Misc results, datathon 2022

Andrew Zimolzak

2022-04-20

Misc. results (too detailed)

The analysis is pretty sensitive to the phrasing of the descriptors. Limiting to "unpleasant" results in far fewer notes than "pleasant." Also, making that change results in more negativity for: *men, non-Latinos, and black patients,* unlike the prior word list when it was just the substring "pleasant."

In these tables, an upward trend in ratio means positive correlation of underprivileged group status and negative descriptors. In other words, ratios are $r = u/p$, where u is the count from a group hypothesized to be underprivileged, and p is the group hypothesized to be privileged.

Sex vs. negative descriptors

```
table(joined$Sex, joined$negativity) -> x
round((x[1,] / x[2,]), 3) -> f.m.ratio  # adhoc
rbind(x, f.m.ratio) %>% kable()
```

	0	1	2	3	4	5	6	8	9	10	12	13	15	18	20	24
Female	8381.000	269.000	84.000	11.000	9.000	6	3.000	0	1	3	2	0	2	1	1	1
Male	4492.000	145.000	49.000	14.000	16.000	2	8.000	3	0	1	0	1	0	0	0	1
f.m.ratio	1.866	1.855	1.714	0.786	0.562	3	0.375	0	Inf	3	Inf	0	Inf	Inf	Inf	1

```
women <- x[1,]  # adhoc
totals <- x[1,] + x[2,]  # adhoc
prop.trend.test(women, totals)
```

```
##
##  Chi-squared Test for Trend in Proportions
##
## data:  women out of totals ,
##  using scores: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
## X-squared = 4.5055, df = 1, p-value = 0.03379
```

Ethnicity vs. negative descriptors

```
table(joined$Ethnic.Group, joined$negativity)  -> x
round((x[1,] / x[2,]), 3) -> latino.non.ratio  # adhoc
rbind(x, latino.non.ratio) %>% kable()
```

	0	1	2	3	4	5	6	8	9	10	12	13	15	18	20	24
Hispanic or Latino	3000.000	119.000	43.000	4.00	8.000	0	1.0	2	0	0	0	0	0	0	0	0
Not Hispanic or Latino	9873.000	295.000	90.000	21.00	17.000	8	10.0	1	1	4	2	1	2	1	1	2
latino.non.ratio	0.304	0.403	0.478	0.19	0.471	0	0.1	2	0	0	0	0	0	0	0	0

```
latino <- x[1,]  # adhoc
totals <- x[1,] + x[2,]  # adhoc
prop.trend.test(latino, totals)
```

```
##
##  Chi-squared Test for Trend in Proportions
##
## data:  latino out of totals ,
##  using scores: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
## X-squared = 0.0023657, df = 1, p-value = 0.9612
```

Race vs. negative descriptors

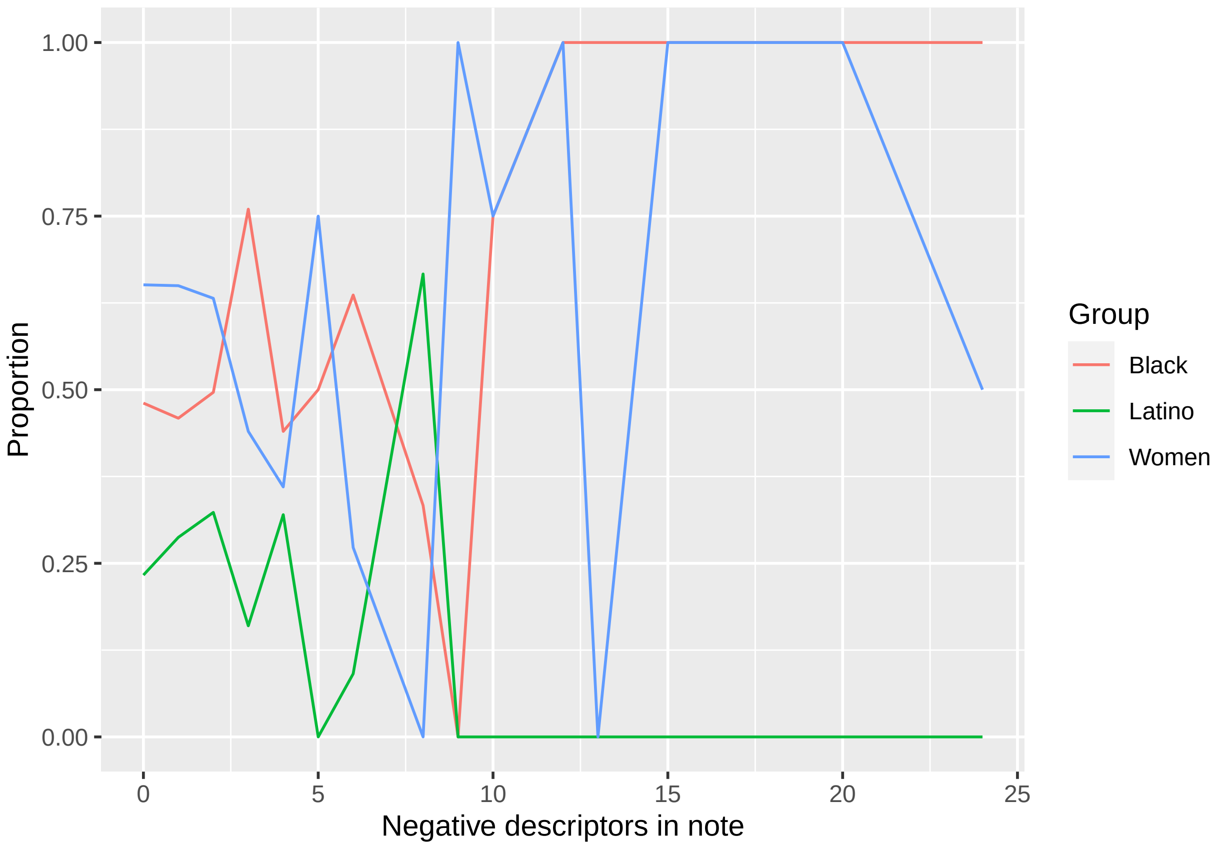
```
table(joined$Race, joined$negativity)  -> x
round((x['Black or African American',] / x['White or Caucasian',]), 2) -> b.w.ratio
rbind(x, b.w.ratio) %>% kable()
```

	0	1	2	3	4	5	6	8	9	10	12	13	15	18	20	24
American Indian or Alaska Native	1.00	0.00	0	0.00	0.00	0	0.00	0.0	0	0	0	0	0	0	0	0
Asian	314.00	1.00	1	0.00	0.00	0	0.00	0.0	0	0	0	0	0	0	0	0
Black or African American	6188.00	190.00	66	19.00	11.00	4	7.00	1.0	0	3	2	1	2	1	1	2
Native Hawaiian or Other Pacific Islander	15.00	0.00	0	0.00	0.00	0	0.00	0.0	0	0	0	0	0	0	0	0
Unable to Determine	17.00	0.00	0	0.00	0.00	0	0.00	0.0	0	0	0	0	0	0	0	0
White or Caucasian	6338.00	223.00	66	6.00	14.00	4	4.00	2.0	1	1	0	0	0	0	0	0
b.w.ratio	0.98	0.85	1	3.17	0.79	1	1.75	0.5	0	3	Inf	Inf	Inf	Inf	Inf	Inf

```
black <- x['Black or African American',]
totals <- colSums(x)
prop.trend.test(black, totals)
```

```
##
##  Chi-squared Test for Trend in Proportions
##
## data:  black out of totals ,
##  using scores: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
## X-squared = 6.9277, df = 1, p-value = 0.008487
```

Single proportion line plot



Combined race/ethnicity vs. negative descriptors

Without a "cap" on negativity

	0	1	2	3	4	5	6	8	9	10	12	13	15	18	20	24
American Indian or Alaska Native	1.000	0.000	0.000	0.00	0.00	0.0	0.000	0.000	0	0.00	0	0	0	0	0	0
Asian	314.000	1.000	1.000	0.00	0.00	0.0	0.000	0.000	0	0.00	0	0	0	0	0	0
Black or African American	6188.000	190.000	66.000	19.00	11.00	4.0	7.000	1.000	0	3.00	2	1	2	1	1	2
Latino	3000.000	119.000	43.000	4.00	8.00	0.0	1.000	2.000	0	0.00	0	0	0	0	0	0
White or Caucasian	3370.000	104.000	23.000	2.00	6.00	4.0	3.000	0.000	1	1.00	0	0	0	0	0	0
Total	12873.000	414.000	133.000	25.00	25.00	8.0	11.000	3.000	1	4.00	2	1	2	1	1	2
p_asian	0.024	0.002	0.008	0.00	0.00	0.0	0.000	0.000	0	0.00	0	0	0	0	0	0
p_black	0.481	0.459	0.496	0.76	0.44	0.5	0.636	0.333	0	0.75	1	1	1	1	1	1
p_latino	0.233	0.287	0.323	0.16	0.32	0.0	0.091	0.667	0	0.00	0	0	0	0	0	0
p_white	0.262	0.251	0.173	0.08	0.24	0.5	0.273	0.000	1	0.25	0	0	0	0	0	0

```r
prop.trend.test(x['White or Caucasian',], Total)
```

```
##
##  Chi-squared Test for Trend in Proportions
##
## data:  x["White or Caucasian", ] out of Total ,
##  using scores: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
## X-squared = 4.8102, df = 1, p-value = 0.02829
```

```r
prop.trend.test(x['Latino',], Total)
```

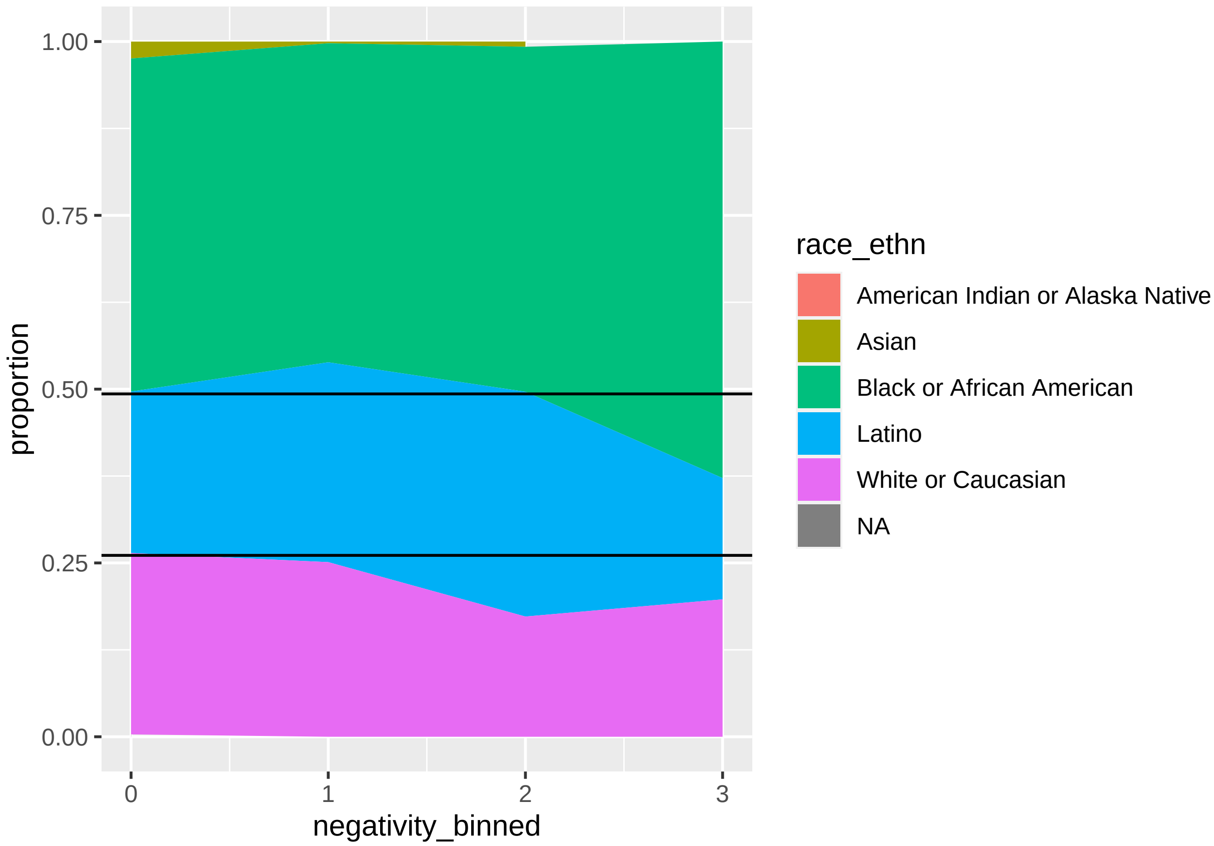
```
##
##  Chi-squared Test for Trend in Proportions
##
## data:  x["Latino", ] out of Total ,
##  using scores: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
## X-squared = 0.0023657, df = 1, p-value = 0.9612
```

```r
prop.trend.test(x['Black or African American',], Total)
```

```
##
##  Chi-squared Test for Trend in Proportions
##
## data:  x["Black or African American", ] out of Total ,
##  using scores: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
## X-squared = 6.9277, df = 1, p-value = 0.008487
```

With a "cap" on negativity

	0	1	2	3
American Indian or Alaska Native	1.000	0.000	0.000	0.000
Asian	314.000	1.000	1.000	0.000
Black or African American	6188.000	190.000	66.000	54.000
Latino	3000.000	119.000	43.000	15.000
White or Caucasian	3370.000	104.000	23.000	17.000
Total	12873.000	414.000	133.000	86.000
p_asian	0.024	0.002	0.008	0.000
p_black	0.481	0.459	0.496	0.628
p_latino	0.233	0.287	0.323	0.174
p_white	0.262	0.251	0.173	0.198




```r
prop.trend.test(x['White or Caucasian',], Total)
```

```
##
##  Chi-squared Test for Trend in Proportions
##
## data:  x["White or Caucasian", ] out of Total ,
##  using scores: 1 2 3 4
## X-squared = 5.818, df = 1, p-value = 0.01586
```

```r
prop.trend.test(x['Latino',], Total)
```

```
##
##  Chi-squared Test for Trend in Proportions
##
## data:  x["Latino", ] out of Total ,
##  using scores: 1 2 3 4
## X-squared = 2.8946, df = 1, p-value = 0.08887
```

```r
prop.trend.test(x['Black or African American',], Total)
```

```
##
##  Chi-squared Test for Trend in Proportions
##
## data:  x["Black or African American", ] out of Total ,
##  using scores: 1 2 3 4
## X-squared = 2.5899, df = 1, p-value = 0.1075
```

Logistic

Big model

```r
my_model <- glm(negativity_any ~ ., data = logit_me, family = "binomial")
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(my_model)
```

```
##
## Call:
## glm(formula = negativity_any ~ ., family = "binomial", data = logit_me)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9424  -0.3732  -0.2878  -0.2296   3.4558
##
## Coefficients:
##                                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)                         -7.928e+10  4.558e+11  -0.174 0.861904
## SexMale                              8.636e-01  1.712e-01   5.045 4.53e-07 ***
## race_ethnAsian                       7.928e+10  4.558e+11   0.174 0.861904
## race_ethnBlack or African American   7.928e+10  4.558e+11   0.174 0.861904
## race_ethnLatino                      7.928e+10  4.558e+11   0.174 0.861904
## race_ethnWhite or Caucasian          7.928e+10  4.558e+11   0.174 0.861904
## Employment.StatusFull Time           8.895e-01  4.378e-01   2.032 0.042171 *
## Employment.StatusNot Employed        2.468e-01  4.516e-01   0.547 0.584710
## Employment.StatusPart Time           1.476e+00  6.251e-01   2.361 0.018209 *
## Employment.StatusRetired             7.072e-01  4.293e-01   1.647 0.099466 .
## Employment.StatusStudent - Full time 3.270e-01  7.286e-01   0.449 0.653574
## Employment.StatusUnknown            -2.110e+01  2.066e+05   0.000 0.999919
## interpreterY                         2.486e+01  8.210e+04   0.000 0.999758
## LanguageFarsi, Persian               7.928e+10  4.558e+11   0.174 0.861904
## LanguageLaotian                     -4.864e+01  3.462e+05   0.000 0.999888
## LanguageSpanish                     -2.452e+01  8.210e+04   0.000 0.999762
## LanguageUnknown                     -2.315e+01  3.236e+05   0.000 0.999943
## LanguageVietnamese                  -4.744e+01  8.455e+04  -0.001 0.999552
## Marital.StatusLegally Separated     -4.263e+00  6.327e-01  -6.738 1.60e-11 ***
## Marital.StatusLife Partner          -2.735e+00  6.205e-01  -4.408 1.04e-05 ***
## Marital.StatusMarried               -2.940e+00  5.453e-01  -5.391 7.01e-08 ***
## Marital.StatusSingle                -2.587e+00  5.268e-01  -4.910 9.10e-07 ***
## Marital.StatusUnknown               -2.559e+01  2.066e+05   0.000 0.999901
## Marital.StatusWidow/Widower         -1.529e+00  5.011e-01  -3.050 0.002286 **
## mcaidTRUE                           -1.142e+00  1.154e+00  -0.990 0.322415
## mcareTRUE                           -7.069e-01  2.509e-01  -2.818 0.004837 **
## Financial.ClassMedicaid             -4.554e+01  3.431e+05   0.000 0.999894
## Financial.ClassMedicaid Mgd Care     6.436e-01  2.928e-01   2.198 0.027925 *
## Financial.ClassMedicare              1.202e+00  3.271e-01   3.674 0.000239 ***
## Financial.ClassMedicare Mgd Care     9.291e-01  3.576e-01   2.598 0.009367 **
## Financial.ClassSelf-Pay              1.653e+00  3.631e-01   4.553 5.29e-06 ***
## Age                                 -6.148e-03  1.000e-02  -0.615 0.538804
```

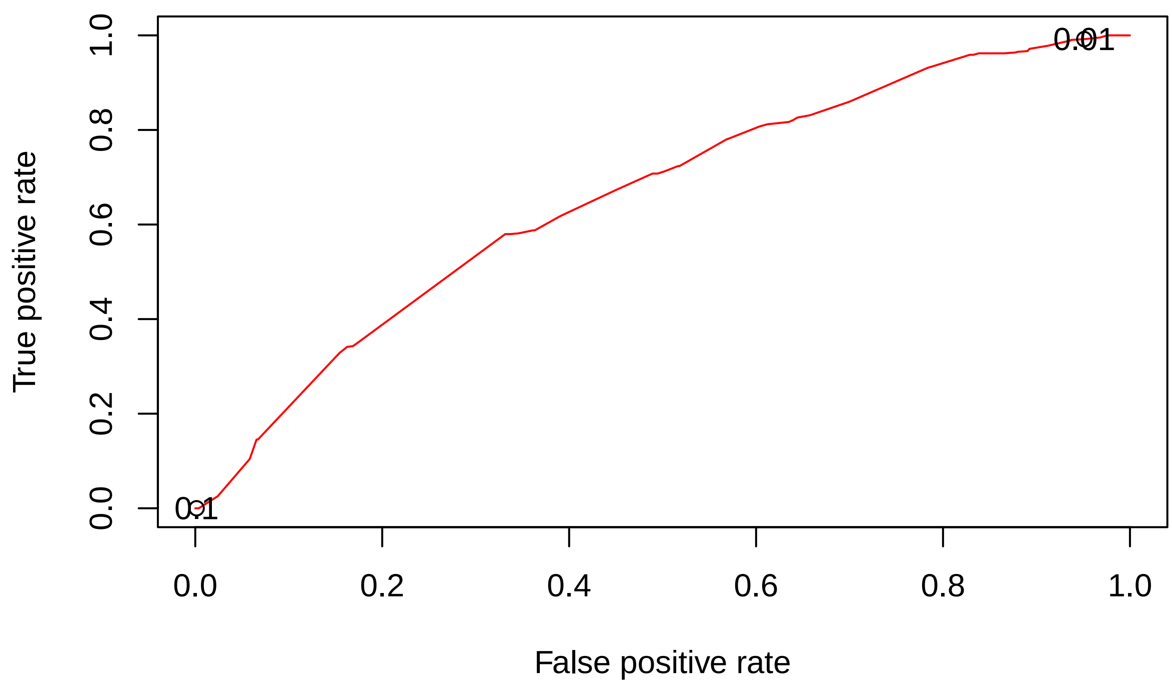
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5110.3  on 13505  degrees of freedom
## Residual deviance: 4926.5  on 13474  degrees of freedom
##   (44 observations deleted due to missingness)
## AIC: 4990.5
##
## Number of Fisher Scoring iterations: 25
```

ROC curve

```
rocr_pred = prediction(my_model$fitted.values, my_model$y)
rocr_perf <- performance(rocr_pred, measure = "tpr", x.measure = "fpr")
auc = performance(rocr_pred, measure = "auc")
auc@y.values
```

```
## [[1]]
## [1] 0.6562994
```

```
plot(rocr_perf, col=rainbow(10), print.cutoffs.at=c(0.01, 0.1))
```



Little model

```
little_model <- glm(negativity_any ~ Sex + race_ethn + Age, data = logit_me, family = "binomial")
summary(little_model)
```

```
##
```

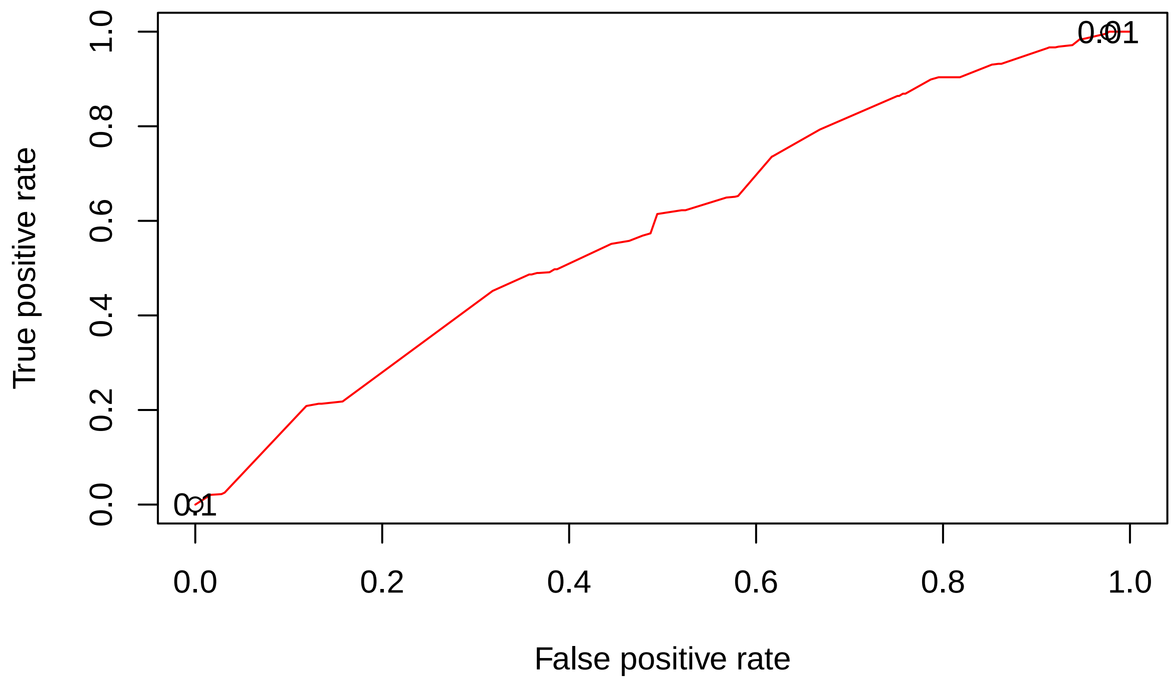
```
## Call:
## glm(formula = negativity_any ~ Sex + race_ethn + Age, family = "binomial",
##     data = logit_me)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.4268  -0.3438  -0.3024  -0.2753   3.1178
##
## Coefficients:
##                                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -13.083070 196.967874  -0.066    0.947
## SexMale                             0.079726   0.085267   0.935    0.350
## race_ethnAsian                      6.528620 196.968976   0.033    0.974
## race_ethnBlack or African American  8.917175 196.967700   0.045    0.964
## race_ethnLatino                     8.960270 196.967707   0.045    0.964
## race_ethnWhite or Caucasian         8.465686 196.967706   0.043    0.966
## Age                                 0.018912   0.003518   5.375 7.65e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5110.3  on 13505  degrees of freedom
## Residual deviance: 5053.2  on 13499  degrees of freedom
##   (44 observations deleted due to missingness)
## AIC: 5067.2
##
## Number of Fisher Scoring iterations: 10
```

ROC curve

```r
rocr_pred = prediction(little_model$fitted.values, little_model$y)
rocr_perf <- performance(rocr_pred, measure = "tpr", x.measure = "fpr")
auc = performance(rocr_pred, measure = "auc")
auc@y.values
```

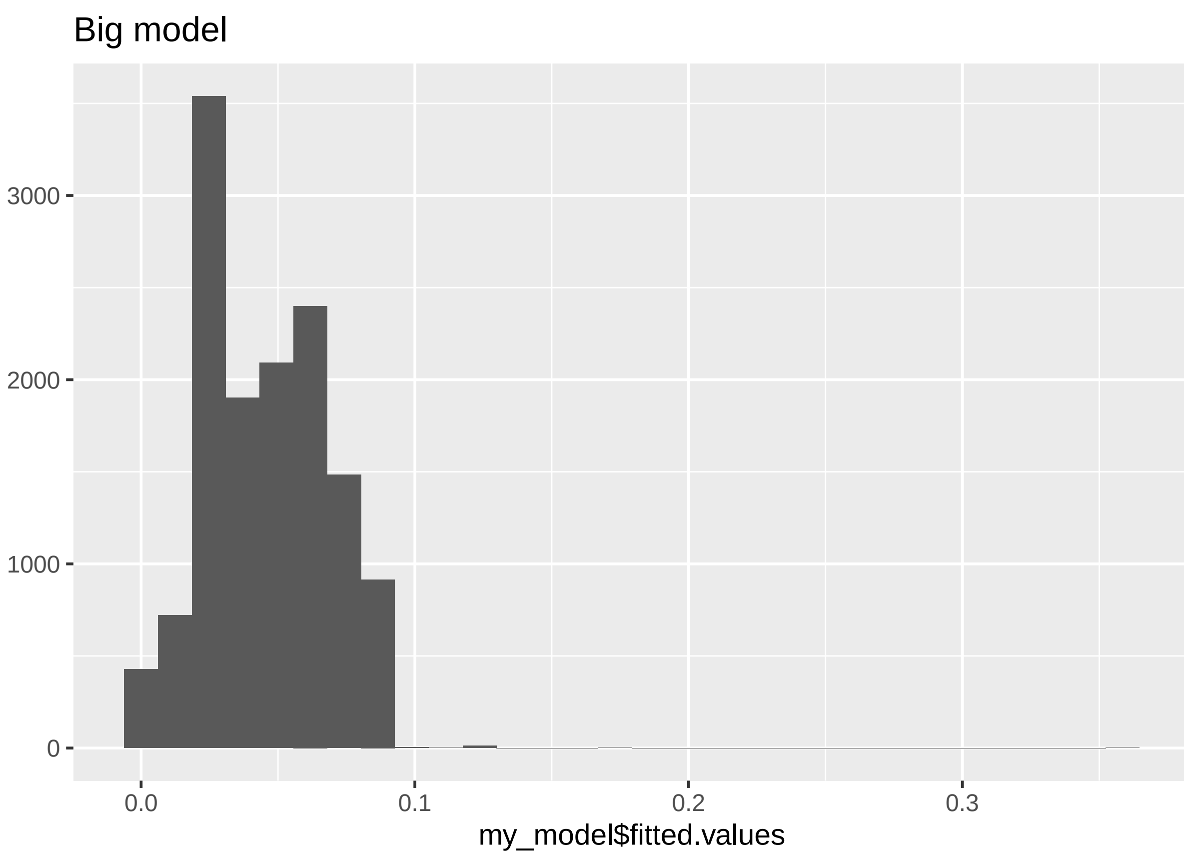
```
## [[1]]
## [1] 0.5858342
```

```r
plot(rocr_perf, col=rainbow(10), print.cutoffs.at=c(0.01, 0.1))
```

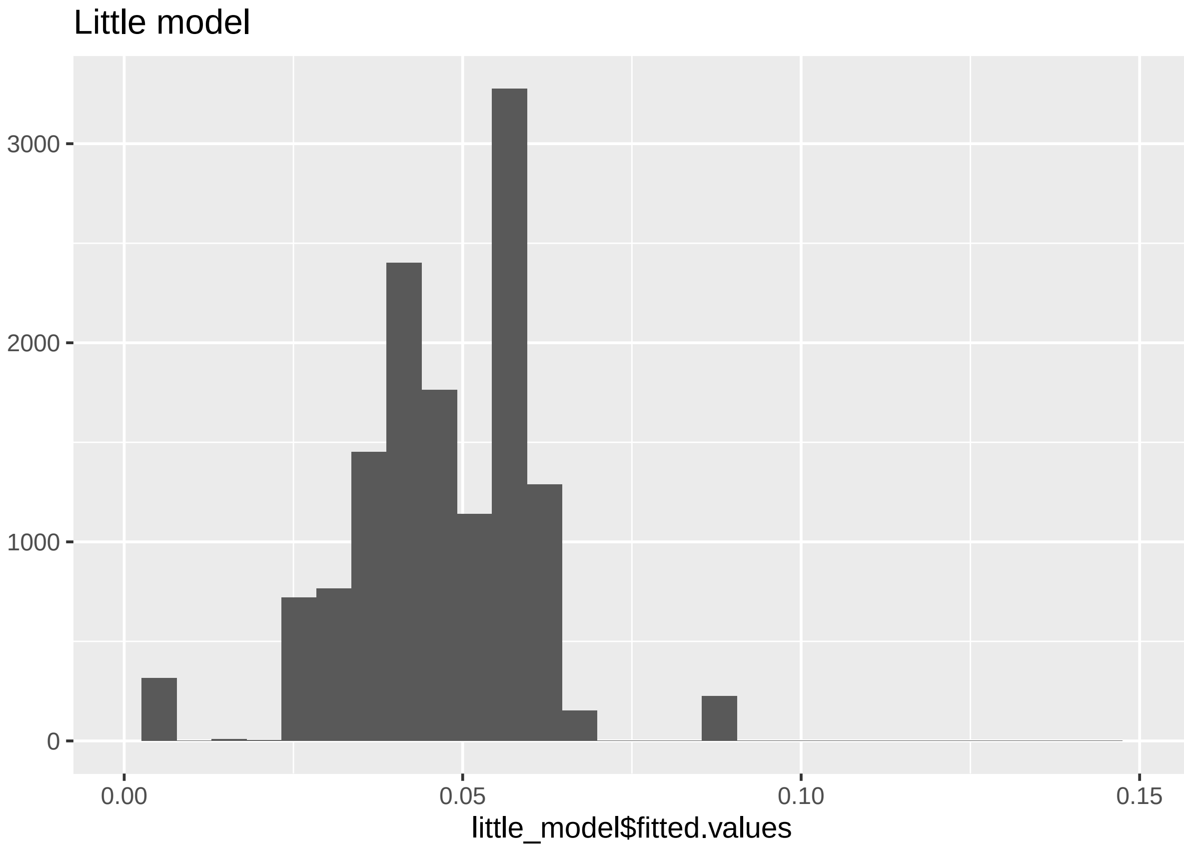


Predicted probabilities, big vs. little

```
qplot(my_model$fitted.values) + labs(title = 'Big model')
```



```r
qplot(little_model$fitted.values) + labs(title = 'Little model') + xlim(0, 0.15)
```



I get the feeling it hates unbalanced classes.

Try column plot of coefficients

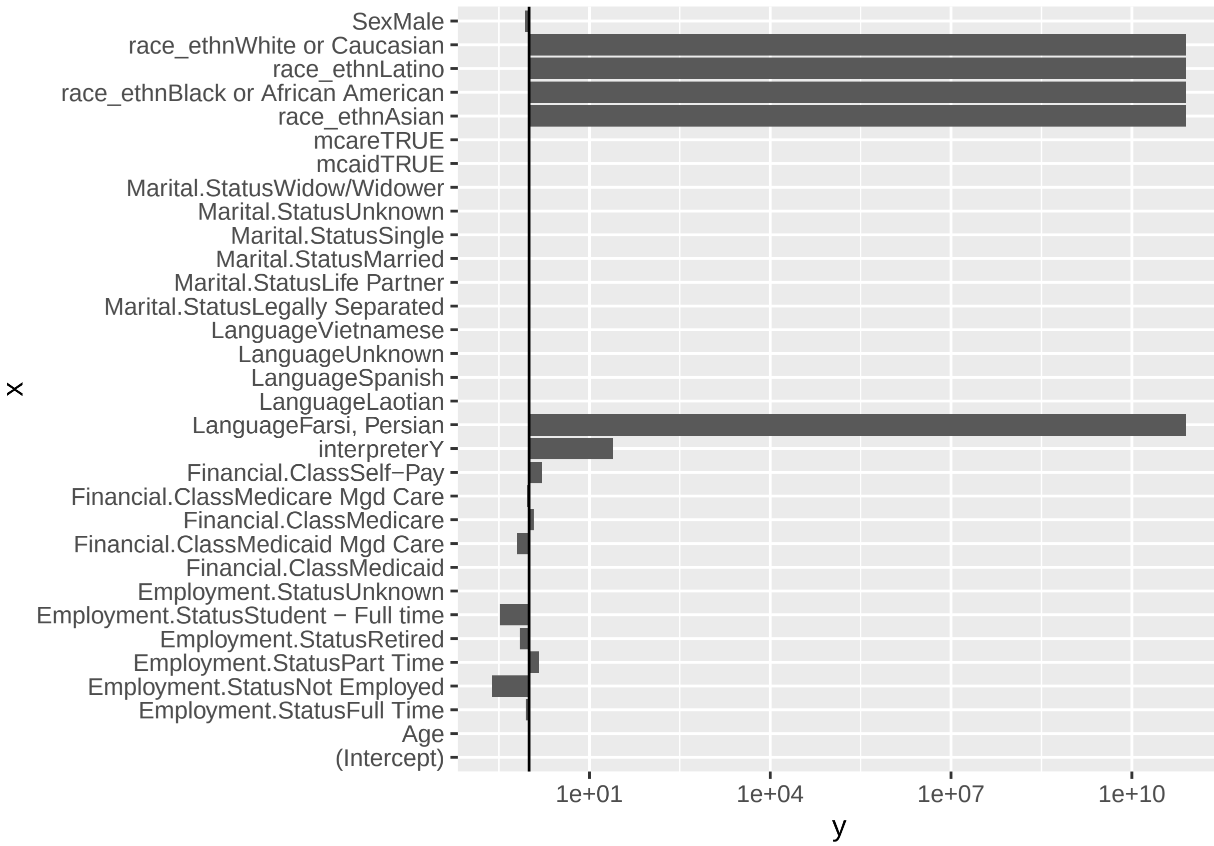
```r
df = data.frame(
  y=my_model$coefficients,
  x=names(my_model$coefficients)
)

ggplot(df, aes(x=y, y=x) ) + geom_col() + scale_x_log10() +
  geom_vline(xintercept = 1)
```

```
## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous x-axis

## Warning: Removed 16 rows containing missing values (position_stack).
```



To do, or not

- Limit only to those discharged from ED to home.
- More rigorous handling of those with multiple demographic entries.
- Get complete note data from IT.
- Show length distribution of notes.