Predicting County-Level Heart Disease Mortality in the United States

Charlotte Abrams, Laura Cosgrove, Alyssa Vanderbeek

7 April 2019

Introduction

Heart disease remains one of the leading causes of death in adults in the US. Understanding risk factors for diseases of this kind is an important task in working towards reducing the number of lives lost. There is obvious benefit in doing this at an individual level, examining personal lifestyle, genetic profile, and family history. But there may be important environmental components that have predictive value in assessing risk for heart disease mortality. Identifying county-level environmental and demographic predictor for heart disease mortality could aid in effective targeting of national campaigns to increase education and reduce risk. We examine economic, health, and demographic data for thousands of counties across the US. This data is synthesized from several sources, including the USDA Economic Research Service, Bureau of Labor Statistics, US Census, Behavioral Risk Factor Surveillance System, the CDC, and others. Our goal in this project is to most effectively predict the county-level heart disease mortality rate per 100,000 persons. We build 6 predictive models (stepwise linear regression, Lasso, Ridge, PCR, GAM, and MARS), and compare them on their predictive capacity quantified by the root mean squared error (RMSE).

Methods

In our data pre-processing and exploratory analysis, we examined variable distributions (checking for skewness, missing data, etc). We eliminated two variables with over 90% missing data, and used K-nearest neighbors (KNN) to impute remaining missing values. The dataset was split into training and testing subsets at a parsing 2:1 ratio. There was one categorical variable for which individual factor levels had near-zero variance, but because these levels were part of a larger parameter, we did not remove them. The final dataset includes the outcome (heart disease mortality rate per 100k) and 28 predictors relating to economic, demographic, and health-related county charateristics.

We built predictive models on the training data using 6 methods: stepwise linear regression, Lasso, Ridge, PCR, GAM, and MARS. All model builds were performed using the `caret` package, and optimal model parameters were selected based on minimizing MSE (or, in the case of MARS, generalized cross-validation error) with 10-fold cross-validation, repeated five times. The optimal model was selected based on training RMSE, and models were again compared based on test RMSE. All R code used for model fitting is provided in the Appendix.

Results

Exploratory Data Analysis

The average county-level heart disease mortality per 100,000 residents is 279.37, with standard deviation 58.95. Though approximately normally distributed, the exact density of the outcome distribution differs slightly among levels of categorical variables [FIGURE].

In the category of health, statistics related to lifestyle characteristics (obesity, smoking, diabetes, low birthweight babies, excessive drinking, and physical inactivity) are highly intercorrelated. Less so are more environmental - or "acts of God" - factors: air particulate matter, homocides, motor vehicle crashes, and rates of dentists and doctors (though the last two are highly correlated). For demography, we found categorical

variables masquerading as separate predictors, leading to high intercorrelation within those categories: percentages of residents who are a given race accounts for 5 variables; age-related bins (less than 18, greater than 65) account for 2 variables; birth and death rate; and percentages of residents who complete a given level of education account for 4 variables. Given some of the subtleties in these distinctions, we condensed only race into "white" and "non-white". For economics, unsurprisingly, the percent of adults and the percent of children without health insurance are highly correlated, as well as percent civilian labor and unemployment rate.

Between categories, education-related variables are highly correlated with % civilian labor, % uninsured adults, and % physical inactivity. Predictors strongly associated with heart disease mortality are % physical inactivity, % diabetes, % adult obesity, education-related variables, % low birthweight, and overall death rate per 1,000. Many of the variables most strongly-associated with the outcome are associated with one another.

Predictive Models

Final models and RMSE

All models used the majority of all 28 variables (where categorical variables were split into dummy variables).

Table 1 presents the RMSE of the predicted heart disease mortality rate for all models on both the training and testing datasets. GAM outperforms all other models in both fields, though MARS has similar performce on predicting outcomes in the test data. All linear models have similar performance.

Coefficient Shrinkage: Lasso and Ridge

For better visualization, a `glmnet` model was fit and the lambda value selected by `caret` was plotted.

Investigation of MARS

The minimum generalized cross-validation error was achieved for a total of 25 features, including a two-way interaction term. Variable importance is assessed by tracking GCV for each predictor and accumulating the reduction in GCV when each predictor's feature is added to the model – the total reduction is used as the measure of variable importance. If a predictor was never used in any MARS basis function, it has an importance value of zero; 18 predictors were used in a MARS basis function.

MARS retained interactions between many sets of health-related variables, which one might expect given collinearity (and relatedness) between the within-category variable sets. Interaction terms were also present across categories.

Investigation of GAM

Compared to MARS and PCR, GAM gave more weight to the factor variable metropolitan status and metropolitan adjacency.

MARS, GAM, and PCR, had similar distributions of residuals.

Discussion

GAM was our best-performing model. With GAM, we were able to achieve a cross-validated RMSE of median 32.45 and bootstrapped interquartile range of (31.18, 33.34. GAM, and MARS, also performed well on the test data.
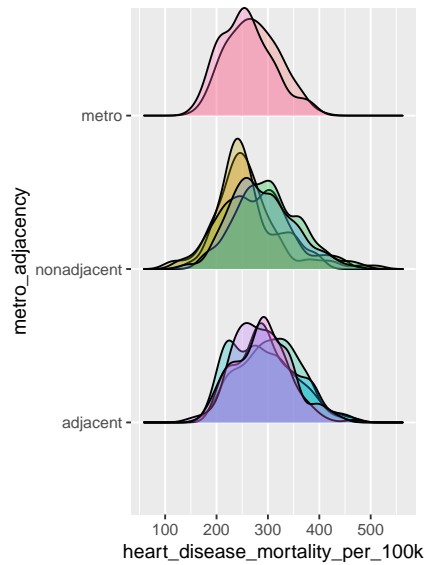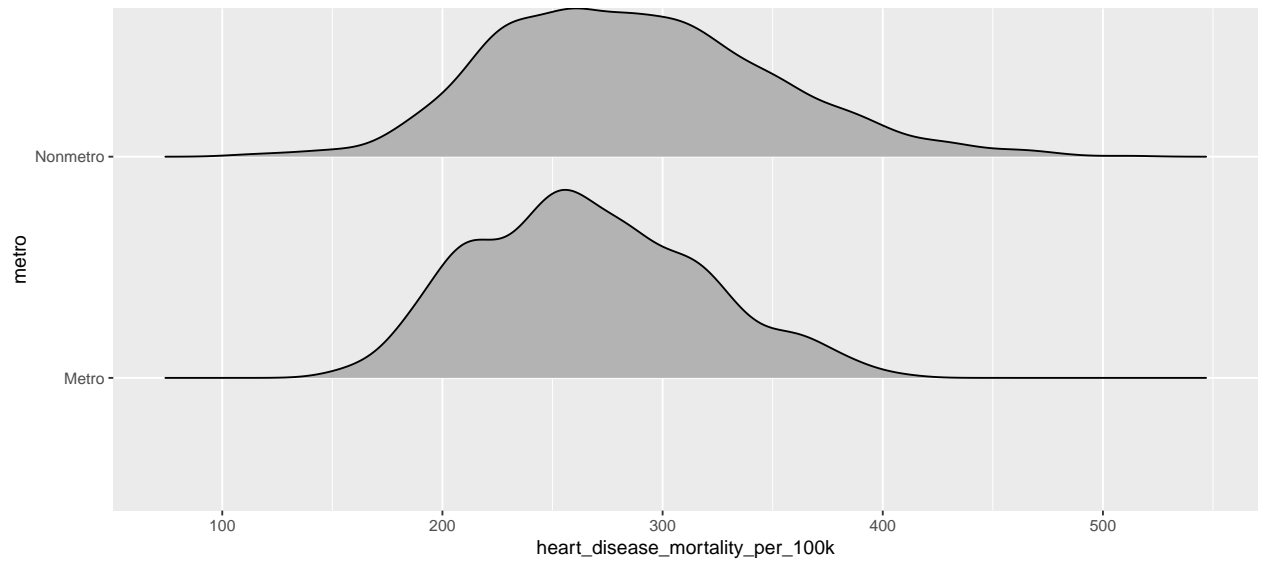
This modeling problem was perhaps not best addressed by fitting lasso and ridge. Neither of the methods shrunk many coefficients in such a way that completely removed them from the model and improved the RMSE. The best visualization of the effect of shrinkage can be seen in the ridge coefficient plot [FIGURE], where it can be seen that the vast majority of coefficients shrunk equally. Lasso's more stringent shrinking power did not improve RMSE for this problem. Had these methods been more appropriate in this context,

parameter selection might have been more obvious. As it is, as shown [IN FIGURE BLANK], the standard error of the mean cross-validated error is quite wide.

As described, GAM and MARS outperformed the linear models, and if the primary goal of the model is prediction, than we suggest using GAM. However, the linear models were close in performance to these more flexible models, and if simplicity and interpretability are major concerns, one of these may be preferred.

Variable importance was relatively consistent through each model, as measured by loss-perturbed RMSE in the `caret` cross-validation and shown in Appendix [FIGURE BLANK]. However, compared to other models, GAM gave much more weight to the value of the geography-related factor variables Metro and Metro Adjacency.

Tables and Figures

Table 1: Highly Correlated Predictors Between Categories

variable	with	correlation
demo___pct_adults_less_than_a_high_school_diploma	econ___pct_civilian_labor	-0.63
demo___pct_adults_less_than_a_high_school_diploma	econ___pct_uninsured_adults	0.67
demo___pct_adults_bachelors_or_higher	health___pct_adult_obesity	-0.63
demo___pct_adults_bachelors_or_higher	health___pct_physical_inacticity	-0.67
demo___pct_adults_with_high_school_diploma	health___pct_physical_inacticity	0.60

Table 2: Correlation with Outcome

variable	correlation
heart_disease_mortality_per_100k	1.00
health___pct_physical_inacticity	0.73
health___pct_diabetes	0.66
health___pct_adult_obesity	0.64
demo___pct_adults_bachelors_or_higher	-0.59
demo___pct_adults_less_than_a_high_school_diploma	0.58
health___pct_low_birthweight	0.53
demo___death_rate_per_1k	0.52
econ___pct_civilian_labor	-0.52
health___pct_adult_smoking	0.52
health___motor_vehicle_crash_deaths_per_100k	0.51
demo___pct_adults_with_high_school_diploma	0.48
econ___pct_uninsured_adults	0.40
econ___pct_unemployment	0.39
demo___pct_adults_with_some_college	-0.33
health___pop_per_dentist	0.33
health___pop_per_primary_care_physician	0.22
health___air_pollution_particulate_matter	0.17
demo___birth_rate_per_1k	0.15
demo___pct_nonwhite	0.12
demo___pct_below_18_years_of_age	0.10
demo___pct_female	0.08
econ___pct_uninsured_children	0.03
demo___pct_aged_65_years_and_older	0.02

Resampled Training RMSE
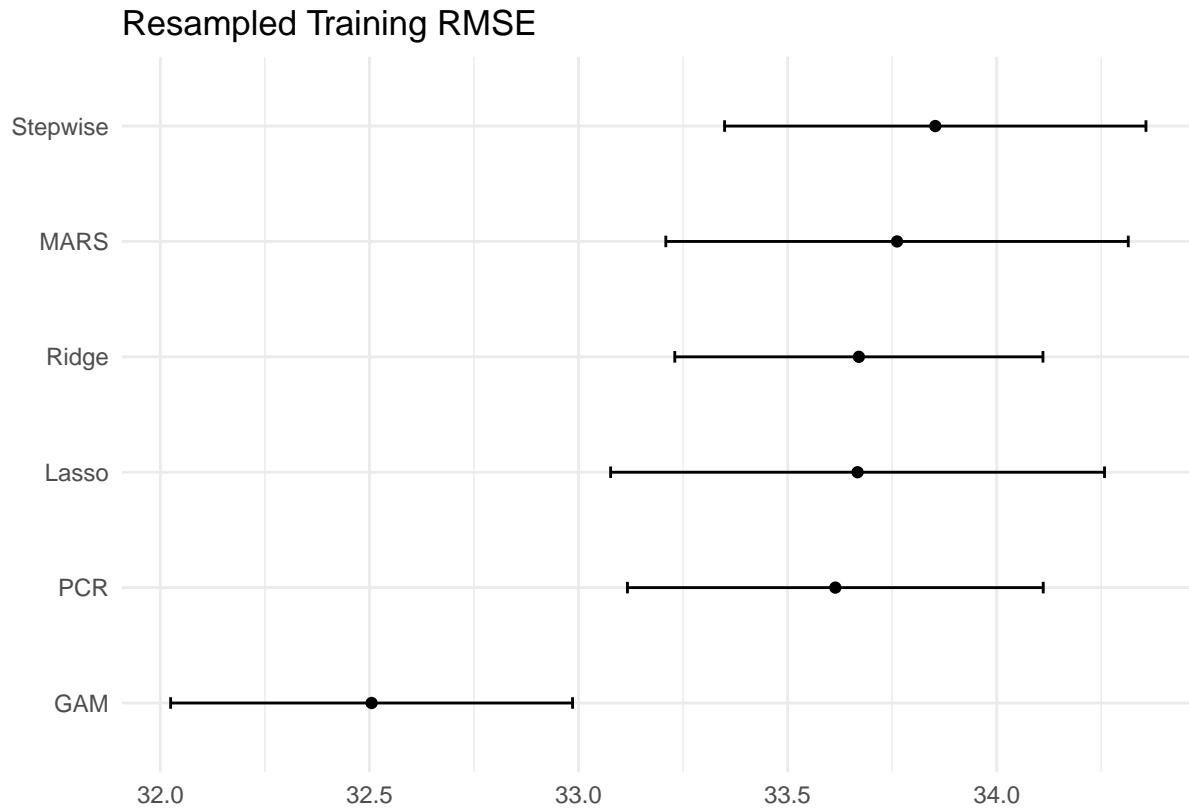
Table 3: Predictive RMSE on training and test data

	Train RMSE	Test RMSE
Stepwise	33.85305	32.90754
Ridge	33.67035	33.04168
Lasso	33.66725	32.89532
PCR	33.61412	32.86776
GAM	32.50513	30.67289
MARS	33.76163	30.83814

```
## [1] 48  1
```

Ridge
CV−selected lambda in blue

```
## integer(0)
```


Lasso
CV−selected lambda in blue

```
## integer(0)
```
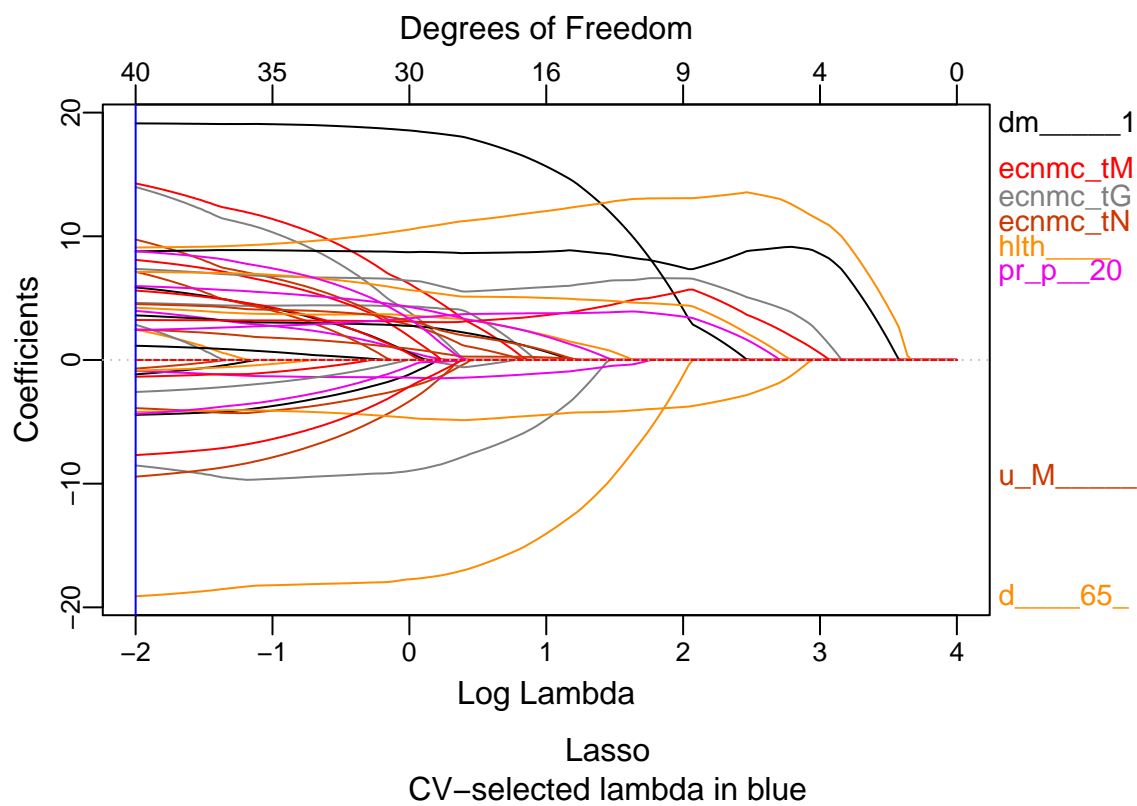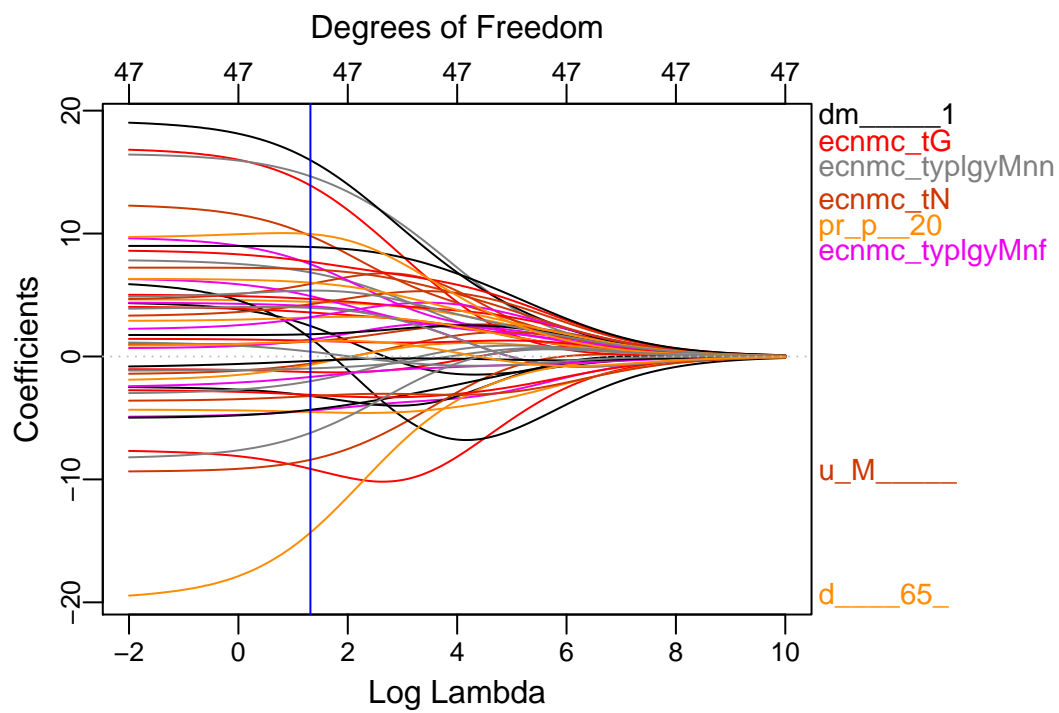
Table 4: Top Absolute-Values Coefficients, Ridge

row	column	value
(Intercept)	1	268.695291
demo___death_rate_per_1k	1	15.898754
economic_typologyMining	1	14.612043
economic_typologyGovernment	1	13.841795
pure_populationmore_than_20k	1	9.947095
economic_typologyNonspecialized	1	9.753701
health___pct_physical_inacticity	1	8.899230
health___pct_diabetes	1	7.686107
economic_typologyManufacturing	1	7.460263
health___pct_low_birthweight	1	7.079183

Table 5: Top Absolute-Valued Coefficients, Lasso

row	column	value
(Intercept)	1	269.100366
demo___death_rate_per_1k	1	19.125559
economic_typologyMining	1	14.248826
economic_typologyGovernment	1	13.960475
economic_typologyNonspecialized	1	9.716805
health___pct_physical_inacticity	1	9.097034
pure_populationmore_than_20k	1	8.797225
health___pct_diabetes	1	8.781068
urban_influenceLarge_in_a_metro_area_with_at_least_1_million_residents_or_more	1	8.028903
demo___pct_adults_less_than_a_high_school_diploma	1	7.367846

Ridge Parameter Selection

Lambda chosen by caret
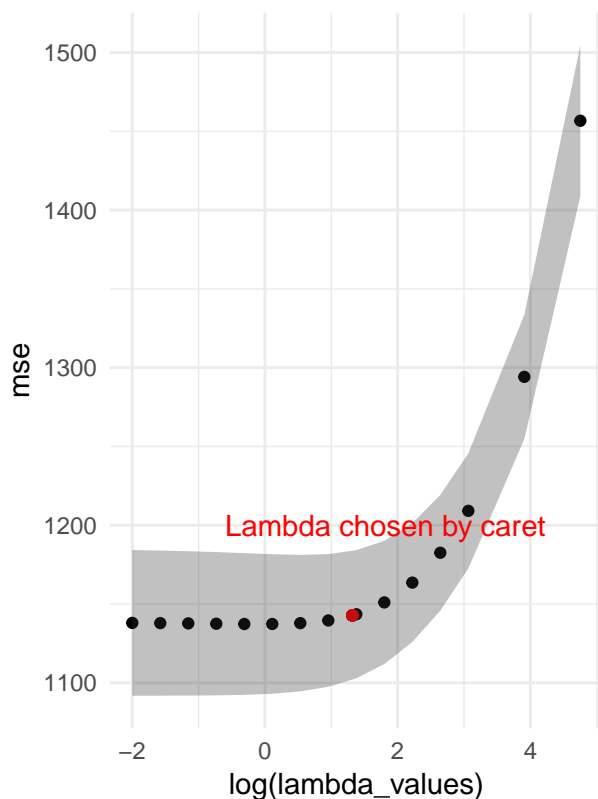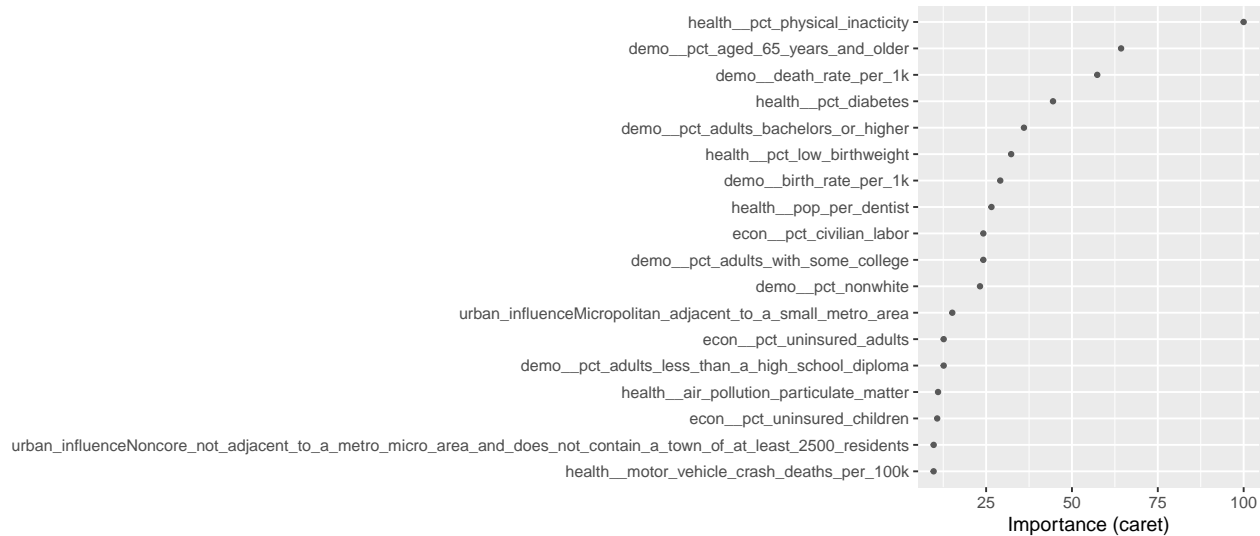
Lasso Parameter Selection
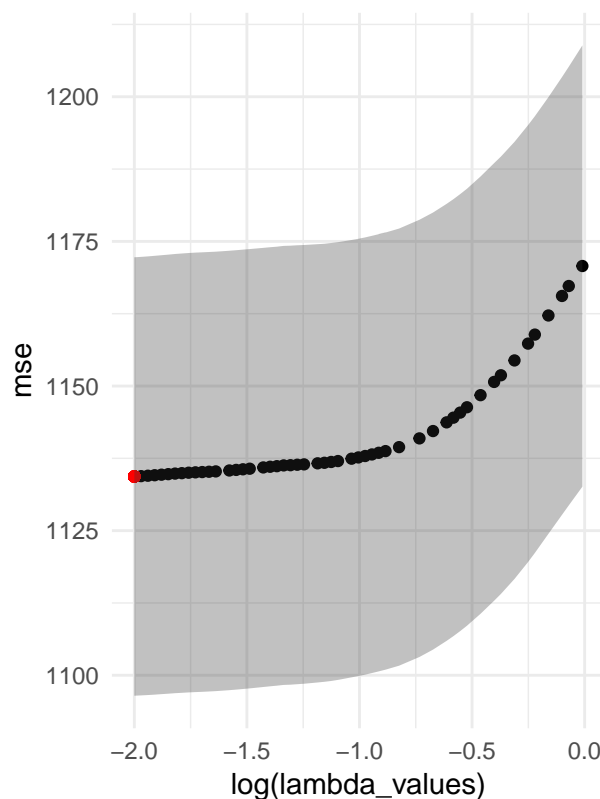
GCV

```
## # A tibble: 25 x 2
##    names                                                        x
##    <chr>                                                    <dbl>
## 1 (Intercept)                                               312.
## 2 h(pct_physical_inacticity- -1.58282)                      20.4
## 3 h(pct_aged_65_years_and_older- -0.609182) * h(pct_physical_ina~   -4.48
## 4 h(pct_adults_bachelors_or_higher- -0.090806)             -7.09
## 5 h(-0.090806-pct_adults_bachelors_or_higher)               16.6
## 6 h(pct_physical_inacticity- -1.58282) * h(0.103008-pct_nonwhite)   -8.39
```

```
##  7 h(pct_aged_65_years_and_older- -1.78344) * h(0.122171-pct_diab~    -7.59
##  8 h(-1.78344-pct_aged_65_years_and_older) * h(0.122171-pct_diabe~   -30.2
##  9 h(pct_diabetes-0.122171) * h(-0.965113-pop_per_dentist)           341.
## 10 h(pct_low_birthweight- -1.11807)                                   19.9
## # ... with 15 more rows
```
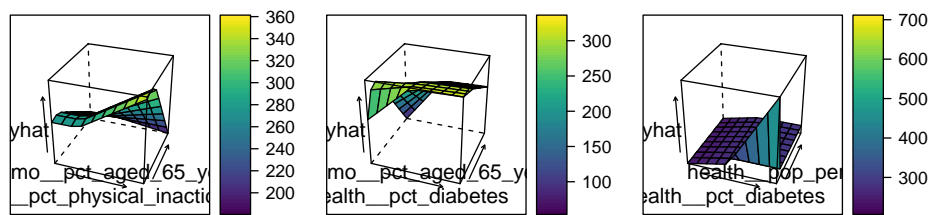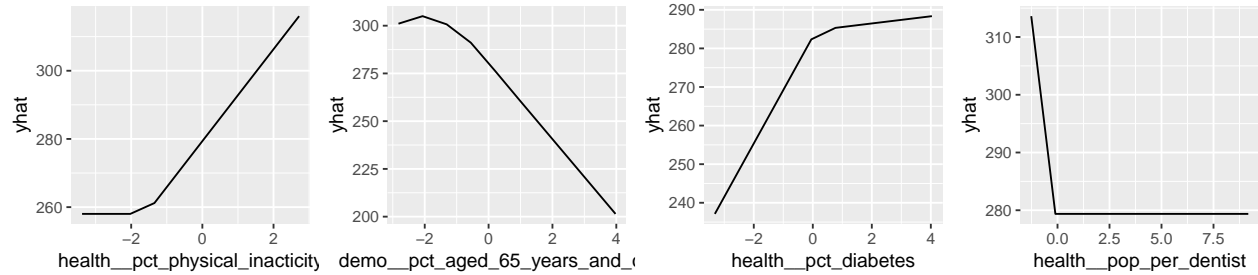

Distribution of |residual|

Appendix
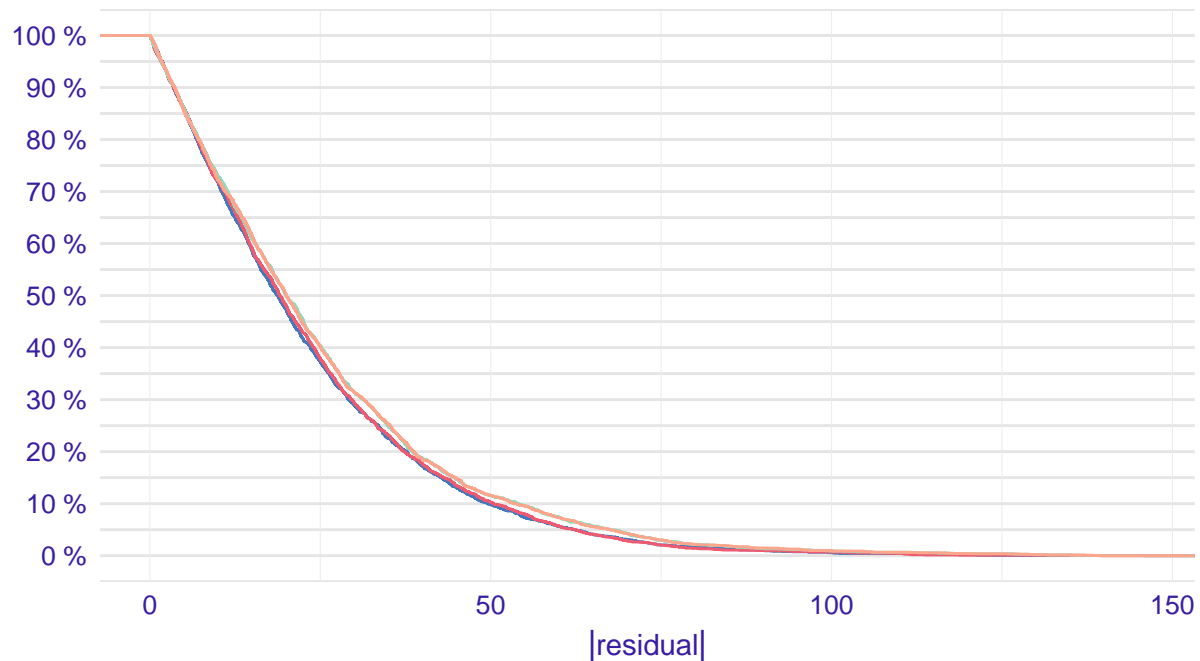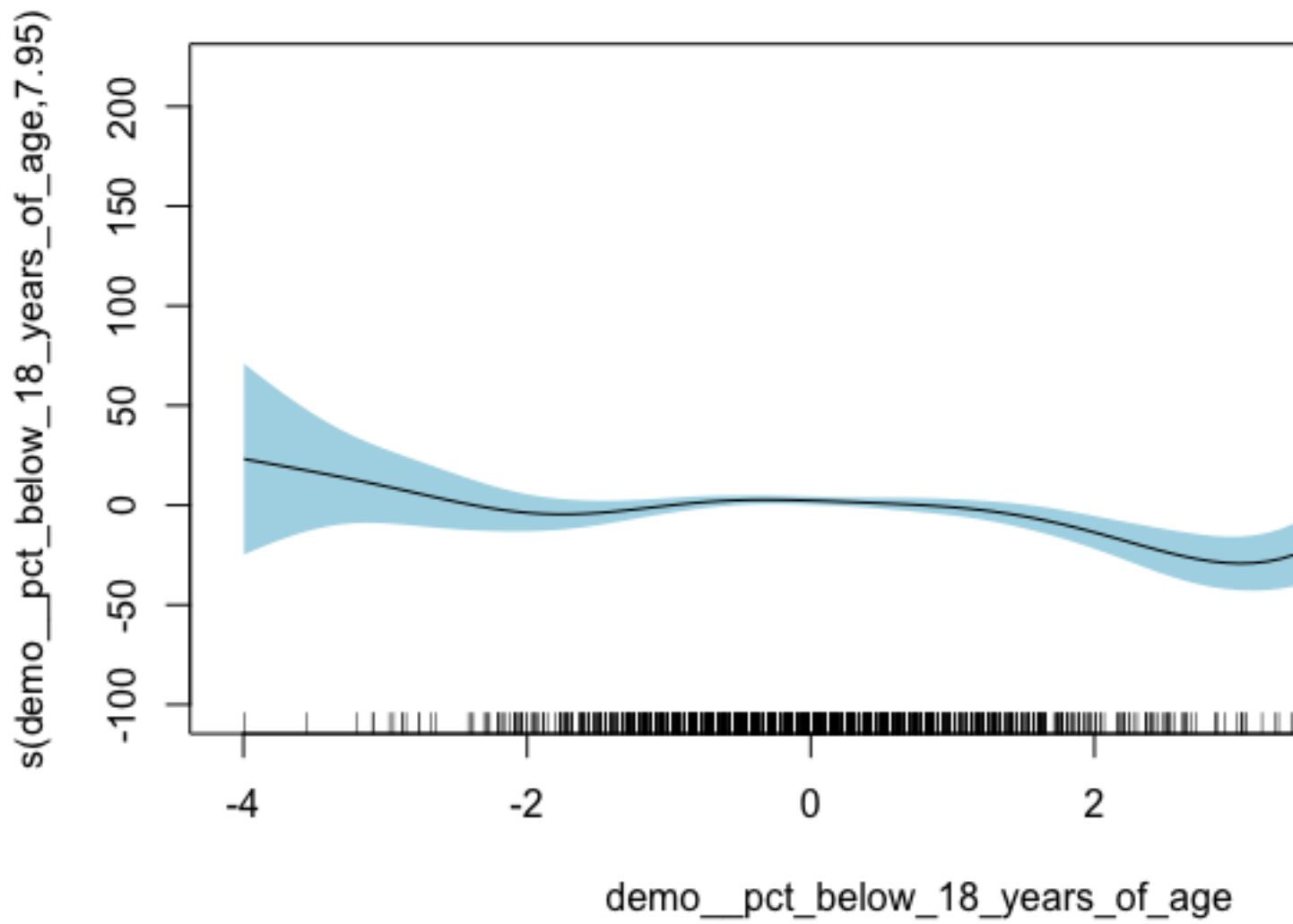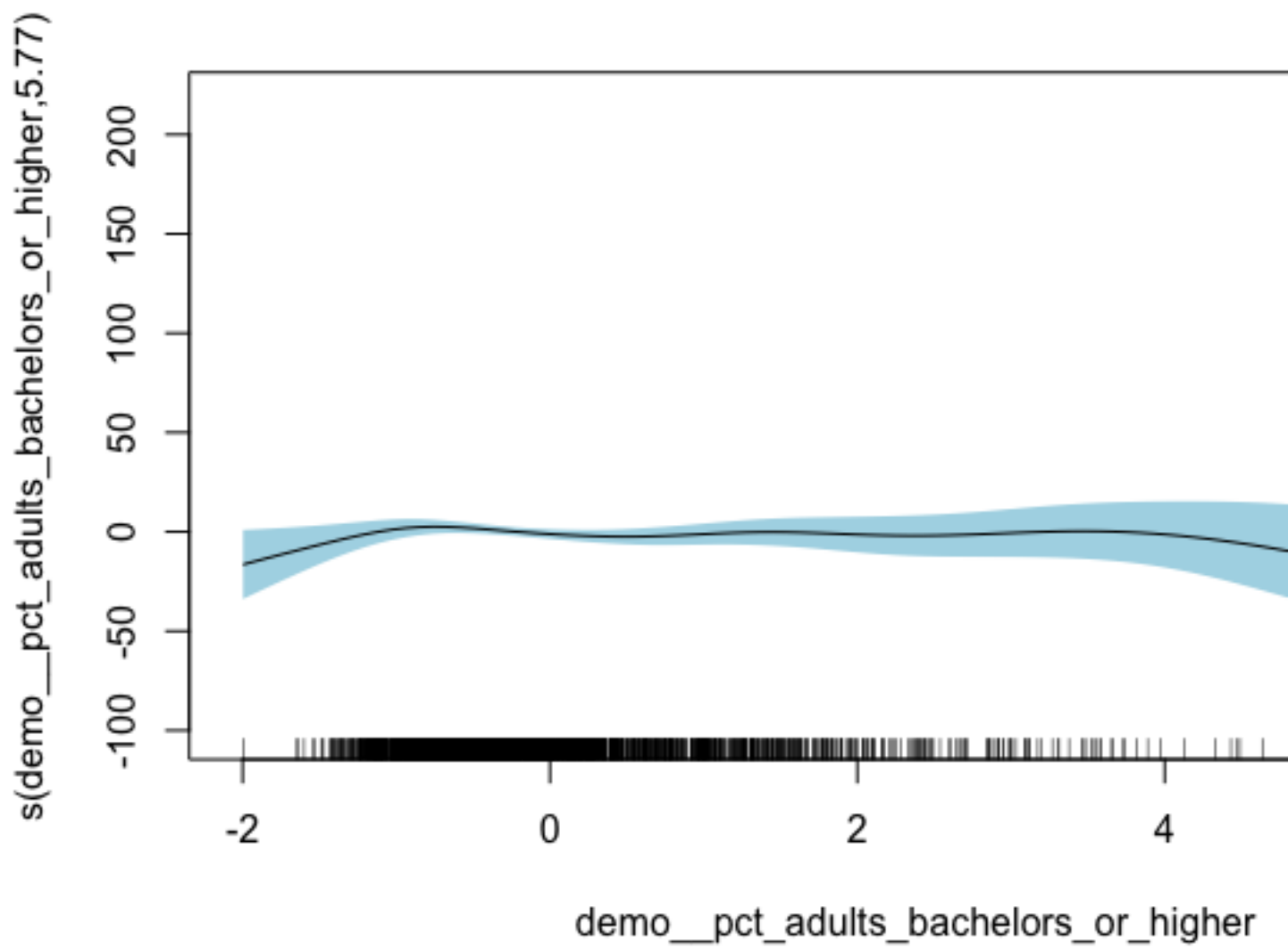
Data prep

```r
## Data import
predictors <- read_csv("./data/Training_values.csv")
response <- read_csv("./data/Training_labels.csv")



## Manipulation
data <- response %>%
  full_join(predictors, by = "row_id") %>%
  separate(col = area__rucc, into = c('metro', 'population'), sep = ' - ') %>%
  rename(urban_influence = area__urban_influence,
         economic_typology = econ__economic_typology) %>%
  mutate(pure_population = fct_collapse(as.factor(population),
                                "more_than_1mil" = "Counties in metro areas of 1 million popula
                                "250k_to_1mil" = "Counties in metro areas of 250,000 to 1 milli
                                "less_than_250k" = "Counties in metro areas of fewer than 250,00
                                "more_than_20k" = c("Urban population of 20,000 or more, adjacen
                                                    "Urban population of 20,000 or more, not
                                "2500_to_20k" = c("Urban population of 2,500 to 19,999, adjacen
                                                  "Urban population of 2,500 to 19,999, not a
                                "less_than_2500" = c("Completely rural or less than 2,500 urban
                                                     "Completely rural or less than 2,500 popu
         economic_typology = as.factor(recode(economic_typology,
                                "Nonspecialized" = "Nonspecialized",
                                "Manufacturing-dependent" = "Manufacturing",
                                "Farm-dependent" = "Farming",
                                "Federal/State government-dependent" = "Government",
                                "Mining-dependent" = "Mining",
                                "Recreation" = "Recreation")),
         metro = factor(metro,
                        levels = c("Metro", "Nonmetro")),
         urban_influence = str_replace_all(urban_influence, " |/|-", "_"), # replace problematic charact
         urban_influence = str_replace_all(urban_influence, ",", ""), # replace problematic characters
         demo__pct_nonwhite = demo__pct_hispanic + demo__pct_asian + demo__pct_american_indian_or_alaskar
         urban_influence = fct_rev(urban_influence),
         metro_adjacency = fct_collapse(population,
                                metro = c("Counties in metro areas of 1 million population or mor
                                          "Counties in metro areas of 250,000 to 1 million popula
                                          "Counties in metro areas of fewer than 250,000 populati
                                adjacent = c("Urban population of 20,000 or more, adjacent to a r
                                             "Urban population of 2,500 to 19,999, adjacent to a
                                             "Completely rural or less than 2,500 urban populatio
                                nonadjacent = c("Urban population of 20,000 or more, not adjacent
                                                "Urban population of 2,500 to 19,999, not adjacer
                                                "Completely rural or less than 2,500 urban popula
  dplyr::select(-demo__pct_hispanic,
                -demo__pct_asian,
                -demo__pct_american_indian_or_alaskan_native,
                -demo__pct_non_hispanic_african_american,
                -demo__pct_non_hispanic_white,
```

```
                      -health__homicides_per_100k, # >90% missing
                      -health__pct_excessive_drinking, # >90% missing
                      -yr,
                      -population,
                      -row_id)
```

Training and testing data split. Imputation on missing data.

```
## training/test data
set.seed(1)
train_ind <- sample(seq_len(nrow(data)), size = 2/3*nrow(data)) # select rows in 2:1 ratio

train <- data[train_ind, ] # training dataset
test <- data[-train_ind, ] # testing dataset

# Imputation for missing values with caret, based on training data
training_preproc = caret::preProcess(train[,-1],
                                     method = "knnImpute", # automatically centers and scales data
                                     pcaComp = 10,
                                     na.remove = TRUE,
                                     k = 5,
                                     knnSummary = mean,
                                     outcome = NULL,
                                     fudge = .2,
                                     numUnique = 3,
                                     verbose = TRUE)

# Impute training imputation on both training and testing datasets
train_imputed = predict(training_preproc, train)
test_imputed = predict(training_preproc, test)

#save files to Rdata: was not saving the factor structure in read from csv
saveRDS(train_imputed, file = './data/train_imputed.Rdata')
saveRDS(train_imputed, file = './data/test_imputed.Rdata')
```

Linear Models

Set up `caret` training control. We will use this for all models.

```
set.seed(100)
ctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 5)
```

Stepwise regression:

```
set.seed(2)
step.fit = caret::train(x, y,
                        method = 'glmStepAIC',
                        metric = "RMSE",
                        trControl = ctrl)
saveRDS(step.fit, "lm_step_imputed.rds")
```

Lasso:

```
set.seed(2)
lasso_fit <- caret::train(x, y,
                          method = "glmnet",
                          metric = "RMSE",
```

```
                             tuneGrid = expand.grid(alpha = 1,
                                                    lambda = exp(seq(-2, 4, length = 200))),
                             trControl = ctrl)
plot(lasso_fit, xTrans = function(x) log(x)) #in correct range

saveRDS(lasso_fit, "lasso_imputed.rds")
```

Ridge:

```
set.seed(100)

ridge_fit <- caret::train(x, y,
                    method = "glmnet",
                    tuneGrid = expand.grid(alpha = 0,
                                           lambda = exp(seq(-2, 10, length = 200))),
                    trControl = ctrl1)


plot(ridge_fit, xTrans = function(x) log(x)) #in correct range

best_lambda_ridge = ridge_fit$bestTune$lambda

saveRDS(ridge_fit, "ridge.rds")
```

PCR:

```
set.seed(2)
pcr_fit <- caret::train(x, y,
                        method = "pcr",
                        trControl = ctrl,
                        metric = "RMSE",
                        tuneLength = 200)
saveRDS(pcr_fit, "pcr_imputed.rds")
```

Non-linear models

GAM:

```
set.seed(2)
gam_fit <- caret::train(x, y,
                        method = "gam",
                        metric = 'RMSE',
                        tuneGrid = data.frame(method = "GCV.Cp", select = c(TRUE, FALSE)),
                        trControl = ctrl)
summary(gam_fit)
saveRDS(gam_fit, "gam_fit_imputed.rds")
```

MARS:

```
mars_grid <- expand.grid(degree = 1:3, # degree: 1 vs 2 vs 3, no interaction vs. interaction;
                         nprune = 10:40) # nprune is number of coef

set.seed(2)

mars_fit <- caret::train(x, y,
                        method = "earth",
                        tuneGrid = mars_grid,
```

```
                      trControl = ctrl)

saveRDS(mars_fit, "mars.rds")

#based on initial results, choose parsimonious version
mars_grid_refined <- expand.grid(degree = 2, nprune = 25:40)

mars_fit_refined <- caret::train(x, y,
                                 method = "earth",
                                 tuneGrid = mars_grid_refined,
                                 trControl = ctrl)
saveRDS(mars_fit_refined, "mars2.rds")
```