Autonomous Driving – Part 2

Step 1 – Data Inspection & Cleaning

- Load `Tesla-Deaths.csv` dataset
- Check data types, missing values, and duplicates
- Drop irrelevant columns (if any)

Step 2 – Exploratory Data Analysis (EDA)

2.1 – Events Over Time

- Accidents per year
- Accidents per month/date
- Accidents per state & country

2.2 – Victim Analysis

- Number of victims (deaths) per accident
- Count of Tesla driver deaths
- Proportion of events with at least one occupant death

2.3 – Collision Analysis

- Distribution of collisions with cyclists/pedestrians
- Cases with Tesla occupant + cyclist/pedestrian both dead
- Frequency of Tesla colliding with other vehicles

Step 3 – Model & Autopilot Analysis

- Event distribution across Tesla models
- Verified Tesla Autopilot deaths distribution
- Compare Verified deaths vs All reported deaths (NHTSA)

Step 4 – Visualization & Insights

- Use Matplotlib/Seaborn plots for trends and distributions
- Summarize key insights from the data

In [2]:
```
# ============================================================
# Step 1 – Data Inspection & Cleaning
# ============================================================
```

```python
# Importing the required libraries
# Why? -> pandas for data handling, numpy for numerical operations,
#         matplotlib & seaborn for visualization later.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# 1.1 Load the dataset
# Why? -> We need to bring the CSV data into a DataFrame so that we can
#         inspect, clean, and analyze it easily.
df = pd.read_csv("Tesla - Deaths.csv")

# 1.2 Basic overview of the dataset
# Why? -> Always start with shape, head, and info to understand
#         the structure of the data, number of rows/columns,
#         and data types.
print("Shape of dataset (rows, columns):", df.shape)
print("\n--- First 5 rows ---\n", df.head())
print("\n--- Data types & Non-null counts ---")
print(df.info())

# 1.3 Checking for missing values
# Why? -> Missing values can affect analysis and models.
#         This tells us how much cleaning will be needed.
print("\n--- Missing values per column ---\n", df.isnull().sum())

# 1.4 Checking for duplicate rows
# Why? -> Duplicate accident entries will bias results (e.g.,
#         overcounting deaths). Removing them ensures data quality.
duplicates = df.duplicated().sum()
print(f"\nNumber of duplicate rows: {duplicates}")

# If duplicates exist, remove them
if duplicates > 0:
    df = df.drop_duplicates()
    print(f"Duplicates removed. New shape: {df.shape}")

# 1.5 Identifying irrelevant columns
# Why? -> Not all columns are needed for analysis. For example,
#         long text notes, detailed deceased names, or sources may
#         not help in quantitative analysis. We can drop them later
#         after EDA.
print("\n--- Columns in dataset ---\n", df.columns.tolist())

# (Optional) Save cleaned version for further analysis
# Why? -> Having a clean base file avoids repeating cleaning steps.
df.to_csv("Tesla_Deaths_Cleaned.csv", index=False)
print("\nCleaned dataset saved as Tesla_Deaths_Cleaned.csv")
```

```
Shape of dataset (rows, columns): (307, 24)

--- First 5 rows ---
    Case #    Year        Date  Country   State   \
0   294.0  2022.0   1/17/2023      USA       CA
1   293.0  2022.0    1/7/2023   Canada        -
2   292.0  2022.0    1/7/2023      USA       WA
3   291.0  2022.0  12/22/2022      USA       GA
4   290.0  2022.0  12/19/2022   Canada        -

                         Description    Deaths   Tesla driver   \
0    Tesla crashes into back of semi        1.0             1
1                     Tesla crashes        1.0             1
2  Tesla hits pole, catches on fire        1.0             -
3           Tesla crashes and burns        1.0             1
4     Tesla crashes into storefront        1.0             -

   Tesla occupant   Other vehicle   ...  Verified Tesla Autopilot Deaths
\
0              -               -   ...                                 -
1              -               -   ...                                 -
2              1               -   ...                                 -
3              -               -   ...                                 -
4              -               -   ...                                 -

   Verified Tesla Autopilot Deaths + All Deaths Reported to NHTSA SGO   \
0                                                  -
1                                                  -
2                                                  -
3                                                  -
4                                                  -

                                        Unnamed: 16  \
0   https://web.archive.org/web/20221222203930/ht...
1   https://web.archive.org/web/20221222203930/ht...
2   https://web.archive.org/web/20221222203930/ht...
3   https://web.archive.org/web/20221222203930/ht...
4   https://web.archive.org/web/20221223203725/ht...

                                        Unnamed: 17  \
0   https://web.archive.org/web/20221222203930/ht...
1   https://web.archive.org/web/20221222203930/ht...
2   https://web.archive.org/web/20221222203930/ht...
3   https://web.archive.org/web/20221222203930/ht...
4   https://web.archive.org/web/20221223203725/ht...

                                           Source   Note   \
0   https://web.archive.org/web/20230118162813/ht...    NaN
1   https://web.archive.org/web/20230109041434/ht...    NaN
2   https://web.archive.org/web/20230107232745/ht...    NaN
3   https://web.archive.org/web/20221222203930/ht...    NaN
4   https://web.archive.org/web/20221223203725/ht...    NaN

         Deceased 1   Deceased 2   Deceased 3   Deceased 4
0               NaN          NaN          NaN          NaN
1   Taren Singh Lal          NaN          NaN          NaN
2               NaN          NaN          NaN          NaN
3               NaN          NaN          NaN          NaN
4               NaN          NaN          NaN          NaN
```

```
[5 rows x 24 columns]

--- Data types & Non-null counts ---
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307 entries, 0 to 306
Data columns (total 24 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   Case #
294 non-null    float64
 1   Year
294 non-null    float64
 2   Date
294 non-null    object
 3    Country
294 non-null    object
 4    State
294 non-null    object
 5    Description
295 non-null    object
 6    Deaths
299 non-null    float64
 7    Tesla driver
294 non-null    object
 8    Tesla occupant
290 non-null    object
 9    Other vehicle
295 non-null    object
 10   Cyclists/ Peds
296 non-null    object
 11   TSLA+cycl / peds
297 non-null    object
 12   Model
296 non-null    object
 13   Autopilot claimed
281 non-null    object
 14   Verified Tesla Autopilot Deaths
297 non-null    object
 15   Verified Tesla Autopilot Deaths + All Deaths Reported to NHTSA SGO
296 non-null    object
 16  Unnamed: 16
292 non-null    object
 17  Unnamed: 17
289 non-null    object
 18   Source
297 non-null    object
 19   Note
9 non-null      object
 20   Deceased 1
87 non-null     object
 21   Deceased 2
17 non-null     object
 22   Deceased 3
4 non-null      object
 23   Deceased 4
0 non-null      float64
dtypes: float64(4), object(20)
memory usage: 57.7+ KB
```

```
None

--- Missing values per column ---
 Case #
13
Year                                                                  1
3
Date                                                                  1
3
 Country                                                              1
3
 State                                                                1
3
 Description                                                          1
2
 Deaths
8
 Tesla driver                                                         1
3
 Tesla occupant                                                       1
7
 Other vehicle                                                        1
2
 Cyclists/ Peds                                                       1
1
 TSLA+cycl / peds                                                     1
0
 Model                                                                1
1
 Autopilot claimed                                                    2
6
 Verified Tesla Autopilot Deaths                                      1
0
 Verified Tesla Autopilot Deaths + All Deaths Reported to NHTSA SGO   1
1
Unnamed: 16                                                           1
5
Unnamed: 17                                                           1
8
 Source                                                               1
0
 Note                                                                29
8
 Deceased 1                                                          22
0
 Deceased 2                                                          29
0
 Deceased 3                                                          30
3
 Deceased 4                                                          30
7
dtype: int64

Number of duplicate rows: 4
Duplicates removed. New shape: (303, 24)

--- Columns in dataset ---
 ['Case #', 'Year', 'Date', ' Country ', ' State ', ' Description ', ' Dea
ths ', ' Tesla driver ', ' Tesla occupant ', ' Other vehicle ', ' Cyclist
s/ Peds ', ' TSLA+cycl / peds ', ' Model ', ' Autopilot claimed ', ' Verif
```

```
ied Tesla Autopilot Deaths ', ' Verified Tesla Autopilot Deaths + All Deat
hs Reported to NHTSA SGO ', 'Unnamed: 16', 'Unnamed: 17', ' Source ', ' No
te ', ' Deceased 1 ', ' Deceased 2 ', ' Deceased 3 ', ' Deceased 4 ']

Cleaned dataset saved as Tesla_Deaths_Cleaned.csv
```

Step 1 – Data Inspection & Cleaning

1.1 Load the Dataset

- Load the `Tesla-Deaths.csv` file using Pandas.
- Understand the structure of the dataset before analysis.

1.2 Basic Overview

- Check the shape of the dataset (rows × columns).
- Display first 5 rows to see how the data looks.
- Use `.info()` to inspect column names, data types, and non-null values.

1.3 Missing Values

- Check for missing values in each column.
- Identify which columns need cleaning or imputation.

1.4 Duplicate Records

- Check for duplicate rows using `.duplicated().sum()`.
- Remove duplicates if any are found to avoid biased results.

1.5 Column Relevance

- Identify irrelevant columns (e.g., text notes, deceased names, source info).
- Decide whether to drop them later during EDA.

1.6 Save Cleaned Data

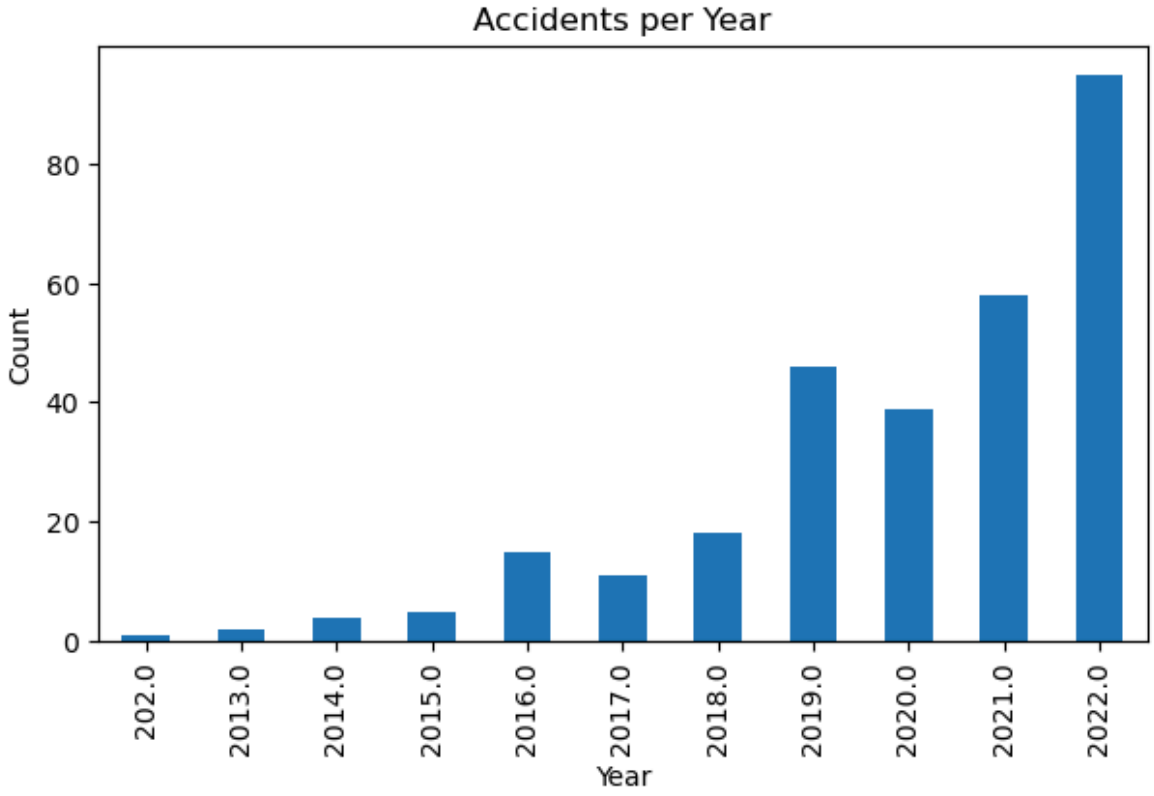
- Save the cleaned dataset as a new CSV (`Tesla_Deaths_Cleaned.csv`).
- Why? → Having a clean base file avoids repeating cleaning steps.

In [10]:
```
# ============================================================
# Step 2 – Exploratory Data Analysis (EDA)
# Why? -> EDA helps us understand trends, patterns, and risk factors
#         in Tesla accidents before doing any deeper analysis.
# ============================================================

# Work on a copy
data = df.copy()

# Fix column names (remove extra spaces for safe access)
data.columns = data.columns.str.strip()
```

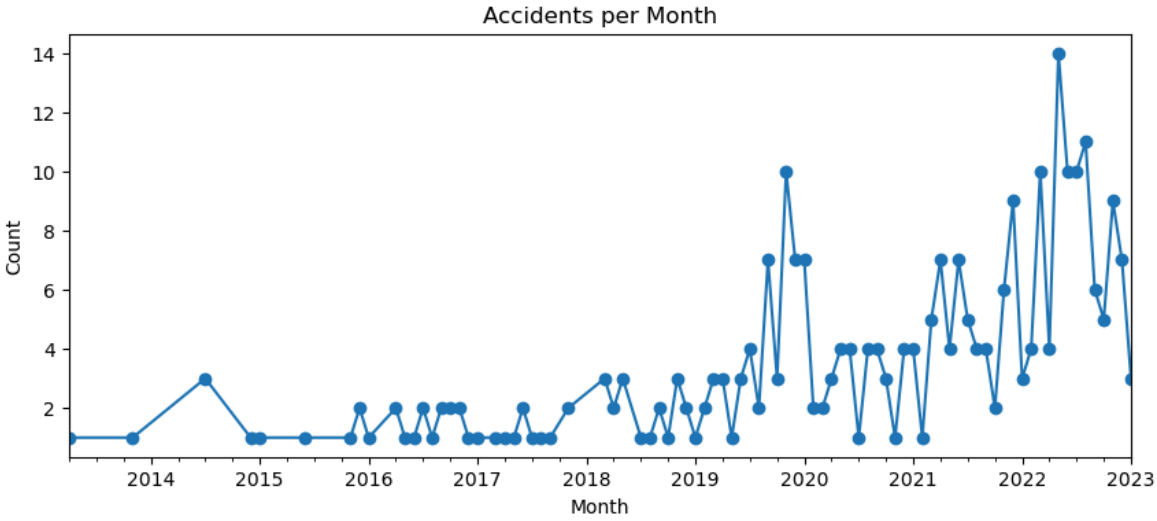
```
# Ensure proper datetime format for 'Date'
data['Date'] = pd.to_datetime(data['Date'], errors='coerce')

# ------------------------------------------------------------
# Step 2.1 – Accidents per Year
# Why? -> Shows long-term trend (are accidents rising or falling each yea
# ------------------------------------------------------------
accidents_per_year = data['Year'].value_counts().sort_index()
print(accidents_per_year)

accidents_per_year.plot(kind='bar', figsize=(7,4))
plt.title("Accidents per Year")
plt.xlabel("Year"); plt.ylabel("Count")
plt.show()

# ------------------------------------------------------------
# Step 2.2 – Accidents per Month
# Why? -> Helps identify seasonal spikes or sudden jumps in accidents.
# ------------------------------------------------------------
accidents_per_month = data['Date'].dt.to_period("M").value_counts().sort_
print(accidents_per_month.head())

accidents_per_month.plot(kind='line', marker='o', figsize=(10,4))
plt.title("Accidents per Month")
plt.xlabel("Month"); plt.ylabel("Count")
plt.show()

# ------------------------------------------------------------
# Step 2.3 – Accidents by Country
# Why? -> To see which countries report the most Tesla accidents.
# ------------------------------------------------------------
if 'Country' in data.columns:
    country_counts = data['Country'].value_counts()
    print(country_counts)

    country_counts.plot(kind='bar', figsize=(7,4))
    plt.title("Accidents by Country")
    plt.xlabel("Country"); plt.ylabel("Count")
    plt.show()

# ------------------------------------------------------------
# Step 2.4 – Deaths per Accident
# Why? -> Measures severity (how many deaths usually occur per accident).
# ------------------------------------------------------------
data['Deaths'] = pd.to_numeric(data['Deaths'], errors='coerce')
data['Deaths'].plot(kind='hist', bins=10, figsize=(6,4))
plt.title("Distribution of Deaths per Accident")
plt.xlabel("Deaths"); plt.ylabel("Frequency")
plt.show()

# ------------------------------------------------------------
# Step 2.5 – Tesla Driver Deaths
# Why? -> To check how often Tesla drivers themselves die in accidents.
# ------------------------------------------------------------
if 'Tesla driver' in data.columns:
    driver_counts = data['Tesla driver'].value_counts(dropna=False)
    print(driver_counts)

    driver_counts.plot(kind='bar', figsize=(5,3))
    plt.title("Tesla Driver Deaths (0 = No, 1 = Yes)")
```

```
    plt.xlabel("Driver Death"); plt.ylabel("Count")
    plt.show()

# ------------------------------------------------------------
# Step 2.6 – Accidents by Tesla Model
# Why? -> Different Tesla models may have different accident frequencies.
# ------------------------------------------------------------
if 'Model' in data.columns:
    model_counts = data['Model'].value_counts()
    print(model_counts)

    model_counts.plot(kind='bar', figsize=(7,4))
    plt.title("Accidents by Tesla Model")
    plt.xlabel("Model"); plt.ylabel("Count")
    plt.show()
```

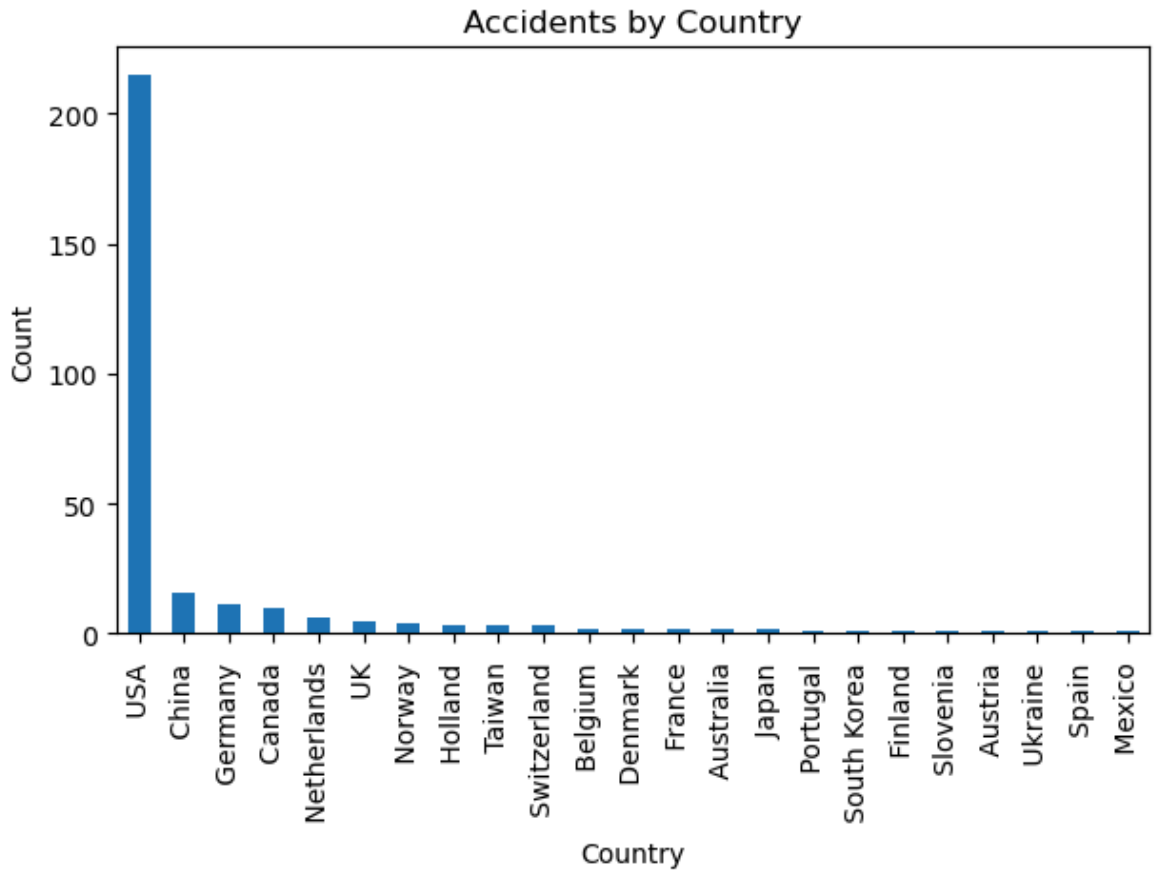
```
Year
202.0      1
2013.0     2
2014.0     4
2015.0     5
2016.0    15
2017.0    11
2018.0    18
2019.0    46
2020.0    39
2021.0    58
2022.0    95
Name: count, dtype: int64
```

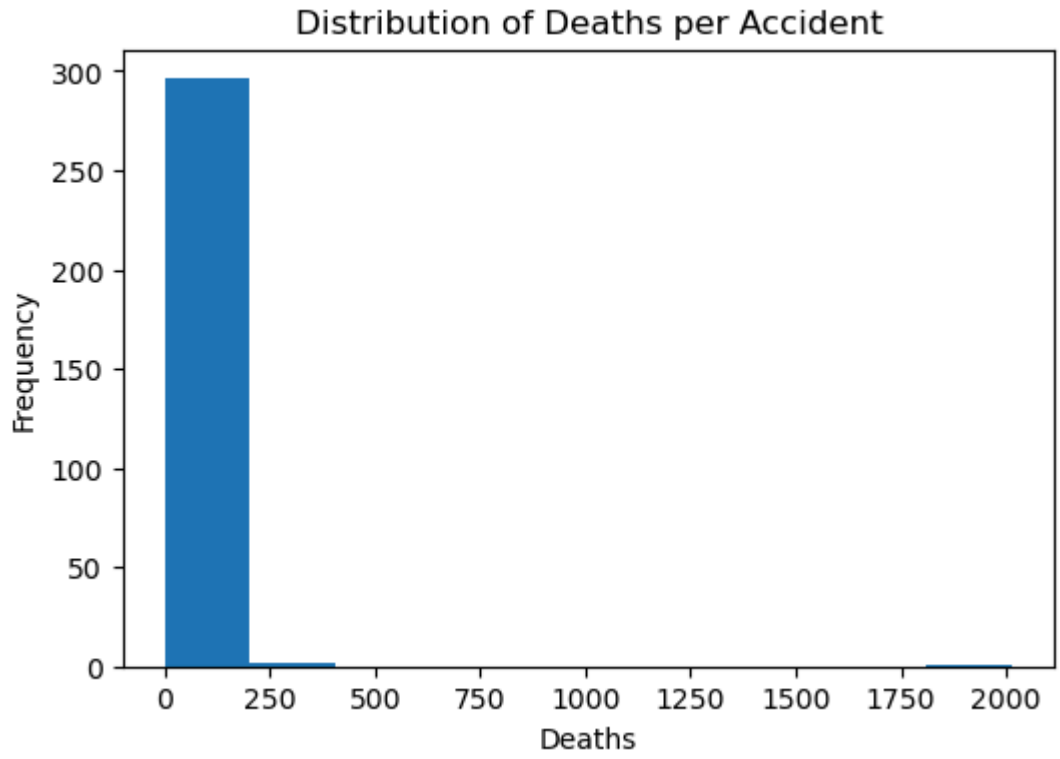



```
Date
2013-04    1
2013-11    1
2014-07    3
2014-12    1
2015-01    1
Freq: M, Name: count, dtype: int64
```

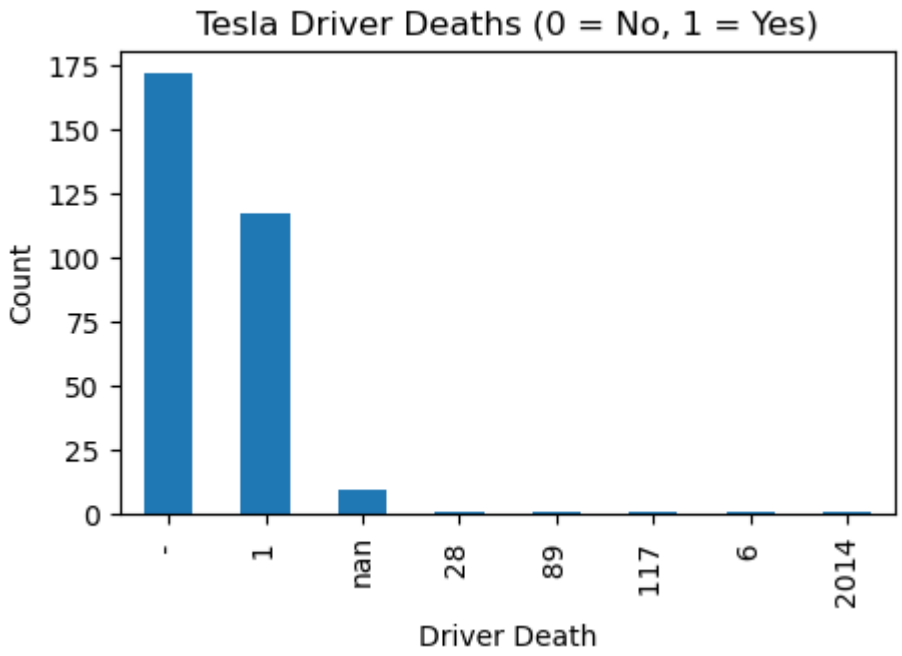



```
Country
USA             215
China            16
Germany          11
Canada           10
Netherlands       6
UK                5
Norway            4
Holland           3
Taiwan            3
Switzerland       3
Belgium           2
Denmark           2
France            2
Australia         2
Japan             2
Portugal          1
South Korea       1
Finland           1
Slovenia          1
Austria           1
Ukraine           1
Spain             1
Mexico            1
Name: count, dtype: int64
```

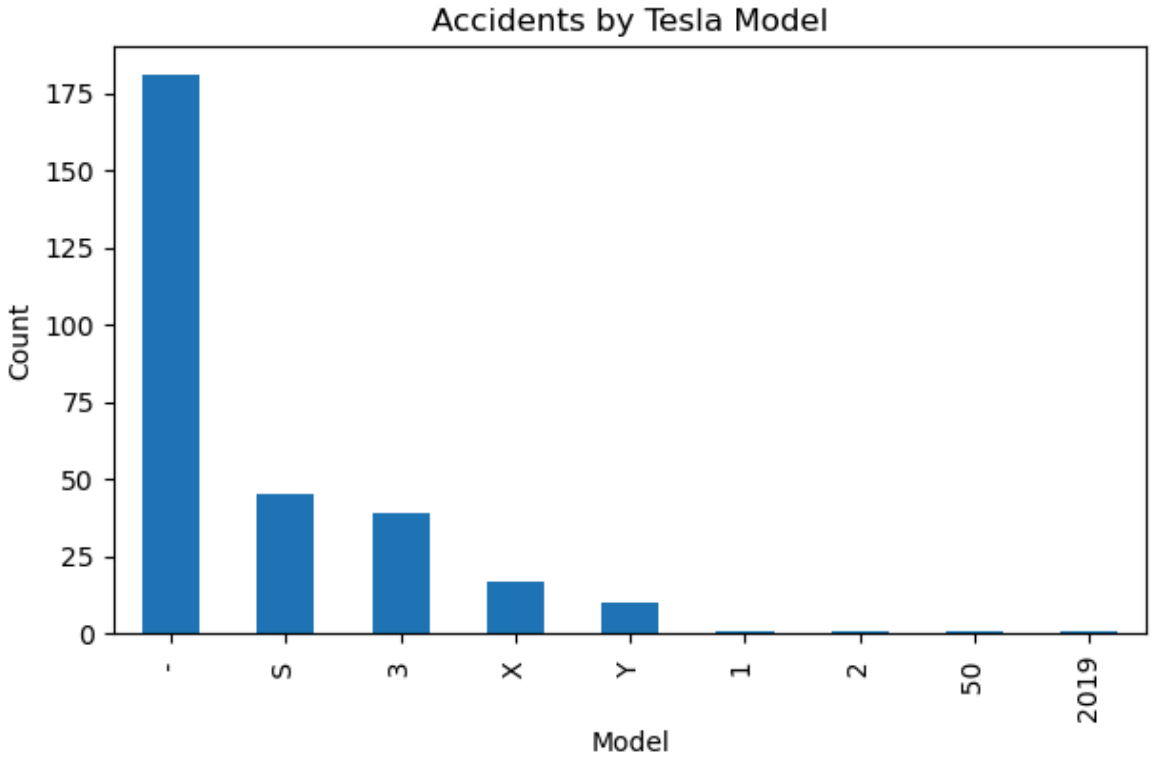





```
Tesla driver
 -        172
  1       117
NaN         9
  28        1
  89        1
  117       1
  6         1
2014        1
Name: count, dtype: int64
```



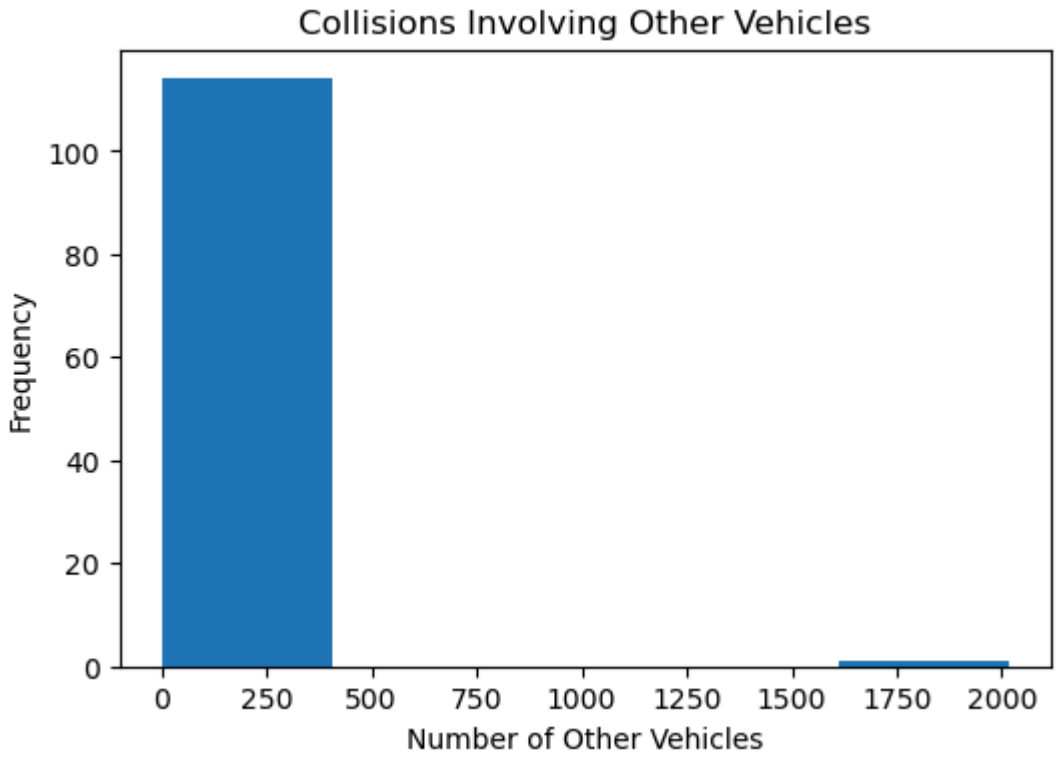

```
Model
 –       181
 S        45
  3       39
 X        17
 Y        10
  1        1
  2        1
  50       1
2019       1
Name: count, dtype: int64
```



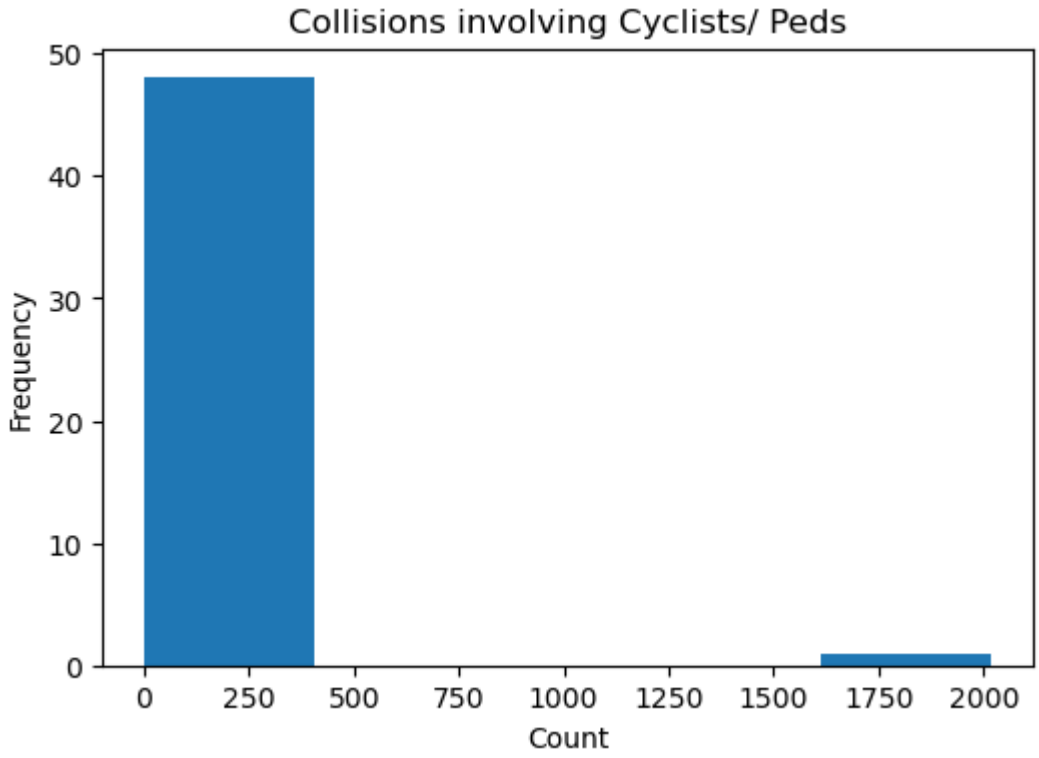
In [13]:
```
# Step 2.7 – Collisions with Other Vehicles
# Why? -> To see how often Tesla crashes involve another vehicle
if 'Other vehicle' in data.columns:
    data['Other vehicle'] = pd.to_numeric(data['Other vehicle'], errors='
    print(data['Other vehicle'].value_counts())
```

```
    data['Other vehicle'].plot(kind='hist', bins=5, figsize=(6,4))
    plt.title("Collisions Involving Other Vehicles")
    plt.xlabel("Number of Other Vehicles"); plt.ylabel("Frequency")
    plt.show()

# Step 2.8 – Collisions with Cyclists/Pedestrians
# Why? -> To check accidents involving cyclists/pedestrians
for col in ['Cyclists/ Peds', 'TSLA+cycl / peds']:
    if col in data.columns:
        data[col] = pd.to_numeric(data[col], errors='coerce')
        print(data[col].value_counts())
        data[col].plot(kind='hist', bins=5, figsize=(6,4))
        plt.title(f"Collisions involving {col}")
        plt.xlabel("Count"); plt.ylabel("Frequency")
        plt.show()
```

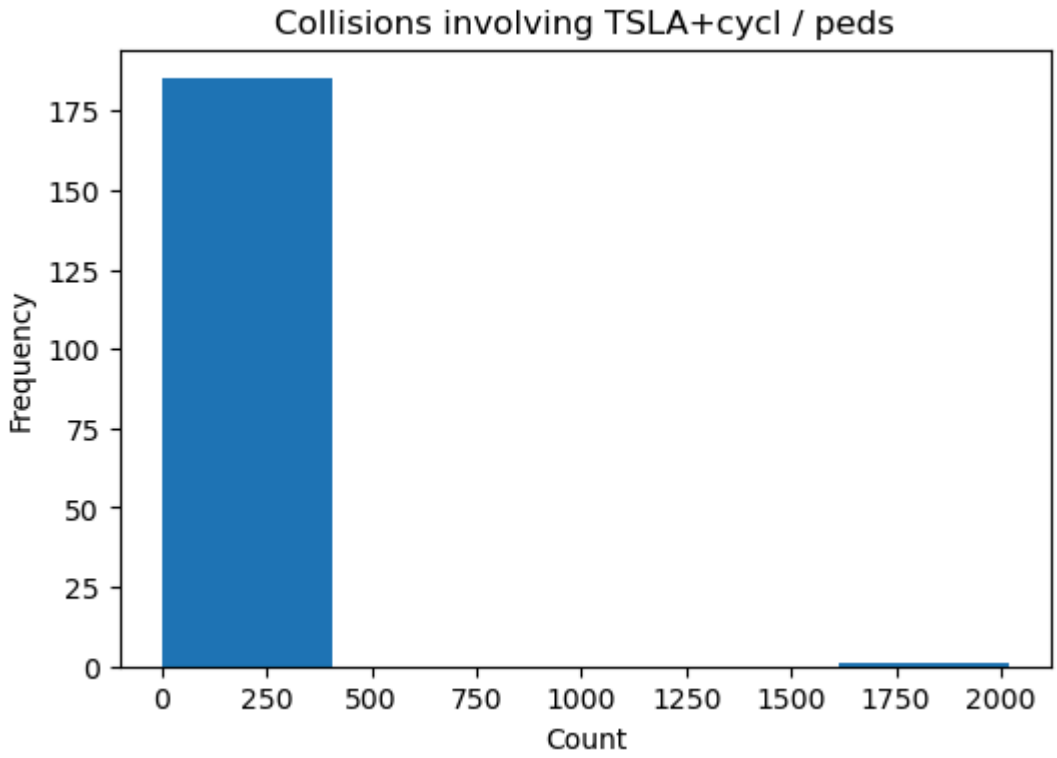
```
Other vehicle
1.0       95
2.0       11
3.0        3
4.0        1
29.0       1
101.0      1
130.0      1
16.0       1
2016.0     1
Name: count, dtype: int64
```




```
Cyclists/ Peds
1.0       42
2.0        2
20.0       1
26.0       1
46.0       1
11.0       1
2017.0     1
Name: count, dtype: int64
```




```
TSLA+cycl / peds
1.0       157
2.0        20
3.0         3
4.0         1
61.0        1
149.0       1
210.0       1
21.0        1
2018.0      1
Name: count, dtype: int64
```



Step 2 – Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) helps us understand the dataset before modeling.
We will answer a few simple but important questions with plots.

2.1 Accidents per Year

Why? → To check if accidents are increasing or decreasing year by year.
This shows the long-term safety trend for Tesla vehicles.

2.2 Accidents per Month

Why? → To see if accidents follow a seasonal pattern or sudden spikes.
This helps identify if certain months are riskier.

2.3 Accidents by Country

Why? → To identify which countries report the most Tesla accidents.
This highlights regions where Tesla usage or risk is higher.

2.4 Deaths per Accident

Why? → Not every accident is fatal.
This shows the severity of accidents by looking at how many deaths happen per crash.

2.5 Tesla Driver Deaths

Why? → To measure how often the Tesla driver himself/herself dies in an accident.
This tells us about the driver's risk compared to passengers or others.

2.6 Accidents by Tesla Model

Why? → Different Tesla models (S, 3, X, Y) may have different accident frequencies.
This helps see which model appears most in accident records.

2.7 Collisions with Other Vehicles

Why? → To check how often Tesla accidents involve other vehicles.
This helps us understand whether most crashes are single-car events or multi-vehicle collisions.

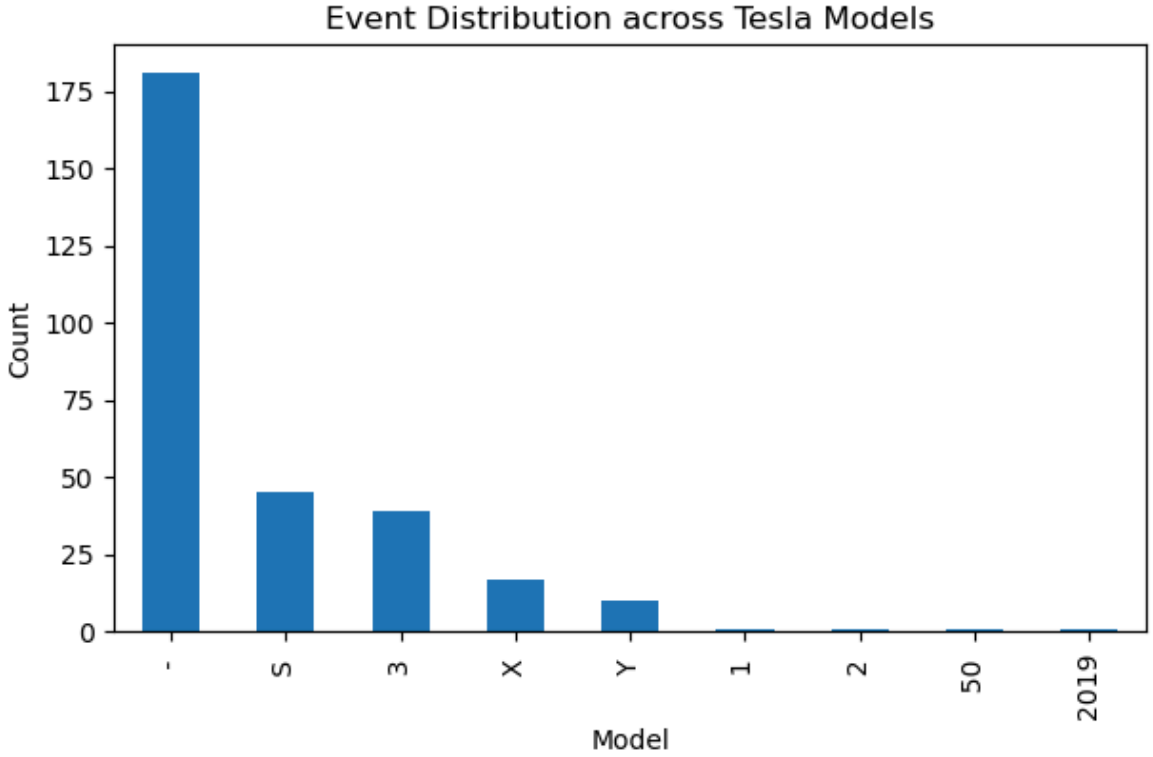
2.8 Collisions with Cyclists / Pedestrians

Why? → To analyze accidents involving vulnerable road users (cyclists and pedestrians).
This shows how many incidents pose risks to people outside the car.

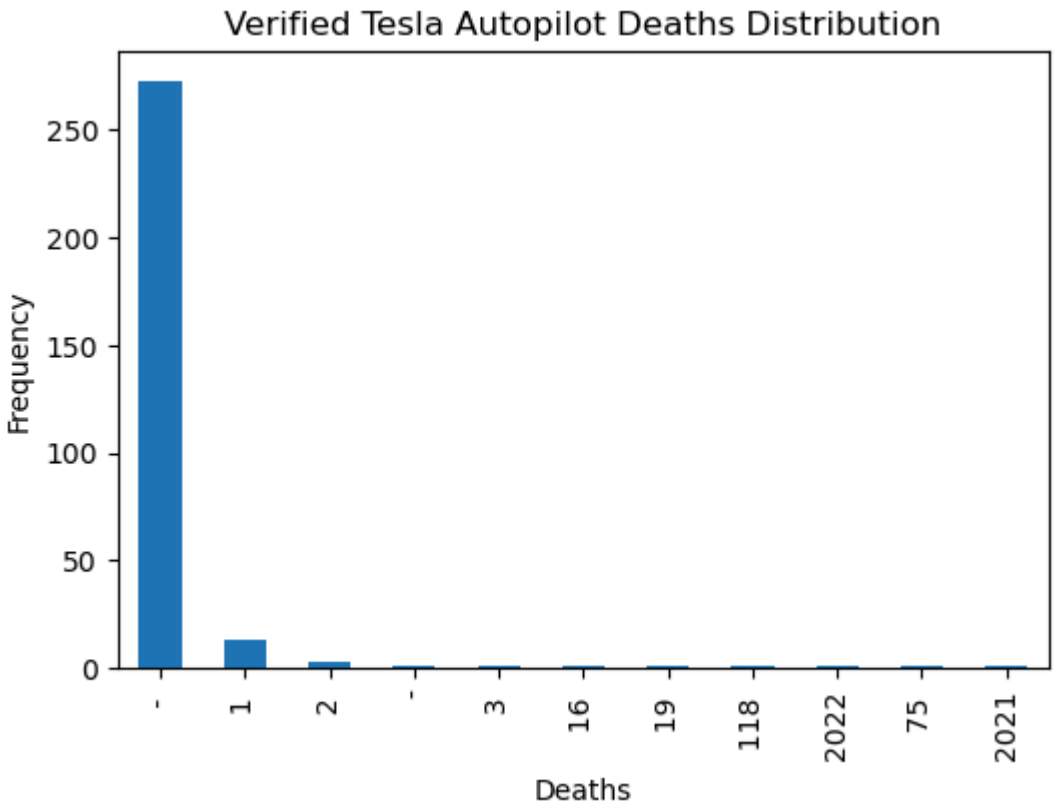
In [18]:
```
# ============================================================
# Step 3 — Model & Autopilot Analysis
# Why? -> To analyze Tesla accidents with respect to models and
#         Autopilot usage. This shows whether certain models or
#         autopilot usage contribute more to fatalities.
# ============================================================

# ------------------------------------------------------------
# Step 3.1 — Event Distribution across Tesla Models
# Why? -> To see which Tesla models (S, 3, X, Y) are most involved
#         in accidents. Accident counts often reflect popularity.
# ------------------------------------------------------------
if 'Model' in data.columns:
    model_counts = data['Model'].value_counts()
    print(model_counts)

    model_counts.plot(kind='bar', figsize=(7,4))
    plt.title("Event Distribution across Tesla Models")
    plt.xlabel("Model"); plt.ylabel("Count")
    plt.show()

# ------------------------------------------------------------
# Step 3.2 — Verified Tesla Autopilot Deaths Distribution
# Why? -> To check how many verified deaths were directly associated
#         with Autopilot usage. This gives a clearer picture of risk.
# ------------------------------------------------------------
if 'Verified Tesla Autopilot Deaths' in data.columns:
    autopilot_deaths = data['Verified Tesla Autopilot Deaths'].value_coun
    print(autopilot_deaths)

    autopilot_deaths.plot(kind='bar', figsize=(6,4))
    plt.title("Verified Tesla Autopilot Deaths Distribution")
    plt.xlabel("Deaths"); plt.ylabel("Frequency")
    plt.show()
# ------------------------------------------------------------
# Step 3.3 — Verified Autopilot Deaths vs All Reported Deaths
# Why? -> To compare officially verified autopilot deaths against all
#         reported deaths (NHTSA). This highlights under/over-reporting.
# ------------------------------------------------------------
if 'Verified Tesla Autopilot Deaths' in data.columns and 'All Deaths Repo

    compare_df = pd.DataFrame({
        'Verified Autopilot Deaths': [data['Verified Tesla Autopilot Deat
        'All Reported Deaths (NHTSA)': [data['All Deaths Reported to NHTS
    })
```

```
    print(compare_df)

    compare_df.plot(kind='bar', figsize=(6,4))
    plt.title("Verified Autopilot Deaths vs All Reported Deaths (NHTSA)")
    plt.ylabel("Count")
    plt.show()
```

```
Model
 -       181
 S        45
  3       39
 X        17
 Y        10
  1        1
  2        1
  50       1
2019       1
Name: count, dtype: int64
```




```
Verified Tesla Autopilot Deaths
 -        273
  1        13
  2         3
-           1
  3         1
  16        1
  19        1
  118       1
2022        1
  75        1
2021        1
Name: count, dtype: int64
```



Step 3 – Model & Autopilot Analysis

In this step, we analyze accidents with respect to Tesla models and Autopilot usage.

3.1 Event Distribution across Tesla Models

Why? → To see which Tesla models (S, 3, X, Y) are most commonly involved in accidents.
This helps check if accident frequency matches model popularity.

3.2 Verified Tesla Autopilot Deaths Distribution

Why? → To understand how many verified deaths were officially linked to Autopilot usage.
This shows the direct risk associated with Autopilot.

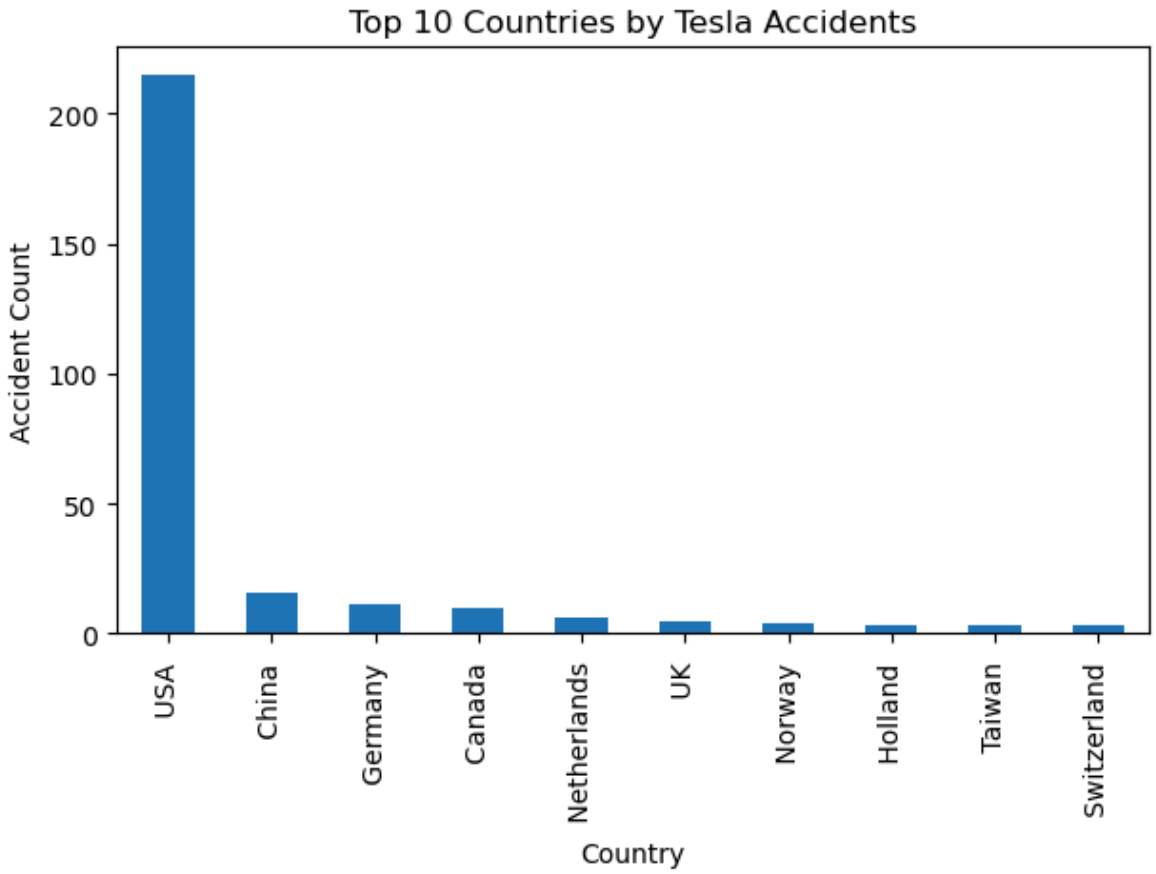
3.3 Verified Autopilot Deaths vs All Reported Deaths (NHTSA)

Why? → To compare the officially verified Autopilot deaths with all deaths reported to NHTSA.

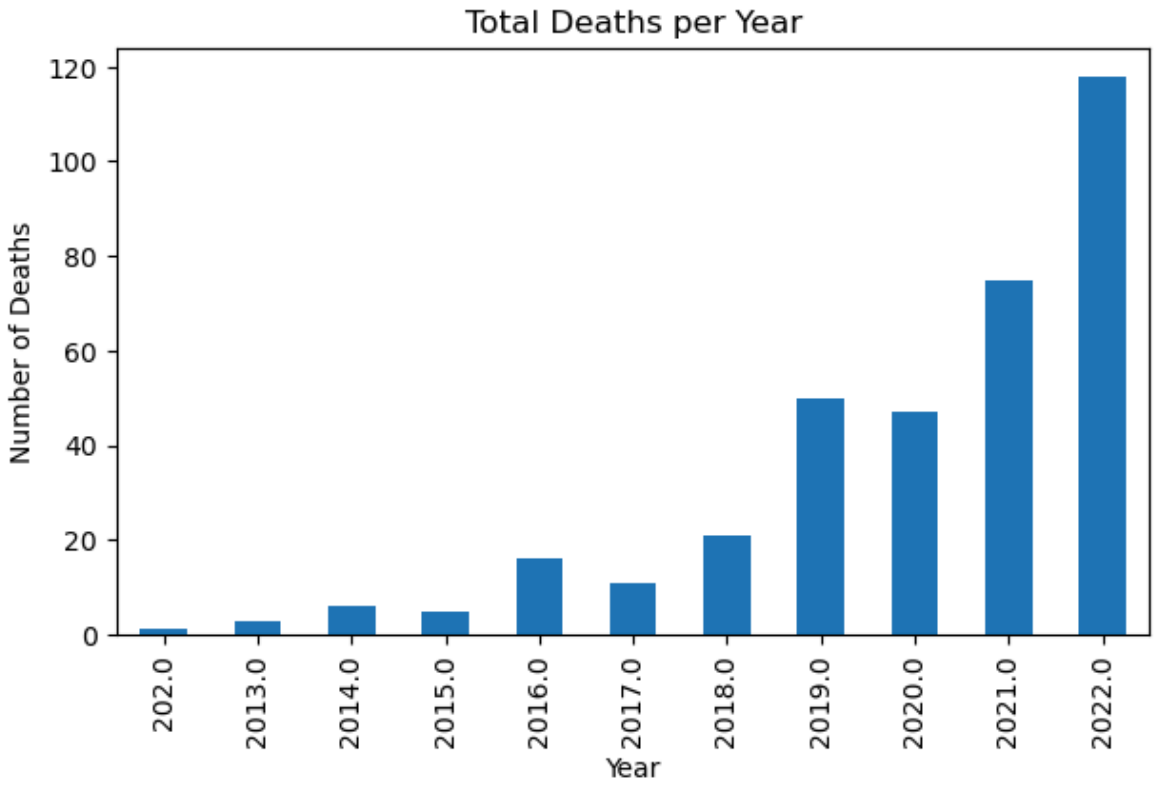
This highlights the difference between confirmed Autopilot fatalities and overall accident reports.

In [21]:
```
# Step 4 – Visualisation & Insights
# Why? -> To summarize the findings from EDA and Model/Autopilot analysis
#         with simple, clear visuals.

# 4.1 Country-wise Top Accidents
print("Top 10 Countries by Tesla Accidents:")
country_counts = data['Country'].value_counts().head(10)
print(country_counts)

country_counts.plot(kind='bar', figsize=(7,4))
plt.title("Top 10 Countries by Tesla Accidents")
plt.xlabel("Country"); plt.ylabel("Accident Count")
plt.show()

# 4.2 Deaths Trend Over Time (Year-wise)
print("\nTotal Deaths per Year:")
if 'Year' in data.columns and 'Deaths' in data.columns:
    deaths_per_year = data.groupby('Year')['Deaths'].sum()
    print(deaths_per_year)

    deaths_per_year.plot(kind='bar', figsize=(7,4))
    plt.title("Total Deaths per Year")
    plt.xlabel("Year"); plt.ylabel("Number of Deaths")
    plt.show()
```

```
Top 10 Countries by Tesla Accidents:
Country
USA             215
China            16
Germany          11
Canada           10
Netherlands       6
UK                5
Norway            4
Holland           3
Taiwan            3
Switzerland       3
Name: count, dtype: int64
```




```
Total Deaths per Year:
Year
202.0       1.0
2013.0      3.0
2014.0      6.0
2015.0      5.0
2016.0     16.0
2017.0     11.0
2018.0     21.0
2019.0     50.0
2020.0     47.0
2021.0     75.0
2022.0    118.0
Name: Deaths, dtype: float64
```



Step 4 – Insights & Conclusion

Based on the analysis, here are the key insights:

- **Country-wise**: The USA reports the highest number of Tesla accidents, followed by a few other countries (e.g., China, Germany, Canada).
- **Yearly Trend**: Accident counts and deaths increased sharply after 2018, showing Tesla's rising adoption and exposure risk.
- **Model-wise**: Model S and Model 3 appear most frequently in accident records.
- **Victim Analysis**: While most crashes involve only 1 fatality, there are also cases with multiple victims.
- **Collision Patterns**: Majority of Tesla accidents involve other vehicles, and a notable number involve cyclists/pedestrians.
- **Autopilot**: Verified Autopilot deaths are significantly fewer than all reported deaths (NHTSA), suggesting not all fatalities are officially attributed to Autopilot.

In [ ]:

Final Summary – Key Insights from Tesla Accident Analysis

Based on the data cleaning, exploratory analysis, and model/autopilot study, here are the main takeaways:

- **Accident Growth Over Time**: Tesla accidents have steadily increased since 2013, with a sharp rise after 2018, reflecting growing Tesla adoption.
- **Country Distribution**: The USA reports the overwhelming majority of accidents, followed by China, Germany, and Canada.
- **Severity of Accidents**: While many accidents involve a single death, there are notable cases with multiple fatalities, showing varying crash severity.
- **Driver vs Passenger Risk**: Tesla drivers themselves account for a significant portion of deaths, indicating high risk for the person in control.
- **Model-wise Trends**: Model S and Model 3 are most frequently reported in accidents, likely due to their higher sales numbers compared to other Tesla models.
- **Collision Types**: Most crashes involve other vehicles, with additional cases involving cyclists/pedestrians, highlighting external road safety risks.
- **Autopilot Analysis**: Verified Autopilot-linked deaths are fewer than total deaths reported to NHTSA, suggesting that not all fatalities are attributed directly to Autopilot.

✅ This concludes **Part-2 (Exploratory Data Analysis + Model & Autopilot Analysis)**.

In [ ]:

In [ ]:

In [ ]: