Data Science for Actuaries (ACT6100)

Arthur Charpentier

Non-Supervisé # 1.1 (ACP - 1)

automne 2020

https://github.com/freakonometrics/ACT6100/

ACP, projeter optimalement n points de $\mathbb{R}^k$ dans $\mathbb{R}^2$

PCA with R & References

- Lebart, Piron & Morineau (2006) Statistique exploratoire multidimensionnelle, Dunod
- Husson, Lê & Pagès (2016) Analyse de données avec R, chapitre 1, PUR
- Husson, Lê & Pagès (2017) Exploratory Multivariate Analysis by Example Using R, chapter 1, CRC
- Jolliffe (2002) Principal Component Analysis, Springer

```
> stats::prcomp()
> stats::princomp()
> FactoMineR::PCA()
> ade4::dudi.pca()
> ExPosition::epPCA()
```

package R fonction R

```
> library(factoextra)
```

Dataset (Countries)

```
> url = "http://user.math.uzh.ch/furrer/download/
   sta121/europejobs.csv"
> jobs = read.csv(url, header=TRUE)
```

cf europejobs.csv

	Agr	Min	Man	PS	Con	SI	Fin	SPS	TC
Belgium	3.3	0.9	27.6	0.9	8.2	19.1	6.2	26.6	7.2
Denmark	9.2	0.1	21.8	0.6	8.3	14.6	6.5	32.2	7.1
France	10.8	0.8	27.5	0.9	8.9	16.8	6.0	22.6	5.7
W. Germany	6.7	1.3	35.8	0.9	7.3	14.4	5.0	22.3	6.1
...	...	...	...	...	...	...	...	...	...

Percentages of employment in: agriculture (`Agr`), mining (`Min`), manufacturing (`Man`), power supply industries (`PS`), construction (`Con`), service industries (`SI`), finance (`Fin`), social and personal services (`SPS`), and transport and communications (`TC`),

Dataset (Countries)

Here is the correlation matrix of $\boldsymbol{X}$,

	Agr	Min	Man	PS	Con	SI	Fin	SPS	TC
Agr	**1.00**	0.04	-0.67	-0.40	-0.54	-0.74	-0.22	-0.75	-0.56
Min	0.04	**1.00**	0.44	0.40	-0.03	-0.40	-0.44	-0.28	0.16
Man	-0.67	0.44	**1.00**	0.38	0.49	0.20	-0.16	0.15	0.35
PS	-0.40	0.40	0.38	**1.00**	0.06	0.20	0.11	0.13	0.38
Con	-0.54	-0.03	0.49	0.06	**1.00**	0.36	0.02	0.16	0.39
SI	-0.74	-0.40	0.20	0.20	0.36	**1.00**	0.37	0.57	0.19
Fin	-0.22	-0.44	-0.16	0.11	0.02	0.37	**1.00**	0.11	-0.25
SPS	-0.75	-0.28	0.15	0.13	0.16	0.57	0.11	**1.00**	0.57
TC	-0.56	0.16	0.35	0.38	0.39	0.19	-0.25	0.57	**1.00**

```
> r = cor(jobs)
> library(corrplot)
```

Dataset (Countries)

```
> corrplot(r, method = "square")
> corrplot(r, method = "number")
> corrplot(r, type="lower")
```

Dataset (Countries)

Consider the following sub-dataset, $\boldsymbol{X}$, with $n = 6$ observations, $k = 3$ variables,

	Agr	Min	Man
Belgium	3.3	0.9	27.6
Denmark	9.2	0.1	21.8
France	10.8	0.8	27.5
W. Germany	6.7	1.3	35.8
Ireland	23.2	1.0	20.7
Italy	15.9	0.6	27.6

- $\boldsymbol{x}_i = \boldsymbol{x}_{i\cdot} \in \mathbb{R}^k$ is row i (observation)
- $\boldsymbol{x}^j = \boldsymbol{x}_{\cdot j} \in \mathbb{R}^n$ is column j (variable)

raw data		centered		scaled
$\boldsymbol{x}$	$\longleftrightarrow$	$\boldsymbol{y}$	$\longleftrightarrow$	$\boldsymbol{z}$

Dataset Transformation

Let $\overline{x}^j = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} \boldsymbol{x}_{ij}$ and $s_j^2 = \dfrac{1}{n-1}\displaystyle\sum_{i=1}^{n}\left(\boldsymbol{x}_{ij} - \overline{x}^j\right)^2$

Define $\boldsymbol{y}_{ij} = \boldsymbol{x}_{ij} - \overline{x}^j$ (centered) and $\boldsymbol{z}_{ij} = \dfrac{\boldsymbol{x}_{ij} - \overline{x}^j}{s_j}$

Given a distance d, $d(\boldsymbol{y}_i, \boldsymbol{y}_{i'}) = d(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) \neq d(\boldsymbol{z}_i, \boldsymbol{z}_{i'})$ (scaled)

The inertia of $\boldsymbol{X}$, associated with d is

$$\mathcal{I}_d(\boldsymbol{X}) = \sum_{i=1}^{n} d^2\left(\boldsymbol{x}_i, \overline{\boldsymbol{x}}\right),\ \overline{\boldsymbol{x}} = (\overline{x}^1, \cdots, \overline{x}^k)$$

If $d = d_\Delta$ for some diagonal matrix $\boldsymbol{\Delta}$, so that $d_\Delta(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{a}^\top \boldsymbol{\Delta} \boldsymbol{b}$

$$\mathcal{I}_\Delta(\boldsymbol{X}) = \sum_{i=1}^{n} \Delta_{j,j} \mathrm{Var}(\boldsymbol{x}^j)$$

Note that $\mathcal{I}_{\mathbb{I}}(\boldsymbol{Z}) = k$.

Dataset (Countries)

Raw data $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_6\}$ in $\mathbb{R}^3$
scatterplot of individual observations
(in the space of variables)

	Agr	Min	Man
Belgium	3.3	0.9	27.6
Denmark	9.2	0.1	21.8
France	10.8	0.8	27.5
W. Germany	6.7	1.3	35.8
Ireland	23.2	1.0	20.7
Italy	15.9	0.6	27.6

Dataset (Countries)

Scaled data $\boldsymbol{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_6\}$ in $\mathbb{R}^3$
scatterplot of individual observations
(in the space of variables)

	Agr	Min	Man
Belgium	-1.16	0.29	0.14
Denmark	-0.33	-1.68	-0.93
France	-0.10	0.04	0.12
W.Germany	-0.68	1.27	1.67
Ireland	1.64	0.53	-1.14
Italy	0.62	-0.45	0.14

Dataset (Countries)

Scaled data $\boldsymbol{Z} = \{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^6\}$ in $\mathbb{R}^3$
scatterplot of scaled variables
(in the space of individuals)

	Belgium	Denmark	France
Agr	-1.13	0.36	0.77
Min	0.69	-1.15	0.46
Man	0.59	-1.15	0.56
PS	0.58	-1.15	0.58
Con	-0.70	-0.44	1.14
SI	1.01	-0.99	-0.01

Dataset (Countries)

Scaled data $\boldsymbol{Z} = \{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^6\}$ in $\mathbb{R}^3$
scatterplot of scaled variables
(in the space of individuals)
vs.
Scaled data $\boldsymbol{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_6\}$ in $\mathbb{R}^3$
scatterplot of individual observations
(in the space of variables)

Which Distance ?

On peut définir, à partir d'une matrice définie positive $\boldsymbol{M}$

- un produit scalaire : $< \boldsymbol{a}, \boldsymbol{b} >_{\boldsymbol{M}} = \boldsymbol{a}^\top \boldsymbol{M} \boldsymbol{b}$;
- une norme: $\|\boldsymbol{a}\|^2_{\boldsymbol{M}} = < \boldsymbol{a}, \boldsymbol{a} >_{\boldsymbol{M}}$
- une distance: $d_{\boldsymbol{M}}(\boldsymbol{a}, \mathbf{b}) = \|\boldsymbol{a} - \boldsymbol{b}\|_{\boldsymbol{M}}$.

$d_{\boldsymbol{I}}$ is the standard Euclidean distance

Let $\boldsymbol{\Sigma} = \text{Var}[\boldsymbol{X}]$, then $d_{\boldsymbol{\Sigma}^{-1}}$ is a distance.

Let $\boldsymbol{\Sigma}_\Delta = \text{diag}(\text{Var}[\boldsymbol{X}])$, then $d_{\boldsymbol{\Sigma}_\Delta^{-1}}(\boldsymbol{x}_i, \mathbf{x}_j) = d_{\mathbb{I}}(\boldsymbol{z}_i, \mathbf{z}_j)$, called scaled distance, or Mahalonobis distance.

Spectral Decomposition

Soit $\boldsymbol{M}$ une matrice $k \times k$. Les k couples de valeurs propres (λ_i) et de vecteurs propres $(\boldsymbol{a}_i)$ forment la décomposition spectrale d'une matrice donnée.

Les valeurs propres correspondent aux k solutions possibles de l'équation :

$$\det(\boldsymbol{M} - \lambda \mathbb{I}_k) = 0$$

avec $k = \text{rang}(\boldsymbol{M})$ le nombre maximal de lignes/colonnes indépendantes dans la matrice considérée. Le vecteur propre associé à la valeur propre λ_i est donné par

$$\boldsymbol{M}\boldsymbol{a}_i = \lambda_i \boldsymbol{a}_i.$$

Notons que

$$\sum_{j=1}^{k} \lambda_j = \text{tr}(\boldsymbol{M}) \text{ et } \prod_{j=1}^{k} \lambda_j = \det(\boldsymbol{M}).$$

Spectral Decomposition

Example:
Spectral decomposition of

$$\boldsymbol{M} = \begin{bmatrix} 7 & 4 \\ 3 & 2 \end{bmatrix}.$$

is

$$\lambda_1 = 8.772 \rightarrow \mathbf{a}_1 = \begin{bmatrix} 0.914 \\ 0.405 \end{bmatrix}$$

$$\lambda_2 = 0.228 \rightarrow \mathbf{a}_2 = \begin{bmatrix} -0.509 \\ 0.8610 \end{bmatrix}.$$

```
> M=c(7,3,4,2)
> dim(M)=c(2,2)
> M
     [,1] [,2]
[1,]    7    4
[2,]    3    2
> eigen(M)
$values
[1] 8.7720019 0.2279981

$vectors
          [,1]       [,2]
[1,] 0.9143009 -0.5085750
[2,] 0.4050357  0.8610177
```

Spectral Decomposition

Example:

$$\boldsymbol{M} = \begin{bmatrix} 7 & 4 & 6 \\ 3 & 2 & 1 \end{bmatrix}.$$

then $\boldsymbol{M}^\top \boldsymbol{M}$ and $\boldsymbol{M}\boldsymbol{M}^\top$ have the "same" eigenvalues

```
> M=c(7,3,4,2,6,1)
> dim(M)=c(2,3)
> M
     [,1] [,2] [,3]
[1,]    7    4    6
[2,]    3    2    1
```

```
> eigen(M%*%t(M))
$values
[1] 113.332338   1.667662

$vectors
          [,1]     [,2]
[1,] -0.94316  0.33232
[2,] -0.33232 -0.94316
> eigen(t(M)%*%M)
$values
[1]  113.33  1.66 -5e-15

$vectors
       [,1]   [,2]   [,3]
[1,] -0.714 -0.390  0.582
[2,] -0.417 -0.431 -0.800
[3,] -0.563  0.814 -0.145
```

Spectral Decomposition

Example:
Si $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{V}^\top$ (SVD)

- $\boldsymbol{U} = [\vec{\boldsymbol{a}}_1, \cdots, \vec{\boldsymbol{a}}_k]$
- $\boldsymbol{V} = [\vec{\boldsymbol{b}}_1, \cdots, \vec{\boldsymbol{b}}_n]$
- $\delta_i = \sqrt{\lambda_i}$

```
> svd(M)$d
[1] 10.64577  1.29138
> svd(M)$d^2
[1] 113.3323  1.66766
```

```
> svd(M)
$d
[1] 10.64577  1.29138

$u
          [,1]      [,2]
[1,] -0.94316 -0.33232
[2,] -0.33232  0.94316

$v
          [,1]      [,2]
[1,] -0.71381  0.38966
[2,] -0.41681  0.43133
[3,] -0.56278 -0.81369
```

Spectral Decomposition

If $\vec{\boldsymbol{a}}$ est un vecteur propre de $\boldsymbol{M}^\top\boldsymbol{M}$, pour la valeur propre λ,

$$\boldsymbol{M}^\top\boldsymbol{M}\vec{\boldsymbol{a}} = \lambda\vec{\boldsymbol{a}}$$

$$\boldsymbol{M}\boldsymbol{M}^\top\underbrace{\boldsymbol{M}\vec{\boldsymbol{a}}}_{\vec{\boldsymbol{b}}} = \lambda\underbrace{\boldsymbol{M}\vec{\boldsymbol{a}}}_{\vec{\boldsymbol{b}}}$$

$\vec{\boldsymbol{b}} = \boldsymbol{M}\vec{\boldsymbol{a}}$ est un vecteur propre de $\boldsymbol{M}\boldsymbol{M}^\top$, pour la valeur propre λ.

If $\vec{\boldsymbol{b}}$ est un vecteur propre de $\boldsymbol{M}\boldsymbol{M}^\top$, pour la valeur propre λ,

$$\boldsymbol{M}\boldsymbol{M}^\top\vec{\boldsymbol{b}} = \lambda\vec{\boldsymbol{b}}$$

$$\boldsymbol{M}^\top\boldsymbol{M}\underbrace{\boldsymbol{M}^\top\vec{\boldsymbol{b}}}_{\vec{\boldsymbol{\alpha}}} = \lambda\underbrace{\boldsymbol{M}^\top\vec{\boldsymbol{b}}}_{\vec{\boldsymbol{\alpha}}}$$

$\vec{\boldsymbol{\alpha}} = \boldsymbol{M}^\top\vec{\boldsymbol{b}}$ est un vecteur propre de $\boldsymbol{M}^\top\boldsymbol{M}$ pour la valeur propre λ.

Analyse en Composantes Principales

- Une analyse en composantes principales peut se faire à partir des données centrées ($\mathbf{Y}$) ou des données centrée-réduites ($\mathbf{Z}$).
- Une analyse en composantes principales se fait en étudiant deux nuages: le nuage des n observations dans $\mathbb{R}^k$ avec la métrique $\mathbf{M} = \mathbb{I}_k$ et le nuage des k variables dans $\mathbb{R}^n$ avec la métrique $\mathbf{N} = (1/n)\mathbb{I}_n$.
- Dans la très grande majorité des cas, on réalise une *analyse en composantes principales normée*, c'est-à-dire à partir de la matrice des corrélations $\mathbf{R}$ construites à partir des données centrées-réduites $\mathbf{Z}$.

Inertie

- L'inertie d'un nuage d'observations est une mesure de la dispersion de ce même nuage. Elle est définie par

$$\mathcal{I}(\mathbf{X}) = \sum_{i=1}^{n} w_i d^2_{\mathbf{M}}(\mathbf{x}_i^T, \bar{\mathbf{x}}), \quad \text{avec } \bar{\mathbf{x}} = \begin{bmatrix} \bar{x}^1 \\ \vdots \\ \bar{x}^k \end{bmatrix}.$$

Proposition Si on choisit une métrique diagonale $\mathbf{M} = \text{diag}(m_1, \ldots, m_k)$ et $w_i = 1/n$, $\forall n$, on obtient

$$\mathcal{I}(\mathbf{X}) = \sum_{i=1}^{k} m_j \text{Var}(\mathbf{x}^j).$$

Si on utilise $\mathbf{M} = \mathbb{I}_k$, alors $\mathcal{I}(\mathbf{Y}) = \sum_{i=1}^{k} \text{Var}(\mathbf{x}^j)$ et $\mathcal{I}(\mathbf{Z}) = k$.

Projection d'observations (sur un plan) ?

Projection d'une observation (sur une droite)

- Dans un premier temps, on cherche un axe défini par un vecteur $\mathbf{v}_1 \in \mathbb{R}^k$ tel que

$$\mathbf{v}_1 = \underset{\mathbf{v}\in\mathbb{R}^k;\, ||\mathbf{v}||=1}{\text{argmax}} \left\{\text{Var}(\mathbf{Z}\mathbf{v})\right\}.$$

- Ce problème d'optimisation sous contraintes peut se réécrire sous la forme

$$\mathbf{v}_1 = \underset{\mathbf{v}\in\mathbb{R}^k;\, ||\mathbf{v}||=1}{\text{argmax}} \left\{\mathbf{v}^\top \mathbf{R}\mathbf{v}\right\},$$

  où $\mathbf{R} = (1/n)\mathbf{Z}^\top\mathbf{Z}$ est la matrice de corrélation.
- On peut démontrer que la solution à ce problème d'optimisation est le vecteur propre $\mathbf{a}_1$ obtenu par une décomposition spectrale de la matrice $\mathbf{R}$.

ACP et optimisation

On veut résoudre

$$\max_{\boldsymbol{v}} \left\{ \sum_{i=1}^{n} \langle \boldsymbol{z}_i, \boldsymbol{v} \rangle \right\} = \max_{\boldsymbol{v}} \left\{ \boldsymbol{v}^\top \boldsymbol{Z}^\top \boldsymbol{Z} \boldsymbol{v} \right\} \text{ subject to } \boldsymbol{v}^\top \boldsymbol{v} = 1$$

Le Lagrangien associé est

$$\mathcal{L}(\boldsymbol{v}, \lambda) = \boldsymbol{v}^\top \boldsymbol{Z}^\top \boldsymbol{Z} \boldsymbol{v} - \lambda(1 - \boldsymbol{v}^\top \boldsymbol{v})$$

$$\nabla_{\boldsymbol{v},\lambda} \mathcal{L}(\boldsymbol{v}, \lambda) = 0 \Leftrightarrow \begin{cases} \boldsymbol{Z}^\top \boldsymbol{Z} \boldsymbol{v} = \lambda \boldsymbol{v} \\ \boldsymbol{v}^\top \boldsymbol{v} = 1 \end{cases}$$

De la première équation, $\boldsymbol{v}$ est un vecteur propre de $\boldsymbol{Z}^\top \boldsymbol{Z}$.
$\boldsymbol{v} = \boldsymbol{a}_1$ est le vecteur propre associé à la première valeur propre.

Projection d'une observation (sur une droite)

- Dans un second temps, on cherche un axe défini par un vecteur $\mathbf{v}_2 \in \mathbb{R}^k$ tel que

$$\mathbf{v}_2 = \underset{\mathbf{v}\in\mathbb{R}^k;\, ||\mathbf{v}||=1;\, \mathbf{v} \text{ et } \mathbf{v}_1 \text{ non corr.}}{\operatorname{argmax}} \{\operatorname{Var}(\mathbf{Z}\mathbf{v})\}.$$

- On peut démontrer que la solution à ce problème d'optimisation est le vecteur propre $\mathbf{a}_2$ obtenu par une décomposition spectrale de la matrice $\mathbf{R}$.

Pour $1 \leq q \leq r = \operatorname{rang}(\mathbf{Z})$, on construit les composantes principales $\mathbf{p}_1, \ldots, \mathbf{p}_q$ qui formeront une droite ($q = 1$), un plan ($q = 2$), etc. sur lequel on projettera les observations.

Projection d'une observation (sur une droite)

- Les composantes principales $(\mathbf{p}_\alpha, \alpha = 1, \ldots, q)$ sont q nouvelles variables telles que
  - les corrélations entre les composantes principales sont nulles
  - la variance est $\text{Var}(\mathbf{p}_\alpha) = \lambda_\alpha$.
- Les q premières composantes principales représentent une inertie totale de

$$\mathcal{I}(\mathbf{p}_1, \ldots, \mathbf{p}_q) = \lambda_1 + \ldots + \lambda_q.$$

Projection d'une variable (sur une droite)

- La projection (**N**-orthogonale) d'une variable $\mathbf{z}^j \in \mathbb{R}^n$ sur un axe G_β décrit par un vecteur $\mathbf{u}_\beta$ de norme 1, c'est-à-dire tel que

$$\mathbf{u}_\beta^\top \mathbf{N} \mathbf{u}_\beta = 1,$$

a pour coordonnée

$$t_{j\beta} =< \mathbf{z}^j, \mathbf{u}_\beta >= (\mathbf{z}^j)^\top \mathbf{N} \mathbf{u}_\beta.$$

- Pour l'ensemble de l'échantillon, on notera

$$\mathbf{t}_\beta = \begin{bmatrix} t_{1\beta} \\ \vdots \\ t_{k\beta} \end{bmatrix} = \mathbf{Z}^\top \mathbf{N} \mathbf{u}_\beta.$$

Projection d'une variable

- Dans un premier temps, on cherche un axe défini par un vecteur $\mathbf{u}_1 \in \mathbb{R}^n$ tel que

$$\mathbf{u}_1 = \underset{\mathbf{u}\in\mathbb{R}^n;\, ||\mathbf{u}||=1}{\text{argmax}} \left\{ ||\mathbf{Z}^\top \mathbf{N}\mathbf{u}||^2 \right\}.$$

- En posant $\mathbf{N} = (1/n)\mathbb{I}_n$, on peut démontrer que la solution à ce problème d'optimisation est le vecteur propre $\mathbf{b}_1$ obtenu par une décomposition spectrale de la matrice $(1/n)\mathbf{Z}^\top\mathbf{Z}$.

Sélection des vecteurs $\mathbf{u}$

- Dans un second temps, on cherche un axe défini par un vecteur $\mathbf{u}_2 \in \mathbb{R}^n$ tel que

$$\mathbf{u}_2 = \underset{\mathbf{u}\in\mathbb{R}^n;\, ||\mathbf{u}||=1;\, \mathbf{u} \text{ et } \mathbf{u}_1 \text{ non corr.}}{\operatorname{argmax}} \left\{ ||\mathbf{Z}^\top \mathbf{N}\mathbf{u}||^2 \right\}.$$

- La solution à ce problème d'optimisation est le vecteur propre $\mathbf{b}_2$ obtenu par une décomposition spectrale de la matrice $(1/n)\mathbf{Z}^\top\mathbf{Z}$.

Pour $1 \leq q^* \leq r^* = \text{rang}(\mathbf{Z}^\top\mathbf{Z})$, on construit les composantes principales $\mathbf{t}_1, \ldots, \mathbf{t}_{q^*}$ qui formeront une droite ($q^* = 1$), un plan ($q^* = 2$), etc. sur lequel on projettera les variables.

Qualité de la projection

- La qualité de la représentation est mesurée par le pourcentage de l'inertie intiale des données que les q premières composantes principales permettent d'expliquer:

$$\mathcal{I}(\mathbf{Z}) = \lambda_1 + \ldots + \lambda_r = p$$
$$\mathcal{I}(\mathbf{p}_1, \ldots, \mathbf{p}_q) = \lambda_1 + \ldots + \lambda_q \leq \mathcal{I}(\mathbf{Z}).$$

- La $i$ème composante principale permet d'expliquer une proportion de

$$\frac{\lambda_i}{\lambda_1 + \ldots + \lambda_r}$$

de l'information initiale.

- Le passage de r à q dimension (avec $q < r$) se fait en préservant une proportion de

$$\frac{\lambda_1 + \ldots + \lambda_q}{\lambda_1 + \ldots + \lambda_r}$$

de l'information initiale.

Qualité de la représentation des observations

- Pour deux observations "bien projetées", la distance en projection est similaire à la distance dans le nuage initiale $\mathbb{R}^k$.
- La qualité de la projection de l'observation i sur l'axe D_α est mesurée par le carré du cosinus de l'angle $\theta_{i\alpha}$ formé entre le vecteur $\mathbf{z}_i$ et l'axe D_α:

$$\cos^2(\theta_{i\alpha}) = \frac{\mathbf{p}_{i\alpha}^2}{||\mathbf{z}_i||^2}.$$

- Si l'espace final est un plan formé des axes D_1 et D_2, la qualité de la projection de l'observation i sur ce plan est donnée par

$$\cos^2(\theta_{i(1,2)}) = \frac{\mathbf{p}_{i1}^2 + \mathbf{p}_{i2}^2}{||\mathbf{z}_i||^2}.$$

- Plus cette valeur est près de 1, meilleure est la projection.

Qualité de la représentation des observations

- Les observations qui contribuent de manière élevée à la construction des axes sont sources d'instabilité. La contribution d'une observation à un axe est mesurée par la proportion de l'inertie de l'axe expliquée par cette observation.
- Pour l'axe D_α, l'inertie totale est $\lambda_\alpha = \sum_{i=1}^{n} \mathbf{p}_{i\alpha}^2$.
- La contribution de l'observation i sur l'axe D_α est mesurée par

$$C(i, \alpha) = \frac{\mathbf{p}_{i\alpha}^2}{\lambda_\alpha}.$$

- Si l'espace final est un plan formé des axes D_1 et D_2, la contribution de l'observation i sur ce plan est donnée par

$$C(i, (1, 2)) = \frac{\mathbf{p}_{i1}^2 + \mathbf{p}_{i2}^2}{\lambda_1 + \lambda_2}.$$

Troncation

- Exponential decay

the variance λ_j of Gaussian noise decays in an exponential manner

Example valeurs propres de $\widehat{\boldsymbol{\Sigma}}$, matrice de variance de $\{\boldsymbol{X}_1, \cdots, \boldsymbol{X}_n\}$ où les $\boldsymbol{X}_i \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_d)$ (ici $d = 1000$)

Trouver k tel que λ_j, pour $j > k$, décroit exponentiellement vite...

- North's rule of thumb

truncate the expansion when the sampling error becomes larger than the distance between two eigenvalues, i.e., when $\lambda_j - \lambda_{j+1} < \sqrt{2/n}\lambda_j$

Countries (subset, 6×3)

```
> X = jobs[1:6,1:3]
> Z=X
> for(j in 1:3) Z[,j]=(X[,j
    ]-mean(X[,j]))/sd(X[,j])
> (R=cor(Z))
       Agr   Min   Man
Agr   1.00 -0.01 -0.56
Min  -0.01  1.00  0.61
Man  -0.56  0.61  1.00
> eigen(R)
eigen() decomposition
$values
[1] 1.83 0.99 0.18

$vectors
     [,1] [,2]  [,3]
[1,]  0.48 0.74 -0.47
[2,] -0.52 0.67  0.52
[3,] -0.70 0.00 -0.71
```

```
> L=eigen(R)$values
> barplot(cumsum(L)/sum(L))
> barplot(L/sum(L),add=T)
> L/sum(L)*100
[1] 61.08983 32.99279
    5.91738
```

Countries (subset, 6×3)

```
> as.matrix(Z)%*%eigen(R)$vectors
            [,1]  [,2]  [,3]
Belgium    -0.80 -0.66  0.60
Denmark     1.38 -1.37 -0.06
France     -0.16 -0.05 -0.02
W.Germany  -2.16  0.36 -0.20
Ireland     1.31  1.57  0.31
Italy       0.43  0.15 -0.63
> phi=eigen(R)$vectors[,1:2]
> row.names(phi) = names(Z)
> colnames(phi) = c("PC1","PC2")
> PC1 = as.matrix(Z) %*% phi[,1]
> PC2 = as.matrix(Z) %*% phi[,2]
```

```
> plot(PC1,PC2)
> text(PC1,PC2,rownames(Z),pos=3)
```

Countries (subset, 6×3)

```
> M=(as.matrix(Z)) %*% t(as
    .matrix(Z))
> eigen(M)
eigen() decomposition
$values
[1] 9.16 4.95 0.89
[4] 0.00 0.00 0.00

$vectors
      [,1]  [,2]  [,3]  [,4
[1,]  0.27 -0.30  0.63 -0.6
[2,] -0.46 -0.62 -0.06  0.0
[3,]  0.05 -0.02 -0.02  0.0
[4,]  0.71  0.16 -0.21  0.0
[5,] -0.43  0.71  0.32 -0.1
[6,] -0.14  0.07 -0.67 -0.7
```

```
> L = eigen(M)$values
> barplot(cumsum(L)/sum(L))
> barplot(L/sum(L),add=T)
> L/sum(L)*100
[1] 61.089 32.992  5.917
[4]  0.000  0.000  0.000
```

Countries (subset, 6×3)

```
> t(as.matrix(Z))%*%eigen(M)$vectors
      [,1] [,2]  [,3] [,4]
Agr -1.45 1.64 -0.45    0
Min  1.59 1.50  0.49    0
Man  2.13 0.00 -0.67    0
> phi = eigen(M)$vectors[,1:2]
> row.names(phi) = rownames(Z)
> colnames(phi) = c("PC1","PC2")
> PC1 = t(as.matrix(Z)) %*% phi[,1]
> PC2 = t(as.matrix(Z)) %*% phi[,2]
```

```
> plot(PC1,PC2)
> text(PC1,PC2,colnames(Z),pos=3)
> arrows(0,0,PC1,PC2)
```

PCA, in a nutshell

Double lecture

- analyse des individus
  dans l'espace des variables
- analyse des variables
  dans l'espace des individus

via Escofier & Pagès (2008)

Pour la suite, des applications...

Countries (subset, 6 × 3)

```
> M = jobs[1:6,1:3]
> M
            Agr Min  Man
Belgium     3.3 0.9 27.6
Denmark     9.2 0.1 21.8
France     10.8 0.8 27.5
W.Germany   6.7 1.3 35.8
Ireland    23.2 1.0 20.7
Italy      15.9 0.6 27.6
> names(M) = rownames(jobs)[1:3]
> library(FactoMineR)
> res.pca = PCA(M, graph = FALSE)
> library("factoextra")
> eig.val = get_eigenvalue(res.pca)
> eig.val
   eigenvalue   var.pct   cum.pct
Dim.1  1.8326    61.089    61.089
Dim.2  0.9897    32.992    94.082
Dim.3  0.1775     5.917   100.000
> fviz_eig(res.pca)
```

Cercle des corrélations

- Si deux variables sont bien projetées, alors leur angle en projection est similaire à celui dans l'espace initial ($\mathbb{R}^n$).
- Corrélation entre deux variables = cosinus de l'angle entre les variables centrées-réduites.
  - un angle droit (90 degrés) correspond à une corrélation nulle;
  - un angle nul correspond à une corrélation de 1; et
  - un angle plat (180 degrés) correspond à un corrélation de -1.

- La corrélation entre la variable i et la composante principale j est $\sqrt{\lambda_j} a_{ij}$, où a_{ij} est le $i$ème élément du vecteur propre correspondant à la $j$ème plus grande valeur propre.

Countries (subset, 6 × 3)

```
> fviz_pca_var(res.pca)
```

avec une couleur donction des $\cos^2$

```
> fviz_pca_var(res.pca,
  col.var = "cos2", repel = TRUE)
```

ou de la contribution des variables

```
> fviz_pca_var(res.pca,
  col.var = "contrib")
```

Qualité de la projection des variables

- La qualité de la projection de la variable j sur l'axe G_β est mesurée par le carré du cosinus de l'angle $\theta_{j\beta}$ formé entre le vecteur $\mathbf{z}^j$ et l'axe G_β:

$$\cos^2(\theta_{j\beta}) = \frac{\mathbf{t}_{j\beta}^2}{||\mathbf{z}^j||^2} = \mathbf{t}_{j\beta}^2.$$

- Si l'espace final est un plan formé des axes G_1 et G_2, la qualité de la projection de la variable j sur ce plan est donnée par

$$\cos^2(\theta_{j(1,2)}) = \mathbf{t}_{j1}^2 + \mathbf{t}_{j2}^2.$$

- On obtient ainsi une flèche de longueur $\sqrt{\cos^2(\theta_{j(1,2)})}$ sur le cercle des corrélations pour chacune des variables. Plus la flèche est près de la circonférence du cercle, meilleure est la représentation de la variable.

```
> fviz_cos2(res.pca, choice = "var", axes = 1:2)
```

Contribution des variables

- Les contributions des variables aux axes permettent de donner une interprétation à ceux-ci. Pour déterminer la contribution d'une variable à un axe, on évalue la proportion de l'inertie totale de l'axe expliquée par la variable.
- Pour l'axe G_β, l'inertie totale est $\lambda_\beta = \sum_{j=1}^{p} \mathbf{t}_{j\beta}^2$.
- La contribution de la variable j sur l'axe G_β est mesurée par

$$C(j,\beta) = \frac{\mathbf{t}_{j\beta}^2}{\lambda_\beta}.$$

- Si l'espace final est un plan formé des axes G_1 et G_2, la contribution de la variable j sur ce plan est donnée par

$$C(j,(1,2)) = \frac{\mathbf{a}_{j1}^2 + \mathbf{a}_{j2}^2}{\lambda_1 + \lambda_2}.$$

```
> fviz_contrib(res.pca, choice = "var", axes =
    1:2)
```

Countries (subset, 6×3)

```
> fviz_pca_ind(res.pca)
```

```
> fviz_pca_ind(res.pca,
  col.ind = "cos2", repel = TRUE)
```

```
> fviz_pca_ind(res.pca,
  col.ind = "cos2",
  pointsize = "cos2",
  pointshape = 21)
```

Countries (subset, 6×3)

```
> fviz_contrib(res.pca, choice = "ind", axes = 1:2)
```

```
> fviz_pca_biplot(res.pca, repel = TRUE)
```

Dualité et ACP

- analyse des individus dans l'espace des variables
- analyse des variables dans l'espace des individus

Dualité et ACP

- analyse des individus dans l'espace des variables
- analyse des variables dans l'espace des individus

Countries (entiere dataset, 26 × 9)

```
> res.pca = PCA(jobs, graph = FALSE)
> fviz_eig(res.pca)
> fviz_cos2(res.pca, choice = "var",
    axes = 1:2)
> fviz_contrib(res.pca, choice = "var
    ", axes = 1)
```

```
> cumsum(eigen(cor(jobs))$values)/sum
    (eigen(cor(jobs))$values)*100
[1] 38.74  62.41  74.62  85.67  91.71
```

Countries (entiere dataset, 26 × 9)

```
> fviz_pca_var(res.pca, col.var = "cos2", repel = TRUE)
> fviz_pca_var(res.pca, col.var = "contrib")
```

```
> var <- get_pca_var(res.pca)
> res.km <- kmeans(var$coord, centers = 3, nstart = 25)
> grp <- as.factor(res.km$cluster)
> fviz_pca_var(res.pca, col.var = grp)
```

Countries (entiere dataset, 26 × 9)

```
> fviz_pca_ind(res.pca, pointsize = "cos2", pointshape = 21, repel = TRUE)
> fviz_contrib(res.pca, choice = "ind", axes = 1:2)
```

Countries (entiere dataset, 26 × 9)

```
> fviz_pca_biplot(res.pca, repel = TRUE)
```

Dataset (Decathlon)

	100m	Long jump	Shot put	High jump	400m
	+	—	—	—	+
SEBRLE	11.04	7.58	14.83	2.07	49.81
CLAY	10.76	7.40	14.26	1.86	49.37
KARPOV	11.02	7.30	14.77	2.04	48.37
BERNARD	11.02	7.23	14.25	1.92	48.93
YURKOV	11.34	7.09	15.19	2.10	50.42
...	...	...	...	...	...

	110m hurdle	Discus	Pole vault	Javeline	1500m
	+	—	—	—	+
SEBRLE	14.69	43.75	5.02	63.19	291.7
CLAY	14.05	50.72	4.92	60.15	301.5
KARPOV	14.09	48.95	4.92	50.31	300.2
BERNARD	14.99	40.87	5.32	62.77	280.1
YURKOV	15.31	46.26	4.72	63.44	276.4
...	...	...	...	...	...

```
> library("FactoMineR")
> decathlon2 = decathlon[1:23, 1:10]
```

Decathlon

```
> res.pca <- PCA(decathlon2, graph =
    FALSE)
> res.pca = PCA(M, graph = FALSE)
> library("factoextra")
> fviz_eig(res.pca)
```

Decathlon

```
> facto_summarize(res.pca, "var", axes = 1:2)[,-1]
              Dim.1  Dim.2 coord  cos2 contrib
100m         -0.841 -0.231 0.760 0.760  13.011
Long.jump     0.776  0.335 0.714 0.714  12.217
Shot.put      0.751 -0.033 0.565 0.565   9.662
High.jump     0.623 -0.592 0.738 0.738  12.625
400m         -0.719  0.042 0.519 0.519   8.878
110m.hurdle  -0.783 -0.045 0.615 0.615  10.522
Discus        0.713 -0.050 0.512 0.512   8.756
Pole.vault   -0.216  0.840 0.752 0.752  12.867
Javeline      0.374  0.330 0.249 0.249   4.264
1500m         0.042  0.647 0.421 0.421   7.198
```

Decathlon

```
> fviz_pca_var(res.pca, col.var = "cos2", repel = TRUE)
> fviz_pca_var(res.pca, col.var = "contrib")
```

Decathlon

```
> fviz_pca_ind(res.pca, col.ind = "cos2",repel = TRUE)
> fviz_pca_ind(res.pca, pointsize = "cos2", pointshape = 21, repel = TRUE)
> fviz_contrib(res.pca, choice = "ind", axes = 1:2)
> fviz_cos2(res.pca, choice = "ind", axes = 1:2)
```

Decathlon

```
> fviz_pca_biplot(res.pca, repel = TRUE)
```

Decathlon

```
> res.pca = PCA(decathlon2, ind.sup = 24:27, quanti.sup = 11:12, quali.sup = 13, graph=FALSE)
> fviz_pca_ind(res.pca, habillage = 13, addEllipses = TRUE, ellipse.type = "confidence")
```

Decathlon, Euclidean vs. Mahalanobis

Decathlon, Euclidean vs. Mahalanobis

```
> res.pca = PCA(decathlon, graph = FALSE, scale.unit =
    FALSE)
```

Decathlon (ranks)

```
> decathlon3 = decathlon2
> for(i in c(1,5,6,10)) decathlon3[,i]=rank(decathlon3
    [,i],ties="random")
> for(i in c(2,3,4,7,8,9)) decathlon3[,i]=rev(rank(
    decathlon3[,i],ties="random"))
> decathlon3
            100m Long.jump Shot.put High.jump 400m
SEBRLE        15        15        5         6   18
CLAY           5        14       16        20   14
KARPOV        13        16        9         4    4
BERNARD       14         6       14        13    9
> res.pca = PCA(decathlon2.active, graph = FALSE)
```

Decathlon (ranks)

```
> decathlon2 = decathlon[1:23, 1:10]
> for(i in c(1,5,6,10)) decathlon2[,i]=rank(decathlon2
    .active[,i],ties="random")
> for(i in c(2,3,4,7,8,9)) decathlon2[,i]=rev(rank(
    decathlon2.active[,i],ties="random"))
```

Dataset (European Temperature)

```
> temperature = read.table("http://factominer.free.fr/
    bookV2/temperature.csv", header=TRUE,sep=";",dec
    =".",row.names=1)
```

35 cities in Europe (rows)
(average) temperature
per month (column)

Dataset (European Temperature)

See european_temperature.csv

	Jan	Feb	Mar	Apr	May	Jun
Amsterdam	2.9	2.5	5.7	8.2	12.5	14.8
Athens	9.1	9.7	11.7	15.4	20.1	24.5
Berlin	-0.2	0.1	4.4	8.2	13.8	16.0
Brussels	3.3	3.3	6.7	8.9	12.8	15.6
Budapest	-1.1	0.8	5.5	11.6	17.0	20.2
...	...	...	...	...	...	...

	Jul	Aug	Sept	Oct	Nov	Dec
Amsterdam	17.1	17.1	14.5	11.4	7.0	4.4
Athens	27.4	27.2	23.8	19.2	14.6	11.0
Berlin	18.3	18.0	14.4	10.0	4.2	1.2
Brussels	17.8	17.8	15.0	11.1	6.7	4.4
Budapest	22.0	21.3	16.9	11.3	5.1	0.7
...	...	...	...	...	...	...

Dataset (European Temperature)

```
> library("FactoMineR")
> temperature2 = temperature[,1:12]
> res.pca = PCA(temperature2, graph =
    FALSE)
> library("factoextra")
> fviz_eig(res.pca)
> fviz_cos2(res.pca, choice = "var",
   axes = 1:2)
```

Dataset (European Temperature)

```
> fviz_pca_var(res.pca, col.var = "cos2", repel = TRUE)
> fviz_pca_var(res.pca, col.var = "contrib")
```

Dataset (European Temperature)

```
> fviz_pca_ind(res.pca, col.ind = "cos2", repel = TRUE)
> fviz_pca_ind(res.pca, col.ind = "cos2", pointsize = "cos2",repel = TRUE)
> fviz_contrib(res.pca, choice = "ind", axes = 1:2)
> fviz_cos2(res.pca, choice = "ind", axes = 1:2)
```

Dataset (European Temperature)

```
> res<-PCA(
    temperature,
  ind.sup=24:35,
  quanti.sup=13:16,
  quali.sup=17)
> plot.PCA(res,
  choix="ind",
  habillage=17)
```

Dataset (European Temperature)

```
> fviz_pca_biplot(res.pca, repel = TRUE)
```

Dataset (European Temperature)

```
> library(rainbow)
> TPS=fts(x = 1:12, y = t(temperature
    [,1:12]), xname = "Month", yname =
     "Temperature")
> fboxplot(data = TPS, plot.type = "
    functional", type = "bag",
    legendpos = "topright",ylim=c
    (-10,34),axis=FALSE)
> fboxplot(data = TPS, plot.type = "
    bivariate", type = "bag",legendpos
     = "topright",axis=FALSE)
```

PCA & pictures

```
> library(keras)
> mnist = dataset_mnist()
> idx3 = which(mnist$train$y %in% c(1,0))
> V = mnist$train$x[idx3[1:2500],,]
> Y = mnist$train$y[idx3][1:2500]
> MV = NULL
> for(i in 1:2500) MV= cbind(MV,as.vector(
    V[i,,]))
> MV = t(MV)
```

PCA & pictures: $\{0,1\}$

```
> idx3 = which(mnist$train$y %in% c(1,0))
```

```
> v = apply(MV,2,var)
> MV = MV[,which(v>0)]
> res.pca = PCA(MV)
> V1 = rep(0,length(v))
> V1[v>0]=res.pca$var$coord[,1]
> MV1=matrix(V1,28,28)
> image(1:28,1:28,t(MV1)[,28:1])
```

PCA & pictures: $\{0, 8\}$

```
> idx3 = which(mnist$train$y %in% c(8,0))
```

```
> v = apply(MV,2,var)
> MV = MV[,which(v>0)]
> res.pca = PCA(MV)
> V1 = rep(0,length(v))
> V1[v>0]=res.pca$var$coord[,1]
> MV1=matrix(V1,28,28)
> image(1:28,1:28,t(MV1)[,28:1])
```

PCA & pictures: $\{2,5\}$

```
> idx3 = which(mnist$train$y %in% c(2,5))
```

```
> v = apply(MV,2,var)
> MV = MV[,which(v>0)]
> res.pca = PCA(MV)
> V1 = rep(0,length(v))
> V1[v>0]=res.pca$var$coord[,1]
> MV1=matrix(V1,28,28)
> image(1:28,1:28,t(MV1)[,28:1])
```

PCA & projections

PCA vs. regression (OLS)

PCA & regression

Why not consider a PCA on the dataset $[Y, \boldsymbol{X}]$?

With PCA, the first component $\boldsymbol{z}_1 = \boldsymbol{\omega}_1^\top \boldsymbol{x}$ is

$$\boldsymbol{\omega}_1 = \underset{\|\boldsymbol{\omega}\|=1}{\text{argmax}} \left\{ \text{Var}[\boldsymbol{\omega}^\top \boldsymbol{X}] \right\} = \underset{\|\boldsymbol{\omega}\|=1}{\text{argmax}} \left\{ < \boldsymbol{\omega}^\top \boldsymbol{x}, \boldsymbol{\omega}^\top \boldsymbol{x} > \right\}$$

With PLS, the first component $\boldsymbol{z}_1 = \boldsymbol{\omega}_1^\top \boldsymbol{x}$ is

$$\boldsymbol{\omega}_1 = \underset{\|\boldsymbol{\omega}\|=1}{\text{argmax}} \left\{ \text{Cov}[Y, \boldsymbol{\omega}^\top \boldsymbol{X}] \right\} = \underset{\|\boldsymbol{\omega}\|=1}{\text{argmax}} \left\{ < \boldsymbol{y}, \boldsymbol{\omega}^\top \boldsymbol{x} > \right\}$$

PCA & neural nets: Autoencoders

$$\boxed{\boldsymbol{X} \in \mathcal{X}} \xrightarrow{\varphi} \boxed{\boldsymbol{Z} \in \mathcal{Z}} \xrightarrow{\psi} \boxed{\boldsymbol{X}' \in \mathcal{X}}$$

The error function of a nonlinear autoencoder is

$$\|\boldsymbol{X}-\boldsymbol{X}'\|^2 = \|\boldsymbol{X}-\psi\circ\varphi(\boldsymbol{X})\|^2 = \sum_{i=1}^{n} \big(\psi\circ\varphi(\boldsymbol{x}_i)-\boldsymbol{x}_i\big)^\top \big(\psi\circ\varphi(\boldsymbol{x}_i)-\boldsymbol{x}_i\big)$$

$$\boxed{\boldsymbol{X} \in \mathcal{X}} \xrightarrow{\boldsymbol{P}} \boxed{\boldsymbol{Z} \in \mathcal{Z}} \xrightarrow{\boldsymbol{P}^\top} \boxed{\boldsymbol{X}' \in \mathcal{X}}$$

The error function of a linear autoencoder is

$$\|\boldsymbol{X}-\boldsymbol{X}'\|^2 = \|\boldsymbol{X}-\boldsymbol{P}^\top\boldsymbol{P}\boldsymbol{X}\|^2 = \sum_{i=1}^{n} \big(\boldsymbol{P}^\top\boldsymbol{P}\boldsymbol{x}_i-\boldsymbol{x}_i\big)^\top \big(\boldsymbol{P}^\top\boldsymbol{P}\boldsymbol{x}_i-\boldsymbol{x}_i\big)$$

Autoencoders

The error function of a linear autoencoder is

$$\sum_{i=1}^{n} \left(\boldsymbol{P}^\top \boldsymbol{P}\boldsymbol{x}_i - \boldsymbol{x}_i\right)^\top \left(\boldsymbol{P}^\top \boldsymbol{P}\boldsymbol{x}_i - \boldsymbol{x}_i\right)$$

i.e.

$$\sum_{i=1}^{n} \text{trace}\left[\left(\boldsymbol{P}^\top \boldsymbol{P} - \mathbb{I}\right)\boldsymbol{x}_i\boldsymbol{x}_i^\top \left(\boldsymbol{P}^\top \boldsymbol{P} - \mathbb{I}\right)\right]$$

$$\text{trace}\left[\left(\boldsymbol{P}^\top \boldsymbol{P} - \mathbb{I}\right)\sum_{i=1}^{n} \boldsymbol{x}_i\boldsymbol{x}_i^\top \left(\boldsymbol{P}^\top \boldsymbol{P} - \mathbb{I}\right)\right]$$

the middle term is a covariance matrix, thus it is $\boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^\top$, so we recognize

$$\|\left(\boldsymbol{P}^\top \boldsymbol{P} - \mathbb{I}\right)\boldsymbol{V}\boldsymbol{\Delta}^{1/2}\|_F^2$$

where $\|\cdot\|_F$ denotes the Froebenius / entrywise ℓ_2 norm of a matrix,

$$\|\boldsymbol{M}\|_F^2 = \sum_{i,j} M_{i,j}^2 = \text{trace}\left(\boldsymbol{M}\boldsymbol{M}^\top\right) \text{ where } \boldsymbol{M} = \left[M_{i,j}\right]$$

Alternative Approach

Let $\|\cdot\|_F$ denote the Froebenius / entrywise ℓ_2 norm of a matrix,

$$\|\boldsymbol{M}\|_F^2 = \sum_{i,j} M_{i,j}^2 \text{ where } \boldsymbol{M} = \left[M_{i,j}\right]$$

Consider the following problem,

$$\min_{\boldsymbol{Y}} \left\{\|\boldsymbol{X} - \boldsymbol{Y}\|_F^2\right\} \text{ subject to } \text{rank}(\boldsymbol{Y}) = k \quad (\leq \text{rank}(\boldsymbol{X}))$$

If $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{V}^\top$ then $\boldsymbol{Y} = \boldsymbol{U}_k\boldsymbol{\Delta}_k\boldsymbol{V}_k^\top$ where we keep the first k columns of $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{\Delta}$.
One can rewrite

$$\min_{\boldsymbol{P}\in\Pi} \left\{\|\boldsymbol{X} - \boldsymbol{P}\boldsymbol{X}\|_F^2\right\} \text{ subject to } \text{rank}(\boldsymbol{P}) = k$$

where Π is the set of projection matrices.
If $\boldsymbol{S} = \boldsymbol{X}^\top\boldsymbol{X}$, we can write (equivalently)

$$\max_{\boldsymbol{P}\in\Pi} \left\{\text{trace}(\boldsymbol{S}\boldsymbol{P})\right\} \text{ subject to } \text{rank}(\boldsymbol{P}) = k$$

Alternative Approach

or

$$\max_{\boldsymbol{P}\in\mathcal{P}} \left\{\text{trace}(\boldsymbol{S}\boldsymbol{P})\right\}$$

where

$$\mathcal{P} = \left\{\boldsymbol{P} \in \Pi : \text{eigenvalues}(\boldsymbol{P}) \in \{0,1\} \text{ and } \text{trace}(\boldsymbol{P}) = k\right\}$$

Then $\boldsymbol{P}^\star = \boldsymbol{V}_k \boldsymbol{V}_k^\top$
Note that $\mathcal{P}$ is note convex. But one can consider

$$\mathcal{P}_\star = \left\{\boldsymbol{P} \in \Pi : \text{eigenvalues}(\boldsymbol{P}) \in [0,1] \text{ and } \text{trace}(\boldsymbol{P}) = k\right\}$$

which is the convex Hull of $\mathcal{P}$. And one can prove that the convex problem

$$\max_{\boldsymbol{P}\in\mathcal{P}_\star} \left\{\text{trace}(\boldsymbol{S}\boldsymbol{P})\right\}$$

Same solution as the non-convex one (proof in On a Theorem of Weyl Concerning Eigenvalues of Linear Transformations).