Big data visualization

R for Data Science

Juan R Gonzalez
juanr.gonzalez@isglobal.org

BRGE - Bioinformatics Research Group in Epidemiology
ISGlobal - Barcelona Institute for Global Health
http://brge.isglobal.org

Goal

- Suport exploratory analysis large data sets in R
- Efficient (1,000 x 100,000,000)
- Fast (100,000,000 less than 5s)

bigvis package

- Install package from Hadley's repository

```
devtools::install_github("hadley/bigvis")
```

- Main manuscript: http://vita.had.co.nz/papers/bigvis.pdf

Process

- Condense (bin & summarise) [`bin()`, `condense()`]
- Smooth [`smooth()`, `best_h()`, `peel()`]
- Visualise [`autoplot()` and standard functions]

Condense

- bin:

$$\lfloor \frac{x - origin}{width} \rfloor$$

- summarize:
  - count: histogram
  - mean: regression, loess
  - quantiles: boxplots, quantile regression

Smooth

- Fix over binning
- Dampen effect of outliers
- Focus on main trends
- Limitation: best bandwith

Example

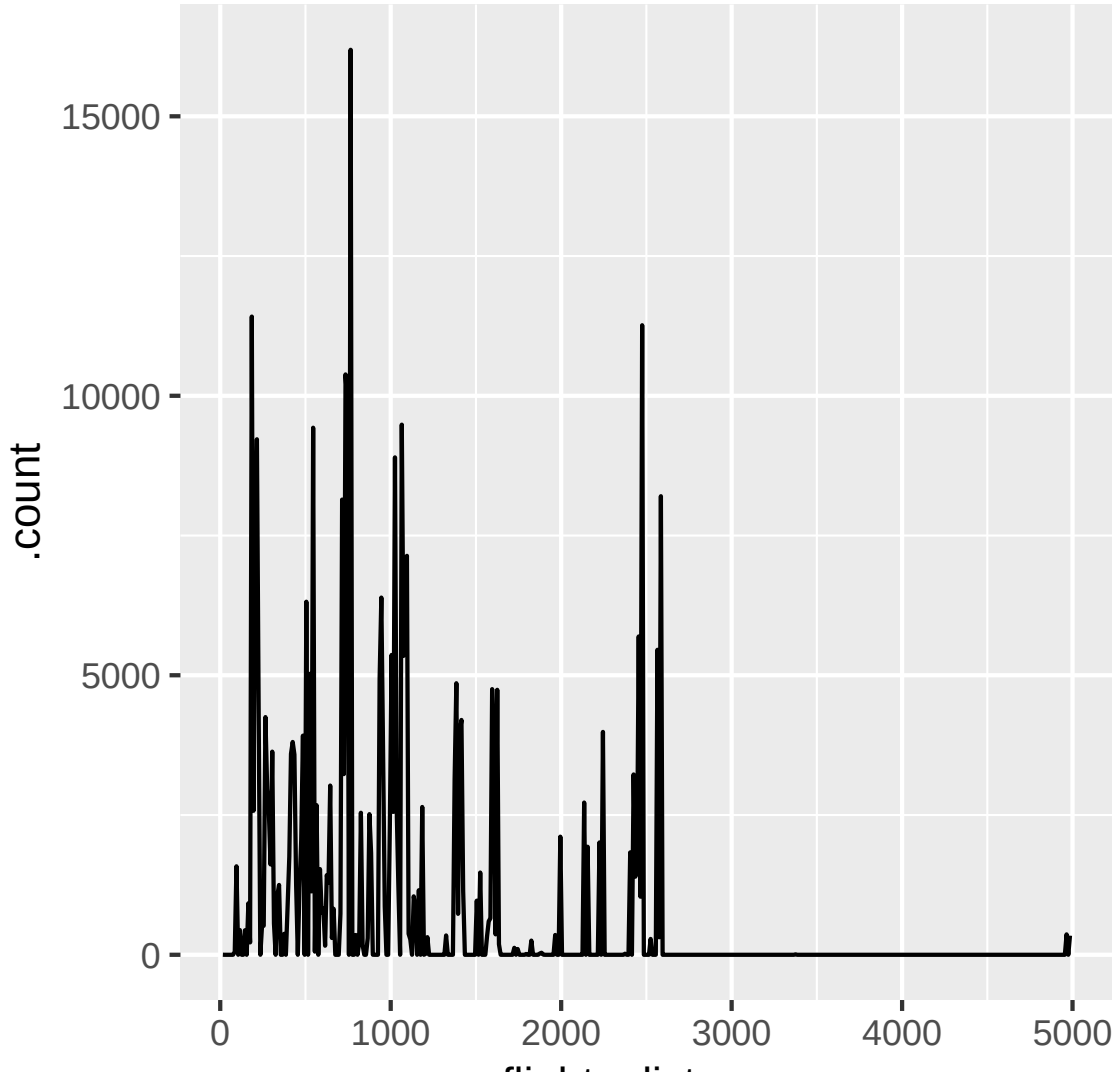
Open this link

http://shiny.rstudio.com/gallery/faithful.html

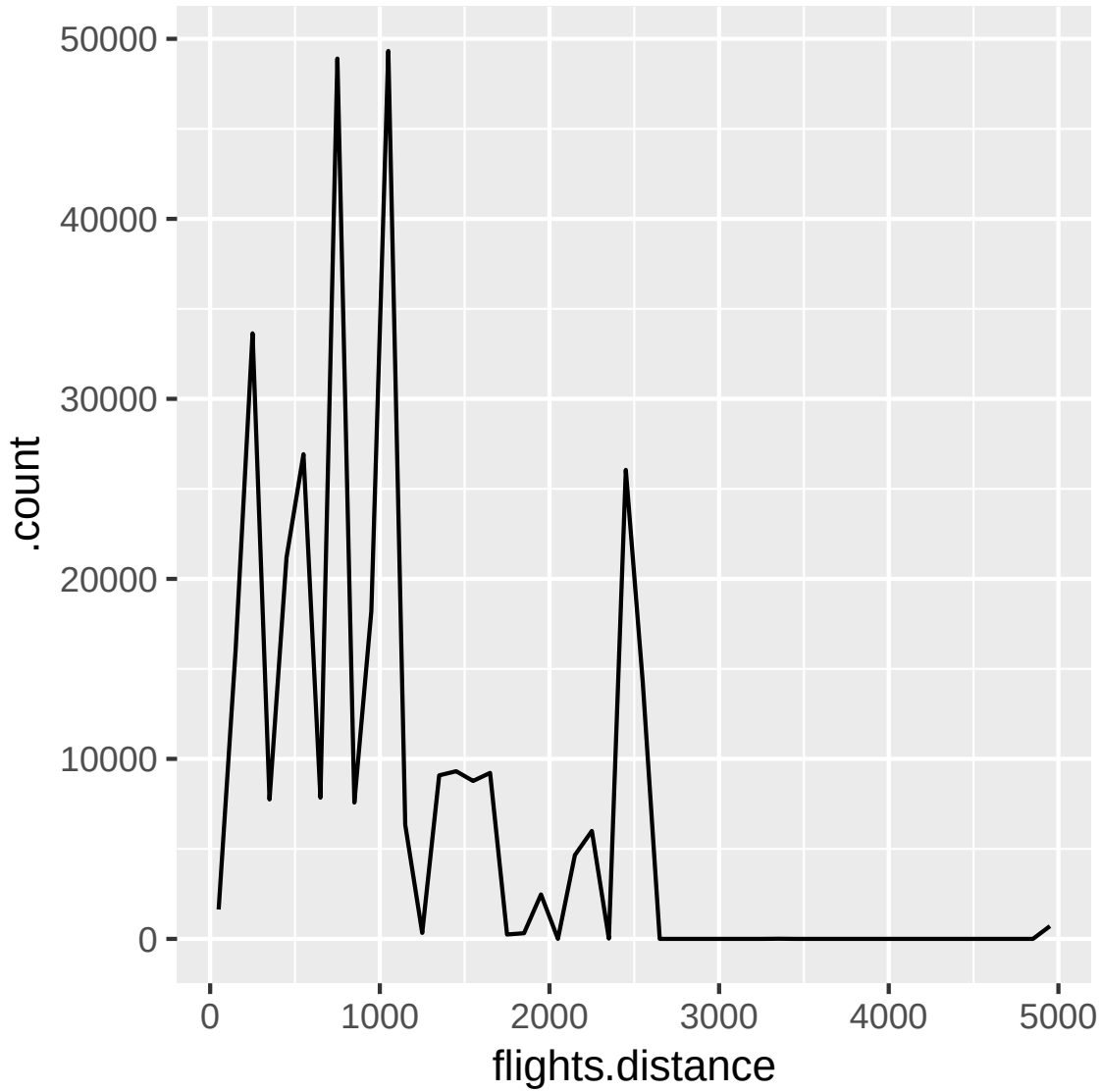
and play by changing the bandwith size and smoothing parameter to see how data visualization is changing

Visualizing one variable

```
library(bigvis);library(tidyverse);library(nycflights13)
dist_s <- condense(bin(flights$distance, 10))
autoplot(dist_s)
```

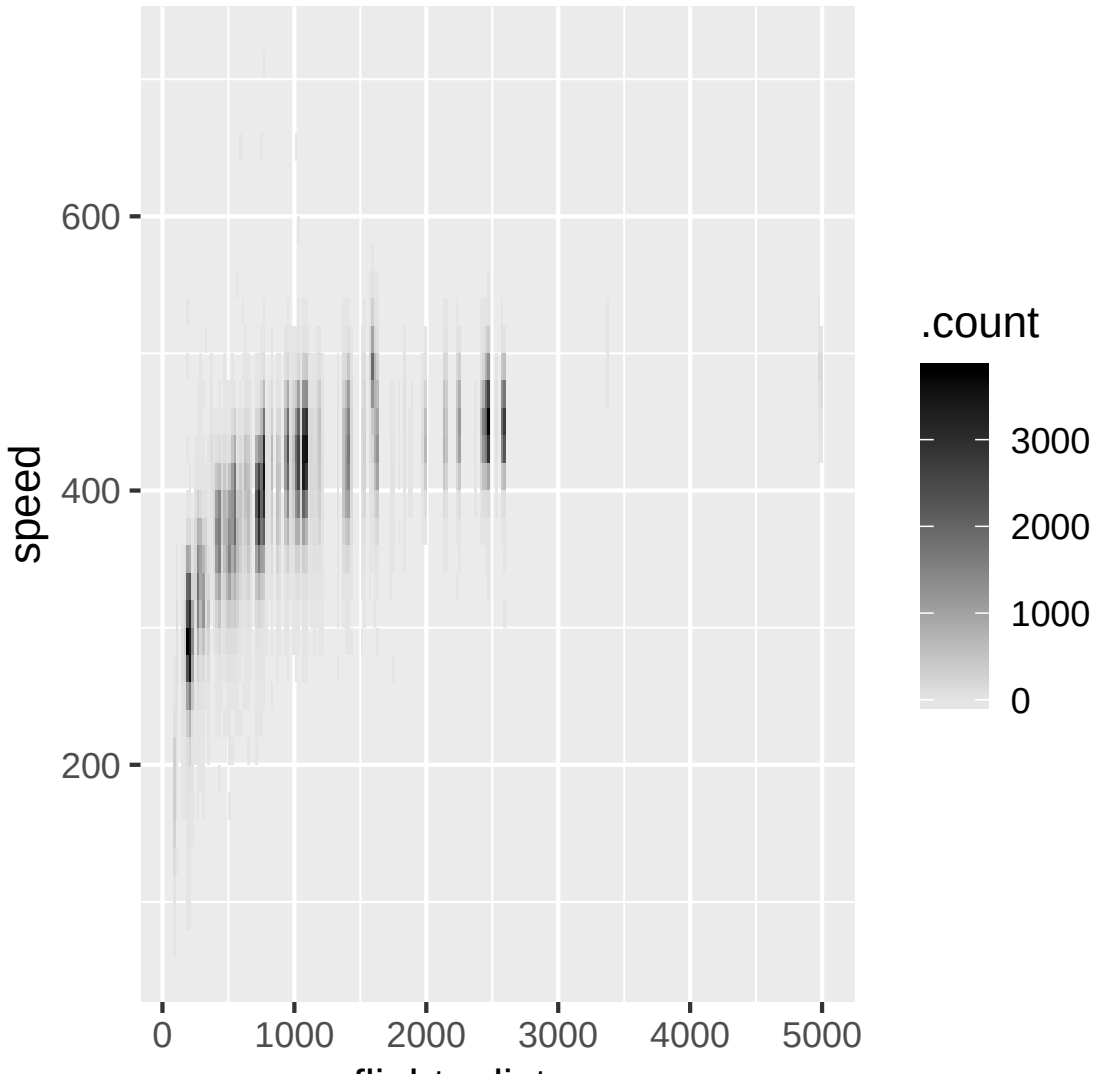


```
dist_s <- condense(bin(flights$distance, 100))
autoplot(dist_s)
```

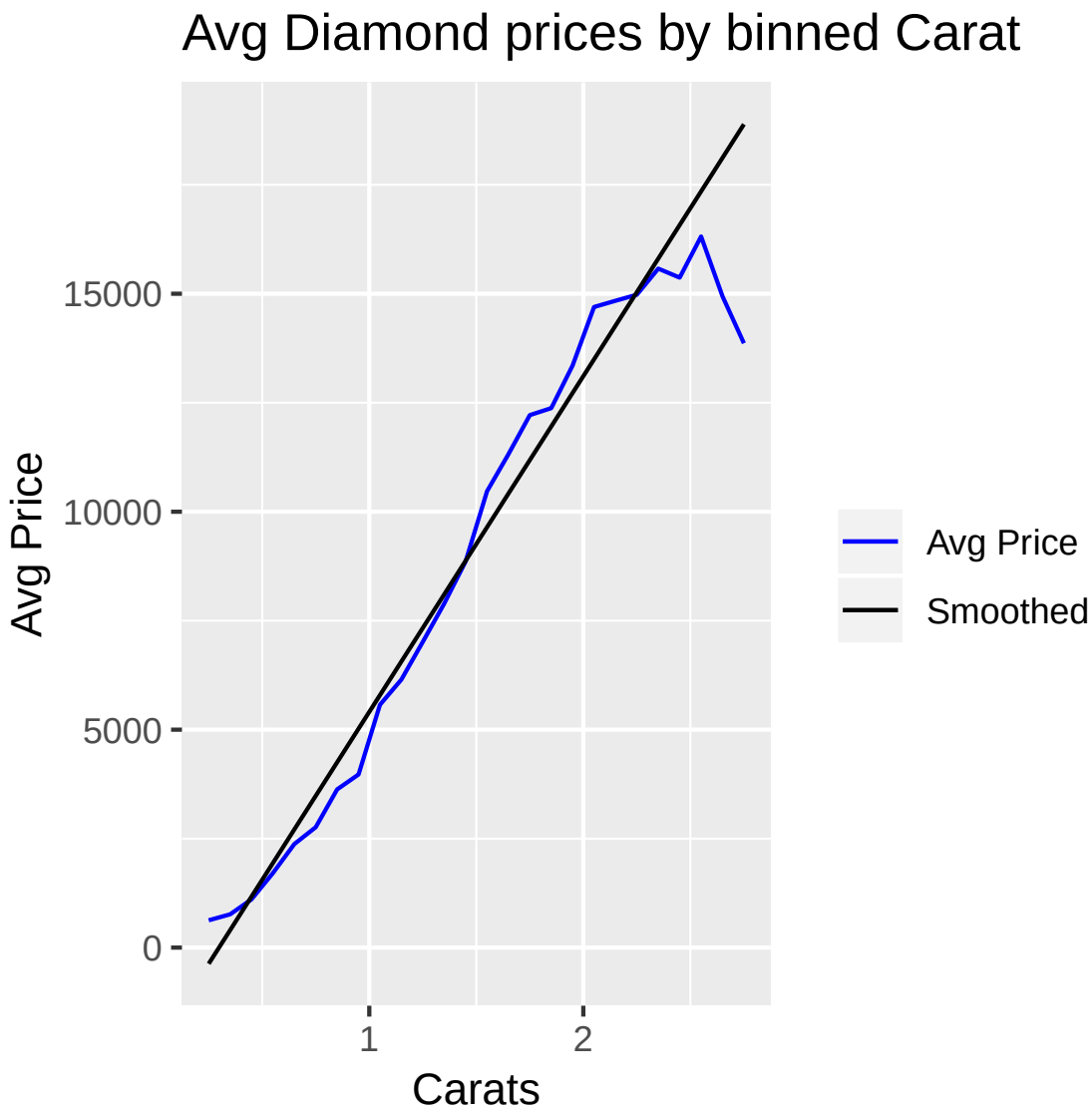


Visualizing two variables

```
speed <- with(flights, distance / air_time * 60)
sd2 <- condense(bin(flights$distance, 20), bin(speed, 20))
autoplot(sd2)
```



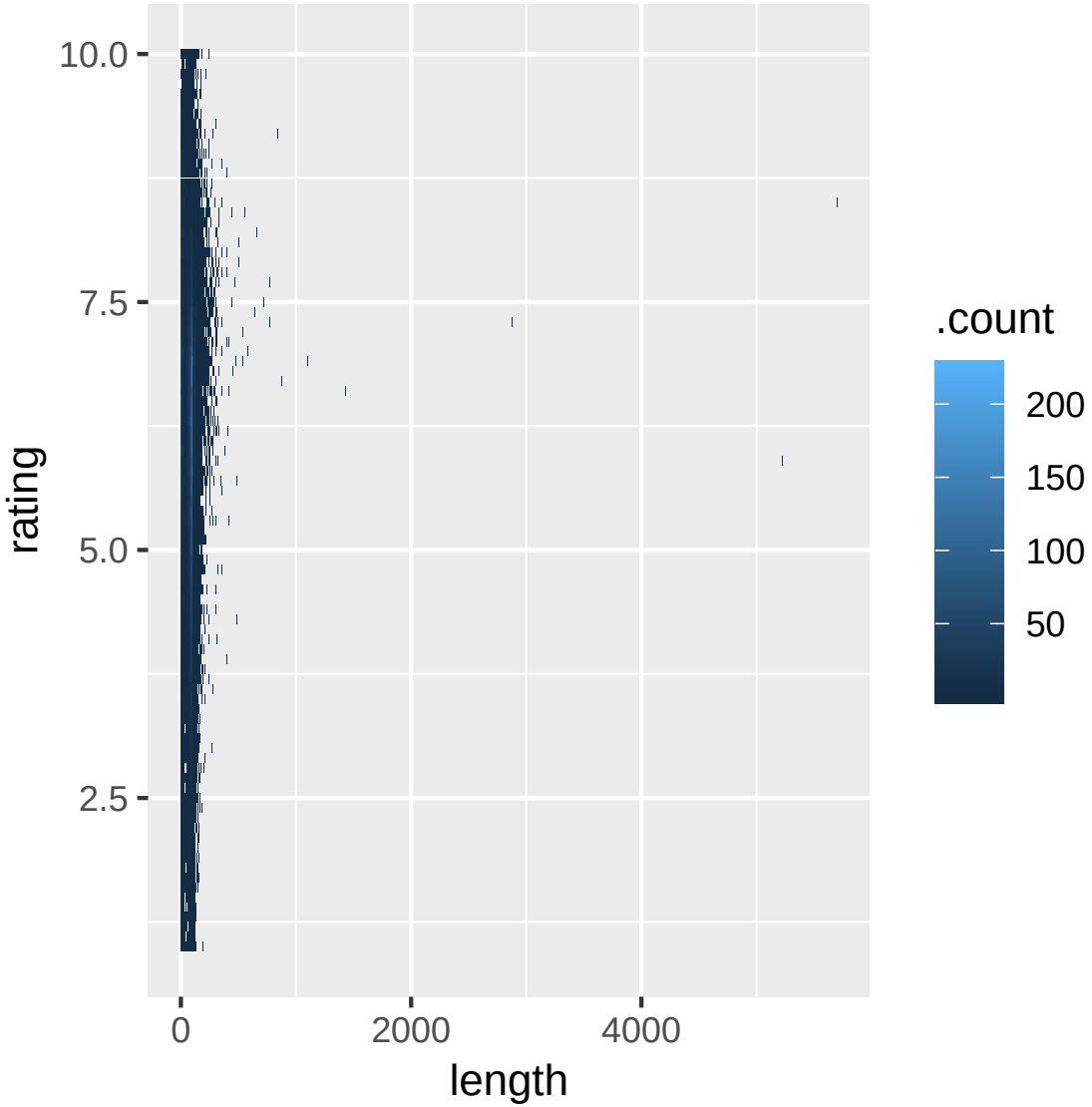
```r
# subset the diamonds data
mydiamonds <- subset(diamonds, carat < 2.75)

# condense avg price based on bins of carat sizes
# of .1 carat intervals
myd <- condense(bin(mydiamonds$carat, .1),
                z=mydiamonds$price, summary="mean")

# smooth out the trend
myds <- smooth(myd, 50, var=".mean", type="robust")

# plot the orginal binned prices vs the smoothed trend line
ggplot() + geom_line(data=myd, aes(x=mydiamonds.carat,
                                   y=.mean, colour="Avg Price"))
           geom_line(data=myds, aes(x=mydiamonds.carat,
                                    y=.mean, colour="Smoothed"))
           ggtitle("Avg Diamond prices by binned Carat") +
           ylab("Avg Price") +
           xlab("Carats") +
scale_colour_manual("", breaks=c("Avg Price","Smoothed"),
                    values=c("blue", "black"))
```


Avg Diamond prices by binned Carat
Avg Price
15000
10000
5000
0
1
2
Carats
Avg Price
Smoothed

Outliers

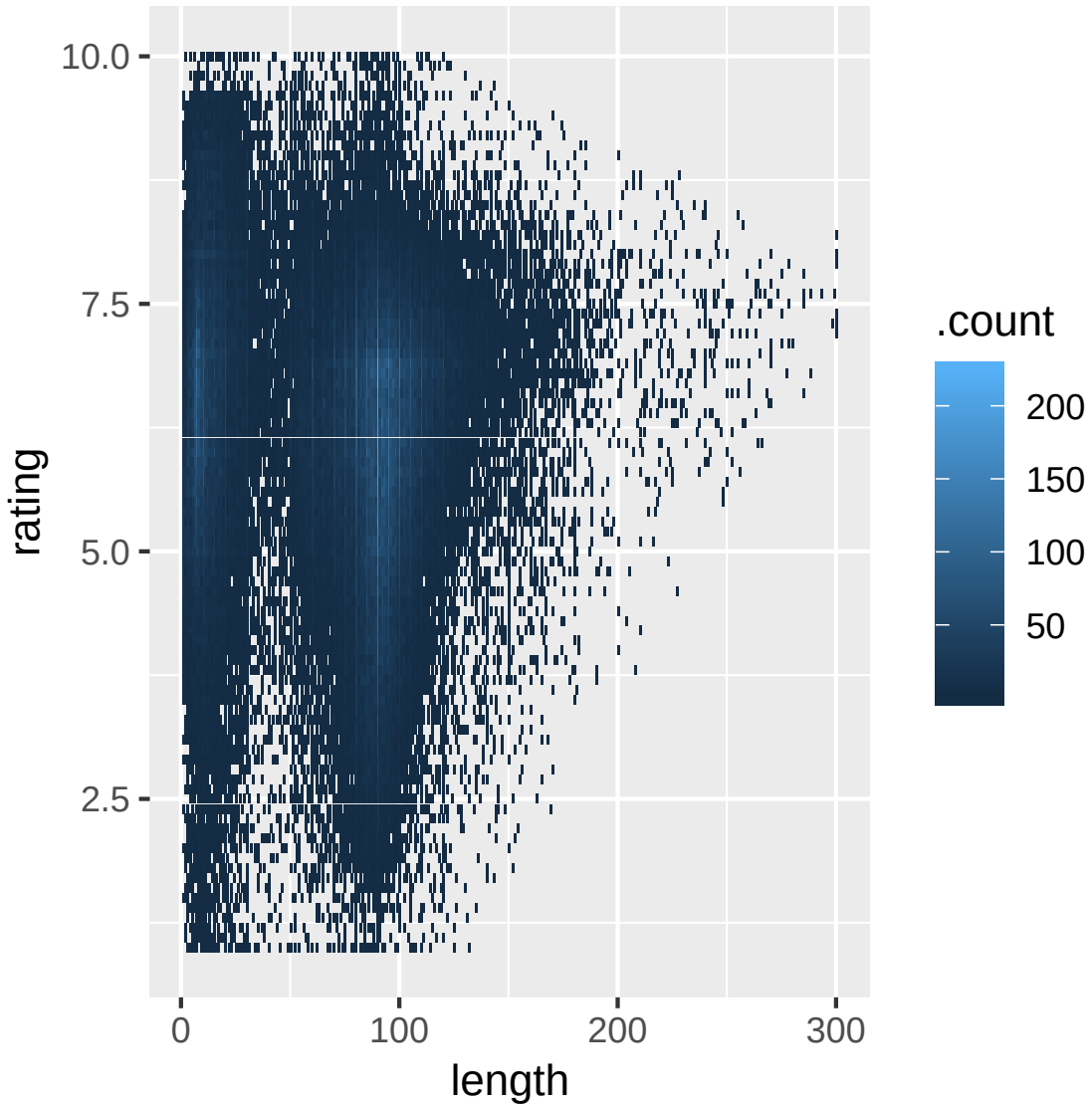
Let us analyze another big data set. `bigvis::movies` contains information about 130K movies from http://imdb.com/. It includes data about length of the movie (`length`) and had been rated (`rating`) by at least one imdb user. Let's start by plotting the length of the movie and their rating

```
Nbin <- 1e4 # number of bins
binData <- with(movies,
                condense(bin(length, find_width(length, Nbin)),
                         bin(rating, find_width(rating, Nbin))))
ggplot(binData, aes(length, rating, fill = .count)) +
  geom_tile()
```


10.0
7.5
5.0
2.5
rating
0
2000
4000
length
.count
200
150
100
50

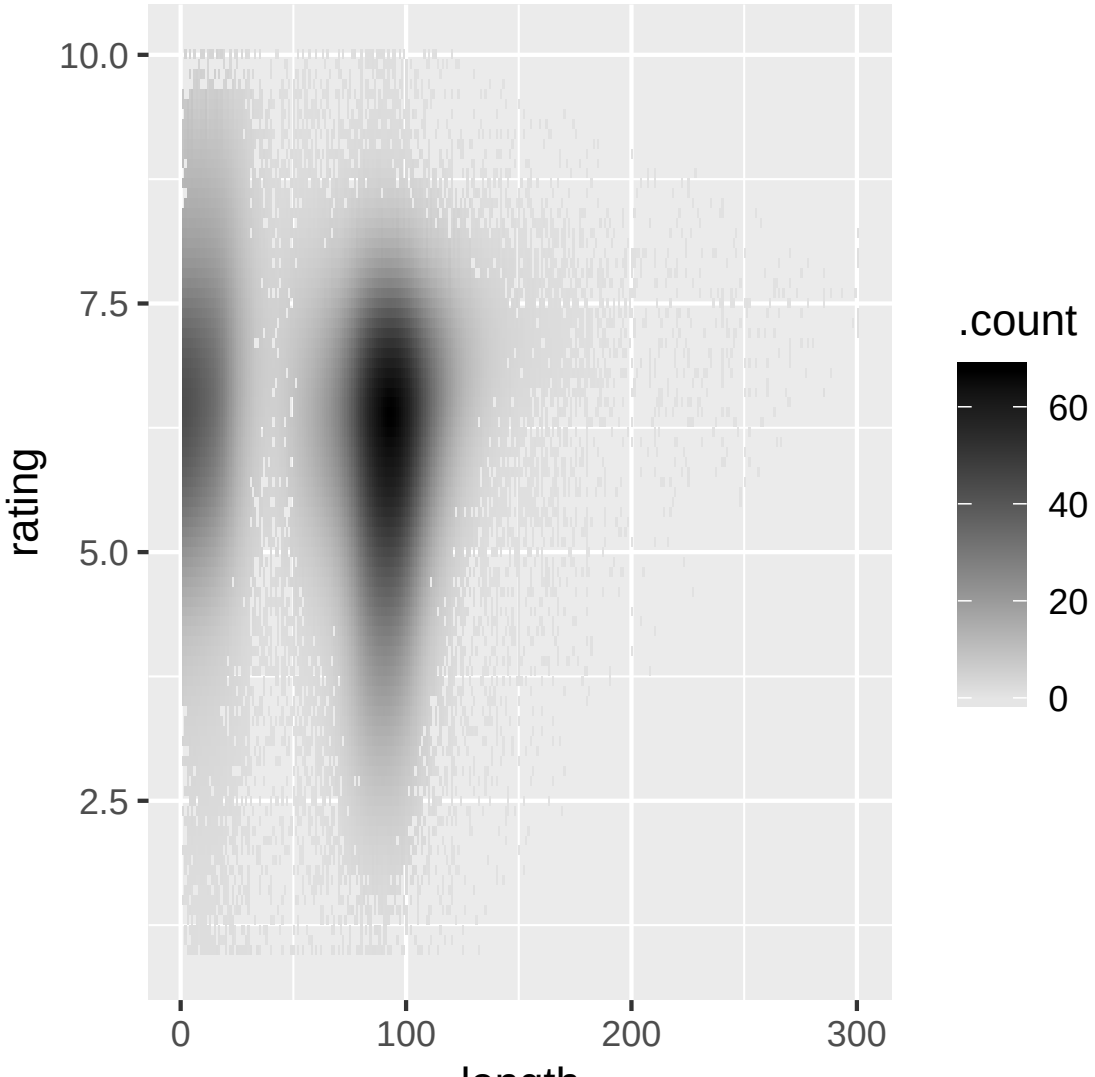
Let's remove outliers to improve visualization. It can be done by using `peel` function:

```
last_plot() %+% peel(binData) # same plot, different data
```



Visualization can be improved by smoothing with different bandwith (e.g. `h`) for `length` (h=20) and `rating` (h=1)

```
smoothBinData <- smooth(peel(binData), h=c(20, 1))
autoplot(smoothBinData)
```



Key messages

- Preprocessing to generate statistical summaries is the key to visualizing Big Data
- Big data means very rare cases can occur. This implies outliers may be more of a problem
- Smoothing is very important to highlight trends & suppress noise
- The R `bigvis` package is a very powerful tool for plotting large datasets. It includes features to strip outliers, smooth & summarise data

Exercises (data visualization)

1. Data available at https://raw.githubusercontent.com/fivethirtyeight/uber-tlc-foil-response/master/uber-trip-data/uber-raw-data-sep14.csv is providing information about more than 1M trips made by UBER in september 2014. Data contains latitud and longitud of departures (variables `Lat` and `Lon`). Visualize where departures trips are located. The main goal of UBER is to know zones having more trips to increase the number of cars in those zones. NOTE: Investigate the function `peel` at `bigvis` package. Could this function help you in improving visualization?

Session info

```
sessionInfo()
```

```
R version 3.5.0 (2018-04-23)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 17134)

Matrix products: default

locale:
[1] LC_COLLATE=Spanish_Spain.1252  LC_CTYPE=Spanish_Spain.1252
[3] LC_MONETARY=Spanish_Spain.1252 LC_NUMERIC=C
[5] LC_TIME=Spanish_Spain.1252

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

loaded via a namespace (and not attached):
 [1] compiler_3.5.0  backports_1.1.2 magrittr_1.5    rprojroot_1.3-2
 [5] tools_3.5.0     htmltools_0.3.6 yaml_2.1.19     Rcpp_0.12.18
 [9] stringi_1.2.2   rmarkdown_1.9   knitr_1.20      stringr_1.3.1
[13] digest_0.6.15   evaluate_0.10.1
```