Project 05

Indrajith Wasala Mudiyanselage

Question 01

a

I think standardization of the variables before performing the PCA would be a good idea. If ht e data is standardized, the scale of all variables would be the same. Otherwise, the variable with higher variance will have higher wights while other variables with lower variance get lower weights. Also, standardized data would not be affected by the change of units.

b

I would recommend the first six principal components as it approximately explains 90% of the total variance.

	PC1	PC2	PC3	PC4	PC5	PC6
AtBat	0.20	0.31	-0.23	0.06	-0.05	0.07
Hits	0.20	0.31	-0.23	0.04	-0.04	0.08
HmRun	0.21	0.22	-0.06	0.02	0.33	0.15
Runs	0.20	0.32	-0.19	0.01	0.06	0.14
RBI	0.24	0.26	-0.16	0.02	0.17	0.11
Walks	0.21	0.17	-0.17	0.01	0.09	-0.00
Years	0.28	-0.25	0.10	-0.01	-0.10	-0.03
CAtBat	0.32	-0.20	0.04	0.02	-0.12	-0.02
CHits	0.32	-0.19	0.03	0.02	-0.12	-0.03
CHmRun	0.32	-0.12	0.06	0.00	0.11	0.04
CRuns	0.33	-0.18	0.04	-0.00	-0.09	-0.00
CRBI	0.33	-0.17	0.04	0.02	-0.01	-0.01
CWalks	0.31	-0.20	0.05	-0.01	-0.05	-0.04
PutOuts	0.08	0.10	-0.15	0.02	0.21	-0.95
Assists	-0.00	0.11	-0.18	0.07	-0.64	-0.02
Errors	-0.01	0.14	-0.21	0.08	-0.54	-0.14
League_A	0.08	0.26	0.40	-0.01	-0.07	-0.04
League_N	-0.08	-0.26	-0.40	0.01	0.07	0.04
Division_E	0.03	0.04	-0.05	-0.70	-0.05	-0.01
Division_W	-0.03	-0.04	0.05	0.70	0.05	0.01
NewLeague_A	0.07	0.25	0.42	-0.01	-0.09	-0.06
NewLeague_N	-0.07	-0.25	-0.42	0.01	0.09	0.06

Table 1: Loadings of forst 6 principal components

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12
Standard deviation	2.7088	2.0888	1.8726	1.4079	1.2931	0.9162	0.8331	0.7188	0.5160	0.4993	0.4298	0.3615
Proportion of Variance	0.3335	0.1983	0.1594	0.0901	0.0760	0.0382	0.0316	0.0235	0.0121	0.0113	0.0084	0.0059
Cumulative Proportion	0.3335	0.5319	0.6913	0.7813	0.8573	0.8955	0.9271	0.9505	0.9626	0.9740	0.9824	0.9883

Table 2: Proportion of Variance and Cumulative Proportions of fist 12 principal components

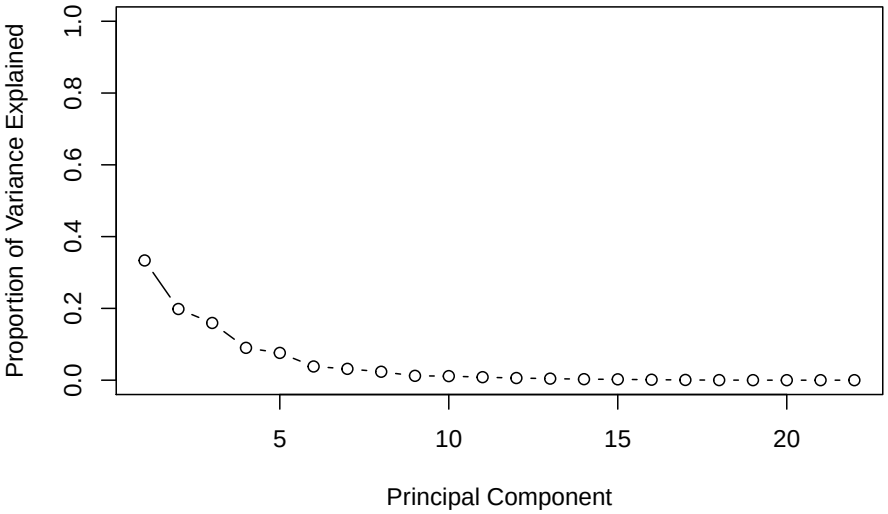

Figure 1: Scree plot

c

	AtBat	Hits	HmRun	Runs	RBI	Walks	Years	CAtBat
PC1	0.545	0.539	0.566	0.551	0.647	0.565	0.749	0.877
PC2	0.646	0.638	0.454	0.668	0.549	0.351	-0.519	-0.423

	CHits	CHmRun	CRuns	CRBI	CWalks	PutOuts	Assists	Errors
PC1	0.878	0.855	0.901	0.906	0.842	0.206	-0.004	-0.025
PC2	-0.407	-0.255	-0.375	-0.365	-0.409	0.216	0.238	0.283

Table 3: Correlation of the standardized quantitative variables with the first two components

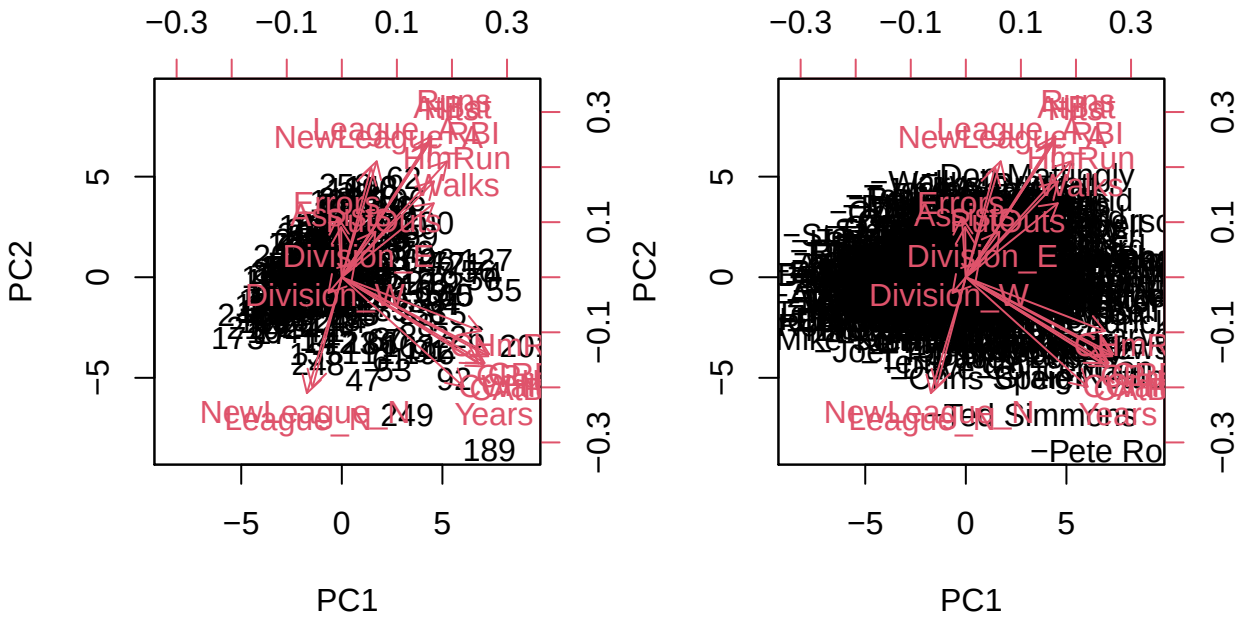

Figure 2: Biplots with observation number and Player names

Based on the above biplot, predictors like CHmRun, CRBI, RBI, CRuns, CAtBat, Years loaded heavily on PC1 while predictors like Runs, AtBat, RBI, HmRun, Walks, League_A, NewLeague_A loaded heavily on PC2. Players like “Don Baylor”, “Darrell Evans”, and “Pete Rose” (Observations 50, 55, and 189) score high on PC1. Players like “Don Mattingly”, “Wally Joyner” and “Jose Canseco” score high on PC2 (Observations 62, 258, and 109).

Question 02

a

Standardization of the variables before clustering is dependent on the given scenario. If we would like to allocate larger weights to the variables with high variances, it is better not to standardize variables. Otherwise, in general, standardization is recommended.

b

Since most of the variables (out of 19 variables) are not strongly correlated, I would use matric-based distance to cluster the players.

c

In the first cluster, there are 235 players while 28 players in the second cluster. The mean salary of the first cluster ($ 482.35) is higher than the second cluster($ 985.55). The second cluster means of the variables are higher than the first cluster means of the variables except for the variables Assists, Errors, League_A, Division_E, and NewLeague_A.

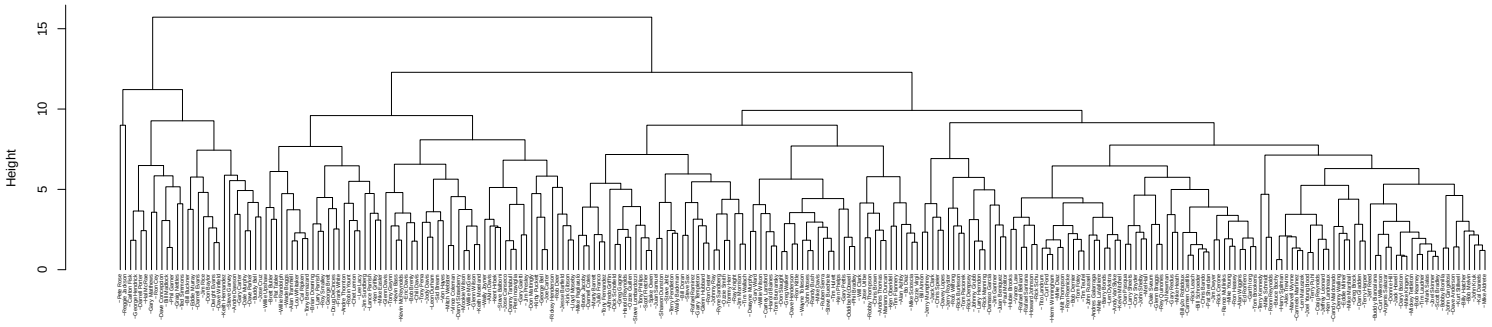

Figure 3: Hierarchical Clustering with Scaled Features

	players
1	235
2	28

Table 4: Number of players in each cluster

	AtBat	Hits	HmRun	Runs	RBI	Walks	Years	CAtBat
First.cluster	399.57	106.81	10.99	54.18	49.45	39.91	6.30	2100.63
Second.cluster	437.82	116.39	16.93	59.50	68.61	51.25	15.82	7331.64

	CAtBat	CHits	CHmRun	CRuns	CRBI	CWalks	PutOuts	Assists	Errors
First.cluster	2100.63	563.64	48.03	279.94	242.55	198.24	282.35	124.74	8.72
Second.cluster	7331.64	2052.82	247.29	1043.39	1067.86	780.82	360.89	68.54	7.54

	League_A	League_N	Division_E	Division_W	NewLeague_A	NewLeague_N
First.cluster	0.54	0.46	0.50	0.50	0.54	0.46
Second.cluster	0.43	0.57	0.43	0.57	0.46	0.54

Table 5: Cluster means of the variables

	Mean Salary
First.cluster.sal.mean	482.35
Second.cluster.sal.mean	985.55

Table 6: Mean salary of the players in the two clusters

d

In the first cluster, there are 188 players, while 75 players in the second cluster. The mean salary of the first cluster ($ 380.67) is higher than the second cluster($ 925.11). The second cluster means of the variables are higher than the first cluster means of the variables except for the variables Assists, Errors, League_N, Division_W, and NewLeague_N.

	players.km
1	188
2	75

Table 7: Number of players in each cluster (K-means)

	AtBat	Hits	HmRun	Runs	RBI	Walks	Years	CAtBat
First.km.cluster	381.36	100.57	9.41	50.01	45.27	36.44	5.23	1544.48
Second.km.cluster	459.49	126.03	17.15	66.61	67.08	52.83	12.52	5447.63

	CAtBat	CHits	CHmRun	CRuns	CRBI	CWalks	PutOuts	Assists	Errors
First.km.cluster	1544.48	407.08	30.93	196.04	169.54	134.49	267.43	127.90	9.03
Second.km.cluster	5447.63	1512.05	165.27	775.27	733.68	575.53	349.07	95.84	7.51

	League_A	League_N	Division_E	Division_W	NewLeague_A	NewLeague_N
First.km.cluster	0.46	0.54	0.45	0.55	0.48	0.52
Second.km.cluster	0.71	0.29	0.59	0.41	0.68	0.32

Table 8: Cluster means of the variables (K-means)

	Mean Salary
First.cluster.sal.mean.km	380.67
Second.cluster.sal.mean.km	925.11

Table 9: Mean salary of the players in the two clusters (K-means)

e

Both methods give very similar clustering. Therefore identifying the better algorithm that gives more sensible results would be a difficult choice.

Question 3

a

The Linear regression model was fit and the test MSE is given in the below table 10.

b

The PCR model with M chosen optimally via LOOCV was fit. The test MSE is given in the below table 10.

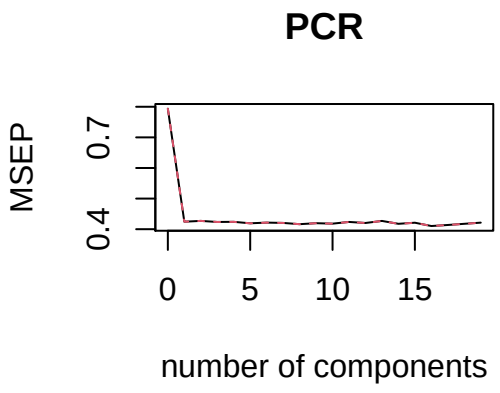

Figure 4: Validation plots for PCR

c

The PLS model with M chosen optimally via LOOCV was fit. The test MSE is given in the below table 10.

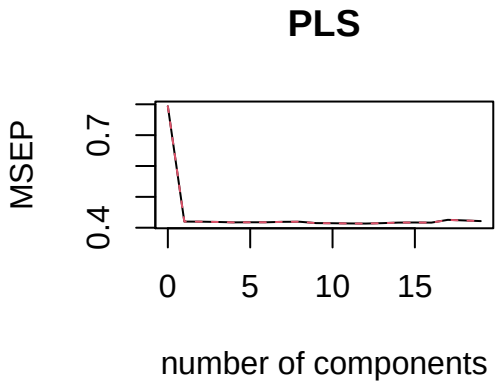

Figure 5: Validation plots for PLS

d

Ridge regression with penalty parameter chosen optimally via LOOCV was fit. The test MSE is given in the below table 10.

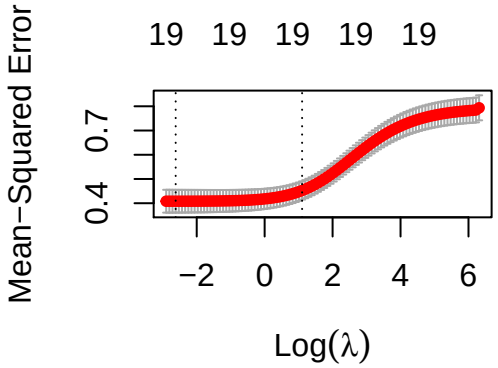

Figure 6: Plot of test MSE vs log(lambda) using ridge regression

e

According to the table, I would recommend the Ridge regression model as it has the lowest test MSE. On the other, PCR and PSL have similar values for test MSE, while the linear regression approach has the largest test MSE.

	Linear Reg.	PCR	PLS	Ridge Reg
Test MSE	0.4214	0.4138	0.4148	0.3607

Table 10: *Summary of test MSE*

R Codes

```
## ----setup, include=FALSE-------------------------------------
knitr::opts_chunk$set(echo = TRUE)


## ----include=FALSE--------------------------------------------
# Question 01

library(ISLR)
Hitters.new<-na.omit(Hitters)
str(Hitters.new)


## ----include=FALSE--------------------------------------------
# b)

# Create dummy variables
library(fastDummies)
str(Hitters.new)
dummy<-dummy_cols(Hitters.new[,c(14,15,20)])
Hitter.pred<-cbind(Hitters.new[,-c(14,15,19,20)],dummy[,4:9])
str(Hitter.pred)
Hitters.pca <- prcomp(Hitter.pred, center = T, scale = T)

#Get the loading matrix
Hitters.pca$rotation


## ----results="asis", echo=FALSE-------------------------------

library(xtable)
options(xtable.comment=FALSE)
xtable(Hitters.pca$rotation[,1:6],caption = "Loadings of forst 6 principal components")
xtable(summary(Hitters.pca),caption = "Proportion of Variance and Cumulative Proportions of fist 12 principal co


## ----echo=FALSE,fig.align="center",fig.cap="Scree plot",  out.width = "60%"----
# Scree plot
pc.var <- Hitters.pca$sdev^2
pve <- pc.var/sum(pc.var)
plot(pve, xlab = "Principal Component", ylab = "Proportion of Variance Explained", ylim = c(0,1), type = 'b')


## ----results="asis", echo=FALSE-------------------------------
# c)

# Standardize quantitative variables
x.std <- apply(Hitter.pred[,1:16], 2, function(x){(x-mean(x))/sd(x)})
xtable(cor(Hitters.pca$x[,c(1,2)],x.std)[,1:8],digits=c(0,3,3,3,3,3,3,3,3))
xtable(cor(Hitters.pca$x[,c(1,2)],x.std)[,9:16],digits=c(0,3,3,3,3,3,3,3,3), caption = "Correlation of the stand


## ----echo=FALSE,fig.align="center",fig.cap="Biplots with observation number and Player names",  out.width = "8
Hitter.pred.no<-Hitter.pred
row.names(Hitter.pred.no)<-seq(1:263)
Hitters.pca.new <- prcomp(Hitter.pred.no, center = T, scale = T)
par(mfrow=c(1,2))
biplot(Hitters.pca.new, scale=0)
biplot(Hitters.pca, scale=0)

## ----include=FALSE--------------------------------------------
row.names(Hitter.pred)[c(50,55,189)]
```

```r
row.names(Hitter.pred)[c(62,258,109)]


## ----include=FALSE-----------------------------------------
#########################################################################################################################
# Question 02

# b)

cor(Hitter.pred)


## ---- include=FALSE----------------------------------------
# c)

xsc <- scale(Hitter.pred)
xsc.no <- scale(Hitter.pred.no)
xsc.hc.complete <- hclust(dist(xsc), method = "complete")
xsc.hc.no.complete <- hclust(dist(xsc.no), method = "complete")
players<-cutree(xsc.hc.complete, 2)


## ----echo=FALSE,fig.align="center",fig.height=6, fig.width=20,fig.cap="Hierarchical Clustering with Scaled Fea
plot(xsc.hc.complete, main = "", xlab = "", sub = "", hang = -1, cex = 0.4)

## ----results="asis", echo=FALSE----------------------------
print(xtable(table(players), caption = "Number of players in each cluster"),table.placement="H")
# 1st cluster means of the variables
First.cluster<-apply(Hitter.pred[players==1,],2,mean)
# 2nd cluster means of the variables
Second.cluster<-apply(Hitter.pred[players==2,],2,mean)
comb.means<-rbind(First.cluster,Second.cluster)
print(xtable(comb.means[,1:8]),table.placement="H")
print(xtable(comb.means[,8:16]),table.placement="H")
print(xtable(comb.means[,17:22],caption = "Cluster means of the variables"),table.placement="H")

salaries<-Hitters.new[,19]

# mean salary of the players in the two clusters
First.cluster.sal.mean<-mean(salaries[players==1])
Second.cluster.sal.mean<-mean(salaries[players==2])
comb.salary<-rbind(First.cluster.sal.mean,Second.cluster.sal.mean)
colnames(comb.salary)<-"Mean Salary"
print(xtable(comb.salary,caption = "Mean salary of the players in the two clusters"),table.placement="H")


## ----results="asis", echo=FALSE----------------------------
# d)

# K-means with K = 2
set.seed(1)
km.out <- kmeans(xsc, 2, nstart = 20)
players.km<-km.out$cluster

print(xtable(table(players.km), caption = "Number of players in each cluster (K-means)"))

# Cluster means of the variables
# 1st cluster means of the variables
First.km.cluster<-apply(Hitter.pred[players.km==1,],2,mean)
# 2nd cluster means of the variables
Second.km.cluster<-apply(Hitter.pred[players.km==2,],2,mean)
km.comb.means<-rbind(First.km.cluster,Second.km.cluster)
print(xtable(km.comb.means[,1:8]),table.placement="H")
```

```
print(xtable(km.comb.means[,8:16]),table.placement="H")
print(xtable(km.comb.means[,17:22],caption = "Cluster means of the variables (K-means)"),table.placement="H")

# mean salary of the players in the two clusters
First.cluster.sal.mean.km<-mean(salaries[players.km==1])
Second.cluster.sal.mean.km<-mean(salaries[players.km==2])
km.comb.salary<-rbind(First.cluster.sal.mean.km,Second.cluster.sal.mean.km)
colnames(km.comb.salary)<-"Mean Salary"
print(xtable(km.comb.salary,caption = "Mean salary of the players in the two clusters (K-means)"),table.placemen



## ----include=FALSE-----------------------------------------
#################################################################################################################
# Question 3

# a)
set.seed(1)
library(caret)
# cross-validation method
ctrl <- trainControl(method = "LOOCV")

#fit a regression model and use LOOCV to evaluate performance
linear.fit <- train(log(Salary)~., data = Hitters.new, method = "lm", trControl = ctrl)

#view summary of LOOCV
MSE_a<-as.numeric(linear.fit$results[2])^2


## ----include=FALSE-----------------------------------------
# b)

library(pls)
# Fit PCR
set.seed(1)
pcr.fit <- pcr(log(Salary) ~ ., data = Hitters, scale = TRUE, validation = "LOO")

# To get MSE
M1<-which.min(MSEP(pcr.fit)$val[1, 1,])
MSEP(pcr.fit)


## ----echo=FALSE,fig.height=2.5, fig.width=3, fig.cap="Validation plots for PCR"----

validationplot(pcr.fit, val.type = "MSEP",main="PCR")


## ----include=FALSE-----------------------------------------
# c)

library(pls)
# Fit PCR
set.seed(1)
pls.fit <- plsr(log(Salary) ~ ., data = Hitters.new, scale = TRUE, validation = "LOO")

# To get MSE
M2<-which.min(MSEP(pls.fit)$val[1, 1,])
MSEP(pls.fit)


## ----echo=FALSE,fig.height=2.5, fig.width=3, fig.cap="Validation plots for PLS"----
validationplot(pls.fit, val.type = "MSEP",main="PLS")
```

```
## ---- include=FALSE-----------------------------------------
# d)

# Create response vector and the design matrix (without the first column of 1s)
y <- log(Hitters.new$Salary)
x <- model.matrix(log(Salary) ~ ., Hitters.new)[, -1]
n<-nrow(Hitters.new)
grid <- 10^seq(10, -2, length = 100)


## ---- include=FALSE-----------------------------------------
# Use cross-validation to estimate test MSE from training data
library(glmnet)
set.seed(1)
cv.out <- cv.glmnet(x,y, alpha = 0,nfolds=n,grouped = FALSE)

# Find the best value of lambda
bestlam <- cv.out$lambda.min
bestlam

# Test MSE for the best value of lambda
ridge.mod <- glmnet(x, y, alpha = 0, lambda = grid)
ridge.pred <- predict(ridge.mod, s = bestlam, newx =x)
d.mse<-mean((ridge.pred - y)^2)


## ----q1ef,echo=FALSE,fig.height=2.5, fig.width=3, fig.cap="Plot of test MSE vs log(lambda) using ridge regress
plot(cv.out)
```