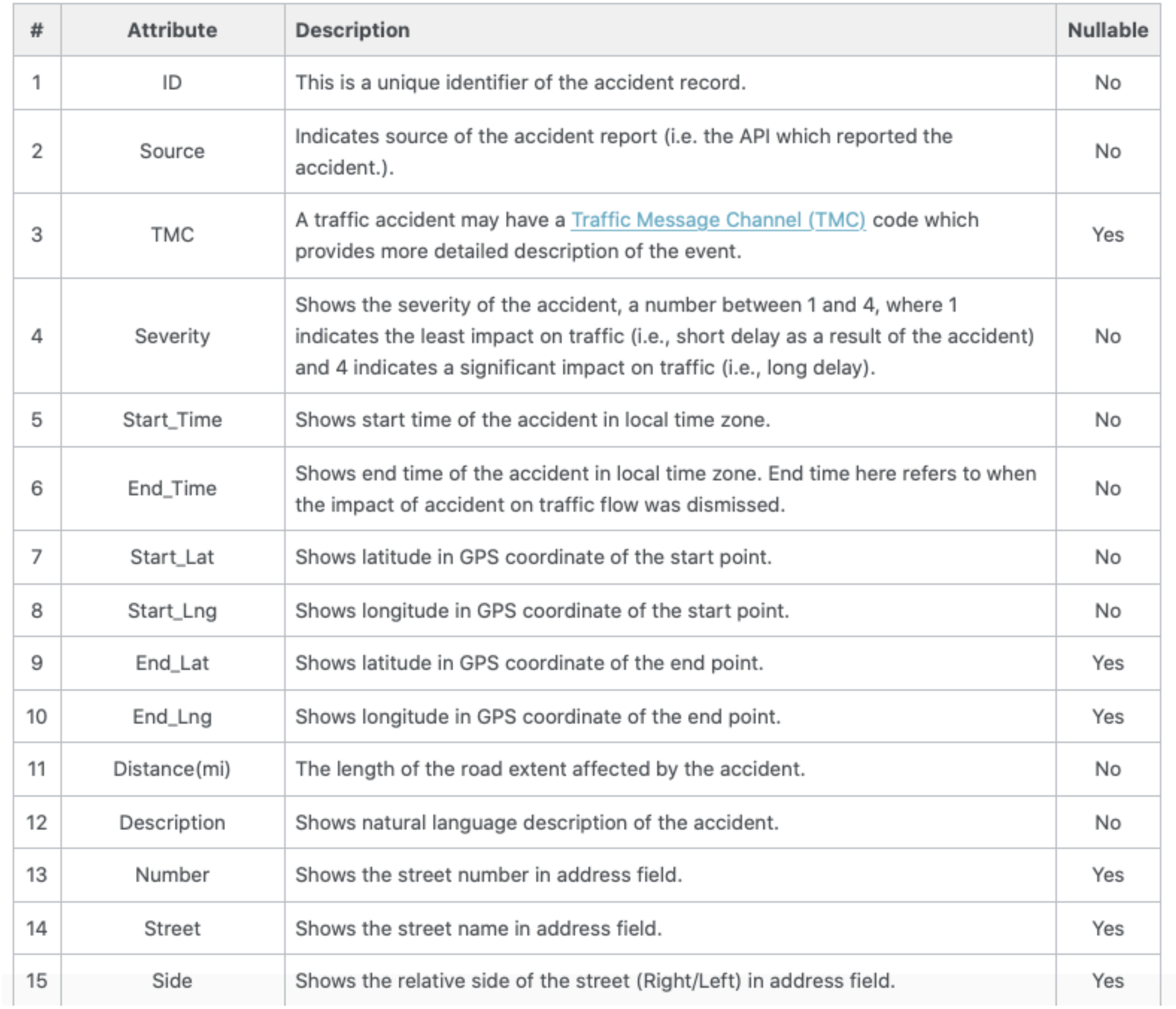
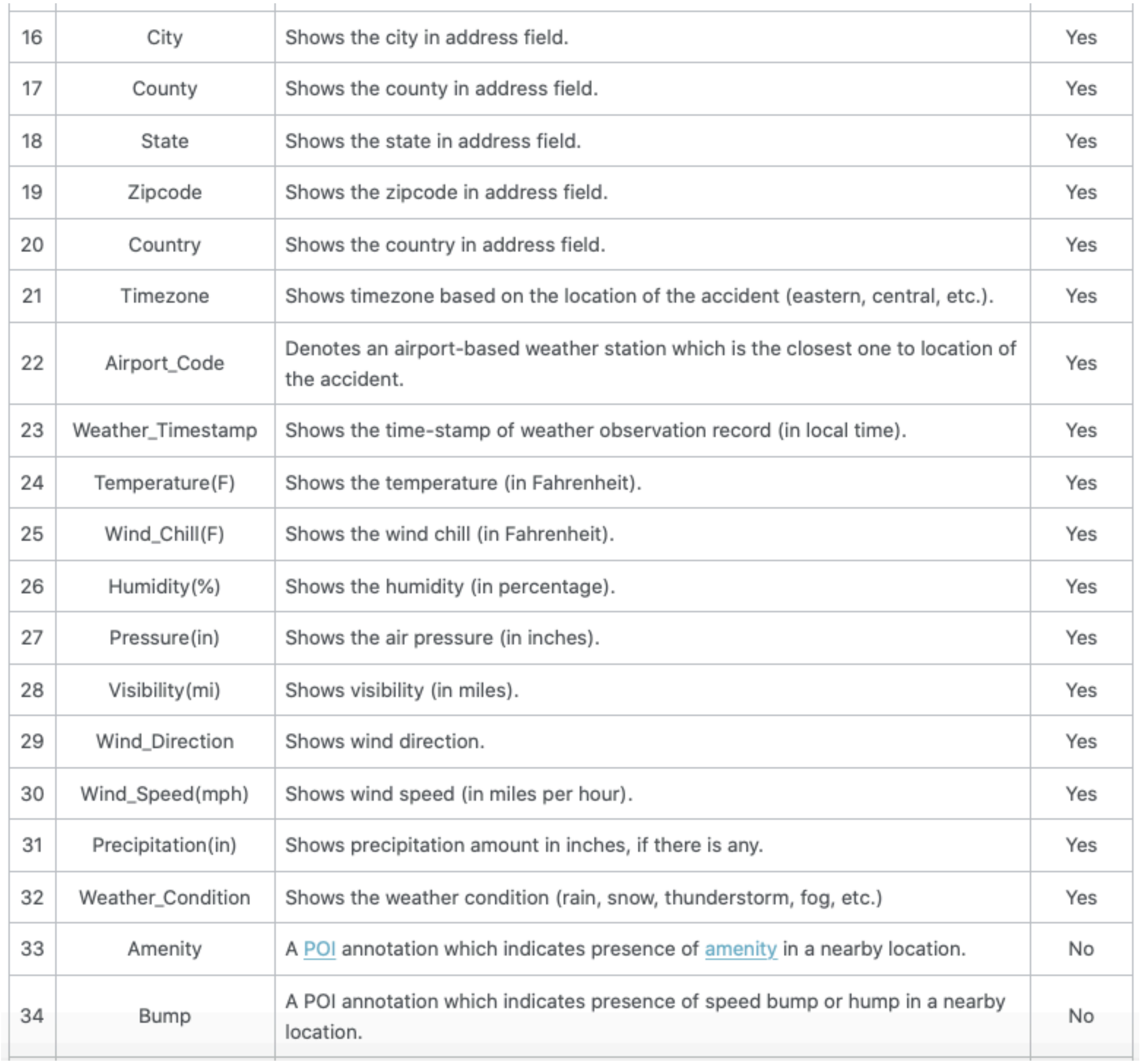
Prediction of Severity in Car Accident

Juan Carlos Zambrano

08 - 10 - 20

Data Set





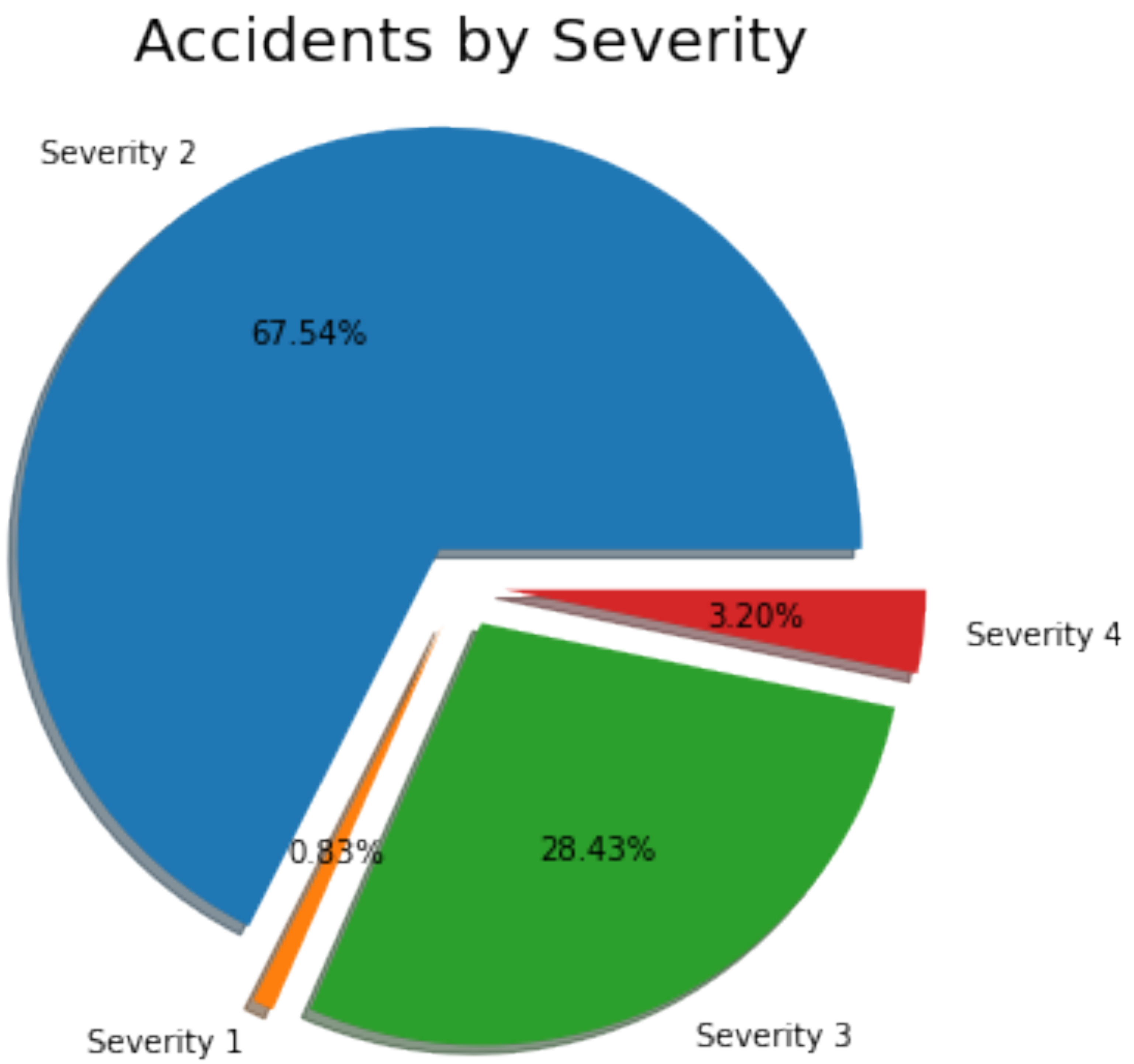
#	Attribute	Description	Nullable
1	ID	This is a unique identifier of the accident record.	No
2	Source	Indicates source of the accident report (i.e. the API which reported the accident.).	No
3	TMC	A traffic accident may have a Traffic Message Channel (TMC) code which provides more detailed description of the event.	Yes
4	Severity	Shows the severity of the accident, a number between 1 and 4, where 1 indicates the least impact on traffic (i.e., short delay as a result of the accident) and 4 indicates a significant impact on traffic (i.e., long delay).	No
5	Start_Time	Shows start time of the accident in local time zone.	No
6	End_Time	Shows end time of the accident in local time zone. End time here refers to when the impact of accident on traffic flow was dismissed.	No
7	Start_Lat	Shows latitude in GPS coordinate of the start point.	No
8	Start_Lng	Shows longitude in GPS coordinate of the start point.	No
9	End_Lat	Shows latitude in GPS coordinate of the end point.	Yes
10	End_Lng	Shows longitude in GPS coordinate of the end point.	Yes
11	Distance(mi)	The length of the road extent affected by the accident.	No
12	Description	Shows natural language description of the accident.	No
13	Number	Shows the street number in address field.	Yes
14	Street	Shows the street name in address field.	Yes
15	Side	Shows the relative side of the street (Right/Left) in address field.	Yes
16	City	Shows the city in address field.	Yes
17	County	Shows the county in address field.	Yes
18	State	Shows the state in address field.	Yes
19	Zipcode	Shows the zipcode in address field.	Yes
20	Country	Shows the country in address field.	Yes
21	Timezone	Shows timezone based on the location of the accident (eastern, central, etc.).	Yes
22	Airport_Code	Denotes an airport-based weather station which is the closest one to location of the accident.	Yes
23	Weather_Timestamp	Shows the time-stamp of weather observation record (in local time).	Yes
24	Temperature(F)	Shows the temperature (in Fahrenheit).	Yes
25	Wind_Chill(F)	Shows the wind chill (in Fahrenheit).	Yes
26	Humidity(%)	Shows the humidity (in percentage).	Yes
27	Pressure(in)	Shows the air pressure (in inches).	Yes
28	Visibility(mi)	Shows visibility (in miles).	Yes
29	Wind_Direction	Shows wind direction.	Yes
30	Wind_Speed(mph)	Shows wind speed (in miles per hour).	Yes
31	Precipitation(in)	Shows precipitation amount in inches, if there is any.	Yes
32	Weather_Condition	Shows the weather condition (rain, snow, thunderstorm, fog, etc.)	Yes
33	Amenity	A POI annotation which indicates presence of amenity in a nearby location.	No
34	Bump	A POI annotation which indicates presence of speed bump or hump in a nearby location.	No

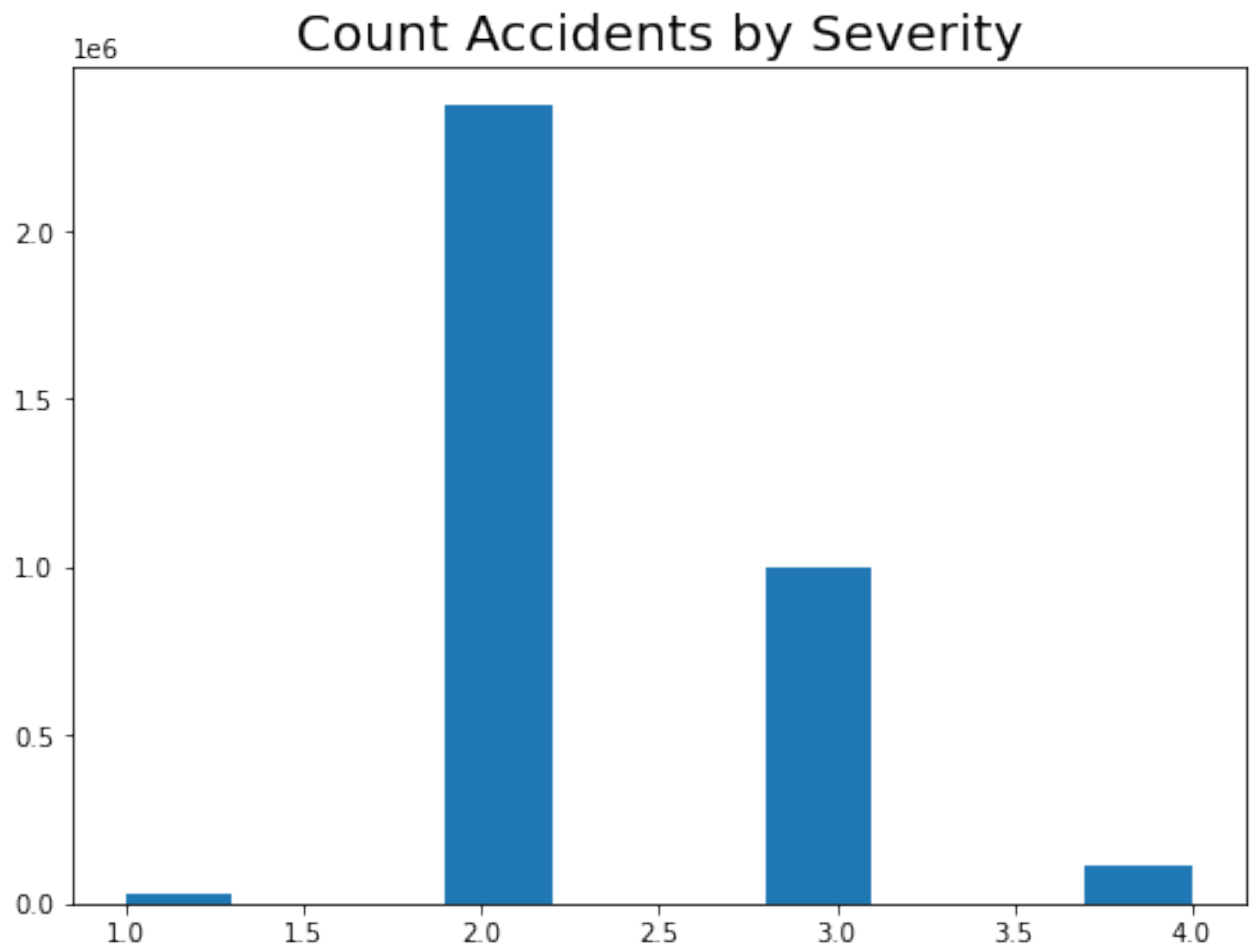
Acknowledgments

Please cite the following papers if you use this dataset:

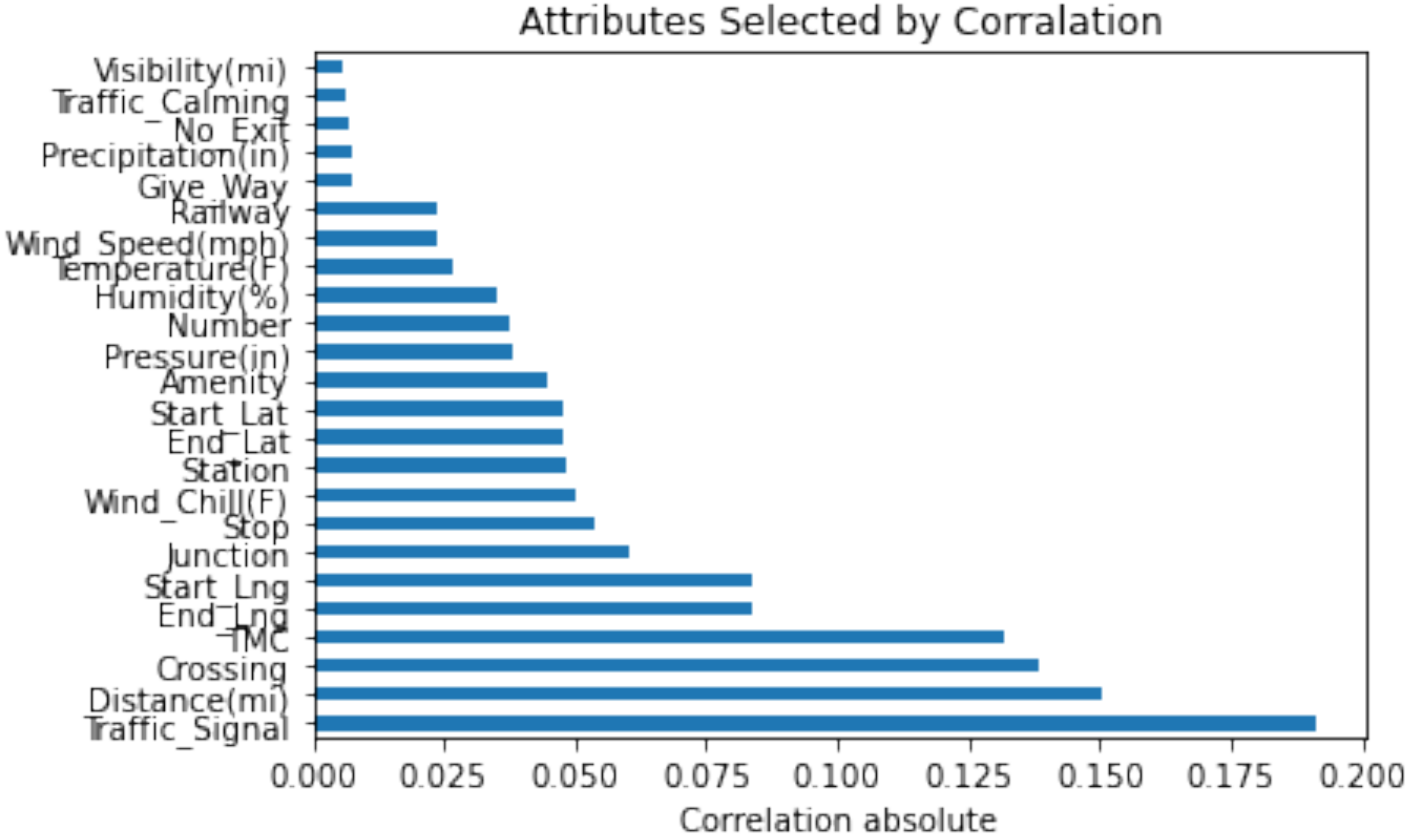
- Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, and Rajiv Ramnath. "A Countrywide Traffic Accident Dataset.", arXiv preprint arXiv:1906.05409 (2019).
- Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, Radu Teodorescu, and Rajiv Ramnath. "Accident Risk Prediction based on Heterogeneous Sparse Data: New Dataset and Insights." In proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2019.

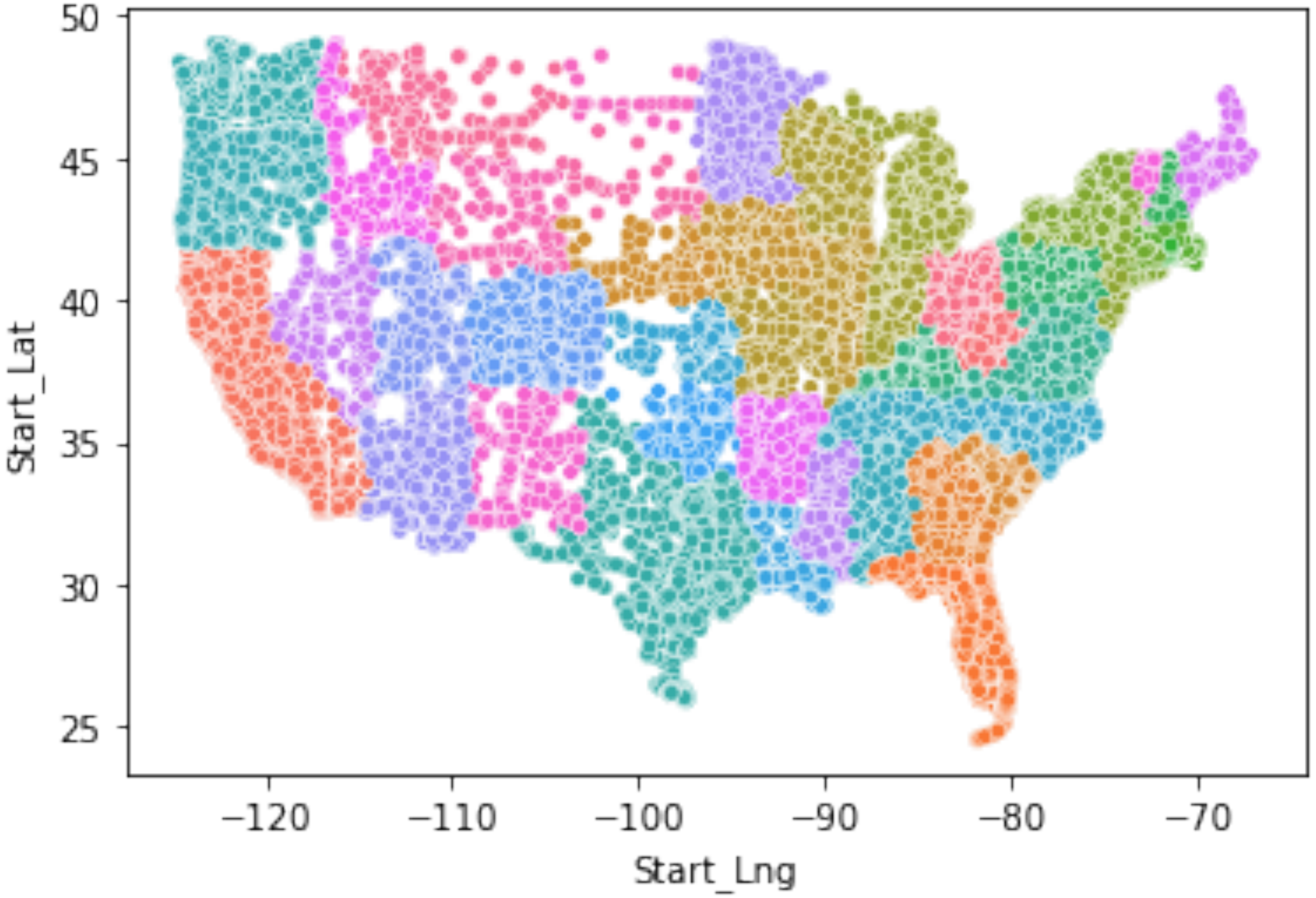
About Severity distribution in dataset



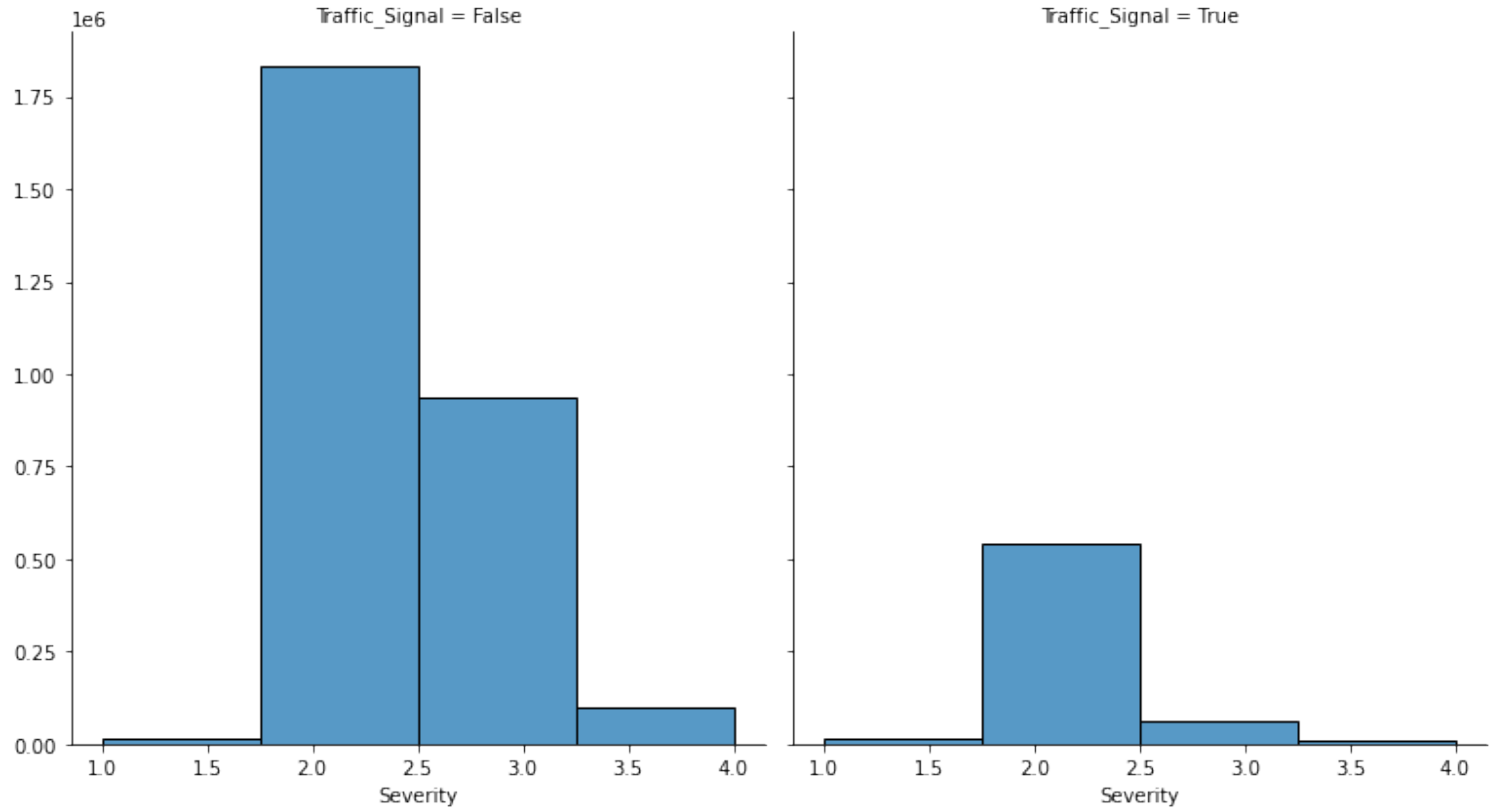


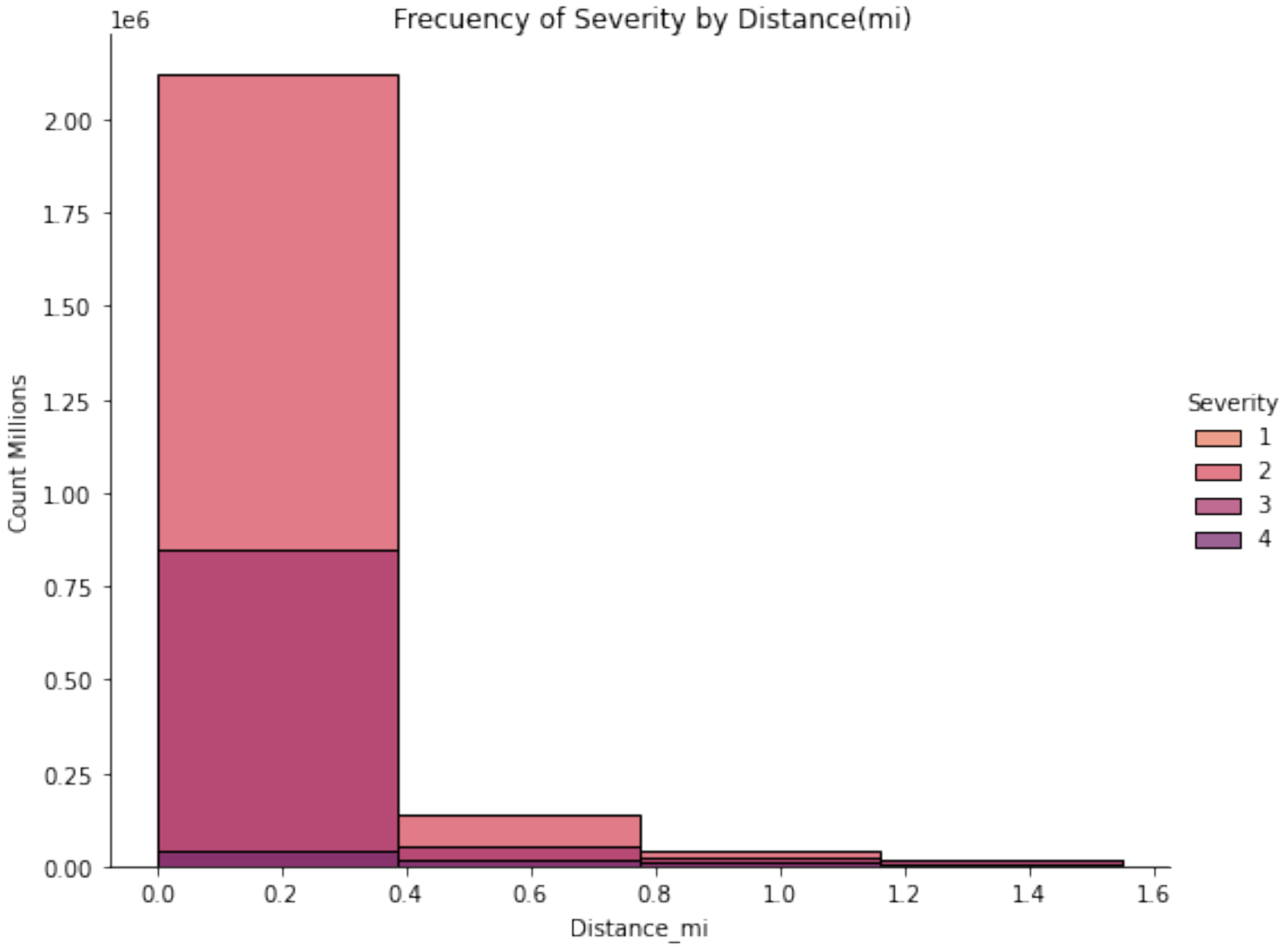
Data Understanding



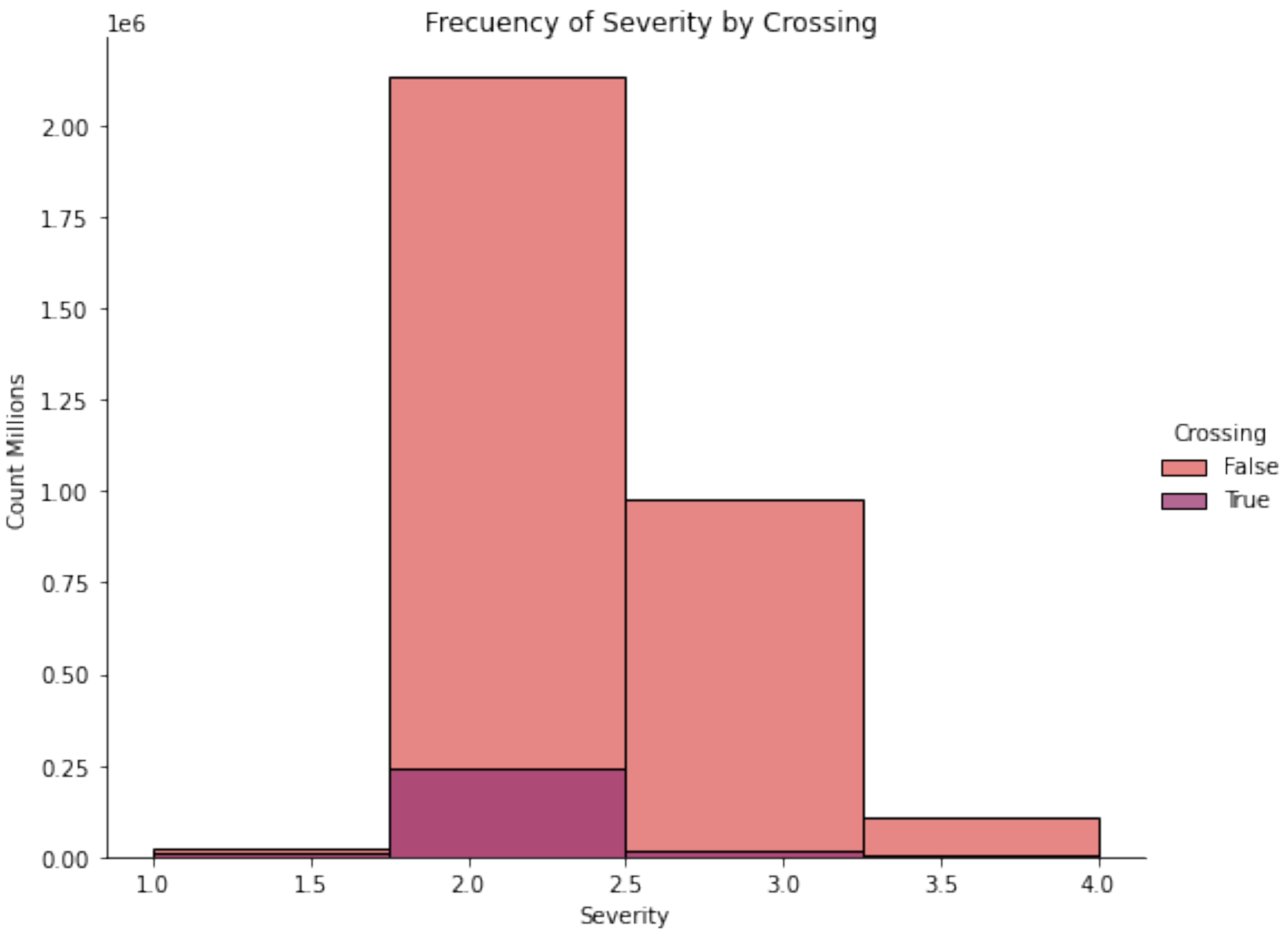


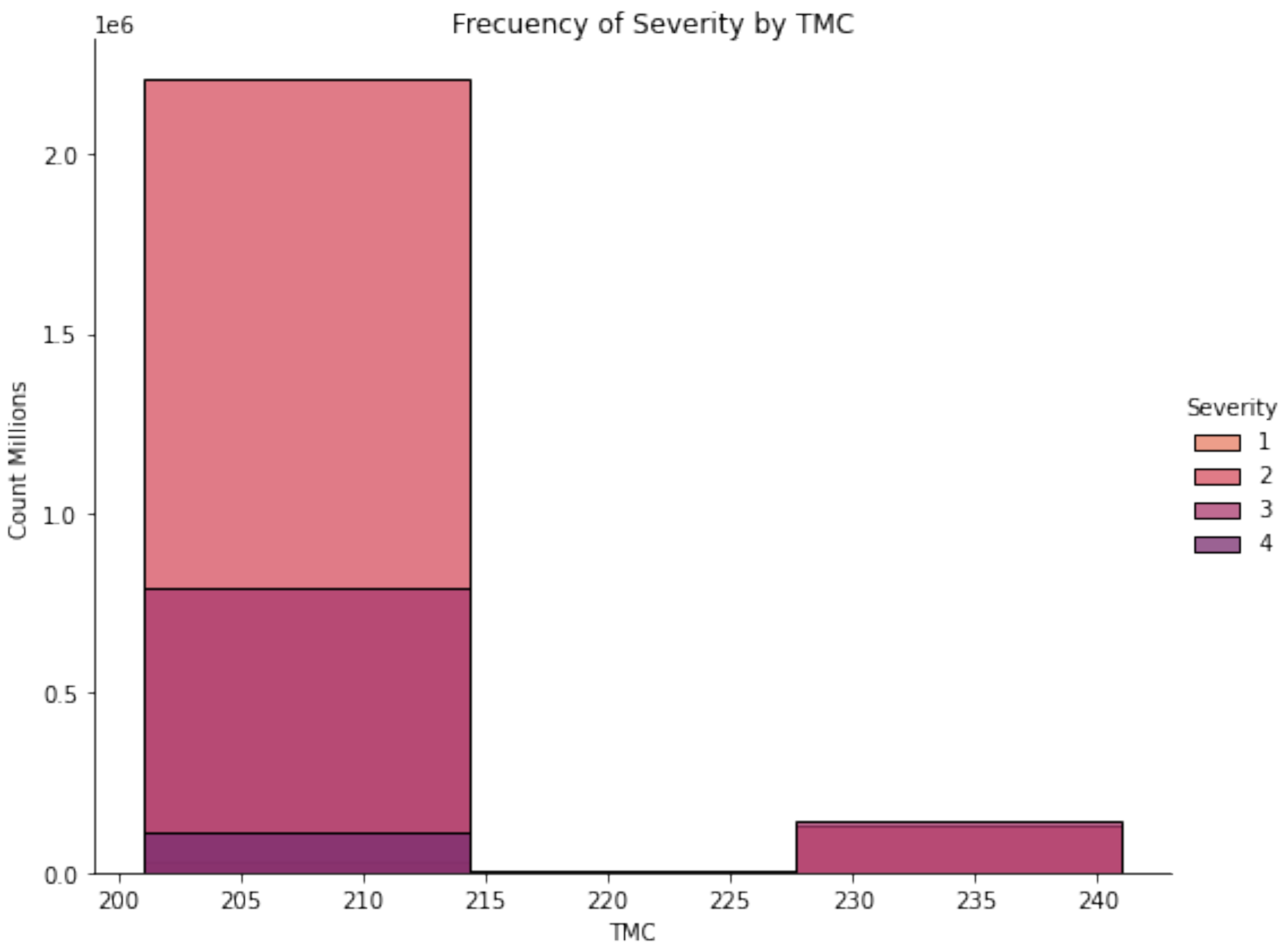
Data Understanding



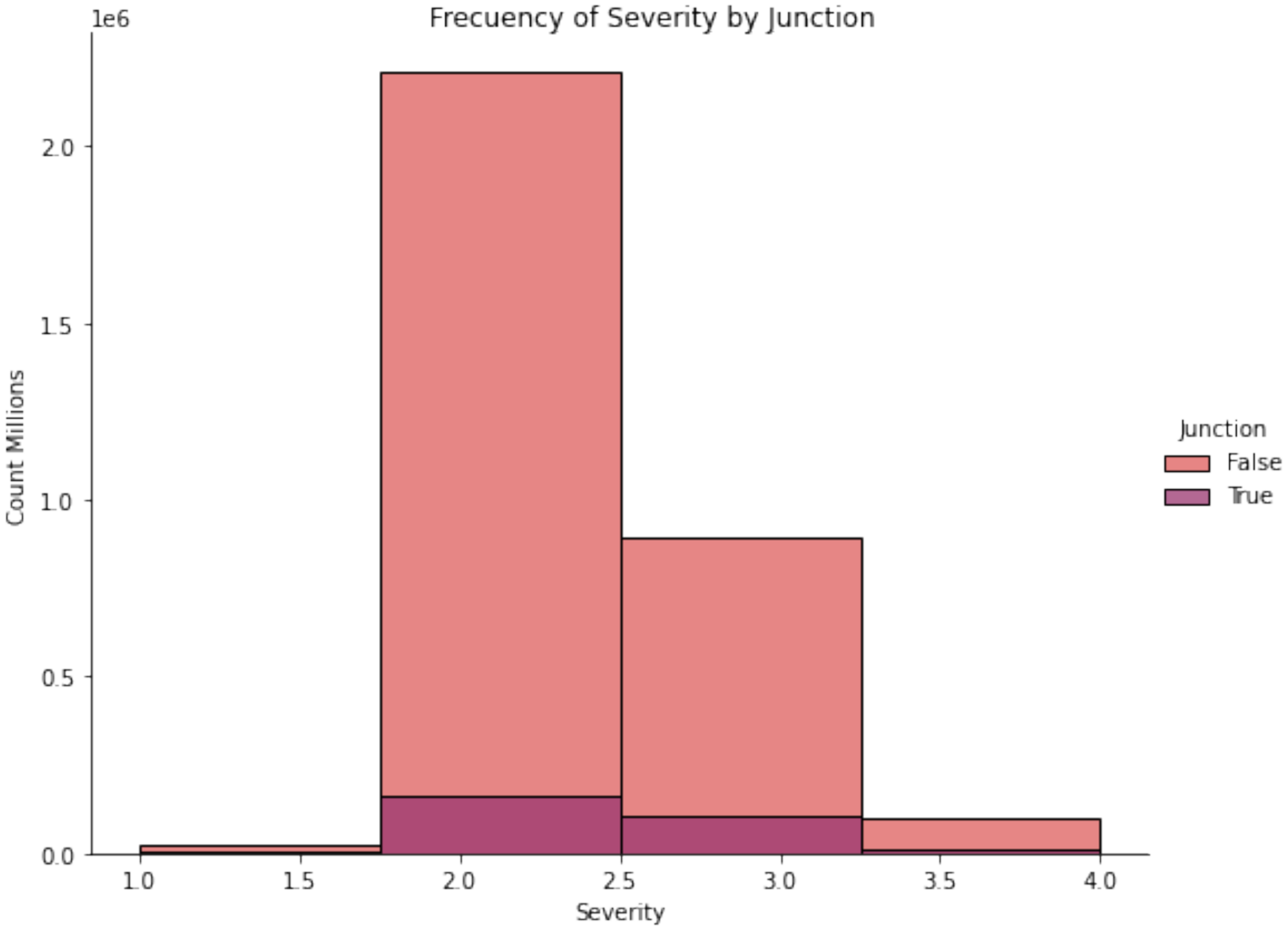


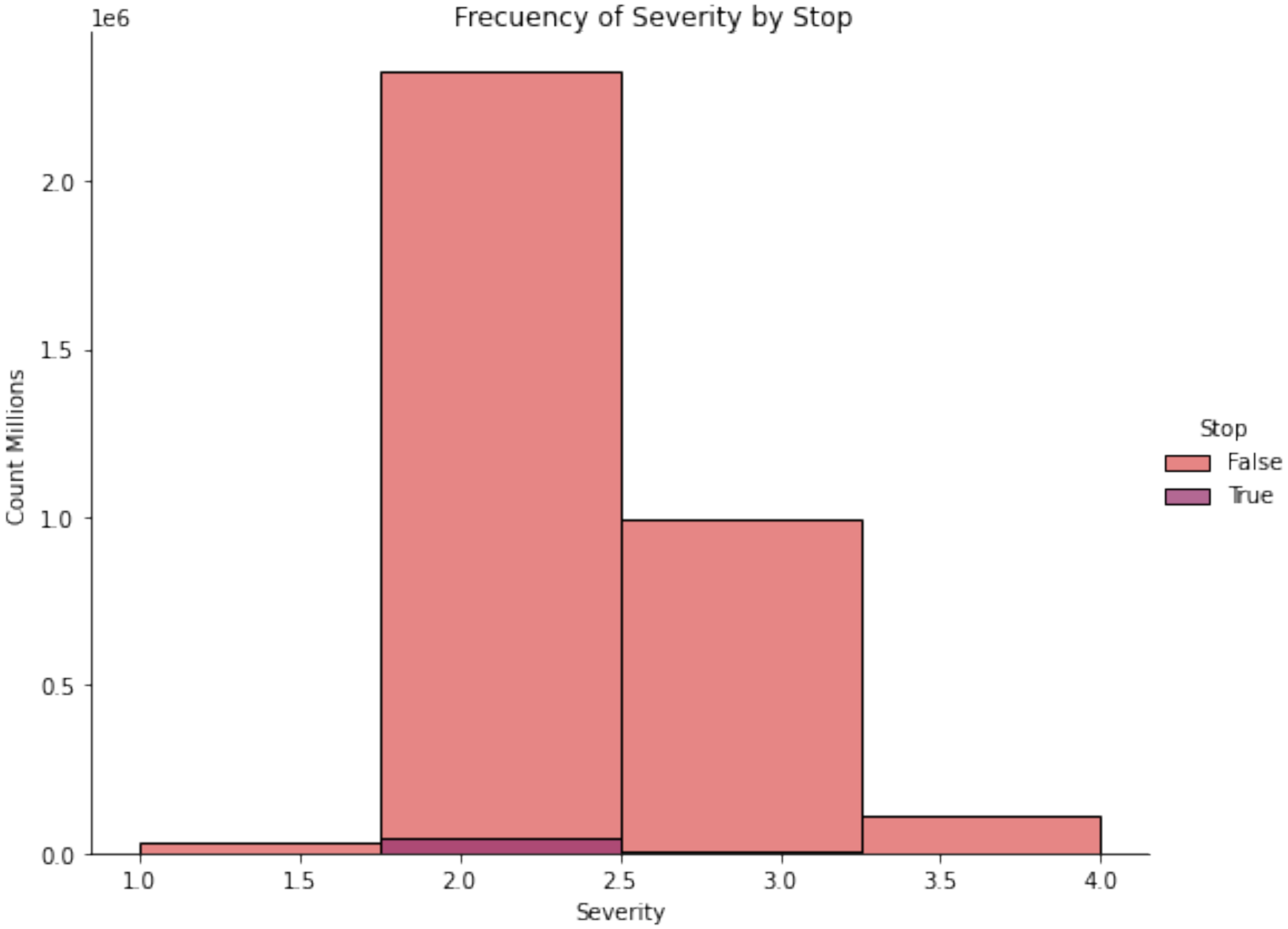
Data Understanding



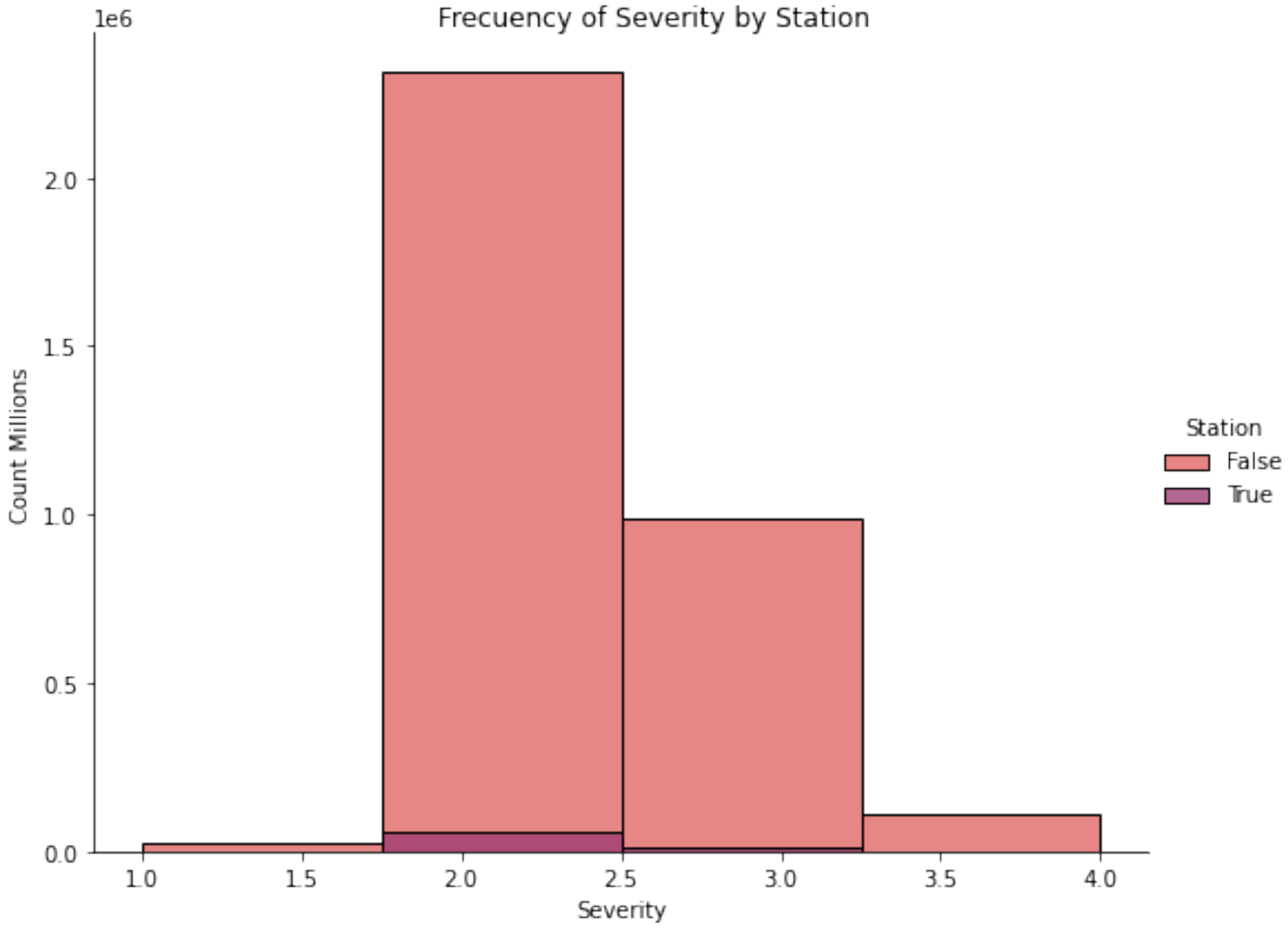


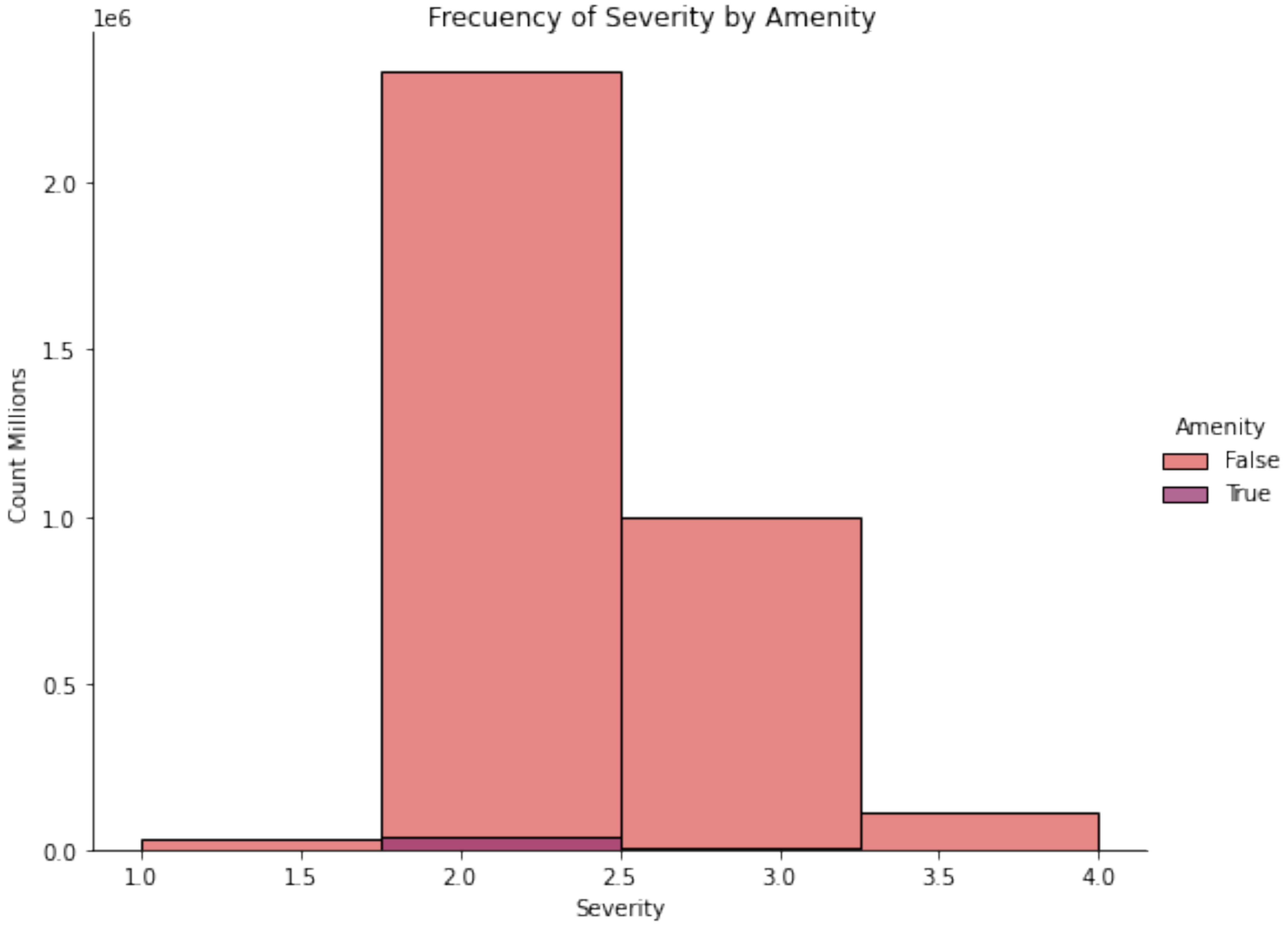
Data Understanding



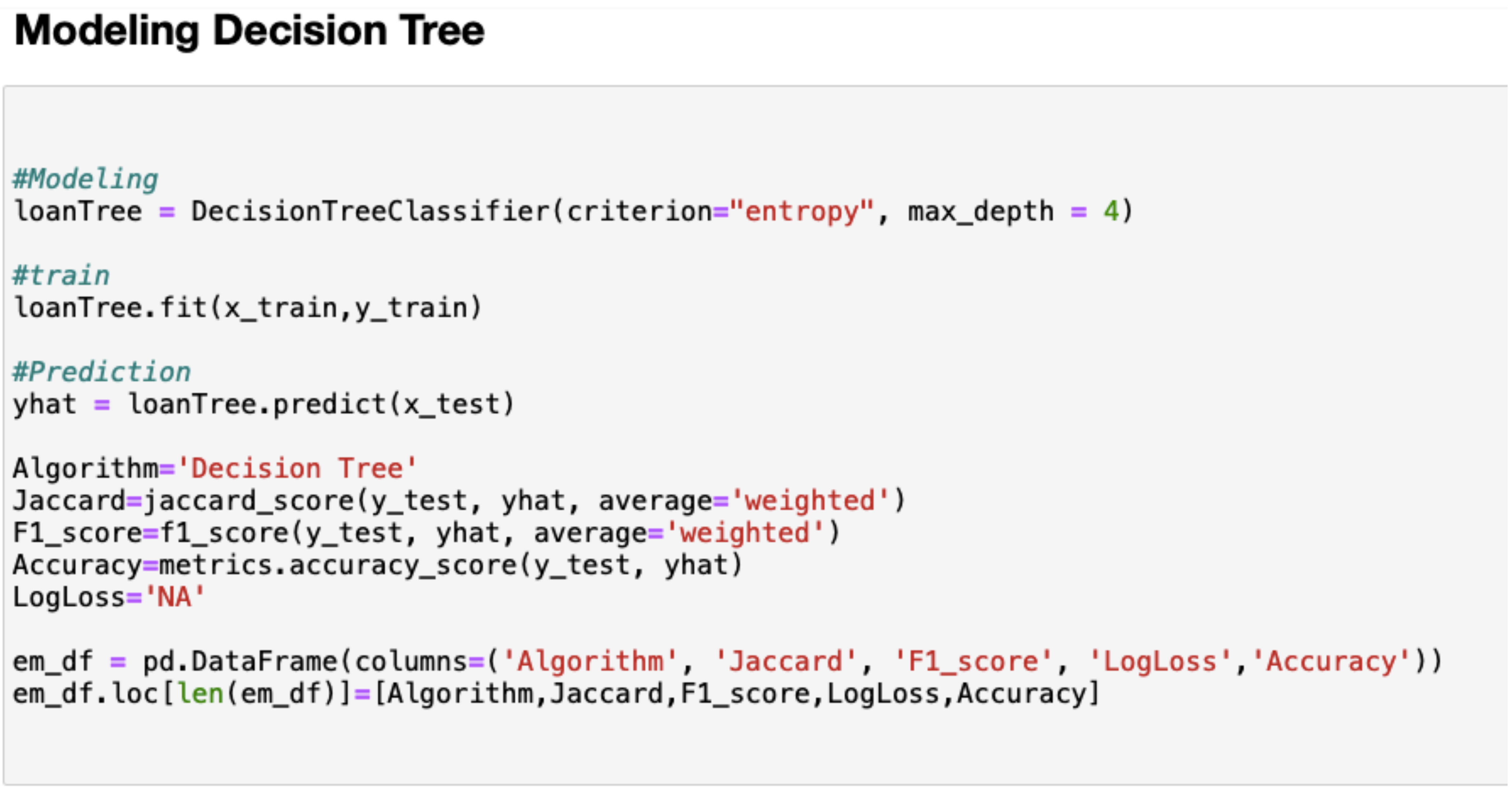


Data Understanding



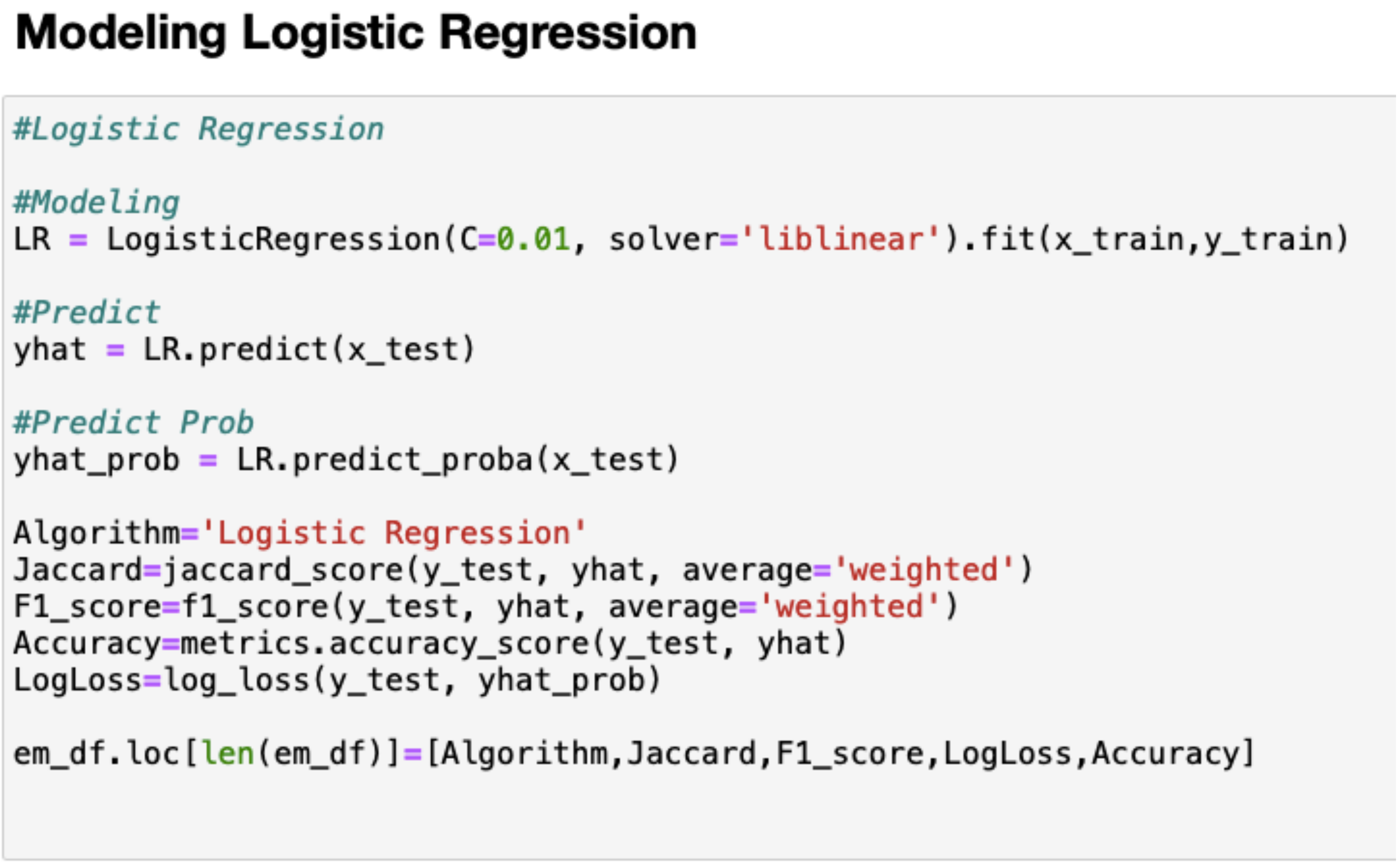


MODELING



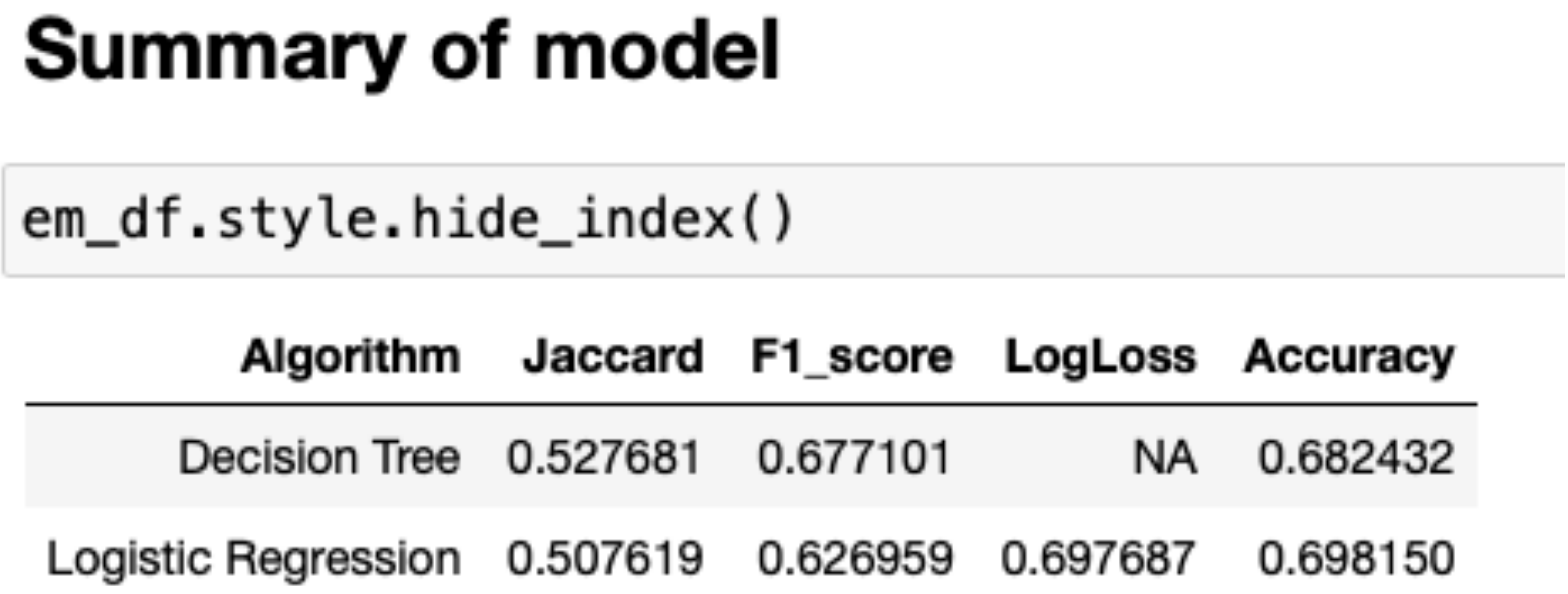
Modeling Decision Tree

```
#Modeling
loanTree = DecisionTreeClassifier(criterion="entropy", max_depth = 4)

#train
loanTree.fit(x_train,y_train)

#Prediction
yhat = loanTree.predict(x_test)

Algorithm='Decision Tree'
Jaccard=jaccard_score(y_test, yhat, average='weighted')
F1_score=f1_score(y_test, yhat, average='weighted')
Accuracy=metrics.accuracy_score(y_test, yhat)
LogLoss='NA'

em_df = pd.DataFrame(columns=('Algorithm', 'Jaccard', 'F1_score', 'LogLoss','Accuracy'))
em_df.loc[len(em_df)]=[Algorithm,Jaccard,F1_score,LogLoss,Accuracy]
```



Modeling Logistic Regression

```
#Logistic Regression

#Modeling
LR = LogisticRegression(C=0.01, solver='liblinear').fit(x_train,y_train)

#Predict
yhat = LR.predict(x_test)

#Predict Prob
yhat_prob = LR.predict_proba(x_test)

Algorithm='Logistic Regression'
Jaccard=jaccard_score(y_test, yhat, average='weighted')
F1_score=f1_score(y_test, yhat, average='weighted')
Accuracy=metrics.accuracy_score(y_test, yhat)
LogLoss=log_loss(y_test, yhat_prob)

em_df.loc[len(em_df)]=[Algorithm,Jaccard,F1_score,LogLoss,Accuracy]
```



Summary of model

```
em_df.style.hide_index()
```

Algorithm	Jaccard	F1_score	LogLoss	Accuracy
Decision Tree	0.527681	0.677101	NA	0.682432
Logistic Regression	0.507619	0.626959	0.697687	0.698150