

Frequency analysis

```
In [1]: from __future__ import unicode_literals
```

```
In [2]: crypted_text = """РГШВГЕЕКГИШБТНВВЦОВБФЕОЫЛАНЖВЕЕВБОГШЕТГЗЧЕЫНВЕНЕРБВБДНИЖГ  
Р  
ЕУБАНЖНЗИГРБРЫГТГШГЗИЕЗРГНЕЗТЛЯЕТГВРЫРБЖШЕЕРЦФНТРГИЗИБРСЛРВБУ  
БТНЫГШБЛНОБТРЗРГЭШНЖНРВЭЕЗИНОДГЖГВГИИЛШБВНРЦНКЯБТГВАЦТЯНВБ  
ИВБАНШВГЕШРГЖЮВСНСГИГЖБЮЛЬНЖТБРЖГШБОРИГРЖНЬЮСБСГВВБОГШЕТ  
ЗЮРГИЧНКЯНЬДГТНОГКЮЕЗИРНВВЦНЛДЖБЯВНВЕЮЗСГЖНЫГЛИНФЕТЕГВРЦЗ  
ИЖГЕТШГЬДГЗГАЗИРНВВГЬЛДТБВЛКБРНТЛЗНАЮЗЛСГВВЛЭМБАЖЕСЛЛИЖЕТ  
ШГОГШЦЕЗИБТДГУЕИБИЧЗНАЮЛЬВНЕФЕБУНТГРНСГЬРГРЗНЬГСГТГИСНРУНЬЕВ  
НДЖНСГЗТГРЕТЕНЬЛЗГЗНШЕДЖЕНКЯБРФЕНСВНЬЛЫГЗИЕИЧЗЗРГЕЬЕЗНЬНЕЗИРБ  
ЬЕЕЗГАБСБЬЕРАЛШВЕГВОВГШЕТРДТЕЗРГЕСЛЖИСНДГДЖБКШВЕСБЬВБШНРБТЗН  
ЖИЛСЕКЗЛСВБШГЬБФВНЕЖБАГИЦЗБЬКБДЕЗЦРБТЖБЗОГШЕВЕУНЫГВНУЕИБТСЖ  
ГЫЗНВБИЗСЕОРНШГЫГЗИНЕ"""
```

```
In [3]: import re  
import pandas as pd  
from collections import Counter  
import matplotlib  
import numpy as np  
import matplotlib.pyplot as plt  
%matplotlib inline  
matplotlib.rc('font',**{'family':'Droid Serif'})
```

Get alphabet frequencies (from Wikipedia)

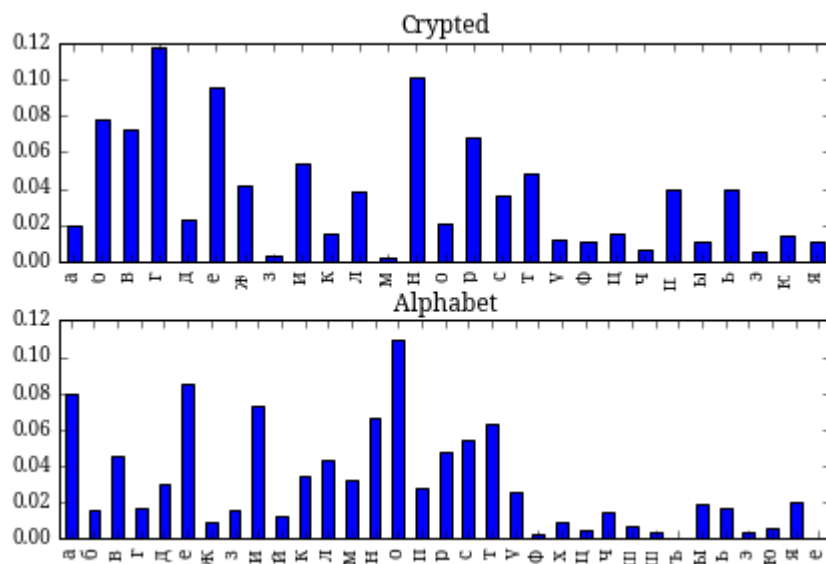
```
In [4]: alphabet_frequencies = pd.Series.from_csv('alphabet_frequencies.csv', encod  
ing='utf-8')  
# alphabet_frequencies.plot(kind='bar')
```

Get frequencies from text

```
In [5]: # prepare text for treatment  
symbols_list = []  
for symbol in crypted_text.lower():  
    if re.match('[а-я]', symbol):  
        symbols_list.append(symbol)  
  
letters_counts = Counter(symbols_list) #{'alpha' : count}  
  
# normalization to 1  
alphas_count = sum(letters_counts.values())  
letters_frequencies = {}  
for alpha, count in letters_counts.iteritems():  
    letters_frequencies[alpha] = float(count) / alphas_count  
crypted_frequencies = pd.Series(data=letters_frequencies)
```

Plot results

```
In [6]: fig, axes = plt.subplots(nrows=2, ncols=1)
fig.tight_layout()
encrypted_frequencies.sort_index(inplace=True, ascending=True) # you can replace it with sort_values
alphabet_frequencies.sort_index(inplace=True, ascending=True)
encrypted_frequencies.plot(kind='bar', title='Crypted', ax=axes[0])
alphabet_frequencies.plot(kind='bar', title='Alphabet', ax=axes[1])
plt.show()
```



Testing by replace symbols with highes frequency

```
In [20]: encrypted_alpha = encrypted_frequencies.sort_values().index[-1]
alphabet_alpha = alphabet_frequencies.sort_values().index[-1]
print("crypted alpha '{}' is uncrypted '{}'".format(encrypted_alpha, alphabet_alpha))
```

crypted alpha 'r' is uncrypted 'o'

```
In [21]: # Let's try to uncript text
diff = ord(alphabet_alpha) - ord(encrypted_alpha)
```

```
In [22]: uncrypted_text = []
start = ord(u'a')
end = ord(u'я')
n = end - start

for crypted_alpha in symbols_list:
    index_in_alphabet = ord(crypted_alpha) - start
    uncrypted_alpha = unichr(start + (index_in_alphabet - diff + n) % n)
    uncrypted_text.append(uncrypted_alpha)
```

Crypted text

```
In [23]: print "".join(symbols_list)
```

ргшвгеекгишбтнvwцовбфеоыланжвеевбогшетгчеьнвенербvbднижгреубанжнигрбрьгтгшги
ергнетляетгврырбжшеерцфнтргиибрслрвбубтныгшблнобтрргэшнжнрвэеинодгжгвгиилшбв
нрцнкябтгвацтянвбивбаншвгешргжювснсгигжбюльнжтбржгшборигржньюсбсгввбогшетюрг
ичнкяньдгтногкюеирнvwцнлджбявнвеюсгжгныглинфетегврцижетшгьдггаирнvwгьлдтбвл
кбрнтлнаюлсгввлэмбажесллижетшгогшцеибтдгуеибичнаюльвнефеьунтгрнсгьргрзньгсг
тгиснруньевнджнсгзтгретеньлгншедженкябрфенсвньлыгиеичргеьеньнеирбьегабсбьер
алшвегвогшетрдтегреслжисндгджбкшвесбьвбшнрбтнжилсеклсвбшгьбфвнежагицбькбде
црбтжбогшевеуныгвнуеибтсжгьннvwбисеорншгьгине

Uncrypted text

```
In [24]: print "".join(uncrypted_text)
```

ечнццщцючънхзвццлгцхйщграфвъцщццхгчнщзчмщсвцщвщехцхшвъъчещихфвъвъчехесчзчнчь
щечвщзафщзчццерехънщцелйвзечъъхежаецхихзврчнхавгхзеечтнвъвещтщъвгшчъцчъьанхц
велвюфхзчцфлзфвцхъцхфвнцчщнечъуцжвжчъчъхуасвъзхеъчнхгеьчеьвсужхжчццхгчнщзуеч
ьмвюфвсшчзвгчюущъевццлвашъхфцвцщужчъчврчаъвйщзщццельъчщзнчсшччфъевццчсашзхца
юхевзавфуажчццатбхфъщжааъъчщзнчгчнлщъхзшчищъхьмвфуасцвщйщсивзчевжчсечеывсчжч
зчъжвеивсщцвшъвжчызчещзщвсачвнщшъщвюфхейщвжцвсарчъщъмечщсщвсщъьехсщччфхжхсше
фанцщчцгчнщзешзщчечщжаъъжвшчшъхюнцщжхсцхнвехзвъъажщюажцхнчсхйцвщъхфчълхсюхшщ
лехзъхгчнщцщиврчцвищъхзжъчсввцхъжщгевнчсчъвщ

```
In [ ]:
```