SPARSITY BLUES

BRIAN A. FANNIN

MARCH 21, 2018

OVERVIEW

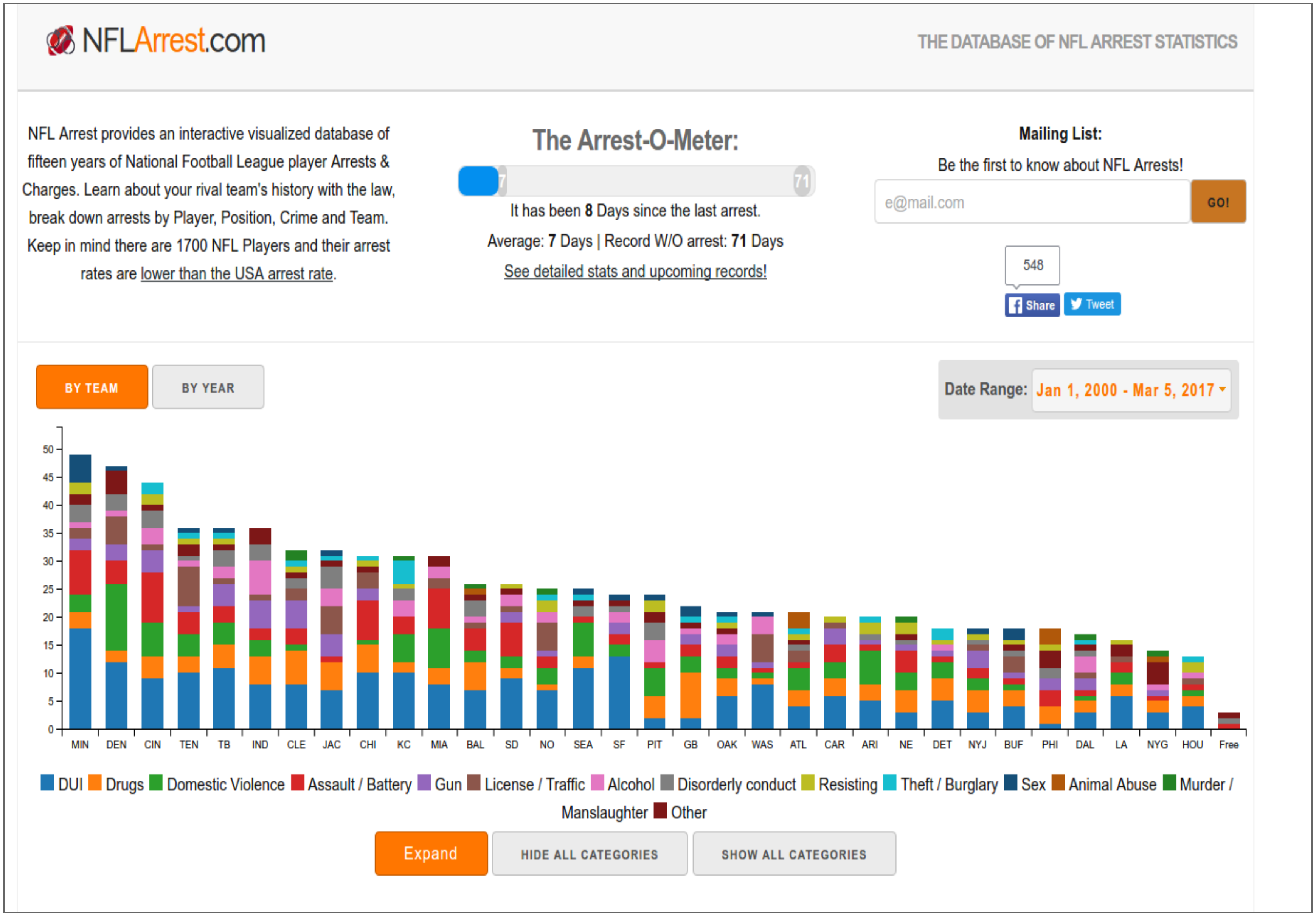
- Where did this talk come from?
- Categorical vs continuous data
- Naive Bayes
- Decision trees
- Multiple Correspondance Analysis
- Let’s model!

ORIGINS

ORIGINS

- I gave a **talk** last year about APIs.
- As an afterthought, I tried to fit a model.
- The fits were challenging because the data was largely categorical.

THE DATA


NFLArrest.com
THE DATABASE OF NFL ARREST STATISTICS
NFL Arrest provides an interactive visualized database of fifteen years of National Football League player Arrests & Charges. Learn about your rival team's history with the law, break down arrests by Player, Position, Crime and Team. Keep in mind there are 1700 NFL Players and their arrest rates are lower than the USA arrest rate.
The Arrest-O-Meter:
It has been 8 Days since the last arrest.
Average: 7 Days | Record W/O arrest: 71 Days
See detailed stats and upcoming records!
Mailing List:
Be the first to know about NFL Arrests!
e@mail.com
GO!
548
Share
Tweet
BY TEAM
BY YEAR
Date Range: Jan 1, 2000 - Mar 5, 2017
0
5
10
15
20
25
30
35
40
45
50
MIN DEN CIN TEN TB IND CLE JAC CHI KC MIA BAL SD NO SEA SF PIT GB OAK WAS ATL CAR ARI NE DET NYJ BUF PHI DAL LA NYG HOU Free
DUI Drugs Domestic Violence Assault / Battery Gun License / Traffic Alcohol Disorderly conduct Resisting Theft / Burglary Sex Animal Abuse Murder / Manslaughter Other
Expand
HIDE ALL CATEGORIES
SHOW ALL CATEGORIES

BEFORE ANYONE GETS CARRIED AWAY…

From nflarrests.com:

> *Keep in mind there are 1700 NFL Players and their arrest rates are lower than the USA arrest rate.*

Also: arrest != conviction

WHAT I TRIED TO MEASURE

I tried to measure whether a player would get a second arrest.

- Rate of 1st arrest requires player statistics for each season, which means a second source.
- I'm lazy. Let's check rate of second arrest.

JUST THE BASIC FACTS

- Number of players who've been arrested: 651
- Number of players w/more than one arrest: 146
- Probability of second arrest: 22.4%

So there is a small probability of having more than one arrest. Compare this to Bailey/Simon probability of second accident.

CATEGORICAL VS CONTINUOUS DATA

THERE ARE ONLY 2 KINDS OF DATA

- Continuous
- Categorical
  - Ordinal
  - Unordered
- OK, three would be a mixed distribution (zero-inflated, etc.)

Outcomes (for supervised learning) are either categorical or continuous (classification or regression).

CATEGORICAL DATA

- Gender
- Smoking
- Safe driver program
- Drug testing policy
- ...

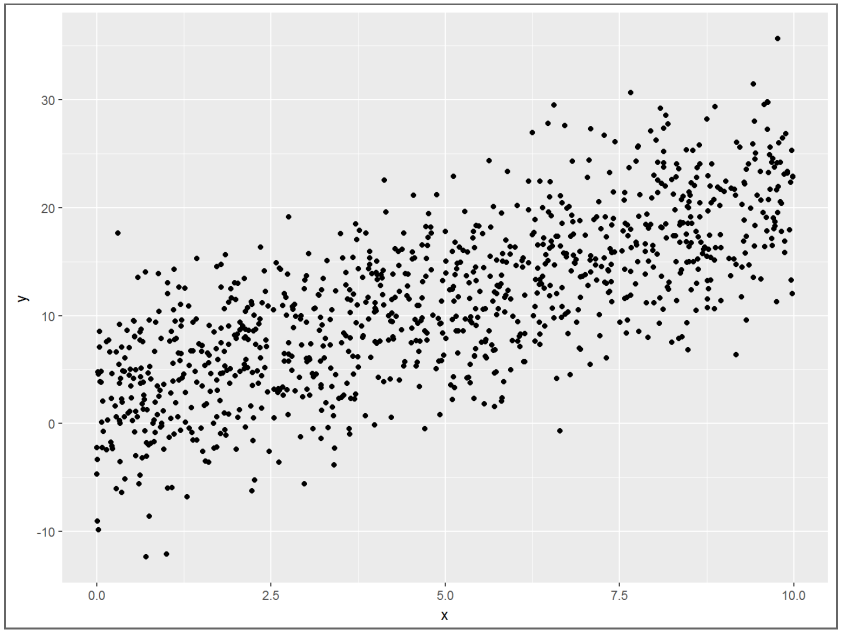
Basically anything to which you could apply a schedule mod. And also:

- Class code
- Territory
- Zip code

And those are just the ones that might be in a rating manual.

CONTINUOUS OUTCOME

```
sims <- 1e3
tbl_linear <- tibble(
    x = runif(sims, 0, 10)
  , e = rnorm(sims, sd = 5)
) %>%
  mutate(
    y = 1.5 + 2 * x + e
  )
```

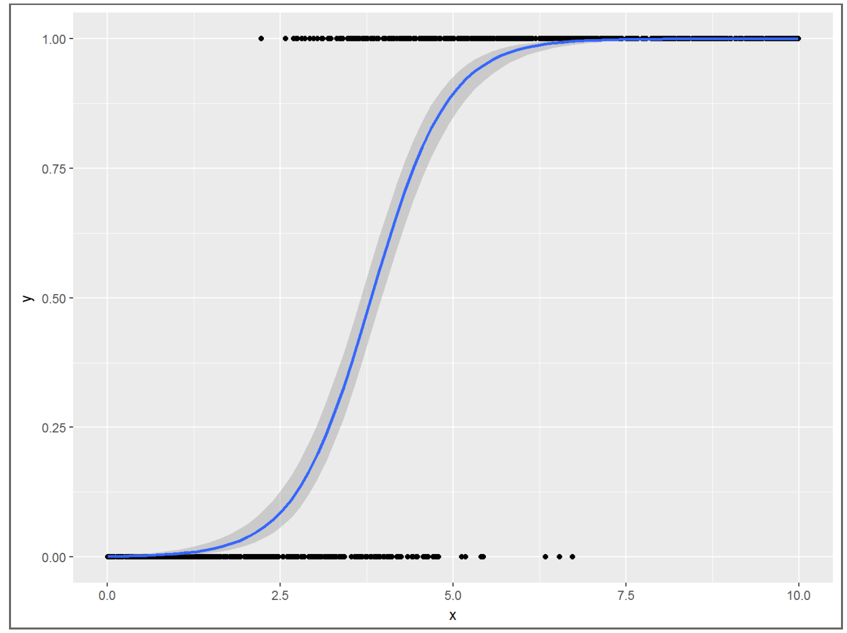


CATEGORICAL OUTCOME

- Logistic regression
- Support vector machine
- Tree methods

CATEGORICAL OUTCOME

```
tbl_logistic <- tbl_linear %>%
  mutate(
      e = rlogis(1e3)
    , latent = -7.5 + 2 * x + e
    , y = as.integer(latent > 0)
  )
```

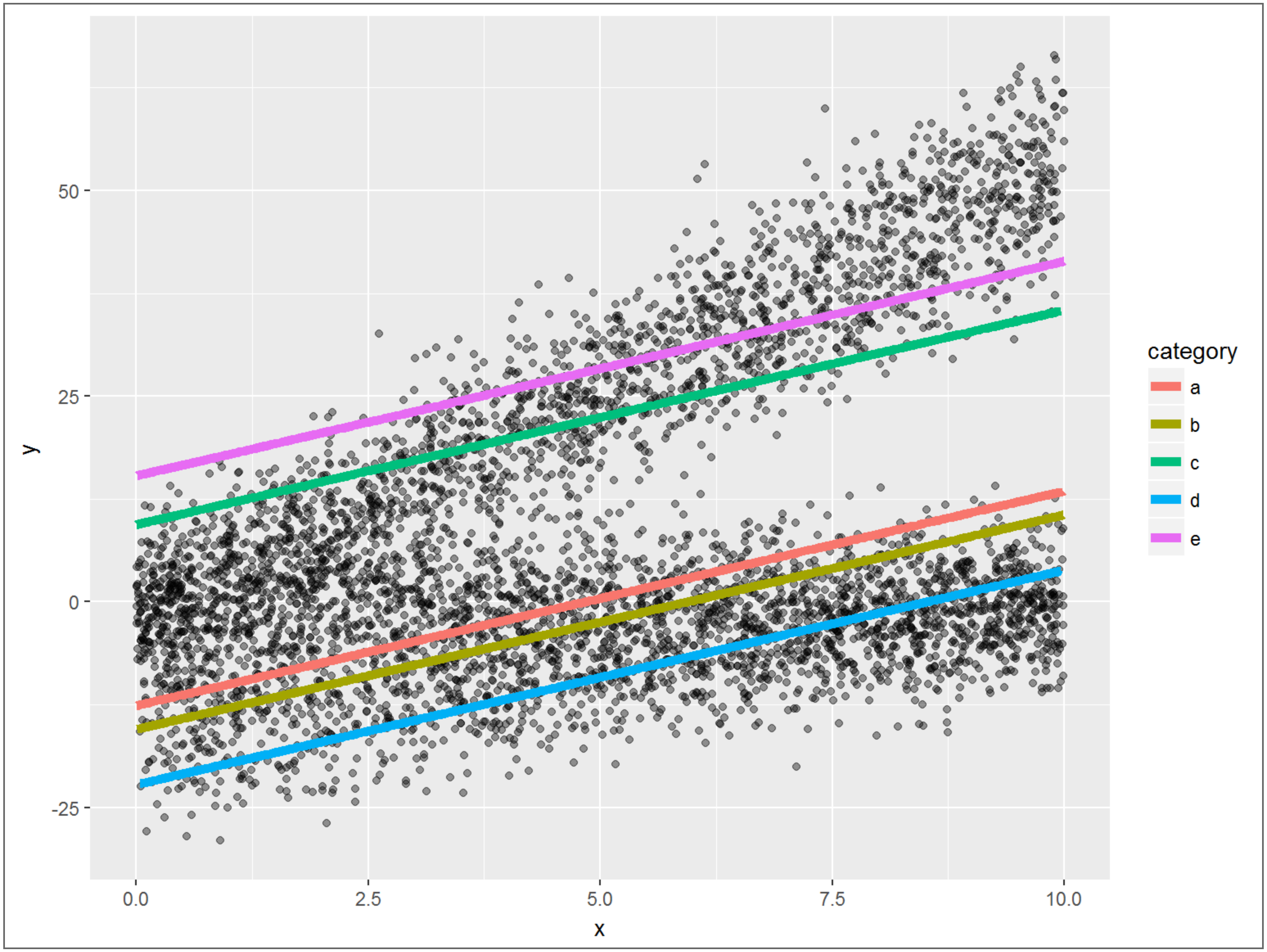


CATEGORICAL PREDICTORS IN A LINEAR MODEL

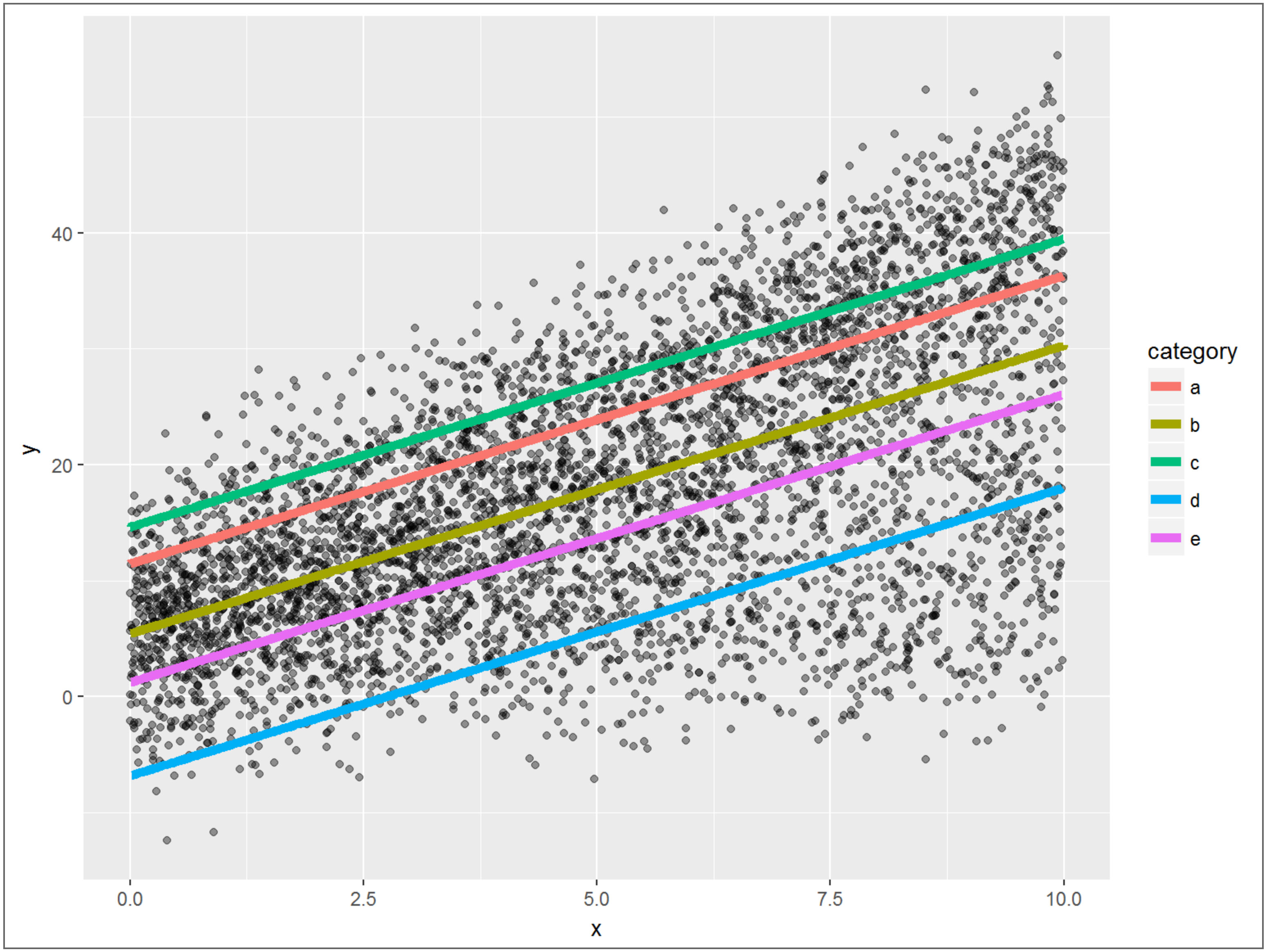
```
set.seed(1234)
tbl_one_cat <- function(cat_label = 'a', sims = 1e3) {
  slope <- rnorm(1, 2, 2)
  intercept <- rnorm(1, 0, 10)
  tibble(
    x = runif(sims, 0, 10)
  , e = rnorm(sims, sd = 5)
  , category = rep(cat_label, sims)
) %>%
  mutate(
    y = intercept + slope * x + e
  )
}

tbl_cat <- map_dfr(letters[1:5], tbl_one_cat)
```

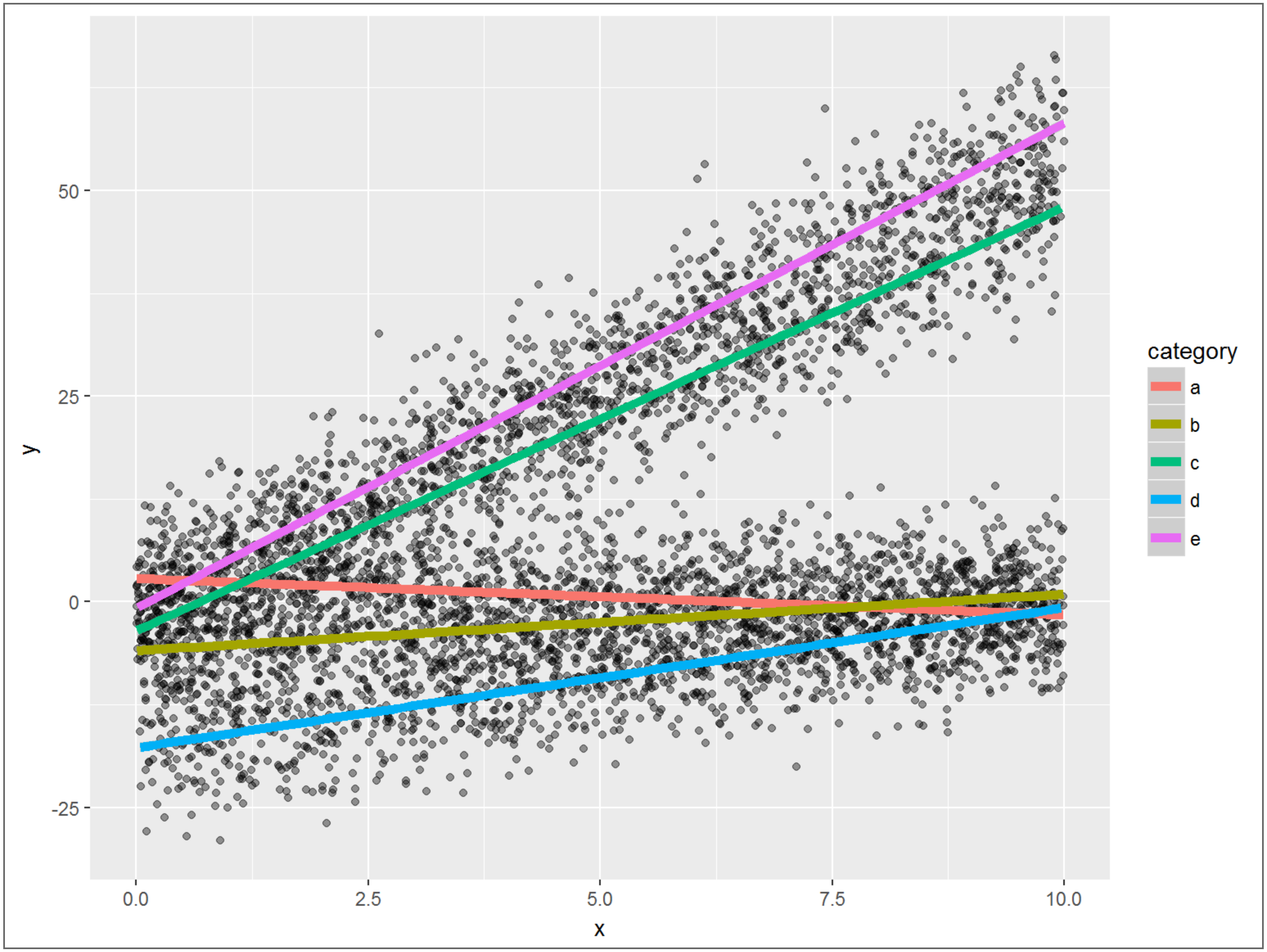
DIFFERENT INTERCEPTS



DIFFERENT SLOPES



OR BOTH



ISSUES

- Grouped data is looped data
- Handle this with credibility/hierarchical models
- What if we *only* have categorical predictors?

THE DESIGN MATRIX

categorya	categoryb	categoryc	categoryd	categorye	categorya:x	categoryb:x
1	0	0	0	0	8.6091538	0
1	0	0	0	0	6.4031061	0
1	0	0	0	0	0.0949576	0
1	0	0	0	0	2.3255051	0
1	0	0	0	0	6.6608376	0
1	0	0	0	0	5.1425114	0
1	0	0	0	0	6.9359129	0
1	0	0	0	0	5.4497484	0
1	0	0	0	0	2.8273358	0
1	0	0	0	0	9.2343348	0

Let’s try some non-linear methods

NAIVE BAYES

BAYES

$$Pr(Y = y|X = x) = \frac{Pr(Y = y) * Pr(X = x|Y = y)}{Pr(X = x)}$$

FIT

```r
library(naivebayes)

fit_nb <- naive_bayes(
    formula = MultiArrest ~ PositionType
  , data = tbl_players
)
```

```
## ===================== Naive Bayes =====================
## Call:
## naive_bayes.formula(formula = MultiArrest ~ PositionType, data = tbl_players)
##
## A priori probabilities:
##
##     FALSE      TRUE
## 0.7757296 0.2242704
##
## Tables:
##
## PositionType       FALSE        TRUE
##            D 0.534653465 0.541095890
##            O 0.457425743 0.424657534
##            S 0.007920792 0.034246575
```

CAN WE WORK THAT OUT MANUALLY?

```
prior_y <- sum(tbl_players$MultiArrest) / nrow(tbl_players)
prob_x <- sum(tbl_players$PositionType == 'D') / nrow(tbl_players)
tbl_cond <- tbl_players %>% filter(MultiArrest)
prob_x_cond <- sum(tbl_cond$PositionType == 'D') / nrow(tbl_cond)
prior_y * prob_x_cond / prob_x
## [1] 0.226361
predict(fit_nb, type = 'prob')[1, 'TRUE']
##     TRUE
## 0.226361

prior_y
## [1] 0.2242704
```

TWO CATEGORIES

One:

$$Pr(Y = y|X = x) = \frac{Pr(Y = y) * Pr(X = x|Y = y)}{Pr(X = x)}$$

Two:

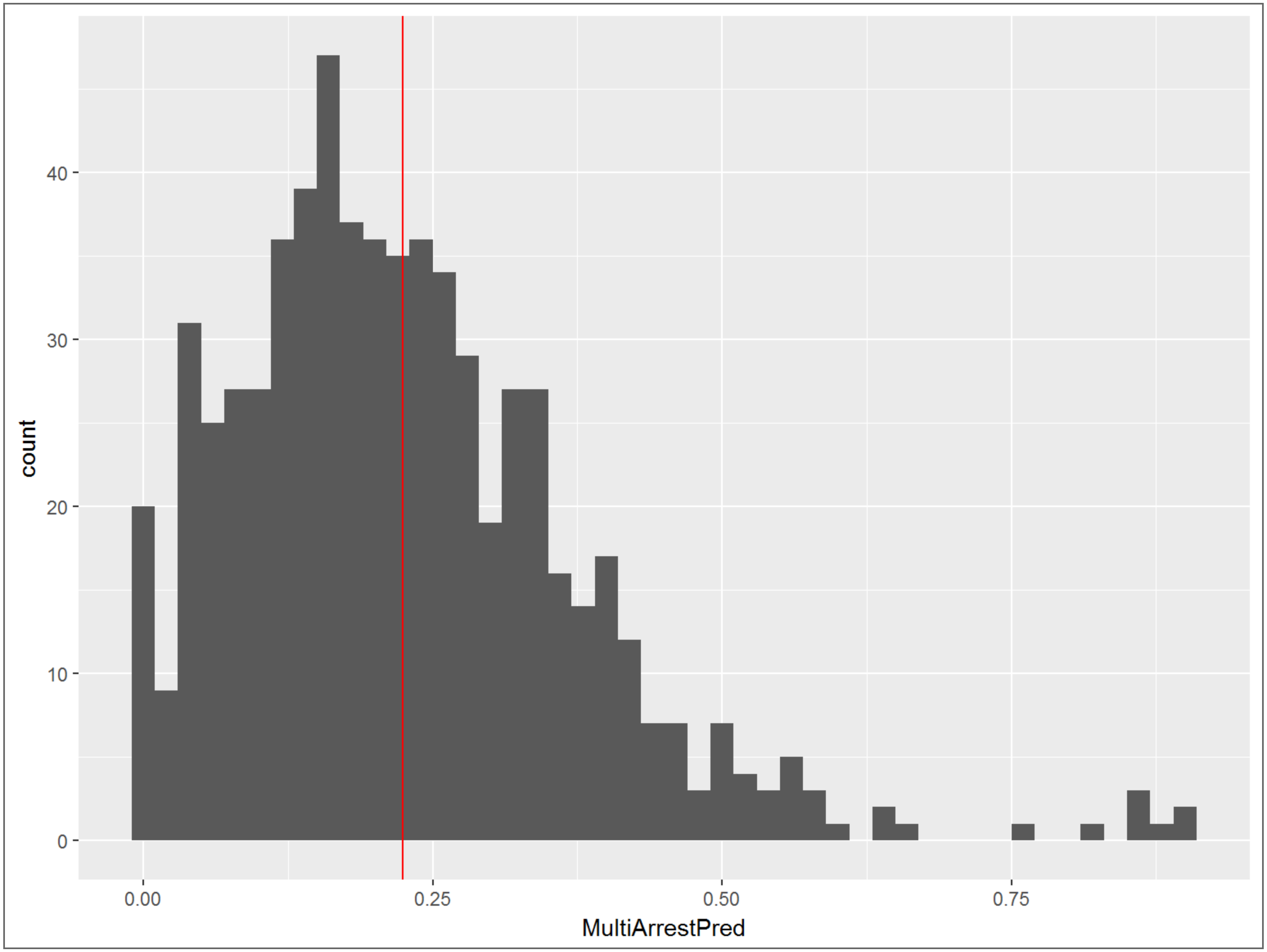
$$Pr(Y = y|X = x, Z = z)$$

$$= \frac{Pr(Y = y) * Pr(X = x|Y = y) * Pr(Z = z|Y = y)}{Pr(X = x) * Pr(Z = z)}$$

HOW ABOUT A LOT OF CATEGORIES?

```
fit_nb <- naive_bayes(
    formula = MultiArrest ~ TeamAbbr + Conference + Division + Position
        + PositionType + Encounter + CrimeCategory + ArrestSeasonState
    + DayOfWeek
  , data = tbl_players
)
```

HOW DO OUR PLAYERS LOOK?



NAIVE BAYES

- Often used in text processing
- Great for a sparse matrix
- It is ‘naive’ because we assume independence between categories

A DECISION TREE

CHARACTERISTICS OF A DECISION TREE

- Divides a sample into regions/subsets
- The ‘prediction’ is a function (usually the mean) of some value within each category
- Membership is assessed by computing some measure of fit. If a split improves the criteria, then it is made.
- Forward only, ‘greedy’
- Number of levels and other criteria control the size and shape of the tree

MEASURES OF FIT

For regression:

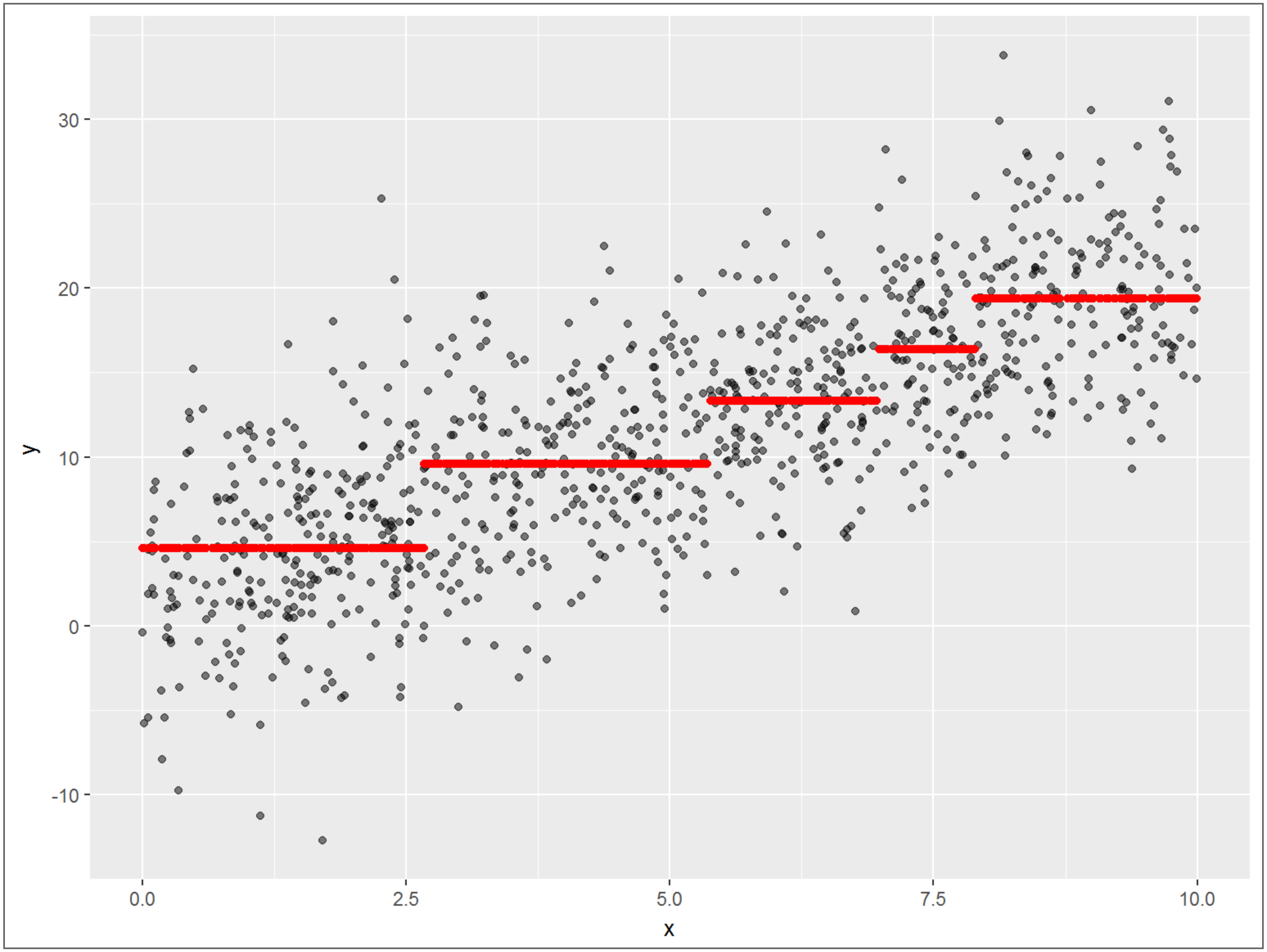
- Construct regions which minimize residual sum of squares

For classification:

- Construct regions which maximize homogeneity

LINEAR FIT

```
library(tree)
fit_tree <- tree::tree(formula = y ~ x, data = tbl_linear)
summary(fit_tree)
##
## Regression tree:
## tree::tree(formula = y ~ x, data = tbl_linear)
## Number of terminal nodes:  5
## Residual mean deviance:  22.64 = 22530 / 995
## Distribution of residuals:
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -17.290000  -3.194000   0.000456   0.000000   3.007000  20.740000
```


30
20
10
0
-10
y
0.0
2.5
5.0
7.5
10.0
x

CATEGORICAL FIT

a	b	output
red	black	1
red	white	1
red	black	0
blue	white	0
blue	black	0

TWO MEASURES OF HOMOGENEITY

$$Gini = \sum p * (1 - p)$$

$$Entropy = -\sum p * log(p)$$

MEASURE TOTAL ENTROPY

```
entropy <- function(y) {
  tbl <- tibble(y) %>%
    group_by(y) %>%
    summarise(prob = n()) %>%
    mutate(
        prob = prob / sum(prob)
      , ent = -prob * log(prob))

  tbl$ent %>% sum()
}
```

MEASURE ENTROPY POST-SPLIT

```r
entropy_post <- function(tbl, out_col, split_col) {

  split_col <- enquo(split_col)
  out_col <- enquo(out_col)

  tbl %>%
    group_by(!! split_col) %>%
    summarise(
        ent = entropy(!! out_col)
      , group_pct = n() / nrow(tbl)
    ) %>%
    ungroup() %>%
    summarise(
      ent_post = sum(ent * group_pct)
    ) %>%
    pull(ent_post)
}
```

WHICH COLUMN WORKS BETTER ON OUR TOY DATA?

```
entropy(tbl_toy$output)
## [1] 0.6730117

tbl_toy %>%
  entropy_post(output, a)
## [1] 0.3819085

tbl_toy %>%
  entropy_post(output, b)
## [1] 0.6591674
```

a	b	output
red	black	1
red	white	1
red	black	0
blue	white	0
blue	black	0

POTENTIAL NODE SPLITS

```
entropy(tbl_players$MultiArrestNum)
## [1] 0.5322599
tbl_players %>%
  entropy_post(MultiArrestNum, PositionType)
## [1] 0.528498

tbl_players %>%
  entropy_post(MultiArrestNum, Season)
## [1] 0.5092358

tbl_players %>%
  entropy_post(MultiArrestNum, ArrestSeasonState)
## [1] 0.5313111
```

WHAT SPLITS?

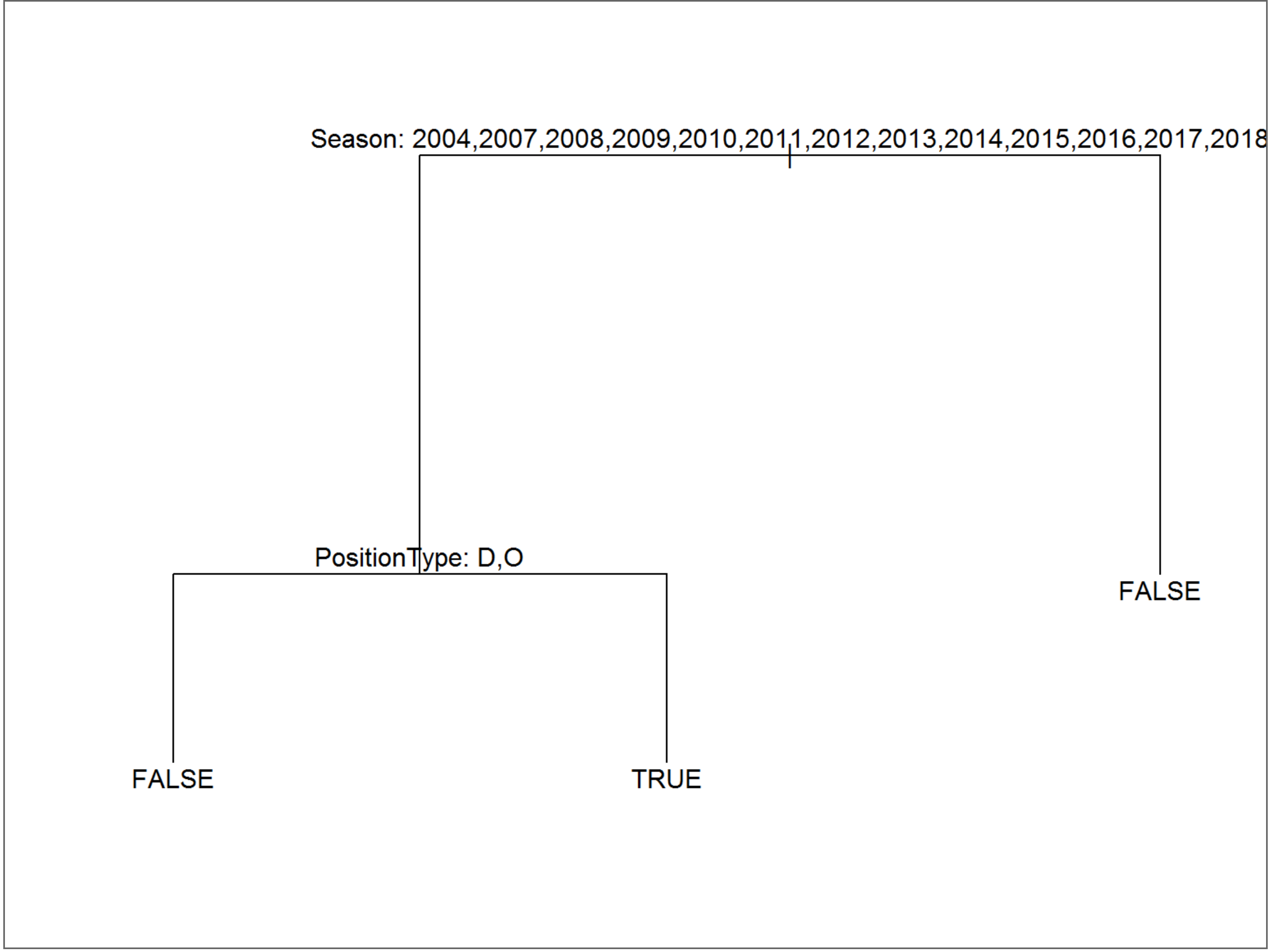
```
library(rpart)

fit_tree <- tree(
    data = tbl_players
  , formula = MultiArrestFactor ~ PositionType + Season + ArrestSeasonState)

summary(fit_tree)
##
## Classification tree:
## tree(formula = MultiArrestFactor ~ PositionType + Season + ArrestSeasonState,
##     data = tbl_players)
## Variables actually used in tree construction:
## [1] "Season"       "PositionType"
## Number of terminal nodes:  3
## Residual mean deviance:  1.021 = 661.6 / 648
## Misclassification error rate: 0.2197 = 143 / 651
```

PLOT THE TREE

```
plot(fit_tree)
text(fit_tree, pretty = 0)
```



Season: 2004,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018
PositionType: D,O
FALSE
FALSE
TRUE

NOTE

1. Full disclosure: I used both `rpart` and `tree` for the fit. For reasons that I've not yet debugged, `rpart` gave me no nodes.
2. A package's insistence on using factors may cause you to lose your mind.

BAGGING/RANDOM FORESTS

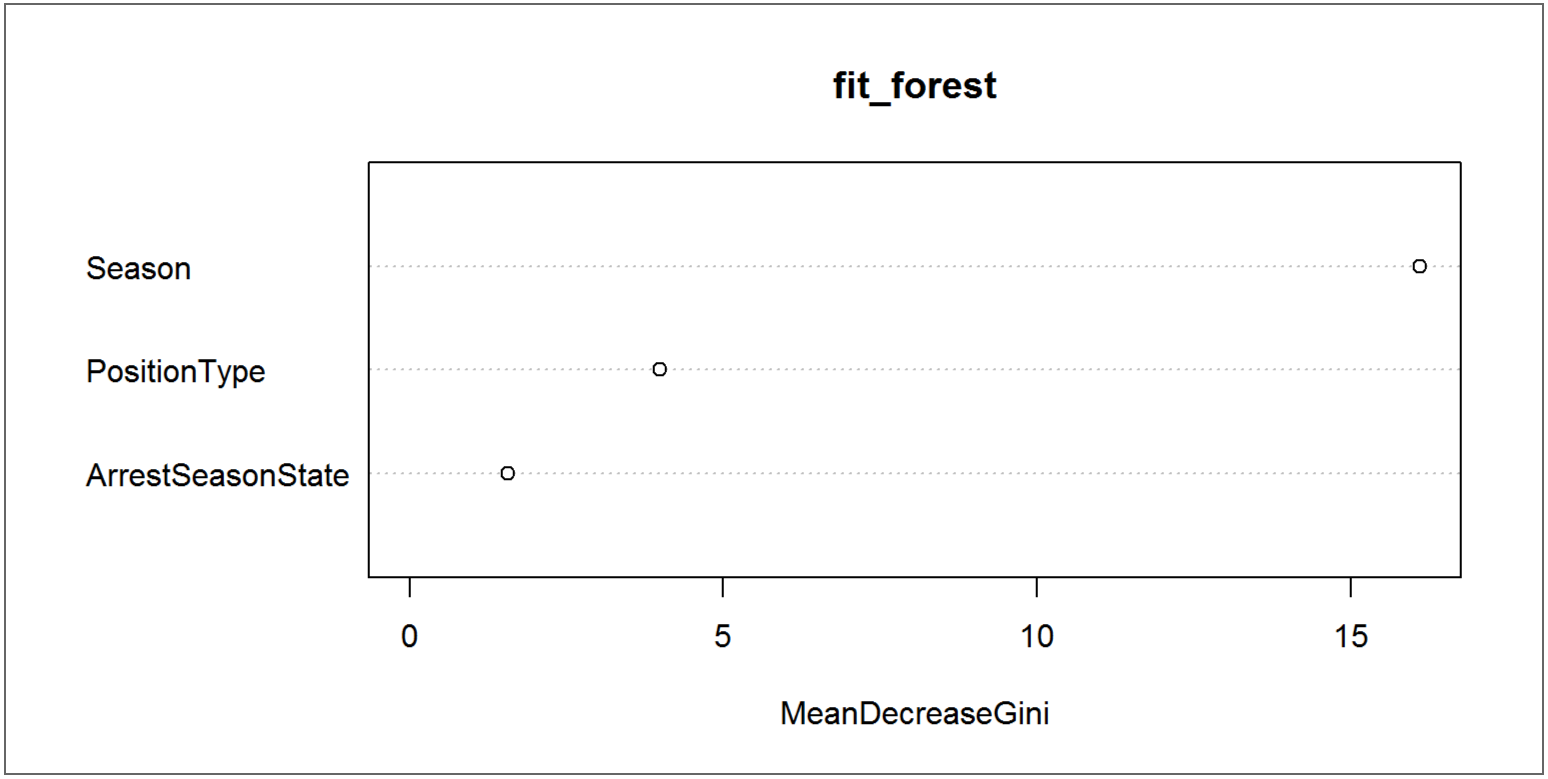
- Avoid overfit by bootstrapping
- Fit hundreds of resampled trees
- Take the average of results
- We don’t get that sweet tree plot

RANDOM FOREST

```
library(randomForest)
fit_forest <- randomForest(
    formula = MultiArrestFactor ~ PositionType + Season + ArrestSeasonState
  , data = tbl_players
)
```

VARIABLE IMPORTANCE

```
varImpPlot(fit_forest)
```



MULTIPLE CORRESPONDENCE ANALYSIS

WHAT IS MCA?

- PCA, but for categories
- CA, but for multiple variables

WHY MCA?

- Dimensionality reduction
- Could also consider (hierarchical) cluster analysis
- Others?

HOW DOES IT WORK?

- Candidly, I can’t easily explain it.
- Creates a “complete disjunctive table”, i.e. a “one hot encoding” table
- This creates points in a high-dimensional space
- Synthesizes new dimensions which capture the most variance between the points

COMPLETE DISJUNCTIVE TABLE

id	metro	region
1	urban	north
2	urban	south
3	rural	east
4	urban	north

CDT, OR “ONE-HOT ENCODING”

```
tbl_toy_mca_one_hot <- tbl_toy_mca %>%
  gather(category, value, -id) %>%
  unite(cdt, -id) %>%
  mutate(count = 1L) %>%
  tidyr::spread(cdt, count, fill = 0L)

tbl_toy_mca_one_hot %>% knitr::kable()
```

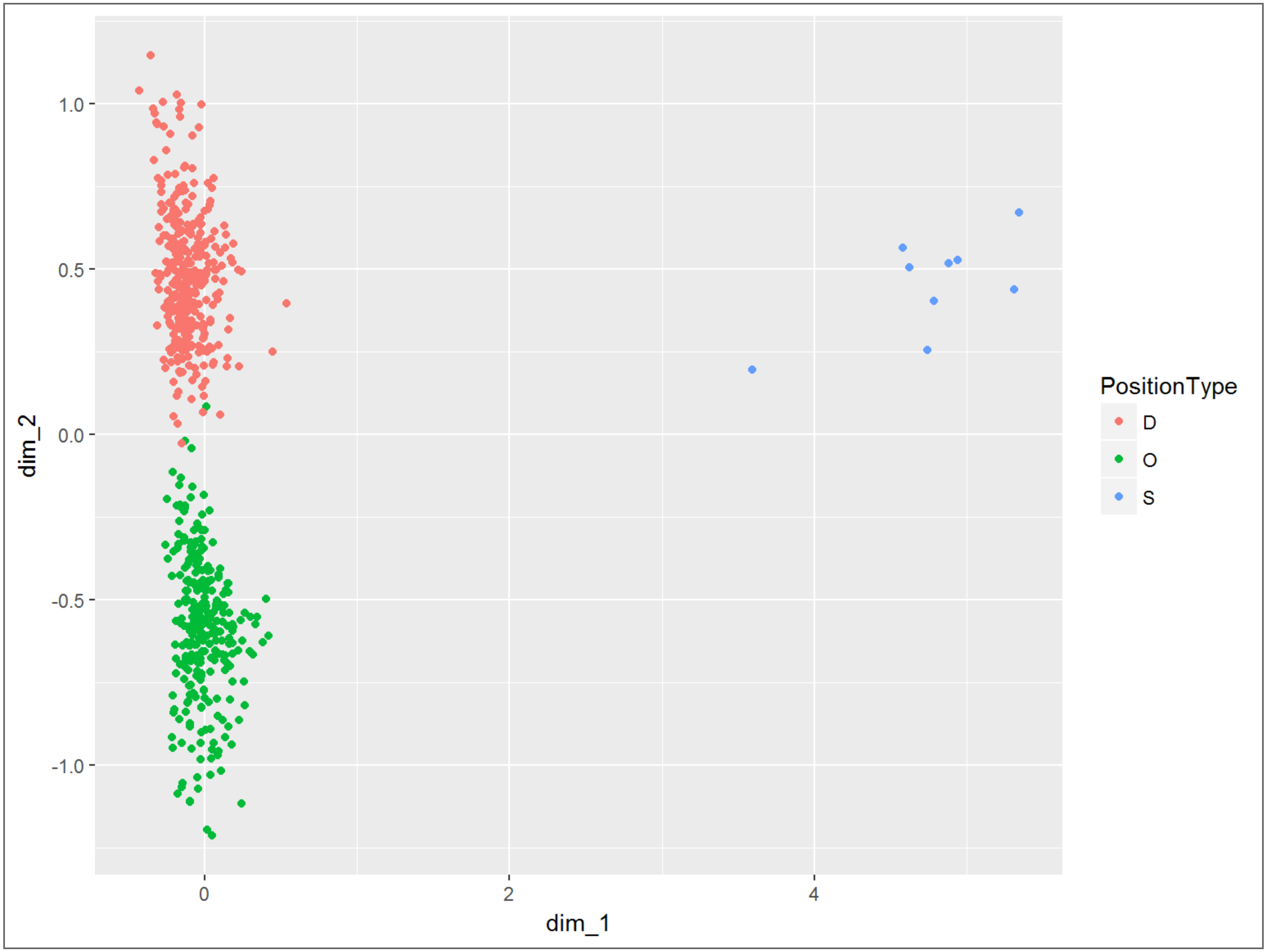
id	metro_rural	metro_urban	region_east	region_north	region_south
1	0	1	0	1	0
2	0	1	0	0	1
3	1	0	1	0	0
4	0	1	0	1	0

EXTRACT DATA FOR PROCESSING

```r
tbl_cats <- tbl_players %>%
  ungroup() %>%
  select(
      CrimeCategory, ArrestSeasonState, Conference
    , Division, DayOfWeek, Outcome, Position, PositionType
    , Season) %>%
  mutate_if(is.character, as.factor)

library(FactoMineR)
fit_mca <- MCA(tbl_cats, graph = FALSE)
```

VISUALIZE IN THE REDUCED DIMENSIONS



MCA: CATEGORICAL -> CONTINUOUS

Call: glm(formula = MultiArrestNum ~ 0 + dim_1 + dim_2, family = binomial(), data = tbl_players)

Deviance Residuals: Min 1Q Median 3Q Max
-1.815 -1.182 -1.158 -1.129 1.236

Coefficients: Estimate Std. Error z value Pr(>|z|)
dim_1 0.27532 0.16245 1.695 0.0901 . dim_2 -0.05509 0.13790 -0.399 0.6895
— Signif. codes: 0 '' ***0.001*** '' *0.01* '' 0.05 '.' 0.1 '' 1

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 902.48  on 651  degrees of freedom
```

Residual deviance: 898.83 on 649 degrees of freedom AIC: 902.83

Number of Fisher Scoring iterations: 4

LET'S MODEL!

HOW WE'LL MODEL

1. Pick a performance measure
2. Setup cross-validation
3. Train some models
4. Measure performance

OUR PERFORMANCE MEASURE

Misclassification rate

Other options:

- True positive rate
- False positive rate
- Other confusion matrix metrics
- Area under the curve (AUC): A number close to 1 is good

MEASURES

```r
misclass <- function(tbl_test, fit_obj) {
  tbl_test <- tbl_test %>%
    mutate(
        pred = predict(fit_obj, type = 'class', newdata = tbl_test)
      , misclass = pred != MultiArrestFactor
    )
  sum(tbl_test$misclass) / nrow(tbl_test)
}
```

N-FOLD CROSS VALIDATION

```r
library(modelr)
set.seed(1234)
tbl_folds <- crossv_kfold(tbl_players, k = 10)
```

TBL_FOLDS

```
tbl_folds %>% head()
## # A tibble: 6 x 3
##   train          test           .id  
##   <list>         <list>         <chr>
## 1 <S3: resample> <S3: resample> 01   
## 2 <S3: resample> <S3: resample> 02   
## 3 <S3: resample> <S3: resample> 03   
## 4 <S3: resample> <S3: resample> 04   
## 5 <S3: resample> <S3: resample> 05   
## 6 <S3: resample> <S3: resample> 06
```

WHAT'S IN TBL_FOLDS?

- Each row in the tibble holds:
- a training `resample` object
- a test `resample` object
- an id

A `resample` object is a list which contains a data frame and a vector of row indices.

```
tbl_folds$train[[1]] %>% class()
## [1] "resample"
```

ASSESS ONE FOLD

```
assess_fold <- function(obj_train, obj_test, method, the_formula) {
  tbl_train <- obj_train %>% as.data.frame()
  tbl_test <- obj_test %>% as.data.frame()

  fit <- do.call(
      method
    , args = list(formula = the_formula, data = tbl_train))

  misclass(tbl_test, fit)

}

one_fold_misclass <- assess_fold(
    tbl_folds$train[[1]]
  , tbl_folds$test[[1]]
  , tree::tree
  , as.formula('MultiArrestFactor ~ PositionType + Season'))
```

ASSESS ALL FOLDS

```
cross_validate <- function(formula, tbl_folds, method) {
  map2_dbl(
    tbl_folds$train
  , tbl_folds$test
  , assess_fold
  , method
  , formula
  ) %>% mean()
}

misclasses <- cross_validate(
    as.formula('MultiArrestFactor ~ PositionType + Season')
  , tbl_folds
  , tree::tree
)

misclasses <- cross_validate(
    as.formula('MultiArrestFactor ~ PositionType + Season')
  , tbl_folds
  , naive_bayes
)
```

MAKE FORMULAS

```r
make_formula <- function(predictors, target, intercept = TRUE) {
  str_predictors <- paste(predictors, collapse = '+')
  if (intercept) {
    str_formula <- paste(target, '~ 1 + ')
  } else {
    str_formula <- paste(target, '~')
  }

  str_formula <- paste(str_formula, str_predictors)
  as.formula(str_formula)
}
```

A FEW FORMULAS

```
the_formulas <- list(
      c('PositionType', 'Season')
    , c('PositionType', 'Season', 'DayOfWeek')
    , c('PositionType', 'Season', 'DayOfWeek')
    , c('PositionType', 'Season', 'DayOfWeek', 'Conference')
    , c('PositionType', 'Season', 'DayOfWeek', 'Conference', 'Division')
    , c('PositionType', 'Season', 'DayOfWeek', 'Conference', 'Division', 'TeamCity')
  ) %>%
  map(make_formula, 'MultiArrestFactor', intercept = FALSE) %>%
  as.vector()

tbl_models <- tibble(
  formula = the_formulas
)
```

OUR MODELS TIBBLE

formula
MultiArrestFactor ~ PositionType + Season
MultiArrestFactor ~ PositionType + Season + DayOfWeek
MultiArrestFactor ~ PositionType + Season + DayOfWeek
MultiArrestFactor ~ PositionType + Season + DayOfWeek + Conference
MultiArrestFactor ~ PositionType + Season + DayOfWeek + Conference + Division
MultiArrestFactor ~ PositionType + Season + DayOfWeek + Conference + Division + TeamCity

ASSESS ALL FOLDS, ALL FORMULAS, ALL MODELS

```
tbl_models <- tbl_models %>%
  mutate(
      misclass_tree = map_dbl(formula, cross_validate, tbl_folds, tree::tree)
    , misclass_nb = map_dbl(formula, cross_validate, tbl_folds, naive_bayes)
  )
```

formula	misclass_tree	misclass_nb
MultiArrestFactor ~ PositionType + Season	0.224289	0.2273427
MultiArrestFactor ~ PositionType + Season + DayOfWeek	0.224289	0.2304196
MultiArrestFactor ~ PositionType + Season + DayOfWeek	0.224289	0.2304196
MultiArrestFactor ~ PositionType + Season + DayOfWeek + Conference	0.224289	0.2319580
MultiArrestFactor ~ PositionType + Season + DayOfWeek + Conference + Division	0.224289	0.2319580
MultiArrestFactor ~ PositionType + Season + DayOfWeek + Conference + Division + TeamCity	0.224289	0.2550350

CONCLUSION

WHAT DID WE LEARN CHARLIE BROWN?

- Categorical data is ubiquitous, but tricky to model
- Non-linear approaches like tree-based methods and Naive Bayes look at categorical differently
- MCA can address “curse of dimensionality” with categorical data
- Let’s all keep doing this! Fitting categorical data is hard. Research is light.

Slides may be found here:

http://pirategrunt.com/sparsity_blues/#/

All of the code - even stuff you didn't see - is on GitHub

https://github.com/pirategrunt

THANK YOU!

Q&A

REFERENCES

- **http://www.gastonsanchez.com/visually-enforced/how-to/2012/10/13/MCA-in-R/**
- **http://rpubs.com/dgrtwo/cv-modelr**
- **https://drsimonj.svbtle.com/k-fold-cross-validation-with-modelr-and-broom**
- **http://www.casact.org/pubs/forum/09wforum/flynn_francis.pdf**