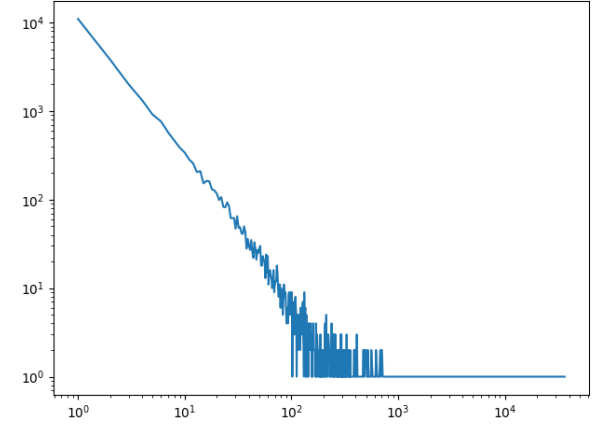
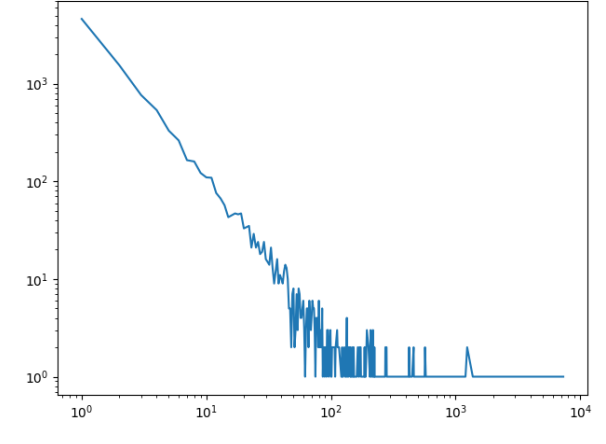
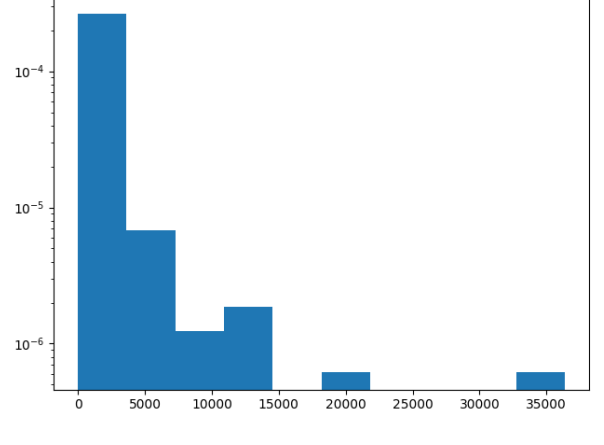
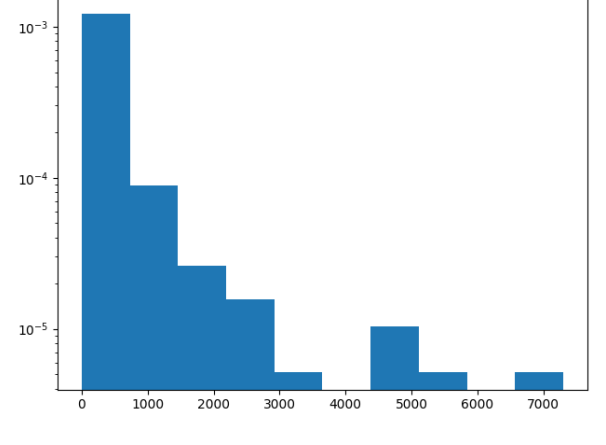
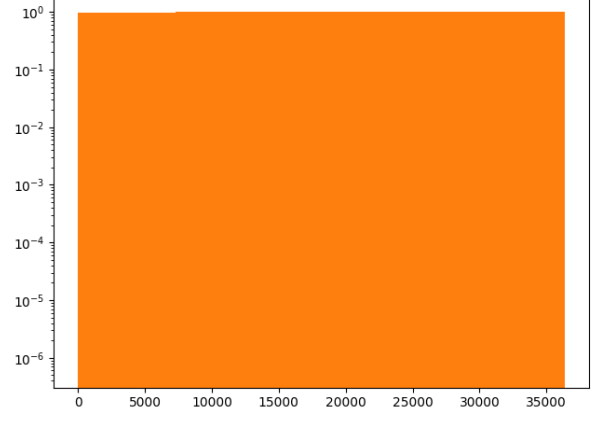
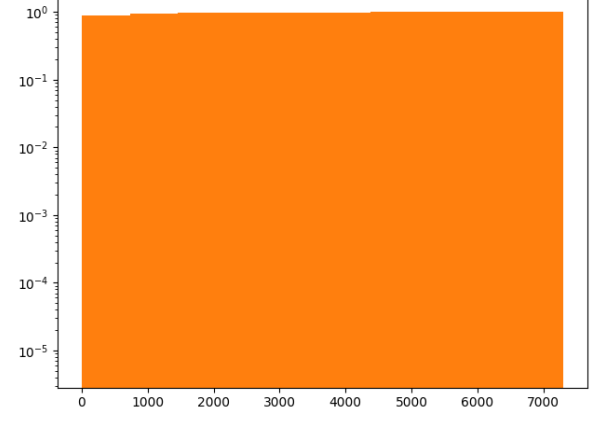


Les Miserables	Dracula
<p>4a</p> 	<p>4a</p> 
4b included as a separate .pdf	
<p>4c</p> 	<p>4c</p> 
<p>4d</p> 	<p>4d</p> 
<p>4e</p> <p>lm MLE = 1.9008609856258776</p>	<p>4e</p> <p>dr MLE = 2.021954879734034</p>

Code:

```
###
import collections
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import colors
from matplotlib.ticker import PercentFormatter
import scipy.stats
from math import e
from numpy import log as ln
from scipy import stats

# THIS CODE IS MY OWN WORK, IT WAS WRITTEN WITHOUT CONSULTING A TUTOR OR
# CODE WRITTEN BY OTHER STUDENTS - Miles Robertson

a_file = open("les-miserables.txt")
file_contents = a_file.read()
contents_split = file_contents.splitlines()
a_file.close()
lm = contents_split

a_file = open("dracula.txt")
file_contents = a_file.read()
contents_split = file_contents.splitlines()
a_file.close()
dr = contents_split

# initializing the list; specifying which .txt file to run
random_list = dr
set_filename = 'dr'
# using Counter to find frequency of elements
frequency = collections.Counter(random_list)
# converting to dictionary
frequency = dict(frequency)
n = len(frequency)

freq_freq = {}
for word in frequency:
    if frequency[word] in freq_freq:
        freq_freq[frequency[word]] += 1
    else:
        freq_freq[frequency[word]] = 1
freq_freq_sorted = {k: v for k, v in sorted(freq_freq.items(), key=lambda
item: item[0])}
```

```

# print(len(freq_freq_sorted))

x = freq_freq_sorted.keys()
y = freq_freq_sorted.values()

# 4a
plt.loglog(x, y) # plt.loglog(x, y[, linewidth, color, basex, basey, ...])
plt.tight_layout()
plt.savefig(set_filename + '-loglog.png')

plt.clf() # clears graph

# 4e
mle_denom_sum = 0
for xi in frequency.values():
    mle_denom_sum += ln(xi)
MLE = 1 + (n/mle_denom_sum)
print(set_filename + ' MLE = ' + str(MLE))

# 4e plug in alpha?
power_law_fit_likelihood = 0
for xi in frequency.values():
    power_law_fit_likelihood += ln((MLE - 1)) - (MLE * ln(xi))
# print(set_filename + ' Likelihood = ' + str(power_law_fit_likelihood))

# 4c
plt.hist(x, bins=None, range=None, density=True, weights=None,
cumulative=False, bottom=None, histtype='bar', align='mid',
orientation='vertical', rwidth=None, log=True, color=None, label=None,
stacked=False, data=None)
# res = stats.linregress(x, y)
# plt.plot(x, res.intercept + res.slope*x, 'r', label='fitted line')
plt.savefig(set_filename + '-hist-pdf.png')

# 4d
plt.hist(x, bins=None, range=None, density=True, weights=None,
cumulative=True, bottom=None, histtype='bar', align='mid',
orientation='vertical', rwidth=None, log=True, color=None, label=None,
stacked=False, data=None)
plt.savefig(set_filename + '-hist-cdf.png')

```