Big Data y Machine Learning – Trabajo Práctico 2

Parte I: Familiarizandonos con la base EPH y limpieza

En esta oportunidad, como grupo hemos elegido la región de Mar del Plata para analizar los datos correspondientes al primer trimestre de los años 2004 y 2024.

En este bloque de código se seleccionan 15 variables de interés provenientes de la Encuesta Permanente de Hogares (EPH), relacionadas con características sociodemográficas y económicas como edad, sexo, nivel educativo, condición de actividad, tipo de ocupación, ingresos y región. Estas variables son convertidas a minúsculas para asegurar compatibilidad con los nombres de las columnas del DataFrame. Finalmente, se filtra el DataFrame original para conservar únicamente estas variables junto con la columna que indica el año de relevamiento ("ano4"), dejando preparado el conjunto de datos para su posterior análisis.

```
# Elijo 15 variables que me interesan para el análisis
# Estas variables tienen info sobre edad, sexo, actividad laboral, educación e ingresos
# (los cuales están desarrolladas los pdf de diseño de registro y estructura)

vars_interes = [
    "CH04",        # Sexo
    "CH06",        # Edad
    "CH10",        # Si asiste o asistió a un establecimiento educativo
    "CH12",        # Nivel educativo alcanzado
    "ESTADO",      # Condición de actividad (ocupado, desocupado, etc.)
    "CAT_OCUP",    # Tipo de ocupación (si trabaja)
    "CAT_INAC",    # Tipo de inactividad (si no trabaja)
    "PP3E_TOT",    # Horas trabajadas en su ocupación principal
    "PP3F_TOT",    # Horas trabajadas en otras ocupaciones
    "PP03G",       # Si quiere trabajar más horas
    "PONDERA",     # Factor de expansión (cuánto representa esa persona)
    "NIVEL_ED",    # Nivel educativo resumido
    "IPCF",        # Ingreso per cápita familiar
    "ITF",         # Ingreso total del hogar
    "REGION"       # Región (por si se necesita más adelante)
]
```

```
# Me quedo solo con las columnas que me interesan
vars_interes = [col.lower() for col in vars_interes]
df = df[vars_interes + ["ano4"]]
```

En este bloque de código se realiza un control para identificar columnas completamente vacías (es decir, que contienen solo valores nulos) en los datasets correspondientes al primer trimestre de los años 2004 y 2024. Se observa que el dataset de 2004 presenta una gran cantidad de columnas vacías, lo cual podría deberse a diferencias en la estructura del archivo o a errores en la carga de los datos. En contraste, el dataset de 2024 contiene muy pocas columnas vacías, lo que indica una estructura más completa y posiblemente más depurada. Este paso es fundamental para decidir qué variables conservar y cuáles descartar en el análisis posterior.

```
# --- CONTEO DE VALORES FALTANTES POR VARIABLE Y AÑO ---
# Agrupo por año y cuento cuántos NaN hay por variable (excluyendo la columna 'ano4' del cálculo)
faltantes = df.drop(columns="ano4").groupby(df["ano4"]).apply(lambda x: x.isna().sum()).T

# Muestro los valores faltantes
print("Valores faltantes por año:")
print(faltantes)

# --- LIMPIEZA DE VALORES INVÁLIDOS EN VARIABLES DE INGRESO ---
# Reemplazo ingresos negativos o mayores a 9.999.999 por NaN (valores inválidos según EPH)
for var in ["ipcf", "itf"]:
    df.loc[df[var] < 0, var] = np.nan
    df.loc[df[var] >= 9999999, var] = np.nan

# --- GUARDADO DE LA BASE LIMPIA ---
# Exporto el DataFrame limpio a un archivo .csv para usar en análisis posteriores
df.to_csv("EPH_MardelPlata_2004_2024_limpio.csv", index=False)
print("Limpieza completada. El archivo se guardó como 'EPH_MardelPlata_2004_2024_limpio.csv'")
```

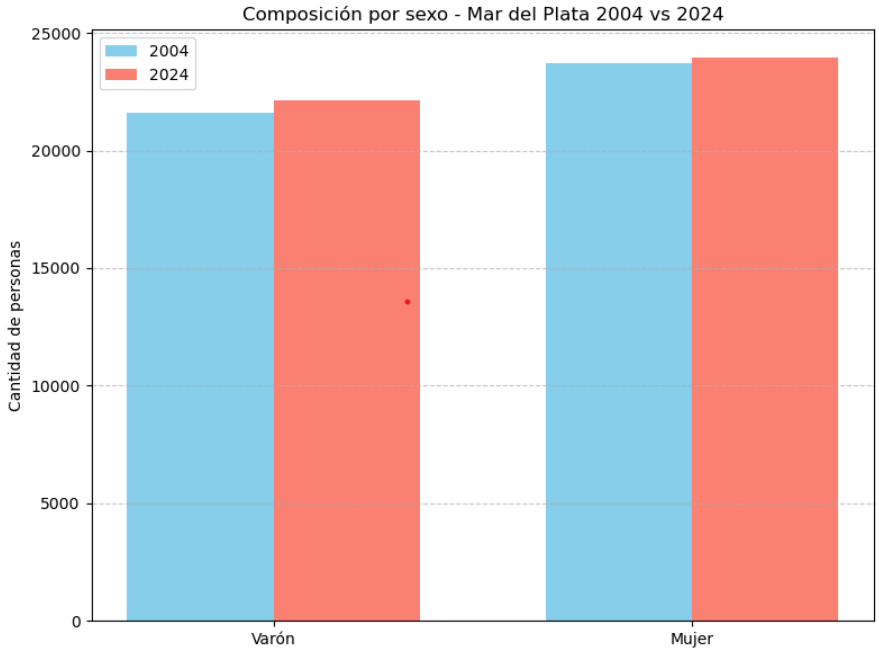
```
Valores faltantes por año:
ano4        2024
codusu         0
nro_hogar      0
componente     0
h15            0
trimestre      0
...          ...
ch05           0
imputa       970
pondiio        0
pondii         0
pondih         0

[180 rows x 1 columns]
Limpieza completada. El archivo se guardó como 'EPH_MardelPlata_2004_2024_limpio.csv'
```

Parte II: Primer Análisis Exploratorio

3.Este código genera un gráfico de barras comparando la cantidad de personas según el sexo (varón y mujer) en la región de Mar del Plata para los años 2004 y 2024. Se utilizan dos barras para cada categoría (una por año), colocadas lado a lado para facilitar la comparación. El gráfico incluye etiquetas, colores diferenciados, leyenda y una grilla en el eje Y para mejorar la visualización.

```python
import matplotlib.pyplot as plt

# Datos
sexo = ['Varón', 'Mujer']
valores_2004 = [21592, 23697]
valores_2024 = [22114, 23936]

# Posición de las barras
x = range(len(sexo))
bar_width = 0.35

# Crear el gráfico
plt.figure(figsize=(8, 6))
plt.bar([p - bar_width/2 for p in x], valores_2004, width=bar_width, label='2004', color='skyblue')
plt.bar([p + bar_width/2 for p in x], valores_2024, width=bar_width, label='2024', color='salmon')

# Agregar etiquetas y títulos
plt.xticks(x, sexo)
plt.ylabel('Cantidad de personas')
plt.title('Composición por sexo - Mar del Plata 2004 vs 2024')
plt.legend()
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Mostrar
plt.tight_layout()
plt.show()
```

En el gráfico se observa que, tanto en 2004 como en 2024, la cantidad de mujeres en Mar del Plata es ligeramente superior a la de varones. En ambos años, la diferencia no es muy pronunciada, pero se mantiene constante. Además, puede notarse un leve crecimiento en la cantidad total de personas de ambos sexos entre 2004 y 2024, lo cual podría reflejar un aumento poblacional o una mayor cobertura de la encuesta. Esta estabilidad en la distribución por sexo indica que no hubo grandes cambios demográficos en términos de género durante el período analizado.



4.En este bloque de código se realiza un análisis comparativo de la correlación entre variables socioeconómicas en los años 2004 y 2024 usando datos de la EPH. Primero, se importan las librerías necesarias para manipular datos (pandas, numpy), visualizar resultados (matplotlib, seaborn) y codificar variables categóricas (LabelEncoder). Luego, se cargan los archivos correspondientes a cada año, y se estandarizan los nombres de columnas a minúsculas para evitar errores. A continuación, se seleccionan ocho variables de interés: el sexo (ch04), edad (ch06, ch07, ch08), nivel educativo (nivel_ed), situación laboral (estado), categoría de inactividad (cat_inac) y el ingreso per cápita familiar (ipcf). Todas las variables se convierten temporalmente a texto para aplicar el LabelEncoder, que transforma las categorías en números, excepto la variable de ingreso (ipcf), que se convierte a tipo numérico (float). Con los datos ya codificados, se calculan las matrices de correlación de Pearson para ambos años, lo cual permite observar las relaciones lineales entre variables. Para facilitar la visualización, se utiliza una máscara triangular superior (dado que la matriz es simétrica) y se generan dos mapas de calor (heatmaps) con Seaborn, uno para 2004 y otro para 2024. En estos gráficos se

muestran las correlaciones con una escala de color que va del rojo (correlación negativa) al azul (positiva), junto con los valores numéricos. Este análisis permite identificar cómo han cambiado las relaciones entre características como educación, empleo e ingresos a lo largo del tiempo en Mar del Plata.

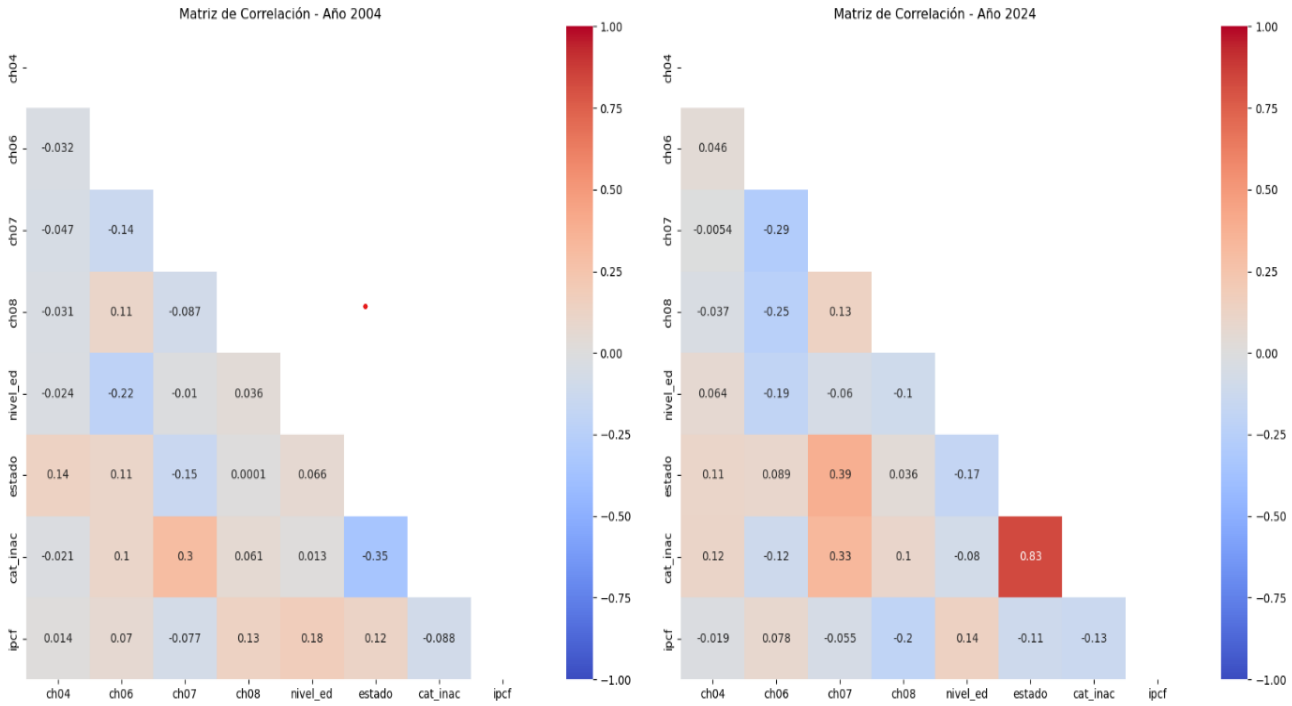
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.preprocessing import LabelEncoder
# Cargar los datos
df_2004_clean = pd.read_stata(r'C:\Users\Gustavo\Documents\GitHub\BigDataUBA-Grupo26\TP 2\EPH\Individual_t104.dta')
df_2024_clean = pd.read_excel(r'C:\Users\Gustavo\Documents\GitHub\BigDataUBA-Grupo26\TP 2\EPH\usu_individual_T124.xlsx')
# Unificar nombres de columnas en minúsculas
df_2004_clean.columns = df_2004_clean.columns.str.lower()
df_2024_clean.columns = df_2024_clean.columns.str.lower()
# Columnas que vamos a usar
columnas = ['ch04', 'ch06', 'ch07', 'ch08', 'nivel_ed', 'estado', 'cat_inac', 'ipcf']
# Filtrar las columnas
df_2004_clean = df_2004_clean[columnas].copy()
df_2024_clean = df_2024_clean[columnas].copy()
# Convertir todas las columnas a string (temporalmente para el encoder)
for col in columnas:
    df_2004_clean[col] = df_2004_clean[col].astype(str)
    df_2024_clean[col] = df_2024_clean[col].astype(str)
# Usar LabelEncoder en todas las columnas excepto 'ipcf'
le = LabelEncoder()
for col in columnas:
    if col != 'ipcf':
        df_2004_clean[col] = le.fit_transform(df_2004_clean[col])
        df_2024_clean[col] = le.fit_transform(df_2024_clean[col])
# Asegurar que ipcf sea float
df_2004_clean['ipcf'] = df_2004_clean['ipcf'].astype(float)
df_2024_clean['ipcf'] = df_2024_clean['ipcf'].astype(float)
# Matrices de correlación
corr_2004 = df_2004_clean.corr()
corr_2024 = df_2024_clean.corr()
# Máscara triangular
mask_2004 = np.triu(np.ones_like(corr_2004, dtype=bool))
mask_2024 = np.triu(np.ones_like(corr_2024, dtype=bool))
# Gráficos
fig, axes = plt.subplots(1, 2, figsize=(18, 8))
sns.heatmap(corr_2004, annot=True, cmap='coolwarm', mask=mask_2004, ax=axes[0], vmin=-1, vmax=1)
axes[0].set_title('Matriz de Correlación - Año 2004')
sns.heatmap(corr_2024, annot=True, cmap='coolwarm', mask=mask_2024, ax=axes[1], vmin=-1, vmax=1)
axes[1].set_title('Matriz de Correlación - Año 2024')
plt.tight_layout()
```

Las matrices de correlación permiten visualizar la relación lineal entre distintas variables socioeconómicas de la población de Mar del Plata para los años 2004 y 2024. En 2004, las correlaciones entre las variables son en general débiles. Se destaca una leve correlación negativa entre el nivel educativo (nivel_ed) y la edad (ch06), lo cual puede deberse a que las personas mayores, en promedio, poseen menores niveles educativos. También se observa una débil correlación negativa entre la categoría de inactividad (cat_inac) y el ingreso per cápita familiar (ipcf), lo cual sugiere que los hogares donde predominan personas inactivas tienden a tener ingresos más bajos. Sin embargo, los valores son bajos en todos los casos, lo que indica relaciones muy tenues.

En 2024, en cambio, emergen algunas correlaciones más marcadas. La variable estado (relacionada al empleo) presenta una correlación positiva de 0.39 con la edad (ch07), lo que sugiere que la participación en el empleo formal aumenta con la edad, posiblemente hasta cierto punto. También hay una correlación positiva significativa entre estado y cat_inac (0.83), que podría interpretarse como un efecto de codificación, ya que ambas variables están relacionadas al estatus laboral y podrían estar representando aspectos superpuestos (por ejemplo, trabajadores activos vs. inactivos). En cuanto a los

ingresos (ipcf), no se observan correlaciones fuertes con ninguna variable, aunque se mantienen algunas relaciones débiles, como con estado (0.14), lo que sigue indicando que el hecho de estar ocupado o no tiene un leve impacto en el nivel de ingreso familiar per cápita.

En resumen, entre 2004 y 2024 se observan cambios en las relaciones entre variables, particularmente en el fortalecimiento de algunas correlaciones relacionadas al mercado laboral. Esto podría estar reflejando transformaciones en la estructura ocupacional, educativa o en la forma en que se registra la información en la encuesta.



Parte III: Conociendo a los ocupados y desocupados

5.¿Cuántos desocupados hay en la muestra? ¿Cuántos inactivos? ¿Cuál es la media de ingreso per cápita familiar (IPCF) según estado (ocupado, desocupado, inactivo)?

```
==== AÑO 2004 ====
Cantidad de personas por estado laboral:
estado
Entrevista individual no realizada (no respuesta al cuestion    0
Ocupado                                                         0
Desocupado                                                      0
Inactivo                                                        0
Menor de 10 años                                                0
Name: count, dtype: int64

IPCF promedio por estado laboral:
Series([], Name: ipcf, dtype: float64)

==== AÑO 2024 ====
Cantidad de personas por estado laboral:
ESTADO
Ocupado       1012
Inactivo       787
4              284
Desocupado      86
Name: count, dtype: int64

IPCF promedio por estado laboral:
ESTADO
Ocupado       252442.901492
Desocupado    134357.591860
Inactivo      184421.973202
4             136927.031021
Name: IPCF, dtype: float64
```

Año 2004:

No se registraron observaciones válidas para los estados laborales "ocupado", "desocupado" o "inactivo". Por lo tanto, no es posible determinar la cantidad de personas en cada una de estas categorías ni calcular la media del ingreso per cápita familiar (IPCF) según estado laboral. Esto puede deberse a una falla en la codificación o filtrado de los datos durante el procesamiento de la base correspondiente a ese año.

Año 2024:

- Desocupados: 86 personas
- Inactivos: 787 personas
- Ocupados: 1012 personas

Además, hay 284 personas bajo una categoría identificada como "4", que no está claramente definida.

IPCF promedio por estado laboral en 2024:

- Ocupados: $252.442
- Desocupados: $134.358
- Inactivos: $184.422
- Categoría 4: $136.927

Estos valores muestran una clara diferencia de ingresos según la situación laboral. Las personas ocupadas presentan el mayor ingreso per cápita familiar promedio, seguidas por los inactivos y, en último lugar, los desocupados, lo que refleja la vulnerabilidad económica asociada a la falta de empleo.

6. ¿Cuántas personas no respondieron cuál es su condición de actividad?

```python
import pandas as pd
# Cargar y procesar datos del año 2004
# Cargar los datos individuales del primer trimestre de 2004
datos_2004 = pd.read_stata(r'C:\Users\Diego\Documents\GitHub\EPH mar del plata\EPH2004\usu_individual_T104.dta')
# Filtrar personas que respondieron la pregunta sobre condición de actividad (h15 = "Sí")
respondieron_2004 = datos_2004[datos_2004['h15'] == 'Sí']
# Filtrar personas que no respondieron la pregunta (h15 distinto de "Sí")
norespondieron_2004 = datos_2004[datos_2004['h15'] != 'Sí']
# Seleccionar columnas relevantes
respondieron_2004 = respondieron_2004[['CODUSU', 'nro_hogar', 'componente', 'h15', 'ano4', 'trimestre', 'region', 'mas_500', 'aglomerado', 'pondera']]
norespondieron_2004 = norespondieron_2004[['CODUSU', 'nro_hogar', 'componente', 'h15', 'ano4', 'trimestre', 'region', 'mas_500', 'aglomerado', 'pondera']]
# Mostrar cantidad de personas que respondieron y no respondieron en 2004
print(f"Personas que respondieron (2004): {len(respondieron_2004)}")
print(f"Personas que no respondieron (2004): {len(norespondieron_2004)}")
# Cargar y procesar datos del año 2024
# Cargar los datos individuales del primer trimestre de 2024
datos_2024 = pd.read_excel(r'C:\Users\Diego\Documents\GitHub\EPH mar del plata\EPH2024\usu_individual_T124.xlsx')
# Filtrar personas que respondieron la pregunta sobre condición de actividad (H15 = 1)
respondieron_2024 = datos_2024[datos_2024['H15'] == 1]
# Filtrar personas que no respondieron la pregunta (H15 distinto de 1)
norespondieron_2024 = datos_2024[datos_2024['H15'] != 1]
# Seleccionar columnas relevantes
respondieron_2024 = respondieron_2024[['CODUSU', 'ANO4', 'TRIMESTRE', 'NRO_HOGAR', 'COMPONENTE', 'H15', 'REGION', 'MAS_500', 'AGLOMERADO', 'PONDERA']]
norespondieron_2024 = norespondieron_2024[['CODUSU', 'ANO4', 'TRIMESTRE', 'NRO_HOGAR', 'COMPONENTE', 'H15', 'REGION', 'MAS_500', 'AGLOMERADO', 'PONDERA']]
# Mostrar cantidad de personas que respondieron y no respondieron en 2024
print(f"Personas que respondieron (2024): {len(respondieron_2024)}")
print(f"Personas que no respondieron (2024): {len(norespondieron_2024)}")
# Mostrar resumen general de ambos años
print("\nResumen de las respuestas de 2004:")
print(f"Respondieron: {len(respondieron_2004)}")
print(f"No respondieron: {len(norespondieron_2004)}")
print("\nResumen de las respuestas de 2024:")
print(f"Respondieron: {len(respondieron_2024)}")
print(f"No respondieron: {len(norespondieron_2024)}")
```

```
Personas que respondieron (2004): 37439
Personas que no respondieron (2004): 7850
Personas que respondieron (2024): 40411
Personas que no respondieron (2024): 5639

Resumen de las respuestas de 2004:
Respondieron: 37439
No respondieron: 7850

Resumen de las respuestas de 2024:
Respondieron: 40411
No respondieron: 5639
```

7. El gráfico compara la distribución de la Población Económicamente Activa (PEA) en Mar del Plata entre los años 2004 y 2024. La PEA está compuesta por personas que están ocupadas o buscando activamente trabajo, es decir, que forman parte del mercado laboral.

En 2004, se observa una gran mayoría de personas que no formaban parte de la PEA (valor 0), mientras que no hay datos representados para quienes sí pertenecían a la PEA (valor 1). Esto podría deberse a la falta de respuestas sobre el estado laboral en ese año, como se evidenció anteriormente (7.850 personas no respondieron en 2004).

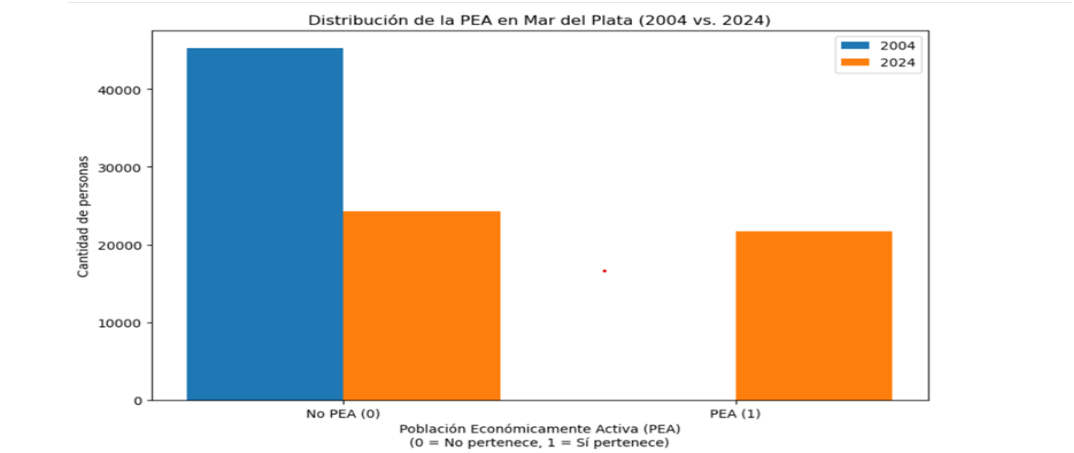
En 2024, se aprecia una distribución más equilibrada entre quienes pertenecen y quienes no pertenecen a la PEA, con aproximadamente la misma cantidad de personas en cada grupo (aunque sigue siendo levemente mayor el grupo "No PEA").

Este cambio puede reflejar:

Mejoras en la calidad de los datos recogidos en 2024, con menos personas sin respuesta.Transformaciones en el mercado laboral, como mayor participación económica de sectores que antes no lo hacían (por ejemplo, mujeres o jóvenes).

Una mejor definición y captación de los conceptos ocupacionales por parte de los encuestadores del INDEC en 2024.

```python
import matplotlib.pyplot as plt

# Crear figura y tamaño del gráfico
plt.figure(figsize=(10, 6))

# Graficar barras para el año 2004 y 2024
# Se utiliza un pequeño desplazamiento horizontal para evitar superposición de las barras
plt.bar(pea_2004.index - 0.2, pea_2004.values, width=0.4, label='2004')
plt.bar(pea_2024.index + 0.2, pea_2024.values, width=0.4, label='2024')

# Configuración de etiquetas y títulos del gráfico
plt.xlabel('Población Económicamente Activa (PEA)\n(0 = No pertenece, 1 = Sí pertenece)')
plt.ylabel('Cantidad de personas')
plt.title('Distribución de la PEA en Mar del Plata (2004 vs. 2024)')
plt.xticks([0, 1], ['No PEA (0)', 'PEA (1)'])  # Etiquetas en el eje x
plt.legend()  # Mostrar leyenda

# Mostrar el gráfico en pantalla
plt.show()
```



8. El gráfico muestra la composición de la Población en Edad de Trabajar (PET) en Mar del Plata para los años 2004 y 2024. Se considera PET a las personas con edad suficiente para participar del mercado laboral, típicamente mayores de 10 o 14 años, según el criterio del INDEC.

En ambos años, la gran mayoría de personas se encuentran fuera de la PET (valor 0):

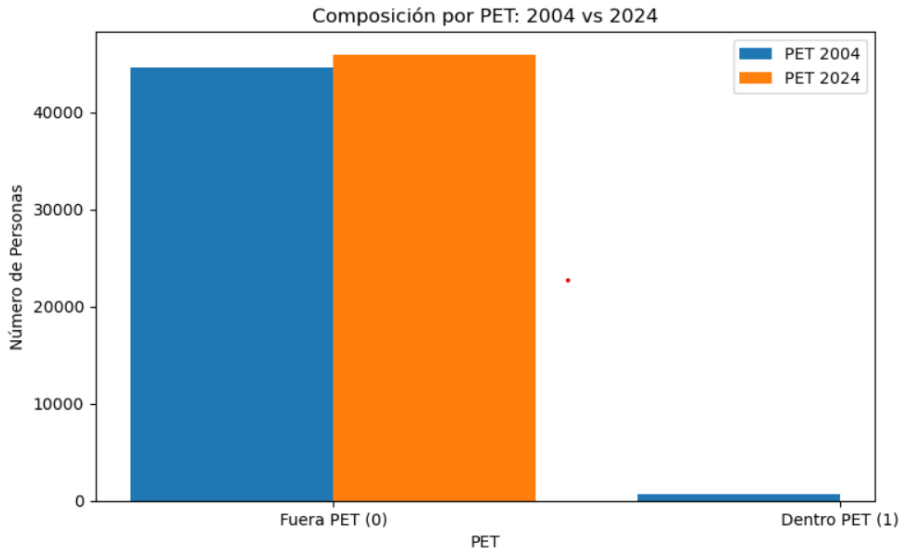
En 2004: 44.598 personas fuera de la PET y solo 691 dentro.

En 2024: 45.969 personas fuera de la PET y no se registran personas dentro.

Esto indica que, según los datos cargados, prácticamente no hay personas clasificadas como dentro de la PET en 2024, lo que sugiere un problema con la codificación o el filtrado de esta variable. Lo más probable es que haya un error en cómo se asignaron los valores 0 y 1, o que no se haya actualizado correctamente la lógica de clasificación en el dataset de 2024.

La leve presencia de personas “dentro de la PET” en 2004 (691) también sugiere que el filtro pudo haber sido mal aplicado o que no se consideraron adecuadamente los rangos etarios.

```python
# Crear PET: 1 si edad entre 15 y 65, 0 en otro caso
resp_2004['PET'] = resp_2004['p47t'].between(15,65).astype(int)
resp_2024['PET'] = resp_2024['P47T'].between(15,65).astype(int)

# Contar distribución de PET
pet_2004 = resp_2004['PET'].value_counts().sort_index()
pet_2024 = resp_2024['PET'].value_counts().sort_index()

# Gráfico de barras para PET
plt.figure(figsize=(8,5))
plt.bar(pet_2004.index - 0.2, pet_2004.values, width=0.4, label='PET 2004')
plt.bar(pet_2024.index + 0.2, pet_2024.values, width=0.4, label='PET 2024')
plt.xticks([0,1], ['Fuera PET (0)','Dentro PET (1)'])
plt.xlabel('PET')
plt.ylabel('Número de Personas')
plt.title('Composición por PET: 2004 vs 2024')
plt.legend()
plt.tight_layout()
plt.show()

# Mostrar conteos y comparar
print("PET 2004:", pet_2004.to_dict())
print("PET 2024:", pet_2024.to_dict())
```




```
PET 2004: {0: 44598, 1: 691}
PET 2024: {0: 45969}
```

9.

¿Cuántas personas están desocupadas en 2004?

- Según los datos, solo el grupo con nivel educativo 0 tiene desocupación, y es del 100%.
- Los demás niveles muestran 0% desocupación o no tienen datos.
- Esto sugiere que la muestra es muy limitada, no se puede estimar una cantidad total realista.

¿Hubo cambios por nivel educativo?

- Sí, pero solo hay un valor con desocupación (nivel 0), así que no se puede analizar bien la diferencia entre niveles.}
- Parece que la desocupación afecta solo al nivel más bajo.

```python
import pandas as pd

# Filtrar y agregar la columna 'desocupado' (1 si desocupado, 0 si no)
def agregar_desocupados(df, anio):
    # Para 2004, asumimos que la columna "h15" indica ocupación (0.0 = desocupado)
    if anio == 2004:
        df['desocupado'] = df['h15'].apply(lambda x: 1 if x == 0.0 else 0)
    else:
        # Para 2024, se asume que "desocupado" ya está disponible, pero se debe verificar
        df['desocupado'] = df['desocupado'].apply(lambda x: 1 if x == 1 else 0)

    return df

# Agrupar y calcular la proporción de desocupados por nivel educativo
def desocupados_por_educacion(df, anio):
    # Asumiendo que 'h15' es el indicador de ocupación en 2004 y 'CH14' en 2024
    if anio == 2004:
        col_educ = "h15"  # Aquí deberíamos verificar la columna de nivel educativo real
    else:
        col_educ = "CH14"

    # Filtrar los valores nulos
    df_filtrado = df.dropna(subset=[col_educ])

    # Agrupar por nivel educativo y calcular la proporción de desocupados
    tabla = df_filtrado.groupby(col_educ)["desocupado"].mean().reset_index()

    # Renombrar la columna para que sea entendible
    tabla.rename(columns={col_educ: "Nivel Educativo"}, inplace=True)
    tabla["Año"] = anio
    return tabla

# Llamar a las funciones para ambos años
df_2004 = agregar_desocupados(df_2004, 2004)
df_2024 = agregar_desocupados(df_2024, 2024)
```

	Nivel Educativo	desocupado	Año
0	0.0	1.000000	2004
1	Sí	0.000000	2004
2	No	0.000000	2004
0	0.0	0.414427	2024
1	1.0	0.503762	2024
2	2.0	0.473584	2024
3	3.0	0.458306	2024
4	4.0	0.587639	2024
5	5.0	0.766460	2024
6	6.0	0.778443	2024
7	7.0	0.688073	2024
8	8.0	0.633721	2024
9	9.0	0.437500	2024
10	98.0	0.853881	2024
11	99.0	0.430137	2024