In [28]:

```
# 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('movies.ok')
```

```
=====================================================================
Assignment: Movie Classification with k-NN
OK, version v1.14.20
=====================================================================
```

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 [29]:

```
movies = Table.read_table('movies.csv')
movies.where("Title", "wild wild west").select(0, 1, 2, 3, 4, 14, 49,
1042, 4004)
```

Out[29]:

Title	Year	Rating	Genre	# Words	breez	england	it	bravo
wild wild west	1999	4.3	comedy	3446	0	0	0.0212635	0

The above cell prints a few columns of the row for the comedy movie *Wild Wild West*. The movie contains 3446 words. The word "it" appears 74 times, as it makes up $\frac{74}{3446} \approx 0.021364$ of the words in the movie. The word "england" 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.

Note: All movies in our dataset have their titles lower-cased.

In [30]:

```
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[30]: Row(Title='the terminator', Year='1984', Rating=8.1, Genre='thriller',
         # Words=2210, she=0.0024084778420038703, decid=0.000963391136801548, t
         alk=0.001926782273603096, wit=0.0, razor=0.0, slam=0.0, credit=0.0, ra
         i=0.0, hugh=0.0, breez=0.0, conscienc=0.0, audienc=0.0, cathi=0.0, log
         =0.0, met=0.0, chosen=0.0, grip=0.0, booz=0.0, bianca=0.0, doubl=0.0,
         agent=0.0, exit=0.0, carpent=0.0, underground=0.0, clemenza=0.0, gain=
         0.0, neg=0.0, majesti=0.0, studio=0.0, chri=0.0, spin=0.0, greater=0.
         0, eaten=0.0, vibrat=0.0, stupid=0.000481695568400774, cigarett=0.0004
         81695568400774, jesu=0.0, mani=0.0, violin=0.0, financi=0.0, bai=0.0,
         cop=0.000481695568400774, neighbor=0.0, cd=0.0, england=0.0, made=0.00
         0481695568400774, conni=0.0, instinct=0.0, took=0.0, jacquelin=0.0, ma
         ce=0.0, disappear=0.000481695568400774, waltz=0.0, behind=0.0, bourbon
         =0.0, favorit=0.0, benni=0.0, manhattan=0.0, nixon=0.0, lunch=0.0, pri
         ncipl=0.0, tradit=0.0, counterfeit=0.0, sophi=0.0, third=0.0, exist=0.
         000963391136801548, wouldv=0.0, hero=0.0, theyr=0.000481695568400774,
         anytim=0.0, christin=0.0, vallei=0.000481695568400774, chess=0.0, paid
         =0.0, burglar=0.0, nostril=0.0, rubber=0.000481695568400774, human=0.0
         00963391136801548, british=0.0, plissken=0.0, eddi=0.0, gee=0.0, offen
         d=0.0, rebecca=0.0, anger=0.0, plant=0.0, famou=0.0, repres=0.0, lates
         t=0.0, rent=0.0, dip=0.0, bell=0.0, andi=0.0, so=0.005780346820809285
         6, london=0.0, cooler=0.0, keaton=0.0, portland=0.0, headlin=0.0, what
         ta=0.0, fatal=0.0, sew=0.0, cheer=0.0, davi=0.0, feed=0.0, hudson=0.0,
         ambros=0.0, digest=0.0, redi=0.0, fri=0.0, staff=0.0, casino=0.0, occa
         sion=0.0, shadow=0.0, work=0.000963391136801548, restrain=0.0, face=0.
         000481695568400774, exercis=0.0, sidnei=0.0, pile=0.0, whyd=0.0, teena
         g=0.0, her=0.0033718689788805418, retir=0.0, hazard=0.0, roth=0.0, hurr
         ican=0.0, impuls=0.0, ranger=0.0, pour=0.0, lester=0.0, slash=0.0, dee
         r=0.0, could=0.0014450867052023214, vital=0.0, qualiti=0.0, coma=0.0,
         incred=0.0, hank=0.0, famili=0.0, duchess=0.0, global=0.0, virgin=0.0,
         scientif=0.0, between=0.0, holidai=0.0, qualifi=0.0, moor=0.0, happili
         =0.0, arizona=0.0, non=0.0, bruce=0.0, ankl=0.0, constant=0.0, buzz=0.
         0, harder=0.0, ing=0.0014450867052023214, christian=0.0, palmer=0.0, t
         ent=0.0, sunset=0.0, damour=0.0, cohaagen=0.0, advertis=0.0, sensat=0.
         0, local=0.0, there=0.007225433526011609, terri=0.0, sedat=0.0, rotten
         =0.0, struck=0.0, deck=0.0, past=0.0, bro=0.0, ann=0.0, dump=0.0, kane
         =0.0, slot=0.0, immun=0.0, block=0.0, lil=0.0, technic=0.0, tactic=0.0
         00481695568400774, pencil=0.0, outsid=0.000963391136801548, laboratori
         =0.0, easi=0.000481695568400774, nephew=0.0, coffin=0.0, pretti=0.0004
         81695568400774, coward=0.0, verbal=0.0, permiss=0.0, bartend=0.0, wont
         =0.0014450867052023214, watch=0.0, lindenmey=0.0, cosmo=0.0, capabl=0.
         0, flirt=0.0, huge=0.0, berkelei=0.0, max=0.0, walter=0.0, lime=0.0, r
         ico=0.0, marvin=0.0, aboard=0.0, bacon=0.0, account=0.0, kirk=0.0, qua
         id=0.0, stunt=0.0, closet=0.0, due=0.0, nuclear=0.0, blind=0.0, pussi=
         0.0, howdi=0.0, snuff=0.0, eas=0.0, now=0.002890173410404644, leak=0.
         0, underwear=0.0, westlei=0.0, mayb=0.0014450867052023214, theo=0.0, l
         imo=0.0, cousin=0.0, illeg=0.0, silli=0.0, against=0.0, done=0.0009633
         91136801548, district=0.0, invad=0.0, ryan=0.0, wait=0.000963391136801
         548, grudg=0.0, charact=0.0, hick=0.0, jami=0.0, lifetim=0.0, lecktor=
         0.0, and=0.010115606936416251, republican=0.0, life=0.0009633911368015
         48, hidden=0.0, wire=0.0, paranoia=0.0, network=0.000481695568400774,
         messi=0.0, uthatu=0.0, effort=0.0, carri=0.0, windham=0.0, fun=0.00048
         1695568400774, psychologist=0.0, sean=0.0, scent=0.0, answer=0.0009633
         91136801548, mom=0.000481695568400774, wake=0.0, sign=0.0, ho=0.0, rel
         at=0.0, jame=0.0, fat=0.000481695568400774, myself=0.0, disrupt=0.0, s
         can=0.0, vagu=0.0, basket=0.0, christma=0.0, estim=0.0, em=0.0, union=
         0.0, involv=0.0, norman=0.0, suspici=0.0, becom=0.000481695568400774,
         shoe=0.0, librari=0.0, administr=0.0, ford=0.0, complic=0.0, stuck=0.
```

```
0, justic=0.0, attack=0.0, releas=0.000481695568400774, econom=0.0, he
sit=0.0, autopsi=0.0, jurisdict=0.0, four=0.000481695568400774, factor
=0.0, inquiri=0.0, lion=0.0, meanwhil=0.0, prison=0.0, blair=0.0, seri
=0.000963391136801548, groceri=0.0, surgeri=0.000481695568400774, seas
on=0.0, christi=0.0, clean=0.000481695568400774, ow=0.0004816955684007
74, wrestl=0.0, en=0.0, moral=0.0, hungri=0.0, cole=0.0, surfer=0.0, s
ixteen=0.0, angl=0.0, shame=0.0, barrel=0.0, major=0.0, ago=0.0, lott=
0.0, airplan=0.0, worth=0.000481695568400774, train=0.0, easili=0.0, f
eller=0.0, valentin=0.0, harvei=0.0, wherev=0.000481695568400774, fran
cisco=0.0, true=0.0, dramat=0.0, boston=0.0, besid=0.0, inspector=0.0,
orlean=0.0, opportun=0.0, nearli=0.0, lindsei=0.0, photograph=0.0, fra
me=0.0, at=0.002890173410404644, psychopath=0.0, press=0.0009633911368
01548, youyou=0.0, havana=0.0, australia=0.0, plai=0.0, mayfield=0.0,
chick=0.0, stewart=0.0, seven=0.000481695568400774, reflect=0.0, outer
=0.0, vega=0.0, anywai=0.000481695568400774, prime=0.0, farmer=0.0, ba
ckyard=0.0, joe=0.0, otherwis=0.0, cowgirl=0.0, grate=0.0, clerk=0.0,
dispos=0.000481695568400774, tow=0.0, mari=0.0, certifi=0.0, thi=0.009
152215799614708, wheel=0.0, privaci=0.0, todai=0.0, nathan=0.0, teller
=0.0, plot=0.0, correct=0.0, couch=0.0, job=0.000481695568400774, hurt
=0.000481695568400774, inject=0.0, chocol=0.0, session=0.0, outrag=0.
0, reduc=0.0, knew=0.000963391136801548, jd=0.0, perfum=0.0, fabric=0.
0, bodyguard=0.0, think=0.003371868978805418, il=0.0, yesterdai=0.0, s
ide=0.000481695568400774, doesnt=0.000481695568400774, ronni=0.0, blan
k=0.0, jess=0.0, push=0.0, ahh=0.0, jealou=0.0, alter=0.0, blew=0.0, b
u=0.0, off=0.0, sweetheart=0.0, abl=0.0, angelo=0.0, nicer=0.0, coupla
=0.0, resum=0.0, coke=0.0, strangl=0.0, gut=0.0, morn=0.0, miracl=0.0,
bit=0.000481695568400774, intimid=0.0, pipelin=0.0, sour=0.0, shep=0.
0, vivian=0.0, grave=0.0, chemic=0.0, czech=0.0, scholarship=0.0, oldf
ashion=0.0, accent=0.0, spitz=0.0, dirti=0.0, shot=0.00096339113680154
8, lit=0.0, cedar=0.0, pirat=0.0, weather=0.0, stun=0.0, learn=0.00096
3391136801548, wick=0.0, bring=0.000481695568400774, slack=0.0, brave=
0.0, shakespear=0.0, monkei=0.0, presum=0.0, vacat=0.0, faint=0.0, str
ap=0.0, stephen=0.0, maggi=0.0, indic=0.0, sundai=0.0, nois=0.0, organ
=0.000963391136801548, terranc=0.0, foundat=0.0, littl=0.0, perman=0.
0, insid=0.000481695568400774, stabl=0.000481695568400774, sharp=0.0,
uptight=0.0, wholl=0.0, jeffrei=0.0, root=0.0, thy=0.0, josi=0.0, woma
n=0.0, post=0.0, judg=0.0, ralph=0.0, amaz=0.0, surf=0.0, naughti=0.0,
norm=0.0, glove=0.0, cigar=0.0, wendi=0.0, corpor=0.00048169556840077
4, statement=0.000481695568400774, defin=0.0, drawn=0.0, progress=0.0,
year.1=0.0014450867052023214, shovel=0.0, sequenc=0.0, andand=0.0, ree
l=0.0, held=0.0, youv=0.000963391136801548, trick=0.0, horseman=0.0, w
hoa=0.0, emploi=0.0, chain=0.0, cmon=0.0, brief=0.0, creativ=0.0, mosc
ow=0.0, challeng=0.0, walli=0.0, golf=0.0, abort=0.000481695568400774,
aha=0.000481695568400774, bent=0.0, exclus=0.0, amber=0.0, figur=0.0,
healthi=0.0, ransom=0.0, steer=0.0, blow=0.000481695568400774, bark=0.
0, imbecil=0.0, mother=0.0014450867052023214, had=0.004335260115606965
5, whatev=0.000481695568400774, donit=0.0, goodby=0.0, terrifi=0.0, ca
sh=0.000481695568400774, descript=0.0, spell=0.0, west=0.0, shoulda=0.
0, when=0.003371868978805418, wear=0.000481695568400774, crop=0.0, tra
pper=0.0, donat=0.0, breath=0.000481695568400774, bracelet=0.0, lover=
0.0, afraid=0.000481695568400774, vice=0.0, ms=0.000481695568400774, d
idnt=0.0024084778420038703, kill=0.0014450867052023214, depth=0.0, si=
0.0, experiment=0.0, dna=0.0, desmond=0.0, coconut=0.0, dil=0.0, llewe
lyn=0.0, spoil=0.0, lung=0.0, attent=0.000481695568400774, offens=0.0,
babi=0.0, havin=0.0, mustnt=0.0, creepi=0.0, daniel=0.0, abandon=0.0,
less=0.000481695568400774, go=0.009152215799614708, negoti=0.0, butche
r=0.0, sudden=0.0, templ=0.0, ii=0.0, alex=0.0, deal=0.0, rememb=0.0,
polic=0.0, gasolin=0.0, luggag=0.0, smooth=0.0, declar=0.0, chase=0.0,
host=0.0, uptown=0.0, heavi=0.0, tenni=0.0, picard=0.0, success=0.0, a
s=0.001926782273603096, heroin=0.0, hi=0.000963391136801548, seduc=0.
0, den=0.0, accur=0.0, parasit=0.0, fiddl=0.0, altern=0.0, chees=0.0,
flatter=0.0, lloyd=0.0, collector=0.0, athlet=0.0, useless=0.0, yeh=0.
0, lawsuit=0.0, guitar=0.0, apart=0.000481695568400774, strong=0.0, di
tch=0.0, doc=0.0, wors=0.0, trigger=0.0, mister=0.0, request=0.0, dire
ct=0.0, telephon=0.0, expir=0.0, blond=0.0, energi=0.0, eh=0.0, damag=
0.0, across=0.0, anni=0.0, eleven=0.000481695568400774, episod=0.0, re
liant=0.0, jason=0.0, get=0.007225433526011609, duh=0.0, victori=0.0,
speci=0.0, includ=0.0, time=0.001926782273603096, imagin=0.00048169556
8400774, hang=0.0, estat=0.0, full=0.0, sleep=0.0014450867052023214, r
eceiv=0.0, grei=0.000481695568400774, belief=0.0, itali=0.0, separ=0.
0, claw=0.0, circuit=0.0, memo=0.0, monei=0.000481695568400774, anna=
0.0, term=0.0, drunk=0.0, everydai=0.0, haldeman=0.0, frozen=0.0, cast
ro=0.0, soup=0.0, mornin=0.0, starship=0.0, dawn=0.0, curtain=0.0, pip
e=0.0, twist=0.0, downtown=0.0, drive=0.0, tank=0.0, servant=0.0, circ
umst=0.0, town=0.0, beat=0.0, wooden=0.0, thursdai=0.0, ordinari=0.0,
acid=0.0, must=0.002890173410404644, literatur=0.0, carol=0.0, homewor
k=0.0, present=0.0, lunat=0.0, lifestyl=0.0, rat=0.0, satisfi=0.0, fun
ni=0.0, gang=0.0, jewelri=0.0, develop=0.000481695568400774, yell=0.0,
stash=0.0, ab=0.0, mm=0.0, thumb=0.0, to=0.02312138728323715, lift=0.
0, re=0.0, mud=0.0, taranski=0.0, camel=0.0, cancer=0.0, valiant=0.0,
leavin=0.0, sand=0.0, themselv=0.0, stai=0.000481695568400774, junki=
0.0, dinner=0.0, subject=0.0, ax=0.0, thankyou=0.0, squad=0.0, charl=
0.0, pretend=0.0, mumford=0.0, player=0.0, sorri=0.000481695568400774,
feet=0.000481695568400774, dee=0.0, casual=0.0, check=0.00144508670520
23214, fugit=0.0, garden=0.0, million=0.0, suppos=0.001445086705202321
4, observ=0.0, low=0.000481695568400774, cathol=0.0, parad=0.0, confus
=0.0, appli=0.0, uhura=0.0, sane=0.0, beverli=0.0, princip=0.0, helen=
0.0, drain=0.0, want=0.003853564547206192, maam=0.0, care=0.0, tomorro
w=0.0, abus=0.0, parent=0.0, diner=0.0, proud=0.0, squeez=0.0, allerg=
0.0, displai=0.0, rave=0.0, da=0.0, tabl=0.0, kip=0.0, string=0.0, ik=
0.0, microwav=0.0, tongu=0.0, miner=0.0, montana=0.0, ident=0.0, gasto
n=0.0, individu=0.0, treasur=0.0, brown=0.0, delic=0.0, forbidden=0.0,
blah=0.0, although=0.0, cold=0.000963391136801548, strategi=0.0, psych
olog=0.0, fill=0.0, profession=0.0, rig=0.0, theori=0.0, psychiatr=0.
0, raw=0.0, critic=0.0, contribut=0.0, fart=0.0, bitter=0.0, lemm=0.0,
isol=0.0, drug=0.0, also=0.0, type=0.0, oblig=0.0, jen=0.0, feder=0.0,
note=0.0, perhap=0.0, dwayn=0.0, wolf=0.0, central=0.0, annoi=0.0, cyc
l=0.0, accid=0.0, puzzl=0.0, invis=0.000481695568400774, ronald=0.0, m
ercuri=0.0, escap=0.0, damon=0.0, acr=0.0, spy=0.0, gig=0.0, armor=0.
0, gotten=0.0, swana=0.0, scene=0.0, marla=0.0, penetr=0.0, shock=0.0,
sunk=0.0, iv=0.000481695568400774, diari=0.0, atlant=0.0, absenc=0.0,
corps=0.0, relai=0.0, rip=0.000481695568400774, bull=0.0, requir=0.0,
buck=0.0, complain=0.0, russia=0.0, arctic=0.0, schedul=0.000481695568
400774, casei=0.0, temperatur=0.0, tree=0.0, unbeliev=0.0, graduat=0.
0, place=0.000963391136801548, communist=0.0, nine=0.0, vulner=0.0, hi
ke=0.0, raymond=0.0, laura=0.0, manifest=0.0, cellular=0.0, repress=0.
0, divid=0.0, lesson=0.0, crate=0.0, coin=0.0, bachelor=0.0, carl=0.0,
sport=0.0, simpli=0.0, politician=0.0, destroi=0.0, juno=0.0, castor=
0.0, liter=0.0, dwight=0.0, malkovich=0.0, cord=0.0, edmund=0.0, walk=
0.000481695568400774, cake=0.0, protect=0.0, nearest=0.0, takin=0.0, c
omrad=0.0, dai=0.0, tour=0.0, whose=0.0, partial=0.0, vada=0.0, deligh
t=0.0, godfath=0.0, cheat=0.0, harmless=0.0, romant=0.0, pound=0.0, im
=0.007707129094412382, violat=0.0, chief=0.0, onlin=0.0, pleasur=0.0,
crystal=0.0, luther=0.0, festiv=0.0, along=0.0, admir=0.0, owner=0.0,
prove=0.0, giron=0.0, lydia=0.0, dad=0.0, ultim=0.0, hawkin=0.0, hei=
0.000481695568400774, father=0.000481695568400774, rot=0.0, doll=0.0,
fountain=0.0, williamson=0.0, horror=0.0, ell=0.0, media=0.0, russian=
0.0, carefulli=0.000481695568400774, verifi=0.0, prepar=0.000481695568
400774, cia=0.0, flew=0.0, logic=0.0, affect=0.0, skirt=0.0, nineteen=
0.0, breakdown=0.0, william=0.0, batteri=0.0, shore=0.0, project=0.0,
strength=0.000481695568400774, bail=0.0, piti=0.000481695568400774, ha
rbor=0.0, oper=0.0, yanke=0.0, orphan=0.0, squar=0.0, gari=0.0, driven
=0.0, nanci=0.000481695568400774, leonard=0.0, nonsens=0.0, anybodi=0.
000481695568400774, strict=0.0, riddl=0.0, boulevard=0.0, articl=0.0,
shop=0.0, mexico=0.0, ruin=0.000481695568400774, did=0.002890173410404
644, annett=0.0, piano=0.0, chest=0.0, becker=0.0, strip=0.0, stroll=
0.0, philosophi=0.0, footbal=0.0, whom=0.000481695568400774, flight=0.
0, refriger=0.0, loretta=0.0, geniu=0.0, condit=0.0, for=0.00915221579
9614708, anyon=0.0, ideal=0.000481695568400774, terribl=0.0, otho=0.0,
self=0.0, swore=0.0, ring=0.0, baron=0.0, civilian=0.0, panick=0.0, se
ttl=0.0, spread=0.0, turn=0.0, blast=0.0, growth=0.0, seal=0.0, box=0.
0, locker=0.0, help=0.003371868978805418, cast=0.0, jennif=0.0, regan=
0.0, stole=0.0, yessir=0.0, randi=0.0, left=0.0, explain=0.0, inspect=
0.0, tribun=0.0, pop=0.0, bed=0.0, encourag=0.0, bubbl=0.0, contact=0.
0, patrick=0.0, domino=0.0, reliev=0.0, superman=0.0, sector=0.0, entr
anc=0.0, corner=0.0, mama=0.0, supposedli=0.0, yo=0.0, role=0.0, wai=
0.000963391136801548, vicki=0.0, buddi=0.000481695568400774, wizard=0.
0, heal=0.0, element=0.0, here=0.007225433526011609, leather=0.0, hard
li=0.0, their=0.0, sworn=0.0, africa=0.0, yearold=0.0, sphere=0.0, psy
ch=0.0, behalf=0.0, moren=0.0, shelter=0.0, dr=0.000481695568400774, w
eight=0.0, ami=0.0, pike=0.0, gitt=0.0, kinda=0.000481695568400774, li
ber=0.0, badg=0.0, last=0.000481695568400774, from=0.00337186897880541
8, uhhuh=0.0, afford=0.0, shag=0.0, presenc=0.0, jai=0.0, hall=0.0, wo
rst=0.0, agenc=0.0, threat=0.000481695568400774, jaw=0.0, hold=0.00096
3391136801548, someplac=0.0, birth=0.0, junk=0.0, natali=0.0, graveyar
d=0.0, march=0.0, focu=0.0, graviti=0.0, smoke=0.0, normal=0.0, congre
ss=0.0, ash=0.0, predict=0.0, copper=0.0, court=0.0, wealth=0.0, it=0.
031310211946050305, um=0.0, issu=0.0, misunderstand=0.0, berlin=0.0, s
tructur=0.0, hardwar=0.0, casanova=0.0, worship=0.0, unhappi=0.0, spit
=0.0, quadrant=0.0, downstair=0.0, see=0.004335260115606965 5, gentleme
n=0.0, intern=0.0, conrad=0.0, flame=0.0, within=0.0, drum=0.0, yet=0.
0014450867052023214, differ=0.001926782273603096, jacket=0.0, onto=0.
0, prospect=0.0, nation=0.0, soft=0.0, girl=0.0014450867052023214, cou
nter=0.0, rug=0.0, neutral=0.0, elli=0.0, transmiss=0.0, integr=0.0, c
adillac=0.0, hose=0.0, lillian=0.0, clip=0.0, stanlei=0.0, marriag=0.
0, built=0.000481695568400774, paso=0.0, subwai=0.0, whiskei=0.0, kyle
=0.003853564547206192, sai=0.0014450867052023214, barzini=0.0, virgini
a=0.0, lou=0.0, slip=0.000481695568400774, pageant=0.0, tellin=0.0, ta
ylor=0.0, proof=0.0, yank=0.0, yacht=0.0, bless=0.0, brooklyn=0.0, arg
u=0.0, pistol=0.0, bo=0.0, unnecessari=0.0, share=0.0, th=0.0, sabotag
=0.0, china=0.0, hojon=0.0, intim=0.0, chop=0.0, comic=0.0, compuls=0.
0, bingo=0.0, make=0.004335260115606965 5, vulcan=0.0, parri=0.0, shoul
dnt=0.000481695568400774, disconnect=0.000481695568400774, toon=0.0, f
orc=0.0, annabel=0.0, clown=0.0, sentenc=0.0, with=0.00240847784200387
03, cartoon=0.0, suspicion=0.0, willi=0.0, brook=0.0, revers=0.0, luci
=0.0, visitor=0.0, wave=0.0, stick=0.0, cuban=0.0, sweep=0.0, comment=
0.0, elimin=0.0, spencer=0.0, monica=0.0, debat=0.0, coron=0.0, lie=0.
0, ourselv=0.000963391136801548, weapon=0.000963391136801548, kate=0.
0, distant=0.0, grade=0.0, atf=0.0, balloon=0.0, southern=0.0, chang=
0.0, oppos=0.0, indict=0.0, respect=0.0, sure=0.0014450867052023214, j
ane=0.0, doubt=0.000481695568400774, princ=0.0, admiss=0.0, ador=0.0,
jungl=0.0, paint=0.0, south=0.0, event=0.0, taxi=0.0, voic=0.0, video=
0.0, convert=0.0, couldnt=0.0, citi=0.000481695568400774, motiv=0.0, r
el=0.0, herself=0.0, curios=0.0, buffalo=0.0, terrorist=0.0, write=0.
0, school=0.0, wing=0.0, smack=0.0, marti=0.0, counti=0.0, pud=0.0, th
atd=0.0, readi=0.0, manrai=0.0, airlin=0.0, romeo=0.0, weekend=0.0, ap
olog=0.0, ladi=0.0, emili=0.0, maker=0.0, dian=0.0, barf=0.0, quest=0.
0, crown=0.0, expect=0.0, gull=0.0, pump=0.0, depress=0.0, interrog=0.
0, instant=0.0, barri=0.0, book=0.000481695568400774, ahm=0.0, shutup=
0.0, larri=0.0, seriou=0.0, hobbi=0.0, unotu=0.0, kingdom=0.0, on=0.01
3005780346820898, quarter=0.0, nicki=0.0, pierr=0.0, dream=0.0, claren
c=0.0, buffi=0.0, winner=0.0, scatter=0.0, erik=0.0, bike=0.0, bean=0.
0, unabl=0.0, women=0.000963391136801548, maya=0.0, tommi=0.0, seventy
f=0.0, curs=0.0, maintain=0.0, genesi=0.0, background=0.0, dude=0.0, s
atellit=0.0, hour=0.0014450867052023214, extend=0.0, transport=0.0, hu
ll=0.0, top=0.0, grief=0.0, evid=0.0, gradi=0.0, onc=0.000481695568400
774, philadelphia=0.0, honesti=0.0, super=0.0, florida=0.0, gentleman=
0.0, cannot=0.0, too=0.0014450867052023214, fuss=0.0, hasnt=0.0, looki
n=0.0, arrang=0.0, brill=0.0, sale=0.0, month=0.0, intact=0.0, foolish
=0.0, newspap=0.0, transit=0.0, roi=0.0, whew=0.0, uwhatu=0.0, main=0.
0, bat=0.0, line=0.000481695568400774, snake=0.0, bald=0.0, cage=0.0,
toward=0.0, sold=0.0, till=0.0, mood=0.0, warn=0.0, johnni=0.0, mustv=
0.0, scream=0.0, undress=0.0, absolut=0.000481695568400774, flaw=0.0,
bake=0.0, mmm=0.000481695568400774, jerri=0.0, employe=0.0, quot=0.0,
waitin=0.0, benefit=0.0, werent=0.0, fulfil=0.0, plastic=0.0, tribe=0.
0, yknow=0.000481695568400774, hope=0.0, layer=0.0, tom=0.0, reaction=
0.0, san=0.0, katrina=0.0, identifi=0.0, red=0.0, exorc=0.0, iron=0.0,
beer=0.0, santo=0.0, bruis=0.0, stab=0.0, louis=0.0, agreement=0.0, co
ulda=0.0, film=0.0, rufu=0.0, soil=0.0, playin=0.0, millionair=0.0, mi
ddl=0.0, closer=0.0, spirit=0.0, accident=0.0, yer=0.0, traffic=0.0, w
hat=0.00963391136801548, convict=0.0, sack=0.0, examin=0.0, compound=
0.0, squid=0.0, gimm=0.0, fault=0.0, mysteri=0.0, sea=0.0, barbara=0.
0, surpris=0.0, rm=0.0, hid=0.0, sewer=0.0, kilomet=0.0, lisa=0.0, wor
kshop=0.0, safeti=0.0, touch=0.000963391136801548, jersei=0.0, frequen
t=0.0, enjoi=0.0, loos=0.0, homeless=0.0, ah=0.0, extens=0.0, termin=
0.0014450867052023214, level=0.0, violent=0.0, rush=0.0, coordin=0.0,
earli=0.0, wa=0.007225433526011609, dizzi=0.0, privat=0.0, gwen=0.0, s
uicid=0.0, headquart=0.0, educ=0.0, sort=0.000481695568400774, handsom
=0.0, stella=0.0, ac=0.0, audit=0.0, antiqu=0.0, dot=0.0, technolog=0.
0, motion=0.0, hairi=0.0, site=0.0, student=0.0, up=0.0048169556840077
41, either=0.0, pry=0.0, conspiraci=0.0, basi=0.0, timer=0.0, heat=0.
0, lawson=0.0, hear=0.000481695568400774, ui=0.0, fool=0.0, mere=0.0,
huh=0.000481695568400774, donni=0.0, none=0.000963391136801548, fifth=
0.0, becaus=0.0, lamb=0.0, interpret=0.0, increas=0.0, tower=0.0, mick
ei=0.0, colleagu=0.0, confer=0.0, hollow=0.0, leon=0.0, thou=0.0, fran
=0.0, percent=0.0, six=0.000481695568400774, limp=0.0, arrowai=0.0, ex
plod=0.0, friendli=0.0, breakfast=0.0, greek=0.0, need=0.0004816955684
00774, rome=0.0, beast=0.0, rehab=0.0, ben=0.0, land=0.0, river=0.0, f
rank=0.0, smash=0.000481695568400774, quicker=0.0, former=0.0, lower=
0.0, rap=0.0, nick=0.0, came=0.0, hopeless=0.0, comedian=0.0, right=0.
001926782273603096, yall=0.0, truli=0.0, not=0.008188824662813158, soz
e=0.0, forgotten=0.0, tight=0.000481695568400774, pro=0.0, sona=0.0, u
nder=0.000481695568400774, precis=0.0, center=0.0, stiff=0.0, virtual=
0.0, author=0.000481695568400774, dry=0.0, theyll=0.0, golden=0.0, fin
al=0.0, properli=0.0, paper=0.0, older=0.0, serv=0.0, dant=0.0, whistl
=0.0, suffer=0.0, confid=0.0, fraud=0.0, brain=0.000481695568400774, m
inu=0.0, twentyf=0.0, cut=0.0, atmospher=0.0, bid=0.0, curv=0.0, pizza
=0.0, bench=0.0, tattoo=0.0, poor=0.0, enid=0.0, pink=0.0, bathroom=0.
```

```
0, cramp=0.0, hill=0.0, sight=0.0, patrol=0.000481695568400774, niec=
0.0, calib=0.000481695568400774, hafta=0.0, journei=0.0, poster=0.0, t
hruster=0.0, dela=0.0, celebr=0.0, myer=0.0, ruth=0.0, suzi=0.0, bunni
=0.0, male=0.0, margi=0.0, bate=0.0, naw=0.0, than=0.00096339113680154
8, msieu=0.0, lone=0.0, mole=0.0, briefcas=0.0, rudi=0.0, excel=0.0, m
adman=0.0, nazi=0.0, flop=0.0, invent=0.0, signor=0.0, suggest=0.0, ed
ward=0.0, station=0.0, senat=0.0, amen=0.0, hip=0.0, price=0.0, awai=
0.000481695568400774, randal=0.0, high=0.0, field=0.00048169556840077
4, spiritu=0.0, tone=0.0, citizen=0.0, stair=0.0, equal=0.0, nor=0.0,
sixth=0.0, gettin=0.0, ground=0.0, control=0.000963391136801548, awak=
0.0, oak=0.0, enterpris=0.0, slightli=0.0, lee=0.0, scope=0.0, holli=
0.0, hunch=0.0, ethic=0.0, nasti=0.0, fall=0.0, wednesdai=0.0, gulf=0.
0, dont=0.009152215799614708, pictur=0.000481695568400774, awar=0.0, s
ponsor=0.0, seattl=0.0, english=0.0, introduc=0.0, health=0.0, halluci
n=0.0, quickli=0.0, jacob=0.0, crew=0.0, cuervo=0.0, alabama=0.0, tere
sa=0.0, bain=0.0, precog=0.0, brake=0.0, professor=0.0, somethin=0.0,
formal=0.0, unload=0.0, curiou=0.0, daddi=0.0, porch=0.0, model=0.0, v
ehicl=0.0, wisdom=0.0, find=0.000481695568400774, ruben=0.0, both=0.0,
report=0.0, denver=0.0, helicopt=0.0, complex=0.000481695568400774, ne
ednt=0.0, be=0.010115606936416251, greet=0.0, spock=0.0, song=0.0, bib
l=0.0, lipstick=0.0, stroke=0.0, persuad=0.0, recommend=0.0, deliveri=
0.0, hors=0.0, shut=0.000481695568400774, zone=0.0, bomb=0.0, linda=0.
0, tube=0.000481695568400774, gross=0.0, order=0.000963391136801548, d
illon=0.0, action=0.0, spot=0.000963391136801548, resign=0.0, barn=0.
0, bush=0.0, itu=0.0, cry=0.000481695568400774, decemb=0.0, manufactur
=0.0, satisfact=0.0, fairi=0.0, excit=0.0, extraordinari=0.0, thoma=0.
0, musician=0.0, black=0.0, mmmmm=0.0, fuel=0.0, board=0.0, joei=0.0,
detail=0.0, stewardess=0.0, hunt=0.0, mum=0.0, preacher=0.0, shown=0.
0, bread=0.0, mind=0.000481695568400774, saint=0.0, oclock=0.0, sail=
0.0, stranger=0.0, smile=0.0, product=0.0, rubi=0.0, disabl=0.0, quest
ion=0.0, philip=0.0, exampl=0.0, freddi=0.0, stori=0.0, chariti=0.0, f
ranklin=0.0, routin=0.0, engin=0.0, lax=0.0, blade=0.0, chew=0.0, bulj
anoff=0.0, counselor=0.0, houston=0.0, alright=0.001926782273603096, s
upport=0.0, wheelchair=0.0, goe=0.000481695568400774, rachel=0.0, stuf
f=0.000963391136801548, neck=0.0, have=0.002890173410404644, divis=0.0
00481695568400774, anniversari=0.0, diamond=0.0, sparazza=0.0, try=0.
0, appar=0.0, joker=0.0, dentist=0.0, section=0.0, gallagh=0.0, slept=
0.0, bank=0.0, hurri=0.0, dure=0.0, sunni=0.0, ink=0.0, vein=0.0, immi
gr=0.0, concentr=0.0, kat=0.0, eager=0.0, don=0.0, somewher=0.0, relig
i=0.0, pierc=0.0, bureau=0.0, theyd=0.0, mueller=0.0, familiar=0.0, bo
nu=0.0, austrian=0.0, violenc=0.0, produc=0.0, tonight=0.0, royal=0.0,
breakin=0.0, map=0.0, sayin=0.0, discuss=0.0, hed=0.0, strictli=0.0, l
ed=0.0, mile=0.0, thisll=0.0, dammit=0.0, leav=0.000481695568400774, d
ummi=0.0, reactor=0.0, sauc=0.0, rice=0.0, coupl=0.0, clever=0.0, choi
c=0.000481695568400774, chrissak=0.0, director=0.0, kastl=0.0, borg=0.
0, fax=0.0, brought=0.0, disturb=0.0, poison=0.0, grissom=0.0, shake=
0.000481695568400774, corrupt=0.0, stall=0.0, sarah=0.0033718689788054
18, starter=0.0, alik=0.0, quiet=0.0, entertain=0.0, demonstr=0.0, oxy
gen=0.0, asham=0.0, undercov=0.0, beef=0.0, nice=0.0, televis=0.0, osc
ar=0.0, tourist=0.0, practic=0.0, mount=0.0, swedish=0.0, compani=0.0,
twenti=0.0, shall=0.0, sherman=0.0, daryl=0.0, forev=0.0, crowd=0.0, t
ap=0.0, fix=0.0, store=0.0, grail=0.0, vietnam=0.0, candl=0.0, whoop=
0.0, taken=0.000481695568400774, end=0.0, eleph=0.0, prefer=0.0, brian
=0.0, lamar=0.0, ma=0.0, victoria=0.0, surviv=0.000481695568400774, th
ree=0.0, bobbi=0.0, stage=0.0, steed=0.0, macfarlan=0.0, spider=0.0, t
rial=0.0, suprem=0.0, awfulli=0.0, foot=0.0, ar=0.004816955684007741,
thrown=0.0, ask=0.000963391136801548, cap=0.0, strang=0.0, phoenix=0.
```

```
0, boot=0.0, georgia=0.0, replac=0.0, reckon=0.0, divorc=0.0, arrest=
0.0, horni=0.0, grandfath=0.0, problem=0.0, bust=0.0, pari=0.0, roomma
t=0.0, consum=0.0, ag=0.0, been=0.0014450867052023214, spaghetti=0.0,
minimum=0.0, car=0.0, recruit=0.0, farm=0.0, dave=0.0, tape=0.0, regul
ar=0.0, decor=0.0, shirt=0.0, multipl=0.0, mechan=0.0, effici=0.0, all
=0.0019267822736030096, thiev=0.0, pose=0.0, silenc=0.0, lenni=0.0, rad
iat=0.0, doesn=0.0, worri=0.0, garrison=0.0, bound=0.0, superior=0.0,
cure=0.0, belli=0.0, mmmm=0.0, of=0.007707129094412382, bride=0.0, hol
e=0.0, toler=0.0, content=0.0, applic=0.0, frequenc=0.0, sore=0.0, ran
ch=0.0, fair=0.0, nowher=0.0, monitor=0.0, peanut=0.0, presid=0.0, coo
per=0.0, speech=0.0, canyon=0.0, humili=0.0, primari=0.0, anchor=0.0,
everi=0.0, fiance=0.0, temporari=0.0, nyah=0.0, greenleaf=0.0, marvel=
0.0, enough=0.000963391136801548, extra=0.0, breast=0.0, properti=0.0,
social=0.0, hug=0.0, tempt=0.0, dracula=0.0, richard=0.0, rob=0.0, gui
nea=0.0, wanna=0.0, hate=0.0014450867052023214, dash=0.0, pull=0.0, la
b=0.000481695568400774, disast=0.0, lobbi=0.0, plug=0.0, rear=0.0, con
veni=0.0, bounti=0.0, auggi=0.0, thee=0.0, proposit=0.0, jeep=0.0, pee
=0.0, josephin=0.0, gestur=0.0, profil=0.0, mimi=0.0, infect=0.0, coin
cid=0.0, mel=0.0, knock=0.0, search=0.0, pan=0.0, if=0.002890173410404
644, odd=0.0, jenni=0.0, howd=0.0, simon=0.0, dodg=0.0, outfit=0.0, fl
oor=0.000481695568400774, uhh=0.0, nah=0.0, jimmi=0.0, chuck=0.0019267
822736030096, rich=0.0, emot=0.0, destruct=0.0, blue=0.0, game=0.0, pec
uliar=0.0, internet=0.0, paranoid=0.0, paradis=0.0, deposit=0.0, look=
0.003371868978805418, count=0.0, ooz=0.0, gale=0.0, plumb=0.0, shouldv
=0.0, giant=0.0, sooz=0.0, gino=0.0, waitress=0.000481695568400774, co
nceiv=0.000481695568400774, messag=0.0014450867052023214, door=0.00048
1695568400774, mondai=0.0, mayor=0.0, fanci=0.0, cuba=0.0, most=0.0009
63391136801548, barrier=0.0, eject=0.0, grew=0.000481695568400774, ali
c=0.0, jazz=0.0, thread=0.000481695568400774, commun=0.0, treati=0.0,
lieuten=0.000481695568400774, tran=0.0, mackelwai=0.0, colonel=0.0, vi
ncent=0.0, pant=0.0, simpl=0.0, shave=0.0, snoop=0.0, flower=0.0, bart
on=0.0, trailer=0.0, towel=0.0, sheldon=0.0, comput=0.0024084778420038
703, medic=0.0, hit=0.0, grow=0.0, ventur=0.0, mistaken=0.0, difficult
i=0.0, thirtyf=0.0, pm=0.0, peter=0.0, orang=0.0, kei=0.0, earl=0.0, e
x=0.0, vault=0.0, doe=0.0, music=0.0, romanc=0.0, blanket=0.0, colleg=
0.0, fog=0.0, bad=0.000481695568400774, novel=0.0, kidnap=0.0, consult
=0.0, recogn=0.0, laugh=0.0, shooter=0.0, galaxi=0.0, paulin=0.0, proc
edur=0.0, seed=0.0, radioact=0.0, ars=0.0, total=0.0, theater=0.0, cut
e=0.0, iraq=0.0, net=0.0, laplant=0.0, waiter=0.0, print=0.0, milo=0.
0, judgment=0.0, act=0.0, gone=0.0014450867052023214, lecter=0.0, draf
t=0.0, build=0.0024084778420038703, vow=0.0, steam=0.0, veteran=0.0, e
ighth=0.0, insect=0.0, entir=0.000481695568400774, hon=0.0, okai=0.003
853564547206192, fourteen=0.0, meet=0.000963391136801548, tens=0.0, do
in=0.0, hat=0.0, sub=0.0, counsel=0.0, channel=0.0, close=0.0, dayligh
t=0.0, loser=0.0, tend=0.0, judgement=0.0, accus=0.0, coast=0.0, every
wher=0.0, welli=0.0, yep=0.0, label=0.0, chines=0.0, kim=0.0, rather=
0.0, publish=0.0, neat=0.0, askin=0.0, pearl=0.0, bond=0.0, labor=0.0,
cover=0.0, forget=0.000963391136801548, warrior=0.0, asid=0.0, favor=
0.000481695568400774, sheep=0.0, crook=0.0, imag=0.0, wimp=0.0, coach=
0.0, seller=0.0, impact=0.0, shoot=0.0, soda=0.0, mitch=0.0, western=
0.0, gift=0.0, shred=0.0, compromis=0.0, fridai=0.000481695568400774,
scout=0.0, steve=0.0, solar=0.0, evan=0.0, rule=0.0, dylan=0.0, han=0.
0, faith=0.0, umyu=0.0, wise=0.0, tortur=0.0, girlfriend=0.00048169556
8400774, anywher=0.0, air=0.0, murder=0.0, younger=0.0, handl=0.000481
695568400774, albanian=0.0, laser=0.0, dull=0.0, guidanc=0.0, live=0.0
024084778420038703, dish=0.0, dancer=0.0, heh=0.0, samuel=0.0, lawn=0.
0, defeat=0.0, photon=0.0, mason=0.0, goal=0.0, prescott=0.0, insur=0.
```

```
0, deep=0.0, lugosi=0.0, singl=0.0, chancellor=0.0, avail=0.0, acknowl
edg=0.0, dread=0.0, manual=0.0, ll=0.0, stew=0.0, pai=0.00096339113680
1548, crawford=0.0, janet=0.0, loan=0.0, rib=0.0, peel=0.0, hormon=0.
0, about=0.005298651252408513, ohio=0.0, advisor=0.0, crusher=0.0, cen
turi=0.0, govern=0.000481695568400774, probabl=0.000481695568400774, s
low=0.0, homicid=0.0, shift=0.0, sentiment=0.0, sophist=0.0, mock=0.0,
supper=0.0, washington=0.0, inn=0.0, polici=0.0, call=0.00192678227360
3096, surgic=0.0, talent=0.0, futur=0.0024084778420038703, appl=0.0, h
iya=0.0, should=0.000481695568400774, hoop=0.0, joseph=0.0, uisu=0.0,
unit=0.000481695568400774, radar=0.0, magnific=0.0, rendezv=0.0, anxio
u=0.0, featur=0.0, discov=0.0, dental=0.0, legal=0.000481695568400774,
shuttl=0.0, word=0.0, tool=0.0, deceas=0.0, consider=0.0, noon=0.00048
1695568400774, york=0.0, lauri=0.0, fell=0.0, reject=0.0, ton=0.0, lea
d=0.0, marin=0.0, name=0.002890173410404644, commiss=0.0, windshield=
0.0, captain=0.0, winter=0.0, slap=0.0, burst=0.0, mallori=0.0, goodni
ght=0.0, polish=0.0, letter=0.0, voyag=0.0, sandi=0.0, ta=0.0, commiss
ion=0.0, joke=0.0, idea=0.0, deserv=0.0, shove=0.0, rd=0.0, mistak=0.0
00481695568400774, earthquak=0.0, worm=0.0, abil=0.0, guarante=0.0, ou
tta=0.0, mutual=0.0, prize=0.0, welcom=0.0, seventi=0.0, trunk=0.0, te
as=0.0, access=0.0, jacki=0.0, refresh=0.0, affirm=0.0, ill=0.00096339
1136801548, gregor=0.0, chet=0.0, complet=0.0, mill=0.0, speak=0.0, wh
atiya=0.0, prei=0.0, fed=0.0, flash=0.0, crab=0.0, wet=0.0, warm=0.0,
hoover=0.0, flood=0.0, regard=0.0, hamburg=0.0, anim=0.0, random=0.0,
lick=0.0, liabl=0.0, unknown=0.0, histori=0.0, highli=0.0, fred=0.0, p
eggi=0.0, emerg=0.0, maid=0.0, relax=0.000481695568400774, whaddya=0.
0, give=0.000963391136801548, starfleet=0.0, rout=0.0, resourc=0.0, ni
ght=0.000481695568400774, statu=0.0, spare=0.0, abov=0.0, save=0.0, fe
tt=0.0, goat=0.0, career=0.0, van=0.0, puke=0.000481695568400774, disa
gre=0.0, reveng=0.0, femal=0.0, penni=0.0, better=0.001445086705202321
4, rex=0.0, marg=0.0, but=0.004816955684007741, border=0.0, phil=0.0,
twentyseven=0.0, sixti=0.0, delai=0.0, jewel=0.0, uiu=0.0, leash=0.0,
berni=0.0, nickel=0.0, aim=0.0, uyouu=0.0, advic=0.0, gold=0.0, actres
s=0.0, whisper=0.0, goin=0.0, particularli=0.0, gossip=0.0, runnin=0.
0, peni=0.0, minut=0.000481695568400774, scandal=0.0, awkward=0.0, cra
nk=0.0, ev=0.0, owen=0.0, diet=0.0, cargo=0.0, fashion=0.0, tale=0.0,
quietli=0.0, son=0.000963391136801548, geek=0.0, devot=0.0, madelein=
0.0, luke=0.0, laval=0.0, seventeen=0.0, yourselv=0.0, sleev=0.0, robb
eri=0.0, debt=0.0, flow=0.0, mark=0.0, subtl=0.0, symbol=0.0, prank=0.
0, illus=0.0, restor=0.0, catch=0.000481695568400774, predat=0.0, temp
er=0.0, globe=0.0, pacif=0.0, ethel=0.0, phaser=0.0, midnight=0.0, loc
k=0.0, adult=0.0, brandon=0.0, kitchen=0.0, noi=0.0, toni=0.0, mo=0.0,
awhil=0.0, actor=0.0, length=0.0, pinch=0.0, throw=0.0, tunnel=0.0, ma
teri=0.0, function=0.0, zip=0.0, visa=0.0, josh=0.0, shed=0.0, crash=
0.0, craig=0.0, he=0.0062620423892100615, perri=0.000481695568400774,
pathet=0.0, eastern=0.0, demand=0.0, sheet=0.0, data=0.0, fbi=0.0, ins
ist=0.0, regist=0.0, fame=0.0, particular=0.0, upon=0.0, screen=0.0, b
ecki=0.0, ya=0.001926782273603096, attornei=0.0, clai=0.0, mate=0.0, i
nterview=0.0, yard=0.0, duke=0.0, footprint=0.0, user=0.0, influenc=0.
0, swallow=0.0, lula=0.0, plate=0.0, queen=0.0, until=0.00048169556840
0774, rode=0.0, creat=0.0, disappoint=0.000481695568400774, region=0.
0, marcia=0.0, defens=0.001926782273603096, sensit=0.0, defend=0.0, mi
seri=0.0, take=0.000963391136801548, oil=0.0, holi=0.0, polit=0.0, cra
zi=0.000481695568400774, interrupt=0.0, luca=0.0, franc=0.0, pete=0.0,
afternoon=0.0, distress=0.0, finder=0.0, around=0.000481695568400774,
europ=0.0, sit=0.0, follow=0.000481695568400774, intellectu=0.0, tee=
0.0, planet=0.0, perk=0.0, dame=0.0, wife=0.0, distanc=0.0, breed=0.0,
height=0.0, compens=0.0, storm=0.0, appoint=0.0, daphn=0.0, salesman=
0.0, scoobi=0.0, distinct=0.0, alon=0.0, remot=0.0, boss=0.00048169556
8400774, pawn=0.0, wonder=0.0, easier=0.0, lt=0.0, zero=0.001445086705
2023214, stake=0.0, food=0.0, forest=0.0, june=0.0, pickup=0.0, sensor
=0.0, choke=0.0, situat=0.0, david=0.0, blackmail=0.0, atom=0.0, neces
sarili=0.0, albert=0.0, advantag=0.0, fli=0.0, horn=0.0, metal=0.00048
1695568400774, consequ=0.0, sloan=0.0, tragic=0.0, determin=0.0, meter
=0.0, toast=0.0, disguis=0.0, loud=0.0, employ=0.0, xxxxxx=0.0, cat=0.
000963391136801548, math=0.0, sue=0.0, foul=0.0, like=0.00770712909441
2382, ferri=0.0, praetor=0.0, marri=0.0, stamp=0.0, lainei=0.0, gai=0.
0, potato=0.0, licens=0.000481695568400774, some=0.002890173410404644,
sylvia=0.0, tag=0.0, caught=0.000963391136801548, runner=0.0, unpleas=
0.0, typic=0.0, dive=0.0, devic=0.0, alien=0.0, deadlin=0.0, landlord=
0.0, bridg=0.0, sink=0.0, affair=0.0, record=0.000481695568400774, thi
nkin=0.0, gate=0.0, austin=0.0, poker=0.0, liquor=0.0, cow=0.0, run=0.
000963391136801548, yourself=0.0, tie=0.0, slightest=0.0, businessman=
0.0, martini=0.0, dell=0.0, stone=0.0, iti=0.0, dri=0.0, scanner=0.000
481695568400774, effect=0.000481695568400774, appreci=0.0, shark=0.0,
child=0.000481695568400774, bend=0.0, slice=0.0, hair=0.0, patienc=0.
0, ga=0.0, disco=0.0, expert=0.0, implant=0.0, dealer=0.0, oooh=0.0, b
eavi=0.0, troubl=0.0, lesli=0.0, lectur=0.0, youd=0.00096339113680154
8, baldwin=0.0, scienc=0.0, surg=0.0, uallu=0.0, ooh=0.0, harvard=0.0,
smart=0.0, statist=0.0, sheila=0.0, haul=0.0, livingston=0.0, exhaust=
0.0, jail=0.000481695568400774, biggest=0.0, hudsuck=0.0, rocket=0.0,
trace=0.000481695568400774, punish=0.0, contain=0.0, twomblei=0.0, int
end=0.0, flynn=0.0, describ=0.0, contract=0.0, straighten=0.0, reilli=
0.0, regula=0.0, tremend=0.0, love=0.0, setup=0.0, enforc=0.0, castl=
0.0, lousi=0.0, bloke=0.0, marietta=0.0, cooki=0.0, someon=0.000963391
136801548, design=0.0, address=0.0, knowi=0.0, award=0.0, tick=0.0, un
consci=0.0, weve=0.0, cleveland=0.0, sweeti=0.0, intellect=0.0, beard=
0.0, surrend=0.0, vinci=0.0, hildi=0.0, tail=0.0, dignan=0.0, moss=0.
0, photo=0.0, myth=0.0, tuesdai=0.0, stress=0.0, limb=0.0, wallet=0.0,
tooth=0.0, bare=0.0, ten=0.000481695568400774, guilti=0.0, inform=0.0,
list=0.0, husband=0.0, visual=0.0, ross=0.0, jabez=0.0, cape=0.0, atla
nta=0.0, champion=0.0, valuabl=0.0, respons=0.0, addit=0.0, spike=0.0,
lighten=0.0, wagon=0.0, owl=0.0, dog=0.0014450867052023214, santa=0.0,
fantasi=0.0, belt=0.0, unowu=0.0, enorm=0.0, kind=0.0, light=0.0009633
91136801548, investig=0.0, lila=0.0, laundri=0.0, jessica=0.0, jodi=0.
0, masturb=0.0, porter=0.0, la=0.0, couldn=0.0, canada=0.0, dress=0.0,
port=0.0, alwai=0.0, explor=0.0, bateman=0.0, governor=0.0, sat=0.0, f
ought=0.0, perimet=0.0, wasnt=0.000481695568400774, testimoni=0.0, con
klin=0.0, cart=0.0, caus=0.0, number=0.0, crane=0.0, loyalti=0.0, fake
=0.0, puff=0.0, accompani=0.0, cultur=0.0, fenc=0.0, outa=0.0, cal=0.
0, whenev=0.0, roach=0.0, transmitt=0.0, fingerprint=0.0, mail=0.0, ka
ufman=0.0, pole=0.0, engag=0.0, plain=0.0, suppli=0.0, reserv=0.0, few
=0.000481695568400774, sandwich=0.0, fredo=0.0, safe=0.000963391136801
548, hockei=0.0, hous=0.0, elain=0.0, knox=0.0, oath=0.0, wander=0.0,
latin=0.0, same=0.0024084778420038703, git=0.0, garbag=0.0, charg=0.0,
knowledg=0.0, uknowu=0.0, veronica=0.0, john=0.001926782273603096, con
nel=0.0, anticip=0.0, drown=0.0, rose=0.0, carpet=0.0, circu=0.0, seba
stian=0.0, cmere=0.0, irish=0.0, robert=0.0, fork=0.0, lazi=0.0, flare
=0.0, bobo=0.0, guess=0.000481695568400774, hippi=0.0, the=0.040462427
74566501, specif=0.0, offici=0.000963391136801548, god=0.0, exhibit=0.
0, madam=0.0, privileg=0.0, scrambl=0.0, con=0.0, handi=0.0, rifl=0.0,
peep=0.0, moron=0.0, resent=0.0, lai=0.000481695568400774, scotch=0.0,
brenner=0.0, lake=0.0, therel=0.0, gordo=0.0, everett=0.0, neural=0.0,
madison=0.0, realiz=0.0, georg=0.0, skate=0.0, drink=0.000481695568400
774, goddam=0.0, anyhow=0.0, jew=0.0, popul=0.0, freedom=0.0, togeth=
0.0, attempt=0.0, adel=0.0, coat=0.0, mob=0.0, higher=0.0, cowboi=0.0,
heel=0.0, inconveni=0.0, respond=0.0, patch=0.0, mostli=0.0, fan=0.0,
highwai=0.0, reason=0.000481695568400774, soon=0.000963391136801548, h
elpless=0.0, then=0.003371868978805418, cobb=0.0, weird=0.000963391136
801548, parker=0.0, seen=0.000963391136801548, happier=0.0, roast=0.0,
tip=0.000481695568400774, legend=0.000963391136801548, behavior=0.0, s
ecur=0.0, tatum=0.0, especi=0.000481695568400774, jeez=0.0, your=0.009
63391136801548, dispatch=0.0, heart=0.000481695568400774, test=0.0, we
bster=0.0, categori=0.0, chicken=0.0, alibi=0.0, wouldn=0.0, hyster=0.
0, mankind=0.0, north=0.0, kiss=0.0, mighti=0.0, levi=0.0, fond=0.0, r
ain=0.0, obsess=0.0, michael=0.0, corleon=0.0, vision=0.0, edg=0.0, co
mmand=0.0, nineti=0.0, gin=0.0, nix=0.0, amus=0.0, afterward=0.0, want
a=0.0, thorwald=0.0, by=0.002890173410404644, rag=0.0, jack=0.0, deser
t=0.0, maureen=0.0, own=0.000963391136801548, heck=0.0, walker=0.0, em
pti=0.0, grunemann=0.0, relief=0.0, narrow=0.0, saavik=0.0, seymour=0.
0, anonym=0.0, footag=0.0, territori=0.0, maniac=0.0, envelop=0.0, des
pis=0.0, definit=0.0, return=0.0, nobodi=0.0024084778420038703, eventu
=0.0, rita=0.0, pepper=0.0, prep=0.0, stand=0.0, deepli=0.0, park=0.0,
aunt=0.0, youth=0.0, water=0.0, els=0.001926782273603096, starl=0.0, a
llei=0.0, snow=0.0, pope=0.0, wax=0.0, object=0.000481695568400774, ba
sic=0.000481695568400774, file=0.0, dismiss=0.0, much=0.00192678227360
3096, sometim=0.000481695568400774, warrant=0.0, drama=0.0, android=0.
0, wast=0.0, brick=0.0, restless=0.0, unusu=0.0, orbit=0.0, headach=0.
0, ambul=0.0, pier=0.0, talkin=0.0, instal=0.0, toss=0.0, pure=0.0, ma
xin=0.0, canadian=0.0, amount=0.0, shatter=0.0, spoke=0.0, danc=0.0, i
r=0.0, proven=0.0, broke=0.0, ninotchka=0.0, lili=0.0, dat=0.0, scum=
0.0, psychic=0.0, guest=0.0, architect=0.0, vancouv=0.0, balconi=0.0,
my=0.003853564547206192, resist=0.0, juic=0.0, reliabl=0.0, will=0.003
853564547206192, launch=0.0, tire=0.0, villag=0.0, ordel=0.0, broad=0.
0, juri=0.0, mccoi=0.0, italian=0.0, item=0.0, process=0.0, sake=0.0,
diseas=0.0, kelli=0.0, contest=0.0, champagn=0.0, cabinet=0.0, plead=
0.0, era=0.0, sergeant=0.0, nervou=0.0, imposs=0.0, cruel=0.0, claud=
0.0, movi=0.000481695568400774, cruis=0.0, daughter=0.0, premier=0.0,
law=0.0, intent=0.0, liz=0.0, nake=0.000481695568400774, petti=0.0, mi
rror=0.0, ha=0.001926782273603096, each=0.0, never=0.00240847784200387
03, do=0.005298651252408513, exact=0.0, rescu=0.0, drivin=0.0, limit=
0.0, ti=0.0, manag=0.000481695568400774, thirsti=0.0, bargain=0.000481
695568400774, origin=0.0, acceler=0.0, pilot=0.0, guid=0.0, mummi=0.0,
mad=0.0, frighten=0.0, factori=0.000481695568400774, cotton=0.0, rape=
0.0, chef=0.0, leg=0.0, calm=0.0, fail=0.0, interior=0.0, germ=0.0, in
ch=0.0, donald=0.0, smear=0.0, berserk=0.0, finest=0.0, hard=0.0004816
95568400774, bourn=0.0, brilliant=0.0, solv=0.0, fireman=0.0, electr=
0.0, moment=0.0, wound=0.0, begun=0.0, while=0.000481695568400774, mil
itari=0.000481695568400774, scoop=0.0, saw=0.000481695568400774, machi
n=0.0014450867052023214, german=0.0, enemi=0.000963391136801548, creat
ur=0.0, throat=0.0, stuf=0.0, impli=0.0, burnt=0.0, mall=0.0, navi=0.
0, rude=0.0, lean=0.0, bullet=0.0, sissi=0.0, dough=0.0, reynold=0.0,
promot=0.0, burn=0.0, rise=0.0, guard=0.0, deton=0.0, gloriou=0.0, con
fidenti=0.0, lost=0.000963391136801548, desk=0.0, thin=0.0, superhero=
0.0, wive=0.0, pooch=0.0, col=0.0, gambl=0.0, crimin=0.0, can=0.005780
3468208092856, tiger=0.0, pair=0.0, below=0.0, him=0.00481695568400774
1, un=0.0, mortgag=0.0, toto=0.0, dine=0.0, gun=0.0, thirti=0.00096339
1136801548, warp=0.0, sid=0.0, standard=0.0, deliv=0.0, couldv=0.0, sa
ndra=0.0, wide=0.0, daydai=0.0, nail=0.0, novemb=0.0, speck=0.0, broke
n=0.0, fulli=0.0, di=0.000481695568400774, concept=0.0, plenti=0.0, sa
ra=0.0, toilet=0.0, smyth=0.0, dirt=0.0, skull=0.0, match=0.0, win=0.0
00481695568400774, abduct=0.0, chamber=0.0, putter=0.0, advanc=0.0, ba
```

```
xter=0.0, group=0.0, onli=0.0, commerci=0.0, uniqu=0.0, bet=0.00048169
5568400774, back=0.0014450867052023214, aisl=0.0, darlin=0.0, creation
=0.0, target=0.0, sip=0.0, sam=0.0, worker=0.0, five=0.0, ladder=0.0,
capit=0.0, smaller=0.0, merri=0.0, won=0.000481695568400774, howard=0.
0, eleg=0.0, bee=0.0, digit=0.0, lowel=0.0, outstand=0.0, meant=0.0009
63391136801548, precaut=0.0, degre=0.0, suspend=0.0, week=0.0004816955
68400774, evacu=0.0, wow=0.0, loui=0.0, iii=0.0, per=0.0, tenth=0.0, o
pen=0.0, possibl=0.000963391136801548, old=0.0, primit=0.0, keyser=0.
0, fianc=0.0, point=0.000481695568400774, bone=0.0, alreadi=0.00144508
67052023214, straight=0.0, henri=0.0, hack=0.0, rank=0.0, scott=0.0, f
old=0.0, nothin=0.0, kendal=0.0, bar=0.0, ant=0.0, meredith=0.0, musta
=0.0, dollar=0.0, retard=0.0, volum=0.0, ocean=0.0, acquaint=0.0, jon=
0.0, dozen=0.0, treat=0.000481695568400774, fund=0.0, crush=0.0, accom
plish=0.0, cabl=0.0, stock=0.0, pig=0.0, recent=0.0, ed=0.0, escort=0.
0, east=0.0, vodka=0.0, natur=0.0, fella=0.0, missil=0.0, ancient=0.0,
how=0.0014450867052023214, rumor=0.0, torn=0.0, palei=0.0, well=0.0014
450867052023214, fear=0.000481695568400774, detroit=0.0, civil=0.0, cu
rrent=0.0, advis=0.0, elect=0.0, pill=0.0, artist=0.0, bastaldi=0.0, m
arket=0.0, symptom=0.0, therapi=0.0, toi=0.0, scumbag=0.0, satan=0.0,
ouch=0.0, jeff=0.0, cemeteri=0.0, sorta=0.0, deni=0.0, sister=0.0, giv
in=0.0, boi=0.0, closest=0.0, mission=0.000963391136801548, stabil=0.
0, identif=0.0, set=0.001926782273603096, rebel=0.0, salli=0.0, joint=
0.0, zoo=0.0, version=0.000481695568400774, camper=0.0, directli=0.0,
kid=0.001926782273603096, jim=0.0, viru=0.0, desir=0.0, tide=0.0, cell
=0.0, mulwrai=0.0, sod=0.0, domini=0.0, dealt=0.0, pane=0.0, trade=0.
0, spose=0.0, eleanor=0.0, someth=0.0014450867052023214, milk=0.0, hon
estli=0.0, sap=0.0, graham=0.0, tune=0.0, secretari=0.0, chao=0.0, bac
kup=0.0, freez=0.000481695568400774, obvious=0.0, believ=0.0, delmar=
0.0, sting=0.0, happiest=0.0, broadcast=0.0, haunt=0.0, haven=0.0, kon
g=0.0, gonna=0.0024084778420038703, without=0.000481695568400774, lip=
0.0, spill=0.0, psychiatrist=0.0, suddenli=0.0, realli=0.0004816955684
00774, byeby=0.0, coffe=0.0, opinion=0.0, borrow=0.0, lechter=0.0, cuf
f=0.0, overload=0.0, attract=0.0, rid=0.0, bertrand=0.0, mordechai=0.
0, amanda=0.0, nicknam=0.0, gather=0.0, mr=0.000481695568400774, miss=
0.0, fascin=0.0, hound=0.0, lid=0.0, joi=0.0, weak=0.0, though=0.0, al
fr=0.0, comfort=0.0, ich=0.0, gallon=0.0, narcot=0.0, evelyn=0.0, duck
=0.0, bucket=0.0, cabin=0.0, settlement=0.0, send=0.00096339113680154
8, milli=0.0, israel=0.0, alcohol=0.0, suspect=0.000481695568400774, r
obinson=0.0, doug=0.0, alan=0.0, sleepi=0.0, adventur=0.0, ridicul=0.
0, crabtre=0.0, lawyer=0.0, reward=0.0, cloth=0.0, riot=0.0, seek=0.0,
other=0.000963391136801548, pet=0.0, everyth=0.0024084778420038703, fo
und=0.000963391136801548, basketbal=0.0, liar=0.0, wouldnt=0.0, fit=0.
0, stop=0.000963391136801548, cleaner=0.0, paul=0.0, execut=0.0, alpha
=0.0, movement=0.0, cheek=0.0, recov=0.0, phillip=0.0, seem=0.0, leo=
0.0, rand=0.0, bright=0.0, moon=0.0, fallen=0.0, prayer=0.0, betti=0.
0, fantast=0.0, theyv=0.000481695568400774, drove=0.0, turkei=0.0, mod
ern=0.0, edi=0.0, jonah=0.0, exagger=0.0, mine=0.0, envi=0.0, big=0.00
0481695568400774, krueger=0.0, faster=0.0, alli=0.0, fine=0.0, iraqi=
0.0, appear=0.0, independ=0.0, poni=0.0, invest=0.0, hooker=0.0, dicta
t=0.0, medicin=0.0, honei=0.0, shine=0.0, comedi=0.0, unou=0.0, who=0.
002890173410404644, late=0.0014450867052023214, boost=0.0, gentli=0.00
0481695568400774, sun=0.0, dunno=0.0, improv=0.0, chairman=0.0, honest
=0.0, shout=0.0, invas=0.0, hadnt=0.0, fox=0.0, powder=0.0, quick=0.0,
breaker=0.0, breach=0.0, dope=0.0, frustrat=0.0, cent=0.0, long=0.0, i
nterest=0.0, freak=0.0, marsh=0.0, maria=0.0, shhhh=0.0, cloak=0.0, so
lid=0.0, id.1=0.000481695568400774, nobl=0.0, swing=0.0, evalu=0.0, me
mber=0.0, skunk=0.0, zira=0.0, conduct=0.0, buyer=0.0, instanc=0.0, fi
```

```
nanc=0.0, protest=0.0, karl=0.0, dent=0.0, jake=0.0, goofi=0.0, au=0.
0, plu=0.0, ahead=0.000481695568400774, possess=0.0, prototyp=0.0, wha
td=0.0, road=0.0, fifti=0.0, down=0.000963391136801548, viciou=0.0, de
stin=0.0, ride=0.0, hmm=0.0, democrat=0.0, daili=0.0, remov=0.0, cathe
rin=0.0, unlock=0.0, hot=0.0014450867052023214, yuh=0.0, lewi=0.0, con
cern=0.0, mouth=0.0, didn=0.0, merci=0.000481695568400774, lotta=0.0,
sacrific=0.0, chekov=0.0, rock=0.0, instead=0.0, egon=0.0, sank=0.0, p
robe=0.0, pit=0.0, rest=0.000481695568400774, confront=0.0, magic=0.0,
mean=0.000481695568400774, similar=0.0, itd=0.0, flesh=0.0004816955684
00774, bunch=0.0, donut=0.0, soldier=0.000481695568400774, dork=0.0, s
hort=0.000481695568400774, innoc=0.0, fight=0.000481695568400774, grai
=0.0, bout=0.0, befor=0.001926782273603096, encount=0.0, lombardo=0.0,
arriv=0.0, date=0.0, worn=0.0, frankenstein=0.0, budget=0.0, gag=0.0,
part=0.0, forgot=0.0, blake=0.0, himself=0.0, facil=0.0, thousand=0.0,
frankli=0.0, whatcha=0.0, struggl=0.0, lad=0.0, a=0.02263969171483636
7, resid=0.0, salad=0.0, told=0.000481695568400774, brother=0.0, rmph=
0.0, which=0.0, philosoph=0.0, clear=0.0, cracker=0.0, zavitz=0.0, tel
l=0.002890173410404644, conclus=0.0, collin=0.0, shhh=0.0, tickl=0.0,
delus=0.0, ship=0.0, band=0.0, naiv=0.0, sever=0.0, wreck=0.0, torch=
0.0, gym=0.0, bang=0.0, man=0.000481695568400774, aid=0.0, purchas=0.
0, review=0.0, lantern=0.0, new=0.0024084778420038703, nunez=0.0, wish
=0.0, nest=0.0, fortun=0.0, bud=0.0, excus=0.0, rattl=0.0, knight=0.0,
broadwai=0.0, destini=0.0, smell=0.0, rick=0.0, mommi=0.0, klingon=0.
0, staci=0.0, calcul=0.0, signific=0.0, arrog=0.0, mac=0.0, offic=0.00
0481695568400774, pitch=0.0, detect=0.0, hood=0.0, relationship=0.0, s
unshin=0.0, arlyn=0.0, gentl=0.0, heard=0.000481695568400774, babe=0.
0, beg=0.000481695568400774, princess=0.0, memphi=0.0, addict=0.0, inc
id=0.0, reput=0.0, forth=0.0, club=0.0, theme=0.0, anthoni=0.0, corn=
0.0, suitcas=0.0, went=0.0, worthi=0.0, jessi=0.0, announc=0.0, panic=
0.0, pleasant=0.0, bath=0.0, pauli=0.0, sampl=0.0, policeman=0.0, co=
0.0, sonni=0.0, gave=0.000481695568400774, weigh=0.0, attic=0.0, incid
ent=0.0, quarantin=0.0, got=0.004335260115606965 5, napkin=0.0, chair=
0.0, grandmoth=0.0, theft=0.0, disord=0.0, blame=0.0, stronger=0.00048
1695568400774, perform=0.0, absurd=0.0, mitchel=0.0, cab=0.0, mutant=
0.0, corbett=0.0, travi=0.0, erica=0.0, pressur=0.0, larg=0.0004816955
68400774, toddi=0.0, gum=0.0, trap=0.0, twelv=0.0, treatment=0.0, xrai
=0.0, callin=0.0, rope=0.0, japan=0.0, pot=0.0, method=0.0, enter=0.0,
confess=0.0, killer=0.0014450867052023214, constantli=0.0, scar=0.0, c
oncert=0.0, thirteen=0.0, debbi=0.0, knot=0.0, pension=0.0, bob=0.0004
81695568400774, home=0.000963391136801548, walt=0.0, troi=0.0, argumen
t=0.0, desper=0.0, jeremi=0.0, epp=0.0, havent=0.000963391136801548, u
gh=0.0, dewei=0.0, dumb=0.0, deputi=0.0, phone=0.000481695568400774, w
orf=0.0, spook=0.0, avenu=0.0, hm=0.0, keep=0.0014450867052023214, hap
pen=0.0, plane=0.0, cartel=0.0, brush=0.0, betrai=0.0, geez=0.0, spark
=0.0, tryin=0.0, injur=0.0, lauren=0.0, jam=0.0, jill=0.0, mikei=0.0,
cocain=0.0, drop=0.0, harri=0.0, paus=0.0, space=0.0, sound=0.0, julia
=0.0, warehous=0.0, remark=0.0, rum=0.0, snap=0.0, meantim=0.0, candi=
0.0, proper=0.0, realist=0.0, arthur=0.0, sh=0.0, survivor=0.000481695
568400774, teeth=0.0, de=0.0, sens=0.0, maneuv=0.0, poetri=0.0, overni
ght=0.0, bow=0.0, dalla=0.0, discharg=0.0, bill=0.0, fly=0.0, realiti=
0.0, compar=0.000481695568400774, upstair=0.0, pritchett=0.0, terrif=
0.0, anoth=0.000963391136801548, kiddin=0.0, unfair=0.0, brace=0.0, ch
imera=0.0, feather=0.0, regul=0.0, hump=0.0, thank=0.00144508670520232
14, matur=0.0, valu=0.0, even=0.0024084778420038703, shoulder=0.0, boo
th=0.0, ribbon=0.0, ceas=0.0, base=0.0, dan=0.0, smokei=0.0, coup=0.0,
thick=0.0, bui=0.000481695568400774, stud=0.0, scenario=0.0, laid=0.0,
mexican=0.0, prior=0.0, bra=0.0, leader=0.0, charli=0.0, fairli=0.0, n
```

```
urs=0.0, ahhh=0.0, texa=0.0, lui=0.0, except=0.000481695568400774, aug
ust=0.0, sittin=0.0, bait=0.0, power=0.000481695568400774, kansa=0.0,
 scotti=0.0, swipe=0.0, videotap=0.0, opposit=0.0, expand=0.0, neighbo
rhood=0.0, out=0.006743737957610836, melt=0.0, difficult=0.0, crow=0.
0, occup=0.0, swine=0.0, expens=0.0, brad=0.0, balanc=0.00048169556840
0774, tension=0.0, smarter=0.0, team=0.0, salari=0.0, expos=0.0, incom
=0.0, special=0.000481695568400774, thea=0.0, needl=0.0, aggress=0.0,
 necessari=0.0, burk=0.0, uniform=0.0, crawl=0.0, hundr=0.0, radio=0.
0, into=0.000481695568400774, singer=0.0, explan=0.0, sollozzo=0.0, be
auti=0.0, whole=0.000963391136801548, attach=0.0, bleed=0.0, card=0.0,
longer=0.0, good=0.0024084778420038703, draw=0.0, rate=0.0, enlighten=
0.0, startin=0.0, demon=0.0, street=0.0, thatll=0.0, convers=0.0, nerv
=0.0, telegram=0.0, therer=0.000481695568400774, stan=0.0, momma=0.0,
 nightmar=0.0, combat=0.000481695568400774, brand=0.0, greatest=0.0, c
olor=0.0, sourc=0.0, swap=0.0, preciou=0.0, greas=0.0, manipul=0.0, bl
ood=0.0, batman=0.0, hitler=0.0, visit=0.0, india=0.0, poke=0.0, balti
mor=0.0, tear=0.0, fate=0.000963391136801548, cours=0.0004816955684007
74, stu=0.0, vampir=0.0, vessel=0.0, creep=0.0, margaret=0.0, bigger=
0.0, error=0.0, repair=0.0, grab=0.0, nope=0.0, period=0.0, construct=
0.0, email=0.0, shinzon=0.0, activ=0.0, bradi=0.0, sprai=0.0, steel=0.
0, shack=0.0, til=0.000963391136801548, woodi=0.0, roman=0.0, risk=0.
0, peach=0.0, instruct=0.0, plagu=0.0, provid=0.0, grandma=0.0, tough=
0.000481695568400774, schmuck=0.0, souvenir=0.0, stain=0.0, wine=0.0,
 carla=0.0, mckenna=0.0, rexroth=0.0, spend=0.0, i=0.0366088631984588
3, susi=0.0, brodi=0.0, far=0.0, sweater=0.0, appeal=0.0, vibe=0.0, po
et=0.0, oz=0.0, finch=0.0, busi=0.000481695568400774, perfect=0.0, boy
friend=0.0, lemon=0.0, hand=0.0, charm=0.0, sustain=0.0, retain=0.0, c
ontinu=0.0, hammer=0.0, ripper=0.0, personnel=0.0, beaten=0.0, remind=
0.0, joan=0.0, witch=0.0, second=0.0014450867052023214, impress=0.0, h
int=0.0, avoid=0.0, billi=0.0, wipe=0.000481695568400774, calvin=0.0,
 shrink=0.0, darn=0.0, is=0.007707129094412382, show=0.000481695568400
774, first=0.0014450867052023214, intellig=0.0, kit=0.0, collar=0.0, b
undi=0.0, feel=0.001926782273603096, martha=0.0, fortyeight=0.0, frank
i=0.0, rehears=0.0, reunion=0.0, previou=0.0, fire=0.00048169556840077
4, specimen=0.0, exactli=0.0, gotta=0.000963391136801548, claric=0.0,
 you=0.039017341040462686, psychot=0.0, liberti=0.0, ever=0.0009633911
36801548, indian=0.0, karen=0.0, ear=0.0, uncl=0.0, mar=0.0, surround=
0.000481695568400774, wood=0.0, airport=0.0, jean=0.0, prescript=0.0,
 occupi=0.0, slave=0.0, yup=0.0, cri=0.0, threw=0.0, extort=0.0, cheap
=0.0, melvin=0.0, strain=0.0, solut=0.0, skye=0.0, packag=0.0, young=
0.000481695568400774, agre=0.0, purs=0.0, puppet=0.0, clearli=0.0, ell
en=0.0, serious=0.0, applejack=0.0, threaten=0.0, swear=0.0, journalis
t=0.0, beth=0.0, feelin=0.0, danni=0.0, hmmm=0.0, stomach=0.0, certif=
0.0, them=0.0024084778420038703, wigand=0.0, electron=0.0, surfac=0.0,
op=0.0, edit=0.0, backward=0.0, buckaroo=0.0, fare=0.0, mhm=0.0, cynic
=0.0, fee=0.0, blown=0.0, workin=0.0, eric=0.0, gear=0.0, chill=0.0, p
ublic=0.000481695568400774, elizabeth=0.0, dy=0.000481695568400774, gr
eat=0.000963391136801548, load=0.0, melani=0.0, unless=0.0, mora=0.0,
 nativ=0.0, gene=0.0, wind=0.000481695568400774, code=0.0, bribe=0.0,
 dieter=0.0, boil=0.0, deliber=0.0, ritual=0.0, dolor=0.0, pride=0.0,
 molli=0.0, felt=0.0, oswald=0.0, drank=0.0, el=0.000481695568400774,
 soap=0.0, suck=0.0, psycho=0.0, prom=0.0, dig=0.0, sword=0.0, exchang
=0.0, duti=0.0, parti=0.0, juliet=0.0, seventh=0.0, given=0.0, hawk=0.
0, fry=0.0, thei=0.0043352601156069655, vinc=0.0, notion=0.0, consid=
0.0, upset=0.0, ve=0.0, yeah=0.000481695568400774, compet=0.0, ticket=
0.0, doom=0.0, ski=0.0, courthous=0.0, alert=0.0, tan=0.0, nam=0.0, po
sit=0.0, urgent=0.0, tast=0.0, makeup=0.0, parol=0.0, schwartz=0.0, do
```

```
ctor=0.000963391136801548, aw=0.0, christ=0.0, scari=0.0, walkin=0.0,
 wrist=0.0, cattl=0.0, truck=0.0, eighteen=0.0, ethan=0.0, idiot=0.0,
 hotel=0.0, hook=0.000963391136801548, notic=0.0, heali=0.0, benjamin=
0.0, octob=0.0, ought=0.0, isnt=0.000481695568400774, drew=0.0, whethe
r=0.0, seat=0.0, matter=0.0, er=0.0, int=0.0, arent=0.0004816955684007
74, aubrei=0.0, lookout=0.0, gabe=0.0, cliff=0.0, hail=0.0, horribl=0.
0, heaven=0.0, hospit=0.0, whatsoev=0.0, supernatur=0.0, lot=0.0004816
95568400774, establish=0.0, select=0.0, motor=0.0, brenda=0.0, guilt=
0.0, glori=0.0, move=0.000963391136801548, rage=0.0, decent=0.0, relig
ion=0.0, pillow=0.0, sincer=0.0, steven=0.0, winston=0.0, arm=0.0, sys
tem=0.000481695568400774, score=0.0, isn=0.0, wallac=0.0, lane=0.0, bu
rger=0.0, bowl=0.0, umeu=0.0, journal=0.0, lonnegan=0.0, teddi=0.0, pd
=0.0, trooper=0.0, puppi=0.0, prosecut=0.0, badli=0.0, thought=0.00048
1695568400774, troop=0.0, forti=0.000963391136801548, highest=0.0, dus
t=0.0, begin=0.000481695568400774, lapd=0.0, priest=0.0, turk=0.0, haw
aii=0.0, amateur=0.0, honor=0.000963391136801548, ap=0.0, lodg=0.0, lo
mbard=0.0, academi=0.0, gibson=0.0, injuri=0.0, wha=0.0, therefor=0.0,
victor=0.0, profess=0.0, strike=0.0, deed=0.0, grand=0.0, penthous=0.
0, yellow=0.0, insult=0.0, genuin=0.0, ward=0.0, mi=0.0, stolen=0.0, b
anana=0.0, speed=0.0, piec=0.0, born=0.000963391136801548, oughta=0.0,
hacker=0.0, stream=0.0, smith=0.0, choir=0.0, variou=0.0, weed=0.0, bo
undari=0.0, panti=0.0, accord=0.0, best=0.0, condom=0.0, whatll=0.0, c
arlo=0.0, promis=0.0, barrett=0.0, pentagon=0.0, sooner=0.0, nexu=0.0,
bloodi=0.0, said=0.0014450867052023214, mississippi=0.0, legitim=0.0,
 syndrom=0.0, repli=0.0, purpl=0.0, break=0.0, wash=0.0, motel=0.0, wr
ap=0.0, fifteen=0.0, sell=0.0, bravo=0.0, world=0.000481695568400774,
 beaumont=0.0, switch=0.0, adrian=0.0, after=0.000963391136801548, van
ish=0.0, window=0.0, interfer=0.0, countri=0.000481695568400774, wretc
h=0.0, sugar=0.0, captur=0.000481695568400774, wed=0.00144508670520232
14, adrenalin=0.0, dorothi=0.0, jude=0.0, gorgeou=0.0, corneliu=0.0, f
lip=0.0, beyond=0.0, mix=0.0, bandit=0.0, rabbit=0.0, rare=0.0, cecil=
0.0, belong=0.0, collaps=0.0, resort=0.0, cuz=0.0, trevor=0.0, lovebir
d=0.0, express=0.0, dutch=0.0, scare=0.0, kept=0.0, stephani=0.0, sign
atur=0.0, warren=0.0, battl=0.0, insight=0.0, next=0.00048169556840077
4, lamp=0.0, trail=0.0, kent=0.0, forg=0.0, ran=0.0, proce=0.0, shade=
0.0, asylum=0.0, kubelik=0.0, vallen=0.0, in=0.011560693641618576, eve
ryon=0.0, kidnei=0.0, hostil=0.0, pie=0.0, written=0.0, bye=0.0, refer
=0.0, explos=0.0, tucker=0.0, swayzak=0.0, lord=0.0, room=0.0, split=
0.0, udou=0.0, whack=0.0, lo=0.0, drawer=0.0, wild=0.0, hire=0.0, amon
g=0.0, disgust=0.0, pocket=0.0, margo=0.0, ignor=0.0, goodlook=0.0, cl
aim=0.0, approach=0.0, fruit=0.0, raid=0.0, phrase=0.0, luck=0.0, kevi
n=0.0, autograph=0.0, core=0.0, download=0.0, evil=0.0, strand=0.0, wo
ulda=0.0, branch=0.0, ghost=0.0, pack=0.0, robber=0.0, dime=0.0, angri
=0.0, ju=0.0, uyouru=0.0, drill=0.0, preserv=0.0, serial=0.0, option=
0.0, oldest=0.0, pattern=0.000481695568400774, herb=0.0, row=0.0, ball
=0.0, equip=0.000481695568400774, drift=0.0, mike=0.0, tyler=0.0, choo
s=0.0, nag=0.0, bookstor=0.0, happi=0.0, haircut=0.0, candid=0.0, tim=
0.0, friend=0.0, sneak=0.0, those=0.0, wayn=0.0, straw=0.0, cynthia=0.
0, ad=0.0, two=0.0014450867052023214, eyebal=0.0, carv=0.0, overlook=
0.0, contractor=0.0, languag=0.0, fortyf=0.0, admit=0.0, slight=0.0, p
opular=0.0, compliment=0.0, del=0.0, listen=0.0014450867052023214, mes
s=0.0, free=0.0, keen=0.0, noth=0.0014450867052023214, acquir=0.0, ear
lier=0.0, bluff=0.0, twentyfour=0.0, steak=0.0, california=0.0, dial=
0.0, industri=0.0, hopefulli=0.0, probli=0.0, certainli=0.0, magnet=0.
0, mobil=0.0, elvi=0.0, parlor=0.0, utah=0.0, we=0.006743737957610836,
phoni=0.0, sec=0.0, fellow=0.0, eagl=0.0, achiev=0.0, denni=0.0, head=
0.0, bore=0.0, dyou=0.0, junior=0.0, bite=0.0, through=0.0028901734104
```

```
04644, grenad=0.0, phase=0.000481695568400774, fabul=0.0, diplomat=0.
0, wrong=0.000963391136801548, itll=0.0014450867052023214, again=0.000
963391136801548, dock=0.0, bear=0.0, minor=0.0, white=0.00048169556840
0774, environ=0.0, patient=0.0, mug=0.0, lighthous=0.0, upsid=0.0, nas
al=0.0, navig=0.0, sad=0.0, research=0.0, case=0.0014450867052023214,
 gu=0.0, brutal=0.0, aint=0.0, sinc=0.0, nichola=0.0, intrud=0.0, dare
=0.0, were=0.003853564547206192, import=0.000481695568400774, pi=0.0,
 babysit=0.0, race=0.0, oh=0.000481695568400774, flag=0.0, shaw=0.0, t
eacher=0.0, pimp=0.0, diego=0.0, victim=0.0, anymor=0.0004816955684007
74, round=0.000481695568400774, rough=0.0, occas=0.0, sulu=0.0, woke=
0.0, chose=0.0, convent=0.0, prioriti=0.0, treadston=0.0, drag=0.0, ma
ud=0.0, monster=0.0, ic=0.0, or=0.0024084778420038703, gener=0.0004816
95568400774, penguin=0.0, earth=0.0, dc=0.0, cave=0.0, fleet=0.0, wrot
e=0.0, goddammit=0.0, venic=0.0, wade=0.0, spoken=0.0, unfortun=0.0, c
ampaign=0.0, fever=0.0, marcu=0.0, almost=0.000481695568400774, these=
0.000481695568400774, sick=0.0, reach=0.000481695568400774, screw=0.00
0963391136801548, grace=0.0, anyth=0.0014450867052023214, empir=0.0, r
ack=0.0, nell=0.0, divin=0.0, violet=0.0, sing=0.0, friendship=0.0, lo
omi=0.0, basebal=0.0, stood=0.0, clark=0.0, bottl=0.0, marylin=0.0, si
n=0.0, juli=0.0, halfwai=0.0, spanish=0.0, hannah=0.0, shower=0.000481
695568400774, murphi=0.0, pal=0.000481695568400774, why=0.002408477842
0038703, leagu=0.0, our=0.000481695568400774, die=0.00048169556840077
4, glad=0.0, depart=0.000481695568400774, mightv=0.0, deborah=0.0, spe
nt=0.0, heather=0.0, grant=0.0, delici=0.0, bloom=0.0, halloween=0.0,
 selfish=0.0, mack=0.0, pleas=0.000481695568400774, later=0.0004816955
68400774, tomb=0.0, veri=0.000963391136801548, al=0.0, filthi=0.0, bei
n=0.0, cream=0.0, invit=0.0, ay=0.0, mountain=0.0, tax=0.0, liquid=0.
0, suffici=0.0, start=0.000963391136801548, particl=0.0, taught=0.0004
81695568400774, worthless=0.0, caitlin=0.0, paperwork=0.0, duffi=0.0,
 fade=0.0, lothar=0.0, courag=0.000481695568400774, puls=0.0, fighter=
0.000481695568400774, galleri=0.0, restaur=0.0, sky=0.0, hollywood=0.
0, accept=0.0, trip=0.0, grid=0.000481695568400774, vigo=0.0, cellar=
0.0, urg=0.0, recal=0.0, task=0.0, nose=0.0, circul=0.0, sperm=0.0, ex
peri=0.0, rust=0.0, thirtyseven=0.0, lame=0.0, editor=0.0, univers=0.
0, children=0.0, slug=0.0, ration=0.0, adam=0.0, mailbox=0.0, twentytw
o=0.0, reveal=0.0, eat=0.0, tobacco=0.0, schuyler=0.0, widow=0.0, forg
iv=0.0, punch=0.0, transfer=0.000481695568400774, humor=0.0, thrill=0.
0, jewish=0.0, elbow=0.0, teach=0.0, flush=0.0, pace=0.0, copi=0.0, lo
cat=0.0, senior=0.0, mantan=0.0, auto=0.0, chemistri=0.0, bolt=0.0, st
eadi=0.0, kick=0.0, ego=0.0, underneath=0.0, approv=0.0, cloud=0.0, st
udi=0.0, hunter=0.000481695568400774, clue=0.0, roof=0.0, french=0.0,
 billion=0.0, streak=0.0, chauncei=0.0, judi=0.0, cost=0.0, campbel=0.
0, contrari=0.0, bought=0.0, negro=0.0, seein=0.0, area=0.0, testifi=
0.0, remain=0.0, summer=0.0, dunbar=0.0, racket=0.0, no=0.005780346820
8092856, decis=0.0, gardin=0.0, miami=0.0, dark=0.000481695568400774,
 bird=0.0, reverend=0.0, loyal=0.0, clair=0.0, master=0.0, finger=0.0,
stink=0.0, expedit=0.0, skip=0.0, pardon=0.0, jone=0.0, fals=0.0, toe=
0.0, gal=0.0, might=0.0, approxim=0.0, norvil=0.0, sheriff=0.0, behav=
0.0, punk=0.0, uareu=0.0, inde=0.0, maximum=0.0, vernon=0.0, hide=0.00
0963391136801548, hampshir=0.0, put=0.001926782273603096, shh=0.0, pro
gram=0.0, plan=0.0, scratch=0.0, ban=0.0, adopt=0.0, ninth=0.0, gui=0.
001926782273603096, physic=0.0, enhanc=0.0, dead=0.000963391136801548,
bounc=0.0, perspect=0.0, column=0.0, refus=0.0, knife=0.0, dragon=0.0,
caesar=0.0, spacecraft=0.0, step=0.0, genet=0.0, bowler=0.0, driver=0.
000481695568400774, result=0.0, bishop=0.0, crime=0.0, ram=0.0, howr=
0.0, st=0.0, memori=0.0, dana=0.0, upper=0.0, front=0.0, twice=0.0, su
san=0.0, whoever=0.0, know=0.0057803468208092856, actual=0.0, war=0.00
```

```
2890173410404644, egg=0.0, johnson=0.0, island=0.0, pervert=0.0, muscl
=0.0, darryl=0.0, umm=0.0, fourth=0.0, fresh=0.0, recept=0.0, crack=0.
000481695568400774, vote=0.0, betcha=0.0, skywir=0.0, donovan=0.0, soc
ieti=0.0, uncomfort=0.0, commit=0.0, mai=0.000481695568400774, april=
0.0, purpos=0.0, stare=0.0, common=0.0, campu=0.0, mask=0.0, opera=0.
0, kennedi=0.0, bree=0.0, acm=0.0, dug=0.0, climb=0.0, chicago=0.0, ey
=0.0, pinta=0.0, lack=0.0, hung=0.0, camp=0.0, clinic=0.0, spat=0.0, u
thisu=0.0, sock=0.0, dawson=0.0, museum=0.0, vacuum=0.0, leap=0.0, rom
ulan=0.0, soviet=0.0, freewai=0.0, half=0.0, profit=0.0, iri=0.0, real
=0.0, client=0.0, an=0.000963391136801548, dynamit=0.0, eugen=0.0, nuk
e=0.0, robin=0.0, rub=0.0, gloria=0.0, becam=0.0, deceiv=0.0, distract
=0.0, matthew=0.0, ken=0.0, where=0.002890173410404644, america=0.0, b
ela=0.0, often=0.0, sox=0.0, dear=0.0, partner=0.0, devil=0.0, harm=0.
0, shotgun=0.0, adjust=0.0, scheme=0.0, let=0.003371868978805418, tras
h=0.0, sherri=0.0, script=0.0, join=0.0, clock=0.0, allison=0.0, confi
rm=0.0, clau=0.0, probat=0.0, rhyme=0.0, whale=0.0, nsa=0.0, gruner=0.
0, somedai=0.0, tripl=0.0, forward=0.0, pool=0.0, flat=0.0, sole=0.0,
 american=0.0, pentang=0.0, grown=0.000481695568400774, frederick=0.0,
perfectli=0.0, potenti=0.0, mccaffrei=0.0, form=0.0, nap=0.0, slide=0.
0, elev=0.0, thelma=0.0, fink=0.0, pumpkin=0.0, lilli=0.0, farewel=0.
0, swann=0.0, meal=0.0, fergu=0.0, mention=0.0, cant=0.004816955684007
741, pain=0.0014450867052023214, tri=0.0, ok=0.0, crisi=0.0, jose=0.0,
would=0.000481695568400774, beach=0.000481695568400774, size=0.0, ange
l=0.0, cancel=0.0, tuck=0.0, cook=0.0, began=0.0, pregnanc=0.0, king=
0.0, finish=0.0, automat=0.0, overwhelm=0.0, fast=0.0, intens=0.0, tra
nslat=0.0, sucker=0.0, etern=0.0, still=0.000963391136801548, beneath=
0.0, funer=0.0, immedi=0.0, prai=0.0, theatr=0.0, ceremoni=0.0, pregna
nt=0.0, capac=0.0, dya=0.0, killain=0.0, maroon=0.0, sammi=0.0, gordon
=0.0, bother=0.0, hah=0.0, cartman=0.0, crude=0.0, bedroom=0.0, mustan
g=0.0, offer=0.0, path=0.0, doyl=0.0, viktor=0.0, iim=0.0, buff=0.0, d
isk=0.0, birthdai=0.0, kilo=0.0, lap=0.0, am=0.000963391136801548, poe
m=0.0, least=0.000481695568400774, pick=0.0, pad=0.0, slowli=0.0004816
95568400774, garag=0.0, person=0.000481695568400774, permit=0.0, furth
er=0.0, connect=0.0, surgeon=0.0, freezer=0.0, obviou=0.0, dinosaur=0.
0, audrei=0.0, furnitur=0.0, scientist=0.0, roll=0.0, collect=0.0, cha
t=0.0, attend=0.0, document=0.0, chapter=0.0, death=0.0, venkman=0.0,
 fingernail=0.0, sweet=0.0, ol=0.0, stalk=0.0, salt=0.0, tobi=0.0, cam
era=0.0, entri=0.0, congratul=0.0, inspir=0.0, jackson=0.0, simul=0.0,
friedman=0.0, pale=0.0, kathryn=0.0, somebodi=0.0, splendid=0.0, eight
i=0.0, eras=0.0, digniti=0.0, swim=0.0, ly=0.0, star=0.0, ye=0.0024084
778420038703, shy=0.0, pin=0.0, outpost=0.0, rais=0.0, yah=0.0, leopar
d=0.0, exposur=0.0, puttin=0.0, martin=0.0, chip=0.0, somehow=0.0, jae
ger=0.0, silent=0.0, fact=0.0, ni=0.0, tub=0.0, bag=0.0004816955684007
74, initi=0.0, buri=0.0, council=0.0, andrew=0.0, brazil=0.0, neil=0.
0, suit=0.0, aliv=0.0, us=0.0038535645472061192, gabriel=0.0, expertis=
0.0, secret=0.0, cherri=0.0, crippl=0.0, cross=0.0, silver=0.0, thief=
0.0, whip=0.0, insan=0.0, riplei=0.0, foreign=0.0, ted=0.0, glimps=0.
0, lebowski=0.0, me=0.012524084778420123, peopl=0.000481695568400774,
 koessler=0.0, minist=0.0, kristen=0.0, faze=0.0, payment=0.0, dean=0.
0, regret=0.0, histor=0.0, slaughter=0.0, near=0.0, popcorn=0.00048169
5568400774, japanes=0.0, fuse=0.000481695568400774, lucki=0.0, mous=0.
0, wherer=0.0, powel=0.0, vanessa=0.0, inherit=0.0, meat=0.0, sang=0.
0, bert=0.0, scale=0.0, scam=0.0, servic=0.0, porno=0.0, spring=0.0, s
torag=0.0, small=0.0, thing=0.001926782273603096, tragedi=0.0, makin=
0.0, just=0.007225433526011609, costum=0.0, averag=0.0, experienc=0.0,
click=0.0, lose=0.0, folk=0.0, danger=0.000481695568400774, mayflow=0.
0, more=0.001926782273603096, view=0.000481695568400774, jasper=0.0, h
```

```
ostag=0.000481695568400774, swamp=0.0, cup=0.0, hatch=0.0, echo=0.0, c
indi=0.0, buddyboi=0.0, depend=0.0, rang=0.000481695568400774, harold=
0.0, instrument=0.0, deeper=0.0, beam=0.0, fish=0.0, writer=0.0, stir=
0.0, penelop=0.0, react=0.0, knive=0.0, mafia=0.0, carter=0.0, whod=0.
0, lincoln=0.0, swell=0.0, boat=0.0, certain=0.0, jet=0.0, steal=0.000
481695568400774, safer=0.0, larger=0.0, magazin=0.0, edgar=0.0, pittsb
urgh=0.0, committe=0.0, truth=0.0, green=0.0, over=0.0028901734104046
4, prevent=0.000481695568400774, impati=0.0, tv=0.0, ian=0.0, alarm=0.
0, boom=0.0, tripp=0.0, basement=0.0, torpedo=0.0, such=0.000481695568
400774, sir=0.0, whatr=0.0, jump=0.0, shari=0.0, kenni=0.0, oti=0.0, s
weat=0.000481695568400774, spine=0.0, everybodi=0.0, salon=0.0, asian=
0.0, miser=0.0, church=0.0, trust=0.000963391136801548, peac=0.0, surv
eil=0.0, understand=0.0024084778420038703, picnic=0.0, skipper=0.0, bo
di=0.000481695568400774, itself=0.0, mental=0.0, come=0.00289017341040
4644, shortli=0.0, bizarr=0.0, evolv=0.0, tea=0.0, known=0.0, unlik=0.
0, arni=0.0, whistler=0.0, firm=0.0, occur=0.0, grandpa=0.0, traitor=
0.0, that=0.01589595375722554, gosh=0.0, wynant=0.0, hopkin=0.0, cough
=0.0, butch=0.0, agn=0.0, wichita=0.0, tall=0.000481695568400774, sent
=0.000481695568400774, manner=0.0, artifici=0.0, dumper=0.0, palac=0.
0, track=0.0, twin=0.0, darl=0.0, stark=0.0, despit=0.0, alvi=0.0, pen
=0.0, wore=0.0, ani=0.000963391136801548, marshal=0.0, deaf=0.0, stret
ch=0.000481695568400774, chopper=0.0, coloni=0.0, shaft=0.0, pat=0.0,
 hal=0.0, allow=0.0, knee=0.0, ugli=0.0, combin=0.0, sheldrak=0.0, wal
l=0.0, complaint=0.0, understood=0.0, varieti=0.0, roger=0.0, page=0.
0, alphabet=0.0, men=0.0, palm=0.0, terror=0.0, li=0.0, skill=0.0, chi
ldhood=0.0, wilder=0.0, elliot=0.0, specialist=0.0, travel=0.0, howev=
0.0, exploit=0.0, quit=0.0, mph=0.0, traci=0.0, extrem=0.0, torranc=0.
0, fiction=0.0, measur=0.0, bug=0.0, lawrenc=0.0, pendergast=0.0, cb=
0.0, art=0.0, convinc=0.0, float=0.0, hop=0.0, armi=0.0, socal=0.0, sa
ilor=0.0, circl=0.0, pursu=0.0, notifi=0.0, creasi=0.0, sum=0.0, restr
ict=0.0, shield=0.0, rocco=0.0, asleep=0.0, failur=0.0, memor=0.000481
695568400774, mon=0.0, cool=0.0, emma=0.0, shape=0.0, state=0.0, disco
unt=0.0, youll=0.001926782273603096, repeat=0.000481695568400774, neit
her=0.0, chart=0.0, soul=0.0, dose=0.0, kai=0.0, wilson=0.0, valet=0.
0, tore=0.0, sacr=0.0, rorschach=0.0, custom=0.0, imit=0.0, rekal=0.0,
railroad=0.0, monsieur=0.0, read=0.0, lighter=0.0, utterli=0.0, loss=
0.0, appropri=0.0, veget=0.0, oklahoma=0.0, oliv=0.0, habit=0.0, slick
=0.0, intuit=0.0, rod=0.0, dedic=0.0, propos=0.0, corridor=0.0, nut=0.
0, auction=0.0, flu=0.0, bottom=0.0, patron=0.0, uwhyu=0.0, richi=0.0,
link=0.0, signal=0.0, custodi=0.0, chanc=0.000481695568400774, sayer=
0.0, skin=0.000963391136801548, mortal=0.0, add=0.0, honeymoon=0.0, re
ceipt=0.0, nun=0.0, usual=0.0, starv=0.000481695568400774, eight=0.0,
 morgu=0.0, comin=0.0, style=0.0, cadet=0.0, uh=0.0, uhuh=0.0, earn=0.
0, tini=0.0, volunt=0.000963391136801548)
```

For example, the fastest way to find the frequency of "none" in the movie *The Terminator* is to access the `'none'` item from its row. Check the original table to see if this worked for you!

```
In [31]: row_for_title('the terminator').item('none')
```

```
Out[31]: 0.000963391136801548
```

Question 1.0

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

```
BEGIN QUESTION
name: q1_0
```

In [32]:

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

In [33]:

```
# TEST
0 < expected_row_sum
```

Out[33]:

```
True
```

In [34]:

```
# HIDDEN TEST
int(expected_row_sum) == 1
```

Out[34]:

```
True
```

This dataset was extracted from a dataset from Cornell University (http://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html). 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 [35]:

```
print('Words with frequencies:', movies.drop(np.arange(5)).num_columns)
print('Movies with genres:', movies.num_rows)
```

```
Words with frequencies: 5000
Movies with genres: 370
```

1.1. Word Stemming

The columns other than "Title", "Year", "Rating", "Genre", 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 (https://en.wikipedia.org/wiki/Inflection) 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 [36]:

```
# 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[36]:

Stem	Word
bond	bonding
bone	bone
bone	boning
bone	bones
bonu	bonus
book	bookings
book	books
book	booking
book	booked
book	book

Question 1.1.1

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

```
BEGIN QUESTION
name: q1_1_1
```

In [37]:

```
stemmed_message = vocab_table.where('Word', are.equal_to('vegetables'
)).column('Stem').item(0) # SOLUTION
stemmed_message
```

Out[37]:

```
'veget'
```

In [38]:

```
# TEST
len(stemmed_message) < len('message')
```

Out[38]:

```
True
```

In [39]:

```
# HIDDEN TEST
stemmed_message
```

Out[39]:

```
'veget'
```

Question 1.1.2

What stem in the dataset has the most words that are shortened to it? Assign `most_stem` to that stem.

```
BEGIN QUESTION
name: q1_1_2
```

In [40]:

```
most_stem = vocab_table.group("Stem").sort("count",descending=True).column("Stem").item(0) # SOLUTION
most_stem
```

Out[40]:

```
'gener'
```

In [41]:

```
# TEST
type(most_stem) == str
```

Out[41]:

```
True
```

In [42]:

```
# HIDDEN TEST
most_stem
```

Out[42]:

```
'gener'
```

Question 1.1.3

What is the longest word in the dataset whose stem wasn't shortened? Assign that to `longest_uncut` . Break ties alphabetically from Z to A (so if your options are "albatross" or "batman", you should pick "batman").

```
BEGIN QUESTION
name: q1_1_3
```

In [43]:

```
# In our solution, we found it useful to first add columns with
# the length of the word and the length of the stem,
# and then to add a column with the difference between those lengths.
# What will the difference be if the word is not shortened?

tbl_with_lens = vocab_table.with_columns("Word len", vocab_table.apply(len, "Word"), "Stem len", vocab_table.apply(len, "Stem")) # SOLUTION
tbl_with_dif = tbl_with_lens.with_column("len dif", tbl_with_lens.column("Word len") - tbl_with_lens.column("Stem len")) # SOLUTION


longest_uncut = tbl_with_dif.where("len dif", 0).sort("Word len", descending=True).column("Word").item(0) # SOLUTION
longest_uncut
```

Out[43]:

```
'misunderstand'
```

```
In [44]: # TEST
         type(longest_uncut) == str

Out[44]: True
```

```
In [45]: # HIDDEN TEST
         longest_uncut == 'misunderstand'

Out[45]: True
```

1.2. Exploratory Data Analysis: Linear Regression

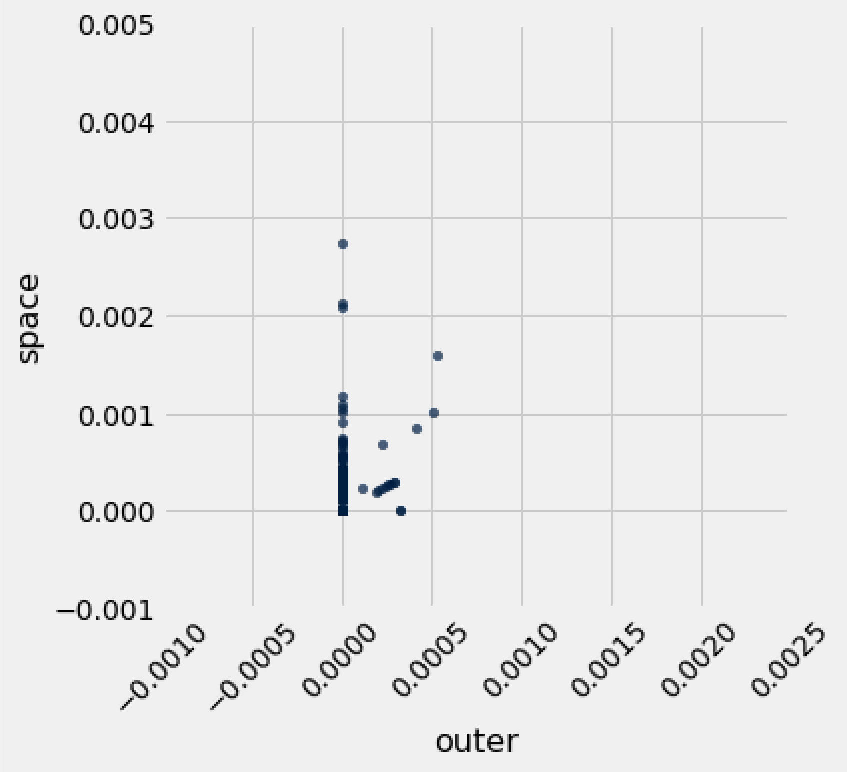
Let's explore our dataset before trying to build a classifier. To start, we'll look at the relationship between words in proportions.

The first association we'll investigate is the association between the proportion of words that are "outer" and the proportion of words that are "space".

As usual, we'll investigate our data visually before performing any numerical analysis.

Run the cell below to plot a scatter diagram of space proportions vs outer proportions and to create the `outer_space` table.

```
In [46]: # Just run this cell!
         outer_space = movies.select("outer", "space")
         outer_space.scatter("outer", "space")
         plots.axis([-0.001, 0.0025, -0.001, 0.005]);
         plots.xticks(rotation=45);
```



Question 1.2.1

Looking at that chart it is difficult to see if there is an association. Calculate the correlation coefficient for the association between proportion of words that are "outer" and the proportion of words that are "space" for every movie in the dataset, and assign it to `outer_space_r` .

```
BEGIN QUESTION
name: q1_2_1
```

```
In [47]: # Our solution took multiple lines
         # these two arrays should make your code cleaner!
         outer = movies.column("outer")
         space = movies.column("space")

         outer_su = (outer - np.mean(outer)) / np.std(outer) # SOLUTION
         space_su = (space - np.mean(space)) / np.std(space) # SOLUTION

         outer_space_r = np.mean(space_su * outer_su) # SOLUTION
         outer_space_r
```

```
Out[47]: 0.2829527833012742
```

In [48]:

```
# TEST
0.15 < outer_space_r < 0.3
```

Out[48]:

```
True
```

In [49]:

```
# HIDDEN TEST
np.round(outer_space_r, 3) == .283
```

Out[49]:

```
True
```

Question 1.2.2

Choose two *different* words in the dataset with a correlation higher than 0.2 or smaller than -0.2 that are not *outer* and *space* and plot a scatter plot with a line of best fit for them. The code to plot the scatter plot and line of best fit is given for you, you just need to calculate the correct values to `r` , `slope` and `intercept` .

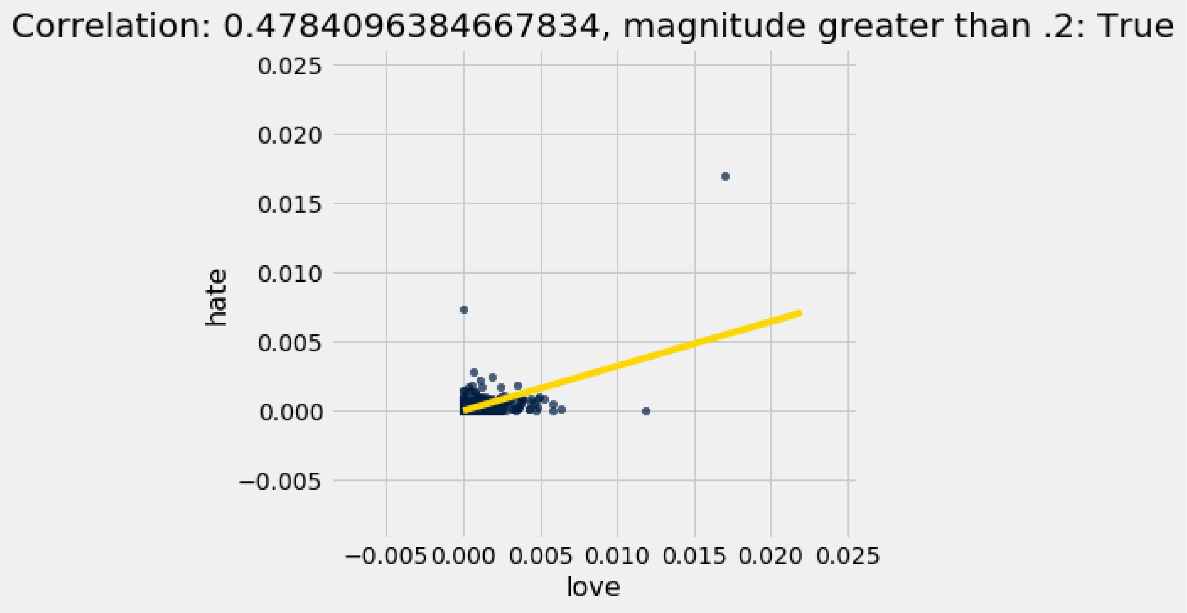
Hint: It's easier to think of words with a positive correlation, i.e. words that are often mentioned together.

Hint 2: Try to think of common phrases or idioms.

```
BEGIN QUESTION
name: q1_2_2
manual: true
image: true
```

In [50]:

```
word_x = "love" # SOLUTION
word_y = "hate" # SOLUTION

# These arrays should make your code cleaner!
arr_x = movies.column(word_x)
arr_y = movies.column(word_y)

x_su = (arr_x - np.mean(arr_x)) / np.std(arr_x) # SOLUTION
y_su = (arr_y - np.mean(arr_y)) / np.std(arr_y) # SOLUTION

r = np.mean(x_su * y_su) # SOLUTION

slope = np.std(arr_y) / np.std(arr_x) * r # SOLUTION
intercept = np.mean(arr_y) - (np.mean(arr_x) * slope) # SOLUTION

# DON'T CHANGE THESE LINES OF CODE
movies.scatter(word_x, word_y)
max_x = max(movies.column(word_x))
plots.title(f"Correlation: {r}, magnitude greater than .2: {abs(r) >=
 0.2}")
plots.plot([0, max_x * 1.3], [intercept, intercept + slope * (max_x*1.
3)], color='gold');
```



1.3. 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 [51]:

```
# 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:  314 ; Test:  56
```

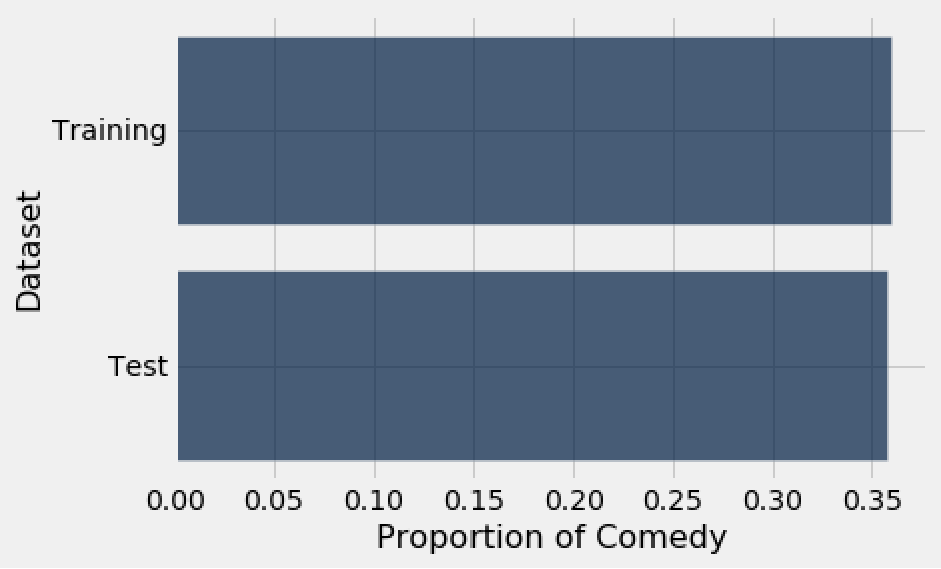
Question 1.3.1

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

```
BEGIN QUESTION
name: q1_3_1
manual: true
image: true
```

In [52]:

```
# BEGIN SOLUTION NO PROMPT
def comedy_proportion(table):
    """Return the proportion of movies in a table that have the Comedy genre."""
    return table.where('Genre', are.equal_to('comedy')).num_rows / table.num_rows

Table().with_columns(
        'Dataset', make_array('Training', 'Test'),
        'Proportion of Comedy', make_array(comedy_proportion(train_movies), comedy_proportion(test_movies)))\
       .barh('Dataset')
# END SOLUTION
""" # BEGIN PROMPT
def comedy_proportion(table):
    # Return the proportion of movies in a table that have the Comedy genre.
    return ...

# 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
..."""; # END PROMPT
```



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: comedy and thriller. 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 "water" and "feel" in the movie. Taking the movie *Monty Python and the Holy Grail* (in the test set), 0.000804074 of its words are "water" and 0.0010721 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 $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

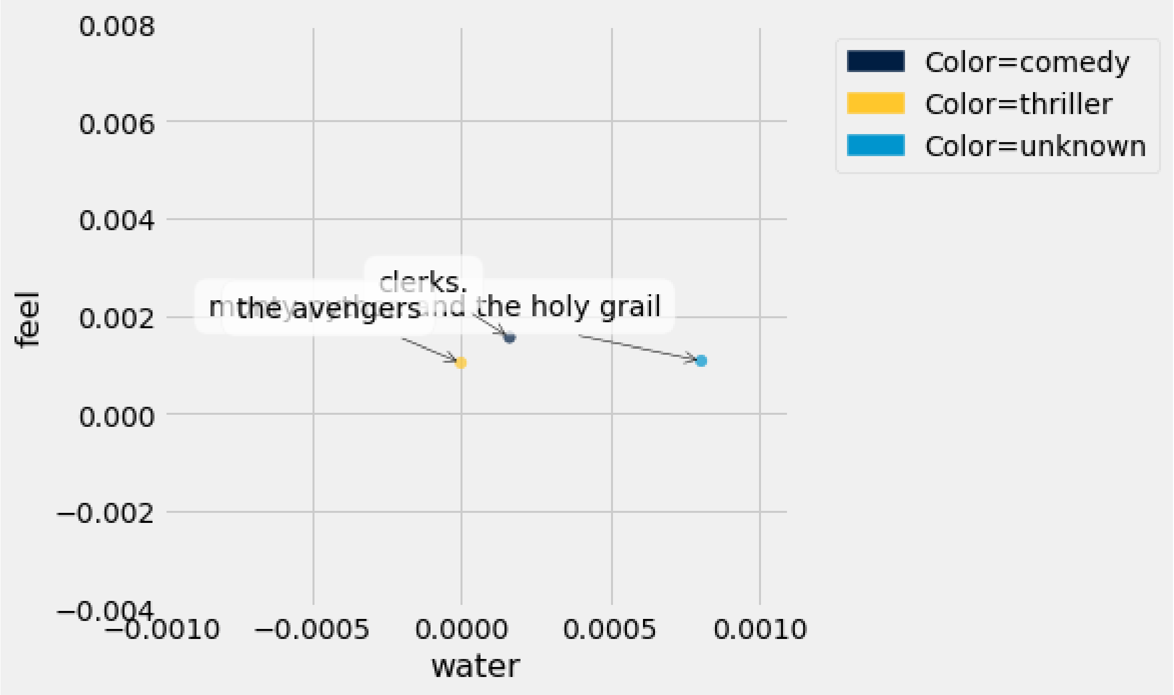
For example, in the movie *Clerks.* (in the training set), 0.00016293 of all the words in the movie are "water" and 0.00154786 are "feel". Its distance from *Monty Python and the Holy Grail* on this 2-word feature set is $\sqrt{(0.000804074 - 0.000162933)^2 + (0.0010721 - 0.00154786)^2} \approx 0.000798379$. (If we included more or different features, the distance could be different.)

A third movie, *The Avengers* (in the training set), is 0 "water" and 0.00103173 "feel".

The function below creates a plot to display the "water" and "feel" features of a test movie and some training movies. As you can see in the result, *Monty Python and the Holy Grail* is more similar to "Clerks." than to the *The Avengers* based on these features, which is makes sense as both movies are comedy movies, while *The Avengers* is a thriller.

In [53]:

```
# 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 = ["clerks.", "the avengers"]
plot_with_two_features("monty python and the holy grail", training, "water", "feel")
plots.axis([-0.001, 0.0011, -0.004, 0.008]);
```



Question 2.1.1

Compute the Euclidean distance (defined in the section above) between the two movies, *Monty Python and the Holy Grail* and *The Avengers*, using the `water` and `feel` features only. Assign it the name `one_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.

```
BEGIN QUESTION
name: q2_1_1
```

In [54]:

```
title_index = movies.index_by('Title')
python = row_for_title("monty python and the holy grail")
avengers = row_for_title("the avengers")

one_distance = ((python.item("water") - avengers.item("water"))**2 + (
python.item("feel") - avengers.item("feel"))**2)**0.5 # SOLUTION
one_distance
```

Out[54]:

```
0.0008050869157478908
```

In [55]:

```
# TEST
0 < one_distance < .01
```

Out[55]:

```
True
```

In [56]:

```
# HIDDEN TEST
np.round(one_distance, 6)
```

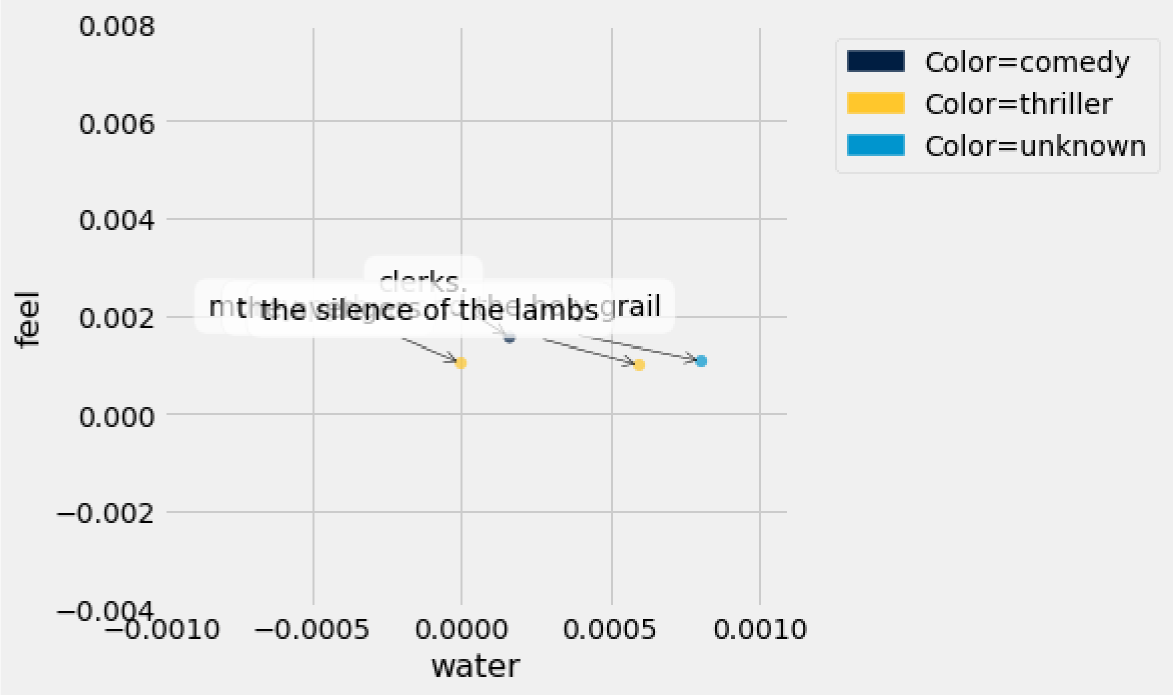
Out[56]:

```
0.000805
```

Below, we've added a third training movie, *The Silence of the Lambs*. Before, the point closest to *Monty Python and the Holy Grail* was *Clerks.*, a comedy movie. However, now the closest point is *The Silence of the Lambs*, a thriller movie.

In [57]:

```
training = ["clerks.", "the avengers", "the silence of the lambs"]
plot_with_two_features("monty python and the holy grail", training, "water", "feel")
plots.axis([-0.001, 0.0011, -0.004, 0.008]);
```



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 *Monty Python and the Holy Grail* is closer to *The Silence of the Lambs* than it is to *Clerks*.

```
BEGIN QUESTION
name: q2_1_2
```

In [58]:

```
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) # SOLUTION
    row1 = row_for_title(title1) # SOLUTION
    return ((row0.item(x_feature) - row1.item(x_feature))**2 + (row0.item(y_feature) - row1.item(y_feature))**2)**0.5 # SOLUTION

for movie in make_array("clerks.", "the silence of the lambs"):
    movie_distance = distance_two_features(movie, "monty python and the holy grail", "water", "feel")
    print(movie, 'distance:\t', movie_distance)
```

```
clerks. distance:        0.0007983810687227951
the silence of the lambs distance:       0.00022256314855568916
```

In [59]:

```
# TEST
# Make sure you can use any two movies
correct_dis = 0.000541242
dis = distance_two_features("clerks.", "the avengers", "water", "feel")
np.isclose(np.round(dis, 9), correct_dis)
```

Out[59]:

```
True
```

In [60]:

```
# TEST
# Make sure you can use any two features
correct_dis = 0.006486728
dis = distance_two_features("clerks.", "the avengers", "your", "that")
np.isclose(np.round(dis, 9), correct_dis)
```

Out[60]:

```
True
```

Question 2.1.3

Define the function `distance_from_python` 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!

```
BEGIN QUESTION
name: q2_1_3
```

In [61]:
```
def distance_from_python(title):
    """The distance between the given movie and "monty python and the
 holy grail",
    based on the features "water" and "feel".

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

    return distance_two_features(title, "monty python and the holy gra
il", "water", "feel") # SOLUTION
```

In [62]:
```
# TEST
np.isclose(distance_from_python('clerks.'), 0.00079838)
```

Out[62]:
```
True
```

In [63]:
```
# HIDDEN TEST
np.isclose(distance_from_python('the avengers'), 0.0008050869)
```

Out[63]:
```
True
```

Question 2.1.4

Using the features `"water"` and `"feel"`, what are the names and genres of the 5 movies in the **training set** closest to *Monty Python and the Holy Grail*? To answer this question, make a table named `close_movies` containing those 5 movies with columns `"Title"`, `"Genre"`, `"water"`, and `"feel"`, as well as a column called `"distance from python"` that contains the distance from *Monty Python and the Holy Grail*. The table should be **sorted in ascending order by `distance from python`**.

```
BEGIN QUESTION
name: q2_1_4
```

In [67]:
```
# BEGIN SOLUTION NO PROMPT
water_feel = train_movies.select("Title", "Genre", "water", "feel")
close_movies = water_feel.with_column('distance from python', water_fe
el.apply(distance_from_python, 'Title'))\
                       .sort('distance from python')\
                       .take(np.arange(5))
# END SOLUTION

""" # BEGIN PROMPT
# The staff solution took multiple lines.
...
close_movies = ...
close_movies
"""; # END PROMPT
```

In [68]:

```
# TEST
set(close_movies.labels) >= {'Genre', 'Title', 'feel', 'water'}
```

Out[68]:

```
True
```

In [69]:

```
# TEST
close_movies.num_rows == 5
```

Out[69]:

```
True
```

In [70]:

```
# TEST
close_movies.column("Title").item(0) != "monty python and the holy gra
il"
```

Out[70]:

```
True
```

In [73]:

```
# HIDDEN TEST
[title[:6] for title in close_movies.column('Title')] == ['alien', 'to
morr', 'the si', 'inners', 'some l']
```

Out[73]:

```
True
```

Question 2.1.5

Next, we'll clasify *Monty Python and the Holy Grail* 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.

```
BEGIN QUESTION
name: q2_1_5
```

In [74]:

```
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.group(label).sort('count', descending=True).column(la
bel).item(0) # SOLUTION

# Calling most_common on your table of 5 nearest neighbors classifies
# "monty python and the holy grail" as a thriller movie, 3 votes to 2.
most_common('Genre', close_movies)
```

Out[74]:

```
'thriller'
```

```
In [75]: # TEST
         [most_common('Genre', close_movies.take(range(k))) for k in range(1, 5
         , 1)]

Out[75]: ['thriller', 'thriller', 'thriller', 'thriller']
```

```
In [76]: # TEST
         [most_common('Genre', close_movies.take(np.arange(4, k, -1))) for k in
         range(3, -1, -1)]

Out[76]: ['comedy', 'comedy', 'comedy', 'thriller']
```

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 *Monty Python and the Holy Grail* as a thriller movie (unless you count the thrilling holy hand grenade scene). 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 [77]:
```
_ = ok.submit()
```

```
Saving notebook...

---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call
 last)
<ipython-input-77-cc46ca874451> in <module>
----> 1 _ = ok.submit()

~/sp20/teaching/teaching_env/lib/python3.7/site-packages/client/api/no
tebook.py in submit(self)
     69         messages = {}
     70         self.assignment.set_args(submit=True)
---> 71         if self.save(messages):
     72             return self.run('backup', messages)
     73         else:

~/sp20/teaching/teaching_env/lib/python3.7/site-packages/client/api/no
tebook.py in save(self, messages, delay, attempts)
     78
     79     def save(self, messages, delay=0.5, attempts=3):
---> 80         saved = self.save_notebook()
     81         if not saved:
     82             return None

~/sp20/teaching/teaching_env/lib/python3.7/site-packages/client/api/no
tebook.py in save_notebook(self)
    115         # Wait for first .ipynb to save
    116         if ipynbs:
--> 117             if wait_for_save(ipynbs[0]):
    118                 print("Saved '{}'.".format(ipynbs[0]))
    119             else:

~/sp20/teaching/teaching_env/lib/python3.7/site-packages/client/api/no
tebook.py in wait_for_save(filename, timeout)
    160     Returns True if a save was detected, and False otherwise.
    161     """
--> 162     modification_time = os.path.getmtime(filename)
    163     start_time = time.time()
    164     while time.time() < start_time + timeout:

~/anaconda3/lib/python3.7/genericpath.py in getmtime(filename)
     53 def getmtime(filename):
     54     """Return the last modification time of a file, reported b
y os.stat()."""
---> 55     return os.stat(filename).st_mtime
     56
     57

FileNotFoundError: [Errno 2] No such file or directory: 'movies.ipynb'
```

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 3.0

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.

Note: If you're working offline: Depending on the versions of your packages, you may need to convert rows to arrays using the following instead: `np.array(list(t.row(0))`

```
BEGIN QUESTION
name: q3_0
```

In [78]:

```
def distance(features_array1, features_array2):
    """The Euclidean distance between two arrays of feature values."""
    return sum((features_array1 - features_array2)**2) ** 0.5 # SOLUTI
ON

distance_first_to_second = distance(np.array(list(train_movies.drop(np
.arange(5)).row(0))), np.array(list(train_movies.drop(np.arange(5)).ro
w(1)))) # SOLUTION
distance_first_to_second
```

Out[78]:

```
0.03335446890881372
```

In [79]:

```
# TEST
0.0 <= distance_first_to_second <= 0.1
```

Out[79]:

```
True
```

In [80]:
```
# TEST
np.isclose(distance(make_array(1, 2), make_array(1, 2)), 0)
```

Out[80]:
```
True
```

In [81]:
```
# TEST
np.isclose(distance(make_array(1, 2, 3), make_array(2, 4, 5)), 3)
```

Out[81]:
```
True
```

In [83]:
```
# HIDDEN TEST
np.isclose(round(distance_first_to_second, 5), 0.03335)
```

Out[83]:
```
True
```

In [84]:
```
# HIDDEN TEST
a1 = np.array([1, 2, 3])
a2 = np.array([3, 4, 5])
np.isclose(round(distance(a1, a2), 4), 3.4641)
```

Out[84]:
```
True
```

In [85]:
```
# HIDDEN TEST
a2 = np.array([3, 4, 5])
a3 = np.array([9, 5, 4])
np.isclose(round(distance(a2, a3), 4), 6.1644)
```

Out[85]:
```
True
```

In [86]:
```
# HIDDEN TEST
a1 = np.array([1, 2, 3])
a3 = np.array([9, 5, 4])
np.isclose(round(distance(a1, a3), 4), 8.6023)
```

Out[86]:
```
True
```

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 3.1.1

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

Note: The line graphed is the line of best fit, NOT a y=x

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 common in both comedy and thriller movies
2. The word is uncommon in comedy movies and common in thriller movies
3. The word is common in comedy movies and uncommon in thriller movies
4. The word is uncommon in both comedy and thriller 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.

```
BEGIN QUESTION
name: q3_1_1
```

In [79]:
```
bottom_left = 4 # SOLUTION
```

In [80]:
```
# TEST
# It looks like you've choosen an illegal option (not within 1-5)
bottom_left >= 1 and bottom_left <= 5
```

Out[80]:
```
True
```

In [81]:
```
# HIDDEN TEST
bottom_left == 4
```

Out[81]:
```
True
```

Question 3.1.2

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

```
BEGIN QUESTION
name: q3_1_2
```

```
In [82]: bottom_right = 3 # SOLUTION
```

```
In [83]: # TEST
         # It looks like you've choosen an illegal option (not within 1-5)
         bottom_right >= 1 and bottom_right <= 5

Out[83]: True
```

```
In [84]: # HIDDEN TEST
         bottom_right == 3

Out[84]: True
```

Question 3.1.3

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

```
BEGIN QUESTION
name: q3_1_3
```

```
In [85]: top_right = 1 # SOLUTION
```

```
In [86]: # TEST
         # It looks like you've choosen an illegal option (not within 1-5)
         top_right >= 1 and top_right <= 5

Out[86]: True
```

```
In [87]: # HIDDEN TEST
         top_right == 1

Out[87]: True
```

Question 3.1.4

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

```
BEGIN QUESTION
name: q3_1_4
```

In [88]:
```
top_left = 2 # SOLUTION
```

In [89]:
```
# TEST
# It looks like you've choosen an illegal option (not within 1-5)
top_left >= 1 and top_left <= 5
```

Out[89]:
```
True
```

In [90]:
```
# HIDDEN TEST
top_left == 2
```

Out[90]:
```
True
```

Question 3.1.5

If we see a movie with a lot of words that are common for comedy movies but uncommon for thriller 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 a thriller movie.
2. It is a comedy movie.
```

```
BEGIN QUESTION
name: q3_1_5
```

In [91]:
```
movie_genre_guess = 2 # SOLUTION
```

In [92]:
```
# TEST
movie_genre_guess >= 1 and movie_genre_guess <= 2
```

Out[92]:
```
True
```

In [93]:
```
# HIDDEN TEST
movie_genre_guess == 2
```

Out[93]:
```
True
```

Question 3.1.6

Using the plot above, make an array of at least 10 common words that you think might let you distinguish between comedy and thriller movies. Make sure to choose words that are frequent enough that every movie contains at least one of them. Don't just choose the most frequent words, 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.

```
BEGIN QUESTION
name: q3_1_6
```

In [94]:
```
# Set my_20_features to an array of 20 features (strings that are column labels)

my_features = make_array('i', 'the', 'to', 'a', 'it', 'and', 'that', 'of', 'your', 'what', 'in', 'me', 'is', 'do', 'thi', 'dont', 'he', 'for', 'know') # SOLUTION

# Select the 20 features of interest from both the train and test sets
train_my_features = train_movies.select(my_features)
test_my_features = test_movies.select(my_features)
```

In [95]:
```
# TEST
len(my_features) >= 10
```

Out[95]:
```
True
```

In [96]:
```
# TEST
np.all([f in test_movies.labels for f in my_features])
```

Out[96]:
```
True
```

In [97]:
```
# TEST
# It looks like there are many movies in the training set that
# don't have any of your chosen words.  That will make your
# classifier perform very poorly in some cases.  Try choosing
# at least 1 common word.
train_f = train_movies.select(my_features)
np.count_nonzero(train_f.apply(lambda r: np.sum(np.abs(np.array(list(r)))) == 0)) < len(my_features)
```

Out[97]:
```
True
```

In [98]:

```
# TEST
# It looks like there are many movies in the test set that
# don't have any of your chosen words.  That will make your
# classifier perform very poorly in some cases.  Try choosing
# at least 1 common word.
test_f = test_movies.select(my_features)
np.count_nonzero(test_f.apply(lambda r: np.sum(np.abs(np.array(list(r
)))) == 0)) < 5
```

Out[98]:

```
True
```

In [99]:

```
# TEST
# It looks like you may have duplicate words! Make sure not to!
len(set(my_features)) >= 10
```

Out[99]:

```
True
```

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 3.1.7

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

```
BEGIN QUESTION
name: q3_1_7
manual: True
```

SOLUTION: The staff features don't work very well. It's a good idea to pick words that are common in thriller movies but uncommon in comedy movies and vice-versa. These are points that are far away from the diagonal in the above plot.

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 [100]:

```
print("Movie:")
test_movies.take(0).select('Title', 'Genre').show()
print("Features:")
test_my_features.take(0).show()
```

```
Movie:
```

Title	Genre
new nightmare	thriller

```
Features:
```

i	the	to	a	it	and	that	of	y
0.0222384	0.0317171	0.0185928	0.0211447	0.0273423	0.0156763	0.0120306	0.0102078	0.00874

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 `my_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 [101]:

```
# 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_my_features.row(0)).
      train_table: A table of features (for example, the whole
        table train_my_features)."""
    assert train_table.num_columns < 50, "Make sure you're not using a
ll the features of the movies table."
    counts_matrix = np.asmatrix(train_table.columns).transpose()
    diff = np.tile(np.array(list(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 3.1.8

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 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 ascending order by distance to the first test movie**.

```
BEGIN QUESTION
name: q3_1_8
```

In [102]:

```
# BEGIN SOLUTION NO PROMPT
distances = fast_distances(test_my_features.row(0), train_my_features)

genre_and_distances = Table().with_columns(
                                'Genre',    train_movies.column('Genre'
),
                                'Distance', distances)\
                             .sort('Distance')
genre_and_distances.show(4)
# END SOLUTION
""" # BEGIN PROMPT
# The staff solution took multiple lines of code.
genre_and_distances = ...
genre_and_distances
"""; # END PROMPT
```

Genre	Distance
thriller	0.0155651
thriller	0.0161311
thriller	0.0170161
thriller	0.0170332

... (310 rows omitted)

In [103]:

```
# TEST
genre_and_distances.labels == ('Genre', 'Distance')
```

Out[103]:

```
True
```

In [104]:

```
# TEST
genre_and_distances.num_rows == train_movies.num_rows
```

Out[104]:

```
True
```

In [105]:

```
# TEST
print(genre_and_distances.group('Genre'))
```

```
Genre    | count
comedy   | 113
thriller | 201
```

In [106]:

```
# TEST
np.allclose(genre_and_distances.column('Distance'), sorted(fast_distan
ces(test_my_features.row(0), train_my_features)))
```

Out[106]:

```
True
```

Question 3.1.9

Now compute the 7-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 7 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.)

```
BEGIN QUESTION
name: q3_1_9
```

In [107]:

```
# Set my_assigned_genre to the most common genre among these.
my_assigned_genre = most_common("Genre", genre_and_distances.take(np.a
range(7))) # SOLUTION

# 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 = my_assigned_genre == test_movies.colum
n("Genre").item(0) # SOLUTION

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

```
The assigned genre, thriller, was correct.
```

In [108]:

```
# TEST
genre_and_distances.take(np.arange(7)).group('Genre').index_by('Genre'
)[my_assigned_genre][0].item('count') >= 4
```

Out[108]:

```
True
```

In [109]:

```
# TEST
my_assigned_genre_was_correct == (my_assigned_genre == 'thriller')
```

Out[109]:

```
True
```

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_my_features.row(0)`).
- A table with a column for each feature (e.g., `train_my_features`).
- An array of classes (e.g. the labels "comedy" or "thriller") 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 `'comedy'` or the string `'thriller'`).

```
BEGIN QUESTION
name: q3_2_1
```

In [110]:

```
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_columns('Genre', train_labels, 'Distance', distances).sort('Distance') # SOLUTION
    return most_common("Genre", genre_and_distances.take(np.arange(k))) # SOLUTION
```

In [111]:

```
# TEST
# This test just checks to see if your classify function works correctly
# with k=5 nearest neighbors.
from collections import Counter
g = train_movies.column('Genre')
def check(r, k):
    t = test_my_features.row(r)
    return classify(t, train_my_features, g, k) == Counter(np.take(g, np.argsort(fast_distances(t, train_my_features))[:k])).most_common(1)[0][0]
check_5_nn = [check(i, 5) for i in np.arange(11)]
all(check_5_nn)
```

Out[111]:

```
True
```

In [112]:

```
# TEST
# This test just checks to see if your classify function works correctly
# with k=11 nearest neighbors.
from collections import Counter
g = train_movies.column('Genre')
def check(r, k):
    t = test_my_features.row(r)
    return classify(t, train_my_features, g, k) == Counter(np.take(g, np.argsort(fast_distances(t, train_my_features))[:k])).most_common(1)[0][0]
check_11_nn = [check(i, 11) for i in np.arange(11)]
all(check_11_nn)
```

Out[112]:

```
True
```

Question 3.2.2

Assign `tron_genre` to the genre predicted by your classifier for the movie "tron" in the test set, using **13 neighbors** and using your 20 features.

```
BEGIN QUESTION
name: q3_2_2
```

In [113]:

```
# The staff solution first defined a row called king_kong_features.
tron_features = test_movies.where("Title", "tron").select(my_features).row(0) # SOLUTION
tron_genre = classify(tron_features, train_my_features, train_movies.column("Genre"), 13) # SOLUTION
tron_genre
```

Out[113]:

```
'thriller'
```

In [114]:

```
# TEST
from collections import Counter
g = train_movies.column('Genre')
r = np.where(test_movies['Title'] == "tron")[0][0]
t = test_my_features.row(r)
tron_expected_genre = Counter(np.take(g, np.argsort(fast_distances(t, train_my_features))[:13])).most_common(1)[0][0]
tron_genre == tron_expected_genre
```

Out[114]:

```
True
```

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 13-nearest neighbors algorithm with `train_20` as its training set.

```
BEGIN QUESTION
name: q3_2_3
```

In [115]:

```
def classify_feature_row(row):
    return classify(row, train_my_features, train_movies.column('Genre'), 13) # SOLUTION

# When you're done, this should produce 'Thriller' or 'Comedy'.
classify_feature_row(test_my_features.row(0))
```

Out[115]:

```
'thriller'
```

In [116]:

```
# TEST
# This test just checks that your classify_feature_row works correctly.
def check(r):
    t = test_my_features.row(r)
    return classify(t, train_my_features, train_movies.column('Genre'), 13) == classify_feature_row(t)
all([check(i) for i in np.arange(13)])
```

Out[116]:

```
True
```

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.

```
BEGIN QUESTION
name: q3_3_1
```

In [117]:

```
test_guesses = test_my_features.apply(classify_feature_row) # SOLUTION
proportion_correct = np.count_nonzero(test_guesses == test_movies.column('Genre')) / test_movies.num_rows # SOLUTION
proportion_correct
```

Out[117]:

```
0.5892857142857143
```

In [118]:
```
# TEST
0 <= proportion_correct <= 1
```

Out[118]:
```
True
```

In [119]:
```
# HIDDEN TEST
r = np.count_nonzero(test_guesses == test_movies.column('Genre')) / te
st_movies.num_rows
proportion_correct == r
```

Out[119]:
```
True
```

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.

```
BEGIN QUESTION
name: q3_3_2
```

In [120]:
```
# 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.with_column('Was correct', test_g
uesses == test_movies.column('Genre')).select('Title', 'Genre', 'Was c
orrect') # SOLUTION
test_movie_correctness.sort('Was correct', descending = True).show(5)
```

Title	Genre	Was correct
happy birthday wanda june	comedy	True
sphere	thriller	True
suspect zero	thriller	True
star trek: the wrath of khan	thriller	True
jackie brown	thriller	True

... (51 rows omitted)

In [121]:
```
# TEST
test_movie_correctness.labels == ('Title', 'Genre', 'Was correct')
```

Out[121]:
```
True
```

In [122]:
```
# TEST
test_movie_correctness.num_rows == test_movies.num_rows
```

Out[122]:
```
True
```

In [123]:
```
# TEST
# Make sure that test_movie_correctness does not modify the original
# test_movies table.
print(test_movie_correctness.group('Genre'))
```

```
Genre    | count
comedy   | 20
thriller | 36
```

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.

```
BEGIN QUESTION
name: q3_3_3
manual: true
```

SOLUTION: The classifier tends to misclassify movies that have both comedy and thriller elements in them.

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 [124]:
```python
# 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("laugh", "marri", "dead", "heart", "cop")

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

def another_classifier(row):
    return ...
```

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.

```
BEGIN QUESTION
name: q4_2
manual: true
```

SOLUTION: Any reasonable student solution is fine.

Question 4.3

Briefly describe what you tried to improve your classifier.

```
BEGIN QUESTION
name: q4_3
manual: true
```

SOLUTION: You were meant to put in at least some minimal effort.

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

In [ ]:
```python
# For your convenience, you can run this cell to run all the tests at once!
import os
print("Running all tests...")
_ = [ok.grade(q[:-3]) for q in os.listdir("tests") if q.startswith('q')]
print("Finished running all tests.")
```

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

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 (https://medium.com/@sifium/machine-learning-types-of-classification-9497bd4f2e14)
- Train/test split and cross-validation (https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6)
- More information about k-nearest neighbors (https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7)
- Overfitting (https://elitedatascience.com/overfitting-in-machine-learning)

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 (http://scikit-learn.org/stable/tutorial/basic/tutorial.html)
- TensorFlow information (https://www.tensorflow.org/tutorials/)

...and many more!

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

```
In [ ]:
```