

```
In [2]: import pandas as pd
```

```
In [39]: #1 Cargo los datos en el dataframe
#Cargaria solo la columna pedida y filtrando lo maximo posi
ble directamente pero el enunciado pide que
#cargue el set de datos
informacion_ejercicio = pd.read_csv("Entrenamieto_ECI_2020.
csv")
```

```
In [27]: #2 Mostrar, para la región EMEA, los datos: ID, Territory,
Delivery_Quarter, Delivery_Year para los 10
#proyectos con mayor Total Amount

#nlargest retorna un dataframe con la misma cantidad de col
umnas, ordenado de mayor a menor
(informacion_ejercicio[informacion_ejercicio["Region"] == "
EMEA"].nlargest(10, ["Total_Amount"], "all")
[["ID", "T
erritory", "Delivery_Quarter", "Delivery_Year"]])
```

Out[27]:

	ID	Territory	Delivery_Quarter	Delivery_Year
<b>2179</b>	10808	None	Q2	2016
<b>3667</b>	8037	SE America	Q3	2017
<b>2806</b>	4909	UAE (Dubai)	Q1	2017
<b>1933</b>	10620	Kazakhstan	Q2	2016
<b>2957</b>	13709	Romania	Q2	2016
<b>598</b>	4865	France	Q4	2017
<b>7898</b>	15287	Belgium	Q1	2018
<b>1817</b>	10350	Romania	Q3	2017
<b>3684</b>	20848	Italy	Q4	2016
<b>5669</b>	17668	Germany	Q1	2017

```
In [44]: # Tanto para el punto 2 como el 3 busqué alguna forma más e
# ficiente que el indexado booleano para buscar líneas
# que cumplan alguna condición, así no se recorre el data f
# rame 2 veces y además no se guarda memoria para
# la lista de bools. Encontré el método .query() pero me pa
# reció "muy SQL" asique tal vez no les gustaba,
# si pudieran aclararme cuál de los dos es el mejor lo agra
# decería. Terminé usando el indexado booleano

#3 Calcular el promedio del precio (Columna: Price) para el
# territorio (Columna:Territory) cuyo
# valor sea "Germany"

#Lee los precios como string asique primero le asigno los v
# alores casteados a numeros
informacion_ejercicio["Price"] = pd.to_numeric(informacion_
ejercicio["Price"], errors = "coerce")

# Genero un data frame auxiliar con indexado booleano que f
# iltra por territorio para que sea Germany y
# no por valor de precio para que no se meta en la cuenta l
# os nan ya que los ignora automaticamente
informacion_ejercicio[informacion_ejercicio["Territory"] ==
"Germany"]["Price"].mean()
```

```
Out[44]: 0.26467889908256864
```

```
In [ ]:
```