DataScientists - Producto 3

```
from google.colab import drive
drive.mount('/content/drive')

     Mounted at /content/drive
```

Nueva sección

Instalación:

Primero de todo instalamos el geopandas utilizando el comando:

```
!pip install geopandas

     Collecting geopandas
       Downloading geopandas-0.10.2-py2.py3-none-any.whl (1.0 MB)
          |████████████████████████████████| 1.0 MB 4.3 MB/s
     Requirement already satisfied: shapely>=1.6 in /usr/local/lib/python3.7/dist-package
     Collecting pyproj>=2.2.0
       Downloading pyproj-3.2.1-cp37-cp37m-manylinux2010_x86_64.whl (6.3 MB)
          |████████████████████████████████| 6.3 MB 29.7 MB/s
     Collecting fiona>=1.8
       Downloading Fiona-1.8.21-cp37-cp37m-manylinux2014_x86_64.whl (16.7 MB)
          |████████████████████████████████| 16.7 MB 413 kB/s
     Requirement already satisfied: pandas>=0.25.0 in /usr/local/lib/python3.7/dist-packa
     Requirement already satisfied: six>=1.7 in /usr/local/lib/python3.7/dist-packages (f
     Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages
     Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (fr
     Collecting munch
       Downloading munch-2.5.0-py2.py3-none-any.whl (10 kB)
     Collecting click-plugins>=1.0
       Downloading click_plugins-1.1.1-py2.py3-none-any.whl (7.5 kB)
     Requirement already satisfied: click>=4.0 in /usr/local/lib/python3.7/dist-packages
     Requirement already satisfied: attrs>=17 in /usr/local/lib/python3.7/dist-packages (
     Collecting cligj>=0.5
       Downloading cligj-0.7.2-py3-none-any.whl (7.1 kB)
     Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-package
     Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packag
     Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/di
     Installing collected packages: munch, cligj, click-plugins, pyproj, fiona, geopandas
     Successfully installed click-plugins-1.1.1 cligj-0.7.2 fiona-1.8.21 geopandas-0.10.2
```

2. Importamos los modulos necesarios

Ahora vamos a importar las librerías que vamos a usar, importaremos “pandas” como “pd” y geopandas como “gpd”, seaborn y numpy

```
import pandas as pd
import geopandas as gpd
import dateutil
from shapely.geometry import shape,Point,Polygon,mapping,LineString
import numpy as np
import seaborn as sns
```

b) Subiremos el archivo csv descargado

Generaremos un fichero excel a partir del csv

```
read_file = pd.read_csv ("2017_accidents_gu_bcn.csv")
read_file.to_excel ("2017_accidents_gu_bcn.xlsx", index = None, header=True)
```

c) Leemos el fichero de excel

(excel_file es el path+nombre y lo volcamos en una variable), luego leemos el contenido del fichero volcandolo en otra variable

```
excel_file = '2017_accidents_gu_bcn.xlsx'
accidents_BCN = pd.read_excel(excel_file)
```

d) Mostramos el fichero configurando que muestre las columnas activas

```
pd.options.display.max_columns= None
accidents_BCN
```

	Numero_expedient	Codi_districte	Nom_districte	Codi_barri	Nom_barri	Cc
0	2017S008429	-1	Desconegut	-1	Desconegut	
1	2017S007316	-1	Desconegut	-1	Desconegut	
2	2017S010210	-1	Desconegut	-1	Desconegut	
3	2017S006364	-1	Desconegut	-1	Desconegut	
4	2017S004615	10	Sant Martí	64	el Camp de l'Arpa del Clot	
...	...	...	...	...	...	
10334	2017S003667	9	Sant Andreu	59	el Bon Pastor	
10335	2017S001806	9	Sant Andreu	59	el Bon	
10336	2017S010718	9	Sant Andreu	59	el Bon	

e) creamos la matriz de correlacion

```
corr = accidents_BCN.corr()
corr
```

	Codi_districte	Codi_barri	Codi_carrer	Any	Mes_any
Codi_districte	1.000000	0.987703	0.155259	NaN	0.008264
Codi_barri	0.987703	1.000000	0.158968	NaN	0.010694
Codi_carrer	0.155259	0.158968	1.000000	NaN	0.059767

f) Generamos una mascara para el triangulo superior

```
mask = np.zeros_like(corr, dtype=bool)
mask[np.triu_indices_from(mask)] = True
```

Numero_morts	-0.002664	-0.003866	-0.005197	NaN	0.003663

Generamos a un mapa de color divergente

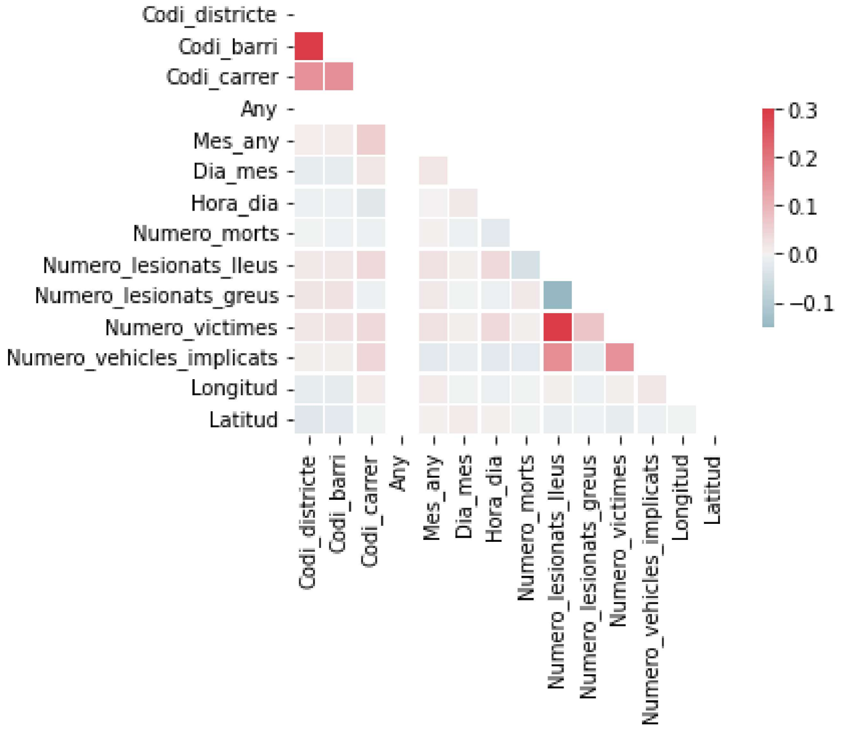
Numero_lesionats_greus	0.022454	0.024794	-0.004814	NaN	0.015151

```
cmap = sns.diverging_palette(220, 10, as_cmap=True)
```

Dibujamos un mapa de calor con la mascara creada

```
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f00232c6950>
```



REGRESION LINEAL SIMPLE

Importamos el csv y printamos las columnas

```
ACCIDENTS_BCN_CSV = pd.read_csv ("2017_accidents_gu_bcn.csv")
ACCIDENTS_BCN_CSV.columns
```

```
Index(['Numero_expedient', 'Codi_districte', 'Nom_districte', 'Codi_barri',
       'Nom_barri', 'Codi_carrer', 'Nom_carrer', 'Num_postal',
       'Descripcio_dia_setmana', 'Dia_setmana', 'Descripcio_tipus_dia', 'Any',
       'Mes_any', 'Nom_mes', 'Dia_mes', 'Hora_dia', 'Descripcio_torn',
       'Descripcio_causa_vianant', 'Numero_morts', 'Numero_lesionats_lleus',
       'Numero_lesionats_greus', 'Numero_victimes',
       'Numero_vehicles_implicats', 'Coordenada_UTM_X', 'Coordenada_UTM_Y',
       'Longitud', 'Latitud'],
      dtype='object')
```

h) Importamos la api de statsmodel

```
import statsmodels.formula.api as smf
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarni
  import pandas.util.testing as tm
```

Creamos un modelo ajustado en una linea (Quantitative response=Hora_dia; and predictor=Numero_morts).

```
lm = smf.ols(formula='Hora_dia ~ Numero_victimes', data=ACCIDENTS_BCN_CSV).fit()
```

Printamos los coeficientes de regresion

```
lm.params
```

```
Intercept          13.468930
Numero_victimes     0.290557
dtype: float64
```

Hacemos una inspeccion del total de los resultados

```
print(lm.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               Hora_dia   R-squared:                       0.002
Model:                            OLS   Adj. R-squared:                  0.002
Method:                 Least Squares   F-statistic:                     16.71
Date:                Fri, 06 May 2022   Prob (F-statistic):           4.38e-05
```

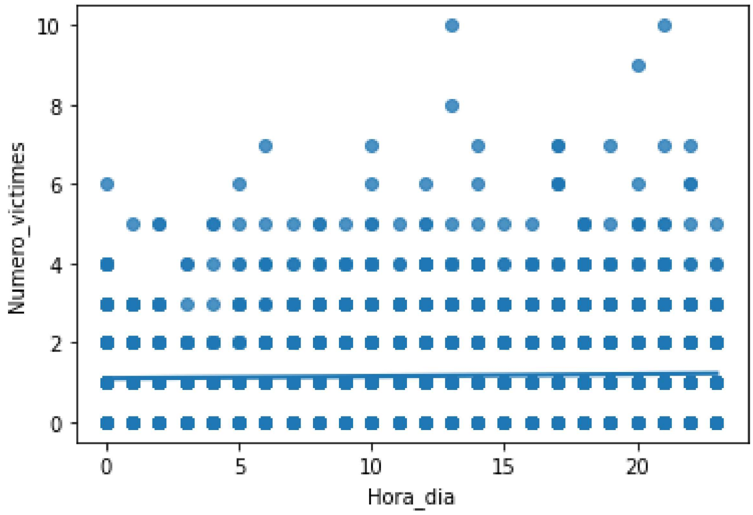
```
Time:                        16:37:58   Log-Likelihood:                -31936.
No. Observations:               10339   AIC:                         6.388e+04
Df Residuals:                   10337   BIC:                         6.389e+04
Df Model:                           1
Covariance Type:            nonrobust
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
Intercept          13.4689      0.099    136.426      0.000      13.275      13.662
Numero_victimes     0.2906      0.071      4.088      0.000       0.151       0.430
==============================================================================
Omnibus:                      347.945   Durbin-Watson:                   1.984
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              364.086
Skew:                          -0.439   Prob(JB):                     8.70e-80
Kurtosis:                       2.725   Cond. No.                         3.72
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
```

Printamos una recta con el modelo de regresion lineal

```
sns.regplot(x='Hora_dia', y='Numero_victimes', data=ACCIDENTS_BCN_CSV)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0016c3d390>
```



✓ 0 s completado a las 18:44 ● ✕