

EjercicioDePrueba

October 2, 2020

```
[46]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

Inciso 1

```
[104]: entrenamiento = pd.read_csv('Entrenamieto_ECI_2020.csv', encoding='latin-1')
entrenamiento.tail()
```

```
[104]:
```

	ID	Region	Territory	Pricing, Delivery_Terms_Quote_Appr	\
16942	8781	EMEA	Austria		1
16943	8786	EMEA	Austria		1
16944	8792	EMEA	Austria		1
16945	28561	Americas	NE America		1
16946	28318	Americas	NE America		1

	Pricing, Delivery_Terms_Approved	Bureaucratic_Code_0_Approval	\
16942	1		1
16943	1		1
16944	1		1
16945	1		1
16946	1		1

	Bureaucratic_Code_0_Approved	Submitted_for_Approval	\
16942	1	0	
16943	1	0	
16944	1	0	
16945	1	0	
16946	1	0	

	Bureaucratic_Code	Account_Created_Date	...	Delivery_Quarter	\
16942	Bureaucratic_Code_4	1/15/2016	...	Q1	
16943	Bureaucratic_Code_4	1/15/2016	...	Q2	
16944	Bureaucratic_Code_4	1/15/2016	...	Q1	
16945	Bureaucratic_Code_4	10/20/2015	...	Q2	

16946 Bureaucratic_Code_4 9/3/2015 ... Q3

	Delivery_Year	Actual_Delivery_Date	TRF	Total_Amount_Currency	\
16942	2016	NaT	0	EUR	
16943	2016	NaT	0	EUR	
16944	2016	NaT	0	EUR	
16945	2016	NaT	4	USD	
16946	2016	NaT	40	USD	

	Total_Amount	Total_Taxable_Amount_Currency	Total_Taxable_Amount	\
16942	103350.00	EUR	299715.0	
16943	93015.00	EUR	299715.0	
16944	103350.00	EUR	299715.0	
16945	2346796.88	USD	0.0	
16946	25603200.00	USD	0.0	

	Stage	Prod_Category_A
16942	Closed Won	Prod_Category_A_None
16943	Closed Won	Prod_Category_A_None
16944	Closed Won	Prod_Category_A_None
16945	Closed Lost	Prod_Category_A_None
16946	Closed Lost	Prod_Category_A_None

[5 rows x 52 columns]

Inciso 2

```
[105]: col_rename = {'i>_ID': 'ID'}
entrenamiento = entrenamiento.rename( columns = col_rename )
entrenamiento.tail()
```

	ID	Region	Territory	Pricing, Delivery_Terms_Quote_Appr	\
16942	8781	EMEA	Austria	1	
16943	8786	EMEA	Austria	1	
16944	8792	EMEA	Austria	1	
16945	28561	Americas	NE America	1	
16946	28318	Americas	NE America	1	

	Pricing, Delivery_Terms_Approved	Bureaucratic_Code_0_Approval	\
16942	1	1	
16943	1	1	
16944	1	1	
16945	1	1	
16946	1	1	

	Bureaucratic_Code_0_Approved	Submitted_for_Approval	\
16942	1	0	

16943	1	0
16944	1	0
16945	1	0
16946	1	0

	Bureaucratic_Code	Account_Created_Date	...	Delivery_Quarter	\
16942	Bureaucratic_Code_4	1/15/2016	...	Q1	
16943	Bureaucratic_Code_4	1/15/2016	...	Q2	
16944	Bureaucratic_Code_4	1/15/2016	...	Q1	
16945	Bureaucratic_Code_4	10/20/2015	...	Q2	
16946	Bureaucratic_Code_4	9/3/2015	...	Q3	

	Delivery_Year	Actual_Delivery_Date	TRF	Total_Amount_Currency	\
16942	2016	NaT	0	EUR	
16943	2016	NaT	0	EUR	
16944	2016	NaT	0	EUR	
16945	2016	NaT	4	USD	
16946	2016	NaT	40	USD	

	Total_Amount	Total_Taxable_Amount_Currency	Total_Taxable_Amount	\
16942	103350.00	EUR	299715.0	
16943	93015.00	EUR	299715.0	
16944	103350.00	EUR	299715.0	
16945	2346796.88	USD	0.0	
16946	25603200.00	USD	0.0	

	Stage	Prod_Category_A
16942	Closed Won	Prod_Category_A_None
16943	Closed Won	Prod_Category_A_None
16944	Closed Won	Prod_Category_A_None
16945	Closed Lost	Prod_Category_A_None
16946	Closed Lost	Prod_Category_A_None

[5 rows x 52 columns]

```
[106]: new_entrenamiento = entrenamiento[['Region', 'ID', 'Territory',
→ 'Delivery_Quarter', 'Delivery_Year', 'Total_Amount']].dropna()
new_entrenamiento.tail()
```

	Region	ID	Territory	Delivery_Quarter	Delivery_Year	\
16942	EMEA	8781	Austria	Q1	2016	
16943	EMEA	8786	Austria	Q2	2016	
16944	EMEA	8792	Austria	Q1	2016	
16945	Americas	28561	NE America	Q2	2016	
16946	Americas	28318	NE America	Q3	2016	

Total_Amount

```

16942    103350.00
16943     93015.00
16944    103350.00
16945    2346796.88
16946   25603200.00

```

```
[107]: new_entrenamiento = new_entrenamiento.set_index('Region')
new_entrenamiento.tail()
```

```
[107]:
```

	ID	Territory	Delivery_Quarter	Delivery_Year	Total_Amount
Region					
EMEA	8781	Austria	Q1	2016	103350.00
EMEA	8786	Austria	Q2	2016	93015.00
EMEA	8792	Austria	Q1	2016	103350.00
Americas	28561	NE America	Q2	2016	2346796.88
Americas	28318	NE America	Q3	2016	25603200.00

```
[108]: new_entrenamiento.loc[('EMEA')].nlargest(10, 'Total_Amount')
```

```
[108]:
```

	ID	Territory	Delivery_Quarter	Delivery_Year	Total_Amount
Region					
EMEA	10808	None	Q2	2016	2.162400e+08
EMEA	8037	SE America	Q3	2017	1.539788e+08
EMEA	4909	UAE (Dubai)	Q1	2017	1.237501e+08
EMEA	10620	Kazakhstan	Q2	2016	3.960007e+07
EMEA	13709	Romania	Q2	2016	3.603600e+07
EMEA	4865	France	Q4	2017	3.480420e+07
EMEA	15287	Belgium	Q1	2018	3.332493e+07
EMEA	10350	Romania	Q3	2017	2.880198e+07
EMEA	20848	Italy	Q4	2016	2.879760e+07
EMEA	17668	Germany	Q1	2017	2.829750e+07

Inciso 3

```
[109]: entrenamiento.loc[:, 'Price'] = pd.to_numeric( entrenamiento['Price'],
↳errors='coerce')
new_entrenamiento = entrenamiento[['Price', 'Territory']].dropna()
new_entrenamiento.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 356 entries, 3 to 16132
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Price       356 non-null   float64
1   Territory   356 non-null   object
dtypes: float64(1), object(1)

```

memory usage: 8.3+ KB

```
[110]: new_entrenamiento = new_entrenamiento.set_index(['Territory'])  
new_entrenamiento.tail()
```

```
[110]:
```

	Price
Territory	
Netherlands	0.265
Germany	0.245
Australia	0.300
Australia	0.300
France	0.270

```
[111]: suma = new_entrenamiento.loc(['Germany']).sum()  
suma/new_entrenamiento.size
```

```
[111]: Price    0.081039  
dtype: float64
```