

A quick a dirty guide to using ggplot2

with data wrangling tips

Charles T. Gray

Salim Group Journal Club

January 6, 2018

Outline

Outline

Themes and objectives

- Leave today confident with everyday ggplotting.
- Also learn some of the symbiotic data wrangling techniques.
- **Leave no person behind** style of coding workshop.
- Use data visualisation and wrangling to explore who survived the Titanic's sinking.

Outline

Structure of a ggplot

ggplot2 is structured according to the **grammar of graphics**.

The basic components of a **ggplot** are:

ggplot a **mapping** of a dataset, so that R can interpret your points visually;

aes **aesthetics**, providing details such as x and y axes;

geom **geometry**, specify type of plot.

Outline

Packages

We begin by installing some packages using `library()`:

- `dplyr` for data wrangling;
- `tidyr` for data wrangling;
- `ggplot2` for data visualisation.

These packages require dataframes with rows as observations and columns as variables.

Titanic data

Today we will use survival data from the Titanic, which you may be familiar with from such ‘documentaries’ as:

Titanic (1997)

Directed by James Cameron. Ultra-Condensed by Samuel Stoddard (Movie-a-minute).

Leonardo DiCaprio

Your social class is stuffy. Let's dance with the ship's rats and have fun.

Kate Winslet

You have captured my heart. Let's run around the ship and giggle.

(The ship SINKS.)

Leonardo DiCaprio

Never let go.

Kate Winslet

I promise. *(Lets go.)*

THE END

Titanic data

```
> # A quick look at the data.
```

```
> titanic.data %>%
```

```
+   select(name, survived, age) %>%
```

```
+   head(3)
```

| | name | survived | age |
|---|--------------------------------|----------|---------|
| 1 | Allen, Miss. Elisabeth Walton | 1 | 29.0000 |
| 2 | Allison, Master. Hudson Trevor | 1 | 0.9167 |
| 3 | Allison, Miss. Helen Lorraine | 0 | 2.0000 |

Outline

Layers: first layer

The basic components of a `ggplot` are:

`ggplot` a mapping of a dataset;

`aes` **aesthetics**, providing details such as x and y axes;

`geom` **geometry**, specify type of plot.

Layers: first layer

Plots are built in layers, added using +.

Example

```
some.data %>% ggplot() + layers
```

Layers: first layer

Plots are built in layers, added using `+`.

Example

```
some.data %>% ggplot() + layers
```

Definition

The **pipe** symbol `%>%` passes the first object into the first argument of the subsequent function. Its keyboard shortcut is `control + shift + M`.

Layers: first layer

Exercise: What happens if you pipe `titanic.data` into `ggplot()` without adding any layers?

Example

```
some.data %>% ggplot()
```

Outline

Aesthetics: global aesthetics

The basic components of a `ggplot` are:

`ggplot` a **mapping** of a dataset;

`aes` **aesthetics**, providing details such as x and y axes;

`geom` **geometry**, specify type of plot.

Aesthetics

Aesthetics apply to the data (i.e., points and lines):

- x and y axes;
- colour;
- size;
- transparency;
- grouping.

Aesthetics can apply globally across all layers, or just to a specific layer.

Aesthetics: global aesthetics

When you map your data, you may specify global aesthetics using `aes()` as an argument within `ggplot()`.

Aesthetics: global aesthetics

When you map your data, you may specify global aesthetics using `aes()` as an argument within `ggplot()`.

Example

```
some.data %>% # Load some data
ggplot( # Map the data for visualisation
  aes(x = x.variable, # Specify x axis
      y = y.variable) # Specify y axis
)
```

These aesthetics set the *x* and *y* variables for all layers.

Aesthetics: global aesthetics

Exercise: Create a first layer for `titanic.data` and set the *x*-axis to `age` and the *y*-axis to `fare`.

Example

```
some.data %>% # Load some data
ggplot( # Map the data for visualisation
  aes(x = x.variable, # Specify x axis
      y = y.variable) # Specify y axis
)
```

Outline

Geometries

The basic components of a `ggplot` are:

`ggplot` a **mapping** of a dataset;

`aes` **aesthetics**, providing details such as x and y axes;

`geom` **geometry**, specify type of plot.

Geometries

Geoms are types of plots. Each geometry function begins with `geom_`:

- `histogram`

- `point` scatterplot

- `density`

- `freqpoly` a density curve of counts

- `smooth` add a smoothed regression line (default loess)

Geometries

Geoms are types of plots. Each geometry function begins with `geom_`:

`histogram`

`point` `scatterplot`

`density`

`freqpoly` a density curve of counts

`smooth` add a smoothed regression line (default loess)

Example

```
some.data %>% # Load some data
ggplot( # Map the data for visualisation
  aes(x = x.variable, # Specify x axis
      y = y.variable) + # Specify y axis
  geom_point() # Add a scatterplot layer
```

Geometries

Exercise: Create a histogram of the ages of the passengers on the Titanic.

Example

```
some.data %>% # Load some data
ggplot( # Map the data for visualisation
  aes(x = x.variable, # Specify x axis
      y = y.variable) + # Specify y axis
  geom_point() # Add a scatterplot layer
```

Local aesthetics

You can specify aesthetics at the layer level. For example, you can modify the geom.

Exercise: Modify your histogram with the histogram-specific aesthetic `binwidth` to something more meaningful.

Example

```
some.data %>%  
ggplot(  
  aes(x = x.variable,  
      y = y.variable) +  
  geom_histogram(aes(geom_specific = my.specification  
) # Add a geom layer with geom specific aesthetic.
```

Data wrangling for missing values

We can use the `filter()` function to select the rows we're interested in for our visualisation.

Exercise: Modify your histogram to filter out the missing age rows; i.e., produce your histogram without error messages.

Example

```
some.data %>%  
filter(my.variable <= some.number ) %>%  
ggplot(  
  aes(x = x.variable,  
      y = y.variable) +  
  geom_thisgeom(aes(geom.specific = my.specification  
) # Add a geom layer with geom specific aesthetic.
```

Outline

Another geom layer

We can add more layers using `+`.

Exercise: Add a `freqpoly` geom layer to your histogram.

Notice that if you have inconsistent geom-specific aesthetics your visualisation can be somewhat confusing.

Example

```
some.data %>%  
filter(my.variable <= some.number ) %>%  
ggplot(  
  aes(x = x.variable,  
      y = y.variable) +  
  geom_thisgeom(aes(geom_specific = my.specification )  
  + geom_anothergeom() # Add another geom layer
```

Other layers

Layers aren't just geoms, but can be:

- Horizontal and vertical lines;
- Titles and axis labels;
- Colour palettes and themes.

Other layers: main title

Exercise: Add a title layer to your histogram with `ggtitle()`.

Example

```
some.data %>%  
  ggplot(  
    aes(x = x.variable,  
        y = y.variable) +  
    geom_some_geom() +  
    ggtitle('my awesome title') # Add a title
```


Outline

Facets

Facets subdivide your plot into subplots by variables. There are two ways to facet:

`wrap` A horizontal row of plots faceted by one variable.

`grid` A grid of plots faceted by two variables.

Example (wrap)

```
some.data %>%  
ggplot(  
  aes(x = x.variable,  
       y = y.variable) +  
geom_some_geom() +  
facet_wrap(~ horizontal.facet.variable)
```

Facets

Facets subdivide your plot into subplots by variables. There are two ways to facet:

`wrap` A horizontal row of plots faceted by one variable.

`grid` A grid of plots faceted by two variables.

Example (grid)

```
some.data %>%  
ggplot(  
  aes(x = x.variable,  
       y = y.variable) +  
geom_some_geom() +  
facet_grid (vertical.facet.variable ~  
horizontal.facet.variable)
```

Outline

Resources

- RStudio data-wrangling and data-visualisation cheat sheets.
- Edward Tufte's *The Visual Display of Quantitative Information* - 'The Strunk and White of visualisation'
- Hadley Wickham's *Elegant Graphics for Data Analysis*

Outline

Who survived the sinking of the Titanic?

Exercise: Create a visualisation that compares how the different genders survived the sinking of the Titanic.

Exercise: Create a visualisation that compares how the different classes survived the sinking of the Titanic.

Exercise: Create a visualisation that compares how the different classes and genders survived the sinking of the Titanic.

Exercise: Create a visualisation that compares how ages and classes survived the sinking of the Titanic.