In [1]:

```
#QUERY 1
import mysql.connector
import pandas as pd
import matplotlib.pyplot as plt

host = "localhost"
port = 3305
user = "root"
password = "root"
database = "dma_project"

conn = mysql.connector.connect(host=host, port=port, user=user, password=password, database=database)
cursor = conn.cursor()

def execute_query(query):
    cursor.execute(query)
    data = cursor.fetchall()
    return data

query = "SELECT * from organization"
data = execute_query(query)
columns = [desc[0] for desc in cursor.description]
df_organization = pd.DataFrame(data, columns=columns)
print("QUERY 1- Retrive the organization table:")

cursor.close()
conn.close()
df_organization
```

```
QUERY 1- Retrive the organization table:
```

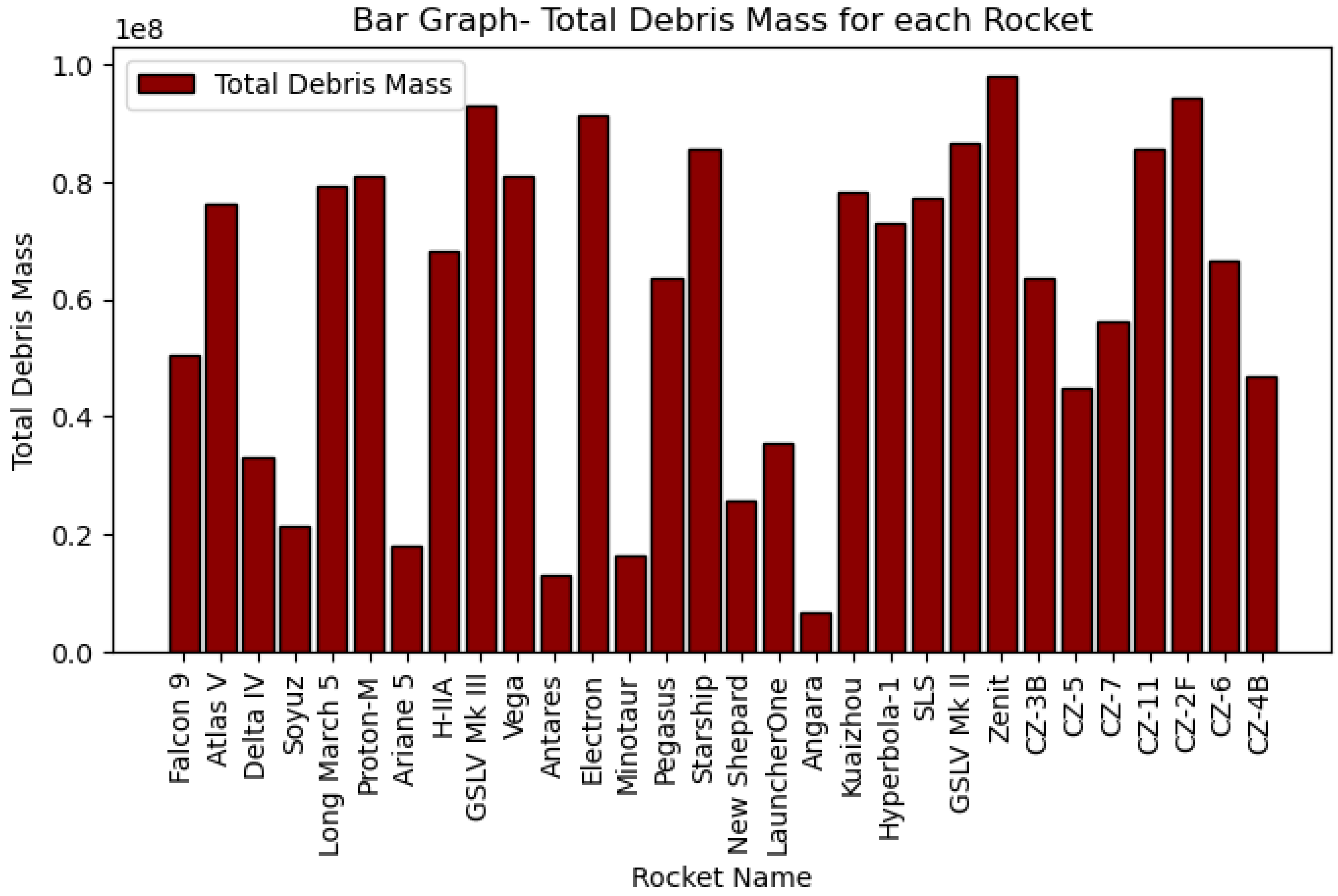
Out[1]:

	ORGANIZATION_ID	O_NAME	O_LOCATION	O_CONTACT
0	SP961	Stellar Systems	Tokyo	309-645-1256
1	SP962	Galactic Enterprises	Mumbai	611-532-0829
2	SP963	Cosmic Ventures	Mumbai	852-367-0844
3	SP964	Celestial Innovations	Dubai	334-357-3610
4	SP965	Nebula Corporation	New York	737-123-5285
5	SP966	AstroTech	Dubai	842-404-0867
6	SP967	Starbound Solutions	Rio de Janeiro	389-308-4833
7	SP968	Lunar Enterprises	Rio de Janeiro	113-739-7324
8	SP969	Solar Nexus	Mumbai	408-146-8508
9	SP970	Interstellar Holdings	Mumbai	780-691-8487
10	SP971	Orion Industries	London	396-604-4017
11	SP972	Nova Enterprises	Tokyo	245-250-2192
12	SP973	Cosmos Corporation	New York	817-750-7309
13	SP974	Astro Dynamics	Dubai	764-143-7956
14	SP975	Galaxy Group	Tokyo	849-863-2692
15	SP976	Comet Enterprises	Dubai	172-120-1968
16	SP977	Saturn Solutions	Tokyo	786-965-4374
17	SP978	Andromeda Innovations	New York	202-433-5356
18	SP979	Milky Way Enterprises	Tokyo	744-497-0441
19	SP980	Solaris Corporation	Toronto	985-218-4721
20	SP981	Supernova Systems	Sydney	626-404-1661
21	SP982	Cosmic Connections	Paris	990-541-7190
22	SP983	Aurora Enterprises	Rio de Janeiro	873-238-8696
23	SP984	Pulsar Innovations	Tokyo	139-394-1417
24	SP985	Nebula Nexus	Sydney	395-177-3133
25	SP986	AstroTech Solutions	Dubai	202-363-5941
26	SP987	Starlight Holdings	London	449-547-9344
27	SP988	Celestial Innovations	Tokyo	689-426-3619
28	SP989	Galactic Nexus	Berlin	850-139-7035
29	SP990	Stellar Solutions	Berlin	805-556-9241

In [2]:

```
#QUERY 2
import mysql.connector
import pandas as pd
import matplotlib.pyplot as plt

host = "localhost"
port = 3305
user = "root"
password = "root"
database = "dma_project"

conn = mysql.connector.connect(host=host, port=port, user=user, password=password, database=database)
cursor = conn.cursor()

query = """
SELECT R.ROCKET_ID, R.R_NAME, SD.DEBRIS_ID, SUM(SD.D_MASS) AS TOTAL_D_MASS
FROM ROCKET R, SPACE_DEBRIS SD, PRESENT_IN P
WHERE R.ROCKET_ID = P.ROCKET_ID
AND P.DEBRIS_ID = SD.DEBRIS_ID
GROUP BY R.ROCKET_ID, R.R_NAME, SD.DEBRIS_ID;
"""
cursor.execute(query)
data = cursor.fetchall()
columns = [desc[0] for desc in cursor.description]
df = pd.DataFrame(data, columns=columns)
cursor.close()
conn.close()
print("QUERY 2- Retrieve the total mass of debris for each rocket along with their respective rocket IDs, names, and de

plt.figure(figsize=(8, 4))
plt.bar(df['R_NAME'], df['TOTAL_D_MASS'], color='darkred', edgecolor='black', label='Total Debris Mass')
plt.xlabel('Rocket Name')
plt.ylabel('Total Debris Mass')
plt.title('Bar Graph- Total Debris Mass for each Rocket')
plt.legend()
plt.xticks(rotation=90)
plt.show()
df.head(8)
```

```
QUERY 2- Retrieve the total mass of debris for each rocket along with their respective rocket IDs, names, and debris
IDs:
```



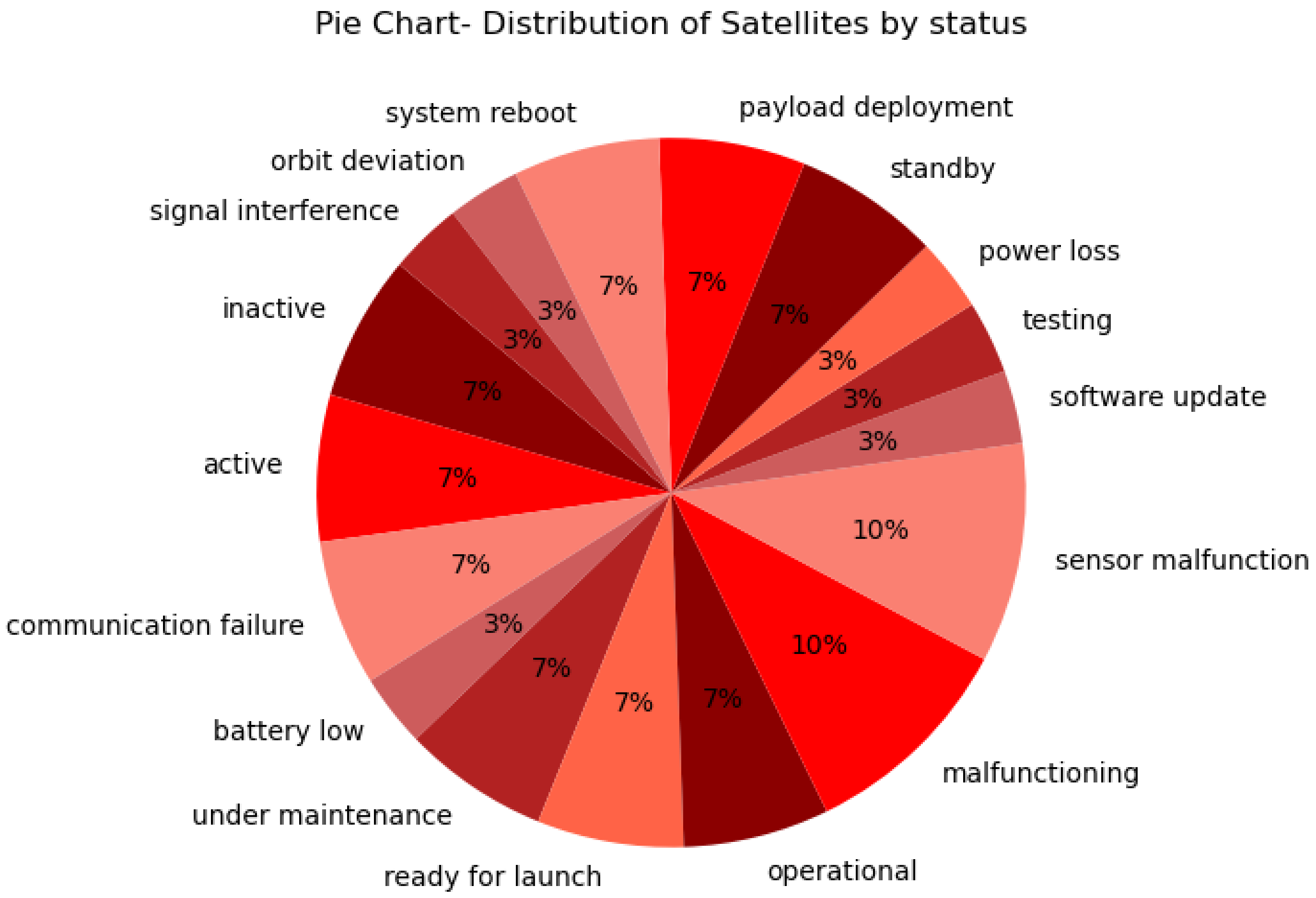
Out[2]:

	ROCKET_ID	R_NAME	DEBRIS_ID	TOTAL_D_MASS
0	1	Falcon 9	100	50627782.50
1	2	Atlas V	101	76311808.23
2	3	Delta IV	102	32977925.00
3	4	Soyuz	103	21346950.03
4	5	Long March 5	104	79321433.33
5	6	Proton-M	105	80919564.80
6	7	Ariane 5	106	17986562.98
7	8	H-IIA	107	68251580.56

In [3]:

```
#QUERY 3
import mysql.connector
import pandas as pd
import matplotlib.pyplot as plt

host = "localhost"
port = 3305
user = "root"
password = "root"
database = "dma_project"

conn = mysql.connector.connect(host=host, port=port, user=user, password=password, database=database)
cursor = conn.cursor()

query = """
SELECT S.S_STATUS, COUNT(S.SATELLITE_ID) AS SATELLITE_COUNT
FROM SATELLITE S
GROUP BY S.S_STATUS;
"""
cursor.execute(query)
data = cursor.fetchall()
columns = [desc[0] for desc in cursor.description]
df = pd.DataFrame(data, columns=columns)
cursor.close()
conn.close()
print("QUERY 3- Retrieve the distribution of satellites based on their operational status:")

plt.figure(figsize=(6, 6))
plt.pie(df['SATELLITE_COUNT'], labels=df['S_STATUS'], autopct='%1.0f%%', startangle=140, colors=['darkred', 'red', 'sa
plt.title('Pie Chart- Distribution of Satellites by status')
plt.show()
df.tail(10)
```

```
QUERY 3- Retrieve the distribution of satellites based on their operational status:
```



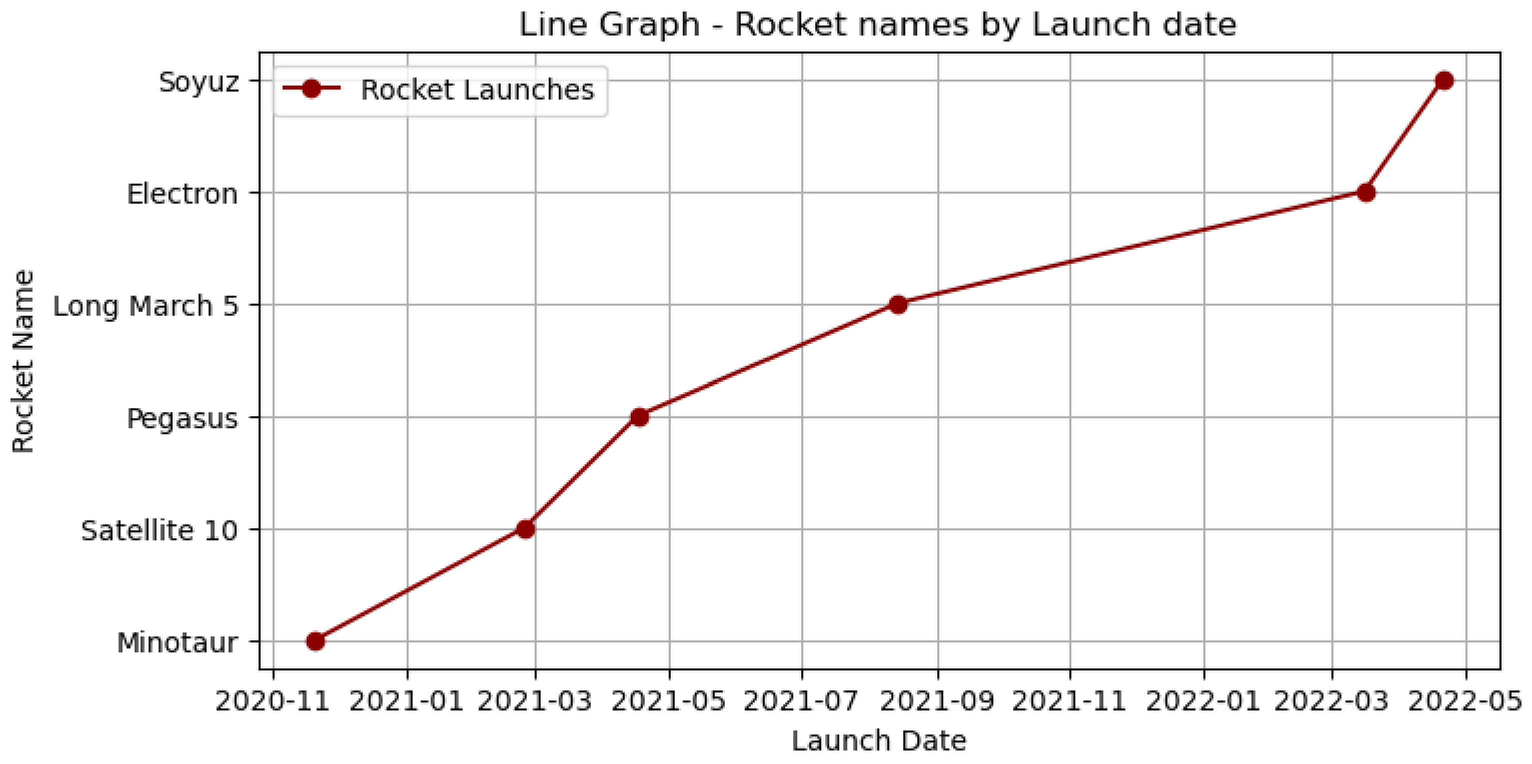
Out[3]:

	S_STATUS	SATELLITE_COUNT
7	malfunctioning	3
8	sensor malfunction	3
9	software update	1
10	testing	1
11	power loss	1
12	standby	2
13	payload deployment	2
14	system reboot	2
15	orbit deviation	1
16	signal interference	1

In [1]:

```
#QUERY 4
import mysql.connector
import pandas as pd
import matplotlib.pyplot as plt

host = "localhost"
port = 3305
user = "root"
password = "root"
database = "dma_project"

conn = mysql.connector.connect(host=host, port=port, user=user, password=password, database=database)
cursor = conn.cursor()

query = """
    SELECT R_NAME AS NEW_NAME, R_LAUNCH_DATE AS NEW_LAUNCH_DATE
    FROM ROCKET
    WHERE R_NAME IN ('Long March 5', 'Proton M', 'Soyuz', 'Electron','Minotaur', 'Pegasus')
    UNION
    SELECT S_NAME, S_LAUNCHED_DATE
    FROM SATELLITE
    WHERE SATELLITE_ID IN (
        SELECT DISTINCT S.SATELLITE_ID
        FROM SATELLITE S
        INNER JOIN LAUNCH_FACILITY F ON S.S_LAUNCHED_DATE = F.F_LAUNCHED_DATE
        WHERE F.F_LOCATION IN ('France', 'Turkey'))
    ORDER BY NEW_LAUNCH_DATE;
"""
cursor.execute(query)
data = cursor.fetchall()
cursor.close()
conn.close()
print("QUERY 4- Retrieve the names and launch dates of rockets along with the names and launch dates of satellites:")

df = pd.DataFrame(data, columns=["NEW_NAME", "NEW_LAUNCH_DATE"])
plt.figure(figsize=(8, 4))
plt.plot(df["NEW_LAUNCH_DATE"], df["NEW_NAME"], marker='o', color='darkred', label='Rocket Launches')
plt.xlabel("Launch Date")
plt.ylabel("Rocket Name")
plt.title("Line Graph - Rocket names by Launch date")
plt.legend()
plt.grid(True)
plt.xticks(rotation=0)
plt.show()
df
```

```
QUERY 4- Retrieve the names and launch dates of rockets along with the names and launch dates of satellites:
```



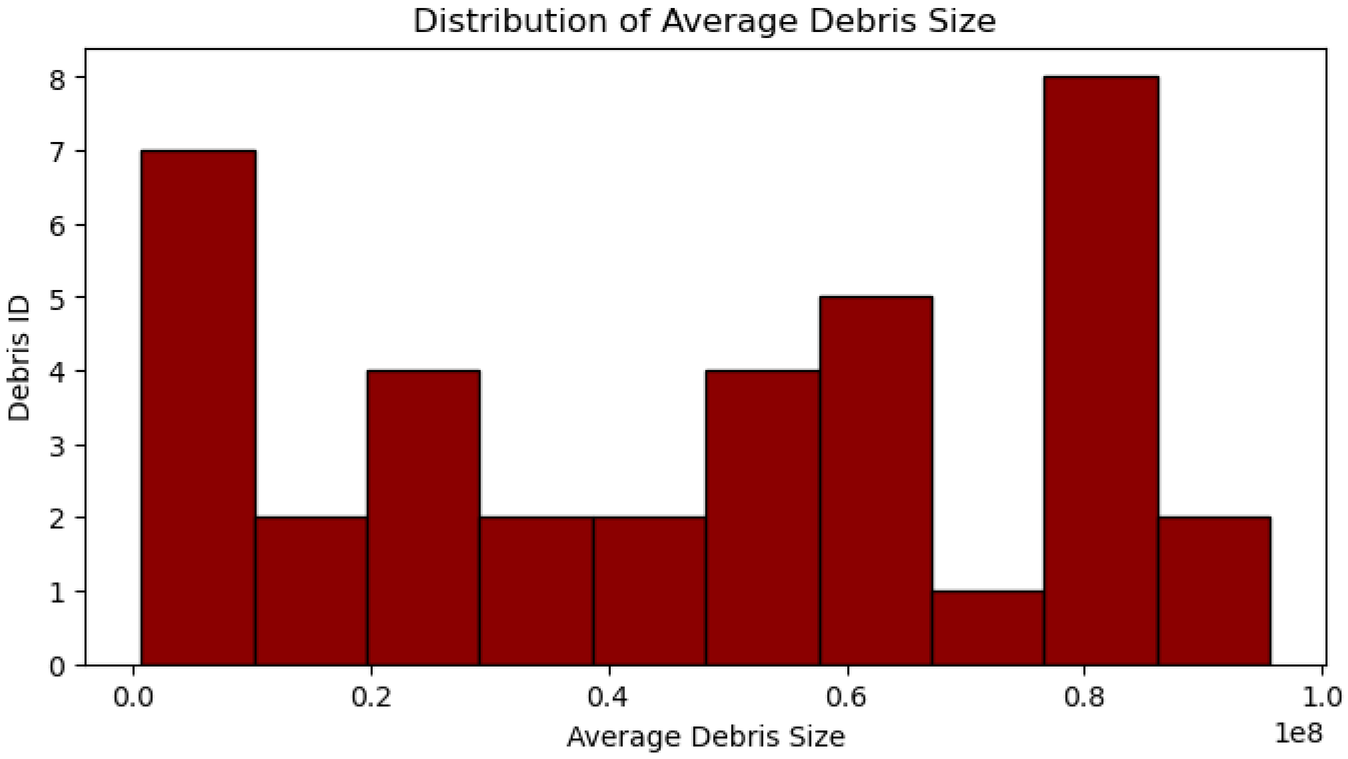
Out[1]:

	NEW_NAME	NEW_LAUNCH_DATE
0	Minotaur	2020-11-20
1	Satellite 10	2021-02-24
2	Pegasus	2021-04-17
3	Long March 5	2021-08-14
4	Electron	2022-03-16
5	Soyuz	2022-04-21

In [23]:

```
#QUERY 5
import mysql.connector
import pandas as pd
import matplotlib.pyplot as plt

host = "localhost"
port = 3305
user = "root"
password = "root"
database = "dma_project"

conn = mysql.connector.connect(host=host, port=port, user=user, password=password, database=database)
cursor = conn.cursor()

query = """
SELECT D.DEBRIS_ID, AVG(D.D_SIZE) AS AVERAGE_D_SIZE, LF.F_NAME
FROM SPACE_DEBRIS D
LEFT JOIN LAUNCH_FACILITY LF ON D.D_LAST_OBSERVED_DATE = LF.F_LAUNCHED_DATE
GROUP BY LF.F_NAME, D.DEBRIS_ID;
"""

cursor.execute(query)
data = cursor.fetchall()
columns = [desc[0] for desc in cursor.description]
df = pd.DataFrame(data, columns=columns)
cursor.close()
conn.close()
print("QUERY 5- Retrieve the details of satellites, including their names, launch dates, and the corresponding rocket r

plt.figure(figsize=(8, 4))
plt.hist(df['AVERAGE_D_SIZE'], bins=10, color='darkred', edgecolor='black')
plt.xlabel('Average Debris Size')
plt.ylabel('Debris ID')
plt.title('Distribution of Average Debris Size')
plt.show()
df.head(9)
```

```
QUERY 5- Retrieve the details of satellites, including their names, launch dates, and the corresponding rocket names
used for their launches:
```



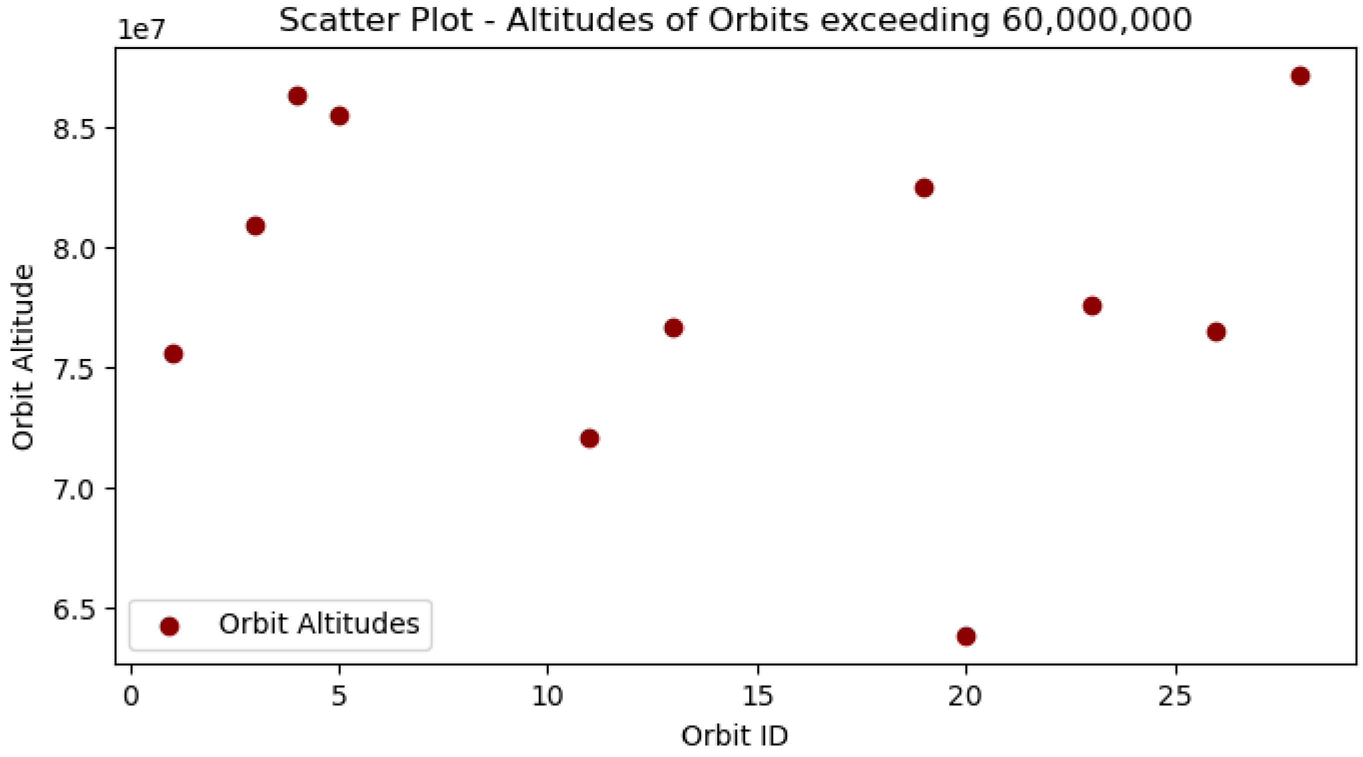
Out[23]:

	DEBRIS_ID	AVERAGE_D_SIZE	F_NAME
0	100	59676192.110000	None
1	101	24162452.760000	None
2	102	60907255.530000	None
3	103	85070511.790000	Alpha Spaceport
4	103	85070511.790000	Zodiac Spaceport
5	103	85070511.790000	Cosmic Launch Center
6	103	85070511.790000	Infinity Launch Center
7	103	85070511.790000	Saturn Spaceport
8	103	85070511.790000	Celestial Launch Center

In [26]:

```
#QUERY 6
import mysql.connector
import pandas as pd
import matplotlib.pyplot as plt

host = "localhost"
port = 3305
user = "root"
password = "root"
database = "dma_project"

conn = mysql.connector.connect(host=host, port=port, user=user, password=password, database=database)
cursor = conn.cursor()

query = """
SELECT * FROM ORBIT
WHERE O_ALTITUDE > 60000000;
"""
cursor.execute(query)
data = cursor.fetchall()
columns = [desc[0] for desc in cursor.description]
df = pd.DataFrame(data, columns=columns)
cursor.close()
conn.close()
print("QUERY 6- Retrieve information about orbits with altitudes exceeding 60,000,000:")

plt.figure(figsize=(8, 4))
plt.scatter(df['ORBIT_ID'], df['O_ALTITUDE'], color='darkred', label='Orbit Altitudes')
plt.xlabel('Orbit ID')
plt.ylabel('Orbit Altitude')
plt.title('Scatter Plot - Altitudes of Orbits exceeding 60,000,000')
plt.legend()
plt.show()
df
```

```
QUERY 6- Retrieve information about orbits with altitudes exceeding 60,000,000:
```



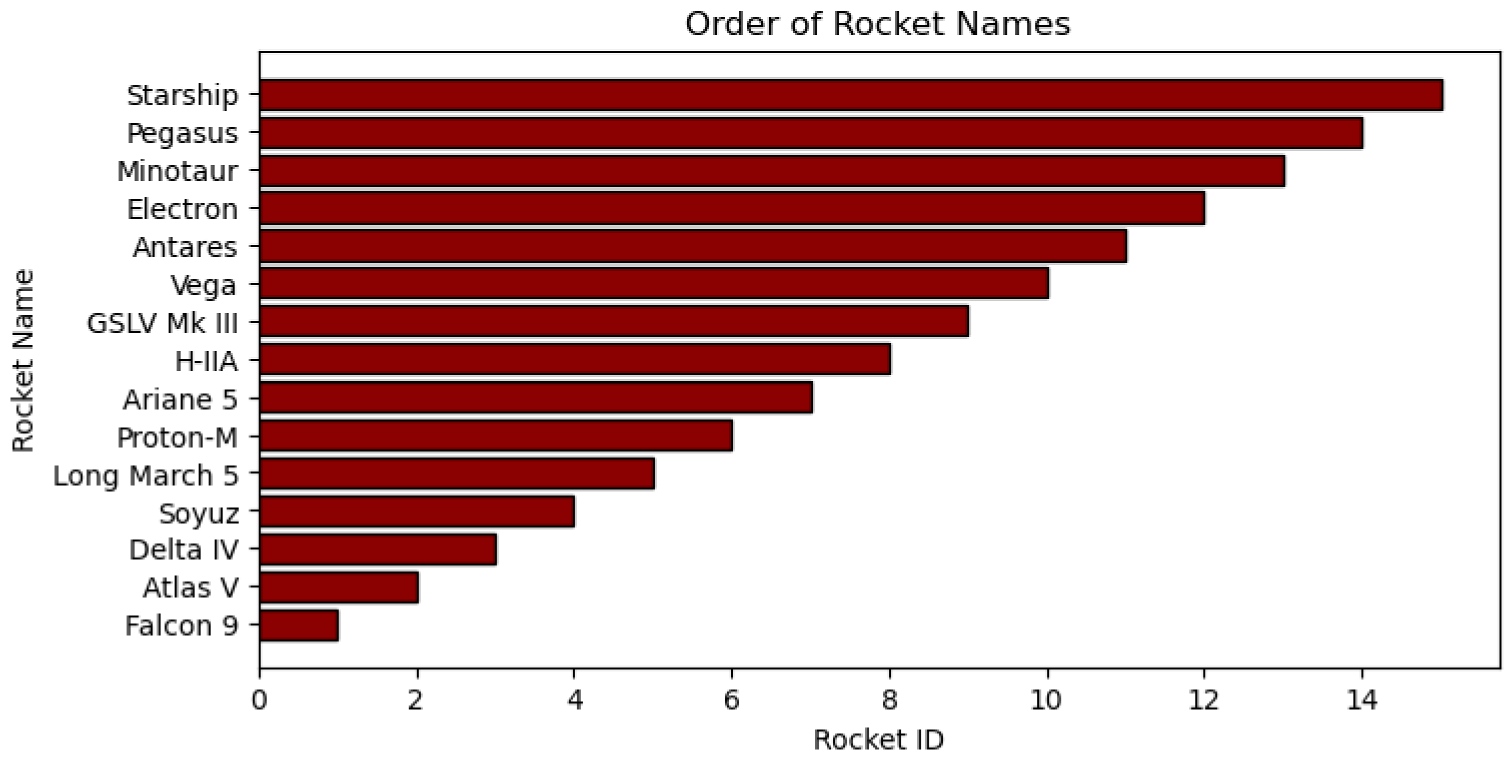
Out[26]:

	ORBIT_ID	O_ALTITUDE	O_INCLINATION	O_PERIOD
0	1	75573096.26	18657393.29	34560278.21
1	3	80929484.33	83275059.04	8496906.52
2	4	86314989.96	46128196.04	25527126.45
3	5	85444208.26	19662222.37	1685086.23
4	11	72108333.33	557270.62	95178955.22
5	13	76654964.60	39045090.84	94566158.73
6	19	82483176.31	41893568.94	32661766.87
7	20	63858377.62	98873271.47	24792772.41
8	23	77554518.92	29233093.42	27625389.62
9	26	76488810.35	65160817.24	28854486.74
10	28	87173473.91	27381324.72	4372014.35

In [24]:

```
#QUERY 7
host = "localhost"
port = 3305
user = "root"
password = "root"
database = "dma_project"

conn = mysql.connector.connect(host=host, port=port, user=user, password=password, database=database)
cursor = conn.cursor()

query = """
    SELECT R.ROCKET_ID, R.R_NAME
    FROM ROCKET R
    WHERE EXISTS (
        SELECT 1
        FROM PRESENT_IN P
        JOIN SENSOR S ON P.ROCKET_ID = R.ROCKET_ID
        WHERE S.SEN_DETECTION_FREQUENCY IS NOT NULL)
        LIMIT 15;
"""
cursor.execute(query)
data = cursor.fetchall()
cursor.close()
conn.close()
print("QUERY 7- Retrieve the Rocket names that have atleast one sensor associated with them:")

df = pd.DataFrame(data, columns=["ROCKET_ID", "R_NAME"])
plt.figure(figsize=(8, 4))
plt.barh(df["R_NAME"], df["ROCKET_ID"],color='darkred',edgecolor='black')
plt.xlabel("Rocket ID")
plt.ylabel("Rocket Name")
plt.title("Order of Rocket Names")
plt.show()
df
```

```
QUERY 7- Retrieve the Rocket names that have atleast one sensor associated with them:
```



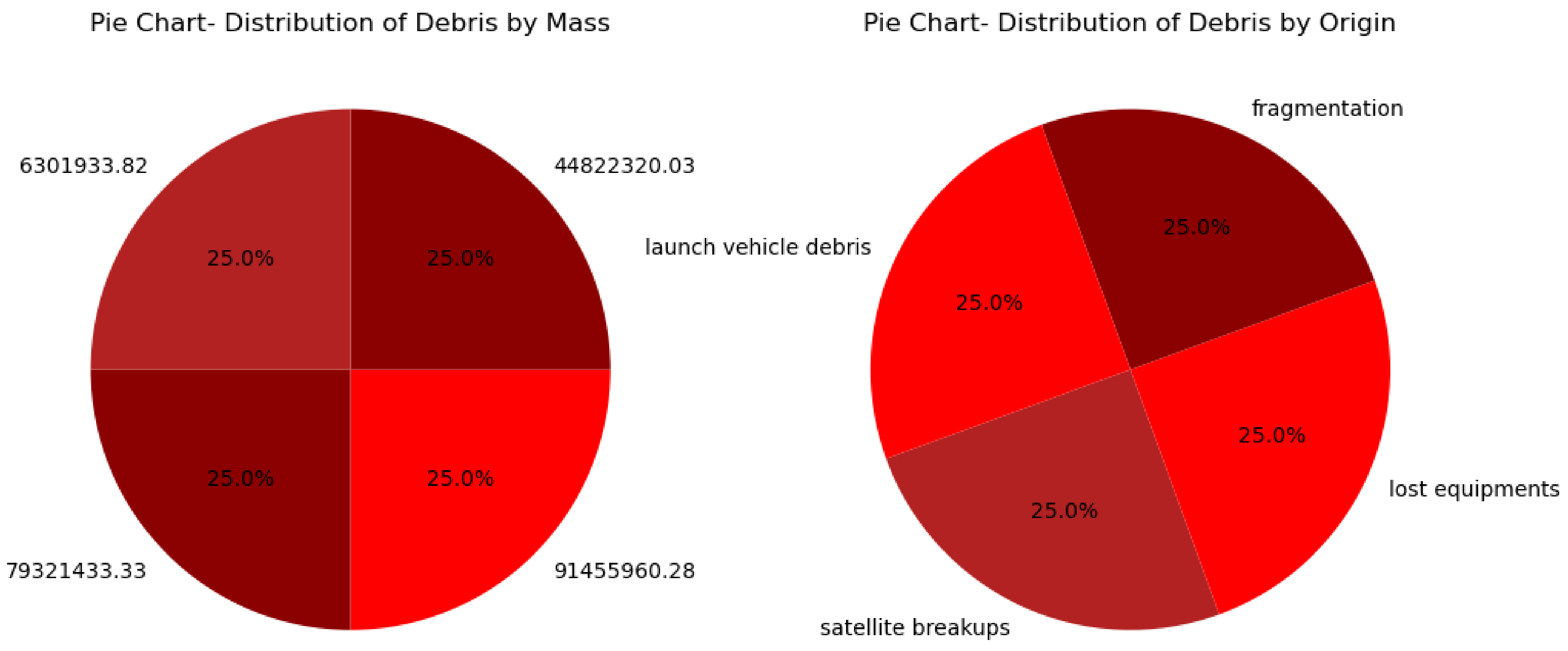
Out[24]:

	ROCKET_ID	R_NAME
0	1	Falcon 9
1	2	Atlas V
2	3	Delta IV
3	4	Soyuz
4	5	Long March 5
5	6	Proton-M
6	7	Ariane 5
7	8	H-IIA
8	9	GSLV Mk III
9	10	Vega
10	11	Antares
11	12	Electron
12	13	Minotaur
13	14	Pegasus
14	15	Starship

In [25]:

```
#QUERY 8
import mysql.connector
import pandas as pd
import matplotlib.pyplot as plt

host = "localhost"
port = 3305
user = "root"
password = "root"
database = "dma_project"

conn = mysql.connector.connect(host=host, port=port, user=user, password=password, database=database)
cursor = conn.cursor()

query = """
SELECT D.DEBRIS_ID, D.D_SIZE, D.D_MASS, D.D_LAST_OBSERVED_DATE, D.D_ORIGIN
FROM SPACE_DEBRIS D
WHERE NOT EXISTS (
    SELECT 1
    FROM PRESENT_IN PI
    WHERE PI.DEBRIS_ID = D.DEBRIS_ID
) OR D.D_LAST_OBSERVED_DATE < '2021-01-01';
"""

cursor.execute(query)
data = cursor.fetchall()
columns = [desc[0] for desc in cursor.description]
df = pd.DataFrame(data, columns=columns)
cursor.close()
conn.close()
print("QUERY 8- Retrieve the debris distribution by mass and origin:")

plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.pie(df['D_MASS'].value_counts(), labels=df['D_MASS'].value_counts().index, autopct='%1.1f%%', startangle=180,color
plt.title('Pie Chart- Distribution of Debris by Mass')

plt.subplot(1, 2, 2)
plt.pie(df['D_ORIGIN'].value_counts(), labels=df['D_ORIGIN'].value_counts().index, autopct='%1.1f%%', startangle=200,c
plt.title('Pie Chart- Distribution of Debris by Origin')
plt.show()
df
```

```
QUERY 8- Retrieve the debris distribution by mass and origin:
```



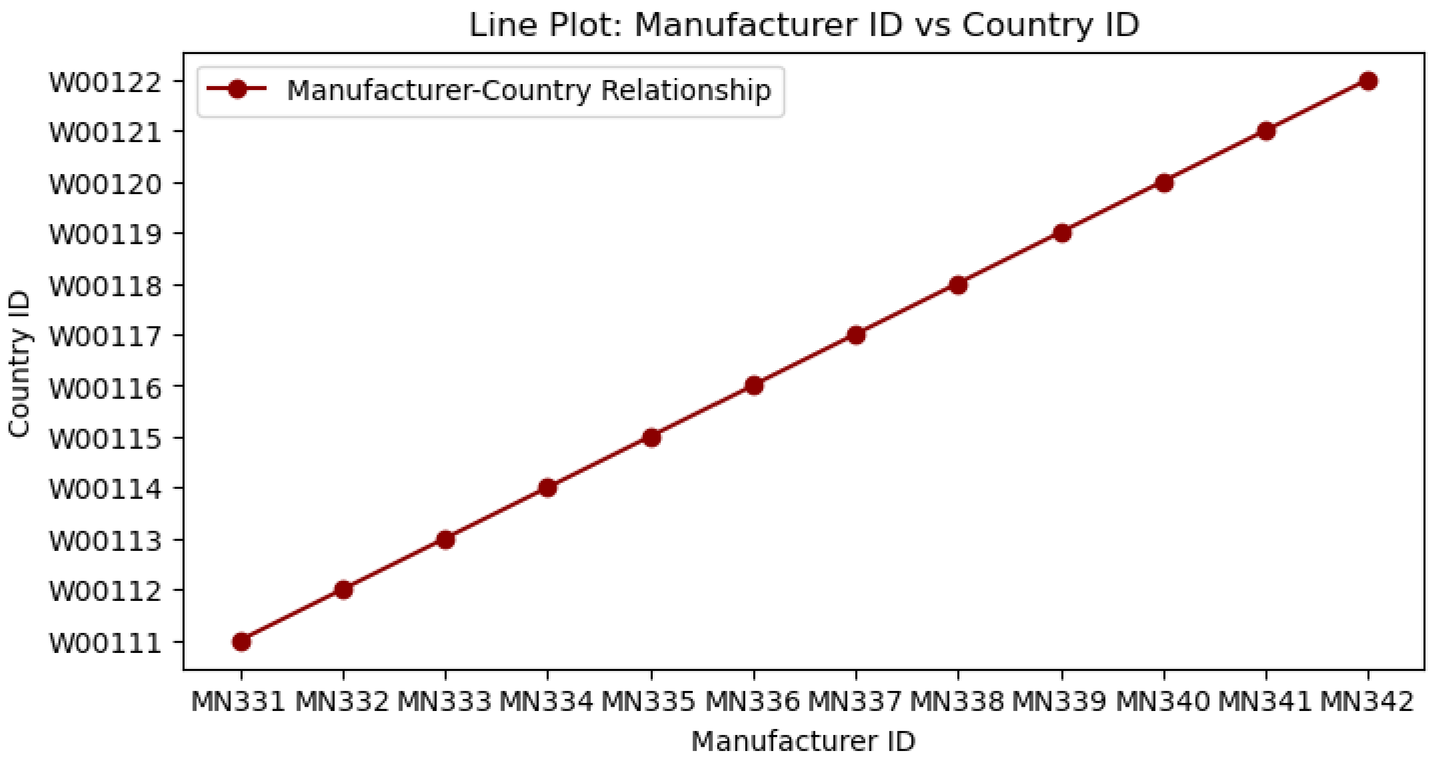
Out[25]:

	DEBRIS_ID	D_SIZE	D_MASS	D_LAST_OBSERVED_DATE	D_ORIGIN
0	104	95575034.80	79321433.33	2020-12-25	satellite breakups
1	111	8732048.57	91455960.28	2020-12-06	lost equipments
2	124	87111456.09	44822320.03	2020-11-13	fragmentation
3	130	45446815.47	6301933.82	2021-04-01	launch vehicle debris

In [34]:

```
#QUERY 9
import mysql.connector
import pandas as pd
import matplotlib.pyplot as plt
import warnings

host = "localhost"
port = 3305
user = "root"
password = "root"
database = "dma_project"

conn = mysql.connector.connect(host=host, port=port, user=user, password=password, database=database)
cursor = conn.cursor()
warnings.filterwarnings("ignore")

query = """
SELECT M.MANUFACTURE_ID,M.M_NAME AS MANUFACTURER_NAME,M.M_LOCATION AS MANUFACTURER_LOCATION,C.COUNTRY_ID,C.C_NAME AS CO
FROM MANUFACTURER M
JOIN COUNTRY_MANUFACTURER CM ON M.MANUFACTURE_ID = CM.MANUFACTURE_ID
JOIN COUNTRY C ON CM.COUNTRY_ID = C.COUNTRY_ID
LIMIT 12; """

df = pd.read_sql_query(query, conn)
cursor.close()
conn.close()
print("QUERY 9- Retrieve information about manufacturers and their countries:")

plt.figure(figsize=(8, 4))
plt.plot(df['MANUFACTURE_ID'], df['COUNTRY_ID'], marker='o', linestyle='-', color='darkred', label='Manufacturer-Countr
plt.title('Line Plot: Manufacturer ID vs Country ID')
plt.xlabel('Manufacturer ID')
plt.ylabel('Country ID')
plt.legend()
plt.show()
df
```

```
QUERY 9- Retrieve information about manufacturers and their countries:
```



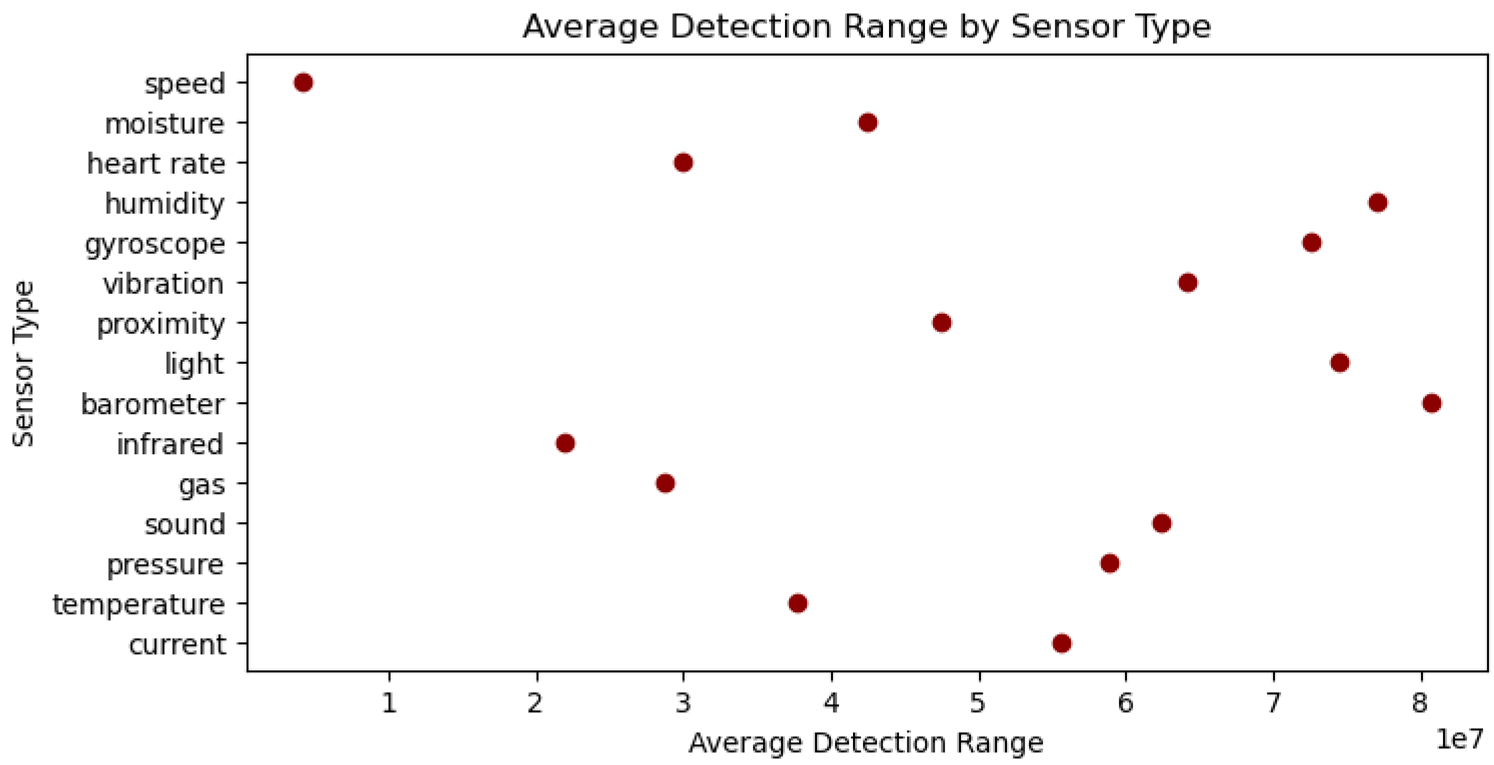
Out[34]:

	MANUFACTURE_ID	MANUFACTURER_NAME	MANUFACTURER_LOCATION	COUNTRY_ID	COUNTRY_NAME
0	MN331	GalacticTech	Australia	W00111	United States
1	MN332	StellarSystems	Australia	W00112	Canada
2	MN333	CosmoParts	Brazil	W00113	Mexico
3	MN334	AstroTech	United States	W00114	Brazil
4	MN335	StarForge	Germany	W00115	Argentina
5	MN336	CelestialMakers	India	W00116	United Kingdom
6	MN337	NebulaIndustries	Italy	W00117	Germany
7	MN338	OrbitTech	Australia	W00118	France
8	MN339	LunarCraft	Japan	W00119	Italy
9	MN340	SaturnSolutions	China	W00120	Spain
10	MN341	CometComponents	India	W00121	Russia
11	MN342	SolarTech	Italy	W00122	China

In [32]:

```
#QUERY 10
host = "localhost"
port = 3305
user = "root"
password = "root"
database = "dma_project"

conn = mysql.connector.connect(host=host, port=port, user=user, password=password, database=database)
cursor = conn.cursor()

query = """
SELECT SEN_TYPE, AVG(SEN_DETECTION_RANGE) AS AvgDetectionRange
FROM SENSOR
GROUP BY SEN_TYPE;
"""
cursor.execute(query)
data = cursor.fetchall()
columns = [desc[0] for desc in cursor.description]
df = pd.DataFrame(data, columns=columns)
cursor.close()
conn.close()
print("QUERY 10- Retrive the average detection range by type of sensor:")

plt.figure(figsize=(8, 4))
plt.scatter(df['AvgDetectionRange'], df['SEN_TYPE'],color='darkred')
plt.xlabel('Average Detection Range')
plt.ylabel('Sensor Type')
plt.title('Average Detection Range by Sensor Type')
plt.show()
df
```

```
QUERY 10- Retrive the average detection range by type of sensor:
```



Out[32]:

	SEN_TYPE	AvgDetectionRange
0	current	55657661.215000
1	temperature	37674382.630000
2	pressure	58866465.065000
3	sound	62333420.543333
4	gas	28760070.063333
5	infrared	21966670.910000
6	barometer	80663383.075000
7	light	74416404.700000
8	proximity	47407092.070000
9	vibration	64098621.020000
10	gyroscope	72619183.020000
11	humidity	77097882.510000
12	heart rate	29935489.130000
13	moisture	42408760.090000
14	speed	4139282.560000