ETC1010: Introduction to Data Analysis
Week 4, part B

Advanced topics in data visualisation

Lecturer: *Nicholas Tierney*

Department of Econometrics and Business Statistics

✉ ETC1010.Clayton-x@monash.edu

April 2020

While the song is playing...

Draw a mental model / concept map of last lectures content on joins.

recap

- Joins

Joins with a person and a coat, by Leight Tami

Upcoming Due Dates

- Assignment 1: Due April 8 at 5pm (Today!)

Exploring life expectancy and income

We want to plot life expectancy vs income, but there's a problem:

```
gap_life_au
## # A tibble: 9 x 3
##   country    year life_expectancy
##   <chr>     <dbl>           <dbl>
## 1 Australia  2012            82.5
## 2 Australia  2013            82.6
## 3 Australia  2014            82.5
## 4 Australia  2015            82.5
## 5 Australia  2016            82.5
## 6 Australia  2017            82.4
## 7 Australia  2018            82.5
## 8 Australia  2019            82.7
## 9 Australia  2020            82.8
```

```
gap_income_au
## # A tibble: 9 x 3
##   country     year   gdp
##   <chr>      <dbl> <dbl>
## 1 Australia   2012 42800
## 2 Australia   2013 43200
## 3 Australia   2014 43700
## 4 Australia   2015 44100
## 5 Australia   2016 44600
## 6 Australia   2017 44900
## 7 Australia   2018 45400
## 8 Australia   2019 45500
## 9 Australia   2020 45800
```

We need them in the same dataframe!

We could try `bind_cols()`, to bind dataframes columns together

```
bind_cols(gap_life_au,
          gap_income_au)
## # A tibble: 9 x 6
##   country    year life_expectancy country1  year1   gdp
##   <chr>     <dbl>           <dbl> <chr>     <dbl> <dbl>
## 1 Australia  2012            82.5 Australia  2012 42800
## 2 Australia  2013            82.6 Australia  2013 43200
## 3 Australia  2014            82.5 Australia  2014 43700
## 4 Australia  2015            82.5 Australia  2015 44100
## 5 Australia  2016            82.5 Australia  2016 44600
## 6 Australia  2017            82.4 Australia  2017 44900
## 7 Australia  2018            82.5 Australia  2018 45400
## 8 Australia  2019            82.7 Australia  2019 45500
## 9 Australia  2020            82.8 Australia  2020 45800
```

But this has problems:

1. It produces messy output (country1, year1)
2. It doesn't work if the data doesn't have the same number of rows

```
## # A tibble: 9 x 6
##   country    year life_expectancy country1  year1   gdp
##   <chr>     <dbl>           <dbl> <chr>     <dbl> <dbl>
## 1 Australia  2012            82.5 Australia  2012 42800
## 2 Australia  2013            82.6 Australia  2013 43200
## 3 Australia  2014            82.5 Australia  2014 43700
## 4 Australia  2015            82.5 Australia  2015 44100
## 5 Australia  2016            82.5 Australia  2016 44600
## 6 Australia  2017            82.4 Australia  2017 44900
## 7 Australia  2018            82.5 Australia  2018 45400
## 8 Australia  2019            82.7 Australia  2019 45500
## 9 Australia  2020            82.8 Australia  2020 45800
```

How to bind data?

For example, how do we add this co2 data to income or life?

```
gap_co2_au
## # A tibble: 3 x 3
##   country    year   co2
##   <chr>     <dbl> <dbl>
## 1 Australia  2012  17  
## 2 Australia  2013  16.1
## 3 Australia  2014  15.4
```

How to bind data?

We can't use `bind_cols()`

```
bind_cols(gap_co2_au,
          gap_income_au)

Error: Argument 2 must be length 3, not 9
```

We could think about a more complex approach using `filter`, and so on...

But surely this must be a problem that we encounter in data analysis?

Someone must have thought of a solution to this before?

They did! **Joins**!

Joins!

We can use `left_join()` to combine the income and life expectancy data

```
left_join(x = gap_income_au,
          y = gap_life_au,
          by = c("country", "year"))
## # A tibble: 9 x 4
##   country    year   gdp life_expectancy
##   <chr>     <dbl> <dbl>           <dbl>
## 1 Australia  2012 42800            82.5
## 2 Australia  2013 43200            82.6
## 3 Australia  2014 43700            82.5
## 4 Australia  2015 44100            82.5
## 5 Australia  2016 44600            82.5
## 6 Australia  2017 44900            82.4
## 7 Australia  2018 45400            82.5
## 8 Australia  2019 45500            82.7
## 9 Australia  2020 45800            82.8
```

Add co2 data with another join:

We get missings for co2, because we don't have c02 values for 2015 and beyond.

```
left_join(x = gap_income_au,
          y = gap_life_au,
          by = c("country", "year")) %>%
  left_join(gap_co2_au,
            by = c("country", "year"))
## # A tibble: 9 x 5
##   country    year   gdp life_expectancy   co2
##   <chr>     <dbl> <dbl>           <dbl> <dbl>
## 1 Australia  2012 42800            82.5  17  
## 2 Australia  2013 43200            82.6  16.1
## 3 Australia  2014 43700            82.5  15.4
## 4 Australia  2015 44100            82.5  NA  
## 5 Australia  2016 44600            82.5  NA  
## 6 Australia  2017 44900            82.4  NA  
## 7 Australia  2018 45400            82.5  NA  
## 8 Australia  2019 45500            82.7  NA  
## 9 Australia  2020 45800            82.8  NA
```

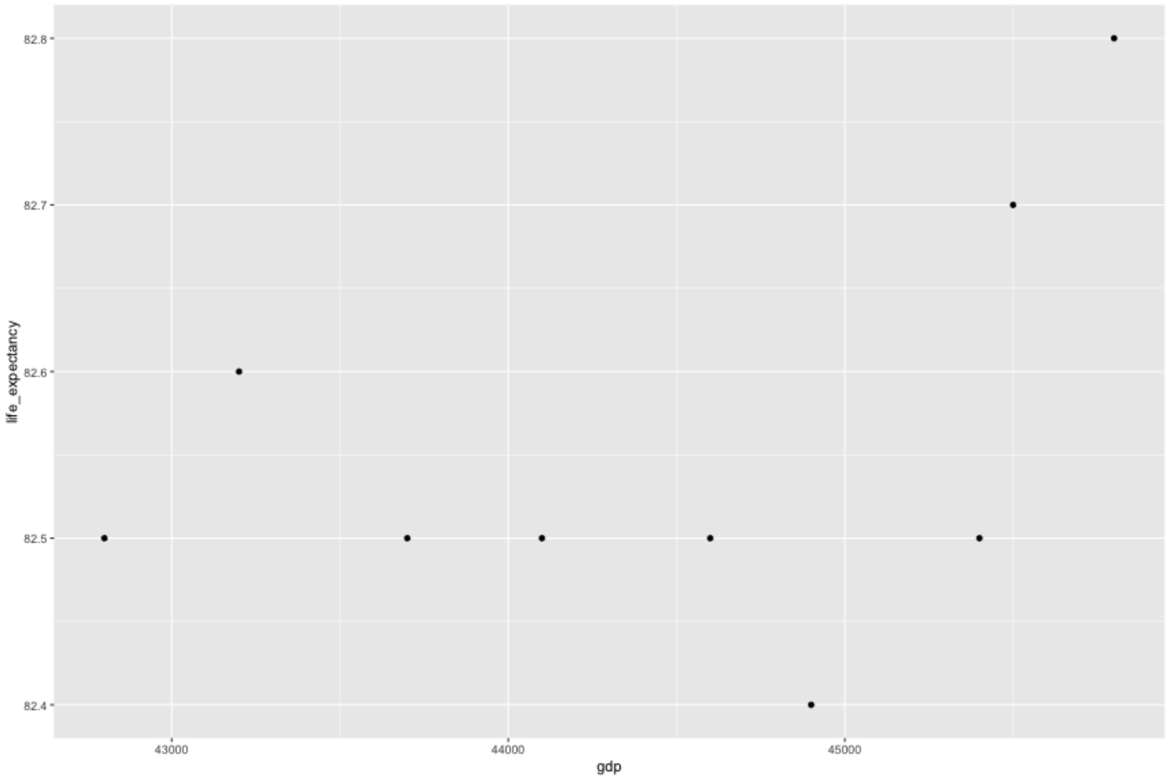
So now we can combine that together like so:

```
gap_au <- left_join(x = gap_income_au,
          y = gap_life_au,
          by = c("country", "year")) %>%
  left_join(gap_co2_au,
            by = c("country", "year"))

gap_au
## # A tibble: 9 x 5
##   country    year   gdp life_expectancy   co2
##   <chr>     <dbl> <dbl>           <dbl> <dbl>
## 1 Australia  2012 42800            82.5  17
## 2 Australia  2013 43200            82.6  16.1
## 3 Australia  2014 43700            82.5  15.4
## 4 Australia  2015 44100            82.5  NA
## 5 Australia  2016 44600            82.5  NA
## 6 Australia  2017 44900            82.4  NA
## 7 Australia  2018 45400            82.5  NA
## 8 Australia  2019 45500            82.7  NA
## 9 Australia  2020 45800            82.8  NA
```

Now we can make a plot!

```
ggplot(gap_au,
       aes(x = gdp,
           y = life_expectancy)) +
  geom_point()
```



Your Turn: go to exercises on rstudio.cloud

open "joins.Rmd"

Discuss with your partner why these two joins produce different results?

```
left_join(gap_co2_au,
          gap_life_au)
## # A tibble: 3 x 4
##   country    year   co2 life_expecta
##   <chr>     <dbl> <dbl>            <
## 1 Australia  2012  17
## 2 Australia  2013  16.1
## 3 Australia  2014  15.4
```

```
left_join(gap_life_au,
          gap_co2_au)
## # A tibble: 9 x 4
##   country    year life_expectancy
##   <chr>     <dbl>           <dbl> <
## 1 Australia  2012            82.5
## 2 Australia  2013            82.6
## 3 Australia  2014            82.5
## 4 Australia  2015            82.5
## 5 Australia  2016            82.5
## 6 Australia  2017            82.4
## 7 Australia  2018            82.5
## 8 Australia  2019            82.7
## 9 Australia  2020            82.8
```

Your Turn:

What happens when we add data from New Zealand into the mix?

How can you join that data together?

Making effective data plots

1. Principles / science of data visualisation
2. Features of graphics

Principles / science of data visualisation

- Palettes and colour blindness
- change blindness
- using proximity
- hierarchy of mappings

Features of graphics

- Layering statistical summaries
- Themes
- adding interactivity

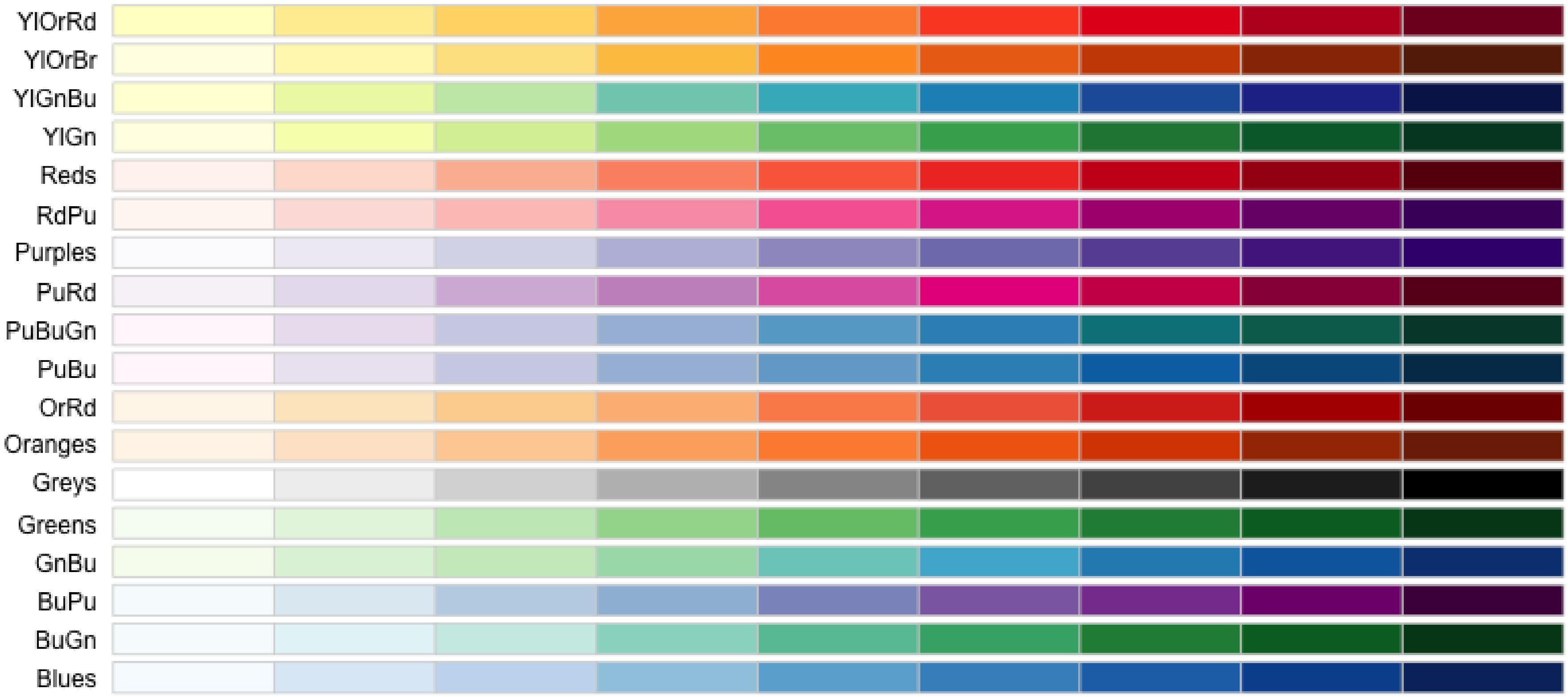
Palettes and colour blindness

There are three main types of colour palette:

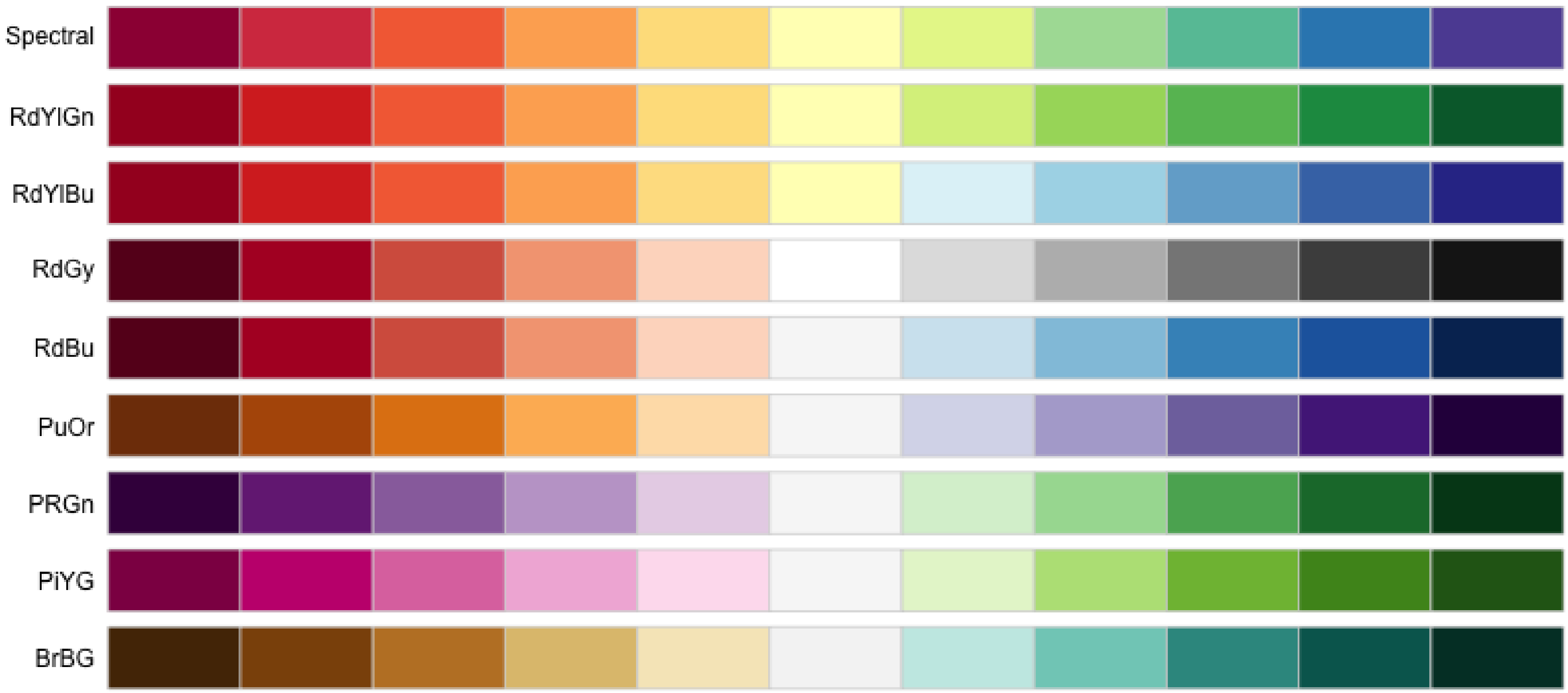
- Qualitative: categorical variables
- Sequential: low to high numeric values
- Diverging: negative to positive values

Qualitative: categorical variables

Sequential: low to high numeric values



Diverging: negative to positive values



Example: TB data

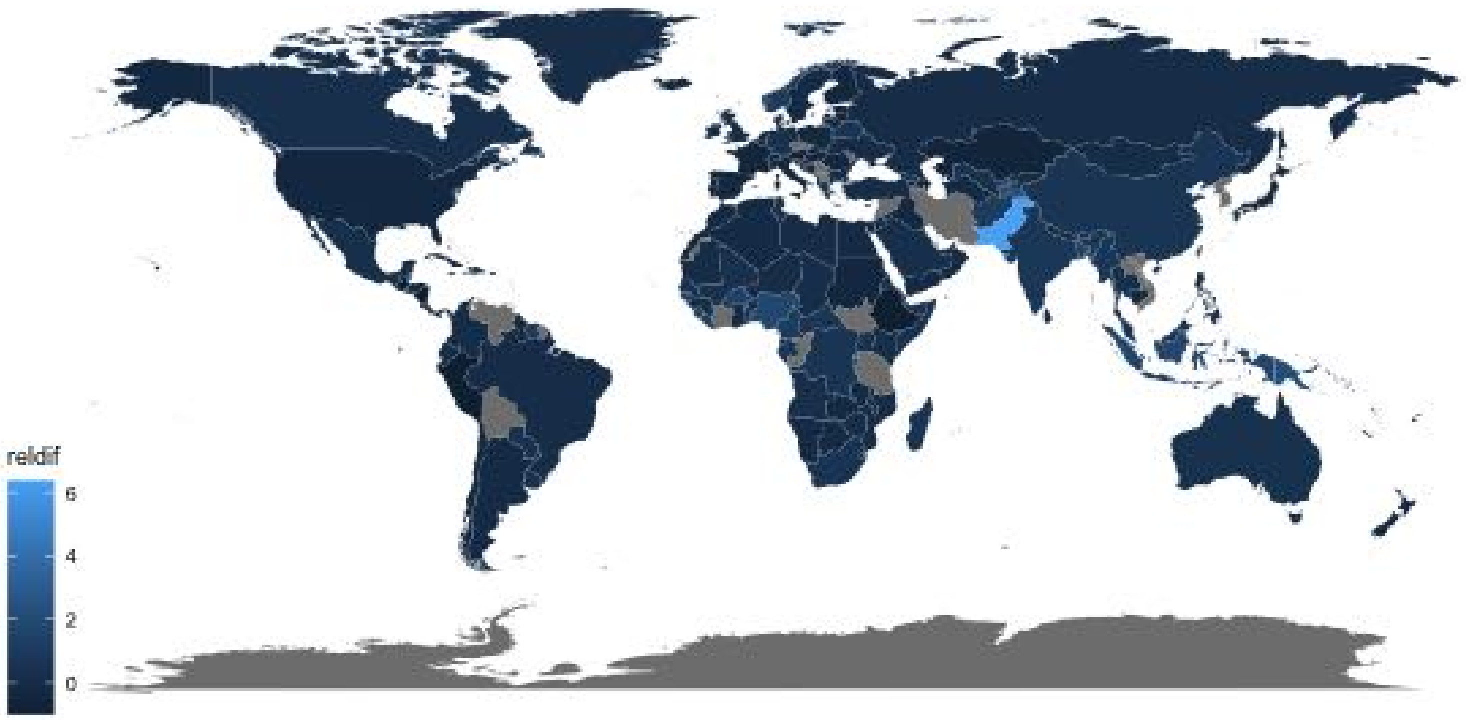
```
## # A tibble: 157,820 x 5
##    country      year count gender age  
##    <chr>       <dbl> <dbl> <chr>  <chr>
##  1 Afghanistan  1980    NA m      04   
##  2 Afghanistan  1981    NA m      04   
##  3 Afghanistan  1982    NA m      04   
##  4 Afghanistan  1983    NA m      04   
##  5 Afghanistan  1984    NA m      04   
##  6 Afghanistan  1985    NA m      04   
##  7 Afghanistan  1986    NA m      04   
##  8 Afghanistan  1987    NA m      04   
##  9 Afghanistan  1988    NA m      04   
## 10 Afghanistan  1989    NA m      04   
## # … with 157,810 more rows
```

Example: TB data: adding relative change

```
## # A tibble: 219 x 4
##    country             `2002` `2012`  reldif
##    <chr>                <dbl>  <dbl>   <dbl>
##  1 Afghanistan           6509  13907  1.14
##  2 Albania                225    185 -0.178
##  3 Algeria               8246   7510 -0.0893
##  4 American Samoa           1      0 -1
##  5 Andorra                  2      2  0
##  6 Angola               17988  22106  0.229
##  7 Anguilla                 0      0  0
##  8 Antigua and Barbuda      4      1 -0.75
##  9 Argentina             5383   4787 -0.111
## 10 Armenia                511    316 -0.382
## # … with 209 more rows
```

Example: Sequential colour with default palette

```
ggplot(tb_map) + geom_polygon(aes(x = long, y = lat, group = group, fill = reldif))
  theme_map()
```

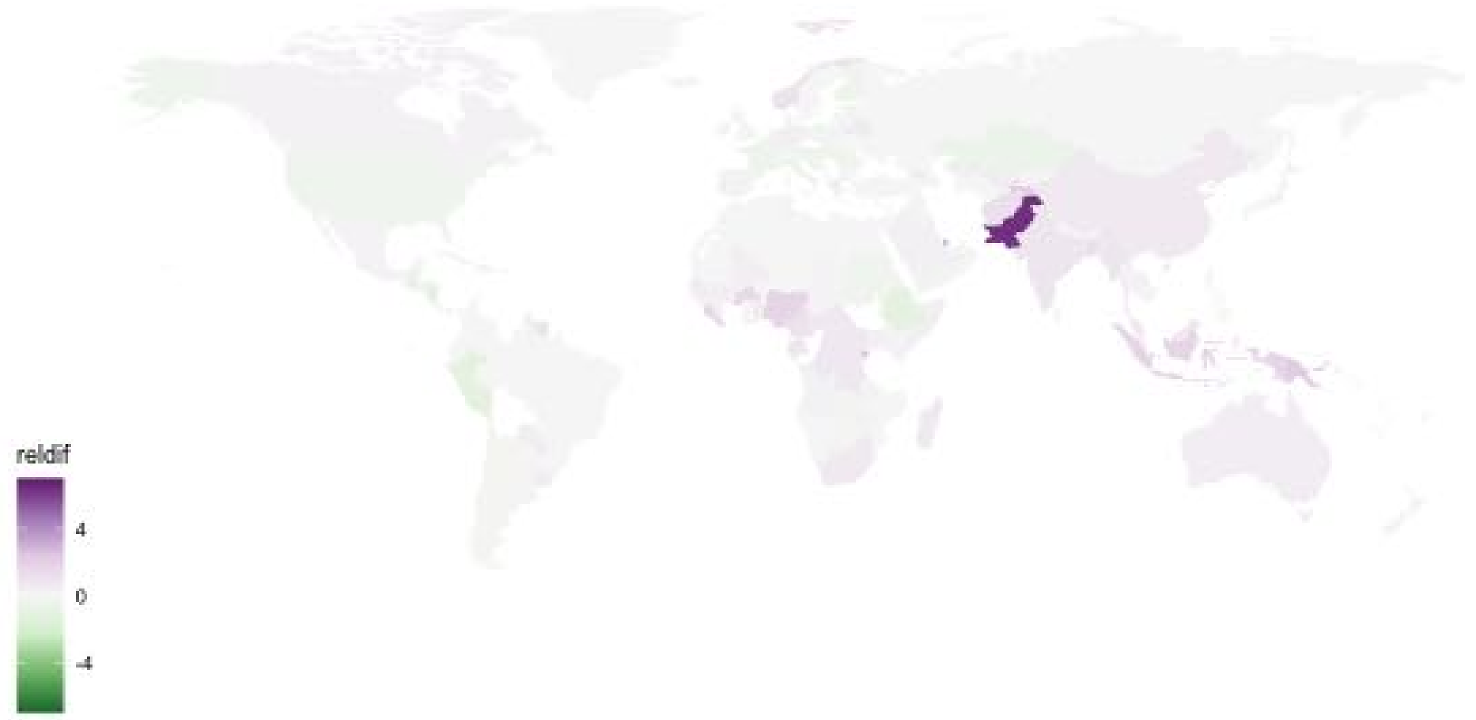


Example: (improved) sequential colour with default palette

```r
library(viridis)
ggplot(tb_map) +
  geom_polygon(aes(x = long, y = lat, group = group, fill = reldif)) +
  theme_map() + scale_fill_viridis(na.value = "white")
```

Example: Diverging colour with better palette

```
ggplot(tb_map) +
  geom_polygon(aes(x = long, y = lat, group = group, fill = reldif)) +
  theme_map() +
  scale_fill_distiller(palette = "PRGn", na.value = "white", limits = c(-7, 7))
```



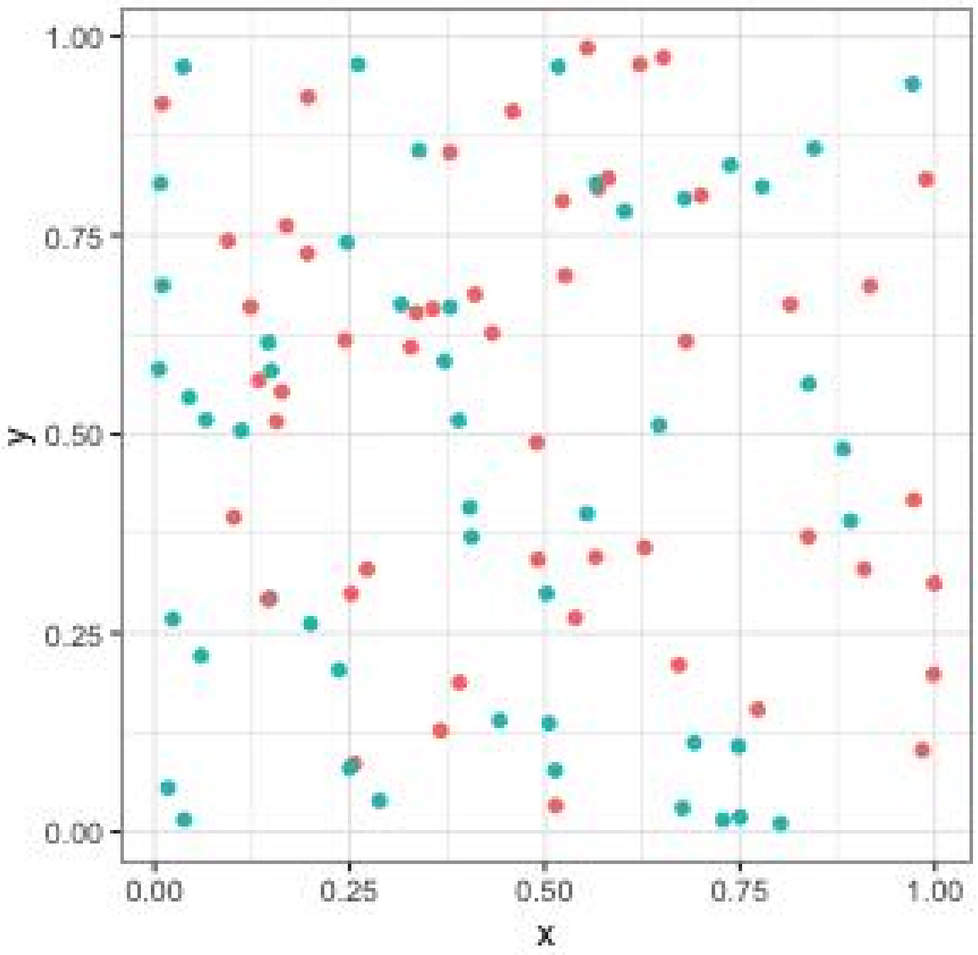
Summary on colour palettes

- Different ways to map colour to values:
    - Qualitative: categorical variables
    - Sequential: low to high numeric values
    - Diverging: negative to positive values

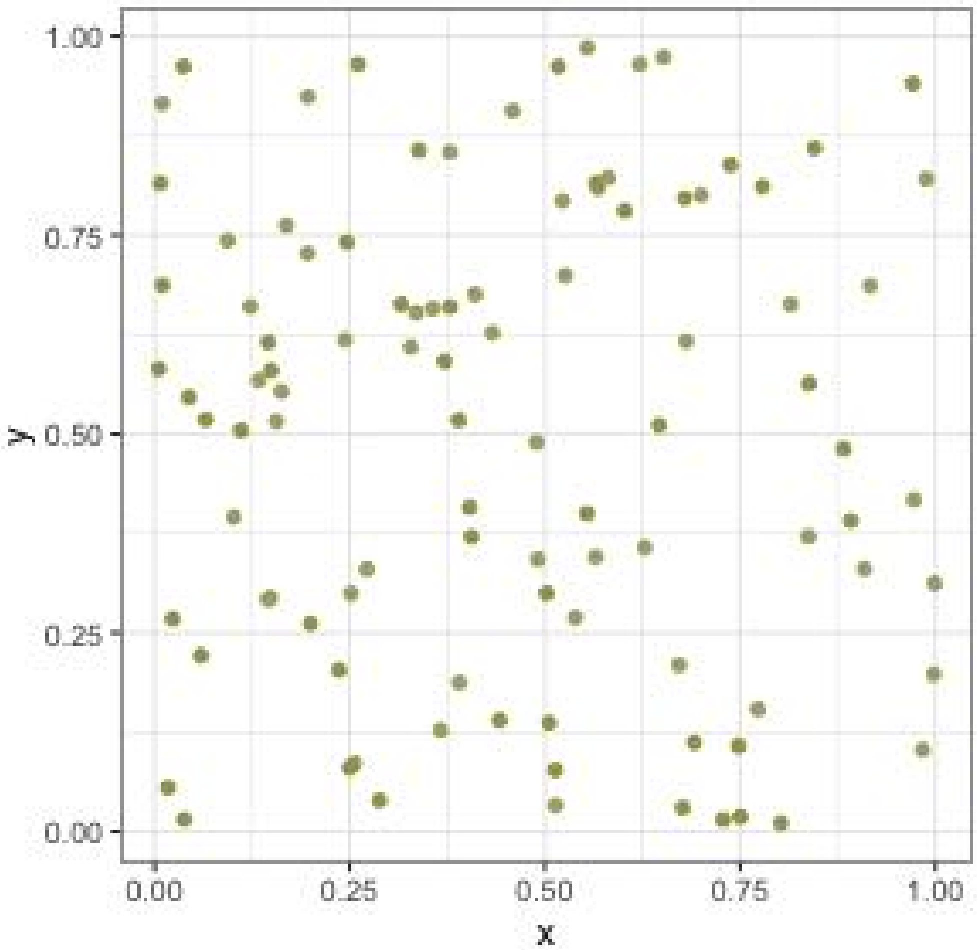
Colour blindness

- About 8% of men (about 1 in 12), and 0.5% women (about 1 in 200) population have difficulty distinguishing between red and green.
- Several colour blind tested palettes: RColorbrewer has an associated web site colorbrewer.org where the palettes are labelled. See also `viridis`, and `scico`.

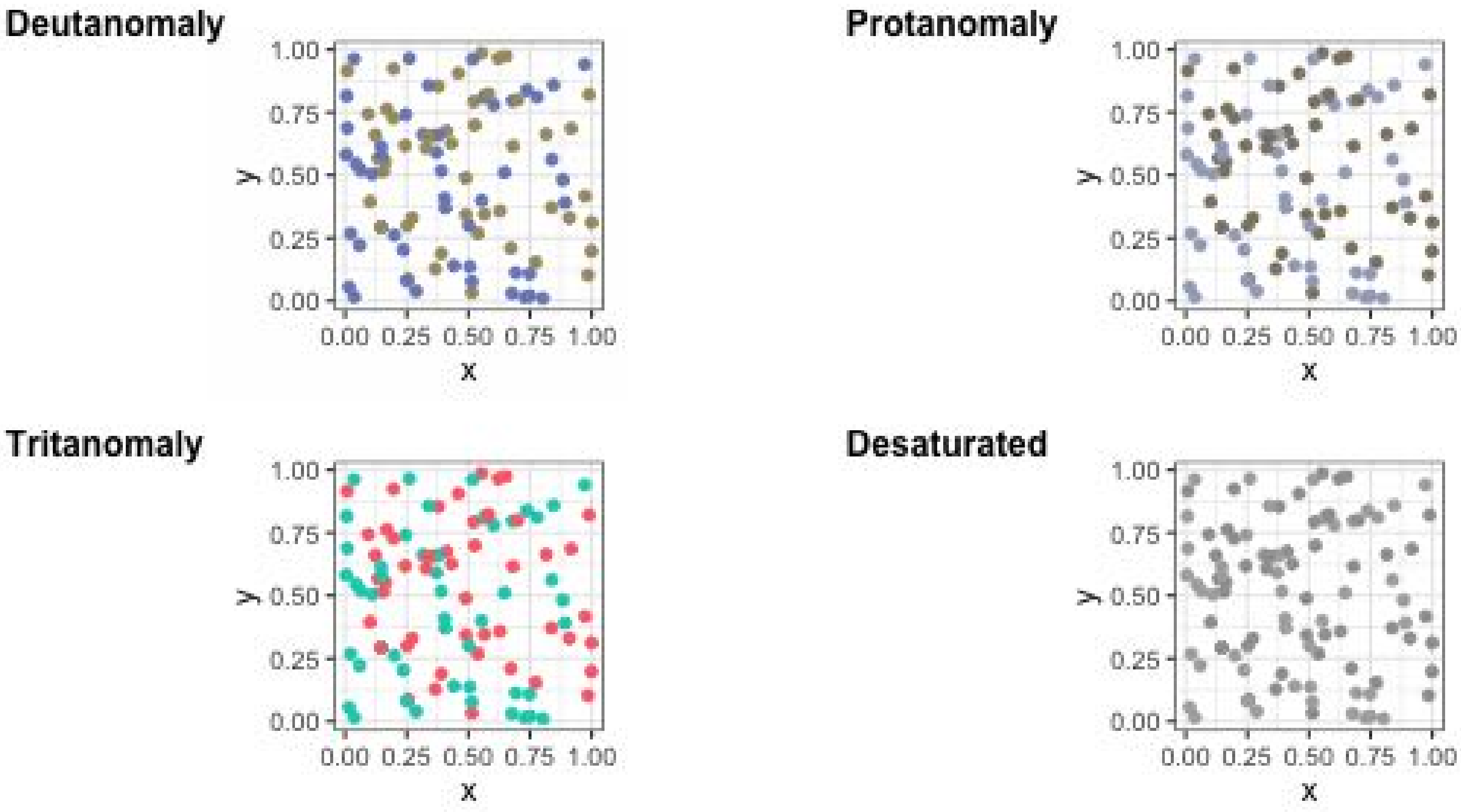
Plot of two coloured points: Normal Mode



Plot of two coloured points: dicromat mode

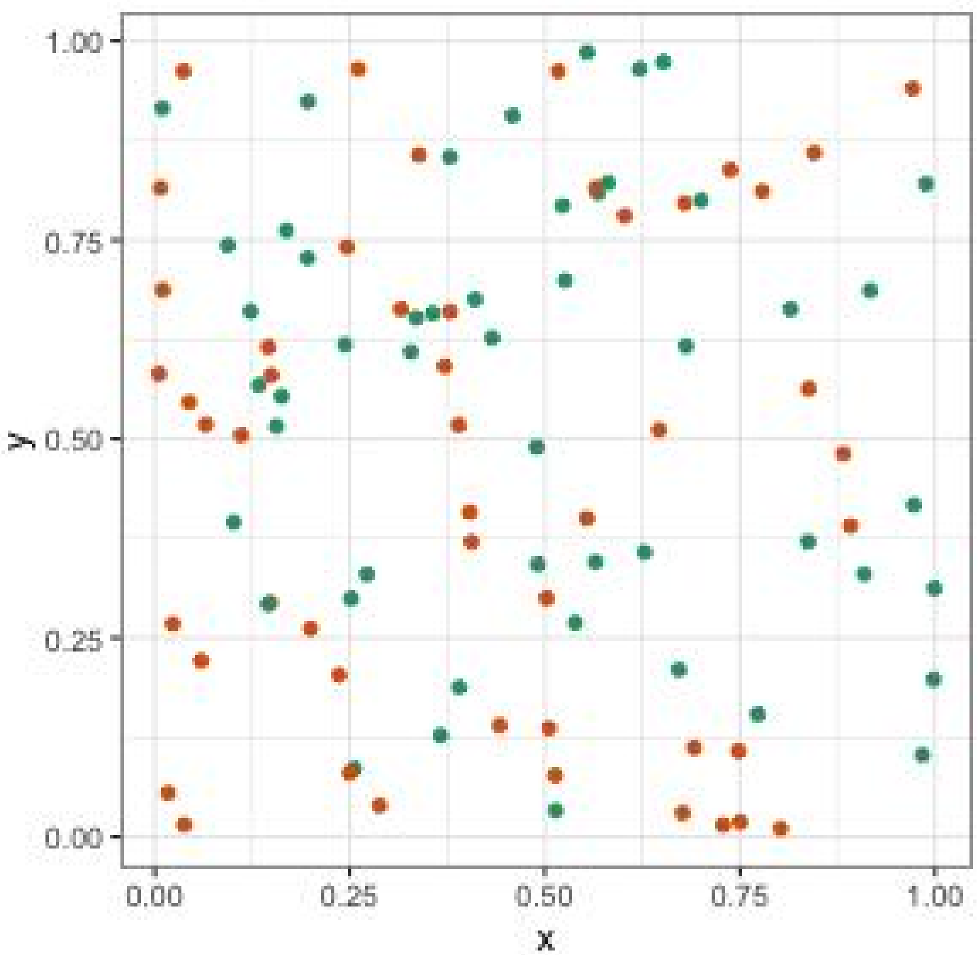


Showing all types of colourblindness

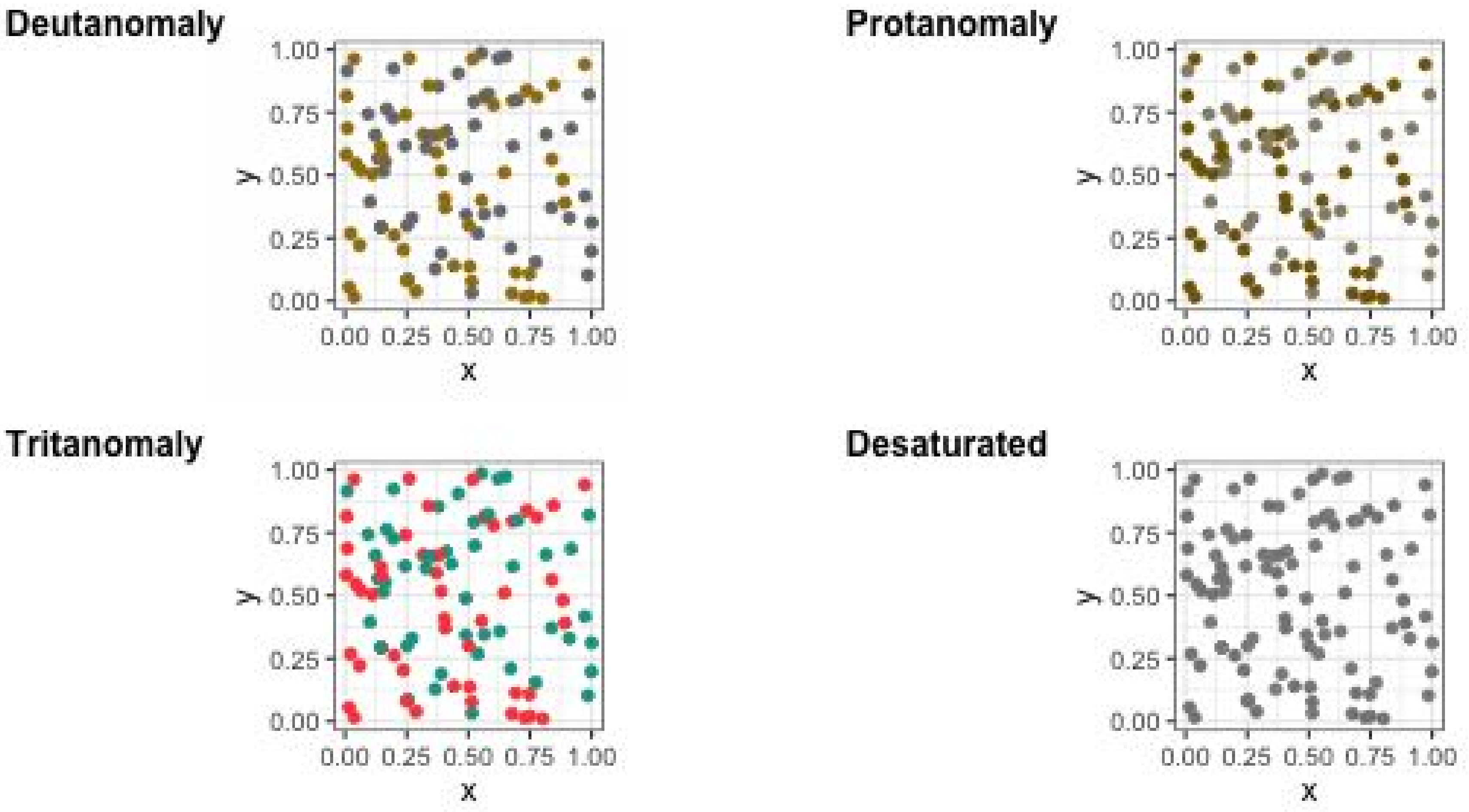


Impact of colourblind-safe palette

```
p2 <- p + scale_colour_brewer(palette = "Dark2")
p2
```

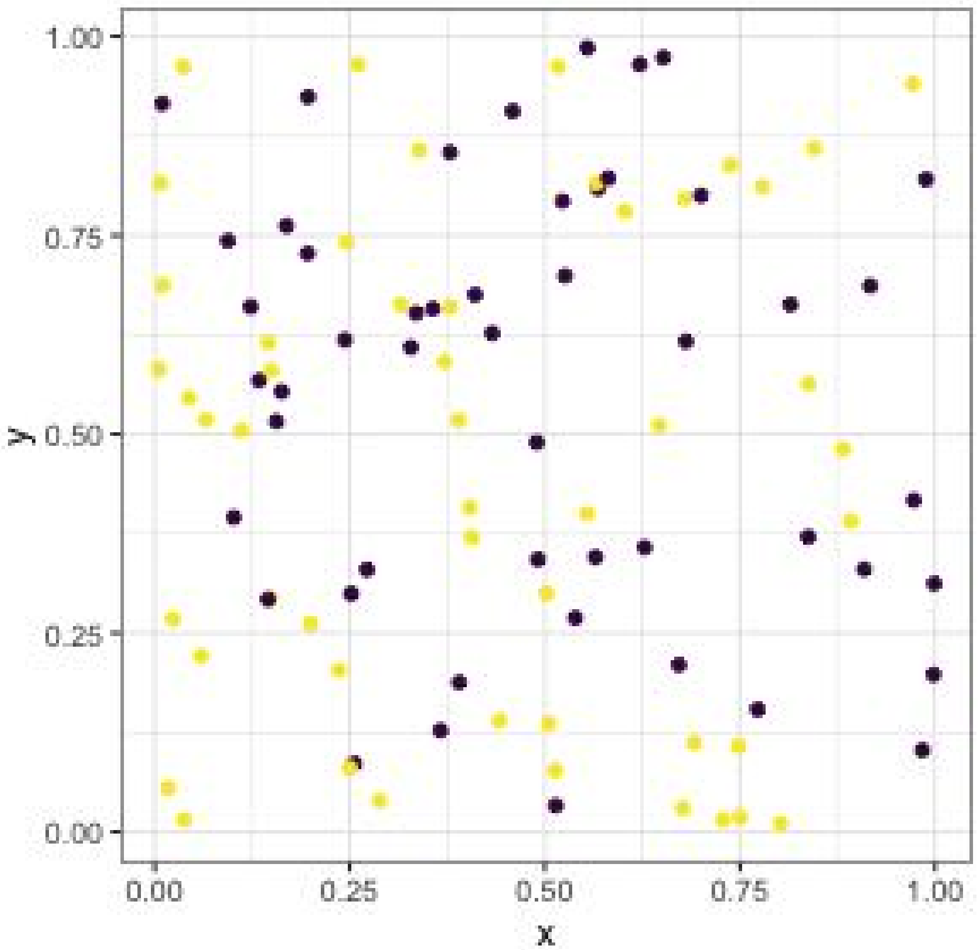


Impact of colourblind-safe palette

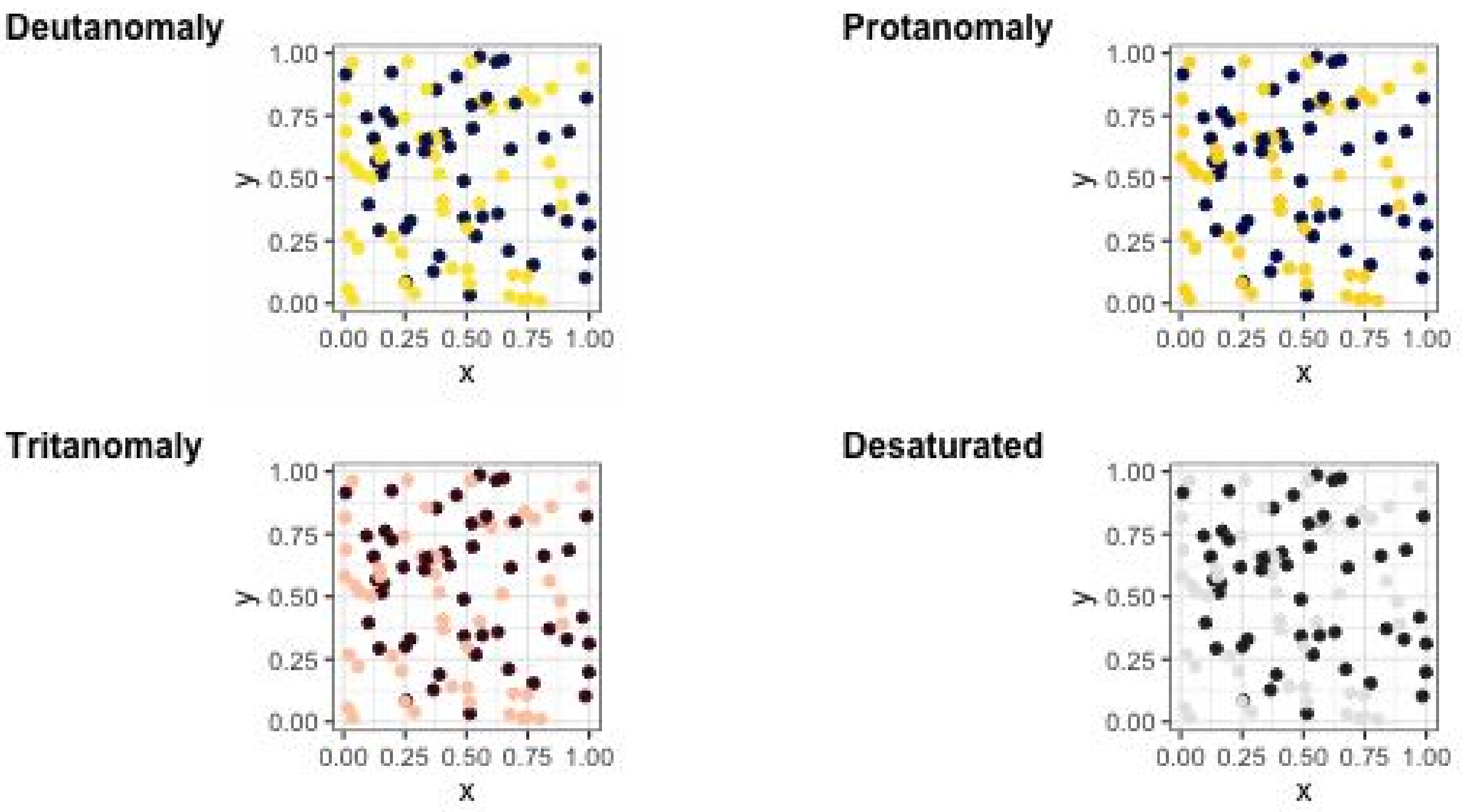


Impact of colourblind-safe palette

```
p3 <- p + scale_colour_viridis_d()
p3
```



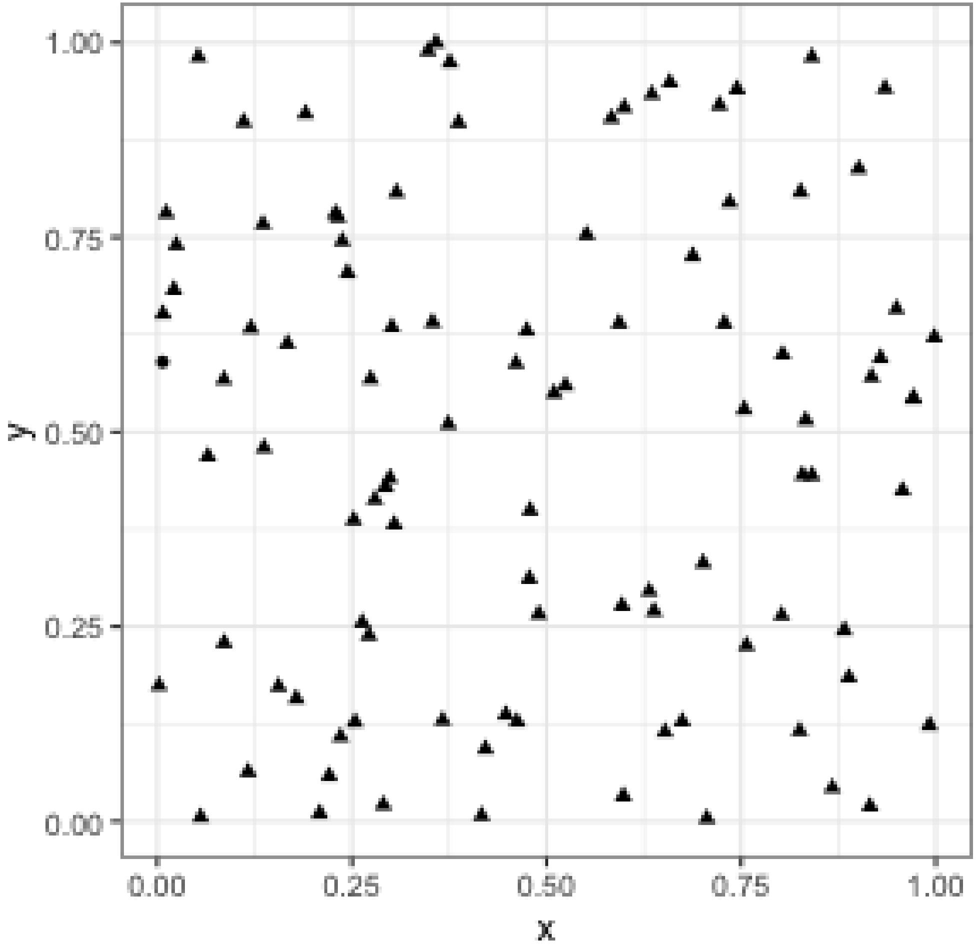
Impact of colourblind-safe palette



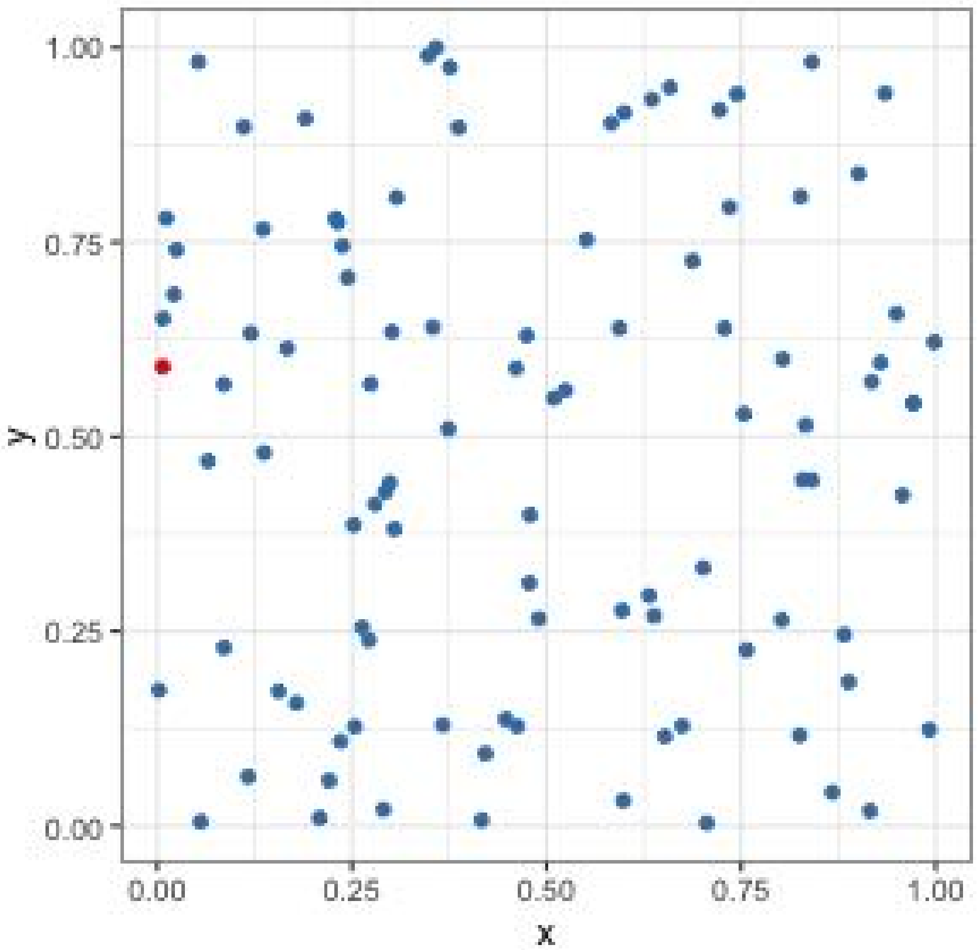
Summary colour blindness

- Apply colourblind-friendly colourscales
  - `+ scale_colour_viridis()`
  - `+ scale_colour_brewer(palette = "Dark2")`
  - `scico` R package

Pre-attentiveness: Find the odd one out?



Pre-attentiveness: Find the odd one out?



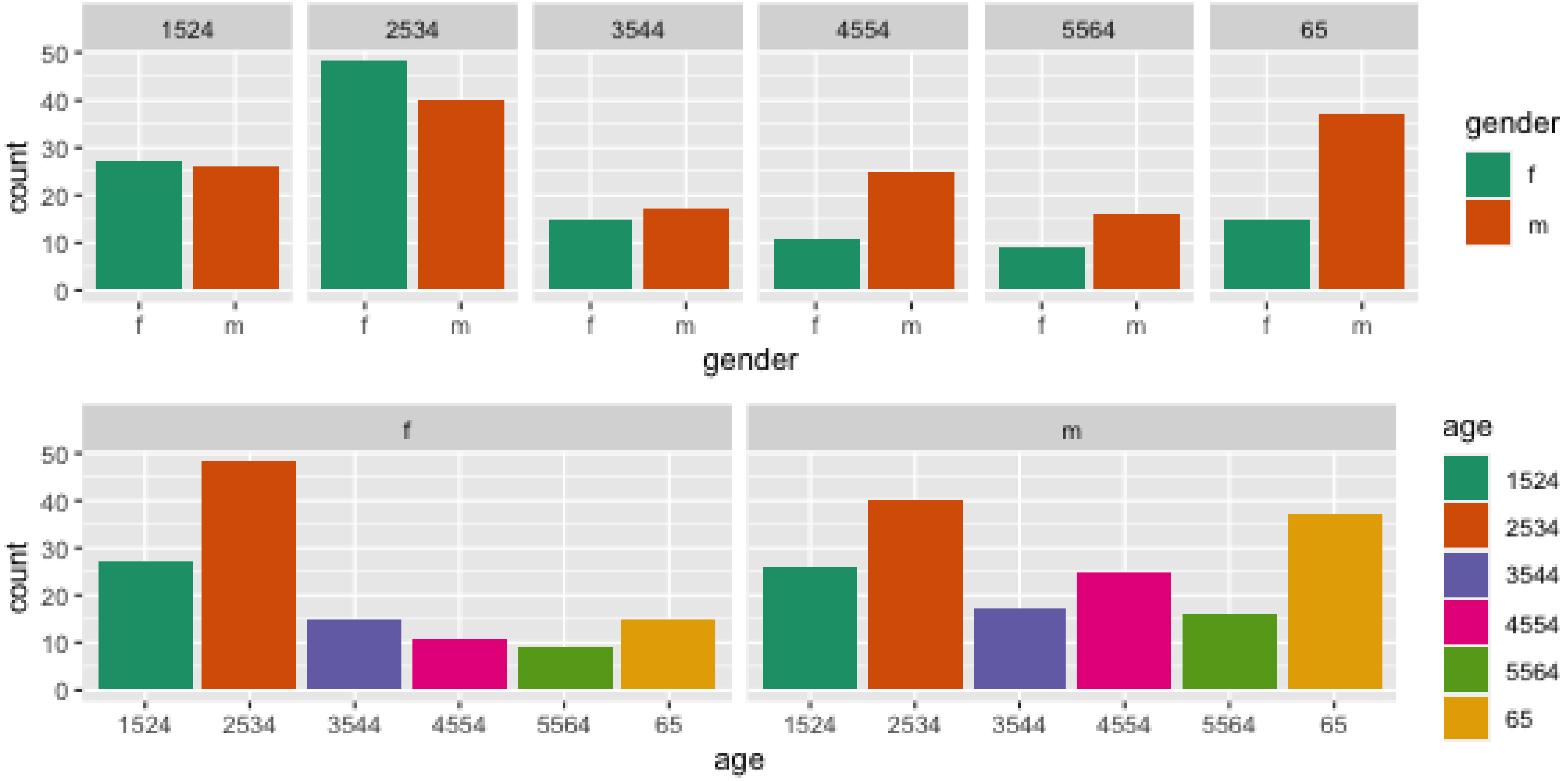
Using proximity in your plots

Basic rule: place the groups that you want to compare close to each other

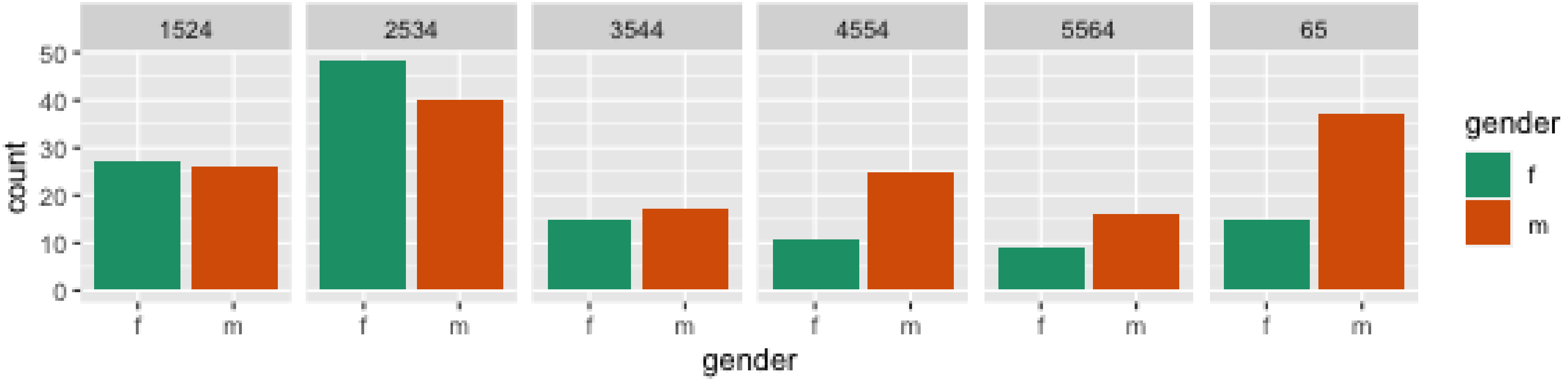
Which plot answers which question?

- "Is the incidence similar for males and females in 2012 across age groups?"
- "Is the incidence similar for age groups in 2012, across gender?"

incidence similar for: (M and F) or (age, across gender) ?"

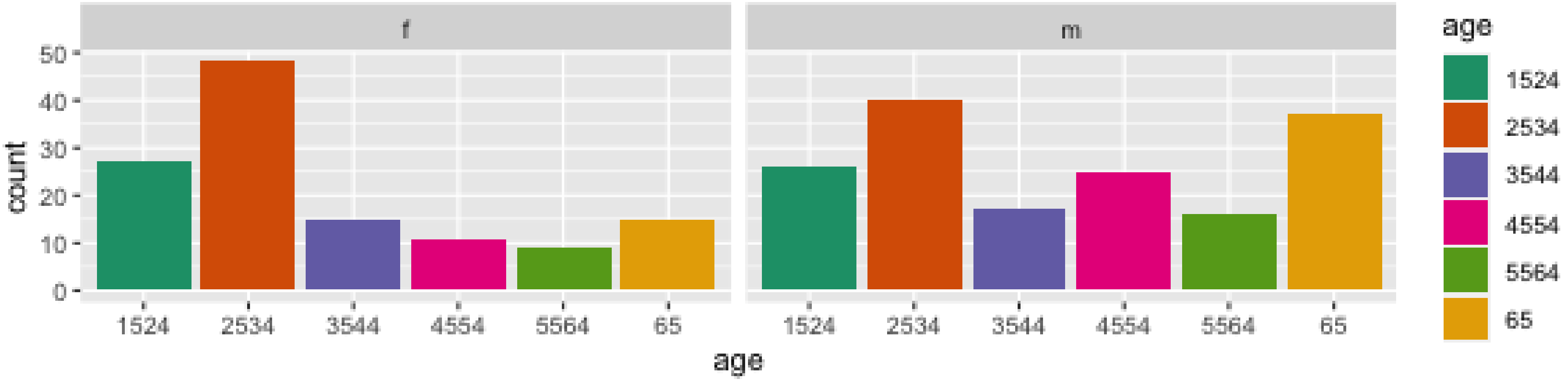


"Incidence similar for M & F in 2012 across age?"



- Males & females next to each other: relative heights of bars is seen quickly.
- Auestion answer: "No, the numbers were similar in youth, but males are more affected with increasing age."

"Incidence similar for age in 2012, across gender?"



- Puts the focus on age groups
- Answer to the question: "No, among females, the incidence is higher at early ages. For males, the incidence is much more uniform across age groups."

Proximity wrap up

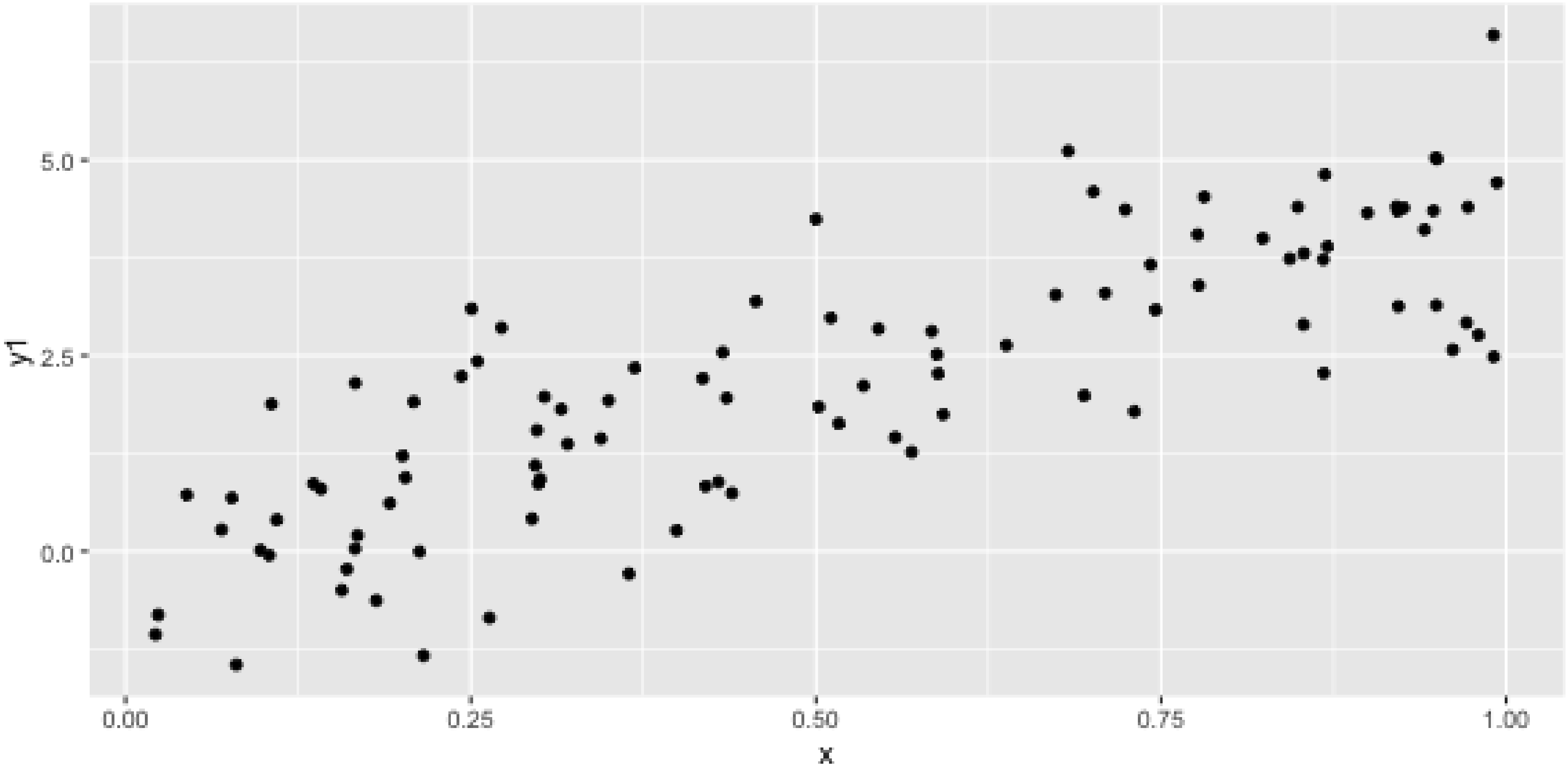
- Facetting of plots, and proximity are related to change blindness, an area of study in cognitive psychology.
- There are a series of fabulous videos illustrating the effects of making a visual break, on how the mind processes it by Daniel Simons lab.
- Here's one example:
  The door study

Layering

- *Statistical summaries:* It is common to layer plots, particularly by adding statistical summaries, like a model fit, or means and standard deviations. The purpose is to show the **trend** in relation to the **variation**.
- *Maps:* Commonly maps provide the framework for data collected spatially. One layer for the map, and another for the data.

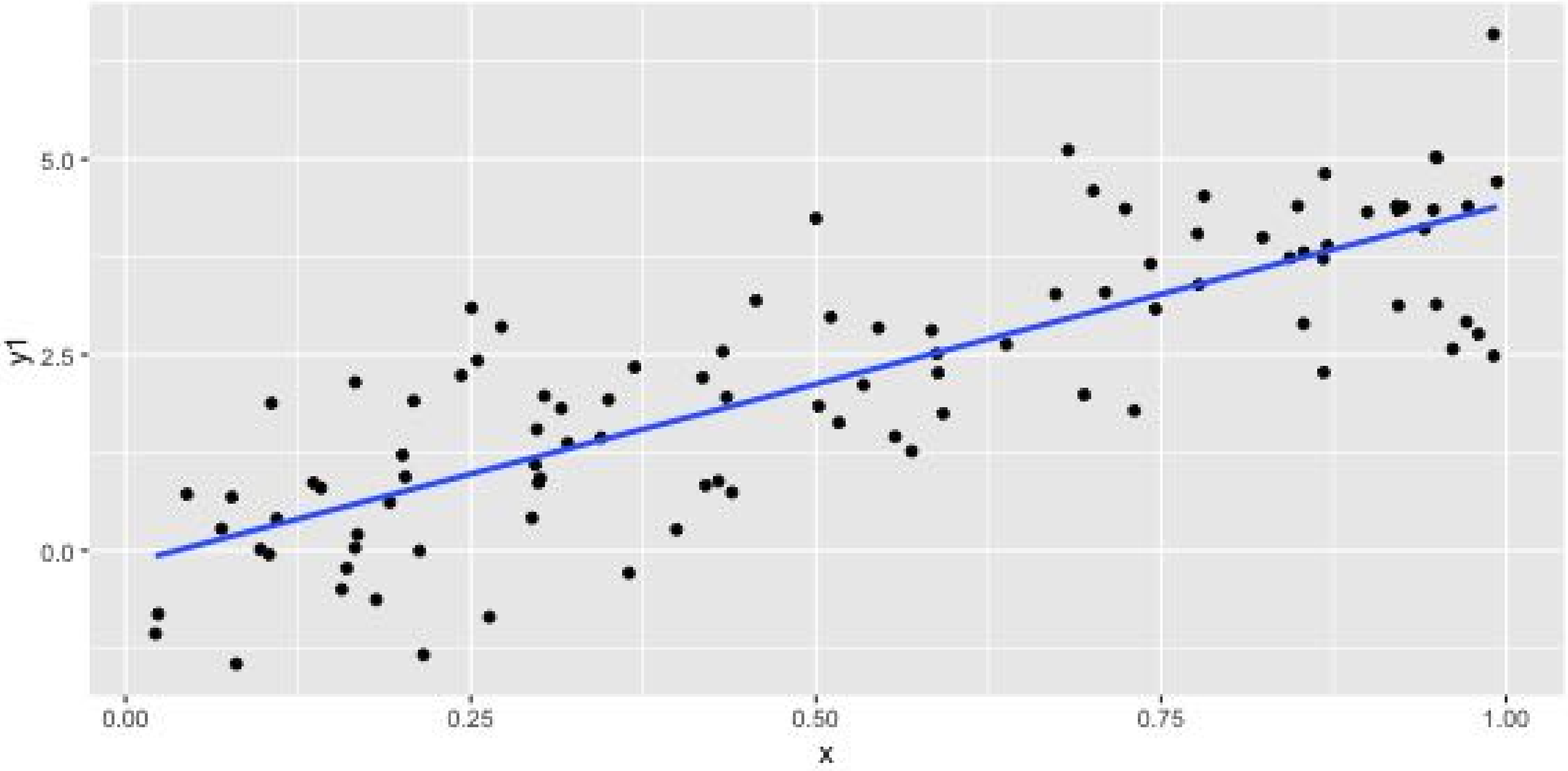
geom_point()

```
ggplot(df, aes(x = x, y = y1)) + geom_point()
```



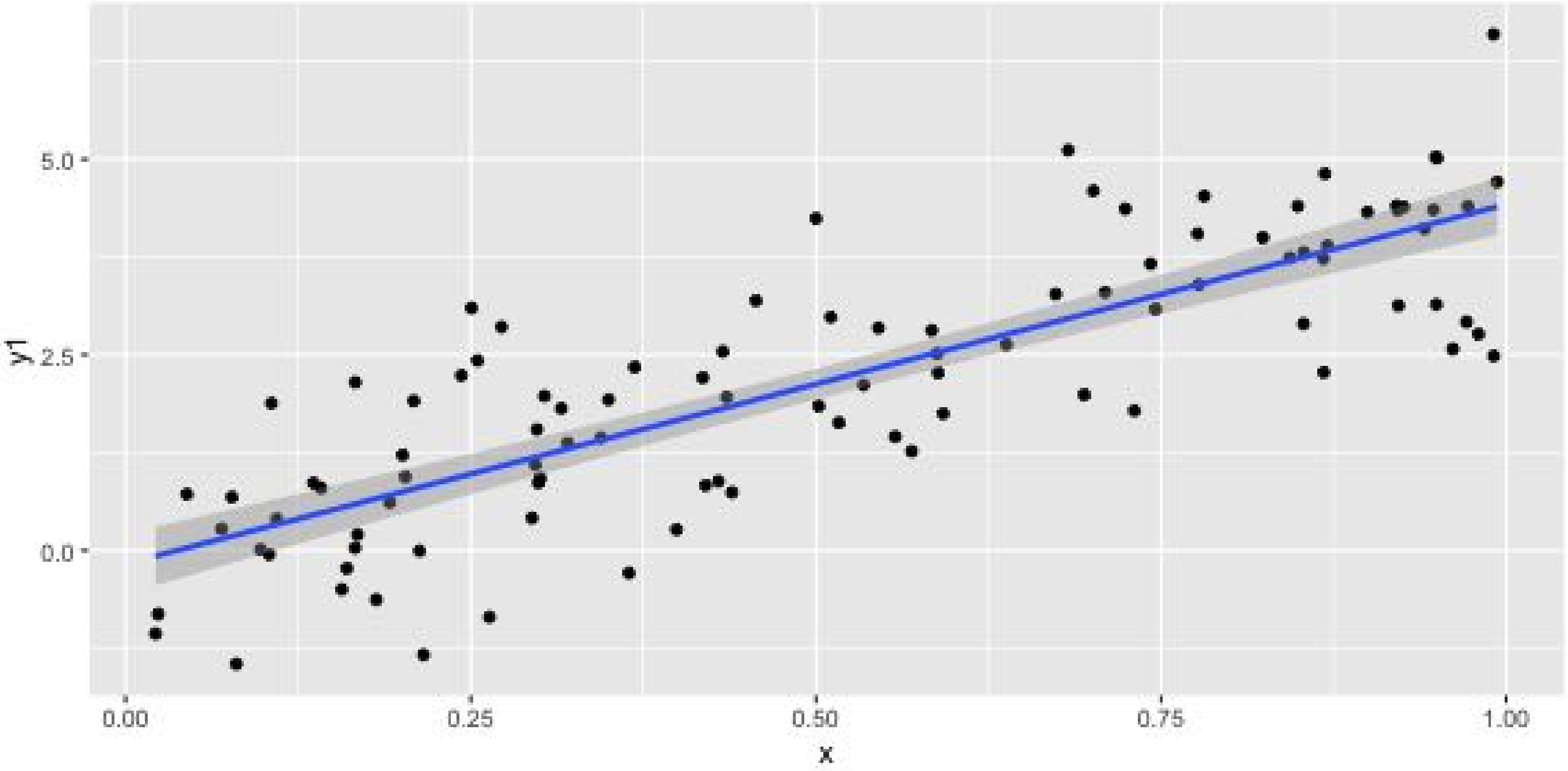
geom_smooth(method = "lm", se = FALSE)

```
ggplot(df, aes(x = x, y = y1)) + geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```



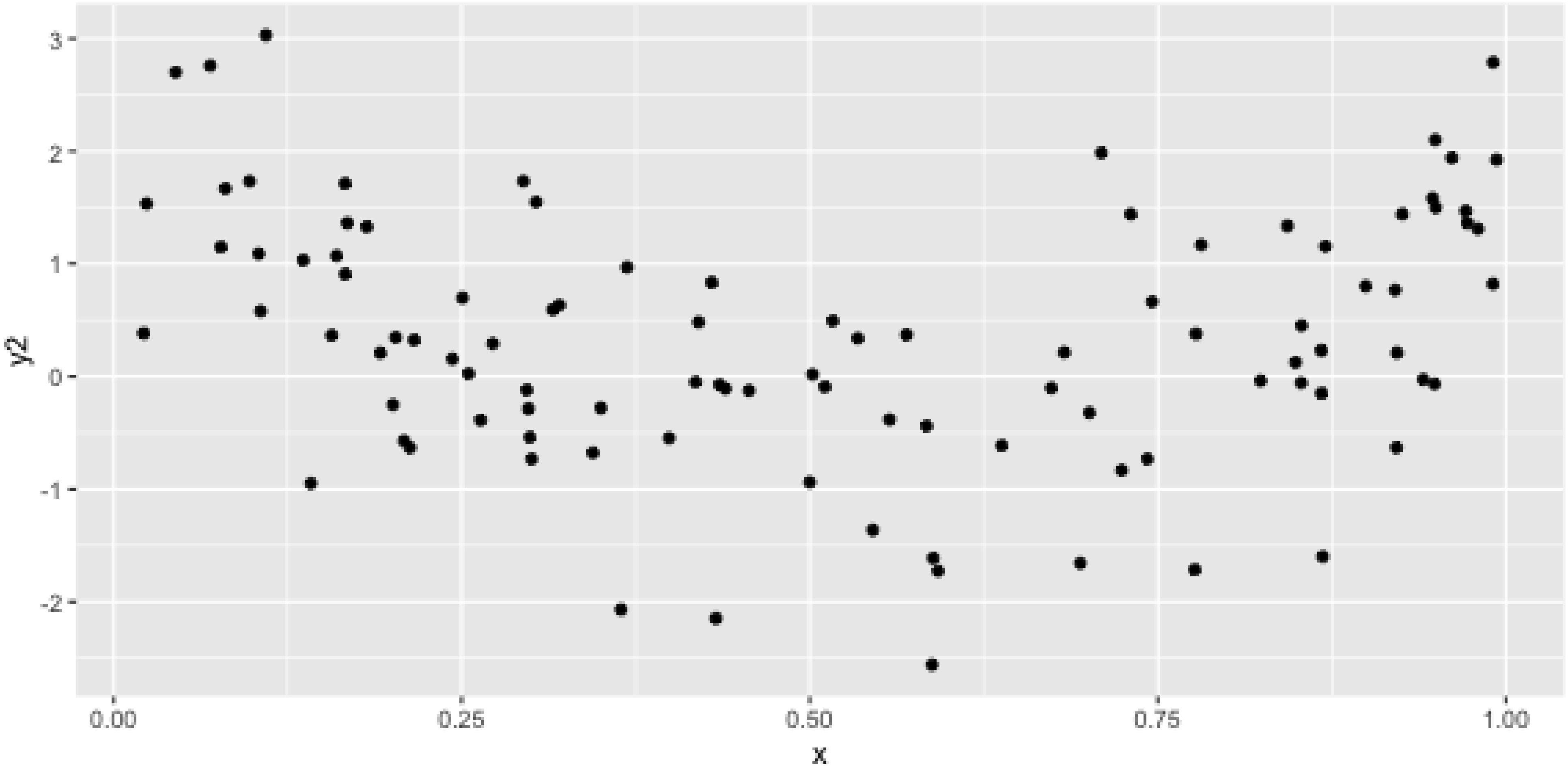
geom_smooth(method = "lm")

```
ggplot(df, aes(x = x, y = y1)) + geom_point() +
  geom_smooth(method = "lm")
```



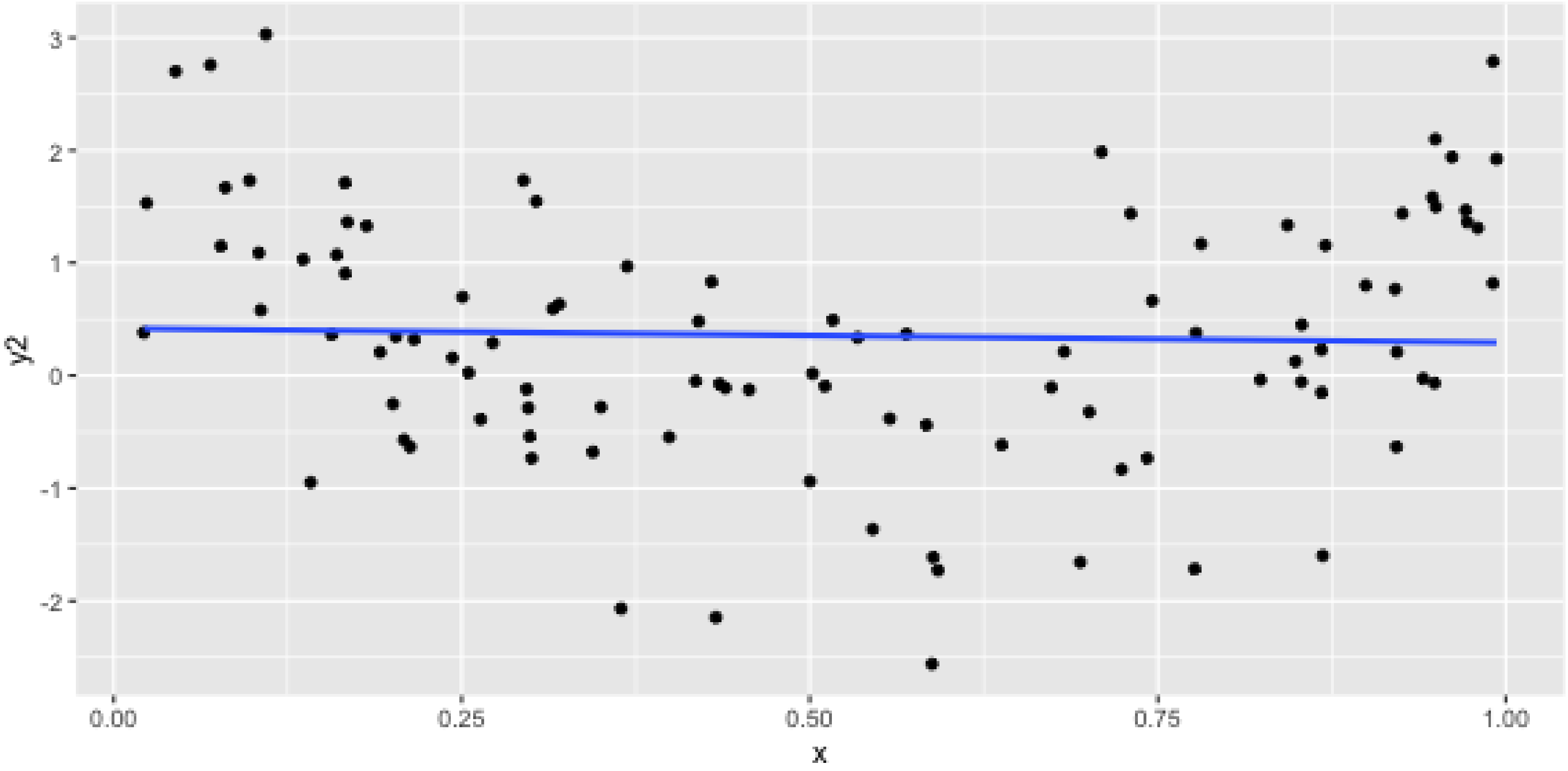
geom_point()

```
ggplot(df, aes(x = x, y = y2)) + geom_point()
```



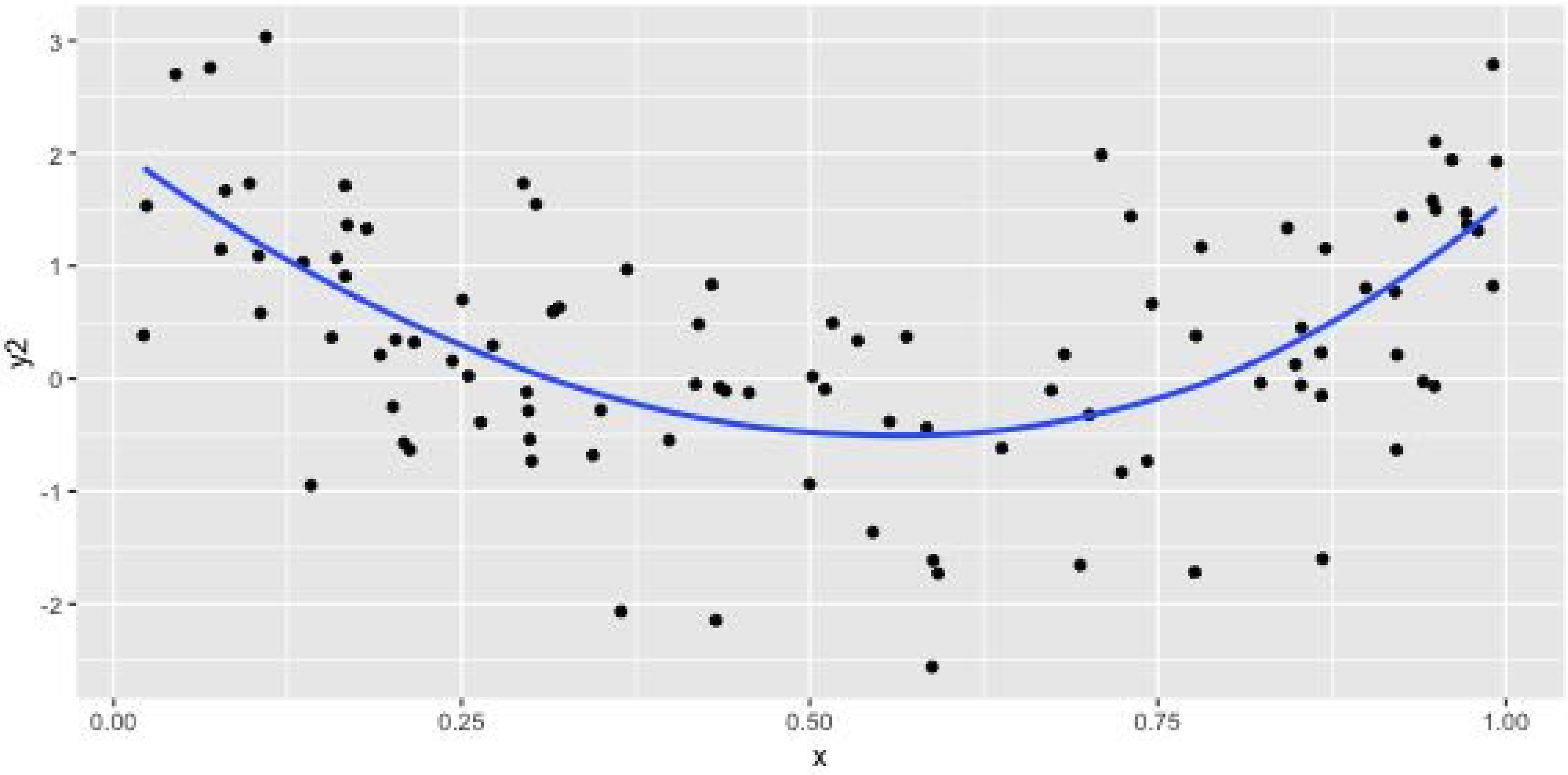
geom_smooth(method = "lm", se = FALSE)

```
ggplot(df, aes(x = x, y = y2)) + geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```



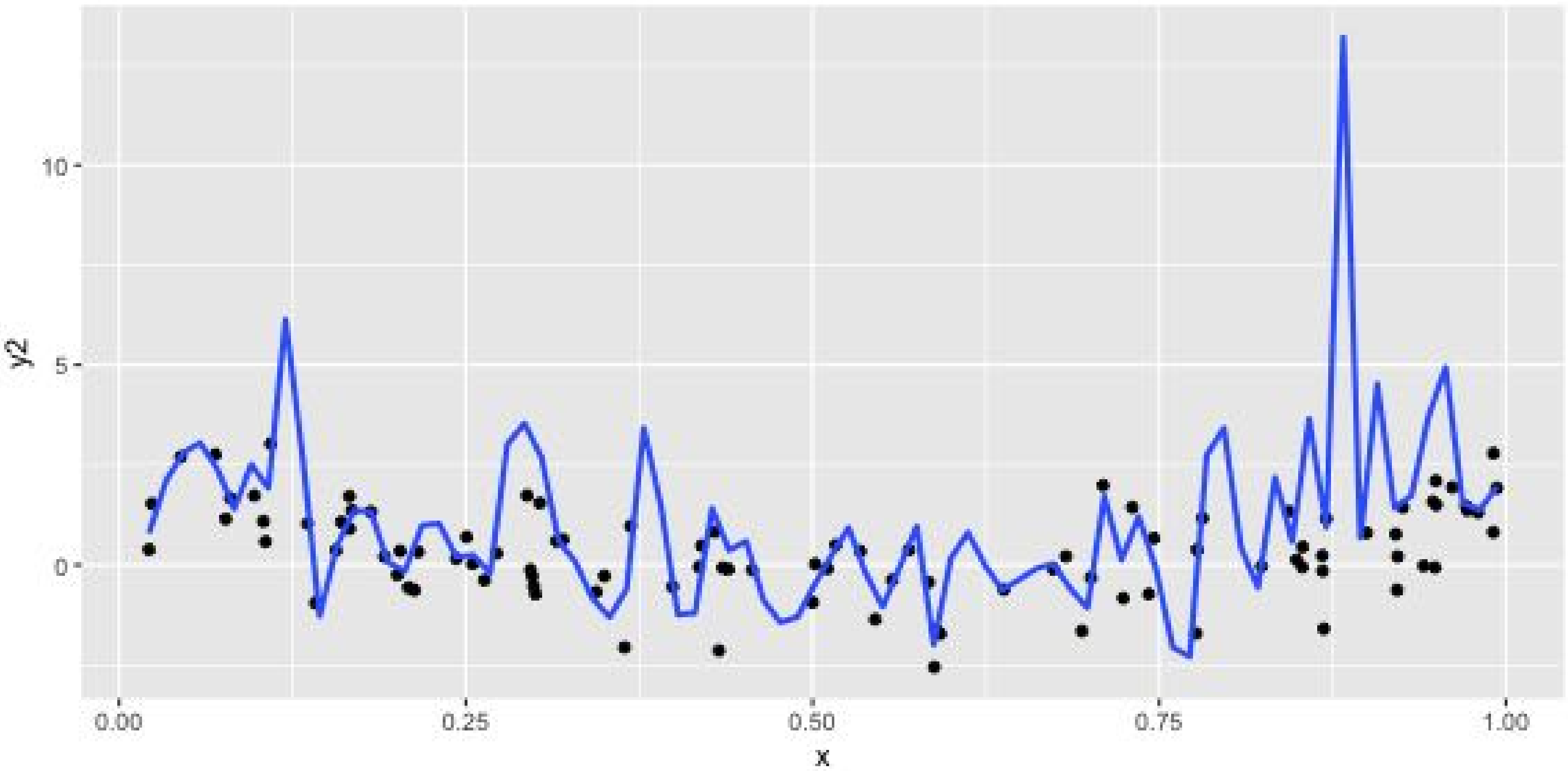
geom_smooth(se = FALSE)

```
ggplot(df, aes(x = x, y = y2)) + geom_point() +
  geom_smooth(se = FALSE)
```



geom_smooth(se = FALSE, span = 0.05)

```
ggplot(df, aes(x = x, y = y2)) + geom_point() +
  geom_smooth(se = FALSE, span = 0.05)
```



geom_smooth(se = FALSE, span = 0.2)

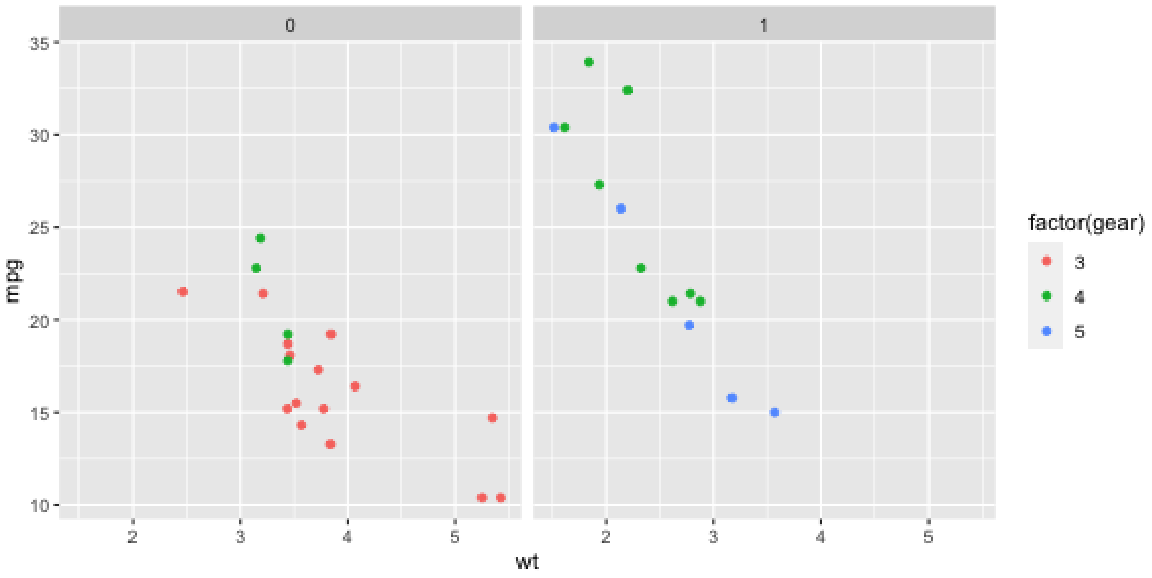
```r
p1 <- ggplot(df, aes(x = x, y = y2)) + geom_point() +
  geom_smooth(se = FALSE, span = 0.2)
p1
```

Interactivity with magic plotly

```r
library(plotly)
ggplotly(p1)
```

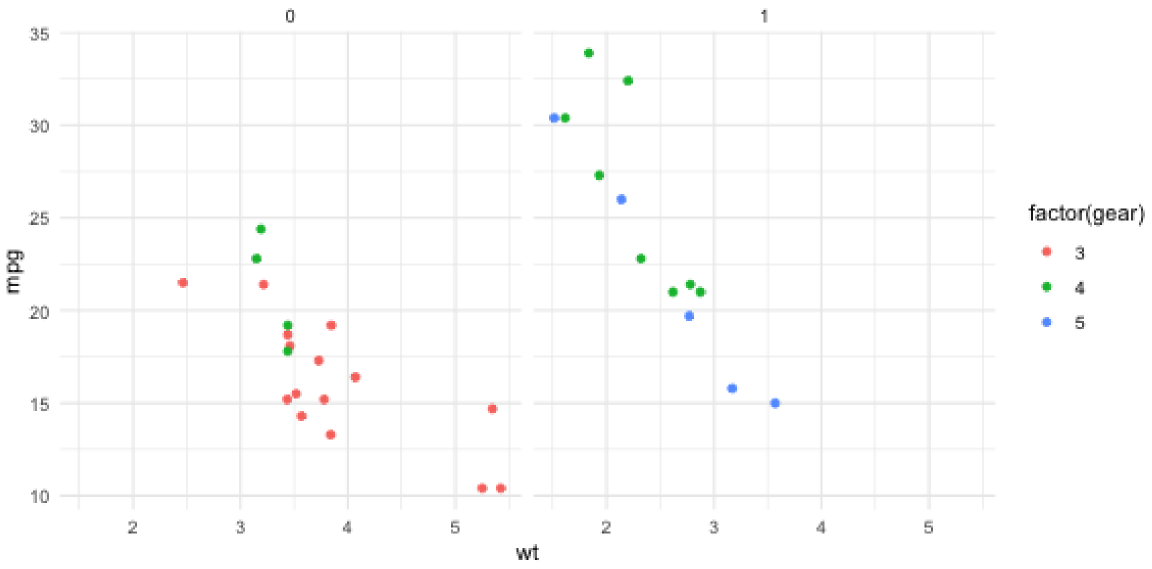
Themes: Add some style to your plot

```
p <- ggplot(mtcars) +
  geom_point(aes(x = wt,
                 y = mpg,
                 colour = factor
  facet_wrap(~am)
p
```



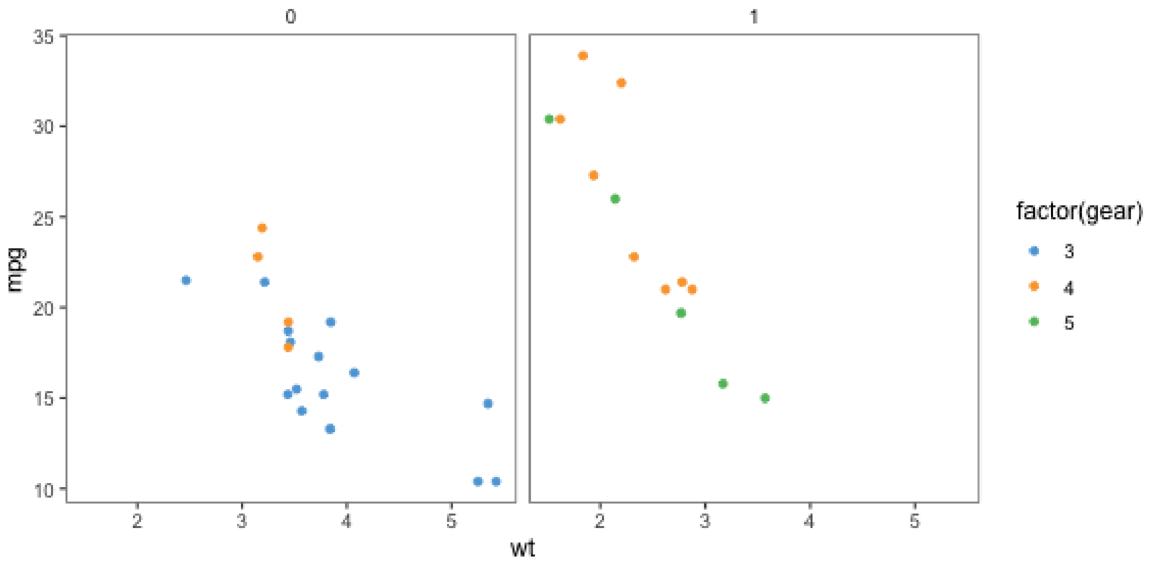
Theme: theme_minimal

```
p +
  theme_minimal()
```



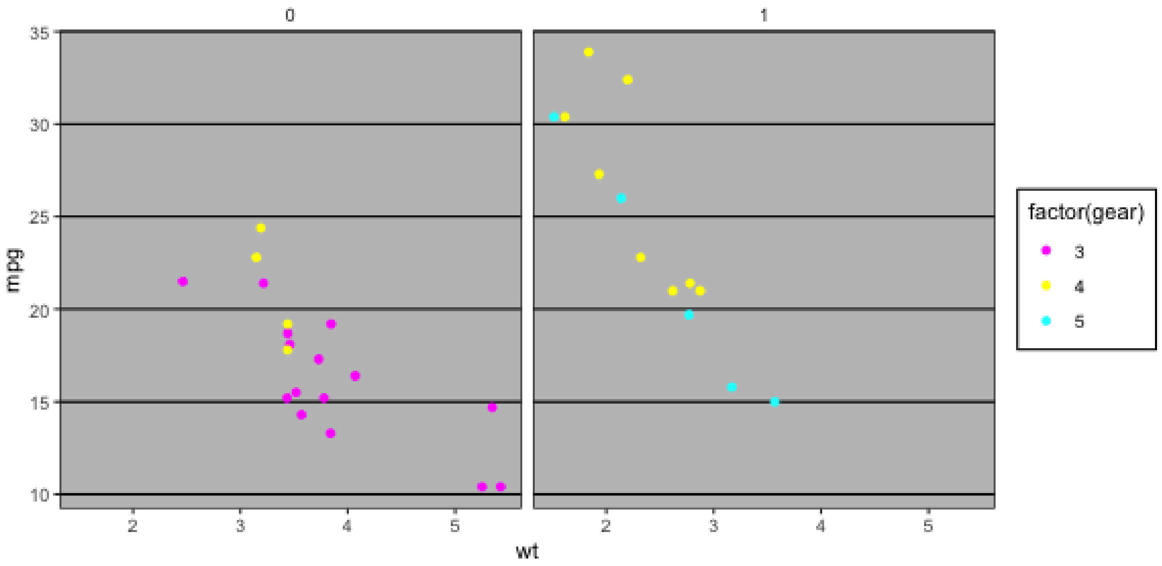
Theme: ggthemes `theme_few()`

```
p +
  theme_few() +
  scale_colour_few()
```



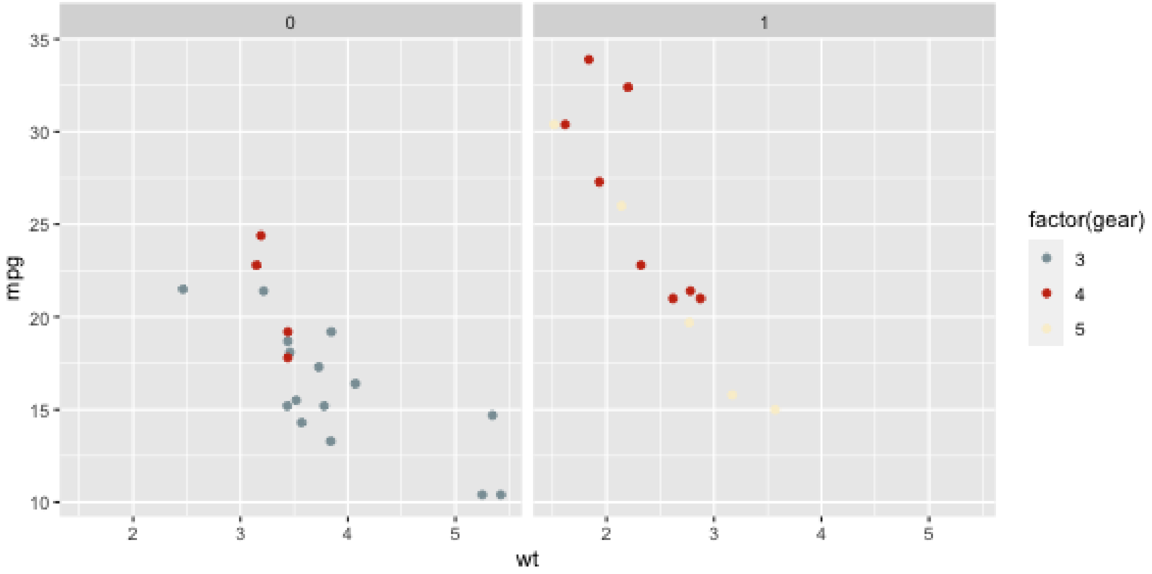
Theme: ggthemes `theme_excel()` 💊

```
p +
  theme_excel() +
  scale_colour_excel()
```



Theme: for fun

```
library(wesanderson)
p +
  scale_colour_manual(
    values = wes_palette("Royal1
    )
```



Summary: themes

- The `ggthemes` package has many different styles for the plots.
- Other packages such as `xkcd`, `skittles`, `wesanderson`, `beyonce`, `ochre`, ...

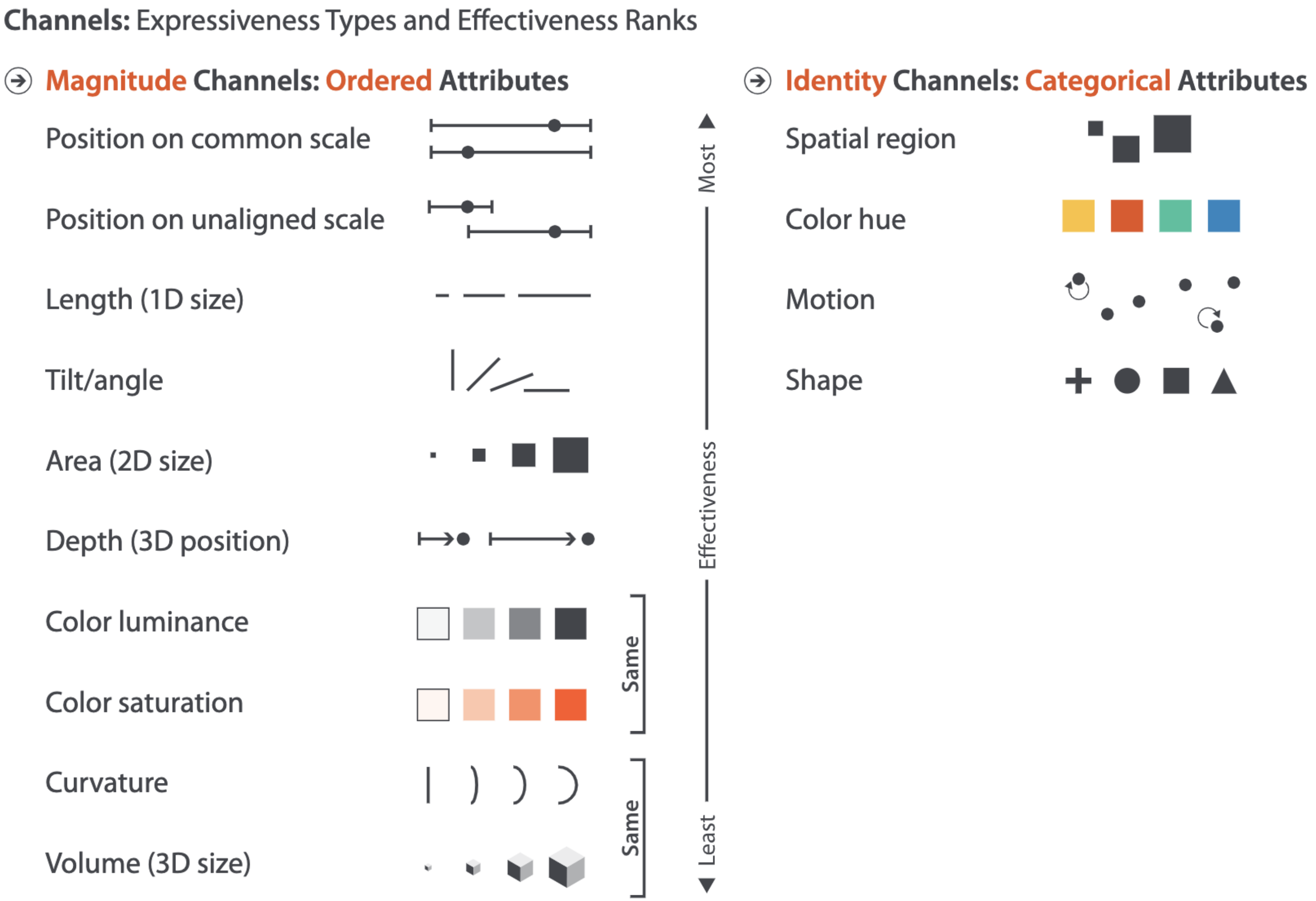


Figure 5.1. The effectiveness of channels that modify the appearance of marks depends on matching the expressiveness of channels with the attributes being encoded.

Hierarchy of mappings

1. Position - common scale (BEST): axis system
2. Position - nonaligned scale: boxes in a side-by-side boxplot
3. Length, direction, angle: pie charts, regression lines, wind maps
4. Area: bubble charts
5. Volume, curvature: 3D plots
6. Shading, color (WORST): maps, points coloured by numeric variable

- Di's crowd-sourcing expt
- Nice explanation by Peter Aldous
- General plotting advice and a book from Naomi Robbins

Your Turn:

- lab quiz open (requires answering questions from Lab exercise)
- go to rstudio.cloud and check out exercise 4-B
- If you want to use R / Rstudio on your laptop:
  - Install R + Rstudio (see Stuart Lee's instructions)
  - open R
  - type the following:

```r
# install.packages("usethis")
library(usethis)
use_course("https://ida.numbat.space/exercises/4b/ida-exercise-4b.zip")
```

Resources

- Kieran Healy Data Visualization
- Winston Chang (2012) Cookbook for R
- Antony Unwin (2014) Graphical Data Analysis
- Naomi Robbins (2013) Creating More Effective Charts