

Année académique 24/25  
Introduction à IIA

## Compte rendu de deuxième TP



**Encadré par**

Madame BANOUAR Oumayma

**Réalisé par**

Bouzidi Safa

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Les Questions 1 jusqu'a 3 : Préparation de Dataset</b>	<b>5</b>
1	Vérifier s'il y a des doublons et des lignes vides ou pas . . . . .	5
2	Séparation des features et des targets . . . . .	6
3	Standardisation . . . . .	7
<b>3</b>	<b>Question 4 et 5 : Réalisation de l'ACP</b>	<b>8</b>
<b>4</b>	<b>Question 6 : Choix du nombre de composantes à conserver :</b>	<b>9</b>
<b>5</b>	<b>Question 7 : Représentation Graphique des individus et les sports en exploitant les deux dimensions</b>	<b>10</b>
<b>6</b>	<b>Travail en autonomie</b>	<b>13</b>
<b>7</b>	<b>Conclusions</b>	<b>19</b>
<b>8</b>	<b>Bibliographie</b>	<b>20</b>

## Table des figures

1	Les 5 premières lignes du DATABASE . . . . .	5
2	Ligne vide . . . . .	5
3	Ligne vide . . . . .	6
4	Séparation des variables . . . . .	6
5	Les valeurs standardisé . . . . .	7
6	Représentation des valeurs propres . . . . .	8
7	Sortie Question7 . . . . .	11
8	Distribution des données . . . . .	13
9	Comparaison entre les compétitions . . . . .	15
10	Meilleurs athlètes (1er à 3e rang) par compétition . . . . .	16
11	La heatmap . . . . .	17
13	Les classements . . . . .	19

# 1 Introduction

L'analyse des performances sportives est cruciale pour mieux comprendre les facteurs clés de succès dans des disciplines comme le décathlon, où les athlètes participent à plusieurs épreuves aux caractéristiques variées. Dans ce projet, nous allons explorer un jeu de données contenant les résultats d'athlètes dans différentes épreuves sportives et extraire des informations pertinentes à travers des techniques d'analyse statistique et de modélisation. L'objectif est de séparer les variables explicatives (les performances dans chaque épreuve) et les variables cibles (le classement et les points) afin de mieux comprendre les relations entre ces facteurs. Nous commencerons par un prétraitement des données, notamment la gestion des données manquantes et la standardisation, avant de procéder à une analyse plus approfondie, incluant des méthodes comme l'Analyse en Composantes Principales (ACP). Cette approche nous permettra d'identifier les épreuves ayant le plus grand impact sur le classement des athlètes, et d'optimiser les stratégies de préparation pour les compétitions futures.

## 2 Les Questions 1 jusqu'a 3 : Préparation de Dataset

La première étape de cet exercice consiste en le traitement et la préparation de la base de données, une étape cruciale dans tout processus de modélisation en machine learning. Cette phase permet de bien comprendre la structure des données, de les nettoyer, de les normaliser et de les préparer pour les différents modèles à utiliser par la suite.

```
#Charger les donn es
import pandas as pd
# Load the dataset
file_path = '/content/decathlon.csv'
data = pd.read_csv(file_path)
# Display the first few rows to understand the data structure
data.head()
```

Résultat :



	Athlets	100m	Long.jump	Shot.put	High.jump	400m	110m.hurdle	Discus	Pole.vault	Javeline	1500m	Rank	Points	Competition
0	SEBRLE	11.04	7.58	14.83	2.07	49.81	14.69	43.75	5.02	63.19	291.7	1	8217	Decastar
1	CLAY	10.76	7.40	14.26	1.86	49.37	14.05	50.72	4.92	60.15	301.5	2	8122	Decastar
2	KARPOV	11.02	7.30	14.77	2.04	48.37	14.09	48.95	4.92	50.31	300.2	3	8099	Decastar
3	BERNARD	11.02	7.23	14.25	1.92	48.93	14.99	40.87	5.32	62.77	280.1	4	8067	Decastar
4	YURKOV	11.34	7.09	15.19	2.10	50.42	15.31	46.26	4.72	63.44	276.4	5	8036	Decastar

FIGURE 1 – Les 5 premières lignes du DATABASE

**Interprétation :** Le script commence par charger le fichier CSV .La fonction `head()` en Python est utilisée pour afficher les premières lignes d'un DataFrame

### 1 Vérifier s'il y a des doublons et des lignes vides ou pas

```
# Check for missing values in the dataset
missing_values = data.isnull().sum()
data.info()
# Display columns with missing values
missing_values[missing_values > 0]
#checker s'il y a des doublons
import pandas as pd
# Lire les donn es
data = data = pd.read_csv('/content/decathlon.csv')
data.loc[data['Athlets'].duplicated(keep=False),:]
```

Résultat :



	Athlets	100m	Long.jump	Shot.put	High.jump	400m	110m.hurdle	Discus	Pole.vault	Javeline	1500m	Rank	Points	Competition
--	---------	------	-----------	----------	-----------	------	-------------	--------	------------	----------	-------	------	--------	-------------

FIGURE 2 – Ligne vide

```

11 Rank          41 non-null    int64
12 Points        41 non-null    int64
13 Competition   41 non-null    object
dtypes: float64(10), int64(2), object(2)
memory usage: 4.6+ KB

```

a

FIGURE 3 – Ligne vide

**Interprétation :** le rapport indique que toutes les colonnes, y compris les variables Rank, Points, et Competition, sont complètes avec 41 valeurs non nulles. Cela signifie que nous n'avons pas besoin de gérer des données manquantes ou des doublons dans cette base de données

## 2 Séparation des features et des targets

```

# Séparation des features et des targets
X = data.drop(columns=['Athlets', 'Rank', 'Points', 'Competition'])
# Features
y = data[['Rank', 'Points', 'Competition']] # Variable cible

X.head()
y.head()

```



	Rank	Points	Competition
0	1	8217	Decastar
1	2	8122	Decastar
2	3	8099	Decastar
3	4	8067	Decastar
4	5	8036	Decastar

(a) Les 5 premières lignes de X

	100m	Long Jump	Shot Put	High Jump	400m	1500m	Hurdle	Discus	Pole Vault	Javelin	1500m
0	11.04	7.58	14.83	2.07	49.81	14.09	43.75	5.02	63.19	291.7	
1	10.76	7.40	14.26	1.86	49.37	14.05	50.72	4.92	60.15	301.5	
2	11.02	7.30	14.77	2.04	48.37	14.09	48.95	4.92	50.31	300.2	
3	11.02	7.23	14.25	1.92	48.83	14.99	40.87	5.32	62.77	280.1	
4	11.34	7.09	15.19	2.10	50.42	15.31	46.26	4.72	63.44	276.4	

(b) Les 5 premières lignes de y

FIGURE 4 – Séparation des variables

### Interprétation :

Dans cette étape, vous avez séparé les features (les variables descriptives des performances dans différentes épreuves comme le 100m, le Long jump, etc.) et la variable cible (le Rank, les Points, et la Competition). Cette séparation est essentielle pour les modèles d'apprentissage automatique, où les features sont utilisées pour prédire la variable cible. Cela permet de traiter les performances des athlètes dans les différentes épreuves comme des variables explicatives, tandis que le Rank et les Points servent de variables à prédire ou à analyser, en fonction du modèle choisi.

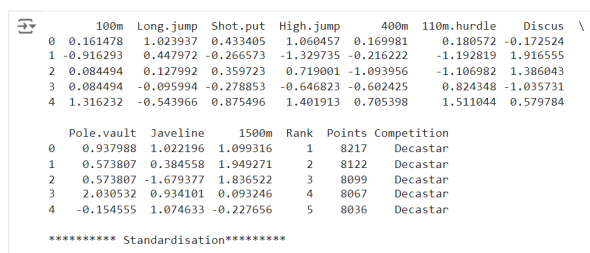
### 3 Standardisation

L'ACP est également sensible à toutes déviation d'échelle entre les variables. S'il existe une variable avec une unité de mesure différente, tels les scores divers allant de 0 à 100 comparativement, à partir de 0 à 10, celle-ci a une influence incommensurable sur les premiers axes principaux, même si elle ne contient pas plus d'information que les autres. La standardisation est l'exhaustivité le centrage et normage chaque variable (soustraction de la moyenne et et diviser les variables par l'écart-type ) de sorte que les moyennes soient égales à 0 et les écarts-types égaux à 1, s'assurant de ce fait que chacune des variables contribue également à l'analyse.

```
#Standardisation
import pandas as pd
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
# Convertir les donn es standardis es en DataFrame
X_scaled_df = pd.DataFrame(X_scaled, columns=X.columns)
# Ajouter la colonne 'tissue_status'      nouveau pour la variable
      cible
X_scaled_df[['Rank', 'Points', 'Competition']] = y
# Afficher un aper u des donn es standardis es
print(X_scaled_df.head())

print('\n*****_Standardisation*****\n')
```

#### Résultat



```

100m Long.jump Shot.put High.jump 400m 110m.hurdle Discus \
0 0.161478 1.023937 0.433405 1.060457 0.169981 0.180572 -0.172524
1 -0.916293 0.447972 -0.266573 -1.329735 -0.216222 -1.192819 1.916555
2 0.084494 0.127992 0.359723 0.719001 -1.093956 -1.106982 1.386043
3 0.084494 -0.095994 -0.278853 -0.646821 -0.602425 0.824348 -1.035731
4 1.316232 -0.543966 0.875496 1.401913 0.705398 1.511044 0.579784

Pole.vault Javeline 1500m Rank Points Competition
0 0.937988 1.022196 1.099316 1 8217 Decastar
1 0.573807 0.384558 1.949271 2 8122 Decastar
2 0.573807 -1.679377 1.836522 3 8099 Decastar
3 2.030532 0.934101 0.093246 4 8067 Decastar
4 -0.154555 1.074633 -0.227656 5 8036 Decastar

***** Standardisation*****

```

FIGURE 5 – Les valeurs standardisé

#### Interprétation :

L'étape de standardisation permet de rendre les données comparables en supprimant l'effet des différentes échelles des variables. En utilisant la standardisation (z-score), chaque variable est centrée autour de 0 et a un écart-type de 1. Cela garantit que toutes les épreuves (comme le 100m, Long.jump, etc.) contribuent de manière égale à l'analyse, évitant qu'une variable avec une grande échelle (par exemple le 1500m en secondes) ne domine les autres. Après standardisation, les variables peuvent être utilisées de manière plus cohérente dans des algorithmes d'analyse comme l'ACP ou les modèles de machine learning.

### 3 Question 4 et 5 : Réalisation de l'ACP

Pourquoi choisir les 10 colonnes liées aux performances des athlètes ? : Les 10 colonnes représentent les performances réelles des athlètes sur les 10 épreuves du décathlon. En les utilisant, on peut analyser les corrélations entre les différentes disciplines et dégager des tendances générales.

**L'importance des valeurs propres :** Les valeurs propres représentent la quantité de variance expliquée par chaque axe principal. Plus la valeur propre est élevée, plus l'axe est important pour expliquer les variations dans les données.

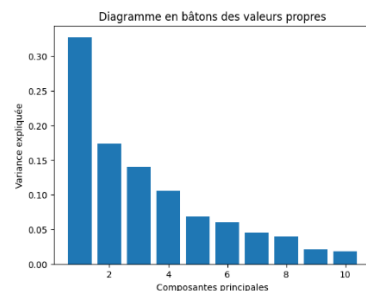
```
from sklearn.decomposition import PCA
# Réalisation de l'ACP
pca = PCA(n_components=10)
X_pca = pca.fit_transform(X_scaled)
# Affichage de l'explication de la variance par composante
print(pca.explained_variance_ratio_)

import matplotlib.pyplot as plt
explained_variance=pca.explained_variance_ratio_
for i,variance in enumerate(explained_variance):
    print(f'variance expliquée par la composante {i+1}:{variance:.4f}')
# Diagramme en bâtons des valeurs propres
plt.bar(range(1, 11), pca.explained_variance_ratio_)
plt.xlabel('Composantes principales')
plt.ylabel('Variance expliquée')
plt.title('Diagramme en bâtons des valeurs propres')
plt.show()
```

Résultat :

```
variance expliqué par la composante 1:0.3272
variance expliqué par la composante 2:0.1737
variance expliqué par la composante 3:0.1405
variance expliqué par la composante 4:0.1057
variance expliqué par la composante 5:0.0685
variance expliqué par la composante 6:0.0599
variance expliqué par la composante 7:0.0451
variance expliqué par la composante 8:0.0397
variance expliqué par la composante 9:0.0215
variance expliqué par la composante 10:0.0182
```

(a) Les valeurs des variances



(b) Diagramme des valeurs propres

FIGURE 6 – Représentation des valeurs propres

**Interprétation :**

**Composante 1 :** Cette première composante principale explique environ 32 % de la variance totale. Cela signifie que la majeure partie de l'information présente dans les



données (environ un tiers) est capturée par ce seul axe principal.

**Composante 2** : La deuxième composante explique environ 17 % de la variance, ce qui est également significatif, bien que moins que la première. Ensemble, les deux premières composantes expliquent près de 50 % de la variance totale.

**Composante 3** : La troisième composante explique 14 % de la variance, ce qui renforce le fait que les premières composantes sont les plus significatives pour représenter la majorité de la variation dans les données.

**Les composantes suivantes (4, 5, etc.)** expliquent des proportions de variance de plus en plus petites, ce qui montre que l'information contenue dans les dernières composantes est moins significative.

## 4 Question 6 : Choix du nombre de composantes à conserver :

Pour déterminer combien de composantes principales je dois conserver, un critère commun est de choisir celles qui expliquent un pourcentage cumulé de variance suffisamment élevé (généralement 80 % ou plus). Dans ce cas, les 5 premières composantes expliquent environ 81 % de la variance totale, ce qui est un seuil souvent suffisant pour une bonne représentation des données tout en réduisant la dimensionnalité.

## 5 Question 7 : Représentation Graphique des individus et les sports en exploitant les deux dimensions

```

# Importer les bibliothèques nécessaires
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

# Appliquer l'ACP avec les 2 premières composantes principales
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

# Extraire les charges (les coefficients des variables sur les
# composantes principales)
loadings = pca.components_.T * np.sqrt(pca.explained_variance_)

# Tracer le cercle de corrélation
def plot_correlation_circle(loadings, pca, labels):
    plt.figure(figsize=(8, 8))
    plt.quiver(np.zeros(loadings.shape[0]), np.zeros(loadings.
        shape[0]),
        loadings[:, 0], loadings[:, 1],
        angles='xy', scale_units='xy', scale=1)
    for i, label in enumerate(labels):
        plt.text(loadings[i, 0], loadings[i, 1], label, color='
            black', ha='center', va='center')

# Cercles de corrélation
circle = plt.Circle((0, 0), 1, color='blue', fill=False,
    linestyle='--')
plt.gca().add_artist(circle)

plt.xlim(-1.1, 1.1)
plt.ylim(-1.1, 1.1)
plt.axhline(0, color='grey', lw=1)
plt.axvline(0, color='grey', lw=1)
plt.xlabel(f"Composante 1 ({pca.explained_variance_ratio_
    [0]*100:.1f}% de variance expliquée)")
plt.ylabel(f"Composante 2 ({pca.explained_variance_ratio_
    [1]*100:.1f}% de variance expliquée)")
plt.title('Cercle de corrélation des variables')
plt.grid()

```

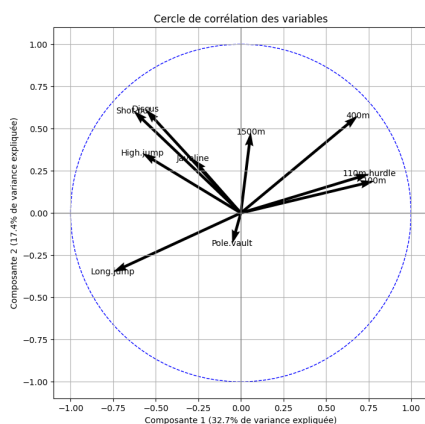
```
plt.show()

# Représentation des individus
def plot_individuals(X_pca, y):
    plt.figure(figsize=(8, 6))
    plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y['Rank'], cmap='
viridis')
    plt.colorbar(label='Classement')
    plt.title('Projection des athlètes sur les deux premières
composantes principales')
    plt.xlabel(f'Composante 1 ({pca.explained_variance_ratio_
[0]*100:.1f}% de variance expliquée)')
    plt.ylabel(f'Composante 2 ({pca.explained_variance_ratio_
[1]*100:.1f}% de variance expliquée)')
    plt.grid()
    plt.show()

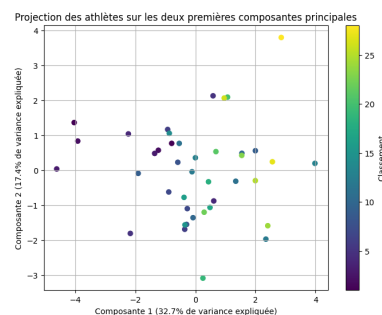
# Tracer le cercle de corrélation
variables = X.columns
plot_correlation_circle(loadings, pca, variables)

# Tracer la projection des individus (athlètes)
plot_individuals(X_pca, y)
```

Résultat :



(a) Cercle de Corrélation des variables



(b) Projection des athlètes sur les 2 premières composantes

FIGURE 7 – Sortie Question7

Interprétation :

**Variables fortement corrélées avec la composante 1 (PC1) :**

Les épreuves comme le 400m, 110m haies, 1500m, et 100m sont fortement corrélées avec la première composante principale. Cela signifie que ces épreuves influencent significativement la variation des performances sur ce premier axe.

Ces épreuves peuvent représenter un certain type d'athlète ayant de bonnes performances en courses de vitesse et d'endurance.

**Variables corrélées avec la composante 2 (PC2) :**

Des épreuves comme le lancer de poids (shot put) et le lancer du disque sont davantage corrélées avec la composante 2. Ce deuxième axe peut être lié à la force physique, puisque ces épreuves sont des épreuves de lancer, qui exigent de la puissance.

**Épreuves opposées dans le cercle :**

Lorsque deux épreuves sont opposées dans le cercle, cela signifie qu'elles sont corrélées de manière négative. Par exemple, le saut en longueur (long jump) est à l'opposé des épreuves de course (1500m, 400m), ce qui suggère que les athlètes performants dans le saut en longueur peuvent avoir des performances relativement moins bonnes dans les épreuves de course de fond et vice versa. 2. Interprétation de la projection des athlètes : Le graphique de projection des athlètes sur les deux premières composantes principales permet d'analyser les profils des athlètes en fonction de leur classement et de leurs performances dans les différentes épreuves.

**Est-ce possible de définir des profils des athlètes ? si oui lesquels ?**

**Oui, il est possible de définir des profils d'athlètes :**

Les athlètes qui se situent à droite de la composante 1 sont probablement ceux qui excellent dans les courses de sprint et de fond (100m, 400m, 1500m, 110m haies).

Ceux qui se situent plus haut sur la composante 2 ont probablement de bonnes performances dans les épreuves de lancer (lancer du poids, lancer du disque).

Les athlètes dispersés autour du centre du graphique ont probablement des performances équilibrées dans différentes épreuves, ce qui pourrait indiquer des profils d'athlètes polyvalents.

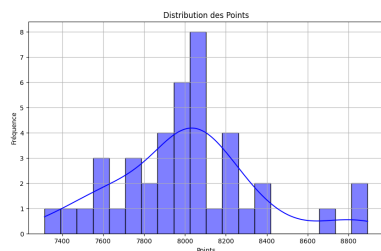
## 6 Travail en autonomie

### • Étape 1 : Exploration des données

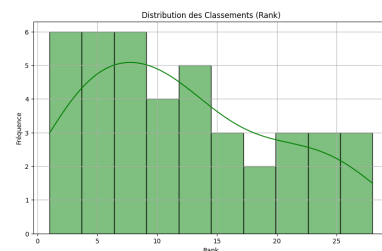
```
# Distribution des points et des rangs
plt.figure(figsize=(10, 6))
sns.histplot(data['Points'], bins=20, kde=True, color='blue')
plt.title('Distribution des Points')
plt.xlabel('Points')
plt.ylabel('Fréquence')
plt.grid(True)
plt.show()

plt.figure(figsize=(10, 6))
sns.histplot(data['Rank'], bins=10, kde=True, color='green')
plt.title('Distribution des Classements (Rank)')
plt.xlabel('Rank')
plt.ylabel('Fréquence')
plt.grid(True)
plt.show()
```

### Résultat



(a) Distribution des Points



(b) Distribution des Classements

FIGURE 8 – Distribution des données

### Interprétation

L'histogramme de la distribution des points montre que la majorité des athlètes obtiennent entre 7800 et 8200 points, avec un pic autour de 8000 points, ce qui suggère que la plupart des performances des athlètes sont concentrées dans cette plage. Il y a cependant des valeurs plus extrêmes vers 7400 et 8800 points, ce qui indique quelques athlètes ayant des performances bien au-dessous ou au-dessus de la moyenne.

Pour la distribution des rangs, nous observons que les rangs les plus fréquents sont autour de 5 à 7, suggérant une forte compétition entre les meilleurs athlètes. Le nombre d'athlètes diminue progressivement au fur et à mesure que le rang augmente, avec une légère irrégularité entre les rangs 15 et 20. Ces distributions montrent une compétition

relativement équilibrée, avec une concentration de performances solides dans les positions médianes à supérieures.

### • Étape 2 : Comparaison entre les compétitions

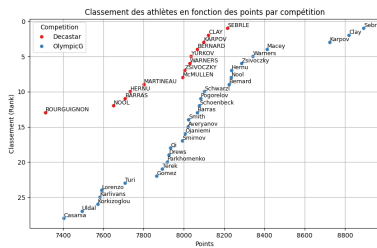
```
# Scatterplot des athlètes en fonction des points et du
# classement (Rank), color par la compétition
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Points', y='Rank', hue='Compétition', data=
    data, palette='Set1')

# Ajouter les noms des athlètes à chaque point
for i in range(data.shape[0]):
    plt.text(data['Points'][i], data['Rank'][i], data['Athlètes'][
        i],
            fontsize=9, verticalalignment='bottom')

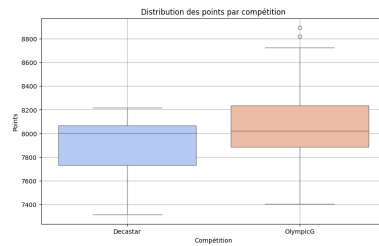
plt.title('Classement des athlètes en fonction des points par
    compétition')
plt.xlabel('Points')
plt.ylabel('Classement (Rank)')
plt.gca().invert_yaxis() # Inverser l'axe des y pour que le
    classement 1 soit en haut
plt.grid(True)
plt.show()

# Boxplot pour visualiser la distribution des points par
    compétition
plt.figure(figsize=(10, 6))
sns.boxplot(x='Compétition', y='Points', data=data, palette='
    coolwarm')
plt.title('Distribution des points par compétition')
plt.xlabel('Compétition')
plt.ylabel('Points')
plt.grid(True)
plt.show()
```

Résultat :



(a) Classement (Rank)



(b) Distribution des points par compétition

FIGURE 9 – Comparaison entre les compétitions

**Interprétation :****Graphique "Classement des athlètes en fonction des points par compétition" :**

Ce scatterplot montre la relation entre les points obtenus et le classement des athlètes, différenciés par compétition. Nous observons que les athlètes des Jeux Olympiques (OlympicG) **En Bleu** ont tendance à obtenir plus de points, atteignant même des sommets avec des performances exceptionnelles comme celles de **Sebrle**. En revanche, les athlètes du **Décastar En Rouge** obtiennent globalement des scores légèrement inférieurs, avec quelques exceptions comme **Clay** et **Karpov**, qui rivalisent avec les meilleurs des Jeux Olympiques. Cela montre une compétition plus féroce aux Jeux Olympiques, où les performances les plus élevées sont concentrées.

**Graphique "Distribution des points par compétition" :**

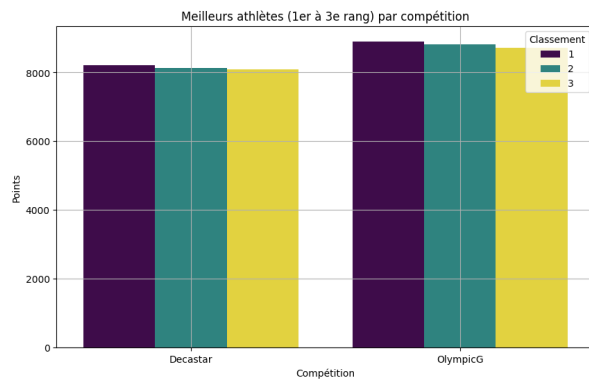
Le boxplot compare la distribution des points entre les deux compétitions, OlympicG et Décastar. Il montre que les athlètes des Jeux Olympiques ont une médiane de points plus élevée (environ 8200), tandis que ceux du Décastar tournent autour de 8000 points.

**• Étape 3 : Analyse des meilleurs profils**

```
# Extraire les athlètes ayant les meilleurs rangs (1er à 3e rang)
best_athletes = data[data['Rank'] <= 3]

plt.figure(figsize=(10, 6))
sns.barplot(x='Compétition', y='Points', hue='Rank', data=
    best_athletes, palette='viridis')
plt.title('Meilleurs athlètes (1er à 3e rang) par compétition')
plt.xlabel('Compétition')
plt.ylabel('Points')
plt.legend(title='Classement')
plt.grid(True)
plt.show()
```

**Résultat :**



**Interprétation :** Le graphique montre que les trois meilleurs athlètes des Jeux Olympiques obtiennent des scores légèrement plus élevés (supérieurs à 8200 points) par rapport à ceux du Décastar (environ 8000 points). Les performances sont assez homogènes dans les deux compétitions, avec une faible différence de points entre les athlètes classés 1er, 2e et 3e, indiquant une forte compétition entre les meilleurs.

FIGURE 10 – Meilleurs athlètes (1er à 3e rang) par compétition

#### • Étape 4 : Analyse des relations entre les variables

```
#observer la corrélation entre Rank, Points et les performances
dans les différentes preuves
performance_columns = ['100m', 'Long.jump', 'Shot.put', 'High.
                        jump', '400m', '110m.hurdle',
                        'Discus', 'Pole.vault', 'Javeline', '1500m
                        ']
plt.figure(figsize=(12, 8))
corr_matrix = data[['Rank', 'Points'] + performance_columns].corr
()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', linewidths
            =0.5)
plt.title('Corrélation entre Rank, Points et les performances')
plt.show()
```

Résultat :



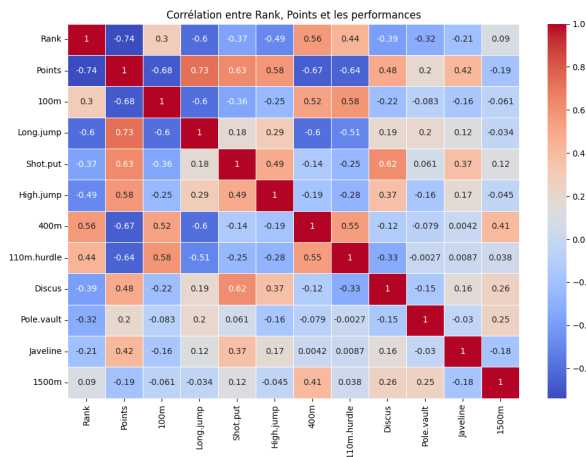


FIGURE 11 – La heatmap

**Interprétation :** La heatmap montre une forte corrélation négative entre le Rank et les Points (-0.74), ce qui est logique : plus un athlète obtient de points, meilleur est son classement. Les épreuves comme le Long jump (0.73) et le 100m (-0.68) sont fortement corrélées aux Points, indiquant que de bonnes performances dans ces épreuves influencent fortement le total des points. Par contre, les épreuves comme le Pole vault et la Javeline ont une corrélation plus faible, suggérant qu'elles ont moins d'impact sur le classement global. En résumé, les épreuves de vitesse et de saut semblent être les plus décisives pour un bon classement.

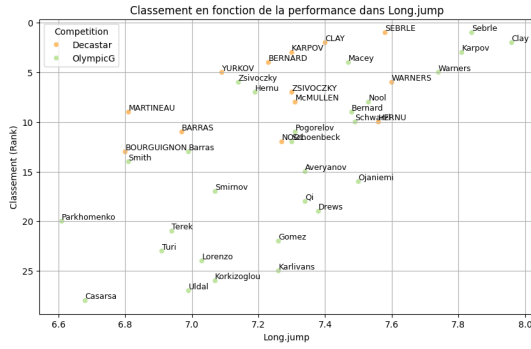
### ●Étape 5 : Visualisation des rangs par épreuve

```
# Scatterplot des points par preuve avec le rang et les noms
des athlètes
for event in performance_columns:
    plt.figure(figsize=(10, 6))
    sns.scatterplot(x=event, y='Rank', data=data, hue='
Competition', palette='Spectral')
    # Ajouter les noms des athlètes chaque point
    for i in range(data.shape[0]):
        plt.text(data[event][i], data['Rank'][i], data['Athlets'
][i],
                fontsize=9, verticalalignment='bottom')

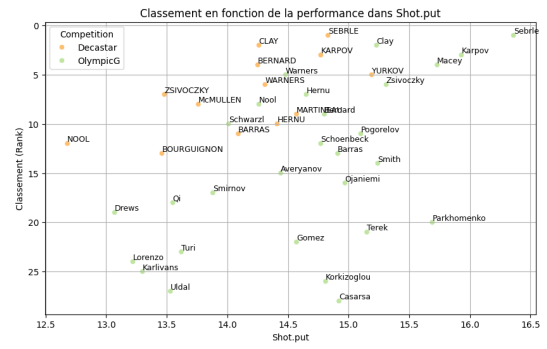
    plt.title(f'Classement en fonction de la performance dans {
event}')
    plt.xlabel(event)
    plt.ylabel('Classement (Rank)')
    plt.gca().invert_yaxis()
    plt.grid(True)
    plt.show()
```

### Résultat :

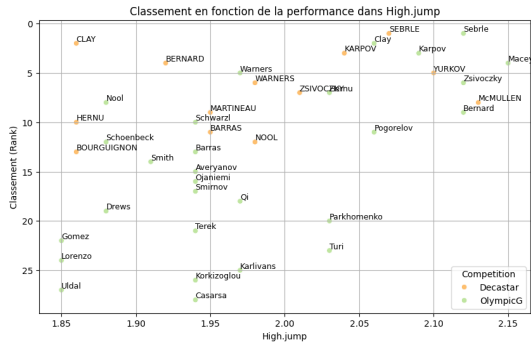
Voici les différentes figures représentant les classements en fonction des performances dans diverses disciplines.



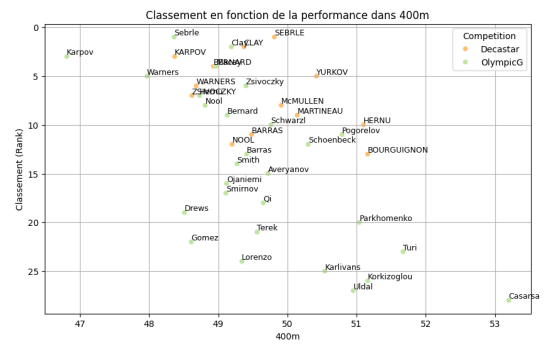
(a) Classement Long.jump



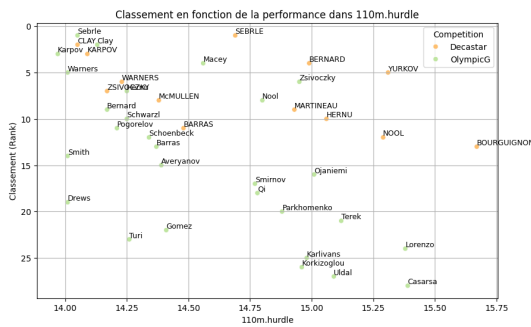
(b) Classement Shot.put



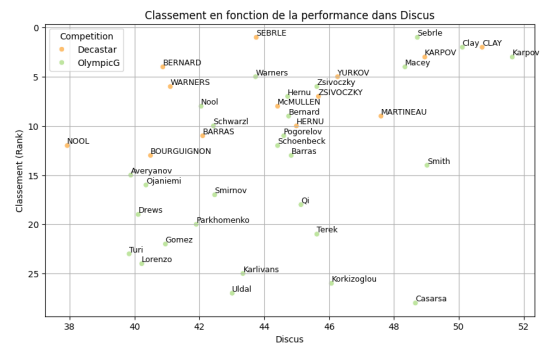
(c) Classement High.jump



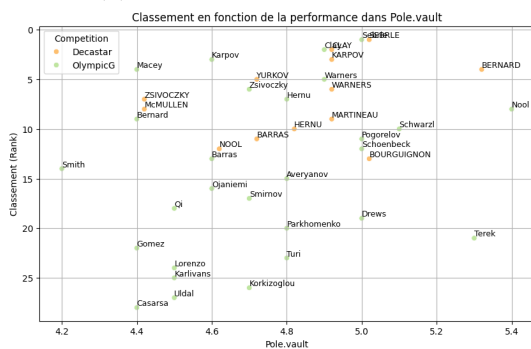
(d) Classement 400m



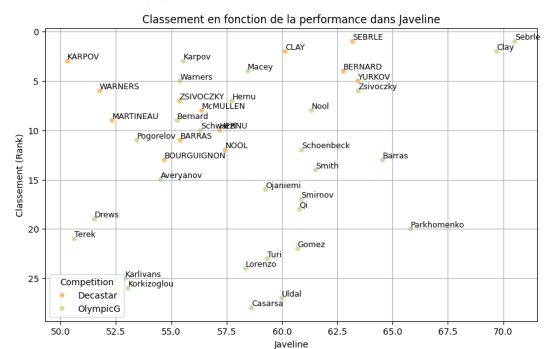
(e) Classement 110m.hurdle



(f) Classement Discus



(g) Classement Pole.vault



(h) Classement Javelin

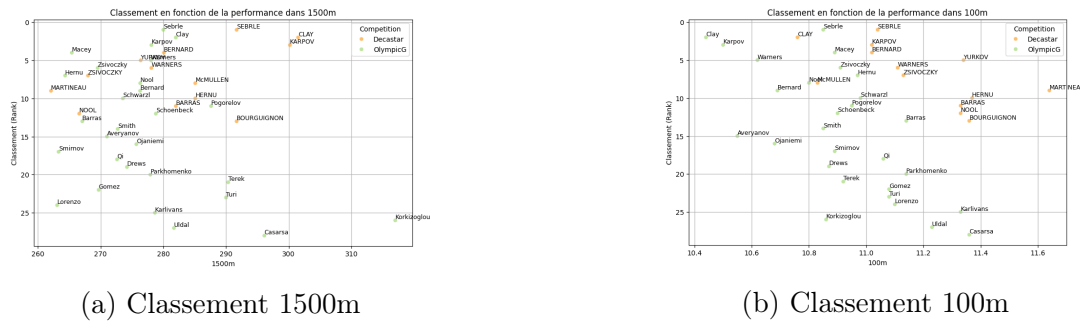


FIGURE 13 – Les classements

### Interprétation :

En analysant les scatterplots, on constate que les athlètes qui dominent les épreuves de sprint (**100m, 110m haies**), comme **Clay** et **Sebrle**, se classent généralement très bien dans les compétitions globales. Ils maintiennent également de bonnes performances dans des épreuves comme le saut en longueur et le lancer de poids, ce qui leur permet de rester en tête du classement.

De plus, les épreuves de saut (**saut en longueur, saut en hauteur**) semblent être des indicateurs forts des meilleurs classements globaux, avec des athlètes comme **Karpov** et **Zsivoczky** qui se distinguent. Les épreuves de course longue, comme le **1500m**, montrent que certains athlètes comme **Sebrle** excellent également dans les courses de fond, ce qui contribue à leur succès global.

### • Profils gagnants :

**Sebrle et Clay** sont clairement les meilleurs athlètes aux **Jeux Olympiques**, excellant dans des épreuves polyvalentes comme le 100m, 110m haies, et saut en hauteur.

**Karpov et Zsivoczky**, en plus d'exceller dans les épreuves **de saut**, se démarquent dans le lancer **de javelot**, ce qui les aide à maintenir un bon classement.

Dans le Décastar, **Nool et Martineau** sont plus compétitifs dans les épreuves de sprint et de saut, mais ils sont moins dominants dans les épreuves de lancer comme le disque.

## 7 Conclusions

L'analyse des performances des athlètes dans ce travail a permis de mieux comprendre les facteurs qui influencent leur classement global dans des compétitions comme les Jeux Olympiques et le Décastar. À travers l'Analyse en Composantes Principales (ACP), nous avons pu identifier les épreuves qui ont le plus d'impact sur le total des points et le classement, notamment les épreuves de sprint, de saut, et de certaines disciplines de lancer comme le poids et le javelot.

## 8 Bibliographie

— [Lien vers mon code Google Colab](#)