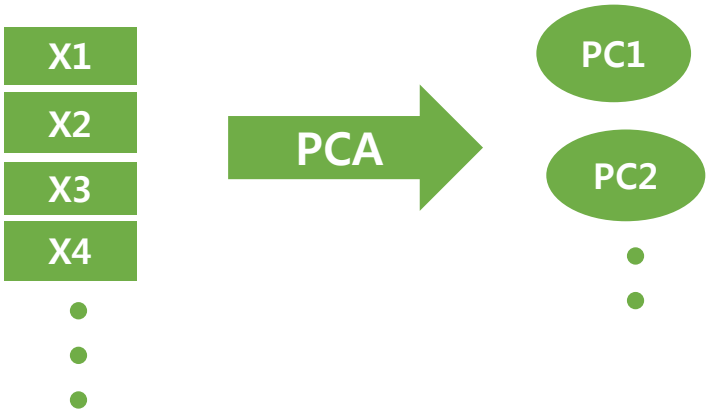
	단위별 학습내용 (Week15)
wk15-1	주성분분석 (Principle Component Analysis)
wk15-2	주성분회귀분석 (Principle Component Regression)
wk15-3	부분최소자승회귀 (Partial Least Square Regression)

Wk15-1 : 주성분분석
(Principle Component Analysis)

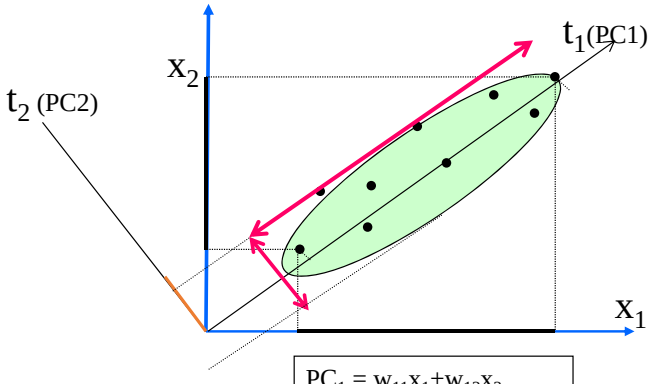
1. 주성분분석 (Principle Component Analysis)

- 다변량분석기법
- '주성분'이라고 불리는 선형조합으로 표현하는 기법
- 여기서 주성분은 공분산(X^TX)으로부터 eigenvector와 eigenvalue를 도출하여 계산됨



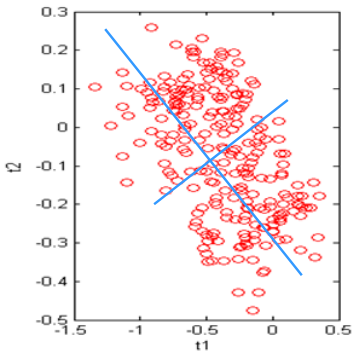
1. 주성분분석 (Principle Component Analysis)

- 주성분간의 수직관계
  - 1st 주성분 (PC1) : 독립변수들의 변동(분산)을 가장 많이 설명하는 성분
  - 2nd 주성분 (PC2) : PC1과 수직인 주성분
    (첫번째 주성분이 설명하지 못하는 변동에 대해 두번째로 설명하는 성분)



$PC_1 = w_{11}x_1 + w_{12}x_2$

$PC_2 = w_{21}x_1 + w_{22}x_2$



1. 주성분분석 (Principle Component Analysis)

- iris데이터(4개변수)의 주성분분석 – 차원축소 & 예측력 향상

input변수(독립변수) output변수(종속변수, 타겟변수)

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3	1.4	0.1	setosa

타겟변수(y) : setosa, versicolor, virginica

Iris setosa Iris versicolor Iris virginica

주성분분석 (Principle Component Analysis)

- iris데이터(4개변수)의 주성분 도출 – 차원축소 & 예측력 향상

```
# lec15_1_pca.r
# Multivariate analysis
# Principle Component Analysis

# set working directory
setwd("D:/tempstore/moocr/wk15")

#input data(iris)
iris<-read.csv(file="iris.csv")
attach(iris)
head(iris)
```

주성분분석은 추가패키지 필요없음

데이터 불러들이기

```
#Check correlation
cor(iris[1:4])
```

독립변수간 상관관계 확인

```
> cor(iris[1:4])
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

0.96, 0.87등 높은 상관계수가 관찰됨

1. 주성분분석 (Principle Component Analysis)

• 주성분분석을 위한 함수 : prcomp(독립변수들, center= , scale=)

```
# 1.PCA(center=T->mean=0, scale.=T->variance=1)
ir.pca<-prcomp(iris[,1:4],center=T,scale.=T)
ir.pca
summary(ir.pca)
```

- 옵션을 주지않으면 center=T, scale=F
- center=T, scale=T는 변수들의 평균을 빼고, 편차로 나누어 표준화한다는 의미.

```
> ir.pca<-prcomp(iris[,1:4],center=T,scale.=T)
> ir.pca
Standard deviations (1, .., p=4):
[1] 1.7083611 0.9560494 0.3830886 0.1439265

Rotation (n x k) = (4 x 4):
                    PC1         PC2        PC3        PC4
Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
```

PC1 = 0.5211*Sepal.Length -0.2693* Sepal.Width + 0.5804 * Petal.Length + 0.5649*Petal.Width

1. 주성분분석 (Principle Component Analysis)

• 전체분산 중 각 주성분의 설명하는 비율

```
> summary(ir.pca)
Importance of components:
                          PC1    PC2     PC3     PC4
Standard deviation     1.7084 0.9560 0.38309 0.14393
Proportion of Variance 0.7296 0.2285 0.03669 0.00518
Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

- PC1은 전체분산의 72.96%를 설명
- PC2은 전체분산의 22.85%를 설명
- PC3는 전체분산의 3.67%를 설명
- PC4는 전체분산의 0.5%를 설명

누적설명비율을 보면 PC1와 PC2, 두개의 성분으로 전체분산의 95.81%를 설명

그러면 몇 개의 주성분으로 전체분산을 설명하는게 최적?

1. 주성분분석 (Principle Component Analysis)

• 최적 주성분 수는? – scree plot을 그려보고 급격히 떨어지기 전까지의 PC를 선택

```
# 2.scree plot : to choose the number of components
plot(ir.pca,type="l")
```

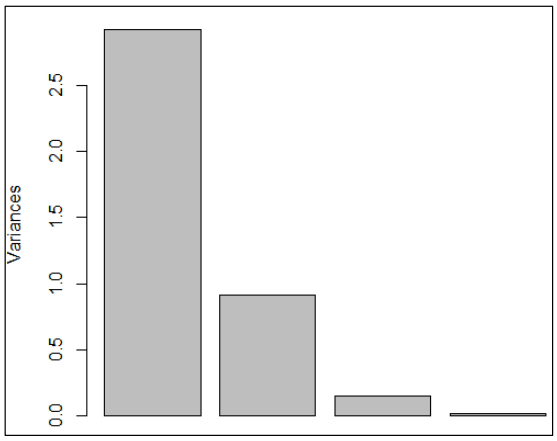
- 3rd PC에서 설명력이 급격하게 떨어짐을 볼 수 있음
- 기울기가 꺾이는 PC3을 'elbow point'라 부름

=> 이 경우는 PC1, PC2 까지 사용하는 것을 추천

1. 주성분분석 (Principle Component Analysis)

• screeplot함수를 이용 : screeplot(pca결과)

```
# either way to draw scree plot
screeplot(ir.pca)
```



앞의 그림과 동일한 결과
=> PC1, PC2까지 사용 추천

1. 주성분분석 (Principle Component Analysis)

- PC계산= X_data(n*p) %*% PCA_weight(p*p)

```
#3. calculate component=x_data%*% PCA weight
PRC<-as.matrix(iris[,1:4])%*%ir.pca$rotation
head(PRC)
```

PRC는 n*p행렬, 여기서는 150*4
head(PRC)는 첫번째 6줄을 보여줌

```
> PRC<-as.matrix(iris[,1:4])%*%ir.pca$rotation
> head(PRC)
          PC1       PC2      PC3         PC4
[1,] 2.640270 -5.204041 2.488621 -0.1170332
[2,] 2.670730 -4.666910 2.466898 -0.1075356
[3,] 2.454606 -4.773636 2.288321 -0.1043499
[4,] 2.545517 -4.648463 2.212378 -0.2784174
[5,] 2.561228 -5.258629 2.392226 -0.1555127
[6,] 2.975946 -5.707321 2.437245 -0.2237665
```

PC1 = 0.5211*Sepal.Length -0.2693* Sepal.Width + 0.5804 * Petal.Length + 0.5649*Petal.Width

2. 주성분을 이용한 분류모형

- iris data => iris.pc data구성

```
# 4. classification using principal components
# make data with components
iris.pc<-cbind(as.data.frame(PRC), Species)
head(iris.pc)
```

iris.pc 데이터

```
> head(iris.pc)
        PC1       PC2      PC3         PC4 Species
1 2.640270 -5.204041 2.488621 -0.1170332  setosa
2 2.670730 -4.666910 2.466898 -0.1075356  setosa
3 2.454606 -4.773636 2.288321 -0.1043499  setosa
4 2.545517 -4.648463 2.212378 -0.2784174  setosa
5 2.561228 -5.258629 2.392226 -0.1555127  setosa
6 2.975946 -5.707321 2.437245 -0.2237665  setosa
```

iris데이터

```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

PC1 = 0.5211*Sepal.Length -0.2693* Sepal.Width + 0.5804 * Petal.Length + 0.5649*Petal.Width

2. 주성분을 이용한 분류모형

- 주성분을 이용한 서포트벡터머신 수행 (iris data)

```
# install package for support vector machine
#install.packages("e1071")
library (e1071)

# classify all data using PC1-PC4 using support vector machine
m1<- svm(Species ~., data = iris.pc, kernel="linear")
# m2<- svm(Species ~PC1+PC2, data = iris.pc, kernel="linear")
summary(m1)
```

```
> m1<- svm(Species ~., data = iris.pc, kernel="linear")
> # m2<- svm(Species ~PC1+PC2, data = iris.pc, kernel="linear")
> summary(m1)

Call:
svm(formula = Species ~ ., data = iris.pc, kernel = "linear")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  1
      gamma:  0.25

Number of Support Vectors:  32

 ( 3 16 13 )
```

PC1-PC4까지 모두를 input으로
분류모형(서포트벡터머신) 수행

2. 주성분을 이용한 분류모형

- 주성분을 이용한 서포트벡터머신 수행 (iris data)

```
# predict class for all data
x<-iris.pc[, -5]
pred <- predict(m1, x)
# check accuracy between true class and predicted class
y<-iris.pc[,5]
table(pred, y)
```

```
> x<-iris.pc[, -5]
> pred <- predict(m1, x)
> # check accuracy between true class and predicted class
> y<-iris.pc[,5]
> table(pred, y)
            y
pred         setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         0
  virginica       0          2        50
```

주성분을 이용한 분류 오분류율 :
2/150=0.013 (1.33%)

비교해봅시다!!

Week11_1_SVM (12page)오분류율 :
(2+2)/150=0.0266 (2.66%)

```
> y<-iris[,5]
> table(pred, y)
            y
pred         setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0          2        48
```