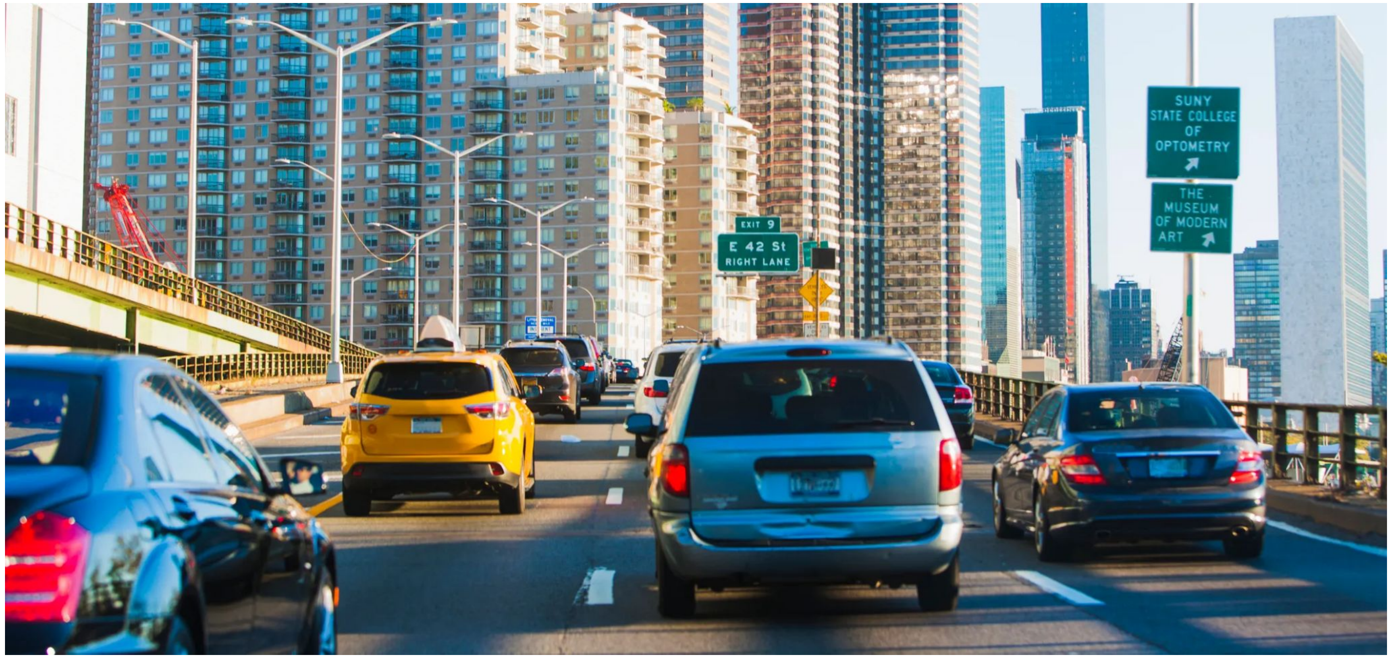
Assessing Fatal Factors in NYC Car-Pedestrian Crashes

ROSS BOEHME



Introduction

Background

- NYC is a famously **pedestrian-focused city**
- But cars kill an avg of **~100 pedestrians** each year
- The NYPD has data on **every registered car crash** since 2012
- Do these data contain patterns as to **what makes a car crash fatal**?
- If yes, **NYC policy and resources** can be adjusted to limit deaths

Research Question

To what degree are these factors predictive of **whether a car crashing into a pedestrian is likely to kill that pedestrian**?

- Cause of crash
- Time of day
- Vehicle type

Data Overview

- Data from NYC Open Data website
- One row per crash for July 2012-Dec 2023. 2M total observations.
- 11 initial fields of interest:

1. crash_date - Date when crash_occurred
2. crash_time - Time when crash occurred
3. persons_injured - Total persons (cyclists, motorists, pedestrians) injured by the crash
4. persons_killed - Total persons (cyclists, motorists, pedestrians) killed by the crash
5. pedestrians_injured - Pedestrians (not in a vehicle) injured by the crash
6. pedestrians_killed - Pedestrians (not in a vehicle) killed by the crash
7. location - Crash location (e.g. cross street)
8. borough - NYC borough
9. zip code - Zip location for crash
10. contributing_factor_vehicle1 - Contributing factor for first vehicle in crash
11. contributing_factor_vehicle2 - Contributing factor for second (if applicable) vehicle in crash

Presentation Outline

1. Data Acquisition and Transformation
2. Exploratory Data Analysis (EDA)
3. Research Question Analysis
4. Conclusions and Next Steps
5. Appendix (Abstract, Link to Jupyter notebook)

Data Acquisition and Transformation

Data Acquisition/Cleaning (1/3)

1. Use JSON to acquire dataframe from Citi Bike website

```
df = pd.DataFrame(requests.get("https://data.cityofnewyork.us/resource/h9gi-nx95.json?$limit=2500000").json())
```

2. Adjust data types so I can calculate summary statistics, aggregate, etc.

```
for i in numeric_cols:
    df[i] = pd.to_numeric(df[i])
df['crash_date'] = pd.to_datetime(df['crash_date'])
```

3. Assess initial summary statistics to refine approach

```
df.describe()
```

	number_of_persons_injured	number_of_persons_killed	number_of_pedestrians_injured	number_of_pedestrians_killed
count	2,050,483.00	2,050,470.00	2,050,501.00	2,050,501.00
mean	0.31	0.00	0.06	0.00
std	0.70	0.04	0.24	0.03
min	0.00	0.00	0.00	0.00
25%	0.00	0.00	0.00	0.00
50%	0.00	0.00	0.00	0.00
75%	0.00	0.00	0.00	0.00
max	43.00	8.00	27.00	6.00

Data Acquisition/Cleaning (2/3)

3. Filter for only rows where a pedestrian was injured or killed. 2M -> 111K rows.

```
df2 = df[(df['number_of_pedestrians_killed'].astype(int) > 0) |(df['number_of_pedestrians_injured'].astype(int) > 0) ]
```

4. Use domain knowledge to reduce number of inputs/features/X-variables. Remove if:
   a. No obvious reason to affect likelihood of a pedestrian being killed (e.g. crash_date)
   b. Too many values for me to dummify and correlate with number_of_pedestrians_killed (e.g. cross_street)
   c. Won't use in my EDA

```
df2 = df2[['crash_date','crash_time','number_of_pedestrians_injured','number_of_pedestrians_killed','vehicle_type_code1',
'contributing_factor_vehicle_1','crash_year','borough']]
df2.reset_index(drop=True)
```

```
df2.head()
```

	crash_date	crash_time	number_of_pedestrians_injured	number_of_pedestrians_killed	vehicle_type_code1	contributing_factor_vehicle_1	crash_year	borough
23	2021-12-14	3:43	1	0	Station Wagon/Sport Utility Vehicle	Unspecified	2021	NaN
25	2021-12-14	17:31	1	0	Sedan	Unspecified	2021	BROOKLYN
33	2021-12-16	6:59	1	0	NaN	Traffic Control Disregarded	2021	NaN
39	2021-07-09	0:43	0	1	Bus	Unspecified	2021	NaN
42	2022-04-22	17:17	1	0	E-Bike	Traffic Control Disregarded	2022	NaN

Data Acquisition/Cleaning (3/3)

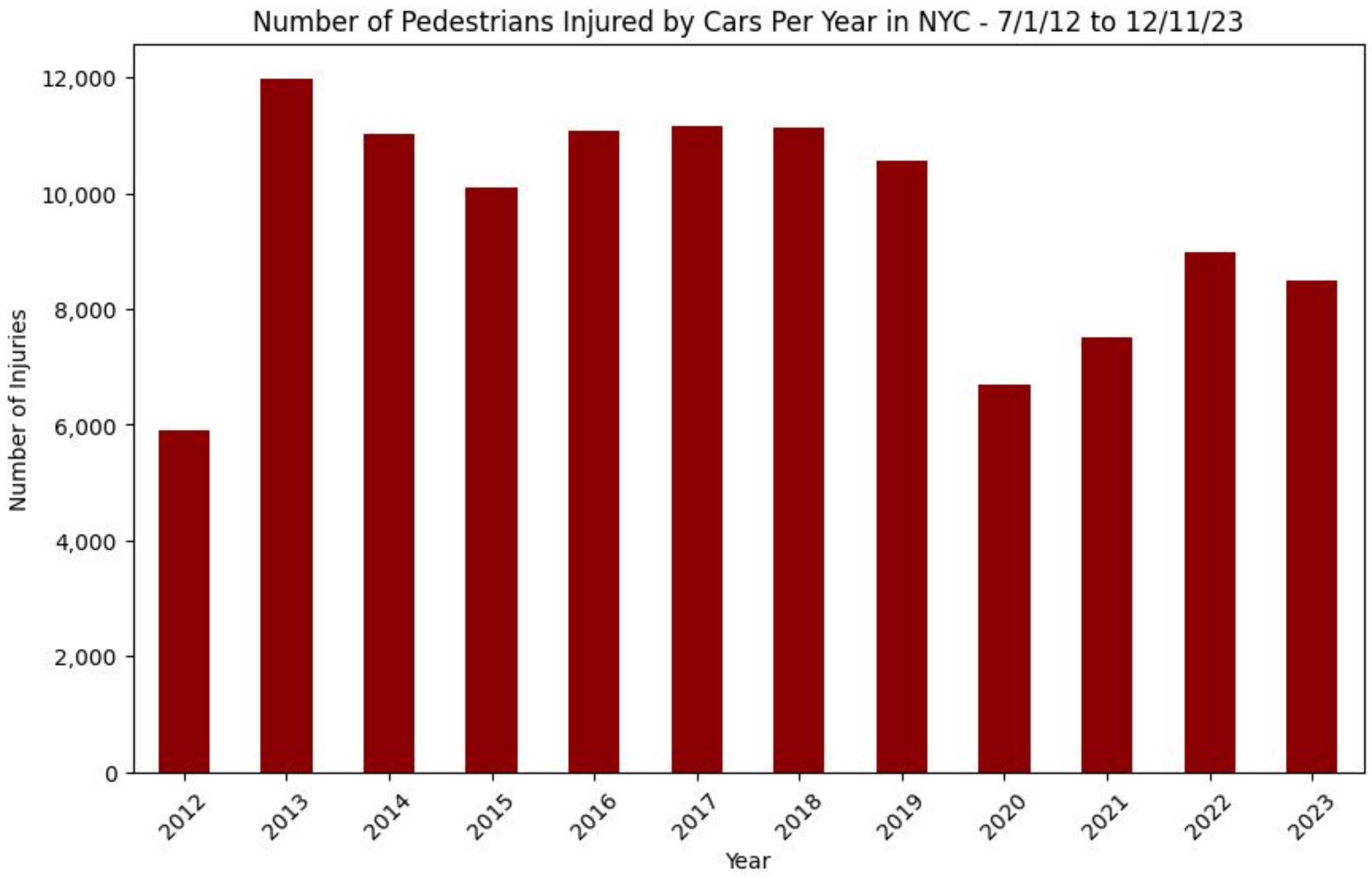
5. Group columns with many unique values: crash cause and vehicle type

```
df2['vehicle_clean'] = df2['vehicle_type_code1'].apply(lambda x:
      'Small Commercial Vehicle' if 'Taxi' in x else 'Bicycle' if 'Bicy' in x else 'E-Bike' if 'E-Bik' in x else 'Large Commercial
Vehicle' if 'ambul' in x else 'Large Commercial Vehicle' if 'bus' in x else 'Small Personal Vehicle' if 'Sedan' in x else 'Large
Commercial Vehicle' if ('Trac' or 'Trail') in x or 'Trail' in x else 'TBD')
      #Initially attempted to create rules but manual mapping is actually faster, easier ,and more accurate

vehicle_clean_map = {
    'Sedan':'Small Personal Vehicle', 'Convertible':'Small Personal Vehicle', 'Bike':'Bicycle', 'Minibike': 'Motorcycle', 'Schoolbus':
'Large Commercial Vehicle', 'Firetruck': 'Large Commercial Vehicle', Motorscooter: 'Moped', 'Pedicab':Bicycle'}
      #Only a sample of my mapping dictionary because there were more than 200 values

crash_cause_clean_map = {
    'Driver Inattention/Distraction': 'Driver Inattention', 'Failure to Yield Right-of-Way': 'Driver Error', 'Backing Unsafely':
'Driver Error', 'Pedestrian/Bicyclist/Other Pedestrian Error/Confusion': 'Pedestrian Error', 'View Obstructed/Limited': 'Driver',
'Passenger Distraction': 'Driver Inattention', 'Traffic Control Disregarded': 'Driver Error'} #Only a sample

df2['vehicle_clean'] = df2['vehicle_type_code1'].map(vehicle_clean_map)
df2['crash_cause'] = df2['contributing_factor_vehicle_1'].map(crash_cause_clean_map)
```

6. Categorizing blanks/unknowns

```
df2[vehicle_clean] = df2[vehicle_clean].fillna('Unknown/Other')
df2[crash_cause] = df2[crash_cause].fillna('Unspecified/Unknown')
```

Exploratory Data Analysis



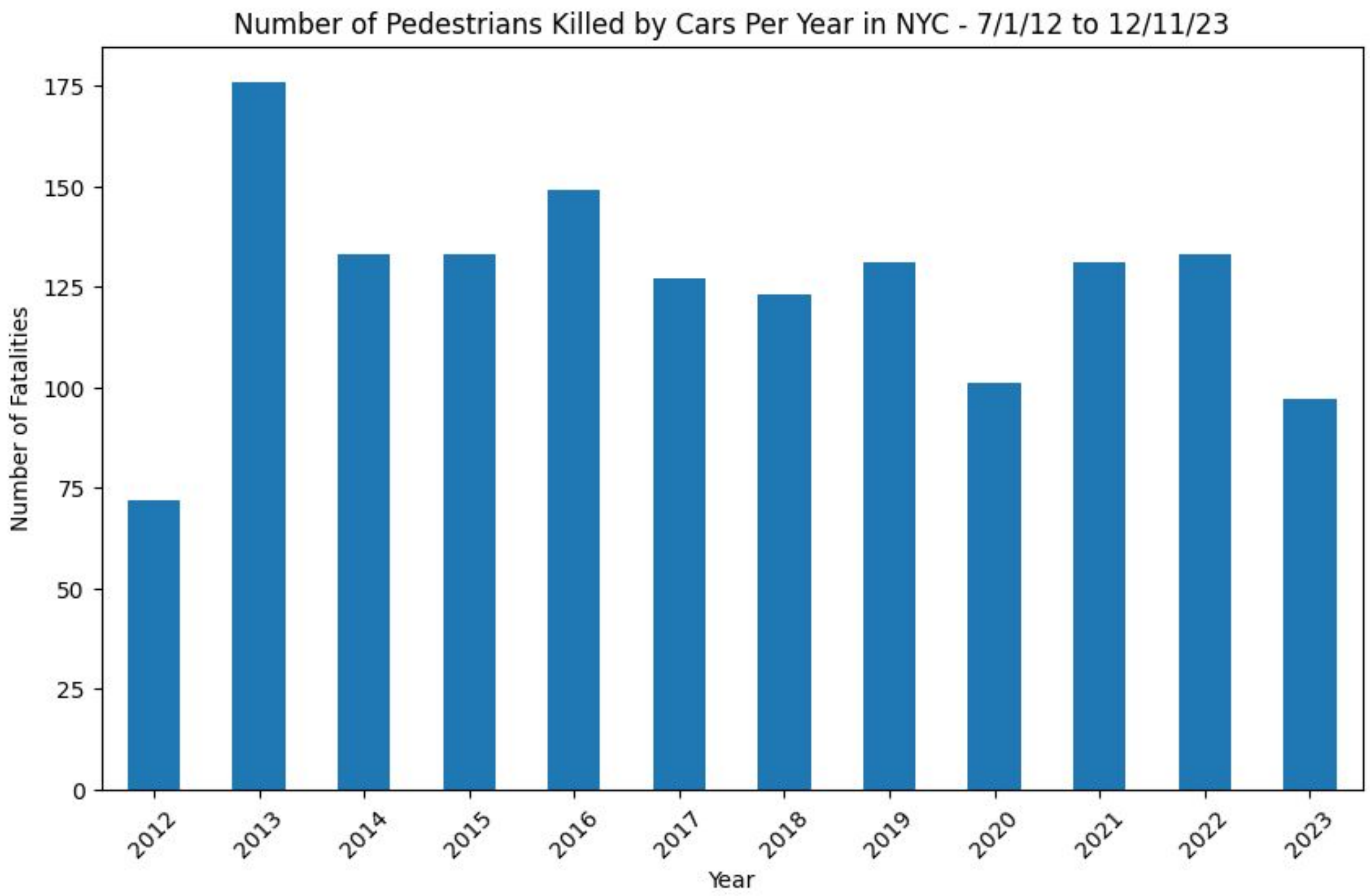
Data for 2012 and 2023 are incomplete. 2012 contains only data after 7/1/12 and 2023 contains only data through 12/11/23.

Slight decrease in injuries by year

Peaked in 2013, lowest in 2020

Avg of ~9.5K/year for 11 years

Earliest data available 7/1/12



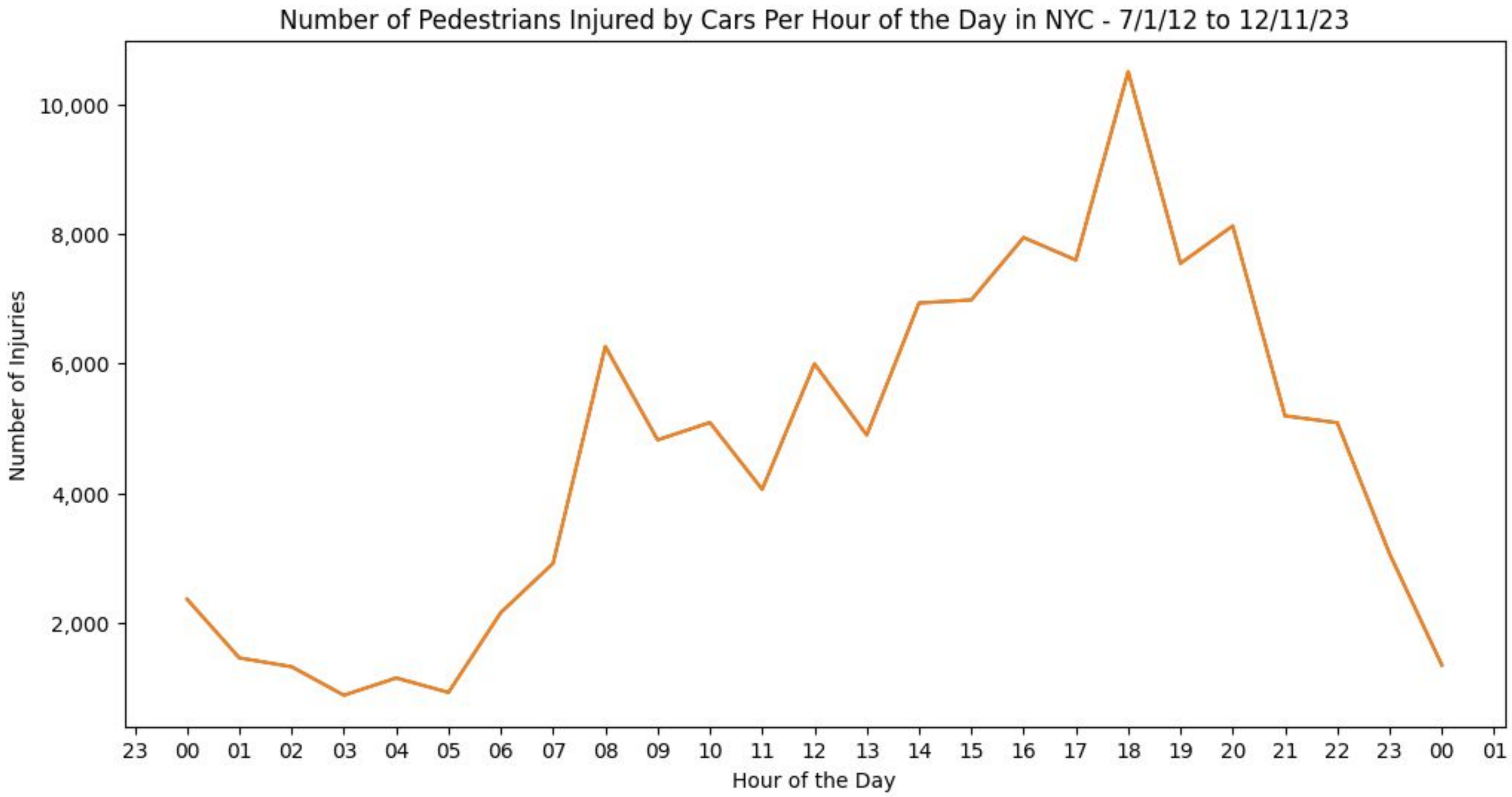
Data for 2012 and 2023 are incomplete. 2012 contains only data after 7/1/12 and 2023 contains only data through 12/11/23.

Slight decrease in deaths by year

Injuries/deaths generally correlated

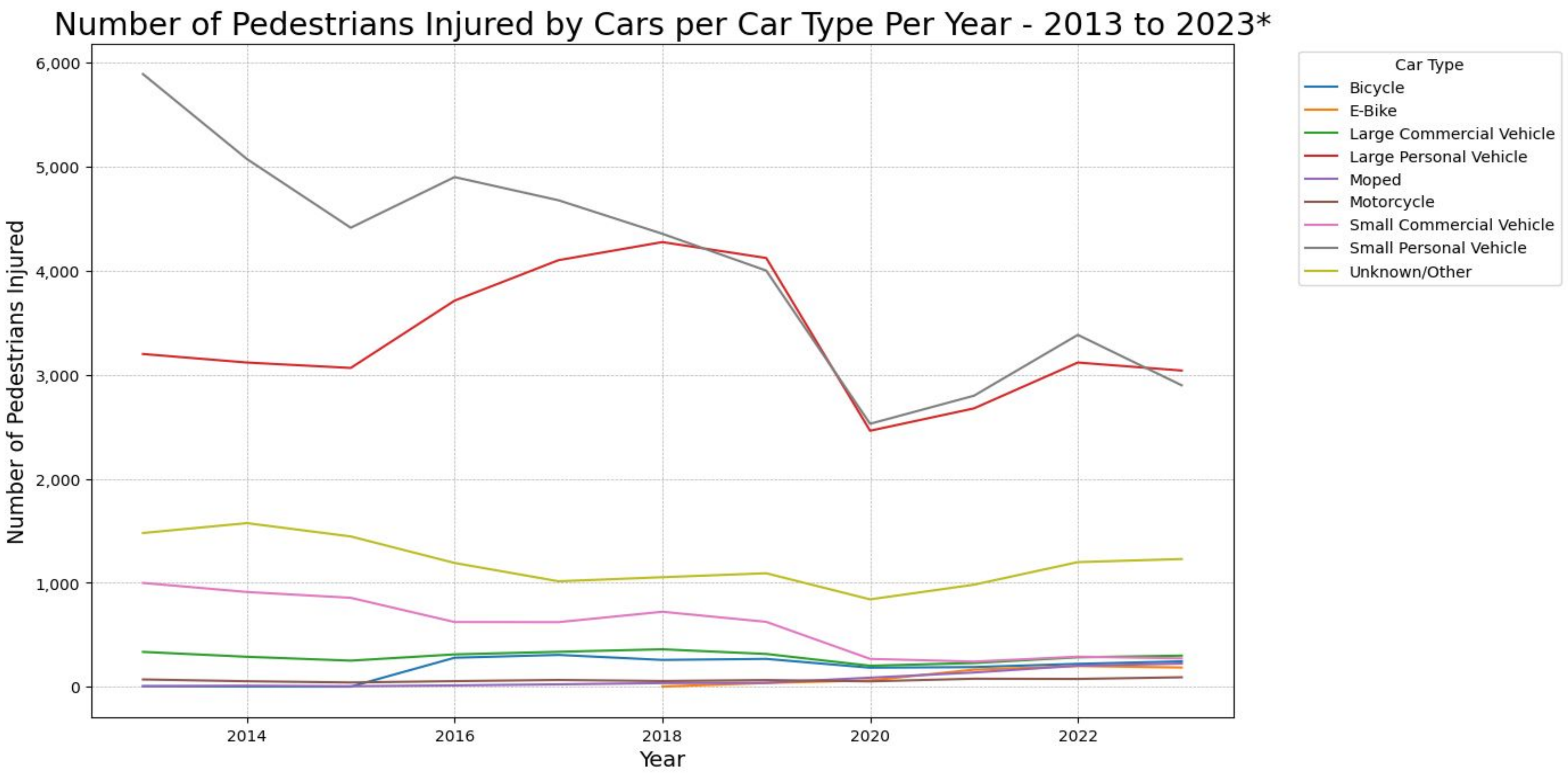
Avg ~125/year for 11 years

Smaller sample -> higher variance



Peaks during commute times, esp. afternoon

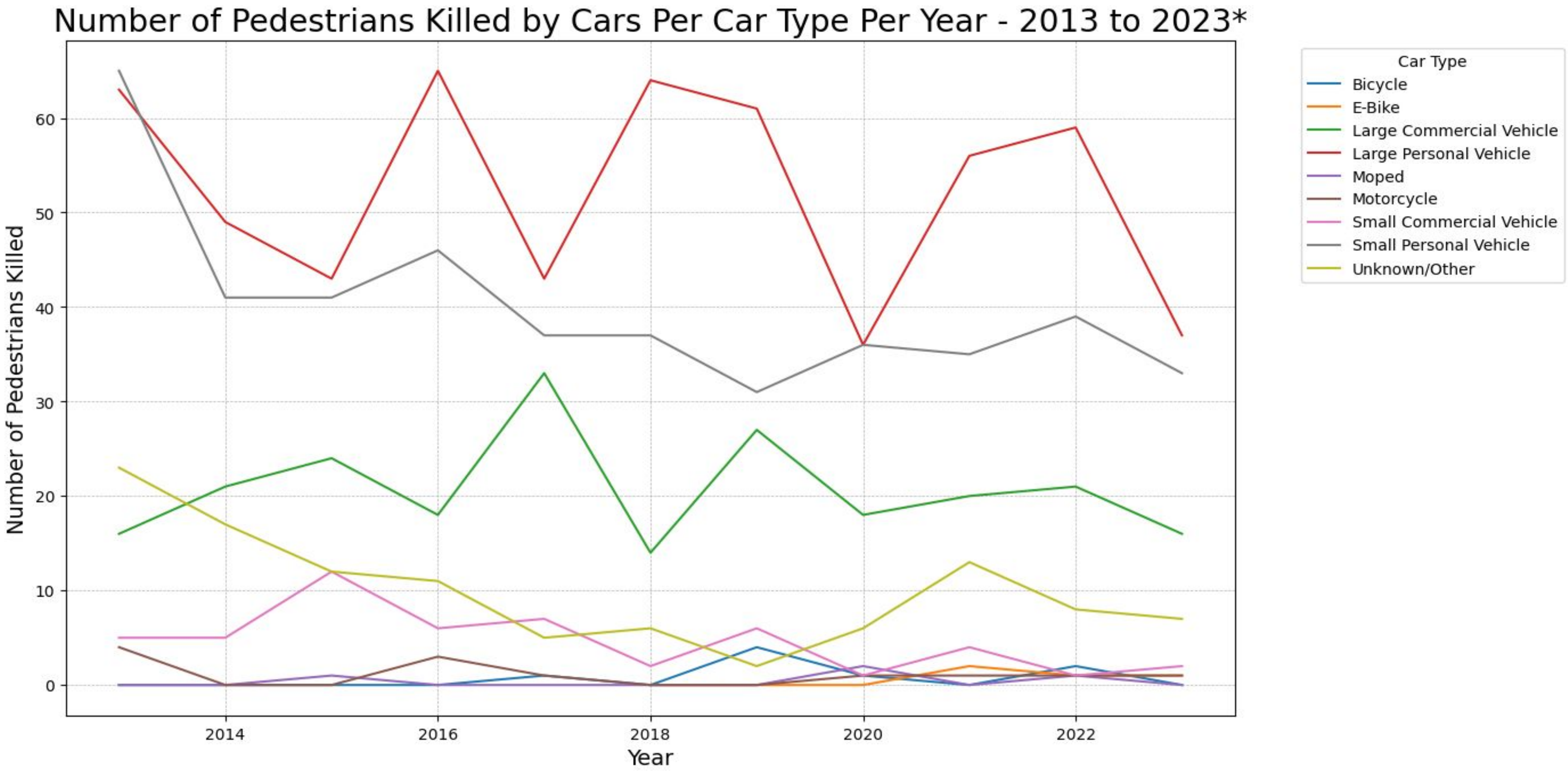
Afternoon commute has lower visibility combined with pedestrians at leisure



*2023 only contains only data through 12/11/23

Personal vehicles most likely to injure pedestrians

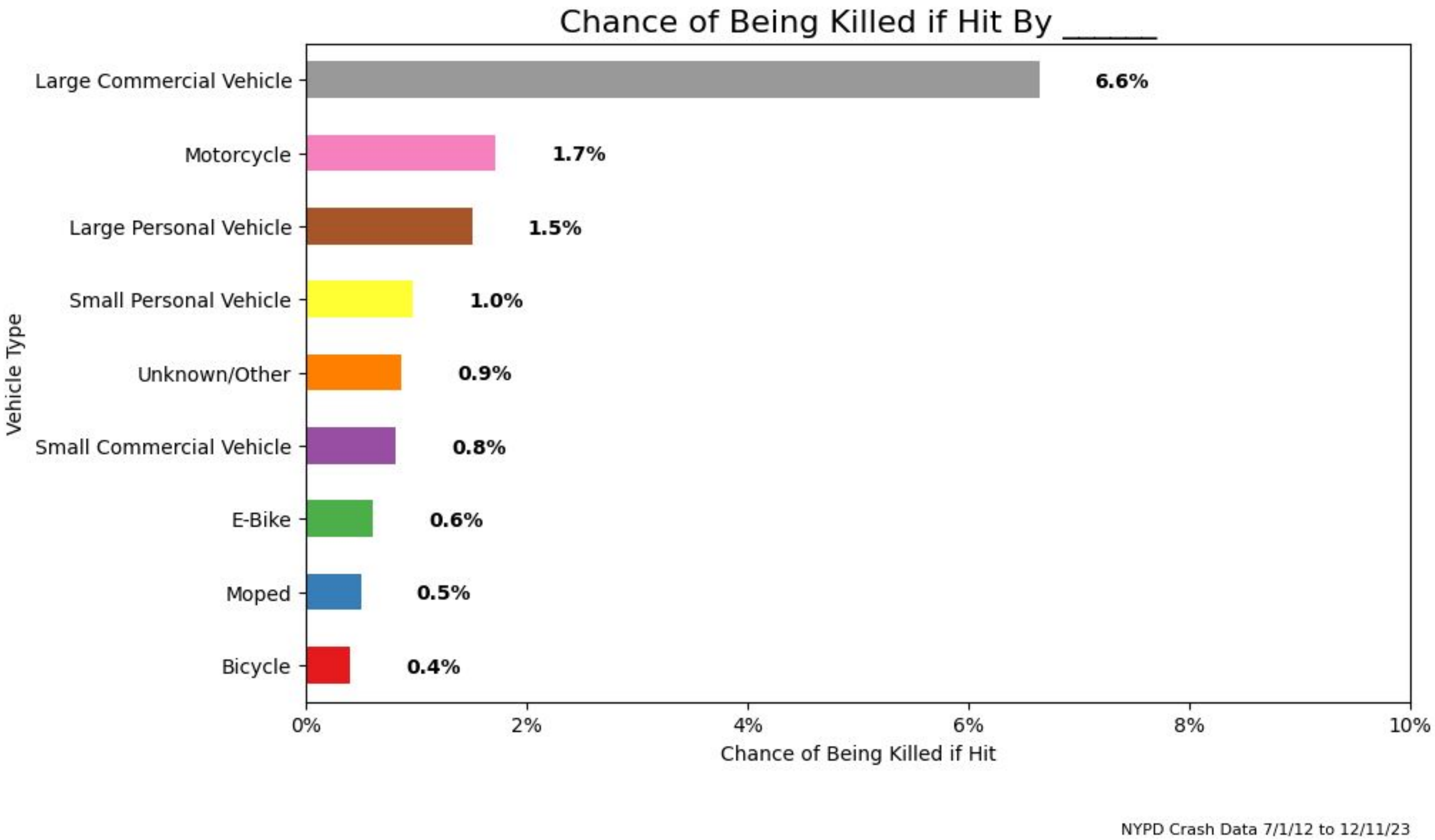
But personal vehicles also account for by far the most trips



*2023 only contains only data through 12/11/23

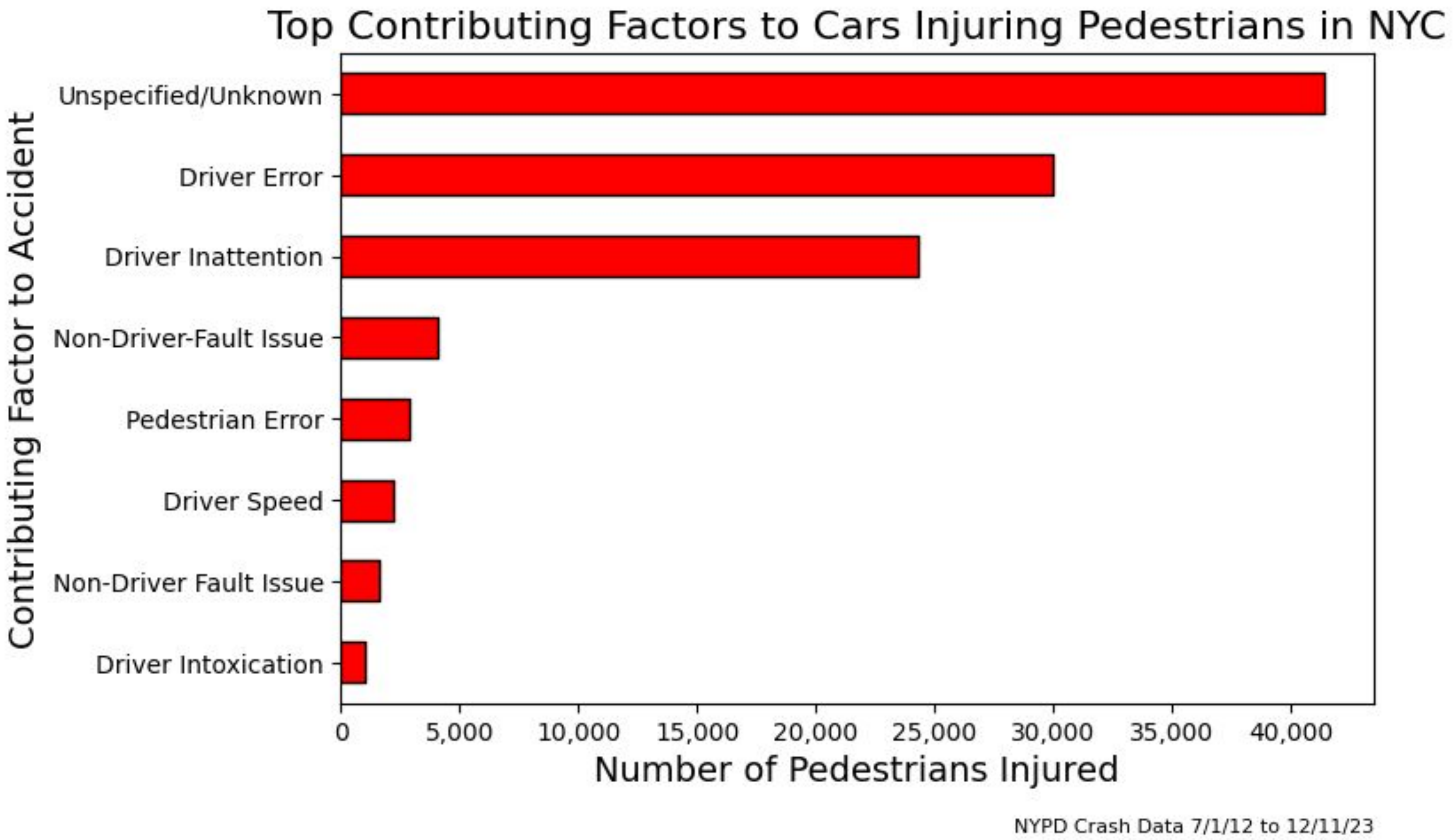
Personal vehicles still most likely to kill

But large vehicles, personal or commercial, more prominent on this graph



Large vehicles generally deadly

Large commercial vehicles heaviest -> deadliest



Issues with not reporting causes

~40% of crashes have no explanation listed

Drivers most often at fault

Of tracked causes: 87% driver fault, pedestrian 4%

Research Question Analysis

Research Question

To what degree are these factors predictive of **whether a car crashing into a pedestrian is likely to kill that pedestrian**?

- Cause of crash
- Time of day
- Vehicle type

Model Data Engineering (1/4)

1. Evaluate current state of data

crash_date	crash_time	crash_year	vehicle_type	pedestrians_injured	pedestrians_killed	borough	crash_cause	crash_hour
2021-12-14	2000-01-01 03:43:00	2021	Large Personal Vehicle	1	0	Unknown	Unspecified/Unknown	2000-01-01 04:00:00
2021-12-14	2000-01-01 17:31:00	2021	Small Personal Vehicle	1	0	BROOKLYN	Unspecified/Unknown	2000-01-01 18:00:00
2021-12-16	2000-01-01 06:59:00	2021	Unknown/Other	1	0	Unknown	Driver Error	2000-01-01 07:00:00
2021-07-09	2000-01-01 00:43:00	2021	Large Commercial Vehicle	0	1	Unknown	Unspecified/Unknown	2000-01-01 01:00:00
2022-04-22	2000-01-01 17:17:00	2022	E-Bike	1	0	Unknown	Driver Error	2000-01-01 17:00:00

2. Drop irrelevant columns and group relevant columns into larger categories

```
df4 = df4[['vehicle_type','pedestrians_injured','pedestrians_killed','crash_cause','crash_hour']]
df4['crash_hour'] = pd.to_datetime(df4['crash_hour'])
hour_bins = [0, 6, 12, 18, 24]
hour_labels = ['night', 'morning', 'afternoon', 'night']
df4['crash_time'] = pd.cut(df4['crash_hour'].dt.hour, bins=hour_bins, labels=hour_labels, include_lowest=True, ordered=False)
```

vehicle_type	crash_cause	crash_time	pedestrians_killed
Large Personal Vehicle	Unspecified/Unknown	night	0
Small Personal Vehicle	Unspecified/Unknown	afternoon	0
Unknown/Other	Driver Error	morning	0
Large Commercial Vehicle	Unspecified/Unknown	night	1
E-Bike	Driver Error	afternoon	0

Model Data Engineering (2/4)

3. Dummify features/inputs/X-variables

```
df5 = pd.get_dummies(df4, columns=['vehicle_type', 'crash_cause', 'crash_time'])
```

4. Create Pearson correlation matrix to remove features w/ lowest correlation to label

```
correlation_matrix = df5.corr().abs()
low_corr_cols = correlation_matrix[abs(correlation_matrix['pedestrians_killed']) < 0.02].index.tolist()
```

	pedestrians_killed
vehicle_type_motorcycle	0.00
vehicle_type_e_bike	0.00
vehicle_type_moped	0.01
crash_cause_non_driver_fault_issue	0.01
crash_cause_pedestrian_error	0.01
vehicle_type_bicycle	0.01
vehicle_type_small_commercial_vehicle	0.01
crash_time_morning	0.01
crash_cause_driver_inattention	0.01
vehicle_type_large_personal_vehicle	0.01
vehicle_type_unknown_other	0.01
crash_cause_driver_error	0.02
crash_cause_driver_speed	0.02
crash_cause_unspecified_unknown	0.02
vehicle_type_small_personal_vehicle	0.02

Model Data Engineering (3/4)

3. Evaluate remaining features and label to identify best model for predicting pedestrian deaths

pedestrians_killed	vehicle_type_large_commercial_vehicle	crash_cause_driver_intoxication	crash_time_night
0	0	0	1
0	0	0	0
0	0	0	0
1	1	0	1
0	0	0	0

Poisson? Probably not, regression coefficients are negative but should be positive

```
poisson_model = sm.GLM(df5["pedestrians_killed"], df5.iloc[:, 1:], family=sm.families.Poisson()).fit()
print(poisson_model.summary())
```

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:     pedestrians_killed   No. Observations:               111247
Model:                            GLM   Df Residuals:                   111242
Model Family:                 Poisson   Df Model:                            4
Link Function:                    Log   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -23692.
Date:                Sun, 17 Dec 2023   Deviance:                       44397.
Time:                        18:26:31   Pearson chi2:                 5.11e+05
No. Iterations:                     9   Pseudo R-squ. (CS):            -0.3259
Covariance Type:            nonrobust
=========================================================================================================
                                            coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------------------------
vehicle_type_large_commercial_vehicle    -1.6846      0.065    -26.093      0.000      -1.811      -1.558
vehicle_type_small_personal_vehicle      -3.2677      0.046    -70.846      0.000      -3.358      -3.177
crash_cause_driver_intoxication          -0.7426      0.140     -5.301      0.000      -1.017      -0.468
crash_time_afternoon                     -4.1065      0.049    -83.215      0.000      -4.203      -4.010
crash_time_night                         -3.3696      0.036    -93.773      0.000      -3.440      -3.299
=========================================================================================================
```

Model Data Engineering (4/4)

3. (Continued) OLS? In theory not best but coefficients look good even if R-squared is weak.

```
X1 = df5.drop('pedestrians_killed', axis=1)
y1 = df5['pedestrians_killed']
X1 = sm.add_constant(X1)
model1 = sm.OLS(y1, X1).fit()
print(model1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     pedestrians_killed   R-squared:                       0.010
Model:                            OLS   Adj. R-squared:                  0.010
Method:                 Least Squares   F-statistic:                     369.2
Date:                Sun, 17 Dec 2023   Prob (F-statistic):         1.17e-238
Time:                        19:08:53   Log-Likelihood:                 80779.
No. Observations:              111247   AIC:                        -1.616e+05
Df Residuals:                  111243   BIC:                        -1.615e+05
Df Model:                           3
Covariance Type:            nonrobust
=========================================================================================================
                                         coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------------------
const                                  0.0078      0.000     17.660      0.000       0.007       0.009
vehicle_type_large_commercial_vehicle  0.0591      0.002     29.145      0.000       0.055       0.063
crash_cause_driver_intoxication        0.0365      0.004      9.807      0.000       0.029       0.044
crash_time_night                       0.0099      0.001     13.549      0.000       0.009       0.011
==============================================================================
Omnibus:                   167304.298   Durbin-Watson:                   1.983
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         79512261.959
Skew:                           9.398   Prob(JB):                         0.00
Kurtosis:                     132.616   Cond. No.                         11.4
==============================================================================
```

Do the **cause** of crash, **time** of day, and/or **type of vehicle** involved predict whether a pedestrian will be killed?

- Certain vehicle types, crash causes, and times of day have a **statistically significant** ($p < 0.05$) relationship with pedestrian deaths

- Pedestrian **base likelihood** of death if hit by a car: 0.78%
    - If hit by a **large commercial vehicle**: +5.9%
    - If hit by a **drunk driver**: +3.6%
    - If hit at **night**: +1.0%

- But my OLS regression model using these factors can only explain a very small amount (R^2=0.01) of the likelihood of a pedestrian death.
- **More factors are needed** (e.g. speed, location) to improve model

Conclusions / Next Steps

Conclusions

- Assessing vehicle types, causes of crashes, and hour of day, my model found the crash factors which most increase the likelihood of a pedestrian being killed when hit by a car are: (These percentages are relative to a randomly selected crash in NYC)
  - Being hit by a **large commercial vehicle**: +5.9% risk of death
  - Being hit by a **drunk driver**: +3.6%
  - Being hit at **night**: +1.0%

- My model may be **underestimating** as its R^2=0.01
- **Vehicle weight** appears to be the deadliest factor in a crash
- Within the data, the % of pedestrians killed when hit per vehicle type are:
  - Large commercial vehicles: 6.6%
  - Motorcycles: 1.7%
  - Large personal vehicles: 1.5%

Next Steps

For improving the model:

- Add **more inputs** such as crash speed, location to increase R^2
- Explore **more model options** beyond Poisson and OLS

For implementing findings:

- NYC perform a full cost-benefit analysis about whether there are feasible **policy changes** to discourage lethal crash factors. Eg:
    - Large vehicles **taxed or fined** at higher rates
    - Investment in **public infrastructure** to prevent drunk drivers
    - Locations with recurring deaths should be **better lit** at night
- Target ***drivers*** with changes as they're at fault 87% of the time

Appendix

Abstract

NYC is known for being a pedestrian friendly city but more than 10,000 people are hit by cars every year, including more than 100 who are killed. These numbers haven't meaningfully improved in the 11 years since records have been kept. What's more, there are consistent factors in lethal car crashes which could be ameliorated by government policy or investment. This analysis examines vehicle types, the cause of the crash, and time of day to understand what makes a fatal accident. Conclusions include that heavier vehicles, drunk drivers, and night driving greatly increase the statistical likelihood of a pedestrian being killed if a car hits them. Policy and infrastructure changes should be focused on drivers as they were at fault in 87% of all analyzed car/pedestrian collisions.

Jupyter Notebook Link

- RPubs file including works cited: https://colab.research.google.com/drive/1cFcYSuXvp8oEXFmFIqirF-mU0486hcW-1?usp=sharing

Thank You