

M15uf2: BigData



Pràctica: Anàlisi de dades amb Python

Curs: 2019-20

CFGs: DAM2

Alumne : Pau Desunvila
Arnau Subirós

Data : 03/12/2019

Nom i CognomsPau Desumvila
Arnau Subirós**Data**

03/12/2019

Pràctica: Anàlisi de dades amb Python

❖ Descripció

Aquest projecte consisteix en desenvolupar un projecte que calcula estadístics bàsics d'un conjunt de dades. Cada grup ha de triar un conjunt de dades de

- <https://github.com/fivethirtyeight/data>

.Les dades les obtenim de la web FiveThirtyEight

- (<https://fivethirtyeight.com/>), que realitza articles basats en dades sobre esports i notícies, i que posa a disposició pública els conjunts de dades (<https://github.com/fivethirtyeight/data>) que recull per als seus articles.

❖ Objectius

- Treballar amb Python, la seva POO, les estructures de dades i la I/O
- Treballar amb JupyterLab i/o Google Colab
- Desenvolupar una solució d'anàlisi de dades amb l'ús de les biblioteques Pandas,
- NumPy i Matplotlib

❖ Lliuraments

El lliurament es farà en els lliuraments dels treballs en el termini especificat. Es farà mitjançant un notebook de Jupyter. S'adjuntarà amb un PDF explicatiu de les decisions preses i del funcionament de la solució d'anàlisi de dades.

Nom i Cognoms

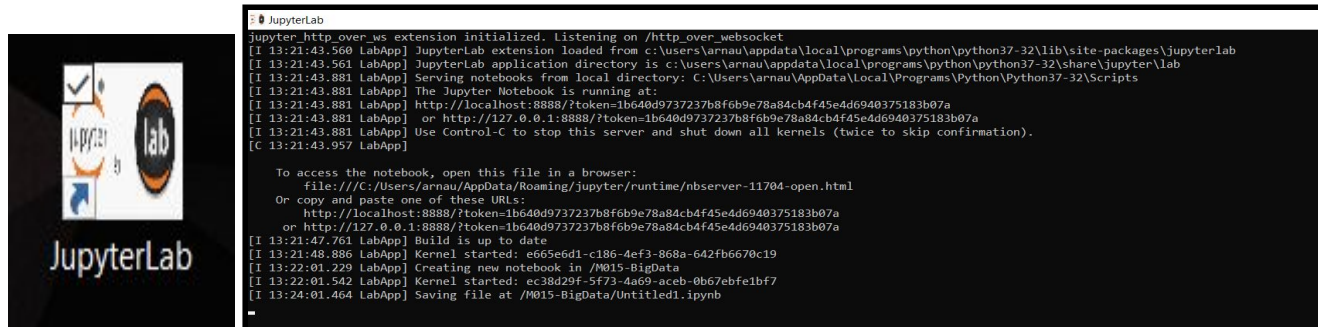
Pau Desumvila
Arnau Subirós

Data

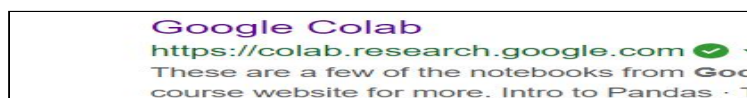
03/12/2019

Comentaris de la pràctica

Abans de començar amb bla pràctica previament hem tingut que instal.lar el Jupiter Notebook i iniciar-lo.



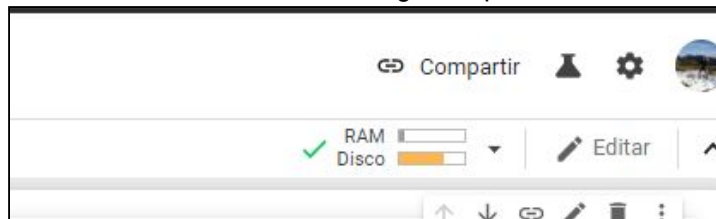
Un cop fet aquest cas, accedim a <https://colab.research.google.com>



Ens registrem i vinculem la compte al google drive.



I ens connectem a l'entorn de Google on podrem modificar el codi en temps real.



Crearem un nou arxiu amb el nom **ProjecteBigData** que utilitzarà Python(versió 3)

En aquesta pràctica s'ha consultat les dades la següent font :

<https://github.com/fivethirtyeight/data/tree/master/hate-crimes>

Nom i Cognoms

 Pau Desumvila
 Arnau Subirós

Data

03/12/2019

Comencem en importar les 3 llibreries :

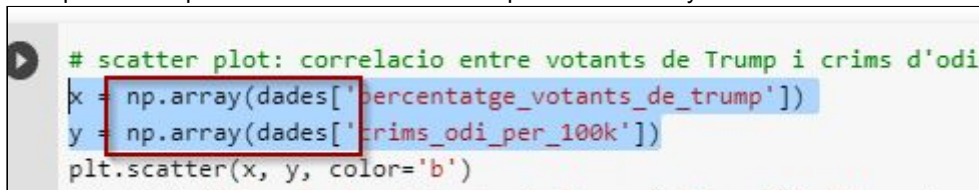
- 1) **Pandas** : és una llibreria de Python destinada a l'anàlisi de dades, que proporciona unes estructures de dades flexibles i que permeten treballar amb ells de forma molt eficient. En aquesta pràctica farem servir les següents estructures de dades
 - a) **Series**: Són arrays unidimensionals amb indexació (arrays amb índex o etiquetades) similars als diccionaris. Poden generar-se a partir de llistes o diccionaris
 - b) **DataFrame** : Són estructures de dades similars a les taules de base de dades relacionades (estil SQL)

NOTA: De la llibreria Pandas s'ha utilitzat **dataframe.corr()** s'utilitza per a trobar la correlació per parells de totes les columnes en el marc de dades. Qualsevol na/Na s'exclou automàticament. Per a qualsevol columna de tipus de dades no numèric en el marc de dades, s'ignora

DataFrame.corr(self , method = 'pearson' , min_periods = 1)

- **el metode : "pearson"** : Que és un coeficient de correlació estàndard
- **el metode : "kendall"** : Que és coeficient de correlació de Kendall Tau (" És una mesura de dependència no paramètrica que identifica els parells concordants i discordants de dues variables. Una vegada identificats, es calculen els totals i es fa el quocient.")

- 2) **Numpy**: és una llibreria de Python que significa "Numerical Python", que proporciona potents estructures de dades, implementant matrius i matrius multidimensionals. Aquestes estructures generen càlculs eficients amb matrius
 - a) En la pràctica es pot veure com l'hem fet servir per crear un array amb les files d'una columna



```
# scatter plot: correlacio entre votants de Trump i crims d'odi
x = np.array(dades['percentatge_votants_de_trump'])
y = np.array(dades['crims_odi_per_100k'])
plt.scatter(x, y, color='b')
```

- 3) **Matplotlib**: és una llibreria de traçat per al llenguatge de Python i la seva llibreria matemàtica Numpy. Generant gràfiques amb 2D. Hem utilitzat **matplotlib.pyplot** que és una col·lecció de funcions d'estil de comandament que fan l'estil de comandament que fan que matplotlib funcioni com a MATLAB. Hem utilitzat les següents funcions:

- a) **scatter()**: És una funció incorporada per crear diagrames de dispersió com mostrarem a la pràctica
- b) **title()**: És una funció que ens permet canviar el títol de la figura
- c) **xlabel()**: És una funció que ens permet canviar el títol de l'eix X
- d) **ylabel()**: És una funció que ens permet canviar el títol de l'eix Y
- e) **matshow()**: És una funció que ens permet imprimir mapes de calor
- f) **xticks()**: És una funció que ens permet fer alteracions en els índexos de l'eix de X. (Ex. rotació, color, taman, etc..)
- g) **yticks()**: És una funció que ens permet fer alteracions en els índexos de l'eix de Y. (Ex. rotació, color, taman, etc..)
- h) **bar()**: És una funció que ens permet imprimir i alterar una barra com a llegenda

Nom i Cognoms

Pau Desumvila
Arnau Subirós

Data

03/12/2019



```
# importem les biblioteques:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

- Carreguem les dades que hem consultat a :

https://raw.githubusercontent.com/fivethirtyeight/data/master/hate-crimes/hate_crimes.csv

```
# carreguem les dades:
dades = pd.read_csv("https://raw.githubusercontent.com/fivethirtyeight/data/master/hate-crimes/hate_crimes.csv")
dades.columns=["estat", "mitjana_ingresos_familiars", "percentatge_desempleats_temporers", "percentatge_habitants_areas_metropolitanes", "percentatge_habitants_amb_titol_high_school", "percentatge_habitants_sense_nacionalitat", "percentatge_habitants"]
dades.head(n=10)
```

	estat	mitjana_ingresos_familiars	percentatge_desempleats_temporers	percentatge_habitants_areas_metropolitanes	percentatge_habitants_amb_titol_high_school	percentatge_habitants_sense_nacionalitat	percentatge_habitants
0	Alabama	42278	0.060	0.64	0.821	0.02	
1	Alaska	67629	0.064	0.63	0.914	0.04	
2	Arizona	49254	0.063	0.90	0.842	0.10	
3	Arkansas	44922	0.052	0.69	0.824	0.04	
4	California	60487	0.059	0.97	0.806	0.13	
5	Colorado	60940	0.040	0.80	0.893	0.06	
6	Connecticut	70161	0.052	0.94	0.886	0.06	
7	Delaware	57522	0.049	0.90	0.874	0.05	
8	District of Columbia	68277	0.067	1.00	0.871	0.11	
9	Florida	46140	0.052	0.96	0.853	0.09	

- Exemple de serie utilitant la llibreria **Pandas**

```
##Una manera de crear sèries es crear objectes de sèries. Utilitzant la llibreria Pandas
#Exemple##

### fem una serie de la columna estat
pd.Series(["Alabama", "Alaska", "Arizona", "Arkansas", "California", "Colorado", "Connecticut", "Delaware", "District of Columbia", "Florida"])
```

0	Alabama
1	Alaska
2	Arizona
3	Arkansas
4	California
5	Colorado
6	Connecticut
7	Delaware
8	District of Columbia
9	Florida

dtype: object

Nom i Cognoms

 Pau Desumvila
 Arnau Subirós

Data

03/12/2019

Amb els objectes del Dataframe hem creat diversos dict per assignar noms a les columnes utilitzant l'estructura de dades "Series" de llibreria Pandas

```
#### Els objectes del Dataframe poden crear un dict que assigni noms de columnes de string a les series corresponents
#### Si las Series no coincideixen amb la seva longitud. Els valors que falten es completen amb NA/NaN.

estat=pd.Series(["Alabama","Alaska","Arizona","Arkansas","California","Colorado","Connecticut","Delaware","District of Columbia","Florida"])
mitjana_ingresos= pd.Series([42278, 67629, 49254,44922,60487,60940,70161,57522,68277,46140 ])
mitjana_ingresos2 = pd.Series
percentatge_habitants=pd.Series([0.02,0.04,0.10,0.04,0.13,0.06,0.05,0.11,0.09])
percentatge_votants_de_trump1=pd.Series([0.63,0.53,0.50,0.60,0.33,0.44,0.41,0.42,0.04,0.049])
crims_odi=pd.Series([0.125839,0.143740,0.225320,0.069061,0.255805,0.390523,0.335392,0.322754,1.522302,0.187521])
pd.DataFrame({'estat':estat1,'mitjana_ingresos_familiars':mitjana_ingresos,'percentatge_habitants_sense_nacionalitat':percentatge_habitants,'percentatge_votants_de_trump':percentatge_votants_de_trump1,'crims_odi_per_100':crims_odi_per_100})
```

	estat	mitjana_ingresos_familiars	percentatge_habitants_sense_nacionalitat	percentatge_votants_de_trump	crims_odi_per_100
0	Alabama	42278	0.02	0.630	0.125839
1	Alaska	67629	0.04	0.530	0.143740
2	Arizona	49254	0.10	0.500	0.225320
3	Arkansas	44922	0.04	0.600	0.069061
4	California	60487	0.13	0.330	0.255805
5	Colorado	60940	0.06	0.440	0.390523
6	Connecticut	70161	0.05	0.410	0.335392
7	Delaware	57522	0.11	0.420	0.322754
8	District of Columbia	68277	0.09	0.040	1.522302
9	Florida	46140	NaN	0.049	0.187521

En primer lloc farem un scatter plot amb la correlació entre el % de votants de Trump i el numero de crims d'odi per cada estat dels EEUU. Per fer aquest scatter plot farem servir el següent codi:

```
[ ] # scatter plot: correlació entre votants de Trump i crims d'odi
x = np.array(dades['percentatge_votants_de_trump'])
y = np.array(dades['crims_odi_per_100k'])
plt.scatter(x, y, color='b')
plt.title('Correlació: Votants de Trump / Crims d\'odi', color='w')
plt.xlabel('Percentatge votants de Trump', color='w')
plt.ylabel('Crims d\'odi per 100.000 habitants', color='w')
```

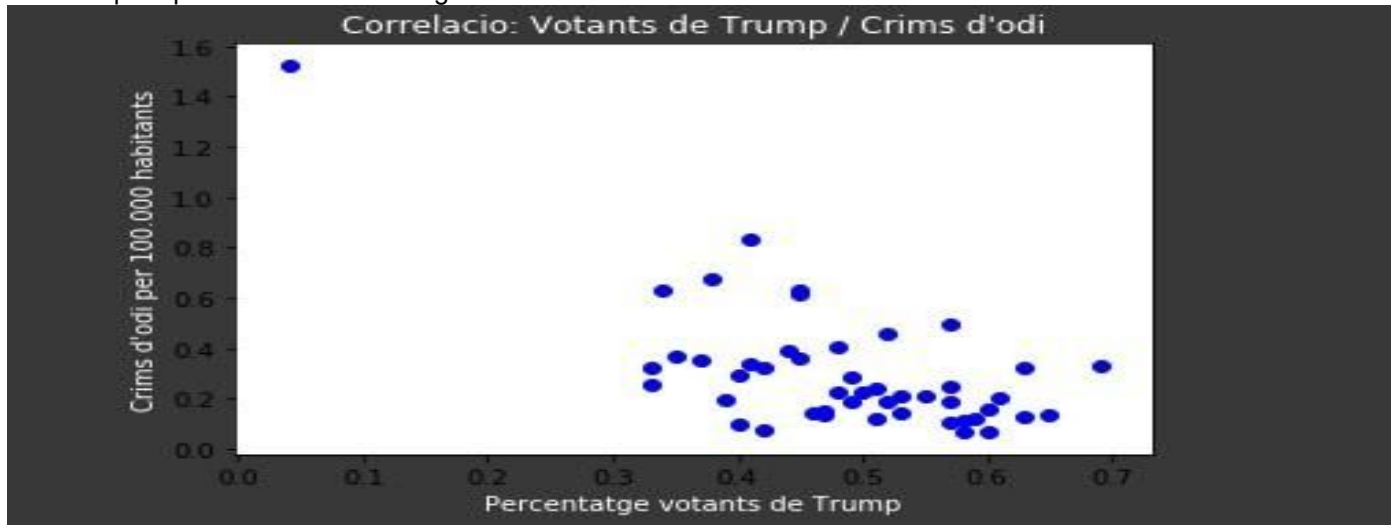

Nom i Cognoms

Pau Desumvila
Arnau Subirós

Data

03/12/2019

L'scatter plot que obtindrem es el següent:

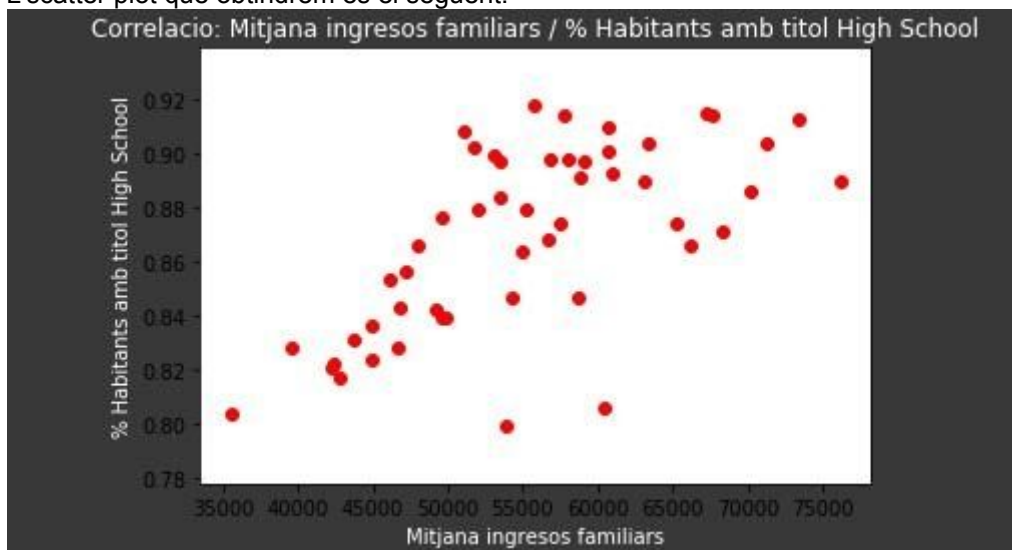


Habiem decidit fer aquesta correlacio per comprobar si els prejudicis de la gent entorn al votants de Trump son certes. Tot i que aquesta taula no demostra res, podem comprobar que hi ha una certa correlacio negativa entre el numero de votants de Trump i la quantitat de crims d'odi que podem trobar en cada estat dels EEUU.

A continuacio farem un scatter plot amb la correlacio entre la mitjana d'ingresos familiars i el % d'habitants amb titol High school. Per fer aquest scatter plot hem fet servir el següent codi:

```
[ ] # scatter plot: correlacio entre mitjana ingresos familiars i % habitants amb titol high school:
x = np.array(dades['mitjana_ingresos_familiars'])
y = np.array(dades['percentatge_habitants_amb_titol_high_school'])
plt.scatter(x, y, color='r')
plt.title('Correlacio: Mitjana ingresos familiars / % Habitants amb titol High School', color='w')
plt.xlabel('Mitjana ingresos familiars', color='w')
plt.ylabel('% Habitants amb titol High School', color='w')
```

L'scatter plot que obtindrem es el següent:



Nom i Cognoms

 Pau Desumvila
 Arnau Subirós

Data

03/12/2019

Amb aquest exemple volíem comprovar si es cert que estudiar t'obre portes la vida. Com podem comprovar a la imatge, hi ha una correlació positiva entre la mitjana d'ingresos familiars i el % d'habitants amb títol d'High school. És a dir, com més estudis té la gent, més beneficis obte del seu treball.

Fins ara hem tractat les nostres dades fent correlacions entre 2 tipus de valors. Tot seguit intentarem fer la correlació per a totes les columnes de les nostres dades. En primer lloc farem la correlació seguint el mètode Pearson. Tot seguit farem la correlació entre les nostres columnes seguint el mètode Kendall. En les següents imatges podem comprobar el codi i el resultat per a les correlacions amb els dos mètodes comentats:

```
# calculem correlació entre les columnes amb el mètode pearson:
dades.corr(method='pearson')
```

	mitjana_ingresos_familiars	percentatge_desempleats_temporers
mitjana_ingresos_familiars	1.000000	-0.376722
percentatge_desempleats_temporers	-0.376722	1.000000
percentatge_habitants_areas_metropolitanes	0.286480	0.358113
percentatge_habitants_amb_titol_high_school	0.653490	-0.621168
percentatge_habitants_sense_nacionalitat	0.302288	0.278899
percentatge_habitants_blancs_pobres	-0.818045	0.209440
index_gini	-0.178821	0.541659
percentatge_no_caucasics	0.103775	0.431847
percentatge_votants_de_trump	-0.597528	-0.148192
crims_odi_per_100k	0.350714	0.083292
mitjana_crims_odi_per_100k_fbi	0.318246	0.073936

```
# calculem correlació entre les columnes amb el mètode kendall:
dades.corr(method='kendall')
```

	mitjana_ingresos_familiars	percentatge_desempleats_temporers
mitjana_ingresos_familiars	1.000000	-0.280410
percentatge_desempleats_temporers	-0.280410	1.000000
percentatge_habitants_areas_metropolitanes	0.214922	0.246696
percentatge_habitants_amb_titol_high_school	0.482084	-0.500406
percentatge_habitants_sense_nacionalitat	0.235290	0.186902
percentatge_habitants_blancs_pobres	-0.663161	0.165757
index_gini	-0.237574	0.424076
percentatge_no_caucasics	0.023726	0.428914
percentatge_votants_de_trump	-0.445428	-0.063923
crims_odi_per_100k	0.241443	-0.104728
mitjana_crims_odi_per_100k_fbi	0.188571	-0.032153

Nom i Cognoms

Pau Desumvila
Arnau Subirós

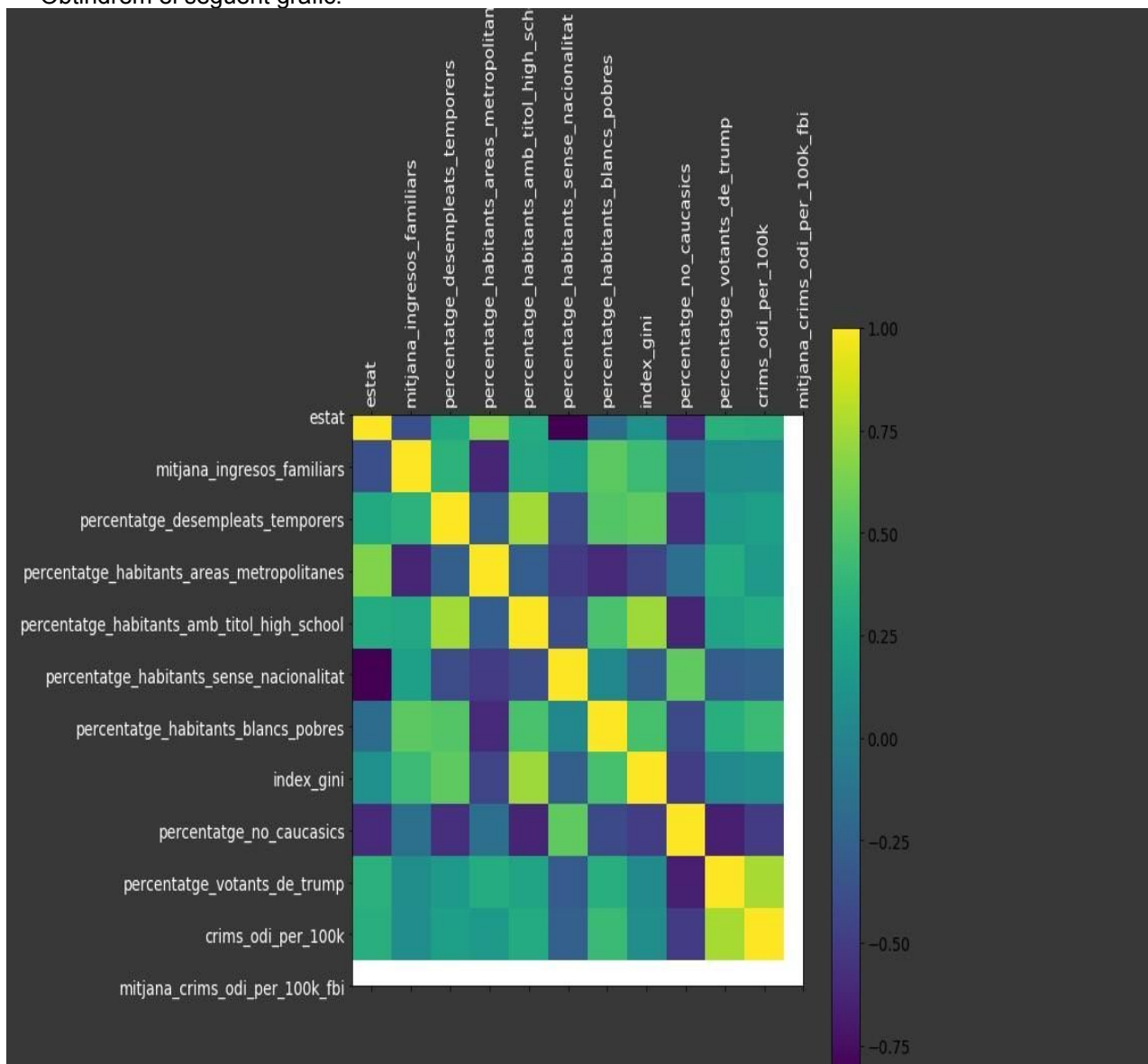
Data

03/12/2019

Per intentar graficar la correlacio entre columnes d'una manera atractiva introduirem el següent codi:

```
[ ] # printem la correlacio entre columnes amb estil personalitzat:
f = plt.figure(figsize=(10, 10))
plt.matshow(dades.corr(), fignum=f.number)
plt.xticks(range(dades.shape[1]), dades.columns, fontsize=14, rotation=90, color='w')
plt.yticks(range(dades.shape[1]), dades.columns, fontsize=14, color='w')
cb = plt.colorbar()
cb.ax.tick_params(labelsize=14)
```

Obtindrem el següent grafic:



Per intentar printar de manera correcta la correlació entre les diferents columnes en un mapa de calor introduïrem el següent codi:

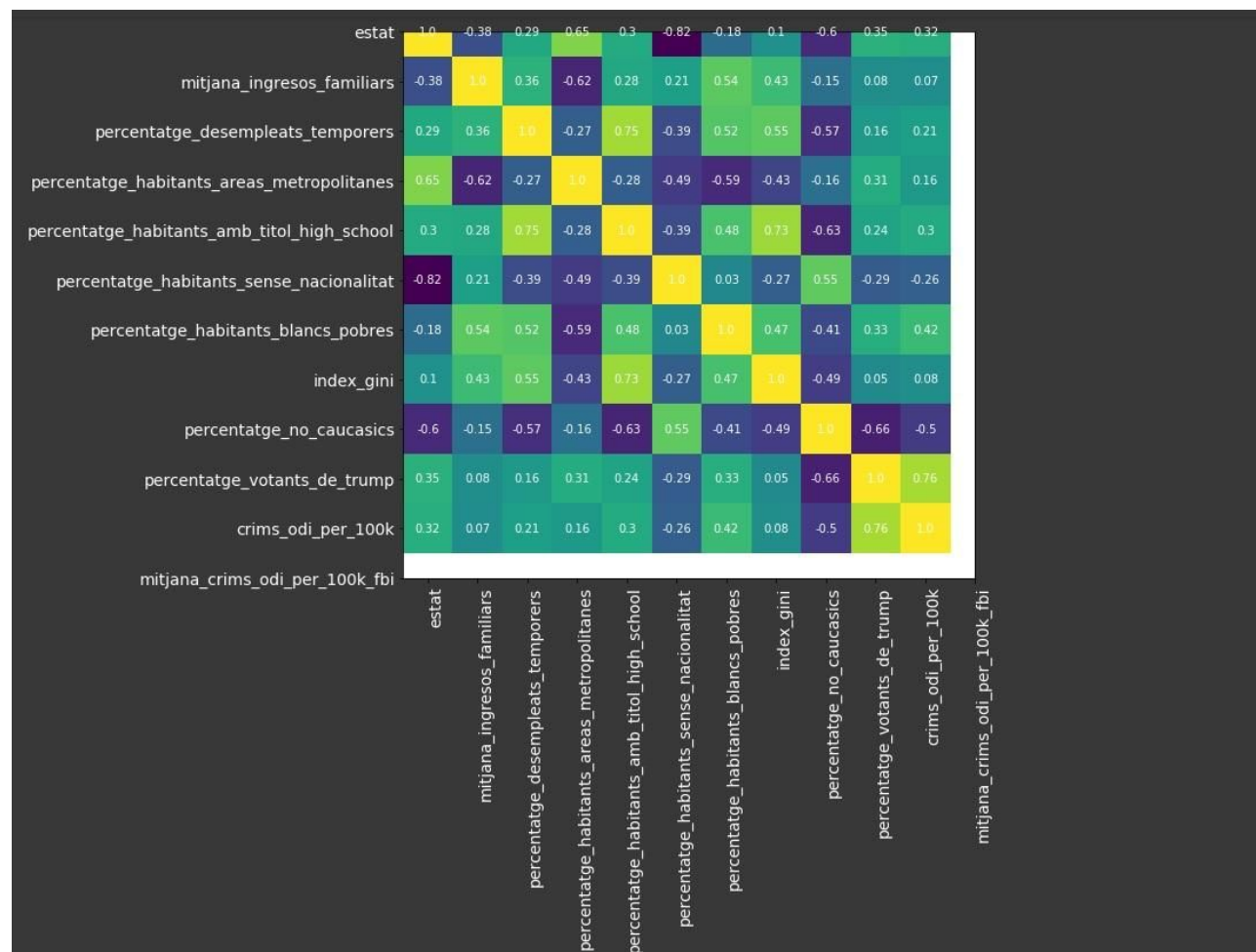
```
[ ] # una altra manera de fer la correlació entre columnes:
x = dades.columns
y = dades.columns
correlation = np.array(dades.corr())
fig, ax = plt.subplots(figsize=(12,12))
im = ax.imshow(correlation)

ax.set_xticks(np.arange(len(x)))
ax.set_yticks(np.arange(len(y)))
ax.set_xticklabels(x, color = 'w', fontsize=14)
ax.set_yticklabels(y, color = 'w', fontsize=14)

plt.setp(ax.get_xticklabels(), rotation = 90, ha = "right", rotation_mode = "anchor")

for i in range(len(x)-1):
    for j in range(len(y)-1):
        text = ax.text(j, i, round(correlation[i, j], 2), ha = "center", va = "center", color = "w")

fig.tight_layout()
plt.show()
```



Nom i Cognoms

Pau Desumvila
Arnau Subirós

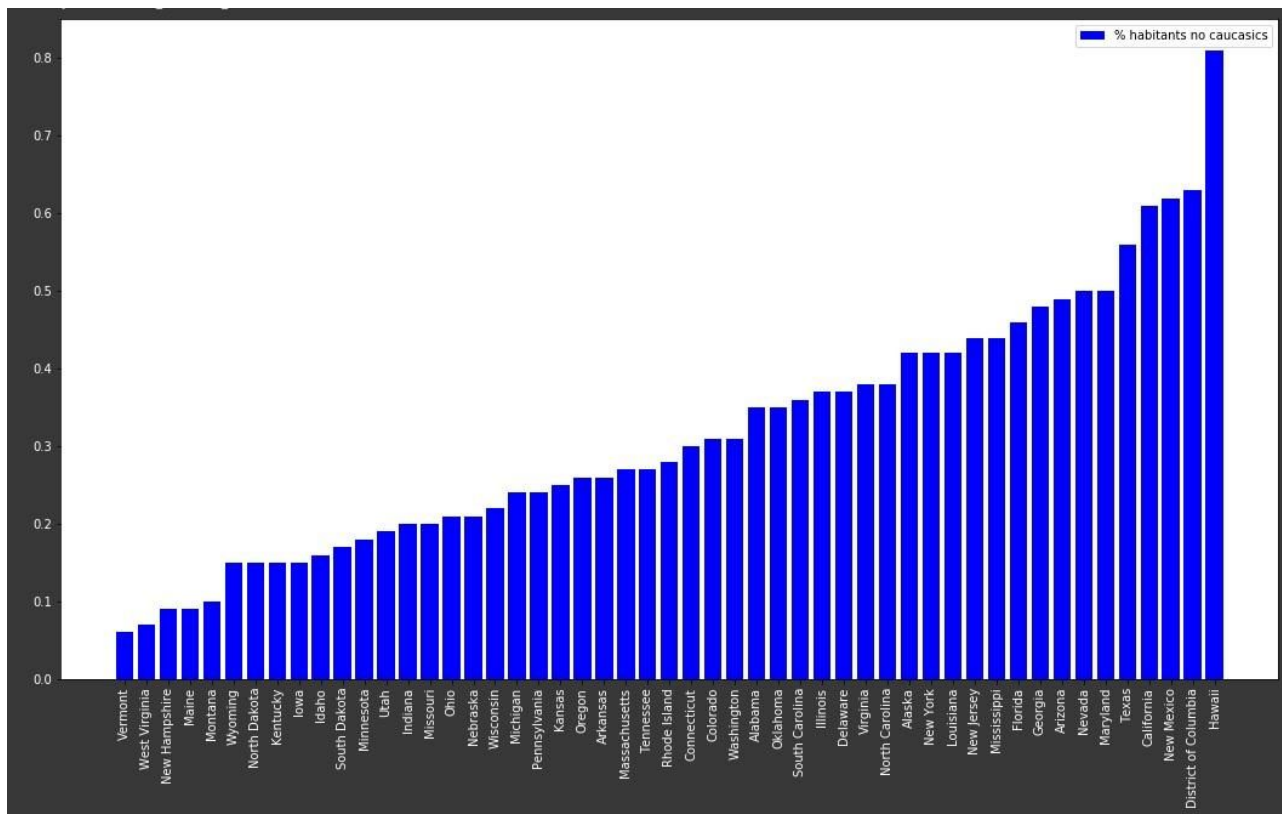
Data

03/12/2019

Com podem comprobar a la imatge hem afegit el valor per a cada correlacio calculada. Malauradament les vores blanques continuen apareixent.

Com a ultim grafic intentarem graficar el % d'habitants no caucasics per cada estat dels EEUU, i que aparegui de manera ordenada de menor a mayor. Per fer-ho farem servir el seguent codi:

```
[ ] # grafic de barres que mostra percentatge no caucasics per estat ordenat de menor a major
g = plt.figure(figsize=(18, 10))
sorted = dades.sort_values(by=['percentatge_no_caucasics'])
plt.bar(sorted['estat'], sorted['percentatge_no_caucasics'], label = '% habitants no caucasics', color = 'b')
plt.xticks(rotation = 90, color = 'w')
plt.yticks(color = 'w')
plt.legend()
```



Com podem observar a la imatge la taula s'ha generat correctament. A més a més, els resultats corresponen amb el que esperabem. L'estat amb major % de poblacio no caucasica es Hawaii. Seguit d'estats que es troben aprop de la frontera com New mexico, California, Texas, etc.

Nom i Cognoms
Data

 Pau Desumvila
 Arnau Subirós

03/12/2019

Com a segon bloc, hem creat un directori on s'ha creat un exemple de classe amb herència

```
##### CREACIO de CLASSES al directori BigData #####

[ ]

[11] ### Mirem les carpetes actuals i els seus permisos
ls -la

total 16
drwxr-xr-x 1 root root 4096 Nov 21 16:30 ./
drwxr-xr-x 1 root root 4096 Dec 3 12:21 ../
drwxr-xr-x 1 root root 4096 Nov 21 16:30 .config/
drwxr-xr-x 1 root root 4096 Nov 21 16:30 sample_data/

[12] ## creacio del directori BigData
mkdir BigData

# confirme que el directori s'ha creat
ls -la | grep BigData

drwxr-xr-x 2 root root 4096 Dec 3 13:03 BigData/
```

```
[14] ## accedim al directori
cd BigData

/content/BigData

[15] ### confiremque estem treballant en el directori creat
pwd

'/content/BigData'

### Creacio clase principal
class American:
    def __init__(self,nom,cognom):
        self.nom=nom
        self.cognom=cognom

    def __str__(self):
        cadena=self.nom+", "+self.cognom
        return cadena
```

Nom i Cognoms**Data**Pau Desumvila
Arnau Subirós

03/12/2019

```
[17] ###creacio dels objectes de la clase American
american1=American("Faye C.", "Washington")
american2=American("Tameka M.", "Johns")
american3=American("Felicitas A.", "Cole")
american4=American("MaAnthony L.", "Taylorrtinez")

[18] ## imprimint per pantalla els objectes creats
print(str(americana1)+"-"+str(americana2)+"-"+str(americana3) + "-"+str(americana4))

Faye C.,Washington-Tameka M.,Johns-Felicitas A.,Cole-MaAnthony L.,Taylorrtinez

[19] ###Creació de les subclasses
class VotantTrump(American):
    pass

[20] ###Creació de les subclasses
class NoCaucassic(American):
    pass

[23] #####creacio dels objectes de les subclasses

##Subclasse VotantTrump
trump1=VotantTrump("Aaron M.", "Merrill")
trump2=VotantTrump("Margaret R.", "Mattia")

##Subclasse NoCaucassic
no_caucassic1=NoCaucassic("Pedro", "Lopez")
no_caucassic2=NoCaucassic("Roberto", "Gonzalez")

#### imprimim els objectes creats
print(str(trump1)+"-"+str(trump2))
print(str(no_caucassic1)+"-"+str(no_caucassic2))

Aaron M.,Merrill-Margaret R.,Mattia
Pedro, Lopez-Roberto, Gonzalez
```