JacobsonDSC640Weeks9-10Exercise

August 9, 2025

[2]:
```
import pandas as pd
sub = pd.read_csv('complaints-by-subcategory.csv')
cat = pd.read_csv('complaints-by-category.csv')
air = pd.read_csv('complaints-by-airport.csv')
iata = pd.read_csv('iata-icao.csv')
```

[18]:
```
sub.head()
```

[18]:
```
  pdf_report_date airport                             category  \
0         2019-02     ABE         Hazardous Materials Safety
1         2019-02     ABE  Mishandling of Passenger Property
2         2019-02     ABE         Hazardous Materials Safety
3         2019-02     ABE  Mishandling of Passenger Property
4         2019-02     ABE         Hazardous Materials Safety

                          subcategory year_month  count  \
0                             General    2015-01      0
1  Damaged/Missing Items--Checked Baggage    2015-01      0
2                             General    2015-02      0
3  Damaged/Missing Items--Checked Baggage    2015-02      0
4                             General    2015-03      0

                           clean_cat                             clean_subcat  \
0         Hazardous Materials Safety                                  General
1  Mishandling of Passenger Property  *Damaged/Missing Items--Checked Baggage
2         Hazardous Materials Safety                                  General
3  Mishandling of Passenger Property  *Damaged/Missing Items--Checked Baggage
4         Hazardous Materials Safety                                  General

  clean_cat_status clean_subcat_status  is_category_prefix_removed
0         original            original                       False
1         original            original                       False
2         original            original                       False
3         original            original                       False
4         original            original                       False
```

[4]:
```
cat.head()
```

```
[4]:   pdf_report_date airport                           category year_month  \
     0         2019-02     ABE         Hazardous Materials Safety    2015-01
     1         2019-02     ABE  Mishandling of Passenger Property    2015-01
     2         2019-02     ABE         Hazardous Materials Safety    2015-02
     3         2019-02     ABE  Mishandling of Passenger Property    2015-02
     4         2019-02     ABE         Hazardous Materials Safety    2015-03

        count                          clean_cat clean_cat_status
     0      0         Hazardous Materials Safety         original
     1      0  Mishandling of Passenger Property         original
     2      0         Hazardous Materials Safety         original
     3      0  Mishandling of Passenger Property         original
     4      0         Hazardous Materials Safety         original
```

```
[5]: air.head()
```

```
[5]:   pdf_report_date airport year_month  count
     0         2019-02     ABE    2015-01      0
     1         2019-02     ABE    2015-02      0
     2         2019-02     ABE    2015-03      0
     3         2019-02     ABE    2015-04      0
     4         2019-02     ABE    2015-05      2
```

```
[11]: iata[iata['country_code']=='US'].groupby('region_name').count().head()
```

```
[11]:              country_code  iata  icao  airport  latitude  longitude
      region_name
      Alabama                28    28    27       28        28         28
      Alaska                331   331   200      331       331        331
      Arizona                48    48    34       48        48         48
      Arkansas               32    32    32       32        32         32
      California            148   148   130      148       148        148
```

```
[32]: for index,row in sub[['clean_cat','clean_subcat']].
        ↪groupby(['clean_cat','clean_subcat']).count().reset_index().iterrows():
          print(row['clean_cat'],"-",row['clean_subcat'],",")
          break
```

```
      Additional Information Required/Insufficient Information - EMAIL ONLY ,
```

```
[33]: sub[['clean_cat','clean_subcat']].groupby(['clean_cat','clean_subcat']).count().
        ↪shape
```

```
[33]: (179, 0)
```

```
[14]: temp = pd.merge(sub,iata,left_on='airport', right_on='iata')

      temp.shape
```

[14]: (489742, 18)

[38]:
```
smaller =␣
↪temp[['year_month','clean_cat','clean_subcat','count','airport_x','airport_y','country_code
smaller = smaller.rename(columns={
    'airport_x':'airport_code',
    'airport_y':'airport_name',
    'clean_cat':'category',
    'clean_subcat':'subcategory',
})
smaller.to_csv('complaints.csv')
```

[39]:
```
smaller.head()
```

[39]:
```
  year_month                           category  \
0    2015-01         Hazardous Materials Safety
1    2015-01  Mishandling of Passenger Property
2    2015-02         Hazardous Materials Safety
3    2015-02  Mishandling of Passenger Property
4    2015-03         Hazardous Materials Safety

                              subcategory  count airport_code  \
0                                 General      0          ABE
1  *Damaged/Missing Items--Checked Baggage      0          ABE
2                                 General      0          ABE
3  *Damaged/Missing Items--Checked Baggage      0          ABE
4                                 General      0          ABE

                        airport_name country_code   region_name  latitude  \
0  Lehigh Valley International Airport           US  Pennsylvania   40.6521
1  Lehigh Valley International Airport           US  Pennsylvania   40.6521
2  Lehigh Valley International Airport           US  Pennsylvania   40.6521
3  Lehigh Valley International Airport           US  Pennsylvania   40.6521
4  Lehigh Valley International Airport           US  Pennsylvania   40.6521

   longitude
0   -75.4408
1   -75.4408
2   -75.4408
3   -75.4408
4   -75.4408
```

[51]:
```
sorted = smaller[['category','count']].groupby('category').sum().reset_index().
↪sort_values('count',ascending=False)
top_9 = sorted.head(9)
total = sorted.sum()['count']
subtotal = top_9.sum()['count']
```

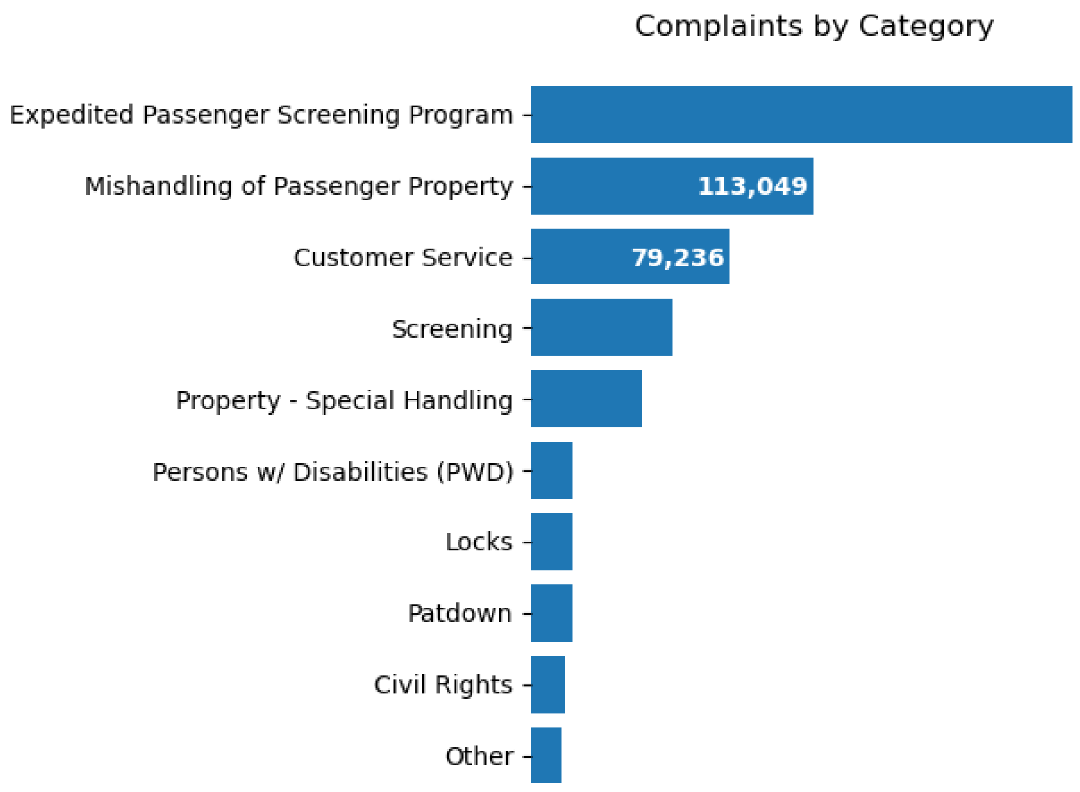
```
print("Top 9 account for ",str(subtotal/total*100),"%")
```

```
Top 9 account for  97.98938968391958 %
```

[62]:
```
keep = list(top_9['category'])
renames = {}
for category in list(sorted['category']):
    if category in keep:
        continue
    renames[category] = 'Other'
renamed = sorted.replace(renames)
top_10 = renamed.groupby('category').sum().sort_values('count',ascending=False).
 ↪reset_index()
```

[71]:
```
import matplotlib.pyplot as plt

categories = top_10['category']
values = top_10['count']

fig, ax = plt.subplots()
bars = ax.barh(categories, values)

for i, bar in enumerate(bars):
    if i>0 and i<3:
        width = bar.get_width()
        ax.text(
            width - 2000,                        # x-position just inside the␣
 ↪bar
            bar.get_y() + bar.get_height()/2 +.03, # y-position centered
            f"{width:,.0f}",                     # formatted label
            ha='right', va='center',            # align right and center
            fontsize=10, fontweight='bold',
            color='white'
        )
    elif i>2:
        continue
        width = bar.get_width()
        ax.text(
            width + 2000,                        # x-position just outside the␣
 ↪bar
            bar.get_y() + bar.get_height()/2 +.03, # y-position centered
            f"{width:,.0f}",                     # formatted label
            ha='left', va='center',             # align right and center
            fontsize=10, fontweight='bold',
            color='black'
        )
```

```
# Remove all borders (spines)
for spine in ax.spines.values():
    spine.set_visible(False)

ax.get_xaxis().set_visible(False)

plt.title('Complaints by Category')
plt.gca().invert_yaxis()  # Optional: puts the highest count at the top
plt.tight_layout()
plt.show()
```

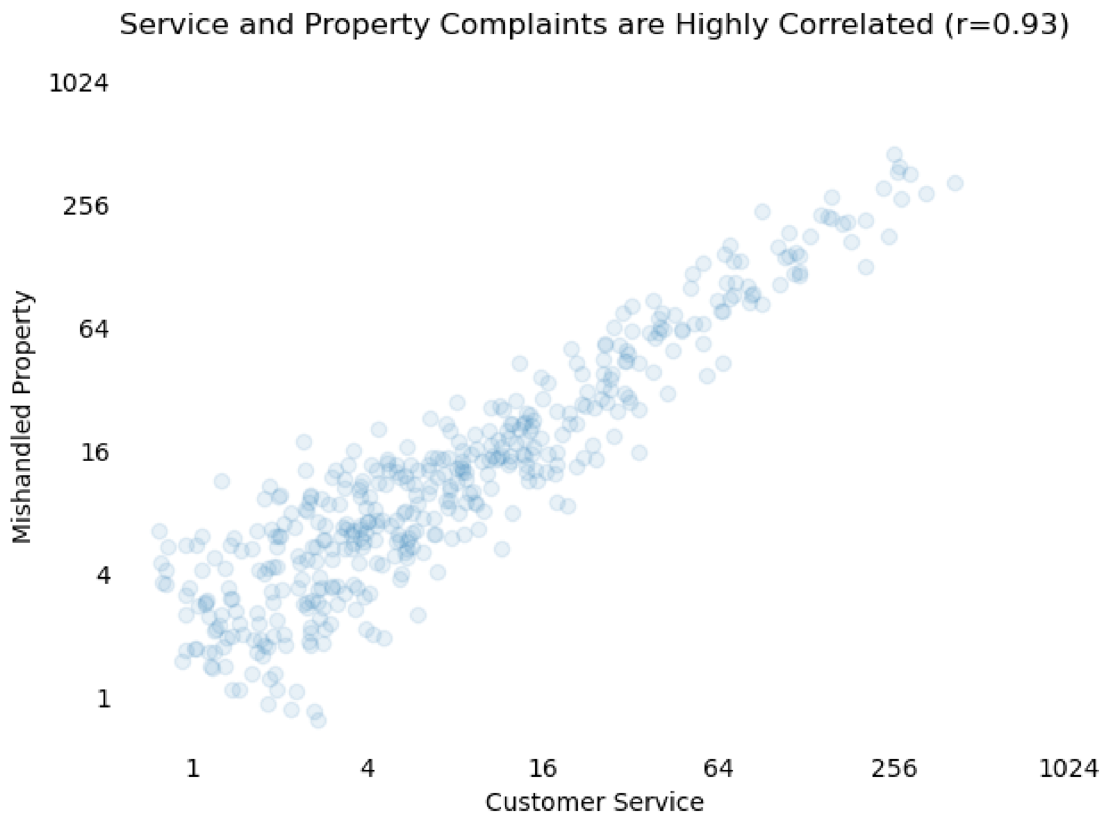


[86]:
```
focus = ['Mishandling of Passenger Property','Customer Service']
scatter = smaller[['category','airport_code','count']]
scatter = scatter[scatter['category'].isin(focus)].
  ↪groupby(['airport_code','category']).sum()
reshaped = scatter.pivot_table(index='airport_code', columns='category',␣
  ↪values='count', fill_value=0)
reshaped = reshaped.reset_index()
reshaped.sort_values('Mishandling of Passenger Property').head()
```

[86]:
```
category airport_code  Customer Service  Mishandling of Passenger Property
99                CVN               2.0                                0.0
292               OGD               2.0                                0.0
265               MGW               1.0                                0.0
146               FOE               2.0                                0.0
257               MCW               1.0                                0.0
```

[106]:
```
matrix = np.corrcoef(log_counts['Customer Service'],log_counts['Mishandling of␣
↪Passenger Property'])
complaint_r = correlation_matrix[0, 1]

import numpy as np

# Apply log to all columns except 'airport_code'
log_counts = reshaped.copy()
log_counts.iloc[:, 1:] = np.log(log_counts.iloc[:, 1:]+ 1)

jitter_strength = 0.4
fig, ax = plt.subplots()

# Apply jitter to both columns
x_jittered = log_counts['Customer Service'] + np.random.
↪uniform(-jitter_strength, jitter_strength, size=len(log_counts))
y_jittered = log_counts['Mishandling of Passenger Property'] + np.random.
↪uniform(-jitter_strength, jitter_strength, size=len(log_counts))
ax.scatter(x_jittered, y_jittered, alpha=0.1)

yticks = ax.get_yticks()[1:7]
ytick_labels = [int(2**tick) for tick in yticks]
ax.set_yticks(yticks)
ax.set_yticklabels(ytick_labels)
ax.tick_params(axis='y', which='both', length=0)

xticks = ax.get_xticks()[1:7]
xtick_labels = [int(2**tick) for tick in xticks]
ax.set_xticks(yticks)
ax.set_xticklabels(ytick_labels)
ax.tick_params(axis='x', which='both', length=0)


# Optional: Add axis labels and a title
ax.set_xlabel('Customer Service')
ax.set_ylabel('Mishandled Property')
ax.set_title(f'Service and Property Complaints are Highly Correlated␣
↪(r={complaint_r:.2f})',y=1.05)

# Remove all borders (spines)
```

```
for spine in ax.spines.values():
    spine.set_visible(False)

plt.tight_layout()
plt.show()
```



[145]:
```
box = smaller[['category','airport_code','count']].replace(renames)
box = box.replace({
    'Expedited Passenger Screening Program':'Screening Program',
    'Mishandling of Passenger Property':'Mishandled Property',
    'Property - Special Handling':'Special Handling',
    'Persons w/ Disabilities (PWD)':'Disabilities',
})
box.head()
```

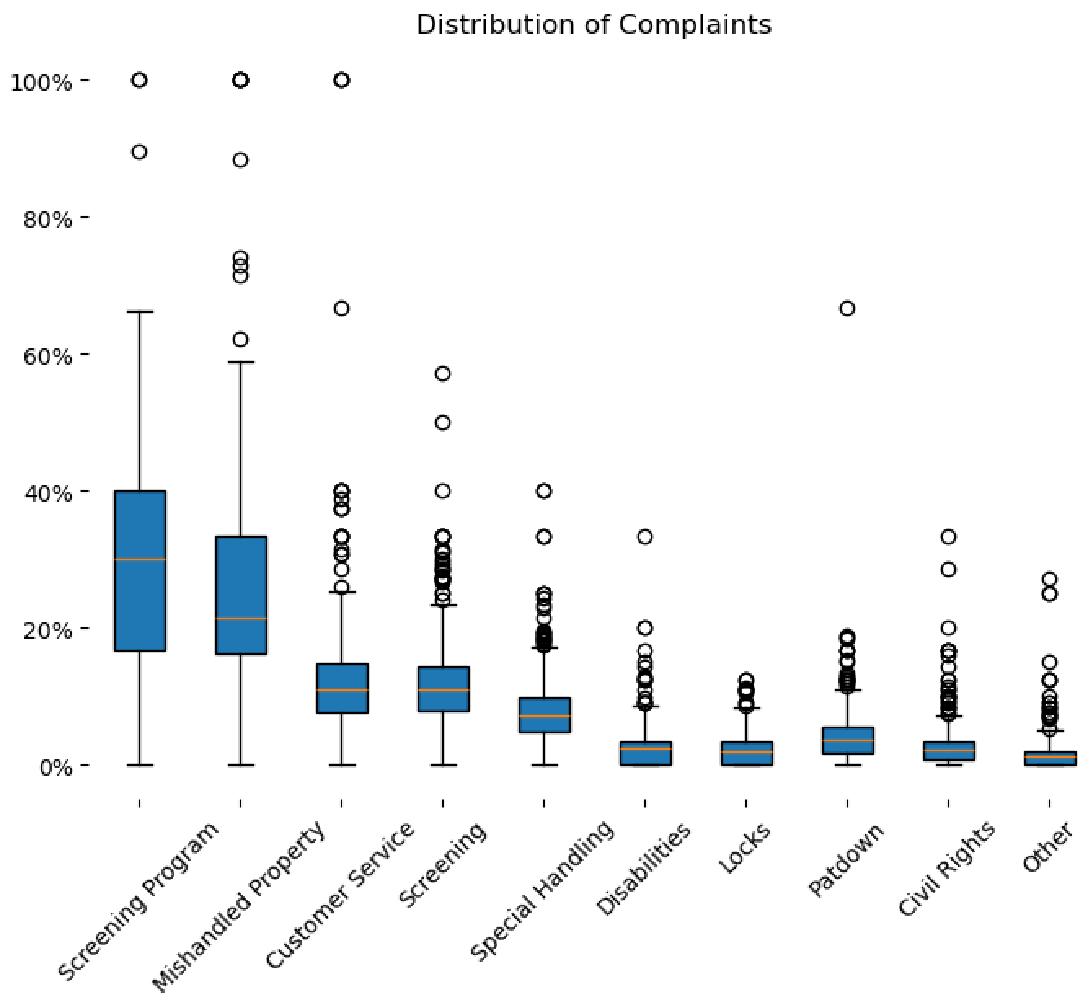
[145]:
```
             category airport_code  count
0               Other          ABE      0
1  Mishandled Property          ABE      0
2               Other          ABE      0
3  Mishandled Property          ABE      0
```

```
4                 Other          ABE      0
```

[184]:
```
box = box.groupby(['airport_code','category']).sum().reset_index()
categories = list(box[['category','count']].groupby('category').sum().
 ↪reset_index().sort_values('count',ascending=False)['category'])
reshaped = box.pivot_table(index='airport_code', columns='category',␣
 ↪values='count', fill_value=0)
reshaped = reshaped.reset_index()
reshaped['total'] = reshaped[categories].sum(axis=1)

normalized = reshaped[categories].div(reshaped['total'], axis=0)

# Optional: Add normalized columns back to original DataFrame
for col in categories:
    reshaped[col] = normalized[col]

cleaned = reshaped[categories].dropna()

cleaned.head()
```

[184]:
```
category  Screening Program  Mishandled Property  Customer Service  Screening  \
0                  0.533477             0.153348          0.071274   0.090713
1                  0.266055             0.165138          0.174312   0.146789
2                  0.340194             0.264730          0.086764   0.094431
3                  0.233333             0.300000          0.050000   0.116667
4                  0.144068             0.474576          0.059322   0.067797

category  Special Handling  Disabilities     Locks   Patdown  Civil Rights  \
0                 0.041037      0.019438  0.023758  0.034557      0.025918
1                 0.073394      0.036697  0.018349  0.027523      0.045872
2                 0.079903      0.027845  0.037934  0.031073      0.025827
3                 0.133333      0.016667  0.050000  0.016667      0.050000
4                 0.101695      0.042373  0.033898  0.025424      0.016949

category     Other
0         0.006479
1         0.045872
2         0.011299
3         0.033333
4         0.033898
```

[152]:
```
from matplotlib.ticker import FuncFormatter

fig, ax = plt.subplots(figsize=(8, 6))  # Create figure and axis together

ax.boxplot([cleaned[cat] for cat in categories],
            labels=categories,
            patch_artist=True)

# Add titles and labels
ax.set_title('Distribution of Complaints')
ax.grid(True)

# Remove all borders (spines)
ax.grid(False)
for spine in ax.spines.values():
    spine.set_visible(False)

ax.yaxis.set_major_formatter(FuncFormatter(lambda y, _: f'{y:.0%}'))

# Show plot
plt.xticks(rotation=45)

plt.show()
```



[162]:
```
map = smaller[smaller['country_code']=='US'][['region_name','category','count']]
us_state_abbrev = {
    'Alabama': 'AL', 'Alaska': 'AK', 'Arizona': 'AZ', 'Arkansas': 'AR',
    'California': 'CA', 'Colorado': 'CO', 'Connecticut': 'CT', 'Delaware': 'DE',
    'Florida': 'FL', 'Georgia': 'GA', 'Hawaii': 'HI', 'Idaho': 'ID',
    'Illinois': 'IL', 'Indiana': 'IN', 'Iowa': 'IA', 'Kansas': 'KS',
    'Kentucky': 'KY', 'Louisiana': 'LA', 'Maine': 'ME', 'Maryland': 'MD',
    'Massachusetts': 'MA', 'Michigan': 'MI', 'Minnesota': 'MN', 'Mississippi':␣
  ↪'MS',
    'Missouri': 'MO', 'Montana': 'MT', 'Nebraska': 'NE', 'Nevada': 'NV',
    'New Hampshire': 'NH', 'New Jersey': 'NJ', 'New Mexico': 'NM', 'New York':␣
  ↪'NY',
    'North Carolina': 'NC', 'North Dakota': 'ND', 'Ohio': 'OH', 'Oklahoma':␣
  ↪'OK',
    'Oregon': 'OR', 'Pennsylvania': 'PA', 'Rhode Island': 'RI', 'South␣
  ↪Carolina': 'SC',
    'South Dakota': 'SD', 'Tennessee': 'TN', 'Texas': 'TX', 'Utah': 'UT',
    'Vermont': 'VT', 'Virginia': 'VA', 'Washington': 'WA', 'West Virginia':␣
  ↪'WV',
    'Wisconsin': 'WI', 'Wyoming': 'WY'
}
map['state'] = map['region_name'].map(us_state_abbrev)
totals = map[['state','count']].groupby('state').sum().reset_index()
service = map[map['category']=='Customer Service'][['state','count']].
  ↪groupby('state').sum().reset_index()
merged = pd.merge(totals,service,on='state')
merged = merged.rename(columns={
    'count_x':'total',
    'count_y':'service',
})
merged['ratio']= merged['service']/merged['total']*100
```

[163]:
```
import plotly.offline as pyo
import plotly.graph_objs as go
# Set notebook mode to work in offline
pyo.init_notebook_mode()
import plotly.graph_objects as go

import pandas as pd

fig = go.Figure(data=go.Choropleth(
    locations=merged['state'], # Spatial coordinates
    z = merged['ratio'].astype(float), # Data to be color-coded
    locationmode = 'USA-states', # set of locations match entries in `locations`
    colorscale = 'Reds',
    colorbar_title = "Percent Service Complaints",
))

fig.update_layout(
    title_text = 'Customer Service Complaint Percentage by State',
    geo_scope='usa', # limite map scope to USA
)

fig.show()
```

[188]:
```
heat =␣
 ↪smaller[smaller['country_code']=='US'][['year_month','category','count']].
 ↪replace(renames)
heat = heat.replace({
    'Expedited Passenger Screening Program':'Screening Program',
    'Mishandling of Passenger Property':'Mishandled Property',
    'Property - Special Handling':'Special Handling',
    'Persons w/ Disabilities (PWD)':'Disabilities',
})
heat['month'] = pd.to_datetime(heat['year_month']).dt.strftime('%B')
heat.head()
```

[188]:
```
  year_month             category  count     month
0    2015-01                Other      0   January
1    2015-01  Mishandled Property      0   January
2    2015-02                Other      0  February
3    2015-02  Mishandled Property      0  February
4    2015-03                Other      0     March
```

[194]:
```
heat = heat[['month','category','count']].groupby(['month','category']).sum().
 ↪reset_index()
months = list(heat['month'].unique())
heatmap_data = heat.pivot_table(index='category', columns='month',␣
 ↪values='count', aggfunc='sum').reset_index()
heatmap_data['category'] = pd.Categorical(heatmap_data['category'],␣
 ↪categories=categories, ordered=True)
heatmap_data = heatmap_data.sort_values('category')
heatmap_data
```

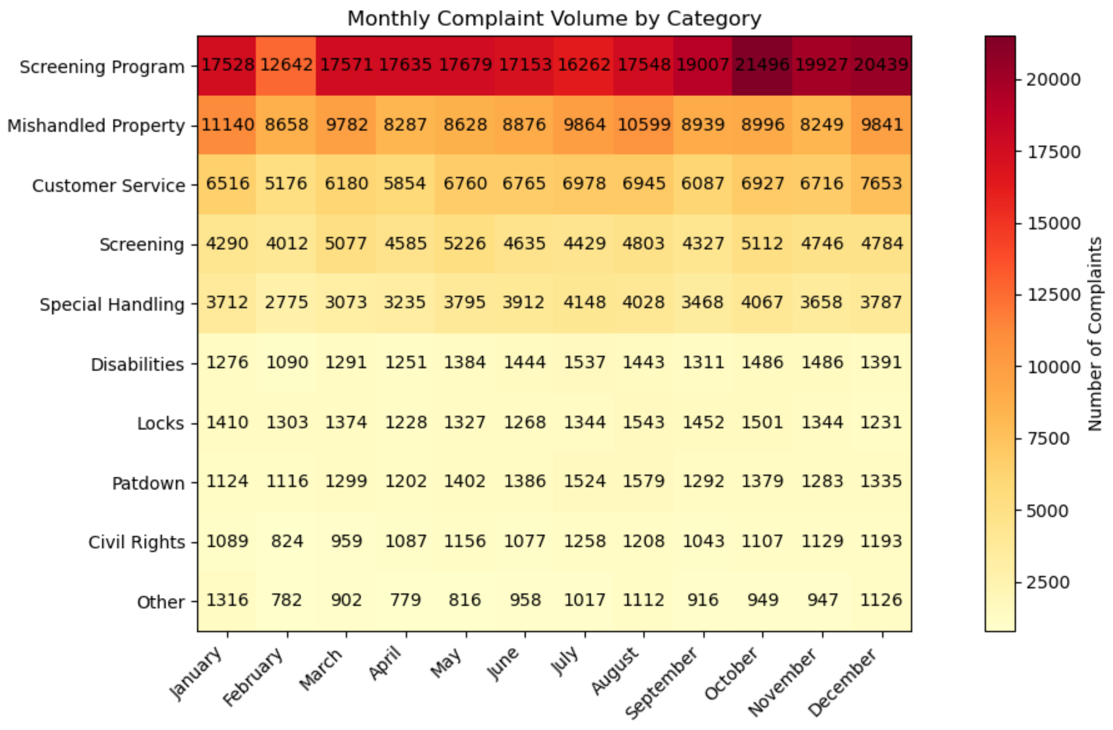
[194]:
```
month             category  April  August  December  February  January   July  \
8        Screening Program  17635   17548     20439     12642    17528  16262
4      Mishandled Property   8287   10599      9841      8658    11140   9864
1         Customer Service   5854    6945      7653      5176     6516   6978
7                Screening   4585    4803      4784      4012     4290   4429
9         Special Handling   3235    4028      3787      2775     3712   4148
2             Disabilities   1251    1443      1391      1090     1276   1537
3                    Locks   1228    1543      1231      1303     1410   1344
```

```
6                 Patdown   1202    1579      1335      1116      1124   1524
0            Civil Rights   1087    1208      1193       824      1089   1258
5                   Other    779    1112      1126       782      1316   1017

month    June  March    May  November  October  September
8       17153  17571  17679     19927    21496      19007
4        8876   9782   8628      8249     8996       8939
1        6765   6180   6760      6716     6927       6087
7        4635   5077   5226      4746     5112       4327
9        3912   3073   3795      3658     4067       3468
2        1444   1291   1384      1486     1486       1311
3        1268   1374   1327      1344     1501       1452
6        1386   1299   1402      1283     1379       1292
0        1077    959   1156      1129     1107       1043
5         958    902    816       947      949        916
```

[187]:
```
#heatmap_data[0,0]
type(heatmap_data)
print(categories)
```

```
['Screening Program', 'Mishandled Property', 'Customer Service', 'Screening',
'Special Handling', 'Disabilities', 'Locks', 'Patdown', 'Civil Rights', 'Other']
```

[195]:
```
import matplotlib.pyplot as plt
import numpy as np

month_order = ['January', 'February', 'March', 'April', 'May', 'June',
               'July', 'August', 'September', 'October', 'November', 'December']

# Reorder the columns in your DataFrame
heatmap_data = heatmap_data[month_order]

# Create the heatmap
fig, ax = plt.subplots(figsize=(12, 6))
heatmap = ax.imshow(heatmap_data, cmap='YlOrRd')

# Add labels
ax.set_xticks(np.arange(len(months)))
ax.set_yticks(np.arange(len(categories)))
ax.set_xticklabels(month_order)
ax.set_yticklabels(categories)

# Rotate x-axis labels
plt.setp(ax.get_xticklabels(), rotation=45, ha='right')

# Add values to each cell
for i in range(len(categories)):
    for j in range(len(months)):
        ax.text(j, i, heatmap_data.iloc[i, j], ha='center', va='center',␣
  ↪color='black')

# Add colorbar
cbar = plt.colorbar(heatmap)
cbar.set_label('Number of Complaints')

# Title
ax.set_title('Monthly Complaint Volume by Category')

plt.tight_layout()
plt.show()
```

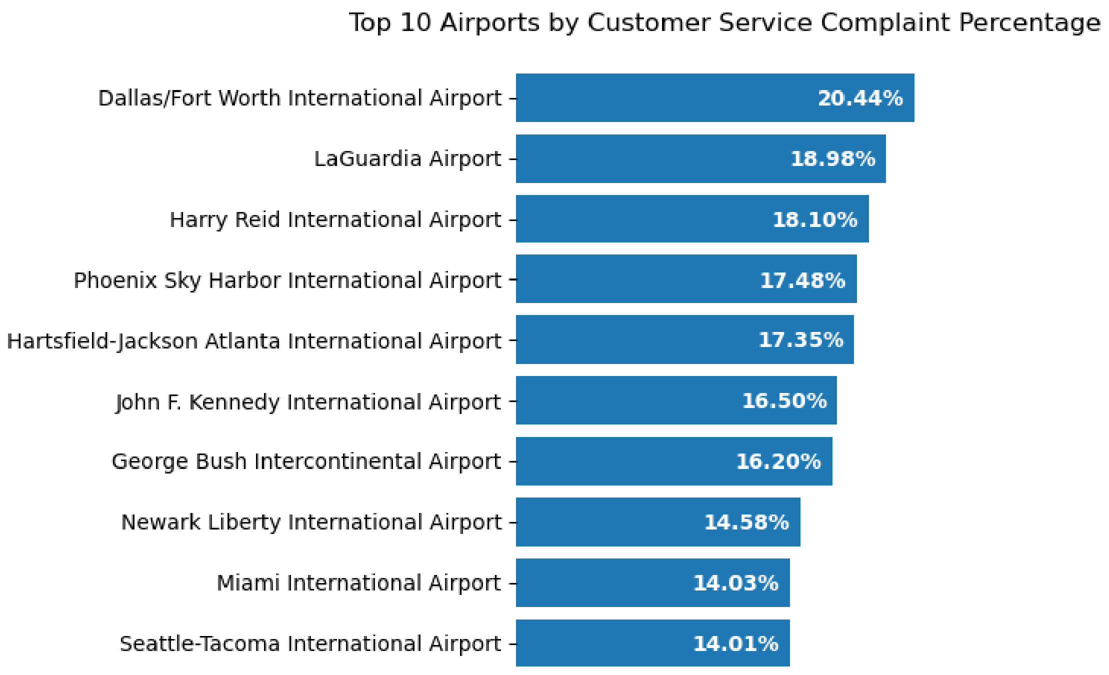


```
[208]: airports =␣
  ↪smaller[smaller['country_code']=='US'][['airport_name','region_name','category','count']]
print(airports.head())

totals = airports[['airport_name','count']].groupby('airport_name').sum().
  ↪reset_index()
service = airports[airports['category']=='Customer␣
  ↪Service'][['airport_name','count']].groupby('airport_name').sum().
  ↪reset_index()
merged = pd.merge(totals,service,on='airport_name')
merged = merged.rename(columns={
    'count_x':'total',
    'count_y':'service',
})
merged['ratio']= merged['service']/merged['total']*100
merged = merged[merged['total']>10000].sort_values('ratio',ascending=False)
top_10 = merged.head(10)
```

```
                          airport_name   region_name  \
0  Lehigh Valley International Airport  Pennsylvania
1  Lehigh Valley International Airport  Pennsylvania
2  Lehigh Valley International Airport  Pennsylvania
3  Lehigh Valley International Airport  Pennsylvania
4  Lehigh Valley International Airport  Pennsylvania

                            category  count
0         Hazardous Materials Safety      0
1  Mishandling of Passenger Property      0
2         Hazardous Materials Safety      0
3  Mishandling of Passenger Property      0
4         Hazardous Materials Safety      0
```

[220]:
```
import matplotlib.pyplot as plt

categories = top_10['airport_name']
values = top_10['ratio']

fig, ax = plt.subplots()
bars = ax.barh(categories, values)

for i, bar in enumerate(bars):
        width = bar.get_width()
        ax.text(
            width-0.5,                          # x-position just inside the bar
            bar.get_y() + bar.get_height()/2 +.03, # y-position centered
            f"{width:,.2f}%",                    # formatted label
            ha='right', va='center',            # align right and center
            fontsize=10, fontweight='bold',
            color='white'
        )


# Remove all borders (spines)
for spine in ax.spines.values():
    spine.set_visible(False)

ax.get_xaxis().set_visible(False)
```

```
plt.title('Top 10 Airports by Customer Service Complaint Percentage')
plt.gca().invert_yaxis()  # Optional: puts the highest count at the top
plt.tight_layout()
plt.show()
```



```
[ ]:
```