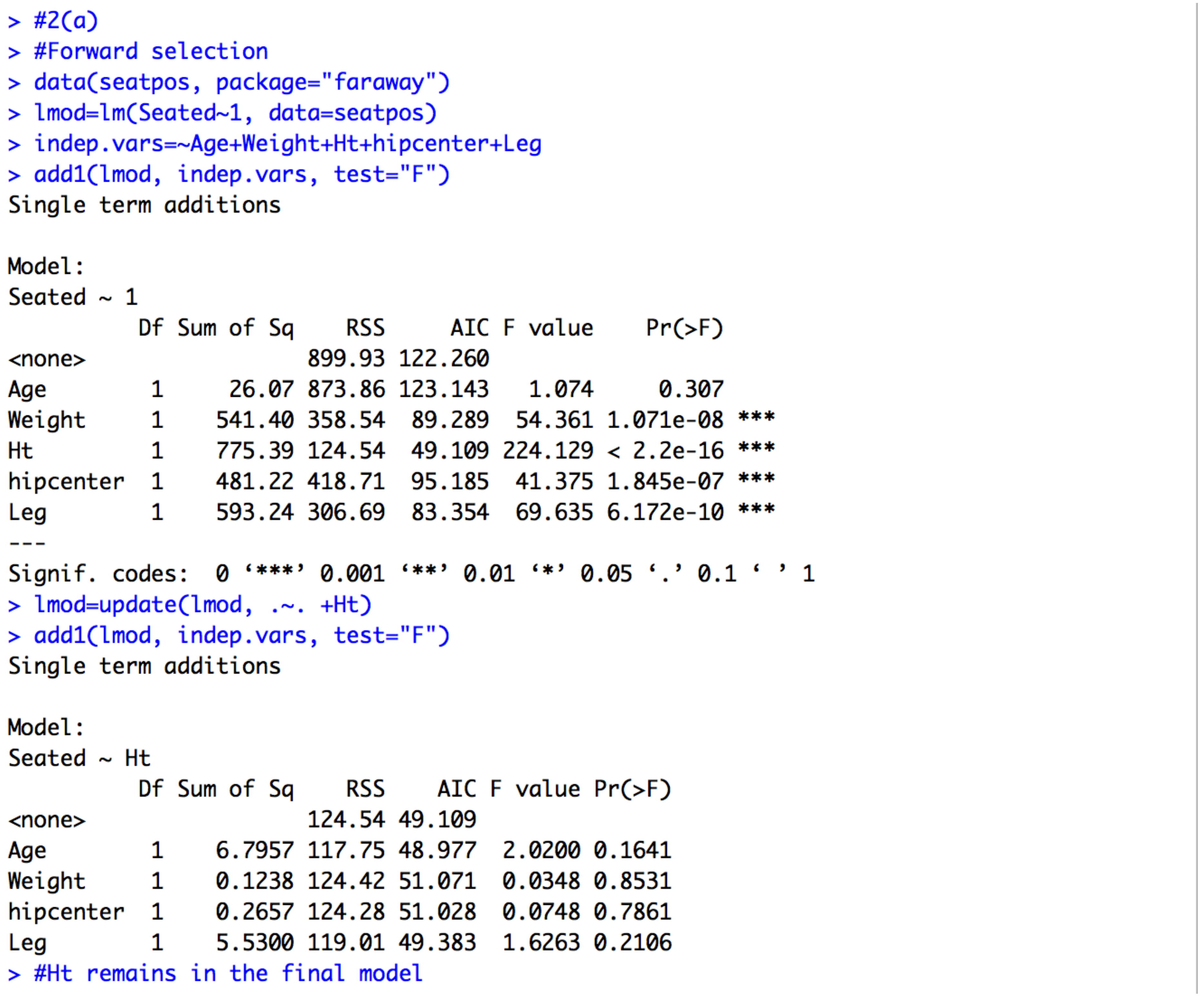
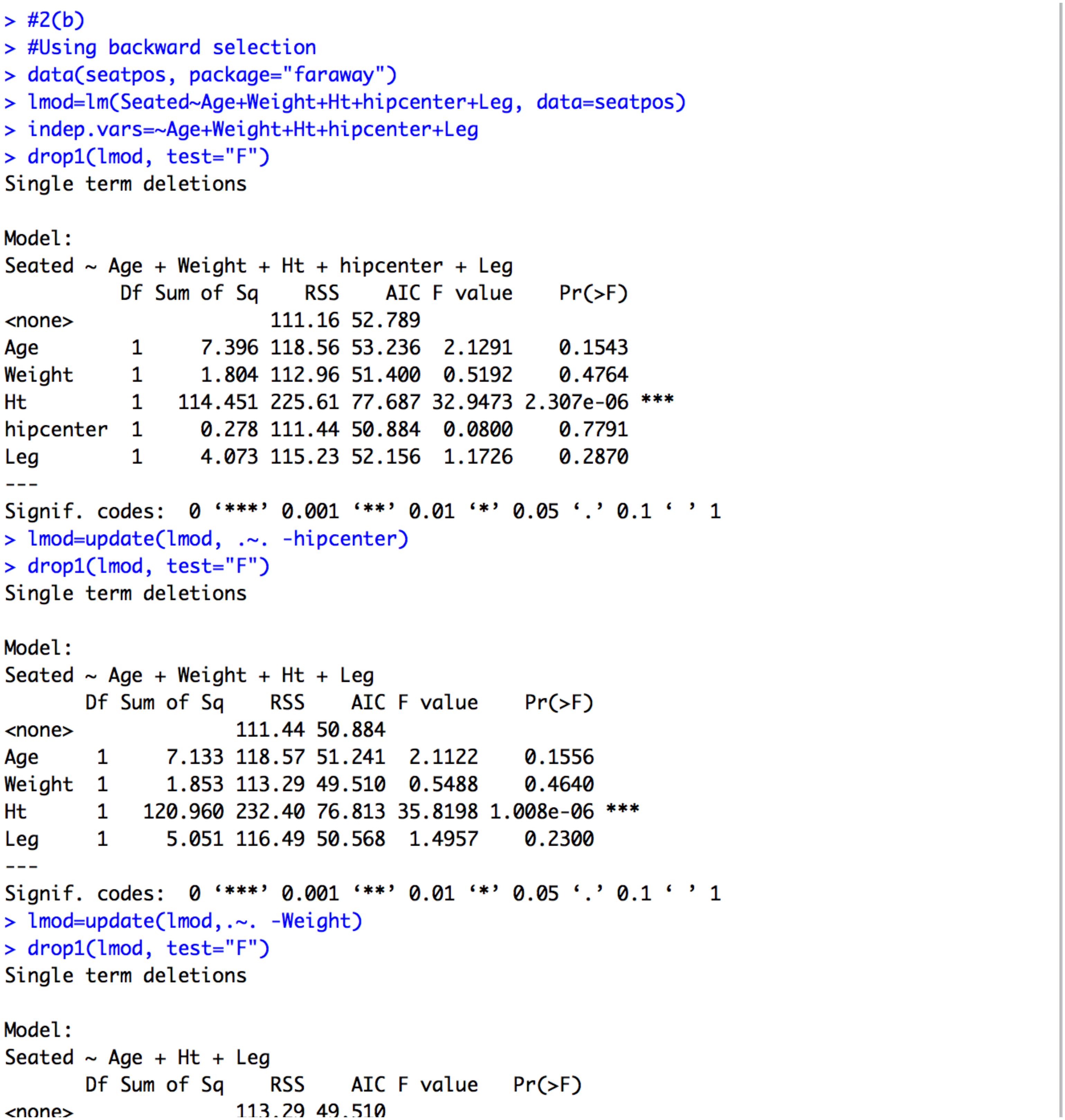
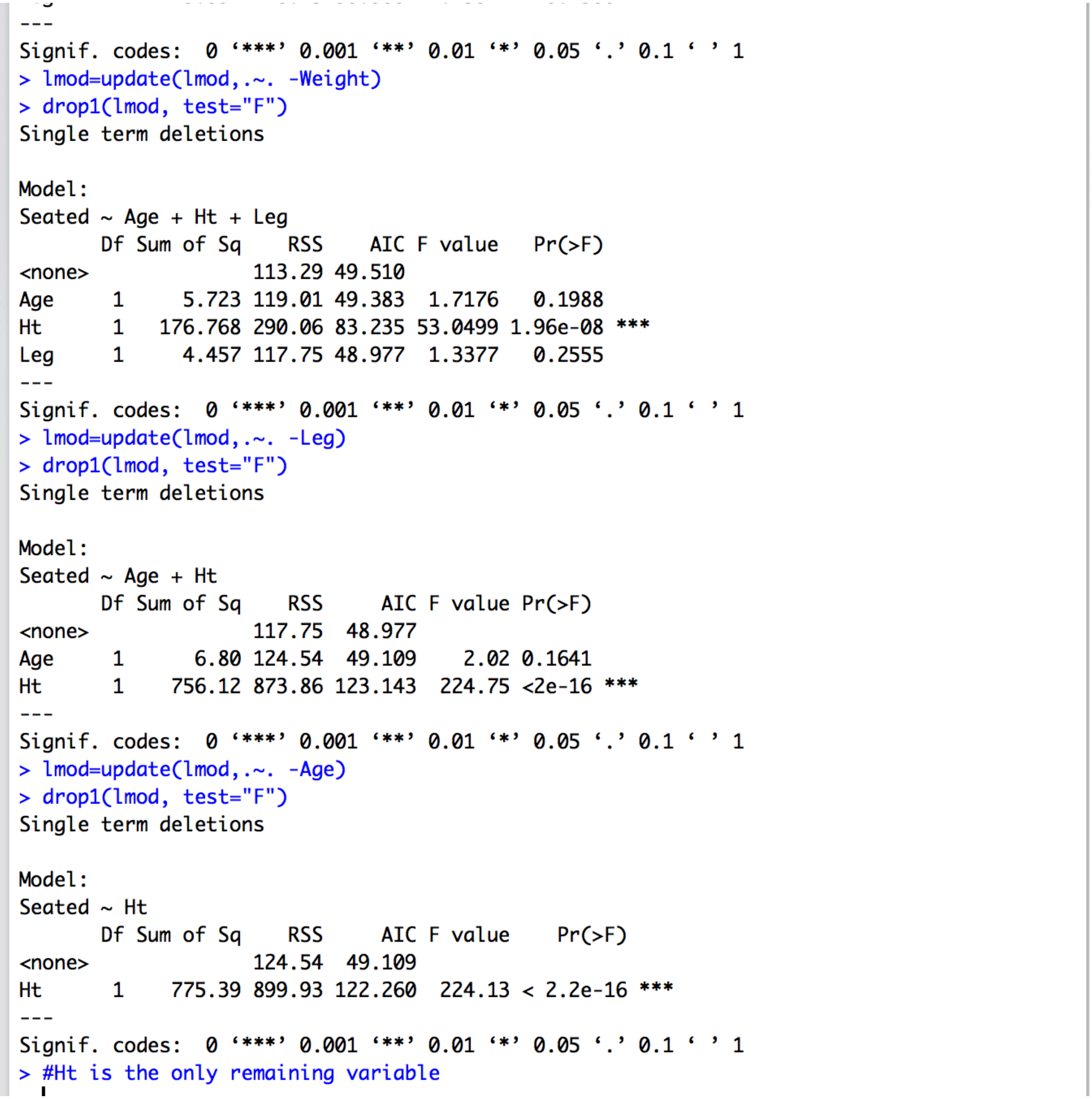


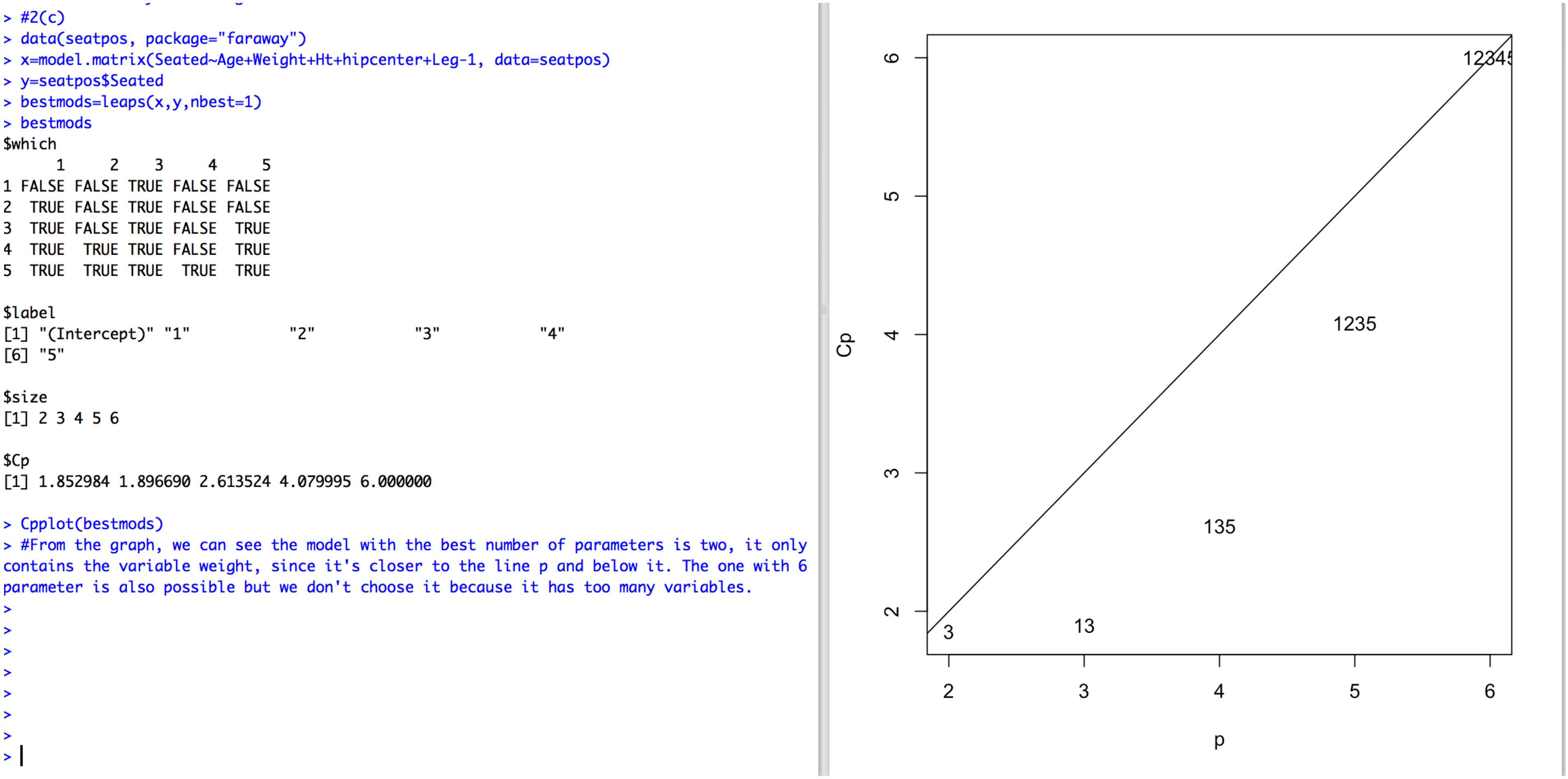
```
> #2(a)
> #Forward selection
> data(seatpos, package="faraway")
> lmod=lm(Seated~1, data=seatpos)
> indep.vars=~Age+Weight+Ht+hipcenter+Leg
> add1(lmod, indep.vars, test="F")
Single term additions

Model:
Seated ~ 1
          Df Sum of Sq    RSS     AIC F value    Pr(>F)    
<none>                 899.93 122.260                      
Age        1     26.07 873.86 123.143   1.074     0.307    
Weight     1    541.40 358.54  89.289  54.361 1.071e-08 ***
Ht         1    775.39 124.54  49.109 224.129 < 2.2e-16 ***
hipcenter  1    481.22 418.71  95.185  41.375 1.845e-07 ***
Leg        1    593.24 306.69  83.354  69.635 6.172e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> lmod=update(lmod, .~. +Ht)
> add1(lmod, indep.vars, test="F")
Single term additions

Model:
Seated ~ Ht
          Df Sum of Sq    RSS    AIC F value Pr(>F)
<none>                 124.54 49.109               
Age        1    6.7957 117.75 48.977  2.0200 0.1641
Weight     1    0.1238 124.42 51.071  0.0348 0.8531
hipcenter  1    0.2657 124.28 51.028  0.0748 0.7861
Leg        1    5.5300 119.01 49.383  1.6263 0.2106
> #Ht remains in the final model
```



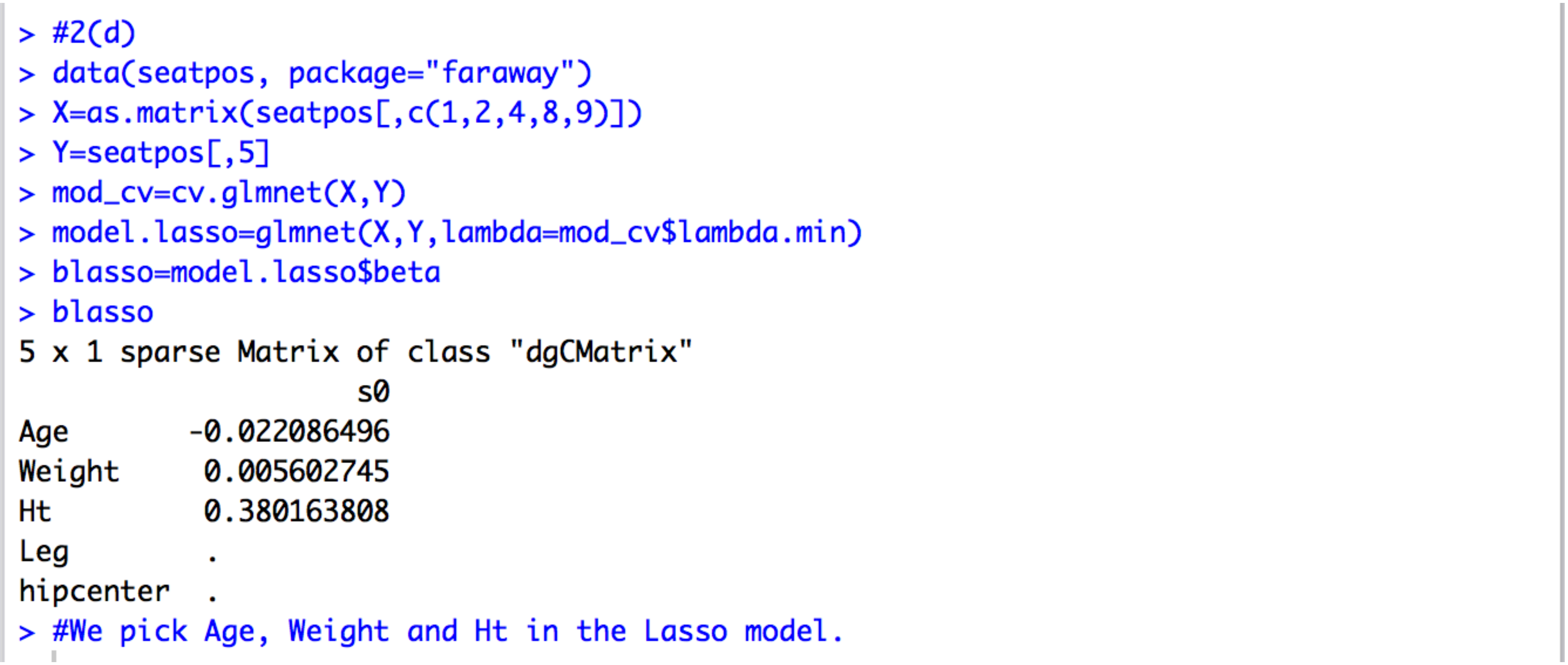
```
> #2(b)
> #Using backward selection
> data(seatpos, package="faraway")
> lmod=lm(Seated~Age+Weight+Ht+hipcenter+Leg, data=seatpos)
> indep.vars=~Age+Weight+Ht+hipcenter+Leg
> drop1(lmod, test="F")
Single term deletions

Model:
Seated ~ Age + Weight + Ht + hipcenter + Leg
          Df Sum of Sq    RSS    AIC F value    Pr(>F)    
<none>                 111.16 52.789                      
Age        1     7.396 118.56 53.236  2.1291    0.1543    
Weight     1     1.804 112.96 51.400  0.5192    0.4764    
Ht         1   114.451 225.61 77.687 32.9473 2.307e-06 ***
hipcenter  1     0.278 111.44 50.884  0.0800    0.7791    
Leg        1     4.073 115.23 52.156  1.1726    0.2870    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> lmod=update(lmod, .~. -hipcenter)
> drop1(lmod, test="F")
Single term deletions

Model:
Seated ~ Age + Weight + Ht + Leg
       Df Sum of Sq    RSS    AIC F value    Pr(>F)    
<none>              111.44 50.884                      
Age     1     7.133 118.57 51.241  2.1122    0.1556    
Weight  1     1.853 113.29 49.510  0.5488    0.4640    
Ht      1   120.960 232.40 76.813 35.8198 1.008e-06 ***
Leg     1     5.051 116.49 50.568  1.4957    0.2300    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> lmod=update(lmod,.~. -Weight)
> drop1(lmod, test="F")
Single term deletions

Model:
Seated ~ Age + Ht + Leg
       Df Sum of Sq    RSS    AIC F value   Pr(>F)    
<none>              113.29 49.510
```



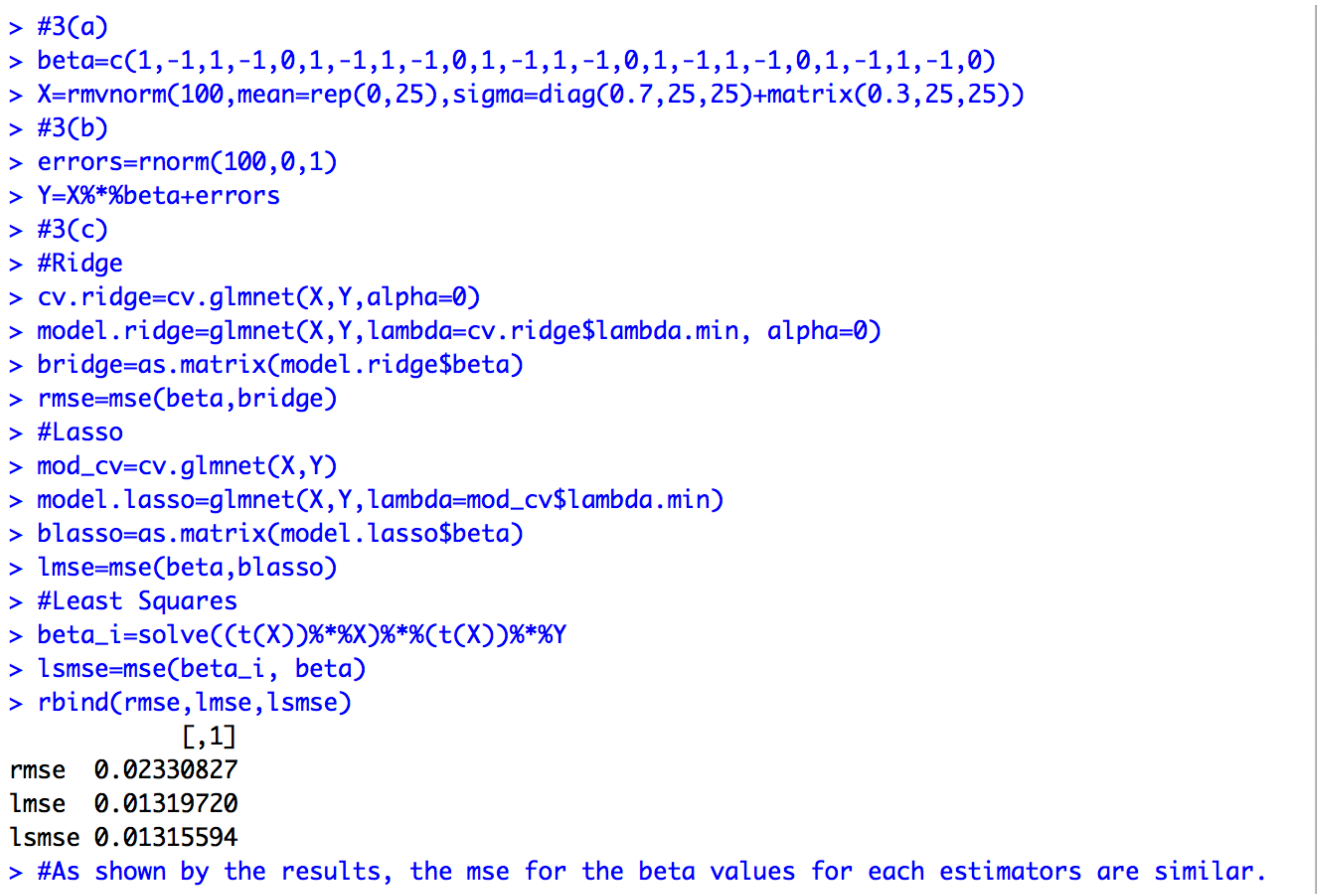
```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> lmod=update(lmod,.~. -Weight)
> drop1(lmod, test="F")
Single term deletions

Model:
Seated ~ Age + Ht + Leg
       Df Sum of Sq    RSS    AIC F value    Pr(>F)    
<none>              113.29 49.510                      
Age     1     5.723 119.01 49.383  1.7176    0.1988    
Ht      1   176.768 290.06 83.235 53.0499 1.96e-08 ***
Leg     1     4.457 117.75 48.977  1.3377    0.2555    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> lmod=update(lmod,.~. -Leg)
> drop1(lmod, test="F")
Single term deletions

Model:
Seated ~ Age + Ht
       Df Sum of Sq    RSS     AIC F value Pr(>F)    
<none>              117.75  48.977                   
Age     1      6.80 124.54  49.109    2.02 0.1641    
Ht      1    756.12 873.86 123.143  224.75 <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> lmod=update(lmod,.~. -Age)
> drop1(lmod, test="F")
Single term deletions

Model:
Seated ~ Ht
       Df Sum of Sq    RSS     AIC F value    Pr(>F)    
<none>              124.54  49.109                      
Ht      1    775.39 899.93 122.260  224.13 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> #Ht is the only remaining variable
```



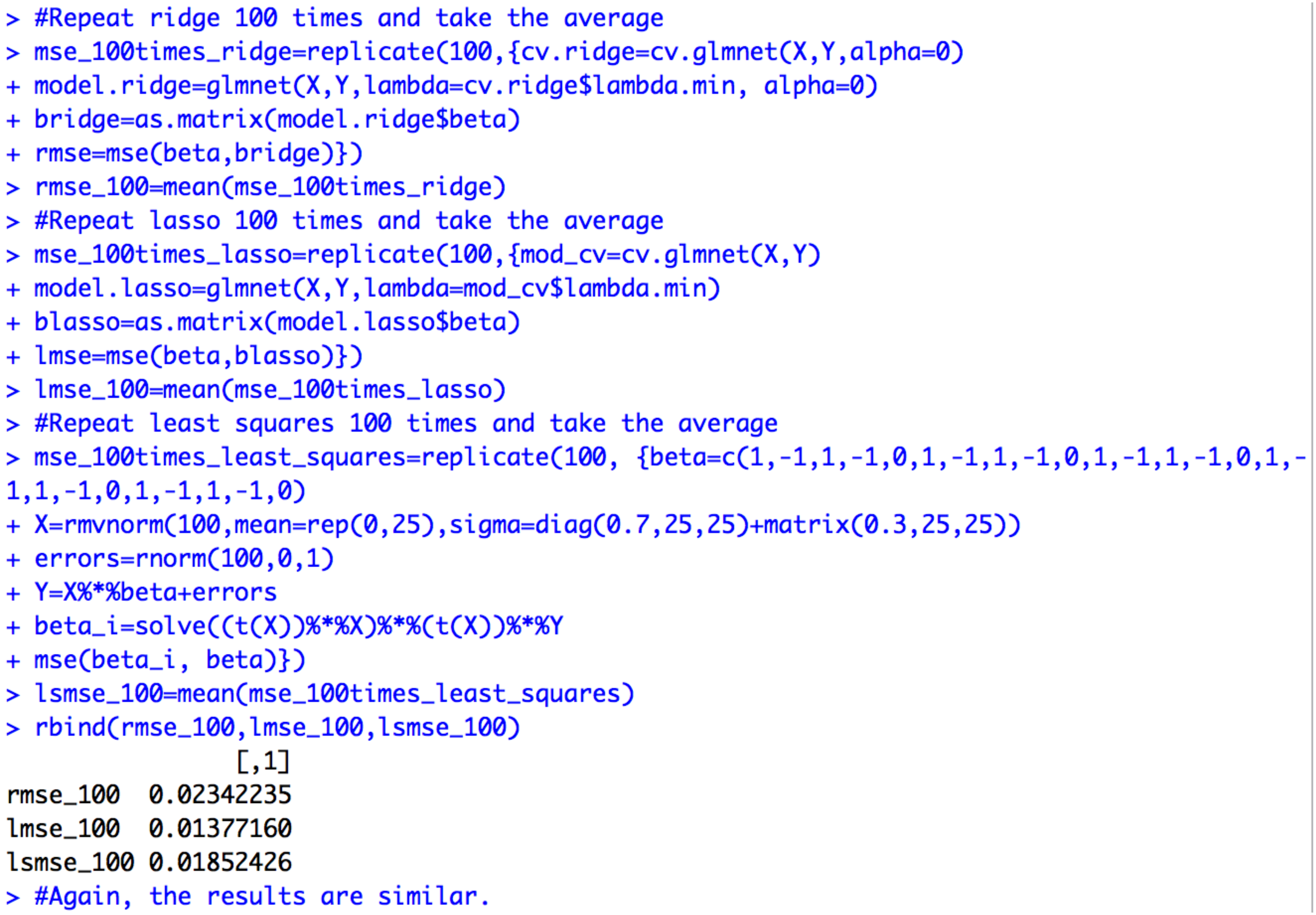
```
> #2(c)
> data(seatpos, package="faraway")
> x=model.matrix(Seated~Age+Weight+Ht+hipcenter+Leg-1, data=seatpos)
> y=seatpos$Seated
> bestmods=leaps(x,y,nbest=1)
> bestmods
$which
      1     2    3     4     5
1 FALSE FALSE TRUE FALSE FALSE
2  TRUE FALSE TRUE FALSE FALSE
3  TRUE FALSE TRUE FALSE  TRUE
4  TRUE  TRUE TRUE FALSE  TRUE
5  TRUE  TRUE TRUE  TRUE  TRUE

$label
[1] "(Intercept)" "1"           "2"           "3"           "4"
[6] "5"

$size
[1] 2 3 4 5 6

$Cp
[1] 1.852984 1.896690 2.613524 4.079995 6.000000

> Cpplot(bestmods)
> #From the graph, we can see the model with the best number of parameters is two, it only
contains the variable weight, since it's closer to the line p and below it. The one with 6
parameter is also possible but we don't choose it because it has too many variables.
>
>
>
>
>
>
>
> 
```



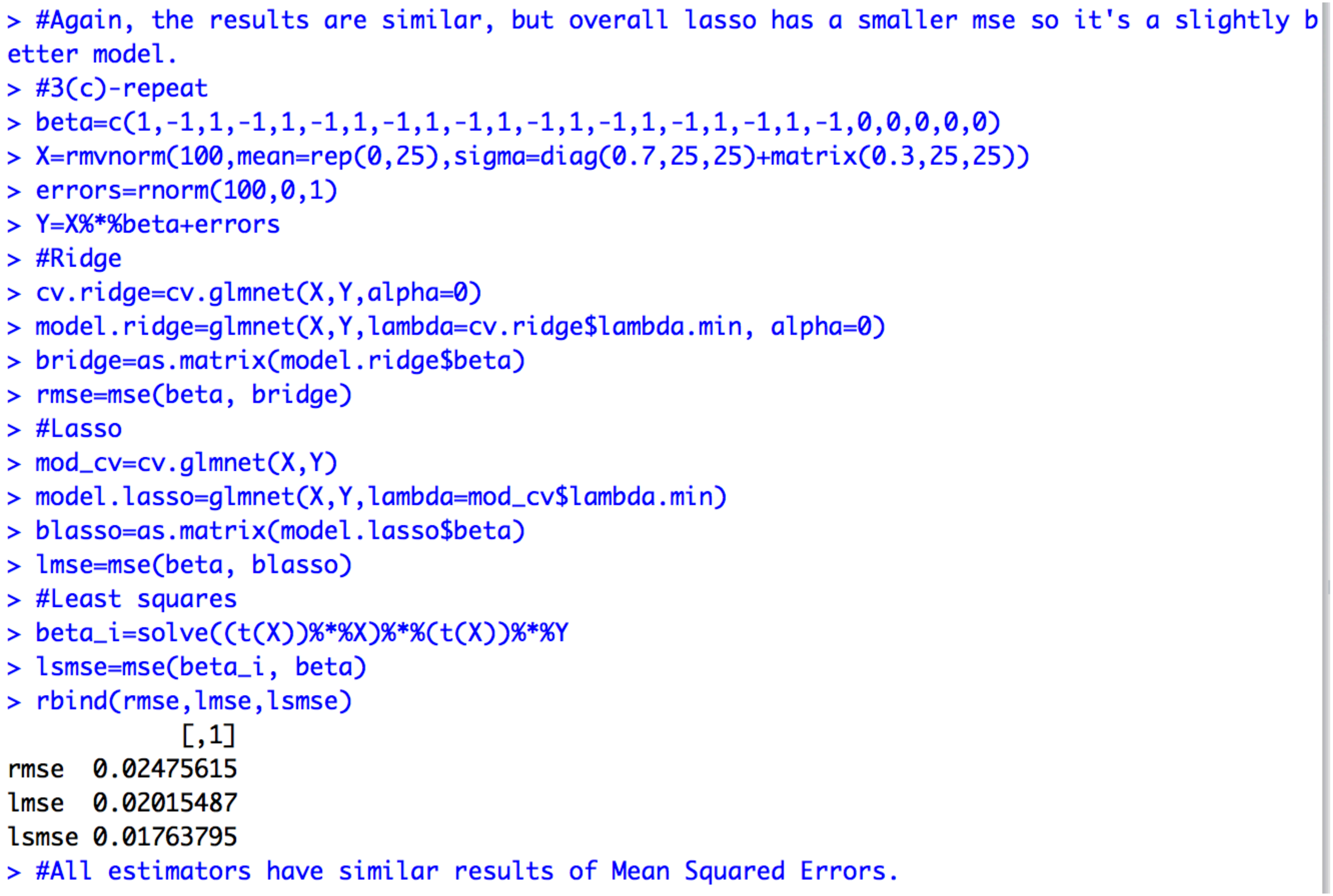
```
> #2(d)
> data(seatpos, package="faraway")
> X=as.matrix(seatpos[,c(1,2,4,8,9)])
> Y=seatpos[,5]
> mod_cv=cv.glmnet(X,Y)
> model.lasso=glmnet(X,Y,lambda=mod_cv$lambda.min)
> blasso=model.lasso$beta
> blasso
5 x 1 sparse Matrix of class "dgCMatrix"
                    s0
Age        -0.022086496
Weight      0.005602745
Ht          0.380163808
Leg         .
hipcenter   .
> #We pick Age, Weight and Ht in the Lasso model.
```



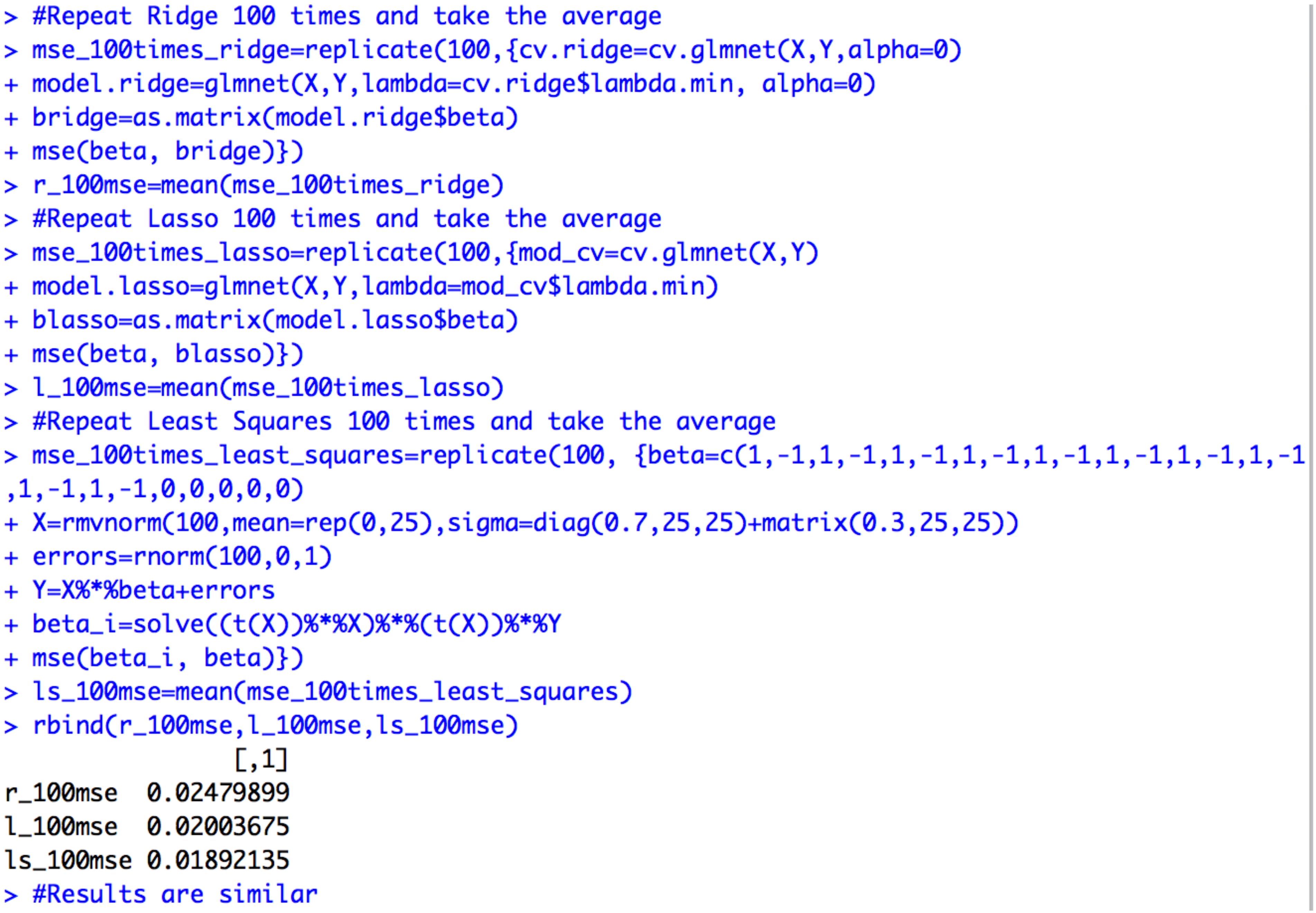
```
> #3(a)
> beta=c(1,-1,1,-1,0,1,-1,1,-1,0,1,-1,1,-1,0,1,-1,1,-1,0,1,-1,1,-1,0)
> X=rmvnorm(100,mean=rep(0,25),sigma=diag(0.7,25,25)+matrix(0.3,25,25))
> #3(b)
> errors=rnorm(100,0,1)
> Y=X%*%beta+errors
> #3(c)
> #Ridge
> cv.ridge=cv.glmnet(X,Y,alpha=0)
> model.ridge=glmnet(X,Y,lambda=cv.ridge$lambda.min, alpha=0)
> bridge=as.matrix(model.ridge$beta)
> rmse=mse(beta,bridge)
> #Lasso
> mod_cv=cv.glmnet(X,Y)
> model.lasso=glmnet(X,Y,lambda=mod_cv$lambda.min)
> blasso=as.matrix(model.lasso$beta)
> lmse=mse(beta,blasso)
> #Least Squares
> beta_i=solve((t(X))%*%X)%*%(t(X))%*%Y
> lsmse=mse(beta_i, beta)
> rbind(rmse,lmse,lsmse)
            [,1]
rmse  0.02330827
lmse  0.01319720
lsmse 0.01315594
> #As shown by the results, the mse for the beta values for each estimators are similar.
```



```
> #Repeat ridge 100 times and take the average
> mse_100times_ridge=replicate(100,{cv.ridge=cv.glmnet(X,Y,alpha=0)
+ model.ridge=glmnet(X,Y,lambda=cv.ridge$lambda.min, alpha=0)
+ bridge=as.matrix(model.ridge$beta)
+ rmse=mse(beta,bridge)})
> rmse_100=mean(mse_100times_ridge)
> #Repeat lasso 100 times and take the average
> mse_100times_lasso=replicate(100,{mod_cv=cv.glmnet(X,Y)
+ model.lasso=glmnet(X,Y,lambda=mod_cv$lambda.min)
+ blasso=as.matrix(model.lasso$beta)
+ lmse=mse(beta,blasso)})
> lmse_100=mean(mse_100times_lasso)
> #Repeat least squares 100 times and take the average
> mse_100times_least_squares=replicate(100, {beta=c(1,-1,1,-1,0,1,-1,1,-1,0,1,-1,1,-1,0,1,-
1,1,-1,0,1,-1,1,-1,0)
+ X=rmvnorm(100,mean=rep(0,25),sigma=diag(0.7,25,25)+matrix(0.3,25,25))
+ errors=rnorm(100,0,1)
+ Y=X%*%beta+errors
+ beta_i=solve((t(X))%*%X)%*%(t(X))%*%Y
+ mse(beta_i, beta)})
> lsmse_100=mean(mse_100times_least_squares)
> rbind(rmse_100,lmse_100,lsmse_100)
                [,1]
rmse_100  0.02342235
lmse_100  0.01377160
lsmse_100 0.01852426
> #Again, the results are similar.
```



```
> #Again, the results are similar, but overall lasso has a smaller mse so it's a slightly b
etter model.
> #3(c)-repeat
> beta=c(1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,0,0,0,0,0)
> X=rmvnorm(100,mean=rep(0,25),sigma=diag(0.7,25,25)+matrix(0.3,25,25))
> errors=rnorm(100,0,1)
> Y=X%*%beta+errors
> #Ridge
> cv.ridge=cv.glmnet(X,Y,alpha=0)
> model.ridge=glmnet(X,Y,lambda=cv.ridge$lambda.min, alpha=0)
> bridge=as.matrix(model.ridge$beta)
> rmse=mse(beta, bridge)
> #Lasso
> mod_cv=cv.glmnet(X,Y)
> model.lasso=glmnet(X,Y,lambda=mod_cv$lambda.min)
> blasso=as.matrix(model.lasso$beta)
> lmse=mse(beta, blasso)
> #Least squares
> beta_i=solve((t(X))%*%X)%*%(t(X))%*%Y
> lsmse=mse(beta_i, beta)
> rbind(rmse,lmse,lsmse)
            [,1]
rmse  0.02475615
lmse  0.02015487
lsmse 0.01763795
> #All estimators have similar results of Mean Squared Errors.
```



```
> #Repeat Ridge 100 times and take the average
> mse_100times_ridge=replicate(100,{cv.ridge=cv.glmnet(X,Y,alpha=0)
+ model.ridge=glmnet(X,Y,lambda=cv.ridge$lambda.min, alpha=0)
+ bridge=as.matrix(model.ridge$beta)
+ mse(beta, bridge)})
> r_100mse=mean(mse_100times_ridge)
> #Repeat Lasso 100 times and take the average
> mse_100times_lasso=replicate(100,{mod_cv=cv.glmnet(X,Y)
+ model.lasso=glmnet(X,Y,lambda=mod_cv$lambda.min)
+ blasso=as.matrix(model.lasso$beta)
+ mse(beta, blasso)})
> l_100mse=mean(mse_100times_lasso)
> #Repeat Least Squares 100 times and take the average
> mse_100times_least_squares=replicate(100, {beta=c(1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1
,1,-1,1,-1,0,0,0,0,0)
+ X=rmvnorm(100,mean=rep(0,25),sigma=diag(0.7,25,25)+matrix(0.3,25,25))
+ errors=rnorm(100,0,1)
+ Y=X%*%beta+errors
+ beta_i=solve((t(X))%*%X)%*%(t(X))%*%Y
+ mse(beta_i, beta)})
> ls_100mse=mean(mse_100times_least_squares)
> rbind(r_100mse,l_100mse,ls_100mse)
                [,1]
r_100mse  0.02479899
l_100mse  0.02003675
ls_100mse 0.01892135
> #Results are similar
```