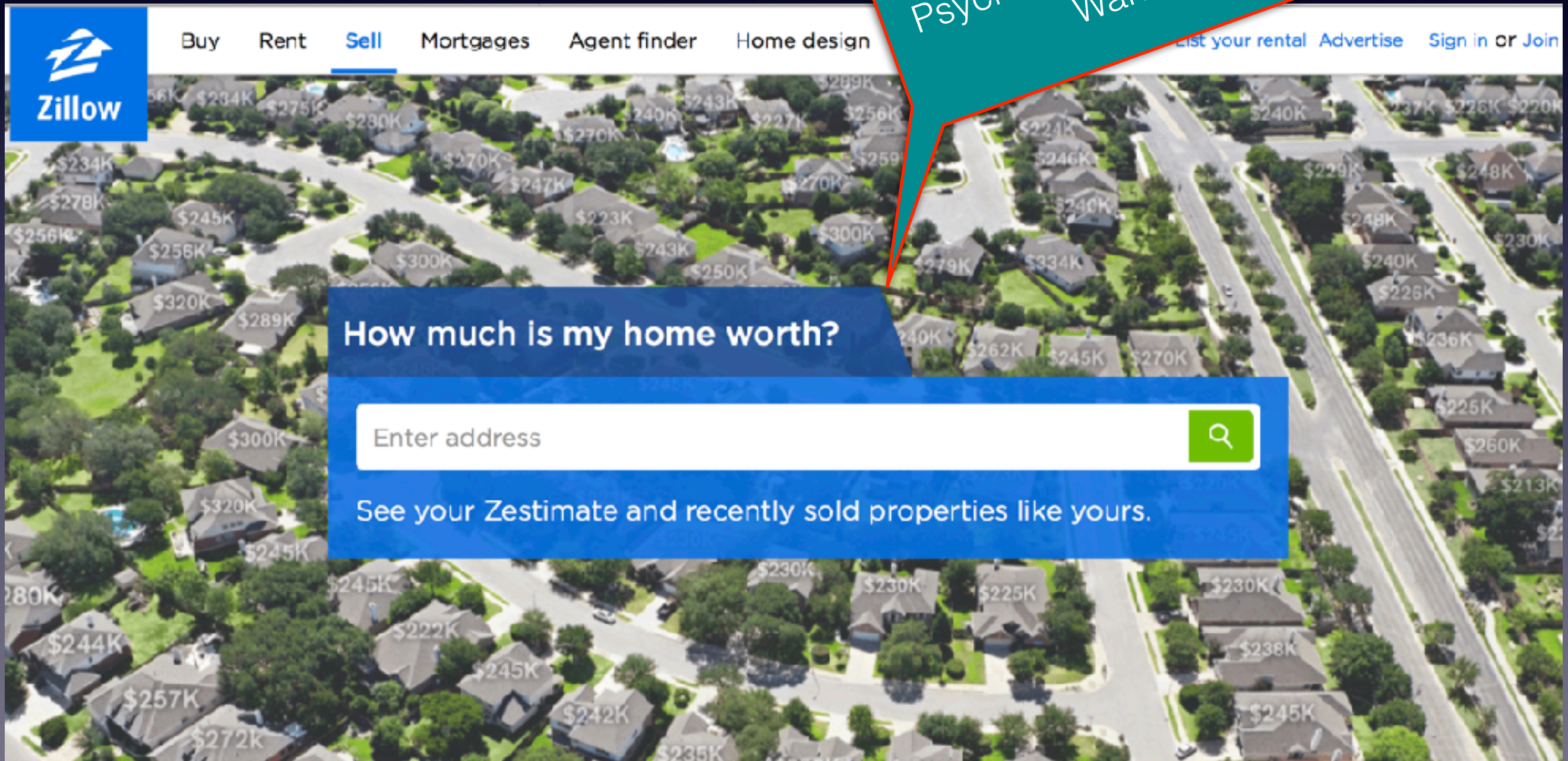


Zillow Case

Predict the “LogError” Kaggle / Zillow Challenge

Psyche! Not What They Want!



What are we predicting?
This ...

$$\logerror = \log(Zestimate) - \log(SalePrice)$$

$$\logerror = \log\left(\frac{Zestimate}{SalePrice}\right)$$

$$e^{\logerror} = \frac{Zestimate}{SalePrice}$$

General Overview

- Initial Exploratory Data Analysis
- Data Cleaning/Imputation
- Further Exploratory Data Analysis
- Modeling

Missingness & Imputation

Missingness Percent:

airconditioningtypeid	72.8154101	architecturalstyletypeid	99.7969662	basementsqft	99.9454646	bathroomcnt	0.3839587
bedroomcnt	0.3835567	buildingclasstypeid	99.5769487	buildingqualitytypeid	35.0637491	calculatedbathnbr	4.3183460
decktypeid	99.4273113	finishedfloor1squarefeet	93.2093044	calculatedfinishedsquarefeet	1.8613387	finishedsquarefeet12	9.2466645
finishedsquarefeet13	99.7430003	finishedsquarefeet15	93.6085718	finishedsquarefeet50	93.2093044	finishedsquarefeet6	99.2630017
fips	0.3831212	fireplacecnt	89.5271600	fullbathcnt	4.3183460	garagecarcnt	70.4119667
garagetotalsqft	70.4119667	hashottuborspa	97.6881413	heatingorsystemtypeid	39.4884526	latitude	0.3831212
longitude	0.3831212	lotsizesquarefeet	9.2488754	poolcnt	82.6634379	poolsizeum	99.0633847
pooltypeid10	98.7626025	pooltypeid2	98.9255387	pooltypeid7	83.7378991	propertycountylandusecode	0.4112599
propertylandusetypeid	0.3831212	propertyzoningdesc	33.7190898	rawcensustractandblock	0.3831212	regionidcity	2.1052071
regionidcounty	0.3831212	regionidneighborhood	61.2623806	regionidzip	0.4683077	roomcnt	0.3843942
storytypeid	99.9455986	threequarterbathnbr	89.5608594	typeconstructiontypeid	99.7739863	unitcnt	33.7572444
yardbuildingsqft17	97.3082359	yardbuildingsqft26	99.9113297	yearbuilt	2.0074923	numberofstories	77.1517782
fireplaceflag	99.8270477	structuretaxvaluedollarcnt	1.8418092	taxvaluedollarcnt	1.4253570	assessmentyear	0.3831882
landtaxvaluedollarcnt	2.2689473	taxamount	1.0468251	taxdelinquencyflag	98.1086132	taxdelinquencyyear	98.1085462
censustractandblock	2.5166010						

Is Anything Not Missing Values?

Missingness & Imputation

md.package() relationships

```
full %>% select(regionidcity, latitude, longitude, fips) %>% md.pattern()
```

```
##          latitude longitude  fips regionidcity
## 2922495         1         1     1           1    0
##  51410         1         1     1           0    1
##  11437         0         0     0           0    4
##          11437    11437 11437        62847 97158
```

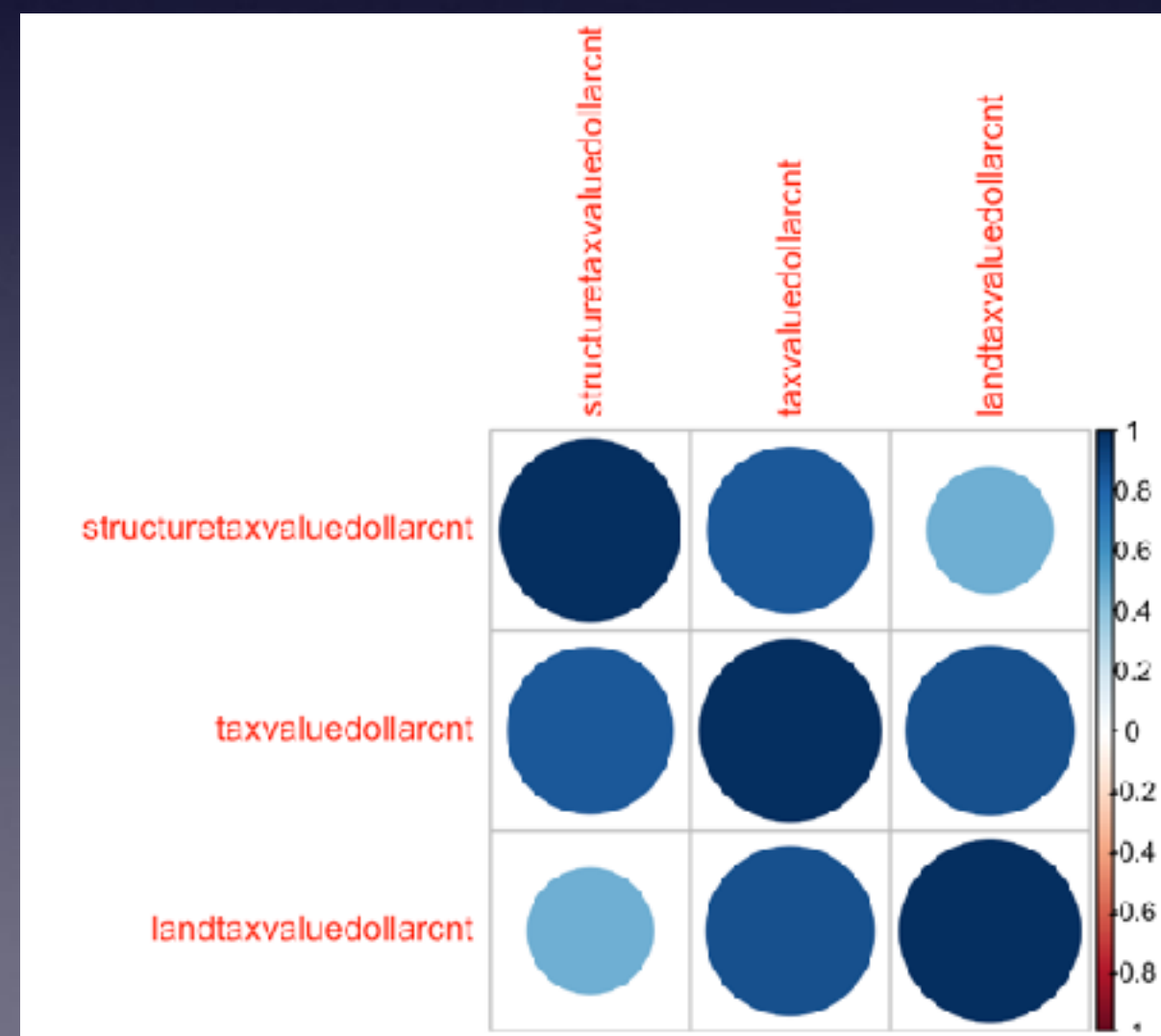
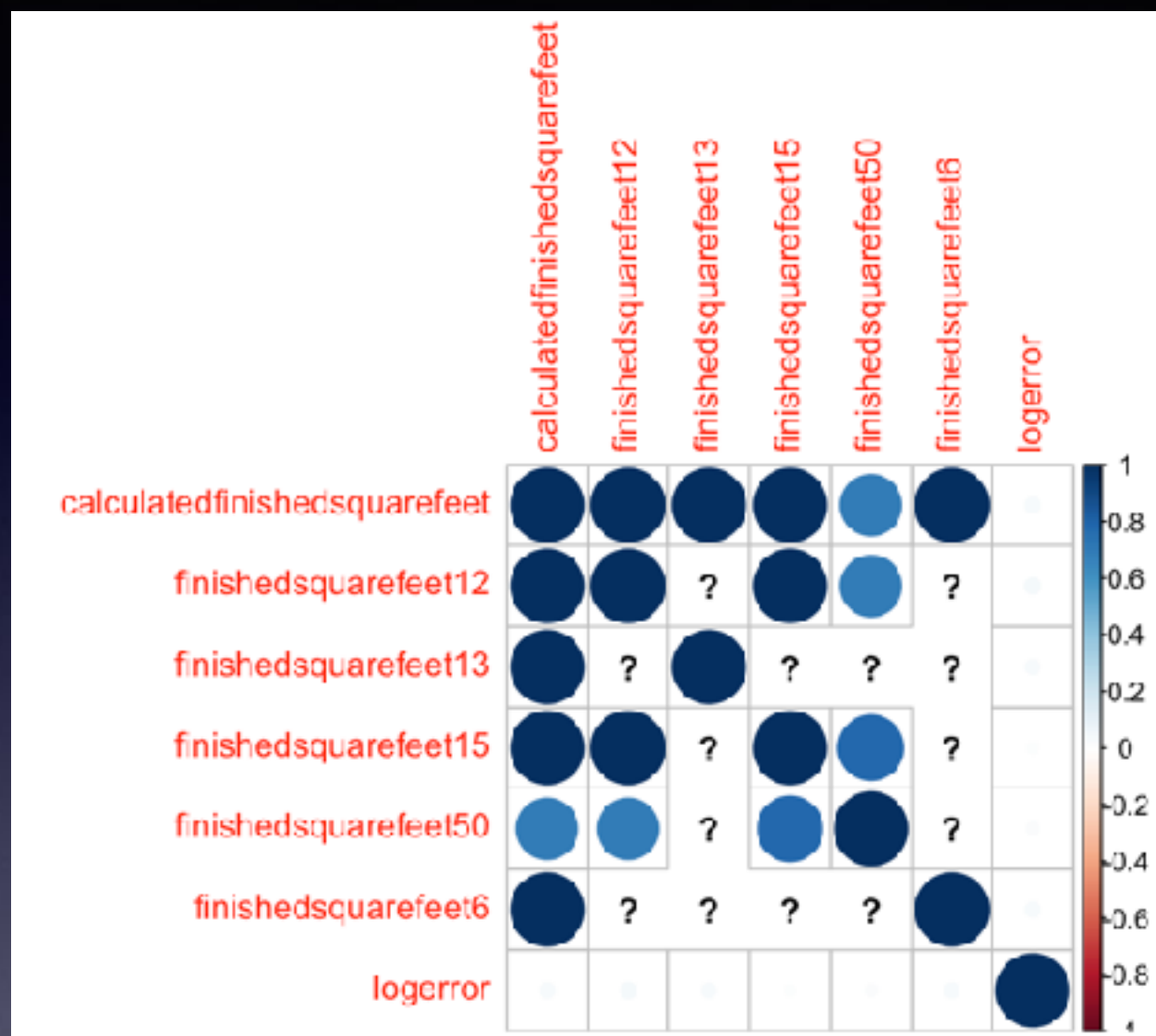
```
full %>% select(regionidcity, latitude, longitude, fips,
               regionidcounty, regionidneighborhood, regionidzip) %>% md.pattern() #
```

```
##          latitude longitude  fips regionidcounty regionidzip regionidcity regionidneighborhood
## 1156377         1         1     1           1           1           1           1           0
## 1765541         1         1     1           1           1           1           0           1
##      81         1         1     1           1           0           1           1           1
##  49444         1         1     1           1           1           0           0           2
##   496         1         1     1           1           0           1           0           2
##  1966         1         1     1           1           0           0           0           3
##  11437         0         0     0           0           0           0           0           7
##          11437    11437 11437        11437    13980        62847        1828884 1951459
```

```
calculatedfinishedsquarefeet finishedsquarefeet12 finishedsquarefeet50 finishedsquarefeet15 logerror finishedsquarefeet6 finishedsquarefeet13
5 1 0 1 1 1 0 0 3
6851 1 1 1 0 1 0 0 3
4 1 1 1 1 0 0 0 3
421 1 0 0 0 1 1 0 4
3559 1 0 0 1 1 0 0 4
33 1 0 0 0 1 0 1 4
78745 1 1 0 0 1 0 0 4
370 1 0 1 1 0 0 0 4
195493 1 1 1 0 0 0 0 4
21582 1 0 0 0 0 1 0 5
186869 1 0 0 1 0 0 0 5
7639 1 0 0 0 0 0 1 5
2428202 1 1 0 0 0 0 0 5
661 0 0 0 0 1 0 0 6
```

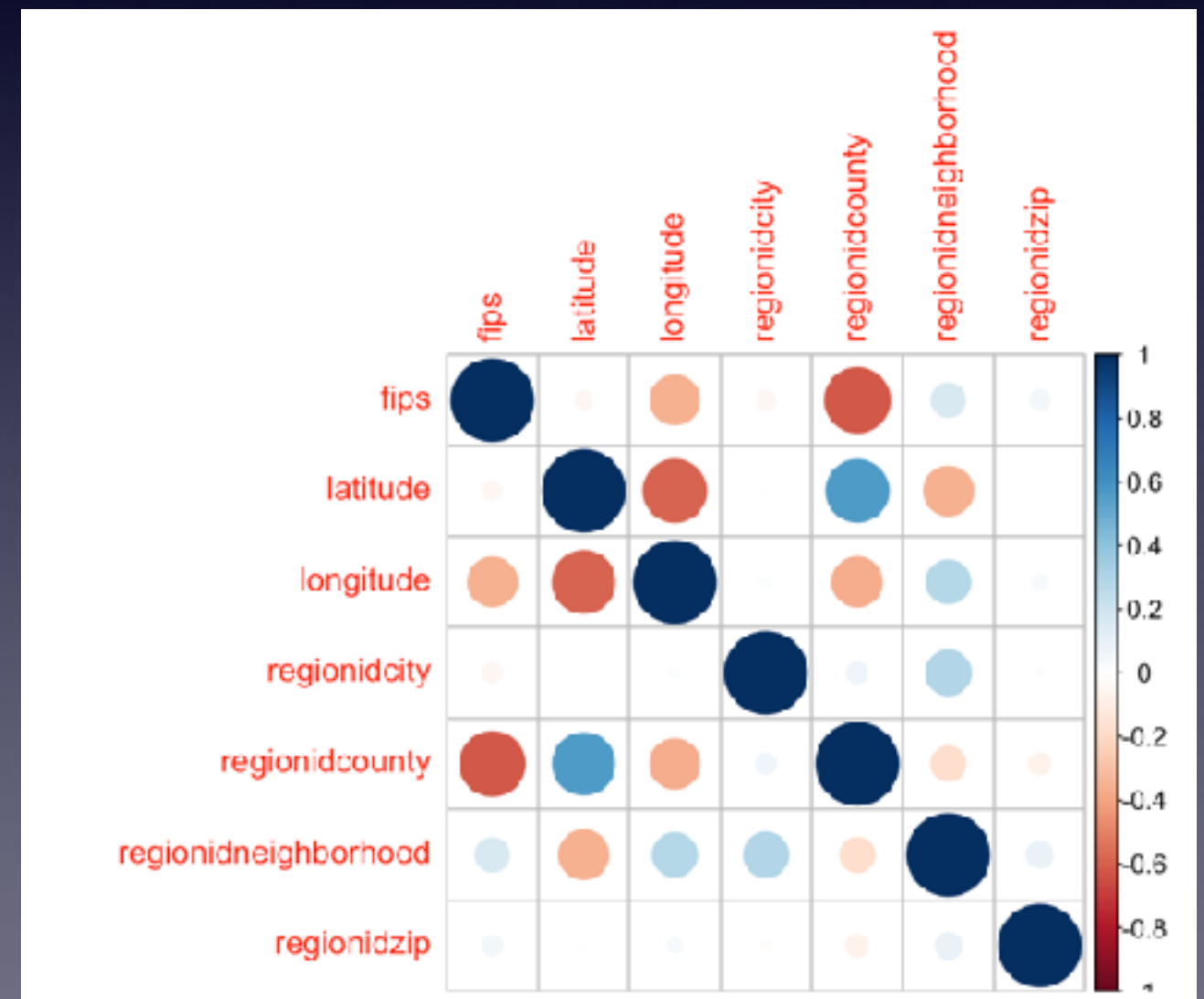
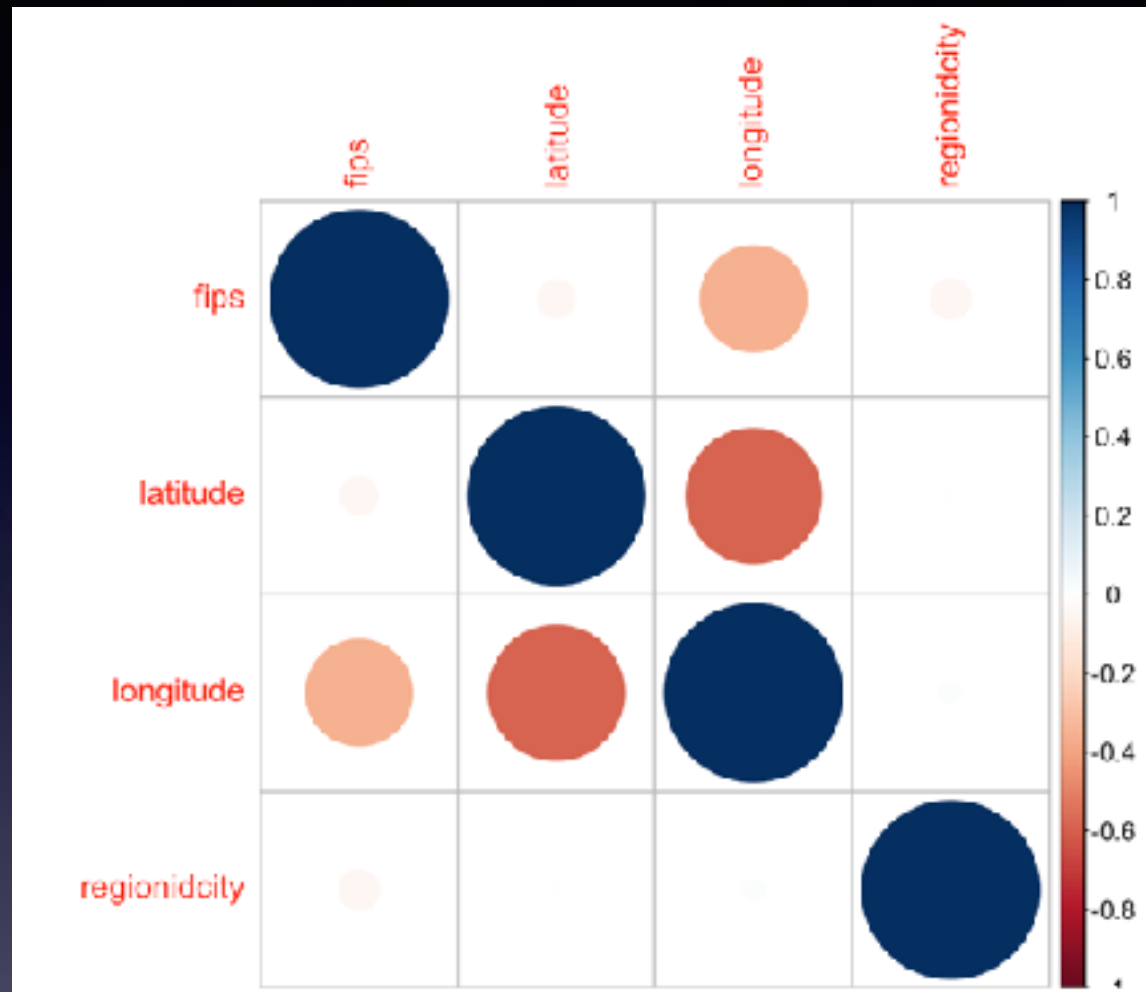
Missingness & Imputation

Correlations - Visually



Missingness & Imputation

Correlations - Visually



Looks Correlated - LM() Imputations?

```
fips.impute.model <- lm(fips ~ regionidcity, data=full)
summary(fips.impute.model)
```

```
##
## Call:
## lm(formula = fips ~ regionidcity, data = full)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.72  -11.54  -10.88   10.64   63.28
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  6.049e+03  1.443e-02 419197.17  <2e-16 ***
## regionidcity -1.985e-05  2.341e-07   -84.76  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.3 on 2922493 degrees of freedom
## (62847 observations deleted due to missingness)
## Multiple R-squared:  0.002452,    Adjusted R-squared:  0.002452
## F-statistic:  7185 on 1 and 2922493 DF,  p-value: < 2.2e-16
```

```
fips.impute.model <- lm(latitude ~ regionidcity, data=full)
summary(fips.impute.model)
```

```
##
## Call:
## lm(formula = latitude ~ regionidcity, data = full)
```

- P-Values looked good
- R² usually crap:

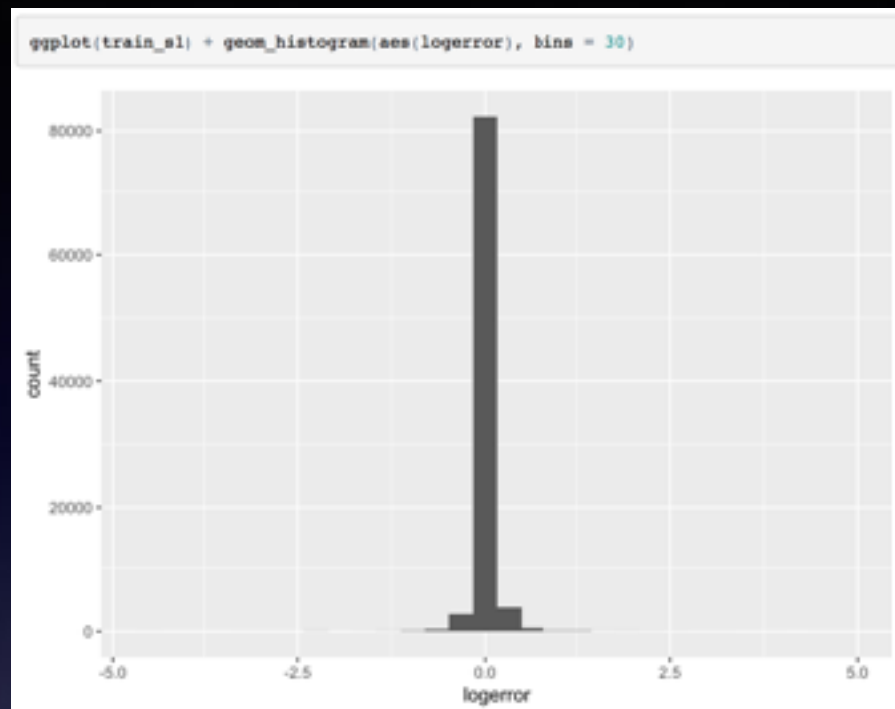
R-squared: 9.404e-07

Looks Correlated - LM() Imputations?

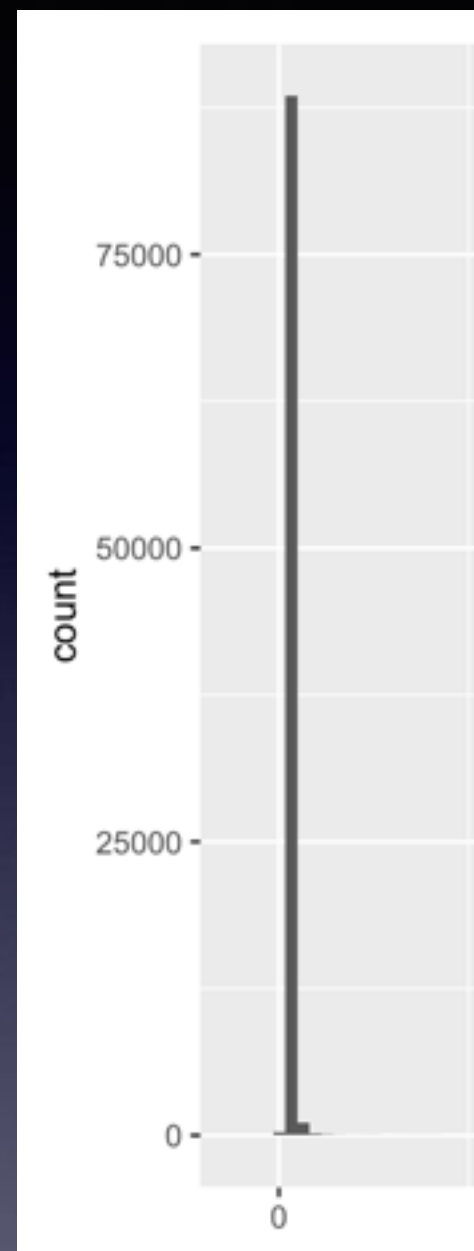
- Tax amount ~ taxvaluedollarcent
- R² about 90%

```
831
832 #An imputation for taxamount from taxvaluedollarcent (may include landvaluedollarcent later)
833 lm1 = lm(formula = taxamount ~ taxvaluedollarcent, data = full)
834
835 ## test results for above model:
836 # Call:
837 # lm(formula = taxamount ~ taxvaluedollarcent, data = full)
838 # Residuals:
839 #   Min       1Q   Median       3Q      Max
840 # -862470   -349    -134    103 1871995
841 # Coefficients:
842 #   Estimate Std. Error t value Pr(>|t|)
843 # (Intercept)    2.411e+02  1.957e+00   123.2  <2e-16 ***
844 # taxvaluedollarcent 1.229e-02  2.368e-06  5192.8  <2e-16 ***
845 # ---
846 # Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
847 # Residual standard error: 2883 on 2930063 degrees of freedom
848 # (55152 observations deleted due to missingness)
849 # Multiple R-squared:  0.902, Adjusted R-squared:  0.902
850 # F-statistic: 2.697e+07 on 1 and 2930063 DF,  p-value: < 2.2e-16
```

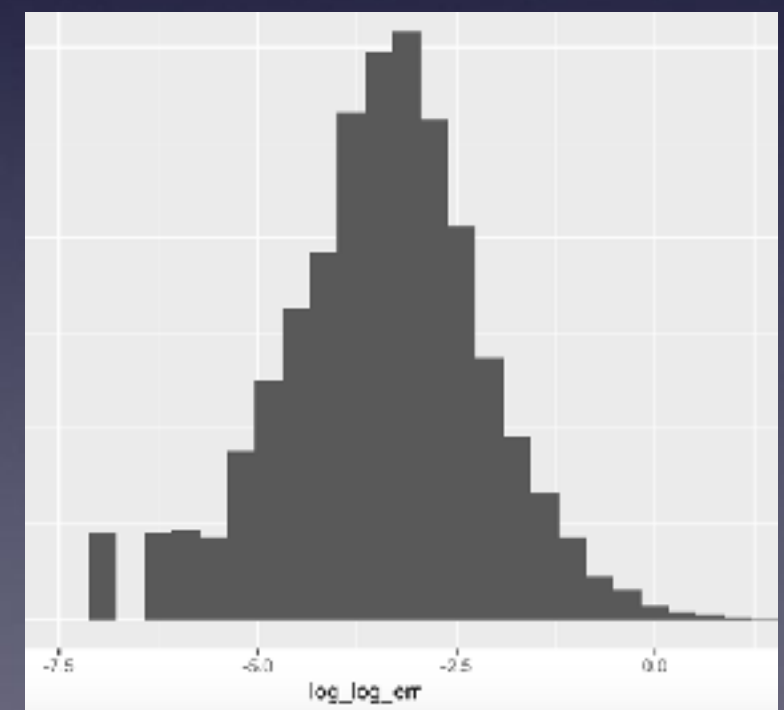
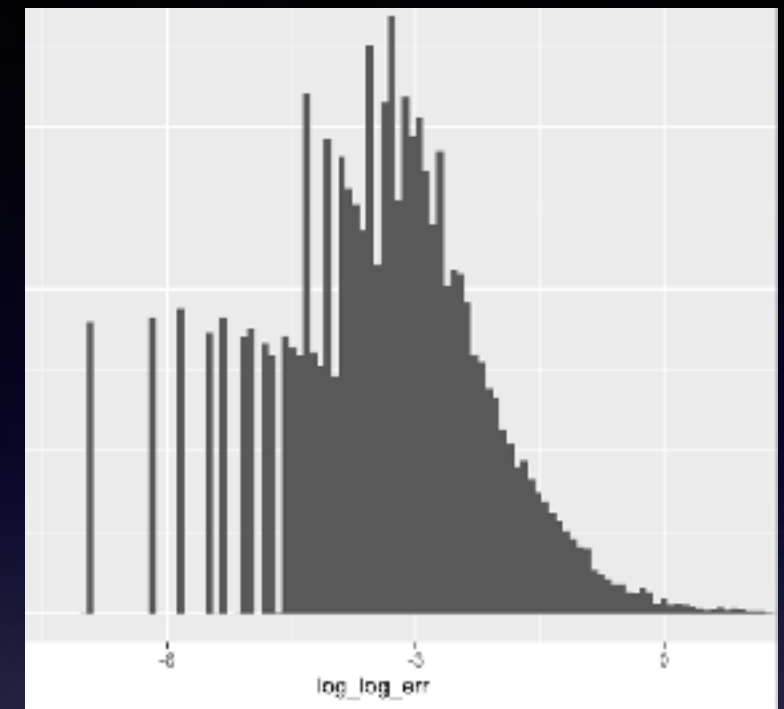
Looking At The “Log” in LogError



Logerror



exp(logerror)

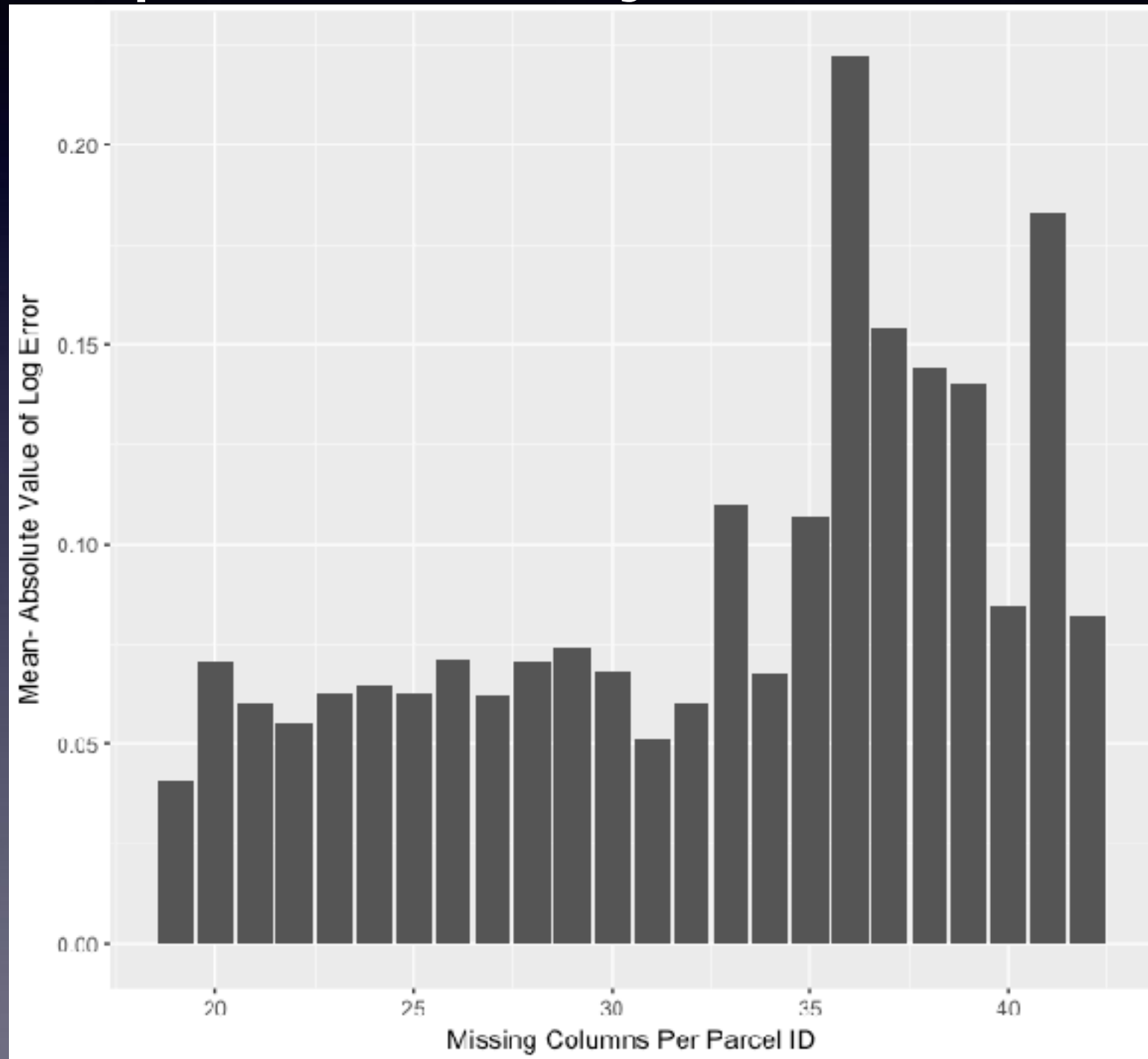


log(logerror)

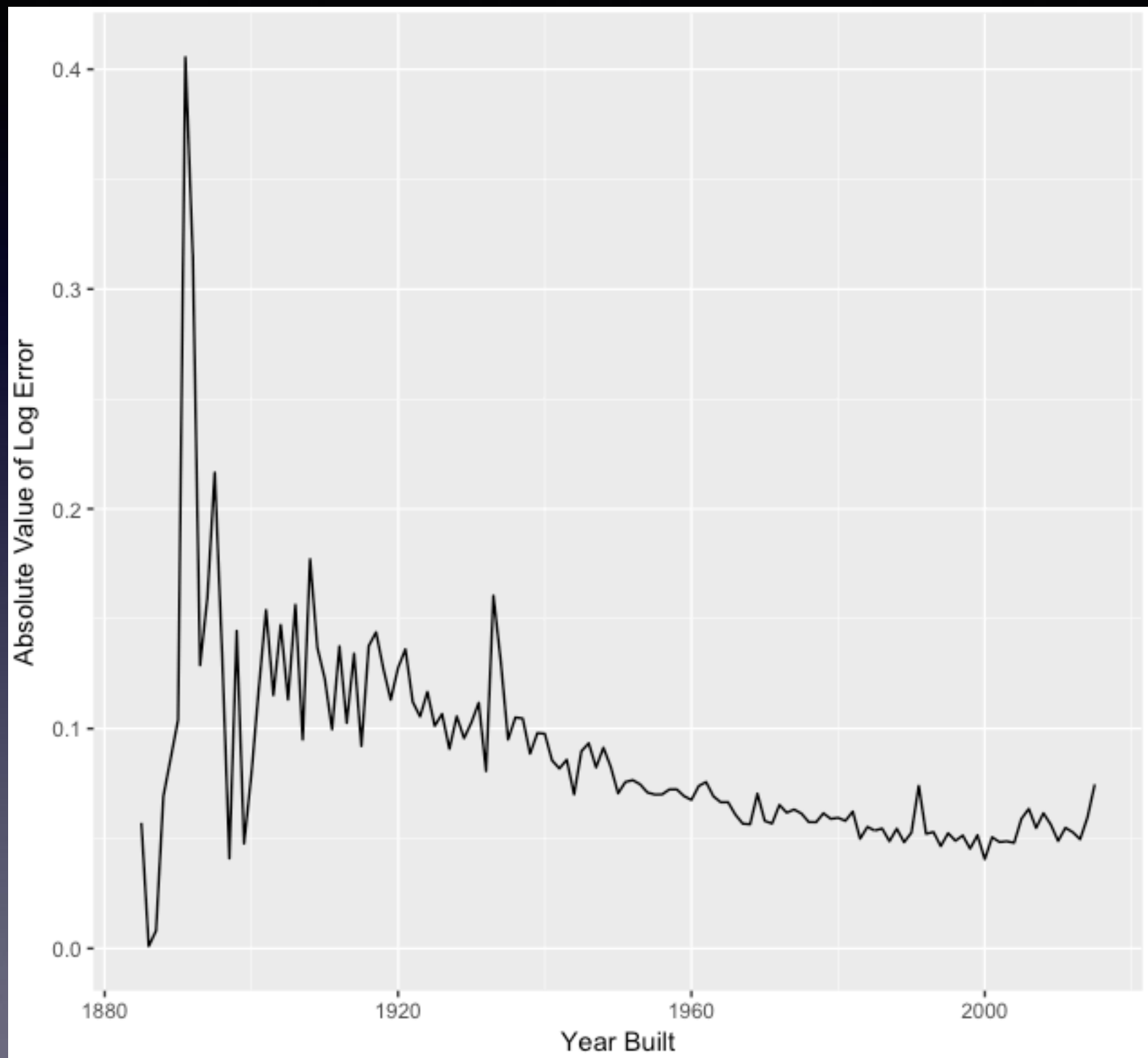
The real relationship:

$$\text{logerror} = \log(\text{Zestimate}) - \log(\text{SalePrice})$$

Missingness as an explanatory variable?



Year Built vs Absolute Value of Log Error



Multivariate Outliers/ Inaccurate Data

- Even when values were NOT missing, many of them were baffling- to the point of seeming wildly inaccurate (or being extreme outliers.)
- IF these values can be eliminated from the training process, maybe predictive accuracy will increase?
- Examples?

Multivariate Outliers/ Inaccurate Data

	parcelid [⬇]	bathroomcnt [⬆]	bedroomcnt [⬆]	logerror [⬆]
1	12325767	20	0	NA
2	12875313	20	0	NA
3	10875978	20	0	NA
4	11843438	20	0	NA
5	12718575	20	0	NA

20 bathrooms
0 bedrooms

YES!
My Dream
Home!!

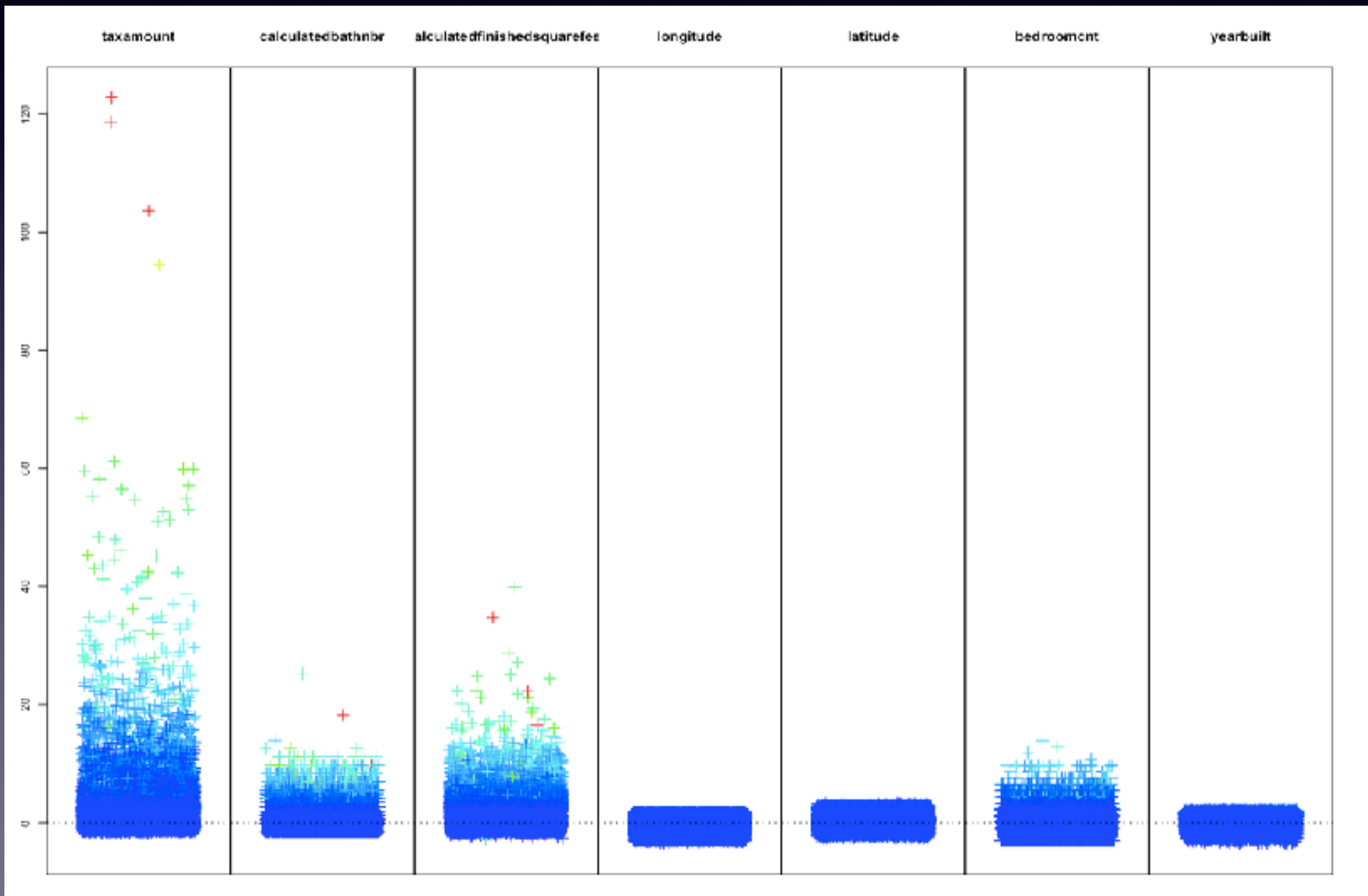

	parcelid [⬇]	bathroomcnt [⬆]	bedroomcnt [⬆]	roomcnt [⬆]	fireplacecnt [⬆]	poolcnt [⬆]	taxamount [⬆]	landtaxvaluedollarcnt [⬆]	logerror [⬆]
1	14167696	0	0	0	1	1	97.7	11151	NA
2	14167459	0	0	0	1	1	97.7	6700	NA

Who needs rooms when you have a
fireplace AND a pool??

Uni Plot

(Univariate Presentation of Multivariate Outliers)

Tax Bath Sq-Ft Long Lat Bed Year Built



The Models

- Linear
- Linear - with transformation (Yeo-Johnson)
- Regularization (Ridge / Lasso / Elastic Net)
- RandomForest
- Gradient Boosting
- XGBoost

Will's "Stupid Regression"

```
#stupid multi regression  
lr1 <- lm(logerror ~ calculatedfinishedsquarefeet + yearbuilt, data=full)
```

MAE



submit_2.csv

8 days ago by WillMarkowitz

linear

0.0648835

submit_1.csv

8 days ago by WillMarkowitz

[add submission details](#)

Error ⓘ

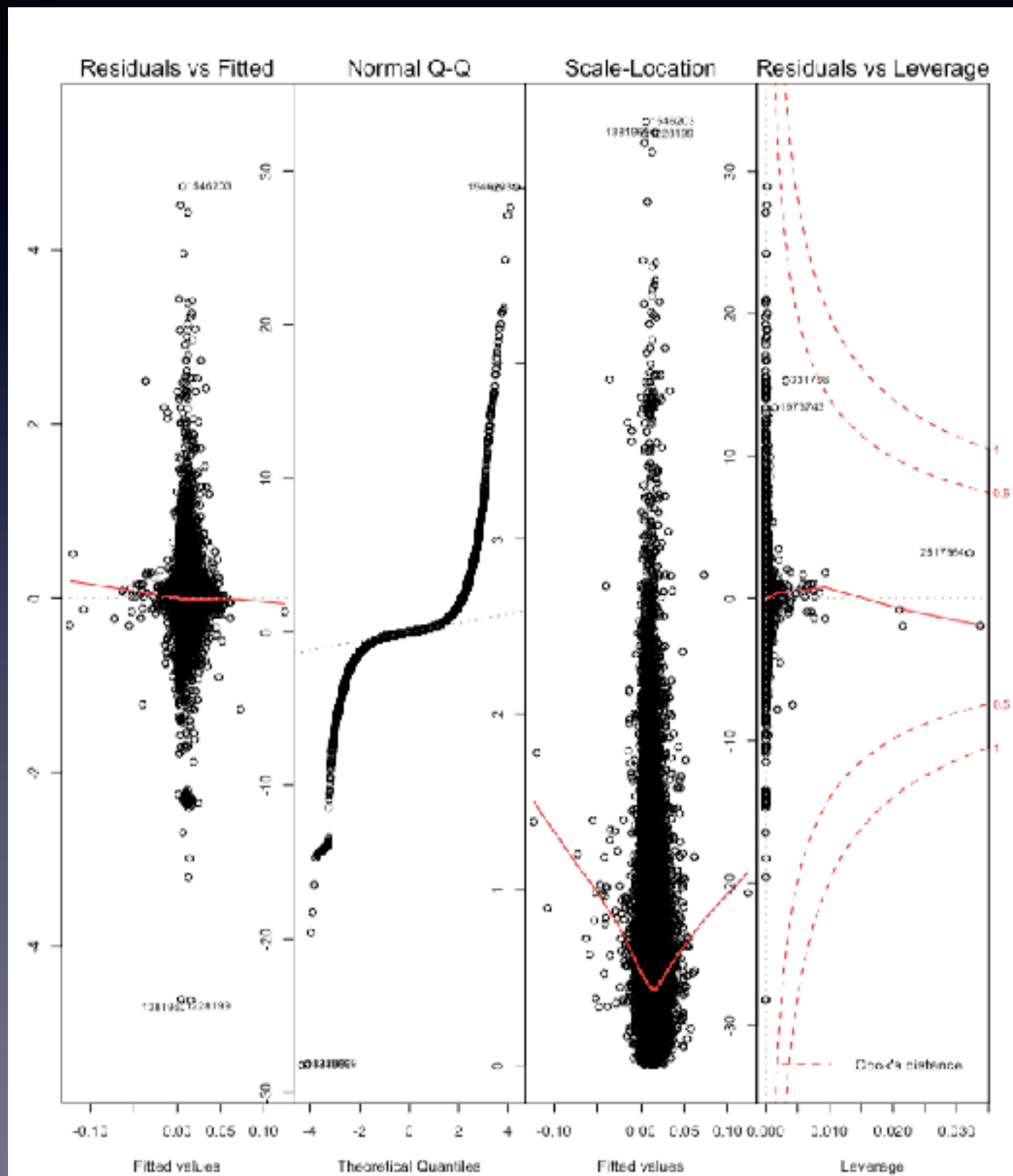
submit_1.csv

8 days ago by WillMarkowitz

[add submission details](#)

Error ⓘ

Linear Models



- Linearity violated
- Constant variance violated
- Normality violated

What about transformation?

We have negative values, and Box-Cox cannot take those

.....

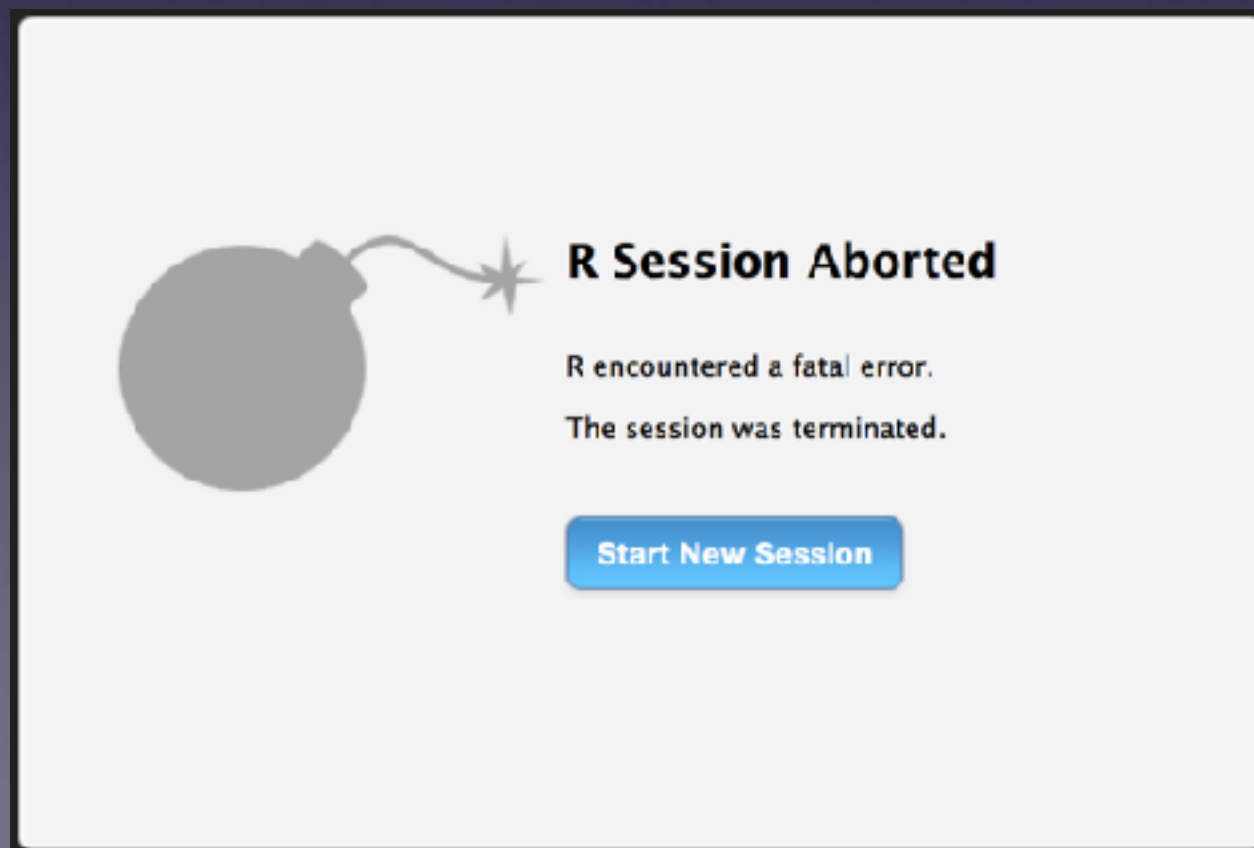
Yeo-Johnson Transformation

```
model_yeo = train(logerror ~ yearbuilt + calculatedfinishedsquarefeet,  
                  method = 'lm', data = sub_train,  
                  preProcess = "YeoJohnson", trControl = myControl)  
# Linear model with Yeo-Johnson  
yeo1 = predict(model_yeo, newdata = sub_test)  
# Predicting on test data  
sum(abs(yeo1-sub_test$logerror))/nrow(sub_test)  
# 0.06674093 MAE
```

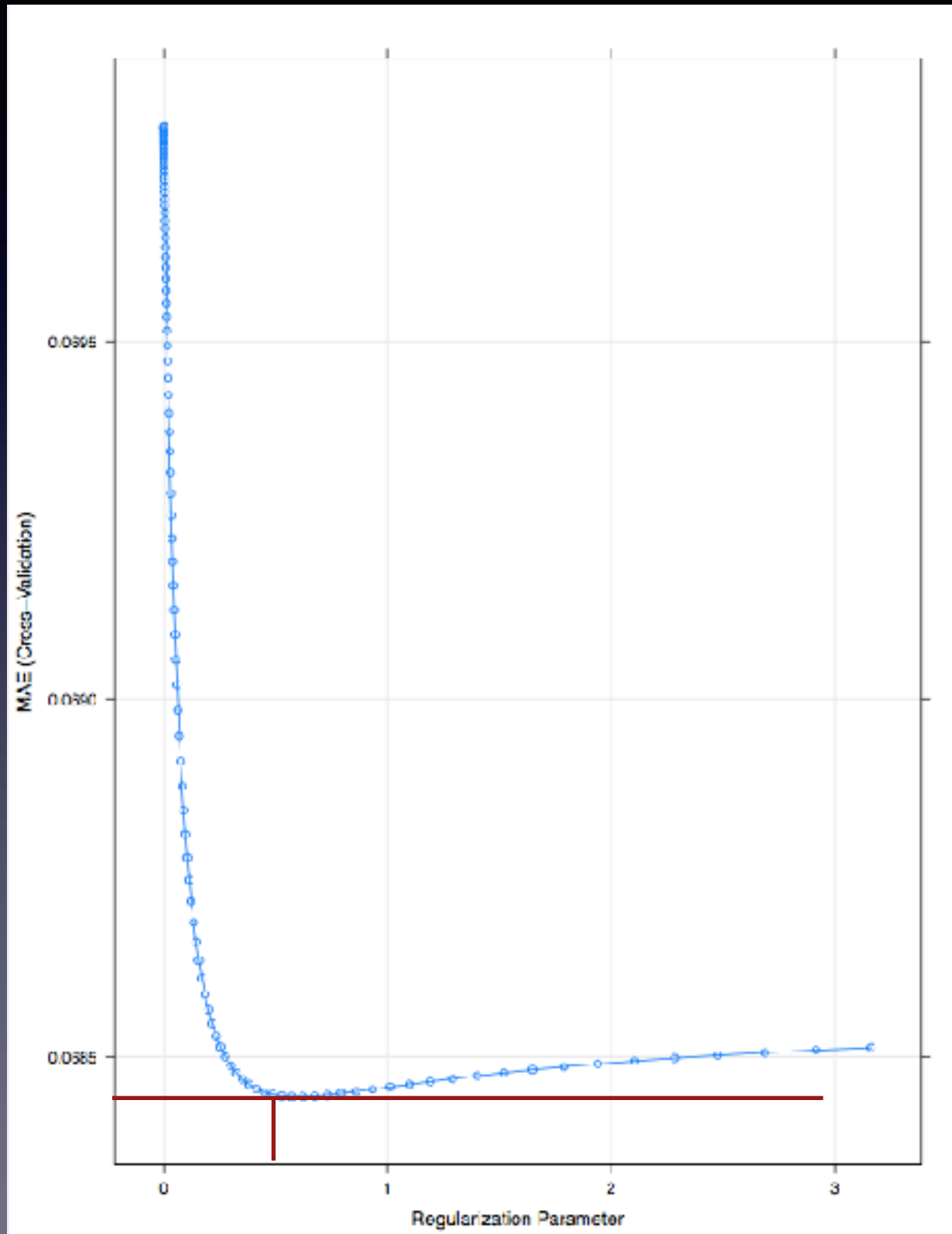
- Works very similarly to Box-Cox but CAN handle negative values
- Slight improvement, but still not great (did not improve the “stupid regression”)

Regularization Regressions

- Performed Ridge, Lasso, and Elastic Net Regressions (mix of L1 and L2 penalties)
- Ridge performed the best of the three
- Elastic Net, (after multiple “R-bombs”), yielded an optimal α of ~ 0.01 through cross-validation, which is essentially Ridge.



Ridge Regression



Optimal $\lambda \sim 0.446$

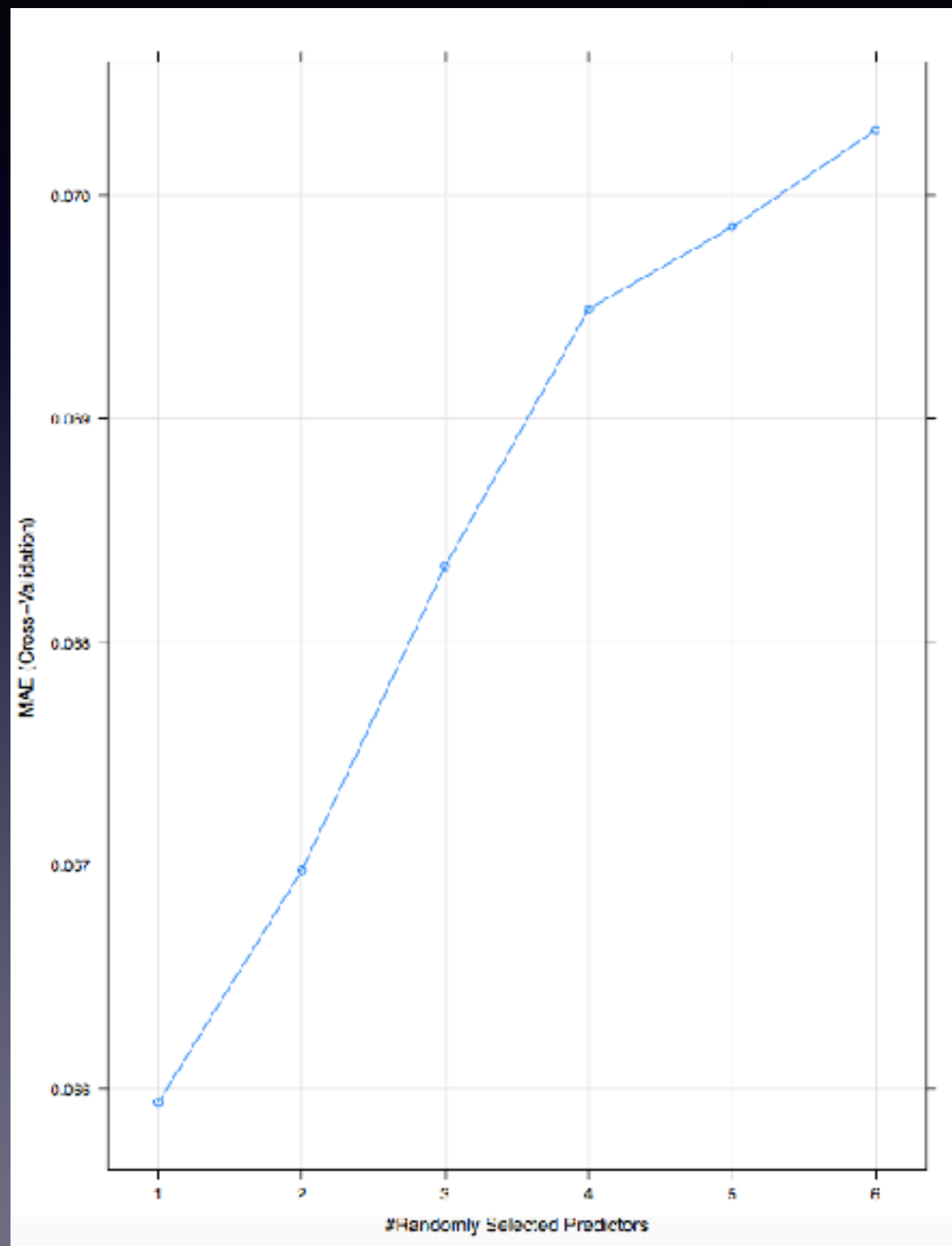
n-folds = 10

One of the better
Kaggle Scores!!

(Did not top Will's
"stupid multi regression")

MAE = 0.0649250

Random Forest

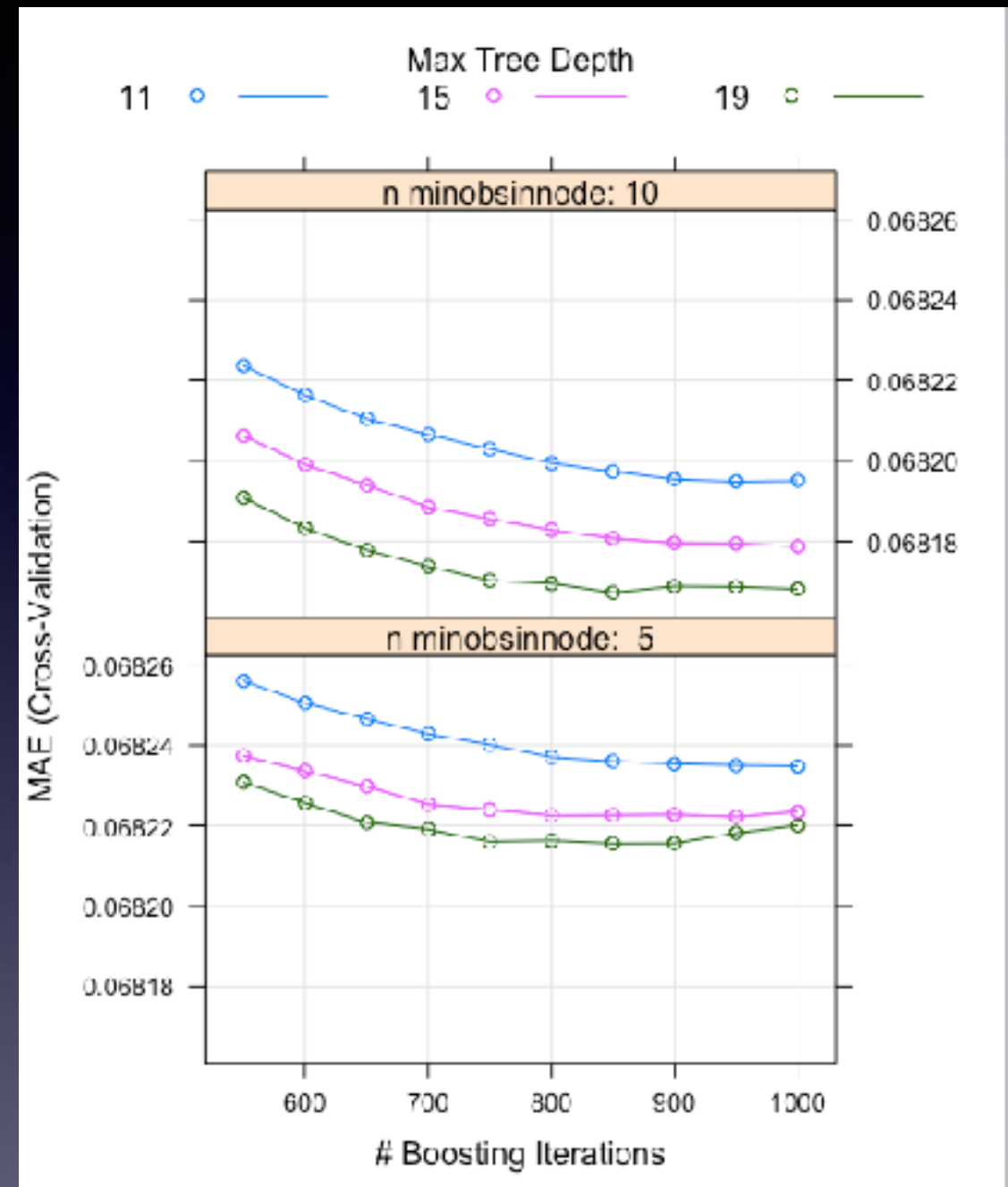


GBM Caret Tune

```
gridSearch <- trainControl(method = "cv",
  number = 3,
  summaryFunction = maeSummary,
  verboseIter = TRUE)

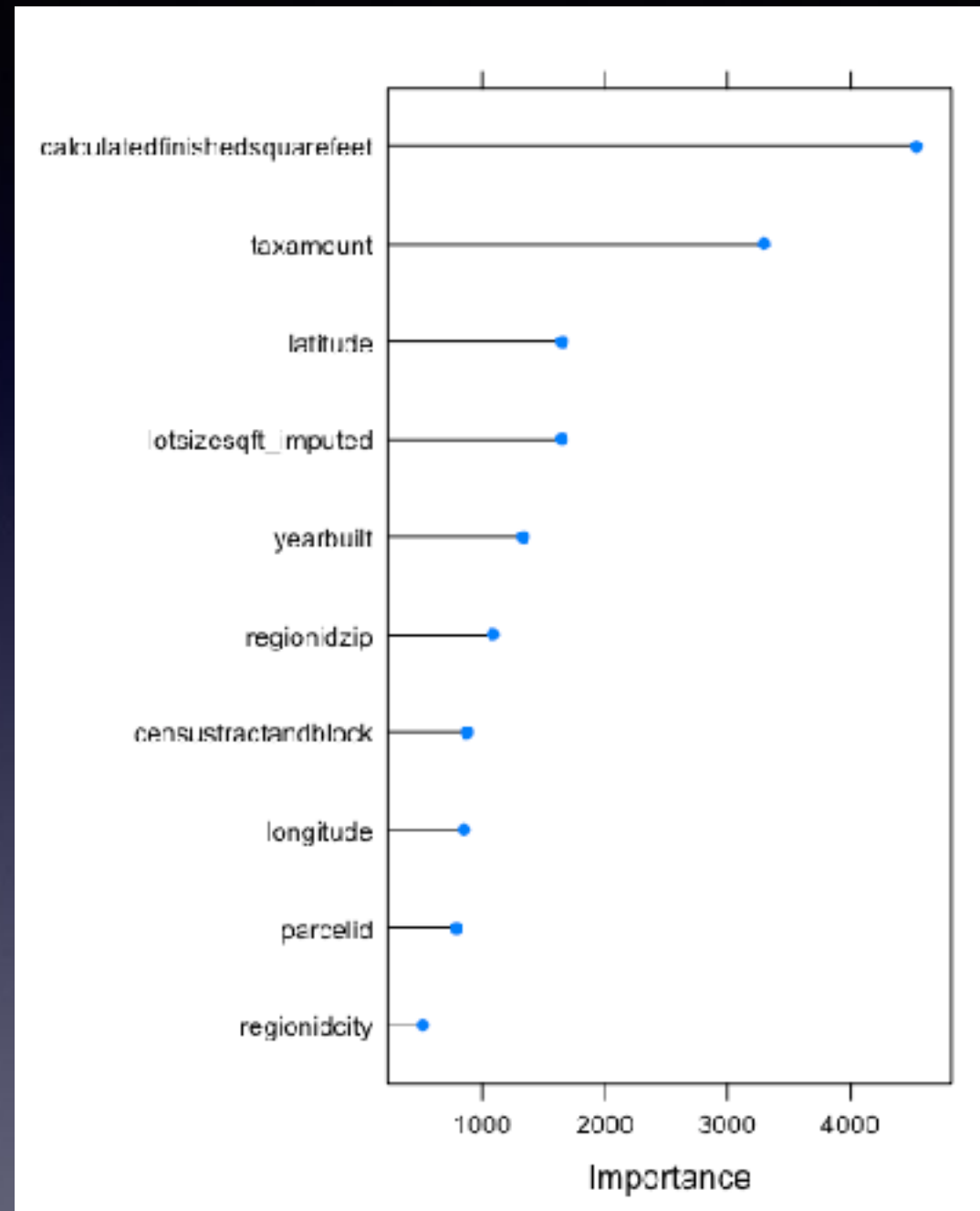
gbmGrid <- expand.grid(interaction.depth = c(11,15,19),
  n.trees = (11:20)*50,
  shrinkage = c(.001),
  n.minobsinnode = c(5,10))

set.seed(0)
gbmFit4 <- train(logerror ~ . - propertycountylandusecode,
  data = full.train1,
  method = "gbm",
  preProcess = c("center", "scale"),
  metric = "MAE",
  maximize = FALSE,
  tuneGrid = gbmGrid,
  trControl = gridSearch,
  verbose = FALSE)
```



```
> gbmFit4$bestTune
  n.trees interaction.depth shrinkage n.minobsinnode
57      850              19    0.001             10
```

Gradient Boosted Model



submit_7.csv

3 days ago by WillMarkowitz

GBM

0.0649035

XGBoost Caret Tune

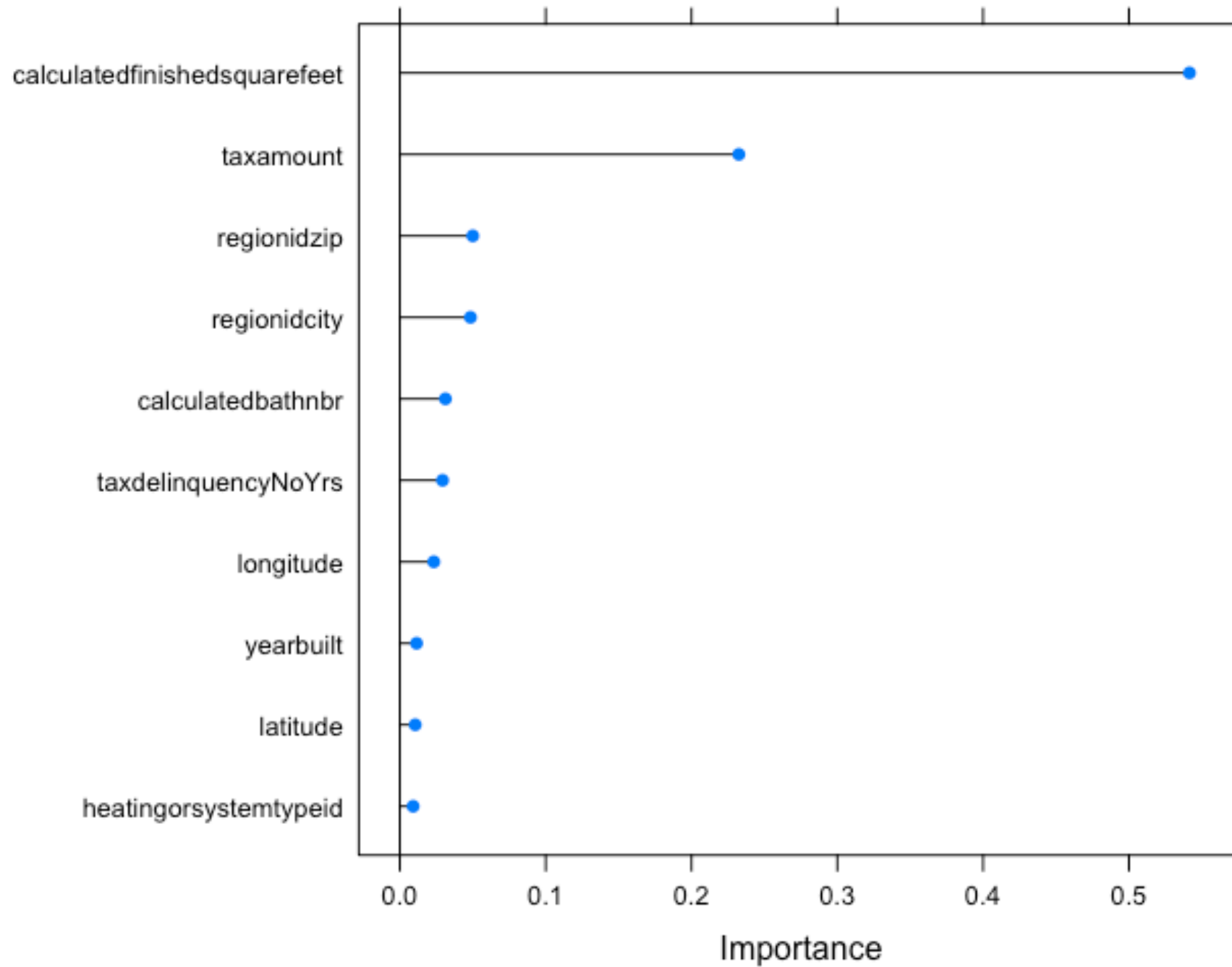
```
cv.ctrl <- trainControl(method = "cv", number = 4,
                        classProbs = TRUE,
                        summaryFunction = maeSummary,
                        verboseIter = TRUE,
                        allowParallel=T)

ntrees = 100
xgbGrid <- expand.grid( #Owen Zhang parameter tuning slides
  eta = (2:10)/ntrees, #learning rate
  max_depth = c(4,6,8,10),
  nrounds = ntrees,
  gamma = 0, #default=0
  colsample_bytree = c(.4, .6, .8, 1), #default=1
  min_child_weight = c(1,3,5), #default=1
  subsample = c(.5, .75, 1)
)

set.seed(45)
xgb_tune <- train(logerror ~ . -propertycountylandusecode-numberofstories,
                 data=small.full.train1,
                 method="xgbTree",
                 trControl=cv.ctrl,
                 tuneGrid=xgbGrid,
                 verbose=T,
                 verboseIter = T,
                 #preProcess = c("center", "scale"),
                 metric="MAE",
                 maximize = FALSE
                 #objective = 'reg:linear',
)
```

```
> fwrite(xgb_tune$results, 'xgb2.csv')
> xgb_tune$bestTune
      nrounds max_depth  eta gamma colsample_bytree min_child_weight subsample
112       100        10 0.02    0           0.4              3         0.5
> |
```

XGBoost Importance



[submit_xgb_2.csv](#)

12 hours ago by [WillMarkowitz](#)

[add submission details](#)

0.0649232

Best Kaggle Score

```
#stupid multi regression  
lr1 <- lm(logerror ~ calculatedfinishedsquarefeet + yearbuilt, data=full)
```

+ GBM

=

1297

▲ 82

WillMarkowitz



0.0648091

8

~10s

Your Best Entry ↑

You advanced 95 places on the leaderboard!

Your submission scored 0.0648091, which is an improvement of your previous score of 0.0648835. Great job!



Tweet this!

Future Work

- Explore why Linear Model did so well
- Run XGBoost on columns with less imputation
- Model Ensembling
- Techniques that require more computing power (ex: KNN, LOESS)