Advanced Dogfood Eating

Interactive graphics from R with Shiny and GoogleVis

Josh Laurito

To Dos

- Feedback
- Cover last module's homework
- Moving to interactive graphics
- googleVis
- shiny
- Next module's reading
- Next module's assignment

Announcemnents

- Understand that course has been confusing
- When in doubt, check the syllabus
- Also, ask me!
- Meetup 3/10, will try to organize a social one for class as well

Last module's homework

- Thank you all for being great about installing software early
- How did you combine files? I used `csvkit` http://csvkit.readthedocs.org/
- load what we need

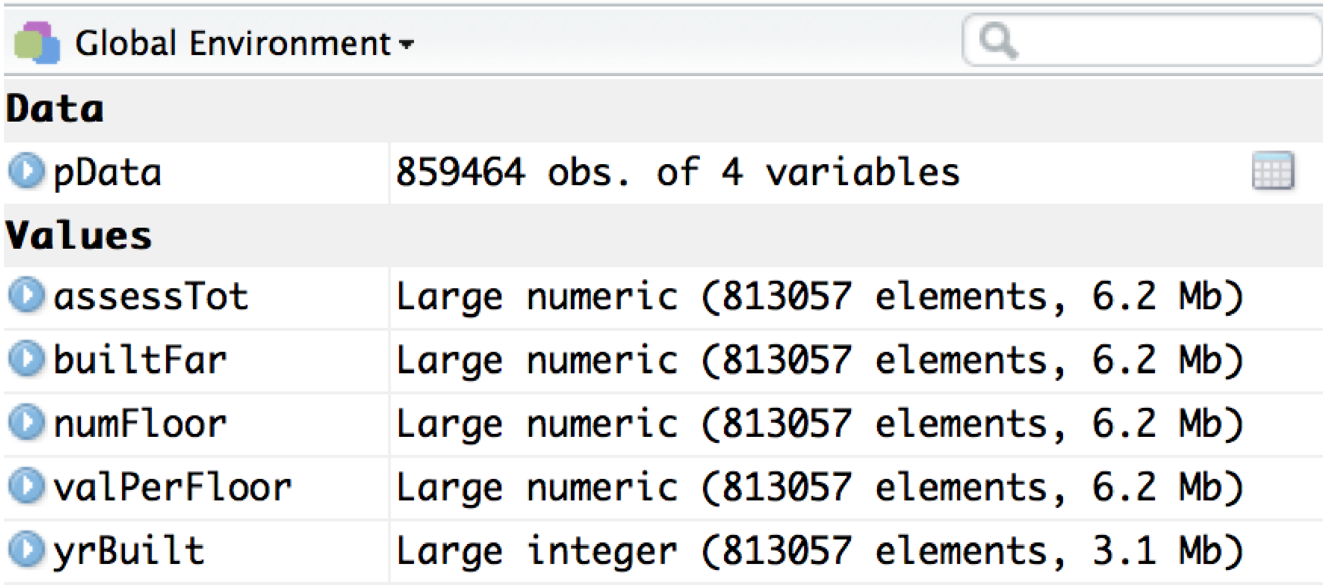
```
library("ggplot2")
library("plyr")
library("bigvis")
pData <- read.csv("all_PLUTO_data.csv")
```

Last week's homework - data cleaning

```
builtFar <- pData$BuiltFAR[pData$YearBuilt > 1850 & pData$YearBuilt < 2017 & pData$NumFloors != 0  ]

numFloor <- pData$NumFloors[pData$YearBuilt > 1850 & pData$YearBuilt < 2017 & pData$NumFloors != 0 ]

yrBuilt  <- pData$YearBuilt[pData$YearBuilt > 1850 & pData$YearBuilt < 2017 & pData$NumFloors != 0 ]

assessTot <- pData$AssessTot[pData$YearBuilt > 1850 & pData$YearBuilt < 2017 & pData$NumFloors != 0]

valPerFloor <- assessTot/numFloor
```

Last week's homework - lots of data

- You probably noticed that there was a lot of data
- Slows down analysis: viz becomes more important and harder to do



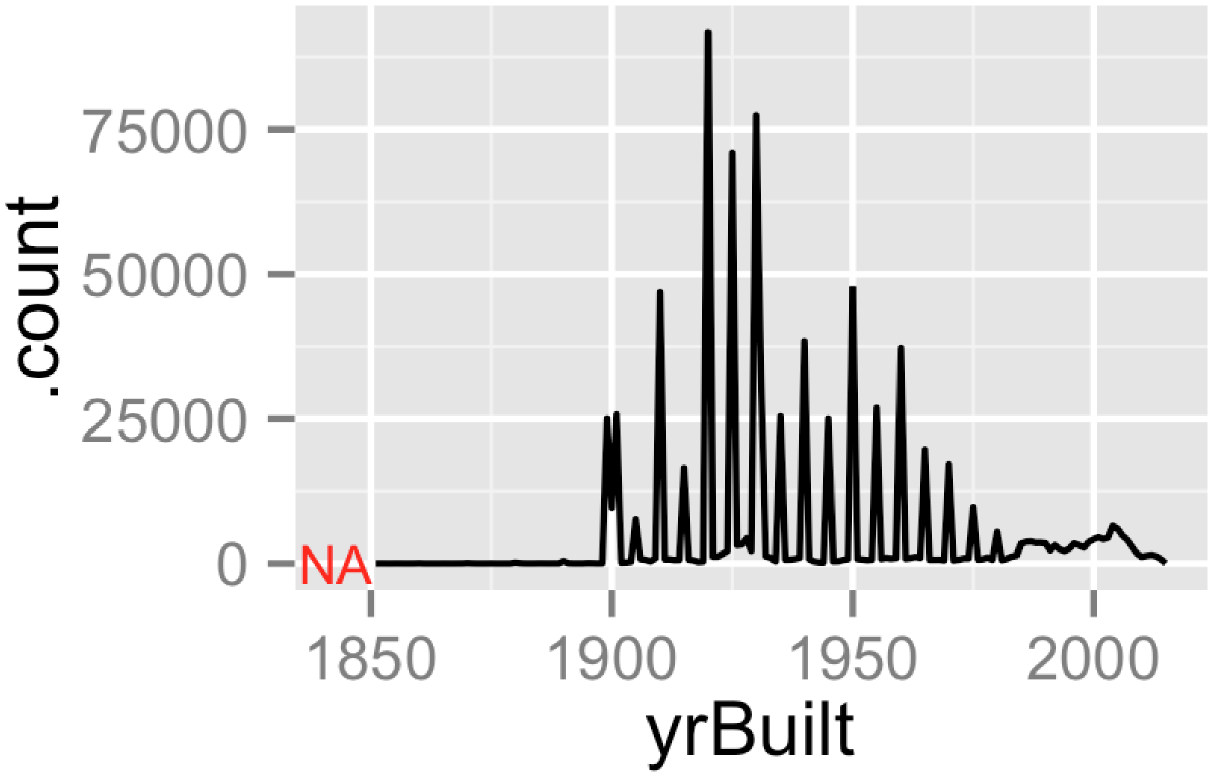
Data	
pData	859464 obs. of 4 variables
Values	
assessTot	Large numeric (813057 elements, 6.2 Mb)
builtFar	Large numeric (813057 elements, 6.2 Mb)
numFloor	Large numeric (813057 elements, 6.2 Mb)
valPerFloor	Large numeric (813057 elements, 6.2 Mb)
yrBuilt	Large integer (813057 elements, 3.1 Mb)

Last week's homework - building 'cut-off date'

- Poorly specified problem
- General concept: what does it mean to be an 'old building'
- We should start by checking when buildings were built

```
summary(yrBuilt)
yr <- condense(bin(yrBuilt, 1))
autoplot(yr)
ggsave('assets/img/yrBuilt.png',height=2, width = 3)
```

Last week's homework - building 'cut-off date'



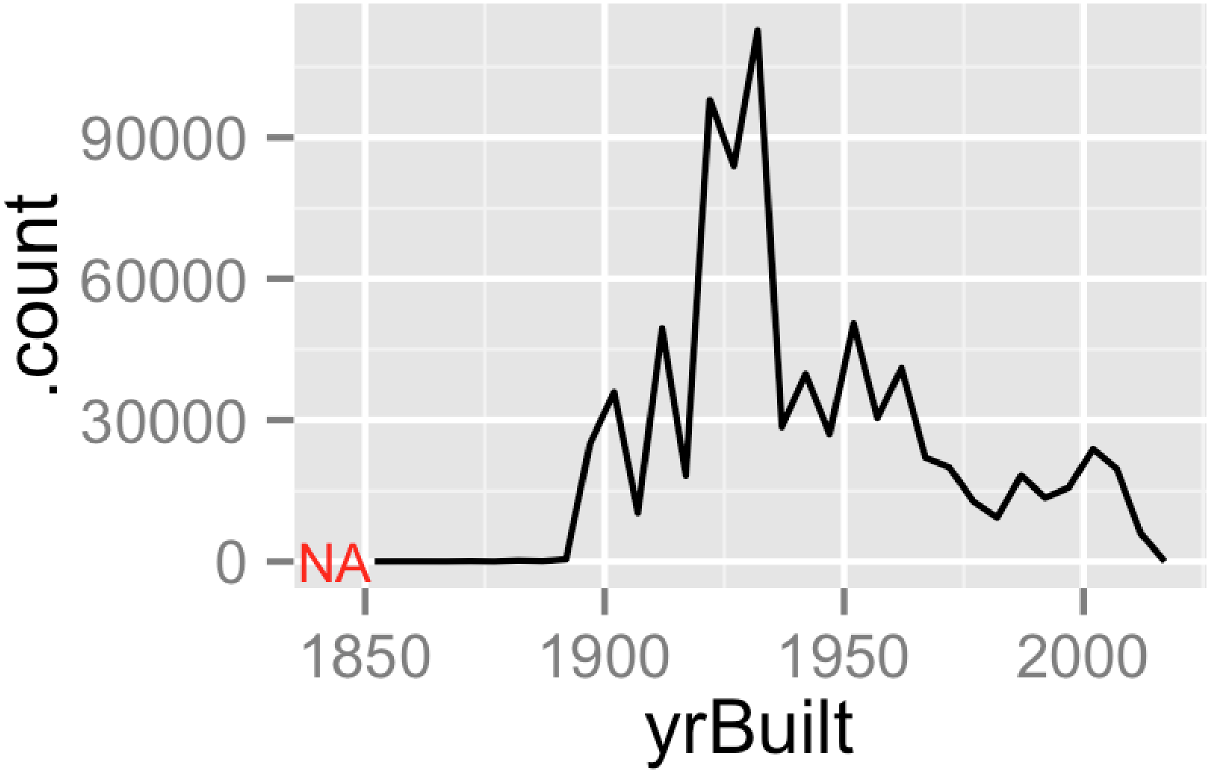
/

Last week's homework - building 'cut-off date'

- Oh no! Our data stinks!
- Clearly estimated when year =< 1980
- Can we proceed? Depends on context of questions

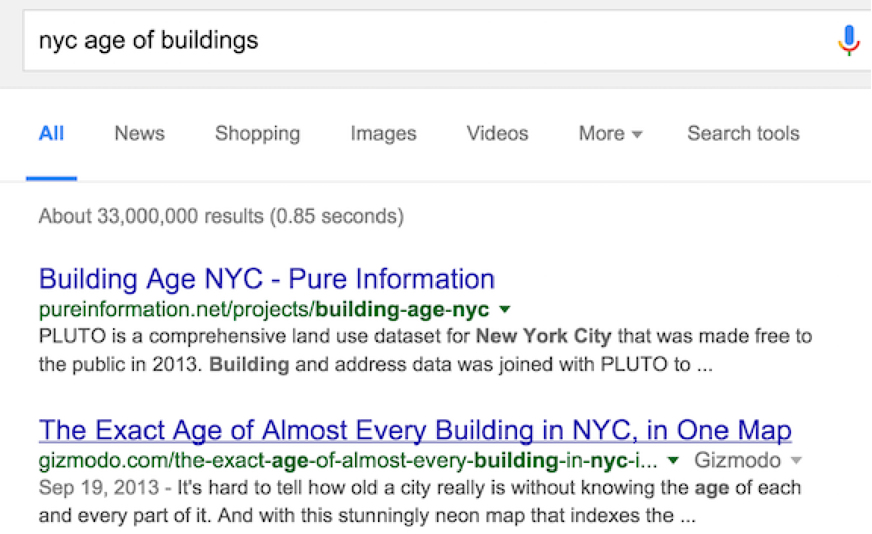
```
yr <- condense(bin(yrBuilt, 5))
autoplot(yr)
ggsave('assets/img/yrBuilt5.png', height=2, width = 3)
```

Last week's homework - building 'cut-off date'



/

Last week's homework - double check suspicious data



nyc age of buildings

About 33,000,000 results (0.85 seconds)

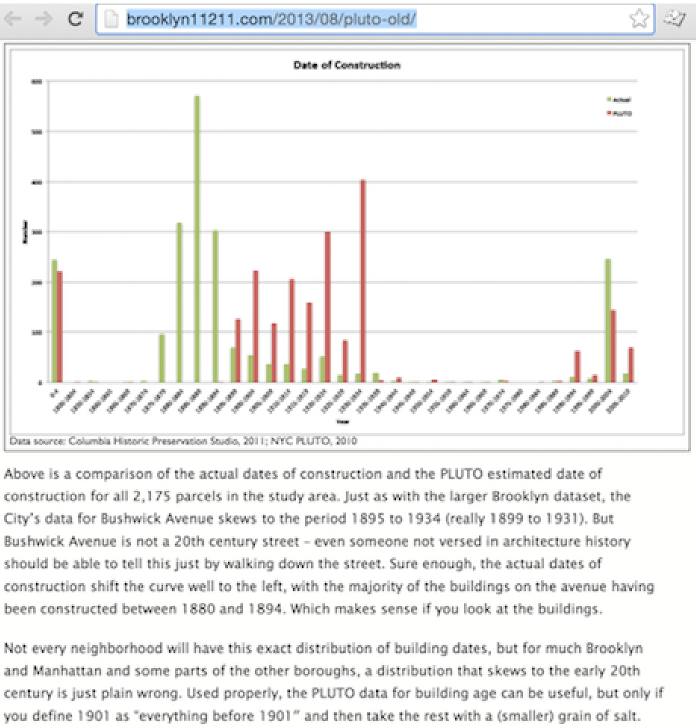
Building Age NYC - Pure Information
pureinformation.net/projects/**building-age-nyc**
PLUTO is a comprehensive land use dataset for **New York City** that was made free to the public in 2013. **Building** and address data was joined with PLUTO to ...

The Exact Age of Almost Every Building in NYC, in One Map
gizmodo.com/the-exact-**age**-of-almost-every-**building**-in-**nyc**-i... Gizmodo
Sep 19, 2013 - It's hard to tell how old a city really is without knowing the **age** of each and every part of it. And with this stunningly neon map that indexes the ...

/

Last week's homework - double check suspicious data

- Criticism of data at http://brooklyn11211.com/2013/08/pluto-old/
- Domain knowledge research is important!



Data source: Columbia Historic Preservation Studio, 2011; NYC PLUTO, 2010

Above is a comparison of the actual dates of construction and the PLUTO estimated date of construction for all 2,175 parcels in the study area. Just as with the larger Brooklyn dataset, the City's data for Bushwick Avenue skews to the period 1895 to 1934 (really 1899 to 1931). But Bushwick Avenue is not a 20th century street – even someone not versed in architecture history should be able to tell this just by walking down the street. Sure enough, the actual dates of construction shift the curve well to the left, with the majority of the buildings on the avenue having been constructed between 1880 and 1894. Which makes sense if you look at the buildings.

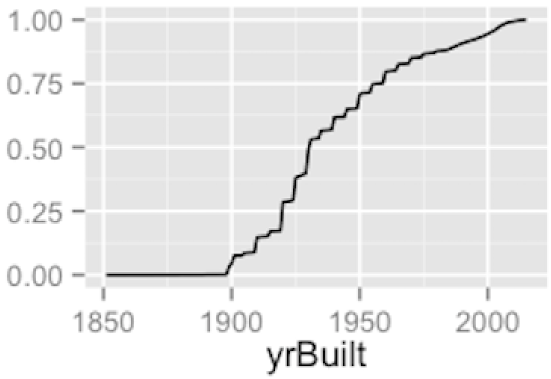
Not every neighborhood will have this exact distribution of building dates, but for much Brooklyn and Manhattan and some parts of the other boroughs, a distribution that skews to the early 20th century is just plain wrong. Used properly, the PLUTO data for building age can be useful, but only if you define 1901 as "everything before 1901" and then take the rest with a (smaller) grain of salt.

/

Last week's homework - finding cut-offs

```
yr2 <- condense(bin(yrBuilt, 1))
total <- sum(yr2$.count)
yr2$perc_built <- cumsum(yr2$.count)/total
ggplot(yr2, aes(x= yrBuilt, y=perc_built)) + geom_line() +ylab('')
ggsave('assets/img/cumcurve.png', height=2, width = 3, dpi = 100)
```

Last week's homework - finding cut-offs



/

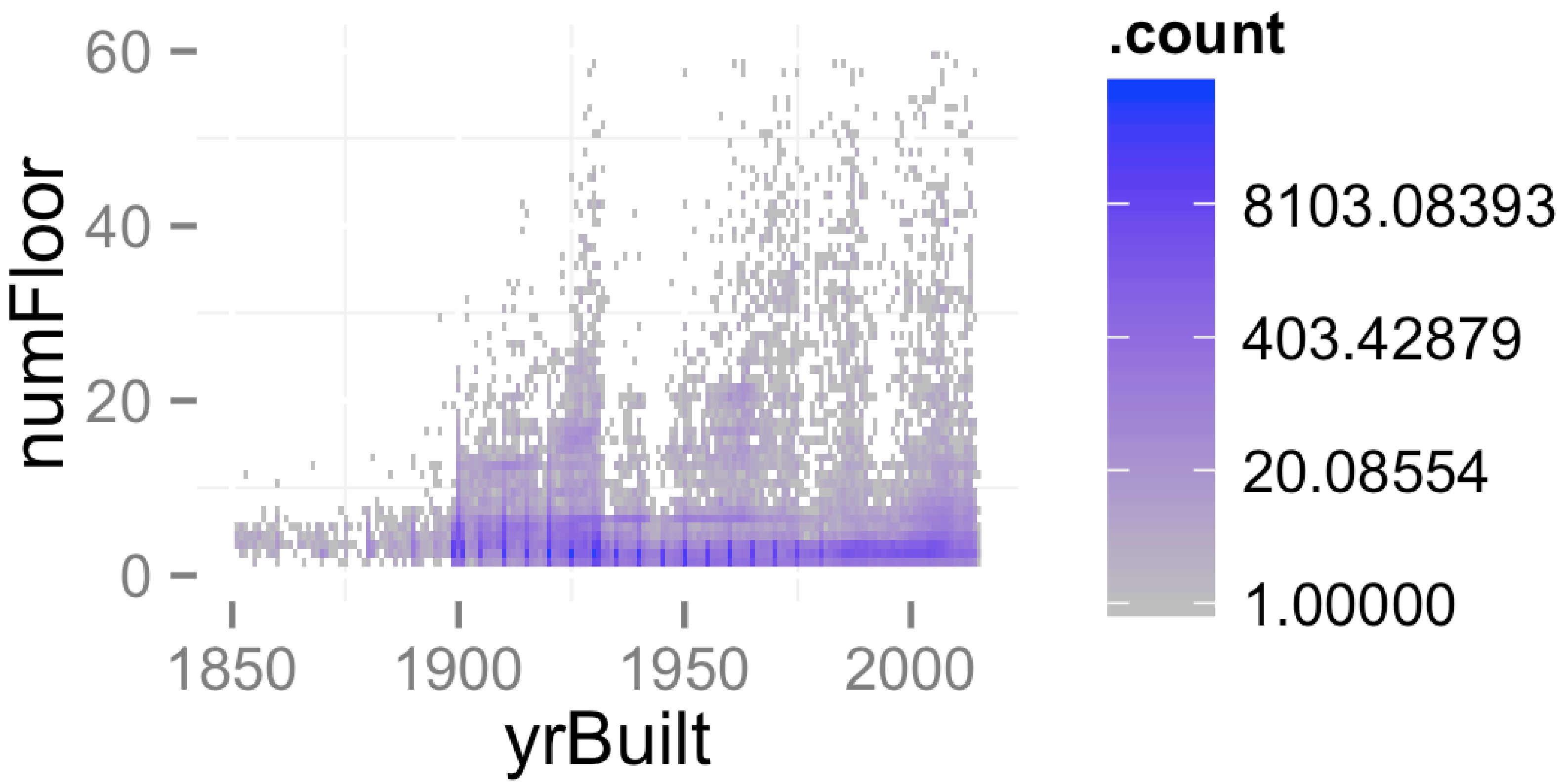
Last week's homework - cut-offs by group

- Similar question to previous, only with groups

```
flrVsYr <- condense(bin(yrBuilt, 1), bin(numFloor, 1))
p <- autoplot(flrVsYr) + theme(panel.background=element_rect(fill='white')) + ylim(0,60)
p + scale_fill_gradient(limits= c(1,100000),
                        low='grey',
                        high='#0000ff',
                        trans="log")
ggsave('assets/img/yr_v_flr.png', height=2, width = 4, dpi = 300)
```

Last week's homework - cut-offs by group

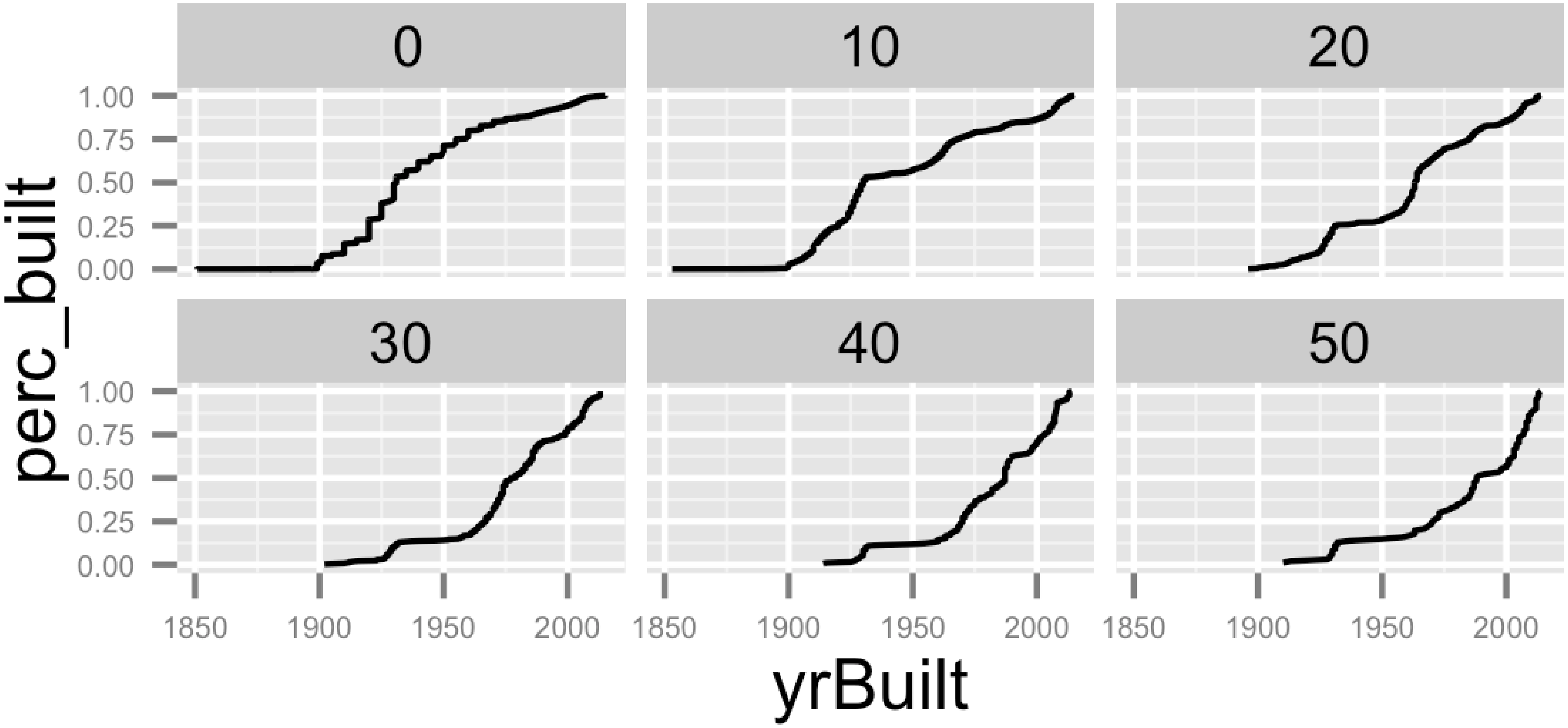
- Overplotting issue



Last week's homework - cut-offs by group

```r
flrVsYr$stories <- 10*trunc(flrVsYr$numFloor/10)
flrVsYr$stories <- sapply(flrVsYr$stories, min, 50)
flrVsYr$num_built <- ave(flrVsYr$.count,flrVsYr$stories, FUN=cumsum)
flrVsYr$perc_built <- flrVsYr$num_built/ ave(flrVsYr$.count,flrVsYr$stories, FUN=sum)
ggplot(flrVsYr[complete.cases(flrVsYr),], aes(x=yrBuilt, y=perc_built, group=stories)) + geom_line() +
facet_wrap(~ stories) + theme(axis.text=element_text(size=5))
ggsave('assets/img/stories.png', height=2, width = 4, dpi = 300)
```

Last week's homework - cut-offs by group

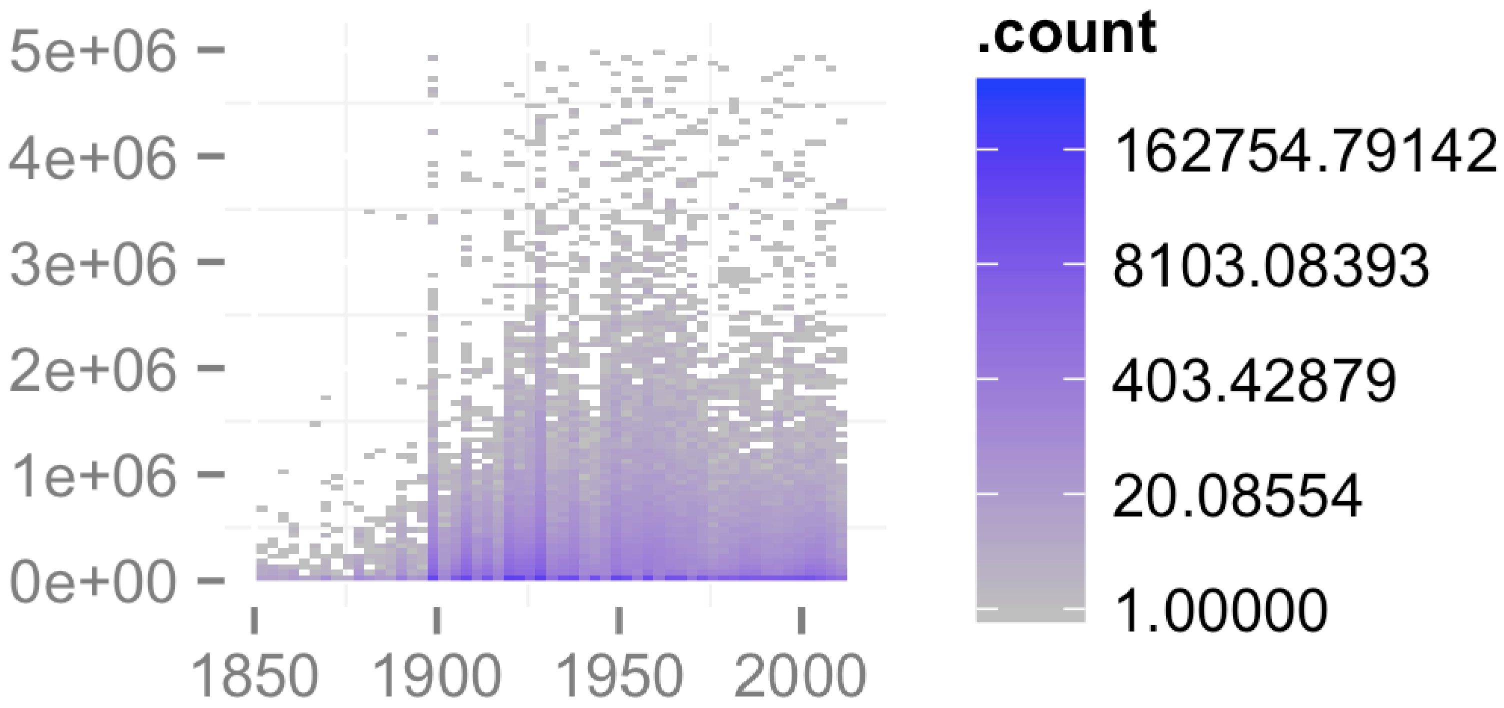


Last week's homework - cut-offs by group

```
assessVsYr <- condense(bin(yrBuilt[assessTot < 5000000],3), bin(valPerFloor[assessTot < 5000000], 50000))
p <- autoplot(assessVsYr) + theme(panel.background=element_rect(fill='white')) + xlim(1850,2015) + xlab('') + yl
p + scale_fill_gradient(limits= c(1,1000000),
                        low='grey',
                        high='#0000ff',
                        trans="log")
ggsave('assets/img/assess.png', height=2, width = 4, dpi = 300)
```

/

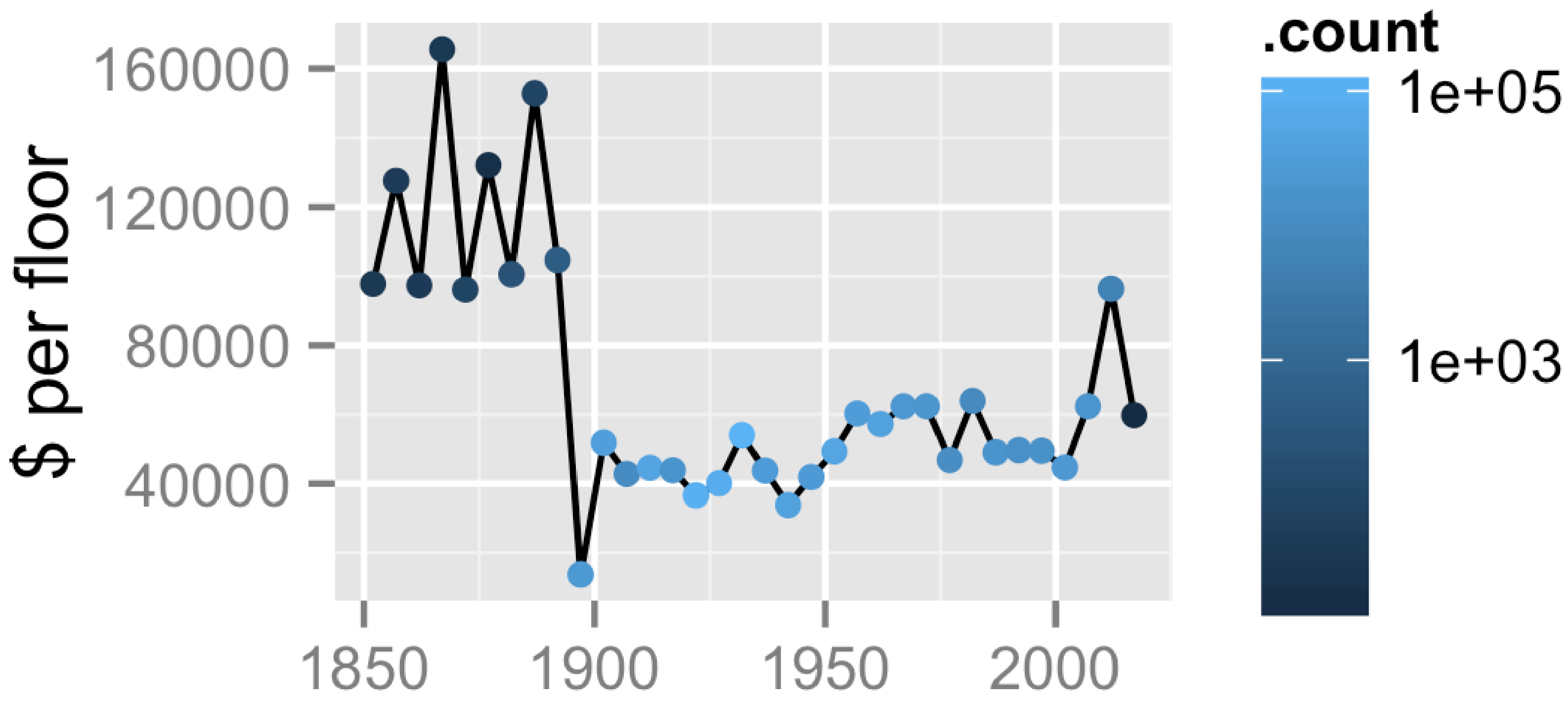
Last week's homework - cut-offs by group



Last week's homework - wartime

```
flrVal <- condense(bin(yrBuilt[assessTot < 5000000],5), z =valPerFloor[assessTot < 5000000] )
autoplot(flrVal) +xlab('') + ylab('$ per floor')
ggsave('assets/img/flrVal.png', height=2, width = 4, dpi = 300)
```

Last week's homework - cut-offs by group



Moving to interactive graphics

- Painful to create multiple different charts for investigation
- We are moving away from statistics here and towards design
- `ggplot2` is doing this too- moving to `ggvis`

/

Moving to interactive graphics - terms

- 'Server-side': what happens on the server (back-end, database)
- 'Client-side': what happens on the user's computer (Browser, JS)
- Scalable: how a site handles many visits at once

/

googleVis

- Often known as 'Hans Robling style charts'
- Interface to Google Chart API
- Compiles to HTML/Javascript
- Great demo by running `demo('googleVis')`

/

googleVis - advantages

- Fast and Easy
- Output is interactive and in-browser
- Very simple to combine charts

/

googleVis - drawbacks

- Warning: not all google charts are secure
- Need web access to run
- Risky: Google has history of deprecating projects

shiny

- Sponsored by RStudio: similar to googleVis but has a server-side component
- Can be integrated with googleVis
- Open Source

/

Next module's reading

- All on Blackboard

/

Next module's assignment

- CDC Wonder Data http://wonder.cdc.gov/ucd-icd10.html

/