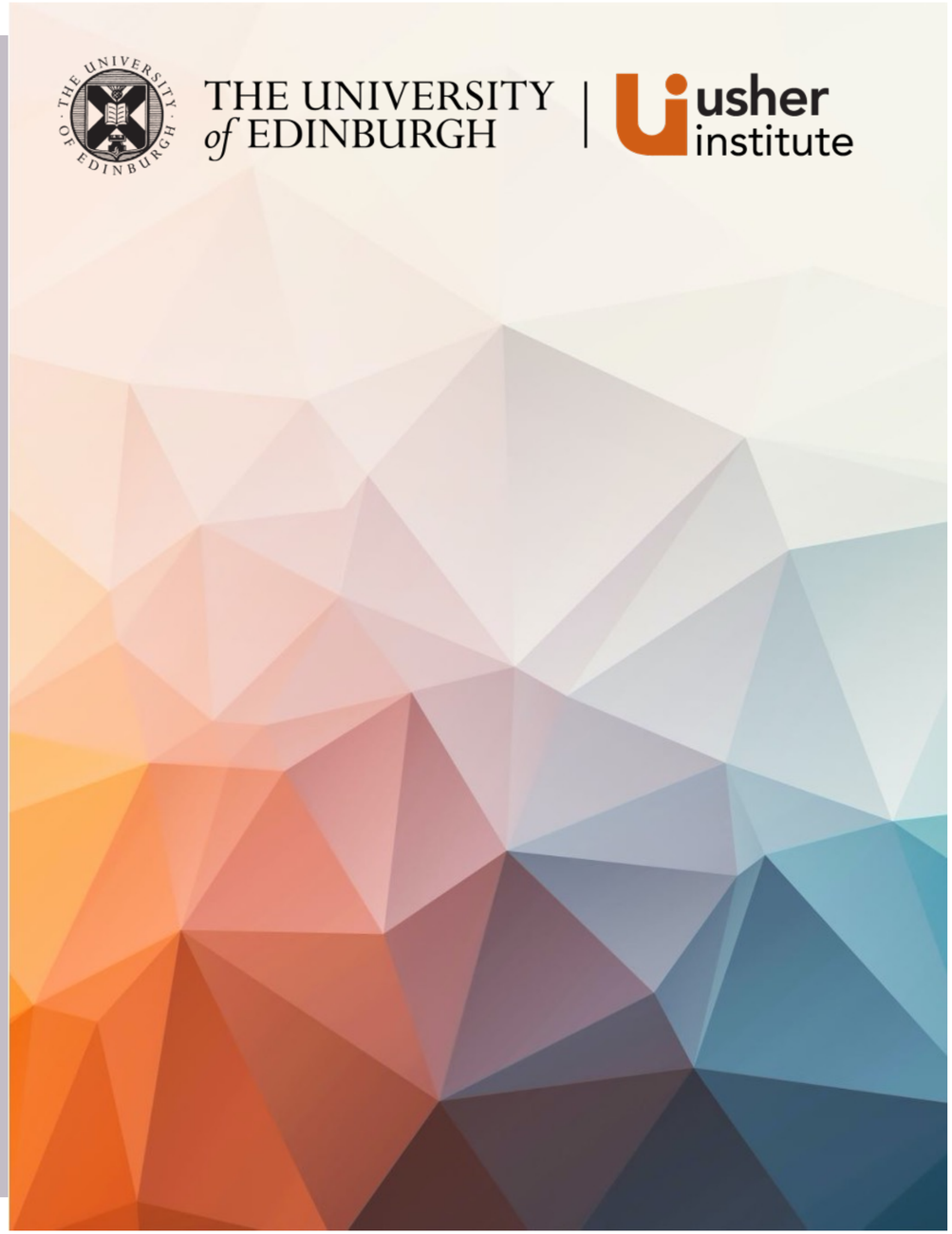
Introduction to data science in health and social care

Week 6

Brittany Blankinship | 2 November 2022



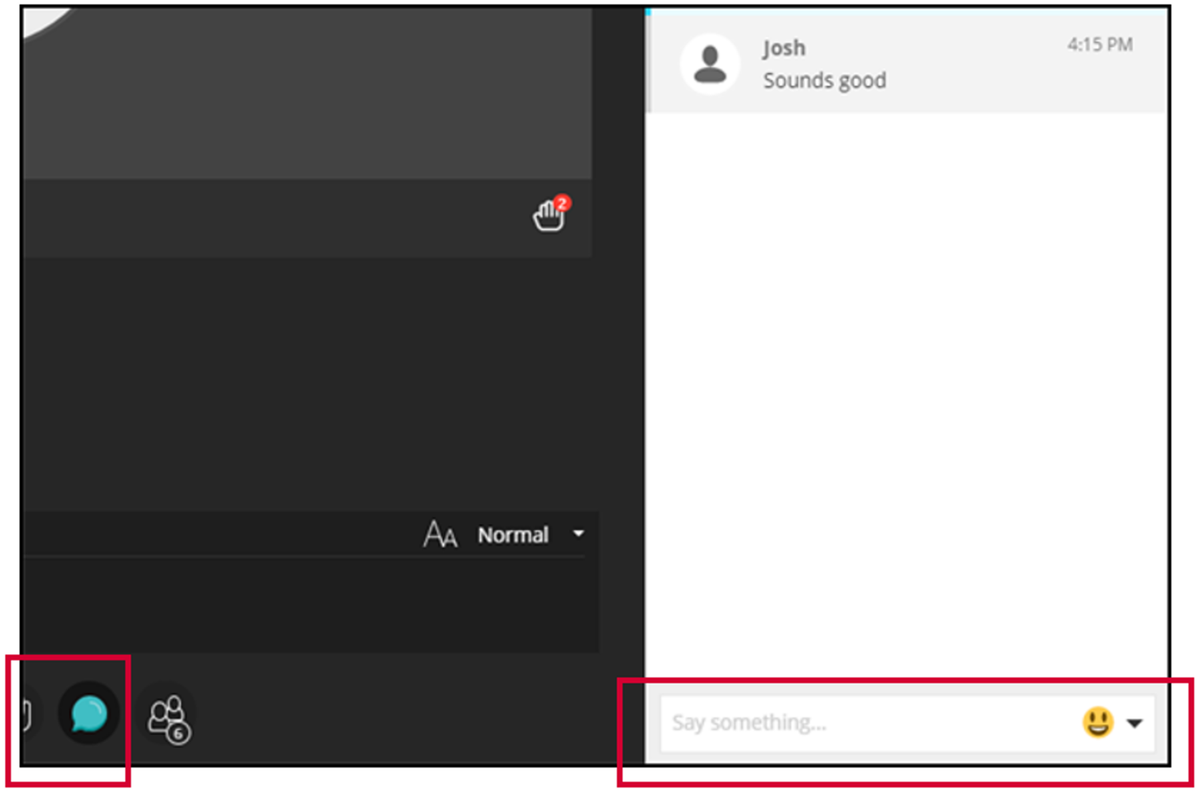
Audio check

Can you hear the presenter talking?

Please type **yes** or **no** in the “Text chat area”

If you can’t hear:

- Check your Audio/Visual settings in the Collaborate Panel
- Try signing out and signing back into the session
- Type into the chat box and a moderator will try to assist you

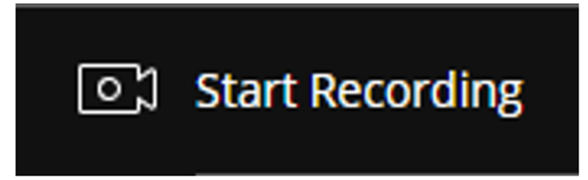


ddi.hsc.talent@ed.ac.uk

Supported by

Recording

This session will now be recorded. Any further information that you provide during a session is optional and in doing so you give us consent to process this information.



These sessions will be stored by the University of Edinburgh for one year and published for 30 days after the event. Schools or Services may use the recordings for up to a year on relevant websites.

By taking part in a session, you give us your consent to process any information you provide during it.

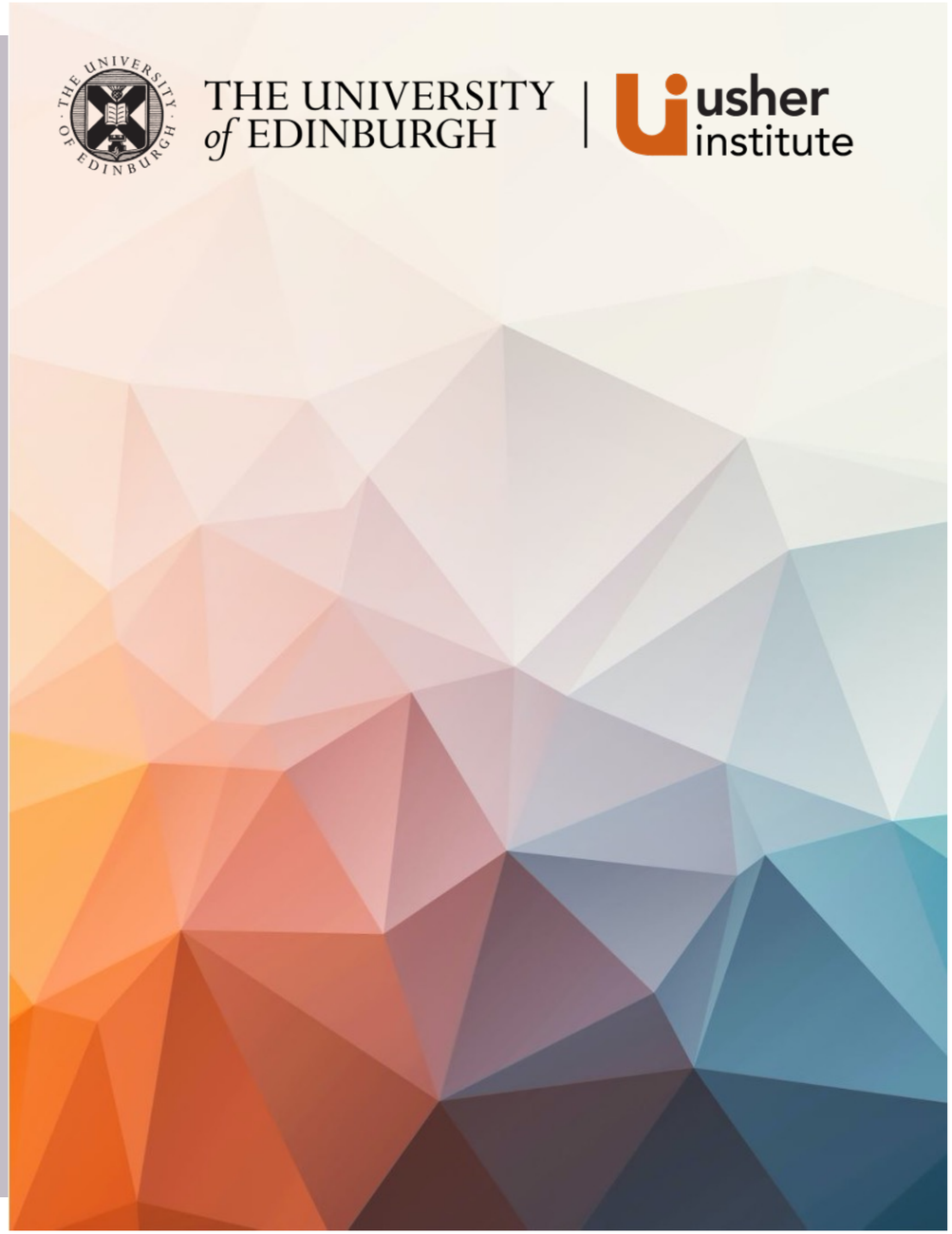
ddi.hsc.talent@ed.ac.uk

Supported by

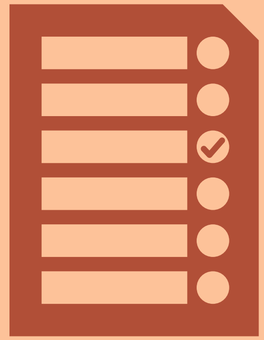
Introduction to data science in health and social care

Week 6

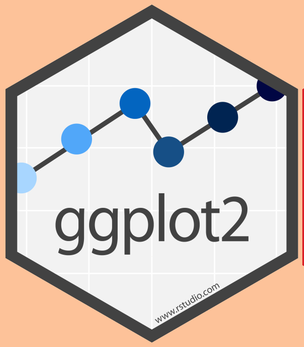
Brittany Blankinship | 2 November 2022

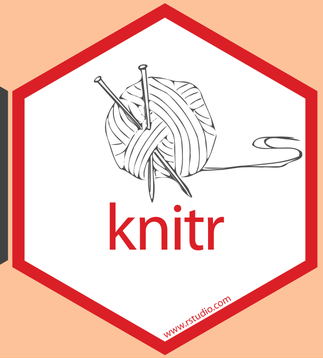


Agenda



- **Course/assessment updates**
- **tables vs figures**
- `ggplot2`
- **data storytelling with data visualisation**





2nd Graded Discussion Post

- Due: **11 November 12:00 UK time**

- Reply to a peer's response to receive full marks
- Can view other posts after submitting yours
  - Submit your response by creating a thread

Marking Criteria

This post is marked out of 10 total points and is worth 10% of your overall grade in the course. You will be assessed on your res

Criteria	Description	Marks
Completion	Submission of a discussion board post	2
Narrative	Describe (a) the data, (b) the health and social context of the data you have chosen, and (c) your intended audience.	3
Reflective practice	Report on why your team has chosen this specific data set, your contributions thus far, and experience thus far working as a team on this project (avoid using individual names, rather refer to team members generally).	3
Online participation	After adding your discussion post, reply to at least one of your peers' posts as well. **Note: this should be a peer in a different team than your own**. By replying to one of your classmates' posts you will receive full points, but you are welcome to respond to as many posts as you would like to.	2

- Example Datathon lead statement, elevator pitch, and group presentation now on Learn under assessments
- Example from last year’s group, using a different set of data around COVID-19

Datathon example group project

Do you think figures or tables are more useful for data visualisation?

Why do graphs and tables matter?

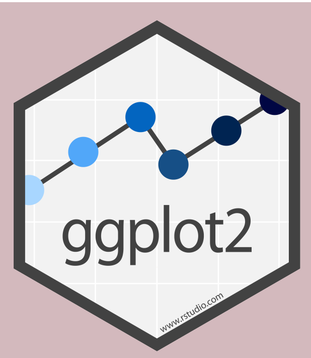
Both graphs AND tables are tools for communication

***Informative* and well-presented graphs & tables ARE better communication**

When to use tables vs graphs?

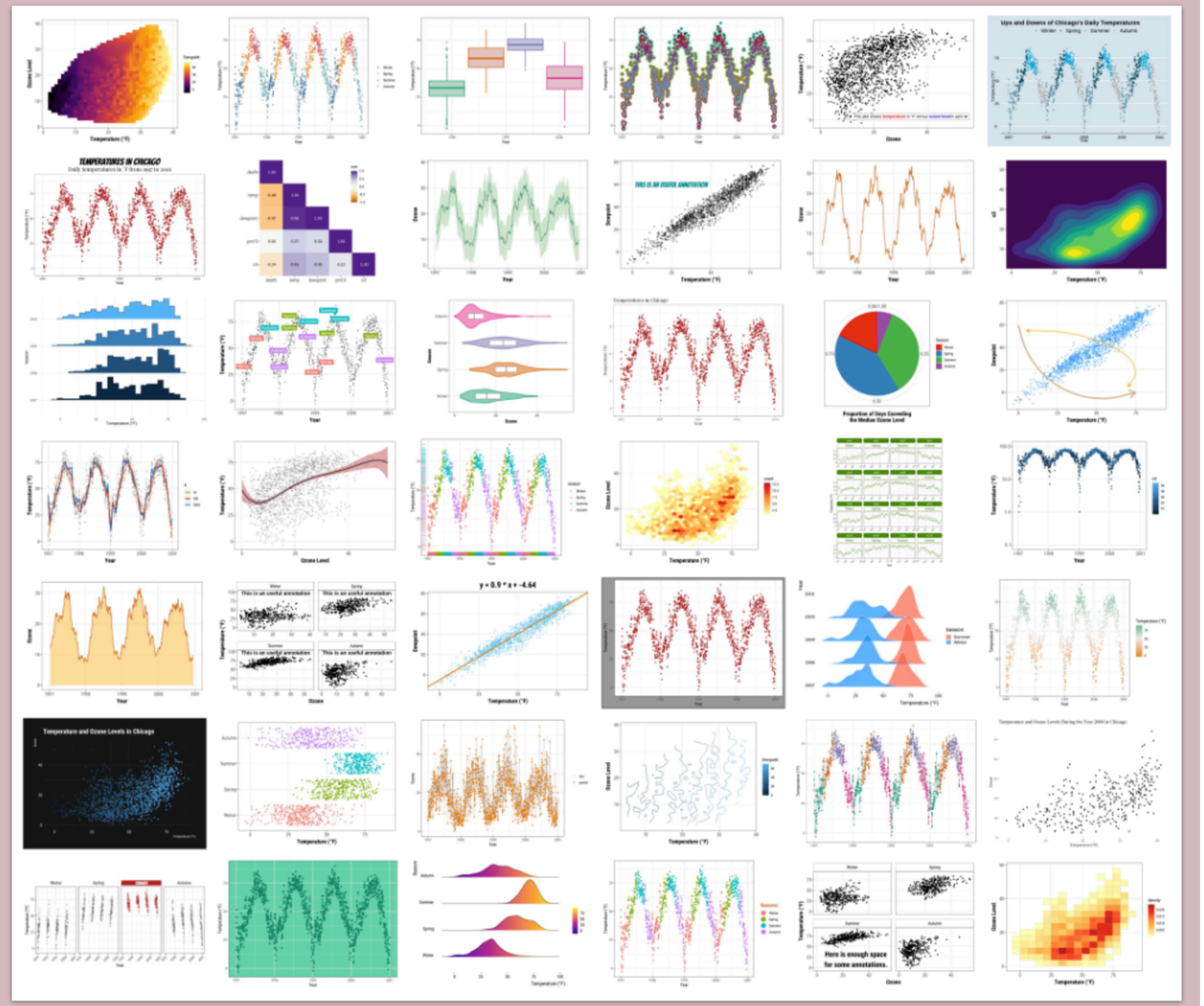
Use Tables When	Use Graphs When
• The display will be used to look up individual values	• The display will be used to reveal relationships among whole sets of values
• It will be used to compare individual values	• The message is contained in the shape of the values (e.g., patterns, trends, exceptions)
• Precise values are required	
• Quantitative values include more than one unit of measure	
• Both detail and summary values are included	

Adapted from:
Few, Stephen. (2012). *Show Me the Numbers: Designing Tables and Graphs to Enlighten*.(4)57



ggplot2

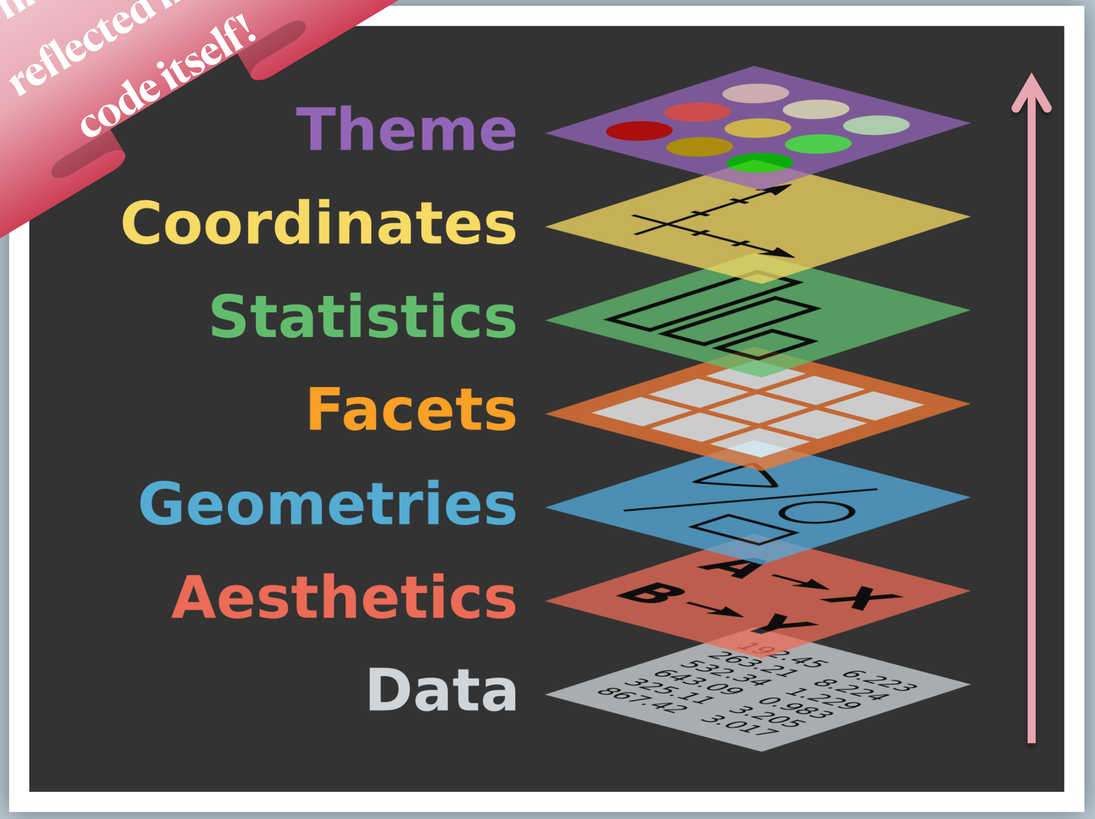
- Application of Leland Wilkinson's [grammar of graphics] for R
- A system for iteratively and declaratively creating graphics
- Just like grammar in a language constructs sentences, grammar of graphics constructs data visualisations



Credit: Cédric Sherer
[ggplot2 in-depth tutorial]

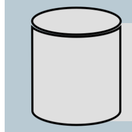
Building blocks of a `ggplot`

This layering is reflected in the code itself!



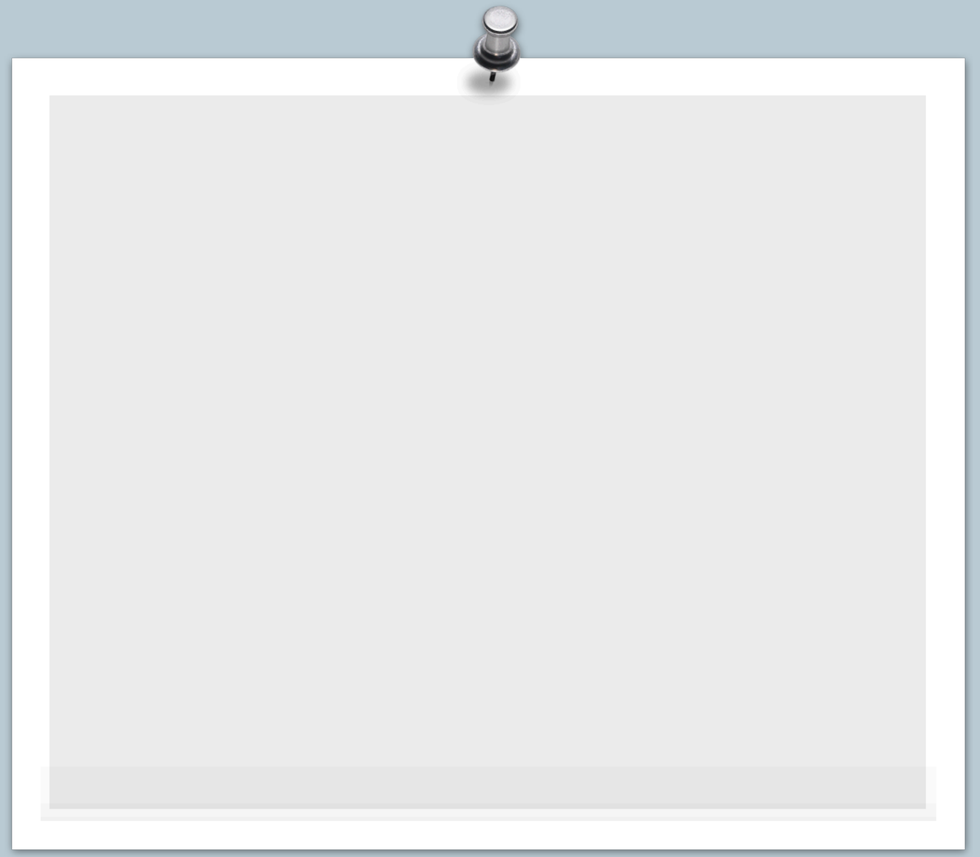
Element	Description	Code
Data	The dataset being plotted	`ggplot(data = ,`
Aesthetics	The scales onto which we map our data	`aes(x, y, fill, color))`
Geometries	The visual elements used for our data	`+ geom_()`
Facets	Plotting small multiples	`+ facet_()`
Statistics	Representations of our data to aid understanding	`+ stat_()`
Coordinates	The space on which the data will be plotted	`+ coord_()`
Themes	All non-data ink, design elements	`+ theme_()`

The key ingredients

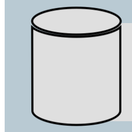


```
ggplot(data = cancelled ,
```

data layer use the data cancelled THEN



The key ingredients



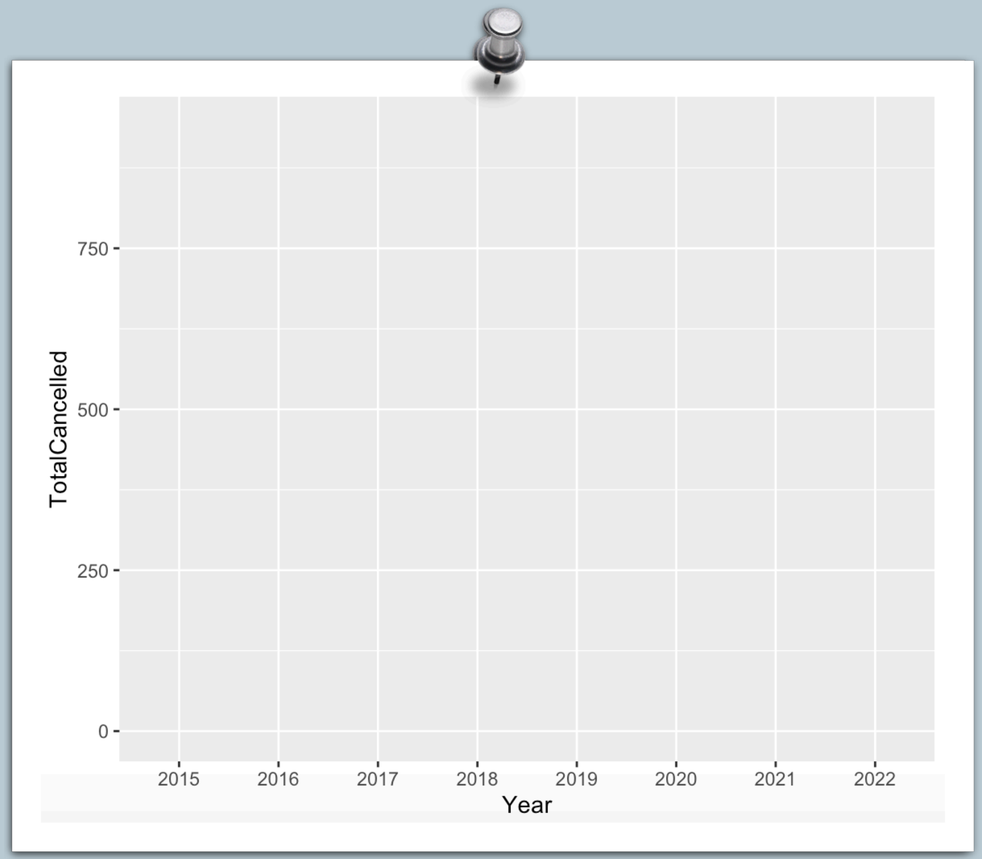
```
ggplot(data = cancelled ,
```

data layer — use the data cancelled — THEN

```
aes(x = Year, y = TotalCancelled))   +
```

use the X axis to show “Year” then the Y axis to show ”Total Cancelled” — AND

aesthetics layer



The key ingredients

```
ggplot(data = cancelled ,
```

data layer — use the data cancelled — THEN

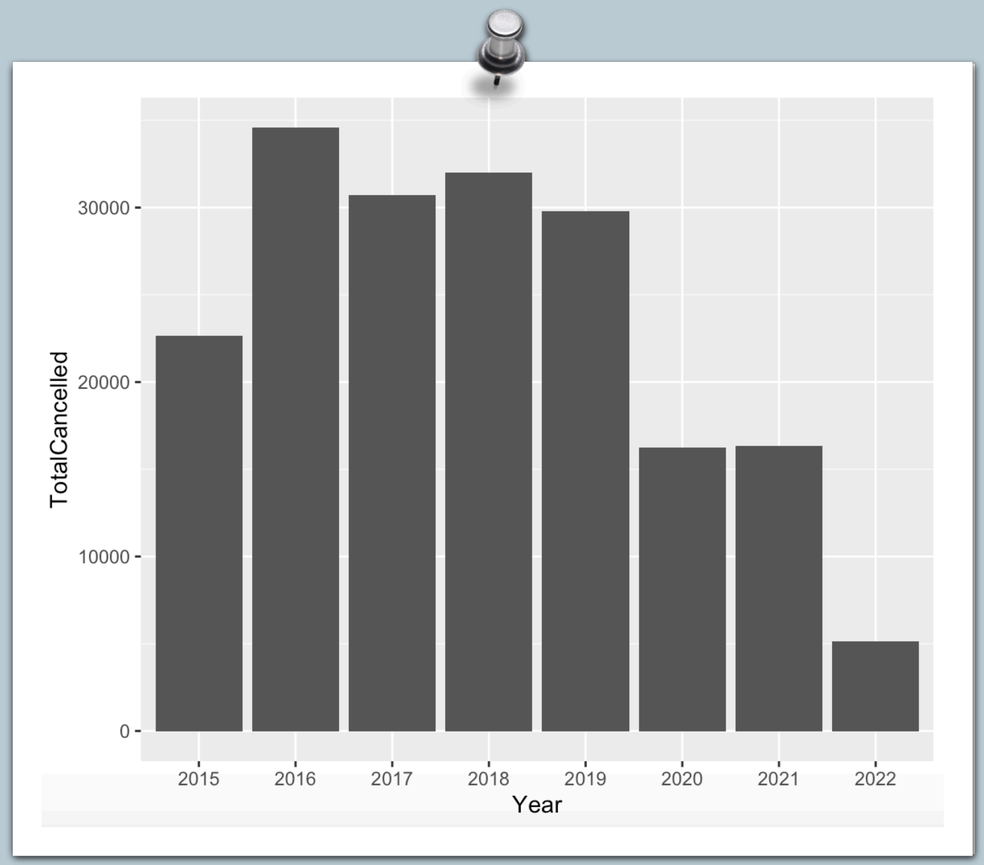
```
aes(x = Year, y = TotalCancelled))  +
```

use the X axis to show “Year” then the Y axis to show ”Total Canceled” — AND

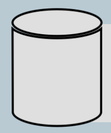
aesthetics layer

```
geom_bar()
```

geometry layer — …. take all that and make a bar chart

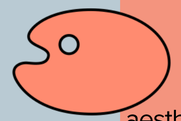


The key ingredients



```
ggplot(data = cancelled ,
```

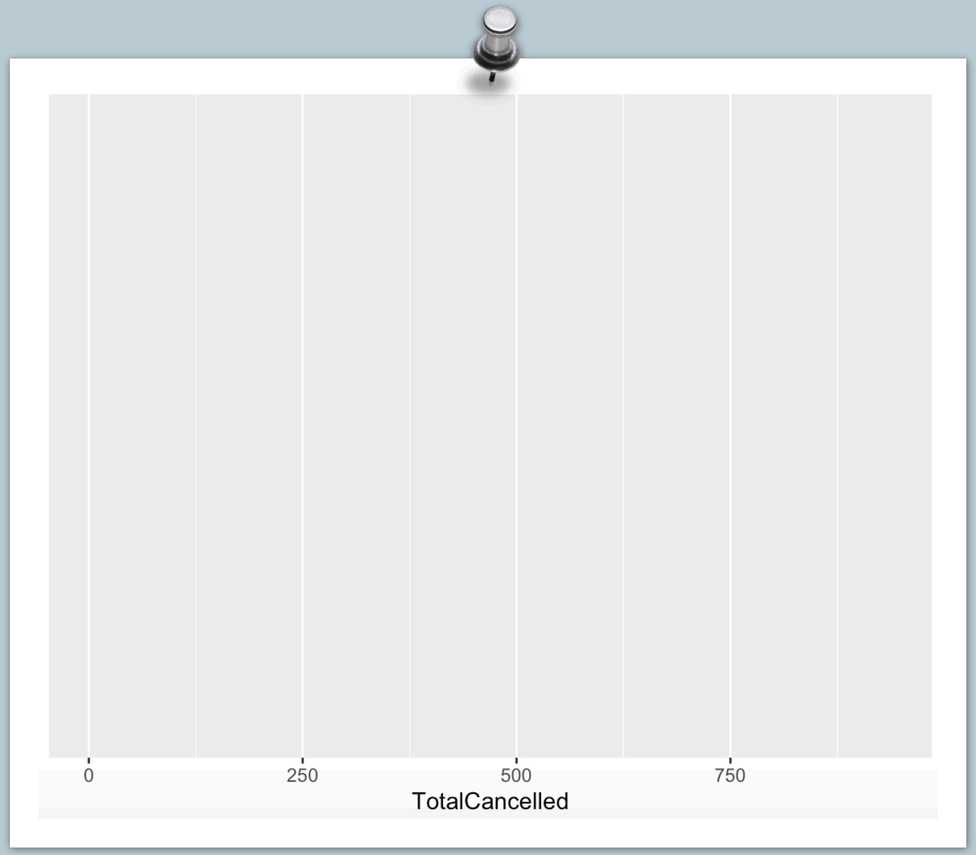
data layer — use the data cancelled — THEN



```
aes(x = TotalCancelled))   +
```

use the X axis to show ”Total Cancelled” — AND

aesthetics layer



The key ingredients

```
ggplot(data = cancelled ,
```

data layer — use the data cancelled — THEN

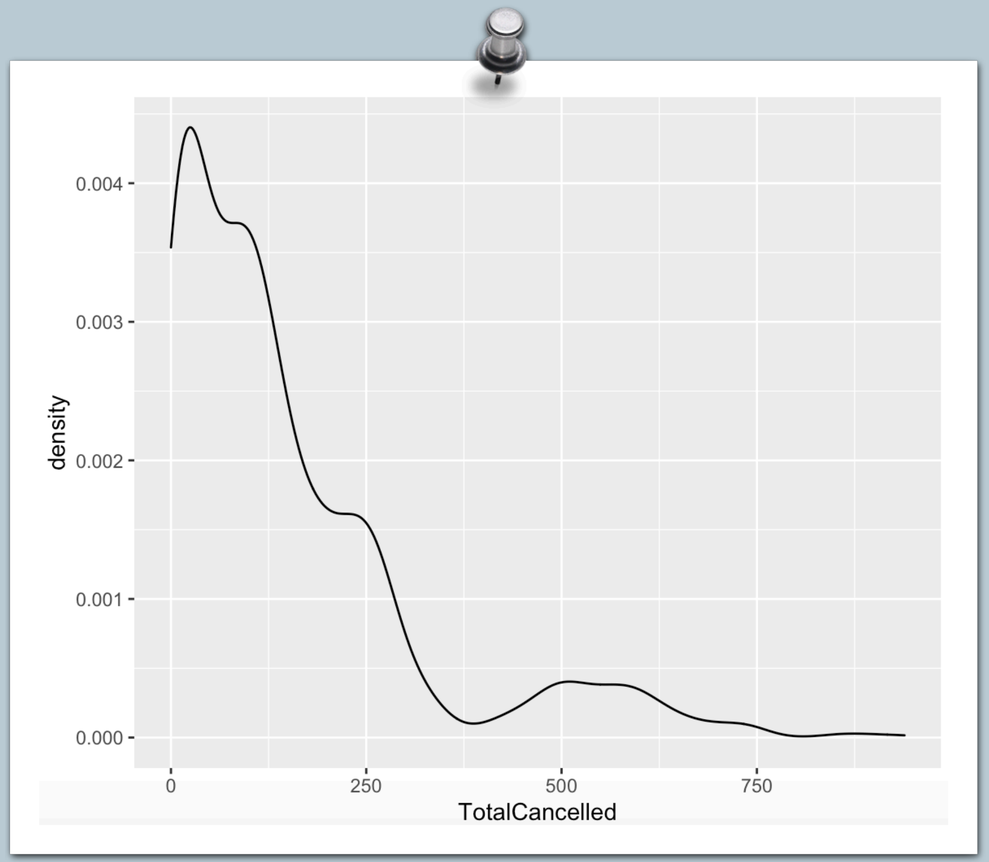
```
aes(x = TotalCancelled))   +
```

use the X axis to show "Total Cancelled" — AND

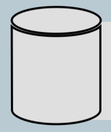
aesthetics layer

```
geom_density()
```

geometry layer .... take all that and make a density chart

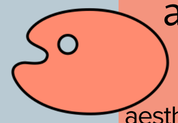


The key ingredients



```
ggplot(data = cancelled ,
```

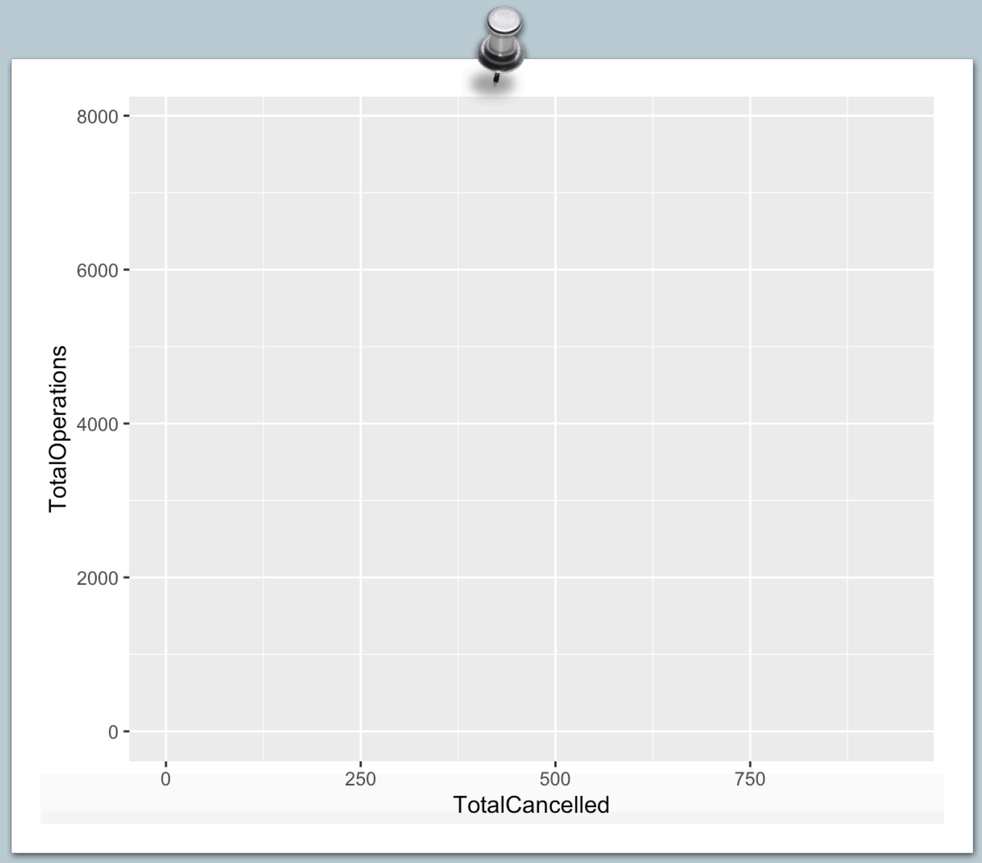
data layer use the data cancelled THEN



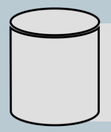
```
aes(x = TotalCancelled, y = TotalOperations))  +
```

use the X axis to show “Total Cancelled” then the Y axis to show ”Total Operations” AND

aesthetics layer

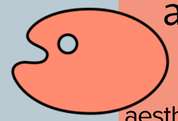


The key ingredients



```
ggplot(data = cancelled ,
```

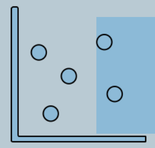
data layer use the data cancelled THEN



```
aes(x = TotalCancelled, y = TotalOperations))  +
```

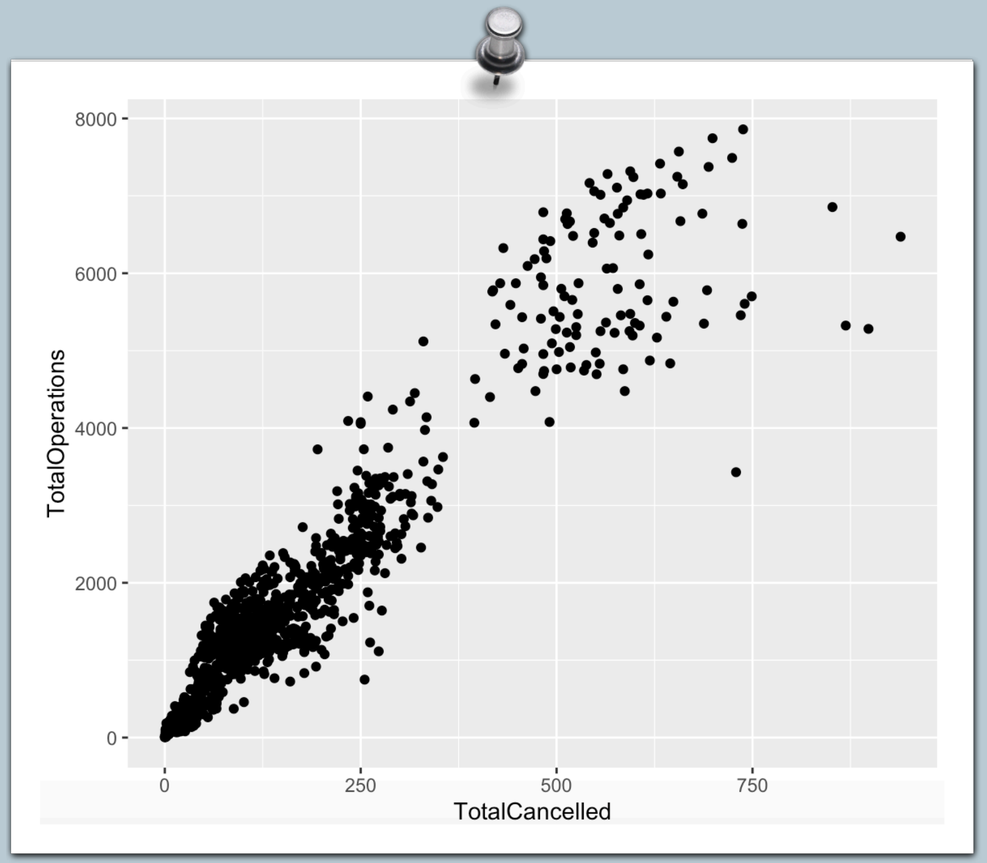
use the X axis to show “Total Cancelled” then the Y axis to show ”Total Operations” AND

aesthetics layer



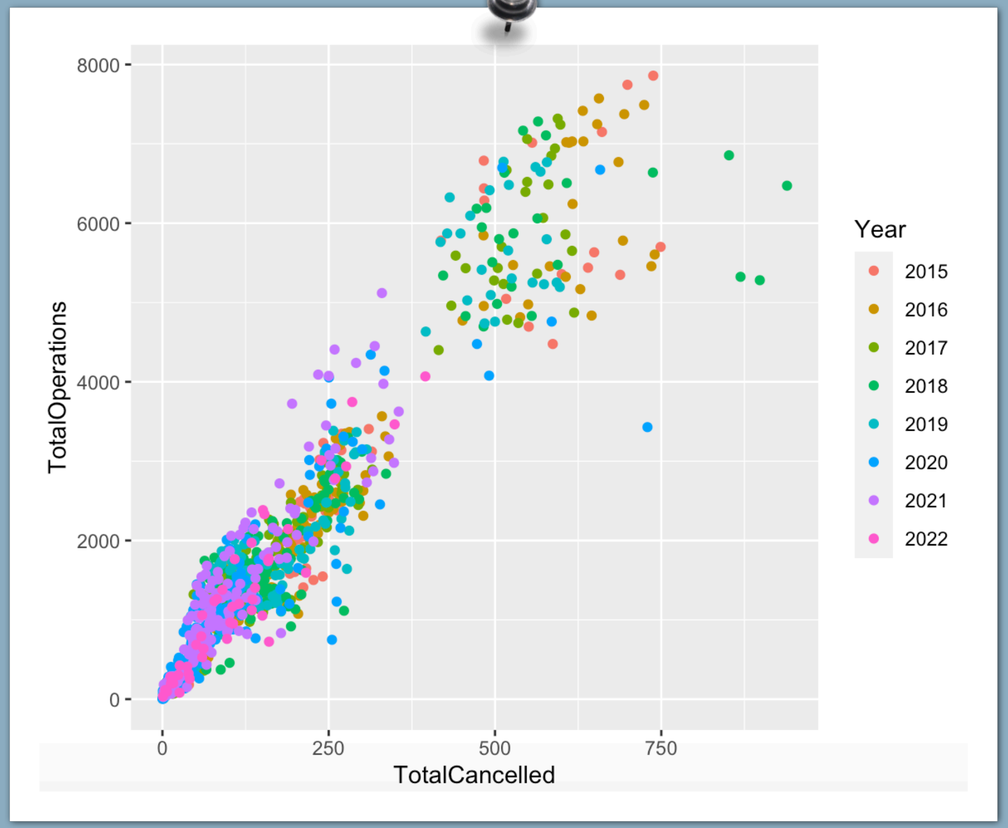
```
geom_point()
```

geometry layer .... take all that and make a scatter plot

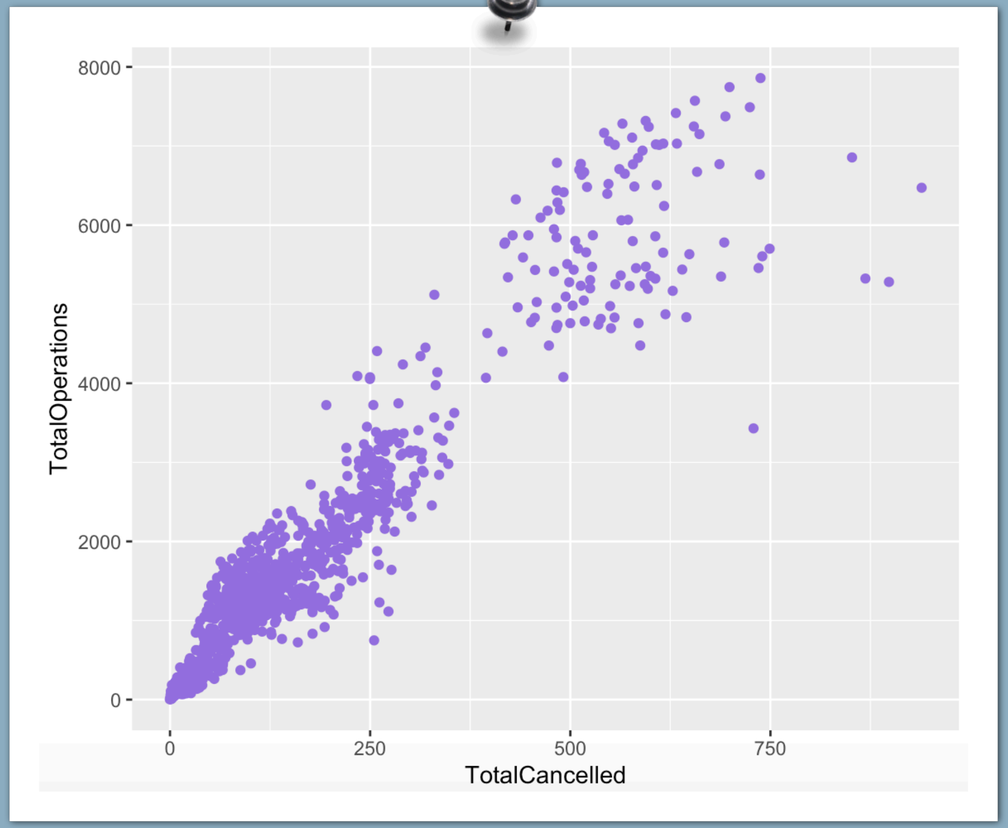


Aesthetics: Inside, Outside, or both

- **Within** `aes()`: the argument changes based on the values of the variable

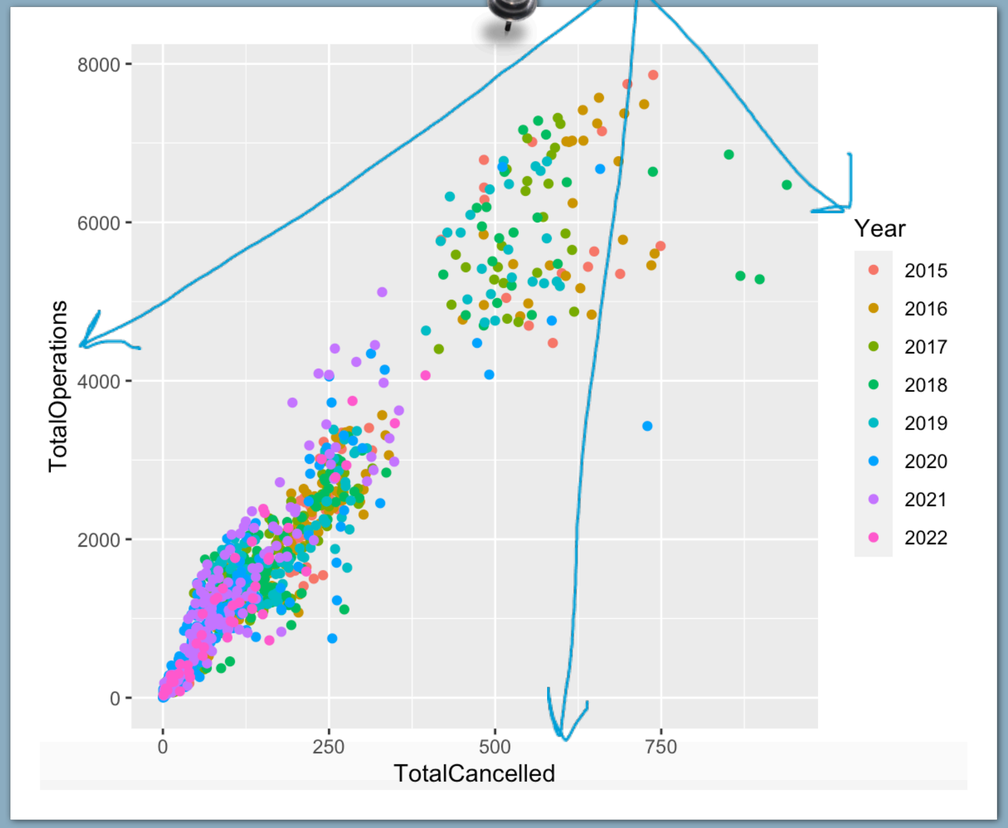


- **Outside** `aes()`: the argument is given a single value and *doesn't* change based on the values of the variable

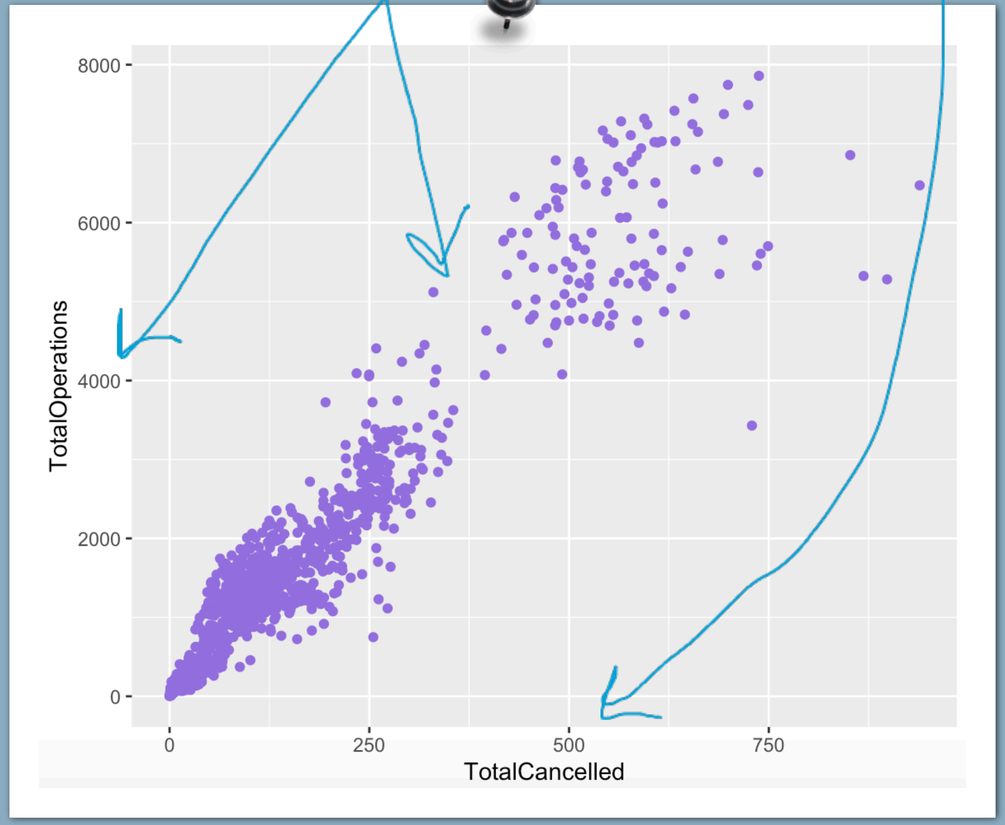


Aesthetics: Inside vs Outside

```
ggplot(cancelled, aes(x = TotalCancelled, y = TotalOperations,
                      color = Year)) +
  geom_point()
```



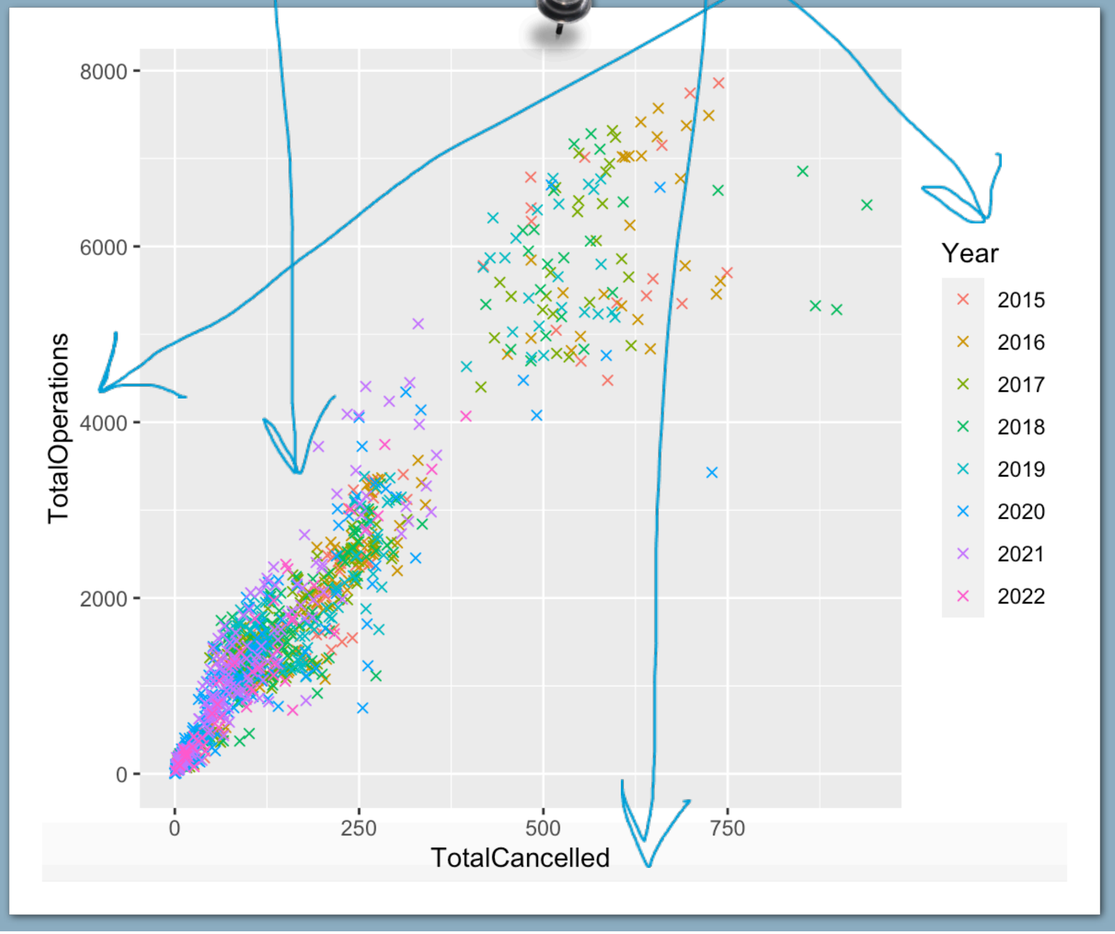

```
ggplot(cancelled, aes(x = TotalCancelled,
                      y = TotalOperations) +
  geom_point(color = "mediumpurple")
```



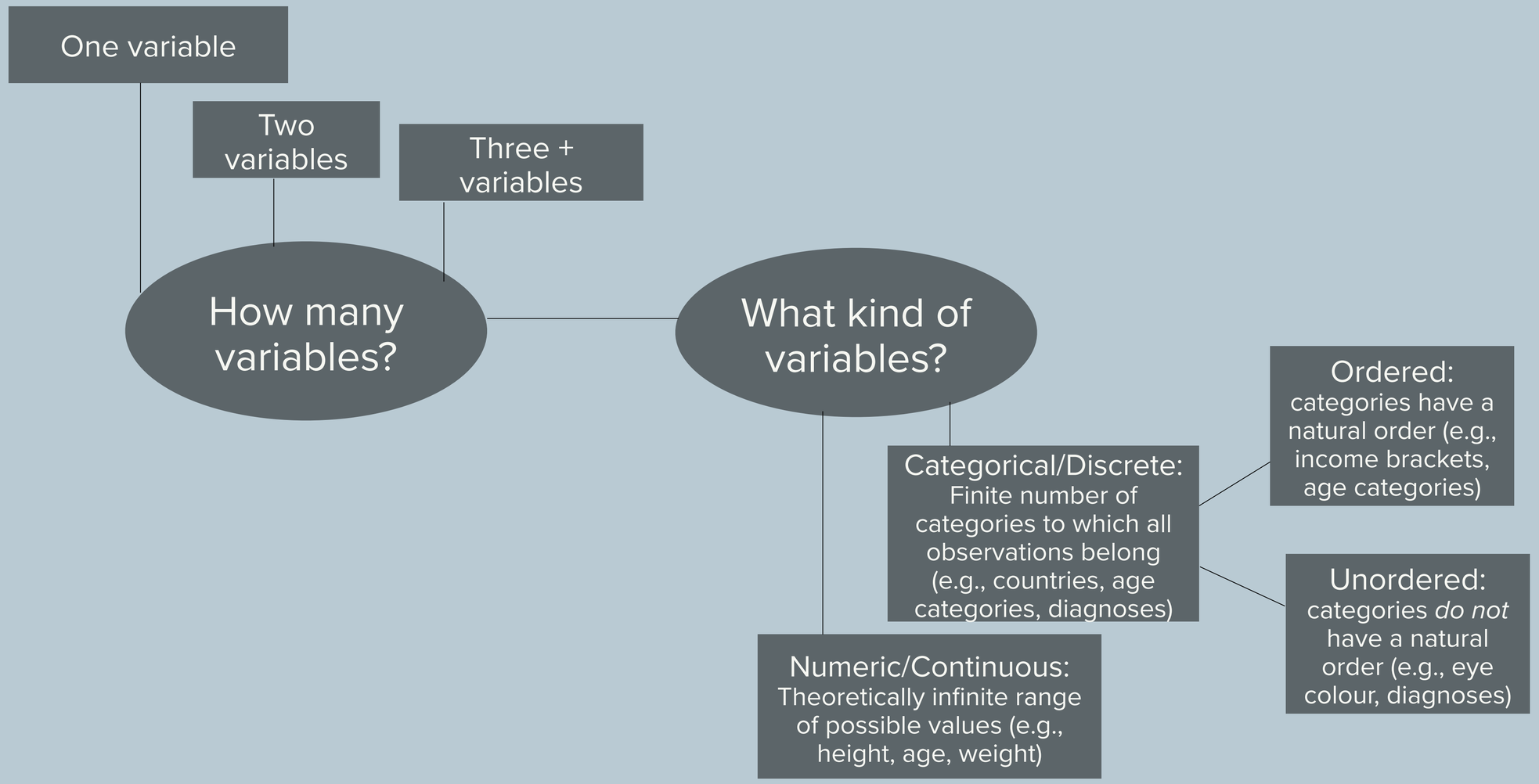
Aesthetics: Inside, Outside, or both

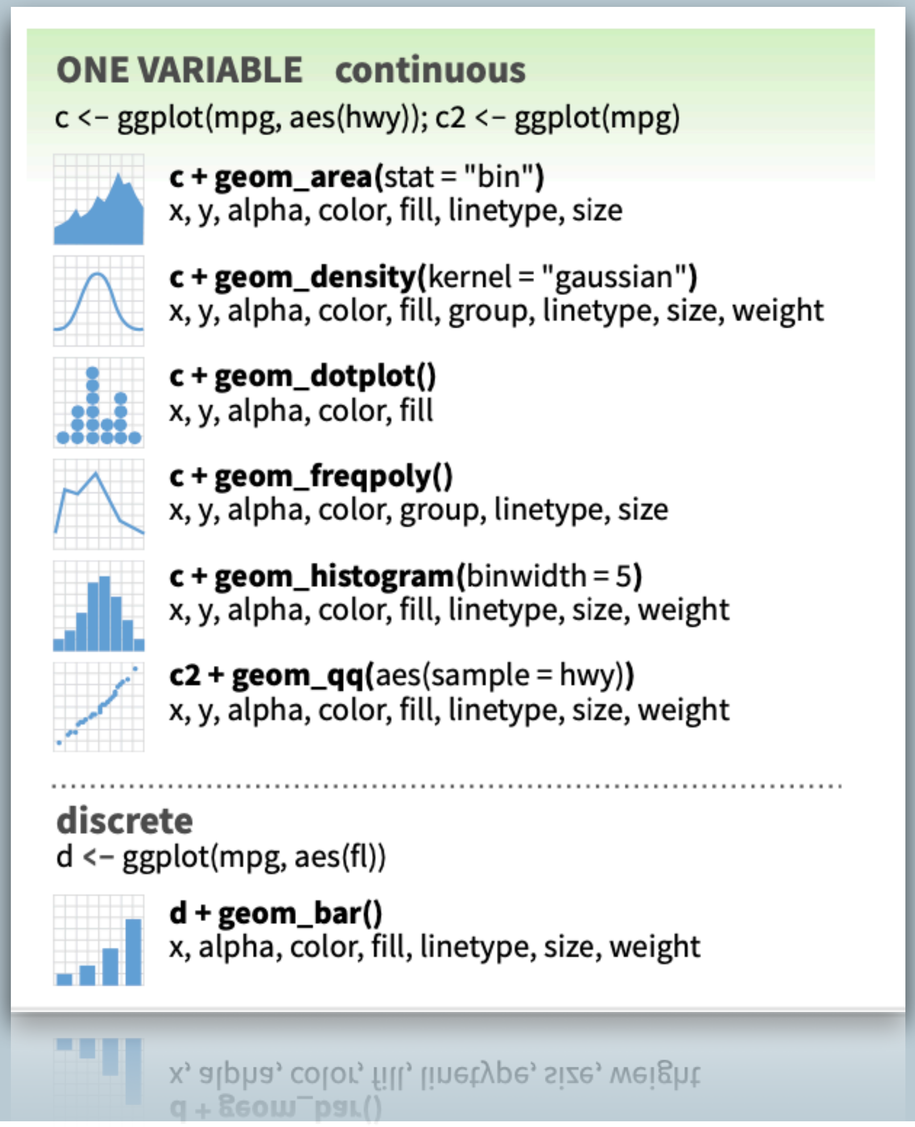
- To learn more about the different aesthetic specifications
- For a list of colors in R

```
ggplot(cancelled, aes(x = TotalCancelled, y = TotalOperations,
                      color = Year)) +
  geom_point(shape = 4)
```



Geoms: decisions, decisions





One variable geoms

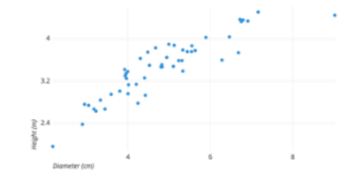
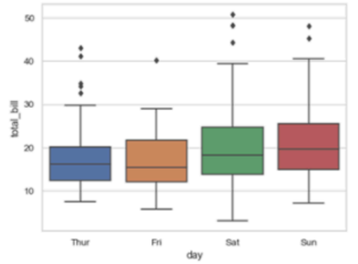
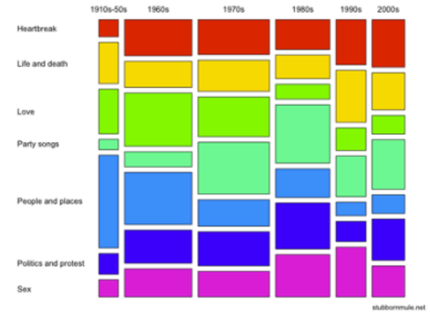
Continuous/numeric X

- Histogram
- Density plot
- Dot plot
- Etc.

Discrete/Categorical X

- Bar charts

When visualising 2 variables

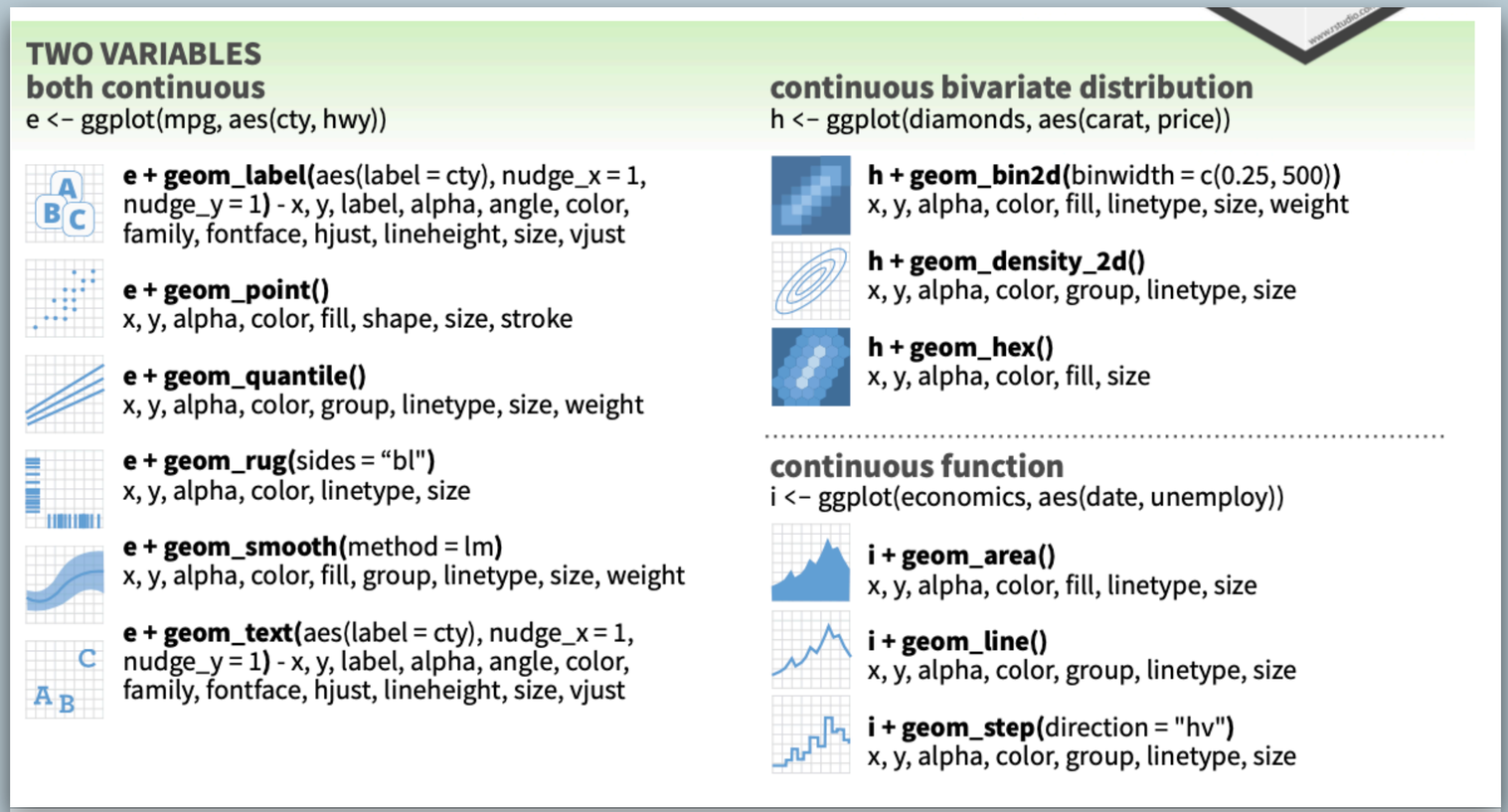
	Numerical	Categorical
Numerical	 • Scatter plot (point) • 2D binning (bin2d, hex) • Contour plot (density2d) • Quantiles (quantile, qq) • Lines (line, smooth) • Ribbons (ribbon, area)	 • Boxplot (boxplot, violin) • Counts (count, tile) • Error bars (errorbar) • Columns (col)
Categorical	***Which ggplot2 data viz is right for your data?*** (geoms in parentheses)	 • Mosaic (ggmosaic::geom_mosaic) • Counts (count, tile)

From Dr Sam Tyner's guest lecture optional video

2 variable geoms: numeric

Continuous/numeric X & Y

- Line chart
- Scatter plot
- Etc.



TWO VARIABLES
both continuous
e <- ggplot(mpg, aes(cty, hwy))

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1**)** - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_point()
x, y, alpha, color, fill, shape, size, stroke

e + geom_quantile()
x, y, alpha, color, group, linetype, size, weight

e + geom_rug(sides = "bl"**)**
x, y, alpha, color, linetype, size

e + geom_smooth(method = lm**)**
x, y, alpha, color, fill, group, linetype, size, weight

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1**)** - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

continuous bivariate distribution
h <- ggplot(diamonds, aes(carat, price))

h + geom_bin2d(binwidth = c(0.25, 500)**)**
x, y, alpha, color, fill, linetype, size, weight

h + geom_density_2d()
x, y, alpha, color, group, linetype, size

h + geom_hex()
x, y, alpha, color, fill, size

continuous function
i <- ggplot(economics, aes(date, unemploy))

i + geom_area()
x, y, alpha, color, fill, linetype, size

i + geom_line()
x, y, alpha, color, group, linetype, size

i + geom_step(direction = "hv"**)**
x, y, alpha, color, group, linetype, size

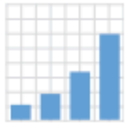
2 variable geoms: numeric & categorical

Continuous/numeric X & Discrete/Categorical Y
- Bar chart

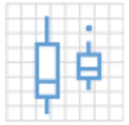
Discrete/Categorical X & Continuous/numeric Y
- Box plot
- Dot plot
- Violin chart

one discrete, one continuous

f <- ggplot(mpg, aes(class, hwy))



f + geom_col()
x, y, alpha, color, fill, group, linetype, size



f + geom_boxplot()
x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

f + geom_dotplot(binaxis = "y", stackdir = “center"**)**
x, y, alpha, color, fill, group

f + geom_violin(scale = “area"**)**
x, y, alpha, color, fill, group, linetype, size, weight

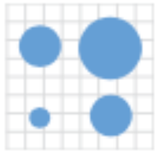
2 variable geoms: categorical

Discrete/categorical X & Y

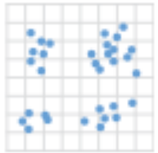
- Mosaic charts
- Counts

both discrete

g <- ggplot(diamonds, aes(cut, color))



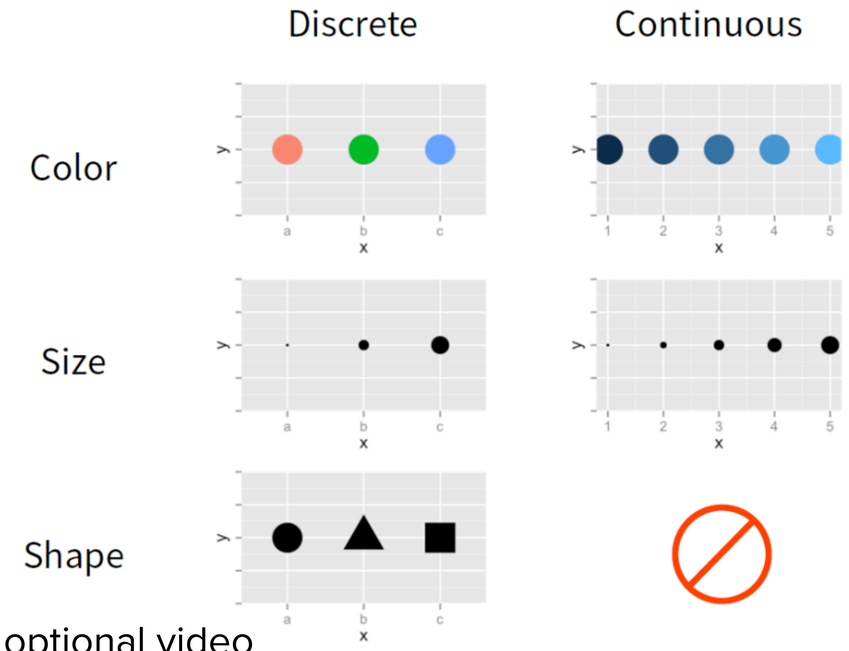
g + geom_count()
x, y, alpha, color, fill, shape, size, stroke



e + geom_jitter(height = 2, width = 2**)**
x, y, alpha, color, fill, shape, size

When visualising 3 variables

Start with a 2-variable visualization, then add color, shape, or size:

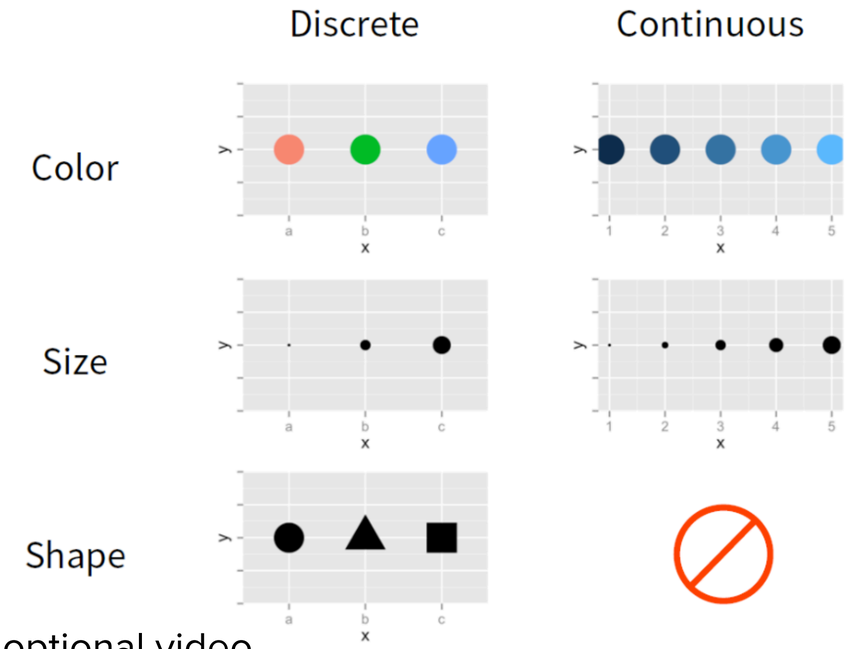


From Dr Sam Tyner's guest lecture optional video

When visualising 3+ variables

Start with a 2-variable visualization, then add color, shape, or size:

Add another variable with Facets



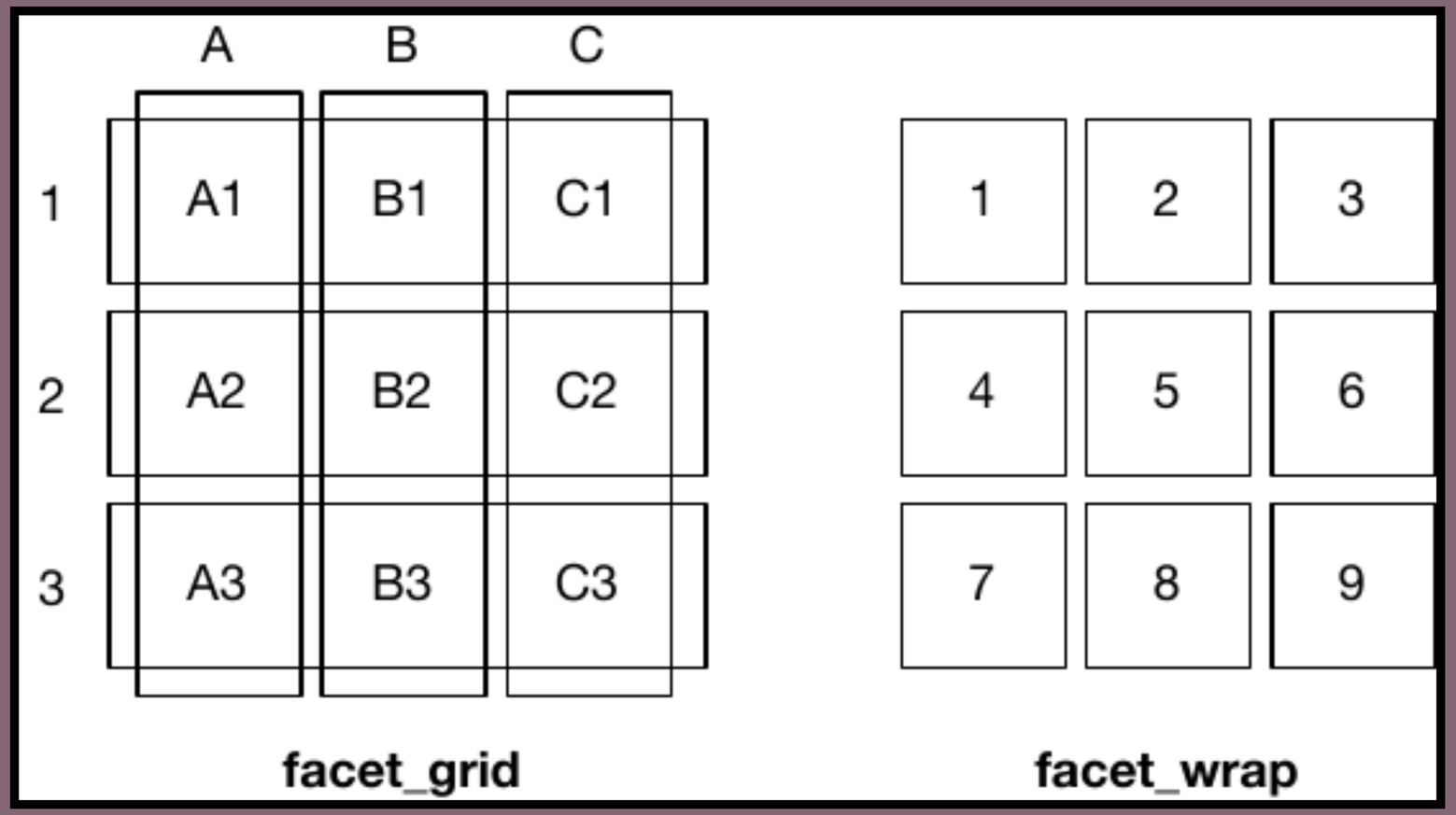
From Dr Sam Tyner's guest lecture optional video

Visualising 3+ variables: Facets

Fundamentally 2 dimensional

`+facet_grid(row ~ column)`

Makes a matrix of panels by row & column faceting variables



Fundamentally 1 dimensional

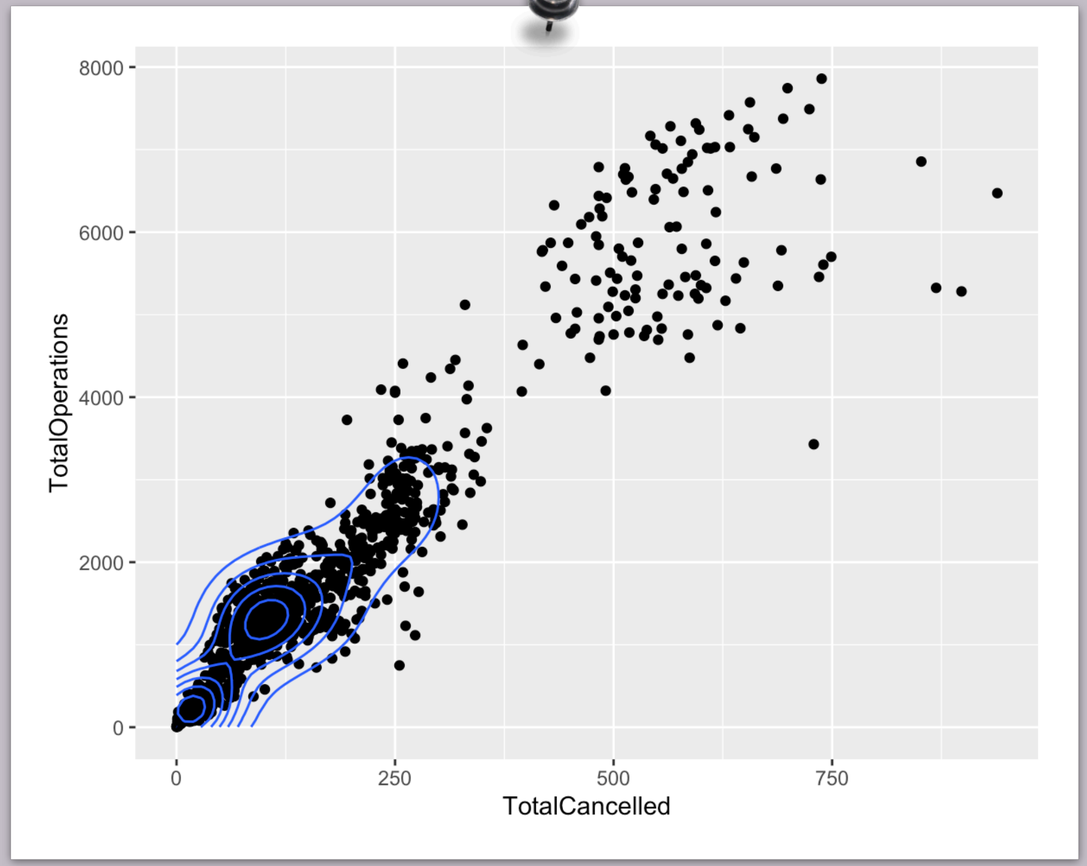
`+facet_wrap(variable)`

Makes a long ribbon of panels

ggplot2 layers

- Stacked in order of code appearance
- Important to keep in mind as elements written later in your code may hide or overwrite previous elements

```
ggplot(cancelled, aes(TotalCancelled, TotalOperations)) +
  geom_point() +
  geom_density2d()
```

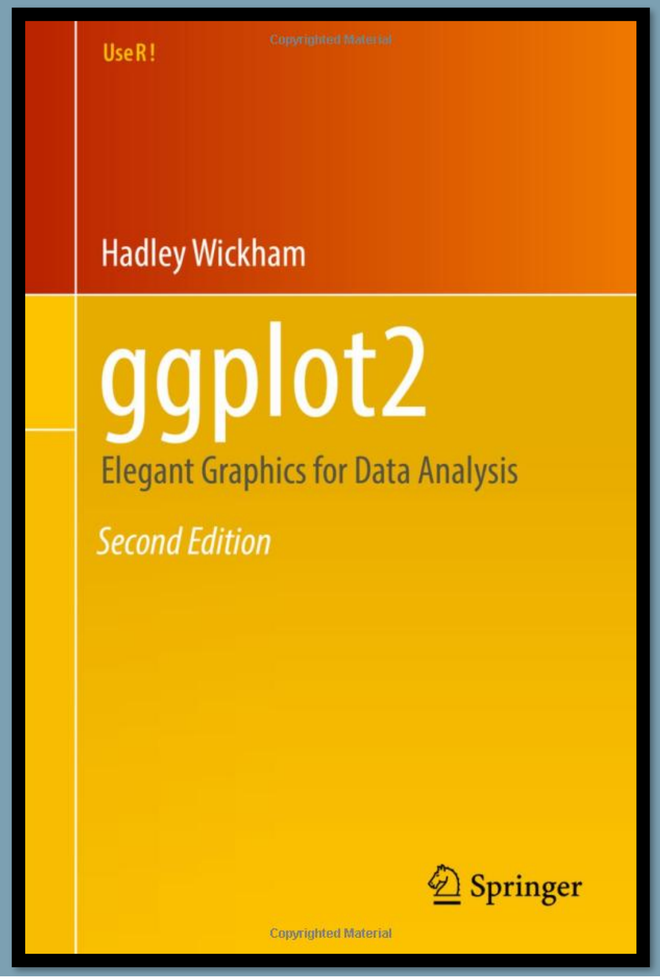


To RStudio!

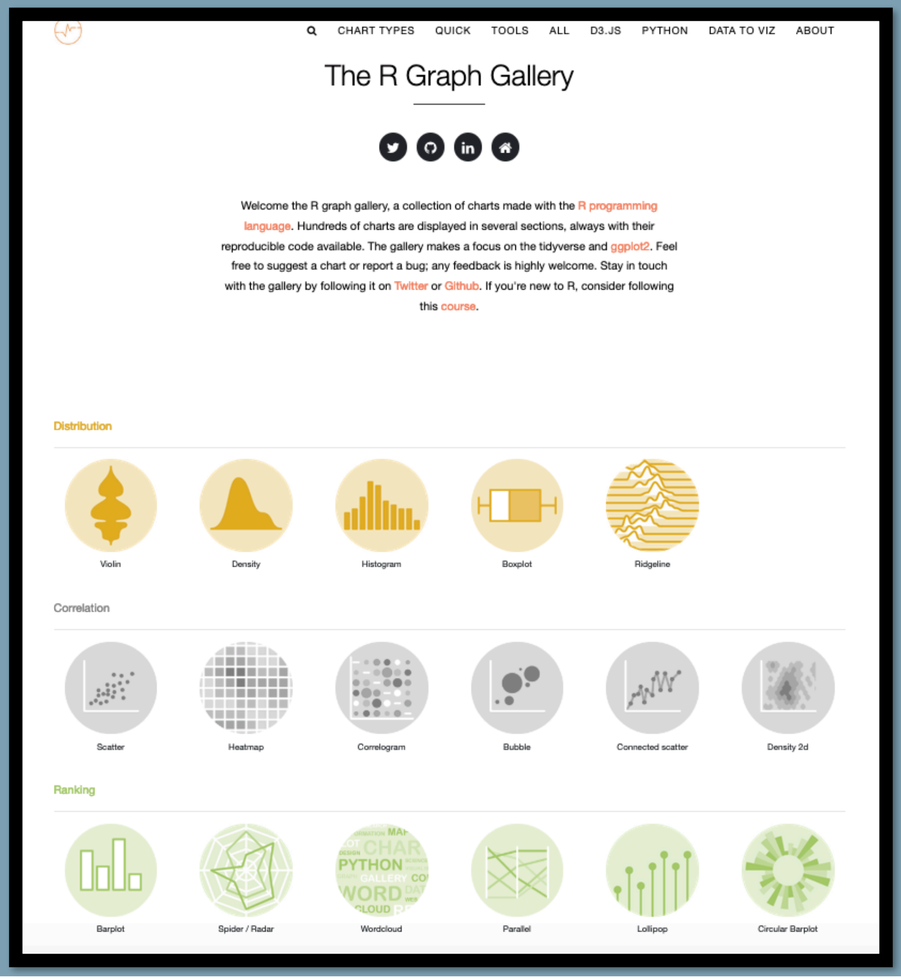
data storytelling with data visualisation

Further Resources

(free!) ggplot2 book



R Graph Gallery



Questions?