Principal component analysis

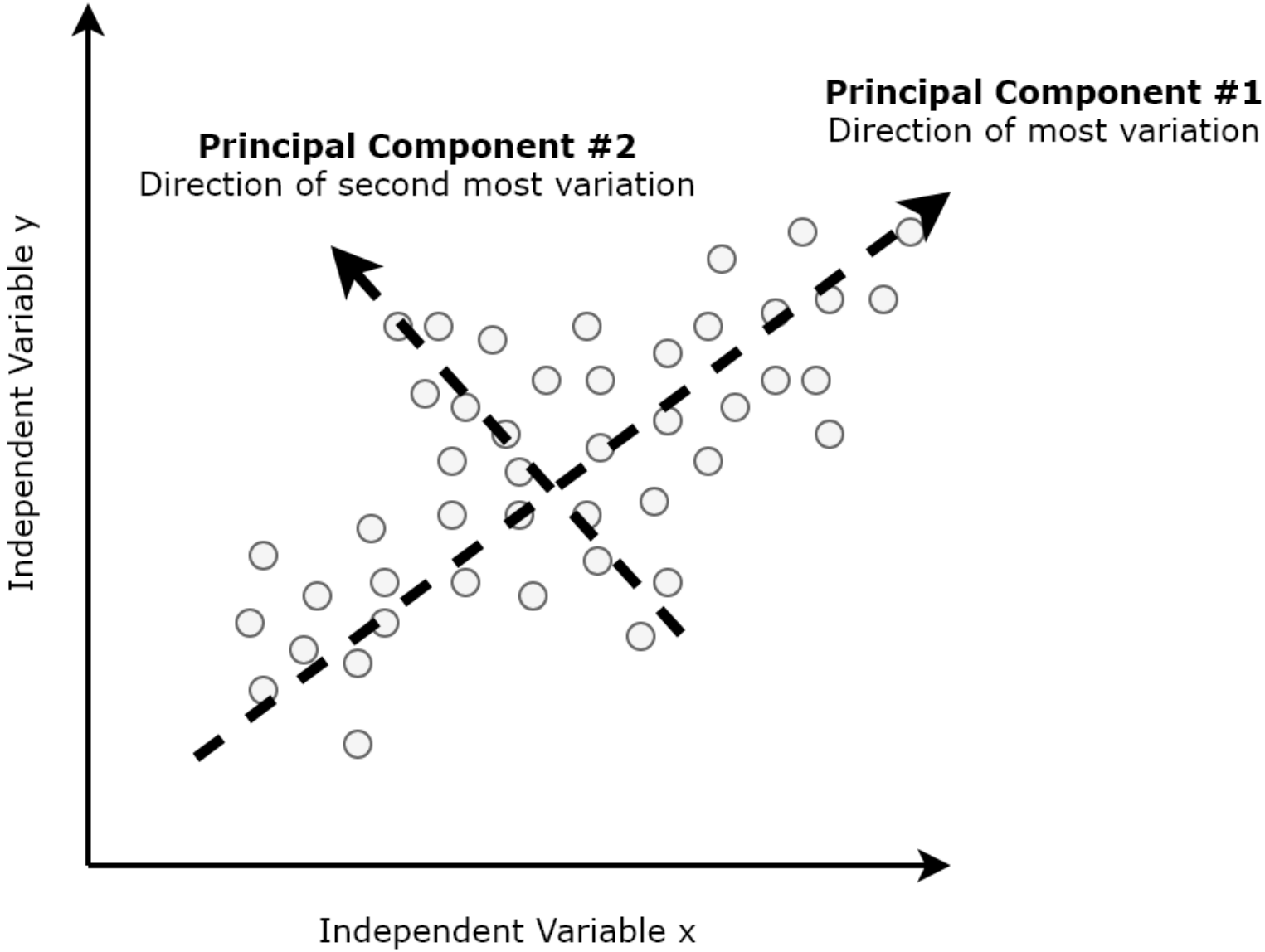
Yang Wang

College of Charleston

Principal component analysis

- Objective: dimension reduction
- When two or more regressors are highly correlated, the near linear dependence between the regressors results in a near singular regression matrix ($\boldsymbol{X}'\boldsymbol{X}$) which is reflected in a high variance inflation factor (VIF).
- In PCA, new variables are created
  - that are weighted linear combinations of the original variables
  - that retain the majority of the information of the full original set
  - that the correlation between them is 0
- PCA is intended for use with numerical variables.
- For categorical variables, other methods such as correspondence analysis are more suitable.

Principal component analysis



Principal components

- Step 1: normalize the data i.e. replace the original variable by a standardized version of the variable that has zero mean and unit variance.
  - subtract the mean and divide by standard deviation.
  - R function *scale()*.
  - Advantage: the differences in units of measurement do not affect the principal components' weights.
  - Suppose, normalized regressor variables are denoted as $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k$.
  - Define a $n \times k$ matrix $\boldsymbol{X}$ with the regressor variables as its columns *i.e.* $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k)$.
- Step 2: Use singular value decomposition (SVD) on the sample covariance matrix of $\boldsymbol{X}$ is $\boldsymbol{S} = \boldsymbol{X}'\boldsymbol{X}/n$ (Note $\boldsymbol{X}$ has zero mean.)

Singular value decomposition (SVD)

- Step 2:
  - $\boldsymbol{X}'\boldsymbol{X} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}'$.
  - The columns of $\boldsymbol{V}$ *i.e.* $\boldsymbol{v}_j$'s are the eigenvectors of $\boldsymbol{X}'\boldsymbol{X}$. They are called *principal component direction* of $\boldsymbol{X}$.
  - The matrix $\boldsymbol{D} = \begin{bmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d_k \end{bmatrix}$ *i.e.* d_j's are the eigenvalues of $\boldsymbol{X}'\boldsymbol{X}$.

Principal components

- The columns of $\boldsymbol{V}$ *i.e.* $\boldsymbol{v}_j$'s are called *principal component direction* of $\boldsymbol{X}$.
- $\boldsymbol{D} = \begin{bmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d_k \end{bmatrix}$. $d_1 \geq d_2 \geq \ldots, \geq d_k \geq \mathbf{0}$ are called the singular values of $\boldsymbol{X}$.
- The *first principal component direction* $\boldsymbol{v}_1$ is associated to the largest eigenvalue d_1. The *second principal component direction* $\boldsymbol{v}_2$ corresponds to the second highest eigenvalue d_2 and so on.
- $\boldsymbol{z}_j = \boldsymbol{X}\boldsymbol{v}_j$ is known as the j^{th} *principal component* of $\boldsymbol{X}$.
- $\boldsymbol{v}_j = (v_{j1}, v_{j2}, \ldots, v_{jk})$ is the j^{th} eigenvector of $\boldsymbol{X}'\boldsymbol{X}$ and denotes the j^{th} principal component direction of $\boldsymbol{X}$.

Principal components

- $\boldsymbol{z}_j = \boldsymbol{X}\boldsymbol{v}_j$ is known as the j^{th} *principal component* of $\boldsymbol{X}$.
- $\boldsymbol{z}_j = \boldsymbol{X}\boldsymbol{v}_j = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k) \begin{bmatrix} v_{j1} \\ v_{j2} \\ \vdots \\ v_{jk} \end{bmatrix} = v_{j1}\boldsymbol{x}_1 + v_{j2}\boldsymbol{x}_2 + \cdots + v_{jk}\boldsymbol{x}_k$
- The the j^{th} principal component $\boldsymbol{z}_j$ is a weighted average of the normalized original regressor variables.
- The weights v_{ji}'s are known as *loadings*.
- Suppose $\boldsymbol{Z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_k)$. Then we note, $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{V}$.
- The 1^{st} *principal component* $\boldsymbol{z}_1$ has the largest sample variance among all normalized linear combinations of the columns of $\boldsymbol{X}$. The 2^{nd} *principal component* $\boldsymbol{z}_2$ has a direction orthogonal to the first principal component and has the second highest sample variance. The j^{th} *principal component* $\boldsymbol{z}_j$ is orthogonal to the first $(j-1)$ principal components and account for the j^{th} highest sample variance.

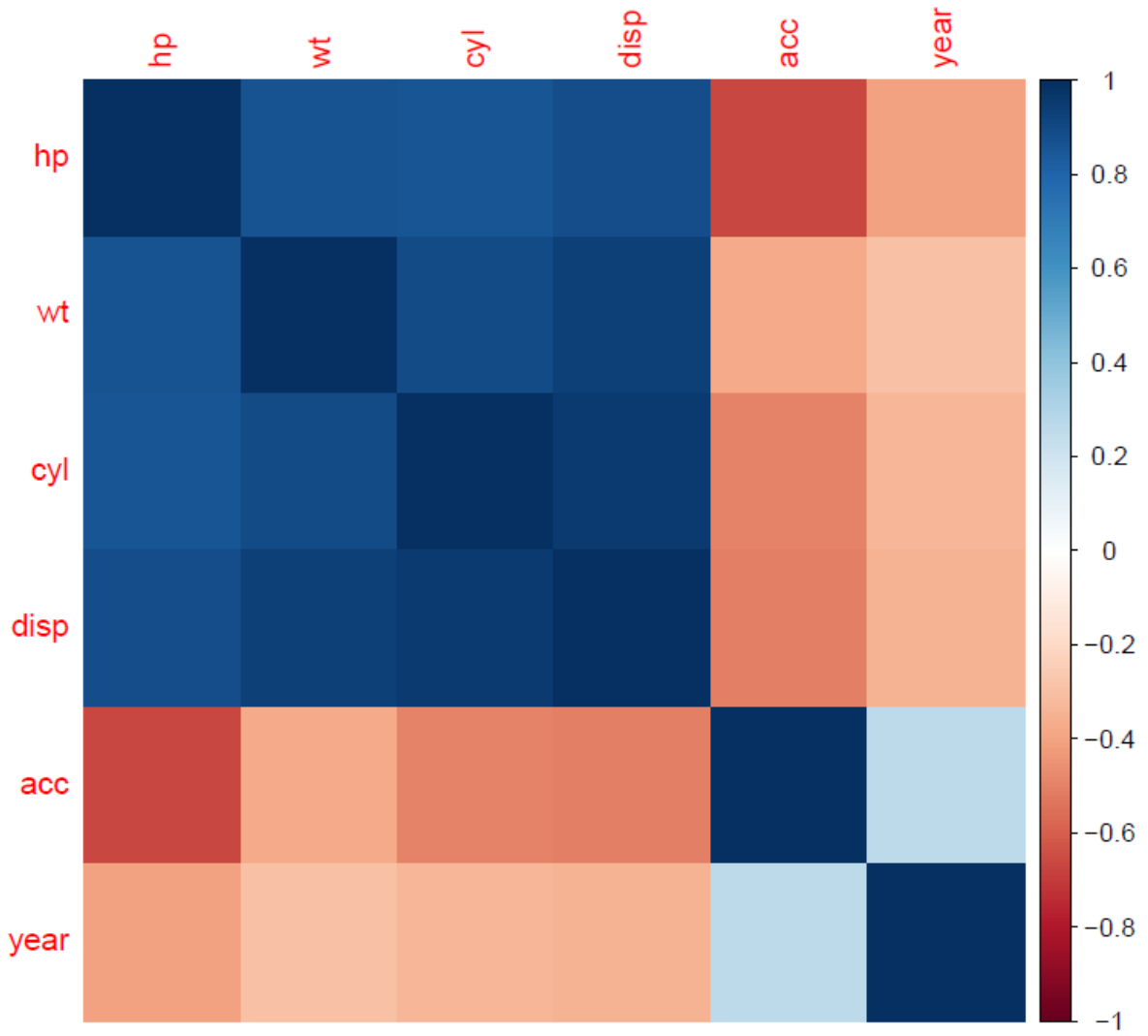
Example on autompg data

```
p3 <- partition.3(Dat, 0.7, 0.2) ## creating 70:20:10 partition
training.data <- p3$data.train
validation.data <- p3$data.val
test.data <- p3$data.test

## Fit MLR model on training data
mlr.train <- lm(mpg ~ ., data = training.data)
> summary(mlr.train)

Call:
lm(formula = mpg ~ ., data = training.data)

Residuals:
    Min      1Q  Median      3Q     Max
-8.9662 -2.4422 -0.1279  2.0443 14.2899

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.646e+01  5.789e+00  -2.843  0.00482 **
cyl         -3.123e-01  4.138e-01  -0.755  0.45108
disp        -1.019e-03  9.314e-03  -0.109  0.91295
hp          -6.495e-04  1.601e-02  -0.041  0.96766
wt          -5.887e-03  8.170e-04  -7.206 5.94e-12 ***
acc          3.504e-02  1.201e-01   0.292  0.77068
year         7.742e-01  6.455e-02  11.995  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.501 on 266 degrees of freedom
Multiple R-squared:  0.8121,Adjusted R-squared:  0.8078
F-statistic: 191.5 on 6 and 266 DF,  p-value: < 2.2e-16
```

Example on autompg data

```
# RMSE for training data
error.train <- training.data$mpg - mlr.train$fitted.values
rmse.train <- sqrt(mean(error.train^2))
> rmse.train
[1] 3.45619

# prediction on validation data
yhat = predict(mlr.train, newdata=validation.data)
# RMSE for validation data
error.val <- validation.data$mpg - yhat
rmse.val <- sqrt(mean(error.val^2))
> rmse.val
[1] 3.141566
```

Example on autompg data: correlation plot

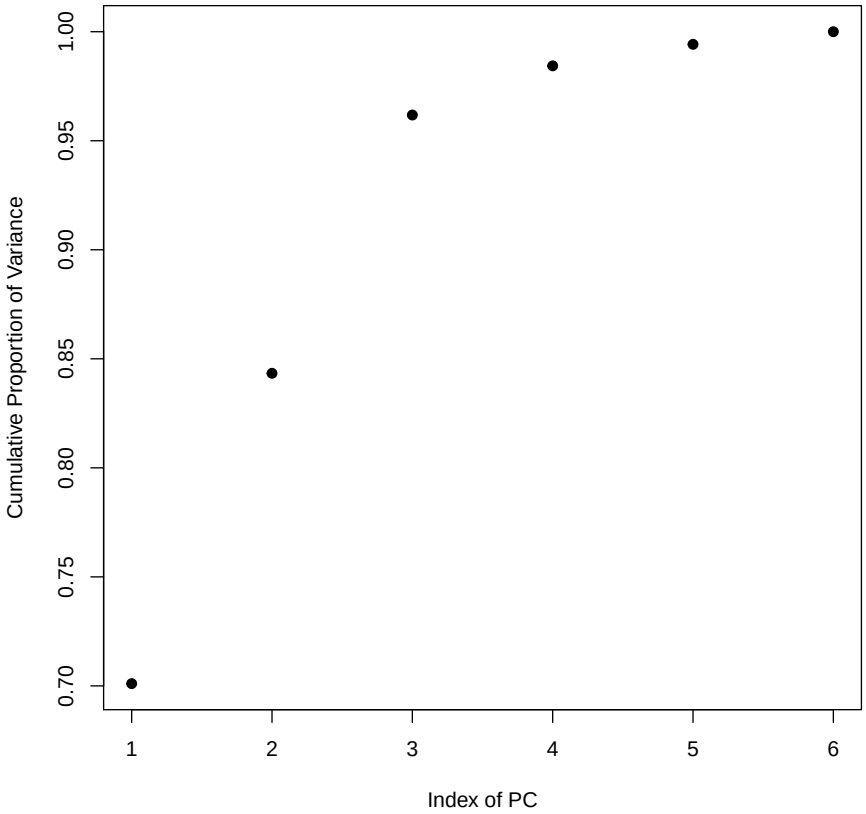


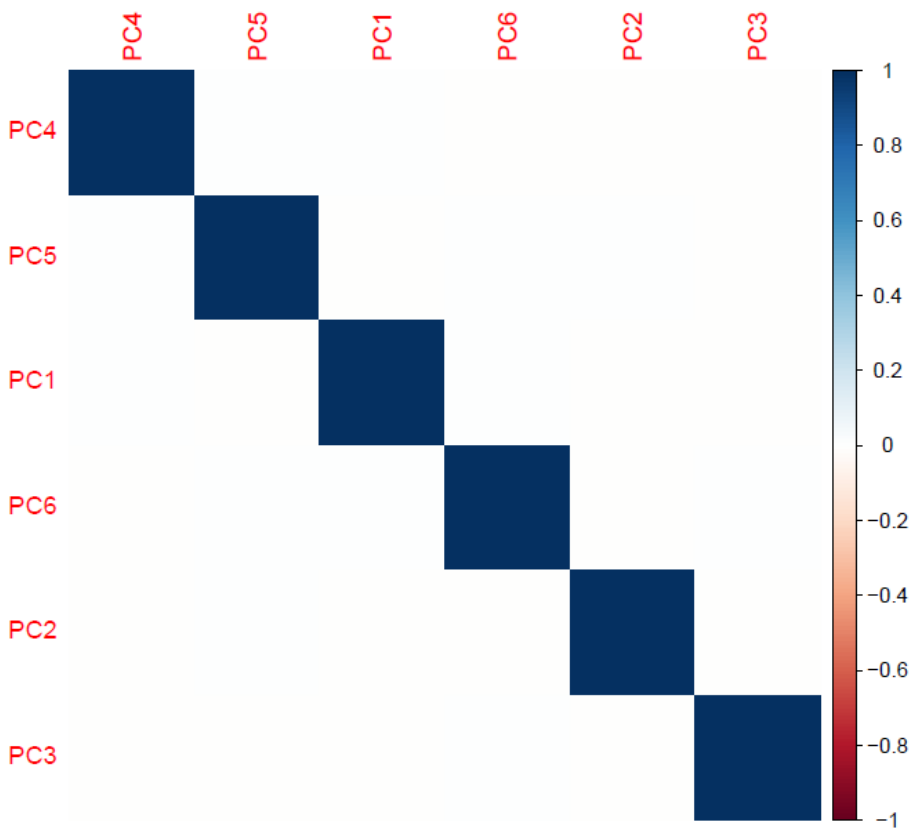
Example on autompg data: creation of principal components

```
# Scale the data
training.scaled <- scale(training.data[,-1], center = TRUE, scale = TRUE)

# Create principal components
pca <- prcomp(training.scaled, center = TRUE, scale. = TRUE)
> summary(pca)
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6
Standard deviation     2.0575 0.9068 0.8412 0.36773 0.25947 0.18538
Proportion of Variance 0.7055 0.1371 0.1179 0.02254 0.01122 0.00573
Cumulative Proportion  0.7055 0.8426 0.9605 0.98305 0.99427 1.00000
```

- The first principal component $PC1$ account for 70.1% variance.
- The second principal component $PC2$ account for 13.71% variance and so on.
- The first two principal components preserve 84.26% variance.
- The first three principal components preserve 96.05% variance and so on.

Example on autompg data: cumulative proportion of variance and correlation plot





Example on autompg data: loadings

```
# Create principal components
pca <- prcomp(training.scaled, center = TRUE, scale. = TRUE)
> summary(pca)
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6
Standard deviation     2.0575 0.9068 0.8412 0.36773 0.25947 0.18538
Proportion of Variance 0.7055 0.1371 0.1179 0.02254 0.01122 0.00573
Cumulative Proportion  0.7055 0.8426 0.9605 0.98305 0.99427 1.00000
> pca$rotation
             PC1         PC2        PC3        PC4         PC5         PC6
cyl   0.4556144 -0.19190206  0.1829264 -0.6144088  0.36533885 -0.45951755
disp  0.4708120 -0.15351665  0.1338182 -0.2050499 -0.03197143  0.83294280
hp    0.4631948 -0.01281404 -0.1041194  0.6857295  0.54854760 -0.05758523
wt    0.4515509 -0.22255767  0.2606873  0.2783420 -0.71614813 -0.29710193
acc  -0.3104407 -0.14749124  0.8950216  0.1717020  0.21972814  0.05520085
year -0.2364961 -0.93174914 -0.2622368  0.0572324  0.05875620  0.02042433
pcs <- as.data.frame(pca$x)
```

- Recall: the j^{th} principal component $\boldsymbol{z}_j$ is a weighted average of the normalized original regressor variables *i.e.* $\boldsymbol{z}_j = v_{j1}\boldsymbol{x}_1 + v_{j2}\boldsymbol{x}_2 + \cdots + v_{jk}\boldsymbol{x}_k$.
- These weights are known as loadings.
- $PC1 = 0.4556144 \times cyl + 0.4708120 \times disp + 0.4631948 \times hp + 0.4515509 \times wt - 0.3104407 \times acc - 0.2364961 \times year$.
- Note that, the regressors in this equation are scaled.

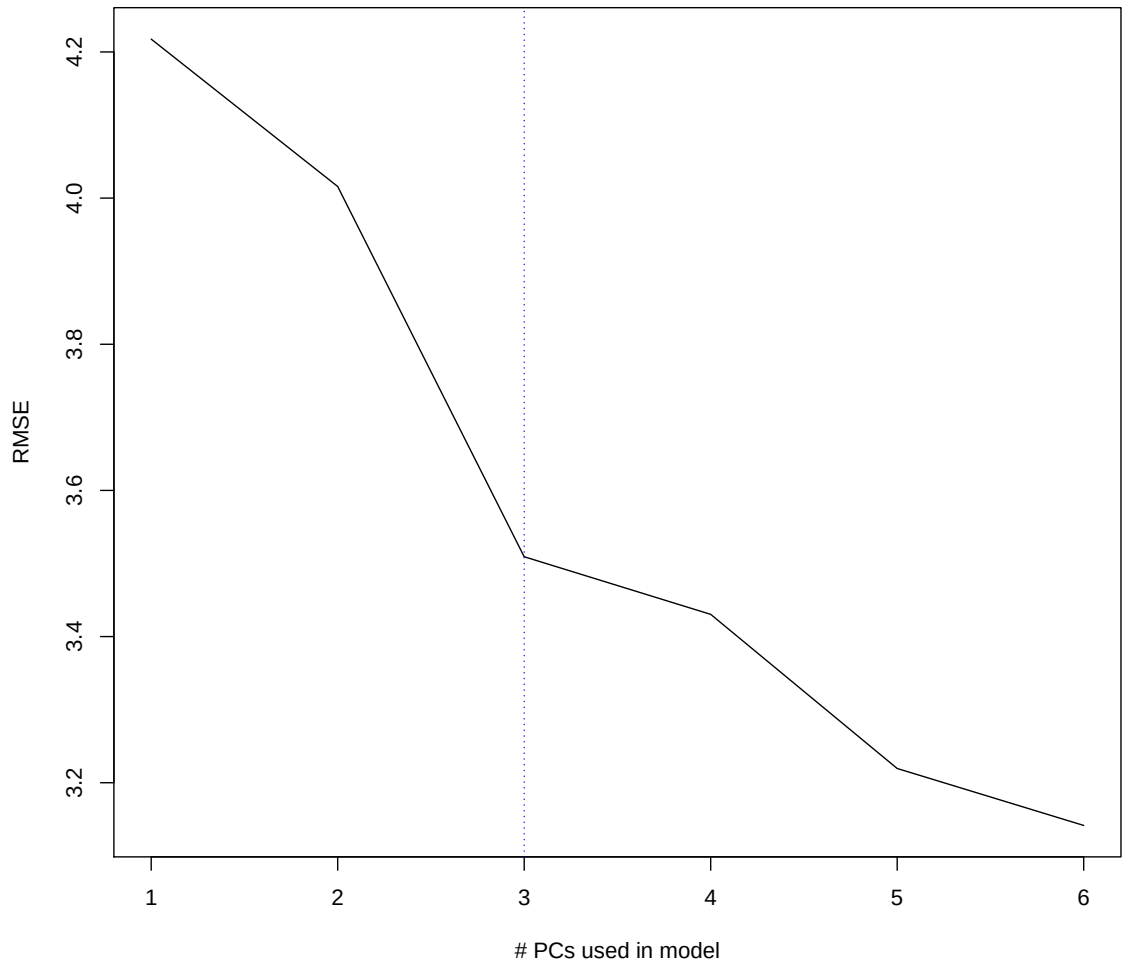
Example on autompg data: use of PCs in MLR

```
pcs <- as.data.frame(pca$x)
# MLR using all PCs
lr.data <- cbind(training.data$mpg, pcs) # create a data set with mpg and principal components
colnames(lr.data)[1] <- "mpg"
mlr.pc.train <- lm(mpg ~ PC1+PC2+PC3+PC4+PC5+PC6, data = lr.data)
> summary(mlr.pc.train)

Call:
lm(formula = mpg ~ PC1 + PC2 + PC3 + PC4 + PC5 + PC6, data = lr.data)

Residuals:
    Min      1Q  Median      3Q     Max
-8.9662 -2.4422 -0.1279  2.0443 14.2899

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  23.5029     0.2119 110.909  < 2e-16 ***
PC1          -3.2942     0.1032 -31.925  < 2e-16 ***
PC2          -1.4242     0.2341  -6.083 4.08e-09 ***
PC3          -2.0950     0.2524  -8.301 5.26e-15 ***
PC4          -0.8988     0.5773  -1.557    0.121
PC5           3.6109     0.8182   4.413 1.48e-05 ***
PC6           1.7268     1.1452   1.508    0.133
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.501 on 266 degrees of freedom
Multiple R-squared:  0.8121,Adjusted R-squared:  0.8078
F-statistic: 191.5 on 6 and 266 DF,  p-value: < 2.2e-16
```

Example on autompg data: use of PCs in MLR - model evaluation

```
# Scaling validation data
training.scaled.attr <- attributes(training.scaled)
val.scaled <- scale(validation.data[,-1],
                    center = training.scaled.attr$`scaled:center`,
                    scale = training.scaled.attr$`scaled:scale`)

# Create PC's on validation data
loading <- pca$rotation
val.pcs <- val.scaled%*%loading

# Model performance on validation data
yhat = predict(mlr.pc.train, newdata=as.data.frame(val.pcs))
error.val.pc <- validation.data$mpg - yhat
rmse.val.pc <- sqrt(mean(error.val.pc^2))
> rmse.val.pc
[1] 3.141566
```

Example on autompg data: determination of optimal number of PCs using validation data



Example on autompg data: use of PCs in MLR - model evaluation on test data

```
training.data.all <- rbind(training.data, validation.data)
# Scale the data
training.scaled.all <- scale(training.data.all[,-1], center = TRUE, scale = TRUE)
# Create principal components
pca <- prcomp(training.scaled.all, center = TRUE, scale. = TRUE)
pcs <- as.data.frame(pca$x)
# MLR using all PCs
lr.data.all <- cbind(training.data.all$mpg, pcs) # create a data set with mpg and principal compon
colnames(lr.data.all)[1] <- "mpg"
mlr.pc.train.all <- lm(mpg ~ PC1+PC2+PC3, data = lr.data.all) # fit model with 3 principal compone
# Scaling test data
training.scaled.all.attr <- attributes(training.scaled.all)
test.scaled <- scale(test.data[,-1],
                     center = training.scaled.all.attr$`scaled:center`,
                     scale = training.scaled.all.attr$`scaled:scale`)

# Create PC's on test data
loading <- pca$rotation
test.pcs <- test.scaled%*%loading
# Model performance on test data
yhat = predict(mlr.pc.train.all, newdata=as.data.frame(test.pcs))
error.test.pc <- test.data$mpg - yhat
rmse.test.pc <- sqrt(mean(error.test.pc^2))
> rmse.test.pc
[1] 4.132863
```

Advantages and disadvantages of PCA

Advantage

- PCA is an effective dimension reduction tool. The use of the first few principal components instead of all original variables improves the model performance, speeds up the algorithm and takes care of the overfitting issue.
- Removes multicollinearity. The principal components are uncorrelated.

Disadvantage

- The original variables are lost. The calculated principal components are linear combinations of the original variables. That is why PCA can not be considered as a variable selection method.
- It is hard to explain a model fitted using principal components. Since principal components are linear combinations of the original variables, it is difficult to know how the original variables are affecting the predicted response.
- Although the first few principal components account for the majority of the variance, some important information might be lost.
- In regression/classification using principal components, the directions of the principal components are obtained in an unsupervised way *i.e.* the response variable does not play any role in identifying the directions of the principal components. Therefore, it is not guaranteed that the directions found would be the optimal direction for explaining the response variable.