



Data visualisation

66

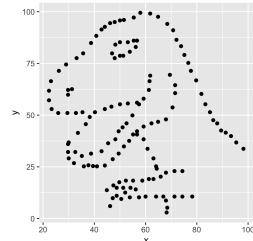
*The greatest value of a picture
is when it forces us to notice
what we never expected to see.
-- John W. Tukey*



2 / 57

numbers vs plots

```
dino
#> # A tibble: 142 x 2
#>   x     y
#>   <dbl> <dbl>
#> 1 55.4  97.2
#> 2 51.5  96.0
#> 3 46.2  94.5
#> 4 42.8  91.4
#> 5 40.8  88.3
#> 6 38.7  84.9
#> # ... with 136 more rows
```



3 / 57

numbers vs plots

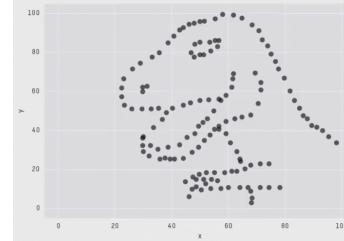


image credit: Steph Locke

X Mean: 54.2659224
Y Mean: 47.8313999
X SD : 16.7649829
Y SD : 26.9342120
Corr. : -0.0642526

4 / 57

Why data visualisation? 📊

“

A picture is worth a thousand words. -- Henrik Ibsen

1. Data visualisation communicates information much quicker than numerical tables.
2. Data visualisation can reveal unexpected structures in data; it is not surprising that data visualisation is one of the key tools in exploratory data analysis.
3. Data plot is usually more eye-catching even if you lose accuracy of the information.

5 / 57

Charts 📈 Graphics

6 / 57

A toy example

```
sci_tbl  
  
#> # A tibble: 4 × 2  
#>   dept      count  
#>   <chr>     <int>  
#> 1 Physics      12  
#> 2 Mathematics    8  
#> 3 Statistics    20  
#> 4 Computer Science 23
```

➤ dept: discrete/categorical
➤ count: quantitative/numeric

What types of plots can we make?

1. bar plot for counts
2. pie chart for proportions

7 / 57

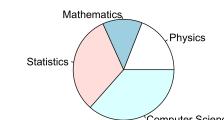
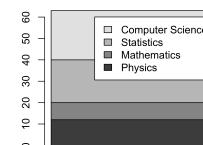
Named charts

➤ Bar plot

```
barplot(as.matrix(sci_tbl$count),  
       legend = sci_tbl$dept)
```

➤ Pie chart

```
pie(sci_tbl$count,  
    labels = sci_tbl$dept)
```



8 / 57

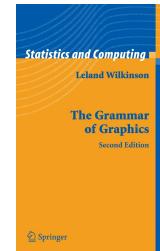
Seems convenient, but ...

- ✗ a limited set of named charts
- ✗ single purpose functions
- ✗ inconsistent inputs

```
barplot(as.matrix(sci_tbl$count),  
       legend = sci_tbl$dept)
```

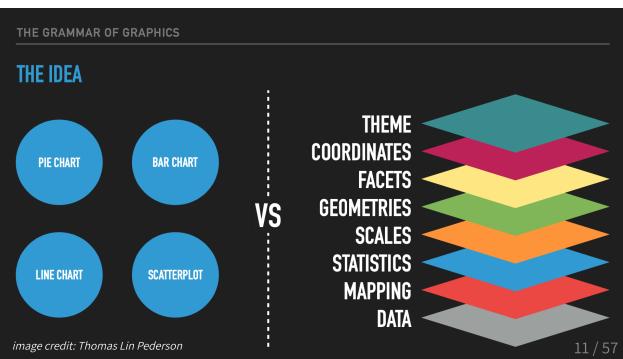
```
pie(sci_tbl$count,  
    labels = sci_tbl$dept)
```

9 / 57



Grammar makes language expressive. A language consisting of words and no grammar (statement = word) expresses only as many ideas as there are words. By specifying how words are combined in statements, a grammar expands a language's scope.

10 / 57

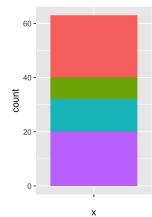


The *grammar of graphics* takes us beyond a limited set of **charts (words)** to an almost unlimited world of **graphical forms (statements)**.

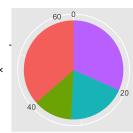
{ggplot2} provides a cohesive system for declaratively creating elegant graphics, based on The Grammar of Graphics.

12 / 57

```
library(ggplot2)
ggplot(data = sci_tbl) +
  geom_bar(
    aes(x = "", y = count, fill = dept),
    stat = "identity"
  )
```



```
ggplot(data = sci_tbl) +
  geom_bar(
    aes(x = "", y = count, fill = dept),
    stat = "identity"
  ) +
  coord_polar(theta = "y")
```



13 / 57

A graphing template

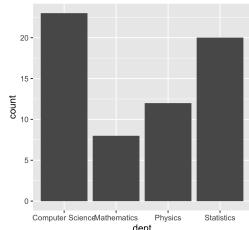
```
ggplot(data = <DATA>, mapping = aes(<MAPPINGS>)) +
  layer(geom = <GEOM>, stat = <STAT>, position = <POSITION>) +
  layer(geom = <GEOM>, stat = <STAT>, position = <POSITION>)
```

1. **data:** tibble/data.frame.
2. **mapping:** aesthetic mappings between data variables and visual elements, via `aes()`.
3. **layer ():** a graphical layer is a combination of data, stat and geom with a potential position adjustment.
 - **geom:** geometric elements to render each data observation.
 - **stat:** statistical transformations applied to the data prior to plotting.
 - **position:** position adjustment, such as "identity", "stack", "dodge" etc.

14 / 57

Layers: a bar chart

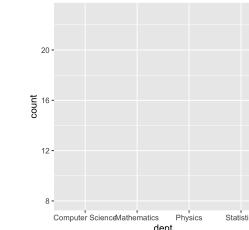
```
ggplot(data = sci_tbl, mapping = aes(x = dept, y = count)) +
  layer(geom = "bar", stat = "identity", position = "identity")
```



15 / 57

Aesthetic mapping: positional

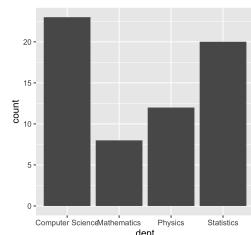
```
p <- ggplot(sci_tbl, aes(x = dept, y = count))
```



16 / 57

Geoms (a shorthand to layer())

```
p +  
  geom_bar(stat = "identity")
```



```
p +  
  geom_col()
```

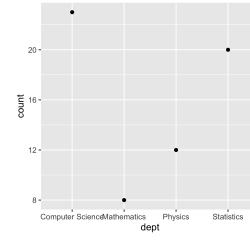
- `stat = "identity"` leaves data as is.
- `geom_col()` is a shortcut to `geom_bar(stat = "identity")`.

Generally, we use `geom_*` instead of `layer()` in practice.

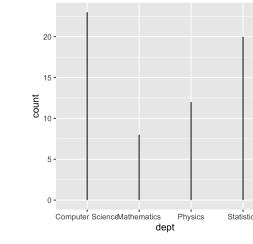
17 / 57

Geoms

```
p +  
  geom_point()
```



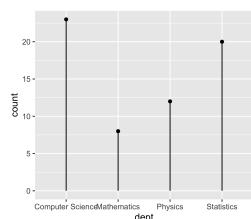
```
p +  
  geom_segment(aes(xend = dept, y = 0, yend = count))
```



18 / 57

Composite geoms: lollipop 🍬 = points + segments

```
p +  
  geom_point() +  
  geom_segment(aes(xend = dept, y = 0, yend = count))
```



19 / 57

Geom catalogue

geom

`geom_abline`, `geom_hline`,
`geom_vline`

Reference lines: horizontal, vertical, and diagonal

`geom_bar`, `geom_col`

Bar charts

`geom_bin2d`

Heatmap of 2d bin counts

`geom_blank`

Draw nothing

`geom_boxplot`

A box and whiskers plot (in the style of Tukey)

Previous 1 2 3 4 5 6 7 Next

source code: Emi Tanaka

20 / 57

Stats

» Aggregated (pre-computed)

```
sci_tbl  
#> # A tibble: 4 x 2  
#>   dept      count  
#>   <chr>     <int>  
#> 1 Physics      12  
#> 2 Mathematics    8  
#> 3 Statistics     20  
#> 4 Computer Science 23  
#> # ... with 57 more rows
```

» Disaggregated

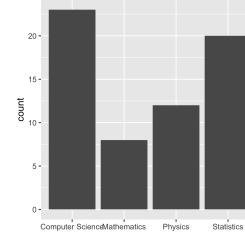
```
sci_tbl0
```

```
#> # A tibble: 63 x 1  
#>   dept  
#>   <chr>  
#> 1 Physics  
#> 2 Mathematics  
#> 3 Statistics  
#> 4 Computer Science  
#> 5 Physics  
#> 6 Mathematics  
#> 7 Statistics  
#> 8 Physics  
#> 9 Mathematics  
#> 10 Statistics  
#> 11 Physics  
#> 12 Mathematics  
#> 13 Statistics  
#> 14 Physics  
#> 15 Mathematics  
#> 16 Statistics  
#> 17 Physics  
#> 18 Mathematics  
#> 19 Statistics  
#> 20 Physics  
#> 21 Mathematics  
#> 22 Statistics  
#> 23 Physics  
#> 24 Mathematics  
#> 25 Statistics  
#> 26 Physics  
#> 27 Mathematics  
#> 28 Statistics  
#> 29 Physics  
#> 30 Mathematics  
#> 31 Statistics  
#> 32 Physics  
#> 33 Mathematics  
#> 34 Statistics  
#> 35 Physics  
#> 36 Mathematics  
#> 37 Statistics  
#> 38 Physics  
#> 39 Mathematics  
#> 40 Statistics  
#> 41 Physics  
#> 42 Mathematics  
#> 43 Statistics  
#> 44 Physics  
#> 45 Mathematics  
#> 46 Statistics  
#> 47 Physics  
#> 48 Mathematics  
#> 49 Statistics  
#> 50 Physics  
#> 51 Mathematics  
#> 52 Statistics  
#> 53 Physics  
#> 54 Mathematics  
#> 55 Statistics  
#> 56 Physics  
#> 57 Mathematics  
#> 58 Statistics  
#> 59 Physics  
#> 60 Mathematics  
#> 61 Statistics  
#> 62 Physics  
#> 63 Mathematics  
#> 64 Statistics
```

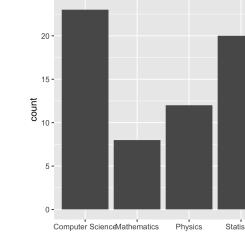
21 / 57

Stats

```
ggplot(sci_tbl, aes(x = dept, y = count)) +  
  geom_bar(stat = "identity")
```



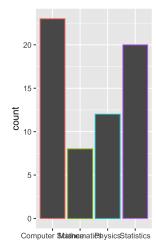
```
ggplot(sci_tbl0, aes(x = dept)) +  
  geom_bar(stat = "count")
```



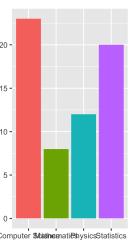
22 / 57

Aesthetic mapping: visual

```
p +  
  geom_col(aes(colour = dept))
```



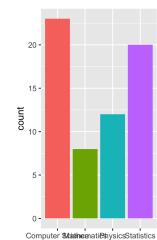
```
p +  
  geom_col(aes(fill = dept))
```



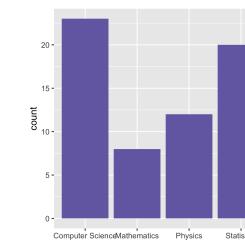
23 / 57

Mapping variables / Setting constants

```
p +  
  geom_col(aes(fill = dept))
```



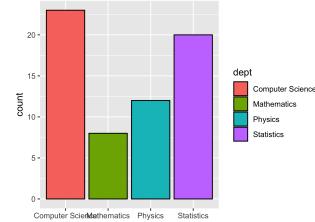
```
p +  
  geom_col(fill = "#756bb1")
```



24 / 57

Mapping variables + Setting constants

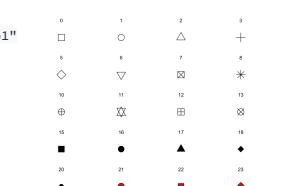
```
p +
  geom_col(aes(fill = dept), colour = "#000000")
```



25 / 57

Visual aesthetics

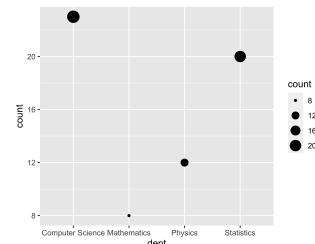
- colour/color, fill:
 - named colours, e.g. "red"
 - RGB specification, e.g. "#756bb1"
- alpha: opacity between 0 and 1
- shape:
 - an integer between 0 and 25
 - a single string, e.g. "triangle open"
- linetype:
 - an integer between 0 and 6
 - a single string, e.g. "dashed"
- size, radius: a numerical value (in millimetres)



26 / 57

Your turn

Describe a bubble chart in terms of grammar of graphics.

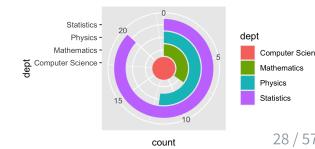


27 / 57

Coords

- Coordinate systems
 - coord_cartesian() (default)
 - coord_flip() (deprecated; now you can simply swap x and y)
 - coord_map()
 - coord_polar()

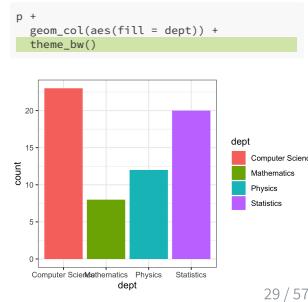
```
p +
  geom_col(aes(fill = dept)) +
  coord_polar(theta = "y")
```



28 / 57

Themes: modify the look

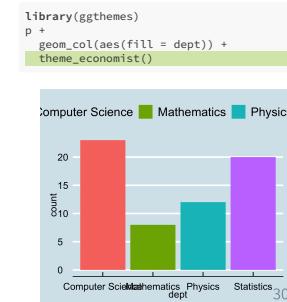
- » Built-in ggplot themes
 - » theme_grey() / theme_gray()
 - » theme_bw(), theme_linedraw()
 - » theme_light(), theme_dark()
 - » theme_minimal(),
 - theme_classic()
 - » theme_void()



29 / 57

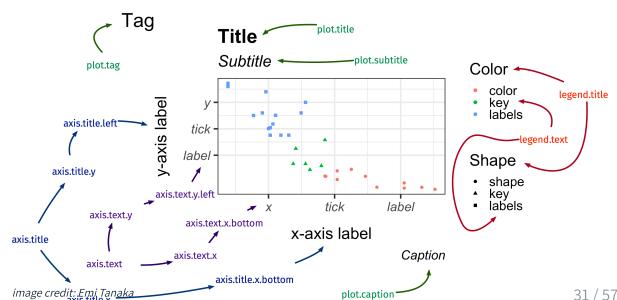
Themes: modify the look

- Many R packages provide themes
 - `{ggthemes}`
 - `{ggthemr}`
 - `{hrbrthemes}`
 - `{ggtech}`



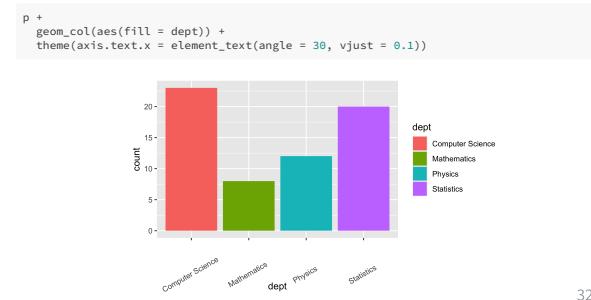
stics 30 / 57

Modify the look of texts with element_text()



31 / 57

Modify the look of texts with element_text()



32 / 57

Modify the look of *lines* with element_line()

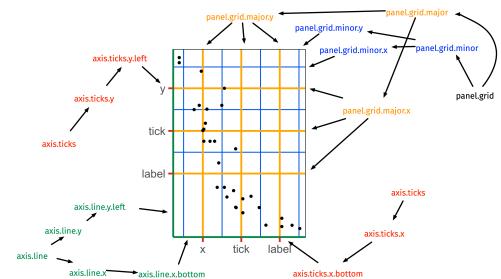


image credit: Emi Tanaka

33 / 57

Modify the look of *regions* with element_rect()

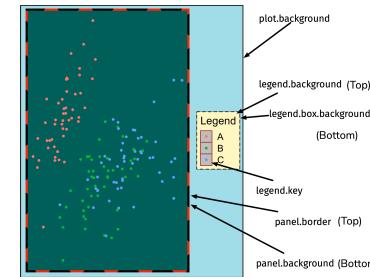


image credit: Emi Tanaka

34 / 57

Small multiples (or trellis/faceting plots)

💡 the idea of conditioning on the values taken on by one or more of the variables in a data set

35 / 57

Facets

mpg data available from {ggplot2}

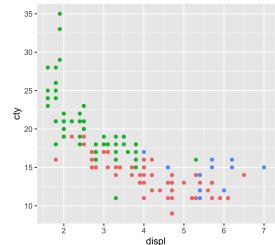
mpg

```
#> # A tibble: 234 x 11
#>   manufacturer model displ year cyl trans drv cty hwy fl class
#>   <chr>        <chr>  <dbl> <int> <chr>  <chr> <int>
#> 1 audi         a4      1.8  1999   4 auto(l5) f     18
#> 2 audi         a4      1.8  1999   4 manual(m_ f     21
#> 3 audi         a4      2    2008   4 manual(m_ f     20
#> 4 audi         a4      2    2008   4 auto(av) f     21
#> 5 audi         a4      2.8  1999   6 auto(l5) f     16
#> 6 audi         a4      2.8  1999   6 manual(m_ f     18
#> # ... with 228 more rows, and 3 more variables: hwy <int>,
#> #   fl <chr>, class <chr>
```

36 / 57

Facets

```
p_mpg <- ggplot(mpg, aes(displ, cty)) +  
  geom_point(aes(colour = drv))  
p_mpg
```

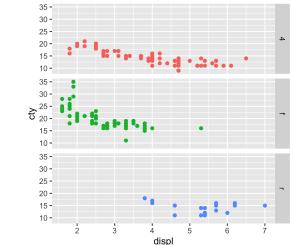


37 / 57

Facets

- `facet_grid()`

```
p_mpg +  
  facet_grid(rows = vars(drv))  
# facet_grid(~drv)
```

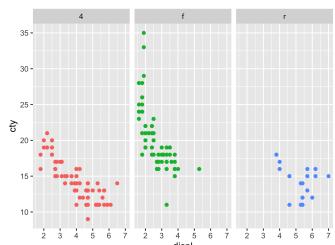


38 / 57

Facets

- `facet_grid()`

```
p_mpg +  
  facet_grid(cols = vars(drv))  
# facet_grid(drv ~ .)
```

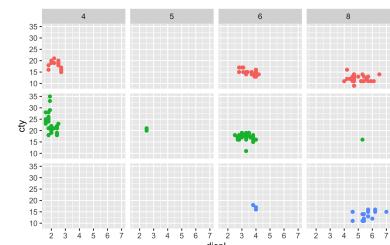


39 / 57

Facets

- `facet_grid()`

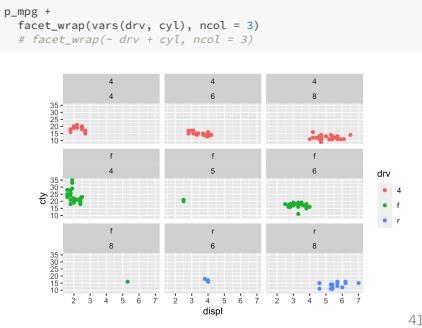
```
p_mpg +  
  facet_grid(rows = vars(drv), cols = vars(cyl))  
# facet_grid(cyl ~ drv)
```



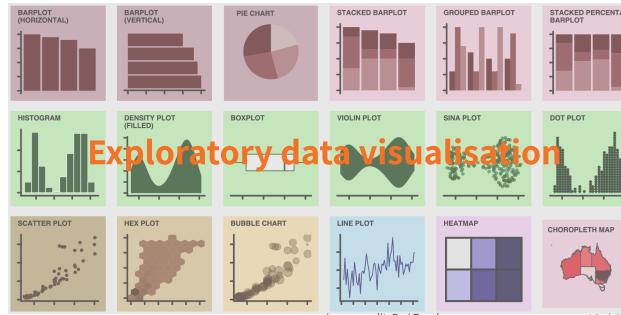
40 / 57

Facets

- `facet_grid()`
- `facet_wrap()`



41 / 57



Exploratory data visualisation

image credit: Emi Tanaka

42 / 57

case study

- `import`

```
movies <- as_tibble(jsonlite::read_json(
  "https://vega.github.io/vega-editor/app/data/movies.json",
  simplifyVector = TRUE))
movies
```

```
#> # A tibble: 3,201 x 16
#>   Title          US_Gross Worldwide_Gross US_DVD_Sales
#>   <chr>         <int>        <dbl>        <int>
#> 1 The Land Girls 146083       146083        NA
#> 2 First Love, Last Ri... 10876        10876        NA
#> 3 I Married a Strange... 203134       203134        NA
#> 4 Let's Talk About Sex 373615       373615        NA
#> 5 Slam            1089819      1087521        NA
#> 6 Mississippi Mermaid 24551        2624551       NA
#> # ... with 3,195 more rows, and 12 more variables:
#> #   Production_Budget <int>, Release_Date <chr>,
#> #   MPAA_Rating <chr>, Running_Time_min <int>,
#> #   Distributor <chr>, Source <chr>, Major_Genre <chr>,
#> #   Creative_Type <chr>, Director <chr>,
#> #   Rotten_Tomatoes_Rating <int>, IMDB_Rating <dbl>,
#> #   IMDB_Votes <int>
```

43 / 57

case study

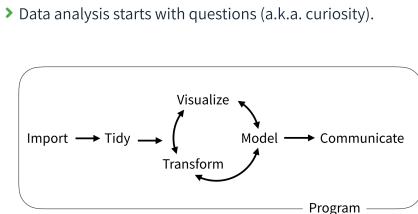
- `import`
- `skim`

```
skimr::skim(movies)
```

Data Summary	
#> Name	movies
#> Number of rows	3201
#> Number of columns	16
#> Column type frequency:	
#> character	8
#> numeric	8
#> Group variables	None
#> Variable type: character	
#> skim_variable n_missing complete_rate min max empty n_unique whitespace	
#> 1 Title 1 1.00 1 66 0 3176 0	
#> 2 Release_Date 7 0.998 8 11 0 1603 0	
#> 3 MPAA_Rating 685 0.911 1 9 0 7 0	
#> 4 Distributor 232 0.928 33 9 0 174 0	
#> 5 Source 365 0.886 6 29 0 18 0	
#> 6 Major_Genre 275 0.914 5 19 0 12 0	
#> 7 Creative_Type 446 0.861 7 23 0 9 0	
#> 8 Director 1331 0.584 7 27 0 550 0	
#> Variable type: numeric	
#> skim_variable n_missing complete_rate mean sd	
#> 1 US_Gross 7 0.998 44002085. 62555311. 44 / 57	
#> 2 Worldwide_Gross 7 0.998 85343400. 149947343.	

case study

- import
- skim
- vis



➤ Data analysis starts with questions (a.k.a. curiosity).

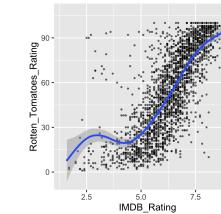
45 / 57

case study

- import
- skim
- vis

② Are movies ratings consistent b/t IMDB & Rotten Tomatoes

```
ggplot(movies, aes(x = IMDB_Rating, y = Rotten_Tomatoes_Rating)) +  
  geom_point(size = 0.5, alpha = 0.5) +  
  geom_smooth(method = "gam") +  
  theme(aspect.ratio = 1)
```



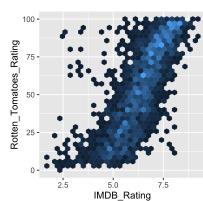
46 / 57

case study

- import
- skim
- vis

② Are movies ratings consistent b/t IMDB & Rotten Tomatoes

```
ggplot(movies, aes(x = IMDB_Rating, y = Rotten_Tomatoes_Rating)) +  
  geom_hex() +  
  theme(aspect.ratio = 1)
```



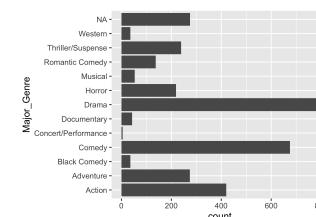
47 / 57

case study

- import
- skim
- vis

② The popularity of major genre

```
ggplot(movies, aes(y = Major_Genre)) +  
  geom_bar()
```



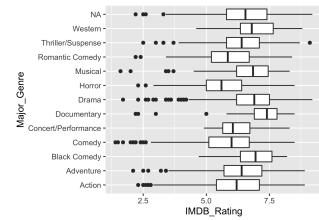
48 / 57

case study

- import
- skim
- vis

② The likeness of major genre

```
ggplot(movies) +  
  geom_boxplot(aes(x = IMDB_Rating, y = Major_Genre))
```



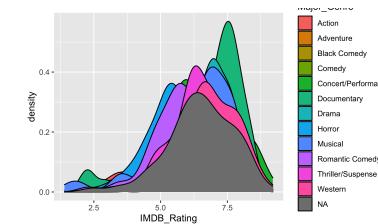
49 / 57

case study

- import
- skim
- vis

② The likeness of major genre

```
ggplot(movies) +  
  geom_density(aes(x = IMDB_Rating, fill = Major_Genre))
```

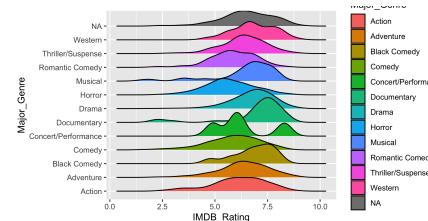


50 / 57

case study

- import
- skim
- vis

```
library(agridges)  
ggplot(movies, aes(x = IMDB_Rating, y = Major_Genre)) +  
  geom_density_ridges(aes(fill = Major_Genre))
```



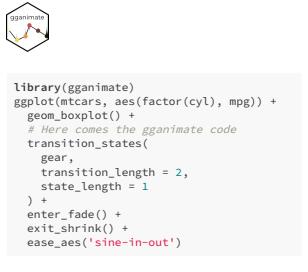
51 / 57

{ggplot2}-ext

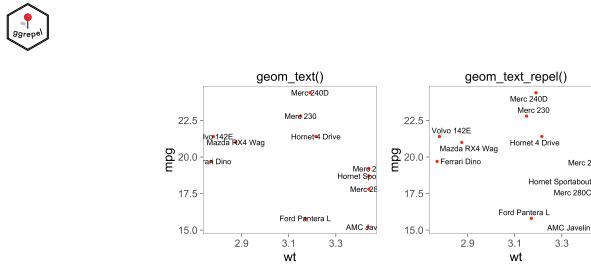
 {ggplot2} now has an official extension mechanism. This means that others can now easily create their own stats, geoms and positions, and provide them in other packages. This should allow the ggplot2 community to flourish, even as less development work happens in ggplot2 itself.

 <https://exts.ggplot2.tidyverse.org/gallery/>

52 / 57

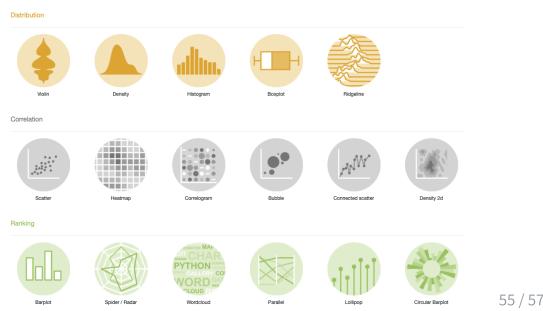


53 / 57



54 / 57

The R Graph Gallery



55 / 57

To be
continued

2

John Burn-Murdoch
[@burnmurdoch](https://twitter.com/burnmurdoch)

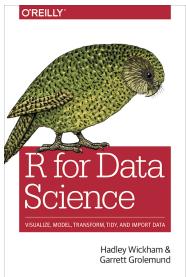
NEW: the Thursday 19 March update of our coronavirus mortality trajectories tracker

- Italy now has more Covid-19 deaths than China's total
 - UK remains on a steeper mortality curve than Italy, while Britain remains far from lockdown

Live version here: ft.com/content/a26fbf

56 / 57

Reading



- [Data visualisation](#)
- [\[ggplot2\] cheatsheet](#)