MÓDULO 1: HERRAMIENTAS BIG DATA

TEMA 4.2. HERRAMIENTAS DE ANALISIS: PROGRAMACIÓN EN PYTHON - EJERCICIOS

Ferran Carrascosa Mallafrè

Licenciado en Matemáticas por la Universidad de Barcelona. Data Scientist

Table of Contents

[Table of Contents 1](#_Toc51490041)

[Ejercicio 1 2](#_Toc51490042)

[Ejercicio 1.1. 3](#_Toc51490043)

[Ejercicio 1.2. 3](#_Toc51490044)

[Ejercicio 1.3. 3](#_Toc51490045)

[Ejercicio 1.4. 3](#_Toc51490046)

[Ejercicio 1.5. 3](#_Toc51490047)

[Ejercicio 1.6. 3](#_Toc51490048)

[Ejercicio 1.7. 4](#_Toc51490049)

[Ejercicio 1.8. 4](#_Toc51490050)

[Ejercicio 1.9. 4](#_Toc51490051)

[Ejercicio 1.10. 4](#_Toc51490052)

[Ejercicio 2 4](#_Toc51490053)

[Ejercicio 2.1. 4](#_Toc51490054)

[Ejercicio 2.2. 5](#_Toc51490055)

[Ejercicio 2.3. 5](#_Toc51490056)

[Ejercicio 2.4. 5](#_Toc51490057)

[Ejercicio 2.5. 5](#_Toc51490058)

[Ejercicio 2.6. 5](#_Toc51490059)

[Ejercicio 2.7. 5](#_Toc51490060)

[Ejercicio 2.8. 6](#_Toc51490061)

[Ejercicio 2.9. 6](#_Toc51490062)

[Ejercicio 2.10. 6](#_Toc51490063)

[**Abre en Colab**](https://colab.research.google.com/github/griu/mbdds_fc20/blob/master/Python/modulo1_tema4_Py_Ejercicio_1.ipynb)

# Ejercicio 1

Para realizar el ejercicio cargamos los datos de *Especies* en STARWARS SWAPI y las librerías principales.

import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns; sns.set() # para el estilo de graficos  
  
entidades = ['planets','starships','vehicles','people','species']  
entidades\_df = {x: pd.read\_pickle('www/' + x + '\_df.pkl') for x in entidades}  
  
# Datos principales  
people\_df = entidades\_df['people'][["height","mass","birth\_year","gender","homeworld"]].dropna()  
  
people\_df

## height ... homeworld  
## name ...   
## Luke Skywalker 172.0 ... http://swapi.dev/api/planets/1/  
## C-3PO 167.0 ... http://swapi.dev/api/planets/1/  
## R2-D2 96.0 ... http://swapi.dev/api/planets/8/  
## Darth Vader 202.0 ... http://swapi.dev/api/planets/1/  
## Leia Organa 150.0 ... http://swapi.dev/api/planets/2/  
## Owen Lars 178.0 ... http://swapi.dev/api/planets/1/  
## Beru Whitesun lars 165.0 ... http://swapi.dev/api/planets/1/  
## Biggs Darklighter 183.0 ... http://swapi.dev/api/planets/1/  
## Obi-Wan Kenobi 182.0 ... http://swapi.dev/api/planets/20/  
## Anakin Skywalker 188.0 ... http://swapi.dev/api/planets/1/  
## Chewbacca 228.0 ... http://swapi.dev/api/planets/14/  
## Han Solo 180.0 ... http://swapi.dev/api/planets/22/  
## Greedo 173.0 ... http://swapi.dev/api/planets/23/  
## Jabba Desilijic Tiure 175.0 ... http://swapi.dev/api/planets/24/  
## Wedge Antilles 170.0 ... http://swapi.dev/api/planets/22/  
## Yoda 66.0 ... http://swapi.dev/api/planets/28/  
## Palpatine 170.0 ... http://swapi.dev/api/planets/8/  
## Boba Fett 183.0 ... http://swapi.dev/api/planets/10/  
## IG-88 200.0 ... http://swapi.dev/api/planets/28/  
## Bossk 190.0 ... http://swapi.dev/api/planets/29/  
## Lando Calrissian 177.0 ... http://swapi.dev/api/planets/30/  
## Lobot 175.0 ... http://swapi.dev/api/planets/6/  
## Ackbar 180.0 ... http://swapi.dev/api/planets/31/  
## Wicket Systri Warrick 88.0 ... http://swapi.dev/api/planets/7/  
## Qui-Gon Jinn 193.0 ... http://swapi.dev/api/planets/28/  
## Padmé Amidala 185.0 ... http://swapi.dev/api/planets/8/  
## Jar Jar Binks 196.0 ... http://swapi.dev/api/planets/8/  
## Darth Maul 175.0 ... http://swapi.dev/api/planets/36/  
## Ayla Secura 178.0 ... http://swapi.dev/api/planets/37/  
## Mace Windu 188.0 ... http://swapi.dev/api/planets/42/  
## Ki-Adi-Mundi 198.0 ... http://swapi.dev/api/planets/43/  
## Plo Koon 188.0 ... http://swapi.dev/api/planets/49/  
## Luminara Unduli 170.0 ... http://swapi.dev/api/planets/51/  
## Barriss Offee 166.0 ... http://swapi.dev/api/planets/51/  
## Dooku 193.0 ... http://swapi.dev/api/planets/52/  
## Jango Fett 183.0 ... http://swapi.dev/api/planets/53/  
##   
## [36 rows x 5 columns]

## Ejercicio 1.1.

Construye un gráfico de dispersión de los personajes donde se visualice: la altura (height), el peso (mass), la edad en años BBY (birth\_year) y el género (gender). Para ello utiliza la función sns.scatterplot() de la librería seaborn. Aprovecha los parámetros: x, y, size, hue y style (consulta la ayuda de la función [.scatterplot()](https://seaborn.pydata.org/generated/seaborn.scatterplot.html).

# Solución:

## Ejercicio 1.2.

Sobre el gráfico del ejercicio 1.1:

* Pon título al gráfico y a los ejes x e y.
* Modifica los límites del eje y para que aparezcan sólo personajes de menos de 150 Kg de peso.
* Sitúa en el gráfico los nombres de “Darth Vader” y “Anakin Skywalker”. ¿Cómo es posible tengan un peso y altura tan distintos si eran la misma persona?

# Solución:

## Ejercicio 1.3.

Utiliza las *list comprehension* para calcular el cuadrado de los valores positivos de la siguiente lista:

Muestra el resultado por pantalla.

val = [5, 6, -1, 2, -3, -7, 9, 1]

# Solución:

## Ejercicio 1.4.

Construye un diccionario donde se identifique, mediante claves y valores, las siguientes características del personaje Yoda: “nombre”, “altura”, “peso”, “edad” y “genero”. Utiliza los datos de people\_df.

Muestra el diccionario por pantalla.

# Solución:

## Ejercicio 1.5.

Calcula, a partir de los vectores numpy de altura y peso, definidos a continuación, el [índice de masa corporal (IMC)](https://es.wikipedia.org/wiki/%C3%8Dndice_de_masa_corporal) de los personajes de star wars contenidos en people\_df:

donde altura está en metros y el peso en kg.

Muestra los datos por pantalla.

altura = people\_df.height.values  
peso = people\_df.mass.values

# Solución:

## Ejercicio 1.6.

A partir del IMC que has calculado en el ejercicio 1.5. Construye un panel con dos histogramas:

* Un histograma con toda la muestra
* Un histograma seleccionando los valores con un IMC inferior a 100.

# Solución:

## Ejercicio 1.7.

A partir del vector 1 y 2 que se definen a continuación contesta las siguientes preguntas:

* Calcula el shape, ndim, size del vector1 y vector2
* Explica cual es la diferencia entre vector1 y vector2 a partir de los que hayas observado

vector1 = np.hstack([altura,peso])  
vector2 = np.vstack([altura,peso])

# Solución:

## Ejercicio 1.8.

Crea una copia de people\_df llamada personajes\_df y renombra las columnas con su traducción al castellano.

Truco: Puedes renombrar el índice con:

personajes\_df.index = personajes\_df.index.rename("nombre")

Muestra los 5 primeros registros del nuevo data frame con .head().

# Solución:

## Ejercicio 1.9.

Haz el mismo cálculo de IMC que has hecho en 1.5. pero directamente sobre el objeto personajes\_df.

Ordena el data frame de mayor a menor IMC y muestra los personajes con IMC por encima de 30.

¿Sabias que IMC por encima de 30 se considera obeso?

# Solución:

## Ejercicio 1.10.

Inserta un valor faltante en los valores de IMC que sean superiores a 100 y dibuja el histograma con .plot.hist().

# Solución:

# Ejercicio 2

Para el ejercicio 2, añadimos los datos del ejercicio 1 los datos de planetas.

# planetas  
planets\_df = entidades\_df['planets'][["orbital\_period","url"]].dropna()  
planets\_df.head()

## orbital\_period url  
## name   
## Tatooine 304.0 http://swapi.dev/api/planets/1/  
## Alderaan 364.0 http://swapi.dev/api/planets/2/  
## Yavin IV 4818.0 http://swapi.dev/api/planets/3/  
## Hoth 549.0 http://swapi.dev/api/planets/4/  
## Dagobah 341.0 http://swapi.dev/api/planets/5/

## Ejercicio 2.1.

Construye una función que diga “buenos días”, “buenas tardes” o “buenas noches” en función de la hora del día.

Truco 1: Para testear la función haz que tenga un parametro de entrada que tenga como valor por defecto: datetime.now() (primero carga from datetime import datetime).

Truco 2: Puedes extraer la hora de un datetime con .hour.

# Solución:

## Ejercicio 2.2.

Elimina del data frame personajes\_df los personajes con valores faltantes en IMC y crea una nueva variable con el trameado de la variable IMC definido en la siguiente tabla:

|  |  |
| --- | --- |
| Categoría de nivel de peso | Intervalo del percentil |
| Bajo peso | < 18.5 |
| Normal | >= 18.5 y < 25 |
| Sobrepeso | >= 25 y <30 |
| Obeso | >= 30 |

Truco: utiliza pd.cut(..., right=False) y modifica las etiquetas con .cat.categories.

# Solución:

## Ejercicio 2.3.

Muestra las frecuencias de la nueva variable definida en 2.2.

# Solución:

## Ejercicio 2.4.

Calcula ahora un trameado de la edad en 5 grupos equiprobables.

Truco: Busca ayuda de la función [pd.qcut()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.qcut.html)

# Solución:

## Ejercicio 2.5.

Presenta la tabla cruzada de tramos de edad (ej. 2.4) por tramos de IMC (ej. 2.2).

¿Qué tramo de edad tiene un mayor número de personajes con Bajo peso?

# Solución:

## Ejercicio 2.6.

Calcula la media del IMC en cada tramo de edad del ejercicio 2.2.

Truco: En el mismo cálculo del IMC medio, calcula también, la mediana de edad de cada tramo de edad para que sea el eje x del gráfico de líneas.

# Solución:

## Ejercicio 2.7.

Presenta los datos del ej. 2.6. como un gráfico de líneas donde el eje x sea la edad y el eje y el IMC medio

# Solución:

## Ejercicio 2.8.

Calcula el cociente del IMC sobre la mediana del IMC de su tramo de edad (definidos en el ejercicio 2.2.) mediante la función groupby(…).apply(…).

truco: primero crea la función que devuelva: x / np.nanmedian(x).

Presenta los datos mediante un boxplot de la nueva variable: [pd.boxplot()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.boxplot.html).

# Solución:

## Ejercicio 2.9.

¿Cual es planeta con menor IMC medio de sus personajes? ¿Que personaje/s son de ese planeta?

# Solución:

## Ejercicio 2.10.

Convierte a datetime los siguientes strings con la función [datetime.strptime()](https://docs.python.org/3/library/datetime.html#strftime-strptime-behavior) (consulta la ayuda si es necesario) de la libreria datetime:

* “1 Enero, 2020”
* “15-feb.-2017”
* “20190701 22:30” # 1 de julio de 2019

Truco: Modifica la configuración regional:

from datetime import datetime  
import locale  
locale.setlocale(locale.LC\_TIME, 'es\_ES.UTF-8')

# Solución: