EE-311—Apprentissage et intelligence artificielle

5. Réduction de dimension

Michael Liebling

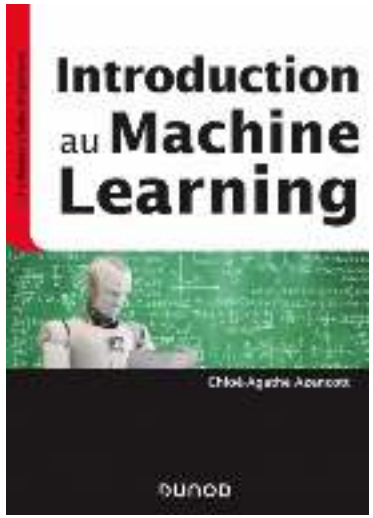
https://moodle.epfl.ch/course/view.php?id=16090

21 mars 2025 (compilé le 20 mars 2025)

idiap RESEARCH INSTITUTE

EPFL

Ouvrage de référence et source



Ces transparents sont basés en grande partie sur le texte de Chloé-Agathe Azencott "Introduction au Machine Learning", Dunod, 2019
ISBN 978-210-080153-4

L'auteure a mis le texte (sans les exercices) à disposition ici : http://cazencott.info/dotclear/public/lectures/IntroML_Azencott.pdf

Avertissement : Bien que ces transparents partagent la notation mathématique, la structure de l'exposition (en partie), et certains exemples avec le livre, ils ne constituent qu'un complément et non un remplacement ou une source unique pour la couverture des matières du cours. À ce titre, ces transparents ne se substituent pas au texte.

Contenu

- Réduction de dimension
  - Motivation
  - 1. sélection de variables
    - a. méthodes de filtrage
    - Intermède : erreur de généralisation
      - découpage des jeux de données (entraînement, validation, test)
      - validation croisée
    - b. méthodes de conteneur :
      - naïve, recherche ascendante, descendante et flottante
  - 2. extraction de variables (PCA)

Rappel : organisation des données d'un problème d'apprentissage (matrice de données X)

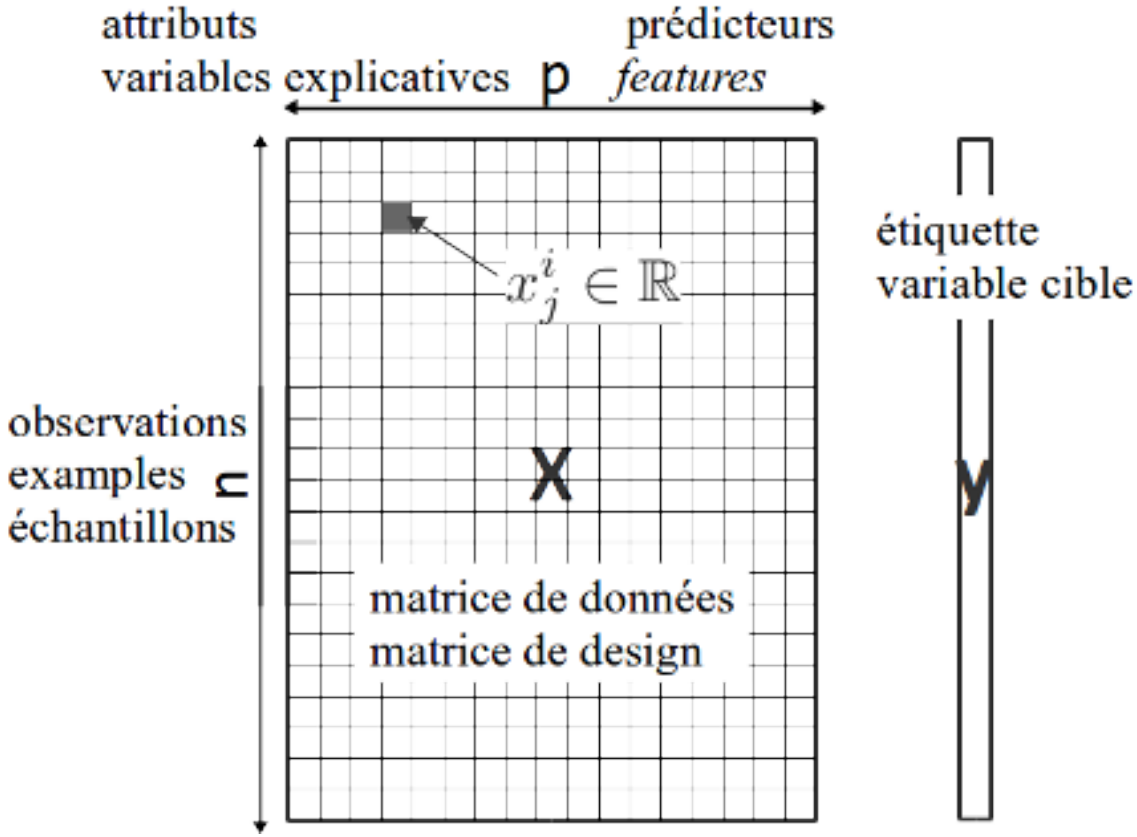


FIGURE 2.1 – Les données d'un problème d'apprentissage supervisé sont organisées en une matrice de design et un vecteur d'étiquettes. Les observations sont représentées par leurs variables explicatives.

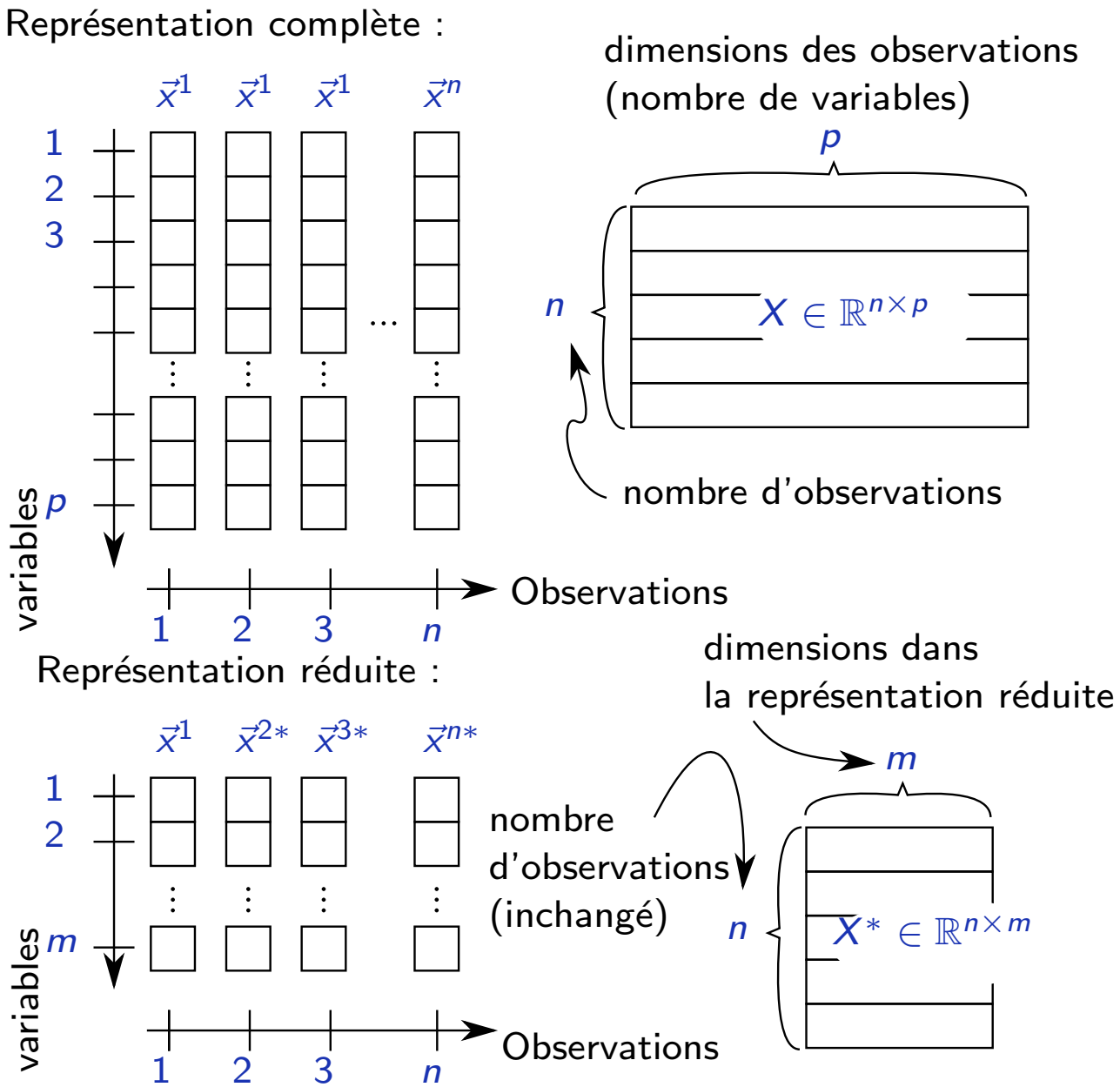
Azencott

But de la réduction de dimension

Le but de la réduction de dimension est de transformer une représentation $X \in \mathbb{R}^{n \times p}$ des données en une représentation $X^* \in \mathbb{R}^{n \times m}$ où $m \ll p$.

n : nombre d'observations
p : nombre de dimensions pour chaque observation (nombre de features)
m : nombre de dimensions dans la représentation réduite

Réduction de dimension



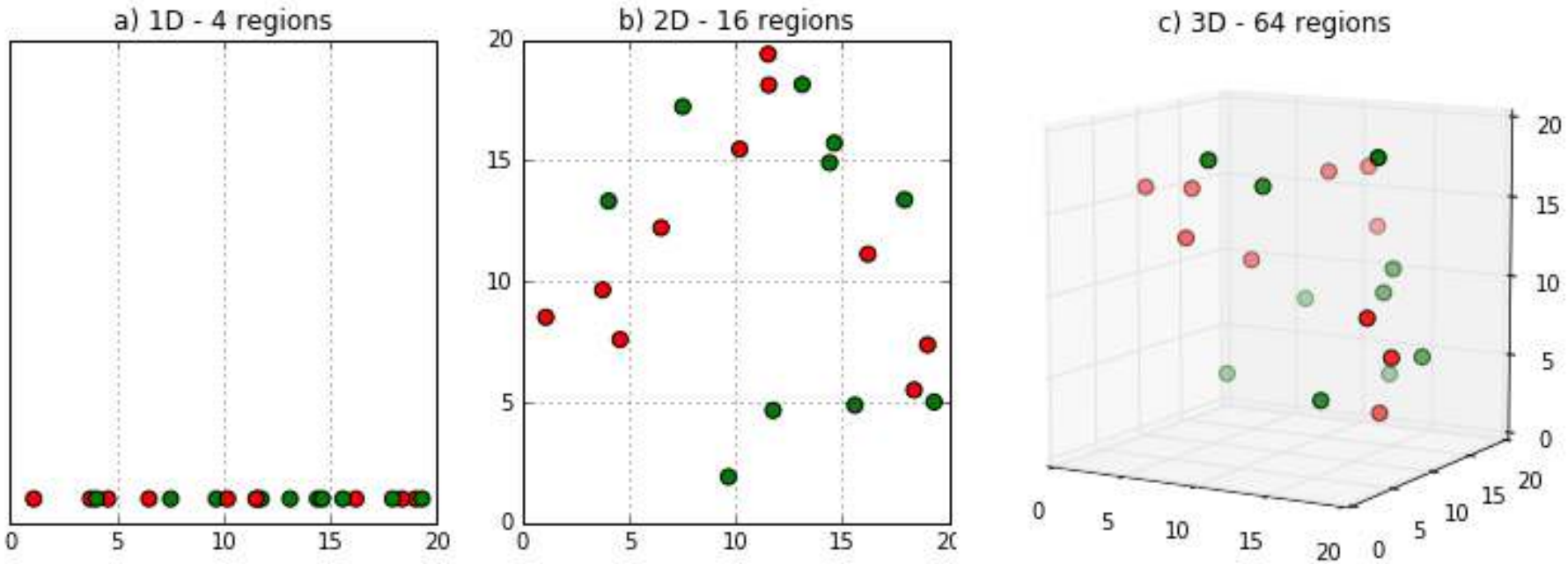
Motivations pour réduire les données

Les raisons de la démarche de réduction sont multiples :

- visualiser les données (plus aisé de représenter des données dans le plan que dans un espace p-dimensionel)
- réduire les coûts algorithmiques (moins de variables donc moins de mémoire)
- améliorer la qualité des modèles
  - moins de variables, moins de risque de sur-apprendre
  - exclure les variables qui ne sont pas pertinentes au problème et pourraient induire l'apprentissage en erreur
  - éviter le phénomène du "fléau de la dimension" (curse of dimensionality) : *les intuitions développées en faible dimension ne s'appliquent pas nécessairement en haute dimension*

Fléau de la dimension (curse of dimensionality)

Plus les exemples comportent un grand nombre de features (= plus la dimension du vecteur qui les charactérise est grande), plus le nombre d'exemples nécessaires pour couvrir l'espace des exemples possibles devient grand : la relation est exponentielle !

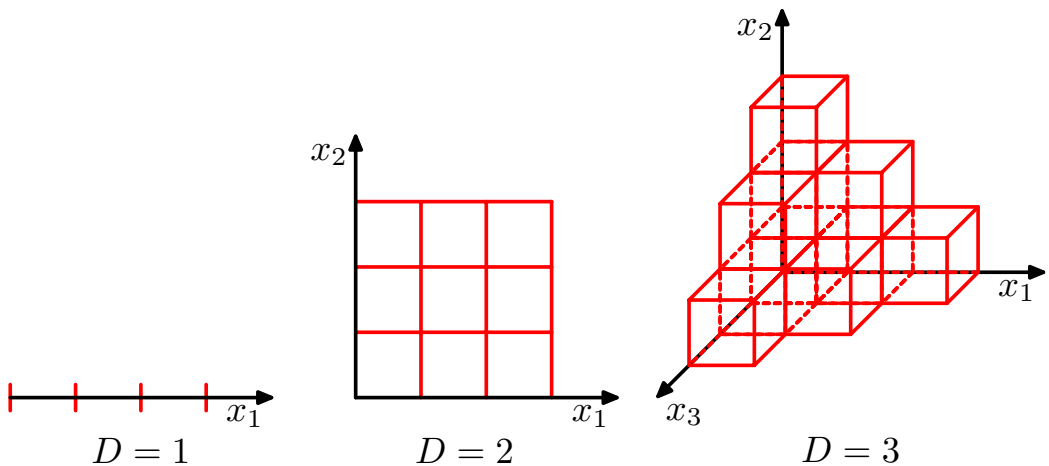


https ://www.kdnuggets.com/2017/04/must-know-curse-dimensionality.html

Curse of dimensionality

Le nombre de régions définies par une grille régulière grandit exponentiellement avec la dimension

Figure 1.21 Illustration of the curse of dimensionality, showing how the number of regions of a regular grid grows exponentially with the dimensionality D of the space. For clarity, only a subset of the cubical regions are shown for $D = 3$.



Bishop, Fig. 1-21

Curse of dimensionality : le voisinage se raréfie

En haute dimension, les exemples d'apprentissage ont tendance à tous être éloignés les uns des autres.

Illustration :

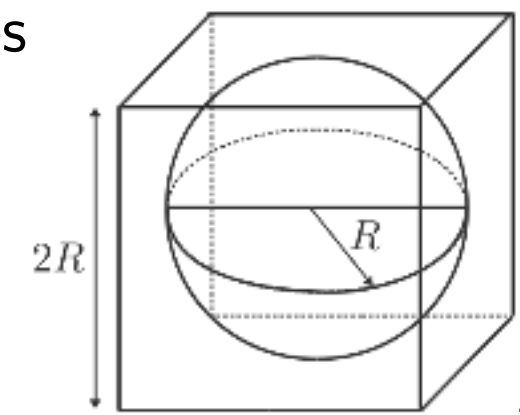
• Hypersphère de rayon $R \in \mathbb{R}_+^*$ centrée sur une observation $\vec{x}$ en dimension p, dénotée $\mathcal{S}(\vec{x}, R)$ (voisinage avec distance $\leq R$) :

$$\text{Volume de } \mathcal{S}(\vec{x}) = \frac{2R^p \pi^{p/2}}{p\Gamma(p/2)}$$

(Rappel : la fonction Gamma de Leonhard Euler $\Gamma(n) = (n-1)!$ quand n est un entier positif)

• Hypercube de côté $2R$, circonscrit à la sphère, dénoté $\mathscr{C}(\vec{x}, R)$ (solide qui comprend les exemples possibles avec chaque coordonnée dans $[-R, R]$)

$$\text{Volume de } \mathscr{C}(\vec{x}, R) = 2^p R^p$$



Azencott

Fléau de la dimension (curse of dimensionality), suite

Rapport du volume de l'hypercube à celui de l'hypersphère :

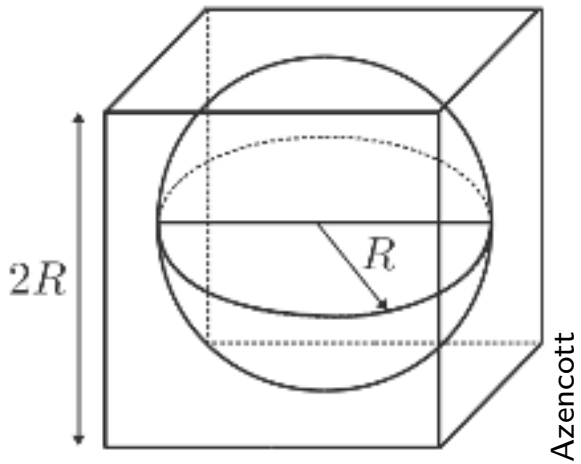
$$\text{ratio}_{p=\infty} = \lim_{p\to\infty} \frac{\text{Vol}(\mathcal{S}(\vec{x}, R))}{\text{Vol}(\mathscr{C}(\vec{x}, R))} = 0$$

$p = 1$ $\quad \text{ratio}_{p=1} = 1$

$p = 2$ $\quad \text{ratio}_{p=2} = \frac{\pi}{4} \approx 0.79$

$p = 3$ $\quad \text{ratio}_{p=3} = \frac{\pi}{6} \approx 0.52$

$p = \infty$ $\quad \text{ratio}_{p=\infty} \to 0$



Lorsque la dimension est grande, le voisinage d'un exemple couvre donc un volume qui s'amenuise en comparaison à celui de l'espace des examples possibles avec coordonnées d'amplitude comparable !

Fléau de la dimension : implications

Implications d'une dimensionalité élevée :

- Le voisinage d'un point (hypersphère) représente une proportion de l'espace de plus en plus petite par rapport au cube (lorsque la dimension augmente)
- Les données sont de plus en plus isolées en dimension élevées
- il faut de plus en plus de données d'entraînement (*rule of thumb*, "à la louche," à appliquer avec précaution et sans garantie. . . : 5 fois le nombre de dimensions, i.e. $n \approx 5p$)

Réduction de la dimensionalité : approches possibles

Deux possibilités s'offrent à nous pour réduire la dimension de nos données :

1. **sélection de variables**, qui consiste à éliminer un nombre $p - m$ de variables de nos données.
   Approches possibles :
   - les méthodes de filtrage
   - les méthodes de conteneur
   - les méthodes embarquées (pas couvertes dans le cours d'aujourd'hui)
2. **extraction de variables**, qui consiste à créer m nouvelles variables à partir des p variables dont nous disposons initialement.

Méthodes de sélection de variables : filtrage

Nous considérons des méthodes supervisées et supposons disposer d'un jeu de données étiqueté $\mathscr{D} = \left\{ \left(\vec{x}^i, y^i \right) \right\}_{i=1,\ldots,n}$ où $\vec{x}^i \in \mathbb{R}^p$.

Sélection de variable par filtrage consiste à appliquer un critère de sélection indépendamment à chacune des p variables

Idée quantifier la pertinence de la p-ème variable du jeu de donnée par rapport à y sur la base de

- la corrélation avec l'étiquette
- un test statistique dans le cas d'un problème de classification
- l'information mutuelle

Si la pertinence est basse on se débarrasse de la variable.

Méthode de sélection de variable par filtrage : corrélation

Pour chaque variable x_j, on calcule la corrélation (de Pearson) entre la $j$ème variable et l'étiquette y

$$R_j = \frac{\sum_{i=1}^{n} \left(y^i - \frac{1}{n} \sum_{\ell=1}^{n} y^\ell \right) \left(x_j^i - \frac{1}{n} \sum_{\ell=1}^{n} x_j^\ell \right)}{\sqrt{\sum_{i=1}^{n} \left(y^i - \frac{1}{n} \sum_{\ell=1}^{n} y^\ell \right)^2} \sqrt{\sum_{i=1}^{n} \left(x_j^i - \frac{1}{n} \sum_{\ell=1}^{n} x_j^\ell \right)^2}}$$

Notes :

- prend des valeurs entre -1 (forte anti-correlation) à 1 (forte corrélation)
- Si $|R_j|$ est proche de zéro (pas de corrélation entre la $j$ème variable x_j et l'étiquette y) on se débarrasse de cette variable.
- elle se calcule comme la corrélation entre une étiquette prédite et une étiquette réelle (voir page suivante)
- aussi appelé *Pearson correlation coefficient*

Exemple coefficient de correlation (Pearson)

Keith Weller, wikipedia

Ivar Leidus, wikipedia

On mesure la table suivante X et étiquettes $\vec{y}$ pour quatre fruits :

	$x_1^i = \frac{\text{hauteur}}{\text{largeur}}$	$x_2^i = \text{masse}$	$y^i = \text{poire ?}$
$i = 1$	1.4	210	1
$i = 2$	0.9	180	-1
$i = 3$	1.6	190	1
$i = 4$	1.1	220	-1

On calcule $R_1 = 0.928$ et $R_2 = 0$ qui semble indiquer que la variable x_1 est corrélée à l'étiquette y alors que la variable x_2 ne l'est pas.

À propos : définissons le Coefficient de détermination

Définition 3.16 (Coefficient de détermination) Étant données n étiquettes réelles $y^1, y^2, \ldots, y^n$ et n prédictions $f\left(\vec{x}^1\right), f\left(\vec{x}^2\right), \ldots, f\left(\vec{x}^n\right)$, on appelle erreur carrée relative, ou RSE de l'anglais relative squared error la valeur

$$\text{RSE} = \frac{\sum_{i=1}^{n}\left(f\left(\vec{x}^i\right) - y^i\right)^2}{\sum_{i=1}^{n}\left(y^i - \frac{1}{n}\sum_{\ell=1}^{n} y^\ell\right)^2}$$

Coefficient de détermination$R^2 = 1 -$ RSE est le carré du coefficient de corrélation entre $\vec{y}$ et $f\left(\vec{x}^1\right), f\left(\vec{x}^2\right), \ldots, f\left(\vec{x}^n\right)$:

$$R = \frac{\sum_{i=1}^{n}\left(y^i - \frac{1}{n}\sum_{\ell=1}^{n} y^\ell\right)\left(f\left(\vec{x}^i\right) - \frac{1}{n}\sum_{\ell=1}^{n} f\left(\vec{x}^\ell\right)\right)}{\sqrt{\sum_{i=1}^{n}\left(y^i - \frac{1}{n}\sum_{\ell=1}^{n} y^\ell\right)^2}\sqrt{\sum_{i=1}^{n}\left(f\left(\vec{x}^i\right) - \frac{1}{n}\sum_{\ell=1}^{n} f\left(\vec{x}^\ell\right)\right)^2}}$$

R indique à quel point les valeurs prédites sont corrélées aux valeurs réelles (attention, également élevé si elles sont anti-corrélées)

Méthode de sélection de variable par filtrage : Information mutuelle

L'information mutuelle entre deux variables aléatoires X_j et Y mesure leur dépendance au sens probabiliste ; elle est nulle si et seulement si les variables sont indépendantes, et croît avec leur degré de dépendance. Elle est définie, dans le cas discret, par

$$I(X_j, Y) = \sum_{x_i, y} \mathbb{P}(X_j = x_j, Y = y) \log \frac{\mathbb{P}(X_j = x_j, Y = y)}{\mathbb{P}(X_j = x_j)\mathbb{P}(Y = y)}$$

et dans le cas continu par

$$I(X_j, Y) = \int_{x_j} \int_{y} p(x_j, y) \log \frac{p(x_j, y)}{p(x_j)p(y)} \mathrm{d}x_j \mathrm{d}y$$

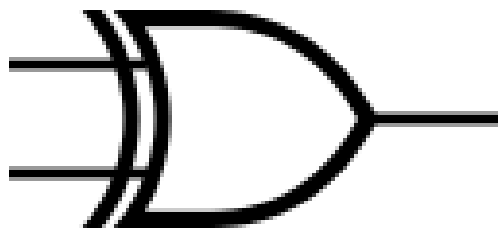
Limitation des méthodes de filtrage

Les méthodes de filtrage souffrent de traiter les variables individuellement : elles ne peuvent pas prendre en compte leurs effets combinés.

Exemple illustrant ce problème : Expliquer la sortie d'une porte logique "ou exclusif" (XOR) par rapport aux entrées :

XOR Table de vérité

Input x_1	Input x_2	Output y
0	0	0
0	1	1
1	0	1
1	1	0



Prise individuellement, x_1 (resp. x_2) est décorrélée de $y = x_1 \text{XOR} x_2$, alors qu'ensemble, ces deux variables expliqueraient parfaitement l'étiquette y.

Estimation empirique de l'erreur de généralisation (Préambule aux autres méthodes de sélection de variables)

L'erreur empirique mesurée sur les observations qui ont permis de construire le modèle est un mauvais estimateur de *l'erreur du modèle sur l'ensemble des données possibles* (appelée *erreur de généralisation*) : si le modèle sur-apprend, cette erreur empirique peut être proche de zéro voire nulle, tandis que l'erreur de généralisation peut être arbitrairement grande.

Pour évaluer la qualité d'un modèle appris, on sépare communément les données en trois jeux de données (pourcentages indicatifs, règle générale) :

1. jeu d'entraînement (60-70% des données)
2. jeu de validation (15-20% des données), e.g. si plusieurs modèles sont considérés ou si le modèle à entraîner a des meta-paramètres
3. jeu de test (15-20% des données)

Jeu d'entraînement, Jeu de test

Pour évaluer un modèle, il est indispensable d'utiliser des données étiquetées qui n'ont **pas** *servi à le construire.*

Définition 3.1 (Jeu d'entraînement, Jeu de test) Étant donné un jeu de données $\mathscr{D} = \left\{\left(\vec{x}^i, y^i\right)\right\}_{i=1,\ldots,n}$ partitionné en deux jeux $\mathscr{D}_{\text{tr}}$ et $\mathscr{D}_{\text{te}}$, on appelle jeu d'entraînement (training set en anglais) l'ensemble $\mathscr{D}_{\text{tr}}$ utilisé pour entraîner un modèle prédictif, et jeu de test (test set en anglais) l'ensemble $\mathscr{D}_{\text{te}}$ utilisé pour son évaluation. **La perte calculée sur ce jeu de test est un estimateur de l'erreur de généralisation.**

Attention : manquer à séparer les jeux d'entraînement et de test (e.g. en présentant comme la performance d'un modèle son erreur sur le jeu d'entraînement) est probablement le pêché capital du machine learning !

Jeu de validation

Considérons la situation où nous devons choisir entre K modèles : nous pouvons entraîner chacun des modèles sur le jeu de données d'entraînement, obtenant ainsi K fonctions de décision $f_1, f_2, \ldots, f_K$.

Comment choisir le meilleur modèle ? Si on calcule l'erreur de chacun de ces modèles sur le jeu de test pour choisir le meilleur, on ne pourra plus utiliser le jeu de test pour évaluer l'erreur de généralisation du modèle choisi.

Plutôt, on définit un jeu de validation $\mathscr{D}_{\text{val}}$, sur lequel on peut choisir le modèle qui a la plus petite erreur :

$$\hat{f} = \underset{k=1,\ldots,K}{\operatorname{argmin}} \frac{1}{|\mathscr{D}_{\text{val}}|} \sum_{\vec{x},y \in \mathscr{D}_{\text{val}}} L\left(y, f_k(\vec{x})\right)$$

Importance de distinguer la sélection d'un modèle de son évaluation : les faire sur les mêmes données peut nous conduire à sous-estimer l'erreur de généralisation et au sur-apprentissage du modèle choisi.

Solutions de découpage des jeux de données

Entraînement d'un seul modèle sans paramètre

1. jeu d'entraînement $\mathscr{D}_{\text{tr}}$ sur lequel on entraîne l'algorithme d'apprentissage
2. jeu de test $\mathscr{D}_{\text{te}}$sur lequel on évalue l'erreur de généralisation du modèle

Entraînement d'un modèle avec paramètres ou lorsque le modèle doit être choisi parmi plusieurs

1. jeu d'entraînement $\mathscr{D}_{\text{tr}}$ sur lequel on entraîne K algorithmes d'apprentissage
2. jeu de validation $\mathscr{D}_{\text{val}}$ sur lequel on évalue les K modèles pour sélectionner le modèle définitif
3. jeu de test $\mathscr{D}_{\text{te}}$ sur lequel on évalue l'erreur de généralisation du modèle choisi.

Éviter les risque d'un découpage arbitraire : Validation croisée

Définition 3.2 (Validation croisée) Étant donné un jeu $\mathscr{D}$ de n observations, et un nombre K, on appelle validation croisée la procédure qui consiste à

1. partitionner $\mathscr{D}$ en K parties de tailles sensiblement similaires, $\mathscr{D}_1, \mathscr{D}_2, ..., \mathscr{D}_K$
2. pour chaque valeur de $k = 1, ..., K$,
   - entraîner un modèle sur $\bigcup_{\ell \neq k} \mathscr{D}_\ell$
   - évaluer ce modèle sur $\mathscr{D}_k$.

Chaque partition de $\mathscr{D}$ en deux ensembles $\mathscr{D}_k$ et $\bigcup_{\ell \neq k} \mathscr{D}_\ell$ est appelée un fold de la validation croisée.

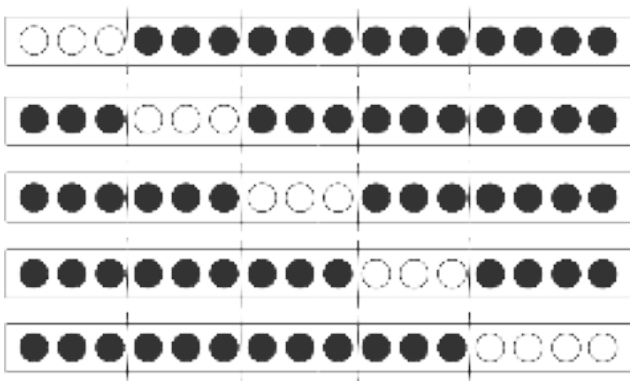


FIGURE 3.1 – Une validation croisée en 5 *folds* : Chaque observation appartient à un des 5 jeux de validation (en blanc) et aux 4 autres jeux d'entraînement (en noir).

Azencott

Évaluation de la performance avec validation croisée (suite)

Méthode 1 : comme chaque obervation étiquetée du jeu $\mathscr{D}$ n'appartient qu'à un unique jeu de test (et à $(K-1)$ jeux d'entraînement) on peut noter l'erreur de prédiction obtenue pour cette observation (c.-à-d., lorsque l'observation à joué le rôle d'observation de test) et répéter l'opération pour toutes les autres observation avant d'en faire la moyenne.

Méthode 2 : évaluer la qualité de chacun de K prédicteurs sur leur jeu de test respectif D_k et soit :

- moyenner les performances.
- calculer leur écart type (qui donne une meilleure indication de la variabilité de la qualité des prédictions en fonction du choix des données d'entraînement)

Évaluation de la performance, validation croisée (suite et fin)

Notes :

- la validation croisée ne permet pas d'améliorer la performance, seulement d'obtenir une meilleure estimation de la performance
- le découpage systématique permet de limiter les effets du choix arbitraire d'un découpage unique

Sélection de variables : méthodes de conteneur

Les méthodes de conteneur, ou wrapper methods en anglais, consistent à essayer de déterminer le meilleur sous-ensemble de variables pour un algorithme d'apprentissage donné.

On parle alors souvent non pas de sélection de variables mais de sélection de sous-ensemble, ou subset selection.

- Méthode naïve (exhaustive)
- Trois méthodes de sélection (non-exhaustives) :
  - recherche ascendante (*forward search*)
  - recherche descendante (*backward search*)
  - recherche flottante

Sélection de variables : méthode de conteneur

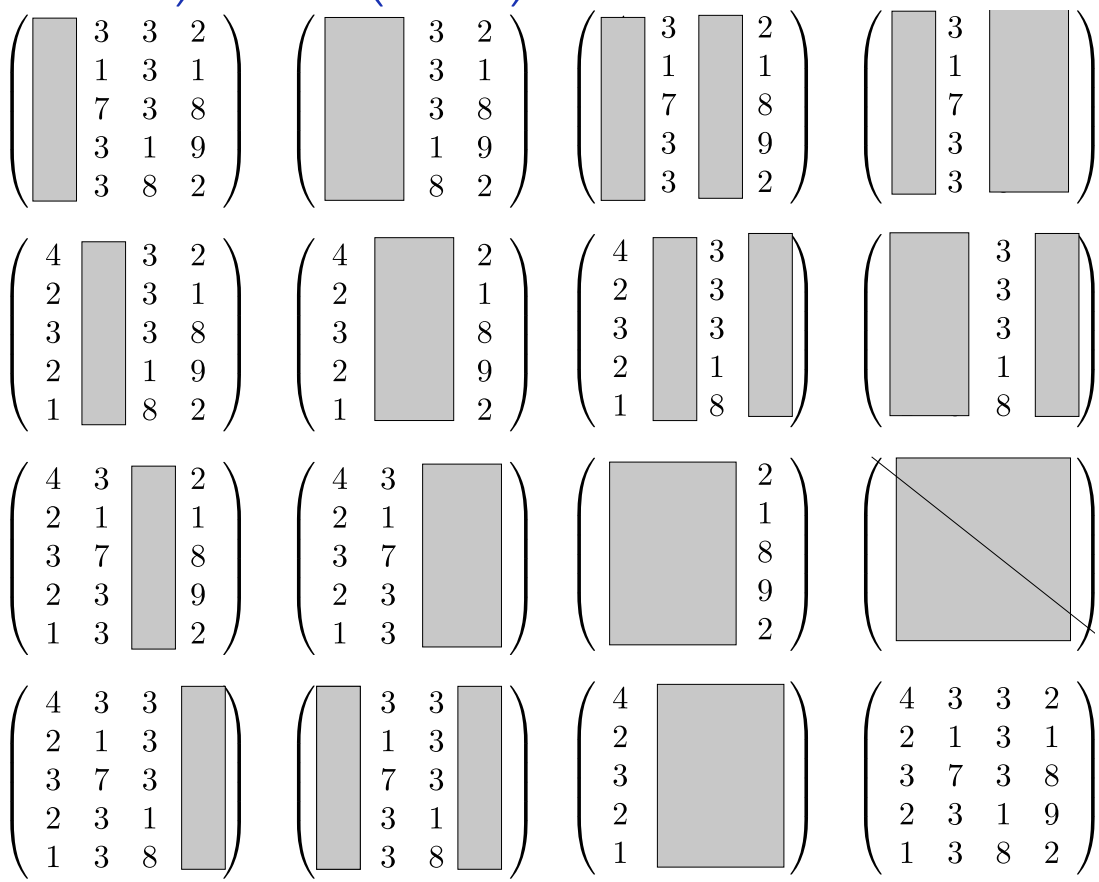
Étant donné un jeu de données $\mathscr{D} = \{(X, \vec{y})\}$ où $X \in \mathbb{R}^{n \times p}$, un sous-ensemble de variables $\mathscr{E} \subset \{1, 2, \ldots, p\}$ et un algorithme d'apprentissage, on notera $X_{\mathscr{E}} \in \mathbb{R}^{n \times |\mathscr{E}|}$ la matrice X restreinte aux variables apparaissant dans $\mathscr{E}$, et $E_{\mathscr{D}}(\mathscr{E})$ l'estimation de l'erreur de généralisation de cet algorithme d'apprentissage, entraîné sur $(X_{\mathscr{E}}, \vec{y})$.

Estimation de l'erreur de généralisation obtenue sur un jeu de test ou par validation croisée.

Example : sélection de sous-ensemble (méthode naïve)

$$X = \begin{pmatrix} 4 & 3 & 3 & 2 \\ 2 & 1 & 3 & 1 \\ 3 & 7 & 3 & 8 \\ 2 & 3 & 1 & 9 \\ 1 & 3 & 8 & 2 \end{pmatrix} \quad \vec{y} = \begin{pmatrix} -1 \\ 1 \\ -1 \\ 1 \\ 1 \end{pmatrix}$$

choix de $2^p - 1$ combinaisons de colonnes à garder (en excluant $\emptyset$) pour former $X_{\mathscr{E}}$:



Méthode naïve : on entraîne des modèles sur chacun des choix (exhaustif !) et on choisit le sous-ensemble qui produit le meilleure modèle

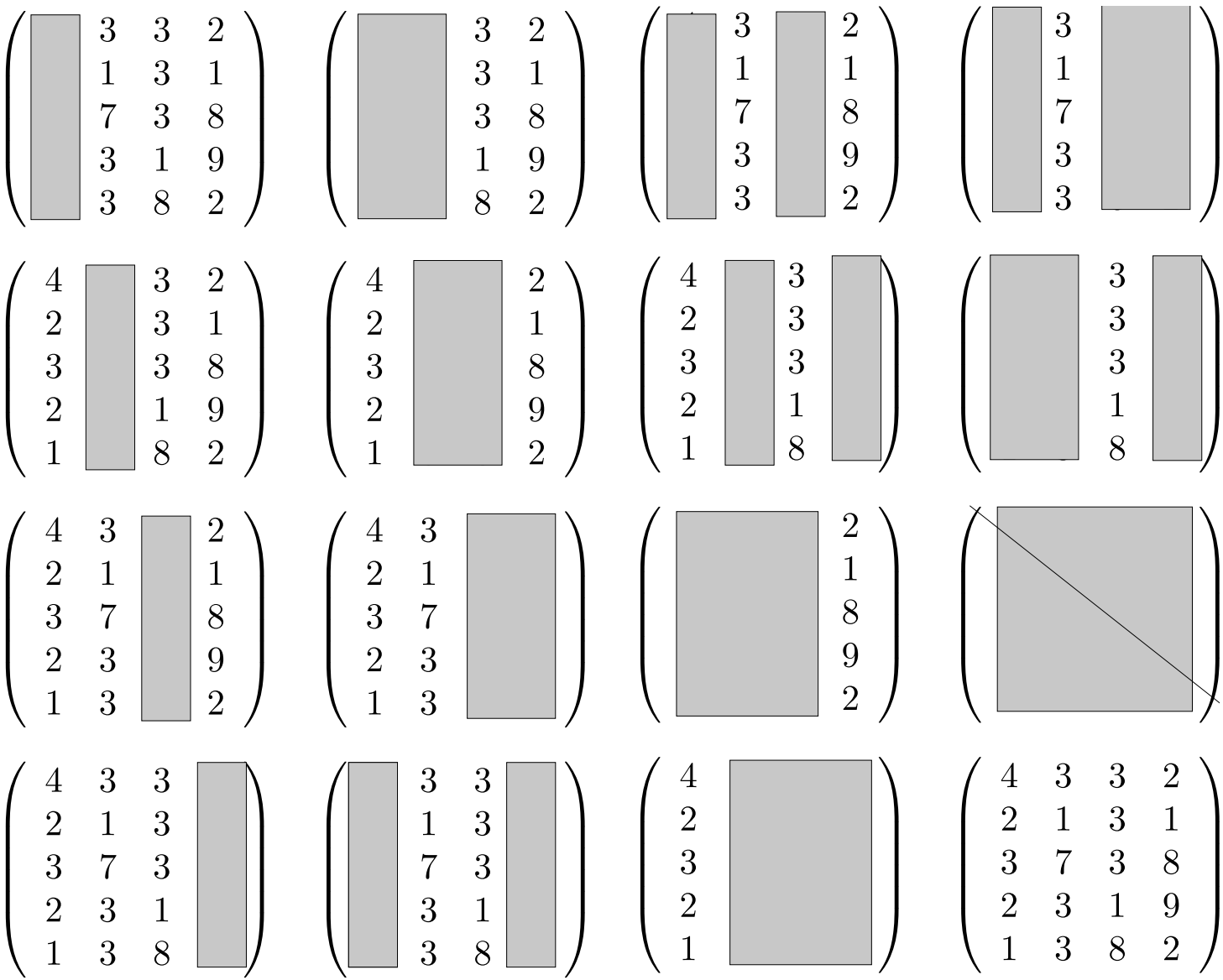
Recherche ascendante

Définition 11.2 (Recherche ascendante) On appelle recherche ascendante, ou forward search en anglais, la procédure gloutonne de sélection de variables suivante :

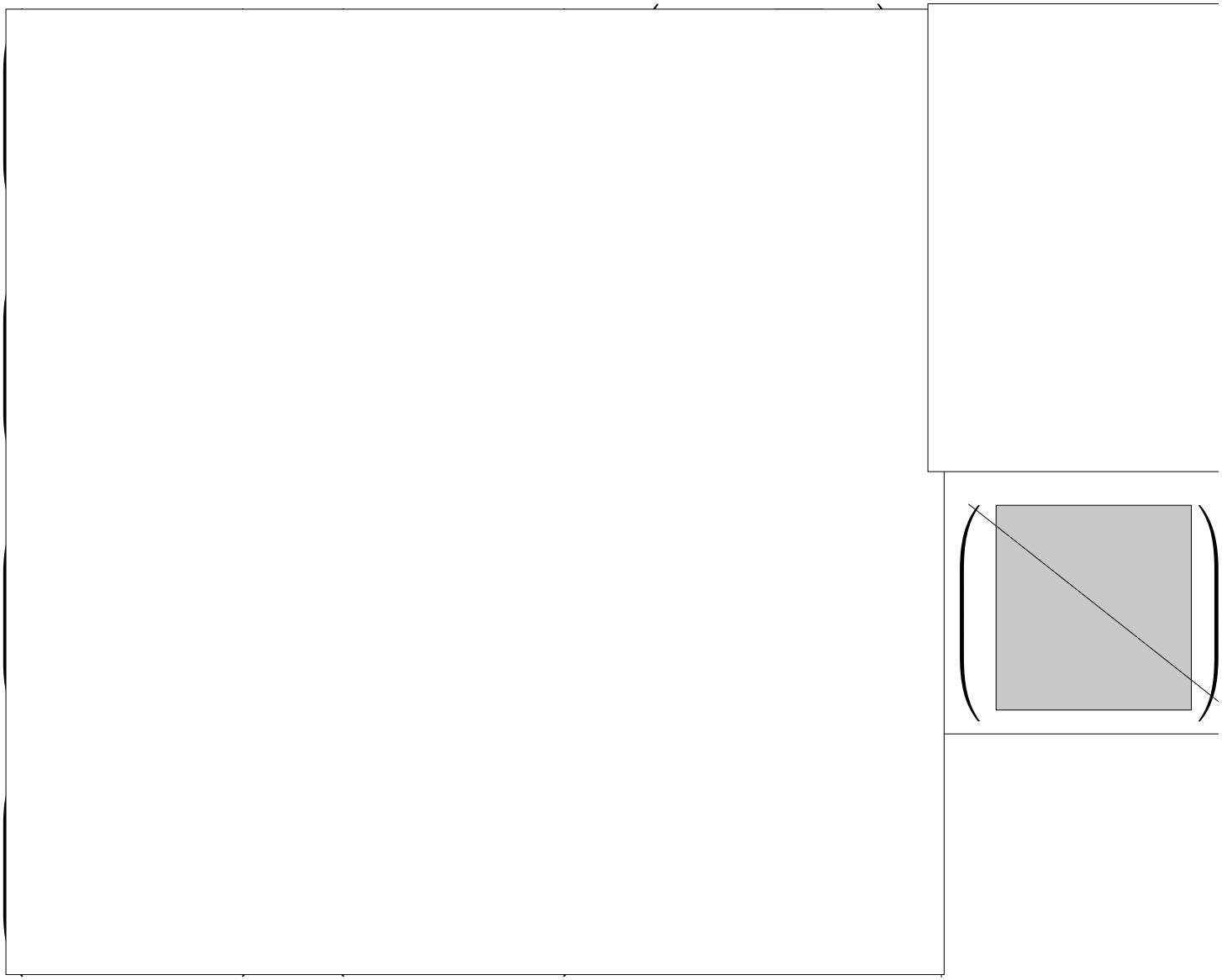
1. Initialiser $\mathscr{F} = \emptyset$
2. Trouver la meilleure variable à ajouter à $\mathscr{F}$:
   $j^* = \underset{j \in \{1,\dots,p\} \setminus \mathscr{F}}{\text{argmin}} \; E_{\mathscr{D}}\left(\mathscr{F} \cup \{j\}\right)$
3. Si $E_{\mathscr{D}}\left(\mathscr{F} \cup \{j\}\right) > E_{\mathscr{D}}\left(\mathscr{F}\right)$: s'arrêter
   Sinon : $\mathscr{F} \leftarrow \mathscr{F} \cup \{j\}$: recommencer 2–3.

Dans le pire des cas (celui où on devra itérer jusqu'à ce que $\mathscr{F} = \{1, 2, \dots, p\}$), cet algorithme requiert de l'ordre de $\mathcal{O}(p^2)$ évaluations de l'algorithme d'apprentissage sur un jeu de données, ce qui peut être intensif, mais est bien plus efficace que $\mathcal{O}(2^p)$ comme requis par l'approche exhaustive.

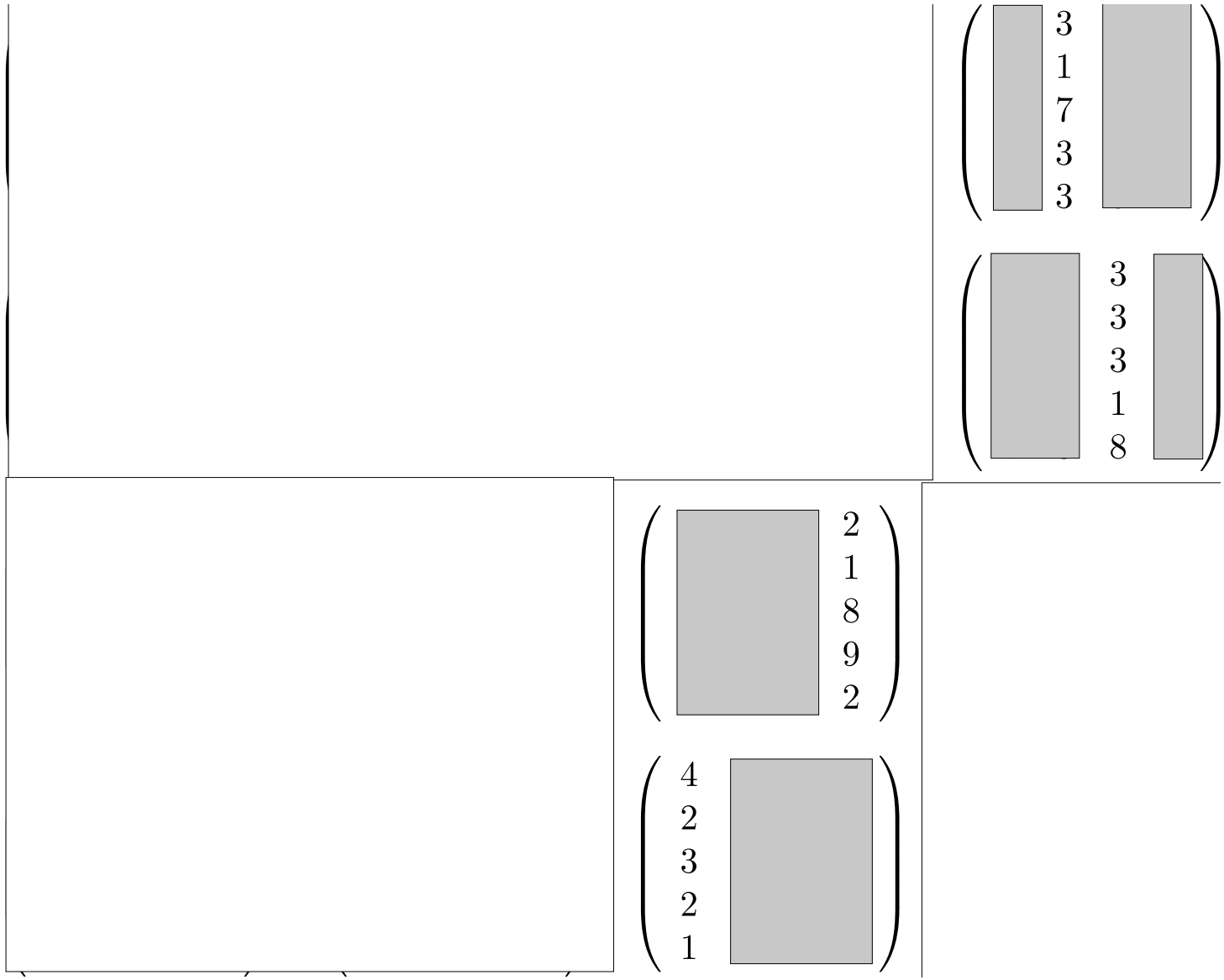
Recherche ascendante : exemple



Initialisation $\mathscr{F} = \emptyset$

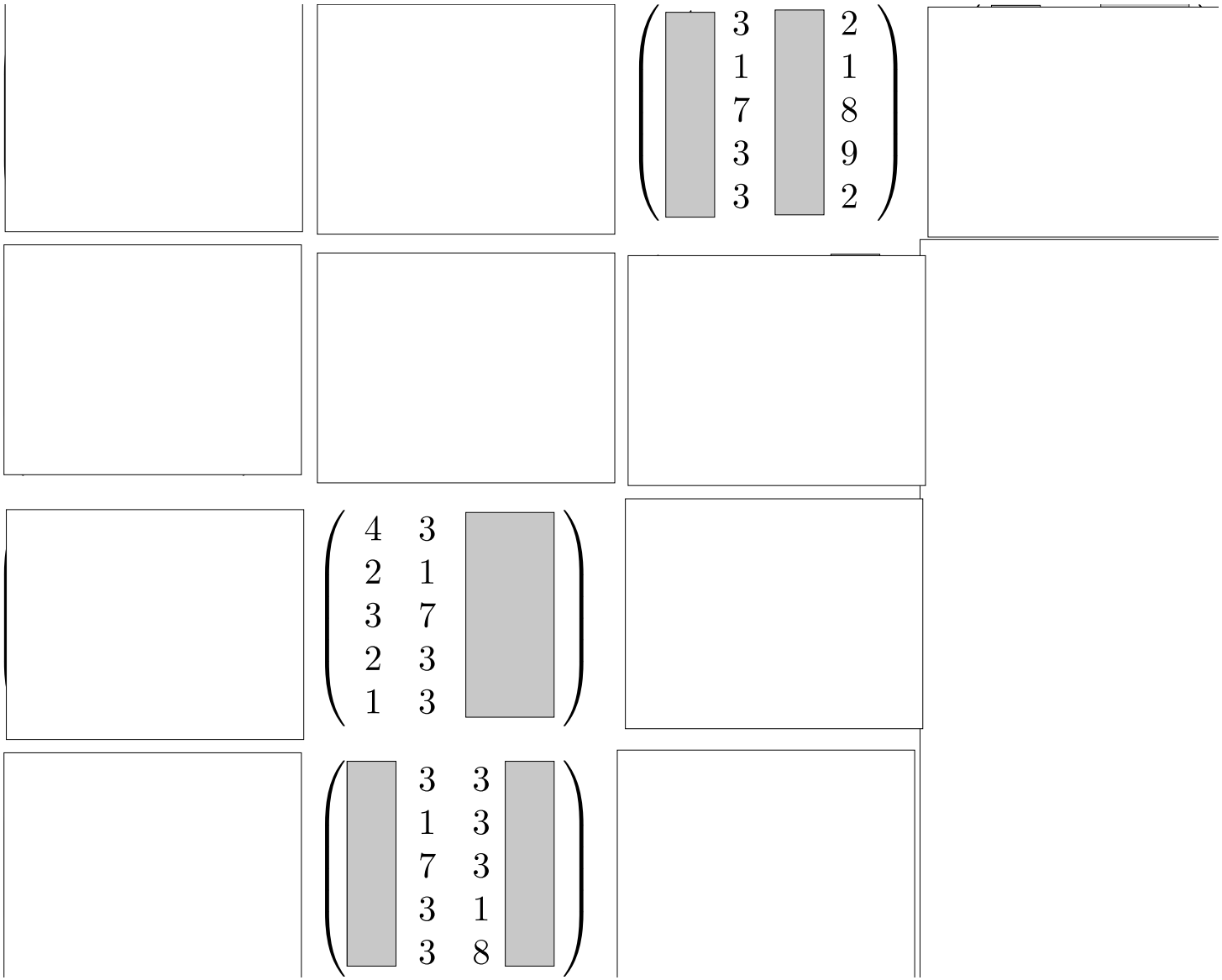


Trouver la meilleure variable à ajouter (4 possibilités = 4 entraînements)



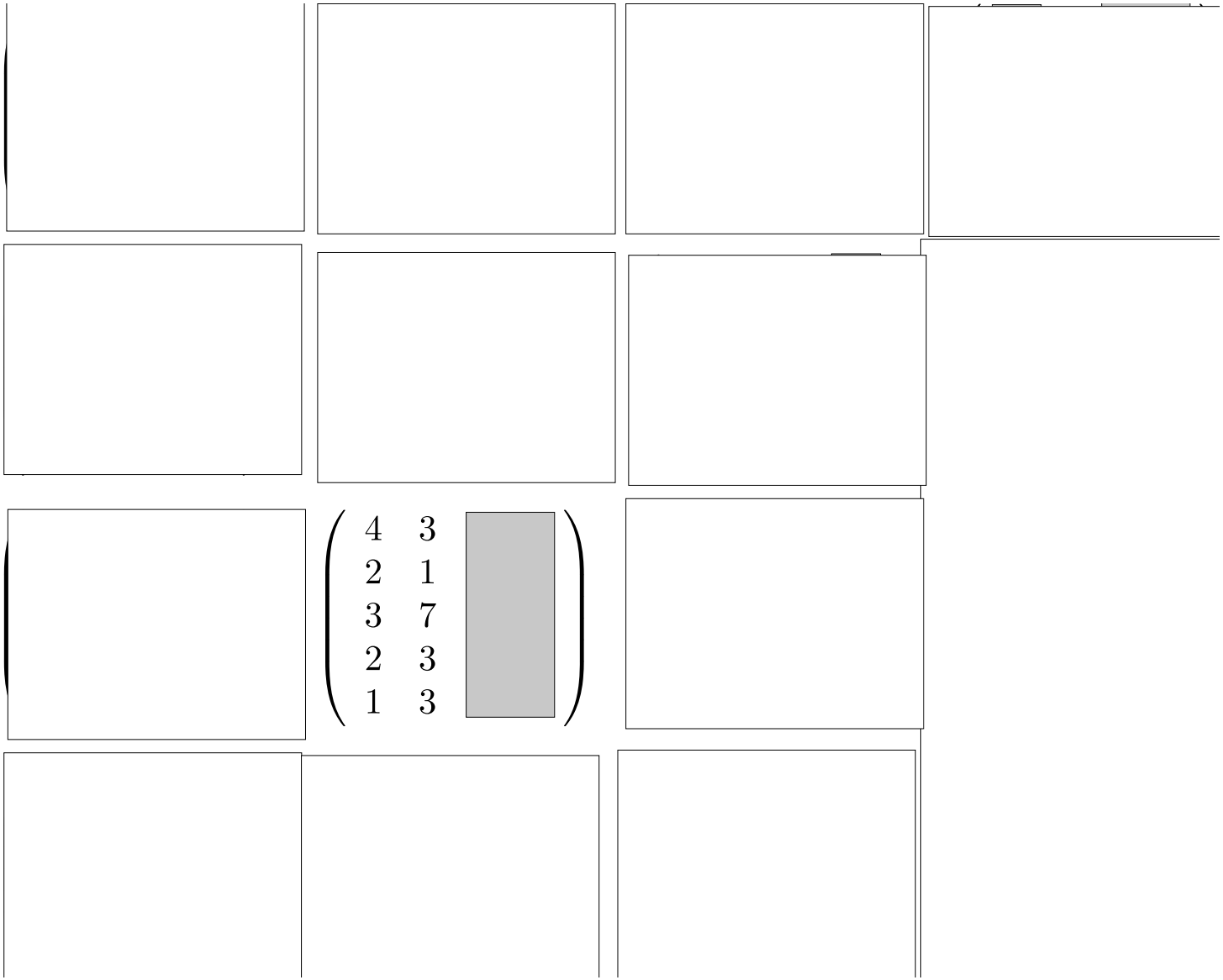
Sélectionner : $\mathscr{F} = \{2\}$

Trouver la meilleure variable à ajouter (3 possibilités= 3 entraînements)

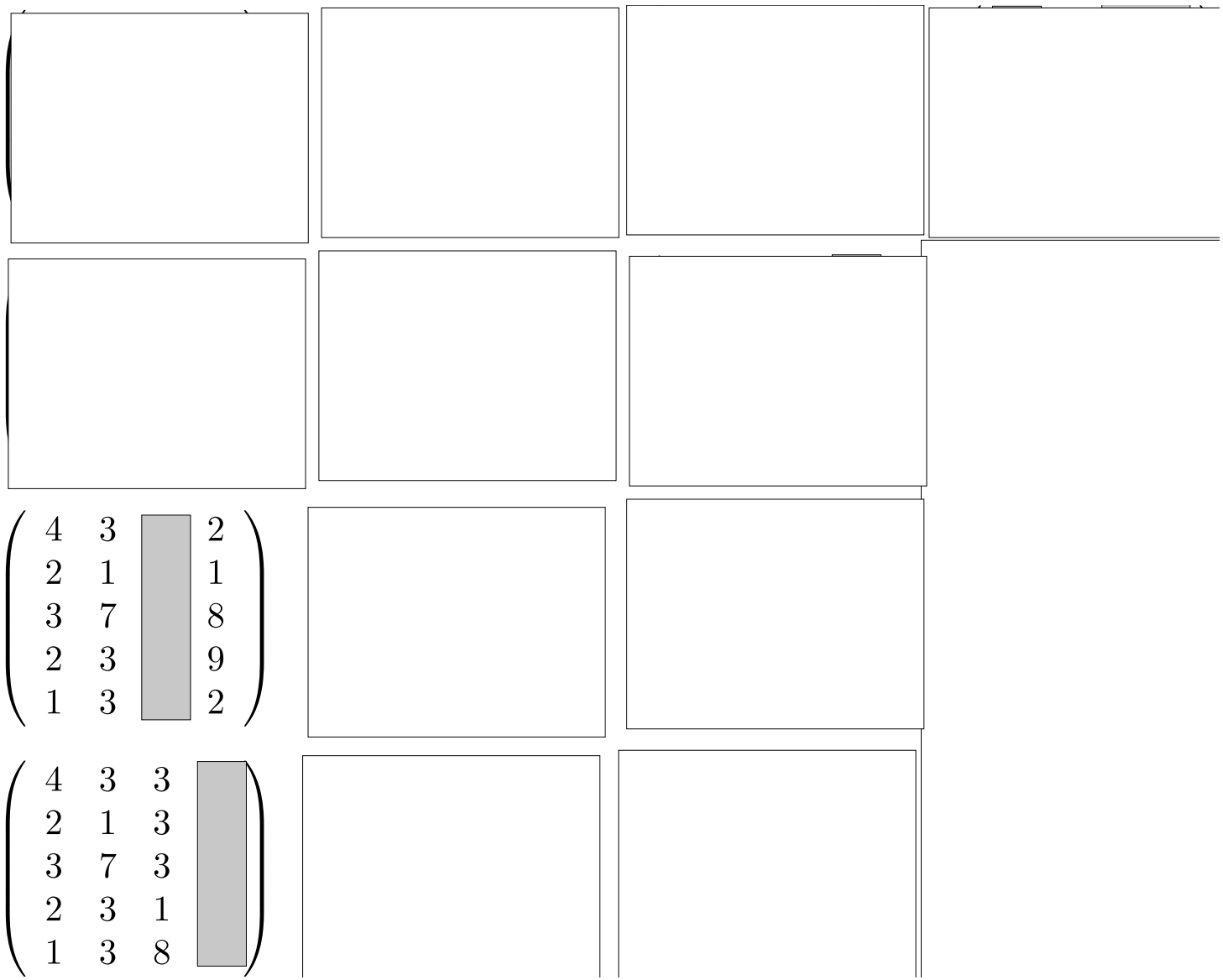


Sélectionner ; meilleur qu'avec une colonne de moins ?

(non $\rightarrow$ STOP ; oui$\rightarrow$ $\mathscr{F} = \{1, 2\}$)

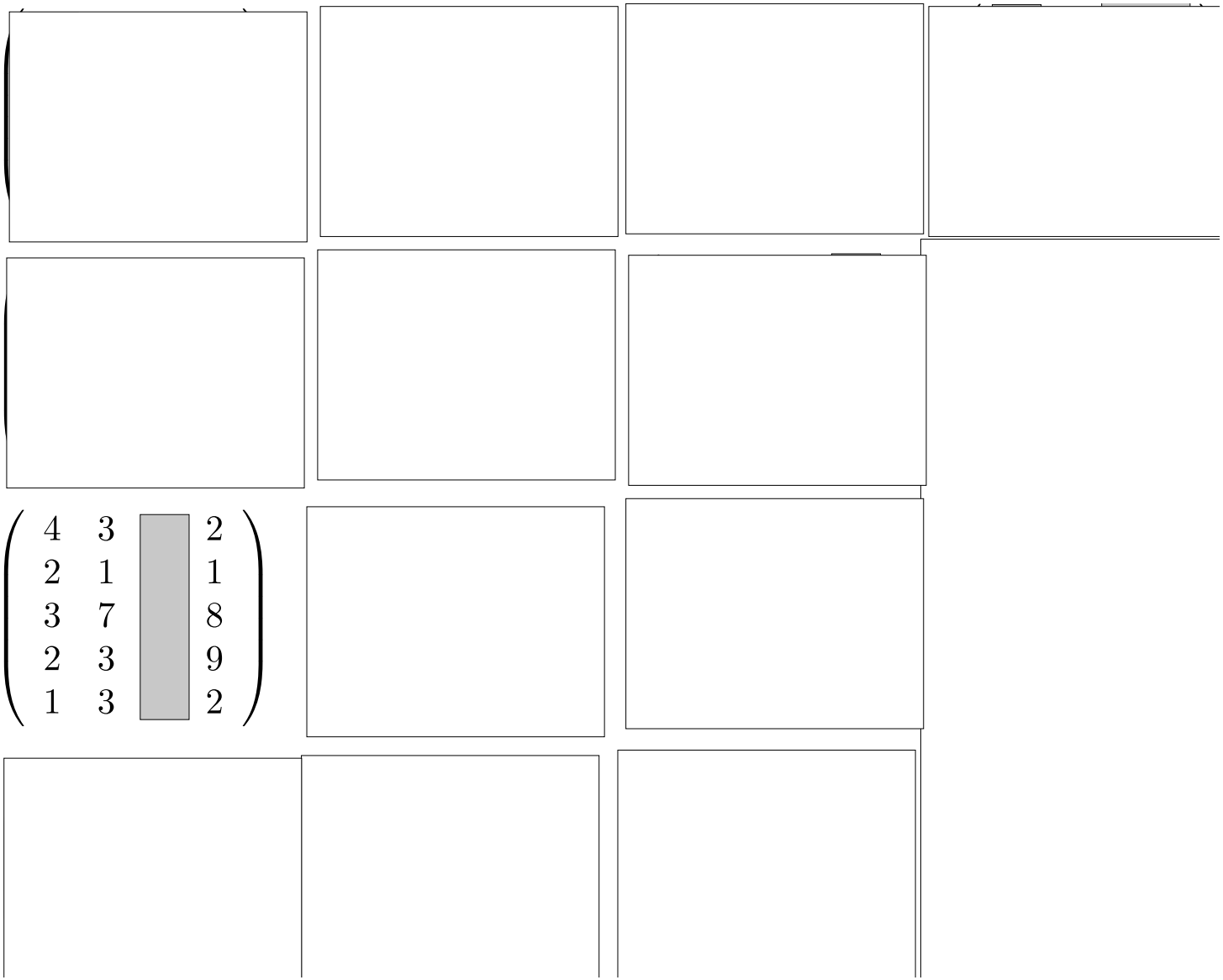


Si meilleur, trouver la meilleure variable à ajouter (2 possibilités= 2 entraînements)

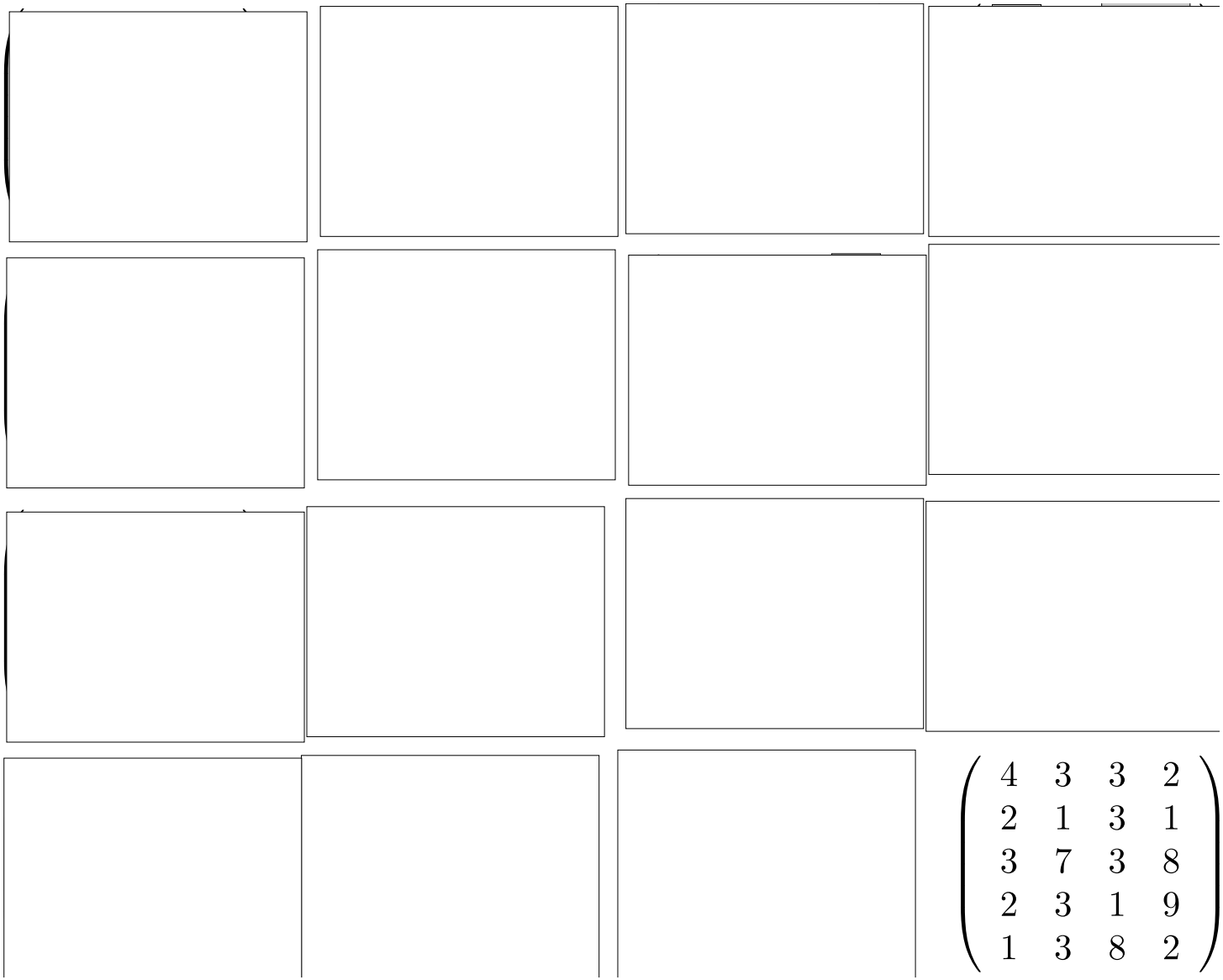


Sélectionner ; meilleur qu'avec une colonne de moins ?

(non $\rightarrow$ STOP ; oui$\rightarrow$ $\mathscr{F} = \{1, 2, 4\}$)



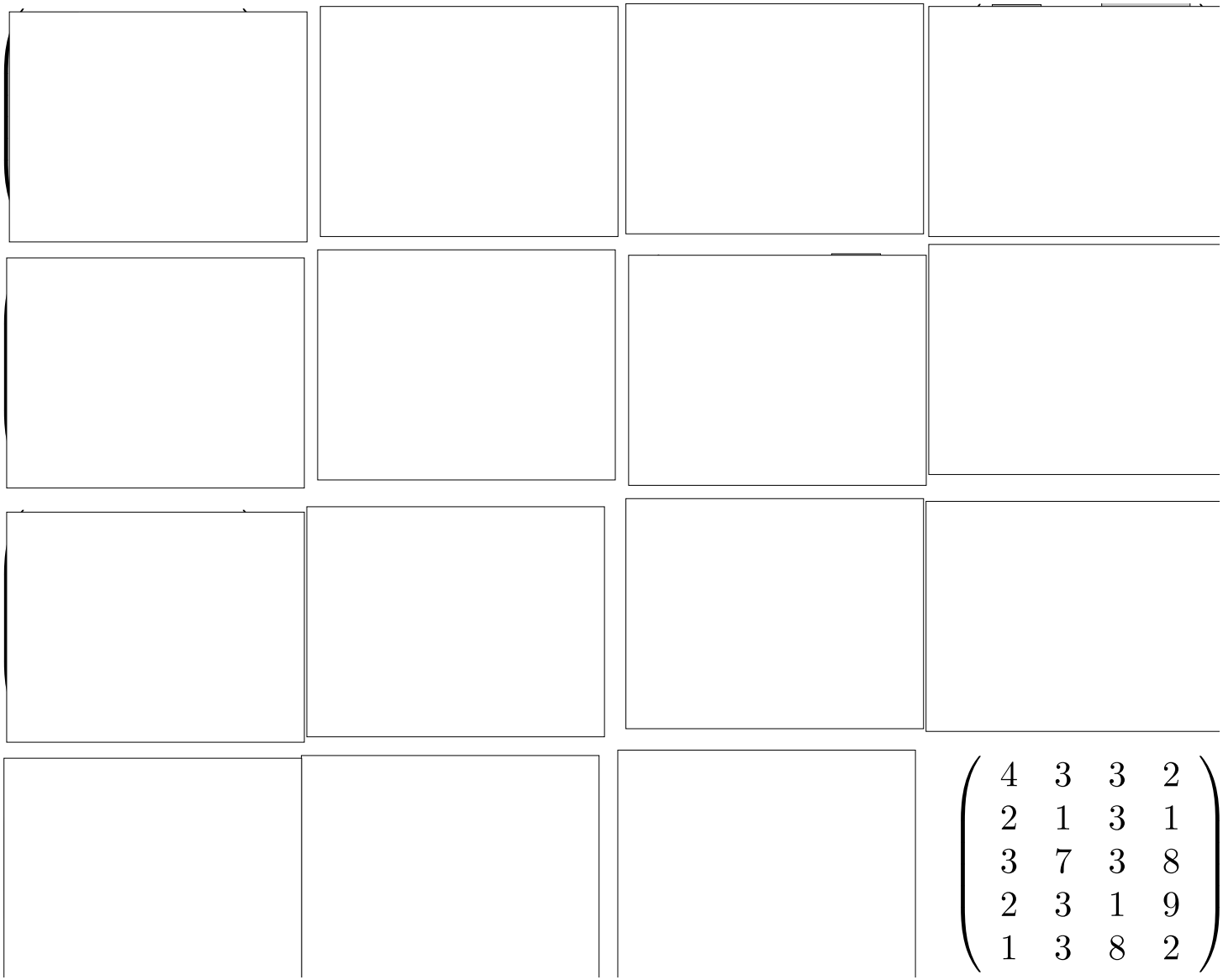
Si meilleur, ajouter la dernière variable (1 entraînement)



$$\begin{pmatrix} 4 & 3 & 3 & 2 \\ 2 & 1 & 3 & 1 \\ 3 & 7 & 3 & 8 \\ 2 & 3 & 1 & 9 \\ 1 & 3 & 8 & 2 \end{pmatrix}$$

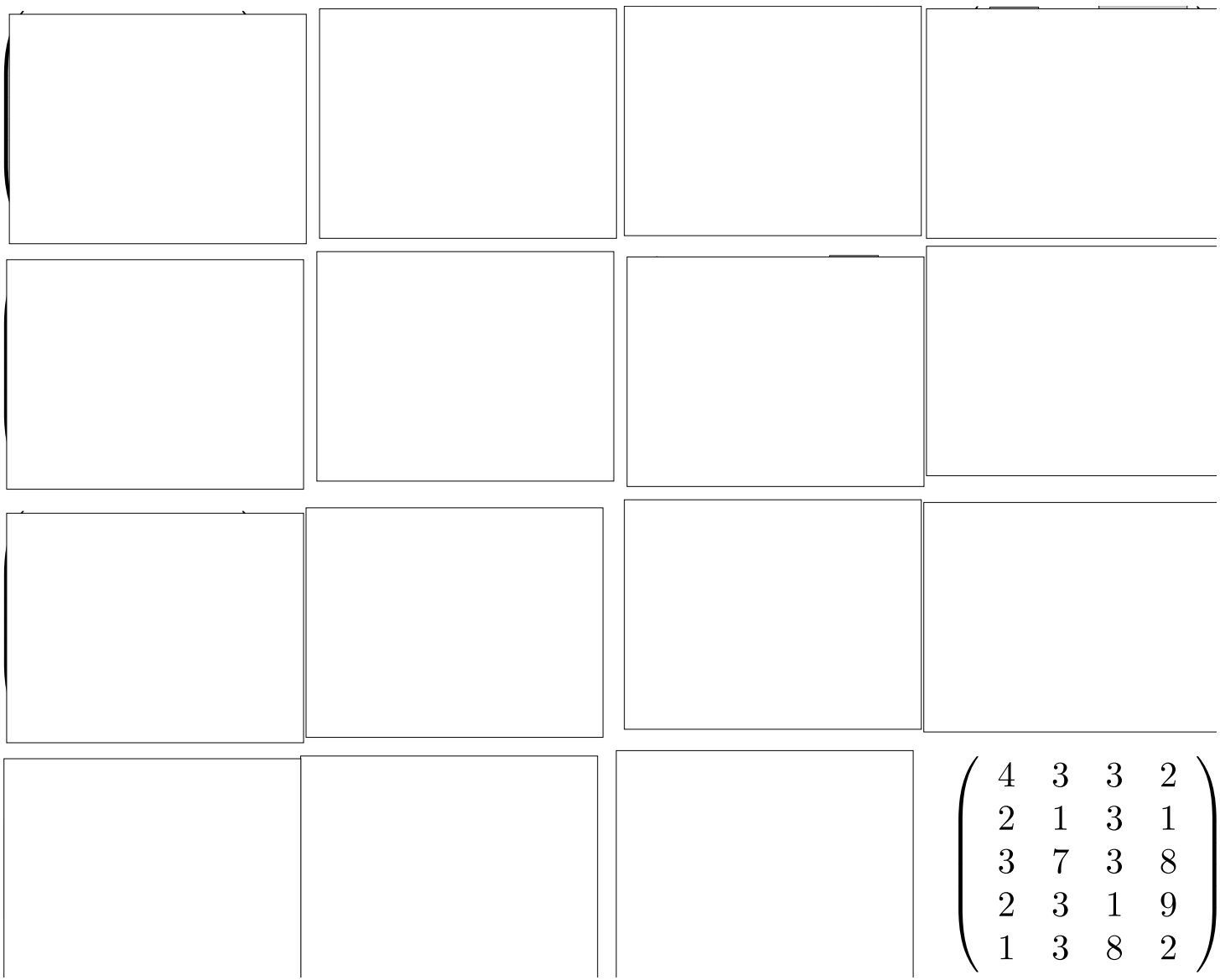
Sélectionner ; meilleur qu'avec une colonne de moins ?

(non $\rightarrow$ STOP ; oui$\rightarrow$ $\mathscr{F} = \{1, 2, 3, 4\}$)



$$\begin{pmatrix} 4 & 3 & 3 & 2 \\ 2 & 1 & 3 & 1 \\ 3 & 7 & 3 & 8 \\ 2 & 3 & 1 & 9 \\ 1 & 3 & 8 & 2 \end{pmatrix}$$

Maximum 4+3+2+1 $\sim \mathcal{O}(p^2)$ entraînements effectués



$$\begin{pmatrix} 4 & 3 & 3 & 2 \\ 2 & 1 & 3 & 1 \\ 3 & 7 & 3 & 8 \\ 2 & 3 & 1 & 9 \\ 1 & 3 & 8 & 2 \end{pmatrix}$$

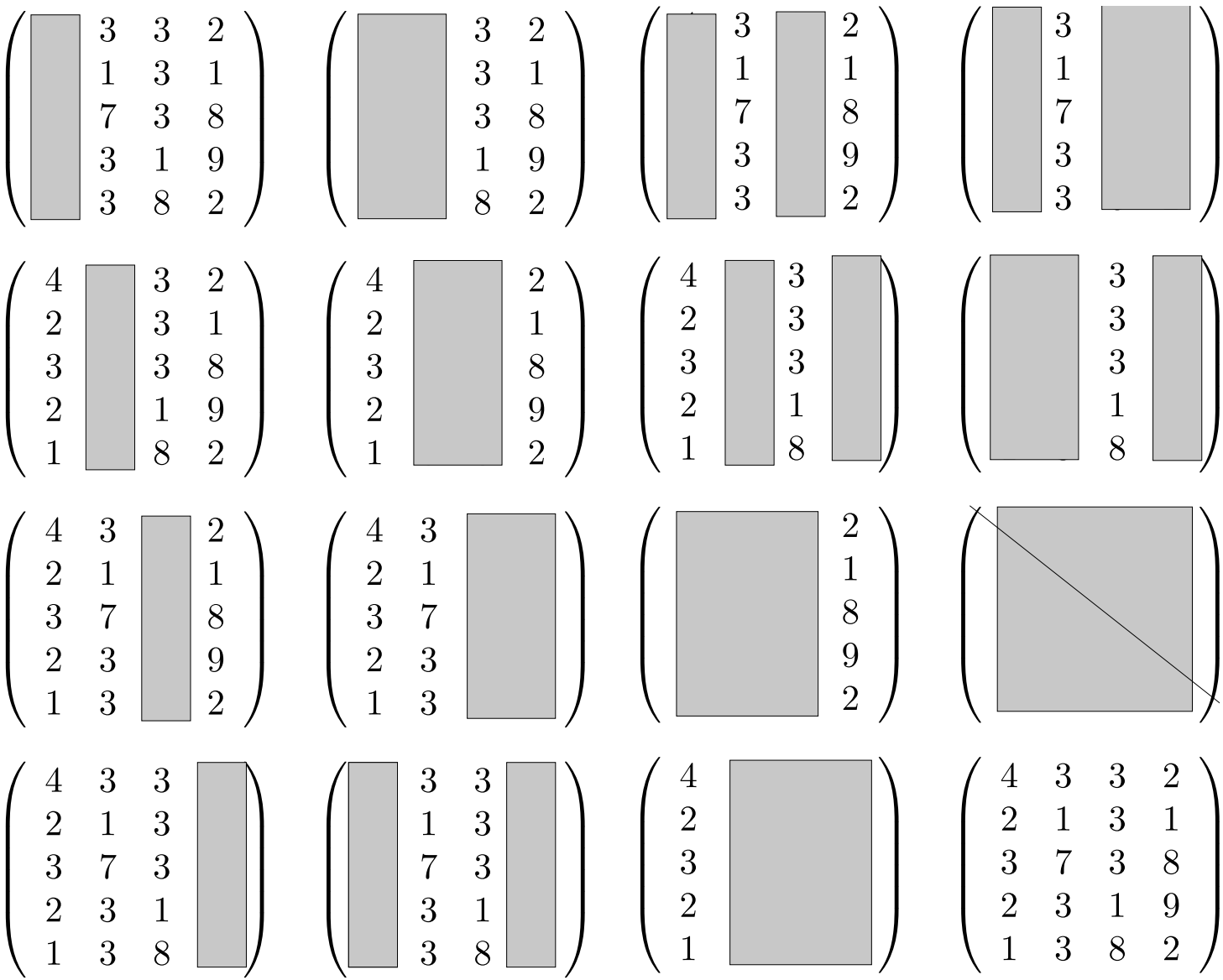
Recherche descendante

Définition 11.3 (Recherche descendante) On appelle recherche descendante, ou backward search en anglais, la procédure gloutonne de sélection de variables suivante :

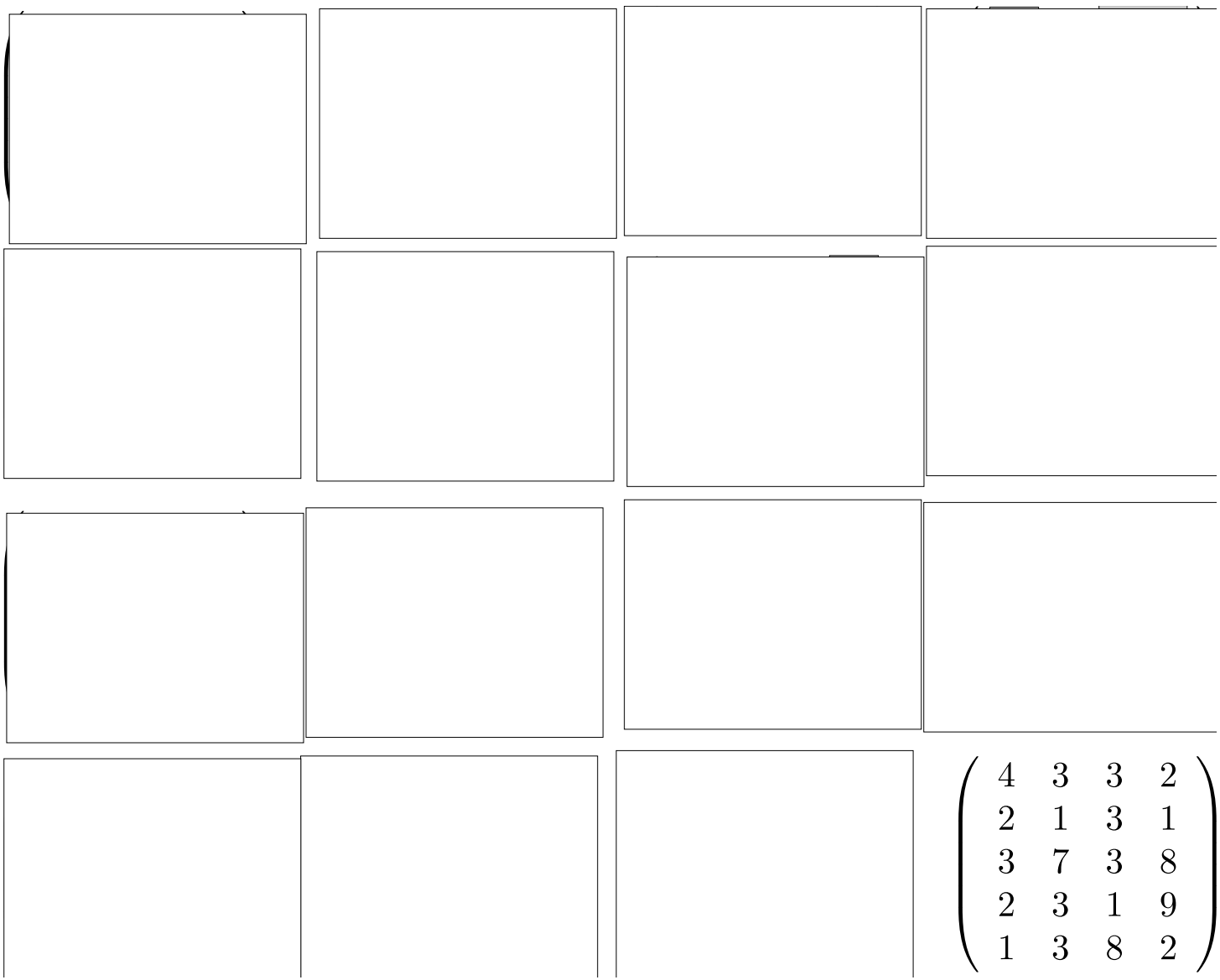
1. Initialiser $\mathscr{F} = \{1, \ldots, p\}$
2. Trouver la meilleure variable à retirer à $\mathscr{F}$:
   $j^* = \underset{j \in \mathscr{F}}{\operatorname{argmin}} E_{\mathscr{D}}\left(\mathscr{F} \setminus \{j\}\right)$
3. Si $E_{\mathscr{D}}\left(\mathscr{F} \setminus \{j\}\right) > E_{\mathscr{D}}\left(\mathscr{F}\right)$: s'arrêter
   Sinon : $\mathscr{F} \leftarrow \mathscr{F} \setminus \{j\}$: recommencer 2–3.

Note : L'avantage de l'approche descendante sur l'approche ascendante est qu'elle fournit nécessairement un sous-ensemble de variables meilleur que l'intégralité des variables. En effet, ce n'est pas parce qu'on ne peut pas, à une étape de la méthode ascendante, trouver de variable à ajouter à $\mathscr{F}$, que la performance de l'algorithme est meilleure sur $(X_{\mathscr{F}}, \vec{y})$ que sur $(X, \vec{y})$.

Recherche descendante : exemple



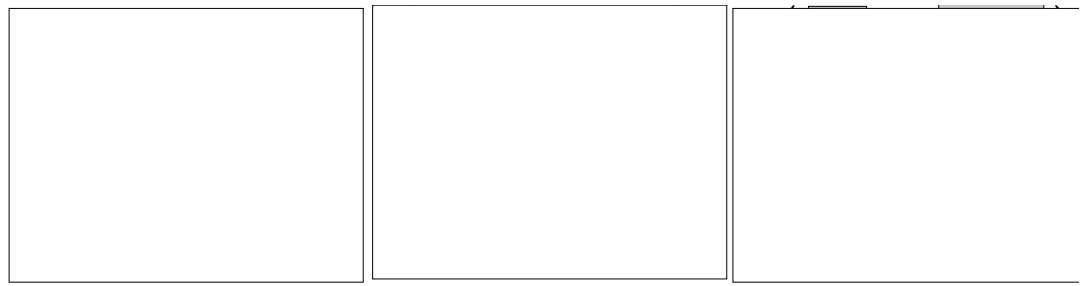
Initialisation $\mathscr{F} = \{1, 2, 3, 4\}$



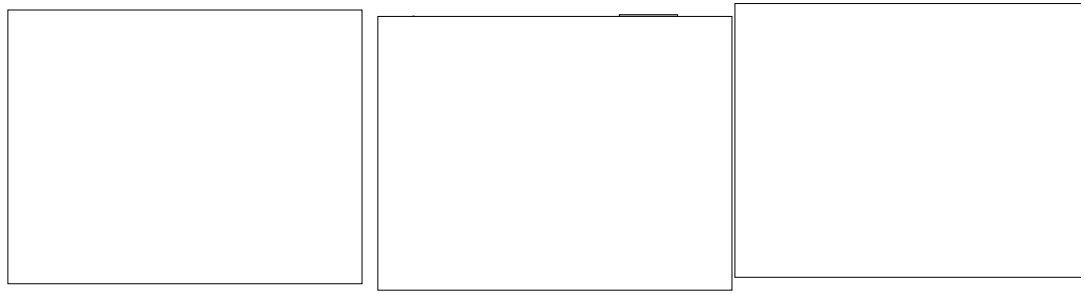
$$\begin{pmatrix} 4 & 3 & 3 & 2 \\ 2 & 1 & 3 & 1 \\ 3 & 7 & 3 & 8 \\ 2 & 3 & 1 & 9 \\ 1 & 3 & 8 & 2 \end{pmatrix}$$

Trouver la meilleure variable à enlever (4 possibilités = 4 entraînements)

$$\begin{pmatrix} \square & 3 & 3 & 2 \\ & 1 & 3 & 1 \\ & 7 & 3 & 8 \\ & 3 & 1 & 9 \\ & 3 & 8 & 2 \end{pmatrix}$$



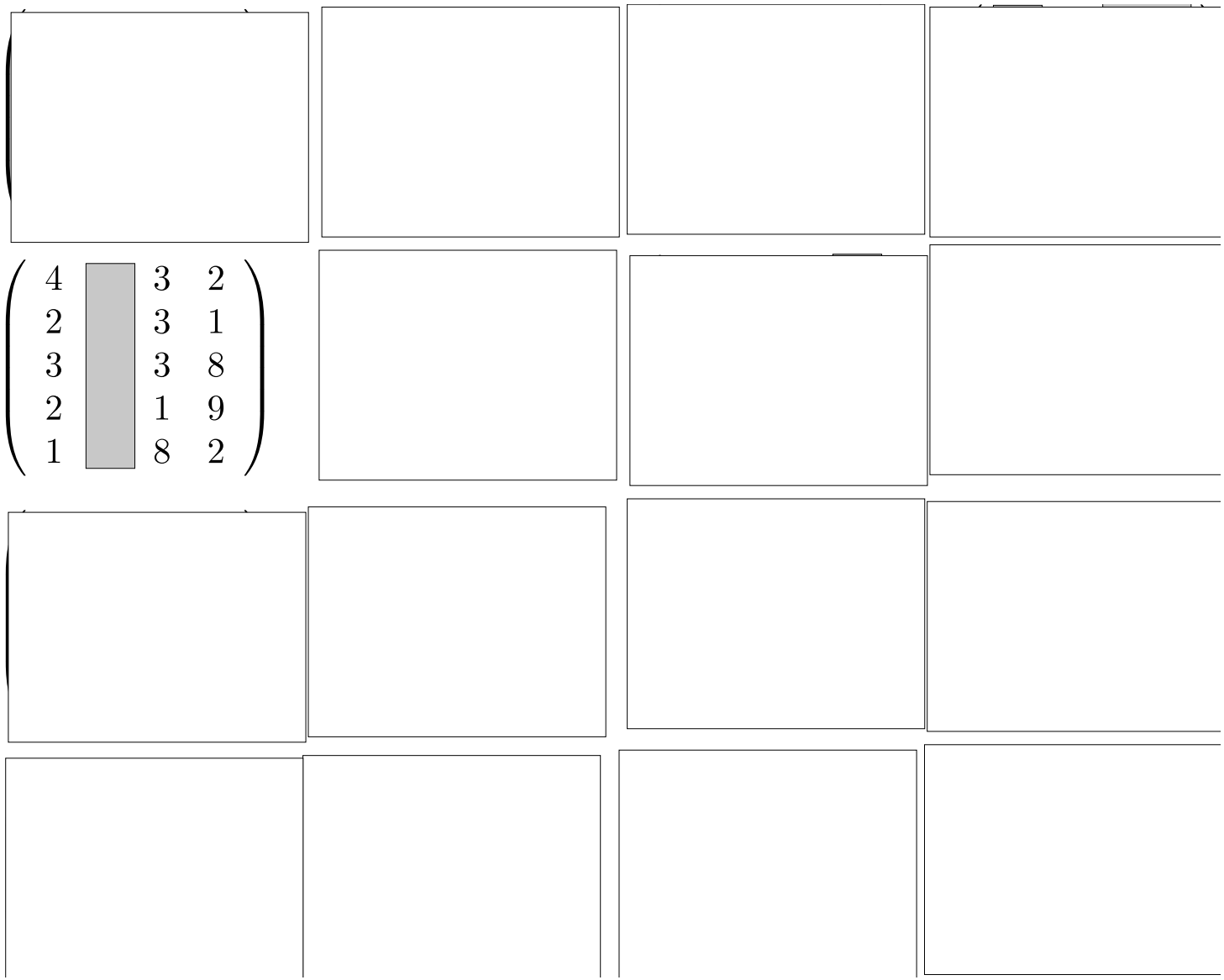
$$\begin{pmatrix} 4 & \square & 3 & 2 \\ 2 & & 3 & 1 \\ 3 & & 3 & 8 \\ 2 & & 1 & 9 \\ 1 & & 8 & 2 \end{pmatrix}$$



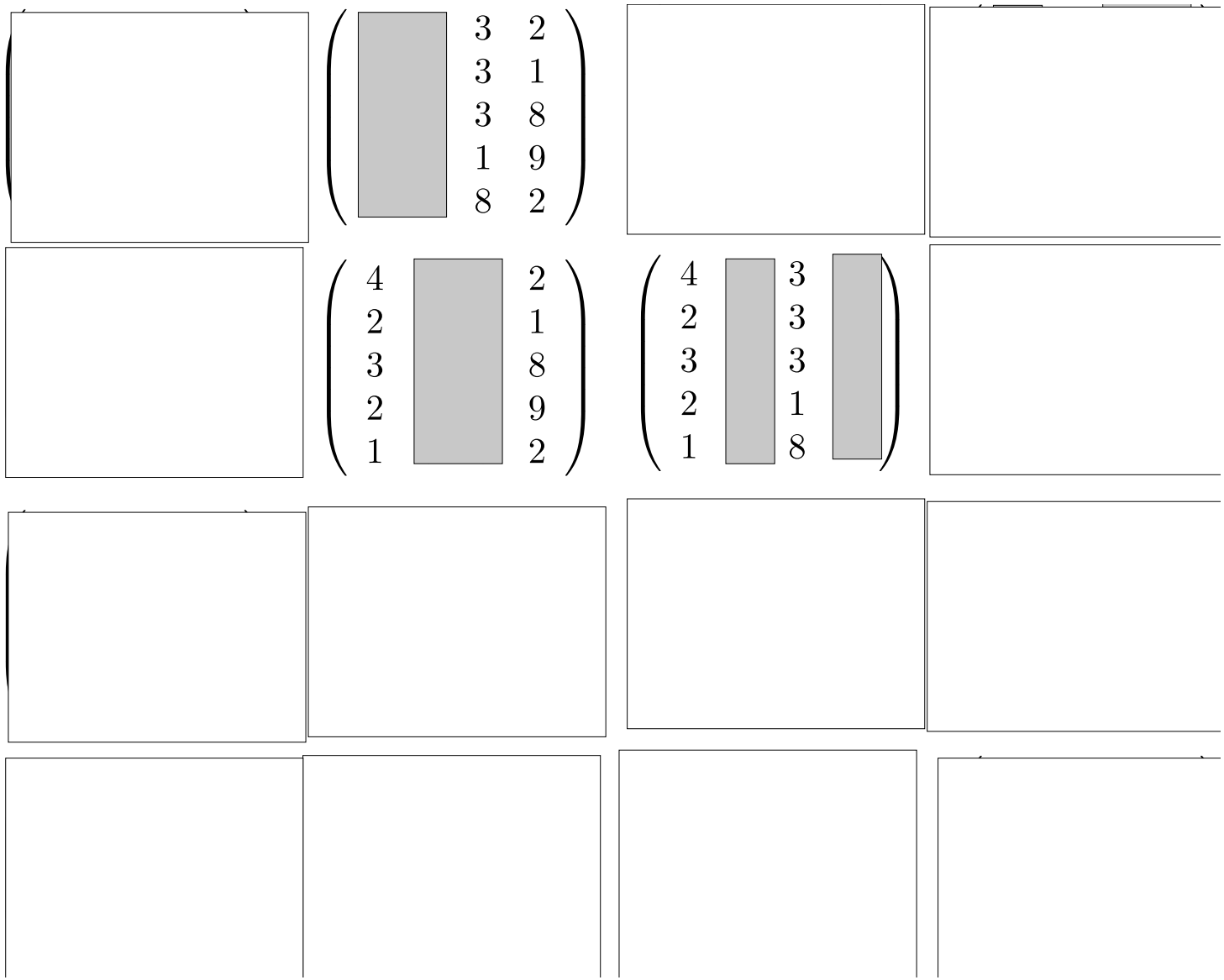
$$\begin{pmatrix} 4 & 3 & \square & 2 \\ 2 & 1 & & 1 \\ 3 & 7 & & 8 \\ 2 & 3 & & 9 \\ 1 & 3 & & 2 \end{pmatrix}$$

$$\begin{pmatrix} 4 & 3 & 3 & \square \\ 2 & 1 & 3 & \\ 3 & 7 & 3 & \\ 2 & 3 & 1 & \\ 1 & 3 & 8 & \end{pmatrix}$$

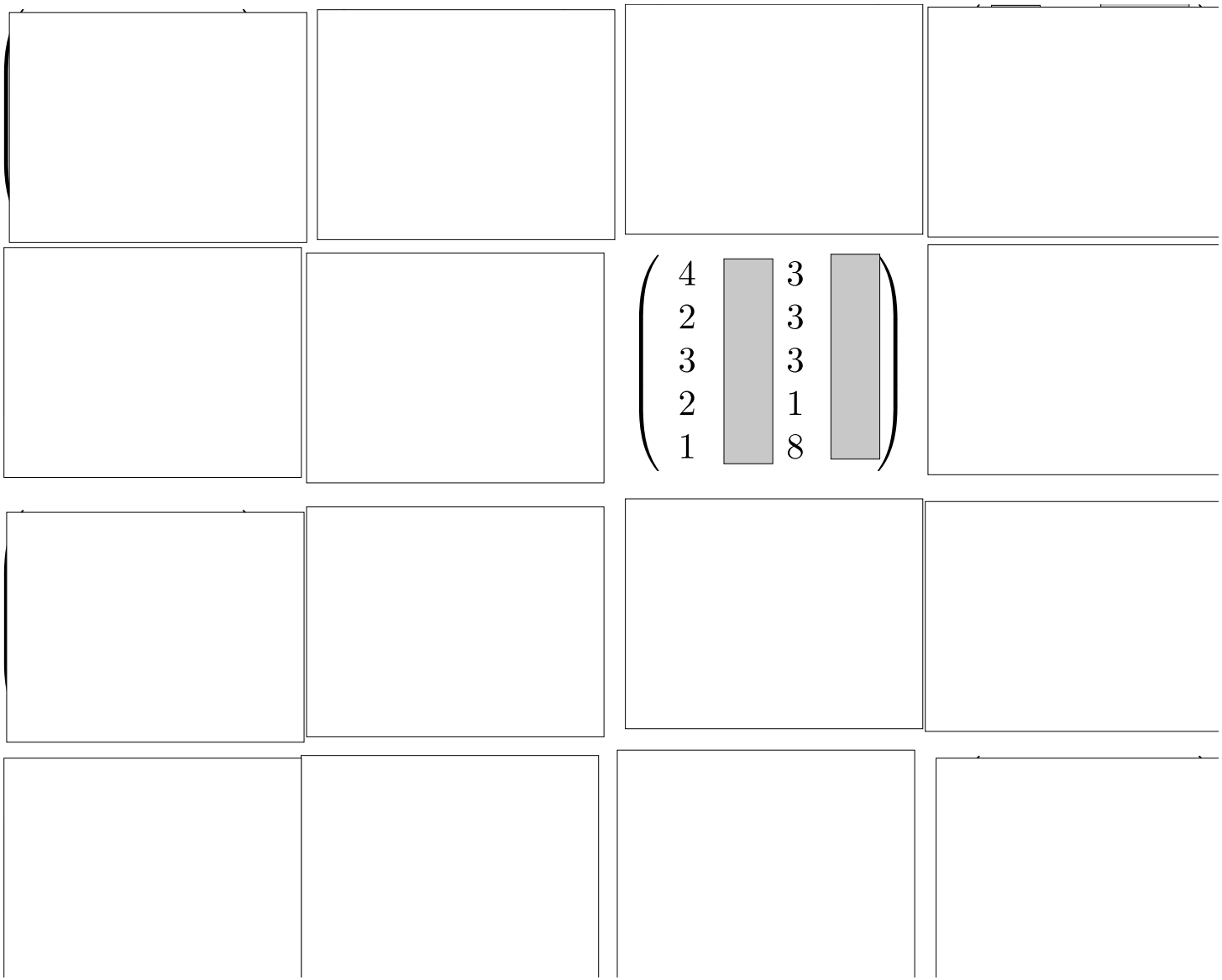
Subset meilleur qu'avec une colonne de plus ? (non $\rightarrow$ STOP ; oui$\rightarrow$ $\mathscr{F} = \{1, 3, 4\}$)



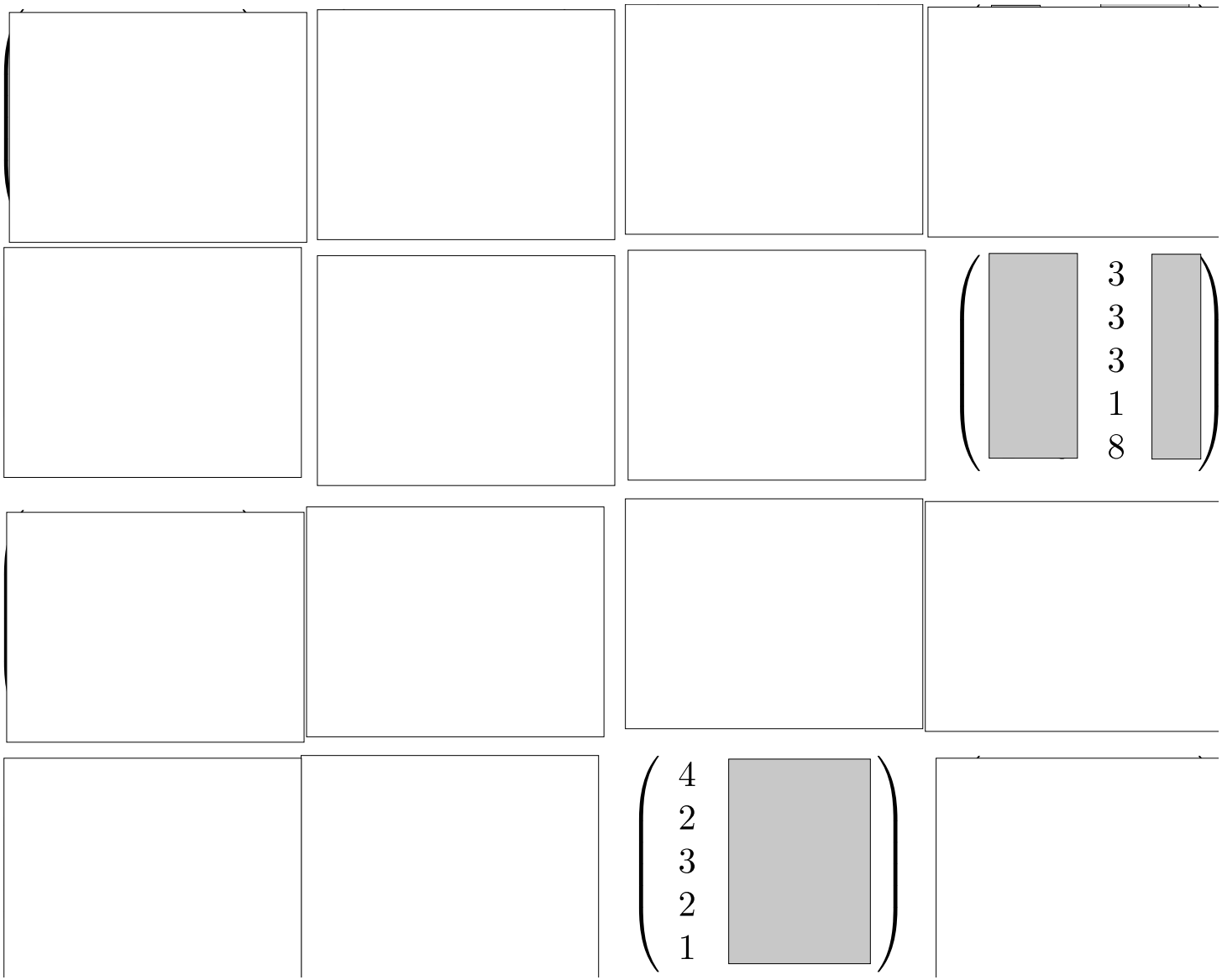
Trouver la meilleure variable à enlever (3 possibilités= 3 entraînements)



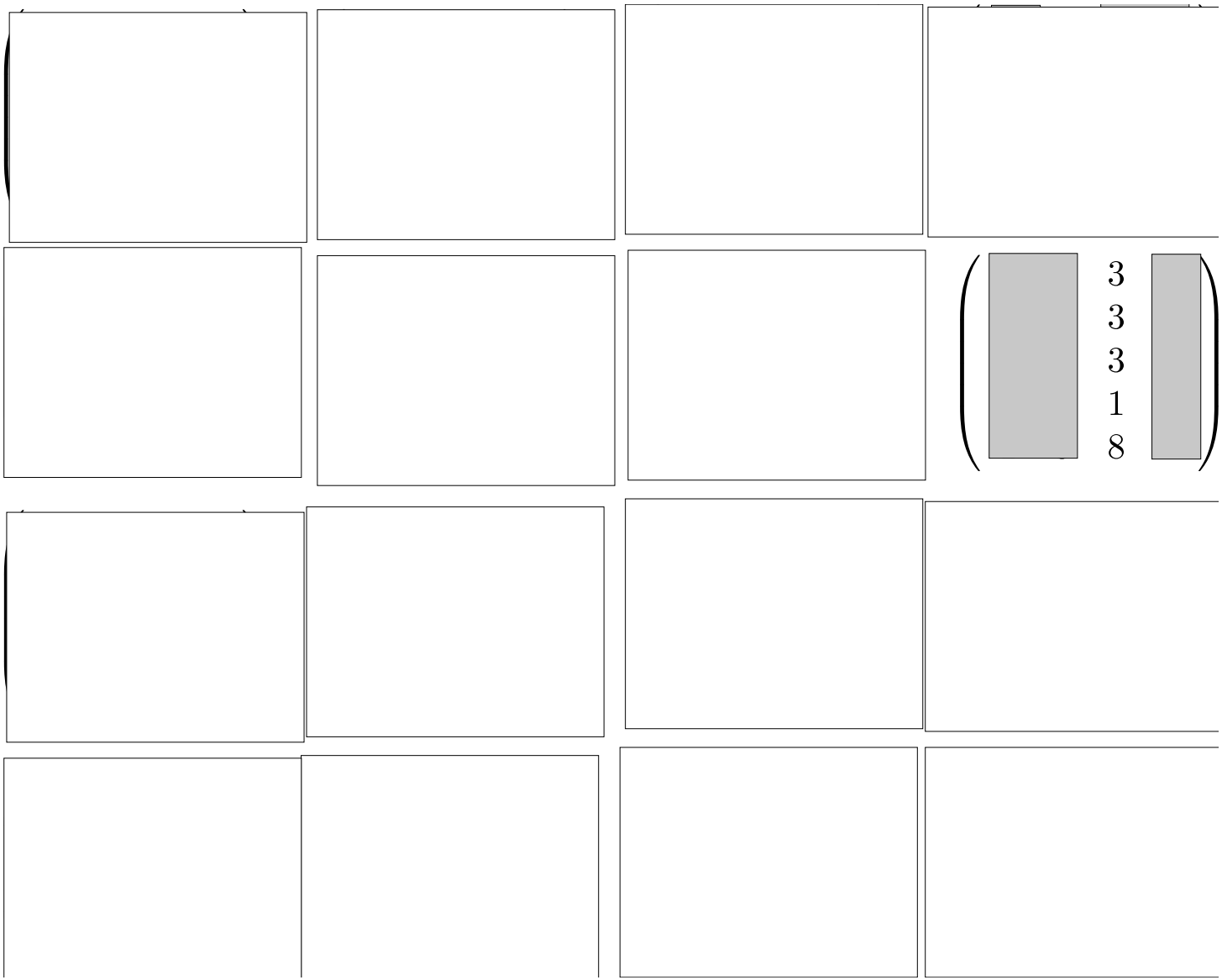
Subset meilleur qu'avec une colonne de plus ? (non $\rightarrow$ STOP ; oui$\rightarrow$ $\mathscr{F} = \{1, 3\}$)



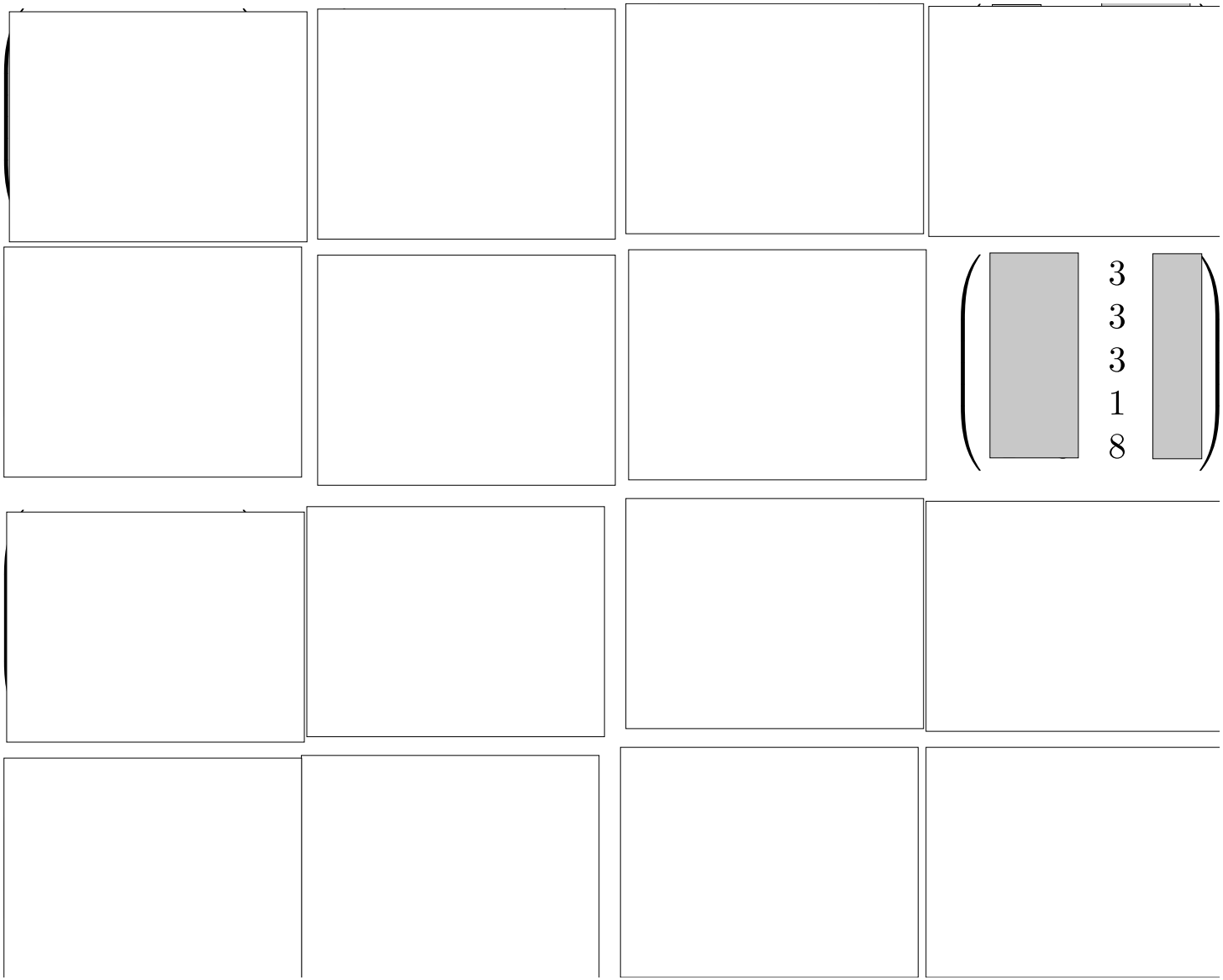
Si meilleur, trouver la meilleure variable à enlever (2 possibilités= 2 entraînements)



Subset meilleur qu'avec une colonne de plus ? (non $\rightarrow$ STOP ; oui$\rightarrow$ $\mathscr{F} = \{3\}$)



Maximum 4+3+2+1 $\sim \mathcal{O}(p^2)$ entraînements effectués

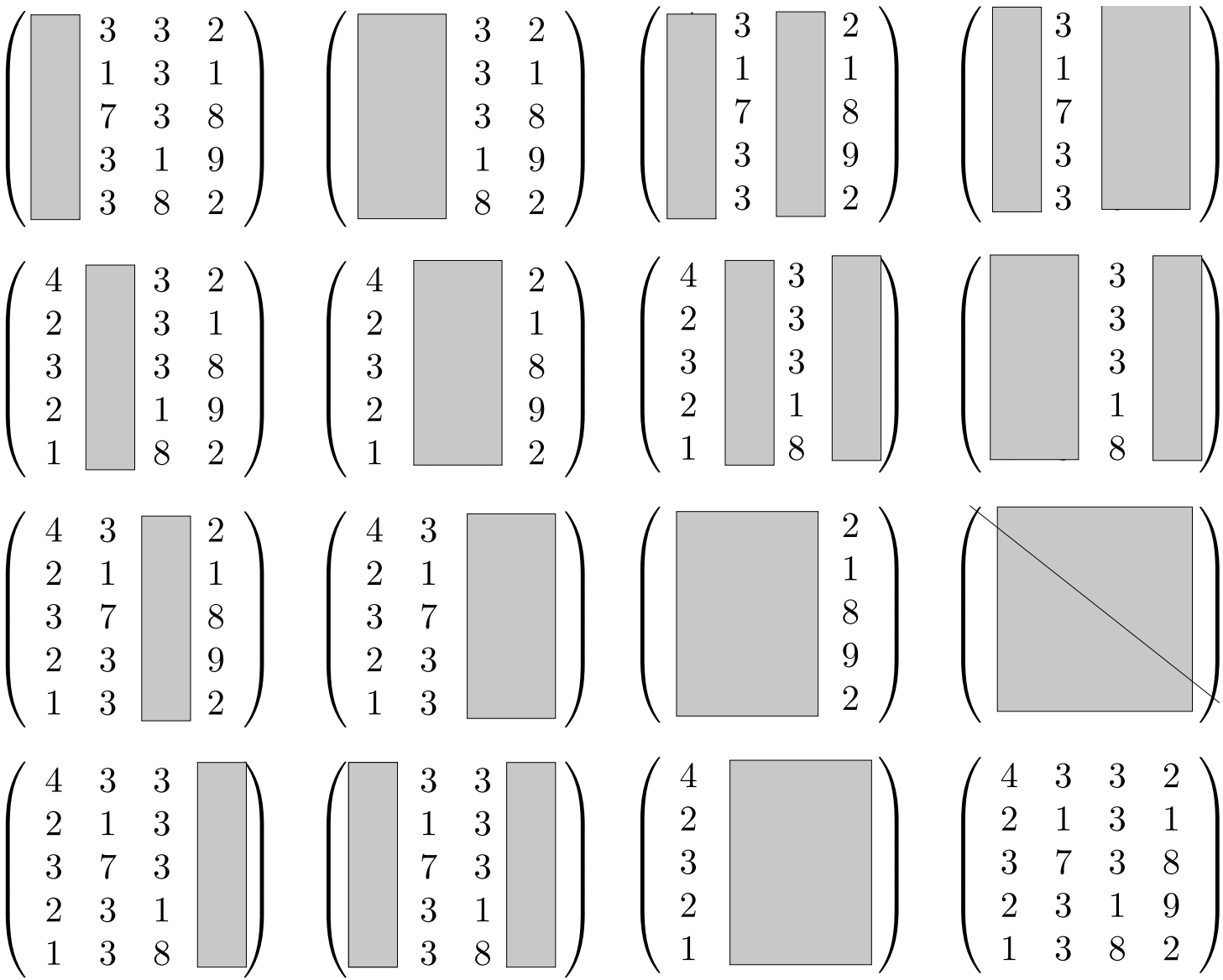


Recherche flottante

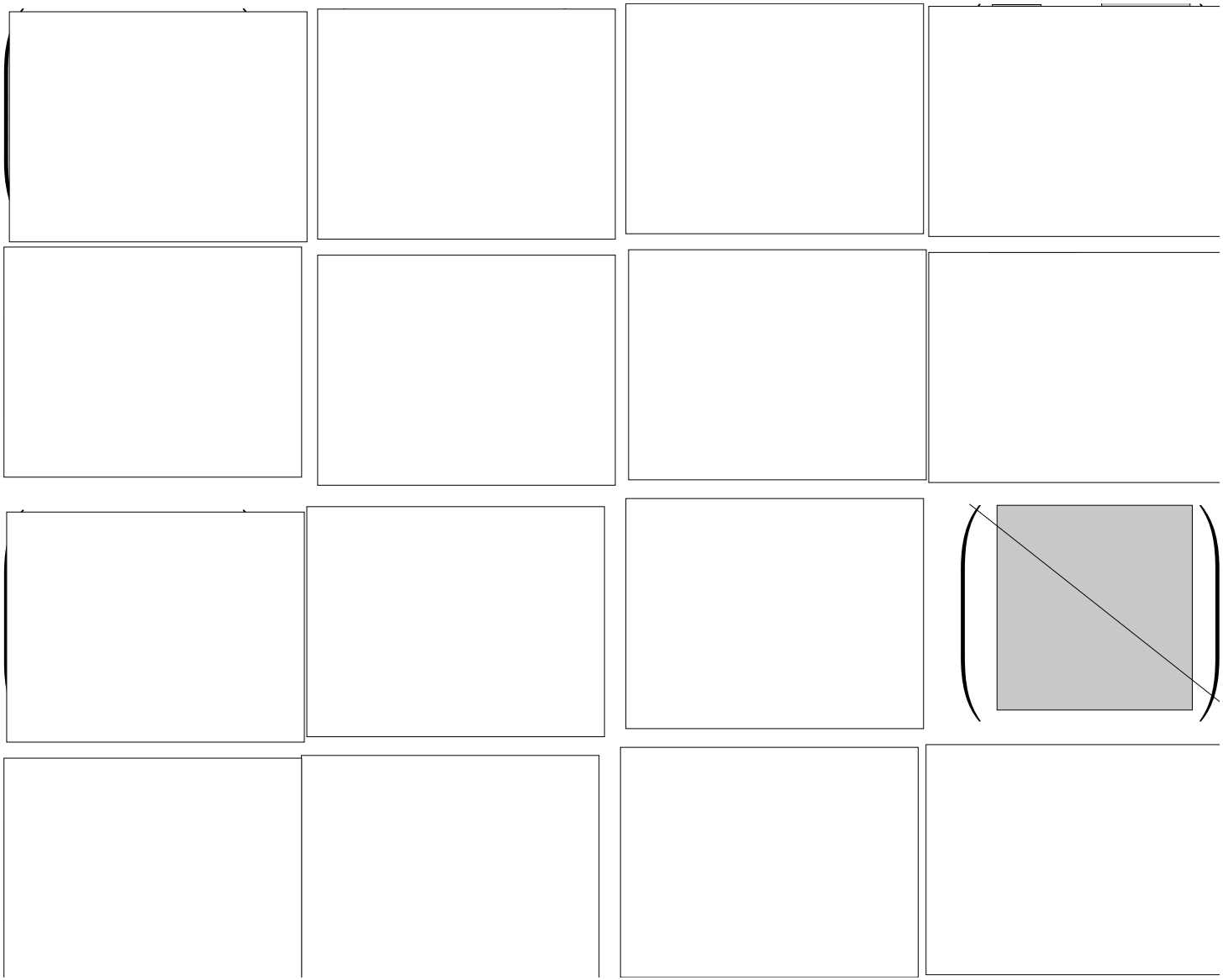
Définition 11.4 (Recherche flottante) Étant donné deux paramètres entiers strictement positifs q et r ($q > r > 0$), on appelle recherche flottante, ou floating search en anglais, la procédure gloutonne de sélection de variables suivante :

1. Initialiser $\mathscr{F} = \emptyset$
2. Trouver les q meilleures variables à ajouter à $\mathscr{F}$:
   $\mathcal{S}^* = \underset{\mathcal{S} \subseteq \{1,\dots,p\} \setminus \mathscr{F}, |\mathcal{S}| = q}{\operatorname{argmin}} E_{\mathscr{D}}(\mathscr{F} \cup \mathcal{S})$
3. Si $E_{\mathscr{D}}(\mathscr{F} \cup \mathcal{S}^*) < E_{\mathscr{D}}(\mathscr{F})$: $\mathscr{F} \leftarrow \mathscr{F} \cup \mathcal{S}^*$
4. trouver les r meilleures variables à retirer de $\mathscr{F}$:
   $\mathcal{S}^* = \underset{\mathcal{S} \subseteq \{1,\dots,p\} \setminus \mathscr{F}, |\mathcal{S}| = q}{\operatorname{argmin}} E_{\mathscr{D}}(\mathscr{F} \setminus \mathcal{S})$
5. Si $E_{\mathscr{D}}(\mathscr{F} \setminus \mathcal{S}^*) > E_{\mathscr{D}}(\mathscr{F})$: s'arrêter
   Sinon $\mathscr{F} \leftarrow \mathscr{F} \setminus \mathcal{S}^*$; recommencer 2–5.

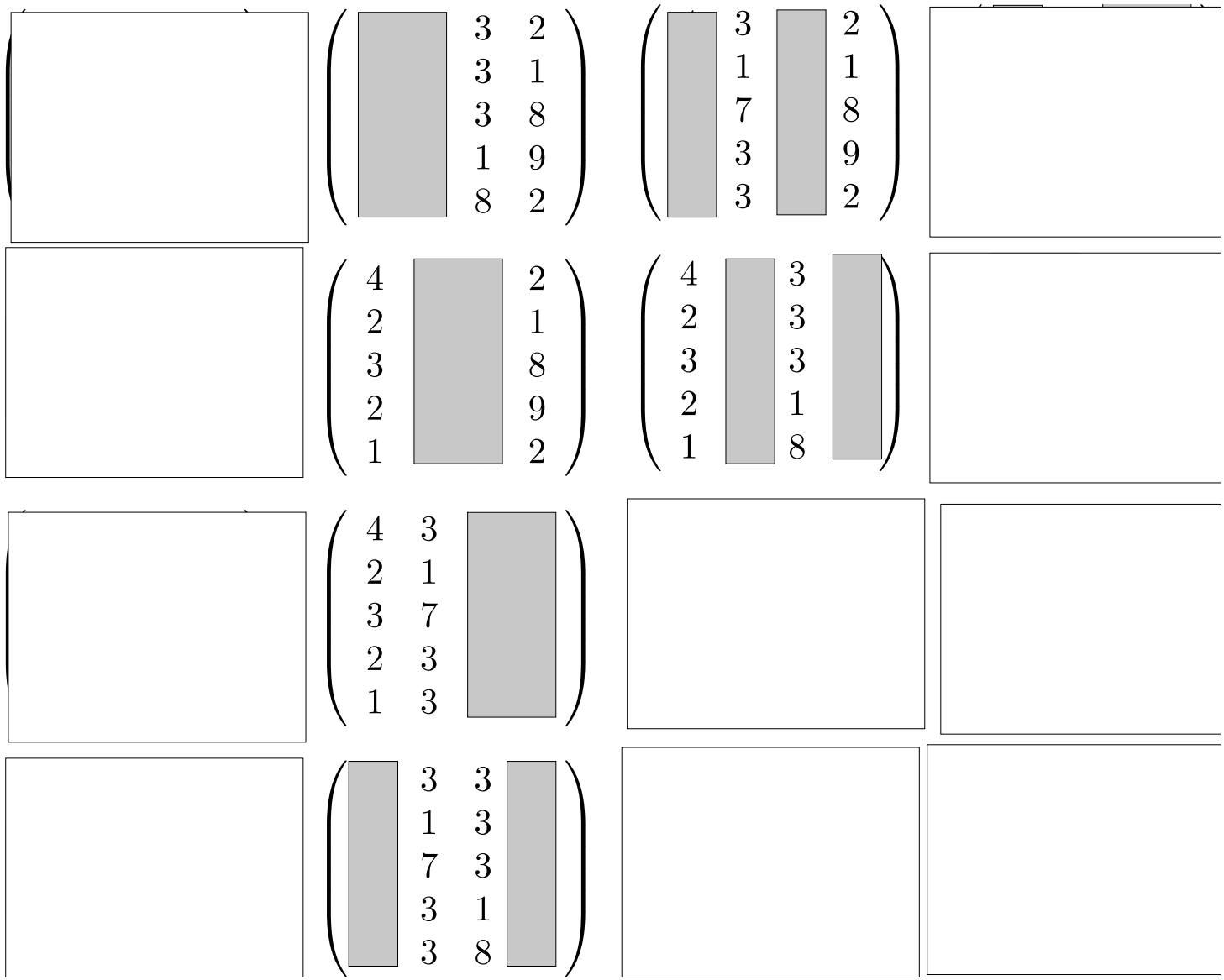
Recherche flottante : exemple ($p = 4 > q = 2 > r = 1 > 0$)



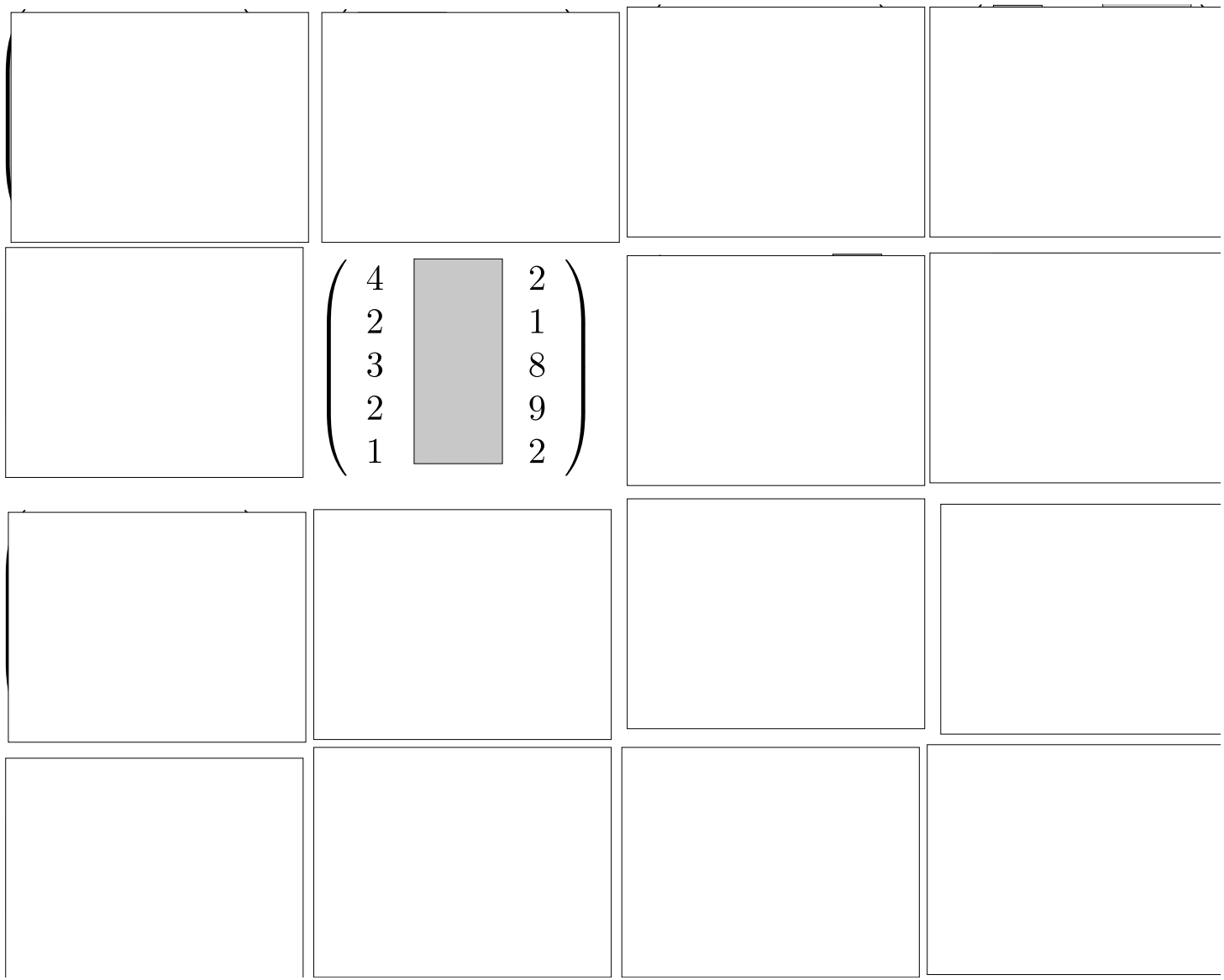
Initialisation $\mathscr{F} = \emptyset$



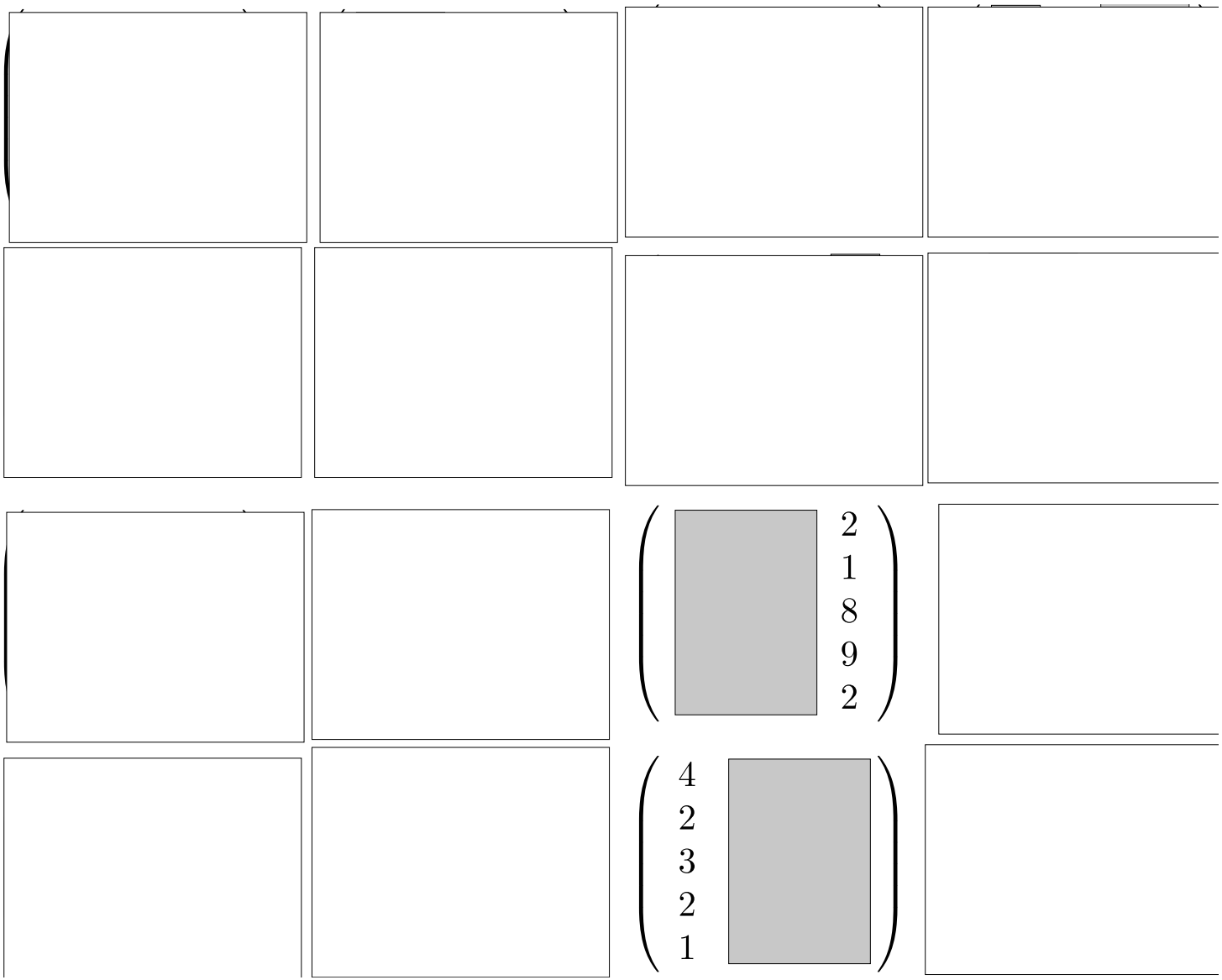
Trouver les $q = 2$ meilleures variables à ajouter (6 possibilités = 6 entraînements)



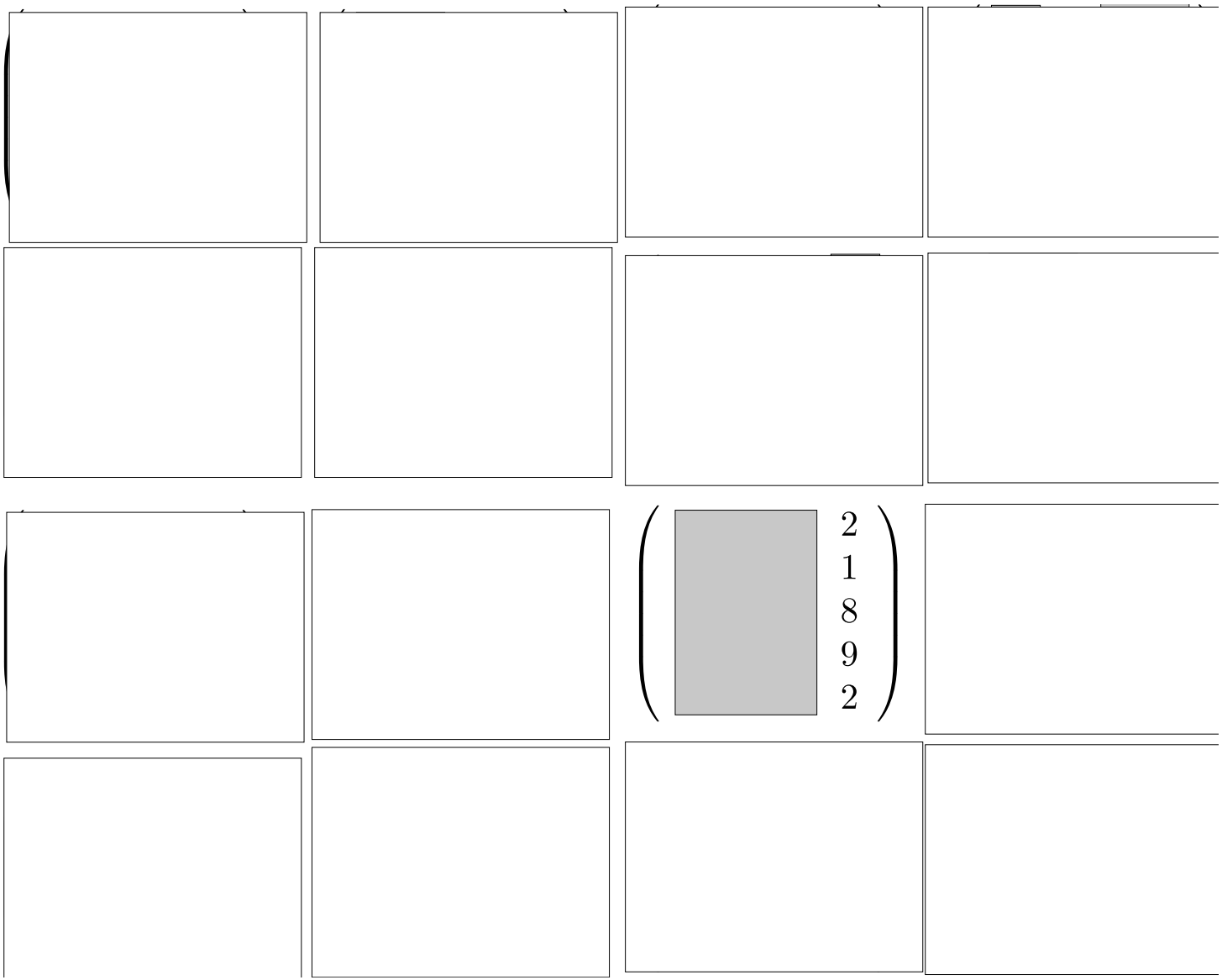
Subset choisi meilleur qu'avec q colonne de moins ? (non $\rightarrow$ STOP ; oui$\rightarrow$ $\mathscr{F} = \{1, 4\}$)



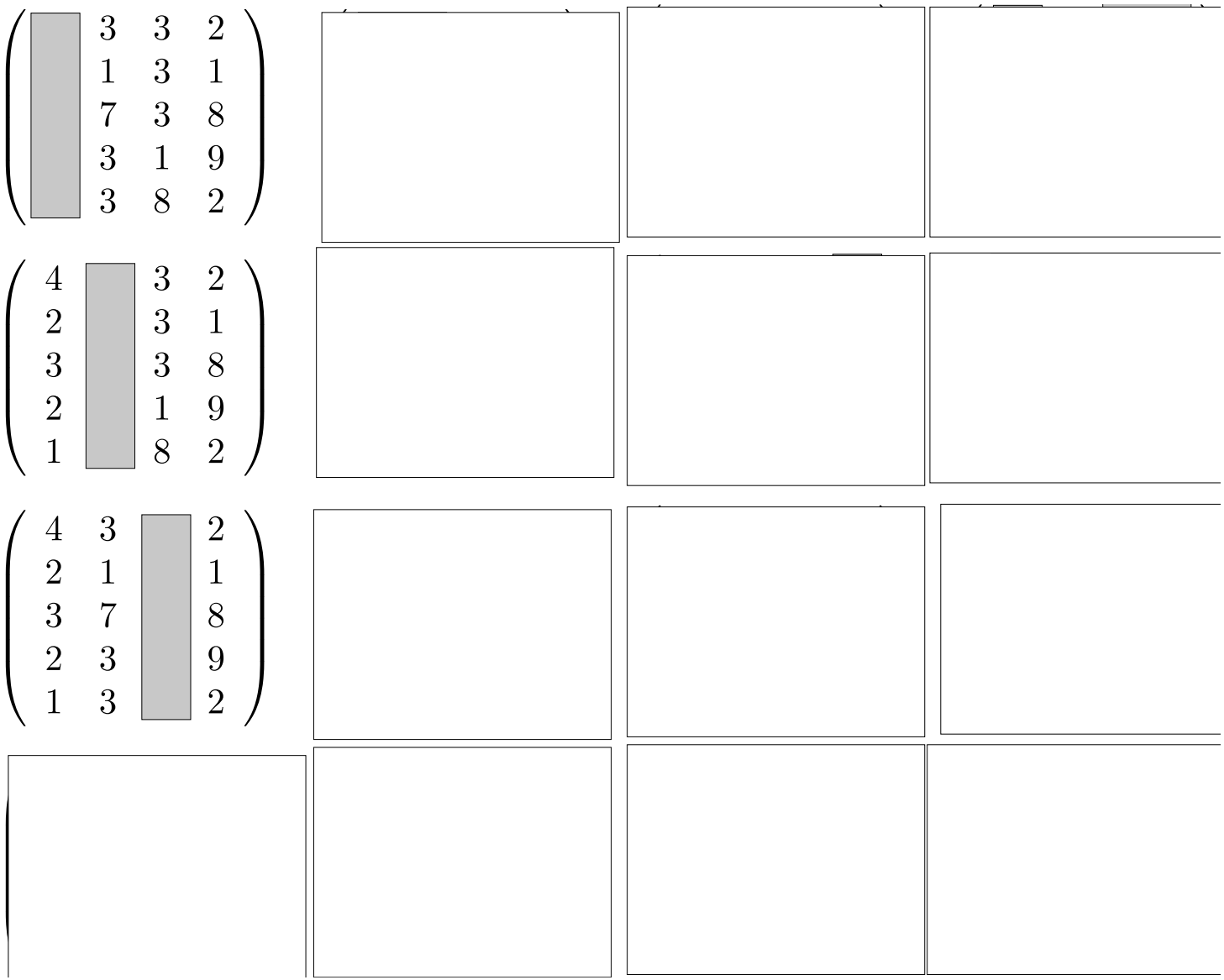
Trouver la ($p = 1$) meilleure variable à enlever (2 possibilités= 2 entraînements)



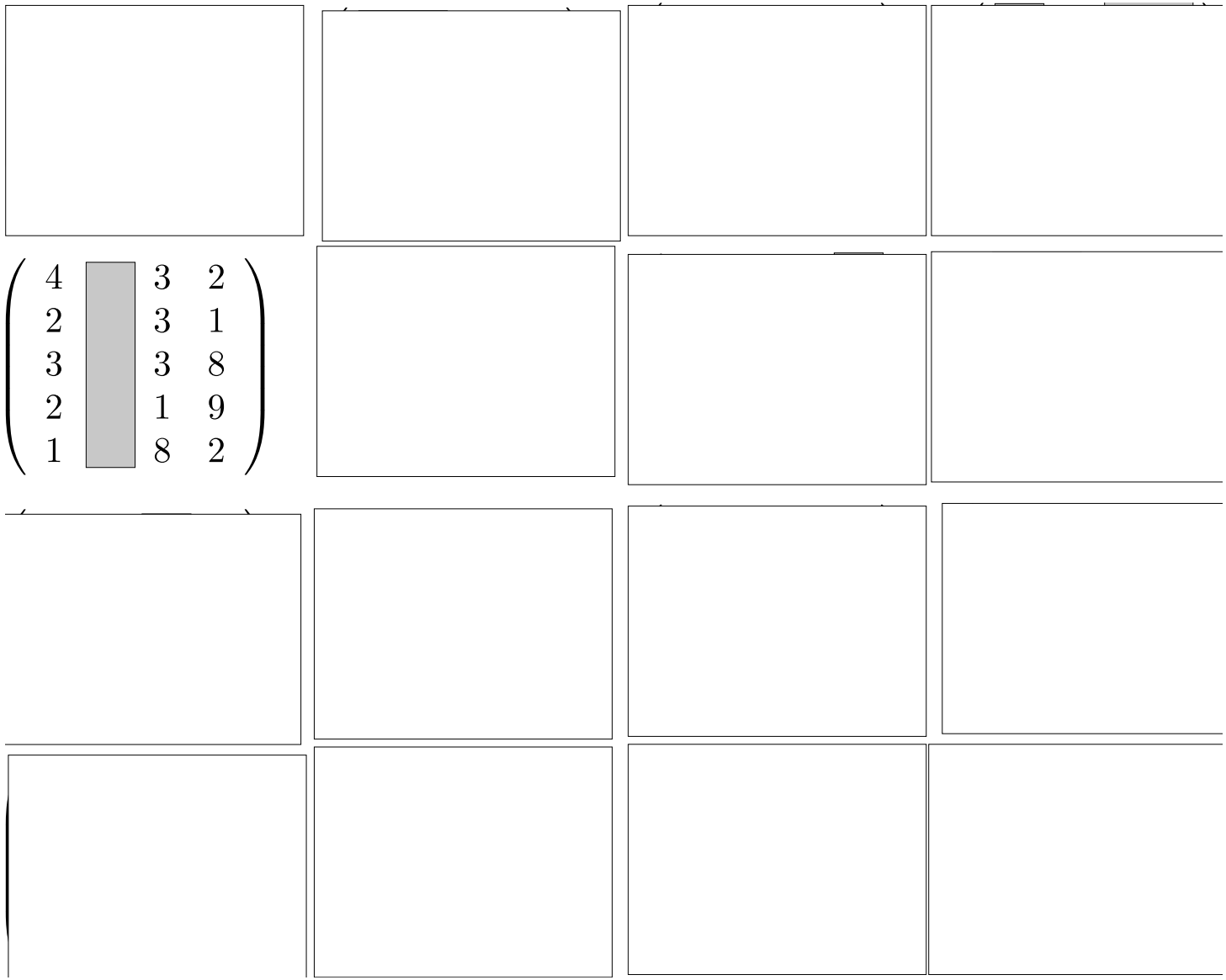
Subset meilleur qu'avec $p = 1$ colonne(s) de plus ? (non $\rightarrow$ STOP ; oui$\rightarrow$ $\mathscr{F} = \{4\}$)



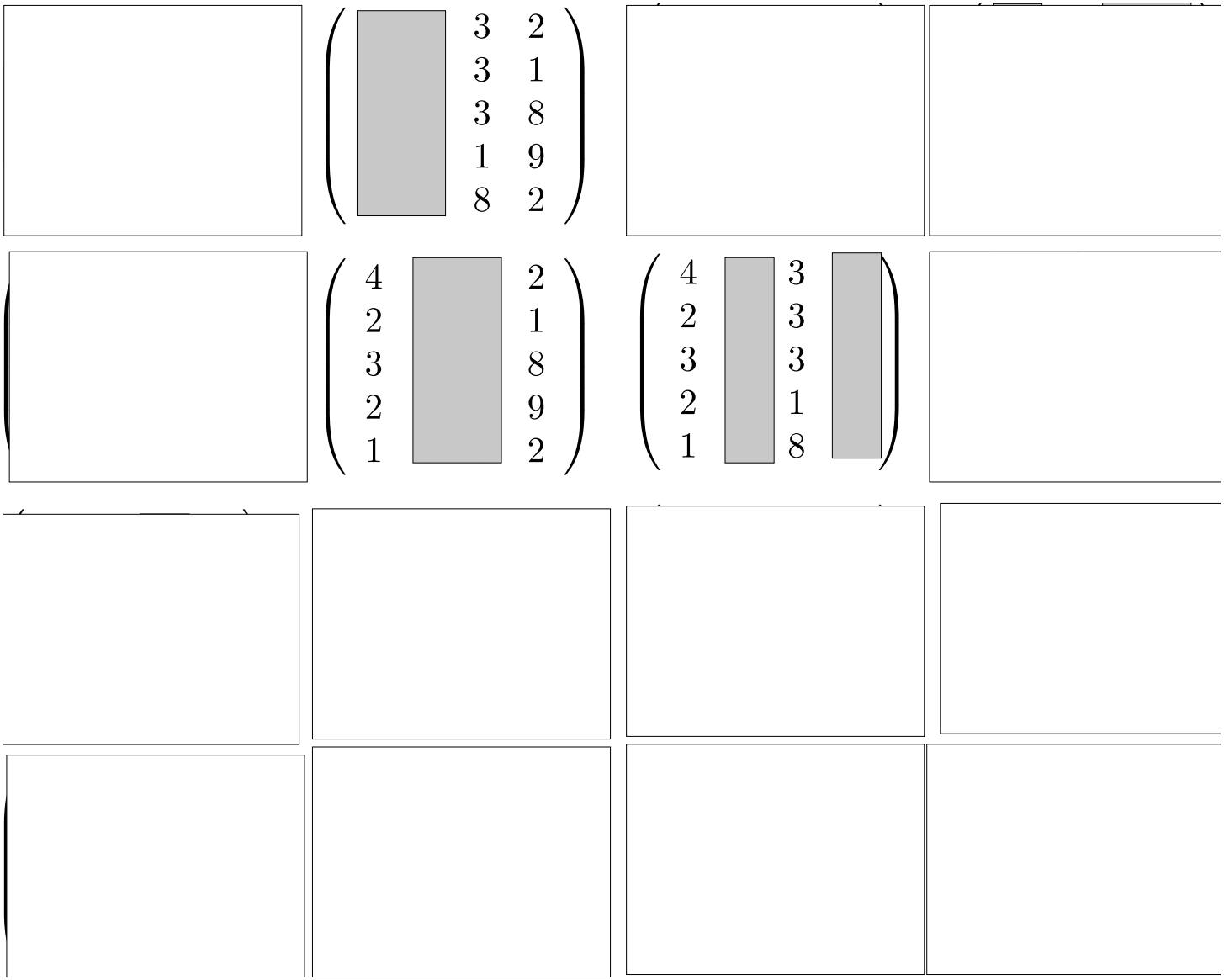
Si meilleur, trouver les $q = 2$ meilleures variable à ajouter (3 possibilités= 3 entraînements)



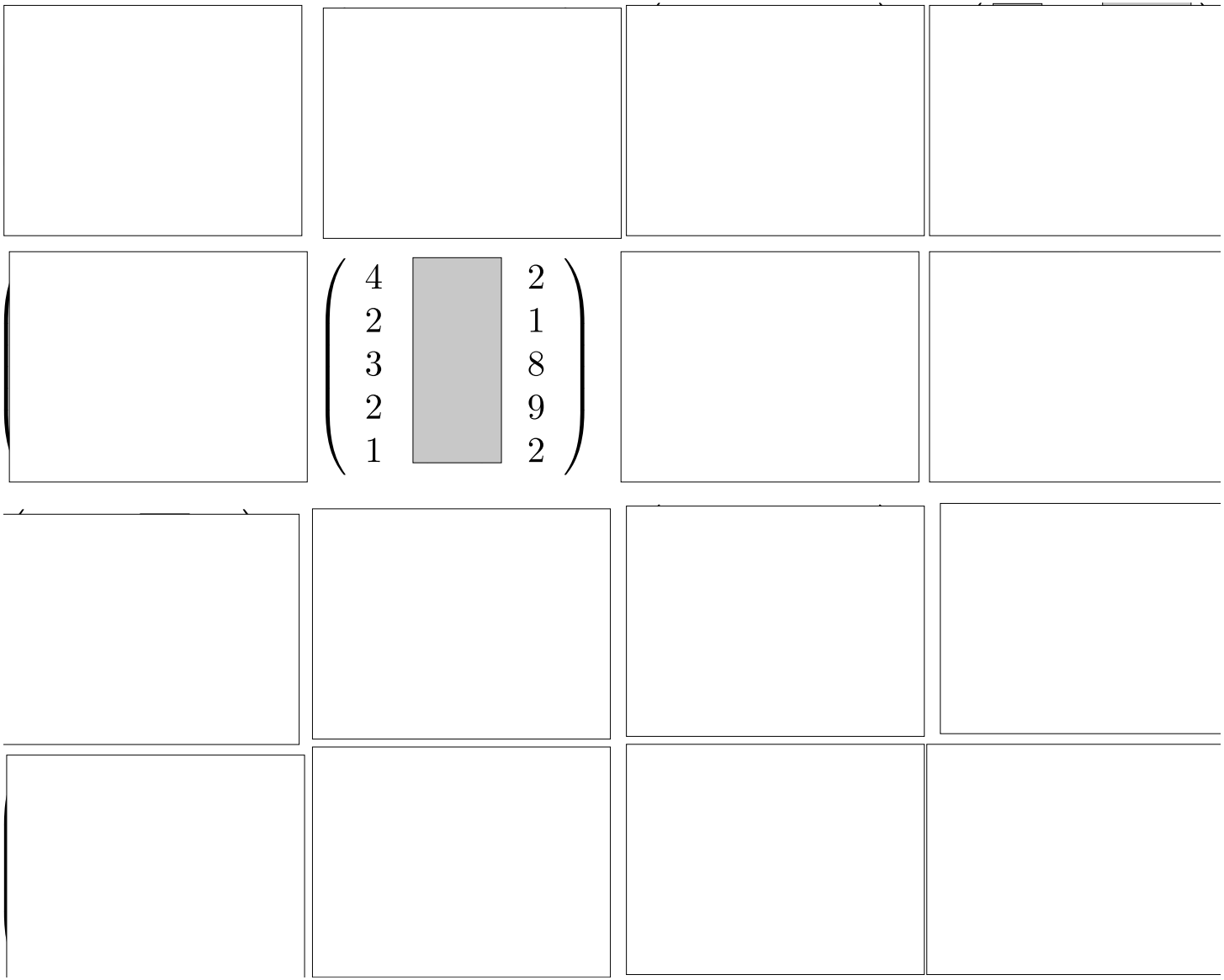
Subset choisi meilleur qu'avec une colonne de plus ? (non $\rightarrow$ STOP ; oui$\rightarrow$ $\mathscr{F} = \{1, 3, 4\}$)



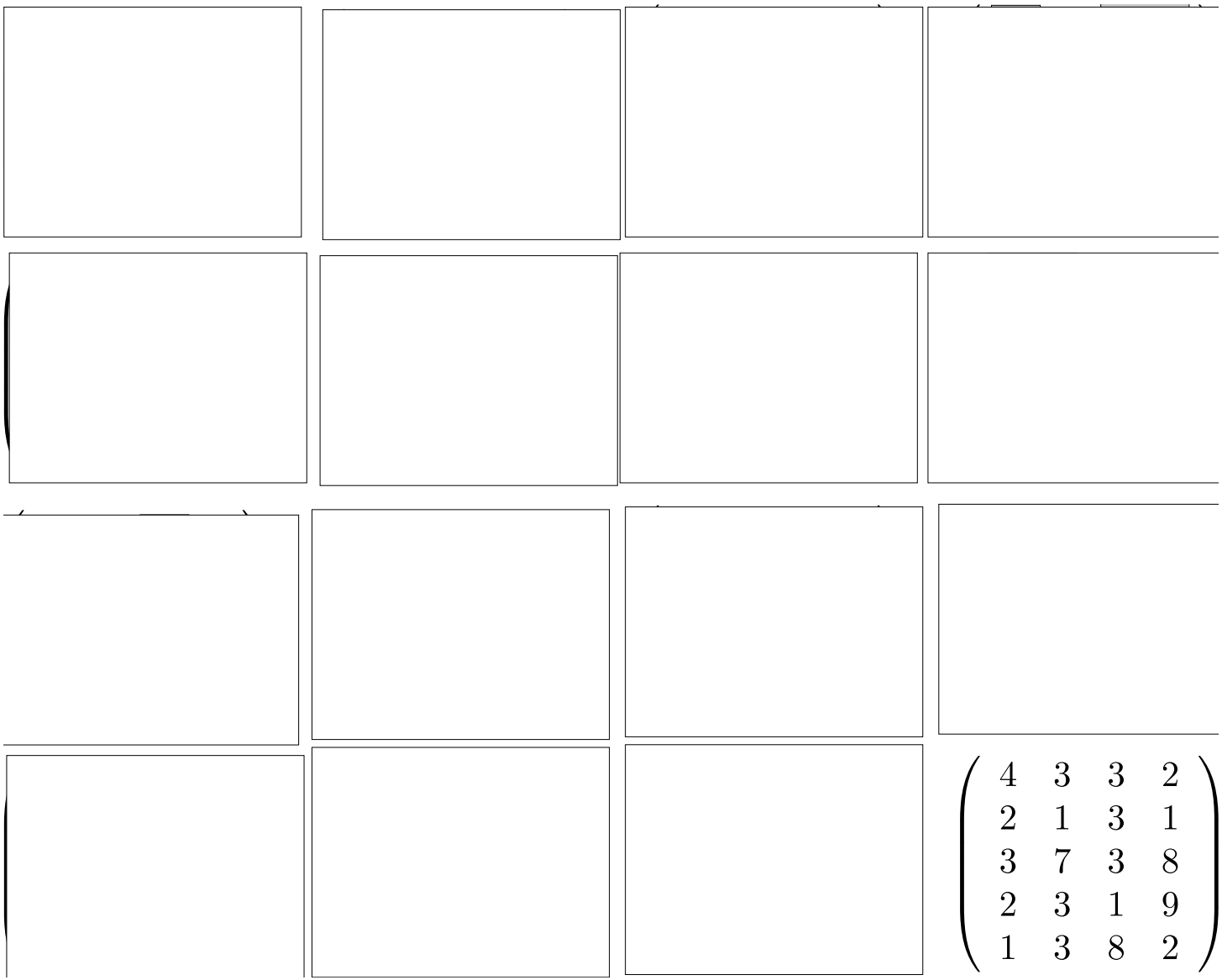
Si meilleur, trouver la ($p = 1$) meilleure variable à enlever (3 possibilités= 3 entraînements)



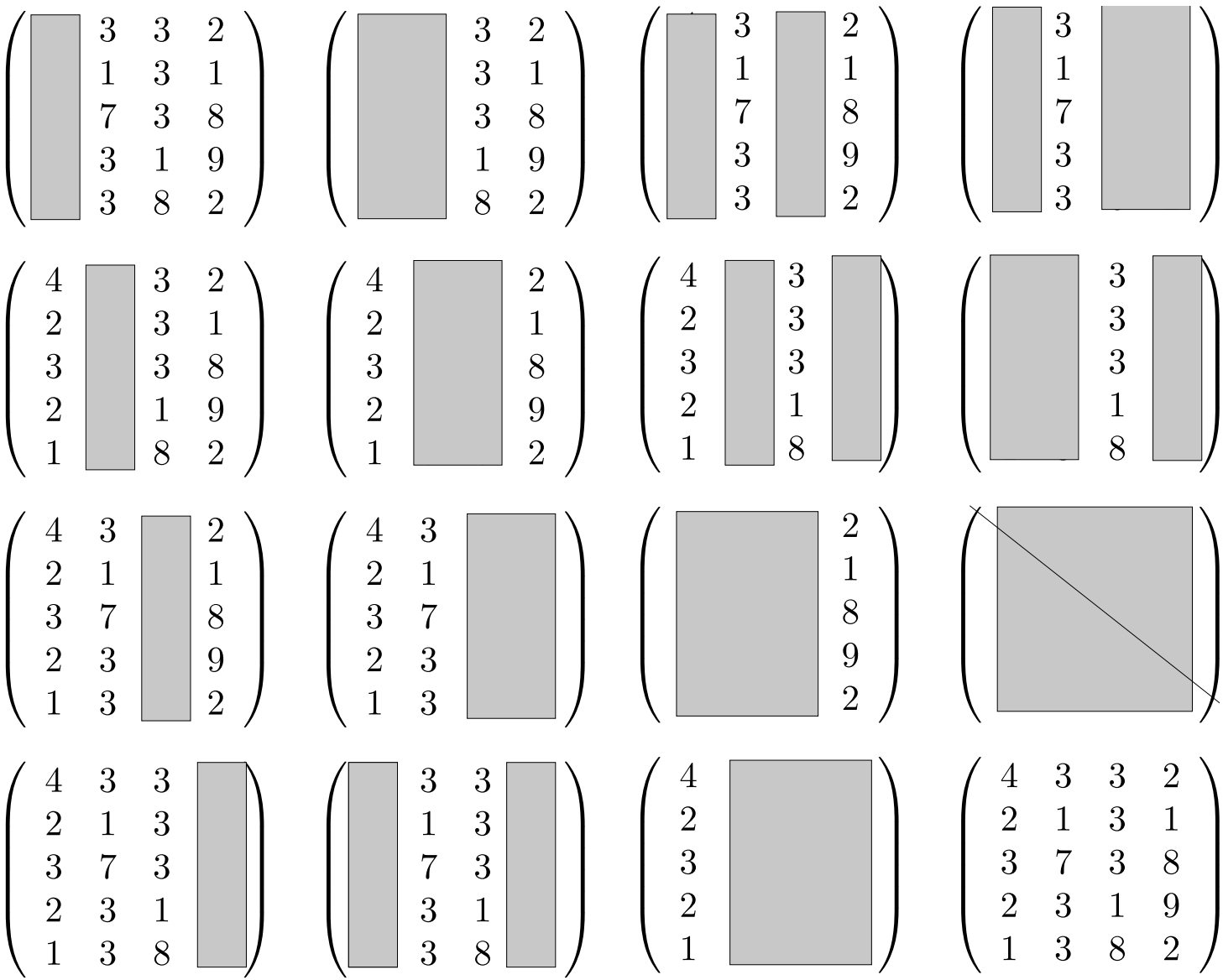
Subset choisi meilleur qu'avec une colonne de plus ? (non $\rightarrow$ STOP ; oui$\rightarrow$ $\mathscr{F} = \{1, 4\}$)



Si meilleur, ajouter $q = 2$ variables et tester si meilleur (1 possibilité= 1 entraînements)



Maximum 6+2+3+3+1 entraînements effectués



Extraction de variables : analyse en composantes principales

La méthode la plus classique pour réduire la dimension d'un jeu de données par extraction de variables est l'analyse en composantes principales, ou ACP. On parle aussi souvent de PCA, de son nom anglais Principal Component Analysis.

Idée centrale de la PCA : Représenter les données de sorte à maximiser leur variance selon les nouvelles dimensions.

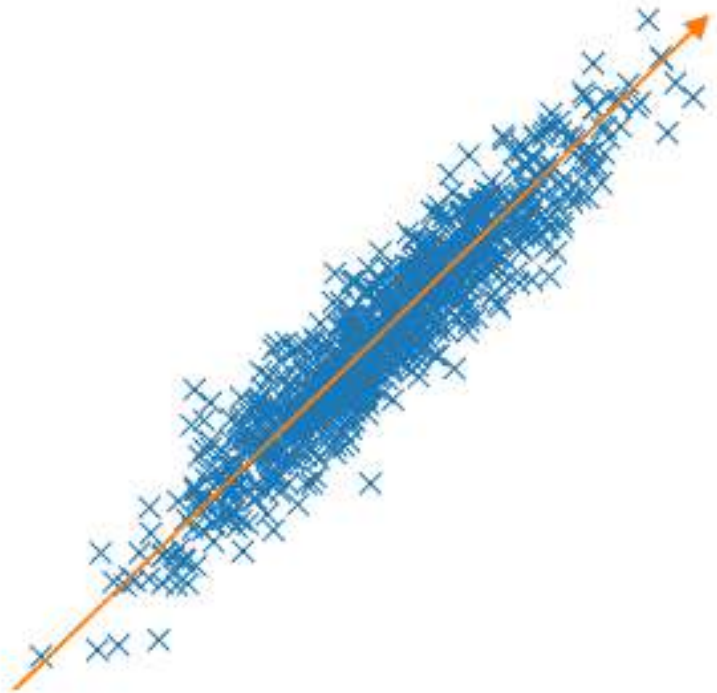


FIGURE 11.3 – La variance des données est maximale selon l'axe indiqué par une flèche. Azencott

Analyse en composantes principales

Formellement, une nouvelle représentation de $\mathcal{X}$ est définie par une base orthonormée sur laquelle projeter la matrice de données X.

Définition 11.5 (Analyse en composantes principales) Une analyse en composantes principales, ou ACP, de la matrice $X \in \mathbb{R}^{n \times p}$ est une transformation linéaire orthogonale qui permet d'exprimer X dans une nouvelle base orthonormée, de sorte que la plus grande variance de X par projection s'aligne sur le premier axe de cette nouvelle base, la seconde plus grande variance sur le deuxième axe, et ainsi de suite.

Les axes de cette nouvelle base sont appelés les composantes principales, abrégées en PC pour Principal Components.

Note sur la normalisation

Centrage On dit que X est *centrée* si chacune de ses colonnes a pour moyenne 0. Pour la suite, nous supposons que les variables ont été centrées de sorte à toutes avoir une moyenne de 0 :

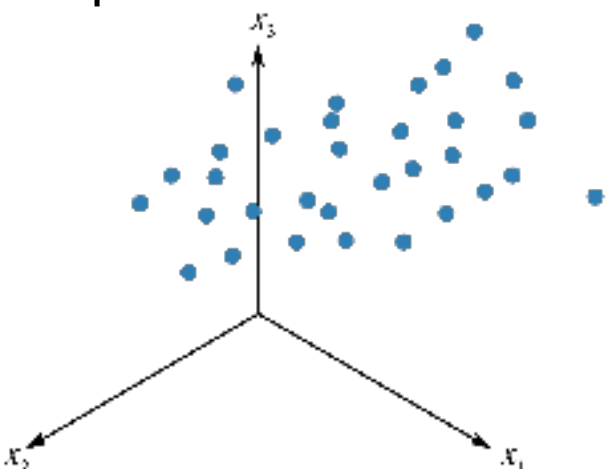
$$x_j^i \leftarrow x_j^i - \bar{x}_j \quad \text{avec} \quad \bar{x}_j = \frac{1}{n}\sum_{\ell=1}^{n} x_j^\ell.$$

Standardisation Travailler avec des variables qui prennent des valeurs dans une gamme comparable (ordre de grandeur similaire) est souvent désirable pour l'application de l'ACP (mais pas requis, contrairement au centrage, qui l'est !). On standardise alors les variables en les centrant et en imposant une variance de 1 pour éviter que les variables qui prennent de grandes valeurs aient plus d'importance que celles qui prennent de faibles valeurs :

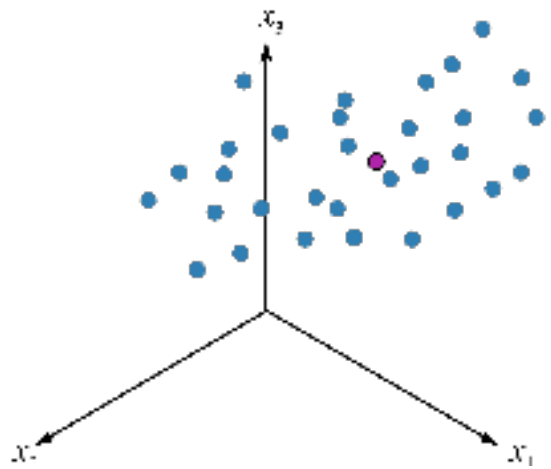
$$x_j^i \leftarrow \frac{x_j^i - \bar{x}_j}{\sqrt{\frac{1}{n}\sum_{\ell=1}^{n}\left(x_j^\ell - \bar{x}_j\right)^2}}$$

Interprétation géométrique de la PCA (centrage)

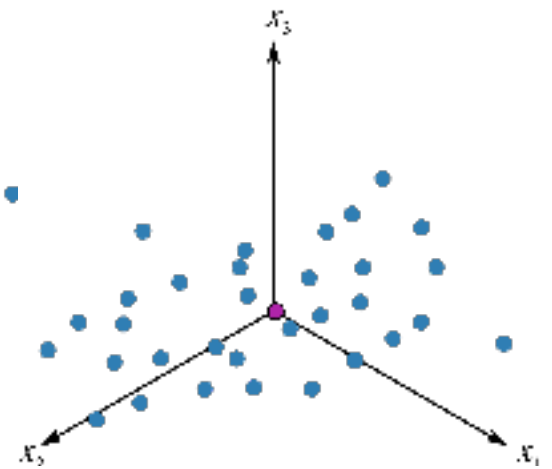
Si on suppose que l'on a $p = 3$ dimensions et $n = 32$ échantillons, on peut représenter les données dans la matrice de donnée X dans l'espace $\mathbb{R}^3$:



Données brutes



données et moyenne



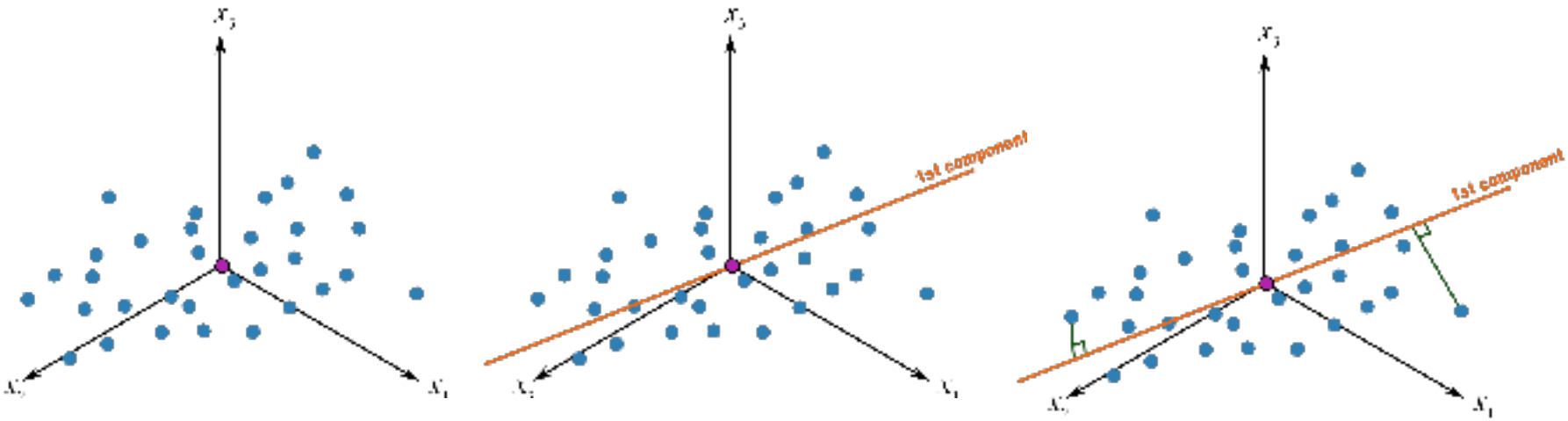
données centrées

learnche.org/pid/latent-variable-modelling/principal-component-analysis/geometric-explanation-of-pca

Interprétation géométrique de la PCA (calcul de la première composante)

La première composante de la PCA (un vecteur) indique une direction qui satisfait (de manière équivalente) :

- la somme des carrés de la distance des données à la droite définie par la composante est **minimale**
- si l'on projette les données sur cette droite, puis qu'on calcule la variance (1 dimension), celle-ci sera **maximale**

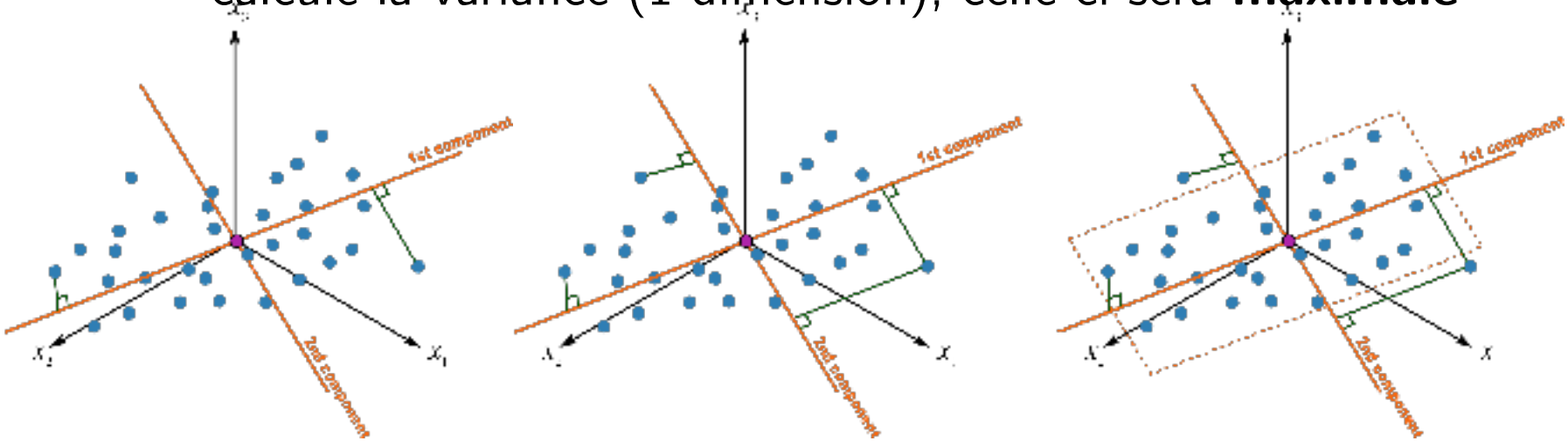


learnche.org/pid/latent-variable-modelling/principal-component-analysis/geometric-explanation-of-pca

Interprétation géométrique de la PCA (calcul de la seconde composante)

La seconde composante de la PCA (un vecteur) satisfait :

1. la direction est perpendiculaire à la première composante et
2. en plus, la direction est telle que les deux conditions sont satisfaites (de manière équivalente) :
   - la somme des carrés de la distance des données à la droite définie par la composante est **minimale**
   - si l'on projette les données sur cette seconde droite, puis qu'on calcule la variance (1 dimension), celle-ci sera **maximale**



learnche.org/pid/latent-variable-modelling/principal-component-analysis/geometric-explanation-of-pca

Calcul des composantes principales

Théorème 11.1 Soit $X \in \mathbb{R}^{n \times p}$ une matrice centrée, avec matrice de covariance $\Sigma = \frac{1}{n} X^\top X \in \mathbb{R}^{p \times p}$. Les composantes principales de X sont les vecteurs propres de Σ, ordonnés par valeur propre décroissante. ■

Interprétation :

$$\underset{p \times p}{\Sigma} = (1/n) \underset{p \times n}{X^\top} \underset{n \times p}{X}$$

Les vecteurs propres v_i satisfont

$$\underset{p \times p}{\Sigma} \underset{p \times 1}{v_i} = \underset{p \times 1}{v_i} \underset{1 \times 1}{\alpha_i}$$

Avec $D = \text{diag}(\alpha_1, \alpha_2, \ldots, \alpha_p)$ et $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_p$, on a :

$$\underset{p \times p}{\Sigma} \underset{p \times p}{V} = \underset{p \times p}{V} \underset{p \times p}{D}$$

où les colonnes de V sont les vecteurs propres v_i, $i = 1, \ldots, p$.

Note : le calcul de $X^\top X$ est, en général, à éviter (coûteux). On utilisera plutôt l'approche par SVD.

Décomposition en valeurs singulières (singular value decomposition, SVD)

Théorème 11.2 Soit $X \in \mathbb{R}^{n \times p}$ une matrice centrée. Les composantes principales de X sont ses vecteurs singuliers à droite ordonnés par valeur singulière décroissante. ■

Démonstration Si l'on écrit X sous la forme $X = UDV^\top$ où $U \in \mathbb{R}^{n \times n}$ et $V \in \mathbb{R}^{p \times p}$ sont orthogonales et $D \in \mathbb{R}^{n \times p}$ est diagonale, alors

$$\Sigma = \frac{1}{n} X^\top X = \frac{1}{n} V D^\top U^\top U D V^\top = V \frac{D^2}{n} V^\top$$

et les valeurs singulières de X (les entrées de D) sont les racines carrées des valeurs propres de Σ après qu'on les multiplie par n, tandis que les vecteurs singuliers à droite de X (les colonnes de V) sont les vecteurs propres de Σ.

Note : les implémentations de la décomposition en valeurs singulières (ou SVD) sont numériquement plus stables que celles de décomposition spectrale. On préfèrera donc calculer les composantes principales de X en calculant la SVD de X plutôt que la décomposition spectrale de $X^\top X$.

Représentation réduite des données (notation alternative à celle livre)

Soit le jeu de données $X \in \mathbb{R}^{n \times p}$.
On calcule les composantes principales (CP) soit par :

vecteurs propres de Σ (=colonnes de V = CP) : $\underset{p \times p}{\Sigma} \; \underset{p \times p}{V} = \underset{p \times p}{V} \; \underset{p \times p}{D}$

ou par SVD (CP=colonnes de V) : $\underset{n \times p}{X} = \underset{n \times n}{U} \; \underset{n \times p}{D} \; \underset{p \times p}{V^\top}$

Soit $1 \leq m \leq p$ le nombre choisi de composantes principales et la matrice $W \in \mathbb{R}^{p \times m}$ est obtenue en prenant les m premières colonnes de $V \in \mathbb{R}^{p \times p}$.
La *représentation réduite* $\tilde{H} \in \mathbb{R}^{n \times m}$ des n observations dans le nouvel espace de dimension m s'obtient en projetant X sur les colonnes de W, autrement dit en calculant

$$\underset{n \times m}{\tilde{H}} = \underset{n \times p}{X} \; \underset{p \times m}{W}$$

La matrice $\tilde{H}$ peut être interprétée comme une représentation latente (ou cachée, "hidden" $\Rightarrow$notation $\tilde{H}$) des données.

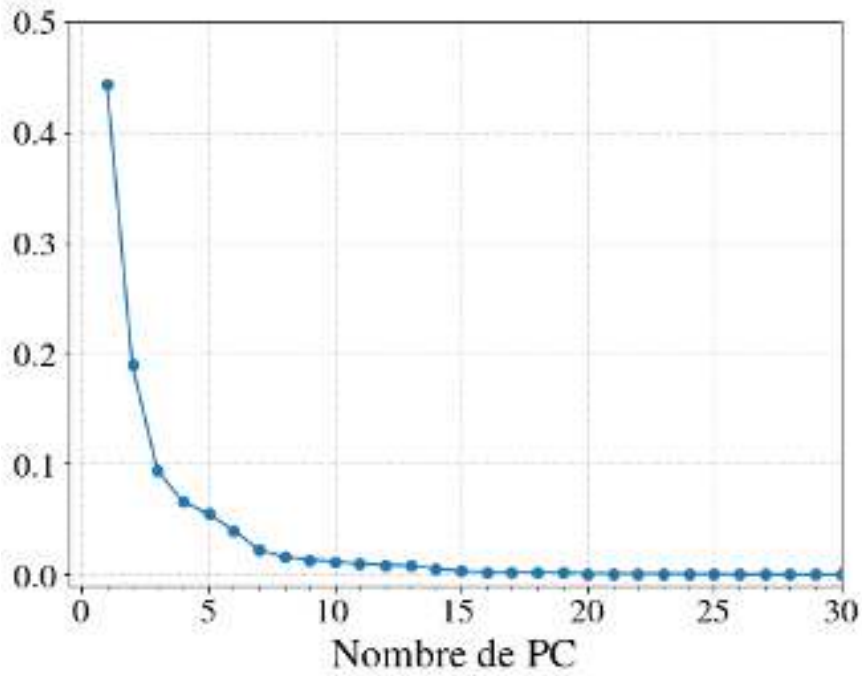
Choix du nombre de composantes principales

Réduire la dimension des données par une ACP implique de choisir un nombre de composantes principales à conserver. Pour ce faire, on utilise la proportion de variance expliquée par ces composantes : la variance de X s'exprime comme la trace de Σ, qui est elle-même la somme de ses valeurs propres. Ainsi, si l'on décide de conserver les m premières composantes principales de X, la proportion de variance qu'elles expliquent est :

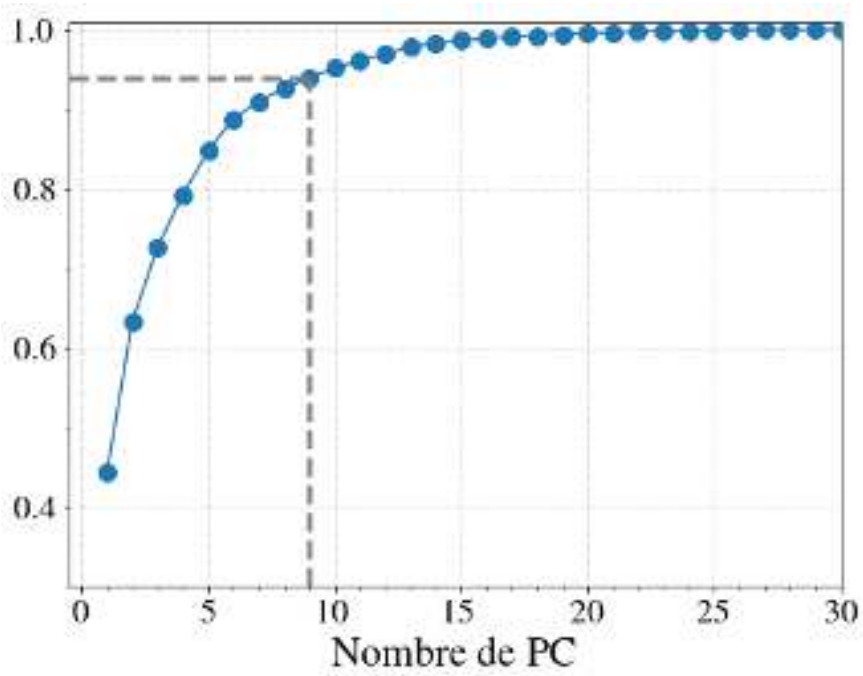
$$\frac{\alpha_1 + \alpha_2 + \cdots + \alpha_m}{\mathrm{Tr}\,(\Sigma)}$$

où $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_p$ sont les valeurs propres de Σ par ordre décroissant.

Évolution de la proportion de variance expliquée par nombre de composantes principales



(A) Pourcentage de variance expliqué par chacune des composantes principales. À partir de 6 composantes principales, ajouter de nouvelles composantes n'est plus vraiment informatif.



(B) Pourcentage cumulé de variance expliquée par chacune des composantes principales. Si on se fixe une proportion de variance expliquée de 95%, on peut se contenter de 10 composantes principales.

FIGURE 11.4 – Pour choisir le nombre de composantes principales, on utilise le pourcentage de variance expliquée.

Azencott

Résumé

- Réduire la dimension des données avant d'utiliser un algorithme d'apprentissage supervisé permet d'améliorer ses besoins en temps et en espace, mais aussi ses performances.
- On distingue la sélection de variables, qui consiste à éliminer des variables redondantes ou peu informatives, de l'extraction de variable, qui consiste à générer une nouvelle représentation des donnée
- Pour éviter le sur-apprentissage, il est essentiel lors de l'étape de sélection du modèle de valider les différents modèles testés sur un jeu de données (jeu de validation) différent de celui utilisé pour l'entraînement.
- Pour estimer la performance en généralisation d'un modèle, il est essentiel de l'évaluer sur des données (jeu de test) qui n'ont été utilisées ni pour l'entraînement, ni pour la sélection de ce modèle.
- De nombreuses méthodes permettent de réduire la dimension des variables

Guide de lecture pour ce cours

Chloé-Agathe Azencott “Introduction au Machine Learning”,
Dunod, 2019, ISBN 978-210-080153-4
Chapitre 11 : Réduction de dimension
Chapitre 3 : Sélection de modèle et validation