**Frequency Analysis Code**-

By this below frequency Python code I have tried to calculate frequency of letters, words and length of cipher:

print(' Substitution Cipher Code after decrypting:')

import pandas as pd

cipher = "type question here"

print('the length of the cipher is:',len(cipher))

inp = []

for i in cipher:

inp.append((i))

print('frequency analysis')

freq = {}

freq\_word = {}

length = 0

len = 0

for i in cipher:

if i in freq and i != " " and i != "-" and i != "." and i != "'":

freq[i] += 1

elif i not in freq and i != " " and i != "-" and i != "." and i != "'":

freq[i] = 1

length = length + 1

freq\_count = freq

print(sorted(freq\_count.items(), key = lambda kv:(kv[1], kv[0])))

# letter frequency calculation

for i in freq:

freq[i] = freq[i] / length \* 100

res = (cipher.split())

# word frequency calculation

for i in res:

if i in freq\_word:

freq\_word[i] += 1

else:

freq\_word[i] = 1

len = len + 1

# print of frequency letters

print(sorted(freq.items(), key = lambda kv:(kv[1], kv[0])))

# print of frequency words

print(sorted(freq\_word.items(), key = lambda kv:(kv[1], kv[0])))

Cipher – 1: Affine

C# code- ignored ‘’ in the code.

* We have total of 1789 letters.
* Here in this S has highest frequency which might be ‘e’ and V might be ‘t’. And there is a trigram VDS which might be “the” and a diagram VU which might be “to” and solve equation for alpha and beta.
* I solved for S:18(cipher) and e:4(plain) equation is cipher=plain\*alpha + beta.
* Another equation is U:20(cipher) and o:14(plain).
* Solve both equation you get alpha:21 and beta:14.

// C# program to illustrate Affine Cipher

using System;

using System.Text.RegularExpressions;

class AffineCipher

{

// Key values of alpha and beta

static int alpha = 21;

static int beta = 12;

static String decryptCipher(String cipher)

{

String decryptmsg = "";

int inverse = 0;

int count = 0;

//formula

for (int i = 0; i < 26; i++)

{

count = (alpha \* i) % 26;

// Check if (a\*i)%26 == 1,

// then i will be the multiplicative inverse of a

if (count == 1)

{

inverse = i;

}

}

for (int i = 0; i < cipher.Length; i++)

{

//trying to append special

if (cipher[i] == ' ')

{

decryptmsg += cipher[i];

}

else if(cipher[i] == '(')

{

decryptmsg = decryptmsg + '(';

}

else if(cipher[i] == ')')

{

decryptmsg+= ')';

}

else if(cipher[i] == '.')

{

decryptmsg+='.';

}

else //Decryption formula alpha^-1 (x - beta ) mod 26

// here x is cipher[i] and added 'A'

// A is added so for ASCII range

{

decryptmsg = decryptmsg + (char) (((inverse \*

((cipher[i] + 'A' - beta)) % 26)) + 'A');

}

}

return decryptmsg;

}

// Main code

public static void Main(String[] args)

{

String ciphermessage = "ADUFVSZYZI VDS GMFYZ NSHFQMFW ZYZSVSSZ NUQFVW VGU VDS ISFEMZA DYV HMCO HW YZVFUXQCYZI M ZSG NUQFVD GDSSJ (EQJVYPJWYZI VDS ZQEHSF UN ASVVYZIA MZUVDSF VGSZVW AYB VYESA) YZVU VDSYF ZMLMJ SZYIEM EMCDYZSA. VDS FSAQJVYZI 'ZSV' GMA OZUGZ VU VDS ISFEMZA MA 'VFYVUZ' MZX VU VDS HFYVYAD MA 'ADMFO'. NUF MJEUAV M WSMF HJSVCDJSW CUQJX EMOS ZU YZFUMXA YZVU ADMFO, MZX MJJYSX JUAASA YZ VDS MVJMZVYC MIMYZ YZCFSMASX MJMFEYZIJW.YZ XSCSEHSF ZYZSVSSZ NUQFVW VGU ADMFO GMA HFUOSZ, HQV ISFEMZ YZZULMVYUZA ESMZV VDMV VDS MJJYSA DMX VU GMYV QZVYJ MQIQAV VDS NUJJUGYZI WSMF HSNUFS ZMLMJ SZYIEM GMA FSIQJMFJW FSMX MIMYZ. HW VDSZ VDS MESFYCMZA GSFS MCVYLS CUEHMVMZVA, PFULYXYZI EQCD-ZSSXSX CUEPQVSF PUGSF VU HJSVCDJSW.HW X-XMW YZ TQZS ZYZSVSSZ NUQFVW NUQF QJVFM GMA ZU JUZISF AU YEPUFVMZV. HQV AVYJJ ZU UZS GMZVSX VDS ISFEMZA VU ASZAS VDMV SZYIEM GMA HSYZI FSMX. GDSZ, M NSG XMWA HSNUFS VDS ZUFEMZXW JMZXYZIA, MZ MESFYCMZ VMAO NUFCS CMPVQFSX M ISFEMZ Q-HUMV GYVD YVA SZYIEM OSWA, MXEYFMJ SFZSAV OYZI, QA CUEEMZXSF YZ CDYSN UN VDS MVJMZVYC NJSSV, VDFSMVSZSX VU CUQFV-EMFVYMJ VDS UNNYCSF YZ CDMFIS NUF SZXMZISFYZI 'UPSFMVYUZ ULSFJUFX', MA VDS PJMZ NUF VDS X-XMW JMZXYZIA GMA OZUGZ.HW DUG EQCD XYX QJVFM YZVSJJYISZCS, IMYZSX NFUE FSMXYZI SZYIEM CYPDSFA, ADUFVSZ VDS GMF? DMFFW DYZAJSW, HMASX MV HJSVCDJSW XQFYZI VDS GMF, AQIISAVA YV GMA M AYIZYNYCMZV MAASV. YN YV XYX ZUV OSSP FUEESJ UQV UN SIWPV YZ ZYZSVSSZ NUQFVW UZS, YV CSFVMYZJW XYX AU VDS NUJJUGYZI WSMF, HW PFSLSZVYZI DYE SBPJUYVYZI DYA LYCVUFW MV IMRMJM.MA ISZSFMJ MJSBMZXSF PQV YV, 'VDS OZUGJSXIS ZUV UZJW UN VDS SZSEW'A PFSCYAS AVFSZIVD MZX XYAPUAYVYUZ, HQV MJAU DUG, GDSZ MZX GDSFS DS YZVSZXA VU CMFFW UQV DYA UPSFMVYUZA HFUQIDV M ZSG XYESZAYUZ VU VDS PFUASCQVYUZ UN VDS GMF.'VDS JUAA UN SIWPV YZ ZYZSVSSZ NUQFVW VGU GUQJX DMLS ASV HMCO VDS FS-CUZKQSAV UN ZUFVD MNFYCM MZX QPASV VDS VYESVMHJS NUF VDS YZLMAYUZ UN NFMZCS. MCCUFXYZI VU DYZAJSW, ULSFJUFX GUQJX PFUHMHJW DMLS HSSZ XSNSFFSX QZVYJ ZYZSVSSZ NUQFVW AYB.HQV HW VDSZ VDS ISFEMZA EYIDV DMLS DYV HMCO GYVD L-GSMPUZA MZX GUFAS. SZYIEM AQCCSAASA MJGMWA ZSSXSX CUEPJSESZVYZI GYVD UVDSF YZVSJJYISZCS EMVSFYMJ, HQV VDS NMCV VDMV VDS MJJYSA OSPV SZYIEM ASCFSV QZVYJ ZYZSVSSZ ASLSZVW NUQF ADUGA DUG EQCD YV ESMZV VU VDSE.";

//message without special character

string msgwospecialchar= Regex.Replace(ciphermessage, "[^a-zA-Z0-9\_().]+", " ");

String cipherText = decryptCipher(msgwospecialchar);

//Decrypted message without special characters like([ ]( ),.-,!@#$%^&\*)

Console.WriteLine("Decrypted Message is : " + cipherText.ToLower());

}

}

**The final output is:** shortening the warin february nineteen fourty two the germans hit back by introducing a new fourth wheel (multiplying the number of settings another twenty six times) into their naval enigma machines. the resulting 'net' was known to the germans as 'triton' and to the british as 'shark'. for almost a year bletchley could make no inroads into shark, and allied losses in the atlantic again increased alarmingly.in december nineteen fourty two shark was broken, but german innovations meant that the allies had to wait until august the following year before naval enigma was regularly read again. by then the americans were active combatants, providing much-needed computer power to bletchley.by d-day in june nineteen fourty four ultra was no longer so important. but still no one wanted the germans to sense that enigma was being read. when, a few days before the normandy landings, an american task force captured a german u-boat with its enigma keys, admiral ernest king, us commander in chief of the atlantic fleet, threatened to court-martial the officer in charge for endangering 'operation overlord', as the plan for the d-day landings was known.by how much did ultra intelligence, gained from reading enigma ciphers, shorten the war? harry hinsley, based at bletchley during the war, suggests it was a significant asset. if it did not keep rommel out of egypt in nineteen fourty one, it certainly did so the following year, by preventing him exploiting his victory at gazala.as general alexander put it, 'the knowledge not only of the enemy's precise strength and disposition, but also how, when and where he intends to carry out his operations brought a new dimension to the prosecution of the war.'the loss of egypt in nineteen fourty two would have set back the re-conquest of north africa and upset the timetable for the invasion of france. according to hinsley, overlord would probably have been deferred until nineteen fourty six.but by then the germans might have hit back with v-weapons and worse. enigma successes always needed complementing with other intelligence material, but the fact that the allies kept enigma secret until nineteen seventy four shows how much it meant to them.

The key after substituting-

['A','s'],['B','x'],['C','c'],['D','h'],['E','m'],['F','r'],['G','w'],['H','b'],['I','g'],['J','l'],['K','q'],['L','v'],['M','a'],['N','f'],['O','k'],['P','p'],['Q','u'],['R','z'],['S','e'],['T','j'],['U','o'],['V','t'],['W','y'],['X','d'],['Y','i'],['Z','n'].

Cipher-2 : Substitution Cipher

Python code- For Decrypting

print(' Substitution Cipher Code after decrypting:')

import pandas as pd

cipher = "ETR YRLNWX BXORPEBYWEFLP TFNRK BX FX ETR UBYVLR FU ETBP NSPERLBFVP LRMLVBERL, GVE XRBETRL YBVPRDDR PERLWXEBXF XFL WXS FU ETR FETRLP HXRA ATF TR AWP. WCC EARXES-SRWL-FCK RXRW KR NBMTBRC HXRA AWP ETWE 'YRFLYBF' VPRK EF YF KFAX KWBCS EF ETR OBCCWYR FU WLMWXYRCF EF GVS EFNWEFRP, FCBORP WXK NRCFXP. MTWNBMF JVBLBMF EFF TWK YFXR BXEF ETR TBCCP WUERL TR NRE EAF BEWCBWX PFCKBRLP BX ETR UBRCKP FU WLMWXYRCF ATF DLFNBPRK TBN UFFK. NWLH NWIFARL: BXPBKR TBECRL'P YLRRMR."

print('the length of the cipher is:',len(cipher))

inp = []

for i in cipher:

inp.append((i))

mapping = pd.DataFrame([['\_','\_'],['-','-'],[',',','],[';',';'],[':',':'],['?','?'],[' ',' '],["'","'"],['.','.'],['"','"'],['(','('],[')',')'],['@','@'],['\*','\*'],['/','/'],['&','&'],['+','+'],['<','<'],['>','>'],['W','a'],['G','b'],['M','c'],['K','d'],['R','e'],['U','f'],['Y','g'],['T','h'],['B','i'],['q','j'],['H','k'],['C','l'],['N','m'],['X','n'],['F','o'],['D','p'],['J','q'],['L','r'],['P','s'],['E','t'],['V','u'],['O','v'],['A','w'],['j','x'],['S','y'],['I','z']],columns = ['Cipher','Plain'])

plaintext\_df = pd.DataFrame(inp, columns = ['Cipher'])

plaintext\_df = pd.merge(plaintext\_df, mapping, on = ['Cipher'], how = 'left').reset\_index(drop = True)

plain\_text = []

plaintext\_df['Plain'] = plaintext\_df['Plain'].astype(str)

plain\_text = plaintext\_df['Plain'].sum()

print(plain\_text)

* The first word is ETR which might be ‘the’ / ‘are’. By frequency analysis R might ‘e’ so the word is ‘the’.
* ETWE might be ‘that’ so W is ‘a’.
* WCC might be ‘all’.
* EF might be ‘to’.
* ATF might be ‘who’.
* FETRLP might be ‘others’.
* HXRA might be ‘knew’ as it has followed by \_ \_ ew who he was.
* There are total 371 letters in the message.

The letters are: 'W','a'],['G','b'],['M','c'],['K','d'],['R','e'],['U','f'],['Y','g'],['T','h'],['B','i'],['','j'],['H','k'],['C','l'],['N','m'],['X','n'],['F','o'],['D','p'],['J','q'],['L','r'],['P','s'],['E','t'],['V','u'],['O','v'],['A','w'],['','x'],['S','y'],['I','z']

After substituting all the words the final decrypting message is –

the german investigators homed in on the figure of this mysterious recruiter, but neither giuseppe sterantino nor any of the others knew who he was. all twenty-year-old enea de michiel knew was that 'georgio' used to go down daily to the village of arcangelo to buy tomatoes, olives and melons. chamico quirico too had gone into the hills after he met two italian soldiers in the fields of arcangelo who promised him food.

mark mazower: inside hitler's greece.

Cipher-5 : Vigenere Cipher

* By shifting each letter in the message I tried to calculate the key value length and found that for every five shifts I am getting a bigger value of shift in the key.
* So break the cipher into 5 groups each group containing 99,99,99,98,98.
* In group one the highest frequency letter is H which might be ‘e’ so the shift would be ‘D’. The first letter key is 03 or “D” and verified bigrams and trigrams and fixed it to be ‘D’.
* In group second the highest frequency letter is E which might be ‘e’ so the shift would be ‘A’. The second letter key is 00 or “A” and verified bigrams and trigrams and fixed it to be ‘A’.
* In group third the highest frequency letter is R which might be ‘e’ so the shift would be ‘N’. The third letter key is 17 or “N” and verified bigrams and trigrams and fixed it to be ‘N’.
* In group fourth the highest frequency letter is H which might be ‘e’ but it is not working so the next letter might be t also not working after word guessing and substituting remaining values we got to know that ‘P’ is ‘w’. So the key is 19 or “T”.
* The fifth letter key is 05 or “E”.

KEY : “DANTE”

Python code for Decrypting-

string = input( 'enter string' ) # "ABCSATLMNSLKJMOP"

# key from manula trying

key="DANTE"

def getinputString( char ):

return char if char.isalpha() else ''

bigString = ''.join( [ getinputString( char ) for char in string ] )

stringArray = list( bigString )

# Diving 5 groups based on key

def getCharGroups( key ):

smllChars = "abcdefghijklmnopqrstuvwxyz"

bigChars = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

bigCharAddrss = { char: index for index, char in enumerate( bigChars ) }

decryptedres = []

for index, keyChar in enumerate( key ):

group = {}

index = bigCharAddrss[ keyChar ]

for smallCharIndex, smallChar in enumerate( smllChars ):

if index >= len( bigChars ): index = 0

group[ bigChars[ index ] ] = smallChar

index += 1

decryptedres.append( group )

return decryptedres

charGroups = getCharGroups( key )

for index in range( 5 ):

group = charGroups[ index ]

crntIndex = index

while( crntIndex < len( bigString ) ):

stringArray[ crntIndex ] = group.get( stringArray[ crntIndex ] ) or stringArray[ crntIndex ]

crntIndex += 5

outputString = ''.join( stringArray )

stringIndex = 0

decryptedres = []

for stringCharacter in string:

if not stringCharacter.isalpha():

decryptedres.append( stringCharacter )

continue

decryptedres.append( stringArray[ stringIndex] )

stringIndex += 1

# final output

print('The Decrypted answer is:')

print( ''.join( decryptedres ) )

The final output is :

i saw one of them drawing forward to embrace me with so great affection, that it moved me to do the like. o shades, empty save in aspect! three times i clasped my hands behind it, and as often returned with them unto my breast. with wonder, i believe, i painted me; whereat the shade smiled and drew back, and i, following it, pressed forward. gently it said, that i should pause; then i knew who it was, and i prayed it that it would stay to speak with me a little. it replied to me: "even as i loved theein the mortal body, so loosed from it i love thee;therefore i stay; but wherefore art thou going?"dante's the divine comedy - purgatory.

Cipher-6 : Hill Cipher :

Input - LM QE TX YE AG TX CT UI EW NC TX LZ EW UA

IS PZ YV AP EW LM GQ WY AX FT CJ MS QC AD AG TX LM DX NX SN PJ QS YV AP RI QS MH NO CV AX FV.

Key is two so try to break into two so here the highest repeated is tx(4 times) so according to diagrams tx might be th