COGS9: Introduction to Data Science
Final Project
Due date: 2020-12-18 23:59:59
Grading: 10% of overall course grade. 40 points total.
Completed as a group. One submission per group on Gradescope.

Group Member Information:

First Name	Last Name	PID
Yunfan	Long	A16665173
Muchan	Li	A15887258
Kening	Li	A16631786
Dingrui	Wang	A16586496
Guanxing	Chen	A16632401

1 . Data Science Question

Under the condition of deployment of driver or passenger airbags for involved parties, how well does Ford-manufactured vehicles that are of model year later than or equal to year 2000 protect their drivers and passengers from injuries compared to those of Toyota-manufactured counterparts in collision accidents in California during year 2018?

---

2. Hypothesis

Ford-manufactured vehicles that are of model year later than or equal to year 2000 protect their drivers and passengers better from injuries compared to those of Toyota-manufactured vehicles in collision accidents under the deployment of airbag situations in year 2018.

---

3. Justification

We choose to focus our research subjects on Ford and Toyota vehicles because after inspecting the “parties” dataset, we found that vehicles from these two manufactuers account for the largest proportions, each taking up 13% of all collision vehicles. We then believe that performing an inferential analysis about Ford and Toyota vehicles’ safety performance would be practically meaningful and beneficial since there are people’s well being at stake. We decide to investigate the severity of injuries under the deployment-of-airbags condition because this is the situation where lives are most critically at risk and the safety performances of the vehicles involved matter more than ever. Lastly, we narrowed our focus to collisions that have happened in the state of California because it is the scope that our dataset--and its original source--California Highway Patrol--provide us with; we focused the time-stamp of all our data to include only year 2018 because we assume one whole year worth of data and records should be sufficiently representative of any pattern that we will detect and these data should have been preserved fairly well given its recency. Moreover, 2018 data should be mostly free from the disturbances of

COVID-19 pandemic compared to 2019 and 2020 and thus considered just as reliable as data from years before that.

---

4. **Background information**

When customers purchase a vehicle among thousands of brands, they try to pick the car that matches their needs the most. Safety is one of the important factors customers will consider during the buying process. In 2020 TOP SAFETY PICKs, Insurance Institute for Highway Safety made a ranking of the most safe car models in 2020. We can see 9 Toyota car models and 4 Ford car models on the ranking. IIHS tests evaluate two aspects of safety: crashworthiness, crash avoidance and mitigation.For each part they conducted several specific tests and evaluable the safeness of a certain vehicle. So if a car scores a higher score under their examination, it means the car is better at protecting occupants in a crash and lessen the severity of a crash.

Sometimes IIHS-HLDI could be biased. They operate a fair test purely on the safeness of a certain car model in the ideal car crash monitor. But they ignore the fact that real car crashes have certain road conditions, drivers conditions, and gender conditions. In The Crash Test Bias: How Male-Focused Testing Puts Female Drivers at Risk, it states that by conducting the car crash testing based on male only is an unsafe choice for women. It is not saying the evaluation by this organization is wrong, it is just impossible to know the actual safety before the accident actually happened. So what this company has done is to make a great prediction of the car safely. But what we can do to prove that prediction is accurate or not is by processing the real data collected from california drivers during each car crash.

In 10 Best Cars with Side Airbags, the author mentions that the 2016 Toyota Camry ups the ante on smaller cars with side airbags. As a midsize sedan, the Camry comes with 10 airbags in total, including seat-mounted side airbags for the front and rear seats, side-curtain airbags, dedicated knee airbags for the driver and front-seat passenger, and dual-stage front airbags. Similar to the design in 2016 Toyota Camry ups, Ford Mondeo were also designed with 10 airbags, aiming to provide complete safety.

In Ford vs. Toyota: Battle of the Brands | US News & World Report, the author states that both Ford and Toyota are popular and reliable car brands which have tons of loyal customers. For safety, Ford and Toyota both have an average U.S. News safety score of 9.3 out of 10, as calculated from the brands’ 2017 models. Ford had a low score of 8.5 for the C-Max Energi and a high score of 9.7 for the Mustang and F-150. Toyota had a low score of 8.5 for the 4Runner and a high score of 10 for the Prius Prime. So it is hard to say which one is safer for drivers when these two brands have a similar ranking score given by the professional institution. At this time there is a need for us to form a data science question and answer the question by collecting and analysing the data.

---

5. **Data**

The raw data come from California Highway Patrol and covers collisions incidents that happened in California from January 1st, 2001 until mid-October, 2020 and was combined into 4 datasets by Alex Gude from Kaggle. We downloaded the data on Kaggle and got a compressed file. After decompressing the files, we found out that the datasets are 4 .sqlite files containing variables of different levels, including “case_id” level, “collision” level, “party” level, and “victims” level. We are using three datasets at “case_id”, “party” level, and “victims” levels to answer our research question.

At “case_id” level:

There are 9.17M incidents of reported car collisions in total. Each incident contains the following information:

- Case Id
- Db year

“case_id” level variables used	
Variable Name	**Description**
case_id	The unique identifier of the collision report (barcode beginning 2002; 19 digit code prior to 2002).
db_year	The year of the original dataset the `case_id` was taken from.

For this dataset, we are going to use the case_id, The unique identifier of the collision report, to identify and keep track of each unique incident. We’ll also use db_year, the year of the original dataset the `case_id` was taken from, to find out when the incidents took place.

At “party” level:

There are 18.2M incidents of reported car collisions in total. Each incident contains the following information:

- Case Id
- Party Number
- Party Type
- At Fault
- Party Sex
- Party Age

- Party Sobriety
- Party Drug Physical
- Direction Of Travel
- Party Safety Equipment 1
- Party Safety Equipment 2
- Special Information 1
- Special Information 2
- Special Information 3
- OAF Violation Code
- OAF Violation Category
- OAF Violation Section
- OAF Violation Suffix
- Other Associated Factor 1
- Other Associated Factor 2
- Party Number Killed
- Party Number Injured
- Movement Preceding Collision
- Vehicle Year
- Vehicle Make
- Statewide Vehicle Type
- CHP Vehicle Type Towing
- CHP Vehicle Type Towed
- Party Race
- Inattention
- Special Info F
- Special Info G

"parties" level variables used	
Variable Name	**Description**
case_id	The unique identifier of the collision report (barcode beginning 2002; 19 digit code prior to 2002).
party_safety_equipment_1	The safety equipment deployed within the vehicle for party #1.
party_safety_equipment_2	The safety equipment deployed within the vehicle for party #2.
party_number_killed	The number of people killed of the involved parties due to collision.
party_number_injured	The number of people injured of the involved parties due to collision.

vehicle_year	The model year of the party's vehicle.
vehicle_make	The full description of the make of the party's vehicle.

For this dataset, we are going to use the case_id (for same purpose), party_safety_equipment_1 and party_safety_equipment_2 (categorical) to find out the accident in which the safety airbag was triggered to go off, party_number_killed and party_number_injured to calculate the number of people injured and died in every incident, vehicle_year and vehicle_make to filter out all the vehicle with year and make consistent with our hypothesis.

At “victims” level:

There are 9.46M incidents of reported car collisions in total. Each incident contains the following information:

- Case Id
- Party Number
- Victim Role
- Victim Sex
- Victim Age
- Victim Degree of Injury
- Victim Seating Position
- Victim Safety Equipment 1
- Victim Safety Equipment 2
- Victim Ejected
- Victim Number

“victims” level variables used	
Variable Name	**Description**
case_id	The unique identifier of the collision report (barcode beginning 2002; 19 digit code prior to 2002).
victim_degree_of_injury	The extent of injury of the victim.
victim_role	The role of the victim.

For this dataset, we are going to use the case_id (for same purpose), victim_degree_of_injury as one of the factors in calculating the extent of the overall severity of each incident, victim_role to filter out only drivers and passenger that are in the car and exclude other victims in roles different from our target, such as pedestrians, because they are presumably unrelated to our hypothesis.

Note:

The original database contains another dataset at collision level, however, we didn't include it because there is not a particular column of variable in it that helps us identify the information we need in order to answer our question; some columns also seem to provide redundant information as variables included in other levels. Therefore we choose not to include any variable from this level for the sake of simplicity.

6. **Analysis Proposal**

Here, you will propose how you would use and analyze data to answer your question of interest but are not required to carry out the analysis to answer your question of interest. You will describe in detail what you would need to do to prepare your dataset for analysis (data wrangling) and what type of analysis you would do to answer your question of interest and explain how you would interpret the results from this analysis. We are looking for the correct conceptual understanding and application of ideas discussed in class, not specific and technical implementations. For example, if you are applying machine learning to some categorical data, it's important to specify whether you will be performing regression or classification. If you are unsure about the details of anything above, ask on Piazza, come to office hours, and/or do further research on your own (Stack Exchange, Google, Wikipedia, etc.).

Specifically, you are required to incorporate *at least four different methods*, exploring ideas from a combination of:

A - Data Collection (web scraping, APIs, etc.)

B - Data Wrangling

C - Statistical Analysis (Inference, A|B testing, etc.)

D - Data Visualization

A. **Data Collection**

We started off our project with data collection. Luckily, we found a set of SQLite databases posted on Kaggle that perfectly organizes the original raw data provided via California Highway Patrol. However, we can't just use the original dataset for our analysis. After all, most of the variables it contains can be irrelevant to our data science question. Exploring the reference webpage linked by the author, we carefully examine every variable in each dataset to find appropriate variables that are helpful to answer our research question. We refine our collected data with a procedure that can be summarized as "exclusion, incorporation, and addition". Exclusion — we exclude variables like "victim_age" and "victim_sex" that are trivial to our hypothesis. Incorporation — we incorporate variables such as "party_safety_equipment_1" and "party_safety_equipment_2" because they can later help us filter out only the scenarios in which airbags are deployed. Addition — we add in some new variables to make clearer distinctions in a originally broad interval. For example, we introduce the variables "severity_score" and "summed_weighted_score" to help us make clearer distinctions between different levels of severity of victim injury: after converting from strings into numbers after data wrangling, this variable can help us quantify the severity of any given incident of car collision.

B. Data wrangling

We did not face much difficulty in terms of collecting or web-scraping data in its rawest form. However, much of our effort was devoted to wrangling the data so that it could be condensed from a 5GB worth of database on Kaggle down to a dataframe that consists just of what we need to evaluate our data science question. As already discussed in both section 5 "Data" and above, we first thoroughly read through the reference webpage that details every database variable with its description to determine which factors are indeed essential to our analysis. As for the actual wrangling process, it can also be summarized as the three-step process of "retrieve, merge, and filter". First, we retrieved only those columns of variables that we deem to be relevant to our analysis using the Pandas "read_sql_query" command from the gigantic database. Then, we merged all data stored in what is now called the Pandas dataframe into one using their unique case identifier. Lastly but also most tediously, we filter and eliminate data to include only those that involve key words such as "Ford", "Toyota", "2018", and such.

C. Statistical Analysis

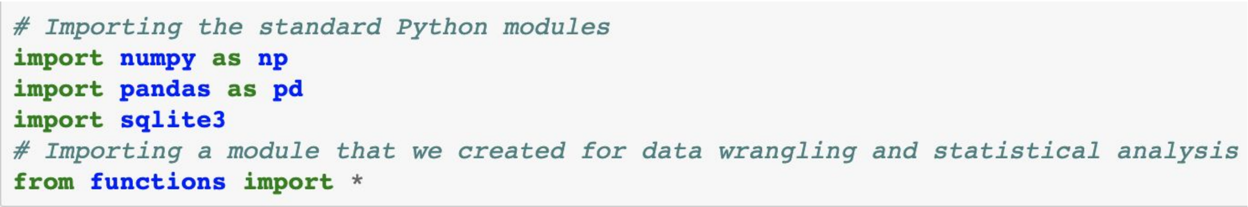
We decided to conduct an one-way independent-samples Student's t-test on our data since we hypothesize our final dataset to consist of two independent series of adjusted severity scores of the collision victims, where the lower mean score goes for Ford and the higher goes for Toyota, indicating that Ford is safer. A one-way Student's t-test would be the most applicable method of statistical analysis since it would tell us whether or not there exists a significant difference between the severity score means of these two unrelated vehicle manufacturers in the direction that we favor, providing us with a result that would either allow us to reject the null and accept the alternative hypothesis or retain the null and conclude that we have failed to detect an effect.

D. Data Visualization

We plan on also creating a pair of box plots for our final dataset since box plots can provide complementary visualization to our t-test results by presenting the distribution of numerical values of a set of data, the skewness, the data quartiles, and the central tendency.

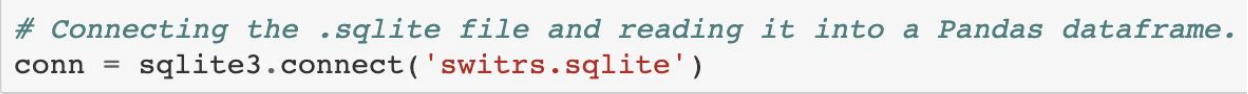
7. Analysis (Presented step by step)

1.



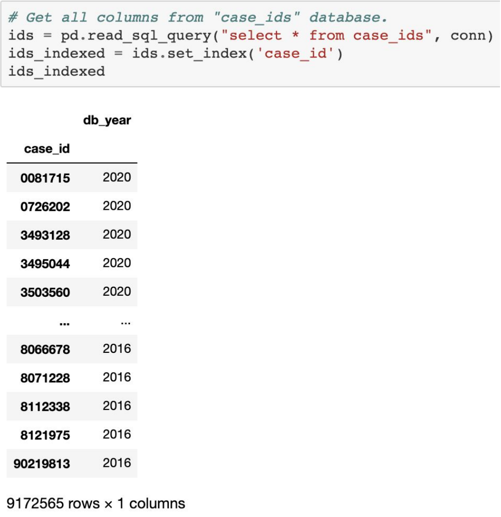
```
# Importing the standard Python modules
import numpy as np
import pandas as pd
import sqlite3
# Importing a module that we created for data wrangling and statistical analysis
from functions import *
```

2.



```
# Connecting the .sqlite file and reading it into a Pandas dataframe.
conn = sqlite3.connect('switrs.sqlite')
```

3.

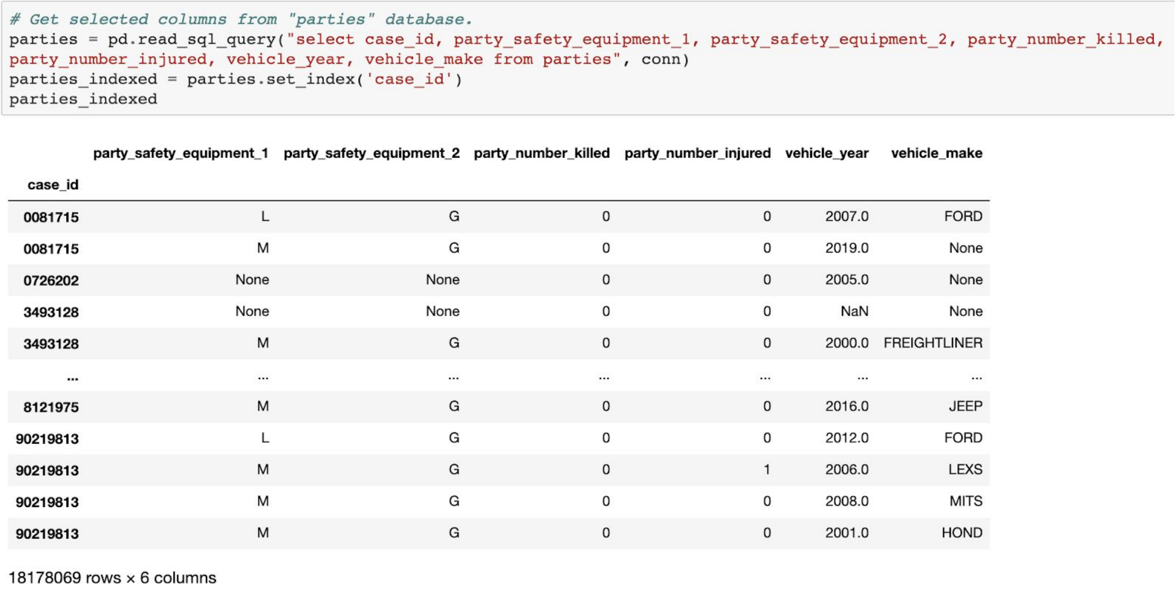


```
# Get all columns from "case_ids" database.
ids = pd.read_sql_query("select * from case_ids", conn)
ids_indexed = ids.set_index('case_id')
ids_indexed
```

	db_year
case_id	
0081715	2020
0726202	2020
3493128	2020
3495044	2020
3503560	2020
...	...
8066678	2016
8071228	2016
8112338	2016
8121975	2016
90219813	2016

9172565 rows × 1 columns

4.

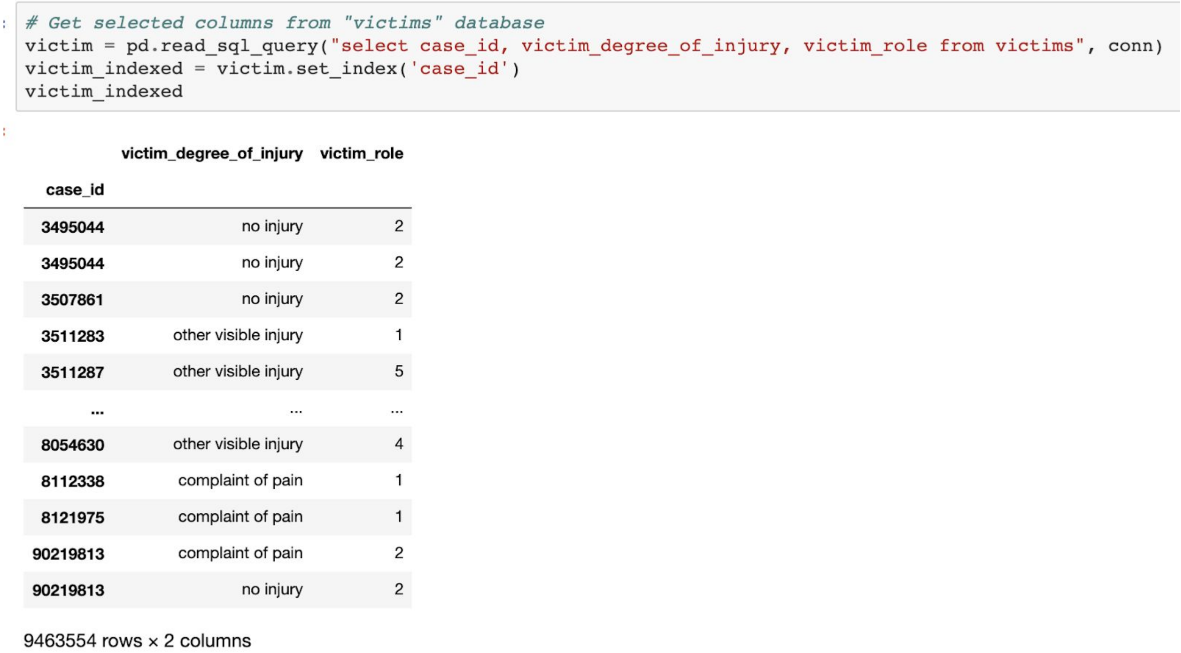


```
# Get selected columns from "parties" database.
parties = pd.read_sql_query("select case_id, party_safety_equipment_1, party_safety_equipment_2, party_number_killed,
party_number_injured, vehicle_year, vehicle_make from parties", conn)
parties_indexed = parties.set_index('case_id')
parties_indexed
```

	party_safety_equipment_1	party_safety_equipment_2	party_number_killed	party_number_injured	vehicle_year	vehicle_make
case_id						
0081715	L	G	0	0	2007.0	FORD
0081715	M	G	0	0	2019.0	None
0726202	None	None	0	0	2005.0	None
3493128	None	None	0	0	NaN	None
3493128	M	G	0	0	2000.0	FREIGHTLINER
...	...	...	...	...	...	...
8121975	M	G	0	0	2016.0	JEEP
90219813	L	G	0	0	2012.0	FORD
90219813	M	G	0	1	2006.0	LEXS
90219813	M	G	0	0	2008.0	MITS
90219813	M	G	0	0	2001.0	HOND

18178069 rows × 6 columns

5.

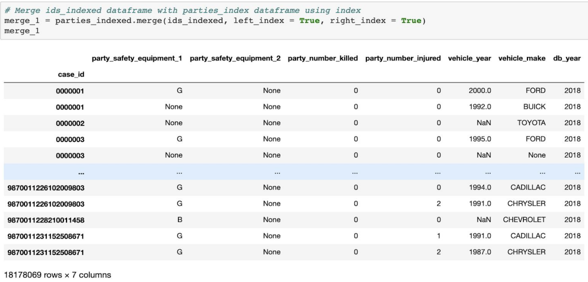


```
# Get selected columns from "victims" database
victim = pd.read_sql_query("select case_id, victim_degree_of_injury, victim_role from victims", conn)
victim_indexed = victim.set_index('case_id')
victim_indexed
```

	victim_degree_of_injury	victim_role
case_id		
3495044	no injury	2
3495044	no injury	2
3507861	no injury	2
3511283	other visible injury	1
3511287	other visible injury	5
...	...	...
8054630	other visible injury	4
8112338	complaint of pain	1
8121975	complaint of pain	1
90219813	complaint of pain	2
90219813	no injury	2

9463554 rows × 2 columns

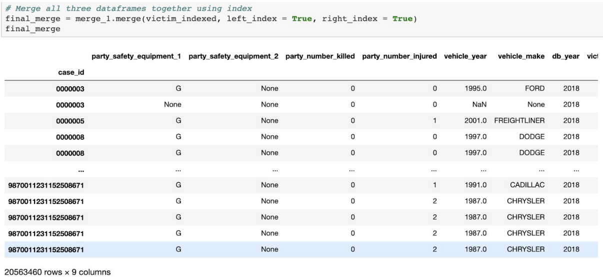
6.



```
# Merge ids_indexed dataframe with parties_index dataframe using index
merge_1 = parties_indexed.merge(ids_indexed, left_index = True, right_index = True)
merge_1
```

	party_safety_equipment_1	party_safety_equipment_2	party_number_killed	party_number_injured	vehicle_year	vehicle_make	db_year
case_id							
0000001	G	None	0	0	2000.0	FORD	2018
0000001	None	None	0	0	1992.0	BUICK	2018
0000002	None	None	0	0	NaN	TOYOTA	2018
0000003	G	None	0	0	1995.0	FORD	2018
0000003	None	None	0	0	NaN	None	2018
...	...	...	...	...	...	...	...
9870011226102009803	G	None	0	0	1994.0	CADILLAC	2018
9870011226102009803	G	None	0	2	1991.0	CHRYSLER	2018
9870011228210011458	B	None	0	0	NaN	CHEVROLET	2018
9870011231152508671	G	None	0	1	1991.0	CADILLAC	2018
9870011231152508671	G	None	0	2	1987.0	CHRYSLER	2018

18178069 rows × 7 columns

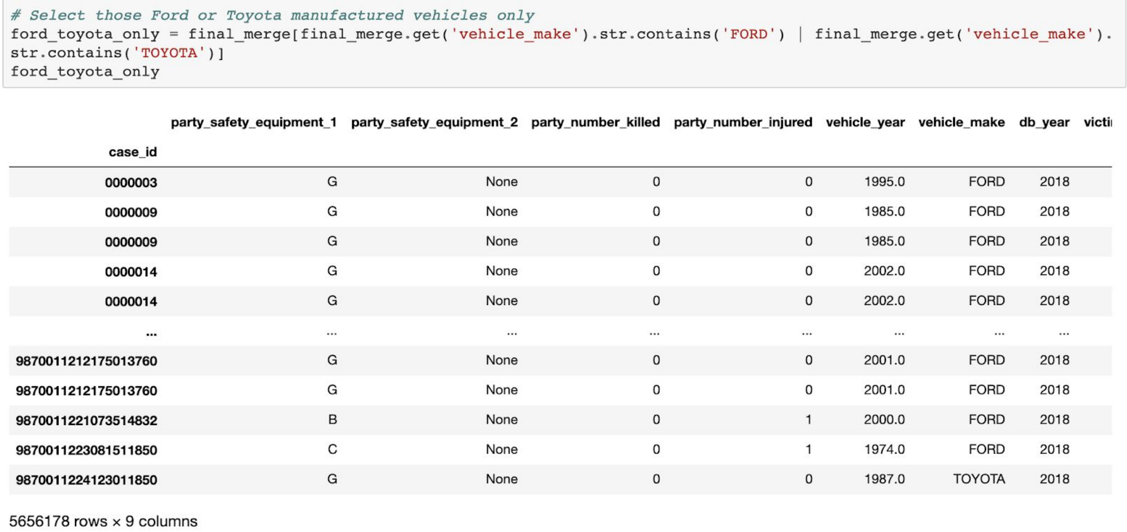


```
# Merge all three dataframes together using index
final_merge = merge_1.merge(victim_indexed, left_index = True, right_index = True)
final_merge
```

	party_safety_equipment_1	party_safety_equipment_2	party_number_killed	party_number_injured	vehicle_year	vehicle_make	db_year	vict
case_id								
0000003	G	None	0	0	1995.0	FORD	2018	
0000003	None	None	0	0	NaN	None	2018	
0000005	G	None	0	1	2001.0	FREIGHTLINER	2018	
0000008	G	None	0	0	1997.0	DODGE	2018	
0000008	G	None	0	0	1997.0	DODGE	2018	
...	...	...	...	...	...	...	...	
9870011231152508671	G	None	0	1	1991.0	CADILLAC	2018	
9870011231152508671	G	None	0	2	1987.0	CHRYSLER	2018	
9870011231152508671	G	None	0	2	1987.0	CHRYSLER	2018	
9870011231152508671	G	None	0	2	1987.0	CHRYSLER	2018	
9870011231152508671	G	None	0	2	1987.0	CHRYSLER	2018	

20563460 rows × 9 columns

7.

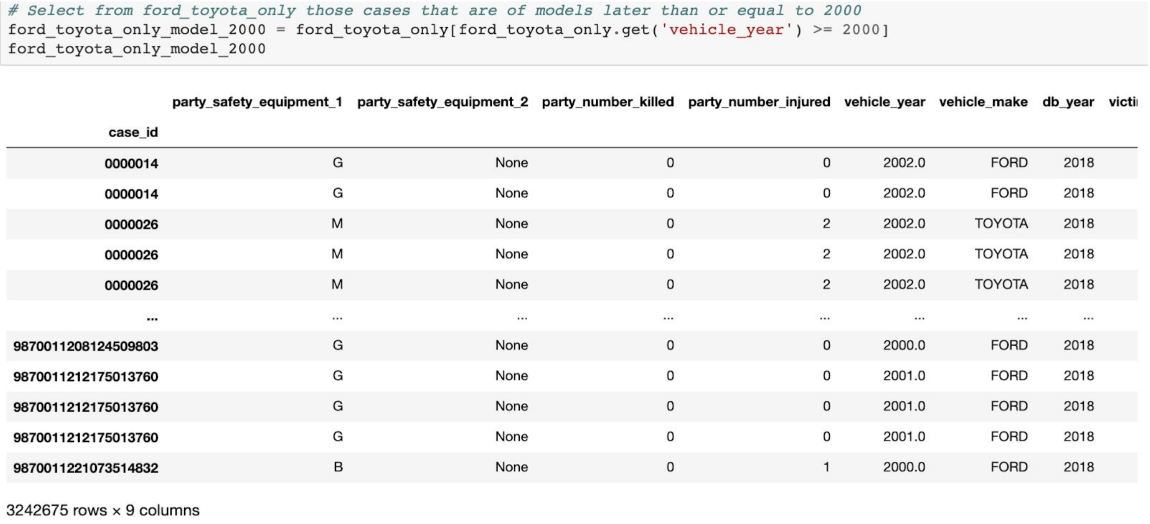


```
# Select those Ford or Toyota manufactured vehicles only
ford_toyota_only = final_merge[final_merge.get('vehicle_make').str.contains('FORD') | final_merge.get('vehicle_make').
str.contains('TOYOTA')]
ford_toyota_only
```

	party_safety_equipment_1	party_safety_equipment_2	party_number_killed	party_number_injured	vehicle_year	vehicle_make	db_year	victi
case_id								
0000003	G	None	0	0	1995.0	FORD	2018	
0000009	G	None	0	0	1985.0	FORD	2018	
0000009	G	None	0	0	1985.0	FORD	2018	
0000014	G	None	0	0	2002.0	FORD	2018	
0000014	G	None	0	0	2002.0	FORD	2018	
...	...	...	...	...	...	...	...	
9870011212175013760	G	None	0	0	2001.0	FORD	2018	
9870011212175013760	G	None	0	0	2001.0	FORD	2018	
9870011221073514832	B	None	0	1	2000.0	FORD	2018	
9870011223081511850	C	None	0	1	1974.0	FORD	2018	
9870011224123011850	G	None	0	0	1987.0	TOYOTA	2018	

5656178 rows × 9 columns

8.

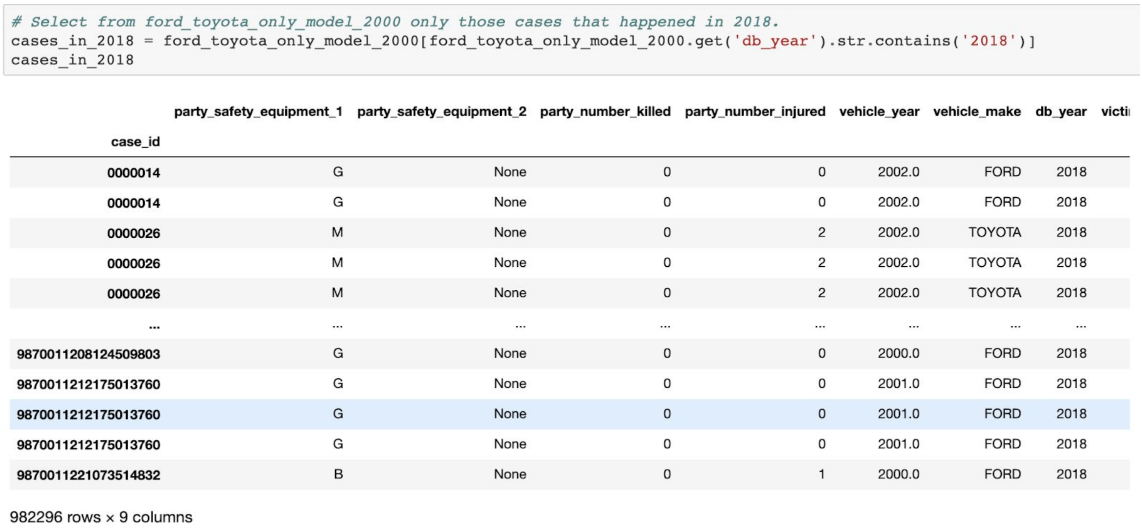


```
# Select from ford_toyota_only those cases that are of models later than or equal to 2000
ford_toyota_only_model_2000 = ford_toyota_only[ford_toyota_only.get('vehicle_year') >= 2000]
ford_toyota_only_model_2000
```

	party_safety_equipment_1	party_safety_equipment_2	party_number_killed	party_number_injured	vehicle_year	vehicle_make	db_year	victi
case_id								
0000014	G	None	0	0	2002.0	FORD	2018	
0000014	G	None	0	0	2002.0	FORD	2018	
0000026	M	None	0	2	2002.0	TOYOTA	2018	
0000026	M	None	0	2	2002.0	TOYOTA	2018	
0000026	M	None	0	2	2002.0	TOYOTA	2018	
...	...	...	...	...	...	...	...	
9870011208124509803	G	None	0	0	2000.0	FORD	2018	
9870011212175013760	G	None	0	0	2001.0	FORD	2018	
9870011212175013760	G	None	0	0	2001.0	FORD	2018	
9870011212175013760	G	None	0	0	2001.0	FORD	2018	
9870011221073514832	B	None	0	1	2000.0	FORD	2018	

3242675 rows × 9 columns

9.

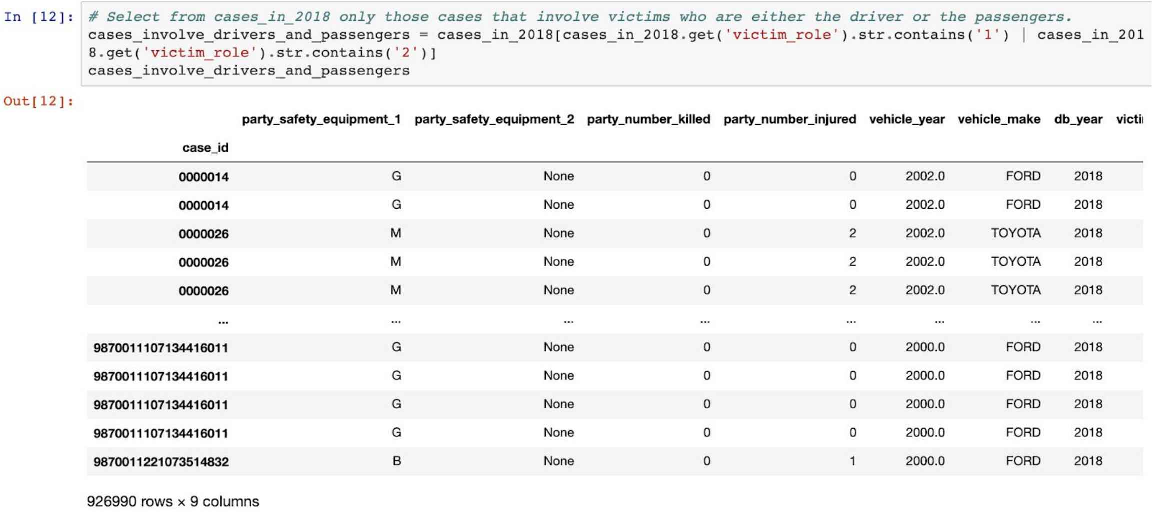


```
# Select from ford_toyota_only_model_2000 only those cases that happened in 2018.
cases_in_2018 = ford_toyota_only_model_2000[ford_toyota_only_model_2000.get('db_year').str.contains('2018')]
cases_in_2018
```

	party_safety_equipment_1	party_safety_equipment_2	party_number_killed	party_number_injured	vehicle_year	vehicle_make	db_year	victi
case_id								
0000014	G	None	0	0	2002.0	FORD	2018	
0000014	G	None	0	0	2002.0	FORD	2018	
0000026	M	None	0	2	2002.0	TOYOTA	2018	
0000026	M	None	0	2	2002.0	TOYOTA	2018	
0000026	M	None	0	2	2002.0	TOYOTA	2018	
...	...	...	...	...	...	...	...	
9870011208124509803	G	None	0	0	2000.0	FORD	2018	
9870011212175013760	G	None	0	0	2001.0	FORD	2018	
9870011212175013760	G	None	0	0	2001.0	FORD	2018	
9870011212175013760	G	None	0	0	2001.0	FORD	2018	
9870011221073514832	B	None	0	1	2000.0	FORD	2018	

982296 rows × 9 columns

10.



```
In [12]: # Select from cases_in_2018 only those cases that involve victims who are either the driver or the passengers.
         cases_involve_drivers_and_passengers = cases_in_2018[cases_in_2018.get('victim_role').str.contains('1') | cases_in_201
         8.get('victim_role').str.contains('2')]
         cases_involve_drivers_and_passengers
```

Out[12]:

	party_safety_equipment_1	party_safety_equipment_2	party_number_killed	party_number_injured	vehicle_year	vehicle_make	db_year	victi
case_id								
0000014	G	None	0	0	2002.0	FORD	2018	
0000014	G	None	0	0	2002.0	FORD	2018	
0000026	M	None	0	2	2002.0	TOYOTA	2018	
0000026	M	None	0	2	2002.0	TOYOTA	2018	
0000026	M	None	0	2	2002.0	TOYOTA	2018	
...	...	...	...	...	...	...	...	
9870011107134416011	G	None	0	0	2000.0	FORD	2018	
9870011107134416011	G	None	0	0	2000.0	FORD	2018	
9870011107134416011	G	None	0	0	2000.0	FORD	2018	
9870011107134416011	G	None	0	0	2000.0	FORD	2018	
9870011221073514832	B	None	0	1	2000.0	FORD	2018	

926990 rows × 9 columns

11.

```
# Select from cases_involve_drivers_and_passengers only those cases where airbags are deployed for either party.
cases_air_bag = cases_involve_drivers_and_passengers[cases_involve_drivers_and_passengers.get('party_safety_equipment_
1').str.contains('L') | cases_involve_drivers_and_passengers.get('party_safety_equipment_2').str.contains('L')]
cases_air_bag
```

	party_safety_equipment_1	party_safety_equipment_2	party_number_killed	party_number_injured	vehicle_year	vehicle_make	db_year	victi
case_id								
0000089	G	L	0	1	2001.0	TOYOTA	2018	
0000089	G	L	0	1	2001.0	TOYOTA	2018	
0000177	L	None	0	2	2002.0	TOYOTA	2018	
0000177	L	None	0	2	2002.0	TOYOTA	2018	
0000200	G	L	0	0	2000.0	FORD	2018	
...	...	...	...	...	...	...	...	
9865010619181014729	L	None	0	1	2000.0	FORD	2018	
9865010629134015271	L	None	0	1	2001.0	TOYOTA	2018	
9865010914113512988	L	None	0	1	2001.0	TOYOTA	2018	
9865011018124509128	L	None	0	0	2001.0	FORD	2018	
9870010418100009329	L	None	0	1	2000.0	FORD	2018	

121345 rows × 9 columns

12.

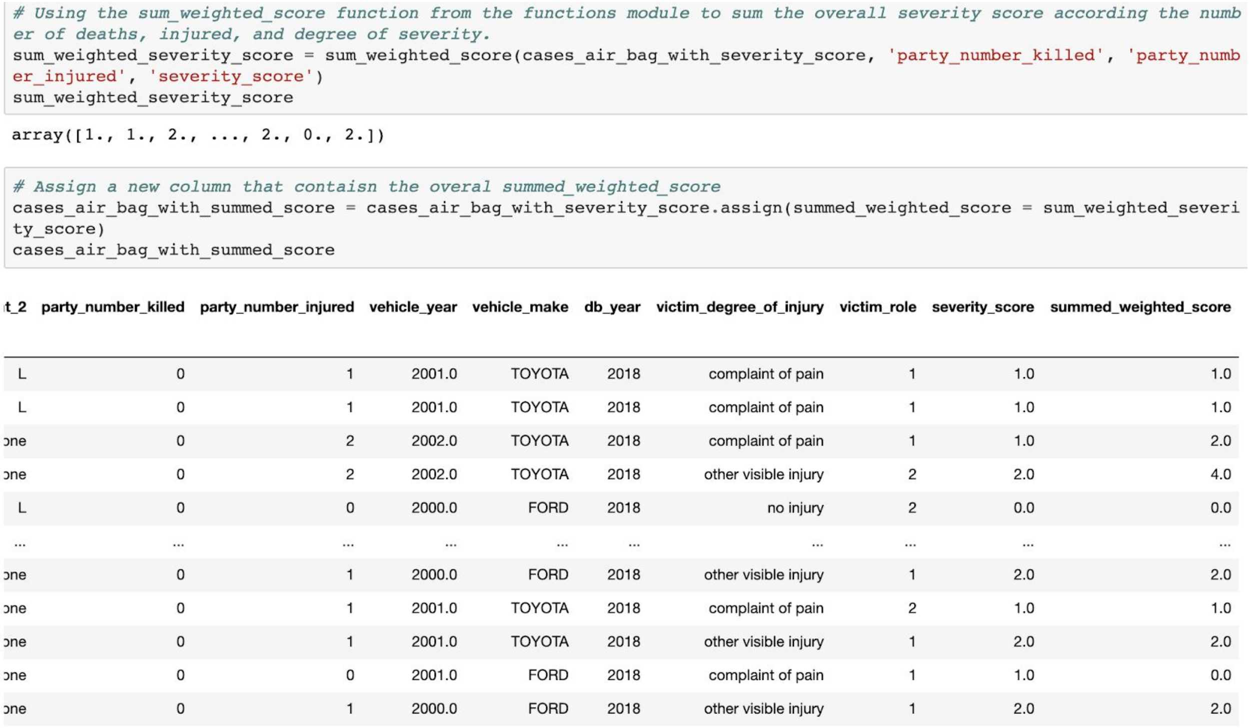
```
# Use the assign_severity_level_score function from the functions module to translate victim's degree of injury to a n
umerical score.
degree_of_severity_score = assign_severity_level_score(cases_air_bag, 'victim_degree_of_injury')
degree_of_severity_score
```

```
array([1., 1., 1., ..., 2., 1., 2.])
```

```
# Add a column that contains the corresponding severity scores to each degree of the vitim's injury.
cases_air_bag_with_severity_score = cases_air_bag.assign(severity_score = degree_of_severity_score)
cases_air_bag_with_severity_score
```

t_1	party_safety_equipment_2	party_number_killed	party_number_injured	vehicle_year	vehicle_make	db_year	victim_degree_of_injury	victim_role	severity_score
G	L	0	1	2001.0	TOYOTA	2018	complaint of pain	1	1.0
G	L	0	1	2001.0	TOYOTA	2018	complaint of pain	1	1.0
L	None	0	2	2002.0	TOYOTA	2018	complaint of pain	1	1.0
L	None	0	2	2002.0	TOYOTA	2018	other visible injury	2	2.0
G	L	0	0	2000.0	FORD	2018	no injury	2	0.0
...	...	...	...	...	...	...	...	...	...
L	None	0	1	2000.0	FORD	2018	other visible injury	1	2.0
L	None	0	1	2001.0	TOYOTA	2018	complaint of pain	2	1.0
L	None	0	1	2001.0	TOYOTA	2018	other visible injury	1	2.0
L	None	0	0	2001.0	FORD	2018	complaint of pain	1	1.0
L	None	0	1	2000.0	FORD	2018	other visible injury	1	2.0

13.



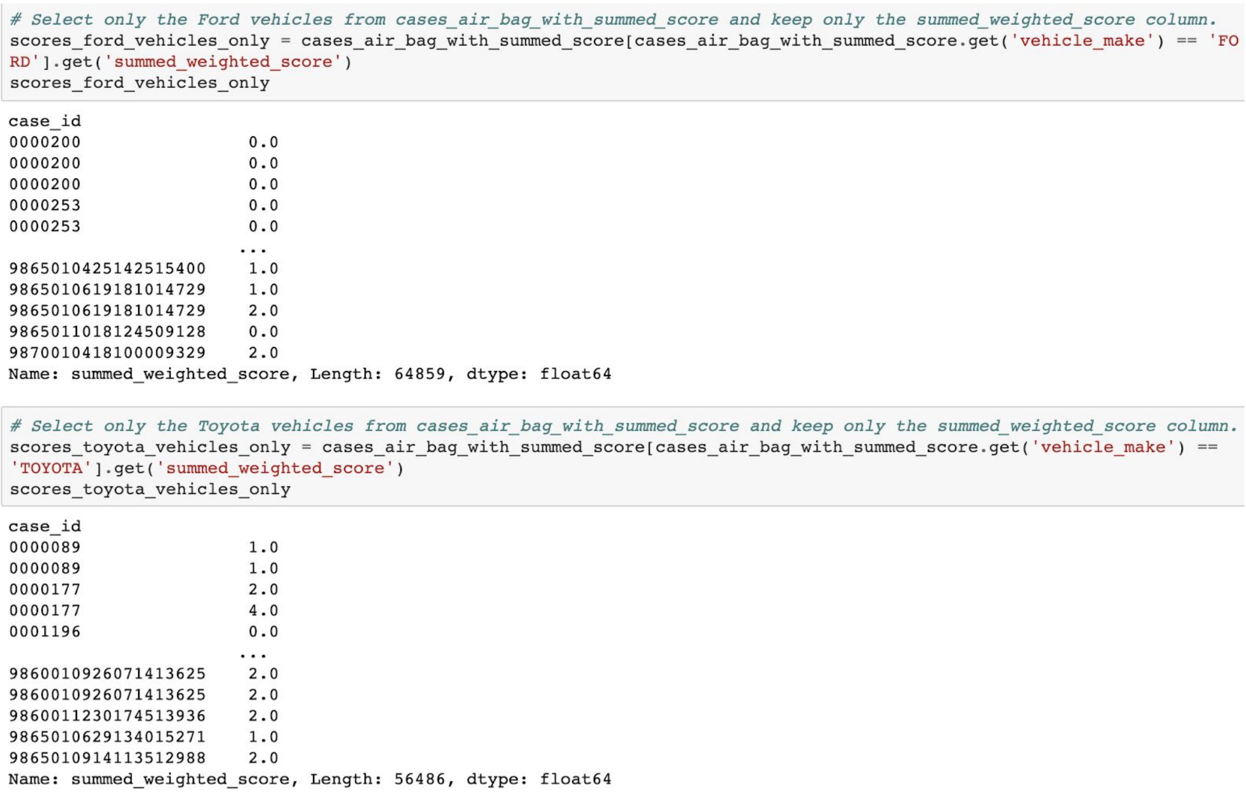
```
# Using the sum_weighted_score function from the functions module to sum the overall severity score according the number of deaths, injured, and degree of severity.
sum_weighted_severity_score = sum_weighted_score(cases_air_bag_with_severity_score, 'party_number_killed', 'party_number_injured', 'severity_score')
sum_weighted_severity_score
```

```
array([1., 1., 2., ..., 2., 0., 2.])
```

```
# Assign a new column that contaisn the overal summed_weighted_score
cases_air_bag_with_summed_score = cases_air_bag_with_severity_score.assign(summed_weighted_score = sum_weighted_severity_score)
cases_air_bag_with_summed_score
```

t_2	party_number_killed	party_number_injured	vehicle_year	vehicle_make	db_year	victim_degree_of_injury	victim_role	severity_score	summed_weighted_score
L	0	1	2001.0	TOYOTA	2018	complaint of pain	1	1.0	1.0
L	0	1	2001.0	TOYOTA	2018	complaint of pain	1	1.0	1.0
one	0	2	2002.0	TOYOTA	2018	complaint of pain	1	1.0	2.0
one	0	2	2002.0	TOYOTA	2018	other visible injury	2	2.0	4.0
L	0	0	2000.0	FORD	2018	no injury	2	0.0	0.0
...	...	...	...	...	...	...	...	...	...
one	0	1	2000.0	FORD	2018	other visible injury	1	2.0	2.0
one	0	1	2001.0	TOYOTA	2018	complaint of pain	2	1.0	1.0
one	0	1	2001.0	TOYOTA	2018	other visible injury	1	2.0	2.0
one	0	0	2001.0	FORD	2018	complaint of pain	1	1.0	0.0
one	0	1	2000.0	FORD	2018	other visible injury	1	2.0	2.0

14.



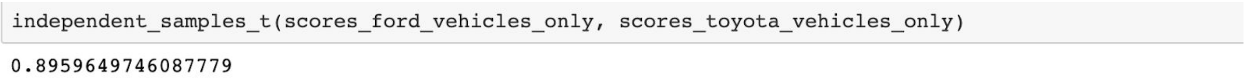
```
# Select only the Ford vehicles from cases_air_bag_with_summed_score and keep only the summed_weighted_score column.
scores_ford_vehicles_only = cases_air_bag_with_summed_score[cases_air_bag_with_summed_score.get('vehicle_make') == 'FORD'].get('summed_weighted_score')
scores_ford_vehicles_only
```

```
case_id
0000200                0.0
0000200                0.0
0000200                0.0
0000253                0.0
0000253                0.0
                      ...
9865010425142515400    1.0
9865010619181014729    1.0
9865010619181014729    2.0
9865011018124509128    0.0
9870010418100009329    2.0
Name: summed_weighted_score, Length: 64859, dtype: float64
```

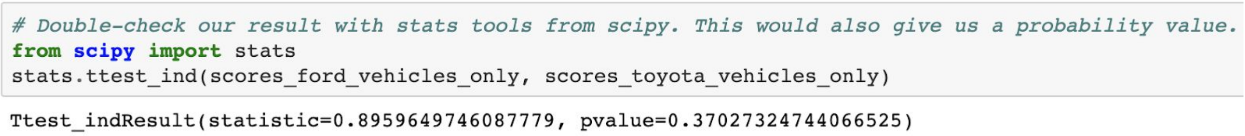
```
# Select only the Toyota vehicles from cases_air_bag_with_summed_score and keep only the summed_weighted_score column.
scores_toyota_vehicles_only = cases_air_bag_with_summed_score[cases_air_bag_with_summed_score.get('vehicle_make') == 'TOYOTA'].get('summed_weighted_score')
scores_toyota_vehicles_only
```

```
case_id
0000089                1.0
0000089                1.0
0000177                2.0
0000177                4.0
0001196                0.0
                      ...
9860010926071413625    2.0
9860010926071413625    2.0
9860011230174513936    2.0
9865010629134015271    1.0
9865010914113512988    2.0
Name: summed_weighted_score, Length: 56486, dtype: float64
```

15.



```
independent_samples_t(scores_ford_vehicles_only, scores_toyota_vehicles_only)

0.8959649746087779
```



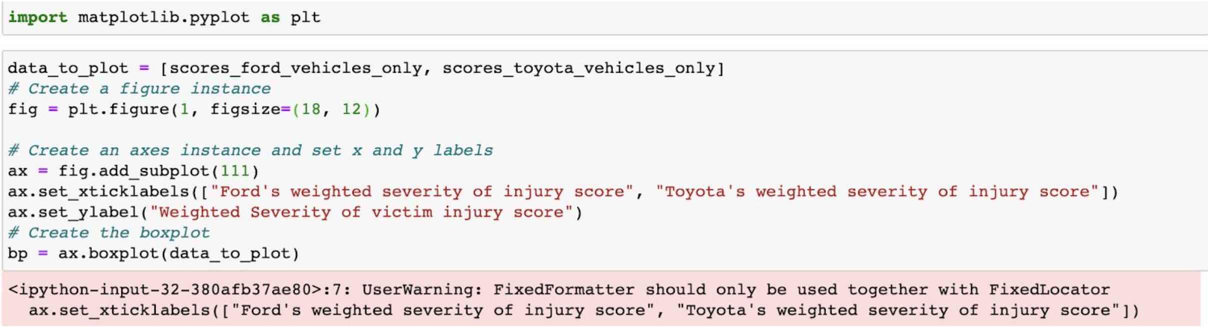
```
# Double-check our result with stats tools from scipy. This would also give us a probability value.
from scipy import stats
stats.ttest_ind(scores_ford_vehicles_only, scores_toyota_vehicles_only)

Ttest_indResult(statistic=0.8959649746087779, pvalue=0.37027324744066525)
```

Here is the link to our GitHub repository that contains all the coding materials used in this project (except the database): https://github.com/Lord-of-Bugs/cogs_9_final_project_fa20
Here is the link to the Google Drive folder that contains the data and the written project: https://drive.google.com/drive/u/1/folders/180S7KXDs8U36_QaycC8zBqSdBe6uN4Jm

---

8. **Visualization**

Code:

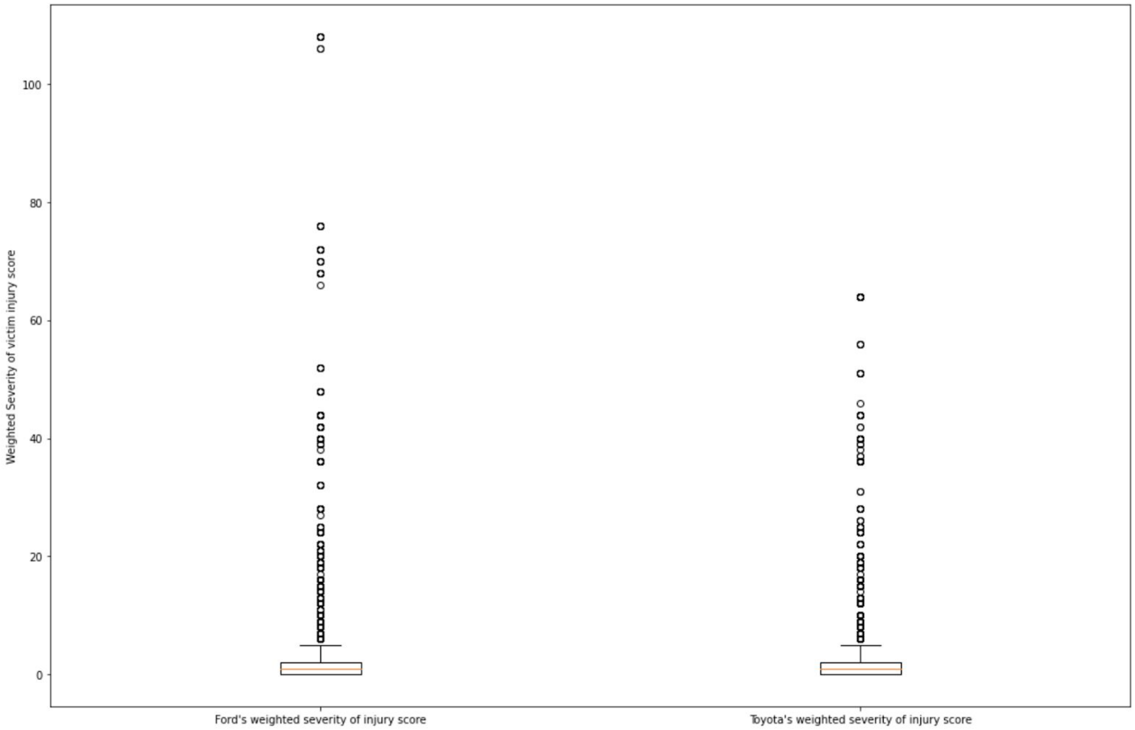


```
import matplotlib.pyplot as plt
```

```
data_to_plot = [scores_ford_vehicles_only, scores_toyota_vehicles_only]
# Create a figure instance
fig = plt.figure(1, figsize=(18, 12))

# Create an axes instance and set x and y labels
ax = fig.add_subplot(111)
ax.set_xticklabels(["Ford's weighted severity of injury score", "Toyota's weighted severity of injury score"])
ax.set_ylabel("Weighted Severity of victim injury score")
# Create the boxplot
bp = ax.boxplot(data_to_plot)
```

```
<ipython-input-32-380afb37ae80>:7: UserWarning: FixedFormatter should only be used together with FixedLocator
  ax.set_xticklabels(["Ford's weighted severity of injury score", "Toyota's weighted severity of injury score"])
```

Plot:



---

9. **Discussion on the results**

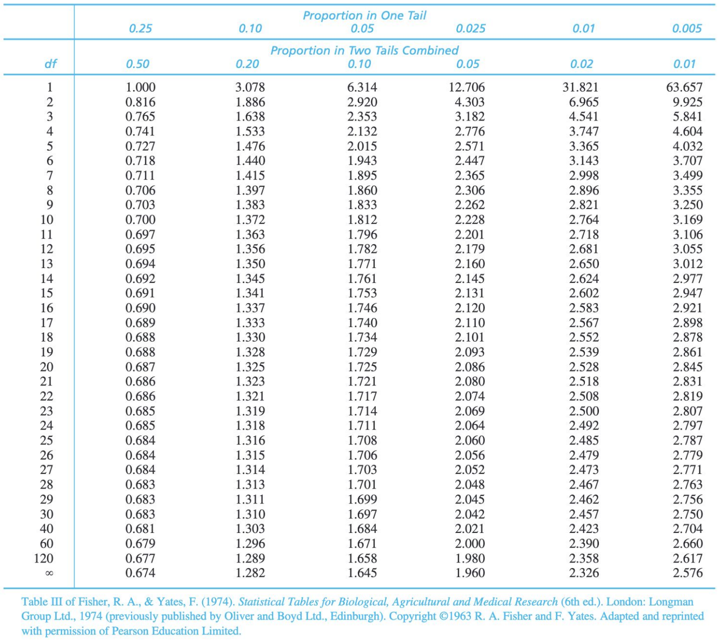
How would you interpret the results of your proposed analysis? What are the limitations, pitfalls, and potential confounds of your methods, or biases in your data sources? (e.g., how does the selection of the sources of your crowds affect your outcomes?) How would you set out to address them? In addition, outline how you would address any societal and/or ethical implications of your proposed project discussed in your Ethical Considerations section. (10 pts)

We obtained a t-statistic of t (degree of freedom = 121343, approaching infinity) = 0.896, with p = 0.370. When interpreting this independent-samples t-test result, we referred to the t-test table (pasted below). We would like to follow the convention by setting alpha value to 0.05 for our hypothesis testing. At alpha = 0.05, the one-tail critical t value has to be at least 1.645 for the result to be significant. Our t of 0.896 is within that critical limit, therefore we fail to reject the statistical null hypothesis that there exists no significant difference between the victim injury severity scores for Ford and Toyota vehicles.

However, this is not a final, definitive conclusion to the Ford vs. Toyota vehicle collision safety research. There are, in fact, many limitations and inconsistencies in our data wrangling procedure and the data itself that could have increased the noises in our data, resulting in our potential “miss” of real effect. For example, due to the volume of our original dataset, we can only inspect so many rows of data entries before letting the rest get hidden under the “...”. This is not ideal since when we continue along the data wrangling procedures, we are essentially extrapolating our understanding and expectations about how the first twenty or so rows of data should perform onto the millions of rows that are hidden underneath. The data structure and the information of those that we can’t inspect directly could have been entirely different. Secondly, after we inspected the “case_id” dataset and the collision dataset, particularly the “db_year” column of the former and the “collision_date” column of the latter, we have already found quite a few mismatches among even just the top 10 rows between the two. For instance, the collision that’s identified under the same “case_id” is shown to have a “db_year” of 2020 while the “collision_date” is starting with year 2008. For our purpose of data analysis, our team chose to calibrate the year of the collision with “db_year”. Some other team can totally disagree and alternatively split the year information off of “collision_date”. It is inevitable that the results both teams end up with would be dissimilar, and neither is wrong since there is no documentation on which time variable is valid and which is not. Moreover, we found it extremely strange when we were unable to query any information from setting the merged data frame to include only those cases that have “db_year” equal to 2020--in actuality, we have seen “2020” appear under “db_year” just by inspection. Limitations like this go on and on.

In summary, we accept our decision of retaining the null hypothesis that Ford-manufactured vehicles that are of model year later than or equal to year 2000 protect their drivers and passengers just as well from injuries as those of Toyota-manufactured counterparts in collision accidents under the deployment of airbag situations in year 2018, while also acknowledging that there have been and can potentially be many more limitations and noises to our data and data wrangling procedures. If we were to revise this project one day, we would devote more time to communicating with the California Highway Patrol to reconcile the contradiction between “db_year” and “collision_date” as mentioned, which we believe would be the most critical factor in helping reduce the noises.

T-table:



	Proportion in One Tail					
	0.25	0.10	0.05	0.025	0.01	0.005
	Proportion in Two Tails Combined					
df	0.50	0.20	0.10	0.05	0.02	0.01
1	1.000	3.078	6.314	12.706	31.821	63.657
2	0.816	1.886	2.920	4.303	6.965	9.925
3	0.765	1.638	2.353	3.182	4.541	5.841
4	0.741	1.533	2.132	2.776	3.747	4.604
5	0.727	1.476	2.015	2.571	3.365	4.032
6	0.718	1.440	1.943	2.447	3.143	3.707
7	0.711	1.415	1.895	2.365	2.998	3.499
8	0.706	1.397	1.860	2.306	2.896	3.355
9	0.703	1.383	1.833	2.262	2.821	3.250
10	0.700	1.372	1.812	2.228	2.764	3.169
11	0.697	1.363	1.796	2.201	2.718	3.106
12	0.695	1.356	1.782	2.179	2.681	3.055
13	0.694	1.350	1.771	2.160	2.650	3.012
14	0.692	1.345	1.761	2.145	2.624	2.977
15	0.691	1.341	1.753	2.131	2.602	2.947
16	0.690	1.337	1.746	2.120	2.583	2.921
17	0.689	1.333	1.740	2.110	2.567	2.898
18	0.688	1.330	1.734	2.101	2.552	2.878
19	0.688	1.328	1.729	2.093	2.539	2.861
20	0.687	1.325	1.725	2.086	2.528	2.845
21	0.686	1.323	1.721	2.080	2.518	2.831
22	0.686	1.321	1.717	2.074	2.508	2.819
23	0.685	1.319	1.714	2.069	2.500	2.807
24	0.685	1.318	1.711	2.064	2.492	2.797
25	0.684	1.316	1.708	2.060	2.485	2.787
26	0.684	1.315	1.706	2.056	2.479	2.779
27	0.684	1.314	1.703	2.052	2.473	2.771
28	0.683	1.313	1.701	2.048	2.467	2.763
29	0.683	1.311	1.699	2.045	2.462	2.756
30	0.683	1.310	1.697	2.042	2.457	2.750
40	0.681	1.303	1.684	2.021	2.423	2.704
60	0.679	1.296	1.671	2.000	2.390	2.660
120	0.677	1.289	1.658	1.980	2.358	2.617
∞	0.674	1.282	1.645	1.960	2.326	2.576

Table III of Fisher, R. A., & Yates, F. (1974). *Statistical Tables for Biological, Agricultural and Medical Research* (6th ed.). London: Longman Group Ltd., 1974 (previously published by Oliver and Boyd Ltd., Edinburgh). Copyright ©1963 R. A. Fisher and F. Yates. Adapted and reprinted with permission of Pearson Education Limited.

---

10. **Team participation**

Name	**Task**
Muchan Li	Carried out the wrangling, analysis, and visualization. Also wrote the discussion on results. Wrote the justification section and the last three components of the analysis proposal section.
Yunfan Long	Collected data and refined the data collected. Go through variables provided in all datasets manually and suggest appropriate variables for our analysis. Helped Muchan to quantify the severity level to perform statistical analysis by strategically assigning numeric value to strings that represent different degrees of severity. Wrote the “Data” part and “Data Collection” section in the “Analysis Proposal” part.
Kening Li	Did research about background info.
Dingrui Wang	Together with Yunfan collected data and refined the data collected. Go through

	variables provided in all datasets manually and suggest appropriate variables for our analysis.
Guanxing Chen	Did research about background info, revised the Ethical Considerations.

11. Ethical Considerations

Ethical consideration in reference to the Deon's Checklist:

1. Data Collection

   a. Informed consent: We are acquiring our raw data from publicly available, open-access web source Kaggle. We did not use API this time , but if we need it in the future we will abide by any rules of regulation and consent when our source of data is privately owned by funded institutions and labs and notify the source of our plans and uses of the data when consent is granted.

   b. Collection bias: We aim to "abandon" all our prior knowledge and assumption about the safeness of car brands like Toyota and Ford, which we might have sort of personal perceptions about, while collecting data in order to maximize the "blindness" in our collection.

   c. Limit personally identifiable information: We presume it would not be helpful for us to include personally identifiable information in our datasets since the acquisition of our raw data will depend on sources and information provided by California Highway Patrol and covers collisions from January 1st, 2001 until mid-October, 2020. If we do come across PII along our data wrangling process in the next step of research, we will make sure to exclude those to maintain confidentiality and privacy.

2. Data storage

   a. Data security in storage: The data that we plan to collect will be primarily from Kaggle and from California Highway Patrol, which is the original source. Therefore, it can be considered state property despite being made publicly-accessible. Thus, we would like to be responsible for both the proprietary and the analytical value of our data and we would do so by cloud-storing our raw data and notebook on github and google drive that only the team members have access to initially. We may make the data and our analysis public in the future if we are made certain that our model and analysis wouldn't be used or interpreted in any way that is harmful.

b. Right to be forgotten: Our source has not decided to revoke our access to the data along the process making process until now. If they decide to do so in the future, we will abide by so.

c. Data retention: We have established a separate repository on github that exclusively stores the chunks of the raw data that we've decided to leave out of our analysis and a documentation that explains the reason we did so.

3. Analysis
    a. Auditability: We had a clear documentation to record each step we did in our analysis process to make sure the process is reproducible. This includes the details of how we wrangling our raw datas and what problems have occurred. We have saved changes as well as the entire Jupyter notebook that we performed our wrangling and analysis with on Github, so it is possible for us and others to go back to examine a specific step of our data wrangling process for reproducible use.

    b. Honest representation: We have carefully examined whether our visualizations, summary statistics, and report are consistent with the underlying data. Besides trying to avoid the collection bias from the beginning, we were highly aware of the way to present our data to minimize the potential risk of misconception. For example, we have provided extra information in the footnote, in case the reader needs further clarification.

    c. Privacy in analysis: We have excluded anything that pertains to PII.

4. Modeling
    a. Proxy discrimination: Based on the best of our knowledge and information acquired from background research, the dataset we obtained on Kaggle and Github are fairly comprehensive and unbiased in terms of unequivocally recording millions of collision accidents from year to year. We will continue to ensure that we do not sample or manipulate relevant components that would discriminate in any way.

    b. Metric selection: There strictly isn't a continuous numerical metric for severity that we can rely on by convention. Our team instead coded a convenience metric that simply matches the hierarchical ranking of injury severity, as well as establishing an exponential factor for cases that are most severe to highlight the effect. We have held our metric to be consistent throughout the analysis and we believe it to be reliable.

    c. Communication bias: We have highlighted any possible shortcomings in our data manipulation and visualization. We have not disguised any biased information to the readers of this report.

5. Deployment
    a. Potential redress: We were very careful in the process of sorting out our data, once the user thinks our data contains any kind of harmful or incorrect information, we will enable our Redress Discusses mechanism.

    b. Concept drift: Since our data comes from public web sources, there are situations in which data changes over time. In such a case, we have regularly checked the update status and accuracy of the information in the data source, and add the updated data to our project every time.

---

12. **Work cited**

link : https://www.kaggle.com/alexgude/california-traffic-collision-data-from-switrs and https://tims.berkeley.edu/help/SWITRS.php

https://cars.usnews.com/cars-trucks/ford-vs-toyota-battle-of-the-brands

https://www.consumerreports.org/car-safety/crash-test-bias-how-male-focused-testing-puts-female-drivers-at-risk/

https://cars.usnews.com/cars-trucks/ford-vs-toyota-battle-of-the-brands

https://www.autobytel.com/car-buying-guides/features/10-best-cars-with-side-airbags-131283/