Programming and frameworks for ML

Python for data analyis
Exercises

About Me

Big Data Consultant at Santander / Big Data Lecturer

- More than 20 years of experience in different environments, technologies, customers, countries ...
- Passionate about data and technology
- Enthusiastic about Big Data world and NoSQL

Daniel Villanueva Jiménez

Arquitecto de Datos at Santander Tecnología

Greater Madrid Metropolitan Area · 500+ connections ·

Santander Tecnología

Universidad Pontificia de Salamanca

daniel.villanueva@immune.institute

Exercise 1 (1/4)

- Load the file “oilgascounty.xls” and generate the following dataset:

```
df_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12985 entries, 0 to 12984
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   fips         12985 non-null  int64
 1   stabr        12985 non-null  object
 2   county_name  12985 non-null  object
 3   geoid        12985 non-null  int64
 4   year         12985 non-null  int64
 5   gas          11078 non-null  float64
 6   oil          11471 non-null  float64
dtypes: float64(2), int64(3), object(2)
memory usage: 710.2+ KB
```

Exercise 1 (2/4)

```
df_final.describe(include = 'all')
```

	fips	stabr	county_name	geoid	year	gas	oil
count	12985.000000	12985	12985	12985.000000	12985.000000	1.107800e+04	1.147100e+04
unique	NaN	32	856	NaN	NaN	NaN	NaN
top	NaN	TX	Jackson County	NaN	NaN	NaN	NaN
freq	NaN	2631	105	NaN	NaN	NaN	NaN
mean	32113.044282	NaN	NaN	32113.044282	2005.558337	1.956302e+07	1.198280e+06
std	14839.581818	NaN	NaN	14839.581818	3.450000	6.058323e+07	6.446772e+06
min	1003.000000	NaN	NaN	1003.000000	2000.000000	1.000000e+00	1.000000e+00
25%	20193.000000	NaN	NaN	20193.000000	2003.000000	1.632258e+05	2.118650e+04
50%	31063.000000	NaN	NaN	31063.000000	2006.000000	1.752248e+06	1.575280e+05
75%	48087.000000	NaN	NaN	48087.000000	2009.000000	1.275291e+07	7.024365e+05
max	56045.000000	NaN	NaN	56045.000000	2011.000000	1.198145e+09	2.087814e+08

Exercise 1 (3/4)

```
df_final.query("stabr == 'FL' and year == 2002")
```

	fips	stabr	county_name	geoid	year	gas	oil
1352	12005	FL	Bay County	12005	2002	2909.0	4231.0
1355	12021	FL	Collier County	12021	2002	108457.0	925205.0
1368	12033	FL	Escambia County	12033	2002	1256509.0	654456.0
1380	12051	FL	Hendry County	12051	2002	18690.0	215672.0
1392	12071	FL	Lee County	12071	2002	12937.0	126354.0
1405	12113	FL	Santa Rosa County	12113	2002	2400355.0	1789777.0

```
df_final.query("geoid == 54019 and year.between(2000,2005)")
```

	fips	stabr	county_name	geoid	year	gas	oil
12253	54019	WV	Fayette County	54019	2000	3436078.0	46.0
12254	54019	WV	Fayette County	54019	2001	3320104.0	32.0
12255	54019	WV	Fayette County	54019	2002	3310641.0	NaN
12256	54019	WV	Fayette County	54019	2003	3293598.0	82.0
12257	54019	WV	Fayette County	54019	2004	3156797.0	58.0
12258	54019	WV	Fayette County	54019	2005	2959227.0	82.0

Exercise 1 (4/4)

```
( df_final
  .query("stabr in ('AL', 'FL', 'MT', 'PA', 'WV')")
  .filter(["stabr", "gas", "oil"])
  .groupby("stabr").aggregate(["count", "sum"])
)
```

	gas		oil	
	count	sum	count	sum
stabr				
AL	242	3.818252e+09	156	96737887.0
FL	63	6.723119e+07	63	31882546.0
MT	360	1.165759e+09	315	306083749.0
PA	377	3.721836e+09	265	25076196.0
WV	568	2.866954e+09	424	19218870.0

Exercise 2 (1/5)

- Load the file "nominalmonthlycountryexchangerates_1_.xls“ and generate the following dataset:

```
: df_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 43608 entries, 0 to 43607
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   CountryCode  43608 non-null  object
 1   Country      43608 non-null  object
 2   Region       43608 non-null  object
 3   Currency     43608 non-null  object
 4   Date         43608 non-null  datetime64[ns]
 5   Change       43608 non-null  float64
dtypes: datetime64[ns](1), float64(1), object(4)
memory usage: 2.3+ MB
```

Exercise 2 (2/5)

```
df_final.sample(10)
```

	CountryCode	Country	Region	Currency	Date	Change
15736	4279	France	Europe	Euro	1993-05-01	8.255724e-01
7165	3070	Venezuela	South America	Bolivares	2015-02-01	6.299800e+00
25725	5330	India	South Asia	Rupees	1997-10-01	3.622960e+01
42846	7910	South Africa	Sub-Shaharan Africa	Rand	1998-07-01	6.224590e+00
9031	3510	Brazil	South America	Nominals	1986-08-01	5.007793e-09
36560	7420	Cameroon	Sub-Shaharan Africa	Francs	1980-09-01	2.079400e+02
2729	2150	Honduras	Central America	Lempiras	2013-06-01	2.038000e+01
31214	5800	South Korea	Northeast Asia	Won	1995-03-01	7.786900e+02
4017	2250	Panama	Central America	Balboas	1982-10-01	1.000000e+00
41737	7830	Tanzania	Sub-Shaharan Africa	Shillings	1998-02-01	6.491300e+02

Exercise 2 (3/5)

```
df_final.describe(include = 'all', datetime_is_numeric=True)
```

	CountryCode	Country	Region	Currency	Date	Change
count	43608	43608	43608	43608	43608	43608.000000
unique	79	79	12	50	NaN	NaN
top	2230	South Korea	Europe	Euro	NaN	NaN
freq	552	552	11592	6072	NaN	NaN
mean	NaN	NaN	NaN	NaN	1992-12-15 16:46:57.391304320	297.224336
min	NaN	NaN	NaN	NaN	1970-01-01 00:00:00	0.000000
25%	NaN	NaN	NaN	NaN	1981-06-23 12:00:00	0.896972
50%	NaN	NaN	NaN	NaN	1992-12-16 12:00:00	4.202000
75%	NaN	NaN	NaN	NaN	2004-06-08 12:00:00	33.150000
max	NaN	NaN	NaN	NaN	2015-12-01 00:00:00	25000.000000
std	NaN	NaN	NaN	NaN	NaN	1857.573131

Exercise 2 (4/5)

```
df_final.query(
    "Country == 'Spain' \
        and Date.between('1974-05-01','1975-01-12')")
```

	CountryCode	Country	Region	Currency	Date	Change
19924	4700	Spain	Europe	Euro	1974-05-01	0.345083
19925	4700	Spain	Europe	Euro	1974-06-01	0.344224
19926	4700	Spain	Europe	Euro	1974-07-01	0.342841
19927	4700	Spain	Europe	Euro	1974-08-01	0.344025
19928	4700	Spain	Europe	Euro	1974-09-01	0.346610
19929	4700	Spain	Europe	Euro	1974-10-01	0.345071
19930	4700	Spain	Europe	Euro	1974-11-01	0.343034
19931	4700	Spain	Europe	Euro	1974-12-01	0.339440
19932	4700	Spain	Europe	Euro	1975-01-01	0.337745

Exercise 2 (5/5)

```
pd.set_option('display.max_rows', 12)
df_final.groupby("Region").agg(
    {"Change": "mean", "Country": "count", "Currency" : "last" })
```

Region	Change	Country	Currency
Caribbean	17.762350	1656	TT$
Central America	40.818580	3312	Balboas
Europe	9.428209	11592	Liras
Middle East	3.031537	2760	Riyals
North Africa	12.144625	2208	Pounds
North America	3.330150	1104	Pesos
Northeast Asia	211.053600	2760	Yen
Oceania	1.323456	1104	NZ$
South America	1453.344677	4968	Pesos
South Asia	40.955549	2208	Rupees
Southeast Asia	912.654017	2760	Pesos
Sub-Shaharan Africa	311.463527	7176	Kwacha

Exercise 3 (1/5)

- Load the gas price information from https://sedeaplicaciones.minetur.gob.es/ServiciosRESTCarburantes/PreciosCarburantes/EstacionesTerrestres/
- Load the ‘Comunidades Autonomas” information from https://sedeaplicaciones.minetur.gob.es/ServiciosRESTCarburantes/PreciosCarburantes/Listados/ComunidadesAutonomas/

NOTE: The Price información is generated every hour, so the values of the dataset will be different every time.

NOTE 2: “Precios.pickle” and “Autonomías.pickle” are the datasets I used to generate the sample screenshots (https://tinyurl.com/y2p9ksua) so you can test your code with them

Exercise 3 (2/5)

- Create the following dataset:

```
df_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 37813 entries, 820 to 136977
Data columns (total 13 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   IDEESS       37813 non-null  object 
 1   IDCCAA       37813 non-null  object 
 2   IDMunicipio  37813 non-null  object 
 3   Municipio    37813 non-null  object 
 4   IDProvincia  37813 non-null  object 
 5   Provincia    37813 non-null  object 
 6   Localidad    37813 non-null  object 
 7   CodPos       37813 non-null  object 
 8   Latitud      37813 non-null  float64
 9   Longitud     37813 non-null  float64
 10  CCAA         37813 non-null  object 
 11  TipoPrecio   37813 non-null  object 
 12  Precio       37813 non-null  float64
dtypes: float64(3), object(10)
memory usage: 4.0+ MB
```

Exercise 3 (3/5)

```
df_final.query("IDEESS == '14187'")
```

	IDEESS	IDCCAA	IDMunicipio	Municipio	IDProvincia	Provincia	Localidad	CodPos	Latitud	Longitud	CCAA	TipoPrecio	Precio
55514	14187	01	2786	Motril	18	GRANADA	MOTRIL	18600	36.770722	-3.55725	Andalucia	Precio Gasoleo A	1.215
76588	14187	01	2786	Motril	18	GRANADA	MOTRIL	18600	36.770722	-3.55725	Andalucia	Precio Gasoleo Premium	1.281
97662	14187	01	2786	Motril	18	GRANADA	MOTRIL	18600	36.770722	-3.55725	Andalucia	Precio Gasolina 95 E5	1.359
108199	14187	01	2786	Motril	18	GRANADA	MOTRIL	18600	36.770722	-3.55725	Andalucia	Precio Gasolina 95 E5 Premium	1.419
129273	14187	01	2786	Motril	18	GRANADA	MOTRIL	18600	36.770722	-3.55725	Andalucia	Precio Gasolina 98 E5	1.491

Exercise 3 (4/5)

```
df_final.TipoPrecio.value_counts()
```

```
Precio Gasoleo A                        10383
Precio Gasolina 95 E5                   10061
Precio Gasoleo Premium                   7349
Precio Gasolina 98 E5                    6066
Precio Gasoleo B                         2348
Precio Gases licuados del petróleo        741
Precio Gasolina 95 E5 Premium             659
Precio Gas Natural Comprimido              92
Precio Gas Natural Licuado                 60
Precio Biodiesel                           37
Precio Gasolina 95 E10                      7
Precio Bioetanol                            6
Precio Gasolina 98 E10                      4
Name: TipoPrecio, dtype: int64
```

Exercise 3 (5/5)

```
(df_final
   .groupby(["CCAA", "TipoPrecio"], as_index = False).agg({ "Precio" : "mean" })
   .pivot_table(index = "CCAA", columns = "TipoPrecio", values = "Precio")
   .reset_index()
   .rename_axis(columns = None)
   .filter(["CCAA", "Precio Gasoleo A", "Precio Gasolina 98 E5"])
   .query("CCAA.str.startswith('C')")
)
```

	CCAA	Precio Gasoleo A	Precio Gasolina 98 E5
4	Canarias	0.949545	1.148364
5	Cantabria	1.202775	1.481557
6	Castilla la Mancha	1.196108	1.480416
7	Castilla y León	1.192382	1.475638
8	Cataluña	1.181074	1.507869
9	Ceuta	0.974000	1.104286
10	Comunidad Valenciana	1.173929	1.491363

THANKS FOR YOUR ATTENTION

Daniel Villanueva Jiménez
daniel.villanueva@immune.institute
@dvillaj