3)Design and develop at least 5 problem statements which demonstrate the use of Control Structures of any data analytics tool.

In [10]:
```
pip install statsmodels
```

```
Requirement already satisfied: statsmodels in c:\users\mayuri\anaconda3\lib\site-pack
ages (0.14.0)
Requirement already satisfied: scipy!=1.9.2,>=1.4 in c:\users\mayuri\anaconda3\lib\si
te-packages (from statsmodels) (1.11.3)
Requirement already satisfied: numpy>=1.18 in c:\users\mayuri\anaconda3\lib\site-pack
ages (from statsmodels) (1.26.1)
Requirement already satisfied: patsy>=0.5.2 in c:\users\mayuri\anaconda3\lib\site-pac
kages (from statsmodels) (0.5.3)
Requirement already satisfied: pandas>=1.0 in c:\users\mayuri\anaconda3\lib\site-pack
ages (from statsmodels) (2.1.1)
Requirement already satisfied: packaging>=21.3 in c:\users\mayuri\anaconda3\lib\site-
packages (from statsmodels) (23.0)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\mayuri\anaconda3\li
b\site-packages (from pandas>=1.0->statsmodels) (2.8.2)
Requirement already satisfied: tzdata>=2022.1 in c:\users\mayuri\anaconda3\lib\site-p
ackages (from pandas>=1.0->statsmodels) (2023.3)
Requirement already satisfied: pytz>=2020.1 in c:\users\mayuri\anaconda3\lib\site-pac
kages (from pandas>=1.0->statsmodels) (2022.7)
Requirement already satisfied: six in c:\users\mayuri\anaconda3\lib\site-packages (fr
om patsy>=0.5.2->statsmodels) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [5]:
```
import pandas as pd
from statsmodels.tsa.arima_model import ARIMA
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
```

In [8]:
```
# Problem Statement 1: Data Filtering and Summation
# Load the dataset
data = pd.read_csv('sales_data.csv')
data
```

Out[8]:

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Country	S
0	2013-11-26	26	November	2013	19	Youth (<25)	M	Canada	B Colu
1	2015-11-26	26	November	2015	19	Youth (<25)	M	Canada	B Colu
2	2014-03-23	23	March	2014	49	Adults (35-64)	M	Australia	New S V
3	2016-03-23	23	March	2016	49	Adults (35-64)	M	Australia	New S V
4	2014-05-15	15	May	2014	47	Adults (35-64)	F	Australia	New S V
...	...	...	...	...	...	...	...	...	
113031	2016-04-12	12	April	2016	41	Adults (35-64)	M	United Kingdom	Eng
113032	2014-04-02	2	April	2014	18	Youth (<25)	M	Australia	Queens
113033	2016-04-02	2	April	2016	18	Youth (<25)	M	Australia	Queens
113034	2014-03-04	4	March	2014	37	Adults (35-64)	F	France	S (
113035	2016-03-04	4	March	2016	37	Adults (35-64)	F	France	S (

113036 rows × 18 columns

In [13]:
```
data.columns
```

Out[13]:
```
Index(['Date', 'Day', 'Month', 'Year', 'Customer_Age', 'Age_Group',
       'Customer_Gender', 'Country', 'State', 'Product_Category',
       'Sub_Category', 'Product', 'Order_Quantity', 'Unit_Cost', 'Unit_Price',
       'Profit', 'Cost', 'Revenue'],
      dtype='object')
```

In [11]:
```
# Filter for a specific product category and time period
filtered_data = data[(data['Product_Category'] == 'Electronics') & (data['Date'] >= '2
```

In [19]:
```
# Calculate the total Revenue for the filtered data
total_Revenue = filtered_data['Revenue'].sum()

print("Total Revenue for Electronics in 2023:", total_Revenue)
```

```
Total Revenue for Electronics in 2023: 0
```

In [23]:
```
# Problem Statement 2: Conditional Aggregation
# Load the dataset of customer reviews
```

```
data = pd.read_csv('customer_data.csv', encoding='unicode_escape')
data
```

Out[23]:

	Review Title	Customer name	Rating	Date	Category	Comments	Useful
0	Another Midrange killer Smartphone by Xiaomi	Rishikumar Thakur	4.0 out of 5 stars	on 1 October 2018	Display	Another Midrange killer Smartphone by Xiaomi\n...	
1	vry small size mobile	Raza ji	3.0 out of 5 stars	on 15 September 2018	Others	All ok but vry small size mobile	7 people found this helpful
2	Full display not working in all application.	Vaibhav Patel	3.0 out of 5 stars	on 18 September 2018	Others	Quite good	7 people found this helpful
3	Value for Money	Amazon Customer	5.0 out of 5 stars	on 28 September 2018	Display	Redmi has always have been the the king of bud...	2 people found this helpful
4	Not worth for the money	Sudhakaran Wadakkancheri	2.0 out of 5 stars	on 18 September 2018	Others	worst product from MI. I am a hardcore fan of ...	6 people found this helpful
...	...	...	...	...	...	...	...
275	Cemera quality,face unlock most important in t...	Rahul	5.0 out of 5 stars	on 19 September 2018	Others	I like This Phone, Awesome look and design.\nI...	NaN
276	Mi is best phone	Sunil Soni	4.0 out of 5 stars	on 18 September 2018	Others	Product is avasome but invoice is note include...	NaN
277	Its a OK Phone	D.C.Padhi	3.0 out of 5 stars	on 15 September 2018	Battery	Redmi Note4, Note5, now 6pro..It seems the old...	NaN
278	Redmi	Mahesh	5.0 out of 5 stars	on 21 September 2018	Others	I love mi	NaN
279	Not worth for the price.	Vinod	1.0 out of 5 stars	on 17 September 2018	Camera	Same old configurations with higher price.\nNo...	NaN

280 rows × 7 columns

In [24]:
```
data.columns
```

Out[24]:
```
Index(['Review Title', 'Customer name', 'Rating', 'Date', 'Category',
       'Comments', 'Useful'],
      dtype='object')
```

In [30]:
```
# Calculate the average rating for positive and negative reviews
positive_reviews = data[data['Review Title'] == 'positive']
negative_reviews = data[data['Review Title'] == 'negative']
```

In [27]:
```
average_positive_rating = positive_reviews['Rating'].mean()
average_negative_rating = negative_reviews['Rating'].mean()

print("Average Rating for Positive Reviews:", average_positive_rating)
print("Average Rating for Negative Reviews:", average_negative_rating)
```

```
Average Rating for Positive Reviews: nan
Average Rating for Negative Reviews: nan
```

In [31]:
```
# Problem Statement 3: Data Transformation
# Load the dataset of temperature readings in Celsius
data = pd.read_csv('temperature.csv')
```

In [32]:
```
# Convert Celsius to Fahrenheit
data['temperature_F'] = (data['temperature_C'] * 9/5) + 32
```

In [33]:
```
# Identify days with temperatures exceeding a threshold
threshold = 90
hot_days = data[data['temperature_F'] > threshold]
print("Days with Temperature Exceeding 90°F:", hot_days)
```

```
Days with Temperature Exceeding 90°F:       temperature_C  temperature_F
15              32.9          91.22
16              32.7          90.86
37              33.0          91.40
38              32.5          90.50
39              32.7          90.86
...              ...            ...
1754            32.3          90.14
1755            32.3          90.14
1758            33.3          91.94
1759            32.3          90.14
1761            32.4          90.32

[275 rows x 2 columns]
```

In [39]:
```
# Problem Statement 4: Time-Series Analysis and Forecasting
# Load historical stock price data
data = pd.read_csv('stock_price.csv')
data
```

Out[39]:

	Date	Price
0	14-08-2018	23.020000
1	15-08-2018	23.150000
2	16-08-2018	23.500000
3	17-08-2018	23.400000
4	20-08-2018	23.549999
...	...	...
247	08-08-2019	22.600000
248	09-08-2019	22.450001
249	12-08-2019	22.219999
250	13-08-2019	21.850000
251	14-08-2019	22.110001

252 rows × 2 columns

In [40]:
```
data.columns
```

Out[40]:
```
Index(['Date', 'Price'], dtype='object')
```

In [42]:
```
import statsmodels.api as sm
```

In [49]:
```
# Analyze trends and patterns (not shown here)
# ...

# Forecast stock prices for the next week using ARIMA
model = sm.tsa.arima.ARIMA(data['Price'], order=(5,1,0))
model_fit = model.fit()
forecast = model_fit.forecast(steps=7)
print("Stock Price Forecast for the Next Week:", forecast)
```

```
Stock Price Forecast for the Next Week: 252    22.058556
253    22.069017
254    22.072505
255    22.105591
256    22.084573
257    22.090281
258    22.089011
Name: predicted_mean, dtype: float64
```

In [68]:
```
#Problem Statement 5: iris data Segmentation
# Load iris data
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris
```

In [73]:
```
# Load the Iris dataset
data = pd.read_csv('iris.csv')
data
```

Out[73]:

	Unnamed: 0	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
0	1	5.1	3.5	1.4	0.2	setosa
1	2	4.9	3.0	1.4	0.2	setosa
2	3	4.7	3.2	1.3	0.2	setosa
3	4	4.6	3.1	1.5	0.2	setosa
4	5	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	virginica
146	147	6.3	2.5	5.0	1.9	virginica
147	148	6.5	3.0	5.2	2.0	virginica
148	149	6.2	3.4	5.4	2.3	virginica
149	150	5.9	3.0	5.1	1.8	virginica

150 rows × 6 columns

In [79]:
```
X = pd.DataFrame(data.data, columns=data.feature_names)
Y = pd.Series(data.target, name="species")

# Combine X and Y into a single DataFrame
iris_df = pd.concat([X, Y], axis=1)

# View the first few rows of the DataFrame
print(iris_df.head())
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[79], line 1
----> 1 X = pd.DataFrame(data.data, columns=data.feature_names)
      2 Y = pd.Series(data.target, name="species")
      4 # Combine X and Y into a single DataFrame

File ~\anaconda3\lib\site-packages\pandas\core\generic.py:6204, in NDFrame.__getattr_
_(self, name)
   6197 if (
   6198     name not in self._internal_names_set
   6199     and name not in self._metadata
   6200     and name not in self._accessors
   6201     and self._info_axis._can_hold_identifiers_and_holds_name(name)
   6202 ):
   6203     return self[name]
-> 6204 return object.__getattribute__(self, name)

AttributeError: 'DataFrame' object has no attribute 'data'
```

In [ ]:
```
# Determine the optimal number of clusters (K) using the Elbow method
inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(X)
    inertia.append(kmeans.inertia_)
```

In [ ]:
```
# Plot the Elbow curve
plt.plot(range(1, 11), inertia, marker='o', linestyle='--')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('Inertia')
plt.title('Elbow Method for Optimal K')
plt.grid()
plt.show()
```

In [ ]:
```

```