```
 1  import argparse
 2  import nltk
 3  import numpy
 4  import pylab
 5  import random
 6
 7  def zipf_law_plot(input_text, title):
 8      fdist = nltk.FreqDist(input_text)
 9      pylab.plot(range(fdist.B()), fdist.values())
10      pylab.xscale('log')
11      pylab.yscale('log')
12      pylab.title(title)
13      pylab.show()
14
15  def main():
16      parser = argparse.ArgumentParser()
17      parser.add_argument('-r', '--random', action='store_true',
    help="Generate random text")
18      parser.add_argument('-i', '--input', type=str, help="Zipf plot of
    input text")
19      args = parser.parse_args()
20
21      if not args.random and not args.input:
22          print("Error: must pass in an argument")
23          return
24
25      if args.random:
26          target_length = 808080
27
28          random_string = ""
29          while len(random_string) < target_length:
30              random_string += random.choice('abcdefg ')
31
32          # Tokenize random string and generate the Zipf plot
33          random_text = random_string.split()
34          print("Plot of Zipf's Law for random text")
35          zipf_law_plot(random_text, "Zipf Plot for Random Text")
36
37      if args.input:
38          zipf_law_plot(random_text, "Zipf Plot for %s Text" %
    args.input)
39
40
41  if __name__ == "__main__":
42      main()
```