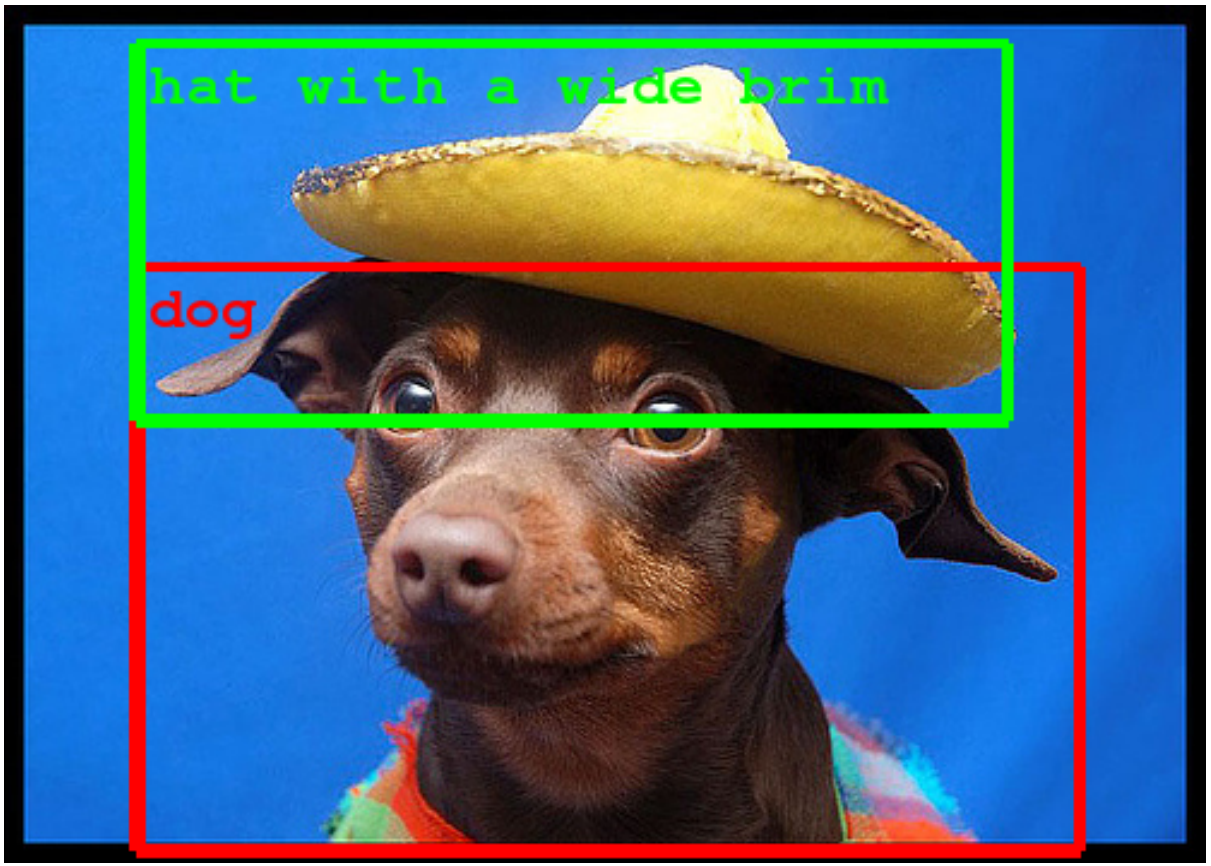
Chapter 5: Dependency Relationships

Joyce Robbins

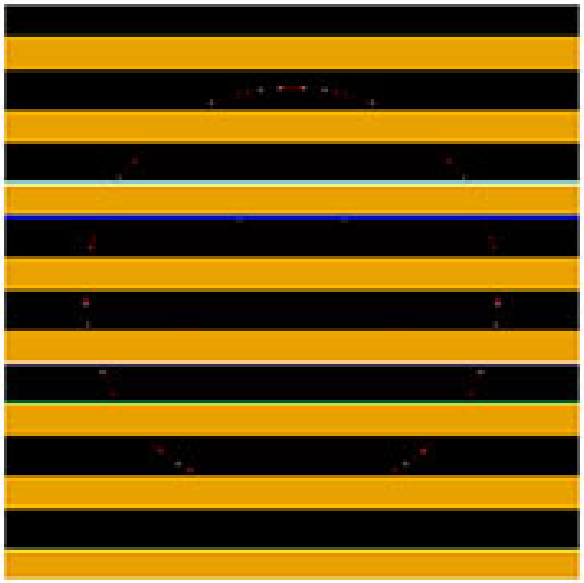
Human vision and interpretation

- https://www.cnn.com/2016/07/11/us/baton-rouge-protester-photograph/index.html

Computer vision and interpretation (successful)

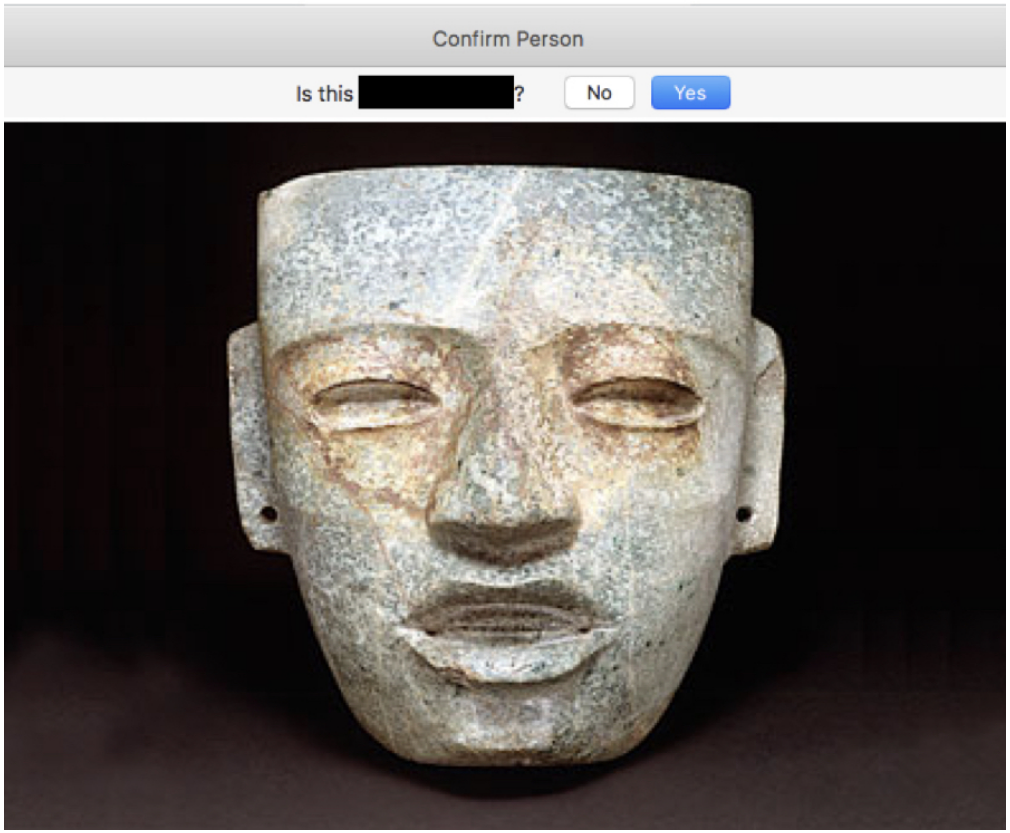


Computer vision and interpretation (unsuccessful)



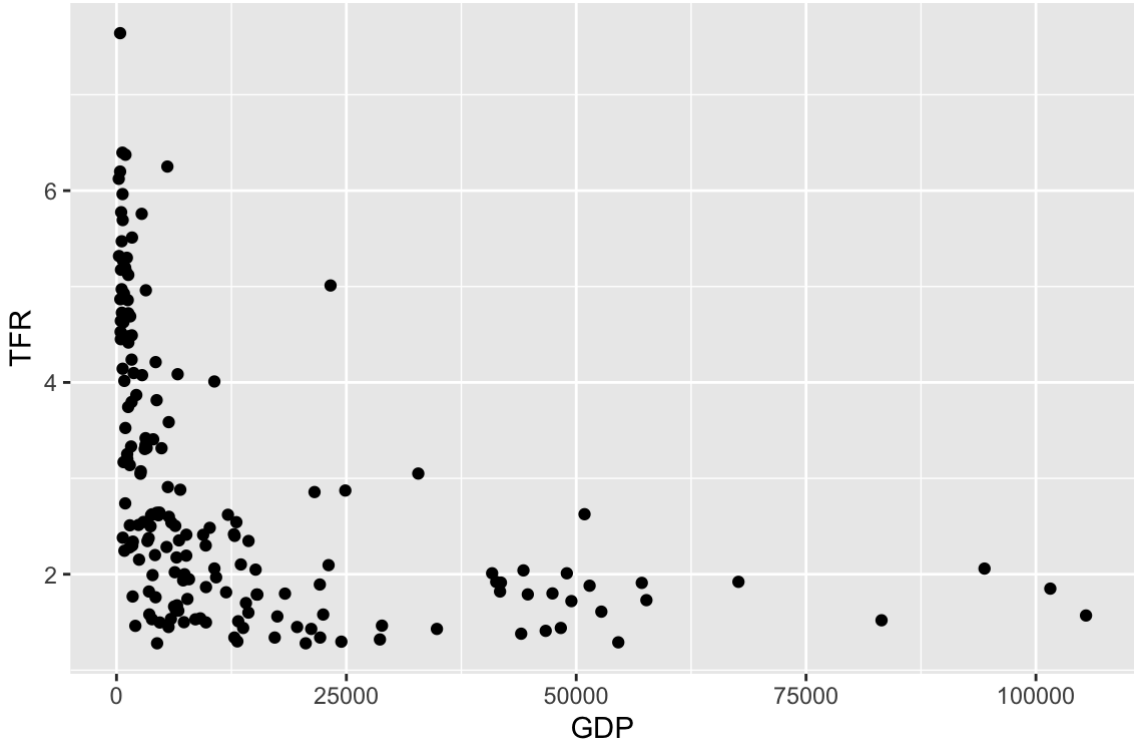
school bus

Face recognition (unsuccessful)



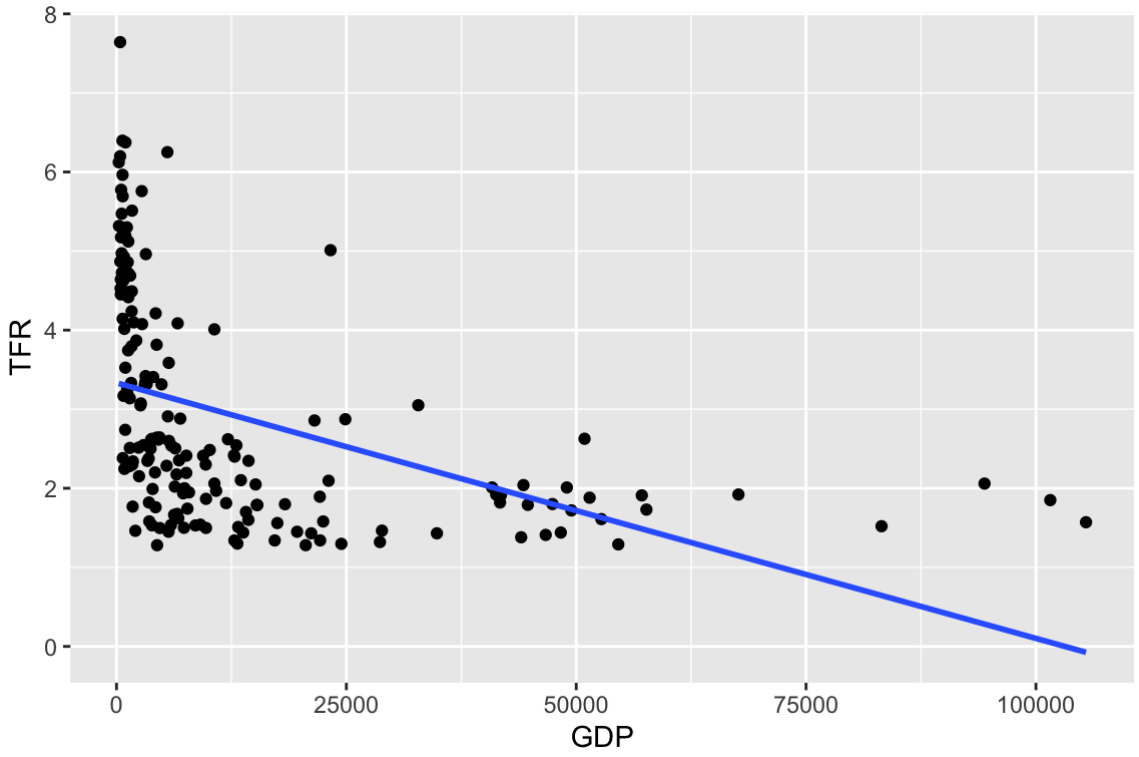
Scatterplots

We plot two continuous variables to investigate how they are related



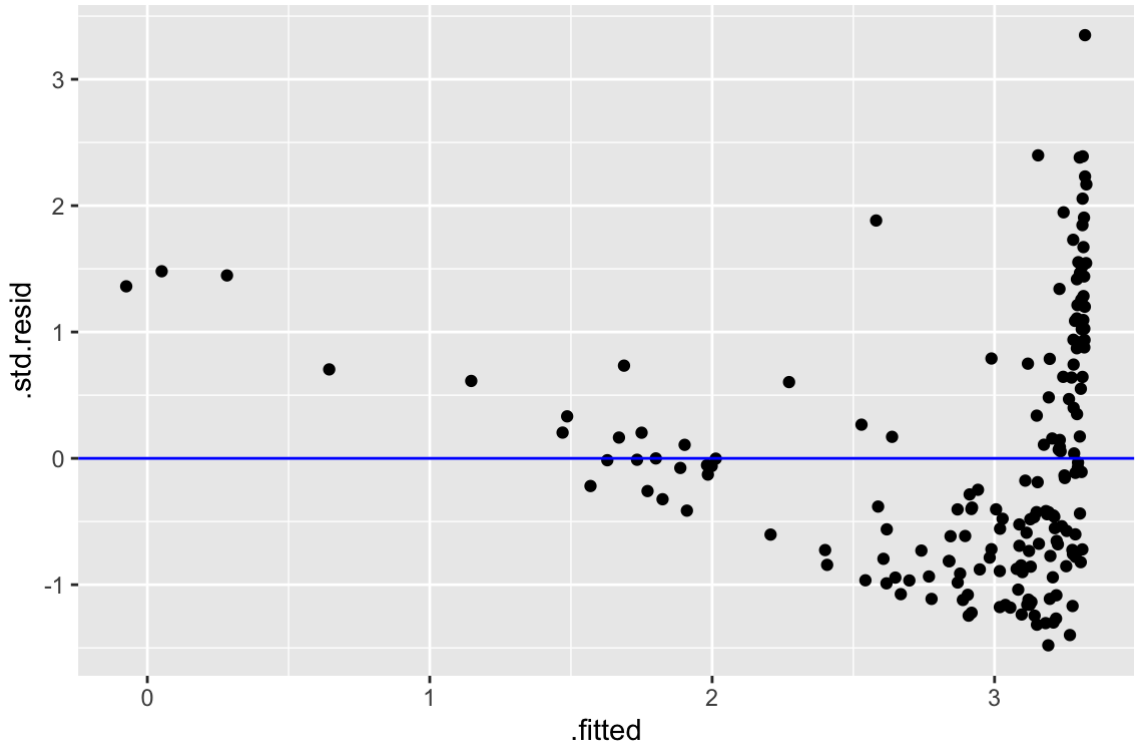
- Why not use linear models instead?

Linear model

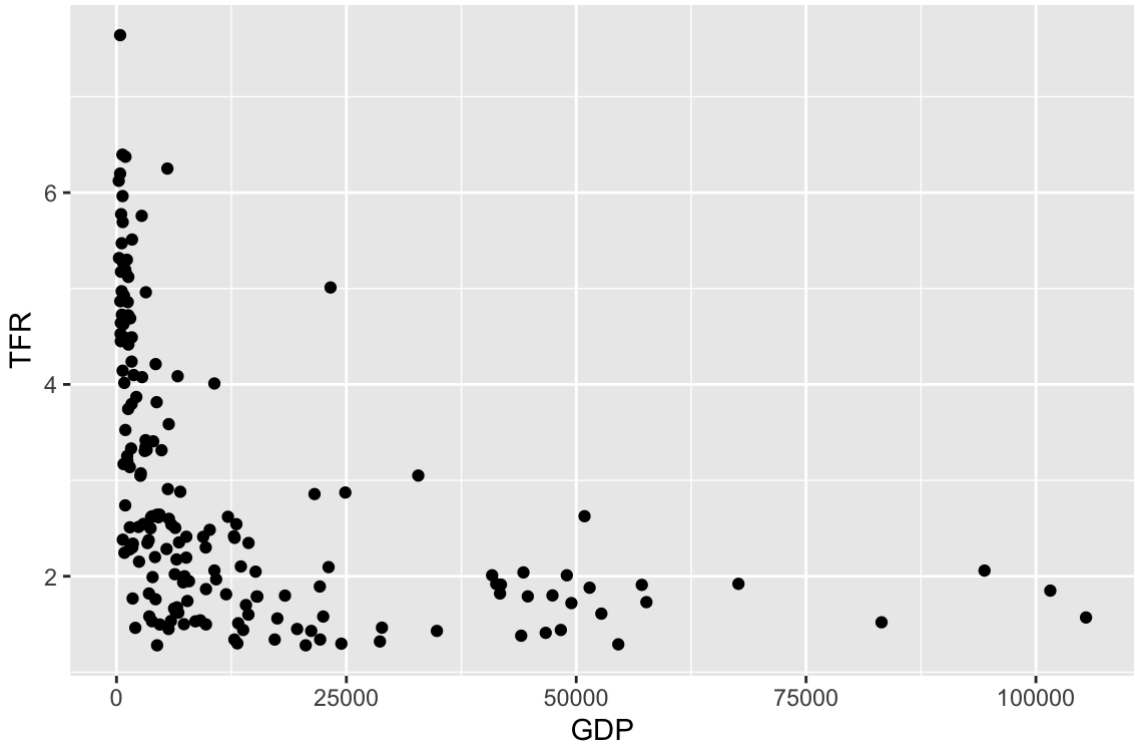


- R-squared: 0.192

Residual plot

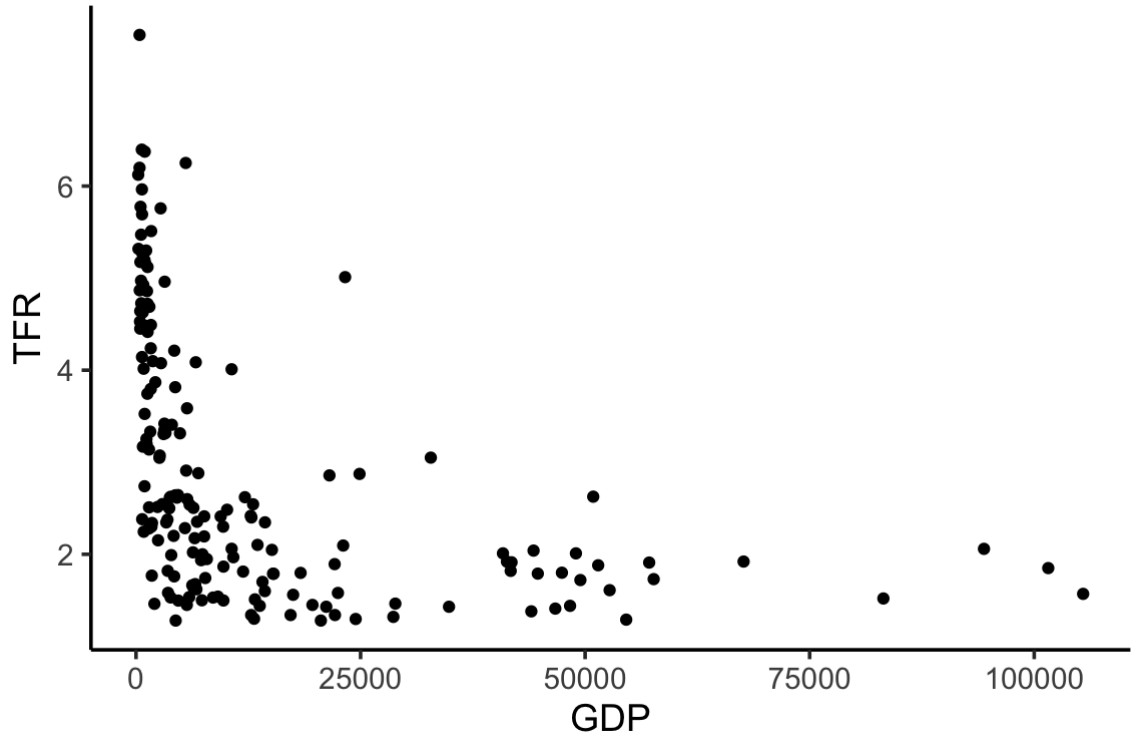


Total Fertility Rate vs. Gross Domestic Product



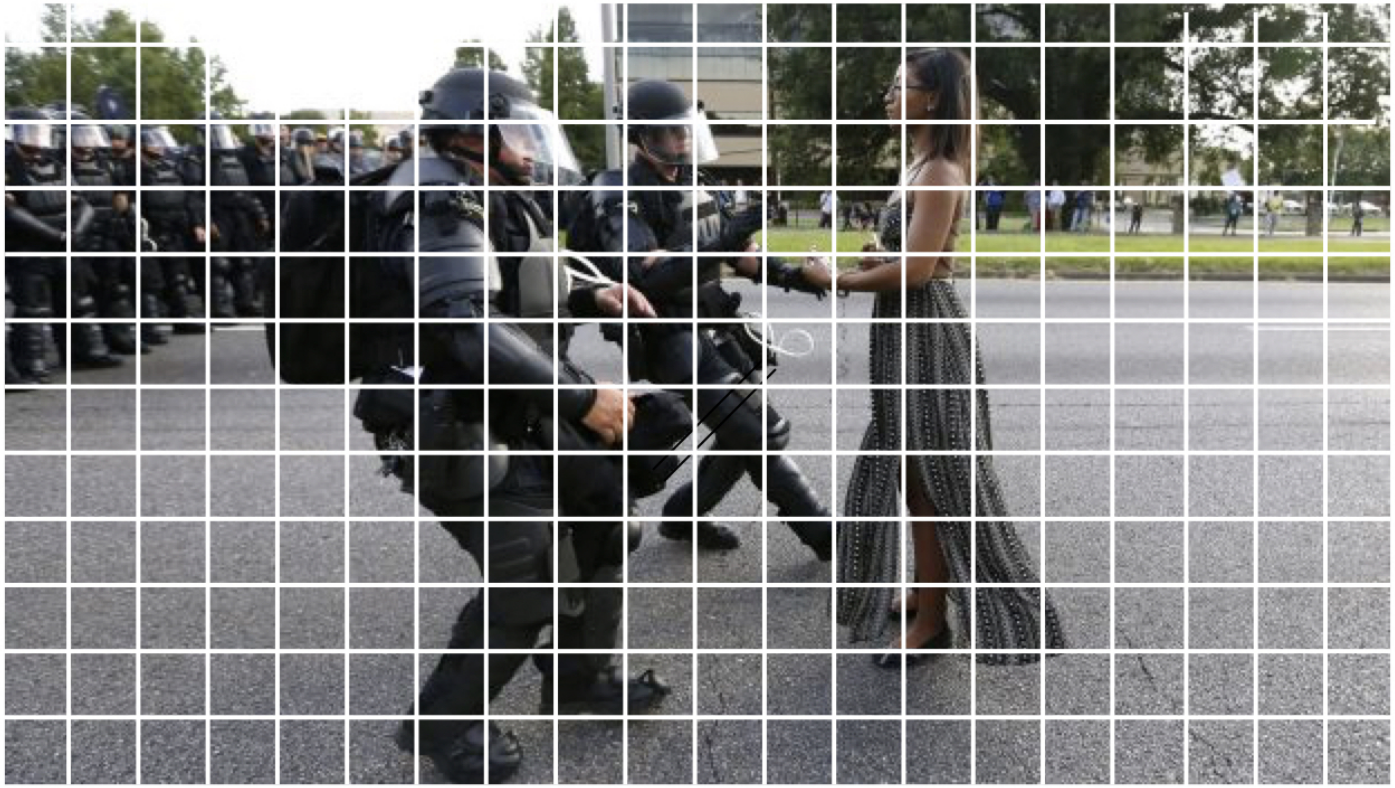
(easier to read individual values with gridlines)

Total Fertility Rate vs. Gross Domestic Product



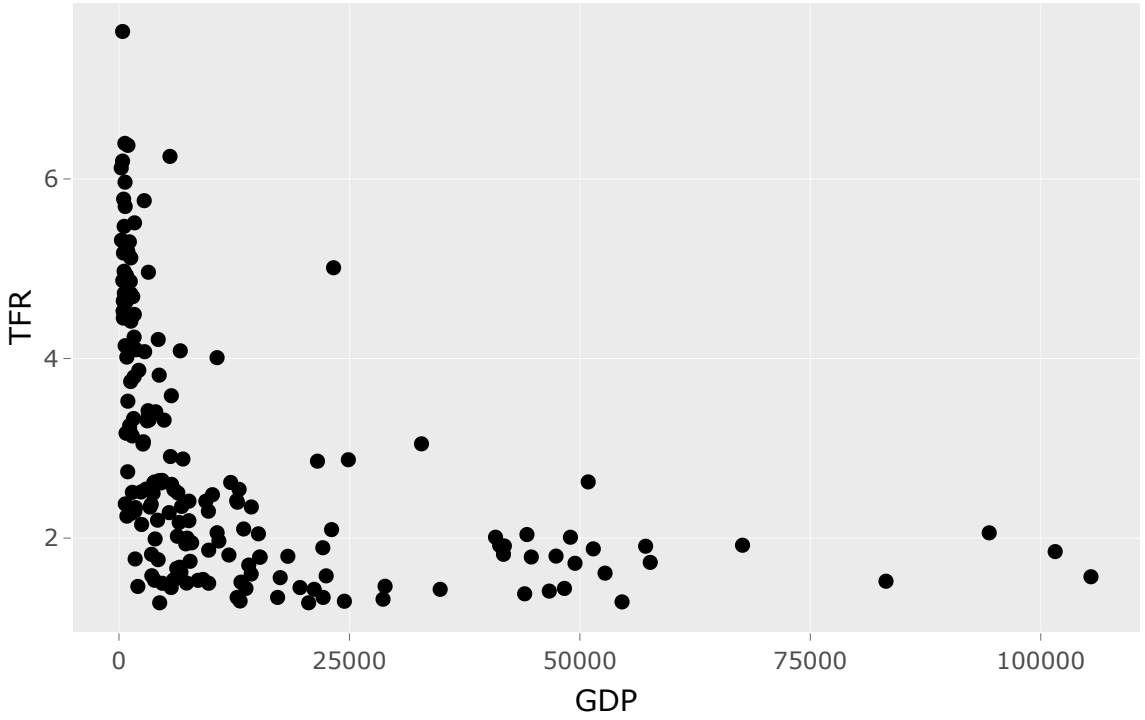
easier to detect spatial patterns without gridlines

Gridlines



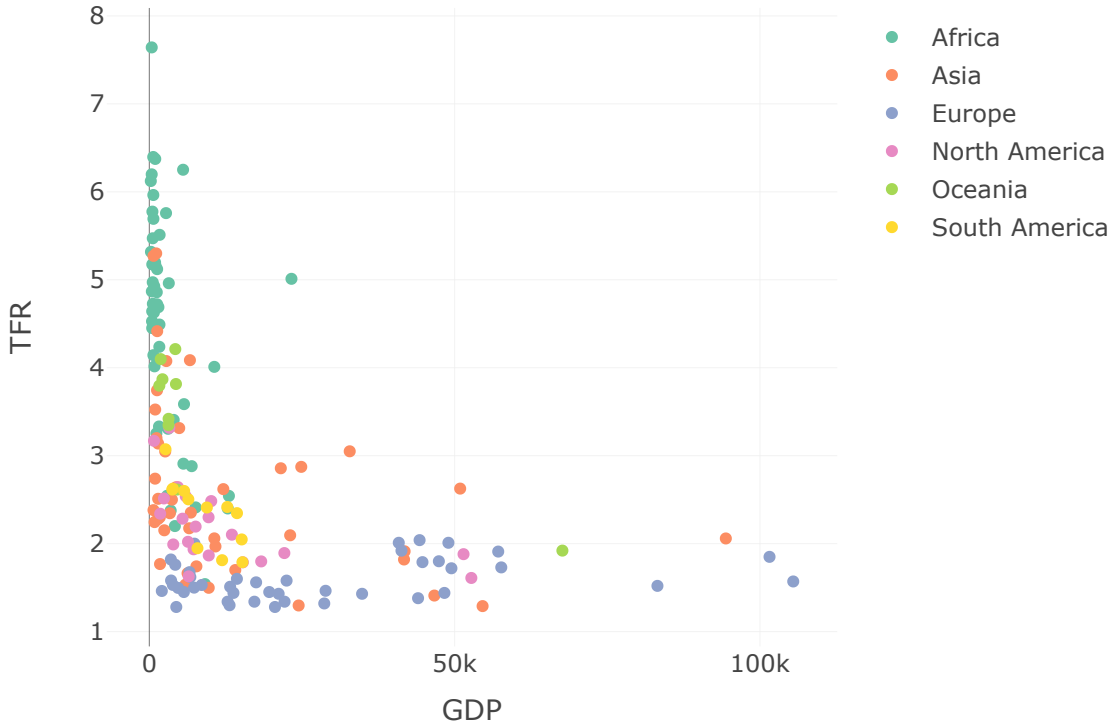
Interactive (Plotly ggplot2 library)

```
# https://plot.ly/ggplot2/
library(plotly)
ggplotly(g)
```



Interactive (Plotly R library)

```r
# https://plot.ly/r/
plot_ly(world, x = ~GDP, y = ~TFR,
        color = ~CONTINENT, text = ~COUNTRY,
        hoverinfo = 'text') %>%
  add_markers()
```



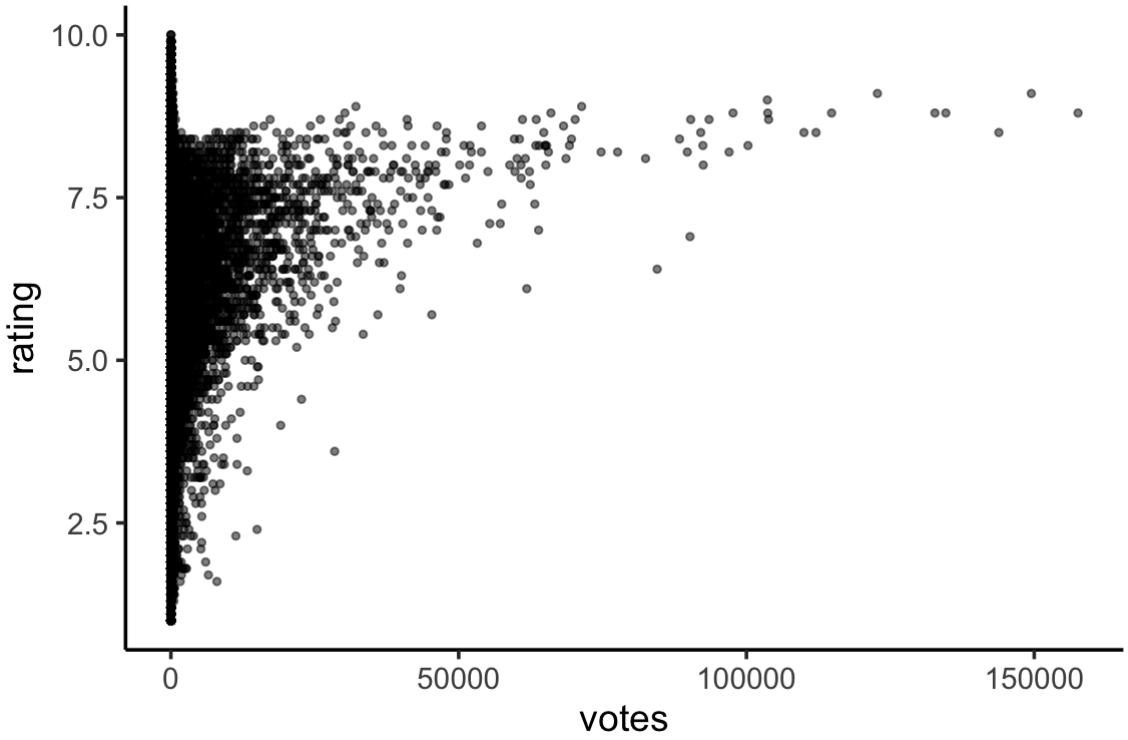
What features might be visible in scatterplots? (p. 77)

- Causal relationships

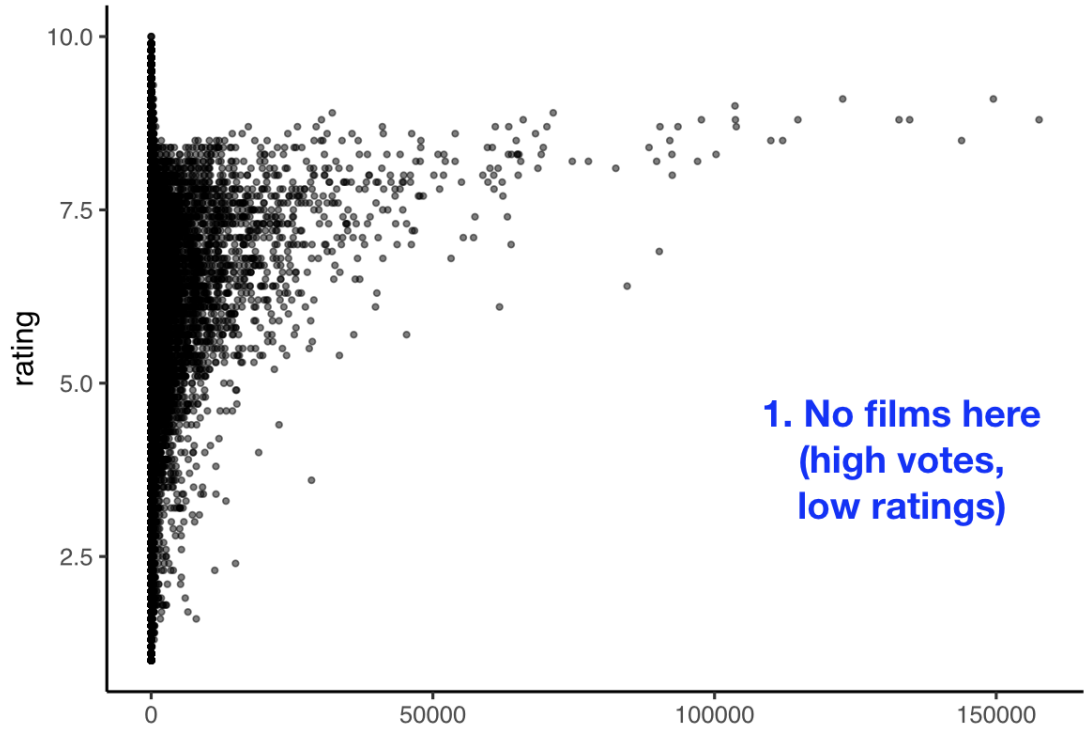
What features might be visible in scatterplots? (p. 77)

- ~~Causal relationships~~ (correlation ≠ causation, but still use y-axis for what appears to be the dependent variable)
- Associations
  describe what you see
- Outliers
- Clusters
- Gaps
- Barriers (boundaries)
- Conditional relationships
  (different relationships for different intervals of x)

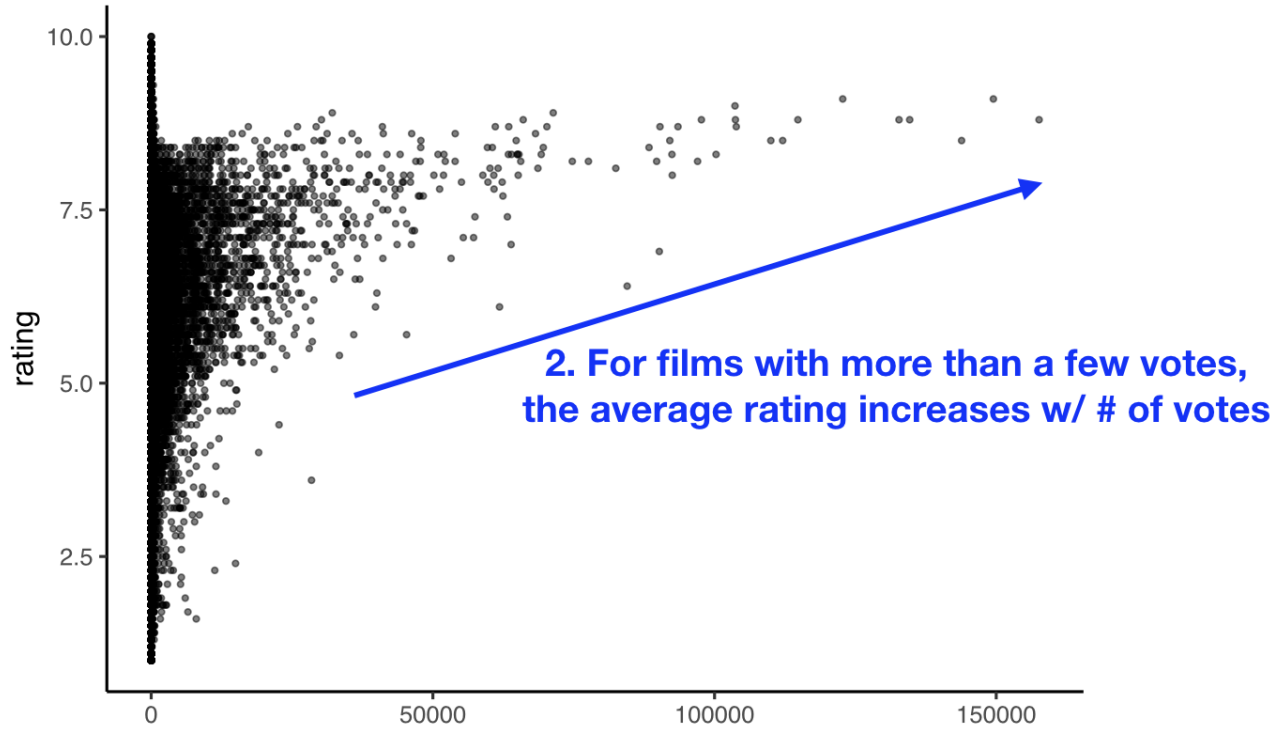
`ggplot2movies`: rating vs. votes (y ~ x)



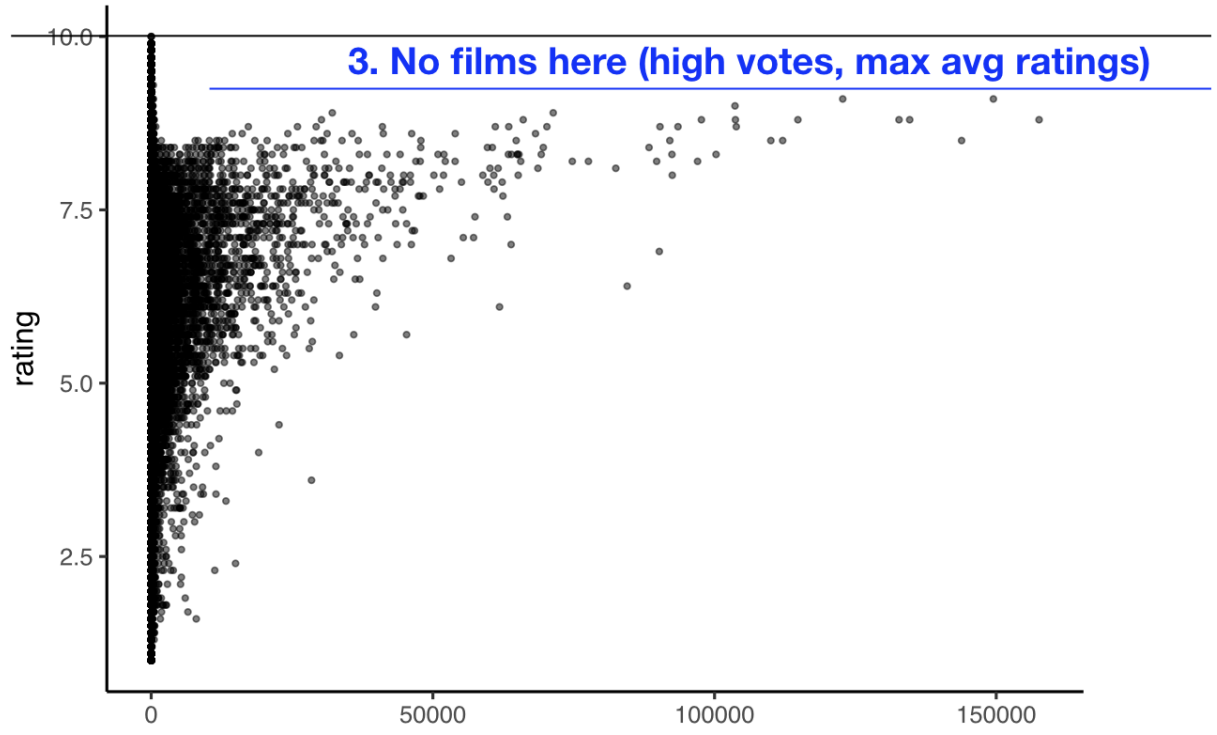
Movie Ratings (GDwR, p. 82)



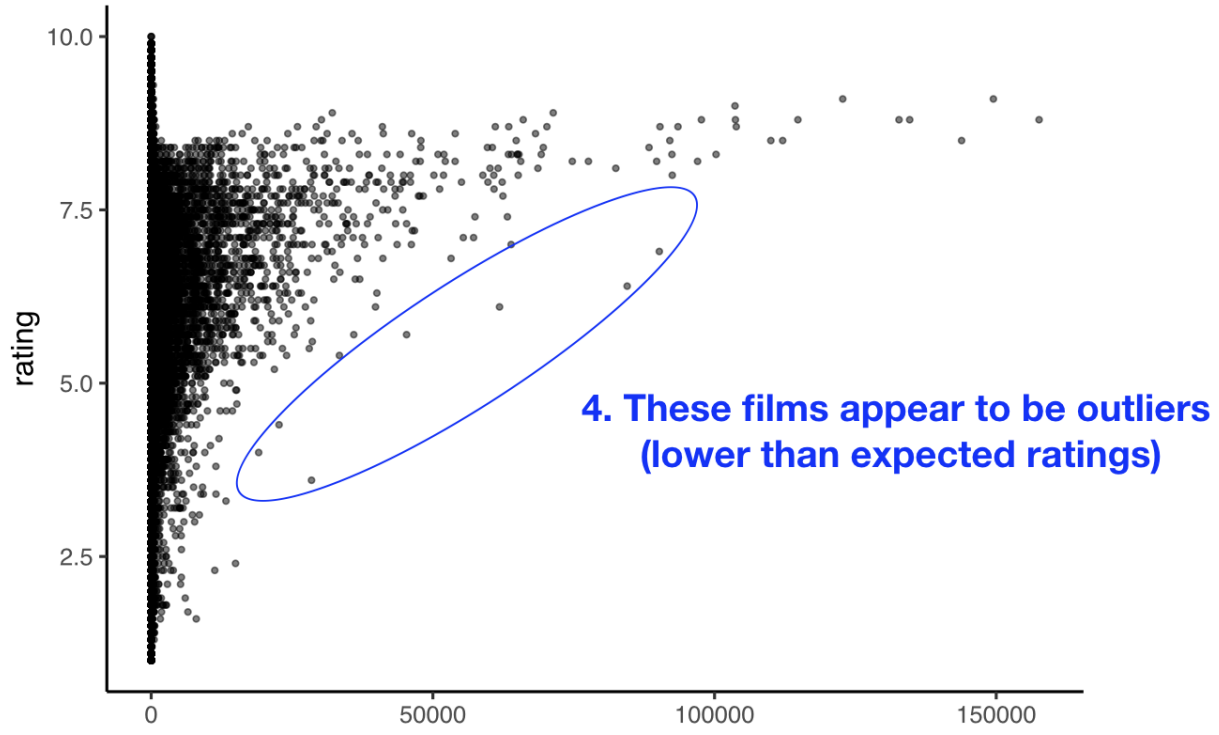
Movie Ratings (GDwR, p. 82)



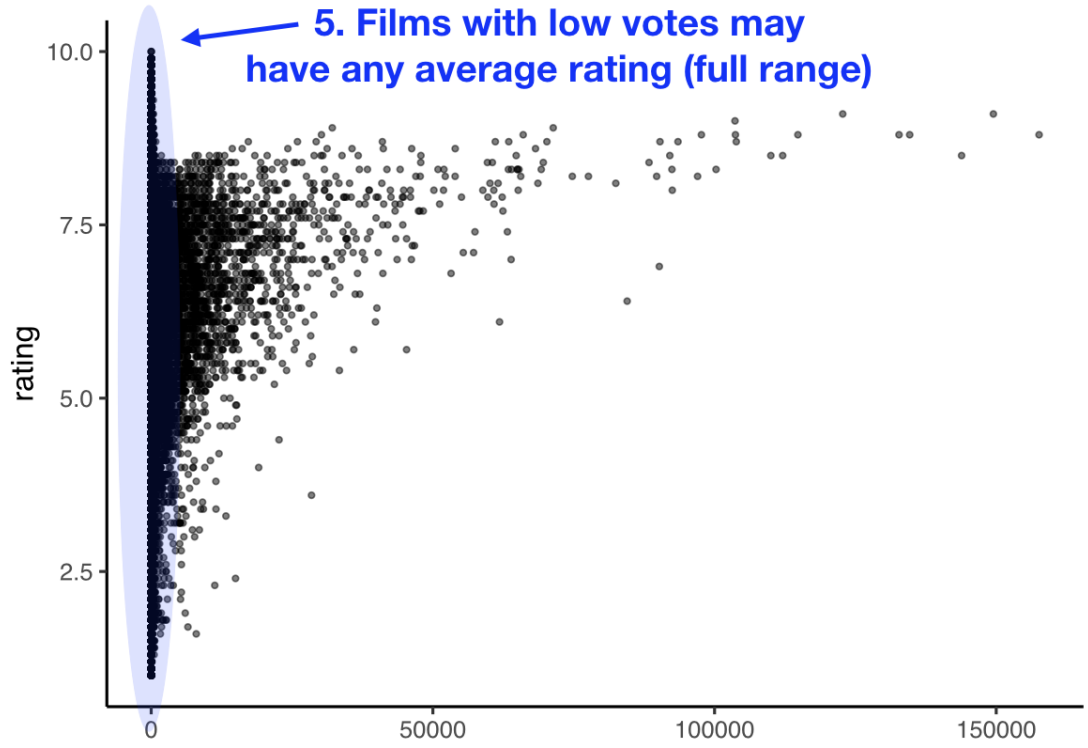
Movie Ratings (GDwR, p. 82)



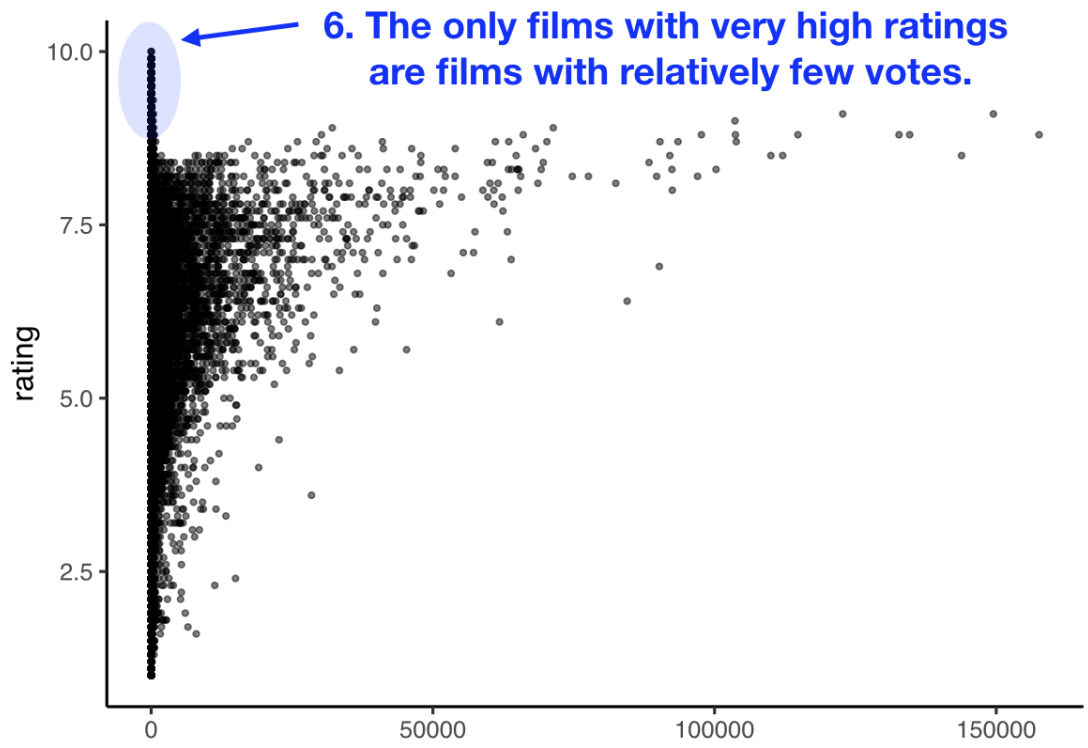
Movie Ratings (GDwR, p. 82)



Movie Ratings (GDwR, p. 82)

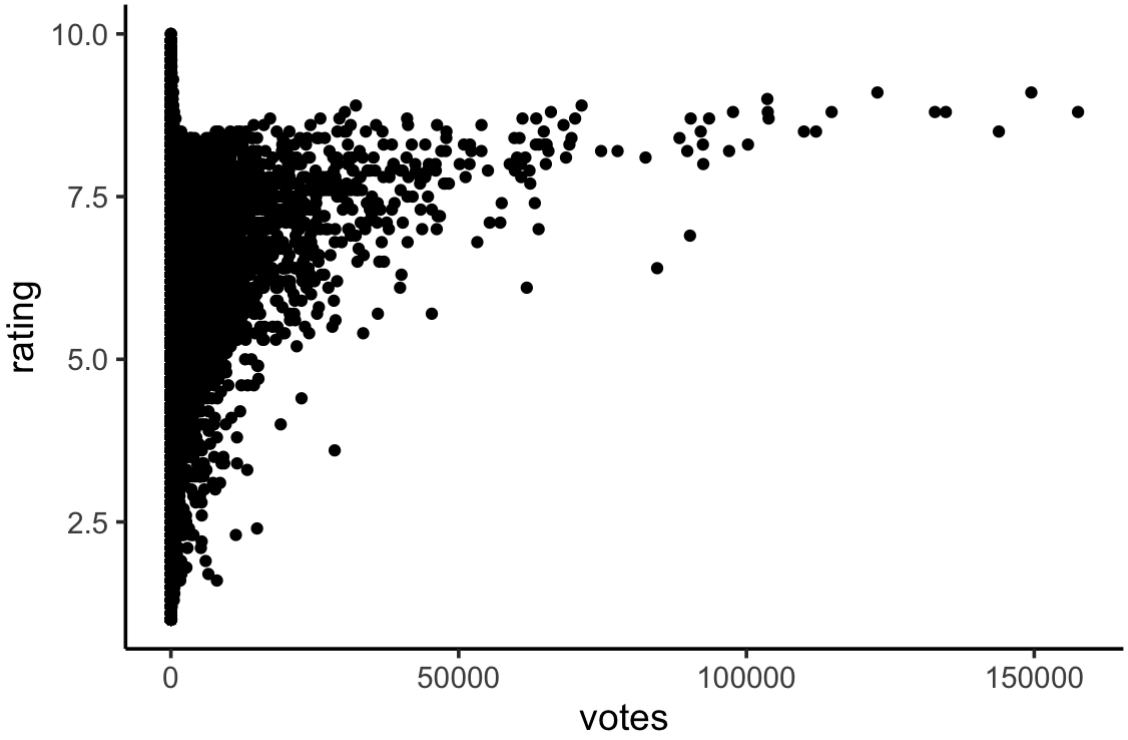


Movie Ratings (GDwR, p. 82)



Scatterplot strategies for dealing with overplotting

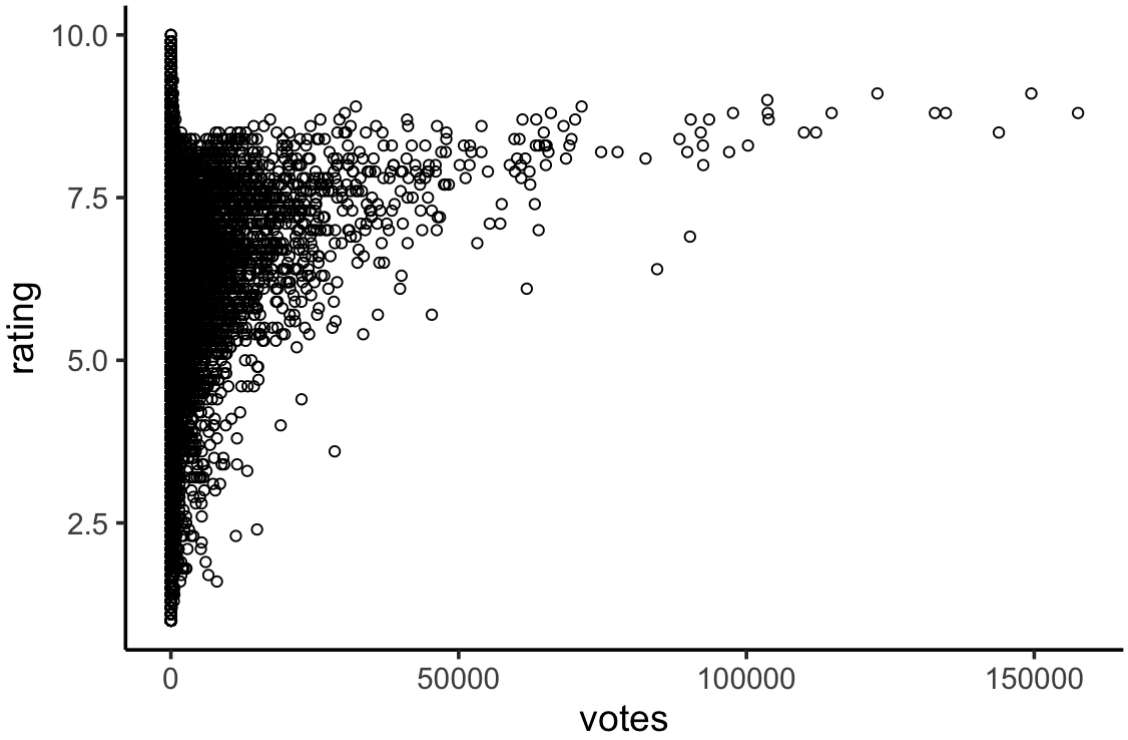
How can we get a better view of the data?



Scatterplot strategies for dealing with overplotting

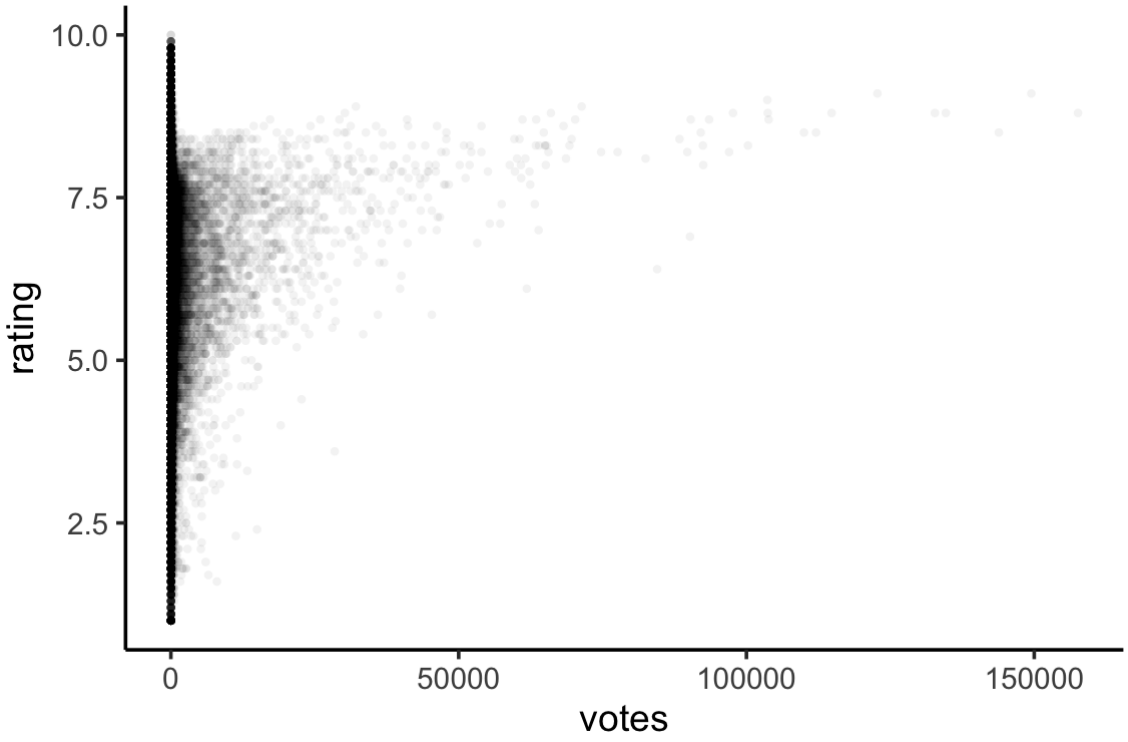
- Change the points in some way
  - *open circles*
  - *alpha blending*
  - *smaller circles*
- Don’t plot all points
  - *randomly sample data*
  - *subset data (static or interactive)*
  - *remove outliers*
- Transform to log scale

Open circles



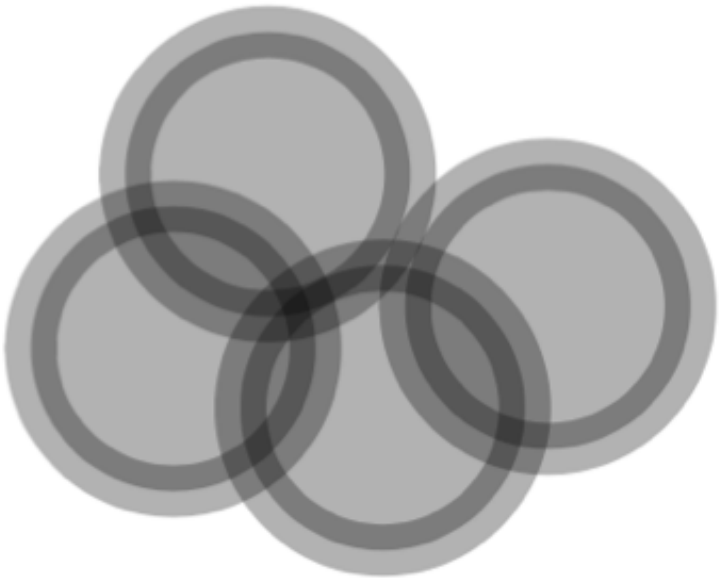
Alpha blending

```
alpha = .05
```

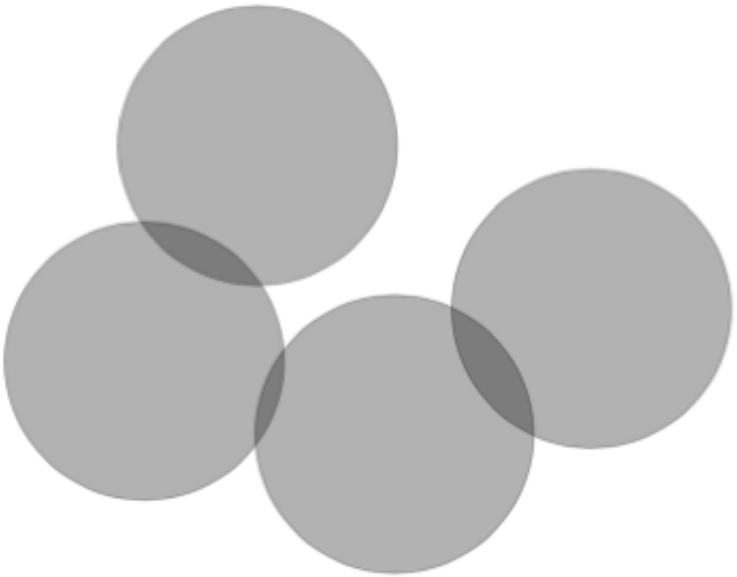


Alpha blending tip (ggplot2)

```
geom_point(alpha = .3)
```



```
geom_point(alpha = .3,
           stroke = 0)
```

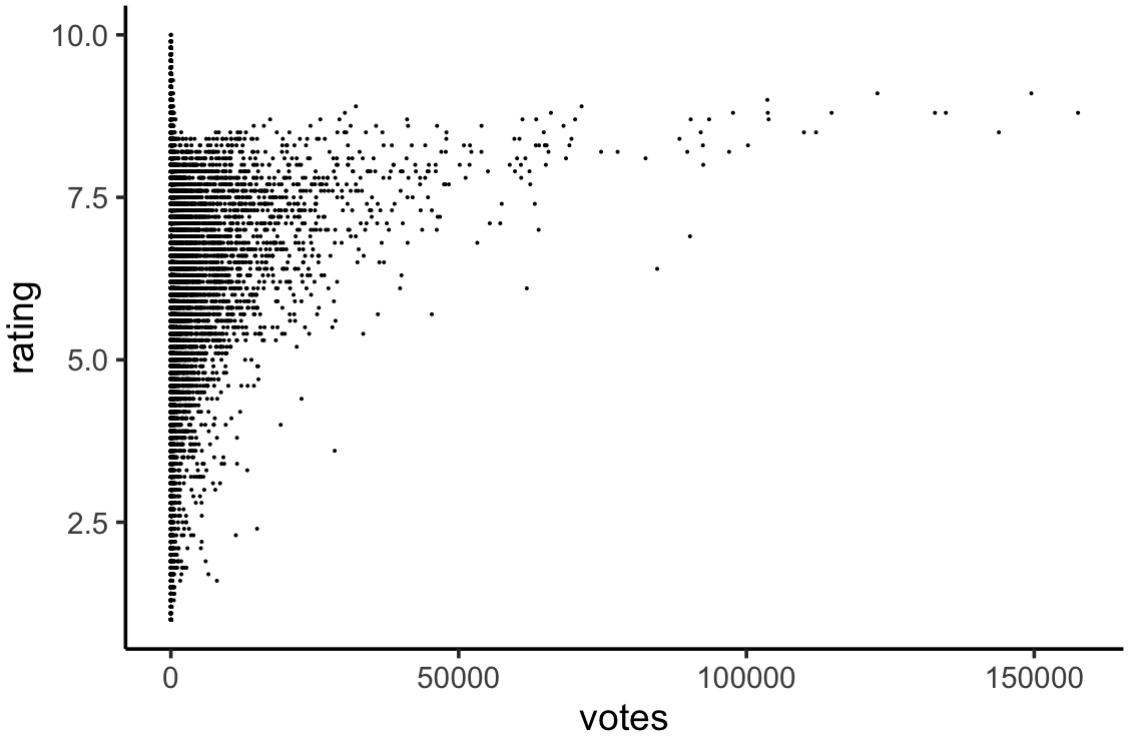


Smaller dots

```
ggplot2:::check_subclass("point", "Geom")$default_aes
```

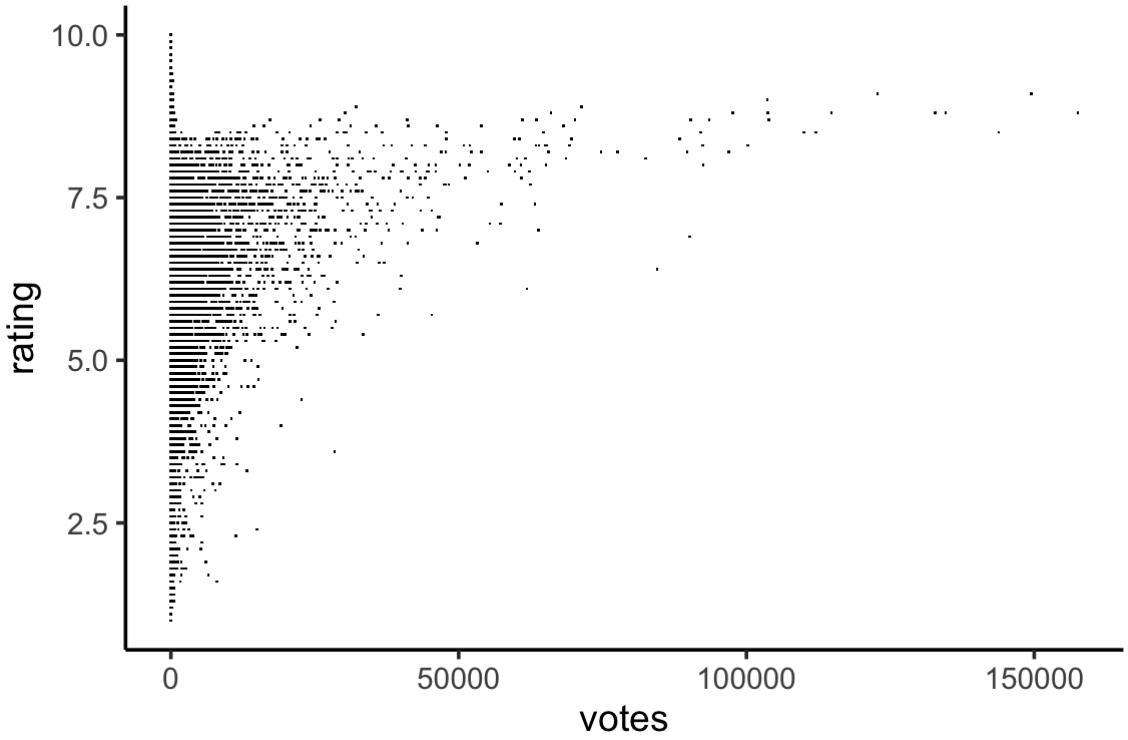
```
## Aesthetic mapping:
## * `shape`  -> 19
## * `colour` -> "black"
## * `size`   -> 1.5
## * `fill`   -> NA
## * `alpha`  -> NA
## * `stroke` -> 0.5
```

`size = .05`



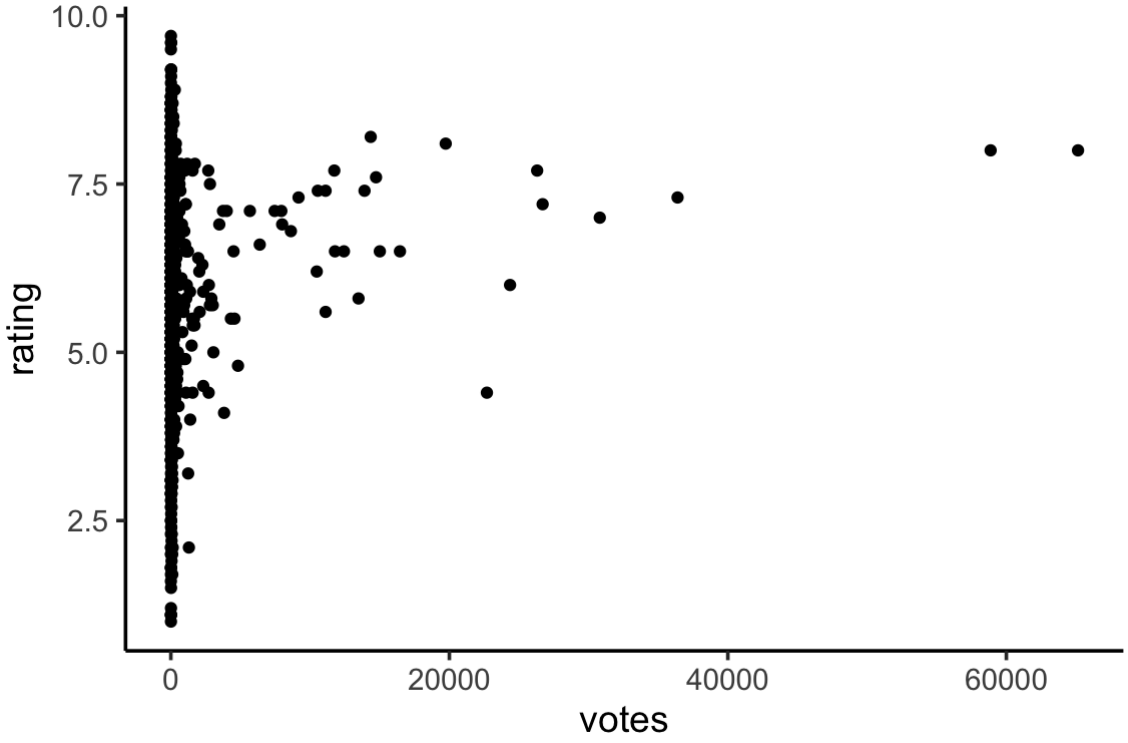
Smaller dots

```
shape = "."
```



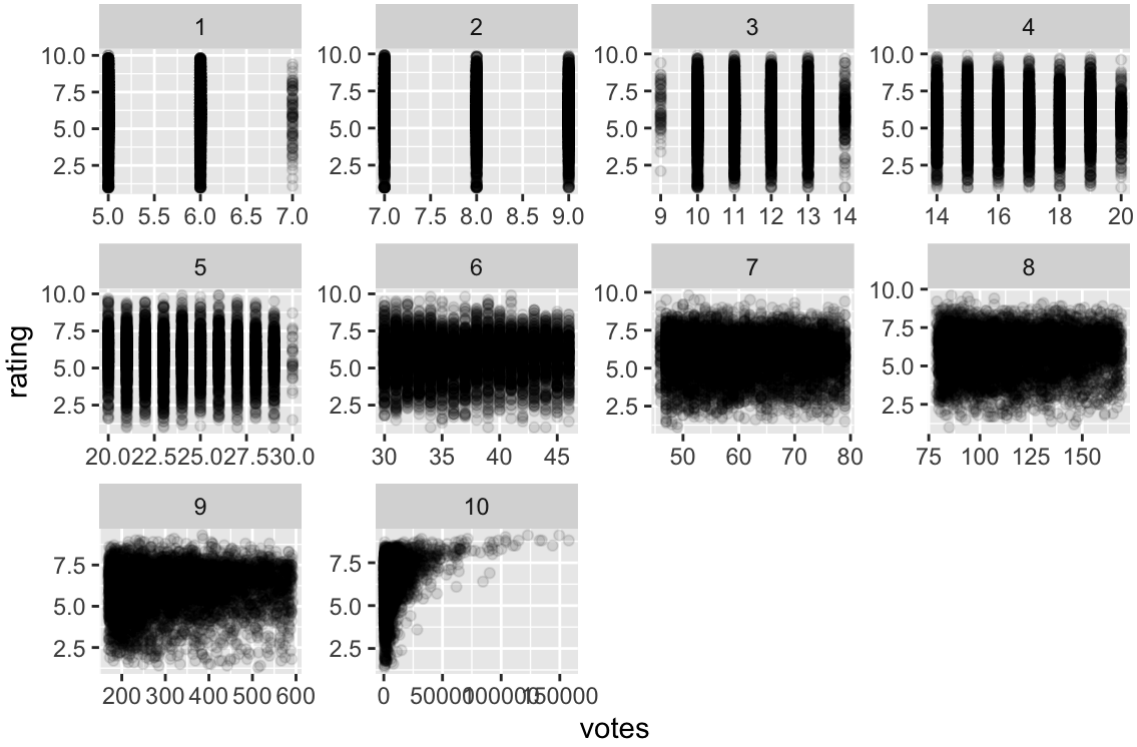
Randomly sample data

```
moves |> slice_sample(n = 1000)
```

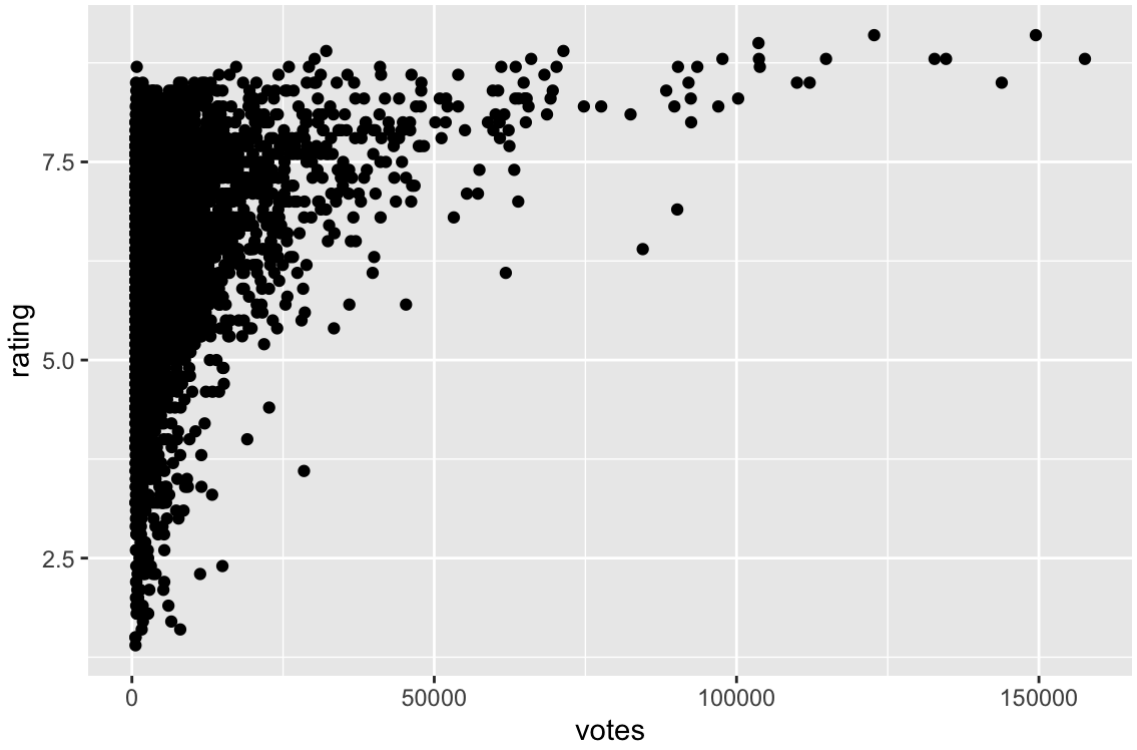


Subset

```
## # A tibble: 6 × 3
##   rating votes mybin
##    <dbl> <int> <int>
## 1    6.4   348     9
## 2    6      20     4
## 3    8.2     5     1
## 4    8.2     6     1
## 5    3.4    17     4
## 6    4.3    45     6
```



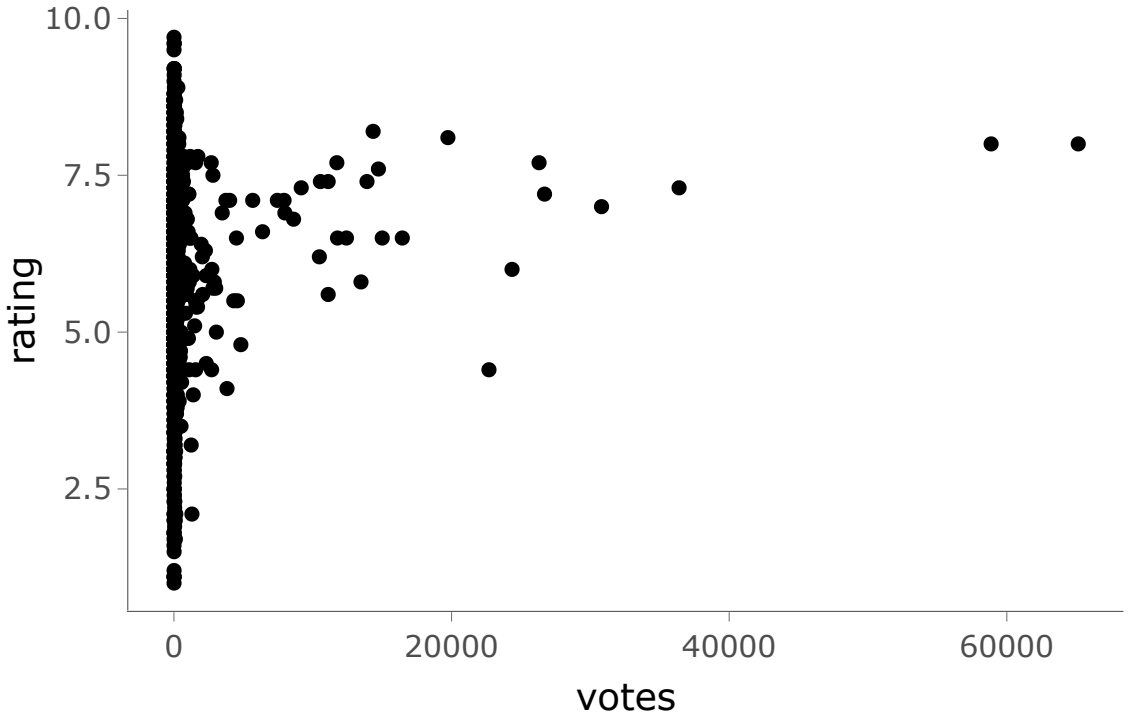
10% with highest number of votes



Subset interactively

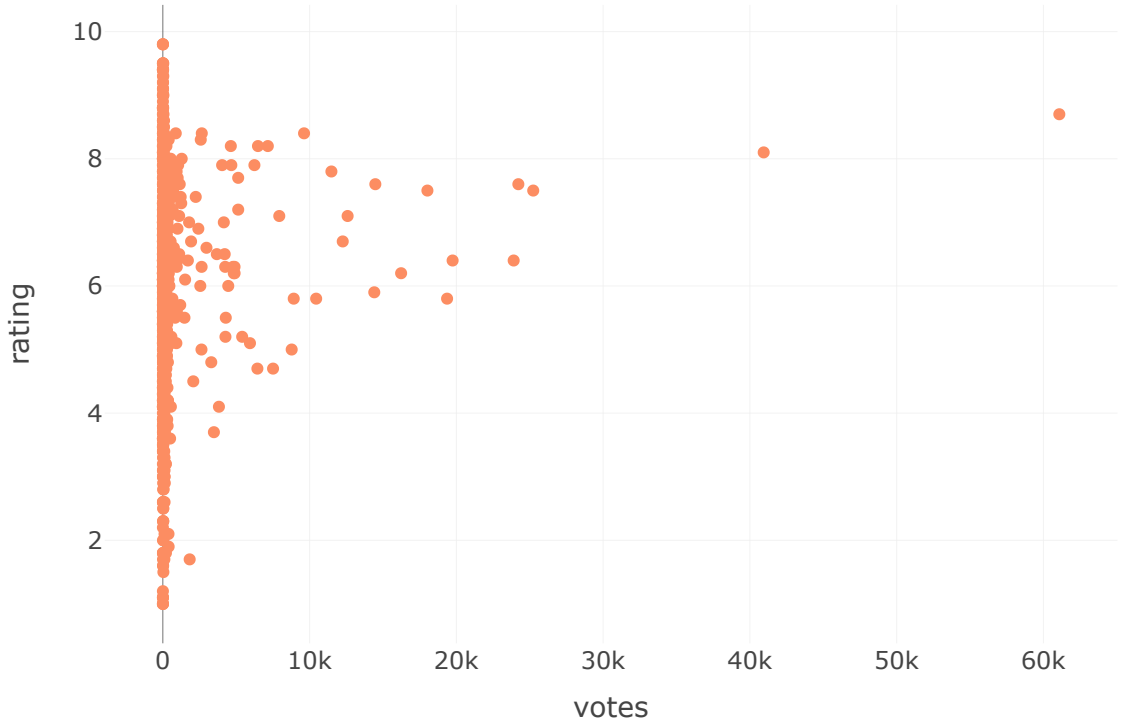
`ggplotly()`

(sample of 1000 movies)

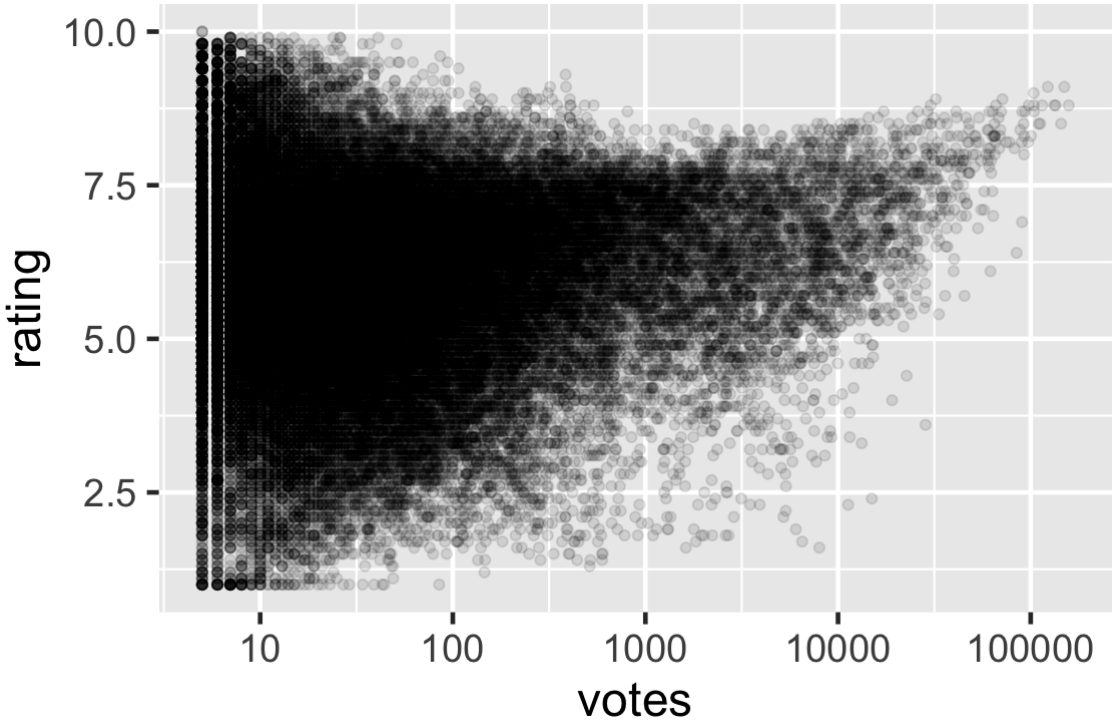


Subset interactively

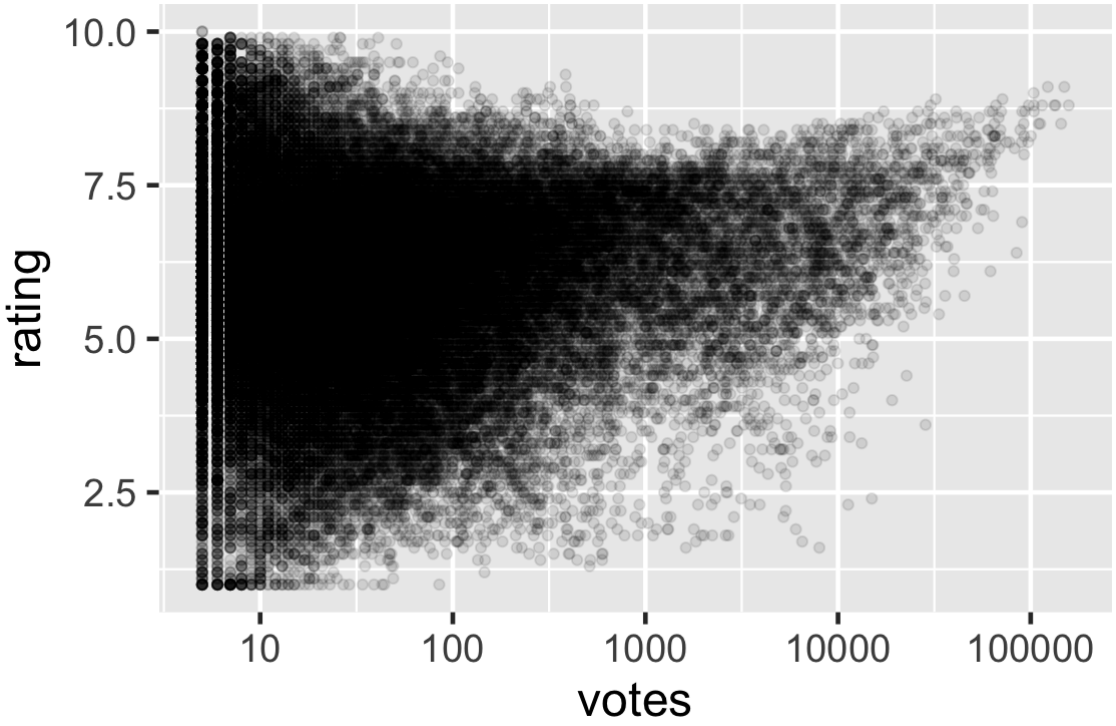
```
plot_ly()
```



Log scale



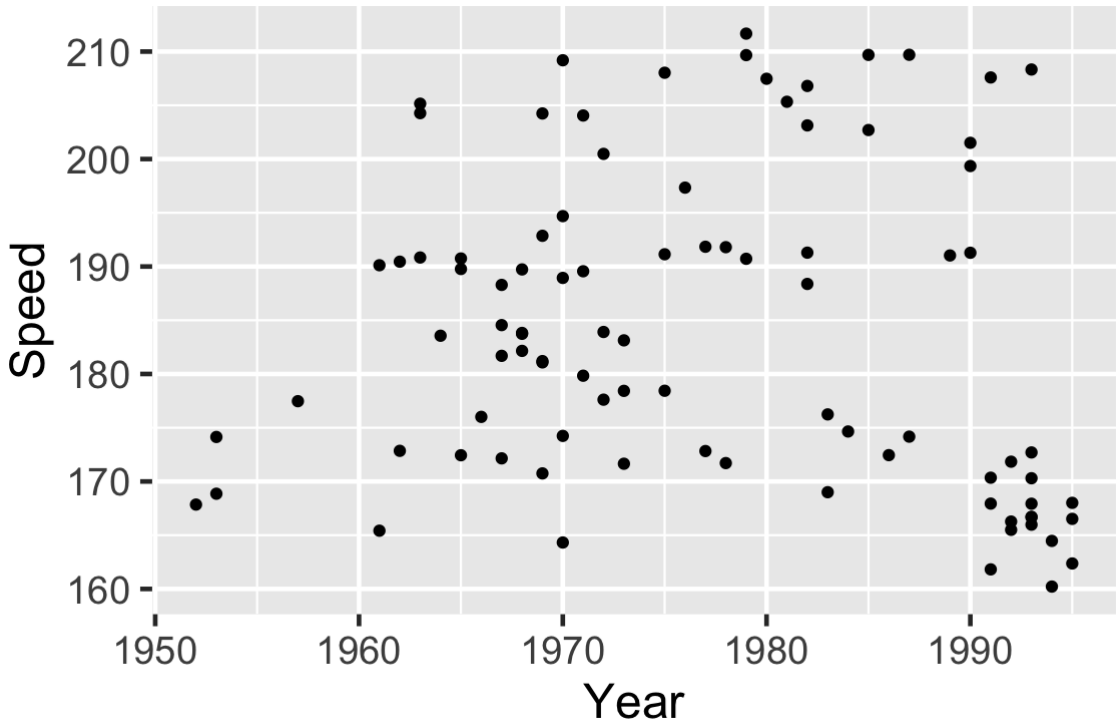
Log scale



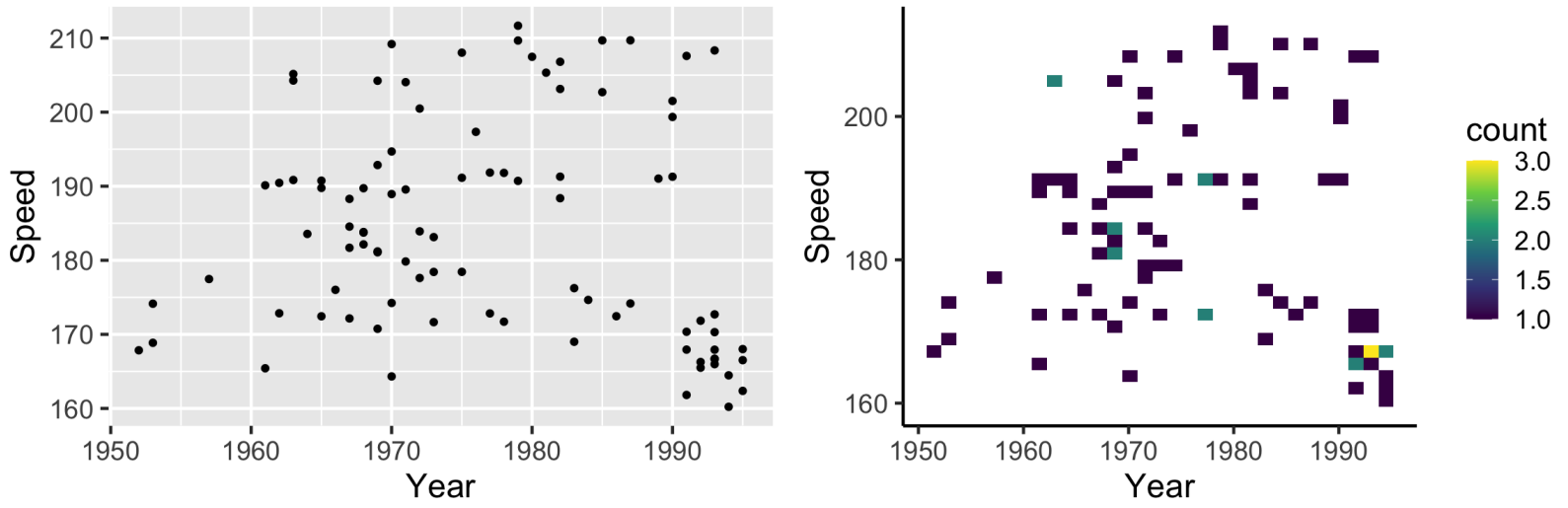
Alternatives to scatterplots

- Heatmaps (bin counts or density estimates)
- Density contour lines

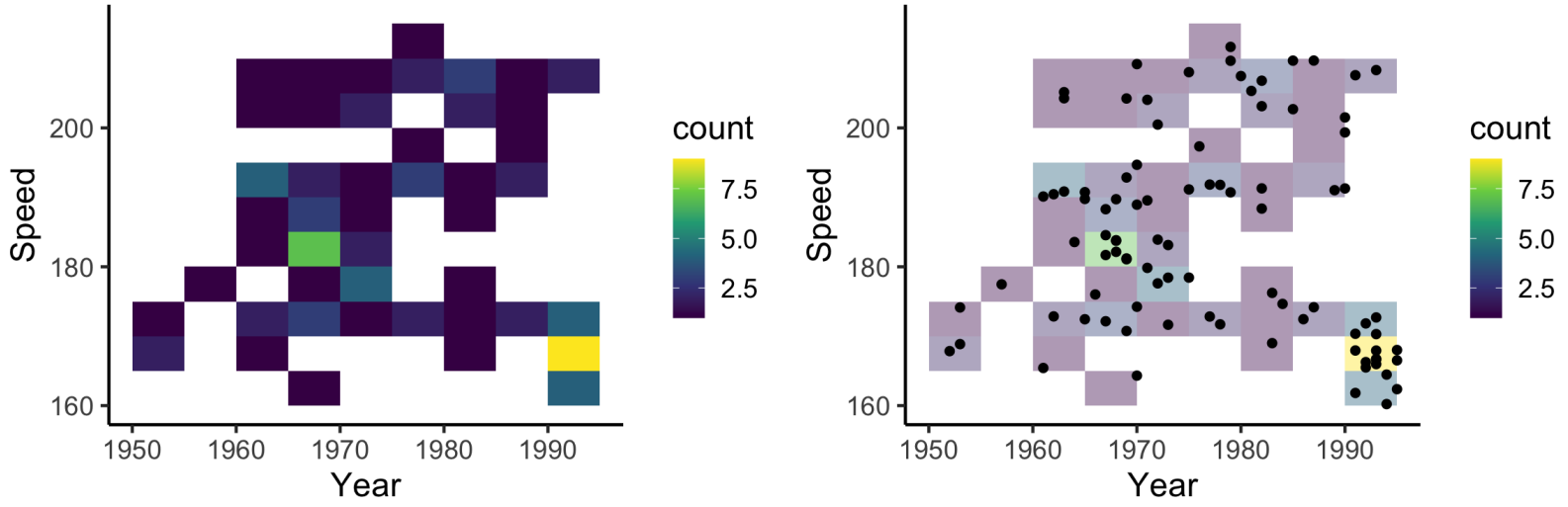
SpeedSki data (2011)



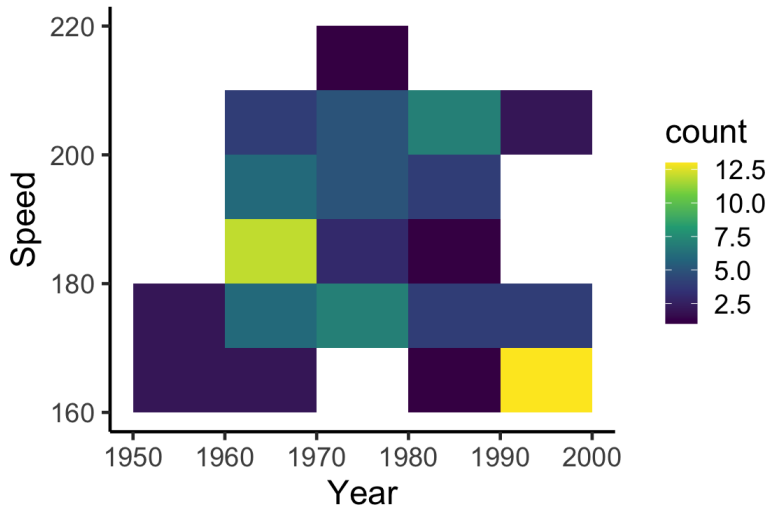
Square heatmap of bin counts (default: 30 bins)

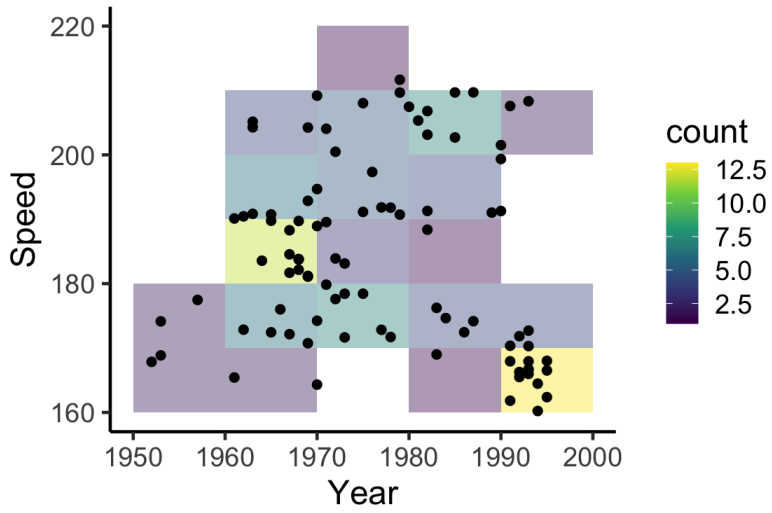


Square heatmap of bin counts (binwidth = 5)

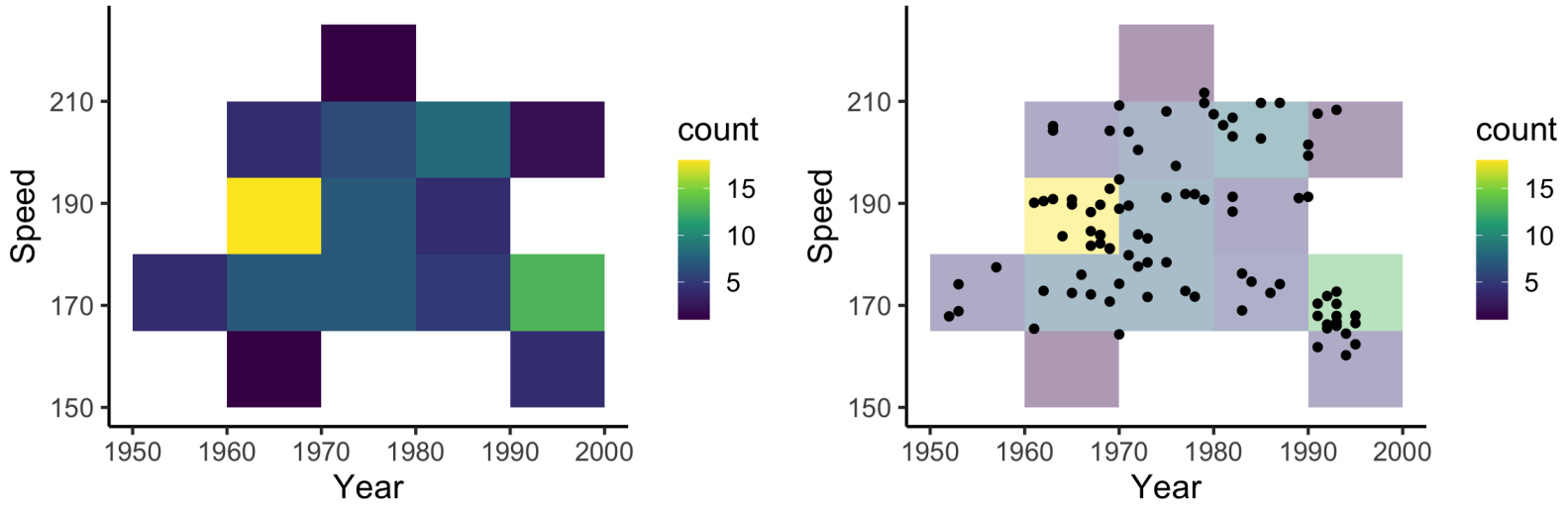


Square heatmap of bin counts (binwidth = 10)

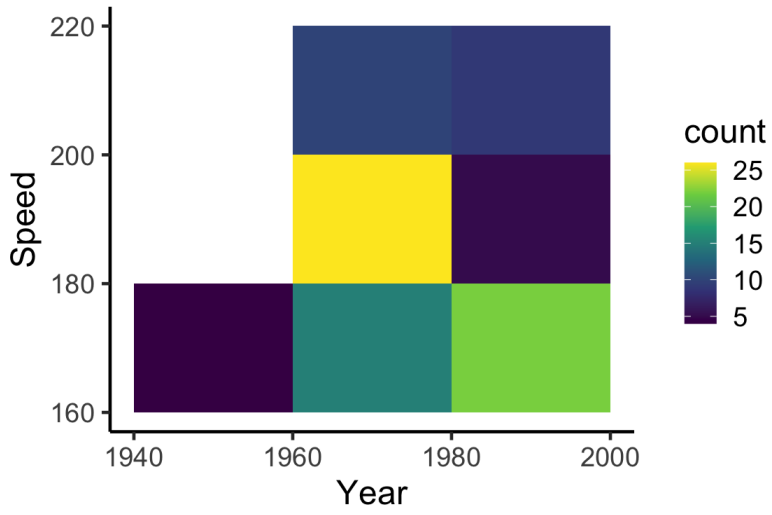


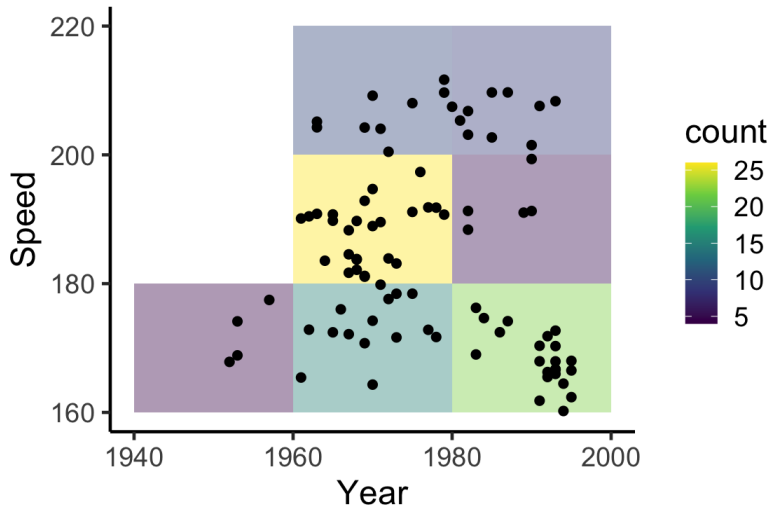


Square heatmap of bin counts (binwidth(x, y) = 10, 15)

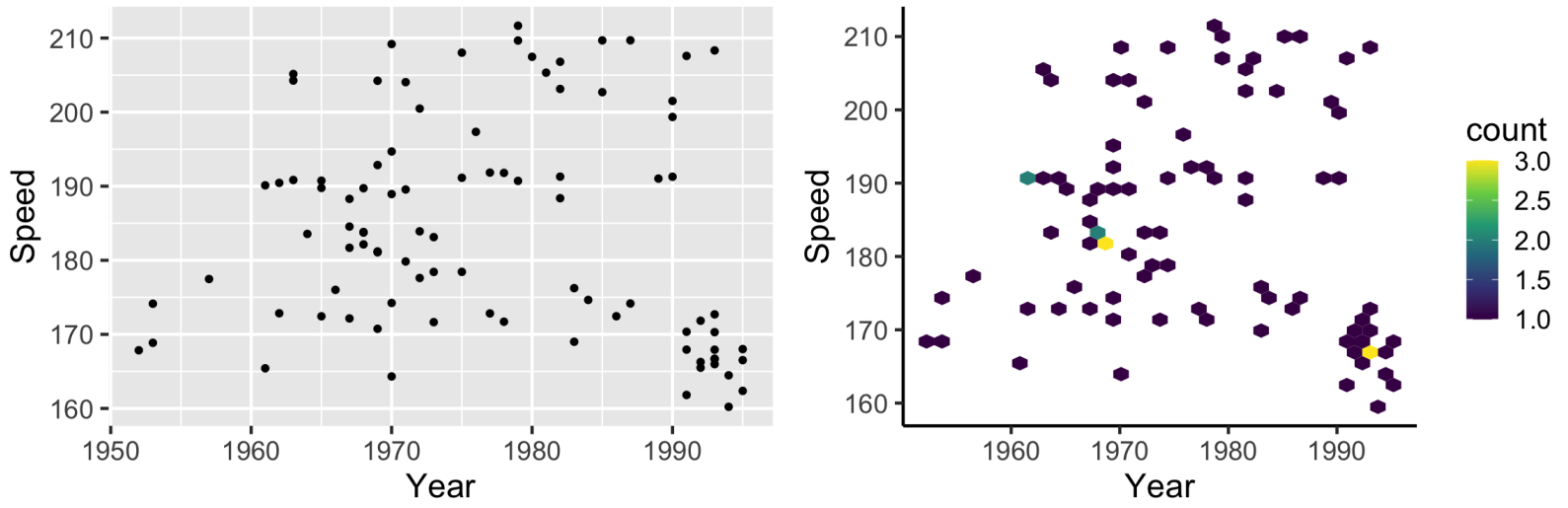


Square heatmap of bin counts (binwidth = 20)

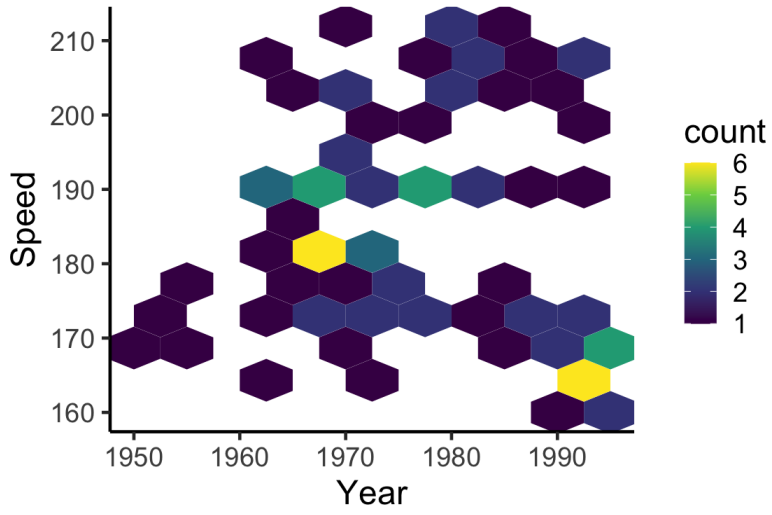


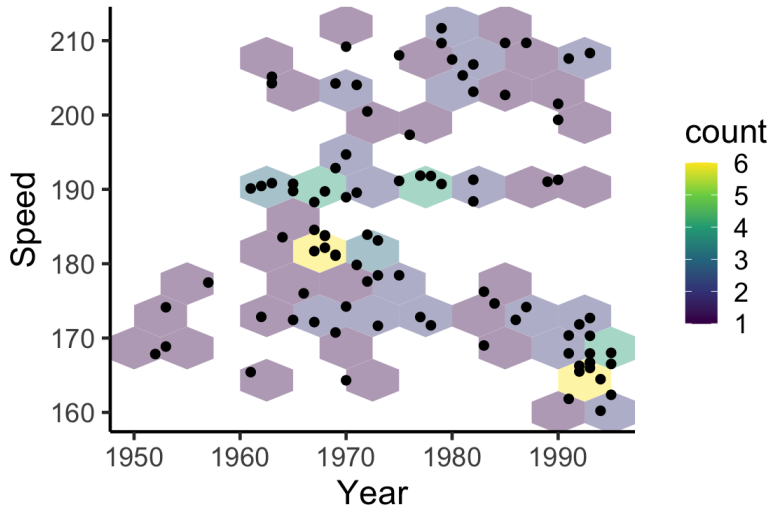


Hex heatmap of bin counts (default: 30 bins)

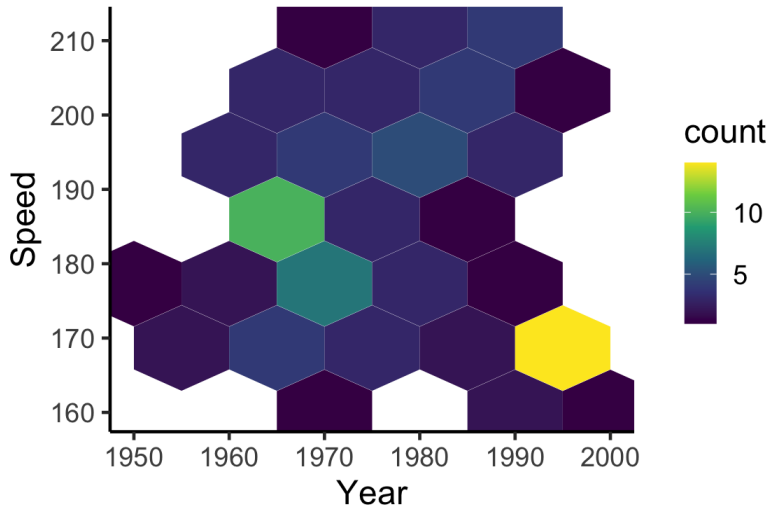


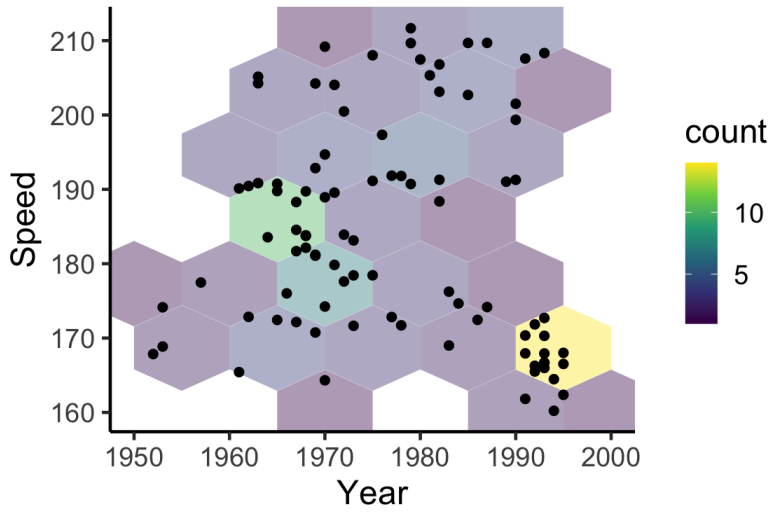
Hex heatmap of bin counts (binwidth = 5)



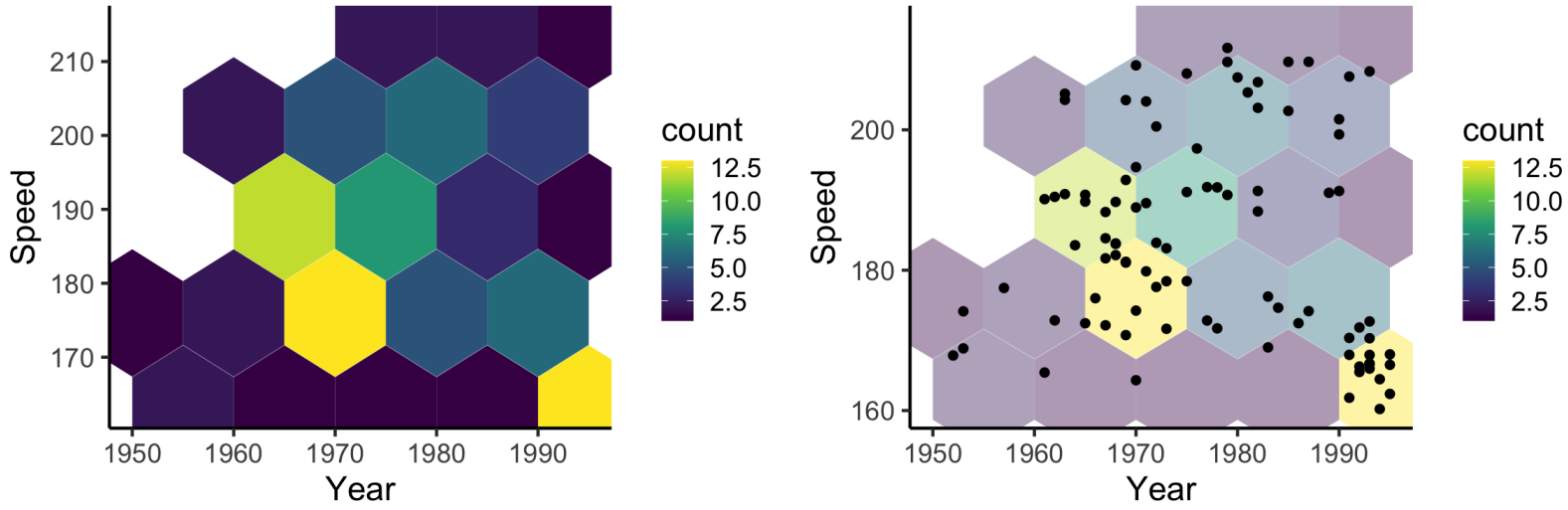


Hex heatmap of bin counts (binwidth = 10)

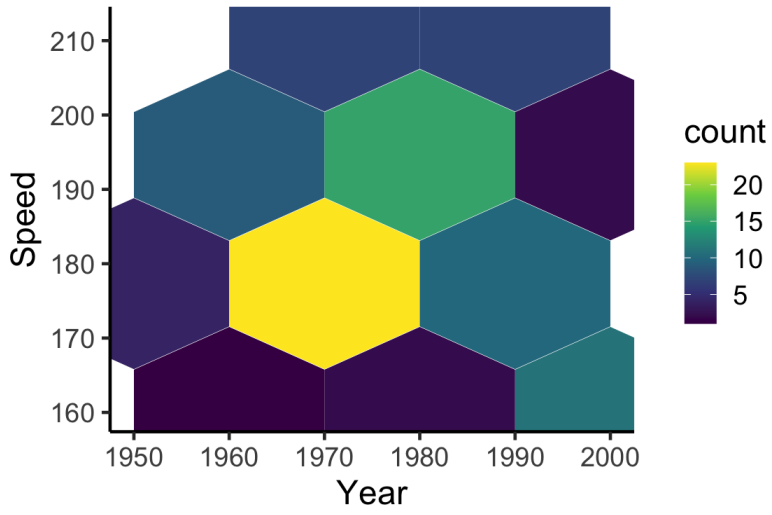


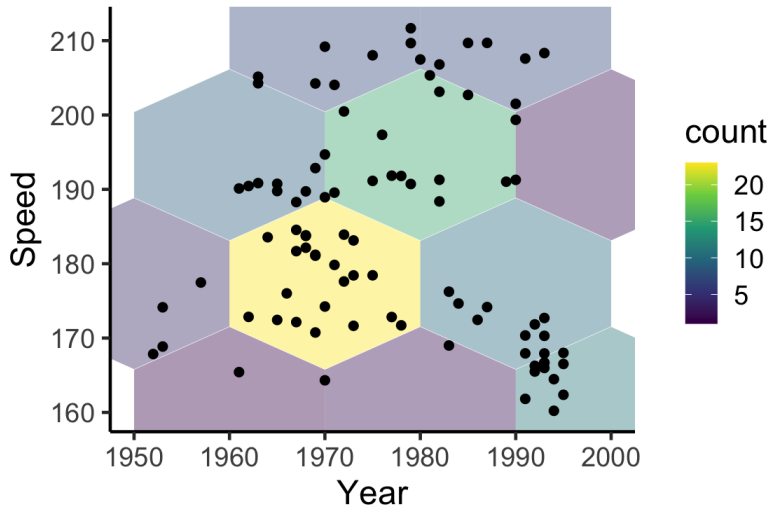


Hex heatmap of bin counts (binwidth(x, y) = 10, 15)

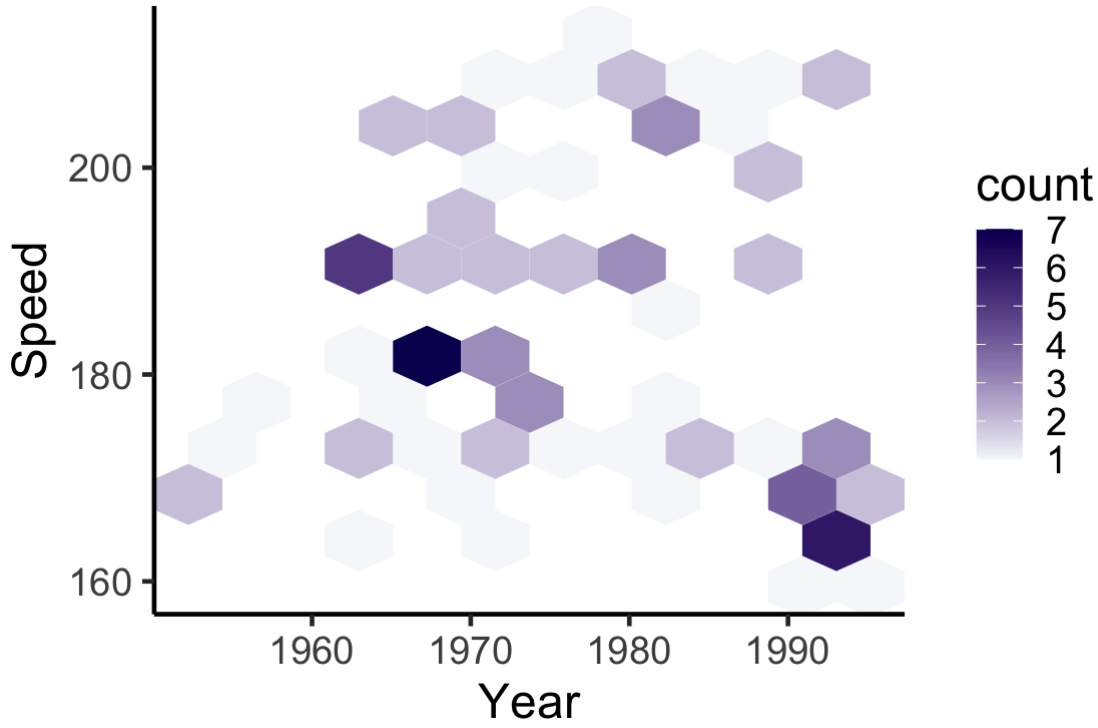


Hex heatmap of bin counts (binwidth = 20)



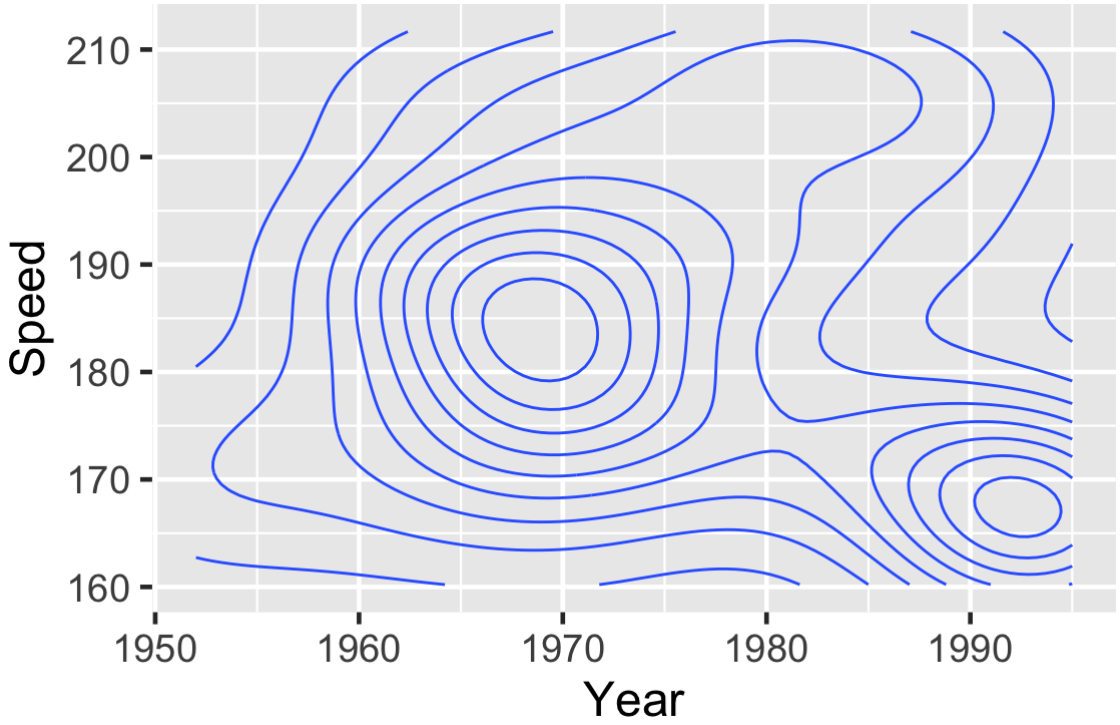


Alternative (better) approach to color

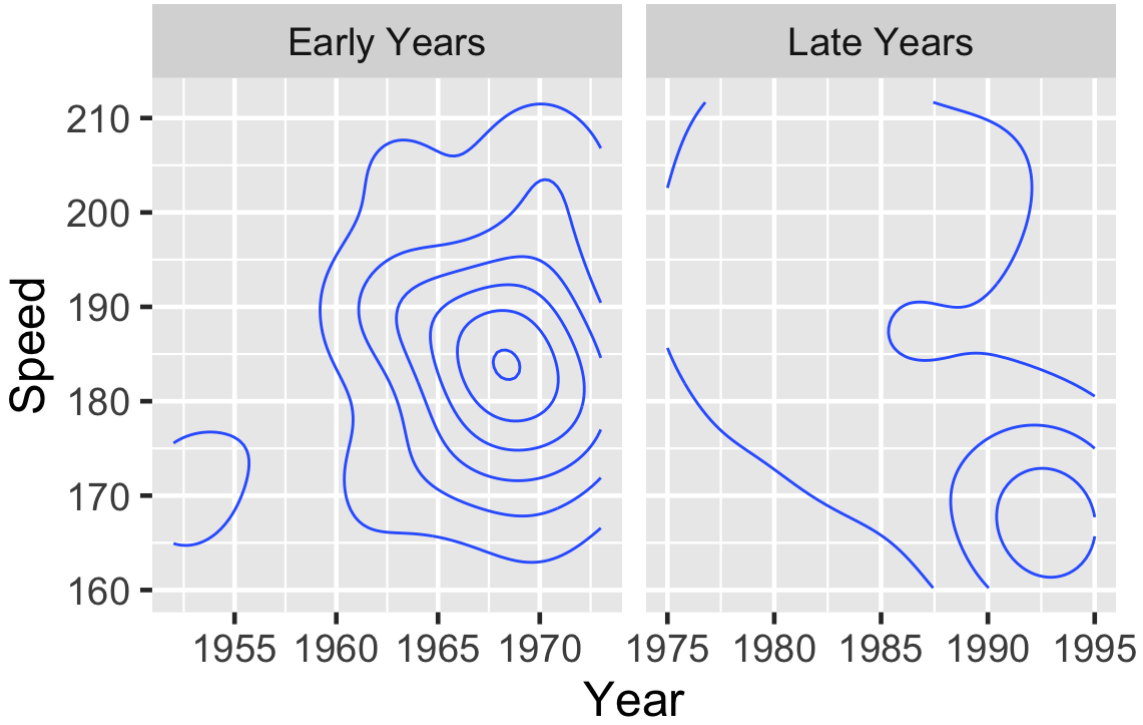


Density estimate contour lines

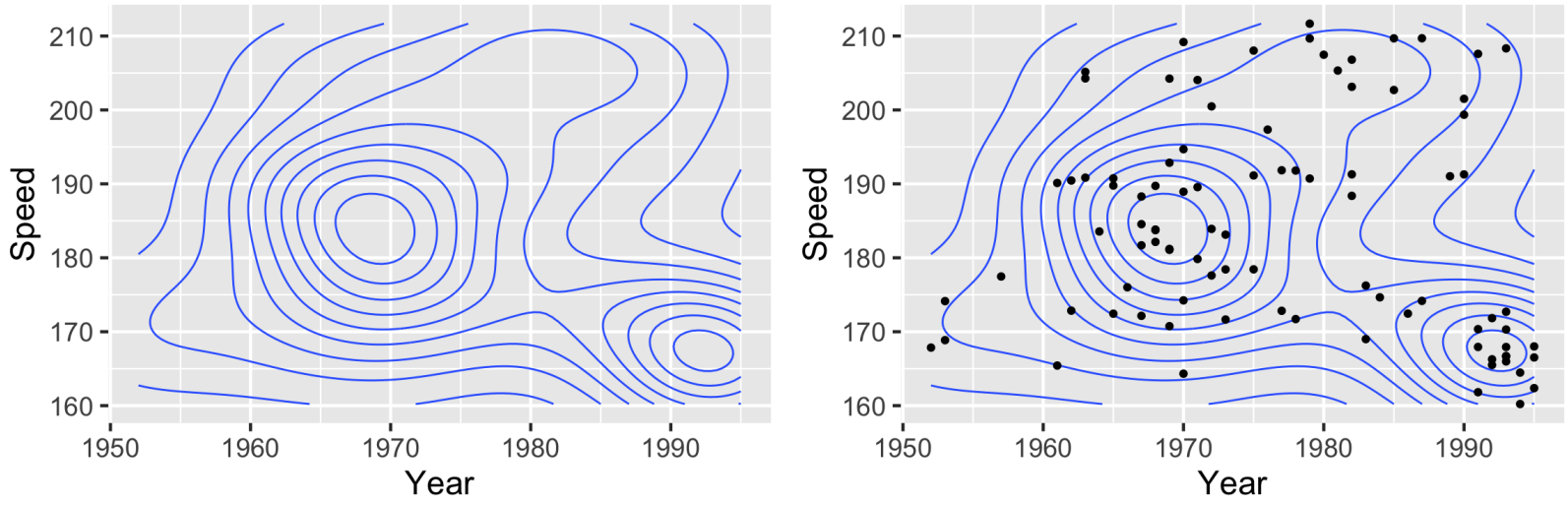
Uses 2D kernel density estimate from `MASS` package



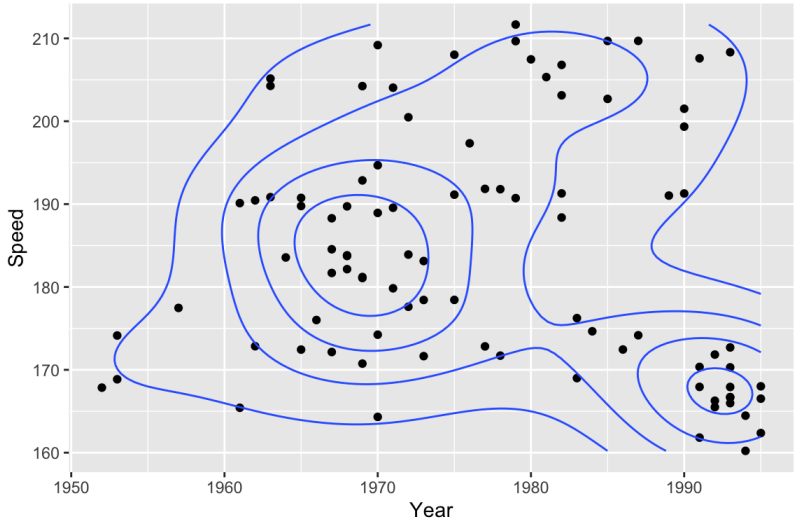
Watch out for boundaries



Density estimate contour lines



Density estimate contour lines (change # of bins)



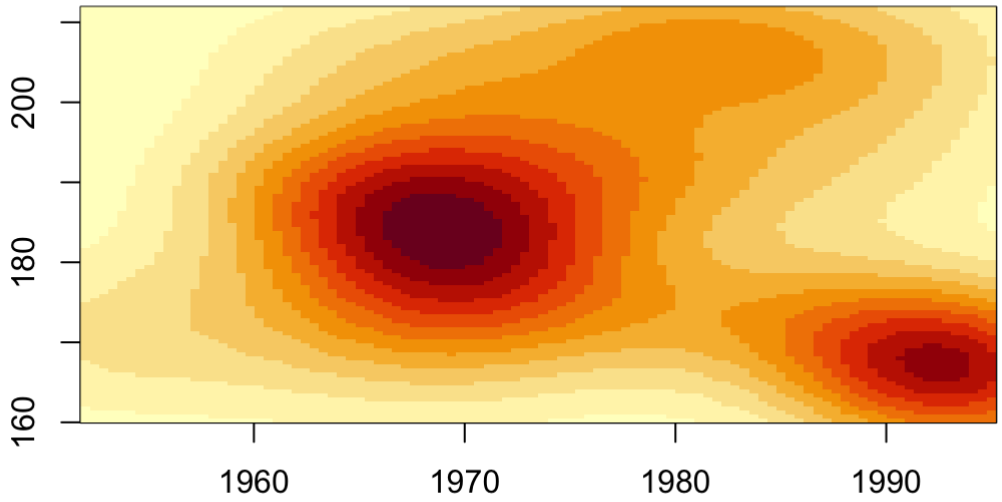
Base graphics

```
library(MASS)
f1 <- kde2d(SpeedSki$Year, SpeedSki$Speed, n = 100)
str(f1)
```

```
## List of 3
##  $ x: num [1:100] 1952 1952 1953 1953 1954 ...
##  $ y: num [1:100] 160 161 161 162 162 ...
##  $ z: num [1:100, 1:100] 0.000063 0.000065 0.0000667 0.0000683 0.0000697 ...
```

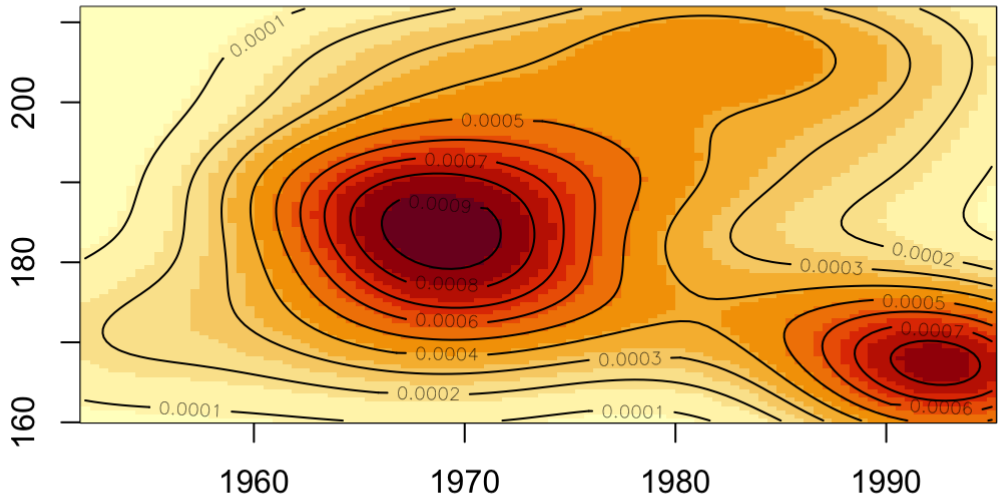
2D kernel density estimate

```
image(f1)
```



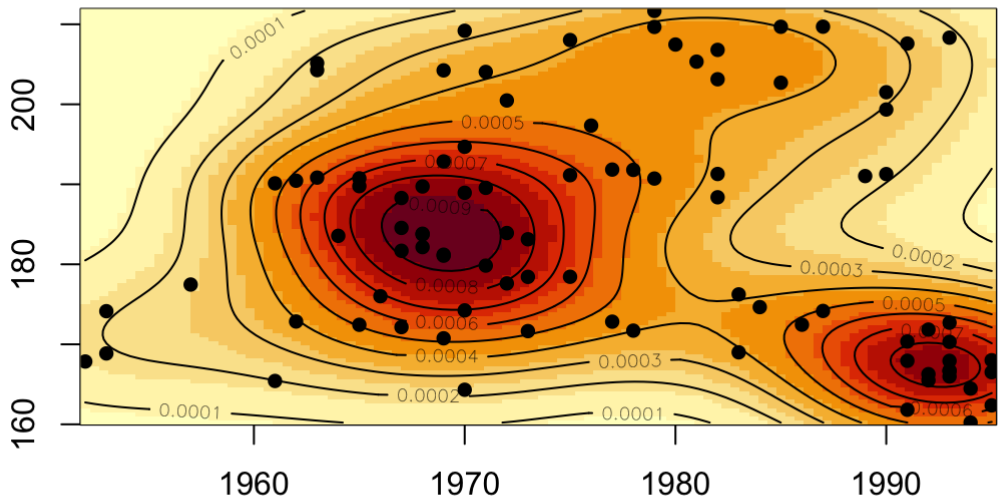
2D kernel density estimate w/ contour lines

```
image(f1)
contour(f1, add = T)
```



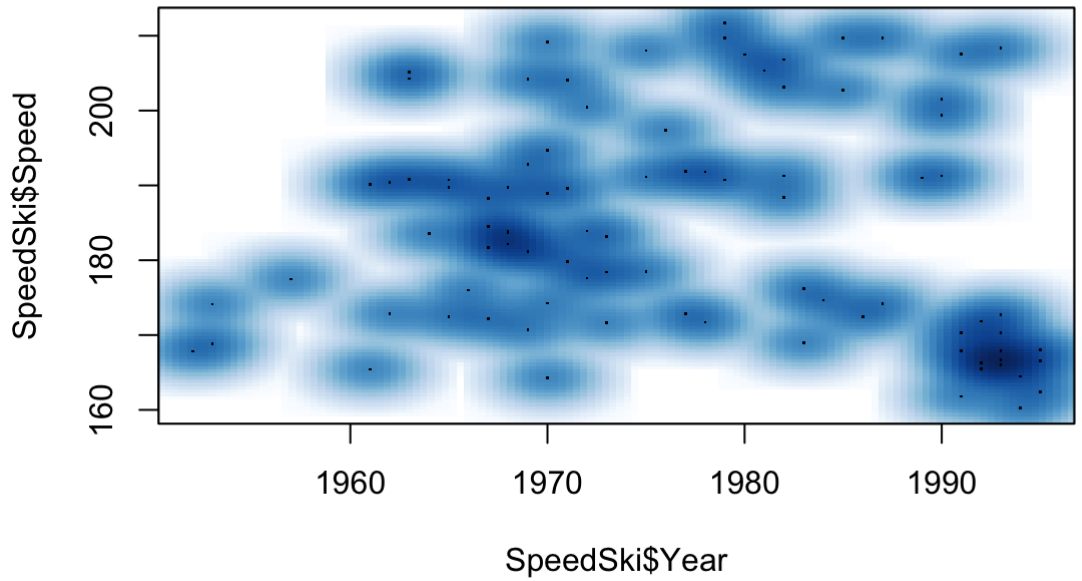
2D kernel density estimate w/ contour lines & points

```
image(f1)
contour(f1, add = T)
points(SpeedSki$Year, SpeedSki$Speed, pch = 16)
```



Another 2D kernel density estimation

```
smoothScatter(SpeedSki$Year, SpeedSki$Speed)
```

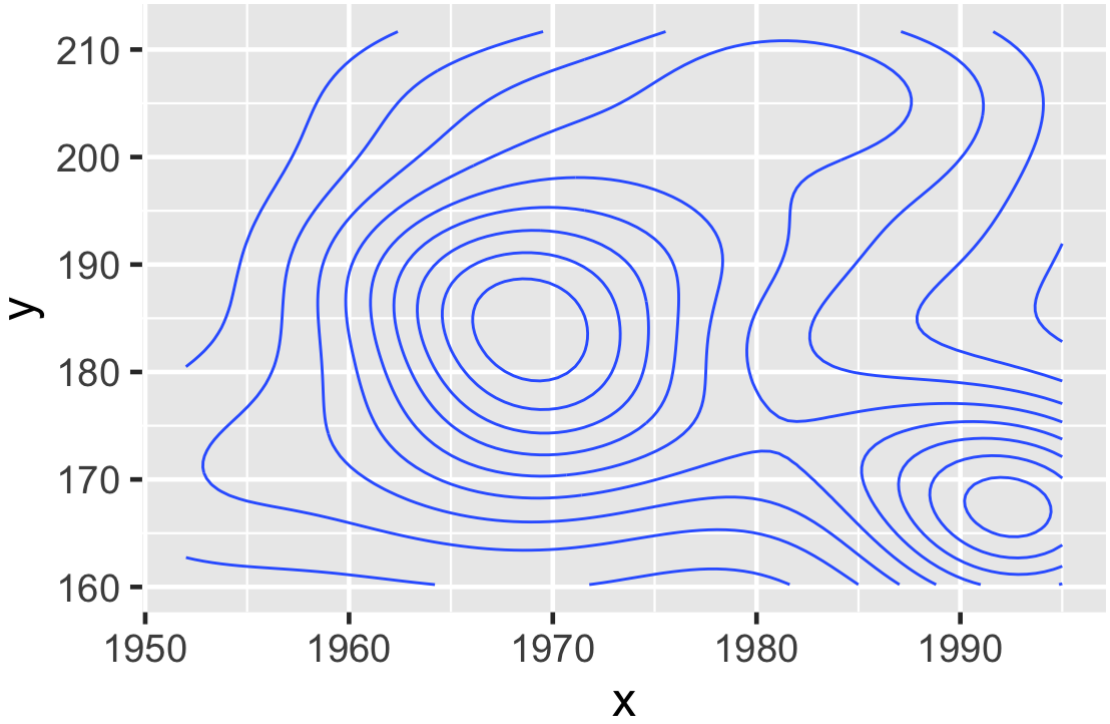


Calculate the kde, plot with ggplot2

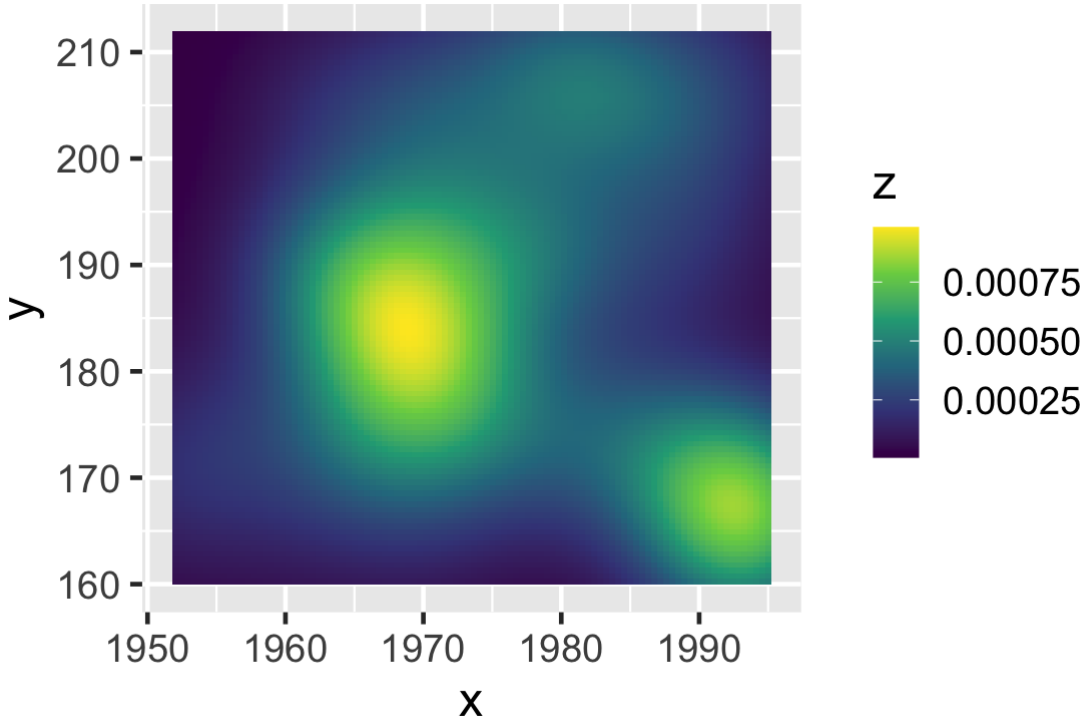
```
df <- con2tr(f1)
head(df)
```

```
##      x   y         z
## 1 1952 160 0.0000630
## 2 1952 160 0.0000650
## 3 1953 160 0.0000667
## 4 1953 160 0.0000683
## 5 1954 160 0.0000697
## 6 1954 160 0.0000709
```

```
ggplot(df, aes(x, y)) + geom_contour(aes(z = z)) +
  theme_grey(18)
```

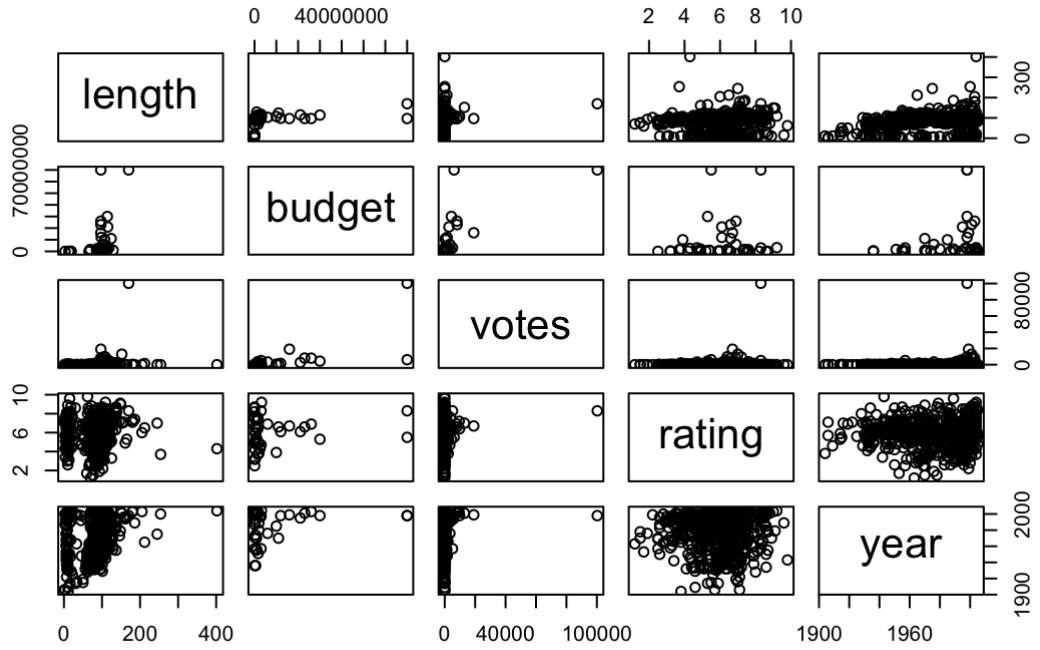



```
ggplot(df, aes(x, y)) + geom_tile(aes(fill = z)) +
  scale_fill_viridis_c() +
  theme_grey(18)
```

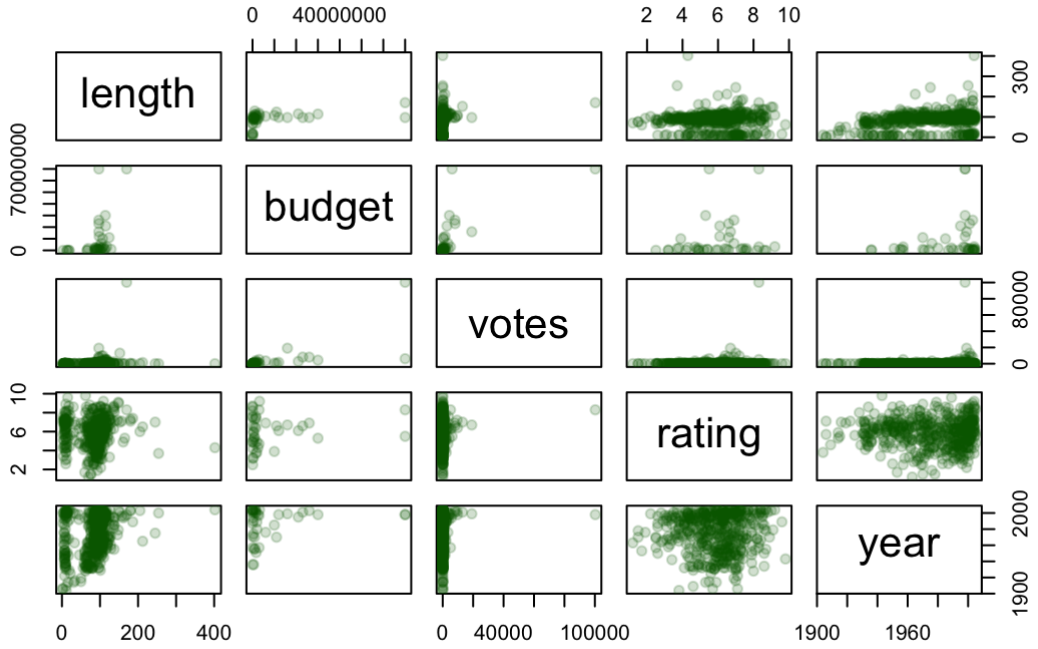

210
200
190
180
170
160
y
1950
1960
1970
1980
1990
x
z
0.00075
0.00050
0.00025

Scatterplot matrices for multiple variables

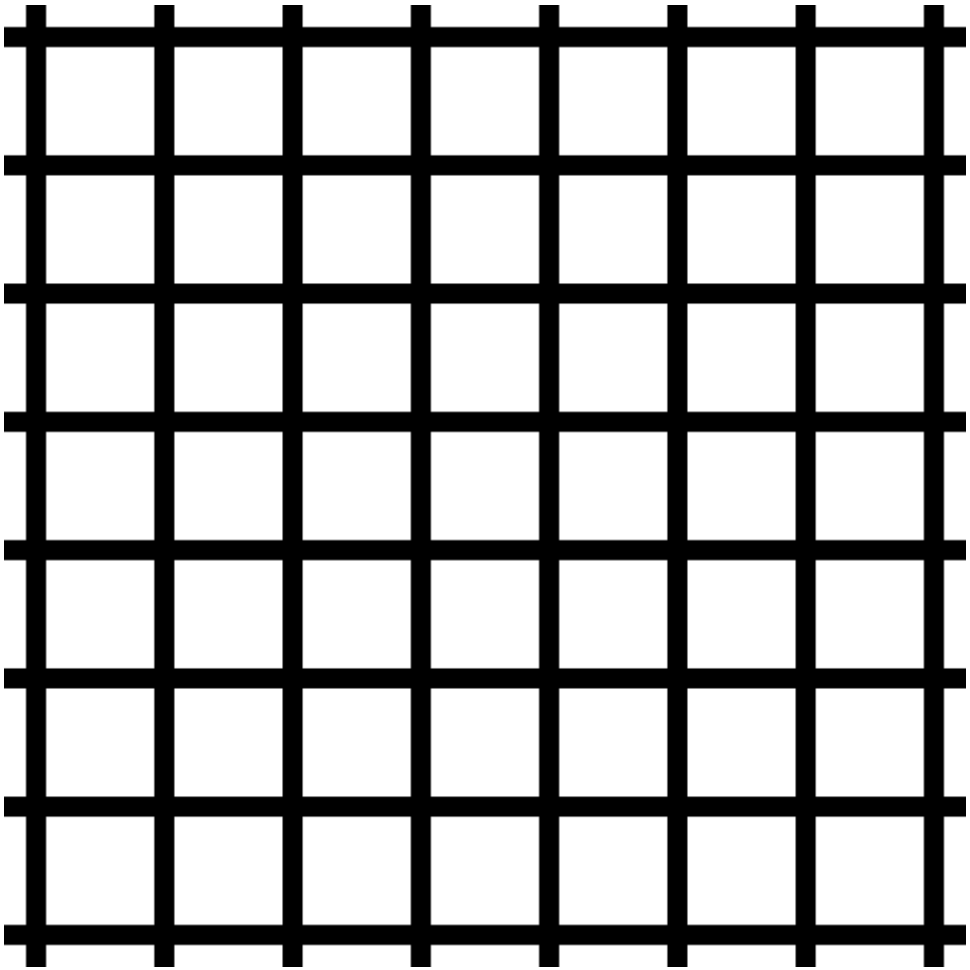
(random sample of 500 rows)



Add alpha blending



Hermann grid illusion



Scatterplot matrix (lattice)