Shuting Chen

Midterm

```
> setwd("/Users/shutingchen/Desktop/ISE 529          Data Analytics/Midterm")
> d0=read.csv("homes.csv",header = T)
```

> #1

> #a

```
>
> d0$style=as.factor(d0$style)
> d0$age=2018-d0$year
> str(d0)
> d0$year=NULL
> d1=d0[,c(1,2,3,4,5,7,12)]
> cor(d1)
            price       area       beds      baths     garage   lotsize        age
price    1.0000000  0.8194701  0.4133239  0.6836854  0.5777863 0.2241685 -0.5555164
area     0.8194701  1.0000000  0.5578378  0.7552729  0.5337665 0.1575247 -0.4411967
beds     0.4133239  0.5578378  1.0000000  0.5834469  0.3168137 0.1265384 -0.2686924
baths    0.6836854  0.7552729  0.5834469  1.0000000  0.4898981 0.1470066 -0.5128410
garage   0.5777863  0.5337665  0.3168137  0.4898981  1.0000000 0.1522193 -0.4617604
lotsize  0.2241685  0.1575247  0.1265384  0.1470066  0.1522193 1.0000000  0.1004519
age     -0.5555164 -0.4411967 -0.2686924 -0.5128410 -0.4617604 0.1004519  1.0000000
> #        area       beds      baths     garage   lotsize        age
> #price   0.8194701  0.4133239  0.6836854  0.5777863 0.2241685 -0.5555164
> #r2 is the square of cor(d1)
> #so the best numeric predictor is the area the r^2 is 0.8194701*0.8194701=0.67153
> m1=lm(price~style,d0)
> summary(m1)
> #Multiple R-squared:  0.1741
> m2=lm(price~ac,d0)
> #Multiple R-squared:  0.08329
> m3=lm(price~pool,d0)
> summary(m3)
> #Multiple R-squared:  0.02149
> m4=lm(price~quality,d0)
> #Multiple R-squared:  0.6601
> m5=lm(price~highway,d0)
> summary(m5)
> #Multiple R-squared:  0.002598
```

```
> #best predictor is area and worst predictors is highway
```

> #b

```
> library(MASS)
> set.seed(1)
> m0=lm(price~.,d0)
> step1=stepAIC(m0)

Step:  AIC=11454.88
price ~ area + baths + garage + style + lotsize + quality + highway +
    age

          Df  Sum of Sq        RSS   AIC
<none>                  1.6518e+12 11455
- garage   1 1.1667e+10 1.6635e+12 11457
- highway  1 1.4252e+10 1.6661e+12 11457
- baths    1 1.6305e+10 1.6681e+12 11458
- style    9 1.2195e+11 1.7738e+12 11474
- lotsize  1 1.0379e+11 1.7556e+12 11485
- age      1 1.5566e+11 1.8075e+12 11500
- quality  2 5.5591e+11 2.2078e+12 11602
- area     1 5.7865e+11 2.2305e+12 11610
> m1=lm(price~area + baths + garage + style + lotsize + quality + highway + age,data=d0)
> summary(m1)
>#Multiple R-squared:  0.8333,
> d1 = d0
> levels(d1$style)[-c(2,7)]="0"
> m1=lm(price~area + baths + garage + style + lotsize + quality + highway + age,data=d1)
> summary(m1)
> #Multiple R-squared:  0.829
> r1=data.frame(fitted=m1$fitted.values,observed=d0$price)
> b=cor(r1)
> r2=b^2
> r2
            fitted  observed
fitted   1.0000000 0.8289666
observed 0.8289666 1.0000000
> #0.8289666
> #so we can verify the r^2 is equal to the square of the correlation between the fitted and observed prices
```

> #c

```
> n1=nrow(d0)
> step2=stepAIC(m0,k=log(n1))
Step:  AIC=11504.33
price ~ area + lotsize + quality + age

           Df  Sum of Sq        RSS   AIC
<none>                   1.8106e+12 11504
- lotsize  1 1.6211e+11 1.9727e+12 11543
- age      1 1.8972e+11 2.0003e+12 11550
- quality  2 7.8661e+11 2.5972e+12 11680
- area     1 1.1251e+12 2.9357e+12 11750
> m2=lm(price ~ area + lotsize + quality + age,data=d0)
> summary(m2)

Call:
lm(formula = price ~ area + lotsize + quality + age, data = d0)

Residuals:
    Min      1Q  Median      3Q     Max
-218538  -28132   -4808   21918  303866

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    2.344e+05  2.002e+04  11.711  < 2e-16 ***
area           9.332e+01  5.212e+00  17.907  < 2e-16 ***
lotsize        1.570e+00  2.309e-01   6.797 2.96e-11 ***
qualityLOW    -1.614e+05  1.354e+04 -11.924  < 2e-16 ***
qualityMEDIUM -1.490e+05  9.985e+03 -14.923  < 2e-16 ***
age           -1.413e+03  1.922e+02  -7.353 7.65e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 59240 on 516 degrees of freedom
Multiple R-squared:  0.8173,  Adjusted R-squared:  0.8155
F-statistic: 461.7 on 5 and 516 DF,  p-value: < 2.2e-16

> newval1=data.frame(area=3500,lotsize=24000,quality="MEDIUM",age=50)
> predict(m2,newval1)
       1
379019.2
> #price of it is 379019.2
>
```

> #d

```
> set.seed(1)
> library(boot)
> m1=glm(price~area + baths + garage + style + lotsize + quality + highway + age,data=d1)
> cverrors1=cv.glm(d1,m1,K=7)$delta[1]
> cverrors1
[1] 3438713508
> #mspe of model 1 is 3438713508
> set.seed(1)
> m2=glm(price ~ area + lotsize + quality + age,data=d0)
> cverrors2=cv.glm(d0,m2,K=7)$delta[1]
> cverrors2
[1] 3613129176
> #mspe of model 2 is 3613129176
>
>
```

> #2

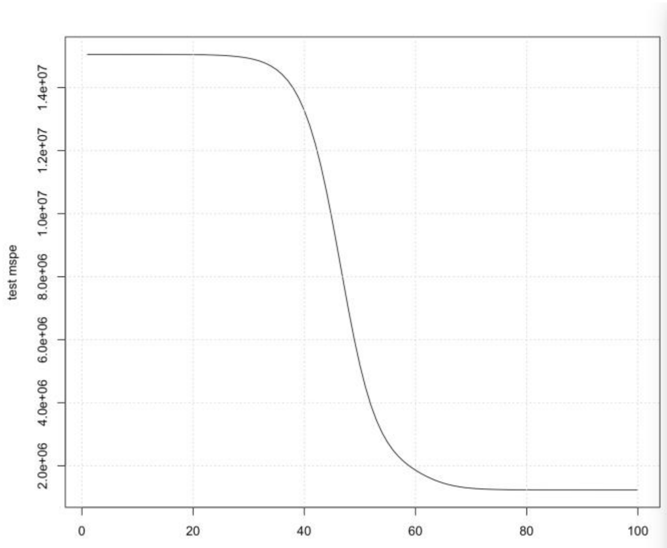
> #a

```
> set.seed(1)
> library(ISLR)
> #library(Matrix)
> #library(foreach)
> library(glmnet)
> d2=College
> str(d2)
'data.frame':   777 obs. of  18 variables:
 $ Private    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
 $ Apps       : num  1660 2186 1428 417 193 ...
 $ Accept     : num  1232 1924 1097 349 146 ...
 $ Enroll     : num  721 512 336 137 55 158 103 489 227 172 ...
 $ Top10perc  : num  23 16 22 60 16 38 17 37 30 21 ...
 $ Top25perc  : num  52 29 50 89 44 62 45 68 63 44 ...
 $ F.Undergrad: num  2885 2683 1036 510 249 ...
 $ P.Undergrad: num  537 1227 99 63 869 ...
 $ Outstate   : num  7440 12280 11250 12960 7560 ...
 $ Room.Board : num  3300 6450 3750 5450 4120 ...
 $ Books      : num  450 750 400 450 800 500 500 450 300 660 ...
 $ Personal   : num  2200 1500 1165 875 1500 ...
 $ PhD        : num  70 29 53 92 76 67 90 89 79 40 ...
 $ Terminal   : num  78 30 66 97 72 73 93 100 84 41 ...
 $ S.F.Ratio  : num  18.1 12.2 12.9 7.7 11.9 9.4 11.5 13.7 11.3 11.5 ...
 $ perc.alumni: num  12 16 30 37 2 11 26 37 23 15 ...
 $ Expend     : num  7041 10527 8735 19016 10922 ...
 $ Grad.Rate  : num  60 56 54 59 15 55 63 73 80 52 ...
```

```
> n2=nrow(d2)
> n2
[1] 777
>
> x=model.matrix(Apps~.,d2)[,-1]
> y=d2$Apps
> # lambdas from 10^10 to 10^{-2}
> a = seq(from=10,to=-2,length=100)
> grid=10^a
> models=glmnet(x,y,alpha=0,lambda=grid)
> cv.out=cv.glmnet(x,y,alpha=0,lambda=grid,nfolds=10)
> cv.out$cvm
  [1] 15052158 15052146 15052130 15052109 15052081 15052044 15051995 15051930
15051845 15051732
 [11] 15051582 15051385 15051124 15050779 15050323 15049720 15048923 15047869
15046477 15044637
 [21] 15042204 15038990 15034743 15029133 15021722 15011935 14999016 14981968
14959487 14929865
 [31] 14890975 14839791 14772659 14684827 14570289 14421561 14229511 13983306
13670610 13278186
 [41] 12793126 12204862 11508005 10705670  9812463  8855788  7874084  6911465  6009575
5200474
 [51]  4501718  3915840  3436091  3047975  2737036  2487173  2285230  2119022  1979332
1859468
 [61]  1754344  1661231  1579596  1508953  1449431  1399974  1360303  1329001  1304727
1286385
 [71]  1272758  1262359  1254725  1249030  1244898  1241694  1239512  1237728  1236558
1235548
 [81]  1234871  1234286  1233842  1233636  1233388  1233257  1233124  1232992  1232864
1232740
 [91]  1232622  1232511  1232408  1232312  1232224  1232143  1232071  1232005  1231946
1231893
>
```

> #b

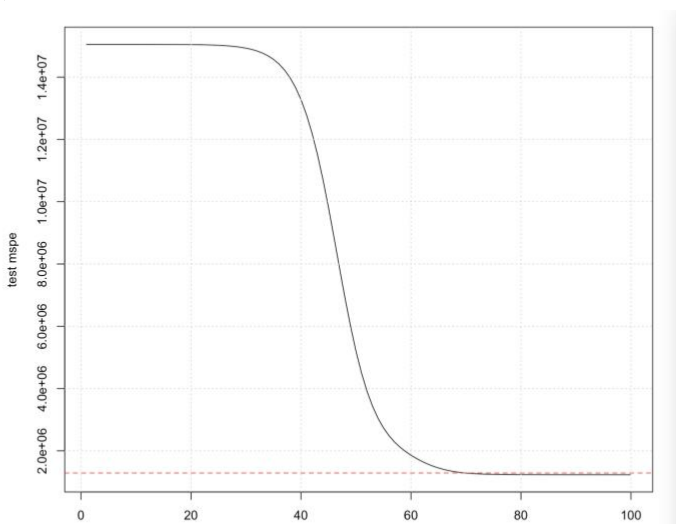
```
> plot(cv.out$cvm,type="l",ylab="test mspe",xlab="")
> grid()
>
```



>

> #c

```
> set.seed(1)
> m3=glm(Apps~.,data=d2)
> cv.glm(d2,m3,K=10)$delta[1]
[1] 1287415
> abline(h=1287415,lty=2,col="red")
>
```



> #3

> #a

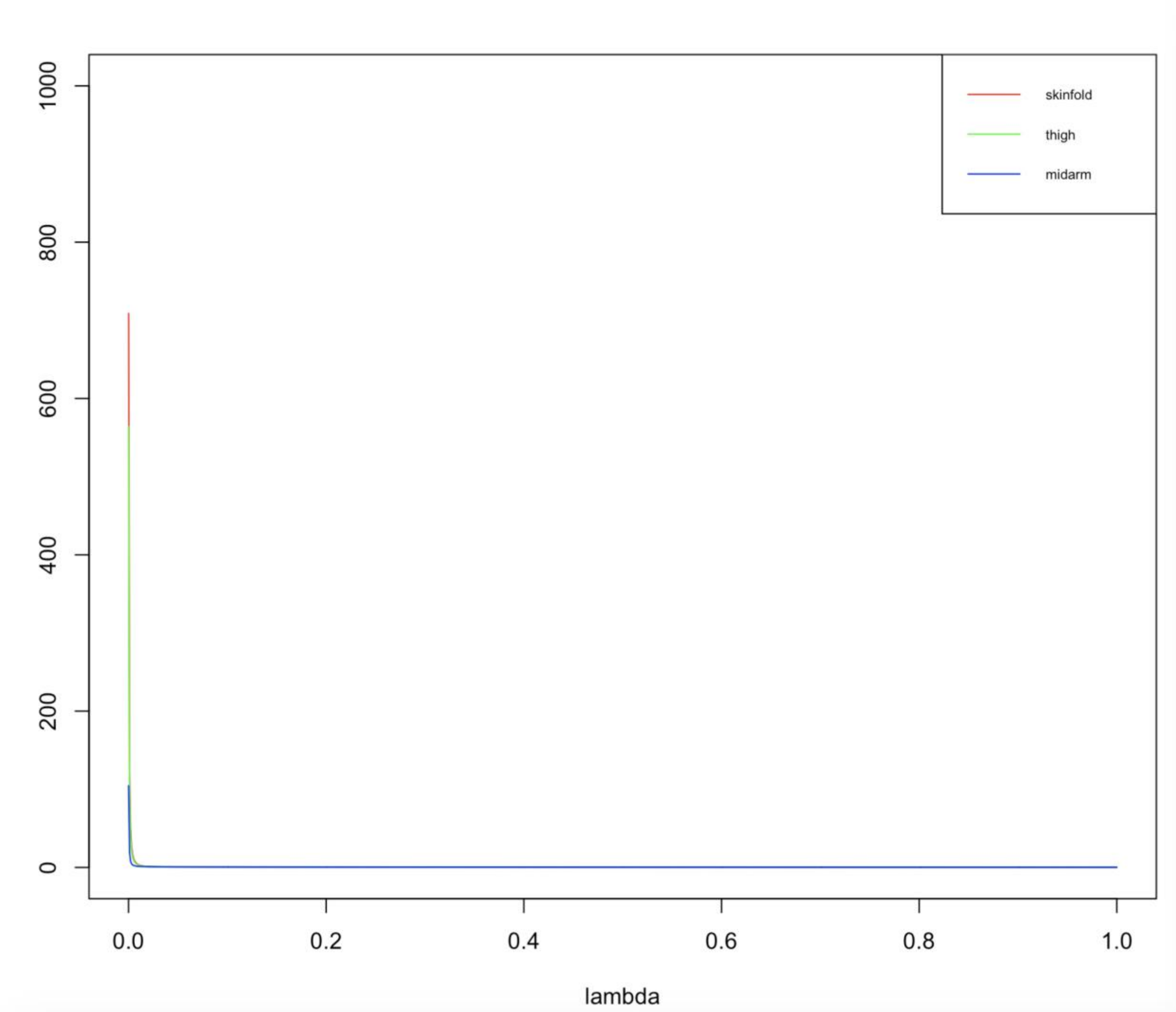
```
> d3=read.csv("bodyfat.csv",header = T)
> d4=d3[,-4]
> str(d4)
'data.frame':	20 obs. of  3 variables:
 $ skinfold: num  19.5 24.7 30.7 29.8 19.1 25.6 31.4 27.9 22.1 25.5 ...
```

```
 $ thigh   : num  43.1 49.8 51.9 54.3 42.2 53.9 58.5 52.1 49.9 53.5 ...
 $ midarm  : num  29.1 28.2 37 31.1 30.9 23.7 27.6 30.6 23.2 24.8 ...
> rxx=cor(d4)
> rxx
         skinfold     thigh    midarm
skinfold 1.0000000 0.9238425 0.4577772
thigh    0.9238425 1.0000000 0.0846675
midarm   0.4577772 0.0846675 1.0000000
> #low correlate predictors are midarm and thigh
> m4=lm(midarm~.,d4)
> summary(m4)

Call:
lm(formula = midarm ~ ., data = d4)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.58200 -0.30625  0.02592  0.29526  0.56102 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 62.33083    1.23934   50.29   <2e-16 ***
skinfold     1.88089    0.04498   41.82   <2e-16 ***
thigh       -1.60850    0.04316  -37.26   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.377 on 17 degrees of freedom
Multiple R-squared:  0.9904,  Adjusted R-squared:  0.9893 
F-statistic: 880.7 on 2 and 17 DF,  p-value: < 2.2e-16

> #Multiple R-squared:  0.9904, and as we can see from p-value of summary p-value: < 2.2e-16
> #so skinfold and thigh are both klinearly releted to midarm
```

> #b

```
> det(rxx)
[1] 0.001400637
> #0.001400637>0
> solve(rxx)
         skinfold     thigh    midarm
skinfold  708.8429 -631.9152 -270.9894
thigh    -631.9152  564.3434  241.4948
midarm   -270.9894  241.4948  104.6060
> lambda=seq(0,1,0.001)
```

```
> length(lambda)
[1] 1001
> m=matrix(0,1001,3)
> id=diag(1,3)
> for(i in 1:1001){
+   bb1=solve(rxx + (lambda[i] * id))
+   bb2=bb1%*%rxx%*%bb1
+   m[i,]=diag(bb2)
+ }
> colnames(m) = c("skinfold", "thigh","midarm")
> head(m)
      skinfold      thigh     midarm
[1,] 708.84291 564.343386 104.606005
[2,] 125.73087 100.274032  19.280967
[3,]  50.55919  40.448310   8.279700
[4,]  27.17501  21.837601   4.856184
[5,]  16.98157  13.724723   3.362792
[6,]  11.64342   9.475922   2.579850
> d5=cbind(m,lambda)
> d5=data.frame(d5)
> head(d5)
   skinfold      thigh     midarm lambda
1 708.84291 564.343386 104.606005  0.000
2 125.73087 100.274032  19.280967  0.001
3  50.55919  40.448310   8.279700  0.002
4  27.17501  21.837601   4.856184  0.003
5  16.98157  13.724723   3.362792  0.004
6  11.64342   9.475922   2.579850  0.005
> plot(1, type="n", xlab="lambda", ylab="", xlim=c(0, 1), ylim=c(0, 1000))
>
lines(d5$lambda[order(d5$lambda,decreasing=TRUE)],d5$skinfold[order(d5$skinfold)],col='red
',type = "l")
>
lines(d5$lambda[order(d5$lambda,decreasing=TRUE)],d5$thigh[order(d5$skinfold)],col='green'
,type = "l")
>
lines(d5$lambda[order(d5$lambda,decreasing=TRUE)],d5$midarm[order(d5$skinfold)],col='blu
e',type = "l")
> legend("topright",c("skinfold","thigh","midarm"),cex = 0.6,lty=1,col = c("red","green","blue"))
>
```



skinfold
thigh
midarm
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
lambda