Understanding Why People Vote

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.4.2
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  2.0.1     v dplyr   0.7.8
## v tidyr   0.8.0     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
## Warning: package 'tibble' was built under R version 3.4.4
```

```
## Warning: package 'tidyr' was built under R version 3.4.3
```

```
## Warning: package 'purrr' was built under R version 3.4.4
```

```
## Warning: package 'dplyr' was built under R version 3.4.4
```

```
## Warning: package 'forcats' was built under R version 3.4.3
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 3.4.3
```

```
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 3.4.4
```

In August 2006 three researchers (Alan Gerber and Donald Green of Yale University, and Christopher Larimer of the University of Northern Iowa) carried out a large scale field experiment in Michigan, USA to test the hypothesis that one of the reasons people vote is social, or extrinsic, pressure. To quote the first paragraph of their 2008 research paper:

Among the most striking features of a democratic political system is the participation of millions of voters in elections. Why do large numbers of people vote, despite the fact that ... "the casting of a single vote is of no significance where there is a multitude of electors"? One hypothesis is adherence to social norms. Voting is widely regarded as a citizen duty, and citizens worry that others will think less of them if they fail to participate in elections. Voters' sense of civic duty has long been a leading explanation of vote turnout...

In this homework problem we will use both logistic regression and classification trees to analyze the data they collected.

The data The researchers grouped about 344,000 voters into different groups randomly - about 191,000 voters were a "control" group, and the rest were categorized into one of four "treatment" groups. These five groups correspond to five binary variables in the dataset.

"Civic Duty" (variable civicduty) group members were sent a letter that simply said "DO YOUR CIVIC DUTY - VOTE!"

"Hawthorne Effect" (variable hawthorne) group members were sent a letter that had the "Civic Duty" message plus the additional message "YOU ARE BEING STUDIED" and they were informed that their voting behavior would be examined by means of public records.

"Self" (variable self) group members received the "Civic Duty" message as well as the recent voting record of everyone in that household and a message stating that another message would be sent after the election with updated records.

"Neighbors" (variable neighbors) group members were given the same message as that for the "Self" group, except the message not only had the household voting records but also that of neighbors - maximizing social pressure.

"Control" (variable control) group members were not sent anything, and represented the typical voting situation.

Additional variables include **sex** (0 for male, 1 for female), **yob** (year of birth), and the dependent variable **voting** (1 if they voted, 0 otherwise).

1.1) Exploration and Logistic Regression

What proportion of people in this dataset voted in this election?

```
gerber = read.csv('gerber.csv')

head(gerber)
```

```
##   sex  yob voting hawthorne civicduty neighbors self control
## 1   0 1941      0         0         1         0    0       0
## 2   1 1947      0         0         1         0    0       0
## 3   1 1982      1         1         0         0    0       0
## 4   1 1950      1         1         0         0    0       0
## 5   0 1951      1         1         0         0    0       0
## 6   1 1959      1         0         0         0    0       1
```

```
gerber %>%
  glimpse()
```

```
## Observations: 344,084
## Variables: 8
## $ sex       <int> 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0...
## $ yob       <int> 1941, 1947, 1982, 1950, 1951, 1959, 1956, 1981, 1968...
## $ voting    <int> 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0...
## $ hawthorne <int> 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0...
## $ civicduty <int> 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ neighbors <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ self      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1...
## $ control   <int> 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0...
```

```
vote_yes = gerber %>%
  filter(voting == 1) %>%
  count()
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
vote_yes/nrow(gerber)
```

```
##           n
## 1 0.3158996
```

```
table(gerber$voting)
```

```
##
##      0      1
## 235388 108696
```

Explanation

Load the dataset into R by using the read.csv command:

gerber = read.csv("gerber.csv")

Then we can compute the percentage of people who voted by using the table function:

table(gerber$voting)

The output tells us that 235,388 people did not vote, and 108,696 people did vote. This means that 108696/(108696+235388) = 0.316 of all people voted in the election.

1.2) Exploration and Logistic Regression

Which of the four "treatment groups" had the largest percentage of people who actually voted (voting = 1)?

```
prop.table(table(subset(gerber, hawthorne == 1)$voting))
```

```
##
##         0         1
## 0.6776254 0.3223746
```

```
prop.table(table(subset(gerber, civicduty == 1)$voting))
```

```
##
##         0         1
## 0.6854623 0.3145377
```

```
prop.table(table(subset(gerber, neighbors == 1)$voting))
```

```
##
##         0         1
## 0.6220518 0.3779482
```

```
prop.table(table(subset(gerber, self == 1)$voting))
```

```
##
##         0         1
## 0.6548485 0.3451515
```

Explanation

There are several ways to get this answer. One is to use the tapply function, and compute the mean value of "voting", sorted by whether or not the people were in each group:

tapply(gerber$voting, gerber$civicduty, mean)

tapply(gerber$voting, gerber$hawthorne, mean)

tapply(gerber$voting, gerber$self, mean)

tapply(gerber$voting, gerber$neighbors, mean)

The variable with the largest value in the "1" column has the largest fraction of people voting in their group - this is the Neighbors group.

```
tapply(gerber$voting, gerber$civicduty, mean)
```

```
##         0         1
## 0.3160698 0.3145377
```

```
tapply(gerber$voting, gerber$hawthorne, mean)
```

```
##         0         1
## 0.3150909 0.3223746
```

```
tapply(gerber$voting, gerber$self, mean)
```

```
##         0         1
## 0.3122446 0.3451515
```

```
tapply(gerber$voting, gerber$neighbors, mean)
```

```
##         0         1
## 0.3081505 0.3779482
```

1.3) Exploration and Logistic Regression

Build a logistic regression model for voting using the four treatment group variables as the independent variables (civicduty, hawthorne, self, and neighbors). Use all the data to build the model (DO NOT split the data into a training set and testing set). Which of the following coefficients are significant in the logistic regression model? Select all that apply.

```
mod_1 = glm(voting ~ civicduty + hawthorne + self + neighbors,
            data = gerber,
            family = binomial)

summary(mod_1)
```

```
##
## Call:
## glm(formula = voting ~ civicduty + hawthorne + self + neighbors,
##     family = binomial, data = gerber)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9744  -0.8691  -0.8389   1.4586   1.5590
##
## Coefficients:
##              Estimate Std. Error  z value Pr(>|z|)
## (Intercept) -0.863358   0.005006 -172.459  < 2e-16 ***
## civicduty    0.084368   0.012100    6.972 3.12e-12 ***
## hawthorne    0.120477   0.012037   10.009  < 2e-16 ***
## self         0.222937   0.011867   18.786  < 2e-16 ***
## neighbors    0.365092   0.011679   31.260  < 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: 429238  on 344083  degrees of freedom
## Residual deviance: 428090  on 344079  degrees of freedom
## AIC: 428100
##
## Number of Fisher Scoring iterations: 4
```

1.4) Exploration and Logistic Regression

Using a threshold of 0.3, what is the accuracy of the logistic regression model? (When making predictions, you don't need to use the newdata argument since we didn't split our data.)

```
pred = predict(mod_1, type = 'response')
table(gerber$voting, pred > 0.3)
```

```
##
##      FALSE   TRUE
##   0 134513 100875
##   1  56730  51966
```

```
accu = ((134513 + 51966) / (134513 + 51966+ 100875 + 56730))

accu
```

```
## [1] 0.5419578
```

Explanation

First compute predictions:

predictLog = predict(LogModel, type="response")

Then, use the table function to make a confusion matrix:

table(gerber$voting, predictLog > 0.3)

We can compute the accuracy of the sum of the true positives and true negatives, divided by the sum of all numbers in the table:

(134513+51966)/(134513+100875+56730+51966) = 0.542

1.5 - Exploration and Logistic Regression

Using a threshold of 0.5, what is the accuracy of the logistic regression model?

```
pred = predict(mod_1, type = 'response')
table(gerber$voting, pred > 0.5)
```

```
##
##      FALSE
##   0 235388
##   1 108696
```

```
235388 / nrow(gerber)
```

```
## [1] 0.6841004
```

Explanation

First compute predictions:

predictLog = predict(LogModel, type="response")

Then, use the table function to make a confusion matrix:

table(gerber$voting, predictLog > 0.5)

We can compute the accuracy of the sum of the true positives and true negatives, divided by the sum of all numbers in the table:

(235388+0)/(235388+108696) = 0.684

1.6 - Exploration and Logistic Regression

Compare your previous two answers to the percentage of people who did not vote (the baseline accuracy) and compute the AUC of the model. What is happening here?

```
table(gerber$voting)
```

```
##
##      0      1
## 235388 108696
```

```
235388 / nrow(gerber)
```

```
## [1] 0.6841004
```

```
ROCR_pred = prediction(pred, gerber$voting)

auc = as.numeric(performance(ROCR_pred, "auc")@y.values)

auc
```

```
## [1] 0.5308461
```

Explanation

You can compute the AUC with the following commands (if your model's predictions are called "predictLog"):

library(ROCR)

ROCRpred = prediction(predictLog, gerber$voting)

as.numeric(performance(ROCRpred, "auc")@y.values)

Even though all of our variables are significant, our model does not improve over the baseline model of just predicting that someone will not vote, and the AUC is low. So while the treatment groups do make a difference, this is a weak predictive model.

2.1 - Trees

We will now try out trees. Build a CART tree for voting using all data and the same four treatment variables we used before. Don't set the option method="class" - we are actually going to create a regression tree here. We are interested in building a tree to explore the fraction of people who vote, or the probability of voting. We'd like CART to split our groups if they have different probabilities of voting. If we used method='class', CART would only split if one of the groups had a probability of voting above 50% and the other had a probability of voting less than 50% (since the predicted outcomes would be different). However, with regression trees, CART will split even if both groups have probability less than 50%.

```
CART_model = rpart(voting ~ civicduty + hawthorne + self + neighbors, data = gerber)
prp(CART_model)
```

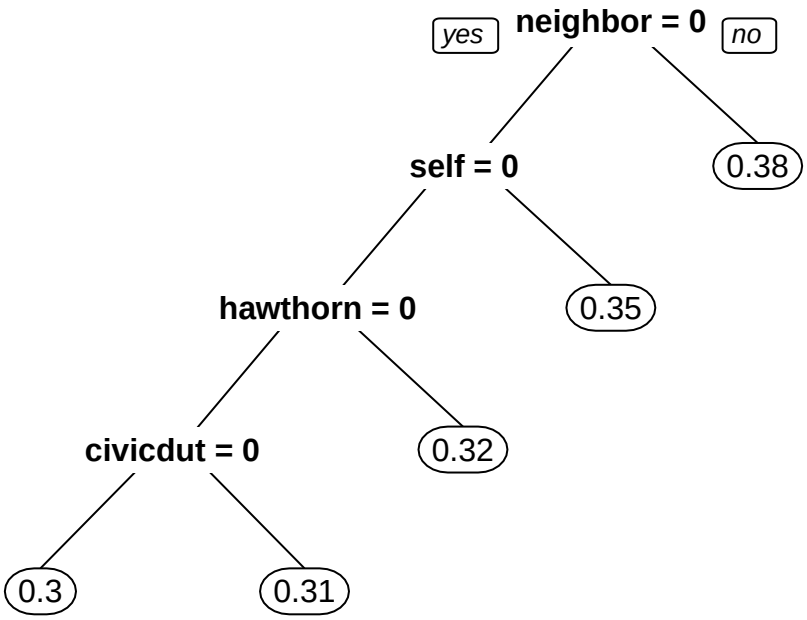
0.32

Explanation

If you plot the tree, with prp(CARTmodel), you should just see one leaf! There are no splits in the tree, because none of the variables make a big enough effect to be split on.

2.2. Trees

```
CART_model2 = rpart(voting ~ civicduty + hawthorne + self + neighbors, data = gerber, cp = 0.0)
prp(CART_model2)
```



2.3 - Trees

Using only the CART tree plot, determine what fraction (a number between 0 and 1) of "Civic Duty" people voted:

Explanation

You can find this answer by reading the tree - the people in the civic duty group correspond to the bottom right split, which has value 0.31 in the leaf.

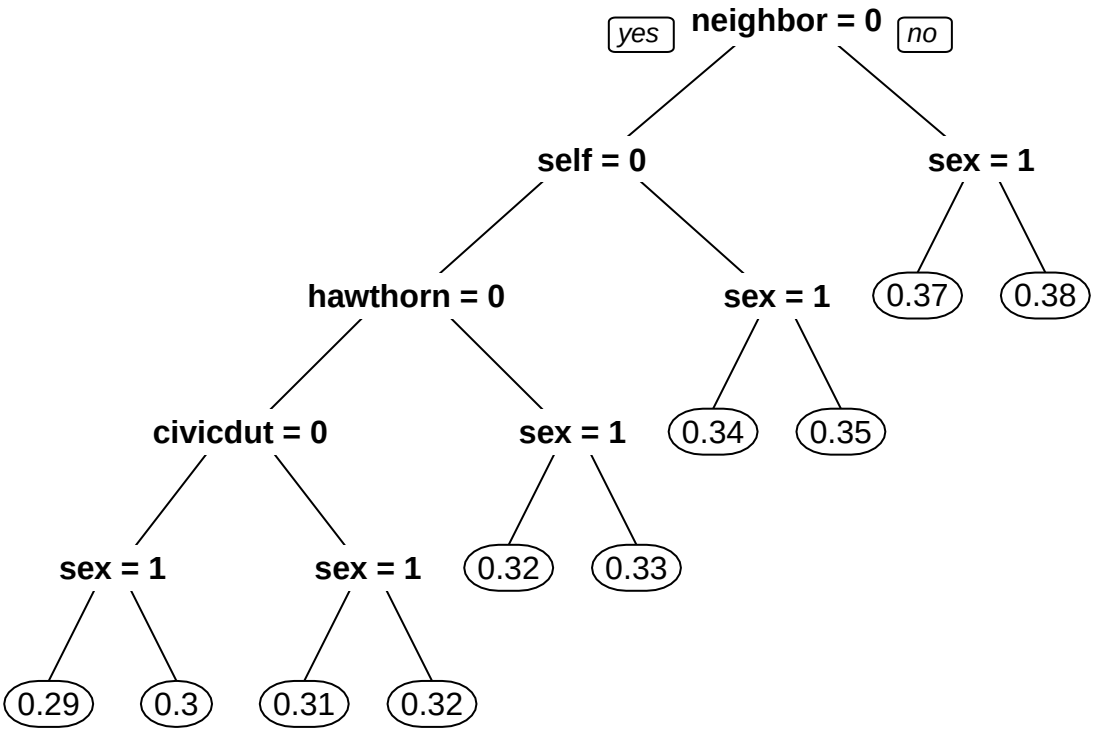
2.4 - Trees

Make a new tree that includes the "sex" variable, again with cp = 0.0. Notice that sex appears as a split that is of secondary importance to the treatment group.

In the control group, which gender is more likely to vote?

```
CART_model3 = rpart(voting ~ civicduty + hawthorne + self + neighbors + sex, data = gerber, cp = 0.0)

prp(CART_model3)
```



Explanation

You can generate the new tree using the command:

CARTmodel3 = rpart(voting ~ civicduty + hawthorne + self + neighbors + sex, data=gerber, cp=0.0)

Then, if you plot the tree with prp(CARTmodel3), you can see that there is a split on the "sex" variable after every treatment variable split. For the control group, which corresponds to the bottom left, sex = 0 (male) corresponds to a higher voting percentage.

For the civic duty group, which corresponds to the bottom right, sex = 0 (male) corresponds to a higher voting percentage.

3.1) Interaction Terms

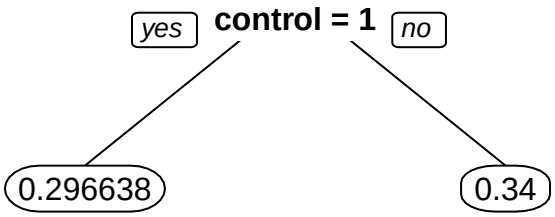
We know trees can handle "nonlinear" relationships, e.g. "in the 'Civic Duty' group and female", but as we will see in the next few questions, it is possible to do the same for logistic regression. First, let's explore what trees can tell us some more.

Let's just focus on the "Control" treatment group. Create a regression tree using just the "control" variable, then create another tree with the "control" and "sex" variables, both with cp=0.0.

In the "control" only tree, what is the absolute value of the difference in the predicted probability of voting between being in the control group versus being in a different group? You can use the absolute value function to get answer, i.e. abs(Control Prediction - Non-Control Prediction). Add the argument "digits = 6" to the prp command to get a more accurate estimate.

```
tree_reg1 = rpart(voting ~ control,
            data = gerber,
            cp = 0.0)

prp(tree_reg1, digits = 6)
```




```
abs(0.34 - 0.296638)
```

```
## [1] 0.043362
```

Explanation

You can build the two trees with the following two commands:

CARTcontrol = rpart(voting ~ control, data=gerber, cp=0.0)

CARTsex = rpart(voting ~ control + sex, data=gerber, cp=0.0)

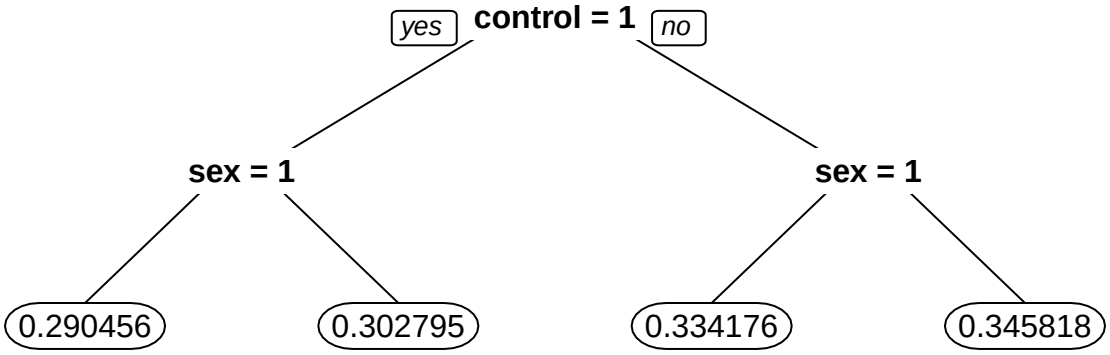
Then, plot the "control" tree with the following command:

prp(CARTcontrol, digits=6)

The split says that if control = 1, predict 0.296638, and if control = 0, predict 0.34. The absolute difference between these is 0.043362

```
tree_reg2 = rpart(voting ~ control + sex,
            data = gerber,
            cp = 0.0)

prp(tree_reg2, digits = 6)
```



3.2) Interaction Terms

Now, using the second tree (with control and sex), determine who is affected more by NOT being in the control group (being in any of the four treatment groups):

```
f = abs(0.334176 - 0.290456)

m = abs(0.302795 - 0.345818)

f - m
```

```
## [1] 0.000697
```

Explanation

You can plot the second tree using the command:

prp(CARTsex, digits=6)

The first split says that if control = 1, go left. Then, if sex = 1 (female) predict 0.290456, and if sex = 0 (male) predict 0.302795. On the other side of the tree, where control = 0, if sex = 1 (female) predict 0.334176, and if sex = 0 (male) predict 0.345818. So for women, not being in the control group increases the fraction voting by 0.04372. For men, not being in the control group increases the fraction voting by 0.04302. So men and women are affected about the same.

3.3) Interaction Terms

Going back to logistic regression now, create a model using "sex" and "control". Interpret the coefficient for "sex":

```
logreg_sex = glm(voting ~ sex, data = gerber, family = binomial)

logreg_control = glm(voting ~ control, data = gerber, family = binomial)
```

```
summary(logreg_sex)
```

```
##
## Call:
## glm(formula = voting ~ sex, family = binomial, data = gerber)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8814  -0.8814  -0.8613   1.5057   1.5307
##
## Coefficients:
##              Estimate Std. Error  z value Pr(>|z|)
## (Intercept) -0.745102   0.005157 -144.494  < 2e-16 ***
## sex         -0.055519   0.007335   -7.569 3.77e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 429238  on 344083  degrees of freedom
## Residual deviance: 429181  on 344082  degrees of freedom
## AIC: 429185
##
## Number of Fisher Scoring iterations: 4
```

Explanation

You can create the logistic regression model by using the following command:

LogModelSex = glm(voting ~ control + sex, data=gerber, family="binomial")

If you look at the summary of the model, you can see that the coefficient for the "sex" variable is -0.055791. This means that women are less likely to vote, since women have a larger value in the sex variable, and a negative coefficient means that larger values are predictive of 0.

3.4) Interaction Terms

The regression tree calculated the percentage voting exactly for every one of the four possibilities (Man, Not Control), (Man, Control), (Woman, Not Control), (Woman, Control). Logistic regression has attempted to

do the same, although it wasn't able to do as well because it can't consider exactly the joint possibility of being a woman and in the control group.

We can quantify this precisely. Create the following dataframe (this contains all of the possible values of sex and control), and evaluate your logistic regression using the predict function (where "LogModelSex" is the name of your logistic regression model that uses both control and sex):

```
logreg_sex_control = glm(voting ~ sex + control, data = gerber, family = binomial)
possibilities = data.frame(sex=c(0,0,1,1),control=c(0,1,0,1))
predict(logreg_sex_control, newdata = possibilities, type="response")
```

```
##         1         2         3         4
## 0.3462559 0.3024455 0.3337375 0.2908065
```

The four values in the results correspond to the four possibilities in the order they are stated above ((Man, Not Control), (Man, Control), (Woman, Not Control), (Woman, Control)). What is the absolute difference between the tree and the logistic regression for the (Woman, Control) case? Give an answer with five numbers after the decimal point.

```
abs(0.290456 - 0.2908065)
```

```
## [1] 0.0003505
```

3.5) Interaction Terms

So the difference is not too big for this dataset, but it is there. We're going to add a new term to our logistic regression now, that is the combination of the "sex" and "control" variables - so if this new variable is 1, that means the person is a woman AND in the control group. We can do that with the following command:

```
log_reg_sex_control_intr = glm(voting ~ sex + control + sex:control, data = gerber, family = "binomial")
summary(log_reg_sex_control_intr)
```

```
##
## Call:
## glm(formula = voting ~ sex + control + sex:control, family = "binomial",
##     data = gerber)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9213  -0.9019  -0.8284   1.4573   1.5724
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.637471   0.007603 -83.843  < 2e-16 ***
## sex         -0.051888   0.010801  -4.804 1.55e-06 ***
## control     -0.196553   0.010356 -18.980  < 2e-16 ***
## sex:control -0.007259   0.014729  -0.493    0.622
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 429238  on 344083  degrees of freedom
## Residual deviance: 428442  on 344080  degrees of freedom
## AIC: 428450
##
## Number of Fisher Scoring iterations: 4
```

How do you interpret the coefficient for the new variable in isolation? That is, how does it relate to the dependent variable?

Explanation

This coefficient is negative, so that means that a value of 1 in this variable decreases the chance of voting. This variable will have variable 1 if the person is a woman and in the control group.

3.6) Interaction Terms

Run the same code as before to calculate the average for each group:

```
possibilities = data.frame(sex=c(0,0,1,1),control=c(0,1,0,1))
predict(log_reg_sex_control_intr, newdata = possibilities, type="response")
```

```
##         1         2         3         4
## 0.3458183 0.3027947 0.3341757 0.2904558
```

Now what is the difference between the logistic regression model and the CART model for the (Woman, Control) case? Again, give your answer with five numbers after the decimal point.

```
abs(0.290456 - 0.2904558)
```

```
## [1] 2e-07
```

Explanation

The logistic regression model now predicts 0.2904558 for the (Woman, Control) case, so there is now a very small difference (practically zero) between CART and logistic regression.

3.7) Interaction Terms

This example has shown that trees can capture nonlinear relationships that logistic regression can not, but that we can get around this sometimes by using variables that are the combination of two variables. Should we always include all possible interaction terms of the independent variables when building a logistic regression model?

Explanation

We should not use all possible interaction terms in a logistic regression model due to overfitting. Even in this simple problem, we have four treatment groups and two values for sex. If we have an interaction term for every treatment variable with sex, we will double the number of variables. In smaller data sets, this could quickly lead to overfitting.