Report on BRFSS dataset

There are 3 files that I uploaded:

- elastic_net.R: perform all the elastic net models on depression and binge drinking
- helpers.R: contains all the function that makes the elastic_net.R looks a little bit cleaner
- download_files.R: download the BRFSS 2011-2015 and extract it in our system
- decode_variable.R: scrape through the `2015 codebook.pdf` file and create a `variable.csv` to get the description out of variables names.
- variable.csv: variable description from `decode_variable.R`
- state_code: all the encoded number of the states as used in their reports. I figured that it’s not worth scraping the pdf so I just typed down everything.

I have cleaned the code a little bit by creating a `helpers.R` file to contain most of our functions. The functions in `helpers.R` includes:

- elastic_var (Global Variable): all the relevant variables used for elastic net models on depression and binge drinking
- find_me: find the meaning of variables by name
- clean_data_depression: clean data for elastic net model on depression
- clean_data_binge: clean data for elastic net model on binge drinking
- var_important: based on **caret** package, finding percentage importance of variables in the model
- text_wrap: wrapping the text representation if it is too long to be presented in a graph.
- chart: draw the chart of variables’ levels of importance
- download_file: used in `download_files.R` to download files into the system.

Analysis:

Our dataset in 2011 misses a lot of variables, so we have to use our 2013 dataset as our train set and the 2015 dataset as our validation set

A. Days of derpession against variables

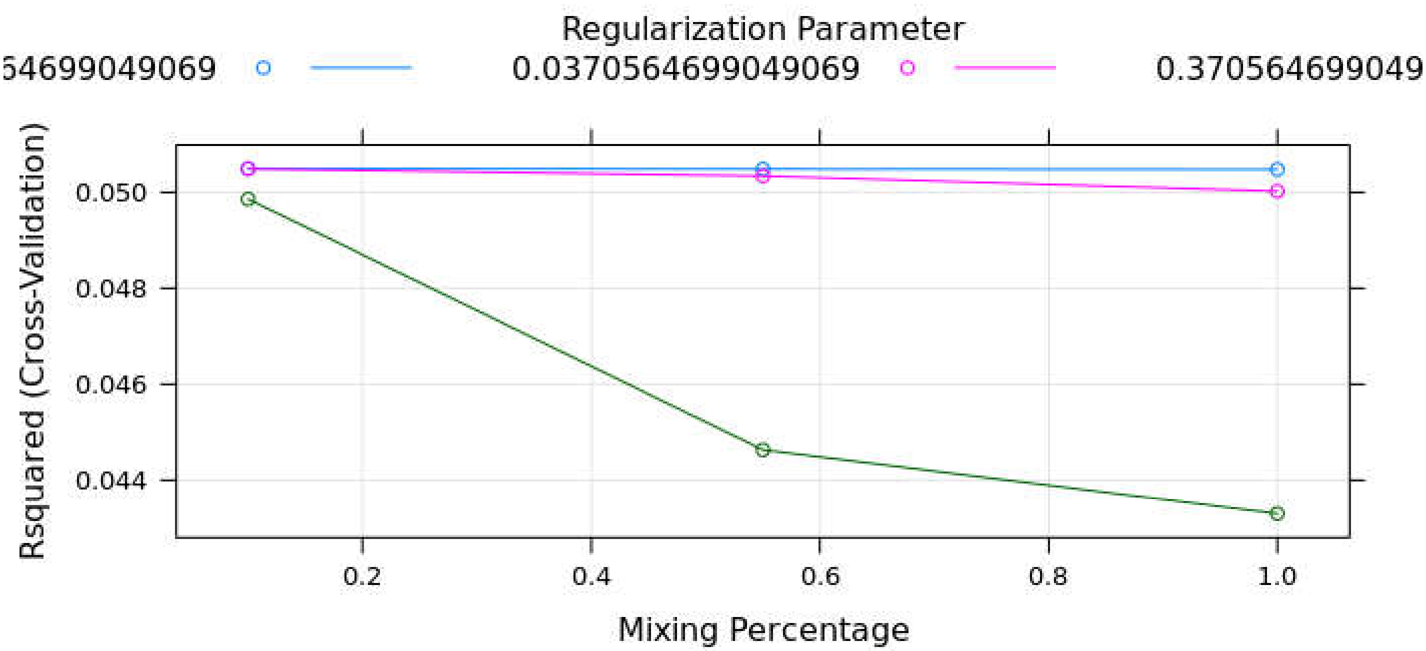
The description for the dependent variable, `MENTHLTH` is as followed:

> “Now thinking about your mental health, which includes stress, depression, and problems with emotions, for how manydays during the past 30 days was your mental health not good?”

Dimension of the data after cleaning is as follows (completed cases):

- depression_2015 has **30** independent variables with **132 972** data entries
- depression_2013 has **30** independent variables with **148 648** data entries

We fit elastic net model and finding the best tuning **alpha** and **lambda** using R-squared



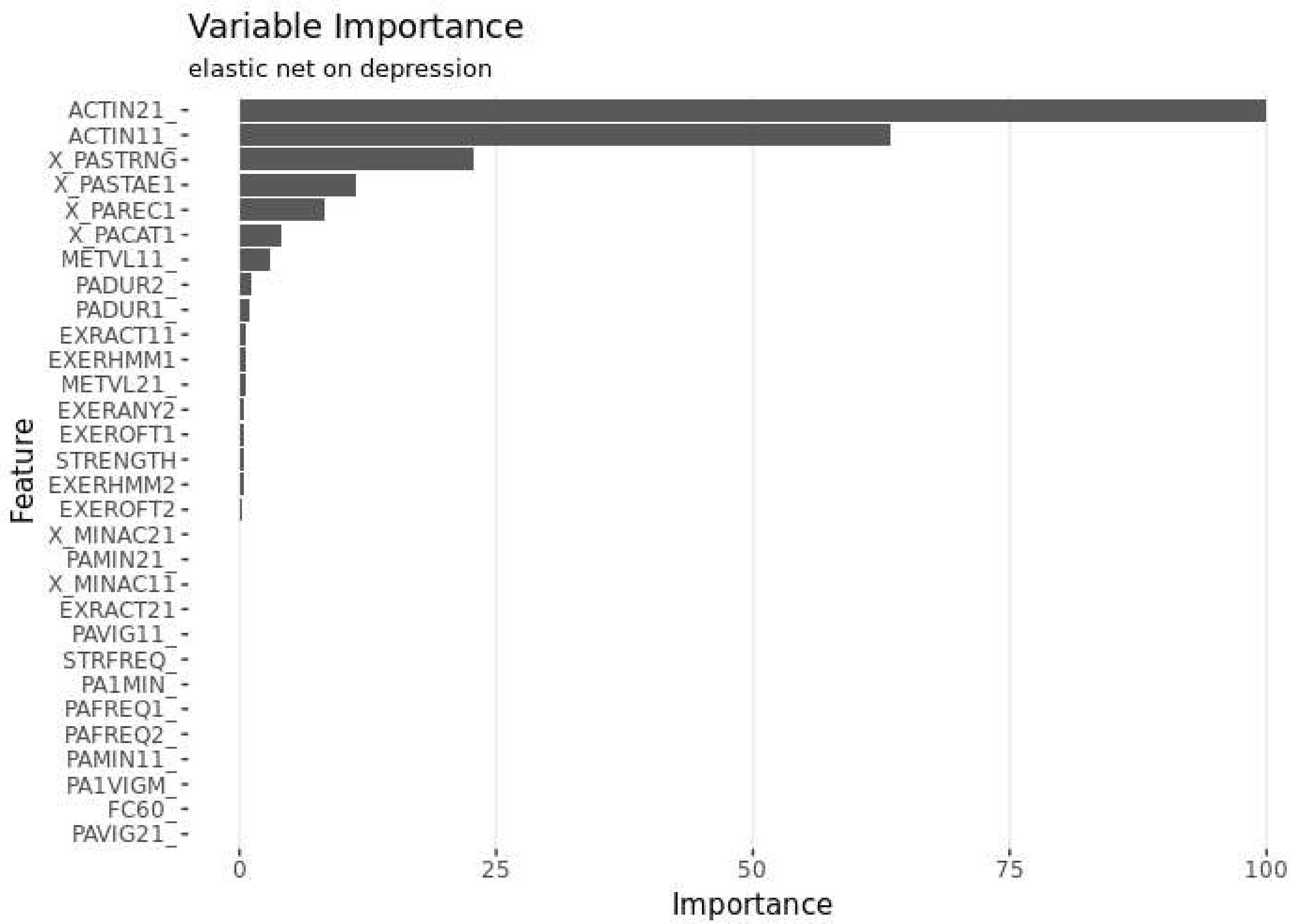
The best tuning parameters are:

```
alpha          lambda
  0.1     0.003705647
```

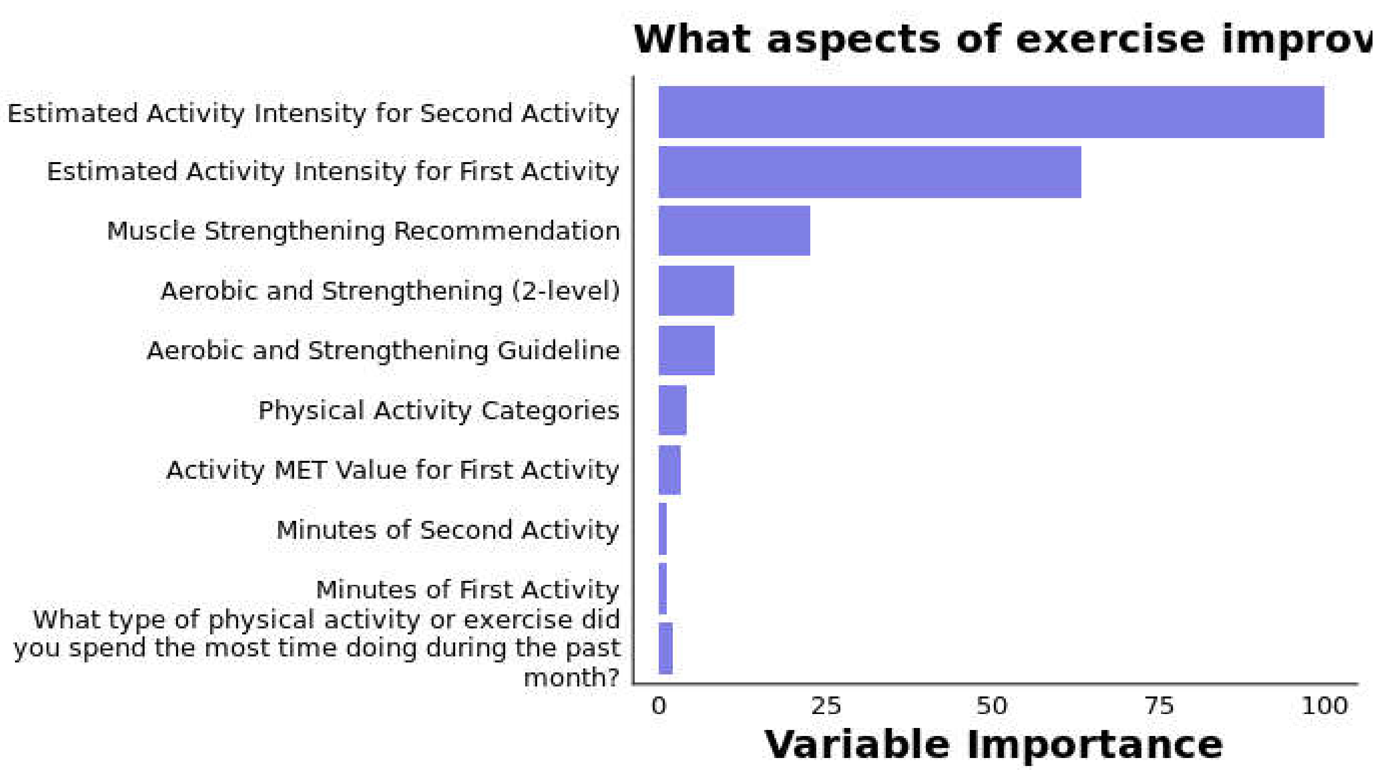
So the optimal model on the 2013 dataset is close to Ridge model with $\lambda = 0.003705647$

When we apply this model on the 2015 dataset, the $MSE = 102.778$ and the $RMSE = 10.13$.

We use the `caret` package to find the importance of variables to our model



Interpretting the variable:



So Intensity of first and second Activity are the most important variables in the model

B. Binge drinking against all other variables:

1. Amanda's approach: elastic net of `X_RFBING5` against the other variables

The best tuning parameters are: $\alpha = 0.1$ $\lambda = 0.001673393$. So it approximates Ridge penalty.

I'm not very sure about this part, since I copied Amanda's code to find the importance of variables. Obviously, the total does not sum up to 100 in this case. Is it something wrong with the original code or is it supposed to be intepreted as some sort of *confidence* of the variables

When I use this model and apply on the 2015 data, the $MSE = 2.491$.

2. My approach

The way I look at this question is that it does not seem to be a regression model as much as a classification question. Variable `X_RFBING5` (Binge drinkers (males having five or more drinks on one occasion, females having four or more drinks on one occasion)) only has 3 factors:

- 1: No
- 2: Yes
- 9: Don't know/Missing

So it makes a lot more sense to approach this question as classification. I used Penalized Logistic Regression (and compared with normal Logistic Regression)

a. Cleanning data

We removed all entries that are classified as **Don't know/Missing** and left with 2 classes: **1 (No)** and **2 (Yes)**

The count of each class in the each dataset is as follows:

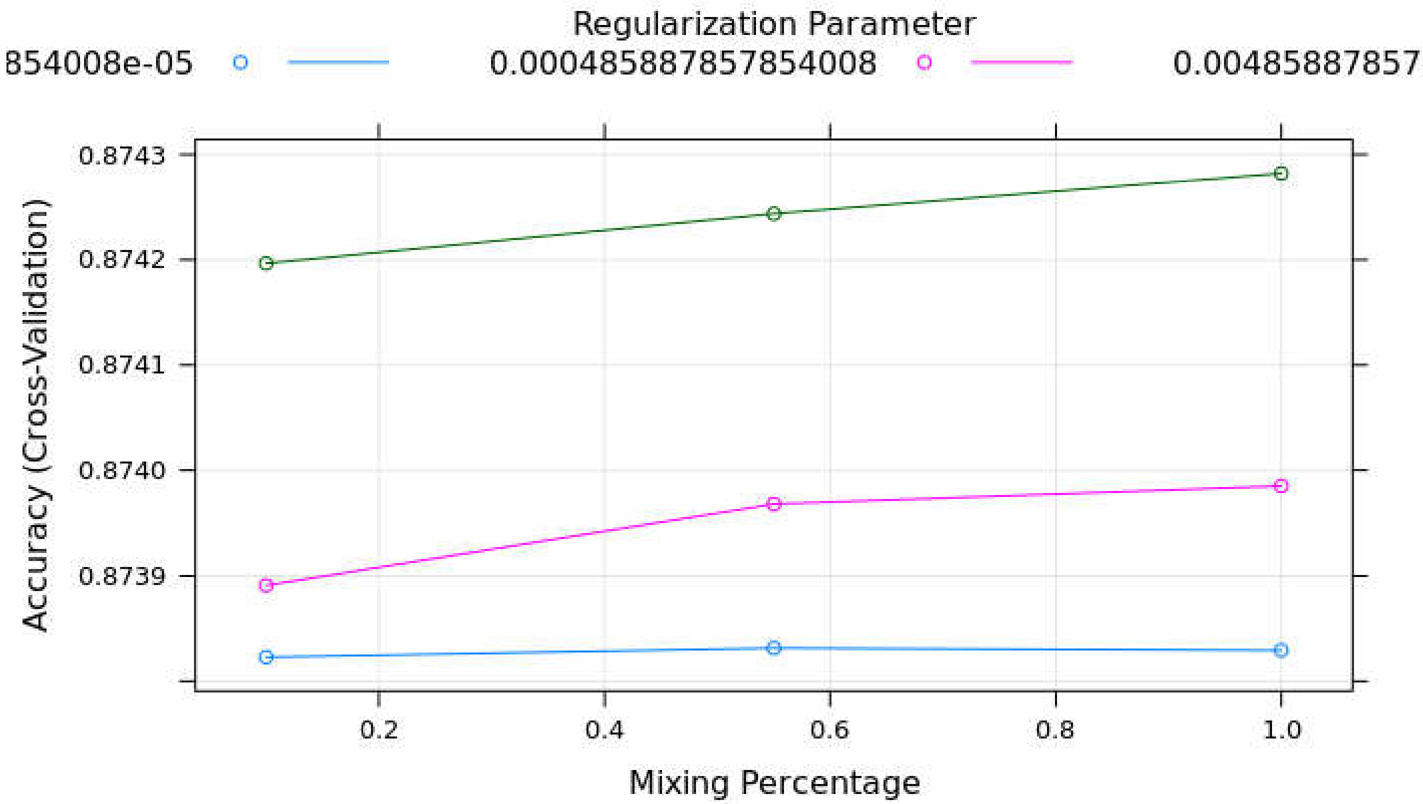
```
binge_data_2013$X_RFBING5:
     1      2
409209  58831
```

```
binge_data_2015$X_RFBING5:
     1      2
365239  50606
```

So the data in both year is disproportionally in favor of No class

b. Penalized Logistic Regression

Using cross validation to apply the Penalized logistic regression to tune the alpha and lambda value:



"

The optimal lambda (the value at which misclassication rate is minizized) is **alpha = 1** and **lambda = 0.004858879**, which is LASSO at **lambda = 0.004858879**

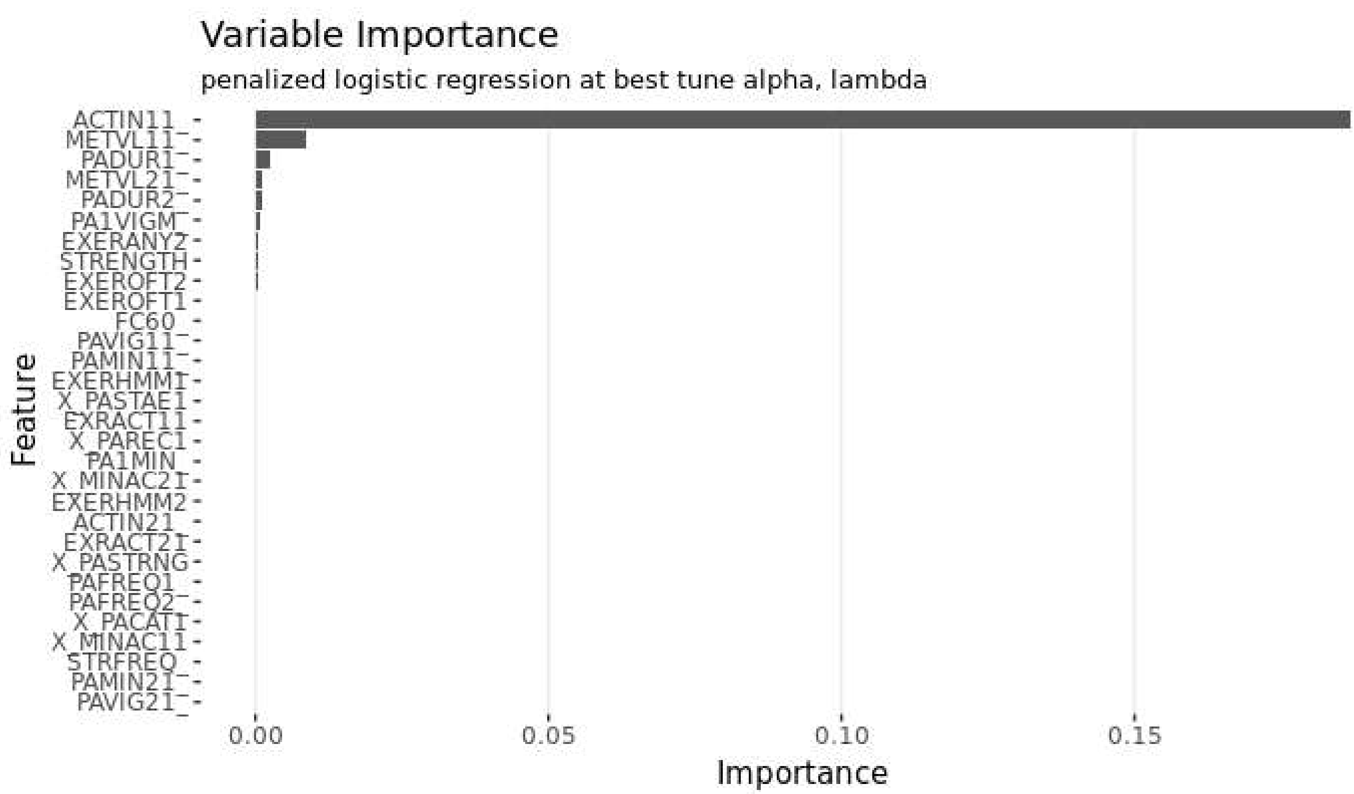
The thing about this model is that since LASSO allows variables to be reduced to zero, this penalized Logistic Model reduced almost all variables (accept the intercept) to be zero.

```
(Intercept) -1.939557

EXERANY2     .

EXRACT11     .

EXEROFT1     .

EXERHMM1     .

EXRACT21     .

EXEROFT2     .

EXERHMM2     .

STRENGTH     .

(etc)
```

When we apply this model on the 2015 dataset, we get the following statistics:

```
Confusion Matrix and Statistics

          Reference
Prediction      1      2
         1 365222  50600
         2     17      6

               Accuracy : 0.8783
                 95% CI : (0.8773, 0.8793)
    No Information Rate : 0.8783
    P-Value [Acc > NIR] : 0.522

                  Kappa : 1e-04
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.9999535
            Specificity : 0.0001186
         Pos Pred Value : 0.8783133
         Neg Pred Value : 0.2608696
             Prevalence : 0.8783056
         Detection Rate : 0.8782647
   Detection Prevalence : 0.9999447
      Balanced Accuracy : 0.5000360

       'Positive' Class : 1
```

The Importance of the Variables is as follows:



"

c. Normal Logistic Regression :

The contrast as encoded by R is as follows:

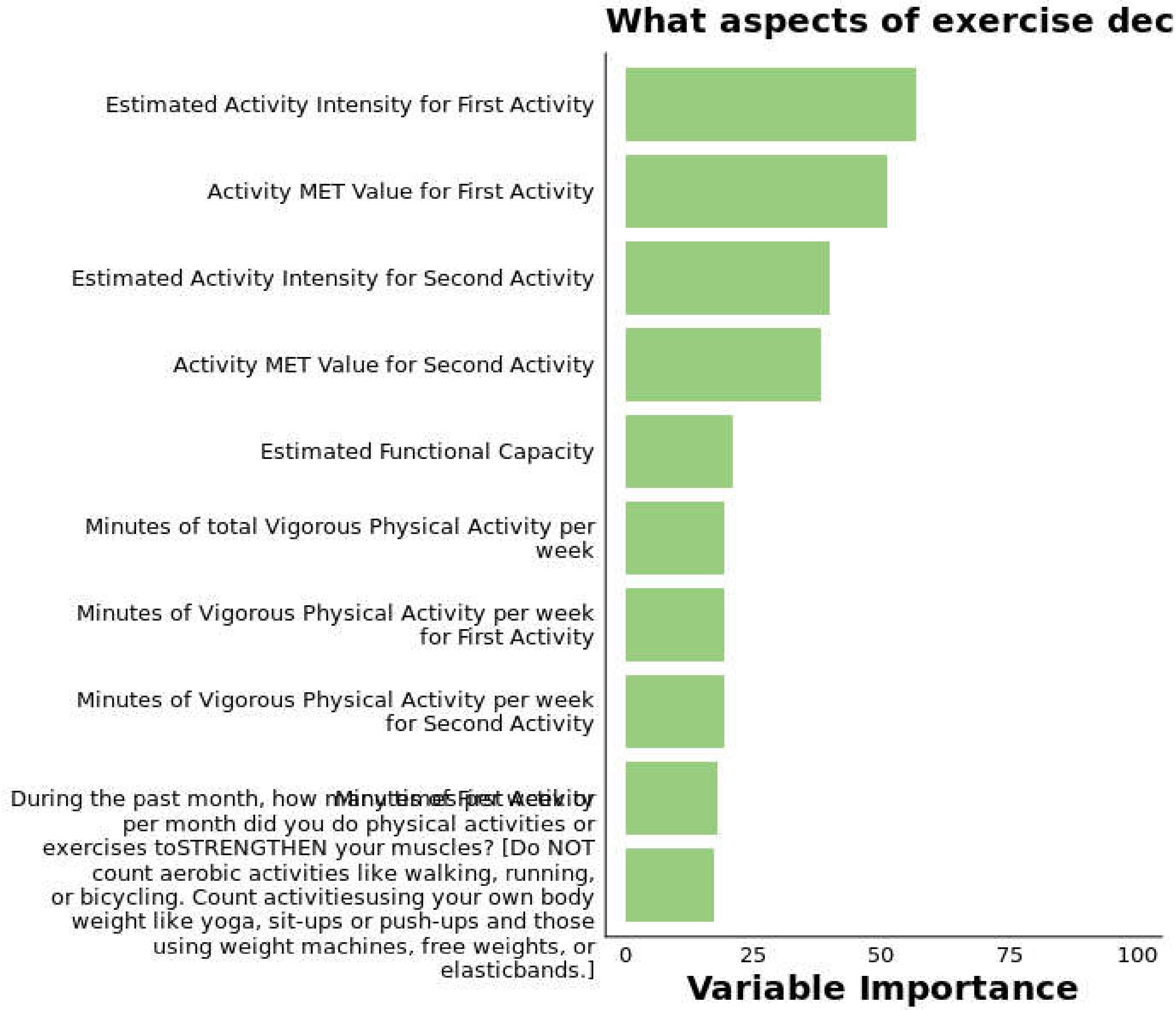
```
contrasts(binge_data_2013$X_RFBING5)
  2
1 0
2 1
```

So Class 1 is encoded as 0 and Class2 is encoded as 1

When we apply this model on the 2015 dataset, we get the following statistics:

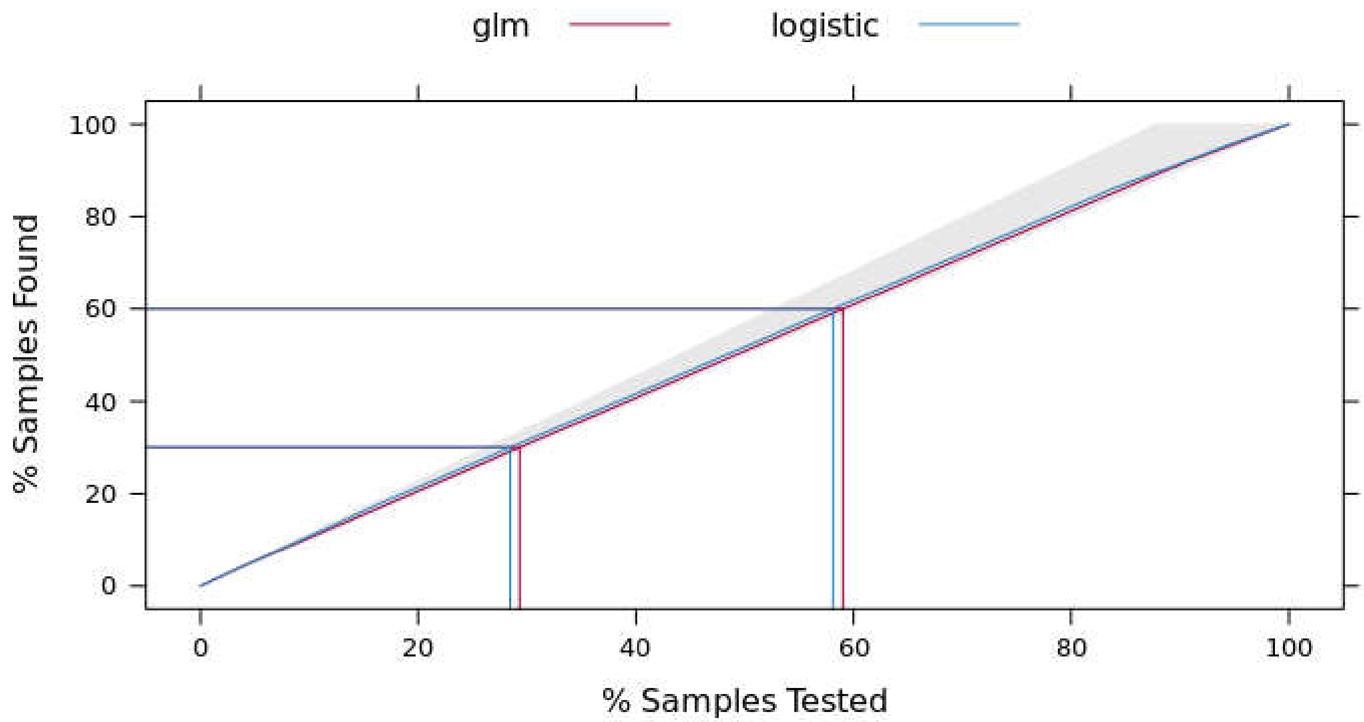
```
Confusion Matrix and Statistics

          Reference
Prediction      1      2
         1 364783  50359
         2    456    247

               Accuracy : 0.8778
                 95% CI : (0.8768, 0.8788)
    No Information Rate : 0.8783
    P-Value [Acc > NIR] : 0.8398

                  Kappa : 0.0063
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.998752
            Specificity : 0.004881
         Pos Pred Value : 0.878695
         Neg Pred Value : 0.351351
             Prevalence : 0.878306
         Detection Rate : 0.877209
   Detection Prevalence : 0.998309
      Balanced Accuracy : 0.501816

       'Positive' Class : 1
```

The Importance of Variable is shown as follows:



d. Lift Curve and Calibration Curve to compare performance of 2 models

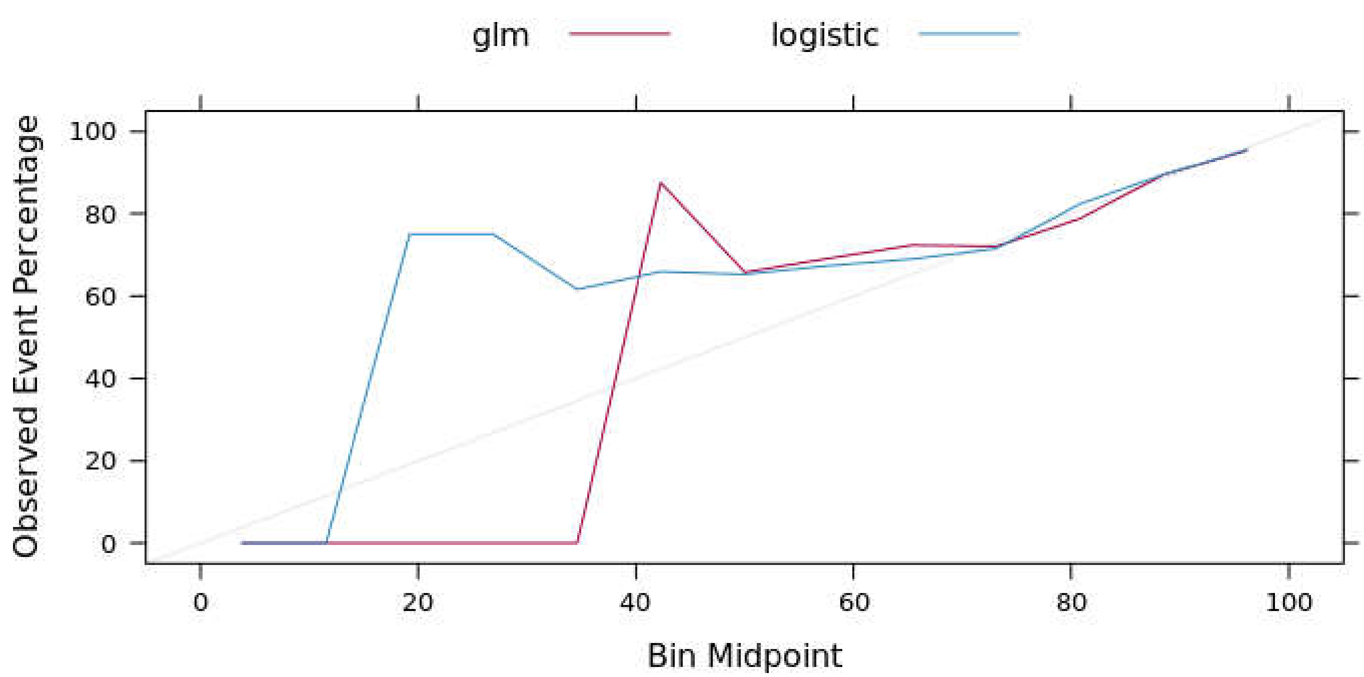
Lift Curve

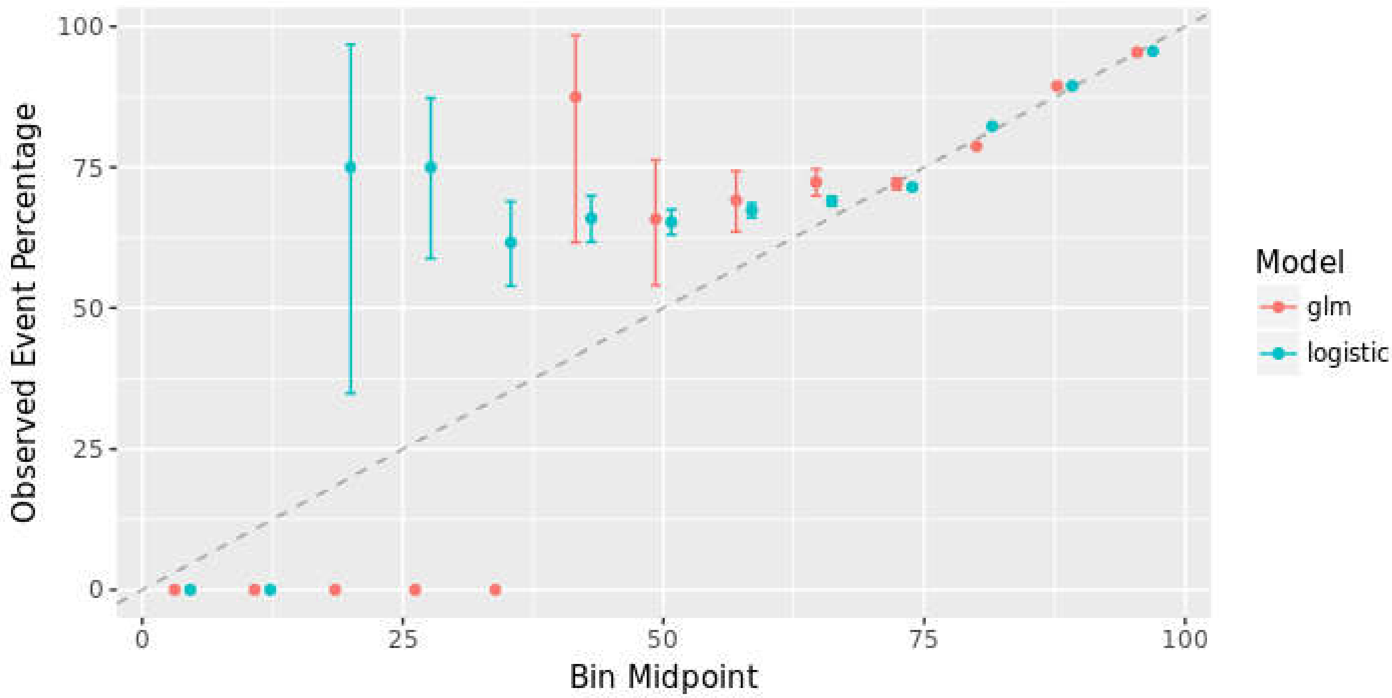


Lift measures effectiveness of predictive model by showing the ratio of the model to a random guess. I may be wrong here but the way `caret` package implements and plots `lift` function looks more like a **cumulative gain chart** , which measures *the total number of events captured by a model over a given number of sample*

The way I read the chart is that: to find 30% (or 60%) of the hits (event that we get predicted class as No - the way I implement in the code), we need a little less than 30% (and 60%) of the data to be sampled. Normal logistic outperforms glm a little bit.

<u>Calibration Curve</u>





Observed Event Percentage
100
75
50
25
0
0
25
50
75
100
Bin Midpoint
Model
glm
logistic