Data Analysis of Canada Car Accident during 1999 - 2014

Term Project Report

Project Group #3, 3250-15

**Boris Korotkov**

**Chaewon Kim**

**Patricia Luo**

**Ryan He**

School of Continuing Studies, University of Toronto

April 2018

Table of Contents

[Project Overview 2](#_Toc510916853)

[Data Preparation 2](#_Toc510916854)

[What was the data source? 2](#_Toc510916855)

[How good was the data quality? 2](#_Toc510916856)

[How we procured source data? 3](#_Toc510916857)

[What tools/code we used to prepare it for analysis? 3](#_Toc510916858)

[What challenges we were facing? 3](#_Toc510916859)

[Analysis 5](#_Toc510916860)

[What trends, correlations and/or patterns we saw in the data? 5](#_Toc510916861)

[Some common consideration applied in analysis 5](#_Toc510916862)

[Individual analysis and the result we discovered: 5](#_Toc510916863)

[Conclusions 11](#_Toc510916864)

[What we learned from the data set? 11](#_Toc510916865)

[What …. 11](#_Toc510916866)

[Appendices 13](#_Toc510916867)

[A-1 sample dataset 13](#_Toc510916868)

[A-2 Sample Codes 13](#_Toc510916869)

# Project Overview

Our team analyzed the Canadian car accident data from 1999 to 2014 (obtained from Kaggle), and gained some insight from it.

Our team created an open repository on GitHub (link), including all project background information, codes (in Jupyter Note Book format) and reports for easy collaboration and tracking changes between team members.

We provided some suggestions based on our analysis at the end, hope which could be helpful to the audience for avoiding high risk driving conditions in the future.

# Data Preparation

Project group utilized the data from Kaggle, an open platform hosting a lot of real world data sets for predictive modeling and analytics competition. After sense check and cleansing of the data, our team members manipulated the datasets for individual analysis respectively.

## What was the data source?

The data source is from an open dataset hosted on Kaggle: [Canadian Car Accidents 1999-2014](https://www.kaggle.com/tbsteal/canadian-car-accidents-19942014/data). While the [full dataset](https://www.kaggle.com/tbsteal/canadian-car-accidents-19942014/downloads/NCDB_1999_to_2014.csv/1) is provided in CSV file (including header), which includes about 6 million records (15 years’ car collisions registered in Canada from 1999 to 2014), a [pdf file](https://www.kaggle.com/tbsteal/canadian-car-accidents-19942014/downloads/drivingLegend.pdf/1) provided the data dictionary. Each line of the dataset provide one registered collision record of one person involved in one collision, and there are 22 columns in the datasets, including:

* Collision level data elements (***year***, ***month***, ***day of week***, ***collision hour***, ***collision severity***, ***number of vehicles involved in collision***, ***collision configuration***, ***roadway configuration***, ***weather condition***, ***road surface***, ***road alignment***, and ***traffic control***);
* Vehicle level data elements (***vehicle sequence number***, ***vehicle type***, and ***vehicle model year***); and
* Person level data elements (***person sequence number***, ***person sex***, ***person age***, ***person position***, ***medical treatment required***, ***safety device used***, and ***road user class***).

## How good was the data quality?

The dataset is generally in good shape, including several missing numbers, and some unknown or not applicable data, which is well defined in the data dictionary. As an example, besides the 17 vehicle types defined in ***vehicle type*** (number in string), there are other four possible values:

* ‘NN’ not applicable, e.g., “dummy” vehicle record created for the pedestrian;
* ‘QQ’ choice other than defined vehicle types;
* ‘UU’ unknown values; and
* ‘XX’ jurisdiction does not provide the data.

Also, the data provided ***day of week*** instead of date, which brought some difficult to the team when trying to put a time series index for analysis (detail in below sections).

Further, since every data column contains some non-numeric value, pandas treated all columns as strings. Team has to manipulate them when trying to utilize numeric only features, e.g., scatter plot (detail in below sections).

## How we procured source data?

Team downloaded the dataset CSV file from the URL mentioned above, use system command to perform some sense check, then used *pandas.read\_csv* method loading the full dataset into a *dataframe*. See Appendix **[A-2.1]**.

## What tools/code we used to prepare it for analysis?

Team was able to perform all the data cleansing and preparation with python. The general cleansing is including:

* Non-numeric check for date time related columns;
* Drop the *nan* values as they represent insignificant samples; and
* Composed a *date* column from ***year*** and ***month*** information for possible usage in analysis.

See **[A-2.2]** for the sample code.

Note: Team comes up two different ways to generate the date column, and decides to use one or another in individual analysis to compare the difference. See below for detail.

## What challenges we were facing?

Team experienced following challenges in prepare the data, and overcame them after experiment and discussion:

* The dataset didn’t include a ***date*** but a ***day of week***. Team decide to combine ***year*** and ***month*** as the date information, but keep the ***day of week*** in its column for certain analysis.
* All values in dataset are in string type. Team decide to keep them in general dataset, and change the type when make a cloning slice, and proceed with further analysis accordingly.
* Considering some unknown values may represent significant samples, and/or bring meaningful insight, team decide to keep them in general dataset, and treat them differently in individual analysis instead. However, the unknown values in date time columns are dropped, as they represent tiny portion of the data, which is very insignificant and bringing difficulty for most of the analysis.
* Team experiments two different ways to engineering date column, and noted their advantage versus disadvantage. The *pandas.DatetimeIndex* method is faster, however the result includes a date of month value ‘1’, which is not accurately represent the collision record; the *pandas.PeriodIndex* with ‘M’ frequency accurately shows the year and month, but the method is about 10 times slower given the huge dataset our team was using. Team decides to use either way in individual analysis to compare the difference.

# Analysis

## What trends, correlations and/or patterns we saw in the data?

Sample sample sample sample sample

* List
* List
* List

Sample sample sample sample sample

## Some common consideration applied in analysis

Sample sample sample sample sample

* List
* List
* List

Sample sample sample sample sample

## Individual analysis and the result we discovered:

1. **Road Alignment**

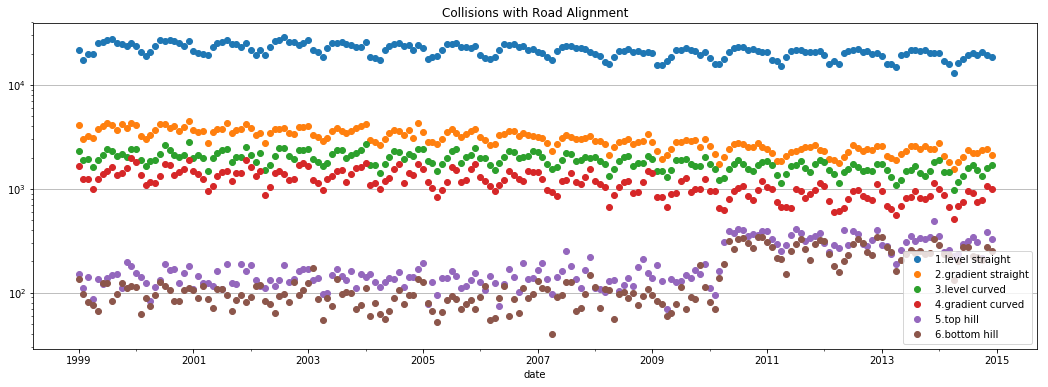
***Road alignment*** is a column indicating road straight/curving, and level/gradient status for certain collision record. We analyzed different relation between it and other variables.

* 1. Road Alignment & Time relation

Before doing each analysis, we took a slice from general dataset to keep the integrity of original data, which ease the debugging, and avoid reload the huge dataset into memory again (see sample code in **[A-2.3]**).

Because many rows of the dataset may represent collisions in same month, we sliced and grouped the data by time (*date* represent month) and road alignment column (*c\_raln*), then using *size* method aggregated the sum for each possible (*date, c\_raln*) pair into a two level indexed *pandas.Series*, then using *unstack* method to make it into a new *Dataframe* (sample code in **[A-2.4]**).

The Dataframe is plotted as below:

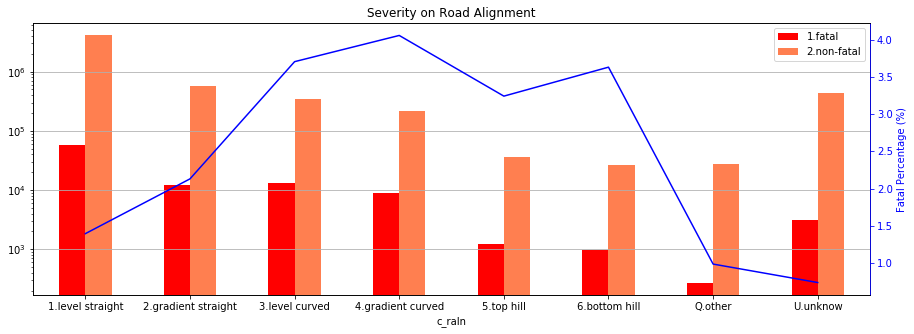


From it we saw most collisions happened on level and straight road, reason could be the vehicle of high speed, and driver not paying enough attention given the good road condition. Also noted clear seasonal pattern, see the auto-correlation analysis for seasonal patterns.

* 1. Road Alignment & Collision Severity

***Collision severity*** is a column indicating a collision is fatal or not.

In addition to generate a bar plot showing the relation between collision severity and road alignment, we also plot a second axis using *twinx* method of a *subplot* to show the fatal percentage for a collision happened in certain road alignment (the blue curve below with the y axis on right).



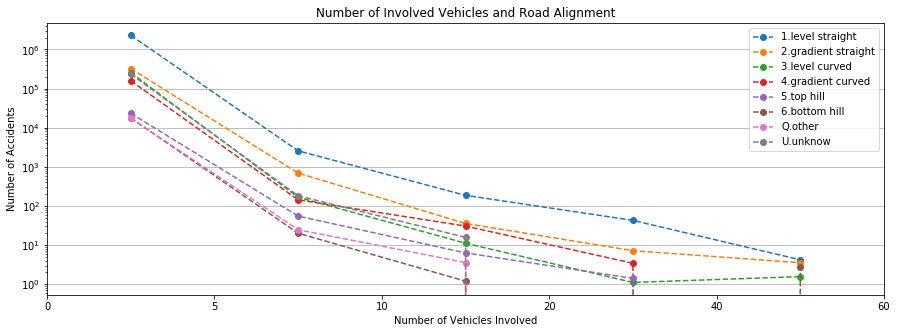
Fatal injury represent a very small portion of all accident (less than 4% in any given cases). It's also obvious that the level-straight road has less fatal injury, though the total number is much more than other cases; instead, the uneven condition (straight or curved) may produce severe accident. This could be caused by limited sight.

* 1. Road Alignment & Number of Vehicles Involved in Collision

***Number of vehicles involved in collision*** (*c\_vehs*) is a column indicating how many vehicles involved in certain records. In this analysis, all unknown value in this column are dropped as they didn’t provide insight.

To gain better understanding, we count the records for certain values in *c\_vehs*, and confirmed that every vehicle involved in one accident has its own record (e.g., there are 58 records in the dataset have 57 vehicles involved, the outstanding one could be a pedestrian), thus we calculated the portion of each accident, and put them in a new column in order to project the number of accidents that involved certain number of vehicles, by each road alignment (sample code in **[A-2.5]**).

We also binned the *c\_vehs* value into several bins to show the trend instead of focusing on each certain number, and plot in a cleaner graph below (sample code in **[A-2.5]**):

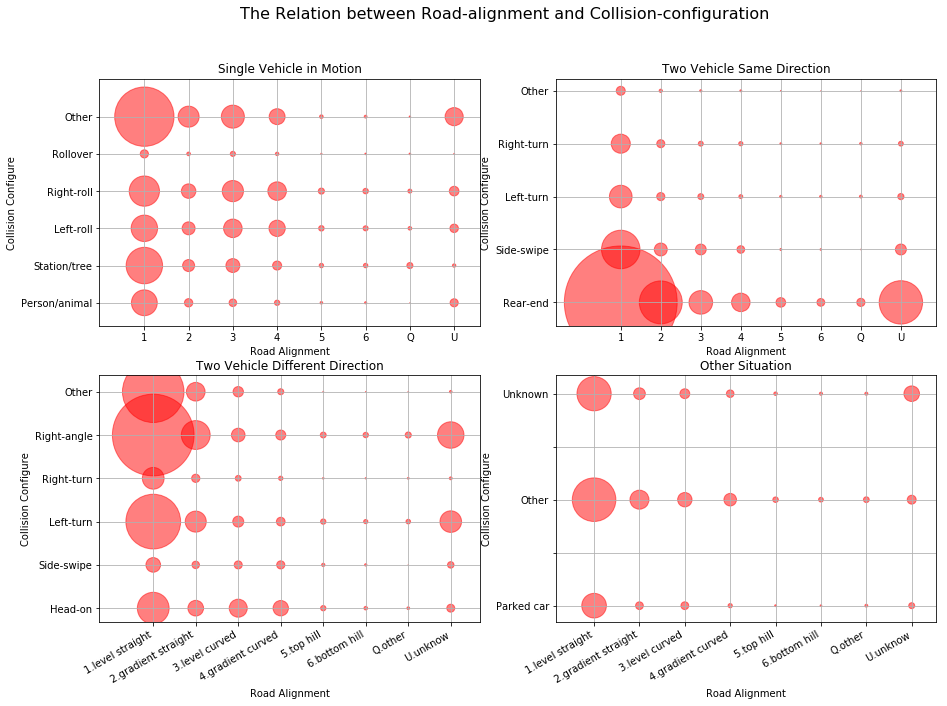


Majority accident are involving less than 5 cars. As the plots are very similar for different road alignments, no specific relationship observed between the two variables.

* 1. Road Alignment & Collision Configuration

***Collision configuration*** (*c\_conf*) column indicates one or two vehicle involved in collision, their motion, direction and relative position. As we want to plot them in scatter graph, which only supports numeric axials, all nonnumeric value of involved columns are replaced, and type is changed to integer.

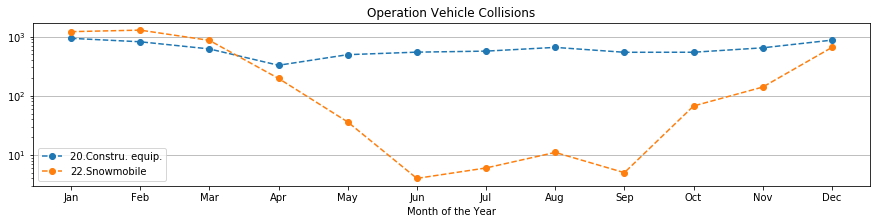
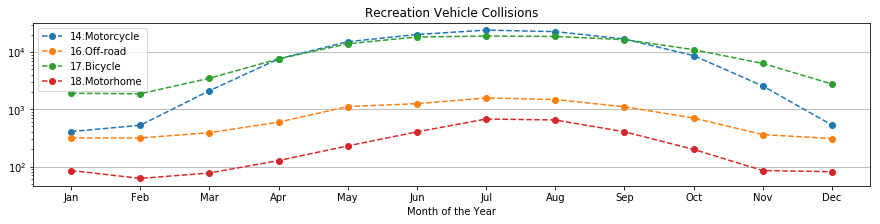
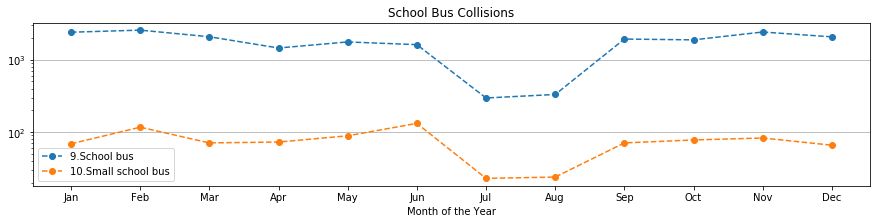
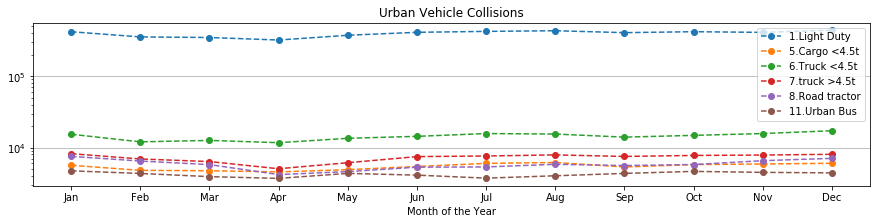
Considering the complexity of collision configurations, we also grouped them into four categories by number of vehicles and their directions, and plot them into four subplots.



The size of each dot indicate the collision numbers in each setting, same scale factor used for all subplots, which makes the number in each plot comparable to its neighbour. In all cases, clear trend shows that the collision chance increases when road-alignment getting better, due to higher speed and less caution. Except the unknown or "other" conditions, we can see in each cases:

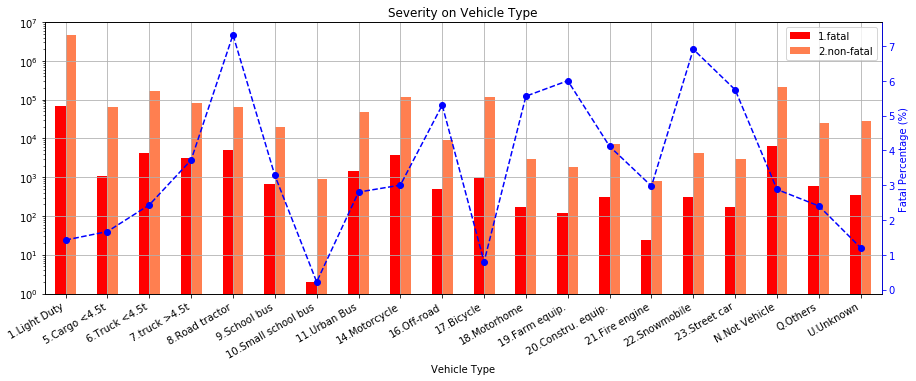
* Most collision for one car accident is hitting a static object, followed closely by right and left roll on to shoulder.
* For two car in same direction, most accidents are rear-end collision, which clearly increase along road-alignment.
* For two car in different direction, right-angle collision is huge, guess is in intersection. Left turn also post a lot risk, but right-turn is very low, even lower than head-on collision. This may be a result of the inherent complexity of left turn.

1. **Vehicle Type**
   1. Vehicle Type & Time Relation



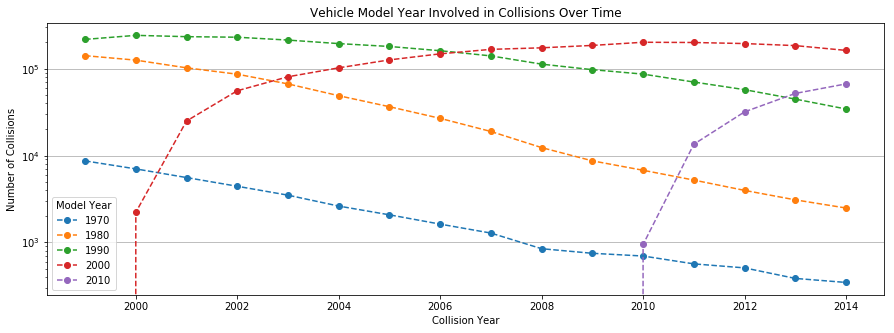
By looking at the plots above, clear pattern shown in different vehicle types:

* Vast majority collisions are caused by light duty vehicles, followed by light truck, motorcycle and bicycle. This is due to the dominant population of the personal vehicles, and the unprofessional nature of the drivers.
* School bus involved collisions happened much less in July and August, due to summer vacation.
* More accident related to bicycle, motorcycle, off-road vehicle and motorhome happened from April to October, as they are used for outdoor activities during summer time; in opposite, snow-mobile collisions are more from October to April.
  1. Vehicle Type & Severity



From the figure above we see most fatal collisions are caused by light-duty vehicles, followed by motorcycle, road tractor, and trucks. However, the highest percentage of fatal accidents is caused by road tractor and Snowmobile (about 7%), and followed by street car, farm equipment and motorhome (around 6%). Also noticed that small school bus produces least fatal collisions, and bicycle number is also non-surprisingly low.

1. **Vehicle Model Year**
   1. Vehicle Model Year & Time Relation



As the data covers collisions from 1999 to 2014, most cars involved in collisions are made in recent thirty years. Clear trend observed that the model getting involved in collision peaks in its 10 years' age, which could be a result of the market share growth and the reliability drops for old vehicles.

# Conclusions

## What we learned from the data set?

1. **Road Alignment and Accident**

From the analysis we performed above, following observations:

* Most collisions happened in level and aligned road. Drivers should pay more attention when drive on good road condition, and follow speed limit.
* However, bad road alignment may post higher risk of fatal accident. Drivers need to anticipate possible outcome when the view is not so wide, and always be cautious to avoid serious injury as possible.
* Rear-end collision is the most popular collision type. Keep enough space to the car in front of you, especially on well aligned road.
* Right-angle collision and left-turn collision are also popular, wait one more second before turning your wheel, or rushing into the intersection.

1. **Vehicle Type and Accident**

Clear seasonal patterns observed for certain types:

* School bus involved in collisions out of summer vacation (July and August);
* Vehicles for out-door activities (i.e., bicycle, motorcycle, off-road vehicle and motorhome) involved in collisions mostly during summer time (April to October);
* In opposite, snow-mobile collisions are more from October to April.

While light duty vehicles, light trucks, motorcycles and bicycles involved in most of the collisions, most fatal collision ratio observed from following types:

* Road tractor and Snowmobile is around 7%;
* Street car, farm equipment and motorhome is about 6%.

If you see those vehicle when you drive, try stay away from them.

1. **Vehicle Model Year and Accident**

As the collision rate peaks in a vehicle model’s 10th year, maybe you should consider a new car when it get into that age to reduce the risk of an accident.

## What ….

Sample sample sample sample sample

* List
* List
* List

Sample sample sample sample sample

# Appendices

## A-1 sample dataset

* Data dictionary:



* Sample data:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **C\_YEAR** | **C\_MNTH** | **C\_WDAY** | **C\_HOUR** | **C\_SEV** | **C\_VEHS** | **C\_CONF** | **C\_RCFG** | **C\_WTHR** | **C\_RSUR** | **C\_RALN** | **C\_TRAF** | **V\_ID** | **V\_TYPE** | **V\_YEAR** | **P\_ID** | **P\_SEX** | **P\_AGE** | **P\_PSN** | **P\_ISEV** | **P\_SAFE** | **P\_USER** |
| 1999 | 1 | 1 | 20 | 2 | 2 | 34 | UU | 1 | 5 | 3 | 3 | 1 | 6 | 1990 | 1 | M | 41 | 11 | 1 | UU | 1 |

## A-2 Sample Codes

* **[A-2.1]** load the data into a *pandas.Dataframe* (in “G3\_Project\_Ryan.ipynb”).

|  |
| --- |
| accident = pd.read\_csv('NCDB\_1999\_to\_2014.csv', engine = 'python')  *#replace header to lower case*  accident.columns = accident.columns.map(lambda x: x[:].lower()) |

* **[A-2.2]** clean and prepare the data for analysis (in “G3\_Project\_Ryan.ipynb”).

|  |
| --- |
| *#Check the non-numeric values in date-time columns*  accident[accident['c\_mnth'].str.contains('[^0-9]')|  accident['c\_wday'].str.contains('[^0-9]')|  accident['c\_hour'].str.contains('[^0-9]')]  *#Make a copy "df" for further analysis, avoid mess up the original data "accident".*  *#Remove all special values (unknown to us) in date-time columns, prepare for using*  *# time series as index.*  df = accident[:]  df[['c\_mnth','c\_wday','c\_hour']]=df[['c\_mnth','c\_wday','c\_hour']].\  replace('[^0-9]+',np.nan,regex=True)replace('[^0-9]+',np.nan,regex=True)  df.dropna(axis=0,subset=['c\_mnth','c\_wday','c\_hour'],inplace=True) |

* **[A-2.3]** make a copy of the dataset for easy debugging and faster processing (in “G3\_Project\_Ryan.ipynb”).

|  |
| --- |
| *#Slice the columns I would analyse from original dataset.*  dfa = df.loc[:,['date','c\_year','c\_mnth','c\_wday','c\_hour',  'c\_sev','c\_vehs','c\_conf','c\_wthr','c\_raln',  'v\_type','v\_year']]  *# Make a slice only focusing on c\_raln: road alignment.*  df1 = dfa.loc[:,['date','c\_raln']] |

* **[A-2.4]** aggregate by a pair, and reform into a *Dataframe* (in “G3\_Project\_Ryan.ipynb”).

|  |
| --- |
| *#Aggregate the events by month into a summarized dataframe.*  df2 = df1.groupby(['date','c\_raln']).size().unstack() |

* **[A-2.5]** calculate accident portion for each collision, and bin the numbers of vehicle involved in each collision (in “G3\_Project\_Ryan.ipynb”).

|  |
| --- |
| *#Calculate accident portion of each event, considering 58 records that have 57 vehicle*  *# involved, that should be just one accident, every relevant record should be only about 1/57.*  df1['acc']= 1/df1.c\_vehs  *#Categorize number of involved vehicles into bins.*  bins = [0,5,10,20,40,60]  df1['cats'] = pd.cut(df1.c\_vehs,bins) |

* **[A-2.4]** aggregate by a pair, and reform into a *Dataframe* (in “G3\_Project\_Ryan.ipynb”).

|  |
| --- |
| *#Aggregate the events by month into a summarized dataframe.*  df2 = df1.groupby(['date','c\_raln']).size().unstack() |

* **[A-2.4]** aggregate by a pair, and reform into a *Dataframe* (in “G3\_Project\_Ryan.ipynb”).

|  |
| --- |
| *#Aggregate the events by month into a summarized dataframe.*  df2 = df1.groupby(['date','c\_raln']).size().unstack() |

* **[A-2.4]** aggregate by a pair, and reform into a *Dataframe* (in “G3\_Project\_Ryan.ipynb”).

|  |
| --- |
| *#Aggregate the events by month into a summarized dataframe.*  df2 = df1.groupby(['date','c\_raln']).size().unstack() |

* **[A-2.4]** aggregate by a pair, and reform into a *Dataframe* (in “G3\_Project\_Ryan.ipynb”).

|  |
| --- |
| *#Aggregate the events by month into a summarized dataframe.*  df2 = df1.groupby(['date','c\_raln']).size().unstack() |

* **[A-2.4]** aggregate by a pair, and reform into a *Dataframe* (in “G3\_Project\_Ryan.ipynb”).

|  |
| --- |
| *#Aggregate the events by month into a summarized dataframe.*  df2 = df1.groupby(['date','c\_raln']).size().unstack() |

* **[A-2.4]** aggregate by a pair, and reform into a *Dataframe* (in “G3\_Project\_Ryan.ipynb”).

|  |
| --- |
| *#Aggregate the events by month into a summarized dataframe.*  df2 = df1.groupby(['date','c\_raln']).size().unstack() |