Lab 6

1

```
Hitters <- Hitters
data.frame(colMeans(is.na(Hitters)))
```

```
##           colMeans.is.na.Hitters..
## AtBat                    0.0000000
## Hits                     0.0000000
## HmRun                    0.0000000
## Runs                     0.0000000
## RBI                      0.0000000
## Walks                    0.0000000
## Years                    0.0000000
## CAtBat                   0.0000000
## CHits                    0.0000000
## CHmRun                   0.0000000
## CRuns                    0.0000000
## CRBI                     0.0000000
## CWalks                   0.0000000
## League                   0.0000000
## Division                 0.0000000
## PutOuts                  0.0000000
## Assists                  0.0000000
## Errors                   0.0000000
## Salary                   0.1832298
## NewLeague                0.0000000
```

```
Hitters <- Hitters %>% na.omit()
data.frame(colMeans(is.na(Hitters)))
```

```
##           colMeans.is.na.Hitters..
## AtBat                            0
## Hits                             0
## HmRun                            0
## Runs                             0
## RBI                              0
## Walks                            0
## Years                            0
## CAtBat                           0
## CHits                            0
## CHmRun                           0
## CRuns                            0
## CRBI                             0
## CWalks                           0
## League                           0
## Division                         0
## PutOuts                          0
## Assists                          0
## Errors                           0
## Salary                           0
```

```
## NewLeague                          0
```

```
design.m <- model.matrix(Salary~., data = Hitters) # eliminating the constant column
X <- scale(design.m[,-1])
y <- Hitters$Salary

X.svd <- svd(X)
Z <- X %*% X.svd$v

pcr_fit <- pcr(Salary ~ ., data = Hitters, scale = TRUE, validation = "none")
colMeans((abs(pcr_fit$scores) - abs(Z)) <= 0.1^5)
```

```
##  Comp 1  Comp 2  Comp 3  Comp 4  Comp 5  Comp 6  Comp 7  Comp 8  Comp 9
##       1       1       1       1       1       1       1       1       1
## Comp 10 Comp 11 Comp 12 Comp 13 Comp 14 Comp 15 Comp 16 Comp 17 Comp 18
##       1       1       1       1       1       1       1       1       1
## Comp 19
##       1
```

2

```
z1 <- Z[,1]
b1 <- solve(t(z1) %*% z1)%*%t(z1) %*% y
y.hat.1 <- b1 %*% z1

sum(((y.hat.1 + mean(y)) - pcr_fit$fitted.values[,,1]))
```

```
## [1] -6.82121e-13
```

3

```
b.pcr <- solve(t(Z) %*% Z) %*% t(Z) %*% y
y.hat <- Z %*% b.pcr

sum((y.hat + mean(y)) - pcr_fit$fitted.values[,,19])
```

```
## [1] -1.017497e-11
```

4

```
b1.star <- X.svd$v[,1] %*% t(X.svd$u[,1]) %*% y/X.svd$d[1]

sum(b1.star - as.numeric((b1%*%X.svd$v[,1])))
```

```
## [1] 1.151093e-12
```

Partial Least Squares Regression

```
#remove observations with missing salaries from Hitters
Hitters <- Hitters %>% na.omit()
data.frame(colMeans(is.na(Hitters)))
```

```
##           colMeans.is.na.Hitters..
## AtBat                            0
## Hits                             0
## HmRun                            0
## Runs                             0
## RBI                              0
## Walks                            0
## Years                            0
## CAtBat                           0
## CHits                            0
## CHmRun                           0
## CRuns                            0
## CRBI                             0
## CWalks                           0
## League                           0
## Division                         0
## PutOuts                          0
## Assists                          0
## Errors                           0
## Salary                           0
## NewLeague                        0
```

```
#generating matrix X
design.m <- model.matrix(Salary~., data = Hitters) # eliminating the constant column
X.plsr <- scale(design.m[,-1])
#generating vector y
y.plsr <- scale(Hitters$Salary)
```

```
normalize <- function(x) {x / sqrt(sum(x^2))}
```

```
X.temp <- X.plsr
y.temp <- y.plsr
for (h in 1:1) {
  w1 <- t(X.temp) %*% y.temp
  w1 <- normalize(w1)
  z1 <- (X.temp  %*% w1)/as.numeric(t(w1) %*% w1)
  p1 <- (t(X.temp) %*% z1)/as.numeric(t(z1) %*% z1)
  #updating X
  X.temp <- X.temp - z1 %*% t(p1)
  b1 <- (t(y.temp) %*% z1)/as.numeric(t(z1) %*% z1)
  #updating Y
  y.temp <- y.temp - as.numeric(b1) * z1
}
```

```
pls.fit <- plsr(y.plsr~X.plsr)
sum(abs(pls.fit$loading.weights[,1] - w1))
```

```
## [1] 7.218618e-16
```

```r
sum(abs(pls.fit$scores[,1] - z1))
```

```
## [1] 1.035257e-13
```

```r
sum(abs(pls.fit$loadings[,1] - p1))
```

```
## [1] 7.225123e-16
```

```r
plsr.custom <- function(X, y) {
  #### initalizing
  r <- ncol(X)
  X.temp <- X
  y.temp <- y
  weights <- vector()
  scores <- vector()
  loading <- vector()
  residuals <- vector()
  coefficients <- as.numeric()
  #### computation
  for (h in 1:r) {
    w.temp <- t(X.temp) %*% y.temp
    w.temp <- normalize(w.temp)
    #
    weights <- cbind(weights, w.temp)
    z.temp <- (X.temp  %*% w.temp)/as.numeric(t(w.temp) %*% w.temp)
    #
    scores <- cbind(scores, z.temp)
    p.temp <- (t(X.temp) %*% z.temp)/as.numeric(t(z.temp) %*% z.temp)
    View(p.temp)
    #
    loading <- cbind(loading, p.temp)
    #updating X
    X.temp <- X.temp - z.temp %*% t(p.temp)
    b.temp <- (t(y.temp) %*% z.temp)/as.numeric(t(z.temp) %*% z.temp)
    #
    coefficients[h] <- as.numeric(b.temp)
    #updating Y
    y.temp <- y.temp - as.numeric(b.temp) * z.temp
    #
    residuals <- cbind(residuals, y.temp)
  }
  return(list(Weights=weights,
              Scores=scores,
              Loadings=loading,
              Residuals=residuals,
              Coefficients=coefficients))
}
```