Principal Component Analysis, Principal Component Regression and Partial Least Squares Regression

Consider the Hitters dataset from the ISLR package in R. It consists of 20 variables measured on 263 major league baseball players (after removing those with missing data). Salary is the response variable and the remaining 19 are predictors. Some of the predictor variables are categorical with two classes. The goal is to perform a principal components analysis (PCA) of the data

a) Standardization dataset is necessary as it removes the biases in the original variables. For example, when the data for each variable is collected on different units. The standardized variables will be unitless and have a similar variance.In the given dataset Hitters, the variables are measured in different scales. Hence it is a good idea to standardize the variables.

```
library(ISLR)
str(Hitters)
```

```
'data.frame':   322 obs. of  20 variables:
 $ AtBat    : int  293 315 479 496 321 594 185 298 323 401 ...
 $ Hits     : int  66 81 130 141 87 169 37 73 81 92 ...
 $ HmRun    : int  1 7 18 20 10 4 1 0 6 17 ...
 $ Runs     : int  30 24 66 65 39 74 23 24 26 49 ...
 $ RBI      : int  29 38 72 78 42 51 8 24 32 66 ...
 $ Walks    : int  14 39 76 37 30 35 21 7 8 65 ...
 $ Years    : int  1 14 3 11 2 11 2 3 2 13 ...
 $ CAtBat   : int  293 3449 1624 5628 396 4408 214 509 341 5206 ...
 $ CHits    : int  66 835 457 1575 101 1133 42 108 86 1332 ...
 $ CHmRun   : int  1 69 63 225 12 19 1 0 6 253 ...
 $ CRuns    : int  30 321 224 828 48 501 30 41 32 784 ...
 $ CRBI     : int  29 414 266 838 46 336 9 37 34 890 ...
 $ CWalks   : int  14 375 263 354 33 194 24 12 8 866 ...
 $ League   : Factor w/ 2 levels "A","N": 1 2 1 2 2 1 2 1 2 1 ...
 $ Division : Factor w/ 2 levels "E","W": 1 2 2 1 1 2 1 2 2 1 ...
 $ PutOuts  : int  446 632 880 200 805 282 76 121 143 0 ...
 $ Assists  : int  33 43 82 11 40 421 127 283 290 0 ...
 $ Errors   : int  20 10 14 3 4 25 7 9 19 0 ...
 $ Salary   : num  NA 475 480 500 91.5 750 70 100 75 1100 ...
 $ NewLeague: Factor w/ 2 levels "A","N": 1 2 1 2 2 1 1 1 2 1 ...
```

```
Hitters <- na.omit(Hitters)
dim(Hitters)
```

```
[1] 263  20
```

b) Based on Table 1 it can be observed that the first 4 principle components accounted for 80% of the total variance.4 principle components seem to be explained most of the variance and therefore 4 principle components are recommended.Screeplot for PCs is shown in **Figure** 1 and it confirms the above results.

```
Hitters.X<-Hitters[-19]
Hitters.X$League<-ifelse(Hitters.X$League=="A",0,1)
Hitters.X$Division<-ifelse(Hitters.X$Division=="E",0,1)
Hitters.X$NewLeague<-ifelse(Hitters.X$NewLeague=="A",0,1)
```

```r
#Part b)
pca.Hit <- prcomp(Hitters.X, center = T, scale = T)
```

```r
library(xtable)
options(xtable.comment=FALSE)
xtable(summary(pca.Hit),caption = "Proportion of Variance and Cumulative Proportions")
```

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12
Standard deviation	2.6981	2.0371	1.4249	1.2476	0.9993	0.9085	0.8303	0.7163	0.5007	0.4299	0.3705	0.3570
Proportion of Variance	0.3831	0.2184	0.1069	0.0819	0.0526	0.0435	0.0363	0.0270	0.0132	0.0097	0.0072	0.0067
Cumulative Proportion	0.3831	0.6016	0.7084	0.7903	0.8429	0.8863	0.9226	0.9496	0.9628	0.9726	0.9798	0.9865

Table 1: Proportion of Variance and Cumulative Proportions

```r
par(mar = c(3.8, 3.8,1,1))
screeplot(pca.Hit, type="lines",main="")
```

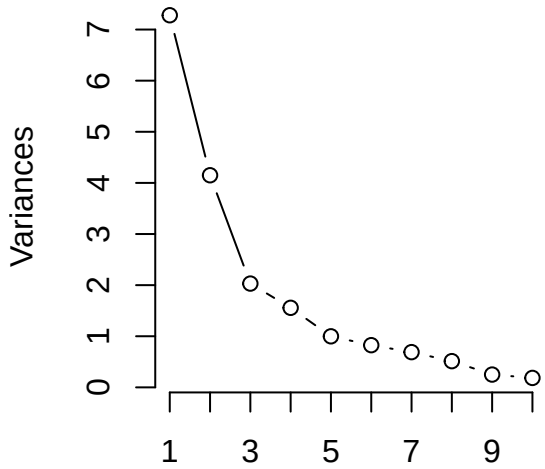

Figure 1: *Scree plot for Principal components of Hitters Data*

```r
options(xtable.comment=FALSE)
print(xtable(pca.Hit$rotation[,1:4],caption = "Loading matrix")
,table.placement="H")
```

	PC1	PC2	PC3	PC4
AtBat	0.20	0.38	-0.09	0.03
Hits	0.20	0.38	-0.07	0.02
HmRun	0.20	0.24	0.22	-0.24
Runs	0.20	0.38	0.02	-0.05
RBI	0.24	0.31	0.07	-0.14
Walks	0.21	0.23	-0.05	-0.13
Years	0.28	-0.26	-0.03	0.10
CAtBat	0.33	-0.19	-0.08	0.09
CHits	0.33	-0.18	-0.09	0.08
CHmRun	0.32	-0.13	0.09	-0.07
CRuns	0.34	-0.17	-0.05	0.07
CRBI	0.34	-0.17	-0.01	0.01
CWalks	0.32	-0.19	-0.04	0.03
League	-0.05	-0.10	-0.55	-0.40
Division	-0.03	-0.04	0.02	0.04
PutOuts	0.08	0.16	-0.05	-0.29
Assists	-0.00	0.17	-0.40	0.52
Errors	-0.01	0.20	-0.38	0.42
NewLeague	-0.04	-0.08	-0.54	-0.42

Table 2: Loading matrix

c) The correlation between the first two principle components and the standardized variables are given in Table 2. According to the **Figure** 2 we can see that variables predictors Years, CHmRun, CRBI are correlated and heavily loaded on PCA1. Moreover predictors like Runs, RB1, Errors, Assist are heavily on PCA2. Also, it can be observed that most of the observations corresponds to a score that ranges from -5 to 5 with respect to both principle components. Observations Ted Simons , Pete Rose seems belongs to PCA1 as those observations corresponds to lower values of PCA2 and higher values of PCA1.

```
# Standardize quantitative variables
std.Hit=scale(Hitters.X)
cor.Hit=cor(pca.Hit$x[,c(1,2)],std.Hit)
xtable(cor.Hit[,1:8],digits=c(0,4,4,4,4,4,4,4,4))
```

	AtBat	Hits	HmRun	Runs	RBI	Walks	Years	CAtBat
PC1	0.5350	0.5285	0.5514	0.5351	0.6345	0.5637	0.7624	0.8916
PC2	0.7818	0.7685	0.4831	0.7695	0.6407	0.4677	-0.5345	-0.3930

```
xtable(cor.Hit[,9:16],digits=c(0,4,4,4,4,4,4,4,4), caption = "Correlation of the standardized quantitative varia
```

	CHits	CHmRun	CRuns	CRBI	CWalks	League	Division	PutOuts
PC1	0.8924	0.8606	0.9125	0.9183	0.8548	-0.1470	-0.0694	0.2096
PC2	-0.3726	-0.2573	-0.3509	-0.3424	-0.3918	-0.1940	-0.0747	0.3173

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

```
par(mar = c(3.8, 3.8,1,1))
# Display a biplot the results (shows both pc scores and loading vectors)
biplot(pca.Hit, scale=0)
```

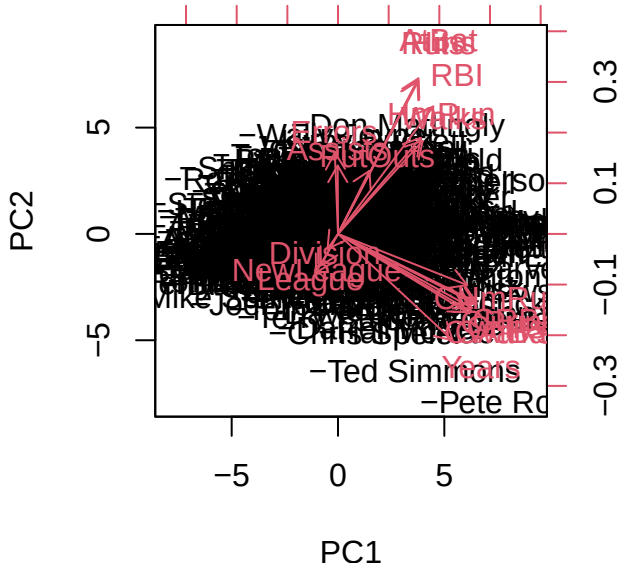

Figure 2: *Biplots for PCA of Hitters Data*

d) Multiple linear regression model using all predictors was performed and calculated test MSE using LOOCV.

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

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

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

e) PCR was performed and M chosen optimally via LOOCV. $M = 17$ was chosen as the optimal M as it gives the lowest MSE. **Figure** 3 shows the validation plot.

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

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

```
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV          0.7937   0.4245   0.4271   0.4236    0.424   0.4193   0.4215
adjCV       0.7937   0.4245   0.4271   0.4236    0.424   0.4192   0.4215
       7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
CV      0.4205   0.4164   0.4196    0.4181    0.4237    0.4203    0.4271
adjCV   0.4204   0.4164   0.4195    0.4180    0.4236    0.4202    0.4269
       14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
CV       0.4176    0.4211    0.4104    0.4138    0.4176    0.4214
adjCV    0.4174    0.4210    0.4103    0.4137    0.4175    0.4213
```

f) PLS was performed and M chosen optimally via LOOCV. $M = 13$ was chosen as the optimal M as it gives the lowest MSE. **Figure** 3 shows the validation plot.

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

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

```
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV          0.7937   0.4198   0.4195   0.4186   0.4172   0.4176   0.4176
adjCV       0.7937   0.4198   0.4195   0.4185   0.4171   0.4175   0.4175
       7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
CV      0.4191   0.4193   0.4149    0.4145    0.4138    0.4135    0.4148
adjCV   0.4189   0.4192   0.4148    0.4144    0.4136    0.4134    0.4146
       14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
CV       0.4166    0.4169    0.4165    0.4255    0.4236    0.4214
adjCV    0.4164    0.4167    0.4163    0.4253    0.4234    0.4213
```

```
par(mfrow=c(1,2))
par(mar = c(3.8, 3.8,1,1))
validationplot(pcr.fit, val.type = "MSEP",main="PCR")
validationplot(pls.fit, val.type = "MSEP",main="PLS")
```

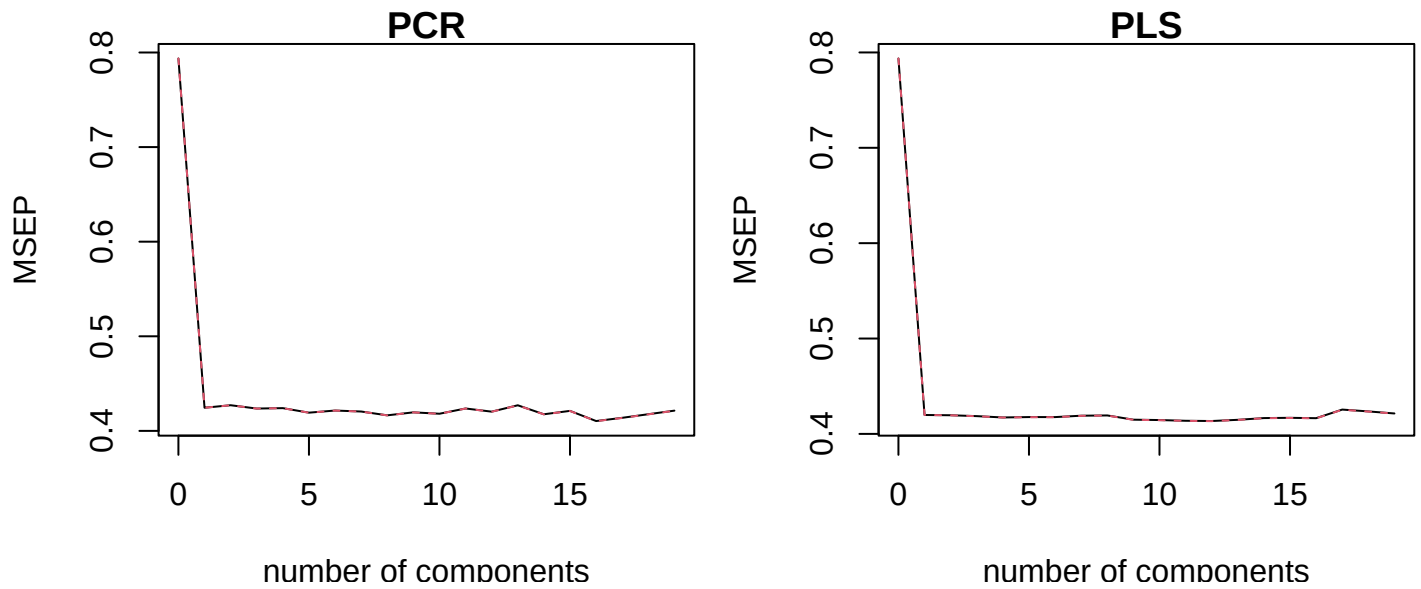

Figure 3: *Validation plots for PCR and PLS*

g) Ridge regression was performed and penalty parameter chosen optimally via LOOCV. Best λ value is 0.07286764. **Figure** 4 shows plot of test MSE vs $\log(\lambda)$.

```
#Part d)
# Create response vector and the design matrix (without the first column of 1s)
y <- log(Hitters$Salary)
```

```r
x <- model.matrix(log(Salary) ~ ., Hitters)[, -1]
n<-nrow(Hitters)
grid <- 10^seq(10, -2, length = 100)
```

```r
# 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
```

```
[1] 0.07286764
```

```r
# 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)
```

```r
par(mar = c(3.8, 3.8,1,1))
plot(cv.out)
```

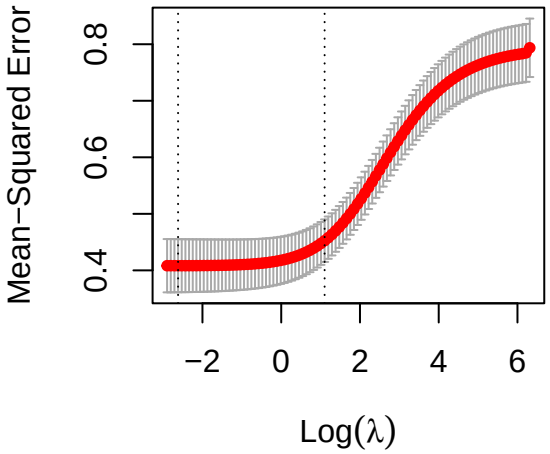

Figure 4: *plot of test MSE vs log(lambda) using ridge regression*

h) According to the following table Ridge regression model has the lowest MSE. Moreover PCR and PSL gives fairly similar MSE rate and parameter estimates. Therefore we can select model chosen from the ridge regression as the best model.

	Linear regression	PCR	PLS	Ridge regression
Test MSE	0.4214	0.4138	0.4148	0.3607

Table 4: *Summary of test MSE*