Chapter 4- More on Regression

Shivakumar Panuganti, spanugan@asu.edu

April 25, 2018

Linear Regression Model with Dummy Variables

```
setwd("C:\\Users\\Shivakumar Panuganti\\Documents\\R")
df= read.csv("Salary.csv")
head(df)
```

```
##   Obs Salary Age Gender
## 1   1  1.548 3.2      1
## 2   2  1.629 3.8      1
## 3   3  1.011 2.7      0
## 4   4  1.229 3.4      0
## 5   5  1.746 3.6      1
## 6   6  1.528 4.1      1
```

```
dim(df)
```

```
## [1] 15  4
```

```
salary= df[,2]
age=df[,3]
gender=as.factor(df[,4]) #it converts to dummy
```

```
reg= lm(salary~age+gender)
summary(reg)
```

```
##
## Call:
## lm(formula = salary ~ age + gender)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.136697 -0.067380  0.001351  0.054888  0.154863
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.73206    0.23558   3.107  0.00906 **
## age          0.11122    0.07208   1.543  0.14880
## gender1      0.45868    0.05346   8.580 1.82e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09679 on 12 degrees of freedom
## Multiple R-squared:   0.89,  Adjusted R-squared:  0.8717
## F-statistic: 48.54 on 2 and 12 DF,  p-value: 1.773e-06
```

```
park_data<- read.csv("http://goo.gl/HKnl74")
head(park_data)
```

```
##   weekend num.child  distance rides games wait clean overall
```

```
## 1     yes          0 114.64826    87    73   60    89      47
## 2     yes          2  27.01410    87    78   76    87      65
## 3      no          1  63.30098    85    80   70    88      61
## 4     yes          0  25.90993    88    72   66    89      37
## 5      no          4  54.71831    84    87   74    87      68
## 6      no          5  22.67934    81    79   48    79      27
```

```
park_data$num.child.factor<- factor(park_data$num.child)
park_data$logdistance<- log(park_data$distance)
head(park_data)
```

```
##   weekend num.child  distance rides games wait clean overall
## 1     yes          0 114.64826    87    73   60    89      47
## 2     yes          2  27.01410    87    78   76    87      65
## 3      no          1  63.30098    85    80   70    88      61
## 4     yes          0  25.90993    88    72   66    89      37
## 5      no          4  54.71831    84    87   74    87      68
## 6      no          5  22.67934    81    79   48    79      27
##   num.child.factor logdistance
## 1                0    4.741869
## 2                2    3.296359
## 3                1    4.147901
## 4                0    3.254626
## 5                4    4.002198
## 6                5    3.121454
```

```
data_std<-park_data[,-3]
data_std[,3:7]<-scale(data_std[,3:7]) #Normalization
data_std$has.child<-factor(data_std$num.child>0)
```

Interaction Terms

```
m1<- lm(overall~wait+has.child+wait:has.child, data= data_std)
summary(m1)
```

```
##
## Call:
## lm(formula = overall ~ wait + has.child + wait:has.child, data = data_std)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.16371 -0.44052 -0.07234  0.43560  1.85301
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -0.67778    0.05343 -12.685  < 2e-16 ***
## wait                 0.28882    0.05272   5.479 6.83e-08 ***
## has.childTRUE        0.97747    0.06395  15.286  < 2e-16 ***
## wait:has.childTRUE   0.42678    0.06349   6.722 4.95e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6562 on 496 degrees of freedom
## Multiple R-squared:  0.572,  Adjusted R-squared:  0.5694
```

```
## F-statistic:   221 on 3 and 496 DF,  p-value: < 2.2e-16
```

Let's choose random interaction terms has.child, weekend

```r
m2<- lm(overall~ rides+games+wait+clean+weekend+has.child+rides:has.child+games:has.child+wait:has.chil
summary(m2)
```

```
##
## Call:
## lm(formula = overall ~ rides + games + wait + clean + weekend +
##     has.child + rides:has.child + games:has.child + wait:has.child +
##     clean:has.child + rides:weekend + games:weekend + wait:weekend +
##     clean:weekend, data = data_std)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.12487 -0.31083 -0.00631  0.30854  1.47476
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -0.676657   0.043054 -15.716  < 2e-16 ***
## rides                0.141487   0.067878   2.084  0.03764 *
## games                0.084026   0.049264   1.706  0.08872 .
## wait                 0.126712   0.044226   2.865  0.00435 **
## clean                0.315824   0.079693   3.963 8.51e-05 ***
## weekendyes          -0.025870   0.041057  -0.630  0.52892
## has.childTRUE        0.997867   0.044839  22.254  < 2e-16 ***
## rides:has.childTRUE  0.063469   0.072972   0.870  0.38485
## games:has.childTRUE -0.066827   0.052781  -1.266  0.20607
## wait:has.childTRUE   0.353438   0.047215   7.486 3.38e-13 ***
## clean:has.childTRUE -0.007105   0.079645  -0.089  0.92895
## rides:weekendyes     0.062176   0.067788   0.917  0.35949
## games:weekendyes     0.011651   0.048755   0.239  0.81123
## wait:weekendyes      0.038689   0.044399   0.871  0.38398
## clean:weekendyes    -0.022650   0.070948  -0.319  0.74967
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4524 on 485 degrees of freedom
## Multiple R-squared:  0.8011, Adjusted R-squared:  0.7954
## F-statistic: 139.5 on 14 and 485 DF,  p-value: < 2.2e-16
```

which ha s better AIC and BIC score?

```r
AIC(m1);AIC(m2)
```

```
## [1] 1003.615
## [1] 642.4129
```

```r
BIC(m1);BIC(m2)
```

```
## [1] 1024.688
## [1] 709.8466
```

Regression with variable selection

```r
#Loading discover data
discover_df= read.csv("Discover_step.csv", head= TRUE)
dim(discover_df)
```

```
## [1] 244  15
```

```r
summary(discover_df)
```

```
##      respid          q4             q5a             q5b       
##  Min.   :  14   Min.   :1.000   Min.   :1.000   Min.   :1.000  
##  1st Qu.:1192   1st Qu.:4.000   1st Qu.:4.000   1st Qu.:4.000  
##  Median :2672   Median :4.000   Median :5.000   Median :4.000  
##  Mean   :2737   Mean   :3.988   Mean   :4.258   Mean   :4.283  
##  3rd Qu.:3891   3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:5.000  
##  Max.   :6106   Max.   :5.000   Max.   :5.000   Max.   :5.000  
##       q5c             q5d             q5e             q5f       
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000  
##  1st Qu.:4.000   1st Qu.:3.000   1st Qu.:4.000   1st Qu.:3.000  
##  Median :5.000   Median :4.000   Median :5.000   Median :4.000  
##  Mean   :4.373   Mean   :3.848   Mean   :4.398   Mean   :3.725  
##  3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:5.000  
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000  
##       q5g             q5h             q5i             q5j       
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000  
##  1st Qu.:4.000   1st Qu.:4.000   1st Qu.:4.000   1st Qu.:4.000  
##  Median :4.000   Median :4.000   Median :5.000   Median :5.000  
##  Mean   :4.201   Mean   :4.225   Mean   :4.385   Mean   :4.512  
##  3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:5.000  
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000  
##       q5k             q5l            q5m       
##  Min.   :1.000   Min.   :1.00   Min.   :1.000  
##  1st Qu.:2.000   1st Qu.:3.00   1st Qu.:3.000  
##  Median :4.000   Median :4.00   Median :4.000  
##  Mean   :3.324   Mean   :3.68   Mean   :3.775  
##  3rd Qu.:4.000   3rd Qu.:5.00   3rd Qu.:5.000  
##  Max.   :5.000   Max.   :5.00   Max.   :5.000  
```

```r
target<- discover_df[,2]
features<-as.matrix(discover_df[,3:15])
```

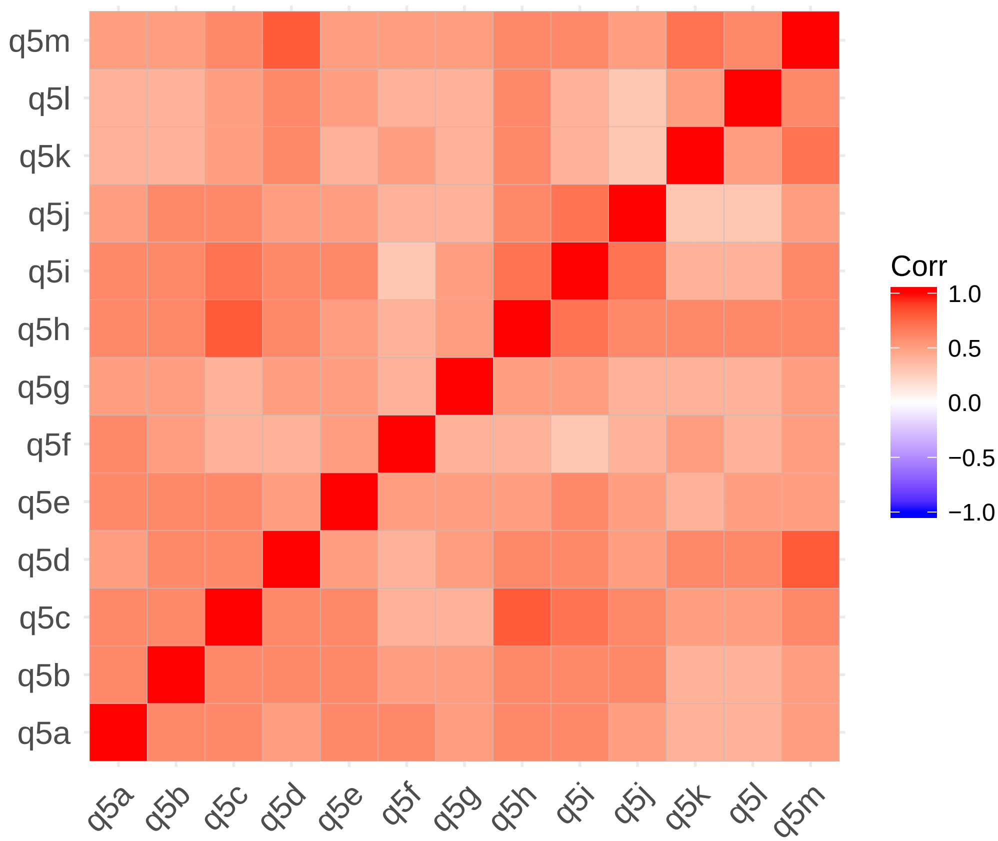
```r
library(Hmisc)
```

```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

## 
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
## 
##     format.pval, units
```

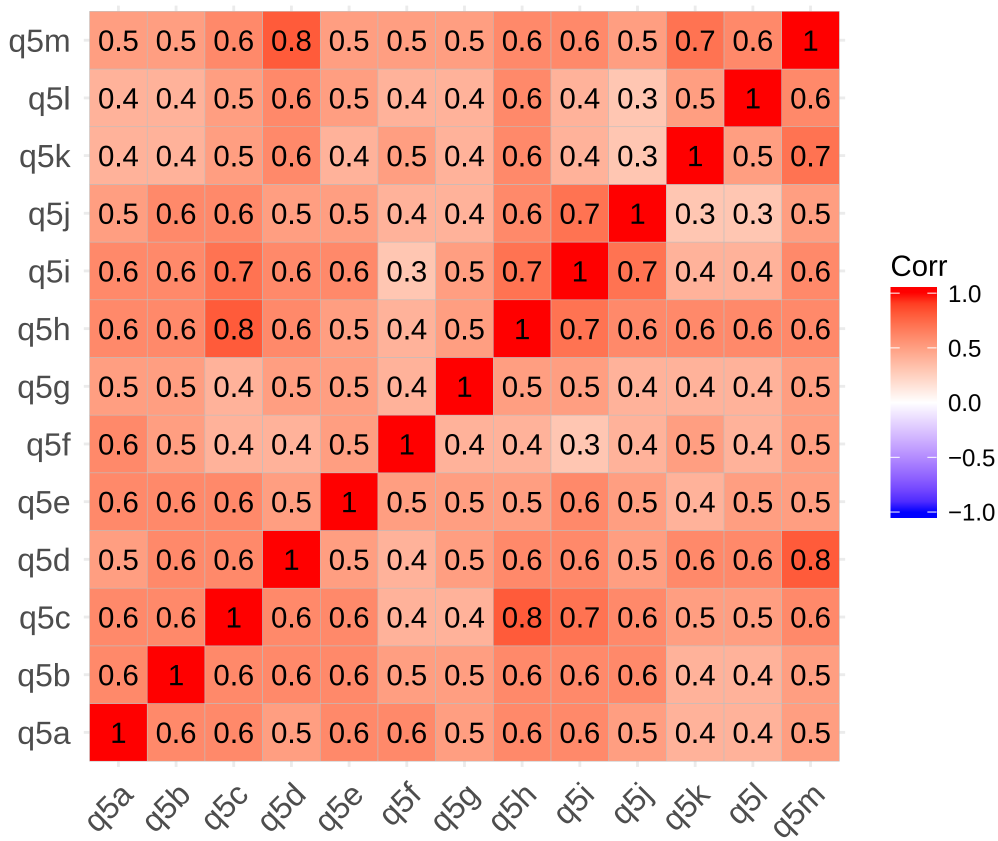
```r
rcorr(features)
```

```
##      q5a  q5b  q5c  q5d  q5e  q5f  q5g  q5h  q5i  q5j  q5k  q5l  q5m
## q5a 1.00 0.62 0.57 0.54 0.59 0.55 0.48 0.58 0.58 0.52 0.41 0.39 0.53
## q5b 0.62 1.00 0.62 0.55 0.60 0.45 0.50 0.63 0.63 0.63 0.45 0.43 0.54
## q5c 0.57 0.62 1.00 0.59 0.59 0.37 0.42 0.75 0.72 0.63 0.49 0.50 0.58
## q5d 0.54 0.55 0.59 1.00 0.54 0.41 0.46 0.63 0.57 0.46 0.65 0.58 0.78
## q5e 0.59 0.60 0.59 0.54 1.00 0.49 0.55 0.54 0.62 0.52 0.40 0.48 0.51
## q5f 0.55 0.45 0.37 0.41 0.49 1.00 0.35 0.38 0.34 0.37 0.47 0.41 0.49
## q5g 0.48 0.50 0.42 0.46 0.55 0.35 1.00 0.47 0.46 0.42 0.36 0.38 0.48
## q5h 0.58 0.63 0.75 0.63 0.54 0.38 0.47 1.00 0.67 0.57 0.61 0.58 0.63
## q5i 0.58 0.63 0.72 0.57 0.62 0.34 0.46 0.67 1.00 0.67 0.43 0.43 0.55
## q5j 0.52 0.63 0.63 0.46 0.52 0.37 0.42 0.57 0.67 1.00 0.34 0.33 0.46
## q5k 0.41 0.45 0.49 0.65 0.40 0.47 0.36 0.61 0.43 0.34 1.00 0.52 0.73
## q5l 0.39 0.43 0.50 0.58 0.48 0.41 0.38 0.58 0.43 0.33 0.52 1.00 0.58
## q5m 0.53 0.54 0.58 0.78 0.51 0.49 0.48 0.63 0.55 0.46 0.73 0.58 1.00
##
## n= 244
##
##
## P
##     q5a q5b q5c q5d q5e q5f q5g q5h q5i q5j q5k q5l q5m
## q5a      0   0   0   0   0   0   0   0   0   0   0   0
## q5b  0       0   0   0   0   0   0   0   0   0   0   0
## q5c  0   0       0   0   0   0   0   0   0   0   0   0
## q5d  0   0   0       0   0   0   0   0   0   0   0   0
## q5e  0   0   0   0       0   0   0   0   0   0   0   0
## q5f  0   0   0   0   0       0   0   0   0   0   0   0
## q5g  0   0   0   0   0   0       0   0   0   0   0   0
## q5h  0   0   0   0   0   0   0       0   0   0   0   0
## q5i  0   0   0   0   0   0   0   0       0   0   0   0
## q5j  0   0   0   0   0   0   0   0   0       0   0   0
## q5k  0   0   0   0   0   0   0   0   0   0       0   0
## q5l  0   0   0   0   0   0   0   0   0   0   0       0
## q5m  0   0   0   0   0   0   0   0   0   0   0   0
```

```r
library(ggcorrplot)
corr<- round(cor(features),1)
ggcorrplot(corr)
```




```
ggcorrplot(corr, lab= TRUE)
```




```r
mul_reg<- lm(target~features)
summary(mul_reg)
```

```
##
## Call:
## lm(formula = target ~ features)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.12319 -0.44940  0.04618  0.66642  1.97852
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.575020   0.396742   6.490  5.2e-10 ***
## featuresq5a -0.065842   0.081508  -0.808  0.42004
## featuresq5b  0.020615   0.096113   0.214  0.83036
## featuresq5c  0.006663   0.124641   0.053  0.95741
## featuresq5d  0.121766   0.085011   1.432  0.15340
## featuresq5e  0.083305   0.090990   0.916  0.36087
## featuresq5f  0.157260   0.058280   2.698  0.00749 **
## featuresq5g -0.137847   0.071595  -1.925  0.05542 .
## featuresq5h  0.196360   0.117125   1.676  0.09500 .
## featuresq5i -0.005235   0.117918  -0.044  0.96463
## featuresq5j -0.115196   0.121708  -0.946  0.34489
## featuresq5k -0.033056   0.072470  -0.456  0.64873
## featuresq5l -0.057548   0.064525  -0.892  0.37340
```

```
## featuresq5m  0.203630   0.090576   2.248  0.02551 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9056 on 230 degrees of freedom
## Multiple R-squared:   0.23,  Adjusted R-squared:  0.1864
## F-statistic: 5.284 on 13 and 230 DF,  p-value: 2.826e-08
```

```r
library(MASS)
step_both<- stepAIC(mul_reg,direction = "both")
```

```
## Start:  AIC=-34.8
## target ~ features
##
##            Df Sum of Sq    RSS     AIC
## <none>                  188.63 -34.801
## - features 13    56.333 244.96   2.961
```

```r
summary(step_both)
```

```
##
## Call:
## lm(formula = target ~ features)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.12319 -0.44940  0.04618  0.66642  1.97852
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.575020   0.396742   6.490  5.2e-10 ***
## featuresq5a -0.065842   0.081508  -0.808  0.42004
## featuresq5b  0.020615   0.096113   0.214  0.83036
## featuresq5c  0.006663   0.124641   0.053  0.95741
## featuresq5d  0.121766   0.085011   1.432  0.15340
## featuresq5e  0.083305   0.090990   0.916  0.36087
## featuresq5f  0.157260   0.058280   2.698  0.00749 **
## featuresq5g -0.137847   0.071595  -1.925  0.05542 .
## featuresq5h  0.196360   0.117125   1.676  0.09500 .
## featuresq5i -0.005235   0.117918  -0.044  0.96463
## featuresq5j -0.115196   0.121708  -0.946  0.34489
## featuresq5k -0.033056   0.072470  -0.456  0.64873
## featuresq5l -0.057548   0.064525  -0.892  0.37340
## featuresq5m  0.203630   0.090576   2.248  0.02551 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9056 on 230 degrees of freedom
## Multiple R-squared:   0.23,  Adjusted R-squared:  0.1864
## F-statistic: 5.284 on 13 and 230 DF,  p-value: 2.826e-08
```

```r
step_back<-stepAIC(mul_reg, direction="backward")
```

```
## Start:  AIC=-34.8
## target ~ features
##
```

```
##              Df Sum of Sq    RSS     AIC
## <none>                    188.63 -34.801
## - features 13    56.333 244.96   2.961
```

```r
summary(step_back)
```

```
##
## Call:
## lm(formula = target ~ features)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.12319 -0.44940  0.04618  0.66642  1.97852
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.575020   0.396742   6.490  5.2e-10 ***
## featuresq5a -0.065842   0.081508  -0.808  0.42004
## featuresq5b  0.020615   0.096113   0.214  0.83036
## featuresq5c  0.006663   0.124641   0.053  0.95741
## featuresq5d  0.121766   0.085011   1.432  0.15340
## featuresq5e  0.083305   0.090990   0.916  0.36087
## featuresq5f  0.157260   0.058280   2.698  0.00749 **
## featuresq5g -0.137847   0.071595  -1.925  0.05542 .
## featuresq5h  0.196360   0.117125   1.676  0.09500 .
## featuresq5i -0.005235   0.117918  -0.044  0.96463
## featuresq5j -0.115196   0.121708  -0.946  0.34489
## featuresq5k -0.033056   0.072470  -0.456  0.64873
## featuresq5l -0.057548   0.064525  -0.892  0.37340
## featuresq5m  0.203630   0.090576   2.248  0.02551 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9056 on 230 degrees of freedom
## Multiple R-squared:   0.23,  Adjusted R-squared:  0.1864
## F-statistic: 5.284 on 13 and 230 DF,  p-value: 2.826e-08
```

Ridge Regression

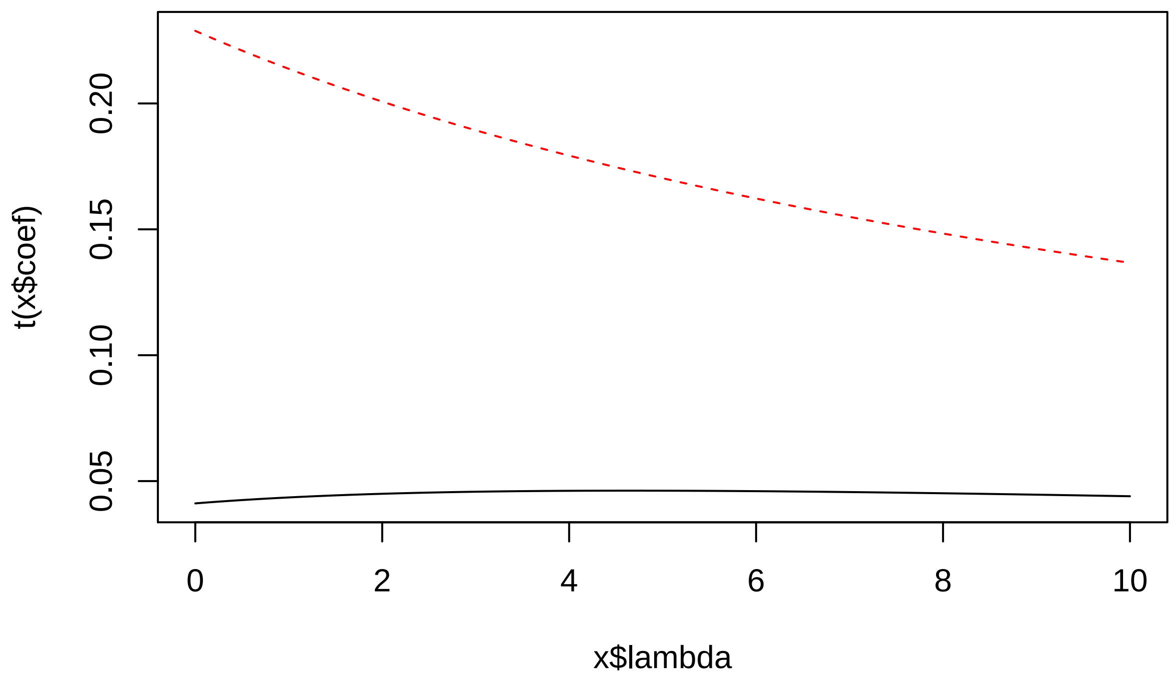
```r
## Using the Salary Data
lm.ridge(salary~age+gender, df, lambda=10)
```

```
##                   age   gender1
## 0.8047131 0.1188942 0.2740182
```

```r
lm(salary~age+gender)
```

```
##
## Call:
## lm(formula = salary ~ age + gender)
##
## Coefficients:
## (Intercept)          age      gender1
##      0.7321       0.1112       0.4587
```

```r
plot(lm.ridge(salary~age+gender, df, lambda=seq(0,10,0.001)))
```



Using Discover Data

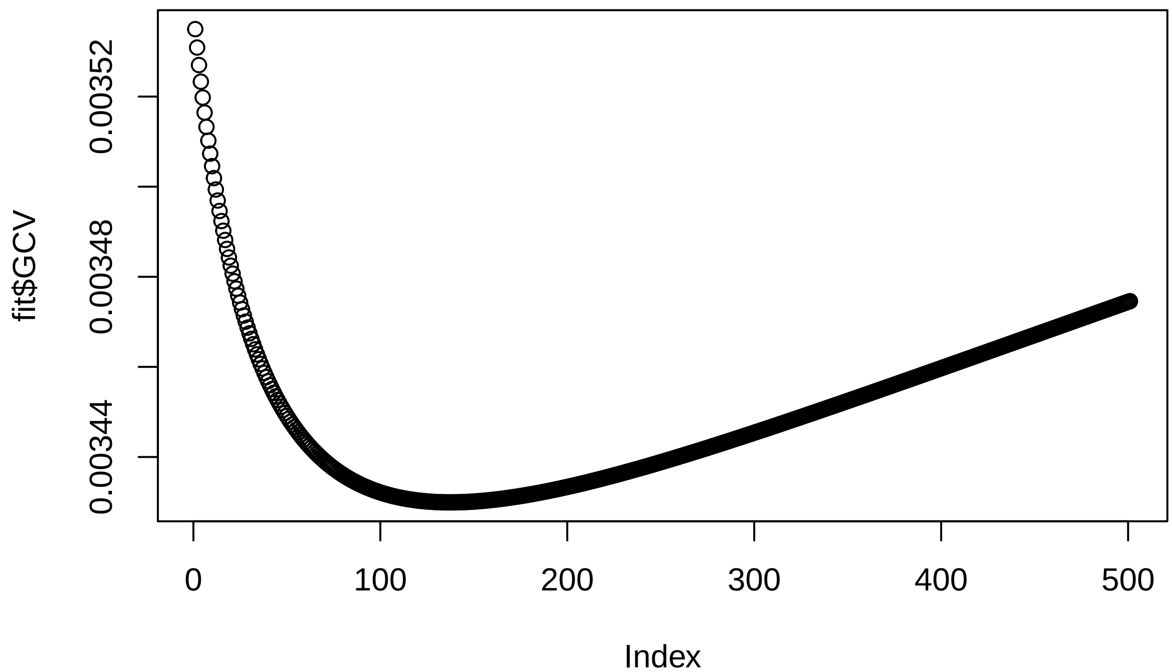
```r
lm(q4~., data= discover_df[,-1])
```

```
##
## Call:
## lm(formula = q4 ~ ., data = discover_df[, -1])
##
## Coefficients:
## (Intercept)          q5a          q5b          q5c          q5d
##    2.575020    -0.065842     0.020615     0.006663     0.121766
##         q5e          q5f          q5g          q5h          q5i
##    0.083305     0.157260    -0.137847     0.196360    -0.005235
##         q5j          q5k          q5l          q5m
##   -0.115196    -0.033056    -0.057548     0.203630
```

```r
lm.ridge(q4~., data=discover_df[,-1],lambda = 1)
```

```
##                      q5a          q5b          q5c          q5d
##  2.571516478 -0.064072124  0.020513942  0.007716169  0.121134125
##          q5e          q5f          q5g          q5h          q5i
##  0.082365796  0.155848368 -0.136478048  0.192864355 -0.004909299
##          q5j          q5k          q5l          q5m
## -0.113480282 -0.030756740 -0.056066262  0.201079145
```

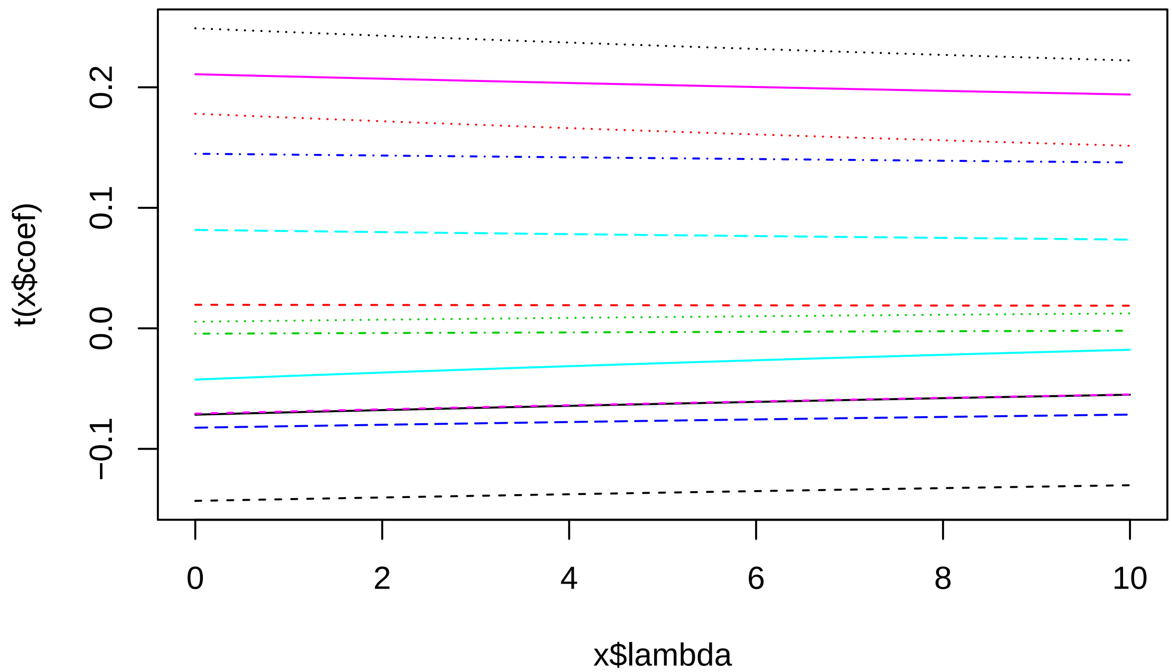
```
fit<- lm.ridge(q4~., data=discover_df[,-1], lambda = seq(0,500,by=1))
plot(fit$GCV) #Generalized Cross Validation
```



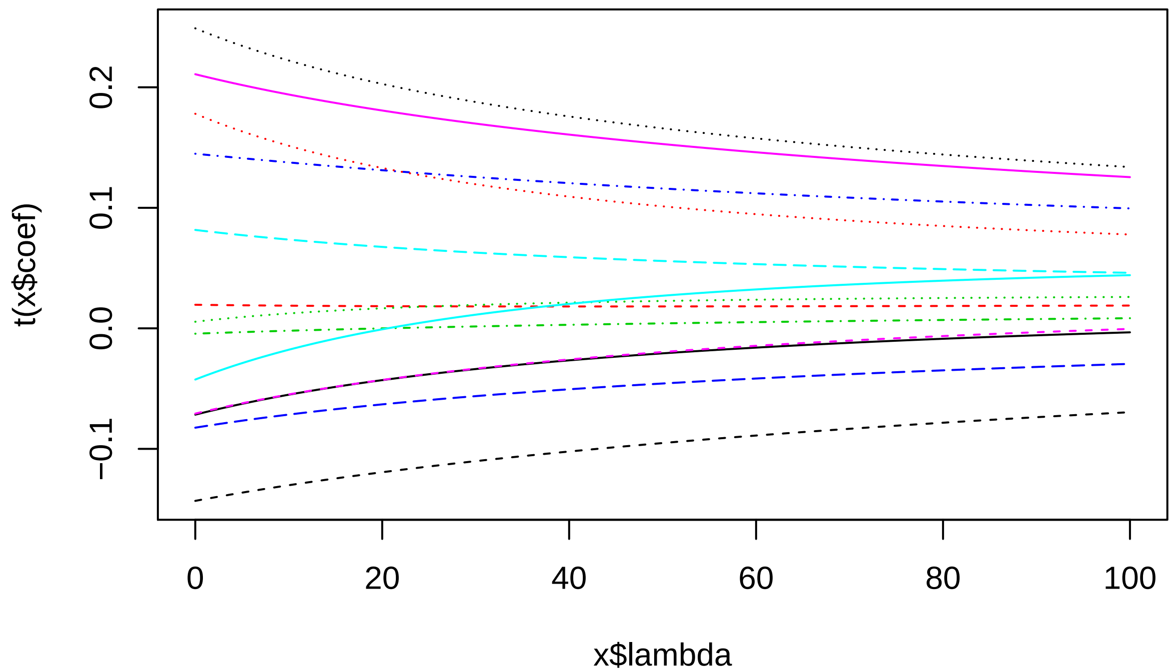

```
lm.ridge(q4~.,data=discover_df[,-1],lambda = 120)
```

```
##                        q5a          q5b          q5c          q5d
##  2.4285585271 0.0006671241 0.0202976928 0.0323742498 0.0795932120
##           q5e          q5f          q5g          q5h          q5i
##  0.0445694799 0.0878443011 -0.0599531029 0.0799384505 0.0113377502
##           q5j          q5k          q5l          q5m
## -0.0351152000 0.0365658750 0.0033067592 0.1026211208
```

```
#Weak shrinkage
plot(lm.ridge(q4~., data=discover_df[,-1],lambda = seq(0,10,0.1)))
```




```r
#Strong shrinkage
plot(lm.ridge(q4~., data=discover_df[,-1],lambda = seq(0,100,0.1)))
```

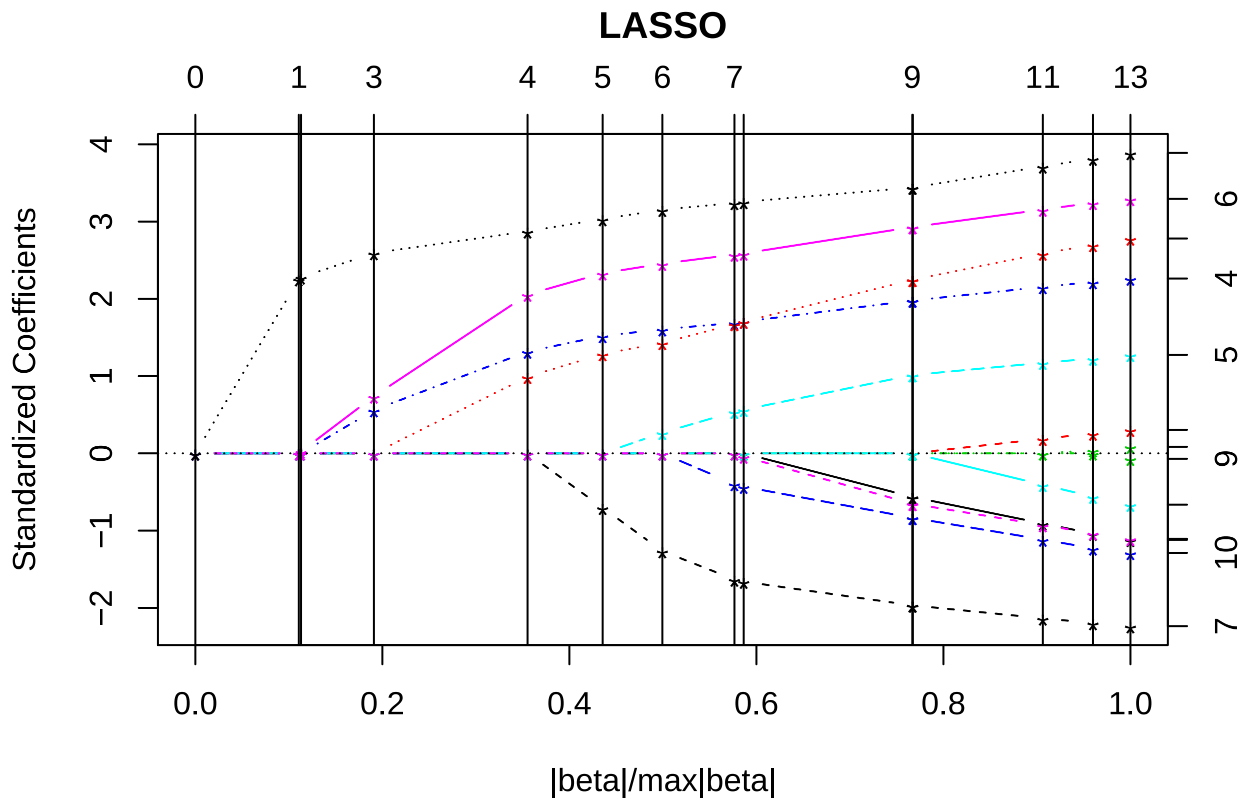


Lasso Regression

```
library(lars)
```

```
## Loaded lars 1.2
```

```
las_reg= lars(features, target, type= "lasso")
plot(las_reg)
```



Coefficients are shrinking into zero values depending on alpha parameter in the lasso formula

```
dim(las_reg$beta)
```

```
## [1] 14 13
```

```
las_reg$lambda
```

```
##  [1] 6.45857932 4.20419242 4.16837315 3.06923469 0.92606749 0.62694064
##  [7] 0.42151054 0.27441411 0.26326984 0.11423874 0.11375943 0.04392235
## [13] 0.01750177
```

```
las_reg$beta[1,]
```

```
## q5a q5b q5c q5d q5e q5f q5g q5h q5i q5j q5k q5l q5m
##   0   0   0   0   0   0   0   0   0   0   0   0   0
```

```
las_reg$lambda[1]
```

```
## [1] 6.458579
```

```
las_reg$beta[6,] #Five significant Variables
```

```
##         q5a         q5b         q5c         q5d         q5e         q5f
##  0.00000000  0.00000000  0.00000000  0.08177843  0.00000000  0.11133033
##         q5g         q5h         q5i         q5j         q5k         q5l
## -0.04340905  0.09080984  0.00000000  0.00000000  0.00000000  0.00000000
##         q5m
##  0.15881836
```