Level 1

- 1.
    - Import the sprint10.xlsx file as a DataFrame. Make sure the file is imported correctly, with the corresponding column names, without manipulating the original file.
    - Sort the DataFrame by country of origin. In case of a tie, sort by city name.
    - Show the first 10 rows.

Additionally, take a *printout* to verify that the DNI only has unique values.

import pandas as pd

df = pd.read_excel(r"C:\Users\sarav\Downloads\sprint10.xlsx",header =3)

df

	Unnamed: 0	Nom	Cognoms	DNI	País d'origen	Ciutat	Dia de Naixement	Mes de Naixement	Any de Naixement	Gènere	Salari mensual	Fills	No Fills	Grup Professional
0	0	Inês	Ferreira Silva	16928694K	Portugal	Lisboa	25	2	1953	D	1.144 €	NaN	1.0	Grup B
1	1	Clara	Sánchez Martínez	27724652S	Espanya	Barcelona	18	3	1996	D	1.253 €	1.0	NaN	Grup A
2	2	Fatima	Fassi	38141675A	Marroc	Rabat	6	11	2005	A	1.441 €	1.0	NaN	Grup A
3	3	Khadija	Bennani Bennani	59157262R	Marroc	Rabat	20	1	1995	D	1.944 €	NaN	1.0	Grup B
4	4	Toni	Sánchez García	69630528M	Espanya	Barcelona	9	8	1999	H	1.043 €	NaN	1.0	Grup A
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	995	Marta	Ferrer Ferrer	25161375F	Espanya	Sevilla	1	6	1951	D	1.216 €	NaN	1.0	Grup B
996	996	Joan	García	52145541P	Espanya	Sevilla	11	4	1959	H	971 €	NaN	1.0	Grup A
997	997	Laia	Ferrer Martínez	69760120X	Espanya	Barcelona	11	11	1980	D	682 €	NaN	1.0	Grup A
998	998	Jordi	García	82947791W	Espanya	Barcelona	23	5	1984	H	1.699 €	1.0	NaN	Grup C
999	999	Clara	Sánchez López	89253307W	Espanya	Bilbao	1	8	1952	D	1.217 €	NaN	1.0	Grup A

df2 = df.sort_values(by = ["País d'origen","Ciutat"])

df2

	Unnamed: 0	Nom	Cognoms	DNI	País d'origen	Ciutat	Dia de Naixement	Mes de Naixement	Any de Naixement	Gènere	Salari mensual	Fills	No Fills	Grup Professional
21	21	Mia	Schneider Fischer	28973553Z	Alemanya	Berlín	22	10	1976	A	951 €	NaN	1.0	Grup A
154	154	Laura	Schneider Fischer	37399141L	Alemanya	Berlín	2	2	1958	D	1.769 €	1.0	NaN	Grup B
224	224	Lea	Schneider Schneider	37368317L	Alemanya	Berlín	23	10	2005	D	2.013 €	NaN	1.0	Grup B
278	278	Mia	Fischer	21390098Z	Alemanya	Berlín	11	8	1950	D	1.557 €	1.0	NaN	Grup B
602	602	Jonas	Schneider	44060014R	Alemanya	Berlín	22	11	1985	H	2.754 €	1.0	NaN	Grup D
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
547	547	Emily	Taylor Jones	89577876S	Regne Unit	Manchester	28	3	1958	D	2.033 €	NaN	1.0	Grup B
728	728	George	Brown Jones	57441590Y	Regne Unit	Manchester	27	12	1979	H	1.130 €	1.0	NaN	Grup A
751	751	Olivia	Brown Brown	58204038A	Regne Unit	Manchester	28	8	1952	A	1.023 €	NaN	1.0	Grup A
854	854	Isla	Jones Brown	28367234K	Regne Unit	Manchester	28	3	1999	D	1.197 €	NaN	1.0	Grup A

df2.head(10)

[5]:

	Unnamed: 0	Nom	Cognoms	DNI	País d'origen	Ciutat	Dia de Naixement	Mes de Naixement	Any de Naixement	Gènere	Salari mensual	Fills	No Fills	Grup Professional
21	21	Mia	Schneider Fischer	28973553Z	Alemanya	Berlín	22	10	1976	A	951 €	NaN	1.0	Grup A
154	154	Laura	Schneider Fischer	37399141L	Alemanya	Berlín	2	2	1958	D	1.769 €	1.0	NaN	Grup B
224	224	Lea	Schneider Schneider	37368317L	Alemanya	Berlín	23	10	2005	D	2.013 €	NaN	1.0	Grup B
278	278	Mia	Fischer	21390098Z	Alemanya	Berlín	11	8	1950	D	1.557 €	1.0	NaN	Grup B
602	602	Jonas	Schneider	44060014R	Alemanya	Berlín	22	11	1985	H	2.754 €	1.0	NaN	Grup D
871	871	Lea	Fischer	14773153R	Alemanya	Berlín	9	9	1986	D	1.370 €	1.0	NaN	Grup A
281	281	Lea	Müller	23266650S	Alemanya	Hamburg	14	4	2003	D	1.314 €	NaN	1.0	Grup A
435	435	Anna	Müller	83274277X	Alemanya	Hamburg	1	1	1987	D	2.464 €	NaN	1.0	Grup C
444	444	Laura	Schmidt Müller	60161784X	Alemanya	Hamburg	15	6	1987	NC	2.035 €	1.0	NaN	Grup C
487	487	Lukas	Müller Fischer	60982309S	Alemanya	Hamburg	28	3	1971	H	2.042 €	NaN	1.0	Grup B

2.

Create a column that is the full name.

Create a column if the person was born in Spain or not.

Put the ID as the index of the DataFrame (row names).

Replace the name of the columns Day of Birth, Month of Birth and Year of Birth with Day, Month and Year.

Replace H with Male, D with Female, A with Other and NC with a missing data (nan/null/na).

Show all the changes you've made in a single table.

```
df2['nomcomplet'] = df2['Nom'] + df2['Cognoms']

df2[' nascuda_Espanya'] = df2["País d'origen"].apply(lambda x :'yes' if x=='Espanya' else 'No')

df2.set_index('DNI', inplace = True)

df2.rename(columns = {'Dia de Naixement' : ' Dia','Mes de Naixement' : 'Mes','Any de Naixement' : 'Any'}, inplace = True)


Gender = {'H': 'Home', 'D': 'Dona', 'A': 'Altres', 'NC': 'Nan'}
df2['Gènere'] = df2['Gènere'].replace(Gender)
df2
```

[11]:

DNI	Unnamed: 0	Nom	Cognoms	País d'origen	Ciutat	Dia	Mes	Any	Gènere	Salari mensual	Fills	No Fills	Grup Professional	nomcomplet	nascuda_Espanya
28973553Z	21	Mia	Schneider Fischer	Alemanya	Berlín	22	10	1976	Altres	951 €	NaN	1.0	Grup A	MiaSchneider Fischer	No
37399141L	154	Laura	Schneider Fischer	Alemanya	Berlín	2	2	1958	Dona	1.769 €	1.0	NaN	Grup B	LauraSchneider Fischer	No
37368317L	224	Lea	Schneider Schneider	Alemanya	Berlín	23	10	2005	Dona	2.013 €	NaN	1.0	Grup B	LeaSchneider Schneider	No
21390098Z	278	Mia	Fischer	Alemanya	Berlín	11	8	1950	Dona	1.557 €	1.0	NaN	Grup B	MiaFischer	No
44060014R	602	Jonas	Schneider	Alemanya	Berlín	22	11	1985	Home	2.754 €	1.0	NaN	Grup D	JonasSchneider	No
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
89577876S	547	Emily	Taylor Jones	Regne Unit	Manchester	28	3	1958	Dona	2.033 €	NaN	1.0	Grup B	EmilyTaylor Jones	No
57441590Y	728	George	Brown Jones	Regne Unit	Manchester	27	12	1979	Home	1.130 €	1.0	NaN	Grup A	GeorgeBrown Jones	No
58204038A	751	Olivia	Brown Brown	Regne Unit	Manchester	28	8	1952	Altres	1.023 €	NaN	1.0	Grup A	OliviaBrown Brown	No
28367234K	854	Isla	Jones Brown	Regne Unit	Manchester	28	3	1999	Dona	1.197 €	NaN	1.0	Grup A	IslaJones Brown	No

Merge the Children and Not Children columns into a single column, using the .apply() method and defining a function that solves the problem.

The new column should be called "Children" and take the values "Yes" or "No".

```
def children(row):
    if row['Fills'] == 1.0:
        return 'Si'
    elif row['No Fills']  == 1.0:
        return 'No'
    else:
        return None



df2['Fills'] = df2.apply(children,axis=1)
df2
```

DNI	Unnamed: 0	Nom	Cognoms	País d'origen	Ciutat	Dia	Mes	Any	Gènere	Salari mensual	Fills	No Fills	Grup Professional	nomcomplet	nascuda_Espanya
28973553Z	21	Mia	Schneider Fischer	Alemanya	Berlín	22	10	1976	Altres	951 €	No	1.0	Grup A	MiaSchneider Fischer	No
37399141L	154	Laura	Schneider Fischer	Alemanya	Berlín	2	2	1958	Dona	1.769 €	Si	NaN	Grup B	LauraSchneider Fischer	No
37368317L	224	Lea	Schneider Schneider	Alemanya	Berlín	23	10	2005	Dona	2.013 €	No	1.0	Grup B	LeaSchneider Schneider	No
21390098Z	278	Mia	Fischer	Alemanya	Berlín	11	8	1950	Dona	1.557 €	Si	NaN	Grup B	MiaFischer	No
44060014R	602	Jonas	Schneider	Alemanya	Berlín	22	11	1985	Home	2.754 €	Si	NaN	Grup D	JonasSchneider	No
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
89577876S	547	Emily	Taylor Jones	Regne Unit	Manchester	28	3	1958	Dona	2.033 €	No	1.0	Grup B	EmilyTaylor Jones	No
57441590Y	728	George	Brown Jones	Regne Unit	Manchester	27	12	1979	Home	1.130 €	Si	NaN	Grup A	GeorgeBrown Jones	No
58204038A	751	Olivia	Brown Brown	Regne Unit	Manchester	28	8	1952	Altres	1.023 €	No	1.0	Grup A	OliviaBrown Brown	No
28367234K	854	Isla	Jones	Regne	Manchester	28	3	1999	Dona	1.197 €	No	1.0	Grup A	IslaJones	No

df2.drop(columns = ['No Fills'],inplace = True)

df2

DNI	Unnamed: 0	Nom	Cognoms	País d'origen	Ciutat	Dia	Mes	Any	Gènere	Salari mensual	Fills	Grup Professional	nomcomplet	nascuda_Espanya
28973553Z	21	Mia	Schneider Fischer	Alemanya	Berlín	22	10	1976	Altres	951 €	No	Grup A	MiaSchneider Fischer	No
37399141L	154	Laura	Schneider Fischer	Alemanya	Berlín	2	2	1958	Dona	1.769 €	Si	Grup B	LauraSchneider Fischer	No
37368317L	224	Lea	Schneider Schneider	Alemanya	Berlín	23	10	2005	Dona	2.013 €	No	Grup B	LeaSchneider Schneider	No
21390098Z	278	Mia	Fischer	Alemanya	Berlín	11	8	1950	Dona	1.557 €	Si	Grup B	MiaFischer	No
44060014R	602	Jonas	Schneider	Alemanya	Berlín	22	11	1985	Home	2.754 €	Si	Grup D	JonasSchneider	No
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
89577876S	547	Emily	Taylor Jones	Regne Unit	Manchester	28	3	1958	Dona	2.033 €	No	Grup B	EmilyTaylor Jones	No
57441590Y	728	George	Brown Jones	Regne Unit	Manchester	27	12	1979	Home	1.130 €	Si	Grup A	GeorgeBrown Jones	No
58204038A	751	Olivia	Brown Brown	Regne Unit	Manchester	28	8	1952	Altres	1.023 €	No	Grup A	OliviaBrown Brown	No
28367234K	854	Isla	Jones Brown	Regne Unit	Manchester	28	3	1999	Dona	1.197 €	No	Grup A	IslaJones Brown	No

Create a summary table that allows you to see the average, median, minimum and maximum salary by Gender.

Sort the table based on average salary.

```
df2['Salari mensual'] = (
    df2['Salari mensual']
    .str.replace('€','',regex = False)
    .str.replace('.','',regex = False)
    .str.replace(',','.',regex = False)
    .str.strip()
    .astype(float)
)
```

df2

DNI	Unnamed: 0	Nom	Cognoms	País d'origen	Ciutat	Dia	Mes	Any	Gènere	Salari mensual	Fills	Grup Professional	nomcomplet	nascuda_Espanya
28973553Z	21	Mia	Schneider Fischer	Alemanya	Berlín	22	10	1976	Altres	951.0	No	Grup A	MiaSchneider Fischer	No
37399141L	154	Laura	Schneider Fischer	Alemanya	Berlín	2	2	1958	Dona	1769.0	Si	Grup B	LauraSchneider Fischer	No
37368317L	224	Lea	Schneider Schneider	Alemanya	Berlín	23	10	2005	Dona	2013.0	No	Grup B	LeaSchneider Schneider	No
21390098Z	278	Mia	Fischer	Alemanya	Berlín	11	8	1950	Dona	1557.0	Si	Grup B	MiaFischer	No
44060014R	602	Jonas	Schneider	Alemanya	Berlín	22	11	1985	Home	2754.0	Si	Grup D	JonasSchneider	No
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
89577876S	547	Emily	Taylor Jones	Regne Unit	Manchester	28	3	1958	Dona	2033.0	No	Grup B	EmilyTaylor Jones	No
57441590Y	728	George	Brown Jones	Regne Unit	Manchester	27	12	1979	Home	1130.0	Si	Grup A	GeorgeBrown Jones	No
58204038A	751	Olivia	Brown Brown	Regne Unit	Manchester	28	8	1952	Altres	1023.0	No	Grup A	OliviaBrown Brown	No
28367234K	854	Isla	Jones Brown	Regne Unit	Manchester	28	3	1999	Dona	1197.0	No	Grup A	IslaJones Brown	No

```
summary = df2.groupby('Gènere')['Salari mensual'].agg(
    Average = 'mean',
    Median = 'median',
    Maximum   = 'max',
```

```
    Minimum  = 'min'
).reset_index()

summary[['Average','Median','Maximum','Minimum']] =
summary[['Average','Median','Maximum','Minimum']].round(2)

print(summary)
```

	Gènere	Average	Median	Maximum	Minimum
0	Altres	1626.59	1545.0	3175.0	703.0
1	Dona	1469.44	1361.5	3021.0	665.0
2	Home	1643.25	1531.0	3356.0	737.0
3	Nan	1568.87	1443.0	2969.0	758.0

```
summary = summary.sort_values(by = 'Average', ascending = False)

print(summary)
```

	Gènere	Average	Median	Maximum	Minimum
2	Home	1643.25	1531.0	3356.0	737.0
0	Altres	1626.59	1545.0	3175.0	703.0
3	Nan	1568.87	1443.0	2969.0	758.0
1	Dona	1469.44	1361.5	3021.0	665.0

Create a summary table with the average salary by gender (rows) and country of origin (columns).

Add the averages to the margins of the table.

(EXTRA): Apply conditional formatting to the table to see the highest values in a more intense colo

```
pivot = df2.pivot_table(
    index='Gènere',
    values='Salari mensual',
    aggfunc='mean',
    columns="País d'origen",
    margins=True,
    margins_name='Mitjana General'
).round(2)
```

```
print(pivot)
```

```
País d'origen    Alemanya  Argentina  Colòmbia  Espanya   França   Itàlia
Gènere
Altres             951.00    1141.00   1030.00  1706.18      NaN  1423.00
Dona              1804.31    1291.80   1497.75  1460.16  1566.47  1247.18
Home              2067.43    1583.29   1554.67  1682.11  1389.25  1672.88
Nan               1931.50    1135.67   1252.00  1597.14  1573.00  1316.00
Mitjana General   1858.35    1431.68   1489.13  1582.16  1465.36  1421.17

País d'origen     Marroc    Mèxic  Portugal  Regne Unit  Mitjana General
Gènere
Altres           1365.00  1372.00   1765.00     1921.00          1626.59
Dona             1405.21  1517.80   1488.55     1489.46          1469.44
Home             1531.00  1625.00   1497.00     1162.56          1643.25
Nan              1365.50  1583.00   1553.50     1758.00          1568.87
Mitjana General  1441.69  1559.08   1527.23     1448.33          1561.46
```

```
styled_pivot = pivot.style.background_gradient(
    cmap='PuBu',
```

```
    axis=None
).format("{:,.2f} €")
```

styled_pivot

País d'origen Gènere	Alemanya	Argentina	Colòmbia	Espanya	França	Itàlia	Marroc	Mèxic	Portugal	Regne Unit	Mitjana General
Altres	951.00 €	1,141.00 €	1,030.00 €	1,706.18 €	nan €	1,423.00 €	1,365.00 €	1,372.00 €	1,765.00 €	1,921.00 €	1,626.59 €
Dona	1,804.31 €	1,291.80 €	1,497.75 €	1,460.16 €	1,566.47 €	1,247.18 €	1,405.21 €	1,517.80 €	1,488.55 €	1,489.46 €	1,469.44 €
Home	2,067.43 €	1,583.29 €	1,554.67 €	1,682.11 €	1,389.25 €	1,672.88 €	1,531.00 €	1,625.00 €	1,497.00 €	1,162.56 €	1,643.25 €
Nan	1,931.50 €	1,135.67 €	1,252.00 €	1,597.14 €	1,573.00 €	1,316.00 €	1,365.50 €	1,583.00 €	1,553.50 €	1,758.00 €	1,568.87 €
Mitjana General	1,858.35 €	1,431.68 €	1,489.13 €	1,582.16 €	1,465.36 €	1,421.17 €	1,441.69 €	1,559.08 €	1,527.23 €	1,448.33 €	1,561.46 €

Create a new column that is the date of birth in Datetime format from the day, month and year columns. Using this column create a function that, given a date, calculates your current age as of today.

Use the function you just created to generate a new column in the DataFrame with the current age.

print(df2.columns)

```
Index(['Unnamed: 0', 'Nom', 'Cognoms', 'País d'origen', 'Ciutat', ' Dia',
       'Mes', 'Any', 'Gènere', 'Salari mensual', 'Fills', 'Grup Professional',
       'nomcomplet', ' nascuda_Espanya'],
```

```
      dtype='object')

df2.columns = df2.columns.str.strip()
print(df2.columns)

Index(['Unnamed: 0', 'Nom', 'Cognoms', 'País d'origen', 'Ciutat', 'Dia', 'Mes',
       'Any', 'Gènere', 'Salari mensual', 'Fills', 'Grup Professional',
       'nomcomplet', 'nascuda_Espanya'],
      dtype='object')
print(df2.columns.tolist())

['Unnamed: 0', 'Nom', 'Cognoms', "País d'origen", 'Ciutat', 'Dia', 'Mes', 'Any', 'Gènere', 'Salari mensual', 'Fills', 'Grup Professional', 'nomcomplet', 'nascuda_Espanya']

df2['data'] = pd.to_datetime(df2['Any'].astype(str) + '-' +
                             df2['Mes'].astype(str) + '-' +
                             df2['Dia'].astype(str), format='%Y-%m-%d')
df2
```

DNI	Unnamed: 0	Nom	Cognoms	País d'origen	Ciutat	Dia	Mes	Any	Gènere	Salari mensual	Fills	Grup Professional	nomcomplet	nascuda_Espanya	data
28973553Z	21	Mia	Schneider Fischer	Alemanya	Berlín	22	10	1976	Altres	951.0	No	Grup A	MiaSchneider Fischer	No	1976-10-22
37399141L	154	Laura	Schneider Fischer	Alemanya	Berlín	2	2	1958	Dona	1769.0	Si	Grup B	LauraSchneider Fischer	No	1958-02-02
37368317L	224	Lea	Schneider Schneider	Alemanya	Berlín	23	10	2005	Dona	2013.0	No	Grup B	LeaSchneider Schneider	No	2005-10-23
21390098Z	278	Mia	Fischer	Alemanya	Berlín	11	8	1950	Dona	1557.0	Si	Grup B	MiaFischer	No	1950-08-11
44060014R	602	Jonas	Schneider	Alemanya	Berlín	22	11	1985	Home	2754.0	Si	Grup D	JonasSchneider	No	1985-11-22
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
89577876S	547	Emily	Taylor Jones	Regne Unit	Manchester	28	3	1958	Dona	2033.0	No	Grup B	EmilyTaylor Jones	No	1958-03-28
57441590Y	728	George	Brown Jones	Regne Unit	Manchester	27	12	1979	Home	1130.0	Si	Grup A	GeorgeBrown Jones	No	1979-12-27
58204038A	751	Olivia	Brown Brown	Regne Unit	Manchester	28	8	1952	Altres	1023.0	No	Grup A	OliviaBrown Brown	No	1952-08-28

```
def calculate_age(birthdate):
    today = pd.Timestamp('today')
    age = today.year - birthdate.dt.year
    age -= ((today.month < birthdate.dt.month) |
            ((today.month == birthdate.dt.month) & (today.day < birthdate.dt.day)))
    return age


# Create age column
df2['edat'] = calculate_age(df2['data'])


df2
```

DNI	Unnamed: 0	Nom	Cognoms	País d'origen	Ciutat	Dia	Mes	Any	Gènere	Salari mensual	Fills	Grup Professional	nomcomplet	nascuda_Espanya	data
28973553Z	21	Mia	Schneider Fischer	Alemanya	Berlín	22	10	1976	Altres	951.0	No	Grup A	MiaSchneider Fischer	No	1976-10-22
37399141L	154	Laura	Schneider Fischer	Alemanya	Berlín	2	2	1958	Dona	1769.0	Si	Grup B	LauraSchneider Fischer	No	1958-02-02
37368317L	224	Lea	Schneider Schneider	Alemanya	Berlín	23	10	2005	Dona	2013.0	No	Grup B	LeaSchneider Schneider	No	2005-10-23
21390098Z	278	Mia	Fischer	Alemanya	Berlín	11	8	1950	Dona	1557.0	Si	Grup B	MiaFischer	No	1950-08-11
44060014R	602	Jonas	Schneider	Alemanya	Berlín	22	11	1985	Home	2754.0	Si	Grup D	JonasSchneider	No	1985-11-22
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
89577876S	547	Emily	Taylor Jones	Regne Unit	Manchester	28	3	1958	Dona	2033.0	No	Grup B	EmilyTaylor Jones	No	1958-03-28
57441590Y	728	George	Brown Jones	Regne Unit	Manchester	27	12	1979	Home	1130.0	Si	Grup A	GeorgeBrown Jones	No	1979-12-27

Level 2

1.

Using the following DataFrame, append the "Increment" column to the dataframe of the previous level.

Update the salary column based on the percentages attached. Don't manually modify the increments, write Python code to do the necessary conversions.

df_increment = pd.DataFrame({"Group":["Group A","Group B","Group C", "Group D"] , "Increment":

["5%","3.5%","2%","8%"]})

df_increment = pd.DataFrame({"Grup":["Grup A","Grup B","Grup C", "Grup D"] , "Increment":

["5%","3.5%","2%","8%"]})

df_increment

	Grup	Increment
0	Grup A	5%
1	Grup B	3.5%
2	Grup C	2%
3	Grup D	8%

```
df_increment['Increment'] = df_increment['Increment'].str.rstrip('%').astype(float) / 100
```

```
df2 = df2.merge(df_increment, left_on='Grup Professional', right_on='Grup', how='left')
```

```
df2
```

Unnamed: 0	Nom	Cognoms	País d'origen	Ciutat	Dia	Mes	Any	Gènere	Salari mensual	Fills	Grup Professional	nomcomplet	nascuda_Espanya	data	edat	Grup
21	Mia	Schneider Fischer	Alemanya	Berlín	22	10	1976	Altres	951.0	No	Grup A	MiaSchneider Fischer	No	1976-10-22	49	Grup A
154	Laura	Schneider Fischer	Alemanya	Berlín	2	2	1958	Dona	1769.0	Si	Grup B	LauraSchneider Fischer	No	1958-02-02	67	Grup B
224	Lea	Schneider Schneider	Alemanya	Berlín	23	10	2005	Dona	2013.0	No	Grup B	LeaSchneider Schneider	No	2005-10-23	20	Grup B
278	Mia	Fischer	Alemanya	Berlín	11	8	1950	Dona	1557.0	Si	Grup B	MiaFischer	No	1950-08-11	75	Grup B
602	Jonas	Schneider	Alemanya	Berlín	22	11	1985	Home	2754.0	Si	Grup D	JonasSchneider	No	1985-11-22	39	Grup D
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
547	Emily	Taylor Jones	Regne Unit	Manchester	28	3	1958	Dona	2033.0	No	Grup B	EmilyTaylor Jones	No	1958-03-28	67	Grup B
728	George	Brown Jones	Regne Unit	Manchester	27	12	1979	Home	1130.0	Si	Grup A	GeorgeBrown Jones	No	1979-12-27	45	Grup A
751	Olivia	Brown Brown	Regne Unit	Manchester	28	8	1952	Altres	1023.0	No	Grup A	OliviaBrown Brown	No	1952-08-28	73	Grup A
854	Isla	Jones Brown	Regne Unit	Manchester	28	3	1999	Dona	1197.0	No	Grup A	IslaJones Brown	No	1999-03-28	26	Grup A

df2['Salari mensual'] = df2['Salari mensual'] * (1 + df2['Increment'])

df2

	Unnamed: 0	Nom	Cognoms	País d'origen	Ciutat	Dia	Mes	Any	Gènere	Salari mensual	Fills	Grup Professional	nomcomplet	nascuda_Espanya	data	edat
0	21	Mia	Schneider Fischer	Alemanya	Berlín	22	10	1976	Altres	998.550	No	Grup A	MiaSchneider Fischer	No	1976-10-22	49
1	154	Laura	Schneider Fischer	Alemanya	Berlín	2	2	1958	Dona	1830.915	Si	Grup B	LauraSchneider Fischer	No	1958-02-02	67
2	224	Lea	Schneider Schneider	Alemanya	Berlín	23	10	2005	Dona	2083.455	No	Grup B	LeaSchneider Schneider	No	2005-10-23	20
3	278	Mia	Fischer	Alemanya	Berlín	11	8	1950	Dona	1611.495	Si	Grup B	MiaFischer	No	1950-08-11	75
4	602	Jonas	Schneider	Alemanya	Berlín	22	11	1985	Home	2974.320	Si	Grup D	JonasSchneider	No	1985-11-22	39
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	547	Emily	Taylor Jones	Regne Unit	Manchester	28	3	1958	Dona	2104.155	No	Grup B	EmilyTaylor Jones	No	1958-03-28	67
996	728	George	Brown Jones	Regne Unit	Manchester	27	12	1979	Home	1186.500	Si	Grup A	GeorgeBrown Jones	No	1979-12-27	45
997	751	Olivia	Brown Brown	Regne Unit	Manchester	28	8	1952	Altres	1074.150	No	Grup A	OliviaBrown Brown	No	1952-08-28	73
998	854	Isla	Jones Brown	Regne Unit	Manchester	28	3	1999	Dona	1256.850	No	Grup A	IslaJones Brown	No	1999-03-28	26

2.

Using a loop, export the data for each Professional Group into 4 files (.xlsx or .csv format).

For example: "data_GroupA.xlsx", "data_GroupB.xlsx" ...

Export a 5th DataFrame in .xlsx or .csv format that contains how many workers there are for each Professional Group,

what their average salary is, and what their median age is.

```
for group,data in df2.groupby('Grup Professional'):
    group_clean=group.replace(' ','')
    data.to_excel(f"C:/Personal/Susi/Barcelona_Activa/Export/data_{group_clean}.xlsx",index = False)

summary1 = df2.groupby('Salari mensual').agg(
    Num_workers = ('Nom','count'),
    Average_salary = ('Salari mensual','mean'),
    Median_Age = ('edat','median')
).reset_index()

summary1.to_excel('C:/Personal/Susi/Barcelona_Activa/Export/summary_group.xlsx' ,index = False)

import glob
print(glob.glob("*.xlsx"))
```

['data_GrupA.xlsx', 'data_GrupB.xlsx', 'data_GrupC.xlsx', 'data_GrupD.xlsx', 'summary_group.xlsx']

Level 3

Level 3 of this sprint is completely different from other sprints you've done so far,

as they are more abstract exercises that require a lot of fighting.

They don't continue with the same dataset from the previous levels,

but rather present you with two new situations that are completely different from each other.

1.

Create a function that takes a dataframe as an input parameter.

The function should automatically create (and export) a chart for each column in the dataframe. For example:

a histogram/boxplot if the variable is numeric

some bars of the most frequent values if it is categorical

some bars of the most frequent years if the data is in date format.

The idea is to create a function that works for any dataframe, not just the one we've worked with so far.

Show the result of the function on one of the example datasets contained in the seaborn package.

For example, iris , penguins or titanic .

Keep in mind that in the next sprint you will work exclusively with graphics.

The goal of this exercise is not to create very elaborate graphics, but to solve a need quickly and automatically.

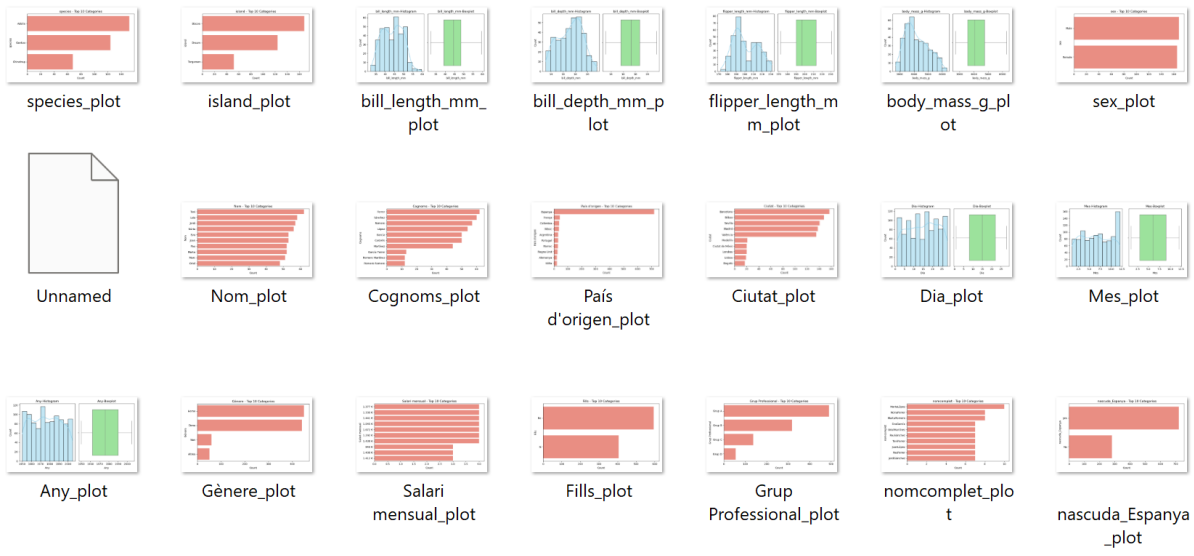
import pandas as pd

import seaborn as sns

```
import matplotlib.pyplot as plt
import os

def auto_plot(df,export_path="charts"):
    export_path = os.path.abspath(export_path)
    os.makedirs(export_path,exist_ok = True)
    print(f"The folder chart saved to:{export_path}\n")

    for col in df.columns:
        series=df[col].dropna()
        print(f"creating chart for:{col} ({series.dtype})")
        plt.figure(figsize=(7,4))

        if pd.api.types.is_numeric_dtype(series):
            plt.subplot(1,2,1)
            sns.histplot(series,kde=True,color="skyblue")
            plt.title(f"{col}-Histogram")



            plt.subplot(1,2,2)
            sns.boxplot(x=series,color = "lightgreen")
            plt.title(f"{col}-Boxplot")

        elif pd.api.types.is_datetime64_any_dtype(series):
            year_counts=series.dt.year.value_counts().sort_index()
            sns.barplot(x=year_counts.index, y=year_counts.values, color="orange")
```

```
            plt.title(f"{col} -Counts by year")
            plt.xlabel("Year")
            plt.ylabel("count")
        else:
            top_values = series.value_counts().head(10)
            sns.barplot(x=top_values.values,y=top_values.index,color = "salmon")
            plt.title(f"{col} - Top 10 Categories")
            plt.xlabel("Count")
            plt.ylabel(col)

        plt.tight_layout()
        file_path = os.path.join(export_path,f"{col}_plot.png")
        plt.savefig(file_path)
        plt.close()

    print(f"\n All charts exported successfully to: {export_path}")

auto_plot(df2)
```

```
The folder chart saved to:C:\Users\sarav\charts

creating chart for:Unnamed: 0 (int64)
creating chart for:Nom (object)
creating chart for:Cognoms (object)
creating chart for:País d'origen (object)
creating chart for:Ciutat (object)
creating chart for: Dia (int64)
creating chart for:Mes (int64)
creating chart for:Any (int64)
creating chart for:Gènere (object)
creating chart for:Salari mensual (object)
creating chart for:Fills (object)
creating chart for:Grup Professional (object)
creating chart for:nomcomplet (object)
creating chart for: nascuda_Espanya (object)

 All charts exported successfully to: C:\Users\sarav\charts
```



df_penguins = sns.load_dataset("penguins")

auto_plot(df_penguins)

```
The folder chart saved to:C:\Users\sarav\charts

creating chart for:species (object)
creating chart for:island (object)
creating chart for:bill_length_mm (float64)
creating chart for:bill_depth_mm (float64)
creating chart for:flipper_length_mm (float64)
creating chart for:body_mass_g (float64)
creating chart for:sex (object)

 All charts exported successfully to: C:\Users\sarav\charts
```

- 2.

Load the file matriu_distancies.xlsx into pandas, so that the row names and column names are those of the cities. Delete "Las Palmas de Gran Canaria" and "Palma" so that we can make the trip by car.

Source: Best Routes

We are interested in visiting all the main cities in Spain while traveling the shortest possible distance.

You don't have to do it optimally, we are interested in you developing a reasonable solution using the tools you currently have.

For example, a simple (but not optimal) approach would be to always go to the nearest city that we haven't visited yet.

Make a function that, given the distance matrix and the origin city, proposes a route that is as short as possible, returning a list with the order of visits. Also give the total distance traveled.

(EXTRA) From which city would the route be shortest with the proposed algorithm?

import pandas as pd

import numpy as np

```
def load_distance_matrix(filepath):
    df = pd.read_excel(filepath, index_col=0)
    df = df.drop(["Las Palmas de Gran Canaria", "Palma"], errors='ignore')
    df = df.drop(columns=["Las Palmas de Gran Canaria", "Palma"], errors='ignore')
    return df



def nearest_neighbor_route(dist_matrix, origin):
    cities = list(dist_matrix.index)
    visited = [origin]
    total_distance = 0
    current_city = origin

    while len(visited) < len(cities):
        unvisited = [c for c in cities if c not in visited]
        next_city = dist_matrix.loc[current_city, unvisited].idxmin()
        dist = dist_matrix.loc[current_city, next_city]
        total_distance += dist
        visited.append(next_city)
        current_city = next_city

    # Return to origin
    total_distance += dist_matrix.loc[current_city, origin]
    visited.append(origin)

    return visited, total_distance
```

```
def find_best_start_city(dist_matrix):
    results = {}
    for city in dist_matrix.index:
        route, distance = nearest_neighbor_route(dist_matrix, city)
        results[city] = distance

    best_city = min(results, key=results.get)
    return best_city, results[best_city], results


#  Main execution

df = load_distance_matrix("C:/Users/sarav/Downloads/matriu_distancies (1).xlsx")

route, total = nearest_neighbor_route(df, "Barcelona")
print("Route:", " → ".join(route))
print("Total distance:", round(total, 2), "km")

best_city, best_distance, all_results = find_best_start_city(df)
print(f"\nBest starting city: {best_city} ({best_distance:.0f} km)")
```

```
Route: Barcelona → Hospitalet de Llobregat → Zaragoza → Valencia → Alicante → Murcia → Córdoba → Sevilla → Málaga
→ Valladolid → Gijón → Bilbao → Vigo → Barcelona
Total distance: 3686.0 km

Best starting city: Alicante (3541 km)
```