PS 04

borui sun

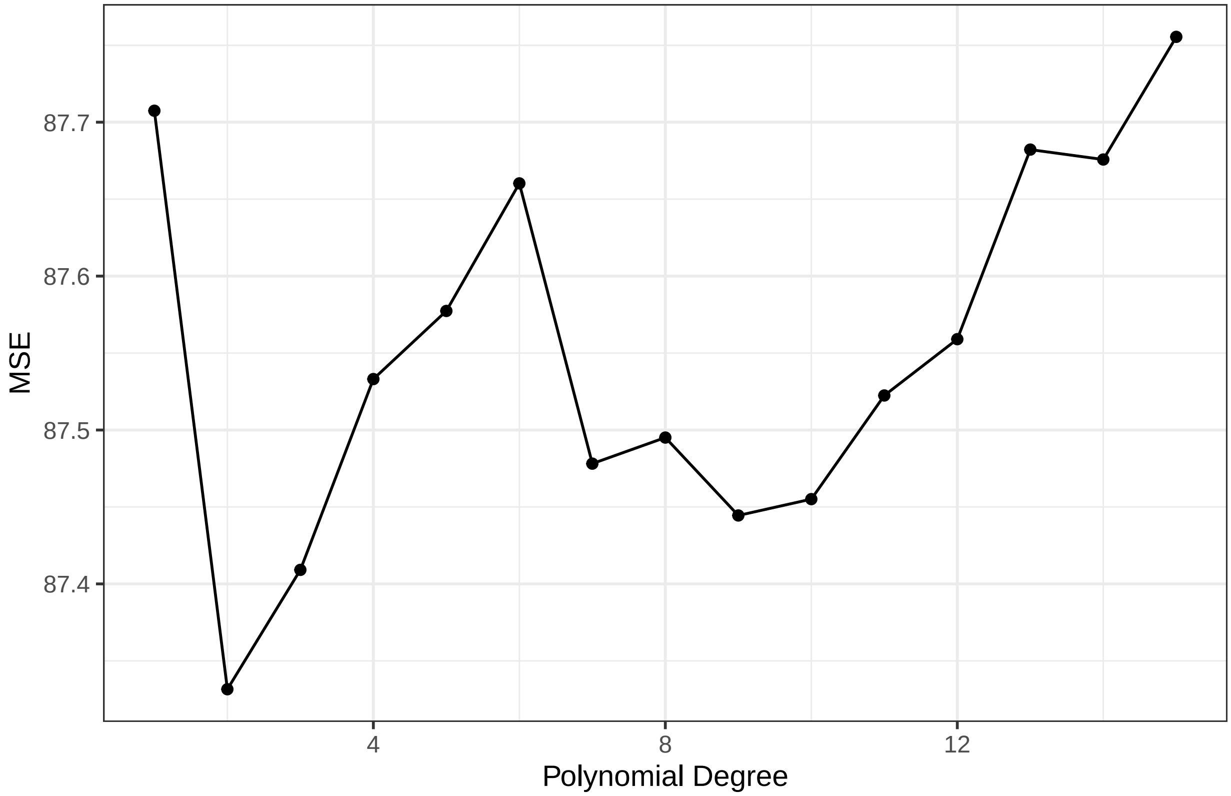
2/14/2020

Egalitarianism and income 1. (20 points) Perform polynomial regression to predict `egalit_scale` as a function of `income06`. Use and plot 10-fold cross-validation to select the optimal degree d for the polynomial based on the MSE. Plot the resulting polynomial fit to the data, and also graph the average marginal effect (AME) of `income06` across its potential values. Be sure to provide substantive interpretation of the results.

According to Fig.1, the optimal degree of the polynomial regression is 2. While the MSE results are conditional on this paricular cross-validation split, in general, polynomial regression with degree of 2 is less flexible and hence more consistent. As polynomial degree increases, our model is prone to overfitting. Fig.2 shows the average marginal effects of income, which is decreasing monotonically. As income increases to ~2.5, the AME becomes negative. However, the magnitude of the income effect on egalitarian scale is less 1. This is also shown in Fig.3 where as income increases from 1 to 25, the egalitarian scale decreases from 23-ish to 15. Depending on the unit of the income variable, its effect may not be economically significant.

```r
set.seed(625)
gss_cv10 <- vfold_cv(data = gss_train, v = 10)

MSE <- function(split, order){
  train <- analysis(split)
  validation <- assessment(split)

  model <- lm(egalit_scale~poly(income06, order, raw = TRUE), train)
  mse <- model %>% predict(validation) %>%
    {(. - validation[["egalit_scale"]])^2} %>% mean()

  mse
}

CV_MSE <- function(cv, order){
  cv %>% map(~MSE(., order)) %>% unlist() %>% mean()
}

tibble(order = 1:15) %>% mutate(MSE = map_dbl(order, ~CV_MSE(gss_cv10$splits, .))) %>%
  ggplot(aes(x = order, y = MSE)) +
  geom_point() +
  geom_line() +
  labs(x = "Polynomial Degree",
       y = "MSE",
       title = "Fig.1 MSE of Polynomial Fits with Different Degrees")
```

Fig.1 MSE of Polynomial Fits with Different Degrees



```r
optimal_degree <- lm(egalit_scale ~ I(income06^2)+income06, data = gss_train)

margins(optimal_degree, at = list(income06 = unique(gss_train$income06))) %>%
  summary() %>% as.tibble() %>%
  ggplot(aes(x = income06, y = AME)) +
  geom_line() +
  scale_y_continuous(limits = c(-1, 1)) +
  geom_hline(yintercept = 0, linetype = "dashed", alpha = 0.5) +
  labs(x = "Income", title = "Fig.2 Average Marginal Effects of Polynomial Regression with Degree of 9"
```

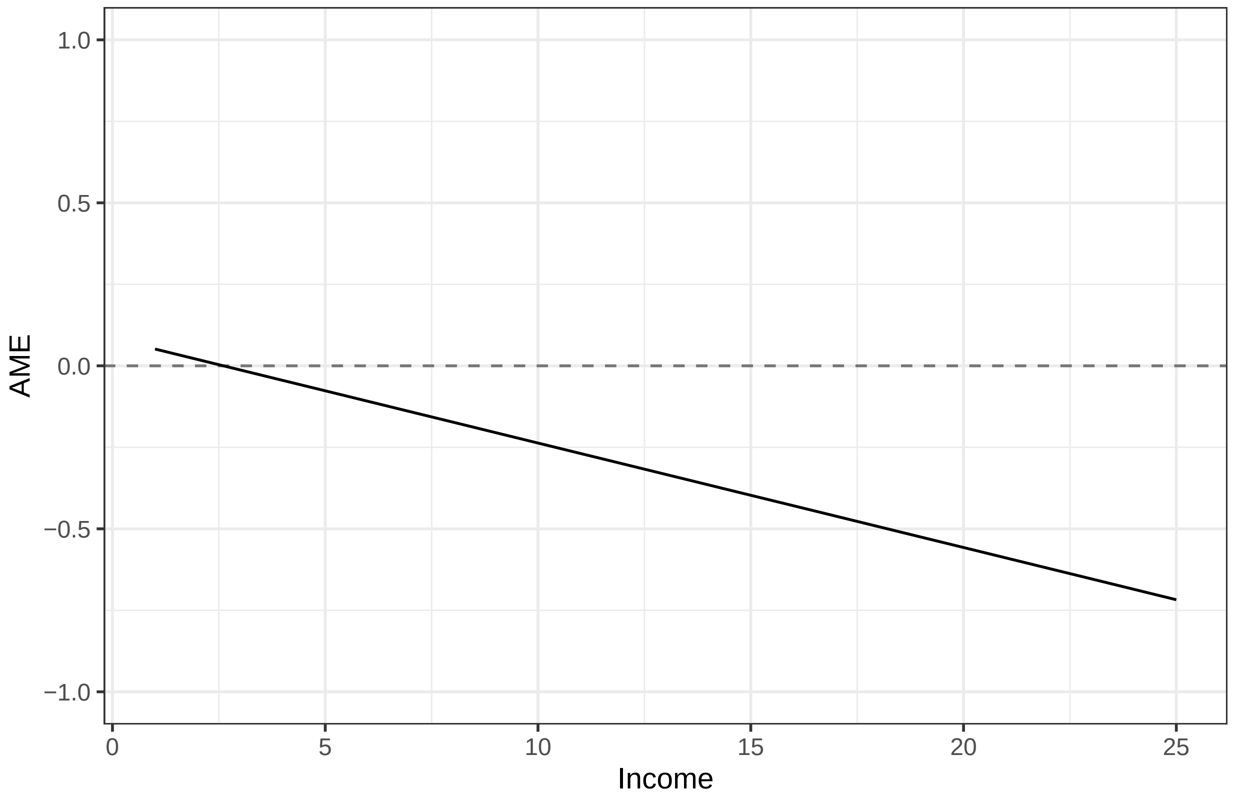
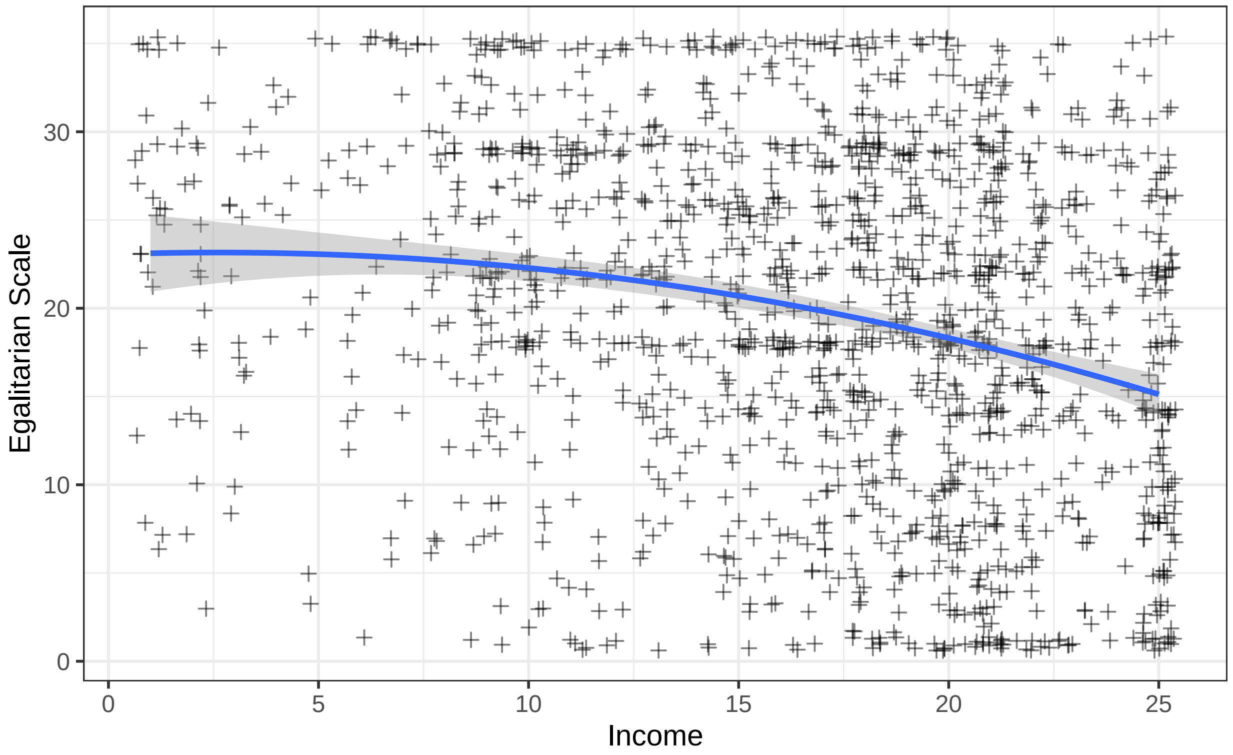


Fig.2 Average Marginal Effects of Polynomial Regression with Degree of 9

```
gss_train %>%
  ggplot(aes(x = income06, y = egalit_scale)) +
  geom_jitter(alpha = 0.5, shape = 3) +
  geom_smooth(method = "lm",
              formula = y~poly(x, 2)) +
  labs(y = "Egalitarian Scale", x = "Income",
       title = "Fig.3 Relationship between Income and Egalitarian Scale",
       subtitle = "Polynomial Regression with Degree of 9")
```

Fig.3 Relationship between Income and Egalitarian Scale

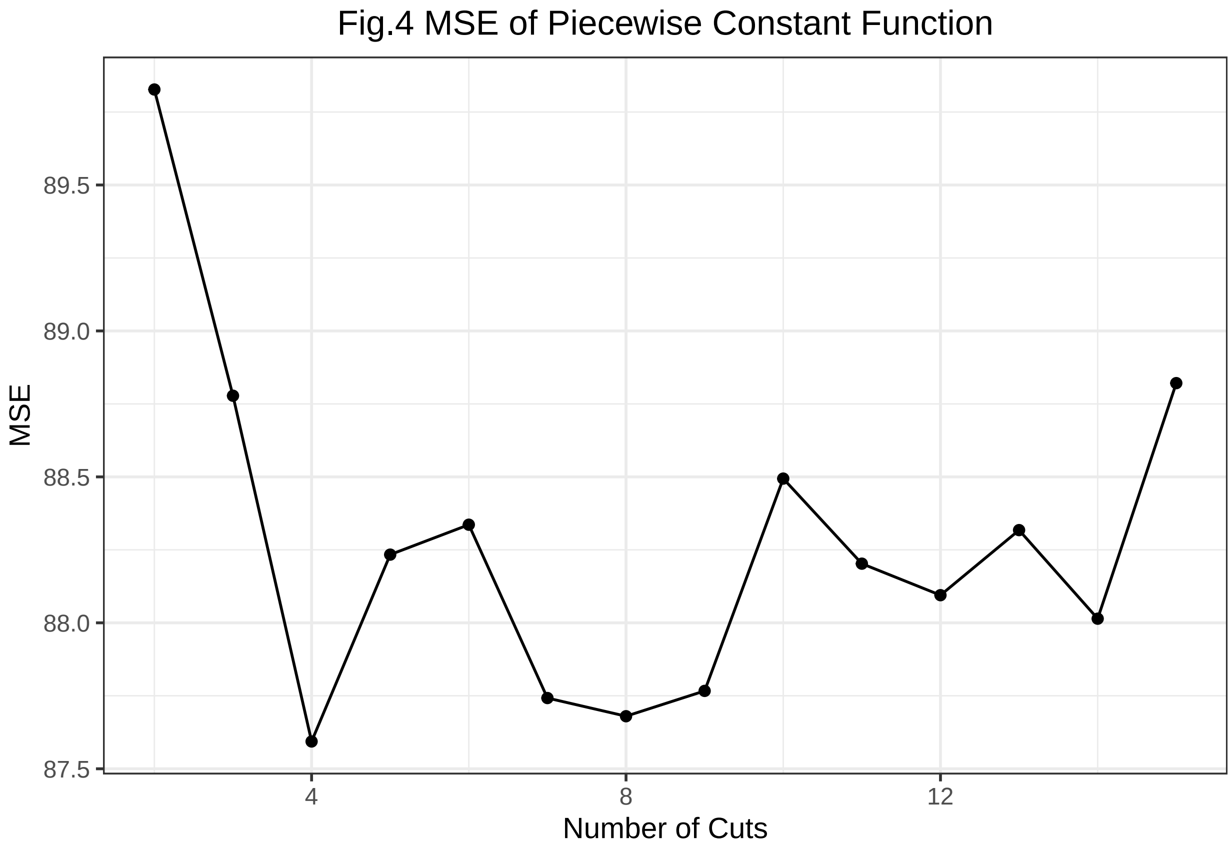
Polynomial Regression with Degree of 9



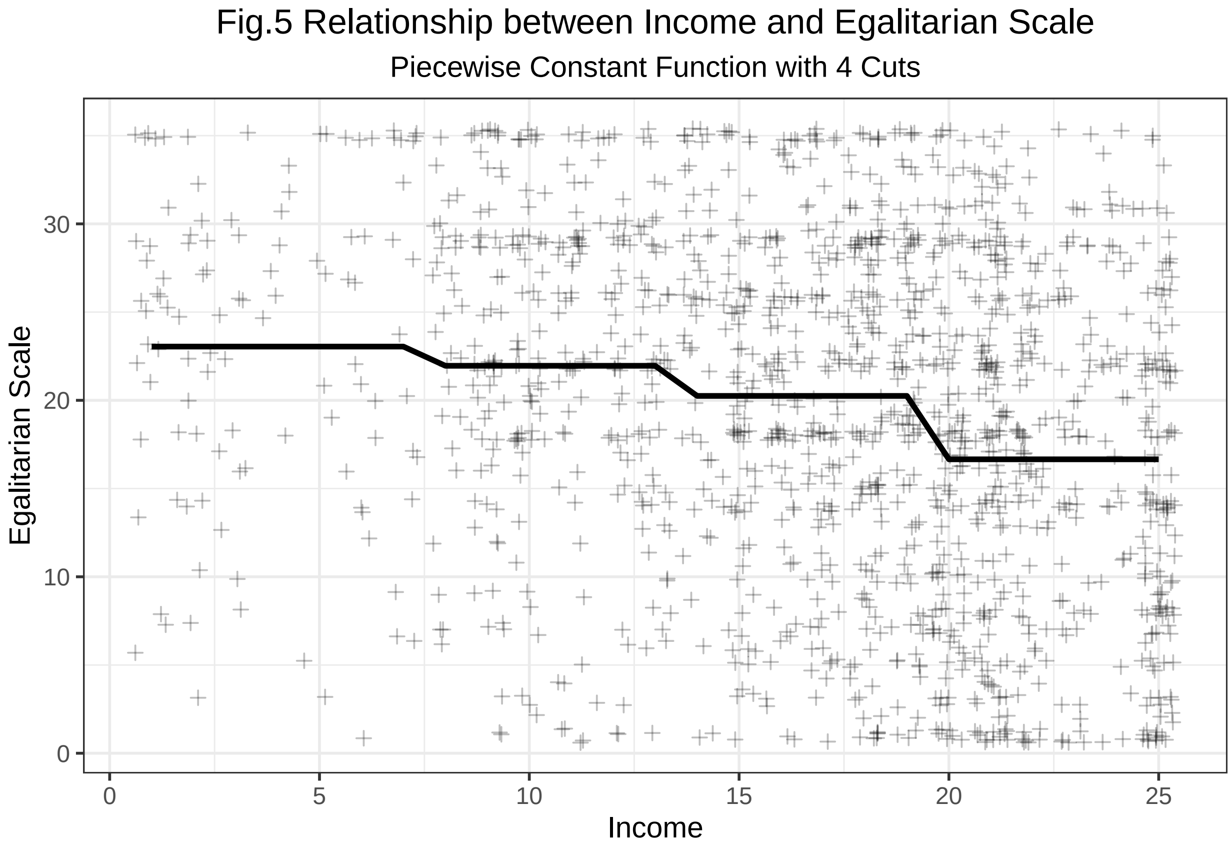
2. (20 points) Fit a step function to predict `egalit_scale` as a function of `income06`, and perform 10-fold cross-validation to choose the optimal number of cuts. Plot the fit and interpret the results.

According to Fig.4, piecewise constant function with cuts of 4 performs the best, as it has the lowest MSE. The results are also conditional on this particular seed. Fig.5 indicates a negative effect of income on egalitarian scale. As income increases, the egalitarian scale decreases. We also notice that the magnitude of the income effect increases as we move from lower cut intervals to higher cut intervals. The trend of our piecewise constant line roughly resembles the trend of the polynomial regression but is less smooth at the cut points. The MSE of step function is greater than the MSE of polynomial regression with degree of 2.

```
MSE <- function(split, breaks){
  train <- analysis(split)
  validation <- assessment(split)

  model <- lm(egalit_scale~ cut(income06, breaks, include.lowest = T), train)
  mse <- model %>% predict(validation) %>%
    {(. - validation[["egalit_scale"]])^2} %>% mean()
  mse
}


CV_MSE <- function(cv, breaks){
  cv %>% map(~MSE(., breaks)) %>% unlist() %>% mean()
}

tibble(breaks = 2:15) %>% mutate(MSE = map_dbl(breaks, ~CV_MSE(gss_cv10$splits, .))) %>%
  ggplot(aes(x = breaks, y = MSE)) +
  geom_line() +
  geom_point() +
  labs(x = "Number of Cuts",
       y = "MSE",
       title = "Fig.4 MSE of Piecewise Constant Function")
```



```
optimal_cut <- lm(egalit_scale~ cut(income06, 4, include.lowest = T), gss_train)

optimal_cut %>% predict(gss_train) %>% as.tibble() %>%
  bind_cols(gss_train) %>%
  ggplot(aes(x = income06)) +
  geom_jitter(aes(y = egalit_scale), alpha = 0.25, shape = 3) +
  geom_line(aes(y = value), size = 1) +
  labs(y = "Egalitarian Scale", x = "Income",
       title = "Fig.5 Relationship between Income and Egalitarian Scale",
       subtitle = "Piecewise Constant Function with 4 Cuts")
```

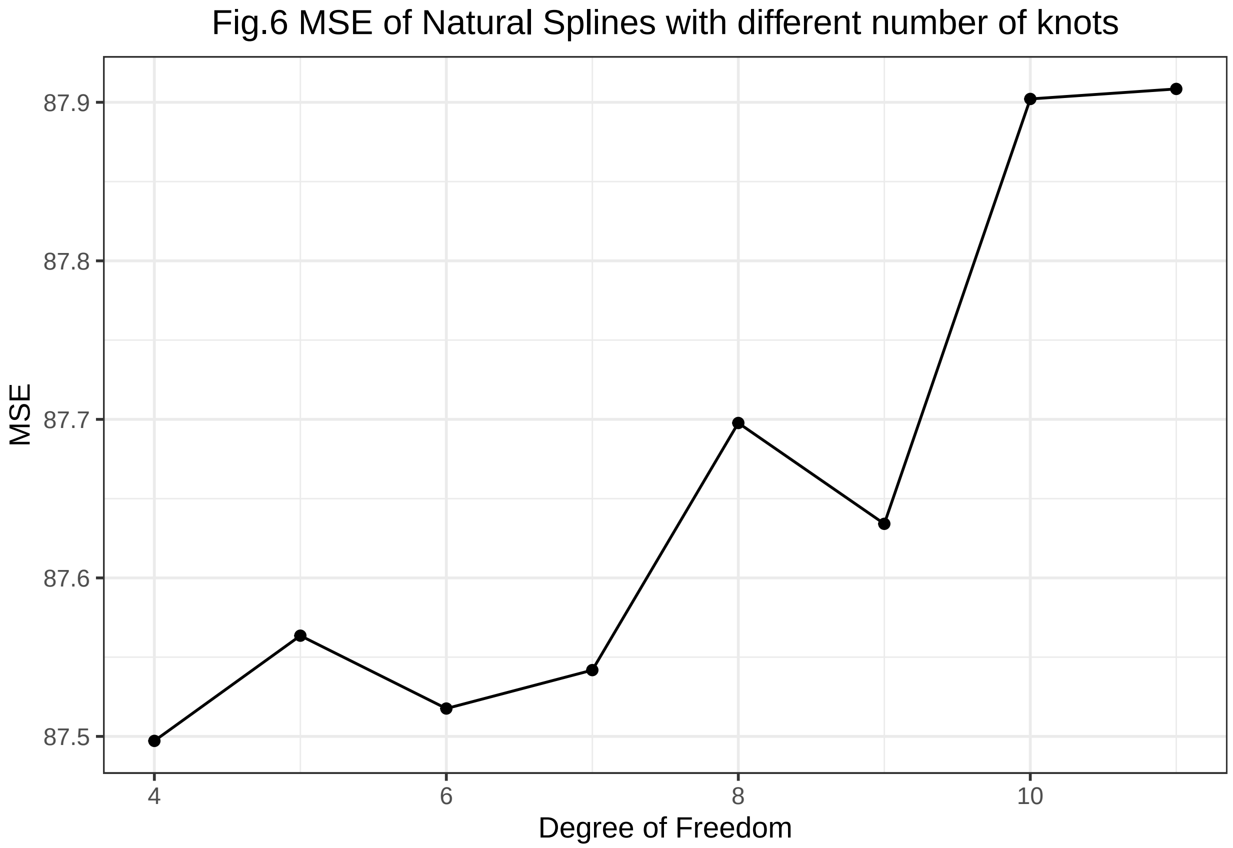


3. (20 points) Fit a natural regression spline to predict `egalit_scale` as a function of `income06`. Use 10-fold cross-validation to select the optimal number of degrees of freedom, and present the results of the optimal model.

According to Fig.6, we can see that a natural regression spline with 3 knots (one interior and two at the boundary) performs the best with the lowest MSE. Similarly, the results are conditional on this particular seed. Looking at Fig.7, the shape of our natural spline largely resembles the shape of our polynomial regression in Fig.3. However, the MSE of natural regression spline with 3 knots is about 87.5, which is slightly less than the MSE of step function but greater than than polynomial regression with degree of 2.

```
MSE <- function(split, df){
  train <- analysis(split)
  validation <- assessment(split)

  model <- glm(egalit_scale~ ns(income06, df = df), train, family = gaussian)
  mse <- model %>% predict(validation) %>%
    {(. - validation[["egalit_scale"]])^2} %>% mean()

  mse
}


CV_MSE <- function(cv, df){
  cv %>% map(~MSE(., df)) %>% unlist() %>% mean()
}

# Only allows for 1- 11 df
```

```r
tibble(df = 4:11) %>% mutate(MSE = map_dbl(df, ~CV_MSE(gss_cv10$splits, .))) %>%
  ggplot(aes(x = df, y = MSE)) +
  geom_line() +
  geom_point() +
  labs(x = "Degree of Freedom",
       title = "Fig.6 MSE of Natural Splines with different number of knots")
```

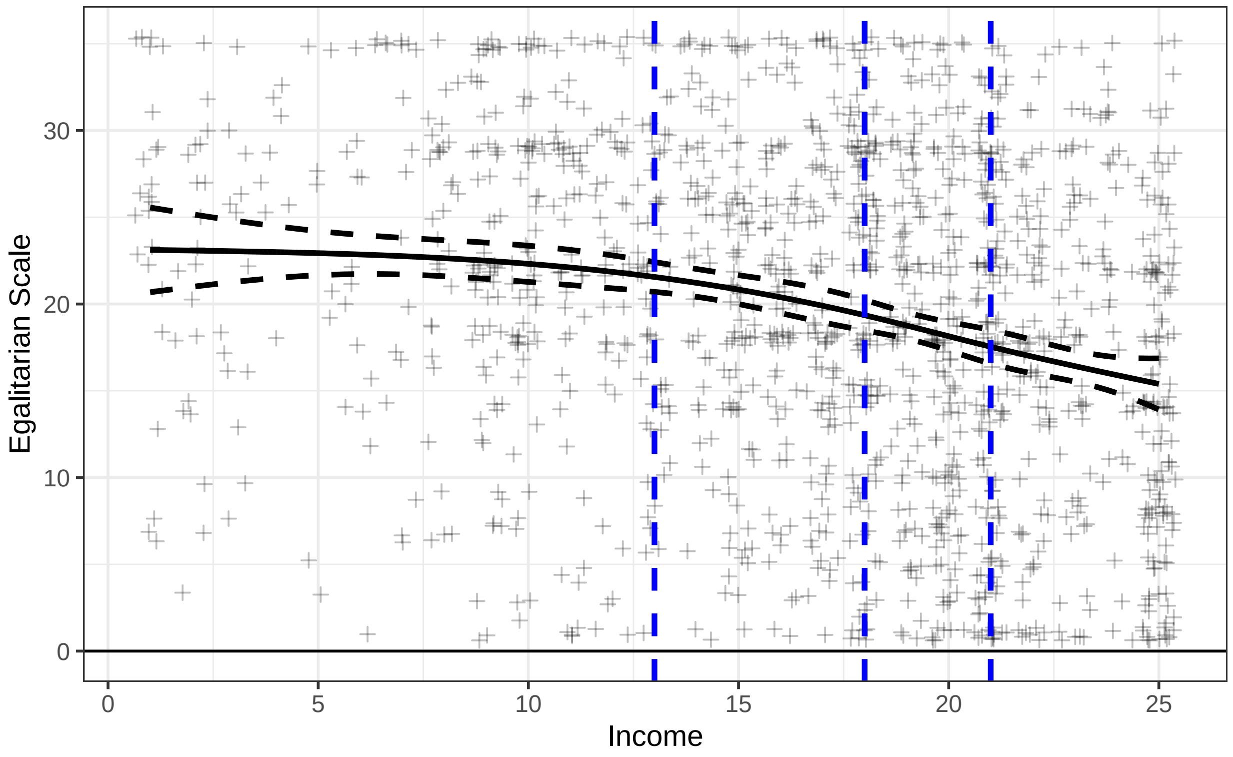


```r
# estimate model
glm(egalit_scale ~ ns(income06, df = 4), data = gss_train) %>%
  cplot("income06", what = "prediction", n = 100, draw = FALSE) %>%
  ggplot(aes(x = xvals)) +
  geom_jitter(data = gss_train, aes( x = income06, y = egalit_scale),
              alpha = 0.25, shape = 3) +
  geom_line(aes(y = yvals), size = 1) +
  geom_line(aes(y = upper), linetype = 2, size = 1) +
  geom_line(aes(y = lower), linetype = 2, size = 1) +
  geom_hline(yintercept = 0, linetype = 1) +
  geom_vline(xintercept = attr(ns(gss_train$income06, df = 4), "knots"),
             linetype = 2, color = "blue", size = 1) +
  labs(title = "Fig.7 Relationship between Income and Egalitarian Scale",
       subtitle = "Natural spline with 3 knots",
       x = "Income",
       y = "Egalitarian Scale")
```

```
##        xvals     yvals     upper     lower
## 1  1.000000 23.11796 25.55986 20.67606
## 2  1.242424 23.10910 25.46404 20.75416
## 3  1.484848 23.10019 25.36923 20.83114
## 4  1.727273 23.09116 25.27556 20.90676
## 5  1.969697 23.08197 25.18317 20.98077
## 6  2.212121 23.07256 25.09219 21.05292
## 7  2.454545 23.06286 25.00278 21.12295
## 8  2.696970 23.05284 24.91510 21.19058
## 9  2.939394 23.04243 24.82933 21.25553
## 10 3.181818 23.03157 24.74563 21.31752
## 11 3.424242 23.02022 24.66418 21.37626
## 12 3.666667 23.00831 24.58517 21.43145
## 13 3.909091 22.99579 24.50880 21.48278
## 14 4.151515 22.98261 24.43524 21.52997
## 15 4.393939 22.96870 24.36467 21.57273
## 16 4.636364 22.95402 24.29727 21.61077
## 17 4.878788 22.93851 24.23317 21.64384
## 18 5.121212 22.92210 24.17249 21.67172
## 19 5.363636 22.90476 24.11529 21.69423
## 20 5.606061 22.88642 24.06161 21.71123
```

Fig.7 Relationship between Income and Egalitarian Scale

Natural spline with 3 knots



#### Egalitarianism and everything

4. (20 points total) Estimate the following models using all the available predictors (be sure to perform appropriate data pre-processing (e.g., feature standardization) and hyperparameter tuning (e.g. lambda for PCR/PLS, lambda and alpha for elastic net). Also use 10-fold cross-validation for each model to estimate the model's performance using MSE):

```
# Data Preprocessing
# Convert character type variables to numerics and scale
gss_train <- gss_train %>%
  mutate_if(is.character, as.factor) %>%
  mutate_if(is.factor, ~as.numeric(.)-1) %>%
  mutate_all(scale) # default center = TRUE

# Same for test dataset
gss_test <- gss_test %>%
  mutate_if(is.character, as.factor) %>%
  mutate_if(is.factor, ~as.numeric(.)-1) %>%
  mutate_all(scale)


# Create an empty dataframe to store Train MSE and Test MSE for different models
MSE <- tibble(model = c("Linear Regression", "Elastic Net Regression",
                        "PCR", "PLS"),
              train_mse = vector(mode = "numeric", length = length(model)),
              test_mse = vector(mode = "numeric", length = length(model)))
```

a. (5 points) Linear regression

```
set.seed(233)

ctrl <- trainControl(method = "cv", number = 10)

olsFit <- train(
  egalit_scale ~ .,
  data = gss_train,
  method = "lm",
  trControl = ctrl
)

MSE$train_mse[grep("Linear", MSE$model)] <- olsFit$results$RMSE^2
MSE$test_mse[grep("Linear", MSE$model)] <- mean((predict(olsFit, gss_test) - gss_test$egalit_scale)^2)
```

b. (5 points) Elastic net regression

```
netFit <- train(
  egalit_scale ~ .,
  data = gss_train,
  method = "glmnet",
  trControl = ctrl,
  tuneGrid = expand.grid(alpha = seq(0, 1, by = 0.01),
                         lambda = seq(0, 1, by =0.01))
)

netFit_optimal <- train(
  egalit_scale ~ .,
  data = gss_train,
  method = "glmnet",
  trControl = ctrl,
  tuneGrid = expand.grid(alpha = netFit$bestTune$alpha,
                         lambda = netFit$bestTune$lambda)
)

MSE$train_mse[grep("Elastic", MSE$model)] <- netFit_optimal$results$RMSE^2
MSE$test_mse[grep("Elastic", MSE$model)] <- mean((predict(netFit_optimal, gss_test) - gss_test$egalit_sc
```

c. (5 points) Principal component regression

```
pcrFit <- train(
  egalit_scale~.,
  data = gss_train,
  method = "pcr",
  trControl = ctrl,
  tuneLength = 100
  )

pcrFit_optimal <- train(
  egalit_scale ~ .,
  data = gss_train,
  method = "pcr",
  tuneGrid = expand.grid(ncomp = pcrFit$bestTune$ncomp)
)

MSE$train_mse[grep("PCR", MSE$model)] <- pcrFit_optimal$results$RMSE^2
MSE$test_mse[grep("PCR", MSE$model)] <- mean((predict(pcrFit_optimal, gss_test) - gss_test$egalit_scale
```

d. (5 points) Partial least squares regression

```
plsFit <- train(egalit_scale ~ .,
                data = gss_train,
                method = "pls",
                trControl = ctrl,
                tuneLength = 100)

plsFit_optimal <- train(egalit_scale ~ .,
                        data = gss_train,
                        method = "pls",
                        trControl = ctrl,
                        tuneGrid = expand.grid(ncomp = plsFit$bestTune$ncomp))

MSE$train_mse[grep("PLS", MSE$model)] <- plsFit_optimal$results$RMSE^2
MSE$test_mse[grep("PLS", MSE$model)] <- mean((predict(plsFit_optimal, gss_test) - gss_test$egalit_scale
```

```
MSE %>% kable()
```

model	train_mse	test_mse
Linear Regression	0.6896794	0.7053012
Elastic Net Regression	0.6823771	0.6914248
PCR	0.7074362	0.7004358
PLS	0.6919989	0.7052995

5. (20 points) For each final tuned version of each model fit, evaluate feature importance by generating feature interaction plots. Upon visual presentation, be sure to discuss the substantive results for these models and in comparison to each other (e.g., talk about feature importance, conditional effects, how these are ranked differently across different models, etc.).

For each of the four models, we select the top 2 important features and create an interactio plot. When calculating the feature importance, the author use the canned function in caret package- varImp - to measure feature importance for linear regression and elastic net. However, for PCR and PLS, a different calculation method is adopted. The author first derives the loading vectors ϕ of all principal components (42 for PCR and 9 for PLS) and use the average loadings of each feature to measure feature importance. Since linear regression and elastic net regression identify the same top 2 most important features, only one interaction plot is created.

According to the result tables, we can see that survey respondents' vote in the 2008 presidential election has relatively greater impacts on the egalitarian scale, identified as most important by linear regression, elastic net regression and partical least squares. As indicated in Fig.1 and Fig.3, people who voted for Obama tend to have higher egalitarian scores. Party affiliation and the measure of economic liberalism together show that people with more conservative ideologies tend to score less on the egalitarian scale than people with more democratic views. However, when it comes to people's stands on specific social-political issues- for example, whether the respondent supports whether anti-religionist should be allowed to teach- , its effect is conditional on the person's ideology. In addition, Fig.3 also shows that African Americans tend to have greater egalitarian scores.

Overall, among the top 10 most important features identified by the four models, linear regression and elastic net regression share a lot of similarities, while having few overlappings with PCR or PLS. PCR and PLS also have a large degree of disagreements.

1. Linear Regression

```
gss_train <-read_csv("./data/gss_train.csv")

olsFit %>% varImp() %>%
  {.$importance} %>% rownames_to_column() %>%
  {.[order(.$Overall, decreasing = TRUE),]} %>%
  {.[1:10,]} %>% kable()
```

	rowname	Overall
29	pres08	100.00000
25	partyid_3	92.37881
16	grass	61.90463
1	age	52.37114
20	income06	52.02336
26	polviews	41.49074
40	tolerance	39.68525
33	sex	38.15006
24	owngun	29.54579
41	tvhours	28.29138

2. Elastic Net

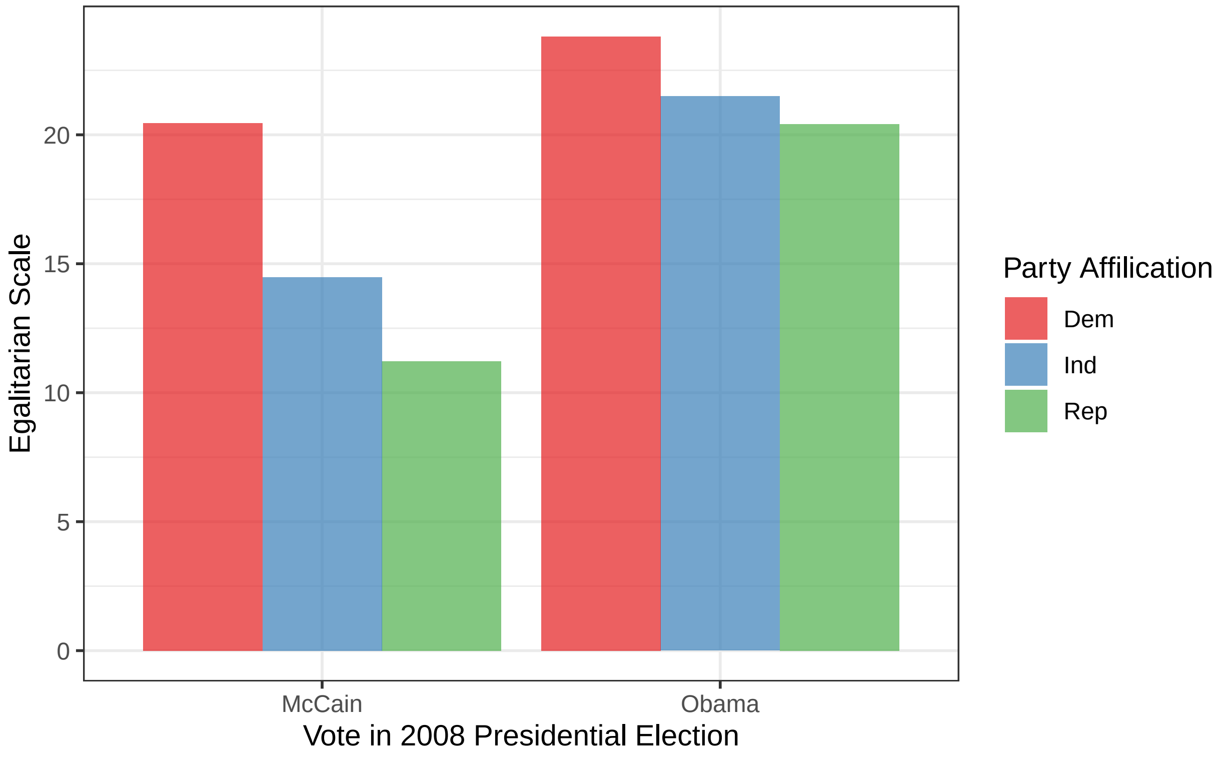
```
netFit_optimal %>% varImp() %>%
  {.$importance} %>% rownames_to_column() %>%
  {.[order(.$Overall, decreasing = TRUE),]} %>%
  {.[1:10,]} %>% kable()
```

	rowname	Overall
29	pres08	100.00000
25	partyid_3	82.80484
16	grass	46.39468
20	income06	41.82440
1	age	39.52528
40	tolerance	32.09277
26	polviews	30.38839
33	sex	27.05782
24	owngun	25.32298
41	tvhours	20.08684

```
gss_train %>% group_by(pres08, partyid_3) %>%
  summarise(egalit_scale = mean(egalit_scale)) %>%
  ggplot( aes(y = egalit_scale)) +
  geom_bar(aes(x = pres08, fill = as.factor(partyid_3)), stat = "identity", position = "dodge", alpha =
  scale_fill_brewer(palette = "Set1") +
  labs(x = "Vote in 2008 Presidential Election",
       y = "Egalitarian Scale",
       fill = "Party Affilication",
       title = "Fig. 1 Interactions between Most Important Features",
       subtitle = "Identified by Linear/Elastic Net Regression")
```

Fig. 1 Interactions between Most Important Features

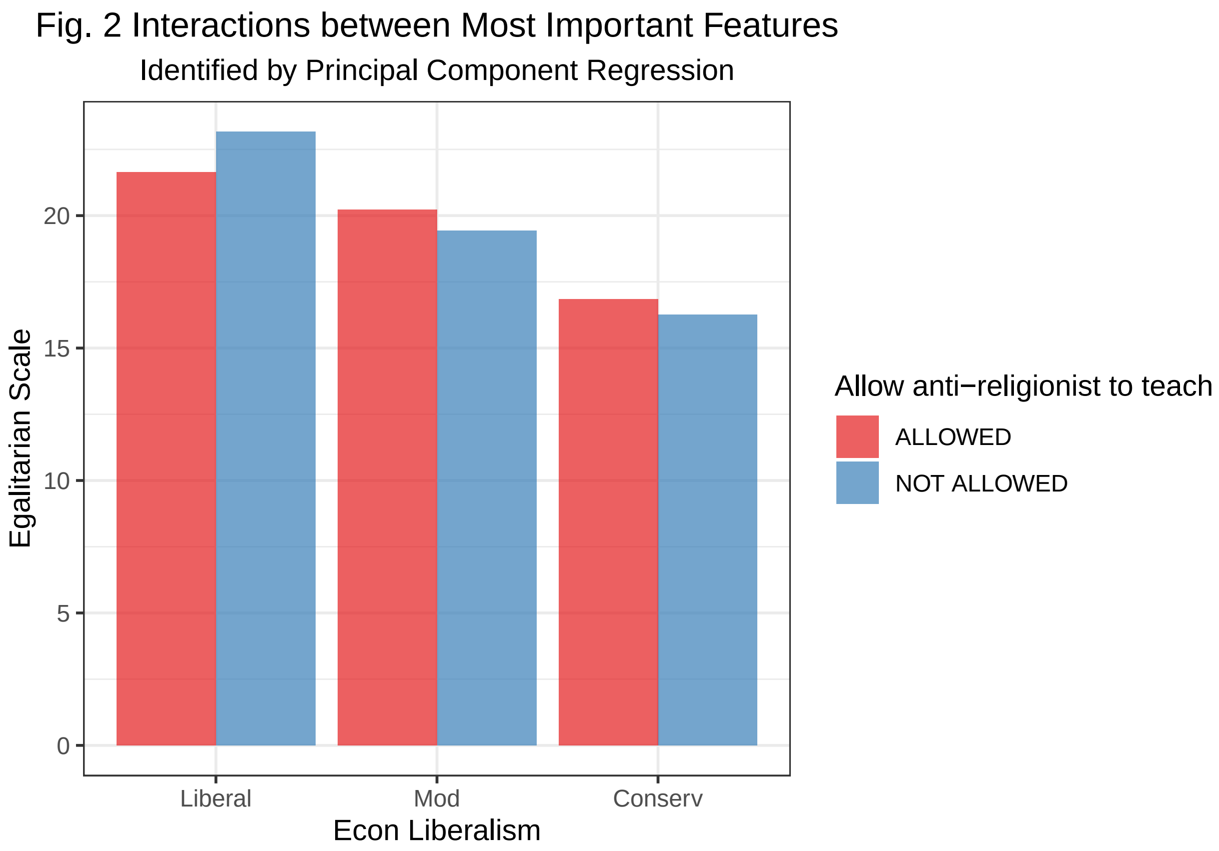
Identified by Linear/Elastic Net Regression



```
pcrFit_optimal$finalModel %>% loadings() %>%
  {.[,1:42]} %>% as.data.frame() %>% #rownames_to_column() #%>%
  apply(1, mean) %>% as.data.frame() %>%rownames_to_column() %>%
  #`colnames<-`(c("Feature", "First Principal Component")) %>%
  {.[order(.$., decreasing = TRUE),]} %>%
  {.[1:10,]} %>% kable()
```

	rowname	.
38	spend3	0.0476135
7	colath	0.0401549
3	authoritarianism	0.0317431
24	owngun	0.0293923
19	homosex	0.0288733
23	news	0.0272759
17	happy	0.0239760
6	childs	0.0189864
10	colmil	0.0174365
11	colhomo	0.0147130

```
gss_train %>% group_by(spend3, colath) %>%
  summarise(egalit_scale = mean(egalit_scale)) %>%
  ggplot( aes(y = egalit_scale)) +
  geom_bar(aes(x = factor(spend3, levels = c("Liberal", "Mod", "Conserv")), fill = as.factor(colath)), 
  scale_fill_brewer(palette = "Set1") +
  labs(x = "Econ Liberalism",
       y = "Egalitarian Scale",
       fill = "Allow anti-religionist to teach",
       title = "Fig. 2 Interactions between Most Important Features",
       subtitle = "Identified by Principal Component Regression")
```



```
plsFit_optimal$finalModel %>% loadings() %>%
  {.[,1:9]} %>% as.data.frame() %>% #rownames_to_column() #%>%
  apply(1, mean) %>% as.data.frame() %>%rownames_to_column() %>%
  #`colnames<-`(c("Feature", "First Principal Component")) %>%
  {.[order(.$., decreasing = TRUE),]} %>%
  {.[1:10,]} %>% kable()
```

	rowname	.
29	pres08	0.0705759
34	sibs	0.0408953
36	social_cons3	0.0370892
6	childs	0.0348653
41	tvhours	0.0278454
26	polviews	0.0196717
2	attend	0.0173596
30	reborn_r	0.0173486
31	relig	0.0170704

	rowname	.
7	colath	0.0163359

```
gss_train %>% group_by(pres08, black) %>%
  summarise(egalit_scale = mean(egalit_scale)) %>%
  ggplot( aes(y = egalit_scale)) +
  geom_bar(aes(x = pres08, fill = as.factor(black)), stat = "identity", position = "dodge", alpha = 0.7
  scale_fill_brewer(palette = "Set1") +
  labs(x = "Vote in 2008 Presidential Election",
       y = "Egalitarian Scale",
       fill = "African American",
       title = "Fig. 3 Interactions between Most Important Features",
       subtitle = "Identified by Linear/Elastic Net Regression")
```

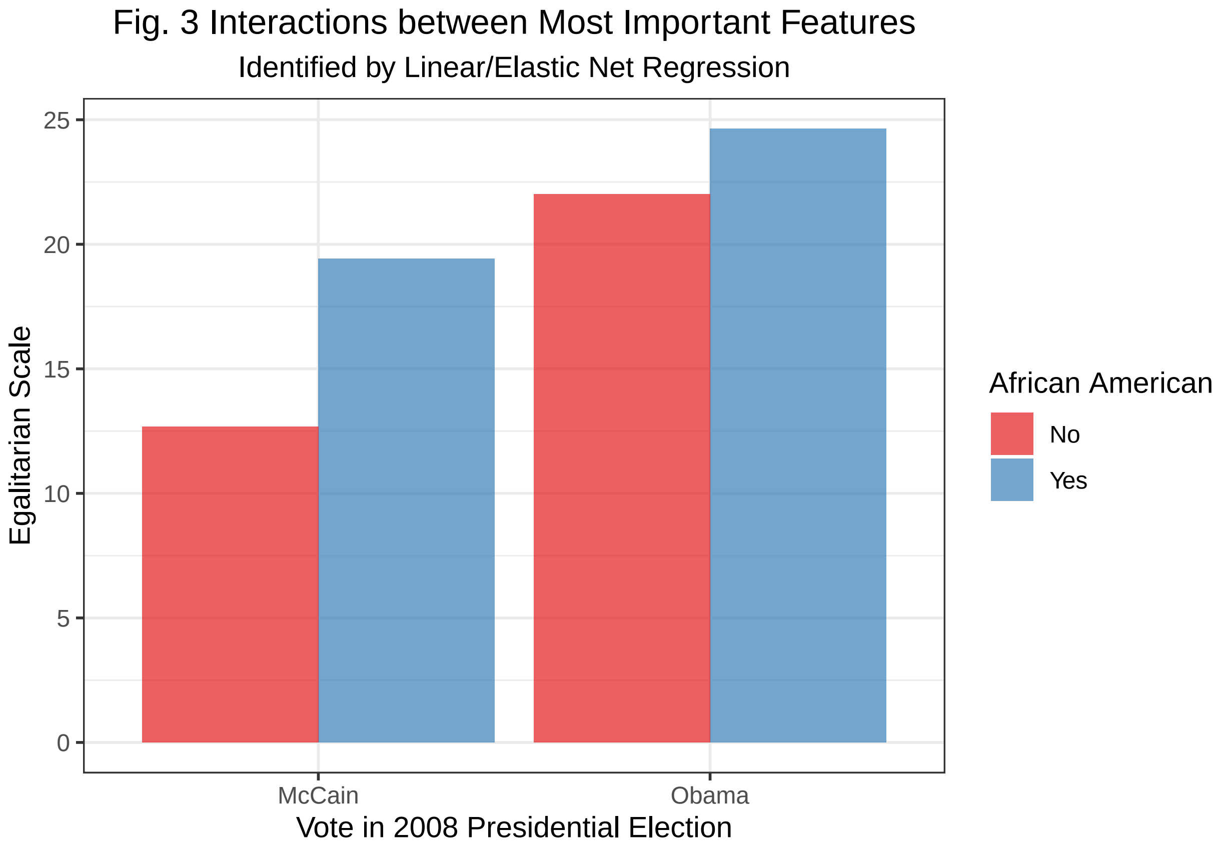


Fig. 3 Interactions between Most Important Features
Identified by Linear/Elastic Net Regression
Egalitarian Scale
McCain
Obama
Vote in 2008 Presidential Election
African American
No
Yes