knn-project

Jordan Hilton

February 23, 2019

We're going to perform a simple linear regression analysis of the data first. Let's begin with loading the data

```
data<-read.csv("projectdata.csv")
```

Let's take a glance at the full data before we go any further:

```
head(data)
```

```
##     X Months.since.Last.Donation Number.of.Donations
## 1 619                          2                  50
## 2 664                          0                  13
## 3 441                          1                  16
## 4 160                          2                  20
## 5 358                          1                  24
## 6 335                          4                   4
##   Total.Volume.Donated..c.c.. Months.since.First.Donation
## 1                       12500                         98
## 2                        3250                         28
## 3                        4000                         35
## 4                        5000                         45
## 5                        6000                         77
## 6                        1000                          4
##   Made.Donation.in.March.2007
## 1                           1
## 2                           1
## 3                           1
## 4                           1
## 5                           0
## 6                           0
```

Note that it appears that every donation is 250 c.c., so the "total volume donated" column is a linear multiple of the "number of donations" column. Our linear regression will not like it if we include both columns, so we're going to drop the total volume column before proceeding. While we're at it let's drop the first "id" column since it's not relevant to analysis.

```
data<-data[-c(1,4)]
```

Now let's just create the full multivariate linear regression model and examine it;

```
fullmodel<-lm(Made.Donation.in.March.2007~Months.since.Last.Donation+Number.of.Donations+Months.since.F
summary(fullmodel)
```

```
##
## Call:
## lm(formula = Made.Donation.in.March.2007 ~ Months.since.Last.Donation +
##     Number.of.Donations + Months.since.First.Donation, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97312 -0.28375 -0.16537  0.01444  0.97220
##
```

```
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  0.3115899  0.0335642   9.283  < 2e-16 ***
## Months.since.Last.Donation  -0.0094864  0.0022395  -4.236 2.65e-05 ***
## Number.of.Donations          0.0221996  0.0040023   5.547 4.45e-08 ***
## Months.since.First.Donation -0.0030232  0.0009528  -3.173  0.00159 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4026 on 572 degrees of freedom
## Multiple R-squared:  0.1166, Adjusted R-squared:  0.112
## F-statistic: 25.17 on 3 and 572 DF,  p-value: 2.613e-15
```

While the model as a whole is statistically significant with a p-value of $2.6 * 10^{-15}$, the low R^2 indicates that our 3 independent variables don't do a good job of predicting blood donation in the linear model. Each model is significant in the full model, but let's formally check that it's appropriate to use each variable:

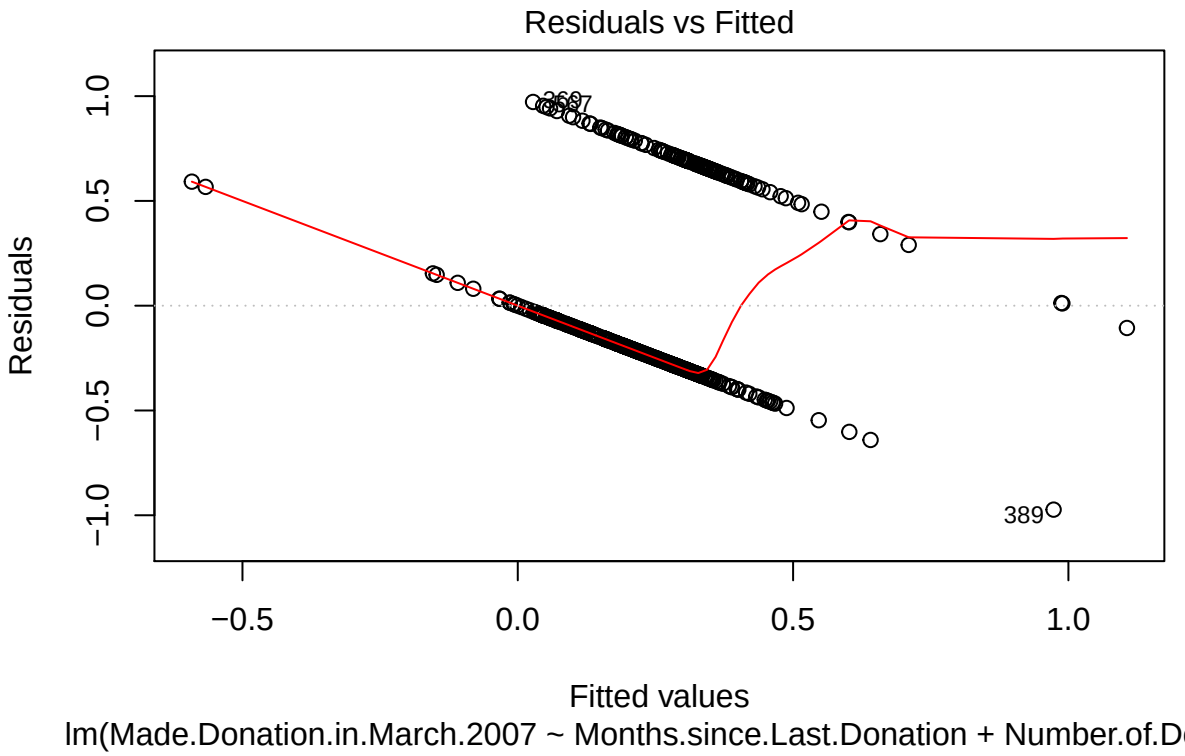
```
step(fullmodel)
```

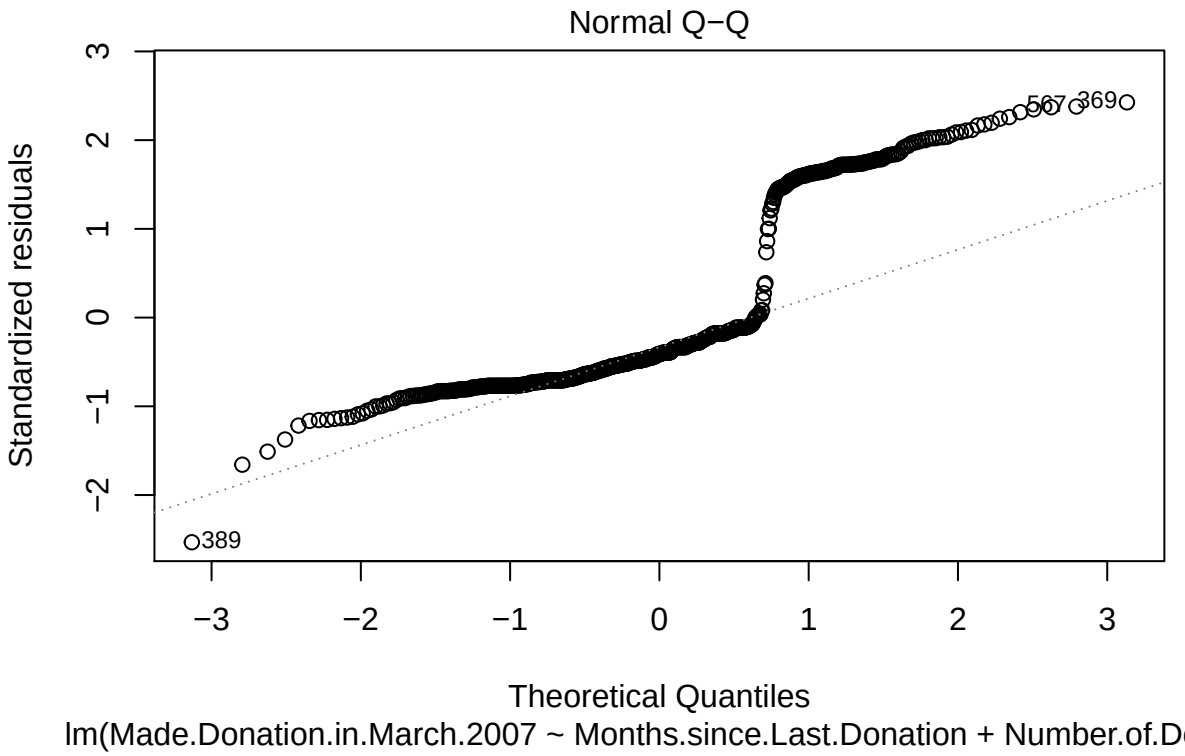
```
## Start:  AIC=-1044.21
## Made.Donation.in.March.2007 ~ Months.since.Last.Donation + Number.of.Donations +
##     Months.since.First.Donation
##
##                               Df Sum of Sq    RSS     AIC
## <none>                                     92.699 -1044.2
## - Months.since.First.Donation  1    1.6315 94.330 -1036.2
## - Months.since.Last.Donation   1    2.9079 95.607 -1028.4
## - Number.of.Donations          1    4.9860 97.685 -1016.0
```

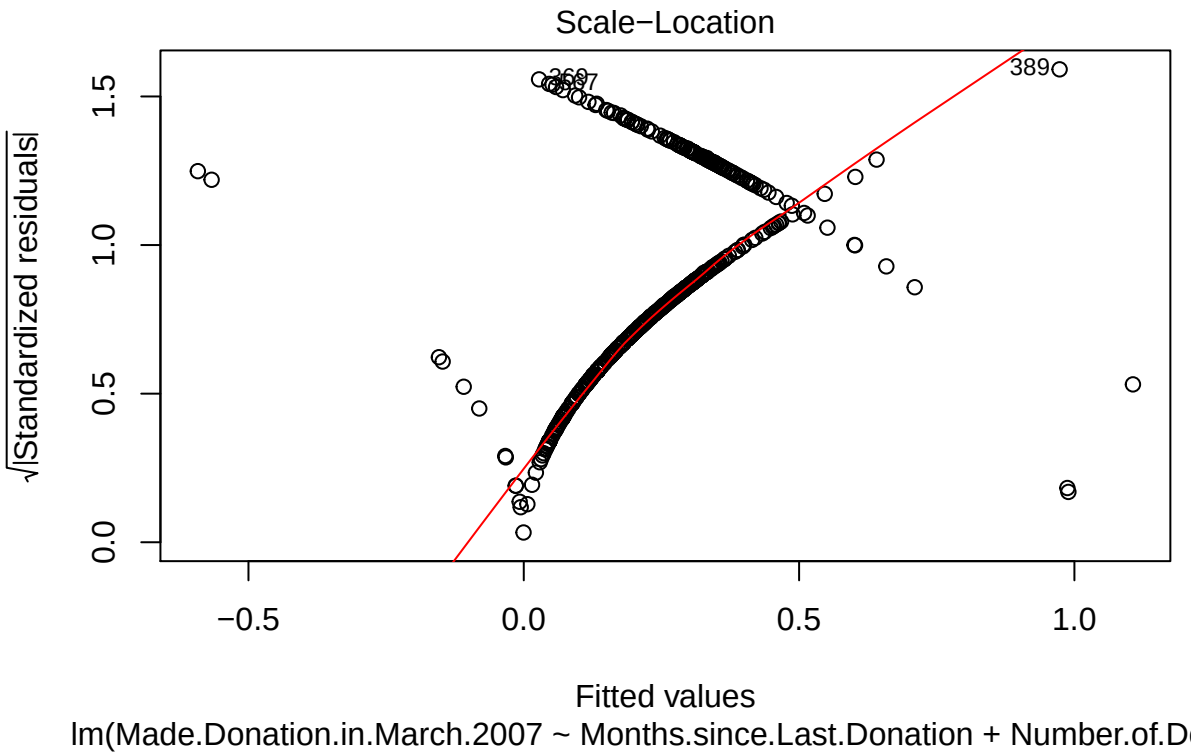
```
##
## Call:
## lm(formula = Made.Donation.in.March.2007 ~ Months.since.Last.Donation +
##     Number.of.Donations + Months.since.First.Donation, data = data)
##
## Coefficients:
##                 (Intercept)   Months.since.Last.Donation
##                    0.311590                    -0.009486
##         Number.of.Donations  Months.since.First.Donation
##                    0.022200                    -0.003023
```

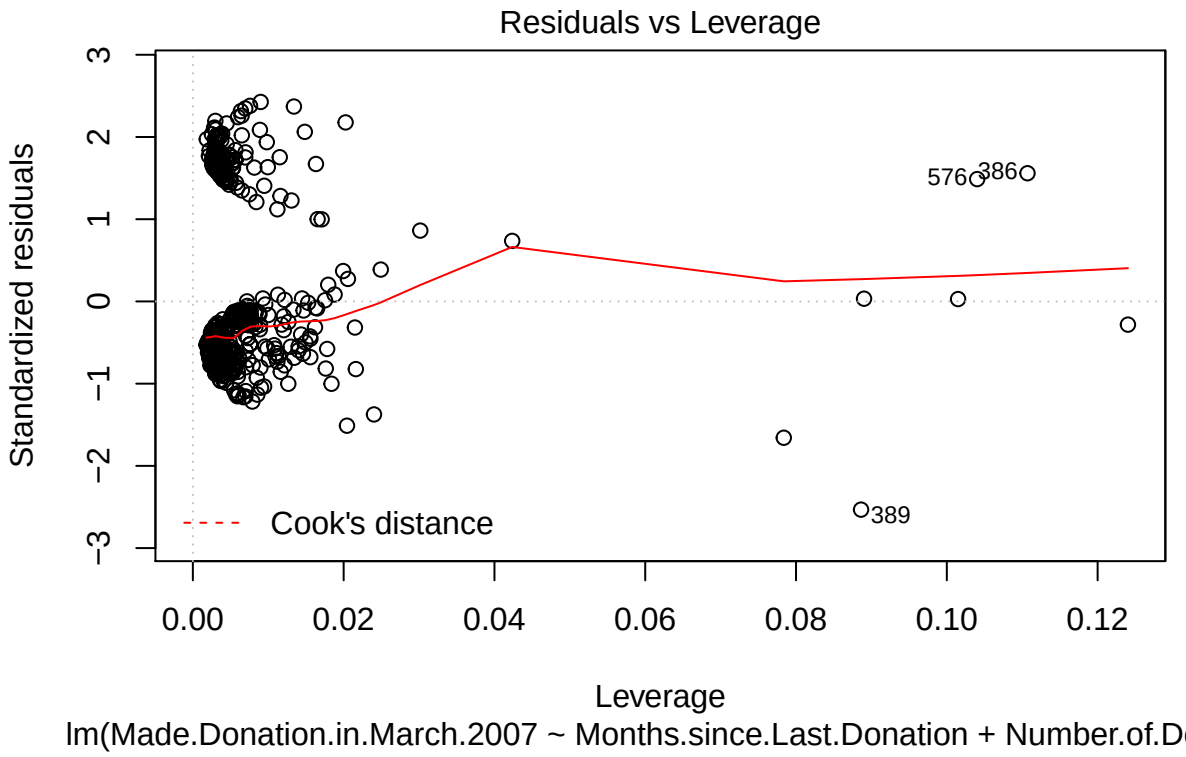
Each variable does contribute sufficiently to a reduction in the sum of the squares of error, and we can't reduce our AIC by eliminating a variable. Let's examine some residual plots:

```
plot(fullmodel)
```


Residuals vs Fitted
Residuals
1.0
0.5
0.0
−0.5
−1.0
−0.5
0.0
0.5
1.0
389
Fitted values
lm(Made.Donation.in.March.2007 ~ Months.since.Last.Donation + Number.of.Don ..


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Made.Donation.in.March.2007 ~ Months.since.Last.Donation + Number.of.Don ..
389
567
369
−3
−2
−1
0
1
2
3


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
−0.5
0.0
0.5
1.0
Fitted values
lm(Made.Donation.in.March.2007 ~ Months.since.Last.Donation + Number.of.Don ..



These residual plots look awful- our error is not normally distributed, and there are high leverage points. We could attempt to transform the data to be more appropriate, but with this distribution of error, our low R^2, and the binary nature of our class variable what we should do instead is just say that this problem is not appropriate for linear modeling.

Just for the purpose of checking our other models, here is the prediction the linear model makes for each point in the test data. We can intepret these as predictions of the likelihood of a row in the test set donating blood.

```
testdata<-read.csv("project test data.csv")
testdata<-testdata[-c(1,4)]
lmpredictions<-predict(fullmodel, testdata)
head(lmpredictions)
```

```
##         1         2         3         4         5         6
## 0.4018055 0.1528907 0.2837510 0.3365531 0.4372504 0.6223474
```