Homework 6

Sarah Adilijiang

Problem 1

(a) Pairwise Correlations

```
library(faraway)
data(seatpos)
model_ls = lm(hipcenter~., seatpos)
summary(model_ls)
```

```
##
## Call:
## lm(formula = hipcenter ~ ., data = seatpos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -73.827 -22.833  -3.678  25.017  62.337
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 436.43213  166.57162   2.620   0.0138 *
## Age           0.77572    0.57033   1.360   0.1843
## Weight        0.02631    0.33097   0.080   0.9372
## HtShoes      -2.69241    9.75304  -0.276   0.7845
## Ht            0.60134   10.12987   0.059   0.9531
## Seated        0.53375    3.76189   0.142   0.8882
## Arm          -1.32807    3.90020  -0.341   0.7359
## Thigh        -1.14312    2.66002  -0.430   0.6706
## Leg          -6.43905    4.71386  -1.366   0.1824
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.72 on 29 degrees of freedom
## Multiple R-squared:  0.6866, Adjusted R-squared:  0.6001
## F-statistic:  7.94 on 8 and 29 DF,  p-value: 1.306e-05
```

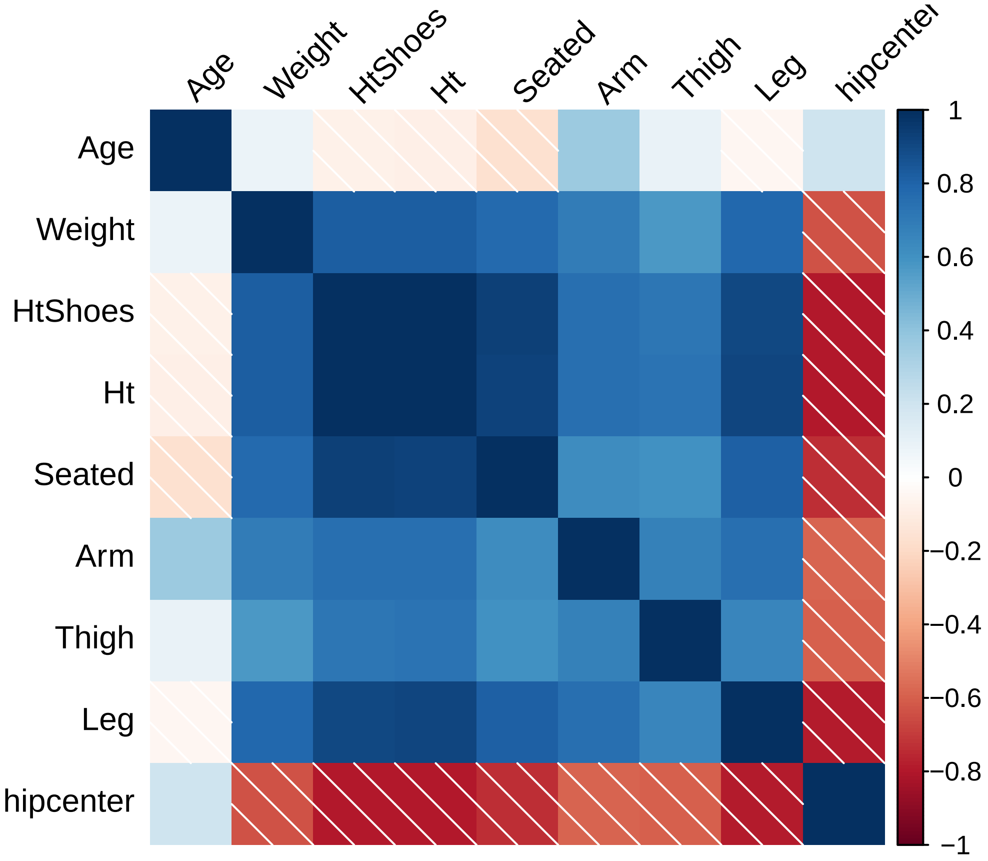
```
# pairwise correlations
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
pair_cor = round(cor(seatpos),3); pair_cor
```

```
##            Age Weight HtShoes     Ht Seated   Arm  Thigh    Leg
## Age      1.000  0.081  -0.079 -0.090 -0.170 0.360  0.091 -0.042
## Weight   0.081  1.000   0.828  0.829  0.776 0.698  0.573  0.784
## HtShoes -0.079  0.828   1.000  0.998  0.930 0.752  0.725  0.908
## Ht      -0.090  0.829   0.998  1.000  0.928 0.752  0.735  0.910
## Seated  -0.170  0.776   0.930  0.928  1.000 0.625  0.607  0.812
## Arm      0.360  0.698   0.752  0.752  0.625 1.000  0.671  0.754
## Thigh    0.091  0.573   0.725  0.735  0.607 0.671  1.000  0.650
## Leg     -0.042  0.784   0.908  0.910  0.812 0.754  0.650  1.000
## hipcenter  0.205 -0.640  -0.797 -0.799 -0.731 -0.585 -0.591 -0.787
##           hipcenter
## Age           0.205
## Weight       -0.640
## HtShoes      -0.797
## Ht           -0.799
## Seated       -0.731
## Arm          -0.585
## Thigh        -0.591
## Leg          -0.787
## hipcenter     1.000
```

```
corrplot(pair_cor, method="shade", tl.col="black",tl.srt=45)
```



Answer:

There are several large pairwise correlations both between covariates and between covariates and the response, which indicates multicollinearity problem of the dataset. Especially, the covariates "HtShoes", "Ht", and "Seated" are highly correlated with each other.

Highly collinearity will lead to imprecise estimate of coefficients, inflate the variance of the coefficients, and fail to reveal significant factors via t-tests. We can see in the model summary, the p-value of F-test is quite small but none of the individual covariates is significant.

Therefore, it's better to keep only one of highly correlated predictors in the model to keep the model simple and reduce the inflated variance.

(b) Standardization of covariates

```
# standardization (z-score normalization) of covariates
n = nrow(seatpos)
seatpos2 = scale(seatpos[-9], center = colMeans(seatpos[-9]), scale=FALSE)  # or center=TRUE
seatpos2 = scale(seatpos2, center=FALSE,
                 scale = sqrt(colSums(seatpos2^2)/n) ) # if use scale=TRUE, it's dividing by (n-1)
seatpos3 = cbind(data.frame(seatpos2), seatpos$hipcenter)
colnames(seatpos3)[9] = "hipcenter"

# Check the standardization results
colMeans(seatpos3)
```

```
##           Age        Weight       HtShoes            Ht        Seated
## -1.789504e-17  2.934422e-16  9.659671e-16  1.994475e-16 -1.064938e-15
##           Arm         Thigh           Leg     hipcenter
## -1.124831e-16  1.044486e-16 -1.209741e-16 -1.648849e+02
```

```
colSums(seatpos3^2)
```

```
##       Age    Weight   HtShoes        Ht    Seated       Arm     Thigh
##        38        38        38        38        38        38        38
##       Leg hipcenter
##        38   1164746
```

Answer:

According to the results check, we have standardized each covariate to have zero mean and squared norm of n=38. And the response “hipcenter” was not standardized.

(c) Ridge regression

```
# ridge regression
library(MASS)
model_ridge = lm.ridge(hipcenter~., seatpos3, lambda = c(0,0.1,1,2,5,10,20,50))
betahat = coef(model_ridge); betahat
```

```
##                        Age     Weight    HtShoes        Ht     Seated
##  0.0 -164.8849 11.763894  0.9290423 -29.618082  6.630015  2.5974846
##  0.1 -164.8849 11.501678  0.9733863 -18.375451 -4.775794  2.2699896
##  1.0 -164.8849 11.209797  0.3796354 -11.749208 -9.785972  0.3649019
##  2.0 -164.8849 10.970371 -0.1942487 -10.701957 -9.759135 -0.9677302
##  5.0 -164.8849 10.237244 -1.3963694  -9.629934 -9.327971 -3.1470988
## 10.0 -164.8849  9.171969 -2.5307142  -8.974283 -8.878673 -4.7001831
## 20.0 -164.8849  7.612264 -3.5585050  -8.304574 -8.296598 -5.7236525
## 50.0 -164.8849  5.089632 -4.2130982  -7.115605 -7.138796 -5.8725225
##            Arm     Thigh        Leg
##  0.0 -4.418228 -4.370897 -21.626207
##  0.1 -4.354180 -4.120937 -21.397236
##  1.0 -4.640491 -4.293360 -20.059262
##  2.0 -4.822986 -4.495120 -18.783753
##  5.0 -4.940371 -4.809908 -16.052201
## 10.0 -4.819280 -4.992430 -13.486776
## 20.0 -4.571745 -5.015310 -10.990252
## 50.0 -4.169567 -4.652486  -8.169737
```

Answer:

Without ridge regularization ($\lambda = 0$), the range of estimated coefficients (not inlcuding intercept) is large:

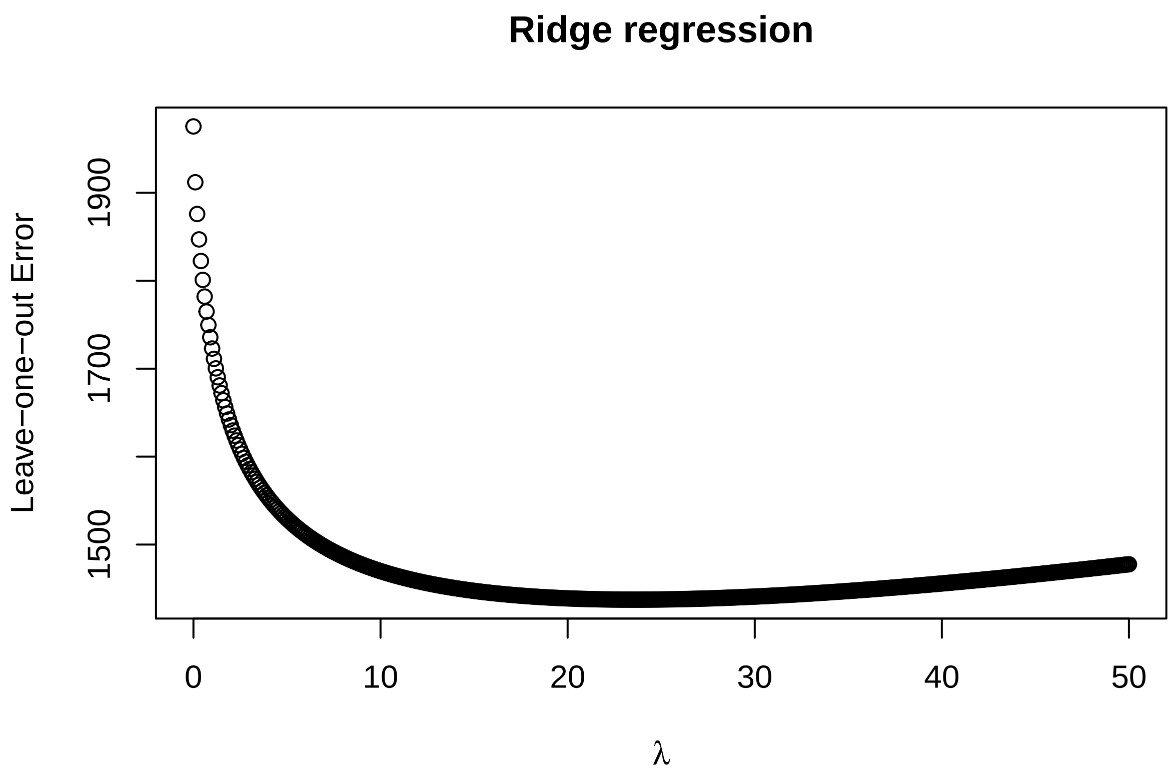
$abs(\hat{\beta}_j) \in (0.93, 29.62)$.

With ridge regularization, range of estimated coefficients start to shrink, i.e. the size of coefficients are becoming closer to each other, and the larger the coefficient is, the faster it will shrink. With λ becoming larger, the smaller the range is and the smaller the L2-norm of coefficient vector is. At largest $\lambda = 50$, the range of estimated coefficients (not inlcuding intercept) is the smallest: $abs(\hat{\beta}_j) \in (4.17, 8.17)$.

In particular, the $abs(\hat{\beta}_j)$ of covariates "HtShoes", "Ht", and "Seated" are in the range of (2.60, 29.62) without ridge regularization, and becomes spread nearly equally within the range of (5.87, 7.14) when $\lambda = 50$. The coefficient of "HtShoes" shrinks very fast while the coefficients of "Ht" and "Seated" change from positive values to negative values and increase their sizes to close to coefficient of "HtShoes". Therefore, for highly correlated covariates, ridge regression will prefer the correlated coefficients to be spread equally when the L2-norm of coefficients are becoming smaller.

(d) Leave-one-out cross-validation of Ridge regression

```r
# leave-one-out cross-validation of ridge regression
n = nrow(seatpos3)
lambdas = seq(0,50,by=0.1)
ave_square_error = rep(0,length(lambdas))

for (k in 1:length(lambdas)){
    store_pred_error = rep(0,n)
    for (i in 1:n) {
        model_ridge = lm.ridge(hipcenter~., seatpos3[-i,], lambda=lambdas[k])
        betahat = coef(model_ridge)
        xi = seatpos3[i,-9]
        fitted_yi = unlist(c(1,xi)) %*% as.vector(betahat)
        store_pred_error[i] = seatpos3[i,9] - fitted_yi
    }
    ave_square_error[k] = sum(store_pred_error^2)/n
}

# plot leave-one-out error against lambda
plot(lambdas, ave_square_error, main="Ridge regression",
     xlab=expression(lambda), ylab="Leave-one-out Error")
```



```
# find the best lambda value
best_lambda_ridge = lambdas[which.min(ave_square_error)]
print(paste0("best lambda = ",best_lambda_ridge))
```

```
## [1] "best lambda = 23.6"
```

Answer:

According to the plots, as λ increases from 0 to 23.6, the average of squared leave-one-out prediction error substantially decreases, especially within the region of $\lambda \in (0, 10)$. Then, as λ increases from 23.6 to 50, the average of squared leave-one-out prediction error slowly increases again.

Therefore, ridge regularization dose offer substantial improvement of the prediction error. And here it reduces the prediction error to the most extent at $\lambda = 23.6$.

```
# fit the ridge regression model with the best lambda value and get the coefficients
model_ridge = lm.ridge(hipcenter~., seatpos3, lambda = best_lambda_ridge)
coef(model_ridge)
```

```
##                     Age     Weight    HtShoes         Ht     Seated
## -164.884868    7.179574  -3.750094  -8.123357  -8.125969  -5.856485
##         Arm       Thigh        Leg
##   -4.506207   -4.986708 -10.439293
```

```
# compare with coefficients of least squares model (after standardizing covariates)
model_ls = lm(hipcenter~., seatpos3)
coef(model_ls)
```

```
##  (Intercept)          Age       Weight      HtShoes           Ht
## -164.8848684   11.7638935    0.9290423  -29.6180818    6.6300155
##       Seated          Arm        Thigh           Leg
##    2.5974846   -4.4182282   -4.3708966  -21.6262073
```

Answer:

The intercept is not penalized thus not changed. As for the other coefficients, the range of estimated coefficients in the least squares model (after standardizing covariates) is large: $abs(\hat{\beta_j}) \in (0.93, 29.62)$. And the range of estimated coefficients in the ridge regression model with best lambda value (after standardizing covariates) is smaller: $abs(\hat{\beta_j}) \in (3.75, 10.44)$.

In particular, the coefficient of "HtShoes" has an opposite sign with and its size is much larger than coefficients of "Ht" and "Seated" in the least squares model, but the size of "HtShoes" shrinks fast and finally becomes nearly equal to coefficients of "Ht" and "Seated" in the ridge model. And the coefficients of "Ht" and "Seated" change from positive values to negative values and increase their sizes to close to coefficient of "HtShoes" in the ridge model. Therefore, for highly correlated covariates, ridge regression will prefer the correlated coefficients to be spread equally when the L2-norm of coefficients are becoming smaller.

(e) Lasso regression

```
# ridge regression
library(glmnet)
```

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

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-16
```

```
model_lasso = glmnet(x = as.matrix(seatpos3[-9]), y = as.matrix(seatpos3[9]),
                     lambda = c(0,0.1,1,2,5,10,20,50))
betahat = rbind(model_lasso$a0, as.matrix(model_lasso$beta))
colnames(betahat) = model_lasso$lambda
rownames(betahat)[1] = "(Intercept)"
betahat
```

```
##                    50         20         10           5           2
## (Intercept) -164.8849 -164.88487 -164.88487 -164.884868 -164.884868
## Age            0.0000    0.00000    0.00000    4.068237    7.471686
## Weight         0.0000    0.00000    0.00000    0.000000    0.000000
## HtShoes        0.0000    0.00000    0.00000    0.000000   -6.814971
## Ht             0.0000  -17.80373  -23.03997  -24.501311  -15.739548
## Seated         0.0000    0.00000    0.00000    0.000000    0.000000
## Arm            0.0000    0.00000    0.00000    0.000000    0.000000
## Thigh          0.0000    0.00000    0.00000    0.000000   -2.946868
## Leg            0.0000  -10.13364  -15.36996  -18.868288  -21.590190
##                       1          0.1            0
## (Intercept) -164.884868 -164.8848684 -164.8848684
## Age            9.514490   11.4099393   11.6563177
## Weight         0.000000    0.4934133    0.9687952
## HtShoes      -14.643985  -21.1339220  -24.2208321
## Ht            -6.302104    0.0000000    1.2230348
## Seated         0.000000    0.9783011    2.4749935
## Arm           -2.060838   -4.1919797   -4.3830743
## Thigh         -3.851717   -4.3683493   -4.2584959
## Leg          -21.836091  -21.7327331  -21.6460224
```

Answer:

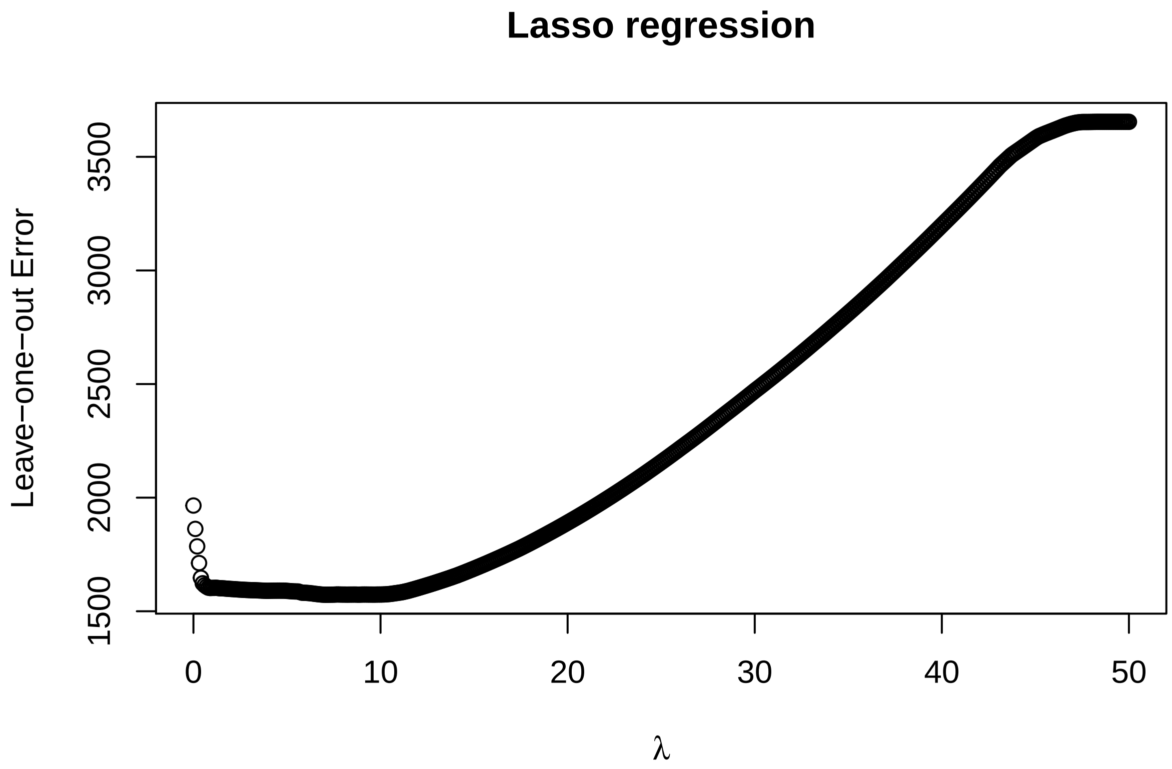
Without lasso regularization ($\lambda = 0$), the range of estimated coefficients (not inlcuding intercept) is large: $abs(\hat{\beta}_j) \in (0.97, 24.22)$ and are all nonzeros.

With lasso regularization, most of estimated coefficients start to shrink and are finally forced to zeros when keep increasing the λ value. With λ becoming larger, the lower number of coefficients are kept in the model (fewer nonzeros), and the smaller the L1-norm of coefficient vector is. At largest $\lambda = 50$, only the intercept is left in the model, all the other coefficients are forced to zeros.

In particular, the $abs(\hat{\beta}_j)$ of covariates "HtShoes" and "Seated" shrink more as λ becomes larger, and finally shrink to zeros and do not change any more when keep increasing λ. However, the coefficient of "Ht" first shrink to zero and then change the sign to negative values and starts to increase its size as λ becomes larger. When coefficients of "HtShoes" and "Seated" have become zeros after $\lambda = 5$, the coefficient of "Ht" becomes the only one left in the model among these three highly correlated covariates. Therefore, for highly correlated covariates, lasso regression will prefer sparse coefficients, giving all the weights to a single one of correlated coefficients and keeping only that one in the model.

(f) Leave-one-out cross-validation of Lasso regression

```
# leave-one-out cross-validation of lasso regression
n = nrow(seatpos3)
lambdas = seq(0,50,by=0.1)
ave_square_error = rep(0,length(lambdas))

for (k in 1:length(lambdas)){
    store_pred_error = rep(0,n)
    for (i in 1:n) {
        model_lasso = glmnet(x = as.matrix(seatpos3[-i,-9]), y = as.matrix(seatpos3[-i,9]),
                               lambda=lambdas[k])
        betahat = rbind(model_lasso$a0, as.matrix(model_lasso$beta))
        xi = seatpos3[i,-9]
        fitted_yi = unlist(c(1,xi)) %*% as.vector(betahat)
        store_pred_error[i] = seatpos3[i,9] - fitted_yi
    }
    ave_square_error[k] = sum(store_pred_error^2)/n
}

# plot leave-one-out error against lambda
plot(lambdas, ave_square_error, main="Lasso regression",
     xlab=expression(lambda), ylab="Leave-one-out Error")
```



```r
# find the best lambda value
best_lambda_lasso = lambdas[which.min(ave_square_error)]
print(paste0("best lambda = ",best_lambda_lasso))
```

```
## [1] "best lambda = 7.1"
```

Answer:

According to the plots, as λ increases from 0 to 7.1, the average of squared leave-one-out prediction error substantially decreases, especially within the region of $\lambda \in (0, 1)$. Then, as λ increases from 7.1 to 50, the average of squared leave-one-out prediction error substantially increases again and reaches a plateau near 50.

Therefore, lasso regularization dose offer substantial improvement of the prediction error. And here it reduces the prediction error to the most extent at $\lambda = 7.1$.

```r
# fit the lasso regression model with the best lambda value and get the coefficients
model_lasso = glmnet(x = as.matrix(seatpos3[-9]), y = as.matrix(seatpos3[9]),
                     lambda = best_lambda_lasso)
betahat = rbind(model_lasso$a0, as.matrix(model_lasso$beta))
colnames(betahat) = model_lasso$lambda
rownames(betahat)[1] = "(Intercept)"
betahat
```

```
##                    7.1
## (Intercept) -164.884868
## Age            2.083547
## Weight         0.000000
## HtShoes        0.000000
## Ht           -23.933743
## Seated         0.000000
## Arm            0.000000
## Thigh          0.000000
## Leg          -17.368648
```

```
# compare with coefficients of least squares model (after standardizing covariates)
model_ls = lm(hipcenter~., seatpos3)
coef(model_ls)
```

```
##  (Intercept)          Age       Weight      HtShoes           Ht
## -164.8848684   11.7638935    0.9290423  -29.6180818    6.6300155
##       Seated          Arm        Thigh          Leg
##    2.5974846   -4.4182282   -4.3708966  -21.6262073
```

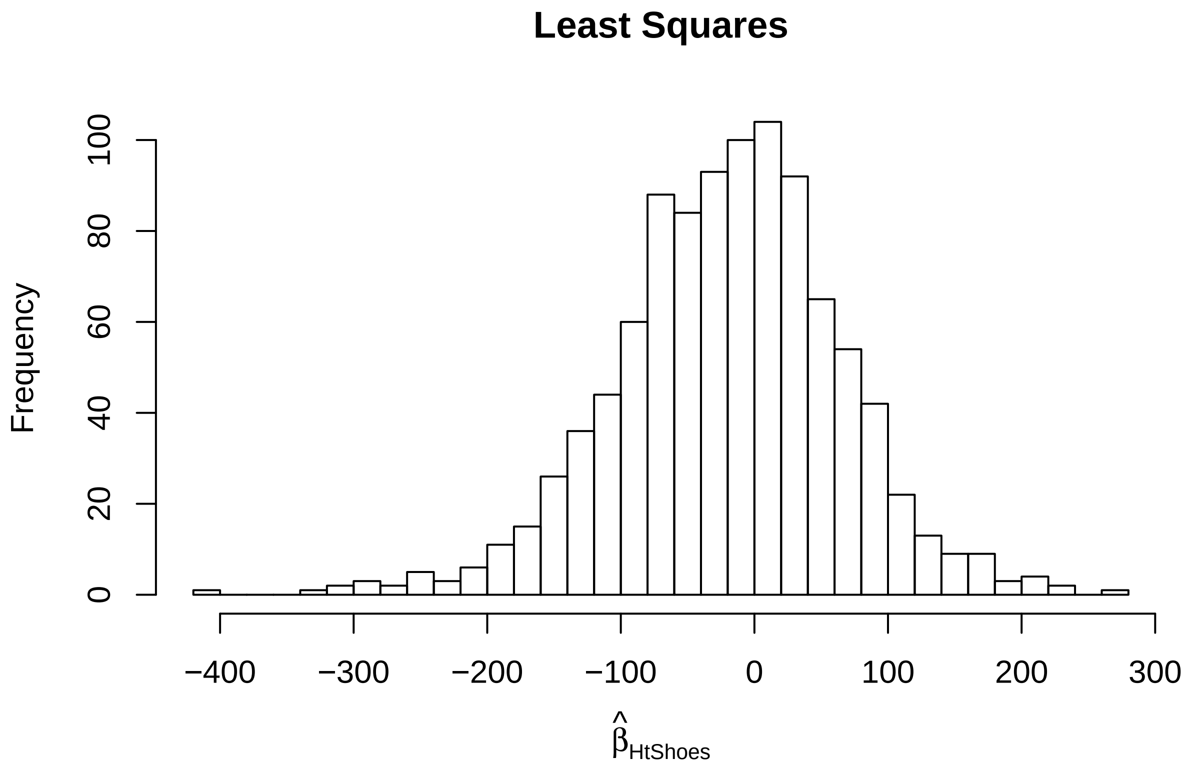
Answer:

The intercept is not penalized thus not changed. As for the other coefficients, the estimated coefficients of the least squares model (after standardizing covariates) are all nonzeros. However, in the lasso regression model (after standardizing covariates) , most of the estimated coefficients are forced to zeros. Only the coefficients of "Age", "Ht", and "Leg" are nonzeros, and the coefficients of "Age" and "Leg" have shrunk the size comparing with the least squares model, while the coefficient of "Ht" has changed the sign and increased its size.

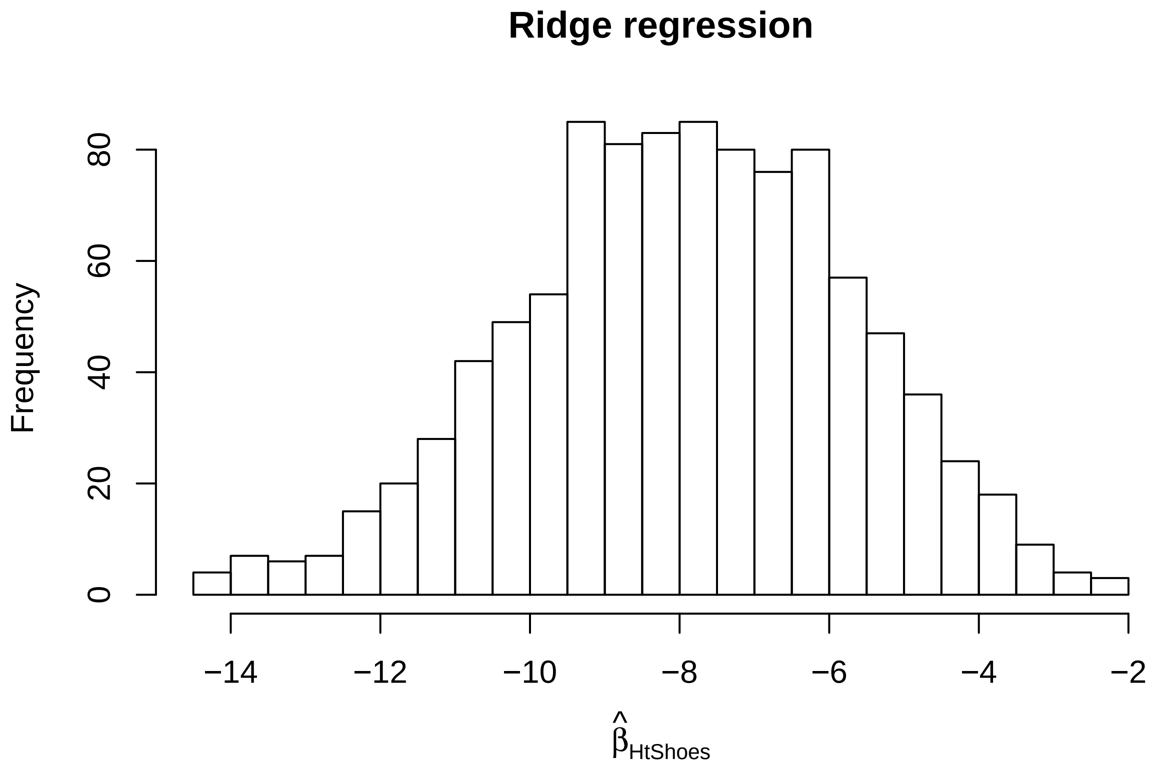
In particular, only one of covariates "Ht", "HtShoes" and "Seated" is left in the lasso regression model, which is "Ht". Therefore, for highly correlated covariates, lasso regression will prefer sparse coefficients, giving all the weights to a single one of correlated coefficients and keeping only that one in the model.

(g) Bootstrap

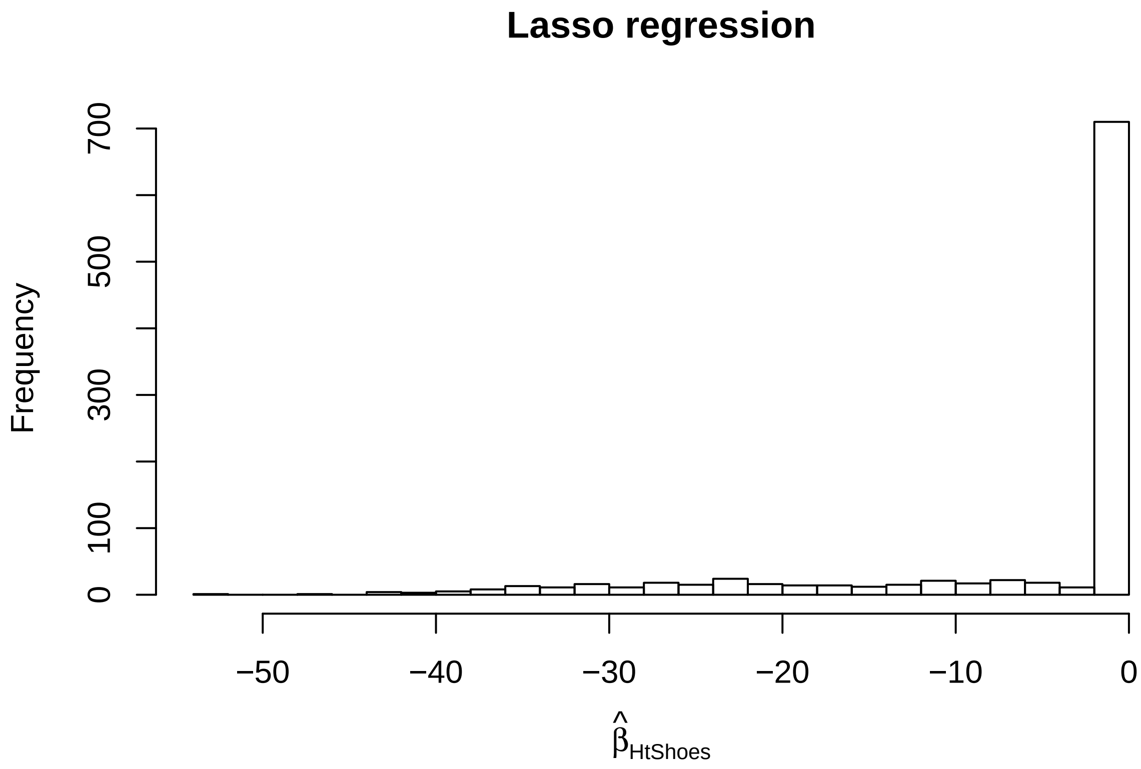
```
# bootstrap data 1000 times
set.seed(0)
n = nrow(seatpos3)
betahat_ls = matrix(rep(0,9000), nrow=9, ncol=1000)
betahat_ridge = matrix(rep(0,9000), nrow=9, ncol=1000)
betahat_lasso = matrix(rep(0,9000), nrow=9, ncol=1000)

for (i in 1:1000){
    indices = sample(1:n, n, replace = TRUE)
    boot_data = seatpos3[indices, ]

    # least squares model
    model_ls = lm(hipcenter~., boot_data)
    betahat_ls[,i] = coef(model_ls)

    # Ridge regression model
    model_ridge = lm.ridge(hipcenter~., boot_data, lambda = best_lambda_ridge)
    betahat_ridge[,i] = coef(model_ridge)

    # Lasso regression model
    model_lasso = glmnet(x = as.matrix(boot_data[-9]), y = as.matrix(boot_data[9]),
                         lambda = best_lambda_lasso)
    betahat_lasso[,i] = rbind(model_lasso$a0, as.matrix(model_lasso$beta))
}

hist(betahat_ls[4,], main="Least Squares", xlab=expression(hat(beta)[HtShoes]), breaks=30)
```



```
hist(betahat_ridge[4,], main="Ridge regression", xlab=expression(hat(beta)[HtShoes]), breaks=30)
```



```
hist(betahat_lasso[4,], main="Lasso regression", xlab=expression(hat(beta)[HtShoes]), breaks=30)
```



Answer:

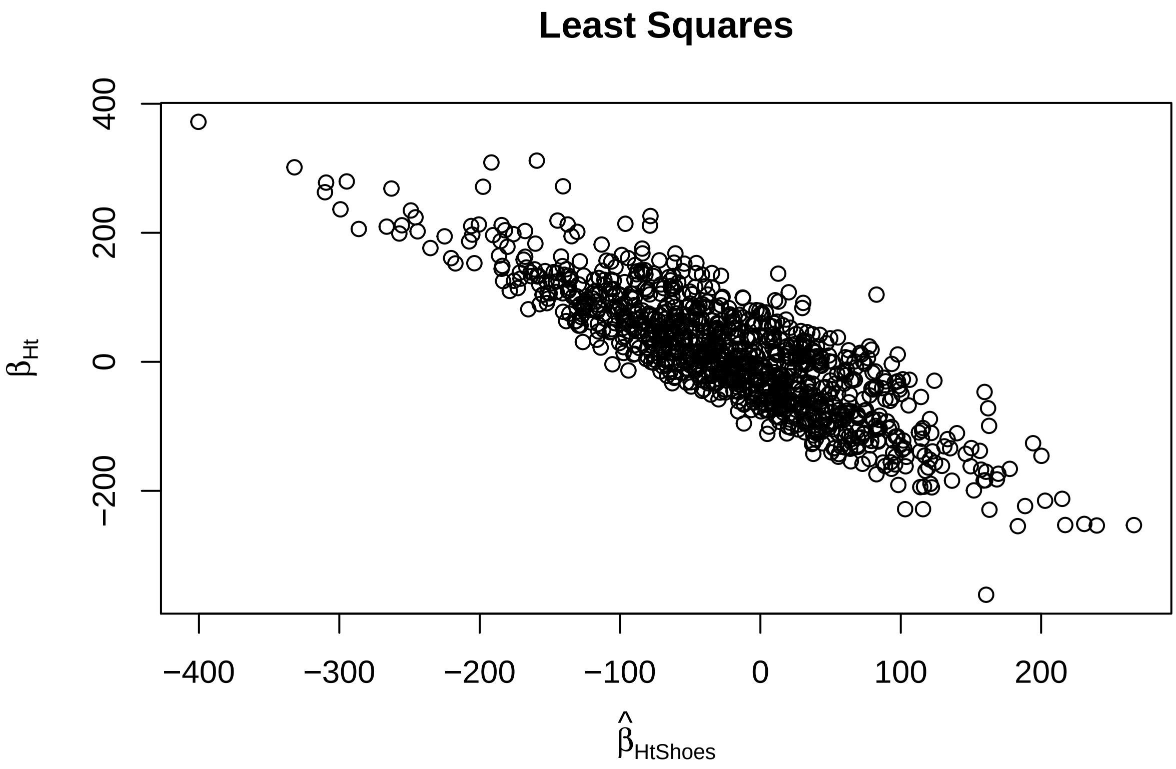
(1) In the least squares model, the range of $\hat{\beta}_{HtShoes}$ is large, nearly from -400 to 300, and the majority fall in the range of (-80, 40). This indicates that when the design matrix is collinear the least squares estimates of coefficients will be unstable and have inflated variance.

(2) In the ridge regression model, the range of $\hat{\beta}_{HtShoes}$ is much smaller, nearly from -14 to -2, and the majority fall in the range of (-9.5, -6). This indicates that ridge regularization will shrink the size of coefficients and reduce the variance of coefficients (at the price of increasing the bias).

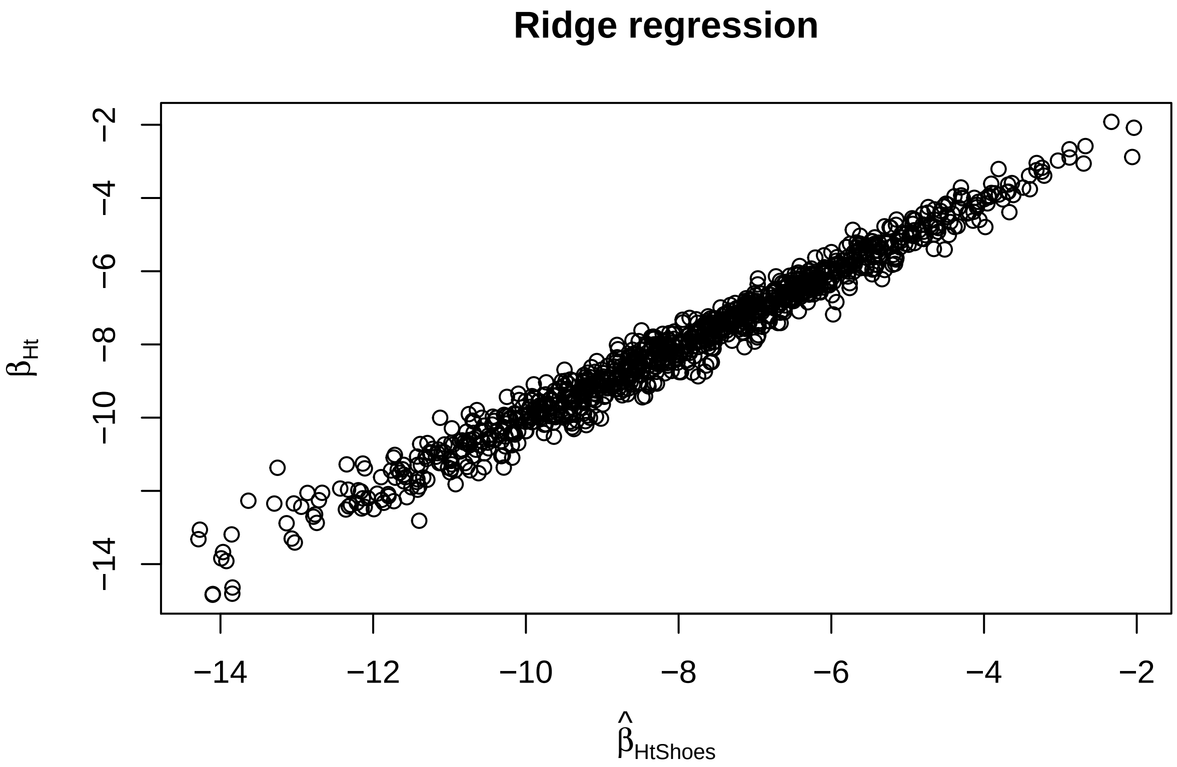
(3) In the lasso regression model, the range of $\hat{\beta}_{HtShoes}$ is much smaller than that of least squares model but larger than the ridge regression model, nearly from -50 to 0. However, the distribution of $\hat{\beta}_{HtShoes}$ is highly skewed and mostly forced to zeros. This indicates that lasso regularization will also shrink the size of coefficients and reduce the variance of coefficients (at the price of increasing the bias), but it will prefer and reach sparsity of coefficients (i.e. substantially reduce the number of coefficients).

(h) Bootstrap

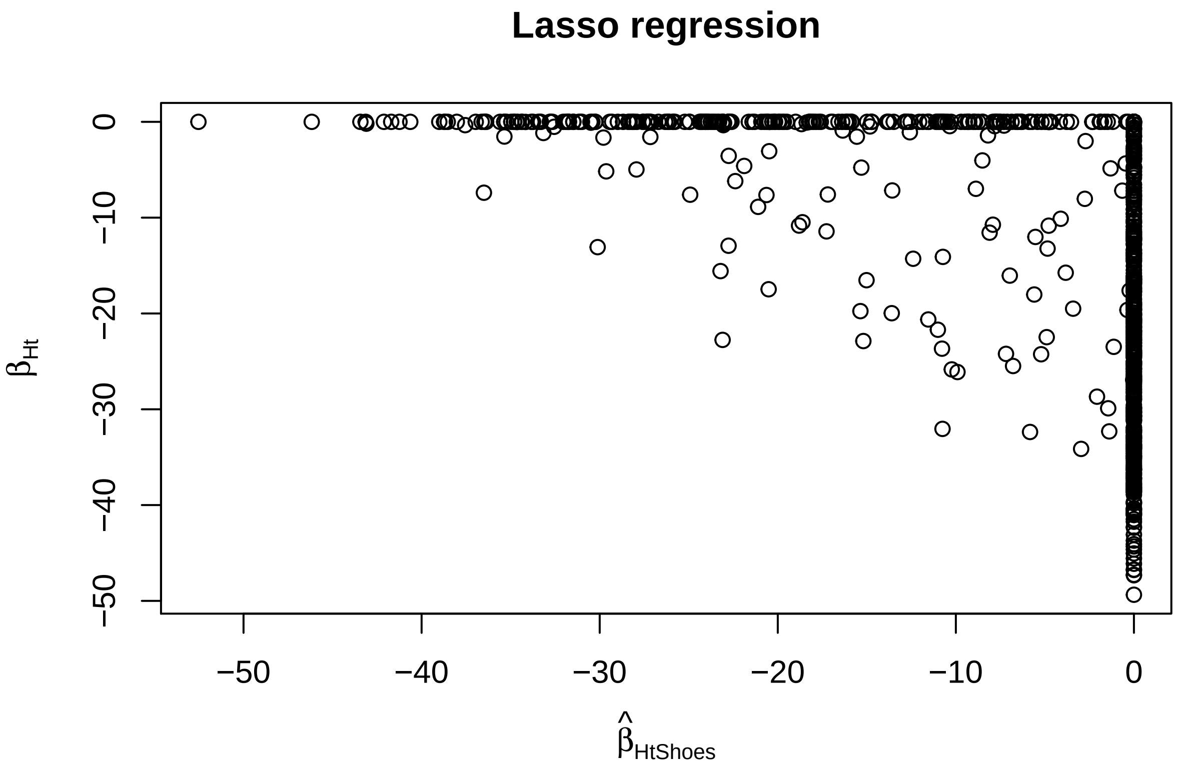
```r
plot(betahat_ls[4,], betahat_ls[5,], main="Least Squares",
     xlab = expression(hat(beta)[HtShoes]), ylab = expression(hat(beta)[Ht]))
```



```
plot(betahat_ridge[4,], betahat_ridge[5,], main="Ridge regression",
     xlab = expression(hat(beta)[HtShoes]), ylab = expression(hat(beta)[Ht]))
```



```r
plot(betahat_lasso[4,], betahat_lasso[5,], main="Lasso regression",
     xlab = expression(hat(beta)[HtShoes]), ylab = expression(hat(beta)[Ht]))
```



Answer:

(1) In the least squares model, both the range of $\hat{\beta}_{HtShoes}$ and $\hat{\beta}_{Ht}$ are similar and large, nearly from -400 to 400. And $\hat{\beta}_{HtShoes}$ and $\hat{\beta}_{Ht}$ are highly negative correlated. This indicates that when the covariates are highly positive correlated, their coefficients will be highly negative correlated in the least squares model and have inflated large variances.

(2) In the ridge regression model, both the range of $\hat{\beta}_{HtShoes}$ and $\hat{\beta}_{Ht}$ are similar and much smaller than those of least squares model, nearly from -14 to -2. And $\hat{\beta}_{HtShoes}$ and $\hat{\beta}_{Ht}$ are even more highly positive correlated. This indicates that when the covariates are highly positive correlated, their coefficients will be also highly positive correlated in the ridge regression model and have reduced variances than the least quares model.

(3) In the lasso regression model, both the range of $\hat{\beta}_{HtShoes}$ and $\hat{\beta}_{Ht}$ are similar and much smaller than that of least squares model but larger than the ridge regression model, nearly from -50 to 0. However, in the most cases, either one and only one of $\hat{\beta}_{HtShoes}$ and $\hat{\beta}_{Ht}$ is forced to zero. And there are some cases where both of them are forced to zeros. This indicates that when the covariates are highly positive correlated, lasso regularization will prefer to keep only one (or none) of them in the model to substantially reduce the number of coefficients. Also, lasso regularization will shrink the size of remaining coefficients in the model and reduce the variance of coefficients.

Problem 2

```r
library(faraway)
data(trees)

# fit a raw second-order polynomial regression including the interaction term
model_raw = lm(log(Volume)~Girth+Height+I(Girth^2)+I(Height^2)+I(Girth*Height), trees)
```

```r
# same with : model = lm(log(Volume)~polym(Girth, Height, degree=2, raw=TRUE), trees)

summary(model_raw)
```

```
##
## Call:
## lm(formula = log(Volume) ~ Girth + Height + I(Girth^2) + I(Height^2) +
##     I(Girth * Height), data = trees)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.159718 -0.041905 -0.003371  0.055167  0.133780
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -1.9660208  2.0066922  -0.980  0.33660
## Girth              0.2808126  0.0786856   3.569  0.00149 **
## Height             0.0484196  0.0567321   0.853  0.40150
## I(Girth^2)        -0.0042410  0.0032183  -1.318  0.19953
## I(Height^2)       -0.0002022  0.0004186  -0.483  0.63326
## I(Girth * Height) -0.0001975  0.0018089  -0.109  0.91395
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08469 on 25 degrees of freedom
## Multiple R-squared:  0.9784, Adjusted R-squared:  0.9741
## F-statistic: 226.7 on 5 and 25 DF,  p-value: < 2.2e-16
```

The summary shows that the regression is significant, however, only one covariate "Girth" is significant in this model.

Then, we use several ways to check whether this model may be reasonably simplified. Notice that we should not remove the interaction term without removing the corresponding second-order terms. So there are six possible smaller models to consider.

(1) Adjusted R-squared

```r
# compare the adjusted-R squares of all smaller models with the full model
model1 = lm(log(Volume)~Girth+Height, trees)
model2 = lm(log(Volume)~Girth+Height+I(Girth^2), trees)
model3 = lm(log(Volume)~Girth+Height+I(Height^2), trees)
model4 = lm(log(Volume)~Girth+Height+I(Girth*Height), trees)
model5 = lm(log(Volume)~Girth+Height+I(Girth^2)+I(Girth*Height), trees)
model6 = lm(log(Volume)~Girth+Height+I(Height^2)+I(Girth*Height), trees)

adj_R_squared = rbind(
    summary(model1)$adj.r.squared,
    summary(model2)$adj.r.squared,
    summary(model3)$adj.r.squared,
    summary(model4)$adj.r.squared,
    summary(model5)$adj.r.squared,
    summary(model6)$adj.r.squared,
    summary(model_raw)$adj.r.squared
); adj_R_squared
```

```
##           [,1]
## [1,] 0.9661964
## [2,] 0.9756755
## [3,] 0.9686219
## [4,] 0.9742928
## [5,] 0.9748647
## [6,] 0.9733674
## [7,] 0.9741010
```

```
which.max(adj_R_squared)
```

```
## [1] 2
```

Answer:

Here the adjusted R-squared has the maximum value for model2: log(Volume) ~ Girth + Height + I(Girth^2), so model2 is preferred in this case.

(2) Forward selection

```
# forward selection using AIC
fit_start = lm(log(Volume)~Girth+Height, trees)
fit_forward_aic = step(fit_start, log(Volume)~Girth+Height+I(Girth^2)+I(Height^2)+I(Girth*Height),
                       direction = "forward")
```

```
## Start:  AIC=-141.96
## log(Volume) ~ Girth + Height
## 
##                       Df Sum of Sq     RSS     AIC
## + I(Girth^2)           1  0.080245 0.18189 -151.29
## + I(Girth * Height)    1  0.069906 0.19223 -149.57
## + I(Height^2)          1  0.027500 0.23464 -143.39
## <none>                             0.26214 -141.96
## 
## Step:  AIC=-151.29
## log(Volume) ~ Girth + Height + I(Girth^2)
## 
##                       Df  Sum of Sq     RSS     AIC
## <none>                              0.18189 -151.29
## + I(Height^2)          1 0.00248670 0.17941 -149.72
## + I(Girth * Height)    1 0.00089844 0.18100 -149.44
```

```
fit_forward_aic
```

```
## 
## Call:
## lm(formula = log(Volume) ~ Girth + Height + I(Girth^2), data = trees)
## 
## Coefficients:
## (Intercept)        Girth       Height   I(Girth^2)  
##   -0.783931     0.285333     0.015701    -0.004954
```

Answer:

Here the AIC of forward selection also prefers model2: log(Volume) ~ Girth + Height + I(Girth^2), so model2 is selected in this case.

(3) Backward elimination

```
# backward selection using AIC
fit_backward_aic = step(model_raw, direction = "backward")
```

```
## Start:  AIC=-147.73
## log(Volume) ~ Girth + Height + I(Girth^2) + I(Height^2) + I(Girth *
##     Height)
##
##                      Df Sum of Sq     RSS     AIC
## - I(Girth * Height)  1  0.000085 0.17941 -149.72
## - I(Height^2)        1  0.001674 0.18100 -149.44
## - Height             1  0.005225 0.18455 -148.84
## <none>                           0.17932 -147.73
## - I(Girth^2)         1  0.012456 0.19178 -147.65
## - Girth              1  0.091356 0.27068 -136.97
##
## Step:  AIC=-149.71
## log(Volume) ~ Girth + Height + I(Girth^2) + I(Height^2)
##
##                Df Sum of Sq     RSS     AIC
## - I(Height^2)  1  0.002487 0.18189 -151.29
## - Height       1  0.005377 0.18478 -150.80
## <none>                     0.17941 -149.72
## - I(Girth^2)   1  0.055232 0.23464 -143.39
## - Girth        1  0.248603 0.42801 -124.76
##
## Step:  AIC=-151.29
## log(Volume) ~ Girth + Height + I(Girth^2)
##
##               Df Sum of Sq     RSS     AIC
## <none>                     0.18189 -151.29
## - I(Girth^2)  1   0.08025 0.26214 -141.96
## - Height      1   0.21815 0.40004 -128.85
## - Girth       1   0.32692 0.50881 -121.40
```

```r
fit_backward_aic
```

```
##
## Call:
## lm(formula = log(Volume) ~ Girth + Height + I(Girth^2), data = trees)
##
## Coefficients:
## (Intercept)        Girth       Height   I(Girth^2)
##   -0.783931     0.285333     0.015701    -0.004954
```

Answer:

Here the AIC of backward elimination again prefers model2: log(Volume) ~ Girth + Height + I(Girth^2), so model2 is selected in this case.

(4) Orthogonal polynomials

```r
# fit an orthogonal second-order polynomial regression including the interaction term
model_orthogonal = lm(log(Volume)~polym(Girth, Height, degree=2), trees)
summary(model_orthogonal)
```

```
##
## Call:
## lm(formula = log(Volume) ~ polym(Girth, Height, degree = 2),
##     data = trees)
##
```

```
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.159718 -0.041905 -0.003371  0.055167  0.133780
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          3.27472    0.02370 138.163  < 2e-16
## polym(Girth, Height, degree = 2)1.0  2.51882    0.11972  21.039  < 2e-16
## polym(Girth, Height, degree = 2)2.0 -0.24312    0.18449  -1.318    0.200
## polym(Girth, Height, degree = 2)0.1  0.54249    0.11339   4.784 6.52e-05
## polym(Girth, Height, degree = 2)1.1 -0.11845    1.08511  -0.109    0.914
## polym(Girth, Height, degree = 2)0.2 -0.05025    0.10402  -0.483    0.633
##
## (Intercept)                         ***
## polym(Girth, Height, degree = 2)1.0 ***
## polym(Girth, Height, degree = 2)2.0
## polym(Girth, Height, degree = 2)0.1 ***
## polym(Girth, Height, degree = 2)1.1
## polym(Girth, Height, degree = 2)0.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08469 on 25 degrees of freedom
## Multiple R-squared:  0.9784, Adjusted R-squared:  0.9741
## F-statistic: 226.7 on 5 and 25 DF,  p-value: < 2.2e-16
```

Answer:

Here the orthogonal polynomial model summary shows that only the simplest model with first-order terms is preferred, which is model1: log(Volume) ~ Girth + Height.

(5) Final decision

```
# compare model1 with model2
anova(model1, model2)
```

```
## Analysis of Variance Table
##
## Model 1: log(Volume) ~ Girth + Height
## Model 2: log(Volume) ~ Girth + Height + I(Girth^2)
##   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
## 1     28 0.26214
## 2     27 0.18189  1  0.080245 11.912 0.001851 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer:

The p-value of F-test is 0.001851, so we reject null hypothesis and prefer model2: log(Volume) ~ Girth + Height + I(Girth^2).

Finally, we pick model2 as our reasonably simplified model: log(Volume) ~ Girth + Height + I(Girth^2).

Problem 3

see next page

STAT 34700 HW6 Sarah Adilijiang

Problem 3 :

(a) for any $X \in R^{n\times p}$. $Y \in R^{n}$. $\beta \in R^{p}$

$$|<Y, X\beta>| = |(X\beta)^T Y| = \left|\sum_{i=1}^{n}\left(Y_i \cdot \sum_{j=1}^{p} X_{ij}\beta_j\right)\right| = \left|\sum_{i=1}^{n}\sum_{j=1}^{p}(X_{ij}\cdot Y_i \cdot \beta_j)\right| = \left|\sum_{j=1}^{p}\left(\sum_{i=1}^{n} X_{ij}\cdot Y_i\right)\cdot\beta_j\right|$$

$$= \left|\sum_{j=1}^{p}(X_j^T Y)\beta_j\right| \le \sum_{j=1}^{p}\left|(X_j^T Y)\beta_j\right| = \sum_{j=1}^{p}|X_j^T Y|\cdot|\beta_j| \le \sum_{j=1}^{p}\max_j|X_j^T Y|\cdot|\beta_j|$$

$$= \max_j |X_j^T Y|\cdot\sum_j|\beta_j|$$

$\Rightarrow$ $|<Y, X\beta>| \le \max_j |X_j^T Y|\cdot\sum_j|\beta_j|$ for any X. Y. β

(b) when $\lambda \ge \max_j |X_j^T Y|$, for any $\beta \in R^p$.

$$Loss(\beta) = \frac{1}{2}\|Y - X\beta\|_2^2 + \lambda\sum_j|\beta_j| = \frac{1}{2}(Y-X\beta)^T(Y-X\beta) + \lambda\sum_j|\beta_j|$$

$$= \frac{1}{2}\left(Y^TY - (X\beta)^TY - Y^T(X\beta) + (X\beta)^T(X\beta)\right) + \lambda\sum_j|\beta_j|$$ ($\because$ $(X\beta)^TY = Y^T(X\beta)$, since $(X\beta)^TY$ is a scalar)

$$= \frac{1}{2}\left(\|Y\|_2^2 - 2(X\beta)^TY + \|X\beta\|_2^2\right) + \lambda\sum_j|\beta_j|$$ ($\because$ $\|X\beta\|_2^2 \ge 0$)

$$\ge \frac{1}{2}\left(\|Y\|_2^2 - 2(X\beta)^TY\right) + \lambda\sum_j|\beta_j|$$

$$= \frac{1}{2}\|Y\|_2^2 - (X\beta)^TY + \lambda\sum_j|\beta_j|$$ ($\because$ $\lambda \ge \max_j|X_j^TY|$)

$$\ge \frac{1}{2}\|Y\|_2^2 - (X\beta)^TY + \max_j|X_j^TY|\cdot\sum_j|\beta_j|$$ ($\because$ $\max_j|X_j^TY|\cdot\sum_j|\beta_j| \ge |<Y, X\beta>| = |(X\beta)^TY|$)

$$\ge \frac{1}{2}\|Y\|_2^2 - (X\beta)^TY + |(X\beta)^TY|$$ ($\because$ $|(X\beta)^TY| \ge (X\beta)^TY$ since $(X\beta)^TY$ is a scalar)

$$\ge \frac{1}{2}\|Y\|_2^2$$

$$= Loss(0_p)$$

$\Rightarrow$ when $\lambda \ge \max_j|X_j^TY|$, $Loss(\beta) \ge Loss(0_p)$ for any β

i.e. $\hat{\beta} = 0_p = (0, \cdots, 0)$

STAT34700 HW6 Sarah Adilijiang

(c) when $\lambda \geq \max_j |X_j^T Y|$ and for any $\beta \neq 0_p$, we have:

① when $\|X\beta\|_2 = 0 \Rightarrow \|X\beta\|_2^2 = 0$ & $X\beta = 0_n \in R^n$.

$$\Rightarrow Loss(\beta) = \frac{1}{2}\|Y - X\beta\|_2^2 + \lambda \sum_j |\beta_j| = \frac{1}{2}\|Y\|_2^2 + \lambda \sum_j |\beta_j| = Loss(0_p) + \lambda \sum_j |\beta_j|$$

$\because \lambda \geq \max_j |X_j^T Y| > 0$, $\beta \neq 0_p \Rightarrow \sum_j |\beta_j| > 0$

$$\Rightarrow Loss(\beta) = Loss(0_p) + \lambda \sum_j |\beta_j| > Loss(0_p)$$

② when $\|X\beta\|_2 > 0 \Rightarrow \|X\beta\|_2^2 > 0$.

$$\Rightarrow Loss(\beta) = \frac{1}{2}\|Y - X\beta\|_2^2 + \lambda \sum_j |\beta_j| = \frac{1}{2}\|Y\|_2^2 - (X\beta)^T Y + \frac{1}{2}\|X\beta\|_2^2 + \lambda \sum_j |\beta_j|$$

$$= loss(0_p) + \frac{1}{2}\|X\beta\|_2^2 + \lambda \sum_j |\beta_j| - (X\beta)^T Y$$

$$\geq loss(0_p) + \frac{1}{2}\|X\beta\|_2^2 + \max_j |X_j^T Y| \cdot \sum_j |\beta_j| - (X\beta)^T Y$$

$$\geq loss(0_p) + \frac{1}{2}\|X\beta\|_2^2 + |(X\beta)^T Y| - (X\beta)^T Y$$

$$\geq Loss(0_p) + \frac{1}{2}\|X\beta\|_2^2$$

$$> loss(0_p)$$

$\Rightarrow$ in both cases, we have $Loss(\beta) > Loss(0_p)$ if $\beta \neq 0_p$

$\Rightarrow$ $\hat{\beta} = 0_p = (0, \cdots, 0)$ is the unique minimizer when we have large $\lambda \geq \max_j |X_j^T Y|$