In [1]:
```
# Initialize OK
from client.api.notebook import Notebook
ok = Notebook('project3.ok')
```

```
=====================================================================
Assignment: Project 3: Movie Classification
OK, version v1.18.1
=====================================================================
```

In [2]:
```
import otter
```

Project 3: Movie Classification

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

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

Logistics

Deadline. This project is due at 11:59pm on Friday 5/01. You can earn an early submission bonus point by submitting your completed project by 11:59 on Thursday 4/30. It's **much** better to be early than late, so start working now.

Checkpoint. For full credit, you must also **complete Part 2 of the project (out of 4) and submit it by 11:59pm on Friday 4/24**. You will not have lab time to work on these questions, we recommend that you start early on each part to stay on track.

Partners. You may work with one other partner; this partner **does not** need to be from the same lab. Only one of you is required to submit the project. On okpy.org, the person who submits should also designate their partner so that both of you receive credit.

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

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

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

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

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

In [18]:
```
import numpy as np
import math
import datascience
from datascience import *

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

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 [12]:
```
movies = Table.read_table('movies.csv')
movies.where("Title", "wild wild west").select(0, 1, 2, 3, 4, 14, 49, 1042, 4004)
```

Out[12]:

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 [13]:
```
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[13]: Row(Title='the terminator', Year='1984', Rating=8.1, Genre='thriller', # Words=2210, she
=0.0024084778420038, decid=0.0009633911368015, talk=0.001926782273603, wit=0.0, razor=0.
0, slam=0.0, credit=0.0, rai=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=0.00048169556
84007, cigarett=0.0004816955684007, jesu=0.0, mani=0.0, violin=0.0, financi=0.0, bai=0.
0, cop=0.0004816955684007, neighbor=0.0, cd=0.0, england=0.0, made=0.0004816955684007, c
onni=0.0, instinct=0.0, took=0.0, jacquelin=0.0, mace=0.0, disappear=0.0004816955684007,
waltz=0.0, behind=0.0, bourbon=0.0, favorit=0.0, benni=0.0, manhattan=0.0, nixon=0.0, lu
nch=0.0, principl=0.0, tradit=0.0, counterfeit=0.0, sophi=0.0, third=0.0, exist=0.000963
3911368015, wouldv=0.0, hero=0.0, theyr=0.0004816955684007, anytim=0.0, christin=0.0, va
llei=0.0004816955684007, chess=0.0, paid=0.0, burglar=0.0, nostril=0.0, rubber=0.0004816
955684007, human=0.0009633911368015, 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, latest=0.0, rent=0.0, d
ip=0.0, bell=0.0, andi=0.0, so=0.0057803468208092, london=0.0, cooler=0.0, keaton=0.0, p
ortland=0.0, headlin=0.0, whatta=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, occasion=
0.0, shadow=0.0, work=0.0009633911368015, restrain=0.0, face=0.0004816955684007, exercis
=0.0, sidnei=0.0, pile=0.0, whyd=0.0, teenag=0.0, her=0.0033718689788054, retir=0.0, haz
ard=0.0, roth=0.0, hurrican=0.0, impuls=0.0, ranger=0.0, pour=0.0, lester=0.0, slash=0.
0, deer=0.0, could=0.0014450867052023, vital=0.0, qualiti=0.0, coma=0.0, incred=0.0, han
k=0.0, famili=0.0, duchess=0.0, global=0.0, virgin=0.0, scientif=0.0, between=0.0, holid
ai=0.0, qualifi=0.0, moor=0.0, happili=0.0, arizona=0.0, non=0.0, bruce=0.0, ankl=0.0, c
onstant=0.0, buzz=0.0, harder=0.0, ing=0.0014450867052023, christian=0.0, palmer=0.0, te
nt=0.0, sunset=0.0, damour=0.0, cohaagen=0.0, advertis=0.0, sensat=0.0, local=0.0, there
=0.0072254335260116, terri=0.0, sedat=0.0, rotten=0.0, struck=0.0, deck=0.0, past=0.0, b
ro=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=0.0004816955684007, pencil=0.0, outsid=0.0009633911368015, laboratori=0.0, eas
i=0.0004816955684007, nephew=0.0, coffin=0.0, pretti=0.0004816955684007, coward=0.0, ver
bal=0.0, permiss=0.0, bartend=0.0, wont=0.0014450867052023, watch=0.0, lindenmey=0.0, co
smo=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, quaid=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, n
ow=0.0028901734104046, leak=0.0, underwear=0.0, westlei=0.0, mayb=0.0014450867052023, th
eo=0.0, limo=0.0, cousin=0.0, illeg=0.0, silli=0.0, against=0.0, done=0.000963391136801
5, district=0.0, invad=0.0, ryan=0.0, wait=0.0009633911368015, grudg=0.0, charact=0.0, h
ick=0.0, jami=0.0, lifetim=0.0, lecktor=0.0, and=0.0101156069364162, republican=0.0, lif
e=0.0009633911368015, hidden=0.0, wire=0.0, paranoia=0.0, network=0.0004816955684007, me
ssi=0.0, uthatu=0.0, effort=0.0, carri=0.0, windham=0.0, fun=0.0004816955684007, psychol
ogist=0.0, sean=0.0, scent=0.0, answer=0.0009633911368015, mom=0.0004816955684007, wake=
0.0, sign=0.0, ho=0.0, relat=0.0, jame=0.0, fat=0.0004816955684007, myself=0.0, disrupt=
0.0, scan=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=0.0004816955684007, 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=0.000481695568400
7, econom=0.0, hesit=0.0, autopsi=0.0, jurisdict=0.0, four=0.0004816955684007, factor=0.
0, inquiri=0.0, lion=0.0, meanwhil=0.0, prison=0.0, blair=0.0, seri=0.0009633911368015,
groceri=0.0, surgeri=0.0004816955684007, season=0.0, christi=0.0, clean=0.00048169556840
07, ow=0.0004816955684007, wrestl=0.0, en=0.0, moral=0.0, hungri=0.0, cole=0.0, surfer=
0.0, sixteen=0.0, angl=0.0, shame=0.0, barrel=0.0, major=0.0, ago=0.0, lott=0.0, airplan
=0.0, worth=0.0004816955684007, train=0.0, easili=0.0, feller=0.0, valentin=0.0, harvei=
0.0, wherev=0.0004816955684007, francisco=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, f
rame=0.0, at=0.0028901734104046, psychopath=0.0, press=0.0009633911368015, youyou=0.0, h
avana=0.0, australia=0.0, plai=0.0, mayfield=0.0, chick=0.0, stewart=0.0, seven=0.000481
6955684007, reflect=0.0, outer=0.0, vega=0.0, anywai=0.0004816955684007, prime=0.0, farm
er=0.0, backyard=0.0, joe=0.0, otherwis=0.0, cowgirl=0.0, grate=0.0, clerk=0.0, dispos=
0.0004816955684007, tow=0.0, mari=0.0, certifi=0.0, thi=0.0091522157996147, wheel=0.0, p
rivaci=0.0, todai=0.0, nathan=0.0, teller=0.0, plot=0.0, correct=0.0, couch=0.0, job=0.0
004816955684007, hurt=0.0004816955684007, inject=0.0, chocol=0.0, session=0.0, outrag=0.
0, reduc=0.0, knew=0.0009633911368015, jd=0.0, perfum=0.0, fabric=0.0, bodyguard=0.0, th

ink=0.0033718689788054, il=0.0, yesterdai=0.0, side=0.0004816955684007, doesnt=0.0004816
955684007, ronni=0.0, blank=0.0, jess=0.0, push=0.0, ahh=0.0, jealou=0.0, alter=0.0, ble
w=0.0, bu=0.0, off=0.0, sweetheart=0.0, abl=0.0, angelo=0.0, nicer=0.0, coupla=0.0, resu
m=0.0, coke=0.0, strangl=0.0, gut=0.0, morn=0.0, miracl=0.0, bit=0.0004816955684007, int
imid=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, oldfashion=0.0, accent=0.0, spitz=0.0, dirti=0.0, shot=0.00096339113680
15, lit=0.0, cedar=0.0, pirat=0.0, weather=0.0, stun=0.0, learn=0.0009633911368015, wick
=0.0, bring=0.0004816955684007, slack=0.0, brave=0.0, shakespear=0.0, monkei=0.0, presum
=0.0, vacat=0.0, faint=0.0, strap=0.0, stephen=0.0, maggi=0.0, indic=0.0, sundai=0.0, no
is=0.0, organ=0.0009633911368015, terranc=0.0, foundat=0.0, littl=0.0, perman=0.0, insid
=0.0004816955684007, stabl=0.0004816955684007, sharp=0.0, uptight=0.0, wholl=0.0, jeffre
i=0.0, root=0.0, thy=0.0, josi=0.0, woman=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=0.0004816955684
007, statement=0.0004816955684007, defin=0.0, drawn=0.0, progress=0.0, year.1=0.00144508
67052023, shovel=0.0, sequenc=0.0, andand=0.0, reel=0.0, held=0.0, youv=0.00096339113680
15, trick=0.0, horseman=0.0, whoa=0.0, emploi=0.0, chain=0.0, cmon=0.0, brief=0.0, creat
iv=0.0, moscow=0.0, challeng=0.0, walli=0.0, golf=0.0, abort=0.0004816955684007, aha=0.0
004816955684007, bent=0.0, exclus=0.0, amber=0.0, figur=0.0, healthi=0.0, ransom=0.0, st
eer=0.0, blow=0.0004816955684007, bark=0.0, imbecil=0.0, mother=0.0014450867052023, had=
0.0043352601156069, whatev=0.0004816955684007, donit=0.0, goodby=0.0, terrifi=0.0, cash=
0.0004816955684007, descript=0.0, spell=0.0, west=0.0, shoulda=0.0, when=0.0033718689788
054, wear=0.0004816955684007, crop=0.0, trapper=0.0, donat=0.0, breath=0.000481695568400
7, bracelet=0.0, lover=0.0, afraid=0.0004816955684007, vice=0.0, ms=0.0004816955684007,
didnt=0.0024084778420038, kill=0.0014450867052023, depth=0.0, si=0.0, experiment=0.0, dn
a=0.0, desmond=0.0, coconut=0.0, dil=0.0, llewelyn=0.0, spoil=0.0, lung=0.0, attent=0.00
04816955684007, offens=0.0, babi=0.0, havin=0.0, mustnt=0.0, creepi=0.0, daniel=0.0, aba
ndon=0.0, less=0.0004816955684007, go=0.0091522157996147, negoti=0.0, butcher=0.0, sudde
n=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, pic
ard=0.0, success=0.0, as=0.001926782273603, heroin=0.0, hi=0.0009633911368015, 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.
0004816955684007, strong=0.0, ditch=0.0, doc=0.0, wors=0.0, trigger=0.0, mister=0.0, req
uest=0.0, direct=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=0.0004816955684007, episod=0.0, reliant=0.0, jason=0.0, get
=0.0072254335260116, duh=0.0, victori=0.0, speci=0.0, includ=0.0, time=0.00192678227360
3, imagin=0.0004816955684007, hang=0.0, estat=0.0, full=0.0, sleep=0.0014450867052023, r
eceiv=0.0, grei=0.0004816955684007, belief=0.0, itali=0.0, separ=0.0, claw=0.0, circuit=
0.0, memo=0.0, monei=0.0004816955684007, anna=0.0, term=0.0, drunk=0.0, everydai=0.0, ha
ldeman=0.0, frozen=0.0, castro=0.0, soup=0.0, mornin=0.0, starship=0.0, dawn=0.0, curtai
n=0.0, pipe=0.0, twist=0.0, downtown=0.0, drive=0.0, tank=0.0, servant=0.0, circumst=0.
0, town=0.0, beat=0.0, wooden=0.0, thursdai=0.0, ordinari=0.0, acid=0.0, must=0.00289017
34104046, literatur=0.0, carol=0.0, homework=0.0, present=0.0, lunat=0.0, lifestyl=0.0,
rat=0.0, satisfi=0.0, funni=0.0, gang=0.0, jewelri=0.0, develop=0.0004816955684007, yell
=0.0, stash=0.0, ab=0.0, mm=0.0, thumb=0.0, to=0.0231213872832371, 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=0.0004816955684007, 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=0.0004816955684007, fe
et=0.0004816955684007, dee=0.0, casual=0.0, check=0.0014450867052023, fugit=0.0, garden=
0.0, million=0.0, suppos=0.0014450867052023, observ=0.0, low=0.0004816955684007, cathol=
0.0, parad=0.0, confus=0.0, appli=0.0, uhura=0.0, sane=0.0, beverli=0.0, princip=0.0, he
len=0.0, drain=0.0, want=0.0038535645472061, maam=0.0, care=0.0, tomorrow=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, gaston=0.0, individu=0.0, treasur=0.0, brown=0.0, delic=0.0, forbidden=0.0, b
lah=0.0, although=0.0, cold=0.0009633911368015, strategi=0.0, psycholog=0.0, fill=0.0, p
rofession=0.0, rig=0.0, theori=0.0, psychiatr=0.0, raw=0.0, critic=0.0, contribut=0.0, f
art=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, cycl=0.
0, accid=0.0, puzzl=0.0, invis=0.0004816955684007, ronald=0.0, mercuri=0.0, escap=0.0, d
amon=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=0.0004816955684007, diari=0.0, atlant=0.0, abse

nc=0.0, corps=0.0, relai=0.0, rip=0.0004816955684007, bull=0.0, requir=0.0, buck=0.0, co
mplain=0.0, russia=0.0, arctic=0.0, schedul=0.0004816955684007, casei=0.0, temperatur=0.
0, tree=0.0, unbeliev=0.0, graduat=0.0, place=0.0009633911368015, communist=0.0, nine=0.
0, vulner=0.0, hike=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, malkovic
h=0.0, cord=0.0, edmund=0.0, walk=0.0004816955684007, cake=0.0, protect=0.0, nearest=0.
0, takin=0.0, comrad=0.0, dai=0.0, tour=0.0, whose=0.0, partial=0.0, vada=0.0, delight=
0.0, godfath=0.0, cheat=0.0, harmless=0.0, romant=0.0, pound=0.0, im=0.0077071290944123,
violat=0.0, chief=0.0, onlin=0.0, pleasur=0.0, crystal=0.0, luther=0.0, festiv=0.0, alon
g=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=0.0004816955684007, father=0.0004816955684007, 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=0.000481695568
4007, verifi=0.0, prepar=0.0004816955684007, cia=0.0, flew=0.0, logic=0.0, affect=0.0, s
kirt=0.0, nineteen=0.0, breakdown=0.0, william=0.0, batteri=0.0, shore=0.0, project=0.0,
strength=0.0004816955684007, bail=0.0, piti=0.0004816955684007, harbor=0.0, oper=0.0, ya
nke=0.0, orphan=0.0, squar=0.0, gari=0.0, driven=0.0, nanci=0.0004816955684007, leonard=
0.0, nonsens=0.0, anybodi=0.0004816955684007, strict=0.0, riddl=0.0, boulevard=0.0, arti
cl=0.0, shop=0.0, mexico=0.0, ruin=0.0004816955684007, did=0.0028901734104046, 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=0.0004816955684007, flight=0.0, refriger=0.0, loretta=0.0, geniu=0.0, condit=0.0, f
or=0.0091522157996147, anyon=0.0, ideal=0.0004816955684007, terribl=0.0, otho=0.0, self=
0.0, swore=0.0, ring=0.0, baron=0.0, civilian=0.0, panick=0.0, settl=0.0, spread=0.0, tu
rn=0.0, blast=0.0, growth=0.0, seal=0.0, box=0.0, locker=0.0, help=0.0033718689788054, c
ast=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, entranc=0.0, corner=0.0, mama=0.
0, supposedli=0.0, yo=0.0, role=0.0, wai=0.0009633911368015, vicki=0.0, buddi=0.00048169
55684007, wizard=0.0, heal=0.0, element=0.0, here=0.0072254335260116, leather=0.0, hardl
i=0.0, their=0.0, sworn=0.0, africa=0.0, yearold=0.0, sphere=0.0, psych=0.0, behalf=0.0,
moren=0.0, shelter=0.0, dr=0.0004816955684007, weight=0.0, ami=0.0, pike=0.0, gitt=0.0,
kinda=0.0004816955684007, liber=0.0, badg=0.0, last=0.0004816955684007, from=0.003371868
9788054, uhhuh=0.0, afford=0.0, shag=0.0, presenc=0.0, jai=0.0, hall=0.0, worst=0.0, age
nc=0.0, threat=0.0004816955684007, jaw=0.0, hold=0.0009633911368015, someplac=0.0, birth
=0.0, junk=0.0, natali=0.0, graveyard=0.0, march=0.0, focu=0.0, graviti=0.0, smoke=0.0,
normal=0.0, congress=0.0, ash=0.0, predict=0.0, copper=0.0, court=0.0, wealth=0.0, it=0.
0313102119460503, um=0.0, issu=0.0, misunderstand=0.0, berlin=0.0, structur=0.0, hardwar
=0.0, casanova=0.0, worship=0.0, unhappi=0.0, spit=0.0, quadrant=0.0, downstair=0.0, see
=0.0043352601156069, gentlemen=0.0, intern=0.0, conrad=0.0, flame=0.0, within=0.0, drum=
0.0, yet=0.0014450867052023, differ=0.001926782273603, jacket=0.0, onto=0.0, prospect=0.
0, nation=0.0, soft=0.0, girl=0.0014450867052023, counter=0.0, rug=0.0, neutral=0.0, ell
i=0.0, transmiss=0.0, integr=0.0, cadillac=0.0, hose=0.0, lillian=0.0, clip=0.0, stanlei
=0.0, marriag=0.0, built=0.0004816955684007, paso=0.0, subwai=0.0, whiskei=0.0, kyle=0.0
038535645472061, sai=0.0014450867052023, barzini=0.0, virginia=0.0, lou=0.0, slip=0.0004
816955684007, pageant=0.0, tellin=0.0, taylor=0.0, proof=0.0, yank=0.0, yacht=0.0, bless
=0.0, brooklyn=0.0, argu=0.0, pistol=0.0, bo=0.0, unnecessari=0.0, share=0.0, th=0.0, sa
botag=0.0, china=0.0, hojon=0.0, intim=0.0, chop=0.0, comic=0.0, compuls=0.0, bingo=0.0,
make=0.0043352601156069, vulcan=0.0, parri=0.0, shouldnt=0.0004816955684007, disconnect=
0.0004816955684007, toon=0.0, forc=0.0, annabel=0.0, clown=0.0, sentenc=0.0, with=0.0024
084778420038, cartoon=0.0, suspicion=0.0, willi=0.0, brook=0.0, revers=0.0, luci=0.0, vi
sitor=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=0.0009633911368015, weapon=0.000
9633911368015, kate=0.0, distant=0.0, grade=0.0, atf=0.0, balloon=0.0, southern=0.0, cha
ng=0.0, oppos=0.0, indict=0.0, respect=0.0, sure=0.0014450867052023, jane=0.0, doubt=0.0
004816955684007, 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=0.0004816955684007,
motiv=0.0, rel=0.0, herself=0.0, curios=0.0, buffalo=0.0, terrorist=0.0, write=0.0, scho
ol=0.0, wing=0.0, smack=0.0, marti=0.0, counti=0.0, pud=0.0, thatd=0.0, readi=0.0, manra
i=0.0, airlin=0.0, romeo=0.0, weekend=0.0, apolog=0.0, ladi=0.0, emili=0.0, maker=0.0, d
ian=0.0, barf=0.0, quest=0.0, crown=0.0, expect=0.0, gull=0.0, pump=0.0, depress=0.0, in
terrog=0.0, instant=0.0, barri=0.0, book=0.0004816955684007, ahm=0.0, shutup=0.0, larri=
0.0, seriou=0.0, hobbi=0.0, unotu=0.0, kingdom=0.0, on=0.0130057803468208, quarter=0.0,

nicki=0.0, pierr=0.0, dream=0.0, clarenc=0.0, buffi=0.0, winner=0.0, scatter=0.0, erik=
0.0, bike=0.0, bean=0.0, unabl=0.0, women=0.0009633911368015, maya=0.0, tommi=0.0, seven
tyf=0.0, curs=0.0, maintain=0.0, genesi=0.0, background=0.0, dude=0.0, satellit=0.0, hou
r=0.0014450867052023, extend=0.0, transport=0.0, hull=0.0, top=0.0, grief=0.0, evid=0.0,
gradi=0.0, onc=0.0004816955684007, philadelphia=0.0, honesti=0.0, super=0.0, florida=0.
0, gentleman=0.0, cannot=0.0, too=0.0014450867052023, fuss=0.0, hasnt=0.0, lookin=0.0, a
rrang=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=0.0004816955684007, 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=0.0004816955684007, flaw=0.0, bake=0.0, mmm=
0.0004816955684007, 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=0.0004816955684007, hope=0.0, layer=0.0, to
m=0.0, reaction=0.0, san=0.0, katrina=0.0, identifi=0.0, red=0.0, exorc=0.0, iron=0.0, b
eer=0.0, santo=0.0, bruis=0.0, stab=0.0, louis=0.0, agreement=0.0, coulda=0.0, film=0.0,
rufu=0.0, soil=0.0, playin=0.0, millionair=0.0, middl=0.0, closer=0.0, spirit=0.0, accid
ent=0.0, yer=0.0, traffic=0.0, what=0.0096339113680154, 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, surp
ris=0.0, rm=0.0, hid=0.0, sewer=0.0, kilomet=0.0, lisa=0.0, workshop=0.0, safeti=0.0, to
uch=0.0009633911368015, jersei=0.0, frequent=0.0, enjoi=0.0, loos=0.0, homeless=0.0, ah=
0.0, extens=0.0, termin=0.0014450867052023, level=0.0, violent=0.0, rush=0.0, coordin=0.
0, earli=0.0, wa=0.0072254335260116, dizzi=0.0, privat=0.0, gwen=0.0, suicid=0.0, headqu
art=0.0, educ=0.0, sort=0.0004816955684007, 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=0.0
048169556840077, either=0.0, pry=0.0, conspiraci=0.0, basi=0.0, timer=0.0, heat=0.0, law
son=0.0, hear=0.0004816955684007, ui=0.0, fool=0.0, mere=0.0, huh=0.0004816955684007, do
nni=0.0, none=0.0009633911368015, fifth=0.0, becaus=0.0, lamb=0.0, interpret=0.0, increa
s=0.0, tower=0.0, mickei=0.0, colleagu=0.0, confer=0.0, hollow=0.0, leon=0.0, thou=0.0,
fran=0.0, percent=0.0, six=0.0004816955684007, limp=0.0, arrowai=0.0, explod=0.0, friend
li=0.0, breakfast=0.0, greek=0.0, need=0.0004816955684007, rome=0.0, beast=0.0, rehab=0.
0, ben=0.0, land=0.0, river=0.0, frank=0.0, smash=0.0004816955684007, quicker=0.0, forme
r=0.0, lower=0.0, rap=0.0, nick=0.0, came=0.0, hopeless=0.0, comedian=0.0, right=0.00192
6782273603, yall=0.0, truli=0.0, not=0.0081888246628131, soze=0.0, forgotten=0.0, tight=
0.0004816955684007, pro=0.0, sona=0.0, under=0.0004816955684007, precis=0.0, center=0.0,
stiff=0.0, virtual=0.0, author=0.0004816955684007, dry=0.0, theyll=0.0, golden=0.0, fina
l=0.0, properli=0.0, paper=0.0, older=0.0, serv=0.0, dant=0.0, whistl=0.0, suffer=0.0, c
onfid=0.0, fraud=0.0, brain=0.0004816955684007, minu=0.0, twentyf=0.0, cut=0.0, atmosphe
r=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=0.0004816955684007, niec=0.0, ca
lib=0.0004816955684007, hafta=0.0, journei=0.0, poster=0.0, thruster=0.0, dela=0.0, cele
br=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=0.0009633911368015, msieu=0.0, lone=0.0, mole=0.0, briefcas=0.0, rudi=0.0, excel=0.
0, madman=0.0, nazi=0.0, flop=0.0, invent=0.0, signor=0.0, suggest=0.0, edward=0.0, stat
ion=0.0, senat=0.0, amen=0.0, hip=0.0, price=0.0, awai=0.0004816955684007, randal=0.0, h
igh=0.0, field=0.0004816955684007, 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=0.0009633911368015, awak=0.0, o
ak=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=0.0091522157996147, pictur=0.00048
16955684007, awar=0.0, sponsor=0.0, seattl=0.0, english=0.0, introduc=0.0, health=0.0, h
allucin=0.0, quickli=0.0, jacob=0.0, crew=0.0, cuervo=0.0, alabama=0.0, teresa=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, vehicl=0.0, wisdom=0.0, find=0.0004816955684007,
ruben=0.0, both=0.0, report=0.0, denver=0.0, helicopt=0.0, complex=0.0004816955684007, n
eednt=0.0, be=0.0101156069364162, greet=0.0, spock=0.0, song=0.0, bibl=0.0, lipstick=0.
0, stroke=0.0, persuad=0.0, recommend=0.0, deliveri=0.0, hors=0.0, shut=0.00048169556840
07, zone=0.0, bomb=0.0, linda=0.0, tube=0.0004816955684007, gross=0.0, order=0.000963391
1368015, dillon=0.0, action=0.0, spot=0.0009633911368015, resign=0.0, barn=0.0, bush=0.
0, itu=0.0, cry=0.0004816955684007, 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, sho
wn=0.0, bread=0.0, mind=0.0004816955684007, saint=0.0, oclock=0.0, sail=0.0, stranger=0.
0, smile=0.0, product=0.0, rubi=0.0, disabl=0.0, question=0.0, philip=0.0, exampl=0.0, f
reddi=0.0, stori=0.0, chariti=0.0, franklin=0.0, routin=0.0, engin=0.0, lax=0.0, blade=

0.0, chew=0.0, buljanoff=0.0, counselor=0.0, houston=0.0, alright=0.001926782273603, sup
port=0.0, wheelchair=0.0, goe=0.0004816955684007, rachel=0.0, stuff=0.0009633911368015,
neck=0.0, have=0.0028901734104046, divis=0.0004816955684007, 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, s
lept=0.0, bank=0.0, hurri=0.0, dure=0.0, sunni=0.0, ink=0.0, vein=0.0, immigr=0.0, conce
ntr=0.0, kat=0.0, eager=0.0, don=0.0, somewher=0.0, religi=0.0, pierc=0.0, bureau=0.0, t
heyd=0.0, mueller=0.0, familiar=0.0, bonu=0.0, austrian=0.0, violenc=0.0, produc=0.0, to
night=0.0, royal=0.0, breakin=0.0, map=0.0, sayin=0.0, discuss=0.0, hed=0.0, strictli=0.
0, led=0.0, mile=0.0, thisll=0.0, dammit=0.0, leav=0.0004816955684007, dummi=0.0, reacto
r=0.0, sauc=0.0, rice=0.0, coupl=0.0, clever=0.0, choic=0.0004816955684007, chrissak=0.
0, director=0.0, kastl=0.0, borg=0.0, fax=0.0, brought=0.0, disturb=0.0, poison=0.0, gri
ssom=0.0, shake=0.0004816955684007, corrupt=0.0, stall=0.0, sarah=0.0033718689788054, st
arter=0.0, alik=0.0, quiet=0.0, entertain=0.0, demonstr=0.0, oxygen=0.0, asham=0.0, unde
rcov=0.0, beef=0.0, nice=0.0, televis=0.0, oscar=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, c
rowd=0.0, tap=0.0, fix=0.0, store=0.0, grail=0.0, vietnam=0.0, candl=0.0, whoop=0.0, tak
en=0.0004816955684007, end=0.0, eleph=0.0, prefer=0.0, brian=0.0, lamar=0.0, ma=0.0, vic
toria=0.0, surviv=0.0004816955684007, three=0.0, bobbi=0.0, stage=0.0, steed=0.0, macfar
lan=0.0, spider=0.0, trial=0.0, suprem=0.0, awfulli=0.0, foot=0.0, ar=0.004816955684007
7, thrown=0.0, ask=0.0009633911368015, cap=0.0, strang=0.0, phoenix=0.0, boot=0.0, georg
ia=0.0, replac=0.0, reckon=0.0, divorc=0.0, arrest=0.0, horni=0.0, grandfath=0.0, proble
m=0.0, bust=0.0, pari=0.0, roommat=0.0, consum=0.0, ag=0.0, been=0.0014450867052023, spa
ghetti=0.0, minimum=0.0, car=0.0, recruit=0.0, farm=0.0, dave=0.0, tape=0.0, regular=0.
0, decor=0.0, shirt=0.0, multipl=0.0, mechan=0.0, effici=0.0, all=0.001926782273603, thi
ev=0.0, pose=0.0, silenc=0.0, lenni=0.0, radiat=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=0.0077071290944123, bride=0.
0, hole=0.0, toler=0.0, content=0.0, applic=0.0, frequenc=0.0, sore=0.0, ranch=0.0, fair
=0.0, nowher=0.0, monitor=0.0, peanut=0.0, presid=0.0, cooper=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=0.0009633911368015, 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, guinea=0.0, wanna=
0.0, hate=0.0014450867052023, dash=0.0, pull=0.0, lab=0.0004816955684007, disast=0.0, lo
bbi=0.0, plug=0.0, rear=0.0, conveni=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, coincid=
0.0, mel=0.0, knock=0.0, search=0.0, pan=0.0, if=0.0028901734104046, odd=0.0, jenni=0.0,
howd=0.0, simon=0.0, dodg=0.0, outfit=0.0, floor=0.0004816955684007, uhh=0.0, nah=0.0, j
immi=0.0, chuck=0.001926782273603, rich=0.0, emot=0.0, destruct=0.0, blue=0.0, game=0.0,
peculiar=0.0, internet=0.0, paranoid=0.0, paradis=0.0, deposit=0.0, look=0.0033718689788
054, 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=0.0004816955684007, conceiv=0.0004816955684007, messag=0.0014450867052023, d
oor=0.0004816955684007, mondai=0.0, mayor=0.0, fanci=0.0, cuba=0.0, most=0.0009633911368
015, barrier=0.0, eject=0.0, grew=0.0004816955684007, alic=0.0, jazz=0.0, thread=0.00048
16955684007, commun=0.0, treati=0.0, lieuten=0.0004816955684007, tran=0.0, mackelwai=0.
0, colonel=0.0, vincent=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, medic=0.0, hit=
0.0, grow=0.0, ventur=0.0, mistaken=0.0, difficulti=0.0, thirtyf=0.0, pm=0.0, peter=0.0,
orang=0.0, kei=0.0, earl=0.0, ex=0.0, vault=0.0, doe=0.0, music=0.0, romanc=0.0, blanket
=0.0, colleg=0.0, fog=0.0, bad=0.0004816955684007, novel=0.0, kidnap=0.0, consult=0.0, r
ecogn=0.0, laugh=0.0, shooter=0.0, galaxi=0.0, paulin=0.0, procedur=0.0, seed=0.0, radio
act=0.0, ars=0.0, total=0.0, theater=0.0, cute=0.0, iraq=0.0, net=0.0, laplant=0.0, wait
er=0.0, print=0.0, milo=0.0, judgment=0.0, act=0.0, gone=0.0014450867052023, lecter=0.0,
draft=0.0, build=0.0024084778420038, vow=0.0, steam=0.0, veteran=0.0, eighth=0.0, insect
=0.0, entir=0.0004816955684007, hon=0.0, okai=0.0038535645472061, fourteen=0.0, meet=0.0
009633911368015, tens=0.0, doin=0.0, hat=0.0, sub=0.0, counsel=0.0, channel=0.0, close=
0.0, daylight=0.0, loser=0.0, tend=0.0, judgement=0.0, accus=0.0, coast=0.0, everywher=
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=0.0009633911368015, wa
rrior=0.0, asid=0.0, favor=0.0004816955684007, 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=0.0004816955684007, 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, girlf
riend=0.0004816955684007, anywher=0.0, air=0.0, murder=0.0, younger=0.0, handl=0.0004816

955684007, albanian=0.0, laser=0.0, dull=0.0, guidanc=0.0, live=0.0024084778420038, 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, acknowledg=0.0, dread=0.0, manual=0.0, ll=0.0, stew=0.0, pai=0.0009633911368015, craw
ford=0.0, janet=0.0, loan=0.0, rib=0.0, peel=0.0, hormon=0.0, about=0.0052986512524085,
ohio=0.0, advisor=0.0, crusher=0.0, centuri=0.0, govern=0.0004816955684007, probabl=0.00
04816955684007, slow=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=0.001926782273603, surgic=0.0, tal
ent=0.0, futur=0.0024084778420038, appl=0.0, hiya=0.0, should=0.0004816955684007, hoop=
0.0, joseph=0.0, uisu=0.0, unit=0.0004816955684007, radar=0.0, magnific=0.0, rendezv=0.
0, anxiou=0.0, featur=0.0, discov=0.0, dental=0.0, legal=0.0004816955684007, shuttl=0.0,
word=0.0, tool=0.0, deceas=0.0, consider=0.0, noon=0.0004816955684007, york=0.0, lauri=
0.0, fell=0.0, reject=0.0, ton=0.0, lead=0.0, marin=0.0, name=0.0028901734104046, commis
s=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, commission=0.0, joke=0.0,
idea=0.0, deserv=0.0, shove=0.0, rd=0.0, mistak=0.0004816955684007, earthquak=0.0, worm=
0.0, abil=0.0, guarante=0.0, outta=0.0, mutual=0.0, prize=0.0, welcom=0.0, seventi=0.0,
trunk=0.0, teas=0.0, access=0.0, jacki=0.0, refresh=0.0, affirm=0.0, ill=0.0009633911368
015, gregor=0.0, chet=0.0, complet=0.0, mill=0.0, speak=0.0, whatiya=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, fr
ed=0.0, peggi=0.0, emerg=0.0, maid=0.0, relax=0.0004816955684007, whaddya=0.0, give=0.00
09633911368015, starfleet=0.0, rout=0.0, resourc=0.0, night=0.0004816955684007, statu=0.
0, spare=0.0, abov=0.0, save=0.0, fett=0.0, goat=0.0, career=0.0, van=0.0, puke=0.000481
6955684007, disagre=0.0, reveng=0.0, femal=0.0, penni=0.0, better=0.0014450867052023, re
x=0.0, marg=0.0, but=0.0048169556840077, 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, actress=0.0, whisper=0.0, goin=0.0, particularli=0.0, gossip=0.0, r
unnin=0.0, peni=0.0, minut=0.0004816955684007, scandal=0.0, awkward=0.0, crank=0.0, ev=
0.0, owen=0.0, diet=0.0, cargo=0.0, fashion=0.0, tale=0.0, quietli=0.0, son=0.0009633911
368015, geek=0.0, devot=0.0, madelein=0.0, luke=0.0, laval=0.0, seventeen=0.0, yourselv=
0.0, sleev=0.0, robberi=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=0.0004816955684007, predat=0.0, temper=0.0, globe=0.0,
pacif=0.0, ethel=0.0, phaser=0.0, midnight=0.0, lock=0.0, adult=0.0, brandon=0.0, kitche
n=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, materi=0.0, function=0.0, zip=0.0, visa=0.0, josh=0.0, shed=0.0, crash=0.
0, craig=0.0, he=0.00626204238921, perri=0.0004816955684007, pathet=0.0, eastern=0.0, de
mand=0.0, sheet=0.0, data=0.0, fbi=0.0, insist=0.0, regist=0.0, fame=0.0, particular=0.
0, upon=0.0, screen=0.0, becki=0.0, ya=0.001926782273603, attornei=0.0, clai=0.0, mate=
0.0, interview=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=0.0004816955684007, rode=0.0, creat=0.0, disa
ppoint=0.0004816955684007, region=0.0, marcia=0.0, defens=0.001926782273603, sensit=0.0,
defend=0.0, miseri=0.0, take=0.0009633911368015, oil=0.0, holi=0.0, polit=0.0, crazi=0.0
004816955684007, interrupt=0.0, luca=0.0, franc=0.0, pete=0.0, afternoon=0.0, distress=
0.0, finder=0.0, around=0.0004816955684007, europ=0.0, sit=0.0, follow=0.000481695568400
7, 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=0.0004816955684007, pawn=0.0, wonder=0.0, e
asier=0.0, lt=0.0, zero=0.0014450867052023, stake=0.0, food=0.0, forest=0.0, june=0.0, p
ickup=0.0, sensor=0.0, choke=0.0, situat=0.0, david=0.0, blackmail=0.0, atom=0.0, necess
arili=0.0, albert=0.0, advantag=0.0, fli=0.0, horn=0.0, metal=0.0004816955684007, conseq
u=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=0.0009633911368015, math=0.0, sue=0.0, foul=0.0, like=0.0077
071290944123, ferri=0.0, praetor=0.0, marri=0.0, stamp=0.0, lainei=0.0, gai=0.0, potato=
0.0, licens=0.0004816955684007, some=0.0028901734104046, sylvia=0.0, tag=0.0, caught=0.0
009633911368015, runner=0.0, unpleas=0.0, typic=0.0, dive=0.0, devic=0.0, alien=0.0, dea
dlin=0.0, landlord=0.0, bridg=0.0, sink=0.0, affair=0.0, record=0.0004816955684007, thin
kin=0.0, gate=0.0, austin=0.0, poker=0.0, liquor=0.0, cow=0.0, run=0.0009633911368015, y
ourself=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=0.0004816955684007, effect=0.0004816955684007, appreci=0.0, sh
ark=0.0, child=0.0004816955684007, bend=0.0, slice=0.0, hair=0.0, patienc=0.0, ga=0.0, d
isco=0.0, expert=0.0, implant=0.0, dealer=0.0, oooh=0.0, beavi=0.0, troubl=0.0, lesli=0.

0, lectur=0.0, youd=0.0009633911368015, baldwin=0.0, scienc=0.0, surg=0.0, uallu=0.0, oo
h=0.0, harvard=0.0, smart=0.0, statist=0.0, sheila=0.0, haul=0.0, livingston=0.0, exhaus
t=0.0, jail=0.0004816955684007, biggest=0.0, hudsuck=0.0, rocket=0.0, trace=0.0004816955
684007, punish=0.0, contain=0.0, twomblei=0.0, intend=0.0, flynn=0.0, describ=0.0, contr
act=0.0, straighten=0.0, reilli=0.0, regula=0.0, tremend=0.0, love=0.0, setup=0.0, enfor
c=0.0, castl=0.0, lousi=0.0, bloke=0.0, marietta=0.0, cooki=0.0, someon=0.00096339113680
15, design=0.0, address=0.0, knowi=0.0, award=0.0, tick=0.0, unconsci=0.0, weve=0.0, cle
veland=0.0, sweeti=0.0, intellect=0.0, beard=0.0, surrend=0.0, vinci=0.0, hildi=0.0, tai
l=0.0, dignan=0.0, moss=0.0, photo=0.0, myth=0.0, tuesdai=0.0, stress=0.0, limb=0.0, wal
let=0.0, tooth=0.0, bare=0.0, ten=0.0004816955684007, guilti=0.0, inform=0.0, list=0.0,
husband=0.0, visual=0.0, ross=0.0, jabez=0.0, cape=0.0, atlanta=0.0, champion=0.0, valua
bl=0.0, respons=0.0, addit=0.0, spike=0.0, lighten=0.0, wagon=0.0, owl=0.0, dog=0.001445
0867052023, santa=0.0, fantasi=0.0, belt=0.0, unowu=0.0, enorm=0.0, kind=0.0, light=0.00
09633911368015, 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, fought=0.0, perimet=0.0, wasnt=0.0004816955684007, t
estimoni=0.0, conklin=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, kaufman=0.0, pole=0.0, engag=0.0, plain=
0.0, suppli=0.0, reserv=0.0, few=0.0004816955684007, sandwich=0.0, fredo=0.0, safe=0.000
9633911368015, hockei=0.0, hous=0.0, elain=0.0, knox=0.0, oath=0.0, wander=0.0, latin=0.
0, same=0.0024084778420038, git=0.0, garbag=0.0, charg=0.0, knowledg=0.0, uknowu=0.0, ve
ronica=0.0, john=0.001926782273603, connel=0.0, anticip=0.0, drown=0.0, rose=0.0, carpet
=0.0, circu=0.0, sebastian=0.0, cmere=0.0, irish=0.0, robert=0.0, fork=0.0, lazi=0.0, fl
are=0.0, bobo=0.0, guess=0.0004816955684007, hippi=0.0, the=0.040462427745665, specif=0.
0, offici=0.0009633911368015, 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=0.000481695568400
7, scotch=0.0, brenner=0.0, lake=0.0, therel=0.0, gordo=0.0, everett=0.0, neural=0.0, ma
dison=0.0, realiz=0.0, georg=0.0, skate=0.0, drink=0.0004816955684007, goddam=0.0, anyho
w=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=0.0004816955684007, soon=0.0009633911368015, helpless=0.
0, then=0.0033718689788054, cobb=0.0, weird=0.0009633911368015, parker=0.0, seen=0.00096
33911368015, happier=0.0, roast=0.0, tip=0.0004816955684007, legend=0.0009633911368015,
behavior=0.0, secur=0.0, tatum=0.0, especi=0.0004816955684007, jeez=0.0, your=0.00963391
13680154, dispatch=0.0, heart=0.0004816955684007, test=0.0, webster=0.0, categori=0.0, c
hicken=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, rain=0.0, obsess=0.0, michael=0.0, corleon=0.0, vision=0.0, edg
=0.0, command=0.0, nineti=0.0, gin=0.0, nix=0.0, amus=0.0, afterward=0.0, wanta=0.0, tho
rwald=0.0, by=0.0028901734104046, rag=0.0, jack=0.0, desert=0.0, maureen=0.0, own=0.0009
633911368015, heck=0.0, walker=0.0, empti=0.0, grunemann=0.0, relief=0.0, narrow=0.0, sa
avik=0.0, seymour=0.0, anonym=0.0, footag=0.0, territori=0.0, maniac=0.0, envelop=0.0, d
espis=0.0, definit=0.0, return=0.0, nobodi=0.0024084778420038, eventu=0.0, rita=0.0, pep
per=0.0, prep=0.0, stand=0.0, deepli=0.0, park=0.0, aunt=0.0, youth=0.0, water=0.0, els=
0.001926782273603, starl=0.0, allei=0.0, snow=0.0, pope=0.0, wax=0.0, object=0.000481695
5684007, basic=0.0004816955684007, file=0.0, dismiss=0.0, much=0.001926782273603, someti
m=0.0004816955684007, 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, to
ss=0.0, pure=0.0, maxin=0.0, canadian=0.0, amount=0.0, shatter=0.0, spoke=0.0, danc=0.0,
ir=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=0.0038535645472061, resist=0.0, j
uic=0.0, reliabl=0.0, will=0.0038535645472061, 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, disea
s=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=0.0004816955684007, cruis=0.0, d
aughter=0.0, premier=0.0, law=0.0, intent=0.0, liz=0.0, nake=0.0004816955684007, petti=
0.0, mirror=0.0, ha=0.001926782273603, each=0.0, never=0.0024084778420038, do=0.00529865
12524085, exact=0.0, rescu=0.0, drivin=0.0, limit=0.0, ti=0.0, manag=0.0004816955684007,
thirsti=0.0, bargain=0.0004816955684007, origin=0.0, acceler=0.0, pilot=0.0, guid=0.0, m
ummi=0.0, mad=0.0, frighten=0.0, factori=0.0004816955684007, cotton=0.0, rape=0.0, chef=
0.0, leg=0.0, calm=0.0, fail=0.0, interior=0.0, germ=0.0, inch=0.0, donald=0.0, smear=0.
0, berserk=0.0, finest=0.0, hard=0.0004816955684007, 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=0.0004816955684007, mil
itari=0.0004816955684007, scoop=0.0, saw=0.0004816955684007, machin=0.0014450867052023,
german=0.0, enemi=0.0009633911368015, creatur=0.0, throat=0.0, stuf=0.0, impli=0.0, burn
t=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, confidenti=0.0,
lost=0.0009633911368015, 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=0.0057803468208092, tiger=0.0, pair=0.0, below=0.0, him=0.
0048169556840077, un=0.0, mortgag=0.0, toto=0.0, dine=0.0, gun=0.0, thirti=0.00096339113
68015, warp=0.0, sid=0.0, standard=0.0, deliv=0.0, couldv=0.0, sandra=0.0, wide=0.0, day
dai=0.0, nail=0.0, novemb=0.0, speck=0.0, broken=0.0, fulli=0.0, di=0.0004816955684007,
concept=0.0, plenti=0.0, sara=0.0, toilet=0.0, smyth=0.0, dirt=0.0, skull=0.0, match=0.
0, win=0.0004816955684007, abduct=0.0, chamber=0.0, putter=0.0, advanc=0.0, baxter=0.0,
group=0.0, onli=0.0, commerci=0.0, uniqu=0.0, bet=0.0004816955684007, back=0.00144508670
52023, aisl=0.0, darlin=0.0, creation=0.0, target=0.0, sip=0.0, sam=0.0, worker=0.0, fiv
e=0.0, ladder=0.0, capit=0.0, smaller=0.0, merri=0.0, won=0.0004816955684007, howard=0.
0, eleg=0.0, bee=0.0, digit=0.0, lowel=0.0, outstand=0.0, meant=0.0009633911368015, prec
aut=0.0, degre=0.0, suspend=0.0, week=0.0004816955684007, evacu=0.0, wow=0.0, loui=0.0,
iii=0.0, per=0.0, tenth=0.0, open=0.0, possibl=0.0009633911368015, old=0.0, primit=0.0,
keyser=0.0, fianc=0.0, point=0.0004816955684007, bone=0.0, alreadi=0.0014450867052023, s
traight=0.0, henri=0.0, hack=0.0, rank=0.0, scott=0.0, fold=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=0.0004816955684007, fund=0.0, crush=0.0, accompl
ish=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=0.0014450867052023, rumor=0.0, t
orn=0.0, palei=0.0, well=0.0014450867052023, fear=0.0004816955684007, detroit=0.0, civil
=0.0, current=0.0, advis=0.0, elect=0.0, pill=0.0, artist=0.0, bastaldi=0.0, market=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, givin=0.0, boi=0.0, closest=0.0, mission=0.0009633
911368015, stabil=0.0, identif=0.0, set=0.001926782273603, rebel=0.0, salli=0.0, joint=
0.0, zoo=0.0, version=0.0004816955684007, camper=0.0, directli=0.0, kid=0.00192678227360
3, jim=0.0, viru=0.0, desir=0.0, tide=0.0, cell=0.0, mulwrai=0.0, sod=0.0, domini=0.0, d
ealt=0.0, pane=0.0, trade=0.0, spose=0.0, eleanor=0.0, someth=0.0014450867052023, milk=
0.0, honestli=0.0, sap=0.0, graham=0.0, tune=0.0, secretari=0.0, chao=0.0, backup=0.0, f
reez=0.0004816955684007, obvious=0.0, believ=0.0, delmar=0.0, sting=0.0, happiest=0.0, b
roadcast=0.0, haunt=0.0, haven=0.0, kong=0.0, gonna=0.0024084778420038, without=0.000481
6955684007, lip=0.0, spill=0.0, psychiatrist=0.0, suddenli=0.0, realli=0.000481695568400
7, byeby=0.0, coffe=0.0, opinion=0.0, borrow=0.0, lechter=0.0, cuff=0.0, overload=0.0, a
ttract=0.0, rid=0.0, bertrand=0.0, mordechai=0.0, amanda=0.0, nicknam=0.0, gather=0.0, m
r=0.0004816955684007, miss=0.0, fascin=0.0, hound=0.0, lid=0.0, joi=0.0, weak=0.0, thoug
h=0.0, alfr=0.0, comfort=0.0, ich=0.0, gallon=0.0, narcot=0.0, evelyn=0.0, duck=0.0, buc
ket=0.0, cabin=0.0, settlement=0.0, send=0.0009633911368015, milli=0.0, israel=0.0, alco
hol=0.0, suspect=0.0004816955684007, robinson=0.0, doug=0.0, alan=0.0, sleepi=0.0, adven
tur=0.0, ridicul=0.0, crabtre=0.0, lawyer=0.0, reward=0.0, cloth=0.0, riot=0.0, seek=0.
0, other=0.0009633911368015, pet=0.0, everyth=0.0024084778420038, found=0.00096339113680
15, basketbal=0.0, liar=0.0, wouldnt=0.0, fit=0.0, stop=0.0009633911368015, 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=0.0004816955684007, drove=0.0, turkei=0.0, modern=0.0, edi=0.0, jonah=0.0, e
xagger=0.0, mine=0.0, envi=0.0, big=0.0004816955684007, 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, dict
at=0.0, medicin=0.0, honei=0.0, shine=0.0, comedi=0.0, unou=0.0, who=0.0028901734104046,
late=0.0014450867052023, boost=0.0, gentli=0.0004816955684007, sun=0.0, dunno=0.0, impro
v=0.0, chairman=0.0, honest=0.0, shout=0.0, invas=0.0, hadnt=0.0, fox=0.0, powder=0.0, q
uick=0.0, breaker=0.0, breach=0.0, dope=0.0, frustrat=0.0, cent=0.0, long=0.0, interest=
0.0, freak=0.0, marsh=0.0, maria=0.0, shhhh=0.0, cloak=0.0, solid=0.0, id.1=0.0004816955
684007, nobl=0.0, swing=0.0, evalu=0.0, member=0.0, skunk=0.0, zira=0.0, conduct=0.0, bu
yer=0.0, instanc=0.0, financ=0.0, protest=0.0, karl=0.0, dent=0.0, jake=0.0, goofi=0.0,
au=0.0, plu=0.0, ahead=0.0004816955684007, possess=0.0, prototyp=0.0, whatd=0.0, road=0.
0, fifti=0.0, down=0.0009633911368015, viciou=0.0, destin=0.0, ride=0.0, hmm=0.0, democr
at=0.0, daili=0.0, remov=0.0, catherin=0.0, unlock=0.0, hot=0.0014450867052023, yuh=0.0,
lewi=0.0, concern=0.0, mouth=0.0, didn=0.0, merci=0.0004816955684007, lotta=0.0, sacrifi
c=0.0, chekov=0.0, rock=0.0, instead=0.0, egon=0.0, sank=0.0, probe=0.0, pit=0.0, rest=

0.0004816955684007, confront=0.0, magic=0.0, mean=0.0004816955684007, similar=0.0, itd=
0.0, flesh=0.0004816955684007, bunch=0.0, donut=0.0, soldier=0.0004816955684007, dork=0.
0, short=0.0004816955684007, innoc=0.0, fight=0.0004816955684007, grai=0.0, bout=0.0, be
for=0.001926782273603, encount=0.0, lombardo=0.0, arriv=0.0, date=0.0, worn=0.0, franken
stein=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=0.0226396917148363, resi
d=0.0, salad=0.0, told=0.0004816955684007, brother=0.0, rmph=0.0, which=0.0, philosoph=
0.0, clear=0.0, cracker=0.0, zavitz=0.0, tell=0.0028901734104046, 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, t
orch=0.0, gym=0.0, bang=0.0, man=0.0004816955684007, aid=0.0, purchas=0.0, review=0.0, l
antern=0.0, new=0.0024084778420038, 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=0.00048
16955684007, pitch=0.0, detect=0.0, hood=0.0, relationship=0.0, sunshin=0.0, arlyn=0.0,
gentl=0.0, heard=0.0004816955684007, babe=0.0, beg=0.0004816955684007, princess=0.0, mem
phi=0.0, addict=0.0, incid=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=0.000481695
5684007, weigh=0.0, attic=0.0, incident=0.0, quarantin=0.0, got=0.0043352601156069, napk
in=0.0, chair=0.0, grandmoth=0.0, theft=0.0, disord=0.0, blame=0.0, stronger=0.000481695
5684007, 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=0.0004816955684007, toddi=0.0, gum=0.0, trap=0.0, twel
v=0.0, treatment=0.0, xrai=0.0, callin=0.0, rope=0.0, japan=0.0, pot=0.0, method=0.0, en
ter=0.0, confess=0.0, killer=0.0014450867052023, constantli=0.0, scar=0.0, concert=0.0,
thirteen=0.0, debbi=0.0, knot=0.0, pension=0.0, bob=0.0004816955684007, home=0.000963391
1368015, walt=0.0, troi=0.0, argument=0.0, desper=0.0, jeremi=0.0, epp=0.0, havent=0.000
9633911368015, ugh=0.0, dewei=0.0, dumb=0.0, deputi=0.0, phone=0.0004816955684007, worf=
0.0, spook=0.0, avenu=0.0, hm=0.0, keep=0.0014450867052023, happen=0.0, plane=0.0, carte
l=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, pro
per=0.0, realist=0.0, arthur=0.0, sh=0.0, survivor=0.0004816955684007, teeth=0.0, de=0.
0, sens=0.0, maneuv=0.0, poetri=0.0, overnight=0.0, bow=0.0, dalla=0.0, discharg=0.0, bi
ll=0.0, fly=0.0, realiti=0.0, compar=0.0004816955684007, upstair=0.0, pritchett=0.0, ter
rif=0.0, anoth=0.0009633911368015, kiddin=0.0, unfair=0.0, brace=0.0, chimera=0.0, feath
er=0.0, regul=0.0, hump=0.0, thank=0.0014450867052023, matur=0.0, valu=0.0, even=0.00240
84778420038, shoulder=0.0, booth=0.0, ribbon=0.0, ceas=0.0, base=0.0, dan=0.0, smokei=0.
0, coup=0.0, thick=0.0, bui=0.0004816955684007, stud=0.0, scenario=0.0, laid=0.0, mexica
n=0.0, prior=0.0, bra=0.0, leader=0.0, charli=0.0, fairli=0.0, nurs=0.0, ahhh=0.0, texa=
0.0, lui=0.0, except=0.0004816955684007, august=0.0, sittin=0.0, bait=0.0, power=0.00048
16955684007, kansa=0.0, scotti=0.0, swipe=0.0, videotap=0.0, opposit=0.0, expand=0.0, ne
ighborhood=0.0, out=0.0067437379576108, melt=0.0, difficult=0.0, crow=0.0, occup=0.0, sw
ine=0.0, expens=0.0, brad=0.0, balanc=0.0004816955684007, tension=0.0, smarter=0.0, team
=0.0, salari=0.0, expos=0.0, incom=0.0, special=0.0004816955684007, 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
=0.0004816955684007, singer=0.0, explan=0.0, sollozzo=0.0, beauti=0.0, whole=0.000963391
1368015, attach=0.0, bleed=0.0, card=0.0, longer=0.0, good=0.0024084778420038, draw=0.0,
rate=0.0, enlighten=0.0, startin=0.0, demon=0.0, street=0.0, thatll=0.0, convers=0.0, ne
rv=0.0, telegram=0.0, therer=0.0004816955684007, stan=0.0, momma=0.0, nightmar=0.0, comb
at=0.0004816955684007, brand=0.0, greatest=0.0, color=0.0, sourc=0.0, swap=0.0, preciou=
0.0, greas=0.0, manipul=0.0, blood=0.0, batman=0.0, hitler=0.0, visit=0.0, india=0.0, po
ke=0.0, baltimor=0.0, tear=0.0, fate=0.0009633911368015, cours=0.0004816955684007, 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=0.0009633911368015, 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=0.0004816955684
007, 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, susi=0.0, brodi=0.0, far=0.0, sweater=0.0, appeal=0.
0, vibe=0.0, poet=0.0, oz=0.0, finch=0.0, busi=0.0004816955684007, perfect=0.0, boyfrien
d=0.0, lemon=0.0, hand=0.0, charm=0.0, sustain=0.0, retain=0.0, continu=0.0, hammer=0.0,
ripper=0.0, personnel=0.0, beaten=0.0, remind=0.0, joan=0.0, witch=0.0, second=0.0014450
867052023, impress=0.0, hint=0.0, avoid=0.0, billi=0.0, wipe=0.0004816955684007, calvin=

0.0, shrink=0.0, darn=0.0, is=0.0077071290944123, show=0.0004816955684007, first=0.00144
50867052023, intellig=0.0, kit=0.0, collar=0.0, bundi=0.0, feel=0.001926782273603, marth
a=0.0, fortyeight=0.0, franki=0.0, rehears=0.0, reunion=0.0, previou=0.0, fire=0.0004816
955684007, specimen=0.0, exactli=0.0, gotta=0.0009633911368015, claric=0.0, you=0.039017
3410404626, psychot=0.0, liberti=0.0, ever=0.0009633911368015, indian=0.0, karen=0.0, ea
r=0.0, uncl=0.0, mar=0.0, surround=0.0004816955684007, 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=0.0004816955684007, ag
re=0.0, purs=0.0, puppet=0.0, clearli=0.0, ellen=0.0, serious=0.0, applejack=0.0, threat
en=0.0, swear=0.0, journalist=0.0, beth=0.0, feelin=0.0, danni=0.0, hmmm=0.0, stomach=0.
0, certif=0.0, them=0.0024084778420038, wigand=0.0, electron=0.0, surfac=0.0, op=0.0, ed
it=0.0, backward=0.0, buckaroo=0.0, fare=0.0, mhm=0.0, cynic=0.0, fee=0.0, blown=0.0, wo
rkin=0.0, eric=0.0, gear=0.0, chill=0.0, public=0.0004816955684007, elizabeth=0.0, dy=0.
0004816955684007, great=0.0009633911368015, load=0.0, melani=0.0, unless=0.0, mora=0.0,
nativ=0.0, gene=0.0, wind=0.0004816955684007, 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=0.0004816955684007, 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=0.0043352601156069, vinc=0.0, notion=0.0, consid=0.0, upset=0.0, ve=0.0, yeah=0.000
4816955684007, compet=0.0, ticket=0.0, doom=0.0, ski=0.0, courthous=0.0, alert=0.0, tan=
0.0, nam=0.0, posit=0.0, urgent=0.0, tast=0.0, makeup=0.0, parol=0.0, schwartz=0.0, doct
or=0.0009633911368015, 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=0.0009633911368015, notic
=0.0, heali=0.0, benjamin=0.0, octob=0.0, ought=0.0, isnt=0.0004816955684007, drew=0.0,
whether=0.0, seat=0.0, matter=0.0, er=0.0, int=0.0, arent=0.0004816955684007, aubrei=0.
0, lookout=0.0, gabe=0.0, cliff=0.0, hail=0.0, horribl=0.0, heaven=0.0, hospit=0.0, what
soev=0.0, supernatur=0.0, lot=0.0004816955684007, establish=0.0, select=0.0, motor=0.0,
brenda=0.0, guilt=0.0, glori=0.0, move=0.0009633911368015, rage=0.0, decent=0.0, religio
n=0.0, pillow=0.0, sincer=0.0, steven=0.0, winston=0.0, arm=0.0, system=0.00048169556840
07, score=0.0, isn=0.0, wallac=0.0, lane=0.0, burger=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, tho
ught=0.0004816955684007, troop=0.0, forti=0.0009633911368015, highest=0.0, dust=0.0, beg
in=0.0004816955684007, lapd=0.0, priest=0.0, turk=0.0, hawaii=0.0, amateur=0.0, honor=0.
0009633911368015, ap=0.0, lodg=0.0, lombard=0.0, academi=0.0, gibson=0.0, injuri=0.0, wh
a=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, banana=0.0, speed
=0.0, piec=0.0, born=0.0009633911368015, oughta=0.0, hacker=0.0, stream=0.0, smith=0.0,
choir=0.0, variou=0.0, weed=0.0, boundari=0.0, panti=0.0, accord=0.0, best=0.0, condom=
0.0, whatll=0.0, carlo=0.0, promis=0.0, barrett=0.0, pentagon=0.0, sooner=0.0, nexu=0.0,
bloodi=0.0, said=0.0014450867052023, mississippi=0.0, legitim=0.0, syndrom=0.0, repli=0.
0, purpl=0.0, break=0.0, wash=0.0, motel=0.0, wrap=0.0, fifteen=0.0, sell=0.0, bravo=0.
0, world=0.0004816955684007, beaumont=0.0, switch=0.0, adrian=0.0, after=0.0009633911368
015, vanish=0.0, window=0.0, interfer=0.0, countri=0.0004816955684007, wretch=0.0, sugar
=0.0, captur=0.0004816955684007, wed=0.0014450867052023, adrenalin=0.0, dorothi=0.0, jud
e=0.0, gorgeou=0.0, corneliu=0.0, flip=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, lovebird=
0.0, express=0.0, dutch=0.0, scare=0.0, kept=0.0, stephani=0.0, signatur=0.0, warren=0.
0, battl=0.0, insight=0.0, next=0.0004816955684007, 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=0.0115606936
416185, everyon=0.0, kidnei=0.0, hostil=0.0, pie=0.0, written=0.0, bye=0.0, refer=0.0, e
xplos=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, among=0.0, disgust=0.0, pocket=0.0, margo=0.0, i
gnor=0.0, goodlook=0.0, claim=0.0, approach=0.0, fruit=0.0, raid=0.0, phrase=0.0, luck=
0.0, kevin=0.0, autograph=0.0, core=0.0, download=0.0, evil=0.0, strand=0.0, woulda=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, dr
ill=0.0, preserv=0.0, serial=0.0, option=0.0, oldest=0.0, pattern=0.0004816955684007, he
rb=0.0, row=0.0, ball=0.0, equip=0.0004816955684007, drift=0.0, mike=0.0, tyler=0.0, cho
os=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=0.0014450867052023,
eyebal=0.0, carv=0.0, overlook=0.0, contractor=0.0, languag=0.0, fortyf=0.0, admit=0.0,
slight=0.0, popular=0.0, compliment=0.0, del=0.0, listen=0.0014450867052023, mess=0.0, f
ree=0.0, keen=0.0, noth=0.0014450867052023, acquir=0.0, earlier=0.0, bluff=0.0, twentyfo

ur=0.0, steak=0.0, california=0.0, dial=0.0, industri=0.0, hopefulli=0.0, probli=0.0, ce
rtainli=0.0, magnet=0.0, mobil=0.0, elvi=0.0, parlor=0.0, utah=0.0, we=0.006743737957610
8, 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=0.0028901734104046, grenad=0.0, phase=0.00048169
55684007, fabul=0.0, diplomat=0.0, wrong=0.0009633911368015, itll=0.0014450867052023, ag
ain=0.0009633911368015, dock=0.0, bear=0.0, minor=0.0, white=0.0004816955684007, environ
=0.0, patient=0.0, mug=0.0, lighthous=0.0, upsid=0.0, nasal=0.0, navig=0.0, sad=0.0, res
earch=0.0, case=0.0014450867052023, gu=0.0, brutal=0.0, aint=0.0, sinc=0.0, nichola=0.0,
intrud=0.0, dare=0.0, were=0.0038535645472061, import=0.0004816955684007, pi=0.0, babysi
t=0.0, race=0.0, oh=0.0004816955684007, flag=0.0, shaw=0.0, teacher=0.0, pimp=0.0, diego
=0.0, victim=0.0, anymor=0.0004816955684007, round=0.0004816955684007, 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,
maud=0.0, monster=0.0, ic=0.0, or=0.0024084778420038, gener=0.0004816955684007, penguin=
0.0, earth=0.0, dc=0.0, cave=0.0, fleet=0.0, wrote=0.0, goddammit=0.0, venic=0.0, wade=
0.0, spoken=0.0, unfortun=0.0, campaign=0.0, fever=0.0, marcu=0.0, almost=0.000481695568
4007, these=0.0004816955684007, sick=0.0, reach=0.0004816955684007, screw=0.000963391136
8015, grace=0.0, anyth=0.0014450867052023, empir=0.0, rack=0.0, nell=0.0, divin=0.0, vio
let=0.0, sing=0.0, friendship=0.0, loomi=0.0, basebal=0.0, stood=0.0, clark=0.0, bottl=
0.0, marylin=0.0, sin=0.0, juli=0.0, halfwai=0.0, spanish=0.0, hannah=0.0, shower=0.0004
816955684007, murphi=0.0, pal=0.0004816955684007, why=0.0024084778420038, leagu=0.0, our
=0.0004816955684007, die=0.0004816955684007, glad=0.0, depart=0.0004816955684007, mightv
=0.0, deborah=0.0, spent=0.0, heather=0.0, grant=0.0, delici=0.0, bloom=0.0, halloween=
0.0, selfish=0.0, mack=0.0, pleas=0.0004816955684007, later=0.0004816955684007, tomb=0.
0, veri=0.0009633911368015, al=0.0, filthi=0.0, bein=0.0, cream=0.0, invit=0.0, ay=0.0,
mountain=0.0, tax=0.0, liquid=0.0, suffici=0.0, start=0.0009633911368015, particl=0.0, t
aught=0.0004816955684007, worthless=0.0, caitlin=0.0, paperwork=0.0, duffi=0.0, fade=0.
0, lothar=0.0, courag=0.0004816955684007, puls=0.0, fighter=0.0004816955684007, galleri=
0.0, restaur=0.0, sky=0.0, hollywood=0.0, accept=0.0, trip=0.0, grid=0.0004816955684007,
vigo=0.0, cellar=0.0, urg=0.0, recal=0.0, task=0.0, nose=0.0, circul=0.0, sperm=0.0, exp
eri=0.0, rust=0.0, thirtyseven=0.0, lame=0.0, editor=0.0, univers=0.0, children=0.0, slu
g=0.0, ration=0.0, adam=0.0, mailbox=0.0, twentytwo=0.0, reveal=0.0, eat=0.0, tobacco=0.
0, schuyler=0.0, widow=0.0, forgiv=0.0, punch=0.0, transfer=0.0004816955684007, humor=0.
0, thrill=0.0, jewish=0.0, elbow=0.0, teach=0.0, flush=0.0, pace=0.0, copi=0.0, locat=0.
0, senior=0.0, mantan=0.0, auto=0.0, chemistri=0.0, bolt=0.0, steadi=0.0, kick=0.0, ego=
0.0, underneath=0.0, approv=0.0, cloud=0.0, studi=0.0, hunter=0.0004816955684007, clue=
0.0, roof=0.0, french=0.0, billion=0.0, streak=0.0, chauncei=0.0, judi=0.0, cost=0.0, ca
mpbel=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=0.0057803468208092, decis=0.0, gardin=0.0,
miami=0.0, dark=0.0004816955684007, 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, uar
eu=0.0, inde=0.0, maximum=0.0, vernon=0.0, hide=0.0009633911368015, hampshir=0.0, put=0.
001926782273603, shh=0.0, program=0.0, plan=0.0, scratch=0.0, ban=0.0, adopt=0.0, ninth=
0.0, gui=0.001926782273603, physic=0.0, enhanc=0.0, dead=0.0009633911368015, 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=0.0004816955684007, result=0.0, bishop=0.0, crim
e=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,
susan=0.0, whoever=0.0, know=0.0057803468208092, actual=0.0, war=0.0028901734104046, 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=0.0004816955684007, vote=0.0, betcha=0.0, skywir=0.0, donov
an=0.0, societi=0.0, uncomfort=0.0, commit=0.0, mai=0.0004816955684007, april=0.0, purpo
s=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, uthisu=0.0, sock=0.0, dawson=0.0, museum=0.0, vacuum=0.0, leap=0.
0, romulan=0.0, soviet=0.0, freewai=0.0, half=0.0, profit=0.0, iri=0.0, real=0.0, client
=0.0, an=0.0009633911368015, dynamit=0.0, eugen=0.0, nuke=0.0, robin=0.0, rub=0.0, glori
a=0.0, becam=0.0, deceiv=0.0, distract=0.0, matthew=0.0, ken=0.0, where=0.00289017341040
46, america=0.0, bela=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=0.0033718689788054, trash=0.0, sherri=0.0, s
cript=0.0, join=0.0, clock=0.0, allison=0.0, confirm=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=0.0004816955684007, frederick=0.0, perfe

ctli=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, mentio
n=0.0, cant=0.0048169556840077, pain=0.0014450867052023, tri=0.0, ok=0.0, crisi=0.0, jos
e=0.0, would=0.0004816955684007, beach=0.0004816955684007, size=0.0, angel=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, ove
rwhelm=0.0, fast=0.0, intens=0.0, translat=0.0, sucker=0.0, etern=0.0, still=0.000963391
1368015, beneath=0.0, funer=0.0, immedi=0.0, prai=0.0, theatr=0.0, ceremoni=0.0, pregnan
t=0.0, capac=0.0, dya=0.0, killain=0.0, maroon=0.0, sammi=0.0, gordon=0.0, bother=0.0, h
ah=0.0, cartman=0.0, crude=0.0, bedroom=0.0, mustang=0.0, offer=0.0, path=0.0, doyl=0.0,
viktor=0.0, iim=0.0, buff=0.0, disk=0.0, birthdai=0.0, kilo=0.0, lap=0.0, am=0.000963391
1368015, poem=0.0, least=0.0004816955684007, pick=0.0, pad=0.0, slowli=0.000481695568400
7, garag=0.0, person=0.0004816955684007, permit=0.0, further=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, rol
l=0.0, collect=0.0, chat=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, camera=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, eighti=0.0, eras=0.0, digniti=0.0, swim=0.0, ly=0.0, s
tar=0.0, ye=0.0024084778420038, 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, jaeger=0.0, silent=0.
0, fact=0.0, ni=0.0, tub=0.0, bag=0.0004816955684007, 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=0.0038535645472061, gabriel=0.
0, expertis=0.0, secret=0.0, cherri=0.0, crippl=0.0, cross=0.0, silver=0.0, thief=0.0, w
hip=0.0, insan=0.0, riplei=0.0, foreign=0.0, ted=0.0, glimps=0.0, lebowski=0.0, me=0.012
5240847784201, peopl=0.0004816955684007, 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=0.000
4816955684007, japanes=0.0, fuse=0.0004816955684007, lucki=0.0, mous=0.0, wherer=0.0, po
wel=0.0, vanessa=0.0, inherit=0.0, meat=0.0, sang=0.0, bert=0.0, scale=0.0, scam=0.0, se
rvic=0.0, porno=0.0, spring=0.0, storag=0.0, small=0.0, thing=0.001926782273603, tragedi
=0.0, makin=0.0, just=0.0072254335260116, costum=0.0, averag=0.0, experienc=0.0, click=
0.0, lose=0.0, folk=0.0, danger=0.0004816955684007, mayflow=0.0, more=0.001926782273603,
view=0.0004816955684007, jasper=0.0, hostag=0.0004816955684007, swamp=0.0, cup=0.0, hatc
h=0.0, echo=0.0, cindi=0.0, buddyboi=0.0, depend=0.0, rang=0.0004816955684007, harold=0.
0, instrument=0.0, deeper=0.0, beam=0.0, fish=0.0, writer=0.0, stir=0.0, penelop=0.0, re
act=0.0, knive=0.0, mafia=0.0, carter=0.0, whod=0.0, lincoln=0.0, swell=0.0, boat=0.0, c
ertain=0.0, jet=0.0, steal=0.0004816955684007, safer=0.0, larger=0.0, magazin=0.0, edgar
=0.0, pittsburgh=0.0, committe=0.0, truth=0.0, green=0.0, over=0.0028901734104046, preve
nt=0.0004816955684007, impati=0.0, tv=0.0, ian=0.0, alarm=0.0, boom=0.0, tripp=0.0, base
ment=0.0, torpedo=0.0, such=0.0004816955684007, sir=0.0, whatr=0.0, jump=0.0, shari=0.0,
kenni=0.0, oti=0.0, sweat=0.0004816955684007, spine=0.0, everybodi=0.0, salon=0.0, asian
=0.0, miser=0.0, church=0.0, trust=0.0009633911368015, peac=0.0, surveil=0.0, understand
=0.0024084778420038, picnic=0.0, skipper=0.0, bodi=0.0004816955684007, itself=0.0, menta
l=0.0, come=0.0028901734104046, 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=
0.0158959537572255, gosh=0.0, wynant=0.0, hopkin=0.0, cough=0.0, butch=0.0, agn=0.0, wic
hita=0.0, tall=0.0004816955684007, sent=0.0004816955684007, manner=0.0, artifici=0.0, du
mper=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=0.0009633911368015, marshal=0.0, deaf=0.0, stretch=0.000481695568400
7, 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, wall=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, childhood=
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, lawren
c=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, sailor=0.0, circl=0.0, pursu=0.0, notifi=0.0, creasi=0.0, sum=0.0, restrict=0.0, s
hield=0.0, rocco=0.0, asleep=0.0, failur=0.0, memor=0.0004816955684007, mon=0.0, cool=0.
0, emma=0.0, shape=0.0, state=0.0, discount=0.0, youll=0.001926782273603, repeat=0.00048
16955684007, neither=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, monsie
ur=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, corrido
r=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=0.0004816955684007, sayer=0.0, skin=0.0009633911368

```
015, mortal=0.0, add=0.0, honeymoon=0.0, receipt=0.0, nun=0.0, usual=0.0, starv=0.000481
6955684007, eight=0.0, morgu=0.0, comin=0.0, style=0.0, cadet=0.0, uh=0.0, uhuh=0.0, ear
n=0.0, tini=0.0, volunt=0.0009633911368015)
```

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 [14]:
```
row_for_title('the terminator').item('none')
```

Out[14]:
```
0.0009633911368015
```

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.

In [15]:
```
# Set row_sum to a number that's the (approximate) sum of each row of word proportions.
expected_row_sum = 1
```

In [16]:
```
ok.grade("q1_0");
```

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

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

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

In [17]:
```
print('Words with frequencies:', movies.drop(np.arange(5)).num_columns)
print('Movies with genres:', movies.num_rows)
```

```
/opt/conda/lib/python3.9/site-packages/datascience/tables.py:1252: FutureWarning: elemen
twise comparison failed; returning scalar instead, but in the future will perform elemen
twise comparison
  if i not in exclude and c not in exclude])
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 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 [19]:
```
# 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[19]:

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".

In [20]:
```
stemmed_message = vocab_table.where("Word","vegetables").column(1).item(0)
stemmed_message
```

Out[20]:
```
'vegetables'
```

In [21]:
```
ok.grade("q1_1_1");
```

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

---------------------------------------------------------------------
q1_1_1 > Suite 1 > Case 2

>>> len(stemmed_message) < len('message')
False

# Error: expected
#     True
# but got
#     False

Run only this test case with "python3 ok -q q1_1_1 --suite 1 --case 2"
---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 1
[ooook.....] 50.0% passed
```

Question 1.1.2

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

In [30]:
```
shorted_stem=vocab_table.group("Stem")
most_stem=shorted_stem.sort("count",descending=True).column(0).item(0)
most_stem
```

Out[30]:
```
'gener'
```

In [31]:
```
ok.grade("q1_1_2");
```

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

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

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").

In [32]:
```
# 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?
Length_word= vocab_table.apply(len,"Word")
Length_stem= vocab_table.apply(len,"Stem")
tbl_with_lens = vocab_table.with_columns("Word Length",Length_word,"Stem Length",Length_
tbl_with_dif = tbl_with_lens.with_columns("Length Difference", tbl_with_lens.column("Wor


longest_uncut = tbl_with_dif.where("Length Difference",0).sort("Word Length", descending
longest_uncut
```

Out[32]:
```
'extraordinary'
```

In [33]:
```
ok.grade("q1_1_3");
```

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

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

1.2. Exploratory Data Analysis: Linear Regression

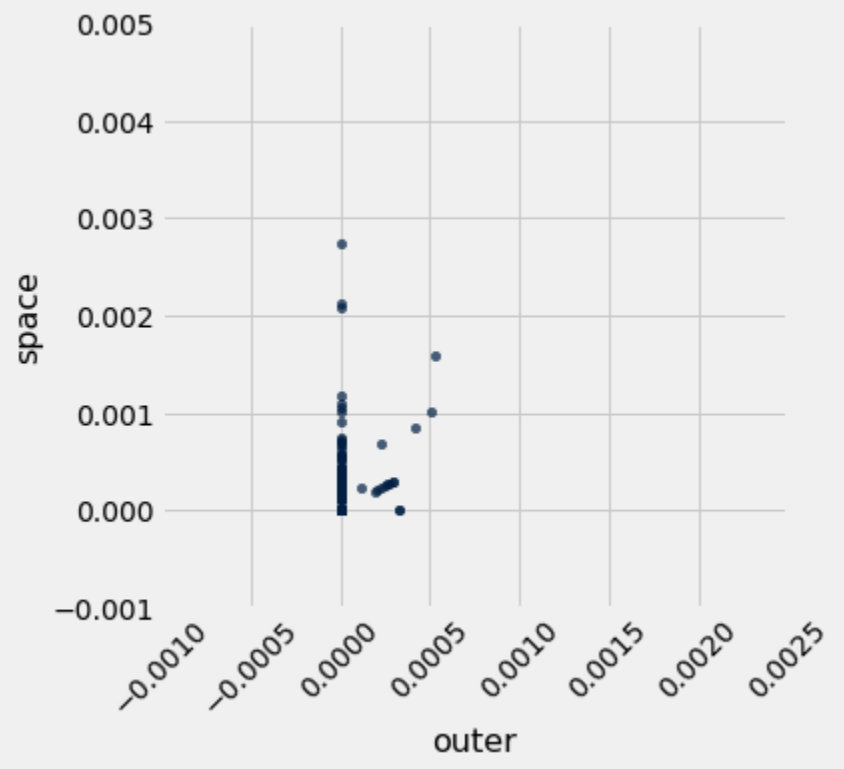
Let's explore our dataset before trying to build a classifier. To start, we'll look at the relationship between words in proportions.

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 [34]:
```
# 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`.

In [35]:
```
# 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)
space_su = (space-np.mean(space))/np.std(space)

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

Out[35]:
```
0.2829527833012746
```

In [36]:
```
ok.grade("q1_2_1");
```

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

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

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`.

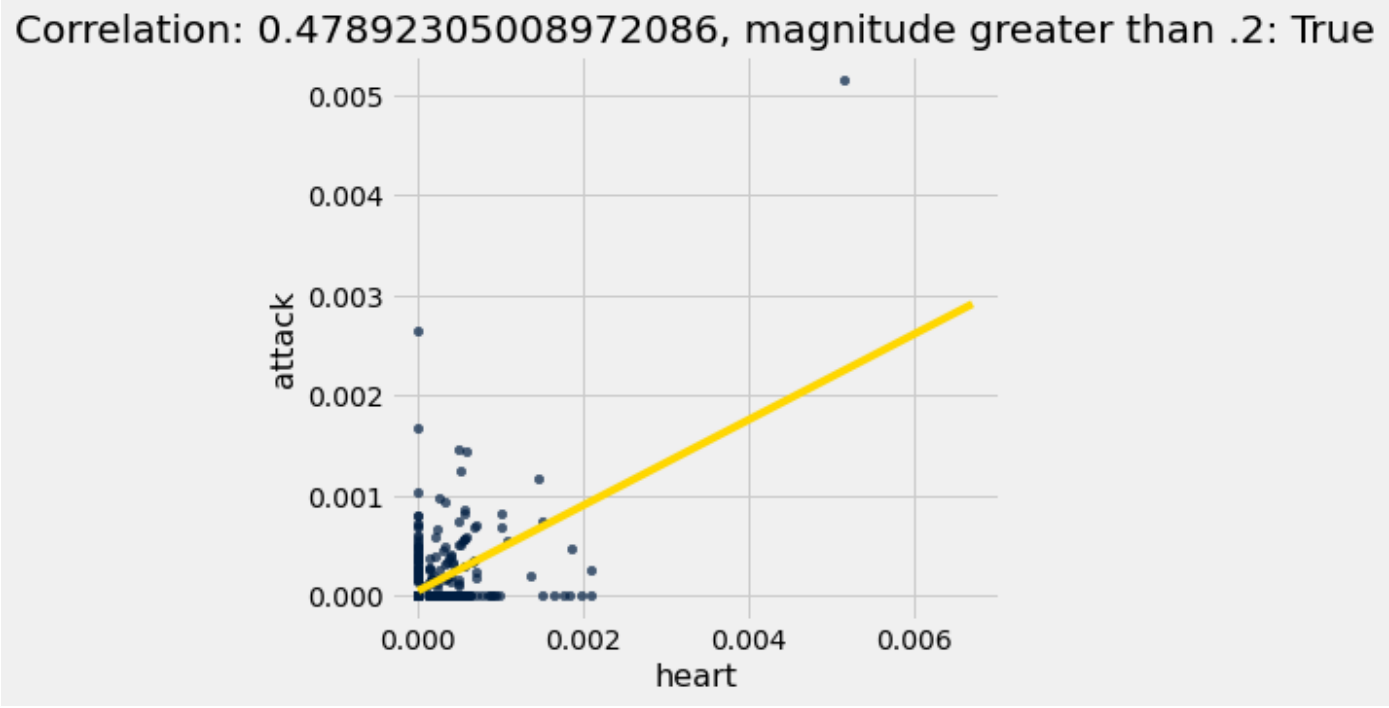
Hint: It's easier to think of words with a positive correlation, i.e. words that are often mentioned together.

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

In [37]:
```
def standard_units(any_numbers):
    "Convert any array of numbers to standard units."
    return (any_numbers - np.mean(any_numbers))/np.std(any_numbers)
def correlation(t, x, y):
    return np.mean(standard_units(t.column(x))*standard_units(t.column(y)))
correlation(movies,"san","heart")
```

Out[37]:
```
-0.03404164172227834
```

In [38]:
```
word_x = "heart"
word_y = "attack"

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

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

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

# 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 [39]:
```
# 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.

In [40]:
```
def comedy_proportion(table):
    # Return the proportion of movies in a table that have the Comedy genre.
    comedy_movies= table.where("Genre",are.equal_to("comedy")).num_rows
    return comedy_movies/table.num_rows

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

Out[40]:
```
0.35987261146496813
```

In [41]:
```
make_array_comedy=make_array(comedy_proportion(train_movies),comedy_proportion(test_movi
make_array_comedy
```

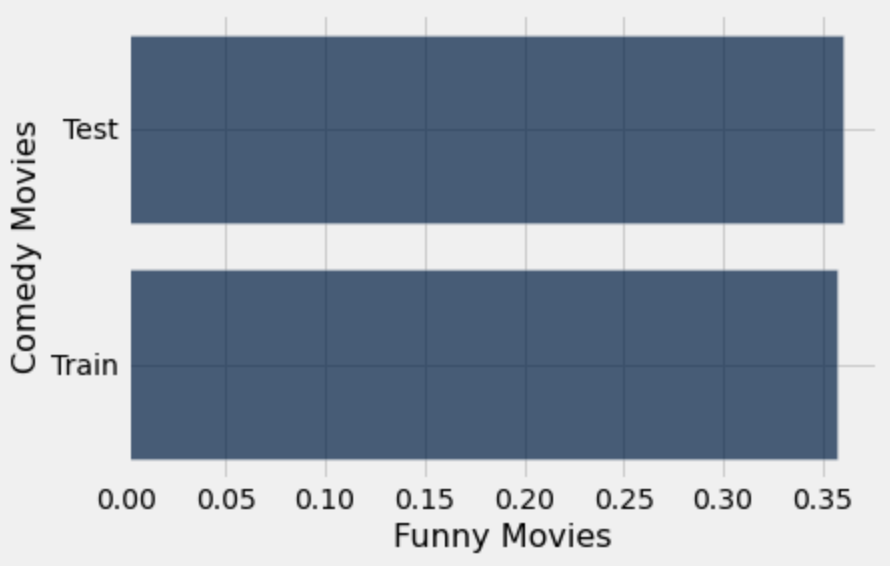
Out[41]:
```
array([0.35987261, 0.35714286])
```

In [42]:
```
comedy_movies_table= Table().with_columns( "Comedy Movies", make_array("Test","Train"),
                                           "Funny Movies",make_array_comedy)
comedy_movies_table
```

Out[42]:

Comedy Movies	Funny Movies
Test	0.359873
Train	0.357143

In [43]:
```
comedy_movies_table.barh("Comedy Movies","Funny Movies")
```



In [44]:
```
movies.where("Genre",are.equal_to("comedy")).num_rows
```

Out[44]:
```
133
```

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 $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

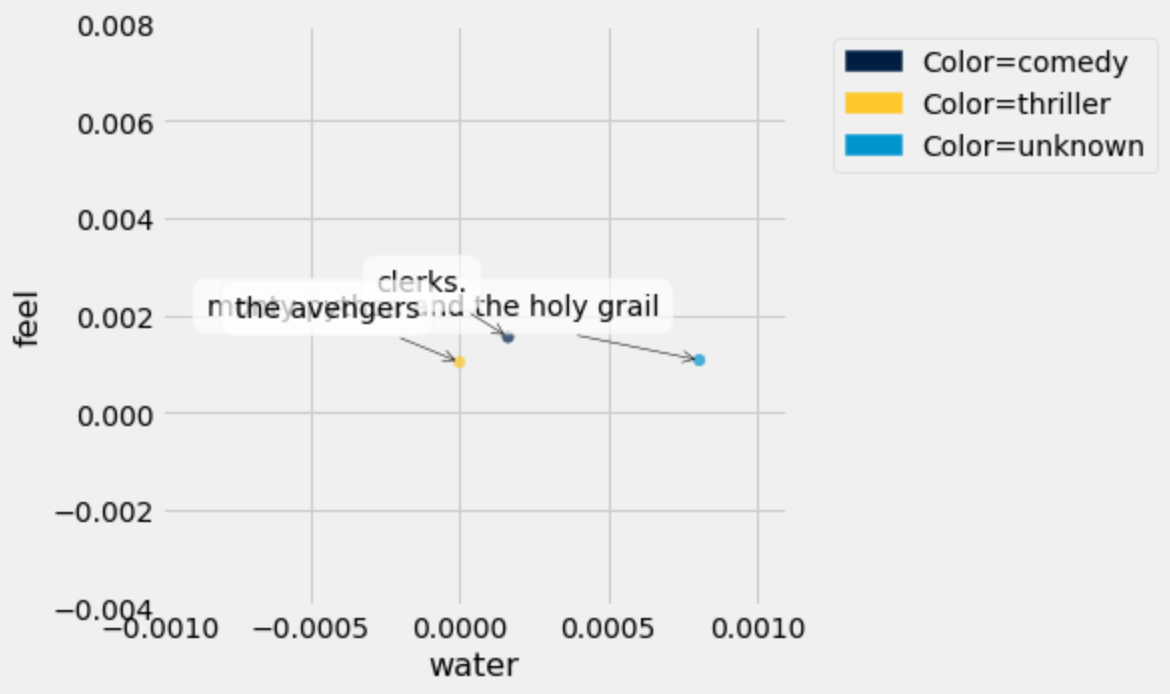
For example, in the movie *Clerks.* (in the training set), 0.00016293 of all the words in the movie are "water" and 0.00154786 are "feel". Its distance from *Monty Python and the Holy Grail* on this 2-word feature set is $\sqrt{(0.000804074 - 0.000162933)^2 + (0.0010721 - 0.00154786)^2} \approx 0.000798379$. (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 [45]:
```
# 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'), m
    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.

In [46]:
```
title_index = movies.index_by('Title')
python = row_for_title("monty python and the holy grail")
avengers = row_for_title("the avengers")

one_distance = np.sqrt((python.item("feel")-avengers.item("feel"))**2 + (python.item("wa
                                                                          )**2)
one_distance
```

Out[46]:
```
0.0008050869157478146
```

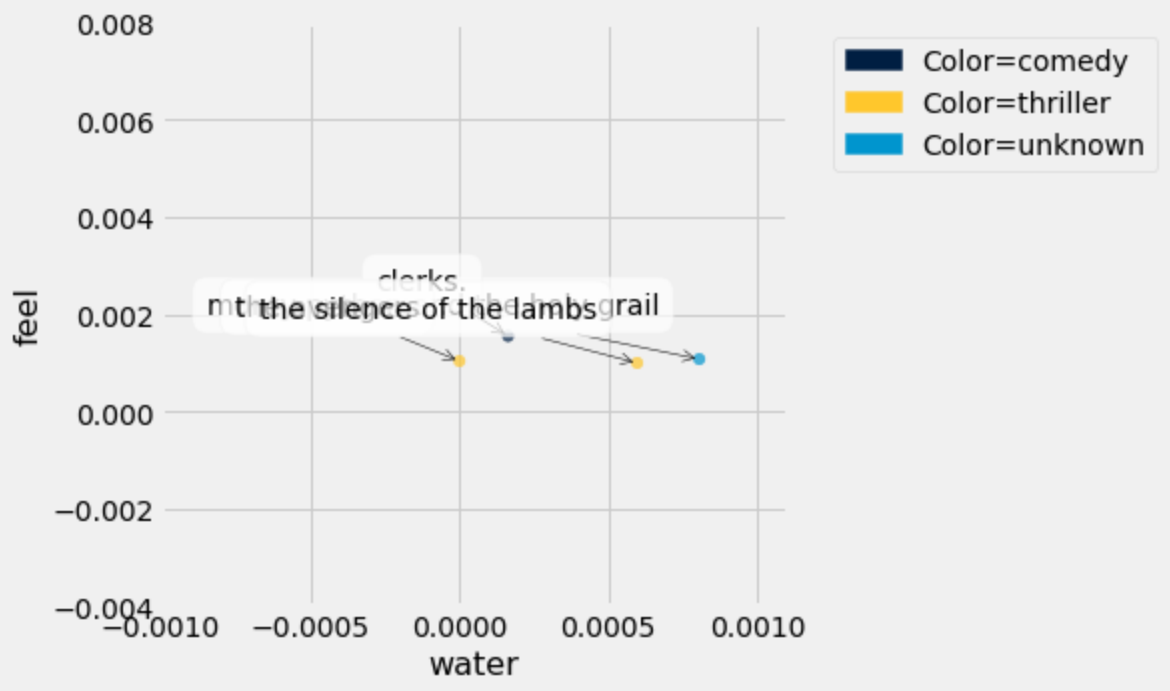
In [47]:
```
ok.grade("q2_1_1");
```

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

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

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 [48]:
```
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*.

In [49]:
```
def distance_two_features(title0, title1, x_feature, y_feature):
    """Compute the distance between two movies with titles title0 and title1

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

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

```
clerks. distance:        0.0007983810687227716
the silence of the lambs distance:       0.00022256314855564847
```

In [50]:
```
ok.grade("q2_1_2");
```

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

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

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!

In [51]:
```
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.
    """
    python = row_for_title("monty python and the holy grail")
    row1= row_for_title(title)
    return  np.sqrt((python.item("water")-(row1.item("water")))**2 + (python.item("feel"
```

In [52]:
```
ok.grade("q2_1_3");
```

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

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

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`**.

In [53]:
```
# The staff solution took multiple lines.
train_set= train_movies.select("Title","Genre","water","feel")
train_movies_distance= train_set.apply(distance_from_python,"Title")
movies_table_train= train_set.with_columns("distance from python",train_movies_distance)
close_movies = movies_table_train.sort("distance from python").take(np.arange(0,5))
close_movies
```

Out[53]:

Title	Genre	water	feel	distance from python
alien	thriller	0.00070922	0.00124113	0.000193831
tomorrow never dies	thriller	0.000888889	0.000888889	0.00020189
the silence of the lambs	thriller	0.000595948	0.000993246	0.000222563
innerspace	comedy	0.000522193	0.00104439	0.00028324
some like it hot	comedy	0.000528541	0.000951374	0.00030082

In [54]:
```
ok.grade("q2_1_4");
```

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

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

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.

In [55]:
```
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(label).item(0)
# 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[55]:
```
'thriller'
```

In [56]:
```
ok.grade("q2_1_5");
```

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

---------------------------------------------------------------------
q2_1_5 > Suite 1 > Case 2

>>> [most_common('Genre', close_movies.take(np.arange(4, k, -1))) for k in range(3, -1,
-1)]
['comedy', 'comedy', 'comedy', 'comedy']

# Error: expected
#     ['comedy', 'comedy', 'comedy', 'thriller']
# but got
#     ['comedy', 'comedy', 'comedy', 'comedy']

Run only this test case with "python3 ok -q q2_1_5 --suite 1 --case 2"
---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 1
[ooook.....] 50.0% passed
```

Congratulations are in order -- you've classified your first movie! However, we can see that the classifier doesn't work too well since it categorized *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.

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 five 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))`

In [58]:
```
def distance(features_array1, features_array2):
    """The Euclidean distance between two arrays of feature values."""
    return np.sqrt(sum((features_array1 - features_array2)**2))

distance_first_to_second = distance(np.array(train_movies.drop(np.arange(5)).row(0)
                                             ),np.array(train_movies.drop(np.arange(5)).r
distance_first_to_second
```

Out[58]:
```
0.033354468908813176
```

In [59]:
```
ok.grade("q3_0");
```

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

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

3.1. Creating your own feature set

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

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

Question 3.1.1

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

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.

In [60]:
```
bottom_left = 4
```

In [61]:
```
ok.grade("q3_1_1");
```

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

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

Question 3.1.2

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

In [62]:
```
bottom_right = 3
```

In [63]:
```
ok.grade("q3_1_2");
```

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

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

Question 3.1.3

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

In [64]:
```
top_right = 1
```

In [65]:
```
ok.grade("q3_1_3");
```

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

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

Question 3.1.4

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

In [66]:
```
top_left = 2
```

In [67]:
```
ok.grade("q3_1_4");
```

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

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

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.

In [68]:
```
movie_genre_guess = 2
```

In [69]:
```
ok.grade("q3_1_5");
```

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

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

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.

In [70]:
```
# Set my_20_features to an array of 20 features (strings that are column labels)

my_features = make_array("jake","bobbi","price","file","marri","murder","frank","captain
                         "gun","knew","great","dead","kill","down","cop","ship","oh","wh

# 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 [71]:
```
ok.grade("q3_1_6");
```

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

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

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

Question 3.1.7

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

on the other side of the best fit line

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 [72]:
```
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:

jake	bobbi	price	file	marri	murder	frank	captain	gun	knew	great	dead	kill	
0	0	0	0	0	0	0	0	0	0	0.00109369	0.000364564	0.000729129	0.00

In [ ]:

```
```

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 [73]:

```
# 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 all the features of
    counts_matrix = np.asmatrix(train_table.columns).transpose()
    diff = np.tile(np.array(list(test_row)), [counts_matrix.shape[0], 1]) - counts_matri
    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**.

In [103…

```
# The staff solution took multiple lines of code.
genre_and_distances = Table().with_columns(
                        "Genre",train_movies.column("Genre"),
                         "Distance",fast_distances(test_my_features.row(0),train_my_featu
genre_and_distances
```

Out[103]:

Genre	Distance
thriller	0.00116602
comedy	0.00117851
comedy	0.00125727
comedy	0.00150922
comedy	0.00152487
thriller	0.00154288
comedy	0.00158206
comedy	0.00164574
comedy	0.00175433
thriller	0.00178706

... (304 rows omitted)

In [76]:
```
ok.grade("q3_1_8");
```

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

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

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.)

In [ ]:
```
# Set my_assigned_genre to the most common genre among these.
my_assigned_genre = most_common("Genre",genre_and_distances.take(np.arange(7)))

# 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 = most_common("Genre",genre_and_distances.take(np.arange(7

print("The assigned genre, {}, was{}correct.".format(my_assigned_genre, " " if my_assign
```

In [78]:
```
ok.grade("q3_1_9");
```

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

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

3.2. A classifier function

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

Question 3.2.1

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

- A row of features for a movie to classify (e.g., `test_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'`).

In [79]:
```
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",descending=False).take(np.
    nearest_classifer= genre_and_distances.group("Genre").sort("count",descending=True).
    return nearest_classifer
```

In [80]:
```
classify(test_my_features.row(0),train_my_features,train_movies.column("Genre"),5)
```

Out[80]:
```
'comedy'
```

In [81]:
```
ok.grade("q3_2_1");
```

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

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

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.

In [82]:
```
test_my_features.row(0)
```

Out[82]:
```
Row(jake=0.0, bobbi=0.0, price=0.0, file=0.0, marri=0.0, murder=0.0, frank=0.0, captain=
0.0, gun=0.0, knew=0.0, great=0.001093693036821, dead=0.000364564345607, kill=0.00072912
8691214, down=0.001458257382428, cop=0.000729128691214, ship=0.0, oh=0.001458257382428,
where=0.001458257382428, their=0.000729128691214, question=0.0)
```

In [106…
```
# The staff solution first defined a row called king_kong_features.
tron_features = test_movies.where("Title",are.equal_to("tron")).select(my_features).row(
tron_genre = classify(tron_features,train_my_features,train_movies.column("Genre"),13)
tron_genre
```

Out[106]: 'thriller'

In [107…
```
ok.grade("q3_2_2");
```

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

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

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

Question 3.2.3

Create a classification function that takes as its argument a row containing your 20 features and classifies that row using the 13-nearest neighbors algorithm with `train_20` as its training set.

In [85]:
```
classify(test_my_features.row(0),train_my_features,train_movies.column("Genre"),5)
```

Out[85]: 'comedy'

In [86]:
```
def classify_feature_row(row):
    return classify(row,train_my_features, train_movies.column("Genre"), 13)

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

Out[86]: 'comedy'

In [87]:
```
ok.grade("q3_2_3");
```

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

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

3.3. Evaluating your classifier

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

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

In [88]:
```
test_guesses = test_my_features.apply(classify_feature_row)
proportion_correct = np.count_nonzero(test_guesses==test_movies.column("Genre"))/test_mo
proportion_correct
```

Out[88]: 0.7678571428571429

In [89]:
```
ok.grade("q3_3_1");
```

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

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

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

In [90]:
```
# Feel free to use multiple lines of code
# but make sure to assign test_movie_correctness to the proper table!
test_movie_correctness = Table().with_columns("Title",test_movies.column("Title"),
                                              "Genre", test_movies.column("Genre"),
                                              "Was correct",test_guesses==test_movies.col
test_movie_correctness.sort('Was correct', descending = True).show(5)
```

Title	Genre	Was correct
the body snatcher	thriller	True
rear window	thriller	True
u turn	thriller	True
jason goes to hell: the final friday	thriller	True
the crow: salvation	thriller	True

... (51 rows omitted)

In [108…
```
ok.grade("q3_3_2");
```

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

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

In [109…
```
test_movie_correctness.where("Was correct",True).group("Genre")
```

Out[109]:

Genre	count
comedy	14
thriller	29

In [94]:
```
test_movie_correctness.pivot("Genre","Was correct")
```

```
/opt/conda/lib/python3.8/site-packages/datascience/tables.py:920: VisibleDeprecationWarn
ing: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists
-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to
do this, you must specify 'dtype=object' when creating the ndarray
  values = np.array(tuple(values))
```

Out[94]:

Was correct	comedy	thriller
False	6	7
True	14	29

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.

A pattern that we notice is that our classifer is more accurate for thriller movies than comedy because out of 20 comedy movies we got 14 correct and 6 wrongly classified which means we had accuracy of 70% while our classifer for thriller was that out of 36 movies we got 29 correctly classfied and 7 wrong which is 80% correctly classified. In conculsion our classifier is some what accurate for thriller movies but not for comedy.

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 [ ]:

```
# 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",
                          "humor", "joke", "thrill", 'kill',
                          'walter', 'dead', 'murder', 'oh', 'well',
                          'love','great', 'yeah', 'found', 'nice',
                          'where', 'your', 'their', 'and',
                          'gun', 'mean', 'ship', 'shock')
```

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

def another_classifier(row):
    return classify (row,train_new,train_movies.column("Genre"),13)
guess_genre= test_new.apply(another_classifier)
compare_guess= np.count_nonzero(test_movies.column("Genre")==guess_genre)
proportion_guess= compare_guess/ test_movies.num_rows
proportion_guess
```

In [96]:
```
table_correct=Table().with_columns("Title",test_movies.column("Title"),
                    "Genre",test_movies.column("Genre"),
                    "Was correct",guess_genre==test_movies.column("Genre"))
table_correct
```

Out[96]:

Title	Genre	Was correct
new nightmare	thriller	True
the grifters	thriller	False
the body snatcher	thriller	True
smoke	comedy	False
godzilla	thriller	True
mystery of the wax museum	thriller	False
rear window	thriller	True
u turn	thriller	True
jason goes to hell: the final friday	thriller	True
the crow: salvation	thriller	True

... (46 rows omitted)

In [97]:
```
table_correct.pivot("Genre","Was correct")
```

Out[97]:

Was correct	comedy	thriller
False	4	5
True	16	31

In [ ]:
```

```

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.

A pattern we notice is that our classifier is more accurate for thriller movies than comedy movies because out 20 comedy movies we got 16 correct and 4 wrongly classified which accounts for a 86% accuracy, whereas our classifier out of 36 thriller movies we got 31 correctly classified and 5 wrong which accounts for about 80% accuracy. In conclusion, our classifier is more accurate for predicting thriller movies compared to comedy movies, but both have a high prediction accuracy of above 80%.

Question 4.3

Briefly describe what you tried to improve your classifier.

I improved my classifer by choosing words that are not only outliers but also commonly used to describe comedy and thriller movies. The line of best best demonstrates words that are both found in comedy and thriller movies, therefore choosing outliers have a preference toward one genre. Outliers that are below the line of best fit have a preference toward comedy movies whereas words that are above the line of best fit have a preference in thriller movies. I chose words that describe comedy movies such as humor and joke as well as words that describe thriller movies such as thrill and shock. I also increased the number of features to classify the movie genre. By increasing the features, we are able to provide greater distinction between comedy and thriller.

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

In [ ]:
```
_ = 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
- Train/test split and cross-validation
- More information about k-nearest neighbors
- Overfitting

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

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

Modules to think about using:

- Scikit-learn tutorial
- TensorFlow information

...and many more!

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

In [ ]:

In [ ]:
```
# 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') and len(q) <= 10
print("Finished running all tests.")
```