# 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 [50]:
```
# 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 [51]:
```
df_atletas= pd.read_csv(r"C:\Users\hecto\OneDrive\Documentos\IT Data Science\Sprint5
```

In [52]:
```
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 [53]:
```
df_atletas_pre = df_atletas
df_atletas_pre.fillna(0, inplace=True)
```

In [54]:
```
df_atletas_ok = df_atletas_pre[(df_atletas_pre['Age']>0)&(df_atletas_pre['Height']>0
```

In [55]:
```
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 [56]:
```
df_atletas_ok.head(5)
```

Out[56]:

| | 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 [57]:
```
df_atletas_ok.describe()
```

Out[57]:

| | 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 [58]:
```
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[58]:

| | 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 |

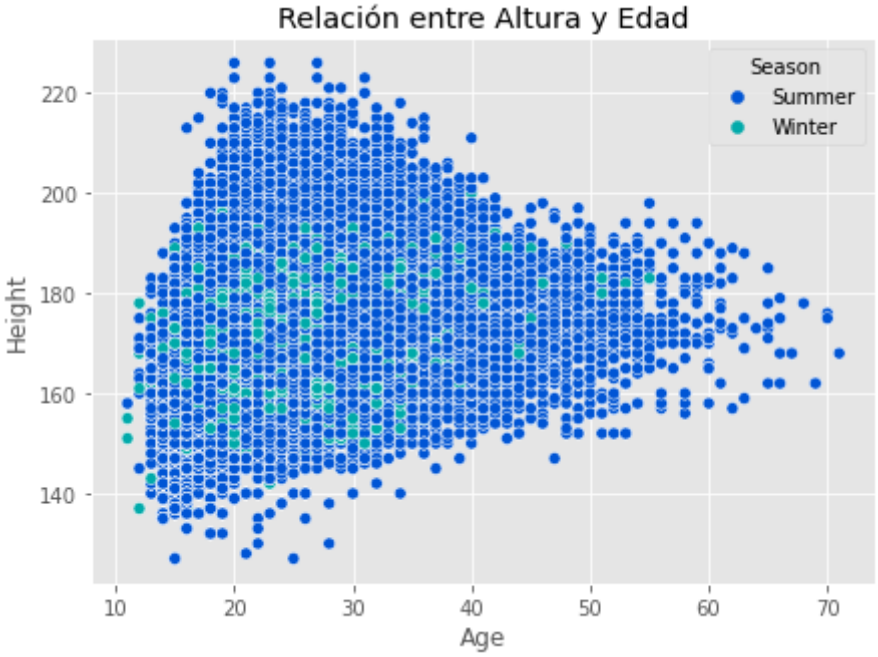
# Exercici 2

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

### 2.1 Relación entre Altura y Edad de los Atletas

In [59]:
```
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[59]:
```
<AxesSubplot:title={'center':'Relación entre Altura y Edad'}, xlabel='Age', ylabel
='Height'>
```



In [60]:
```
df=df_atletas_ok[["Age","Height","Weight"]]
```

In [61]:
```
# Cálculo de correlación con Pandas
# ================================================================================
print('Correlación Pearson: ', df['Height'].corr(df['Age'], method='pearson'))
print('Correlación Spearman: ', df['Height'].corr(df['Age'], method='spearman'))
print('Correlación kendall: ', df['Height'].corr(df['Age'], method='kendall'))
```

```
Correlación Pearson:  0.14168449010056783
Correlación Spearman:  0.1480008001470008
Correlación kendall:  0.10314825236519815
```

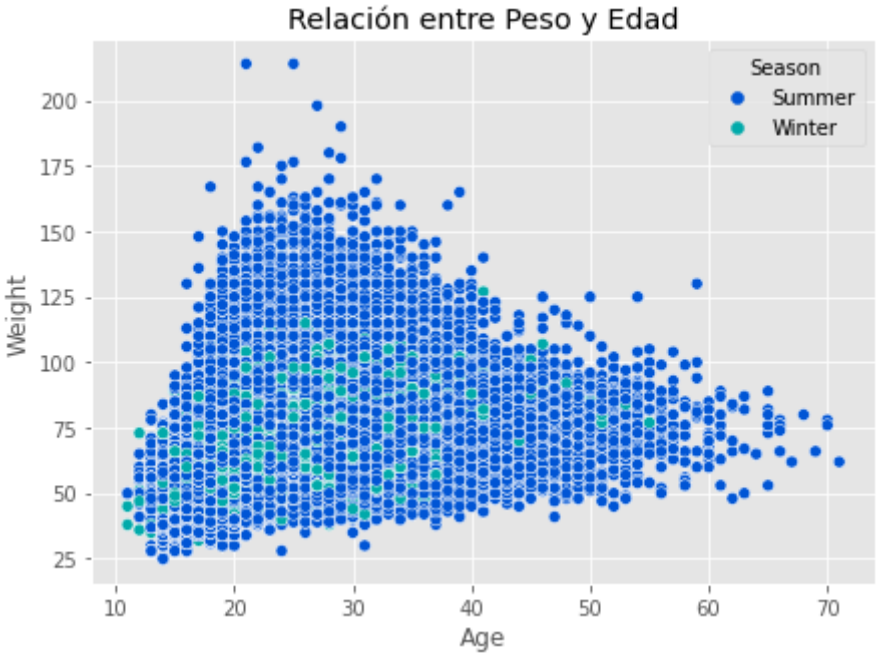
**A partir del gráfico no se desprende una correlación lineal evidente entre las variables, Altura y Edad de los atletas**

**Se ha verificado con el cálculo de diferentes coeficientes de correlación, que efectivamente la correlación es débil ente estas dos variables, con coeficientes de correlación de Pearson del 1.142**

## 2.2 Relación entre el Peso y la Edad de los Atletas

In [62]:
```
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[62]:
```
<AxesSubplot:title={'center':'Relación entre Peso y Edad'}, xlabel='Age', ylabel='We
ight'>
```




In [63]:
```
# Cálculo de correlación con Pandas
# ================================================================================
print('Correlación Pearson: ', df['Weight'].corr(df['Age'], method='pearson'))
print('Correlación Spearman: ', df['Weight'].corr(df['Age'], method='spearman'))
print('Correlación kendall: ', df['Weight'].corr(df['Age'], method='kendall'))
```

```
Correlación Pearson:  0.2120407215342155
Correlación Spearman:  0.2169521776600894
Correlación kendall:  0.15265704686093168
```

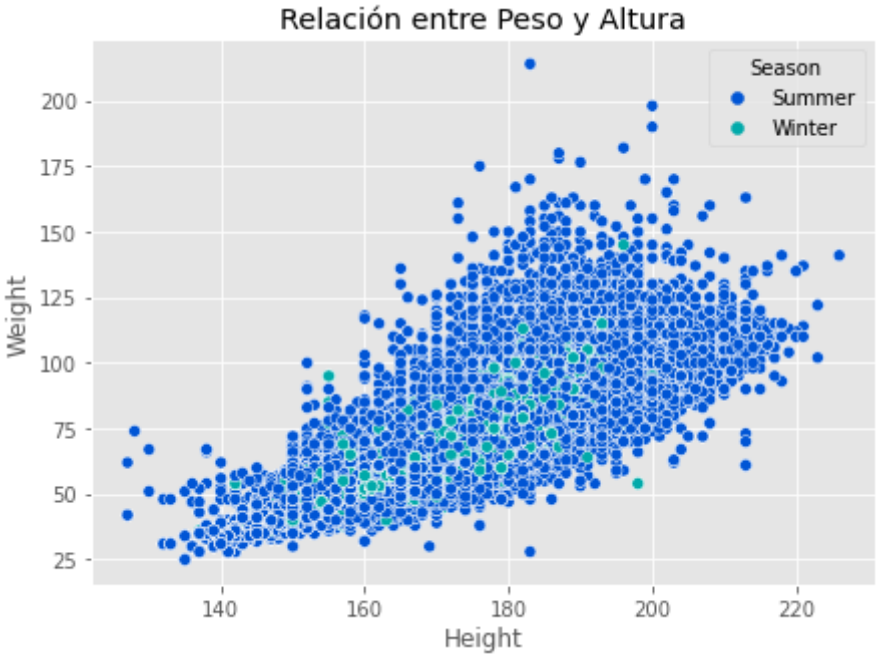
**Como en el caso anterior a partir del Gráfico que muestra la relación entre las variables, Peso y Edad de los atletas, no parece existir una clara correlación lineal entre ambas.**

El cálculo de los diferentes coeficientes de correlación, muestra que, aunque si parece existir una mayor relación entre estas variables que el el caso anterio, el coeficiente de correlación de Pearson p.e. es de 0.212 por lo cual no podemos inferir que exista una clara correlación lineal entre las variablea Peso y Edad.

## 2.3 Relación Entre Peso y Altura

In [64]:
```
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[64]:
```
<AxesSubplot:title={'center':'Relación entre Peso y Altura'}, xlabel='Height', ylabe
l='Weight'>
```




In [65]:
```
# 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
```

A partir del Gráfico que muestra la relación entre las variables, Peso y Altura de los atletas, se observa una clara correlación lineal entre ambas variables.

El cálculo de los diferentes coeficientes de correlación, muestra que sí existe una mayor corelación lineal entre estas variables dos variables, que ne los casos anteriores.

El coeficiente de correlación de Pearson correspondiente es de 0.796, que valida la correlación lineal entre las variablea Peso y Altura.

## 2.4 Resumen de Coeficientes de Correlación de Pearson de las variables: Peso, Altura y Edad

In [66]:
```
corr = df.corr()
corr.style.background_gradient(cmap ='Blues')
```

Out[66]:

| | Age | Height | Weight |
|---|---|---|---|
| **Age** | 1.000000 | 0.141684 | 0.212041 |

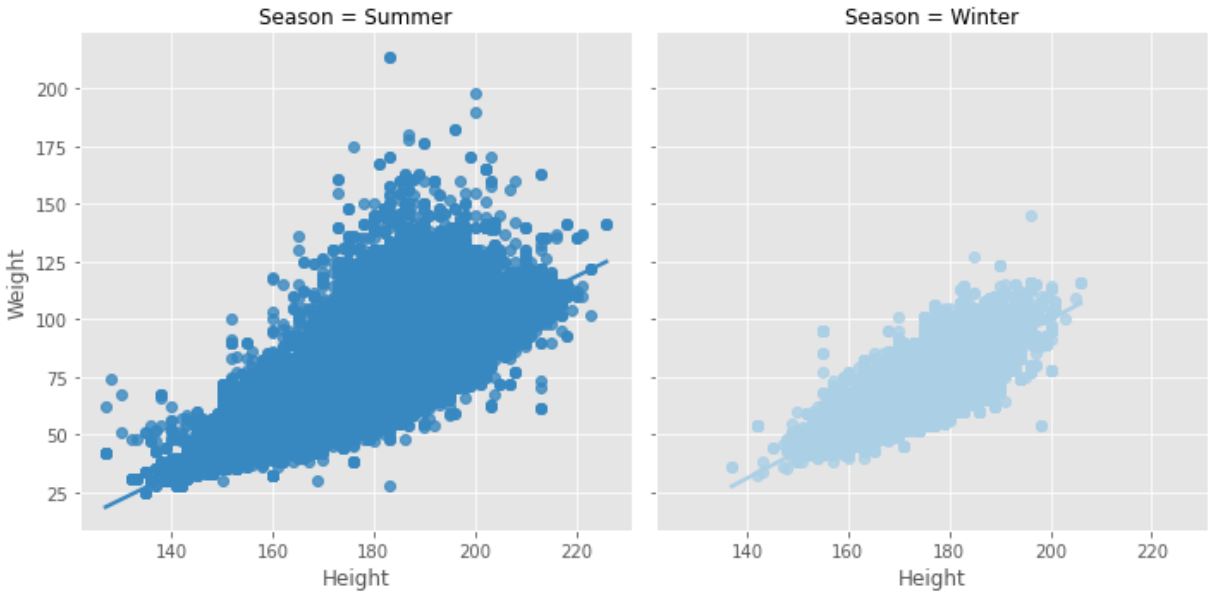
| | Age | Height | Weight |
|---|---|---|---|
| **Height** | 0.141684 | 1.000000 | 0.796573 |
| **Weight** | 0.212041 | 0.796573 | 1.000000 |

Como se refleja en el cuadro, las variables má correlacionadas son Altura y Peso, mientras que no existe una elevada correlación entre la Edad de los atletas y las variables de Altura ó Peso.

## 2.5 Relación entre Peso y Altura de los Atletas por "Season" (Juegos de Invierno o Juegos de Verano)

In [67]:
```
sns.lmplot(x = "Height", y = "Weight", data = df_atletas_ok, hue="Season" , col = "S
```

Out[67]:
```
<seaborn.axisgrid.FacetGrid at 0x22aaa4d3ac0>
```




In [68]:
```
summer= df_atletas_ok.loc[:,'Season'] == 'Summer'
df_summer_H = df_atletas_ok.Height.loc[summer]
df_summer_W = df_atletas_ok.Weight.loc[summer]
```

In [69]:
```
# Correlación lineal entre las dos variables Altura y Peso, Season= Summer
# ========================================================================
altura= df_summer_H
peso = df_summer_W
corr_test = pearsonr(x = altura, y = peso)
print("Coeficiente de correlación de Pearson: ", corr_test[0])
```

```
Coeficiente de correlación de Pearson:  0.7951829879446266
```

In [70]:
```
winter= df_atletas_ok.loc[:,'Season'] == 'Winter'
df_winter_H = df_atletas_ok.Height.loc[winter]
df_winter_W = df_atletas_ok.Weight.loc[winter]
```

In [71]:
```
# Correlación lineal entre las dos variables Altura y Peso, Season= Winter
# ========================================================================
altura= df_winter_H
peso = df_winter_W
corr_test = pearsonr(x = altura, y = peso)
print("Coeficiente de correlación de Pearson: ", corr_test[0])
```

```
Coeficiente de correlación de Pearson:  0.810605631100543
```

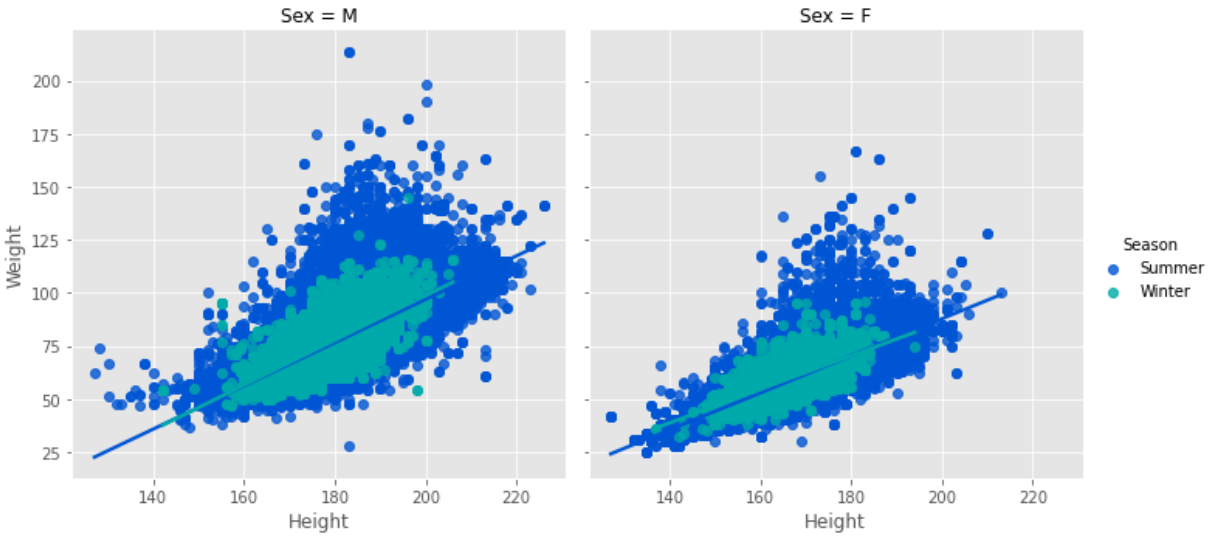
**Existe también una elevada correlación lineal de las variables Altura y Peso, cuando se desglosan los datos en función de si los juegos corresponden a los Juegos Olimpicos de Verano o de Invierno.**

**El coeficiente de Pearson de esas dos variables es de 0.795 para los atletas de los Juegos de Verano y de 0.8106 en el caso de los datos referidos a los atletas de los Juegos de invierno, por tanto existe mayor correlación lineal de las variables Altura y Peso entre los atletas que compitieron en los Juegos de Invierno.**

## 2.6 Relación entre Peso y Altura de los atletas por Sex (Femenino y Masculino)

In [72]:
```
sns.lmplot(x = "Height", y = "Weight", data = df_atletas_ok,hue = "Season", col = "S
```

Out[72]:
```
<seaborn.axisgrid.FacetGrid at 0x22aaa4e50d0>
```




In [73]:
```
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 [74]:
```
# 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 [75]:
```
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 [76]:
```
# 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
```

Existe también una elevada correlación lineal de las variables Altura y Peso, cuando se desglosan los datos en función del la variable Sexo de los atletas.

El coeficiente de Pearson de esas dos variables es de 0.727 para los atletas de "Sex" = Masculino, y un poco superior, 0.740 en el caso de los datos referidos a las atletas femeninas.

# 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 [108…
```
df=df_atletas_ok[['Sex','Age','Height','Weight','Year','Season','Medal']]
```

In [109…
```
df.head(5)
```

Out[109…

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

In [117…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 206165 entries, 0 to 271115
Data columns (total 9 columns):
 #   Column         Non-Null Count   Dtype  
---  ------         --------------   -----  
 0   Age            206165 non-null  float64
 1   Height         206165 non-null  float64
 2   Weight         206165 non-null  float64
 3   Year           206165 non-null  int64  
 4   Sex_M          206165 non-null  uint8  
 5   Season_Winter  206165 non-null  uint8  
 6   Medal_Bronze   206165 non-null  uint8  
 7   Medal_Gold     206165 non-null  uint8  
 8   Medal_Silver   206165 non-null  uint8  
dtypes: float64(3), int64(1), uint8(5)
memory usage: 16.9 MB
```

In [110…
```
df=pd.get_dummies(df, columns=["Sex"],drop_first = True)
```

In [111…
```
df = pd.get_dummies(df, columns = ["Season"],drop_first = True)
```

In [112…
```
df=pd.get_dummies(df, columns = ["Medal"],drop_first = True)
```

In [113…

```
df.head(5)
```

Out[113...

| | Age | Height | Weight | Year | Sex_M | Season_Winter | Medal_Bronze | Medal_Gold | Medal_Silver |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 24.0 | 180.0 | 80.0 | 1992 | 1 | 0 | 0 | 0 | 0 |
| **1** | 23.0 | 170.0 | 60.0 | 2012 | 1 | 0 | 0 | 0 | 0 |
| **4** | 21.0 | 185.0 | 82.0 | 1988 | 0 | 1 | 0 | 0 | 0 |
| **5** | 21.0 | 185.0 | 82.0 | 1988 | 0 | 1 | 0 | 0 | 0 |
| **6** | 25.0 | 185.0 | 82.0 | 1992 | 0 | 1 | 0 | 0 | 0 |

**Hemos transformado las variables "Sex", "Season" y "Medal" de categóricas a numéricas para poder extraer información sobre la correlación con las varialbes de Altura y Peso.**

**En el Caso de "Sex", se ha transformado en la variable "Sex_M" que tiene los valores 0 y 1, de forma que los atletas masculinos se corresponden con el 1 y las atletas femeninas con el 0.**

**La variable "Season", se ha transformado en "Season_Winter" con los valores 0 en el caso de los Juegos de Verano y 1 para los Juegos de Invierno.**

**La vaiable "Medal" se ha desglosado en tres variables "Medal_Gold", "Medal_Silver" y "Medal_Bronze" para realizar un análisis mas exhaustivo de las correlaciones entre las diferentes variables y los atletas obtuvieron medallas olímpicas.**

In [115...

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 206165 entries, 0 to 271115
Data columns (total 9 columns):
 #   Column         Non-Null Count   Dtype  
---  ------         --------------   -----  
 0   Age            206165 non-null  float64
 1   Height         206165 non-null  float64
 2   Weight         206165 non-null  float64
 3   Year           206165 non-null  int64  
 4   Sex_M          206165 non-null  uint8  
 5   Season_Winter  206165 non-null  uint8  
 6   Medal_Bronze   206165 non-null  uint8  
 7   Medal_Gold     206165 non-null  uint8  
 8   Medal_Silver   206165 non-null  uint8  
dtypes: float64(3), int64(1), uint8(5)
memory usage: 16.9 MB
```

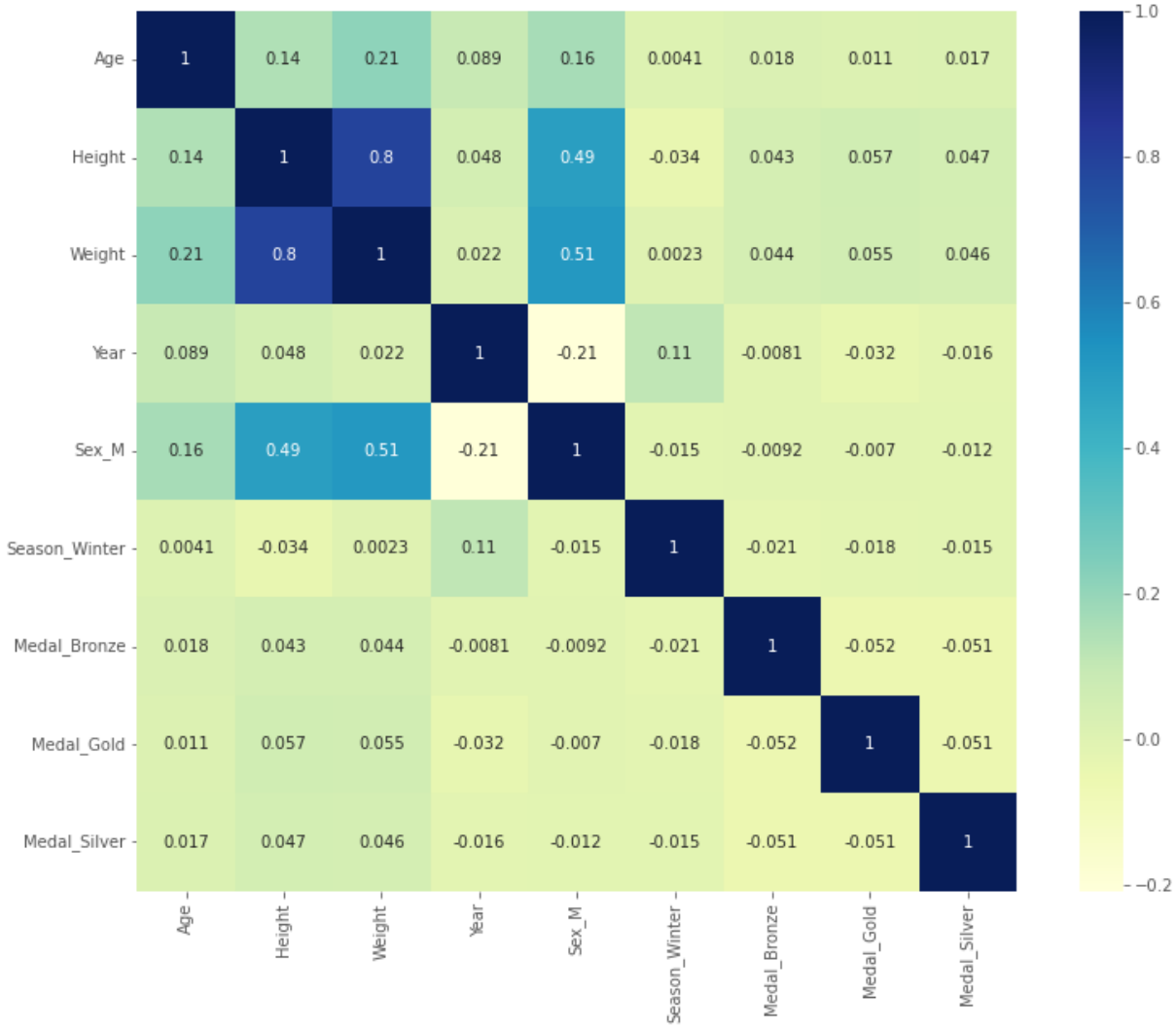
## 3.1 Mapa de Colores de correlaciones entre diferentes atributos

In [114...

```
plt.figure(figsize=(14,10))
sns.heatmap(df.corr(),cmap="YlGnBu", square = True,annot=True)
#sns.heatmap(df_atletas_ok[['Age', 'Height', 'Weight', 'Year','Season', 'Medal']].co
```

Out[114...

```
<AxesSubplot:>
```




**Como se puede observar en el mapa de correlaciones, la correlación lineal más evidente es la que se produce 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 lineal entre Año de las olimpiadas y ninguna de las variables númericas usadas en este ejercicio: Edad, Altura ó Peso.**

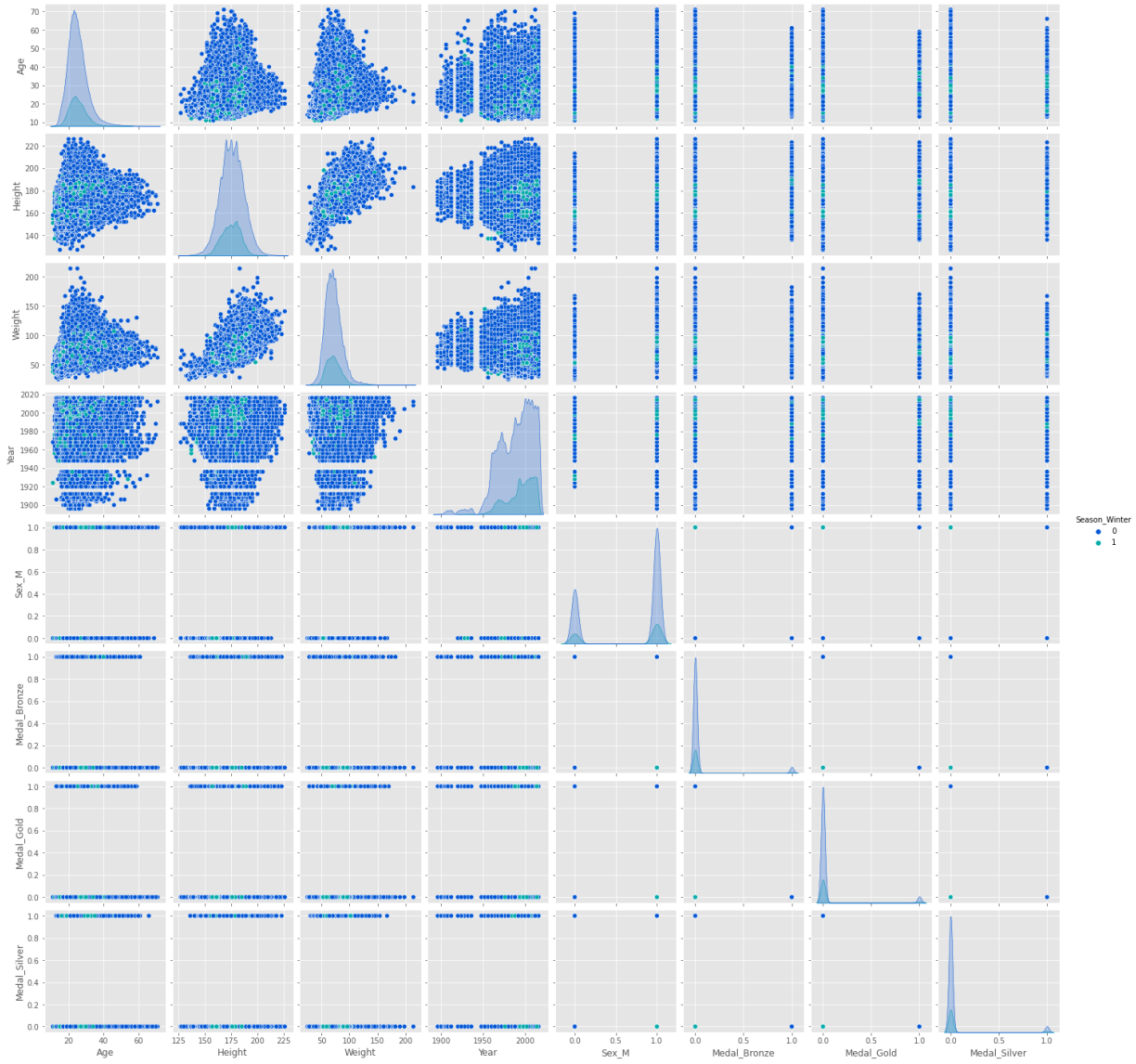
**Sin embargo si parece que existe cierta correlación entre el Sexo y las variables Altura ó Peso, con coeficientes de correlación del 0.49 y 0.51 respectivamente.**

**No existe correlación evidente entrel las variables Altura ó Peso con el resto de variables categóricas tranformadas: Season ó Medal.**

## 3.2 Grafico por pares de la distribución de las variables transformadas

In [119…

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




En el grafico por pares se refleja lo que ya se anticipaba en el mapa de color, donde las variables que presentan realmente una correlación lineal significativa son Altura y Peso.

Sin embargo no se puede observar a simple vista, la correlación establecida entre la variable transformada Sex_M y la Altua ó el Peso (Height & Weight) ya que los puntos se concentran en los valores 0 y 1, pero parece que existe una distribución similar de los datos de altura y peso entre hombres y mujeres, que justifica el coeficiente de correlación reseñado en el apartado anterior.

# 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 [125…

```
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 [126…
```
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 [127…
```
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
```