Homework 2 Solutions

Contents

Load some handy packages:

```
suppressPackageStartupMessages(library(dplyr))
suppressPackageStartupMessages(library(ggplot2))
```

Review + Residual Analysis

```
#load the fivethirtyeight library
suppressPackageStartupMessages(library(fivethirtyeight))

#load data
data(hate_crimes)
```

Exercise 1

```
hate_crimes <- hate_crimes %>%
    mutate(preHate=avg_hatecrimes_per_100k_fbi/365,
       postHate=hate_crimes_per_100k_splc/10,
       trump50=share_vote_trump>0.50) %>%
    mutate(changeHate = postHate-preHate)

mean(hate_crimes$preHate, na.rm=TRUE)
```

```
## [1] 0.006486611
```

```r
mean(hate_crimes$postHate, na.rm=TRUE)
```

```
## [1] 0.0304093
```

```r
table(hate_crimes$trump50)
```

```
##
## FALSE  TRUE
##    27    24
```

```r
mean(hate_crimes$changeHate, na.rm=TRUE)
```

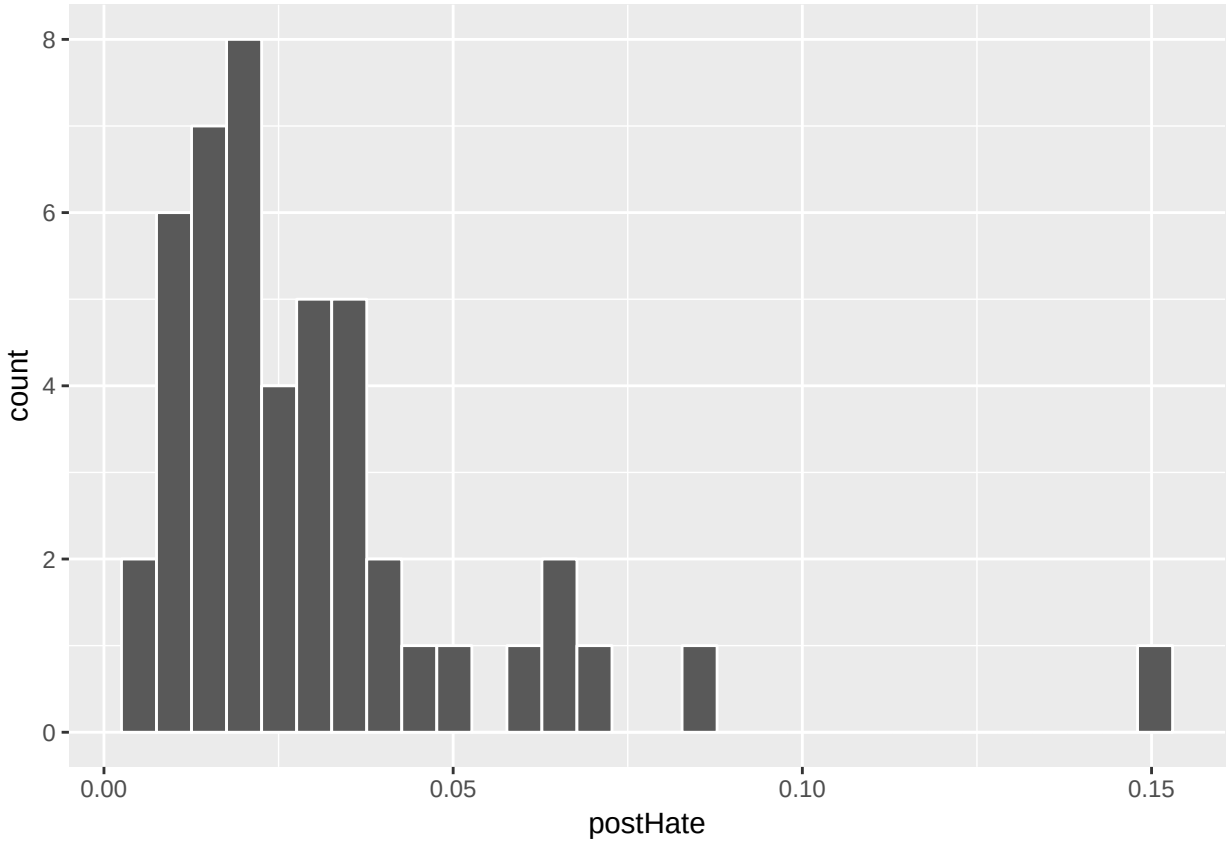
```
## [1] 0.02399304
```

Exercise 2

```r
#a
ggplot(hate_crimes, aes(x=postHate)) +
    geom_histogram(color="white")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 4 rows containing non-finite values (stat_bin).
```




```r
#b
hate_crimes %>% filter(postHate > 0.1)
```

```
## # A tibble: 1 x 16
```

```
##                  state median_house_inc share_unemp_seas share_pop_metro
##                  <chr>            <int>            <dbl>           <dbl>
## 1 District of Columbia            68277            0.067               1
## # ... with 12 more variables: share_pop_hs <dbl>, share_non_citizen <dbl>,
## #   share_white_poverty <dbl>, gini_index <dbl>, share_non_white <dbl>,
## #   share_vote_trump <dbl>, hate_crimes_per_100k_splc <dbl>,
## #   avg_hatecrimes_per_100k_fbi <dbl>, preHate <dbl>, postHate <dbl>,
## #   trump50 <lgl>, changeHate <dbl>
```

```
#c
hate_sub <- hate_crimes %>% filter(postHate < 0.1)
dim(hate_sub)
```

```
## [1] 46 16
```

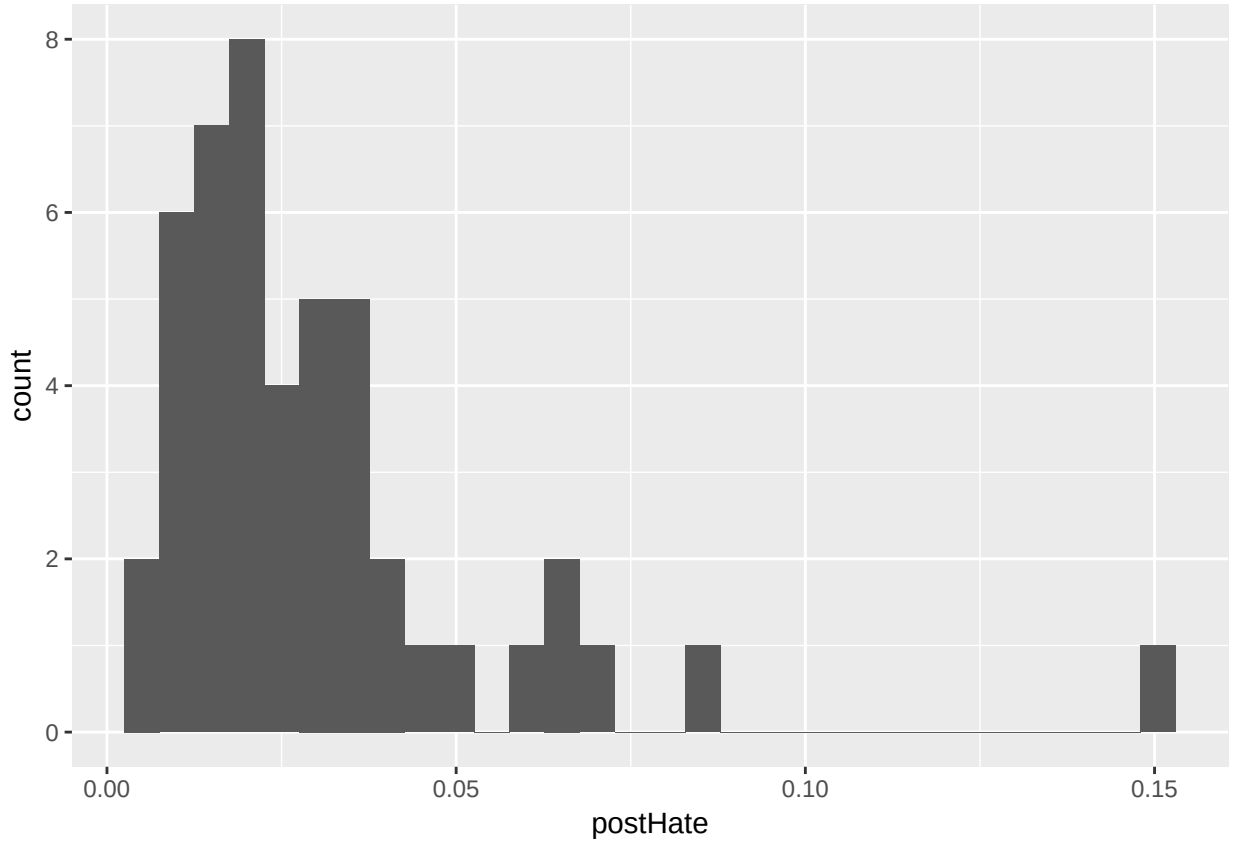
```
mean(hate_sub$postHate, na.rm=TRUE)
```

```
## [1] 0.02776102
```

```
#d
ggplot(hate_crimes, aes(x=postHate)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 4 rows containing non-finite values (stat_bin).
```




```
hate_crimes %>% filter(postHate > 0.1)
```

```
## # A tibble: 1 x 16
##                  state median_house_inc share_unemp_seas share_pop_metro
```

```
##                   <chr>            <int>            <dbl>           <dbl>
## 1 District of Columbia            68277            0.067               1
## # ... with 12 more variables: share_pop_hs <dbl>, share_non_citizen <dbl>,
## #   share_white_poverty <dbl>, gini_index <dbl>, share_non_white <dbl>,
## #   share_vote_trump <dbl>, hate_crimes_per_100k_splc <dbl>,
## #   avg_hatecrimes_per_100k_fbi <dbl>, preHate <dbl>, postHate <dbl>,
## #   trump50 <lgl>, changeHate <dbl>
```

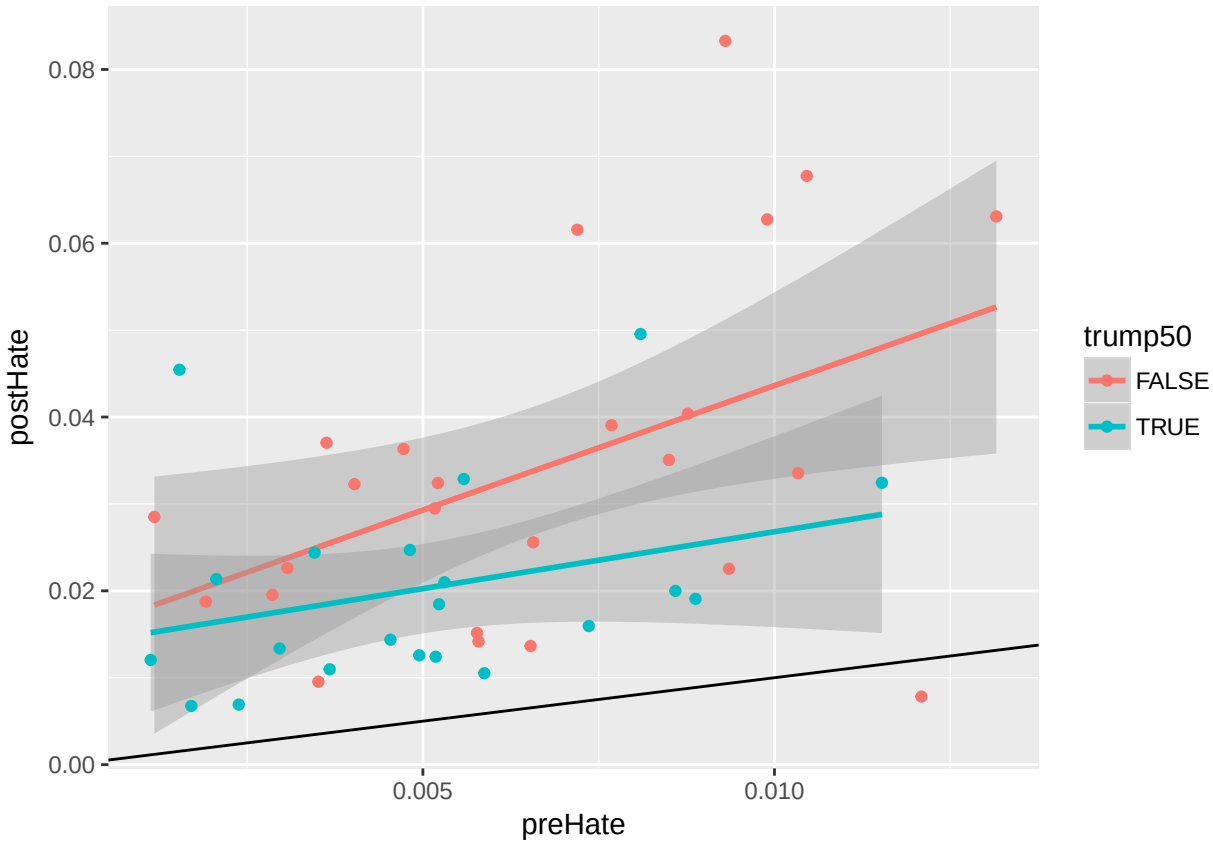
```
hate_sub <- hate_crimes %>% filter(postHate < 0.1)
```

Exercise 3

```
#a
mod <- lm(postHate ~ preHate*trump50, hate_sub)
summary(mod)
```

```
##
## Call:
## lm(formula = postHate ~ preHate * trump50, data = hate_sub)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.041779 -0.007955 -0.001900  0.006064  0.041668
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.014970   0.007117   2.104  0.04144 *
## preHate              2.864926   0.964621   2.970  0.00491 **
## trump50TRUE         -0.001260   0.010024  -0.126  0.90057
## preHate:trump50TRUE -1.556091   1.577140  -0.987  0.32946
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01522 on 42 degrees of freedom
## Multiple R-squared:  0.3153, Adjusted R-squared:  0.2664
## F-statistic: 6.448 on 3 and 42 DF,  p-value: 0.001086
```

```
#b
ggplot(hate_sub, aes(y=postHate, x=preHate, color=trump50)) +
    geom_smooth(method="lm") +
    geom_point() +
    geom_abline(intercept=0, slope=1)
```



c:

postHate = 0.014970 + 2.864926preHate - 0.001260trump50TRUE - 1.556091preHate*trump50TRUE

- 0.014970: This is the average post-election hate crime rate in the theoretical state with 0 hate crimes pre-election and where Trump received <50% of the vote.
- 2.864926: In states where Trump received <50% of the vote, every extra pre-election hate crime (per 100000) corresponds to an extra 2.86 hate crimes (per 100000) post-election.
- -0.001260: The intercept is lower in states where Trump received >50% of the vote.
- -1.556091: The increase in post-election hate crimes with pre-election hate crimes is lower in states where Trump received >50% of the vote.

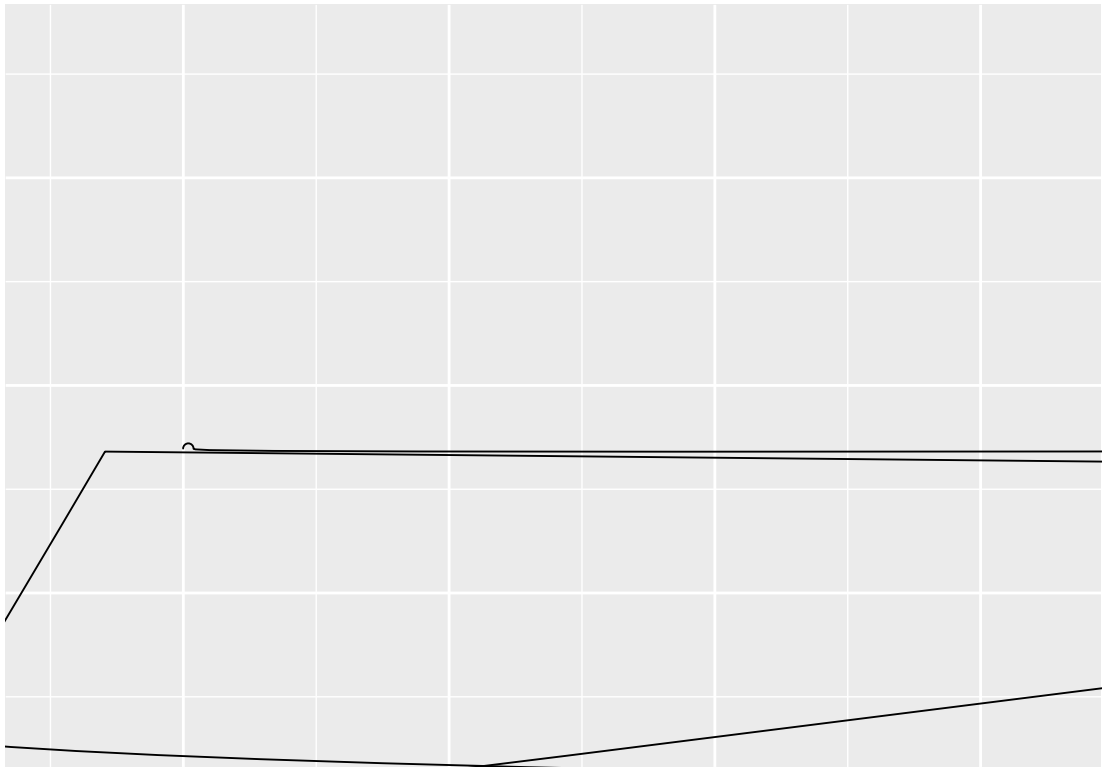
d:

the relationship between hate crime rates post- and pre-election vary among states where Trump receive 50+% of the vote and states where he didn't

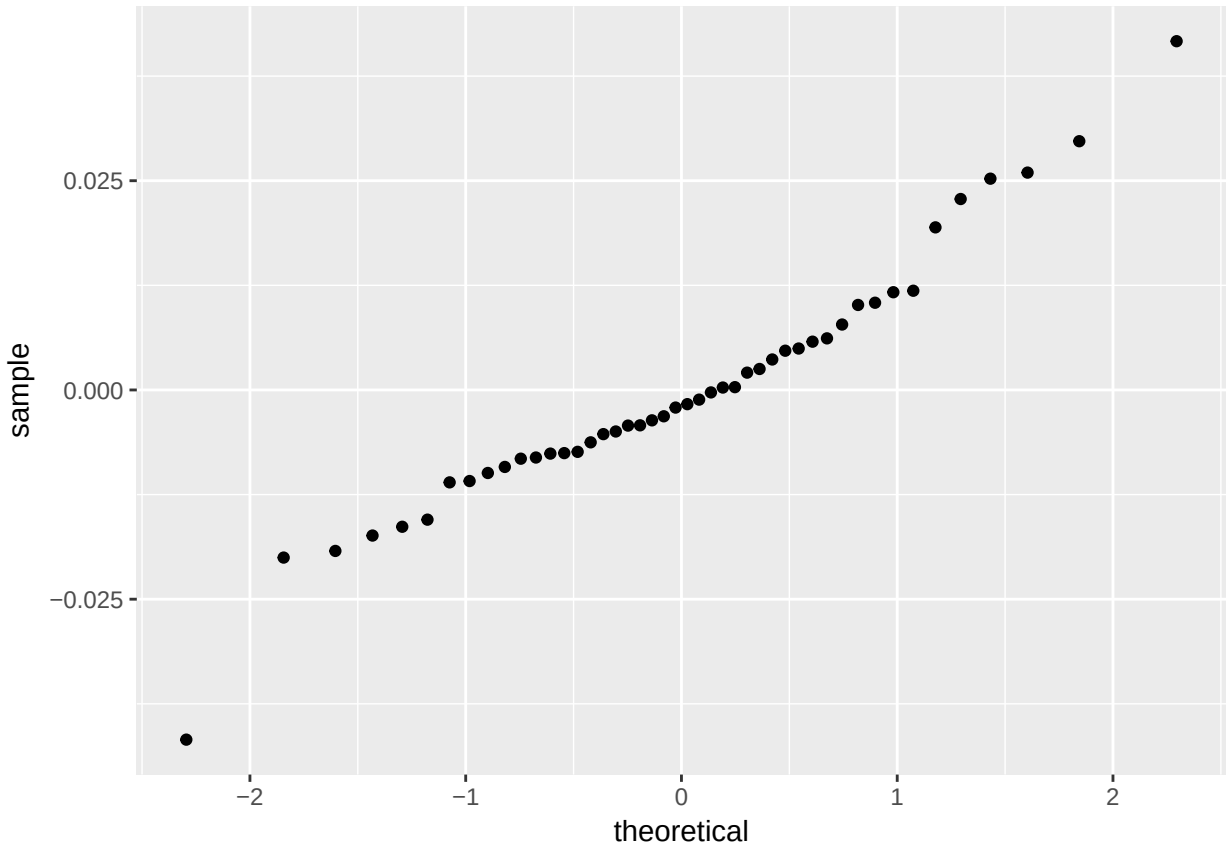
Exercise 4

```
#a
#Perhaps.  There might be some weak geographical trends in hate crimes
#that cross state lines, though it's probably safe to treat the states
#as independent in this analysis.

#b
#combine the raw responses, model predictions, and model residuals
```

```
ModResults <- data.frame(observed=hate_sub$postHate, predicted=mod$fitted.values,
        residual=mod$residuals)
#a plot of residuals versus the predictions
#the points are smattered above and below 0 with no obvious patterns
#there's a small suggestion of heteroskedasticity -
#more error among states with higher hate crime rates
ggplot(ModResults, aes(y=residual, x=predicted)) +
    geom_point() +
    geom_hline(yintercept=0)
```



```
#a Q-Q plot of the residuals
#the residuals look fairly normal
ggplot(ModResults, aes(sample=residual)) +
    geom_qq()
```



R^2

Exercise 5

model	structure	R^2
1	changeHate ~ share_non_white	0.0616
2	changeHate ~ median_house_inc	0.0693
3	changeHate ~ share_non_white + median_house_inc	0.1233
4	changeHate ~ share_vote_trump	0.1645
5	changeHate ~ trump50	0.1371
6	changeHate ~ share_vote_trump + trump50	0.1675

```
suppressPackageStartupMessages(library(mosaic))
rsquared(lm(changeHate ~ share_non_white, data=hate_sub))
```

```
## [1] 0.06160705
```

```
rsquared(lm(changeHate ~ median_house_inc, data=hate_sub))
```

```
## [1] 0.06938072
```

```
rsquared(lm(changeHate ~ share_non_white + median_house_inc, data=hate_sub))
```

```
## [1] 0.1233249
```

```
rsquared(lm(changeHate ~ share_vote_trump, data=hate_sub))
```

```
## [1] 0.1644774
```

```
rsquared(lm(changeHate ~ trump50, data=hate_sub))
```

```
## [1] 0.1370963
```

```
rsquared(lm(changeHate ~ share_vote_trump + trump50, data=hate_sub))
```

```
## [1] 0.1675282
```

Exercise 6

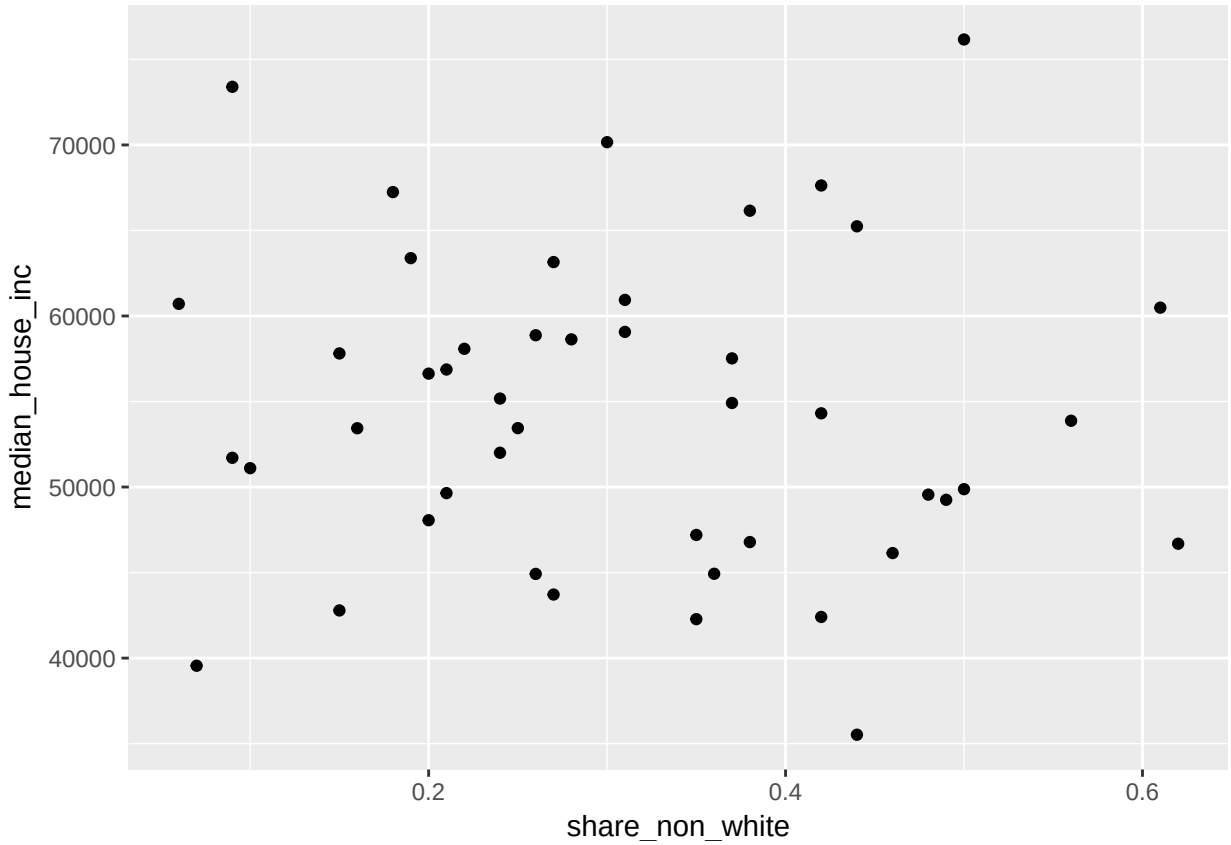
a:

R^2 increases with new predictors

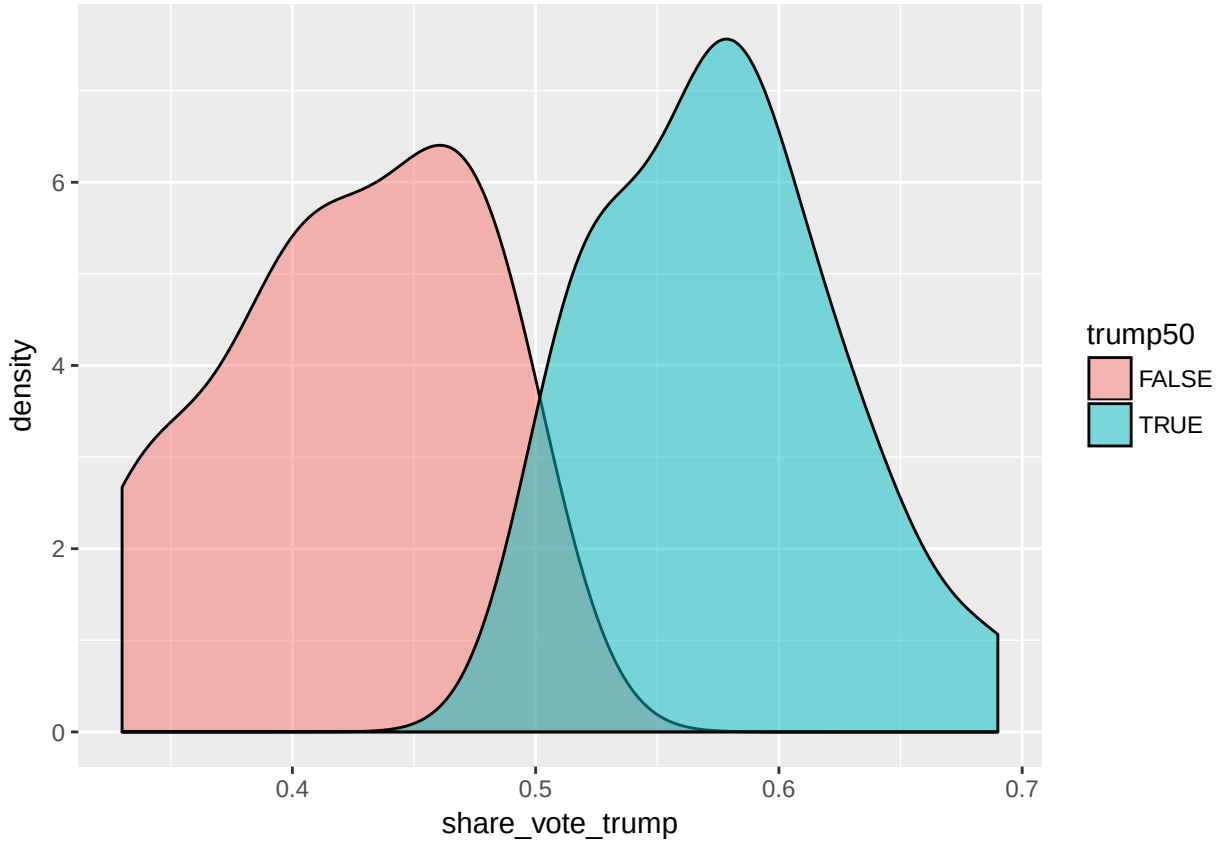
b

Unlike share_non_white & median_house_inc, predictors share_vote_trump & trump50 are strongly related themselves. Thus they contain similar information - including them both in the model won't increase the quality by much. (see part c for supporting evidence).

```
#c
ggplot(hate_sub, aes(x=share_non_white, y=median_house_inc)) +
    geom_point()
```




```
ggplot(hate_sub, aes(x=share_vote_trump, fill=trump50)) +
    geom_density(alpha=0.5)
```



Exercise 7

Everything becomes more di□cult (interpretations, predictions, visualization, communicability), with little gain in R^2.

Cross Validation

```
#Load the data:
suppressPackageStartupMessages(library(DAAG))
data(socsupport)
```

Exercise 8

```
#Step 1: define train and test sets
#the train set includes all but the 1st case of socsupport
data_train <- socsupport[-1,]
#the test set includes only the 1st case of socsupport
data_test <- socsupport[1,]

#Step 2: fit a model using the train set
train_mod <- lm(BDI ~ psisat, data=data_train)
```

```r
#Step 3: use train_mod to make a prediction for the test set
train_pred <- makeFun(train_mod)
test_predictions <- train_pred(psisat = data_test$psisat)


#Step 4: calculate the mean squared error for the test set
test_residuals <- data_test$BDI - test_predictions
MSE <- mean(test_residuals^2)


#check results
dim(data_train)
```

```
## [1] 94 20
```

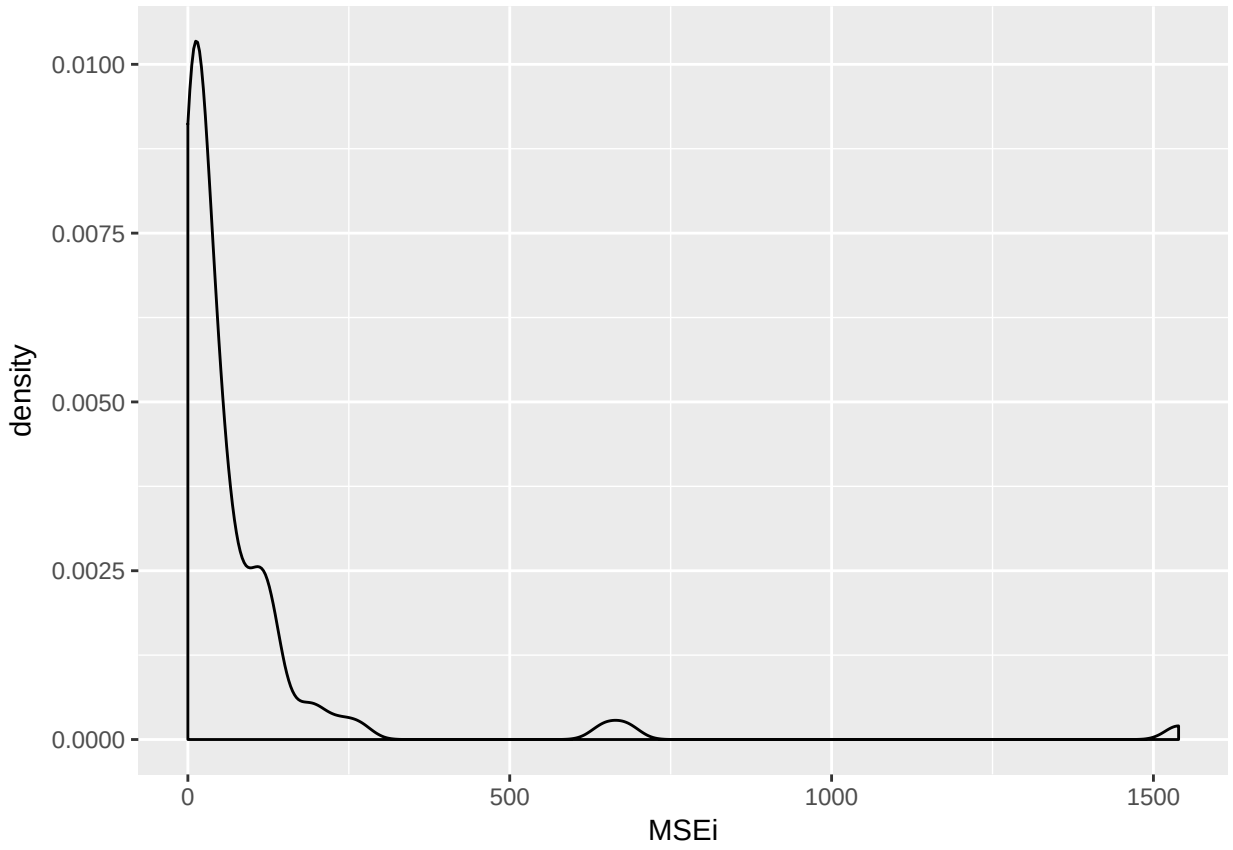
```r
dim(data_test)
```

```
## [1]  1 20
```

```r
MSE
```

```
## [1] 18.3571
```

```r
#b
#initialize the for loop: set up a vector in which to store the MSEi
MSEi <- rep(0,95)

#start the for loop
for(i in 1:95){
    #Step 1: define train and test sets
    data_train <- socsupport[-i,]
    data_test  <- socsupport[i,]

    #Step 2: fit a model using the train set
    train_mod <- lm(BDI ~ psisat, data=data_train)

    #Step 3: use train_mod to make a prediction for the test set case i
    train_pred <- makeFun(train_mod)
    test_predictions <- train_pred(psisat = data_test$psisat)

    #Step 4: calculate the squared error for the test set case i & store in MSEi
    test_residuals <- data_test$BDI - test_predictions
    MSEi[i] <- mean(test_residuals^2)
}


#c
#outliers are unusual cases that, when taken out as testing data,
#aren't predictable based on the training data
MSEdata <- data.frame(MSEi)
ggplot(MSEdata, aes(x=MSEi)) +
    geom_density()
```




```
#d
mean(MSEi)
```

```
## [1] 75.70166
```

```
#e
#The CV error is slightly larger than the MSE calculated from BDImod
#(which uses the same data for training and testing).
#Thus we originally overestimated the quality of predictions
#calculated from this model
```

Exercise 9

```
suppressPackageStartupMessages(library(boot))

#fit the model using the glm() not lm() function
mod <- glm(BDI ~ psisat, data=socsupport, family="gaussian")

#perform n-fold cross validation using cv.glm()
cv.err <- cv.glm(socsupport, mod, K=95)

#report the CV error (value on the left)
cv.err$delta
```

```
## [1] 75.70166 75.68084
```

Exercise 10

a:

Leave-one-out requires more computation (looping n times instead of 10). Further, there's a lot more overlap, thus correlation, among training sets in leave-one-out. This correlation leads to increased variability/error in the cross validation estimate. Read this for more detail: https://stats.stackexchange.com/questions/154830/10-fold-cross-validation-vs-leave-one-out-cross-validation

b:

Using k=2 only utilizes half the data to train models thus will likely lead to overestimates of the prediction error. We're also more likely to get biased splits in the training/testing data.

```
#c
set.seed(50)

cv10.err <- cv.glm(socsupport, mod, K=10)
cv10.err$delta
```

```
## [1] 76.97777 76.69912
```

```
cv2.err <- cv.glm(socsupport, mod, K=2)
cv2.err$delta
```

```
## [1] 83.53292 79.61649
```

```
#d
#k=2 is more pessimistic
```

Overfitting

Exercise 11

```
#a
#R^2 = 0.1012
polymod <- lm(changeHate ~ poly(median_house_inc, degree=3), hate_sub)
summary(polymod)
```

```
##
## Call:
## lm(formula = changeHate ~ poly(median_house_inc, degree = 3),
##     data = hate_sub)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.032097 -0.009619 -0.002811  0.006247  0.047482
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         0.021858   0.002392   9.138 1.54e-11
## poly(median_house_inc, degree = 3)1  0.029210   0.016223   1.801    0.079
## poly(median_house_inc, degree = 3)2 -0.018649   0.016223  -1.150    0.257
## poly(median_house_inc, degree = 3)3 -0.006580   0.016223  -0.406    0.687
##
```

```
## (Intercept)                          ***
## poly(median_house_inc, degree = 3)1 .
## poly(median_house_inc, degree = 3)2
## poly(median_house_inc, degree = 3)3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01622 on 42 degrees of freedom
## Multiple R-squared:  0.1012, Adjusted R-squared:  0.03698
## F-statistic: 1.576 on 3 and 42 DF,  p-value: 0.2094
```

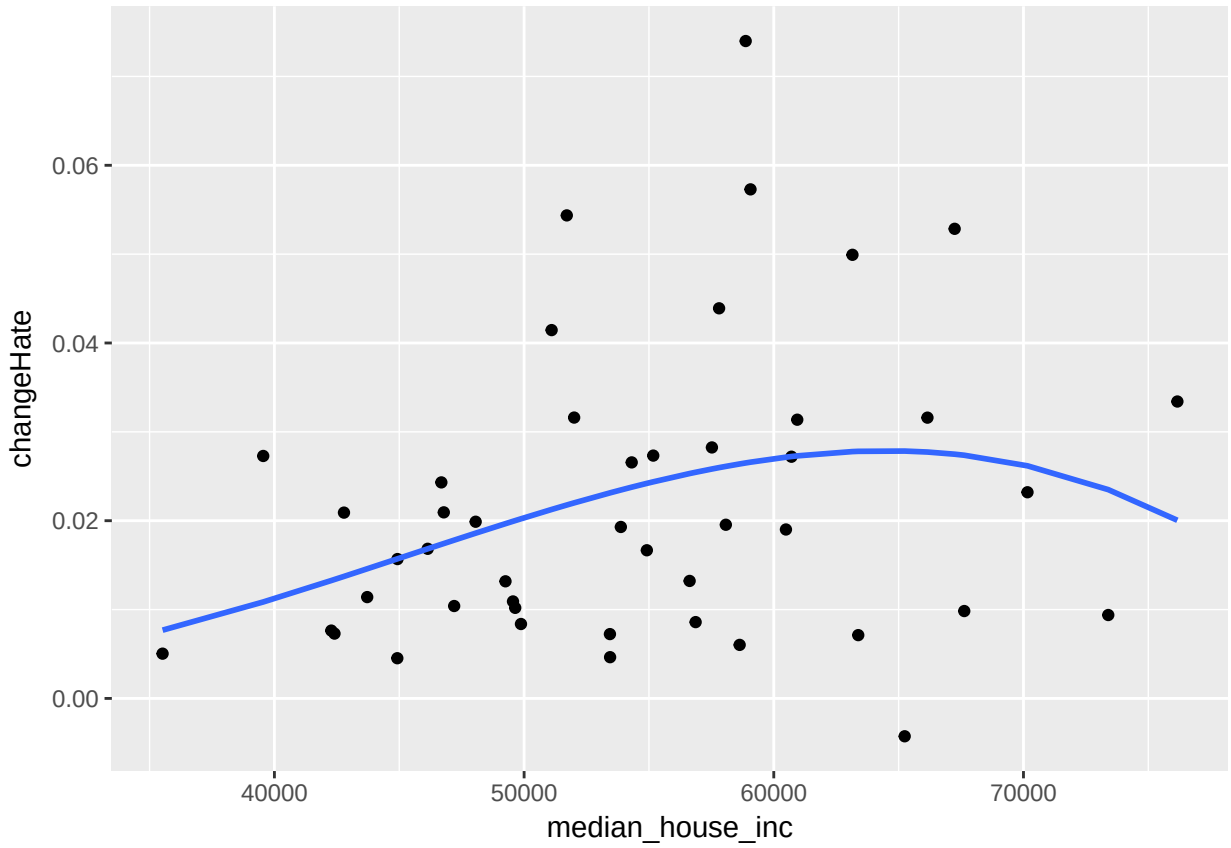
```
#b
set.seed(2000)
mod <- glm(changeHate ~ poly(median_house_inc, degree=3), hate_sub, family="gaussian")
cv.err <- cv.glm(hate_sub, mod, K=10)
```

```
## Warning in cv.glm(hate_sub, mod, K = 10): 'K' has been set to 9.000000
```

```
cv.err$delta
```

```
## [1] 0.0003094298 0.0003048410
```

```
#c
#for each state's median_house_inc, store the model predicted changeHate
polymodresults <- data.frame(median_house_inc=hate_sub$median_house_inc,
                        prediction=polymod$fitted.values)

#order the cases by median_house_inc
polymodresults <- arrange(polymodresults, median_house_inc)

#plot model and cases
ggplot(hate_sub, aes(x=median_house_inc, y=changeHate)) +
    geom_point() +
    geom_smooth(data=polymodresults, aes(x=median_house_inc, y=prediction), stat="identity")
```



Exercise 12

```
#a
#R^2 = 0.2921
polymod <- lm(changeHate ~ poly(median_house_inc, degree=16), hate_sub)
summary(polymod)
```

```
##
## Call:
## lm(formula = changeHate ~ poly(median_house_inc, degree = 16),
##     data = hate_sub)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.031114 -0.008174 -0.000041  0.007564  0.035276
##
## Coefficients:
##                                      Estimate Std. Error t value
## (Intercept)                          0.021858   0.002555   8.556
## poly(median_house_inc, degree = 16)1   0.029210   0.017327   1.686
## poly(median_house_inc, degree = 16)2  -0.018649   0.017327  -1.076
## poly(median_house_inc, degree = 16)3  -0.006580   0.017327  -0.380
## poly(median_house_inc, degree = 16)4   0.011176   0.017327   0.645
## poly(median_house_inc, degree = 16)5   0.018597   0.017327   1.073
## poly(median_house_inc, degree = 16)6   0.000248   0.017327   0.014
```

```
## poly(median_house_inc, degree = 16)7   0.010596   0.017327   0.612
## poly(median_house_inc, degree = 16)8  -0.004871   0.017327  -0.281
## poly(median_house_inc, degree = 16)9   0.019569   0.017327   1.129
## poly(median_house_inc, degree = 16)10  0.009402   0.017327   0.543
## poly(median_house_inc, degree = 16)11 -0.007900   0.017327  -0.456
## poly(median_house_inc, degree = 16)12 -0.026720   0.017327  -1.542
## poly(median_house_inc, degree = 16)13 -0.001523   0.017327  -0.088
## poly(median_house_inc, degree = 16)14  0.018815   0.017327   1.086
## poly(median_house_inc, degree = 16)15 -0.001137   0.017327  -0.066
## poly(median_house_inc, degree = 16)16 -0.011625   0.017327  -0.671
##                                      Pr(>|t|)
## (Intercept)                             2e-09 ***
## poly(median_house_inc, degree = 16)1     0.103
## poly(median_house_inc, degree = 16)2     0.291
## poly(median_house_inc, degree = 16)3     0.707
## poly(median_house_inc, degree = 16)4     0.524
## poly(median_house_inc, degree = 16)5     0.292
## poly(median_house_inc, degree = 16)6     0.989
## poly(median_house_inc, degree = 16)7     0.546
## poly(median_house_inc, degree = 16)8     0.781
## poly(median_house_inc, degree = 16)9     0.268
## poly(median_house_inc, degree = 16)10    0.592
## poly(median_house_inc, degree = 16)11    0.652
## poly(median_house_inc, degree = 16)12    0.134
## poly(median_house_inc, degree = 16)13    0.931
## poly(median_house_inc, degree = 16)14    0.286
## poly(median_house_inc, degree = 16)15    0.948
## poly(median_house_inc, degree = 16)16    0.508
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01733 on 29 degrees of freedom
## Multiple R-squared:  0.2921, Adjusted R-squared:  -0.09854
## F-statistic: 0.7477 on 16 and 29 DF,  p-value: 0.7257
```

```
#b
set.seed(2000)
mod <- glm(changeHate ~ poly(median_house_inc, degree=16), hate_sub, family="gaussian")
cv.err <- cv.glm(hate_sub, mod, K=10)
```

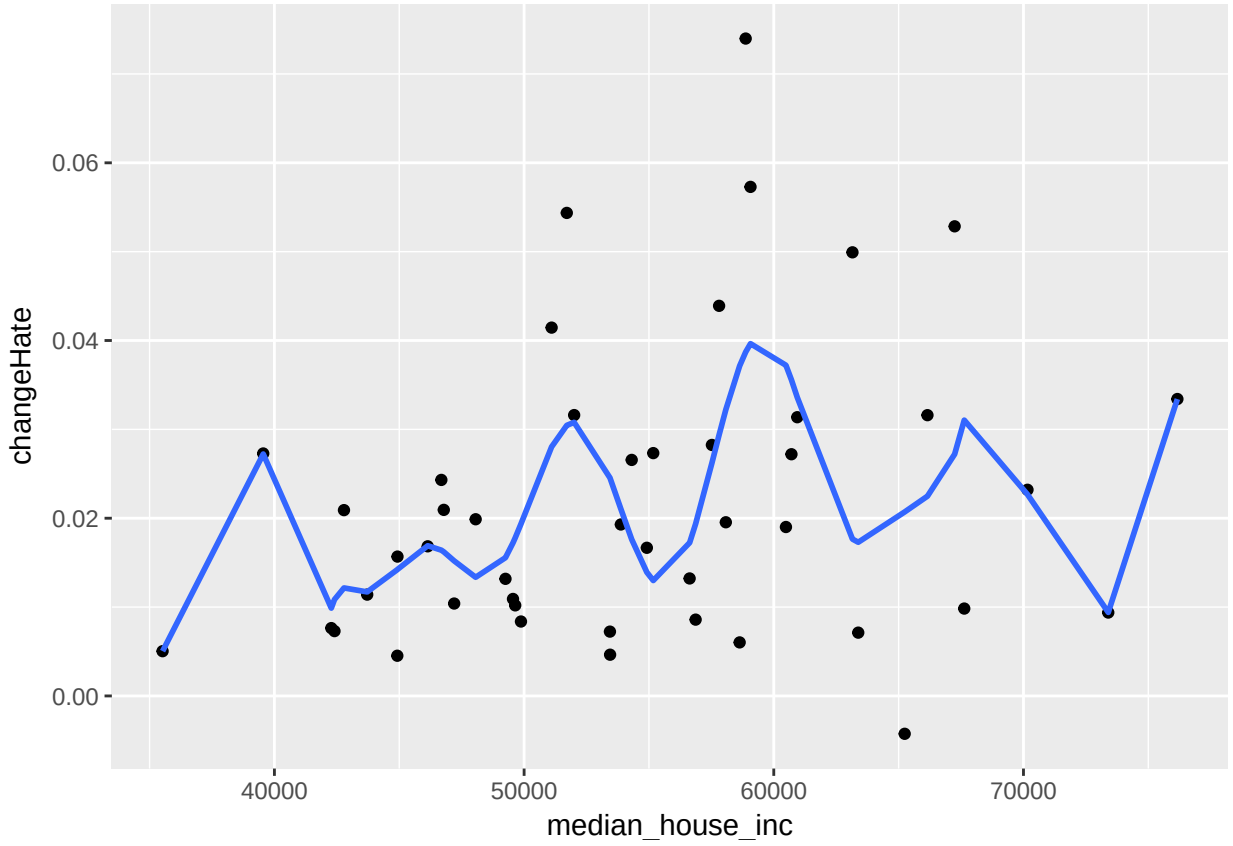
```
## Warning in cv.glm(hate_sub, mod, K = 10): 'K' has been set to 9.000000
```

```
cv.err$delta
```

```
## [1] 246272.2 219498.3
```

```
#c
#for each state's median_house_inc, store the model predicted changeHate
polymodresults <- data.frame(median_house_inc=hate_sub$median_house_inc, prediction=polymod$fitted.valu

#order the cases by median_house_inc
polymodresults <- arrange(polymodresults, median_house_inc)

#plot model and cases
ggplot(hate_sub, aes(x=median_house_inc, y=changeHate)) +
    geom_point() +
```

```
    geom_smooth(data=polymodresults, aes(x=median_house_inc, y=prediction), stat="identity")
```



Exercise 13

The greater the number of predictors, the greater the risk of overfitting the model to our particular sample of data. That is, the greater the risk of our model losing the general trend, hence, the model's generalizability to the greater population.
