SYRCityline Request Analysis

TIA & LIZ

SYRCityline Data

- Our dataset is a collection of user reported service requests submitted through Syracuse's Cityline program (https://seeclickfix.com/Syracuse)
- SYRCityline is a tool that allows city residents to report non-emergency service requests such as public litter, illegal trash setouts and street and sewage concerns.
- Our dataset consisted of 23 columns and 68,505 rows

Objective

* Use the data to determine the effectiveness of this tool. We hope to answer the question: does the SYRCityline application perform as it was intended to and if not, why and how can it be improved. We will be determining effectiveness by resolution time.

SYRCityline Dataset

- **Summary** - String that users select to categorize the nature of their complaint. Can be either Large or Bulk Items, Illegal Setouts, Sewer Back-ups, Weekly Trash Pickup, Large or Bulk Items - Skipped Pickup, Home & Building Maintenance, Sewer-related Concerns, Recycling, Other Housing & Property Maintenance Concern, Street Lights, or Other.
- **Latitude** - Latitude GPS coordinate where the address is.
- **Longitude** - Longitude GPS coordinate where the address is.
- **Export tagged places** - Which quadrant of the city is this address matched to (Northeast, Southeast, Northwest, or Southwest).
- **Created at local** - When this complaint or service was requested. (This is in the format of MM/DD/YYYY - HH:MM(AM/PM)).

SYRCityline Dataset

- **Agency Name** - What type of City Department was this complaint assigned to. These include:
- Streets, Sidewalks & Transportation
- Garbage, Recycling & Graffiti
- Housing & Property Maintenance
- Feedback to the City
- Parking & Vehicles
- Green Spaces, Trees & Public Utilities
- Water & Sewage
- Animals

SYRCityline Dataset

- **Acknowledged at local** - When this complaint or service request was acknowledged by the City department. (This is in the format of MM/DD/YYYY - HH:MM(AM/PM)).
- **Closed at local** - When this complaint or service request was marked as being closed by the City department. (This is in the format of MM/DD/YYYY - HH:MM(AM/PM)).
- **Minutes to acknowledged** - The amount of time, in minutes, after it was Created at Local to being marked Acknowledged at local.
- **Minutes to closed** - The amount of time, in minutes, after service request was created at local to when it was marked as Closed at local.
- **Assignee name** - Which city Department was assigned to this request.

SYRCityline Dataset

- **SLA Limit** - This is the limit assigned by the City of Syracuse, that puts a limit on how a request can stay in the list of tasks untouched. That amount of time, in hours, SeeClickFix will forward the request to the department head as well as an administrator to help ensure that requests are addressed in a timely manner.
- **Report source** - How this service request was obtained: Either Web-Mobile, iPhone, Portal, Web-Desktop, Android, or Request Form.*

SYRCityline Dataset

- **ID** - Identification number assigned to each incident
- **Rating** - The number of followers on the Request in SeeClickFix.
- **Address** - Address of the service request or complaint, provided by the community member.
- **Description** - Write up of the service request or complaint, provided by the community member.
- **Request Type** – Request type ID number
- **URL** - Unique website address (url) that the complaint as well as comments from the City personnel can be viewed at.

SYRCityline Dataset

- **Category** – The way a request is categorized. This can be Potholes, Large or Bulk Items, Water-related Concerns, Home & Building Maintenance, Street Lights, Weekly Trash Pickup, Public Trash Can, Yard Waste, Report Litter on Private Land, among other categories.

Data Cleaning and Transforming

- **Adding columns:**
  - Split the date & time data into their own columns (*Created at local, Closed at local, & Acknowledged at local*)
  - *SLA in mins, SLA in days, Hours to closed, Days to closed*
  - Y/N columns for *SLA met, Assigned, & Acknowledged*
  - *Closed Category*, which contains bins to categorize resolution time (i.e. Two weeks, Less than a day, etc.)

Data Cleaning and Transforming

- **Dealing with NAS:**
  - Found over 56,099 lines containing NAs, but chose to keep most of them in the dataset
  - NAs that were removed came from *Lat & Lng* (76), *Export tagged places* (76), and all *SLA* related NAs(23)
  - Cleaned up the data in the *Export tagged places* column to keep data uniform. Ensured all data in these columns within data set fit within the 4 main classification: NW, NE, SE & SW. We were ablet o update most, those that could not be updated were converted into NAs and dropped

Data Cleaning and Transforming

- The last step was to select all the columns we planned to use for our analysis
- After cleaning and transforming the data, our dataset consisted of 24 rows and 68,406 rows.

Exploratory

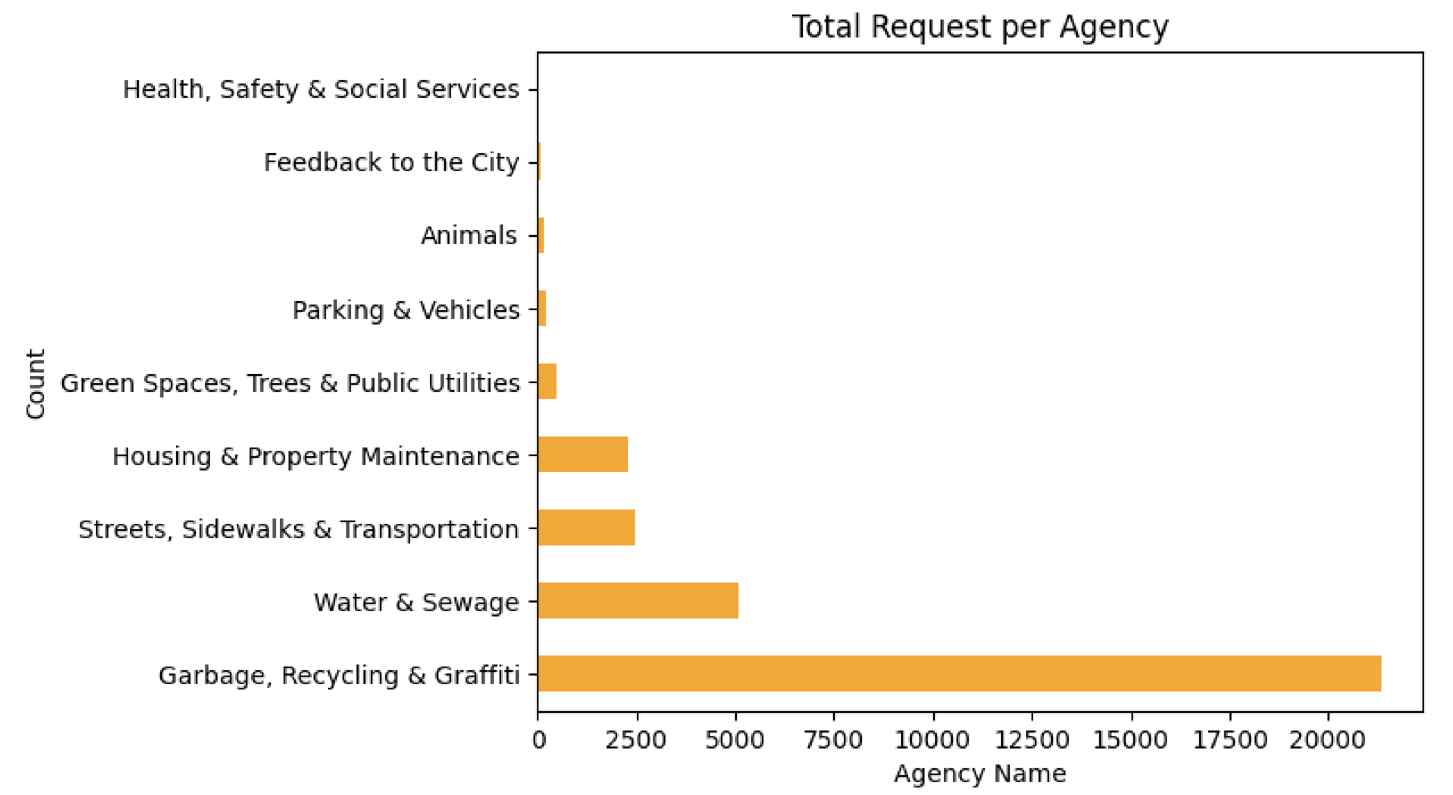
- **What we learned**
  - There are 56 potential assignees, and most are allocated to ***Streets, Sidewalks & Transportation***, ***Housing & Property Maintenance*** and ***Garbage, Recycling & Graffiti***

Category	# of Potential Asignees
Animals	9
Feedback to the City	23
Garbage, Recycling & Graffiti	31
Green Spaces, Trees & Public Utilities	25
Health, Safety & Social Services	8
Housing & Property Maintenance	32
Parking & Vehicles	11
Streets, Sidewalks & Transportation	36
Water & Sewage	8

```
#Count of assignee per agency

# Use groupby and count to Total count of Assignees per Agency name. Used nunique to prevent duplicates
count_assignee= cityline22.groupby('Agency_Name')['Assignee_name'].nunique().reset_index()

# Rename the columns
count_assignee.columns = ['Category', '# of Potential Asignees']

print(count_assignee)
```

Exploratory

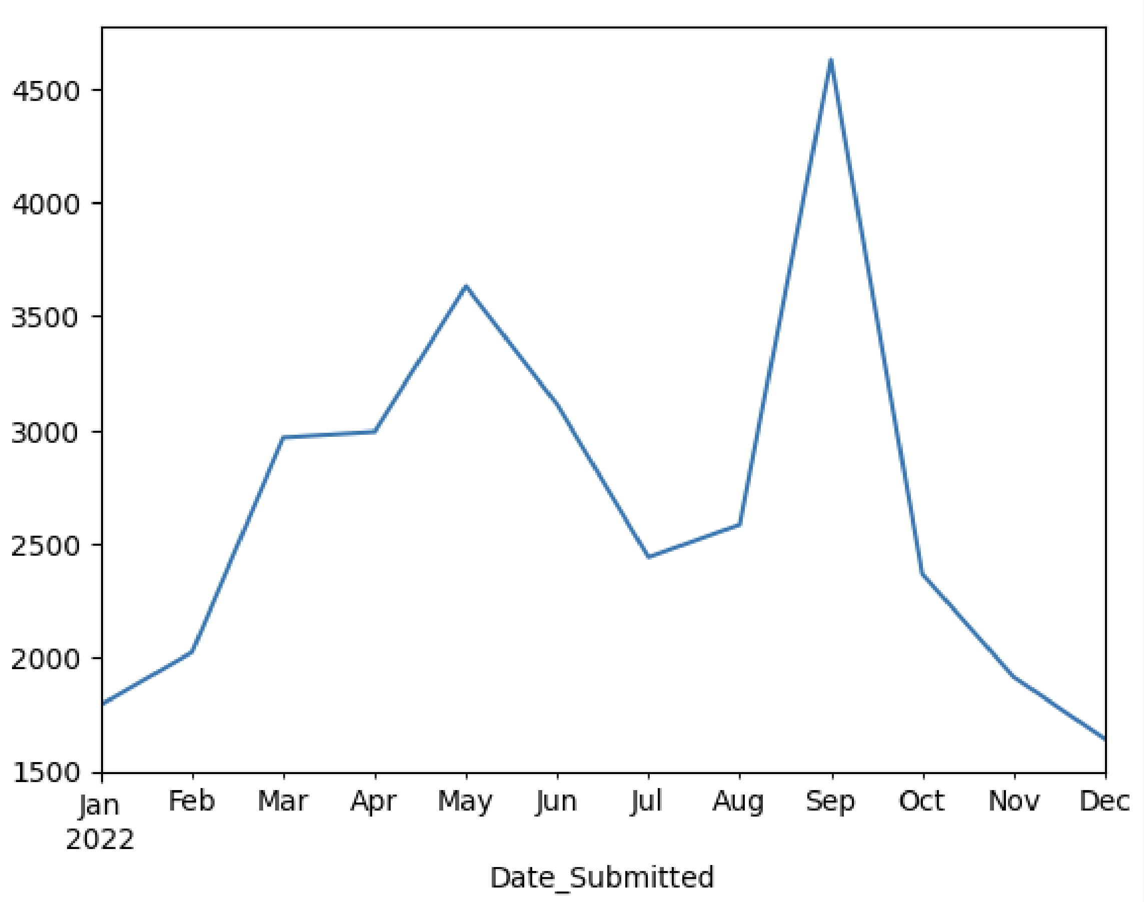
- **What we learned**
  - There are 9 agencies that handle service requests
  - Of the 9 agencies, the Garbage, Recycling & Graffiti Dept. Received the most service requests




```
#Plot the Agency_Name data:Most complaints to to the Garbage, Recycling & Graffiti Dept.
cityline22['Agency_Name'].value_counts().plot.barh(color="orange") #Count all Agency_Names in cityline22 and plot

#Adding labels and titles
plt.title('Total Request per Agency') #Add a title
plt.xlabel('Agency Name')
plt.ylabel('Count')
plt.figure(figsize=(18, 9)) #Set size of plot 18w/9h
plt.show()
```

Exploratory

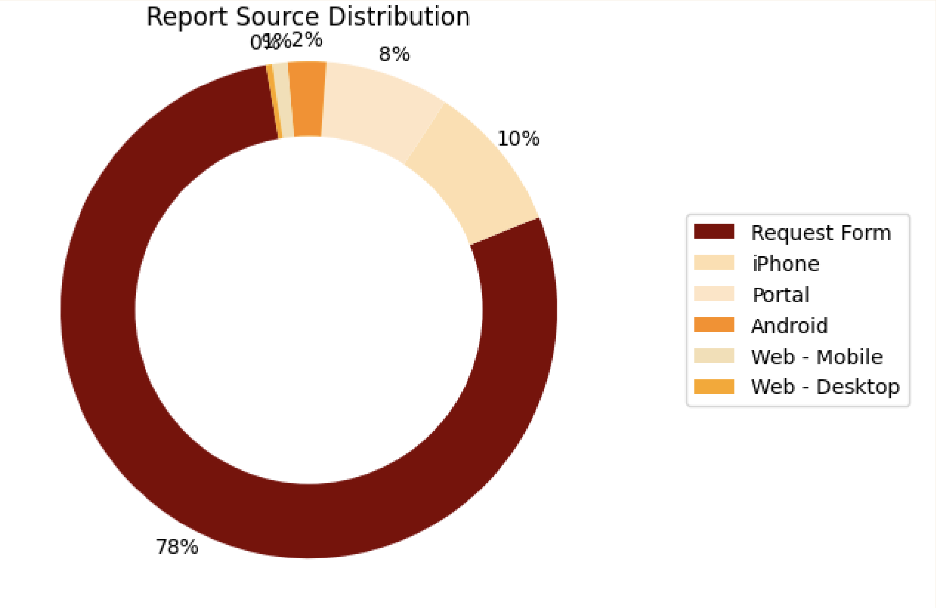
- **What we learned**
  - September received the most requests and requests declined from there




```
#Count of issues per month:
cityline22.set_index("Date_Submitted", inplace=True) #Setting "Date submitted as index
monthly_issues = cityline22['Agency_Name'].resample('M').count()  #resample, count issues per month
monthly_issues.sort_index(inplace=True) #Sort data to keep in order

#Plot data by month
monthly_issues.plot.line()
```

Exploratory

- **What we learned**
  - Majority of requests get submitted by paper form




```python
# Calculate percentages
formPercents = cityline22['Report_Source'].value_counts(normalize=True) #Get the count of each Report source, then normalize

# Create a pie chart with % labels(using autopct) outside the pie (using pctdistance and labeldistance), and specific color scheme (using colors)
plt.pie(formPercents, autopct='%1.0f%%', startangle=100, pctdistance=1.09, colors=['maroon', 'navajowhite', 'bisque', 'darkorange','wheat','orange'], labeldistance=5.7)

#Used to make pie chart a donut chart
centre_circle = plt.Circle((0, 0), 0.70, fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

# Equal aspect ratio ensures that pie is drawn as a circle
plt.axis('equal')

# Maintain shape and location of plot
plt.legend(labels=formPercents.index, loc='center left', bbox_to_anchor=(1, 0.5), fontsize=10)

# Add title
plt.title('Report Source Distribution')

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

Is the city performing well? Are they fulfilling their SLAs?

- **What we learned**
  - 76% or service request met SLA requirements, while 24% not.
  - These results are consistent month to month

Ratio of SLA met requests for year

```
cityline22.Sla_met.value_counts(normalize=True)*100
```

```
Sla_met
Y    76.152023
N    23.847977
Name: proportion, dtype: float64
```

Sla_met	N	Y
Date_Resolved		
2022-01-31	0.181597	0.818403
2022-02-28	0.276467	0.723533
2022-03-31	0.287399	0.712601
2022-04-30	0.323050	0.676950
2022-05-31	0.264934	0.735066
2022-06-30	0.317256	0.682744
2022-07-31	0.302317	0.697683
2022-08-31	0.209356	0.790644
2022-09-30	0.092117	0.907883
2022-10-31	0.154817	0.845183
2022-11-30	0.201393	0.798607
2022-12-31	0.248481	0.751519

```
#Get Ratio of SLA met requests by moth
cityline22.set_index("Date_Resolved", inplace=True) #set Date resolved column as index

#group and count data sla met and agency name by month. Use unstack to remove data from index and fillna to take care of any nas in data
monthly_issues = cityline22.groupby([pd.Grouper(freq='M'), 'Sla_met'])['Agency_Name'].count().unstack().fillna(0)

monthly_issues

#Make values percentages
percentSLA = monthly_issues.div(monthly_issues.sum(axis=1), axis=0)
percentSLA
```

What agencies struggle the most?

- **What we learned**
  - A closer look at the data shows the agencies that seem to struggle the most with SLA are ***Feedback to the City and Health, Safety & Social Services*** and ***Parking & Vehicles***

Agency_Name	Y	N	Total	Y_Percentage	N_Percentage
Animals	78	74	152	51.315789	48.684211
Feedback to the City	27	60	87	31.034483	68.965517
Garbage, Recycling & Graffiti	16479	4843	21322	77.286371	22.713629
Green Spaces, Trees & Public Utilities	306	158	464	65.948276	34.051724
Health, Safety & Social Services	11	15	26	42.307692	57.692308
Housing & Property Maintenance	1502	777	2279	65.906099	34.093901
Parking & Vehicles	100	97	197	50.761421	49.238579
Streets, Sidewalks & Transportation	1343	1110	2453	54.749287	45.250713
Water & Sewage	4579	515	5094	89.890067	10.109933

```
#SLA ratio by agency
#Create seperate datas of "Y" and "N" sla-met status
Yes = cityline22[cityline22['Sla_met'] =='Y']
No = cityline22[cityline22['Sla_met'] =='N']

#Get totals
y = Yes.Agency_Name.value_counts()
n = No.Agency_Name.value_counts()

#Add data back into a single dataframe
agencySLA = pd.DataFrame({ 'Y' : y, 'N' : n, })

# Calculate the percentage of 'Y' and 'N' values for each agency row-wise
agencySLA['Total'] = agencySLA.sum(axis=1)  # Calculate total counts for each row
agencySLA['Y_Percentage'] = (agencySLA['Y'] / agencySLA['Total']) * 100
agencySLA['N_Percentage'] = (agencySLA['N'] / agencySLA['Total']) * 100

agencySLA
```

Does acknowledgement status affect an agencies resolution time?

- **What we learned**
  - Whether something is acknowledged makes a big difference, likely these take more time to resolve

Agency Name	Acknowleged	Average Days to Closed
Animals	N	5.3
Animals	Y	8.7
Feedback to the City	N	9.7
Feedback to the City	Y	28.4
Garbage, Recycling & Graffiti	N	1.7
Garbage, Recycling & Graffiti	Y	4.3
Green Spaces, Trees & Public Utilities	N	9.5
Green Spaces, Trees & Public Utilities	Y	29.5
Health, Safety & Social Services	N	3.8
Health, Safety & Social Services	Y	31.0
Housing & Property Maintenance	N	5.1
Housing & Property Maintenance	Y	12.2
Parking & Vehicles	N	8.5
Parking & Vehicles	Y	14.2
Streets, Sidewalks & Transportation	N	5.9
Streets, Sidewalks & Transportation	Y	21.8
Water & Sewage	N	0.2
Water & Sewage	Y	12.6

```
#DOES ACKNOWLEDGEMENT MATTER? - IT APPEARS THAT IT DOES

#Group by calculate the mean of Days to closed
grouped_df3 = cityline22.groupby(['Agency_Name', 'Acknowledged'])['Days_to_closed'].mean().reset_index()

# Round the mean values to the nearest tenth
grouped_df3.columns = ['Agency Name', 'Acknowleged','Average Days to Closed']
grouped_df3['Average Days to Closed'] = grouped_df3['Average Days to Closed'].round(1)

print(grouped_df3)
```

Recommendations

- Update SLA limits and categories
  - Difficult to compare SLA when the categories themselves can contain multiple limits
  - Parsing limits will help distinguish requests that will take more time over others
  - Information is possibly presented different depending on request form causing minor differences
- Suggest estimate time for completion
  - It did not appear that information was provided to residents once a request was submitted

Issues & Continued Work

- Possible Model Prediction for completion times
  - Help ensure tickets are being completed on time/sooner
- Consider what information may be correlated together
- Additional column to identify assigned by individual or team