Who Wants to Visualize Like a Baller?!

Jim Weiss

Crum & Foster

CAS RPM Seminar

March, 2019

Boston, MA

Madeline Main Downie

Willis Towers Watson

Antitrust Notice

- **The Casualty Actuarial Society is committed to adhering strictly to the letter and spirit of the antitrust laws. Seminars conducted under the auspices of the CAS are designed solely to provide a forum for the expression of various points of view on topics described in the programs or agendas for such meetings.**

- **Under no circumstances shall CAS seminars be used as a means for competing companies or firms to reach any understanding – expressed or implied – that restricts competition or in any way impairs the ability of members to exercise independent business judgment regarding matters affecting competition.**

- **It is the responsibility of all seminar participants to be aware of antitrust regulations, to prevent any written or verbal discussions that appear to violate these laws, and to adhere in every respect to the CAS antitrust compliance policy.**

BACKGROUND AND SET-UP

Learning Objectives

- Choose the right visual for what you are trying to explain
- Prepare data to maximize visualization flexibility
- Apply best practices in visualization using R

Data Visualization: Defining the Concept

- Visual representations that support the exploration, examination and communication of data
- Key elements of these visual representations:
  - Computer-supported
  - Interactive
  - “Map” abstract data to visual characteristics
  - Amplify cognition

Sources:

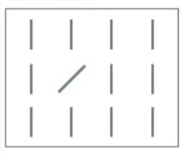
Now You See It: Simple Visualization Techniques for Quantitative Analysis, Stephen Few, Analytics Press, Oakland, CA, 2009.

Readings in Information Visualization: Using Vision to Think, Stuart K. Card, Jock D. MacKinlay, and Ben Shneiderman, Academic Press, San Diego, CA, 1999.

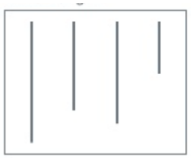
Pre-Attentive Attributes:

Maximizing Visual Perception

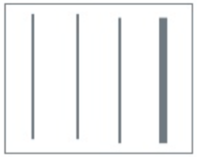
- Slope



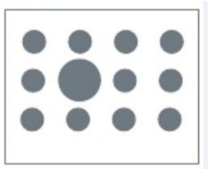
- Length



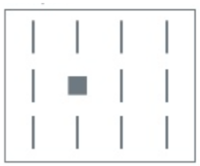
- Width



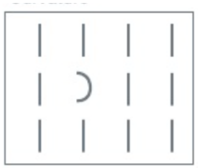
- Size



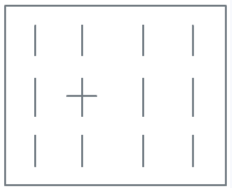
- Shape



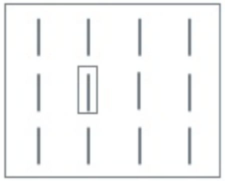
- Curvature



- Added marks



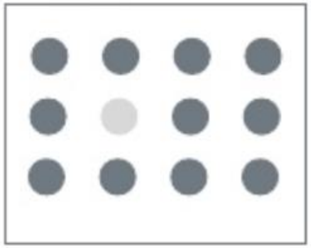
- Enclosure



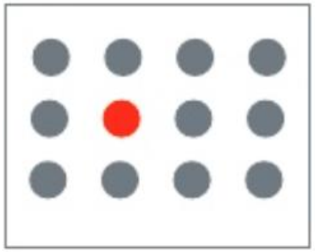
Pre-Attentive Attributes:

Maximizing Visual Perception

- Color value/intensity



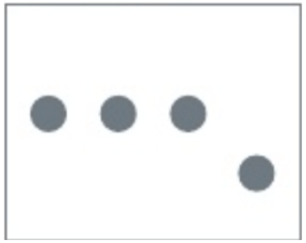
- Color hue



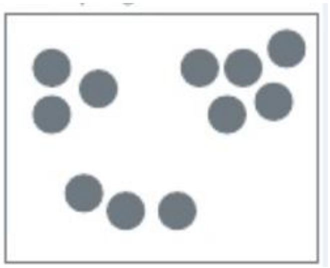
Pre-Attentive Attributes:

Maximizing Visual Perception

- 2-D position



- Spatial grouping



Suggested “ground rules”

- Work together with those around you
- The goal is not perfection; nor to mimic perfectly what is shown up-front
- Encountering bugs is okay!
- Sharing is encouraged

What’s the problem, Boston?

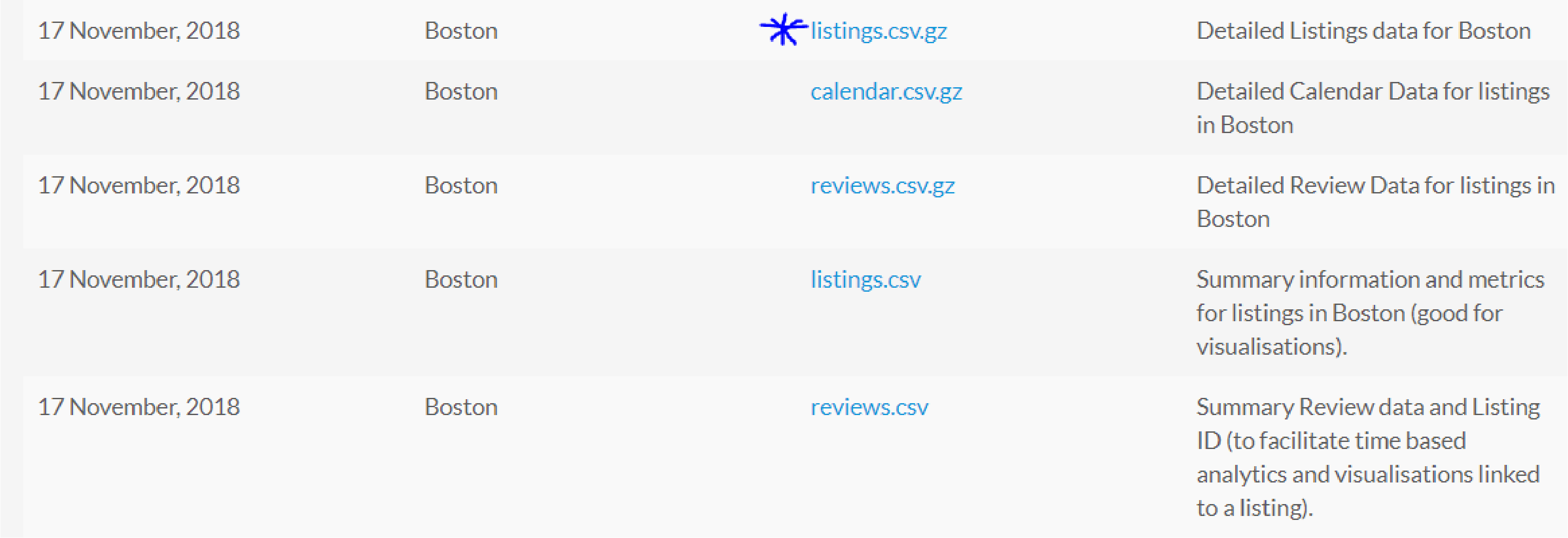
Rent a place …
That everybody wants …
But nobody’s booked …
In Boston come November, …
Without breaking the budget

Maximize absolute difference …
Between E(Availability) …
And actual Availability…
By modeling listings data, …
Subject to some constraints

For convenience we will assume flexible travel dates over a thirty day period, and that we are looking to book a stay of 1 to 3 nights in the City of Boston

Download the data

http://insideairbnb.com/get-the-data.html



17 November, 2018	Boston	* listings.csv.gz	Detailed Listings data for Boston
17 November, 2018	Boston	calendar.csv.gz	Detailed Calendar Data for listings in Boston
17 November, 2018	Boston	reviews.csv.gz	Detailed Review Data for listings in Boston
17 November, 2018	Boston	listings.csv	Summary information and metrics for listings in Boston (good for visualisations).
17 November, 2018	Boston	reviews.csv	Summary Review data and Listing ID (to facilitate time based analytics and visualisations linked to a listing).

EXPLORATION

First we’ll read in our data and some libraries

```
.libPaths("//put/your/library/path/here/lib")


library(ggplot2)
library(readr)
library(tm)
library(wordcloud)
```

First we'll read in our data and some libraries

```
library(corrgram)
library(corrplot)
library(igraph)
library(gains)
library(rlist)
library(dplyr)
# library(devtools)
# install_github("arilamstein/choroplethrZip")
library(choroplethrZip)
library(rworldmap)

data.listings <- read.csv("//put/your/data/path/here/listings.csv")

data<-subset(data.listings, minimum_nights<=3)
data$numericprice <- as.numeric(gsub('\\$|,','',as.character(data$price)))
```

Initial Exploratory Data Analysis: Let’s use ggplot2 to start exploring our data

- **Grammar of graphics**
  - **Data** set
  - **Coordinate system**
  - **Geoms**

Basic format:

```
Ggplot (data = <DATA>) + <GEOM_FUNCTION> (mapping = aes (<MAPPINGS>),
stat= <STAT>, position = <POSITION>) + <COORDINATE FUNCTION> + <FACET
FUNCTION> + <SCALE FUNCTION> + <THEME FUNCTION>
```

Basic scatter plot:

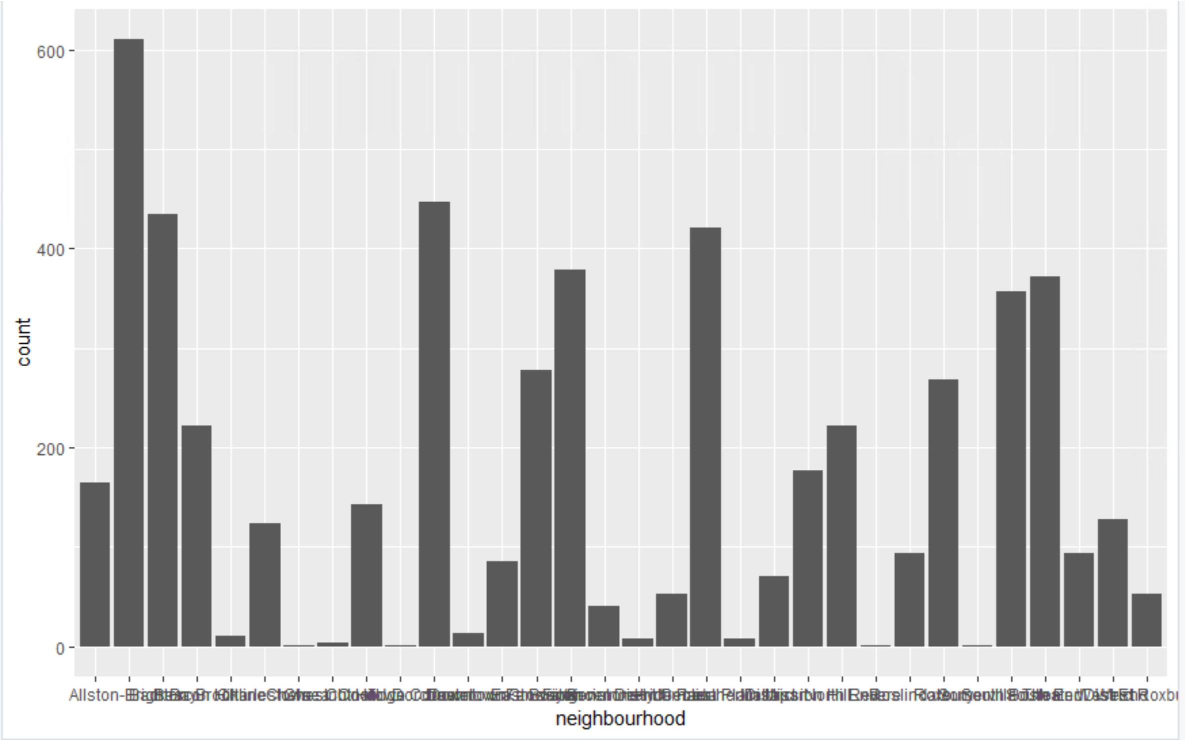
```
ggplot(data = diamonds, aes(x = carat, y = price)) +
  geom_point()
```

For more information – see https://www.rstudio.com/wp-content/uploads/2016/11/ggplot2-cheatsheet-2.1.pdf

Initial Exploratory Data Analysis: Use ggplot2 to make bar graphs

- Let’s look at neighborhood first – create a basic bar chart

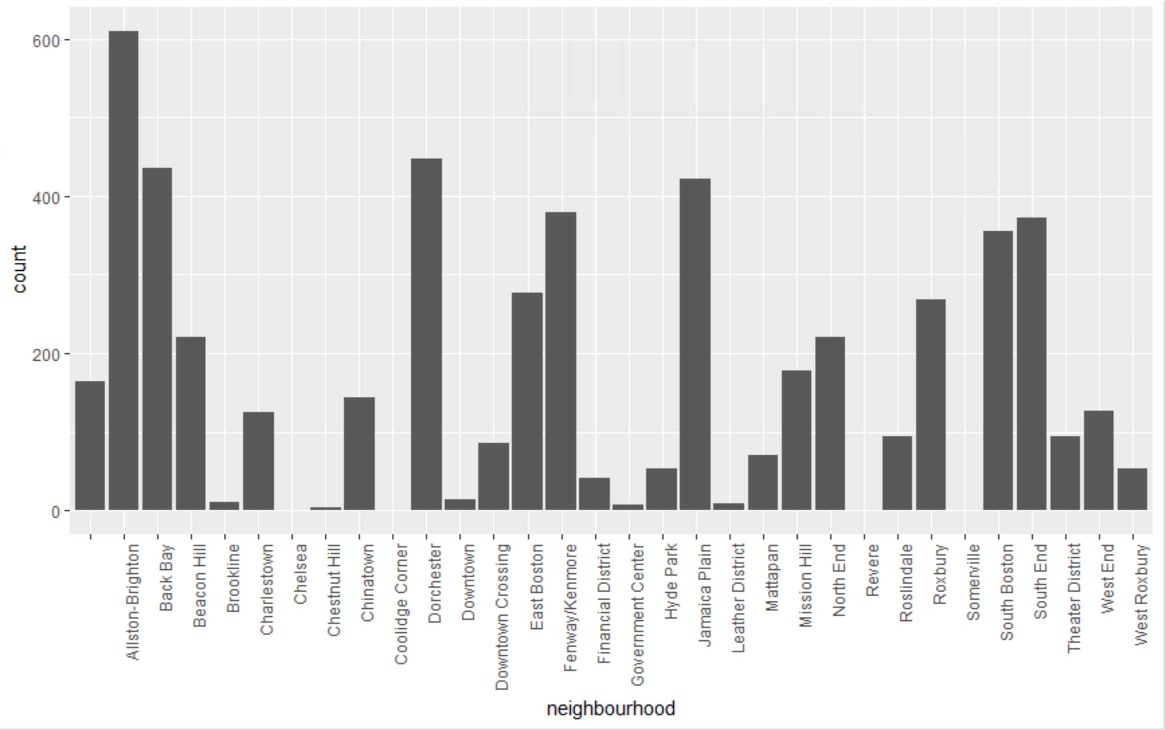
```
ggplot(data = data, aes(neighbourhood)) +
 geom_bar()
```



Initial Exploratory Data Analysis: Use ggplot2 to make bar graphs

Let’s clean up the x-axis to make it easier to read

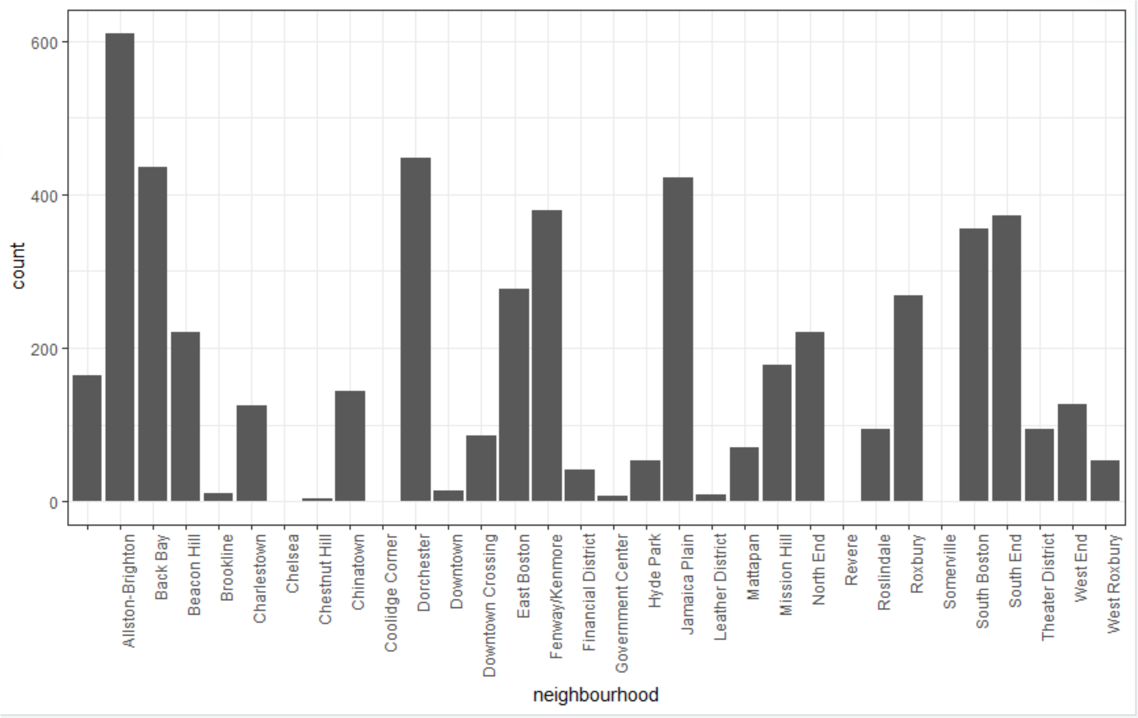
```
ggplot(data = data, aes(neighbourhood)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Initial Exploratory Data Analysis: Use ggplot2 to make bar graphs

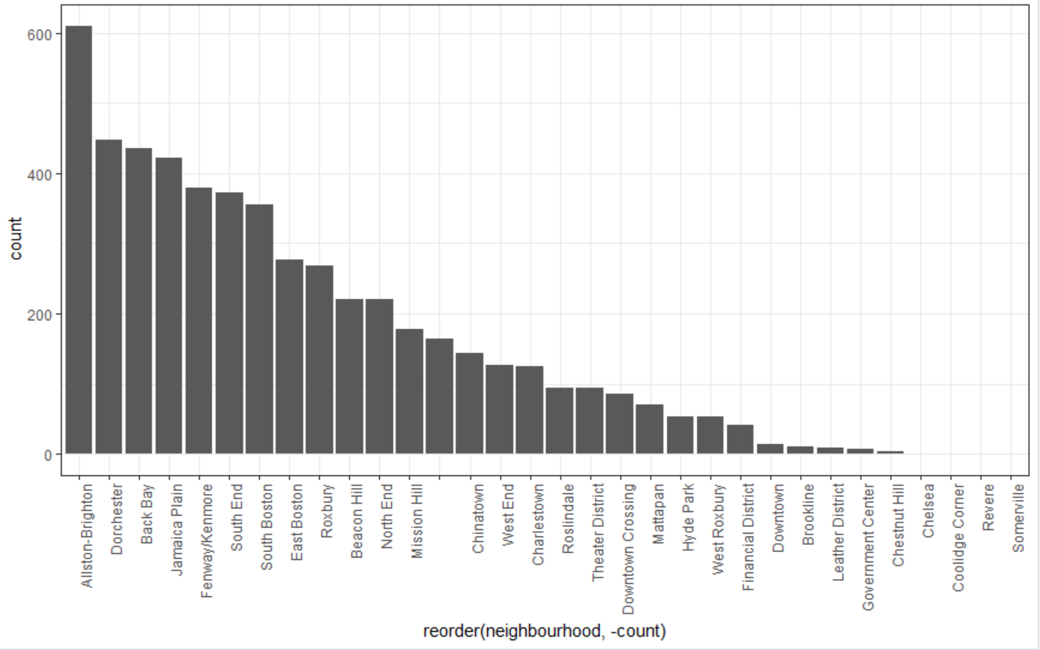
- Add a full theme

```
ggplot(data = data, aes(neighbourhood)) +
  geom_bar() +
  theme_bw() + theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Initial Exploratory Data Analysis: Reordering our bar graph

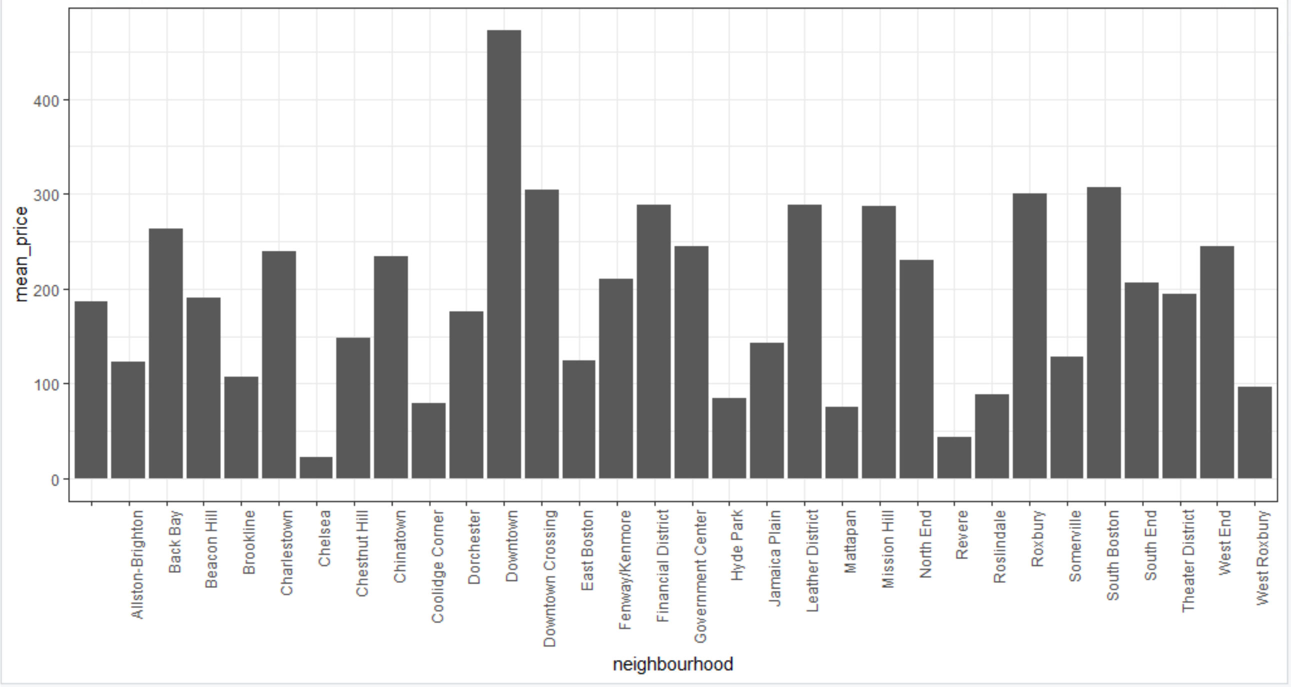
```
data %>%
  count(neighbourhood) %>%
  mutate(count = n) -> data3

ggplot(data3, aes(x = reorder(neighbourhood, -count), y = count)) + geom_bar(stat = "identity")+
  theme_bw() + theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Initial Exploratory Data Analysis:

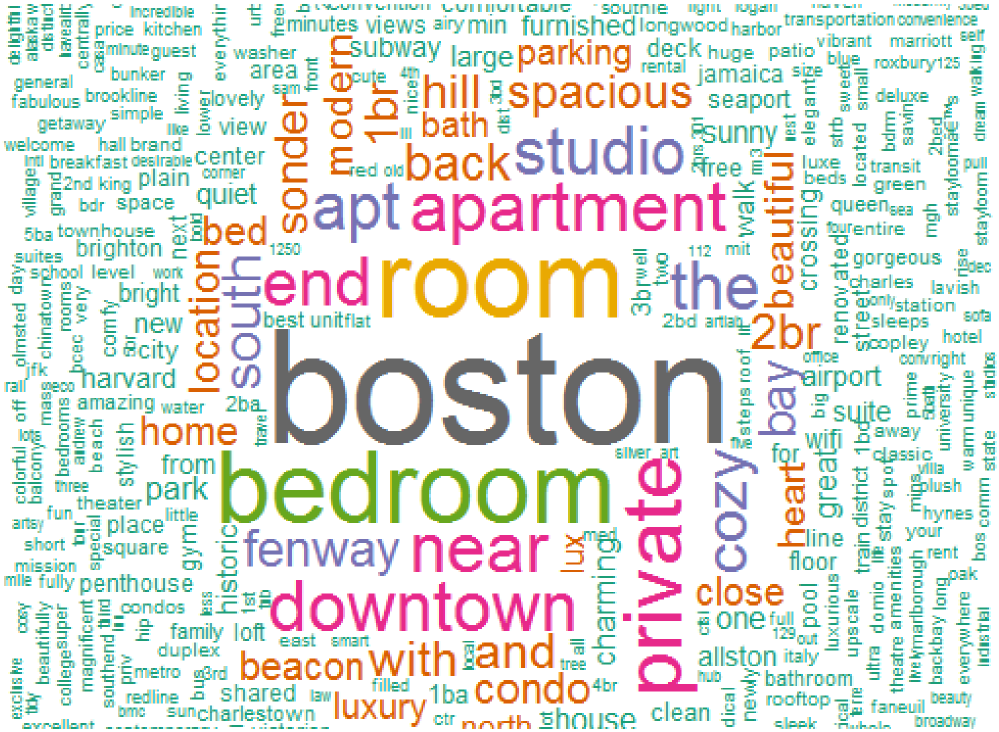
What is the most expensive neighborhood?

```
group_by(data, neighbourhood) %>% summarize(mean_price = mean(numericprice)) -> data4

ggplot(data = data4, aes(x = neighbourhood, y = mean_price)) +
  geom_bar(stat = "identity")+
  theme_bw() + theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Create Word Cloud

docs = Corpus(VectorSource(data$name))
docs <- tm_map(docs, content_transformer(tolower))
wordcloud (docs, scale=c(5,0.5), random.order=FALSE, rot.per=0.35, use.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))



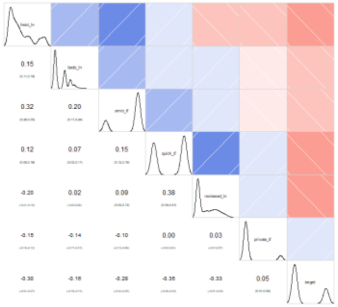
MODEL TRAINING

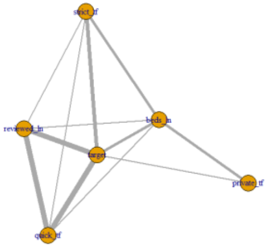
Features and logistic regression

```
data$properties<-1
data$target<-ifelse(data$availability_30 > 0, 0, 1)
data$reviewed_ln<-ifelse(is.na(data$reviews_per_month) | data$reviews_per_month <= 1, 0 ,
log(data$reviews_per_month) )
data$listct_ln<-ifelse(is.na(data$host_listings_count) | data$host_listings_count <= 1, 0 , log(data$host_listings_count))
data$beds_ln<-ifelse(is.na(data$beds) | data$beds <= 1, 0, log(data$beds))
data$quick_tf<-ifelse( grepl('an hour',tolower(data$host_response_time)) == TRUE, 1, 0)
data$strict_tf<-ifelse( grepl('strict',data$cancellation_policy) == TRUE | data$cancellation_policy == 'moderate', 1, 0)
data$private_tf<-ifelse( grepl('private',tolower(data$name)) == TRUE, 1, 0)

set.seed(123)
varstrg <- 'reviewed_ln + listct_ln + beds_ln + quick_tf + strict_tf + private_tf'
data$rand<-as.numeric(runif(nrow(data)))
holdout<-subset(data, rand<= 0.5)
train<-subset(data, rand >= 0.5)
glmobj <- glm(formula(paste('target~',varstrg)),family=binomial(link='logit'),data=train)
summary(glmobj)

holdout$pred<-predict(glmobj,holdout,type="response")
```

Correlation:
Some alternatives to consider

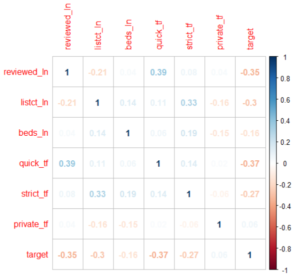
Throwing shade

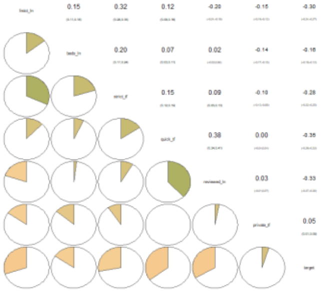




Network like a pro

Just the facts





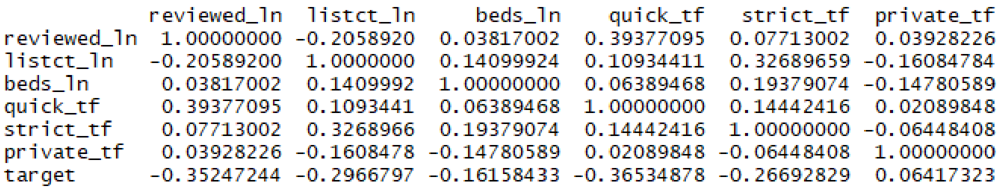
Pie shop

Why these options? Visit: https://extremepresentation.typepad.com/files/choosing-a-good-chart-09.pdf

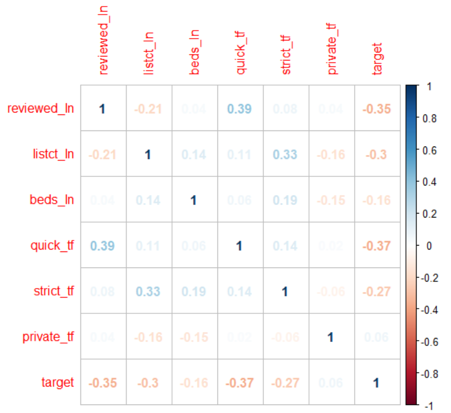
Correlation:
use **base** or **corrplot** to produce correlation matrices

- Ew.

```
correl<-subset(data,select=c(reviewed_ln,listct_ln,beds_ln,quick_tf,strict_tf,private_tf,target))
cor(correl)

corrplot(cor(correl),method="number")
```



```
            reviewed_ln  listct_ln     beds_ln    quick_tf   strict_tf  private_tf
reviewed_ln  1.00000000 -0.2058920  0.03817002  0.39377095  0.07713002  0.03928226
listct_ln   -0.20589200  1.0000000  0.14099924  0.10934411  0.32689659 -0.16084784
beds_ln      0.03817002  0.1409992  1.00000000  0.06389468  0.19379074 -0.14780589
quick_tf     0.39377095  0.1093441  0.06389468  1.00000000  0.14442416  0.02089848
strict_tf    0.07713002  0.3268966  0.19379074  0.14442416  1.00000000 -0.06448408
private_tf   0.03928226 -0.1608478 -0.14780589  0.02089848 -0.06448408  1.00000000
target      -0.35247244 -0.2966797 -0.16158433 -0.36534878 -0.26692829  0.06417323
```

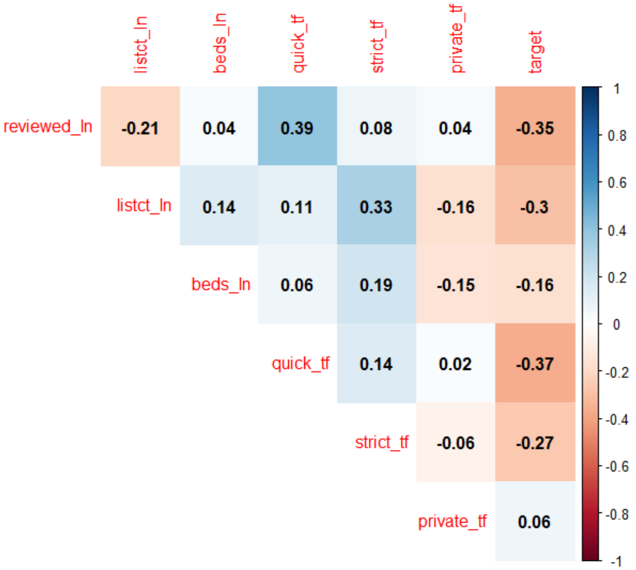


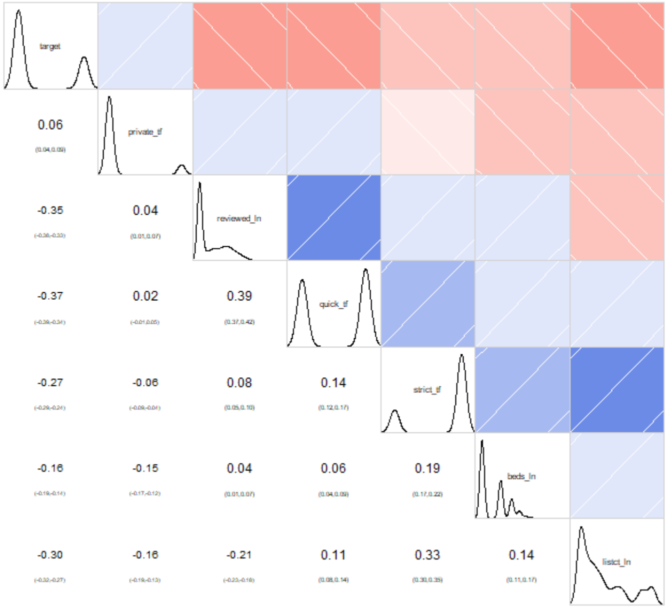
	reviewed_ln	listct_ln	beds_ln	quick_tf	strict_tf	private_tf	target
reviewed_ln	1	-0.21	0.04	0.39	0.08	0.04	-0.35
listct_ln	-0.21	1	0.14	0.11	0.33	-0.16	-0.3
beds_ln	0.04	0.14	1	0.06	0.19	-0.15	-0.16
quick_tf	0.39	0.11	0.06	1	0.14	0.02	-0.37
strict_tf	0.08	0.33	0.19	0.14	1	-0.06	-0.27
private_tf	0.04	-0.16	-0.15	0.02	-0.06	1	0.06
target	-0.35	-0.3	-0.16	-0.37	-0.27	0.06	1

Correlation:
use **corrplot** or **corrgram** to incorporate shading

- Shade rectangles with coefficients superimposed or separate

```
corrplot(cor(correl),  method="color",type="upper",addCoef.col  = "black",diag=FALSE)

corrgram(correl,order=TRUE,lower.panel=panel.conf,upper.panel=panel.shade,diag.panel=panel.density)
```

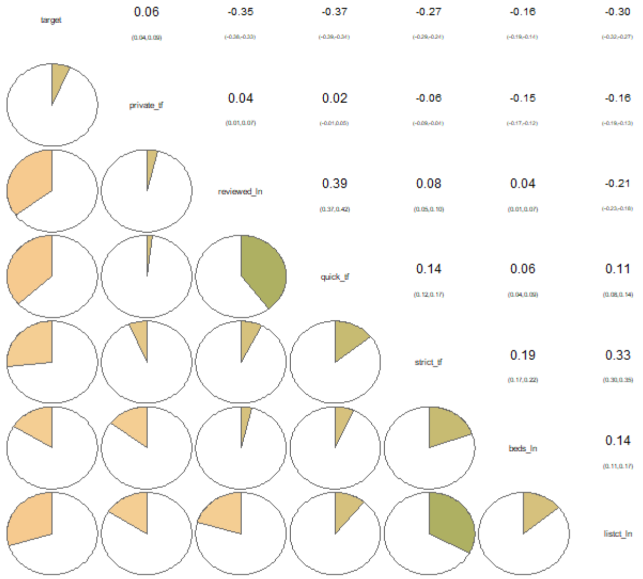




Correlation: use **corrgraph** to incorporate size/area

- Use pie charts (!) to allow for easier comparisons

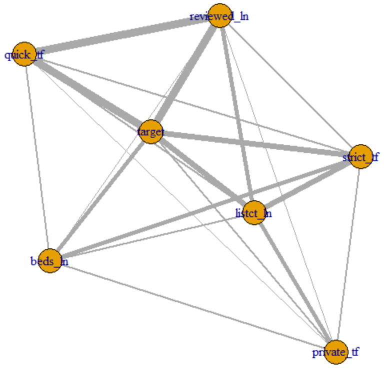
```
corrgram(correl, order=TRUE, upper.panel=panel.conf, lower.panel=panel.pie, text.panel=panel.txt,
col.regions=colorRampPalette(c("darkgoldenrod4", "burlywood1","darkkhaki", "darkgreen")))
```



Correlation:
use **igraph** to incorporate width and spatial grouping

- Create dataset of pairwise combinations from matrix, via loop(s)
- Visualize as network diagram to allow quick multi-comparisons

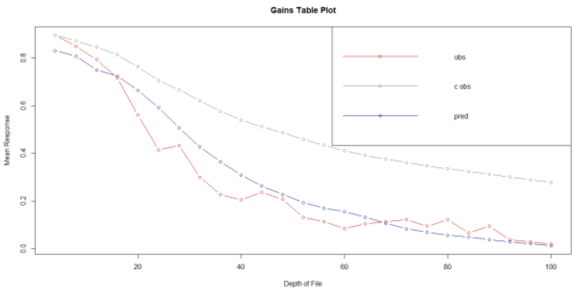
```
corrmatrix<-as.data.frame(cor(correl))
network<-as.data.frame( cbind(rownames(corrmatrix)[1], colnames(corrmatrix)[1], as.numeric(corrmatrix[1,1])))
for (ctr2 in 1:ncol(corrmatrix)) {
for (ctr1 in 1:nrow(corrmatrix)) {
temp1<-as.data.frame( cbind(rownames(corrmatrix)[ctr1], colnames(corrmatrix)[ctr2], as.numeric(corrmatrix[ctr1,ctr2])))
if (ctr1> ctr2) { network<-rbind(network,temp1)} } }
network<-subset(network, as.character(V1) != as.character(V2))
rm(corrmatrix,temp1)

network$V3<-round( abs(as.numeric(as.character(network$V3))), digits=1)
diagram <- graph_from_data_frame(network, directed=FALSE)
diagram <- set_edge_attr(diagram, "weight", value = 25*network$V3)
plot(diagram, edge.width = E(diagram)$weight)
```

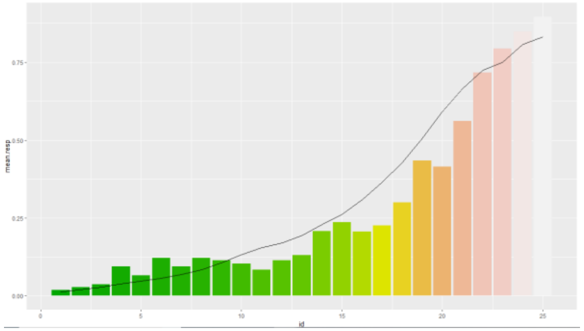


MODEL VALIDATION

Gains: some alternatives to consider

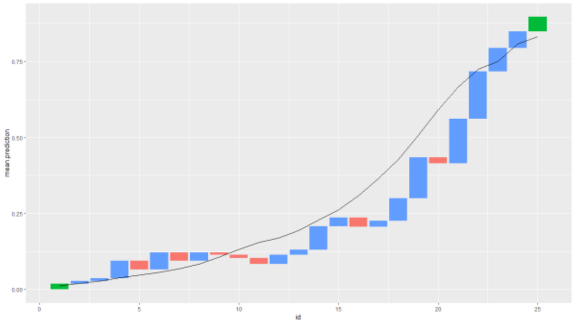
Lazy river

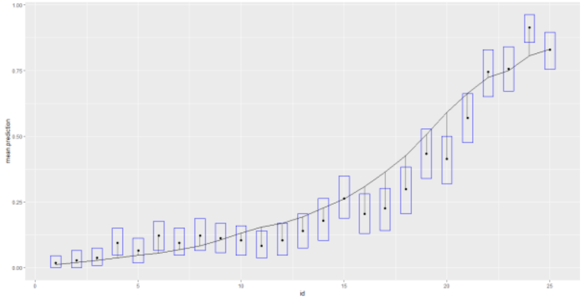




Greek columns

Waterfalls of Boston





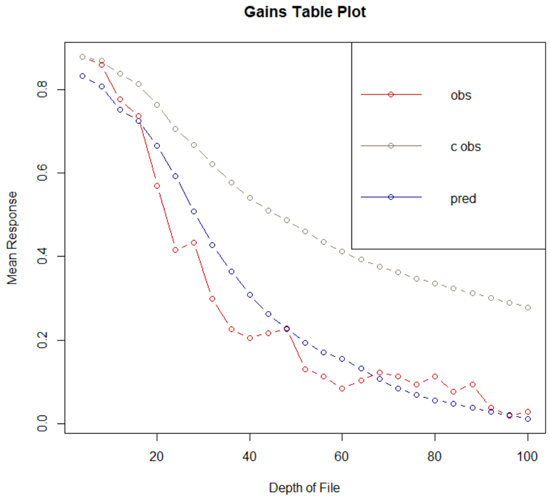
Scatter brain

Why these options? Visit: https://extremepresentation.typepad.com/files/choosing-a-good-chart-09.pdf

Gains: use **gains** to manipulate data and represent orientation/slope

- Order by prediction and calculated predicted vs. observed for 25 equal groups
- Include bootstrapped confidence intervals for later use
- Plot gains chart using lines to represent predicted and observed

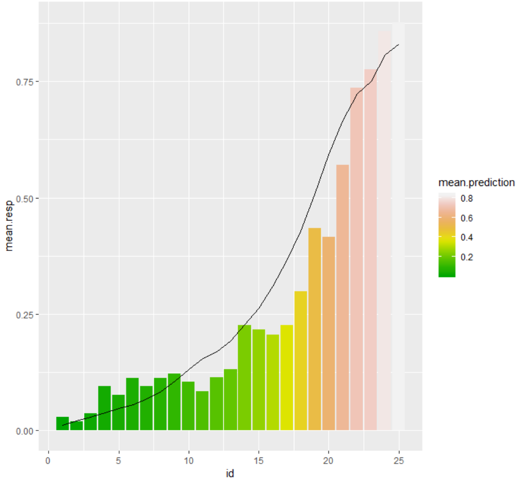
```
bkt<-25
gainslist<-gains(holdout$target,holdout$pred,ties.method="random",conf="boot",groups=bkt)
plot.gains(gainslist,legend=c("obs","c obs","pred"))
```



Gains: use **ggplot2** to incorporate more traditional look and feel

- Reorder ascending
- Plot observed as columns
- Plot predicted as lines

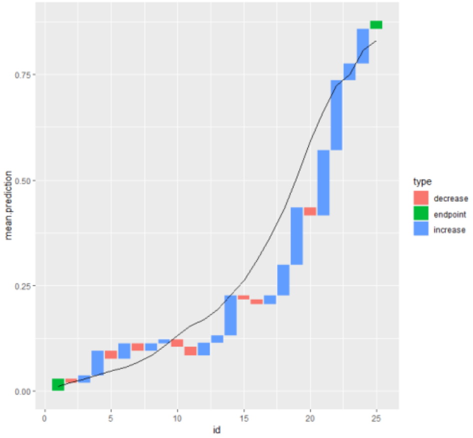
```
lift<-as.data.frame(list.flatten(gainslist))
lift <- lift[order(-lift$cume.obs),]
lift$id<-1:nrow(lift)
ggplot(data=lift) + geom_bar(aes(x=id,y=mean.resp,fill=mean.prediction),stat="identity") +
scale_fill_gradientn(colours=terrain.colors(10)) + geom_line(aes(x=id,y=mean.prediction,group=1),color="black")
```



Gains: use **ggplot2** to incorporate length and 2D position

- Create variable with *prior* quantile’s observed, via loop
- Classify each row as increase, decrease, origin – via (same) loop
- Plot predicted (line) vs. observed (waterfall)

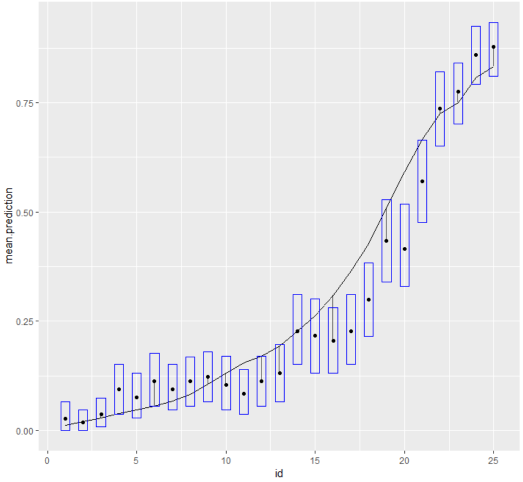
```
box<-lift
box$mean.resp.prior<-0
for (ctr in 2:bkt) { box$mean.resp.prior[ctr]<-box$mean.resp[ctr-1]  }
box$type<- ifelse(  box$id ==1 | box$id==nrow(box),  "endpoint", ifelse(box$mean.resp  >= box$mean.resp.prior,
"increase", "decrease"))
ggplot(box) + geom_rect(aes(id, fill=type,  xmin  = id - 0.45, xmax  = id + 0.45, ymin  = box$mean.resp,  ymax  =
box$mean.resp.prior))  + geom_line(aes(x=id,y=mean.prediction,group=1),color="black")
```



Gains: use **ggplot2** to incorporate enclosure

- Plot predicted (line) vs.
- Observed (point, and – optionally – narrow line dropping from predicted) vs.
- Bootstrapped confidence interval (rectangle)

```
ggplot(box) + geom_rect(aes(id, xmin = id - 0.01, xmax = id + 0.01, ymin = pmin(box$mean.resp,box$mean.prediction),
ymax = pmax(box$mean.resp,box$mean.prediction)))  + geom_line(aes(x=id,y=mean.prediction,group=1),color="black")
+ geom_point(aes(x=id,y=mean.resp))  + geom_rect(aes(id, xmin = id - 0.25, xmax = id + 0.25, ymin = box$conf.lower,
ymax  =box$conf.upper),colour="blue",fill=NA)
```

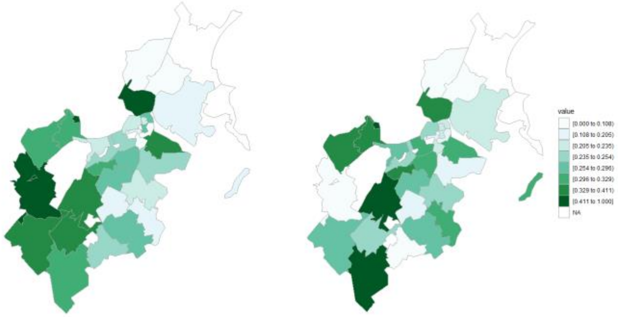


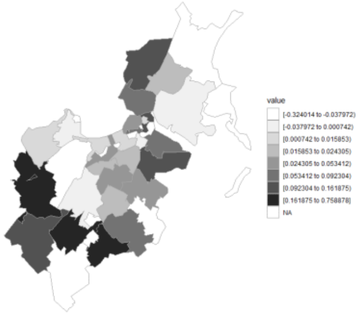
MODEL IMPLEMENTATION

Maps: some alternatives to consider

Seeing double

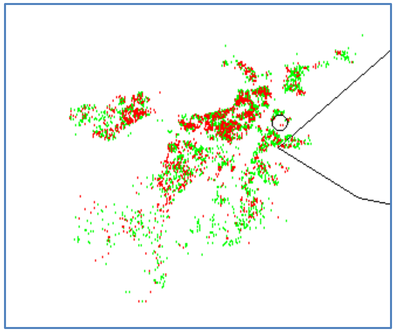
Seeing double

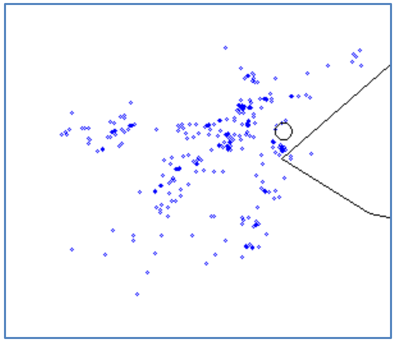




Comedy of errors

Tiny bubbles





Here, there, everywhere

Why these options? Visit: https://extremepresentation.typepad.com/files/choosing-a-good-chart-09.pdf

Maps: prepare data for heat mapping

- Score full dataset (training + holdout)
- Aggregate predicted, observed to ZIP level
- Calculate ZIP averages
- Load geographic data and join with modeling data

```
data(zip.regions)
geo <- filter(zip.regions, state.name == "massachusetts")
scored<-data
scored$pred<-predict(glmobj,scored,type="response")
scored$zip<- as.character(substr(scored$zipcode,1,5))

heat<-aggregate(cbind(properties,target,pred)~zip,scored,sum)
heat$value1<-ifelse(heat$properties==0,  0 , heat$pred/heat$properties)
heat$value2<-ifelse(heat$properties==0,  0 , heat$target/heat$properties)
heat$value3<-ifelse(heat$properties==0,  0 , heat$value1 - heat$value2)
heat <- left_join(geo, heat, by = c("region" = "zip"))
```

Maps: prepare data for density mapping

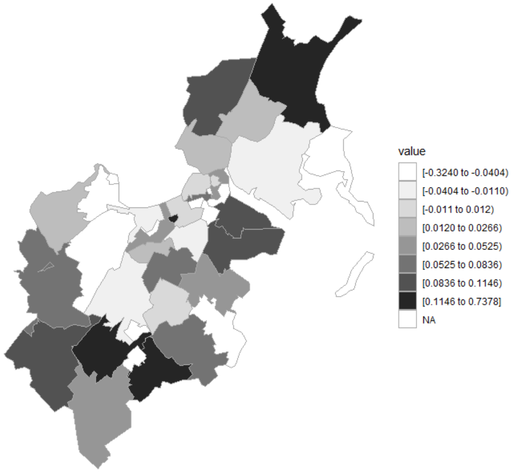
- Load world map
- Partition property-level data into available vs. booked
- Create dataset with in-criteria properties

```
newmap <- getMap(resolution = "low")

density<-scored
density$pred<-predict(glmobj,density,type="response")
opens<-subset(density,target==0)
booked<-subset(density,target==1)
arbit<-subset(density,  (pred - target) > 0.25 & as.numeric(price)  < 100)
```

Maps: use **choroplethrZip** to compare arbitrage opportunities by intensity/hue

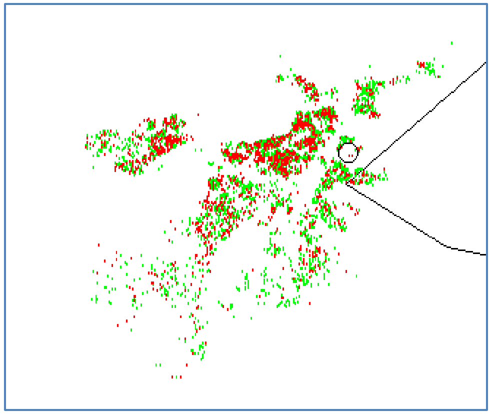
- Zoom on Suffolk County, MA
- Shade based on difference of average predicted versus observed

```
chordata<-heat
chordata$value<-chordata$value3
chordata <- unique ( subset(chordata,!is.na(state.name) & !is.na(value),select=c(region,value)) )
zip_choropleth(chordata, num_colors=8, county_zoom=25025) + scale_fill_brewer(palette=6)
```

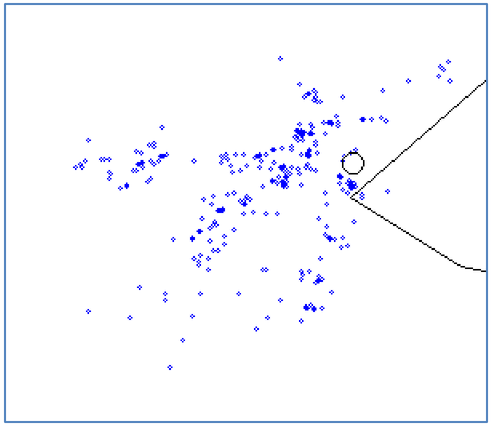


Maps: use **rworldmap** and **base** to exhibit choice opportunities by space/distance

- Zoom in on frame encompassing Boston
- Plot black marker for Westin Boston Waterfront
- Plot green marker for each available property
- Plot red marker for each booked property
- Plot blue marker for each arbitrage opportunity



```
plot(newmap, xlim=c(-71.2,-70.8), ylim=c(42.15,42.25), asp = 1)
points(opens$longitude, opens$latitude, col =c("green"), cex = .2)
points(booked$longitude, booked$latitude, col =c("red"), cex = .2)
points(-71.0448975,42.3461303, col =c("black"), cex = 2)

plot(newmap, xlim=c(-71.2,-70.8), ylim=c(42.15,42.25), asp = 1)
points(arbit$longitude, arbit$latitude, col =c("blue"), cex = .4)
points(-71.0448975,42.3461303, col =c("black"), cex = 2)
```



Thoughts for the road

- The story of modeling analyses can be told almost completely visually
- There is a reason many people like bars
- A little effort on cleanup can go a long way on understanding
- Color, intensity, size, slope and position help accentuate key points
- Don’t feel pressured to do too much with one visual
- A pretty good location was chosen for this conference