In [119…

```
import pandas as pd
from tld import get_tld
import tldextract

tranco_df = pd.read_csv("tranco_Y5G4G.csv")
phishTank_df = pd.read_csv("PhishTank-online-banking-phishing-urls-final.csv")
majestic_df = pd.read_csv("majestic_million.csv")
master_list_df = pd.read_csv("c2-allmasterlist-high.txt")
```

In [120…

```
new_row = pd.DataFrame({'1':1, 'google.com':'google.com'}, index=[0])
tranco_df = pd.concat([new_row, tranco_df]).reset_index(drop = True)
tranco_df.rename(columns = {'1':'Rank'}, inplace = True)
tranco_df.rename(columns = {'google.com':'Domain'}, inplace = True)
```

In [121…

```
Domain_no_TLD = []
Domain_len_list = []
Domain_no_list = []
len_list = []
digit_count_list = []
for x in tranco_df["Domain"]:
    this = x.split('.')
    Domain_no_list.append(len(this[0]))
    Domain_no_TLD.append(this[0])
tranco_df["Domain no TLD"] = Domain_no_TLD
tranco_df["Domain no TLD len"] = Domain_no_list

for x in tranco_df["Domain"]:
    if len(x) <= 5 and len(x) >= 1:
        len_list.append("1-5")
    elif len(x) >= 6 and len(x) <= 10:
        len_list.append("6-10")
    elif len(x) >= 11 and len(x) <= 15:
        len_list.append("11-15")
    elif len(x) >= 16:
        len_list.append("16+")
    z = 0
    for y in x:
        if y == '0' or y == "1" or y == "2"or y == "3" or y == "4" or y == "5" o
            z = z + 1
        else:
            z = z
    digit_count_list.append(z)
    Domain_len_list.append(len(x))

tranco_df["Domain Len"] = Domain_len_list
tranco_df["Grouped Len"] = len_list
tranco_df["digit count"] = digit_count_list
tranco_df.head(100)
```

Out[121…

	Rank	Domain	Domain no TLD	Domain no TLD len	Domain Len	Grouped Len	digit count
0	1	google.com	google	6	10	6-10	0
1	2	facebook.com	facebook	8	12	11-15	0

	Rank	Domain	Domain no TLD	Domain no TLD len	Domain Len	Grouped Len	digit count
2	3	a-msedge.net	a-msedge	8	12	11-15	0
3	4	youtube.com	youtube	7	11	11-15	0
4	5	microsoft.com	microsoft	9	13	11-15	0
...	...	...	...	...	...	...	...
95	96	myfritz.net	myfritz	7	11	11-15	0
96	97	ebay.com	ebay	4	8	6-10	0
97	98	google.com.hk	google	6	13	11-15	0
98	99	nytimes.com	nytimes	7	11	11-15	0
99	100	fandom.com	fandom	6	10	6-10	0

100 rows × 7 columns

In [122…

```
def is_ip(address):
    return address.replace('.', '').isnumeric()
hostname_list = []
ip_address_list = []
domain_list = []
domain_list_len = []
len_list = []
digit_count_list = []
for url in phishTank_df['Indicator']:
    hostname_with_path = url.split("//")[1]
    hostname_only = hostname_with_path.split("/")[0]
    if is_ip(hostname_only):
        ip_address_list.append(hostname_only)
    else:
        hostname_list.append(hostname_only)
        domain = tldextract.extract(hostname_only).domain
        if len(domain) <= 5 and len(domain) >= 1:
            len_list.append("1-5")
        elif len(domain) >= 6 and len(domain) <= 10:
            len_list.append("6-10")
        elif len(domain) >= 11 and len(domain) <= 15:
            len_list.append("11-15")
        elif len(domain) >= 16:
            len_list.append("16+")
        z = 0
        for y in domain:
            if y == '0' or y == "1" or y == "2"or y == "3" or y == "4" or y == "
                z = z + 1
            else:
                z = z
        digit_count_list.append(z)
        domain_list_len.append(len(domain))
        domain_list.append(domain)

phishTank_new_df = pd.DataFrame(hostname_list, columns=['hostname'])
phishTank_new_df['domain'] = domain_list
phishTank_new_df['domain length'] = domain_list_len
phishTank_new_df['Grouped Len'] = len_list
```

```
phishTank_new_df['digit count'] = digit_count_list
phishTank_new_df.sample(n=100)
```

Out[122…

	hostname	domain	domain length	Grouped Len	digit count
104	momentumsurfandskate.com	momentumsurfandskate	20	16+	0
89	www.portugalkaraoke.com	portugalkaraoke	15	11-15	0
216	www.asianstss.org	asianstss	9	6-10	0
336	bankohlventures.com	bankohlventures	15	11-15	0
359	bclbank.com	bclbank	7	6-10	0
...	...	...	...	...	...
381	minhon.pt	minhon	6	6-10	0
241	greeneandassociates.biz	greeneandassociates	19	16+	0
209	xarabank.com.mt	xarabank	8	6-10	0
227	www.qualityhandles.com	qualityhandles	14	11-15	0
12	allstarprintz.com	allstarprintz	13	11-15	0

100 rows × 5 columns

In [124…

```
Domain_no_TLD = []
Domain_len_list = []
digit_count_list = []
unique_list = []
for x in majestic_df["Domain"]:
    this = x.split('.')
    Domain_len_list.append(len(this[0]))
    Domain_no_TLD.append(this[0])

majestic_df["Domain no TLD"] = Domain_no_TLD
majestic_df["Domain len"] = Domain_len_list
majestic_df.head(100)

Domain_TLD_len_list = []
len_list = []
for x in majestic_df["Domain"]:
    Domain_TLD_len_list.append(len(x))
    if len(x) <= 5 and len(x) >= 1:
        len_list.append("1-5")
    elif len(x) >= 6 and len(x) <= 10:
        len_list.append("6-10")
    elif len(x) >= 11 and len(x) <= 15:
        len_list.append("11-15")
    elif len(x) >= 16:
        len_list.append("16+")
    z = 0
    for y in x:
        if y == '0' or y == "1" or y == "2"or y == "3" or y == "4" or y == "5" o
            z = z + 1
        else:
            z = z
    digit_count_list.append(z)
```

```
majestic_df["Domain TLD len"] = Domain_TLD_len_list
majestic_df['Grouped Len'] = len_list
majestic_df['digit count'] = digit_count_list
majestic_df.head(100)
```

Out[124…

	GlobalRank	TldRank	Domain	TLD	RefSubNets	RefIPs	IDN_Domain	IDN_TLD	P
0	1	1	google.com	com	493225	2423262	google.com	com	
1	2	2	facebook.com	com	491991	2576708	facebook.com	com	
2	3	3	youtube.com	com	443833	2089772	youtube.com	com	
3	4	4	twitter.com	com	437495	2073037	twitter.com	com	
4	5	5	instagram.com	com	375641	1733941	instagram.com	com	
…	…	…	…	…	…	…	…	…	
95	96	66	youtube-nocookie.com	com	84737	208923	youtube-nocookie.com	com	
96	97	67	nginx.com	com	84535	165377	nginx.com	com	
97	98	68	imdb.com	com	84275	218180	imdb.com	com	
98	99	69	bloomberg.com	com	84239	196751	bloomberg.com	com	
99	100	1	harvard.edu	edu	84049	193996	harvard.edu	edu	

100 rows × 17 columns

In [125…

```
new_master_list_df = pd.DataFrame(master_list_df['Domain'])
Domain_len_list =[]
len_list = []
digit_count_list = []
for x in new_master_list_df['Domain']:
    if len(x) <= 5 and len(x) >= 1:
        len_list.append("1-5")
    elif len(x) >= 6 and len(x) <= 10:
        len_list.append("6-10")
    elif len(x) >= 11 and len(x) <= 15:
        len_list.append("11-15")
    elif len(x) >= 16:
        len_list.append("16+")
    z = 0
    for y in x:
        if y == '0' or y == "1" or y == "2"or y == "3" or y == "4" or y == "5" o
            z = z + 1
        else:
            z = z
    digit_count_list.append(z)
    Domain_len_list.append(len(x))

new_master_list_df["Domain Len"] = Domain_len_list
new_master_list_df['Grouped Len'] = len_list
new_master_list_df['digit count'] = digit_count_list
new_master_list_df.head(100)
```

Out[125…

	Domain	Domain Len	Grouped Len	digit count
0	ns1.backdates0.org	18	16+	2
1	ns1.backdates10.com	19	16+	3
2	ns1.backdates12.com	19	16+	3
3	ns1.backdates14.com	19	16+	3
4	ns1.backdates18.com	19	16+	3
...	...	...	...	...
95	ngbmfsbuql.yi.org	17	16+	0
96	oalierb.com	11	11-15	0
97	pcajqcaof.yi.org	16	16+	0
98	qpyosxkmcc.yi.org	17	16+	0
99	qwzsprieo.yi.org	16	16+	0

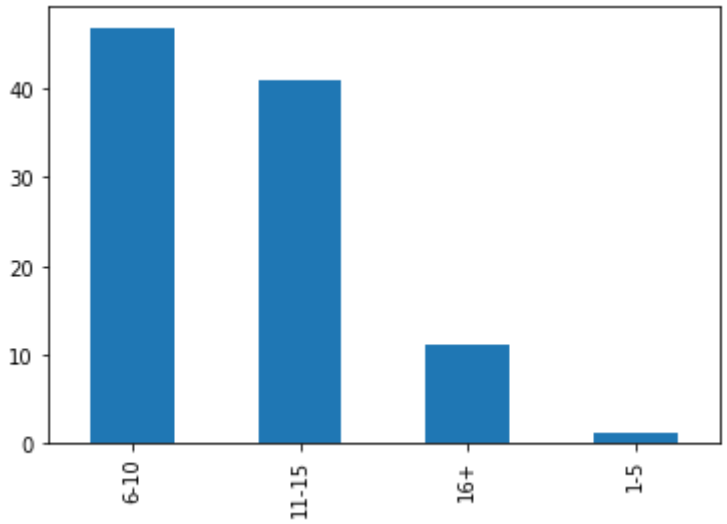
100 rows × 4 columns

In [126…

```
tranco_df['Grouped Len'].head(100).value_counts().plot(kind='bar')
```

Out[126…

```
<AxesSubplot:>
```

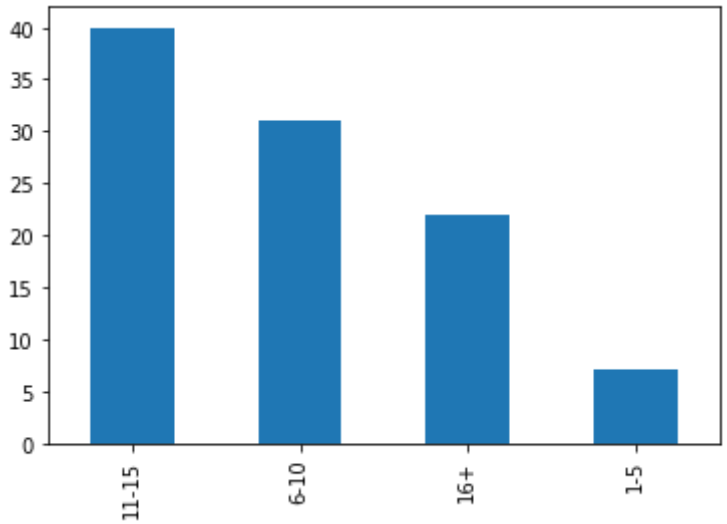


In [127…

```
phishTank_new_df['Grouped Len'].head(100).value_counts().plot(kind='bar')
```

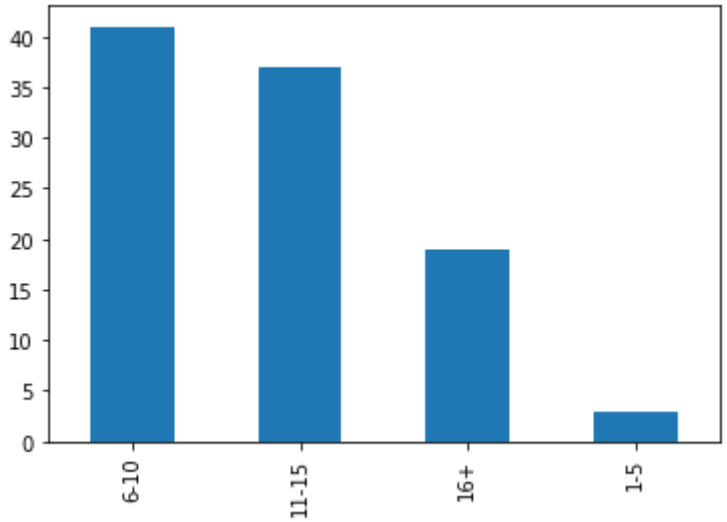
Out[127…

```
<AxesSubplot:>
```



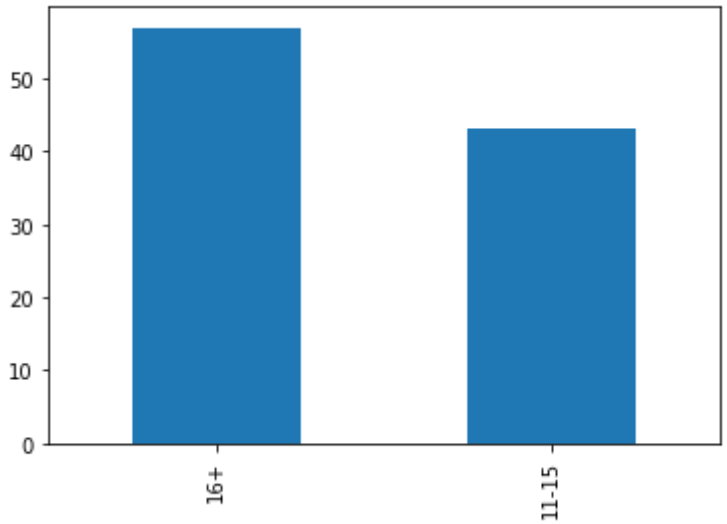
In [128…
```
majestic_df['Grouped Len'].head(100).value_counts().plot(kind='bar')
```

Out[128…
```
<AxesSubplot:>
```



In [129…
```
new_master_list_df['Grouped Len'].head(100).value_counts().plot(kind='bar')
```

Out[129…
```
<AxesSubplot:>
```



Yes the average domain length for malicious is much larger than good urls

In [135…

```
def count_unique_chars(domain):
    tld = domain.split('.')[-1]
    domain_no_tld = domain[:-len(tld)-1]
    unique_chars = set(domain_no_tld)
    return len(unique_chars)

tranco_df["unique char count"] = tranco_df['Domain'].apply(count_unique_chars)
tranco_df.head(100)
```

Out[135…

	Rank	Domain	Domain no TLD	Domain no TLD len	Domain Len	Grouped Len	digit count	unique char count
0	1	google.com	google	6	10	6-10	0	4
1	2	facebook.com	facebook	8	12	11-15	0	7
2	3	a-msedge.net	a-msedge	8	12	11-15	0	7
3	4	youtube.com	youtube	7	11	11-15	0	6
4	5	microsoft.com	microsoft	9	13	11-15	0	8
...	...	...	...	...	...	...	...	...
95	96	myfritz.net	myfritz	7	11	11-15	0	7
96	97	ebay.com	ebay	4	8	6-10	0	4
97	98	google.com.hk	google	6	13	11-15	0	7
98	99	nytimes.com	nytimes	7	11	11-15	0	7
99	100	fandom.com	fandom	6	10	6-10	0	6

100 rows × 8 columns

In [136…

```
phishTank_new_df["unique char count"] = phishTank_new_df['hostname'].apply(count
phishTank_new_df.head(100)
```

Out[136…

	hostname	domain	domain length	Grouped Len	digit count	unique char count
0	vysodagiva0.xhost.ro	xhost	5	1-5	0	13
1	woodfloorcreations.com	woodfloorcreations	18	16+	0	13
2	hghsuppliers.com	hghsuppliers	12	11-15	0	9
3	marcaldeataide.com.br	marcaldeataide	14	11-15	0	11
4	citymarket.imperiavkusov.ru	imperiavkusov	13	11-15	0	15
…	…	…	…	…	…	…
95	ehss.co.th	ehss	4	1-5	0	6
96	www.scatolificiogiani.it	scatolificiogiani	17	16+	0	12
97	www.familylifebc.com	familylifebc	12	11-15	0	11
98	foundus.my	foundus	7	6-10	0	6
99	foundus.my	foundus	7	6-10	0	6

100 rows × 6 columns

In [137…

```
majestic_df["unique char count"] = majestic_df['Domain'].apply(count_unique_char
majestic_df.head(100)
```

Out[137…

	GlobalRank	TldRank	Domain	TLD	RefSubNets	RefIPs	IDN_Domain	IDN_TLD	P
0	1	1	google.com	com	493225	2423262	google.com	com	
1	2	2	facebook.com	com	491991	2576708	facebook.com	com	
2	3	3	youtube.com	com	443833	2089772	youtube.com	com	
3	4	4	twitter.com	com	437495	2073037	twitter.com	com	
4	5	5	instagram.com	com	375641	1733941	instagram.com	com	
…	…	…	…	…	…	…	…	…	
95	96	66	youtube-nocookie.com	com	84737	208923	youtube-nocookie.com	com	
96	97	67	nginx.com	com	84535	165377	nginx.com	com	
97	98	68	imdb.com	com	84275	218180	imdb.com	com	
98	99	69	bloomberg.com	com	84239	196751	bloomberg.com	com	
99	100	1	harvard.edu	edu	84049	193996	harvard.edu	edu	

100 rows × 18 columns

In [138…

```
new_master_list_df["unique char count"] = new_master_list_df['Domain'].apply(cou
new_master_list_df.head(100)
```

Out[138…

	Domain	Domain Len	Grouped Len	digit count	unique char count
0	ns1.backdates0.org	18	16+	2	12
1	ns1.backdates10.com	19	16+	3	12
2	ns1.backdates12.com	19	16+	3	12
3	ns1.backdates14.com	19	16+	3	12
4	ns1.backdates18.com	19	16+	3	12
...	...	...	...	...	...
95	ngbmfsbuql.yi.org	17	16+	0	12
96	oalierb.com	11	11-15	0	7
97	pcajqcaof.yi.org	16	16+	0	10
98	qpyosxkmcc.yi.org	17	16+	0	11
99	qwzsprieo.yi.org	16	16+	0	11

100 rows × 5 columns

In [141…

```
def top_3_TLDs(group):
    tld_counts = group.str.split('.').str[-1].value_counts(normalize=True)
    return tld_counts[:3].apply(lambda x: f'{x:.2%}')

result = tranco_df.head(100).groupby('Grouped Len')['Domain'].apply(top_3_TLDs)
print(result)
```

```
Grouped Len
1-5          co     100.00%
11-15        com     56.10%
             net     31.71%
             org      7.32%
16+          com     72.73%
             net     27.27%
6-10         com     68.09%
             net     12.77%
             ru       4.26%
Name: Domain, dtype: object
```

In [143…

```
result = phishTank_new_df.head(100).groupby('Grouped Len')['hostname'].apply(top
print(result)
```

```
Grouped Len
1-5          com    28.57%
             ro     14.29%
             net    14.29%
11-15        com    47.50%
             org    17.50%
             br     10.00%
16+          com    72.73%
             org     9.09%
             my      4.55%
6-10         com    41.94%
             net    16.13%
```

```
                my      9.68%
Name: hostname, dtype: object
```

In [145…

```
result = majestic_df.head(100).groupby('Grouped Len')['Domain'].apply(top_3_TLDs
print(result)
```

```
Grouped Len
1-5           me     66.67%
              co     33.33%
11-15         com    75.68%
              org    13.51%
              cn      2.70%
16+           com    73.68%
              org    15.79%
              cn      5.26%
6-10          com    65.85%
              org     7.32%
              gov     4.88%
Name: Domain, dtype: object
```

In [146…

```
result = new_master_list_df.head(100).groupby('Grouped Len')['Domain'].apply(top
print(result)
```

```
Grouped Len
11-15         com    58.14%
              org    20.93%
              net     9.30%
16+           com    59.65%
              org    31.58%
              net     8.77%
Name: Domain, dtype: object
```

In [ ]: