Linear Regression

Ruizhe Li

4/18/2020

```
library(ISLR)
library(MASS)
library(tree)
library(cluster)
library(mlbench)
library(dplyr)
```

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

## The following object is masked from 'package:MASS':
##
##     select

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

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

```
library(ggplot2)
library(maps)
```

```
##
## Attaching package: 'maps'

## The following object is masked from 'package:cluster':
##
##     votes.repub
```

```
library(highcharter)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Highcharts (www.highcharts.com) is a Highsoft software product which is

## not free for commercial and Governmental use
```

```
library(caTools)
```

```
fifa <- read.csv("fifa_cleaned.csv")
df <- read.csv("data.csv")
colnames(fifa)
```

```
##  [1] "ID"                       "Name"
##  [3] "Age"                      "Overall"
##  [5] "Potential"                "Club"
##  [7] "Value"                    "Wage"
##  [9] "Special"                  "Preferred.Foot"
## [11] "International.Reputation" "Weak.Foot"
## [13] "Skill.Moves"              "Work.Rate"
## [15] "Body.Type"                "Position"
## [17] "Height"                   "Weight"
## [19] "Crossing"                 "Finishing"
## [21] "HeadingAccuracy"          "ShortPassing"
## [23] "Volleys"                  "Dribbling"
## [25] "Curve"                    "FKAccuracy"
## [27] "LongPassing"              "BallControl"
## [29] "Acceleration"             "SprintSpeed"
## [31] "Agility"                  "Reactions"
## [33] "Balance"                  "ShotPower"
## [35] "Jumping"                  "Stamina"
## [37] "Strength"                 "LongShots"
## [39] "Aggression"               "Interceptions"
## [41] "Positioning"              "Vision"
## [43] "Penalties"                "Composure"
## [45] "Marking"                  "StandingTackle"
## [47] "SlidingTackle"            "GKDiving"
## [49] "GKHandling"               "GKKicking"
## [51] "GKPositioning"            "GKReflexes"
## [53] "Release.Clause"
```

```r
fifa.gk <- fifa[fifa[,"Position"]=="GK",]
fifa.gk<-select(fifa.gk, select=-c("ID","Name","Age","Club","Preferred.Foot","Work.Rate","Body.Type","P
train <- sample(1:dim(fifa.gk)[1],dim(fifa.gk)[1]*0.7,rep=FALSE)
test <- -train
train = fifa.gk[train,]
test = fifa.gk[test,]

fit2 = lm(Overall~GKDiving+GKHandling+GKPositioning+GKKicking+GKReflexes+Release.Clause,data=train, na.a

summary(fit2)
```

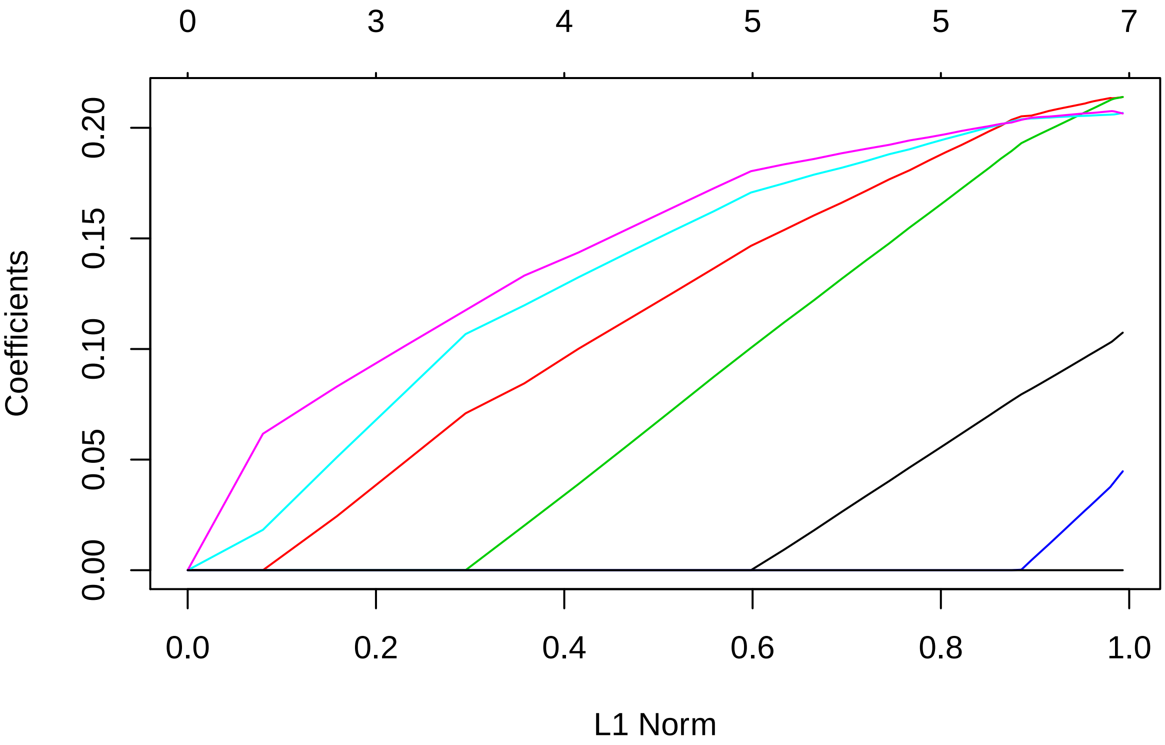
```
##
## Call:
## lm(formula = Overall ~ GKDiving + GKHandling + GKPositioning +
##     GKKicking + GKReflexes + Release.Clause, data = train, na.action = na.omit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.44253 -0.43327  0.04447  0.48568  2.25353
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1.718e-01  2.099e-01  -0.818 0.413232
## GKDiving          2.318e-01  6.276e-03  36.936  < 2e-16 ***
## GKHandling        2.273e-01  5.596e-03  40.617  < 2e-16 ***
## GKPositioning     2.420e-01  5.042e-03  48.008  < 2e-16 ***
## GKKicking         5.353e-02  3.887e-03  13.772  < 2e-16 ***
## GKReflexes      2.535e-01  5.903e-03  42.950  < 2e-16 ***
## Release.Clause  1.071e-08  2.951e-09   3.628 0.000296 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.724 on 1410 degrees of freedom
## Multiple R-squared:  0.9905, Adjusted R-squared:  0.9904
## F-statistic: 2.441e+04 on 6 and 1410 DF,  p-value: < 2.2e-16
```

```
library(glmnet)
```

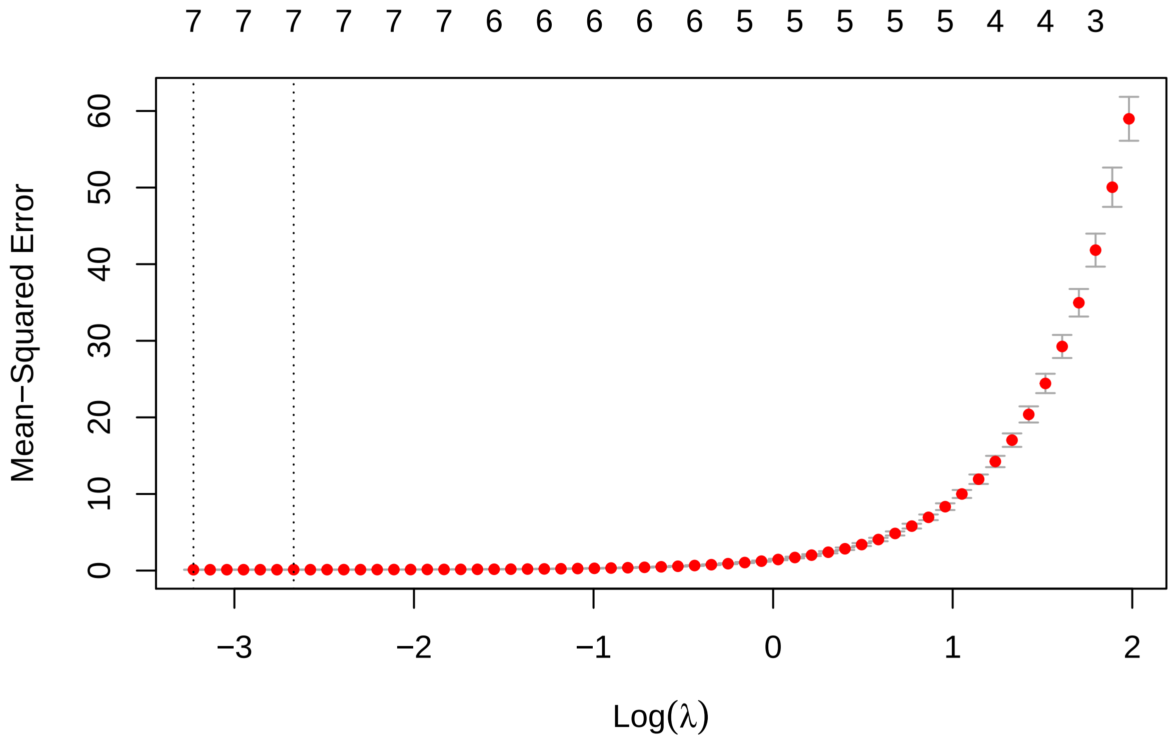
```
## Loading required package: Matrix
```

```
## Loaded glmnet 3.0-2
```

```
set.seed(1)
train=sample(c(TRUE,FALSE),nrow(fifa.gk),rep=TRUE)
test=(!train)
x=model.matrix(Overall~.-1, fifa.gk)
y=fifa.gk$Overall
lasso.mod=glmnet(x[train,],y[train],alpha=1)
plot(lasso.mod)
```



```
cv.out=cv.glmnet(x[train,],y[train],alpha=1)
plot(cv.out)
```



```
bestlam=cv.out$lambda.min
lasso.pred=predict(lasso.mod,s=bestlam,newx=x[test,])
out=glmnet(x,y,alpha=1)
lasso.coef=predict(out,type="coefficient",s=bestlam)
lasso.coef
```

```
## 42 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept)     1.517983e+00
## Potential       .
## Special         .
## Weak.Foot       .
## Skill.Moves     .
## Height          .
## Weight          .
## Crossing        .
## Finishing       .
## HeadingAccuracy .
## ShortPassing    .
## Volleys         .
## Dribbling       .
## Curve           .
## FKAccuracy      .
## LongPassing     .
## BallControl     .
## Acceleration    .
## SprintSpeed     .
## Agility         .
## Reactions       1.074205e-01
## Balance         .
## ShotPower       .
## Jumping          .
## Stamina          .
## Strength         .
## LongShots        .
## Aggression       .
## Interceptions    .
## Positioning      .
## Vision           .
## Penalties        .
## Composure        .
## Marking          .
## StandingTackle   .
## SlidingTackle    .
## GKDiving         2.102739e-01
## GKHandling       2.120253e-01
## GKKicking        4.630482e-02
## GKPositioning    2.068897e-01
## GKReflexes       2.084530e-01
## Release.Clause   7.232934e-09
```

```
rownames(lasso.coef)[which(lasso.coef!=0)]
```

```
## [1] "(Intercept)"    "Reactions"      "GKDiving"       "GKHandling"
## [5] "GKKicking"      "GKPositioning"  "GKReflexes"     "Release.Clause"
```

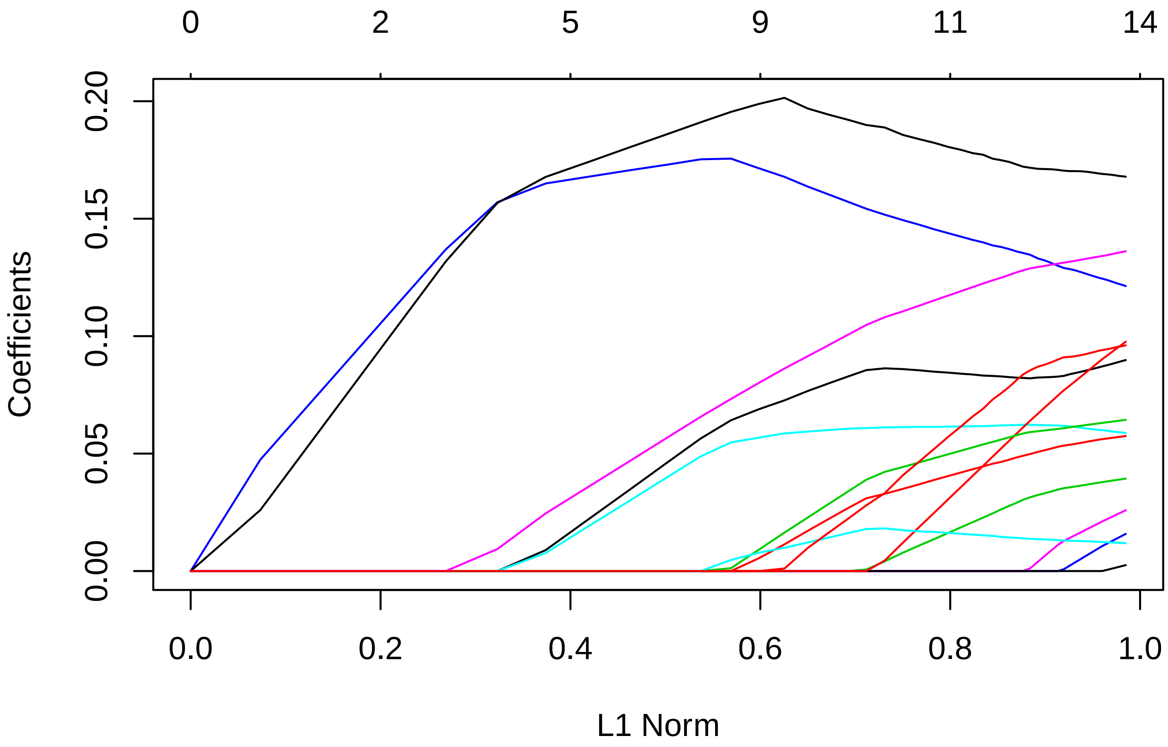
```
fit2 <- lm(Overall~GKDiving+GKHandling+GKPositioning+GKKicking+GKReflexes+Release.Clause,data=fifa.gk)

summary(fit2)
```

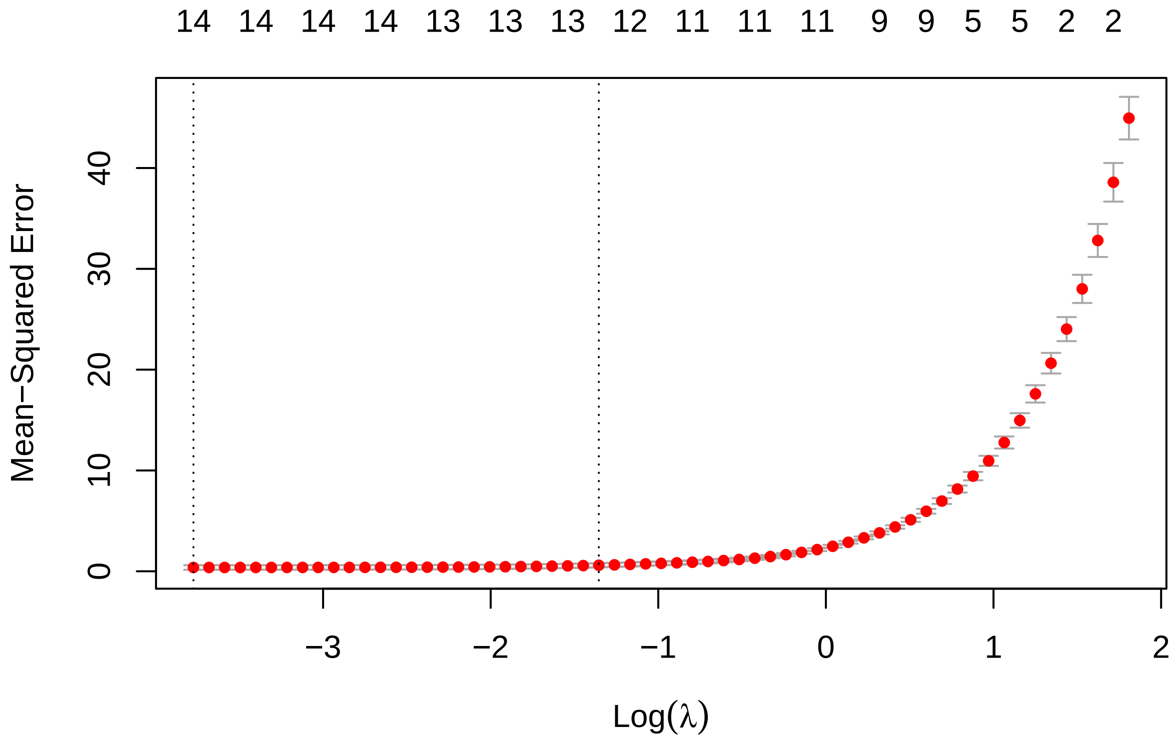
```
##
## Call:
## lm(formula = Overall ~ GKDiving + GKHandling + GKPositioning +
##     GKKicking + GKReflexes + Release.Clause, data = fifa.gk)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.68416 -0.42502  0.03961  0.47734  2.63224
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.210e-02  1.744e-01  -0.069    0.945
## GKDiving        2.322e-01  5.279e-03  43.980  < 2e-16 ***
## GKHandling      2.267e-01  4.691e-03  48.316  < 2e-16 ***
## GKPositioning   2.421e-01  4.216e-03  57.407  < 2e-16 ***
## GKKicking       5.353e-02  3.249e-03  16.475  < 2e-16 ***
## GKReflexes      2.514e-01  4.984e-03  50.433  < 2e-16 ***
## Release.Clause  1.129e-08  2.169e-09   5.205 2.14e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.728 on 2018 degrees of freedom
## Multiple R-squared:  0.9908, Adjusted R-squared:  0.9908
## F-statistic: 3.639e+04 on 6 and 2018 DF,  p-value: < 2.2e-16
```

```
fifa.back <- fifa[fifa[,"Position"]=="CB",]
fifa.back<-select(fifa.back, select=-c("ID","Name","Age","Club","Preferred.Foot","Work.Rate","Body.Type
# train <- sample(1:dim(fifa.gk)[1],dim(fifa.gk)[1]*0.7,rep=FALSE)
# test <- -train
# train = fifa.gk[train,]
# test = fifa.gk[test,]
#
# fit2 = lm(Overall~.,data=train, na.action=na.omit)
#
# summary(fit2)
```

```
library(glmnet)
set.seed(1)
train=sample(c(TRUE,FALSE),nrow(fifa.back),rep=TRUE)
test=(!train)
x=model.matrix(Overall~.-1, fifa.back)
y=fifa.back$Overall
lasso.mod=glmnet(x[train,],y[train],alpha=1)
plot(lasso.mod)
```



```
cv.out=cv.glmnet(x[train,],y[train],alpha=1)
plot(cv.out)
```



```
bestlam=cv.out$lambda.min
lasso.pred=predict(lasso.mod,s=bestlam,newx=x[test,])
out=glmnet(x,y,alpha=1)
lasso.coef=predict(out,type="coefficient",s=bestlam)
#lasso.coef
rownames(lasso.coef)[which(lasso.coef!=0)]
```

```
##  [1] "(Intercept)"      "HeadingAccuracy" "ShortPassing"    "BallControl"
##  [5] "SprintSpeed"      "Reactions"       "Jumping"         "Stamina"
##  [9] "Strength"         "Aggression"      "Interceptions"   "Composure"
## [13] "Marking"          "StandingTackle"  "SlidingTackle"
```

```
fit2 <- lm(Overall~HeadingAccuracy+ShortPassing+BallControl+SprintSpeed+Reactions+Jumping+Strength+Inte

summary(fit2)
```

```
##
## Call:
## lm(formula = Overall ~ HeadingAccuracy + ShortPassing + BallControl +
##     SprintSpeed + Reactions + Jumping + Strength + Interceptions +
##     Marking + StandingTackle + SlidingTackle + Aggression + Reactions +
##     Stamina + Composure, data = fifa.back)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7931 -0.2706 -0.0275  0.2530 13.1219
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.443084   0.160595   8.986  < 2e-16 ***
## HeadingAccuracy 0.095377   0.002360  40.414  < 2e-16 ***
## ShortPassing    0.051953   0.001971  26.355  < 2e-16 ***
## BallControl     0.041912   0.001991  21.046  < 2e-16 ***
## SprintSpeed     0.017389   0.001202  14.472  < 2e-16 ***
## Reactions       0.054771   0.002641  20.739  < 2e-16 ***
## Jumping         0.028997   0.001176  24.667  < 2e-16 ***
## Strength        0.097810   0.001727  56.640  < 2e-16 ***
## Interceptions   0.126186   0.003111  40.558  < 2e-16 ***
## Marking         0.138696   0.002515  55.158  < 2e-16 ***
## StandingTackle  0.172272   0.004483  38.427  < 2e-16 ***
## SlidingTackle   0.096675   0.003915  24.691  < 2e-16 ***
## Aggression      0.065015   0.001562  41.635  < 2e-16 ***
## Stamina         0.001293   0.001347   0.960  0.33739
## Composure       0.006468   0.002010   3.219  0.00131 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4879 on 1763 degrees of freedom
## Multiple R-squared:  0.9945, Adjusted R-squared:  0.9945
## F-statistic: 2.284e+04 on 14 and 1763 DF,  p-value: < 2.2e-16
```