gynimas4_finalas

December 12, 2022

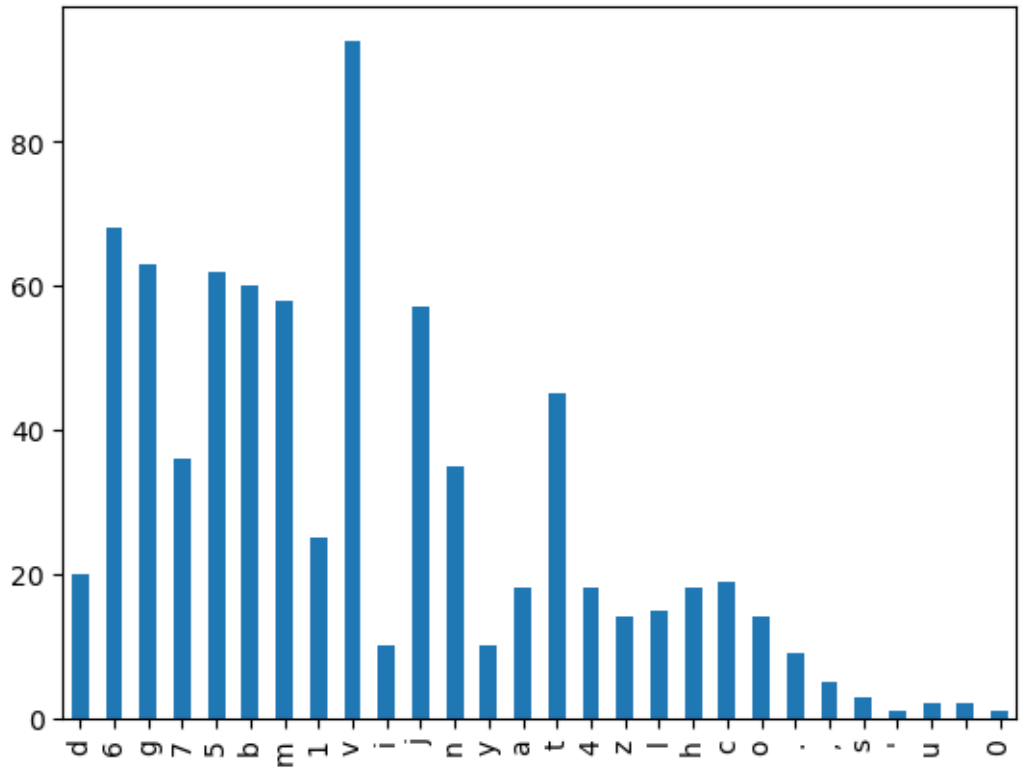
[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[ ]:
```
cypher = """d6g756b 6m 1vm6ibv1 g5 jnn5y 6gm amvtm g5 mvb1 jb1 tv7v64v zjlhvbgm␣
 ↪y6gc jb j77vzgjdnv nv4vn 5o zt64j7l jm yvnn jm jbl 5gcvt o5th 5o h5bvl.␣
 ↪c5yv4vt, d6g756b 6m b5g jb5blh5am jb1 7jbb5g 5oovt gcv mjhv nv4vn 5o zt64j7l␣
 ↪jm 7jmc. gcv amv 5o d6g756b nvj4vm vsgvbm64v zadn67 tv75t1m. 4jt65am␣
 ↪hv7cjb6mhm vs6mg g5 zt5gv7g amvtm' zt64j7l, jb1 h5tv jtv 6b 1v4vn5zhvbg.␣
 ↪c5yv4vt, gcvtv 6m mg6nn y5tu g5 dv 15bv dvo5tv gcvmv ovjgatvm jtv amv1␣
 ↪75ttv7gnl dl h5mg d6g756b amvtm.
m5hv 75b7vtbm cj4v dvvb tj6mv1 gcjg zt64jgv gtjbmj7g65bm 75an1 dv amv1 o5t␣
 ↪6nnvijn zatz5mvm y6gc d6g756b. c5yv4vt, 6g 6m y5tgc b5g6bi gcjg d6g756b y6nn␣
 ↪ab15adgv1nl dv mad0v7gv1 g5 m6h6njt tvianjg65bm gcjg jtv jntvj1l 6b znj7v␣
 ↪6bm61v vs6mg6bi o6bjb76jn mlmgvhm. d6g756b 7jbb5g dv h5tv jb5blh5am gcjb␣
 ↪7jmc jb1 6g 6m b5g n6uvnl g5 ztv4vbg 7t6h6bjn 6b4vmg6ijg65bm ot5h dv6bi␣
 ↪75b1a7gv1. j116g65bjnnl, d6g756b 6m jnm5 1vm6ibv1 g5 ztv4vbg j njtiv tjbiv␣
 ↪5o o6bjb76jn 7t6hvm.
"""
```

[ ]:
```
alphabet = "abcdefghijklmnoprstuvwxyz0123456789"
```

[ ]:
```
# plot the frequency of each letter in the cypher
all_freq = {}

for i in cypher:
    if i == " ":
        continue
    if i in all_freq:
        all_freq[i] += 1
    else:
        all_freq[i] = 1
#plot frequencies
frequencies = pd.Series(all_freq)
frequencies.plot(kind='bar')
plt.show()
```



[ ]:
```
# get the most frequent 5 letters
frequencies.sort_values(ascending=False).head(10).index
```

[ ]:
```
Index(['v', '6', 'g', '5', 'b', 'm', 'j', 't', '7', 'n'], dtype='object')
```

[ ]:
```
map1 = {}
i = 0
for letter in alphabet:
    map1[letter] = i
    i += 1
```

[ ]:
```
# Most popular letters: e, a, r, i, o, t, n, s
# Most popular letters in cypher: v, 6, g, 5, b

# Create most likely mapping pairs
```

[ ]:
```
# Most popular letters: e, a, r, i, o, t, n, s
# Most popular letters in cypher: v, 6, g, 5, b
#  e->v ; a->6
# solve affine cypher
```

```
# solve c_1 = a * t_1 + b (mod 35)
# solve c_2 = a * t_2 + b (mod 35)

c_1 = map1['v'] # place of v
t_1 = map1['e'] # place of e

c_2 = map1['6'] # place of 6
t_2 = map1['a'] # place of a

# solve for a and b
a = (c_2 - c_1)*(t_2 - t_1)**-1 % 35
b = (c_1 - a*t_1) % 35

print(a ,b) # does not exist
```

32.25 31.0

[ ]:
```
# Most popular letters: e, a, r, i, o, t, n, s
# Most popular letters in cypher: v, 6, g, 5, b
#  e->v ; a->g
# solve affine cypher


# solve c_1 = a * t_1 + b (mod 35)
# solve c_2 = a * t_2 + b (mod 35)

c_1 = map1['v']
t_1 = map1['e']

c_2 = map1['g']
t_2 = map1['a']

# solve for a and b
a = (c_2 - c_1)*(t_2 - t_1)**-1 % 35
b = (c_1 - a*t_1) % 35

print(a ,b) # does not exist
```

3.5 6.0

[ ]:
```
# Most popular letters: e, a, r, i, o, t, n, s
# Most popular letters in cypher: v, 6, g, 5, b
#  e->v ; a->5
# solve affine cypher


# solve c_1 = a * t_1 + b (mod 35)
```

```
# solve c_2 = a * t_2 + b (mod 35)

c_1 = map1['v']
t_1 = map1['e']

c_2 = map1['5']
t_2 = map1['a']

# solve for a and b
a = (c_2 - c_1)*(t_2 - t_1)**-1 % 35
b = (c_1 - a*t_1) % 35

print(a ,b) # does not exist
```

```
32.5 30.0
```

[ ]:
```
# Most popular letters: e, a, r, i, o, t, n, s
# Most popular letters in cypher: v, 6, g, 5, b
#  e->v ; a->b
# solve affine cypher


# solve c_1 = a * t_1 + b (mod 35)
# solve c_2 = a * t_2 + b (mod 35)

c_1 = map1['v']
t_1 = map1['e']

c_2 = map1['b']
t_2 = map1['a']

# solve for a and b
a = (c_2 - c_1)*(t_2 - t_1)**-1 % 35
b = (c_1 - a*t_1) % 35

print(a ,b) # does not exist
```

```
4.75 1.0
```

[ ]:
```
# Most popular letters: e, a, r, i, o, t, n, s
# Most popular letters in cypher: v, 6, g, 5, b
#  a->v ; e->6
# solve affine cypher


# solve c_1 = a * t_1 + b (mod 35)
# solve c_2 = a * t_2 + b (mod 35)
```

```
c_1 = map1['v']
t_1 = map1['a']

c_2 = map1['6']
t_2 = map1['e']

# solve for a and b
a = (c_2 - c_1)*(t_2 - t_1)**-1 % 35
b = (c_1 - a*t_1) % 35

print(a ,b) # does not exist
```

```
2.75 20.0
```

[ ]:
```
# Most popular letters: e, a, r, i, o, t, n, s
# Most popular letters in cypher: v, 6, g, 5, b
#  a->v ; r->6
# solve affine cypher


# solve c_1 = a * t_1 + b (mod 35)
# solve c_2 = a * t_2 + b (mod 35)

c_1 = map1['v']
t_1 = map1['a']

c_2 = map1['6']
t_2 = map1['r']

# solve for a and b
a = (c_2 - c_1)*(t_2 - t_1)**-1 % 35
b = (c_1 - a*t_1) % 35

print(a ,b) # does not exist
```

```
0.6875 20.0
```

[ ]:
```
# Most popular letters: e, a, r, i, o, t, n, s
# Most popular letters in cypher: v, 6, g, 5, b
#  e->v ; r->6
# solve affine cypher


# solve c_1 = a * t_1 + b (mod 35)
# solve c_2 = a * t_2 + b (mod 35)
```

```
c_1 = map1['v']
t_1 = map1['e']

c_2 = map1['6']
t_2 = map1['r']

# solve for a and b
a = (c_2 - c_1)*(t_2 - t_1)**-1 % 35
b = (c_1 - a*t_1) % 35

print(a ,b) # does not exist
```

```
0.9166666666666666 16.333333333333332
```

[ ]:
```
# Most popular letters: e, a, r, i, o, t, n, s
# Most popular letters in cypher: v, 6, g, 5, b
#  e->v ; r->6
# solve affine cypher


# solve c_1 = a * t_1 + b (mod 35)
# solve c_2 = a * t_2 + b (mod 35)

c_1 = map1['v']
t_1 = map1['e']

c_2 = map1['6']
t_2 = map1['r']

# solve for a and b
a = (c_2 - c_1)*(t_2 - t_1)**-1 % 35
b = (c_1 - a*t_1) % 35

print(a ,b) # does not exist
```

```
0.9166666666666666 16.333333333333332
```

[ ]:
```
# Nerandu raktu, ieskau lyg teisingai, bet nerandu sveiku poru, labai tikiuosi␣
↪5 uz pastangas parasysit.. 0_0
```