In [1]:

```
import pandas as pd
import ipaddress
import matplotlib.pyplot as plt
```

Task 1

In [2]:

```
tranco_data = pd.read_csv('tranco_Y5G4G.csv')
tranco_data.columns = ['num', 'url']
tranco_data = tranco_data.loc[:, ['url']]
tranco_data = tranco_data[:100]
print(tranco_data)
```

```
               url
0     facebook.com
1     a-msedge.net
2      youtube.com
3    microsoft.com
4    amazonaws.com
..             ...
95        ebay.com
96   google.com.hk
97     nytimes.com
98      fandom.com
99     dropbox.com

[100 rows x 1 columns]
```

In [3]:

```
mm_data = pd.read_csv('majestic_million.csv')
mm_data = mm_data[:100]
mm_data = mm_data.loc[:, ['Domain']]
mm_data = mm_data.rename(columns={'Domain': 'url'})
print(mm_data)
```

```
                      url
0              google.com
1            facebook.com
2             youtube.com
3             twitter.com
4           instagram.com
..                    ...
95   youtube-nocookie.com
96              nginx.com
97               imdb.com
98          bloomberg.com
99            harvard.edu

[100 rows x 1 columns]
```

In [4]:

```
phish_data = pd.read_csv('./PhishTank-online-banking-phishing-urls-final.csv')
phish_data = phish_data.loc[:, ['Indicator']]
phish_data = phish_data[:100]
remove = ['http://', 'ftp://','www.']
phish_data = phish_data.rename(columns={'Indicator': 'url'})
for string in remove:
    phish_data['url'] = phish_data['url'].str.replace(string, '')
phish_data['url'] = phish_data['url'].str.split('/', expand=True).get(0)
phish_data['url'] = phish_data['url'].str.strip()


print(phish_data)
```

```
                       url
0     vysodagiva0.xhost.ro
1           188.128.111.33
2           115.28.157.120
3    woodfloorcreations.com
4           115.28.157.120
..                     ...
95       segurosandina.com
96    christmascartoons.org
97    christmascartoons.org
98              mautam.org
99              ehss.co.th

[100 rows x 1 columns]

C:\Users\Lucas\AppData\Local\Temp\ipykernel_1064\685111792.py:7: FutureWarnin
g: The default value of regex will change from True to False in a future vers
ion.
  phish_data['url'] = phish_data['url'].str.replace(string, '')
```

In [5]:

```
c2_data = pd.read_csv('./c2-allmasterlist-high.txt', sep=',', skiprows=21, hea
c2_data = c2_data.loc[:, [0]]
c2_data = c2_data.rename(columns={0: 'url'})
c2_data = c2_data[:100]
print(c2_data)
```

```
                    url
0    ns1.backdates0.org
1   ns1.backdates10.com
2   ns1.backdates12.com
3   ns1.backdates14.com
4   ns1.backdates18.com
..                  ...
95    ngbmfsbuql.yi.org
96          oalierb.com
97     pcajqcaof.yi.org
98    qpyosxkmcc.yi.org
99     qwzsprieo.yi.org

[100 rows x 1 columns]
```

Task 2

In [6]:

```
def extract_domain(url):
    try:
        ipaddress.ip_address(url)
        return url
    except ValueError:
        return '.'.join(url.split('.')[-2:-1])

def extract_tld(url):
    try:
        ipaddress.ip_address(url)
        return url
    except ValueError:
        return url.split('.')[-1]
```

In [7]:

```
tranco_data['domain'] = tranco_data['url'].apply(extract_domain)
tranco_data['tld'] = tranco_data['url'].apply(extract_tld)
tranco_data['domain_length'] = tranco_data['domain'].apply(lambda x: len(x))
bins = [0, 5, 10, 15, float('inf')]
labels = ['1-5', '6-10', '11-15', '16+']
tranco_data['domain_length_group'] = pd.cut(tranco_data['domain_length'], bins
tranco_data.head(10)
```

Out[7]:

	url	domain	tld	domain_length	domain_length_group
0	facebook.com	facebook	com	8	6-10
1	a-msedge.net	a-msedge	net	8	6-10
2	youtube.com	youtube	com	7	6-10
3	microsoft.com	microsoft	com	9	6-10
4	amazonaws.com	amazonaws	com	9	6-10
5	twitter.com	twitter	com	7	6-10
6	baidu.com	baidu	com	5	1-5
7	cloudflare.com	cloudflare	com	10	6-10
8	instagram.com	instagram	com	9	6-10
9	apple.com	apple	com	5	1-5

In [8]:

```
mm_data['domain'] = mm_data['url'].apply(extract_domain)
mm_data['tld'] = mm_data['url'].apply(extract_tld)
mm_data['domain_length'] = mm_data['domain'].apply(lambda x: len(x))
bins = [0, 5, 10, 15, float('inf')]
labels = ['1-5', '6-10', '11-15', '16+']
mm_data['domain_length_group'] = pd.cut(mm_data['domain_length'], bins=bins, l
mm_data.head(10)
```

Out[8]:

	url	domain	tld	domain_length	domain_length_group
0	google.com	google	com	6	6-10
1	facebook.com	facebook	com	8	6-10
2	youtube.com	youtube	com	7	6-10
3	twitter.com	twitter	com	7	6-10
4	instagram.com	instagram	com	9	6-10
5	linkedin.com	linkedin	com	8	6-10
6	apple.com	apple	com	5	1-5
7	microsoft.com	microsoft	com	9	6-10
8	wikipedia.org	wikipedia	org	9	6-10
9	googletagmanager.com	googletagmanager	com	16	16+

In [9]:

```
phish_data['domain'] = phish_data['url'].apply(extract_domain)
phish_data['tld'] = phish_data['url'].apply(extract_tld)
phish_data['domain_length'] = phish_data['domain'].apply(lambda x: len(x))
bins = [0, 5, 10, 15, float('inf')]
labels = ['1-5', '6-10', '11-15', '16+']
phish_data['domain_length_group'] = pd.cut(phish_data['domain_length'], bins=b
phish_data.head(10)
```

Out[9]:

	url	domain	tld	domain_length	domain_length_gro
0	vysodagiva0.xhost.ro	xhost	ro	5	
1	188.128.111.33	188.128.111.33	188.128.111.33	14	11
2	115.28.157.120	115.28.157.120	115.28.157.120	14	11
3	woodfloorcreations.com	woodfloorcreations	com	18	1
4	115.28.157.120	115.28.157.120	115.28.157.120	14	11
5	115.28.157.120	115.28.157.120	115.28.157.120	14	11
6	hghsuppliers.com	hghsuppliers	com	12	11
7	marcaldeataide.com.br	com	br	3	
8	citymarket.imperiavkusov.ru	imperiavkusov	ru	13	11
9	semazen.net	semazen	net	7	6

In [10]:
```
c2_data['domain'] = c2_data['url'].apply(extract_domain)
c2_data['tld'] = c2_data['url'].apply(extract_tld)
c2_data['domain_length'] = c2_data['domain'].apply(lambda x: len(x))
bins = [0, 5, 10, 15, float('inf')]
labels = ['1-5', '6-10', '11-15', '16+']
c2_data['domain_length_group'] = pd.cut(c2_data['domain_length'], bins=bins, l
c2_data.head(10)
```

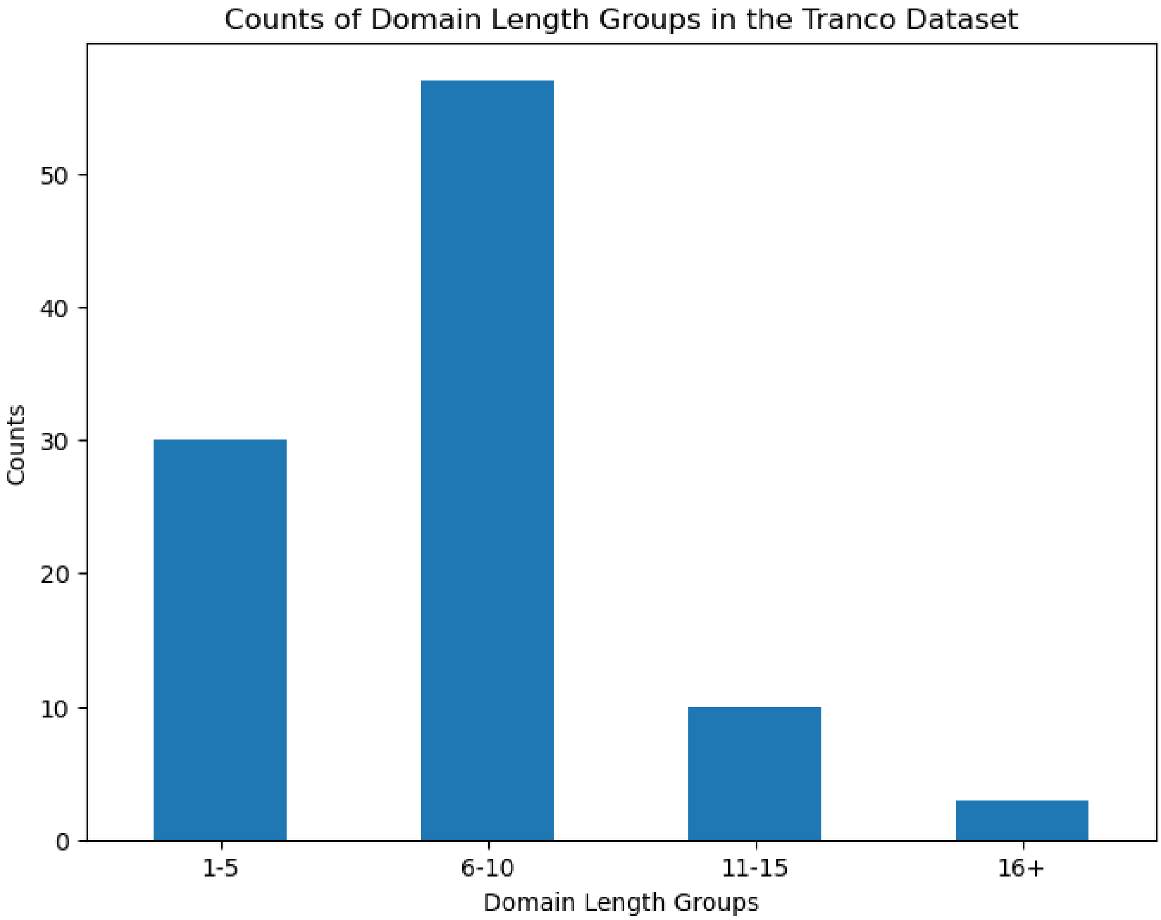
Out[10]:

	url	domain	tld	domain_length	domain_length_group
0	ns1.backdates0.org	backdates0	org	10	6-10
1	ns1.backdates10.com	backdates10	com	11	11-15
2	ns1.backdates12.com	backdates12	com	11	11-15
3	ns1.backdates14.com	backdates14	com	11	11-15
4	ns1.backdates18.com	backdates18	com	11	11-15
5	ns1.backdates20.com	backdates20	com	11	11-15
6	ns1.backdates2.org	backdates2	org	10	6-10
7	ns1.backdates3.org	backdates3	org	10	6-10
8	ns1.backdates4.org	backdates4	org	10	6-10
9	ns1.backdates5.org	backdates5	org	10	6-10

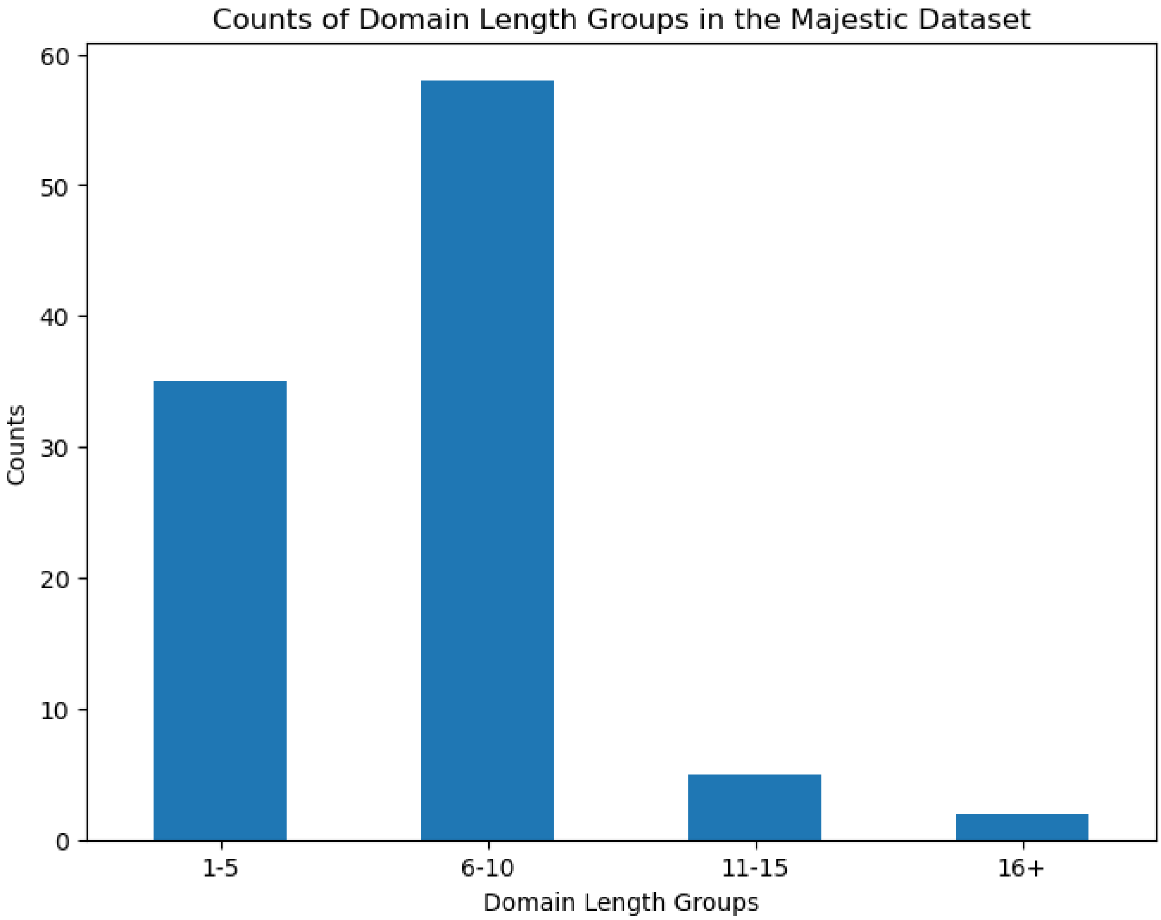
Task 3

In [11]:

```
plt.figure(figsize=(8, 6))
tranco_data.groupby('domain_length_group').size().plot(kind='bar')
plt.title('Counts of Domain Length Groups in the Tranco Dataset')
plt.xlabel('Domain Length Groups')
plt.ylabel('Counts')
plt.xticks(rotation=0)
plt.show()
```

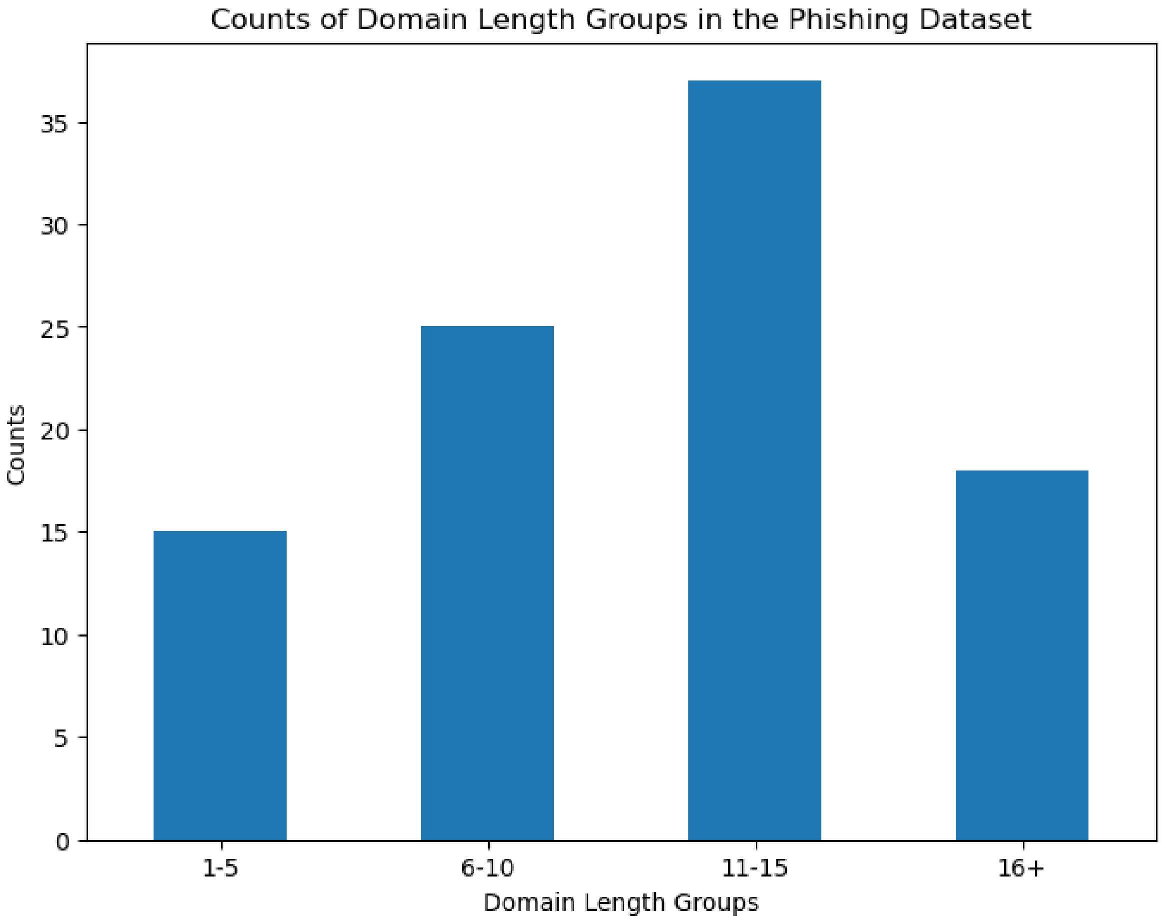


In [12]:
```
plt.figure(figsize=(8, 6))
mm_data.groupby('domain_length_group').size().plot(kind='bar')
plt.title('Counts of Domain Length Groups in the Majestic Dataset')
plt.xlabel('Domain Length Groups')
plt.ylabel('Counts')
plt.xticks(rotation=0)
plt.show()
```

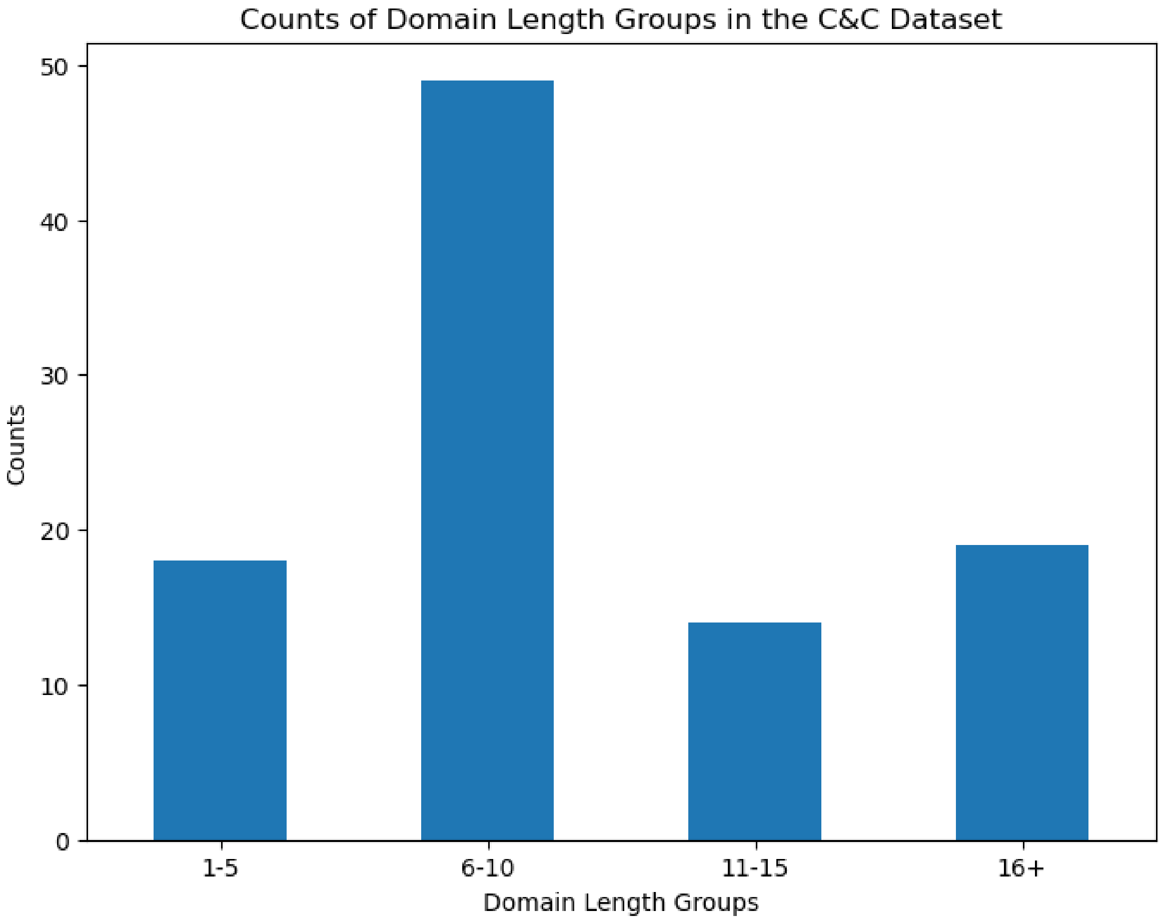


In [13]:

```
plt.figure(figsize=(8, 6))
phish_data.groupby('domain_length_group').size().plot(kind='bar')
plt.title('Counts of Domain Length Groups in the Phishing Dataset')
plt.xlabel('Domain Length Groups')
plt.ylabel('Counts')
plt.xticks(rotation=0)
plt.show()
```



In [14]:
```
plt.figure(figsize=(8, 6))
c2_data.groupby('domain_length_group').size().plot(kind='bar')
plt.title('Counts of Domain Length Groups in the C&C Dataset')
plt.xlabel('Domain Length Groups')
plt.ylabel('Counts')
plt.xticks(rotation=0)
plt.show()
```



task 4

In [15]:

```
#benign
adl_tranco = tranco_data['domain'].apply(len).mean()
adl_mm = mm_data['domain'].apply(len).mean()
#malicous
adl_c2 = c2_data['domain'].apply(len).mean()
adl_phish = phish_data['domain'].apply(len).mean()

adl_benign = (adl_tranco + adl_mm)/2
adl_mal = (adl_c2 + adl_phish)/2

print("Benign Data Sets")
print("The average length of the domain in the Tranco Data set is:", adl_tranc
print("The average length of the domain in the Majestic Data set is:", adl_mm,
print("Malicious Data Sets")
print("The average length of the domain in the C&C Data set is:", adl_c2)
print("The average length of the domain in the Phishing Data set is:", adl_phi

print("Benign VS. Malicious Average Length of Domain")
print("The average length of the domain in the benign data sets is:", adl_beni
print("The average length of the domain in the malicious data sets is: {:.2f}"
```

```
Benign Data Sets
The average length of the domain in the Tranco Data set is: 7.16
The average length of the domain in the Majestic Data set is: 6.41

Malicious Data Sets
The average length of the domain in the C&C Data set is: 10.18
The average length of the domain in the Phishing Data set is: 10.76

Benign VS. Malicious Average Length of Domain
The average length of the domain in the benign data sets is: 6.785
The average length of the domain in the malicious data sets is: 10.47
```

Task 5

In [16]:

```
tranco_data['digit_count'] = tranco_data['domain'].apply(lambda x: sum(c.isdig
mm_data['digit_count'] = mm_data['domain'].apply(lambda x: sum(c.isdigit() for

c2_data['digit_count'] = c2_data['domain'].apply(lambda x: sum(c.isdigit() for
phish_data['digit_count'] = phish_data['domain'].apply(lambda x: sum(c.isdigit

adc_tranco = tranco_data['digit_count'].mean()
adc_mm = mm_data['digit_count'].mean()

adc_c2 = c2_data['digit_count'].mean()
adc_phish = phish_data['digit_count'].mean()

adc_benign = (adc_tranco + adc_mm)/2
adc_mal = (adc_c2 + adc_phish)/2

print("Benign Data Sets")
print("The average digit counts in the Tranco Data set is:", adc_tranco)
print("The average digit counts in the Majestic Data set is:", adc_mm, "\n")
print("Malicious Data Sets")
print("The average digit counts in the C&C Data set is:", adc_c2)
print("The average digit counts in the Phishing Data set is:", adc_phish, "\n"

print("Benign VS. Malicious Average Digit Counts")
print("Average of digit counts in Benign data sets is:", adc_benign)
print("Average of digit counts in Malicious data sets is: {:.2f}".format(adc_m
```

```
Benign Data Sets
The average digit counts in the Tranco Data set is: 0.07
The average digit counts in the Majestic Data set is: 0.01

Malicious Data Sets
The average digit counts in the C&C Data set is: 1.24
The average digit counts in the Phishing Data set is: 0.48

Benign VS. Malicious Average Digit Counts
Average of digit counts in Benign data sets is: 0.04
Average of digit counts in Malicious data sets is: 0.86
```

Task 6

In [17]:

```
tranco_data['unique_char_count'] = tranco_data['domain'].apply(lambda x: len(s
mm_data['unique_char_count'] = mm_data['domain'].apply(lambda x: len(set(x)))

c2_data['unique_char_count'] = c2_data['domain'].apply(lambda x: len(set(x)))
phish_data['unique_char_count'] = phish_data['domain'].apply(lambda x: len(set

ucc_tranco = tranco_data['unique_char_count'].mean()
ucc_mm = mm_data['unique_char_count'].mean()

ucc_c2 = c2_data['unique_char_count'].mean()
ucc_phish = phish_data['unique_char_count'].mean()

ucc_benign = (ucc_tranco + ucc_mm)/2
ucc_mal = (ucc_c2 + ucc_phish)/2

print("Benign Data Sets")
print("The average unique character counts in the Tranco Data set is:", ucc_tr
print("The average unique character counts in the Majestic Data set is:", ucc_
print("Malicious Data Sets")
print("The average unique character counts in the C&C Data set is:", ucc_c2)
print("The average unique character counts in the Phishing Data set is:", ucc_

print("Benign vs Malicious Average Unnique Character counts")
print("Average of unique character counts in benign data sets is:", ucc_benigr
print("Average of unique character counts in malicious data sets is: {:.2f}".f
```

```
Benign Data Sets
The average unique character counts in the Tranco Data set is: 5.94
The average unique character counts in the Majestic Data set is: 5.22

Malicious Data Sets
The average unique character counts in the C&C Data set is: 8.16
The average unique character counts in the Phishing Data set is: 7.64

Benign vs Malicious Average Unnique Character counts
Average of unique character counts in benign data sets is: 5.58
Average of unique character counts in malicious data sets is: 7.90
```

Task 7

In [23]:

```
#benign
tc_tranco = tranco_data['tld'].value_counts(normalize=True) * 100
tc_mm = mm_data['tld'].value_counts(normalize=True) * 100
#malicous
tc_c2 = c2_data['tld'].value_counts(normalize=True) * 100
tc_phish = phish_data['tld'].value_counts(normalize=True) * 100
# report the top 3 TLD distributions
top_3_tranco = tc_tranco[:3].apply(lambda x: '{:.2f} %'.format(x))
top_3_mm = tc_mm[:3].apply(lambda x: '{:.2f} %'.format(x))
top_3_c2 = tc_c2[:3].apply(lambda x: '{:.2f} %'.format(x))
top_3_phish = tc_phish[:3].apply(lambda x: '{:.2f} %'.format(x))

print("Benign Datasets")
print("Top 3 TLD's for the Tranco dataset")
print(top_3_tranco.to_string(header=False),"\n")
print("Top 3 TLD's for the Majestic dataset")
print(top_3_mm.to_string(header=False),"\n")
print("Malicious Datasets")
print("Top 3 TLD's for the C&C dataset")
print(top_3_c2.to_string(header=False),"\n")
print("Top 3 TLD's for the Phishing dataset")
print(top_3_phish.to_string(header=False),"\n")
```

```
Benign Datasets
Top 3 TLD's for the Tranco dataset
com    63.00 %
net    22.00 %
org     3.00 %

Top 3 TLD's for the Majestic dataset
com    69.00 %
org    11.00 %
gov     3.00 %

Malicious Datasets
Top 3 TLD's for the C&C dataset
com    59.00 %
org    27.00 %
net     9.00 %

Top 3 TLD's for the Phishing dataset
com    45.00 %
net    11.00 %
org    10.00 %
```