Data 550: Data Visualization I

Lecture 4b: Comparing Distributions in R

Dr. Irene Vrbik
University of British Columbia Okanagan

https://github.com/ubco-mds-2022/Data-550

Introduction

- Up until this point we have provided examples mostly in Altair with the understanding that ggplot has a similar counterpart.
- As Altair is relatively new, and ggplot2 is one of the most widely used and documented packages in R, it does have functionalities that Altair has yet to implement.
- One such example is violin plots.

https://github.com/ubco-mds-2022/Data-550

Learning Outcomes

- Create density, box plots, and violin plots using ggplot

Data

Below is the reprocessed movies data frame (to see how it was processed see the accompanying ipynb)

```r
# the above is the cleaned version
library(rjson)
library(tidyverse)
movies <- fromJSON(file = 'data/lec-movies.json') %>%
    as_tibble() %>%
    unnest(-c(countries, genres))

head(movies)
```

id <dbl>	title <chr>
12	Finding Nemo
22	Pirates of the Caribbean: The Curse of the Black Pearl

https://github.com/ubco-mds-2022/Data-550

id <dbl>	title <chr>
35	The Simpsons Movie
58	Pirates of the Caribbean: Dead Man's Chest
75	Mars Attacks!
117	The Untouchables

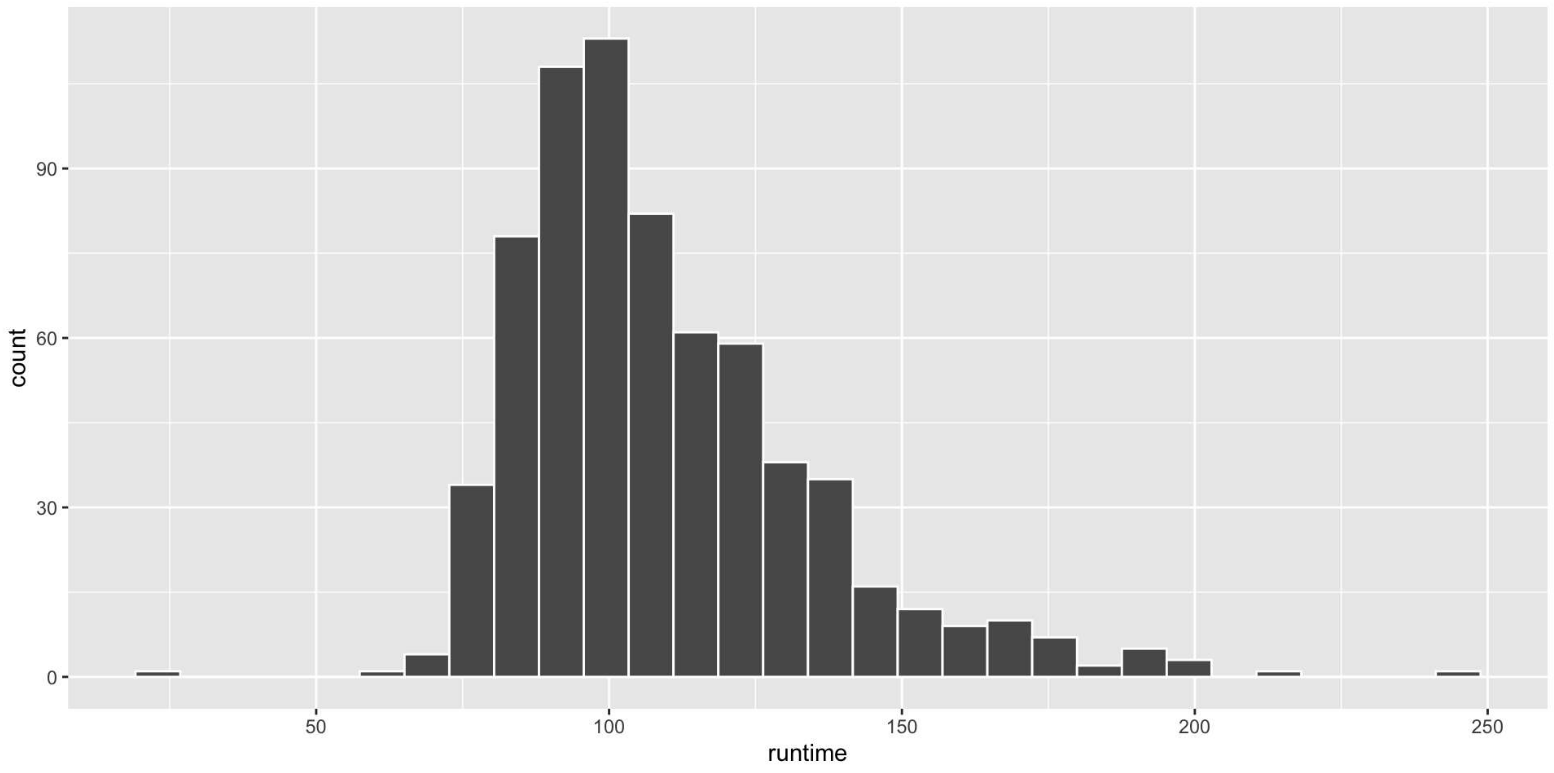
6 rows | 1-2 of 11 columns

https://github.com/ubco-mds-2022/Data-550

Histogram

Let’s recall how to make a histogram.

```r
ggplot(movies, aes(x = runtime)) +
    geom_histogram(color = 'white')
```

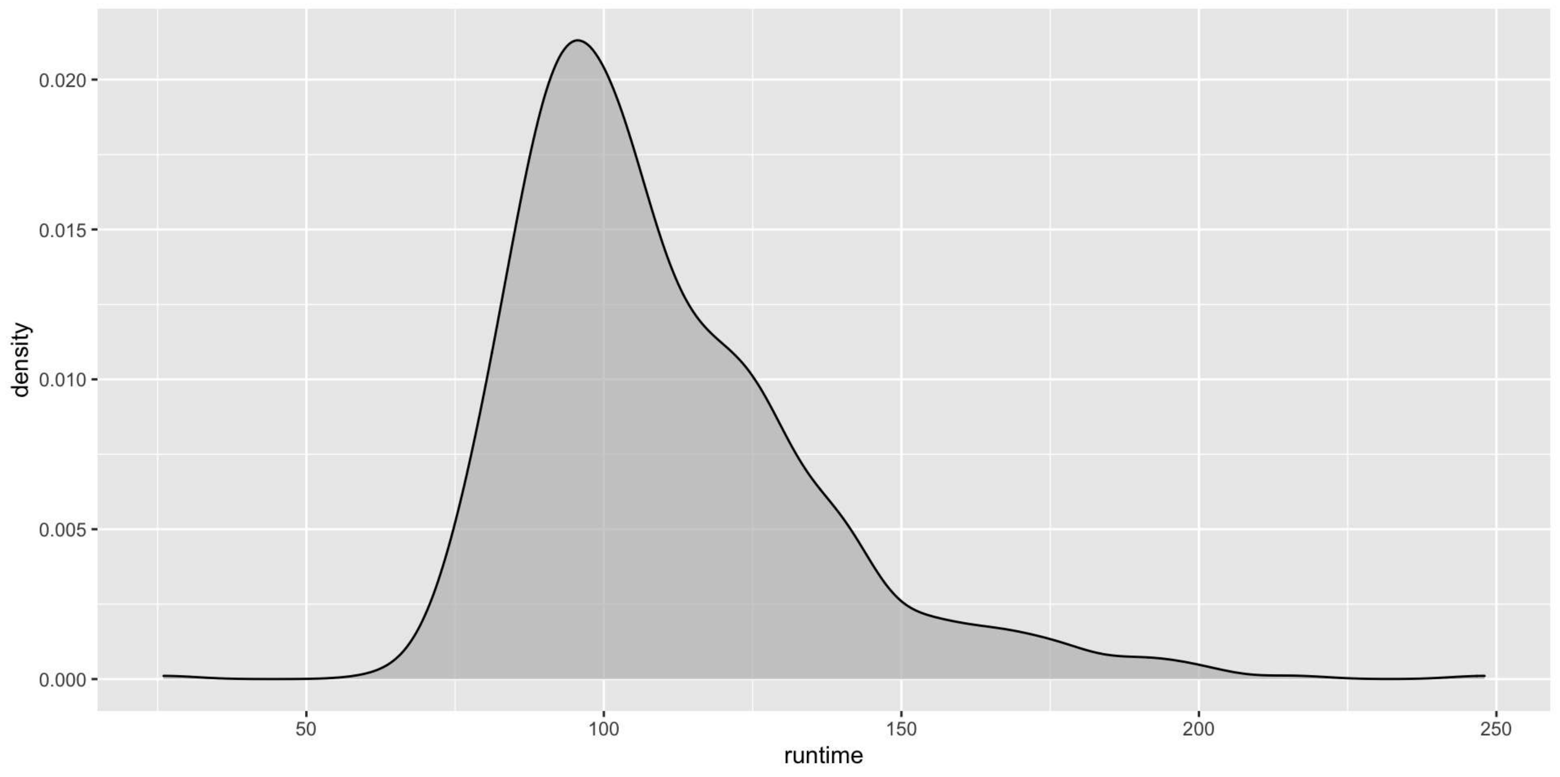


https://github.com/ubco-mds-2022/Data-550

Density plot

Unlike Altair, ggplot has it’s own density mark, …

```r
ggplot(movies, aes(x = runtime)) +
    geom_density(fill = 'grey', alpha = 0.7)
```



https://github.com/ubco-mds-2022/Data-550

Unnesting the data

We need to unnest/explode on `genres` and `countries`.

```
free_genres <- movies %>% unnest(genres)
free_countries <- movies  %>%  unnest(countries)
free_both <- movies %>% unnest(genres) %>%  unnest(countries)
```

```
free_genres %>%
  filter(, title ==  "All Dogs Go
  select(genres, countries)
```

genres <chr>	countries <named list>
Fantasy	<chr [2]>
Animation	<chr [2]>

2 rows

```
free_both %>%
  filter(title ==  "All Dogs Go to
  select(genres, countries)
```

genres <chr>	countries <chr>
Fantasy	United Kingdom
Fantasy	United States of America
Animation	United Kingdom
Animation	United States of America

4 rows

https://github.com/ubco-mds-2022/Data-550

Layered Density Plot

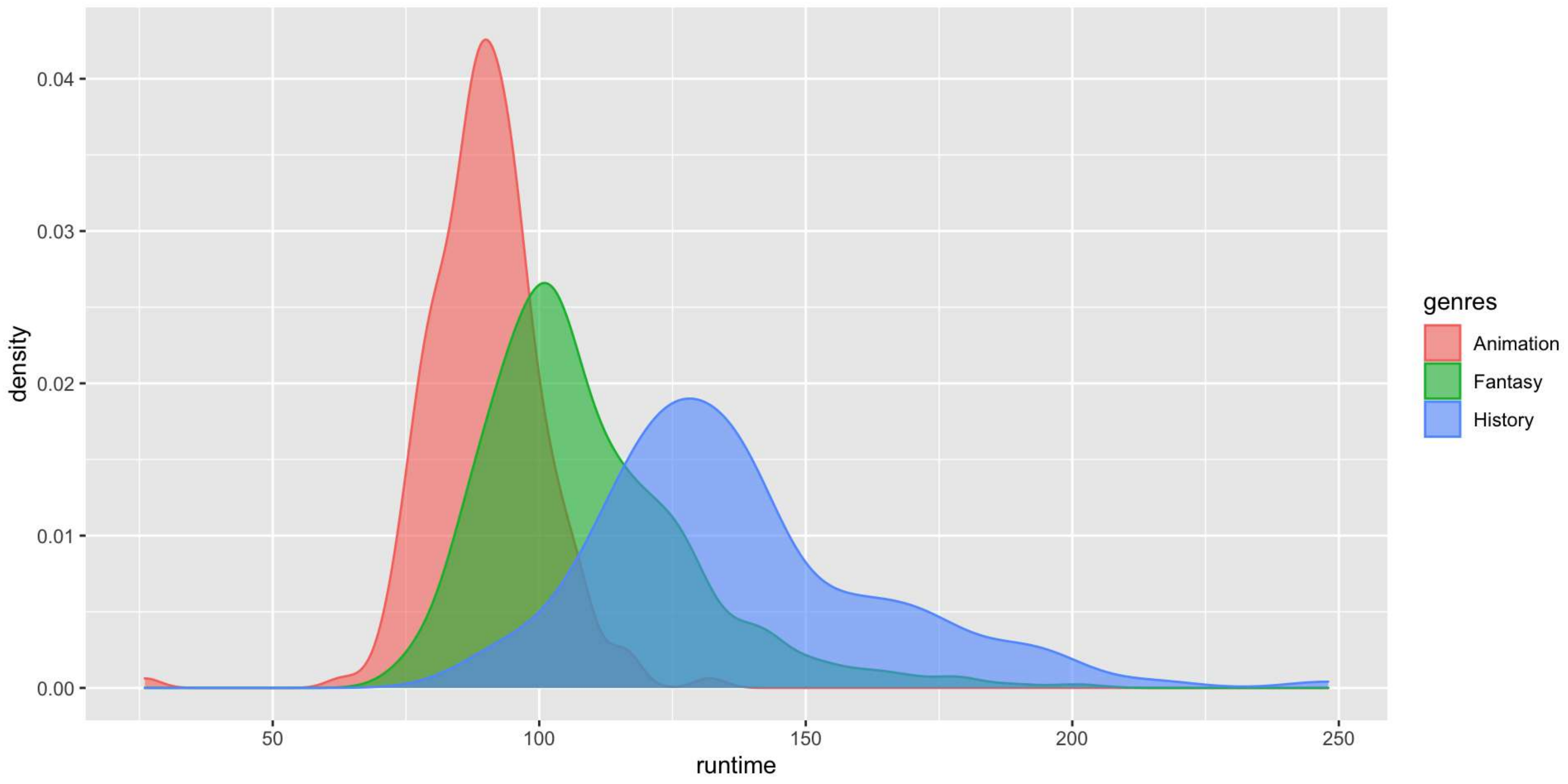
```r
ggplot(free_genres, aes(x = runtime,
        fill = genres,
        color = genres)) +
    geom_density(alpha = 0.6)
```

Notice how you can add the aesthetic rather than including it as an argument within `ggplot()`:

```r
ggplot(free_genres) +
    aes(x = runtime,
        fill = genres,
        color = genres) +
    geom_density(alpha = 0.6)
```

https://github.com/ubco-mds-2022/Data-550

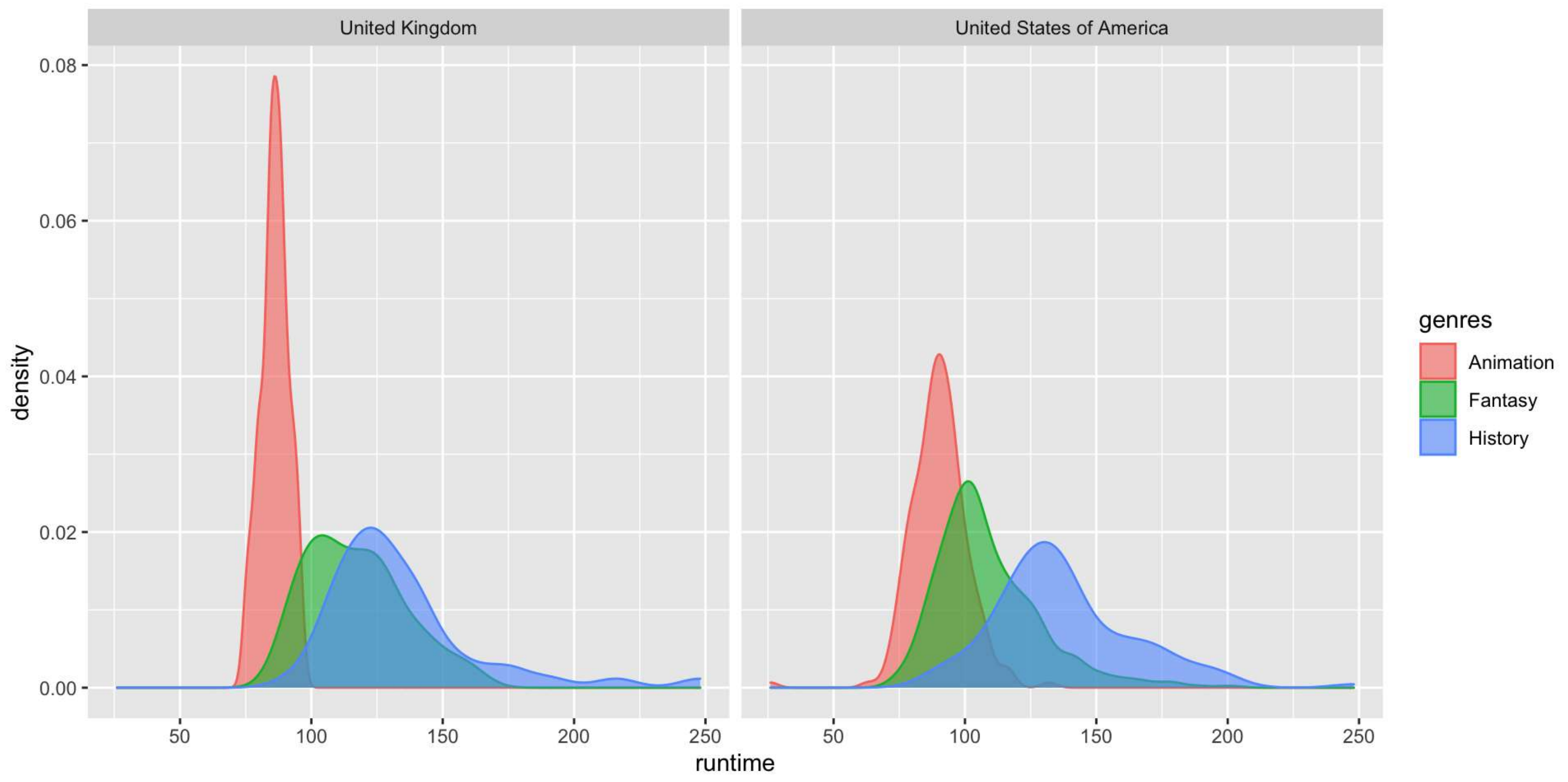
Layered Density Plot



https://github.com/ubco-mds-2022/Data-550

Faceting

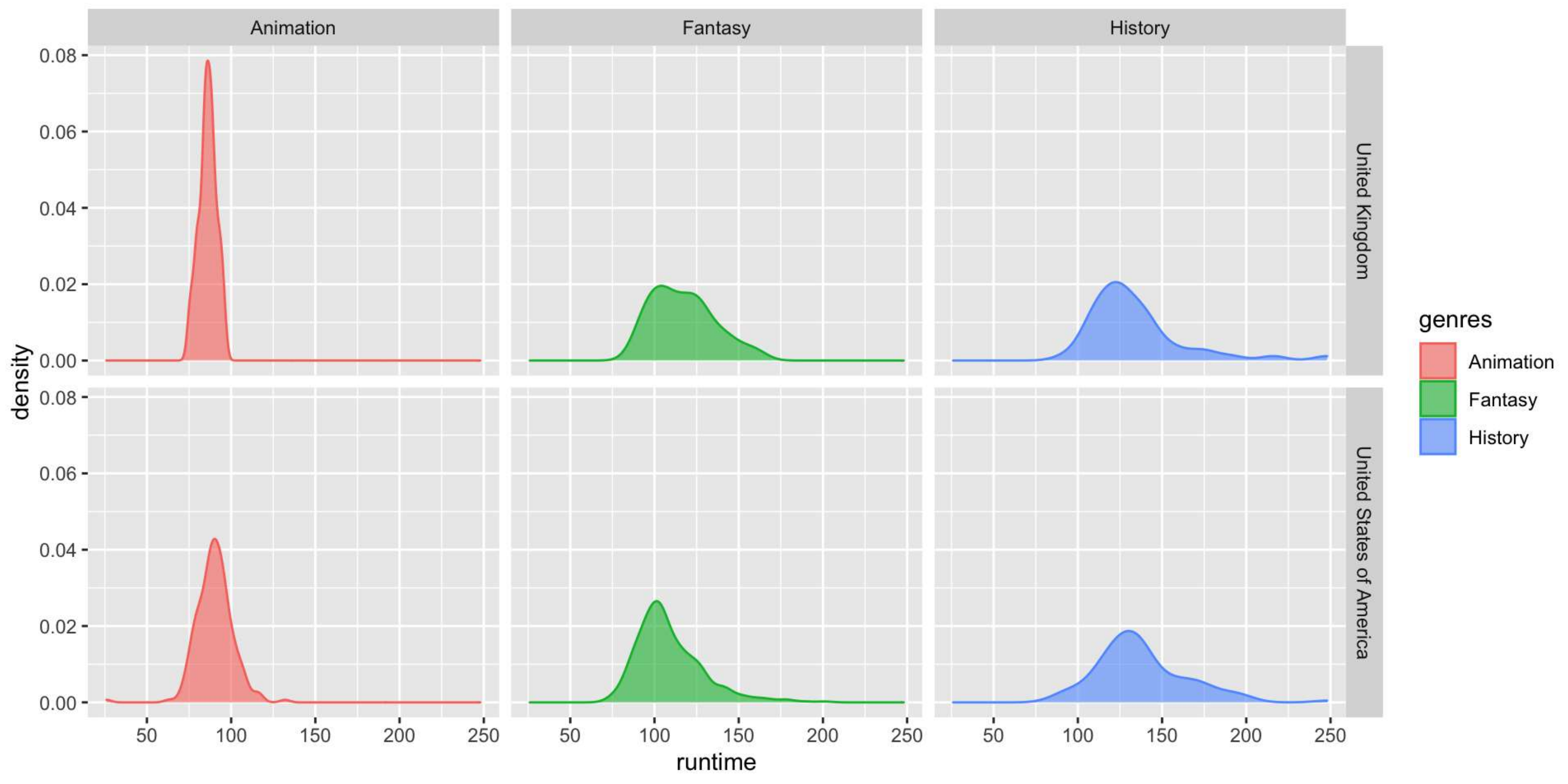
```r
ggplot(free_both) +
    aes(x = runtime, fill = genres, color = genres) +
    geom_density(alpha = 0.6) +
    facet_wrap(~countries)
```



https://github.com/ubco-mds-2022/Data-550

Faceting (row and column)

```r
ggplot(free_both, show.legend = FALSE) +
    aes(x = runtime, fill = genres, color = genres) +
    geom_density(alpha = 0.6) +
    facet_grid(countries~genres)
```

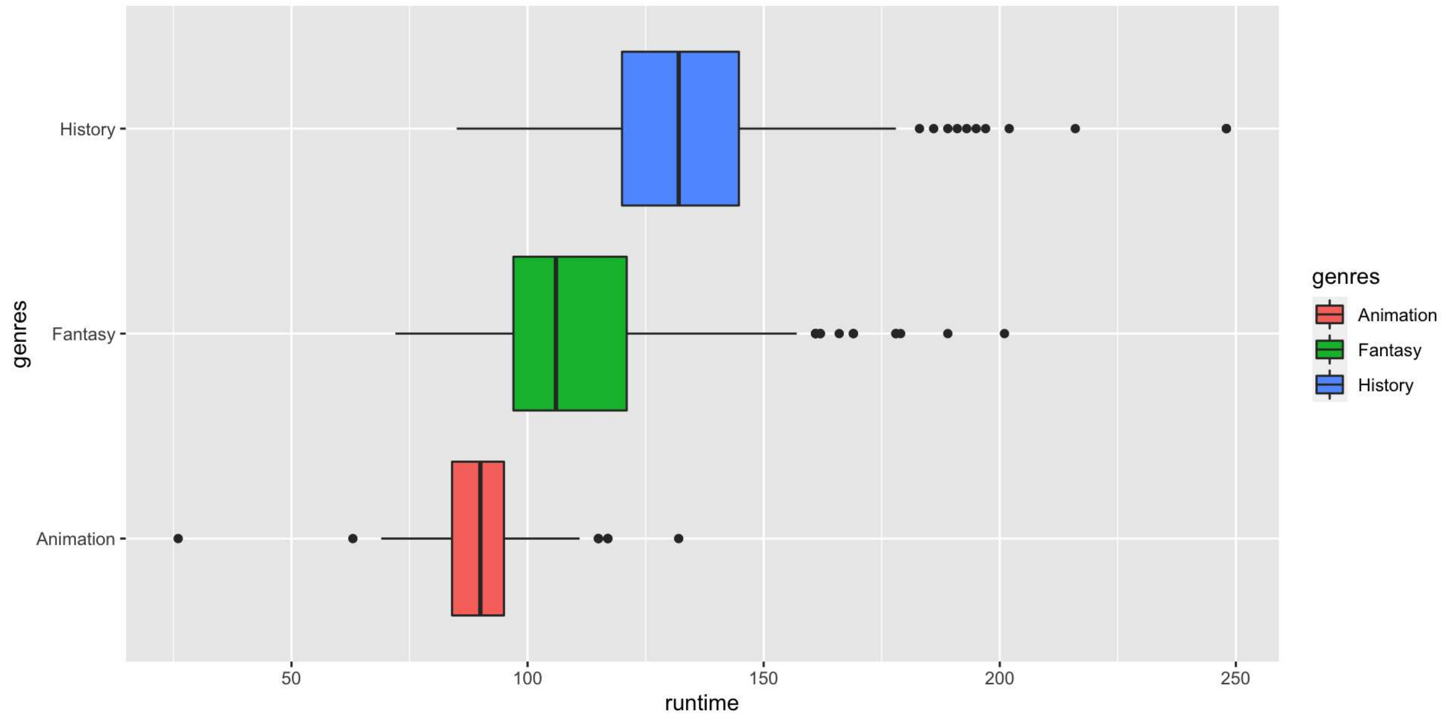


https://github.com/ubco-mds-2022/Data-550

Boxplots

As in Altair, ggplot unsuprisingly has a boxplot geom, eg.

```r
ggplot(free_both) +
    aes(x = runtime, y = genres,fill = genres) +
    geom_boxplot()
```

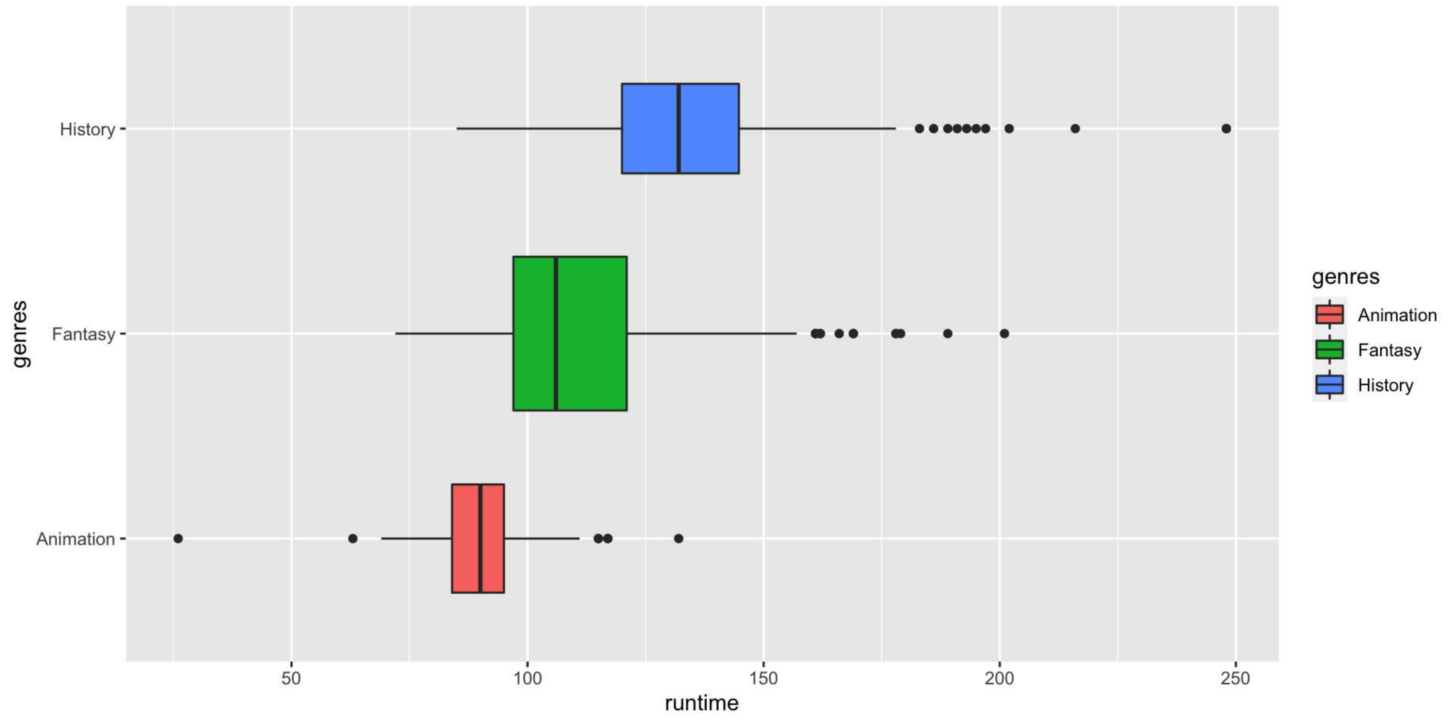


https://github.com/ubco-mds-2022/Data-550

Scaled Boxplots

As in Altair, ggplot unsuprisingly has a boxplot geom, eg.

```r
ggplot(free_both) +
    aes(x = runtime, y = genres,fill = genres) +
    geom_boxplot(varwidth = TRUE)
```



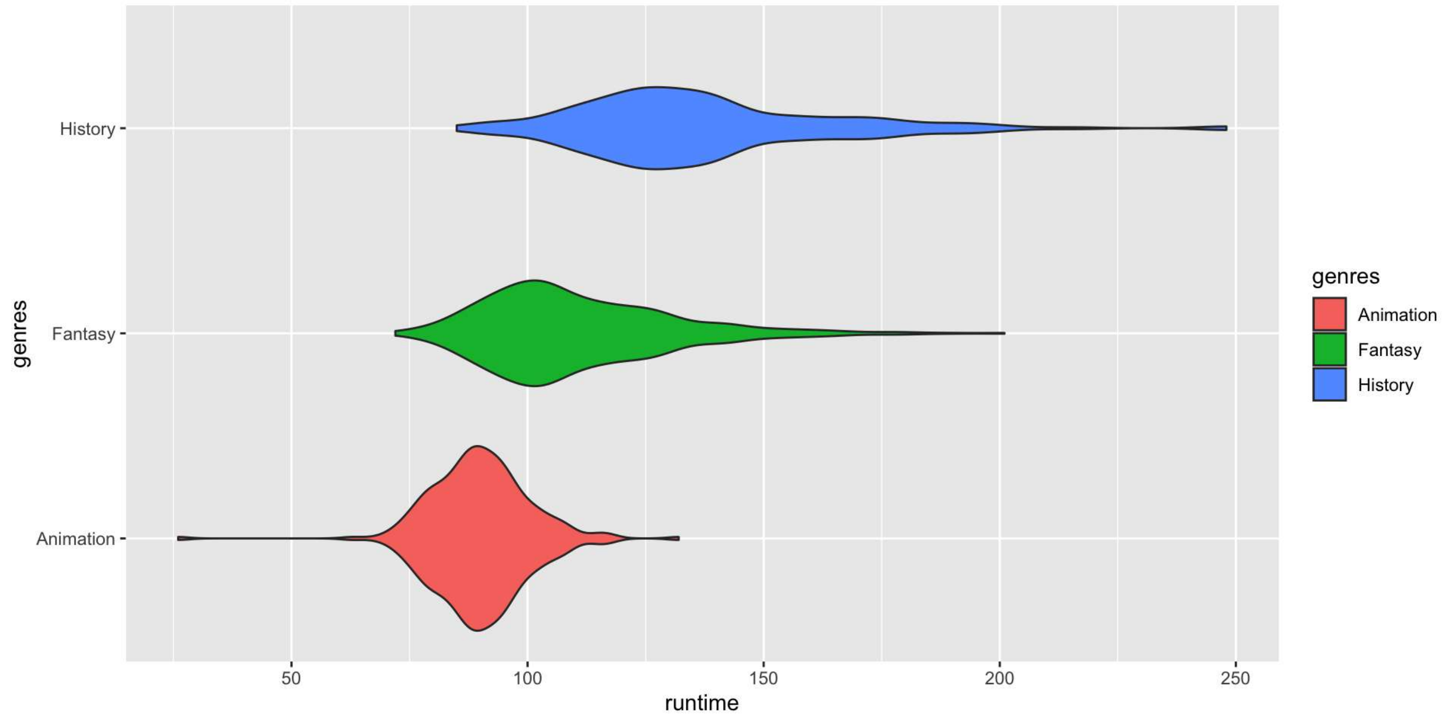
https://github.com/ubco-mds-2022/Data-550

13

Violin Plots

The change from boxplot to violin is extremely simple

```r
ggplot(free_both) +
    aes(x = runtime, y = genres,fill = genres) +
    geom_violin()
```



https://github.com/ubco-mds-2022/Data-550

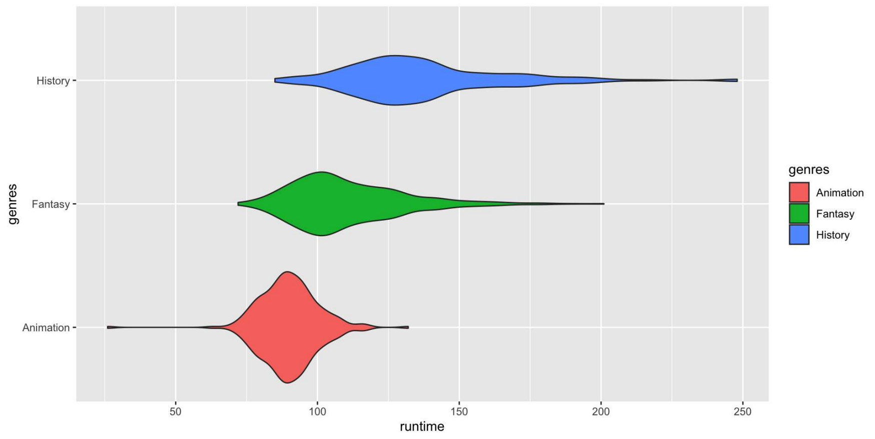
What are violin plots

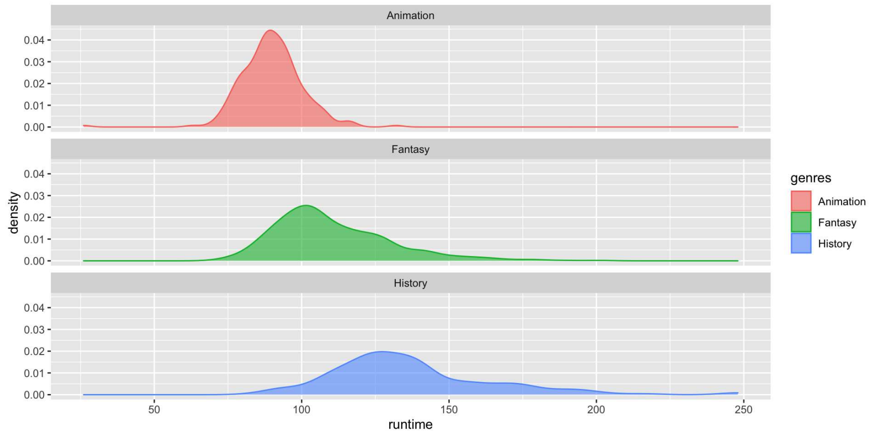
- Violin plots are similar to box plots, except that they also show the kernel probability density of the data at different values.
- Typically, violin plots will include a marker for the median of the data and a box indicating the interquartile range, as in standard box plots.
- The function `geom_violin()` is used to produce a violin plot.

Violin vs Faceted Density Plots

```r
ggplot(free_both) +
    aes(x = runtime,
        y = genres,
        fill = genres) +
    geom_violin()
```

```r
ggplot(free_both) +
    aes(x = runtime, fill = genres
    geom_density(alpha = 0.6) +
    facet_wrap(~genres, ncol = 1)
```



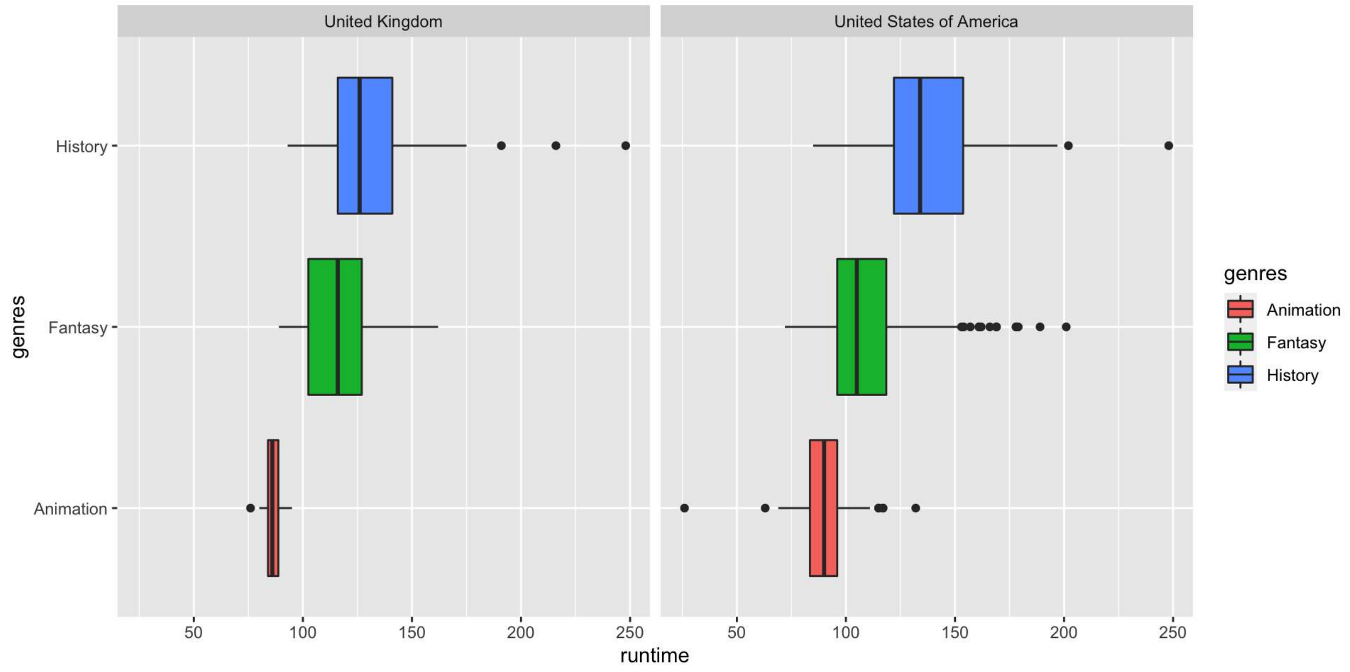


https://github.com/ubco-mds-2022/Data-550

Faceted Boxplots

As with out density plots, we can also facet by country, eg.

```r
ggplot(free_both) +
    aes(x = runtime, y = genres,fill = genres) +
    geom_boxplot() +
    facet_wrap(~countries)
```

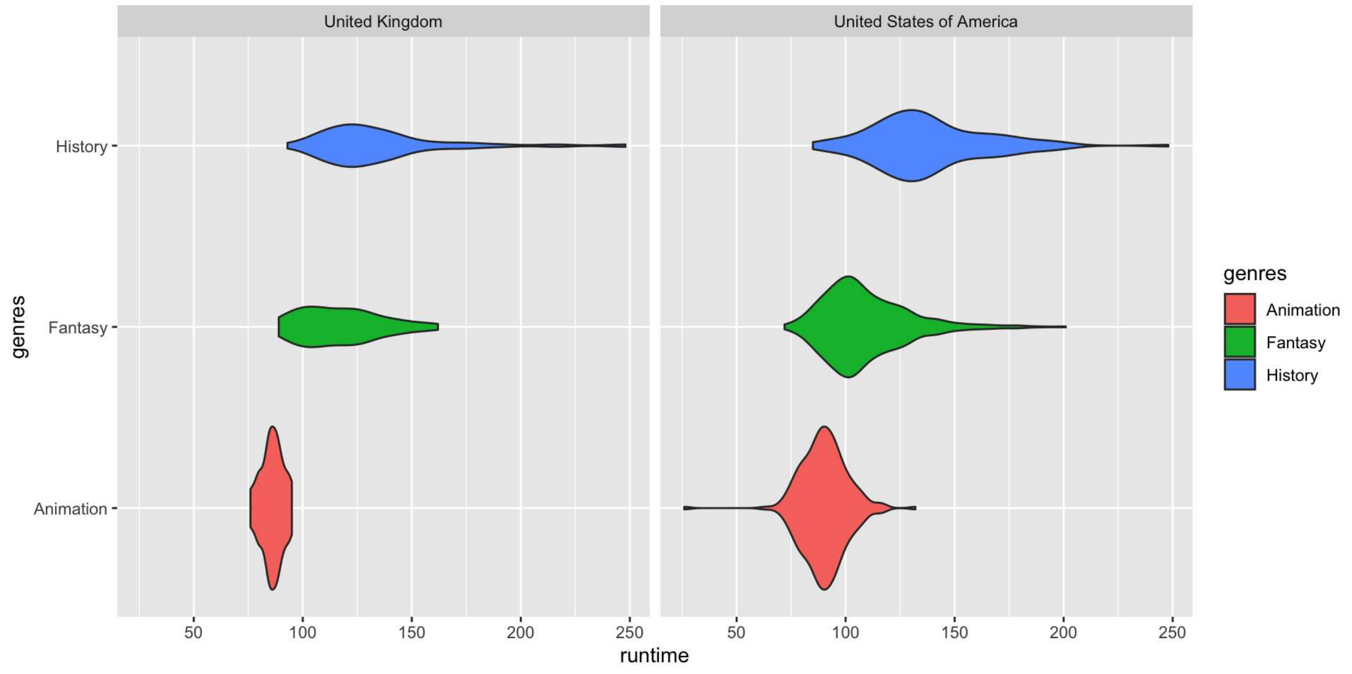


https://github.com/ubco-mds-2022/Data-550

Violin Plots

To get the violin plots, we simply change the geom:

```
ggplot(free_both) +
    aes(x = runtime, y = genres,fill = genres) +
    geom_violin() +
    facet_wrap(~countries)
```

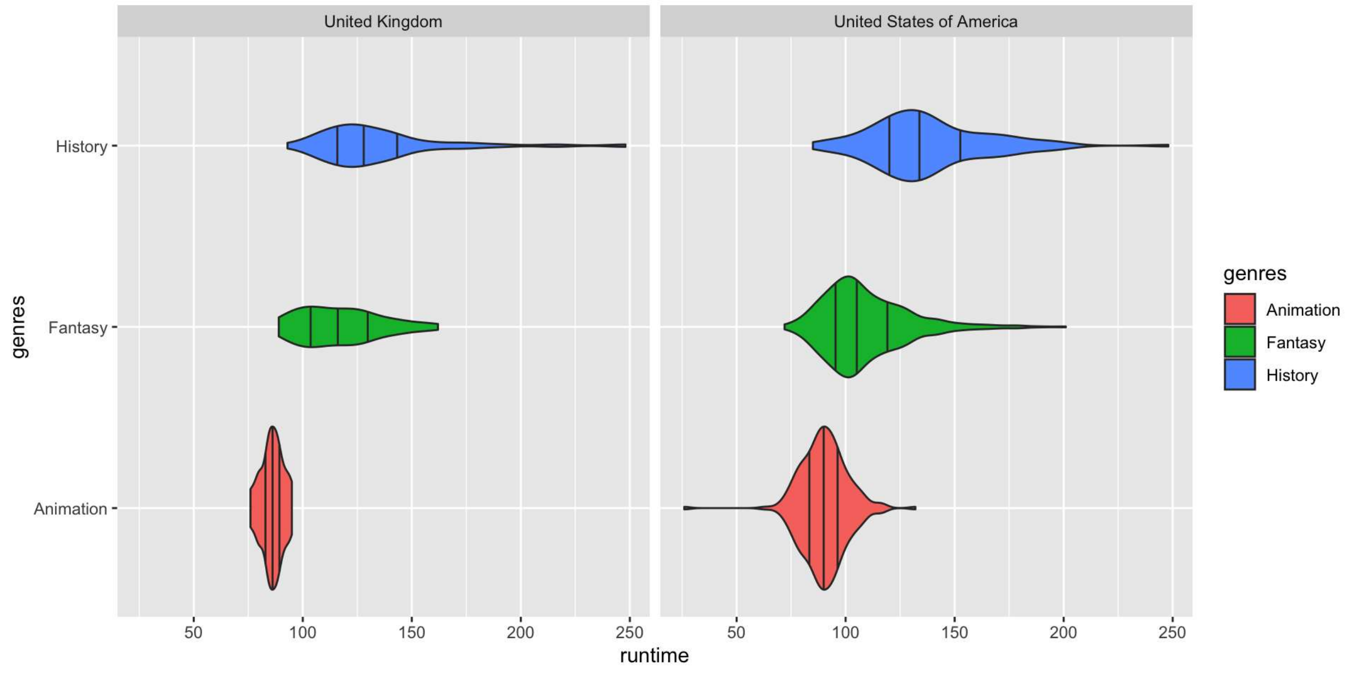


https://github.com/ubco-mds-2022/Data-550

Layering Quanties

We can layer the quantiles shown in the box plots

```r
ggplot(free_both) +
    aes(x = runtime, y = genres, fill = genres) +
    geom_violin(draw_quantiles = c(0.25, 0.5, 0.75)) +
    facet_wrap(~countries)
```



https://github.com/ubco-mds-2022/Data-550

Comments

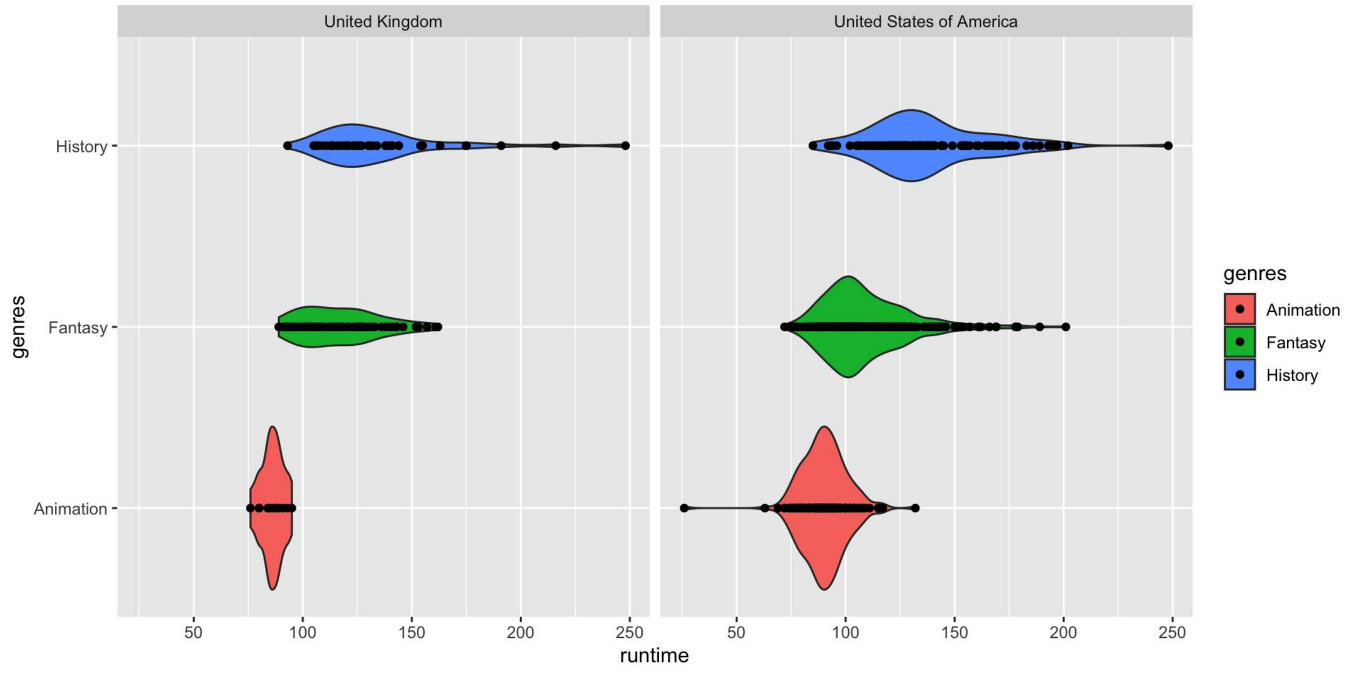
- When possible, it is a good idea to have a look at where the individual data points are.
- Of course we could always layer on different marking of our data (using `geom_point()` for example)
- However when we have a lot of data, this could be impossible to read.
- For this we can use a categorical scatter plot where the dots are spread/jittered[1] randomly on the non-value axis so that they don't all overlap via `geom_jitter()`.

1. jittering is not yet available in Altair

Layering Points

We can layer the points onto the violin plots:

```r
ggplot(free_both) +
    aes(x = runtime, y = genres, fill = genres) +
    geom_violin() + geom_point() +
    facet_wrap(~countries)
```

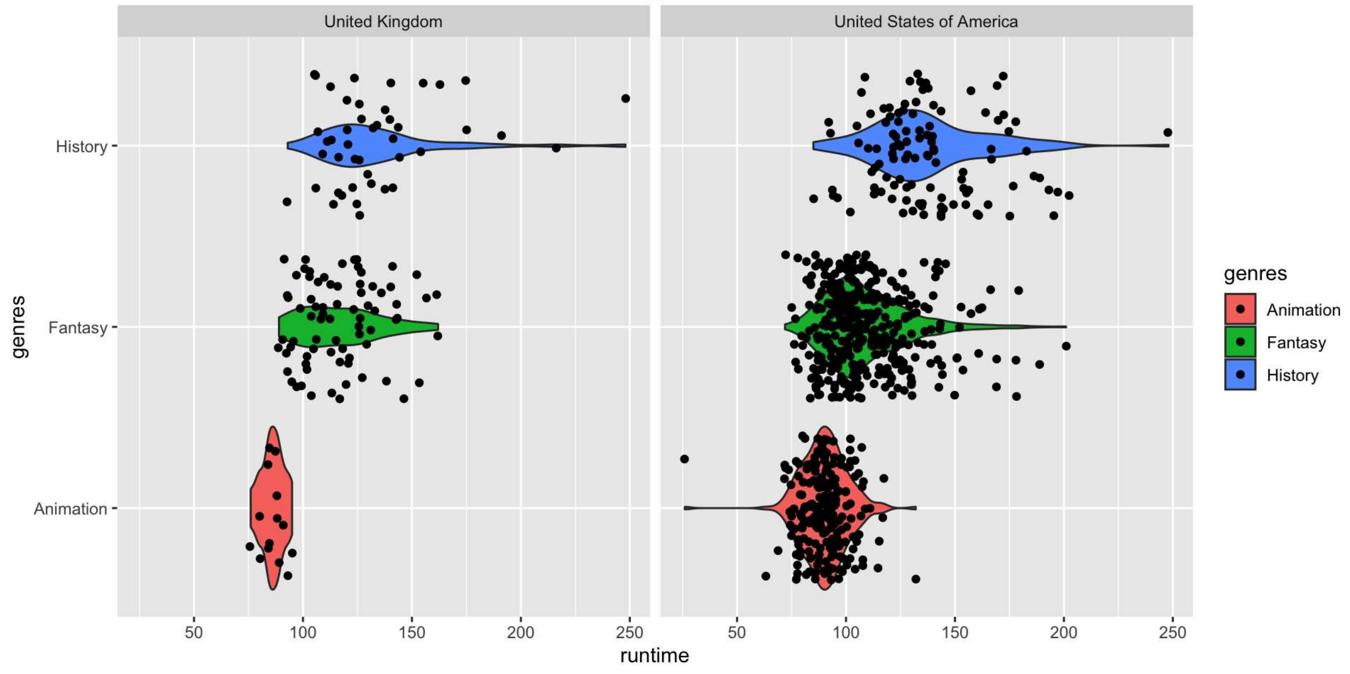


https://github.com/ubco-mds-2022/Data-550

Jittering Data

“jittering” adds some noise to the location of each point

```r
ggplot(free_both) +
    aes(x = runtime, y = genres, fill = genres) +
    geom_violin() + geom_jitter() +
    facet_wrap(~countries)
```

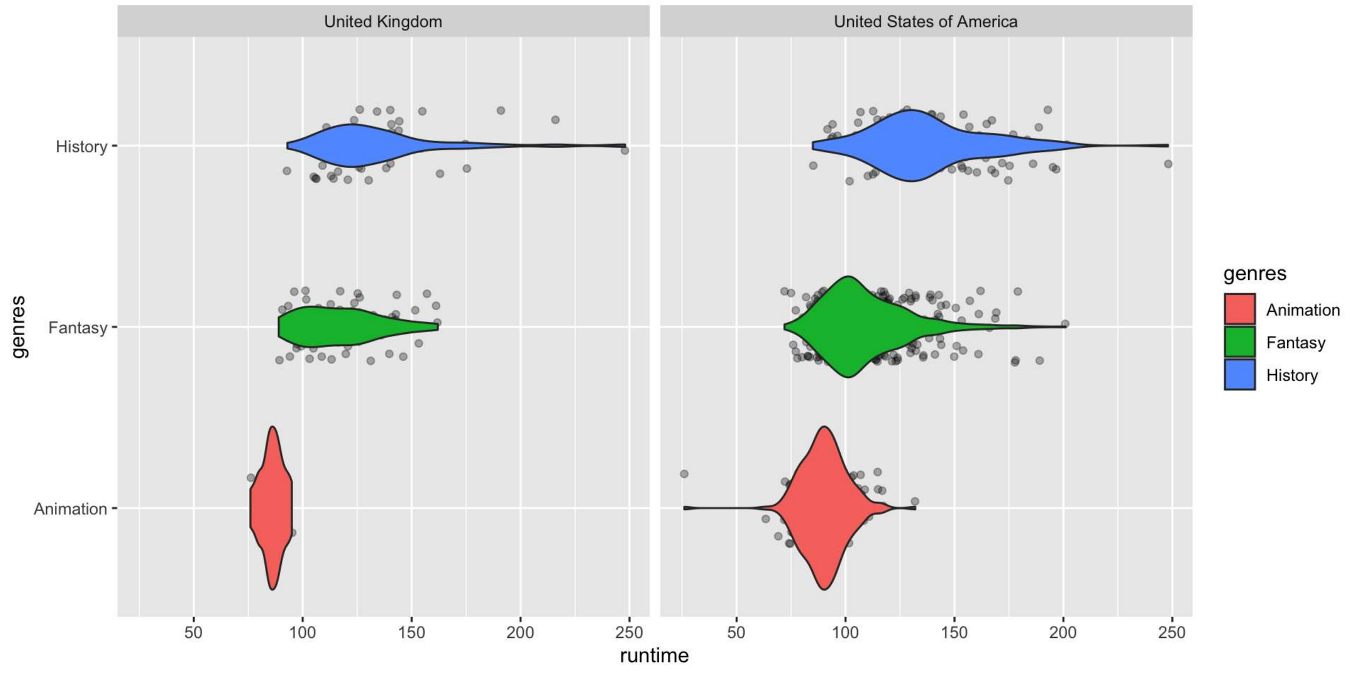


https://github.com/ubco-mds-2022/Data-550

Order matters

We can change the default height and order or layers

```r
ggplot(free_both) +
    aes(x = runtime, y = genres, fill = genres) +
    geom_jitter(height = 0.2, alpha = 0.3) + geom_violin() +
    facet_wrap(~countries)
```

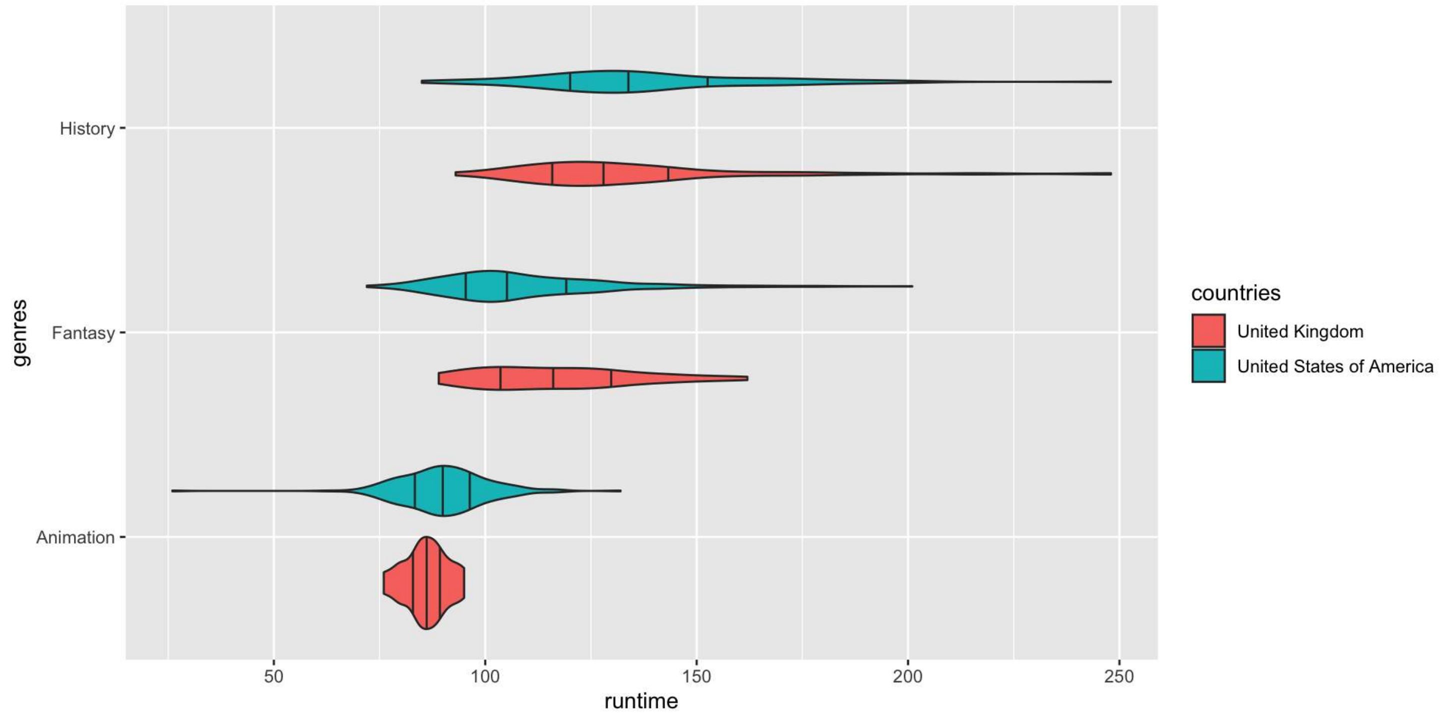


https://github.com/ubco-mds-2022/Data-550

Unfaceting

Rather than faceting we could fill by `countries`

```r
ggplot(free_both) +
    aes(x = runtime, y = genres, fill = countries) +
    geom_violin(draw_quantiles = c(0.25, 0.5, 0.75))
```



https://github.com/ubco-mds-2022/Data-550

https://github.com/ubco-mds-2022/Data-550