Problem 1

Question 4)

When considering the descriptive statistics, we cannot get an idea of the pattern of datasets and range of data sets, but when using a plot to display data sets, you can detect the shape of the range of data sets and pattern. What are the min and max values in the data set that can be detected easily? graphical interfaces are more useful when you are analysing large data sets (Python file has attached)

Problem 2

Question 1)

I created two scatters between “Median” , "Unemplyed,” and "Median," "Unemplyment_rate” to analyse this question. Considering these 173 sets of data with two scatter plots,the answer is "yes. Higher majors with higher median earnings have a very lower chance of unemployment. (Python file has attached.)

Question 2)

Yes according to bar plot, its true (Python file has attached)

BusinessCcw.py | UserReviews.py | Prob4.py | testFile.py | user_reviews.csv | recentGrads.py | recent-grads.

```python
import pandas as pd  # import the pandas library
import matplotlib.pyplot as plt# import matplotlib.pyplot

df = pd.read_csv("anscombe.csv")  # import/load the anscombe.csv file

df1 = df[["x1", "y1"]]  # four data frames (df1, df2, df3, df4) to
df2 = df[["x2", "y2"]]  # store the four data sets in anscombe.csv file
df3 = df[["x3", "y3"]]
df4 = df[["x4", "y4"]]

df1dEscrptive_statistics = df1.describe()  # basic descriptive statistics
df2dEscrptive_statistics = df2.describe()  # of df1,df2,df3 and df4
df3dEscrptive_statistics = df3.describe()
df4dEscrptive_statistics = df4.describe()

print("\n",df1dEscrptive_statistics)  # display above those created basic descriptive statistics
print("\n",df2dEscrptive_statistics)
print("\n",df3dEscrptive_statistics)
print("\n",df4dEscrptive_statistics)

df1.plot(kind='scatter',x='x1',y='y1')# scatter plot for each data set(df1,df2,df3,df4)
df2.plot(kind='scatter',x='x2',y='y2')
df3.plot(kind='scatter',x='x3',y='y3')
df4.plot(kind='scatter',x='x4',y='y4')

plt.show()
```

BusinessCcw.py | UserReviews.py | Prob4.py | testFile.py | user_reviews.csv | recentGrads.py | recent-grads.

```python
import pandas as pd  # import the pandas library
import matplotlib.pyplot as plt # import matplotlib.pyplot
df=pd.read_csv("recent-grads.csv")

df.plot(kind='scatter',x='Median',y='Unemployed') # two scatter plots to analyse Median and Unemployment
df.plot(kind='scatter',x='Median',y='Unemployment_rate')
plt.show()

Median_above_60000 = df[df["Median"] > 60000] # bar plot for compare earnings between majors
Median_above_60000.set_index("Major")[["P25th", "Median", "P75th"]].plot(kind="bar", figsize=(12,8))
plt.title("",rotation=0)
plt.ylabel("Earnings")
plt.tight_layout()
plt.show()

```

BusinessCcw.py | UserReviews.py | Prob4.py | testFile.py | user_reviews.csv | recentGrads.py | recent-grads.

```python
import pandas as pd

df = pd.read_csv("user_reviews.csv")
df['len_txt'] = df['text'].apply(len)  # add a len_txt column to the dataframe and content
print(df.head())  # display all heads in the df for checking above adding new column is success or not


1 usage
def super_Category(value):
    while True:
        if value['len_txt'] > 1000 or value['grade'] >= 9:  # function for checking those conditions and create the
            # column
            return 'expert reviewer'
        elif value['grade'] <= 1 and value['len_txt'] > 1000:
            return 'opposed reviewer'
        else:
            return 'neutral reviewer'


df['super category'] = df.apply(super_Category, axis=1)  # add column to the df, and content
print(df.head())  # display all columns in df

```

BusinessCcw.py | UserReviews.py | Prob4.py | testFile.py | user_reviews.csv | recentGrads.py | recent-grads.

```python
import sqlite3

connect = sqlite3.connect("db")
cursor = connect.cursor()
# create table ,locations
cursor.execute('''
CREATE TABLE IF NOT EXISTS locations (
location_id INT(3),
street_address VARCHAR(50) unique not null,
postal_code VARCHAR(10) not null,
city VARCHAR(50) not null,
state_province VARCHAR(50) not null, country
VARCHAR(50) not null,
constraint l_lid_pk PRIMARY KEY (location_id)
);
''')
# create table,departments
cursor.execute('''
CREATE TABLE IF NOT EXISTS departments (
department_id INT(4),
department_name VARCHAR(20) unique not null,
location_id INT(3),
constraint d_did_pk PRIMARY KEY (department_id),
constraint d_lid_fk FOREIGN KEY (location_id) references locations(location_id)
);
''')
# enter data to locations
```

38:91 CRLF UTF-8 4 spaces Python 3.11

BusinessCcw.py | UserReviews.py | Prob4.py | testFile.py | user_reviews.csv | recentGrads.py | recent-grads.

```python
cursor.executemany(
    '''INSERT INTO
locations (location_id, street_address, postal_code, city, state_province, country)
VALUES(?,?,?,?,?,?)''', [
        (100, '2 Nice Road', 'N2 7TH', 'London', 'Greater London', 'UK'),
        (200, '23 Pretty Road', 'BS1 8FD', 'Bristol', 'Bristol County', 'UK'),
        (300, '26 Great Street', 'BN1 4BF', 'Brigthon', 'Sussex', 'UK'),
        (400, '143 Lovely Road', 'CB1 2NV', 'Cambridge', 'Cambridgeshire', 'UK')])
# enter data to departments
cursor.executemany(
    '''INSERT INTO departments (department_id, department_name, location_id) VALUES(?,?,?)''', [
        (10, 'IT', 100),
        (20, 'Operations', 200),
        (30, 'Sales', 300),
        (40, 'Marketing', 200),
        (50, 'Management', None)])
cursor.execute('''
SELECT d.department_id, d.department_name, l.location_id, l.city
FROM departments d
LEFT JOIN locations l ON d.location_id = l.location_id;
''')
print("All departments and their locations:")
for row in cursor.fetchall():
    print(row)


cursor.execute('''
```