DÉTECTEZ DES FAUX BILLETS

Frédéric Gainza - Mai 2021

RAPPEL DE LA MISSION

Commanditaire :

Office central pour la répression du faux monnayage

Objectif :

Créer un algorithme de détection de faux billets

Données à disposition

Jeu de données comprenant des caractéristiques géométriques de billets

SOMMAIRE

PARTIE 1

ANALYSE DES DONNÉES

A/ Analyse du fichier notes.csv

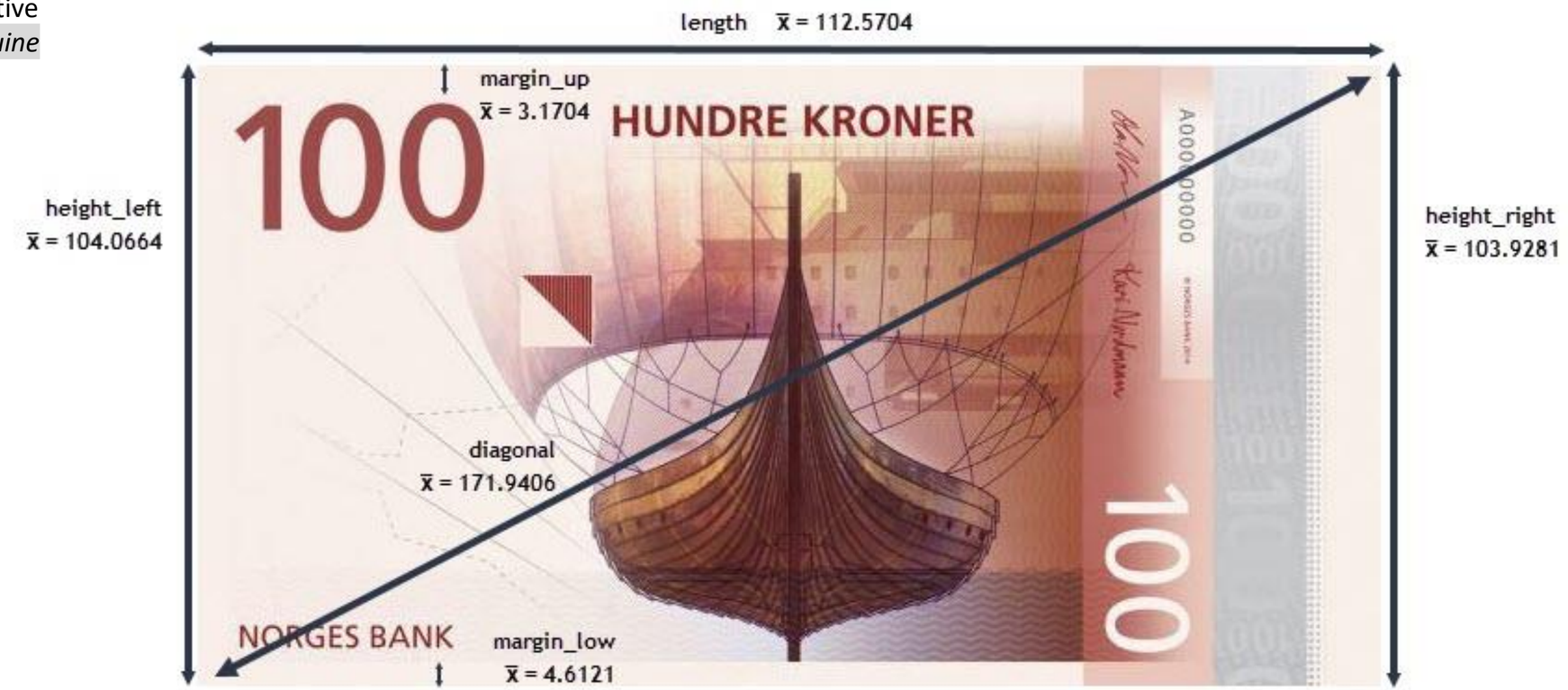
170 observations

- 'is_genuine' = True — Vrais Billets — n = 100 (58,82%)
- 'is_genuine = False — Faux Billets — n = 70 (41,18%)

7 variables

- 1 var. binaire qualitative *is_genuine*
- 6 var. quantitatives :

diagonal	float64
height_left	float64
height_right	float64
margin_low	float64
margin_up	float64
length	float64



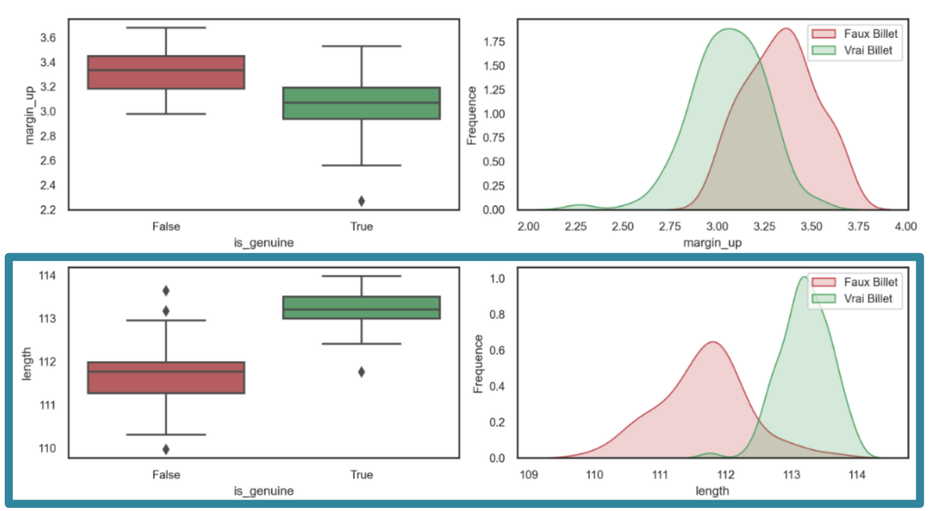
Absence de valeurs nulles et de doublons

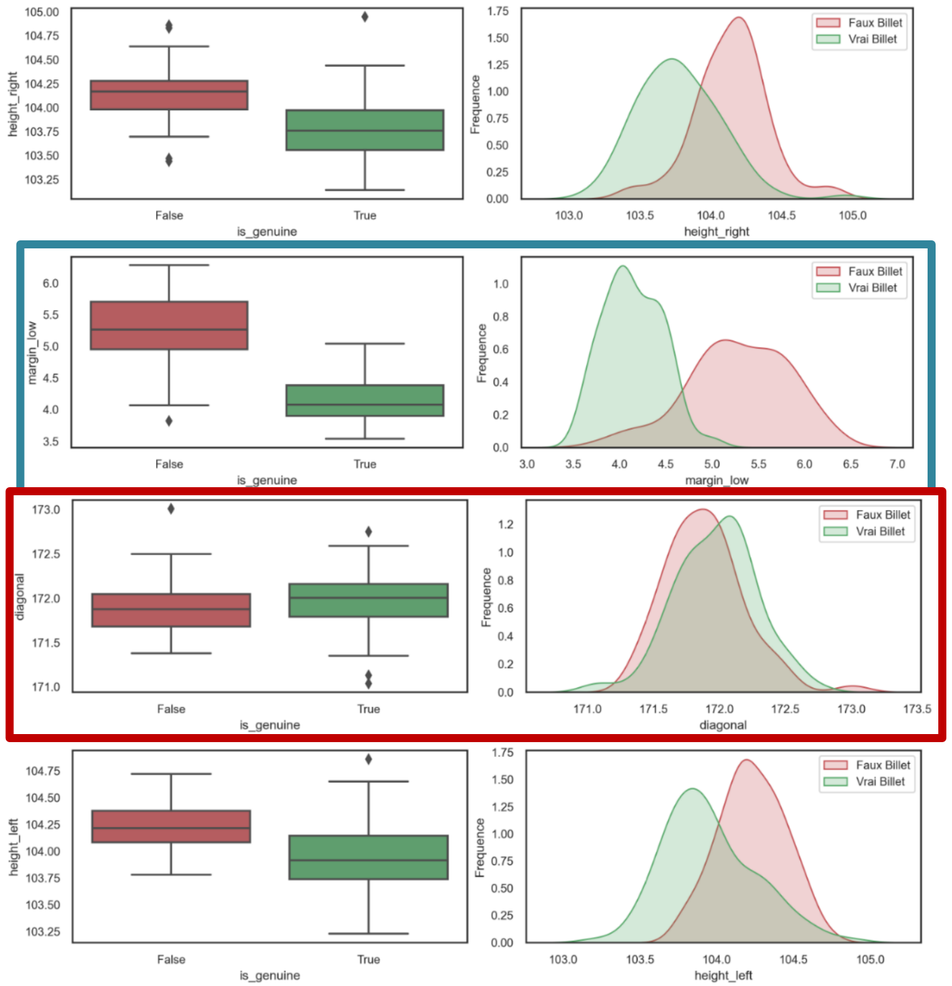
```
Entrée [11]:  1 data.isna().sum()
              2 data.duplicated().sum()

Out[11]: is_genuine      0
         diagonal        0
         height_left     0
         height_right    0
         margin_low      0
         margin_up       0
         length          0
         dtype: int64

Out[11]: 0
```

B/ Analyses univariées

Obj : différencier vrais et faux billets → Etudier pour chaque variable les 2 groupes 'is_genuine' = True et 'is_genuine' = False → Recherche des variables pour lesquelles il existe des différences entre ces 2 groupes

1. Analyse graphique





- Les 2 variables dont la distribution est la plus différente selon le groupe "vrai billet" / "faux billet sont : *length* et *margin_low*
- Une variable semble avoir la même distribution pour les 2 groupes : *diagonal*

B/ Analyses univariées

2. Test de comparaison

Principe :

↳ Pour chacune des 6 variables quantitatives, on teste les 2 groupes "is_genuine" = True et "is_genuine" = False pour vérifier s'ils sont significativement différents :

- Vérification de la normalité des distributions
- Test d'égalité des variances, puis test d'égalité des moyennes :
  - Si les moyennes sont différentes, alors les 2 groupes sont différents
    → la distribution est différente entre groupe de vrais billets et groupe de faux billets
  - Si les moyennes sont égales, alors les 2 groupes ne sont pas différents
    → même distribution pour les 2 groupes

Exemple : Test d'égalité des moyennes

Hypothèse nulle H0 : "Les moyennes des 2 groupes sont égales"
-> Nos 2 groupes ne sont donc pas significativement différents

Hypoyhèse alternative H1 : "Les moyennes des 2 groupes ne sont pas égales"
-> Nos 2 groupes sont donc significativement différents

si p <= α : on rejette H0 au profit de l'alternative H1 => on rejette l'hypothèse d'égalité des moyennes
si p > α : on ne peut pas rejeter H0 => on valide l'hypothèse d'égalité des moyennes

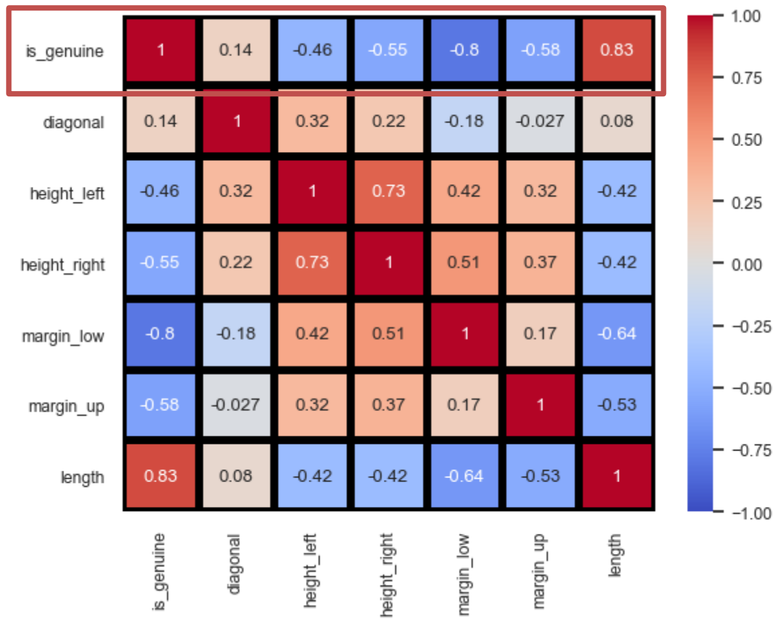
Résultat :

- Une seule variable pour laquelle on ne peut pas rejeter l'hypothèse HO d'égalité des moyennes : *diagonal*

  ↳ Il n'y a donc pas de différences entre les groupes "vrais billets" et "faux billets" pour la variable *diagonal*

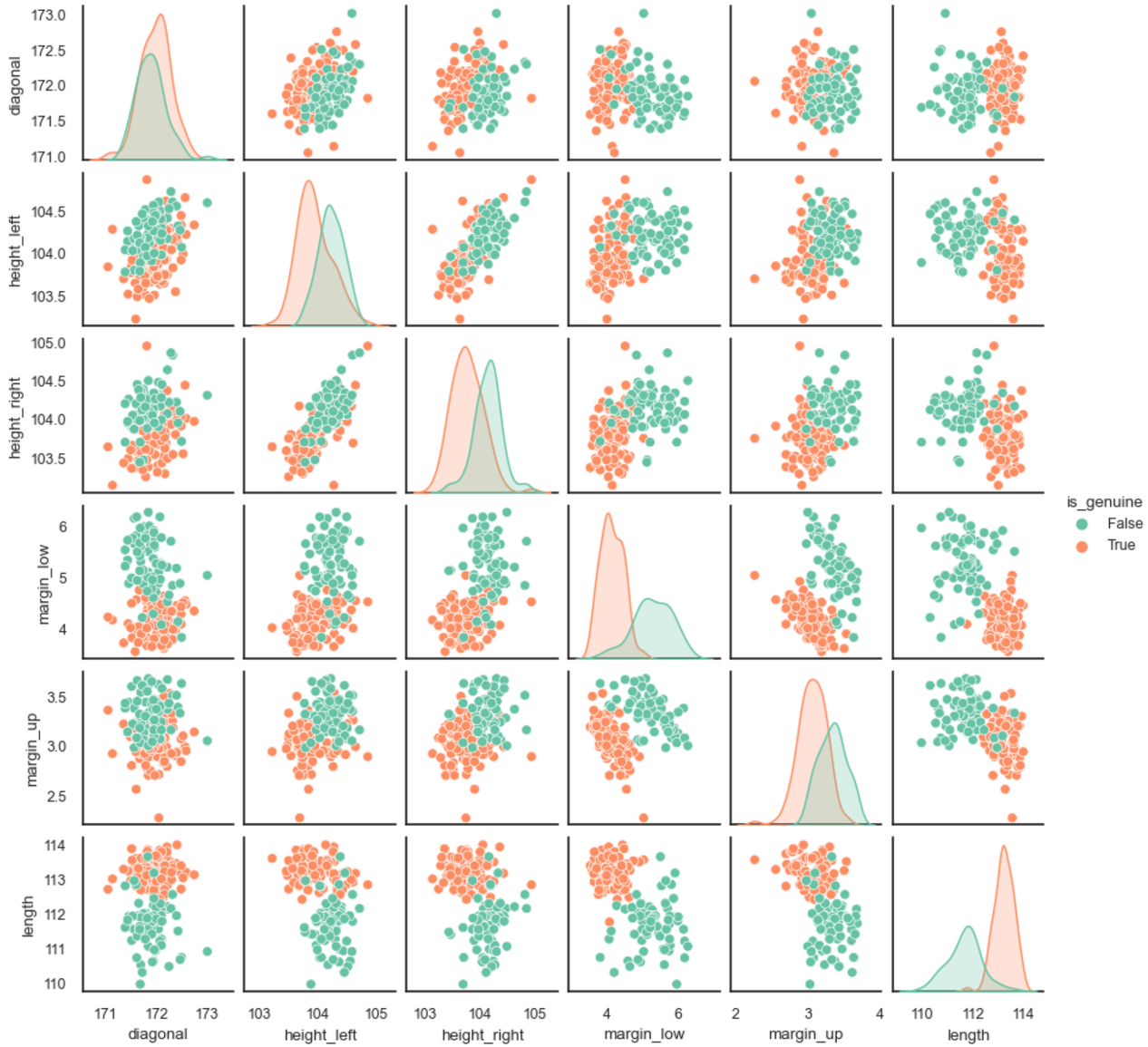
C/ Analyses bivariées

Analyse des corrélations



Corrélations entre *is_genuine* et les variables quantitatives confirment les résultats précédents :

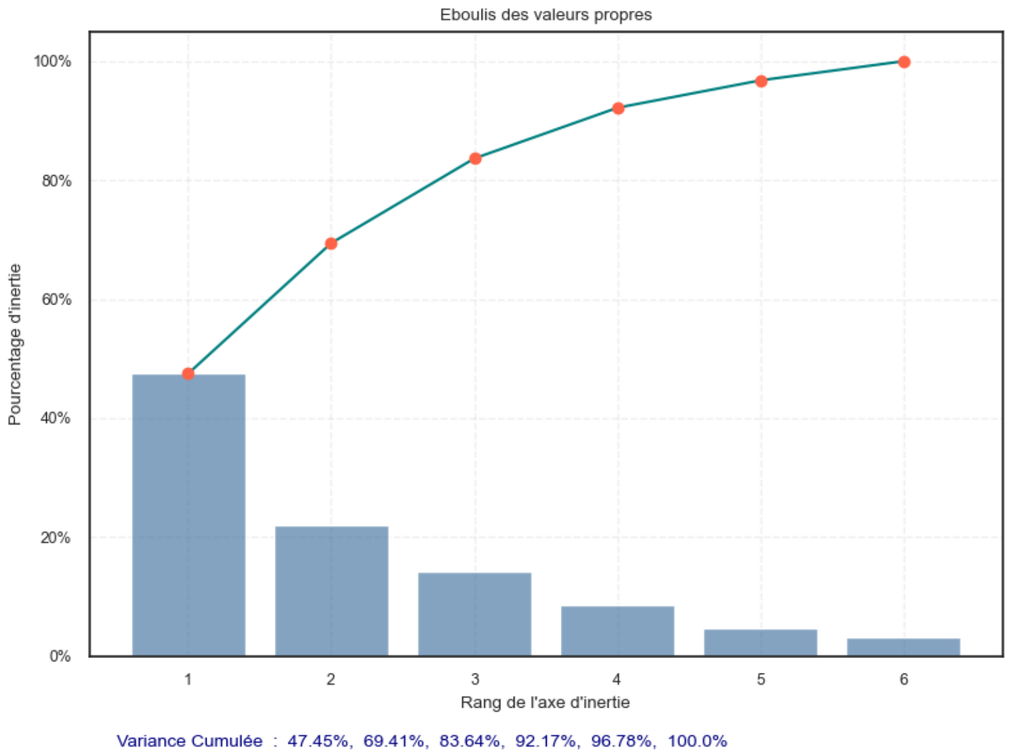
- faible corrélation avec *diagonal*
- fortes corrélations avec *length* et *margin_low*



PARTIE 2

ANALYSE EN COMPOSANTES PRINCIPALES

A/ Eboulis des valeurs propres



✓ Diagramme qui permet de déterminer graphiquement le nombre de composantes à retenir

✓ Il indique le pourcentage d'inertie associé à chaque axe

Cf. la part de l'information initiale captée par chacune des composantes principales

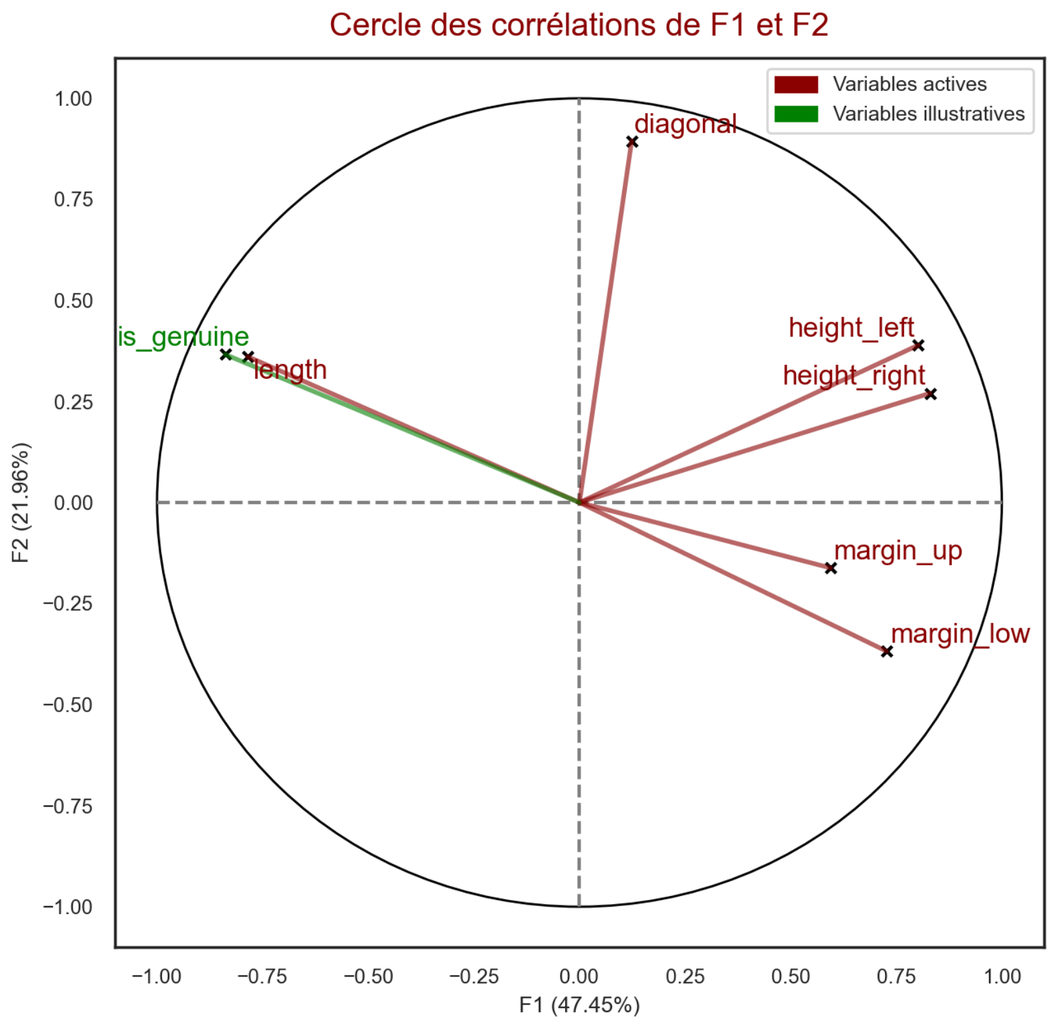
➔ ici, en se limitant au 1er plan factoriel (donc aux 2 premiers axes), on récupère 70% de l'information initiale

✓ Méthodes pour choisir le nombre de composantes à retenir :

- méthode du coude
- critère de Kaiser
  soit p le nombre total de dimensions
  alors on considère comme non importantes les dimensions avec une inertie inférieure à (100/p)%, soit ici **100/6 = 16,67%**

➔ on se limite donc aux 2 premières dimensions

B/ Représentation des variables par le cercle des corrélations



1. Qualité de représentation des variables

- Graphiquement :

→ une variable est d'autant mieux représentée que l'extrémité du vecteur qui la représente est proche du cercle de corrélations

- Par le calcul :

→ la qualité de représentation dépend du COS ² (COS² obtenus à partir de la matrice de corrélations des variables avec les axes factoriels)

→ on additionne le COS² des facteurs 1 et 2

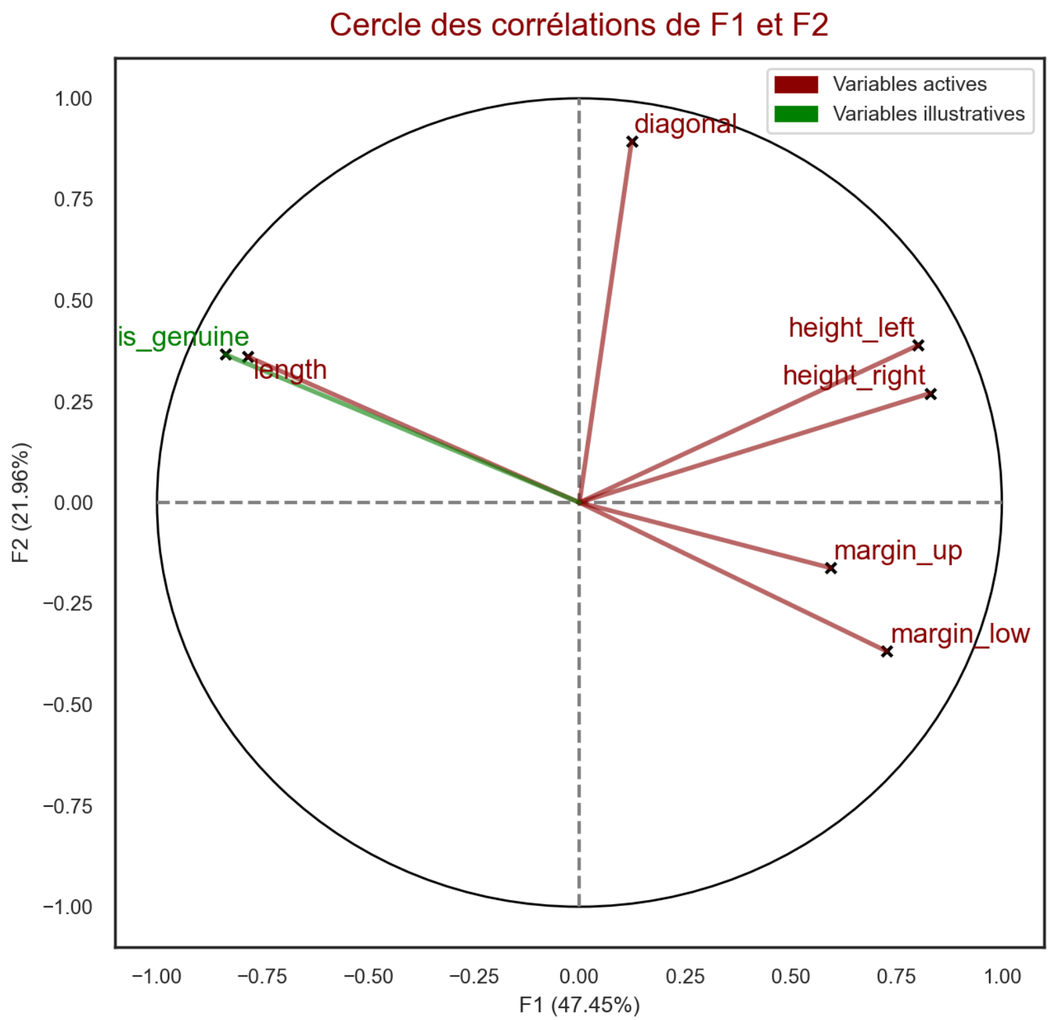
↳ plus la valeur est proche de 1, meilleure est la représentation de la variable sur le 1er plan factoriel

	id	COS²_var_F1	COS²_var_F2
0	diagonal	0.0153	0.8008
1	height_left	0.6437	0.1516
2	height_right	0.6886	0.0731
3	margin_low	0.5289	0.1354
4	margin_up	0.3538	0.0262
5	length	0.6166	+ 0.1303

	id	qualite_1er_plan
0	diagonal	0.8161
1	height_left	0.7953
2	height_right	0.7617
3	margin_low	0.6643
4	margin_up	0.3800
5	length	0.7469

➔ Les variables sont bien représentées à l'exception de *margin_up*

B/ Représentation des variables par le cercle des corrélations



2. Caractérisation des axes factoriels

- la **projection** sur l'axe factoriel de l'**extrémité de la flèche** représentant une variable correspond au **coefficient de corrélation** entre la variable et l'axe factoriel

- Graphiquement :

Axe 1 : 47,5% de l'info initiale

- les 3 variables *height_left*, *height_right* et *margin_low* sont fortement corrélées à l'axe 1 et de façon positive (aussi le cas pour *margin_up* mais corrélation plus faible)
- la variable *length* est fortement corrélée à l'axe 1 et de façon négative

Axe 2 : 22% de l'info initiale

- seule la variable *diagonal* est corrélée de façon significative à l'axe 2 (corrélation positive)

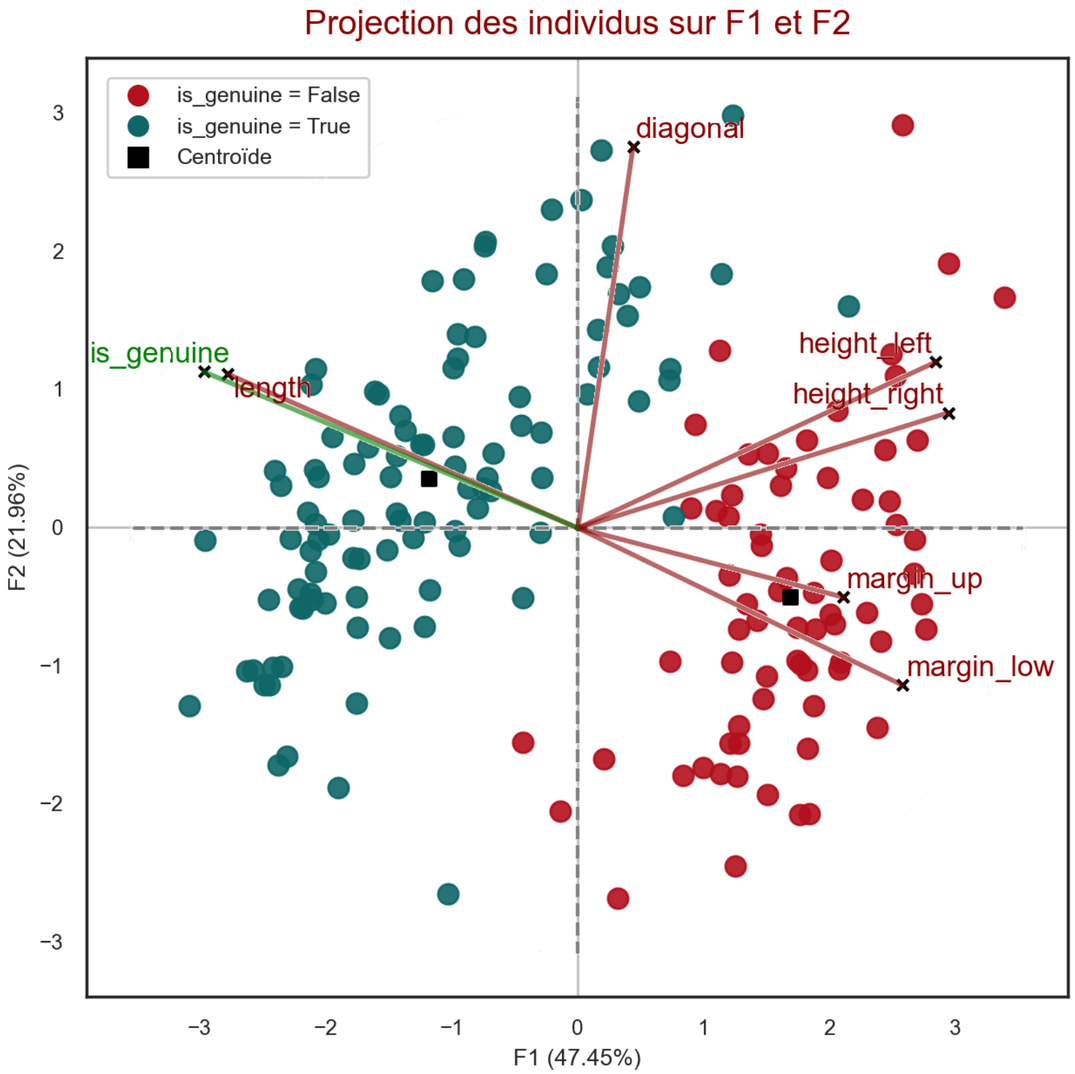
- Par le calcul :

→ cf. le calcul des corrélations par axe factoriel

	id	COR_F1	COR_F2
0	diagonal	0.1236	0.8949
1	height_left	0.8023	0.3894
2	height_right	0.8298	0.2704
3	margin_low	0.7273	-0.3679
4	margin_up	0.5948	-0.1620
5	length	-0.7852	0.3610

➔ Résultats cohérents avec ceux de l'analyse graphique

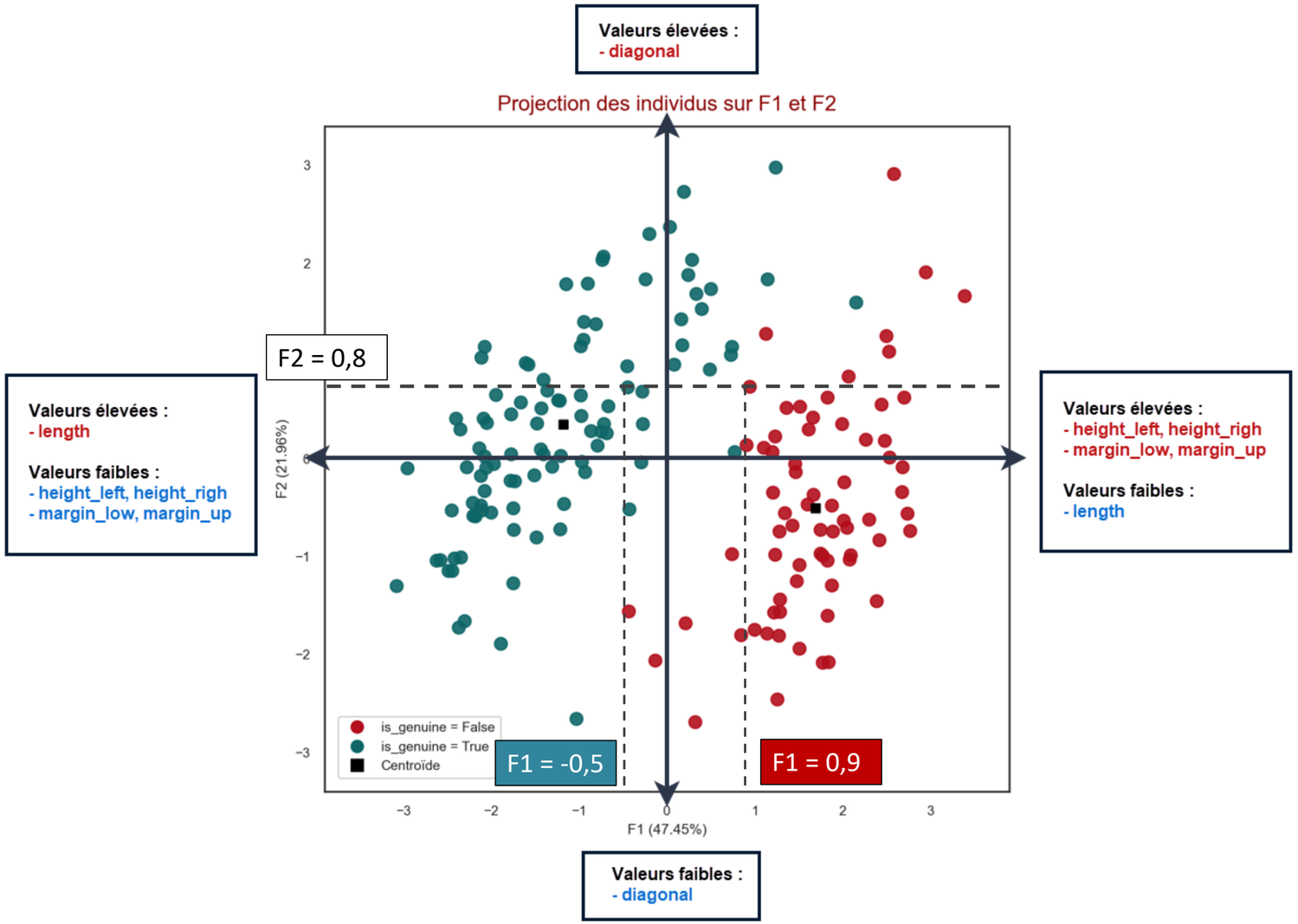
C/ Représentation des individus par les plans factoriels



- On constate que les 2 groupes "Vrais billets" et "Faux billets" sont relativement bien séparés

- Pour pouvoir analyser ce graphique de projections des individus, il faut le lire en fonction du cercle des corrélations :
  - On peut superposer les 2 graphiques
  - On peut aussi les représenter de la façon suivante :

C/ Représentation des individus par les plans factoriels



➔ on constate que les vrais billets sont différenciés des faux billets uniquement sur l'axe 1 :

- pour une même valeur de F2 on peut associer une valeur de F1 qui correspond à un vrai billet mais aussi une valeur de F1 qui correspond à un faux

- à l'inverse, dans la grande majorité des cas, à une même valeur de F1, quelle que soit la valeur de F2 associée, cela correspondra toujours au même groupe de billet

Remarque :

Résultat cohérent avec l'analyse des variables de la première partie :

- pour la variable *diagonal*, pas de différence de moyennes entre groupes "vrais billets" et "faux billets"

- faible corrélation entre la variable *diagonal* et la variable cible *is_genuine* (r=0,14)

C/ Représentation des individus par les plans factoriels

1. Qualité de représentation des individus

➔ Même logique que pour les variables

- On récupère les COS² depuis la fonction ACP

	id	F1	F2	F3	F4	F5	F6
92	92	0.8587	0.0391	0.0000	0.0234	0.0577	0.0211
72	72	0.2166	0.0067	0.0451	0.0023	0.4021	0.3272
167	167	0.4988	0.0835	0.0435	0.3430	0.0016	0.0297

- On additionne les cos² des 2 premiers facteurs

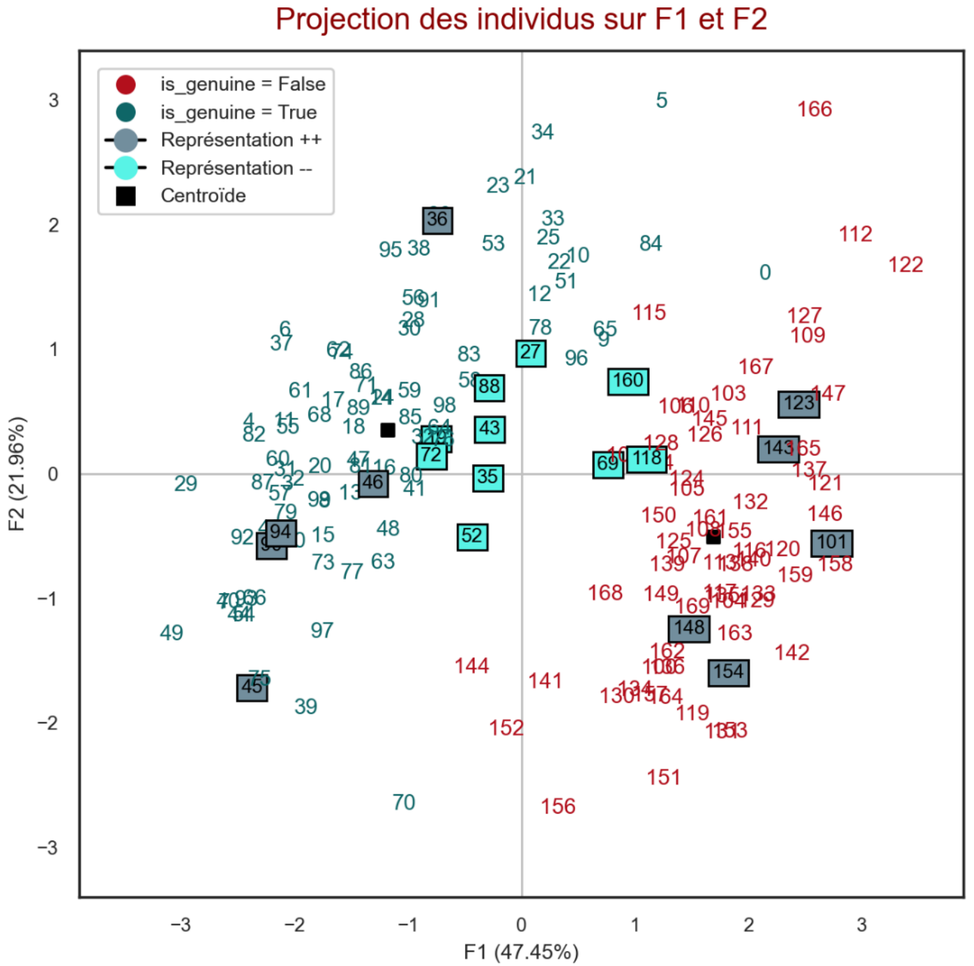
  → plus la valeur est proche de 1, meilleure est la représentation de l'individu sur le 1er plan factoriel

- On affiche les 10 individus les mieux représentés et les 10 individus les moins bien représentés

id_ind	qualite_1er_plan
148	0.9890
143	0.9811
46	0.9656
45	0.9626
154	0.9605
90	0.9596
123	0.9521
94	0.9518
101	0.9459
36	0.9443

id_ind	qualite_1er_plan
35	0.0251
43	0.1021
160	0.1116
88	0.1591
52	0.1828
72	0.2233
118	0.2320
27	0.2365
69	0.2416
19	0.2714

On affiche ces individus sur le graphique de projection des individus



Les individus les moins bien représentés sont ceux qui sont **"à la frontière" des 2 groupes**

C/ Représentation des individus par les plans factoriels

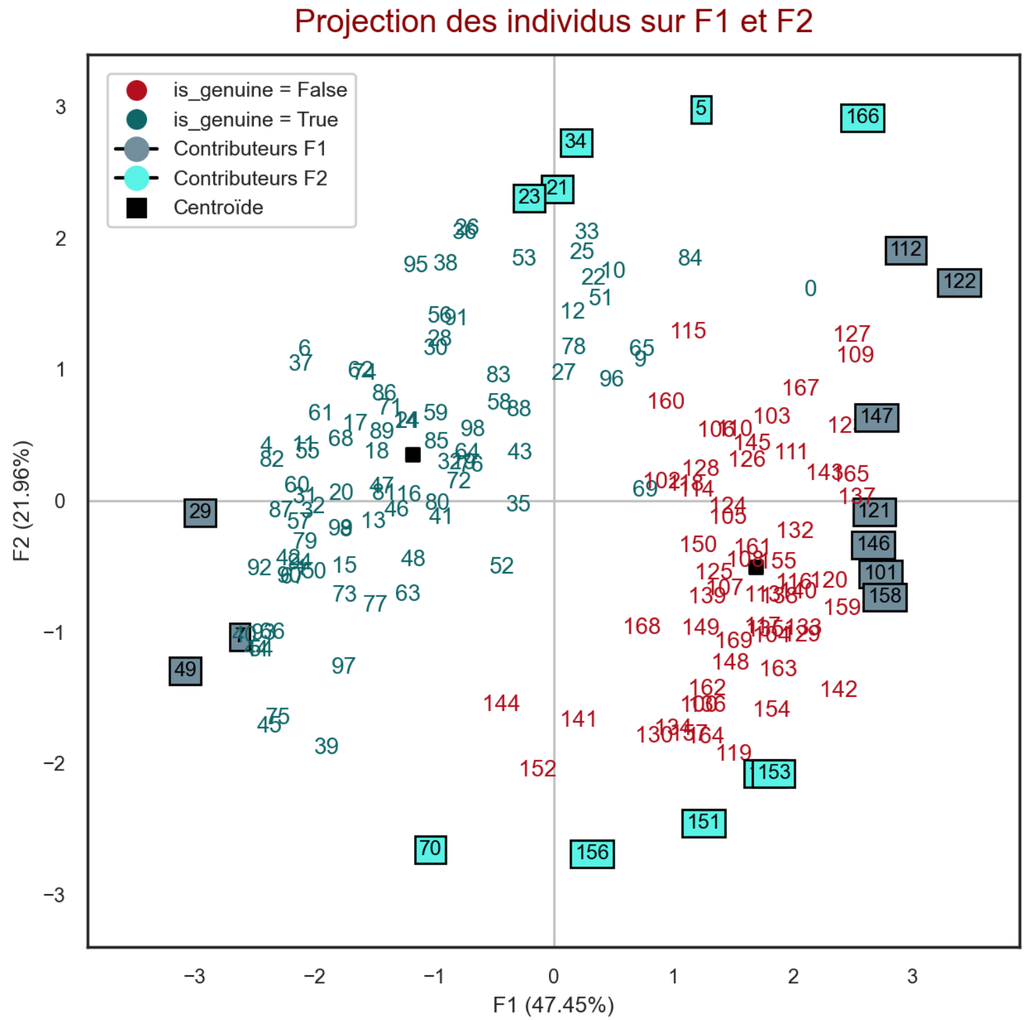
2. Contributions des individus à la création des axes

- elles permettent de déterminer les individus qui pèsent le plus dans la définition de chaque facteur

- On récupère les valeurs depuis la fonction ACP

F1		F2	
id_ind		id_ind	
122	2.38%	5	3.97%
49	1.96%	166	3.79%
29	1.81%	34	3.33%
112	1.8%	156	3.23%
158	1.58%	70	3.15%
101	1.55%	151	2.69%
147	1.51%	21	2.52%
121	1.49%	23	2.37%
146	1.48%	131	1.93%
7	1.42%	153	1.93%

On affiche ces individus sur le graphique de projection des individus



- Il n'y a pas d'individus qui contribuent de manière excessive à la création des axes
- Les individus qui contribuent le plus à la création des axes sont les individus situés à leurs extrémités

PARTIE 3

ALGORITHME DE CLASSIFICATION : LE K-MEANS

A/ Objectif et Principe

- On va appliquer un algorithme de classification sur nos données initiales, sans prendre en compte la variable cible *is_genuine*

- Obj : vérifier que les groupes "Vrais Billets" et "Faux Billets" sont bien des groupes différents,
  c'est-à-dire des groupes composés d'individus aux caractéristiques similaires

- Méthode d'apprentissage non supervisé choisi : K-means

Définition

- Le k-means est un algorithme itératif qui minimise l'inertie intraclasse
  → c'est-à-dire qui minimise la somme des distances entre chaque individu et son centroïde

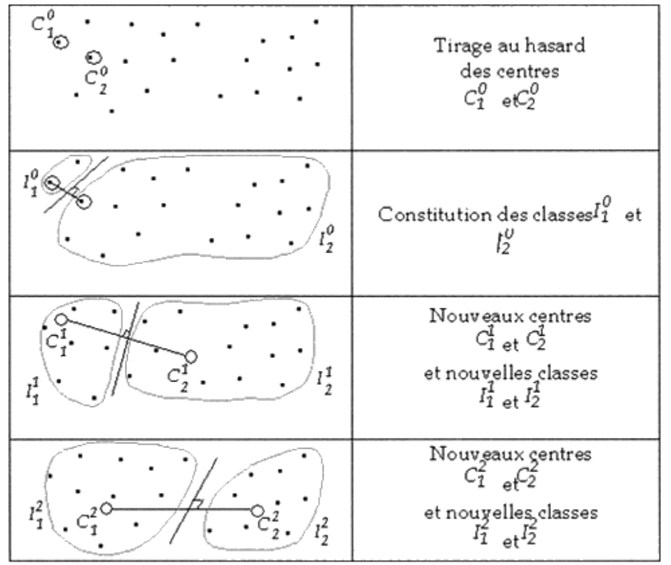
Remarque : Nécessité de spécifier le nombre de clusters que l'on souhaite (k = nombre de clusters)

Principe

- Initialisation : - on choisit aléatoirement k individus comme centres de gravité des clusters (cf. les centroïdes)

- Itérations : - chaque individu est affecté au groupe dont il est le plus proche de son centroïde
  - calcul des nouveaux centroïdes suite à l'affectation d'un individu

- Convergence : - fin de l'algorithme lorsque le modèle a convergé :
  c'est-à-dire lorsqu'on ne peut plus diminuer l'inertie intraclasse



Source : Data mining et statistique décisionnelle : l'intelligence des données, Stéphane Tufféry

B/ Application du k-means et analyse du résultat

- On lance l'algorithme du k-means

```
km = KMeans(n_clusters=2)
km = km.fit(X_scaled)
clusters = km.labels_
data_new=data_kmeans.copy()
data_new['clusters']=clusters
```

- On analyse la partition obtenue

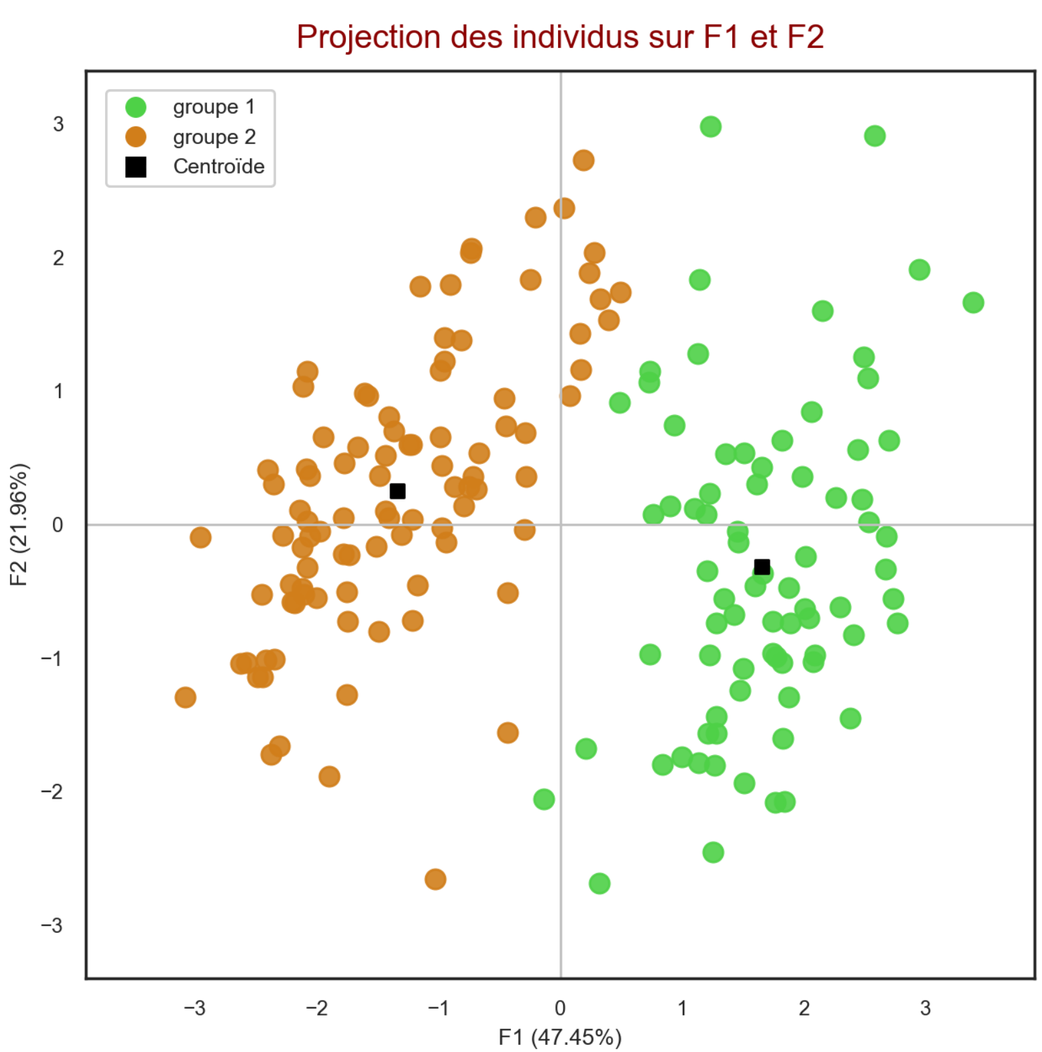
→ On compare les groupes obtenus par le k-means avec les groupes initiaux de vrais et faux billets

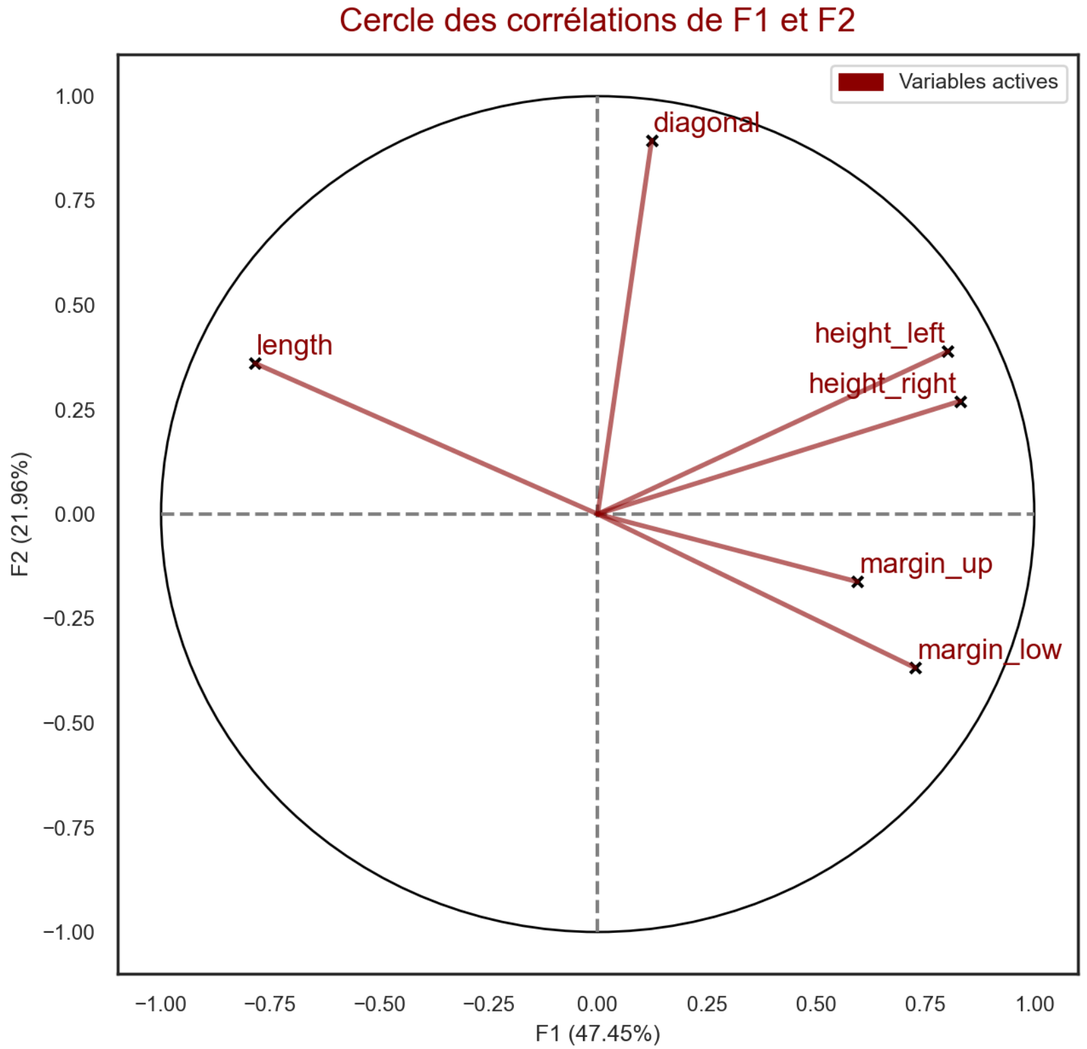
```
# indices des groupes initiaux vrais et faux billets
vrais_billets_ind = data[data['is_genuine'] == True].index
faux_billets_ind = data[data['is_genuine'] == False].index

group_test=data_new[data_new['clusters'] == 1].index
# différencier les cas où groupe1 du kmeans correpond aux vrais billets...
if 18 in group_test:
    data_new['cluster']=data_new['clusters']
# ... des cas où groupe1 du kmeans correpond aux faux billets
else:
    data_new['cluster']=data_new['clusters'].apply(lambda x: (x-1 if x==1 else x+1))

# indices des groupes obtenus suite au kmeans
groupe_0_kmeans = data_new[data_new['cluster'] == 0].index
groupe_1_kmeans = data_new[data_new['cluster'] == 1].index
```

→ Résultat :

- 7 vrais billets interprétés comme des 'faux billets' par le kmeans → soit **7 faux négatifs**
- 1 Faux billet interprété comme un 'vrai billet' par le kmeans : → soit **1 faux positif**

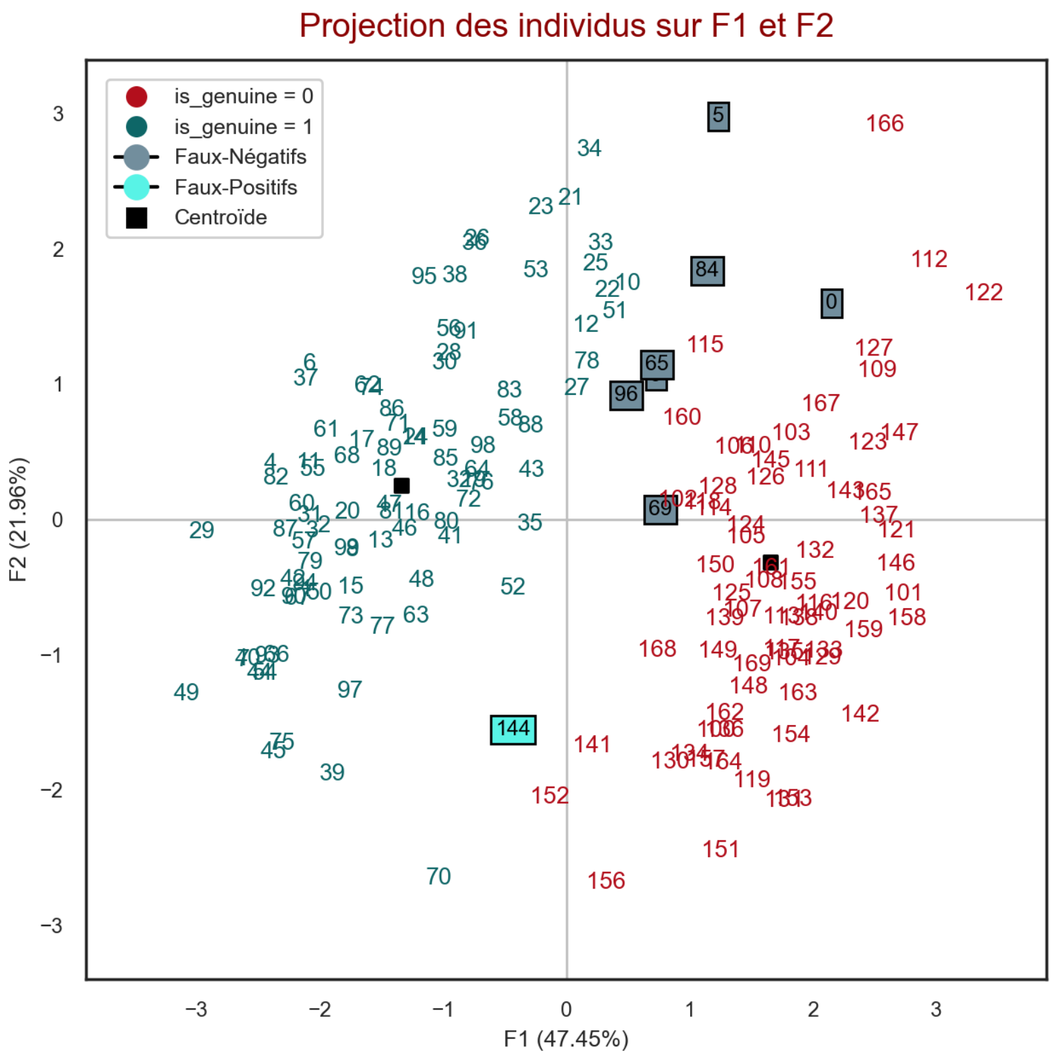
C/ Visualisation du K-means sur le 1er plan de l'ACP

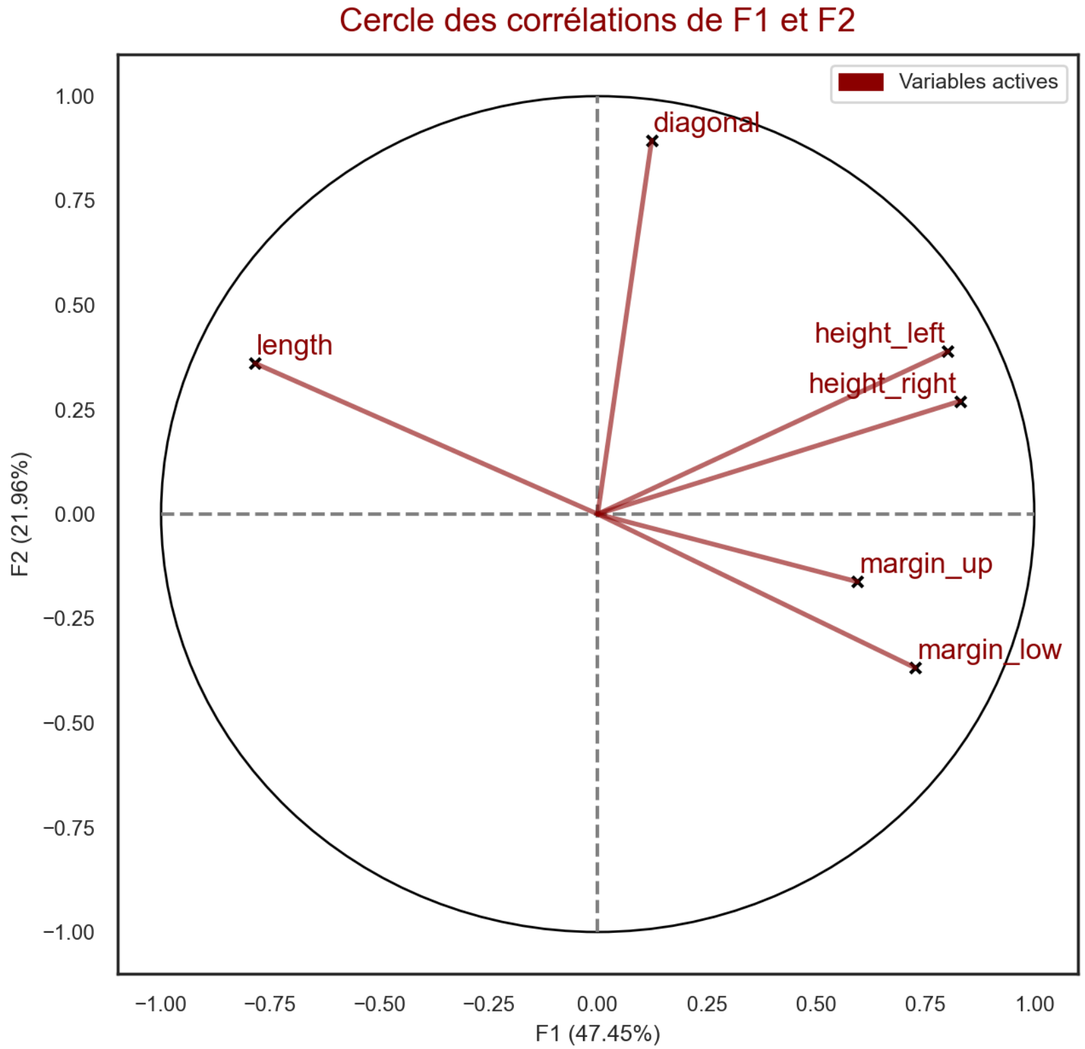




➔ On va mettre en évidence les différences avec les groupes "vrais billets" et "faux billets"

C/ Visualisation du K-means sur le 1er plan de l'ACP



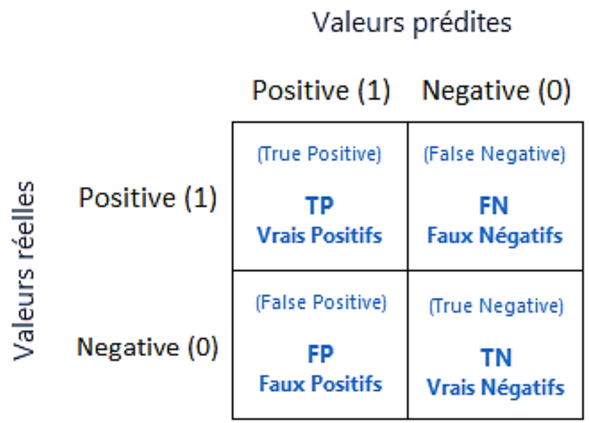


➔ Les individus mal classés se situent à la frontière des 2 groupes

D/ Analyse de la partition

La Matrice de confusion

- tableau récapitulatif utilisé pour évaluer les performances d'un modèle de classification.
- indique nombre de prédictions correctes et incorrectes pour chacune des 2 classes *True* or *False*



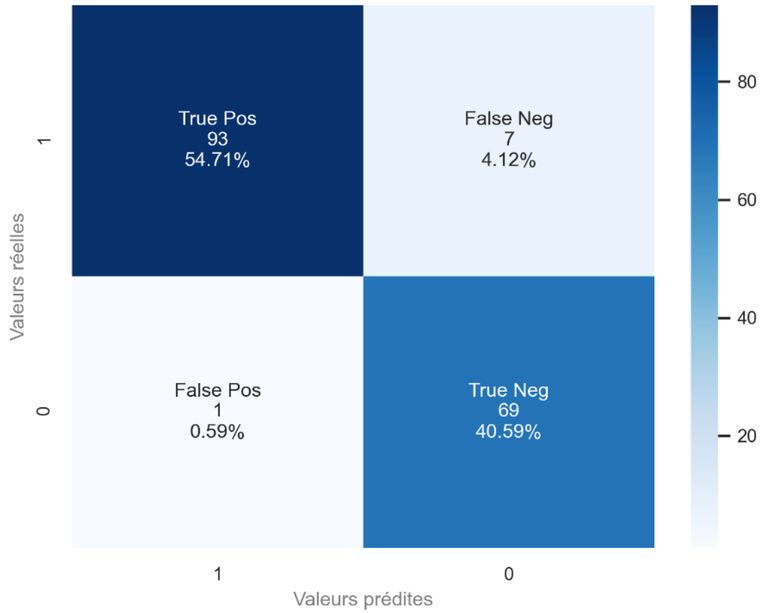
TP : Vrais Positifs
TN : Vrais Négatifs → Val. prédites = Valeurs réelles

FP : Faux Positifs
FN : Faux Négatifs → Val. prédites != Valeurs réelles

Faux **Positifs** : considérés positifs **par le modèle** alors que en réalité négatifs
-> Erreur de type I

Faux **Négatifs** : considéré négatifs **par le modèle** alors que en réalité positifs
-> Erreur de type II

D/ Analyse de la partition



On peut déduire de la matrice de confusion différents critères de performances : les METRICS

$$Sensibilite = Rappel = \frac{TP}{TP + FN}$$

taux de vrais positifs
↳proportion de positifs correctement identifiée par le modèle

$$Specificite = \frac{TN}{TN + FP}$$

taux de vrais négatifs
↳proportion de négatifs correctement identifiée par le modèle

$$Precision = \frac{TP}{TP + FP}$$

→ **proportion de prédictions correctes** parmi les individus que l'on a prédits positifs

$$F - mesure = F1score = 2 \times \frac{Precision \times Sensibilite}{Precision + Sensibilite} = \frac{2TP}{2TP + FP + FN}$$

→ **moyenne** harmonique de la **précision et** de la **sensibilité**

Critères de performances obtenus par le k-means

	precision	recall	f1-score	support
Faux_billets	0.91	0.99	0.95	70
Vrais Billets	0.99	0.93	0.96	100
accuracy			0.95	170
macro avg	0.95	0.96	0.95	170
weighted avg	0.96	0.95	0.95	170

Rappel :

- objectif de la mission :
  → créer un algo de détection de faux billets
  - On veut donc avant tout éviter les Faux Positifs
  - On privilégiera la Spécificité à la Sensibilité

PARTIE 4

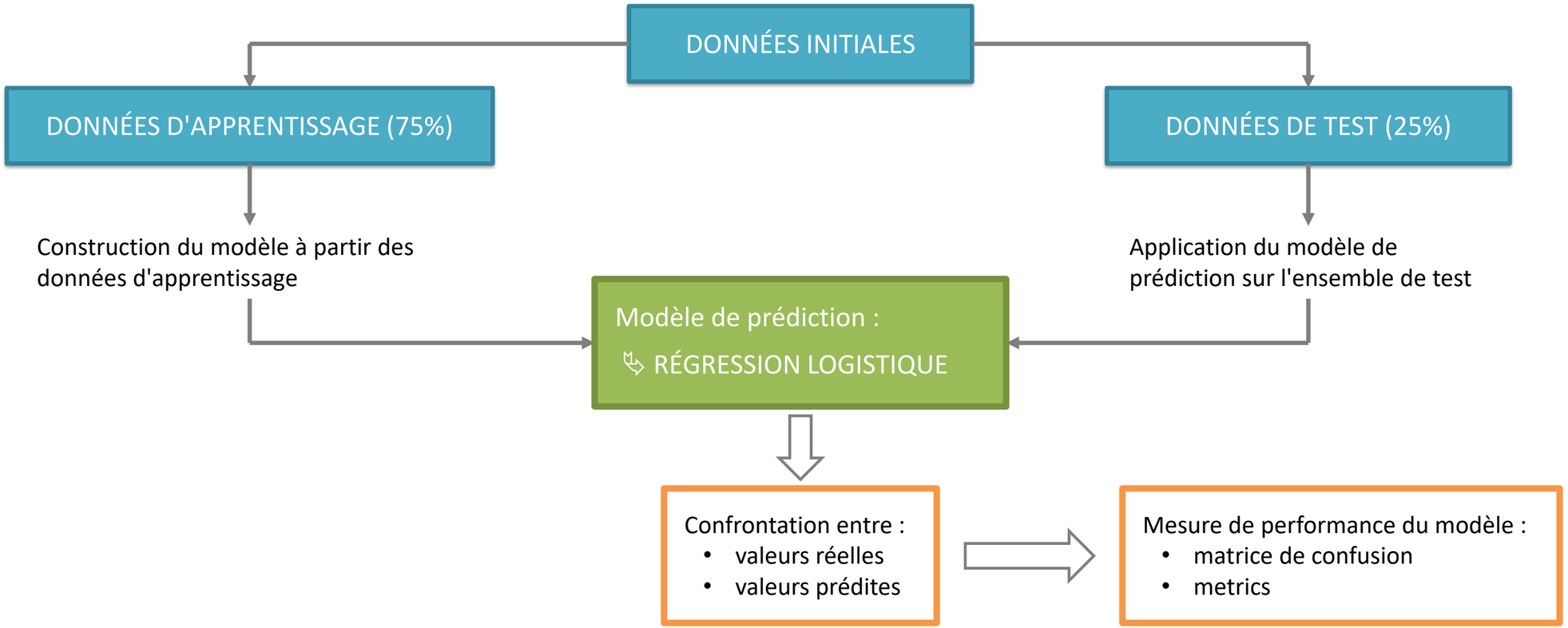
RÉGRESSION LOGISTIQUE

A/ Objectif et Méthode

Objectif

- construire un modèle qui permette de **prédire si un billet est vrai ou faux** (cf. la cible, variable binaire qualitative) **à partir de ses caractéristiques géométriques** (cf. les variables explicatives)

Méthode



B/ Préparation des données

Variable cible et variables explicatives

```
# variables explicatives
x_cols = ['length', 'height_left', 'height_right', 'margin_low', 'margin_up', 'diagonal']

# variable cible
y_col = 'is_genuine'

# df variables explicatives
X = data_reg[x_cols]

# df variable cible
y = data_reg[[y_col]]
```

Partition aléatoire du jeu de données initiales en données d'entrainement et de test

```
from sklearn.model_selection import train_test_split

#Partition aléatoire du jeu de données en 75% pour créer le modèle, 25% pour tester le modèle
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, stratify = y, random_state=23)
```

C/ Construction du modèle

Création et entraînement du modèle

```
from sklearn import linear_model

# instanciation du modele
logreg=linear_model.LogisticRegression()

# entrainement du modèle
logreg.fit(X_train,np.ravel(y_train))
```

Prédiction (jeu de test)

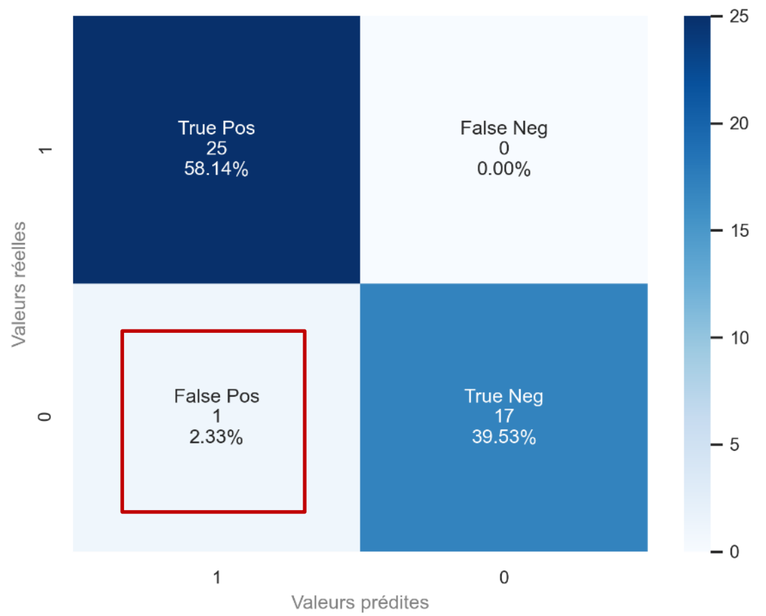
```
# stockages de prédictions
y_pred = logreg.predict(X_test)
```

Confrontation données prédites et données réelles

```
from fonctions_perso import heatmap_matrice_confusion

# on affiche la matrice de confusion
conf_mat_reg = confusion_matrix(y_test, y_pred)
conf_mat_reg
heatmap_matrice_confusion(conf_mat_reg, num_graph=9)
```

```
# on affiche les mesures des performances
target_names = ['Faux_billets', 'Vrais Billets']
print(classification_report(y_test, y_pred, target_names=target_names))
```

Matrice de confusion



Mesures de performances

	precision	recall	f1-score	support
Faux_billets	1.00	0.94	0.97	18
Vrais Billets	0.96	1.00	0.98	25
accuracy			0.98	43
macro avg	0.98	0.97	0.98	43
weighted avg	0.98	0.98	0.98	43

D/ Analyse des probabilités

- On peut récupérer les probabilités associées aux prédictions :

```
# probabilités associées aux prédictions
proba_pred = logreg.predict_proba(X_test)

# on affiche les 5 premiers éléments
proba_pred[:5]
```

```
array([[0.84972937, 0.15027063],
       [0.01002712, 0.98997288],
       [0.93793928, 0.06206072],
       [0.00879464, 0.99120536],
       [0.3911693 , 0.6088307 ]])
```

Pour chaque prédiction, *predic_proba* renvoie 2 probabilités :

- Proba ('is_genuine' = False) → probabilité que le billet soit un faux
- Proba ('is_genuine' = True) → probabilité que le billet soit un vrai

avec : Proba ('is_genuine' = False) + Proba ('is_genuine' = True) = 1

(remarque : par défaut, *predict* prend une probabilité de 0,5 pour classer les prédictions, mais possibilité de modifier ce seuil)

- On vérifie la probabilité du Faux Positif

	is_genuine	diagonal	height_left	height_right	margin_low	margin_up	length	proba_vrai_billet	index_billets
26	0	171.9400	104.2100	104.1000	4.2800	3.4700	112.2300	0.6088	102

VALEUR RÉELLE :

'is_genuine' = 0 ⇔ 'is_genuine' = False

➔ C'est un Faux billet

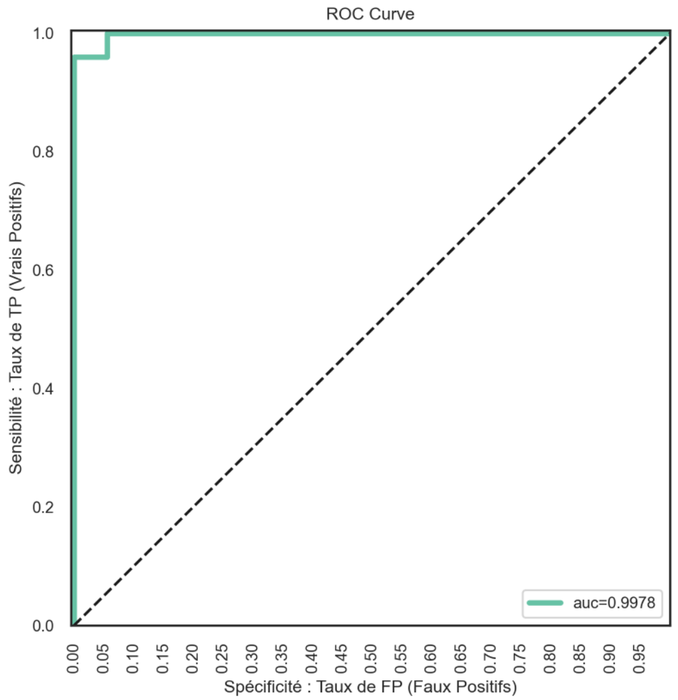
PRÉDICTION :

Proba (Vrai_Billet) > 0,5 ➔ Considéré comme un vrai billet par le modèle

Faux billet considéré par le modèle comme un vrai billet ➔ Faux Positif

E/ Courbe ROC et valeur AUC

- On peut représenter la performance du modèle de régression grâce à une courbe ROC (*Receiver Operating Characteristic*)
  - Cette courbe donne le taux de vrais positifs (fraction des positifs qui sont effectivement détectés : cf. la ***sensibilité*** ou *rappel*) en fonction du taux de faux positifs (fraction des négatifs qui sont incorrectement détectés : cf. la ***spécificité***).



- lecture du graphique :

On peut mesurer la performance du modèle à l'aide de la valeur AUC (cf. l'aire sous la courbe ROC) :
- valeurs possibles : de 0 à 1 → plus la valeur est proche de 1, meilleur est le modèle

```
# calcul de la valeur AUC
auc = metrics.roc_auc_score(y_test, y_pred_proba)
auc
```

```
0.9977777777777778
```

PARTIE 5

PROGRAMME DE DÉTECTION DE FAUX BILLETS

Principe et lien

LE PRINCIPE :

- on utilise le modèle généré précédemment
- on applique le modèle au jeu de test uploadé :
  - pour chaque observation, on récupère la probabilité que le billet soit un vrai : Proba('is_genuine' = True)
  - on affiche cette probabilité
  - on classe la prédiction de la façon suivante :
    - si p >= 0.5 alors le billet est considéré comme vrai
    - si p < 0.5 alors le billet est considéré comme faux

Lien pour tester un jeu de données

- Veuillez vous rendre sur la page suivante pour uploader votre fichier csv et visualiser directement les résultats : page de test

Conclusion

Plusieurs méthodes de traitement du jeu de données initiales :

- **L'analyse des variables** → avoir un aperçu des variables qui jouent un rôle dans la composition des 2 groupes

- **L'ACP** → visualiser sur le 1er plan factoriel la répartition des 2 groupes
  → synthétiser les 6 variables initiales en 2 composantes principales

- **Le k-means** → constituer 2 groupes sans prendre en compte la variable cible
  → analyser les différences grâce à la matrice de confusion

- **La régression logistique** → construire un modèle de prédiction
  → score plus important qu'avec la méthode du k-means

MERCI POUR VOTRE ATTENTION

QUESTIONS ?