

```
1 import nltk
2 from nltk.corpus import stopwords
3 from nltk.tokenize import word_tokenize
4 from nltk.probability import FreqDist
5 import matplotlib.pyplot as plt
6 from wordcloud import WordCloud
7 from nltk.tokenize import sent_tokenize
8
9 # Ensure NLTK resources are downloaded
10 # nltk.download('punkt')
11 # nltk.download('stopwords')
12
13 # Load the dataset
14 with open('8_BikeInjury.txt', 'r') as file:
15     data = file.read()
16
17 # Tokenize the sentences into words
18 tokens = word_tokenize(data)
19
20 # Convert to lowercase and remove non-alphabetic
    tokens
21 words = [word.lower() for word in tokens if word.
    isalpha()]
22
23 # Remove stopwords
24 stop_words = set(stopwords.words('english'))
25 filtered_words = [word for word in words if word not
    in stop_words]
26
27 # Compute the frequency distribution of words
28 fdist = FreqDist(filtered_words)
29
30 # Display the 25 most common injury descriptions
31 common_injuries = fdist.most_common(25)
32 for word, frequency in common_injuries:
33     print(f"{word}: {frequency}")
34
35 # Separate the words and frequencies
36 words, frequencies = zip(*common_injuries)
37
38 # Adjust the figure size
```

```
39 plt.figure(figsize=(15,10))
40
41 # Plot the frequencies using horizontal bars
42 bars = plt.barh(words, frequencies, color='#ffd166')
43
44 # Set the title
45 plt.title('Top 25 Biking Injury Descriptions')
46
47 # Annotate each bar with its respective count
48 for bar in bars:
49     width = bar.get_width()
50     plt.text(width + 5, # Increase this value to
        move numbers further to the right
51             bar.get_y() + bar.get_height() / 2,
52             str(int(width)),
53             ha='center',
54             va='center')
55
56 # Invert the y-axis to have the word with the highest
    count on top
57 plt.gca().invert_yaxis()
58 plt.tight_layout()
59 plt.show()
60
61 # WordCloud
62 wordcloud = WordCloud(
63     width=1500,
64     height=800,
65     background_color='white',
66     max_words=50,
67     min_font_size=10,
68     contour_width=3,
69     contour_color='steelblue'
70 ).generate(' '.join(filtered_words))
71
72 plt.figure(figsize=(18,10))
73 plt.imshow(wordcloud, interpolation="bilinear")
74 plt.axis('off')
75 plt.title("Biking Injury Descriptions Word Cloud")
76 plt.show()
77
```

```
78 # # Tokenize the text into sentences
79 # tokenized_sentences = sent_tokenize(data)
80 #
81 # # Display several injury description sentences
82 # for i, sentence in enumerate(tokenized_sentences[:
    35]): # Change 5 to the number of sentences you
    want to display
83 #     print(f"Sentence {i+1}: {sentence}")
84
85
```