plot_olds_accidents

In [1]:

```
import matplotlib.pyplot as plt
import numpy as np

from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split

import pandas as pd
```

In [2]:

```
df = pd.read_csv("./dataset/accidents.csv")

df
```

Out[2]:

	x	y
0	493782	164
1	572059	43
2	608827	98
3	626932	101
4	642200	100
5	754844	197
6	783600	134
7	902195	229
8	1048319	83
9	1211537	142
10	1235786	171
11	1274923	194
12	1293953	260
13	1711263	254
14	1808344	411
15	1819046	521
16	1998257	395

	x	y
17	2233169	296
18	2673400	704
19	2688418	461
20	2844658	900
21	2926324	390
22	3405565	291
23	3421399	456
24	3450654	774
25	4012012	1046
26	4041769	964
27	4301261	665
28	4447100	1154
29	4468976	904
30	4919479	567
31	5130632	1150
32	5296486	643
33	5363675	792
34	5595211	1130
35	5689283	1288
36	5894121	563
37	6080485	947
38	6349097	476
39	7078515	925
40	8049313	1557
41	8186453	1634
42	8414350	731
43	9938444	1159
44	11353140	1286
45	12281054	1490
46	12419293	1356
47	15982378	3244
48	18976457	1493
49	20851820	3583
50	33871648	4120

In [3]:

```
x = np.array(df.get('x'))
y = np.array(df.get('y'))
```

In [4]:

```
# split 80% 20%
X_train, X_test, y_train, y_test = train_test_split(
    x, y, test_size=0.2, random_state=42
```

```
)

print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

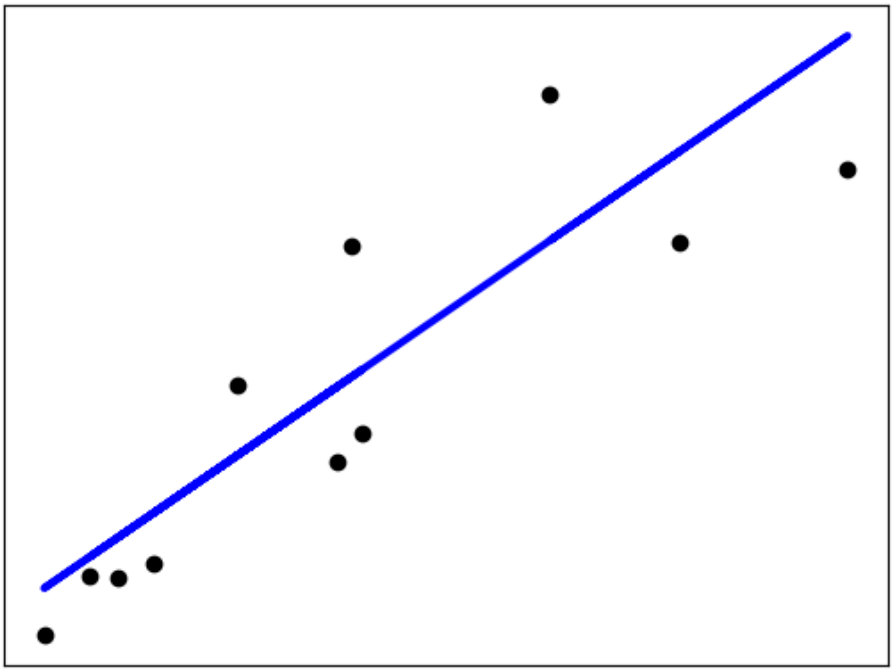
```
(40,)
(11,)
(40,)
(11,)
```

In [5]:

```
# Reshape X_train to a 2D array
X_train_reshaped = X_train.reshape(-1, 1)

print(X_train_reshaped.shape)


# Create linear regression object
regr = linear_model.LinearRegression()

# Train the model using the training sets
regr.fit(X_train_reshaped, y_train)

# Make predictions using the testing set
y_pred = regr.predict(X_test.reshape(-1, 1))

# The coefficients
print("Coefficients: \n", regr.coef_)
# The mean squared error
print("Mean squared error: %.2f" % mean_squared_error(y_test, y_pred))
# The coefficient of determination: 1 is perfect prediction
print("Coefficient of determination: %.2f" % r2_score(y_test, y_pred))

# Plot outputs
plt.scatter(X_test, y_test, color="black")
plt.plot(X_test, y_pred, color="blue", linewidth=3)

plt.xticks(())
plt.yticks(())

plt.show()
```

```
(40, 1)
Coefficients:
 [0.0001263]
```

```
Mean squared error: 56835.65
Coefficient of determination: 0.75
```



In [ ]: