



# Carles M. Bosch Herrera

## Anàlisi de 3 tècniques de visualització de dades (PAC2)

### ENUNCIAT

L'objectiu d'aquesta segona prova d'avaluació continuada consisteix en crear tres petites visualitzacions usant les 3 tècniques les diferents que han estat assignades pel vostre professor amb la selecció. Cada estudiant haurà d'escollar unes dades idònies per a cada una de les tècniques proposades i decidir amb quin programari les crea. Les dades es podran escollir de qualsevol font de dades obertes. L'estudiant publicarà les representacions a Internet i les presentarà en un vídeo. Per a publicar les representacions cal que aquestes siguin accessibles directament amb una URL, sense necessitat de registres ni inicis de sessió. Cadascuna de les tres presentacions han de tenir una durada d'uns 2-3 minuts. En total, si es fa un vídeo únic amb les tres tècniques aquest no pot excedir els 7 minuts. Es penalitzarà sortir d'aquest rang, perquè es valora la capacitat de síntesi i de comunicació. Es recomana fer un vídeo estàndard, és a dir amb pantalla principal per a la visualització i finestra petita per a la cara del/de la presentador/a, vosaltres heu d'aparèixer al vídeo en algun moment. El vídeo ha d'incloure els apartats següents (marcats en groc):

**1. [10%]** Presenteu-vos seguint l'esquema de la PAC 1. Es a dir, indiqueu el vostre nom i cognoms i dieu que esteu presentant la vostra PAC 2 que fa referència a l'anàlisi de les 3 tècniques que us han estat assignades pel professor de l'assignatura de Visualització de Dades del Màster Universitari de Ciència de Dades de la UOC. Recordeu incloure els elements següents: títol, descripció, crèdits de la visualització, data de publicació.

Bona dia. El meu nom és Carles M. Bosch Herrera. Actualment estic cursant el Màster en Ciència de Dades de la UOC i aquest semestre curso entre d'altres l'assignatura de Visualització de Dades.

En relació amb la segona prova d'avaluació continuada estudiaré 3 tècniques de visualització de dades. Una de les eines fonamentals per a estadística o matemàtica són les gràfiques, i és que aquestes són representacions visuals capaces de plasmar dades conceptuais o numèriques. Així mateix, també són eficients per transmetre informació administrativa, demogràfica, científica i tecnològica. En aquesta PAC2 de visualització de dades tinc assignades les 3 tècniques de visualització que presentaré a continuació: Sankey diagrams, Hexagonal binning i Icon Chart.

**2. [1+2: 5%]** Indiqueu on està penjada la visualització per poder accedir-hi.

El vídeo de visualització i el pdf-resum corresponen a la PAC2 han estat lliurat a l'aula i el vídeo corresponent a la visualització d'aquesta PAC2 esta disponible a l'adreça següent del meu GitHub: cboschh/VD-PACs: VD-PACs (github.com.)

L'enllaç corresponent està a l'adreça següent: <https://github.com/cboschh/VD-PACs/>



### 3. [30%] Definir cada tècnica de visualització de forma general: nom, origen, descripció/funcionament, exemples d'aplicació, etc.

En aquest tercer apartat es defineix cada tècnica d'una manera general indicant el seu nom, la seva descripció i funcionament i alguns exemples d'aplicació.

#### SANKEY DIAGRAMS

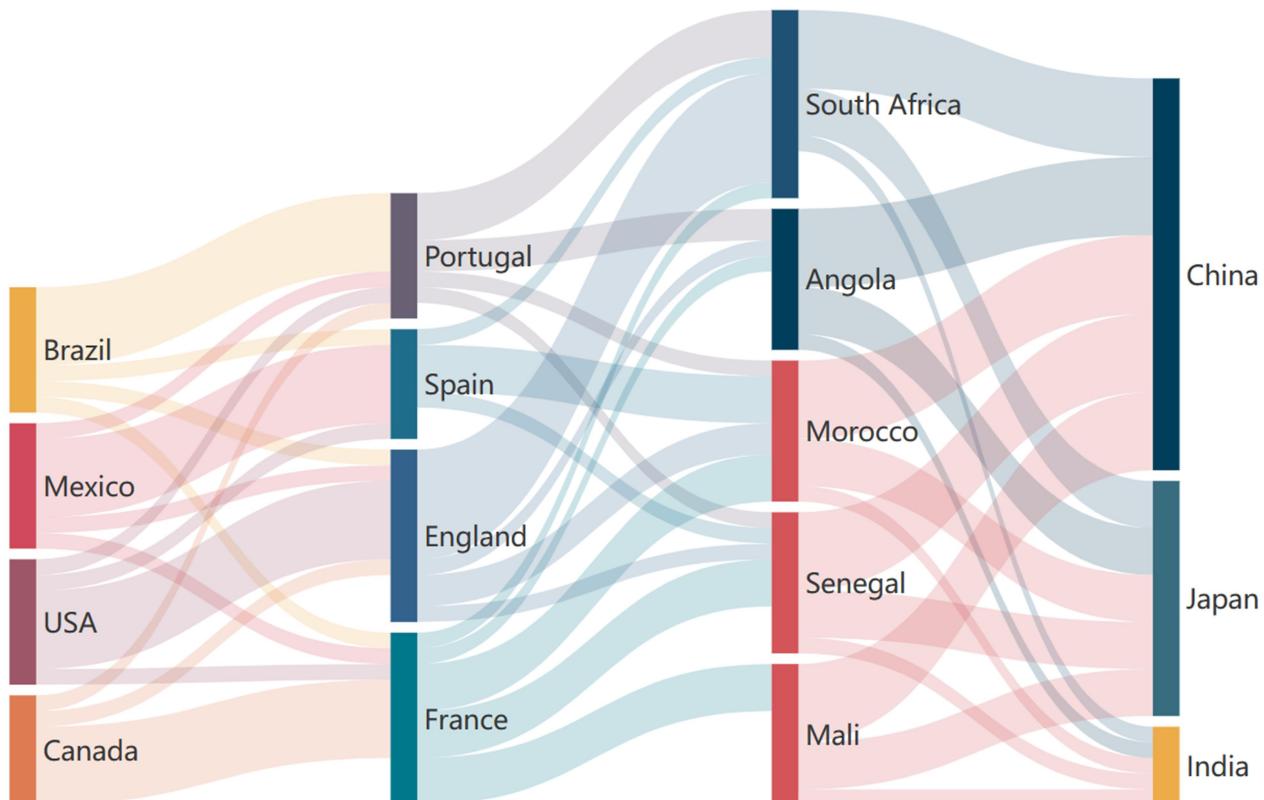
Els **Sankey diagrams** són una eina de visualització de dades que representa el flux de valors d'una etapa a una altra mitjançant nodes i enllaços.

Els nodes representen les etapes, i els enllaços o bandes mostren com es divideix el flux a cada etapa, sent l'amplada d'aquests proporcionals al flux que passa d'un node a un altre.

Alguns exemples d'aplicació són: la visualització de la cadena de subministrament, l'anàlisi de fluxos d'energia o recursos, la representació de fluxos financers o [flux d'emigració entre països](#).

Aquesta darrera aplicació és la que he triat per la meva visualització. He creat un codi en Python que representa mitjançant un Sankey diagram el flux migratori entre diversos països. L'amplada de les connexions que hi ha representa més o menys flux migratori

En el gràfic següent es mostren diferents països (nodes) i els enllaços o bandes que els uneixen d'amplada proporcional al que representen (flux migratori)





## HEXAGONAL BINNING

El diagrama **Hexagonal Binning** és una tècnica per visualitzar la densitat d'un conjunt de punts. Utilitza hexàgons per crear una graella regular d'hexàgons en el pla XY i desenvolupa un histograma espacial.

El pla XY es divideix en una graella regular d'hexàgons. Els punts comptabilitzats dins de cada hexàgon es representen mitjançant un esquema de colors o variant el radi de l'hexàgon segons el nombre de punts.

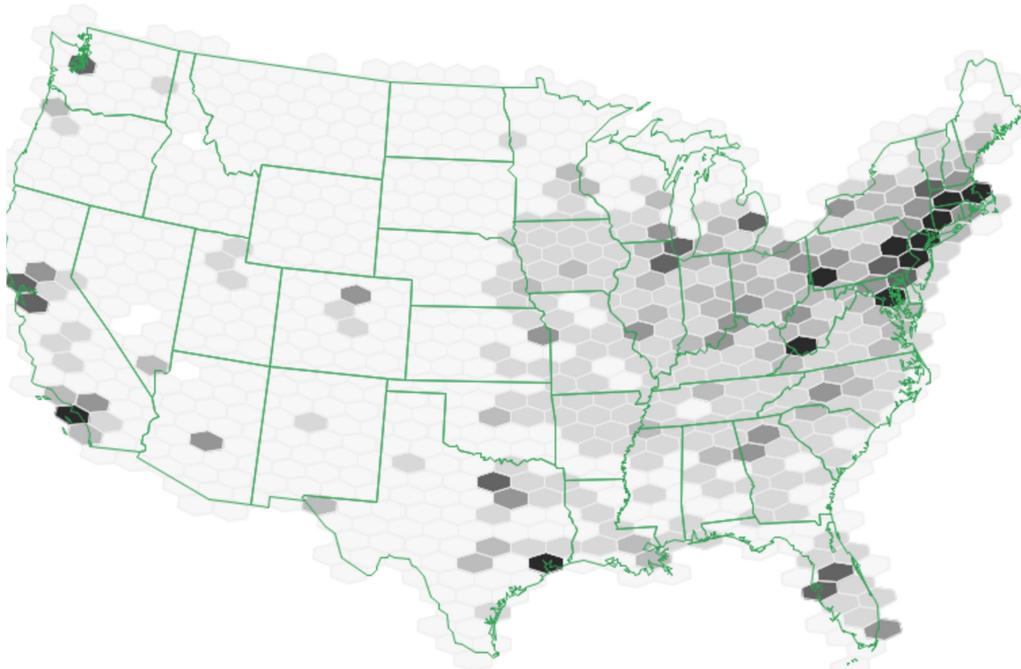
El més freqüent és augmentar la intensitat del color dependent de la densitat dels punts que es representen (un color intens representa una major densitat de punts i menys intens el contrari)

Alguns **exemples d'aplicació** serien la visualització de dades geoespitals o [estudis de densitat de població en mapes](#).

Un mapa Hexbin alguns cops pot ser confús de visualitzar. De fet, és més difícil establir l'enllaç amb un mapa real, ja que no hi ha punts de referència geogràfics habituals. La definició d'un gràfic Hexbin es basa en un objecte geoespacial no massa comú, on totes les regions del mapa es representen com a hexàgons.

Per exemple podríem representar el mapa dels EUA, on la densitat dels codis postals (Zip Code) cada localitat vindria representat com un hexàgon. El codi ZIP és el sistema de codis postals que utilitza el Servei Postal dels Estats Units. Les lletres ZIP són en anglès un acrònim de “Zone Improvement Plan”, i s’han d'escriure en lletres majúscules. El terme ZIP va ser elegit per suggerir que el correu postal viatja en forma més eficient, i per tant més ràpid, quan els remitents utilitzen el codi.

**Geographic Density of US Zip Codes**

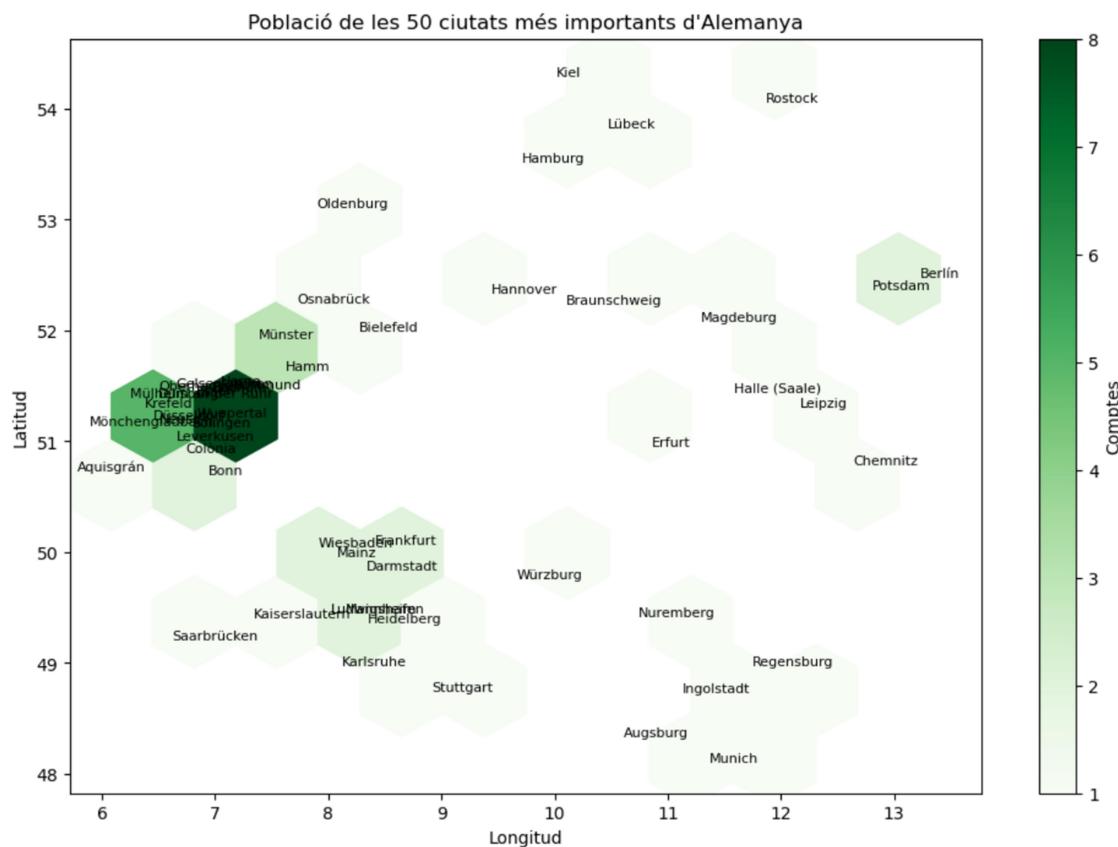




En aquest gràfic es mostra com es distribueixen els codis postals a diferents estats dels Estats Units. Veiem que la major densitat de codis postals es troben a les costes est i oest (hexàgons més intensos).

L'entrada que cal fer servir de les dades proporciona un valor per cada estat i un fitxer de forma específica amb límits hexagonals.

Es precisament la darrera aplicació que he dit ([estudis de densitat de població en mapes](#)) la que he triat per la meva visualització. He creat un codi en Python que representat la població dels 16 Länders alemanys en un diagrama Hexagonal Binning i mostra també la població de les 50 ciutats més importants d'Alemanya. La intensitat del color dels hexàgons existents per a cada localització indica més o menys població. En el cas dels Länders aquests es veuen similars, però en el cas de les ciutats es veu clarament aquells lloc on la població és més elevada,



## ICON CHART

L'**Icon Chart** és un conjunt d'ícones amb la mateixa forma (sovint cercles o quadrats). Hi ha diferents tipus de gràfics que ens serveixen per fer la representació visual de dades com són: ícones, pictogrames, imatges, símbols, etc.,



**Una icona és la representació d'un objecte o cosa de manera similar o semblant al que representa.** La paraula icona prové de la paraula grega “eikon” que significa imatge o indicí.

Aquest terme s'utilitza per donar referències a imatges, signes i símbols. Un exemple podrien ser les emoticons que fem servir a aplicacions com Whatsapp o Telegram, les quals tenen una amplia carta d'ícones (Icon Chart). Les icones tenen més llibertat estètica, mentre que els pictogrames són més esquemàtics i universals en la seva representació,

La **taula d'emoticons** com a exemple de Icon Chart és l'exemple que he triat per la meva visualització i he comparat els emoticons que representen els estats d'ànim de les persones amb fotografies real son es pot veure com els emoticons representen de manera semblant l'estat d'ànim de les persones representades a les fotografies



Aqui, a continuació, es presenta un exemple d'una Icon Chart que mostra una graella amb 24 ícones de significat clarament identifiable:





A les 3 icones següents es comparen imatges reals amb les seves respectives icones que els representen. Veiem que, clarament, es manté una relació de semblança amb l'objecte representat.



**4. [10%]** Descriure el tipus de dades que es poden representar amb cada tècnica (dades quantitatives, qualitatives? Quina estructura han de tenir per a cada tècnica?). Explicar les limitacions quant a dades (hi ha mida mínima i màxima del joc de dades per a cada tècnica?).

En aquest quart apartat es descriuen el tipus de dades que utilitza cada tècnica., quina és l'estructura de les dades utilitzades i possibles limitacions de cada tècnica i de les dades emprades.

### SANKEY DIAGRAMS

**Tipus de dades:** En els Sankey diagrams el tipus de dades que s'utilitzen tant dades qualitatives (per cada un dels elements d'origen i de destí) com dades quantitatives per designar el valor corresponent al flux de dades entre origen i destí. Els Sankey diagrams van molt bé per representar fluxos o relacions entre diferents categories. Això pot incloure dades com ara transferències de fons, moviments de mercaderies, fluxos d'energia o fluxos migratoris entre d'altres,

**Estructura:** L'estructura que presenten els diagrames d'aquest tipus ve donada per 2 elements importants: els nodes que representen les categories o etapes (origen i destí) i els enllaços que connecten els nodes i mostren el flux entre ells. La seva amplada és proporcional a la quantitat de flux. L'exemple que he triat és, com he dit, el flux migratori entre diferents països.

**Limitacions i mida de les dades:** En quan a les limitacions i la mida de les dades en els Sankey diagrams tenim el [nombre de nodes](#). Si hi ha massa nodes, el text i els gràfics es tornen minúsculs o cal fer desplaçament, la qual cosa no és ideal. Una altra limitació és que [no són adequats per a representar dades contínues, com ara dades temporals](#), llevat que es converteixin en dades categòriques o discretes (com anys o períodes de temps específics). En quan a la [grandària del conjunt de dades](#) s'han d'evitar conjunts de dades massa grans o massa disperses tot i que en realitat no trobem que hi hagi una mida específica, però és important mantenir que el diagrama sigui manejable per evitar problemes de llegibilitat.



## HEXAGONAL BINNING

**Tipus de dades:** En els Hexagonal Binning el tipus de dades que es fan servir són de tipus quantitatiu ja que els diagrames Hexagonal Binning es fan servir per visualitzar la densitat de punts en un espai bidimensional. Són molt adequats per a dades quantitatives com ara coordenades geogràfiques, intensitat de trànsit, població en un país o regió, etc. L'exemple que he triat és, com he dit, la població en els diferents Länders o en les 50 ciutats més importants d'Alemanya.

**Estructura:** L'estructura d'un diagrama Hexagonal Binning consisteix en una espècie de graella hexagonal en la que el pla XY es divideix en hexàgons regulars i en ella cada hexàgon compta els punts que cauen dins seu. L'exemple que he triat és, com he dit, es la població dels Länders i de les 50 ciutats més importants d'Alemanya.

**Limitacions i mida de les dades:** En quan a les limitacions i la mida de les dades en els Hexagonal Binning ens podríem trobar amb el [problema de l'overplotting](#): Els Hexagonal Binning són útils per gestionar aquest problema de la superposició excessiva de punts en una zona determinada. No obstant això, si el [conjunt de dades és massa gran](#), pot ser difícil de comprendre què correspon a cada localització com m'ha passat a la visualització triada de la població de les ciutats alemanyes on a la zona est es pot veure una elevada densitat de punts. Crec que no existeix una mida específica, però [s'han de tenir en compte les dimensions de la graella hexagonal i la quantitat de punts per mantenir una visualització clara](#).

## ICON CHART

**Tipus de dades:** En els **Icon Chart** no es representen dades sinó més bé [es representen imatges](#). En els Icon Chart representem un objecte o cosa de manera similar o semblant al que representa com és el cas de les emoticons que expressen estats d'ànim que és l'exemple que he triat comparant alguns emoticons que expressen estats d'ànim amb fotografies reals que mostren l'estat d'ànim de diferents persones. En altres tipus d'imatges gràfiques com els pictogrames es poden utilitzar dades quantitatives per representar quantitats o dades qualitatives utilitzant colors per representar categories o variables addicionals.

**Estructura:** En quan a estructura les diferents icones les podem disposar en una graella tal com podem veure en aplicacions com Whatsapp o Telegram.

**Limitacions i mida de les dades:** En quan a les limitacions i la mida de les dades en els Icon Chart ens podem trobar que tinguem un [espai limitat](#) per a representar-les. La mida generalment petita de les icones limita la quantitat d'informació que es pot mostrar. Si hi ha massa icones, l'espai pot tornar-se insuficient. [No permeten representar detalls molt complexos](#) ja que recordem que una icona és només un tipus d'imatge que intent esquematitzar i tenir semblança amb l'objecte que representa. En un Icon Chart [cal mantenir una quantitat raonable d'icones dintre de l'espai del que disposem per a evitar la saturació visual](#).



**5. [20%] Fer una representació amb cadascuna de les tècniques usant un conjunt de dades obertes (vegeu més avall algunes fonts de dades obertes). En total cal fer 3 representacions simples amb 3 conjunts de dades escollits per l'estudiant.**

En aquest cinquè apartat mostro el programa creat per representar cada tècnica i la visualització corresponent que s'obté com a resultat.

Els codis corresponents, creats en Python, per crear cada visualització, i que es mostren també aqui també estan disponibles (juntament amb la seva conversió a pdf) al lloc web on he penyat el vídeo de la visualització. El lloc web on es pot accedir està a GitHub a l'adreça següent: <https://github.com/cboschh/VD-PACs/>

## SANKEY DIAGRAMS

### Sankey diagrams (fluxos migratoris entre païssos)

Importem les llibreries

```
In [1]: import plotly.graph_objects as go
import matplotlib.pyplot as plt
```

Creem les dades

```
In [2]: etiquetes = ["Brasil", "Mèxic", "USA", "Canadà", "Portugal", "Espanya", "UK", "França", "Sudàfrica", "Angola", "Marroc", "Senegal", "Mali", "Xina", "Japó", "Índia"]
index = [ 0,      1,      2,      3,      4,      5,      6,      7,      8,      9,      10,     11,     12,     13,     14,     15]
origen = [0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3,
          4, 5, 6, 7, 4, 5, 6, 7, 4, 5, 6, 7, 4, 5, 6, 7,
          8, 9, 10, 11, 12, 8, 9, 10, 11, 12, 8, 9, 10, 11, 12, 8, 9, 10, 11, 12]
desti = [4, 4, 4, 4, 5, 5, 5, 5, 6, 6, 6, 6, 7, 7, 7, 7,
         8, 8, 8, 8, 9, 9, 9, 9, 10, 10, 10, 10, 11, 11, 11, 11, 12, 12, 12, 12, 12,
         13, 13, 13, 14, 14, 14, 15, 15, 15]
valors = [7, 4, 1, 1, 1, 5, 1, 2, 1, 1, 5, 1, 1, 2, 1, 5,
           3, 1, 2, 2, 0, 1, 3, 1, 0, 0, 1, 2, 1, 4, 0, 1, 3, 2, 1, 3,
           3, 3, 2, 5, 3, 1, 5, 3, 1, 5, 3, 2, 5, 3, 1]
```

Imprimim el diagrama de Sankey

```
In [3]: fig = go.Figure(go.Sankey(
    node=dict(
        pad=15,
        thickness=20,
        line=dict(color="black", width=0.5),
        label=etiquetes,
    ),
    link=dict(
        source=origen,
        target=desti,
        value=valors
    )
))

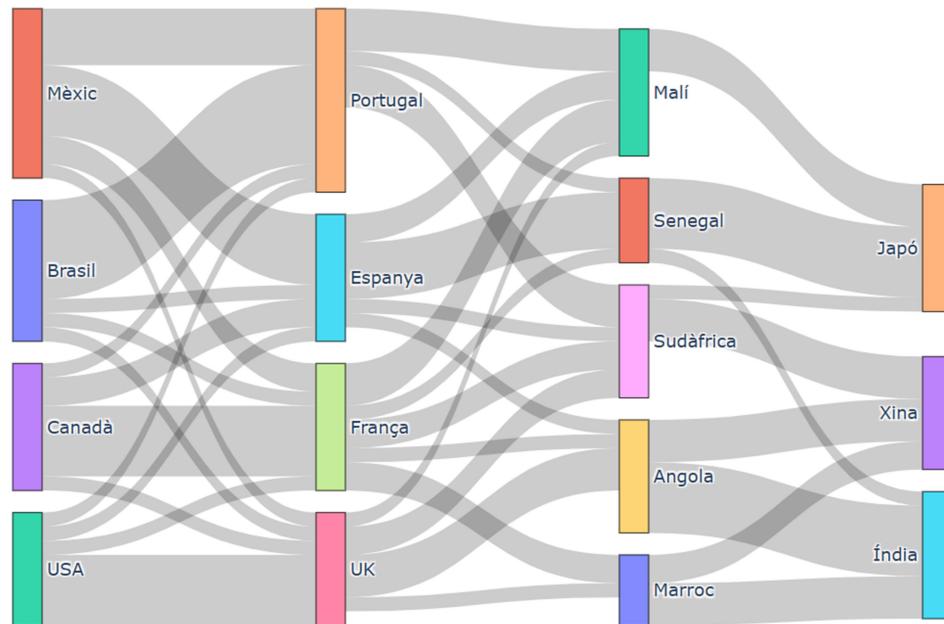
# Poso un títol al gràfic i l'ajusto per a que quedi central
fig.update_layout(title="Flux migratori entre païssos", title_x=0.5, width=800, height=600)

# Mostro el gràfic
fig.show()
```

L'exemple que he triat per aquesta primera visualització utilitzant els **diagrames de Sankey** és el flux migratori entre diferents països. Per la qual cosa he creat un programa en Python a partir de les dades que he trobat pels diferents països d'origen, els països de destí i el valor d'aquest flux migratori. El resultat per aquest tipus de diagrama que resulta d'executar el programa anterior mostra el flux migratori entre els diferents països

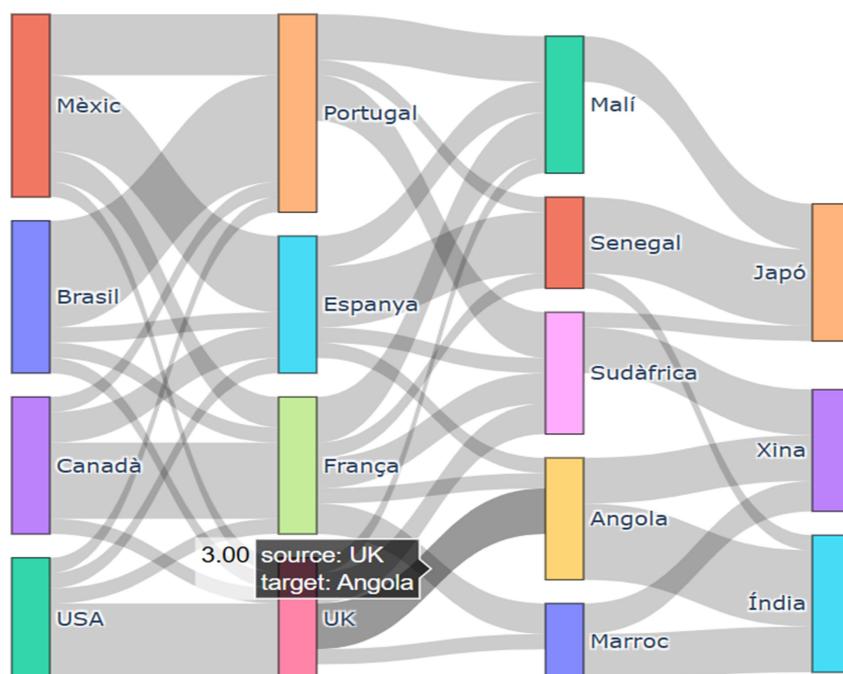


Flux migratori entre païssos



Si passem per damunt de les connexions es mostra el país d'origen, el país de destí i el valor del flux migratori com es mostra a l'exemple següent entre UK i Angola al que li correspon un flux de 3.0

Flux migratori entre païssos





## HEXAGONAL BINNING

La segona tècnica assignada correspon al **Hexagonal Binning**. L'exemple que he triat per aquest tipus de diagrama és l'estudi de la població dels diferents Länders o estats d'Alemanya. [Com ja et vaig comentar a l'inici del semestre he intentat fer un estudi complet amb dades que he trobat d'aquest país per tal de realitzar la PAC3 i la Pràctica, cosa que ja vaig començar a fer el mes que teníem lliure abans de l'inici del semestre.](#)

Per fer aquest tipus de diagrama he creat dos visualitzacions i dos programes en Python a partir de les dades que he per la població d'Alemanya i que després he ajuntat amb un mateix programa perquè es mostrin els resultats un a continuació de l'altre. Tot i que les dues visualitzacions són molt semblants mostren resultats lleugerament diferents La primera part de l'estudi fa referència a la població que trobem en els 16 Länders o Estats d'Alemanya (on el diagrama Hexagonal Binning veiem que no mostra gaires diferències significatives en el total de la població de cada un dels Estats).

### Hexagonal Binning diagrams: Població per a cada un dels 16 Länders o Estats d'Alemanya

Importem les llibreries

```
In [1]: import matplotlib.pyplot as plt
import numpy as np
```

Creem les dades de població

```
In [2]: # Dades de la població per a cada un dels 16 Länders o Estats d'Alemanya
poblacio_landers = {
    "Baden-Württemberg": 11100394,
    "Baviera": 13124737,
    "Berlín": 3669491,
    "Brandenburg": 2521893,
    "Bremen": 681202,
    "Hamburg": 1847253,
    "Hesse": 6288080,
    "Mecklemburg-Pomerània Occidental": 1608138,
    "Baixa Saxònia": 7982448,
    "Rin del Nord-Westfàlia": 17947221,
    "Renània-Palatinat": 4093903,
    "Saarland": 990509,
    "Saxònia": 4077937,
    "Saxònia-Anhalt": 2194782,
    "Slesvig-Holstein": 2903773,
    "Turíngia": 2133378
}
```

Obtenim les coordenades reals per a cada un dels 16 Länders o Estats d'Alemanya

```
In [3]: # Coordenades reals per a cada un dels 16 Länders o Estats d'Alemanya
coordenades = {
    "Baden-Württemberg": (48.6616, 9.3501),
    "Baviera": (48.7904, 11.4979),
    "Berlín": (52.5200, 13.4050),
    "Brandenburg": (52.4125, 12.5316),
    "Bremen": (53.0793, 8.8017),
    "Hamburg": (53.5511, 9.9937),
    "Hesse": (50.6521, 8.3188),
    "Mecklemburg-Pomerània Occidental": (53.6127, 12.4296),
    "Baixa Saxònia": (52.6367, 9.8451),
    "Rin del Nord-Westfàlia": (51.4332, 7.6616),
    "Renània-Palatinat": (49.9929, 7.2354),
    "Saarland": (49.3964, 6.7561),
    "Saxònia": (51.1045, 13.2017),
    "Saxònia-Anhalt": (51.9503, 11.6923),
    "Slesvig-Holstein": (54.2194, 9.6961),
    "Turíngia": (50.8278, 11.9866)
}
```



### Recupero les coordenades per cada un dels Länders

```
In [4]: # Recupero les coordenades per a cada Lander
x = [coordenades[state][1] for state in poblacio_landers.keys()]
y = [coordenades[state][0] for state in poblacio_landers.keys()]
```

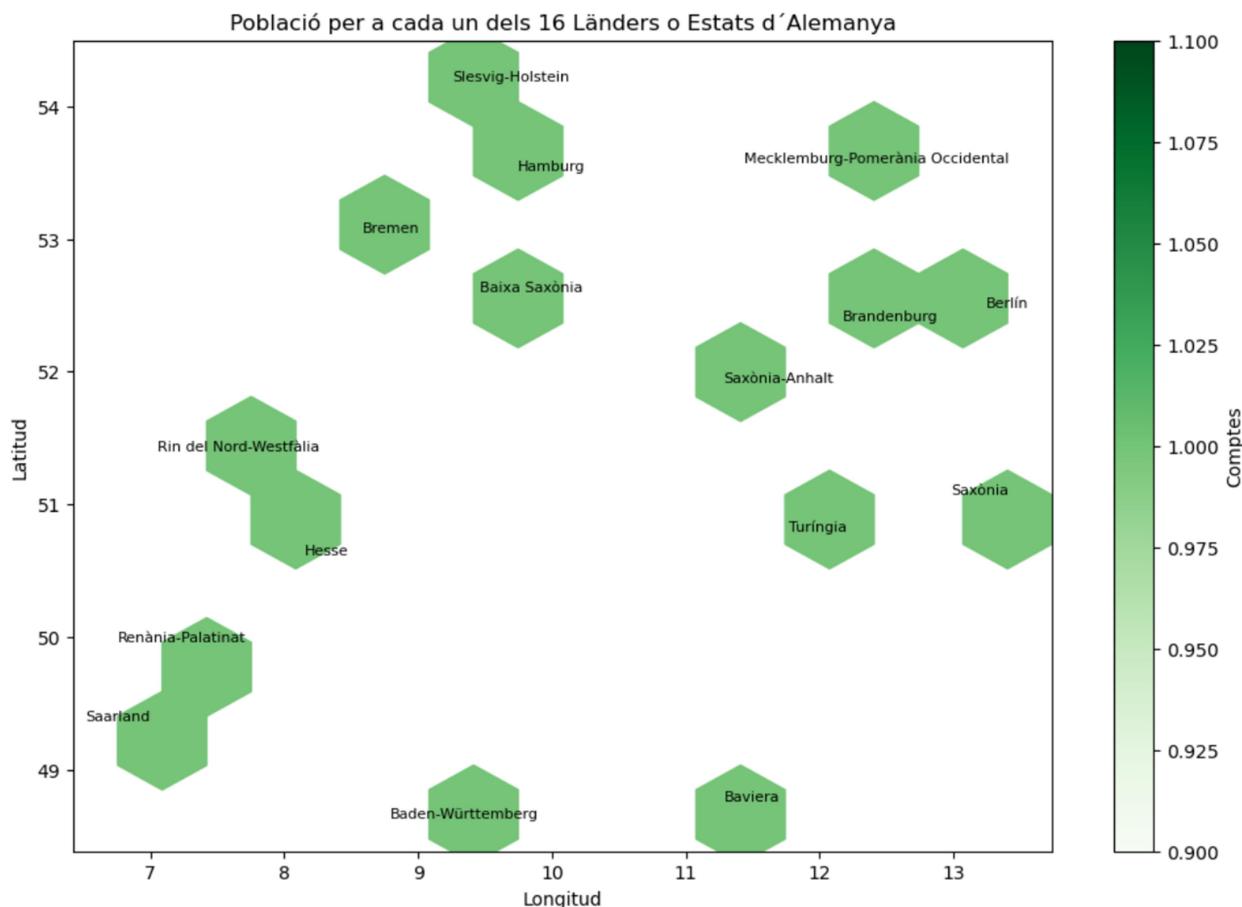
### Creem el gràfic hexagonal bin sobre el mapa d'Alemanya

```
In [5]: # Creem el gràfic hexagonal bin sobre el mapa d'Alemanya
fig, ax = plt.subplots(figsize=(12, 8))
hb = ax.hexbin(x, y, gridsize=10, cmap='Greens', mincnt=1)

# Etiquetem els punts amb els noms dels estats
for i, state in enumerate(poblacio_landers.keys()):
    ax.text(x[i], y[i], state, ha='center', va='center', fontsize=8)

ax.set_title("Població per a cada un dels 16 Länders o Estats d'Alemanya")
ax.set_xlabel("Longitud")
ax.set_ylabel("Latitud")
cb = fig.colorbar(hb, ax=ax, label='Comptes')

# Mostrem el gràfic (Hexagonal Binning Gràfic)
plt.show()
```





## Hexagonal Binning diagrams: Població de les 50 ciutats més importants d'Alemanya

A la segona part de l'estudi el diagrama Hexagonal Binning l'estudi fa referència a les 50 ciutats més importants d'Alemanya i podem veure clarament que la majoria de la població s'agrupa a la zona est d'Alemanya, on es troben grans nuclis de població com Colònia, Bonn, Munic o Düsseldorf on es mostren els hexàgons en un color més intens i també suposo que hi ha superposicions que fan que el color es mostri més intens.

Altres grans poblacions com Berlin a la dreta o Frankfurt o Heidelberg al sud que també es mostren amb un color més intens.

A la resta de les ciutats estudiades el color dels hexàgons és molt més clar i això representa que aquestes poblacions tenen menys població. Volia representar els hexàgons damunt del mapa d'Alemanya però no he aconseguit fer-ho.

### Importem les llibreries

```
In [6]: import matplotlib.pyplot as plt
import numpy as np
```

### Creem les dades de població

```
In [7]: # Dades de la població per a cada ciutat (en milions)
poblacio_ciutats = {
    "Berlín": 3.7,
    "Hamburg": 1.8,
    "Munich": 1.8,
    "Frankfurt": 0.7,
    "Stuttgart": 0.6,
    "Colònia": 1.1,
    "Düsseldorf": 0.6,
    "Dortmund": 0.6,
    "Essen": 0.6,
    "Leipzig": 0.6,
    "Hannover": 0.5,
    "Nuremberg": 0.5,
    "Duisburg": 0.5,
    "Bochum": 0.3,
    "Wuppertal": 0.3,
    "Bielefeld": 0.3,
    "Bonn": 0.3,
    "Mannheim": 0.3,
    "Karlsruhe": 0.3,
    "Münster": 0.3,
    "Augsburg": 0.3,
    "Wiesbaden": 0.2,
    "Gelsenkirchen": 0.2,
    "Mönchengladbach": 0.2,
    "Braunschweig": 0.2,
    "Chemnitz": 0.2,
    "Kiel": 0.2,
    "Aquisgrán": 0.2,
    "Magdeburg": 0.2,
    "Krefeld": 0.2,
    "Halle (Saale)": 0.2,
    "Kaiserslautern": 0.2,
    "Oberhausen": 0.2,
    "Lübeck": 0.2,
    "Erfurt": 0.2,
    "Rostock": 0.2,
    "Mainz": 0.2,
```



```
"Hamm": 0.2,  
"Saarbrücken": 0.2,  
"Herne": 0.2,  
"Mülheim an der Ruhr": 0.2,  
"Osnabrück": 0.2,  
"Solingen": 0.2,  
"Ludwigshafen": 0.2,  
"Leverkusen": 0.2,  
"Oldenburg": 0.2,  
"Neuss": 0.2,  
"Potsdam": 0.2,  
"Heidelberg": 0.2,  
"Darmstadt": 0.2,  
"Regensburg": 0.2,  
"Ingolstadt": 0.2,  
"Würzburg": 0.2  
}
```

Obtenim les coordenades reals per a cada una de les 50 principals ciutats d'Alemanya

```
In [8]: # Coordenades reals per les 50 ciutats més importants d'Alemanya  
coordenades = {  
    "Berlín": (52.5200, 13.4050),  
    "Hamburg": (53.5511, 9.9937),  
    "Munich": (48.1351, 11.5820),  
    "Frankfurt": (50.1109, 8.6821),  
    "Stuttgart": (48.7758, 9.1829),  
    "Colònia": (50.9375, 6.9603),  
    "Düsseldorf": (51.2277, 6.7735),  
    "Dortmund": (51.5136, 7.4653),  
    "Essen": (51.4556, 7.0116),  
    "Leipzig": (51.3397, 12.3731),  
    "Hannover": (52.3759, 9.7320),  
    "Nuremberg": (49.4521, 11.0767),  
    "Duisburg": (51.4344, 6.7623),  
    "Bochum": (51.4818, 7.2195),  
    "Wuppertal": (51.2562, 7.1508),  
    "Bielefeld": (52.0302, 8.5325),  
    "Bonn": (50.7374, 7.0982),  
    "Mannheim": (49.4875, 8.4660),  
    "Karlsruhe": (49.0069, 8.4037),  
    "Münster": (51.9607, 7.6261),  
    "Augsburg": (48.3705, 10.8978),  
    "Wiesbaden": (50.0782, 8.2398),  
    "Gelsenkirchen": (51.5177, 7.0857),  
    "Mönchengladbach": (51.1801, 6.4425),  
    "Braunschweig": (52.2689, 10.5268),  
    "Chemnitz": (50.8278, 12.9214),  
    "Kiel": (54.3233, 10.1228),  
    "Aquisgrán": (50.7753, 6.0839),  
    "Magdeburg": (52.1205, 11.6276),  
    "Krefeld": (51.3381, 6.5853),  
    "Halle (Saale)": (51.4828, 11.9699),  
    "Kaiserslautern": (49.4431, 7.7688),  
    "Oberhausen": (51.4964, 6.8638),  
    "Lübeck": (53.8650, 10.6866),  
    "Erfurt": (50.9848, 11.0299),  
    "Rostock": (54.0924, 12.0991),  
    "Mainz": (49.9929, 8.2473),  
    "Hamm": (51.6747, 7.8164),  
    "Saarbrücken": (49.2402, 6.9969),  
    "Herne": (51.5388, 7.2283),  
    "Mülheim an der Ruhr": (51.4309, 6.8807),  
    "Osnabrück": (52.2799, 8.0472),  
    "Solingen": (51.1704, 7.0543),  
    "Ludwigshafen": (49.4816, 8.4351),  
    "Leverkusen": (51.0456, 7.0047),  
    "Oldenburg": (53.1435, 8.2146),  
    "Neuss": (51.2045, 6.6850),  
    "Potsdam": (52.4009, 13.0598),
```



```

    "Heidelberg": (49.3988, 8.6724),
    "Darmstadt": (49.8786, 8.6516),
    "Regensburg": (49.0134, 12.1016),
    "Ingolstadt": (48.7666, 11.4256),
    "Würzburg": (49.7913, 9.9534)
}
```

Recupero les coordenades per a cada una de les ciutats

```
In [9]: # Recupero les coordenades per a cada ciutat
x = [coordenades[ciutat][1] for ciutat in poblacio_ciutats.keys()]
y = [coordenades[ciutat][0] for ciutat in poblacio_ciutats.keys()]
```

Creem el gràfic hexagonal bin sobre el mapa d'Alemanya

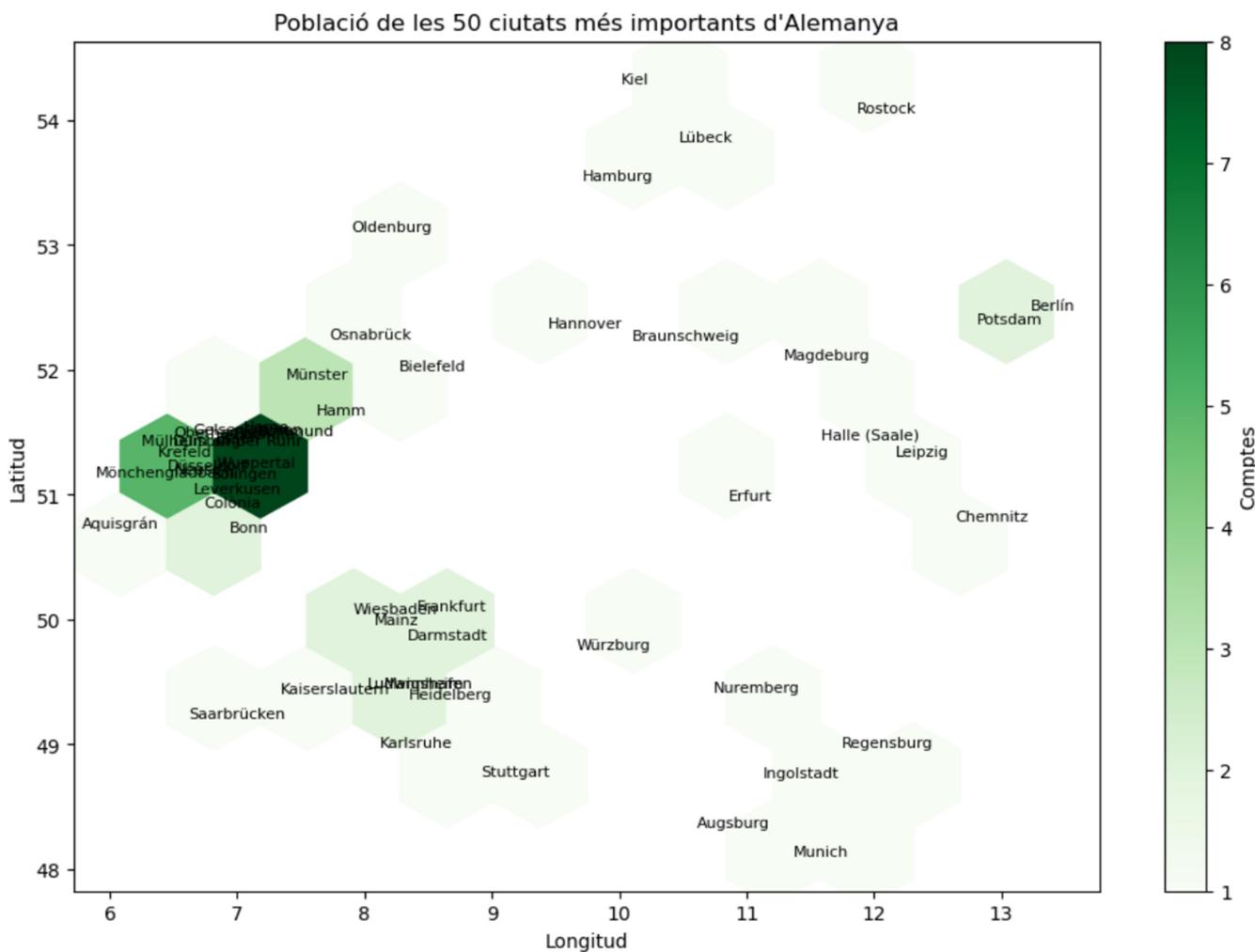
```

In [10]: # Creem el gràfic hexagonal bin sobre el mapa d'Alemanya
fig, ax = plt.subplots(figsize=(12, 8))
hb = ax.hexbin(x, y, gridsize=10, cmap='Greens', mincnt=1)

# Etiquetem els punts amb els noms dels estats
for i, ciutat in enumerate(poblacio_ciutats.keys()):
    ax.text(x[i], y[i], ciutat, ha='center', va='center', fontsize=8)

ax.set_title("Població de les 50 ciutats més importants d'Alemanya")
ax.set_xlabel("Longitud")
ax.set_ylabel("Latitud")
cb = fig.colorbar(hb, ax=ax, label='Comptes')

# Mostrem el gràfic (Hexagonal Binning Gràfic)
plt.show()
```





## ICON CHART

# ICON CHART

## Estats d'ànim comparant fotografies de persones vs les emoticons corresponents.

A continuació es mostra una infografia que he creat mitjançant una taula representant diferents estats d'ànim de les persones i que conté imatges reals que he trobat per Internet i les he comparat amb l'icana corresponent.

Com podem comprovar una icona és un signe o figura que té semblança amb l'objecte que representa i a la taula podem veure que presenten una semblança força evident

La icon chart vindria representada per la tercera columna la qual es correspondria amb algunes icons que surten a la taula d'icons corresponent que es mostra a l'inici.

```
In [ ]: import os
from skimage.transform import resize
import matplotlib.pyplot as plt
```

```
In [ ]: from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
In [ ]: os.chdir('/content/drive/My Drive/')
```

**Ara per a mostrar les imatges cal seguir 3 passos:**

- Carregar de la imatge i redimensionar-la
- Ajustar el tamany del canvas de Matplotlib
- Representar la imatge

```
In [ ]: # Carregar de la imatge i redimensionar-la
# -----
# Carrego La imatge anomenada '0.jpg' des del meu Drive al programa.
# Aquesta imatge es guarda com una matriu de pixels.
img0 = plt.imread('0.jpg')
# Redimensiono La imatge per tampteig a una mida de 800x850 pixels utilitzant la funció resize.
# La imatge redimensionada La guardo a La variable res.
res = resize(img0, (800, 850))

# Ajustar el tamany del canvas de Matplotlib
# -----
# Defineixo La resolució en punts per polzada (dots per inch, DPI) per al gràfic.
dpi = 100

# Calculo Les dimensions del pixels del canvas en polzades.
# En aquest cas, el canvas tindrà 800x850 pixels.
figsize = 800 / dpi, 850 / dpi

# Representació de La imatge
# -----
# Creo una nova figura (gràfic) amb les dimensions especificades.
plt.figure(figsize=figsize, dpi=dpi)
# Mostro La imatge redimensionada a La figura.
plt.imshow(res)
# Ometo els eixos (etiquetes i marques) a La visualització.
plt.axis('off')
# Mostro La figura amb La imatge.
plt.show()
```



```
In [ ]: # Carregar de la imatge i redimensionar-la
# -----
# Carrego la imatge anomenada '2.jpg' des del meu Drive al programa.
# Aquesta imatge es guarda com una matriu de pixels.
img1 = plt.imread('1.jpg')
# Redimensiono la imatge per tampteig a una mida de 1000x1500 pixels utilitzant la funció resize.
# La imatge redimensionada la guardo a la variable res.
res = resize(img1, (1000, 1500))

# Ajustar el tamany del canvas de Matplotlib
# -----
# Defineixo la resolució en punts per polzada (dots per inch, DPI) per al gràfic.
dpi = 100

# Calculo les dimensions del pixels del canvas en polzades.
# En aquest cas, el canvas tindrà 1000x1500 pixels.
figsize = 1000 / dpi, 1500 / dpi

# Representació de la imatge
# -----
# Creo una nova figura (gràfic) amb les dimensions especificades.
plt.figure(figsize=figsize, dpi=dpi)
# Mostro la imatge redimensionada a la figura.
plt.imshow(res)
# Ometo els eixos (etiquetes i marques) a la visualització.
plt.axis('off')
# Mostro la figura amb la imatge.
plt.show()
```

**ICON-CHART**

Estats d'ànim comparant fotografies de persones vs les emoticons corresponents.

ESTAT D'ÀNIM	IMATGE REAL	ICONA
Callat		
Cansat		
Content		

```
In [ ]: # Carregar de la imatge i redimensionar-la
# -----
# Carrego la imatge anomenada '2.jpg' des del meu Drive al programa.
# Aquesta imatge es guarda com una matriu de píxels.
img2 = plt.imread('2.jpg')
# Redimensiono la imatge per tampteig a una mida de 1550x1240 pixels utilitzant la funció resize.
# La imatge redimensionada la guardo a la variable res.
res = resize(img2, (1550, 1240))

# Ajustar el tamany del canvas de Matplotlib
# -----
# Defineixo la resolució en punts per polzada (dots per inch, DPI) per al gràfic.
dpi = 100

# Calculo les dimensions del píxels del canvas en polzades.
# En aquest cas, el canvas tindrà 1550x1240 pixels.
figsize = 1550 / dpi, 1240 / dpi

# Representació de la imatge
# -----
# Creo una nova figura (gràfic) amb les dimensions especificades.
plt.figure(figsize=figsize, dpi=dpi)
# Mostro la imatge redimensionada a la figura.
plt.imshow(res)
# Ometo els eixos (etiquetes i marques) a la visualització.
plt.axis('off')
# Mostro la figura amb la imatge.
plt.show()
```



Enamorat		
Enfadat		
Malalt		
Plorós		
Molt content		
Orgullós		
Pensatiu		
Preocupat		



```
In [ ]: # Carrega de La imatge i redimensionar-La
# -----
# Carrego La imatge anomenada '3.jpg' des del meu Drive al programa.
# Aquesta imatge es guarda com una matriu de píxels.
img3 = plt.imread('3.jpg')
# Redimensiono La imatge per tampteig a una mida de 1000x1000 pixels utilitzant la funció resize.
# La imatge redimensionada La guardo a La variable res.
res = resize(img3, (1000, 1000))

# Ajustar el tamany del canvas de Matplotlib
# -----
# Defineixo la resolució en punts per polzada (dots per inch, DPI) per al gràfic.
dpi = 100

# Calculo les dimensions del píxels del canvas en polzades.
# En aquest cas, el canvas tindrà 1000x1000 pixels.
figsize = 1000 / dpi, 1000 / dpi

# Representació de La imatge
# -----
# Creo una nova figura (gràfic) amb les dimensions especificades.
plt.figure(figsize=figsize, dpi=dpi)
# Mostro La imatge redimensionada a La figura.
plt.imshow(res)
# Ometo els eixos (etiquetes i marques) a La visualització.
plt.axis('off')
# Mostro La figura amb La imatge.
plt.show()
```

Rabiós		
Somnolent		
Tímid		
Trist		
Espantat		



Sorprès		
---------	--	--

### Icon Chart (estats d'ànim)

L'exemple que he triat per aquesta tercera visualització, com he comentat abans a la pregunta 3, ha estat fer una taula a partir d'un **Icon Chart** molt conegut, la taula d'emoticons que tots coneixem i fem servir amb aplicacions com Whatsapp o Telegram.



L'Icon Chart creat representa estats d'ànim comparant fotografies de persones vs les emoticons corresponents.

Aquesta graella mostrada que he creat amb Python representa diferents estats d'ànim de les persones i que conté imatges reals que he trobat per Internet que les he comparat amb l'icana corresponent. Com podem comprovar **una icona és un signe o figura que té semblança amb l'objecte que representa** i a la taula podem veure que presenten una semblança força evident,

**Compte! No cal confondre les icons amb els símbols, pictogrames o senyals.**

**6. [30%] Comentar breument les tres representacions indicant què es representa i qui o quins aspectes mostra o demostra cada representació.**

A aquest sisè apartat, tot i que ja ha estat comentat indico aquí, molt breument, el que s'ha representat i alguns aspectes que s'han mostrat en cada representació.

### SANKEY DIAGRAM

El **Sankey diagram** creat a partir del codi en Python que s'ha mostrat, es **mostra el flux migratori entre països**. Els nodes es representen amb rectangles o text, i les fletxes o arcs de



diferent amplada mostren els fluxos entre els diferents països. L'amplada dels enllaços és proporcional a la importància del flux. El gràfic mostra el flux de persones que emigren d'un país (origen) a un altre (destí)

### HEXAGONAL BINNING

Els dos **Hexagonal Binning** creats a partir del codi en Python que s'han mostrat, **visualitzen la població dels 16 Länders i de les 50 ciutats principals d'Alemanya**. S'utilitza una graella hexagonal per agrupar les dades i crear una mena d'histograma espacial. El gràfic mostra la distribució de població en els Länders i en les 50 ciutats principals alemanyes representades.

### ICON CHART

**Icon Chart.** Finalment, pel darrer apartat, **es representa un Icon Chart molt conegut, la taula d'emoticones** que tots coneixem i fem servir amb aplicacions com Whatsapp o Telegram. Respecte a aquesta Icon Chart, que tots coneixem, podem veure a la graella creada, exemples que representen l'estat d'ànim de les persones de manera "força precisa" tal com es pot veure a la següent Icon Chart i comparant amb imatges de fotografies reals que he cercat per Internet,

**Recordem que una icona és un signe o figura que té semblança amb l'objecte que representa.** Com s'ha pogut comprovar, la semblança que presenta cada icona és molt evident si les comparem amb les fotografies de persones reals

L'**Icon Chart**, que he mostrat a la graella creada, compara emoticones amb fotografies reals. Cada icona representa una emoció o estat d'ànim i a la graella es mostren 15 estats d'ànim, fotografies de persones amb aquest estats anímics i les icons (emoticones) que els representen. El gràfic creat, com s'ha vist, mostra emoticones que representen cada icona al costat de fotografies reals que expressen les mateixes emocions que a les fotografies

**7. [5%] Cadascuna de les tres presentacions han de tenir una durada d'uns 2-3 minuts. En total, el vídeo (únic) amb les tres tècniques no pot excedir els 7 minuts. Es penalitzarà sortir d'aquest rang, perquè es valora la capacitat de síntesi i de comunicació.**

La durada total del vídeo s'ha ajustat a 7 minuts

### **WEBGRAFIA I FONTS CONSULTADES:**

Com a la PAC1, en aquest document adjunt, mostro totes les pàgines web i recursos que he consultat per la elaboració d'aquesta PAC2. També he consultat les altres fonts de dades obertes recomanades a l'enunciat.



## GENERAL

<https://datavizproject.com/>

<https://www.data-to-viz.com/>

<https://datavizcatalogue.com/>

[https://lucid.app/documents#/templates?folder\\_id=home](https://lucid.app/documents#/templates?folder_id=home)

<https://www.bbp.cat/index.php?>

<https://es.slideshare.net/marisolpr64/las-grficas-y-tablas-en-la-investigacin-cientfica-n>

[https://www.eldiario.es/cultura/arte/incipnables-visualizacion-datos\\_1\\_4794759.html](https://www.eldiario.es/cultura/arte/incipnables-visualizacion-datos_1_4794759.html)

<https://plantillaarbolgenealogico.net/diagramas/tipos-de-graficos/>

## DIAGRAMES SANKEY

<https://online.visual-paradigm.com/app/diagrams/#infoart:proj=0&type=SankeyDiagrams&gallery=/repository/1de56f68-0fd1-4420-a047-56eff26c2742.xml&name=Immigrant%20Flow%20In%20Last%20Month%20Sankey%20Diagram>

<https://www.whatstnew.com/2023/05/27/como-crear-facilmente-diagramas-de-sankey-online-y-gratis/>

<https://blog.visual-paradigm.com/es/how-to-create-a-sankey-diagram/>

<https://online.visual-paradigm.com/app/diagrams/#infoart:proj=0&type=SankeyDiagrams&gallery=/repository/6b126420-c269-45b0-9546-d99f148bea65.xml&name=Bilateral%20Trade%20Volume%20Sankey%20Diagram>

<https://www.visualitics.es/crear-un-grafico-sankey-con-tableau/>

<https://www.datasketch.co/es/blog/data-visualization-sankey-diagram/>

<https://www.thedataschool.co.uk/alfred-chan/how-to-build-a-sankey-chart-in-tableau/>

<https://www.youtube.com/watch?v=LclaJGRXRFU&t=586s>

[https://community.microstrategy.com/s/article/Sankey-diagram-eCharts?language=en\\_US](https://community.microstrategy.com/s/article/Sankey-diagram-eCharts?language=en_US)

<https://www.offidocs.com/es/blog/how-to-create-a-sankey-diagram-in-google-sheets/#:~:text=Antes%20de%20generar%20el%20diagrama%20de%20Sankey%2C%20deber%C3%A1,origen%20destino%20y%20flujo%20en%20los%20men%C3%BAAs%20desplegables.>

<https://extensiongallery.tableau.com/es-ES/products/932>

<https://physicsworld.com/a/more-than-one-quarter-of-physics-nobel-laureates-are-immigrants-reveal-updated-infographics/>

<https://www.youtube.com/watch?v=hPNJHXbUQQE>

<https://www.youtube.com/watch?v=LgyhBVAIDc0&t=40s>

<https://es.slideshare.net/MilenaPeralta/graficos-estadsticos-27067608>

[Sankey diagram – from Data to Viz \(data-to-viz.com\)](#)

<https://www.data-to-viz.com/graph/sankey.html>

[From data to Viz - Find the graphic you need](#)

[Data Viz Project | Collection of data visualizations to get inspired and find the right type](#)



<https://datavizproject.com/data-type/sankey-diagram/>

[Diagrama de Sankey - Wikipedia, la enciclopedia libre](#)

[https://es.wikipedia.org/wiki/Diagrama\\_de\\_Sankey](https://es.wikipedia.org/wiki/Diagrama_de_Sankey)

[Sankey Diagram - Learn about this chart and tools to create it \(datavizcatalogue.com\)](#)

[https://datavizcatalogue.com/methods/sankey\\_diagram.html](https://datavizcatalogue.com/methods/sankey_diagram.html)

## HEXAGONAL BINNING

<https://datavizproject.com/data-type/hexagonal-binning/>

<https://www.brc.ac.uk/theme/air-pollution>

<https://gisenteritorio.wordpress.com/2016/08/23/desarrollo-de-la-poblacion-europea/>

<https://blogs.sas.com/content/iml/2014/09/02/hexagonal-bin-plot.html#:~:text=A%20hexagonal%20bin%20plot%20is%20created%20by%20covering,which%20a%20scatter%20plot%20would%20suffer%20from%20overplotting.>

<https://www.meccanismocomplesso.org/en/hexagonal-binning-a-new-method-of-visualization-for-data-analysis/>

[https://www.bing.com/search?q=Hexagon+Binning&cvid=1ee53984a0744e2cb81e6af4605fb117&gs\\_lcrp=EgZjaHJvbWUyBggAEUYOTIGCAEQRg8MgYIAhBFGDzSAQg2OTQ2ajBqNKgCCLACAQ&FORM=ANAB01&PC=HCTS](https://www.bing.com/search?q=Hexagon+Binning&cvid=1ee53984a0744e2cb81e6af4605fb117&gs_lcrp=EgZjaHJvbWUyBggAEUYOTIGCAEQRg8MgYIAhBFGDzSAQg2OTQ2ajBqNKgCCLACAQ&FORM=ANAB01&PC=HCTS)

[Hexbin map – from Data to Viz \(data-to-viz.com\)](#)

<https://www.data-to-viz.com/graph/hexbinmap.html>

[Hexagonal Binning | Data Viz Project](#)

<https://datavizproject.com/data-type/hexagonal-binning/>

[https://es.wikipedia.org/wiki/Mapa\\_coropletico](https://es.wikipedia.org/wiki/Mapa_coropletico)

[https://ca.wikipedia.org/wiki/Mapa\\_coropletic](https://ca.wikipedia.org/wiki/Mapa_coropletic)

[https://ca.wikipedia.org/wiki/Codi\\_postal](https://ca.wikipedia.org/wiki/Codi_postal)

[https://es.wikipedia.org/wiki/Codigo\\_ZIP](https://es.wikipedia.org/wiki/Codigo_ZIP)

## ICON CHART

<https://www.domestika.org/es/blog/2492-que-es-un-icono-y-cuales-son-sus-diferencias-con-los-pictogramas>

<https://www.youtube.com/watch?v=yMguU1SPwr8&t=1s>

[https://www.bing.com/search?q=diferencia+entre+pictograma+e+icono&cvid=dd50cf296845493196064bec6834354e&gs\\_lcrp=EgZjaHJvbWUqBggAEUYOzIGCAAQRRg7MgYIARAAGEAyBggIEAAYQNIBCTExMzQxajBqNKgCCLACAQ&FORM=ANAB01&PC=HCTS](https://www.bing.com/search?q=diferencia+entre+pictograma+e+icono&cvid=dd50cf296845493196064bec6834354e&gs_lcrp=EgZjaHJvbWUqBggAEUYOzIGCAAQRRg7MgYIARAAGEAyBggIEAAYQNIBCTExMzQxajBqNKgCCLACAQ&FORM=ANAB01&PC=HCTS)

[https://www.bing.com/search?q=icon+chart&qs=UT&pq=icon+chart&sc=10-10&cvid=B37A69396F2843A383657DDE282664A8&FORM=QBRE&sp=1&ghc=1&lq=0&sm=csrm\\_ain](https://www.bing.com/search?q=icon+chart&qs=UT&pq=icon+chart&sc=10-10&cvid=B37A69396F2843A383657DDE282664A8&FORM=QBRE&sp=1&ghc=1&lq=0&sm=csrm_ain)

[Pictorial Unit Chart | Data Viz Project](#)

<https://datavizproject.com/data-type/pictorial-unit-chart/>



Altres fonts de dades obertes (recomanades a l'enunciat)

**1) Energia:**

<https://data.world/datasets/energy>  
<https://www.iea.org/data-and-statistics>  
<https://ourworldindata.org/energy>

Links to an external site.  
Links to an external site.  
Links to an external site.

**2) Opendata portals:**

CAT: [http://governobert.gencat.cat/ca/dades\\_obertes/](http://governobert.gencat.cat/ca/dades_obertes/)  
ES: <https://datos.gob.es/>  
UK: <https://data.gov.uk/>  
USA: <https://www.data.gov/>  
AU: <https://data.gov.au/dataLinks>

Links to an external site.  
Links to an external site.

**3) Altres:**

Data Sonification: <https://sonification.design/>  
RAW: <https://app.rawgraphs.io/> Acceso a los datasets:  
<https://github.com/rawgraphs/rawgraphs-app/tree/master/public/sample-datasets>  
PAX Peace agreements database: <https://www.peaceagreements.org/>  
Information is beautiful: <https://informationisbeautiful.net/data/>  
Kaggle: <https://www.kaggle.com/datasets>  
Directorio de datasets: <https://www.dataquest.io/blog/free-datasets-for-projects/>  
Google dataset search: <https://datasetsearch.research.google.com/>  
Gapminder: <https://www.gapminder.org/data/>  
Europa: <https://www.ecdc.europa.eu/en/covid-19/data> Links to an external site.  
Biblioteca de la UOC: on hi ha accés al recurs [Statista](#) i a l'espai de recursos de ciència de dades de la UOC <https://datascience.recursos.uoc.edu/?filter=.cdades>

**Moltes gràcies per la vostra atenció.**