Econ 128 Take Home Final

Matt Harding
March 15, 2022

In the file econ128.csv you will find records on 23,456 households observed during the months April to August 2010 and 2011. In total you have data on 10 months over two years. Each household has a unique **hh_id**. The households are allocated into two groups **treatment** and **control**. For each household you observe the **zipcode**, whether or not they have **children**, the household size in categories **hhsize2**, …, **hhsize5plus**, income in categories **income2**,…, **income9**, and whether or not the household is a home **owner**.

The variables are as follows:

- **hh_id**: unique household id
- **year**: 2010 and 2011
- **month**: 4-8
- **zipcode**: anonymized zipcode in which home is located
- **control**: household-month is part of control group
- **treatment**: household-month is part of treatment group
- **children**: household has children
- **hhsize2-5plus**: household size
- **income2-9**: income categories <$20k, $20-30k, $30-40k, $40-50k, $50-75k, $75-100k, $100-125k, >$125k
- **owner**: resident owns home

You also observe monthly electricity consumption for each home.

- **lusage**: log(kwh) log of monthly electricity consumption
- **lusage1-6**: log(kwh) for April - September of 2009 (ie pre-sample period)

Households in the treatment group receive letters each month in 2011 encouraging them to conserve energy.

1. **Data cleaning**

Feel free to perform ANY data cleaning tasks you consider to be necessary. This may include imputing missing values or detecting and eliminating outliers. Document any choices that you make.

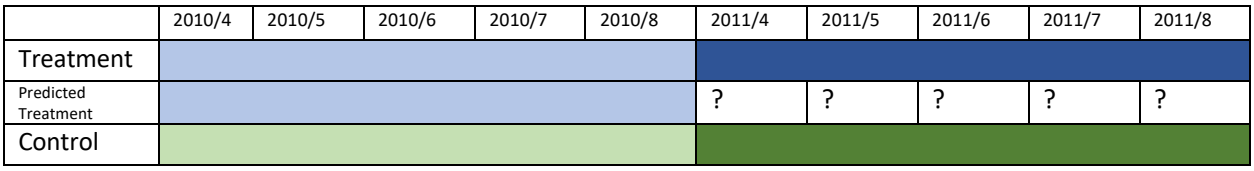
2. **Model Training**
   Using the sample of households in the **CONTROL** group **ONLY**, build a machine learning model with uses the consumption data for ***2010*** **AND** any of the other variables described above to **PREDICT** the consumption data for ***2011***.

3. **Predictions**
   Now focus on the **TREATMENT group only**. Use the model you have built above to predict the electricity consumption of the households for 2011. Note: you already have the **ACTUAL consumption** for the treatment group in 2011. I am asking you to *ignore it* and to use your model to *predict it instead*. You are making 5 predictions for each individual in the treatment group.

4. **Evaluation**
   Using the **TREATMENT group only**, *compare the predicted values for 2011 with the actual values for 2011*. Document any differences you observe between the actual values and predicted values for 2011.

5. **[Extra credit]**
   Recall that households in the treatment group were encouraged each month in 2011 to conserve energy. Discuss what the difference between your predicted values and the actual values for 2011 for the treatment group means. Explain whether or not the same quantity could have been obtained by comparing the actual valued for 2011 for the treatment group with the values for 2011 for the control group.

*Hint: Diagram illustrating the problem. You are asked to build a model using the green data. You are then asked to apply the model to the blue data to make predictions for the observations labeled with **?**. You are then asked to compare those predictions with the dark blue data.*



For full credit you need to EMAIL me your R code/jupyter notebook and prepare a short Powerpoint presentation of your findings.

ECON 128 Final

Michael Evans

2022-03-15

Econ 128 Final - Michael Evans

1. Data Cleaning

First, I set the working directory, set the seed for reproducibility, and read the .csv file:

```
setwd("C:/Users/micha/Desktop/UCI/Winter 2022/ECON 128/Final")
set.seed(9837)
dataset <- read.csv("econ128.csv")
```

And print out some preliminary information:

```
names(dataset)
```

```
##  [1] "hh_id"       "year"        "month"       "zipcode"     "control"
##  [6] "treatment"   "lusage"      "luse1"       "luse2"       "luse3"
## [11] "luse4"       "luse5"       "luse6"       "children"    "hhsize2"
## [16] "hhsize3"     "hhsize4"     "hhsize5"     "hhsize5plus" "income2"
## [21] "income3"     "income4"     "income5"     "income6"     "income7"
## [26] "income8"     "income9"     "owner"
```

```
summary(dataset)
```

```
##      hh_id            year          month      zipcode        control
##  Min.   :    3   Min.   :2010   Min.   :4   Min.   : 1.0   Min.   :0.0000
##  1st Qu.:11791   1st Qu.:2010   1st Qu.:5   1st Qu.:28.0   1st Qu.:1.0000
##  Median :23735   Median :2010   Median :6   Median :42.0   Median :1.0000
##  Mean   :23663   Mean   :2010   Mean   :6   Mean   :38.4   Mean   :0.7965
##  3rd Qu.:35527   3rd Qu.:2011   3rd Qu.:7   3rd Qu.:54.0   3rd Qu.:1.0000
##  Max.   :47356   Max.   :2011   Max.   :8   Max.   :76.0   Max.   :1.0000
##                                             NA's   :1610
##    treatment          lusage          luse1            luse2
##  Min.   :0.0000   Min.   :3.914   Min.   :-0.201   Min.   :0.055
##  1st Qu.:0.0000   1st Qu.:5.965   1st Qu.: 5.741   1st Qu.:5.742
##  Median :0.0000   Median :6.413   Median : 6.145   Median :6.158
##  Mean   :0.2034   Mean   :6.368   Mean   : 6.109   Mean   :6.111
##  3rd Qu.:0.0000   3rd Qu.:6.817   3rd Qu.: 6.518   3rd Qu.:6.523
##  Max.   :1.0000   Max.   :8.457   Max.   : 7.988   Max.   :7.883
##  NA's   :20                       NA's   :11270    NA's   :7010
```

```
##      luse3             luse4             luse5             luse6
##  Min.   :-0.134   Min.   :1.654   Min.   :1.107   Min.   :0.182
##  1st Qu.: 5.786   1st Qu.:5.987   1st Qu.:6.133   1st Qu.:5.840
##  Median : 6.208   Median :6.426   Median :6.566   Median :6.248
##  Mean   : 6.165   Mean   :6.373   Mean   :6.503   Mean   :6.198
##  3rd Qu.: 6.592   3rd Qu.:6.812   3rd Qu.:6.937   3rd Qu.:6.606
##  Max.   : 7.809   Max.   :8.027   Max.   :8.232   Max.   :7.852
##  NA's   :6320     NA's   :6350    NA's   :6330    NA's   :6250
##     children         hhsize2          hhsize3          hhsize4
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.3011   Mean   :0.3105   Mean   :0.2395   Mean   :0.1543
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  NA's   :1610
##     hhsize5          hhsize5plus          income2          income3
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.00000   Median :0.00000   Median :0.0000   Median :0.0000
##  Mean   :0.08522   Mean   :0.06173   Mean   :0.0527   Mean   :0.0937
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000
##                                      NA's   :1610     NA's   :1610
##     income4          income5          income6          income7
##  Min.   :0.000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.114   Mean   :0.1248   Mean   :0.3002   Mean   :0.1336
##  3rd Qu.:0.000   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :1.000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  NA's   :1610    NA's   :1610     NA's   :1610     NA's   :1610
##     income8          income9          owner
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:1.0000
##  Median :0.0000   Median :0.0000   Median :1.0000
##  Mean   :0.0538   Mean   :0.0305   Mean   :0.8351
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  NA's   :1610     NA's   :1610     NA's   :1610
```

But if we look more closely…

```
apply(dataset, 2, mean)
```

```
##        hh_id         year        month      zipcode      control    treatment
## 2.366252e+04 2.010500e+03 6.000000e+00           NA 7.965126e-01           NA
##       lusage        luse1        luse2        luse3        luse4        luse5
## 6.368214e+00           NA           NA           NA           NA           NA
##        luse6     children      hhsize2      hhsize3      hhsize4      hhsize5
##           NA           NA 3.105389e-01 2.395123e-01 1.543315e-01 8.522340e-02
##  hhsize5plus      income2      income3      income4      income5      income6
## 6.173261e-02           NA           NA           NA           NA           NA
```

```
##      income7      income8      income9        owner
##           NA           NA           NA           NA
```

```
apply(dataset, 2, var)
```

```
##        hh_id         year        month      zipcode      control    treatment
## 1.875843e+08 2.500011e-01 2.000009e+00           NA 1.620810e-01           NA
##       lusage        luse1        luse2        luse3        luse4        luse5
## 3.935985e-01           NA           NA           NA           NA           NA
##        luse6     children      hhsize2      hhsize3      hhsize4      hhsize5
##           NA           NA 2.141054e-01 1.821469e-01 1.305139e-01 7.796070e-02
##  hhsize5plus      income2      income3      income4      income5      income6
## 5.792194e-02           NA           NA           NA           NA           NA
##      income7      income8      income9        owner
##           NA           NA           NA           NA
```

There are a bunch of missing values for a lot of the variables! How many missing values are there, exactly?

```
sapply(dataset, function(x) sum(x=="" | is.na(x)))
```

```
##       hh_id        year       month     zipcode     control   treatment
##           0           0           0        1610           0          20
##      lusage       luse1       luse2       luse3       luse4       luse5
##           0       11270        7010        6320        6350        6330
##       luse6    children     hhsize2     hhsize3     hhsize4     hhsize5
##        6250        1610           0           0           0           0
## hhsize5plus     income2     income3     income4     income5     income6
##           0        1610        1610        1610        1610        1610
##     income7     income8     income9       owner
##        1610        1610        1610        1610
```

It seems there are a lot of missing values. If we were to just omit all of the observations with missing values, would most of our data set would be gone?

Finding the number of missing values as a percentage of the total number of observations for a given variable:

```
sapply(dataset, function(x) sum(x=="" | is.na(x)))/nrow(dataset)
```

```
##        hh_id         year        month      zipcode      control    treatment
## 0.000000e+00 0.000000e+00 0.000000e+00 6.863915e-03 0.000000e+00 8.526603e-05
##       lusage        luse1        luse2        luse3        luse4        luse5
## 0.000000e+00 4.804741e-02 2.988574e-02 2.694407e-02 2.707196e-02 2.698670e-02
##        luse6     children      hhsize2      hhsize3      hhsize4      hhsize5
## 2.664563e-02 6.863915e-03 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##  hhsize5plus      income2      income3      income4      income5      income6
## 0.000000e+00 6.863915e-03 6.863915e-03 6.863915e-03 6.863915e-03 6.863915e-03
##      income7      income8      income9        owner
## 6.863915e-03 6.863915e-03 6.863915e-03 6.863915e-03
```

It doesn't seem like there is a huge portion of data missing from any single column.

First, I will try to completely remove any observations that have missing data, and see what is left.

```
households <- na.omit(dataset)
sapply(households, function(x) sum(x=="" | is.na(x)))
```

```
##       hh_id        year       month     zipcode     control   treatment
##           0           0           0           0           0           0
##      lusage       luse1       luse2       luse3       luse4       luse5
##           0           0           0           0           0           0
##       luse6    children     hhsize2     hhsize3     hhsize4     hhsize5
##           0           0           0           0           0           0
## hhsize5plus     income2     income3     income4     income5     income6
##           0           0           0           0           0           0
##     income7     income8     income9       owner
##           0           0           0           0
```

Looks like plenty of the data remains. Taking a look at the modified data set:

```
summary(households)
```

```
##      hh_id            year          month      zipcode         control
##  Min.   :    3   Min.   :2010   Min.   :4   Min.   : 1.00   Min.   :0.0000
##  1st Qu.:11853   1st Qu.:2010   1st Qu.:5   1st Qu.:28.00   1st Qu.:1.0000
##  Median :23713   Median :2010   Median :6   Median :42.00   Median :1.0000
##  Mean   :23668   Mean   :2010   Mean   :6   Mean   :38.35   Mean   :0.7957
##  3rd Qu.:35507   3rd Qu.:2011   3rd Qu.:7   3rd Qu.:54.00   3rd Qu.:1.0000
##  Max.   :47356   Max.   :2011   Max.   :8   Max.   :76.00   Max.   :1.0000
##    treatment          lusage          luse1             luse2
##  Min.   :0.0000   Min.   :3.917   Min.   :-0.2007   Min.   :0.727
##  1st Qu.:0.0000   1st Qu.:5.967   1st Qu.: 5.7456   1st Qu.:5.745
##  Median :0.0000   Median :6.412   Median : 6.1469   Median :6.159
##  Mean   :0.2043   Mean   :6.369   Mean   : 6.1118   Mean   :6.114
##  3rd Qu.:0.0000   3rd Qu.:6.816   3rd Qu.: 6.5190   3rd Qu.:6.522
##  Max.   :1.0000   Max.   :8.318   Max.   : 7.9884   Max.   :7.883
##      luse3           luse4           luse5           luse6
##  Min.   :1.571   Min.   :1.654   Min.   :1.107   Min.   :1.184
##  1st Qu.:5.789   1st Qu.:5.990   1st Qu.:6.136   1st Qu.:5.842
##  Median :6.210   Median :6.427   Median :6.570   Median :6.249
##  Mean   :6.169   Mean   :6.376   Mean   :6.506   Mean   :6.200
##  3rd Qu.:6.592   3rd Qu.:6.813   3rd Qu.:6.937   3rd Qu.:6.607
##  Max.   :7.809   Max.   :8.027   Max.   :8.232   Max.   :7.852
##     children         hhsize2         hhsize3          hhsize4
##  Min.   :0.0000   Min.   :0.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.000   Median :0.0000   Median :0.0000
##  Mean   :0.2991   Mean   :0.313   Mean   :0.2411   Mean   :0.1562
##  3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.000   Max.   :1.0000   Max.   :1.0000
##     hhsize5         hhsize5plus         income2           income3
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.08627   Mean   :0.05548   Mean   :0.05196   Mean   :0.09295
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
```

```
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##     income4           income5           income6           income7
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.1128   Mean   :0.1232   Mean   :0.3024   Mean   :0.1361
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##     income8           income9            owner
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:1.0000
##  Median :0.0000   Median :0.00000   Median :1.0000
##  Mean   :0.0548   Mean   :0.03088   Mean   :0.8429
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.0000
```

Are there any instances where a household was labelled as both a treatment and control? Or neither?

```
sum(households$control==1 & households$treatment==1)
```

```
## [1] 0
```

```
sum(households$control==0 & households$treatment==0)
```

```
## [1] 0
```

That looks good, but what about income levels? (If I printed the whole thing there would be like 300 lines in the Knit file so just know that there are observations where no income level is recorded or multiple income levels are reported in a single month)

```
head(rowSums(households[,c("income2", "income3", "income4", "income5", "income6", "income7", "income8",
```

```
## 1 2 3 4 5 6
## 1 1 1 1 1 1
```

There are some observations where no income level is recorded. I will omit all observations where the number of recorded income levels in a single month are not 1.

```
households$incomes = rowSums(households[,c("income2", "income3", "income4", "income5", "income6", "inco
households = households[households$incomes==1, ]
```

Finally, I will do the same thing for household sizes.

After this I will delete the added variables `incomes` and `sizes`, and print a summary of the cleaned data set `households`. It has gone from $234,560$ total observations down to $173,460$; this is a significant reduction of $61,100$ observations, but I think this reduced data set is much more useful than the original.

```
households$sizes = rowSums(households[,c("hhsize2", "hhsize3", "hhsize4", "hhsize5", "hhsize5plus")])
households = households[households$sizes==1, ]
summary(households)
```

```
##      hh_id            year          month      zipcode         control
##  Min.   :    3   Min.   :2010   Min.   :4   Min.   : 1.00   Min.   :0.0000
##  1st Qu.:11955   1st Qu.:2010   1st Qu.:5   1st Qu.:28.00   1st Qu.:1.0000
##  Median :23755   Median :2010   Median :6   Median :42.00   Median :1.0000
##  Mean   :23698   Mean   :2010   Mean   :6   Mean   :37.59   Mean   :0.7889
##  3rd Qu.:35553   3rd Qu.:2011   3rd Qu.:7   3rd Qu.:54.00   3rd Qu.:1.0000
##  Max.   :47356   Max.   :2011   Max.   :8   Max.   :76.00   Max.   :1.0000
##    treatment          lusage          luse1             luse2
##  Min.   :0.0000   Min.   :3.917   Min.   :-0.2007   Min.   :0.727
##  1st Qu.:0.0000   1st Qu.:6.043   1st Qu.: 5.8208   1st Qu.:5.824
##  Median :0.0000   Median :6.472   Median : 6.1998   Median :6.215
##  Mean   :0.2111   Mean   :6.428   Mean   : 6.1658   Mean   :6.172
##  3rd Qu.:0.0000   3rd Qu.:6.861   3rd Qu.: 6.5563   3rd Qu.:6.564
##  Max.   :1.0000   Max.   :8.318   Max.   : 7.7950   Max.   :7.650
##      luse3           luse4           luse5           luse6
##  Min.   :1.602   Min.   :2.659   Min.   :1.584   Min.   :2.493
##  1st Qu.:5.872   1st Qu.:6.082   1st Qu.:6.233   1st Qu.:5.917
##  Median :6.277   Median :6.498   Median :6.640   Median :6.308
##  Mean   :6.233   Mean   :6.446   Mean   :6.577   Mean   :6.260
##  3rd Qu.:6.642   3rd Qu.:6.863   3rd Qu.:6.986   3rd Qu.:6.649
##  Max.   :7.809   Max.   :8.027   Max.   :8.232   Max.   :7.796
##     children         hhsize2          hhsize3          hhsize4
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.3351   Mean   :0.3649   Mean   :0.2828   Mean   :0.1824
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##     hhsize5        hhsize5plus         income2          income3
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.00000   Median :0.0000   Median :0.0000
##  Mean   :0.1032   Mean   :0.06676   Mean   :0.0516   Mean   :0.1014
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000
##     income4         income5          income6          income7
##  Min.   :0.000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.121   Mean   :0.1333   Mean   :0.3269   Mean   :0.1623
##  3rd Qu.:0.000   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :1.000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##     income8          income9            owner           incomes      sizes
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.0000   Min.   :1   Min.   :1
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:1.0000   1st Qu.:1   1st Qu.:1
##  Median :0.00000   Median :0.00000   Median :1.0000   Median :1   Median :1
##  Mean   :0.06607   Mean   :0.03753   Mean   :0.8985   Mean   :1   Mean   :1
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:1   3rd Qu.:1
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.0000   Max.   :1   Max.   :1
```

Now I will remove the `incomes` and `sizes` columns.

```
households$incomes <- NULL
households$sizes <- NULL
names(households)
```

```
##  [1] "hh_id"       "year"        "month"       "zipcode"     "control"
##  [6] "treatment"   "lusage"      "luse1"       "luse2"       "luse3"
## [11] "luse4"       "luse5"       "luse6"       "children"    "hhsize2"
## [16] "hhsize3"     "hhsize4"     "hhsize5"     "hhsize5plus" "income2"
## [21] "income3"     "income4"     "income5"     "income6"     "income7"
## [26] "income8"     "income9"     "owner"
```

2. Model Training

In order to train my model, I have to split the remaining data into the treatment and control groups.

```
treatment_group = households[households$treatment==1, ]
control_group = households[households$treatment==0, ]
```

Notice that even though I only use `treatment` to determine which group an observation is in, the two groups (`treatment` $= 36,610$ and `control` $= 136,850$) add up to the total number of observations in `households`.

First of all, I want to explore the data for some clues on what to look for in a model with PCA.

```
pr.out <- prcomp(households, scale = TRUE)
pr.out$center
```

```
##        hh_id         year        month      zipcode      control    treatment
## 2.369769e+04 2.010500e+03 6.000000e+00 3.759091e+01 7.889427e-01 2.110573e-01
##       lusage        luse1        luse2        luse3        luse4        luse5
## 6.428433e+00 6.165764e+00 6.172298e+00 6.233482e+00 6.445762e+00 6.577081e+00
##        luse6     children      hhsize2      hhsize3      hhsize4      hhsize5
## 6.259700e+00 3.351205e-01 3.648680e-01 2.827741e-01 1.824052e-01 1.031938e-01
##  hhsize5plus      income2      income3      income4      income5      income6
## 6.675891e-02 5.159691e-02 1.014067e-01 1.209501e-01 1.332872e-01 3.268765e-01
##      income7      income8      income9        owner
## 1.622853e-01 6.606710e-02 3.753027e-02 8.984780e-01
```

```
pr.out$scale
```

```
##        hh_id         year        month      zipcode      control    treatment
## 1.368959e+04 5.000014e-01 1.414218e+00 1.957102e+01 4.080601e-01 4.080601e-01
##       lusage        luse1        luse2        luse3        luse4        luse5
## 6.059556e-01 5.535483e-01 5.469969e-01 5.698808e-01 5.881165e-01 5.798194e-01
##        luse6     children      hhsize2      hhsize3      hhsize4      hhsize5
## 5.435399e-01 4.720339e-01 4.813945e-01 4.503489e-01 3.861792e-01 3.042127e-01
##  hhsize5plus      income2      income3      income4      income5      income6
## 2.496047e-01 2.212125e-01 3.018673e-01 3.260702e-01 3.398859e-01 4.690731e-01
##      income7      income8      income9        owner
## 3.687133e-01 2.484001e-01 1.900578e-01 3.020195e-01
```

```
pr.out$rotation
```

```
##                    PC1           PC2          PC3          PC4          PC5
## hh_id        0.001656694 -0.0051431075  0.005973222 -0.011003552  0.002857446
## year        -0.001414832 -0.0000798172 -0.001579982  0.001527099 -0.002066926
## month        0.017978823  0.0010142684  0.020077448 -0.019405450  0.026265233
## zipcode     -0.024127486  0.0038695265  0.026061066 -0.210219200  0.107499954
## control      0.011277886 -0.7066012281  0.005961742  0.013130975 -0.006630249
## treatment   -0.011277886  0.7066012281 -0.005961742 -0.013130975  0.006630249
## lusage       0.338338972  0.0035321753  0.050114277 -0.033325910  0.032652475
## luse1        0.352875758  0.0065217389  0.068336967 -0.036831072  0.039939268
## luse2        0.377169740  0.0064536549  0.070697873 -0.037452786  0.035638730
## luse3        0.383615251  0.0063106863  0.063275460 -0.026355363  0.027616273
## luse4        0.378584889  0.0054615016  0.060616449 -0.019707659  0.024184209
## luse5        0.371318025  0.0028642704  0.058334341 -0.019857982  0.026036091
## luse6        0.379133016  0.0072431861  0.062231453 -0.029970138  0.026895400
## children     0.068813503 -0.0102241517 -0.480474226 -0.147834595 -0.061793018
## hhsize2     -0.093984885  0.0086630100  0.658309555  0.206749289 -0.005622550
## hhsize3     -0.008932546 -0.0077182064 -0.307606537 -0.351803517  0.479926455
## hhsize4      0.045783571 -0.0021434573 -0.212792742  0.121148034 -0.305167520
## hhsize5      0.052708602  0.0008131874 -0.179365808  0.026519100 -0.162272478
## hhsize5plus  0.062303621 -0.0004569760 -0.166803169  0.016241834 -0.185144568
## income2     -0.050626906 -0.0044613773  0.046038134 -0.186549502 -0.045208723
## income3     -0.064120639 -0.0016536239  0.104909414 -0.387042810  0.029291236
## income4     -0.054993788 -0.0048806393  0.115991616 -0.212426014 -0.013610608
## income5     -0.026336542 -0.0080236877  0.063262230 -0.105173201 -0.093349077
## income6      0.021625846 -0.0019057257 -0.134814057  0.513129299  0.572875399
## income7      0.066090458  0.0069841999 -0.082732315  0.029423551 -0.442281835
## income8      0.052944206  0.0128805336 -0.019703084  0.039615013 -0.183267942
## income9      0.051429627  0.0048610878 -0.013363729  0.009107595 -0.119946233
## owner        0.048962379  0.0237184610 -0.226676635  0.487953805 -0.097370710
##                    PC6           PC7          PC8           PC9          PC10
## hh_id        0.0258049187 -0.0988993964  0.081207330  0.0318187810  0.078505821
## year        -0.0009091883 -0.0014639015 -0.001079536 -0.0024254755 -0.003093215
## month        0.0115534112  0.0186023691  0.013718083  0.0308214667  0.039306695
## zipcode      0.1776377919  0.0446157945  0.054900472  0.1407793273  0.377865835
## control     -0.0060067589  0.0007632981 -0.004162749  0.0001994002 -0.006992402
## treatment    0.0060067589 -0.0007632981  0.004162749 -0.0001994002  0.006992402
## lusage       0.0104000282  0.0120951169  0.007560553  0.0143888760  0.014495699
## luse1        0.0111853569  0.0048993220  0.005234007  0.0034236941  0.015538350
## luse2        0.0097144681  0.0081289155  0.004295651 -0.0002738657  0.012608447
## luse3        0.0022827867  0.0066310260  0.003115872 -0.0036356799  0.004089627
## luse4       -0.0030727377  0.0119569054  0.007424349  0.0012481269  0.007515332
## luse5       -0.0019191673  0.0145163573  0.008743996  0.0045364146  0.011084280
## luse6       -0.0006714706  0.0042672125  0.008125601 -0.0071718013  0.003002828
## children     0.2505463924 -0.1899363046 -0.019253300 -0.0220752685  0.095380964
## hhsize2      0.0147623733 -0.1010112325 -0.058664012  0.0688246007 -0.004915196
## hhsize3     -0.4676911180  0.1324630918 -0.026170694  0.0408382818 -0.151295522
## hhsize4      0.3777626873  0.6189947283  0.263278371 -0.1371290701  0.076861223
## hhsize5      0.1487697190 -0.5772440352  0.025117523  0.0840298212  0.068586849
## hhsize5plus  0.0495819727 -0.2983367316 -0.277587817 -0.0966720689  0.079945259
## income2      0.1873494699 -0.1323448660 -0.041384605  0.0215073776  0.016578659
## income3      0.3536880909  0.0686202410 -0.016309223  0.2636701668 -0.404859226
```

```
## income4     -0.1143850624  0.1292403401 -0.369178940 -0.5360644137  0.468970665
## income5     -0.2775580420 -0.1693356944  0.773028000  0.0086713969  0.233644410
## income6      0.2654818126 -0.0126848219 -0.048465214  0.0745213912  0.054612842
## income7     -0.3146145420  0.1814851231 -0.313335891  0.5617087323  0.052104173
## income8     -0.0842638765 -0.0616066148  0.041863166 -0.4619254221 -0.592956106
## income9     -0.0219508452 -0.1141088901 -0.002204545 -0.2095517704 -0.059571338
## owner       -0.3013992779  0.0227096979  0.015127247  0.0047385606 -0.004364353
##                     PC11         PC12          PC13          PC14
## hh_id        0.0018491031  0.079580804  0.1483909717 -0.1462552425
## year         0.0738756456 -0.001968045  0.0021576408 -0.0025098844
## month       -0.9387667509  0.025008718 -0.0274179861  0.0318940833
## zipcode      0.0493818065  0.082133106 -0.0739543321 -0.1190748143
## control     -0.0005467280 -0.001100872 -0.0028867641  0.0005684589
## treatment    0.0005467280  0.001100872  0.0028867641 -0.0005684589
## lusage      -0.2952453994  0.007625847 -0.0063066331  0.0046077832
## luse1        0.0607675605 -0.007882857  0.0177017321 -0.0090580615
## luse2        0.0631718275 -0.005264204  0.0156772445 -0.0080328634
## luse3        0.0581848883 -0.007034776  0.0131382001 -0.0061849386
## luse4        0.0497302747 -0.007590739  0.0052616489 -0.0080789983
## luse5        0.0441316875 -0.007235072 -0.0001333138 -0.0104737065
## luse6        0.0507925511 -0.002090381  0.0061983810 -0.0020636410
## children    -0.0125751958  0.084481791 -0.0251124741 -0.0937368231
## hhsize2      0.0017418113  0.066739626  0.0012208264 -0.0282025209
## hhsize3      0.0065100566 -0.021463687  0.0796302175  0.0739724654
## hhsize4      0.0041180258 -0.051128484  0.0531352642  0.0408157564
## hhsize5     -0.0198030533 -0.598605965 -0.1412605277 -0.0068767046
## hhsize5plus  0.0026592052  0.718681826 -0.0560708907 -0.1338399853
## income2     -0.0186329424 -0.006337198  0.8373284311  0.1755102441
## income3      0.0190839327  0.074550464 -0.4021705740 -0.0090642484
## income4     -0.0073092479 -0.203823303 -0.1628332065 -0.1163099118
## income5      0.0133831595  0.165983410 -0.0672902392 -0.0914771570
## income6      0.0094032390  0.019572689  0.0041929285 -0.0469084960
## income7     -0.0002284866 -0.071515713  0.0484293749 -0.0569393197
## income8     -0.0540553128 -0.058863692  0.0946022150 -0.3541772185
## income9      0.0278672565  0.109189590 -0.1640649394  0.8623882464
## owner       -0.0150602860  0.030023416 -0.0766333226  0.0310050796
##                     PC15          PC16         PC17          PC18         PC19
## hh_id        0.000000e+00 -9.534648e-01  0.126038176 -0.0161802526  0.021724964
## year        -9.969179e-01  9.961747e-05 -0.001789787 -0.0004913756 -0.001026276
## month       -7.845181e-02 -1.265878e-03  0.022743529  0.0062441024  0.013041297
## zipcode     -4.649059e-16  1.679381e-01  0.819645813 -0.1205344626  0.014607589
## control      0.000000e+00  3.387244e-03  0.012517474  0.0050029609 -0.004624514
## treatment    2.775558e-16 -3.387244e-03 -0.012517474 -0.0050029609  0.004624514
## lusage      -4.732326e-15  2.540855e-05 -0.006228963 -0.0072113979 -0.018697628
## luse1        1.280226e-15  1.199200e-02 -0.013036318  0.0450695639 -0.041745404
## luse2        1.249001e-15  7.540256e-03 -0.005867248  0.0316224063 -0.022030295
## luse3        9.159340e-16  2.043935e-04 -0.013227307  0.0212791390 -0.009142606
## luse4        9.714451e-16 -2.945503e-03 -0.003529394  0.0160297423  0.000557710
## luse5        8.604228e-16 -2.085262e-03  0.002549570  0.0129449129  0.004913600
## luse6        7.216450e-16  5.976872e-04 -0.007299099  0.0109143651 -0.014833578
## children     3.885781e-16  1.778506e-02 -0.173134879 -0.4226111772 -0.635007265
## hhsize2      8.326673e-17  1.305007e-02 -0.011808548 -0.2143033913 -0.361745857
## hhsize3      9.714451e-16 -3.152928e-02  0.039997920 -0.0084417999 -0.003484510
## hhsize4      1.665335e-16 -3.884608e-02 -0.037734179  0.0439455590  0.104363836
```

```
## hhsize5     -2.914335e-15  1.058347e-02  0.049909712  0.1464940353  0.205244911
## hhsize5plus  9.159340e-16  7.892025e-02 -0.051839813  0.1820078520  0.292344053
## income2      5.551115e-17  1.190217e-01  0.034981721  0.2797749059 -0.127508194
## income3      3.920475e-16 -8.616465e-02 -0.014572535  0.3885378784 -0.179393409
## income4     -1.026956e-15 -6.861794e-02 -0.111521737  0.1561466049 -0.105238477
## income5      6.106227e-16  9.254272e-02 -0.161721052  0.0222087880  0.005262603
## income6      1.484923e-15  1.303749e-02 -0.060275953 -0.1081588442  0.142658769
## income7     -1.110223e-15 -1.673029e-02 -0.001504154 -0.2424991643  0.083354710
## income8     -3.053113e-15  5.498323e-02  0.327874548 -0.2325664857  0.048584160
## income9      3.275158e-15 -1.210326e-01  0.186130550 -0.2090058088  0.027182086
## owner       -1.068590e-15 -1.215807e-02  0.294694933  0.5268199113 -0.483388358
##                      PC20          PC21          PC22          PC23
## hh_id       -0.010745684  0.0031645124 -0.0009337180 -0.0008036818
## year         0.007823873  0.0239868407  0.0030459769  0.0010886197
## month       -0.099421012 -0.3048101752 -0.0387064205 -0.0138335173
## zipcode      0.001044796  0.0041385291  0.0035040621  0.0081782734
## control     -0.002104483 -0.0005563791 -0.0002964424  0.0014528251
## treatment    0.002104483  0.0005563791  0.0002964424 -0.0014528251
## lusage       0.230172410  0.8497666048  0.1236337088  0.0447051574
## luse1       -0.683996443  0.1142487144 -0.2134090401 -0.4862378950
## luse2       -0.372642567 -0.0857666889  0.2773588586  0.2102839389
## luse3       -0.016131282 -0.2235190185  0.4981259101  0.3254930297
## luse4        0.340988749 -0.2576495787  0.2498156815 -0.2061960215
## luse5        0.454595445 -0.1989531607 -0.2427795817 -0.4944204639
## luse6        0.059612545 -0.0838026323 -0.7008244724  0.5690151227
## children     0.007012994 -0.0294721868  0.0029044579 -0.0070195050
## hhsize2      0.012180561 -0.0056645661  0.0019924017 -0.0007010844
## hhsize3      0.001378667 -0.0023105962  0.0014882407 -0.0007107355
## hhsize4     -0.009777921  0.0033600086 -0.0023255976  0.0023404984
## hhsize5     -0.004058205  0.0052361667 -0.0014036002  0.0021959436
## hhsize5plus -0.005905133  0.0035135207 -0.0012190004 -0.0036630266
## income2      0.033958194  0.0010735895 -0.0030916249  0.0007583462
## income3      0.021170609 -0.0034826777  0.0009539666  0.0039261695
## income4      0.014230716  0.0040980811  0.0002906526  0.0019086626
## income5      0.001024025  0.0023689325  0.0041548779  0.0013584477
## income6     -0.005985328 -0.0004665910 -0.0022039423  0.0036230591
## income7     -0.022620744 -0.0021596057 -0.0030077782  0.0011445471
## income8     -0.019337643 -0.0004385327  0.0010573958 -0.0097477412
## income9     -0.015465592 -0.0010709443  0.0040468642 -0.0112448053
## owner        0.013404913  0.0052303657  0.0023868136  0.0081753957
##                      PC24          PC25          PC26          PC27
## hh_id        2.914246e-04  1.633755e-04  1.059804e-16  4.075427e-16
## year        -2.438779e-05 -1.174223e-05 -6.896779e-17 -1.958738e-17
## month        3.099052e-04  1.492131e-04 -2.063241e-16 -4.531319e-16
## zipcode     -9.519019e-03  5.409317e-03  2.839606e-16  5.243876e-16
## control     -9.227151e-04 -3.617133e-04 -4.390157e-03 -5.370009e-04
## treatment    9.227151e-04  3.617133e-04 -4.390157e-03 -5.370009e-04
## lusage      -1.049914e-03 -5.133659e-04  7.406793e-16  1.186522e-15
## luse1       -2.828725e-01  1.274970e-01  3.319447e-16  5.761069e-16
## luse2        6.227315e-01 -4.343119e-01  1.037701e-15 -3.251958e-16
## luse3       -1.782637e-01  6.378231e-01 -1.136371e-15  1.832937e-15
## luse4       -5.045525e-01 -5.537815e-01 -1.056667e-15 -1.777105e-15
## luse5        4.758036e-01  2.808399e-01 -3.960625e-17  2.247101e-15
## luse6       -1.390014e-01 -5.210987e-02  9.374269e-17 -1.751141e-15
```

```
## children     3.114958e-03 -1.497705e-03 -7.284948e-16  8.212601e-16
## hhsize2      8.233735e-04  9.760457e-04  5.601207e-01 -6.015197e-03
## hhsize3     -1.423603e-04 -2.055399e-05  5.239979e-01 -5.627270e-03
## hhsize4     -1.225948e-03  2.969970e-04  4.493341e-01 -4.825448e-03
## hhsize5      8.268822e-04 -1.335578e-03  3.539630e-01 -3.801247e-03
## hhsize5plus -4.421720e-04 -6.770733e-04  2.904245e-01 -3.118900e-03
## income2      4.665302e-04  1.018416e-04 -2.628927e-03 -2.449161e-01
## income3     -1.526478e-03  7.206588e-04 -3.587443e-03 -3.342135e-01
## income4     -6.417636e-04  2.873794e-04 -3.875074e-03 -3.610098e-01
## income5      6.755615e-04  1.034851e-03 -4.039262e-03 -3.763058e-01
## income6     -8.036563e-04 -1.173059e-03 -5.574545e-03 -5.193359e-01
## income7      1.736123e-04 -1.819462e-04 -4.381853e-03 -4.082223e-01
## income8      1.356001e-03 -5.163866e-04 -2.952029e-03 -2.750170e-01
## income9      1.648799e-03  3.162060e-04 -2.258679e-03 -2.104231e-01
## owner        3.455755e-03  3.034450e-04  1.090523e-16  3.486109e-17
##                      PC28
## hh_id        2.153056e-17
## year         1.118162e-16
## month        2.422456e-16
## zipcode      7.667929e-17
## control     -7.070929e-01
## treatment   -7.070929e-01
## lusage       1.864026e-16
## luse1        1.924547e-17
## luse2       -6.706598e-16
## luse3        9.092298e-16
## luse4       -4.608888e-16
## luse5        8.934449e-17
## luse6       -1.202767e-16
## children     1.560039e-16
## hhsize2     -3.473076e-03
## hhsize3     -3.249094e-03
## hhsize4     -2.786134e-03
## hhsize5     -2.194778e-03
## hhsize5plus -1.800802e-03
## income2      2.023236e-04
## income3      2.760915e-04
## income4      2.982278e-04
## income5      3.108638e-04
## income6      4.290199e-04
## income7      3.372297e-04
## income8      2.271897e-04
## income9      1.738291e-04
## owner       -1.534436e-16
```

There are 28 principal components because there are 28 variables.

Now I will calculate the variance that each principal component explains as well as the proportion of variance explained (PVE)

Variances for each principal component:

```
pr.var <- pr.out$sdev^2
pr.var
```

```
##  [1] 6.402950e+00 1.999838e+00 1.716627e+00 1.493058e+00 1.356923e+00
##  [6] 1.258443e+00 1.186673e+00 1.158234e+00 1.134034e+00 1.105880e+00
## [11] 1.090295e+00 1.087546e+00 1.066039e+00 1.041478e+00 1.000000e+00
## [16] 9.942373e-01 9.213682e-01 6.684190e-01 5.883711e-01 3.353160e-01
## [21] 1.995940e-01 8.294786e-02 7.217696e-02 2.733072e-02 1.222029e-02
## [26] 2.305872e-28 1.033206e-28 1.262759e-30
```

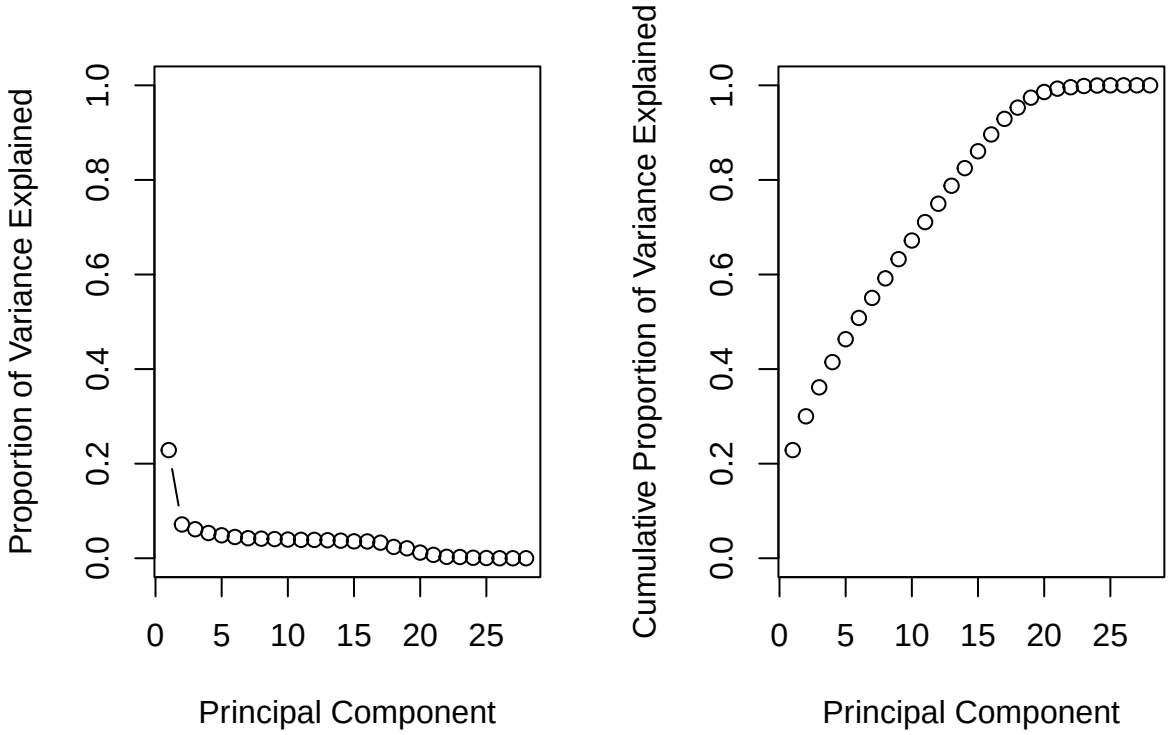
Proportion of Variance Explained:

```
pve <- pr.var / sum(pr.var)
pve
```

```
##  [1] 2.286768e-01 7.142277e-02 6.130810e-02 5.332352e-02 4.846155e-02
##  [6] 4.494439e-02 4.238119e-02 4.136551e-02 4.050120e-02 3.949570e-02
## [11] 3.893912e-02 3.884093e-02 3.807283e-02 3.719566e-02 3.571429e-02
## [16] 3.550847e-02 3.290601e-02 2.387211e-02 2.101325e-02 1.197557e-02
## [21] 7.128357e-03 2.962424e-03 2.577749e-03 9.760971e-04 4.364388e-04
## [26] 8.235256e-30 3.690020e-30 4.509854e-32
```

Plot PVE and Cumulative PVE:

```
par(mfrow = c(1, 2))
plot(pve, xlab = "Principal Component",
    ylab = "Proportion of Variance Explained", ylim = c(0, 1),
    type = "b")
plot(cumsum(pve), xlab = "Principal Component",
    ylab = "Cumulative Proportion of Variance Explained",
    ylim = c(0, 1), type = "b")
```



It appears the first principal component is much more explanatory of the variance, and there is a severe drop-off afterwards.

Time to build a model for the control group. I need to predict the usage for 2011 using the values from 2010 and a best subset selection.

I will start by using subset selection to look at which variables might be most important:

```
library(leaps)
regfit.full <- regsubsets(lusage ~ . - year - control - treatment - hh_id, data = subset(control_group,
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =
## force.in, : 2 linear dependencies found
```

```
## Reordering variables and trying again:
```

```
summary(regfit.full)
```

```
## Subset selection object
## Call: regsubsets.formula(lusage ~ . - year - control - treatment -
##     hh_id, data = subset(control_group, year == 2010), nvmax = 30)
## 23 Variables  (and intercept)
##             Forced in Forced out
## month           FALSE      FALSE
## zipcode         FALSE      FALSE
## luse1           FALSE      FALSE
```

```
## luse2           FALSE      FALSE
## luse3           FALSE      FALSE
## luse4           FALSE      FALSE
## luse5           FALSE      FALSE
## luse6           FALSE      FALSE
## children        FALSE      FALSE
## hhsize2         FALSE      FALSE
## hhsize3         FALSE      FALSE
## hhsize4         FALSE      FALSE
## hhsize5         FALSE      FALSE
## income2         FALSE      FALSE
## income3         FALSE      FALSE
## income4         FALSE      FALSE
## income5         FALSE      FALSE
## income6         FALSE      FALSE
## income7         FALSE      FALSE
## income8         FALSE      FALSE
## owner           FALSE      FALSE
## hhsize5plus     FALSE      FALSE
## income9         FALSE      FALSE
## 1 subsets of each size up to 21
## Selection Algorithm: exhaustive
##           month zipcode luse1 luse2 luse3 luse4 luse5 luse6 children hhsize2
## 1  ( 1 )  " "   " "     " "   " "   " "   " "   "*"   " "   " "      " "
## 2  ( 1 )  "*"   " "     " "   " "   " "   " "   "*"   " "   " "      " "
## 3  ( 1 )  "*"   " "     " "   " "   " "   " "   "*"   "*"   " "      " "
## 4  ( 1 )  "*"   " "     "*"   " "   " "   " "   "*"   "*"   " "      " "
## 5  ( 1 )  "*"   " "     "*"   " "   " "   "*"   "*"   "*"   " "      " "
## 6  ( 1 )  "*"   " "     "*"   " "   " "   "*"   "*"   "*"   "*"      " "
## 7  ( 1 )  "*"   " "     "*"   "*"   " "   "*"   "*"   "*"   "*"      " "
## 8  ( 1 )  "*"   "*"     "*"   "*"   " "   "*"   "*"   "*"   "*"      " "
## 9  ( 1 )  "*"   "*"     "*"   "*"   " "   "*"   "*"   "*"   "*"      " "
## 10  ( 1 ) "*"   "*"     "*"   "*"   " "   "*"   "*"   "*"   "*"      " "
## 11  ( 1 ) "*"   "*"     "*"   "*"   " "   "*"   "*"   "*"   "*"      " "
## 12  ( 1 ) "*"   "*"     "*"   "*"   " "   "*"   "*"   "*"   "*"      " "
## 13  ( 1 ) "*"   "*"     "*"   "*"   " "   "*"   "*"   "*"   "*"      " "
## 14  ( 1 ) "*"   "*"     "*"   "*"   " "   "*"   "*"   "*"   "*"      "*"
## 15  ( 1 ) "*"   "*"     "*"   "*"   " "   "*"   "*"   "*"   "*"      "*"
## 16  ( 1 ) "*"   "*"     "*"   "*"   " "   "*"   "*"   "*"   "*"      "*"
## 17  ( 1 ) "*"   "*"     "*"   "*"   "*"   "*"   "*"   "*"   "*"      "*"
## 18  ( 1 ) "*"   "*"     "*"   "*"   "*"   "*"   "*"   "*"   "*"      "*"
## 19  ( 1 ) "*"   "*"     "*"   "*"   "*"   "*"   "*"   "*"   "*"      "*"
## 20  ( 1 ) "*"   "*"     "*"   "*"   "*"   "*"   "*"   "*"   "*"      "*"
## 21  ( 1 ) "*"   "*"     "*"   "*"   "*"   "*"   "*"   "*"   "*"      "*"
##           hhsize3 hhsize4 hhsize5 hhsize5plus income2 income3 income4 income5
## 1  ( 1 )  " "     " "     " "     " "         " "     " "     " "     " "
## 2  ( 1 )  " "     " "     " "     " "         " "     " "     " "     " "
## 3  ( 1 )  " "     " "     " "     " "         " "     " "     " "     " "
## 4  ( 1 )  " "     " "     " "     " "         " "     " "     " "     " "
## 5  ( 1 )  " "     " "     " "     " "         " "     " "     " "     " "
## 6  ( 1 )  " "     " "     " "     " "         " "     " "     " "     " "
## 7  ( 1 )  " "     " "     " "     " "         " "     " "     " "     " "
## 8  ( 1 )  " "     " "     " "     " "         " "     " "     " "     " "
## 9  ( 1 )  " "     " "     " "     " "         " "     " "     " "     " "
```

```
## 10  ( 1 ) " "     " "     " "     " "         " "     " "     " "     "*"
## 11  ( 1 ) " "     " "     " "     " "         " "     " "     "*"     "*"
## 12  ( 1 ) " "     " "     " "     " "         " "     " "     " "     " "
## 13  ( 1 ) " "     " "     " "     " "         " "     " "     " "     " "
## 14  ( 1 ) " "     " "     " "     " "         " "     " "     " "     " "
## 15  ( 1 ) " "     " "     " "     " "         " "     "*"     " "     " "
## 16  ( 1 ) " "     " "     "*"     " "         " "     "*"     " "     " "
## 17  ( 1 ) " "     " "     "*"     " "         " "     "*"     " "     " "
## 18  ( 1 ) " "     " "     "*"     "*"         " "     "*"     " "     " "
## 19  ( 1 ) " "     " "     "*"     "*"         " "     "*"     " "     "*"
## 20  ( 1 ) "*"     "*"     "*"     " "         " "     "*"     " "     "*"
## 21  ( 1 ) "*"     "*"     "*"     " "         "*"     "*"     "*"     "*"
##           income6 income7 income8 income9 owner
## 1  ( 1 )  " "     " "     " "     " "     " "
## 2  ( 1 )  " "     " "     " "     " "     " "
## 3  ( 1 )  " "     " "     " "     " "     " "
## 4  ( 1 )  " "     " "     " "     " "     " "
## 5  ( 1 )  " "     " "     " "     " "     " "
## 6  ( 1 )  " "     " "     " "     " "     " "
## 7  ( 1 )  " "     " "     " "     " "     " "
## 8  ( 1 )  " "     " "     " "     " "     " "
## 9  ( 1 )  " "     "*"     " "     " "     " "
## 10  ( 1 ) " "     "*"     " "     " "     " "
## 11  ( 1 ) " "     "*"     " "     " "     " "
## 12  ( 1 ) "*"     "*"     "*"     "*"     " "
## 13  ( 1 ) "*"     "*"     "*"     "*"     "*"
## 14  ( 1 ) "*"     "*"     "*"     "*"     "*"
## 15  ( 1 ) "*"     "*"     "*"     "*"     "*"
## 16  ( 1 ) "*"     "*"     "*"     "*"     "*"
## 17  ( 1 ) "*"     "*"     "*"     "*"     "*"
## 18  ( 1 ) "*"     "*"     "*"     "*"     "*"
## 19  ( 1 ) "*"     "*"     "*"     "*"     "*"
## 20  ( 1 ) "*"     "*"     "*"     "*"     "*"
## 21  ( 1 ) "*"     "*"     "*"     " "     "*"
```

```
reg.summary <- summary(regfit.full)
```

The model recognizes `year`, `control`, `treatment`, `hhsize5plus`, and `income9` as linear dependencies. I have decided to leave `hhsize5plus` and `income9` in the model because they should be unique and important to the analysis, but I have removed `year`, `control`, `treatment`, with the addition of `hh_id` because those are either confounders or should not have any meaningful impact on `lusage`.

`regsubsets()` provides quite a few metrics to work with:

```
names(reg.summary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

Looking at R^2, only including `luse5` explains 66 % of the variance on its own. Increasing the model to 2 variables, `luse5` and `month`, increases R^2 to 75 %, and everything else adds little more to the explanatory power of the model.

```
reg.summary$rsq
```

```
##  [1] 0.6606169 0.7521482 0.7794067 0.7859590 0.7871248 0.7875474 0.7876596
##  [8] 0.7877359 0.7877943 0.7878291 0.7878588 0.7878979 0.7879475 0.7879589
## [15] 0.7879603 0.7879610 0.7879612 0.7879614 0.7879614 0.7879614 0.7879614
```

Adjusted R^2 is similar:

```
reg.summary$adjr2
```

```
##  [1] 0.6606120 0.7521410 0.7793970 0.7859465 0.7871093 0.7875288 0.7876378
##  [8] 0.7877111 0.7877664 0.7877981 0.7878247 0.7878607 0.7879072 0.7879155
## [15] 0.7879138 0.7879114 0.7879085 0.7879056 0.7879025 0.7878994 0.7878963
```

Of course, it's better to take a look at several metrics instead of only one. For example, C_p:

```
reg.summary$cp
```

```
##  [1] 41059.737997 11534.888208  2743.662446   631.966274   257.879658
##  [6]   123.559083    89.377711    66.736824    49.900810    40.693469
## [11]    33.099721    22.498151     8.500329     6.799738     8.354442
## [16]    10.122479    12.058361    14.015316    16.000659    18.000028
## [21]    20.000000
```

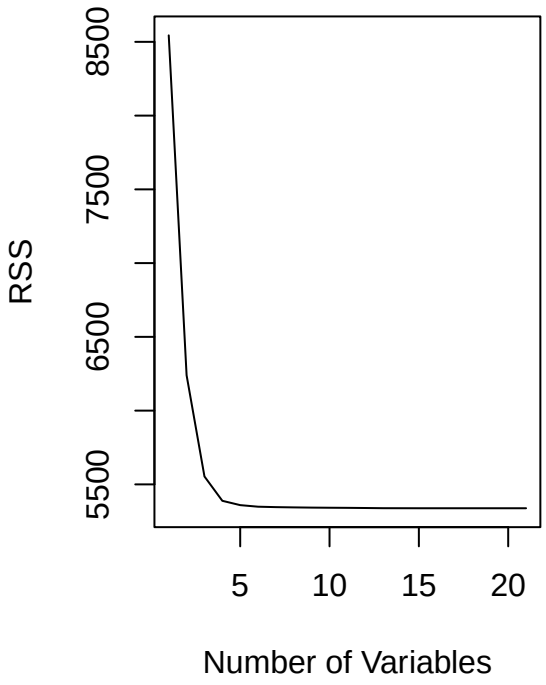
And BIC:

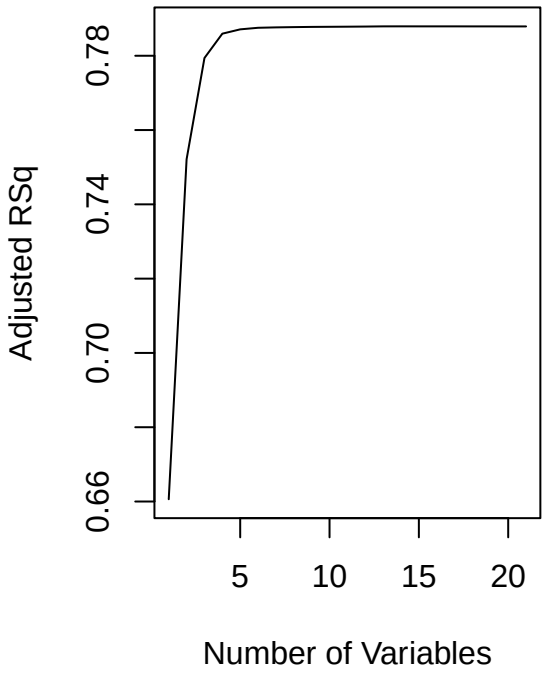
```
reg.summary$bic
```

```
##  [1]  -73919.56  -95414.30 -103375.36 -105427.46 -105790.04 -105914.88
##  [7] -105939.88 -105953.36 -105961.06 -105961.13 -105959.58 -105961.05
## [13] -105965.92 -105958.49 -105947.80 -105936.90 -105925.83 -105914.74
## [19] -105903.62 -105892.49 -105881.35
```

Plotting R^2:

```
par(mfrow = c(1, 2))
plot(reg.summary$rss, xlab = "Number of Variables",
    ylab = "RSS", type = "l")
plot(reg.summary$adjr2, xlab = "Number of Variables",
    ylab = "Adjusted RSq", type = "l")
```





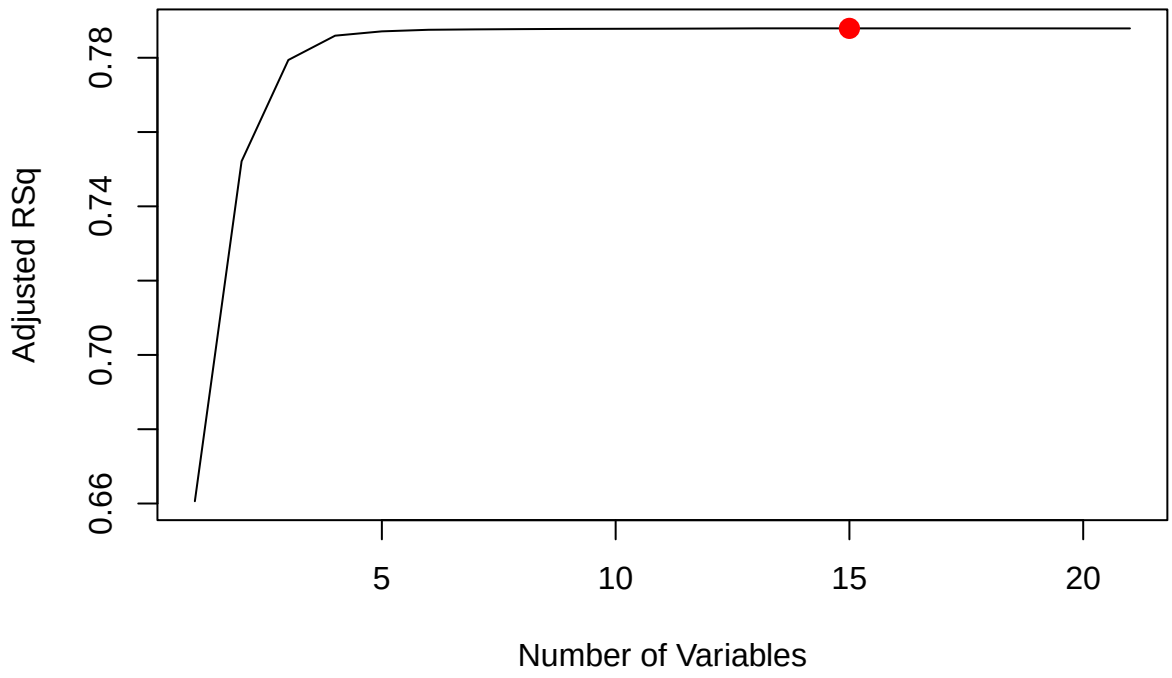
Finding the largest adjusted R^2

```
which.max(reg.summary$adjr2)
```

```
## [1] 14
```

Here is the maximum adjusted R^2 at 15 variables, although it is not much of a difference from 10 or even 5 variables.

```
plot(reg.summary$adjr2, xlab = "Number of Variables",
    ylab = "Adjusted RSq", type = "l")
points(15, reg.summary$adjr2[15], col = "red", cex = 2,
    pch = 20)
```

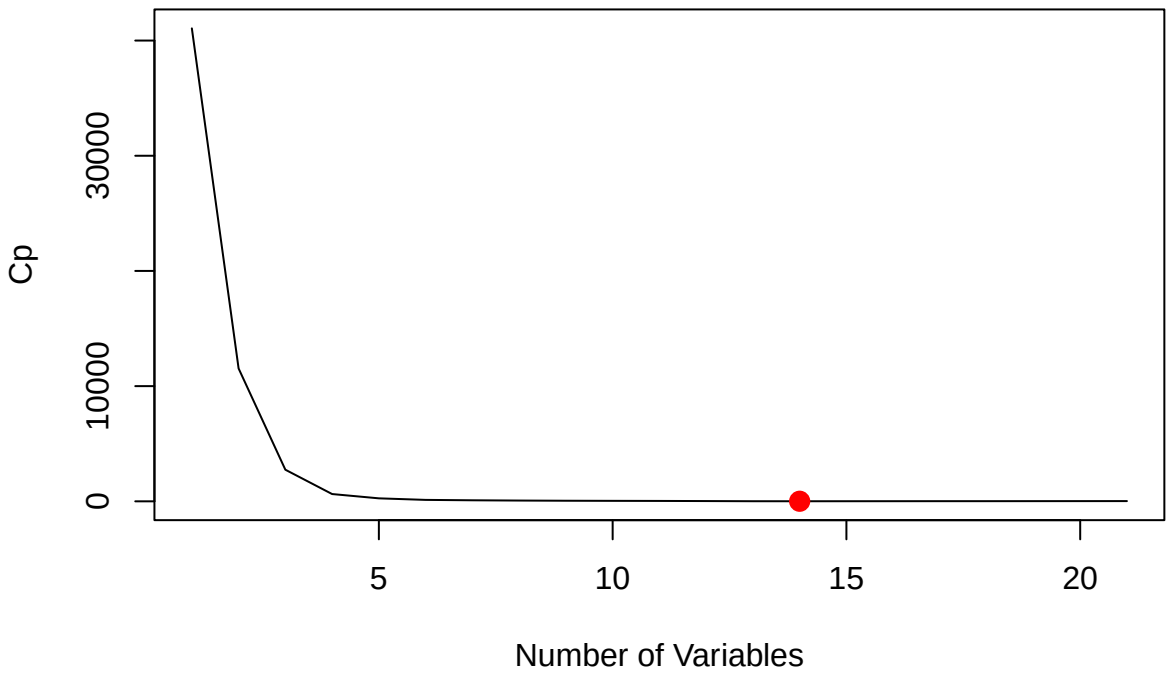


Now for C_p

```
plot(reg.summary$cp, xlab = "Number of Variables",
    ylab = "Cp", type = "l")
which.min(reg.summary$cp)
```

```
## [1] 14
```

```
points(14, reg.summary$cp[14], col = "red", cex = 2,
    pch = 20)
```

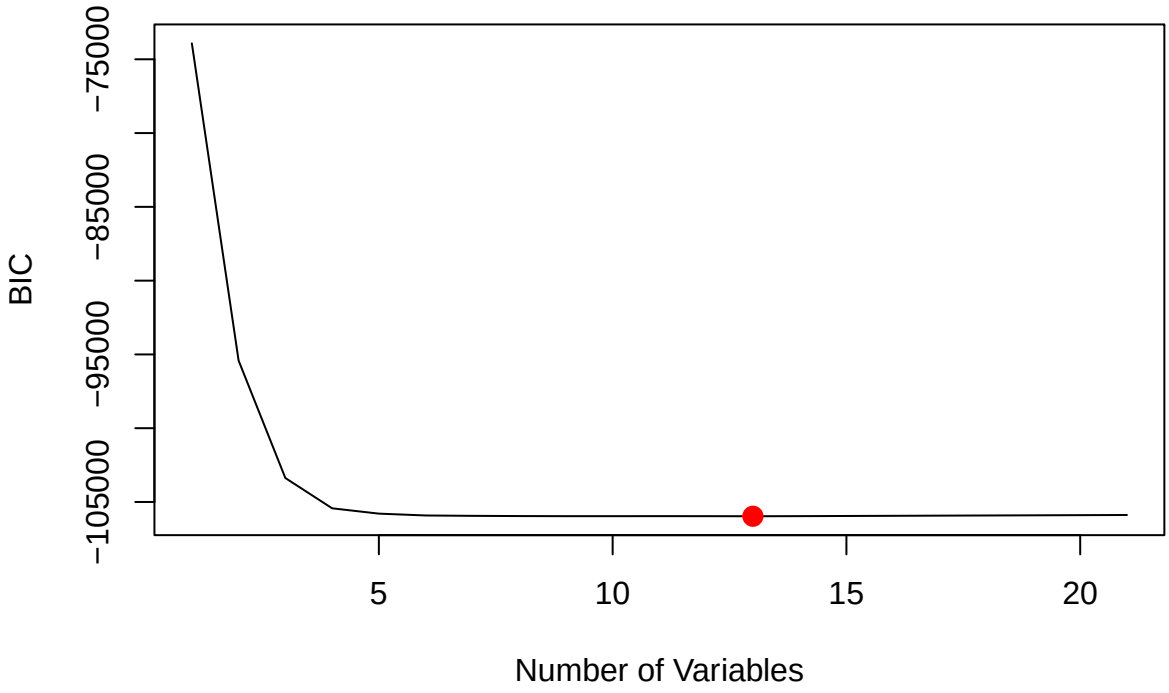


And BIC:

```
which.min(reg.summary$bic)
```

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

```
plot(reg.summary$bic, xlab = "Number of Variables",
    ylab = "BIC", type = "l")
points(13, reg.summary$bic[13], col = "red", cex = 2,
    pch = 20)
```



The recommended number of variables is 15, 14, and 13 for adjusted R^2, C_p, and BIC respectively. Looking at the coefficients for the 15-variable model:

```
coef(regfit.full, 15)
```

```
##   (Intercept)         month       zipcode         luse1         luse2
## -0.2310828256  0.1297630855 -0.0002494763  0.1305602879  0.0475668648
##         luse4         luse5         luse6      children       hhsize2
##  0.1286528103  0.3605468281  0.2564419708  0.0248217759 -0.0047786568
##       income4       income7       income8         owner   hhsize5plus
## -0.0072021287  0.0147354541  0.0128170500 -0.0112115881 -0.0007496701
##       income9
##  0.0163097553
```

A better way of doing this could be to use a lasso or ridge regression.

I need to section the data into 2010 and 2011 data and put the x's and y's into formats that can be handled by `glmnet`

This is a ridge regression. Note the dimensions of the model: 25×100

24 variables plus 1 coefficient.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
control_group.2010 = subset(control_group, year == 2010)
control_group.2010_x = control_group.2010 %>% select(-one_of("hh_id", "year", "control", "treatment"))
control_group.2010_x = model.matrix(control_group.2010$lusage ~ ., control_group.2010_x)

control_group.2011 = subset(control_group, year == 2011)
control_group.2011_x = control_group.2011 %>% select(-one_of("hh_id", "year", "control", "treatment"))
control_group.2011_x = model.matrix(control_group.2011$lusage ~ ., control_group.2011_x)


library(glmnet)
```

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

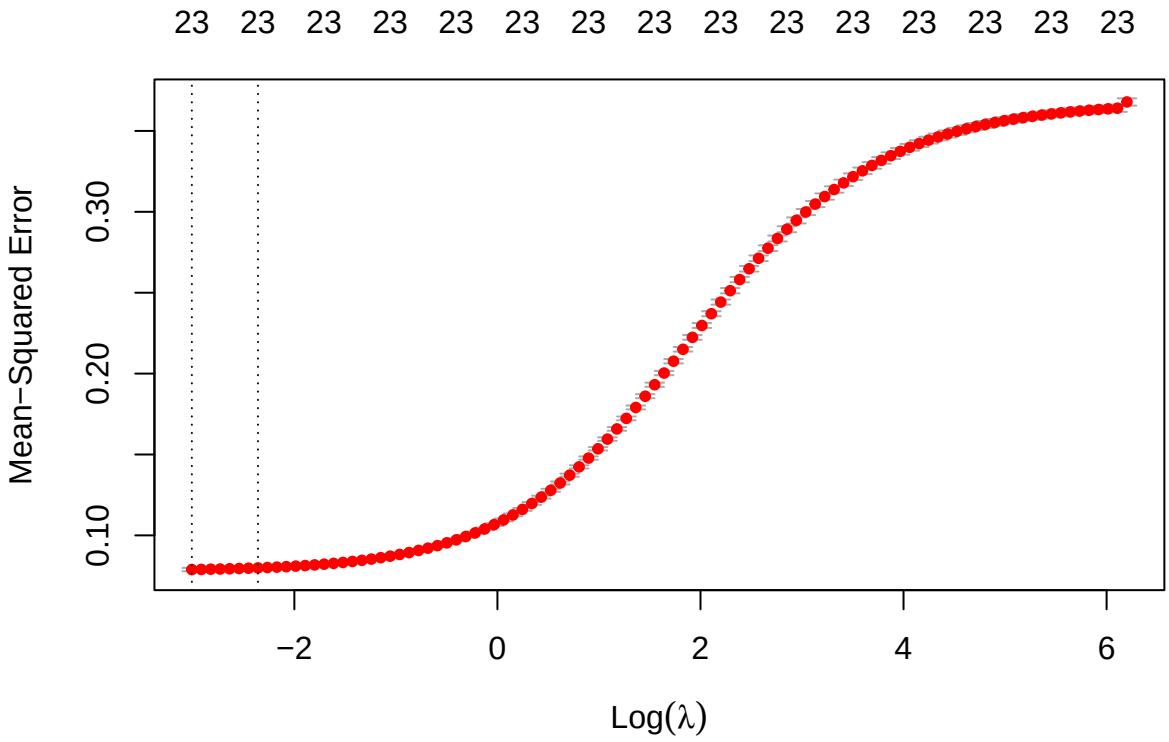
```
## Loaded glmnet 4.1-3
```

```
grid <- 10^seq(10, -2, length = 100)
ridge.mod <- glmnet(x = control_group.2010_x, y = control_group.2010$lusage, alpha = 0, lambda = grid)
dim(coef(ridge.mod))
```

```
## [1]  25 100
```

Now I plot the cross-validation of λ

```
cv.out <- cv.glmnet(control_group.2010_x, control_group.2010$lusage, alpha = 0)
plot(cv.out)
```



It looks like the best value for λ is 0.04930106

```
bestlam.ridge <- cv.out$lambda.min
bestlam.ridge
```

```
## [1] 0.04930106
```

Prediction for CONTROL group consumption data in 2011 using 2010 CONTROL group consumption data

Here is the MSE for the prediction from the model compared to `lusage` for 2011. `ridge.pred` stores the predicted values for the control group in 2011

```
ridge.pred <- predict(ridge.mod, s = bestlam.ridge,
    newx = control_group.2011_x)
mean((ridge.pred - control_group.2011$lusage)^2)
```

```
## [1] 0.1056597
```

Here are the coefficients of the fitted model.

```
out <- glmnet(control_group.2010_x, control_group.2010$lusage, alpha = 0)
predict(out, type = "coefficients", s = bestlam.ridge)[1:25, ]
```

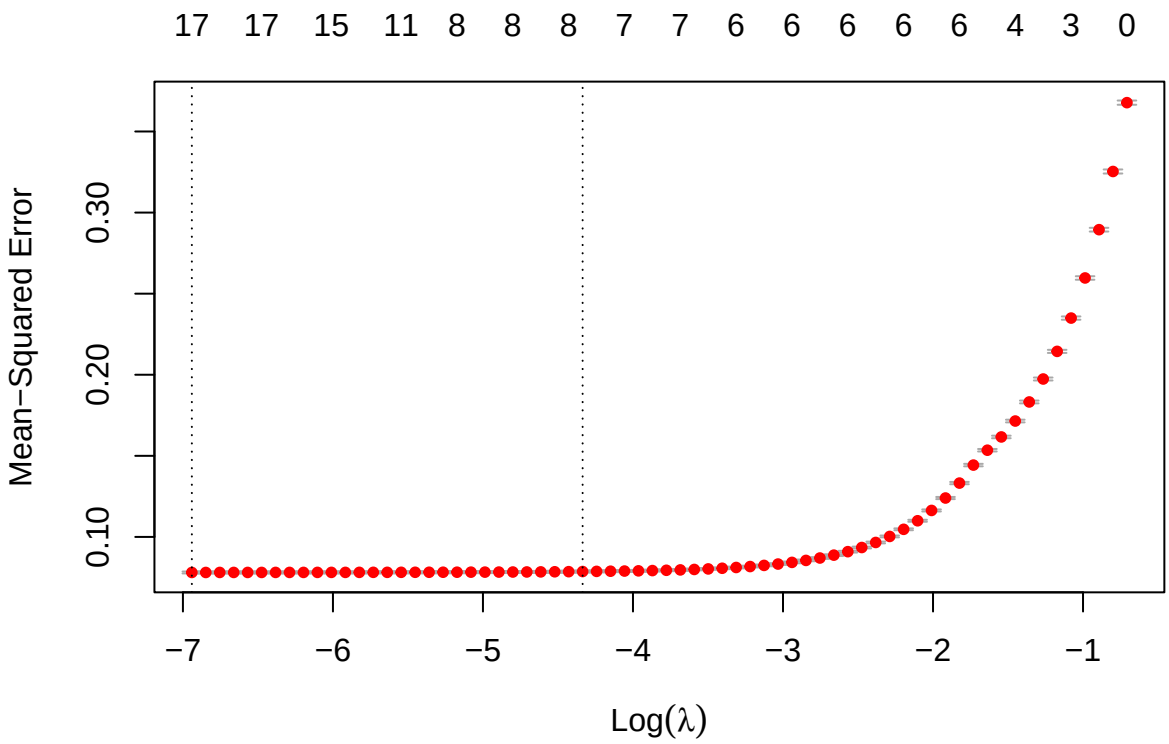
```
##  (Intercept)  (Intercept)        month      zipcode        luse1        luse2
## -0.011962696  0.000000000  0.120008956 -0.000194540  0.108412479  0.075141422
##        luse3        luse4        luse5        luse6     children      hhsize2
##  0.076041812  0.164359880  0.263186613  0.215515771  0.023269561 -0.005136451
##      hhsize3      hhsize4      hhsize5  hhsize5plus      income2      income3
##  0.001508078  0.003089544  0.002074730  0.003793887 -0.014677960 -0.009562810
##      income4      income5      income6      income7      income8      income9
## -0.012267146 -0.010281292  0.003538975  0.014050416  0.012532110  0.017188973
##        owner
## -0.013405594
```

We can do the same thing with lasso fairly easily:

```
lasso.mod <- glmnet(x = control_group.2010_x, y = control_group.2010$lusage, alpha = 1, lambda = grid)
dim(coef(lasso.mod))
```

```
## [1]  25 100
```

```
cv.out <- cv.glmnet(control_group.2010_x, control_group.2010$lusage, alpha = 1)
plot(cv.out)
```



```
bestlam.lasso <- cv.out$lambda.min
bestlam.lasso
```

```
## [1] 0.0009677998
```

`lasso.pred` stores the predicted values for the control group in 2011

```
lasso.pred <- predict(lasso.mod, s = bestlam.lasso,
    newx = control_group.2011_x)
mean((lasso.pred - control_group.2011$lusage)^2)
```

```
## [1] 0.1059928
```

Note that some of the coefficients are set to 0 with the lasso.

```
out <- glmnet(control_group.2010_x, control_group.2010$lusage, alpha = 1)
predict(out, type = "coefficients", s = bestlam.lasso)[1:25, ]
```

```
##   (Intercept)   (Intercept)         month       zipcode         luse1
## -0.2172317669  0.0000000000  0.1290787477 -0.0001972086  0.1279052378
##         luse2         luse3         luse4         luse5         luse6
##  0.0487839591  0.0000000000  0.1291673709  0.3583161339  0.2580397321
##      children       hhsize2       hhsize3       hhsize4       hhsize5
##  0.0233052895 -0.0033360161  0.0000000000  0.0000000000  0.0000000000
##   hhsize5plus       income2       income3       income4       income5
##  0.0000000000 -0.0066864100 -0.0054952713 -0.0091945369 -0.0103475151
##       income6       income7       income8       income9         owner
##  0.0000000000  0.0082187558  0.0050448326  0.0072564171 -0.0079956637
```

3. Predictions

Now I must use my model to predict the power usage for the TREATMENT group in 2011.

First, organize the data:

```
treatment_group.2010 = subset(treatment_group, year == 2010)
treatment_group.2010_x = treatment_group.2010 %>% select(-one_of("hh_id", "year", "control", "treatment
treatment_group.2010_x = model.matrix(treatment_group.2010$lusage ~ ., treatment_group.2010)

treatment_group.2011 = subset(treatment_group, year == 2011)
treatment_group.2011_x = treatment_group.2011 %>% select(-one_of("hh_id", "year", "control", "treatment
treatment_group.2011_x = model.matrix(treatment_group.2011_x$lusage ~ ., treatment_group.2011_x)
```

Using the lasso model from before on the treatment group data from 2011:

```
lasso.pred_treatment <- predict(lasso.mod, s = bestlam.lasso,
    newx = treatment_group.2011_x)
```

`lasso.pred_treatment` stores the predicted values for the electricity consumption of the TREATMENT group in 2011 using the lasso model

And now using the ridge regression:

```
ridge.pred_treatment <- predict(ridge.mod, s = bestlam.ridge,
    newx = treatment_group.2011_x)
```

`ridge.pred_treatment` stores the predicted values for the electricity consumption of the TREATMENT group in 2011 using the ridge model

4. Evaluation

Finally, I need to compare my predicted values for the 2011 TREATMENT group with the actual values for the 2011 TREATMENT group.

```
mean((lasso.pred_treatment - treatment_group.2011$lusage)^2)
```

```
## [1] 0.09568123
```

My lasso and ridge models actually had very similar MSE's, with the ridge just barely improving over the lasso.

```
mean((ridge.pred_treatment - treatment_group.2011$lusage)^2)
```

```
## [1] 0.0954475
```

I will append the predicted values to the treatment data, so that it is easier to compare them.

`treatment_group.2011` has two new columns: `lusage` copied to a new column at the end for easier comparison with the name `actual`, and `predicted`, the values estimated by the ridge regression model.

```
treatment_group.2011$actual <- treatment_group.2011$lusage
treatment_group.2011$predicted <- ridge.pred_treatment

sum(treatment_group.2011$actual) # 117571.4 > 116977.1
```

```
## [1] 116977.1
```

```
sum(treatment_group.2011$predicted)
```

```
## [1] 117571.4
```

I haven't perused the differences in values extensively, but it seems that my predicted values are pretty close to the actual values. It seems that, overall, my model overestimates the amount of electricity people consume, rather than underestimate it.