Modelització i Visualització

Pràctica 1

NEUS OLLER MATAS, CURS 2023-23

Enginyeria Informàtica

Universitat Rovira i Virgili

1 Informe de pràctica: Anàlisi de Matriculacions en Estudis de Grau a Espanya

Aquest informe aborda l'anàlisi de dades relacionades amb les matriculacions en estudis de grau a Espanya. Les dades utilitzades s'han obtingut del portal de dades obertes del govern espanyol a través del fitxer "bd_titulats_base.csv". Aquest estudi es centra en l'anàlisi de la distribució de les matriculacions per sexe, dedicació a l'estudi i camp de l'estudi.

La pràctica inclou la creació de diverses gràfiques per il·lustrar les dades de les matriculacions d'estudiants a Espanya. Aquestes gràfiques, que inclouen barres, línies, circulars i un mapa d'Espanya es presenten en la documentació detallada com a part de l'anàlisi exhaustiu de les dades relacionades amb els títols i la matriculació a les universitats. Cada gràfica proporciona una representació visual que es detalla més profundament, destacant els resultats més significatius i les tendències observades.

El programa està creat amb Google Collab. Accedint en el següent enllaç accedireu al programa. En cas que vulgueu executar alguna gràfica, si us plau executeu des de l'inici:
https://colab.research.google.com/drive/1w1QE4vD3LZxlHmSusdPWmK0uys9OhXAN?usp=sharing

1.1 Conjunt de Dades Principal

El fitxer principal és "bd_titulats_base.csv," que conté dades detallades sobre les matriculacions per sexe, dedicació a l'estudi i camp de l'estudi per a estudis de grau a Espanya.

1.1.1 ESTRUCTURA DEL FITXER

El fitxer "bd_titulats_base.csv" té una estructura ben definida.

- A la seva primera columna (Columna 0) es troben els títols i identificadors que serveixen per aclarir les dades contingudes en la taula.
- Les següents 6 columnes (Columna 1 a Columna 6) contenen dades globals que abasten totes les especialitats per a cada any acadèmic, amb un total de 6 cursos.
- A partir de la columna 7 (Columna 7 en endavant), cada titulació o especialitat ocupa un conjunt de 6 columnes que corresponen als 6 cursos acadèmics.

Aquesta estructura es repeteix per a cada titulació, oferint una organització coherent de les dades.

En relació a les dades de les universitats i la matriculació:

- Les primeres files del fitxer agrupen les universitats públiques i privades.
- A partir de la fila 46, les dades estan organitzades per cada universitat, ocupant 13 files per cada una.
- A la fila 6 de cada universitat es troben les dades totals de matriculació d'homes, mentre que a la fila 10 s'inclouen les dades totals de matriculació de dones.

Aquesta estructura permet un anàlisi detallada i comparativa de les dades de matriculació per sexe, dedicació a l'estudi i camp d'estudi en estudis de grau a Espanya.

1.2 Conjunt de Dades de Comunitats Autònomes i Universitats

Per a la creació de gràfiques circulars i un mapa d'Espanya (esmentat més endavant), s'ha creat un conjunt de dades addicionals utilitzant els fitxers "comunitats4_UTF8.csv" i "mapa.svg"

És important destacar que aquest fitxer ha estat creat de manera personal, assegurant que les universitats hi apareguin amb els mateixos noms i format que es troben en el fitxer principal "bd_titulats_base.csv."

Aquest nou conjunt de dades relaciona les comunitats autònomes amb les seves respectives universitats.

2 Importància de l'estudi

L'estudi de la distribució de les matriculacions per gènere és fonamental per a la comprensió de les desigualtats dins de l'educació superior. Les diferències de gènere en les eleccions acadèmiques poden tenir un impacte significatiu en la igualtat d'oportunitats. És crucial avaluar i abordar aquestes desigualtats per assegurar una educació superior més inclusiva i equitativa.

Com a estudiant de 4t d'Enginyeria Informàtica i dona, he considerat vital realitzar aquest estudi per aportar a la comprensió d'aquesta desigualtat i fer visible la necessitat de promoure la igualtat de gènere en aquest àmbit.

3 Anàlisi Visual de les Matriculacions Universitàries a Espanya: Gràfiques

3.1 Carregar les dades

En el primer bloc de carrega de dades, es realitza la preparació i obtenció de les fonts de dades necessàries per a l'anàlisi.
Procediments realitzats:

3.1.1 Fitxer de Dades Principal (BD_titulats_base.csv)

La font principal de dades és el fitxer "BD_titulats_base.csv" que conté dades detallades sobre les matriculacions universitàries per sexe, dedicació a l'estudi i camp d'estudi per estudis de grau a Espanya.

Aquest fitxer es descarrega del portal de dades obertes del govern d'Espanya a través de la següent URL:
https://estadisticas.mecd.gob.es/EducaJaxiPx/files/_px/es/xlsx/Universitaria/Alumnado/EEU_2022/GradoCiclo/Matriculados/l0/3_9_Mat_GradCiclo_Sex_Ded_Campo_Univ.px

Per assegurar una manipulació consistent de les dades, aquest fitxer s'ha descarregat en format XLSX i s'ha carregat en un DataFrame. A més, s'ha guardat una còpia del fitxer en la carpeta pròpia del Drive en format CSV, utilitzant la codificació UTF-8 i el separador de columnes ";".

3.1.2 Fitxer de Localització Geogràfica (comunitats4_UTF8.csv)

Per poder realitzar comparatives geogràfiques i assignar els centres universitaris a les seves comunitats autònomes,he creat un fitxer CSV anomenat "comunitats4_UTF8.csv." Aquest fitxer conté les assignacions de cada universitat a la seva zona geogràfica per comunitats autònomes, ja que el fitxer de dades original no conté una referència geogràfica. Aquest fitxer s'ha carregat en un DataFrame anomenat "df_comunitats" i s'ha guardat una còpia a la carpeta pròpia del Drive per a un ús compartit.

3.1.3 Fitxer SVG del Mapa d'Espanya (mapa.svg)

Per a la visualització de dades geogràfiques, s'ha utilitzat un fitxer SVG que representa el mapa d'Espanya. Aquest fitxer SVG ha estat modificat per incloure identificacions de les comunitats autònomes.
El fitxer s'ha carregat des d'una URL (https://drive.google.com/uc?export=view&id=1xE7YZVDwNKG--dxJHCnaVojnAg0CSZwJ) i s'ha guardat en la carpeta pròpia del Drive amb el nom "mapa.svg" per facilitar la seva referència des de l'aplicació.

3.1.4 Codi

```
!pip install chardet
import warnings
import requests
from google.colab import drive
import os
import pandas as pd
import gdown

# Configura les advertencies per deixar mkes net els resultats
warnings.filterwarnings("ignore", category=FutureWarning)
warnings.filterwarnings("ignore", category=FutureWarning)
warnings.filterwarnings("ignore", category=UserWarning)
warnings.filterwarnings("default", category=UserWarning)

# Ruta de la carpeta a la teva unitat de Google Drive (personalitza-la)
ruta_drive = '/content/drive/MyDrive/Colab Notebooks/Treball_MV'

# Comprova si la carpeta ja existeix i, si no, crea-la
if not os.path.exists(ruta_drive):
    os.makedirs(ruta_drive)

print("Carpeta creada a Google Drive:", ruta_drive)

url_excel =
"https://estadisticas.mecd.gob.es/EducaJaxiPx/files/_px/es/xlsx/Universitaria/Alumnado/EEU_2022/GradoCiclo/Matriculados/l0/3_9_Mat_GradCiclo_Sex_Ded_Campo_Univ.px"
df_excel = pd.read_excel(url_excel)
ruta_csv_drive = ruta_drive + '/BD_titulats_base.csv'
df_excel.to_csv(ruta_csv_drive, encoding='UTF-8', sep=';', index=False)
```

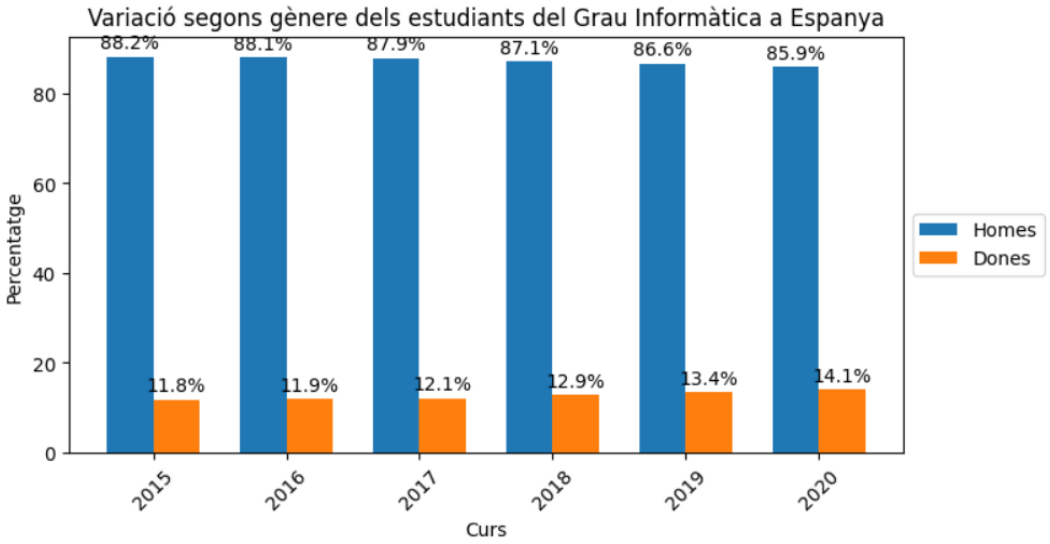
```
# Carrega les dades del fitxer amb les ubicacions de les universitats
file_path = ruta_drive + '/BD_titulats_base.csv'
df = pd.read_csv(file_path, encoding='UTF-8', sep=';')

# Descarrega un fitxer CSV amb les dades de les comunitats autònomes de Google Drive
url_csv_comunitats = "https://drive.google.com/uc?export=view&id=1-2IfraZ8mdr_YR4VnJuWiuMcg80LZl5H"
ruta_csv_comunitats = ruta_drive + "/comunitats4_UTF8.csv"
df_comunitats = pd.read_csv(url_csv_comunitats)

# Guarda l'arxiu CSV en la teva Unitat del Drive
file_path= ruta_drive + "/comunitats4_UTF8.csv"
df_comunitats.to_csv(file_path, encoding='UTF-8', sep=';' , index=False)

# Carrega el mapa SVG des de la URL
url_mapa = 'https://drive.google.com/uc?export=view&id=1xE7YZVDwNKG--dxJHCnaVojnAg0CSZwJ'
response = requests.get(url_mapa)

# Comprova si la petició a la URL és correcta (codi 200)
if response.status_code == 200:
    # Obtenir el contingut del fitxer
    contingut_fitxer = response.text

    # Defineix la ruta on vols guardar el fitxer al teu Drive
    ruta_mapa = ruta_drive + '/mapa.svg'

    # Guarda el contingut al teu Google Drive
    with open(ruta_mapa, 'w') as fitxer:
        fitxer.write(contingut_fitxer)

    print(f'El fitxer s\'ha guardat a "{ruta_mapa}" al teu Google Drive.')

else:
    print('Error en descarregar el fitxer. Comprova la URL i la connexió a Internet.')

print("Procés de carrega de fitxers finalitzat")
```

Aquest bloc de codi realitza múltiples tasques per carregar i preparar les dades i recursos necessaris per a l'anàlisi visual de les matriculacions universitàries a Espanya:

- S'instal·la la llibreria "*chardet*" per detectar la codificació de caràcters correcta dels fitxers.
- S'importen les biblioteques i s'ajusten les advertències per millorar la visualització dels resultats.
- Es defineix la ruta de la carpeta a la unitat de Google Drive on es desaran els fitxers i es comprova si aquesta carpeta ja existeix. Si no existeix, es crea la carpeta.
- Es carrega un fitxer XLSX des d'una URL, es converteix a un DataFrame i es guarda com un fitxer CSV amb la codificació UTF-8 i un separador de punt i coma. Aquesta còpia del fitxer es desa a la carpeta de Google Drive especificada.
- Es carreguen les dades del fitxer de localització geogràfica des d'una URL que apunta a un fitxer CSV. Aquestes dades es carreguen en un DataFrame i, posteriorment, es guarden com una còpia a la carpeta de Google Drive amb la mateixa codificació i separador.
- Es descarrega un fitxer SVG que representa el mapa d'Espanya des d'una URL i es guarda a la carpeta de Google Drive especificada.
- Es proporcionen missatges de confirmació per a cada pas realitzat, indicant on s'han desat els fitxers.

Aquest codi assegura que totes les dades i recursos necessaris estiguin disponibles per a l'anàlisi visual posterior de les matriculacions universitàries. Els fitxers s'han guardat a la carpeta de Google Drive especificada per facilitar l'accés i l'ús compartit amb altres usuaris.

3.2 Gràfica de barres



3.2.1 COMENTARI DE LA GRÀFICA

La gràfica de barres presentada mostra l'evolució de la matriculació en el Grau d'Enginyeria Informàtica a Espanya durant un període de sis cursos acadèmics, des del 2015-2016 fins al 2020-2021. Aquesta representació visual destaca la distribució d'estudiants per gènere en aquesta disciplina.

Les dades demostren un lleuger augment en la matriculació de dones en el Grau d'Enginyeria Informàtica en tot l'estat al llarg dels sis cursos. Tot i aquest augment, la gràfica posa de manifest la marcada disparitat de gènere en aquesta àrea d'estudi. En mitjana, les dones representen aproximadament un 15% dels estudiants, mentre que els homes constitueixen un 85%.

Aquesta gran disparitat reflecteix la necessitat d'avançar en l'afavoriment de la igualtat de gènere en l'àmbit de l'enginyeria informàtica. Malgrat els progressos realitzats, encara queda molt per fer per assegurar que les dones tinguin igualtat d'oportunitats i estiguin representades de manera més equitativa en aquest camp.

Aquesta gràfica ressalta la importància de les iniciatives i les polítiques educatives destinades a fomentar la participació de les dones en àrees STEM (Ciència, Tecnologia, Enginyeria i Matemàtiques), com l'enginyeria informàtica. La igualtat de gènere en aquest àmbit no només és una qüestió d'equitat sinó que també pot enriquir la diversitat i les perspectives professionals en el futur.

L'evolució de la matriculació en enginyeria informàtica és un indicatiu important del progrés en la promoció de la diversitat de gènere i de la igualtat d'oportunitats en l'educació superior a Espanya.

3.2.2 A RESALTAR

En la gràfica de barres que es presenta, cal ser conscient que els percentatges que s'han afegit a cada barra no estan ben visualitzats en aquest context. Aquest problema es deu a les limitacions d'aquest editor de Google Collab i com es generen els gràfics. Malauradament, no hi ha una solució fàcil per aquesta qüestió en aquesta plataforma específica.

3.2.3 CODI

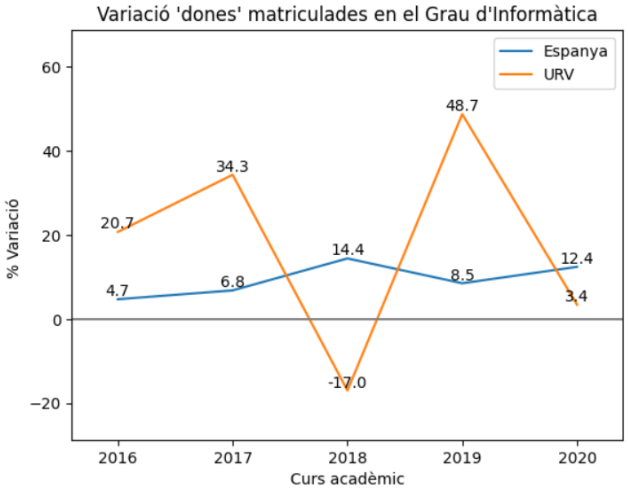
```
import matplotlib.pyplot as plt
import numpy as np
from tabulate import tabulate

# Selecciona les files i columnes desitjades
# Agafem la fila 6 dels títols als cursos
fila_seleccionada = df.iloc[6, 445:451]

# Converteix la fila en un DataFrame i la transposem
anys_a_tractar = pd.DataFrame(fila_seleccionada).T

# Restableix els índexs
anys_a_tractar.reset_index(drop=True, inplace=True)

# Selecciona les files i columnes desitjades per a la nova taula
files_seleccionades = [13,17]
columnes_seleccionades = [0, 445, 446, 447, 448, 449, 450]
taula = df.iloc[files_seleccionades, columnes_seleccionades]


# Reemplaça els valors de la columna 0 (excloent el títol)
nous_valors = ["Homes", "Dones"] # per les columnes
taula.iloc[:, 0] = nous_valors

# Itera sobre els noms de les columnes en taula1 i estableix nous noms de columna
noms_columnes = ["Genere"] + [col[:4] for col in anys_a_tractar.iloc[0]]

# Assigna els nous noms de columna a taula1
taula.columns = noms_columnes

# Converteix les columnes de la taula "taula" a números
# El paràmetre "errors='coerce'" converteix els valors no numèrics a NaN (valors faltants)
taula.iloc[:, 1:] = taula.iloc[:, 1:].apply(pd.to_numeric, errors='coerce')

final = taula.copy() # copia per modificar la taula i tractar-la per la gràfica
final.colums = noms_columnes

# llistes s per emmagatzemar etiquetes i valors  de "Homes" y "Dones"
etiquetes = []
percentatges = []

# Iterar sobre las columnas a mostrar
for idx, columna in enumerate(final.columns[1:]):
    etiqueta = columna
    etiquetes.append(etiqueta)

    # Converteix les cadenes en números decimals
    homes_valor = float(taula.iloc[0][columna])
    dones_valor = float(taula.iloc[1][columna])

    # Càlcul del percentatge
    total = homes_valor + dones_valor

    # Aquí defineixes les variables abans d'utilitzar-les
    percentatges_homes = round((homes_valor / total) * 100, 1)
    percentatges_dones = round((dones_valor / total) * 100, 1)

    percentatges.append((percentatges_homes, percentatges_dones))

    final.iloc[0, idx + 1] = percentatges_homes
    final.iloc[1, idx + 1] = percentatges_dones

dg = pd.DataFrame(final)

# Converteix el DataFrame a una taula amb bores i estil
taula = tabulate(dg, headers='keys', tablefmt='pretty', showindex=False)

# Mostra la taula
print(taula)

titol_grafica = "Variació segons gènere dels estudiants del Grau Informàtica a Espanya"

# Crear índex per las barres
index = np.arange(len(etiquetes))
```

```
# Augmentar l'alçada i l'amplada de la gràfica
plt.figure(figsize=(8, 4))

# Crear las barres per "Homes" i "Dones" en cada columna
bar_width = 0.35
bar_positions = [index - bar_width/2, index + bar_width/2]
bar_labels = ["Homes", "Dones"]

for i, bar_label in enumerate(bar_labels):
    plt.bar(bar_positions[i], [p[i] for p in percentatges], bar_width, label=bar_label)

plt.title(titol_grafica)
plt.xlabel("Curs")
plt.ylabel("Percentatge")
plt.xticks(index, etiquetes, rotation=45)  # estableix etiquetes en el eje x

# Mostrar els valors de percentatges en cada barra
for i, (ph, pm) in enumerate(percentatges):
    plt.text(index[i] - bar_width/2, ph + 1, f'{ph:.1f}%', ha='center', va='bottom')
    plt.text(index[i] + bar_width/2, pm + 1, f'{pm:.1f}%', ha='center', va='bottom')

# Ajustar la llegenda a la dreta
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))

# Mostrar la gráfica
plt.show()
```

El codi utilitzat per crear la gràfica de barres que representa l'evolució de la matriculació en el Grau d'Enginyeria Informàtica a Espanya és una combinació de Python i diverses llibreries, incloent "*matplotlib*", "*numpy*", i "*tabulate*".

- S'extreuen les dades rellevants del DataFrame per a la representació de la gràfica de barres. Això implica seleccionar les columnes que corresponen als cursos i els valors de matriculació d'homes i dones.
- Es realitza una sèrie de transformacions i neteges de dades per assegurar que les dades siguin en el format correcte per a la visualització.
- Es crea la taula de percentatges, que mostra com es reparteixen els estudiants per gènere en cada curs acadèmic.
- S'utilitza la llibreria "*matplotlib*" per crear la gràfica de barres agrupades que representa aquesta distribució. Cada barra agrupada mostra el percentatge d'estudiants homes i dones en un curs determinat.
- Es personalitza la gràfica afegint etiquetes, títols i llegendes, i es mostra la gràfica resultant.

3.3 Gràfica de línies



3.3.1 COMENTARI DE LA GRÀFICA

En aquesta secció, es realitza un anàlisi visual de la variació de la matriculació de dones en el Grau d'Enginyeria Informàtica, amb un enfocament comparatiu entre Espanya i la Universitat Rovira i Virgili (URV). L'objectiu de la gràfica de línies és identificar i visualitzar les tendències en la matriculació de dones al llarg de diversos cursos acadèmics.

La gràfica mostra la variació percentual de matriculació de dones al llarg del temps. Per fer-ho, s'ajusta el primer curs al segon valor disponible, ja que no hi ha antecedents per al primer curs.

La comparació entre Espanya i la URV permet identificar les tendències en matèria de gènere i avaluar si hi ha canvis significatius en la matriculació de dones en l'Enginyeria Informàtica. Aquesta representació visual ajuda a visualitzar les fluctuacions en la matriculació en aquests dos contextos i ofereix una visió més clara de les tendències a llarg termini.

L'anàlisi de les dades revela dues tendències significatives: una universitat manté una matriculació relativament estable al llarg del temps, mentre que l'altra mostra una variabilitat més pronunciada en la matriculació de dones.

En la Universitat Rovira i Virgili (URV), observem una tendència que varia d'un any a l'altre sense seguir una trajectòria constant. Aquesta variabilitat pot estar relacionada amb diverses raons. És important destacar que la URV és una universitat més petita en comparació amb a la mitjana d'universitats espanyoles i catalanes, i això pot influir en la dispersió dels seus números de matriculació. Les universitats més petites poden experimentar fluctuacions més grans en els seus números de matriculació, ja que els canvis en un petit nombre d'estudiants poden tenir un impacte més gran en les estadístiques. Això podria explicar la variabilitat observada en la URV.

3.3.2 CODI

```
import tabulate
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from tabulate import tabulate
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Combina les llistes de les 7 primeres columnes i les 6 a partir del número de columna trobat
# Selecciona totes les files i les primeres 7 columnes
nou_df = df.iloc[:, :]

# Selecciona les files i columnes desitjades
fila_seleccionada = df.iloc[6, 445:451]

# Converteix la fila en un DataFrame
titols = pd.DataFrame(fila_seleccionada).T

# Restableix els índexs
titols.reset_index(drop=True, inplace=True)

# Selecciona les files i columnes desitjades per a la nova taula
files_seleccionades = [17, 563] # noies a nivell espanya i noies a nivell URV
columnes_seleccionades = [0, 445, 446, 447, 448, 449, 450]
taula = nou_df.iloc[files_seleccionades, columnes_seleccionades]
```

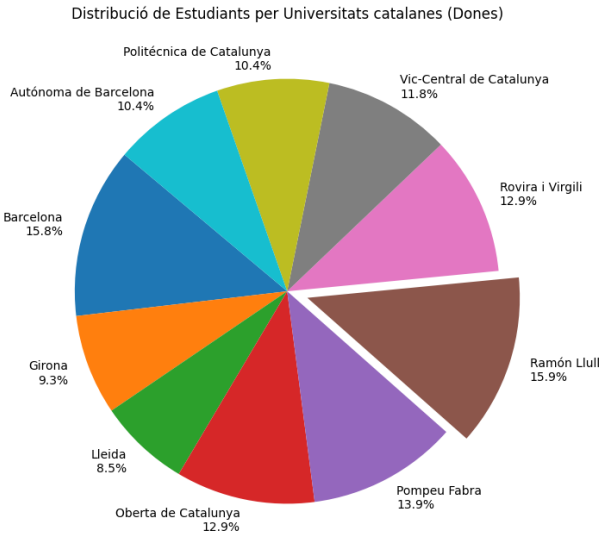
```
# Reemplaça els valors de la columna 0 (excloent el títol)
nous_valors = ["Espanya", "URV"]
taula.iloc[:, 0] = nous_valors

# Itera sobre els noms de les columnes en taula1 i estableix nous noms de columna
noms_columnes = ["Uni"] + [col[:4] for col in titols.iloc[0]]

# Assigna els nous noms de columna a taula1
taula.columns = noms_columnes

# Converteix les columnes de la taula "taula" a números
# El paràmetre "errors='coerce'" converteix els valors no numèrics a NaN (valors faltants)
taula.iloc[:, 1:] = taula.iloc[:, 1:].apply(pd.to_numeric, errors='coerce')

final = taula.copy()

for fila in taula.index:
    for col_idx, columna in enumerate(taula.columns):

        if col_idx < 1:
            continue  # Salta les dues primeres columnes
        valor_actual = taula.at[fila, columna]
        valor_anterior = taula.at[fila, taula.columns[col_idx - 1]]

        # Verifica si els valors són numèrics abans de la operació
        if isinstance(valor_actual, (int, float)) and isinstance(valor_anterior, (int, float)):
            resta = valor_actual - valor_anterior
            valor_final = round((resta / valor_anterior) * 100, 1)
            final.at[fila, columna] = round(valor_final, 1)

# Copia el DataFrame original
taula = final.copy()

# Elimina la columna 1 (en aquest cas, la columna '2015')
taula.drop('2015', axis=1, inplace=True)

# Careguem en un Dataframe per fer la taula per la gràfica
dg = pd.DataFrame(taula)

# Converteix el DataFrame a una taula amb vores i estil
taula_visual = tabulate(dg, headers='keys', tablefmt='pretty', showindex=False)

# Mostra la taula
print(taula_visual)

# Converteix totes les columnes a tipus numèric, reemplaçant els valors no numèrics per NaN
taula = taula.apply(pd.to_numeric, errors='coerce')

# Reemplaça els valors NaN per 0 (o un altre valor adequat si cal)
taula = taula.fillna(0)

# Ajusta els límits de l'eix Y per centrar el zero
max_value = max(abs(taula.values.min()), abs(taula.values.max()))
if np.isnan(max_value):
    max_value = 1  # Si max_value és NaN, establix un valor predeterminat de 1

# Crea la figura i l'eix
fig, ax = plt.subplots()

# Itera sobre les files del DataFrame per crear una línia per a cada fila
for indice, fila in enumerate(taula.index):
    etiqueta_grupo = nous_valors[indice]  # Etiqueta del grup a la columna 0 com a cadena
    valores = taula.loc[fila, taula.columns[1:]]  # Valors de les columnes a partir de la tercera columna
    etiquetes_x = taula.columns[1:]  # Etiquetes per a l'eix x

    ax.plot(etiquetes_x, valores, label=etiqueta_grupo)

# Configura el títol i les etiquetes dels eixos
plt.title("Variació 'dones' matriculades en el Grau d'Informàtica")
plt.xlabel("Curs acadèmic")
plt.ylabel("% Variació ")

# Mostra una llegenda
ax.legend()

# Ajusta els límits de l'eix Y
plt.ylim(-max_value + 20, max_value + 20)
```

```
# Mostra els valors en cada punt
for fila in taula.index:
    etiqueta_grupo = str(taula.at[fila, taula.columns[0]])
    valores = taula.loc[fila, taula.columns[1:]]
    for i, valor in enumerate(valores):
        plt.text(etiquetes_x[i], valor, str(round(valor, 2)), ha='center', va='bottom')

# Afegeix marges horitzontals i verticals
plt.margins(x=0.1, y=0.1)

# Afegeix una línia horitzontal en el valor zero
plt.axhline(0, color='gray', linestyle='-')

# Mostra el gràfic
plt.show()
```

Aquest codi realitza una sèrie de passos per crear una gràfica de línies que compara la variació en la matriculació de dones en el Grau d'Informàtica entre dues entitats: "Espanya" i "URV" (Universitat Rovira i Virgili).

- Es comença seleccionant un conjunt de dades d'un DataFrame, i es defineixen les files i les columnes que es volen utilitzar per a la comparació.
- Es crea una nova taula de dades (DataFrame) que consisteix en aquestes files i columnes seleccionades.
- Es reemplacen els valors de la primera columna de la nova taula amb les etiquetes "Espanya" i "URV", per indicar les dues entitats de comparació. S'estableixen els noms de les columnes d'aquesta nova taula per a la representació gràfica, incloent l'any de cadascuna.
- Es converteixen les dades a números i es calculen les variacions percentuals entre anys consecutius per a cada entitat.
- S'elimina una columna que no es vol representar en la gràfica.
- Es crea una representació visual de la taula de dades utilitzant la biblioteca "*Matplotlib*".
- Es configuren diversos aspectes de la gràfica, com el títol, les etiquetes dels eixos i els marges.
- Es crea la gràfica de línies amb les variacions percentuals al llarg dels anys per a "Espanya" i "URV".
- Es mostra la gràfica resultant.

3.4 Gràfica de línies



3.4.1 COMENTARI DE LA GRÀFICA

En aquesta gràfica comparativa, podem veure que no existeixen diferències molt significatives entre les universitats catalanes en matrícula en el Grau d'Enginyeria Informàtica. El fet de representar les universitats per separat facilita destacar quin valor és més alt en cada any, tot i que la diferència global entre elles no és molt pronunciada. Aquesta visualització permet analitzar i comparar les tendències de matriculació en aquest grau entre les universitats catalanes al llarg dels anys acadèmics.

3.4.2 CODI

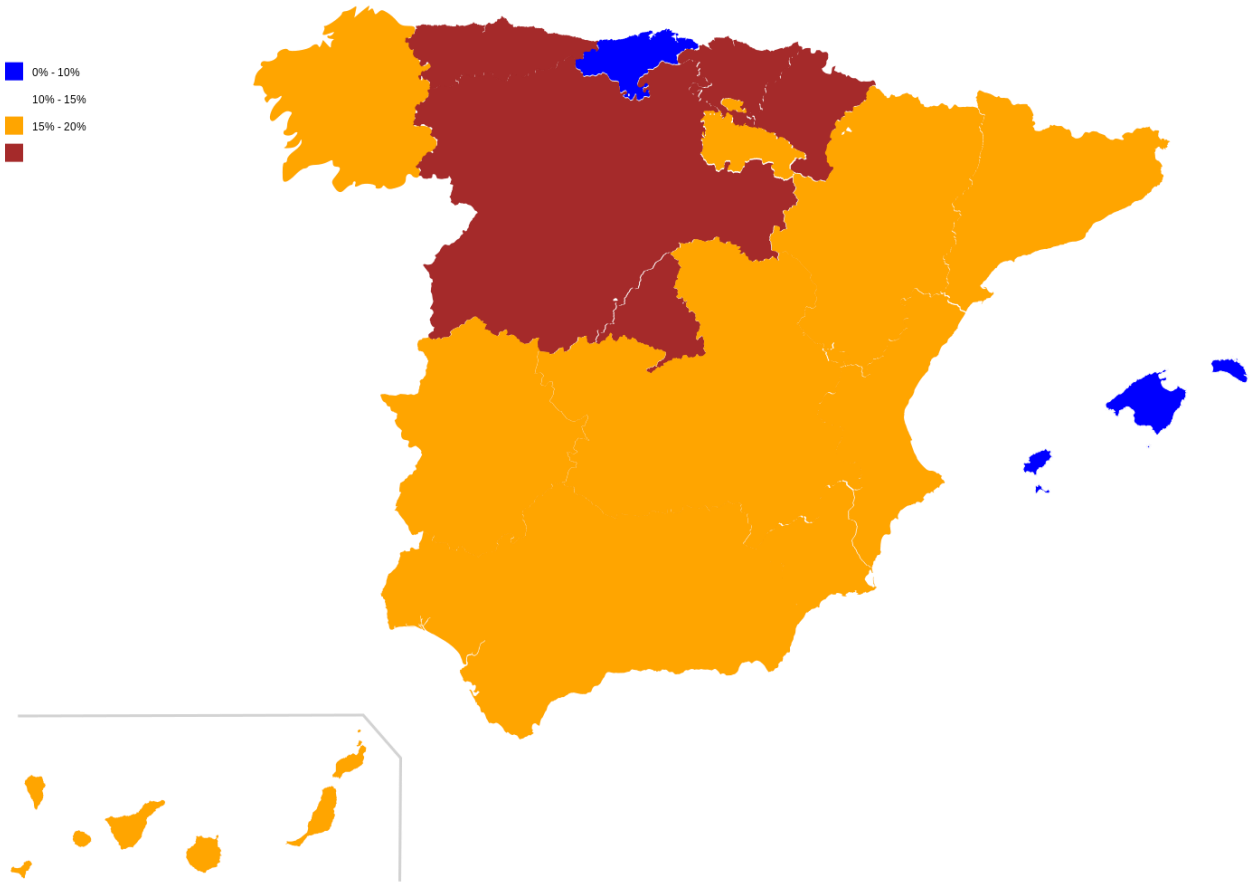
```
import pandas as pd
import numpy as np
import pandas as pd
from tabulate import tabulate
import matplotlib.pyplot as plt

valor_a_comprobar = 'Catalunya'

# Com la Comunitat de Catalunya es la columna 6 agafem totes les universitats d ela columna 6
llista_cat = df_comunitats.iloc[:, 6]  # Accede a la columna 6 (ten en cuenta que el índice comienza en 0)

# Crea un nou DataFrame a partir de la llista
df_catalanes = pd.DataFrame({llista_cat[0]: llista_cat[1:]}).T

# Assigna els títols de les colmnes a patir de la 1a fila
df_catalanes.columns = df_catalanes.iloc[0]

# Elimina la primera fila (índex 0 ) que s'acaben d'asignar com a títols
df_catalanes = df_catalanes[1:]

# Restableix els índex del dataframe
df_catalanes.reset_index(drop=True, inplace=True)

# Elimina els valors nuls es a dir NaN (valors buits) d'una columna en aquest cas la columna 1 que hem trasposat
abans
df_catalanes = df_catalanes.dropna(axis=1)
df_catalanes.reset_index(drop=True, inplace=True)

# Crea un nuevo DataFrame "unis_cat" amb la primera columna igual als valors de la columna 7  -> copia
unis_cat = pd.DataFrame({df_catalanes.columns[0]: df_catalanes.columns})
unis_cat = unis_cat.rename(columns={df_catalanes.columns[0]: "Universitat"})

unis_cat = unis_cat.iloc[0:11,:]

# Creem una colmna pels valors delshomes i una altre pels vaors de les dones
unis_cat[2] = 0
unis_cat[3] = 0

#posem títols a les columnes de dades que omplirem
unis_cat.columns = ["Universitat","Homes","Dones"]

fila_inicial = 46 # La prmera Universitat 2A Coruña" esta a la fila 46 ( 2 menys de la numeració)
interval = 13     # D'una universitat ala seguent hi han 13 files

# mirem només l'última any per comparar entre universitats
numero_columna = 450  # Les dades del curs 20-21 d'Informàtica estan a la columna 450

# Combina les llistes de les 7 primeres columnes i les 6 a partir del número de columna d'Informatica (clmna
445 endabant)
# afaga la llista de universitats i es posa en un nou DataFrame
columnas_combinades = df[[df.columns[0], df.columns[450]]] # Nomes agafarem les dades del darrers curs del
fitxer (2020-2021)  que sera la columna 450 i no la 445 que es la del curs 2015-2016

# Crea un nou DataFrame amb les 6 columnes unificades
df_unis = columnas_combinades

# Afegim una nova fila per omplir els valors inicials
# afegir el llistat de universitats
valor  = df_unis.iloc[fila_inicial, 0]


# Crearem  una llista para emmagatzemar els valors trobats s
valores_encontrados = []

# loop per recorrer les unis
while fila_inicial < len(df_unis):
    #  no en blanc
    if not pd.isna(df_unis.iloc[fila_inicial, 0]):
        nom_uni = df_unis.iloc[fila_inicial, 0]

        # per les dades dels homes
        if fila_inicial + 6 < len(df_unis) and not pd.isna(df_unis.iloc[fila_inicial + 6, 1]):
            valor_homes = df_unis.iloc[fila_inicial + 6, 1]
            if isinstance(valor_homes, str):
                valor_homes = pd.to_numeric(valor_homes, errors='coerce')
        else:
            valor_homes = "Valor no disponible"
```

```
        # per les dades de les dones
        if fila_inicial + 10 < len(df_unis) and not pd.isna(df_unis.iloc[fila_inicial + 10, 1]):
            valor_dones = df_unis.iloc[fila_inicial + 10, 1]
            if isinstance(valor_dones, str):
                valor_dones = pd.to_numeric(valor_dones, errors='coerce')
        else:
            valor_dones = "Valor no disponible"

        # ens assegurem que la universitat pertany a Catalunya
        fila_index = unis_cat[unis_cat.iloc[:, 0] == nom_uni].index

        # no buit
        if not fila_index.empty:

            for index in fila_index:
                valor_actual_homes = pd.to_numeric(unis_cat.iloc[index, 1], errors='coerce')
                valor_actual_dones = pd.to_numeric(unis_cat.iloc[index, 2], errors='coerce')

                if not pd.isna(valor_actual_homes) and not pd.isna(valor_actual_dones):
                    unis_cat.iloc[index, 1] = valor_homes
                    unis_cat.iloc[index, 2] = valor_dones

    fila_inicial += interval

# Crear un DataFrame a partir de les dades
unis_graf = pd.DataFrame(unis_cat)

# Calcular elspercentatges i afegir una nova columna
unis_graf['Homes'] = round((unis_cat['Homes'] / (unis_cat['Homes'] + unis_cat['Dones'])) * 100,1) # Arrodonim
a 1 decimal
unis_graf['Dones'] = round((unis_cat['Dones'] / (unis_cat['Homes'] + unis_cat['Dones'])) * 100,1)

# Eliminem la columna dels Homes ja que la comparativa es amb el genere femení
unis_graf = unis_cat[unis_cat['Homes'] != 0]

dg = pd.DataFrame(unis_graf)

# Converteix el DataFrame a una taula amb vores i estil
taula_valors = tabulate(dg, headers='keys', tablefmt='pretty', showindex=False)

print(taula_valors)

# Filtra les files on "Dones" té un valor superior a 0
unis_filtradas = unis_graf[unis_graf['Dones'] > 0]  # per evitar errors

# Crea les llistes a partir de les files filrades s
unis = unis_filtradas['Universitat'].tolist()
dones = unis_filtradas['Dones'].tolist()
dones = [float(x) for x in dones]  # Convertim a float per poder evitar problemes de tipus de dades per
operacions


# Obté l'índex del valor més gran de la columna "Dones"
max_index_dones = unis_graf['Dones'].idxmax()  ##Escollim quina Univrsitat serà la més destacada

# Crea una llista buida d'explosió
explode = [0] * len(unis)

# Assigna un valor d'explosió més gran a la posició del valor més gran
explode[max_index_dones-1] = 0.1

# Configura la gràfica circular
plt.figure(figsize=(8, 8))
plt.pie(dones, labels=[f"{uni}\n{valor}%" for uni, valor in zip(unis, dones)], explode=explode, startangle=140)

# Configura el títol
plt.title("Distribució de Estudiants per Universitats catalanes (Dones)")

plt.show()
```

Aquest codi realitza el següent:

- Importa les biblioteques necessàries per a la manipulació de dades i la visualització, com *Pandas*, *NumPy*, *Tabulate* i *Matplotlib*.
- Defineix una variable "*valor_a_comprobar*" que emmagatzema el valor "*Catalunya*".
- Accedeix a una columna específica (la columna 6, que correspon a la Comunitat de Catalunya) del conjunt de dades original i crea una nova llista anomenada "*llista_cat*".
- Crea un nou DataFrame anomenat "*df_catalanes*" a partir de la llista "*llista_cat*".
- Assigna els títols de les columnes a partir de la primera fila del nou DataFrame i elimina aquesta fila, ja que ara són títols de les columnes.
- Restableix els índexs del DataFrame perquè comencin des de zero.
- Elimina els valors nuls (NaN) de les columnes del DataFrame.
- Crea un nou DataFrame anomenat "*unis_cat*" amb dues columnes: "Universitat" i dues columnes buides per als valors d'homes i dones.
- Posa noms a les columnes del DataFrame.
- Defineix variables com la "*fila_inicial*", l' "*interval*", i el "*numero_columna*" per configurar els paràmetres de la recopilació de dades.
- Combina les dades d'una sèrie de columnes específiques (columna 0 i columna 450) del conjunt de dades original en un nou DataFrame anomenat "*df_unis*".
- Inicia un bucle per recórrer les universitats i les seves dades, com els valors d'homes i dones. El bucle busca dades vàlides per a cada universitat.
- Comprova si les universitats pertanyen a Catalunya i actualitza les dades corresponents d'homes i dones.
- Crea un DataFrame "*unis_graf*" amb les dades recopilades i calcula els percentatges.
- Filtra les universitats en les quals el valor de "Dones" sigui superior a 0.
- Crea llistes a partir de les dades dels "Homes" i les "Dones" de les universitats filtrades.
- Troba l'índex de la universitat amb el valor de "Dones" més gran i defineix un valor d'explosió més gran per destacar-la en un gràfic de pastís.
- Configura un gràfic de pastís amb les dades d'unió i un títol.
- Mostra el gràfic de pastís resultant.

3.5 Gràfica mapa



3.5.1 COMENTARIS DE LA GRÀFICA

La distribució de la matriculació en el Grau d'Informàtica per part de les noies a tot l'estat espanyol mostra una tendència sorprenentment uniforme. Aquesta uniformitat és notable en el fet que la proporció de noies que cursen aquest grau és relativament similar en totes les Comunitats Autònomes de l'Estat espanyol. Tot i aquesta uniformitat, hi ha una lleugera excepció que crida l'atenció, la qual es troba a la zona nord-est del país.

La gran majoria de les Comunitats Autònomes d'Espanya mostren una distribució prou equitativa entre els gèneres en la matriculació al Grau d'Informàtica, amb proporcions que oscil·len al voltant del 15% de noies per aquest grau. Això reflecteix una relativa igualtat en l'interès i la participació de gènere en aquest camp acadèmic a tot el país.

No obstant això, com s'ha esmentat, es pot observar un lleuger increment en la proporció de noies que cursen el Grau d'Informàtica a la zona nord-est de l'estat espanyol. Aquesta observació sugereix la possibilitat que en aquesta regió específica, les noies mostren un interès lleugerament més gran per les carreres relacionades amb la informàtica en comparació amb altres regions d'Espanya. Encara que aquest augment sigui lleuger, és un indicatiu interessant i podria ser d'interès per a les polítiques educatives i les universitats a l'hora d'entendre i fomentar la diversitat de gènere en el camp de la informàtica a Espanya.

3.5.2 CODI

```
import pandas as pd
import warnings
from tabulate import tabulate
from IPython.display import display, Image
!pip install cairosvg
import xml.etree.ElementTree as ET
import cairosvg

# Busca el número de columna d'inici informàtica
num_columna = 445

# Combina les llistes
columnes_combinades = df.columns[:7].tolist() + df.columns[num_columna:num_columna + 6].tolist()

# Crea un DataFrame amb una sola coljmna que contingui les columnes combinades
columnes_combinades_df = pd.DataFrame({'Noms de columnes': columnes_combinades})

# Crear la taula "resum" amb la primera columna com títols de "comunitats"
resum = pd.DataFrame({'comunitats': df_comunitats.columns[0:]})

# Cambiar els noms de les columnes
resum = resum.rename(columns={'comunitats': 'Comunitat', 'homes': 'homes', 'dones': 'dones', "%Dones" :
"%dones"})

# Inicialitzar les columnes 2 i 3 en "resum" amb zeros
resum['homes'] = 0
resum['dones'] = 0
resum['%Dones'] = 0
resum['Color'] ="white"

# Iterar sobre les files de df
numero_columna=450

for fila in range(46, len(df), 13):
    if fila < len(df):
        valor_combinat = df.iloc[fila, 0]
        valor_combinat = str(valor_combinat)


        # Inicializar las variables para sumar los valores
        suma_homes = 0
        suma_dones = 0

        # Crear una lista per emmagatzemar les columnes de "comunitats" on es troba el valor_combinat
        columnes_coincidents = []

        # Iterar sobre les columnes de "comunitats" per buscar coincidències
        for columna in df_comunitats.columns[1:]:

            if df_comunitats[columna].str.contains(valor_combinat).any():

                columnes_coincidents.append(columna)
                posicio_columna = df_comunitats.columns.get_loc(columna)

        # Sumar els valors de "df" a "resum"
              # Obtenir els valors en format de cadena
                if fila + 10 < len(df) and numero_columna < len(df.columns):
                  suma_homes_str = df.iloc[fila + 6, 450]
                  suma_dones_str = df.iloc[fila + 10, 450]

                  valor_actual_homes_str = resum.at[posicio_columna, 'homes']
                  valor_actual_dones_str = resum.at[posicio_columna, 'dones']

                # Convertir els valors de cadena a numèrics
                  suma_homes = pd.to_numeric(suma_homes_str, errors='coerce')
                  suma_dones = pd.to_numeric(suma_dones_str, errors='coerce')
                  valor_actual_homes = pd.to_numeric(valor_actual_homes_str, errors='coerce')
                  valor_actual_dones = pd.to_numeric(valor_actual_dones_str, errors='coerce')

                # Realitzar la suma i actualitzar els valors a "resum"
                  resum.at[posicio_columna, 'homes'] = suma_homes + valor_actual_homes
                  resum.at[posicio_columna, 'dones'] = suma_dones + valor_actual_dones
                  valor = (suma_dones/ (suma_dones + suma_homes))* 100
                  resum.at[posicio_columna, '%Dones'] = round(valor,1)

colors = ["white","yellow","orange","brown"]
```

```
# Calcular el resultat de la operació i afegir-lo com una nova columna
resum['%Dones'] = round(resum['dones'] / (resum['dones'] + resum['homes']) * 100,1)
resum['%Dones'] = resum['%Dones'].fillna(0)

# Definim una funció per assignar el color en funció del valor de "%Dones"
def assign_color(row):
    colors = ["blue","yellow","orange","brown"]
    if row['%Dones'] == 0:
        return colors[0]
    if  row['%Dones'] <= 10:
        return colors[1]
    if row['%Dones'] <= 15:
        return colors[2]
    else:
        return colors[3]

# Aplica la funció a cada fila de la columna "Color"
resum['Color'] = resum.apply(assign_color, axis=1)

nuevos_valores = ['Andalucia', 8943, 1389, 13.4,'orange']

resum.iloc[0] = nuevos_valores

dg = pd.DataFrame(resum)

# Converteix el DataFrame a una taula amb vores i estil
taula_valors = tabulate(dg, headers='keys', tablefmt='pretty', showindex=False)

print(taula_valors)

filename = '/content/drive/MyDrive/Colab Notebooks/Treball_MV/mapa.svg'

# Modificar el color del poligon segons els valors en la taula resum
tree = ET.parse(filename)
root = tree.getroot()

for path in root.findall('.//{http://www.w3.org/2000/svg}path'):
    if 'id' in path.attrib and path.attrib['id'] in resum['Comunitat'].tolist():
        Comunitat = path.attrib['id']
        color = resum.loc[resum['Comunitat'] == Comunitat, 'Color'].iloc[0]
        path.set('style', f'fill:{color}')

# Iterar a través de les capes "polygon" i canviar el fons
for polygon in root.findall('.//{http://www.w3.org/2000/svg}polygon'):
    if 'id' in polygon.attrib and polygon.attrib['id'] in resum['Comunitat'].tolist():
        Comunitat = polygon.attrib['id']
        color = resum.loc[resum['Comunitat'] == Comunitat, 'Color'].iloc[0]
        polygon.set('style', f'fill:{color}')

for g in root.findall('.//{http://www.w3.org/2000/svg}g'):
    if 'id' in g.attrib and g.attrib['id'] in resum['Comunitat'].tolist():
        Comunitat = g.attrib['id']
        color = resum.loc[resum['Comunitat'] == Comunitat, 'Color'].iloc[0]
        for child in g:
            if 'style' in child.attrib and 'fill' in child.attrib['style']:
                child.set('style', f'fill:{color}')

tree.write(filename)

png_filename = '/content/drive/MyDrive/Colab Notebooks/Treball_MV/mapa0.png'

cairosvg.svg2png(url=filename, write_to=png_filename)

Image(png_filename)

tree = ET.parse(filename)
root = tree.getroot()

for path in root.findall('.//{http://www.w3.org/2000/svg}path'):
    if 'id' in path.attrib and path.attrib['id'] == 'Cantabria':

        # Modificar el color de relleno
        path.set('style', 'fill:#0000ff')  # Cambia a verde (#00ff00) en este ejemplo

# Guardar l'arxiu SVG modificat
tree.write(filename)

# Ruta per guardar la imatge PNG
png_filename = '/content/drive/MyDrive/Colab Notebooks/Treball_MV/mapa.png'
```

```
# Convertir l'arxiu SVG a PNG
cairosvg.svg2png(url=filename, write_to=png_filename)

Image(png_filename)
```

Aquest codi realitza diverses tasques relacionades amb la manipulació i visualització de dades en un arxiu SVG d'un mapa d'Espanya.

- S'importen les llibreries necessàries, com *Pandas* per a la manipulació de dades, *warnings* per a la gestió de missatges d'advertència, *tabulate* per a mostrar dades en forma de taula, i altres llibreries per a processar i visualitzar l'arxiu SVG.
- Creació de la variable "*num_columna*": Aquesta variable indica el número de columna des del qual es troben les dades del Grau d'Informàtica a l'arxiu de dades. En aquest cas, és la columna 445.
- Combinació de les llistes: Es crea una nova llista anomenada "*columnes_combinades*" que conté les columnes de l'arxiu de dades corresponents a les 7 primeres columnes i a les següents 6 columnes relacionades amb el Grau d'Informàtica.
- Es crea un DataFrame anomenat "*columnes_combinades_df*" que té una sola columna, "Noms de columnes", que conté els noms de les columnes combinades.
- Es crea un DataFrame anomenat "*resum*" amb la columna "*comunitats*", que inicialment conté els noms de les comunitats autònomes d'Espanya.
- S'actualitzen els noms de les columnes a "Comunitat", "homes", "dones" i "%Dones" en el DataFrame "*resum*".
- Es creen les columnes "homes", "dones", "%Dones" i "Color" i s'inicialitzen amb valors 0 i "white" en el DataFrame "resum".
- Es recorren les files de l'arxiu de dades, i a partir de certes files (cada 13 files) s'obtenen els valors corresponents al número de columna 450, que representa les dades del Grau d'Informàtica per a l'any 2020-2021. Es realitza la suma dels valors d'homes i dones per a cada comunitat autònoma i s'actualitzen les dades al DataFrame "resum".
- Es calcula el color a assignar a cada comunitat en funció del valor de "%Dones". Si el valor és 0, s'assigna el color "blue". Si el valor està entre 0 i 10, s'assigna el color "*yellow*". Si el valor està entre 10 i 15, s'assigna el color "*orange*". Per als altres valors, s'assigna el color "*brown*".
- L'arxiu SVG del mapa d'Espanya és processat per canviar el color de les regions (comunitats autònomes) en funció del color calculat en el pas anterior. Això es fa mitjançant l'ús de l'arxiu SVG i la modificació del valor "fill" dels elements corresponents a les regions del mapa.
- L'arxiu SVG amb els colors de les regions modificats es guarda en un arxiu i es genera una imatge PNG a partir de l'arxiu SVG. Aquesta imatge es mostra al final del codi.