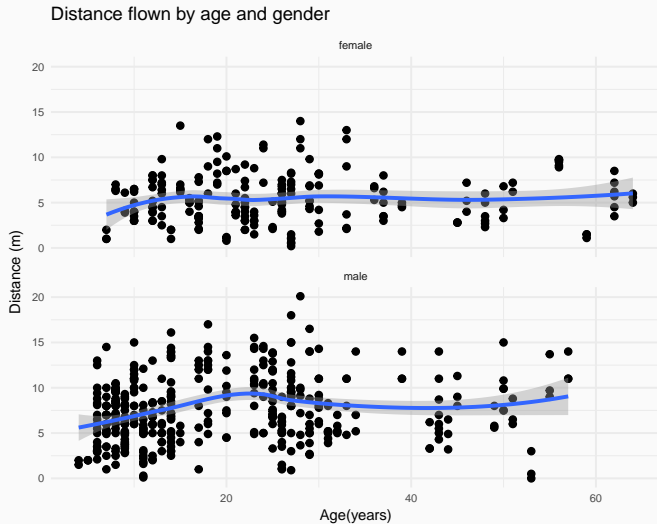
```
library(plotly)
myplot <- ggplot(paperplanes, aes(age, distance)) + geom_point()
ggplotly(myplot)
```

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

Exercise: make a plot like this one

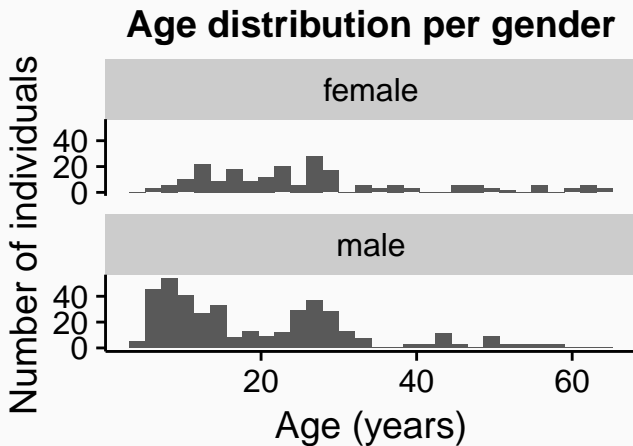


## Exercise: make a plot like this one





Exercise: make a plot like this one





Slides and source code available at

<https://github.com/Pakillo/ggplot-intro>