In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
data=pd.read_csv(r"C:\Users\saisn\Downloads\RPC13_Input_For_Participants\RPC13_Input_F
```

In [3]:
```
data
```

Out[3]:

	month	city_id	new_passengers	repeat_passengers	total_passengers
0	2024-01-01	AP01	2513	650	3163
1	2024-01-01	CH01	3920	720	4640
2	2024-01-01	GJ01	2432	1184	3616
3	2024-01-01	GJ02	2089	544	2633
4	2024-01-01	KA01	1957	172	2129
5	2024-01-01	KL01	4865	795	5660
6	2024-01-01	MP01	2843	1033	3876
7	2024-01-01	RJ01	10423	1422	11845
8	2024-01-01	TN01	1822	392	2214
9	2024-01-01	UP01	3465	1431	4896
10	2024-02-01	AP01	2380	790	3170
11	2024-02-01	CH01	4104	853	4957
12	2024-02-01	GJ01	2254	1313	3567
13	2024-02-01	GJ02	2146	610	2756
14	2024-02-01	KA01	2107	183	2290
15	2024-02-01	KL01	4367	1005	5372
16	2024-02-01	MP01	2878	1103	3981
17	2024-02-01	RJ01	10789	1661	12450
18	2024-02-01	TN01	1647	346	1993
19	2024-02-01	UP01	3529	1659	5188
20	2024-03-01	AP01	2170	923	3093
21	2024-03-01	CH01	3228	872	4100
22	2024-03-01	GJ01	1946	1494	3440
23	2024-03-01	GJ02	1763	759	2522
24	2024-03-01	KA01	1986	208	2194
25	2024-03-01	KL01	4865	1348	6213
26	2024-03-01	MP01	2742	1091	3833
27	2024-03-01	RJ01	7417	1840	9257
28	2024-03-01	TN01	1538	427	1965
29	2024-03-01	UP01	3159	1622	4781
30	2024-04-01	AP01	1845	992	2837
31	2024-04-01	CH01	2496	789	3285
32	2024-04-01	GJ01	1843	1551	3394
33	2024-04-01	GJ02	1637	862	2499

	month	city_id	new_passengers	repeat_passengers	total_passengers
34	2024-04-01	KA01	1836	236	2072
35	2024-04-01	KL01	4939	1576	6515
36	2024-04-01	MP01	2351	1295	3646
37	2024-04-01	RJ01	6120	1736	7856
38	2024-04-01	TN01	1242	480	1722
39	2024-04-01	UP01	2311	1496	3807
40	2024-05-01	AP01	1939	951	2890
41	2024-05-01	CH01	2730	969	3699
42	2024-05-01	GJ01	1611	1606	3217
43	2024-05-01	GJ02	1388	868	2256
44	2024-05-01	KA01	1921	349	2270
45	2024-05-01	KL01	4369	1853	6222
46	2024-05-01	MP01	2028	1563	3591
47	2024-05-01	RJ01	5332	1842	7174
48	2024-05-01	TN01	1039	504	1543
49	2024-05-01	UP01	1825	1662	3487
50	2024-06-01	AP01	1900	802	2702
51	2024-06-01	CH01	2430	867	3297
52	2024-06-01	GJ01	1540	1490	3030
53	2024-06-01	GJ02	1104	703	1807
54	2024-06-01	KA01	1874	329	2203
55	2024-06-01	KL01	3011	1049	4060
56	2024-06-01	MP01	2021	1131	3152
57	2024-06-01	RJ01	5775	1181	6956
58	2024-06-01	TN01	1226	402	1628
59	2024-06-01	UP01	1971	1727	3698

In [4]:
```
data.head(5)
```

Out[4]:

	month	city_id	new_passengers	repeat_passengers	total_passengers
0	2024-01-01	AP01	2513	650	3163
1	2024-01-01	CH01	3920	720	4640
2	2024-01-01	GJ01	2432	1184	3616
3	2024-01-01	GJ02	2089	544	2633
4	2024-01-01	KA01	1957	172	2129

In [5]:
```
data.tail(5)
```

Out[5]:

	month	city_id	new_passengers	repeat_passengers	total_passengers
55	2024-06-01	KL01	3011	1049	4060
56	2024-06-01	MP01	2021	1131	3152
57	2024-06-01	RJ01	5775	1181	6956
58	2024-06-01	TN01	1226	402	1628
59	2024-06-01	UP01	1971	1727	3698

In [6]:
```
data.describe()
```

Out[6]:

	new_passengers	repeat_passengers	total_passengers
count	60.000000	60.000000	60.000000
mean	2949.966667	1021.850000	3971.816667
std	1960.397676	489.245833	2233.462741
min	1039.000000	172.000000	1543.000000
25%	1844.500000	689.750000	2516.250000
50%	2212.000000	980.500000	3417.000000
75%	3287.250000	1491.000000	4675.250000
max	10789.000000	1853.000000	12450.000000

In [9]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60 entries, 0 to 59
Data columns (total 5 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   month              60 non-null     object
 1   city_id            60 non-null     object
 2   new_passengers     60 non-null     int64 
 3   repeat_passengers  60 non-null     int64 
 4   total_passengers   60 non-null     int64 
dtypes: int64(3), object(2)
memory usage: 2.5+ KB
```

In [10]:
```
data.isnull()
```

Out[10]:

	month	city_id	new_passengers	repeat_passengers	total_passengers
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
5	False	False	False	False	False
6	False	False	False	False	False
7	False	False	False	False	False
8	False	False	False	False	False
9	False	False	False	False	False
10	False	False	False	False	False
11	False	False	False	False	False
12	False	False	False	False	False
13	False	False	False	False	False
14	False	False	False	False	False
15	False	False	False	False	False
16	False	False	False	False	False
17	False	False	False	False	False
18	False	False	False	False	False
19	False	False	False	False	False
20	False	False	False	False	False
21	False	False	False	False	False
22	False	False	False	False	False
23	False	False	False	False	False
24	False	False	False	False	False
25	False	False	False	False	False
26	False	False	False	False	False
27	False	False	False	False	False
28	False	False	False	False	False
29	False	False	False	False	False
30	False	False	False	False	False
31	False	False	False	False	False
32	False	False	False	False	False
33	False	False	False	False	False

	month	city_id	new_passengers	repeat_passengers	total_passengers
34	False	False	False	False	False
35	False	False	False	False	False
36	False	False	False	False	False
37	False	False	False	False	False
38	False	False	False	False	False
39	False	False	False	False	False
40	False	False	False	False	False
41	False	False	False	False	False
42	False	False	False	False	False
43	False	False	False	False	False
44	False	False	False	False	False
45	False	False	False	False	False
46	False	False	False	False	False
47	False	False	False	False	False
48	False	False	False	False	False
49	False	False	False	False	False
50	False	False	False	False	False
51	False	False	False	False	False
52	False	False	False	False	False
53	False	False	False	False	False
54	False	False	False	False	False
55	False	False	False	False	False
56	False	False	False	False	False
57	False	False	False	False	False
58	False	False	False	False	False
59	False	False	False	False	False

In [ ]:

In [11]:
```
data.describe().T
```

Out[11]:

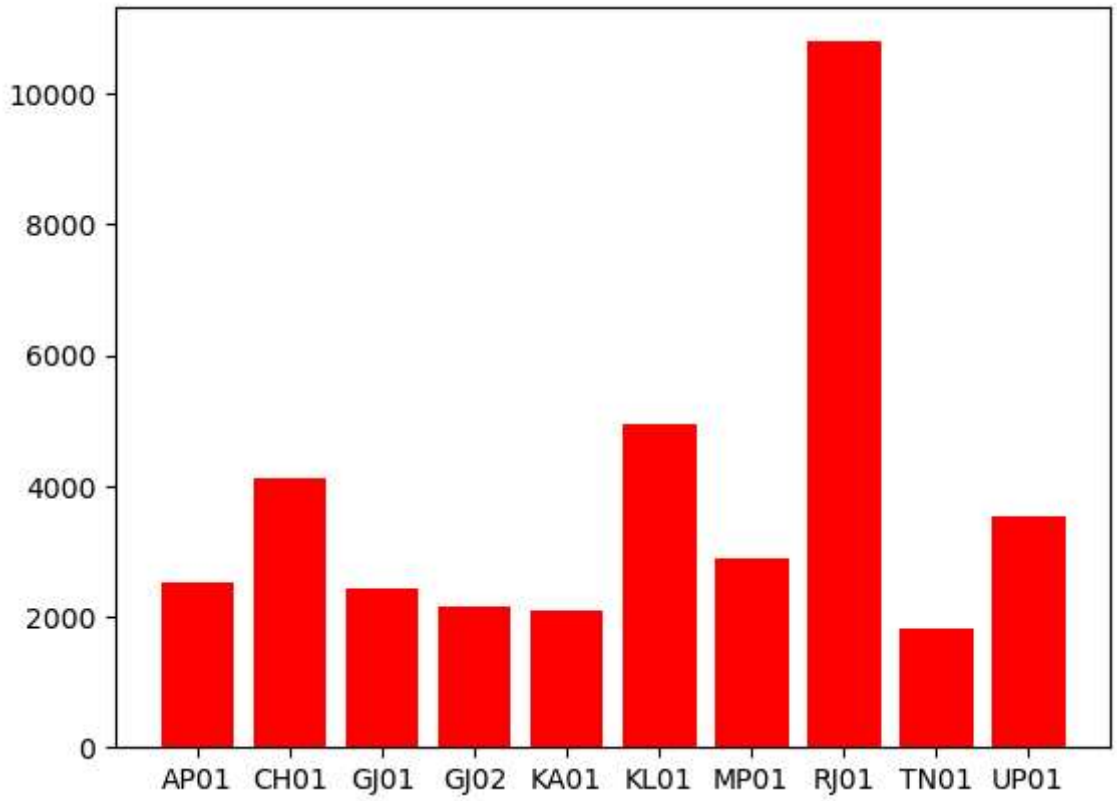
	count	mean	std	min	25%	50%	75%	max
new_passengers	60.0	2949.966667	1960.397676	1039.0	1844.50	2212.0	3287.25	10789.0
repeat_passengers	60.0	1021.850000	489.245833	172.0	689.75	980.5	1491.00	1853.0
total_passengers	60.0	3971.816667	2233.462741	1543.0	2516.25	3417.0	4675.25	12450.0

In [14]:
```
data.columns
```

Out[14]:
```
Index(['month', 'city_id', 'new_passengers', 'repeat_passengers',
       'total_passengers'],
      dtype='object')
```

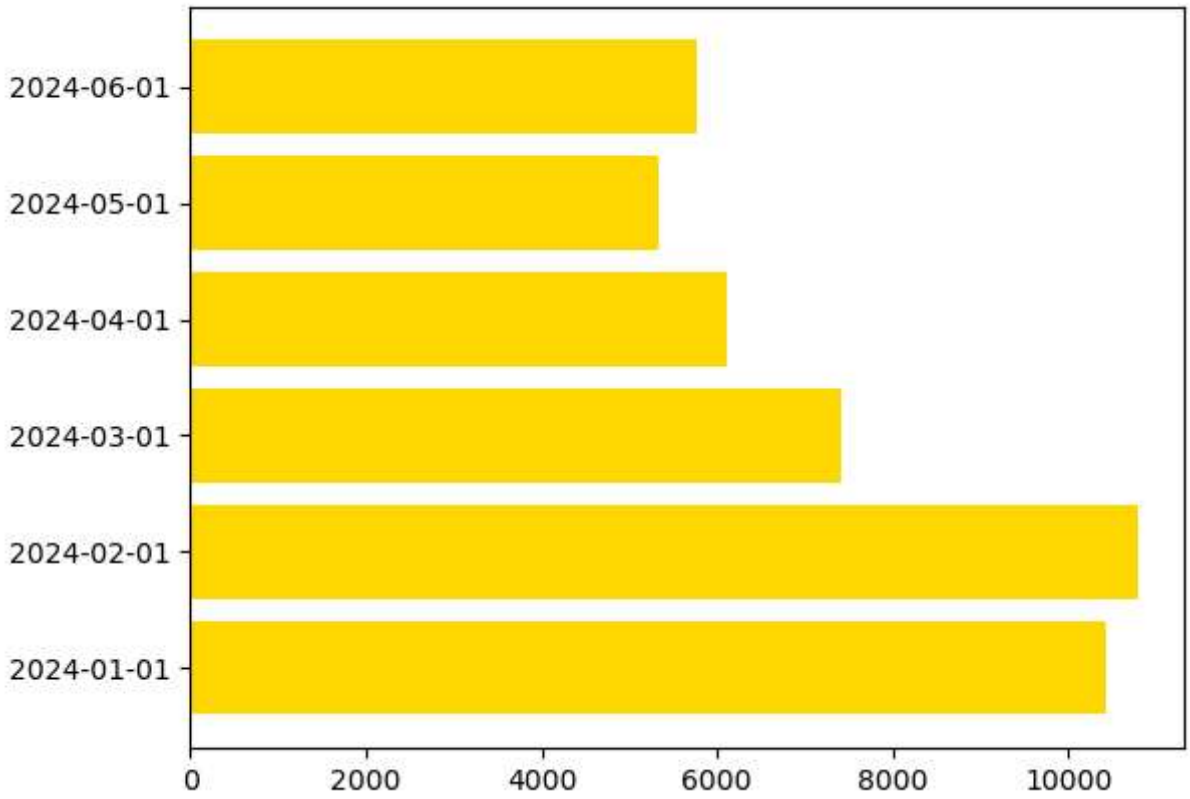
In [50]:
```
plt.bar(data['city_id'],data['new_passengers'],color='red')
```

Out[50]:
```
<BarContainer object of 60 artists>
```



In [64]:
```
plt.barh(data['month'],data['new_passengers'],color="gold")
plt.figure(figsize=(20,10))
```

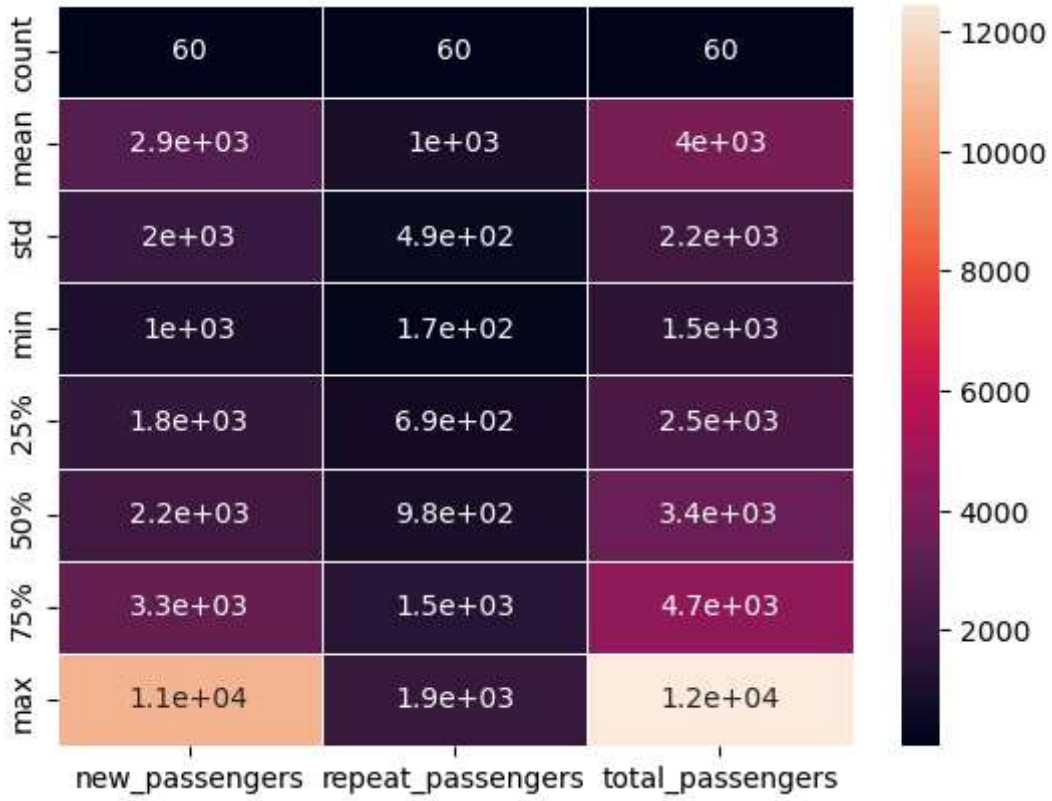
Out[64]:
```
<Figure size 2000x1000 with 0 Axes>
```



<Figure size 2000x1000 with 0 Axes>

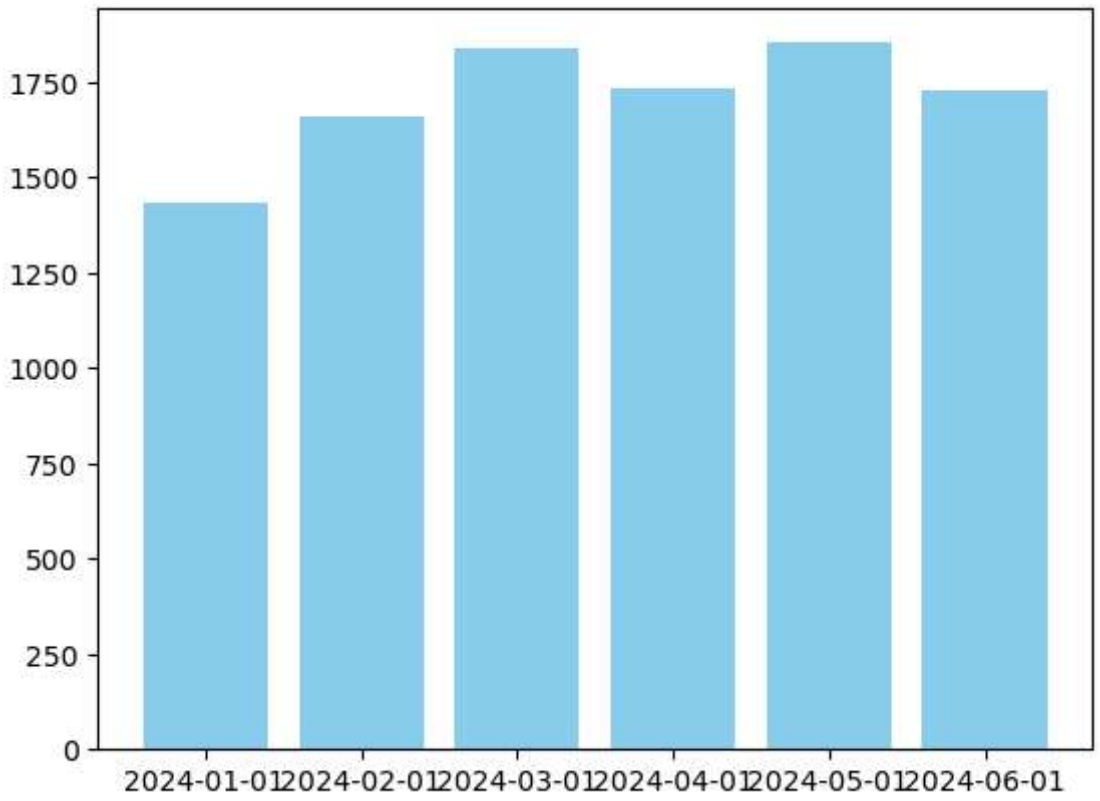
In [23]:
```
sns.heatmap(data.describe(),annot=True,linewidth=0.5)
```

Out[23]: <Axes: >

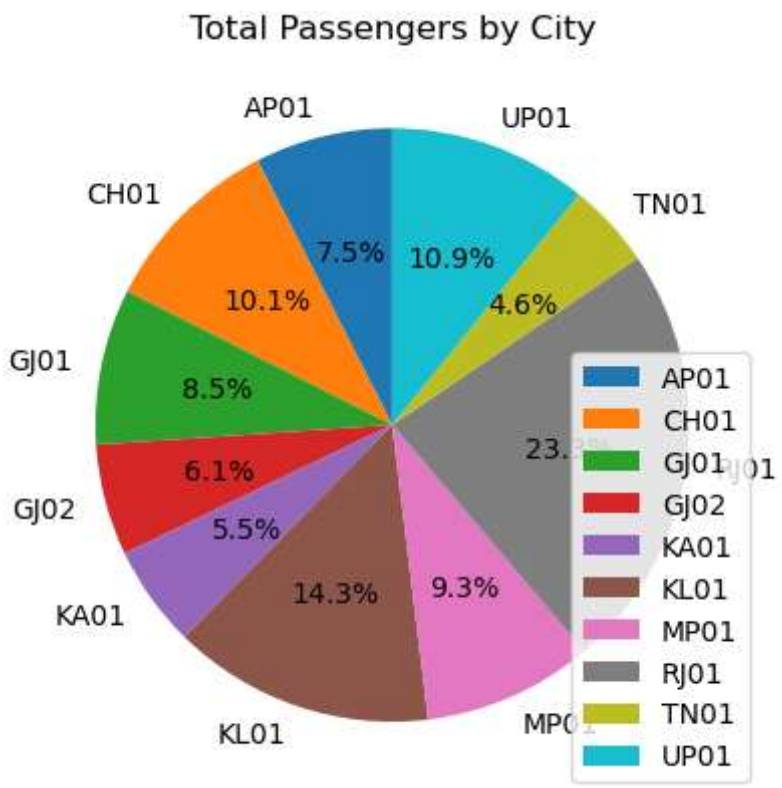


In [54]:
```
plt.bar(data['month'],data['repeat_passengers'],color="skyblue")
```

Out[54]: <BarContainer object of 60 artists>

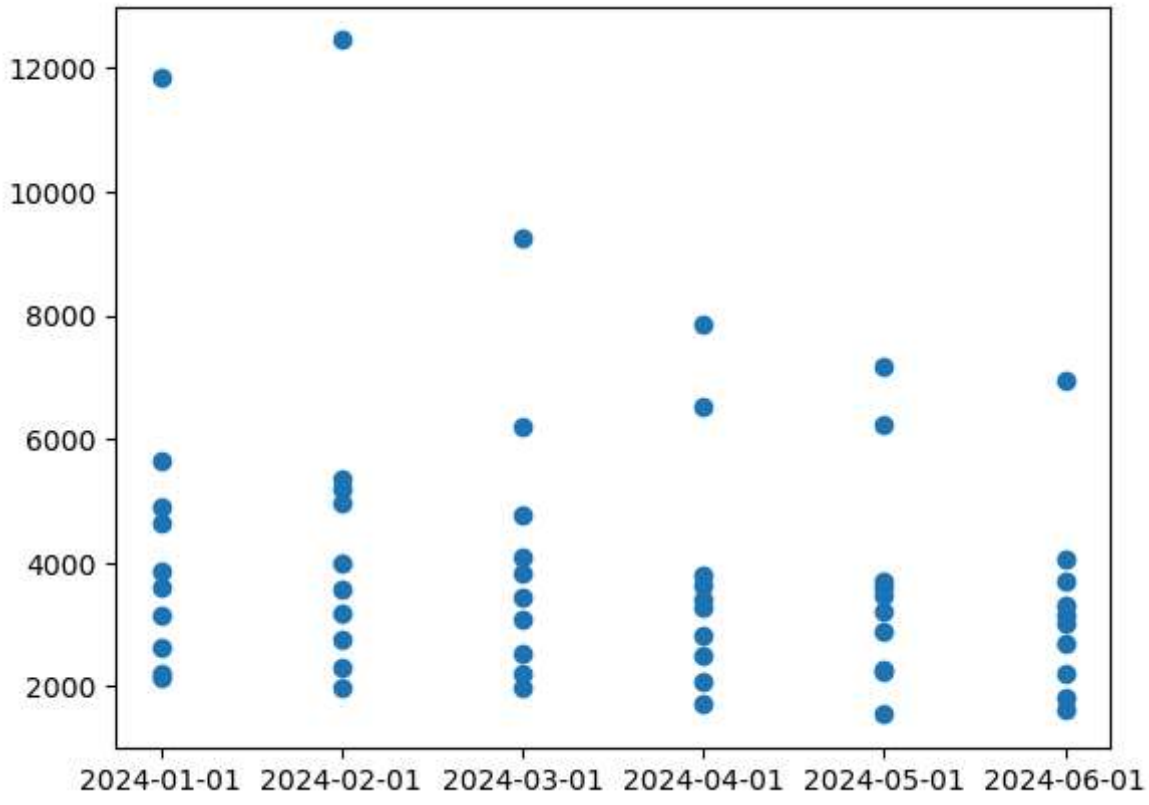


In [59]:
```
import matplotlib.pyplot as plt

# Group data by 'city_id' and calculate the sum of 'total_passengers'
grouped_data = data.groupby('city_id')['total_passengers'].sum()

# Plot pie chart for total passengers by city
plt.pie(grouped_data, labels=grouped_data.index, autopct='%1.1f%%', startangle=90)
plt.title("Total Passengers by City")
plt.legend()
plt.show()
```



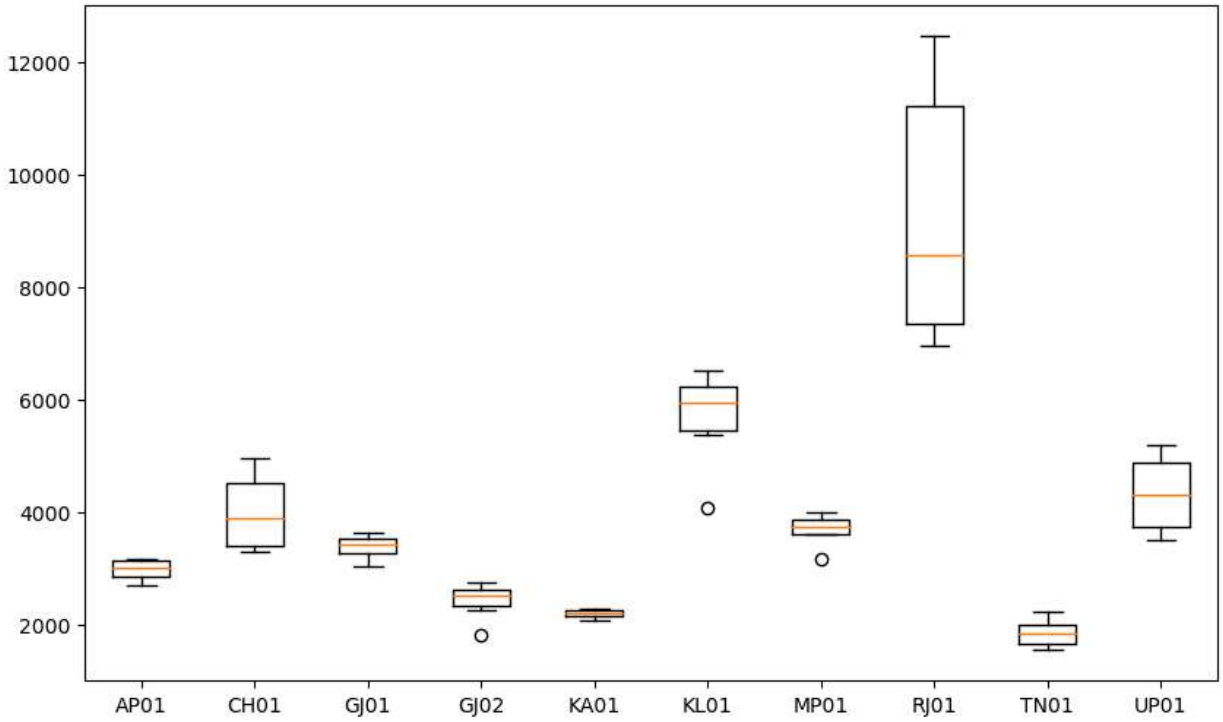
In [39]:
```
plt.scatter(data['month'],data['total_passengers'])
```

Out[39]:
```
<matplotlib.collections.PathCollection at 0x1e7f9048d90>
```



In [49]:
```
import matplotlib.pyplot as plt

grouped_data = [group['total_passengers'].values for name, group in data.groupby('city
```

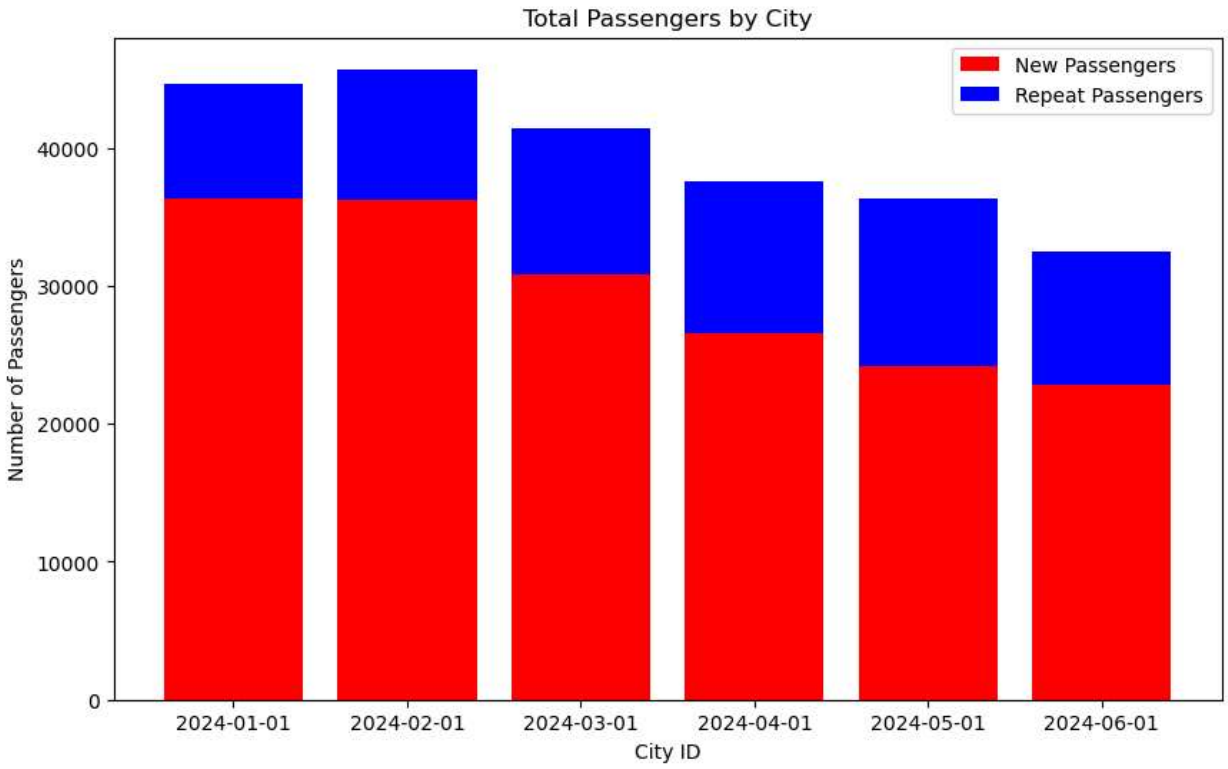
```
city_labels = data['city_id'].unique()

plt.figure(figsize=(10, 6))
plt.boxplot(grouped_data, labels=city_labels)
plt.show()
```



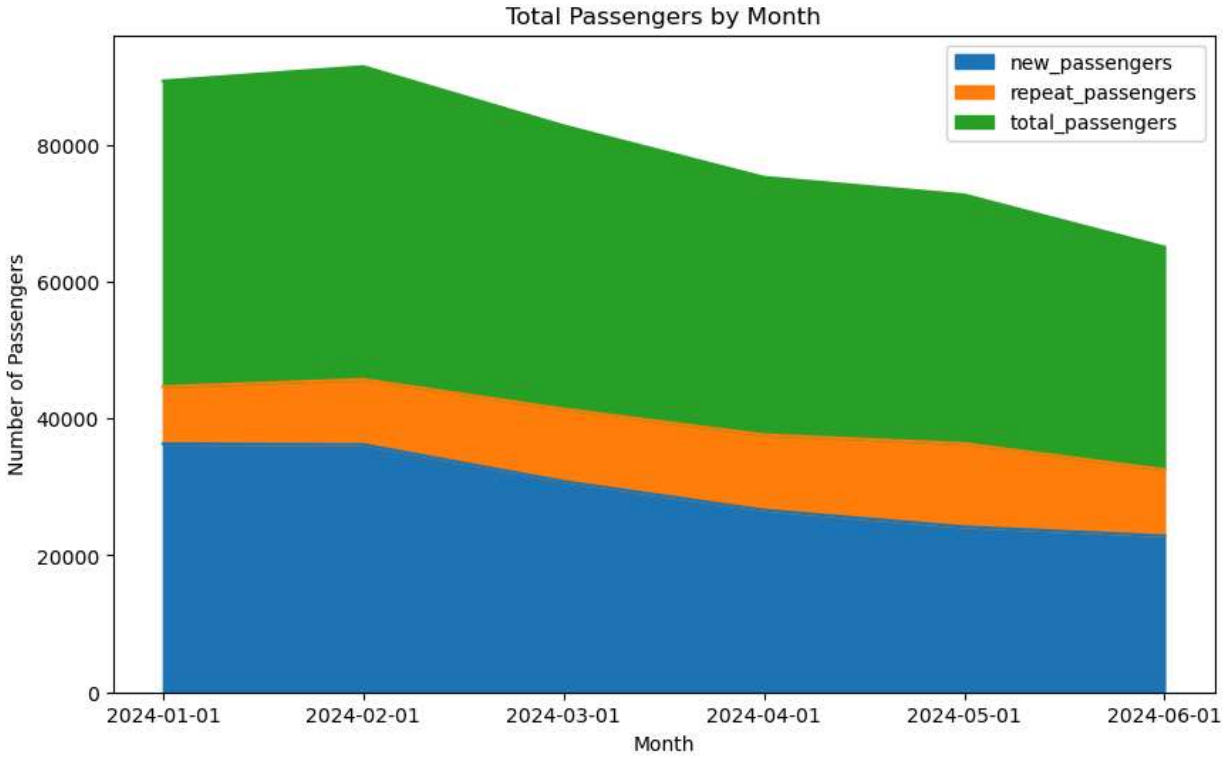
In [46]:

```
import matplotlib.pyplot as plt

# Group data by city and get the total of new and repeat passengers
grouped_data = data.groupby('city_id')[['new_passengers', 'repeat_passengers']].sum()

# Create a stacked bar chart
plt.figure(figsize=(10, 6))
plt.bar(grouped_data.index, grouped_data['new_passengers'], label='New Passengers', co
plt.bar(grouped_data.index, grouped_data['repeat_passengers'], bottom=grouped_data['ne

# Add labels and legend
plt.xlabel('City ID')
plt.ylabel('Number of Passengers')
plt.title('Total Passengers by City')
plt.legend()

# Show the plot
plt.show()
```

In [48]:
```
import matplotlib.pyplot as plt

# Group data by city and get the total of new and repeat passengers
grouped_data = data.groupby('month')[['new_passengers', 'repeat_passengers']].sum()

# Create a stacked bar chart
plt.figure(figsize=(10, 6))
plt.bar(grouped_data.index, grouped_data['new_passengers'], label='New Passengers', co
plt.bar(grouped_data.index, grouped_data['repeat_passengers'], bottom=grouped_data['ne

# Add labels and legend
plt.xlabel('City ID')
plt.ylabel('Number of Passengers')
plt.title('Total Passengers by City')
plt.legend()

# Show the plot
plt.show()
```



In [57]:
```
# Group data by month and get the total of new and repeat passengers
grouped_data = data.groupby('month')[['new_passengers', 'repeat_passengers', 'total_pa

# Plot an area chart using Pandas
grouped_data.plot.area(stacked=True, figsize=(10, 6))

# Add labels and title
plt.xlabel('Month')
plt.ylabel('Number of Passengers')
plt.title('Total Passengers by Month')

# Show the plot
plt.show()
```


Total Passengers by Month
new_passengers
repeat_passengers
total_passengers
Number of Passengers
0
20000
40000
60000
80000
2024-01-01
2024-02-01
2024-03-01
2024-04-01
2024-05-01
2024-06-01
Month

In [ ]: