```
 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
```

```
plt.figure(figsize=(15,10))

# Plot the frequencies using horizontal bars
bars = plt.barh(words, frequencies, color='#ffd166')

# Set the title
plt.title('Top 25 Biking Injury Descriptions')

# Annotate each bar with its respective count
for bar in bars:
    width = bar.get_width()
    plt.text(width + 5,  # Increase this value to move numbers further to the right
             bar.get_y() + bar.get_height() / 2,
             str(int(width)),
             ha='center',
             va='center')

# Invert the y-axis to have the word with the highest count on top
plt.gca().invert_yaxis()
plt.tight_layout()
plt.show()

# WordCloud
wordcloud = WordCloud(
    width=1500,
    height=800,
    background_color='white',
    max_words=50,
    min_font_size=10,
    contour_width=3,
    contour_color='steelblue'
).generate(' '.join(filtered_words))

plt.figure(figsize=(18,10))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.title("Biking Injury Descriptions Word Cloud")
plt.show()

```

```
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
```