seminarioCD5_graph_seaborn

August 27, 2025

[ ]:
```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
```

[ ]:
```
url="/content/drive/MyDrive/Anahuac/2024 Agosto-Diciembre/SeminarioCD/Python/
↪salary.csv"
data = pd.read_csv(url)
```

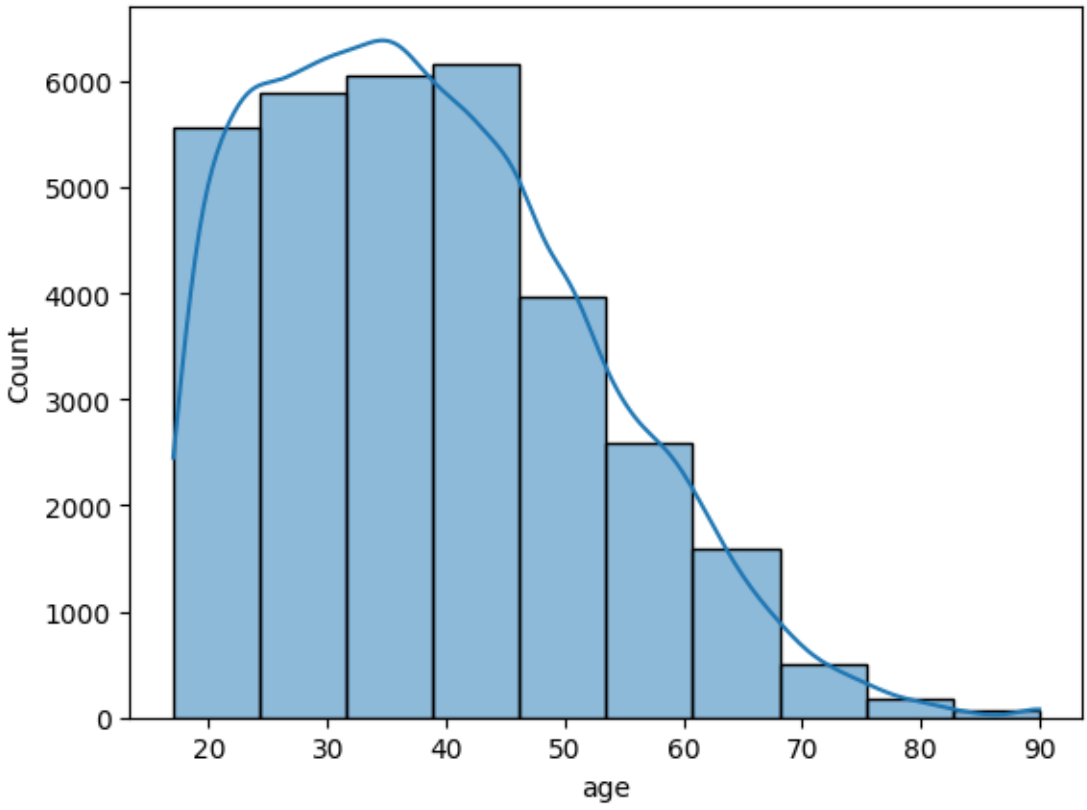
[ ]:
```
data.head()
```

[ ]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	salary
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

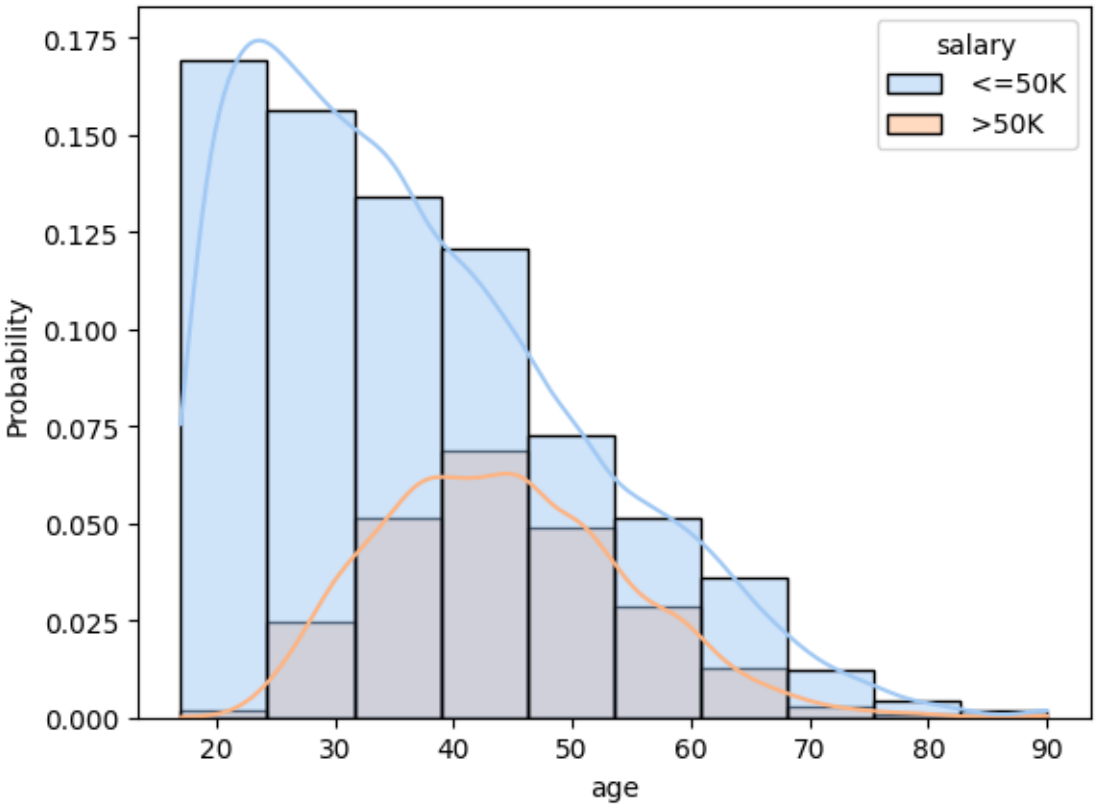
[ ]:
```
sns.histplot(x=data["age"], bins=10, kde=True)
```

[ ]:
```
<Axes: xlabel='age', ylabel='Count'>
```



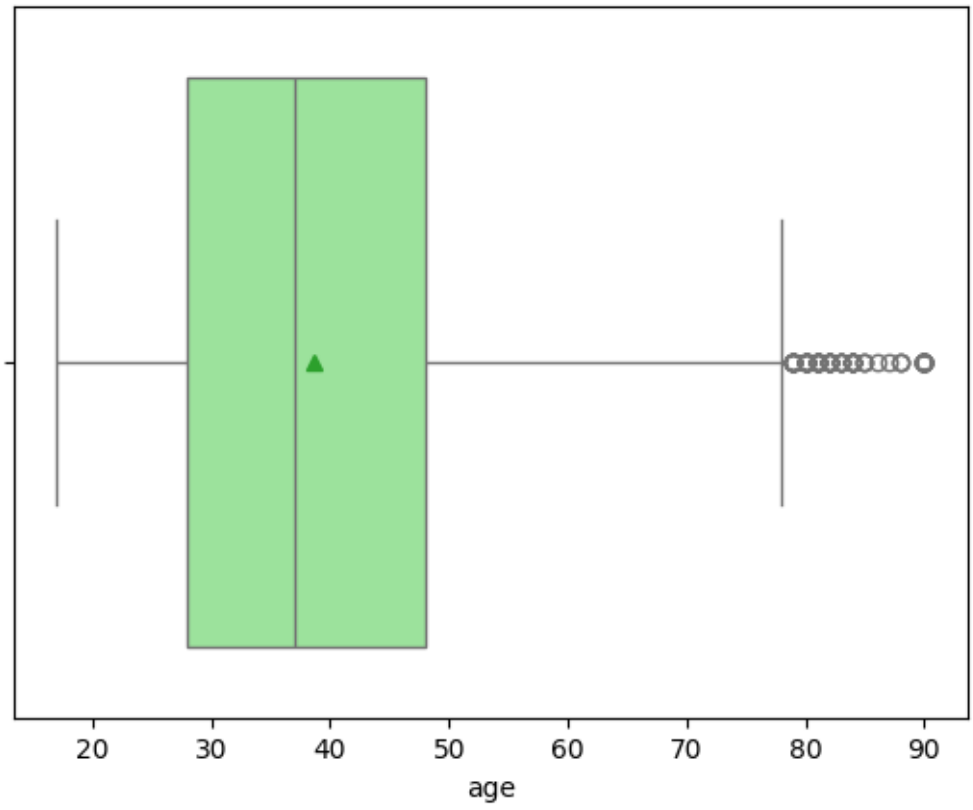
[ ]:
```
sns.histplot(x=data["age"], bins=10,␣
↪kde=True,stat="probability",hue=data["salary"], palette="pastel")
```

[ ]:
```
<Axes: xlabel='age', ylabel='Probability'>
```



[ ]:
```
sns.boxplot(x=data["age"],color="lightgreen", showmeans=True, legend=False)
```

[ ]:
```
<Axes: xlabel='age'>
```



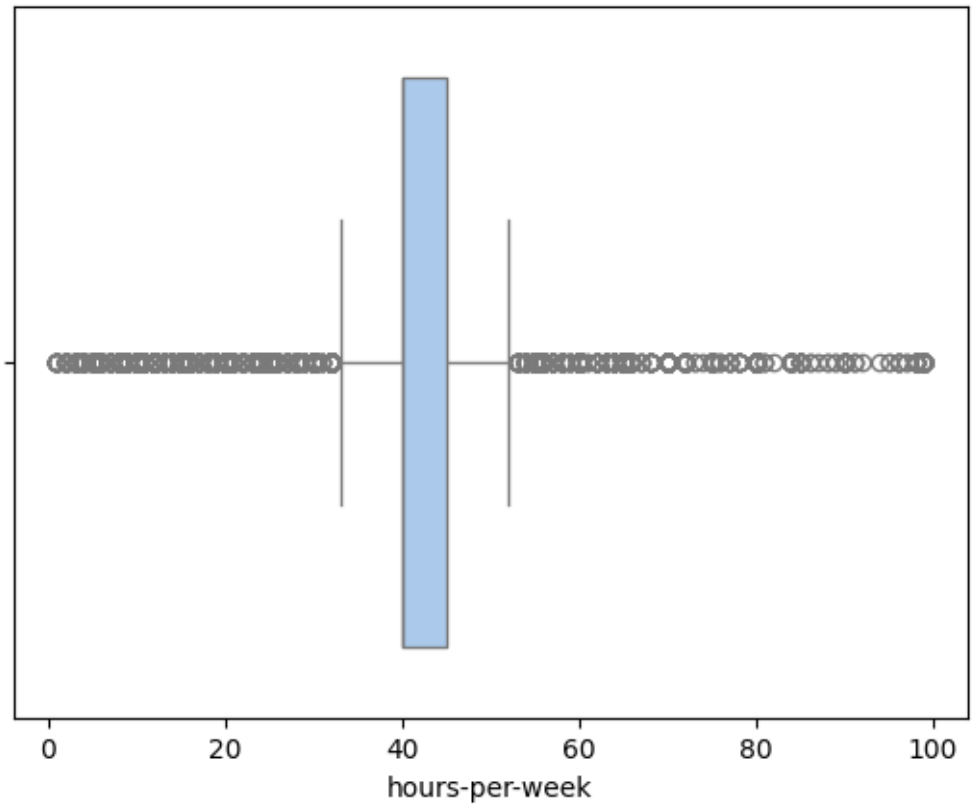
[ ]:
```
sns.boxplot(x=data['hours-per-week'], showmeans=False, palette="pastel")
```

```
/tmp/ipython-input-3107673915.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(x=data['hours-per-week'], showmeans=False, palette="pastel")
```

[ ]:
```
<Axes: xlabel='hours-per-week'>
```



[ ]:
```
tab_fr_sex=data["sex"].value_counts()
```

[ ]:
```
type(tab_fr_sex)
```

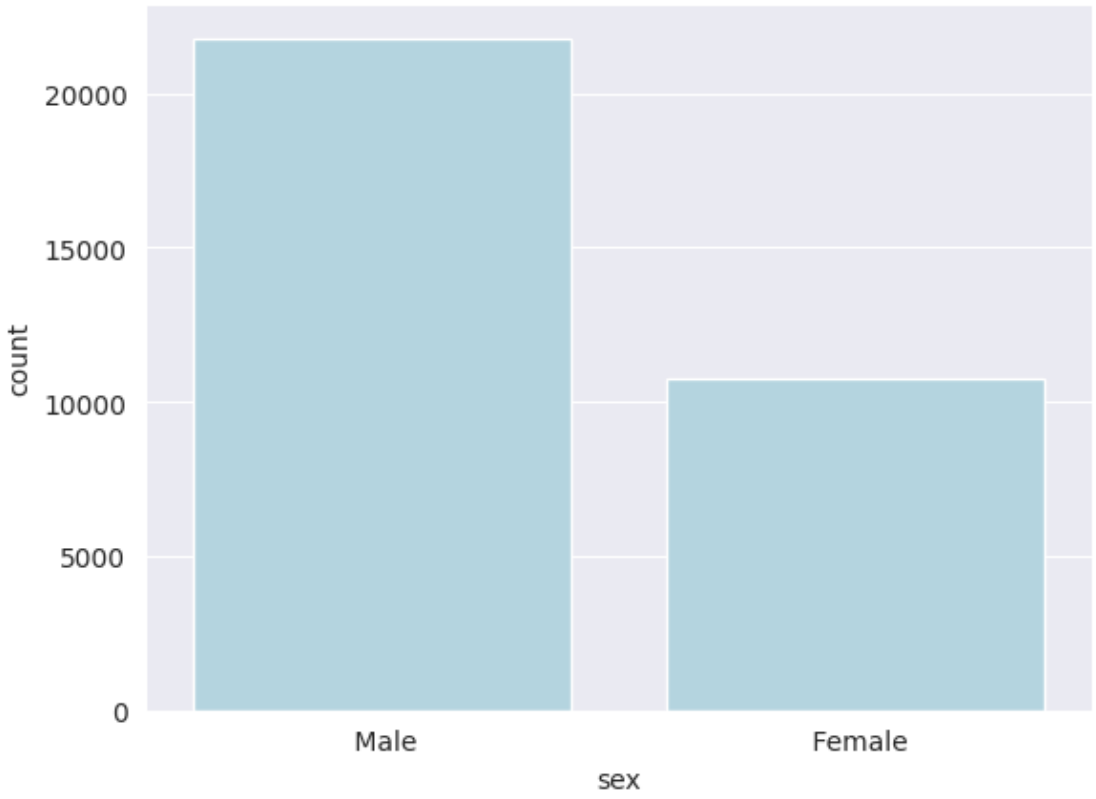
[ ]:
```
pandas.core.series.Series
```

[ ]:
```
tab_fr_sex
```

[ ]:
```
sex
Male      21790
Female    10771
Name: count, dtype: int64
```

[ ]:
```
sns.barplot(tab_fr_sex, color="lightblue")
```

[ ]:
```
<Axes: xlabel='sex', ylabel='count'>
```



[ ]:
```
datos={"labels": ["A", "B", "C", "D"],
       "values": [10, 35, 25, 20]}
df=pd.DataFrame(datos)
```

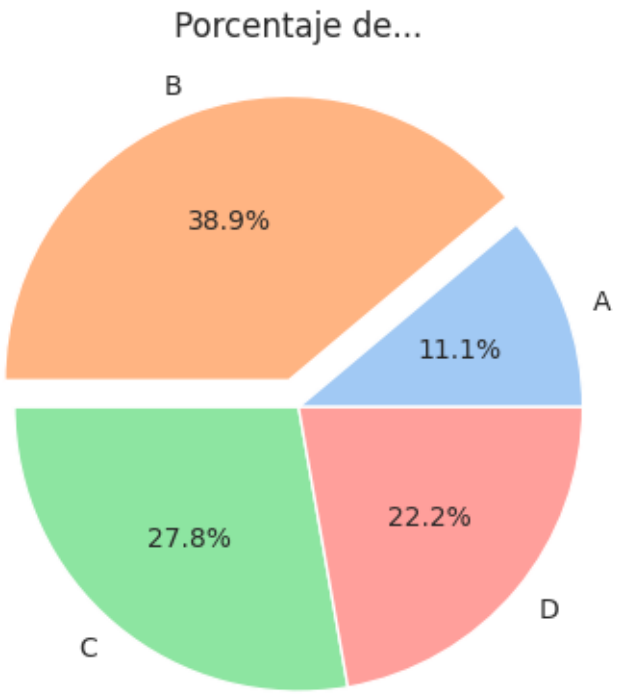
[ ]:
```
df.head()
```

[ ]:

	labels	values
0	A	10
1	B	35
2	C	25
3	D	20

[ ]:
```
expl=[0,0.1,0,0]
```

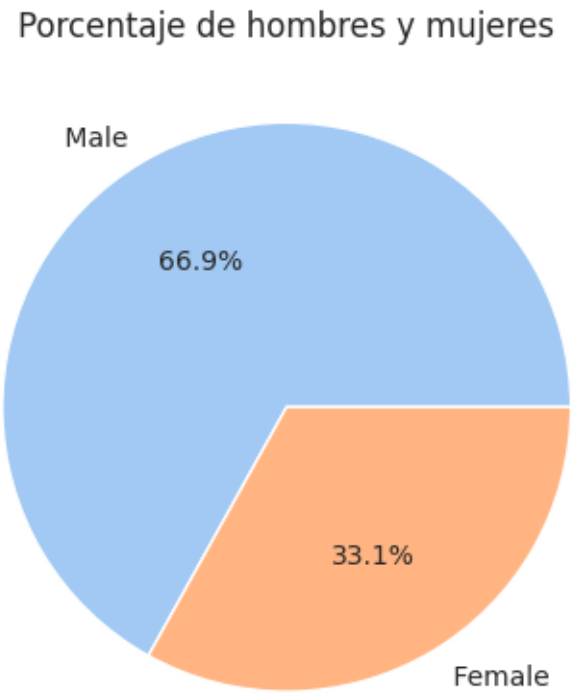
[ ]:
```
sns.set_style("whitegrid")
colores=sns.color_palette("pastel")
plt.pie(df["values"], labels=df["labels"], autopct="%1.1f%%", explode=expl,␣
 ↪colors=colores)
plt.title("Porcentaje de...")
plt.show()
```



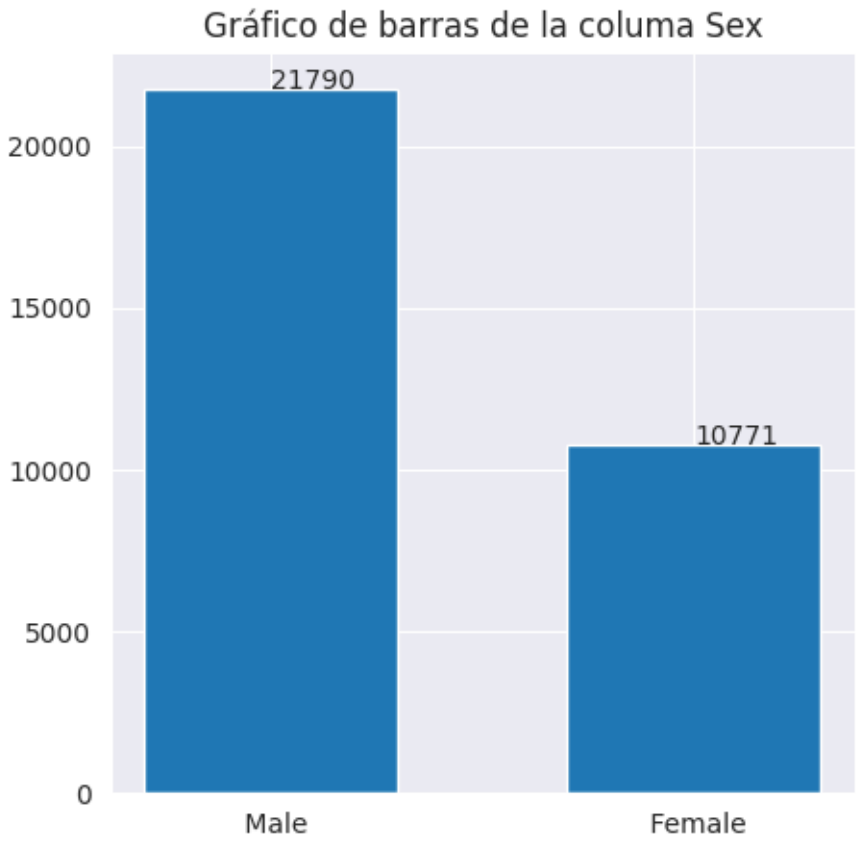
[ ]:
```
data_sex=pd.DataFrame({"sex":tab_fr_sex.index, "frequency":tab_fr_sex.values})
data_sex.head()
```

[ ]:
```
        sex  frequency
0      Male      21790
1    Female      10771
```

[ ]:
```
sns.set_style("darkgrid")
colores=sns.color_palette("pastel")
plt.pie(data_sex["frequency"], labels=data_sex["sex"], autopct="%1.1f%%",␣
 ↪colors=colores)
plt.title("Porcentaje de hombres y mujeres")
plt.show()
```



[ ]:
```
sns.set_style("darkgrid")
plt.figure(figsize=(5,5))
plt.bar(x=data_sex["sex"], height=data_sex["frequency"], width=0.6)
for a, b in enumerate(data_sex["frequency"]):
    plt.text(a, b, b)
plt.title("Gráfico de barras de la columa Sex")
plt.show()
```



[ ]:
```
data_cuanti=data.select_dtypes(include=["float64", "int64"])
data_cuanti.head()
```

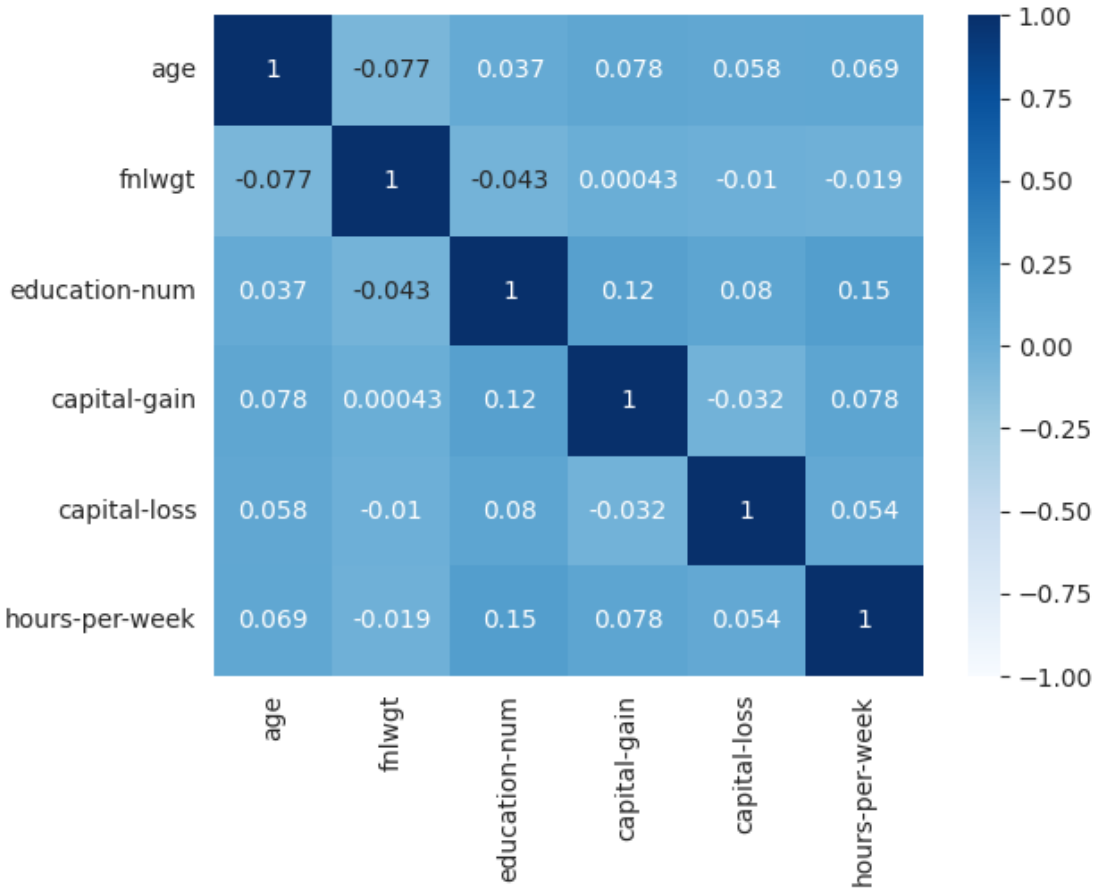
[ ]:

	age	fnlwgt	education-num	capital-gain	capital-loss	hours-per-week
0	39	77516	13	2174	0	40
1	50	83311	13	0	0	13
2	38	215646	9	0	0	40
3	53	234721	7	0	0	40
4	28	338409	13	0	0	40

Se observa en el gráfico que ...

[ ]:
```
correlation=data_cuanti.corr()
sns.heatmap(correlation, annot=True, cmap="Blues", vmin=-1, vmax=1)
```

[ ]: <Axes: >



ACTIVIDAD 0. Elegir sus datos (diversos en cuanto a los tipos de variable, mínimo de 6 variables) 1. Describir los datos: head(), tail(), info(), shape, describe(), etc. 2. Realizar limpieza de ser necesario 3. Realizar al menos 5 preguntas de los datos 4. Responder dichas preguntas con estadísticos (mean(), max(), min(), count(), filtros) y gráficos (barras, pastel, histograma, dispersión, cajas, correlación)

[ ]: