# CAT-PEC4-enun

June 1, 2024

```
<div style="float: left; width: 50%;">
   <img src="https://www.uoc.edu/portal/system/modules/edu.uoc.presentations/resources/img/bra
</div>

<p style="margin: 0; padding-top: 22px; text-align:right;">22.403 · Programació per a la Ciènc
<p style="margin: 0; text-align:right;">Grau en Ciència de Dades Aplicada</p>
<p style="margin: 0; text-align:right; padding-button: 100px;">Estudis d'Informàtica, Multimèd
```

## 1 Programació per a la ciència de dades - PEC4

En aquest Notebook trobareu l'exercici que suposa la quarta i darrera activitat d'avaluació continuada (PAC) de l'assignatura. Aquesta PAC intenta presentar-vos un petit projecte en el qual heu de resoldre diferents exercicis, que engloba molts dels conceptes coberts durant l'assignatura.

L'objectiu d'aquest exercici serà desenvolupar un paquet de Python fora de l'entorn de Notebooks, que ens permeti resoldre el problema donat. Treballareu amb arxius Python plans `.py`. El projecte haurà d'incloure el corresponent codi organitzat lògicament (separat per mòduls, organitzats per funcionalitat,...), la documentació del codi (*docstrings*) i tests. A més, haureu d'incloure els corresponents arxius de documentació d'alt nivell (`README`) així com els arxius de llicència i dependències (`requirements.txt`) comentats a la teoria.

Fer un `setup.py` és opcional, però si es fa es valorarà positivament de cara a la nota de la pràctica i del curs.

## 2 Enunciat

Volem estudiar el comportament de la població dels Estats Units pel que fa a l'ús d'armes de foc. Utilitzarem el següent dataset que ja ha estat incorporat a la carpeta del projecte, provinent del següent enllaç: * https://www.kaggle.com/datasets/pedropereira94/nics-firearm-background-checks

El dataset representa l'acumulació d'informació (per data i estat) de la verificació d'antecedents de gent que es vol treure el permís d'armes. Es vol veure si es pot treure algun tipus de conclusió sobre diferències d'estat, evolució temporal, etc.

- la columna *permit* significa els permisos (verificació d'antecedents)
- la columna *handgun* serien les peticions d'armes curtes (pistoles)
- la columna *long_gun* serien les peticions d'armes llargues

Aquestes seran les columnes que ens interessen a part del mes (*month*) i de l'estat (*state*).

A més, a l'exercici 5 utilitzarem també dades poblacionals per calcular les dades relatives. Per això utilitzarem el següent dataset que també hem inclòs en el projecte: * https://gist.githubusercontent.com/bradoyler/0fd473541083cfa9ea6b5da57b08461c/raw/fa5f59ff1ce7ad9ff792e223b state-populations.csv

El dataset presenta la població (2014) dels diferents estats ubicats als Estats Units i conté les columnes següents:

- la columna *code*, un string de dues lletres que identifica els estats (per exemple Califòrnia s'identifica com a CA i Florida com a FL).
- la columna *state* amb el nom de l'estat sense abreujar.
- la columna *pop_2014*, representant el nombre d'habitants l'any 2014.

# 3 Projecte Python, funcionalitat

Per fer el lliurament més fàcil i homogeni us demanem que organitzeu el codi de tal manera que **des del fitxer principal retorneu totes les respostes que se us demana a la PAC** fent ús de funcions que haureu de definir en mòduls. Per això, en cada exercici us indicarem el format que ha de tenir cada resposta, de manera que executant `main.py` es vagi responent a tota la PAC. Per defecte, `main.py` ha d'executar totes les funcions de la PAC mostrant com funcionen però també ha de permetre executar-les una per una si es desitja. Ho heu de documentar tot molt bé en el *README* de manera que el professor pugui executar el codi sense problemes i sense dubtes. Us recordem que al *README* també heu d'indicar com executar els tests i comprovar-ne la cobertura.

## 3.1 Exercici 1: Lectura i neteja de dades. (0.75 punts)

### 3.1.1 Exercici 1.1. (0.25 punts)

Implementeu una funció anomenada *read_csv*: - **Inputs:** La funció rebrà com a dades d'entrada un únic paràmetre que serà la url del fitxer que volem llegir. - **Funcionalitat:** La funció haurà de llegir el fitxer csv *nics-firearm-background-checks.csv*. Per comprovar que aquestes dades s'han carregat correctament, la funció haurà de mostrar per pantalla les cinc primeres columnes de la base de dades i també la seva estructura. - **Outputs:** La funció tornarà el dataframe que s'ha llegit.

### 3.1.2 Exercici 1.2. (0.25 punts)

Implementeu una funció anomenada *clean_csv*: - **Inputs:** Com a dades d'entrada la funció rebrà l'estructura de dades (dataframe). - **Funcionalitat:** La funció haurà de ser capaç de netejar el dataset inicial, eliminant totes les columnes excepte les cinc que utilitzarem al llarg de l'exercici: *month, state, permit, handgun, long_gun*. Per assegurar-vos que la funció és correcta, s'haurà de mostrar per pantalla el nom de totes les columnes del dataframe. - **Outputs:** La funció tornarà el dataframe contenint únicament les columnes *month, state, permit, handgun, long_gun.*

### 3.1.3 Exercici 1.3. (0.25 punts)

Implementeu una funció anomenada *rename_col*: - **Inputs:** El dataframe amb totes les columnes. - **Funcionalitat:** La funció haurà de ser capaç de canviar el nom de la columna *longgun* per

*long_gun* Per assegurar-vos que la funció és correcta, ens hem d'assegurar que aquesta columna efectivament existeix al dataframe. Així mateix, haurem de mostrar per pantalla el nom de totes les columnes del dataframe dins la mateixa funció. - **Outputs:** El dataframe amb el nom de la columna canviat.

## 3.2 Exercici 2: Processament de dades (1 punt)

La informació de la columna mesos es troba amb un format que no és massa manejable. Per exemple febrer de l'any 2020 apareixerà com *2020-2*. Anem a solucionar aquest problema:

### 3.2.1 Exercici 2.1 (0.5 punts)

Implementeu una funció anomenada *breakdown_date*: - **Inputs:** El dataframe contenint la columna *month* amb el format de dades igual que a l'exemple. - **Funcionalitat:** La funció dividirà la informació que hi ha a la columna *month* creant dues noves columnes al dataframe: la columna *year* i que contindrà el número de l'any i la columna *month* que serà el número del mes. Seguint l'exemple, per al valor *2020-2* la columna *year* serà **2020** i la columna *month* serà el valor **2**. Per assegurar-nos que la funció és correcta, caldrà mostrar les cinc primeres files del dataframe resultant. - **Outputs:** El dataframe amb la informació de la data dividida.

### 3.2.2 Exercici 2.2 (0.5 punts)

Implementeu una funció anomenada *erase_month*: - **Inputs:** La funció rebrà el dataframe contenint la columna *month*. - **Funcionalitat:** Eliminar la columna month. Per comprovar que s'ha realitzat correctament, la funció també ha de mostrar per pantalla les cinc primeres files de dades i el nom de totes les columnes. - **Outputs:** El dataframe sense la columna *month*.

## 3.3 Exercici 3: Agrupament de dades (1 punt)

### 3.3.1 Exercici 3.1. (0.5 punts)

Implementeu una funció anomenada *groupby_state_and_year* - **Inputs:** La funció rebrà el dataframe obtingut a l'exercici 2.2. - **Funcionalitat:** La funció haurà de ser capaç de calcular els valors acumulats totals agrupant les dades per any i per estat: (columnes *year* i *state*). - **Outputs:** El dataframe resultant amb les dades agrupades.

### 3.3.2 Exercici 3.2 (0.25 punts)

Implementeu una funció anomenada *print_biggest_handguns* - **Inputs:** La funció rebrà el dataframe amb les dades agrupades per estat i per any com a resultat de l'exercici 3.1. - **Funcionalitat:** La funció haurà d'imprimir per pantalla un missatge informatiu indicant el nom de l'estat i l'any on s'ha registrat un nombre més gran de *hand_guns*. - **Outputs:** Aquesta funció no tornarà cap valor.

### 3.3.3 Exercici 3.3 (0.25 punts)

Implementeu una funció anomenada *print_biggest_longguns* - **Inputs:** La funció rebrà el dataframe amb les dades agrupades per estat i per any com a resultat de l'exercici 3.1. - **Funcionalitat:** La funció haurà d'imprimir per pantalla un missatge informatiu indicant el nom de l'estat i l'any on s'ha registrat un nombre més gran de *long_guns*. - **Outputs:** Aquesta funció no tornarà cap valor.

### 3.4 Exercici 4: Anàlisi temporal (1 punt)

Per a aquest exercici es demanarà fer un anàlisi temporal per veure l'evolució de les llicències, les pistoles i els rifles d'assalt al llarg dels anys. Per això serà necessari:

#### 3.4.1 Exercici 4.1 (0.75 punts)

Implementeu una funció anomenada *time_evolution()* que creï un gràfic amb les següents característiques: - L'eix X serà el número de l'any (que en el cas d'aquest dataframe hauria de variar des de 1998 fins a 2020), mentre que a l'eix y es mostraran tres sèries temporals amb el nombre total de *permit*, *hand_gun* i *long_gun* registrat per cadascun dels anys.

#### 3.4.2 Exercici 4.2 (0.25 punts)

Comenta el gràfic generat a l'exercici 4.2. Es veu una correlació de llicències, pistoles i rifles d'assalt al llarg dels anys? És la tendència ascendent o descendent? Hi ha hagut canvis durant la pandèmia? Què podríem esperar els propers anys?

**Nota**: A https://cnnespanol.cnn.com/2024/02/15/cultura-armas-estados-unidos-mundo-trax/ hi ha una gràfica sobre el nombre de víctimes de tirotejos massius. El 2017 hi ha un màxim, que sembla coincidir amb els resultats que haureu obtingut.

### 3.5 Exercici 5: Anàlisi dels estats (1.25 punts)

Al llarg d'aquest exercici aplicarem una mica de ciència de dades i traurem una sèrie de conclusions agrupant les dades per cadascun dels estats:

#### 3.5.1 Exercici 5.1 (0.25 punts)

Implementeu una funció anomenada *groupby_state* - **Inputs:** La funció rebrà el dataframe amb les dades agrupades per estat i per any com a resultat de l'exercici 3.1. - **Funcionalitat:** La funció mostrarà els valors totals agrupant els valors únicament per estat i no per any. Per comprovar que la funció és correcta es demanarà també que mostri per pantalla les 5 primeres files del dataframe resultant. - **Outputs:** Aquesta funció haurà de tornar el dataframe amb els valors agrupats únicament per estats.

**Nota** Els resultats obtinguts a la funció de l'exercici 5.1 ens mostren únicament els valors absoluts. Tot i això, també cal tenir en compte que no tots els estats són igual de poblats. Per establir una comparació justa, hauríem de tenir en compte també la població total de cada estat, per calcular així els valors relatius. Per fer-ho, utilitzarem un nou conjunt de dades que hem obtingut de la següent adreça: * https://gist.githubusercontent.com/bradoyler/0fd473541083cfa9ea6b5da57b08461c/raw/fa5f59ff1ce7ad9ff792e223b
state-populations.csv

#### 3.5.2 Exercici 5.2 (0.25 punts)

Els següents estats no apareixen en l'arxiu *us-state-populations.csv*: Guam, Mariana Islands, Puerto Rico i Virgin Islands. Per tant, caldrà eliminar-los del nostre dataframe per poder continuar amb el nostre anàlisi de dades.

Implementeu una funció anomenada *clean_states*: - **Inputs:** La funció rebrà el dataframe amb les dades agrupades per estat com a resultat de l'exercici 5.1. - **Funcionalitat:** La funció primer

comprovarà si existeixen aquests quatre estats (Guam, Mariana Islands, Puerto Rico i Virgin Islands) i, en el cas que existeixin els eliminarà. Per comprovar que la funcionalitat s'ha implementat correctament, la funció també mostrarà per pantalla el nombre d'estats diferents. - **Outputs:** Aquesta funció tornarà el mateix dataset però sense els quatre estats esmentats.

#### 3.5.3 Exercici 5.3 (0.25 punts)

Ara el nostre objectiu serà fusionar els dos datasets:

Implementeu una funció anomenada *merge_datasets*: - **Inputs:** La funció rebrà com a paràmetres d'entrada el conjunt de dades resultant de l'exercici exercici 5.2 i el conjunt de dades poblacionals provinents del fitxer: *us-state-populations.csv*. (Per llegir les dades de la població pots utilitzar la funció creada a l'exercici 1.1). - **Funcionalitat:** La funció fusionarà les dades dels dos datasets rebuts com a paràmetres d'entrada, incloent per cada estat tota la informació procedent de les dues fonts de dades. Per comprovar que s'ha fet correctament, la funció imprimirà per pantalla les cinc primeres files del dataset resultant. - **Outputs:** Aquesta funció tornarà el dataset resultant de fusionar les dades.

#### 3.5.4 Exercici 5.4 (0.25 punts)

A continuació cal calcular els valors relatius:

Implementeu una funció anomenada *calculate_relative_values*: - **Inputs:** La funció rebrà com a paràmetres d'entrada, el conjunt de dades resultant de l'exercici 5.3. - **Funcionalitat:** La funció crearà 3 noves columnes anomenades *permit_perc*, *longgun_perc* i *handgun_perc* (per si hi ha algun despistat que es confongui amb la regla de tres com ja va passar amb la PAC2 us donaré una pista, per exemple, en el cas de *permit_perc* els valors relatius es calcularien amb la fórmula: (permit * 100) / poblacioTotal ) - **Outputs:** Aquesta funció retornarà el dataset resultant amb les tres columnes noves: *permit_perc*, *loggun_perc* i *shotgun_perc* i els valors relatius ja calculats.

#### 3.5.5 Exercici 5.5 (0.25 punts)

1 - En primer lloc, calcularem la mitjana de permisos *permit_perc* amb dos decimals i mostrarem el resultat a la pantalla 2 - En segon lloc, mostrarem per pantalla tota la informació relativa a l'estat de *Kentucky*

**Nota** Tenim un problema tècnic! L'estat de Kentucky és el que s'anomena un outlier o valor atípic. Els *outliers* són valors atípicament alts que distorsionen qualsevol tipus de mètriques estadístiques. En aquest cas, la mitjana està inflada a causa dels valors que té aquest estat. Els *outliers* no només distorsionen les mètriques estadístiques, també fan que algoritmes d'aprenentatge màquina arribin a conclusions errònies i això és un problema.

3- Reemplaçar el valor *permit_perc* de *Kentucky* amb el valor de la mitjana d'aquesta columna. 4- Tornem a calcular la mitjana amb dos decimals. 5- Ha canviat molt el valor? Entens el procés de treure valors atípics? Escriu les teves conclusions.

### 3.6 Exercici 6: Mapes coroplètics (1.5 punts)

Geographic Data Science (GDS) és la branca de la Ciència de Dades que utilitza dades amb informació geogràfica. En aquest enllaç tens disponible un curs de GDS, amb un docker perquè utilitzeu els materials (per quan tingueu temps): *https://darribas.org/gds_course/content/home.html

Els mapes coroplètics són els mapes on pintem una àrea (estat, regió, província, país) d'un color dins d'un mapa de colors per obtenir informació visual.

Farem 3 mapes coroplètics per a *permit_perc*, *handgun_perc* i *longgun_perc*. Hi ha diferents solucions. Proposem fer-ho amb el codi disponible a: * https://python-graph-gallery.com/292-choropleth-map-with-folium/

Per això, necessitaràs instal · lar les llibreries *folium* i *selenium*.

El fitxer *us-states.json* conté tota la informació de les fronteres dels estats. El primer que podeu fer és fer funcionar l'exemple que es proposa (*US_Unemployment_Oct2012.csv*) i després adaptar-lo a la informació que disposem.

Pots generar 3 mapes (m1, m2 i m3), un per a cada variable. Després aquests mapes els podeu desar a imatge amb el codi següent:

Has d'obtenir 3 imatges, una per a cada variable que tens (*permit_perc*, *handgun_perc* i *longgun_perc*).

# 4 Criteris de correcció

Aquesta PAC s'avaluarà seguint els criteris següents:

- **Funcionalitat** (6.5 punts): Es valorarà que el codi implementi tot el que es demana.
  - Exercici 1 (0.75 punts)
  - Exercici 2 (1 punt)
  - Exercici 3 (1 punt)
  - Exercici 4 (1 punt)
  - Exercici 5 (1.25 punts)
  - Exercici 6 (1.5 punts)
- **Documentació** (0.5 punts): Totes les funcions dels exercicis d'aquesta PAC hauran d'estar degudament documentades utilitzant docstrings (en el format que preferiu).
- **Modularitat** (0.5 punts): Es valorarà la modularitat del codi (tant l'organització del codi en mòduls com la creació de funcions).
- **Estil** (0.5 punts): El codi ha de seguir la guia d'estil de Python (PEP8), exceptuant els casos on fer-ho compliqui la llegibilitat del codi.
- **Tests** (1.5 punts): El codi ha de contenir una o diverses suites de tests que permetin comprovar que el codi funciona correctament, amb un mínim del 50% de cobertura.
- **Requeriments** (0.5 punts): Heu d'incloure un fitxer de *requirements.txt* que contingui la llista de llibreries necessàries per executar el codi.
- **README** i **LICENSE** (0.25 punts): Heu d'afegir també un fitxer README, que presenti el projecte i expliqui com executar-lo, així com la inclusió de la llicència sota la qual es distribueix el codi (podeu triar la que vulgueu).

## 4.1 Important

**Nota 1**: De la mateixa manera que en les PACs anteriors, els criteris transversals es valoraran de manera proporcional a la part de funcionalitat implementada.

Per exemple, si el codi només implementa la meitat de la PAC i la documentació està perfecta, la puntuació corresponent a la documentació serà de 0.25.

**Nota 2**: És imprescindible que el paquet que lliuris s'executi correctament en la màquina virtual i que el fitxer REAMDE expliqui clarament com executar el codi per generar els resultats que es demanen. A més, en el README s'ha d'explicar també com s'executaran els test i com es comprovarà la seva cobertura. **El primer que farà el professor quan corregeixi és llegir el fitxer README i seguir les instruccions que allà s'especifica.**

**Nota 3**: Entregueu el paquet com un únic arxiu .zip que contingui només el codi en el Registre d'Avaluació Contínua. **El codi Python estarà escrit amb fitxers plans de Python.**