ENSF-592

Final Project – Calgary Incident Analysis

Mike Lasby

8/11/2020

Lasby Data Analytics

August 11, 2020

Traffic Management Centre Coordinator
City of Calgary
Box 2100, Station M
Calgary AB, T2P 2M5

Re: Calgary Incident Analysis
Data Analysis Report

As per our proposal dated August 6th, 2020, we have completed the analysis of the requested traffic and weather data for the City of Calgary in 2018. The following report outlines our findings, observations, and recommendations regarding the data provided.

BACKGROUND & SCOPE OF WORK

- The City of Calgary is home to approximately 1.4 million people, many of whom drive to work daily. The City of Calgary requested that we perform an analysis on several data sets to determine how specific road features and weather conditions affect the number of automobile incidents ('incidents').
- Road feature data was provided by the City of Calgary. The following road features were included in our analysis:
    - Speed Limit;
    - Traffic Volume;
    - Intersection Safety Cameras ('Traffic Cameras');
    - Traffic Signals;
    - Traffic Signs.
- Climate data was obtained from Environment Canada for daily and hourly conditions at the Calgary International Airport. The following climate conditions were included in our analysis:
    - Hourly Temperature;
    - Average Daily Temperature;
    - Hourly Visibility;
    - Average Daily Visibility;
    - Melt/Freeze Cycles.
- In general terms, our scope of work included the following:
    - Obtaining the above-noted data sets;
    - Dividing the city into 100 equally sized cells to facilitate analysis of road features;
    - Cleaning, manipulating, and analyzing the data to determine the effect, if any, of the road features and climate conditions on the number of incidents;
    - Plotting the data to a variety of graphical formats to visualize the data and;
    - Reporting on our findings

METHODOLOGY

During the course of our analysis, several assumptions were required to facilitate manipulation of the data and to glean meaningful results. These assumptions are detailed in this section along with a description of our analysis methodology in general.

Application Architecture:

Our analysis software was developed to follow the Model-View-Controller design paradigm. In this case, we selected the Jupyter Notebook interactive python shell to be the user-interface 'View' of the application due to its suitability for data analysis applications and ability to display data visualizations in-line with source code.

The View notebook module depends on the Controller module to issue commands to the data Model. The Controller provides the View with an interface which parses input from the user, manipulates the Model, and prepares data into a format suitable for plotting. The majority of the program logic resides in the Controller class, allowing us to completely encapsulate the low-level data manipulation from the user interface, thus ensuring that *even* Calgary traffic engineers will be able to use the program.

The Model contains a collection of data frames and functions for manipulating, populating, and updating those data frames. The model is independent of the user interface and may only be updated by the controller class through the Model's application protocol interface (API).

Speed Limits:

The City of Calgary dataset did not include speed limits for every road or spatial coordinate data for every road segment. Therefore, we normalized our average speed based on the number of points contained within each cell. To illustrate with an example, if we had three (3) data points within a single cell with speed data of 50, 50, and 100 kilometers per hour (km/hr), the average cell speed limit would be calculated as 50*2+100*1/(2+1) for an average speed of 66.66 km/hr.

We note that the lack of speed limit data for many roads reduces our confidence our results with respect to the speed limits. We recommend obtaining a more complete dataset for further analysis. In order to reduce the effect of cells with missing data in our analysis, we excluded cells with no speed limit data from further analysis (ie., filled those cells with np.nan.)

Traffic Volume:

Traffic volumes were reported as constant values for continuous road segments. We understand that these volume counts are typically obtained by measuring the number of vehicles passing a point on the road within a specific timeframe; however, the recording point coordinates and the timeframe was not provided. Therefore, similar to the above, we normalized the traffic volume by multiplying the volume from each road by the number of points from that road segment before dividing but the sum of points from all road segments within the cell.

We note that the lack of volume data for many roads reduces our confidence our results with respect to the speed limits. We recommend obtaining a more complete dataset for further analysis. In order to reduce the effect of cells with missing data in our analysis, we excluded cells with no volume data from further analysis (ie., filled those cells with np.nan.)

Cameras, signals, and signs:

Road features with point coordinates were simply counted within each cell and reported as the sum of the count. Where no values were reported, a count of 0 was reported to the cell.

Volume Normalization:

In order to ensure that the correlations calculated below were not simply dependent on the number of vehicle trips within each cell, we also plotted several graphs depicting the effect of the above-noted road features vs. incidents per million trips. The volume normalized results will be compared with the standard datasets to determine if any correlations found are simply due to a higher number of trips resulting in a larger number of incidents as opposed to a statistically significant correlation.

Climate Conditions:

Climate conditions were compared at the hourly and daily level. We note that due to a lack of time dependent traffic volume data, we are unable to normalize these results with respect to volume. For example, cold conditions often occur overnight when few vehicles are on the road. As a result, correlations between overnight temperatures may be lower than anticipated if the same conditions would have occurred during the morning or evening rush hours.

Melt Freeze:

Calgary is subject to a high frequency of melt/freeze cycles every winter due to our Chinook winds and cold winters. In order to test the hypothesis that incidents are more frequent immediately following a freeze event, we identified each melt/freeze cycle within the data provided and compared the number of incidents within eight (8) hours of the freeze event with the anticipated average number of incidents per hour from a typical hour.

RESULTS & RECOMMENDATIONS:

Refer to Appendix A for the complete selection of visualization plots. Our analysis is divided into three sections, geospatial data, road features, and climate conditions.

Geospatial Data:

Generally, we noted that incident counts tended to increase towards the center of the City and decrease at the peripheries.

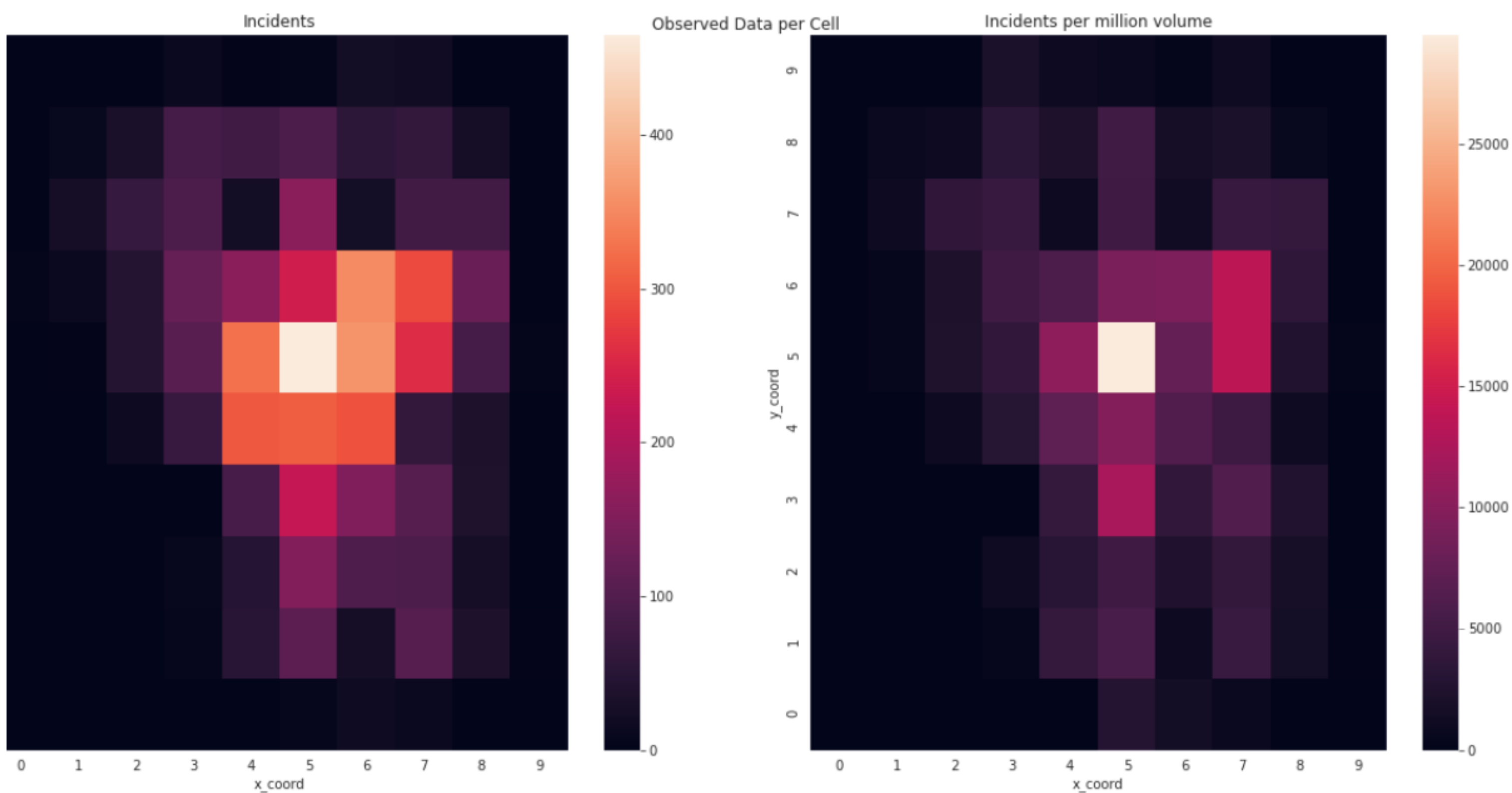


Figure 1: Incidents vs. Cell

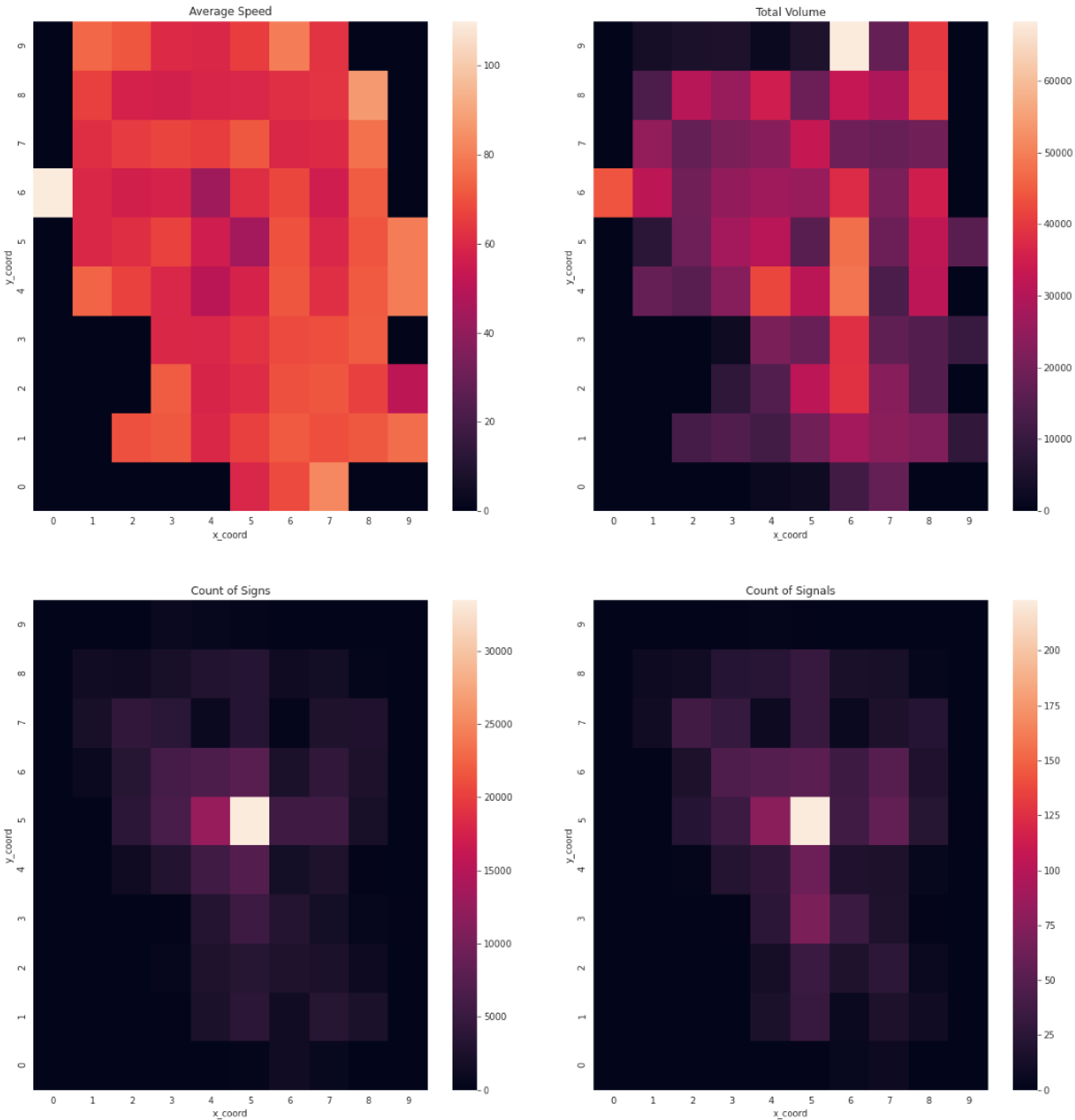


Figure 2: Heat maps of various road features by cell

The highest number of incidents was recorded in the downtown core, which appears as the bright white cell in the centre of the above plots. The outer edges of the city had little to no data provided; therefore, the boundary of the city appears dark on the above figures. We note that the distribution of average speed and volume was variable throughout the City, but sign, signal, and camera counts were most numerous downtown and adjacent cells.

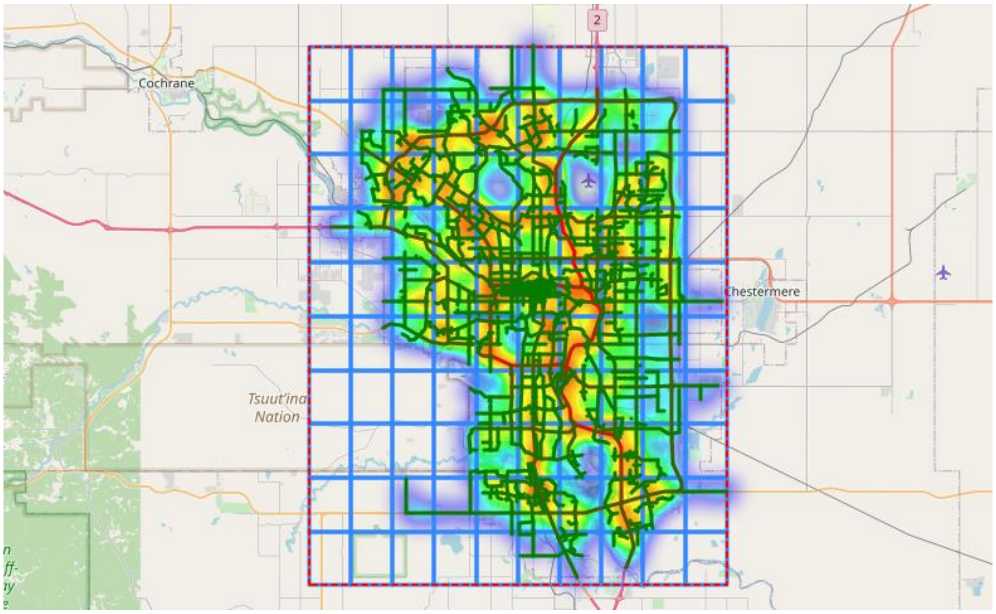


Figure 3: Traffic Volume Heat Map

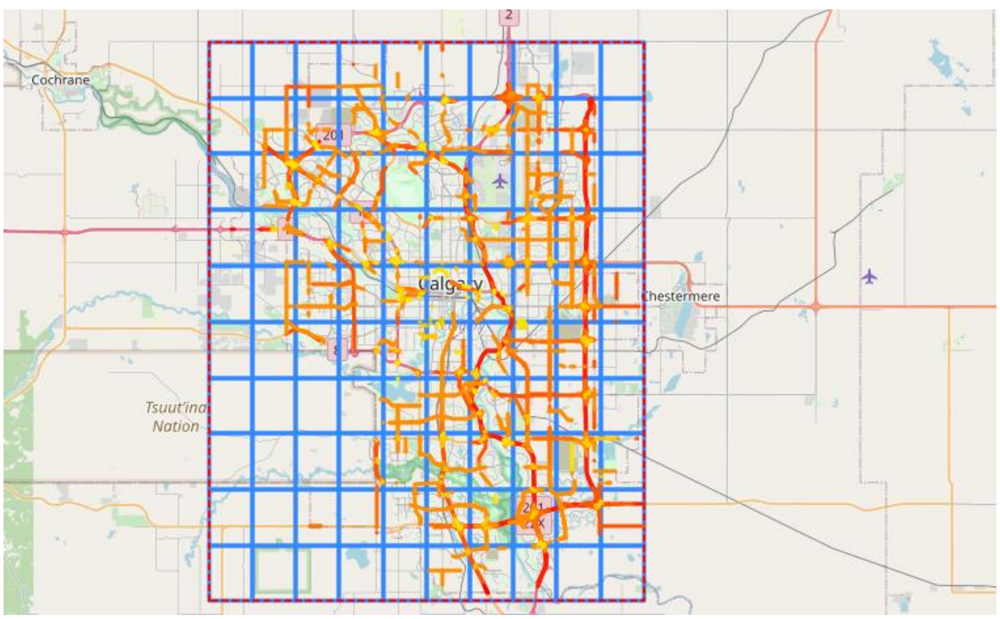


Figure 4: Speed Limit Weighted PolyLine Map

The above map figures depict the average traffic and speed limits throughout the city. We recommend viewing these maps in Jupyter Notebooks as tooltip information is provided for an interactive experience.

Based on the above, we noted that the speed and volume data varies throughout the City without any significant trends noted. However, the number of incidents, cameras, signals, and signs were more numerous near the centre of the City.

Road Features:

Our static analysis includes the road features, which are unchanging with respect to time. The below figure describes the correlation of the various road features:

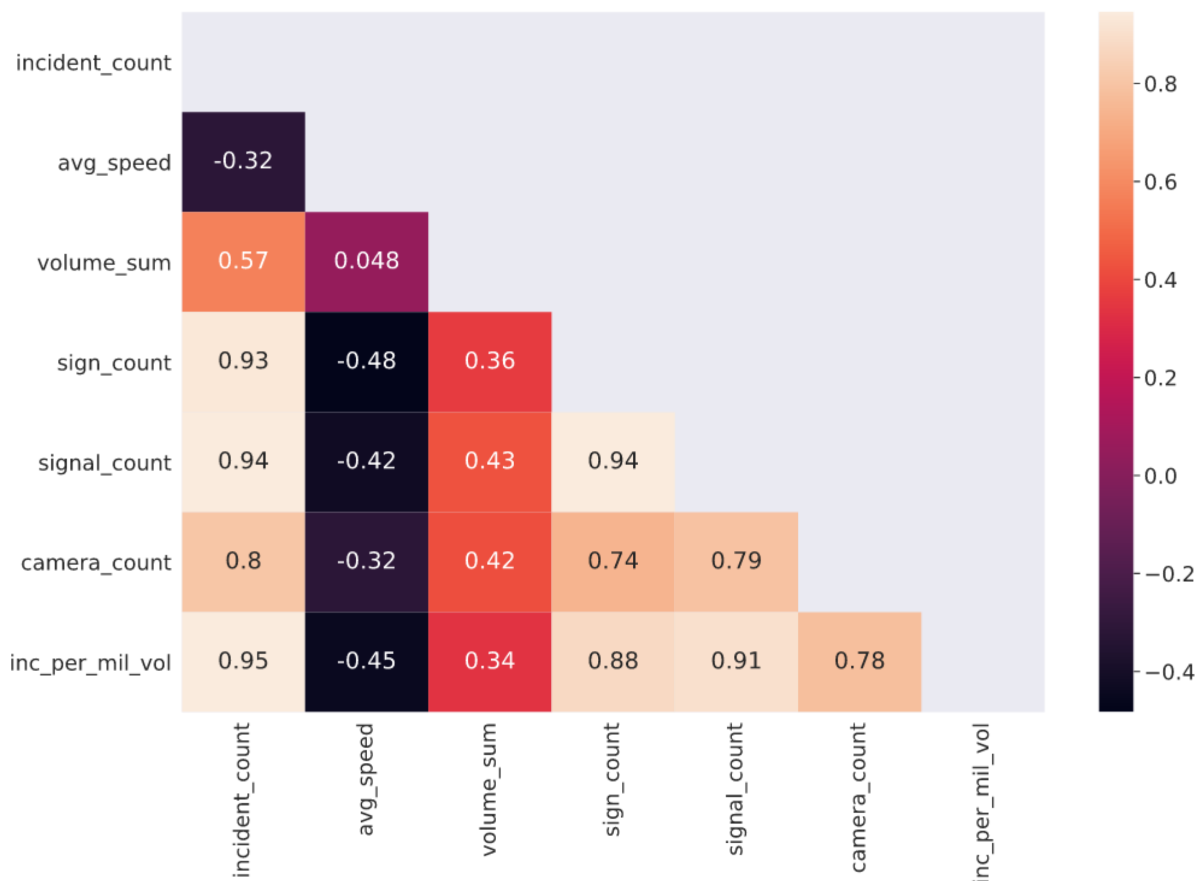


Figure 5: Incidents vs. Static Road Features

As we can see above, camera counts, signal counts, and sign counts strongly correlate with number of incidents. These correlations appear to remain significant even after normalising the data with respect to number of incidents per million vehicle trips. We noted that the highest number of these features was found in the downtown core, which also had the highest number of incidents. Signage and signals are typically required near or at intersections; we recommend further analysis to study the effect of intersection count on the total number of incidents.

We found an inverse correlation with average speed and number of incidents. There are a number of potential explanations for the observed relationship. For example, higher speed limits roads typically have longer sight lines, wider margins, and fewer intersections. We note that our data did not include the severity of the reported incidents. We recommend further analysis to compare severity of the incidents with the average speed limit of the road. Despite incidents being less frequent on high speed road, it may be that the incidents that do occur result in more fatalities than the lower speed but high incident frequency roads.

Figure 6: Multi-Plot for Incidents Vs. Total Volume

Each of the road features investigated was plotted in a histogram plot, box plot, point plot, and line plot as seen above. For average speed, camera counts, sign counts, signal counts, we provided a second multi-plot comparing those features to the incidents normalised by volume.

From the histograms, we noted that the sign, camera, and signal counts were not normally distributed. The vast majority for cells have low numbers of these elements compared to the outliers such as the downtown cell (cell no. 55).

The point plots and box plots depict a moderate correlation between incidents and traffic volume and average speed. However, these features do not appear to be as significant as the number of cameras, signs, or signals. Finally, the line plots for each graph generally show a large variation in the underlying data sets. Due to the large amount of variables that could affect incident frequency, we recommend expanding the study to include multiple years and more data, where possible, to further refine our analysis of the road features.

Climate Conditions:

The daily and hourly climate conditions compared were correlated with the number of incidents. The correlation heat maps below depict the correlation between the number of incidents in a given hour or day with the hourly temperature and visibility or daily average temperature and visibility, respectively. We did not find strong correlations with either dataset, but generally the correlations were stronger when comparing daily conditions as opposed to hourly. We suspect that the hourly data shows a lower correlation due to the variability in traffic volume throughout the day. Cold temperatures and fog often occur in the night when few vehicles are on the road. We recommend conducting further analysis on specific time periods, such as the morning or evening rush hour, to ensure that the number of vehicles on the road remains more constant when comparing climate conditions.

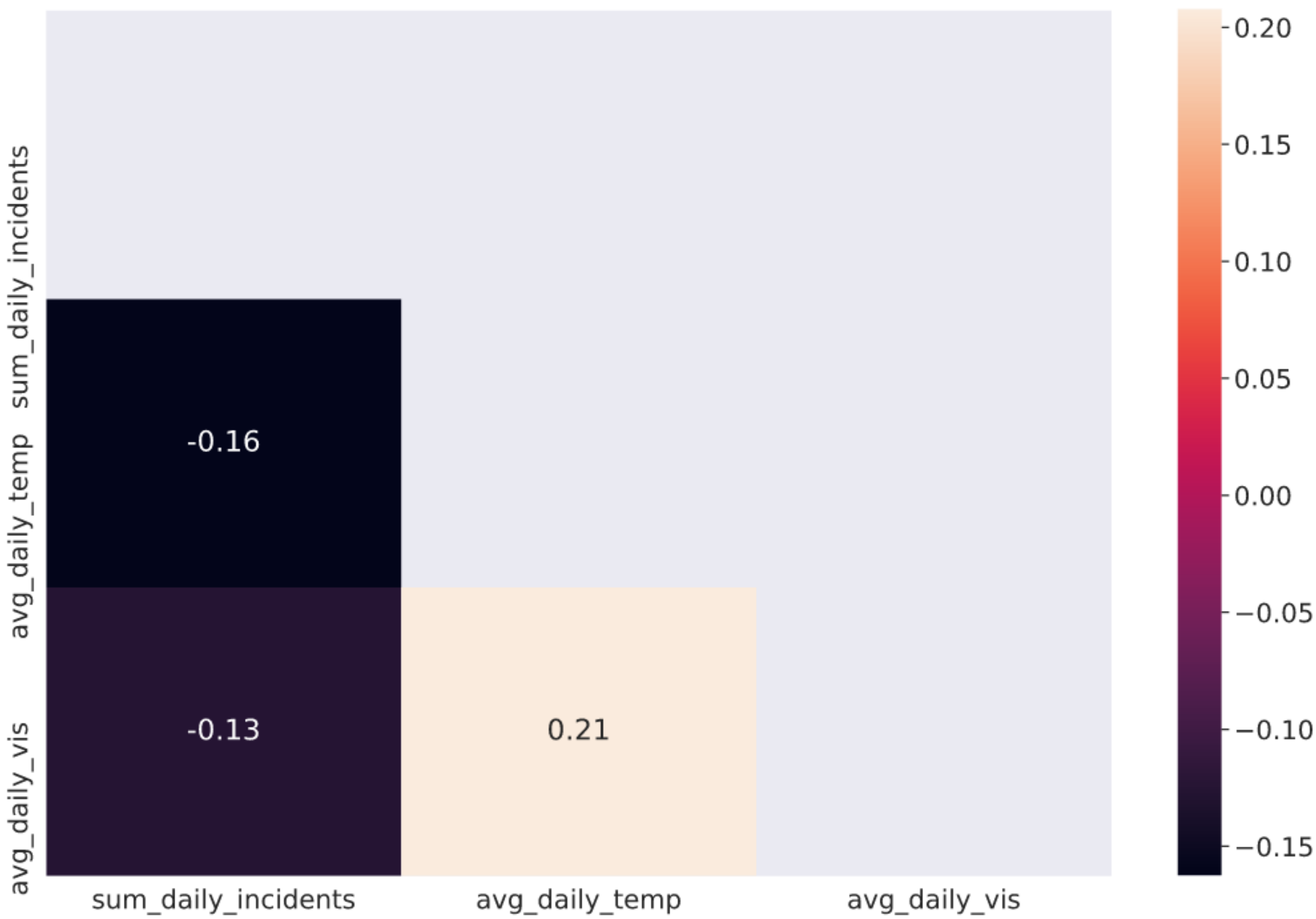


Figure 7: Incidents vs. Daily Average Climate Conditions

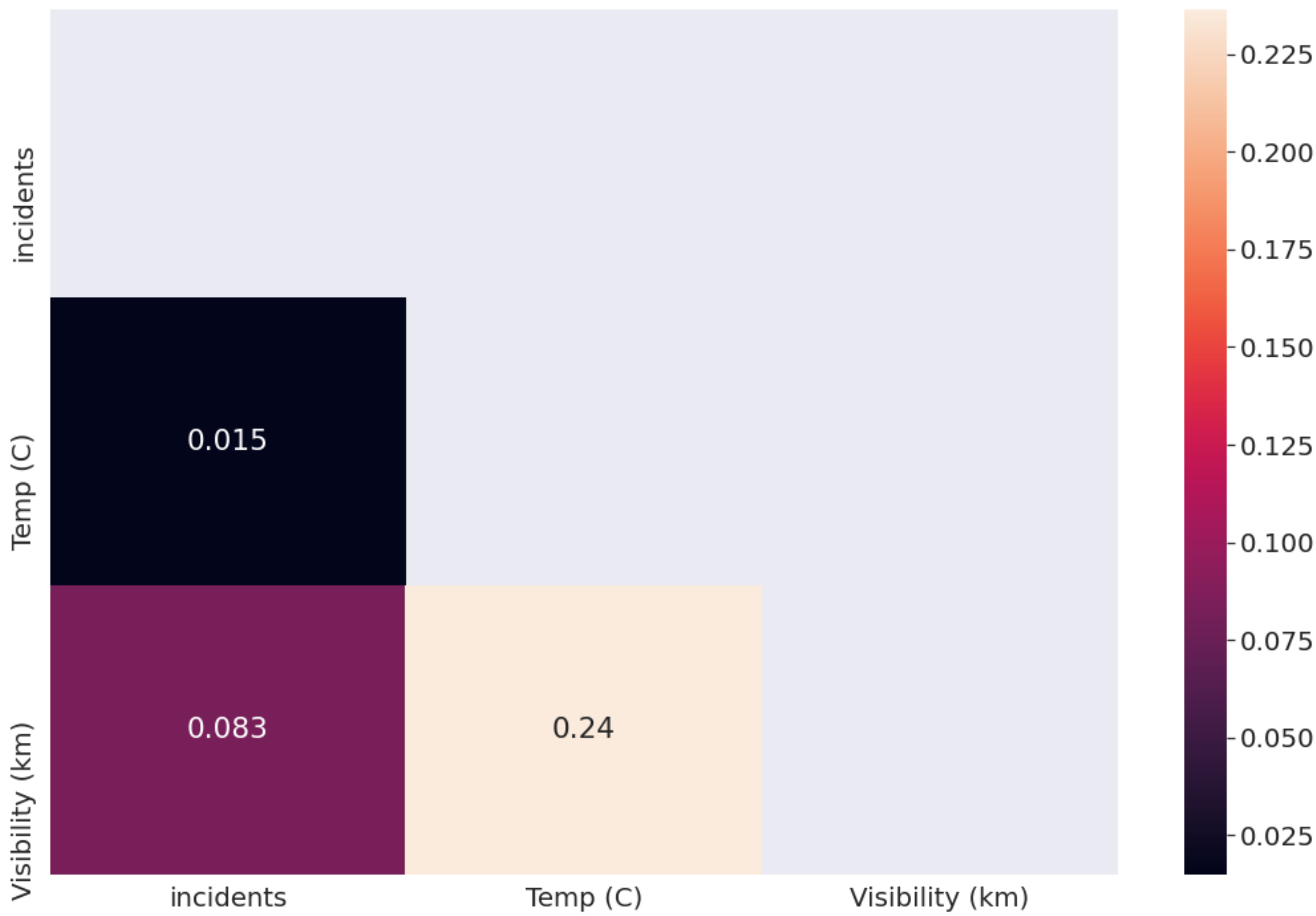


Figure 8: Incidents vs. Hourly Climate Conditions

From the above daily average climate conditions, we can see that both the temperature and visibility are weakly negatively correlated with the number of incidents. Therefore, as the temperature and visibility decrease, the anticipated number of incidents increases. Similar to the discussion above, we suspect that a more targeted analysis of specific time periods may yield a more significant correlation between these variables and the number of incidents. The below multi-plots summarize these climate condition variables.

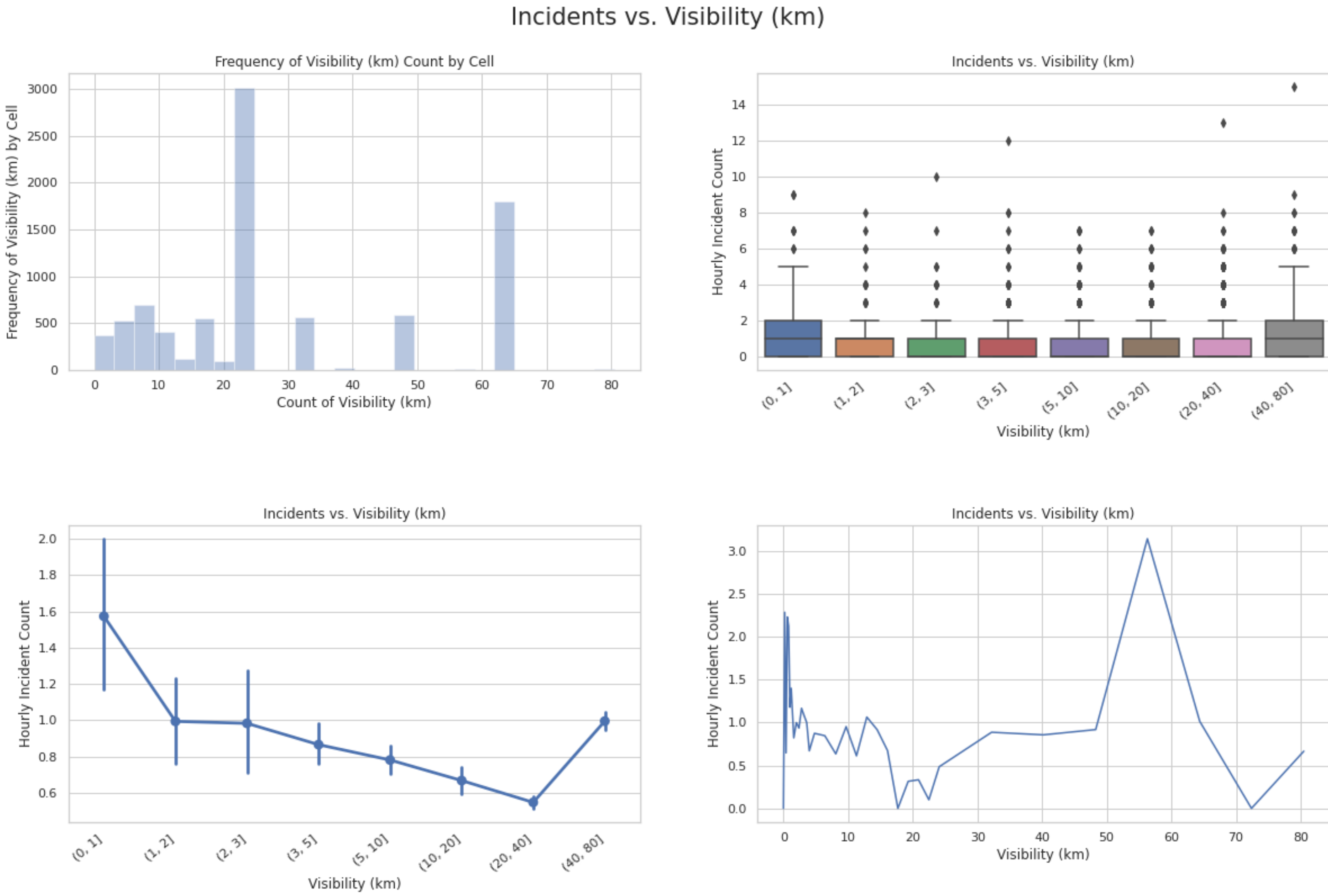


Figure 9: Incidents vs. Visibility (km)

As we expected, the number of incidents increase as visibility decreases. However, we did note a small increase in the number of incidents when very high visibility was observed. We suspect that this may be due to people driving more aggressively during nice weather. As we can see from the box plot, there are a significant number of outliers in this dataset. We recommend expanding the analysis data set to determine if these outliers are statistically significant.

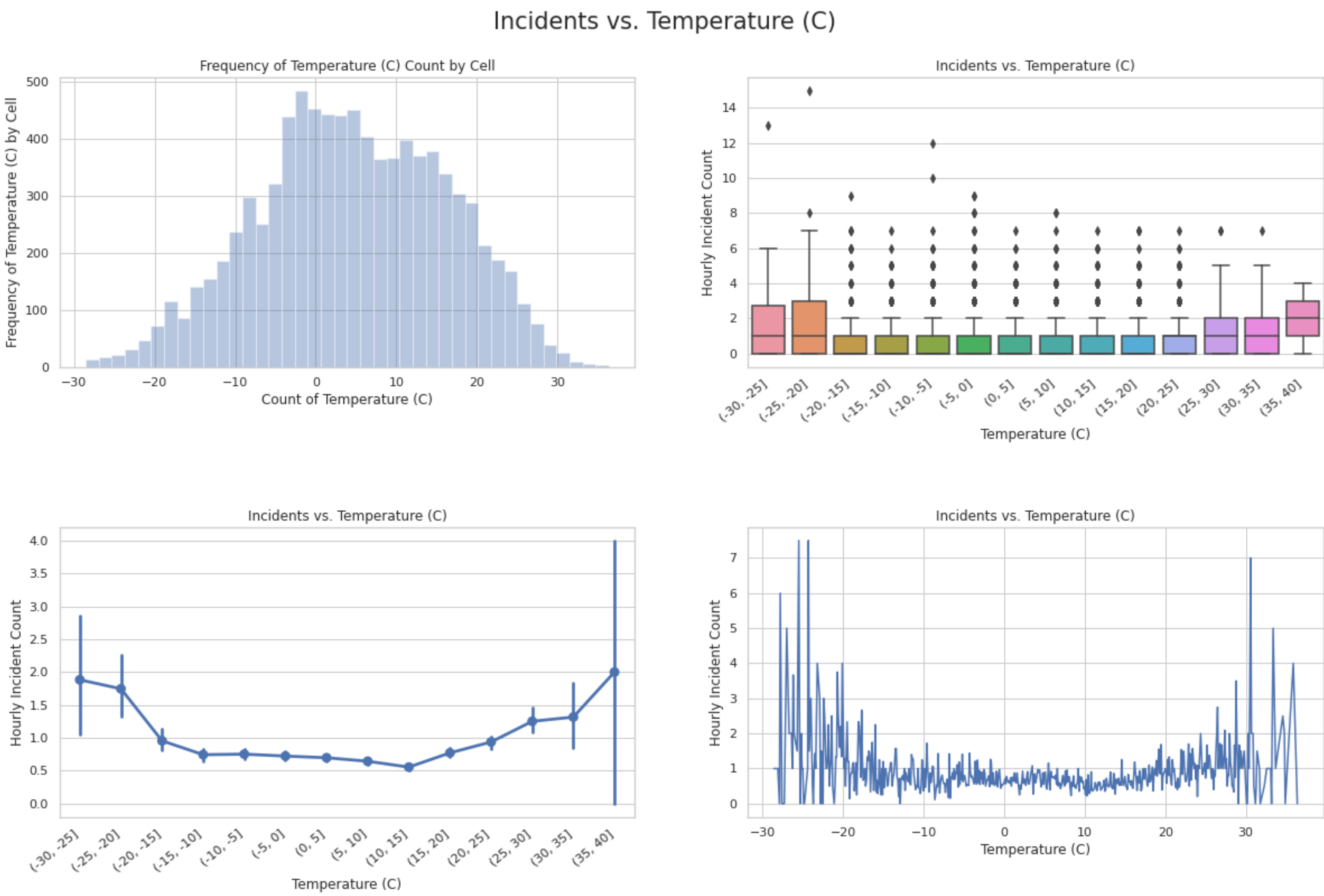


Figure 10: Incidents vs. Temperature (^{o}C)

The temperature results above produce some unexpected results. While we had hypothesised that the number of incidents would increase as temperature decreases, we also noted that the number of incidents increased as the temperature increased. As we have very few data points within a single year at the extreme hot and cold ends of the data set, we recommend expanding the study to include additional years to determine if the relationship observed is statically significant. Similar to the visibility data, we noted that there are many outliers in our data set.

The increase in incidents at high temperatures may be due to increased volume during busy summer months. Correlating the temperature data with traffic counts was not possible with the data provided. We recommend that the City consider recording the number of vehicles per hour at a few select locations throughout the City so that we may correlate volume with temperature and determine if the observed trends exist independent of volume numbers.

We suspect that with a volume normalised dataset, the correlation between temperature and incidents would increase. Many drivers will attempt to avoid driving in bad weather by taking public transit or skipping the trip during cold, snowy weather. Therefore, the number of incidents recorded at cold temperatures may represent a higher number of incidents per traffic volume.

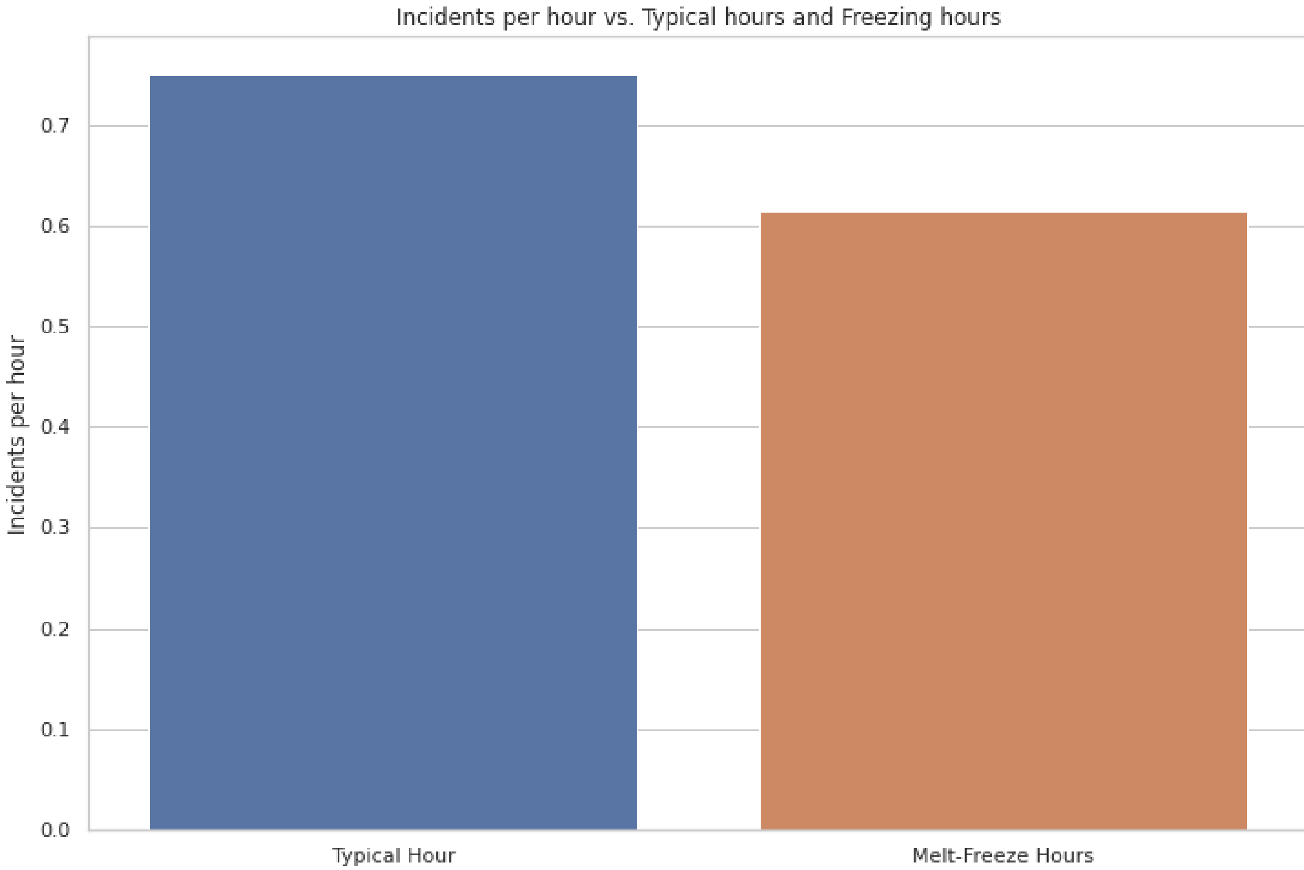


Figure 11: Melt Freeze Analysis

We were surprised that the average number of incidents per hour did not significantly change during the hours following a freeze event. We had hypothesized that more incidents would occur after freeze events due to rapidly changing road conditions. Similar to the temperature data above, normalising by traffic volume may reveal a more significant variation in the average incidents per hour during melt-freeze hours. As overnight freezes are common, it may be that there are simply fewer cars on the road immediately following a freeze event. We selected a target temperature of 0^{o}C and a period of eight (8) hours for this analysis. i.e., if the temperature falls from 1 to 0 at midnight, we compare the hours between midnight and 8 AM with the typical hour throughout the year.

CONCLUSION:

Should any new information come to light, we request the opportunity to review the conclusions and recommendations included in this report.

This report has been prepared for the exclusive use of the City of Calgary and the University of Calgary. Any use a third party makes of this report or decisions made based on the information provided is the responsibility of the third party.

We trust this is the information you require at this time. Should you have any questions, comments, or concerns, please do not hesitate to contact our office.

Lasby Data Analytics

Sincerely,

M. Lasby

Mike Lasby
Lead Data Scientist
mklasby@gmail.com
T: 587-777-9257

APPENDIX A – JUPYTER NOTEBOOK OUTPUT

NOTE: WE STRONGLY RECOMMEND VIEWING IN JUPYTER NOTEBOOK. FOLIUM MAPS CANNOT BE PRINTED TO PDF

ENSF 592 - FINAL PROJECT

Calgary Incident Analysis

By: Mike Lasby

README

This notebook is the entry point of our project and functions as the View. The notebook depends on the Controller.py class to retrieve views of our data from the Model. Please see the enclosed pdf report for a discussion of the data analysis below.

In [1]:
```
%matplotlib inline
import numpy as np
import pandas as pd
import seaborn as sns
from geojson import Point, MultiLineString
import geopandas as gpdnum_points
import re
import math
from folium_0_12 import folium #using pre released v0.12 due to heatm
ap issues
# NOTE:  https://github.com/python-visualization/folium/issues/1271
import matplotlib.pyplot as plt
from controller import Controller
```

In [2]:
```
ctrl = Controller()
ctrl.load_data()
ctrl.add_geo_cols()
ctrl.add_cell_col()
ctrl.get_cell_data()
ctrl.generate_maps()
```

```
Loading Data...
Getting weather at yyc for month 1 in 2018
Getting weather at yyc for month 2 in 2018
Getting weather at yyc for month 3 in 2018
Getting weather at yyc for month 4 in 2018
Getting weather at yyc for month 5 in 2018
Getting weather at yyc for month 6 in 2018
Getting weather at yyc for month 7 in 2018
Getting weather at yyc for month 8 in 2018
Getting weather at yyc for month 9 in 2018
Getting weather at yyc for month 10 in 2018
Getting weather at yyc for month 11 in 2018
Getting weather at yyc for month 12 in 2018
...Data Loaded.
Adding geometry column to speeds from multiline. Flip coords? True
Adding geometry column to volumes from multilinestring. Flip coords?
True
Adding geometry column to incidents from location. Flip coords? False
Adding geometry column to cameras from None. Flip coords? True
Adding geometry column to signals from Point. Flip coords? True
Adding geometry column to signs from POINT. Flip coords? True
Adding geometry column to cells from cell_bounds. Flip coords? False
Adding cell column to speeds
Adding cell column to volumes
Adding cell column to incidents
Adding cell column to cameras
Adding cell column to signals
Adding cell column to signs
Generating cell data...
...cell data generated.
Analyzing cell data...
...cells analyzed.
Generating maps...
...maps generated.
```

Sample Output

In [3]:
```
cells = ctrl.get_frame('cells')
display(cells.sort_values(by='avg_speed', ascending = False).head())
display(cells.sort_values(by='volume_sum', ascending = False).head())
display(cells.sort_values(by='incident_count', ascending = False).hea
d())
```

	cell_bounds	avg_speed	volume_sum	incident_count	sign_count	signal_count
60	[[51.0645838, -114.315796], [51.1015441, -114....	110.00	44000.00	5	116	0
88	[[51.1385044, -113.9510832], [51.1754647000000...	86.88	40318.58	26	346	3
7	[[50.842822, -113.9966723], [50.8797823, -113....	82.53	17805.13	13	756	7
96	[[51.175464700000006, -114.0422614], [51.21242...	80.09	68347.83	22	52	0
59	[[51.027623500000004, -113.9054941], [51.06458...	80.00	16000.00	5	118	1

	cell_bounds	avg_speed	volume_sum	incident_count	sign_count	signal_count
96	[[51.175464700000006, -114.0422614], [51.21242...	80.09	68347.83	22	52	0
46	[[50.9906632, -114.0422614], [51.0276235000000...	71.29	48558.51	297	1813	19
56	[[51.027623500000004, -114.0422614], [51.06458...	70.07	47624.64	362	5590	38
60	[[51.0645838, -114.315796], [51.1015441, -114....	110.00	44000.00	5	116	0
44	[[50.9906632, -114.1334396], [51.0276235000000...	50.43	42047.33	303	5722	28

	cell_bounds	avg_speed	volume_sum	incident_count	sign_count	signal_count
55	[[51.027623500000004, -114.0878505], [51.06458...	44.58	15776.74	465	33465	223
56	[[51.027623500000004, -114.0422614], [51.06458...	70.07	47624.64	362	5590	38
66	[[51.0645838, -114.0422614], [51.1015441, -113...	67.48	37861.24	354	2475	36
54	[[51.027623500000004, -114.1334396], [51.06458...	55.91	31102.52	326	12676	77
45	[[50.9906632, -114.0878505], [51.0276235000000...	58.09	31340.81	307	7204	61

Map Visualizations

Cell Map

In [4]:

```
%%html
<iframe src="cell_map.html" frameborder = "0" width = "960" height = "600" allowfullscreen="true" mozallowfullscreen="true" webkitallowfullscreen="true"></iframe>
```

Average Speed Map

In [5]:

```
ctrl.draw_speed_map()
```

```
map saved
```

In [6]:
```
%%html
<iframe src="speed_map.html" frameborder = "0" width = "960" height =
"600" allowfullscreen="true" mozallowfullscreen="true" webkitallowful
lscreen="true"></iframe>
```

Traffic Volume Heatmap

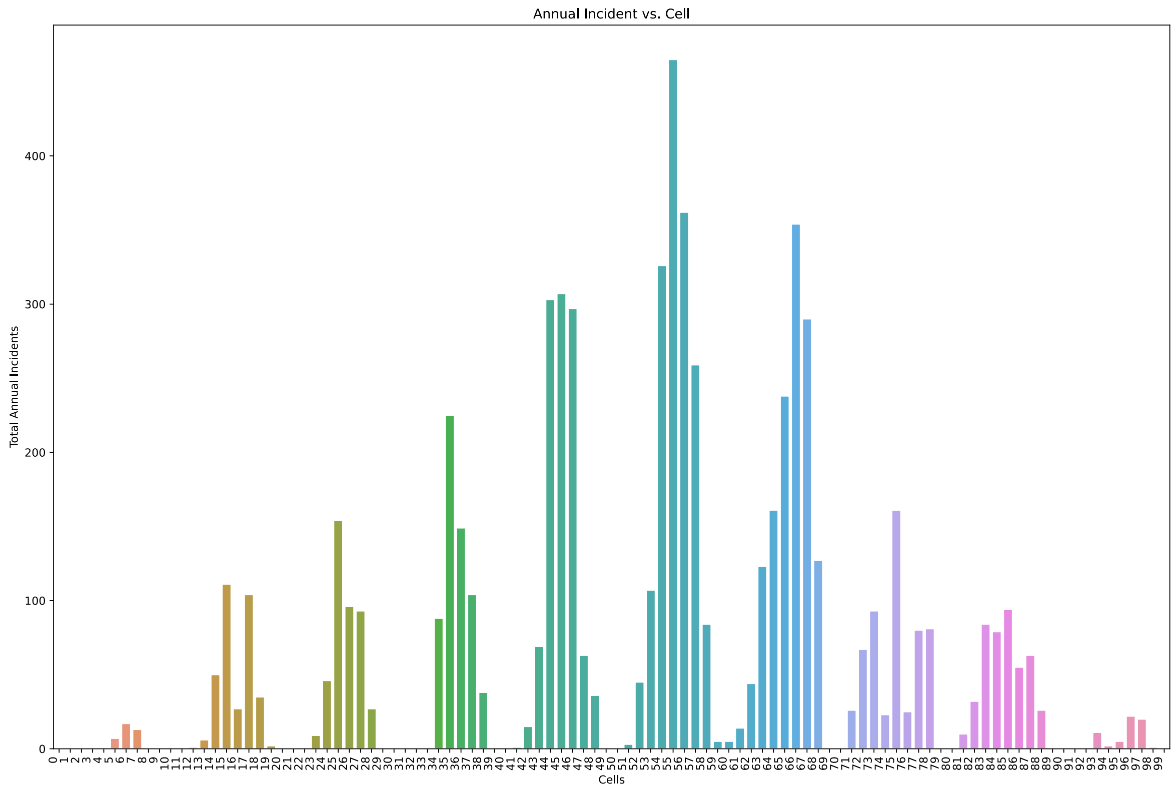
In [7]:
```
ctrl.gen_heatmap()
```

```
map saved
```

In [8]:
```
%%html
<iframe src="volume_map.html" frameborder = "0" width = "960" height
= "600" allowfullscreen="true" mozallowfullscreen="true" webkitallowf
ullscreen="true"></iframe>
```

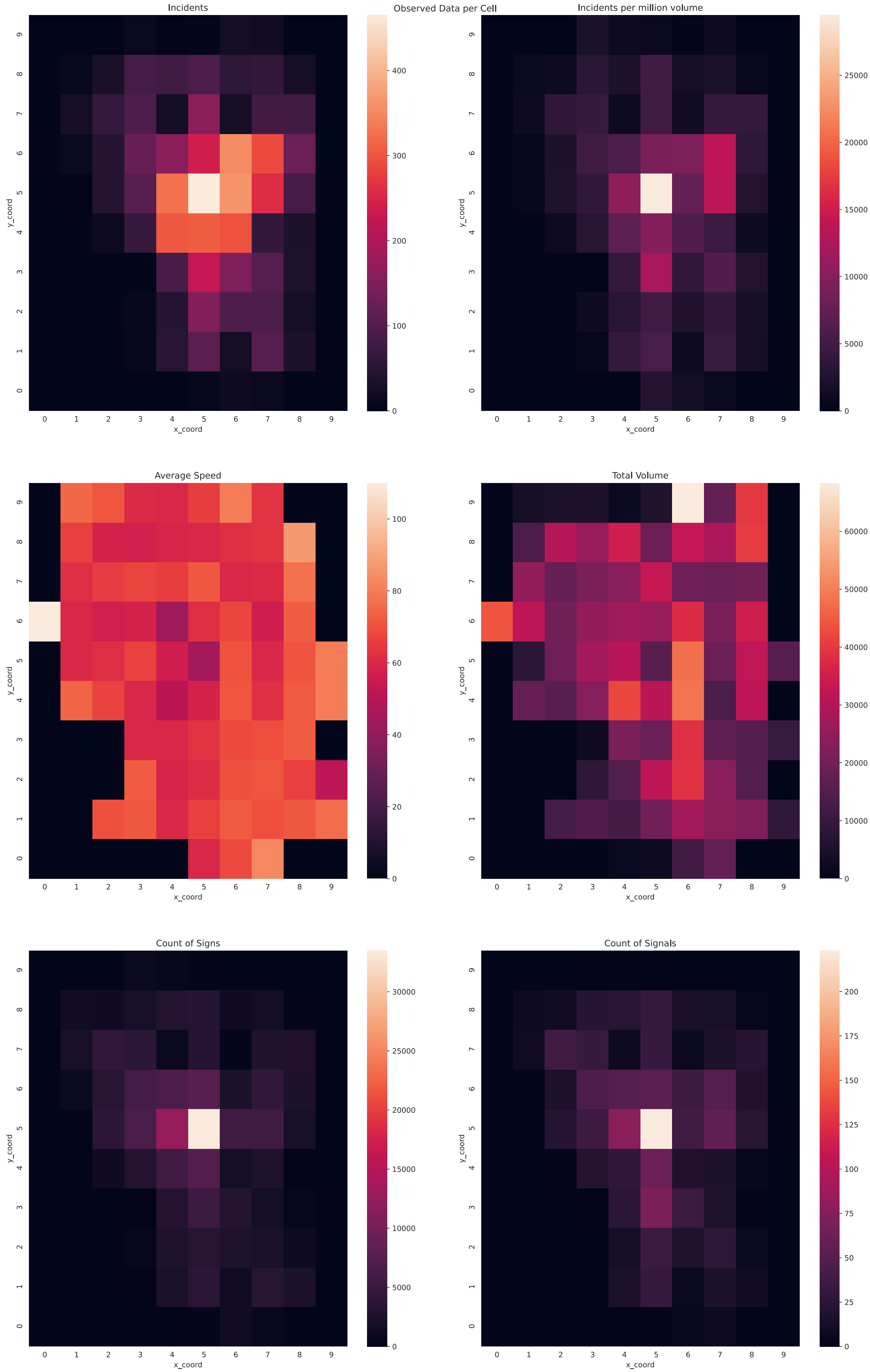
Cell Summary

In [9]:
```
cells = ctrl.get_frame('cells')
fig, ax = plt.subplots(nrows = 1, ncols = 1, figsize = (18,12)) #we want to plot side by side and to that figsize
sns.set_style('whitegrid')
sns.barplot(x=cells.index, y = cells['incident_count'], data =cells)
ax.set_xlabel("Cells")
ax.set_ylabel("Total Annual Incidents")
ax.set_title('Annual Incident vs. Cell')
ax.set_xticklabels(ax.get_xticklabels(), rotation=90, ha='right')
fig.show()
plt.savefig('./plots/annual incident vs cell.png')
```

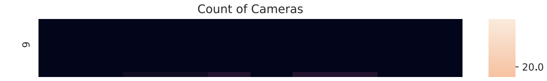


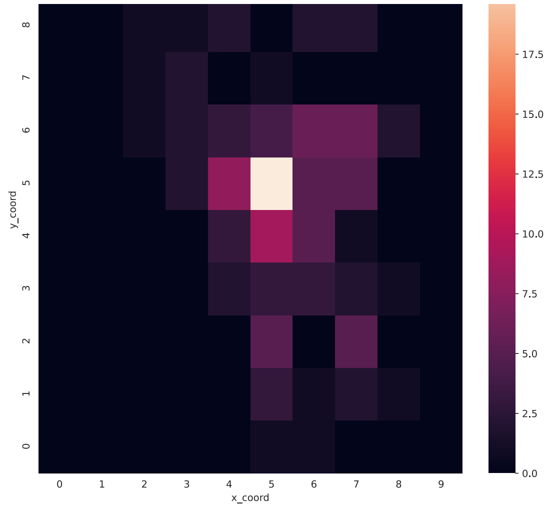
In [10]:
```
cells = ctrl.get_frame('cells')

fig, ( (inc_ax, inc_mil_ax), (speed_ax, vol_ax), (sign_ax, signal_ax
), (camera_ax, unsued_ax) ) = plt.subplots(nrows = 4, ncols = 2, figs
ize=(18,36))

ctrl.cell_heatmap('cells', 'incident_count', inc_ax, 'Incidents')
ctrl.cell_heatmap('cells', 'inc_per_mil_vol', inc_mil_ax, 'Incidents
 per million volume')
ctrl.cell_heatmap('cells', 'avg_speed',speed_ax, 'Average Speed')
ctrl.cell_heatmap('cells', 'volume_sum', vol_ax, 'Total Volume')
ctrl.cell_heatmap('cells', 'sign_count', sign_ax, 'Count of Signs')
ctrl.cell_heatmap('cells', 'signal_count', signal_ax, 'Count of Signa
ls')
ctrl.cell_heatmap('cells', 'camera_count',camera_ax, 'Count of Camera
s')
fig.suptitle("Observed Data per Cell")
fig.tight_layout(pad=5)
unsued_ax.axis('off')
fig.show()
plt.savefig('./plots/heatmaps.png')
```

Observed Data per Cell



Count of Cameras

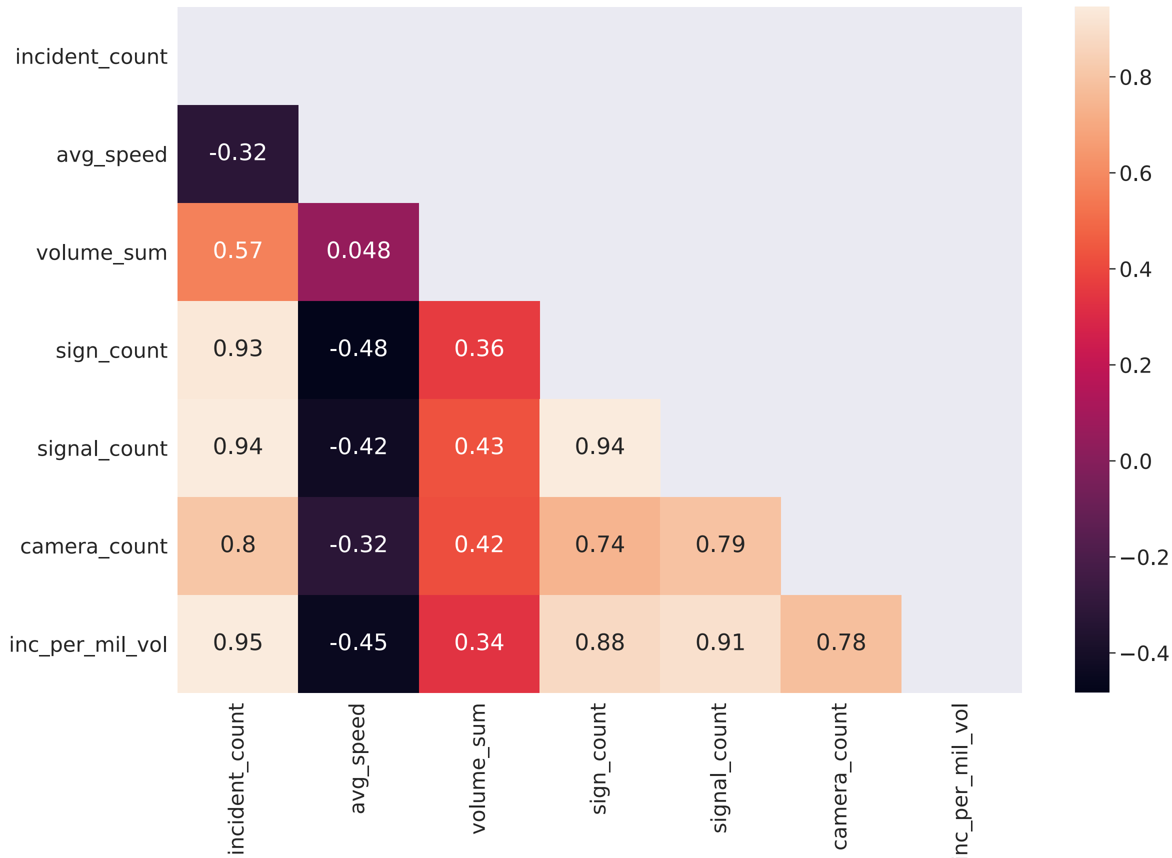




Correlations

Road Feature Correlations

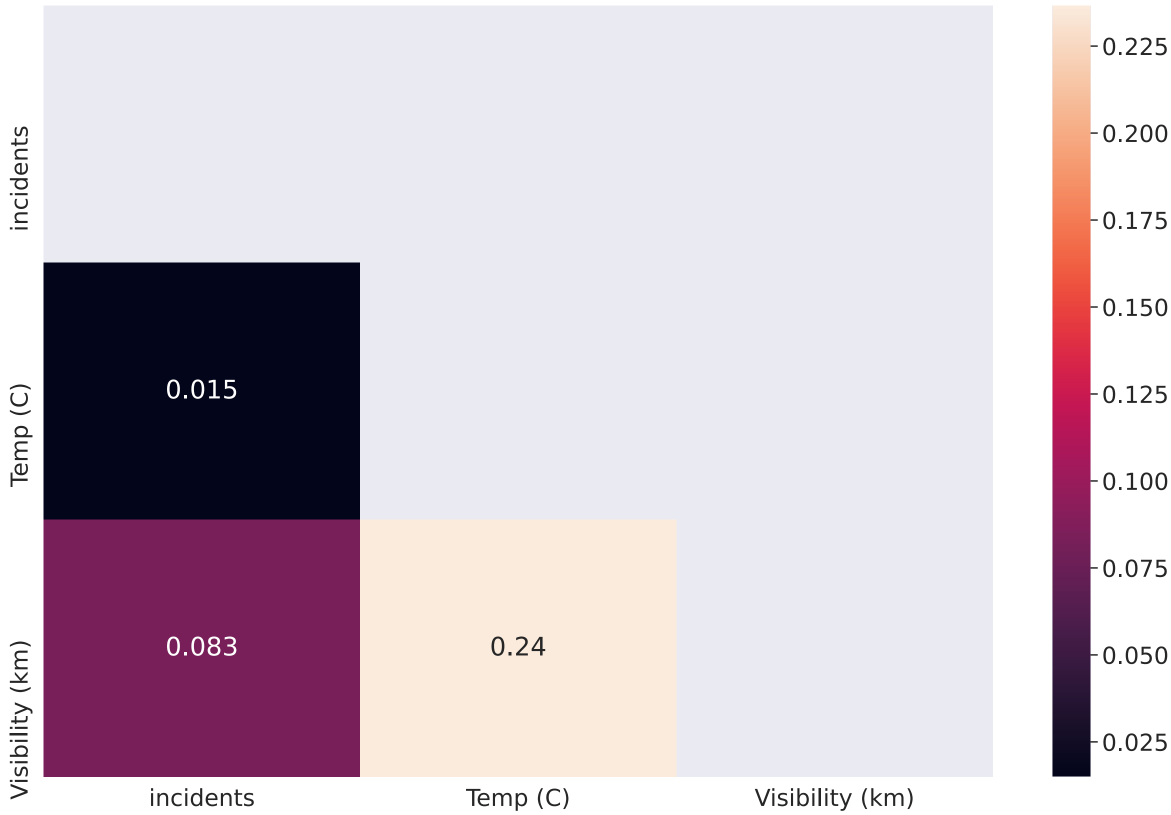
In [11]:
```
static_corr = ctrl.get_frame('cells')[ ['incident_count', 'avg_speed'
, 'volume_sum', 'sign_count', 'signal_count', 'camera_count', 'inc_pe
r_mil_vol'] ]
static_corr = static_corr.corr(method='spearman')
# display(static_corr['incident_count'])
sns.set(font_scale=1.8)
fig, ax = plt.subplots(nrows = 1, ncols = 1, figsize = (18,12))

mask = np.zeros_like(static_corr)
mask[np.triu_indices_from(mask)] = True
sns.heatmap(static_corr, annot= True, mask=mask)
fig.show()
plt.savefig('./plots/static correlations.png')
```



Time Series Correlations

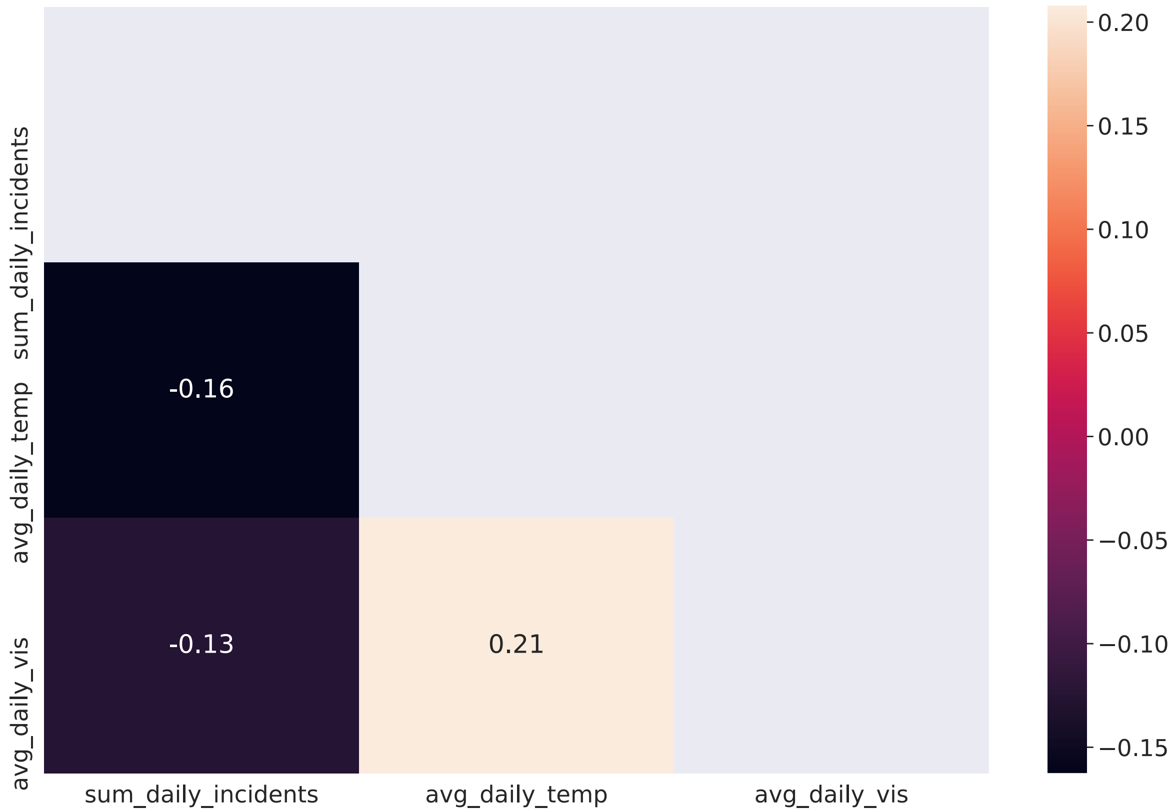
Hourly Correlations

In [12]:
```
hourly_corr = ctrl.get_frame('hourly')[ ['incidents', 'Temp (C)', 'Visibility (km)'] ]
hourly_corr = hourly_corr.corr(method='spearman')
fig, ax = plt.subplots(nrows = 1, ncols = 1, figsize = (18,12))

mask = np.zeros_like(hourly_corr)
mask[np.triu_indices_from(mask)] = True
sns.heatmap(hourly_corr, annot= True, mask=mask)
fig.show()
plt.savefig('./plots/hourly correlations.png')
```



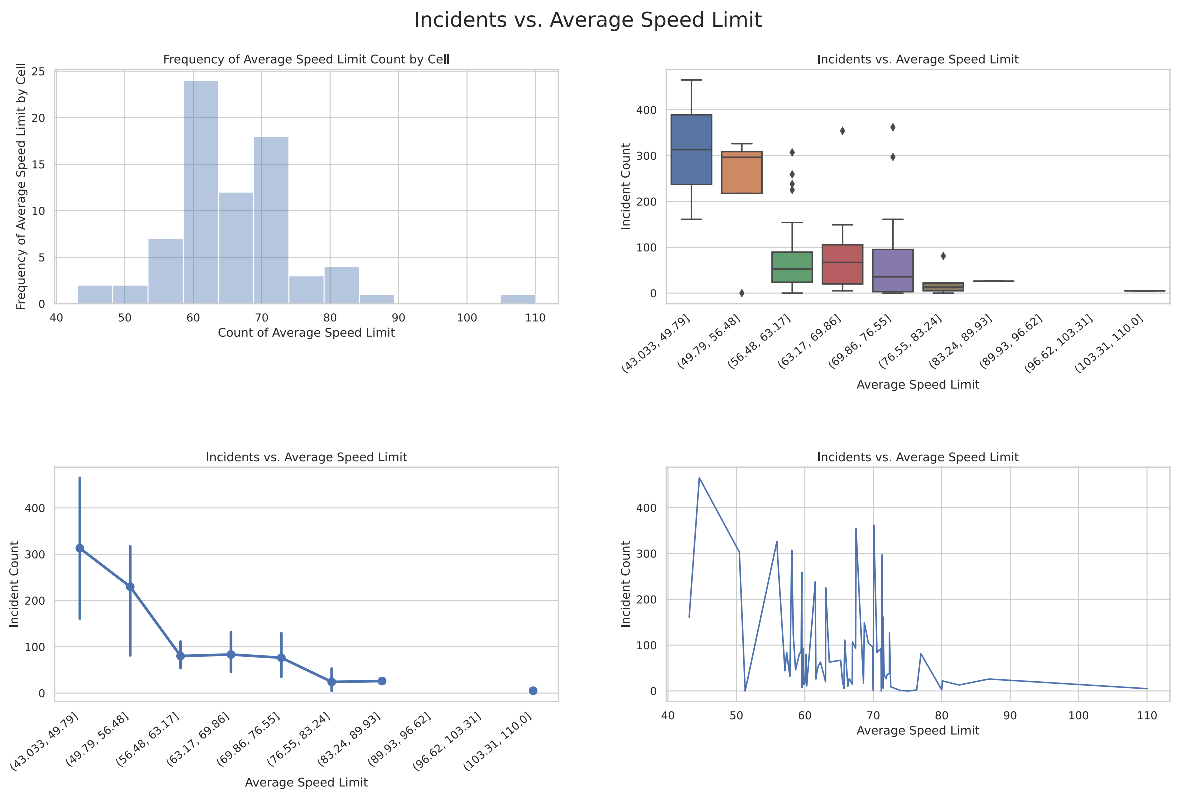
Daily Correlations

In [29]:
```
daily_corr = ctrl.get_frame('daily')[ ['sum_daily_incidents', 'avg_daily_temp', 'avg_daily_vis'] ]
daily_corr = daily_corr.corr(method='spearman')
sns.set(font_scale=1.8)

fig, ax = plt.subplots(nrows = 1, ncols = 1, figsize = (18,12))

mask = np.zeros_like(hourly_corr)
mask[np.triu_indices_from(mask)] = True
sns.heatmap(daily_corr, annot= True, mask=mask)
fig.show()
plt.savefig('./plots/daily correlations.png')
```



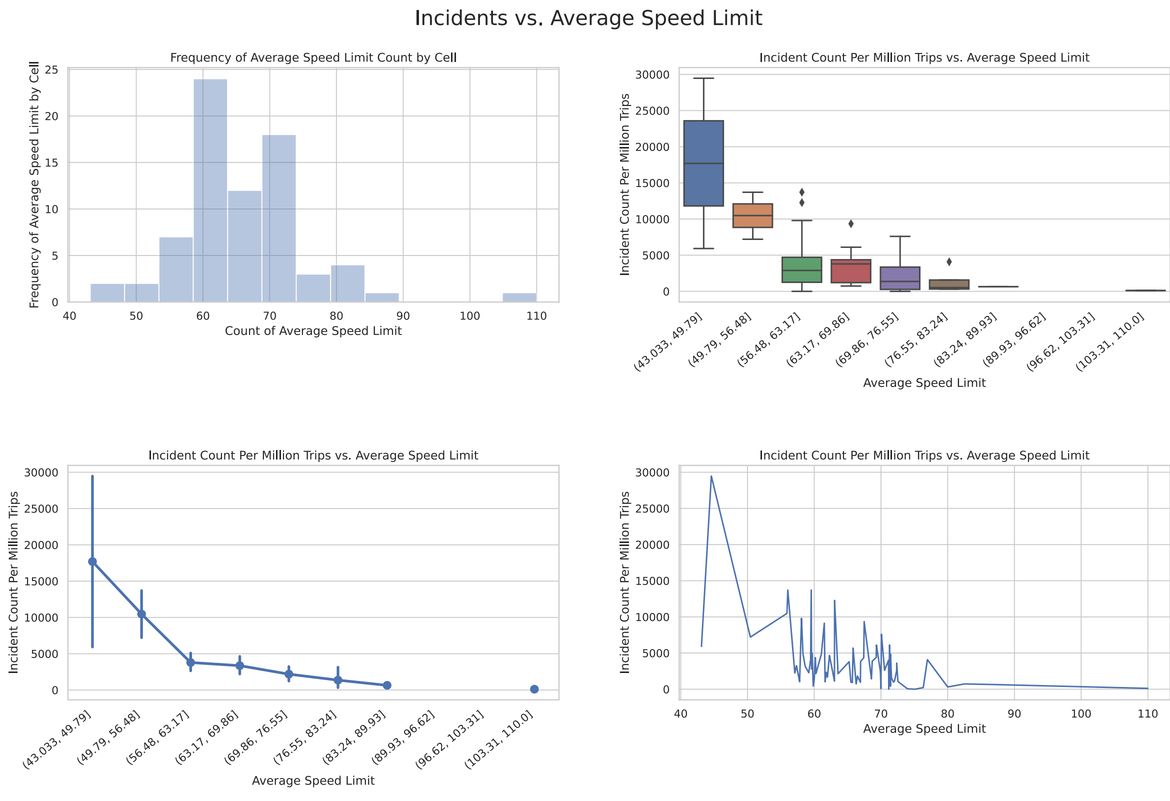
Incidents vs. Speed

In [14]:
```
sns.set(font_scale=1.0)
sns.set_style('whitegrid')
df =  ctrl.get_frame('cells')
target_text = 'Average Speed Limit'
target_col = 'avg_speed'
responding_col = 'incident_count'
x_label = 'Average Speed Limit'
y_label = 'Incident Count'
title = 'Incidents vs. Average Speed Limit'
binned = True
bin_col = 'speed_bins'

fig = ctrl.get_super_plot(df, target_text, target_col, responding_col
, x_label, y_label, title, binned, bin_col)
fig.show()
```



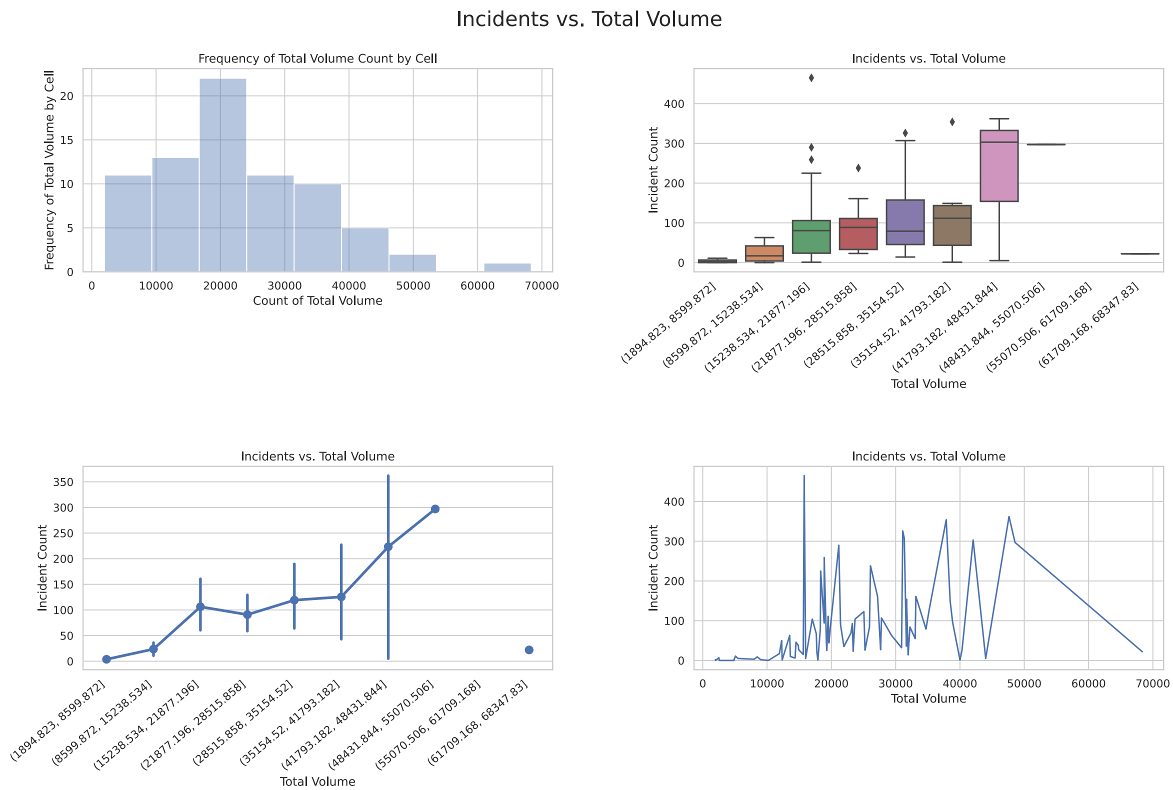
Incidents per million trips vs. Speed

In [15]:
```
sns.set(font_scale=1.0)
sns.set_style('whitegrid')
df =  ctrl.get_frame('cells')
target_text = 'Average Speed Limit'
target_col = 'avg_speed'
responding_col = 'inc_per_mil_vol'
x_label = 'Average Speed Limit'
y_label = 'Incident Count Per Million Trips'
title = 'Incident Count Per Million Trips vs. Average Speed Limit'
binned = True
bin_col = 'speed_bins'

fig = ctrl.get_super_plot(df, target_text, target_col, responding_col
, x_label, y_label, title, binned, bin_col)
fig.show()
```



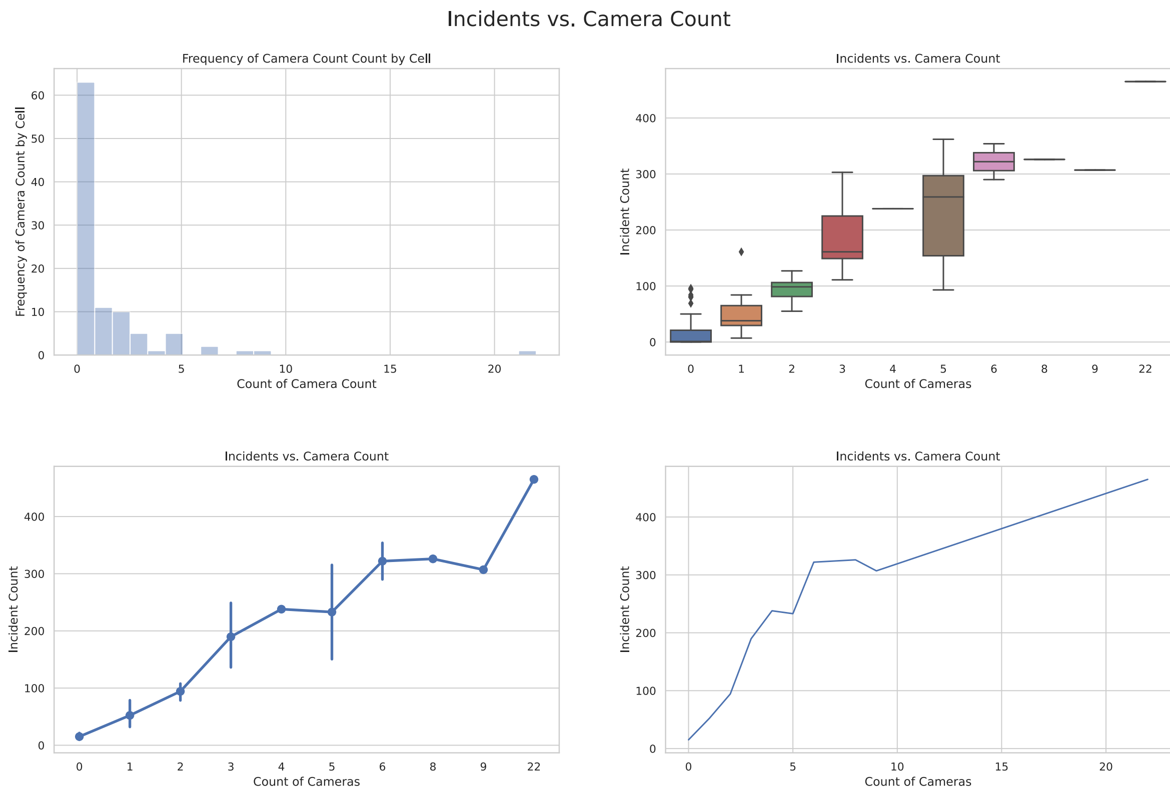
Incidents vs. Volume

In [16]:
```
df =  ctrl.get_frame('cells')
target_text = 'Total Volume'
target_col = 'volume_sum'
responding_col = 'incident_count'
x_label = 'Total Volume'
y_label = 'Incident Count'
title = 'Incidents vs. Total Volume'
binned = True
bin_col = 'volume_bins'

fig = ctrl.get_super_plot(df, target_text, target_col, responding_col
, x_label, y_label, title, binned, bin_col)
fig.show()
```



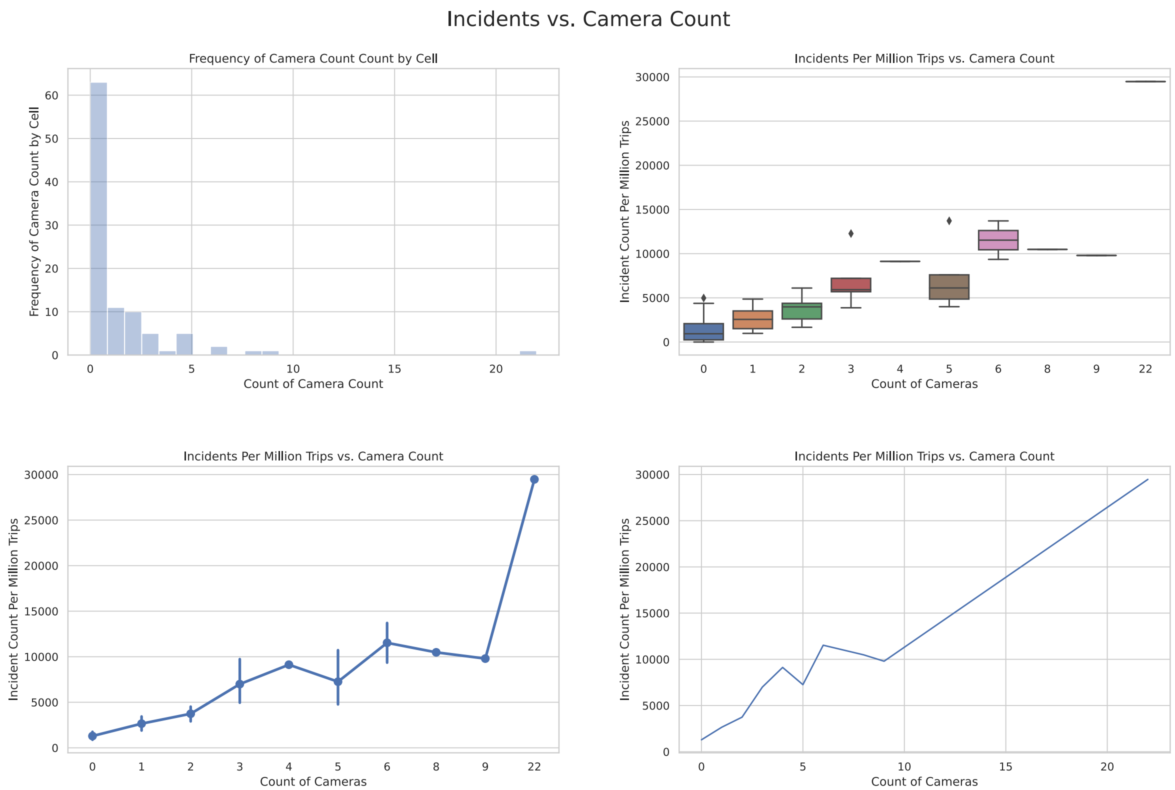
Incidents vs. Cameras

In [17]:
```
df =  ctrl.get_frame('cells')
target_text = 'Camera Count'
target_col = 'camera_count'
responding_col = 'incident_count'
x_label = 'Count of Cameras'
y_label = 'Incident Count'
title = 'Incidents vs. Camera Count'
binned = False

fig = ctrl.get_super_plot(df, target_text, target_col, responding_col
, x_label, y_label, title, binned, bin_col)
fig.show()
```



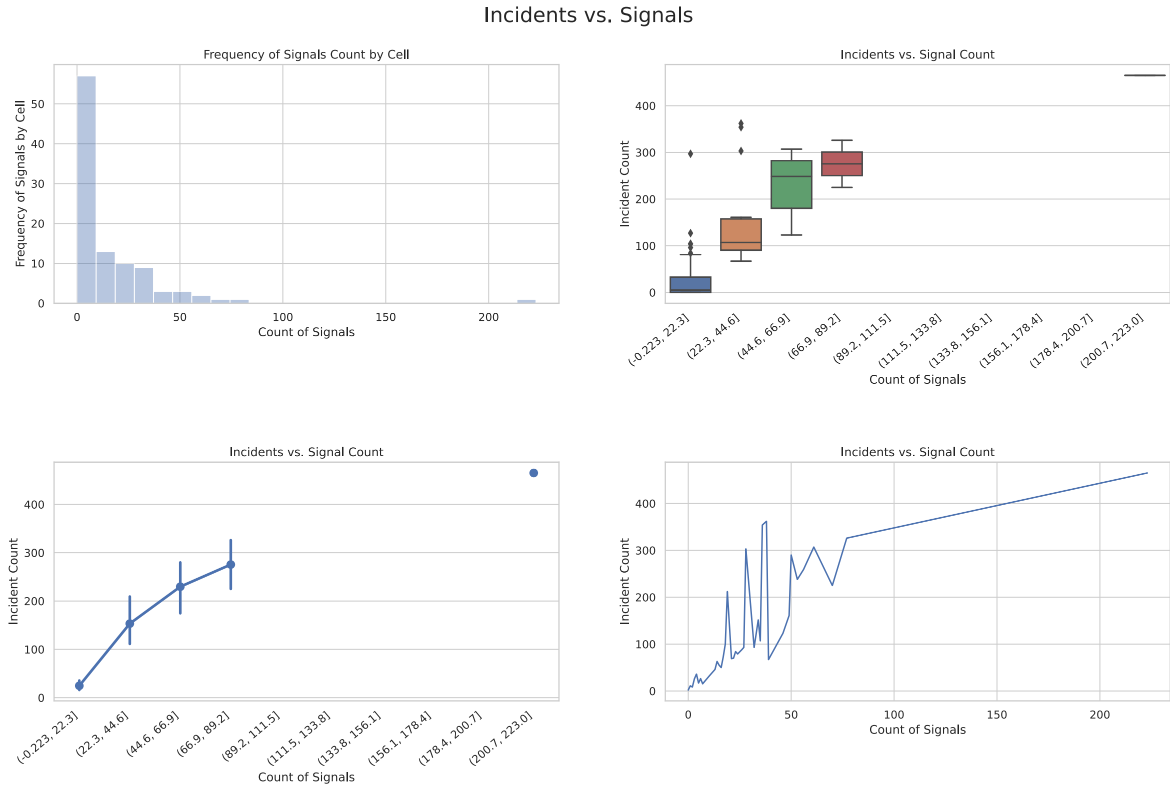
Incidents per million trips vs. Cameras

In [18]:
```
df =  ctrl.get_frame('cells')
target_text = 'Camera Count'
target_col = 'camera_count'
responding_col = 'inc_per_mil_vol'
x_label = 'Count of Cameras'
y_label = 'Incident Count Per Million Trips'
title = 'Incidents Per Million Trips vs. Camera Count'
binned = False

fig = ctrl.get_super_plot(df, target_text, target_col, responding_col
, x_label, y_label, title, binned, bin_col)
fig.show()
```



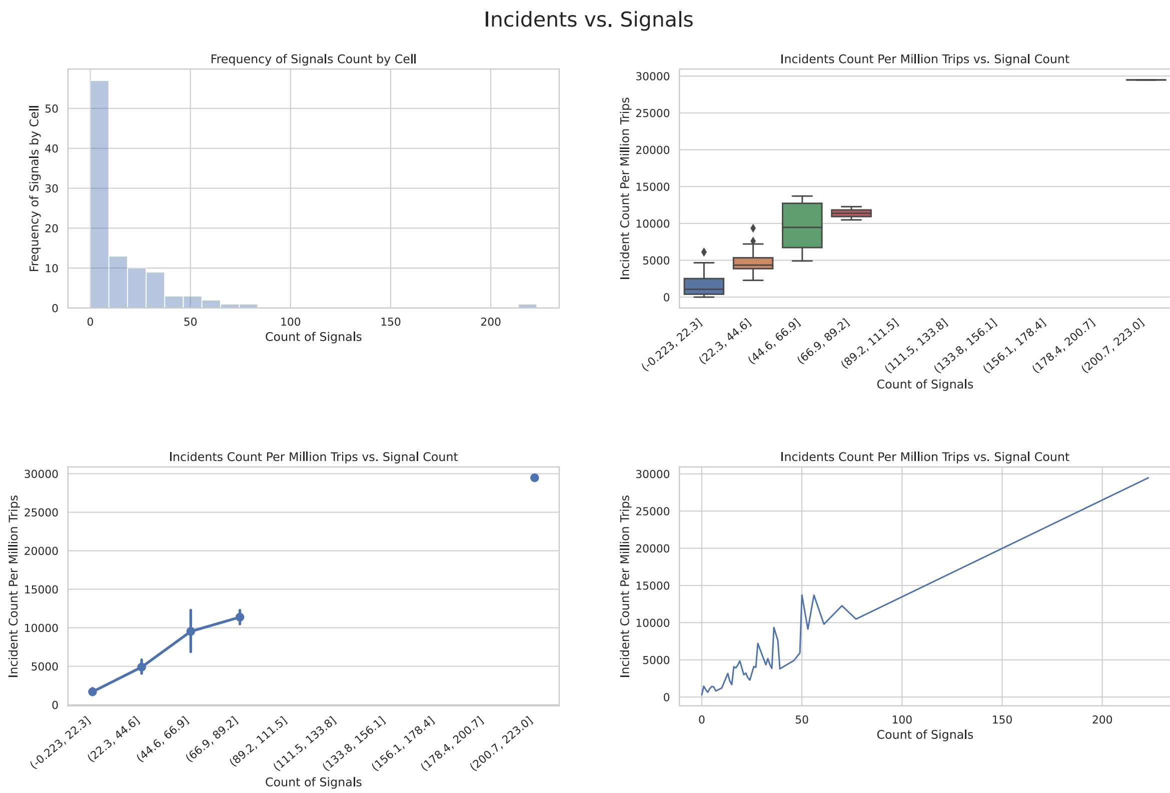
Incidents vs. Signals

In [19]:
```
df =  ctrl.get_frame('cells')
target_text = 'Signals'
target_col = 'signal_count'
responding_col = 'incident_count'
x_label = 'Count of Signals'
y_label = 'Incident Count'
title = 'Incidents vs. Signal Count'
binned = True
bin_col = 'signal_bins'

fig = ctrl.get_super_plot(df, target_text, target_col, responding_col
, x_label, y_label, title, binned, bin_col)
fig.show()
```



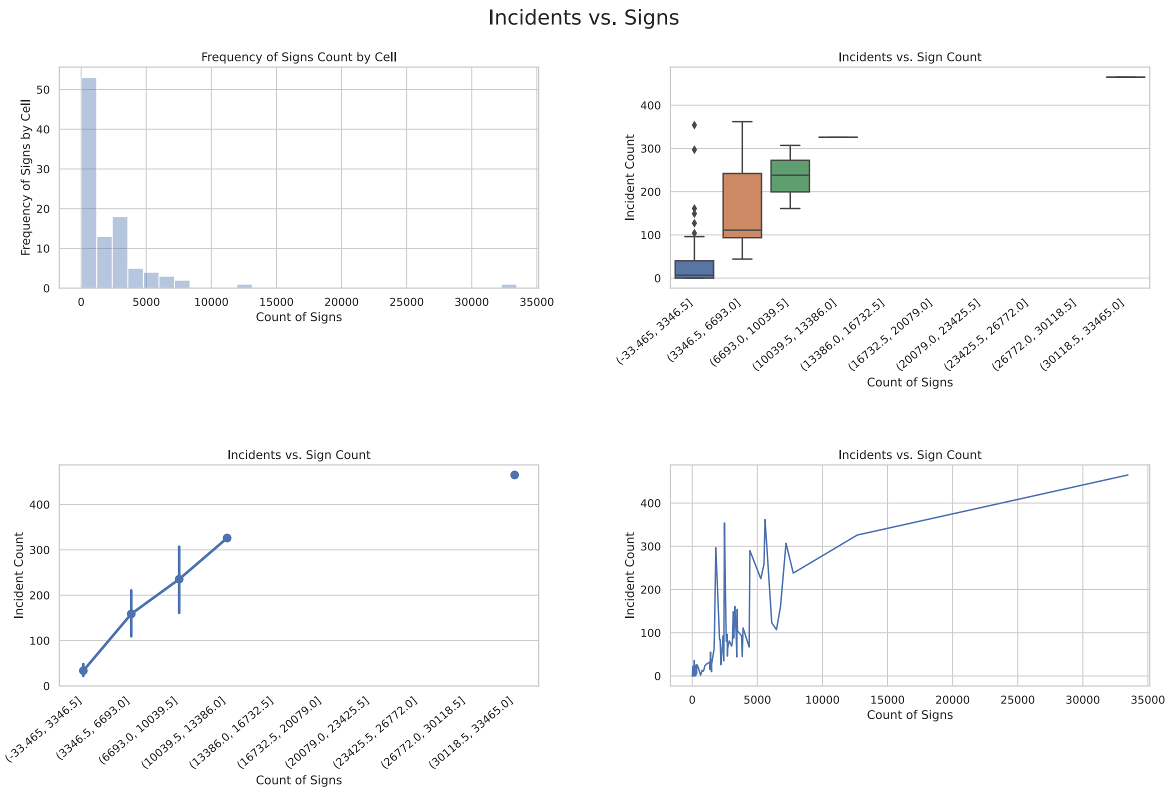
Incidents per million trips vs. Signals

In [20]:
```
df =  ctrl.get_frame('cells')
target_text = 'Signals'
target_col = 'signal_count'
responding_col = 'inc_per_mil_vol'
x_label = 'Count of Signals'
y_label = 'Incident Count Per Million Trips'
title = 'Incidents Count Per Million Trips vs. Signal Count'
binned = True
bin_col = 'signal_bins'

fig = ctrl.get_super_plot(df, target_text, target_col, responding_col
, x_label, y_label, title, binned, bin_col)
fig.show()
```



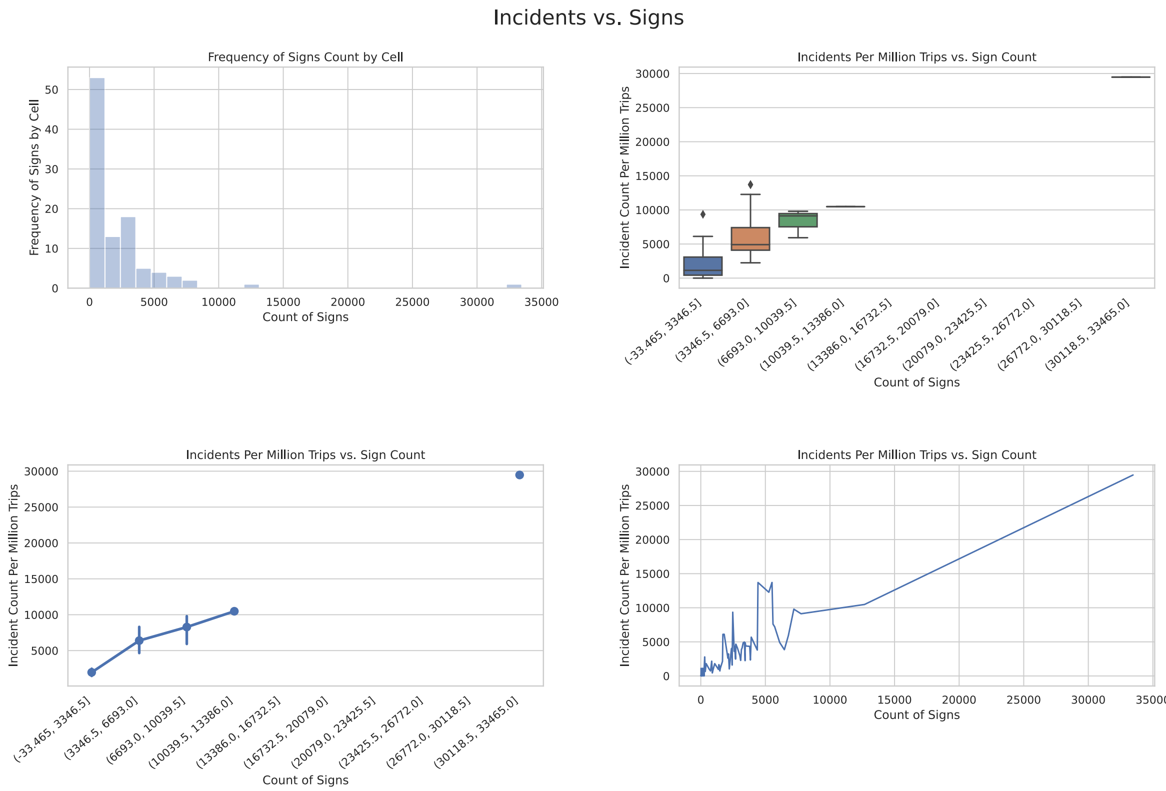
Incidents vs. Signs

In [21]:
```
df =  ctrl.get_frame('cells')
target_text = 'Signs'
target_col = 'sign_count'
responding_col = 'incident_count'
x_label = 'Count of Signs'
y_label = 'Incident Count'
title = 'Incidents vs. Sign Count'
binned = True
bin_col = 'sign_bins'

fig = ctrl.get_super_plot(df, target_text, target_col, responding_col
, x_label, y_label, title, binned, bin_col)
fig.show()
```



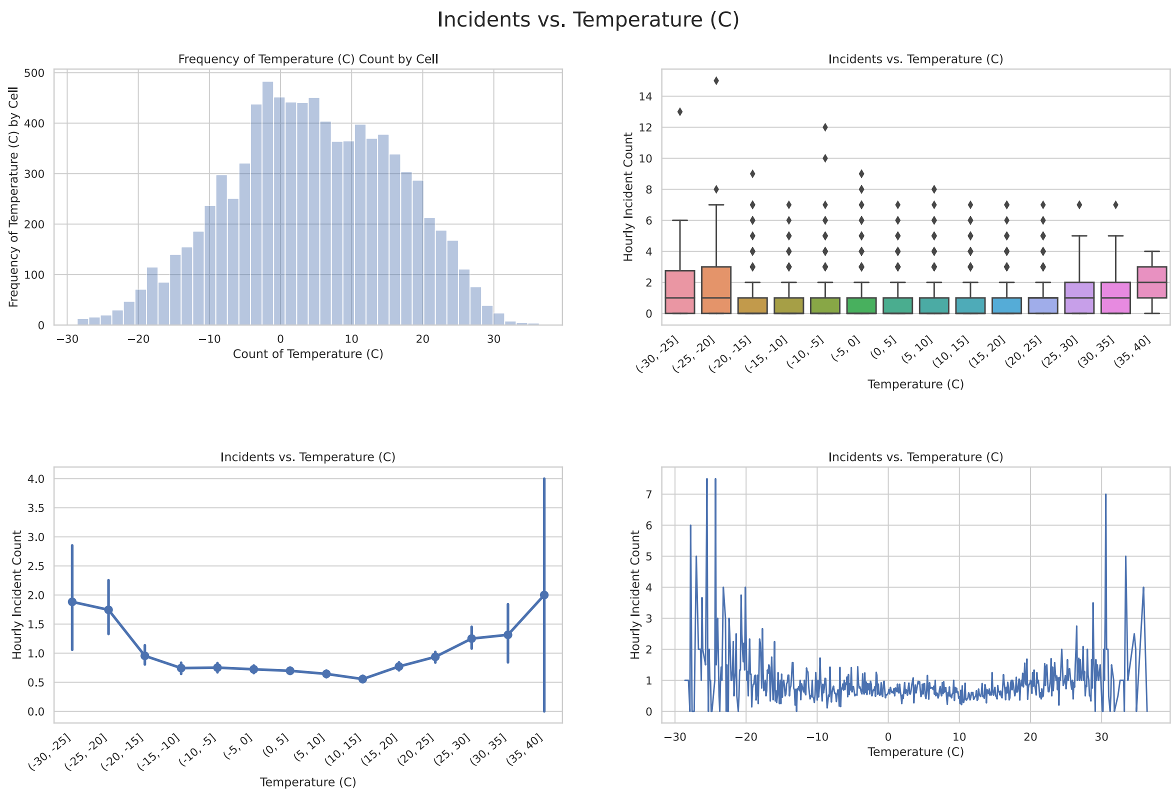
Incident Count Per Million Trip vs. Signs

```
In [22]: df =  ctrl.get_frame('cells')
target_text = 'Signs'
target_col = 'sign_count'
responding_col = 'inc_per_mil_vol'
x_label = 'Count of Signs'
y_label = 'Incident Count Per Million Trips'
title = 'Incidents Per Million Trips vs. Sign Count'
binned = True
bin_col = 'sign_bins'

fig = ctrl.get_super_plot(df, target_text, target_col, responding_col
, x_label, y_label, title, binned, bin_col)
fig.show()
```



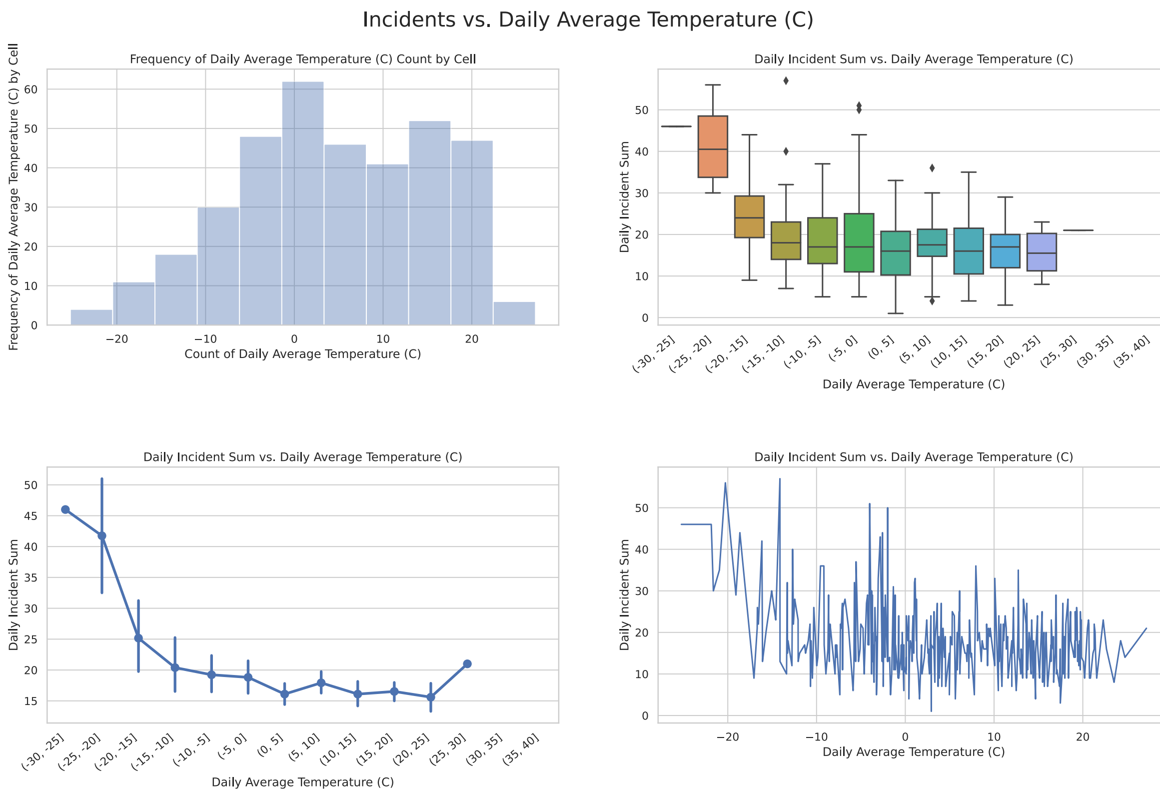
Incidents vs. Hourly Temperature

In [23]:
```
df = ctrl.get_frame('hourly')
target_text = 'Temperature (C)'
target_col = 'Temp (C)'
responding_col = 'incidents'
x_label = 'Temperature (C)'
y_label = 'Hourly Incident Count'
title = 'Incidents vs. Temperature (C)'
binned = True
bin_col = 'temp_bins'

fig = ctrl.get_super_plot(df, target_text, target_col, responding_col
, x_label, y_label, title, binned, bin_col)
fig.show()
```



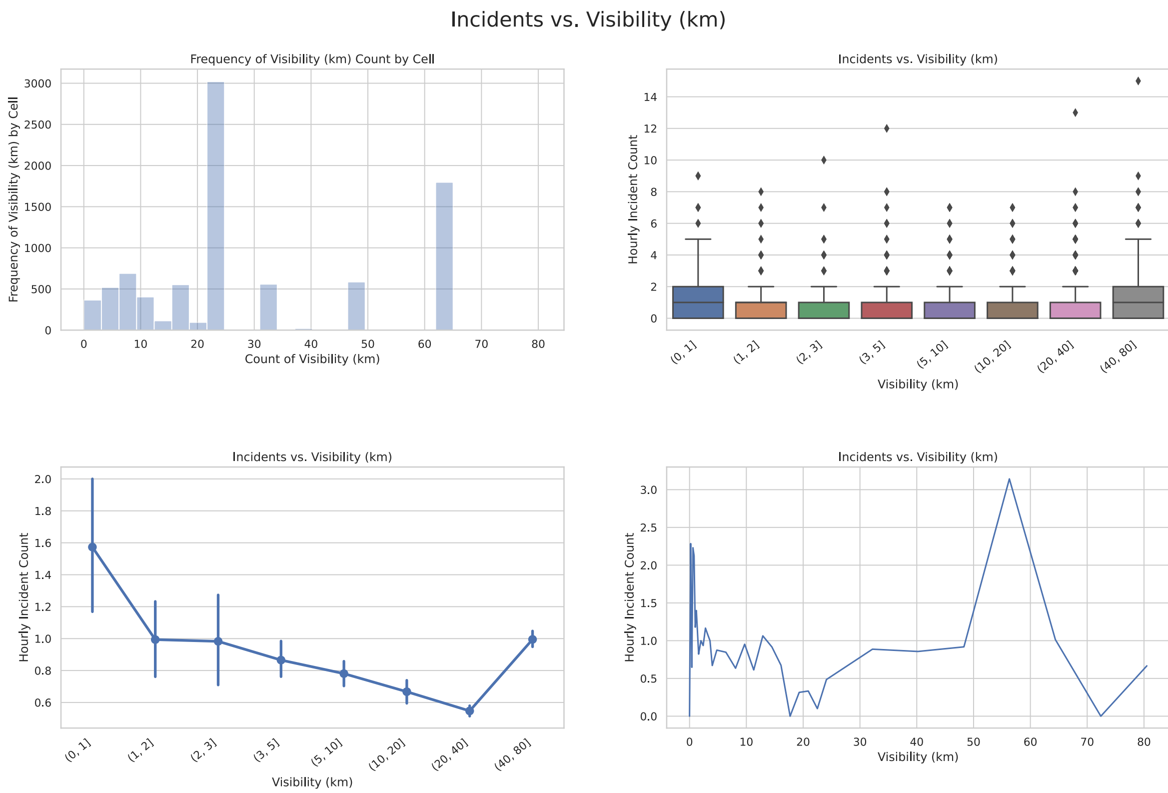
Incidents vs. Daily Average Temperature

In [24]:
```
df = ctrl.get_frame('daily')
target_text = 'Daily Average Temperature (C)'
target_col = 'avg_daily_temp'
responding_col = 'sum_daily_incidents'
x_label = 'Daily Average Temperature (C)'
y_label = 'Daily Incident Sum'
title = 'Daily Incident Sum vs. Daily Average Temperature (C)'
binned = True
bin_col = 'temp_bins'

fig = ctrl.get_super_plot(df, target_text, target_col, responding_col
, x_label, y_label, title, binned, bin_col)
fig.show()
```



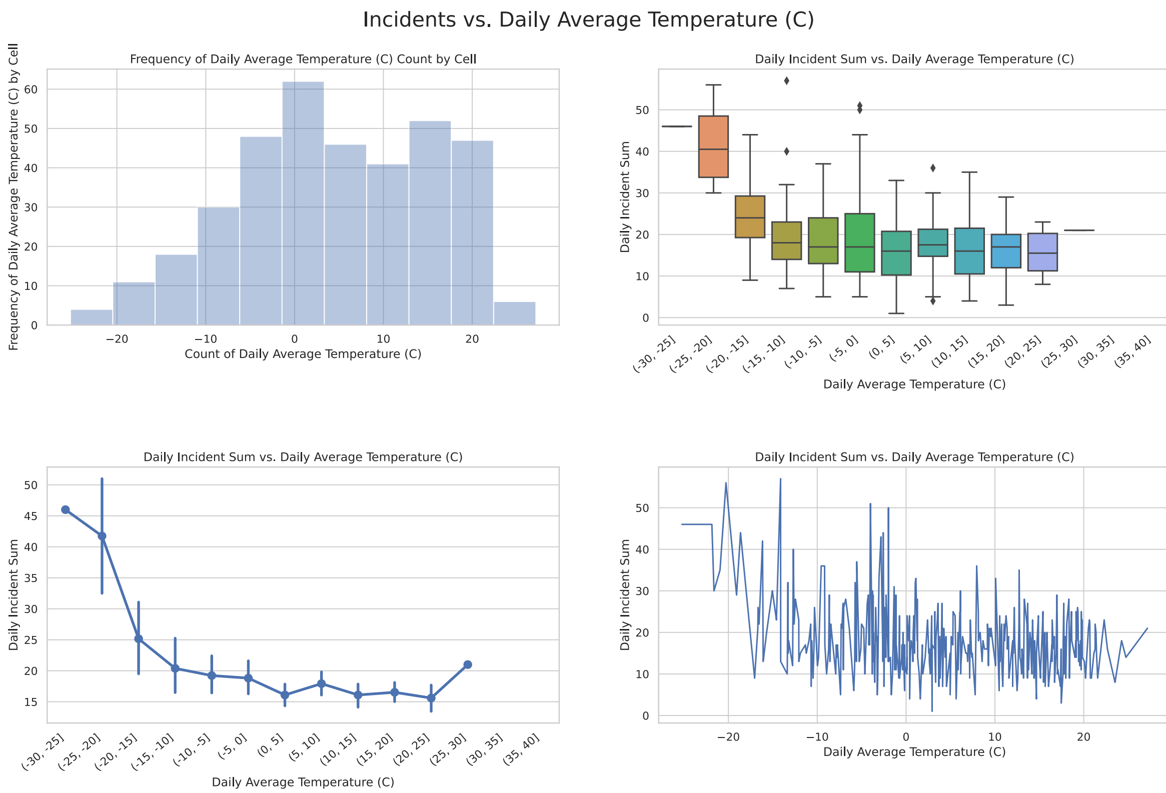
Incidents vs. Hourly Visibility

In [25]:
```
df = ctrl.get_frame('hourly')
target_text = 'Visibility (km)'
target_col = 'Visibility (km)'
responding_col = 'incidents'
x_label = 'Visibility (km)'
y_label = 'Hourly Incident Count'
title = 'Incidents vs. Visibility (km)'
binned = True
bin_col = 'vis_bins'

fig = ctrl.get_super_plot(df, target_text, target_col, responding_col
, x_label, y_label, title, binned, bin_col)
fig.show()
```



Incidents vs. Daily Average Visibility

In [26]:
```
df = ctrl.get_frame('daily')
target_text = 'Daily Average Temperature (C)'
target_col = 'avg_daily_temp'
responding_col = 'sum_daily_incidents'
x_label = 'Daily Average Temperature (C)'
y_label = 'Daily Incident Sum'
title = 'Daily Incident Sum vs. Daily Average Temperature (C)'
binned = True
bin_col = 'temp_bins'

fig = ctrl.get_super_plot(df, target_text, target_col, responding_col
, x_label, y_label, title, binned, bin_col)
fig.show()
```



Melt Freeze Analysis

In [27]:

```
'''
Cell calculates the number of hours during the year where a "melt-fre
eze' cycle occured per ctrl.melt_freeze(). Intent of this analysis to
is determine if melt-freeze cycles correlate with an increase in inci
dents.
'''
temps = ctrl.get_frame('hourly')['Temp (C)']

#See ctrl.melt_freeze() for a full description of below function. In
 this case, returns a boolean mask used to filter the hourly datafram
e for all hours with 8 hours of freeze event.
melt_freeze = ctrl.melt_freeze(temps, 0, 8)

freeze_temps = ctrl.get_frame('hourly')[melt_freeze]

inc_per_hour_freeze = freeze_temps['incidents'].sum() / freeze_temps[
'incidents'].size

inc_per_hour_typ = ctrl.get_frame('hourly')['incidents'].sum() / ctrl
.get_frame('hourly')['incidents'].size

#dataframe simply used to plot below, suprising results. May be due t
o people driving more slowly during a freeze or less traffic if freez
es typical occur at nigth during low traffic volumes.
data = pd.DataFrame({"Incidents per hour": [inc_per_hour_typ, inc_per
_hour_freeze], 'Total Incidents' : [ctrl.get_frame('hourly')['inciden
ts'].sum(),freeze_temps['incidents'].sum()], "Number of Hours" : [ctr
l.get_frame('hourly')['incidents'].size,freeze_temps['incidents'].siz
e]})

data.rename(index={0: 'Typical Hour', 1: 'Melt-Freeze Hours'}, inplac
e=True)

fig, ax = plt.subplots(nrows = 1, ncols = 1, figsize = (12,8), )
sns.barplot(x=data.index, y=data['Incidents per hour'])
ax.set_title('Incidents per hour vs. Typical hours and Freezing hour
s')
plt.savefig(f'./plots/meltfreeze.png')
fig.show()
```

```
There were 143 melt-freeze cycles in 2018!
```