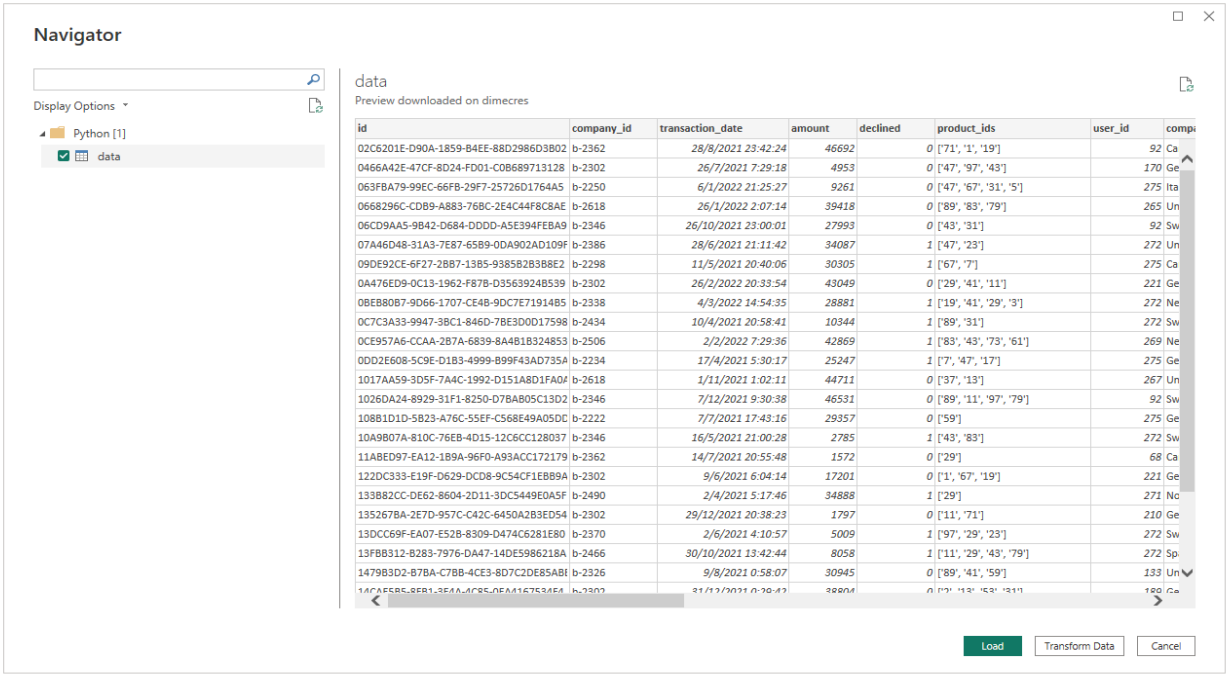
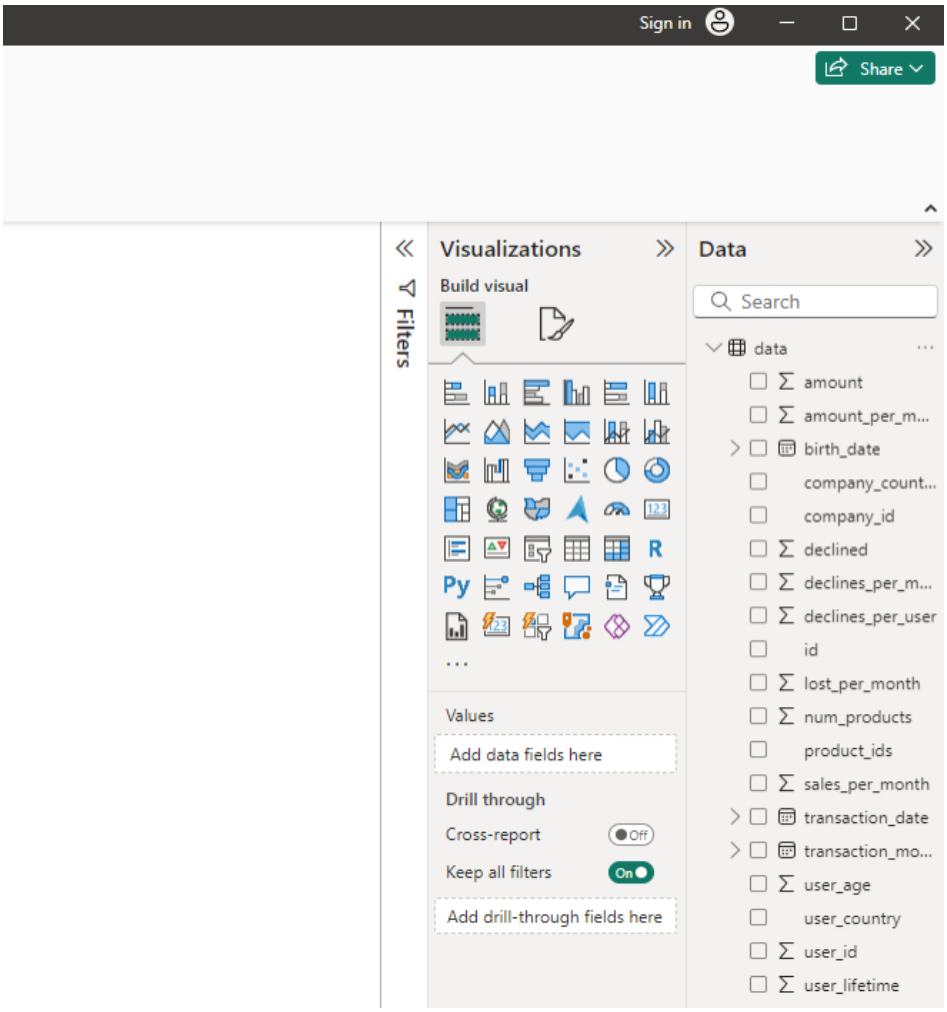
Task 8.02.
Integrating Python and Power BI

MySQL connection to Python was established to load data from previously created `transactions_db` database.

For brevity (to avoid repeating data preprocessing and feature engineering), we will use the `transactions_companies_users.csv` file created in Sprint 8.1, which contains a dataframe with joined tables from the `transactions_db` database.

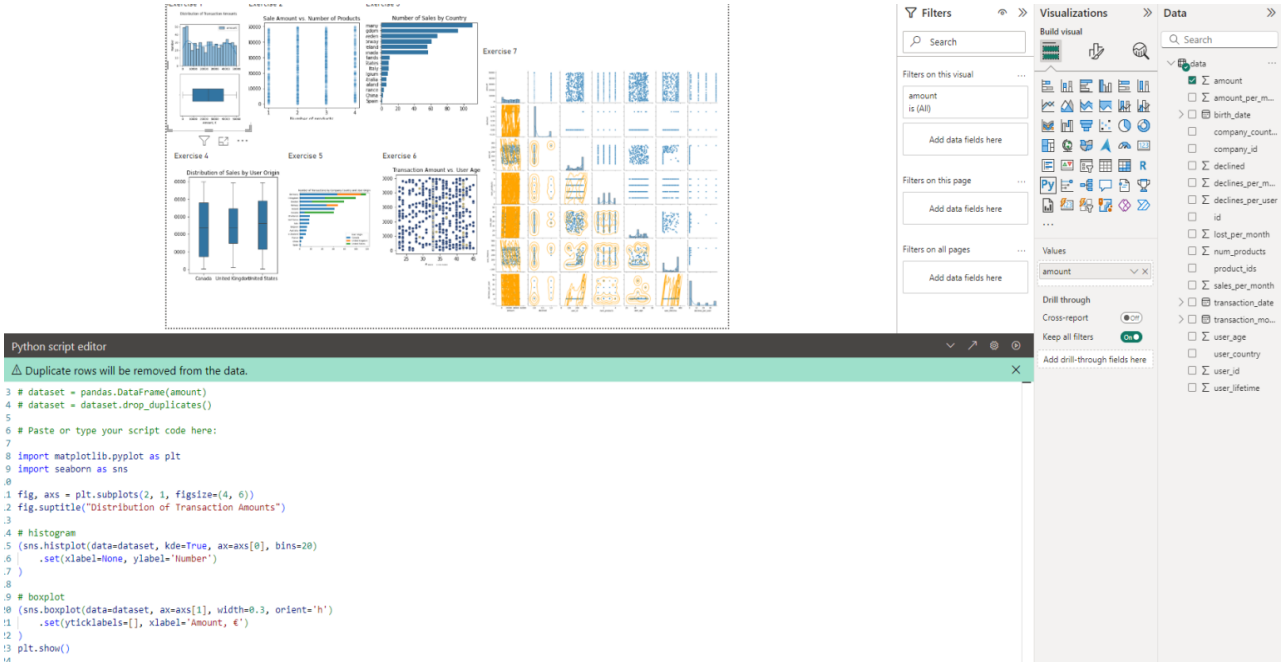


id	company_id	transaction_date	amount	declined	product_ids	user_id	compa
02C6201E-D90A-1859-B4EE-88D2986D3B02	b-2362	28/8/2021 23:42:24	46692	0	['71', '1', '19']	92	Ca
0466A42E-47CF-8D24-FD01-C0B689713128	b-2302	26/7/2021 7:29:18	4953	0	['47', '97', '43']	170	Ge
063FBA79-99EC-66FB-29F7-25726D1764A5	b-2250	6/1/2022 21:25:27	9261	0	['47', '67', '31', '5']	275	Ita
0668296C-CDB9-A883-76BC-2E4C44F8C8AE	b-2618	26/1/2022 2:07:14	39418	0	['89', '83', '79']	265	Un
06CD9AA5-9B42-D684-DDDD-A5E394FEBA9	b-2346	26/10/2021 23:00:01	27993	0	['43', '31']	92	Sw
07A46D48-31A3-7E87-65B9-0DA902AD109F	b-2386	28/6/2021 21:11:42	34087	1	['47', '23']	272	Un
09DE92CE-6F27-2BB7-13B5-9385B2B3B8E2	b-2298	11/5/2021 20:40:06	30305	1	['67', '7']	275	Ca
0A476ED9-0C13-1962-F87B-D3563924B539	b-2302	26/2/2022 20:33:54	43049	0	['29', '41', '11']	221	Ge
0BEB80B7-9D66-1707-CE4B-9DC7E71914B5	b-2338	4/3/2022 14:54:35	28881	1	['19', '41', '29', '3']	272	Ne
0C7C3A33-9947-3BC1-846D-7BE3D0D17598	b-2434	10/4/2021 20:58:41	10344	1	['89', '31']	272	Sw
0CE957A6-CCAA-2B7A-6839-8A4B1B324853	b-2506	2/2/2022 7:29:36	42869	1	['83', '43', '73', '61']	269	Ne
0DD2E608-5C9E-D1B3-4999-B99F43AD735A	b-2234	17/4/2021 5:30:17	25247	1	['7', '47', '17']	275	Ge
1017AA59-3D5F-7A4C-1992-D151A8D1FA0A	b-2618	1/11/2021 1:02:11	44711	0	['37', '13']	267	Un
1026DA24-8929-31F1-8250-D7BAB05C13D2	b-2346	7/12/2021 9:30:38	46531	0	['89', '11', '97', '79']	92	Sw
108B1D1D-5B23-A76C-55EF-C568E49A05DC	b-2222	7/7/2021 17:43:16	29357	0	['59']	275	Ge
10A9B07A-810C-76EB-4D15-12C6CC128037	b-2346	16/5/2021 21:00:28	2785	1	['43', '83']	272	Sw
11ABED97-EA12-1B9A-96F0-A93ACC172179	b-2362	14/7/2021 20:55:48	1572	0	['29']	68	Ca
122DC333-E19F-D629-DCD8-9C54CF1EBB9A	b-2302	9/6/2021 6:04:14	17201	0	['1', '67', '19']	221	Ge
133B82CC-DE62-8604-2D11-3DC5449E0A5F	b-2490	2/4/2021 5:17:46	34888	1	['29']	271	No
135267BA-2E7D-957C-C42C-6450A2B3ED54	b-2302	29/12/2021 20:38:23	1797	0	['11', '71']	210	Ge
13DCC69F-EA07-E52B-8309-D474C6281E80	b-2370	2/6/2021 4:10:57	5009	1	['97', '29', '23']	272	Sw
13FBB312-B283-7976-DA47-14DE5986218A	b-2466	30/10/2021 13:42:44	8058	1	['11', '29', '43', '79']	272	Sp
1479B3D2-B7BA-C7BB-4CE3-8D7C2DE8SABI	b-2326	9/8/2021 0:58:07	30945	0	['89', '41', '59']	133	Un

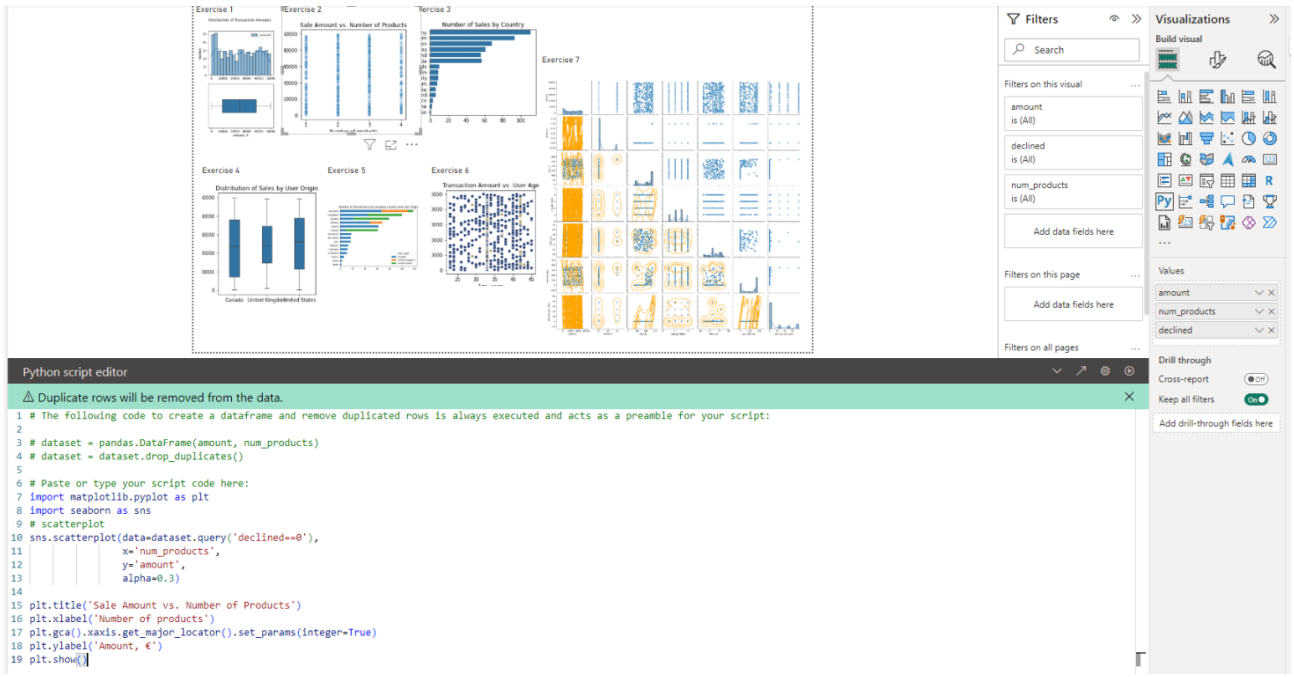

Sign in
Share
Filters
Visualizations
Build visual
Values
Add data fields here
Drill through
Cross-report
Off
Keep all filters
On
Add drill-through fields here
Data
Search
data
amount
amount_per_m...
birth_date
company_count...
company_id
declined
declines_per_m...
declines_per_user
id
lost_per_month
num_products
product_ids
sales_per_month
transaction_date
transaction_mo...
user_age
user_country
user_id
user_lifetime

Level 1.

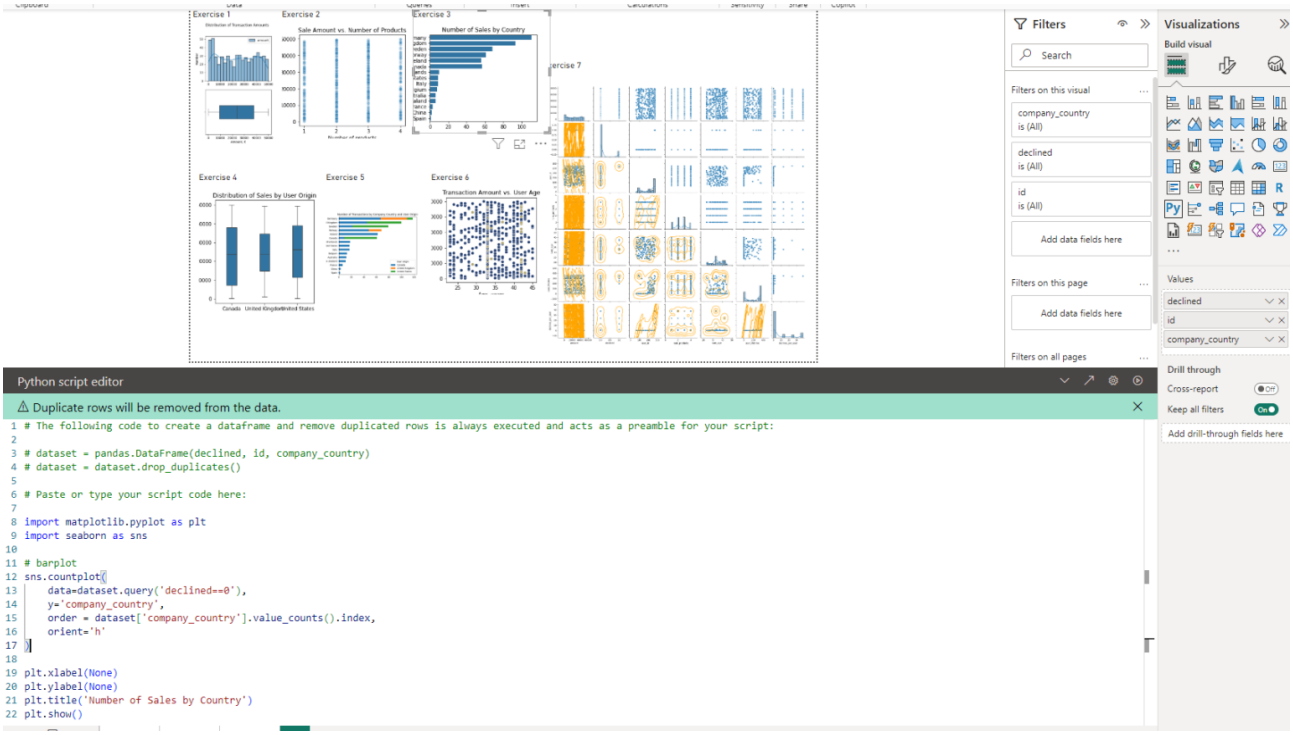
Exercise 1. Single numeric variable.



Exercise 2. Two numerical variables.

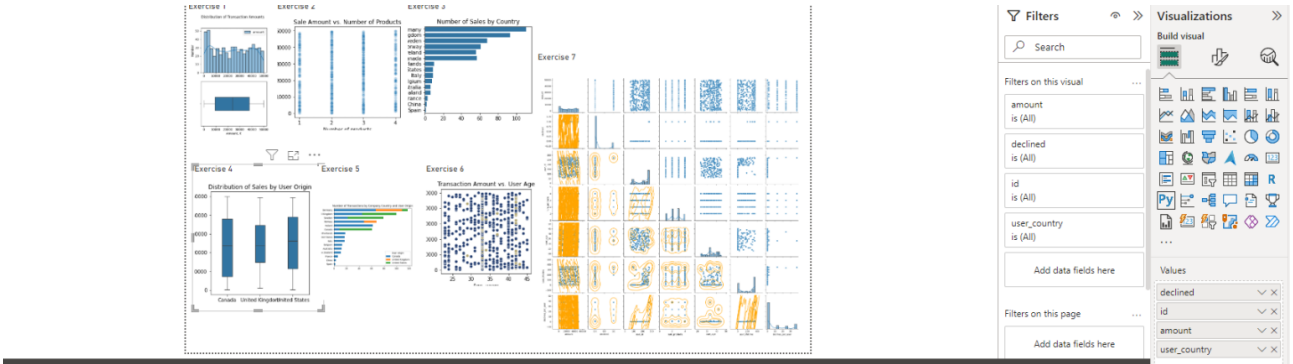


Exercise 3. Single categorical variable



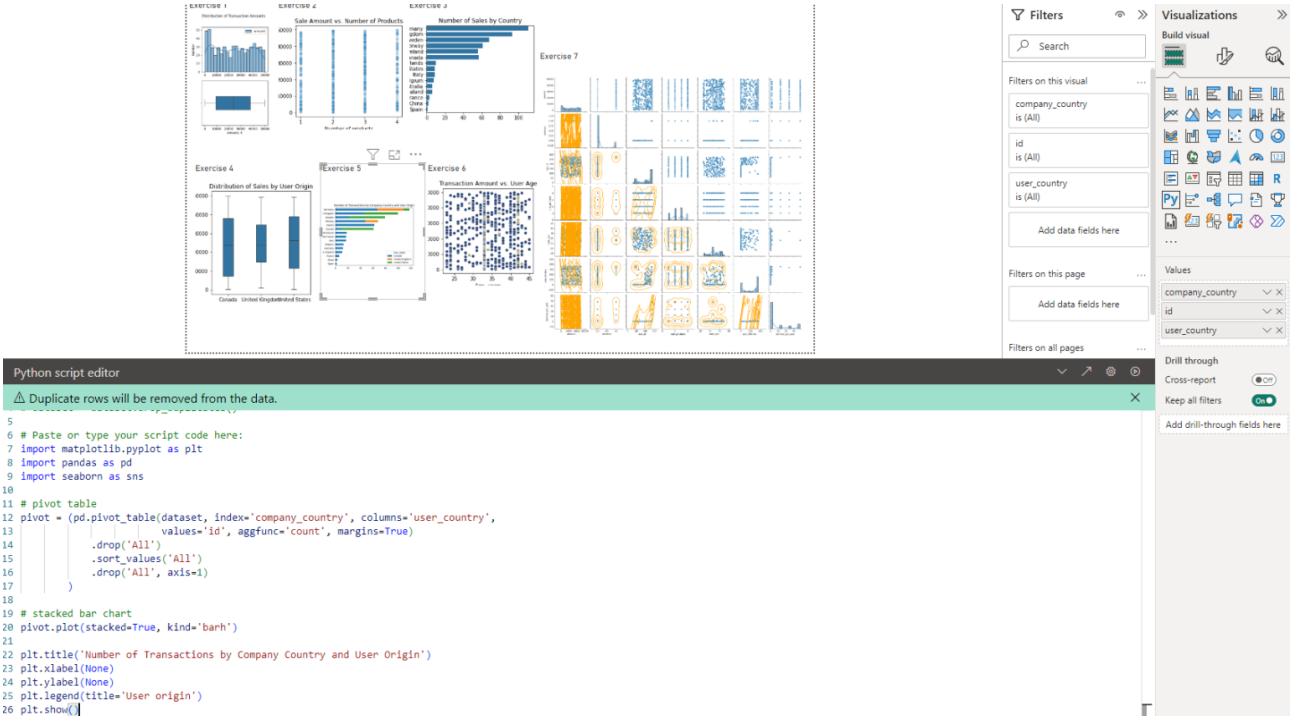

```python
# The following code to create a dataframe and remove duplicated rows is always executed and acts as a preamble for your script:

# dataset = pandas.DataFrame(declined, id, company_country)
# dataset = dataset.drop_duplicates()

# Paste or type your script code here:

import matplotlib.pyplot as plt
import seaborn as sns

# barplot
sns.countplot(
    data=dataset.query('declined==0'),
    y='company_country',
    order = dataset['company_country'].value_counts().index,
    orient='h'
)

plt.xlabel(None)
plt.ylabel(None)
plt.title('Number of Sales by Country')
plt.show()
```

Exercise 4. Categorical and numeric variables.



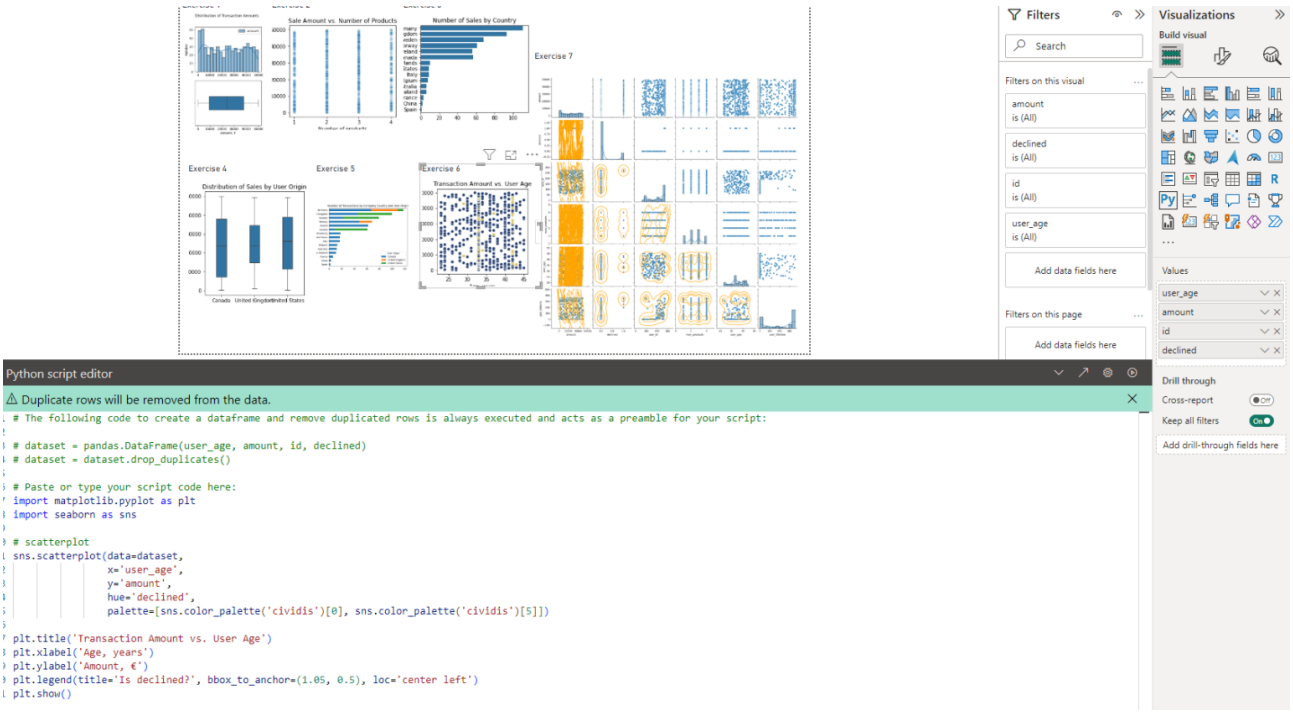

```python
# The following code to create a dataframe and remove duplicated rows is always executed and acts as a preamble for your script:

# dataset = pandas.DataFrame(declined, id, amount, user_country)
# dataset = dataset.drop_duplicates()

# Paste or type your script code here:

import matplotlib.pyplot as plt
import seaborn as sns

# boxplot
sns.boxplot(data=dataset.query('declined==0'),
            x='user_country',
            y='amount',
            width=0.3)

plt.title('Distribution of Sales by User Origin')
plt.ylabel('Amount, €')
plt.xlabel(None)
plt.show()
```

Exercise 5. Two categorical variables.



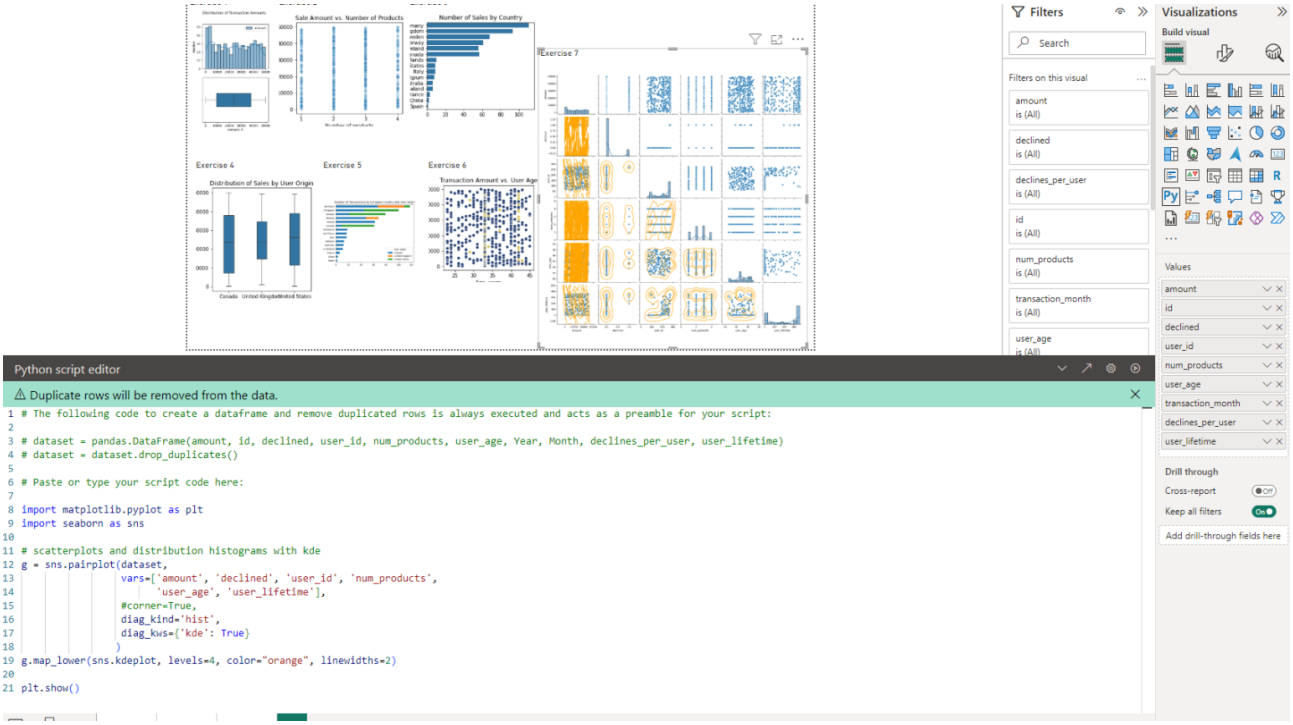

```
5
6  # Paste or type your script code here:
7  import matplotlib.pyplot as plt
8  import pandas as pd
9  import seaborn as sns
10
11 # pivot table
12 pivot = (pd.pivot_table(dataset, index='company_country', columns='user_country',
13                         values='id', aggfunc='count', margins=True)
14           .drop('All')
15           .sort_values('All')
16           .drop('All', axis=1)
17          )
18
19 # stacked bar chart
20 pivot.plot(stacked=True, kind='barh')
21
22 plt.title('Number of Transactions by Company Country and User Origin')
23 plt.xlabel(None)
24 plt.ylabel(None)
25 plt.legend(title='User origin')
26 plt.show()
```

Exercise 6. Three variables.

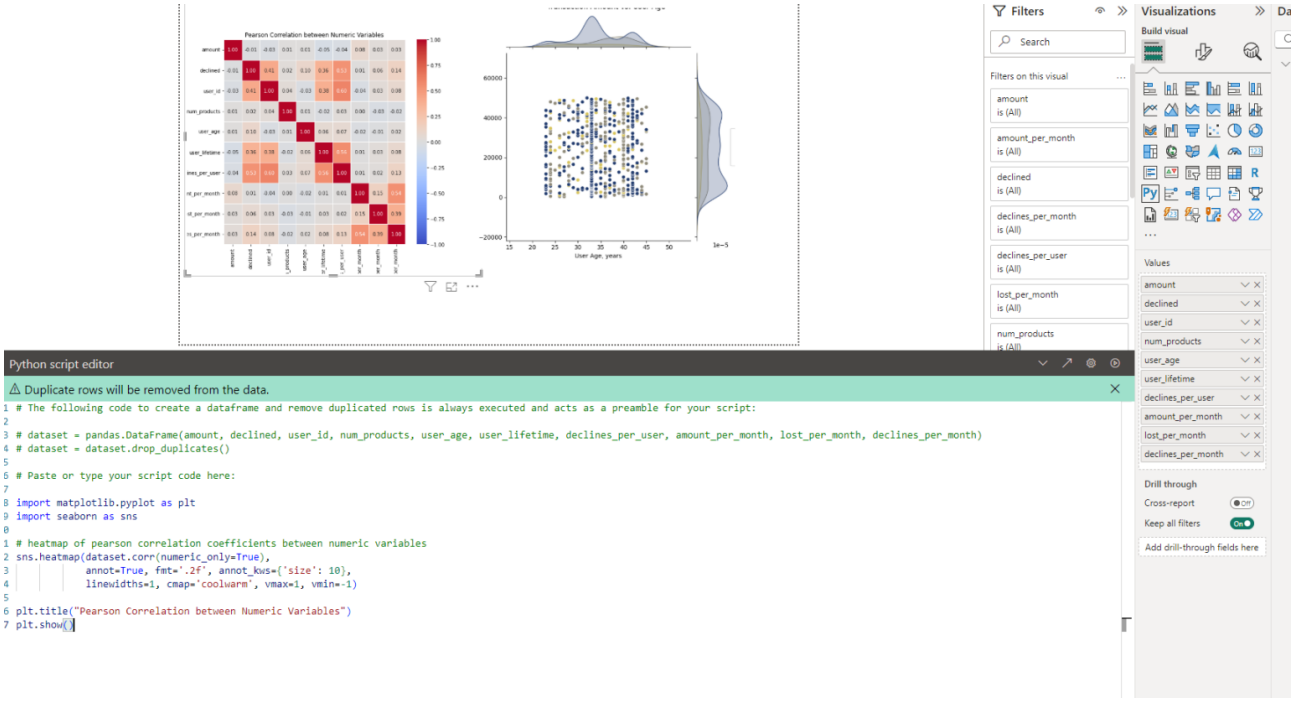



```
# The following code to create a dataframe and remove duplicated rows is always executed and acts as a preamble for your script:

# dataset = pandas.DataFrame(user_age, amount, id, declined)
# dataset = dataset.drop_duplicates()

# Paste or type your script code here:
import matplotlib.pyplot as plt
import seaborn as sns

# scatterplot
sns.scatterplot(data=dataset,
                x='user_age',
                y='amount',
                hue='declined',
                palette=[sns.color_palette('cividis')[0], sns.color_palette('cividis')[5]])

plt.title('Transaction Amount vs. User Age')
plt.xlabel('Age, years')
plt.ylabel('Amount, €')
plt.legend(title='Is declined?', bbox_to_anchor=(1.05, 0.5), loc='center left')
plt.show()
```

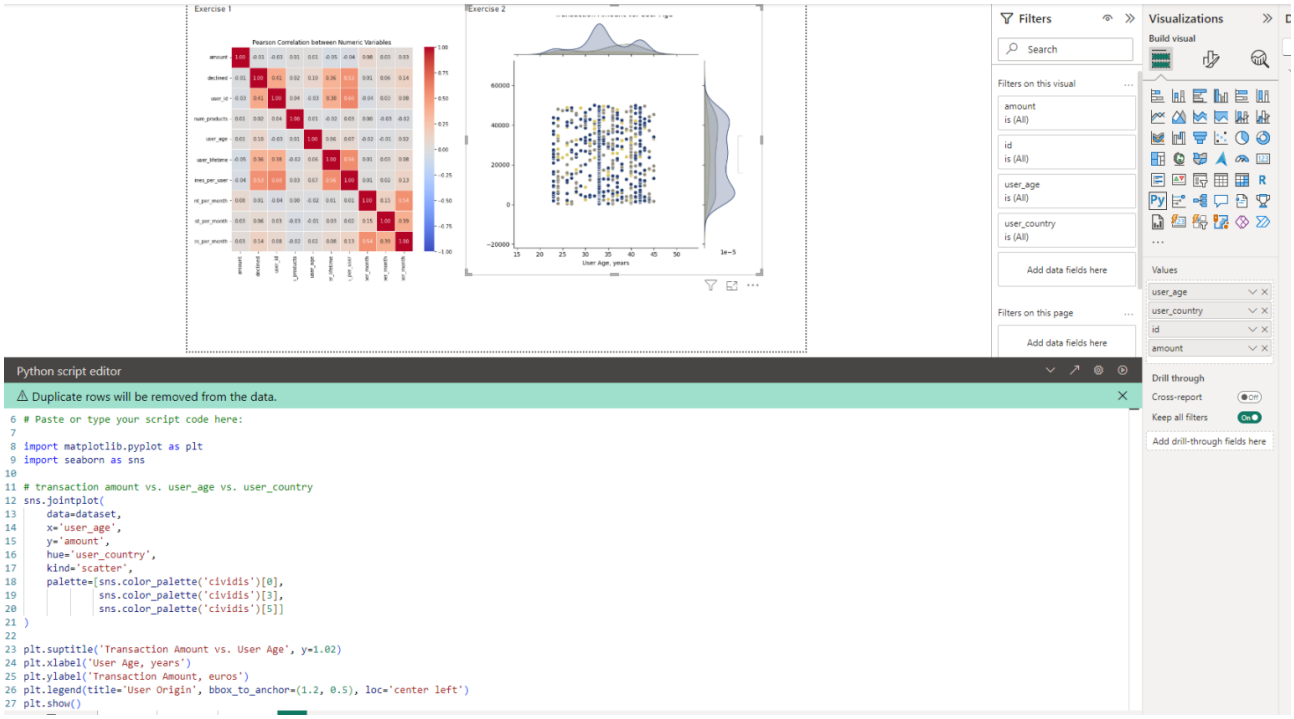
Exercise 7. Pairplot.



Level 2.
Exercise 1. Correlation of all numerical variables.

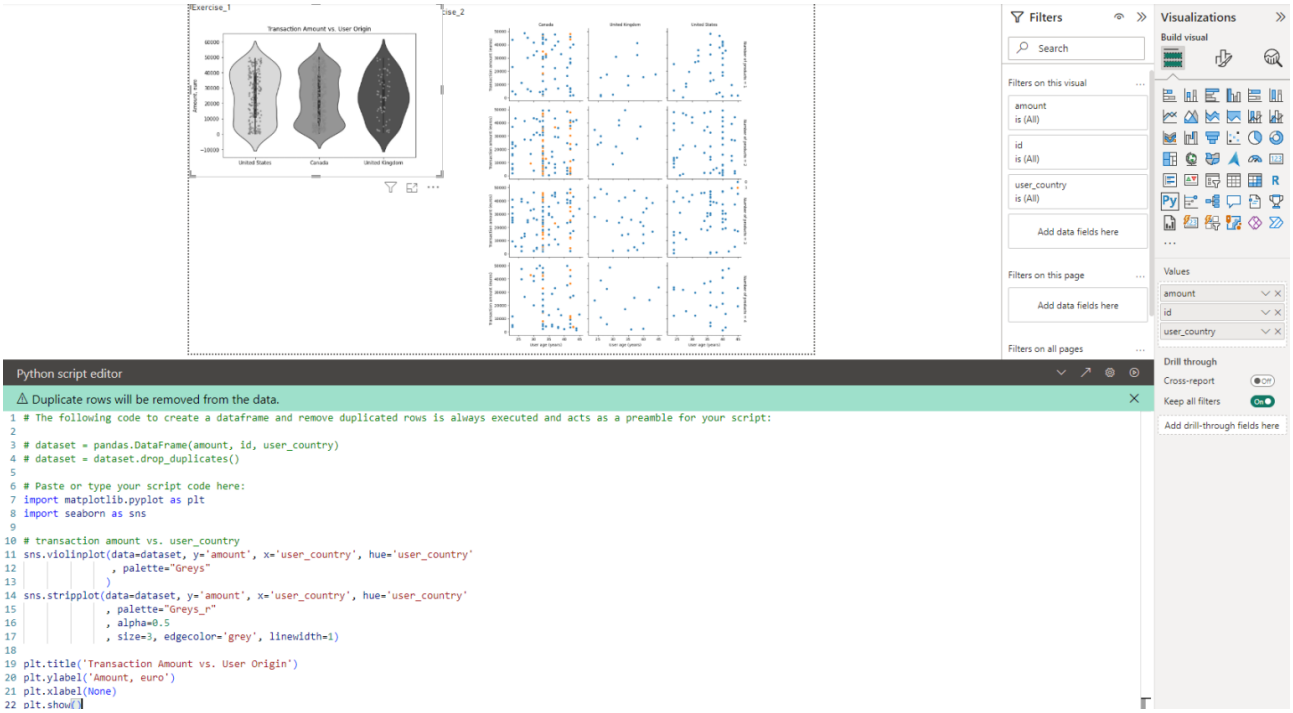


Exercise 2. Jointplot.



Level 3.

Exercise 1. Violinplot combined with another type of graph.



Exercise 2. FacetGrid.