Приложение 3. Криптоанализ подстановочных шифров

Часть 1. Шифр простой замены

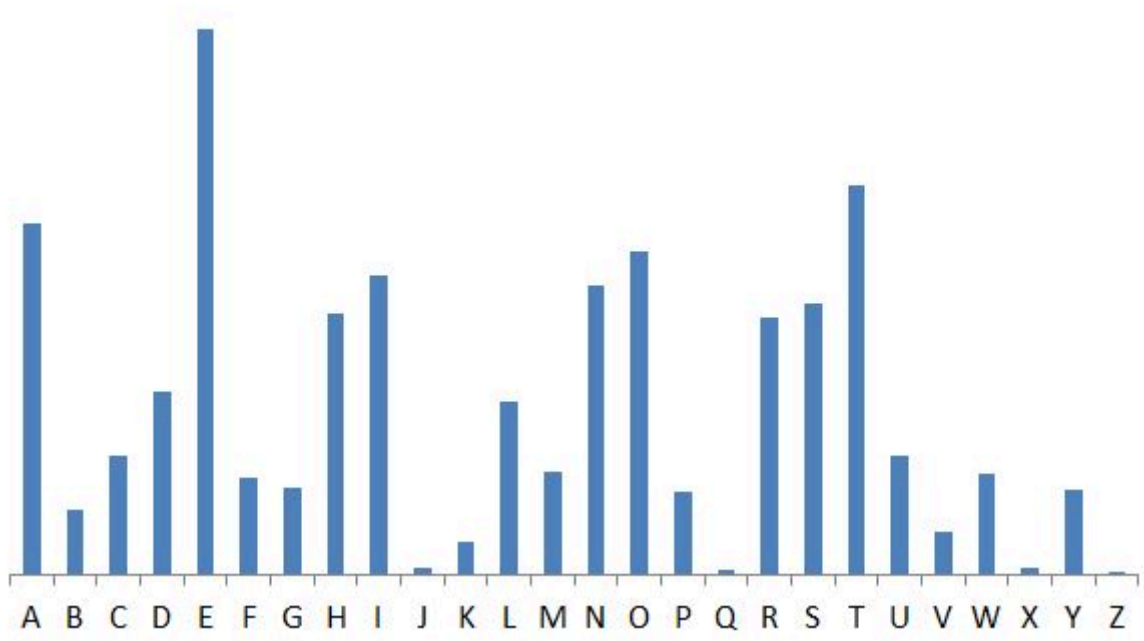
Шифр простой замены поддается частотному анализу так как переносит статистические характеристики языка на шифртекст. Для успешного взлома ключа шифрования шифра простой замены, необходимо иметь достаточно большой текст чтобы корректно проанализировать его.

На рисунке ниже представлена статистика по частотночти букв английского языка в английском тексте.

In [1]:
```
from PIL import Image

Image.open("english.jpg")
```

Out[1]:



In [2]:
```
# Например у нас есть корпус текста который мы хотели бы расшифровать

open_text = """
    Natural language processing is a branch of artificial intelligence
    that enables computers to comprehend, generate, and manipulate human
    language. Natural language processing has the ability to interrogate the
    data with natural language text or voice. This is also called language in.
    Most consumers have probably interacted with NLP without realizing it. For
    instance, NLP is the core technology behind virtual assistants, such as
    the Oracle Digital Assistant, Siri, Cortana, or Alexa. When we ask
    questions of these virtual assistants, NLP is what enables them to not only
    understand the users request, but to also respond in natural language.
    language applies both to written text and speech, and can be applied to all
    human languages. Other examples of tools powered by NLP include web search,
    email spam filtering, automatic translation of text or speech, document
    summarization, sentiment analysis, and grammar spell checking. For example,
    some email programs can automatically suggest an appropriate reply to a
    message based on its content - these programs use language to read, analyze, and
    respond to your message. There are several other terms that are roughly
    synonymous with NLP. Natural language understanding and natural
    language generation refer to using computers to understand and produce
    human language, respectively. NLG has the ability to provide a verbal
    description of what has happened. This is also called language out by
    summarizing by meaningful information into text using a concept known as
    grammar of graphics. In practice, NLU is used to mean NLP. The understanding
    by computers of the structure and meaning of all human languages, allowing
    developers and users to interact with computers using natural sentences and
    communication. Computational linguistics is the scientific field that
    studies computational aspects of human language, while language is the engineering
    discipline concerned with building computational artifacts that understand,
    generate, or manipulate human language. Research on language began shortly after
    the invention of digital computers in the 1950s, and language draws on both
    linguistics and artificial intellect. However, the major breakthroughs of the past
    few years have been powered by machine learning, which is a branch of AI that
    develops systems that learn and generalize from data. Deep learning is a kind of
    machine learning that can learn very complex patterns from large datasets,
    which means that it is ideally suited to learning the complexities of natural
    language from datasets sourced from the web.
"""
```

In [3]:
```
# Немного предобработаю текст - удалю переносы строк и дублирующиеся пробелы
# Также упрощу себе задачу и сведу алфавит до нстрочных букв

open_text = " ".join(open_text.replace("\n", "").split()).lower()
open_text
```

Out[3]:
```
'natural language processing is a branch of artificial intelligence that enables computers to comprehend,
generate, and manipulate human language. natural language processing has the ability to interrogate the da
ta with natural language text or voice. this is also called language in. most consumers have probably inte
racted with nlp without realizing it. for instance, nlp is the core technology behind virtual assistants,
such as the oracle digital assistant, siri, cortana, or alexa. when we ask questions of these virtual assi
stants, nlp is what enables them to not only understand the users request, but to also respond in natural
language. language applies both to written text and speech, and can be applied to all human languages. oth
er examples of tools powered by nlp include web search, email spam filtering, automatic translation of tex
t or speech, document summarization, sentiment analysis, and grammar spell checking. for example, some ema
il programs can automatically suggest an appropriate reply to a message based on its content - these progr
ams use language to read, analyze, and respond to your message. there are several other terms that are rou
ghly synonymous with nlp. natural language understanding and natural language generation refer to using co
mputers to understand and produce human language, respectively. nlg has the ability to provide a verbal de
scription of what has happened. this is also called language out by summarizing by meaningful information
into text using a concept known as grammar of graphics. in practice, nlu is used to mean nlp. the understa
nding by computers of the structure and meaning of all human languages, allowing developers and users to i
nteract with computers using natural sentences and communication. computational linguistics is the scienti
fic field that studies computational aspects of human language, while language is the engineering discipli
ne concerned with building computational artifacts that understand, generate, or manipulate human languag
e. research on language began shortly after the invention of digital computers in the 1950s, and language
draws on both linguistics and artificial intellect. however, the major breakthroughs of the past few years
have been powered by machine learning, which is a branch of ai that develops systems that learn and genera
lize from data. deep learning is a kind of machine learning that can learn very complex patterns from larg
e datasets, which means that it is ideally suited to learning the complexities of natural language from da
tasets sourced from the web.'
```

In [4]:
```
# Зашифрую текст методом простой замены

from simple_cipher import SimpleCipher

encryptor = SimpleCipher()
encrypted_text = encryptor.encrypt(open_text)
encrypted_text
```

Out[4]:
```
'~DuxJDjdjD~}xD}WdrJz$WHHk~}dkHdDdSJD~$[dzedDJukek$kDjdk~uWjjk}W~$Wdu[DudW~DSjWHd$z%rxuWJHduzd$z%rJW[W~FT
d}W~WJDuWTdD~Fd%D~krxjDuWd[x%D~djD~}xD}W1d~DuxJDjdjD~}xD}WdrJz$WHHk~}d[DHdu[WdDSkjkuyduzdk~uWJJz}DuWdu[WdF
DuDd"ku[d~DuxJDjdjD~}xD}WduWEudzJdtzk$W1du[kHdkHdDjHzd$DjjWFdjD~}xD}Wdk~1d%zHud$z~Hx%WJHd[DtWdrJzSDSjydk~u
WJD$uWFd"ku[d~jrd"ku[zxudJWDjk.k~}dku1dezJdk~HuD~$WTd~jrdkHdu[Wd$zJWduW$[~zjz}ydSW[k~FdtkJuxDjdDHHkHuD~uHT
dHx$[dDHdu[WdzJD$jWdFk}kuDjdDHHkHuD~uTdHkJkTd$zJuD~DTdzJdDjWED1d"[W~d"WdDH{dVxWHukz~Hdzedu[WHWdtkJuxDjdDHH
kHuD~uHTd~jrdkHd"[DudW~DSjWHdu[W%duzd~zudz~jydx~FWJHuD~Fdu[WdxHWJHdJWVxWHuTdSxuduzdDjHzdJWHrz~Fdk~d~DuxJDj
djD~}xD}W1djD~}xD}WdDrrjkWHdSzu[duzd"JkuuW~duWEudD~FdHrWW$[TdD~Fd$D~dSWdDrrjkWFduzdDjjd[x%D~djD~}xD}WH1dzu
[WJdWED%rjWHdzeduzzjHdrz"WJWFdSyd~jrdk~$jxFWd"WSdHWDJ$[TdW%DkjdHrD%dekjuWJk~}TdDxuz%Duk$duJD~HjDukz~dzeduW
EudzJdHrWW$[TdFz$x%W~udHx%%DJk.Dukz~TdHW~uk%W~udD~DjyHkHTdD~Fd}JD%%DJdHrWjjd$[W${k~}1dezJdWED%rjWTdHz%WdW%
DkjdrJz}JD%Hd$D~dDxuz%Duk$DjjydHx}}WHudD~dDrrJzrJkDuWdJWrjyduzdDd%WHHD}WdSDHWFdz~dkuHd$z~uW~udXdu[WHWdrJz}
JD%HdxHWdjD~}xD}WduzdJWDFTdD~Djy.WTdD~FdJWHrz~FduzdyzxJd%WHHD}W1du[WJWdDJWdHWtWJDjdzu[WJduWJ%Hdu[DudDJWdJz
x}[jydHy~z~y%zxHd"ku[d~jr1d~DuxJDjdjD~}xD}Wdx~FWJHuD~Fk~}dD~Fd~DuxJDjdjD~}xD}Wd}W~WJDukz~dJWeWJduzdxHk~}d
$z%rxuWJHduzdx~FWJHuD~FdD~FdrJzFx$Wd[x%D~djD~}xD}WTdJWHrW$uktWjy1d~j}d[DHdu[WdDSkjkuyduzdrJztkFWdDdtWJSDjd
FWH$Jkrukz~dzed"[Dud[DHd[DrrW~WF1du[kHdkHdDjHzd$DjjWFdjD~}xD}WdzxudSydHx%%DJk.k~}dSyd%WD~k~}exjdk~ezJ%Dukz
~dk~uzduWEudxHk~}dDd$z~$Wrud{~z"~dDHd}JD%%DJdzed}JDr[k$H1dk~drJD$uk$WTd~jxdkHdxHWFduzd%WD~d~jr1du[Wdx~FWJH
uD~Fk~}dSyd$z%rxuWJHdzedu[WdHuJx$uxJWdD~Fd%WD~k~}dzedDjjd[x%D~djD~}xD}WHTdDjjz"k~}dFWtWjzrWJHdD~FdxHWJHduz
dk~uWJD$ud"ku[d$z%rxuWJHdxHk~}d~DuxJDjdHW~uW~$WHdD~Fd$z%%x~k$Dukz~1d$z%rxuDukz~Djdjk~}xkHuk$HdkHdu[WdH$kW~
ukek$dekWjFdu[DudHuxFkWHd$z%rxuDukz~DjdDHrW$uHdzed[x%D~djD~}xD}WTd"[kjWdjD~}xD}WdkHdu[WdW~}k~WWJk~}dFkH$kr
jk~Wd$z~$WJ~WFd"ku[dSxkjFk~}d$z%rxuDukz~DjdDJukeD$uHdu[Dudx~FWJHuD~FTd}W~WJDuWTdzJd%D~krxjDuWd[x%D~djD~}x
D}W1dJWHWDJ$[dz~djD~}xD}WdSW}D~dH[zJujydDeuWJdu[Wdk~tW~ukz~dzedFk}kuDjd$z%rxuWJHdk~du[WdCPgRHTdD~FdjD~}xD}
WdFJD"Hdz~dSzu[djk~}xkHuk$HdD~FdDJukek$kDjdk~uWjjW$u1d[z"WtWJTdu[Wd%DLzJdSJWD{u[Jzx}[Hdzedu[WdrDHudeW"dyWD
JHd[DtWdSWW~drz"WJWFdSyd%D$[k~WdjWDJ~k~}Td"[k$[dkHdDdSJD~$[dzedDkdu[DudFWtWjzrHdHyHuW%Hdu[DudjWDJ~dD~Fd}W~
WJDjk.WdeJz%dFDuD1dFWWrdjWDJ~k~}dkHdDd{k~Fdzed%D$[k~WdjWDJ~k~}du[Dud$D~djWDJ~dtWJyd$z%rjWEdrDuuWJ~HdeJz%dj
DJ}WdFDuDHWuHTd"[k$[d%WD~Hdu[DudkudkHdkFWDjjydHxkuWFduzdjWDJ~k~}du[Wd$z%rjWEkukWHdzed~DuxJDjdjD~}xD}WdeJz%
dFDuDHWuHdHzxJ$WFdeJz%du[Wd"WS1'
```

In [5]:
```
# Проанализирую частоту символов в шифртексте

statistic = {}

for i in encrypted_text:
    if statistic.get(i):
        statistic[i] += 1
    else:
        statistic[i] = 1

print(statistic)
```

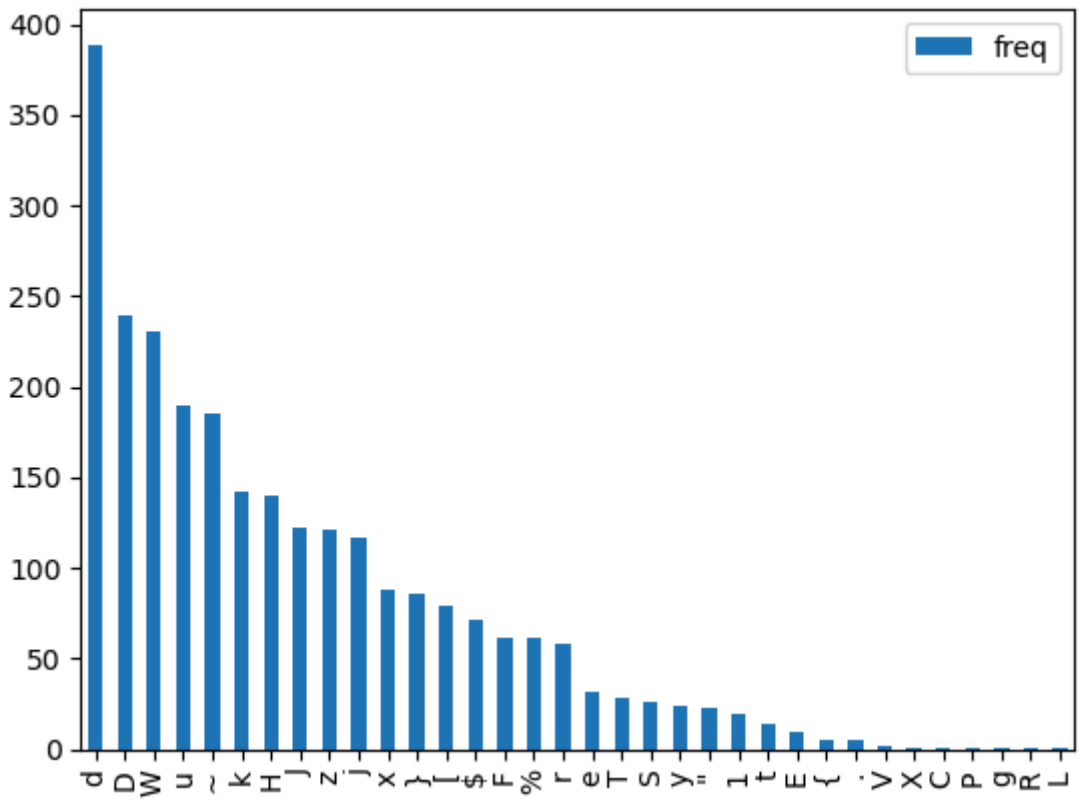
```
{'~': 185, 'D': 239, 'u': 190, 'x': 88, 'J': 122, 'j': 117, 'd': 389, '}': 86, 'W': 230, 'r': 58, 'z': 12
1, '$': 71, 'H': 140, 'k': 142, 'S': 26, '[': 79, 'e': 31, '%': 61, 'F': 61, 'T': 28, '1': 19, 'y': 24,
'"': 23, 'E': 9, 't': 14, '.': 5, '{': 5, 'V': 2, 'X': 1, 'C': 1, 'P': 1, 'g': 1, 'R': 1, 'L': 1}
```

In [6]:
```
# Составлю датафрейм и приведу к табличному виду
# Начертим на графике собранную статистику

import pandas as pd

df = pd.DataFrame.from_dict(statistic, orient='index', columns=['freq']).sort_values('freq', ascending=F
df.plot.bar()
```

Out[6]: <AxesSubplot: >



Частота символа d в шифртексте максимальная и совпадаетс частотой e в английском языке. То же самое можно сказать для следующих пар:

In [7]:
```
# Подбор ключа шифрования (таблицы замен) производится
# Методом восхождения в гору, где сначала подставляются
# элементы максимально приближенные по их частотности,
# и если в результате этого мы получаем текст похожий на
# осмысленный - оставляем кандидата, если нет - проложаем
# подбор
# Среди наиболее вероятных (по частотности) кандидатов
# я сопоставил следующие элементы, которых оказалось
# чтобы понять его суть зашифрованного сообщения, особенно
# если что-то смыслишь в языковых моделях машинного обучения

mapping = {
    'd': ' ', # наиболее распространенный не буквенный символ
    'D': 'a',
    'W': 'e',
    'u': 't',
    '~': 'n',
    'k': 'i',
    'H': 's',
    'J': 'r',
    'z': 'o',
    'j': 'l',
    'x': 'u',
    '}': 'g',
    '[': 'h',
    '$': 'c'
}
```

In [8]:
```
decrypted_array = []

for i in encrypted_text:
    if mapping.get(i):
        decrypted_array.append(mapping[i])
    else:
        decrypted_array.append('*')

result = "".join(decrypted_array)
result
```

Out[8]:
```
'natural language *rocessing is a *ranch o* arti*icial intelligence that ena*les co**uters to co**rehen**
generate* an* *ani*ulate hu*an language* natural language *rocessing has the a*ilit* to interrogate the *a
ta *ith natural language te*t or *oice* this is also calle* language in* *ost consu*ers ha*e *ro*a*l* inte
racte* *ith nl* *ithout reali*ing it* *or instance* nl* is the core technolog* *ehin* *irtual assistants*
such as the oracle *igital assistant* siri* cortana* or ale*a* *hen *e as* *uestions o* these *irtual assi
stants* nl* is *hat ena*les the* to not onl* un*erstan* the users re*uest* *ut to also res*on* in natural
language* language a**lies *oth to *ritten te*t an* s*eech* an* can *e a**lie* to all hu*an languages* oth
er e*a**les o* tools *o*ere* ** nl* inclu*e *e* search* e*ail s*a* *iltering* auto*atic translation o* te*
t or s*eech* *ocu*ent su**ari*ation* senti*ent anal*sis* an* gra**ar s*ell chec*ing* *or e*a**le* so*e e*a
il *rogra*s can auto*aticall* suggest an a**ro*riate re*l* to a *essage *ase* on its content * these *rogr
a*s use language to rea** anal**e* an* res*on* to *our *essage* there are se*eral other ter*s that are rou
ghl* s*non**ous *ith nl** natural language un*erstan*ing an* natural language generation re*er to using co
**uters to un*erstan* an* *ro*uce hu*an language* res*ecti*el** nlg has the a*ilit* to *ro*i*e a *er*al *e
scri*tion o* *hat has ha**ene** this is also calle* language out ** su**ari*ing ** *eaning*ul in*or*ation
into te*t using a conce*t *no*n as gra**ar o* gra*hics* in *ractice* nlu is use* to *ean nl** the un*ersta
n*ing ** co**uters o* the structure an* *eaning o* all hu*an languages* allo*ing *e*elo*ers an* users to i
nteract *ith co**uters using natural sentences an* co**unication* co**utational linguistics is the scienti
*ic *iel* that stu*ies co**utational as*ects o* hu*an language* *hile language is the engineering *isci*li
ne concerne* *ith *uil*ing co**utational arti*acts that un*erstan** generate* or *ani*ulate hu*an language
* research on language *egan shortl* a*ter the in*ention o* *igital co**uters in the ****s* an* language *
ra*s on *oth linguistics an* arti*icial intellect* ho*e*er* the *a*or *rea*throughs o* the *ast *e* *ears
ha*e *een *o*ere* ** *achine learning* *hich is a *ranch o* ai that *e*elo*s s*ste*s that learn an* genera
li*e *ro* *ata* *ee* learning is a *in* o* *achine learning that can learn *er* co**le* *atterns *ro* larg
e *atasets* *hich *eans that it is i*eall* suite* to learning the co**le*ities o* natural language *ro* *a
tasets source* *ro* the *e**'
```

Часть 2. Аффинный шифр

Аффинный шифр по сути ничем от шифра простой замены не отличается. Формула вычисления номера порядкового элемента в аффинном шифре - это по сути линейное выражение некой таблицы замен. Поскольку простая замена не имеет линейной комбинации, а является произвольной комбинацией любых двух элементов из одного словаря - то пространство ключей у шифра простой замены оказжется гораздо шире и равно **n!**, где n - это размерность словаря.

Аффинный шифр так же как и шифр простой замены переносит статистические характеристики языка на шифртекст. Давайте попробуем в этом убедиться и взломать тот же самый текст зашифрованный аффинным шифром с помощью частотного анализа

In [9]:
```
# Зашифрую текст методом аффинного шифрования

from simple_cipher import AffineCipher

encryptor = AffineCipher((8, 80))
encrypted_text = encryptor.encrypt(open_text)
encrypted_text
```

Out[9]:
```
'`=QYA=,+,=`HY=Hr+kAcbrIIX`H+XI+=+_A=`bP+cz+=AQXzXbX=,+X`Qr,,XHr`br+QP=Q+r`=_,rI+bc>kYQrAI+Qc+bc>kArPr`j +
Hr`rA=Qr +=`j+>=`XkY,=Qr+PY>=`+,=`HY=Hrp+`=QYA=,+,=`HY=Hr+kAcbrIIX`H+P=I+QPr+=_X,XQ?+Qc+X`QrAAcH=Qr+QPr+j=
Q=+%XQP+`=QYA=,+,=`HY=Hr+Qr-Q+cA+6cXbrp+QPXI+XI+=,Ic+b=,,rj+,=`HY=Hr+X`p+>cIQ+bc`IY>rAI+P=6r+kAc_=_,?+X`Qr
A=bQrj+%XQP+`,k+%XQPcYQ+Ar=,X{X`H+XQp+zcA+X`IQ=`br +`,k+XI+QPr+bcAr+QrbP`c,cH?+_rPX`j+6XAQY=,+=IIXIQ=`QI +
IYbP+=I+QPr+cA=b,r+jXHXQ=,+=IIXIQ=`Q +IXAX +bcAQ=`= +cA+=,r-=p+%Pr`+%r+=I$+sYrIQXc`I+cz+QPrIr+6XAQY=,+=IIX
IQ=`QI +`,k+XI+%P=Q+r`=_,rI+QPr>+Qc+`cQ+c`,?+Y`jrAIQ=`j+QPr+YIrAI+ArsYrIQ +_YQ+Qc+=,Ic+ArIkc`j+X`+`=QYA=,
+,=`HY=Hrp+,=`HY=Hr+=kk,XrI+_cQP+Qc+%AXQQr`+Qr-Q+=`j+IkrrbP +=`j+b=`+_r+=kk,Xrj+Qc+=,,+PY>=`+,=`HY=HrIp+cQ
PrA+r-=>k,rI+cz+Qcc,I+kc%rArj+_?+`,k+X`b,Yjr+%r_+Ir=AbP +r>=X,+Ik=>+zX,QrAX`H +=YQc>=QXb+QA=`I,=QXc`+cz+Qr
-Q+cA+IkrrbP +jcbY>r`Q+IY>>=AX{=QXc` +Ir`QX>r`Q+=`=,?IXI +=`j+HA=>>=A+Ikr,,+bPrb$X`Hp+zcA+r-=>k,r +Ic>r+r>
=X,+kAcHA=>I+b=`+=YQc>=QXb=,,?+IYHHrIQ+=`+=kkAckAX=Qr+Ark,?+Qc+=+>rII=Hr+_=Irj+c`+XQI+bc`Qr`Q+h+QPrIr+kAcH
A=>I+YIr+,=`HY=Hr+Qc+Ar=j +=`=,?{r +=`j+ArIkc`j+Qc+?cYA+>rII=Hrp+QPrAr+=Ar+Ir6rA=,+cQPrA+QrA>I+QP=Q+=Ar+Ac
YHP,?+I?`c`?>cYI+%XQP+`,kp+`=QYA=,+,=`HY=Hr+Y`jrAIQ=`jX`H+=`j+`=QYA=,+,=`HY=Hr+Hr`rA=QXc`+ArzrA+Qc+YIX`H+b
c>kYQrAI+Qc+Y`jrAIQ=`j+=`j+kAcjYbr+PY>=`+,=`HY=Hr +ArIkrbQX6r,?p+`,H+P=I+QPr+=_X,XQ?+Qc+kAc6Xjr+=+6rA_=,+j
rIbAXkQXc`+cz+%P=Q+P=I+P=kkr`rjp+QPXI+XI+=,Ic+b=,,rj+,=`HY=Hr+cYQ+_?+IY>>=AX{X`H+_?+>r=`X`HzY,+X`zcA>=QXc`
+X`Qc+Qr-Q+YIX`H+=+bc`brkQ+$`c%`+=I+HA=>>=A+cz+HA=kPXbIp+X`+kA=bQXbr +`,Y+XI+YIrj+Qc+>r=`+`,kp+QPr+Y`jrAIQ
=`jX`H+_?+bc>kYQrAI+cz+QPr+IQAYbQYAr+=`j+>r=`X`H+cz+=,,+PY>=`+,=`HY=HrI +=,,c%X`H+jr6r,ckrAI+=`j+YIrAI+Qc+
X`QrA=bQ+%XQP+bc>kYQrAI+YIX`H+`=QYA=,+Ir`Qr`brI+=`j+bc>>Y`Xb=QXc`p+bc>kYQ=QXc`=,+,X`HYXIQXbI+XI+QPr+IbXr`Q
XzXb+zXr,j+QP=Q+IQYjXrI+bc>kYQ=QXc`=,+=IkrbQI+cz+PY>=`+,=`HY=Hr +%PX,r+,=`HY=Hr+XI+QPr+r`HX`rrAX`H+jXIbXk,
X`r+bc`brA`rj+%XQP+_YX,jX`H+bc>kYQ=QXc`=,+=AQXz=bQI+QP=Q+Y`jrAIQ=`j +Hr`rA=Qr +cA+>=`XkY,=Qr+PY>=`+,=`HY=H
rp+ArIr=AbP+c`+,=`HY=Hr+_rH=`+IPcAQ,?+=zQrA+QPr+X`6r`QXc`+cz+jXHXQ=,+bc>kYQrAI+X`+QPr+D~9vI +=`j+,=`HY=Hr+
jA=%I+c`+_cQP+,X`HYXIQXbI+=`j+=AQXzXbX=,+X`Qr,,rbQp+Pc%r6rA +QPr+>=5cA+_Ar=$QPAcYHPI+cz+QPr+k=IQ+zr%+?r=AI
+P=6r+_rr`+kc%rArj+_?+>=bPX`r+,r=A`X`H +%PXbP+XI+=+_A=`bP+cz+=X+QP=Q+jr6r,ckI+I?IQr>I+QP=Q+,r=A`+=`j+Hr`rA
=,X{r+zAc>+j=Q=p+jrrk+,r=A`X`H+XI+=+$X`j+cz+>=bPX`r+,r=A`X`H+QP=Q+b=`+,r=A`+6rA?+bc>k,r-+k=QQrA`I+zAc>+,=A
Hr+j=Q=IrQI +%PXbP+>r=`I+QP=Q+XQ+XI+Xjr=,,?+IYXQrj+Qc+,r=A`X`H+QPr+bc>k,r-XQXrI+cz+`=QYA=,+,=`HY=Hr+zAc>+j
=Q=IrQI+IcYAbrj+zAc>+QPr+%r_p'
```

In [10]:
```
# Проанализирую частоту символов в шифртексте

statistic = {}

for i in encrypted_text:
    if statistic.get(i):
        statistic[i] += 1
    else:
        statistic[i] = 1

print(statistic)
```

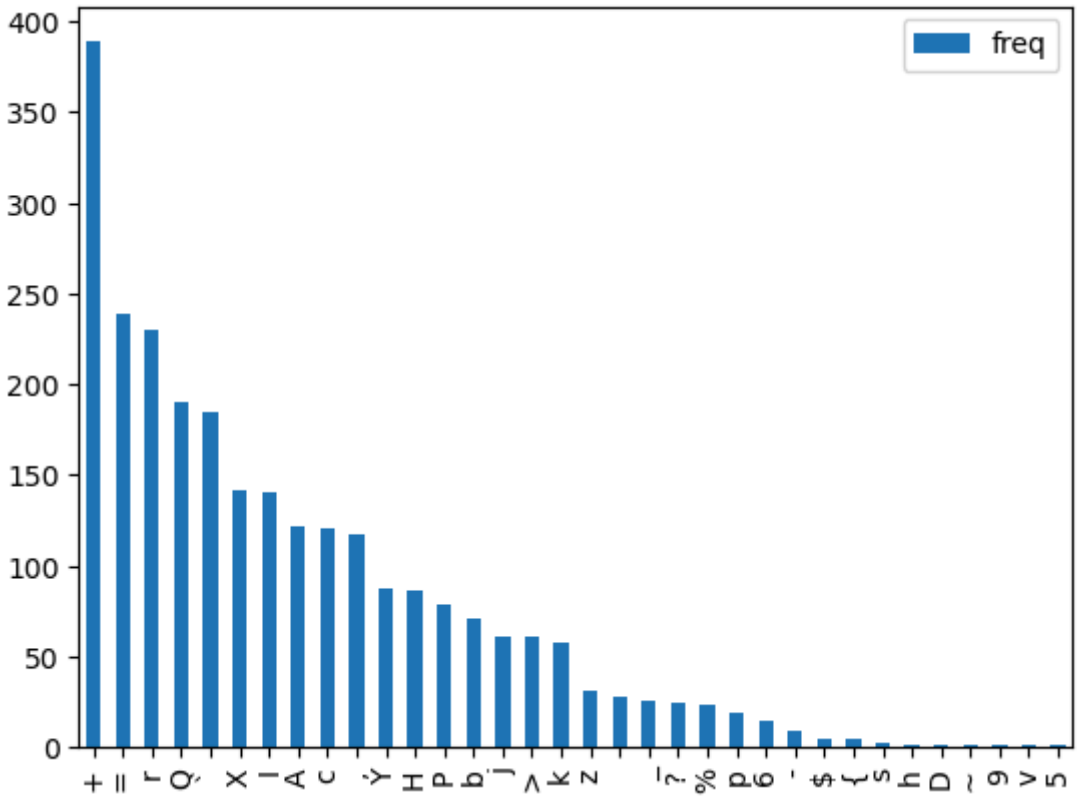
```
{'`': 185, '=': 239, 'Q': 190, 'Y': 88, 'A': 122, ',': 117, '+': 389, 'H': 86, 'r': 230, 'k': 58, 'c': 12
1, 'b': 71, 'I': 140, 'X': 142, '_': 26, 'P': 79, 'z': 31, '>': 61, 'j': 61, ' ': 28, 'p': 19, '?': 24,
'%': 23, '-': 9, '6': 14, '{': 5, '$': 5, 's': 2, 'h': 1, 'D': 1, '~': 1, '9': 1, 'v': 1, '5': 1}
```

In [11]:
```
# Составлю датафрейм и приведу к табличному виду
# Начертим на графике собранную статистику

import pandas as pd

df = pd.DataFrame.from_dict(statistic, orient='index', columns=['freq']).sort_values('freq', ascending=F
df.plot.bar()
```

Out[11]: <AxesSubplot: >



In [12]:
```
# Методом пристального взгляда и простого перебора
# кандидатов исходя из таблицы частотности
# попробуем взломать и аффинный метод шифрования

mapping = {
    '+': ' ', # наиболее распространенный не буквенный символ
    '=': 'a',
    'r': 'e',
    'Q': 't',
    '`': 'n',
    'X': 'i',
    'I': 's',
    'A': 'r',
    'c': 'o',
    ',': 'l',
    'Y': 'u',
    'H': 'g',
    'P': 'h',
    'b': 'c'
}
```

In [13]:
```
decrypted_array = []

for i in encrypted_text:
    if mapping.get(i):
        decrypted_array.append(mapping[i])
    else:
        decrypted_array.append('*')

result = "".join(decrypted_array)
result
```

Out[13]:
```
'natural language *rocessing is a *ranch o* arti*icial intelligence that ena*les co**uters to co**rehen**
generate* an* *ani*ulate hu*an language* natural language *rocessing has the a*ilit* to interrogate the *a
ta *ith natural language te*t or *oice* this is also calle* language in* *ost consu*ers ha*e *ro*a*l* inte
racte* *ith nl* *ithout reali*ing it* *or instance* nl* is the core technolog* *ehin* *irtual assistants*
such as the oracle *igital assistant* siri* cortana* or ale*a* *hen *e as* *uestions o* these *irtual assi
stants* nl* is *hat ena*les the* to not onl* un*erstan* the users re*uest* *ut to also res*on* in natural
language* language a**lies *oth to *ritten te*t an* s*eech* an* can *e a**lie* to all hu*an languages* oth
er e*a**les o* tools *o*ere* ** nl* inclu*e *e* search* e*ail s*a* *iltering* auto*atic translation o* te*
t or s*eech* *ocu*ent su**ari*ation* senti*ent anal*sis* an* gra**ar s*ell chec*ing* *or e*a**le* so*e e*a
il *rogra*s can auto*aticall* suggest an a**ro*riate re*l* to a *essage *ase* on its content * these *rogr
a*s use language to rea** anal**e* an* res*on* to *our *essage* there are se*eral other ter*s that are rou
ghl* s*non**ous *ith nl** natural language un*erstan*ing an* natural language generation re*er to using co
**uters to un*erstan* an* *ro*uce hu*an language* res*ecti*el** nlg has the a*ilit* to *ro*i*e a *er*al *e
scri*tion o* *hat has ha**ene** this is also calle* language out ** su**ari*ing ** *eaning*ul in*or*ation
into te*t using a conce*t *no*n as gra**ar o* gra*hics* in *ractice* nlu is use* to *ean nl** the un*ersta
n*ing ** co**uters o* the structure an* *eaning o* all hu*an languages* allo*ing *e*elo*ers an* users to i
nteract *ith co**uters using natural sentences an* co**unication* co**utational linguistics is the scienti
*ic *iel* that stu*ies co**utational as*ects o* hu*an language* *hile language is the engineering *isci*li
ne concerne* *ith *uil*ing co**utational arti*acts that un*erstan** generate* or *ani*ulate hu*an language
* research on language *egan shortl* a*ter the in*ention o* *igital co**uters in the ****s* an* language *
ra*s on *oth linguistics an* arti*icial intellect* ho*e*er* the *a*or *rea*throughs o* the *ast *e* *ears
ha*e *een *o*ere* ** *achine learning* *hich is a *ranch o* ai that *e*elo*s s*ste*s that learn an* genera
li*e *ro* *ata* *ee* learning is a *in* o* *achine learning that can learn *er* co**le* *atterns *ro* larg
e *atasets* *hich *eans that it is i*eall* suite* to learning the co**le*ities o* natural language *ro* *a
tasets source* *ro* the *e**'
```

Часть 3. Аффинный рекуррентный шифр

Поскольку аффинный рекуррентный шифр на каждом шаге вырабатывает новые ключи для шифрования - то один и тот же символ может быть зашифрован разными символами, и наоборот, два разных шифра могут быть зашифрованы одинаковыми символами.

Таким образом частотный анализ не принесет успеха в криптоанализе. Однако поскольку пространство ключей не столь велико и зависит по-сути от величины словаря, то данный шифр может быть взломан методом грубой силы - то есть простым перебором двух возможных вариантов ключей и при много кратном расшифровании зашифрованного текста - получить в итоге исходный текст.

Спасибо за внимание!