Regression Analysis: Linear Models

Name:Elise Buellesbach Course: STAT 427

Roland Abi Elise Buellesbach Amanda Concepcion
Spencer Grewe

2024-04-24

Table of Contents

1 Executive Summary

In this paper, we dive into California's medical system. Almost 40 million people call California home, as such understanding the efficacy of the California hospital system proves relevant. In this study, we seek to gain a better understanding of the factors that are associated with in-patient deaths in the CA hospital system. Increased staffing, according to the NIH, is associated with decreased levels of in-patient death. Based on this concept, we want to investigate other demographic and economic factors that may be impacting in-patient health.

To start, we completed a series of three regression analyses. In this analysis we want to know if we can predict in-patient deaths due to heart disease per thousand people. We start with a multiple linear regression that provides an unsatisfactory model with only two significant predictor variables. Next, we completed a LASSO regression which yielded insight into how the median income and age of the population impact deaths due to heart disease. The final regression model, a random forest model, reinforces the results from our other models. Through all three models, the number of cases of heart disease and the hospital rating are the most prevalent predictors. Age also continues to be a relevant variable with the population over the age of 55 most at risk. Based on weighting the importance of providing a detailed and robust model with the costs of adding additional variables and the risk of over fitting the data, we recommend using the random forest model.

Next, we turn to a slightly different question regarding the CA hospital system. We complete three classification analyses to try to identify hospitals in Health Professional Shortage Areas (HPSA) based on demographics and geographic location details. To start, we conducted a K-fold model which produced a reduced model based on geographic and population factors. Next, we created a Tree model for the classification where we recommend the usage of the pruned tree. Finally, we used a boost model that produced similar results to the K-fold model. In the end, we recommend the cross-validated boosting model with a shrinkage parameter.

Through this analysis, we gain insight into the California hospital system. We aim to build prediction models for both heart disease in-patient deaths and Health Professional Shortage Areas. HPSA is associated with higher mortality rates of in-patients and thus understanding outcomes in the hospital as well as staffing together provides a broader picture of California hospitals.

2 Introduction, Data Overview

Data preparation code and documentation was omitted from this report, but can be found within the source code in the GitHub repository under `Data_Clean.qmd`.

3 Classification Models

3.1 Method 1

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.0     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
```

3.2 Loading the dataset

```
hpsa <- read_csv("./Data/hpsa.csv")
glimpse(hpsa)
```

```
Rows: 587
Columns: 8
$ metropolitan_indicator             <chr> "Metropolitan", "Non-Metropolitan",~
$ designation_type                   <chr> "Geographic HPSA", "HPSA Population~
$ hpsa_status                        <chr> "Withdrawn", "Withdrawn", "Withdraw~
$ hpsa_score                         <dbl> 0, 0, 0, 0, 11, 8, 8, 0, 0, 10, 10,~
$ hpsa_designation_population        <dbl> 0, 17542, 17542, 17542, 37037, 3608~
$ u_s_mexico_border_county_indicator <chr> "N", "Y", "Y", "N", "N", "N", "N", ~
$ rural_status                       <chr> "non_rural", "rural", "non_rural", ~
$ hpsa_population_type               <chr> "native_american", "native_american~
```

3.2.1 Creating training and testing sets

Here we will be using 60% of our dataset for training and 40% for testing

```
set.seed(123)

training_pct <- .60
Z <- sample(nrow(hpsa), training_pct*nrow(hpsa))

hpsa_train <- hpsa[Z, ]
hpsa_test <- hpsa[-Z, ]

nrow(hpsa_test)
```

```
[1] 235
```

```
nrow(hpsa_train)
```

```
[1] 352
```

3.2.2 Models

```
set.seed(123)
log_full <- glm(as.factor(hpsa_status) ~., data = hpsa_train,
                family = "binomial")

summary(log_full)
```

```
Call:
glm(formula = as.factor(hpsa_status) ~ ., family = "binomial",
    data = hpsa_train)

Coefficients:
                                          Estimate Std. Error z value
(Intercept)                             2.554e+01  7.302e+03   0.003
metropolitan_indicatorMetropolitan      -1.016e+00  7.490e+03   0.000
metropolitan_indicatorNon-Metropolitan  -7.540e-01  7.555e+03   0.000
metropolitan_indicatorUnknown           -2.095e+01  7.302e+03  -0.003
```

```
designation_typeHigh Needs Geographic HPSA -3.710e-01  5.265e-01  -0.705
designation_typeHPSA Population            -2.182e+00  1.101e+00  -1.982
hpsa_score                                 -1.814e-01  5.895e-02  -3.077
hpsa_designation_population                 1.931e-06  6.643e-06   0.291
u_s_mexico_border_county_indicatorY        -8.402e-02  7.896e-01  -0.106
rural_statuspartially_rural                 3.482e-02  5.735e+03   0.000
rural_statusrural                           7.706e-02  4.162e-01   0.185
rural_statusunknown                        -1.807e+00  8.538e-01  -2.117
hpsa_population_typenative_american        -3.201e+00  1.070e+00  -2.992
                                           Pr(>|z|)
(Intercept)                                 0.99721
metropolitan_indicatorMetropolitan          0.99989
metropolitan_indicatorNon-Metropolitan      0.99992
metropolitan_indicatorUnknown               0.99771
designation_typeHigh Needs Geographic HPSA  0.48103
designation_typeHPSA Population             0.04742 *
hpsa_score                                  0.00209 **
hpsa_designation_population                 0.77130
u_s_mexico_border_county_indicatorY         0.91526
rural_statuspartially_rural                 1.00000
rural_statusrural                           0.85312
rural_statusunknown                         0.03427 *
hpsa_population_typenative_american         0.00277 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 453.02  on 351  degrees of freedom
Residual deviance: 191.42  on 339  degrees of freedom
AIC: 217.42

Number of Fisher Scoring iterations: 19
```

```r
set.seed(123)
log_reduced <- glm(as.factor(hpsa_status) ~ designation_type + hpsa_score +
                     rural_status + hpsa_population_type, data = hpsa_train,
                   family = "binomial")

summary(log_reduced)
```

```
Call:
glm(formula = as.factor(hpsa_status) ~ designation_type + hpsa_score +
    rural_status + hpsa_population_type, family = "binomial",
    data = hpsa_train)

Coefficients:
                                          Estimate Std. Error z value
(Intercept)                                6.25862    0.93204   6.715
designation_typeHigh Needs Geographic HPSA -0.64684    0.45256  -1.429
designation_typeHPSA Population            -1.46585    0.60276  -2.432
hpsa_score                                 -0.31939    0.03875  -8.242
rural_statuspartially_rural                13.09281  759.83452   0.017
rural_statusrural                          -0.05037    0.33005  -0.153
rural_statusunknown                        -2.66319    0.83559  -3.187
hpsa_population_typenative_american        -2.41549    0.59263  -4.076
                                          Pr(>|z|)
(Intercept)                               1.88e-11 ***
designation_typeHigh Needs Geographic HPSA  0.15293
designation_typeHPSA Population             0.01502 *
hpsa_score                                 < 2e-16 ***
rural_statuspartially_rural                 0.98625
rural_statusrural                           0.87871
rural_statusunknown                         0.00144 **
hpsa_population_typenative_american        4.58e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 453.02  on 351  degrees of freedom
Residual deviance: 249.95  on 344  degrees of freedom
AIC: 265.95

Number of Fisher Scoring iterations: 15
```

3.2.3 Prediction Accuracy

```r
# Reduced model
# Predictions
preds_reduced <- predict(log_reduced, newdata = hpsa_test, type = "response")
```

```r
# Convert to classification prediction
Yhat_reduced <-  ifelse(preds_reduced >= 0.5, "Withdrawn", "Designated")

# confusion matrix
confm_reduced <- table(Yhat_reduced, hpsa_test$hpsa_status)
confm_reduced
```

```
Yhat_reduced Designated Withdrawn
  Designated         58        15
  Withdrawn          29       133
```

```r
# Correct classification prediction rate
accuracy_reduced <- sum(confm_reduced[1], confm_reduced[4])/sum(confm_reduced)
accuracy_reduced
```

```
[1] 0.812766
```

```r
# test error rate
error_reduced <- mean(Yhat_reduced != hpsa_test$hpsa_status)
error_reduced
```

```
[1] 0.187234
```

```r
# Full model
# Predictions
preds <- predict(log_full, newdata = hpsa_test, type = "response")

# Convert to classification prediction
Yhat <-  ifelse(preds >= 0.5, "Withdrawn", "Designated")

# confusion matrix
confm <- table(Yhat, hpsa_test$hpsa_status)
confm
```

```
Yhat         Designated Withdrawn
  Designated         77        15
  Withdrawn          10       133
```

```r
# Correct classification prediction rate
accuracy <- sum(confm[1], confm[4])/sum(confm)
accuracy
```

```
[1] 0.893617
```

```r
# test error rate
error_full <- mean(Yhat != hpsa_test$hpsa_status)
error_full
```

```
[1] 0.106383
```

The reduced model has an accuracy of 0.8095238 and a mean prediction error rate of 0.1904762 while the full model has an accuracy of 0.7993197 and a prediction error rate of 0.2006803. This reveals an increased accuracy in the reduced model and a decrease in the prediction error rate as compared to the full model.

3.2.4 Using K-fold cross validation to find prediction mean squared error

```r
set.seed(123)

# Loss function
Lossfn <- function(Y, p) {
  mean(1 * (Y == 1 & p <= .50) | (1 * (Y == 0 & p > .50)),
       na.rm = TRUE)
}

# Convert response to numeric 0's and 1's
hpsa$Y <- as.numeric(as.factor(hpsa$hpsa_status)) - 1
```

```r
library(boot)
set.seed(123)

## Prediction error rate
# K = 10

# KFold Reduced
Kfold_reduced <- cv.glm(hpsa, log_reduced, cost = Lossfn, K=10)$delta
Kfold_reduced
```

```
[1] 0.4837234 0.1995639
```

```
# Full Model
Kfold_full <- cv.glm(hpsa, log_full, cost = Lossfn, K=10)$delta
Kfold_full
```

```
[1] 0.4546211 0.1497533
```

While the full and reduced model had similar prediction mean squared error, the reduced model had a slightly lower prediction mean squared error using the k-10 cross validation approach.

3.2.5 Compare models

```
anova(log_reduced, log_full, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: as.factor(hpsa_status) ~ designation_type + hpsa_score + rural_status +
    hpsa_population_type
Model 2: as.factor(hpsa_status) ~ metropolitan_indicator + designation_type +
    hpsa_score + hpsa_designation_population + u_s_mexico_border_county_indicator +
    rural_status + hpsa_population_type
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       344     249.95
2       339     191.42  5   58.529 2.446e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Giving the results obtained and from this approaches and the results of model comparison we lack sufficient evidence ($p = 0.3744$) to conclude that logistic regression (full) model is better than the logistic regression (reduced) model. Consequently, we will recommend the reduced logistic regression model as the better model.

```
library(gt)
tibble(
  models = c("Reduced", "Full"),
  Accuracy = c(accuracy_reduced, accuracy),
  Error = c(error_reduced, error_full)
) |>
```

```
  gt() |>
  tab_header(title = "Accuracy and Prediction Error Estimates Logistic regression models")
```

Accuracy and Prediction Error Estimates Logistic regression models

models	Accuracy	Error
Reduced	0.812766	0.187234
Full	0.893617	0.106383

3.3 Method 2

```
hpsa <- read_csv("Data/hpsa.csv")
```

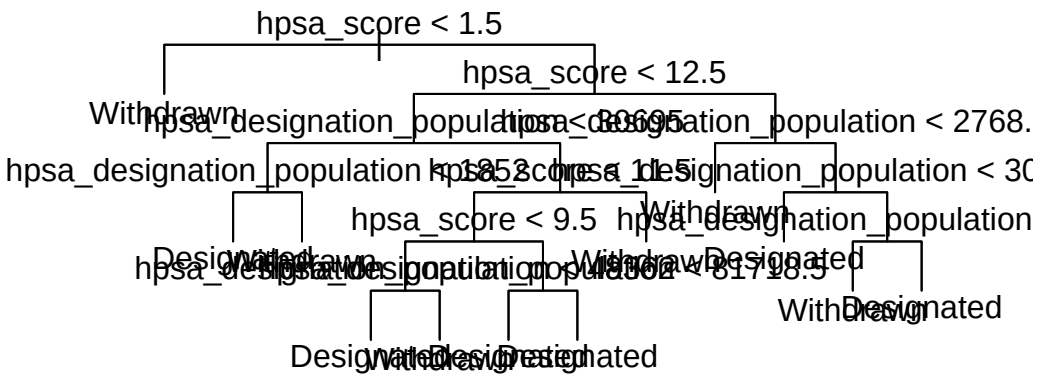
```
set.seed(123)
Z <- sample(1:nrow(hpsa), 260)
train <- hpsa[Z,]
test <- hpsa[-Z,]
#50 score, 0.1
#20 uses designation population , error = 0.15
#70 looks good, score and population ; 0.08571
#90 ^^ ; 0.1
#140, ^^; 0.09286
#170 ^ ; 0.1353
#220 ^ 0.1273
#240 ^ ; 0.1083
#260; 0.1
#280 ^, 0.1107

#STOP AT 410 #70 percent of the observations
```

3.4 Creating an Unpruned Tree

```
set.seed(123)

tr <- tree(as.factor(hpsa_status) ~ ., data = train)
summary(tr)
```

```
Classification tree:
tree(formula = as.factor(hpsa_status) ~ ., data = train)
Variables actually used in tree construction:
[1] "hpsa_score"                  "hpsa_designation_population"
Number of terminal nodes:  12
Residual mean deviance:  0.6527 = 161.9 / 248
Misclassification error rate: 0.1385 = 36 / 260
```

```
plot(tr, type = "uniform")
text(tr)
```




```
Yhat = predict(tr, newdata = test, type = "class")
table(Yhat, test$hpsa_status)
```

```
Yhat         Designated Withdrawn
  Designated         83        35
  Withdrawn          32       177
```

```r
mean(Yhat != test$hpsa_status) #test error rate
```

```
[1] 0.204893
```

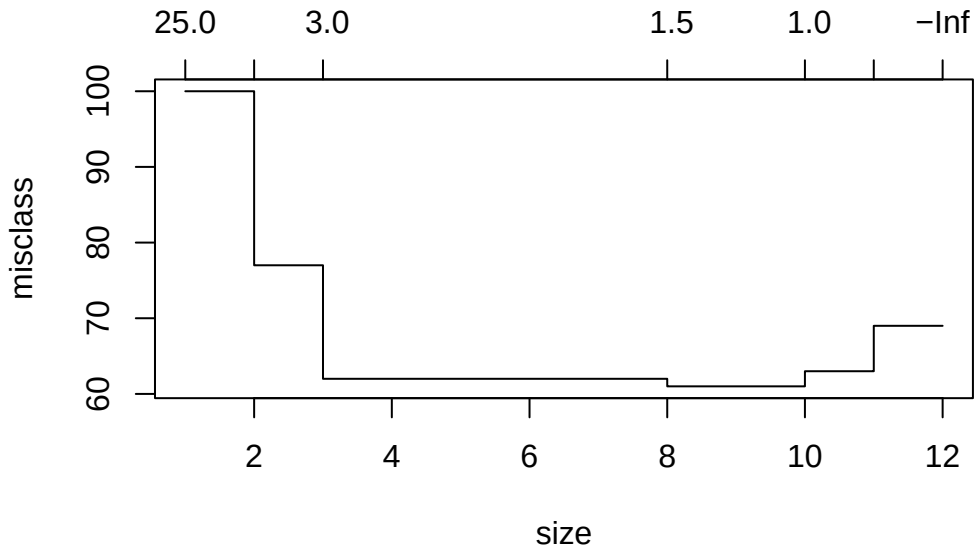
3.5 Creating a Pruned Tree

Uses cv.tree on the training set to determine the optimal tree size based on the misclassification rate

```r
set.seed(123)
cv <- cv.tree(tr, FUN = prune.misclass)
cv
```

```
$size
[1] 12 11 10  8  3  2  1

$dev
[1]  69  69  63  61  62  77 100

$k
[1] -Inf  0.0  1.0  1.5  3.0 13.0 25.0

$method
[1] "misclass"

attr(,"class")
[1] "prune"         "tree.sequence"
```

```r
plot(cv)
```




```
which.min(cv$dev)
```

```
[1] 4
```

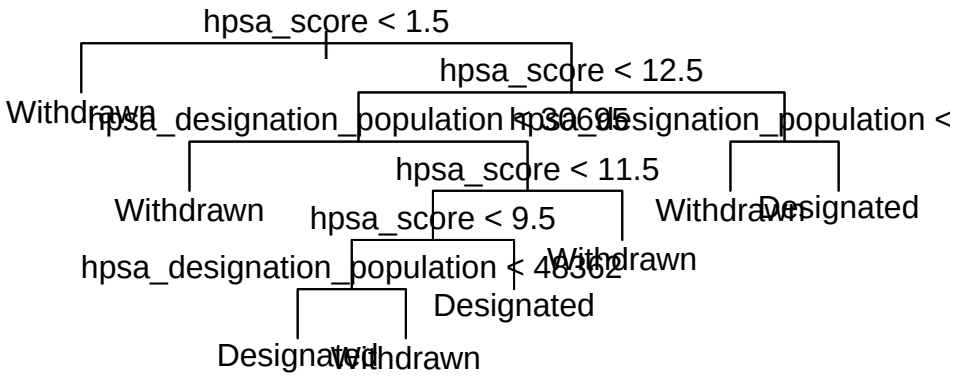
Tree size 8 corresponds to the lowest cross-validated classification error rate with the fewest nodes.

```
set.seed(123)
trp <- prune.misclass(tr, best = 8)
summary(trp)
```

```
Classification tree:
snip.tree(tree = tr, nodes = c(53L, 12L, 15L))
Variables actually used in tree construction:
[1] "hpsa_score"                  "hpsa_designation_population"
Number of terminal nodes:  8
Residual mean deviance:  0.7143 = 180 / 252
Misclassification error rate: 0.1538 = 40 / 260
```

“HPSA Score” and HPSA designation population remains in this node, but with reduced nodes.

```
plot(trp, type = "uniform")
text(trp)
```



3.6 Comparing the Error Rates of the Trees

```
#Unpruned Tree
yhat_unpruned = predict(tr, newdata = test, type = "class")
cfmatrix_unpruned <- table(yhat_unpruned, test$hpsa_status)
test_error_rate_unpruned <- mean(yhat_unpruned != test$hpsa_status)


accuracy_unpruned <- sum(cfmatrix_unpruned[1], cfmatrix_unpruned[4])/sum(cfmatrix_unpruned)

accuracy_unpruned
```

```
[1] 0.795107
```

```
test_error_rate_unpruned
```

```
[1] 0.204893
```

```
#Pruned Tree
yhat_pruned = predict(trp, newdata = test, type = "class")
cfmatrix_pruned <- table(yhat_pruned, test$hpsa_status)
test_error_rate_pruned <- mean(yhat_pruned != test$hpsa_status)

accuracy_pruned <- sum(cfmatrix_pruned[1], cfmatrix_pruned[4])/sum(cfmatrix_pruned)

accuracy_pruned
```

```
[1] 0.7981651
```

```
test_error_rate_pruned
```

```
[1] 0.2018349
```

The test error rates are not equal. The pruned tree has an error rate of 0.2018349 which is a little lower than the unpruned tree which has an error rate of 0.204893. Meanwhile the accuracy of the pruned tree is 0.7981651 which is higher than that of the unpruned tree 0.795107 . We recommend the pruned tree more.

```
library(gt)

tibble(
  models = c("Unpruned", "Pruned"),
  Accuracy = c(accuracy_unpruned, accuracy_pruned),
  Error = c(test_error_rate_unpruned, test_error_rate_pruned)
) |>
  gt() |>
  tab_header(title = "Accuracy and Prediction Error Estimates Tree models")
```

Accuracy and Prediction Error Estimates Tree models

models	Accuracy	Error
Unpruned	0.7951070	0.2048930
Pruned	0.7981651	0.2018349

3.7 Method 3

wants to do boosting model , compared to SVM

```
library(gbm3)
```

```
#Convert each character variable to a factor
glimpse(hpsa)
```

```
Rows: 587
Columns: 8
$ metropolitan_indicator             <chr> "Metropolitan", "Non-Metropolitan",~
$ designation_type                   <chr> "Geographic HPSA", "HPSA Population~
$ hpsa_status                        <chr> "Withdrawn", "Withdrawn", "Withdraw~
$ hpsa_score                         <dbl> 0, 0, 0, 0, 11, 8, 8, 0, 0, 10, 10,~
$ hpsa_designation_population        <dbl> 0, 17542, 17542, 17542, 37037, 3608~
$ u_s_mexico_border_county_indicator <chr> "N", "Y", "Y", "N", "N", "N", "N", ~
$ rural_status                       <chr> "non_rural", "rural", "non_rural", ~
$ hpsa_population_type               <chr> "native_american", "native_american~
```

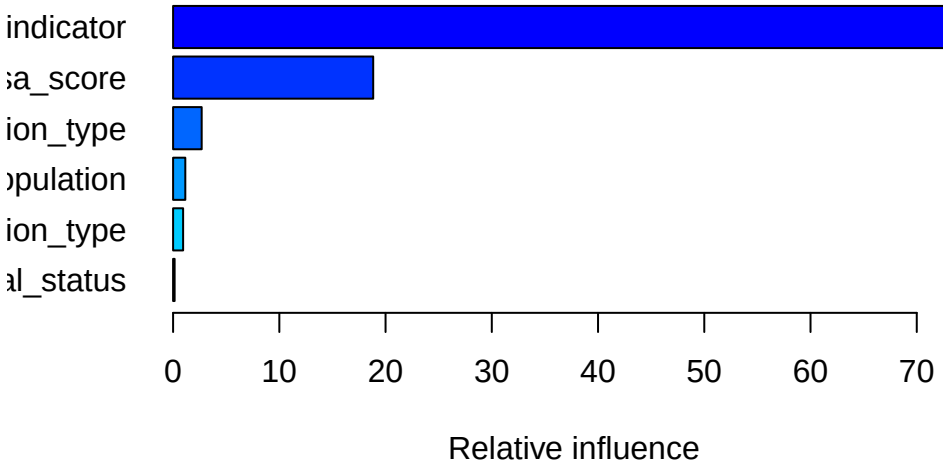
```
hpsa2 <- mutate(hpsa, across(where(is.character), as.factor))
```

```
set.seed(123)
Z <- sample(nrow(hpsa2), nrow(hpsa2)/2)
train <- hpsa2[Z,]
test <- hpsa2[-Z,]
```

```
set.seed(123)
boosth <- gbm(hpsa_status ~., data = train,
              n.trees = 3000, distribution = "Bernoulli")
boosth
```

```
gbm(formula = hpsa_status ~ ., distribution = "Bernoulli", data = train,
    n.trees = 3000)
A gradient boosted model with Bernoulli loss function.
 3000 iterations were performed.
There were 7 predictors of which 7 had non-zero influence.
```

```
summary(boosth, cBars = 6)
```




```
                                                                  var
metropolitan_indicator                         metropolitan_indicator
hpsa_score                                                 hpsa_score
hpsa_population_type                         hpsa_population_type
hpsa_designation_population               hpsa_designation_population
designation_type                                     designation_type
rural_status                                             rural_status
u_s_mexico_border_county_indicator u_s_mexico_border_county_indicator
                                        rel_inf
metropolitan_indicator                76.228639138
hpsa_score                            18.837306324
hpsa_population_type                   2.689845808
hpsa_designation_population            1.149605087
designation_type                       0.944460900
rural_status                           0.142845378
u_s_mexico_border_county_indicator     0.007297366
```

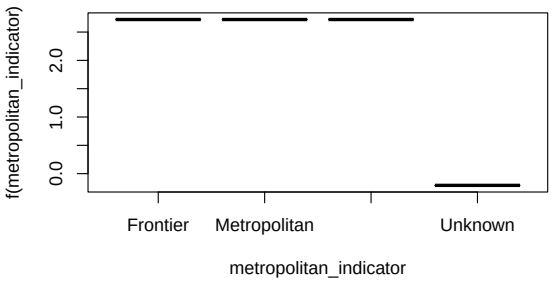
```
#To see the tree structure
gbm::pretty.gbm.tree(boosth)
```

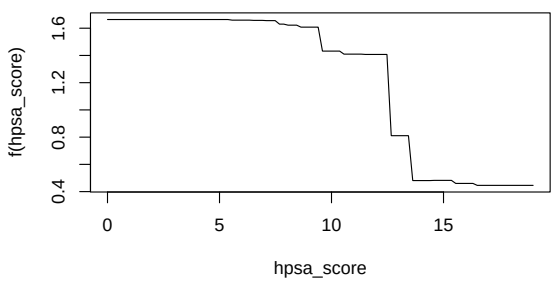
```
  SplitVar SplitCodePred LeftNode RightNode MissingNode ErrorReduction Weight
0        0   0.000000000        1         2           3       15.53109    146
1       -1  -0.001390261       -1        -1          -1        0.00000     78
2       -1   0.001518135       -1        -1          -1        0.00000     68
3       -1   0.000000000       -1        -1          -1        0.00000    146
     Prediction
0 -1.783288e-05
1 -1.390261e-03
2  1.518135e-03
3  0.000000e+00
```

```
gbm::pretty.gbm.tree(boosth, i.tree = 3000)
```

```
  SplitVar SplitCodePred LeftNode RightNode MissingNode ErrorReduction Weight
0        0  1.939000e+03        1         2           3      0.6895292    146
1       -1 -4.150093e-04       -1        -1          -1      0.0000000     78
2       -1  1.059236e-03       -1        -1          -1      0.0000000     68
3       -1  0.000000e+00       -1        -1          -1      0.0000000    146
     Prediction
0  0.0001358127
1 -0.0004150093
2  0.0010592359
3  0.0000000000
```

```
plot(boosth, "metropolitan_indicator")
plot(boosth, "hpsa_score")
#plot(boosth, c("hpsa_population_type", "hpsa_score"))
```





3.8 Prediction

```
yhat <- predict(boosth, newdata = test, n.trees = 3000, type = "response")

# Convert to classification prediction
Yhat_class <-  ifelse(yhat >= 0.5, "Withdrawn", "Designated")
cfm_yhat <- table(Yhat_class, test$hpsa_status)

cfm_yhat
```

```
Yhat_class   Designated Withdrawn
  Designated         93        21
  Withdrawn          15       165
```

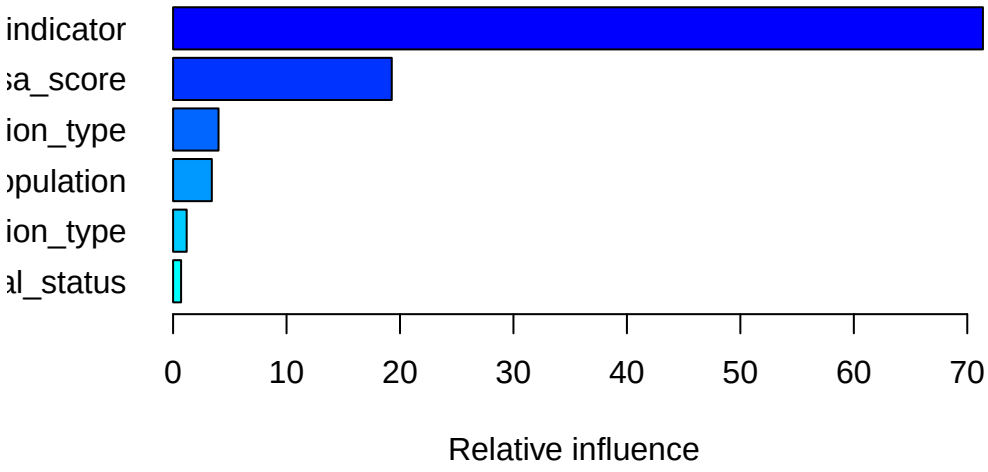
```
# prediction error rate and accuracy
yhat_error <- mean(Yhat_class != test$hpsa_status)
accuracy_yhat <- sum(cfm_yhat[1], cfm_yhat[4])/sum(cfm_yhat)

yhat_error
```

```
[1] 0.122449
```

```
accuracy_yhat
```

```
[1] 0.877551
```

```
#adjust the shrinkage parameter
set.seed(123)
boosth_shrink <- gbm(hpsa_status ~ ., data = train,
              n.trees = 5000, shrinkage = 0.001, distribution = "Bernoulli")
summary(boosth_shrink, cBars = 6)
```




```
                                                                  var
metropolitan_indicator                           metropolitan_indicator
hpsa_score                                                 hpsa_score
hpsa_population_type                         hpsa_population_type
hpsa_designation_population               hpsa_designation_population
designation_type                                     designation_type
rural_status                                             rural_status
u_s_mexico_border_county_indicator u_s_mexico_border_county_indicator
                                        rel_inf
metropolitan_indicator                71.37882009
hpsa_score                            19.27404814
hpsa_population_type                   4.00518646
hpsa_designation_population            3.41247481
designation_type                       1.19690539
rural_status                           0.70524554
u_s_mexico_border_county_indicator     0.02731957
```

```r
yhat2 <- predict(boosth_shrink, newdata = test, n.trees = 3000, type = "response")

# Convert to classification prediction
Yhat2_class <-  ifelse(yhat2 >= 0.5, "Withdrawn", "Designated")
cfm_yhat2 <- table(Yhat2_class, test$hpsa_status)
```

```
cfm_yhat2
```

```
Yhat2_class  Designated Withdrawn
  Designated         93        21
  Withdrawn          15       165
```

```
# prediction error rate and accuracy
yhat2_error <- mean(Yhat2_class != test$hpsa_status)
accuracy_yhat2 <- sum(cfm_yhat2[1], cfm_yhat2[4])/sum(cfm_yhat2)

yhat2_error
```

```
[1] 0.122449
```

```
accuracy_yhat2
```

```
[1] 0.877551
```

Reducing the shrinkage rate will cause the prediction prediction error rate stay the same.

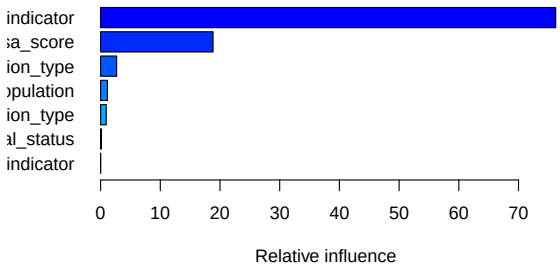
3.9 Cross Validation for the Number of Trees

```
set.seed(123)
boosth_cv <- gbm(hpsa_status ~ ., data = train,
                 n.trees = 3000, shrinkage = 0.001,
                 cv.folds = 10,
                 distribution = "Bernoulli")
boosth_cv
```

```
gbm(formula = hpsa_status ~ ., distribution = "Bernoulli", data = train,
    n.trees = 3000, shrinkage = 0.001, cv.folds = 10)
A gradient boosted model with Bernoulli loss function.
 3000 iterations were performed.
The best cross-validation iteration was 3000.
There were 7 predictors of which 7 had non-zero influence.
```

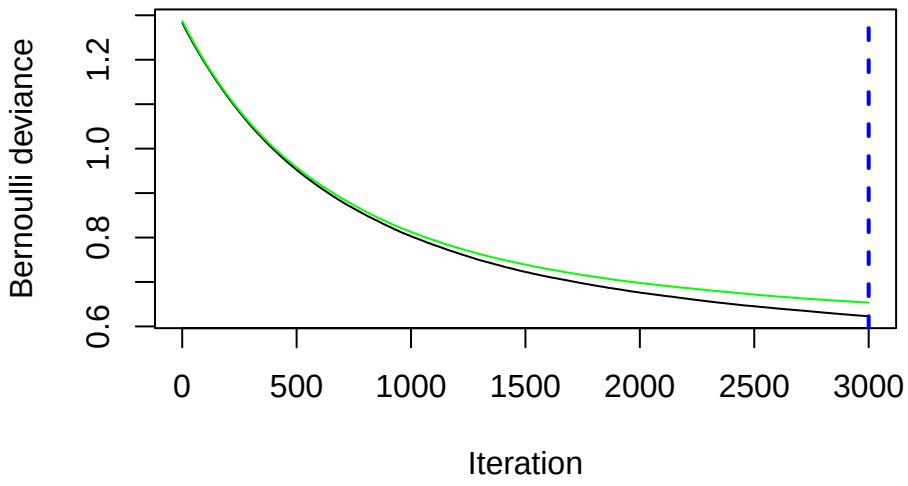
```
Cross-validation confusion matrix:
   0   1 Pred. Acc.
0 85  15       85.0
1 33 160       82.9

Cross-validation prediction Accuracy = 83.62%
```

```
summary(boosth_cv)
```




```
                                                                  metropo
metropolitan_indicator
hpsa_score
hpsa_population_type                                                hpsa_
hpsa_designation_population                                 hpsa_designa
designation_type                                                        d
rural_status
u_s_mexico_border_county_indicator u_s_mexico_border_c
                                                    rel_inf
metropolitan_indicator                           76.228639138
hpsa_score                                       18.837306324
hpsa_population_type                              2.689845808
hpsa_designation_population                       1.149605087
designation_type                                  0.944460900
rural_status                                      0.142845378
u_s_mexico_border_county_indicator                0.007297366
```

```
gbm.perf(boosth_cv, method = "cv")
```




```
[1] 3000
```

```
yhat3 <- predict(boosth_cv, newdata = test, n.trees = 3000, type = "response")

# Convert to classification prediction
Yhat3_class <-  ifelse(yhat3 >= 0.5, "Withdrawn", "Designated")
cfm_yhat3 <- table(Yhat3_class, test$hpsa_status)

cfm_yhat3
```

```
Yhat3_class  Designated Withdrawn
  Designated         93        21
  Withdrawn          15       165
```

```
# prediction error rate and accuracy
yhat3_error <- mean(Yhat3_class != test$hpsa_status)
accuracy_yhat3 <- sum(cfm_yhat3[1], cfm_yhat3[4])/sum(cfm_yhat3)

yhat3_error
```

```
[1] 0.122449
```

```
accuracy_yhat3
```

```
[1] 0.877551
```

The cross-validated boosted model with shrinkage parameter, boosted model with shrinkage parameter and initial boosted model all have the prediction mean squared error. Top important predictors in the initial and boost_shrinkage models were metropolitan indicator and hpsa score. While the top important predictors of hpsa status in the boost_shrinkage_cv model were hpsa designation population, hpsa status and metropolitan indicator. We will recommend the boost_shrinkage_cv model.

```
library(gt)

tibble(
  models = c("initial model", "boost_shrinkage", "boost_shrinkage_cv"),
  Accuracy = c(accuracy_yhat, accuracy_yhat2, accuracy_yhat3),
  Error = c(yhat_error, yhat2_error, yhat3_error)
) |>
  gt() |>
  tab_header(title = "Accuracy and Prediction Error Estimates of Boosting models")
```

Accuracy and Prediction Error Estimates of Boosting models

models	Accuracy	Error
initial model	0.877551	0.122449
boost_shrinkage	0.877551	0.122449
boost_shrinkage_cv	0.877551	0.122449

4 Regression Models

4.1 Method 1. Multiple Linear and Ridge Regression

Here we are working on a basic multiple linear model with cross validation.

Question: Can mortality rate from heart disease per 1000 people be predicted from demographic conditions within counties in California?

To start, we will employ a series of multiple regressions to get a better understanding of the relationships at play and which variables are most useful in predicting the mortality rate from heart disease.

I begin by importing and setting up the data.

```
library(tidyverse); theme_set(theme_bw())
reg_data_long <- read_csv(".\\Data\\reg_data_long.csv")
# head(reg_data_long)
reg_data_long$female_prop <- reg_data_long$FemalePop/reg_data_long$TotalPop
```

When I initially began building this model, I ran into issues in calculating the MSE because of a couple of NA values. To prevent this issue, I checked the data for NA values. Both PercFemale and PercUnder18yo have NA values and were thus removed from the full regression model.

```
na_counts <- colSums(is.na(reg_data_long))
na_counts
```

```
        County       TotalPop        MalePop      FemalePop      MedianAge
             0              0              0              0              0
 MaleMedianAge FemaleMedianAge     Under5yoPop    Under18yoPop   Pop21andOlder
             0              0              0              0              0
 Pop55andOlder   Pop60andOlder   Pop65andOlder      PercFemale   PercUnder18yo
             0              0              0             12             12
     num_desig     amer_indian          black       hispanic          asian
             0              0              0              0              0
    multi_race      pac_island          white        poverty     labor_force
             0              0              0              0              0
    unemployed   median_income     TotalDeaths     TotalCases      prop_hpsa
             0              0              0              0              0
 nonwhite_prop         Rating         Deaths   DeathsperThou     unempl_prop
             0              0              0              0              0
  poverty_prop    prop55older          Cases     HospNumber    CasesperThou
             0              0              0              0              0
   female_prop
             0
```

Splitting up the data

```
set.seed(123)

training_pct <- .8
Z <- sample(nrow(reg_data_long), floor(training_pct*nrow(reg_data_long)))

train <- reg_data_long[Z,]
```

```
# dim(train)
test <- reg_data_long[-Z,]
# dim(test)

test$County <- factor(test$County, levels = levels(train$County))
```

4.1.1 Full Model

Training the full model

```
full_model <- lm(DeathsperThou ~ TotalPop + MalePop + FemalePop + MedianAge + MaleMedianAge +
```

Looking at the full model, we see that the full model is statistically significant with a p-value of almost zero. Further, it has an adjusted R^2 value of 0.9345 telling us that the model explains 93.45% of the variation in the results. This is a strong result. But, only a handful of the variables are significant in the model. ::: {.cell}

```
summary(full_model)
```

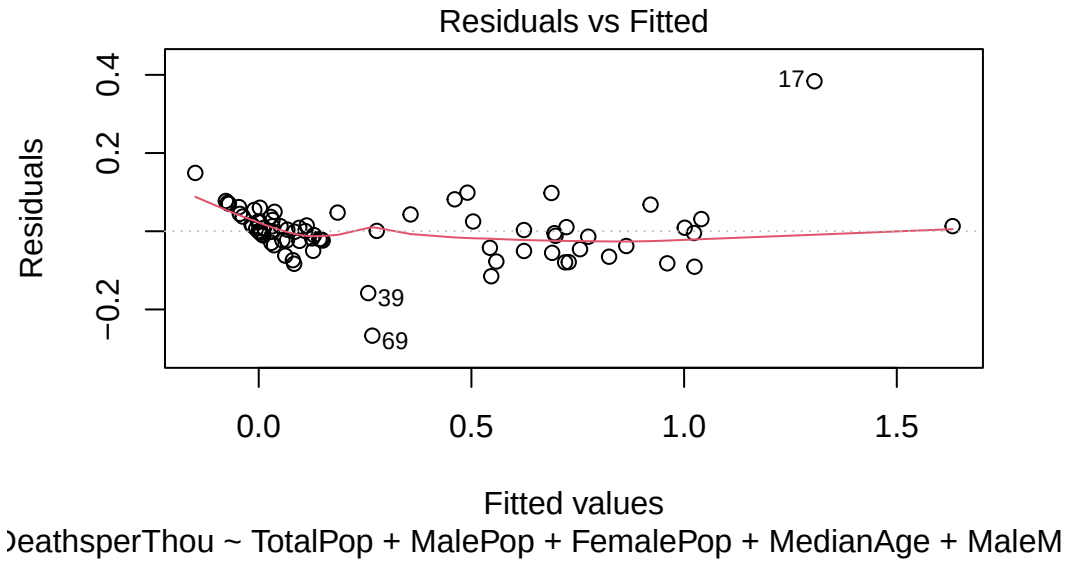
```
Call:
lm(formula = DeathsperThou ~ TotalPop + MalePop + FemalePop +
    MedianAge + MaleMedianAge + FemaleMedianAge + Under5yoPop +
    Under18yoPop + Pop21andOlder + Pop55andOlder + Pop60andOlder +
    Pop65andOlder + num_desig + amer_indian + black + hispanic +
    asian + multi_race + pac_island + white + poverty + labor_force +
    unemployed + median_income + TotalDeaths + TotalCases + prop_hpsa +
    nonwhite_prop + Rating + Deaths + unempl_prop + poverty_prop +
    prop55older + Cases + HospNumber + CasesperThou + female_prop,
    data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.26702 -0.02710  0.00018  0.02693  0.38361

Coefficients: (1 not defined because of singularities)
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      1.476e+01  9.677e+00   1.525  0.13482
TotalPop        -5.931e-05  3.533e-05  -1.679  0.10079
MalePop         -3.238e-05  2.615e-05  -1.238  0.22266
```

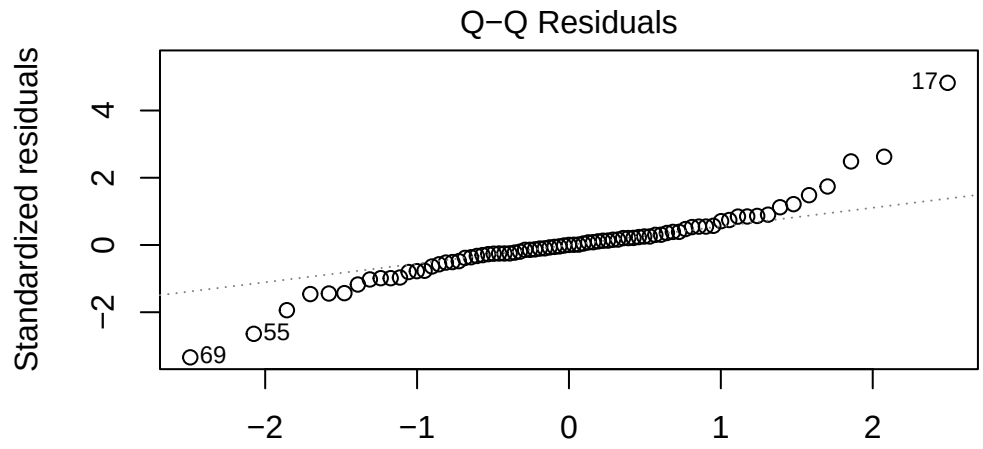
```
FemalePop               NA         NA      NA       NA
MedianAge         3.873e-01  2.140e-01   1.810  0.07769 .
MaleMedianAge    -3.582e-01  1.234e-01  -2.904  0.00592 **
FemaleMedianAge  -4.030e-01  1.508e-01  -2.672  0.01076 *
Under5yoPop      -4.857e-05  7.834e-05  -0.620  0.53872
Under18yoPop      8.693e-05  5.331e-05   1.631  0.11062
Pop21andOlder     9.340e-05  5.649e-05   1.653  0.10588
Pop55andOlder    -6.930e-05  6.245e-05  -1.110  0.27362
Pop60andOlder     1.896e-05  1.107e-04   0.171  0.86482
Pop65andOlder     4.339e-05  7.970e-05   0.544  0.58910
num_desig        -2.381e-03  2.080e-03  -1.145  0.25892
amer_indian       7.318e-06  3.708e-05   0.197  0.84453
black            -6.558e-06  9.342e-06  -0.702  0.48668
hispanic          8.807e-06  5.090e-06   1.730  0.09115 .
asian             4.462e-06  4.295e-06   1.039  0.30494
multi_race        6.853e-05  5.490e-05   1.248  0.21902
pac_island       -2.130e-05  6.064e-05  -0.351  0.72719
white             1.295e-06  4.764e-06   0.272  0.78711
poverty           5.334e-06  5.782e-06   0.922  0.36167
labor_force      -1.302e-05  8.837e-06  -1.474  0.14815
unemployed       -1.713e-05  1.891e-05  -0.906  0.37047
median_income     2.027e-05  1.331e-05   1.523  0.13534
TotalDeaths      -2.665e-04  9.168e-04  -0.291  0.77275
TotalCases       -5.199e-05  1.883e-05  -2.761  0.00858 **
prop_hpsa        -2.052e-02  2.950e-01  -0.070  0.94489
nonwhite_prop    -1.694e+00  1.696e+00  -0.999  0.32367
RatingBetter     -1.170e-01  1.015e-01  -1.153  0.25571
RatingWorse      -4.386e-02  1.007e-01  -0.436  0.66537
Deaths            2.238e-04  1.519e-04   1.473  0.14839
unempl_prop       4.769e+00  4.009e+00   1.190  0.24105
poverty_prop     -4.411e+00  6.737e+00  -0.655  0.51629
prop55older       2.768e+01  1.998e+01   1.386  0.17337
Cases            -3.037e-06  3.359e-06  -0.904  0.37129
HospNumber       -2.326e-04  1.308e-04  -1.778  0.08275 .
CasesperThou      3.264e-02  3.848e-03   8.482 1.46e-10 ***
female_prop      -1.737e+01  1.094e+01  -1.587  0.12011
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.102 on 41 degrees of freedom
Multiple R-squared:  0.9655,    Adjusted R-squared:  0.9345
F-statistic: 31.06 on 37 and 41 DF,  p-value: < 2.2e-16
```

:::

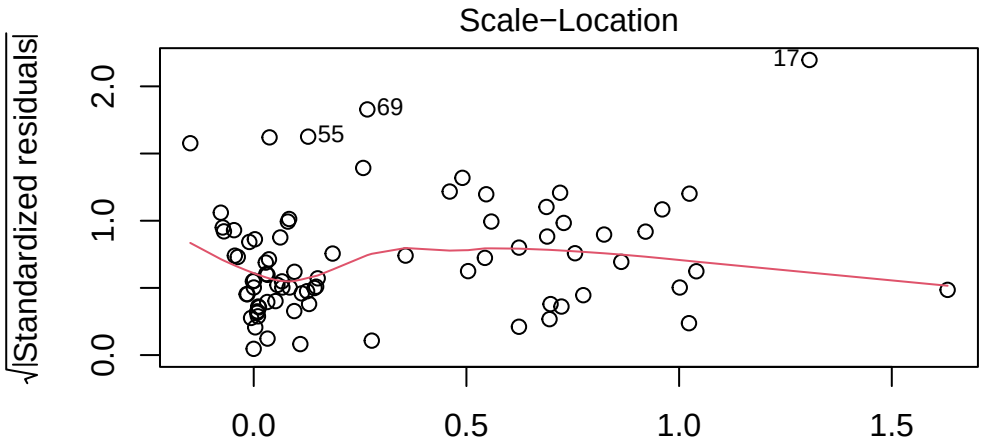
These plots give some insight to the shape of the model. For example, the plotting on the QQ-plot does not demonstrate a strong linear relationship. The edges of the plot deviate from the line suggesting that the model is not perfectly linear. Further, the scattered dots on the Leverage plot show several points on the line of Cook's Distance signifying that they have overstated influence on the model. With these concerns in mind, I want to investigate the linearity of the model further. Does a linear model make sense? To get a better understanding of the data, I created some basic plots to demonstrate the shape of the data and get an idea of if a linear model will make sense here (source on code) ::: {.cell}

```
plot(full_model)
```

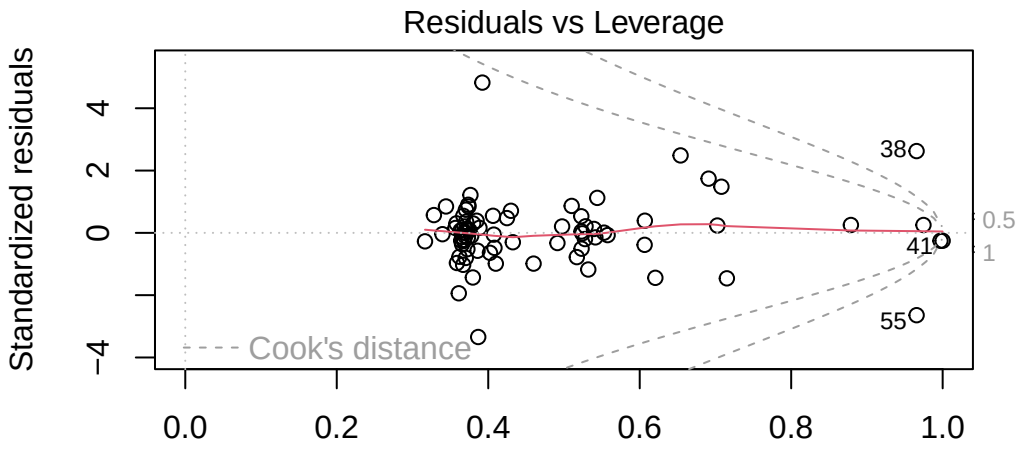




DeathsperThou ~ TotalPop + MalePop + FemalePop + MedianAge + MaleM



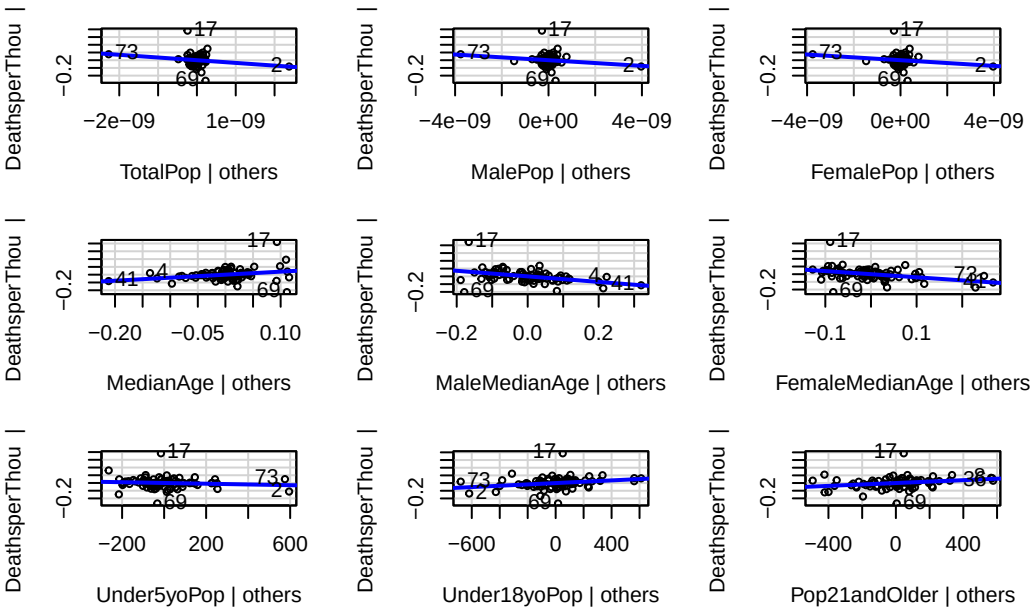
DeathsperThou ~ TotalPop + MalePop + FemalePop + MedianAge + MaleM

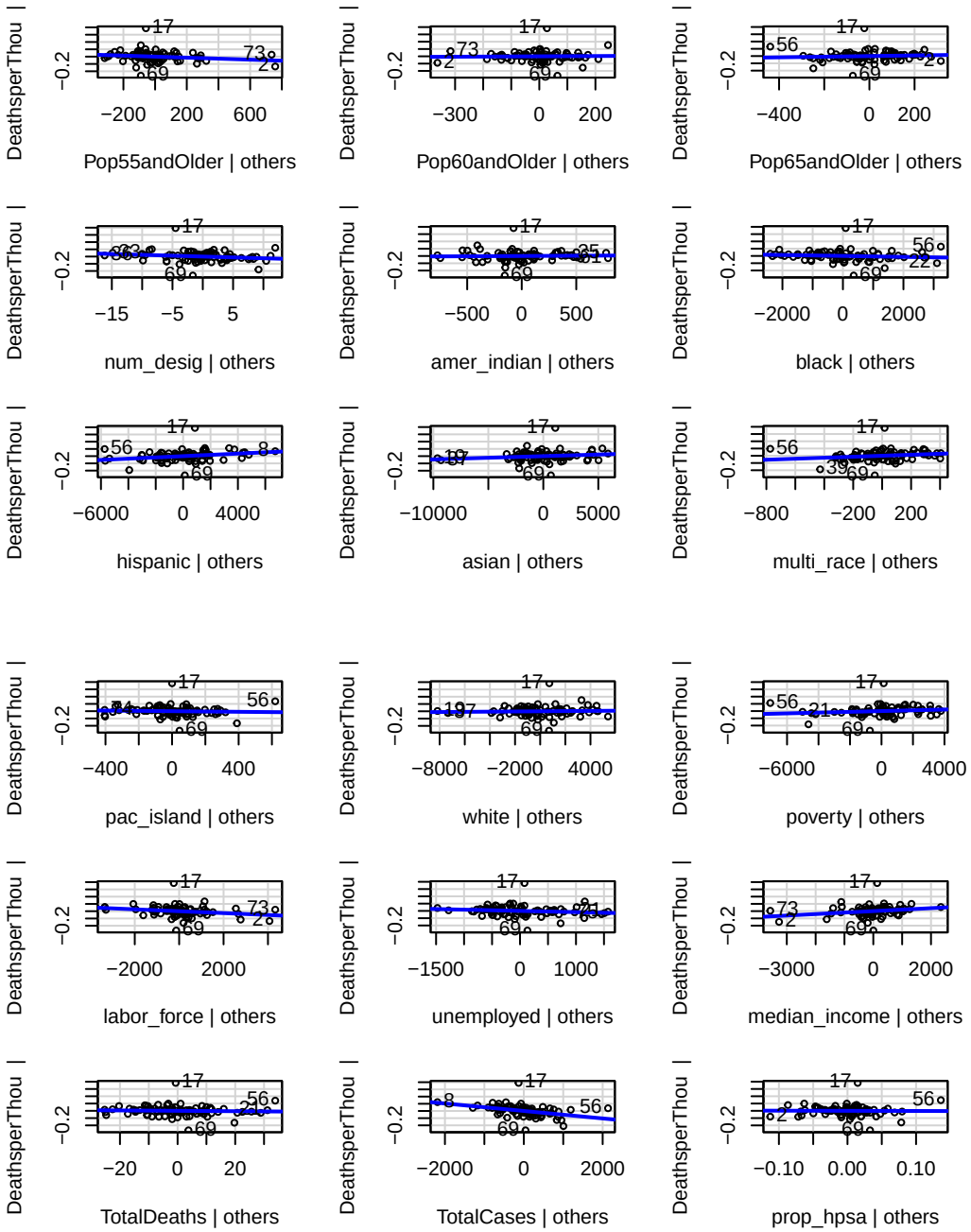


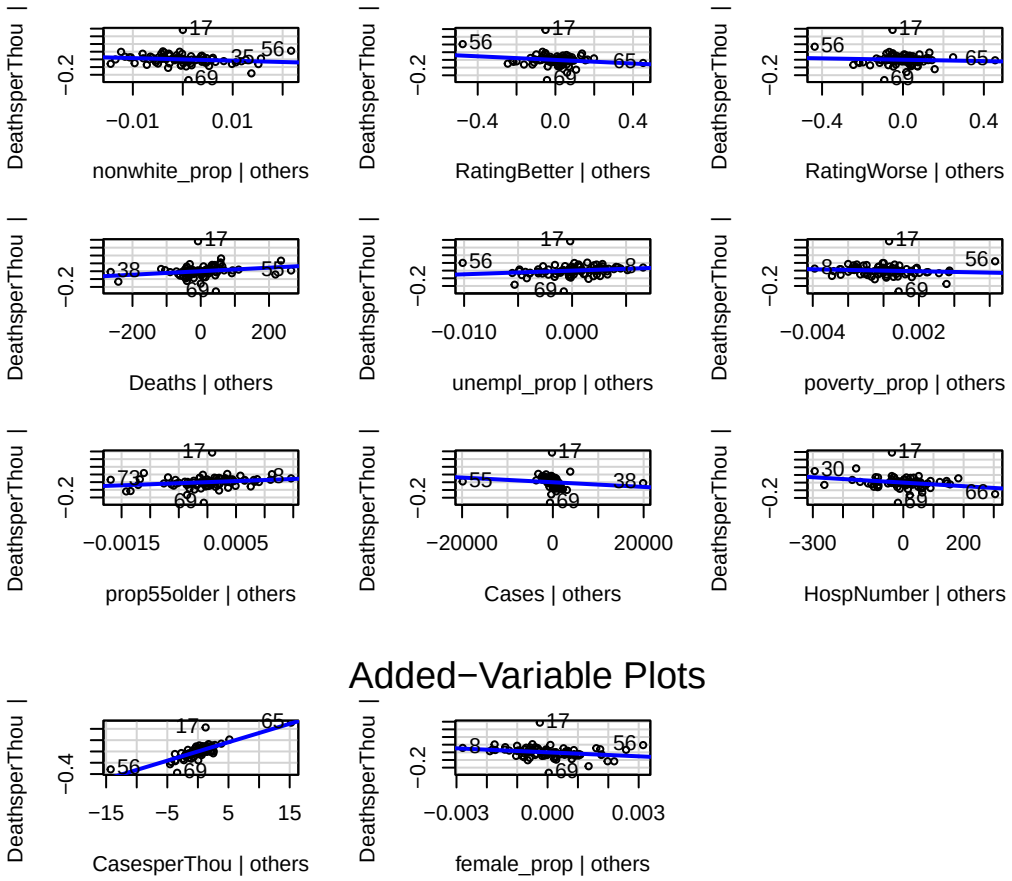
)eathsperThou ~ TotalPop + MalePop + FemalePop + MedianAge + MaleM

:::

```
library(car)
avPlots(full_model)
```




DeathsperThou | others
Pop55andOlder | others
Pop60andOlder | others
Pop65andOlder | others
num_desig | others
amer_indian | others
black | others
hispanic | others
asian | others
multi_race | others
pac_island | others
white | others
poverty | others
labor_force | others
unemployed | others
median_income | others
TotalDeaths | others
TotalCases | others
prop_hpsa | others



Here, we can see that some variables provide a better linear fit than others. For example, variables such as `MaleMedianAge` and `FemaleMedianAge` both are more or less scattered along the regression line. However, other variables such as `MalePop` and `FemalePop` do not fit the line well. Instead, the data is gathered in one spot and has almost no linear shape. These

graphs give some reason for concern that the basic full model will not be the strongest. Not all of the variables are linear so a reduced model might be a better fit.

Next, we will go ahead and calculate the Mean Square Error of the full model.

Testing the full model ::: {.cell}

```
Yhat <- predict(full_model, newdata = test)

mse <- mean((test$DeathsperThou - Yhat)^2)
mse
```

```
[1] 0.07419842
```

::: The MSE is .0742. This is a low MSE in general. Next, we can see how a reduced model will work and then we will have a good point of comparison between the two models.

Here, I will take only the significant variables from the above model for the reduced model. This includes variables that are significant up to the p=.1 threshold This was chosen because so few variables are significant and I want to see what this model will yield. Then, we can remove more variables if the results warrant that action.

4.1.2 Reduced Model

Training the reduced model

```
reduced_model <- lm(DeathsperThou ~ MedianAge + MaleMedianAge + FemaleMedianAge + hispanic +
```

```
summary(reduced_model)
```

```
Call:
lm(formula = DeathsperThou ~ MedianAge + MaleMedianAge + FemaleMedianAge +
    hispanic + CasesperThou + TotalCases + HospNumber, data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.40557 -0.05661 -0.01361  0.05143  0.56690

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     -1.900e-01  1.517e-01  -1.252   0.2146
```

```
MedianAge        2.057e-01  1.394e-01   1.476   0.1443
MaleMedianAge   -1.208e-01  7.612e-02  -1.588   0.1168
FemaleMedianAge -8.090e-02  6.746e-02  -1.199   0.2344
hispanic         3.747e-08  8.951e-08   0.419   0.6768
CasesperThou     3.500e-02  1.482e-03  23.615   <2e-16 ***
TotalCases      -6.504e-07  1.326e-06  -0.491   0.6253
HospNumber      -1.122e-04  4.518e-05  -2.483   0.0154 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1192 on 71 degrees of freedom
Multiple R-squared:  0.9185,    Adjusted R-squared:  0.9104
F-statistic: 114.3 on 7 and 71 DF,  p-value: < 2.2e-16
```

This model yields a p-value of almost zero. It has an adjusted R^2 value of 0.9104 which is lower than the previous value from the full model of 0.9345. However, the decrease in the number of variables might be worth the slightly lower R^2 value.

Testing the reduced model ::: {.cell}

```
prediction_reduced <- predict(reduced_model, newdata = test)
mse_reduced <- mean((test$DeathsperThou - prediction_reduced)^2)
mse_reduced
```

```
[1] 0.05392615
```

:::

However, we see a lower MSE at 0.0539 compared to 0.0742 from the full model. However, `hispanic`, `MedianAge`, `MaleMedianAge`, `FemaleMedianAge`, and `TotalCases` are no longer significant in this model.

To get an even better understanding of the data, we reduce the model again to just the significant variables in the reduced model. We are down to just two predictor variables, `CasesperThou` and `HospNumber`. This makes logical sense that the number of cases per thousand and the ranking of the hospital would be good predictor variables of the deaths per thousand.

```
small_model <- lm(DeathsperThou ~ CasesperThou + HospNumber, data=train)
summary(small_model)
```

```
Call:
lm(formula = DeathsperThou ~ CasesperThou + HospNumber, data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.42998 -0.05724 -0.00882  0.02736  0.69795

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.317e-03  1.768e-02   0.527  0.59983
CasesperThou  3.503e-02  1.451e-03  24.140  < 2e-16 ***
HospNumber   -1.230e-04  4.346e-05  -2.830  0.00595 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.122 on 76 degrees of freedom
Multiple R-squared:  0.9087,    Adjusted R-squared:  0.9063
F-statistic: 378.1 on 2 and 76 DF,  p-value: < 2.2e-16
```

```r
prediction_small <- predict(small_model, newdata = test)
mse_small <- mean((test$DeathsperThou - prediction_reduced)^2)
mse_small
```

```
[1] 0.05392615
```

This small model yields the same MSE as the previous reduced model of 0.0539 and only requires two variables. We still have a p-value of almost zero. However, we continue to lose some power in the adjusted R-squared with a value of 0.9063 compared to 0.9104 in the reduced model and 0.9345 in the full model.

4.1.3 Comparison Between the Two Models and Recommendation

Between the original two models, the reduced model should be chosen. It offers a more accurate prediction of the rate of deaths due to heart disease per thousand people. The reduced model uses only 7 prediction variables compared to the more than 30 variables that the full model requires. Further, the smallest model with only 2 variables gives the same low MSE value with even fewer predictor variables. This implies that many of those variables are either not useful in prediction or they might have high levels of multi-co linearity This concept will be explored further in the next regression models.

4.2 Method 2. LASSO Regression

4.2.1 Further Exploration of Multicollinearity

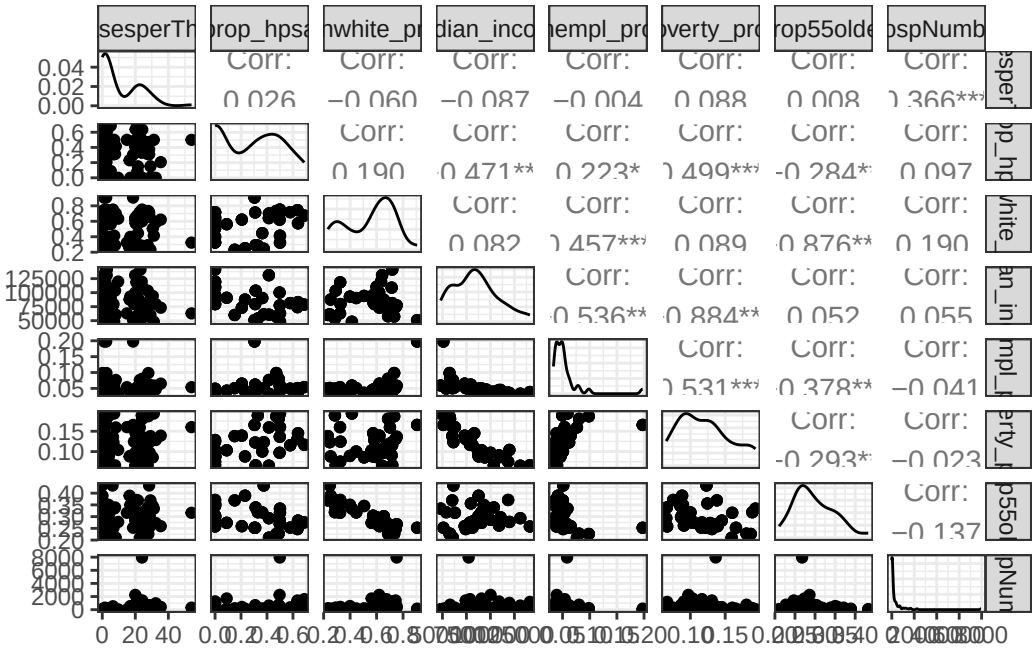


Figure 1: Pairs plot of predictors selected for regression.

In social science and health data, multicollinearity has complex causes and is generally unavoidable. There are a few notable instances of collinearity within the regression predictors. The non-white racial population proportion and the population proportion aged 55 years and older can be reliably predicted from the rest of the data, and certain pairs, such as poverty and unemployment, show evidence of an intuitively strong linear association.

While the cause of these co-correlations is outside of the scope of our analysis, we will now explore two additional regression methods, LASSO and random forest, which use shrinkage or bagging to reduce the variance in the prediction. Unlike simply removing predictors from the analysis, these two methods also allow for unbiased searches across all predictors to construct the best final model.

4.2.2 LASSO Regression

The goal of this regression is to determine whether deaths per 1000 individuals due to heart disease can be predicted given demographic information, as well as information about the

hospitals located within counties in California, controlling for the cases within a county. We employed a 10-fold LASSO regression to determine predictors with non-zero associations with the outcome, controlling for heart disease cases and with a shrinkage penalty applied to lower variance.

Table 1: Coefficients and estimates obtained after LASSO regression at two different L1 penalty strengths.

	Min	X1se
(Intercept)	-0.0936	0.0567
RatingBetter	0.0000	0.0000
RatingWorse	0.0000	0.0000
CasesperThou	0.0328	0.0271
prop_hpsa	0.0000	0.0000
nonwhite_prop	0.0000	0.0000
median_income	0.0000	0.0000
unempl_prop	0.0000	0.0000
poverty_prop	0.0000	0.0000
prop55older	0.4513	0.0000
HospNumber	-0.0001	0.0000
RatingBetter:CasesperThou	-0.0185	0.0000
RatingWorse:CasesperThou	0.0160	0.0000
RatingBetter:prop55older	-0.0587	0.0000
RatingWorse:prop55older	0.0000	0.0000
RatingBetter:HospNumber	0.0000	0.0000
RatingWorse:HospNumber	0.0000	0.0000

Table 2: Prediction performance of the LASSO models at two different L1 penalty strengths.

Model	Prediction MSE
lambda 1se	0.0479827
lambda min	0.0150106

The prediction MSE for the lambda 1se model, which had only one significant predictor, is 0.04798, while the prediction MSE for the lambda min model with seven predictors is 0.01501. The sole predictor in the first model was the heart disease case rate, and the difference in prediction error between the two models is a fairly large 68.72%. It is clear that these variables are lending predictive power to the model, but not enough to significantly outweigh a stronger L1 penalty imposed by LASSO.

In the lambda min model, we find that additional predictors of deaths due to heart disease include the median income, hospital rating, number of hospitals in the county at that rating

level, and proportion of residents 55 years old and older. This agrees with common sense: we can imagine that the overall rate of deaths due to heart disease in a given area is due to demographic effects on the health of the general population and economic effects impeding treatment.

We also find that the rating of a hospital on its own has no statistically significant effect on heart attack deaths until it interacts with the number of heart attack cases. In general, subgroups of hospitals rated "Worse" tended to have higher deaths, while deaths were expected to *decrease overall* for hospitals with "Better" ratings with cases remaining the same. LASSO demonstrates that, given these data, there is a trade-off between model simplicity and prediction power, and more realistic models tend to employ more demographic and hospital-level variables.

4.3 Method 3. Regression Trees

4.3.1 Random Forest

While there are nine unique predictors in the data, an exhaustive search using cross validation produced the optimal number of predictors and number of trees to use for the training data: 7 predictors and 398 trees respectively.

Table 3: Summary of cross-validated random forest results.

Metric	Value
Number of Predictors	7
Number of Trees	398
Prediction MSE	0.0299

The random forest model has a prediction MSE of 0.02992, which corresponds to a 37.64% decrease in prediction MSE with respect to the lambda 1se LASSO regression model. We find that 81.33% of the variation in deaths due to heart disease per 1000 individuals is explained by the combination of 7 variables, and it is again clear that hospital and demographic variables are lending predictive power to the model.

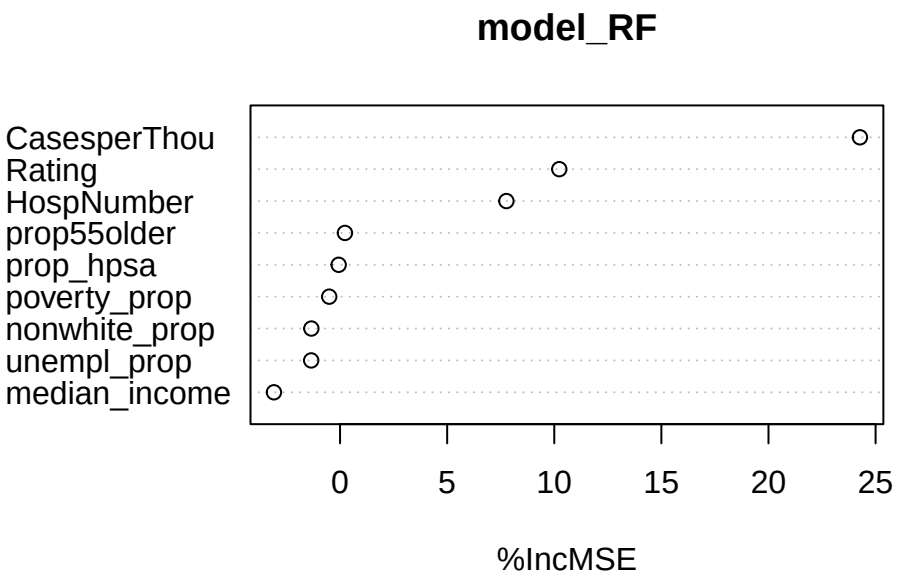


Figure 2: Random forest importance plot of all predictors.

We find that the dominant predictor, as expected, was the number of heart attack cases per thousand residents, but unlike LASSO, the random forest model routinely employs other predictors to keep the prediction error low. We find that hospital rating and amount are significantly used predictors in the random forest, and likely reflect the quality and accessibility of care in different counties. The effect of hospital rating and the number of hospitals far outweigh the effect of demographic factors, including the proportion of residents aged 55 years and older. In general, the random forest model performs with the second lowest prediction error of all regression models explored, and strikes a reasonable balance between bias and complexity in the number of predictors. We fail to find strong evidence of demographic factors, but these regression models illustrate that that hospital quality and number in counties in the state of California are associated with deaths due to heart disease when controlling for variation in the number of heart disease cases.

5 Ethical Analysis

Managing health-related data and models comes with a wide variety of serious ethical concerns. Understanding the risks of biased data or an inaccurate model is critical to safeguarding against these. As such, as we work on these models, we must keep in mind the privacy of patients,

the quality of our data, and its ability to accurately represent the demographics of the whole population, and we must be mindful of the variables that are included or excluded.

1) Managing data: Collecting medical data is a delicate challenge. First and foremost, health data is personal and sensitive. It must be handled with the utmost respect and caution as it contains sensitive and personal information. The US legal system, through The Health Insurance Portability and Accountability Act of 1996 (HIPAA), protects sensitive patient information. However, to study the health system and understand the weaknesses and strengths of the current hospital system, we need data. As such, as data scientists, we must be sure to handle data in a careful manner that ensures that individual and identifiable information is not connected with the data used to build the model. Protecting privacy creates challenges in collecting the data and ensuring that the data is representative. (NIH)

2) Biases and Representation: Medical data must be representative of the population it is measuring or we risk creating an inaccurate and harmful model. Inaccurate data can reinforce stereotypes leading to misdiagnosis and mistreatment. Medical data differs from other data sources because inaccuracies are often a matter of life and death. In this particular set of models, we need to be cautious about the demographics and our ability to generalize the results. For example, this data gives good insight into California's hospital system, however, California is not representative of the nation. California's median age (37.9) is the eleventh youngest in the nation and almost a year younger than the median age nationally (38.8). Further, California is more diverse than other states and has a larger population of immigrants (27%) compared to the rest of the country (12%) (PPIC). Further, the median income in California is more than $15,000 greater than the national median income (US Data). These are a couple of examples where California's data does not represent the greater US. It is important to note these disparities and study how they might impact the data and prevent us from generalizing our data to other hospital systems. (NIH)

3) Inclusion of Variables: In each model, we had to select which variables to include and which to exclude. For example, in the multiple linear regression models, `hispanic` was the only race variable that was significant in the full model and thus was included in the reduced model while others were not. This choice, even based on the data, is one that can impact the outcome. Transparency in which variables are included in the final models is critical to the reproducibility of the model and the preventing harmful biases.

Conversely, good data and strong models can serve as guidance to hospitals and be accountability measures. For example, the NIH has found a 10% increase in in-hospital mortality rates due to unsafe practices while increased staffing is correlated with a 12% decrease in deaths. Accurate data reporting can provide accountability for hospital practices. This gives community members insight into the medical system and the needed improvements in their locality.

Further, better public policy can stem from accurate reporting. For example, when considering our models, overlaying deaths and demographic information can highlight at-risk groups that need further support. We find in our LASSO analysis that patients over the age of 55 are at a higher risk of death. As such, policymakers know that they need to further focus on resources for this community. Continued monitoring of the death rate can demonstrate if policy changes yield a change in mortality rates. Good policy is based on good data and models.

6 Summary of Findings

On these data, the best regression model to given the predictors was the random forests model. It provides the lowest MSE value of 0.02992 while still explaining more than 80% of the variation in the model, and demonstrated a reasonable balance between bias and flexibility. To decrease prediction error, we found we had to sacrifice some of the power of the model: for example, the best multiple linear regression model explained over 90% of the variance, however, because we are handling medical data, accuracy is of the utmost importance. Further, all three models demonstrate the importance of including the number of cases of heart disease as a predictor of the number of deaths. After controlling for heart disease cases, the next more significant factor was hospital ranking. Demographic variables showed weak associations with heart disease deaths, which was surprising to the researchers and would be a good opportunity for further research. Understanding why demographic information did not make a difference, which was an assumption of the research team, would be pressing for future medical studies.

The classification models …

7 Future Directions

Future Directions Text

8 Appendix

8.1 Group Member Accomplishments

Roland Abi

Elise Buellesbach Found and proposed the original CA data set (`California Hospital Inpatient Mortality Rates and Quality Ratings`). Completed the linear regression models (both full and reduced). Wrote the executive summary and ethical analysis.

Amanda Concepcion

Spencer Grewe

Acquired CA demographic data (`data4`). Cleaned datasets 1, 3, and 4. Contributed toward tidying and joining datasets 1, 3, and 4 for the regression tasks. Carried out 627 tasks involving model selection for regression.

9 References