

Part A (soft copy of program)

from collections import Counter

initializing strings

cipherText = "bt jpx rmlx pcuv amlx icvjp ibtwxvr ci m lmt'r pmtn, mtn yvcjx cdv mwmbrtrj jpx amtngxrbah uqctjpx qgmrxv
ci jpx ymgg ci jpx hbtw'r qmgmax; mtn jpx hbtw rmy jpx qmvj ci jpx pmtn jpmj yvcjx.jpvt jpx hbtw'r acutjxtmtax ymr
apmtwxn, mtn pbr jpcuwpjr jvcufgx pbl, rc jpmj jpx scbtjr ci pbrgcbtr yxvx gccrxn, mtn pbr htxxr rlcjx ctx mwmbrtrj mtcjpxv.
jpx hbtw avbxn mgcun jc fvbwtv bt jpxmrjvcgcwxvr, jpx apmgnxmtr, mtn jpx rccjprmexvr. mtn jpx hbtw rqmhx, mtn rmbn jc
jpx ybrx lxt ci fmfeget, ypcrcxdv rpmgg vxmn jbr yvbjbtw, mtn rpy lx jpx btjxvqvjmjbct jpxvxci, rpmgg fxagcpxn ybjp
ramvgxj, mtn pmdx m apmbt ci wcn mfcuj pbr txah, mtn rpmgg fx jpx jpbvn vugxvbt jpx hbtwncl. jpxt amlx bt mgg jpx
hbtw'r ybrx lxt; fuj jpxe acugn tcj vxmn jpx yvbjbtw, tcv lmhxhtcyt jc jpx hbtw jpx btjxvqvjmjbct jpxvxci. jpxt ymr hbtw
fxgrpmoomv wvxmjge jvcufgx, mtnpbr acutjxtmtax ymr apmtwxn bt pbl, mtn pbr gcvnr yxvx mrjctbrpxn. tcy jpx kuxxt, fe
vxmrct cijpx ycvnr ci jpx hbtw mtn pbr gcvnr, amlx btjc jpx fmkuxj pcurx; mtn jpx kuxxt rqmhx mtn rmbn, chbtw, gbdx
icvxdxv; gxj tcj jpe jpcuwpjr jvcufgx jpxx, tcv gxj jpe acutjxtmtax fx apmtwxn; jpxvx brm lmt bt jpe hbtwncl, bt ypcl br jpx
rqbvbj ci jpx pcge wcnr; mtn bt jpx nmer ci jpe ybrncl ci jpxwcnr, ymr icutn bt pbl; ypcl jpx hbtw txfuapmntxoomv jpe
imjpxv, jpx hbtw, b rme, jpe imjpxv,lmnx lmrjxv ci jpx lmwbabmtr, mrjvcgcwxvr, apmgnxmtr, mtn rccjprmexvr; icvmrluap mr
mtxaxggxtj rqbvbj, mtn htcygxnw, mtn utnxvrjmtnbtw, btjxvqvjbtw ci nvxmlr, mtn rpybtw cipmvn rxtjtaxr, mtn
nbrrcgdbtw ci ncufjr,yxvx icutn bt jpx rmlx nmtbxg, ypcl jpx hbtw tmlxnfjgxrpmoomv; tcy gxj nmtbxg fx amggxn, mtn px
ybgg rpy jpx btjxvqvjmjbct."

hamletText = "to be, or not to be, that is the question: whether tis nobler in the mind to suffer the slings and arrows of
outrageous fortune, or to take arms against a sea of troubles and by opposing end them. to die-to sleep, no more; and by a
sleep to say we end the heart-ache and the thousand natural shocks that flesh is heir to: 'tis a consummation devoutly to be
wish'd. to die, to sleep; to sleep, perchance to dream-ay, there's the rub: for in that sleep of death what dreams may come,
when we have shuffled off this mortal coil, must give us pause-there's the respect that makes calamity of so long life. for
who would bear the whips and scorns of time, th'oppressor's wrong, the proud man's contumely, the pangs of dispriz'd
love, the law's delay, the insolence of office, and the spurns that patient merit of th'unworthy takes, when he himself might
his quietus make with a bare bodkin? who would fardels bear, to grunt and sweat under a weary life, but that the dread of
something after death, the undiscover'd country, from whose bourn no traveller returns, puzzles the will, and makes us
rather bear those ills we have than fly to others that we know not of? thus conscience does make cowards of us all, and
thus the native hue of resolution is sicklied o'er with the pale cast of thought, and enterprises of great pitch and moment
with this regard their currents turn awry and lose the name of action."

shylockText = "to bait fish withal. if it will feed nothing else, it will feed my revenge. he hath disgraced me and hindered me
half a million, laughed at my losses, mocked at my gains, scorned my nation, thwarted my bargains, cooled my friends,
heated mine enemies-and what's his reason? i am a jew. hath not a jew eyes? hath not a jew hands, organs, dimensions,
senses, affections, passions? fed with the same food, hurt with the same weapons, subject to the same diseases, healed by
the same means, warmed and cooled by the same winter and summer as a christian is? if you prick us, do we not bleed? if
you tickle us, do we not laugh? if you poison us, do we not die? and if you wrong us, shall we not revenge? if we are like you
in the rest, we will resemble you in that. if a jew wrong a christian, what is his humility? revenge. if a christian wrong a jew,
what should his sufferance be by christian example? why, revenge. the villainy you teach me i will execute-and it shall go
hard but i will better the instruction."

PART B

run frequency analysis on Hamlet's speech and Shylock's speech

hamlet = Counter(hamletText)

hamlet = str(hamlet)

hamlet = hamlet[8:]

hamlet2 = Counter(hamletText[idx : idx + 2] for idx in range(len(hamletText) - 1))

hamlet3 = Counter(hamletText[idx : idx + 3] for idx in range(len(hamletText) - 1))

shylock = Counter(shylockText)

shylock = str(shylock)

shylock = shylock[8:]

shylock2 = Counter(shylockText[idx : idx + 2] for idx in range(len(shylockText) - 1))

shylock3 = Counter(shylockText[idx : idx + 3] for idx in range(len(shylockText) - 1))

get only the first 30 elements from bigram and trigram counts

```

hamlet2 = str(hamlet2)
hamlet2 = hamlet2[8 : 307]
hamlet3 = str(hamlet3)
hamlet3 = hamlet3[8 : 324]
shylock2 = str(shylock2)
shylock2 = shylock2[8 : 307]
shylock3 = str(shylock3)
shylock3 = shylock3[8 : 324]

print("Part B: Frequency Analysis for famous Shakespearean speeches\n")
# print single letter frequencies for hamletText and shylockText
print("Single letter frequencies:")
print("Hamlet Speech: " + str(hamlet) + "\n")
print("Shylock Speech: " + str(shylock) + "\n\n-----\n")

# print bigram frequencies for hamletText and shylockText
print("Top 30 bigram frequencies:")
print("Hamlet Speech: " + str(hamlet2) + "\n")
print("Shylock Speech: " + str(shylock2) + "\n\n-----\n")

# print trigram frequencies for hamletText and shylockText
print("Top 30 trigram frequencies:")
print("Hamlet Speech: " + str(hamlet3) + "\n")
print("Shylock Speech: " + str(shylock3) + "\n\n-----\n")

# PART C #####
print("Part C: Comparison of the two above outputs\nOf the top 5 most frequent single letters in each of the
Shakespearean speeches, 4 are the same in both (e, a, s, and t). Of the top 5 most infrequent single letters (including letters
with a count of 0 which are not shown), again, 4 are the same in both (q, z, v, and k). Based on these similarities, I would say
that frequency analysis performed in this way produces fairly accurate and consistent results. Of course there is variation
because every text is different, but this variation diminishes the more text is being analyzed. I imagine if I were analyzing
entire Shakespearean plays rather than excerpts, the results would be even more similar.\n\n-----\n")

# PART D #####
# gather counts of single letters, bigrams, and trigrams in cipherText
cipher = Counter(cipherText)
cipher = str(cipher)
cipher = cipher[8:]
cipher2 = Counter(cipherText[idx : idx + 2] for idx in range(len(cipherText) - 1))
cipher3 = Counter(cipherText[idx : idx + 3] for idx in range(len(cipherText) - 1))

# get only the first 30 elements from bigram and trigram counts
cipher2 = str(cipher2)
cipher2 = cipher2[8 : 307]
cipher3 = str(cipher3)
cipher3 = cipher3[8 : 324]

print("Part D: Frequency Analysis for Cipher Text and Decoded Cipher")
# print single letter frequencies for cipherText
print("Single letter frequencies:")
print("Cipher Text: " + str(cipher) + "\n")

```

```
# print bigram frequencies for cipherText
print("Top 30 bigram frequencies:")
print("Cipher Text: " + str(cipher2) + "\n")
```

```
def decoder(cipher):  
    code = {  
        "x": "e",  
        "t": "n",  
        "j": "t",  
        "m": "a",  
        "p": "h",  
        "c": "o",  
        "r": "s",  
        "b": "i",  
        "n": "d",  
        "v": "r",  
        "g": "l",  
        "w": "g",  
        "y": "w",  
        "a": "c",  
        "l": "m",  
        "i": "f",  
        "u": "u",  
        "h": "k",  
        "f": "b",  
        "e": "y",  
        "q": "p",  
        "d": "v",  
        "o": "z",  
        "k": "q",  
        "s": "j",  
        "z": "x",  
        ".": ".",  
        ",": ",",  
        ";": ";",  
        ":": ":",  
        "'": "'",  
        '"': '"',  
        "<": "<",  
        ">": ">"  
    }  
  
    return code.get(cipher, "nothing")
```

```
for i in cipherText:
    decodedLetter = decoder(i)
    decodedCipher += decodedLetter
```

Part B: Frequency Analysis for famous Shakespearean speeches

Single letter frequencies:

Hamlet Speech: {' ': 255, 'e': 137, 't': 117, 'o': 93, 'a': 84, 's': 82, 'h': 75, 'r': 68, 'n': 65, 'i': 53, 'd': 42, 'l': 41, 'u': 40, 'f': 34, 'w': 28, 'm': 28, 'c': 24, ',': 24, 'p': 22, 'b': 15, 'g': 14, 'y': 14, '": 12, 'k': 10, 'v': 8, '-': 4, ':': 4, 'z': 3, '': 3, 'q': 2, ';': 2, '?': 2})

Shylock Speech: {' ': 191, 'e': 102, 'i': 67, 'a': 65, 's': 59, 't': 56, 'n': 53, 'h': 47, 'o': 44, 'd': 34, 'l': 34, 'r': 31, 'w': 30, 'm': 27, ',': 24, 'u': 22, 'y': 20, 'f': 19, 'c': 17, 'g': 15, 'b': 11, '?': 10, ':': 7, 'j': 6, 'p': 5, 'v': 5, 'k': 4, '-': 2, 'x': 2, '": 1})

Top 30 bigram frequencies:

Hamlet Speech: {' t': 59, 'th': 50, 'e ': 43, 's ': 40, 'he': 33, 'd ': 25, ' a': 24, ', ': 24, ' o': 23, 't ': 22, ' w': 21, 'o ': 19, 'nd': 18, 'an': 17, 'at': 17, 'of': 17, ' s': 17, 'r ': 16, 'f ': 16, 're': 15, 'er': 15, 'to': 15, ' m': 13, 'le': 13, 'ea': 13, 'ha': 12, 'is': 12, ' b': 12, 'es': 12, 'ar': 11

Shylock Speech: {'e ': 30, ' w': 24, ', ': 24, 'd ': 22, ' i': 19, ' a': 19, 't ': 17, 'th': 17, 's,': 16, 'ed': 15, ' h': 15, 'an': 14, ' m': 13, 'ha': 13, ' t': 13, ' s': 13, 'is': 13, 'me': 12, 'on': 12, 'he': 12, 'ns': 11, 'at': 11, 'y ': 10, 'in': 10, '? ': 10, 'nd': 9, 'wi': 9, 're': 9, 'a ': 9, ' b': 9, 'f

Top 30 trigram frequencies:

Hamlet Speech: {' th': 39, 'the': 26, 'he ': 21, ' of': 17, 'nd ': 16, 'of ': 14, 'to ': 14, ' to': 13, 'and': 13, ' an': 12, ', t': 10, 'at ': 10, 's t': 9, 'e, ': 9, 'tha': 8, 'hat': 8, 'is ': 8, 'er ': 8, ' wh': 8, 'd t': 7, ' co': 7, 'us ': 7, 'ake': 6, 'sl': 6, 'ma': 6, 's a': 6, ' be': 6, 'es ': 6, 'r t': 6, 'f t': 5, 'ee

Shylock Speech: {'s, ': 16, 'ed ': 14, ' th': 10, 'he ': 9, ' a ': 9, ' wi': 9, 'me ': 8, ' if': 8, 'if ': 8, 'the': 8, 'll ': 7, 'not': 7, 'you': 7, ' we': 7, 'd m': 7, 'ill': 7, ' no': 7, ' re': 7, ' yo': 7, 'and': 7, 'ou ': 7, 'hat': 7, 'ns,': 7, ' ha': 6, 'ot ': 6, 'ion': 6, 'we ': 6, 'nd ': 6, 'my ': 6, ' my': 6, 'jew': 5, '?'

Part C: Comparison of the two above outputs

Of the top 5 most frequent single letters in each of the Shakespearean speeches, 4 are the same in both (e, a, s, and t). Of the top 5 most infrequent single letters (including letters with a count of 0 which are not shown), again, 4 are the same in both (q, z, v, and k). Based on these similarities, I would say that frequency analysis performed in this way produces fairly accurate and consistent results. Of course there is variation because every text is different, but this variation diminishes the more text is being analyzed. I imagine if I were analyzing entire Shakespearean plays rather than excerpts, the results would be even more similar.

Part D: Frequency Analysis for Cipher Text and Decoded Cipher

Single letter frequencies:

Cipher Text: {' ': 284, 'x': 161, 't': 126, 'j': 123, 'm': 114, 'p': 107, 'c': 96, 'r': 87, 'b': 83, 'n': 68, 'v': 64, 'g': 49, 'w': 38, ',': 37, 'y': 30, 'a': 29, 'l': 27, 'i': 26, 'u': 25, 'h': 23, 'f': 18, 'e': 15, 'q': 12, ';': 9, ':': 7, 'd': 6, 'o': 6, '": 4, 'k': 3, 'z': 1, 's': 1})

Top 30 bigram frequencies:

Cipher Text: {'jp': 66, 'x ': 63, ' j': 61, 'px': 52, 'bt': 43, 'mt': 42, 'n ': 36, ', ': 35, ' m': 34, 'tn': 29, 'r ': 26, 'tw': 26, 'xv': 25, ' r': 22, 't ': 21, ' y': 21, ' c': 19, 'ci': 18, 'pm': 18, ' p': 16, ' a': 16, 'vx': 16, ' b': 16, ' h': 16, 'cu': 16, 'br': 15, 'hb': 15, ' j': 15, 'jx': 14, 'i ': 14

Top 30 trigram frequencies:

Cipher Text: {' jp': 55, 'jpx': 50, 'px ': 37, 'mtn': 26, ' mt': 24, 'tn ': 24, 'btw': 23, ', m': 17, ' ci': 16, 'hbt': 15, ' hb': 14, 'ci ': 14, ' bt': 13, 'x h': 12, 'tw ': 12, 'bt ': 11, 't j': 10, 'r, ': 9, ' pb': 9, 'n j': 9, 'pbr': 8, 'br ': 8, 'i j': 8, 'tjx': 8, 'n p': 7, 'n r': 7, 'apm': 7, 'x y': 7, 'n, ': 6, 'mgg': 6

Decoded Cipher:

in the same hour came forth fingers of a man's hand, and wrote over against the candlestick upon the plaster of the wall of the king's palace; and the king saw the part of the hand that wrote. then the king's countenance was changed, and his thoughts troubled him, so that the joints of his loins were loosed, and his knees smote one against another. the king cried aloud to bring in the astrologers, the chaldeans, and the soothsayers. and the king spake, and said to the wise men of babylon, whosoever shall read this writing, and show me the interpretation thereof, shall be clothed with scarlet, and have a

chain of gold about his neck, and shall be the third ruler in the kingdom. then came in all the king's wise men; but they could not read the writing, nor make known to the king the interpretation thereof. then was king belshazzar greatly troubled, and his countenance was changed in him, and his lords were astonished. now the queen, by reason of the words of the king and his lords, came into the banquet house; and the queen spake and said, O king, live forever; let not thy thoughts trouble thee, nor let thy countenance be changed; there is a man in thy kingdom, in whom is the spirit of the holy gods; and in the days of thy wisdom of the gods, was found in him; whom the king nebuchadnezzar thy father, the king, I say, thy father, made master of the magicians, astrologers, chaldeans, and soothsayers; forasmuch as an excellent spirit, and knowledge, and understanding, interpreting of dreams, and showing of hard sentences, and dissolving of doubts, were found in the same daniel, whom the king named belteshazzar; now let daniel be called, and he will show the interpretation.