Sprint 6 :S06 T01: Tasca dades, probabilitats i estadístiques

Nivell 1

Exercici 1

Agafa un conjunt de dades de tema esportiu que t'agradi i selecciona un atribut del conjunt de dades.

Calcula la moda, la mediana, la desviació estàndard i la mitjana aritmètica.

In [127…

```
# Tratamiento de datos
# ==============================================================================
import pandas as pd
import numpy as np

# Gráficos
# ==============================================================================
import matplotlib.pyplot as plt
from matplotlib import style
import seaborn as sns

# Preprocesado y análisis
# ==============================================================================
#import statsmodels.api as sm
#import pingouin as pg
from scipy import stats
import random as rd
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import SMOTE
from scipy.stats import pearsonr
from statistics import mode

# Configuración matplotlib
# ==============================================================================
plt.style.use('ggplot')

# Configuración warnings
# ==============================================================================
import warnings
warnings.filterwarnings('ignore')
```

A) Data Frame

Para realizar este Sprint he seleccionado la información contenida en la página web: https://www.kaggle.com/datasets/heesoo37/120-years-of-olympic-history-athletes-and-results

En la web se encuentra disponible la base de datos histórica de los juegos olímpicos de verano e invierno: Athens 1896 - Rio 2016

In [128…

```
df_atletas= pd.read_csv(r"C:\Users\hecto\OneDrive\Documentos\IT Data Science\Sprint5
```

In [129…

```
df_atletas.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271116 entries, 0 to 271115
Data columns (total 15 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   ID      271116 non-null  int64  
 1   Name    271116 non-null  object 
 2   Sex     271116 non-null  object 
 3   Age     261642 non-null  float64
 4   Height  210945 non-null  float64
 5   Weight  208241 non-null  float64
 6   Team    271116 non-null  object 
 7   NOC     271116 non-null  object 
 8   Games   271116 non-null  object 
 9   Year    271116 non-null  int64  
 10  Season  271116 non-null  object 
 11  City    271116 non-null  object 
 12  Sport   271116 non-null  object 
 13  Event   271116 non-null  object 
 14  Medal   39783 non-null   object 
dtypes: float64(3), int64(2), object(10)
memory usage: 31.0+ MB
```

Detalle de los campos:

El archivo atleta_eventos.csv contiene 271116 filas y 15 columnas. Cada fila corresponde a un atleta individual compitiendo en un evento olímpico.

ID - Unique number for each athlete

Name - Athlete's name

Sex - M or F

Age - Integer;

Height - In centimeters

Weight - In kilograms

Team - Team name

NOC - National Olympic Committee 3-letter code

Games - Year and season

Year - Integer

Season - Summer or Winter

City - Host city

Sport - Sport

Event - Event

Medal - Gold, Silver, Bronze, or NA.

In [130…
```
df_atletas_pre = df_atletas
df_atletas_pre.fillna(0, inplace=True)
```

In [131…
```
df_atletas_ok = df_atletas_pre[(df_atletas_pre['Age']>0)&(df_atletas_pre['Height']>0
```

In [132…
```
df_atletas_ok.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 206165 entries, 0 to 271115
```

```
Data columns (total 15 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   ID      206165 non-null  int64
 1   Name    206165 non-null  object
 2   Sex     206165 non-null  object
 3   Age     206165 non-null  float64
 4   Height  206165 non-null  float64
 5   Weight  206165 non-null  float64
 6   Team    206165 non-null  object
 7   NOC     206165 non-null  object
 8   Games   206165 non-null  object
 9   Year    206165 non-null  int64
 10  Season  206165 non-null  object
 11  City    206165 non-null  object
 12  Sport   206165 non-null  object
 13  Event   206165 non-null  object
 14  Medal   206165 non-null  object
dtypes: float64(3), int64(2), object(10)
memory usage: 25.2+ MB
```

In [133…

```
df_atletas_ok.head(5)
```

Out[133…

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary
5	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary
6	5	Christine Jacoba Aaftink	F	25.0	185.0	82.0	Netherlands	NED	1992 Winter	1992	Winter	Albertville

B) Moda, la mediana, la desviació estàndard i la mitjana aritmètica

In [201…

```
df_atletas_ok.describe()
```

Out[201…

	ID	Age	Height	Weight	Year
count	206165.000000	206165.000000	206165.000000	206165.000000	206165.000000
mean	68616.017675	25.055509	175.371950	70.688337	1989.674678
std	38996.514355	5.483096	10.546088	14.340338	20.130865
min	1.000000	11.000000	127.000000	25.000000	1896.000000

	ID	Age	Height	Weight	Year
25%	35194.000000	21.000000	168.000000	60.000000	1976.000000
50%	68629.000000	24.000000	175.000000	70.000000	1992.000000
75%	102313.000000	28.000000	183.000000	79.000000	2006.000000
max	135571.000000	71.000000	226.000000	214.000000	2016.000000

In [234...

```
media = df_atletas_ok.Age.mean(), df_atletas_ok.Height.mean(), df_atletas_ok.Weight.
mediana = df_atletas_ok.Age.median(),df_atletas_ok.Height.median(),df_atletas_ok.Wei
desvStd = df_atletas.Age.std(),df_atletas.Height.std(),df_atletas.Weight.std()
var= df_atletas.Age.var(),df_atletas.Height.var(),df_atletas.Weight.var()
moda= mode(df_atletas_ok["Age"]), mode(df_atletas_ok["Height"]),mode(df_atletas_ok["

resumen =pd.DataFrame({"Media":media,"Mediana":mediana,"Desv_Std":desvStd,"Varianza"
resumen
```

Out[234...

	Media	Mediana	Desv_Std	Varianza	Moda
Age	25.055509	24.0	7.840652	61.475818	23.0
Height	175.371950	175.0	73.450560	5394.984696	180.0
Weight	70.688337	70.0	32.381492	1048.560993	70.0

Exercici 2

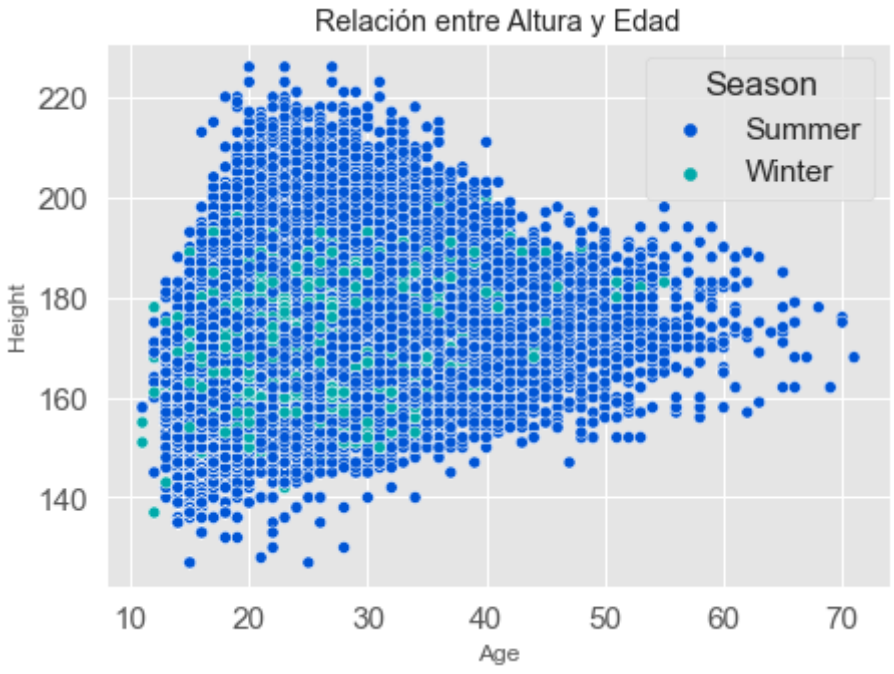
Continuant amb les dades de tema esportiu, selecciona dos atributs i calcula'n la seva correlació.

In [237...

```
plt.figure(figsize=(7,5))
plt.title('Relación entre Altura y Edad')
sns.scatterplot(data = df_atletas_ok, x = "Age", y = "Height", hue="Season", palette
```

Out[237...

```
<AxesSubplot:title={'center':'Relación entre Altura y Edad'}, xlabel='Age', ylabel
='Height'>
```

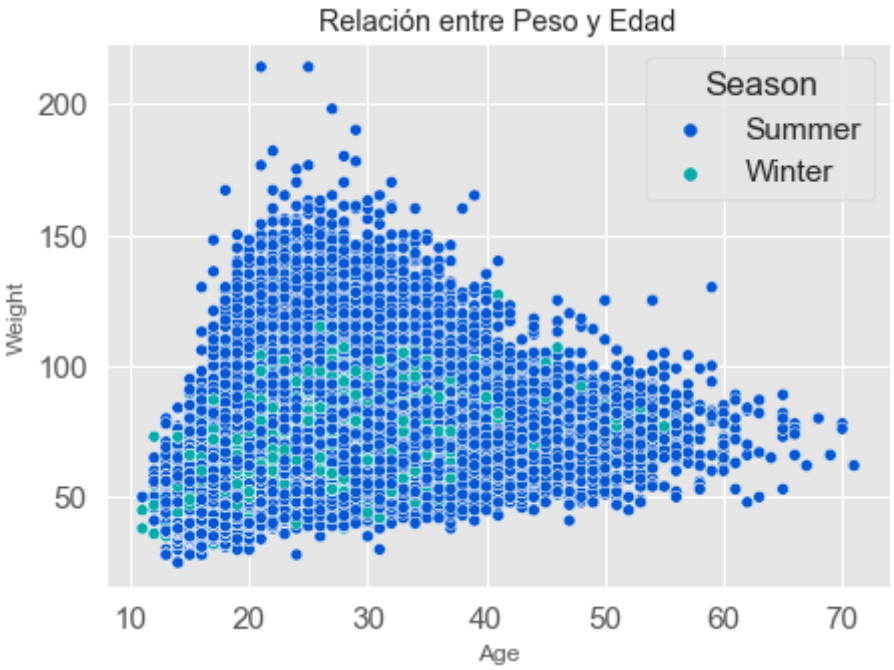


In [238...

```
plt.figure(figsize=(7,5))
plt.title('Relación entre Peso y Edad')
sns.scatterplot(data = df_atletas_ok, x = "Age", y = "Weight", hue="Season",palette=
```

Out[238…

```
<AxesSubplot:title={'center':'Relación entre Peso y Edad'}, xlabel='Age', ylabel='We
ight'>
```

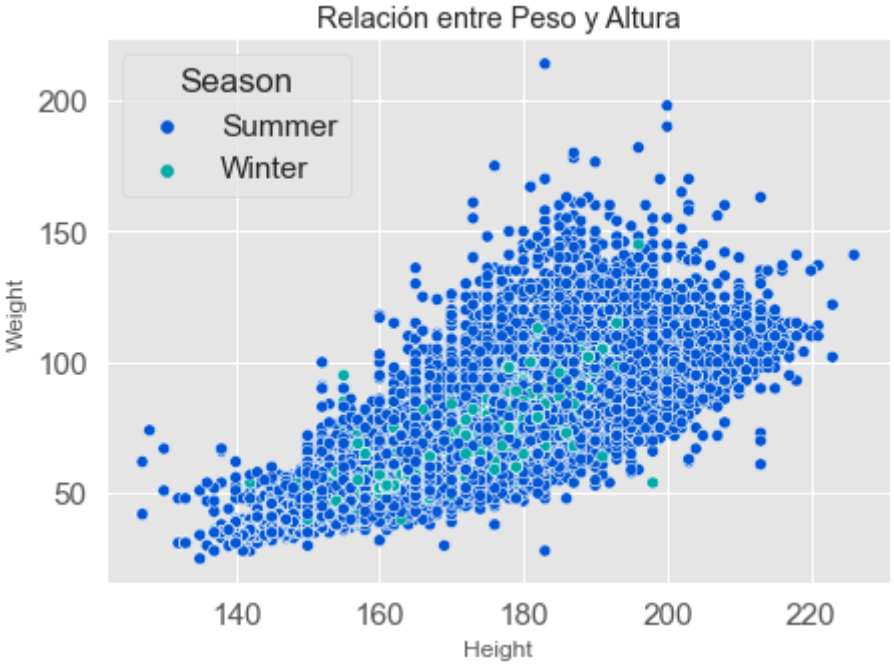


In [239…

```
plt.figure(figsize=(7,5))
plt.title('Relación entre Peso y Altura')
sns.scatterplot(data = df_atletas_ok, x = "Height", y = "Weight", hue="Season",palet
```

Out[239…

```
<AxesSubplot:title={'center':'Relación entre Peso y Altura'}, xlabel='Height', ylabe
l='Weight'>
```



In [153…

```
df=df_atletas_ok[["Age","Height","Weight"]]
```

In [154…

```
corr = df.corr()
corr.style.background_gradient(cmap ='Blues')
```

Out[154...

	Age	Height	Weight
Age	1.000000	0.141684	0.212041
Height	0.141684	1.000000	0.796573
Weight	0.212041	0.796573	1.000000

In [235...

```
# Cálculo de correlación con Pandas
# ==============================================================================
print('Correlación Pearson: ', df['Weight'].corr(df['Height'], method='pearson'))
print('Correlación Spearman: ', df['Weight'].corr(df['Height'], method='spearman'))
print('Correlación kendall: ', df['Weight'].corr(df['Height'], method='kendall'))
```

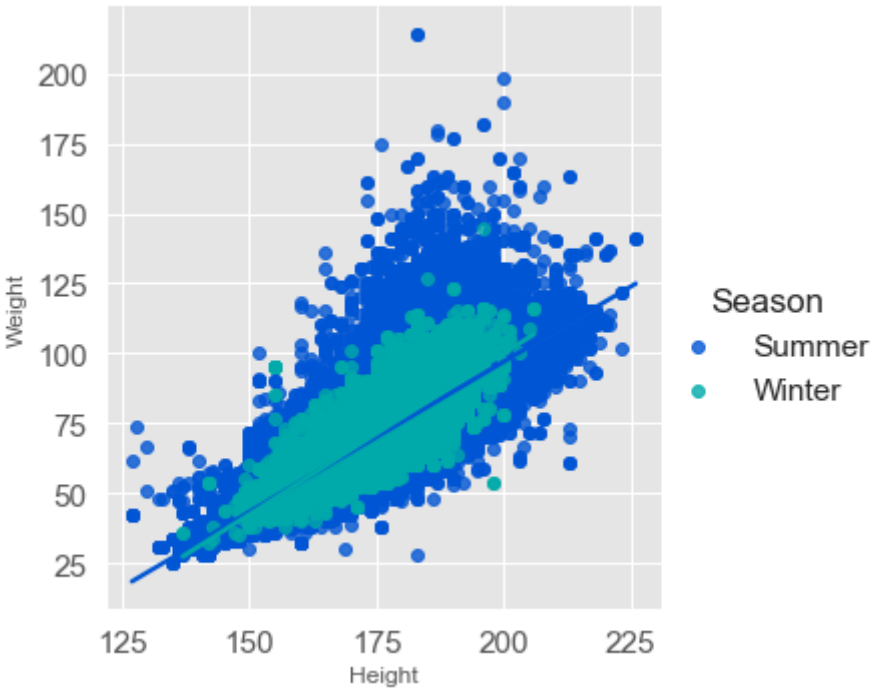
```
Correlación Pearson:  0.7965725794977142
Correlación Spearman:  0.8275000683945971
Correlación kendall:  0.6523525848252549
```

In [236...

```
sns.lmplot(x = "Height", y = "Weight", data = df_atletas_ok,hue = "Season",palette="
```

Out[236...

```
<seaborn.axisgrid.FacetGrid at 0x1c192e11d00>
```



In [157...

```
# Correlación lineal entre las dos variables Altura y Peso
# ==============================================================================
corr_test = pearsonr(x = df_atletas_ok['Height'], y =  df_atletas_ok['Weight'])
print("Coeficiente de correlación de Pearson: ", corr_test[0])
print("P-value: ", corr_test[1])
```

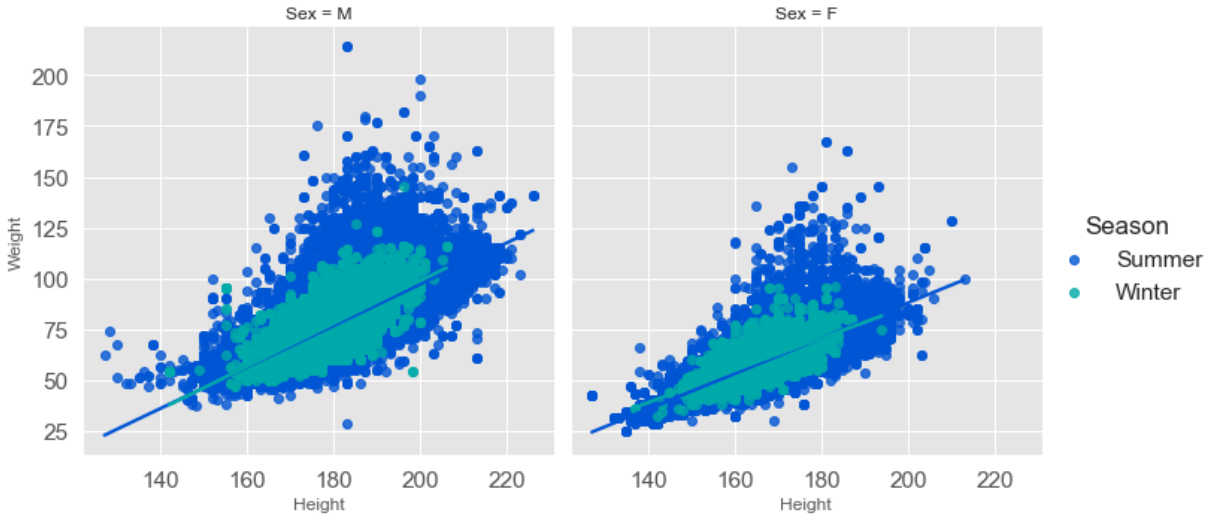
```
Coeficiente de correlación de Pearson:  0.7965725794977134
P-value:  0.0
```

In [240...

```
sns.lmplot(x = "Height", y = "Weight", data = df_atletas_ok,hue = "Season", col = "S
```

Out[240...

```
<seaborn.axisgrid.FacetGrid at 0x1c19718ceb0>
```



In [247…
```
male= df_atletas_ok.loc[:,'Sex'] == 'M'
df_male_H = df_atletas_ok.Height.loc[male]
df_male_W = df_atletas_ok.Weight.loc[male]
```

In [248…
```
female= df_atletas_ok.loc[:,'Sex'] == 'F'
df_female_H = df_atletas_ok.Height.loc[female]
df_female_W = df_atletas_ok.Weight.loc[female]
```

In [254…
```
# Correlación lineal entre las dos variables Altura y Peso, Sex= M
# ================================================================
altura= df_male_H
peso = df_male_W
corr_test = pearsonr(x = altura, y = peso)
print("Coeficiente de correlación de Pearson: ", corr_test[0])
```

```
Coeficiente de correlación de Pearson:  0.7269637061144824
```

In [253…
```
# Correlación lineal entre las dos variables Altura y Peso, Sex= F
# ================================================================
altura= df_female_H
peso = df_female_W
corr_test = pearsonr(x = altura, y = peso)
print("Coeficiente de correlación de Pearson: ", corr_test[0])
```

```
Coeficiente de correlación de Pearson:  0.7400848431796769
```

Nivell 2

Exercici 3

Continuant amb les dades de tema esportiu, calcula la correlació de tots els atributs entre sí i representa'ls en una matriu amb diferents colors d'intensitat.

In [161…
```
df=df_atletas_ok[["Age","Height","Weight","Year","Season","Sex","Medal"]]
```

In [162…
```
df.head(5)
```

Out[162...

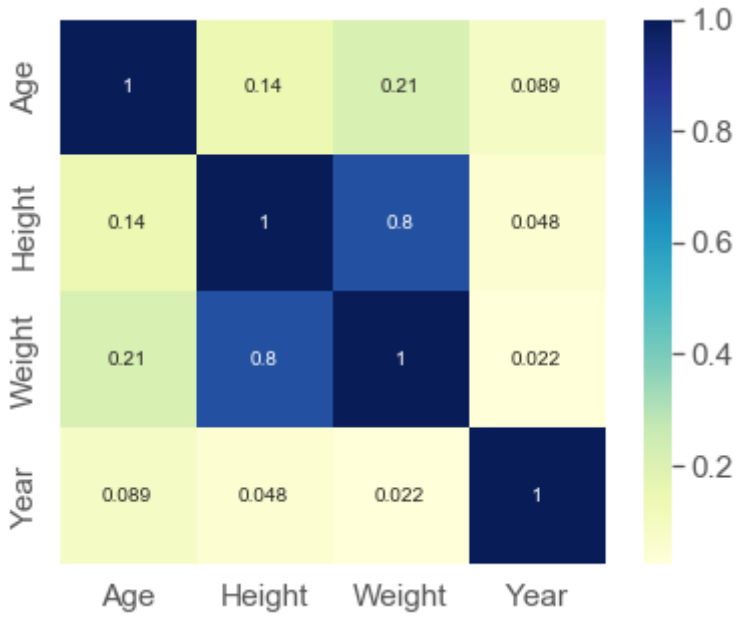
	Age	Height	Weight	Year	Season	Sex	Medal
0	24.0	180.0	80.0	1992	Summer	M	0
1	23.0	170.0	60.0	2012	Summer	M	0
4	21.0	185.0	82.0	1988	Winter	F	0
5	21.0	185.0	82.0	1988	Winter	F	0
6	25.0	185.0	82.0	1992	Winter	F	0

In [181...

```
plt.figure(figsize=(7,5))
sns.heatmap(df.corr(),cmap="YlGnBu", square = True,annot=True)
```

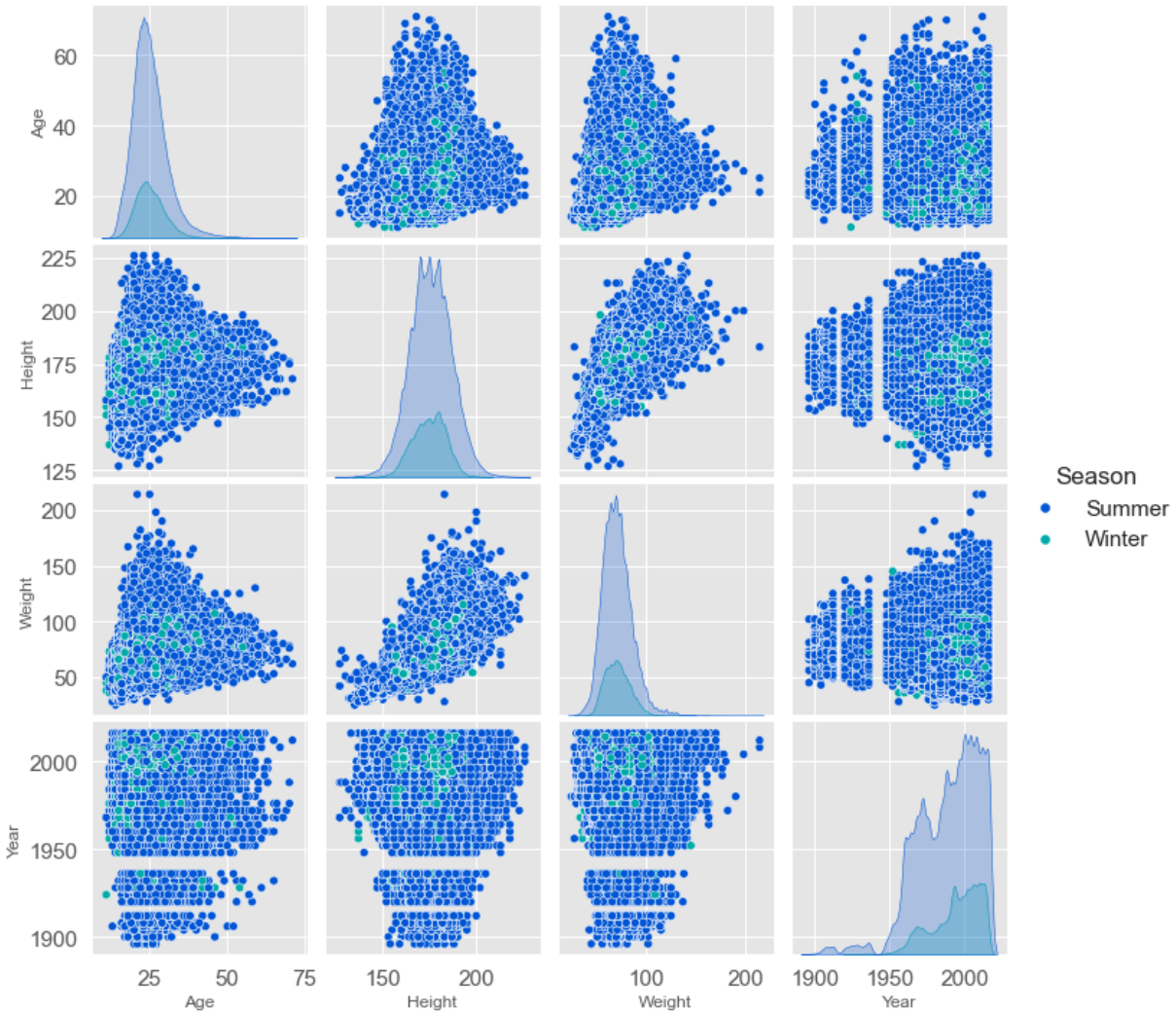
Out[181...

```
<AxesSubplot:>
```



In [191...

```
sns.pairplot(df, hue="Season",  palette= "winter");
```



Como se puede observar en el mapa de correlaciones y en el gráfico por pares, la correlación lineal más evidente es la que se dá entre Altura y peso de los atletas

No existe correlación lineal elevada entre las variables Edad con ALtura ó con Peso

Tampoco existe ninguna correlación entre Año y ninguna de las variables númericas usadas en este ejercicio: Edad, Altua ó Peso

Nivell 3

Exercici 4

Continuant amb les dades de tema esportiu, selecciona un atribut i calcula la mitjana geomètrica i la mitjana harmònica.

4.1 Media Aritmética

La media aritmética es un tipo de media que otorga la misma ponderación a todos los valores y se obtiene con la suma de un conjunto de valores dividida entre el número total de sumandos.

In [265…

```
print("Media Aritmética de la variable -Weight (en Kg)- del total de atletas: ", rou
```

```
Media Aritmética de la variable -Weight (en Kg)- del total de atletas:  70.688
```

4.2 Media Geométrica

La media geométrica es un tipo de media que se calcula como la raíz del producto de un conjunto de números estrictamente positivos.

In [266…
```
print("Media Geométrica de la variable -Weight (en Kg)- del total de atletas: ",roun
```

```
Media Geométrica de la variable -Weight (en Kg)- del total de atletas:  69.29
```

4.3 Media Armónica

La media armónica es igual al número de elementos de un grupo de cifras entre la suma de los inversos de cada una de estas cifras. En otras palabras, la media armónica es una medida estadística recíproca a la media aritmética.

In [267…
```
print("Media Armónica de la variable -Weight (en Kg)- del total de atletas: ", round
```

```
Media Armónica de la variable -Weight (en Kg)- del total de atletas:  67.916
```