Environmental Computing

Better figures with ggplot2

Fonti Kar

@fonti_kar

R Ladies

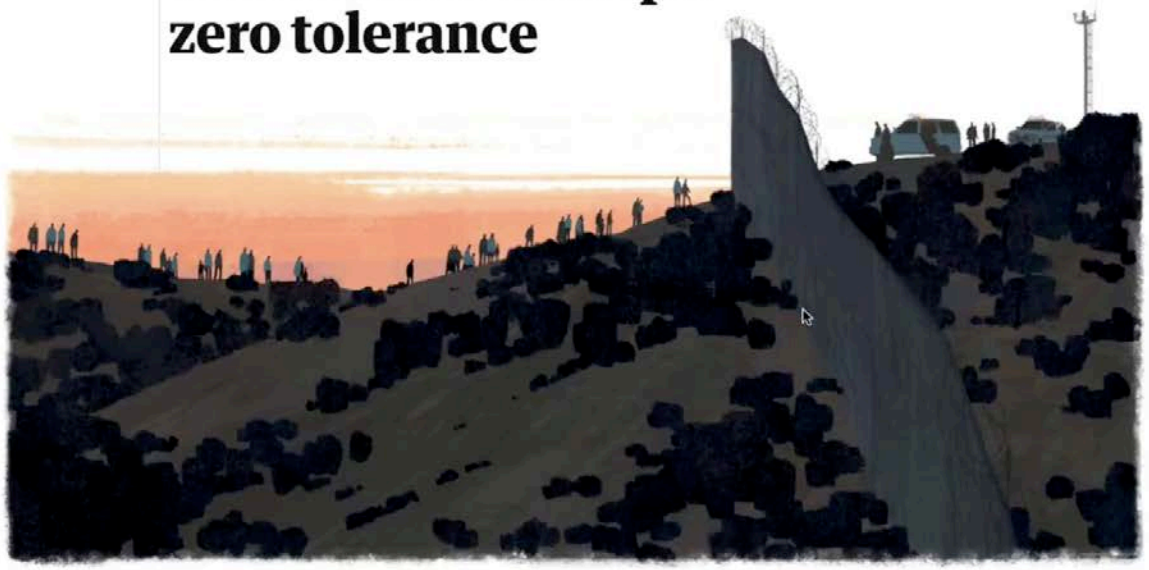
What's the point in plotting?

Explore

- Data entry errors
- Understand the structure
- Types of variables
- Distributions of variables (normal, poisson, etc.)

Communicate

3,121 desperate journeys
Exposing a week of chaos under Trump's zero tolerance



They came to the US seeking a better life. They ended up behind bars. Thousands of documents analyzed by the Guardian provide the most comprehensive picture yet of what happened to immigrants prosecuted under the Trump administration's zero tolerance policy

R vs. ggplot2

https://flowingdata.com/2016/03/22/comparing-ggplot2-and-r-base-graphics/

VS.

- plot() and par()
- Error bars are a nightmare
- Very specific functions/packages for different plots e.g. lattice::

- **Tidy, intuitive, readable coding**
- **Many handy built-in functions**
- **Less time spent on grumpy coding**
- **The ‘profesh’ look**
- Complex customising (if you need it)

https://flowingdata.com/2016/03/22/comparing-ggplot2-and-r-base-graphics/

The recipe

```
ggplot(data, aes(x_var, y_var))
    + geom_plot()
    + theme_ ()
```

'aesthetics' – what you want depicted in your plot

what type of plot do you want?

the 'look' of the plot and other nitty gritty things (e.g. axes, font size, legend position)

An example

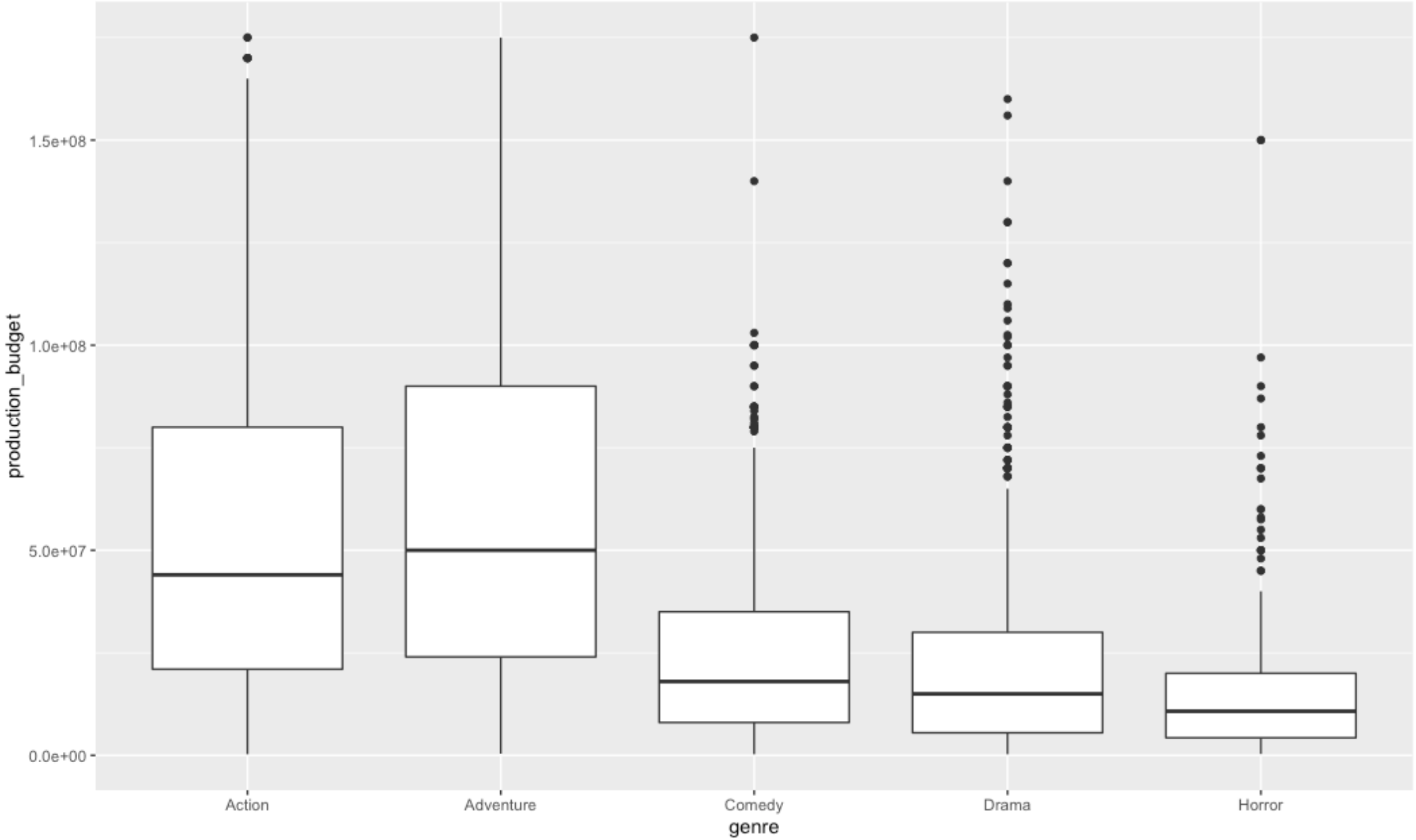
‘aesthetics’ – what you want depicted in your plot

```
ggplot(movies, aes(x = genre,y = production_budget))
        + geom_boxplot()
        + theme_bw()
```

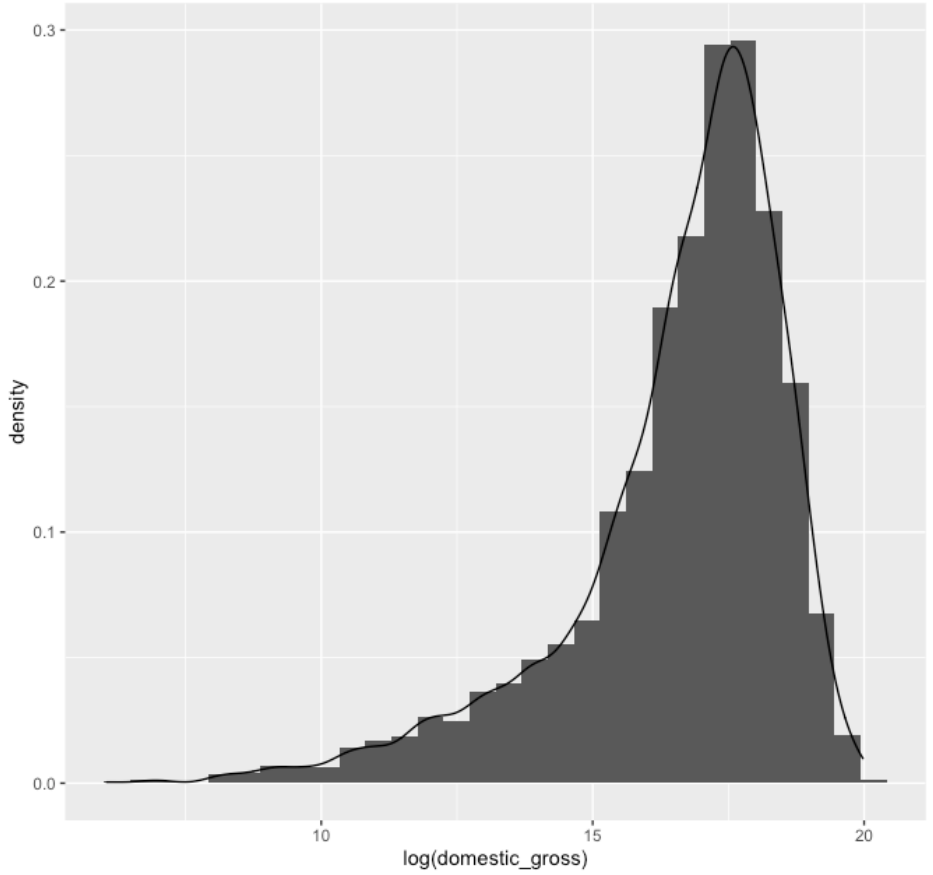
what type of plot do you want?

the ‘look’ of the plot and other nitty gritty things (e.g. axes, font size, legend position)

```
ggplot(movies, aes(x = genre,y = production_budget)) + geom_boxplot() + theme_bw()
```

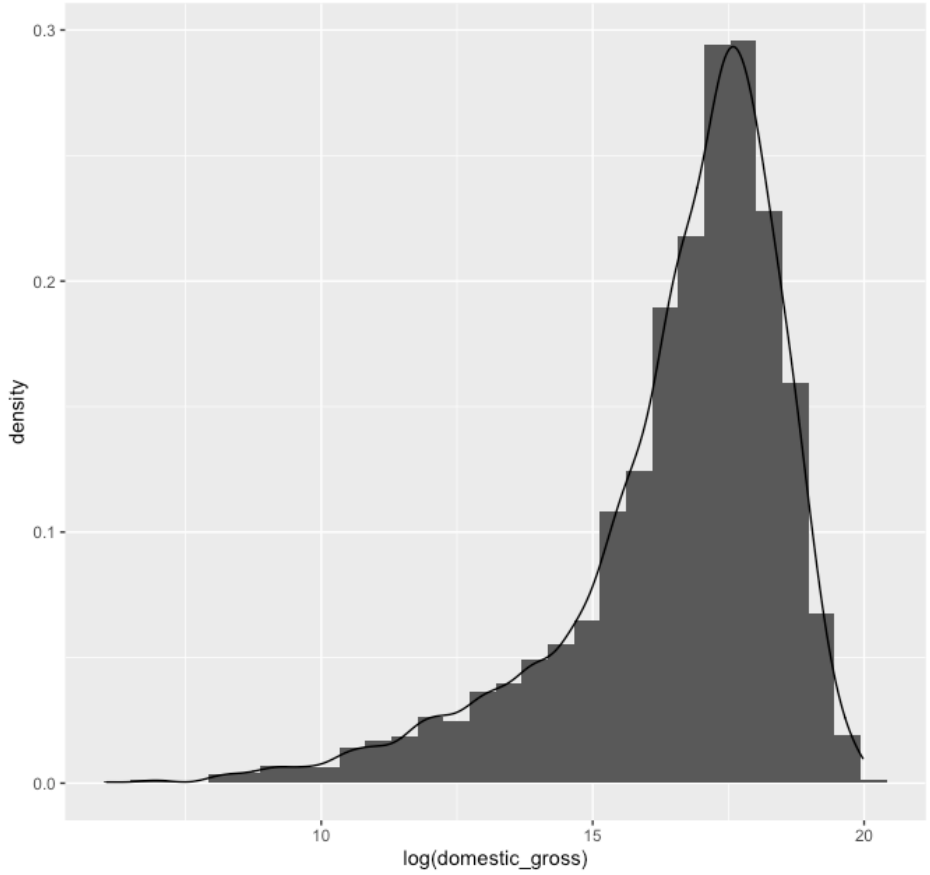


Plots: Explore the data

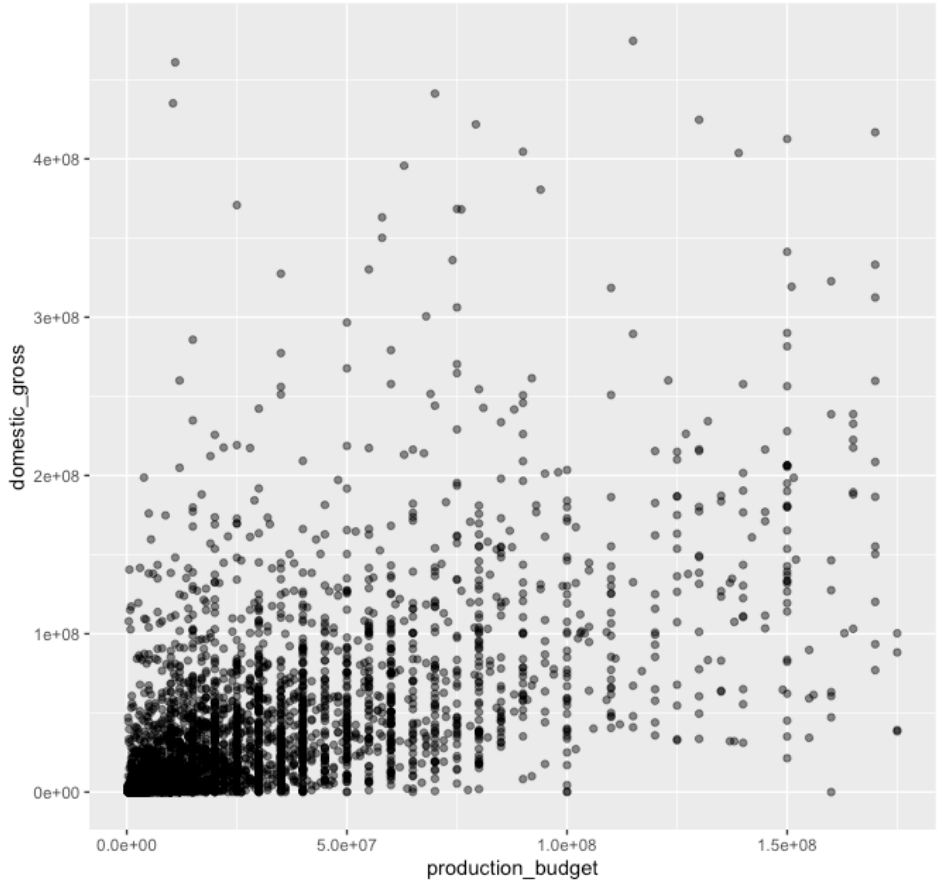



```
ggplot(data = movies, aes(x = log(domestic_gross)))
        + geom_histogram(aes(y = ..density..))
        + geom_density()
```

Plots: Explore the data

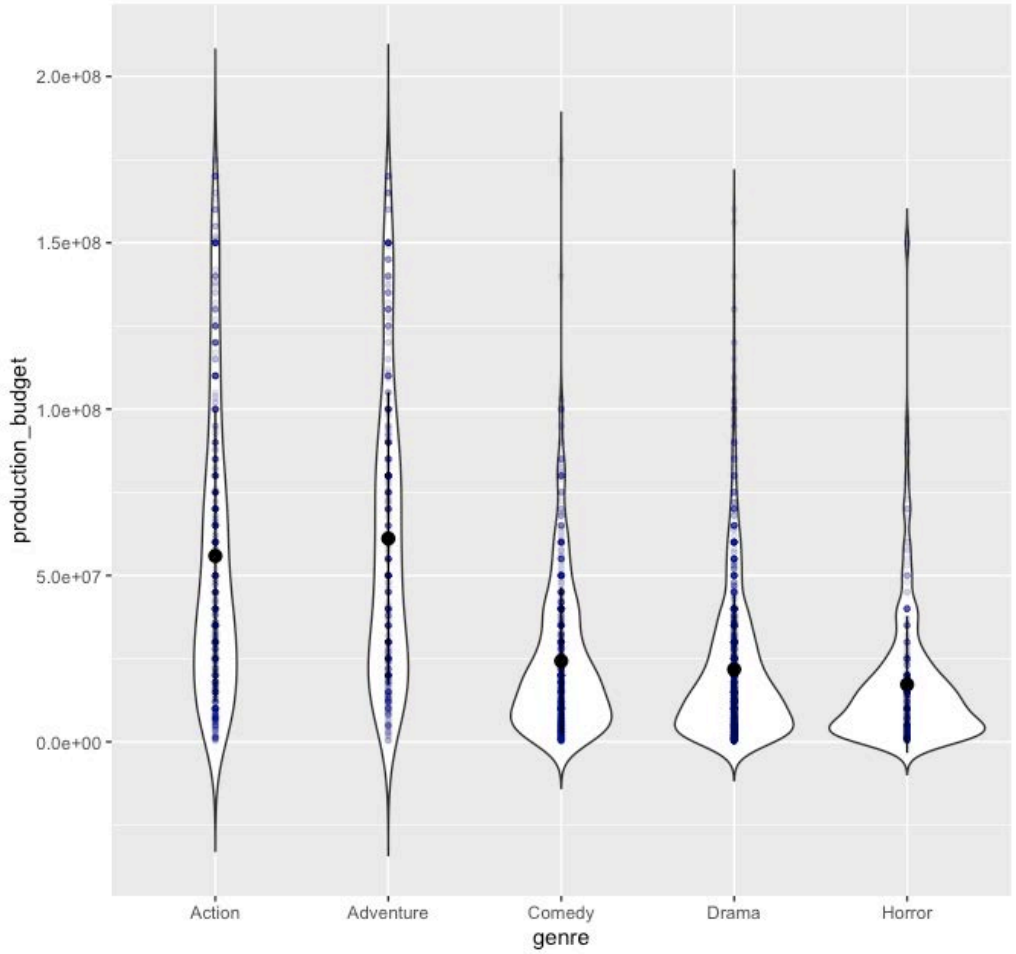



```
ggplot(data = movies, aes(x = log(domestic_gross)))
        + geom_histogram(aes(y = ..density..))
        + geom_density()
```




```
ggplot(data = movies,
aes(x = production_budget, y = domestic_gross))
 + geom_point(alpha = 0.5)
```

Plots: Explore the data



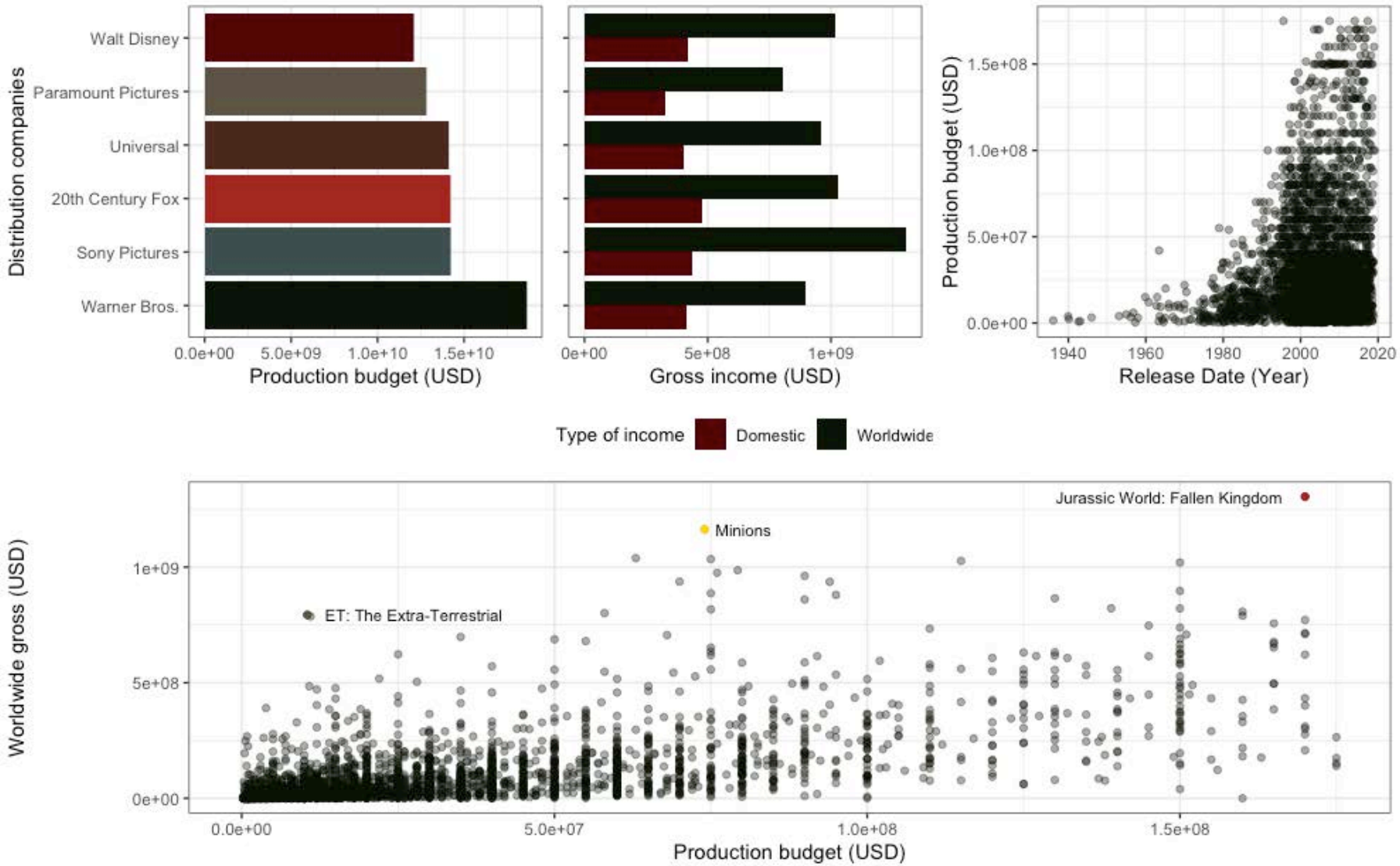

```
ggplot(data = movies, aes(x = genre, y = production_budget))
        + geom_violin(trim = F)
        + geom_point(colour = "dark grey", size = 1, alpha = 0.1)
        + stat_summary(fun.data="data_summary", col = "black")
```

Geom bonanza!

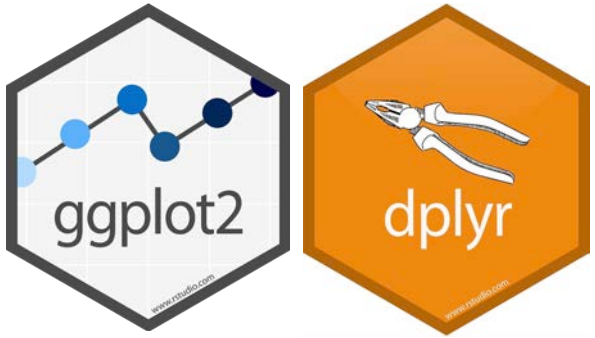
Get the cheat sheet!

https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf

Using a few tools, tips and tricks…

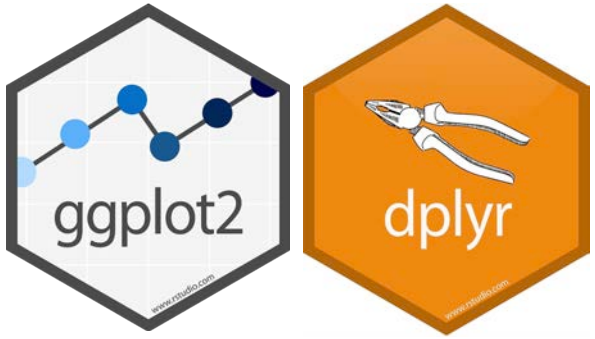


…you can **create and communicate**



Wrangle data and plot it!

I want plot of **mean** worldwide gross and **standard error bars** for movie rating

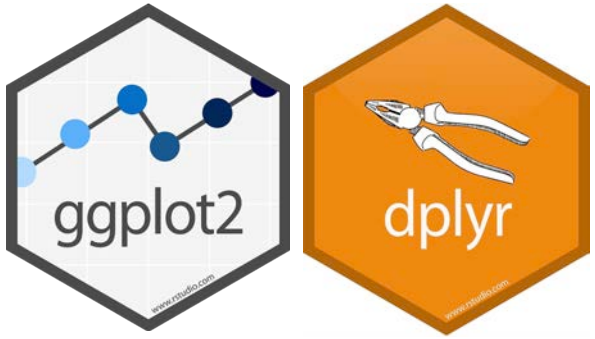


Wrangle data and plot it!

I want plot of **mean** worldwide gross and **standard error bars** for movie rating

1) Compute what you want

```
mpaa_rating_means <- movies %>%                                  #assign to new dataframe
  filter(! is.na(mpaa_rating)) %>%                               #ditching NA
  group_by(mpaa_rating) %>%                                      #group by each rating category
  summarise(mean_worldwide_gross = mean(worldwide_gross),
            se_worldwide_gross = sd(worldwide_gross)/sqrt(length(worldwide_gross)),
            lower_worldwide_gross = mean_worldwide_gross - se_worldwide_gross,
            upper_worldwide_gross = mean_worldwide_gross + se_worldwide_gross)
```

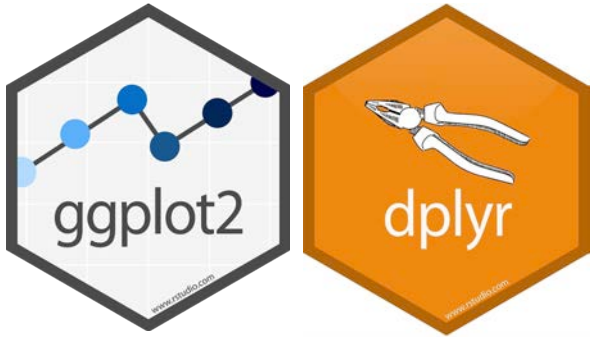


Wrangle data and plot it!

I want plot of **mean** worldwide gross and **standard error bars** for movie rating

2) Check your data

```
> head(mpaa_rating_means)
# A tibble: 4 x 5
  mpaa_rating mean_worldwide_gross se_worldwide_gross lower_worldwide_gross upper_worldwide_gross
  <chr>                      <dbl>               <dbl>                 <dbl>                 <dbl>
1 G                    175861186.          21295468.             154565718.            197156654.
2 PG                   142060672.           7636026.             134424646.            149696699.
3 PG-13                115431806.           4705281.             110726525.            120137088.
4 R                     63201659.           2478172.              60723487.             65679831.
```

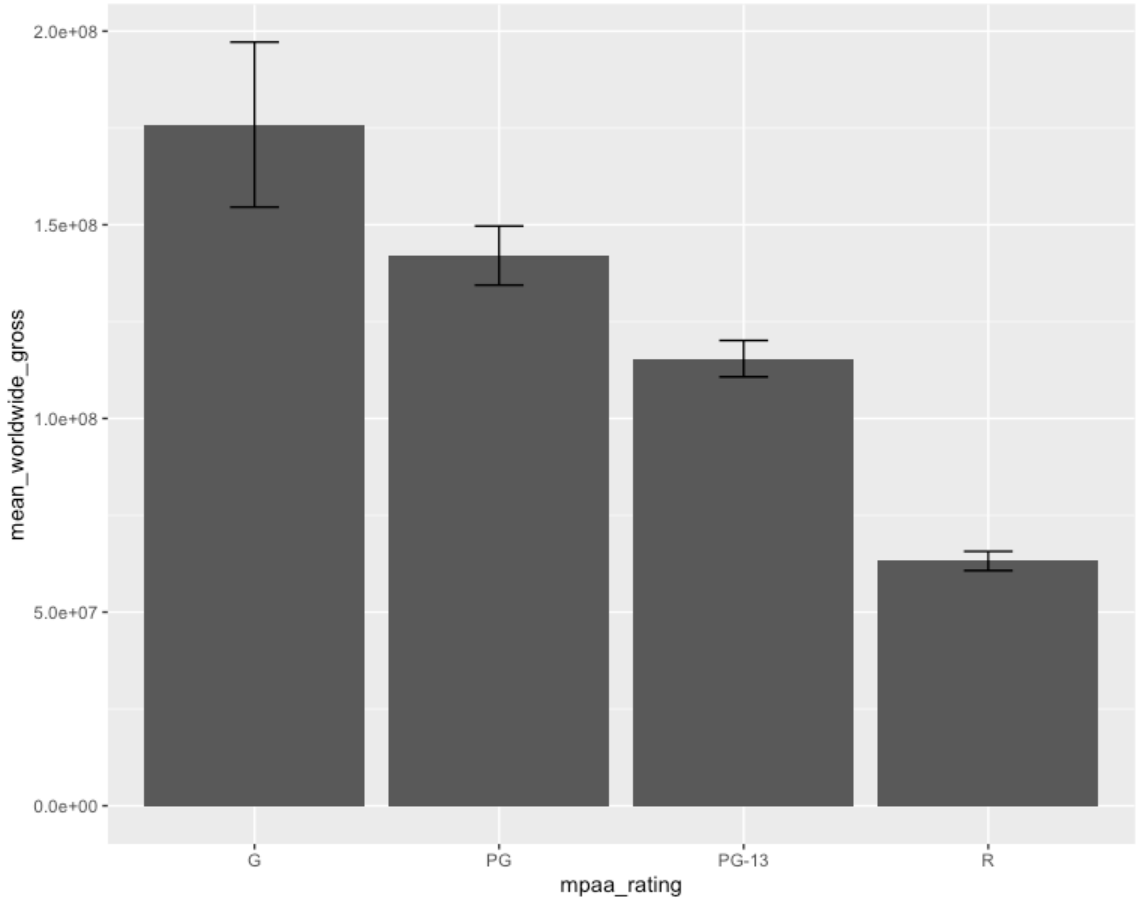


Wrangle data and plot it!

I want plot of **mean** worldwide gross and **standard error bars** for movie rating

3) Plot your data

```
ggplot(data = mpaa_rating_means,
aes(x = mpaa_rating,
    y = mean_worldwide_gross))
+ geom_col()
+ geom_errorbar(aes(
ymin = lower_worldwide_gross,
ymax = upper_worldwide_gross),
                width = 0.2)
```



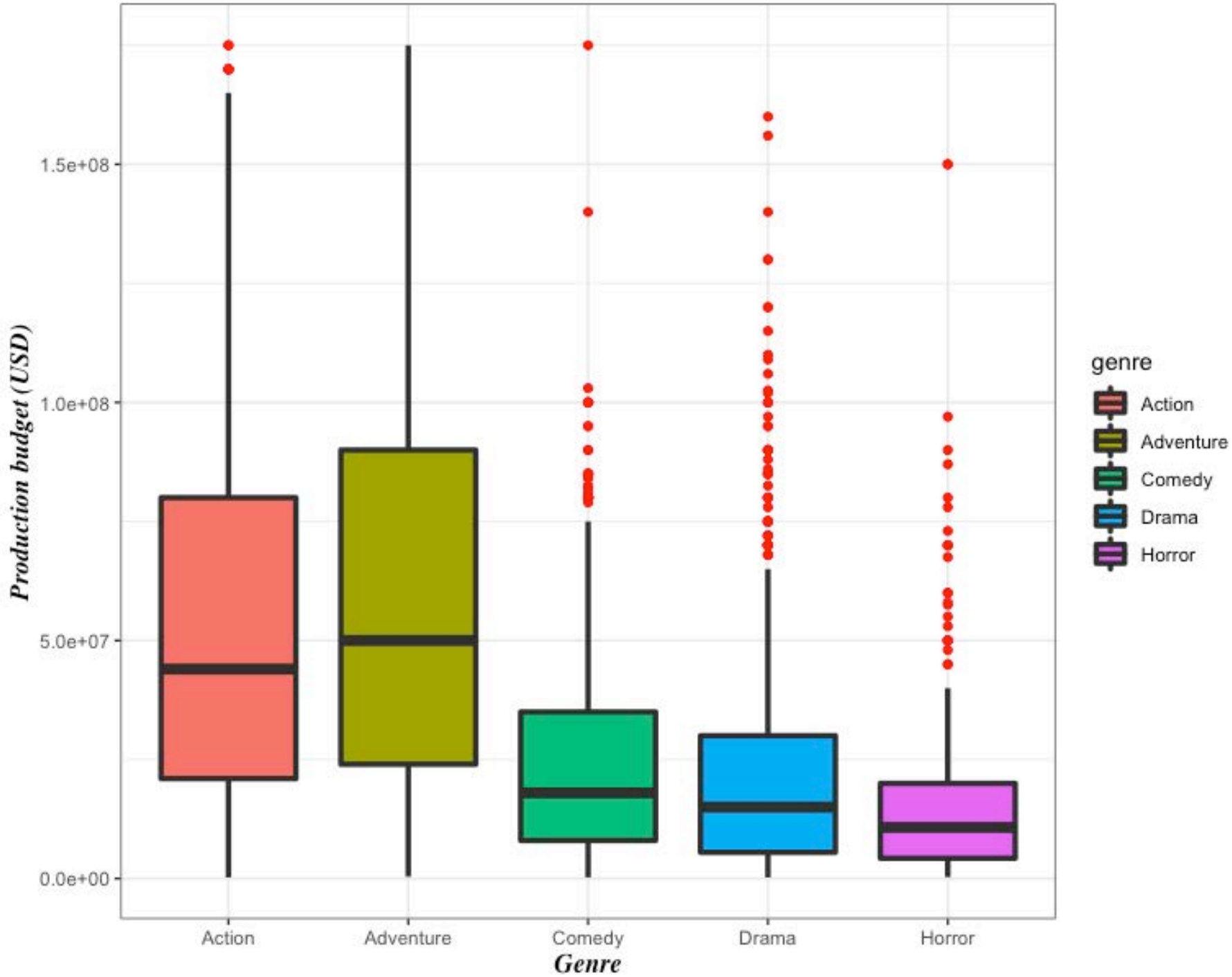
Customising your plot a bit more

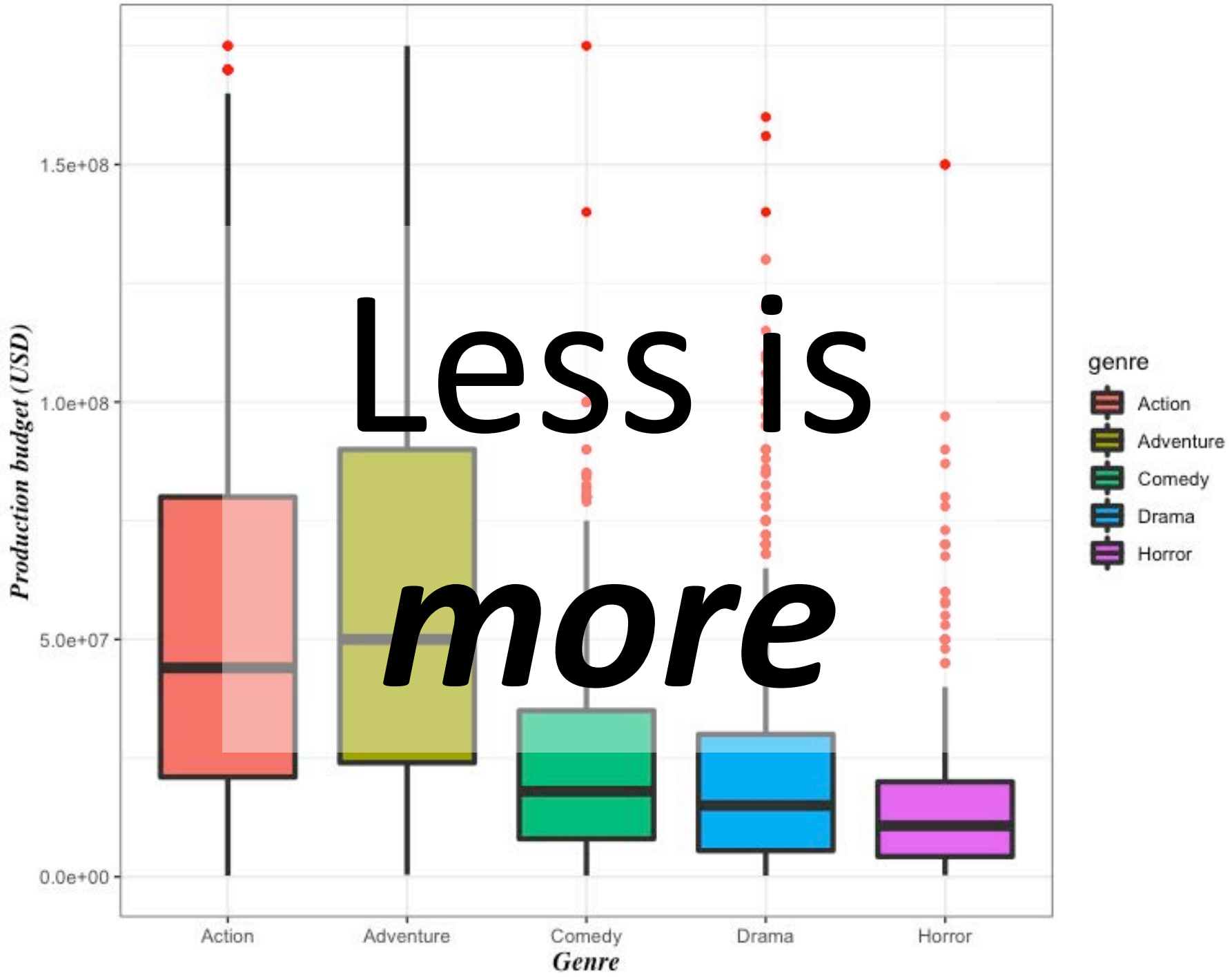
Colour the groups?

```
ggplot(data = movies, aes(x = genre, y = production_budget, fill = genre)) +
  geom_boxplot(outlier.colour = "red", size = 1.5) +
  ylab("Production budget (USD)") +
  xlab("Genre") +
  theme_bw() +
  theme(axis.title.x = element_text(family = "Times", face = "bold.italic", size = 12),
        axis.title.y = element_text(family = "Times", face = "bold.italic", size = 12))
```

Fine-tuning your geom_

Fine-tuning your plot:
Axes labels
Font type, size & style

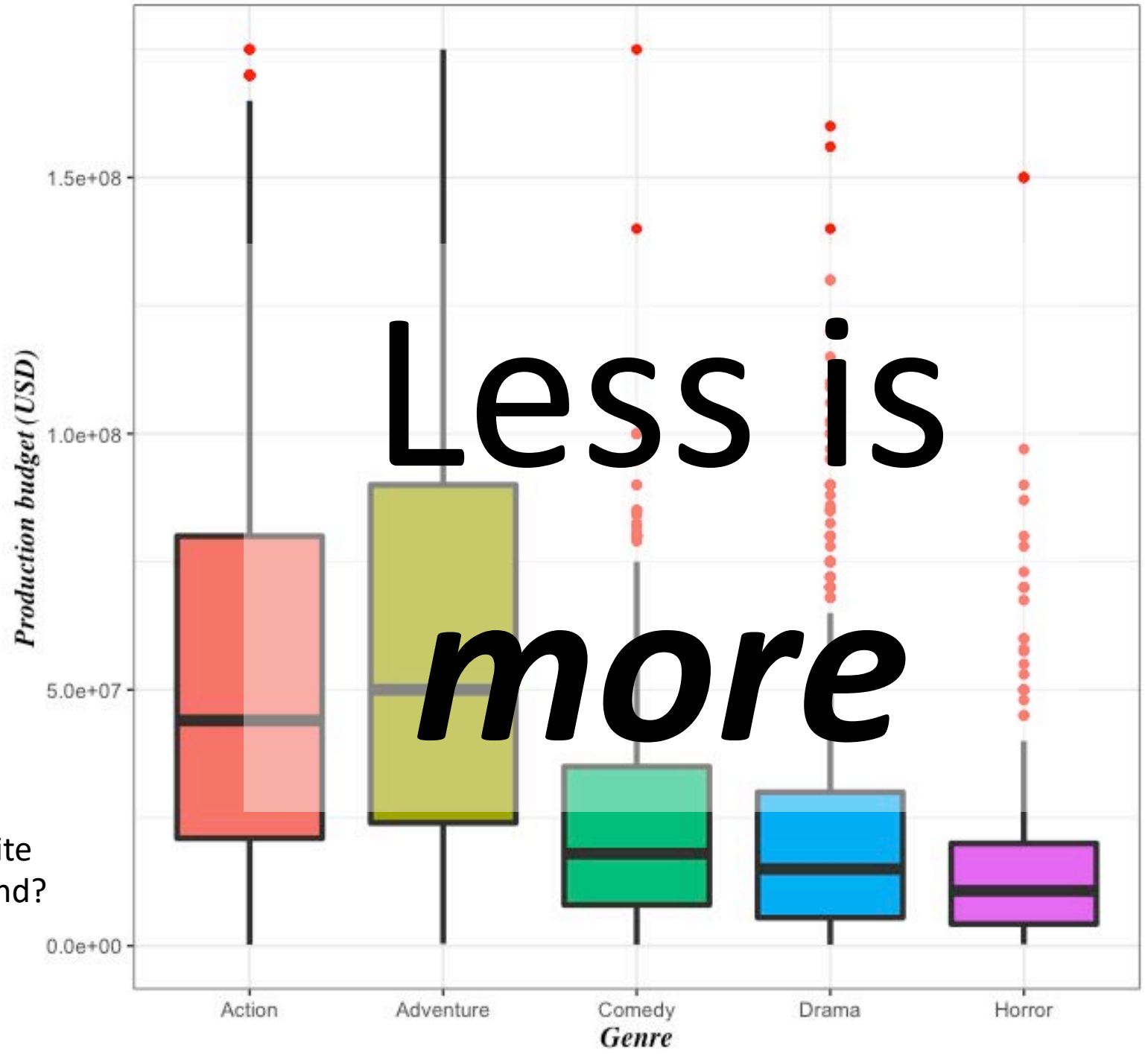

Production budget (USD)
1.5e+08
1.0e+08
5.0e+07
0.0e+00
Action
Adventure
Comedy
Drama
Horror
Genre
genre
Action
Adventure
Comedy
Drama
Horror


Less is
more
Production budget (USD)
1.5e+08
1.0e+08
5.0e+07
0.0e+00
Action
Adventure
Comedy
Drama
Horror
Genre
genre
Action
Adventure
Comedy
Drama
Horror

Is this necessary?

Who is the target audience?

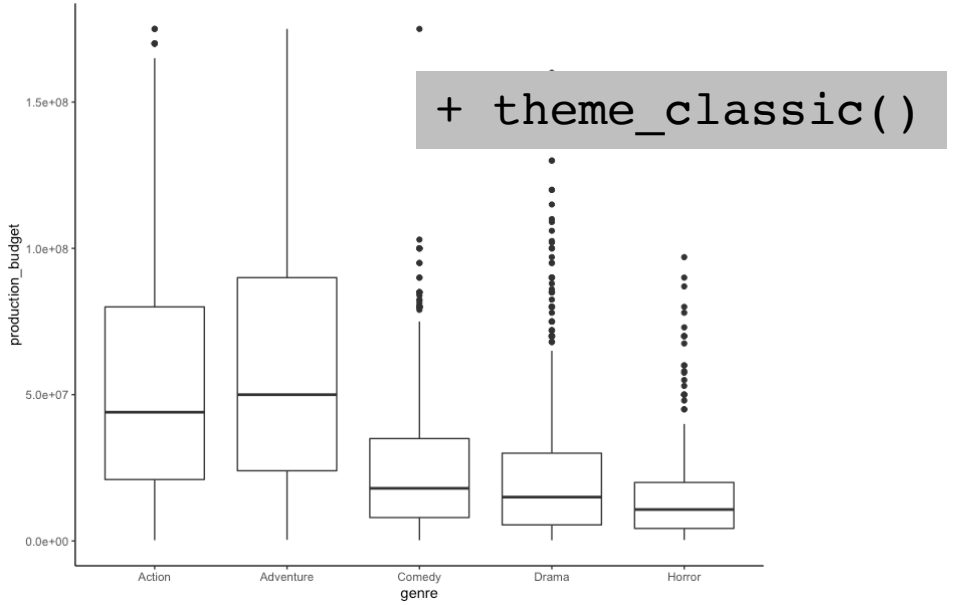
What would I write in the figure legend?

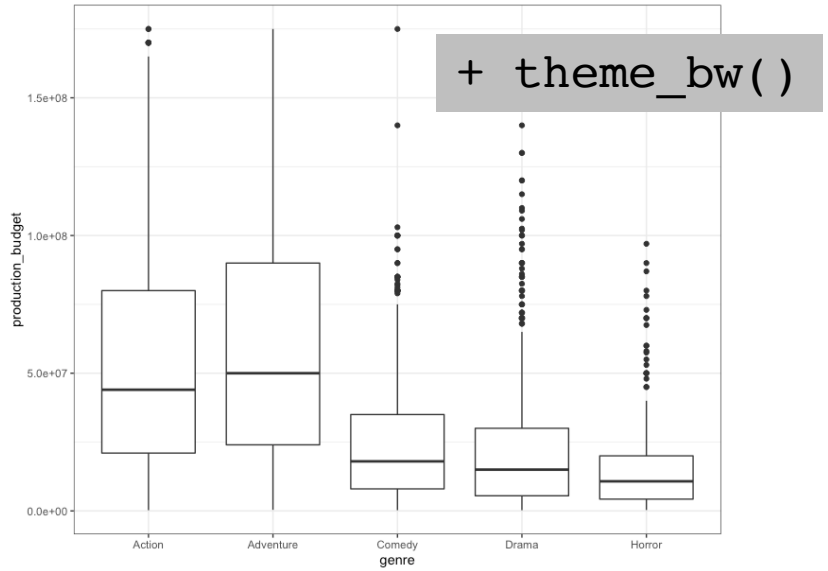


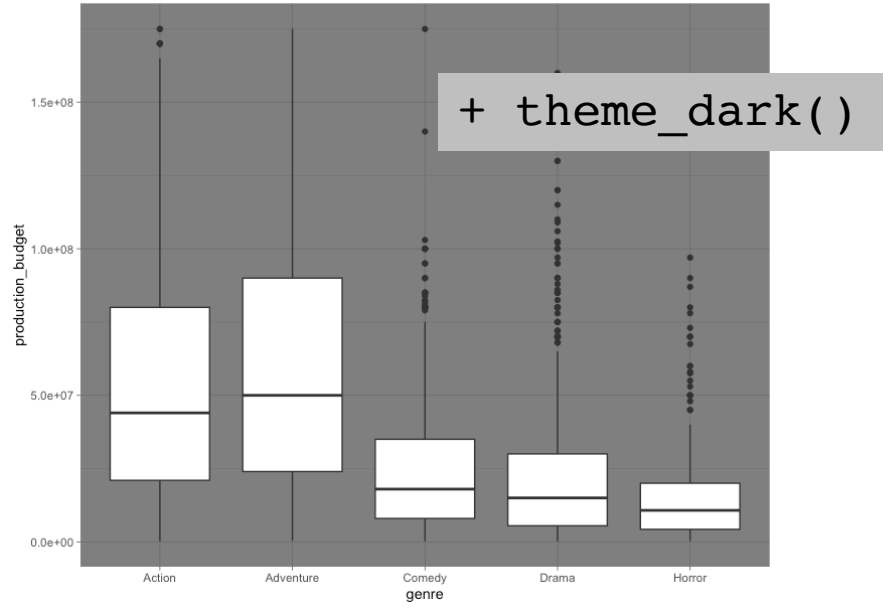
Am I procrastinating?

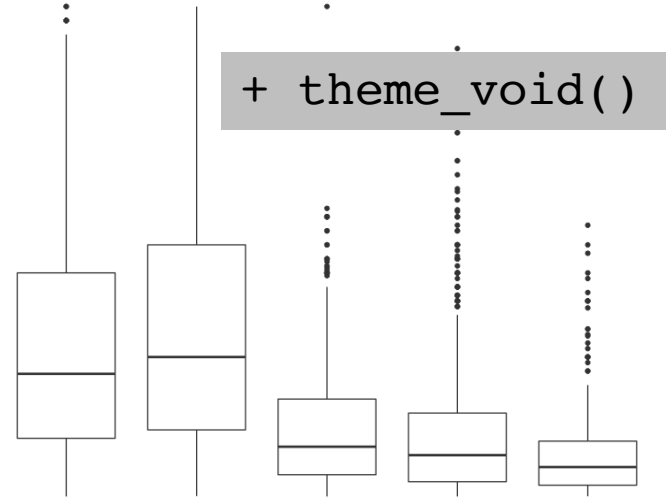
What is the best way to show data?

A quick look at themes







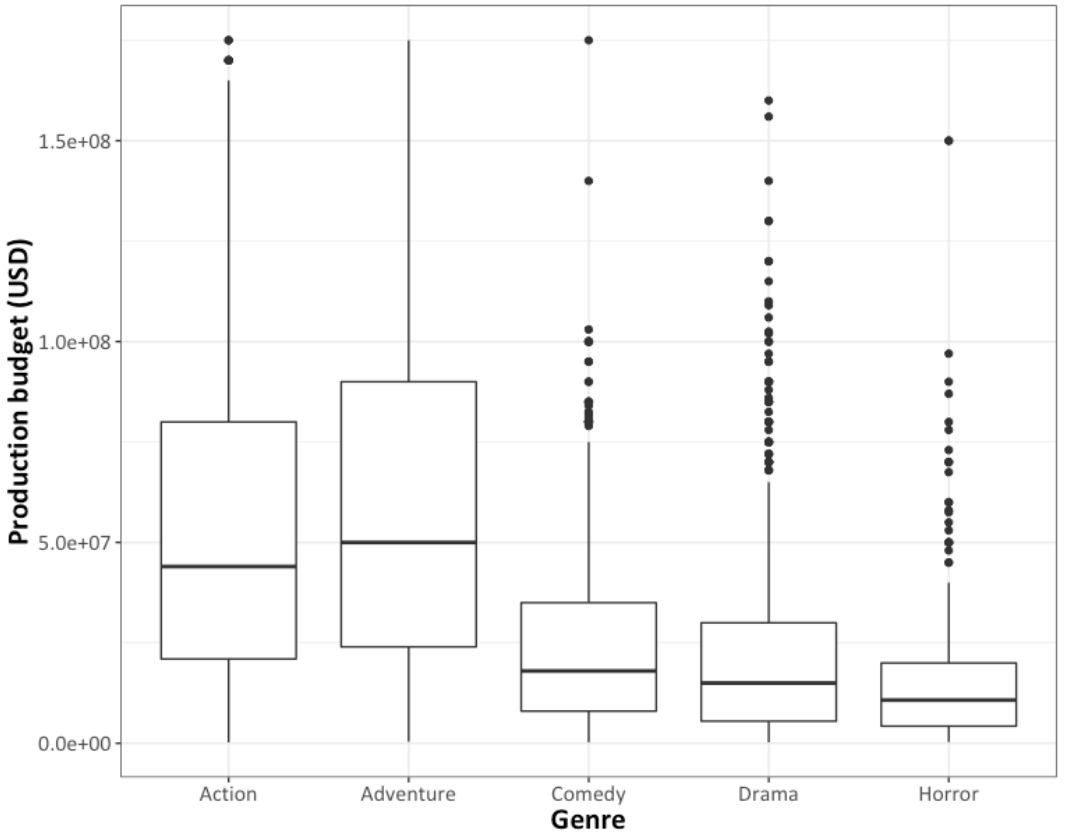


Do-it-yourself theme

```
thesis_theme <- theme_bw() +
  theme(axis.title.x = element_text(family = "Calibri", face = "bold", size = 16),
        axis.title.y = element_text(family = "Calibri", face = "bold", size = 16),
        axis.text.x = element_text(family = "Calibri" , size = 12),
        axis.text.y = element_text(family = "Calibri", size = 12),
        strip.text = element_text(family = "Calibri", size = 12))
```

```
ggplot(data = movies,
aes(x = genre,
    y = production_budget))
+ geom_boxplot()
+ ylab("Production budget (USD)")
+ xlab("Genre")
+ thesis_theme
```

Add this for all your plots from now on!

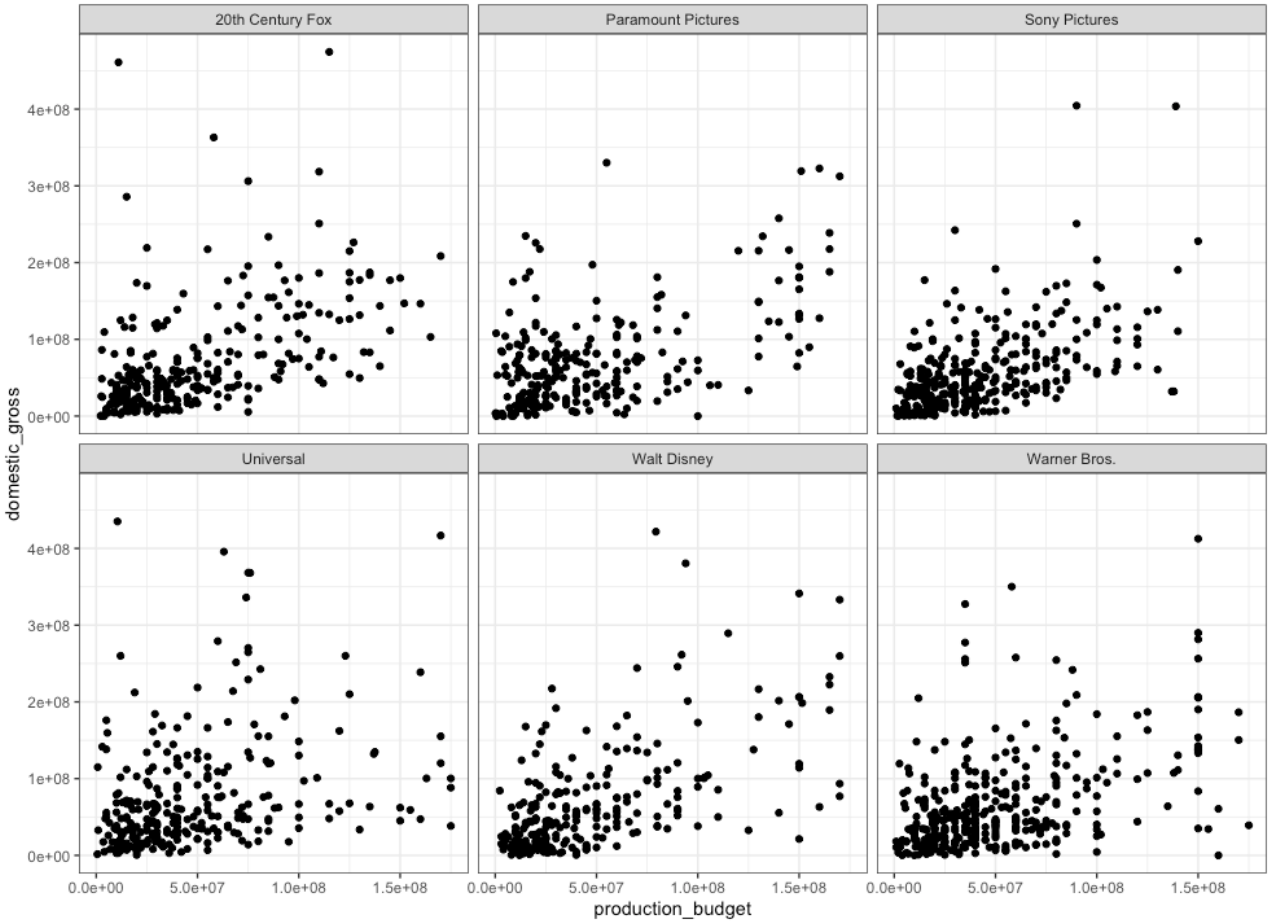


Facets! New favourite thing

Great for multivariate data (i.e. a factor and a continuous variable)

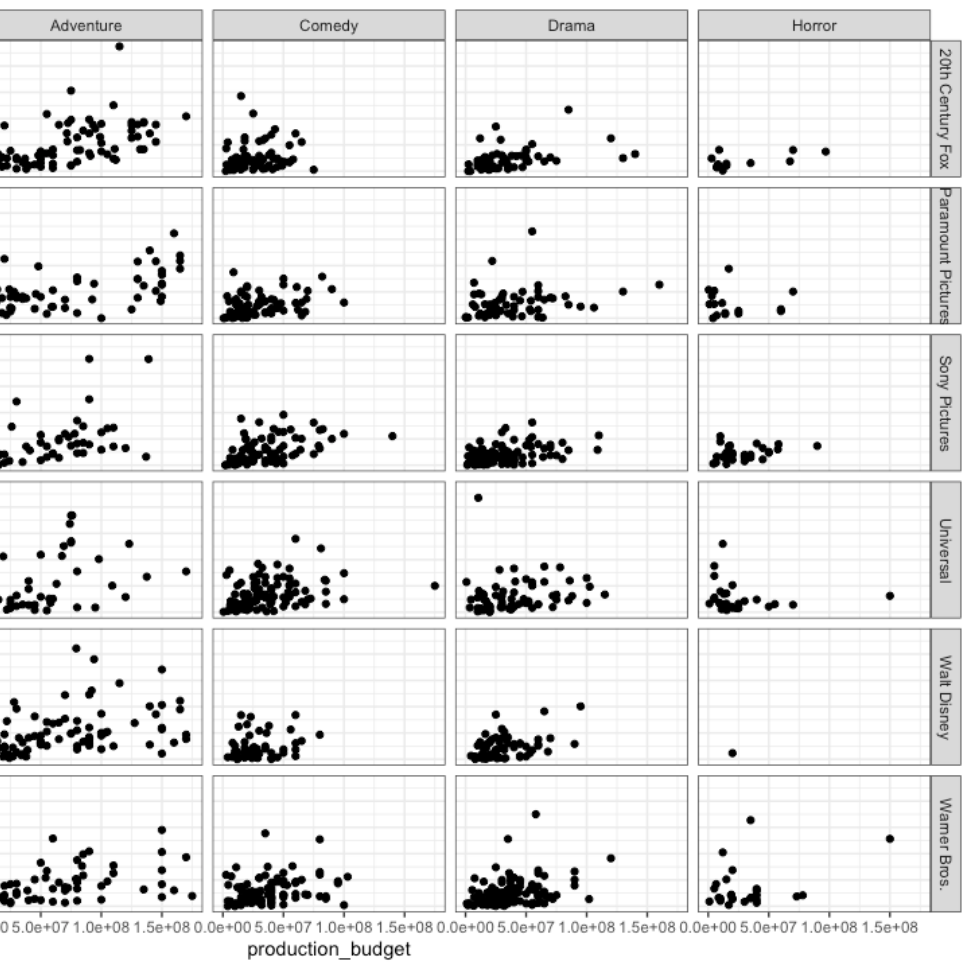
```
top6_plot <- movies %>%
filter(distributor %in% top_6 & !is.na(mpaa_rating)) %>%
  ggplot(data = ., aes(x = production_budget, y = domestic_gross)) +
  geom_point() +
  theme_bw()
```

```
top6_plot + facet_wrap(~distributor)
```



Facets! New favourite thing

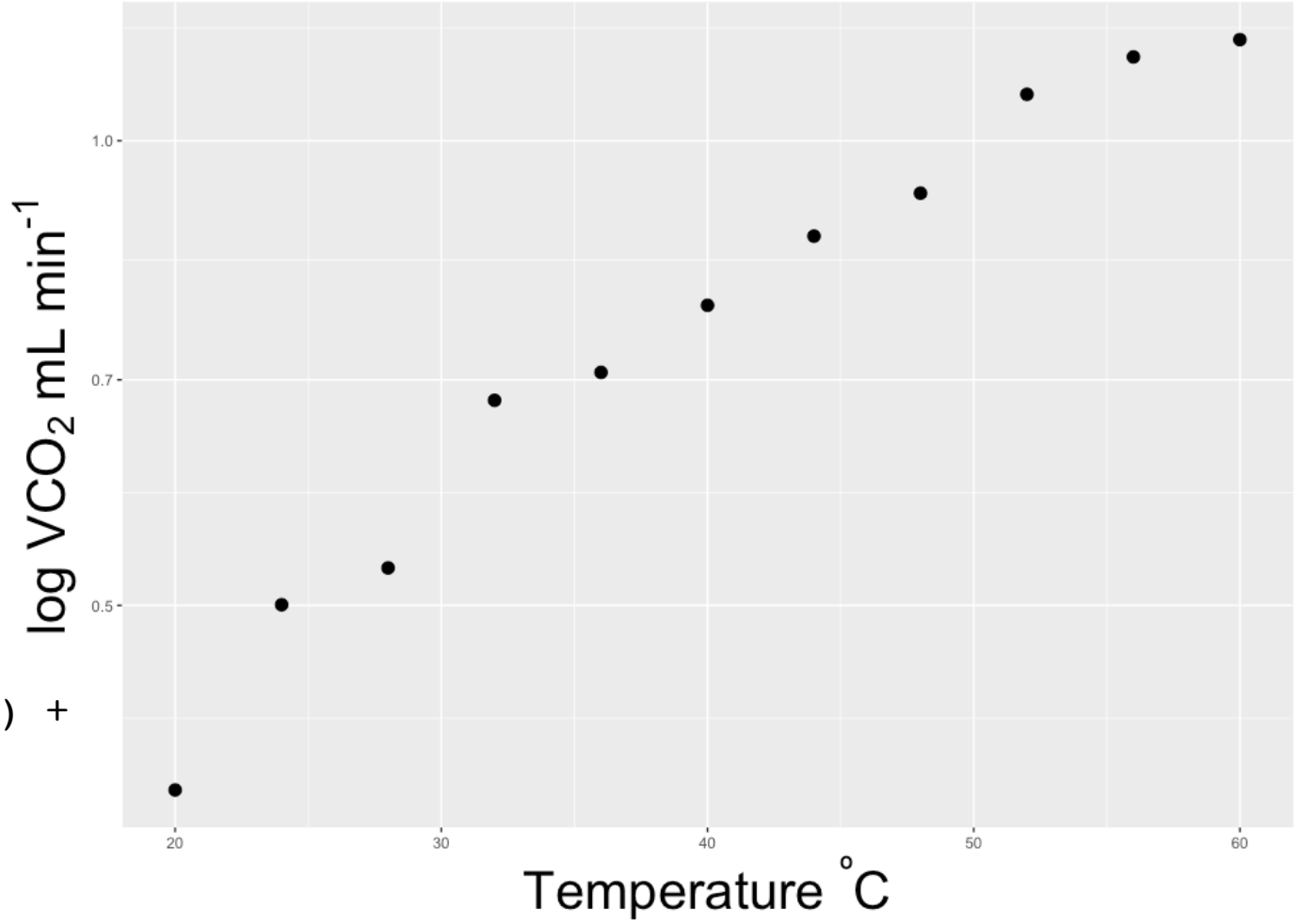
Great for multivariate data (i.e. a factor and a continuous variable)




```
top6_plot + facet_grid(distributor ~ genre)
```

Scientific/mathematic notations

- Reduces post-production edits
- Need to learn a bit of LaTeX
- HEAPS of help online
- library(latex2exp)




```
ggplot(fake_data,
aes(x = temperature, y = metabolic_rate)) +
  geom_point(size = 3) +
  scale_y_log10() +
  ylab(TeX("log VCO_2 mL min^{-1}")) +
  xlab(TeX('Temperature ^{º}C')) +
```

https://cran.r-project.org/web/packages/latex2exp/vignettes/using-latex2exp.html

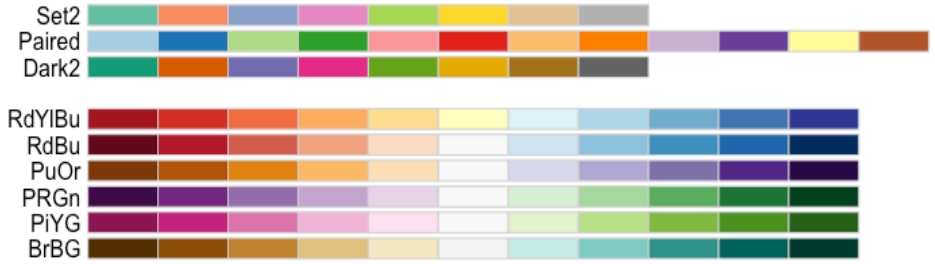
Working with palettes

Some quick ways to make your colours look cohesive

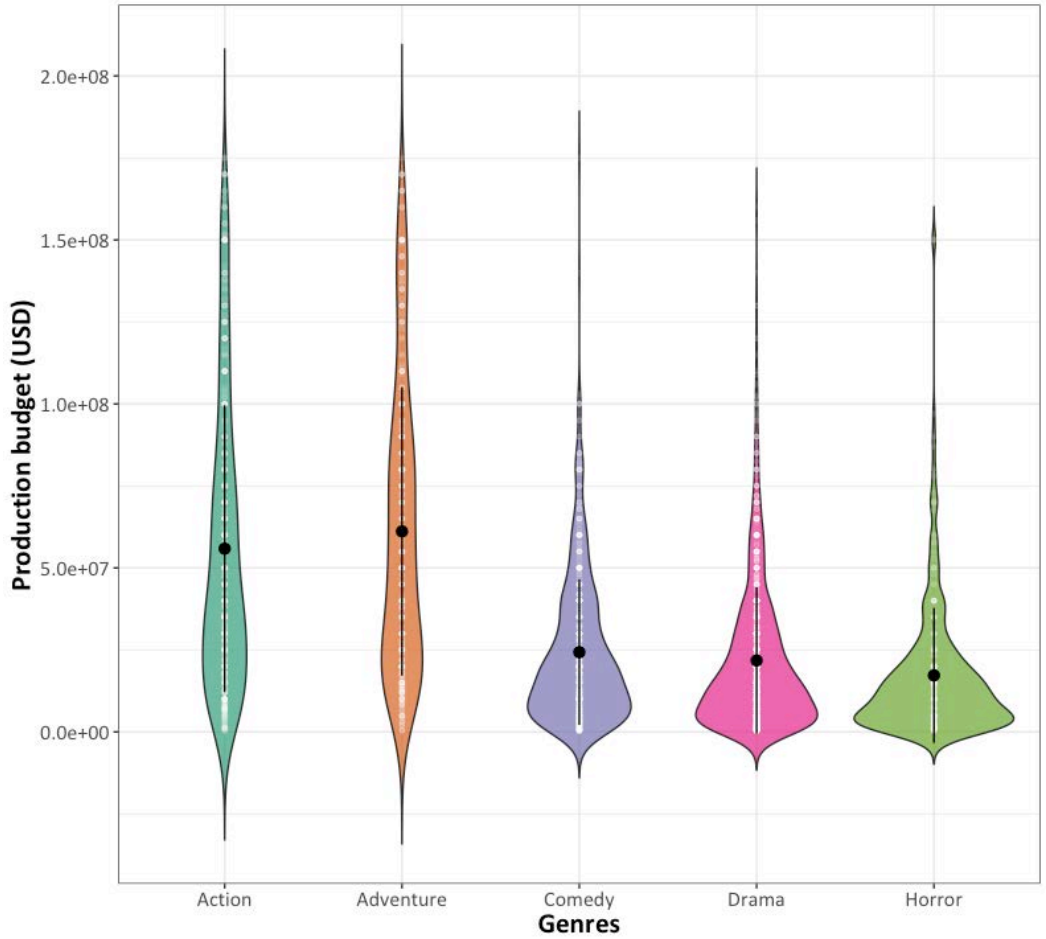
```
install.packages("RColourBrewer")

library(RColorBrewer)
```

Consider colour blindedness!

```
display.brewer.all(colorblindFriendly = T)
```




```
ggplot(data = movies,
aes(x = genre,
    y = production_budget, fill = genre))
+ geom_violin(trim = F, alpha = 0.7)
+ geom_point(colour = "white", size = 1, alpha = 0.1)
+ stat_summary(fun.data="data_summary", col = "black")
+ + scale_fill_brewer(palette = "Dark2")
```

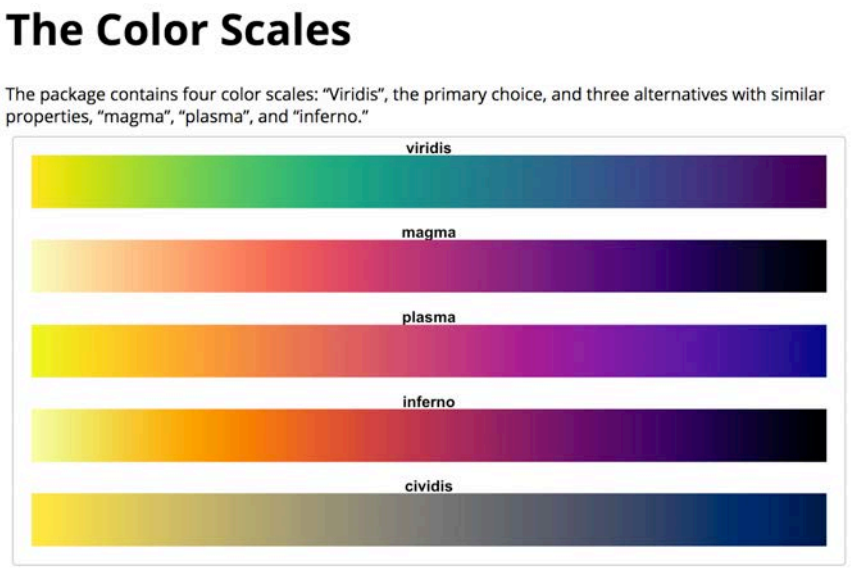


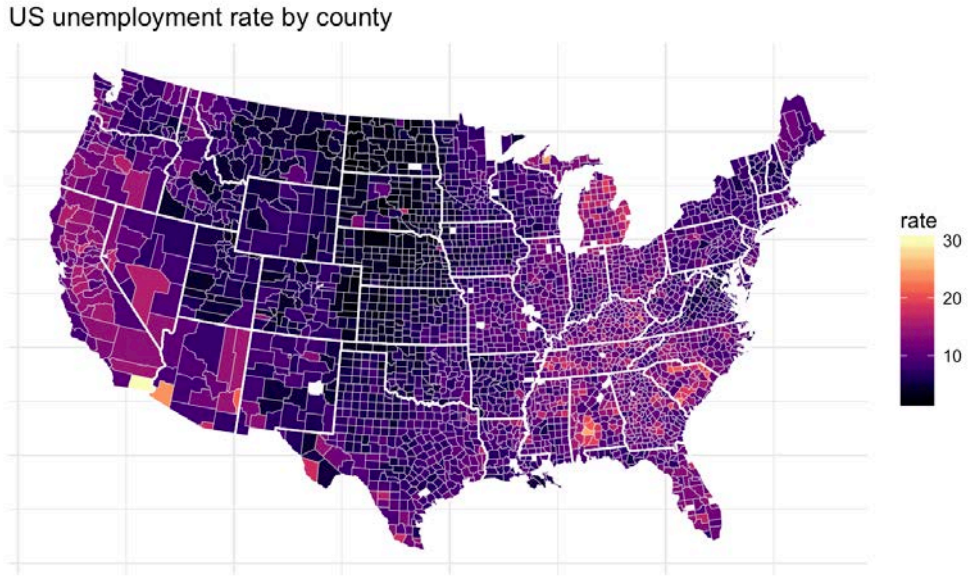
Working with palettes

Some quick ways to make your colours look cohesive

```
install.packages("viridisLite")

library(viridisLite)
```

Great for continuous data and spatial maps



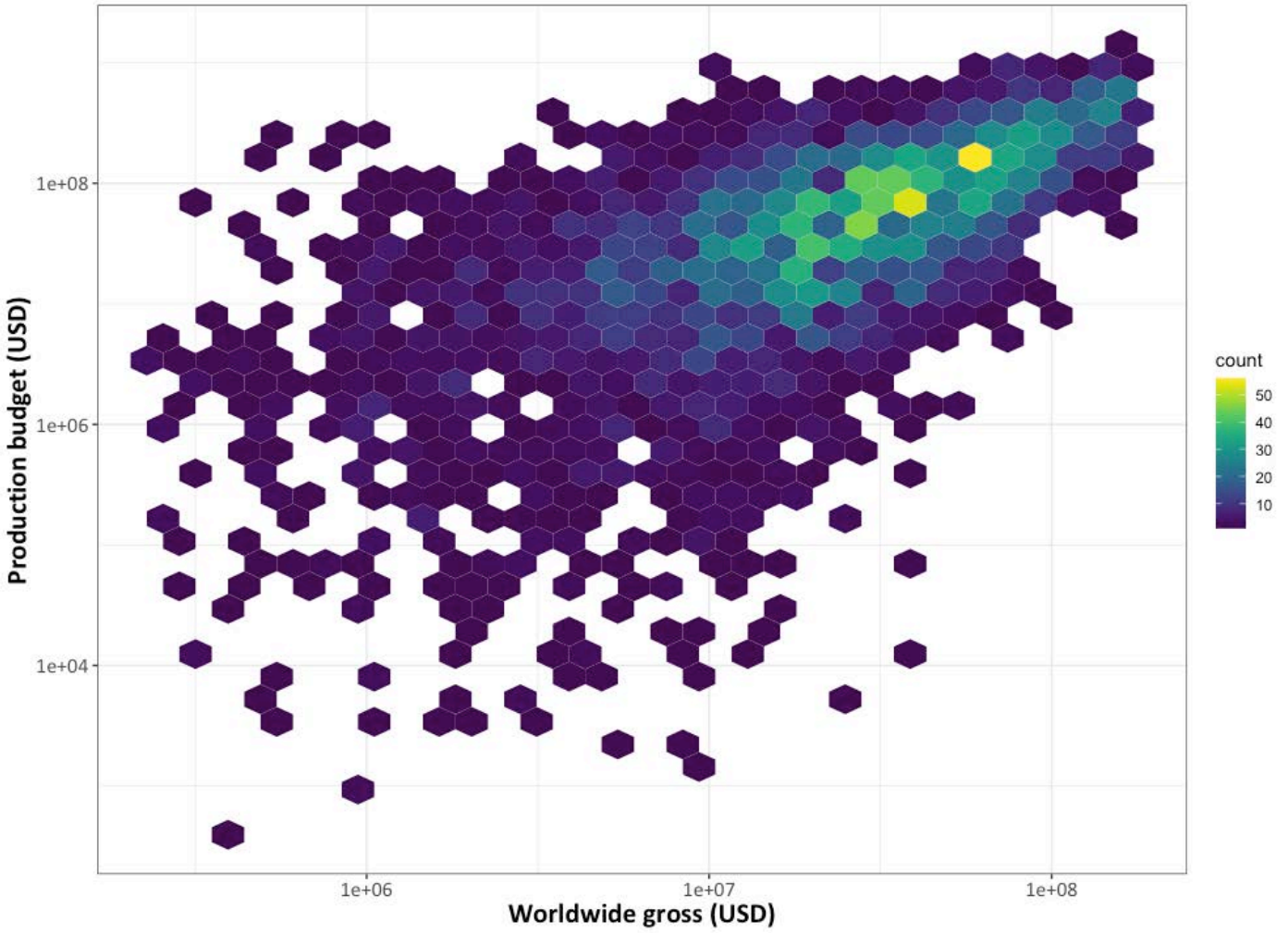


https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-viridis.html

Working with palettes

Some quick ways to make your colours look cohesive

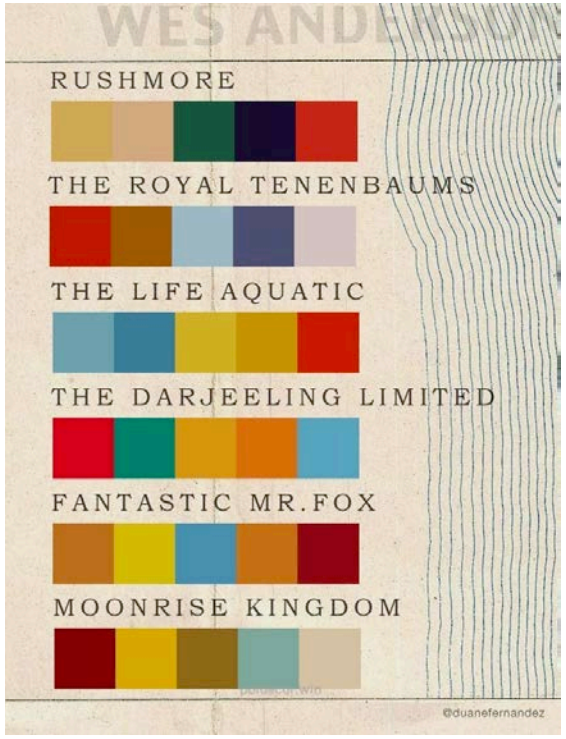
```
ggplot(data = movies,
aes(x = production_budget,
    y = worldwide_gross)) +
  geom_hex() +
  scale_x_log10() +
  scale_y_log10() +
  scale_fill_viridis() +
  ylab("Production budget (USD)") +
  xlab("Worldwide gross (USD)") +
  thesis_theme
```



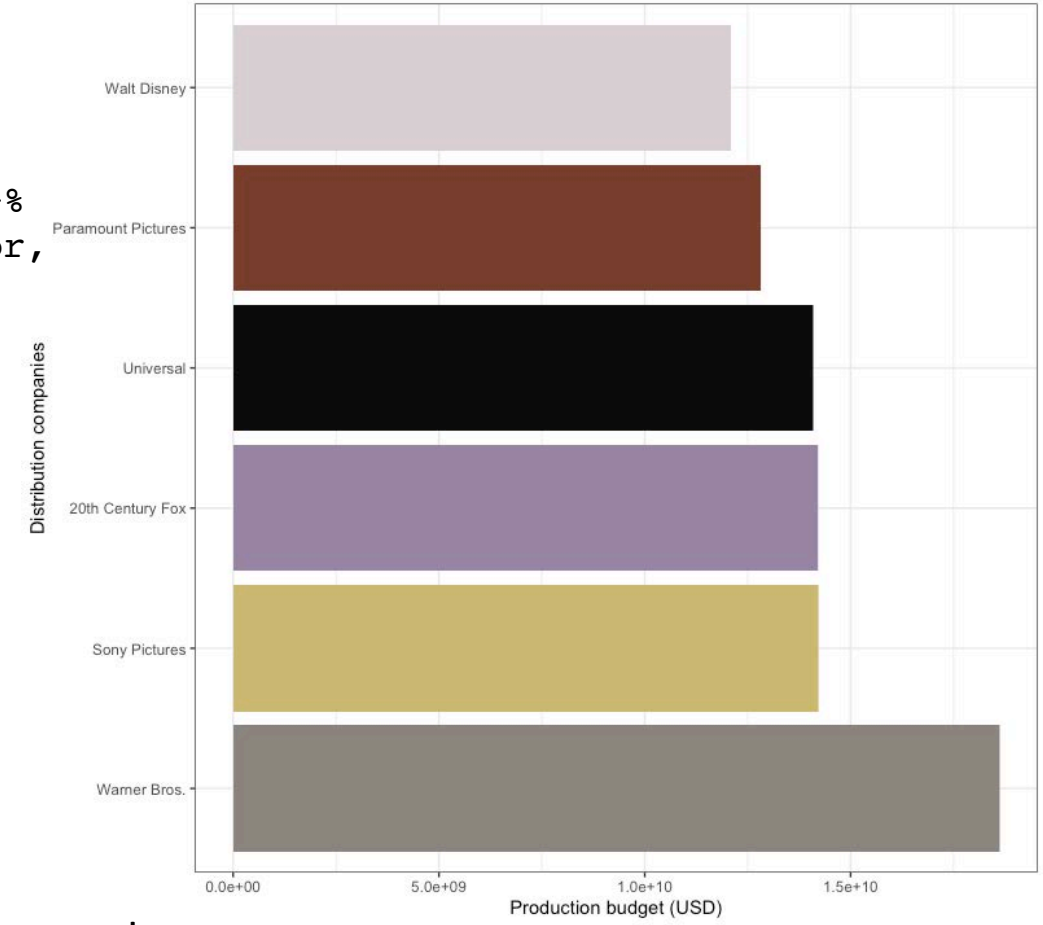
Working with palettes

Some quick ways to make your colours look cohesive

```
devtools::install_github("karthik/wesanderson")

library(wesanderson)
```




```
movies %>%
filter(distributor %in% top_6) %>%
ggplot(aes(x = reorder(distributor,
-production_budget, sum),
           y = production_budget,
fill = distributor))
+ geom_col()
+ scale_fill_manual(values =
wes_palette("IsleofDogs1"))
+ coord_flip()
```



https://wesandersonpalettes.tumblr.com/ https://github.com/karthik/wesanderson

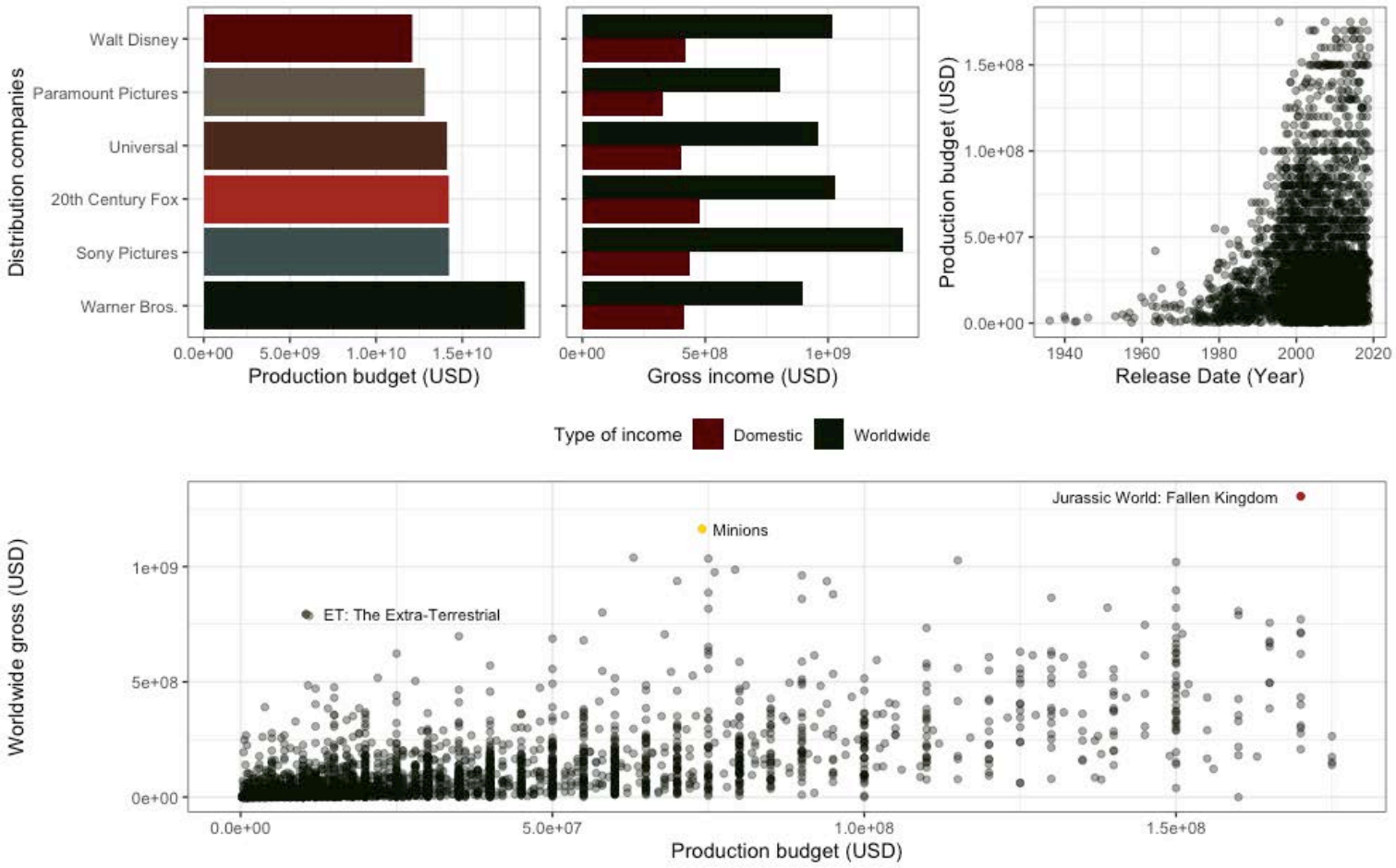
Panels for publications

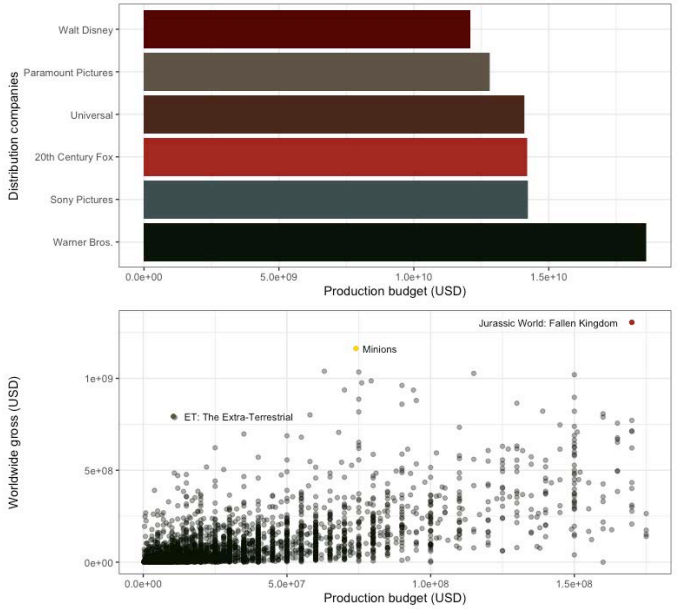
```
install_packages("patchwork")

library(patchwork)
```

1. Make multiple plots
2. Assign them to different objects
3. Arrange the objects

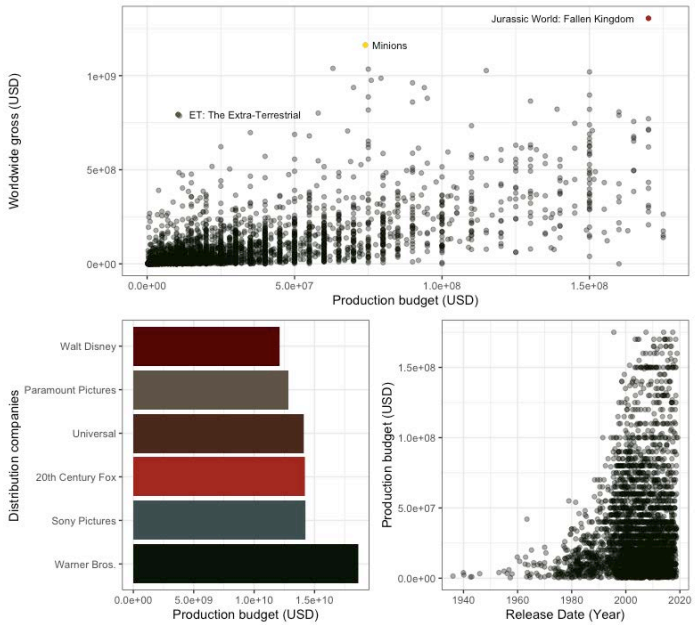
```
A <- ggplot(…) + geom_col()
B <- ggplot(…) + geom_col()
C <- ggplot(…) + geom_point()
D <- ggplot(…) + geom_point()

(A + B + C) / D
```

Less work later (no more post-production!)

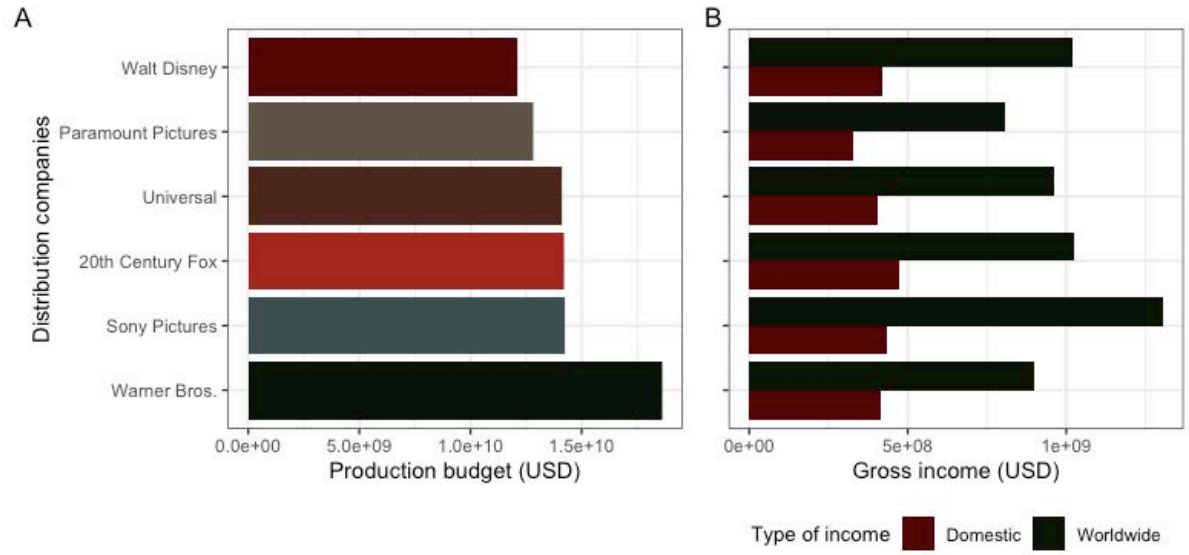


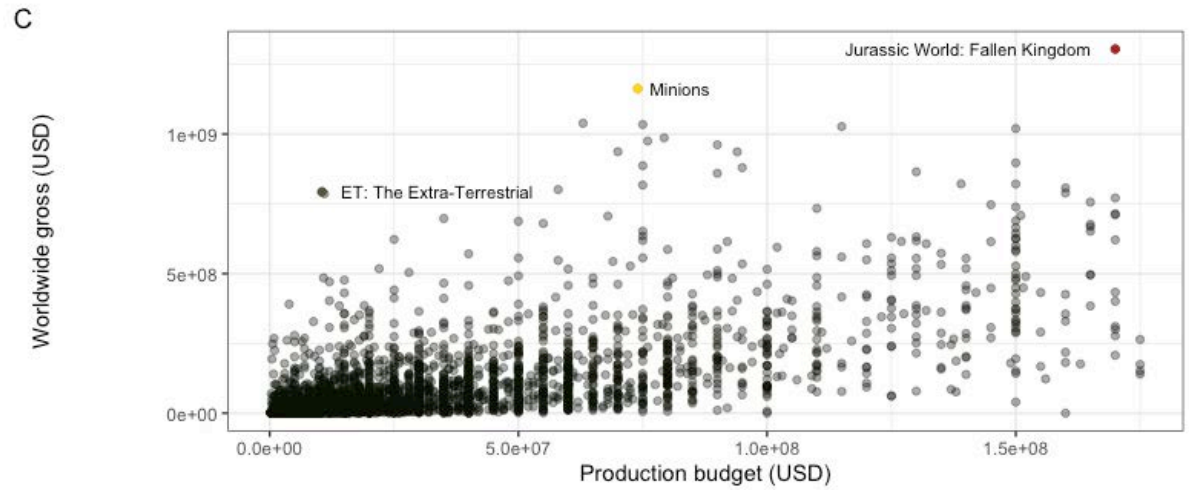


A / D



D / (A + C)



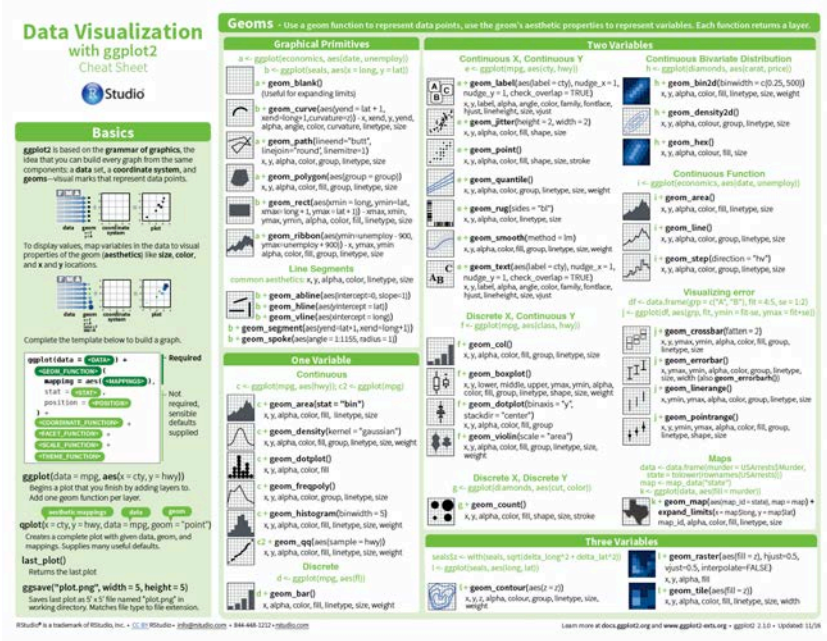


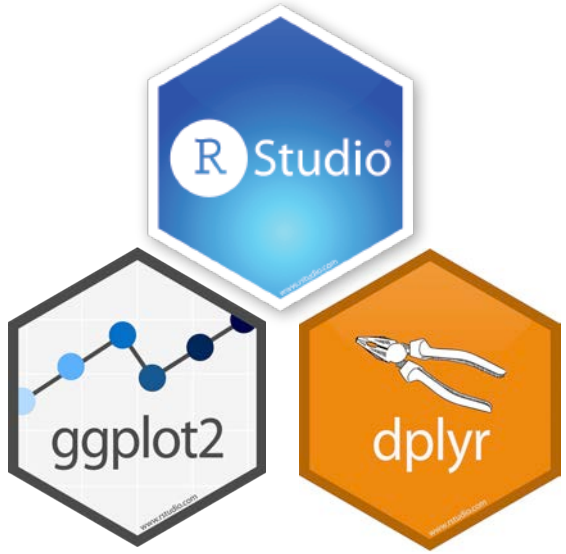

```
#Annotating
(A + B) / D + plot_annotation(tag_levels = 'A')
```

Final words…

- This is only the beginning
- LOTS of google-able help!
- Always think about what the **key message** you are trying to graphically represent
- Less is ***more***

http://environmentalcomputing.net/plotting-with-ggplot/





Environmental Computing