Homework 1

MA 590 Special Topics: Causal Inference

Aukkawut Ammartayakun

18 January, 2023

The goal here is to get familiar with R: working with data and running regressions. First step: download the "acsMASS" dataset from this link and save it in the same folder as this file.

Then load it into R like so:

```
acs <- read.csv('acsMASS.csv')
```

Alternatively, you can save the file anywhere you want and put the entire filepath into `read.csv`.

This is a subset of the American Community Survey consisting of people from Massachusetts.

Here's what the first few rows look like:

```
head(acs)
```

```
##   index groupQuarters   metroArea raceEthnicity  citizenship hcovany         educ
## 1     1          TRUE      Boston         white Born Citizen       1 Some College
## 2     2          TRUE Springfield        Latinx Born Citizen       1 Some College
## 3     3          TRUE      Boston         white Born Citizen       1    Bachelors
## 4     4          TRUE      Boston         white Born Citizen       1 Some College
## 5     5          TRUE      Boston         white Born Citizen       1        no HS
## 6     6          TRUE      Boston         white Born Citizen       1           HS
##           employment  deaf   vet age income hrsWork
## 1 Not in labor force FALSE FALSE  77      0       0
## 2           Employed FALSE FALSE  18   3000       7
## 3           Employed FALSE FALSE  28  35000      60
## 4 Not in labor force FALSE FALSE  22   4000      25
## 5 Not in labor force FALSE FALSE  50  30000       0
## 6 Not in labor force FALSE FALSE  19  12000       0
```

Working with datasets

One way to see what's in a dataset is the `summary()` command:

```
summary(acs)
```

```
##      index       groupQuarters    metroArea         raceEthnicity      citizenship
##  Min.   :    1   Mode :logical   Length:70131       Length:70131       Length:70131
##  1st Qu.:17534   FALSE:64897     Class :character   Class :character   Class :character
##  Median :35066   TRUE :5234      Mode  :character   Mode  :character   Mode  :character
##  Mean   :35066
##  3rd Qu.:52598
##  Max.   :70131
##
##     hcovany           educ            employment          deaf            vet
```

```
##  Min.   :0.0000   Length:70131       Length:70131       Mode :logical   Mode :logical
##  1st Qu.:1.0000   Class :character   Class :character   FALSE:67533     FALSE:66721
##  Median :1.0000   Mode  :character   Mode  :character   TRUE :2598      TRUE :3410
##  Mean   :0.9765
##  3rd Qu.:1.0000
##  Max.   :1.0000
##
##       age            income           hrsWork
##  Min.   : 0.00   Min.   :  -6100   Min.   : 0.0
##  1st Qu.:22.00   1st Qu.:   9900   1st Qu.: 0.0
##  Median :42.00   Median :  31000   Median :20.0
##  Mean   :41.92   Mean   :  53022   Mean   :21.5
##  3rd Qu.:61.00   3rd Qu.:  68000   3rd Qu.:40.0
##  Max.   :95.00   Max.   :1353400   Max.   :99.0
##                  NA's   :10172
```

There are basically two ways of accessing elements of a dataframe (as R refers to datasets). 1. Treating them like a matrix

```
acs[3,2] ## gives the 2nd element of the 3rd row
```

```
## [1] TRUE
```

```
acs[3,] ## gives the entire third row
```

```
##   index groupQuarters metroArea raceEthnicity  citizenship hcovany      educ employment  deaf
## 3     3          TRUE    Boston         white Born Citizen       1 Bachelors   Employed FALSE
##     vet age income hrsWork
## 3 FALSE  28  35000      60
```

```
age <- acs[,11] ## gives the 11th column
head(age)
```

```
## [1] 77 18 28 22 50 19
```

(If you just call a command, R will print out the result to the console. If you assign the result to some character string using <- or =–as above–then you create a named object that you can access later.)

2. Using the $

```
Age = acs$age
```

Test equality using a double equal sign

```
Age[1]==age[1]
```

```
## [1] TRUE
```

```
all(Age==age)
```

```
## [1] TRUE
```

To subset, use the subset() command

```
worcester <- subset(acs,metroArea=="Worcester")
head(worcester)
```

```
##    index groupQuarters metroArea raceEthnicity  citizenship hcovany         educ
## 10    10          TRUE Worcester         white Born Citizen       1   Associates
## 17    17          TRUE Worcester         white Born Citizen       1 Some College
## 18    18          TRUE Worcester      Black-AA Born Citizen       1 Some College
```

```
## 20    20          TRUE Worcester        Latinx Born Citizen       1 Some College
## 28    28          TRUE Worcester         white Born Citizen       1            HS
## 47    47          TRUE Worcester      Black-AA Born Citizen       1         no HS
##            employment  deaf   vet age income hrsWork
## 10 Not in labor force FALSE FALSE  85   4800       0
## 17 Not in labor force FALSE FALSE  19      0       0
## 18 Not in labor force FALSE FALSE  75  28500      10
## 20 Not in labor force FALSE FALSE  19   1000      20
## 28 Not in labor force FALSE FALSE  89  30600       0
## 47           Employed FALSE FALSE  52  20000      30
```

```
minor <- subset(acs,age<18)
head(minor)
```

```
##     index groupQuarters  metroArea raceEthnicity  citizenship hcovany         educ
## 26     26          TRUE Pittsfield         white Born Citizen       1        no HS
## 55     55          TRUE  Worcester      Black-AA Born Citizen       1        no HS
## 59     59          TRUE     Boston      Black-AA Born Citizen       1        no HS
## 77     77          TRUE     Boston         Asian  Not Citizen       1 Some College
## 205   205          TRUE  Worcester        Latinx Born Citizen       1        no HS
## 232   232          TRUE Providence      Black-AA Born Citizen       1        no HS
##             employment  deaf   vet age income hrsWork
## 26  Not in labor force FALSE FALSE  16      0       0
## 55  Not in labor force FALSE FALSE  16      0       0
## 59  Not in labor force FALSE FALSE  16      0       0
## 77            Employed FALSE FALSE  17   2500       6
## 205               <NA> FALSE FALSE  12     NA       0
## 232               <NA> FALSE FALSE  14     NA       0
```

Problem 1

a) Create a subset of `acs` consisting of only people with no high school diploma (see the variable `educ`). Using the `nrow` command, how many rows does it have?

```
## put the code for your answer here
nhsp <- subset(acs,educ=="no HS")
nrow(nhsp)
```

```
## [1] 17437
```

b) How many people in the dataset have an income greater than or equal to one million dollars? (hint: make a subset of the data and use `nrow`... or do it another way)

```
## put the code for your answer here
filthyrich <- subset(acs,income>=1000000)
nrow(filthyrich)
```

```
## [1] 13
```

Simple functions

`sum()`, `mean()`, `sd()`, `var()` do what you think they do. However, if there are NAs, they will return NA. To have them ignore the NA values, include the optional argument `na.rm=TRUE`.

```
mean(acs$income)
```

```
## [1] NA
```

```
mean(acs$income, na.rm=TRUE)
```

```
## [1] 53021.82
```

```
sd(acs$income,na.rm=TRUE)
```

```
## [1] 78784.68
```

If a variable is logical (i.e. `TRUE` or `FALSE`) then `sum()` will give the number of `TRUE`s and `mean()` will give the proportion of `TRUE`s

```
sum(acs$deaf,na.rm=TRUE)
```

```
## [1] 2598
```

```
mean(acs$deaf,na.rm=TRUE)
```

```
## [1] 0.03704496
```

```
mean(minor$deaf,na.rm=TRUE)
```

```
## [1] 0.005468379
```

Problem 2

a) What is the mean income (excluding NAs) of people with no HS diploma?

```
## put the code for your answer here
mean(nhsp$income,na.rm=TRUE)
```

```
## [1] 13962.89
```

b) What is the standard deviation (`sd()`) of income among people whose income was at least a million? What about people whose income is less than $100,000?

```
## put the code for your answer here
print(sd(filthyrich$income,na.rm=TRUE))
```

```
## [1] 127832.6
```

```
print(sd(subset(acs,income<100000)$income,na.rm=TRUE))
```

```
## [1] 27099.53
```

Regression

Ordinary least squares linear regression uses the `lm()` command. To fit the model, say,

$$income_i = \beta_0 + \beta_1 age_i + \epsilon_i$$

run the code:

```
mod1=lm(income~age,data=acs)
summary(mod1)
```

```
##
## Call:
## lm(formula = income ~ age, data = acs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -75944  -39634  -22178   15028 1286797
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 31543.42     828.20   38.09   <2e-16 ***
## age           449.48      15.99   28.11   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 78270 on 59957 degrees of freedom
##   (10172 observations deleted due to missingness)
## Multiple R-squared:  0.01301,    Adjusted R-squared:  0.01299
## F-statistic: 790.3 on 1 and 59957 DF,  p-value: < 2.2e-16
```

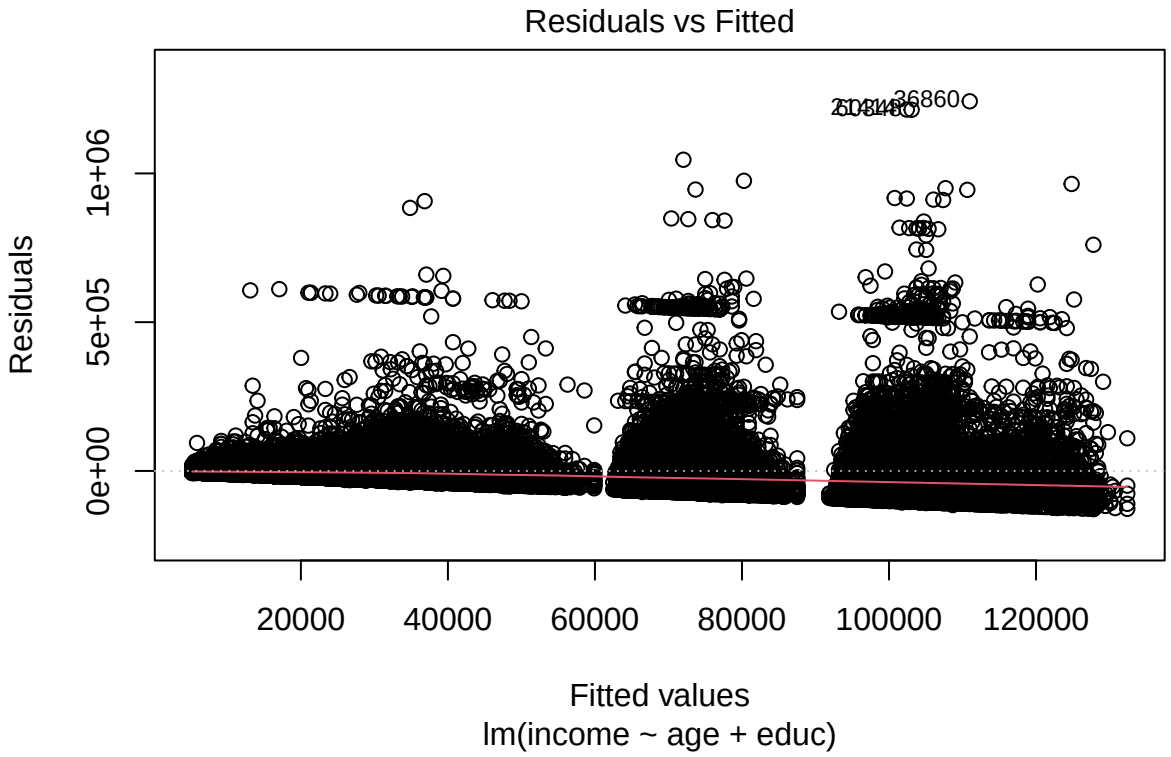
If you include a categorical predictor, R will automatically choose a reference category (the first alphabetically) and include dummy variables for the other categories:

```
mod2=lm(income~age+educ,data=acs)
summary(mod2)
```

```
##
## Call:
## lm(formula = income ~ age + educ, data = acs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -127803  -28507  -11080   11959 1242418
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       28616.13    1385.35   20.66  < 2e-16 ***
## age                 329.26      15.06   21.86  < 2e-16 ***
## educBachelors     27614.20    1296.80   21.29  < 2e-16 ***
## educDoctoral      72516.29    2113.37   34.31  < 2e-16 ***
## educHS           -15160.38    1300.88  -11.65  < 2e-16 ***
## educMasters/Prof  56683.29    1357.27   41.76  < 2e-16 ***
## educno HS        -28340.92    1427.44  -19.85  < 2e-16 ***
## educSome College -10317.01    1339.79   -7.70 1.38e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 72210 on 59951 degrees of freedom
##   (10172 observations deleted due to missingness)
## Multiple R-squared:   0.16,  Adjusted R-squared:  0.1599
## F-statistic:  1631 on 7 and 59951 DF,  p-value: < 2.2e-16
```

To see a plot of residuals versus fitted values, run:

```
plot(mod2,which=1)
```

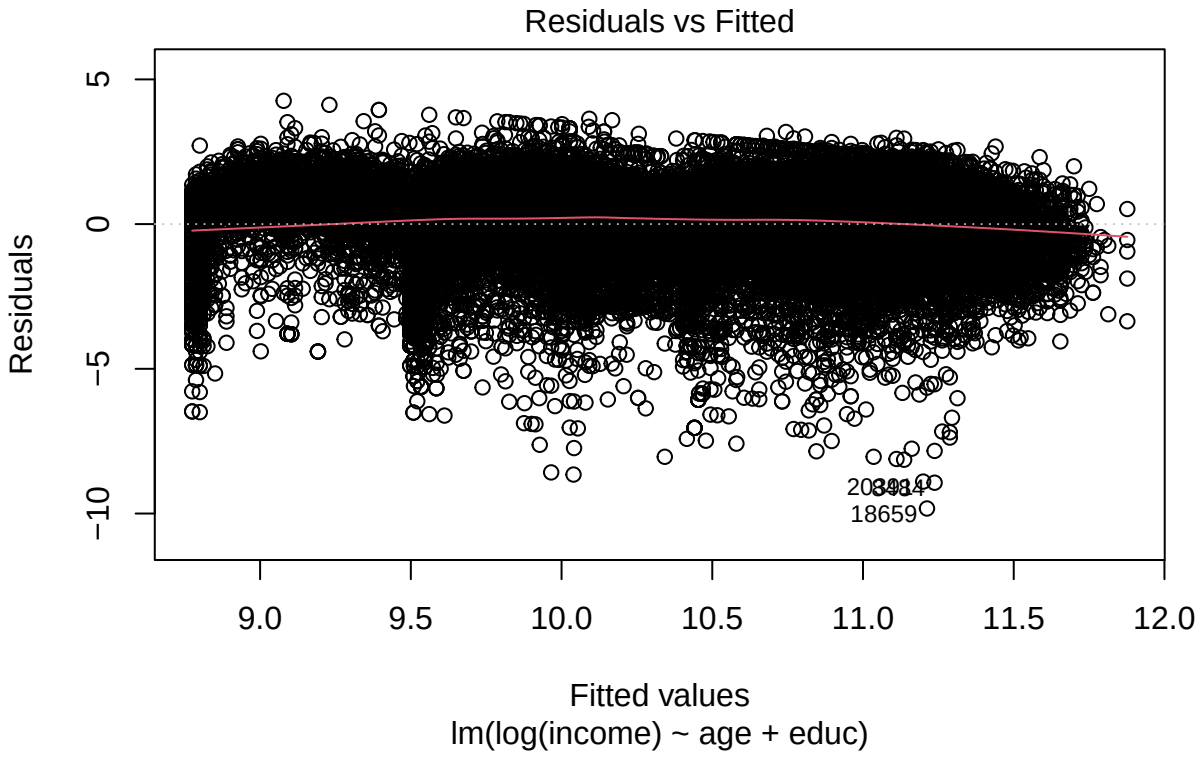


You can also include transformations:

```
mod3 <- lm(log(income)~age+educ,data=subset(acs,income>0))
summary(mod3)
```

```
##
## Call:
## lm(formula = log(income) ~ age + educ, data = subset(acs, income >
##     0))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8260 -0.5975  0.1766  0.7730  4.2594
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        9.7103306  0.0236470  410.64   <2e-16 ***
## age                0.0126428  0.0002643   47.84   <2e-16 ***
## educBachelors      0.4272005  0.0217772   19.62   <2e-16 ***
## educDoctoral       0.9653945  0.0351008   27.50   <2e-16 ***
## educHS            -0.4401164  0.0219706  -20.03   <2e-16 ***
## educMasters/Prof   0.7433631  0.0227339   32.70   <2e-16 ***
## educno HS         -1.1253178  0.0252832  -44.51   <2e-16 ***
## educSome College -0.4291401  0.0226482  -18.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.175 on 53773 degrees of freedom
## Multiple R-squared:  0.234,  Adjusted R-squared:  0.2339
## F-statistic:  2346 on 7 and 53773 DF,  p-value: < 2.2e-16
```

```
plot(mod3,which=1)
```



Problem 3

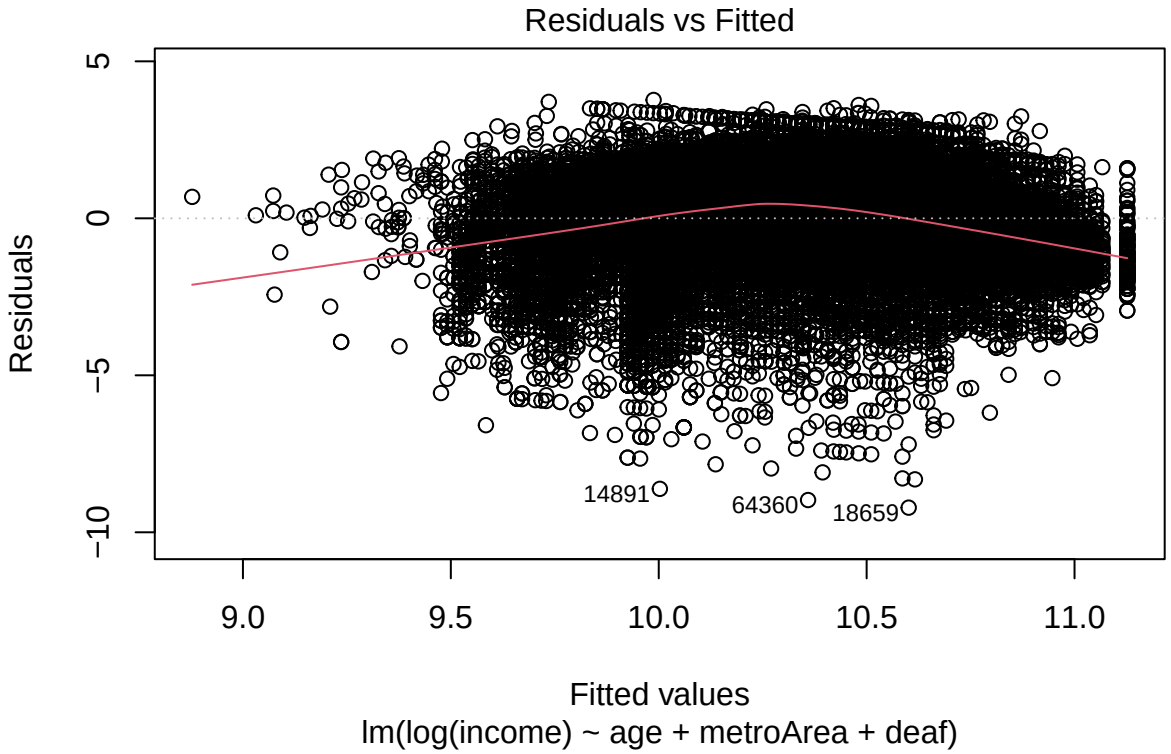
Regress log income on `age`, `metroArea`, and `deaf`, print out a regression table (`summary()`), and plot residulas versus fitted values.

```
## put the code for your answer here
mod4 <- lm(log(income)~age+metroArea+deaf,data=subset(acs,income>0))
summary(mod4)
```

```
##
## Call:
## lm(formula = log(income) ~ age + metroArea + deaf, data = subset(acs,
##     income > 0))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.2148 -0.7638  0.2075  0.9109  3.7691
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       9.5330152  0.0334335 285.134  < 2e-16 ***
## age                               0.0150208  0.0003012  49.866  < 2e-16 ***
## metroAreaBoston                   0.1668550  0.0298586   5.588 2.31e-08 ***
## metroAreanot in identifiable area -0.2819823  0.0421013  -6.698 2.14e-11 ***
## metroAreaPittsfield               -0.1739011  0.0440848  -3.945 8.00e-05 ***
## metroAreaProvidence               -0.0702265  0.0351817  -1.996   0.0459 *
## metroAreaSpringfield              -0.1471654  0.0348847  -4.219 2.46e-05 ***
## metroAreaWorcester                -0.0438700  0.0332462  -1.320   0.1870
```

```
## deafTRUE                          -0.5984169  0.0282509 -21.182  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.306 on 53772 degrees of freedom
## Multiple R-squared:  0.05377,    Adjusted R-squared:  0.05363
## F-statistic: 381.9 on 8 and 53772 DF,  p-value: < 2.2e-16
```

```
plot(mod4,which=1)
```



Logistic regression

When the dependent variable in a regression is binary, you can use logistic regression, like so:

```
logit1 <- glm(deaf~age+vet,data=acs,family=binomial(logit))
summary(logit1)
```

```
##
## Call:
## glm(formula = deaf ~ age + vet, family = binomial(logit), data = acs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1579  -0.2788  -0.1406  -0.0705   3.8236
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.309419   0.093028  -78.57   <2e-16 ***
## age          0.069178   0.001338   51.71   <2e-16 ***
## vetTRUE      0.691488   0.055793   12.39   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22223  on 70130  degrees of freedom
## Residual deviance: 17500  on 70128  degrees of freedom
## AIC: 17506
##
## Number of Fisher Scoring iterations: 7
```

To get predicted probabilities, use `predict()` function with `type="response"`

```
head(predict(logit1,type="response"))
```

```
##           1           2           3           4           5           6
## 0.121032608 0.002319205 0.004621398 0.003056280 0.020826092 0.002484910
```

Problem 4

Using logistic regression, predict whether someone is in groupQuarters as a function of their age, income, and education. Print out the regression summary. What are the first six predicted probabilities?

```
## put the code for your answer here
logit2 <- glm(groupQuarters~age+income+educ,data=acs,family=binomial(logit))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
print(summary(logit2))
```

```
##
## Call:
## glm(formula = groupQuarters ~ age + income + educ, family = binomial(logit),
##     data = acs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2257  -0.4376  -0.1913  -0.0427   8.4904
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.741e+00  1.083e-01 -16.084  < 2e-16 ***
## age              -9.832e-03  7.967e-04 -12.342  < 2e-16 ***
## income           -5.863e-05  1.302e-06 -45.027  < 2e-16 ***
## educBachelors     2.066e-01  1.156e-01   1.787 0.073862 .
## educDoctoral      2.752e-02  2.769e-01   0.099 0.920835
## educHS            1.085e+00  1.057e-01  10.264  < 2e-16 ***
## educMasters/Prof  2.098e-01  1.310e-01   1.602 0.109137
## educno HS         3.860e-01  1.098e-01   3.517 0.000436 ***
## educSome College  1.911e+00  1.049e-01  18.209  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 34733  on 59958  degrees of freedom
## Residual deviance: 25780  on 59950  degrees of freedom
##   (10172 observations deleted due to missingness)
```

```
## AIC: 25798
##
## Number of Fisher Scoring iterations: 9
```

```
head(predict(logit2,type="response"))
```

```
##         1         2         3         4         5         6
## 0.35721237 0.45431016 0.02059016 0.43015113 0.02644517 0.17555812
```

Problem 5

In an experiment with a binary outcome (i.e. $Y = 1$ or 0), write down all of the possible individual treatment effects. Show that the difference in means estimator $\bar{Y}_{Z=1} - \bar{Y}_{Z=0}$ is an unbiased estimator for the difference between the sample proportion of subjects with $\tau_i = 1$ and the sample proportion of subjects with $\tau_i = -1$.

Solution

$\tau_i = 1$ implies that $Y_i(Z_i) = 1$ and $Y_i(\neg Z_i) = 0$. That means, we want to show that $\bar{Y}_{Z=1} - \bar{Y}_{Z=0}$ is an unbiased estimator for $p^+ - p^-$ where p^+ is the sample proportion subject to positive treatment effects and p^- is the sample proportion subject to negative treatment effects.

Now, let's look at the definition for p^+ and p^-, without loss of generality, we can assume that $Z_i = 1$ and $\neg Z_i = 0$ for $i \in \mathcal{T}$ and $Z_i = 0$ and $\neg Z_i = 1$ for $i \in \mathcal{C}$:

$$p^+ = \frac{1}{n} \sum_{i \in \mathcal{T}} Z_i Y_i(1)$$

$$p^- = \frac{1}{n} \sum_{i \in \mathcal{C}} (1 - Z_i) Y_i(0)$$

This can be justified by the fact that p refers to the proportion of subjects with some treatment effect in which we have shown that only one possible τ_i can be assigned to each subject. Thus, this is similar to difference in the mean example.

Looking back to the definition of $\bar{Y}_{Z=1}$ and $\bar{Y}_{Z=0}$, we can see (shown in the lecture) that $\mathbb{E}(\bar{Y}_{Z=0}) = \mathbb{E}(Y(0)) = p^-$ and $\mathbb{E}(\bar{Y}_{Z=1}) = \mathbb{E}(Y(1)) = p^+$. Therefore, $\bar{Y}_{Z=1} - \bar{Y}_{Z=0}$ is an unbiased estimator for $p^+ - p^-$.