PCR and PLS

AUTHOR
Pan Phyu Phyu Hmwe

Principal Component Regression (PCR)

The College dataset provides information about U.S. colleges and universities, including various metrics that describe their characteristics. The data were likely collected from publicly available records and surveys conducted by educational institutions.

We are using Principal Component Regression (PCR) on the College dataset to understand and model the factors influencing the graduation rate (Grad.Rate) of U.S. colleges and universities. By applying PCR, we aim to address potential challenges with high-dimensional data, such as multicollinearity, while reducing the complexity of the model.

```
        library(ISLR2)
```

```
Warning: package 'ISLR2' was built under R version 4.2.3
```

```
        library(dplyr)
```

```
Warning: package 'dplyr' was built under R version 4.2.3
```

```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
        library(tidyr)
```

```
Warning: package 'tidyr' was built under R version 4.2.3
```

```
        #load the dataset
        data("College")
        college.df <- College
        head(college.df)
```

```
                           Private Apps Accept Enroll Top10perc Top25perc
Abilene Christian University     Yes 1660   1232    721        23        52
```

```
Adelphi University               Yes 2186   1924    512        16        29
Adrian College                   Yes 1428   1097    336        22        50
Agnes Scott College              Yes  417    349    137        60        89
Alaska Pacific University        Yes  193    146     55        16        44
Albertson College                Yes  587    479    158        38        62
                           F.Undergrad P.Undergrad Outstate Room.Board Books
Abilene Christian University        2885         537     7440       3300   450
Adelphi University                  2683        1227    12280       6450   750
Adrian College                      1036          99    11250       3750   400
Agnes Scott College                  510          63    12960       5450   450
Alaska Pacific University            249         869     7560       4120   800
Albertson College                    678          41    13500       3335   500
                           Personal PhD Terminal S.F.Ratio perc.alumni Expend
Abilene Christian University     2200  70       78      18.1          12   7041
Adelphi University               1500  29       30      12.2          16  10527
Adrian College                   1165  53       66      12.9          30   8735
Agnes Scott College               875  92       97       7.7          37  19016
Alaska Pacific University        1500  76       72      11.9           2  10922
Albertson College                 675  67       73       9.4          11   9727
                           Grad.Rate
Abilene Christian University      60
Adelphi University                56
Adrian College                    54
Agnes Scott College               59
Alaska Pacific University         15
Albertson College                 55
```

We are converting Private to a numeric variable for regression.

```r
# Convert 'Private' to character
college.df$Private <- as.character(college.df$Private)

#Convert 'Private' to a numeric variable
college.df$Private <- ifelse(college.df$Private == "Yes", 1, 0)

head(college.df)
```

```
                           Private Apps Accept Enroll Top10perc Top25perc
Abilene Christian University       1 1660   1232    721        23        52
Adelphi University                 1 2186   1924    512        16        29
Adrian College                     1 1428   1097    336        22        50
Agnes Scott College                1  417    349    137        60        89
Alaska Pacific University          1  193    146     55        16        44
Albertson College                  1  587    479    158        38        62
                           F.Undergrad P.Undergrad Outstate Room.Board Books
Abilene Christian University        2885         537     7440       3300   450
Adelphi University                  2683        1227    12280       6450   750
Adrian College                      1036          99    11250       3750   400
Agnes Scott College                  510          63    12960       5450   450
Alaska Pacific University            249         869     7560       4120   800
Albertson College                    678          41    13500       3335   500
```

```
                           Personal PhD Terminal S.F.Ratio perc.alumni Expend
Abilene Christian University     2200  70       78      18.1          12   7041
Adelphi University               1500  29       30      12.2          16  10527
Adrian College                   1165  53       66      12.9          30   8735
Agnes Scott College               875  92       97       7.7          37  19016
Alaska Pacific University        1500  76       72      11.9           2  10922
Albertson College                 675  67       73       9.4          11   9727
                           Grad.Rate
Abilene Christian University        60
Adelphi University                  56
Adrian College                      54
Agnes Scott College                 59
Alaska Pacific University           15
Albertson College                   55
```

Then, we will fit the PCR model using cross-validation using Grad.Rate as a response variable. Scale is set to true to ensure that all predictors are on the same scale.

```
        library(pls)
```

```
Attaching package: 'pls'

The following object is masked from 'package:stats':

    loadings
```

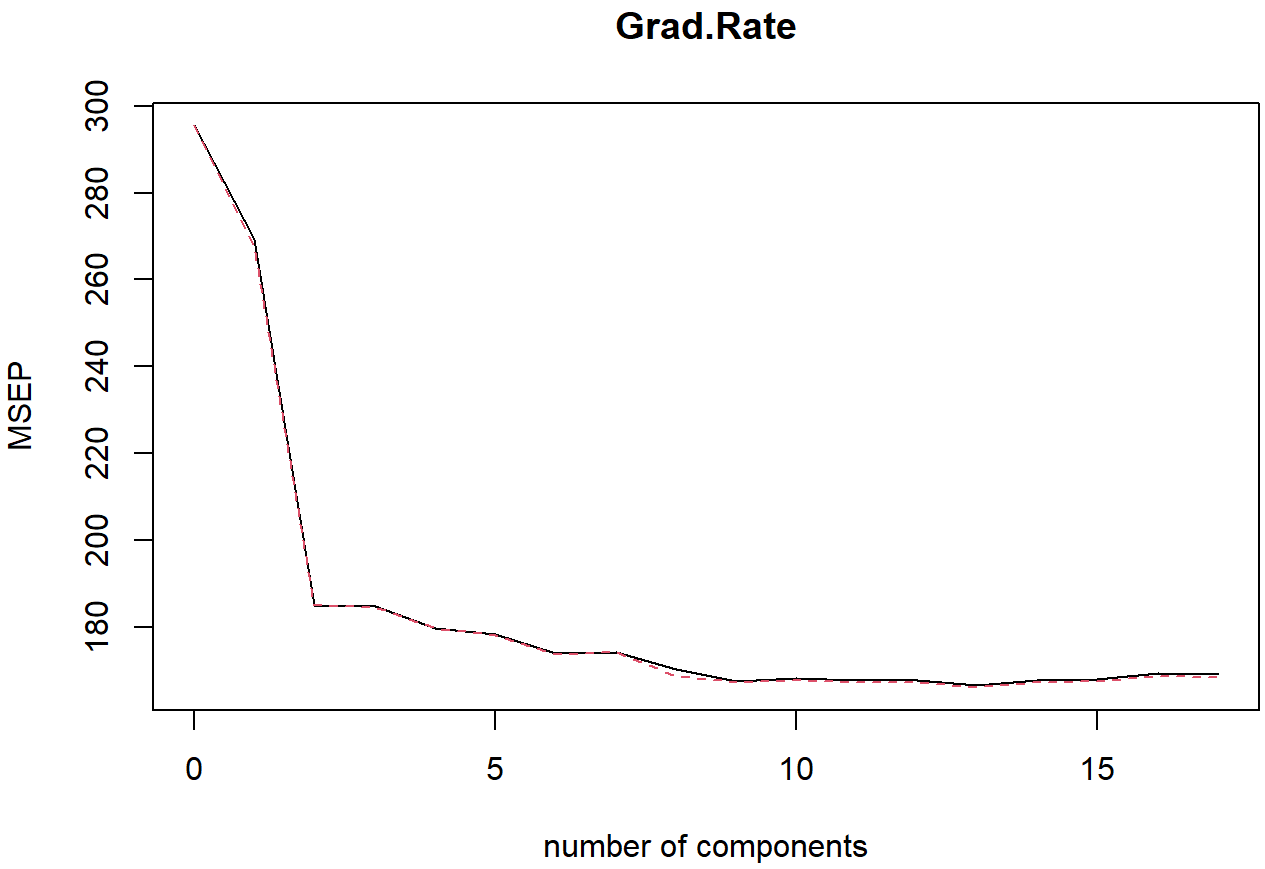
```
        set.seed(1)

        # Fit PCR model using cross-validation
        pcr_model <- pcr(Grad.Rate ~ ., data = college.df,scale = TRUE, validation = "CV")

        # Summary to see cross-validation results
        summary(pcr_model)
```

```
Data:   X dimension: 777 17
    Y dimension: 777 1
Fit method: svdpc
Number of components considered: 17

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV           17.19    16.41     13.6    13.59    13.41    13.35    13.19
adjCV        17.19    16.36     13.6    13.59    13.40    13.35    13.19
       7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
CV        13.2    13.05    12.95     12.97     12.95     12.95     12.91
adjCV     13.2    12.99    12.94     12.96     12.94     12.94     12.89
       14 comps  15 comps  16 comps  17 comps
CV        12.96     12.97     13.02     13.01
```

```
adjCV      12.94     12.95     12.99     12.98

TRAINING: % variance explained
           1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
X            30.61    59.36    66.26    72.24    77.67    82.50    85.96
Grad.Rate    10.58    37.51    38.17    40.03    40.68    42.21    42.23
           8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
X            89.13    92.25     94.36     96.19     97.29     98.29     99.14
Grad.Rate    44.31    44.88     44.91     45.15     45.51     45.80     45.91
           15 comps  16 comps  17 comps
X             99.65     99.86    100.00
Grad.Rate     45.91     45.95     46.15
```

We will now plot the cross validation scores to choose the optimal number of components.

```r
# Plot cross-validation MSE
validationplot(pcr_model, val.type = "MSEP")
```

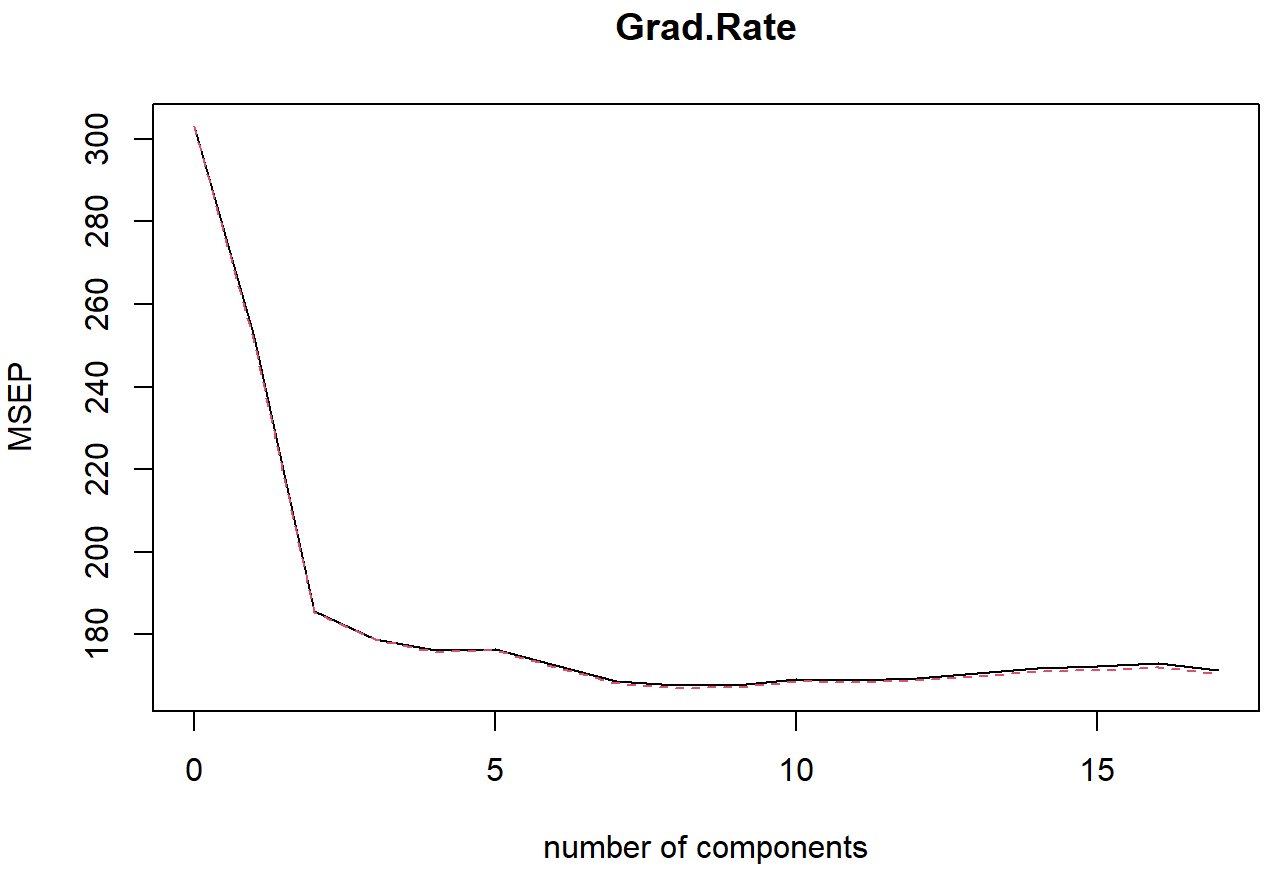


The cross-validation plot shows a sharp decrease in MSEP from 1 to 3 components, suggesting these capture key patterns. Beyond 3, the MSEP levels off, indicating diminishing returns. From the summary, M = 1 explains 30.61% of predictor variance and 10.58% of Grad.Rate, while M = 7 captures 85.96% of predictor variance and 42.23% of Grad.Rate. Using M = 17 (all components) explains 100% of the predictors' variance but only 46.15% of Grad.Rate. Therefore, 3 to 7 components offer a good balance between simplicity and accuracy.

We will now do the PCR on training data and make a plot to assess its performance.

```
set.seed(1)

# Split the College data into training and testing sets (50% each)
train <- college.df %>%
  sample_frac(0.5)

test <- college.df %>%
  setdiff(train)
```

```
# Fit PCR model on training data with cross-validation
pcr_fit2 <- pcr(Grad.Rate ~ ., data = train, scale = TRUE, validation = "CV")

# Plot the cross-validation MSEP to choose the optimal number of components
validationplot(pcr_fit2, val.type = "MSEP")
```



The optimal number of components, M, for PCR model appears to be 5, as this is where the cross-validation error is minimized. Adding more components beyond this point does not provide a significant reduction in error, suggesting that 5 components will strike a good balance between model complexity and predictive accuracy.

We implement the test MSE using M=5 as below:

```r
# Create model matrices for PCR
x_train <- model.matrix(Grad.Rate ~ ., train)[, -1]  # Exclude intercept
x_test <- model.matrix(Grad.Rate ~ ., test)[, -1]    # Exclude intercept

# Extract response variables
y_train <- train %>%
  select(Grad.Rate) %>%
  unlist() %>%
  as.numeric()

y_test <- test %>%
  select(Grad.Rate) %>%
  unlist() %>%
  as.numeric()

# Fit PCR model with the optimal number of components identified from cross-validation (
pcr_pred <- predict(pcr_fit2, x_test, ncomp = 5)

mean((pcr_pred - y_test)^2)
```

```
[1] 181.3046
```

The test Mean Squared Error (MSE) of 181.3046 indicates the average squared difference between the predicted and actual Grad.Rate values on the test set, reflecting the performance of PCR model.

Since we have tested the performance of the test set, we will do the PCR fitting on a full dataset again using 5 components.

```r
# Create model matrix for the full dataset
x_full <- model.matrix(Grad.Rate ~ ., college.df)[, -1]

# Extract the response variable for the full dataset
y_full <- college.df %>%
  select(Grad.Rate) %>%
  unlist() %>%
  as.numeric()

# Fit PCR model on the full dataset using M = 5 components
pcr_full_fit <- pcr(y_full ~ x_full, scale = TRUE, ncomp = 5)

# Summary of the final model
summary(pcr_full_fit)
```

```
Data:   X dimension: 777 17
    Y dimension: 777 1
Fit method: svdpc
Number of components considered: 5
```

```
TRAINING: % variance explained
        1 comps  2 comps  3 comps  4 comps  5 comps
X         30.61    59.36    66.26    72.24    77.67
y_full    10.58    37.51    38.17    40.03    40.68
```

The final PCR model with 5 components provides a reduced-dimension view of the original data, retaining 77.67% of the information from the predictors and explaining 40.68% of the variability in Grad.Rate. This balance between dimensionality reduction and explanatory power is achieved without using all 17 original predictors, supporting a simpler, more interpretable model.

Partial Least Squares (PLS)

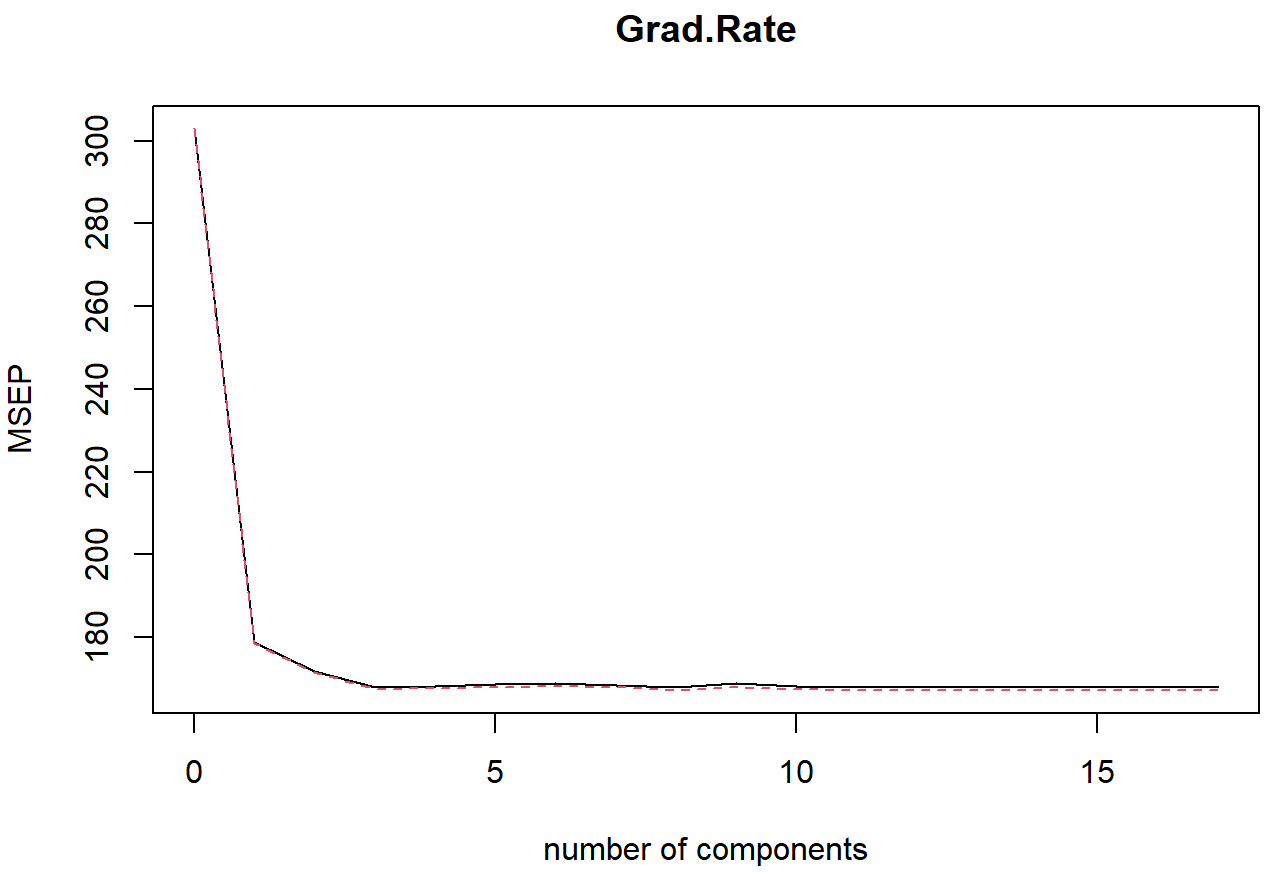
Performing Partial Least Squares (PLS) Regression would be a logical next step to complement PCR analysis. While PCR focuses on reducing the dimensionality of the predictors without directly considering the response variable, PLS aims to find components that maximize the covariance between the predictors and the response, potentially leading to better predictive performance.

```r
# Fit PLS model using cross-validation
pls_fit <- plsr(Grad.Rate ~ ., data = train, scale = TRUE, validation = "CV")

# Summary to see cross-validation results
summary(pls_fit)
```

```
Data:   X dimension: 388 17
    Y dimension: 388 1
Fit method: kernelpls
Number of components considered: 17

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV           17.41    13.37    13.11    12.96    12.97    12.98    13.00
adjCV        17.41    13.36    13.10    12.94    12.95    12.96    12.97
       7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
CV       12.98    12.95    12.99     12.97     12.96     12.96     12.96
adjCV    12.95    12.93    12.96     12.94     12.93     12.93     12.93
       14 comps  15 comps  16 comps  17 comps
CV        12.96     12.96     12.96     12.96
adjCV     12.93     12.93     12.93     12.93

TRAINING: % variance explained
           1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
X            29.78    49.71    65.49    70.22    74.28    77.61    81.56
Grad.Rate    41.86    45.26    47.33    47.84    47.99    48.10    48.17
           8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
X            83.35    86.05     89.27     92.17     93.34     95.36     96.69
Grad.Rate    48.34    48.46     48.52     48.53     48.53     48.53     48.53
           15 comps  16 comps  17 comps
```

```
X            97.72     99.01    100.00
Grad.Rate    48.53     48.53     48.53
```

```r
# Plot cross-validation MSE to choose the optimal number of components
validationplot(pls_fit, val.type = "MSEP")
```



From the cross-validation plot and the summary, we can conclude that the optimal number of components for the PLS model is 3. These components effectively balance predictive accuracy with model simplicity, capturing 65.49% of the variance in the predictors and 47.33% of the variance in Grad.Rate. Adding more components does not significantly reduce the RMSEP or improve the variance explained, indicating that 3 components should be sufficient for a well-performing and interpretable model.

```r
# Make predictions on the test set using 3 components
pls_pred = predict(pls_fit, x_test, ncomp = 3)

# Calculate Test Mean Squared Error (MSE) for PLS
mean((pls_pred - y_test)^2)
```

```
[1] 170.0129
```

The MSE of 170.0129 reflects the average squared difference between the predicted and actual Grad.Rate values on the test set. Since this is a measure of prediction error, a lower value indicates better predictive performance. If we compare this PLS Test MSE (170.0129) to the Test MSE obtained from PCR model

(181.3046), we can see that the PLS model has a lower MSE. This suggests that PLS is slightly better at predicting Grad.Rate than PCR, likely because PLS directly considers the response variable when deriving the components.

Now using M=3 and fitting the PLS on a full dataset

```
# Fit PLS model on the full dataset using the optimal number of components (3)
pls_full_fit <- plsr(Grad.Rate ~ ., data = college.df, scale = TRUE, ncomp = 3)

# Summary of the final model
summary(pls_full_fit)
```

```
Data:   X dimension: 777 17
    Y dimension: 777 1
Fit method: kernelpls
Number of components considered: 3
TRAINING: % variance explained
           1 comps  2 comps  3 comps
X            29.19    39.27    64.21
Grad.Rate    39.78    44.31    45.16
```

The final PLS model using 3 components achieves a good balance between simplifying the predictors and retaining explanatory power. It captures 64.21% of the variance in the predictors and 45.16% of the variance in Grad.Rate. Compared to PCR, PLS appears to offer better predictive performance (as seen by the lower Test MSE of 170.0129), making it the preferred approach for modeling graduation rates in this dataset.

The results suggest that these 3 components efficiently summarize the original data while maintaining a substantial amount of information, allowing for better predictions of Grad.Rate. This makes the final PLS model a solid choice for understanding and predicting graduation rates based on the available college data.