

Weekly Oxford Worldwide

DEPARTMENT FOR
CONTINUING
EDUCATION



Infectious Disease Modelling: Applied Methods in R

Dr Emma L Davis

Week 3



Week 3: Plotting and visualisation in R

- Methods so far: incidence, EpiEstim
 - Example: Ebola
- Plots as objects
- Using ggplot2
- Hints and tips for beautiful plots

Methods so far

Built into packages we've used:

- incidence
 - > plot(incidence_object)
 - > plot(fit_object)
- EpiEstim
 - > plot(R_object)

Methods so far

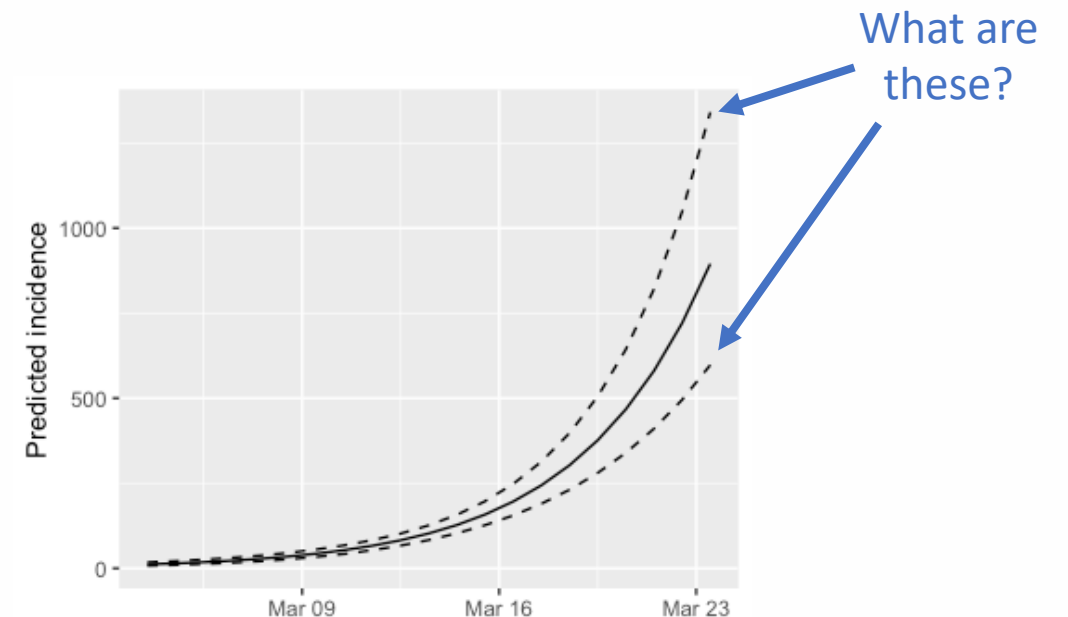
What is the problem with these figures?

- Not very attractive to look at
- Overlapping axis labels
- Non-descriptive or over-complex legend labels
- Missing / unclear labels



groups

newCasesBySpecimenDate



Personalising figures in the incidence and EpiEstim packages

```
> plot(incidence_object, ...)
```



We can put additional
“arguments” in here

- A function argument is basically just an input
- Most functions will have “defaults” – we don’t need to mention them when we run the function
- We can specify if we want to use an option other than the default

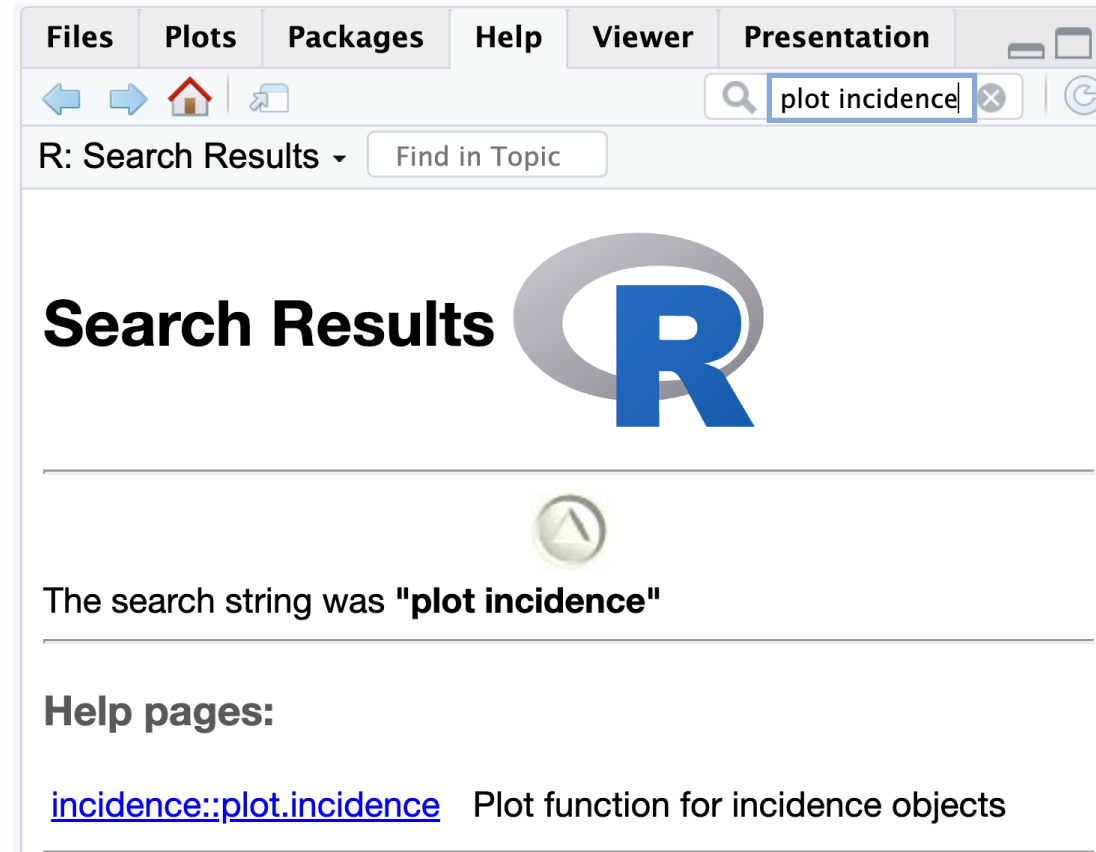
Personalising figures in the incidence and EpiEstim packages

```
> plot(incidence_object, ...)
```

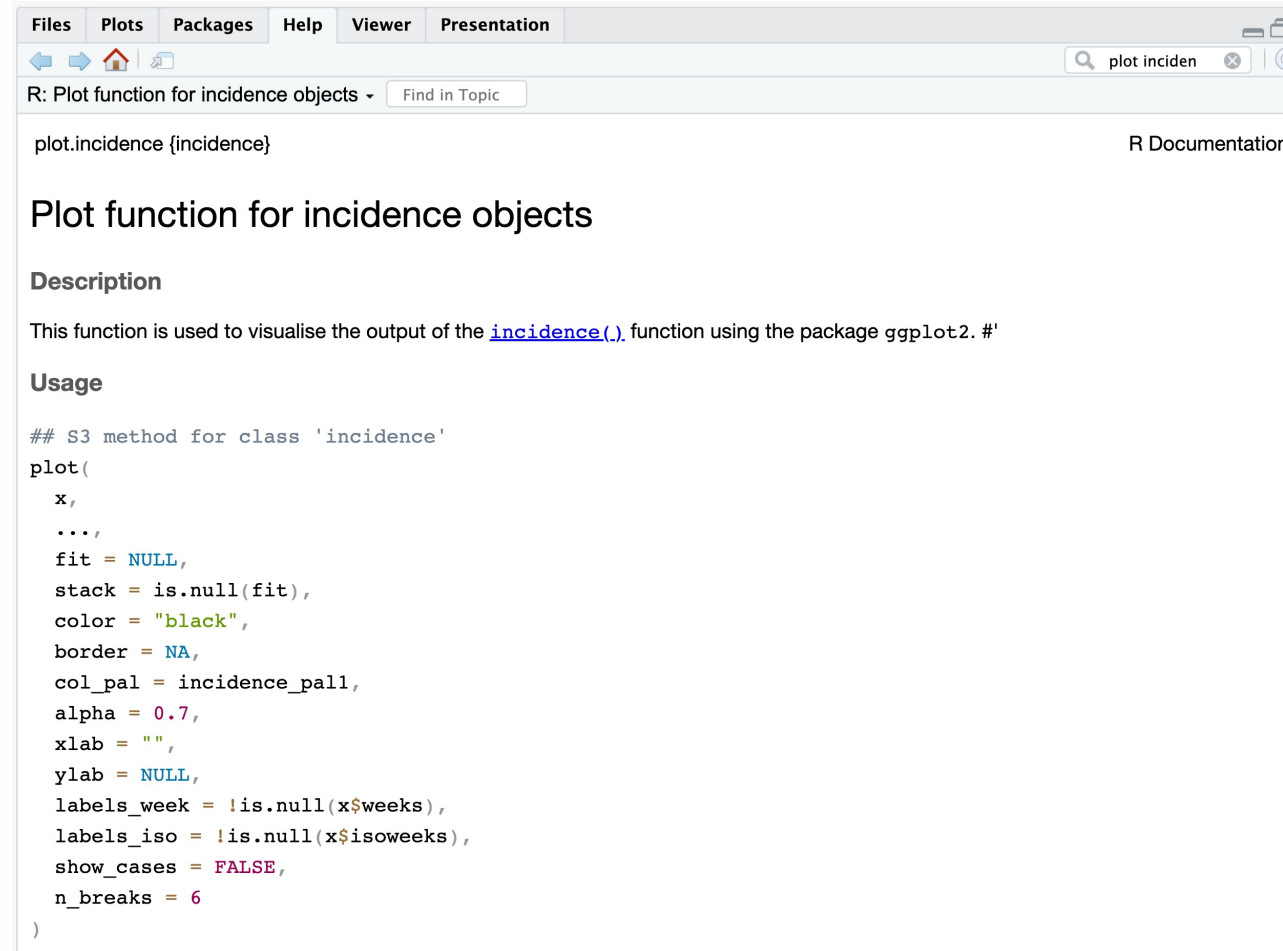
How do we know what arguments a function accepts
and what format we should put them in?

Idea: Try using the Help tab to get more information on our function

Personalising figures in the incidence and EpiEstim packages



Personalising figures in the incidence and EpiEstim packages



The screenshot shows the R Documentation page for the `plot.incidence` function. The browser window has tabs for Files, Plots, Packages, Help, Viewer, and Presentation. The address bar shows the search term 'plot inciden'. The page title is 'R: Plot function for incidence objects'. The main content area displays the function signature `plot.incidence {incidence}` and the title 'Plot function for incidence objects'. Under the 'Description' section, it states: 'This function is used to visualise the output of the [incidence\(.\)](#) function using the package ggplot2. #''. The 'Usage' section shows the following code:

```
## S3 method for class 'incidence'
plot(
  x,
  ...,
  fit = NULL,
  stack = is.null(fit),
  color = "black",
  border = NA,
  col_pal = incidence_pal1,
  alpha = 0.7,
  xlab = "",
  ylab = NULL,
  labels_week = !is.null(x$weeks),
  labels_iso = !is.null(x$isoweeks),
  show_cases = FALSE,
  n_breaks = 6
)
```


Personalising figures in the incidence and EpiEstim packages

```
plot(  
  x,  
  ...,  
  fit = NULL,  
  stack = is.null(fit),  
  color = "black",  
  border = NA,  
  col_pal = incidence_pal1,  
  alpha = 0.7,  
  xlab = "",  
  ylab = NULL,  
  labels_week = !is.null(x$weeks),  
  labels_iso = !is.null(x$isoweeks),  
  show_cases = FALSE,  
  n_breaks = 6  
)
```

Personalising figures in the incidence and EpiEstim packages

Arguments

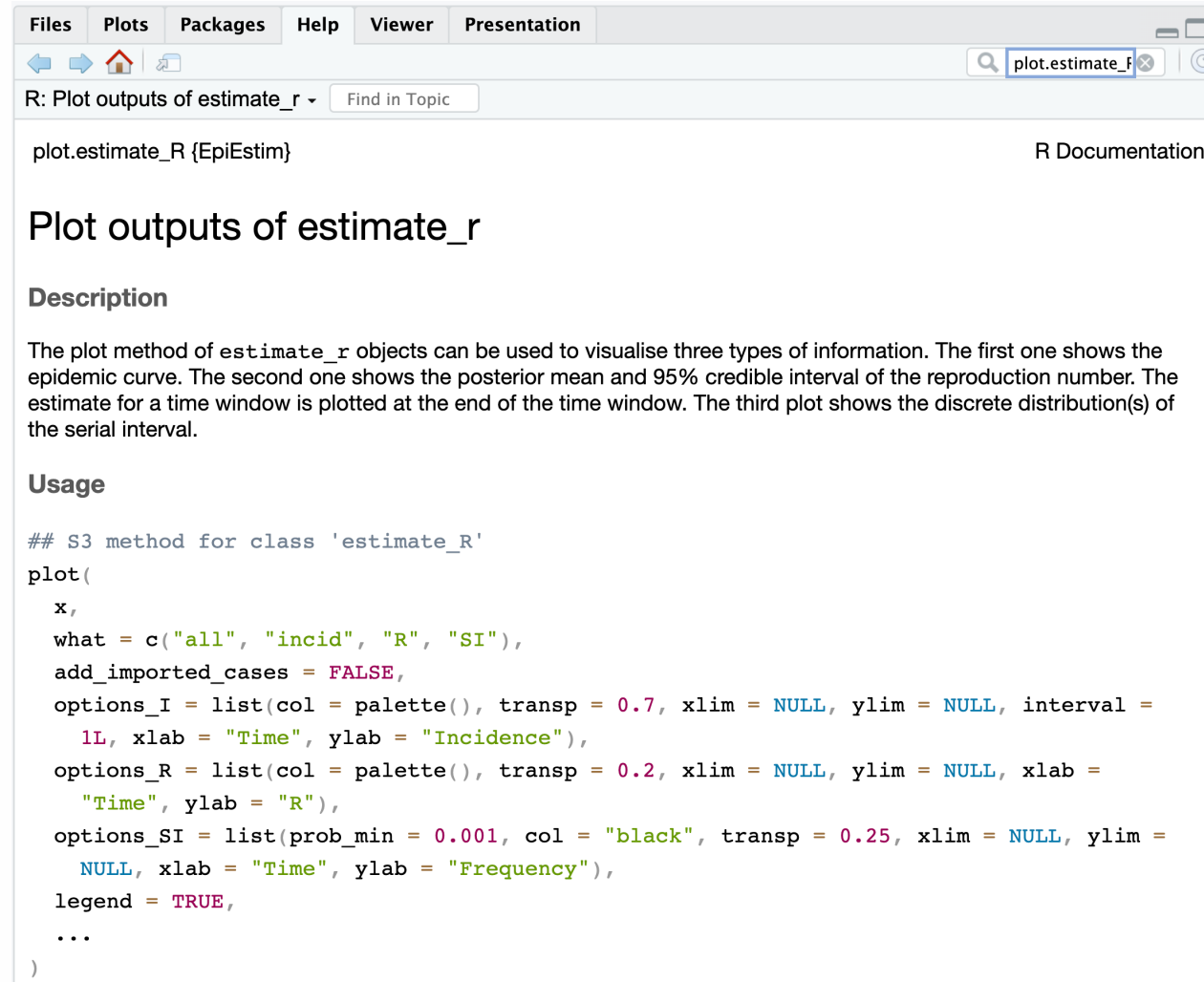
<code>x</code>	An incidence object, generated by the function incidence(.) .
<code>...</code>	arguments passed to ggplot2::scale_x_date(.) , ggplot2::scale_x_datetime(.) , or ggplot2::scale_x_continuous(.) , depending on how the <code>\$date</code> element is stored in the incidence object.
<code>fit</code>	An 'incidence_fit' object as returned by fit(.) .
<code>stack</code>	A logical indicating if bars of multiple groups should be stacked, or displayed side-by-side.
<code>color</code>	The color to be used for the filling of the bars; NA for invisible bars; defaults to "black".
<code>border</code>	The color to be used for the borders of the bars; NA for invisible borders; defaults to NA.
<code>col_pal</code>	The color palette to be used for the groups; defaults to <code>incidence_pal1</code> . See incidence_pal1(.) for other palettes implemented in incidence.
<code>alpha</code>	The alpha level for color transparency, with 1 being fully opaque and 0 fully transparent; defaults to 0.7.
<code>xlab</code>	The label to be used for the x-axis; empty by default.
<code>ylab</code>	The label to be used for the y-axis; by default, a label will be generated automatically according to the time interval used in incidence computation.
<code>labels_week</code>	a logical value indicating whether labels x axis tick marks are in week format YYYY-Www when plotting weekly incidence; defaults to TRUE.
<code>labels_iso</code>	(deprecated) This has been superceded by <code>labels_iso</code> . Previously: a logical value indicating whether labels x axis tick marks are in ISO 8601 week format yyyy-Www when plotting ISO week-based weekly incidence; defaults to be TRUE.
<code>show_cases</code>	if TRUE (default: FALSE), then each observation will be colored by a border. The border defaults to a white border unless specified otherwise. This is normally used outbreaks with a small number of cases. Note: this can only be used if <code>stack = TRUE</code>
<code>n_breaks</code>	the ideal number of breaks to be used for the x-axis labeling
<code>p</code>	An existing incidence plot.

Personalising figures in the incidence and EpiEstim packages

incidence example in R – Ebola (using data from package: outbreaks)

- Step 1: Install package outbreaks
- Step 2: Create a new project and R markdown file
- Step 3: Load package outbreaks and Ebola data

Personalising figures in the incidence and EpiEstim packages



The screenshot shows the RStudio interface with the 'Plots' tab selected. The search bar at the top right contains 'plot.estimate_f'. The main pane displays the documentation for 'plot.estimate_R' from the 'EpiEstim' package. The title is 'Plot outputs of estimate_r'. The 'Description' section explains that the plot method of 'estimate_r' objects can visualise three types of information: the epidemic curve, the posterior mean and 95% credible interval of the reproduction number, and the discrete distribution(s) of the serial interval. The 'Usage' section shows the S3 method signature for the 'plot' function, with various arguments like 'x', 'what', 'add_imported_cases', 'options_I', 'options_R', 'options_SI', 'legend', and '...'.

Files Plots Packages Help Viewer Presentation

R: Plot outputs of estimate_r ▾ Find in Topic

plot.estimate_R {EpiEstim} R Documentation

Plot outputs of estimate_r

Description

The plot method of `estimate_r` objects can be used to visualise three types of information. The first one shows the epidemic curve. The second one shows the posterior mean and 95% credible interval of the reproduction number. The estimate for a time window is plotted at the end of the time window. The third plot shows the discrete distribution(s) of the serial interval.

Usage

```
## S3 method for class 'estimate_R'
plot(
  x,
  what = c("all", "incid", "R", "SI"),
  add_imported_cases = FALSE,
  options_I = list(col = palette(), transp = 0.7, xlim = NULL, ylim = NULL, interval =
    1L, xlab = "Time", ylab = "Incidence"),
  options_R = list(col = palette(), transp = 0.2, xlim = NULL, ylim = NULL, xlab =
    "Time", ylab = "R"),
  options_SI = list(prob_min = 0.001, col = "black", transp = 0.25, xlim = NULL, ylim =
    NULL, xlab = "Time", ylab = "Frequency"),
  legend = TRUE,
  ...
)
```

Personalising figures in the incidence and EpiEstim packages

```
plot(  
  x,  
  what = c("all", "incid", "R", "SI"),  
  add_imported_cases = FALSE,  
  options_I = list(col = palette(), transp = 0.7, xlim = NULL, ylim = NULL, interval =  
    1L, xlab = "Time", ylab = "Incidence"),  
  options_R = list(col = palette(), transp = 0.2, xlim = NULL, ylim = NULL, xlab =  
    "Time", ylab = "R"),  
  options_SI = list(prob_min = 0.001, col = "black", transp = 0.25, xlim = NULL, ylim =  
    NULL, xlab = "Time", ylab = "Frequency"),  
  legend = TRUE,  
  ...  
)
```

Personalising figures in the incidence and EpiEstim packages

Arguments

x	The output of function estimate_R or function wallinga_teunis . To plot simultaneous outputs on the same plot use estimate_R_plots function
what	A string specifying what to plot, namely the incidence time series (what= 'incid'), the estimated reproduction number (what= 'R'), the serial interval distribution (what= 'SI', or all three (what= 'all')).
add_imported_cases	A boolean to specify whether, on the incidence time series plot, to add the incidence of imported cases.
options_I	For what = "incid" or "all". A list of graphical options: <ul style="list-style-type: none">col A color or vector of colors used for plotting incid. By default uses the default R colors.transp A numeric value between 0 and 1 used to monitor transparency of the bars plotted. Defaults to 0.7.xlim A parameter similar to that in par, to monitor the limits of the horizontal axisylim A parameter similar to that in par, to monitor the limits of the vertical axisinterval An integer or character indicating the (fixed) size of the time interval used for plotting the incidence; defaults to 1 day.xlab, ylab Labels for the axes of the incidence plot

Personalising figures in the incidence and EpiEstim packages

`options_R`

For what = "R" or "all". A list of graphical options:

`col`

A color or vector of colors used for plotting R. By default uses the default R colors.

`transp`

A numeric value between 0 and 1 used to monitor transparency of the 95%CrI. Defaults to 0.2.

`xlim`

A parameter similar to that in `par`, to monitor the limits of the horizontal axis

`ylim`

A parameter similar to that in `par`, to monitor the limits of the vertical axis

`xlab, ylab`

Labels for the axes of the R plot

Personalising figures in the incidence and EpiEstim packages

<code>options_SI</code>	For what = "SI" or "all". A list of graphical options:
<code>prob_min</code>	A numeric value between 0 and 1. The SI distributions explored are only shown from time 0 up to the time <code>t</code> so that each distribution explored has probability < <code>prob_min</code> to be on any time step after <code>t</code> . Defaults to 0.001.
<code>col</code>	A color or vector of colors used for plotting the SI. Defaults to black.
<code>transp</code>	A numeric value between 0 and 1 used to monitor transparency of the lines. Defaults to 0.25
<code>xlim</code>	A parameter similar to that in <code>par</code> , to monitor the limits of the horizontal axis
<code>ylim</code>	A parameter similar to that in <code>par</code> , to monitor the limits of the vertical axis
<code>xlab, ylab</code>	Labels for the axes of the serial interval distribution plot
<code>legend</code>	A boolean (TRUE by default) governing the presence / absence of legends on the plots

Personalising figures in the incidence and EpiEstim packages

EpiEstim example in R – Ebola (using data from package: outbreaks)

Personalising figures in the incidence and EpiEstim packages

These methods indirectly use the ggplot2 package

Benefits:

- Assigning plots to objects

```
> g <- plot(incidence_object)
```

```
> g
```
- Wide variety of options
- User friendly
- Outputs look more professional than using 'base' R functions

Plots as objects

Benefits of storing your plot as an object

- Add additional elements
- Change themes / style elements
- Can export as an RData file
 - > saveRDS(my_object, "filename.RDS")
 - > my_object <- loadRDS("filename.RDS")

E.g.

```
> g <- plot(incidence_object)
> saveRDS(g, "Figure1_incidence.RDS")
> ...
> g <- loadRDS("Figure1_incidence.RDS")
```

Using ggplot2

Graphics in ggplot2 are the sum of several layers (or objects) combined using the + symbol

Components:

```
> data %>% ggplot(...) + geom_type(...) + scale_what_how(...) + stat_method(...)
```


Using ggplot2

Graphics in ggplot2 are the sum of several layers (or objects) combined using the + symbol

Components:

```
> data %>% ggplot(...) + geom_type(...) + scale_what_how(...) + stat_method(...)
```



Data: what you want to
plot – usually a tibble

×1

Using ggplot2

Graphics in ggplot2 are the sum of several layers (or objects) combined using the + symbol

Components:

```
> data %>% ggplot(...) + geom_type(...) + scale_what_how(...) + stat_method(...)
```



Defining the ggplot
object

×1

Using ggplot2

Graphics in ggplot2 are the sum of several layers (or objects) combined using the + symbol

Components:

```
> data %>% ggplot(...) + geom_type(...) + scale_what_how(...) + stat_method(...)
```



Geometry: an object that
indicates a type of plot,
e.g. line, point, etc.


×1 +

Using ggplot2

Graphics in ggplot2 are the sum of several layers (or objects) combined using the + symbol

Components:

```
> data %>% ggplot(...) + geom_type(...) + scale_what_how(...) + stat_method(...)
```



One of these must contain *aesthetics*
using the `aes(...)` function – this
indicates which variables to use from
our data and how

Using ggplot2

Graphics in ggplot2 are the sum of several layers (or objects) combined using the + symbol

Components:

```
> data %>% ggplot(...) + geom_type(...) + scale_what_how(...) + stat_method(...)
```



Scale (optional): for
manipulating axes, colors,
etc.

×0 +

Using ggplot2

Graphics in ggplot2 are the sum of several layers (or objects) combined using the + symbol

Components:

```
> data %>% ggplot(...) + geom_type(...) + scale_what_how(...) + stat_method(...)
```



Statistics (optional): allow application of statistical transforms to the data

×0 +

Using ggplot2

Graphics in ggplot2 are the sum of several layers (or objects) combined using the + symbol

Other components:

- Themes
- Labels
- Etc.

Using ggplot2

Using aesthetics: `aes(...)`

- Indicate which variables correspond to x, y, etc. (i.e. which axes)
- Indicate whether you want to link color, size or shape to a variable

E.g. data is a tibble with columns: Date, NumCases, CumCases, Sex

```
> g <- data %>% ggplot(aes(x = Date, y = NumCases, color = Sex)) + geom_line()
> g <- data %>% ggplot() + geom_line(aes(x = Date, y = NumCases, color = Sex))
> g <- data %>% ggplot(aes(x = Date, y = NumCases)) + geom_line(aes(color = Sex))
```

Using ggplot2

Using aesthetics: `aes(...)`

- Indicate which variables correspond to x, y, etc. (i.e. which axes)
- Indicate whether you want to link color, size or shape to a variable

E.g. data is a tibble with columns: Date, NumCases, CumCases, Sex

```
> g <- data %>% ggplot() + geom_line(aes(x = Date, y = NumCases)) +  
  geom_point(aes(x = Date, y = CumCases))
```

Using ggplot2

Types of geometries: `geom_type(...)`

- Line: `geom_line()`
- Point: `geom_point()`
- Boxplot: `geom_boxplot()`
- Density: `geom_density()`
- Errorbars: `geom_errorbar()`

And lots more!
See ggplot2 cheatsheet on Canvas

Using ggplot2

Example: Ebola onset data

Hints and tips for beautiful (and informative) plots

- Think about using a colourblind-friendly palette
 - Would your plot still be readable in black and white?
 - Use color **and** line type / point shape to differentiate between data
- Keep it simple
- Ensure all axes are labelled and use legends if needed
 - Labels should be short and to the point
- Make sure your font and labels are readable
 - Is the font big enough?
 - Is there enough space between labels?

Weekly reading

- R for Graduate Students: Part III

https://bookdown.org/yih_huynh/Guide-to-R-Book/

- Chapter 8: Introduction to graphing
 - Chapter 9: Scatter plots
 - Chapter 10: Line graphs
 - Optional: Chapter 11 and Chapter 12
- Video: Make beautiful graphs in ggplot2
https://www.youtube.com/watch?v=qnw1xDnt_Ec