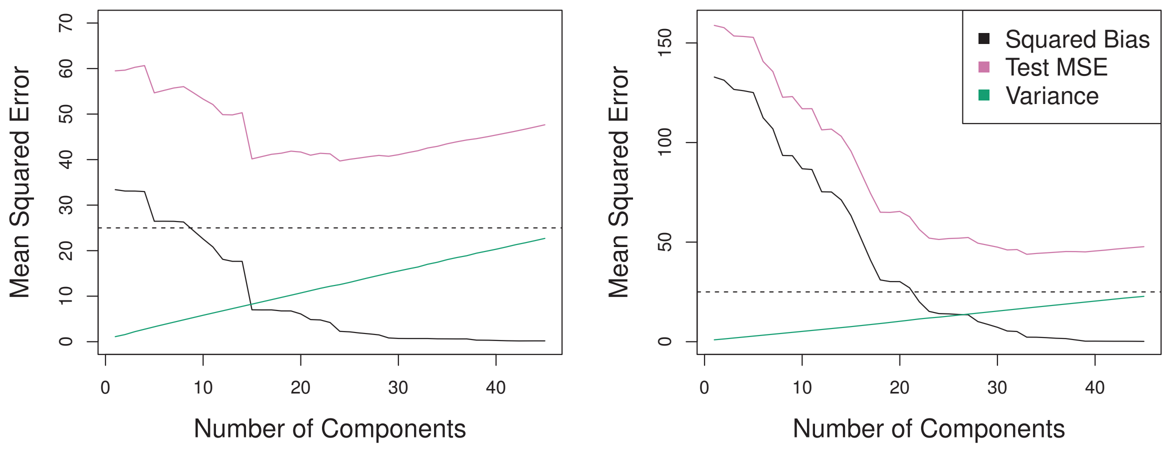
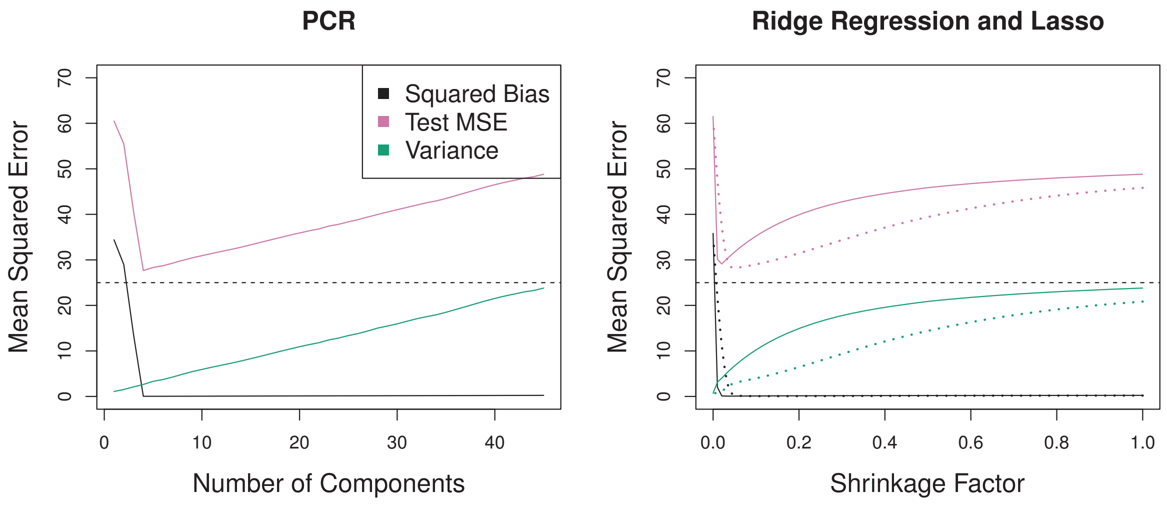
- In the advertising data, the first principal component explains most of the variance in both pop and ad, so a principal component regression that uses this single variable to predict some response of interest, such as sales, will likely perform quite well.



PCR was applied to two simulated data sets with $n = 50$ observations, $p = 45$ predictors. In each panel, the horizontal dashed line represents the irreducible error. Left: All predictors are related to the response. Right: Only 2 out of 45 predictors are related to the response.

- PCR does not perform as well as the two shrinkage methods (the ridge regression and lasso) in this example.
- The relatively worse performance of PCR in the figure above is a consequence of the fact that the data were generated in such a way that many principal components are required in order to adequately model the response.
- In contrast, PCR will tend to do well in cases when the first few principal components are sufficient to capture most of the variation in the predictors as well as the relationship with the response.

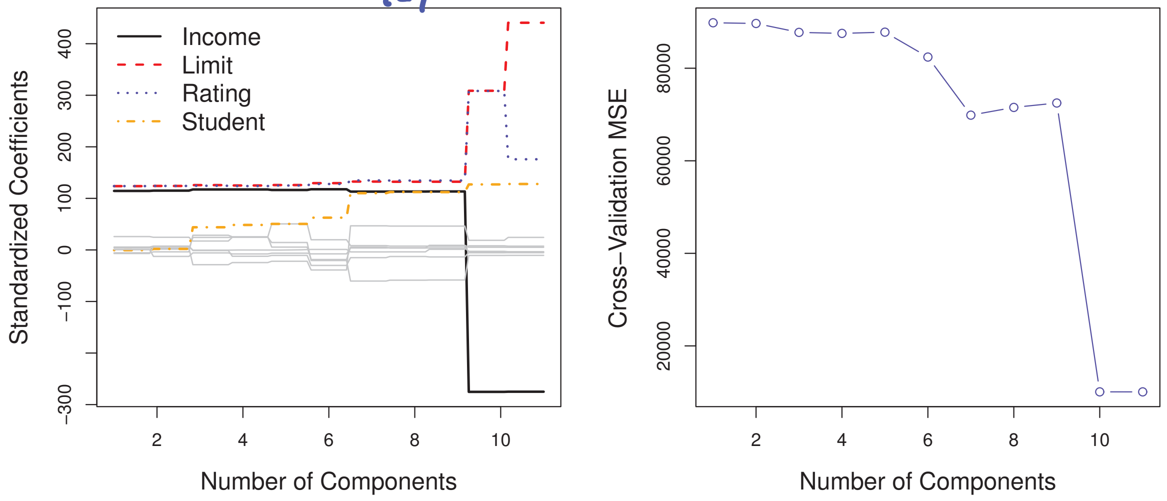


PCR, ridge regression, and the lasso were applied to a simulated data set in which the first five principal components of X contain all the information about the response Y. In each panel, the irreducible error $Var(\epsilon)$ is shown as a horizontal dashed line. Left: Results for PCR. Right: Results for lasso (solid) and ridge regression (dotted). The x-axis displays the shrinkage factor of the coefficient estimates, defined as the ℓ_2 norm of the shrunken coefficient estimates divided by the ℓ_2 norm of the least squares estimate.

- Note that even though PCR provides a simple way to perform regression using $M < p$ predictors, it is not a feature selection method. This is because each of the M principal components used in the regression is a linear combination of all p of the original features.
- Therefore, while PCR often performs quite well in many practical settings, it does not result in the development of a model that relies upon a small set of the original features.
- In this sense, PCR is more closely related to ridge regression than to the lasso.
- In PCR, the number of principal components, M, is typically chosen by cross-validation.

when performing PCR, we need to standardize each predictor using

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)^2}}$$ prior to generating the principal components.



Left: PCR standardized coefficient estimates on the Credit data set for different values of M. Right: The ten-fold cross-validation MSE obtained using PCR, as a function of M.

Performing PCR in R

Principal components regression (PCR) can be performed using the pcr() function, which is part of the pls library. We now apply PCR to the Credit data in order to predict Balance.

```
> library(ISLR2)
> library(pls)
> names(Credit)
 [1] "Income"    "Limit"     "Rating"    "Cards"     "Age"
 [6] "Education" "Own"       "Student"   "Married"   "Region"
[11] "Balance"
> dim(Credit)
[1] 400  11
> set.seed(1)
> pcr_fit <- pcr(Balance ~ ., data = Credit, scale = TRUE, validation = "CV")
```

Setting validation = ”CV” causes pcr() to compute the ten-fold cross-validation error for each possible value of M, the number of principal components used.

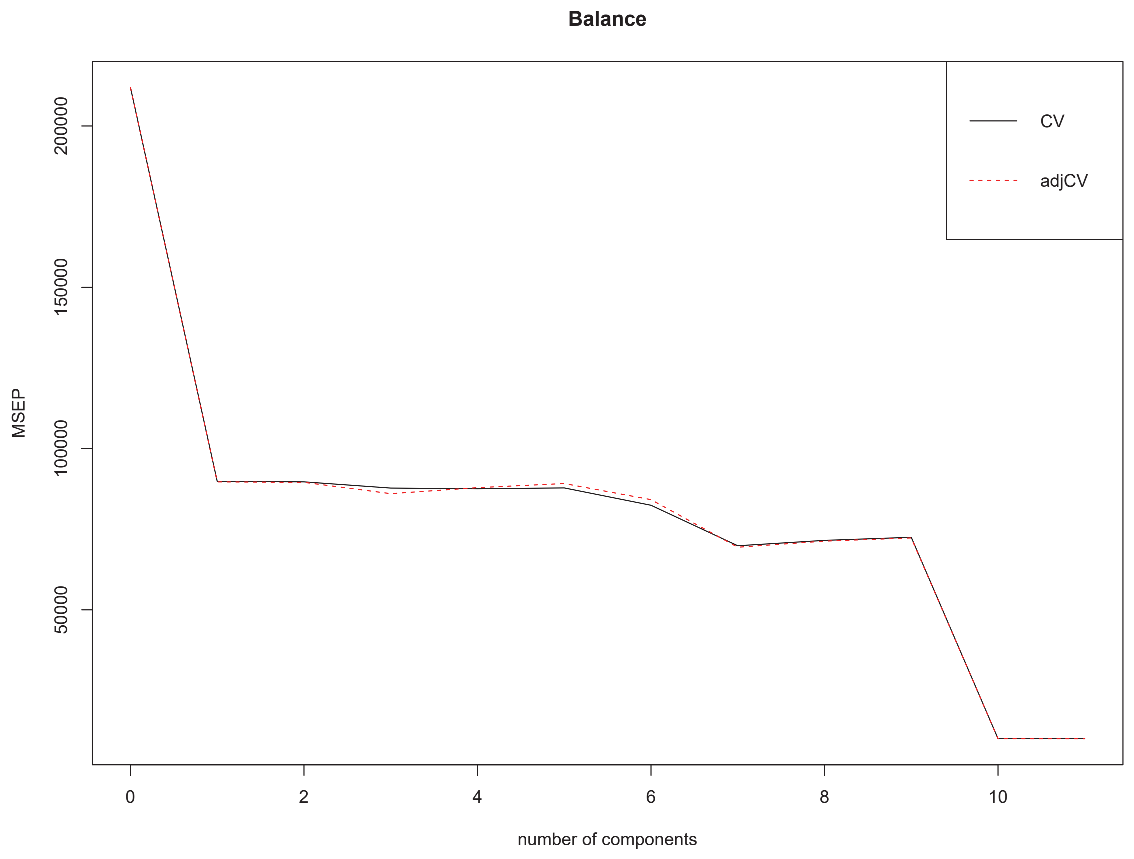
The resulting fit can be examined using summary().

```
> summary(pcr_fit)
Data:  X dimension: 400 11
Y dimension: 400 1
Fit method: svdpc
Number of components considered: 11

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps
CV           460.3    299.7    299.5    296.2    295.9    296.3
adjCV        460.3    299.4    299.2    293.3    296.4    298.6
       6 comps  7 comps  8 comps  9 comps  10 comps  11 comps
CV       287.1    264.3    267.5    269.2     100.3     100.3
adjCV    290.1    263.5    267.0    268.9     100.2     100.2

TRAINING: % variance explained
         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
X          25.05    39.64    49.73    59.74    68.89    77.73
Balance    58.07    58.37    60.78    60.90    61.46    63.11
         7 comps  8 comps  9 comps  10 comps  11 comps
X          86.43    93.91    97.60     99.98    100.00
Balance    68.70    68.71    68.72     95.47     95.51
```

The CV score is provided for each possible number of components, ranging from M = 0 onwards. Note that pcr() reports the root mean squared error; in order to obtain the usual MSE, we must square this quantity.

One can also plot the cross-validation scores using the validationplot() function. Using val.type = "MSEP" will cause the cross-validation MSE to be plotted.

```
> validationplot(pcr_fit, val.type = "MSEP", legendpos = "topright")
```

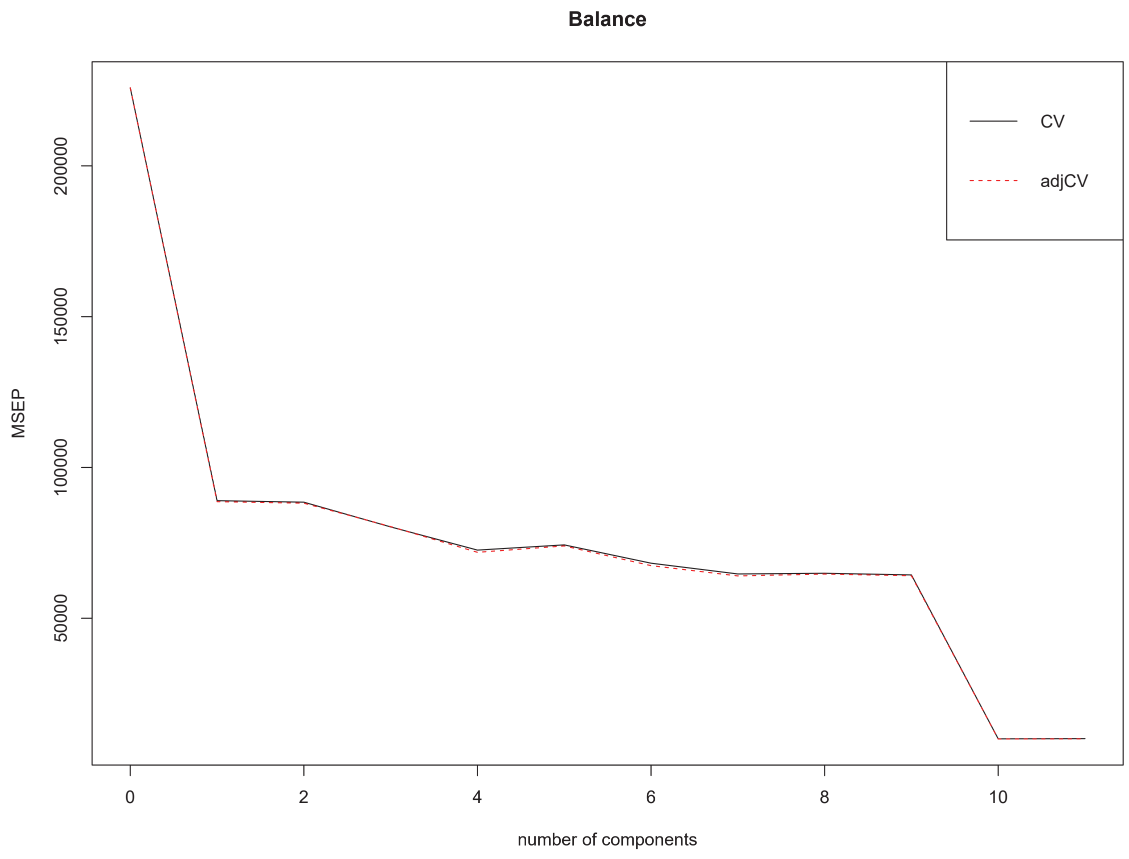


We see that the smallest cross-validation error occurs when $M = 10$ components are used. This is barely fewer than $M = 11$, which amounts to simply performing least squares, because when all of the components are used in PCR no dimension reduction occurs.

The summary() function also provides the percentage of variance explained in the predictors and in the response using different numbers of components.

We now perform PCR on the training data and evaluate its test set performance.

```
> x <- model.matrix(Balance ~ ., Credit)[, -1]
> y <- Credit$Balance
> dim(x)
[1] 400  11
> set.seed(3)
> train <- sample(1:nrow(x), nrow(x) / 2)
> test <- (-train)
> y_test <- y[test]
> pcr_fit <- pcr(Balance ~ ., data = Credit, subset = train, scale = TRUE,
  validation = "CV")
> summary(pcr_fit)
Data:  X dimension: 200 11
Y dimension: 200 1
Fit method: svdpc
Number of components considered: 11

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps
CV           475.3    298.3    297.5    283.4    269.4    272.6
adjCV        475.3    297.8    296.9    283.6    268.1    272.0
       6 comps  7 comps  8 comps  9 comps  10 comps  11 comps
CV       261.3    254.3    254.8    253.7    100.17     100.5
adjCV    259.7    253.1    254.3    253.2     99.87     100.2

TRAINING: % variance explained
         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
X          25.92    40.60    51.24    61.16    70.15    78.72
Balance    60.86    61.33    65.62    69.34    70.03    72.04
         7 comps  8 comps  9 comps  10 comps  11 comps
X          86.96    94.18    98.13     99.98    100.00
Balance    73.72    73.72    74.07     96.01     96.04
> validationplot(pcr_fit, val.type = "MSEP", legendpos = "topright")
```



We find that the lowest cross-validation error occurs when $M = 10$ components are used. We compute the test MSE as follows.

```
> pcr_pred <- predict(pcr_fit, x[test, ], ncomp = 10)
> mean((pcr_pred - y_test)^2)
[1] 10652.07
```

Finally, we fit PCR on the full data set, using $M = 10$, the number of components identified by cross-validation.

```
> pcr_fit <- pcr(y ~ x, scale = TRUE, ncomp = 10)
> summary(pcr_fit)
Data:  X dimension: 400 11
Y dimension: 400 1
Fit method: svdpc
Number of components considered: 10
```

```
TRAINING: % variance explained
   1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
X    25.05    39.64    49.73    59.74    68.89    77.73    86.43
y    58.07    58.37    60.78    60.90    61.46    63.11    68.70
   8 comps  9 comps  10 comps
X    93.91    97.60     99.98
y    68.71    68.72     95.47
```

Partial Least Squares

- The PCR approach involves identifying linear combinations, or *directions*, that best represent the predictors $X_1, \ldots, X_p$.
- These directions are identified in an *unsupervised* way, since the response Y is not used to help determine the principal component directions.

- Consequently, PCR suffers from a drawback: there is no guarantee that the directions that best explain the predictors will also be the best directions to use for predicting the response.
- Like PCR, partial least squares (PLS) is a dimension reduction method, which first identifies a new set of features $Z_1, ..., Z_M$ that are linear combinations of the original features, and then fits a linear model via least squares using these M new features.
- But unlike PCR, PLS identifies these new features in a supervised way—that is, it makes use of the response Y in order to identify new features that not only approximate the old features well, but also that are related to the response.