WilliamsM3LASSORegression

G. Holt Williams

6/1/2020

Introduction

The goal of this project is to practice LASSO regularization in the glmnet R package. Areas of practice include utilizng big data, LASSO model fitting, and linear/logistic regression. The project is broken into two parts, first being the "Regularization Assignment" and the second being "LASSO Regression in R Exercises." The first part looks at an insurance company Datset, while the second utilizes an exercise from "r-exercises" and the diabetes dataset to show the fundamentals of LASSO in R.

Regularizartion Assignment

Data Set Selection

For this assignment, the analysis will utilize the "Insurance Company Benchmark (COIL 2000) Data Set(Putten 2000)." This dataset has 5,822 rows with about 86 variables which are mostly dummy/factor variables. This dataset was retrieved from UCI's Machine Learning Repository which describes the set as "product usage and socio-demographic data(Putten 2000)." Due to the large quanitity of variables exploratory analysis will only look at the varibles that were found to have the most influence in the LASSO model utilized. This assignment will start with the model, then go into a simple exploratory data analysis section on the variables selected. The response variable for this exercise is if the customer has a mobile home insurance policy.

Load Libraries and Datasets

```r
library(glmnet)
library(tidyverse)
library(broom)
df<-read.table("ticdata2000.txt", header=FALSE)
dictionary<-read.csv("dictionary.txt")
var<- (dictionary$DATA.DICTIONARY[2:87])
#Change to factors as required
df$V1<-factor(df$V1)
df$V4<-factor(df$V4)
df$V6<-factor(df$V6)
df$V44<-factor(df$V44)
df$V86<-factor(df$V86)
var<-str_sub(var,3)
var<-gsub(" ","_", var)
```

```r
colnames(df)<-var #Add variable names to data frame
head(df)
```

MOSTYPE_Customer_Subtype_see_L0	MAANTHUI_Number_of_houses_1_-_10	MGEMOMV_Avg_size_househ
33	1	
37	1	
37	1	
9	1	
40	1	
23	1	

```r
nrow(df)
```

```
## [1] 5822
```

The dataset has 5822 rows

```r
summary(df[86])
```

CARAVAN_Number_of_mobile_home_policies_0-_1
0:5474
1: 348

The above table shows us the outcome is binary, which lends the model selection towards a logistic lasso model.

Model Selection

As shown above, the outcome variable is binary which lends to the use of a logsitic regression. With that, the regularization technique of LASSO will be utilized to reduce the quanitity of independant variables used in the final model. The analysis will start with a logistic regression to show us the base case if all variables are included

```r
bench<-glm(df$`_CARAVAN_Number_of_mobile_home_policies_0_-_1` ~ ., data=df, family = "binomial")
coeff<-as.data.frame(summary(bench)$coef)%>%rownames_to_column()
coeff%>%filter(coeff[,5]<0.1)
```

rowname	Estimate	Std. Error	z value	Pr(>|z|)
MGODRK_Roman_catholic_see_L34	-2.0242763	1.1692090	-1.731321	0.0833945
`_MOPLLAAG_Lower_level_education`	-0.2629972	0.1433451	-1.834714	0.0665481
`_PPERSAUT_Contribution_car_policies`	0.2263738	0.0428995	5.276842	0.0000001
`_PTRACTOR_Contribution_tractor_policies`	0.7485394	0.4405851	1.698967	0.0893255
`_PLEVEN_Contribution_life_insurances`	-0.2642706	0.1187790	-2.224894	0.0260883
`_PBRAND_Contribution_fire_policies`	0.2084305	0.0791808	2.632337	0.0084800
`_ALEVEN_Number_of_life_insurances`	0.5323414	0.2288924	2.325728	0.0200331
`_APLEZIER_Number_of_boat_policies`	2.4993503	1.0638484	2.349348	0.0188063

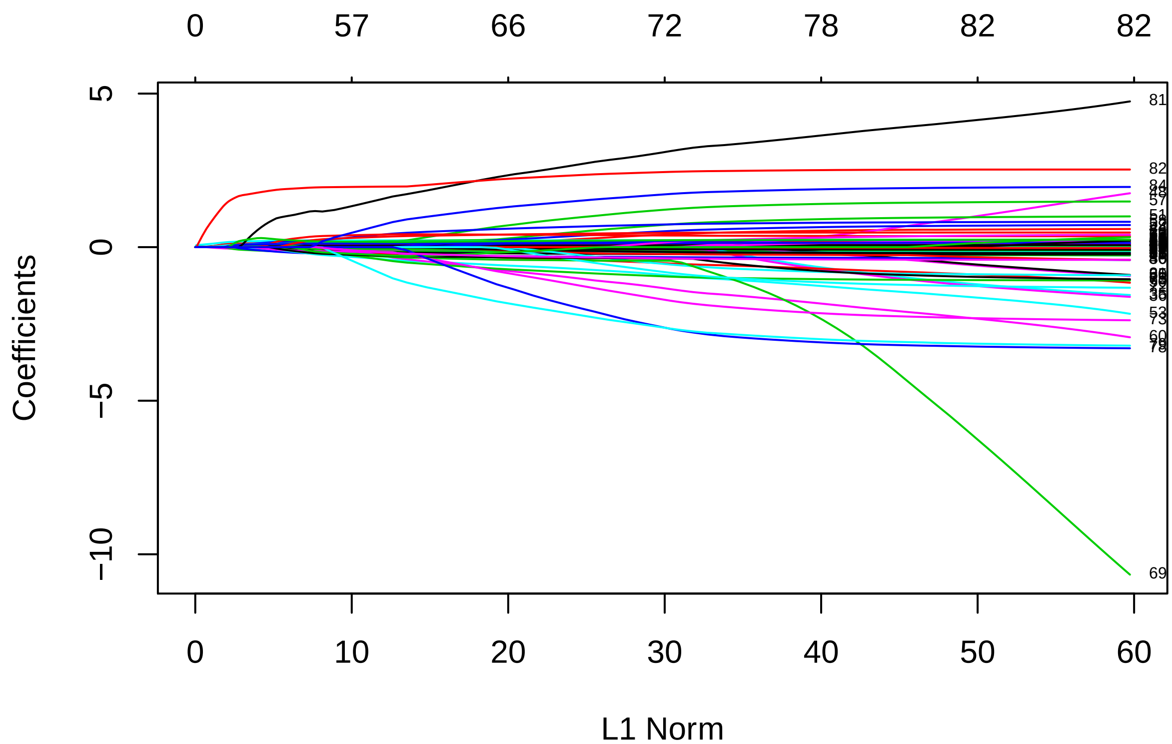
```r
tidy(bench)%>%filter(p.value<0.1)
```

term	estimate	std.error	statistic	p.value
MGODRK_Roman_catholic_see_L34	-2.0242763	1.1692090	-1.731321	0.0833945
`_MOPLLAAG_Lower_level_education`	-0.2629972	0.1433451	-1.834714	0.0665481
`_PPERSAUT_Contribution_car_policies`	0.2263738	0.0428995	5.276842	0.0000001
`_PTRACTOR_Contribution_tractor_policies`	0.7485394	0.4405851	1.698967	0.0893255
`_PLEVEN_Contribution_life_insurances`	-0.2642706	0.1187790	-2.224894	0.0260883
`_PBRAND_Contribution_fire_policies`	0.2084305	0.0791808	2.632337	0.0084800
`_ALEVEN_Number_of_life_insurances`	0.5323414	0.2288924	2.325728	0.0200331
`_APLEZIER_Number_of_boat_policies`	2.4993503	1.0638484	2.349348	0.0188063

To reduce page length the full coefficeint breakdown is not included. Only values that are statistically significant at the 90% confidence interval are shown. The variable with the most influence looks to be the number of boat policies. This suggests there is a larger likliehood of not only having a mobile home, but having an insurance policy for it, if the consumer has a boat policy.

Next, glmnet is run, and only the plotted output is shown. This shows the changes in the coefficients as the model penalizes diffrent variables toward zero. It is important here to set the seed to ensure you are able to reproduce the results later on.

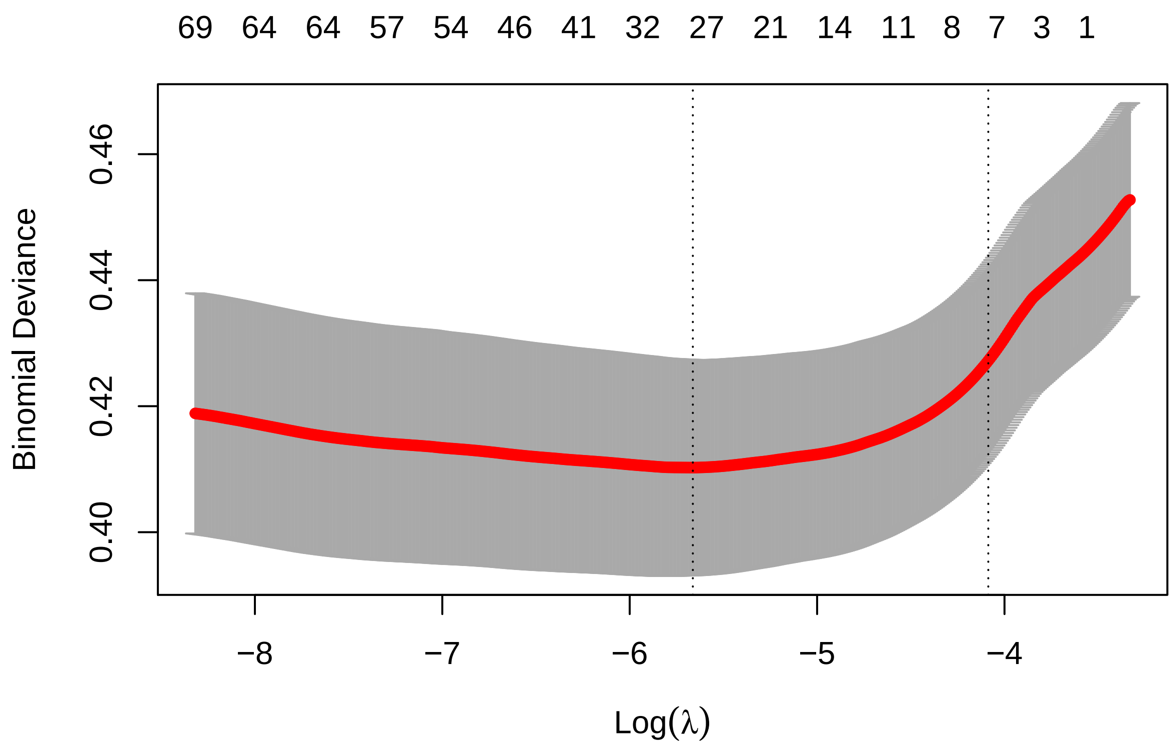
```r
set.seed(555)
glm1<-glmnet(as.matrix(df[c(1:84)]),as.matrix(df$`_CARAVAN_Number_of_mobile_home_policies_0_-_1`),famil
plot(glm1, label=TRUE)
```



This plot is useful as it shows where diffrent variables reduce to zero in viusual way, instead of parsing through all the models that were run.

Next, cross validation is utilized to determine at what lambda the model should be chosen. The model is run with the number of lambdas at 1000.

```
set.seed(555)
cvglm1<-cv.glmnet(data.matrix(df[c(1:84)]),as.matrix(df$`_CARAVAN_Number_of_mobile_home_policies_0_-_1`
plot(cvglm1)
```




```
cvglm1$lambda.min
```

```
## [1] 0.003471954
```

```
cvglm1$lambda.1se
```

```
## [1] 0.01679804
```

The resulting information is utilized to select the lambda for the glmnet. In the above cross validation run a the lambda that give minimum cvm is 0.00347. The output also give the "1se" lambda (0.0168), which is the value of lambda such that error is within one standard erro of the minimum. Outputs of both lambdas are below, but the "1se" lambda will be focused on for the EDA.

```
set.seed(555)
glm_min<- glmnet(data.matrix(df[c(1:84)]),as.matrix(df$`_CARAVAN_Number_of_mobile_home_policies_0_-_1`)
tidy(glm_min)
```

term	step	estimate	lambda	dev.ratio
(Intercept)	1	-4.8830289	0.003472	0.1194912
MGEMLEEF_Avg_age_see_L1	1	0.0214301	0.003472	0.1194912
MGODPR_Protestant_…	1	0.0196720	0.003472	0.1194912
MGODGE_No_religion	1	-0.0060707	0.003472	0.1194912
_MRELGE_Married	1	0.0483120	0.003472	0.1194912
_MRELSA_Living_together	1	-0.0108942	0.003472	0.1194912
_MOPLHOOG_High_level_education	1	0.0466976	0.003472	0.1194912
_MOPLLAAG_Lower_level_education	1	-0.0502239	0.003472	0.1194912
_MBERBOER_Farmer	1	-0.1159890	0.003472	0.1194912
_MBERMIDD_Middle_management	1	0.0252373	0.003472	0.1194912
_MHHUUR_Rented_house	1	-0.0179164	0.003472	0.1194912
_MAUT1_1_car	1	0.0446814	0.003472	0.1194912
MINKM30_Income<_30.000	1	-0.0039364	0.003472	0.1194912
_MINK7512_Income_75-122.000	1	0.0161880	0.003472	0.1194912
MINK123M_Income>123.000	1	-0.0710859	0.003472	0.1194912
_MINKGEM_Average_income	1	0.0465893	0.003472	0.1194912
_MKOOPKLA_Purchasing_power_class	1	0.0416610	0.003472	0.1194912
_PWAPART_Contribution_private_third_party_insurance_see_L4	1	0.1236368	0.003472	0.1194912
PWALAND_Contribution_third_party_insurane(agriculture)	1	-0.1072927	0.003472	0.1194912
_PPERSAUT_Contribution_car_policies	1	0.1991004	0.003472	0.1194912
_PGEZONG_Contribution_family_accidents_insurance_policies	1	0.0745775	0.003472	0.1194912
_PWAOREG_Contribution_disability_insurance_policies	1	0.1343911	0.003472	0.1194912
_PBRAND_Contribution_fire_policies	1	0.0982248	0.003472	0.1194912
_PFIETS_Contribution_bicycle_policies	1	0.0031051	0.003472	0.1194912
_PBYSTAND_Contribution_social_security_insurance_policies	1	0.0932344	0.003472	0.1194912
_ATRACTOR_Number_of_tractor_policies	1	-0.0377480	0.003472	0.1194912
_AZEILPL_Number_of_surfboard_policies	1	0.6515926	0.003472	0.1194912
_APLEZIER_Number_of_boat_policies	1	1.7940802	0.003472	0.1194912
_AFIETS_Number_of_bicycle_policies	1	0.3089942	0.003472	0.1194912

```
set.seed(555)
glm_1se<- glmnet(data.matrix(df[c(1:84)]),as.matrix(df$`_CARAVAN_Number_of_mobile_home_policies_0_-_1`)
tidy(glm_1se)
```

term	step	estimate	lambda	dev.ratio
(Intercept)	1	-3.3790939	0.016798	0.0625356
_MOPLLAAG_Lower_level_education	1	-0.0168025	0.016798	0.0625356
_MINKGEM_Average_income	1	0.0160261	0.016798	0.0625356
_MKOOPKLA_Purchasing_power_class	1	0.0317642	0.016798	0.0625356
_PWAPART_Contribution_private_third_party_insurance_see_L4	1	0.0330581	0.016798	0.0625356
_PPERSAUT_Contribution_car_policies	1	0.1146995	0.016798	0.0625356
_PBRAND_Contribution_fire_policies	1	0.0206656	0.016798	0.0625356
_APLEZIER_Number_of_boat_policies	1	0.9123868	0.016798	0.0625356

Refrences

Putten and Someren (eds) . CoIL Challenge 2000: The Insurance Company Case. Published by Sentient Machine Research, Amsterdam. Also a Leiden Institute of Advanced Computer Science Technical Report 2000-09. June 22, 2000.

LASSO Regression in R Exercises

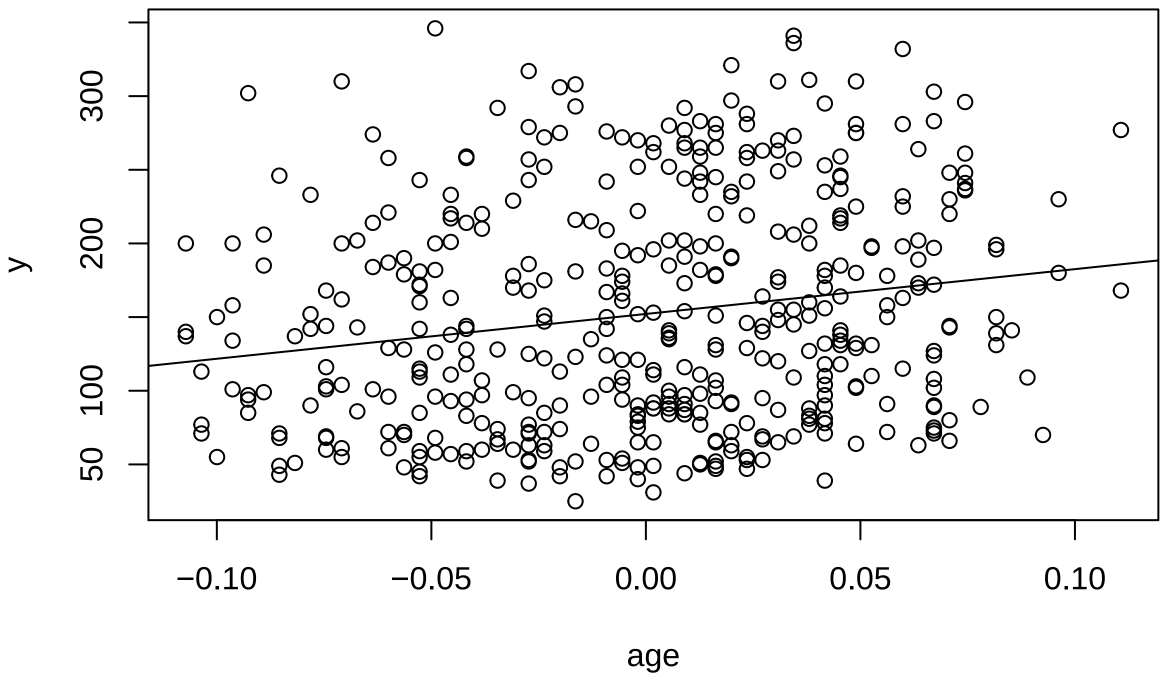
Source: https://www.r-exercises.com/2017/06/12/lasso-regression-in-r-exercises/

Exercise 1:

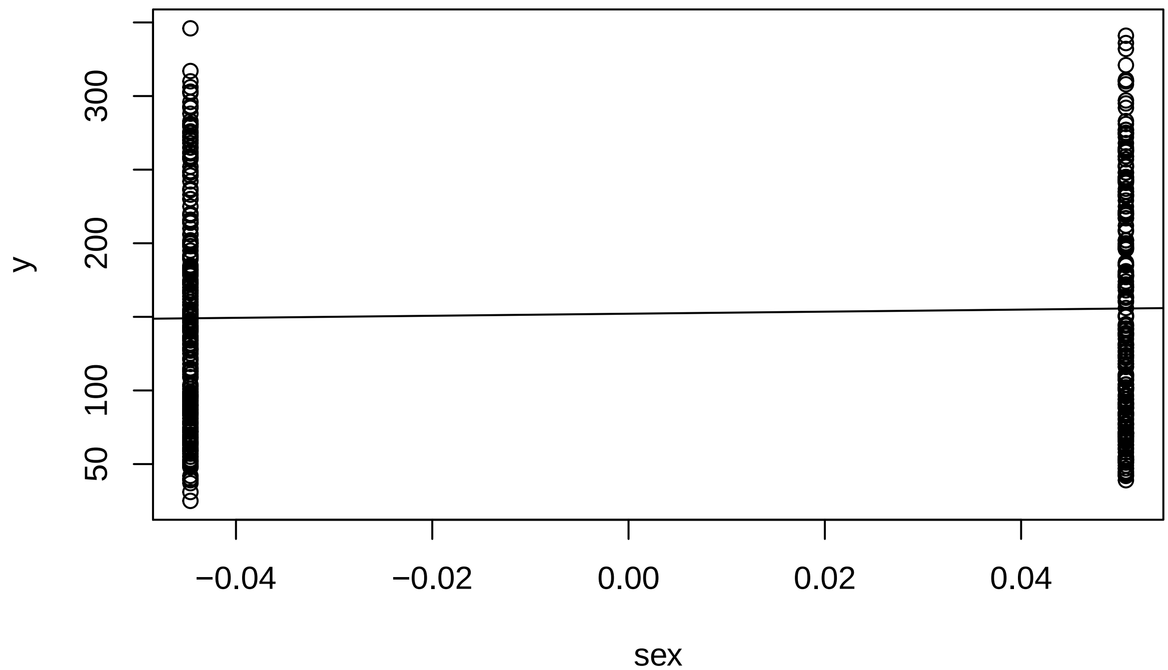
```
library(lars)
library(glmnet)
data("diabetes")
attach(diabetes)
```

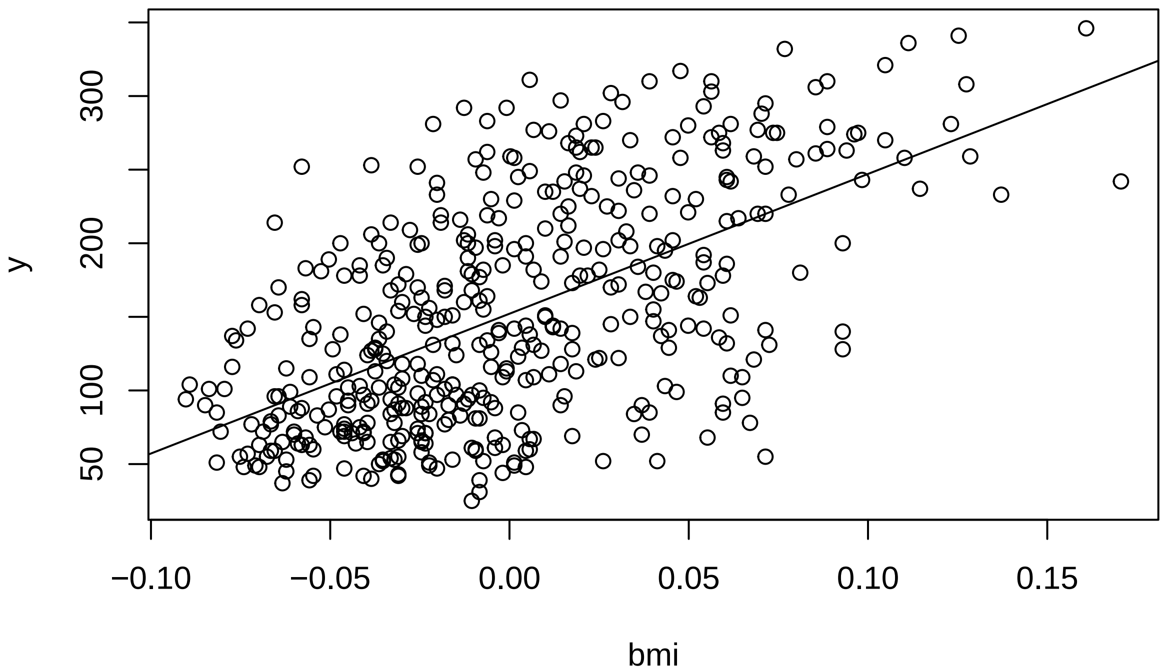
Exercise 2:

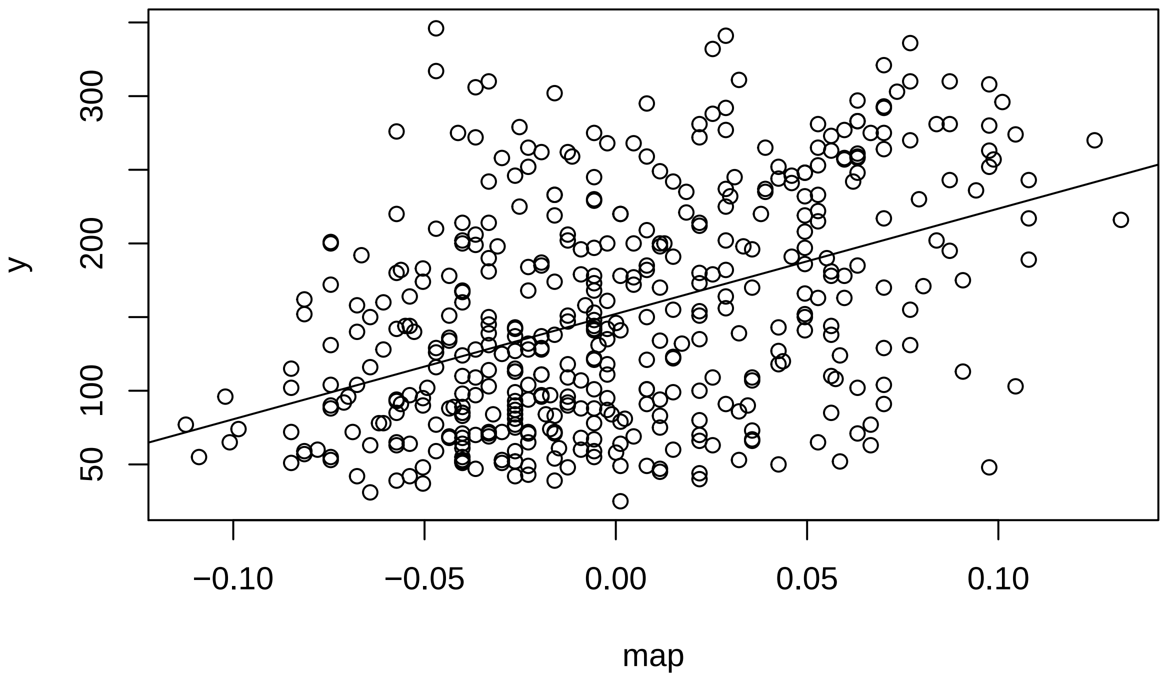
```
for (i in 1:10){
  plot(x[,i],y, xlab =colnames(x)[i])
  abline(lm(y~x[,i]))
}
```



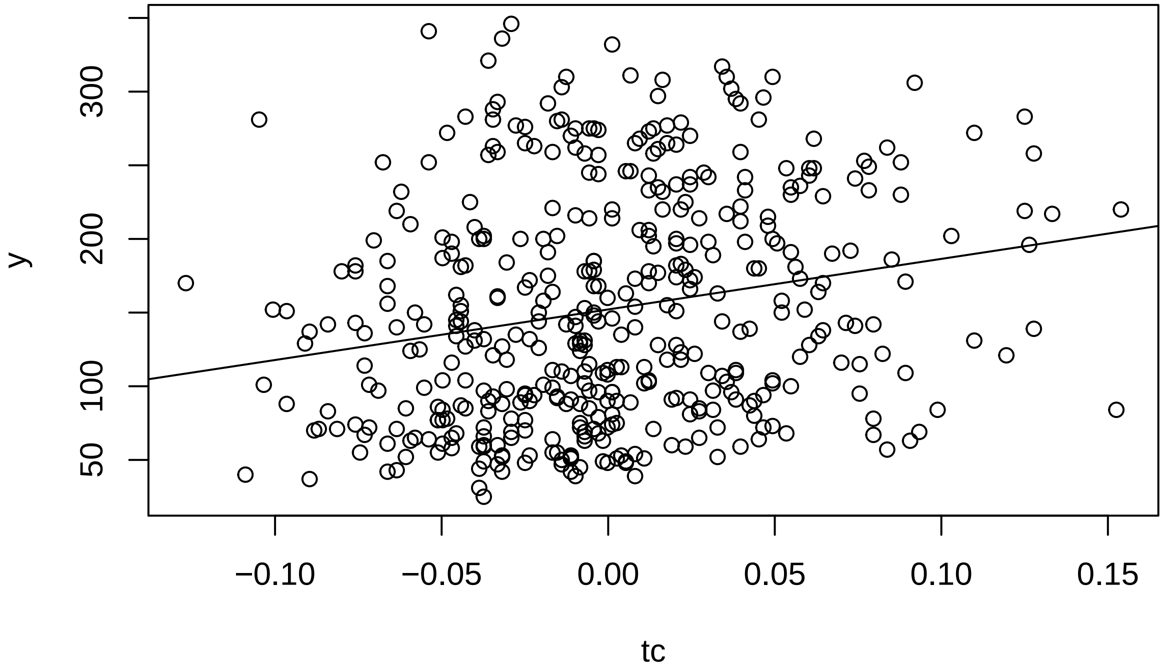
y
50
100
200
300
−0.10
−0.05
0.00
0.05
0.10
age


y
50
100
200
300
−0.04
−0.02
0.00
0.02
0.04
sex

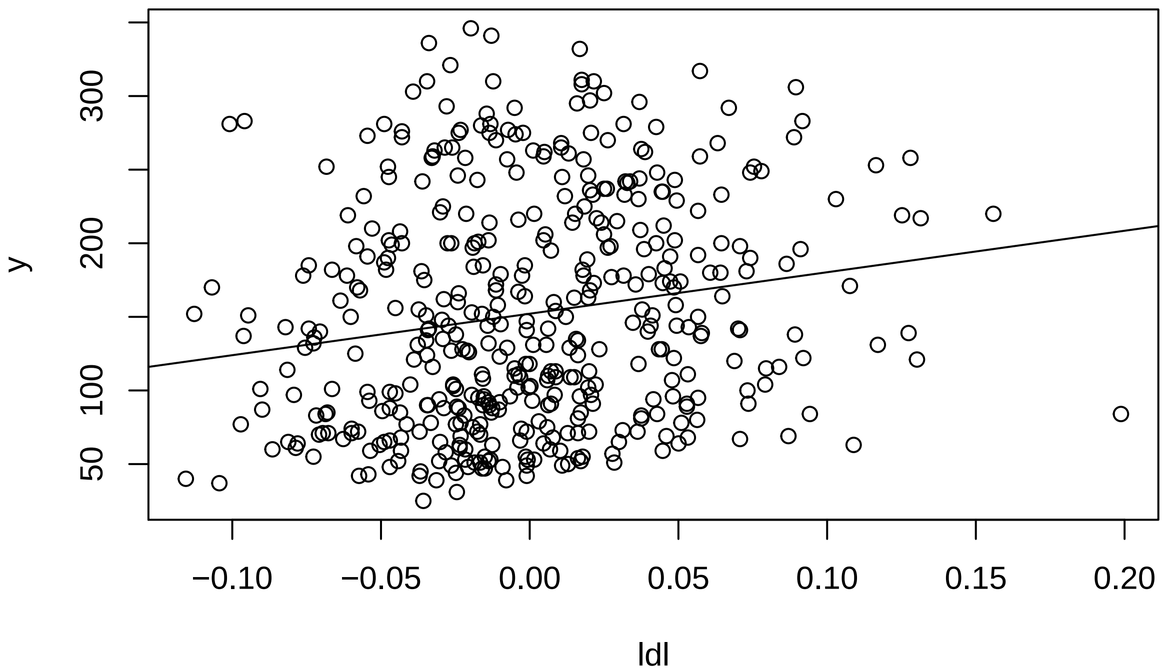

y
300
250
200
150
100
50
−0.10
−0.05
0.00
0.05
0.10
0.15
bmi



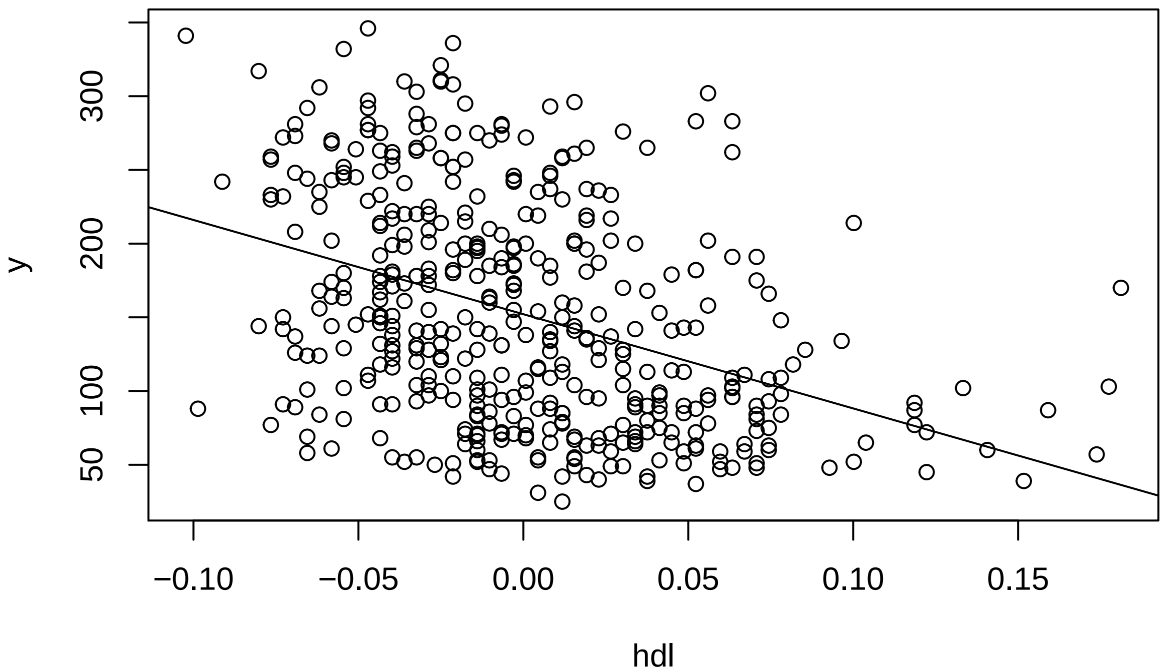
y
50
100
200
300
−0.10
−0.05
0.00
0.05
0.10
map



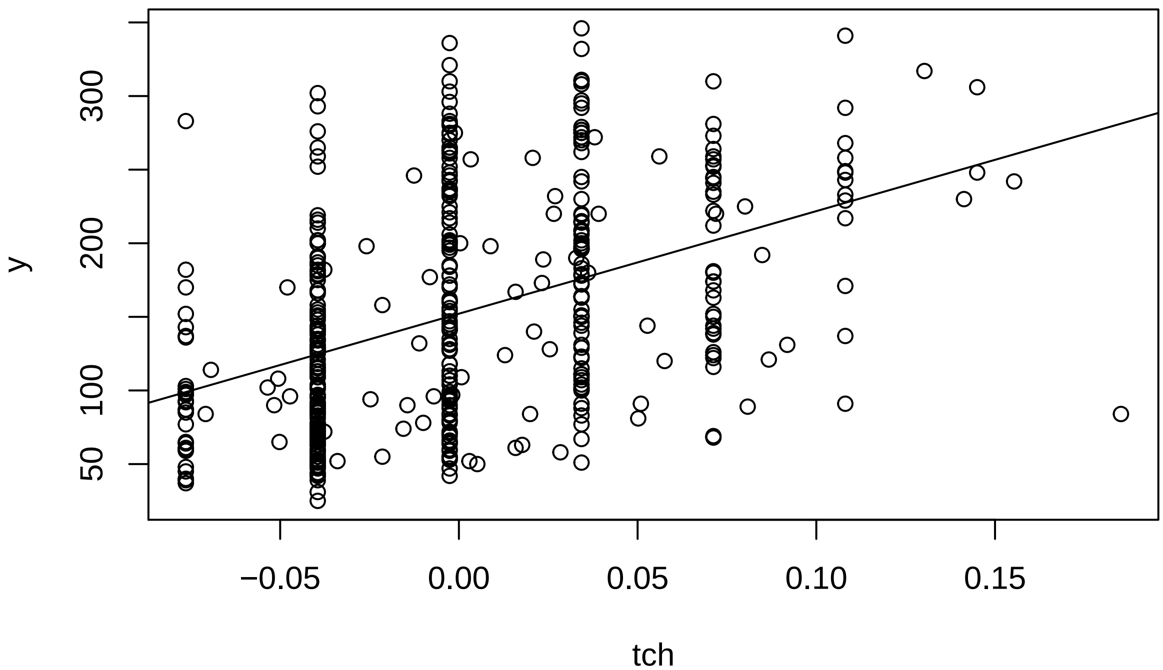
y
50
100
200
300
−0.10
−0.05
0.00
0.05
0.10
0.15
tc



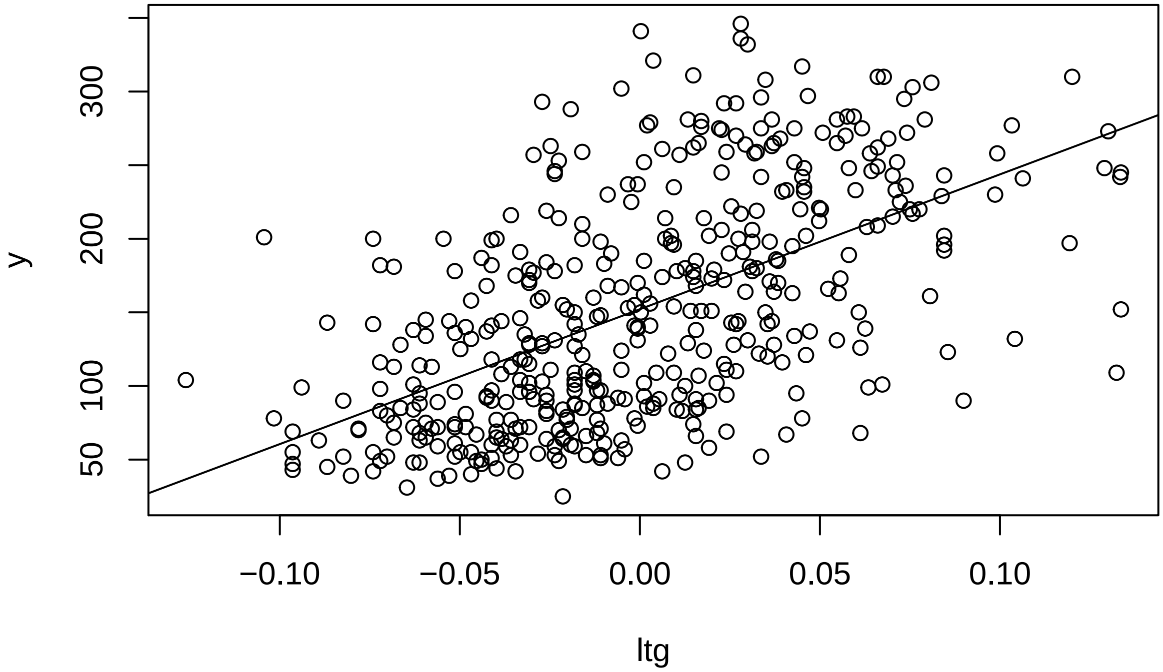
y
300
200
100
50
−0.10
−0.05
0.00
0.05
0.10
0.15
0.20
ldl

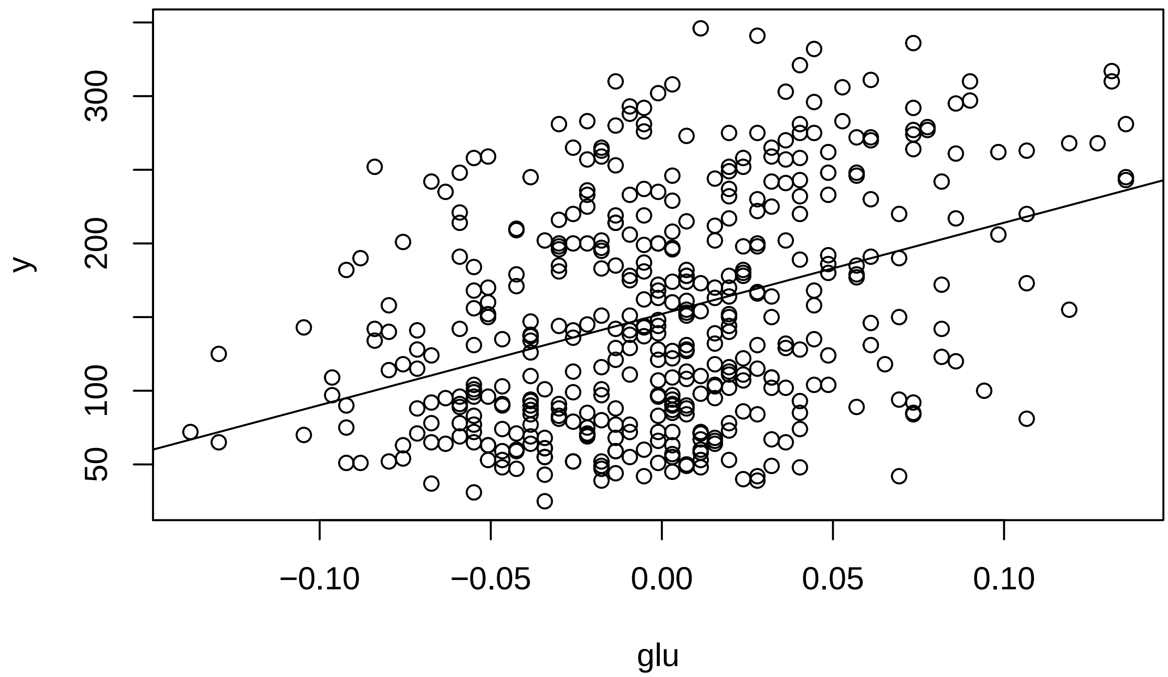


50
100
200
300
y
−0.10
−0.05
0.00
0.05
0.10
0.15
hdl



y
50
100
200
300
−0.05
0.00
0.05
0.10
0.15
tch


y
50
100
200
300
−0.10
−0.05
0.00
0.05
0.10
ltg



Exercise 3:

```r
bench<-lm(y ~ x, data=diabetes)
class(x)
```

```
## [1] "AsIs"
```

```r
summary(bench)
```

```
##
## Call:
## lm(formula = y ~ x, data = diabetes)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -155.829  -38.534   -0.227   37.806  151.355
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  152.133      2.576  59.061  < 2e-16 ***
## xage         -10.012     59.749  -0.168 0.867000
## xsex        -239.819     61.222  -3.917 0.000104 ***
## xbmi         519.840     66.534   7.813 4.30e-14 ***
```

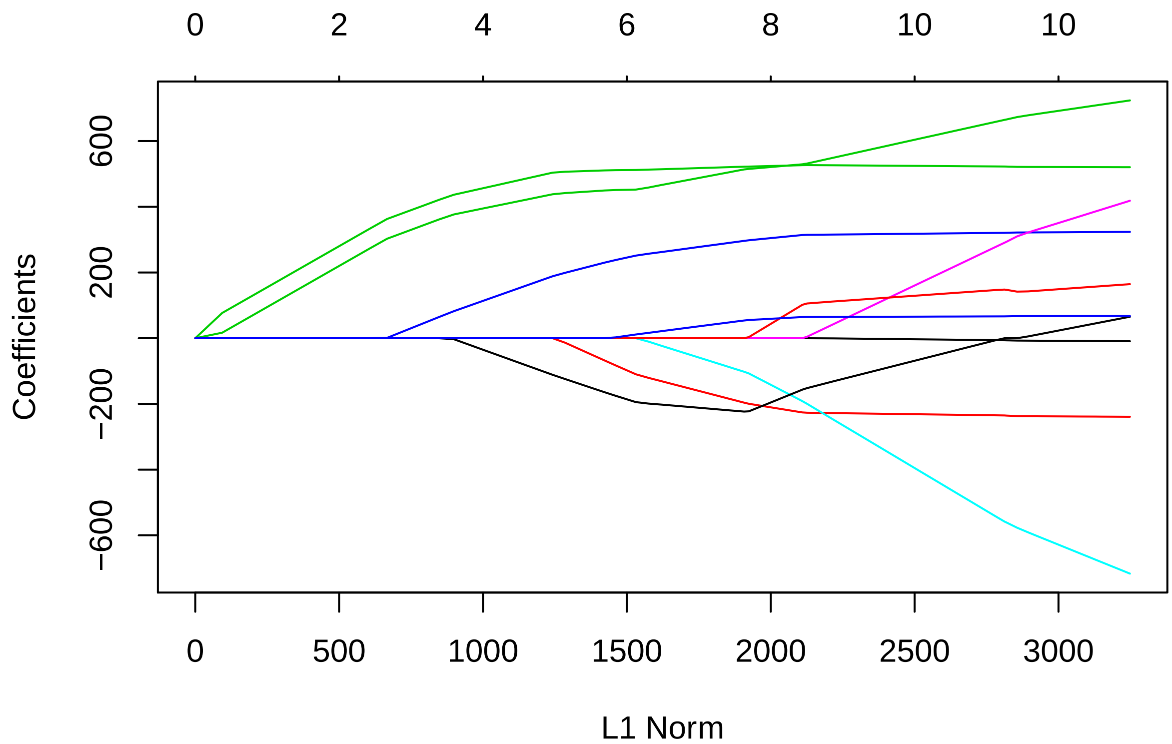
```
## xmap         324.390     65.422   4.958 1.02e-06 ***
## xtc         -792.184    416.684  -1.901 0.057947 .
## xldl         476.746    339.035   1.406 0.160389
## xhdl         101.045    212.533   0.475 0.634721
## xtch         177.064    161.476   1.097 0.273456
## xltg         751.279    171.902   4.370 1.56e-05 ***
## xglu          67.625     65.984   1.025 0.305998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.15 on 431 degrees of freedom
## Multiple R-squared:  0.5177, Adjusted R-squared:  0.5066
## F-statistic: 46.27 on 10 and 431 DF,  p-value: < 2.2e-16
```

Exercise 4:

```
glm1<-glmnet(diabetes$x, diabetes$y)
summary(glm1)
```

	Length	Class	Mode
a0	88	-none-	numeric
beta	880	dgCMatrix	S4
df	88	-none-	numeric
dim	2	-none-	numeric
lambda	88	-none-	numeric
dev.ratio	88	-none-	numeric
nulldev	1	-none-	numeric
npasses	1	-none-	numeric
jerr	1	-none-	numeric
offset	1	-none-	logical
call	3	-none-	call
nobs	1	-none-	numeric

```
plot(glm1)
```




```r
glm1$beta
```

```
## 10 x 88 sparse Matrix of class "dgCMatrix"

##    [[ suppressing 88 column names 's0', 's1', 's2' ... ]]

##
## age .  .          .          .        .        .        .
## sex .  .          .          .        .        .        .
## bmi . 76.88293 130.02092 178.4416 222.5606 262.7603 299.3887 332.7819
## map .  .          .          .        .        .        .
## tc  .  .          .          .        .        .        .
## ldl .  .          .          .        .        .        .
## hdl .  .          .          .        .        .        .
## tch .  .          .          .        .        .        .
## ltg . 16.74669  69.89093 118.3125 162.4325 202.6329 239.2621 272.6289
## glu .  .          .          .        .        .        .
##
## age   .          .          .          .         .          .          .
## sex   .          .          .          .         .          .          .
## bmi 362.882285 384.18816 403.56947 421.2928 436.738959 447.18511 456.78809
## map   1.049765  24.26724  45.37674  64.6454  82.307193  98.99238 114.15538
## tc    .          .          .          .         .          .          .
## ldl   .          .          .          .         .          .          .
## hdl   .          .          .          .        -3.104737 -20.12834 -35.57045
```

```
## tch   .          .          .          .         .          .          .
## ltg 302.763925 324.19336 343.75113 361.5294 376.763057 386.37120 395.13120
## glu   .          .          .          .         .          .          .
##
## age   .         .         .         .         .         .         .
## sex   .         .         .         .         .         .         .
## bmi 465.53779 473.51020 480.77435 487.39318 493.42401  498.9191  503.9260
## map 127.97136 140.55998 152.03025 162.48154 172.00436  180.6812  188.5872
## tc    .         .         .         .         .         .         .
## ldl   .         .         .         .         .         .         .
## hdl -49.64078 -62.46113 -74.14256 -84.78625 -94.48438 -103.3210 -111.3725
## tch   .         .         .         .         .         .         .
## ltg 403.11304 410.38579 417.01246 423.05042 428.55200  433.5648  438.1323
## glu   .         .         .         .         .         .         .
##
## age    .          .          .          .          .           .
## sex  -13.59236  -33.35229  -51.33499  -67.72019  -82.922210  -96.957619
## bmi  506.66309  508.13935  509.50955  510.75796  511.428467  511.695073
## map  199.02672  210.34606  220.64138  230.02210  238.119244  245.071753
## tc     .          .          .          .          .           .
## ldl    .          .          .          .          .           .
## hdl -124.16053 -138.84433 -152.19412 -164.35796 -175.401082 -185.364621
## tch    .          .          .          .          .           .
## ltg  441.69986  444.59064  447.22918  449.63335  451.053356  451.723115
## glu    .          .          .          .           2.612794    7.186267
##
## age    .          .          .          .          .          .
## sex -109.75014 -120.78251 -130.70471 -139.77055 -148.03029 -155.55625
## bmi  511.94125  513.17920  514.21854  515.29864  516.28780  517.18917
## map  251.41407  257.15022  262.23795  266.95584  271.25644  275.17502
## tc     .        -10.53351  -22.94188  -33.91082  -43.89945  -53.00065
## ldl    .          .          .          .          .          .
## hdl -194.45492 -199.00251 -201.97255 -204.92948 -207.62737 -210.08564
## tch    .          .          .          .          .          .
## ltg  452.31596  458.64993  466.27153  472.84408  478.82660  484.27758
## glu   11.35493   16.40720   21.44909   25.95892   30.06523   33.80671
##
## age    .          .          .          .         .          .
## sex -162.41362 -168.66181 -174.35492 -179.5423 -184.26879 -188.54837
## bmi  518.01046  518.75878  519.44063  520.0619  520.62799  521.17946
## map  278.74548  281.99875  284.96301  287.6639  290.12491  292.36363
## tc   -61.29331  -68.84929  -75.73400  -82.0071  -87.72292  -92.98377
## ldl    .          .          .          .         .          .
## hdl -212.32551 -214.36641 -216.22600 -217.9204 -219.46424 -220.81702
## tch    .          .          .          .         .          .
## ltg  489.24430  493.76979  497.89325  501.6504  505.07376  508.26094
## glu   37.21581   40.32205   43.15234   45.7312   48.08096   50.20429
##
## age    .          .           .          .          .          .
## sex -192.47489 -196.05252 -199.613568 -202.92429 -205.94926 -208.70759
## bmi  521.64423  522.06966  522.333887  522.90423  523.38742  523.82271
## map  294.40618  296.26816  298.122927  300.14848  301.97829  303.64301
## tc   -97.72765 -102.04702 -107.259553 -117.92398 -127.52172 -136.20923
## ldl    .          .           .          .          .          .
```

```
## hdl -222.10137 -223.27372 -222.507843 -214.12178 -206.65172 -199.91868
## tch    .          .          3.436326   15.92492   27.10400   37.21320
## ltg  511.10045  513.68443  515.314123  517.05725  518.65315  520.09746
## glu   52.15790   53.93678   55.427498   56.54720   57.58378   58.52938
##
## age    .          .          .          .          .          .
## sex -211.21122 -213.50043 -215.57818 -217.47954 -219.20424 -220.78374
## bmi  524.24174  524.60465  524.95444  525.25300  525.54420  525.78823
## map  305.17148  306.55450  307.82443  308.97137  310.02607  310.97637
## tc  -144.39883 -151.63957 -158.46116 -164.45454 -170.12521 -175.07730
## ldl    .          .          .          .          .          .
## hdl -193.43545 -187.81013 -182.39904 -177.75264 -173.25088 -169.42518
## tch   46.78812   55.21756   63.19674   70.16906   76.80345   82.55659
## ltg  521.46099  522.66556  523.80143  524.79973  525.74382  526.57119
## glu   59.38637   60.17137   60.88245   61.53514   62.12526   62.66858
##
## age    .          .          .          .          .         -0.4845628
## sex -222.21550 -223.52767 -224.71648 -225.79672 -226.77490 -227.9024401
## bmi  526.03050  526.22847  526.42974  526.61997  526.80091  526.3505753
## map  311.85224  312.63865  313.36588  314.03207  314.68215  315.4683407
## tc  -179.78705 -183.87041 -187.77618 -191.41981 -199.57382 -248.2968993
## ldl    .          .          .          .          4.46890   43.1148311
## hdl -165.68510 -162.54705 -159.44704 -156.51457 -151.88564 -130.9537960
## tch   88.06767   92.80035   97.37108  101.64839  105.70126  111.8901407
## ltg  527.35489  528.04154  528.69064  529.29685  531.63068  549.5611438
## glu   63.15828   63.61112   64.01749   64.38638   64.62026   64.7845109
##
## age   -1.415403   -2.166592   -2.851997   -3.478839   -4.046617
## sex -228.969566 -229.918225 -230.783757 -231.575218 -232.292272
## bmi  525.724415  525.235634  524.787698  524.373244  524.004780
## map  316.260073  316.968077  317.614269  318.205707  318.740773
## tc  -294.225344 -335.630187 -373.571736 -408.668750 -439.885861
## ldl   79.828702  112.904016  143.206893  171.223703  196.163421
## hdl -111.472046  -93.897882  -77.785852  -62.862734  -49.616274
## tch  117.279741  122.172495  126.662388  130.831124  134.517261
## ltg  566.628672  581.986597  596.059550  609.076565  620.655759
## glu   65.084847   65.311863   65.518610   65.706745   65.878511
##
## age   -4.566271   -5.038575   -5.467969   -5.860508   -6.164782
## sex -232.948431 -233.544859 -234.087145 -234.582820 -235.082642
## bmi  523.662765  523.354360  523.075946  522.818763  522.574054
## map  319.230937  319.676230  320.080903  320.451094  320.794781
## tc  -468.852729 -494.980350 -518.573846 -540.362823 -558.072609
## ldl  219.291197  240.158959  259.008296  276.408054  290.657723
## hdl  -37.305653  -26.210944  -16.199756   -6.943665    .
## tch  137.953084  141.044772  143.830708  146.412142  148.040169
## ltg  631.399432  641.090408  649.841736  657.923258  664.564459
## glu   66.034778   66.177272   66.307195   66.425441   66.526597
##
## age   -6.405643   -6.62609   -6.829204   -7.015121   -7.184096   -7.441081
## sex -235.520228 -235.98713 -236.400627 -236.792790 -237.137302 -237.450373
## bmi  522.397120  522.13081  521.903038  521.670772  521.479283  521.195477
## map  321.036432  321.24746  321.438205  321.611379  321.769492  321.933108
## tc  -561.236080 -565.68036 -569.561840 -573.394887 -576.648375 -590.910438
```

```
## ldl  293.900176  298.49852  302.520461  306.493473  309.865596  321.830108
## hdl    .           .          .           .           .           5.492946
## tch  147.080457  145.55907  144.236190  142.906289  141.793925  142.538777
## ltg  665.977635  667.99392  669.752138  671.497725  672.972979  678.454078
## glu   66.662225   66.80703   66.936312   67.059604   67.167472   67.274826
##
## age   -7.659526   -7.857256   -8.038924   -8.203286   -8.35194   -8.488398
## sex -237.649098 -237.828235 -237.993264 -238.142233 -238.27663 -238.400297
## bmi  521.101023  521.019499  520.941181  520.872947  520.81377  520.757143
## map  322.137930  322.325037  322.497326  322.652898  322.79336  322.922585
## tc  -606.350411 -620.296633 -633.351965 -644.976480 -655.31543 -664.972371
## ldl  333.728223  344.457292  354.500990  363.444026  371.39770  378.826474
## hdl   12.832719   19.486566   25.710710   31.256119   36.19204   40.799453
## tch  145.212026  147.660815  149.944879  151.984740  153.80560  155.501022
## ltg  684.028537  689.057010  693.767034  697.958786  701.68486  705.166979
## glu   67.308329   67.337779   67.364398   67.388825   67.41122   67.431457
##
## age   -8.613371   -8.725468   -8.827985   -8.923698   -9.008557
## sex -238.513748 -238.614993 -238.707682 -238.794902 -238.871577
## bmi  520.703826  520.659998  520.619095  520.575646  520.542320
## map  323.041061  323.146876  323.243836  323.334944  323.415099
## tc  -673.915878 -681.653876 -688.799997 -695.844049 -701.716448
## ldl  385.706716  391.659643  397.156435  402.574767  407.092329
## hdl   45.064191   48.759346   52.171545   55.527711   58.331889
## tch  157.067258  158.431813  159.691039  160.919288  161.954649
## ltg  708.393088  711.181169  713.756519  716.299417  718.415388
## glu   67.449828   67.466826   67.482203   67.495886   67.508713
##
## age   -9.085676   -9.157794   -9.220679
## sex -238.941193 -239.006809 -239.063410
## bmi  520.512538  520.480489  520.458638
## map  323.488031  323.556704  323.616100
## tc  -707.033644 -712.304505 -716.483385
## ldl  411.181675  415.235190  418.448411
## hdl   60.872788   63.385923   65.386847
## tch  162.894885  163.816744  164.562823
## ltg  720.330659  722.232732  723.735577
## glu   67.520347   67.530678   67.540420
```

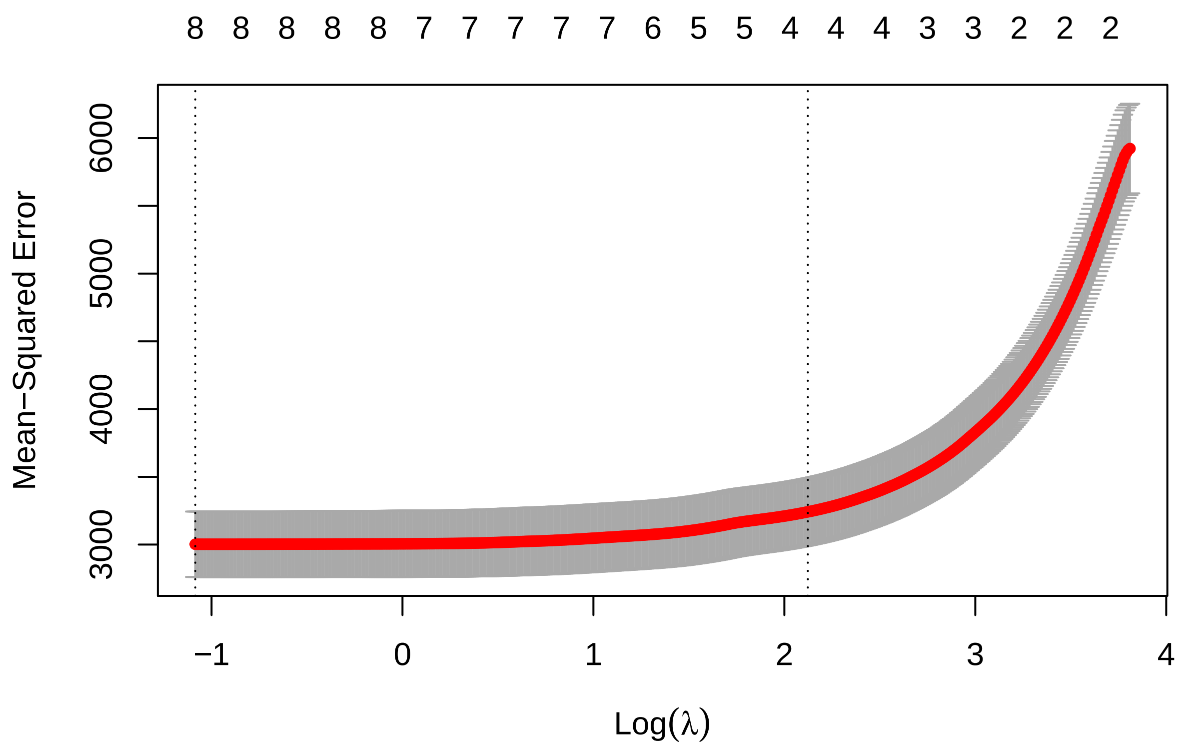
Exercise 5:

```
cvglm1<-cv.glmnet(diabetes$x, diabetes$y, alpha=1, nlambda=1000)
summary(cvglm1)
```

	Length	Class	Mode
lambda	532	-none-	numeric
cvm	532	-none-	numeric
cvsd	532	-none-	numeric
cvup	532	-none-	numeric
cvlo	532	-none-	numeric
nzero	532	-none-	numeric

	Length	Class	Mode
call	5	-none-	call
name	1	-none-	character
glmnet.fit	12	elnet	list
lambda.min	1	-none-	numeric
lambda.1se	1	-none-	numeric

```
plot(cvglm1)
```




```
cvglm1$lambda.min
```

```
## [1] 0.3377755
```

Exercise 6:

```
minValue<- glmnet( diabetes$x, diabetes$y, alpha=1, lambda=cvglm1$lambda.min)
summary(minValue)
```

	Length	Class	Mode
a0	1	-none-	numeric
beta	10	dgCMatrix	S4
df	1	-none-	numeric
dim	2	-none-	numeric
lambda	1	-none-	numeric
dev.ratio	1	-none-	numeric
nulldev	1	-none-	numeric
npasses	1	-none-	numeric
jerr	1	-none-	numeric
offset	1	-none-	logical
call	5	-none-	call
nobs	1	-none-	numeric

```r
minValue$beta
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##             s0
## age    .
## sex -223.08368
## bmi  526.09917
## map  312.37829
## tc  -182.35395
## ldl    .
## hdl -163.72954
## tch   91.08108
## ltg  527.84490
## glu   63.41608
```

Exercise 7:

```r
cvglm1$lambda.1se
```

```
## [1] 8.356444
```

```r
lambda1SE<- glmnet(diabetes$x, diabetes$y, alpha=1, lambda=cvglm1$lambda.1se)
summary(lambda1SE)
```

	Length	Class	Mode
a0	1	-none-	numeric
beta	10	dgCMatrix	S4
df	1	-none-	numeric
dim	2	-none-	numeric
lambda	1	-none-	numeric
dev.ratio	1	-none-	numeric
nulldev	1	-none-	numeric
npasses	1	-none-	numeric
jerr	1	-none-	numeric

	Length	Class	Mode
offset	1	-none-	logical
call	5	-none-	call
nobs	1	-none-	numeric

```
lambda1SE$beta
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##            s0
## age   .
## sex   .
## bmi 488.31841
## map 163.80820
## tc    .
## ldl   .
## hdl -86.12416
## tch   .
## ltg 423.80501
## glu   .
```

Exercise 8:

```
second<-lm(y ~ x2, data=diabetes)
summary(second)
```

```
##
## Call:
## lm(formula = y ~ x2, data = diabetes)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -158.216  -30.809   -3.857   31.348  153.946
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   152.133      2.532  60.086  < 2e-16 ***
## x2age          50.721     65.513   0.774   0.4393
## x2sex        -267.344     65.270  -4.096 5.15e-05 ***
## x2bmi         460.721     84.601   5.446 9.32e-08 ***
## x2map         342.933     72.447   4.734 3.13e-06 ***
## x2tc        -3599.542  60575.187  -0.059   0.9526
## x2ldl        3028.281  53238.699   0.057   0.9547
## x2hdl        1103.047  22636.179   0.049   0.9612
## x2tch          74.937    275.807   0.272   0.7860
## x2ltg        1828.210  19914.504   0.092   0.9269
## x2glu          62.754     70.398   0.891   0.3733
## x2age^2        67.691     69.470   0.974   0.3305
## x2bmi^2        45.849     83.288   0.550   0.5823
## x2map^2        -8.460     71.652  -0.118   0.9061
## x2tc^2       6668.449   7059.159   0.945   0.3454
```

```
## x2ldl^2      3583.174   5326.148   0.673   0.5015
## x2hdl^2      1731.821   1590.574   1.089   0.2769
## x2tch^2       773.374    606.967   1.274   0.2034
## x2ltg^2      1451.581   1730.103   0.839   0.4020
## x2glu^2       114.149     94.122   1.213   0.2260
## x2age:sex     148.678     73.407   2.025   0.0435 *
## x2age:bmi     -18.052     79.620  -0.227   0.8208
## x2age:map      18.534     76.303   0.243   0.8082
## x2age:tc     -158.891    617.109  -0.257   0.7970
## x2age:ldl     -67.285    494.527  -0.136   0.8918
## x2age:hdl     209.245    280.614   0.746   0.4563
## x2age:tch     184.960    210.330   0.879   0.3798
## x2age:ltg     124.667    223.765   0.557   0.5778
## x2age:glu      62.575     80.377   0.779   0.4367
## x2sex:bmi      64.612     77.902   0.829   0.4074
## x2sex:map      88.472     74.744   1.184   0.2373
## x2sex:tc      433.598    590.709   0.734   0.4634
## x2sex:ldl    -352.823    468.951  -0.752   0.4523
## x2sex:hdl    -124.731    273.870  -0.455   0.6491
## x2sex:tch    -131.223    199.714  -0.657   0.5115
## x2sex:ltg    -118.995    226.493  -0.525   0.5996
## x2sex:glu      45.758     73.650   0.621   0.5348
## x2bmi:map     154.720     86.340   1.792   0.0739 .
## x2bmi:tc     -302.045    667.930  -0.452   0.6514
## x2bmi:ldl     241.540    561.026   0.431   0.6671
## x2bmi:hdl     121.942    329.884   0.370   0.7118
## x2bmi:tch     -33.445    230.836  -0.145   0.8849
## x2bmi:ltg     114.673    255.987   0.448   0.6544
## x2bmi:glu      23.377     91.037   0.257   0.7975
## x2map:tc      478.303    682.264   0.701   0.4837
## x2map:ldl    -326.740    574.317  -0.569   0.5697
## x2map:hdl    -187.305    309.589  -0.605   0.5455
## x2map:tch     -58.294    198.601  -0.294   0.7693
## x2map:ltg    -154.795    271.966  -0.569   0.5696
## x2map:glu    -133.476     91.314  -1.462   0.1447
## x2tc:ldl    -9313.775  11771.220  -0.791   0.4293
## x2tc:hdl    -3932.025   3816.572  -1.030   0.3036
## x2tc:tch    -2205.910   1761.843  -1.252   0.2113
## x2tc:ltg    -3801.442  13166.091  -0.289   0.7729
## x2tc:glu     -176.295    595.459  -0.296   0.7673
## x2ldl:hdl    2642.645   3165.926   0.835   0.4044
## x2ldl:tch    1206.822   1470.512   0.821   0.4123
## x2ldl:ltg    2773.697  10960.214   0.253   0.8004
## x2ldl:glu      85.626    505.102   0.170   0.8655
## x2hdl:tch    1188.406   1002.242   1.186   0.2365
## x2hdl:ltg    1467.845   4609.793   0.318   0.7503
## x2hdl:glu     217.541    296.749   0.733   0.4640
## x2tch:ltg     389.805    624.671   0.624   0.5330
## x2tch:glu     235.693    235.064   1.003   0.3167
## x2ltg:glu      83.525    264.726   0.316   0.7525
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 53.23 on 377 degrees of freedom
```

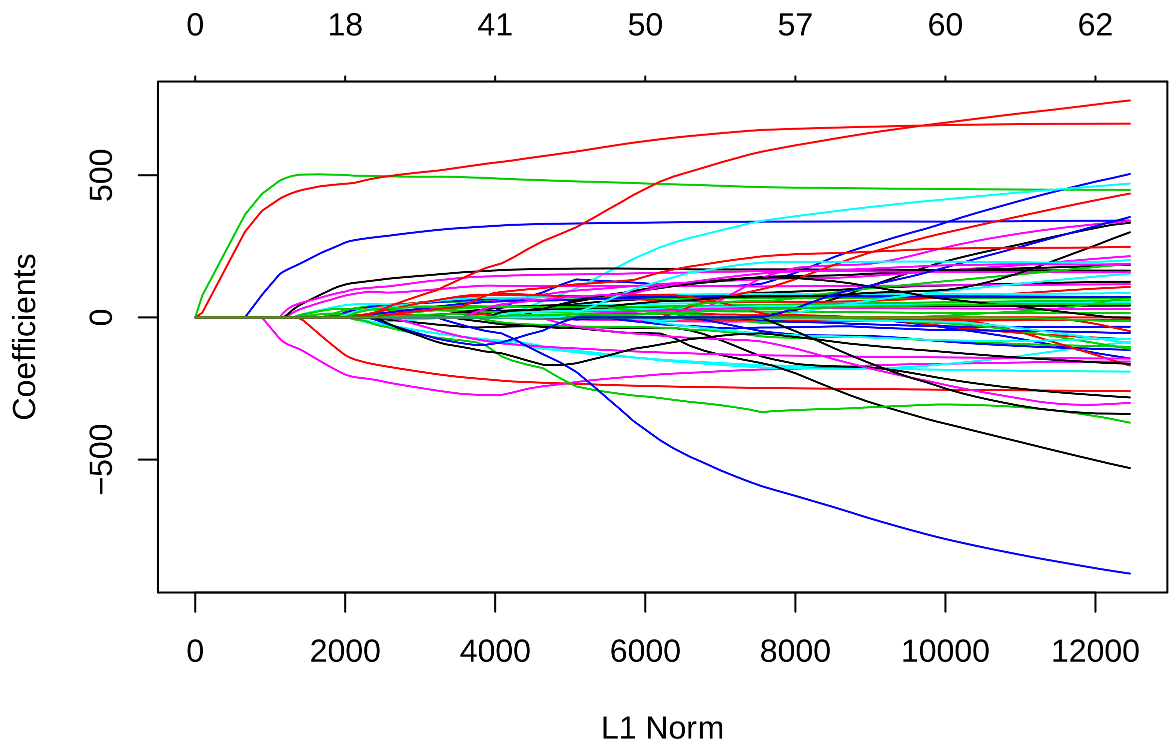
```
## Multiple R-squared:  0.5924, Adjusted R-squared:  0.5233
## F-statistic: 8.563 on 64 and 377 DF,  p-value: < 2.2e-16
```

Exercise 9:

```r
glm2<-glmnet(x2, y)
summary(glm2)
```

	Length	Class	Mode
a0	100	-none-	numeric
beta	6400	dgCMatrix	S4
df	100	-none-	numeric
dim	2	-none-	numeric
lambda	100	-none-	numeric
dev.ratio	100	-none-	numeric
nulldev	1	-none-	numeric
npasses	1	-none-	numeric
jerr	1	-none-	numeric
offset	1	-none-	logical
call	3	-none-	call
nobs	1	-none-	numeric

```r
plot(glm2)
```

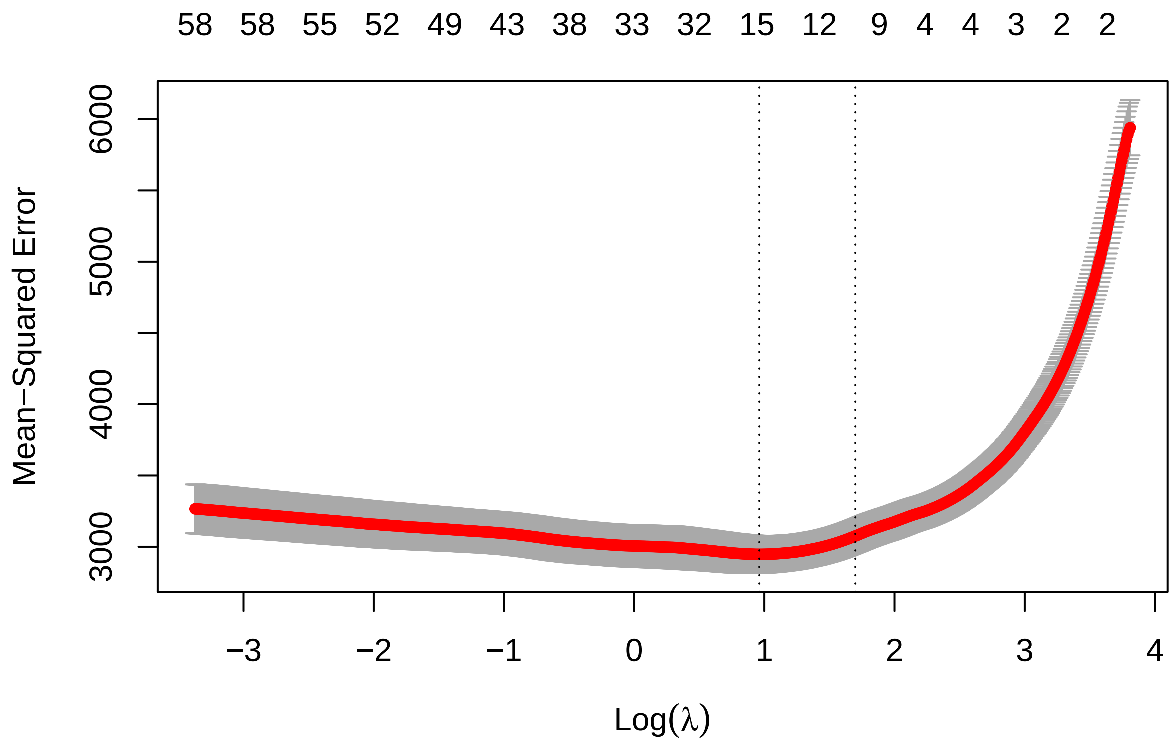


Exercise 10:

```
cvglm2<-cv.glmnet(x2, y, alpha=1, nlambda=1000)
summary(cvglm2)
```

	Length	Class	Mode
lambda	780	-none-	numeric
cvm	780	-none-	numeric
cvsd	780	-none-	numeric
cvup	780	-none-	numeric
cvlo	780	-none-	numeric
nzero	780	-none-	numeric
call	5	-none-	call
name	1	-none-	character
glmnet.fit	12	elnet	list
lambda.min	1	-none-	numeric
lambda.1se	1	-none-	numeric

```
plot(cvglm2)
```




```
cvglm2$lambda.min
```

```
## [1] 2.615273
```

```
minValue2<- glmnet( x2, y, alpha=1, lambda=cvglm2$lambda.min)
summary(minValue2)
```

	Length	Class	Mode
a0	1	-none-	numeric
beta	64	dgCMatrix	S4
df	1	-none-	numeric
dim	2	-none-	numeric
lambda	1	-none-	numeric
dev.ratio	1	-none-	numeric
nulldev	1	-none-	numeric
npasses	1	-none-	numeric
jerr	1	-none-	numeric
offset	1	-none-	logical
call	5	-none-	call
nobs	1	-none-	numeric

```r
minValue2$beta
```

```
## 64 x 1 sparse Matrix of class "dgCMatrix"
##                  s0
## age        .
## sex     -128.304096
## bmi      500.934368
## map      260.961328
## tc         .
## ldl        .
## hdl     -198.230814
## tch        .
## ltg      469.533077
## glu       24.105605
## age^2     15.769532
## bmi^2     42.141611
## map^2      .
## tc^2       .
## ldl^2      .
## hdl^2      .
## tch^2      .
## ltg^2      .
## glu^2     75.308603
## age:sex  114.070727
## age:bmi    .
## age:map   30.542265
## age:tc     .
## age:ldl    .
## age:hdl    .
## age:tch    .
## age:ltg   11.929952
## age:glu    9.766166
## sex:bmi    .
## sex:map    6.349543
## sex:tc     .
## sex:ldl    .
## sex:hdl    .
## sex:tch    .
## sex:ltg    .
## sex:glu    .
## bmi:map   89.697462
## bmi:tc     .
## bmi:ldl    .
## bmi:hdl    .
## bmi:tch    .
## bmi:ltg    .
## bmi:glu    .
## map:tc     .
## map:ldl    .
## map:hdl    .
## map:tch    .
## map:ltg    .
## map:glu    .
```

```
## tc:ldl      .
## tc:hdl      .
## tc:tch      .
## tc:ltg      .
## tc:glu      .
## ldl:hdl     .
## ldl:tch     .
## ldl:ltg     .
## ldl:glu     .
## hdl:tch     .
## hdl:ltg     .
## hdl:glu     .
## tch:ltg     .
## tch:glu     .
## ltg:glu     .
```