Homework 6

Charles Hwang

12/11/2022

Charles Hwang

Dr. Xi

STAT 408-001

2022 December 11

Problem 1

```
rm(list=ls())
library(datasets); data(mtcars)
```

Problem 1a

```
summary(glm(am~mpg+hp,family=binomial,data=mtcars))
```

```
##
## Call:
## glm(formula = am ~ mpg + hp, family = binomial, data = mtcars)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.41460  -0.42809  -0.07021   0.16041   1.66500
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -33.60517   15.07672  -2.229   0.0258 *
## mpg           1.25961    0.56747   2.220   0.0264 *
## hp            0.05504    0.02692   2.045   0.0409 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 43.230  on 31  degrees of freedom
## Residual deviance: 19.233  on 29  degrees of freedom
## AIC: 25.233
##
## Number of Fisher Scoring iterations: 7
```

```
tm<-summary(glm(am~mpg+hp,family=binomial,data=mtcars))$coefficients
```

We can see that the estimated regression coefficients from this logistic model are β_0 = -33.605171, β_{mpg} = 1.2596146, and $\beta_{hp} = 0.0550446$.

Interpretation of β_0 (intercept term): We estimate a hypothetical car with 0 miles-per-gallon and 0 horsepower (which would not make sense) would have a $\frac{e^{-33.60517}}{1+e^{-33.60517}} = 0.00000000000002543664$ percent probability of having a manual transmission ($P(Y=1)$).

Interpretation of β_{mpg} (coefficient for miles per gallon): We estimate there is approximately a $e^{1.259615} - 1 =$ 252.4063154 **percent increase** in the odds of having a manual transmission ($P(Y = 1)$) for every 1 mile-per-gallon increase in a car's fuel economy, holding all other variables constant.

Interpretation of β_{hp} (coefficient for horsepower): We estimate there is approximately a $e^{0.05504458} - 1 =$ 5.6587715 **percent increase** in the odds of having a manual transmission ($P(Y = 1)$) for every 1 horsepower increase in a car's power output, holding all other variables constant.

Problem 1b

```
1-1/(1+exp(-(tm["(Intercept)","Estimate"]+20*tm["mpg","Estimate"]+180*tm["hp","Estimate"])))
```

```
## [1] 0.1831506
```

We predict there is approximately a $1 - \frac{1}{1+e^{-(-33.60517+1.259615(20)+0.05504458(180))}} = 18.3150628$ percent probability that a car with a fuel economy of 20 miles per gallon and a power output of 180 horsepower has an *automatic* transmission ($P(Y = 0) = 1 - P(Y = 1)$).

Problem 1c

```
set.seed(1211)
samp<-sample(1:nrow(mtcars),round(0.8*nrow(mtcars)))
train<-mtcars[samp,]
test<-mtcars[-samp,]
t<-table(test$am,round(predict(glm(am~mpg+hp,family=binomial,data=train),test,type="response")))
sum(diag(t))/nrow(test)
```

```
## [1] 0.6666667
```

We can see the prediction accuracy is 66.6666667 percent.

Problem 1d

```
t
```

```
##
##     0 1
##   0 3 0
##   1 2 1
```

```
Sensitivity<-t["1","1"]/sum(t["1",]) # True positive rate
Specificity<-t["0","0"]/sum(t["0",]) # True negative rate
Precision<-t["1","1"]/sum(t[,"1"])
data.frame(Sensitivity,Specificity,Precision)
```

```
##   Sensitivity Specificity Precision
## 1   0.3333333           1         1
```

We can see the sensitivity (true positive rate) is 0.3333333, the specificity (true negative rate) is 1, and the precision is 1.

Problem 2

```r
s<-read.csv("/Users/newuser/Desktop/Notes/Graduate/STAT 408 - Applied Regression Analysis/seatpos.csv")
```

Problem 2a

```r
summary(lm(hipcenter~.,data=s))
```

```
##
## Call:
## lm(formula = hipcenter ~ ., data = s)
##
## 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
```

We can see the intercept term has by far the strongest magnitude in the model (436.4321282), which balances out the variables with negative coefficients (`HtShoes`, `Arm`, `Thigh`, and `Leg`). The remaining four variables have relatively weak positive coefficients.

We can see from problem 2b that there are several variables that are very highly correlated with one another. It is likely that linear regression is not appropriate to use on these data.

Problem 2b

```r
round(cor(s),3)
```

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

We can see the `HtShoes`, `Ht`, `Seated`, and `Leg` variables are all very highly correlated with one another ($r > 0.9084334$) *except* for the pairing between the `Seated` and `Leg` variables which are strongly correlated with each other ($r = 0.8119143$). The `Weight` variable is also strongly correlated with the `HtShoes` ($r = 0.8281773$) and `Ht` ($r = 0.8285257$) variables.

Looking at the model in problem 2a, there do not appear to be any apparent relations specifically between the high correlations and the model fitting. We can see the coefficients for the four variables have different signs. However, having too many variables highly correlated with one another likely produces misleading results for interpretation, and it is possible the effects of these variables may be negating each other or "cancelling each other out" in the model (Lecture 17, Slide 33).

Problem 2c

```
round(summary(prcomp(s[,-9],scale=TRUE))$importance,4) # Removing response variable (hipcenter)
```

```
##                           PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8
## Standard deviation     2.3818 1.1121 0.6810 0.4909 0.4407 0.3731 0.2244 0.0399
## Proportion of Variance 0.7091 0.1546 0.0580 0.0301 0.0243 0.0174 0.0063 0.0002
## Cumulative Proportion  0.7091 0.8638 0.9217 0.9518 0.9761 0.9935 0.9998 1.0000
```

```
sum(summary(prcomp(s[,-9],scale=TRUE))$importance["Standard deviation",1:2])
```

```
## [1] 3.493954
```

```
summary(prcomp(s[,-9],scale=TRUE))$importance["Cumulative Proportion","PC2"]
```

```
## [1] 0.86375
```

We can see the first two components have approximately 86.375 percent of the variance.

Problem 2d

```
prcomp(s[,-9],scale=TRUE)$rotation[,1:2] # Removing response variable (hipcenter)
```

```
##                  PC1        PC2
## Age     -0.007219379  0.8763467
## Weight  -0.366979122  0.0448877
## HtShoes -0.411460536 -0.1055831
## Ht      -0.412057421 -0.1119799
## Seated  -0.381270226 -0.2178995
## Arm     -0.348771387  0.3742641
## Thigh   -0.327523319  0.1251793
## Leg     -0.389747512 -0.0555930
```

We can see the first principal component is a linear combination of the variables as the signs of the coefficients are all the same. The second principal component appears to compare the `HtShoes`, `Ht`, `Seated`, and `Leg` variables (which we saw in problem 2b are very highly correlated with one another) with the other variables.

Problem 2e

```
spc<-data.frame(s$hipcenter,prcomp(s[,-9],scale=TRUE)$x[,1:2]) # Creating new dataframe
summary(lm(s.hipcenter~.,data=spc))
```

```
##
## Call:
## lm(formula = s.hipcenter ~ ., data = spc)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -84.643 -25.582  -0.743  24.887  61.798
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -164.885      5.772 -28.568  < 2e-16 ***
## PC1           19.701      2.456   8.022 1.93e-09 ***
## PC2           11.321      5.259   2.153   0.0383 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.58 on 35 degrees of freedom
## Multiple R-squared:  0.6634, Adjusted R-squared:  0.6442
## F-statistic:  34.5 on 2 and 35 DF,  p-value: 5.292e-09
```

We can see the first principal component has a positive coefficient (19.7007453) which suggests that all independent variables grouped together are directly proportional to the response variable `hipcenter`. The second principal component also has a positive coefficient (11.3211943) which suggests that the group of highly correlated variables (`HtShoes`, `Ht`, `Seated`, and `Leg`) together are directly proportional to the response variable `hipcenter`.

Looking at the model in problem 2a, we can see the signs of the intercept terms are reversed. The intercept term in the principal component model is the same as the mean response (-164.8848684), which means the baseline observation β_0 has the same value for the response variable. Meanwhile, the intercept term in the linear model (436.4321282) is much further away, making extrapolation more impractical and unreasonable. We can also see that the intercept term and all of the variables in the principal components model significant at the $\alpha = 0.05$ level, while only the intercept term in the linear model is significant.

Problem 3

```r
f<-read.csv("/Users/newuser/Desktop/Notes/Graduate/STAT 408 - Applied Regression Analysis/fat.csv")
testf<-f[seq(1,nrow(f),10),]
trainf<-f[-seq(1,nrow(f),10),]
```

Problem 3a

```r
summary(lm(siri~.-brozek-density,data=trainf)) # Removing brozek and density
```

```
##
## Call:
## lm(formula = siri ~ . - brozek - density, data = trainf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.8605 -0.5784  0.2650  0.9586  2.9291
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.612054   6.408777  -1.032 0.303391
## age          0.004228   0.011419   0.370 0.711590
## weight       0.387944   0.023592  16.444  < 2e-16 ***
## height       0.033490   0.038216   0.876 0.381847
## adipos      -0.470841   0.105948  -4.444 1.43e-05 ***
```

```
## free        -0.573609   0.014389 -39.865  < 2e-16 ***
## neck        -0.023312   0.084028  -0.277 0.781726
## chest        0.122950   0.037208   3.304 0.001119 **
## abdom        0.105760   0.038440   2.751 0.006455 **
## hip         -0.004548   0.054266  -0.084 0.933289
## thigh        0.176306   0.051072   3.452 0.000673 ***
## knee         0.025355   0.090732   0.279 0.780172
## ankle        0.110958   0.095343   1.164 0.245832
## biceps       0.138203   0.061581   2.244 0.025861 *
## forearm      0.204817   0.069502   2.947 0.003572 **
## wrist        0.164980   0.203144   0.812 0.417635
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.46 on 210 degrees of freedom
## Multiple R-squared:  0.9716, Adjusted R-squared:  0.9695
## F-statistic: 478.5 on 15 and 210 DF,  p-value: < 2.2e-16
```

```r
LinearModel<-sqrt(mean((testf$siri-predict(lm(siri~.-brozek-density,data=trainf),testf))^2))
```

Problem 3b

```r
step(lm(siri~.-brozek-density,data=trainf),trace=0,direction="backward") # Removing brozek and density
```

```
##
## Call:
## lm(formula = siri ~ weight + adipos + free + chest + abdom +
##     thigh + ankle + biceps + forearm, data = trainf)
##
## Coefficients:
## (Intercept)       weight       adipos         free        chest        abdom
##     -2.9190       0.3925      -0.5277      -0.5698       0.1246       0.1179
##       thigh        ankle       biceps      forearm
##      0.1561       0.1475       0.1490       0.2146
```

```r
sw<-lm(siri~weight+adipos+free+chest+abdom+thigh+ankle+biceps+forearm,data=trainf)
summary(sw)
```

```
##
## Call:
## lm(formula = siri ~ weight + adipos + free + chest + abdom +
##     thigh + ankle + biceps + forearm, data = trainf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9500 -0.5415  0.2788  0.9282  3.0172
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.91896    3.60860  -0.809 0.419468
## weight       0.39252    0.01951  20.121  < 2e-16 ***
## adipos      -0.52768    0.08579  -6.151 3.69e-09 ***
## free        -0.56977    0.01354 -42.094  < 2e-16 ***
## chest        0.12462    0.03587   3.474 0.000620 ***
## abdom        0.11790    0.03528   3.342 0.000981 ***
```

```
## thigh        0.15611    0.04156   3.756 0.000222 ***
## ankle        0.14752    0.08667   1.702 0.090175 .
## biceps       0.14905    0.06001   2.484 0.013759 *
## forearm      0.21464    0.06609   3.248 0.001350 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.447 on 216 degrees of freedom
## Multiple R-squared:  0.9713, Adjusted R-squared:  0.9701
## F-statistic: 811.4 on 9 and 216 DF,  p-value: < 2.2e-16
```

```
StepwiseAIC<-sqrt(mean((testf$siri-predict(sw,testf))^2))
```

Problem 3c

```
pc<-prcomp(f[,-c(1:3)],scale=TRUE) # Removing brozek, density, and response variable (siri)
round(summary(pc)$importance[,1:7],5)
```

```
##                            PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.07429 1.26329 1.02923 0.81766 0.77470 0.59621 0.56331
## Proportion of Variance 0.63009 0.10639 0.07062 0.04457 0.04001 0.02370 0.02115
## Cumulative Proportion  0.63009 0.73648 0.80710 0.85167 0.89168 0.91538 0.93653
```

```
trainpc<-data.frame(trainf$siri,pc$x[-seq(1,nrow(f),10),1:7]) # Creating new dataframes
testpc<-data.frame(pc$x[seq(1,nrow(f),10),1:7])
summary(lm(trainf.siri~.,data=trainpc))
```

```
##
## Call:
## lm(formula = trainf.siri ~ ., data = trainpc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0922  -2.4742  -0.1345   2.6956   7.9567
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.20324    0.22733  84.473  < 2e-16 ***
## PC1          1.59774    0.07415  21.546  < 2e-16 ***
## PC2         -3.37318    0.17984 -18.756  < 2e-16 ***
## PC3         -1.15329    0.22138  -5.209 4.38e-07 ***
## PC4          0.47506    0.28721   1.654 0.099555 .
## PC5          1.18737    0.30670   3.871 0.000143 ***
## PC6          5.59535    0.37090  15.086  < 2e-16 ***
## PC7          1.75499    0.41510   4.228 3.47e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.414 on 218 degrees of freedom
## Multiple R-squared:  0.8385, Adjusted R-squared:  0.8333
## F-statistic: 161.7 on 7 and 218 DF,  p-value: < 2.2e-16
```

```
PrincipalComponent<-sqrt(mean((testf$siri-predict(lm(trainf.siri~.,data=trainpc),testpc))^2))
```

Problem 3d

```
library(MASS)
r<-lm.ridge(siri~.-brozek-density,data=trainf,lambda=seq(0,0.05,0.002)) # Removing brozek and density
which.min(r$GCV) # Cross-validation for smallest tuning parameter
```

```
## 0.034
##    18
```

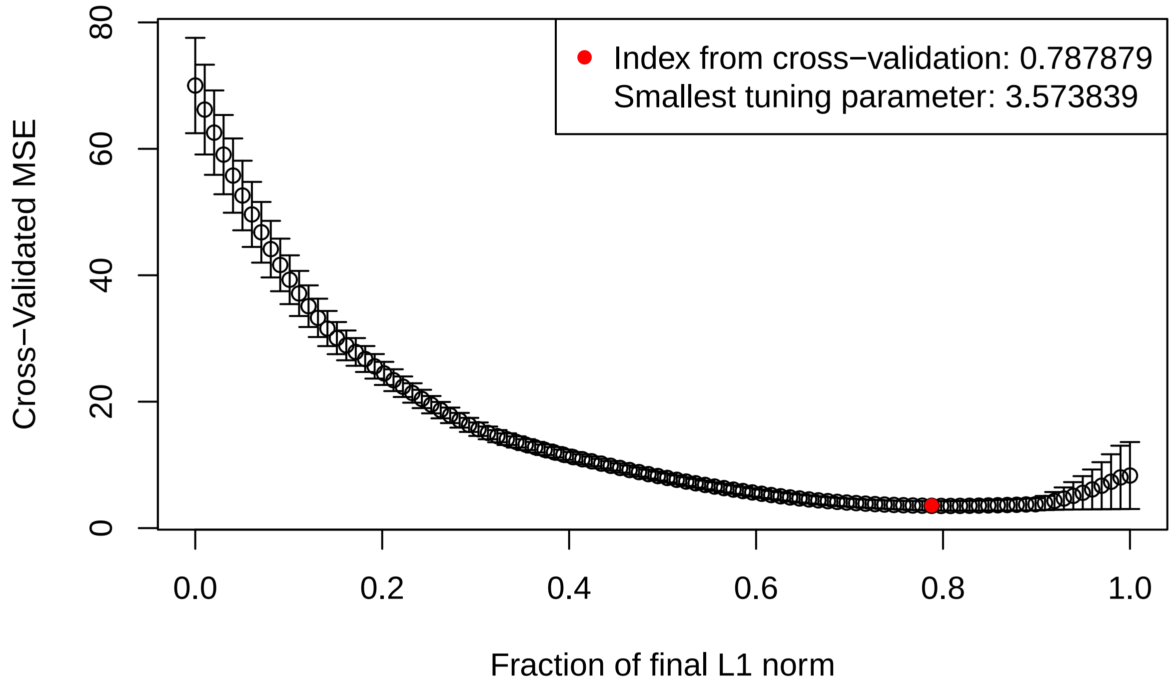
```
lr<-coef(r)[names(which.min(r$GCV)),]
lr
```

```
##                       age       weight       height       adipos         free
## -7.232853450  0.004090899  0.384509958  0.035217648 -0.466756567 -0.572019380
##         neck        chest        abdom          hip        thigh         knee
## -0.022046078  0.123747500  0.108439959 -0.002094340  0.176527454  0.027366782
##        ankle       biceps      forearm        wrist
##  0.113561844  0.139323604  0.205531096  0.162722876
```

```
Ridge<-sqrt(mean((testf$siri-cbind(1,as.matrix(testf[,-c(1:3)]))%*%lr)^2)) # coding 13.R
```

Problem 3e

```
library(lars)
lars<-lars(as.matrix(trainf[,-c(1:3)]),trainf$siri) # Removing brozek, density, and siri
set.seed(1112)                                      # coding 13.R
cv.lars(as.matrix(trainf[,-c(1:3)]),trainf$siri)
cv<-cv.lars(as.matrix(trainf[,-c(1:3)]),trainf$siri,plot.it=FALSE)
ll<-cv$index[which.min(cv$cv)]
points(ll,min(cv$cv),col="red",pch=16) # Cross-validation for smallest tuning parameter
legend("topright",col="red",pch=c(16,NA),legend=c("Index from cross-validation: 0.787879","Smallest tuni
```




```
LASSO<-sqrt(mean((testf$siri-predict(lars,as.matrix(testf[,-c(1:3)]),s=l1,mode="fraction")$fit)^2))
```

Problem 3f

```
data.frame(LinearModel,StepwiseAIC,PrincipalComponent,Ridge,LASSO)
```

```
##   LinearModel StepwiseAIC PrincipalComponent    Ridge    LASSO
## 1    1.946023     1.98911           3.896717 1.937171 1.946278
```

We can see the ridge regression model has the lowest root mean squared error (RMSE) at 1.9371706. The full linear model (1.9460232) and LASSO regression model (1.9462783) had marginally higher RMSE values, followed by the stepwise regression model (1.9891098). The principal component regression (PCR) model had the highest RMSE (3.8967167).

I am not too surprised by the comparison between the models. It makes sense intuitively that more complex methods like ridge and LASSO regression may be able to capture more complex trends in the data, and these methods also use cross-validation to choose the most optimal parameters. It also makes sense that the linear and stepwise regression models would perform slightly worse and similar to each other, as these are basic but robust methods. It may be possible that principal component regression is unsuitable for this dataset or that there are additional information in components 8-18 that were left out of the predictions, which may explain its relatively high RMSE.