In [3]:

```python
# Data 201 Project #1
# Walter Hinkley
```

In [ ]:

```python
# Bag Tax data set comes data montgomery and it gives
# information about store locations, bag count and cost of bags
# used to package goods.

# I am curious if there is a difference in the amount of bags
# used by city.
```

In [5]:

```python
import numpy as np
```

In [7]:

```python
import pandas as pd
```

In [9]:

```python
import matplotlib.pyplot as plt
```

In [11]:

```python
import seaborn as sns
```

In [13]:

```python
bags = pd.read_csv('Bag_Tax_20250404.csv')
```

In [15]:

```python
bags.head()
```

Out[15]:

	File ID	Account	Date From	Date To	Bag Count	Amount Collected	Amount Due	Amount Retained	
0	82554	589	03/01/2022	03/31/2022	12872	643.60	514.88	128.72	04/
1	68385	863	10/01/2020	10/31/2020	454	22.70	18.16	4.54	11/
2	57463	1470	07/01/2019	07/31/2019	0	0.00	0.00	0.00	08/
3	80143	552	12/01/2021	12/31/2021	861	43.05	34.44	8.61	01/
4	46402	1470	04/01/2018	04/30/2018	0	0.00	0.00	0.00	05/

In [17]:

```python
bags.describe()
```

Out[17]:

	File ID	Account	Bag Count	Amount Collected	Amount Due	
count	74045.000000	74045.000000	7.404500e+04	74045.000000	74045.000000	74
mean	46602.325316	1054.333973	1.181932e+04	590.964982	472.772887	
std	30744.384167	1216.081710	7.129513e+04	3564.756520	2851.805161	
min	12.000000	1.000000	0.000000e+00	0.000000	0.000000	
25%	18557.000000	293.000000	2.640000e+02	13.200000	10.560000	
50%	45802.000000	602.000000	1.162000e+03	58.100000	46.480000	
75%	76219.000000	1218.000000	3.603000e+03	180.150000	144.120000	
max	97612.000000	5241.000000	2.134035e+06	106701.750000	85361.400000	21

In [19]:

```
maryland_bags = bags[bags['State'] == 'MD']
```

In [21]:

```
maryland_bags.head()
```

Out[21]:

	File ID	Account	Date From	Date To	Bag Count	Amount Collected	Amount Due	Amount Retained	
5	65348	1300	08/23/2019	06/26/2020	8000	400.00	320.00	80.00	06
9	68185	3838	10/01/2020	10/31/2020	390	19.50	15.60	3.90	11
11	68165	1302	10/01/2020	10/31/2020	5689	284.45	227.56	56.89	11
12	68378	229	10/01/2020	10/31/2020	1620	81.00	64.80	16.20	1
13	68205	1336	10/01/2020	10/31/2020	303	15.15	12.12	3.03	11

In [23]:

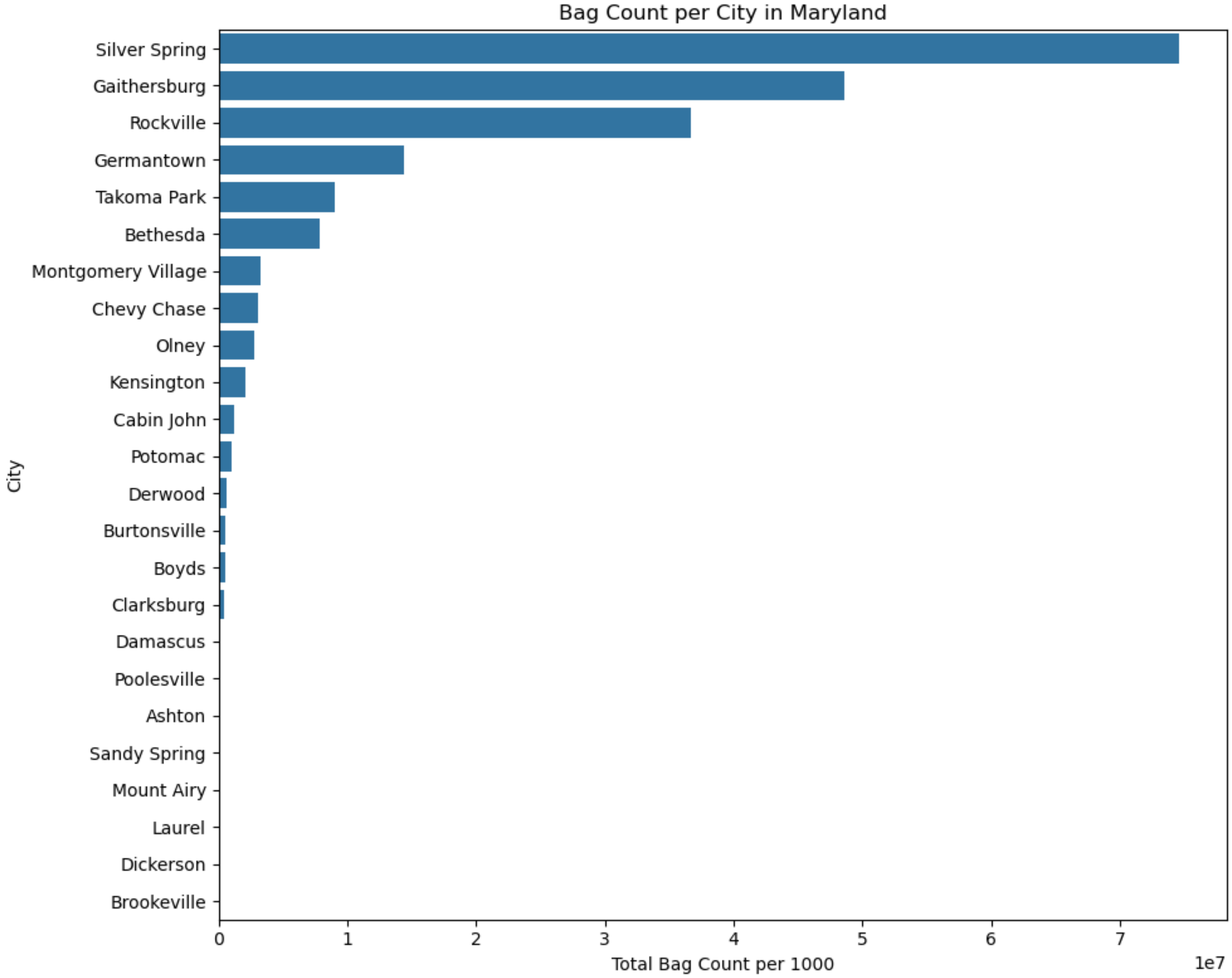
```
city_count = maryland_bags.groupby('City')['Bag Count'].sum().reset_index()
```

In [25]:

```
city_count = city_count.sort_values('Bag Count', ascending=False)
```

In [27]:

```
plt.figure(figsize=(10, 8))
sns.barplot(y='City', x='Bag Count', data=city_count)
plt.title('Bag Count per City in Maryland')
plt.xlabel('Total Bag Count per 1000')
plt.ylabel('City')
plt.tight_layout()
plt.show()
```



In [29]:

```
median_amount_retained = maryland_bags['Amount Retained'].median()
```

In [31]:

```
print(f"Median Amount Retained: {median_amount_retained}")
```

```
Median Amount Retained: 8.54
```

In [35]:

```
num_rows = len(maryland_bags)
print(f"Number of rows in maryland_bags: {num_rows}")
```

```
Number of rows in maryland_bags: 52269
```

In [37]:

```
bootstrap_sample = maryland_bags.sample(n=5300, replace=True, random_state=1
print(f"Bootstrap sample shape: {bootstrap_sample.shape}")
```

```
Bootstrap sample shape: (5300, 14)
```

In [39]:

```
bootstrap_sample.head()
```

Out[39]:

	File ID	Account	Date From	Date To	Bag Count	Amount Collected	Amount Due	Amount Retained
47764	29265	288	10/01/2016	10/31/2016	679	33.95	27.16	6.79
67343	91345	4998	12/01/2023	12/31/2023	327	16.35	13.08	3.27
47408	28950	446	10/01/2016	10/31/2016	5049	252.45	201.96	50.49
48500	29662	1336	12/01/2016	12/31/2016	319	15.95	12.76	3.19
32431	27168	859	07/01/2016	07/31/2016	156	7.80	6.24	1.56

In [41]:

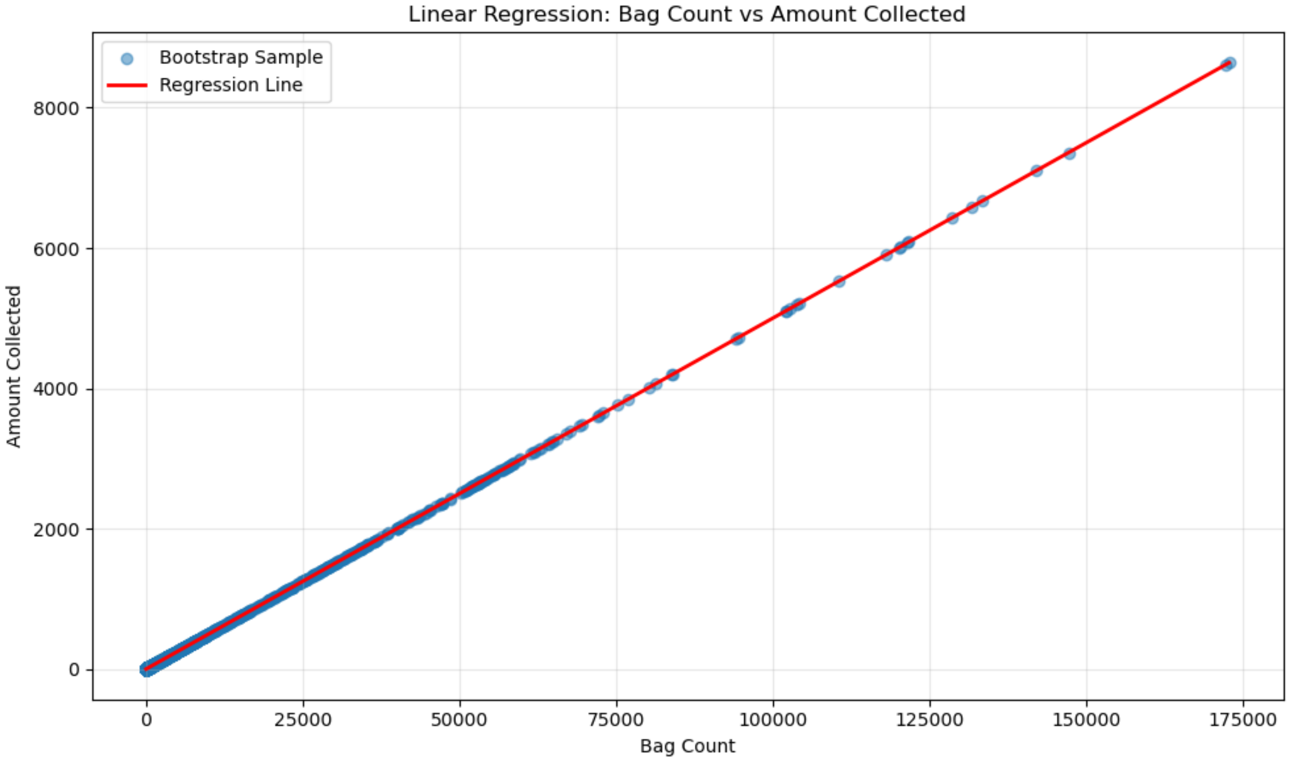
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
```

In [43]:

```
X = bootstrap_sample['Bag Count'].values.reshape(-1, 1)  # Independent varia
y = bootstrap_sample['Amount Collected']  # Dependent variable

# Create and fit the linear regression model
model = LinearRegression()
model.fit(X, y)

# Get model parameters
slope = model.coef_[0]
intercept = model.intercept_
r_squared = model.score(X, y)

# Print results
print(f"Linear Regression Model: y = {slope:.4f}x + {intercept:.4f}")
print(f"R-squared: {r_squared:.4f}")

# Create predictions for plotting
X_pred = np.linspace(X.min(), X.max(), 100).reshape(-1, 1)
y_pred = model.predict(X_pred)
```

```
# Plot the data and regression line
plt.figure(figsize=(10, 6))
plt.scatter(X, y, alpha=0.5, label='Bootstrap Sample')
plt.plot(X_pred, y_pred, color='red', linewidth=2, label='Regression Line')
plt.xlabel('Bag Count')
plt.ylabel('Amount Collected')
plt.title('Linear Regression: Bag Count vs Amount Collected')
plt.legend()
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()
```

```
Linear Regression Model: y = 0.0500x + 0.0000
R-squared: 1.0000
```



In [ ]:

```
# Anaconda assistant used to help with linear regression and
# plot of bag count to amount collected regression
```