CSCI 2000 Final Project

Analysis of Toronto Transit Commission Riderships

December 11th, 2018

Overview

TTC was awarded American Public Transportation Association's (APTA) award for Transit System of the Year in the past, the main objective of this project is to analyze what makes TTC is reliable as a transit system.

Goals

1. **How close the stops from one to another and how frequent is it coming?**
2. **Who are the passengers?**
3. **How far do TTC buses travel?**
4. **Fare, accessibility, and other factors that affect ridership rate?**

Data and Methods

This analysis will be conducted using Python with extension of using Numpy, Pandas API, Matplotlib, and folium. Using groupby, sorting, plotting, slicing, and other similar methods in order to create clean dataset and visualize end result of the analysists.

Most datasets are taken from Toronto Open Data Catalogue:

1. TTC Fare
2. TTC Routes and Schedules
3. TTC Ridership Analysis
4. TTC Passenger Ride Peak
5. TTC Passenger Ride Non-Peak
6. TTC Average Weekday Ridership
7. TTC Bus Delay Data
8. TTC Streetcar Delay Data
9. TTC Subway Delay Data

Analysis and Results

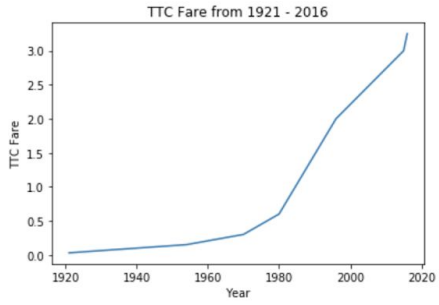
1. TTC Fare

FACTS: Here is data of TTC Fare over the years

What a big jump, eh?

```
In [97]: year = [1921, 1954, 1970, 1980, 1996, 2015, 2016]
         fare = [0.03, 0.15, 0.30, 0.60, 2, 3, 3.25]
         plt.plot(year, fare)
         plt.xlabel("Year")
         plt.ylabel("TTC Fare")
         plt.title("TTC Fare from 1921 - 2016")
```

```
Out[97]: Text(0.5,1,'TTC Fare from 1921 - 2016')
```



2. TTC Stops and Routes

Cleanup datasets and then merge them. Below is the final look on the merged data

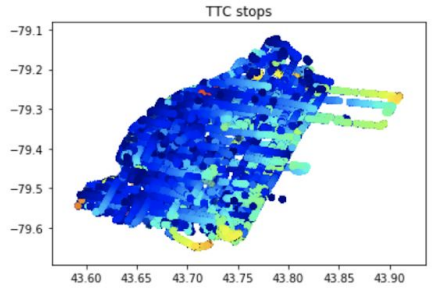
TTC Stop and Routes

Plot scatter plot of all listed stops and routes

`Note` : *Does the shape look familiar?*

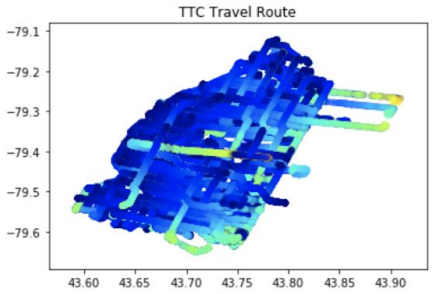
In [15]:
```
xs = route_trip_stop["stop_lat"]
ys = route_trip_stop["stop_lon"]
sz = route_trip_stop["stop_sequence"]
cm = plt.get_cmap('jet')
plt.title("TTC stops")
plt.scatter(xs, ys, c = sz, cmap = cm, alpha=0.3, edgecolors='None')
```

Out[15]: <matplotlib.collections.PathCollection at 0x1a4fe90cf8>



In [16]:
```
xs = shapes["shape_pt_lat"]
ys = shapes["shape_pt_lon"]
sz = shapes["shape_dist_traveled"]
cm = plt.get_cmap('jet')
plt.title("TTC Travel Route")
plt.scatter(xs, ys, c = sz, cmap = cm, alpha=0.3, edgecolors='None')
```

Out[16]: <matplotlib.collections.PathCollection at 0x1a5002f828>



Top 20 TTC Route

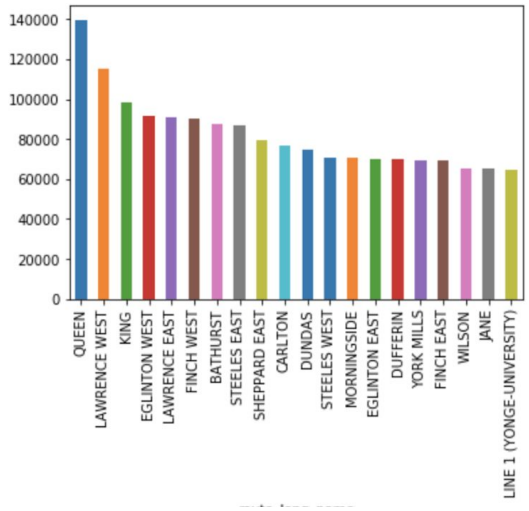
Finding which route that TTC has their most service offer

Bar Plot Top 20 Routes

Breaking down and compare how much different each route

```
In [22]: top20.plot.bar(x = 'route_long_name', y = 'route_id', legend = False)
```

```
Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x1a50062b38>
```



Did further analysis to get where in Toronto these stops are:

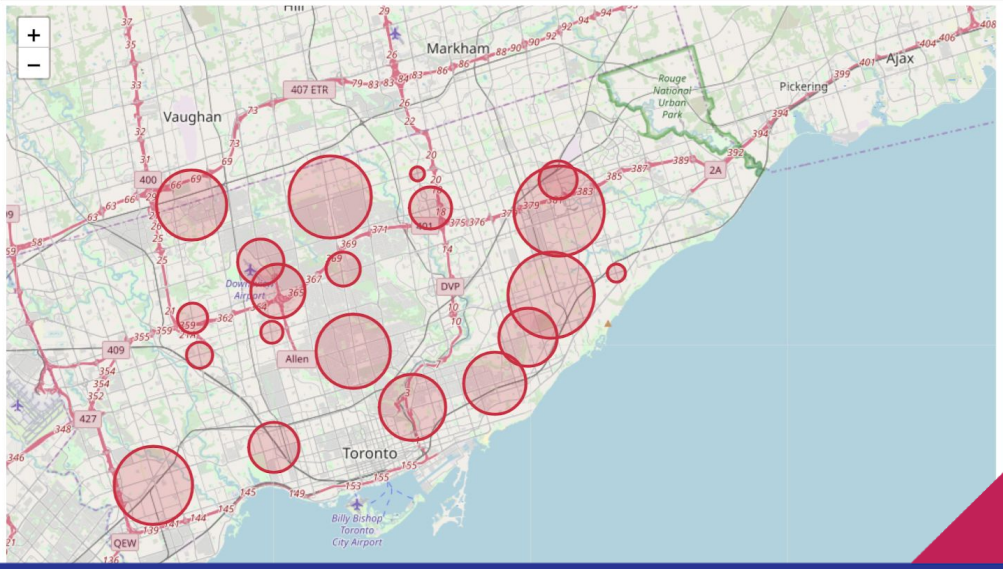
Map the Routes

Using `folium` let's take closer look where are these stops located

`Answer` : *It is all located in Toronto!*

```
In [28]: top20_map = folium.Map(location=[43.6532,-79.3832], zoom_start = 11)
         for i in range(0,len(location)):
             folium.CircleMarker(
                    location=[location.index[i][0], location.index[i][1]],
                    radius = int(45 - (2*i)),
                    color='crimson',
                    fill=True,
             ).add_to(top20_map)
         top20_map
```

```
Out[28]:
```



3. Ridership analysis

a. Who are the passengers? Which service people take the most?

Bar Plot the Results!

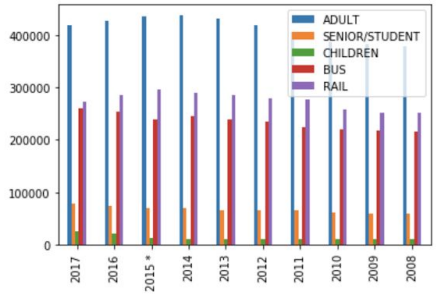
- Transpose data for plot clarity
- Set the new category for column name: "*ADULT*", "*SENIOR/STUDENT*", "*CHILDREN*", "*BUS*", "*RAIL*"
- Using matplotlib bar plot to plot the result

```
In [44]: # Transpose Dataset
passenger = passenger.T

# Set column name (for categories letter in the plot)
passenger.columns = ["ADULT", "SENIOR/STUDENT", "CHILDREN", "BUS", "RAIL"]

# Bar plot the data using matplotlib
passenger.plot.bar()
```

```
Out[44]: <matplotlib.axes._subplots.AxesSubplot at 0x1a0c0467f0>
```



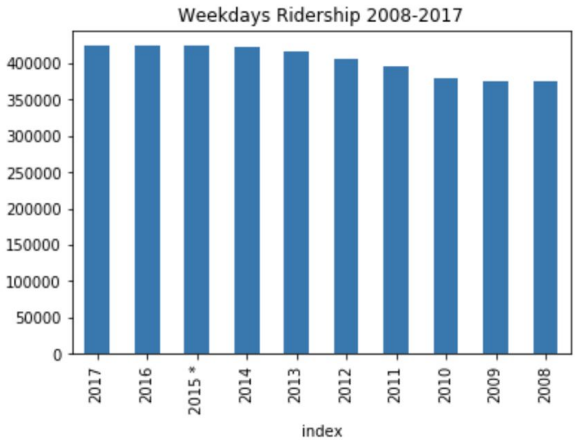
b. When do people take transit the most?

- On weekdays, over the years

Plot TTC Ridership on Weekdays 2008 -2017

```
In [51]: weekdays.plot.bar(legend=False, title="Weekdays Ridership 2008-2017")
```

```
Out[51]: <matplotlib.axes._subplots.AxesSubplot at 0x1a0ca18b00>
```

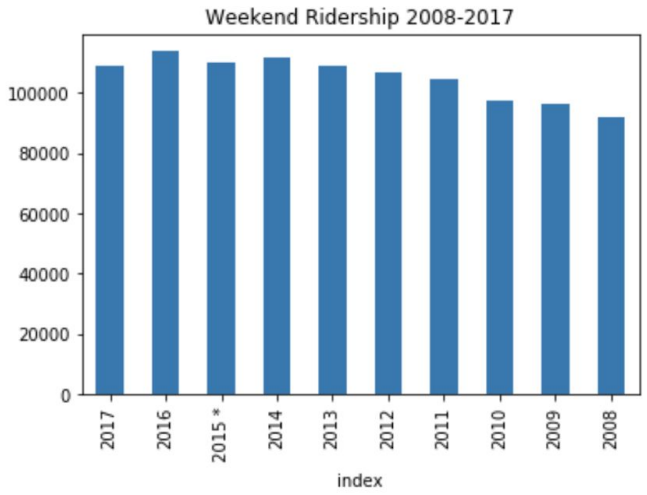


- On weekends, over the years

Plot TTC Ridership on Weekend 2008 -2017 ¶

```
In [52]: weekend.plot.bar(legend=False, title="Weekend Ridership 2008-2017")
```

```
Out[52]: <matplotlib.axes._subplots.AxesSubplot at 0x1a0f114ba8>
```



- Weekdays VS Weekend

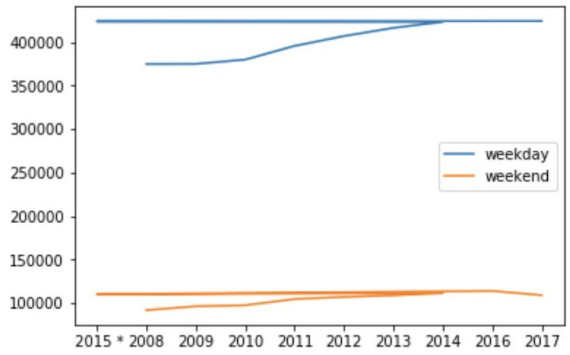
Overall weekdays vs weekend comparison

```
In [55]: # Plot weekdays
         plt.plot(weekdays, label = "weekday")

         # Plot weekend
         plt.plot(weekend, label = "weekend")

          # show plot legend
         plt.legend()
```

```
Out[55]: <matplotlib.legend.Legend at 0x1a0f2aab70>
```



- During Peak Time

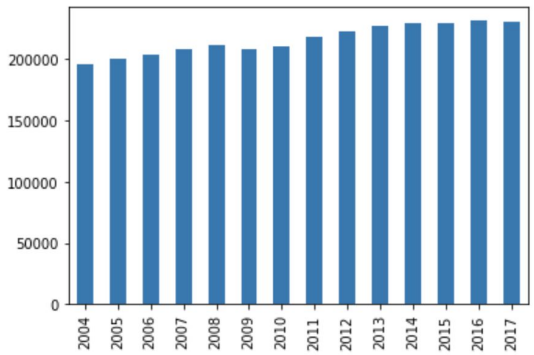
Peak Time

Is it increasing or decreasing?

In [101]:
```
# Transpose Data
peak_passengers = peak_passengers.T

# Plot data using pandas
peak_passengers.plot.bar(legend = False)
```

Out[101]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1f13d0f0>



- During Non-Peak Time

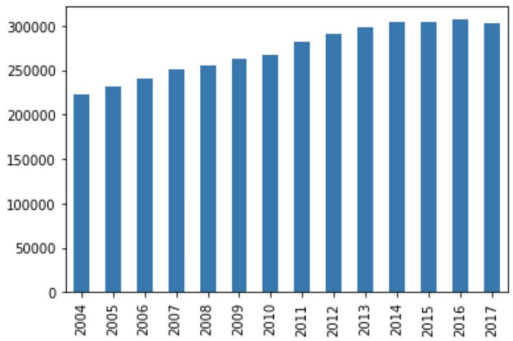
Non-Peak Time

Is it increasing or decreasing?

In [103]:
```
# Transpose data
non_peak_passengers = non_peak_passengers.T

# Plot data using pandas
non_peak_passengers.plot.bar(legend = False)
```

Out[103]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1f37b7b8>



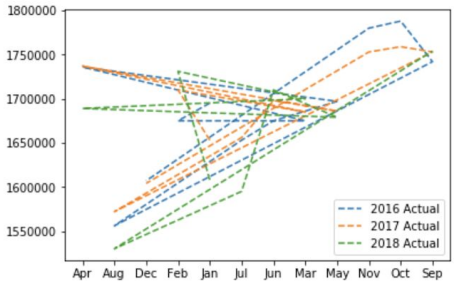
- Average Weekdays Ridership

Analysis: Average Weekday Ridership

How is it correlating over the months and years?

In [104]:
```
# Set 'Year' as index
average_weekday_ridership = average_weekday_ridership.set_index('Year')

# Plot how is it fluctuating depending on the months in the year
for i in range(len(average_weekday_ridership)):
    plt.plot(average_weekday_ridership.iloc[i], '--', label = average_weekday_ridership.index[i])
    plt.legend()
```



Analysis on the yearly average weekday ridership

Average Weekdays Ridership

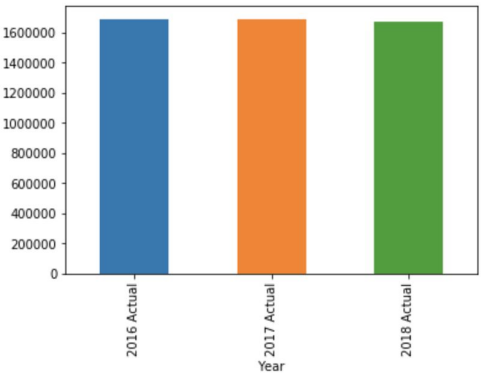
Let's do quick analysis over the yearly average

In [107]:
```
# Transpose Data
yearly_average = average_weekday_ridership.T

# Aggregate mean of the data
yearly_average = yearly_average.agg(np.mean)

# Plot the data
yearly_average.plot.bar(legend=False)
```

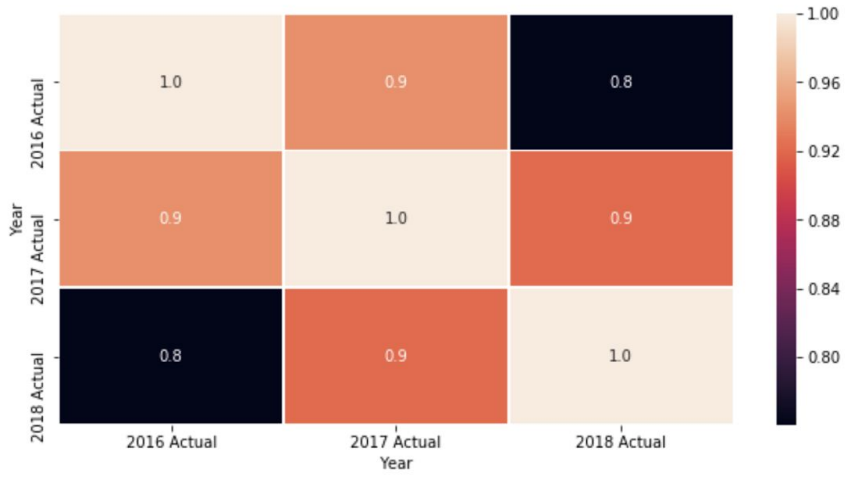
Out[107]: <matplotlib.axes._subplots.AxesSubplot at 0x1a297d9b00>



Correlation

How is the correlation of the mean over the years?

In [109]:
```
yearly_average = average_weekday_ridership.T
f,ax = plt.subplots(figsize=(10,5))
sns.heatmap(yearly_average.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)
plt.show()
```



4. The Most Taken Routes

Out[78]:

	index	Rank	Route #	Route Name	Ridership	nan
1	1	1	504	King	64579	NaN
2	2	2	32	Eglinton West	48684	NaN
3	3	3	36	Finch West	43952	NaN
4	4	4	52	Lawrence West	43882	NaN
5	5	5	510	Spadina	43804	NaN
6	6	6	501	Queen	43464	NaN
7	7	7	29	Dufferin	39721	NaN
8	8	8	506	Carlton	39601	NaN
9	21	21	25	Don Mills	39066	NaN
10	9	9	512	St. Clair	38113	NaN
11	10	10	54	Lawrence East	36277	NaN
12	11	11	35	Jane	32479	NaN
13	12	12	505	Dundas	32410	NaN
14	13	13	60	Steeles West	29819	NaN
15	14	14	34	Eglinton East	29501	NaN
16	15	15	53	Steeles East	28278	NaN
17	16	16	95	York Mills	27485	NaN
18	17	17	85	Sheppard East	27146	NaN
19	18	18	24	Victoria Park	26869	NaN
20	19	19	7	Bathurst	26251	NaN

Do analysis on the it's longitude and latitude then map the route out

In [91]:
```
# Plot route stops using Folium
m2 = folium.Map(location=[43.6532,-79.3832], zoom_start = 11)
for i in range(0,len(temp)):
    folium.CircleMarker(
          location=[route_rank.index[i][0], route_rank.index[i][1]],
          radius=10,
          color='blue',
          fill=True,
    ).add_to(m2)
```

In [117]:
```
# Preview the map
m2
```

Out[117]:

Compare the most TTC service offer VS The most taken route

Analysis: Are the frequent transits covering the popular routes taken by people?

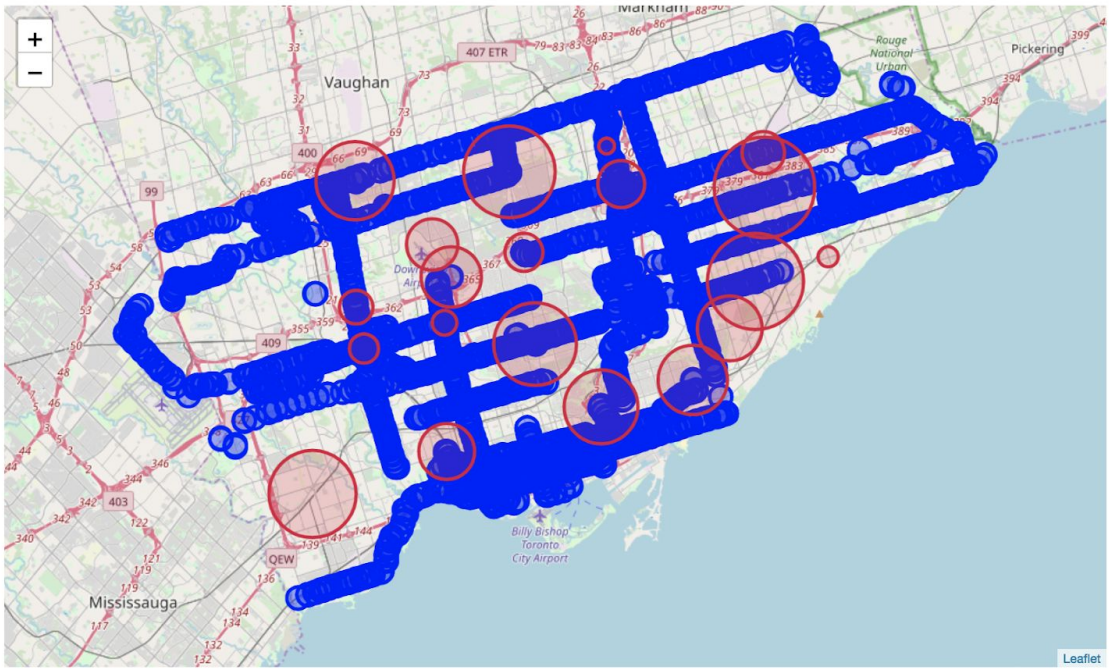
In [95]:
```
m3 = folium.Map(location=[43.6532,-79.3832], zoom_start = 10.5)

# Add all the popular route stop from 2016
for i in range(0,len(temp)):
    folium.CircleMarker(
          location=[route_rank.index[i][0], route_rank.index[i][1]],
          radius=10,
          color='blue',
          fill=True,
    ).add_to(m3)

# Add all top 20 TTC routes
for i in range(0,len(location)):
    folium.CircleMarker(
          location=[location.index[i][0], location.index[i][1]],
          radius = int(45 - (2*i)),
          color='crimson',
          fill=True,
    ).add_to(m3)


# Preview the map
m3
```

Out[95]:



5. TTC Delay

a. What cause the most delay?

Analysis TTC Subway delay 2014 to 2017

How long is the total delay and what cause the delay the most?

`Note: The code will be explained in the analysis document`

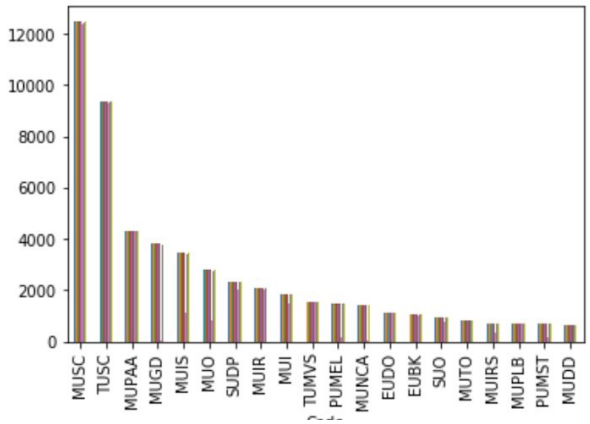
In [112]:
```
# Group by incident code
incident = subway_delay_14_to_17.groupby(['Code']).count()

# Sort by the the longest delay
incident = pd.DataFrame(incident).sort_values(by = "Min Delay", ascending = False)

# Take top 20 incidents
incident = incident.head(20)

# Plot top 20 incidents
incident.plot.bar(legend=False)
```

Out[112]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a29b999e8>
```



Top 5 TTC delay factors:

- MUSC: Miscellaneous Speed Control
- TUSC: Operator Overspeeding
- MUPAA: Passenger Assistance Alarm Activated - No Trouble Found
- MUIS: Injured or ill Customer (In Station) - Transported
- MUO: Miscellaneous Other

b. TTC Service that experience the most delay in 2018

Overview: TTC Delay in 2018

In [113]:
```
# Merge summary of subway and streetcar delay
delay = subway_delay.merge(streetcar_delay, on = "Time")
```

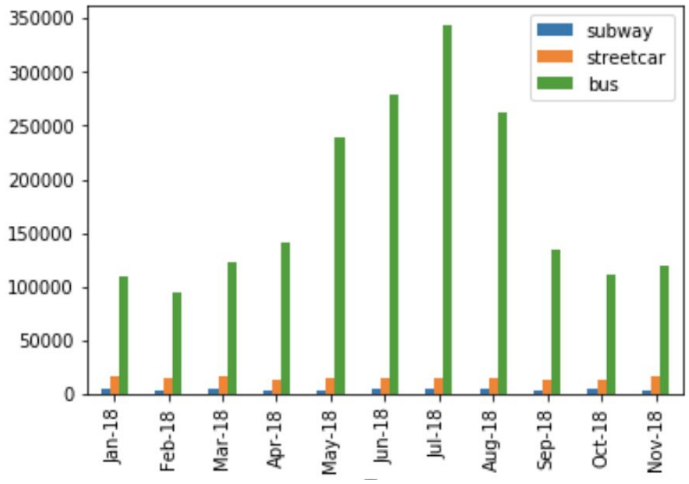
In [114]:
```
# Merge summary of subway + streetcar delay and bus delay
delay = delay.merge(bus_delay, on = "Time")
```

In [115]:
```
# Set Time as index
delay = delay.set_index("Time")

# Set column names
delay.columns = ["subway", "streetcar", "bus"]
```

In [116]:
```
# Plot to compare which TTC service experience the most in 2018
delay.plot.bar()
```

Out[116]: <matplotlib.axes._subplots.AxesSubplot at 0x1a36c04f98>



Conclusion

1. TTC Fare keep increasing over the years by few cents every year other than big jump between 1980 to 1996
2. Top 5 out of 20 TTC Routes (most offered by TTC):
    - Queen
    - Lawrence West
    - King
    - Eglinton West
    - Lawrence East
3. Ridership:
    - People who are identified as 'Adult' take TTC the most
    - People take subway and streetcars more than bus
    - The riderships on weekdays and weekends has gone down over the years
    - There are significant difference between TTC service usage on the weekdays and on the weekend ---people use it more on the weekdays
    - Average weekdays ridership between 2016 and 2018 is pretty stable but when we take closer looks there's a strong positive correlation between 2016 and 2017 and strong positive correlation between 2017 and 2018. Although there is a strong negative correlation between 2016 and 2018.

3. TTC Most Taken Routes:

    - There are some spots where TTC service is in the top 20 most offered but not in the most taken route lines

4. TTC Delay:

    - Highest factor of TTC delay is Miscellaneous Speed Control followed by Operator Overspeeding, Passenger Assistance Alarm Activated - No Trouble Found, Injured or ill Customer (In Station) - Transported, and others.
    - Service that experience the most delay in 2018 is TTC bus (might be also caused by its quantity that is more than subway and streetcars)