In [310]:

```
# Initialize OK
from client.api.notebook import Notebook
ok = Notebook('project3.ok')
```

```
=====================================================================
Assignment: Project 3: Movie Classification
OK, version v1.12.5
=====================================================================
```

Project 3: Movie Classification

Welcome to the third project of Data 8! You will build a classifier that guesses whether a movie is romance or action, using only the number of times words appear in the movies's screenplay. By the end of the project, you should know how to:

1. Build a k-nearest-neighbors classifier.
2. Test a classifier on data.

Logistics

Deadline. This project is due at 11:59pm on Friday 12/06. You can earn an early submission bonus point by submitting your completed project by 11:59 on Thursday 12/05. It's **much** better to be early than late, so start working now.

Checkpoint. For full credit, you must also **complete Part 2 of the project (out of 4) and submit it by 11:59pm on Friday 11/22**. You will not have lab time to work on these questions, so we recommend that you start early on each part to stay on track.

Partners. You may work with one other partner; this partner **must** be enrolled in the same lab section as you are. Only one of you is required to submit the project. On okpy.org, the person who submits should also designate their partner so that both of you receive credit.

Rules. Don't share your code with anybody but your partner. You are welcome to discuss questions with other students, but don't share the answers. The experience of solving the problems in this project will prepare you for exams (and life). If someone asks you for the answer, resist! Instead, you can demonstrate how you would solve a similar problem.

Support. You are not alone! Come to office hours, post on Piazza, and talk to your classmates. If you want to ask about the details of your solution to a problem, make a private Piazza post and the staff will respond. If you're ever feeling overwhelmed or don't know how to make progress, email your TA or tutor for help. You can find contact information for the staff on the course website.

Tests. Passing the tests for a question **does not** mean that you answered the question correctly. Tests usually only check that your table has the correct column labels. However, more tests will be applied to verify the correctness of your submission in order to assign your final score, so be careful and check your work!

Advice. Develop your answers incrementally. To perform a complicated table manipulation, break it up into steps, perform each step on a different line, give a new name to each result, and check that each intermediate result is what you expect. You can add any additional names or functions you want to the provided cells. Also, please be sure to not re-assign variables throughout the notebook! For example, if you use max_temperature in your answer to one question, do not reassign it later on.

To get started, load `datascience`, `numpy`, `plots`, and `ok`.

In [311]:

```
# Run this cell to set up the notebook, but please don't change it.
import numpy as np
import math
from datascience import *

# These lines set up the plotting functionality and formatting.
import matplotlib
matplotlib.use('Agg', warn=False)
%matplotlib inline
import matplotlib.pyplot as plots
plots.style.use('fivethirtyeight')
import warnings
warnings.simplefilter(action="ignore", category=FutureWarning)

# These lines load the tests.
from client.api.notebook import Notebook
ok = Notebook('project3.ok')
_ = ok.auth(inline=True)
```

```
=====================================================================
Assignment: Project 3: Movie Classification
OK, version v1.12.5
=====================================================================

Successfully logged in as epere@berkeley.edu
```

1. The Dataset

In this project, we are exploring movie screenplays. We'll be trying to predict each movie's genre from the text of its screenplay. In particular, we have compiled a list of 5,000 words that occur in conversations between movie characters. For each movie, our dataset tells us the frequency with which each of these words occurs in certain conversations in its screenplay. All words have been converted to lowercase.

Run the cell below to read the `movies` table. **It may take up to a minute to load.**

In [312]:

```
movies = Table.read_table('movies.csv')
movies.where("Title", "the matrix").select(0, 1, 2, 3, 4, 5, 10, 30, 5005)
```

Out[312]:

Title	Genre	Year	Rating	# Votes	# Words	it	not	fling
the matrix	action	1999	8.7	389480	3792	0.030327	0.00870253	0

The above cell prints a few columns of the row for the action movie *The Matrix*. The movie contains 3792 words. The word "it" appears 115 times, as it makes up $\frac{115}{3792} \approx 0.030327$ of the words in the movie. The word "not" appears 33 times, as it makes up $\frac{33}{3792} \approx 0.00870253$ of the words. The word "fling" doesn't appear at all.

This numerical representation of a body of text, one that describes only the frequencies of individual words, is called a bag-of-words representation. A lot of information is discarded in this representation: the order of the words, the context of each word, who said what, the cast of characters and actors, etc. However, a bag-of-words representation is often used for machine learning applications as a reasonable starting point, because a great deal of information is also retained and expressed in a convenient and compact format. In this project, we will investigate whether this representation is sufficient to build an accurate genre classifier.

All movie titles are unique. The `row_for_title` function provides fast access to the one row for each title.

In [313]:

```
title_index = movies.index_by('Title')
def row_for_title(title):
    """Return the row for a title, similar to the following expression (but faster)

    movies.where('Title', title).row(0)
    """
    return title_index.get(title)[0]

row_for_title('the terminator')
```

Out[313]:

```
Row(Title='the terminator', Genre='action', Year=1984, Rating=8.1, # Votes=183538, # Words=1849, i=
0.040021633, the=0.043807463, to=0.025419145, a=0.024878313, it=0.034613304, and=0.011357491, that=
0.016765818999999998, of=0.008653326, your=0.010275825, what=0.009734992, in=0.012979988999999999,
me=0.012979988999999999, is=0.007030827, do=0.005949161999999999, thi=0.010275825, dont=0.01027582
5, he=0.007030827, for=0.008653326, im=0.008653326, know=0.006489995, have=0.0032449970000000003, b
e=0.010275825, my=0.004326663, we=0.007030827, not=0.008653326, on=0.007030827, go=0.010275825, no=
0.005949161999999999, wa=0.008112493, but=0.004867496, with=0.002163332, are=0.005408329, get=0.008
112493, just=0.008112493, like=0.008653326, all=0.001622499, there=0.00757166, about=0.005949161999
999999, want=0.004326663, if=0.0032449970000000003, here=0.008112493, out=0.00757166, well=0.001622
499, think=0.00378583, they=0.004867496, up=0.005408329, him=0.005408329, can=0.005949161999999999,
one=0.00757166, were=0.004326663, how=0.001622499, got=0.004867496, she=0.0027041640000000002, at=
0.0032449970000000003, right=0.002163332, now=0.0032449970000000003, look=0.00378583, come=0.003244
9970000000003, her=0.00378583, see=0.004867496, whi=0.0027041640000000002, did=0.003244997000000000
3, oh=0.000540833, hi=0.001081666, who=0.0032449970000000003, say=0.001622499, as=0.001622499, tell
=0.0027041640000000002, time=0.002163332, good=0.0027041640000000002, ill=0.001081666, ye=0.0027041
640000000002, when=0.00378583, take=0.001081666, yeah=0.000540833, from=0.00378583, where=0.0032449
970000000003, an=0.001081666, thing=0.001622499, let=0.00378583, cant=0.004867496, back=0.00162249
9, make=0.004326663, them=0.0027041640000000002, some=0.0032449970000000003, been=0.001622499, or=
0.002163332, would=0.000540833, us=0.00378583, then=0.00378583, could=0.001622499, didnt=0.00270416
40000000002, man=0.000540833, way=0.001081666, someth=0.001622499, had=0.004867496, mean=0.00054083
3, never=0.002163332, ive=0.000540833, talk=0.002163332, too=0.001622499, okay=0.004326663, need=0.
000540833, sure=0.001622499, realli=0.000540833, down=0.001081666, love=0.0, littl=0.0, more=0.0021
63332, work=0.001081666, our=0.0, call=0.002163332, by=0.0027041640000000002, give=0.001081666, mr=
0.000540833, ani=0.001081666, onli=0.0, off=0.0, gonna=0.0027041640000000002, mayb=0.001622499, ver
i=0.001081666, over=0.0032449970000000003, tri=0.0, two=0.001622499, thank=0.001081666, peopl=0.000
540833, ha=0.002163332, much=0.002163332, year=0.001081666, day=0.0, said=0.001622499, am=0.0010816
66, sorri=0.000540833, even=0.0027041640000000002, should=0.000540833, happen=0.0, feel=0.00216333
2, find=0.000540833, becaus=0.0, sir=0.0, thought=0.000540833, anyth=0.001622499, into=0.000540833,
id=0.000540833, other=0.001081666, theyr=0.000540833, noth=0.001622499, life=0.001081666, ask=0.001
081666, use=0.000540833, befor=0.002163332, night=0.000540833, live=0.0027041640000000002, ever=0.0
01081666, better=0.001622499, still=0.001081666, first=0.001622499, youv=0.001081666, wait=0.001081
666, help=0.0032449970000000003, keep=0.001622499, than=0.000540833, last=0.000540833, put=0.002163
332, these=0.000540833, must=0.001622499, name=0.0032449970000000003, hey=0.000540833, around=0.000
540833, long=0.0, believ=0.0, those=0.0, alway=0.0, leav=0.000540833, wont=0.001622499, place=0.001
081666, pleas=0.000540833, their=0.0, after=0.001081666, doe=0.0, told=0.000540833, friend=0.0, you
ll=0.002163332, isnt=0.000540833, great=0.001081666, again=0.001081666, away=0.000540833, everyth=
0.0027041640000000002, stop=0.001081666, stay=0.000540833, doesnt=0.000540833, old=0.0, money=0.000
540833, lot=0.000540833, care=0.000540833, start=0.001081666, wouldnt=0.0, kind=0.0, big=0.00054083
3, new=0.0027041640000000002, bad=0.000540833, run=0.001081666, rememb=0.0, listen=0.001622499, hom
e=0.001081666, minut=0.000540833, enough=0.001081666, might=0.0, guess=0.000540833, made=0.0, nice=
0.0, through=0.0027041640000000002, cours=0.000540833, wrong=0.001081666, mind=0.000540833, world=
0.000540833, everi=0.0, understand=0.0027041640000000002, boy=0.0, miss=0.0, three=0.0, someon=0.00
1081666, hear=0.000540833, left=0.0, uh=0.0, fine=0.0, anoth=0.001081666, move=0.001081666, yoursel
f=0.0, job=0.000540833, own=0.000540833, real=0.0, father=0.000540833, els=0.002163332, gotta=0.001
081666, turn=0.0, car=0.0, show=0.000540833, busi=0.000540833, done=0.001081666, suppos=0.00162249
9, play=0.0, came=0.0, huh=0.000540833, head=0.0, hous=0.0, wasnt=0.000540833, knew=0.001081666, id
ea=0.0, youd=0.001081666, weve=0.0, which=0.0, best=0.0, mother=0.001622499, marri=0.0, matter=0.0,
hand=0.0, meet=0.001081666, worri=0.0, saw=0.000540833, room=0.0, same=0.0027041640000000002, prett
i=0.000540833, havent=0.001081666, chang=0.0, em=0.0, hour=0.001622499, aint=0.0, week=0.000540833,
tonight=0.0, seen=0.001081666, whole=0.001081666, seem=0.0, five=0.0, mani=0.0, onc=0.000540833, to
morrow=0.0, watch=0.0, walk=0.000540833, without=0.000540833, went=0.0, found=0.001081666, ya=0.002
163332, heard=0.000540833, woman=0.0, person=0.000540833, next=0.000540833, problem=0.0, morn=0.0,
most=0.001081666, hundr=0.0, couldnt=0.0, deal=0.0, hold=0.001081666, togeth=0.0, men=0.0, few=0.00
0540833, quit=0.0, bring=0.000540833, yet=0.001622499, hello=0.0, differ=0.002163332, while=0.00054
0833, alreadi=0.001622499, read=0.0, question=0.0, myself=0.0, word=0.0, took=0.0, second=0.0016224
99, wonder=0.0, end=0.0, forget=0.001081666, wife=0.0, sit=0.0, ago=0.0, part=0.0, probabl=0.000540
833, figur=0.0, hard=0.000540833, caus=0.0, until=0.000540833, pick=0.0, alon=0.0, stori=0.0, anywa
y=0.000540833, drink=0.000540833, point=0.000540833, power=0.000540833, late=0.001622499, brother=
0.0, open=0.0, dad=0.0, exactli=0.0, stuff=0.001081666, eye=0.0, sinc=0.0, four=0.000540833, ten=0.
000540833, hate=0.001622499, crazi=0.000540833, case=0.001622499, hurt=0.000540833, such=0.0, sound
=0.0, pay=0.001081666, close=0.0, beauti=0.0, arent=0.000540833, face=0.0, nobodi=0.00270416400000
0002, stand=0.0, chanc=0.000540833, actual=0.0, school=0.0, check=0.001622499, both=0.0, anyon=0.0,
save=0.0, interest=0.0, six=0.000540833, famili=0.0, gone=0.001622499, captain=0.0, eat=0.0, mine=
0.0, sleep=0.001622499, plan=0.0, offic=0.000540833, doctor=0.001081666, set=0.001622499, afraid=0.
000540833, somebodi=0.0, thousand=0.0, today=0.0, trust=0.001081666, wanna=0.0, hit=0.0, drive=0.0,
shut=0.000540833, month=0.0, readi=0.0, happi=0.0, send=0.001081666, under=0.000540833, lost=0.0010
81666, cut=0.0, shot=0.001081666, later=0.000540833, reason=0.000540833, rest=0.000540833, whatev=
0.000540833, theyll=0.0, side=0.000540833, TRUE=0.0, easi=0.000540833, least=0.000540833, soon=0.00
0540833, wish=0.0, break=0.0, anybodi=0.000540833, gave=0.0, troubl=0.0, dollar=0.0, sometim=0.0005
40833, book=0.000540833, town=0.0, door=0.000540833, order=0.001081666, far=0.0, pull=0.0, each=0.
0, learn=0.001081666, everybodi=0.0, phone=0.000540833, coupl=0.0, anymor=0.000540833, game=0.0, lo
se=0.0, possibl=0.001081666, heart=0.000540833, follow=0.000540833, alright=0.002163332, half=0.0,
promis=0.0, ladi=0.0, ship=0.0, dr=0.000540833, answer=0.001081666, wed=0.001622499, expect=0.0, li
ne=0.000540833, speak=0.0, import=0.000540833, buy=0.000540833, everyon=0.0, truth=0.0, met=0.0, mi
llion=0.0, high=0.0, parti=0.0, write=0.0, either=0.0, insid=0.000540833, women=0.001081666, honey=
0.0, citi=0.000540833, dream=0.0, report=0.0, funni=0.0, along=0.0, light=0.001081666, excus=0.0, g
oe=0.000540833, act=0.0, scare=0.0, war=0.0032449970000000003, wear=0.000540833, street=0.0, sort=
```

```
0.000540833, safe=0.001081666, bit=0.000540833, against=0.0, though=0.0, mom=0.000540833, bed=0.0,
cop=0.000540833, dinner=0.0, hang=0.0, drop=0.0, fall=0.0, fact=0.0, twenti=0.0, bet=0.000540833, d
og=0.001622499, polic=0.0, almost=0.000540833, aliv=0.0, water=0.0, system=0.000540833, touch=0.001
081666, fun=0.000540833, human=0.001081666, train=0.0, ahead=0.000540833, itll=0.001622499, shouldn
t=0.000540833, tire=0.0, piec=0.0, blood=0.0, between=0.0, perhap=0.0, ride=0.0, build=0.0027041640
000000002, date=0.0, husband=0.0, fli=0.0, bodi=0.000540833, full=0.0, moment=0.0, king=0.0, compan
i=0.0, countri=0.000540833, apart=0.000540833, behind=0.0, ah=0.0, control=0.001081666, pictur=0.00
0540833, handl=0.000540833, front=0.0, shall=0.0, state=0.0, finish=0.0, forc=0.0, news=0.0, sens=
0.0, free=0.0, pass=0.0, sent=0.000540833, worth=0.000540833, seriou=0.0, sister=0.0, danger=0.0005
40833, abl=0.0, plane=0.0, secret=0.0, explain=0.0, hair=0.0, danc=0.0, record=0.000540833, hot=0.0
01622499, command=0.0, surpris=0.0, werent=0.0, luck=0.0, cold=0.001081666, also=0.0, becom=0.00054
0833, fast=0.0, catch=0.000540833, bank=0.0, goodby=0.0, letter=0.0, complet=0.0, clear=0.0, straig
ht=0.0, kiss=0.0, clean=0.000540833, theyv=0.000540833, daughter=0.0, top=0.0, gener=0.000540833, h
imself=0.0, till=0.0, land=0.0, begin=0.000540833, beat=0.0, choic=0.000540833, dress=0.0, ok=0.0,
decid=0.001081666, somewher=0.0, sign=0.0, perfect=0.0, brought=0.0, outsid=0.001081666, dark=0.0,
cool=0.0, protect=0.0, hotel=0.0, lucki=0.0, taken=0.000540833, aw=0.0, fool=0.0, eight=0.0, mistak
=0.000540833, count=0.0, paper=0.0, glad=0.0, except=0.0, small=0.0, dear=0.0, bother=0.0, besid=0.
0, throw=0.0, win=0.000540833, known=0.0, natur=0.0, fifti=0.0, cover=0.0, stick=0.0, food=0.0, rat
her=0.0, coffe=0.0, secur=0.0, earth=0.0, imagin=0.0, absolut=0.000540833, sell=0.0, arm=0.0, past=
0.0, special=0.000540833, final=0.0, prison=0.0, honor=0.001081666, air=0.0, lord=0.0, parent=0.0,
lock=0.0, realiz=0.0, unless=0.0, outta=0.0, mouth=0.0, feet=0.000540833, presid=0.0, offer=0.0, le
ss=0.000540833, lead=0.0, poor=0.0, charg=0.0, hors=0.0, children=0.0, red=0.0, seven=0.000540833,
hide=0.001081666, daddi=0.0, cmon=0.0, cannot=0.0, suit=0.0, step=0.0, laugh=0.0, anim=0.0, inform=
0.0, boat=0.0, terribl=0.0, movi=0.000540833, felt=0.0, law=0.0, return=0.0, smart=0.0, total=0.0,
yesterday=0.0, usual=0.0, shell=0.0, york=0.0, hed=0.0, none=0.001081666, enjoy=0.0, cloth=0.0, cer
tainli=0.0, card=0.0, reach=0.000540833, trip=0.0, key=0.0, spend=0.0, joke=0.0, comput=0.002704164
0000000002, park=0.0, planet=0.0, bird=0.0, rule=0.0, howd=0.0, oper=0.0, servic=0.0, jump=0.0, res
pons=0.0, notic=0.0, prove=0.0, goin=0.0, machin=0.001622499, futur=0.002163332, respect=0.0, partn
er=0.0, floor=0.000540833, window=0.0, thirti=0.001081666, truck=0.0, middl=0.0, hurri=0.0, tough=
0.000540833, admir=0.0, ring=0.0, smell=0.0, road=0.0, tape=0.0, posit=0.0, armi=0.0, agent=0.0, ap
preci=0.0, swear=0.0, anywher=0.0, experi=0.0, favor=0.000540833, near=0.0, mission=0.001081666, oc
lock=0.0, fair=0.0, breath=0.000540833, buck=0.0, nine=0.0, nose=0.0, doin=0.0, wors=0.0, relax=0.0
00540833, child=0.000540833, ought=0.0, wast=0.0, welcom=0.0, file=0.0, fault=0.0, join=0.0, age=0.
0, smoke=0.0, bag=0.000540833, glass=0.0, accid=0.0, team=0.0, major=0.0, longer=0.0, fix=0.0, kind
a=0.000540833, simpl=0.0, rock=0.0, uncl=0.0, meant=0.001081666, darl=0.0, creat=0.0, grow=0.0, inv
olv=0.0, boss=0.000540833, caught=0.001081666, radio=0.0, certain=0.0, pain=0.001622499, bar=0.0, v
oic=0.0, practic=0.0, gold=0.0, english=0.0, lieuten=0.000540833, strang=0.0, kick=0.0, evil=0.0, w
hatd=0.0, entir=0.000540833, exist=0.000540833, doubt=0.000540833, memori=0.0, fish=0.0, pari=0.0,
class=0.000540833, privat=0.0, tie=0.0, bastard=0.000540833, public=0.000540833, tast=0.0, given=0.
0, bout=0.0, eh=0.0, station=0.0, guard=0.0, cost=0.0, situat=0.0, messag=0.000540833, owe=0.0, tea
ch=0.0, knock=0.0, sake=0.0, space=0.0, belong=0.0, impress=0.0, star=0.0, french=0.0, quiet=0.0, n
ervou=0.0, prepar=0.000540833, box=0.0, wake=0.0, instead=0.0, serv=0.0, tabl=0.0, account=0.0, leg
=0.0, agre=0.0, forgiv=0.0, ice=0.0, doc=0.0, slow=0.0, present=0.0, track=0.0, master=0.0, born=0.
001081666, christma=0.0, round=0.000540833, blue=0.0, consid=0.0, store=0.0, concern=0.0, quick=0.
0, rich=0.0, surviv=0.0, wall=0.0, field=0.000540833, board=0.0, lunch=0.0, kept=0.0, histori=0.0,
normal=0.0, upset=0.0, jail=0.000540833, excit=0.0, trick=0.0, hospit=0.0, theyd=0.0, steal=0.00054
0833, broke=0.0, share=0.0, sing=0.0, detect=0.0, mention=0.0, stone=0.0, arrest=0.0, list=0.0, sho
rt=0.000540833, earli=0.0, forgot=0.0, amaz=0.0, wyatt=0.0, neither=0.0, hasnt=0.0, bought=0.0, all
ow=0.0, la=0.0, fill=0.0, forev=0.0, colleg=0.0, maam=0.0, paid=0.0, mess=0.0, pack=0.0, strong=0.
0, ground=0.0, risk=0.0, cross=0.0, nothin=0.0, disappear=0.000540833, seat=0.0, across=0.0, ear=0.
0, cri=0.000540833, rais=0.0, madam=0.0, ran=0.0, suggest=0.0, weird=0.001081666, travel=0.0, warn=
0.0, whatr=0.0, visit=0.0, shop=0.0, fellow=0.0, fit=0.0, hire=0.0, peac=0.0, ruin=0.000540833, cas
h=0.000540833, wrote=0.0, feed=0.0, invit=0.0, studi=0.0, data=0.0, depend=0.0, wit=0.0, beg=0.0005
40833, de=0.0, ii=0.0, fifteen=0.0, chief=0.0, blame=0.0, hungri=0.0, direct=0.0, court=0.0, yknow=
0.000540833, marriag=0.0, push=0.0, twelv=0.0, manag=0.000540833, beer=0.0, admit=0.0, pleasur=0.0,
treat=0.000540833, island=0.0, nut=0.0, type=0.0, holi=0.0, music=0.0, bore=0.0, remind=0.0, code=
0.0, gentlemen=0.0, girlfriend=0.000540833, uhhuh=0.0, club=0.0, lawyer=0.0, assum=0.0, arrang=0.0,
govern=0.000540833, level=0.0, pop=0.0, gotten=0.0, hardli=0.0, accept=0.0, honest=0.0, won=0.00054
0833, soul=0.0, attent=0.000540833, program=0.0, shoe=0.0, bust=0.0, sheriff=0.0, univers=0.0, hunt
=0.0, feder=0.0, third=0.0, cat=0.001081666, load=0.0, contact=0.0, discuss=0.0, mask=0.0, south=0.
0, afternoon=0.0, relationship=0.0, roll=0.0, commun=0.0, singl=0.0, thou=0.0, sun=0.0, unit=0.0005
40833, wind=0.000540833, shed=0.0, settl=0.0, depart=0.000540833, bye=0.0, appear=0.0, insan=0.0, e
scap=0.0, somethin=0.0, heaven=0.0, divorc=0.0, form=0.0, definit=0.0, neck=0.0, spent=0.0, spot=0.
001081666, especi=0.000540833, imposs=0.0, hadnt=0.0, flight=0.0, pal=0.000540833, dig=0.0, hildi=
0.0, mister=0.0, recogn=0.0, apolog=0.0, address=0.0, song=0.0, um=0.0, tree=0.0, america=0.0, wire
=0.0, grab=0.0, note=0.0, action=0.0, gimm=0.0, baron=0.0, silli=0.0, deep=0.0, b=0.0, folk=0.0, pr
ess=0.001081666, sad=0.0, river=0.0, twice=0.0, price=0.0, copi=0.0, within=0.0, whether=0.0, lay=
0.000540833, engin=0.0, bridg=0.0, wow=0.0, sooner=0.0, flower=0.0, bunch=0.0, gift=0.0, trap=0.0,
plenti=0.0, cute=0.0, grand=0.0, calm=0.0, stuck=0.0, difficult=0.0, hero=0.0, suspect=0.000540833,
mama=0.0, deserv=0.0, aye=0.0, pardon=0.0, finger=0.0, wine=0.0, area=0.0, sword=0.0, broken=0.0, t
icket=0.0, somehow=0.0, signal=0.0, cell=0.0, connect=0.0, stole=0.0, comfort=0.0, proud=0.0, inter
view=0.0, search=0.0, wound=0.0, split=0.0, empti=0.0, weekend=0.0, tv=0.0, attract=0.0, birthday=
0.0, tear=0.0, pound=0.0, russian=0.0, chicken=0.0, favorit=0.0, north=0.0, creatur=0.0, evid=0.0,
arriv=0.0, condit=0.0, beach=0.000540833, o=0.0, judg=0.0, pilot=0.0, nurs=0.0, egg=0.0, convers=0.
0, rain=0.0, serious=0.0, size=0.0, insur=0.0, goodnight=0.0, duti=0.0, bone=0.0, bu=0.0, beyond=0.
0, ga=0.0, dump=0.0, shield=0.0, engag=0.0, magic=0.0, group=0.0, continu=0.0, onto=0.0, superman=
0.0, cook=0.0, emerg=0.0, asleep=0.0, pretend=0.0, san=0.0, enterpris=0.0, sea=0.0, base=0.0, deser
t=0.0, fortun=0.0, organ=0.001081666, bug=0.0, bathroom=0.0, ridicul=0.0, mountain=0.0, percent=0.
0, confus=0.0, sever=0.0, ourselv=0.000540833, career=0.0, author=0.000540833, switch=0.0, bear=0.
0, built=0.000540833, decis=0.0, pocket=0.0, immedi=0.0, damag=0.0, colonel=0.0, talkin=0.0, film=
0.0, sailor=0.0, polit=0.0, fell=0.0, advic=0.0, paint=0.0, heat=0.0, charm=0.0, warm=0.0, map=0.0,
gee=0.0, cheap=0.0, collect=0.0, wing=0.0, sweetheart=0.0, wild=0.0, expens=0.0, pool=0.0, centuri=
0.0, monster=0.0, nation=0.0, physic=0.0, detail=0.0, guest=0.0, incred=0.0, forti=0.001081666, blo
ck=0.0, fbi=0.0, vacat=0.0, bond=0.0, upstair=0.0, purpos=0.0, bright=0.0, confess=0.0, whose=0.0,
vada=0.0, wash=0.0, bottl=0.0, emot=0.0, main=0.0, correct=0.0, theori=0.0, view=0.000540833, proof
=0.0, camera=0.0, oil=0.0, access=0.0, obvious=0.0, success=0.0, weak=0.0, moon=0.0, odd=0.0, other
wis=0.0, effect=0.000540833, worst=0.0, lookin=0.0, poison=0.0, thatll=0.0, green=0.0, soldier=0.00
0540833, corner=0.0, toward=0.0, often=0.0, match=0.0, televis=0.0, low=0.000540833, heavi=0.0, ass
ign=0.0, farm=0.0, credit=0.0, boyfriend=0.0, assist=0.0, smile=0.0, sight=0.0, target=0.0, dure=0.
0, taught=0.000540833, greatest=0.0, rid=0.0, investig=0.0, camp=0.0, hat=0.0, bite=0.0, west=0.0,
nerv=0.0, kitchen=0.0, profession=0.0, climb=0.0, request=0.0, bloodi=0.0, retir=0.0, hall=0.0, bax
ter=0.0, innoc=0.0, forward=0.0, swim=0.0, due=0.0, develop=0.000540833, scene=0.0, deliv=0.0, subj
ect=0.0, freak=0.0, majesti=0.0, lex=0.0, abov=0.0, lone=0.0, spare=0.0, bullet=0.0, famou=0.0, sha
ke=0.000540833, tight=0.000540833, slip=0.000540833, held=0.0, crew=0.0, cousin=0.0, speech=0.0, my
steri=0.0, cup=0.0, trade=0.0, lift=0.0, nope=0.0, fella=0.0, newspap=0.0, commit=0.0, includ=0.0,
enter=0.0, suddenli=0.0, trial=0.0, easier=0.0, peel=0.0, support=0.0, loos=0.0, bath=0.0, locat=0.
0, familiar=0.0, excel=0.0, claim=0.0, sid=0.0, extra=0.0, tower=0.0, hill=0.0, congratul=0.0, plan
t=0.0, whom=0.000540833, draw=0.0, propos=0.0, receiv=0.0, energi=0.0, page=0.0, battl=0.0, prefer=
0.0, east=0.0, spock=0.0, embarrass=0.0, transport=0.0, victim=0.0, breakfast=0.0, foot=0.0, barn=
0.0, smith=0.0, intend=0.0, center=0.0, quarter=0.0, introduc=0.0, monsieur=0.0, nail=0.0, suffer=
0.0, patient=0.0, bandit=0.0, player=0.0, disappoint=0.000540833, gentleman=0.0, er=0.0, herself=0.
0, itself=0.0, member=0.0, whoever=0.0, trace=0.000540833, intellig=0.0, faster=0.0, rent=0.0, desk
=0.0, comin=0.0, rate=0.0, strength=0.000540833, swell=0.0, plu=0.0, thief=0.0, skin=0.001081666, m
arsh=0.0, wheel=0.0, local=0.0, closer=0.0, convinc=0.0, int=0.0, basebal=0.0, yo=0.0, pig=0.0, pri
nt=0.0, quickli=0.0, weather=0.0, blade=0.0, itd=0.0, medic=0.0, speed=0.0, lake=0.0, jone=0.0, str
ike=0.0, jerk=0.000540833, repeat=0.000540833, terrif=0.0, perform=0.0, truli=0.0, beam=0.0, celebr
=0.0, insist=0.0, wouldv=0.0, appar=0.0, object=0.000540833, concentr=0.0, pant=0.0, driver=0.00054
0833, issu=0.0, couldv=0.0, nah=0.0, lesson=0.0, research=0.0, curiou=0.0, discov=0.0, scienc=0.0,
starfleet=0.0, common=0.0, approach=0.0, afford=0.0, separ=0.0, mood=0.0, doubl=0.0, themselv=0.0,
market=0.0, lip=0.0, london=0.0, cab=0.0, fri=0.0, project=0.0, gettin=0.0, forgotten=0.0, older=0.
0, equip=0.000540833, sudden=0.0, magazin=0.0, term=0.0, guilti=0.0, clever=0.0, nightmar=0.0, lang
uag=0.0, whyd=0.0, drag=0.0, valu=0.0, releas=0.000540833, tea=0.0, cream=0.0, til=0.001081666, civ
il=0.0, threaten=0.0, wave=0.0, perfectli=0.0, ugli=0.0, affair=0.0, invent=0.0, possess=0.0, airpo
rt=0.0, plate=0.0, rush=0.0, defend=0.0, aunt=0.0, client=0.0, teeth=0.0, confirm=0.0, choos=0.0, s
aturday=0.0, spirit=0.0, upon=0.0, comrad=0.0, dude=0.0, sold=0.0, negoti=0.0, billion=0.0, tunnel=
0.0, deed=0.0, scream=0.0, chair=0.0, stock=0.0, yep=0.0, squar=0.0, seal=0.0, everywher=0.0, chara
ct=0.0, italian=0.0, style=0.0, post=0.0, custom=0.0, opinion=0.0, photograph=0.0, edg=0.0, refus=
0.0, pure=0.0, crawl=0.0, dozen=0.0, sport=0.0, degre=0.0, nowher=0.0, properti=0.0, willow=0.0, ob
viou=0.0, tour=0.0, larg=0.000540833, throat=0.0, dri=0.0, england=0.0, eleven=0.000540833, mostli=
0.0, pair=0.0, miracl=0.0, defens=0.002163332, villag=0.0, complic=0.0, intent=0.0, cent=0.0, basic
=0.000540833, contract=0.0, bare=0.0, spoke=0.0, whoa=0.0, dinosaur=0.0, ranch=0.0, licens=0.000540
833, mitch=0.0, wherev=0.000540833, california=0.0, coat=0.0, frighten=0.0, secretari=0.0, chip=0.
0, inde=0.0, justic=0.0, professor=0.0, regular=0.0, bat=0.0, rome=0.0, suppli=0.0, design=0.0, dar
e=0.0, jealou=0.0, stare=0.0, awhil=0.0, washington=0.0, shoulder=0.0, spring=0.0, fan=0.0, rang=0.
000540833, tip=0.000540833, transfer=0.000540833, rough=0.0, opportun=0.0, senat=0.0, princess=0.0,
neg=0.0, liar=0.0, statement=0.000540833, pill=0.0, desper=0.0, shouldv=0.0, betray=0.0, leader=0.
0, lawson=0.0, shower=0.000540833, someplac=0.0, becam=0.0, snake=0.0, british=0.0, friendli=0.0, n
ecessari=0.0, ignor=0.0, shape=0.0, trail=0.0, resist=0.0, biggest=0.0, social=0.0, sourc=0.0, weig
ht=0.0, militari=0.000540833, uyouu=0.0, offici=0.001081666, diamond=0.0, mix=0.0, citizen=0.0, oug
hta=0.0, further=0.0, activ=0.0, whatta=0.0, sixti=0.0, fake=0.0, seek=0.0, romant=0.0, among=0.0,
freez=0.000540833, mustv=0.0, gang=0.0, neighbor=0.0, vampir=0.0, zone=0.0, offens=0.0, threw=0.0,
band=0.0, particular=0.0, unfortun=0.0, bottom=0.0, nuclear=0.0, bless=0.0, freedom=0.0, intern=0.
0, yard=0.0, cowboy=0.0, borrow=0.0, race=0.0, event=0.0, recal=0.0, worm=0.0, lover=0.0, understoo
d=0.0, hmm=0.0, emperor=0.0, cure=0.0, challeng=0.0, confid=0.0, europ=0.0, launch=0.0, lampert=0.
0, knee=0.0, bay=0.0, simpli=0.0, talent=0.0, chick=0.0, snow=0.0, actor=0.0, brave=0.0, schedul=0.
000540833, fighter=0.000540833, avoid=0.0, mayor=0.0, realiti=0.0, mirror=0.0, milk=0.0, whenev=0.
0, neighborhood=0.0, origin=0.0, wood=0.0, stolen=0.0, crowd=0.0, self=0.0, product=0.0, robinson=
0.0, princ=0.0, express=0.0, garden=0.0, imag=0.0, femal=0.0, storm=0.0, rat=0.0, eighteen=0.0, sho
ck=0.0, gut=0.0, teacher=0.0, jacket=0.0, squad=0.0, lenni=0.0, contain=0.0, pray=0.0, friday=0.000
540833, relat=0.0, stranger=0.0, knowledg=0.0, museum=0.0, recommend=0.0, tank=0.0, attitud=0.0, br
own=0.0, fed=0.0, restaur=0.0, clock=0.0, cia=0.0, process=0.0, ghost=0.0, leagu=0.0, nor=0.0, atte
mpt=0.0, cheer=0.0, piti=0.000540833, dunno=0.0, fresh=0.0, da=0.0, accord=0.0, el=0.000540833, lim
it=0.0, shame=0.0, telephon=0.0, pump=0.0, unusu=0.0, soft=0.0, wise=0.0, advanc=0.0, prom=0.0, ass
ur=0.0, attorney=0.0, aboard=0.0, horribl=0.0, wolf=0.0, extrem=0.0, luthor=0.0, empir=0.0, holiday
=0.0, thinkin=0.0, stage=0.0, grate=0.0, cow=0.0, spi=0.0, mail=0.0, pregnant=0.0, awar=0.0, cooper
=0.0, deck=0.0, kidnap=0.0, tall=0.000540833, priest=0.0, blew=0.0, dyou=0.0, exit=0.0, wet=0.0, am
ount=0.0, shift=0.0, alvi=0, score=0.0, mar=0.0, border=0.0, counti=0.0, sunday=0.0, bleed=0.0, gra
il=0.0, rip=0.000540833, requir=0.0, writer=0.0, elev=0.0, wipe=0.000540833, sergeant=0.0, stomach=
0.0, destini=0.0, captur=0.000540833, passion=0.0, rescu=0.0, juic=0.0, kit=0.0, johnson=0.0, polic
i=0.0, coke=0.0, ocean=0.0, fuel=0.0, cage=0.0, clearli=0.0, lousi=0.0, identifi=0.0, msieu=0.0, ma
teri=0.0, statu=0.0, fate=0.000540833, nearli=0.0, graduat=0.0, castl=0.0, chicago=0.0, batman=0.0,
piano=0.0, permiss=0.0, sixteen=0.0, screen=0.0, tail=0.0, jet=0.0, punch=0.0, termin=0.001622499,
replac=0.0, rope=0.0, vulcan=0.0, hmmm=0.0, thursday=0.0, sin=0.0, slept=0.0, drown=0.0, alarm=0.0,
lab=0.000540833, satisfi=0.0, fort=0.0, buzz=0.0, shirt=0.0, wallet=0.0, earlier=0.0, gear=0.0, mid
night=0.0, remov=0.0, below=0.0, sky=0.0, boot=0.0, china=0.0, drove=0.0, downstair=0.0, russia=0.
0, photo=0.0, radiat=0.0, sneak=0.0, pin=0.0, mmm=0.000540833, season=0.0, vega=0.0, gallagh=0.0, r
eput=0.0, guarante=0.0, monday=0.0, regret=0.0, indian=0.0, batteri=0.0, repres=0.0, cloud=0.0, mex
ico=0.0, someday=0.0, bud=0.0, describ=0.0, legal=0.000540833, uniform=0.0, cake=0.0, brilliant=0.
0, sand=0.0, lombard=0.0, threat=0.000540833, nois=0.0, confer=0.0, texa=0.0, occur=0.0, argu=0.0,
grew=0.000540833, wynant=0.0, meat=0.0, harm=0.0, foolish=0.0, depress=0.0, twentyf=0.0, reward=0.
0, garbag=0.0, honestli=0.0, observ=0.0, geniu=0.0, recent=0.0, guid=0.0, insult=0.0, marylin=0.0,
appoint=0.0, provid=0.0, crow=0.0, brand=0.0, tool=0.0, model=0.0, lad=0.0, logic=0.0, thatd=0.0, l
emm=0.0, royal=0.0, pattern=0.000540833, rare=0.0, fund=0.0, disk=0.0, pressur=0.0, howev=0.0, prom
ot=0.0, lifetim=0.0, button=0.0, written=0.0, snap=0.0, fashion=0.0, eventu=0.0, shoulda=0.0, scien
tist=0.0, shadow=0.0, exact=0.0, scar=0.0, casablanca=0.0, gain=0.0, meal=0.0, sugar=0.0, explor=0.
0, flat=0.0, mommi=0.0, compliment=0.0, passport=0.0, vote=0.0, male=0.0, devic=0.0, wander=0.0, be
droom=0.0, estat=0.0, trunk=0.0, option=0.0, function=0.0, parker=0.0, loud=0.0, fenc=0.0, channel=
0.0, precog=0.0, malkovich=0.0, solid=0.0, ninotchka=0.0, younger=0.0, bell=0.0, scratch=0.0, jam=
0.0, popular=0.0, iii=0.0, obsess=0.0, territori=0.0, zero=0.001622499, although=0.0, swing=0.0, fo
ught=0.0, preys=0.0, yuh=0.0, surrend=0.0, ate=0.0, roof=0.0, artist=0.0, loan=0.0, blast=0.0, focu
=0.0, march=0.0, reveng=0.0, mac=0.0, champagn=0.0, railroad=0.0, lo=0.0, viru=0.0, bum=0.00054083
3, zoo=0.0, scan=0.0, palac=0.0, courag=0.0, cave=0.0, hudson=0.0, underground=0.0, navi=0.0, iraqi
=0.0, expert=0.0, ta=0.0, higher=0.0, instruct=0.0, saavik=0.0, birth=0.0, makin=0.0, circumst=0.0,
maya=0.0, ancient=0.0, skip=0.0, downtown=0.0, agenc=0.0, shelter=0.0, associ=0.0, per=0.0, rabbit=
0.0, habit=0.0, convict=0.0, abandon=0.0, torpedo=0.0, fifth=0.0, curs=0.0, creasi=0.0, hawk=0.0, c
orpor=0.000540833, tellin=0.0, safeti=0.0, distanc=0.0, coast=0.0, genesi=0.0, spell=0.0, thrill=0.
0, destruct=0.0, frozen=0.0, miser=0.0, result=0.0, moral=0.0, badg=0.0, pour=0.0, network=0.000540
833, vice=0.0, pen=0.0, rise=0.0, metal=0.000540833, opera=0.0, nobl=0.0, invest=0.0, woke=0.0, low
er=0.0, chain=0.0, crook=0.0, video=0.0, fascin=0.0, fulli=0.0, awak=0.0, cheat=0.0, indi=0.0, earn
=0.0, trigger=0.0, kip=0.0, asham=0.0, monitor=0.0, patienc=0.0, fantast=0.0, student=0.0, sweat=0.
000540833, thumb=0.0, sat=0.0, sale=0.0, firm=0.0, led=0.0, societi=0.0, thee=0.0, fleet=0.0, stake
=0.0, seventeen=0.0, rig=0.0, treasur=0.0, doll=0.0, walli=0.0, mental=0.0, monkey=0.0, progress=0.
0, yell=0.0, examin=0.0, senior=0.0, panic=0.0, shhh=0.0, alert=0.0, smash=0.000540833, directli=0.
0, stair=0.0, coordin=0.0, complain=0.0, precis=0.0, orang=0.0, toy=0.0, bail=0.0, entertain=0.0, w
agon=0.0, director=0.0, abil=0.0, ape=0.0, silver=0.0, kong=0.0, ad=0.0, santa=0.0, mere=0.0, clue=
0.0, document=0.0, central=0.0, chop=0.0, articl=0.0, fanci=0.0, tag=0.0, tradit=0.0, pathet=0.0, c
apabl=0.0, stink=0.0, sector=0.0, sore=0.0, appl=0.0, creep=0.0, headach=0.0, s=0.0, packag=0.0, sa
mpl=0.0, speci=0.0, zira=0.0, grave=0.0, float=0.0, bull=0.0, psycho=0.0, glori=0.0, slowli=0.00054
0833, berlin=0.0, spoken=0.0, friedman=0.0, psychic=0.0, foreign=0.0, 30=0.0, bodyguard=0.0, frankl
i=0.0, inspir=0.0, edi=0.0, yellow=0.0, plastic=0.0, winter=0.0, grade=0.0, link=0.0, warrior=0.0,
punish=0.0, poem=0.0, neo=0.0, tini=0.0, tax=0.0, ms=0.000540833, gate=0.0, combin=0.0, bound=0.0,
circl=0.0, rout=0.0, spoil=0.0, interrupt=0.0, sandwich=0.0, exercis=0.0, refer=0.0, huge=0.0, exam
pl=0.0, standard=0.0, easili=0.0, gum=0.0, kringelein=0.0, add=0.0, nonsens=0.0, port=0.0, eighti=
0.0, demand=0.0, librari=0.0, compar=0.000540833, profit=0.0, hidden=0.0, avail=0.0, blown=0.0, dus
t=0.0, iron=0.0, initi=0.0, rub=0.0, maid=0.0, effort=0.0, stood=0.0, construct=0.0, sensit=0.0, re
```

```
l=0.0, pie=0.0, ident=0.0, bowl=0.0, attend=0.0, commiss=0.0, st=0.0, moscow=0.0, procedur=0.0, blo
nd=0.0, tryin=0.0, flatter=0.0, satellit=0.0, t=0.0, bride=0.0, stew=0.0, loss=0.0, warp=0.0, reser
v=0.0, therel=0.0, fourteen=0.0, electr=0.0, proper=0.0, healthi=0.0, decent=0.0, wreck=0.0, thin=
0.0, behavior=0.0, naw=0.0, jersey=0.0, affect=0.0, advis=0.0, cancel=0.0, particularli=0.0, laszlo
=0.0, spread=0.0, garag=0.0, valuabl=0.0, health=0.0, stretch=0.000540833, helicopt=0.0, workin=0.
0, technic=0.0, kubelik=0.0, gosh=0.0, belt=0.0, sock=0.0, backup=0.0, starv=0.000540833, kat=0.0,
tuesday=0.0, cultur=0.0, exchang=0.0, yup=0.0, nest=0.0, crush=0.0, accus=0.0, morpheu=0.0, dealer=
0.0, rank=0.0, industri=0.0, ador=0.0, manner=0.0, sheet=0.0, chocol=0.0, instinct=0.0, sacrific=0.
0, altern=0.0, pet=0.0, cabl=0.0, impuls=0.0, magnific=0.0, adventur=0.0, inch=0.0, unhappi=0.0, kn
ox=0.0, civilian=0.0, vision=0.0, pirat=0.0, disast=0.0, surfac=0.0, frame=0.0, armor=0.0, rafe=0.
0, dyle=0.0, mate=0.0, sweeti=0.0, flip=0.0, bid=0.0, expos=0.0, ahh=0.0, cast=0.0, routin=0.0, ali
en=0.0, determin=0.0, fox=0.0, prevent=0.000540833, hack=0.0, educ=0.0, dartagnan=0.0, stronger=0.
0, ordinari=0.0, humor=0.0, badli=0.0, footbal=0.0, behav=0.0, silenc=0.0, preciou=0.0, rude=0.0, f
lame=0.0, respond=0.0, lunat=0.0, scout=0.0, nineteen=0.0, surgeon=0.0, sink=0.0, produc=0.0, barga
in=0.000540833, valley=0.000540833, whod=0.0, cap=0.0, golden=0.0, recov=0.0, competit=0.0, vessel=
0.0, westley=0.0, marqui=0.0, financi=0.0, fortress=0.0, clown=0.0, proceed=0.0, duck=0.0, presenc=
0.0, motiv=0.0, asid=0.0, handsom=0.0, klingon=0.0, scari=0.0, champion=0.0, flash=0.0, stu=0.0, am
bassador=0.0, foley=0.0, skill=0.0, chees=0.0, duff=0.0, pink=0.0, predict=0.0, bein=0.0, movement=
0.0, regist=0.0, giant=0.0, explan=0.0, flow=0.0, grown=0.000540833, heal=0.0, humili=0.0, thrown=
0.0, genet=0.0, minist=0.0, fee=0.0, entranc=0.0, wednesday=0.0, dna=0.0, re=0.0, singer=0.0, emplo
y=0.0, specif=0.0, solut=0.0, element=0.0, plain=0.0, publish=0.0, buckaroo=0.0, hollywood=0.0, m=
0.0, staff=0.0, sensor=0.0, dull=0.0, advantag=0.0, sacr=0.0, mass=0.0, mug=0.0, interfer=0.0, rumo
r=0.0, thorwald=0.0, sentiment=0.0, chemic=0.0, bureau=0.0, oblig=0.0, signatur=0.0, incid=0.0, mer
ci=0.000540833, sheldrak=0.0, simul=0.0, classic=0.0, struck=0.0, titl=0.0, winner=0.0, twist=0.0,
sail=0.0, continued:=0.0, itali=0.0, tran=0.0, mccoy=0.0, noon=0.000540833, profil=0.0, motel=0.0,
alter=0.0, appli=0.0, rico=0.0, hong=0.0, grandma=0.0, diari=0.0, saint=0.0, gambl=0.0, dock=0.0, m
atrix=0.0, psycholog=0.0, aim=0.0, anytim=0.0, rifl=0.0, editor=0.0, coincid=0.0, strip=0.0, spice=
0.0, eras=0.0, cabin=0.0, detroit=0.0, individu=0.0, turkey=0.0, prize=0.0, tune=0.0, drill=0.0, av
erag=0.0, inna=0, visa=0.0, dirt=0.0, powder=0.0, quot=0.0, calcul=0.0, toss=0.0, string=0.0, wore=
0.0, stab=0.0, coulda=0.0, tan=0.0, meter=0.0, j=0.0, path=0.0, complex=0.000540833, shuttl=0.0, su
rround=0.000540833, riot=0.0, blank=0.0, venic=0.0, traffic=0.0, mora=0.0, thisll=0.0, employe=0.0,
cougar=0.0, highli=0.0, happili=0.0, tone=0.0, borg=0.0, wide=0.0, spray=0.0, juri=0.0, stamp=0.0,
avenu=0.0, journey=0.0, takin=0.0, former=0.0, jewelri=0.0, safer=0.0, log=0.0, ti=0.0, plagu=0.0,
regard=0.0, mutant=0.0, lobbi=0.0, vigo=0.0, spider=0.0, maverick=0.0, salt=0.0, defeat=0.0, dish=
0.0, dime=0.0, rotten=0.0, donut=0.0, permit=0.0, volunt=0.001081666, chest=0.0, theatr=0.0, announ
c=0.0, audienc=0.0, peanut=0.0, structur=0.0, oneil=0.0, mayo=0.0, venkman=0.0, twentyfour=0.0, atm
ospher=0.0, chairman=0.0, miami=0.0, anyhow=0.0, breaker=0.0, te=0.0, whatll=0.0, mighti=0.0, shotg
un=0.0, coward=0.0, plot=0.0, opposit=0.0, aid=0.0, steadi=0.0, inn=0.0, picard=0.0, occas=0.0, ele
ct=0.0, lotta=0.0, afterward=0.0, disgust=0.0, duke=0.0, capit=0.0, superior=0.0, carrier=0.0, sedu
c=0.0, buffi=0.0, seventi=0.0, section=0.0, cigar=0.0, vehicl=0.0, romulan=0.0, indic=0.0, goal=0.
0, goat=0.0, breed=0.0, anger=0.0, flesh=0.000540833, etern=0.0, warrant=0.0, selfish=0.0, norvil=
0.0, worf=0.0, fabul=0.0, sulu=0.0, del=0.0, sharp=0.0, pleasant=0.0, perman=0.0, howr=0.0, puppet=
0.0, heel=0.0, factori=0.000540833, shrink=0.0, ol=0.0, symbol=0.0, combat=0.000540833, jungl=0.0,
psychiatrist=0.0, legend=0.001081666, mmmm=0.0, mayflow=0.0, jedi=0.0, 100=0.0, nineti=0.0, tale=0.
0, ultim=0.0, schwartz=0.0, victori=0.0, 20=0.0, bourn=0.0, dancer=0.0, pge=0, occasion=0.0, aggres
s=0.0, tourist=0.0, episod=0.0, conscienc=0.0, potenti=0.0, fallen=0.0, solv=0.0, missil=0.0, camel
=0.0, darlin=0.0, current=0.0, coach=0.0, blah=0.0, reaction=0.0, bend=0.0, eastern=0.0, mortal=0.
0, silent=0.0, hostil=0.0, forest=0.0, sentenc=0.0, eaten=0.0, grandmoth=0.0, paranoid=0.0, purs=0.
0, robberi=0.0, mig=0.0, gather=0.0, rag=0.0, lednov=0.0, fingerprint=0.0, distract=0.0, accomplish
=0.0, locker=0.0, patch=0.0, union=0.0, honeymoon=0.0, alley=0.0, delay=0.0, commerci=0.0, costum=
0.0, bunni=0.0, starship=0.0, grey=0.000540833, dame=0.0, orbit=0.0, worthi=0.0, tribun=0.0, neutra
l=0.0, loyalti=0.0, principl=0.0, goddam=0.0, forbidden=0.0, chekov=0.0, needl=0.0, compound=0.0, b
luff=0.0, equal=0.0, feelin=0.0, fourth=0.0, embassi=0.0, mon=0.0, tiger=0.0, musta=0.0, began=0.0,
ganz=0.0, miller=0.000540833, marvel=0.0, seri=0.001081666, toto=0.0, consider=0.0, adjust=0.0, thi
rteen=0.0, haul=0.0, veget=0.0, angl=0.0, lack=0.0, useless=0.0, pitch=0.0, poetri=0.0, shi=0.0, he
lluva=0.0, chew=0.0, chewi=0.0, valiant=0.0, expedit=0.0, goos=0.0, suspici=0.0, temperatur=0.0, sc
ale=0.0, knight=0.0, le=0.0, poker=0.0, instant=0.0, instanc=0.0, conduct=0.0, ranger=0.0, passag=
0.0, hip=0.0, tap=0.0, repair=0.0, wrap=0.0, vader=0.0, chat=0.0, prayer=0.0, cattl=0.0, tend=0.0,
vinci=0.0, mistaken=0.0, root=0.0, council=0.0, twin=0.0, extraordinari=0.0, razor=0.0, raoul=0.0,
dawson=0.0, liberti=0.0, madmartigan=0.0, precrim=0.0, oath=0.0, coloni=0.0, jeep=0.0, version=0.00
0540833, policeman=0.0, distress=0.0, flew=0.0, farmer=0.0, scientif=0.0, measur=0.0, sonora=0.0, s
eventyf=0.0, tube=0.000540833, achiev=0.0, ditch=0.0, ext=0.0, gotham=0.0, steel=0.0, soap=0.0, las
er=0.0, amus=0.0, theo=0.0, paradis=0.0, epp=0.0, magua=0.0, grandfath=0.0, whack=0.0, enzo=0.0, su
per=0.0, shave=0.0, exhaust=0.0, chamber=0.0, grunemann=0, bachelor=0.0, li=0.0, pride=0.0, mm=0.0,
argument=0.0, core=0.0, ouch=0.0, dispos=0.000540833, ripley=0.0, phaser=0.0, driven=0.0, violent=
0.0, violat=0.0, lean=0.0, deputi=0.0, martini=0.0, ex=0.0, worker=0.0, czech=0.0, wanta=0.0, concl
us=0.0, unlik=0.0, acknowledg=0.0, harbor=0.0, kansa=0.0, site=0.0, swana=0.0, swamp=0.0, appeal=0.
0, bean=0.0, alpha=0.0, properli=0.0, widow=0.0, gentl=0.0, convent=0.0, delight=0.0, bailey=0.0, i
mprov=0.0, 24=0.0, corridor=0.0, shakespear=0.0, payment=0.0, fantasi=0.0, eas=0.0, 67=0.0, unbelie
v=0.0, ambros=0.0, woulda=0.0, architect=0.0, outfit=0.0, patrol=0.000540833, pipe=0.0, collar=0.0,
benefit=0.0, hop=0.0, d=0.0, subway=0.0, spanish=0.0, prime=0.0, surgeri=0.000540833, ace=0.0, terr
ifi=0.0, worthless=0.0, superhero=0.0, deliveri=0.0, chopper=0.0, ethic=0.0, shown=0.0, yourselv=0.
0, governor=0.0, user=0.0, envelop=0.0, toast=0.0, novel=0.0, visitor=0.0, sorta=0.0, uniqu=0.0, br
ead=0.0, uncomfort=0.0, phase=0.000540833, con=0.0, clinic=0.0, phoni=0.0, mu=0.0, poet=0.0, rubber
=0.000540833, trash=0.0, bark=0.0, rocket=0.0, experienc=0.0, injur=0.0, cuervo=0.0, soup=0.0, ring
o=0.0, creed=0.0, cobb=0.0, straighten=0.0, golf=0.0, wherer=0.0, consequ=0.0, recruit=0.0, cannon=
0.0, bitter=0.0, mechan=0.0, inspector=0.0, 7=0.0, fare=0.0, tent=0.0, domino=0.0, punk=0.0, oswald
=0.0, belli=0.0, unicorn=0.0, romanc=0.0, couch=0.0, pope=0.0, schuyler=0, whisper=0.0, zack=0.0, f
lyer=0.0, waitin=0.0, divis=0.000540833, fairli=0.0, transmiss=0.0, visual=0.0, evolv=0.0, candid=
0.0, chosen=0.0, sincer=0.0, lap=0.0, steam=0.0, leeloo=0.0, uhuh=0.0, sustain=0.0, thirtyf=0.0, sa
lari=0.0, invis=0.000540833, cedar=0.0, urgent=0.0, stiff=0.0, lung=0.0, mustnt=0.0, maintain=0.0,
10=0.0, smokey=0.0, surveil=0.0, hound=0.0, remark=0.0, austrian=0.0, waitress=0.000540833, worship
=0.0, begun=0.0, hint=0.0, strictli=0.0, crane=0.0, nicki=0.0, junk=0.0, crown=0.0, puppi=0.0, wild
er=0.0, stunt=0.0, therefor=0.0, tick=0.0, curios=0.0, laval=0.0, loyal=0.0, concept=0.0, umm=0.0,
burst=0.0, testifi=0.0, magician=0.0, setup=0.0, modern=0.0, miseri=0.0, chart=0.0, bai=0.0, ceremo
ni=0.0, ton=0.0, parol=0.0, implant=0.0, isol=0.0, brief=0.0, manchest=0.0, sweep=0.0, squeez=0.0,
reunion=0.0, hm=0.0, conklin=0.0, district=0.0, troop=0.0, japan=0.0, revers=0.0, chao=0.0, acm=0.
0, wive=0.0, technolog=0.0, owner=0.0, dumper=0, offend=0.0, pilgrim=0.0, plug=0.0, salad=0.0, mont
ana=0.0, whistl=0.0, swore=0.0, everyday=0.0, sophist=0.0, establish=0.0, pile=0.0, cosgrov=0.0, se
wer=0.0, asset=0.0, stream=0.0, knot=0.0, crank=0.0, committe=0.0, suspicion=0.0, runnin=0.0, lyssa
=0, undercov=0.0, anderton=0.0, infect=0.0, casino=0.0, counter=0.0, bump=0.0, toler=0.0, irish=0.
0, deadli=0.0, dizzi=0.0, slap=0.0, closest=0.0, suitcas=0.0, merri=0.0, broad=0.0, deeper=0.0, wit
wer=0.0, cloak=0.0, shine=0.0, personnel=0.0, explod=0.0, stabil=0.0, popul=0.0, gal=0.0, butcher=
0.0, neednt=0.0, perimet=0.0, phillipp=0.0, stud=0.0, theater=0.0, 40=0.0, discoveri=0.0, 50=0.0, u
meu=0.0, hawaii=0.0, liquid=0.0, underneath=0.0, balanc=0.000540833, grass=0.0, faint=0.0, anxiou=
0.0, resourc=0.0, spark=0.0, highway=0.0, mutat=0.0, thiev=0.0, smooth=0.0, steer=0.0, fortyeight=
0.0, vault=0.0, venza=0.0, basement=0.0, wee=0.0, skull=0.0, nay=0.0, wick=0.0, attach=0.0, exhibit
=0.0, tactic=0.000540833, restor=0.0, administr=0.0, sworn=0.0, penguin=0.0, appropri=0.0, jonah=0.
0, fals=0.0, narrow=0.0, transit=0.0, duffi=0.0, acid=0.0, translat=0.0, philosophi=0.0, defin=0.0,
latest=0.0, host=0.0, puzzl=0.0, toon=0.0, brenner=0, backward=0.0, yall=0.0, hacker=0.0, motion=0.
0, coma=0.0, kingdom=0.0, academi=0.0, closet=0.0, leak=0.0, automat=0.0, foul=0.0, extend=0.0, reg
ul=0.0, remot=0.0, incom=0.0, audit=0.0, hammond=0.0, drawn=0.0, hana=0.0, presum=0.0, airlin=0.0,
banana=0.0, melt=0.0, institut=0.0, poke=0.0, handi=0.0, commission=0.0, dismiss=0.0, strap=0.0, gr
ip=0.0, awkward=0.0, merger=0.0, noir=0.0, raw=0.0, cuz=0.0, predat=0.0, coin=0.0, luggag=0.0, hail
=0.0, tomb=0.0, reckless=0.0, heck=0.0, leia=0.0, ammo=0.0, brass=0.0, divin=0.0, shore=0.0, saloon
=0.0, homework=0.0, spook=0.0, wrist=0.0, prioriti=0.0, palm=0.0, stress=0.0, bush=0.0, muscl=0.0,
gabe=0, co=0.0, india=0.0, independ=0.0, eleph=0.0, flaw=0.0, orlean=0.0, leather=0.0, whatcha=0.0,
putter=0, tempt=0.0, ash=0.0, hammer=0.0, click=0.0, influenc=0.0, qualiti=0.0, c=0.0, ankl=0.0, en
courag=0.0, dedic=0.0, neat=0.0, castor=0, temper=0.0, mill=0.0, entitl=0.0, treadston=0.0, pan=0.
0, increas=0.0, basketbal=0.0, mademoisel=0.0, farewel=0.0, pizza=0.0, fax=0.0, waiter=0.0, quaid=
0.0, circuit=0.0, reilli=0.0, fossil=0.0, arrog=0.0, bewar=0.0, hay=0.0, boost=0.0, pursu=0.0, darn
=0.0, colleagu=0.0, lipstick=0.0, belief=0.0, coupla=0.0, poster=0.0, gregor=0.0, comedi=0.0, parti
cl=0.0, award=0.0, eyebal=0.0, florida=0.0, riddl=0.0, baltimor=0.0, pace=0.0, lectur=0.0, bastaldi
=0.0, rendezv=0.0, finch=0.0, marin=0.0, scotch=0.0, counselor=0.0, roommat=0.0, analysi=0.0, propo
sit=0.0, exclus=0.0, outa=0.0, dictat=0.0, cafe=0.0, stuf=0.0, integr=0.0, racket=0.0, bait=0.0, se
ein=0.0, n=0.0, transform=0.0, prey=0.0, critic=0.0, reliev=0.0, vulner=0.0, compet=0.0, shade=0.0,
duchess=0.0, alcohol=0.0, dine=0.0, larger=0.0, egypt=0.0, resent=0.0, unknown=0.0, smyth=0.0, maro
on=0.0, boston=0.0, potter=0.0, scumbag=0.0, hick=0.0, sire=0.0, partial=0.0, steed=0.0, passeng=0.
0, drank=0.0, 5=0.0, dc=0.0, rebel=0.0, fruit=0.0, resign=0.0, archer=0.0, bike=0.0, privileg=0.0,
content=0.0, ford=0.0, deceiv=0.0, bontec=0, pad=0.0, atlant=0.0, agatha=0.0, deb=0.0, chimera=0.0,
uhh=0.0, caesar=0.0, zavitz=0.0, finest=0.0, facil=0.0, beaten=0.0, collaps=0.0, fade=0.0, western=
0.0, adopt=0.0, raziel=0.0, bolt=0.0, hug=0.0, picnic=0.0, comb=0.0, region=0.0, reflect=0.0, brick
=0.0, weigh=0.0, dealt=0.0, penit=0, shovel=0.0, hiya=0.0, entri=0.0, debt=0.0, devot=0.0, intens=
0.0, mimi=0.0, narcissu=0.0, uiu=0.0, prior=0.0, qualifi=0.0, barrel=0.0, vow=0.0, enforc=0.0, rans
om=0.0, parri=0.0, thick=0.0, bounc=0.0, deton=0.0, thirst=0.0, drift=0.0, corps=0.0, liter=0.0, co
smo=0.0, nyah=0.0, sarg=0.0, journal=0.0, scanner=0.000540833, highest=0.0, bonu=0.0, quinc=0, make
up=0.0, bow=0.0, lame=0.0, label=0.0, parrish=0, strain=0.0, sequenc=0.0, dragon=0.0, radar=0.0, sp
arazza=0, constant=0.0, galaxi=0.0, equat=0.0, curtain=0.0, hudsuck=0.0, atho=0.0, 45=0.0, toe=0.0,
hatch=0.0, 6=0.0, surg=0.0, analyz=0.0, mount=0.0, sub=0.0, uthatu=0.0, cheek=0.0, smarter=0.0, cho
se=0.0, snoopi=0.0, mutual=0.0, hon=0.0, whip=0.0, chill=0.0, fog=0.0, laundri=0.0, prospect=0.0, e
xtinct=0.0, askin=0.0, stabl=0.000540833, ai=0.0, genuin=0.0, ilsa=0.0, gig=0.0, lamb=0.0, relief=
0.0, vanish=0.0, quest=0.0, agreement=0.0, brynner=0.0, walker=0.0, tribe=0.0, templ=0.0, cabinet=
0.0, crisi=0.0, blanket=0.0, greedi=0.0, addit=0.0, forth=0.0, puttin=0.0, you:=0.0, gorgeou=0.0, m
edicin=0.0, x=0.0, mustang=0.0, utah=0.0, cruel=0.0, net=0.0, underwear=0.0, declar=0.0, scrambl=0.
0, enorm=0.0, task=0.0, despis=0.0, manual=0.0, magnet=0.0, guardian=0.0, saddl=0.0, certifi=0.0, m
ob=0.0, lando=0.0, 12=0.0, ike=0.0, heresi=0.0, slightli=0.0, absurd=0.0, journalist=0.0, freeway=
0.0, bingo=0.0, circu=0.0, swayzak=0.0, helmet=0.0, slice=0.0, illus=0.0, tex=0.0, intimid=0.0, scr
ipt=0.0, hid=0.0, uisu=0.0, succeed=0.0, silk=0.0, trex=0.0, extens=0.0, conveni=0.0, si=0.0, juris
dict=0.0, mightv=0.0, impli=0.0, testimoni=0.0, pawn=0.0, pauli=0.0, flynn=0.0, ink=0.0, demonstr=
0.0, maniac=0.0, supper=0.0, guilt=0.0, luxuri=0.0, uhura=0.0, sabotag=0.0, pose=0.0, ole=0.0, gast
on=0.0, taxi=0.0, yu=0.0, manhattan=0.0, e=0.0, egon=0.0, probe=0.0, studio=0.0, height=0.0, freque
nc=0.0, 15=0.0, orient=0.0, foundat=0.0, sack=0.0, invad=0.0, hopeless=0.0, minnesota=0.0, mud=0.0,
shortli=0.0, frustrat=0.0, rot=0.0, nomak=0.0, valentin=0.0, necessarili=0.0, twentyeight=0.0, fels
on=0.0, rehears=0.0, flag=0.0, grusinskaya=0.0, dieter=0.0, clip=0.0, anticip=0.0, bounti=0.0, cany
on=0.0, xxxxxx=0.0, rattl=0.0, nephew=0.0, fame=0.0, collector=0.0, groom=0.0, sayin=0.0, sew=0.0,
twentyseven=0.0, telegram=0.0, await=0.0, spill=0.0, medal=0.0, upsid=0.0, oxygen=0.0, brake=0.0, r
aid=0.0, dispatch=0.0, specimen=0.0, rear=0.0, messi=0.0, tattoo=0.0, dip=0.0, scotland=0.0, cowgir
l=0.0, airplan=0.0, valet=0.0, halloween=0.0, massey=0.0, privaci=0.0, beard=0.0, eject=0.0, envi=
0.0, acquir=0.0, fork=0.0, worn=0.0, youyou=0.0, zaiu=0.0, ministri=0.0, previs=0.0, custodi=0.0, d
isguis=0.0, inject=0.0, ribbon=0.0, dough=0.0, youth=0.0, purpl=0.0, labor=0.0, steak=0.0, fortyf=
0.0, counsel=0.0, prototyp=0.0, vodka=0.0, riddick=0, ow=0.000540833, fond=0.0, reliant=0.0, global
=0.0, scroll=0.0, approxim=0.0, throne=0.0, this:=0.0, toxic=0.0, plato=0.0, lott=0.0, compromis=0.
0, cooki=0.0, scope=0.0, hairdress=0.0, paso=0.0, hast=0.0, crop=0.0, budget=0.0, maker=0.0, linden
mey=0.0, pencil=0.0, rode=0.0, maiden=0.0, shout=0.0, unconsci=0.0, approv=0.0, mace=0.0, maneuv=0.
0, 7.1=0.0, similar=0.0, booth=0.0, acquaint=0.0, wealth=0.0, doucet=0.0, resid=0.0, commodu=0.0, w
ithdraw=0.0, skinni=0.0, ohhh=0.0, circul=0.0, strand=0.0, bain=0, witch=0.0, surf=0.0, dramat=0.0,
joker=0.0, 500=0.0, hustl=0.0, halfway=0.0, bracelet=0.0, pooch=0.0, unotu=0.0, reverend=0.0, capac
=0.0, oooh=0.0, swap=0.0, tore=0.0, accident=0.0, shitti=0.0, formal=0.0, preserv=0.0, psychologist
=0.0, shinzon=0, reveal=0.0, unpredict=0.0, nicknam=0.0, draft=0.0, nicer=0.0, instrument=0.0, pod=
0.0, roast=0.0, struggl=0.0, actress=0.0, praetor=0, rexroth=0.0, discret=0.0, bourbon=0.0, triniti
=0.0, gentli=0.000540833, digniti=0.0, p=0.0, mobil=0.0, artoo=0.0, h=0.0, en=0.0, tooth=0.0, media
=0.0, runner=0.0, blend=0.0, cruis=0.0, pentagon=0.0, cotton=0.0, moren=0.0, download=0.0, stayin=
0.0, strategi=0.0, crippl=0.0, background=0.0, daili=0.0, tokyo=0.0, reinforc=0.0, oppos=0.0, crush
er=0.0, pigeon=0.0, werewolf=0.0, glove=0.0, fulfil=0.0, mainfram=0.000540833, graviti=0.0, comment
=0.0, streak=0.0, inspect=0.0, laboratori=0.0, stall=0.0, anniversari=0.0, ceas=0.0, meantim=0.0, e
lectron=0.0, motor=0.0, annoy=0.0, reliabl=0.0, pinta=0.0, clementin=0.0, scam=0.0, constantli=0.0,
martial=0.0, puls=0.0, tj=0.0, pickup=0.0, limb=0.0, intim=0.0, purchas=0.0, web=0.0, whistler=0.0,
preacher=0.0, fever=0.0, bijou=0.0, subtl=0.0, fuss=0.0, consum=0.0, non=0.0, nod=0.0, cart=0.0, fe
rrari=0.0, wessex=0.0, kalen=0.0, groceri=0.0, nativ=0.0, scoop=0.0, outer=0.0, slide=0.0, radioact
=0.0, deploy=0.0, oak=0.0, swat=0.0, profess=0.0, auction=0.0, legion=0.0, rug=0.0, headlin=0.0, 17
=0.0, lick=0.0, cargo=0.0, nostril=0.0, daylight=0.0, cornwal=0.0, titan=0.0, rekal=0.0, 8=0.0, nsa
=0.0, spitz=0.0, f=0.0, lamp=0.0, math=0.0, octaviu=0.0, unexpect=0.0, nedri=0.0, tenni=0.0, splend
id=0.0, ritual=0.0, assembl=0.0, row=0.0, olsen=0.0, prep=0.0, 1=0.0, corbett=0.0, andand=0.0, sum=
0.0, lovebird=0, friendship=0.0, tissu=0.001081666, huron=0.0, thirsti=0.0, flea=0.0, edit=0.0, lea
vin=0.0, amen=0.0, val=0.0, classifi=0.0, caen=0.0, outlaw=0.0, reactor=0.0, hesit=0.0, flood=0.0,
teas=0.0, seattl=0.0, therapi=0.0, 60=0.0, ration=0.0, philadelphia=0.0, method=0.0, molecular=0.0,
brooklyn=0.0, overwhelm=0.0, scrape=0.0, disciplin=0.0, autograph=0.0, beneath=0.0, accompani=0.0,
misunderstand=0.0, twentytwo=0.0, asylum=0.0, buljanoff=0.0, chromium=0.0, deliber=0.0, atlanta=0.
0, caviar=0.0, oldest=0.0, pit=0.0, greet=0.0, greek=0.0, comic=0.0, butter=0.0, sting=0.0, thu=0.
0, cape=0.0, webb=0.0, flare=0.0, henc=0.0, encount=0.0, injuri=0.0, item=0.0, undress=0.0, candl=
0.0, estim=0.0, jaeger=0.0, walkin=0.0, sensat=0.0, coron=0.0, pragu=0.0, clerk=0.0, ooh=0.0, drawe
r=0.0, arlyn=0, mankind=0.0, destin=0.0, contrari=0.0, contempt=0.0, lt=0.0, krypton=0.0, virg=0.0,
judgment=0.0, baker=0.0, solar=0.0, mush=0.0, environ=0.0, primit=0.0, psychot=0.0, uwhatu=0.0, rea
list=0.0, moreov=0.0, ghostbust=0.0, certif=0.0, typic=0.0, lion=0.0, histor=0.0, shermin=0.0, obey
=0.0, renard=0.0, arami=0.0, unpleas=0.0, ceil=0.0, spin=0.0, hah=0.0, quadrant=0.0, receipt=0.0, b
anner=0.0, touchi=0.0, rage=0.0, immort=0.0, viciou=0.0, factor=0.0, welli=0.0, rack=0.0, evacu=0.
0, paley=0.0, recept=0.0, scent=0.0, satisfact=0.0, mainten=0.0, flush=0.0, reduc=0.0, elbow=0.0, r
egula=0.0, buddyboy=0.0, transmit=0.0, cohaagen=0.0, thruster=0.0, stroll=0.0, exploit=0.0, delici=
0.0, scheme=0.0, shack=0.0, millionair=0.0, skirt=0.0, dissolv=0.0, urg=0.0, gray=0.0, tens=0.0, st
rangl=0.0, distant=0.0, display=0.0, hitch=0.0, malloy=0.0, inherit=0.0, deaf=0.0, microwav=0.0, ah
hh=0.0, nearest=0.0, tucker=0.0, dutch=0.0, 14=0.0, treasuri=0.0, plumb=0.0, zulu=0.0, exposur=0.0,
scholarship=0.0, gag=0.0, anonym=0.0, plissken=0, snoop=0.0, cater=0.0, goodlook=0.0, traitor=0.0,
neural=0.0, leap=0.0, primari=0.0, morgu=0.0, kiddin=0.0, kilomet=0.0, glimps=0.0, teenag=0.0, eagl
=0.0, sunset=0.0, towel=0.0, tickl=0.0, berkeley=0.0, gruner=0.0, shove=0.0, pipelin=0.0, peach=0.
0, happier=0.0, trifl=0.0, wha=0.0, limp=0.0, irrespons=0.0, humbl=0.0, herr=0.0, household=0.0, ad
vertis=0.0, fingernail=0.0, artifact=0.0, swiss=0.0, bartholomew=0.0, select=0.0, reiben=0.0, sec=
0.0, ambush=0.0, occupi=0.0, grissom=0.0, tomato=0.0, cd=0.0, chancellor=0.0, mhm=0.0, ergo=0.0, br
ill=0.0, sleev=0.0, gotcha=0.0, shhhh=0.0, bree=0, upper=0.0, underestim=0.0, haircut=0.0, bathtub=
0.0, kentucki=0.0, temporari=0.0, molecul=0.0, retriev=0.0, auto=0.0, hardwar=0.0, businessman=0.0,
counterfeit=0.0, preach=0.0, hamburg=0.0, contest=0.0, nun=0.0, somewhat=0.0, peculiar=0.0, despit=
0.0, shutup=0.0, athlet=0.0, verona=0.0, forg=0.0, matur=0.0, diet=0.0, pine=0.0, peasant=0.0, hipp
i=0.0, conor=0.0, mornin=0.0, dial=0.0, distinct=0.0, compart=0.0, grudg=0.0, 4=0.0, pea=0.0, groov
i=0.0, slack=0.0, highland=0.0, heali=0.0, descript=0.0, abb=0.0, random=0.0, aisl=0.0, tin=0.0, ba
ke=0.0, hairi=0.0, slightest=0.0, r=0.0, review=0.0, pole=0.0, oz=0.0, settlement=0.0, sona=0, deep
li=0.0, dug=0.0, sittin=0.0, mckenna=0, cozi=0.0, cairo=0.0, imbecil=0.0, column=0.0, brush=0.0, bo
oz=0.0, virtu=0.0, pinot=0.0, unou=0.0, dyin=0.0, scalp=0.0, exot=0.0, stifler=0, renault=0.0, evol
ut=0.0, compass=0.0, australia=0.0, kremlin=0.0, un=0.0, haunt=0.0, trouser=0.0, hauser=0.0, variab
l=0.0, gestur=0.0, greater=0.0, apollo=0.0, thirtyseven=0.0, barrier=0.0, tramel=0.0, hank=0.0, occ
up=0.0, shooter=0.0, meanwhil=0.0, stain=0.0, asleen=0.0, paden=0.0, graciou=0.0, arlo=0.0, skeleto
n=0.0, startin=0.0, phrase=0.0, musket=0.0, interpol=0.0, timer=0.0, livin=0.0, glare=0.0, humperdi
nck=0.0, me:=0.0, conflict=0.0, hitchhik=0.0, metropoli=0.0, applic=0.0, potato=0.0, transplant=0.
0, simpson=0.0, 13=0.0, nap=0.0, masri=0, raptor=0.0, 22=0.0, bavmorda=0.0, debat=0.0, curv=0.0, rh
yme=0.0, whoop=0.0, zammito=0.0, antidot=0.0, chapter=0.0, conquer=0.0, torch=0.0, gull=0.0, catwom
an=0.0, confidenti=0.0, moor=0.0, overload=0.0, pond=0.0, crocodil=0.0, patriot=0.0, mock=0.0, kros
```

```
s=0.0, breach=0.0, spoon=0.0, smaller=0.0, 3=0.0, reaper=0.0, dell=0.0, giron=0.0, spear=0.0, aspen
=0.0, filthi=0.0, effici=0.0, flirt=0.0, ay=0.0, skipper=0.0, nasal=0.0, macleod=0.0, whaddya=0.0,
codex=0.0, morphin=0.0, gladli=0.0, norton=0.0, trajectori=0.0, expand=0.0, ahm=0, burger=0.0, trea
son=0.0, pension=0.0, disc=0.0, flock=0.0, hike=0.0, gideon=0.0, jona=0.0, faeri=0.0, puriti=0.0, b
reez=0.0, optimist=0.0, soil=0.0, gump=0.0, thread=0.000540833, warhead=0.0, scenario=0.0, specul=
0.0, submit=0.0, bee=0.0, fireman=0.0, paycheck=0.0, meteor=0.0, jorel=0.0, sunshin=0.0, cadillac=
0.0, hood=0.0, unfair=0.0, appleton=0.0, favour=0.0, ambul=0.0, portrait=0.0, acceler=0.0, geronimo
=0.0, stool=0.0, shark=0.0, previou=0.0, persuad=0.0, fatal=0.0, headquart=0.0, quietli=0.0, digest
=0.0, ram=0.0, beef=0.0, divers=0.0, rio=0.0, tir=0.0, mall=0.0, sphere=0.0, 25=0.0, baldwin=0.0, h
armless=0.0, financ=0.0, sattler=0.0, pervert=0.0, blackmail=0.0, grain=0.0, amigo=0.0, diplomat=0.
0, ski=0.0, growth=0.0, hawkin=0.0, champ=0.0, callin=0.0, digit=0.0, fin=0.0, gripe=0.0, duplic=0.
0, fabric=0.0, relay=0.0, skate=0.0, su=0.0, plung=0.0, photon=0.0, thirtysix=0.0, constitut=0.0, l
end=0.0, toontown=0.0, friggin=0.0, tender=0.0, sank=0.0, sh=0.0, perfum=0.0, macaulay=0.0, blackja
ck=0.0, fianc=0.0, guinea=0.0, incident=0.0, von=0.0, mortgag=0.0, scatter=0.0, gorkon=0.0, feather
=0.0, pier=0.0, salesman=0.0, mansion=0.0, sweater=0.0, midg=0.0, noi=0.0, album=0.0, inquiri=0.0,
fountain=0.0, ballet=0.0, iraq=0.0, whine=0.0, casanova=0.0, dread=0.0, liver=0.0, loop=0.0, unowu=
0.0, bribe=0.0, convert=0.0, butch=0.0, f14=0.0, endless=0.0, grenad=0.0, parasit=0.0, ostrich=0.0,
gym=0.0, rheya=0.0, mous=0.0, klute=0, maharajah=0.0, databas=0.0, comedian=0.0, infinit=0.0, outra
g=0.0, stubborn=0.0, seventh=0.0, rogu=0.0, teschmach=0.0, politician=0.0, memphi=0.0, easter=0.0,
denham=0.0, ridg=0.0, canada=0.0, asteroid=0.0, sorrow=0.0, anomali=0.0, invas=0.0, griev=0.0, alab
ama=0.0, yank=0.0, burnt=0.0, choke=0.0, storag=0.0, whew=0.0, seamu=0.0, eighth=0.0, gap=0.0, okla
homa=0.0, reno=0.0, ideal=0.000540833, guitar=0.0, nekhorvich=0.0, footprint=0.0, polish=0.0, heir=
0.0, shrimp=0.0, voyag=0.0, inconveni=0.0, 9=0.000540833, fold=0.0, frog=0.0, barg=0.0, boil=0.0, d
ecenc=0.0, exceed=0.0, belov=0.0, conner=0.0, grief=0.0, carbon=0.0, campaign=0.0, uknowu=0.0, comp
ens=0.0, 23=0.0, madison=0.0, armitag=0.0, claw=0.0, playin=0.0, glade=0.0, consult=0.0, symptom=0.
0, decor=0.0, religion=0.0, phantom=0.0, intact=0.0, zip=0.0, martyr=0.0, hobbi=0.0, probli=0.0, or
acl=0.0, worryin=0.0, pursuit=0.0, insignific=0.0, di=0.0, com=0.0, petti=0.0, concert=0.0, memo=0.
0, treatment=0.0, bucket=0.0, shipment=0.0, knive=0.0, jesuit=0.0, twentyon=0.0, password=0.0, noti
fi=0.0, frost=0.0, depth=0.0, whiski=0.0, napkin=0.0, worley=0.0, johana=0.0, pale=0.0, balconi=0.
0, parachut=0.0, anchor=0.0, manufactur=0.0, couer=0.0, burt=0.0, imit=0.0, sling=0.0, debrief=0.0,
modul=0.0, rorschach=0, coconut=0.0, ugart=0.0, ju=0.0, mare=0.0, insect=0.0, admiss=0.0, y=0.0, ge
rmani=0.0, ladder=0.0, porch=0.0, notion=0.0, gladiat=0.0, payrol=0.0, uptown=0.0, rep=0.0, madman=
0.0, hose=0.0, slight=0.0, panel=0.0, lodg=0.0, transmitt=0.0, glow=0.0, farther=0.0, roper=0.0, al
amo=0.0, cord=0.0, eiffel=0.0, quantum=0.0, signific=0.0, contribut=0.0, stir=0.0, lighten=0.0, lig
hter=0.0, shaft=0.0, immun=0.0, op=0.0, imperfect=0.0, reckon=0.0, react=0.0, lemon=0.0, pasta=0.0,
disabl=0.0, bodhi=0.0, footag=0.0, branch=0.0, meaningless=0.0, symphoni=0.0, uwhyu=0.0, modest=0.
0, allerg=0.0, sauc=0.0, unemploy=0.0, k=0.0, sleepi=0.0, creek=0.0, paperwork=0.0, gown=0.0, 90=0.
0, conscious=0.0, gunshot=0.0, yessir=0.0, poach=0.0, 2000=0.0, th=0.0, hirsch=0.0, squirrel=0.0, e
rror=0.0, smuggl=0.0, decad=0.0, tenth=0.0, slayer=0.0, novemb=0.0, vein=0.0, pinch=0.0, crunch=0.
0, recoveri=0.0, nomin=0.0, ingen=0.0, cappuccino=0.0, septemb=0.0, wax=0.0, charenton=0.0, repriev
=0.0, hyster=0.0, cal=0.0, pacif=0.0, excess=0.0, mourn=0.0, chrissak=0.0, albani=0.0, cellular=0.
0, duncan=0.0, implic=0.0, shred=0.0, cramp=0.0, torn=0.0, garter=0.0, drexl=0.0, aveng=0.0, gibari
an=0.0, affirm=0.0, protocol=0.0, immigr=0.0, analyst=0.0, deceas=0.0, careless=0.0, 18=0.0, second
ari=0.0, smear=0.0, swede=0.0, casualti=0.0, deem=0.0, u=0.0, ramel=0.0, cockroach=0.0, fiction=0.
0, iim=0.0, alik=0.0, courtroom=0.0, disagre=0.0, gibson=0.0, goddess=0.0, umyu=0.0, siddown=0.0, g
lobe=0.0, naiv=0.0, scobi=0.0, composit=0.0, filter=0.0, magda=0.0, sanctuari=0.0, newt=0.0, portal
=0.0, extract=0.0, sunlight=0.0, cledu=0.0, childhood=0.0, rosalin=0.0, jazz=0.0, impati=0.0, justi
fi=0.0, knowi=0.0, oona=0.0, instructor=0.0, complaint=0.0, garber=0.0, sip=0.0, echo=0.0, morlock=
0.0, thug=0.0, obtain=0.0, tch=0.0, herd=0.0, dork=0.0, navig=0.0, doublecross=0.0, interpret=0.0,
rider=0.0, currenc=0.0, 2=0.0, rival=0.0, atom=0.0, slot=0.0, landlord=0.0, bundl=0.0, bundi=0.0, s
hreck=0.0, ohh=0.0, au=0.0, incap=0.0, vic=0.0, gallon=0.0, condemn=0.0, ith=0.0, lantern=0.0, slic
k=0.0, impact=0.0, chevali=0.0, wholl=0.0, supervisor=0.0, earp=0.0, soak=0.0, lust=0.0, dreyfuss=
0.0, ohiggin=0.0, championship=0.0, spooki=0.0, greed=0.0, airk=0.0, viznick=0, tragedi=0.0, kenwor
th=0.0, uhoh=0.0, casual=0.0, ego=0.0, grime=0.0, lit=0.0, awesom=0.0, cuff=0.0, havin=0.0, manipul
=0.0, fuzzi=0.0, daikini=0.0, shh=0.0, talley=0, serial=0.0, hunch=0.0, thankyou=0.0, massiv=0.0, n
evada=0.0, mcp=0.0, dash=0.0, ir=0.0, psych=0.0, wretch=0.0, earthquak=0.0, burk=0.0, keeper=0.0, b
loke=0.0, caravaggio=0.0, spiderman=0.0, oppon=0.0, tennesse=0.0, tub=0.0, graveyard=0.0, einstein=
0.0, binocular=0.0, keepin=0.0, surfer=0.0, wager=0.0, plead=0.0, mikey=0.0, chemistri=0.0, gratitu
d=0.0, brazil=0.0, scandal=0.0, miyamoto=0.0, dat=0.0, buster=0.0, virtual=0.0, roston=0.0, orphan=
0.0, countess=0.0, treati=0.0, bind=0.0, reconstruct=0.0, expertis=0.0, ich=0.0, corn=0.0, reli=0.
0, waddaya=0, ventur=0.0, religi=0.0, hinkley=0.0, suspend=0.0, pri=0.0, campbel=0.0, britain=0.0,
bulli=0.0, taupin=0.0, expir=0.0, carlotta=0.0, overlook=0.0, po=0.001081666, discourag=0.0, nasa=
0.0, ladyship=0.0, carrot=0.0, xray=0.0, bronx=0.0, layer=0.0, annual=0.0, strasser=0.0, wooden=0.
0, gorman=0.0, thunder=0.0, cult=0.0, tremend=0.0, traver=0, adapt=0.0, rice=0.0, doth=0.0, udal=0.
0, endur=0.0, legitim=0.0, papa=0.0, instal=0.0, enhanc=0.0, prenup=0.0, boulevard=0.0, dreamer=0.
0, lawn=0.0, merit=0.0, transact=0.0, hoover=0.0, defi=0.0, intel=0.0, exil=0.0, hee=0.0, heh=0.0,
tricki=0.0, quicker=0.0, yakushova=0.0, bypass=0.0, exposit=0.0, spiritu=0.0, limo=0.0, dumbest=0.
0, robot=0.0, thigh=0.0, insight=0.0, length=0.0, spain=0.0, android=0.0, ing=0.001622499, bro=0.0,
l=0.0, guild=0.0, deepest=0.0, oldfashion=0.0, helpless=0.0, pillow=0.0, sweeney=0.0, tragic=0.0, m
onkeybon=0.0, monk=0.0, gloriou=0.0, sixth=0.0, fugit=0.0, 3000=0.0, sybok=0.0, marvosa=0.0, restle
ss=0.0, birdson=0, whatsoev=0.0, grang=0.0, resolv=0.0, syndrom=0.0, gent=0.0, fridg=0.0, starter=
0.0, knowin=0.0, sought=0.0, discharg=0.0, marseil=0.0, skunk=0.0, crate=0.0, rash=0.0, powwow=0.0,
stumbl=0.0, trucker=0.0, wimp=0.0, kittridg=0.0, schmuck=0.0, lonesom=0.0, harass=0.0, stalker=0.0,
seed=0.0, mushu=0.0, brigad=0.0, hartman=0.0, cochran=0.0, sunk=0.0, taho=0.0, pard=0.0, silo=0.0,
amnesia=0.0, 300=0.0, grammi=0, resembl=0.0, fetch=0.0, que=0.0, malfunct=0.0, fright=0.0, pepper=
0.0, mccaffrey=0.0, spike=0.0, vapor=0.0, seminar=0.0, vietnam=0.0, vital=0.0, buyer=0.0, microscop
=0.0, tron=0.0, weep=0.0, michigan=0.0, uimu=0.0, egyptian=0.0, sophomor=0.0, iranoff=0.0, tsch=0,
artifici=0.0, buffalo=0.0, session=0.0, patron=0.0, darkest=0.0, sweeter=0.0, perspect=0.0, academ=
0.0, theme=0.0, bookstor=0.0, givin=0.0, falcon=0.0, despair=0.0, serafin=0.0, droid=0.0, reman=0,
wheelchair=0.0, butterfli=0.0, exagger=0.0, almighti=0.0, persist=0.0, warehous=0.0, econom=0.0, br
utal=0.0, backyard=0.0, lifestyl=0.0, ambiti=0.0, shorter=0.0, hemingway=0.0, supernatur=0.0, slim=
0.0, bent=0.0, partnership=0.0, claremont=0.0, spose=0.0, intrigu=0.0, autom=0.000540833, bishop=0.
0, beryllium=0.0, consent=0.0, handcuff=0.0, bartend=0.0, grandpa=0.0, yeh=0.0, d6=0.0, physician=
0.0, plasma=0.000540833, threepio=0.0, geordi=0.0, sheep=0.0, postcard=0.0, comparison=0.0, underwa
t=0.0, carpet=0.0, gazzo=0.0, embrac=0.0, skywalk=0.0, lightn=0.0, finer=0.0, wacko=0.0, basi=0.0,
elder=0.0, kindli=0.0, antiqu=0.0, cleaner=0.0, cartwright=0.0, vibe=0.0, fastest=0.0, fortytwo=0.
0, keen=0.0, luc=0.0, furnitur=0.0, snicker=0.0, burden=0.0, unlock=0.0, crummi=0.0, decemb=0.0, ac
cur=0.0, altitud=0.0, substanc=0.0, coup=0.0, minim=0.0, bellerophon=0.0, is:=0.0, richest=0.0, imi
m=0.0, diz=0.0, ridin=0.0, plaza=0.0, bolshevik=0.0, agit=0.0, cuban=0.0, tuck=0.0, imperi=0.0, par
lor=0.0, sunris=0.0, elektra=0.0, howl=0.0, carpent=0.0, farragut=0.0, lime=0.0, lapd=0.0, verifi=
0.0, bench=0.0, ravel=0.0, sofa=0.0, hassl=0.0, hath=0.0, difficulti=0.0, connecticut=0.0, mercuri=
0.0, consol=0.0, coil=0.0, waltz=0.0, geek=0.0, pardner=0.0, donat=0.0, asta=0.0, appetit=0.0, sten
ograph=0.0, 1000=0.0, connor=0.0027041640000000002, infant=0.0, dya=0.0, amend=0.0, lookout=0.0, de
tour=0.0, reconsid=0.0, suprem=0.0, nba=0, penalti=0.0, statist=0.0, breakin=0.0, motorcycl=0.0, po
llux=0, paw=0.0, ambit=0.0, tension=0.0, accent=0.0, junki=0.0, richi=0.0, uareu=0.0, et=0.0, endan
g=0.0, arrow=0.0, dugan=0.0, sensibl=0.0, creativ=0.0, role=0.0, dusti=0.0, disconnect=0.000540833,
mantini=0.0, sponsor=0.0, byeby=0.0, panick=0.0, thirtyeight=0.0, homeless=0.0, gingham=0.0, morg=
0.0, brumbi=0.0, kensington=0.0, overtim=0.0, eleg=0.0, bingi=0, ui=0.0, slit=0.0, strateg=0.0, mes
seng=0.0, courtesi=0.0, jad=0.0, debi=0.0, souvenir=0.0, banish=0.0, dam=0.0, vain=0.0, kiddo=0.001
081666, assess=0.0, munro=0.0, zander=0.0, shiit=0.0, vagu=0.0, fuse=0.000540833, retreat=0.0, khan
=0.0, braslow=0.0, heller=0.0, pankot=0.0, clanton=0.0, multipl=0.0, lyin=0.0, carv=0.0, slam=0.0,
razinin=0.0, proven=0.0, essenc=0.0, arena=0.0, duh=0.0, substanti=0.0, ninja=0.0, blush=0.0, nuthi
n=0.0, sneakin=0.0, northern=0.0, delus=0.0, rambo=0.0, feller=0.0, leash=0.0, bacon=0.0, sixtyf=0.
0, winch=0.0, vase=0.0, vitelli=0.0, climat=0.0, wage=0.0, sorsha=0.0, cosmic=0.0, gambler=0.0, lou
der=0.0, hunger=0.0, enchant=0.0, gasolin=0.0, 31=0.0, dreamt=0.0, unnecessari=0.0, interv=0.0, vac
uum=0.0, jealousi=0.0, dodg=0.0, shook=0.0, reynold=0.0, lowri=0.0, frequent=0.0, housekeep=0.0, br
oadcast=0.0, squat=0.0, hmmmm=0.0, coupon=0.0, striker=0.0, havana=0.0, 5000=0.0, lysin=0.0, happie
st=0.0, copper=0.0, dentist=0.0, theft=0.0, quell=0.0, kopalski=0.0, curb=0.0, powel=0.0, miz=0.0,
prais=0.0, nevertheless=0.0, adm=0.0, peddl=0.0, craft=0.0, soda=0.0, thoroughli=0.0, spite=0.0, de
lli=0.0, rodent=0.0, plumber=0.0, refriger=0.0, boast=0.0, visibl=0.000540833, merchandis=0.0, mand
rak=0.0, gulf=0.0, berri=0.0, bummer=0.0, fender=0.0, weed=0.0, disrupt=0.0, whend=0.0, sensenet=0,
bizarr=0.0, v=0.0, resort=0.0, interrog=0.0, swipe=0.0, necklac=0.0, gild=0, 70=0.0, jeanloui=0.0,
momma=0.0, ralli=0.0, flaemm=0.0, yoda=0.0, disgrac=0.0, zowi=0.0, prosecut=0.0, tamper=0.0, oui=0.
0, clone=0.0, extermin=0.0, cuba=0.0, crude=0.0, jog=0.0, largest=0.0, heavili=0.0, sarri=0.0, herb
=0.0, 99=0.0, chump=0.0, siberia=0.0, wisdom=0.0, reassur=0.0, colet=0.0, happenin=0.0, tuna=0.0, o
oooh=0.0, janosz=0.0, bandag=0.000540833, carriag=0.0, behalf=0.0, webster=0.0, gaff=0.0, slob=0.0,
clau=0.0, ab=0.0, reader=0.0, smartest=0.0, buttercup=0.0, bubbl=0.0, cartoon=0.0, hollow=0.0, dobi
sch=0.0, corvi=0.0, inclin=0.0, sultenfuss=0.0, cremat=0.0, peke=0.0, standin=0.0, lazi=0.0, columb
ia=0.0, onion=0.0, varieti=0.0, hypocrit=0.0, sallah=0, recreat=0.0, stead=0.0, fragil=0.0, rail=0.
0, diner=0.0, kuato=0.0, deflector=0.0, southern=0.0, undead=0.0, frontier=0.0, rembrandt=0.0, puff
=0.0, endors=0.0, zorg=0.0, overnight=0.0, trailer=0.0, harmsway=0.0, fiance=0.0, mercutio=0.0, nur
tur=0.0, experiment=0.0, sicili=0.0, fhloston=0.0, balloon=0.0, balboa=0.0, burbag=0.0, mississippi
=0.0, nintendo=0.0, elimin=0.0, reagan=0.0, hoop=0.0, uyouru=0.0, dumber=0.0, vitamin=0.0, warden=
0.0, revel=0.0, bookkeep=0.0, bold=0.0, uthinku=0.0, stallion=0.0, perk=0.0, flu=0.0, shatter=0.0,
montoya=0.0, yield=0.0, derang=0.0, munci=0.0, thatthat=0.0, dew=0.0, socal=0.0, drain=0.0, ireland
=0.0, neutron=0.0, grid=0.000540833, duct=0.0, mayonnais=0.0, intercept=0.000540833, harvest=0.0, a
ft=0.0, lib=0.0, muddi=0.0, hideou=0.0, insert=0.0, oconnel=0.0, noisi=0.0, harvard=0.0, lure=0.0,
barbecu=0.0, cockpit=0.0, ancestor=0.0, precaut=0.0, flop=0.0, outstand=0.0, drivin=0.0, disposit=
0.0, unstabl=0.0, dent=0.0, baku=0, one:=0.0, kelson=0, plaintiff=0, twofac=0.0, hum=0.0, hut=0.0,
pre=0.0, freshman=0.0, formula=0.0, pinkerton=0.0, hump=0.0, sentri=0.0, stalk=0.0, kessler=0.0, ps
ychiatr=0.0, theoret=0.0, crusad=0.0, inmat=0.0, hitter=0.0, myth=0.0, extort=0.0, climber=0.0, cau
tion=0.0, surgic=0.0, burial=0.0, silicon=0.0, freelanc=0.0, dummi=0.0, tech=0.000540833, lethal=0.
0, manson=0.0, lookit=0.0, altar=0.0, tombston=0.0, suffici=0.0, pasadena=0.0, dimens=0.0, prosecut
or=0.0, biolog=0.0, babysit=0.0, goodman=0.0, jahn=0.0, barber=0.0, shhhhh=0.0, grub=0.0, lobster=
0.0, rebuild=0.0, lent=0.0, gurney=0.0, tranquil=0.0, brotherhood=0.0, sang=0.0, defect=0.0, metabo
l=0.0, arson=0.0, blender=0.0, overboard=0.0, lash=0.0, perpetu=0.0, vou=0.0, intuit=0.0, abduct=0.
0, constantinopl=0.0, trubshaw=0.0, spaghetti=0.0, nexu=0.0, two:=0.0, jaw=0.0, lawsuit=0.0, doorwa
y=0.0, indict=0.0, demolit=0.0, guinan=0.0, peak=0.0, philosoph=0.0, divert=0.0, cottag=0.0, soran=
0.0, webber=0.0, coop=0.0, wh=0.0, wu=0.0, subscrib=0.0, yengees=0.0, poss=0.0, absenc=0.0, kitten=
0.0, octob=0.0, wealthi=0.0, ironi=0.0, argo=0.0, blink=0.0, delic=0.0, deuc=0.0, pumpkin=0.0, bode
ga=0, wheat=0.0, pitcher=0.0, mamma=0.0, foster=0.0, pub=0.0, vegetarian=0.0, garrison=0.0, grammoo
=0.0, chimney=0.0, bikini=0.0, richter=0.0, psychopath=0.0, fling=0.0)
```

For example, the fastest way to find the frequency of "hey" in the movie *The Terminator* is to access the `'hey'` item from its row. Check the original table to see if this worked for you!

In [314]:
```
row_for_title('the terminator').item('hey')
```

Out[314]:
```
0.000540833
```

Question 1.1

Set `expected_row_sum` to the number that you **expect** will result from summing all proportions in each row, excluding the first six columns.

In [315]:
```
# Set row_sum to a number that's the (approximate) sum of each row of word proportions.
expected_row_sum = 1
```

In [316]:
```
ok.grade("q1_1");
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[ooooooooook] 100.0% passed
```

This dataset was extracted from a dataset from Cornell University. After transforming the dataset (e.g., converting the words to lowercase, removing the naughty words, and converting the counts to frequencies), we created this new dataset containing the frequency of 5000 common words in each movie.

In [317]:
```
print('Words with frequencies:', movies.drop(np.arange(6)).num_columns)
print('Movies with genres:', movies.num_rows)
```

```
Words with frequencies: 5000
Movies with genres: 242
```

1.1. Word Stemming

The columns other than "Title", "Genre", "Year", "Rating", "# Votes" and "# Words" in the `movies` table are all words that appear in some of the movies in our dataset. These words have been *stemmed*, or abbreviated heuristically, in an attempt to make different inflected forms of the same base word into the same string. For example, the column "manag" is the sum of proportions of the words "manage", "manager", "managed", and "managerial" (and perhaps others) in each movie. This is a common technique used in machine learning and natural language processing.

Stemming makes it a little tricky to search for the words you want to use, so we have provided another table that will let you see examples of unstemmed versions of each stemmed word. Run the code below to load it.

In [318]:
```
# Just run this cell.
vocab_mapping = Table.read_table('stem.csv')
stemmed = np.take(movies.labels, np.arange(3, len(movies.labels)))
vocab_table = Table().with_column('Stem', stemmed).join('Stem', vocab_mapping)
vocab_table.take(np.arange(1100, 1110))
```

Out[318]:

Stem	Word
blame	blamed
blame	blame
blank	blanks
blank	blank
blank	blankness
blanket	blanket
blanket	blankets
blast	blasting
blast	blast
blast	blasted

Question 1.1.1

Assign `stemmed_message` to the stemmed version of the word "alternating".

In [319]:
```
stemmed_message = vocab_table.where('Word', 'alternating').column('Stem').item(0)
stemmed_message
```
Out[319]: 'altern'

In [320]:
```
ok.grade("q1_1_1");
```
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[ooooooooook] 100.0% passed
```

Question 1.1.2

Assign `unstemmed_run` to an array of words in `vocab_table` that have "run" as its stemmed form.

In [321]:
```
unstemmed_run = vocab_table.where('Stem', 'run').column('Word')
unstemmed_run
```
Out[321]: array(['runs', 'running', 'run', 'runned', 'runnings'], dtype='<U17')

In [322]:
```
ok.grade("q1_1_2");
```
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 2
    Failed: 0
[ooooooooook] 100.0% passed
```

Question 1.1.3

Which word in `vocab_table` was shortened the most by this stemming process? Assign `most_shortened` to the word. If there are multiple words, use the word whose first letter is latest in the alphabet (so if your options are albatross or batman, you should pick batman).

It's an example of how heuristic stemming can collapse two unrelated words into the same stem (which is bad, but happens a lot in practice anyway).

In [323]:
```
# In our solution, we found it useful to first make an array
# containing the number of characters that was
# chopped off of each word in vocab_table, but you don't have
# to do that.
len_stem = vocab_table.apply(len, 'Stem')
len_word = vocab_table.apply(len, 'Word')
newtbl = vocab_table.with_column('Added length', len_word - len_stem)

most_shortened_table = newtbl.where('Added length', max(newtbl.column('Added length'))).sort('Word'
, descending = True)
most_shortened = most_shortened_table.column('Word').item(0)

# This will display your answer and its shortened form.
vocab_table.where('Word', most_shortened)
```
Out[323]:

Stem	Word
respons	responsibilities

In [324]:
```
ok.grade("q1_1_3");
```
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[ooooooooook] 100.0% passed
```

1.2. Splitting the dataset

We're going to use our `movies` dataset for two purposes.

1. First, we want to *train* movie genre classifiers.
2. Second, we want to *test* the performance of our classifiers.

Hence, we need two different datasets: *training* and *test*.

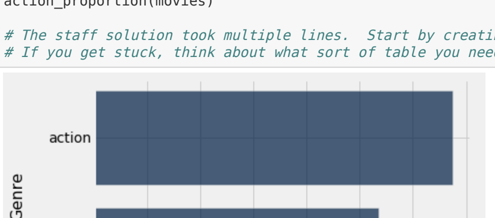
The purpose of a classifier is to classify unseen data that is similar to the training data. Therefore, we must ensure that there are no movies that appear in both sets. We do so by splitting the dataset randomly. The dataset has already been permuted randomly, so it's easy to split. We just take the top for training and the rest for test.

Run the code below (without changing it) to separate the datasets into two tables.

In [325]:
```
# Here we have defined the proportion of our data
# that we want to designate for training as 17/20ths
# of our total dataset.  3/20ths of the data is
# reserved for testing.

training_proportion = 17/20

num_movies = movies.num_rows
num_train = int(num_movies * training_proportion)
num_test = num_movies - num_train

train_movies = movies.take(np.arange(num_train))
test_movies = movies.take(np.arange(num_train, num_movies))

print("Training: ",   train_movies.num_rows, ";",
      "Test: ",       test_movies.num_rows)
```
```
Training:  205 ; Test:  37
```

Question 1.2.1

Draw a horizontal bar chart with two bars that show the proportion of Action movies in each dataset. Complete the function `action_proportion` first; it should help you create the bar chart.

In [326]:
```
def action_proportion(table):
    # Return the proportion of movies in a table that have the Action genre.
    return table.group('Genre').barh('Genre')
action_proportion(movies)

# The staff solution took multiple lines.  Start by creating a table.
# If you get stuck, think about what sort of table you need for barh to work
```


2. K-Nearest Neighbors - A Guided Example

K-Nearest Neighbors (k-NN) is a classification algorithm. Given some numerical *attributes* (also called *features*) of an unseen example, it decides whether that example belongs to one or the other of two categories based on its similarity to previously seen examples. Predicting the category of an example is called *labeling*, and the predicted category is also called a *label*.

An attribute (feature) we have about each movie is *the proportion of times a particular word appears in the movies*, and the labels are two movie genres: romance and action. The algorithm requires many previously seen examples for which both the attributes and labels are known: that's the `train_movies` table.

To build understanding, we're going to visualize the algorithm instead of just describing it.

2.1. Classifying a movie

In k-NN, we classify a movie by finding the `k` movies in the *training set* that are most similar according to the features we choose. We call those movies with similar features the *nearest neighbors*. The k-NN algorithm assigns the movie to the most common category among its `k` nearest neighbors.

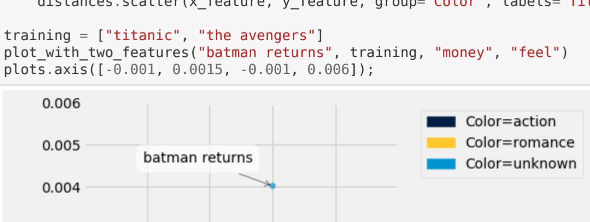
Let's limit ourselves to just 2 features for now, so we can plot each movie. The features we will use are the proportions of the words "money" and "feel" in the movie. Taking the movie "Batman Returns" (in the test set), 0.000502 of its words are "money" and 0.004016 are "feel". This movie appears in the test set, so let's imagine that we don't yet know its genre.

First, we need to make our notion of similarity more precise. We will say that the *distance* between two movies is the straight-line distance between them when we plot their features in a scatter diagram. This distance is called the Euclidean ("yoo-KLID-ee-un") distance, whose formula is $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

For example, in the movie *Titanic* (in the training set), 0.0009768 of all the words in the movie are "money" and 0.0017094 are "feel". Its distance from *Batman Returns* on this 2-word feature set is $\sqrt{(0.000502 - 0.0009768)^2 + (0.004016 - 0.0017094)^2} \approx 0.00235496$. (If we included more or different features, the distance could be different.)

A third movie, *The Avengers* (in the training set), is 0 "money" and 0.001115 "feel".

The function below creates a plot to display the "money" and "feel" features of a test movie and some training movies. As you can see in the result, *Batman Returns* is more similar to *Titanic* than to *The Avengers* based on these features. However, we know that *Batman Returns* and *The Avengers* are both action movies, so intuitively we'd expect them to be more similar. Unfortunately, that isn't always the case. We'll discuss this more later.

In [327]:
```
# Just run this cell.
def plot_with_two_features(test_movie, training_movies, x_feature, y_feature):
    """Plot a test movie and training movies using two features."""
    test_row = row_for_title(test_movie)
    distances = Table().with_columns(
            x_feature, [test_row.item(x_feature)],
            y_feature, [test_row.item(y_feature)],
            'Color',   ['unknown'],
            'Title',   [test_movie]
        )
    for movie in training_movies:
        row = row_for_title(movie)
        distances.append([row.item(x_feature), row.item(y_feature), row.item('Genre'), movie])
    distances.scatter(x_feature, y_feature, group='Color', labels='Title', s=30)

training = ["titanic", "the avengers"]
plot_with_two_features("batman returns", training, "money", "feel")
plots.axis([-0.001, 0.0015, -0.001, 0.006]);
```


Question 2.1.1

Compute the distance between the two action movies, *Batman Returns* and *The Avengers*, using the `money` and `feel` features only. Assign it the name `action_distance`.

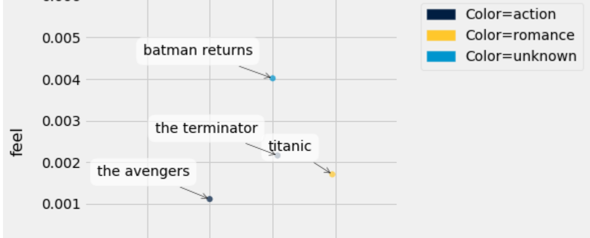
Note: If you have a row, you can use `item` to get a value from a column by its name. For example, if `r` is a row, then `r.item("Genre")` is the value in column `"Genre"` in row `r`.

Hint: Remember the function `row_for_title`, redefined for you below.

In [328]:
```
title_index = movies.index_by('Title')
def row_for_title(title):
    """Return the row for a title, similar to the following expression (but faster)

    movies.where('Title', title).row(0)
    """
    return title_index.get(title)[0]

batman = row_for_title("batman returns")
avengers = row_for_title("the avengers")
action_distance = ((batman.item('money') - avengers.item('money'))**2 + (batman.item('feel') - aven
gers.item('feel'))**2)**0.5
action_distance
```
Out[328]: 0.0029437356216700243

In [329]:
```
ok.grade("q1_2_1");
```
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[ooooooooook] 100.0% passed
```

Below, we've added a third training movie, *The Terminator*. Before, the point closest to *Batman Returns* was *Titanic*, a romance movie. However, now the closest point is *The Terminator*, an action movie.

In [330]:
```
training = ["the avengers", "titanic", "the terminator"]
plot_with_two_features("batman returns", training, "money", "feel")
plots.axis([-0.001, 0.0015, -0.001, 0.006]);
```


Question 2.1.2

Complete the function `distance_two_features` that computes the Euclidean distance between any two movies, using two features. The last two lines call your function to show that *Batman Returns* is closer to *The Terminator* than *The Avengers*.

In [331]:
```
def distance_two_features(title0, title1, x_feature, y_feature):
    """Compute the distance between two movies with titles title0 and title1

    Only the features named x_feature and y_feature are used when computing the distance.
    """
    row0 = row_for_title(title0)
    row1 = row_for_title(title1)
    distance = ((row0.item(x_feature) - row1.item(x_feature))**2 + (row0.item(y_feature) - row1.ite
m(y_feature))**2)**0.5
    return distance

for movie in make_array("the terminator", "the avengers"):
    movie_distance = distance_two_features(movie, "batman returns", "money", "feel")
    print(movie, 'distance:\t', movie_distance)
```
```
the terminator distance:         0.0018531387547749904
the avengers distance:   0.0029437356216700243
```

In [332]:
```
ok.grade("q2_1_2");
```
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 2
    Failed: 0
[ooooooooook] 100.0% passed
```

Question 2.1.3

Define the function `distance_from_batman_returns` so that it works as described in its documentation.

Note: Your solution should not use arithmetic operations directly. Instead, it should make use of existing functionality above!

In [333]:
```
def distance_from_batman_returns(title):
    """The distance between the given movie and "batman returns", based on the features "money" and
 "feel".

    This function takes a single argument:
      title: A string, the name of a movie.
    """

    return distance_two_features(title, 'batman returns', 'money', 'feel')
```

In [334]:
```
ok.grade("q2_1_3");
```
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[ooooooooook] 100.0% passed
```

Question 2.1.4

Using the features `"money" and "feel"`, what are the names and genres of the 7 movies in the **training set** closest to *Batman Returns*? To answer this question, make a table named `close_movies` containing those 7 movies with columns `"Title"`, `"Genre"`, `"money"`, and `"feel"`, as well as a column called `"distance from batman"` that contains the distance from *Batman Returns*. The table should be **sorted in ascending order by `distance from batman`**.

In [335]:
```
# The staff solution took multiple lines.
tbl_with_distance = train_movies.select('Title', 'Genre', 'money', 'feel').with_column('distance fr
om batman', train_movies.apply(distance_from_batman_returns, 'Title')).sort('distance from batman')
close_movies = tbl_with_distance.take(np.arange(1,8))
close_movies
```
Out[335]:

Title	Genre	money	feel	distance from batman
the fisher king	romance	0.000618302	0.00350371	0.000525386
broadcast news	romance	0.000136668	0.00355337	0.000589542
hellboy	action	0	0.00335008	0.000833991
as good as it gets	romance	0.000523104	0.00313862	0.000877696
spider-man	action	0.000347343	0.00312608	0.000903318
harold and maude	romance	0	0.00302343	0.00111235
the wedding date	romance	0.00127227	0.00318066	0.00113631

In [336]:
```
ok.grade("q2_1_4");
```
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 2
    Failed: 0
[ooooooooook] 100.0% passed
```

Question 2.1.5

Next, we'll clasify *Batman Returns* based on the genres of the closest movies.

To do so, define the function `most_common` so that it works as described in its documentation below.

In [337]:
```
def most_common(label, table):
    """The most common element in a column of a table.

    This function takes two arguments:
      label: The label of a column, a string.
      table: A table.

    It returns the most common value in that column of that table.
    In case of a tie, it returns any one of the most common values
    """
    return table.select(label).group(label).sort('count', descending=True).column(label).item(0)

# Calling most_common on your table of 7 nearest neighbors classifies
# "batman returns" as a romance movie, 5 votes to 2.
most_common('Genre', close_movies)
```
Out[337]: 'romance'

In [338]:
```
ok.grade("q2_1_5");
```
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 2
    Failed: 0
[ooooooooook] 100.0% passed
```

Congratulations are in order -- you've classified your first movie! However, we can see that the classifier doesn't work too well since it categorized *Batman Returns* as a romance movie (unless you count the bromance between Alfred and Batman). Let's see if we can do better!

Checkpoint (Due 11/22)

Congratulations, you have reached the first checkpoint! Run the submit cell below to generate the checkpoint submission.

To get full credit for this checkpoint, you must pass all the public autograder tests above this cell.

In [339]:
```
_ = ok.submit()
```
```
Saving notebook... Saved 'project3.ipynb'.
Submit... 100% complete
Submission successful for user: epere@berkeley.edu
URL: https://okpy.org/cal/data8/fa19/project3/submissions/99v25P
```

3. Features

Now, we're going to extend our classifier to consider more than two features at a time.

Euclidean distance still makes sense with more than two features. For `n` different features, we compute the difference between corresponding feature values for two movies, square each of the `n` differences, sum up the resulting numbers, and take the square root of the sum.

Question 3.1

Write a function called `distance` to compute the Euclidean distance between two **arrays** of **numerical** features (e.g. arrays of the proportions of times that different words appear). The function should be able to calculate the Euclidean distance between two arrays of arbitrary (but equal) length.

Next, use the function you just defined to compute the distance between the first and second movie in the training set *using all of the features*. (Remember that the first six columns of your tables are not features.)

Note: To convert rows to arrays, use `np.array`. For example, if `t` was a table, `np.array(t.row(0))` converts row 0 of `t` into an array.

In [340]:
```
def distance(features_array1, features_array2):
    """The Euclidean distance between two arrays of feature values."""
    distance = (sum((features_array1 - features_array2)**2))**0.5
    return distance

tbl_without_firstrows = train_movies.drop('Title','Genre','Year','Rating','# Votes','# Words')
distance_first_to_second = distance(np.array(tbl_without_firstrows.row(0)), np.array(tbl_without_fi
rstrows.row(1)))
distance_first_to_second
```
Out[340]: 0.03869489784999248

In [341]:
```
ok.grade("q3_1");
```
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 3
    Failed: 0
[ooooooooook] 100.0% passed
```

3.1. Creating your own feature set

Unfortunately, using all of the features has some downsides. One clear downside is *computational* -- computing Euclidean distances just takes a long time when we have lots of features. You might have noticed that in the last question!

So we're going to select just 20. We'd like to choose features that are very *discriminative*. That is, features which lead us to correctly classify as much of the test set as possible. This process of choosing features that will make a classifier work well is sometimes called *feature selection*, or, more broadly, *feature engineering*.

Question 3.0.1

In this question, we will help you get started on selecting more effective features for distinguishing romance from action movies. The plot below (generated for you) shows the average number of times each word occurs in a romance movie on the horizontal axis and the average number of times it occurs in an action movie on the vertical axis.

alt text

The following questions ask you to interpret the plot above. For each question, select one of the following choices and assign its number to the provided name.

```
1. The word is uncommon in both action and romance movies
2. The word is common in action movies and uncommon in romance movies
3. The word is uncommon in action movies and common in romance movies
4. The word is common in both action and romance movies
5. It is not possible to say from the plot
```

What properties does a word in the bottom left corner of the plot have? Your answer should be a single integer from 1 to 5, corresponding to the correct statement from the choices above.

In [342]:
```
bottom_left = 1 #or all questions below are 5 bc plot talks about average, not individual movies.
```

In [343]:
```
ok.grade("q3_0_1");
```
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[ooooooooook] 100.0% passed
```

Question 3.0.2

What properties does a word in the bottom right corner have?

In [344]:
```
bottom_right = 3
```

In [345]:
```
ok.grade("q3_0_2");
```
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[ooooooooook] 100.0% passed
```

Question 3.0.3

What properties does a word in the top right corner have?

In [346]:
```
top_right = 4
```

In [347]:
```
ok.grade("q3_0_3");
```
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[ooooooooook] 100.0% passed
```

Question 3.0.4

What properties does a word in the top left corner have?

In [348]:
```
top_left = 2
```

In [349]:
```
ok.grade("q3_0_4");
```
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[ooooooooook] 100.0% passed
```

Question 3.0.5

If we see a movie with a lot of words that are common for action movies but uncommon for romance movies, what would be a reasonable guess about the genre of the movie? Assign `movie_genre` to the number corresponding to your answer:

```
1. It is an action movie.
2. It is a romance movie.
```

In [350]:
```
movie_genre_guess = 1
```

In [351]:
```
ok.grade("q3_0_5");
```
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[ooooooooook] 100.0% passed
```

Question 3.1.1

Using the plot above, choose 20 common words that you think might let you distinguish between romance and action movies. Make sure to choose words that are frequent enough that every movie contains at least one of them. Don't just choose the 20 most frequent, though... you can do much better.

You might want to come back to this question later to improve your list, once you've seen how to evaluate your classifier.

In [352]:
```
# Set my_20_features to an array of 20 features (strings that are column labels)

#my_20_features = make_array('start', 'new', 'hear','rememb', 'show', 'same', 'walk', 'next', 'fe
w', 'understand', 'anoth', 'car', 'afraid', 'troubl', 'came', 'best', 'pay', 'street', 'women', 'bu
y')
#my_20_features = make_array('captain', 'ship', 'cop','head', 'hour', 'power', 'job', 'weve', 'gott
a', 'mani', 'mom', 'mother', 'hello', 'wife', 'happi', 'school', 'huh', 'ladi', 'beauti', 'marri')
my_20_features = make_array('marri', 'captain', 'power', 'run', 'world', 'move', 'three', 'nice',
'boy', 'wouldnt',
                            'enough', 'miss', 'head', 'cours', 'done', 'turn', 'yourself', 'home',
'job', 'knew')
# Select the 20 features of interest from both the train and test sets
train_20 = train_movies.select(my_20_features)
test_20 = test_movies.select(my_20_features)
```

In [353]:
```
ok.grade("q3_1_1");
```
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 5
    Failed: 0
[ooooooooook] 100.0% passed
```

This test makes sure that you have chosen words such that at least one appears in each movie. If you can't find words that satisfy this test just through intuition, try writing code to print out the titles of movies that do not contain any words from your list, then look at the words they do contain.

Question 3.1.2

In two sentences or less, describe how you selected your features.

I looked for words that were furthest from the trend line to make the distinction clear on whether they belong in an action or romance film, and I focused on words toward the top right so that the occurrence was high enough for the word to have occurred more than once, making it a better predictor.

Next, let's classify the first movie from our test set using these features. You can examine the movie by running the cells below. Do you think it will be classified correctly?

In [354]:
```
print("Movie:")
test_movies.take(0).select('Title', 'Genre').show()
print("Features:")
test_20.take(0).show()
```
Movie:

Title	Genre
the mummy	action

Features:

marri	captain	power	run	world	move	three	nice	boy	wouldnt	enough	
0.000321027	0	0	0	0.000321027	0	0.000642055	0.000321027	0.000321027	0.000321027	0	0.0

As before, we want to look for the movies in the training set that are most like our test movie. We will calculate the Euclidean distances from the test movie (using the 20 selected features) to all movies in the training set. You could do this with a `for` loop, but to make it computationally faster, we have provided a function, `fast_distances`, to do this for you. Read its documentation to make sure you understand what it does. (You don't need to understand the code in its body unless you want to.)

In [355]:
```
# Just run this cell to define fast_distances.

def fast_distances(test_row, train_table):
    """Return an array of the distances between test_row and each row in train_rows.

    Takes 2 arguments:
      test_row: A row of a table containing features of one
        test movie (e.g., test_20.row(0)).
      train_table: A table of features (for example, the whole
        table train_20)."""
    assert train_table.num_columns < 50, "Make sure you're not using all the features of the movies
 table."
    counts_matrix = np.asmatrix(train_table.columns).transpose()
    diff = np.tile(np.array(test_row), [counts_matrix.shape[0], 1]) - counts_matrix
    np.random.seed(0) # For tie breaking purposes
    distances = np.squeeze(np.asarray(np.sqrt(np.square(diff).sum(1))))
    eps = np.random.uniform(size=distances.shape)*1e-10 #Noise for tie break
    distances = distances + eps
    return distances
```

Question 3.1.3

Use the `fast_distances` function provided above to compute the distance from the first movie in the test set to all the movies in the training set, **using your set of 20 features**. Make a new table called `genre_and_distances` with one row for each movie in the training set and two columns:

- The `"Genre"` of the training movie
- The `"Distance"` from the first movie in the test set

Ensure that `genre_and_distances` is **sorted in increasing order by distance to the first test movie**.

In [356]:
```
# The staff solution took multiple lines of code.
fast_array = fast_distances(test_20.row(0), train_20)
genre_and_distances = Table().with_column('Genre', train_movies.column('Genre')).with_column('Dista
nce', fast_array).sort('Distance')
genre_and_distances
```
Out[356]:

Genre	Distance
romance	0.00169342
romance	0.00171342
action	0.00172382
romance	0.00184509
action	0.00186555
romance	0.00187958
romance	0.00196601
romance	0.00196691
action	0.00203391
action	0.00205723

... (195 rows omitted)

In [357]:
```
ok.grade("q3_1_3");
```
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 4
    Failed: 0
[ooooooooook] 100.0% passed
```

Question 3.1.4

Now compute the 5-nearest neighbors classification of the first movie in the test set. That is, decide on its genre by finding the most common genre among its 5 nearest neighbors in the training set, according to the distances you've calculated. Then check whether your classifier chose the right genre. (Depending on the features you chose, your classifier might not get this movie right, and that's okay.)

In [358]:
```
# Set my_assigned_genre to the most common genre among these.
my_assigned_genre = genre_and_distances.select('Genre').take(np.arange(5)).group('Genre').sort('cou
nt', descending = True).column('Genre').item(0)
# Set my_assigned_genre_was_correct to True if my_assigned_genre
# matches the actual genre of the first movie in the test set.
my_assigned_genre_was_correct = False

print("The assigned genre, {}, was{}correct.".format(my_assigned_genre, " " if my_assigned_genre_wa
s_correct else " not "))
```
```
The assigned genre, romance, was not correct.
```

In [359]:
```
ok.grade("q3_1_4");
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 2
    Failed: 0
[ooooooooook] 100.0% passed
```

3.2. A classifier function

Now we can write a single function that encapsulates the whole process of classification.

Question 3.2.1

Write a function called `classify`. It should take the following four arguments:

- A row of features for a movie to classify (e.g., `test_20.row(0)`).
- A table with a column for each feature (e.g., `train_20`).
- An array of classes (e.g. the labels "romance" or "action") that has as many items as the previous table has rows, and in the same order.
- `k`, the number of neighbors to use in classification.

It should return the class a `k`-nearest neighbor classifier picks for the given row of features (the string `'romance'` or the string `'action'`).

In [360]:
```
def classify(test_row, train_rows, train_labels, k):
    """Return the most common class among k nearest neigbors to test_row."""
    distances = fast_distances(test_row, train_rows)
    genre_and_distances = Table().with_column('Genre', train_labels).with_column('Distance', distan
ces).sort('Distance').take(np.arange(k))
    classified = genre_and_distances.select('Genre').group('Genre').sort('count', descending = True
).column('Genre').item(0)
    return classified
```

In [361]:
```
ok.grade("q3_2_1");
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 22
    Failed: 0
[ooooooooook] 100.0% passed
```

Question 3.2.2

Assign `king_kong_genre` to the genre predicted by your classifier for the movie "king kong" in the test set, using **11 neighbors** and using your 20 features.

In [362]:
```
# The staff solution first defined a row called king_kong_features.
king_kong_features = test_movies.where('Title', 'king kong').select(my_20_features).row(0)
king_kong_genre = classify(king_kong_features, train_20, train_movies.column('Genre'), 11)
king_kong_genre
```

Out[362]: 'action'

In [363]:
```
ok.grade("q3_2_2");
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[ooooooooook] 100.0% passed
```

Finally, when we evaluate our classifier, it will be useful to have a classification function that is specialized to use a fixed training set and a fixed value of `k`.

Question 3.2.3

Create a classification function that takes as its argument a row containing your 20 features and classifies that row using the 11-nearest neighbors algorithm with `train_20` as its training set.

In [364]:
```
def classify_feature_row(row):
    return classify(row, train_20, train_movies.column('Genre'), 11)

# When you're done, this should produce 'Romance' or 'Action'.
classify_feature_row(test_20.row(0))
```

Out[364]: 'romance'

In [365]:
```
ok.grade("q3_2_3");
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 11
    Failed: 0
[ooooooooook] 100.0% passed
```

3.3. Evaluating your classifier

Now that it's easy to use the classifier, let's see how accurate it is on the whole test set.

Question 3.3.1. Use `classify_feature_row` and `apply` to classify every movie in the test set. Assign these guesses as an array to `test_guesses`. **Then**, compute the proportion of correct classifications.

In [366]:
```
test_guesses = test_20.apply(classify_feature_row)
proportion_correct = sum(test_guesses == test_movies.column('Genre'))/len(test_guesses)
proportion_correct
```

Out[366]: 0.7297297297297297

In [367]:
```
ok.grade("q3_3_1");
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[ooooooooook] 100.0% passed
```

Question 3.3.2. An important part of evaluating your classifiers is figuring out where they make mistakes. Assign the name `test_movie_correctness` to a table with three columns, `'Title'`, `'Genre'`, and `'Was correct'`. The last column should contain `True` or `False` depending on whether or not the movie was classified correctly.

In [368]:
```
# Feel free to use multiple lines of code
# but make sure to assign test_movie_correctness to the proper table!
test_movie_correctness = test_movies.select('Title', 'Genre').with_column('Was correct', test_guess
es == test_movies.column('Genre'))
test_movie_correctness.sort('Was correct', descending = True).show()
```

Title	Genre	Was correct
american outlaws	action	True
solaris	romance	True
logan's run	action	True
48 hrs.	action	True
independence day	action	True
my girl 2	romance	True
the bourne identity	action	True
the apartment	romance	True
rear window	romance	True
an officer and a gentleman	romance	True
blade	action	True
superman iv: the quest for peace	action	True
dune	action	True
the majestic	romance	True
crime spree	action	True
the fantastic four	action	True
mr. deeds goes to town	romance	True
blade ii	action	True
cruel intentions	romance	True
star wars	action	True
starship troopers	action	True
the hudsucker proxy	romance	True
king kong	action	True
sleepless in seattle	romance	True
the bourne supremacy	action	True
witness	romance	True
x-men	action	True
batman returns	action	False
legend	romance	False
the negotiator	action	False
the cider house rules	romance	False
top gun	romance	False
conspiracy theory	action	False
badlands	romance	False
body of evidence	romance	False
men in black	action	False
the mummy	action	False

In [369]:
```
ok.grade("q3_3_2");
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 3
    Failed: 0
[ooooooooook] 100.0% passed
```

Question 3.3.3. Do you see a pattern in the types of movies your classifier misclassifies? In two sentences or less, describe any patterns you see in the results or any other interesting findings from the table above. If you need some help, try looking up the movies that your classifier got wrong on Wikipedia.

The movies above that were classified wrong were mostly crime/mystery related movies, and many of them have romantic relationships as part of their plot.

At this point, you've gone through one cycle of classifier design. Let's summarize the steps:

1. From available data, select test and training sets.
2. Choose an algorithm you're going to use for classification.
3. Identify some features.
4. Define a classifier function using your features and the training set.
5. Evaluate its performance (the proportion of correct classifications) on the test set.

4. Explorations

Now that you know how to evaluate a classifier, it's time to build a better one.

Question 4.1

Develop a classifier with better test-set accuracy than `classify_feature_row`. Your new function should have the same arguments as `classify_feature_row` and return a classification. Name it `another_classifier`. Then, check your accuracy using code from earlier.

You can use more or different features, or you can try different values of `k`. (Of course, you still have to use `train_movies` as your training set!)

Make sure you don't reassign any previously used variables here, such as `proportion_correct` from the previous question.

In [370]:
```
# To start you off, here's a list of possibly-useful features
# Feel free to add or change this array to improve your classifier
new_features = make_array("come", "do", "have", "heart", "make", "never", "now", "wanna", "with",
"yo")

train_new = train_movies.select(new_features)
test_new = test_movies.select(new_features)

def another_classifier(row):
    return classify(row, train_20, train_movies.column('Genre'), 12)
test_guesses_1 = test_20.apply(another_classifier)
test_correctness = test_movies.select('Title','Genre').with_column('Was Correct', test_movies.colum
n('Genre')==test_guesses_1)
test_correctness.group('Was Correct')
```

Out[370]:

Was Correct	count
False	9
True	28

Question 4.2

Do you see a pattern in the mistakes your new classifier makes? What about in the improvement from your first classifier to the second one? Describe in two sentences or less.

Hint: You may not be able to see a pattern.

The classifier improved by increasing the k number of nearest neighbors.

Question 4.3

Briefly describe what you tried to improve your classifier.

I tried putting in larger and smaller values of k until I found the one that resulted in the least number of False predictions

Congratulations: you're done with the required portion of the project! Time to submit.

In [371]:
```
_ = ok.submit()
```

```
Saving notebook... Saved 'project3.ipynb'.
Submit... 100% complete
Submission successful for user: epere@berkeley.edu
URL: https://okpy.org/cal/data8/fa19/project3/submissions/k2Womx
```

5. Other Classification Methods (OPTIONAL)

Note: Everything below is **OPTIONAL**. Please only work on this part after you have finished and submitted the project. If you create new cells below, do NOT reassign variables defined in previous parts of the project.

Now that you've finished your k-NN classifier, you might be wondering what else you could do to improve your accuracy on the test set. Classification is one of many machine learning tasks, and there are plenty of other classification algorithms! If you feel so inclined, we encourage you to try any methods you feel might help improve your classifier.

We've compiled a list of blog posts with some more information about classification and machine learning. Create as many cells as you'd like below--you can use them to import new modules or implement new algorithms.

Blog posts:

- Classification algorithms/methods
- Train/test split and cross-validation
- More information about k-nearest neighbors
- Overfitting

In future data science classes, such as Data Science 100, you'll learn about some about some of the algorithms in the blog posts above, including logistic regression. You'll also learn more about overfitting, cross-validation, and approaches to different kinds of machine learning problems.

There's a lot to think about, so we encourage you to find more information on your own!

Modules to think about using:

- Scikit-learn tutorial
- TensorFlow information

...and many more!

In [ ]:
```
...
```