Homework 4

This homework explores the use of Lasso for ad targeting using experimental data.

1. Simple regressions

1.1 Regression without controls

QUESTION: Load ad_heterog data and regress revenue on treatment without further controls. The data is from an A/B test. Interpret the intercept and the treatment coefficient.

```
library(readr)
ad_heterog = read.csv('ad_heterog.csv')
set.seed(34051)

summary(lm(revenue ~ treatment, data = ad_heterog))
```

```
##
## Call:
## lm(formula = revenue ~ treatment, data = ad_heterog)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.92105 -0.35544 -0.00554  0.33397  1.60933
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.10818    0.01617  315.94   <2e-16 ***
## treatment    0.65076    0.02331   27.91   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5209 on 1998 degrees of freedom
## Multiple R-squared:  0.2806, Adjusted R-squared:  0.2802
## F-statistic: 779.2 on 1 and 1998 DF,  p-value: < 2.2e-16
```

ANSWER: The intercept implies that the control group exhibits an average revenue of 5.10818. The treatment coefficient implies that the treatment corresponds to an increase in revenue of 0.65076.

1.2

QUESTION: Assume that it costs 0.7 Dollars to show the ad. Based on the previous regression, should you show the ad (assuming you either show it to all consumers or to nobody, i.e. you are not able to target the ad)?

ANSWER: Based on the previous regression alone and with no targeting ability, I would choose not to show the ad. This is because the previous regression shows that the treatment corresponds to an increase in revenue of 0.65076, which is less than 0.7.

2. Lasso with interactions

2.1

QUESTION: Run the code below in order to generate a matrix of demographic variables as well as a matrix of interaction terms.

```
# extract columns pertaining to demographic information (all columns except first two)
demo_matrix <- ad_heterog[-c(1,2)]

# generate interactions of each demographic variable with the treatment variable
demo_treat_matrix <- demo_matrix * matrix(ad_heterog$treatment,nrow=2000,ncol=30)
names(demo_treat_matrix) <- gsub("demographic","treat_demo", names(demo_treat_matrix))

# generate treatment variable that is outside of data-frame (useful below)
treatment <- ad_heterog$treatment
```

2.2

QUESTION: Run a lasso regression (not cross-validated yet) without standardization using treatment and the interaction term matrix as X variables (note that we are NOT also using the demographic variables as controls here). Plot how the coefficients behave when changing the penalty parameter. Why do you think the first line is non-monotonic (i.e. it first increases and then decreases)?

```
library(glmnet)
```

```
## Loading required package: Matrix
```

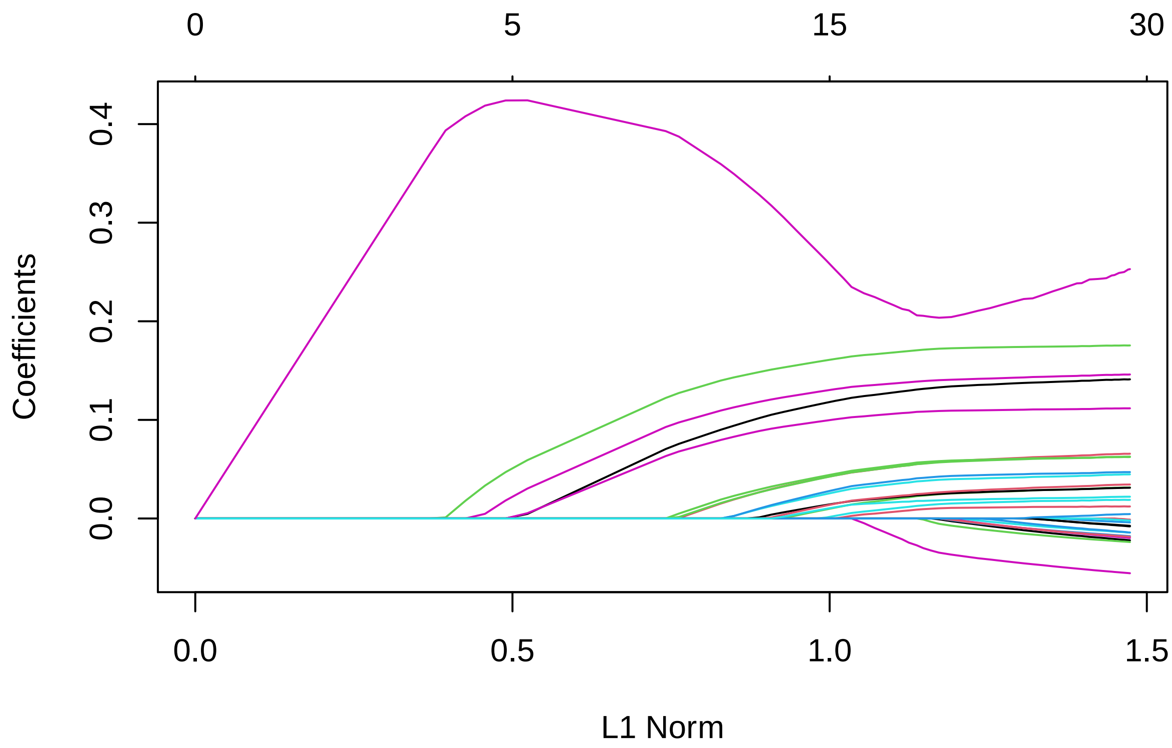
```
## Loaded glmnet 4.1-1
```

```
# X variables to try for lasso
demo_treat_matrix$treatment = treatment
X = sparse.model.matrix(~ ., data = demo_treat_matrix)

# Y variable for lasso
Y = ad_heterog$revenue

# simple version of lasso (with default parameters)
lasso = glmnet(X, Y, standardization = FALSE)

# plot as a function of lambda
plot(lasso)
```



ANSWER: I think the line increases at first since the decreasing lambda allows for a higher coefficient. I think the line begins to decrease as lambda decreases further since more coefficients become nonzero, which allows for these other variables to act as controls, thus decreasing the magnitude of the coefficient represented by the first line. Additionally, since the other variables are interaction terms that include `treatment`, we may also be re-attributing the effect of the treatment across certain demographic groups.

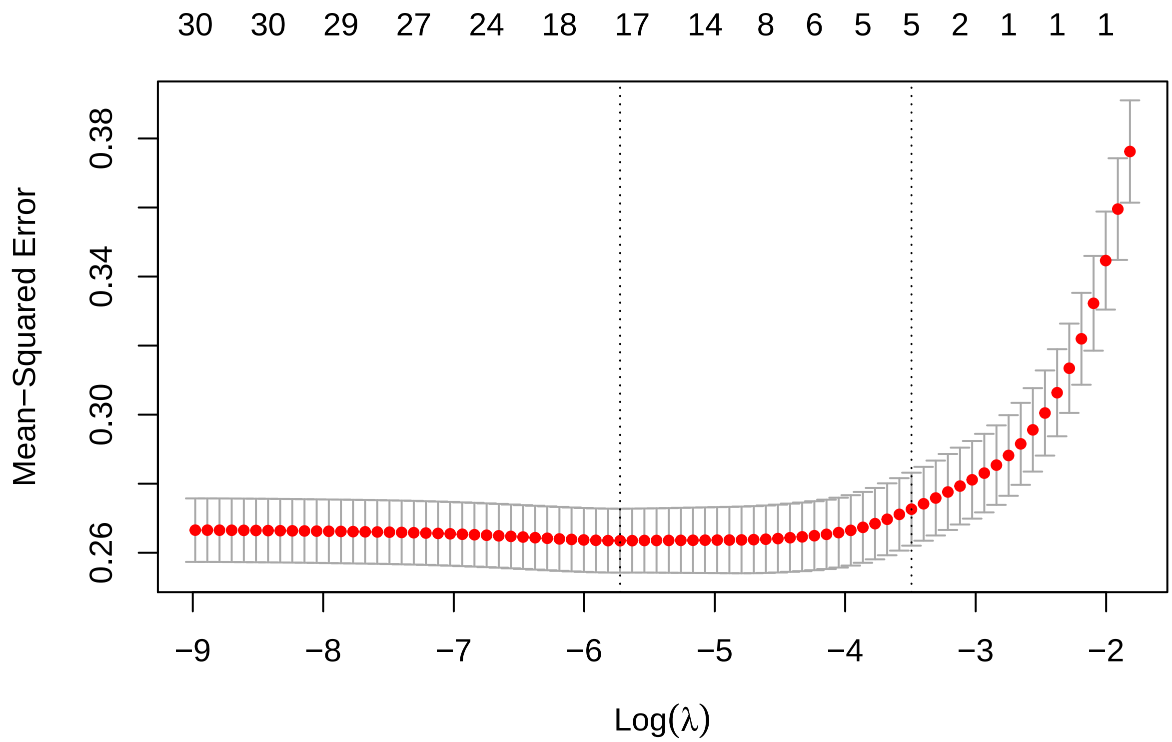
2.3

QUESTION: Run the cross-validated lasso (without standardization) based on the same set of variables. Report the coefficient values for all non-zero coefficients at the optimal penalty value. What do the results suggest regarding the scope for targeting?

```
# X variables to try for lasso
X = sparse.model.matrix(~ ., data = demo_treat_matrix)

# cross-validated lasso
cv.lasso = cv.glmnet(X, Y, standardize = FALSE)

# plot as a function of lambda
plot(cv.lasso)
```




```r
# getting the coefficients and picking those that are not zero
coefficients = coef(cv.lasso, s = 'lambda.min')
coeffnames = rownames(coefficients)[which(coefficients != 0)]
coeffvalues = coefficients[which(coefficients != 0)]

# printing non-zero coefficients
print(cbind(coeffnames, coeffvalues))
```

```
##       coeffnames      coeffvalues
##  [1,] "(Intercept)"   "5.11447199593345"
##  [2,] "treat_demo_1"  "0.12310006321849"
##  [3,] "treat_demo_2"  "0.00305308381459801"
##  [4,] "treat_demo_3"  "0.164823779971674"
##  [5,] "treat_demo_7"  "0.133908400109947"
##  [6,] "treat_demo_9"  "0.0480728380491637"
##  [7,] "treat_demo_12" "0.00580911201007424"
##  [8,] "treat_demo_16" "0.0149743572682944"
##  [9,] "treat_demo_18" "0.0303419378659088"
## [10,] "treat_demo_19" "-0.0131582427491033"
## [11,] "treat_demo_20" "0.0182192828131762"
## [12,] "treat_demo_22" "0.0491796806168972"
## [13,] "treat_demo_23" "0.0329473363298267"
## [14,] "treat_demo_25" "0.103367823701994"
## [15,] "treat_demo_27" "0.0182424047740383"
## [16,] "treat_demo_28" "0.0473673321254911"
## [17,] "treat_demo_30" "0.0134205054316107"
```

```
## [18,] "treatment"     "0.243810974516436"
```

ANSWER: The coefficients of the retained variables can be found above.

The results seem to suggest that there may be a substantial degree of targeting that is worth undertaking since, for example, demographic groups 1, 3, 7, and 25 appear to be especially responsive to the treatment.

3. Lasso with un-interacted and interacted demographics

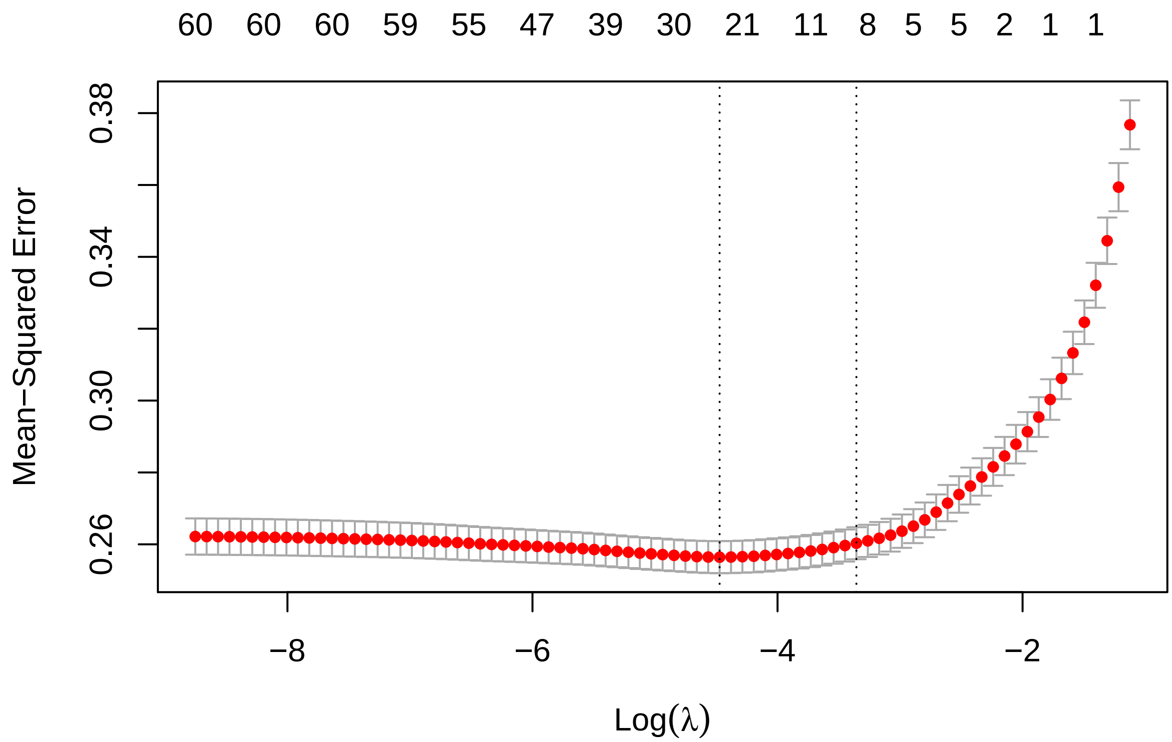
3.1

QUESTION: Run a cross-validated lasso based on the same variables as above, but now also include the un-interacted demographic variables to the matrix of X variables to try for lasso. Make sure you know how to interpret the coefficients for the demographics that are selected as both slope and intercept dummies as opposed to the demographics that only show up as interaction dummies. For example, consider the case of demographics 3 and 7. Can you compute the treatment effect for both characteristics (i.e., the effect of the treatment on individuals with a certain characteristic vs. individuals with the same characteristic in the control group?).

```r
# X variables to try for lasso
conc_df = cbind(demo_matrix, demo_treat_matrix)
X = sparse.model.matrix(~ ., data = conc_df)

# cross-validated lasso (with default parameters)
cv.lasso = cv.glmnet(X, Y)

# plot as a function of lambda
plot(cv.lasso)
```




```r
# getting the coefficients and picking those that are not zero
coefficients = coef(cv.lasso, s = 'lambda.min')
coeffnames = rownames(coefficients)[which(coefficients != 0)]
coeffvalues = coefficients[which(coefficients != 0)]

# printing non-zero coefficients
print(cbind(coeffnames, coeffvalues))
```

```
##       coeffnames       coeffvalues
##  [1,] "(Intercept)"    "4.9827718015203"
##  [2,] "demographic_1"  "0.140400060013595"
##  [3,] "demographic_2"  "-0.00740874603487904"
##  [4,] "demographic_7"  "0.0117304884479961"
##  [5,] "demographic_9"  "0.0294838665069139"
##  [6,] "demographic_11" "-0.0165984496417829"
##  [7,] "demographic_13" "-0.00030376961957279"
##  [8,] "demographic_22" "0.0267357057733842"
##  [9,] "demographic_23" "0.0154394187027486"
## [10,] "demographic_25" "0.0632396732416622"
## [11,] "demographic_28" "0.00772004586094798"
## [12,] "treat_demo_2"   "0.00626220732853982"
## [13,] "treat_demo_3"   "0.16246535880582"
## [14,] "treat_demo_7"   "0.119179293028367"
## [15,] "treat_demo_9"   "0.0154906865053377"
## [16,] "treat_demo_12"  "0.00348891791013666"
## [17,] "treat_demo_16"  "0.0110701874116298"
```

```
## [18,] "treat_demo_18"  "0.0271215604810528"
## [19,] "treat_demo_20"  "0.0158548160944118"
## [20,] "treat_demo_22"  "0.0192245645406146"
## [21,] "treat_demo_23"  "0.0157608153332456"
## [22,] "treat_demo_25"  "0.0369327056221963"
## [23,] "treat_demo_27"  "0.0157972635817982"
## [24,] "treat_demo_28"  "0.0365841027988656"
## [25,] "treat_demo_30"  "0.0119082868607031"
## [26,] "treatment"      "0.383462868365683"
```

ANSWER: We can see from the variables that were retained in the lasso regression that `demographic_3` was not included while its corresponding interaction term (`treat_demo_3`) was, preventing us from being able to calculate the treatment effect in that case. However, both terms (`demographic_7` and `treat_demo_7`) were included for the 7th demographic variables. Based on the coefficients, we can see that individuals with the 7th demographic characteristic had a revenue that is roughly 0.012 higher than those without the characteristic on average, while those with both the 7th demographic characteristic and exposure to the treatment had a revenue that is roughly 0.12 higher than those without either characteristic on average. This suggests a treatment effect of roughly 0.12 - 0.012 = 0.108 for individuals with the 7th demographic characteristic.

3.2

QUESTION: The code below computes predicted revenue without or with treatment. Use those two predictions to compute the expected profit per consumer when showing the ad to everybody versus when showing the ad only to consumers with positive expected profit.

```
# drop previously added treatment column from demo_treat_matrix in exercise 2.2
demo_treat_matrix = subset(demo_treat_matrix, select = -c(treatment))
```

```
# X matrix based on all treated
treat_always = ad_heterog$treatment>=0  # generates column of ones
demo_treat_always <- demo_matrix * matrix(treat_always,nrow=2000,ncol=30)
names(demo_treat_always) <- gsub("demographic","treat_demo", names(demo_treat_matrix) )
X_always <- sparse.model.matrix(~ .,cbind(treat_always,demo_matrix,demo_treat_always))

# X matrix based on nobody is treated
treat_never = ad_heterog$treatment<0  # generates column of zeros
demo_treat_never <- demo_matrix * matrix(treat_never,nrow=2000,ncol=30)
names(demo_treat_never) <- gsub("demographic","treat_demo", names(demo_treat_matrix) )
X_never <- sparse.model.matrix(~ .,cbind(treat_never,demo_matrix,demo_treat_never))

# compute revenue at baseline (no treatment) and under treatment for all consumers
ad_heterog$treatment_rev <- predict(cv.lasso,newx = X_always, s = "lambda.min")
ad_heterog$baseline_rev <- predict(cv.lasso,newx = X_never, s = "lambda.min")
```

```
# indicator for who is shown the ad
ad_heterog$show_ad = ifelse(ad_heterog$treatment_rev - ad_heterog$baseline_rev > 0.7, 1, 0)
```

```
# expected incremental profit with and without targeting
rev_treat = sum(ad_heterog$treatment_rev)
rev_no_treat = sum(ad_heterog$baseline_rev)
rev_increase = rev_treat - rev_no_treat
sprintf('Incremental revenue with no targeting: %s', rev_increase)
```

```
## [1] "Incremental revenue with no targeting: 1157.37895007585"
```

```
df_target = ad_heterog[ad_heterog$show_ad == 1,]
rev_treat_target = sum(df_target$treatment_rev)
```

```r
rev_no_treat_target = sum(df_target$baseline_rev)
rev_increase_target = rev_treat_target - rev_no_treat_target
sprintf('Incremental revenue with targeting: %s', rev_increase_target)
```

```
## [1] "Incremental revenue with targeting: 604.280635188136"
```

```r
# expected profit per consumer

# average revenue increase without targeting
roi_per_person = rev_increase / nrow(ad_heterog)
# average profit per person without targeting
cost_per_person = 0.70
profit_per_person = roi_per_person - cost_per_person
sprintf('Profit per consumer with no targeting: %s', profit_per_person)
```

```
## [1] "Profit per consumer with no targeting: -0.121310524962074"
```

```r
# average_revenue increase with targeting
roi_per_person_target = rev_increase_target / nrow(df_target)
profit_per_person_target = roi_per_person_target - cost_per_person
sprintf('Profit per targeted consumer: %s', profit_per_person_target)
```

```
## [1] "Profit per targeted consumer: 0.122150524065491"
```

ANSWER: We can see that showing everyone the ad leads to a higher incremental revenue, as expected. However, taking into account the fact that the ad costs $0.70 to display, we see that showing everyone the ad leads to a negative profit on average, while targeting only those with a positive expected profit leads to a positive profit on average. This demonstrates the benefit of targeted ads.