BDSI R Workshop – Data Visualisation

Robert Cope and Terry Neeman

2022

## Data setup

We will use the same breast cancer data as the first session.

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ readr 2.1.2 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

cancer\_df <- read\_csv('data/Cuzick\_2010\_breast\_cancer\_density.csv')

## Rows: 1065 Columns: 5  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## dbl (5): case, ARM, AGE, BMI, density  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

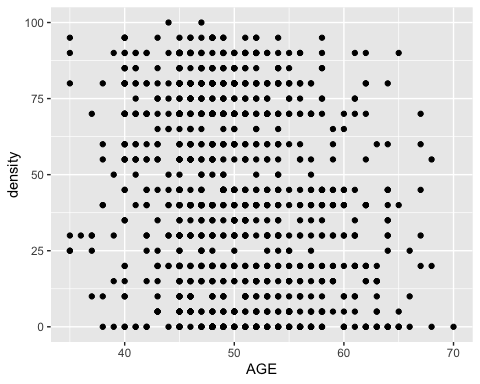
glimpse(cancer\_df)

## Rows: 1,065  
## Columns: 5  
## $ case <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,…  
## $ ARM <dbl> 1, 1, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2,…  
## $ AGE <dbl> 38, 43, 46, 52, 59, 62, 35, 58, 51, 40, 66, 54, 47, 46, 41, 61…  
## $ BMI <dbl> 21.8, 32.3, 23.0, 19.6, 26.2, 23.7, 27.9, 25.8, 27.7, 38.4, 32…  
## $ density <dbl> 40, 5, 45, 40, 40, 80, 25, 15, 10, 20, 0, 0, 0, 80, 80, 10, 95…

## ggplot2

Let us start by making a plot.

ggplot(data = cancer\_df, mapping = aes(x = AGE, y = density)) +  
 geom\_point()



A plot always needs (at least) these three components. 1. data (in the form of a tibble or data frame); 2. a mapping – the aes() function, which stands for aesthetic, that tells R how to map data coordinates to the plot; and, 3. a geometry (which always starts with geom\_) which determines the shape of the plot.

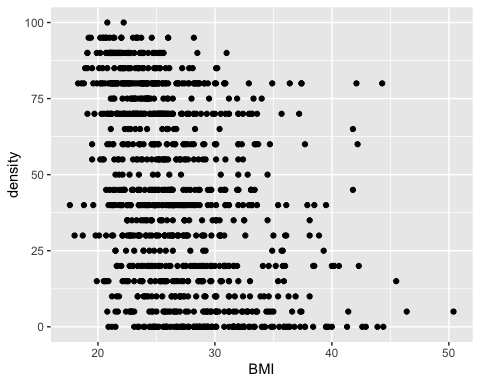
### Exercise – what do you notice about this plot?

* the points are on a grid!! i.e., both age and density take very specific values.
* more subtle, the points are almost certainly on top of each other.

### Exercise – plot density on the y axis and BMI on the x axis.

ggplot(data = cancer\_df, mapping = aes(x = BMI, y = density)) +  
 geom\_point()

## Warning: Removed 16 rows containing missing values (geom\_point).



An important tool is saving your plots. Create a ‘figs’ folder. Save your most recent plot with ggsave() (you can do this in different formats). Note that width and height are in inches.

ggsave('figs/BMI\_density.png', width = 6, height = 4)

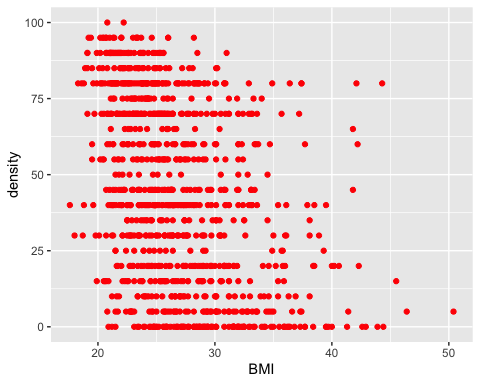
If you look at the help file for your geom (or the cheatsheet), you can see what aesthetics you can modify.

?geom\_point

There are two things we can do. If we just want to do something to every point, we don’t include it in an aes() statement, and we just set it to what we want. E.g.,

ggplot(data = cancer\_df, mapping = aes(x = BMI, y = density)) +  
 geom\_point(colour = 'red')

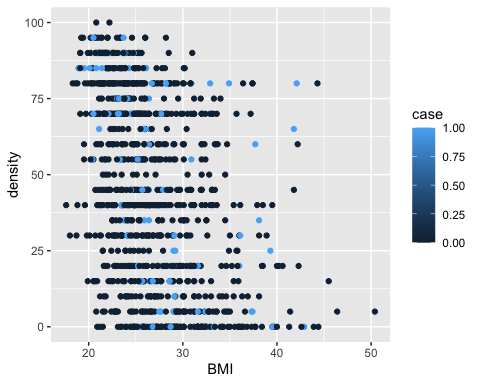
## Warning: Removed 16 rows containing missing values (geom\_point).



But if we want it to depend on the data, it has to go in the aes() statement.

ggplot(data = cancer\_df, mapping = aes(x = BMI, y = density, colour = case)) +  
 geom\_point()

## Warning: Removed 16 rows containing missing values (geom\_point).



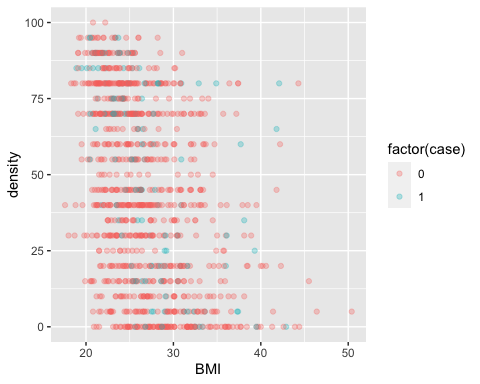
Note that in the first case, we used quotation marks because it was a string, whereas in the second there were no quotations because it is a variable.

Note also that you get a legend by default.

### Exercise – change the transparency (alpha) of the points (to the same, fixed value)

ggplot(data = cancer\_df, mapping = aes(x = BMI, y = density, colour = factor(case))) +  
 geom\_point(alpha = 0.3)

## Warning: Removed 16 rows containing missing values (geom\_point).

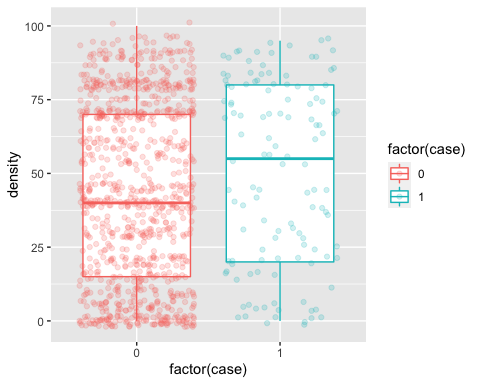


(This plot is terrible/confusing but it does show you what you can do)

## Box plots

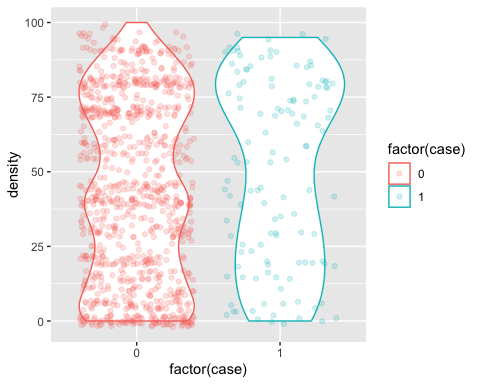
Boxplots were developed to summarise the overall distribution of the data. In this case-control study, we may be interested to explore whether cases have on average lower breast density than Control patients. Boxplots can offer some insight into this question. We can super-impose the data over the boxplots.

ggplot(cancer\_df, aes(x= factor(case), y = density, col = factor(case)))+  
 geom\_boxplot()+  
 geom\_jitter(alpha = 0.2)



A boxplot is a five-number summary of a data distribution, and may not capture the salient features of a distribution (e.g. if the distribution is bimodal). Violin plots can be better at showing some of the “local” concentration areas of a distribution. What do you think?

ggplot(cancer\_df, aes(x= factor(case), y = density, col = factor(case)))+  
 geom\_violin()+  
 geom\_jitter(alpha = 0.2)



### More on factors

One thing you will notice about these data is that case and ARM are numbers, but they always take specific values. This is because they are actually factors! For case, 1 indicates they were positive for breast cancer within the following year, and zero indicates they were not. For ARM, 1 indicates they received the placebo and 2 indicates they received the Tamoxifen. This is a relatively common way to create data (though alternatively the authors could have used text labels in their spreadsheet which may have been cleaer). We should recode these, and make them factors.

cancer\_df <- cancer\_df %>%  
 mutate(case\_f = factor(case, labels = c("Control", "Case")),  
 ARM\_f = factor(ARM, labels = c("Placebo", "Tamoxifen")))  
str(cancer\_df)

## tibble [1,065 × 7] (S3: tbl\_df/tbl/data.frame)  
## $ case : num [1:1065] 1 0 0 0 0 0 0 0 0 0 ...  
## $ ARM : num [1:1065] 1 1 1 2 1 1 2 1 1 2 ...  
## $ AGE : num [1:1065] 38 43 46 52 59 62 35 58 51 40 ...  
## $ BMI : num [1:1065] 21.8 32.3 23 19.6 26.2 23.7 27.9 25.8 27.7 38.4 ...  
## $ density: num [1:1065] 40 5 45 40 40 80 25 15 10 20 ...  
## $ case\_f : Factor w/ 2 levels "Control","Case": 2 1 1 1 1 1 1 1 1 1 ...  
## $ ARM\_f : Factor w/ 2 levels "Placebo","Tamoxifen": 1 1 1 2 1 1 2 1 1 2 ...

Once you’ve re-coded, you should check that you did it correctly.

with(cancer\_df, table(case, case\_f))

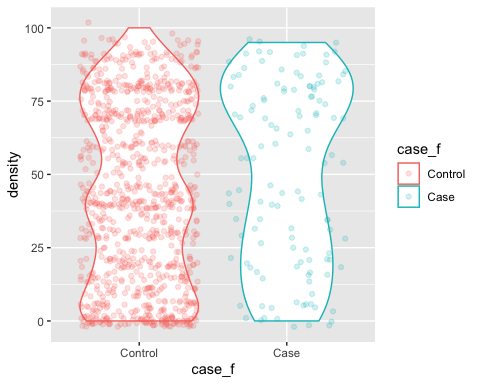
## case\_f  
## case Control Case  
## 0 942 0  
## 1 0 123

with(cancer\_df, table(ARM, ARM\_f))

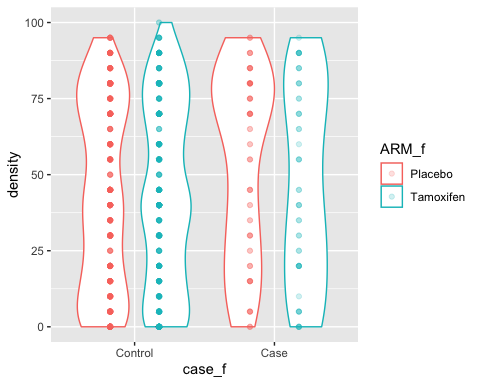
## ARM\_f  
## ARM Placebo Tamoxifen  
## 1 558 0  
## 2 0 507

We can make the boxplot clearer by using the factors to label the violin:

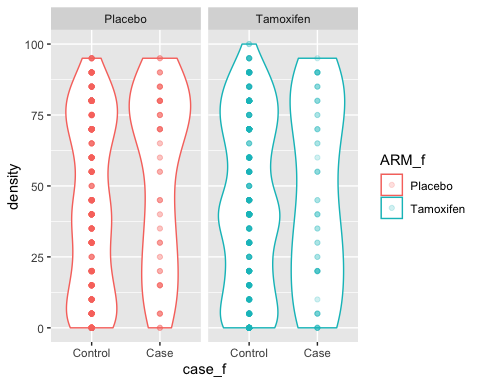
ggplot(cancer\_df, aes(x= case\_f, y = density, col = case\_f))+  
 geom\_violin()+  
 geom\_jitter(alpha = 0.2)

 There are actually 4 populations: Cases and controls for each of the treatment groups. Explore how we can visualise the four density distributions. Here are some ideas. Do any of these exploratory plots help us understand if density distribution is different between cases and controls?

ggplot(cancer\_df, aes(x= case\_f, y = density, col = ARM\_f))+  
 geom\_violin()+  
 geom\_point(alpha = 0.2,position = position\_dodge(width = 0.7))

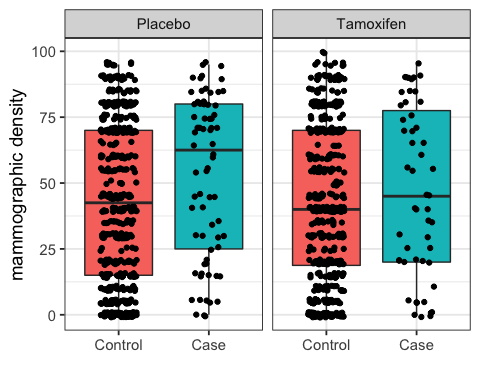


ggplot(cancer\_df, aes(x= case\_f, y = density, col = ARM\_f))+  
 geom\_violin()+  
 geom\_point(alpha = 0.2,position = position\_dodge(width = 0.7))+  
 facet\_wrap(~ARM\_f)



## Labels, legends, theme, etc.

ggplot(cancer\_df, aes(x=case\_f, y = density))+  
 geom\_boxplot(aes(fill = case\_f))+  
 geom\_jitter(width = 0.2, height = 1) +  
 facet\_wrap(~ARM\_f) +   
 theme\_bw(base\_size = 14)+  
 xlab('') + ylab('mammographic density') +   
 theme(legend.position = 'none')



Lets save it.

ggsave('figs/density\_by\_case\_and\_treatment.png',width = 6, height = 4)

The quickest reference (but it is quite dense) is the ggplot cheat sheet. I tend to find it by just googling ‘ggplot cheat sheet’ but you can find it here:

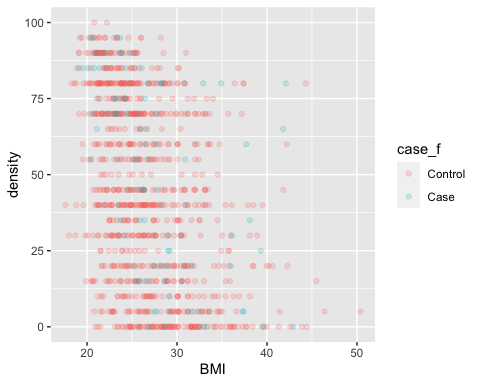
<https://github.com/rstudio/cheatsheets/blob/master/data-visualization-2.1.pdf>

## Group Exercise

Recall this figure:

ggplot(data = cancer\_df, mapping = aes(x = BMI, y = density, colour = case\_f)) +  
 geom\_point(alpha = 0.2)

## Warning: Removed 16 rows containing missing values (geom\_point).



at the moment it is not particularly informative. See if you can learn more about this trend by (a) trying different facets and (b) adding geom\_smooth() to the plot. Experiment with this and any layout settings you like until you produce something you are happy with, then save it. If you want to try something more complex, you could try the cut() function (check the help file) with the AGE variable.

Here are some questions you can explore:

Does breast density change as BMI increases? Is that association case-specific? or treatment-specific? Is that association similar for women of different ages?

### possible solutions

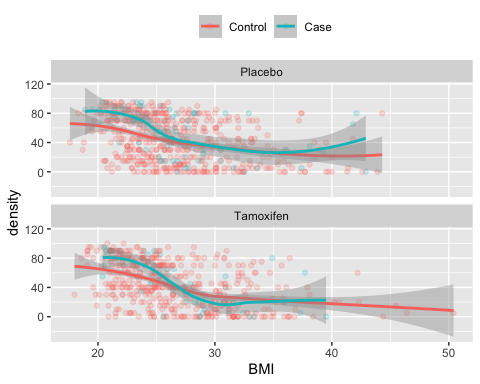
There are a few things we could do with this. geom\_smooth() adds a line of best fit to the data (there are various things you can do with this, the default allows a lot of flexibility).

ggplot(data = cancer\_df, mapping = aes(x = BMI, y = density, colour = case\_f)) +  
 geom\_point(alpha = 0.2) +  
 facet\_wrap(~ARM\_f, ncol = 1)+  
 theme(legend.title=element\_blank(), legend.position = 'top') +  
 geom\_smooth()

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 16 rows containing non-finite values (stat\_smooth).

## Warning: Removed 16 rows containing missing values (geom\_point).



That isn’t bad and maybe tells us something!? Perhaps we could also consider the impact of age as well. To do this, we might want to split age into bins because it will be hard to visualise in the current setup.

cancer\_df$age\_bin <- cut(cancer\_df$AGE,c(25,45,55,75))  
  
ggplot(data = cancer\_df, mapping = aes(x = BMI, y = density, colour = case\_f)) +  
 geom\_point(alpha = 0.2) +  
 facet\_grid(age\_bin~ARM\_f)+  
 theme(legend.title=element\_blank(), legend.position = 'top') +  
 geom\_smooth()

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

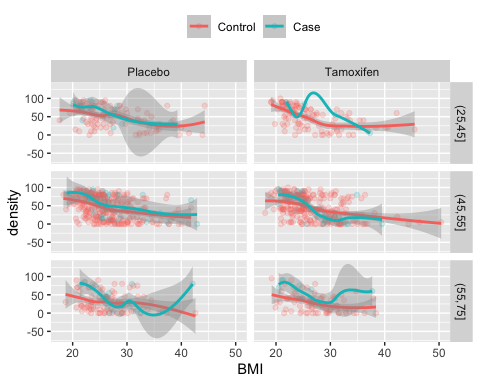
## Warning: Removed 16 rows containing non-finite values (stat\_smooth).

## Warning in sqrt(sum.squares/one.delta): NaNs produced

## Warning in stats::qt(level/2 + 0.5, pred$df): NaNs produced

## Warning: Removed 16 rows containing missing values (geom\_point).

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning  
## -Inf



This is a lot of information! And takes a bit to interpret. You should always try different things to get an idea of what you think could be better.

ggplot(data = cancer\_df, mapping = aes(x = BMI, y = density, colour = age\_bin)) +  
 geom\_point(alpha = 0.4) +  
 facet\_grid(case\_f~ARM\_f)+  
 theme(legend.title=element\_blank(), legend.position = 'top') +  
 geom\_smooth(alpha = 0.2) +   
 theme\_minimal()

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

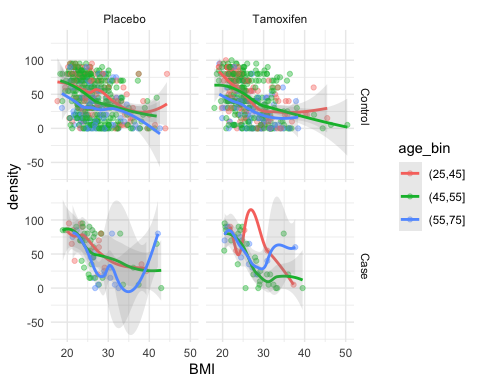
## Warning: Removed 16 rows containing non-finite values (stat\_smooth).

## Warning in sqrt(sum.squares/one.delta): NaNs produced

## Warning in stats::qt(level/2 + 0.5, pred$df): NaNs produced

## Warning: Removed 16 rows containing missing values (geom\_point).

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning  
## -Inf



This seems to indicate that older women have higher mammographic density. And it is a little hard to interpret what is going on with the treatment and cancer.

# Take home messages

* The learning curve is steep – but as you get used to the syntax + your data, it gets easier.
* Don’t rush to the final plot, build up what you want in steps.
* Think about what you want to show, try different ideas, iterate.
* consider the ggplot2 cheat sheet: <https://github.com/rstudio/cheatsheets/blob/master/data-visualization-2.1.pdf>
* a more detailed resource that may help is the R graphics cookbook: <https://r-graphics.org/>