```
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.

That is, response does not supervise the identification of the principal components

- 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.

The PLS approach attempts to find directions that help explain both the response and the predictors.

Details of Partial Least Squares:

- After standardizing the p predictors and response, PLS computes the first direction Z_1 by setting each ϕ_{j1} in

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$$

equal to the coefficient from the simple linear regression of Y onto X_j.

ϕ_{j1} = $Y = \beta_{0j} + \beta_j X_j + \varepsilon_j$

- One can show that this coefficient is proportional to the correlation between Y and X_j.

Hence, in computing $Z_1 = \sum_{j=1}^{p} \phi_{j1} X_j$, PLS places the highest weight on the variables that are most strongly related to the response.

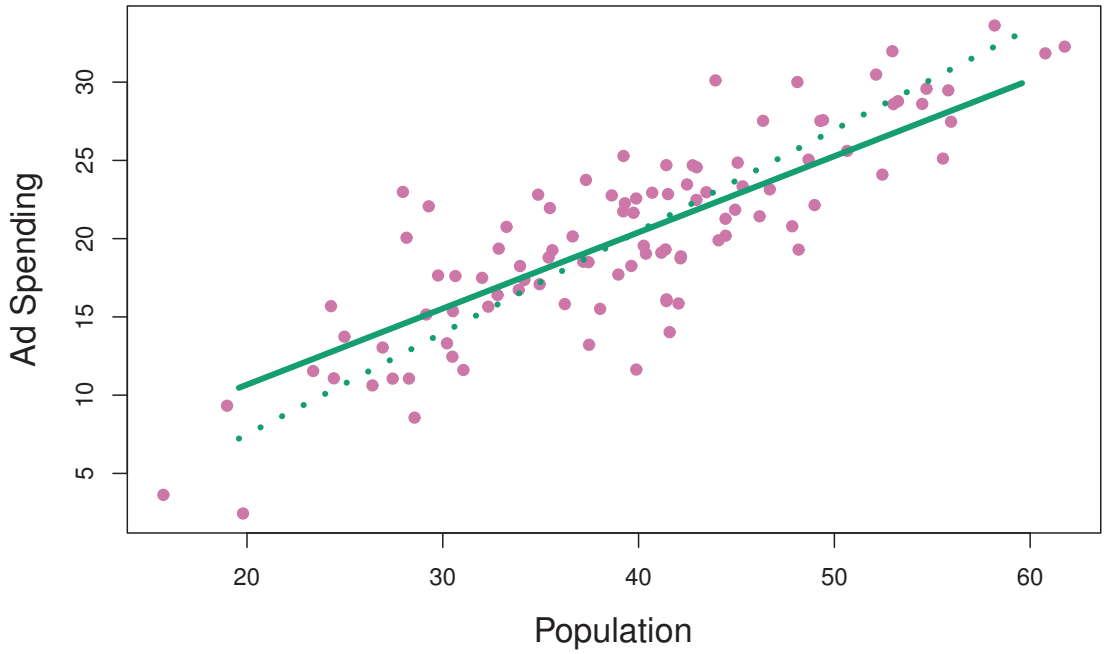
- To identify the second PLS direction we first adjust each of the variables for Z_1, by regressing each variable on Z_1 and taking residuals.

These residuals can be interpreted as the remaining information that has not been explained by the first PLS direction.

- We then compute Z_2 using this orthogonalized data in exactly the same fashion as Z_1 was computed based on the original data.

- This iterative approach can be repeated M times to identify multiple PLS components $Z_1, ..., Z_M$.

- Finally, at the end of this procedure, we use least squares to fit a linear model to predict Y using $Z_1, ..., Z_M$ in exactly the same fashion as for PCR.



For the advertising data (with Sales in each of 100 regions as the response, and two predictors; Population Size and Advertising Spending), the first PLS direction (solid line) and first PCR direction (dotted line) are shown.

The figure suggests that pop is more highly correlated with the response than is ad.

The PLS direction does not fit the predictors as closely as does PCA, but it does a better job explaining the response.

Note: As with PCR, the number M of partial least squares directions used in PLS is a tuning parameter that is typically chosen by cross-validation.

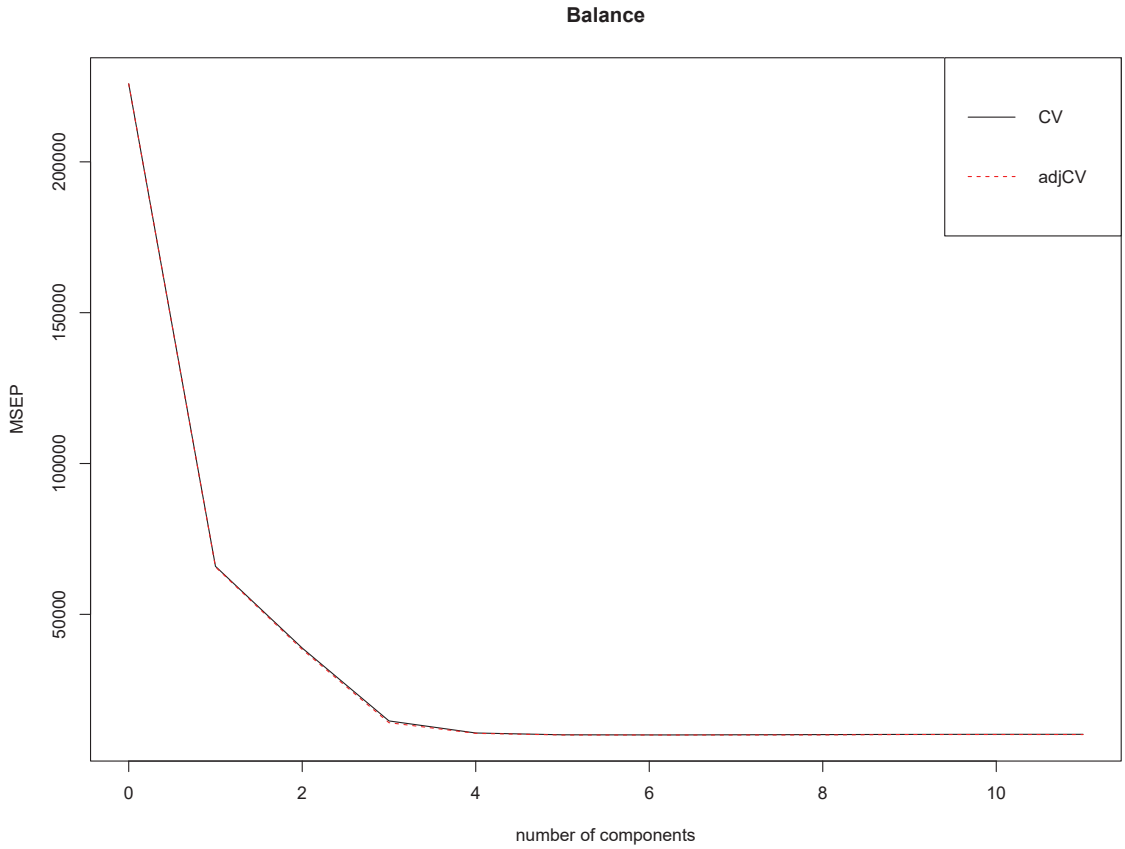
Performing PLS in R

We now apply partial least squares (PLS) to the Credit data in order to predict Balance. We implement PLS using the plsr() function, also in the pls library. The syntax is just like that of the pcr() function.

```
> set.seed(3)
> pls_fit <- plsr(Balance ~ ., data = Credit, subset = train, scale = TRUE,
  validation = "CV")
> summary(pls_fit)
Data:  X dimension: 200 11
Y dimension: 200 1
Fit method: kernelpls
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    256.8    197.0    121.1    103.3    100.4
adjCV        475.3    256.3    195.9    118.7    102.7    100.1
       6 comps  7 comps  8 comps  9 comps  10 comps  11 comps
CV       100.4    100.3    100.5    101.2     101.2     101.2
adjCV    100.0    100.0    100.1    100.8     100.8     100.8

TRAINING: % variance explained
         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
X          25.40    34.83    39.71    51.07     60.8    66.06
Balance    72.19    84.97    94.58    95.89     96.0    96.01
         7 comps  8 comps  9 comps  10 comps  11 comps
X          75.17    83.61    84.30     92.24    100.00
Balance    96.01    96.01    96.04     96.04     96.04
> validationplot(pls_fit, val.type = "MSEP", legendpos = "topright")
```



The lowest cross-validation error occurs when only $M = 7$ partial least squares directions are used. We now evaluate the corresponding test set MSE.

```
> pls_pred <- predict(pls_fit, x[test, ], ncomp = 7)
> mean((pls_pred - y_test)^2)
[1] 10650.9
```

Finally, we perform PLS using the full data set, using $M = 7$, the number of components identified by cross-validation.

```
> pls_fit <- plsr(Balance ~ ., data = Credit, scale = TRUE, ncomp = 7)
> summary(pls_fit)
Data:  X dimension: 400 11
Y dimension: 400 1
```

```
Fit method: kernelpls
Number of components considered: 7
TRAINING: % variance explained
         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
X          24.58    32.53    37.84    50.55    60.80    65.92
Balance    69.67    86.53    94.95    95.46    95.48    95.48
         7 comps
X          73.20
Balance    95.48
```

Notice that the percentage of variance in Balance that the one-component PLS fit explains, 69.67%, is almost as much as that explained using the nine-component model PCR fit, 68.72%.

This is because PCR only attempts to maximize the amount of variance explained in the predictors, while PLS searches for directions that explain variance in both the predictors and the response.

Moving Beyond Linearity

- Linear models are relatively simple to describe and implement, and have advantages over other approaches in terms of interpretation and inference.
- However, standard linear regression can have significant limitations in terms of predictive power.
- This is because the linearity assumption is almost always an approximation, and sometimes a poor one.
- In this chapter we relax the linearity assumption while still attempting to maintain as much interpretability as possible.
- We do this by examining extensions of linear models like
  - polynomial regression
  - step functions
  - regression splines
  - smoothing splines
  - local regression
  - generalized additive models

Polynomial Regression

- The standard way to extend linear regression to settings in which the relationship between the predictors and the response is non-linear has been to replace the standard linear model

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

with a polynomial function

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_d x_i^d + \epsilon_i, \qquad (*)$$

where ϵ_i is the error term. This approach is known as polynomial regression.

(*) is just a standard multiple linear regression model with predictors $x_i, x_i^2, x_i^3, \ldots, x_i^d$. The coefficients can be estimated using least squares.

- For large enough degree d, a polynomial regression allows us to produce an extremely non-linear curve.
- Generally speaking, it is unusual to use d greater than 3 or 4 because for large values of d, the polynomial curve can become overly flexible and can take on some very strange shapes.