18) Ridge Regression, and Least Absolute Shrinkage and Selection Operator (LASSO)

Vitor Kamada

February 2018

Reference

Tables, Graphics, and Figures from

An Introduction to Statistical Learning

James et al. (2017): Chapters: 6.2, and 6.6

Ridge Regression

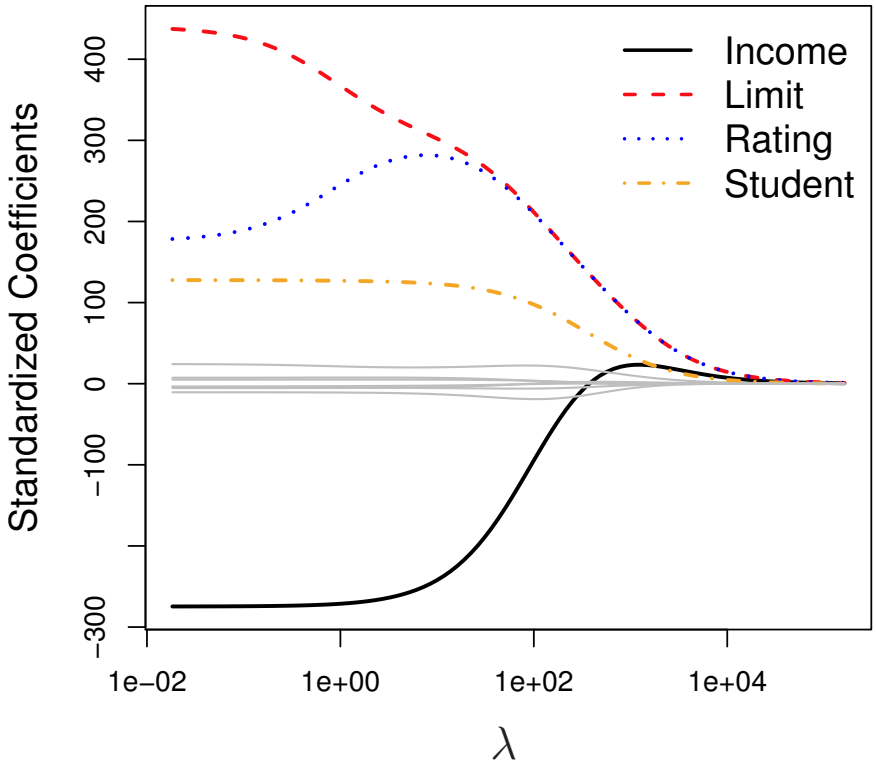
$$\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

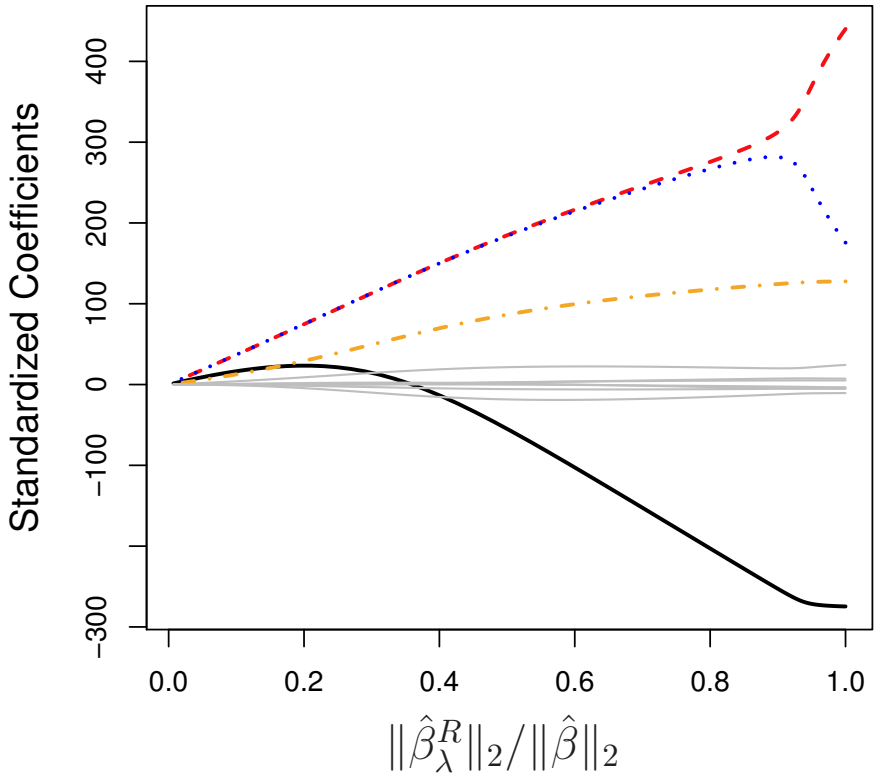
$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}}$$

$$\frac{|| \hat{\beta}_\lambda^R ||_2}{|| \hat{\beta} ||_2}$$

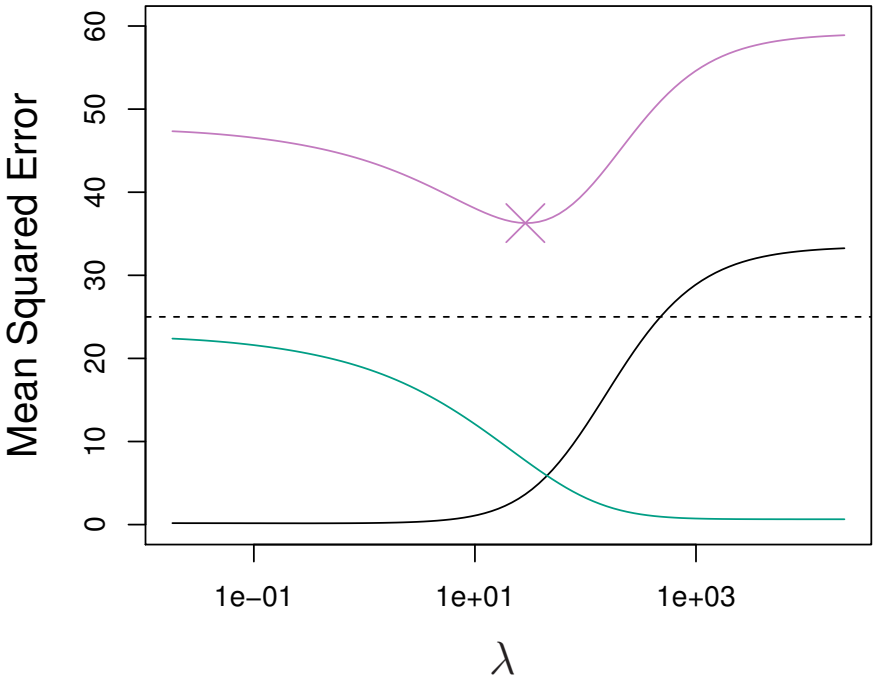
$$|| \beta ||_2 = \sqrt{\sum_{j=1}^{p} \beta_j^2}$$

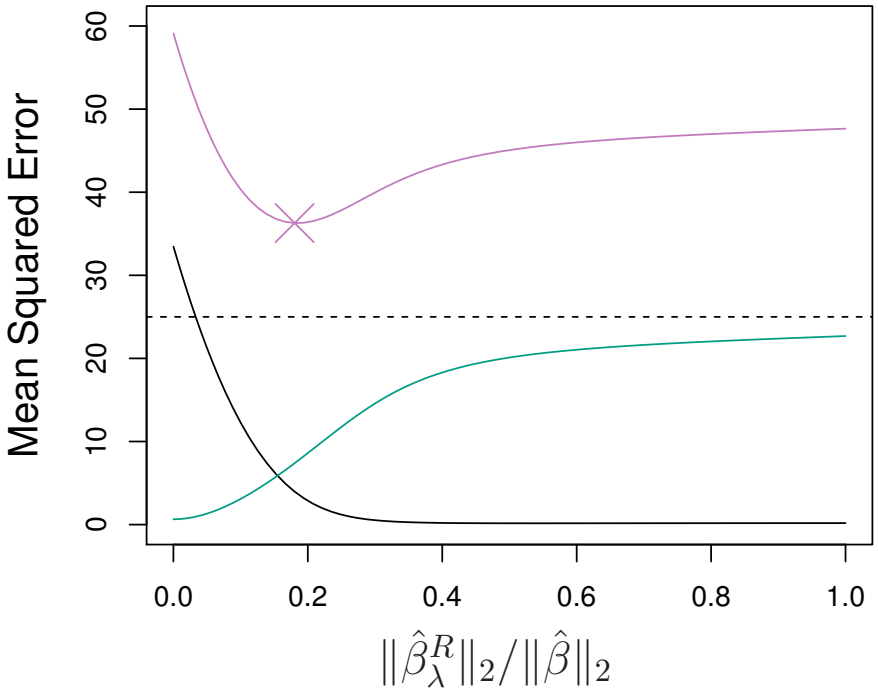
Credit Data Set





Ridge: Squared Bias (Black), Variance (Green), and Test Mean Squared Error (Purple)





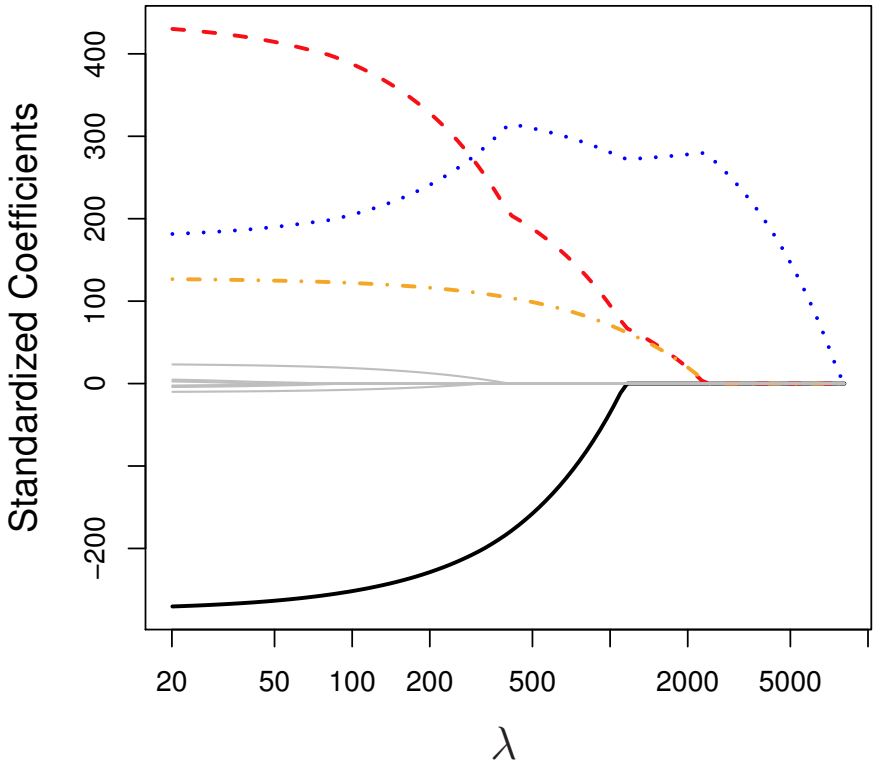
Least Absolute Shrinkage and Selection Operator (LASSO)

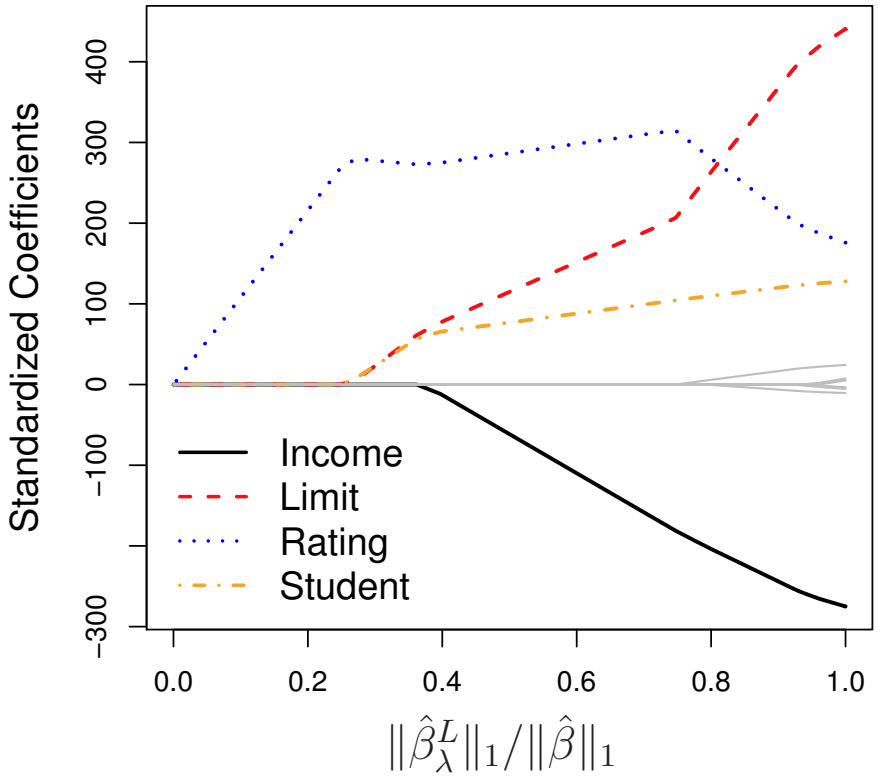
$$\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

$$\frac{|| \hat{\beta}_\lambda^L ||_1}{|| \hat{\beta} ||_1}$$

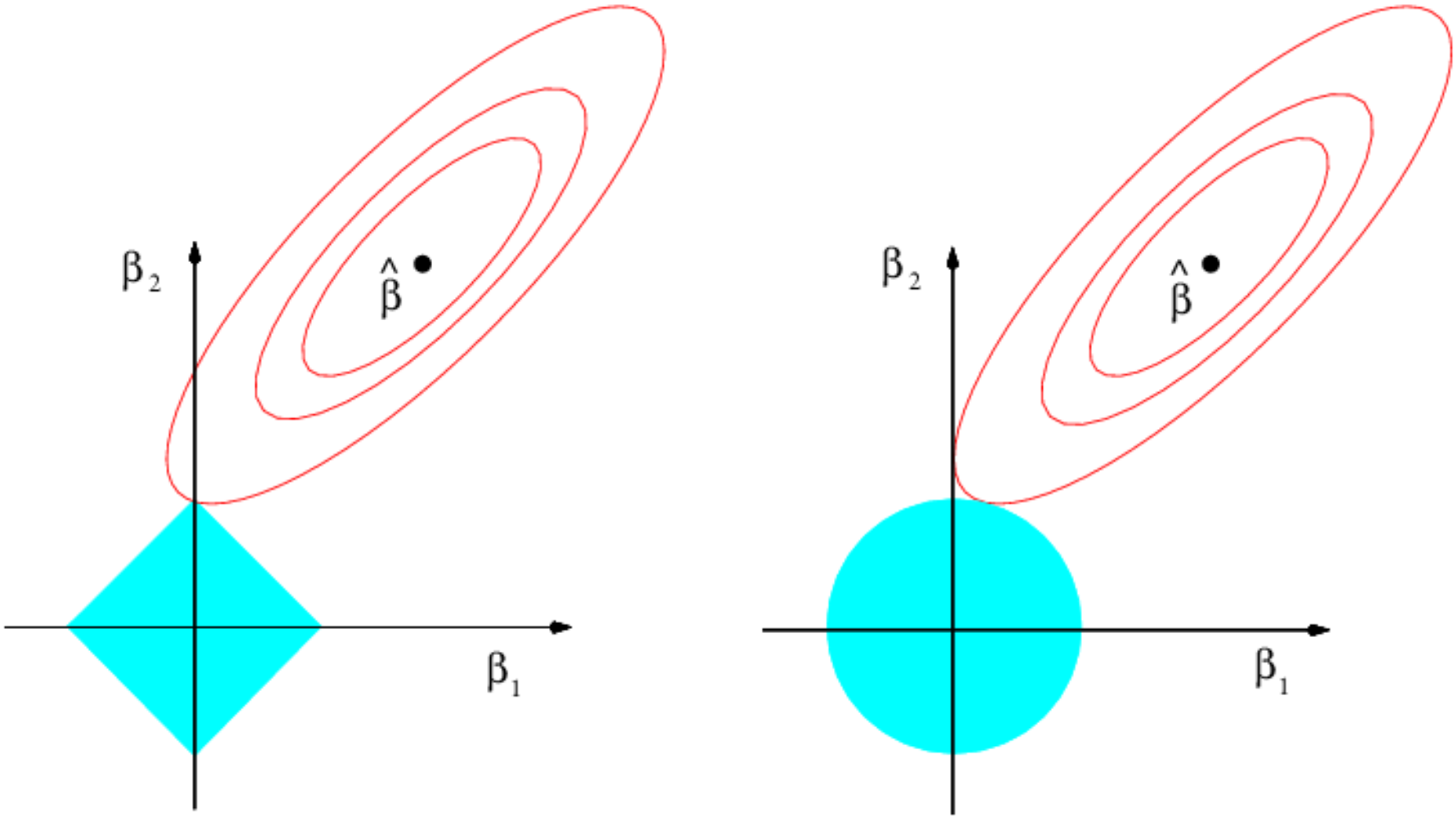
$$|| \beta ||_1 = \Sigma |\beta_j|$$

The Standardized Lasso Coefficients

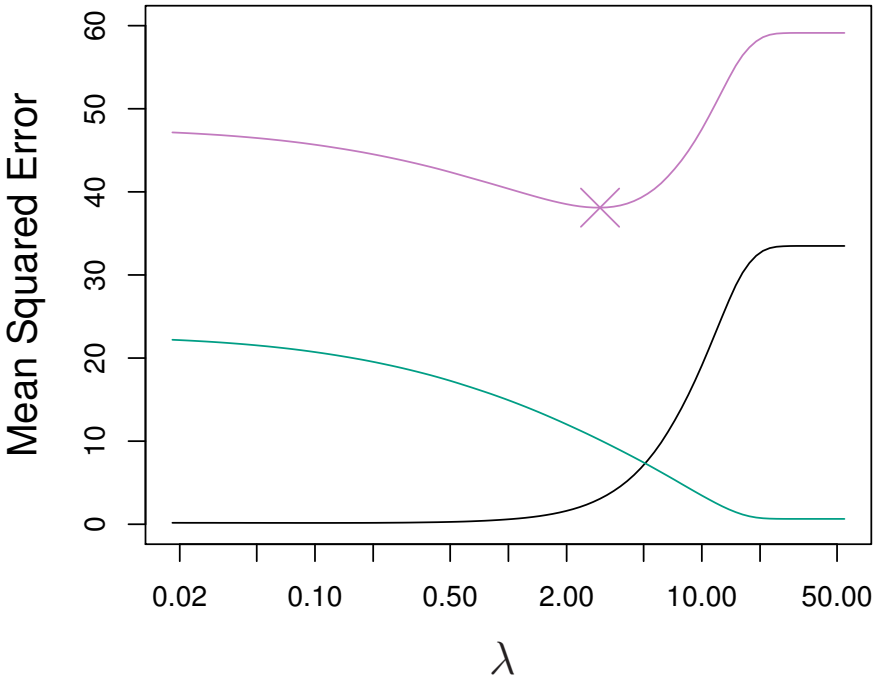


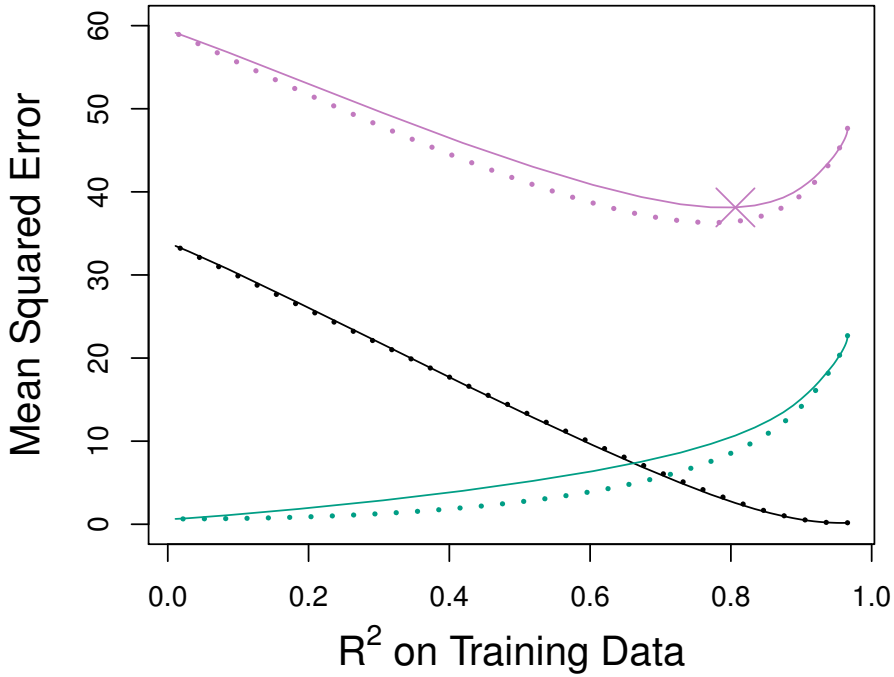


$|\beta_1| + |\beta_2| \leq s$ and $\beta_1^2 + \beta_2^2 \leq s$

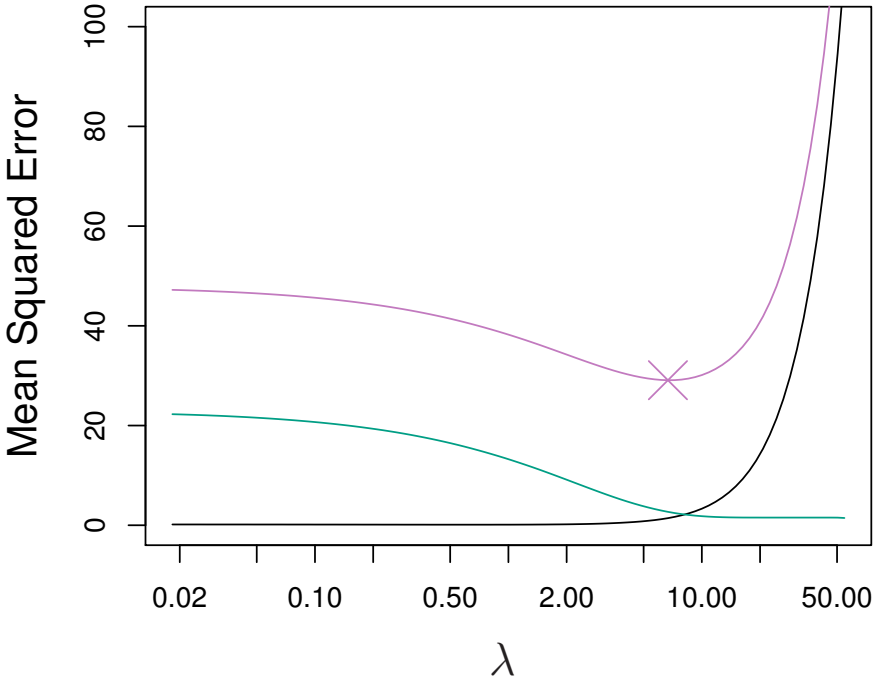


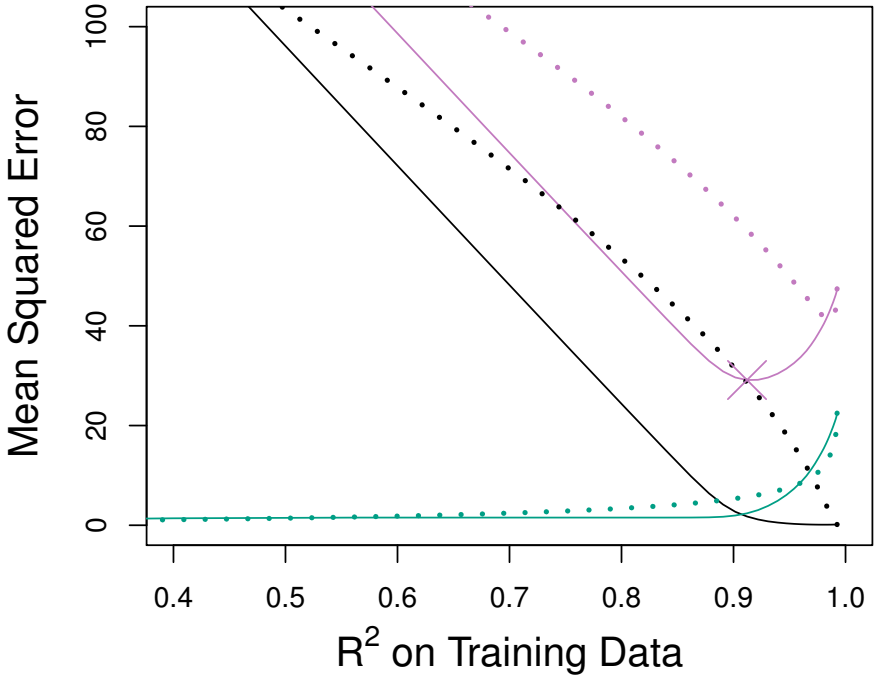
Lasso (Solid) vs Ridge (Dotted)



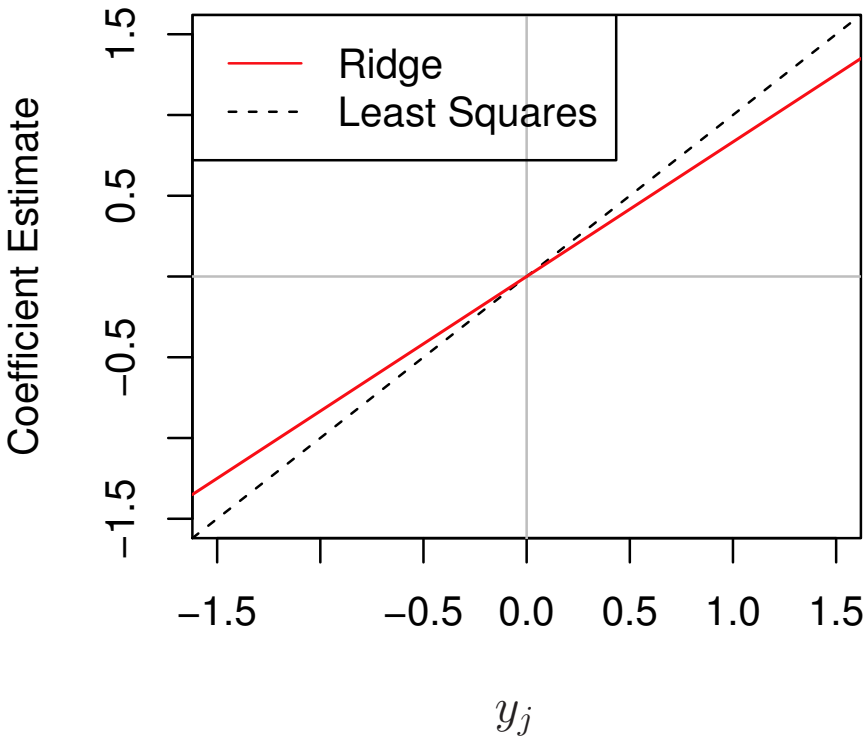


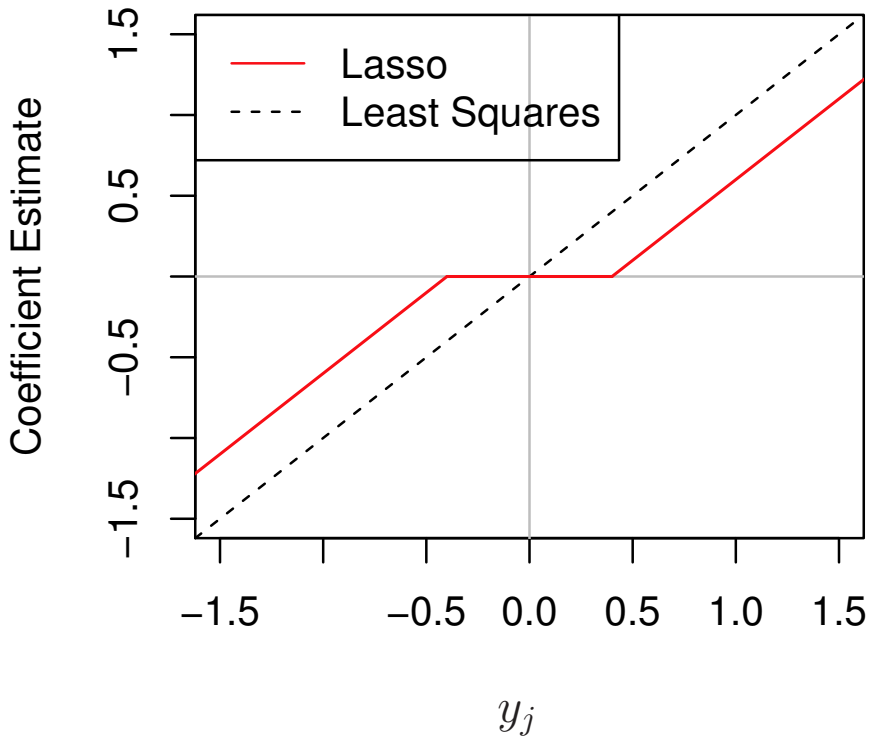
Only 2 Xs are related to the Y





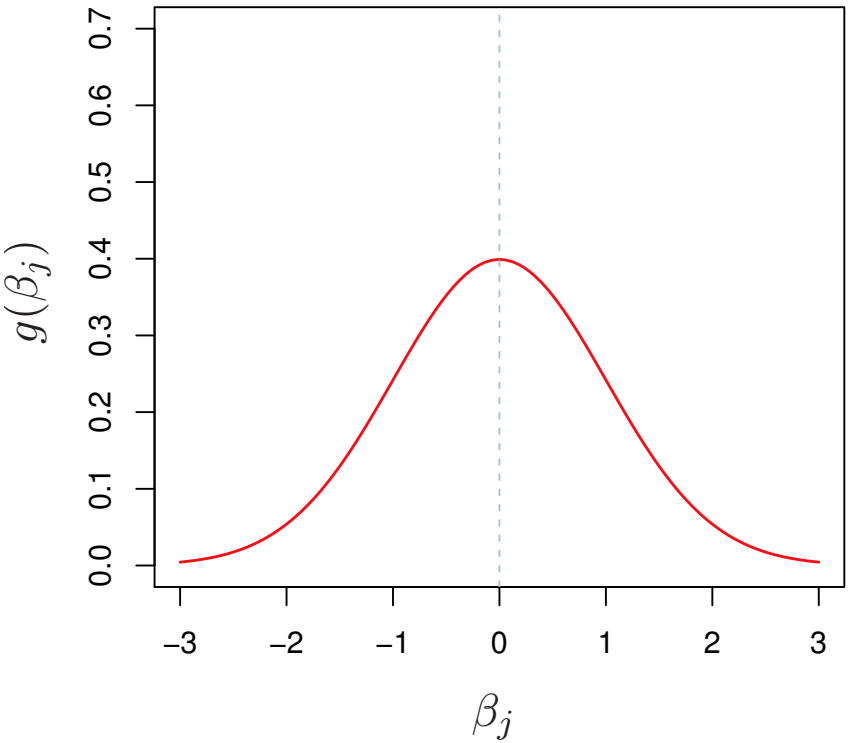
$n = p$ and X a Diagonal Matrix with 1's

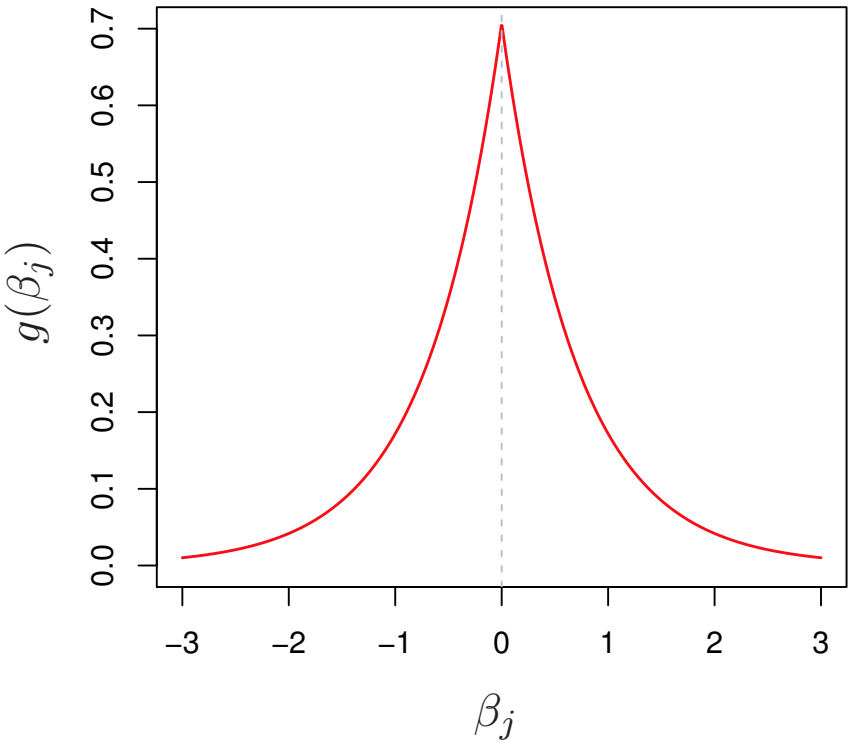




Gaussian Prior vs Double-Exponential Prior

$$p(\beta|X,Y) \propto f(Y|X,\beta)p(\beta|X)$$





library(glmnet)

```
x=model.matrix(Salary~.,Hitters)[,-1]
y=Hitters$Salary
grid=10^seq(10,-2,length=100)
ridge.mod=glmnet(x,y,alpha=0,lambda=grid)
dim(coef(ridge.mod))
```

20 rows (one for each $Xs + \beta_0$)

100 columns (λ)

ridge.mod$lambda[50]

$$\lambda = 11497.57$$

coef(ridge.mod)[,50]

(Intercept)	AtBat	Hits	HmRun
407.356050200	0.036957182	0.138180344	0.524629976
Runs	**RBI**	**Walks**	**Years**
0.230701523	0.239841459	0.289618741	1.107702929
CAtBat	**CHits**	**CHmRun**	**CRuns**
0.003131815	0.011653637	0.087545670	0.023379882
CRBI	**CWalks**	**LeagueN**	**DivisionW**
0.024138320	0.025015421	0.085028114	-6.215440973
PutOuts	**Assists**	**Errors**	**NewLeagueN**
0.016482577	0.002612988	-0.020502690	0.301433531

ridge.mod$lambda[60]

λ =705.48

coef(ridge.mod)[,60]

(Intercept)	AtBat	Hits	HmRun
54.32519950	0.11211115	0.65622409	1.17980910
Runs	RBI	Walks	Years
0.93769713	0.84718546	1.31987948	2.59640425
CAtBat	CHits	CHmRun	CRuns
0.01083413	0.04674557	0.33777318	0.09355528
CRBI	CWalks	LeagueN	DivisionW
0.09780402	0.07189612	13.68370191	-54.65877750
PutOuts	Assists	Errors	NewLeagueN
0.11852289	0.01606037	-0.70358655	8.61181213

predict(ridge.mod, s=160, type="coefficients")[1:20,]

$$\lambda = 160$$

(Intercept)	AtBat	Hits	HmRun
11.717214276	-0.014121915	1.137139060	-0.186846563
Runs	**RBI**	**Walks**	**Years**
1.136787872	0.863048700	1.936051553	-0.891981297
CAtBat	**CHits**	**CHmRun**	**CRuns**
0.010617601	0.070429194	0.481634991	0.141148163
CRBI	**CWalks**	**LeagueN**	**DivisionW**
0.151233370	0.006795748	30.776192699	-98.889003528
PutOuts	**Assists**	**Errors**	**NewLeagueN**
0.206525967	0.054106867	-2.130578770	4.919650522

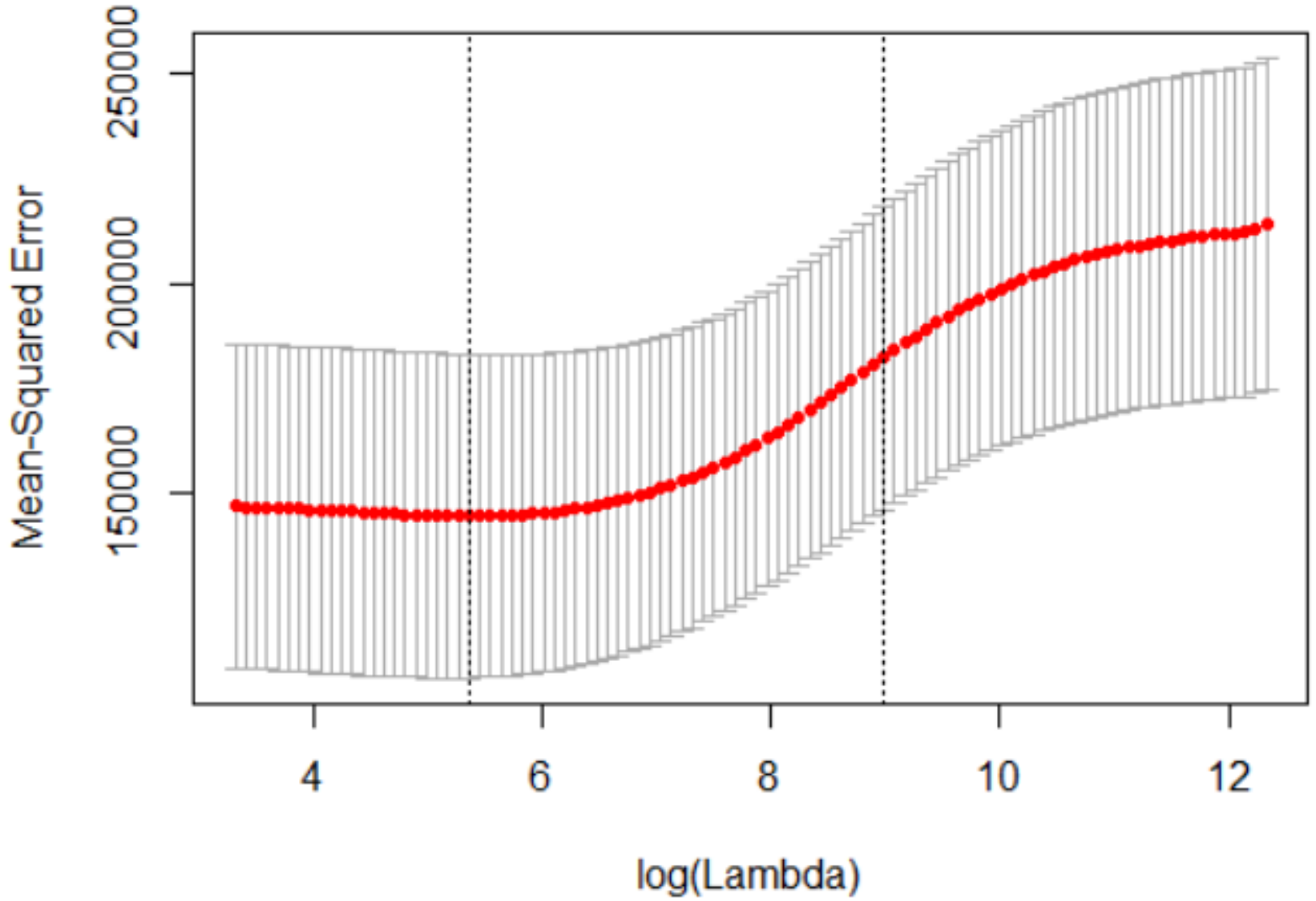
set.seed(1)

```
train=sample(1:nrow(x), nrow(x)/2)
test=(-train); y.test=y[test]
set.seed(1)
cv.out=cv.glmnet(x[train,],y[train],alpha=0)
bestlam=cv.out$lambda.min; bestlam
```

211.74

plot(cv.out)

$$\exp(5.356) = 211.74$$



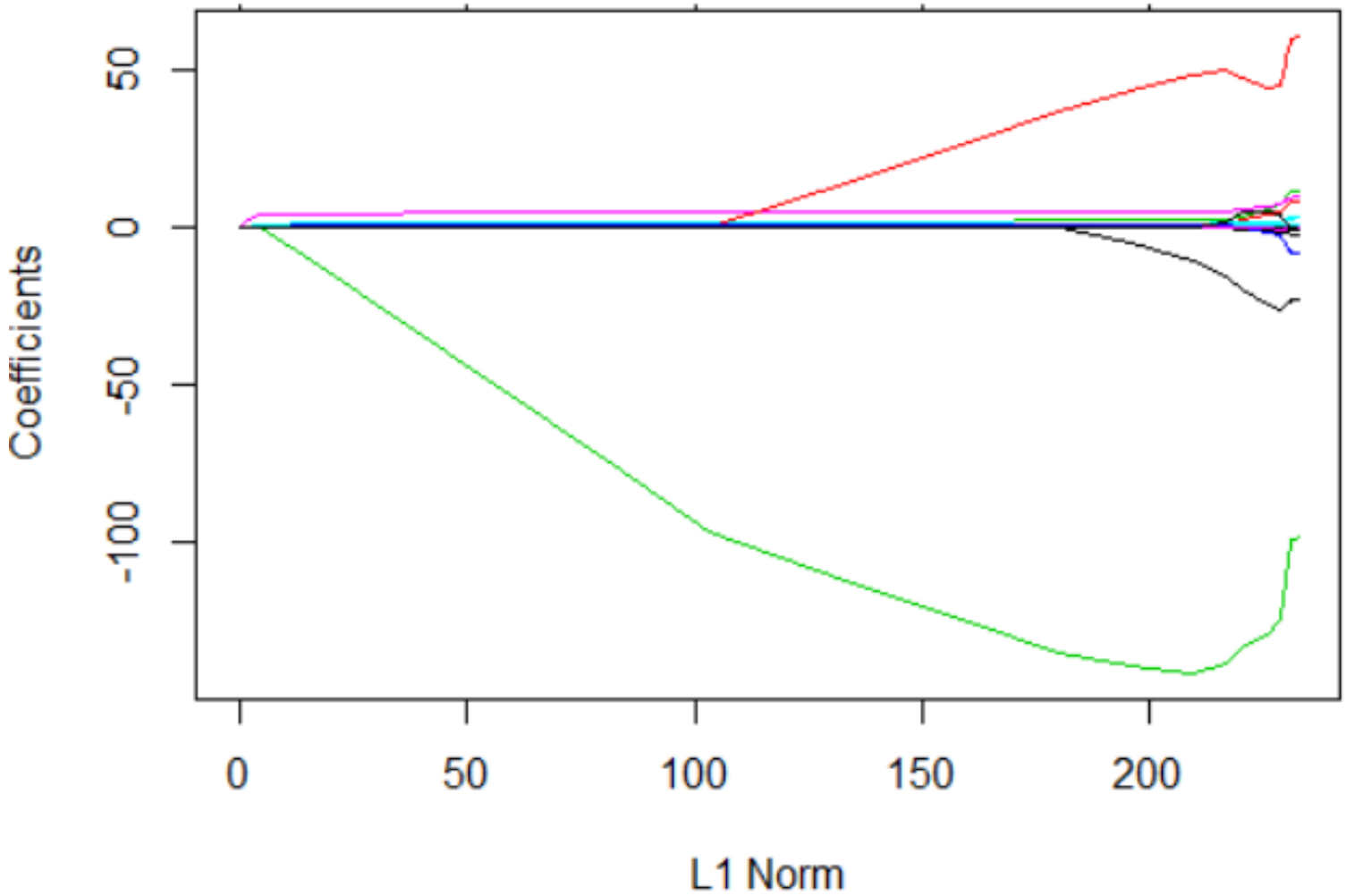
out=glmnet(x,y,alpha=0)

predict(out,type="coefficients",s=bestlam)[1:20,]

(Intercept)	AtBat	Hits	HmRun
9.88487157	0.03143991	1.00882875	0.13927624
Runs	RBI	Walks	Years
1.11320781	0.87318990	1.80410229	0.13074381
CAtBat	CHits	CHmRun	CRuns
0.01113978	0.06489843	0.45158546	0.12900049
CRBI	CWalks	LeagueN	DivisionW
0.13737712	0.02908572	27.18227535	-91.63411299
PutOuts	Assists	Errors	NewLeagueN
0.19149252	0.04254536	-1.81244470	7.21208390

lasso.mod=glmnet(x[train,],y[train], alpha=1,lambda=grid)

plot(lasso.mod)



set.seed(1)

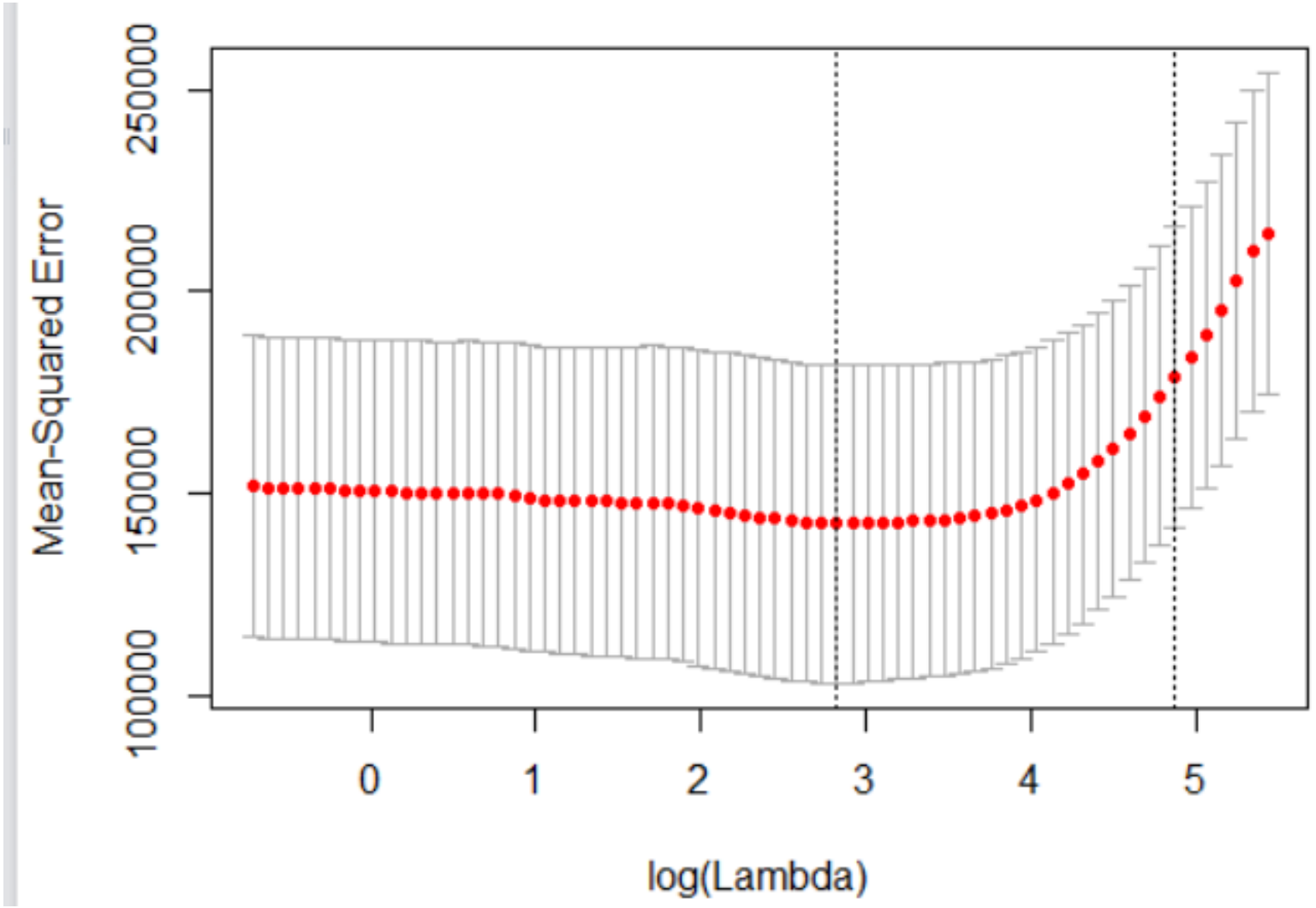
```
cv.out=cv.glmnet(x[train,], y[train],alpha=1)
bestlam=cv.out$lambda.min
bestlam
```

16.78

```
log(bestlam)
```

2.82

plot(cv.out)


Mean-Squared Error
100000
150000
200000
250000
0
1
2
3
4
5
log(Lambda)

out=glmnet(x,y,alpha=1,lambda=grid)

```
lasso.coef=predict(out, type="coefficients",
s=bestlam)[1:20,]
```

(Intercept)	AtBat	Hits	HmRun
18.5394844	0.0000000	1.8735390	0.0000000
Runs	RBI	Walks	Years
0.0000000	0.0000000	2.2178444	0.0000000
CAtBat	CHits	CHmRun	CRuns
0.0000000	0.0000000	0.0000000	0.2071252
CRBI	CWalks	LeagueN	DivisionW
0.4130132	0.0000000	3.2666677	-103.4845458
PutOuts	Assists	Errors	NewLeagueN
0.2204284	0.0000000	0.0000000	0.0000000