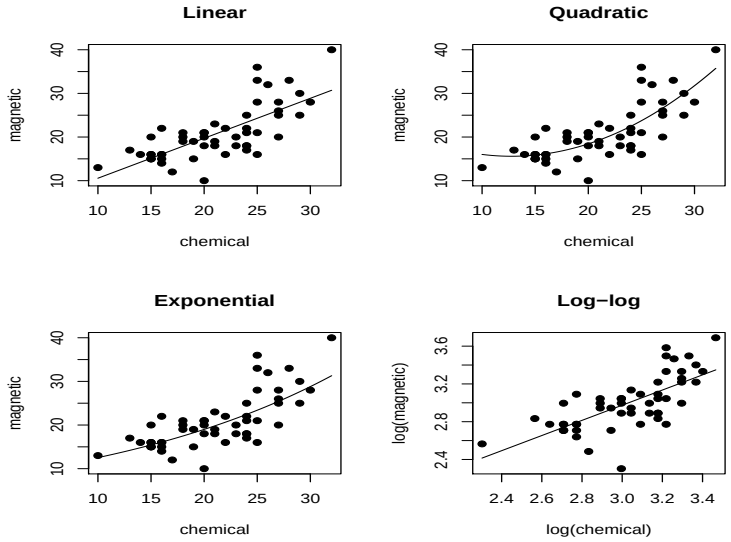
Cross-validation, ridge regression, and bootstrap

```
> par(mfrow=c(2,2))
> head(ironslag)

  chemical magnetic
1       24       25
2       16       22
3       24       17
4       18       21
5       18       20
6       10       13

> attach(ironslag)
> a=seq(min(chemical), max(chemical), length=50)
> #sequence for plotting fitted values
>
> plot(magnetic~chemical, main="Linear",pch=16)
> m1=lm(magnetic~chemical)
> b=coef(m1)
> fitted1=b[1]+b[2]*a
> lines(a, fitted1)
> plot(magnetic~chemical, main="Quadratic",pch=16)
> m2=lm(magnetic~chemical+I(chemical^2))
> b=coef(m2)
> fitted2=b[1]+b[2]*a+b[3]*a^2
> lines(a, fitted2)
> plot(magnetic~chemical,  main="Exponential",pch=16)
> m3=lm(log(magnetic)~chemical)
> b=coef(m3)
> logfitted3=b[1]+b[2]*a
> fitted3=exp(logfitted3)
> lines(a, fitted3)
> plot(log(magnetic)~log(chemical),  main="Log-log",pch=16)
> m4=lm(log(magnetic)~log(chemical))
> b=coef(m4)
> logfitted4=b[1]+b[2]*log(a)
> lines(log(a), logfitted4)
```




```
> #Cross-validation
> #leave one out CV (LOOCV); n-fold CV
>
> n=length(chemical)
> e1 <- e2 <- e3 <- e4 <- numeric(n)
> #n-fold CV
> #fit models on leave-one-out samples
>
> for (k in 1:n){
+   x = chemical[-k] #training sets
+   y = magnetic[-k] #training sets
+   ttx = chemical[k] #testing point
+   tty = magnetic[k]
+
+   J1=lm(y~x)
+   yhat1=J1$coef[1]+J1$coef[2]*ttx
+   e1[k]=tty-yhat1
+
+   J2=lm(y~x+I(x^2))
+   yhat2=J2$coef[1]+J2$coef[2]*ttx+J2$coef[3]*ttx^2
+   e2[k]=tty-yhat2
+
+   J3=lm(log(y)~x)
+   logyhat3=J3$coef[1]+J3$coef[2]*ttx
+   yhat3=exp(logyhat3)
```

```
+   e3[k]=tty-yhat3
+
+   J4=lm(log(y)~log(x))
+   logyhat4=J4$coef[1]+J4$coef[2]*log(ttx)
+   yhat4=exp(logyhat4)
+   e4[k]=tty-yhat4
+ }
> #estimate the mean of the squared prediction errors (MSPE)
> c(mean(e1^2), mean(e2^2), mean(e3^2), mean(e4^2))

[1] 19.55644 17.85248 18.44188 20.45424

> #According to MSPE Model 2 would be the best fit for the data
> library(car)
> J2

Call:
lm(formula = y ~ x + I(x^2))

Coefficients:
(Intercept)            x       I(x^2)
   25.98525     -1.52948      0.05707

> f2=lm(magnetic~chemical+I(chemical^2))
> vif(f2)

     chemical I(chemical^2)
     62.14357      62.14357

> X=model.matrix(f2)
> crossprod(X)

              (Intercept) chemical I(chemical^2)
(Intercept)            53     1120         24990
chemical             1120    24990        584086
I(chemical^2)       24990   584086      14193450

> summary(f2)

Call:
lm(formula = magnetic ~ chemical + I(chemical^2))

Residuals:
    Min      1Q  Median      3Q     Max
```

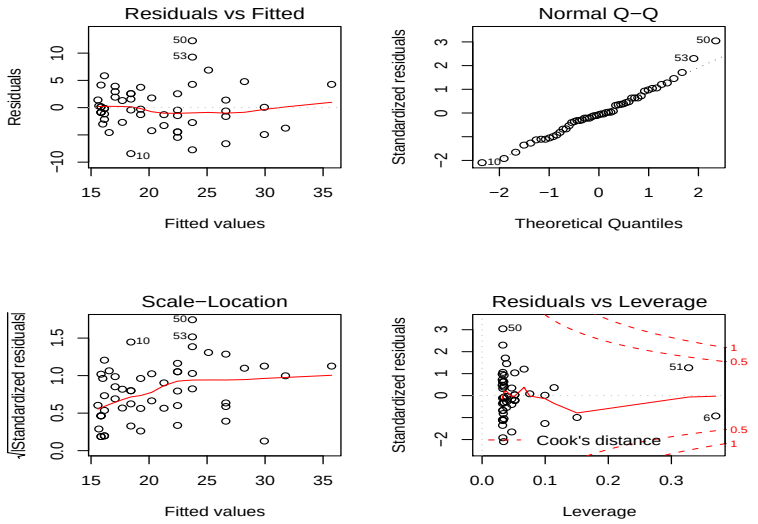
```
-8.4335 -2.7006 -0.2754  2.5446 12.2665

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   24.49262    9.14657   2.678   0.0100
chemical      -1.39334    0.88852  -1.568   0.1232
I(chemical^2)  0.05452    0.02081   2.620   0.0116

Residual standard error: 4.098 on 50 degrees of freedom
Multiple R-squared: 0.5931,        Adjusted R-squared: 0.5768
F-statistic: 36.44 on 2 and 50 DF,  p-value: 1.728e-10
```

```
> s.c=chemical-mean(chemical)
> s.f2=lm(magnetic~s.c+I(s.c^2))
> vif(s.f2)

     s.c I(s.c^2)
1.000276 1.000276
```

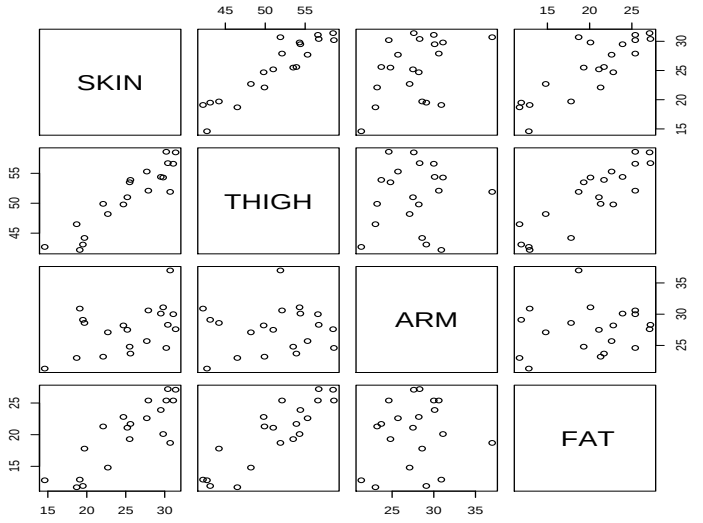
```
> X=model.matrix(s.f2)
> crossprod(X)

             (Intercept)          s.c   I(s.c^2)
(Intercept) 5.300000e+01 6.039613e-14  1322.0755
s.c         6.039613e-14 1.322075e+03   119.0367
I(s.c^2)    1.322075e+03 1.190367e+02 71777.7234
```

```
> summary(s.f2)

Call:
lm(formula = magnetic ~ s.c + I(s.c^2))

Residuals:
    Min      1Q  Median      3Q     Max
-8.4335 -2.7006 -0.2754  2.5446 12.2665

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 19.39475    0.76573   25.33  < 2e-16
s.c          0.91086    0.11273    8.08 1.25e-10
I(s.c^2)     0.05452    0.02081    2.62   0.0116

Residual standard error: 4.098 on 50 degrees of freedom
Multiple R-squared: 0.5931,        Adjusted R-squared: 0.5768
F-statistic: 36.44 on 2 and 50 DF,  p-value: 1.728e-10
```

```
> oldpar <- par(mfrow = c(2,2))
> plot(s.f2)
> par(oldpar)
```




```
> head(bodyfat)

  SKIN THIGH  ARM  FAT
1 19.5  43.1 29.1 11.9
2 24.7  49.8 28.2 22.8
3 30.7  51.9 37.0 18.7
4 29.8  54.3 31.1 20.1
5 19.1  42.2 30.9 12.9
6 25.6  53.9 23.7 21.7

> pairs(bodyfat)
```




```
> #variance inflation factor VIF
> cor(bodyfat)

           SKIN     THIGH       ARM       FAT
SKIN   1.0000000 0.9238425 0.4577772 0.8432654
THIGH  0.9238425 1.0000000 0.0846675 0.8780896
ARM    0.4577772 0.0846675 1.0000000 0.1424440
FAT    0.8432654 0.8780896 0.1424440 1.0000000

> fm1=lm(FAT~., data=bodyfat)
> summary(fm1)

Call:
lm(formula = FAT ~ ., data = bodyfat)

Residuals:
    Min      1Q  Median      3Q     Max
-3.7263 -1.6111  0.3923  1.4656  4.1277

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  117.085     99.782   1.173    0.258
SKIN           4.334      3.016   1.437    0.170
THIGH         -2.857      2.582  -1.106    0.285
ARM           -2.186      1.595  -1.370    0.190
```

```
Residual standard error: 2.48 on 16 degrees of freedom
Multiple R-squared: 0.8014,        Adjusted R-squared: 0.7641
F-statistic: 21.52 on 3 and 16 DF,  p-value: 7.343e-06
```

```
> vif(fm1)

    SKIN    THIGH      ARM
708.8429 564.3434 104.6060
```

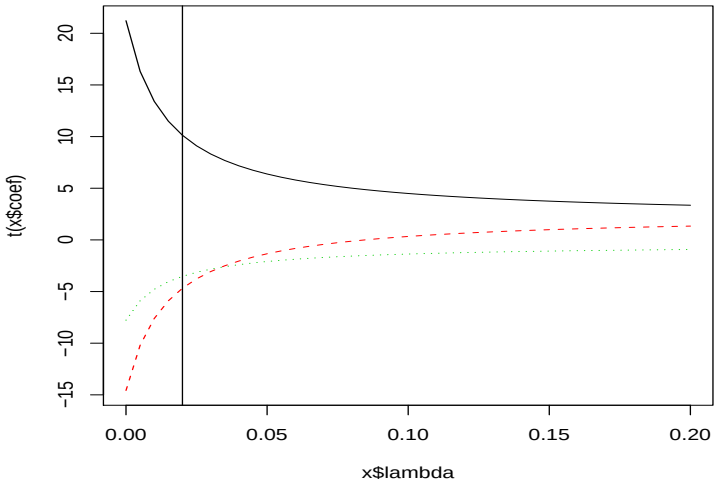
```
> gs=summary(lm(SKIN~THIGH+ARM, data=bodyfat))
> vif.SKIN=1/(1-gs$r.squared)
> vif.SKIN

[1] 708.8429
```

```
> gs=summary(lm(THIGH~SKIN+ARM, data=bodyfat))
> vif.THIGH=1/(1-gs$r.squared)
> vif.THIGH

[1] 564.3434
```

```
> gs=summary(lm(ARM~SKIN+THIGH, data=bodyfat))
> vif.ARM=1/(1-gs$r.squared)
> vif.ARM

[1] 104.606
```

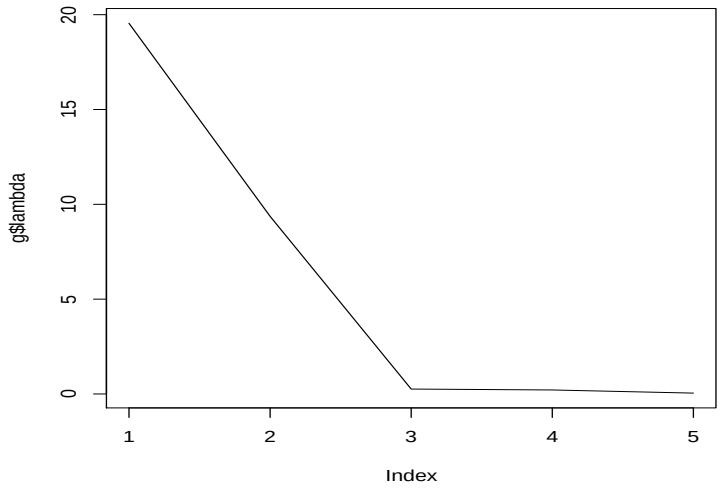
```
> #ridge regression
> library(MASS)
> f2.ridge=lm.ridge(FAT~SKIN+THIGH+ARM, data=bodyfat, lambda=seq(0
> plot(f2.ridge)
> select(f2.ridge)

modified HKB estimator is 0.008505093
modified L-W estimator is 0.3098511
smallest value of GCV  at 0.02
```

```
> lam=0.02
> abline(v=lam)
```

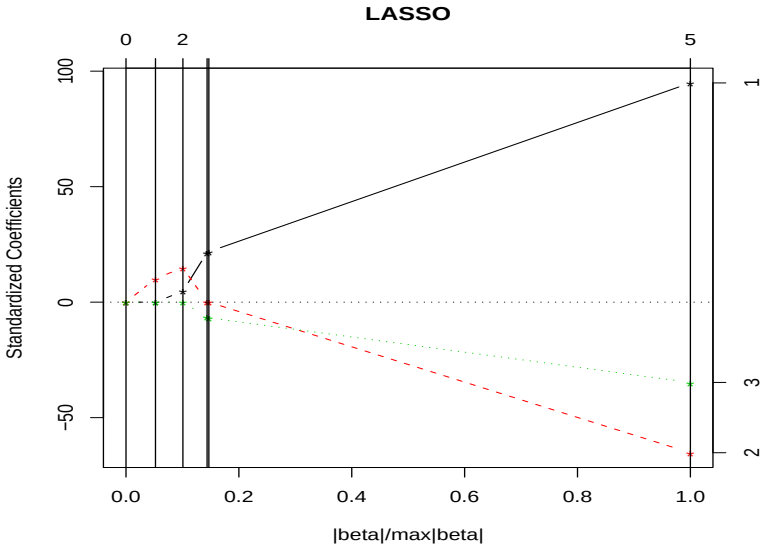



```
> lam=0.02
> r=lm.ridge(FAT~SKIN+THIGH+ARM, data=bodyfat, lambda=lam)
> r$coef

     SKIN     THIGH       ARM
10.126984 -4.682273 -3.527010

> coef(r)

                  SKIN      THIGH        ARM
42.2181995  2.0683914 -0.9177207 -0.9921824

> r$ym

[1] 20.195

> r$xm

  SKIN  THIGH    ARM
25.305 51.170 27.620

> r$scales

    SKIN    THIGH      ARM
4.896067 5.102068 3.554800
```

```
> attach(bodyfat)
> cY <- scale(FAT, scale=FALSE)
> n <- length(cY)
> sX1 <- scale(SKIN) * sqrt(n/(n-1))
> sX2 <- scale(THIGH) *  sqrt(n/(n-1))
> sX3 <- scale(ARM) *  sqrt(n/(n-1))
> ans2 <- lm.ridge(cY ~ sX1 + sX2 + sX3, lambda = lam)
> ans2$coef

      sX1       sX2       sX3
10.126984 -4.682273 -3.527010

> coef(ans2)

                         sX1           sX2           sX3
-2.386183e-15  1.012698e+01 -4.682273e+00 -3.527010e+00

> library(lars)
> xset=cbind(SKIN, THIGH, ARM)
> y=FAT
> g=lars(xset,y)   #lasso least absolute shrinkage and selection o
> plot(g$lambda, type="l")
> g$lambda[1]

[1] 19.54395
```



```
> plot(g)
```

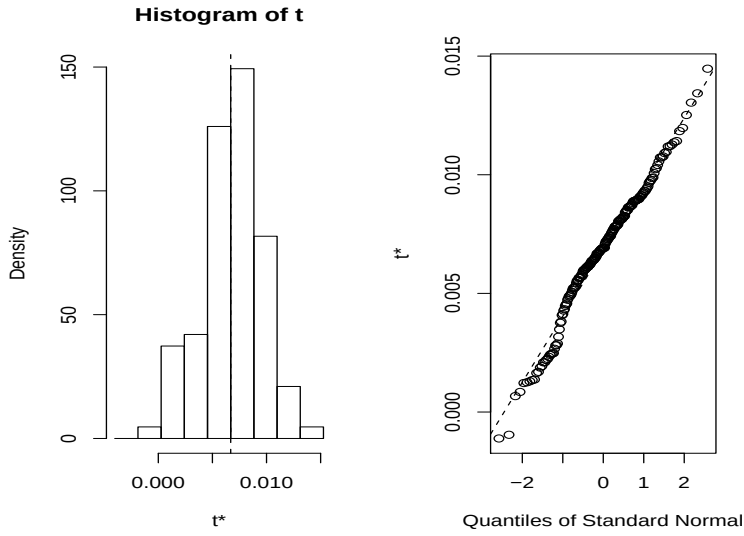



```
> cv.g=cv.lars(xset, y)
> which.min(cv.g$cv)

[1] 10

> #bootstrap
>
> library(boot)
> Fun <- function(a, i)  mean(a[i])
> set.seed(123)
> a=c(3,5,2,1,7)
> mean(a)

[1] 3.6

> R=5 # number of bootstrap samples
> out=boot(a, Fun, R);out

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = a, statistic = Fun, R = R)
```

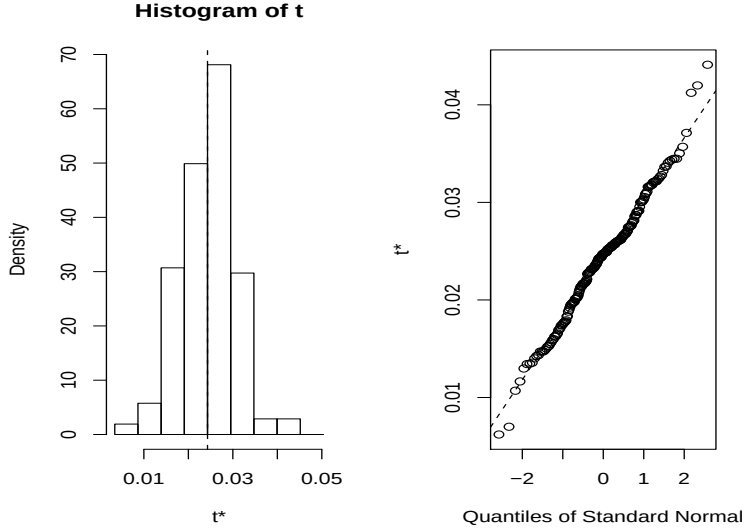
```
Bootstrap Statistics :
    original  bias    std. error
t1*      3.6    0.28    1.432480

> b=boot.array(out); b  #indicate how many time ith obs appeared i

     [,1] [,2] [,3] [,4] [,5]
[1,]    1    1    0    0    3
[2,]    0    1    2    2    0
[3,]    1    0    1    2    1
[4,]    0    1    2    0    2
[5,]    1    0    1    1    2

> tb.theta=(b%*%a)/R; tb.theta

     [,1]
[1,]  5.8
[2,]  2.2
[3,]  2.8
[4,]  4.6
[5,]  4.0

> mean(tb.theta)-mean(a)

[1] 0.28

> sd(tb.theta)

[1] 1.432480

> out=boot(a, Fun, 2000); out

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = a, statistic = Fun, R = 2000)


Bootstrap Statistics :
    original  bias    std. error
t1*      3.6 -0.0322   0.9813484
```

```
> plot(out)

> head(litters) #pig litter data

  lsize bodywt brainwt
1     3  9.447   0.444
2     3  9.780   0.436
3     4  9.155   0.417
4     4  9.613   0.429
5     5  8.850   0.425
6     5  9.610   0.434

> y=litters$brainwt
> x1=litters$lsize
> x2=litters$bodywt
> f0=lm(y~x1+x2, data=litters)
> print(f0)

Call:
lm(formula = y ~ x1 + x2, data = litters)

Coefficients:
(Intercept)           x1           x2
    0.17825      0.00669      0.02431

> summary(f0)

Call:
lm(formula = y ~ x1 + x2, data = litters)

Residuals:
       Min          1Q      Median          3Q         Max
-0.0230005 -0.0098821  0.0004512  0.0092036  0.0180760

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.178247   0.075323   2.366  0.03010
x1          0.006690   0.003132   2.136  0.04751
x2          0.024306   0.006779   3.586  0.00228

Residual standard error: 0.01195 on 17 degrees of freedom
Multiple R-squared: 0.6505,        Adjusted R-squared: 0.6094
F-statistic: 15.82 on 2 and 17 DF,  p-value: 0.0001315
```

```
> anova(f0)

Analysis of Variance Table

Response: y
          Df     Sum Sq    Mean Sq F value    Pr(>F)
x1         1 0.00268418 0.00268418  18.788 0.0004502
x2         1 0.00183687 0.00183687  12.857 0.0022784
Residuals 17 0.00242870 0.00014286

> # random x : case resampling
> boot.litters <- function(data, i){
+   b.data <- data[i,] # select obs. in bootstrap sample
+   y=b.data$brainwt
+   x1=b.data$lsize
+   x2=b.data$bodywt
+   mod <- lm(y ~ x1 + x2, data=b.data)
+   coefficients(mod) # return coefficient vector
+ }
> library(boot)
> out <- boot(litters, boot.litters, 200);out

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = litters, statistic = boot.litters, R = 200)


Bootstrap Statistics :
       original        bias    std. error
t1* 0.178246962 -5.958688e-04 0.067741874
t2* 0.006690331  6.984422e-05 0.002783938
t3* 0.024306344  3.580500e-05 0.006159927

> plot(out, index=2)
```




```
> plot(out, index=3)
```




```
> # fixed x: model-based resampling
>
> fit <- fitted(f0)
> e <- residuals(f0)
> X <- model.matrix(f0)[,-1]
```

```
> boot.litters.fixed <- function(data, i){
+   y <- fit + e[i]
+   mod <- lm(y ~ X )
+   coefficients(mod)
+ }
> out.fix <- boot( litters, boot.litters.fixed, 200); out.fix

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = litters, statistic = boot.litters.fixed, R = 200)


Bootstrap Statistics :
       original        bias    std. error
t1* 0.178246962  8.693893e-04 0.071398245
t2* 0.006690331 -3.196501e-05 0.002966800
t3* 0.024306344 -9.890197e-05 0.006400178
```