# Unsupervised Learning
# (Principal Component Analysis)

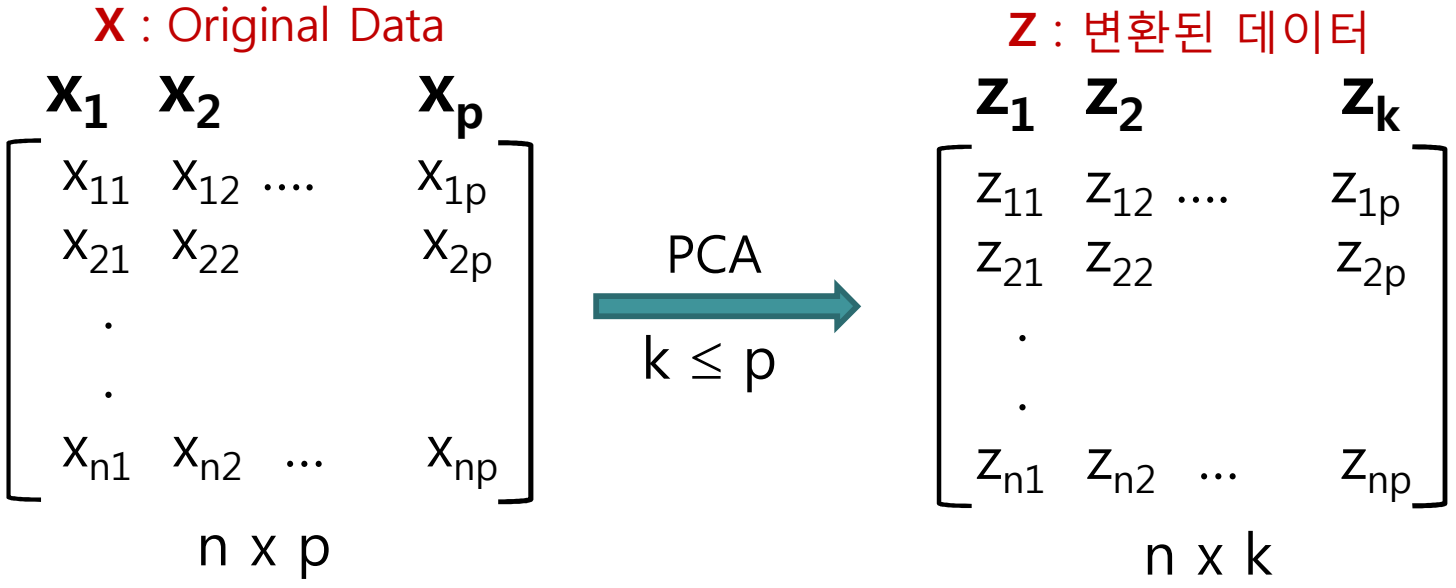
# Unsupervised Learning (자율학습)

- Target/답 역할을 해주는 Y 없이 학습이 이루어지는 형태
- feature ($X_1$, $X_2$, . . . , $X_p$)만 갖고서 이것들로부터 유용한 패턴, 그룹핑, 또는 기타 정보를 추출 함이 목표
- 정답 역할을 하는 response Y가 없으므로 학습이 제대로 이루어지고 있는지, 학습 결과가 잘 된 것인지 판단하기가 어려움
- Response가 없으므로 training set, test set으로 나누기도 의미 없음
- CRM에서 고객들을 유사성에 따라 그룹으로 나누는 Clustering이 대표적인 사례이며, supervised learning의 전단계로 많이 활용

# Principal Component Analysis

정보를 **간단하게 요약하고** 싶을 때가 많다. 중요하고 핵심적인 것은 놓치지 않으면서 중복되고, 정보가치가 없는 것들은 제거하여. Principal Component Analysis (PCA)는 이런 필요가 있을 때 쓸 수 있는 기술 중 하나

p개의 feature를 갖는 n개의 observation들로 된 데이터를 p보다 작은 수 'k' 개의 feature들로 나타낼 수 있을까이다. 이 때 k개 feature들은 feature 선정을 통해 구한 원래 n개 feature 중의 '중요한' feature들이 **아니라 새롭게 만들어진 feature** 임. 이를 matrix 식으로 표현하면;



# Principal Component Analysis (주성분 분석) 의미

- PCA는 p 개의 변수(feature)를 갖는 n observation 들이 있을 때, 기존 p개의 feature들을 분석·재조합해서 일련의 **새로운** feature들을 생성하는데, feature들이 가장 중요한 것(정보를 많이 담고 있는 것) 부터 만들어진다. 이 때 **새로운** feature들은 원래 p개의 feature들이 갖고 있던 가장 흥미로운 정보부터 담으려 하는데, 흥미로운 정보란 원래 데이터에서 **variation (변화량)이 큰** feature들이 담고 있는 것을 말한다.

- 새롭게 만든 이 feature들을 principal component (score)라 한다. **첫 번째 principal component가 가장 (variation) 정보를 많이 담고 있고, 두 번째가 그 다음, 이런 형태다**.

- 따라서 원래 p개의 변수들의 정보가 서로 많이 연관·중복되어 있을 경우 PCA를 하면 훨씬 적은 **앞쪽의** 몇 principal component만 갖고서도 원래 p개의 변수들이 갖고 있던 정보의 대부분을 담을 수 있다. 즉, PCA를 하면 때론 정보를 많이 잃지 않고서도 원래 변수의 개수(dimension 크기, 데이터 양)를 확 줄일 수 있다. 일종의 데이터 요약/압축으로 생각해도 좋다.

- PCA를 하여 dimension을 줄이면 supervised learning에서 "curse of dimension" 을 피할 수 있어 좋고, 또 PCA 결과를 시각화하여 데이터의 특성을 살펴보기에도 좋을 때가 있다

# Principal Component Analysis 설명

- 데이터가 $X_1, X_2, \ldots, X_p$ feature들로 구성되어있다 하자 ($X_i$, i=1, … p : 길이 n의 column vector).

- $X_1, X_2, \ldots, X_p$ 의 **첫 번째 principal component (score)** $Z_1$은 X들의 nomalized linear combination 로써 가장 **큰 variation을 갖도록 한** combination 이다 .

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \ldots + \phi_{p1} X_p$$

이때 normalized라 함은 $\sum_{j=1}^{p} \phi_{j1}^2 = 1$ 를 뜻한다

- 위 식에서 $\mathbf{Z_1}$은 p개의 feature들 각각에 weight $Ø_{11}, \ldots, Ø_{\mathbf{p1}}$ 들을 곱해 합을 구한 형태이다. 이 때 feature들은 이미 관찰된 것이니, 핵심은 $Ø_{11}, \ldots, Ø_{\mathbf{p1}}$ 들을 무슨 값으로 해야 $\mathbf{Z_1}$의 분산이 가장 클 가 이다. Weight $Ø_{11}, \ldots, Ø_{\mathbf{p1}}$ 들을 **첫번째 principal component의 loading** 이라고 한다. 우리는 loading의 "sum of squares" 를 1로 제한을 건다. **PCA는 X에서 loading Ø, Principal Components Z를 구함**.

- n observation과 p feature를 갖는 데이터에서 계산할 수 있는 principal component 의 개수는 min(n-1, p) 이다. 즉, 전체 Principal Component를 나타내는 Z는 n x (n-1) 또는 n x p 행렬 형태가 된다.

# Principal Components Analysis 결과는,

- 일반적으로 p < n 이므로 min(n-1, p)=p 라 간주하면 principal components Z와 feature X, 그리고 우리가 구하려는 loading Ø, PC **Z** 관계가 다음과 같이 표현된다.

$$Z_{nxp} = X_{nxp} \, Ø_{pxp}$$

$$* \; Z_{nx(n-1)} = X_{nxp} \cdot Ø_{px(n-1)} \quad \text{if } p>n$$

$\mathbf{Ø_1}$ : **loadings** of first principal component of X
$\mathbf{X_1}$ : first feature vector
$\mathbf{Z_1}$: $z_{11}, z_{21}, \ldots, z_{n1}$; **scores** of first principal component

Scores(Principal Components) Original Data Loadings/Rotation

$$\begin{matrix} \mathbf{Z_1} & \mathbf{Z_2} & & \mathbf{Z_p} \end{matrix}$$
$$\begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1p} \\ z_{21} & z_{22} & & z_{2p} \\ \cdot & & & \\ \cdot & & & \\ z_{n1} & z_{n2} & \cdots & z_{np} \end{bmatrix}$$
n x p

$$=$$

$$\begin{matrix} \mathbf{X_1} & \mathbf{X_2} & & \mathbf{X_p} \end{matrix}$$
$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & & x_{2p} \\ \cdot & & & \\ \cdot & & & \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$
n x p

$$\times$$

$$\begin{matrix} \mathbf{Ø_1} & \mathbf{Ø_2} & & \mathbf{Ø_p} \end{matrix}$$
$$\begin{bmatrix} Ø_{11} & Ø_{12} & \cdots & Ø_{1p} \\ Ø_{21} & Ø_{22} & & Ø_{2p} \\ \cdot & & & \\ \cdot & & & \\ Ø_{p1} & Ø_{p2} & \cdots & Ø_{pp} \end{bmatrix}$$
p x p

- 위 식에서 $\mathbf{Z_1} = Ø_{11}\mathbf{X_1} + Ø_{21}\mathbf{X_2} + \ldots + Ø_{p1}\mathbf{X_p}$ 임을 알 수 있다. 그리고 column 벡터 $\mathbf{Ø_1}$ 은 첫 번째 PC $\mathbf{Z_1}$ 을 구할 때만 관여한다는 것을 알 수 있다.

# Principal Components 결과를 설명하면,

**A**가 m x n 행렬이고, **B**가 column 들 $\mathbf{b}_1, \ldots, \mathbf{b}_p$ 로 된 n x p 행렬이면, 행렬 곱 **AB**는 m x p 행렬로써,

$$\mathbf{AB} = \mathbf{A}[\mathbf{b}_1 \quad \mathbf{b}_2 \quad \ldots \mathbf{b}_p] = [\,\mathbf{Ab}_1 \quad \mathbf{Ab}_2 \quad \ldots \quad \mathbf{Ab}_p\,]$$

**즉**, **AB**의 각 column은 행렬 **A**와 행렬 **B**의 해당 column과의 Matric-Vector 곱, 즉 Linear Combination

따라서 PCA 에서 $Z = X\varnothing$ 이니,

$Z_1 = \mathbf{X}\varnothing_1$ ( $Z_1, \varnothing_1$: column vector, **X** : n x p 행렬 )

$= \varnothing_{11}X_1 + \varnothing_{21}X_2 + \ldots + \varnothing_{p1}X_p$

→ $Z_1$ 은 Loading 행렬의 첫 번째 loading column $\varnothing_1$ 을 weight로 하는 오리지널 feature $X_1, X_2, \ldots X_p$ 들의 linear combination

→ 마찬가지로, $Z_i$ 는 Loading 행렬의 i 번째 loading column $\varnothing_i$ 를 weight로 하는 feature $X_1, X_2, \ldots X_p$ 들의 linear combination

## R의 "prcomp" 함수 :

- pc <- prcomp( ) : Score, 즉 rotate된 principal component (new) data는 pc$x 임
- pc$x == orignal_data X rotation_matrix 가 정확히 일치하려면 "original_data"가 standardized 되어야 함.

Score (Principal Components) = Original Data x Rotation

$$\underset{n \times p}{\begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1p} \\ z_{21} & z_{22} & & z_{2p} \\ \vdots & & & \\ z_{n1} & z_{n2} & \cdots & z_{np} \end{bmatrix}} = \underset{n \times p}{\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & & x_{2p} \\ \vdots & & & \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}} \times \underset{p \times p}{\begin{bmatrix} \varnothing_{11} & \varnothing_{12} & \cdots & \varnothing_{1p} \\ \varnothing_{21} & \varnothing_{22} & & \varnothing_{2p} \\ \vdots & & & \\ \varnothing_{p1} & \varnothing_{p2} & \cdots & \varnothing_{pp} \end{bmatrix}}$$

- Rotation $\varnothing_1$ is weights for variables X for the first principal component

# Loading 행렬 – 첫번째 loading 구하기,

- 첫 번째 PC의 score인 $\mathbf{Z_1} = [\, z_{11}, z_{21}, \ldots, z_{n1} \,]^T$ 분산을 최대로 만들자.
- X의 모든 feature, 즉 column들의 기대값을 0 으로 한다.
  즉 $\frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0$
- 이 경우 $z_{11}, z_{21}, \ldots, z_{n1}$ 들의 기대값도 0이 된다.
- 따라서 $z_{11}, z_{21}, \ldots, z_{n1}$ 의 가장 큰 분산값을 찾는 $\mathbf{Z_1}$ 구하기는 다음과 같다

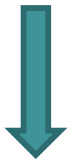
$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}} \left\{ \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2 \right\} \text{ subject to } \sum_{j=1}^{p}\phi_{j1}^2 = 1$$

을 최대화하는 $Ø_{11}$ , $Ø_{21}$ , $\ldots Ø_{p1}$ 들을 구한다

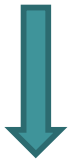
- 위 문제는 행렬 **X** 에 대한 singular value decomposition 방법으로도 풀 수 있다

# Loading(Rotation) 행렬 – Covariance matrix의 eigenvector들

Data Matrix : Feature 들로 된 nxp Matrix (n: observation 수,  p: feature 수)

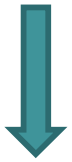


Covariance Matrix : Feature 들간 Covariance를 나타내는 pxp matrix



eigen
decomposition

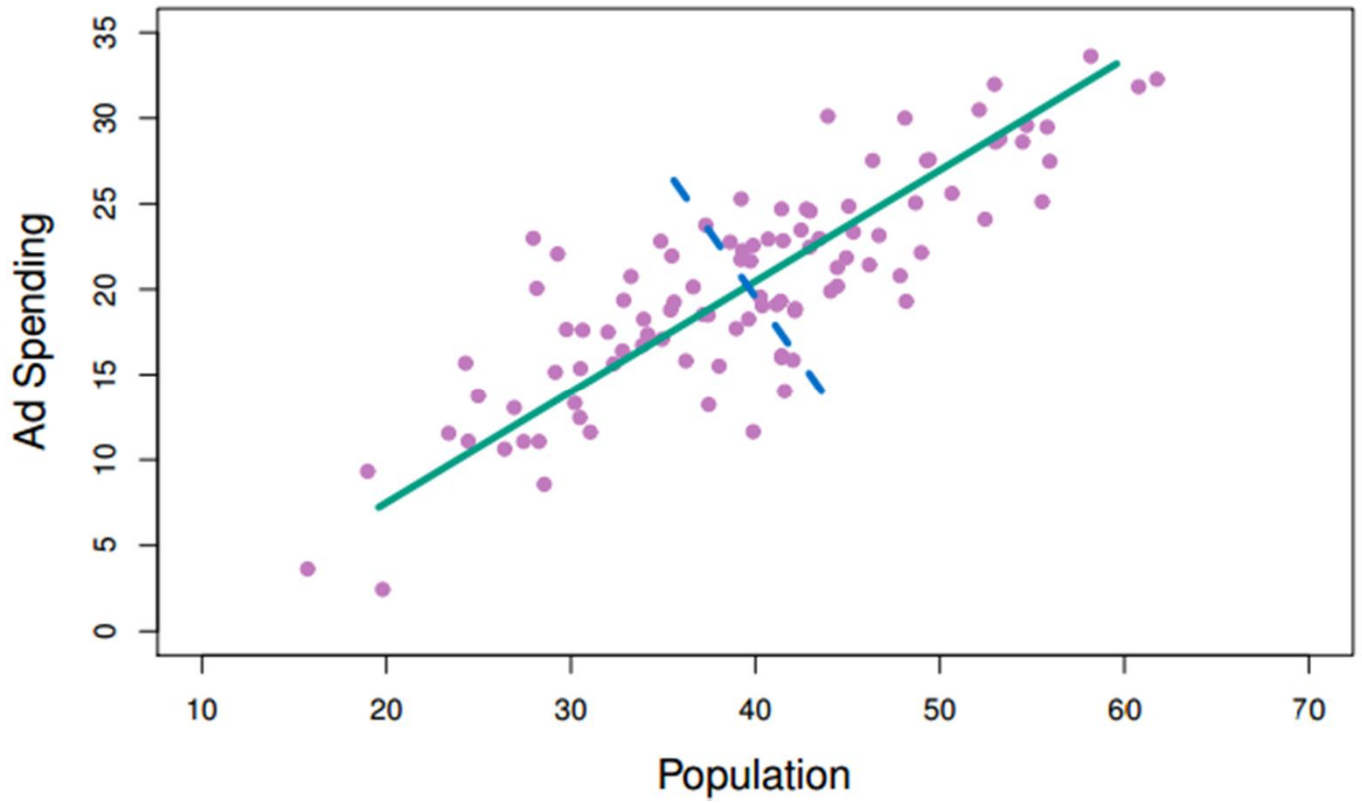
eigenvalue, eigenvector



eigenvalue가 큰 것 부터 eigenvector 정렬

Ø : Loading Matrix

# Principal Component 의 기하학적 의미




n = 100
p = 2
Min(99, 2) =2

$\varnothing_1 = [\varnothing_{11} \quad \varnothing_{21}]^T$
$= [0.839 \quad 0.544]^T$

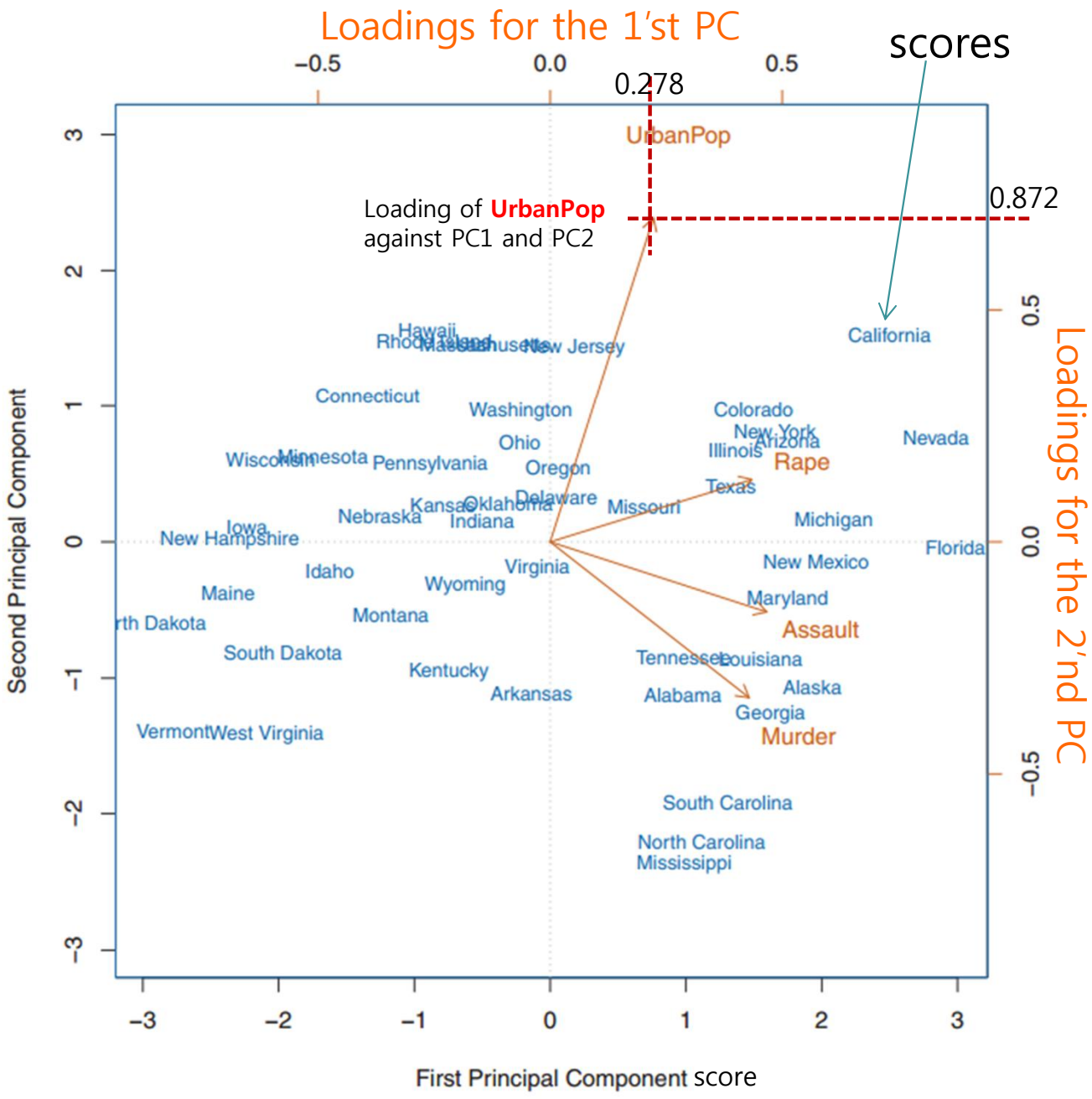
$\varnothing_2 = [\varnothing_{12} \quad \varnothing_{22}]^T$
$= [0.544 \quad -0.839]^T$

$\varnothing_1$ 과 $\varnothing_2$ 은 직각

- 첫번째 principal component의 loading vector $\varnothing_1$ (초록색 선) 은 feature space에 위치한 데이터 점들이 가장 많이 변화하는 (분산도가 최고인) 방향을 나타낸다
- 두번째 principal component $\mathbf{Z_2}$ (**파란 점선**) 역시 p개의 feature vector $\mathbf{X_1, X_2, \ldots, X_p}$의 linear combination인데, **$Z_1$ 와 uncorrelated**인 것 중 가장 큰 variance를 갖는 방향이다

# Principal Component Analysis 시각화

- `USAarrests` data: For each of the fifty states in the United States, the data set contains the number of arrests per $100,000$ residents for each of three crimes: `Assault`, `Murder`, and `Rape`. We also record `UrbanPop` (the percent of the population in each state living in urban areas).
- The principal component score vectors have length $n = 50$, and the principal component loading vectors have length $p = 4$.
- PCA was performed after standardizing each variable to have mean zero and standard deviation one.




## PCA loadings(Rotations in R)

| | PC1 | PC2 |
|---|---|---|
| Murder | 0.5358995 | -0.4181809 |
| Assault | 0.5831836 | -0.1879856 |
| UrbanPop | 0.2781909 | 0.8728062 |
| Rape | 0.5434321 | 0.1673186 |

**첫번째 principal component PC1**은 Murder, Assult, Rape feature 들에게 0.55 근방의 모두 유사한 값(loading)을 주었고, UrbanPop에는 다른 3개의 loading과 떨어진 값 0.278을 주었다. 이런 식으로 feature들의 loading/weight를 주면 observation들의 가장 큰 variation 방향을 잡을 수 있다. PC1은 전반적인 범죄발생률 포착하는 것 같다

**두번째 principal component PC2 는 UrbanPopu**lation에 큰 플러스 loading을 주었고, Murder에는 상당한 마이너스값을 주었다. PC2는 따라서 전반적인 도시화 경향을 포착하며, 도시화와 Murder는 반대의 특성을 보인다고 할 수 있겠다

PC1과 PC2에 대해 모두 큰 **positive score**를 보인 California는 범죄율과 도시화가 크고, Vermont/West Virginia는 그 반대이다

# Principal Component Analysis : Scaling the Variables

1. PCA를 수행하기 변수들의 평균을 0 으로 해야 한다

2. PCA는 feature들의 scale에 영향 받음. 따라서, 변수들의 표준편차가 1이 되게끔 변수들의 값을 scale함이 좋다

3. 하지만, 변수들의 단위가 같다면, 경우에 따라 scale할 수도, 하지 않는 것이 좋을 수도 있다

# 각 Principal Component가 얼마나 데이터를 설명하나?

1. 첫번째 컴포너트 PC1이 원 데이터의 variance를 가장 많이 설명하기에 가장 중요하다

2. 그러면, 앞으로부터 몇 개의 principal component를 채택하면 원 데이터의 정보를 충분히 포함하도록 할 수 있을까?

3. Proportion of Variance Explained (PVE) : 개별 principal component가 원 데이터의 전체 variance중 얼마를 설명하고 있나를 나타내는 값으로 0 ~ 1 사이의 값을 갖는다. 가령 PC1 의 PVE=0.75라면 PC1이 원 데이터의 variance(정보)의 75%를 설명/포함

# 몇개의 Principal Component를 사용하면 되나?

1. 정답은 없다. 분명한 것은 모든 PC들의 PVE을 합치면 1이 되기는 할 것이다. $Z_1$부터 시작해 몇 번째 PC까지 포함할 까는 그 때 그 때 상황에 맞추는 수 밖에 없다. 만약 PVE 값이 급격히 낮아지는 지점이 보이면 해당 지점까지의 PC score만 이용하면 되겠다.

2. PCA를 한 후 Supervised Learning을 할 생각이면, 일부 앞쪽에 있는 score vector $Z_M$ 들만을 사용할 수도 있다. 즉, n x p 전체 데이터 말고 n x M (where M < p) $[Z_1 \ Z_2 \ldots Z_M]$ 행렬을 이용해 regression 이나 classification을 하게 된다. 이 경우 M 이 p 보다 작으므로 차원이 줄어들어(feature 가 줄어들어) cross-validation이나 supervised 모델이나 clustering을 할 때도 유리할 수 있다. **무조건 PCA부터 하고 시작하지는 말자**