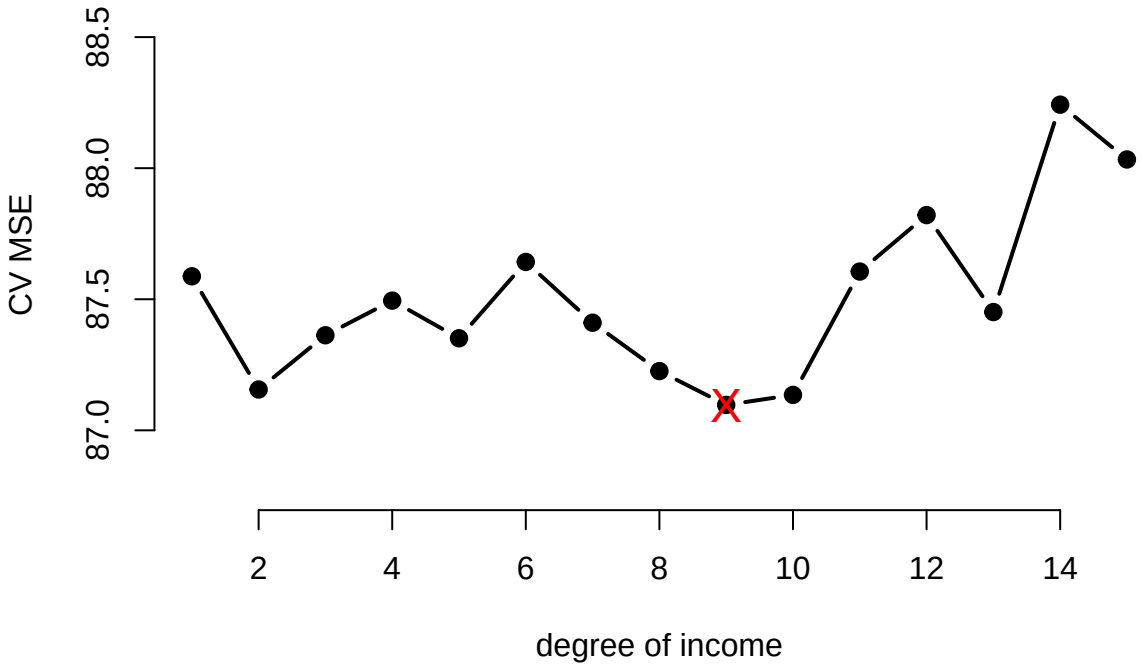
HW4_Responses

Yawei LI

02/17/2020

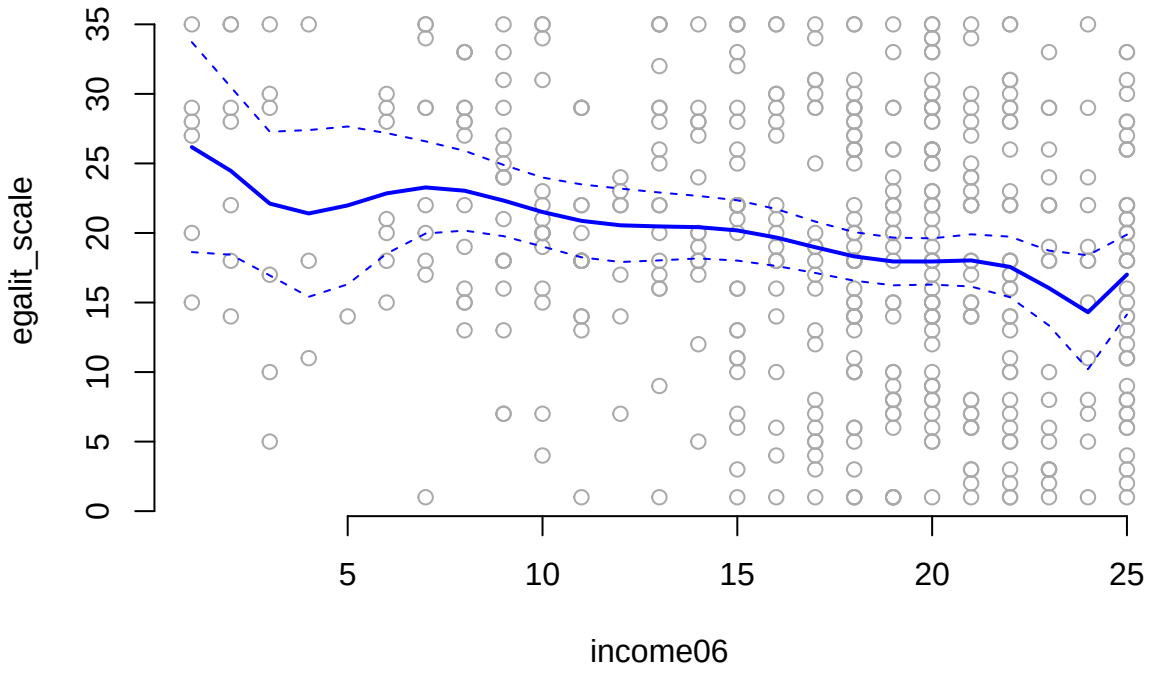
Question 1

```
gss_train <- read.csv('gss_train.csv')
gss_test <- read.csv('gss_test.csv')
set.seed(1234)

poly_mse <- NA

for (i in 1:15) {
  glm.fit <-  glm(egalit_scale ~ poly(income06, i), data = gss_train)
  poly_mse[i] <-  cv.glm(gss_train, glm.fit, K = 10)$delta[1]
}
plot( x = 1:15, y = poly_mse, xlab = "degree of income", ylab = "CV MSE",
      type = "b", pch = 19, lwd = 2, bty = "n",
      ylim = c( min(poly_mse) - sd(poly_mse), max(poly_mse) + sd(poly_mse) ) )

points( x = which.min(poly_mse), y = min(poly_mse), col = "red", pch = "X", cex = 1.5 )
```



From the picture above, we choose degree of income at 9.

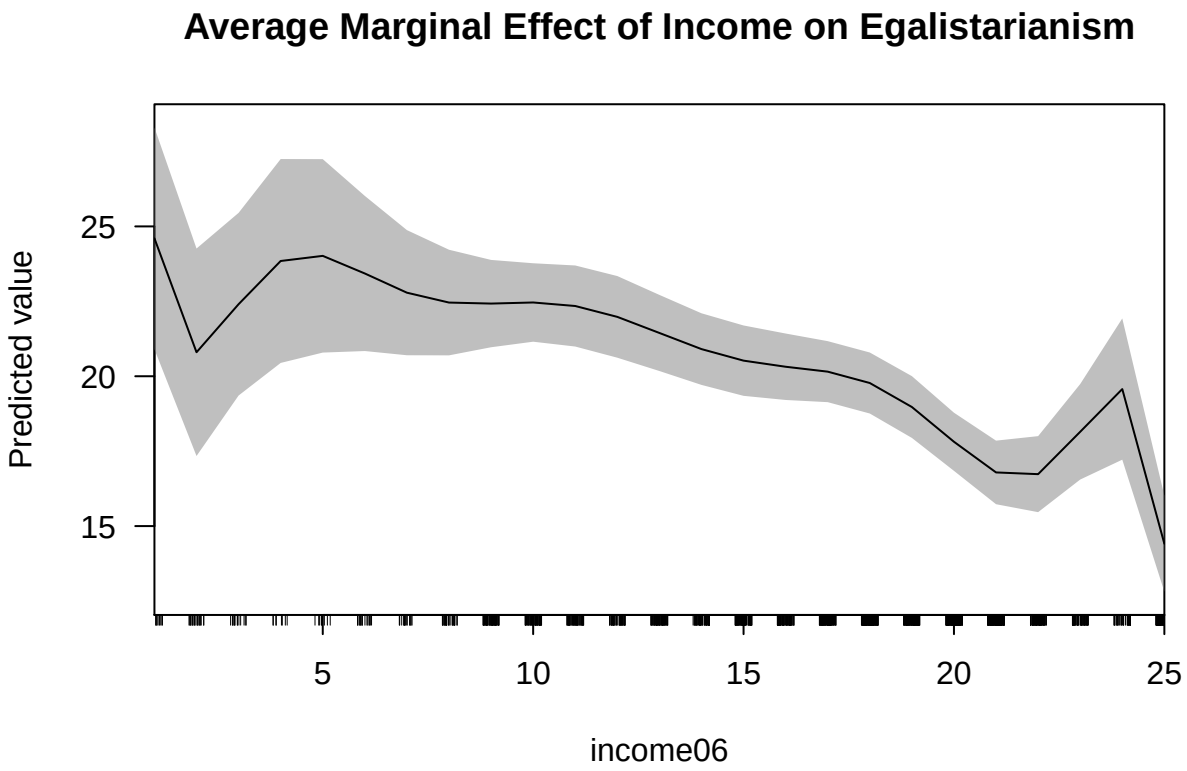
```
plot(egalit_scale ~ income06, data = gss_test, col = "darkgrey",  bty = "n")
income06lims <-  range(gss_test$income06)
income06.grid <-  seq(from = income06lims[1], to = income06lims[2])
lm.fit <-  lm(egalit_scale ~ poly(income06, 9), data = gss_test)
lm.pred <-  predict(lm.fit, data.frame(income06 = income06.grid), se = TRUE)
lines(x = income06.grid , y = lm.pred$fit, col = "blue", lwd = 2)
matlines( x = income06.grid, y = cbind( lm.pred$fit + 2*lm.pred$se.fit, lm.pred$fit - 2*lm.pred$se.fit)
          lty = "dashed", col = "blue")
```




```
glm.fit <-  glm(egalit_scale ~ poly(income06, 9), data = gss_train)

#cplot(glm.fit,'income06',what = 'effect', main = 'Average Marginal Effect of Income on Egalistarianism
# For some unknwon reason, the comment code above did not work.
# Error in poly(income06, 9, coefs = list(alpha = c(15.6468602295746, 11.6238211754894, : could not fin

# Using predicted value instead, to show that cplot() is working.
cplot(glm.fit,'income06', main = 'Average Marginal Effect of Income on Egalistarianism')
```

```
##    xvals    yvals    upper    lower
## 1      1 24.60194 28.30223 20.90166
## 2      2 20.80013 24.26187 17.33839
## 3      3 22.40453 25.45259 19.35647
## 4      4 23.84596 27.24810 20.44383
## 5      5 24.01688 27.24474 20.78901
## 6      6 23.42906 26.01710 20.84101
## 7      7 22.78826 24.87627 20.70025
## 8      8 22.45908 24.22175 20.69641
## 9      9 22.42487 23.88209 20.96765
## 10    10 22.46173 23.77069 21.15278
## 11    11 22.34415 23.69697 20.99133
## 12    12 21.98079 23.34323 20.61836
## 13    13 21.44419 22.71433 20.17406
## 14    14 20.90589 22.10137 19.71040
## 15    15 20.52050 21.69463 19.34636
## 16    16 20.31691 21.42601 19.20781
```

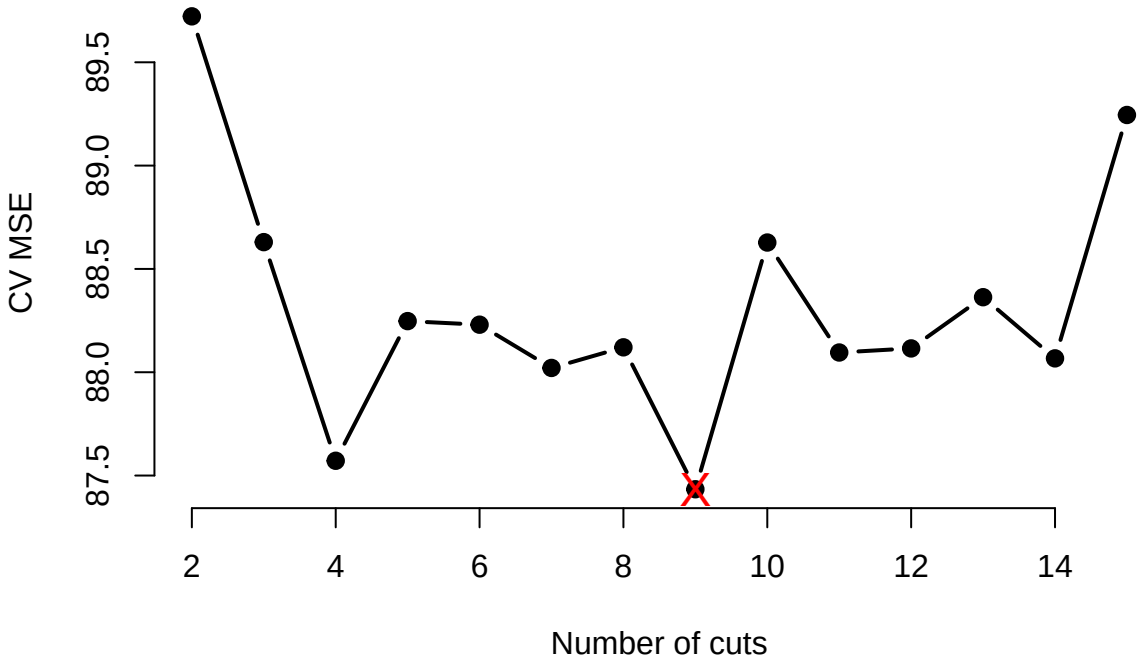
```
## 17    17 20.15326 21.17108 19.13544
## 18    18 19.77405 20.79180 18.75630
## 19    19 18.97296 20.00397 17.94196
## 20    20 17.81331 18.78353 16.84310
```



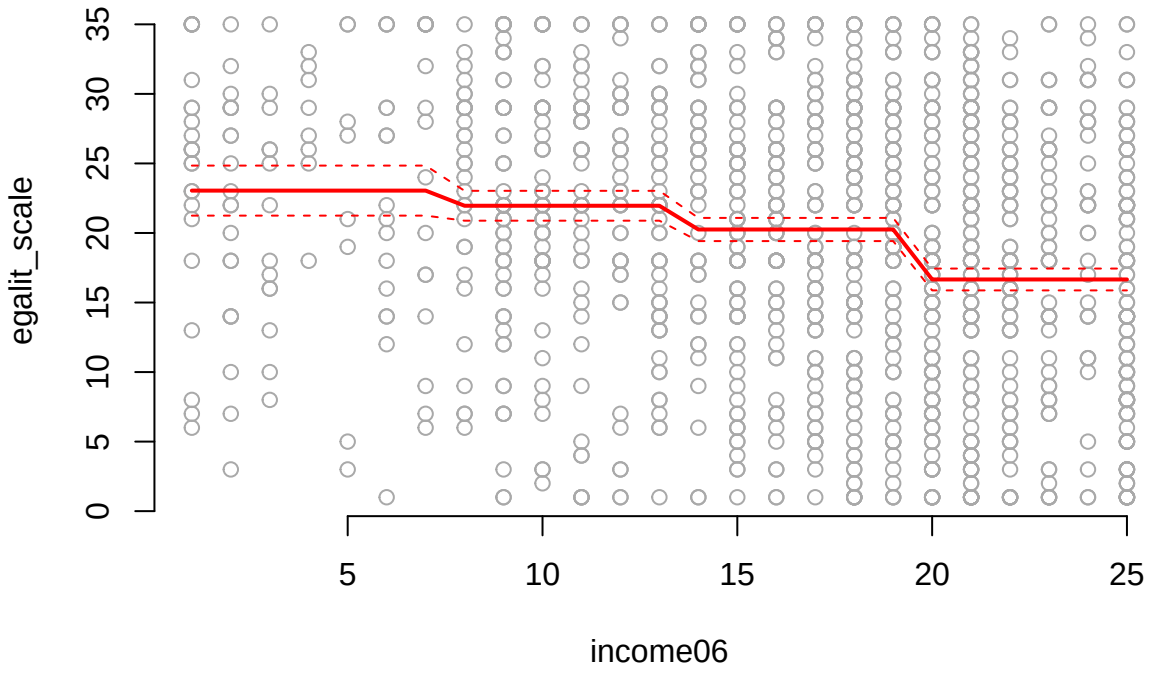
As can be seen from the plot above, income has negative effect on egalistarianism. On lower income levels, its effect is rather limited and unstable. The negative effect stablizes as the income level becomes higher.

Question 2

```r
set.seed(1234)
cut_mse <-  NA
for (i in 2:15) {
  gss_train$income.cut <-  cut(gss_train$income06, i)
  lm.fit <-  glm(egalit_scale ~ income.cut, data = gss_train)
  cut_mse[i] <-  cv.glm(gss_train, lm.fit, K = 10)$delta[1]
}

plot(2:15, cut_mse[-1], xlab = "Number of cuts", ylab = "CV MSE",
     type = "b", pch = 19, lwd = 2, bty ="n")

points( x = which.min(cut_mse), y = min(cut_mse, na.rm = TRUE), col = "red", pch = "X", cex = 1.5 )
```



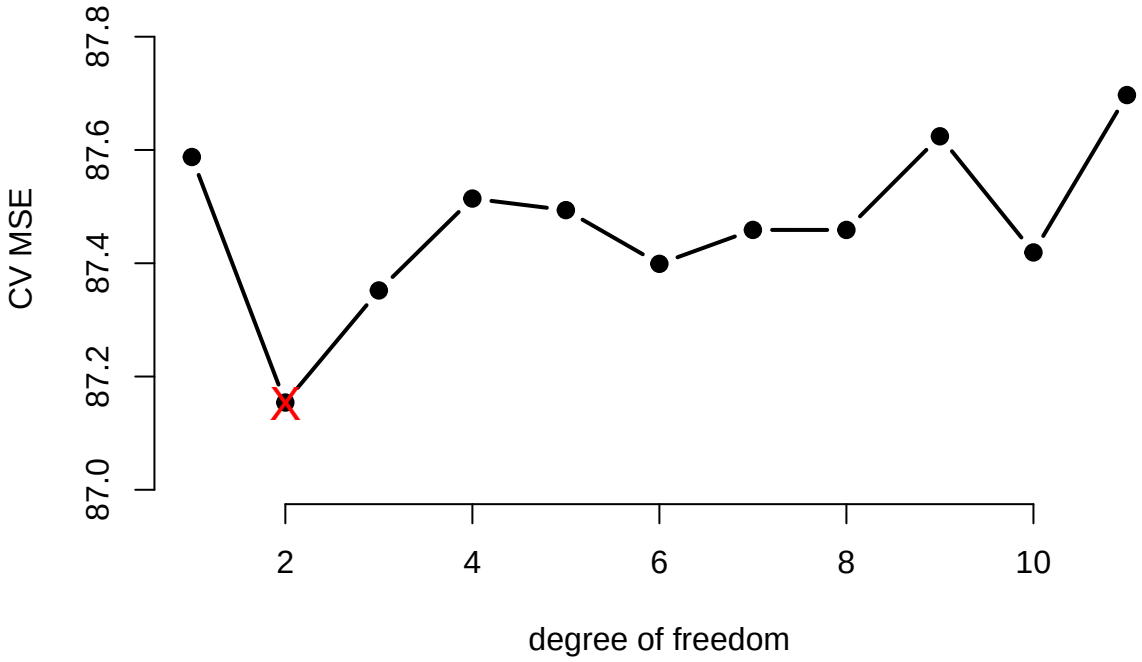

```
lm.fit <-  glm(egalit_scale ~ cut(income06, 4), data = gss_train)
income06lims <-  range(gss_train$income06)
income06.grid <-  seq(from = income06lims[1], to = income06lims[2])
lm.pred <-  predict(lm.fit, data.frame(income06 = income06.grid), se = TRUE)
plot(egalit_scale ~ income06, data = gss_train, col = "darkgrey", bty = "n")
lines(income06.grid, lm.pred$fit, col = "red", lwd = 2)
matlines(income06.grid, cbind( lm.pred$fit + 2* lm.pred$se.fit,
                          lm.pred$fit - 2* lm.pred$se.fit),
         col = "red", lty ="dashed")
```



As can be seen from the picture above, income has a negative effect on egalit_scale. However, compared to polynomial regression, step function yields a much smaller estimate of the effect. Still, the effect of income is higher at higher income levels.

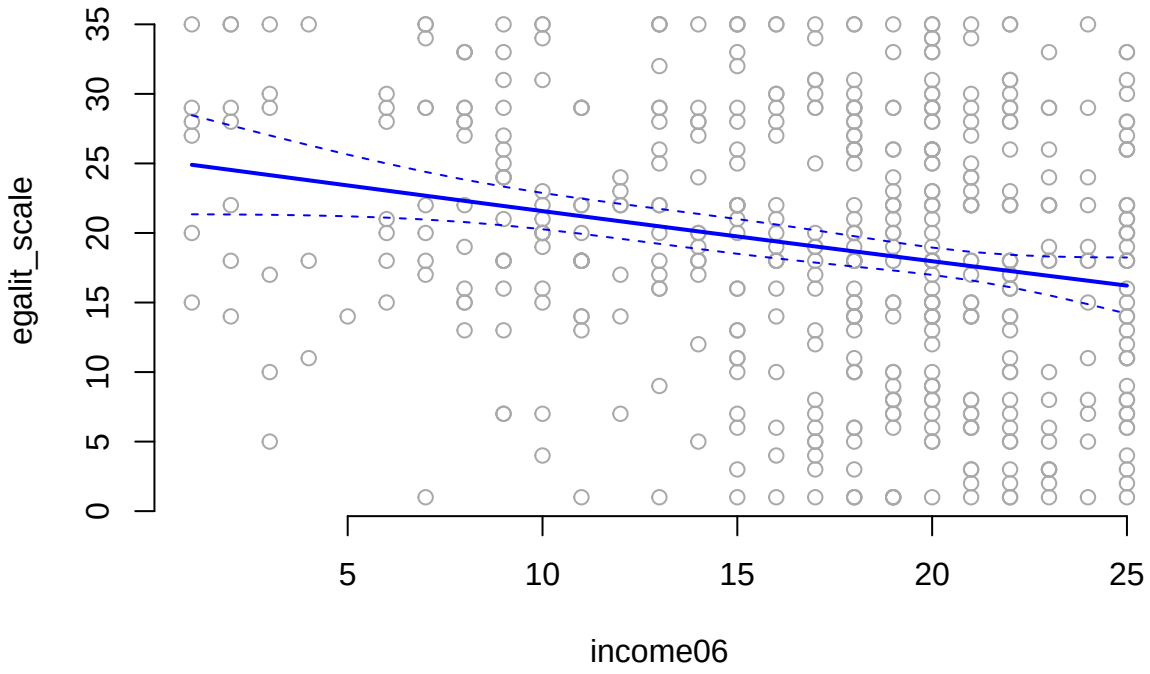
Question 3

```
set.seed(1234)
ns_mse <-  NA

for (i in 1:11) {
  glm.fit <-  glm(egalit_scale ~ ns(income06, df=i), data = gss_train)
  ns_mse[i] <-  cv.glm(gss_train, glm.fit, K = 10)$delta[1]
}

plot( x = 1:11, y = ns_mse, xlab = "degree of freedom", ylab = "CV MSE",
      type = "b", pch = 19, lwd = 2, bty = "n",
      ylim = c( min(ns_mse) - sd(ns_mse), max(ns_mse) + sd(ns_mse) ) )


points( x = which.min(ns_mse), y = min(ns_mse), col = "red", pch = "X", cex = 1.5 )
```



From the picture above, we choose degree of freedom at 2.

```
plot(egalit_scale ~ income06, data = gss_test, col = "darkgrey",  bty = "n")
income06lims <-  range(gss_test$income06)
income06.grid <-  seq(from = income06lims[1], to = income06lims[2])
lm.fit <-  lm(egalit_scale ~ ns(income06, df=2), data = gss_test)
lm.pred <-  predict(lm.fit, data.frame(income06 = income06.grid), se = TRUE)
lines(x = income06.grid , y = lm.pred$fit, col = "blue", lwd = 2)
matlines( x = income06.grid, y = cbind( lm.pred$fit + 2*lm.pred$se.fit, lm.pred$fit - 2*lm.pred$se.fit)
          lty = "dashed", col = "blue")
```




```
summary(lm.fit)
```

```
##
## Call:
## lm(formula = egalit_scale ~ ns(income06, df = 2), data = gss_test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.6780  -7.2064   0.0307   7.4330  18.4330
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             24.903      1.780  13.990  < 2e-16 ***
## ns(income06, df = 2)1  -11.558      3.646  -3.170  0.00162 **
## ns(income06, df = 2)2   -5.830      1.417  -4.114 4.56e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.292 on 490 degrees of freedom
## Multiple R-squared:  0.05041,    Adjusted R-squared:  0.04653
## F-statistic: 13.01 on 2 and 490 DF,  p-value: 3.138e-06
```

The results of the optimal model is summarized above.

Question 4

Linear Regression

```
set.seed(1234)
gss_train <- na.omit(read.csv('gss_train_f.csv'))
gss_test <- na.omit(read.csv('gss_test_f.csv'))
mean_sum_error <- function(x1,x2)
{
  sum = numeric()
  for (i in 1:length((x1))) {
    sum[i] = (x1[i]-x2[i])^2
  }
  return(mean(sum))
}
gss_train_x <- model.matrix(egalit_scale~ ., gss_train)[, -1]
gss_train_y <- gss_train$egalit_scale

gss_test_x <- model.matrix(egalit_scale ~ ., gss_test)[, -1]
gss_test_y <- gss_test$egalit_scale

ls <- glm(egalit_scale ~ .,data = gss_train)
test_mse = numeric(0)
test_mse[1] <-  cv.glm(gss_test, ls, K = 10)$delta[1]

paste('Test MSE for linear model is',test_mse[1])
```

```
## [1] "Test MSE for linear model is 130.945530201372"
```

Elastic Net

```
set.seed(1234)

fold_id <- sample(1:10, size = length(gss_train_y), replace = TRUE)

tuning_grid <- tibble::tibble(
  alpha      = seq(0, 1, by = .1),
  mse_min    = NA,
  lambda_min = NA
)

for(i in seq_along(tuning_grid$alpha)) {
  fit <- cv.glmnet(gss_train_x,
                   gss_train_y,
                   alpha = tuning_grid$alpha[i],
                   foldid = fold_id)

    tuning_grid$mse_min[i]    <- fit$cvm[fit$lambda == fit$lambda.min]
    tuning_grid$lambda_min[i] <- fit$lambda.min
}
  min_mse = min(tuning_grid$mse_min)
  paste('The lowest MSE is',min_mse,'with alpha =',
```

```
        tuning_grid$alpha[which(tuning_grid$mse_min == min_mse)],
        'and lambda =',
        tuning_grid$lambda_min[which(tuning_grid$mse_min == min_mse)])
```

```
## [1] "The lowest MSE is 59.5626796259866 with alpha = 1 and lambda = 0.212784533609135"
```

```
p_net = predict(fit,newdata = gss_test, newx = gss_test_x)
test_mse[2] <- mean_sum_error(p_net,gss_test$egalit_scale)
paste('Test MSE is',test_mse[2])
```

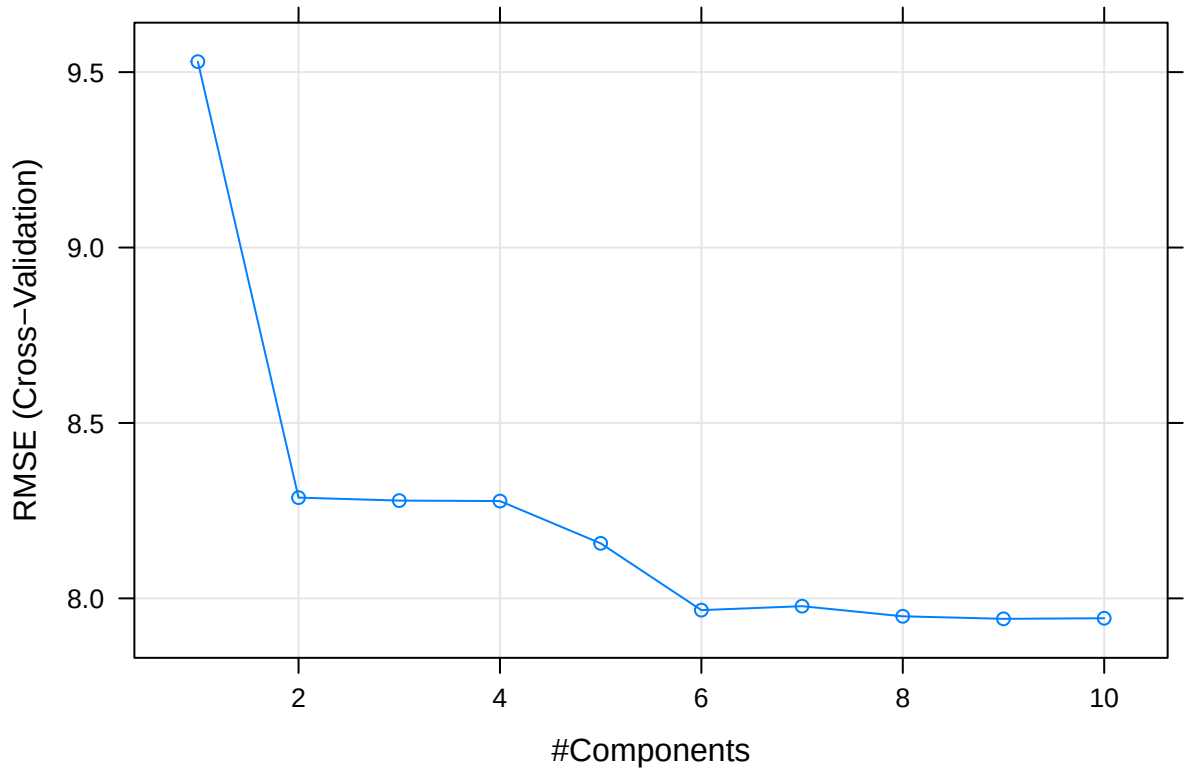
```
## [1] "Test MSE is 62.0982758128179"
```

PCR

```
set.seed(1234)
pcr <- train(
  egalit_scale~., data = gss_train, method = "pcr",
  scale = TRUE,
  trControl = trainControl("cv", number = 10),
  tuneLength = 10
  )
paste("Best tuned model is at",pcr$bestTune,'components')
```

```
## [1] "Best tuned model is at 9 components"
```

```
plot(pcr)
```



```
p_pcr = predict(pcr,newdata = gss_test, newx = gss_test_x)
test_mse[3] <- mean_sum_error(p_pcr,gss_test$egalit_scale)
paste('Test MSE is',test_mse[3])
```

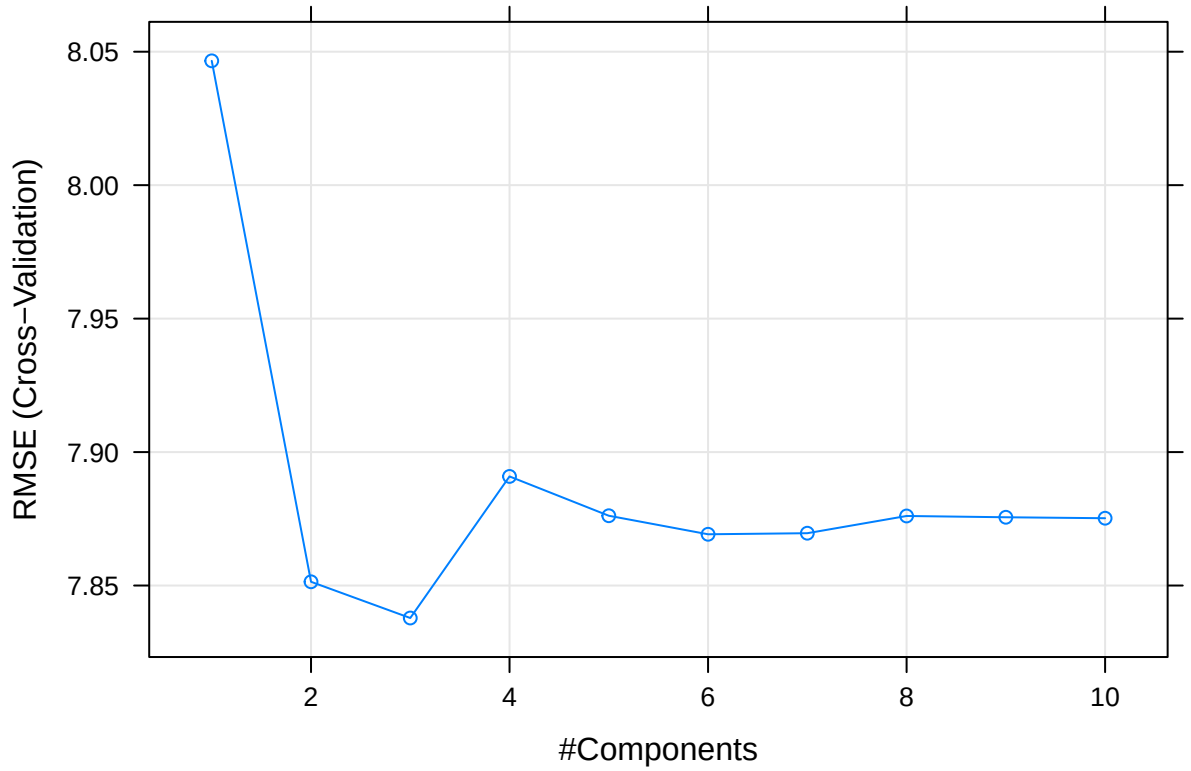
```
## [1] "Test MSE is 60.9096984324058"
```

PLS

```
set.seed(1234)
pls <- train(
  egalit_scale~., data = gss_train, method = "pls",
  scale = TRUE,
  trControl = trainControl("cv", number = 10),
  tuneLength = 10
  )
paste("Best tuned model on training set is at",pls$bestTune,'components')
```

```
## [1] "Best tuned model on training set is at 3 components"
```

```
plot(pls)
```




```
p_pls = predict(pls,newdata = gss_test, newx = gss_test_x)
test_mse[4] <- mean_sum_error(p_pls,gss_test$egalit_scale)
paste('Test MSE is',test_mse[4])
```

```
## [1] "Test MSE is 61.2819302330071"
```

Summary

Performance of each model is summarized below by their test MSEs. PCR performs the best.

```
perf_df <- data.frame(c('Linear','Elastic Net','PCR','PLS'),test_mse)
colnames(perf_df)[1] <- 'Model Type'
print(perf_df)
```

```
##    Model Type  test_mse
## 1      Linear 130.94553
## 2 Elastic Net  62.09828
## 3         PCR  60.90970
## 4         PLS  61.28193
```

Question 5

Question five

```
# Using test data accroding to Jiaxu's answer on Piaza.

predictor.lr <- Predictor$new(
  model = ls,
  data = gss_test[,-14],
  y = gss_test[14],
  predict.fun = function(model, newdata) {return(predict(model, newdata = newdata))}
)

predictor.net <- Predictor$new(
  model = fit,
  data = gss_test[,-14],
  y = gss_test[14],
  predict.fun = function(model, newdata) {return(predict(model, newx = as.matrix(newdata)))}
)

predictor.pcr <- Predictor$new(
  model = pcr$finalModel,
  data = gss_test[,-14],
  y = gss_test[14],
  predict.fun = function(model, newdata) {return(predict(model, newdata = newdata, comps = 8))}
)

predictor.pls <- Predictor$new(
  model = pls$finalModel,
  data = gss_test[,-14],
  y = gss_test[14],
  predict.fun = function(model, newdata) {return(predict(model, newdata = newdata, comps = 6))}
)
```

Linear and Elastic Net

```
interact.lr <- Interaction$new(predictor.lr) %>%
  plot() +
```

```
  ggtitle("LR") +
  theme_minimal(base_size = 12)

interact.ela <- Interaction$new(predictor.net) %>%
   plot() +
   ggtitle("Elastic Net") +
   theme_minimal(base_size = 12)

interact.pcr <- Interaction$new(predictor.pcr) %>%
  plot() +
  ggtitle("PCR") +
  theme_minimal(base_size = 12)

interact.pls <- Interaction$new(predictor.pls) %>%
  plot() +
  ggtitle("PLS") +
  theme_minimal(base_size = 12)

grid.arrange(interact.lr, interact.ela, ncol = 2)
```

As can be (hardly) seen from the plots, the LR model assigns considerable strength to many features, while Elastic Net model assigns weight to only 12 features, with their strengths varies drastically. Compared to the linear regression model, elastic net chooses the most representative feature from each bigger field. For example, the elastic net assigns 'polviews', the general political view, a very high strength, while the linear regression model uses multiple political features, like spending on liberal activities, reading news, etc. Besides, the LR model yields a much weaker interaction between political features compared with the elastic net.

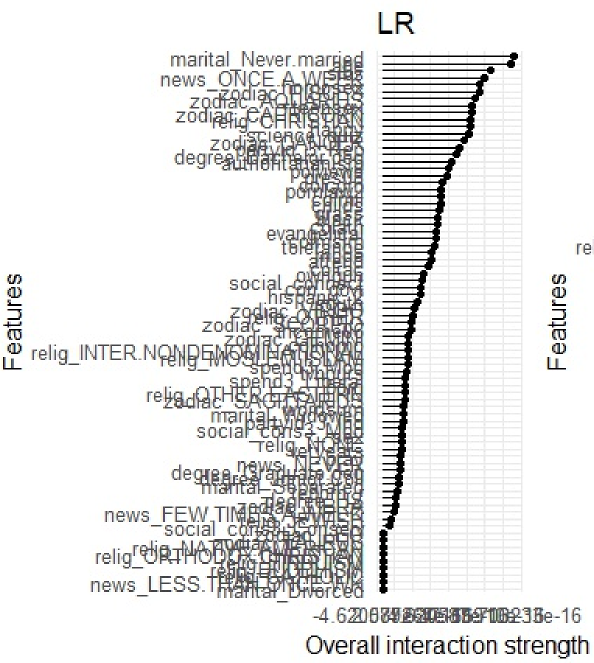
PCR and PLS

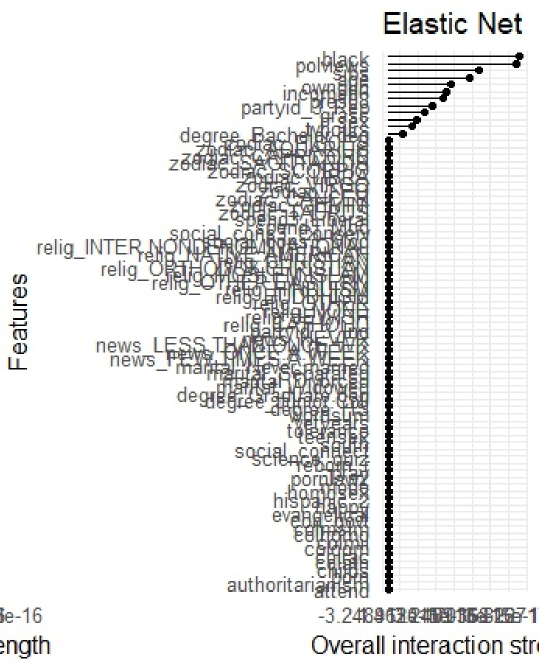
```
grid.arrange(interact.pcr, interact.pls, ncol = 2)
```

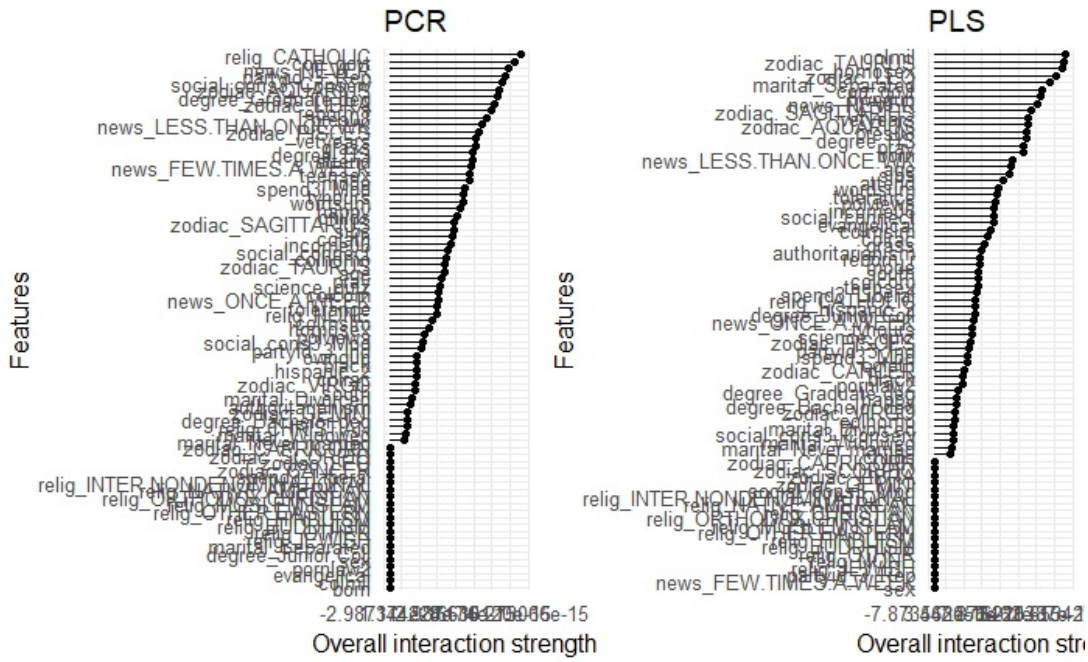
Compared to the former two models, PCR and PLS yield very different assignment on feature strengths, and differs a lot from each other. PCR model assigns the most strength to a religious feature, and relatively downplays political and racial features. PLS assigns most strength to colmil,allowing militarist to teach in college, which is rather surprising since is considered not very related to egalitarianism.

Overall speaking, the four models yield quite different results, not only in the strength of each bigger field, but also the variables that best represents each field. This highlights how model choice will influence the results of our research.

Help on making plots more readable are highly appreciated. My codes for question 5 run normally in Rstudio, but can not be knitted to a pdf. Therefore I attached plots produced in Rstudio in this pdf file.


LR
Features
marital_Never.married
evangelical
tolerance
Overall interaction strength


Elastic Net
Features
black
authoritarianism
attend
Overall interaction str


PCR
Features
Overall interaction strength
PLS
Features
Overall interaction str