

SPRINT 8.2

POWER BI AMB PYTHON



PREPARED BY:

CECIL LUNA

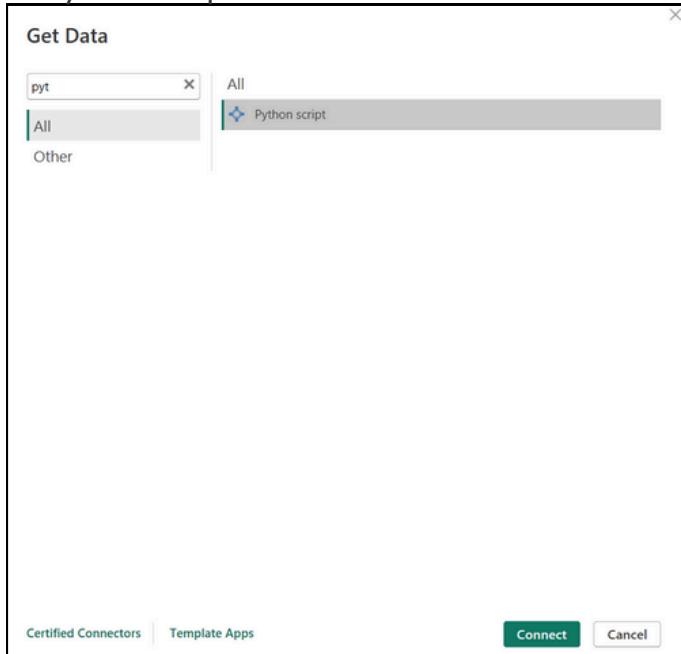
CHECKED AND REVISED BY:

LEVEL 1 PYTHON TO POWER BI CONNECTION

STEPS ON HOW TO CONNECT PYTHON TO POWER BI

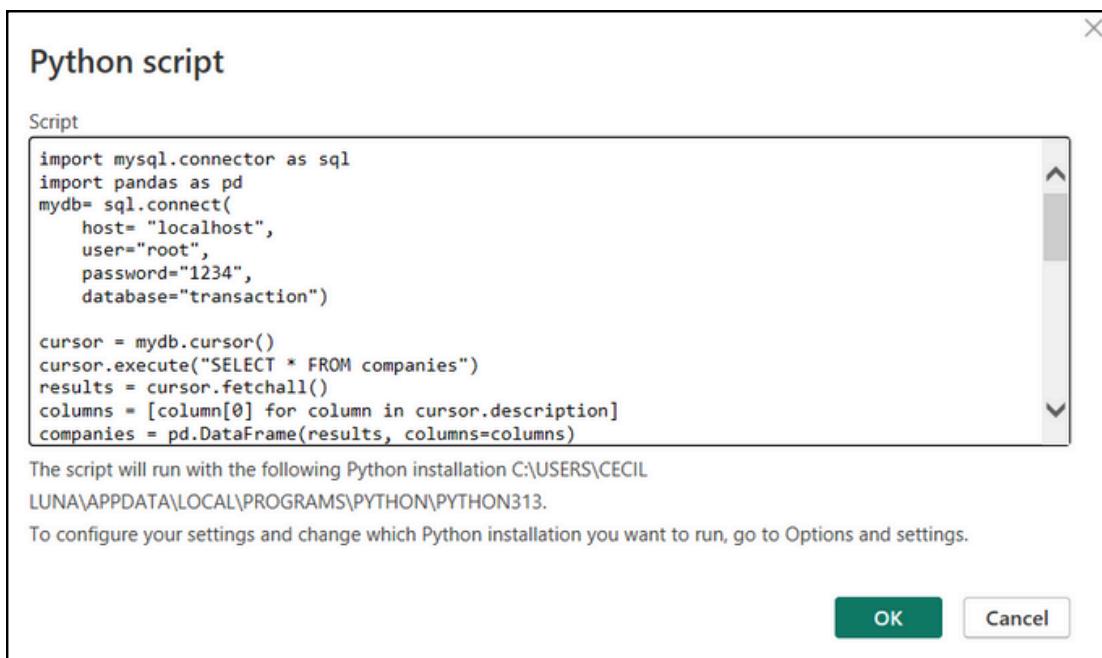
STEP 1:

After creating a new blank report, on the upper toolbox, under home button, click GET DATA. Then search for Python Script.



STEP 2:

A Python Script dialog box will appear then copy the python script from sprint 8.1 to connect the database and fetch the tables then click ok.



STEP 3:

Once connected, select the necessary tables to load.

Navigator

Display Options

Python [5]

- companies
- credit_cards
- products
- transactions
- user

user

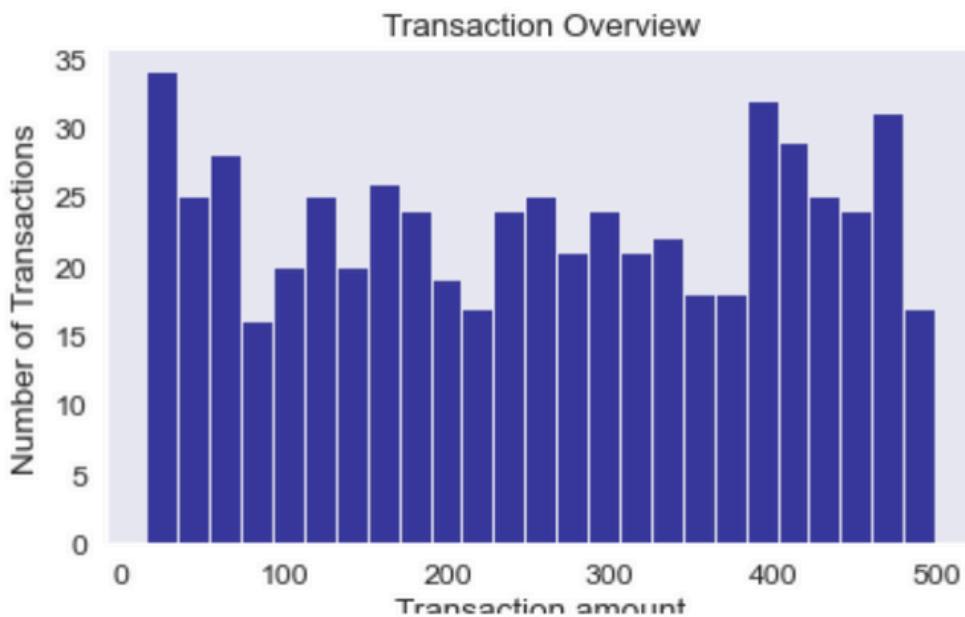
id	name	surname	phone	email
1	Zeus	Gamble	1-282-581-0551	interdum.enim@protonmail.com
2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nibh@protonmail.com
3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.com
4	Howard	Stafford	1-411-740-3269	ornare.egestas@icloud.com
5	Hayfa	Pierce	1-554-541-2077	et.malesuada.fames@hotmail.com
6	Joel	Tyson	(718) 288-8020	gravida.nunc.sed@yahoo.com
7	Rafael	Jimenez	(817) 689-0478	egest@outlook.ca
8	Nissim	Franks	(692) 157-3469	egestas.aliquam.fringilla@outlook.com
9	Mannix	Mcclain	(590) 883-2184	aliquam.nisl@outlook.com
10	Robert	McCarthy	(324) 746-6771	fermentum@protonmail.com
11	Joan	Baird	(981) 429-8106	et@outlook.net
12	Benedict	Wheeler	1-515-824-2855	tincidunt.donec.vitae@outlook.com
13	Allegra	Stanton	1-927-753-6488	proin.egest@protonmail.com
14	Sara	Flynn	1-311-646-9333	integer@outlook.net
15	Noelani	Patrick	1-723-488-5894	sem.magna@google.co.jp
16	Eric	Roth	1-218-549-8253	lorem.sit@yahoo.net
17	Bruce	Gill	(744) 732-8628	metus@aol.net
18	Russell	Jimenez	(657) 779-2438	orci@outlook.edu
19	Nicholas	Travis	1-330-223-9652	libero.du@hotmail.com
20	Kelsey	Bates	(653) 724-4754	ullamcorper.nisl@aol.com
21	Hall	Reeves	(241) 759-9235	erat.egest@hotmail.edu
22	Allistair	Holmes	1-265-323-0812	donec.tempor.est@protonmail.com
23	Kelsie	Bass	1-837-832-5631	consequat@google.ca

LEVEL 1 GRAPHS USING PYTHON

In creating graphs in Power BI using python, it is necessary to import the necessary libraries, in each graph. Unlike Python, in Power BI you need to only tick the columns you want to include your graph and use DATASET as DATAFRAME in the codes.

In merging two tables, you just need to tick the columns in their corresponding tables. NO NEED TO MERGE TABLES USING CODES.

Exercise 1



```
import seaborn as sns
import matplotlib.pyplot as plt

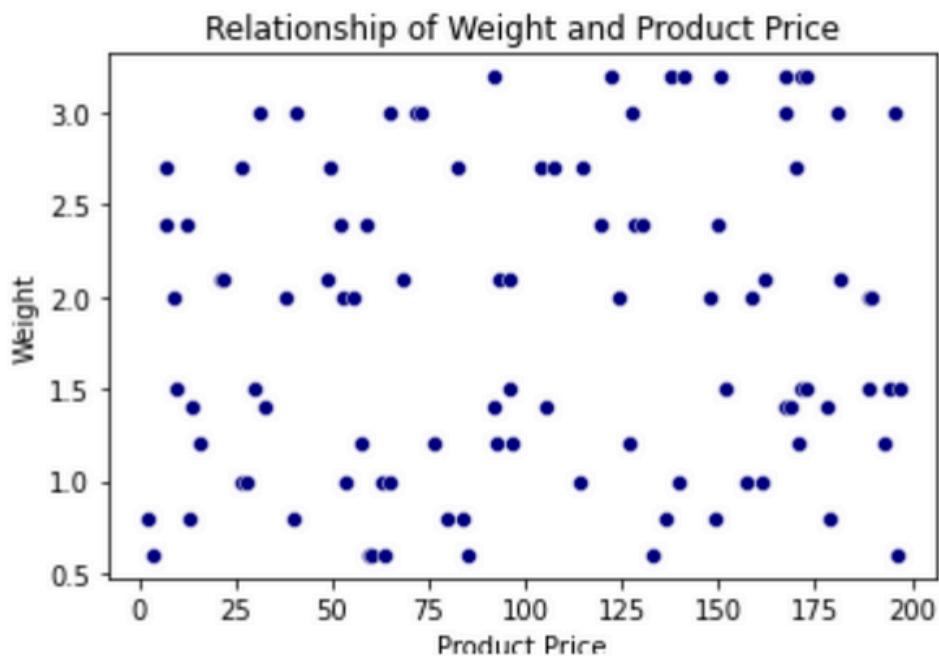
sns.set_style('dark')
sns.set_context('paper', font_scale=1.4)

sns.histplot( data=dataset ['amount'], color='navy', bins = 25)

plt.title('Transaction Overview')
plt.ylabel('Number of Transactions')
plt.xlabel('Transaction amount')

plt.show()
```

Exercise 2

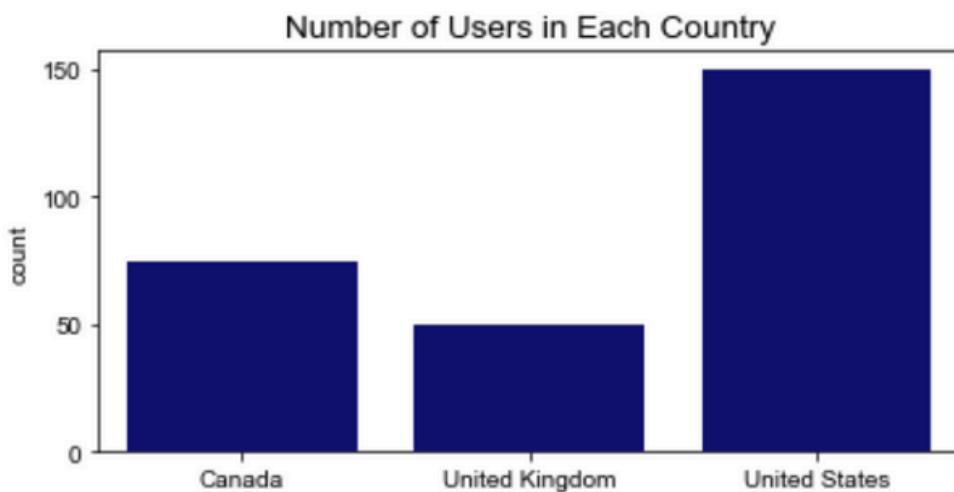


```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

sns.scatterplot(x=dataset['price'], y=dataset['weight'], color= 'navy')

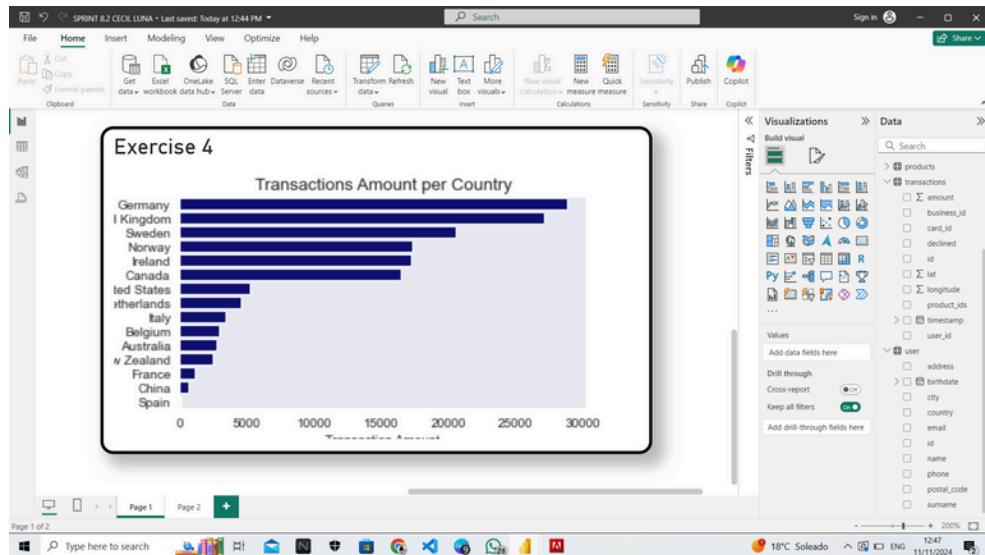
plt.title('Relationship of Weight and Product Price')
plt.xlabel('Product Price')
plt.ylabel('Weight')
plt.show()
```

Exercise 3



```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

sns.countplot(dataset, x='country', color='navy')
sns.set_style('dark')
sns.set_context('paper', font_scale=1.4)
plt.title('Number of Users in Each Country')
plt.show()
```



```

import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

sns.set_style('dark')

# Merge dataset with itself on business_id and company_id

# Group by 'country' and sum the 'amount', then reset the index to convert to a DataFrame
country_sales = dataset.groupby('country')['amount'].sum().sort_values(ascending=False).reset_index()

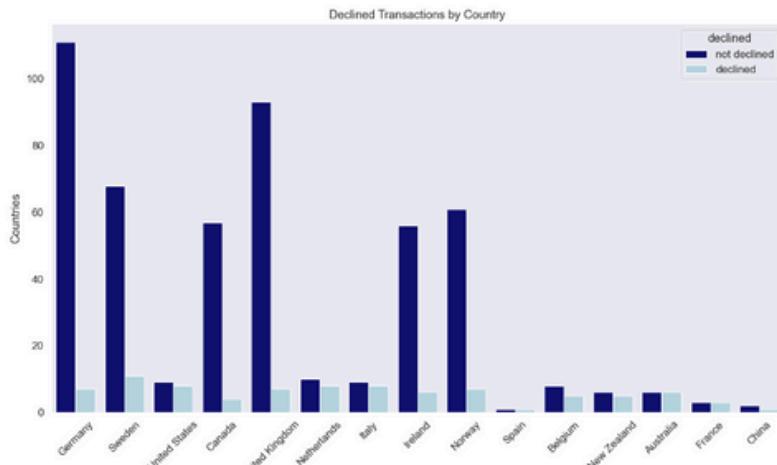
# Create the bar plot with the aggregated data
sns.barplot(data=country_sales, y='country', x='amount', color='navy')
sns.set_style('dark')
sns.set_context('paper', font_scale=1.4)

# Add title and labels
plt.title('Transactions Amount per Country')
plt.ylabel('Countries')
plt.xlabel('Transaction Amount')

# Display plot
plt.show()

```

Exercise 5



```
import seaborn as sns
import matplotlib.pyplot as plt

#This sets the background of the graph
sns.set_style('dark')

# This sets the font style and font size
sns.set_context('paper', font_scale=1.4)

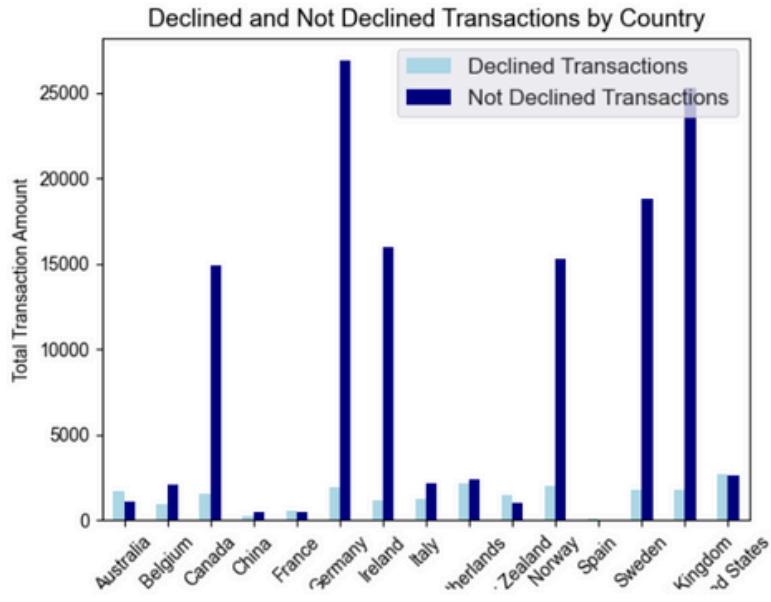
#This sets the size of the graph
plt.figure(figsize=(15, 8))
color=['navy', 'lightblue']

# This is a barplot using seaborn.
# The X-AXIS= declined_by_country.values refers to the number of declined transactions in each country.
# The Y-AXIS= declined_by_country.index refers to the names of the countries.
sns.countplot(dataset, x='country', hue= 'declined', palette= color)

# These are used to customize the the graph
plt.xlabel('Number of Declined Transactions')
plt.ylabel('Countries')
plt.title('Declined Transactions by Country')
plt.xticks(rotation=45)

# This line is used to display the graph.
plt.show()
```

Exercise 6



```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

declined_transactions = dataset[dataset['declined'] == 'declined'].groupby('country')['amount'].sum()

not_declined_transactions = dataset[dataset['declined'] == 'not declined'].groupby('country')['amount'].sum()

transaction_counts = pandas.DataFrame({
    'Declined Transactions': declined_transactions,
    'Not Declined Transactions': not_declined_transactions
}).fillna(0)

# The plot function creates a bar plot with of the new dataframe: transaction_counts, with a specified size and color
transaction_counts.plot(kind='bar', color=['lightblue', 'navy'])

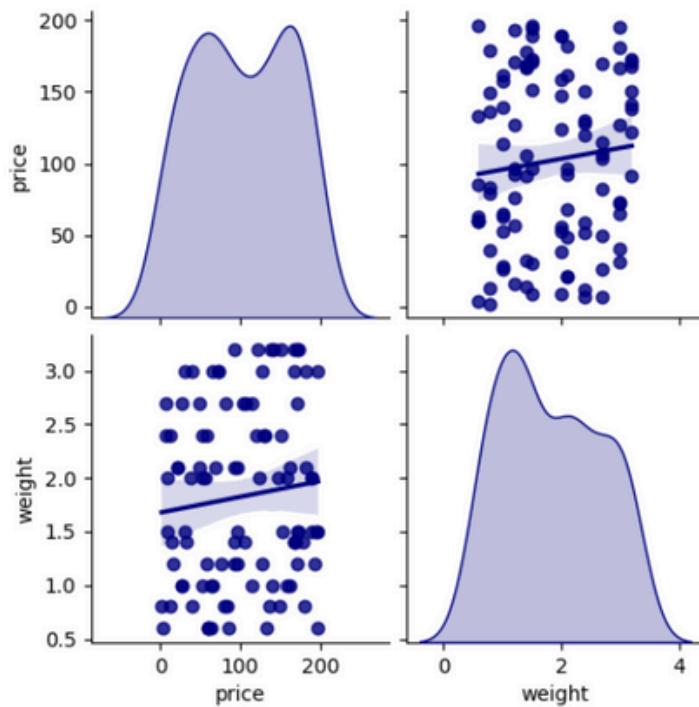
# Set overall style of the plot to a darker background
sns.set_style('dark')

# Set font style and font size
sns.set_context('paper', font_scale=1.4)

# These are used to label the bar graph.
plt.xlabel('Country')
plt.ylabel('Total Transaction Amount')
plt.title('Declined and Not Declined Transactions by Country')
plt.xticks(rotation=45)
plt.legend()

# This line is used to display the graph.
plt.show()
```

Exercise 7



```
import seaborn as sns
import matplotlib.pyplot as plt

sns.pairplot(dataset, vars=['price', 'weight'], diag_kind='kde', kind='reg', plot_kws={'color': 'navy'}, diag_kws={'color': 'navy'})

plt.show()
```