Yelp Dataset Analysis

This project delves into exploratory analysis and building predictive models using the Yelp academic dataset. We explore machine learning tasks in the context of a real-world data set using big data analysis tools.

We have chosen a subset of the Yelp academic dataset for you to work with. This subsampled data is loaded into RDDs in part (0). The complete dataset is available from Yelp's website here.

Part 0: Load the datasets required for the project

We will load four datasets for this project. In addition to the four datasets, we will also load two lists which contain names by gender. These lists are helpful in assigning a gender to a Yelp user by their name, since gender is not available in the Yelp dataset.

In [ ]:
```
import sys
sys.path.append("/opt/packages/spark/latest/python/lib/py4j-0.10.9-src.zip")
sys.path.append("/opt/packages/spark/latest/python/")
sys.path.append("/opt/packages/spark/latest/python/pyspark")
from pyspark.sql import SparkSession
from pyspark import SparkConf, SparkContext
spark = SparkSession.builder \
    .master('local[*]') \
    .config("spark.driver.memory", "40g") \
    .getOrCreate()
sc = spark.sparkContext
sc
```

Out[ ]:

SparkContext

Spark UI

Version	v3.0.1
Master	local[*]
AppName	pyspark-shell

In [ ]:
```
import json
import os
import sys
import os.path
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline

# helper function to load a JSON dataset from a publicly accessible url
```

```
def get_rdd_from_path(path):
    file_reader = open(path, 'r')
    str_contents = file_reader.readlines()
    json_contents = [json.loads(x.strip()) for x in str_contents]
    rdd = sc.parallelize(json_contents,  numSlices=1000)
    return rdd
```

The first dataset we are going to load is information about Yelp businesses. The information of each business will be stored as a Python dictionary within an RDD. The dictionary consists of the following fields:

- "business_id":"encrypted business id"
- "name":"business name"
- "neighborhood":"hood name"
- "address":"full address"
- "city":"city"
- "state":"state -- if applicable --"
- "postal code":"postal code"
- "latitude":latitude
- "longitude":longitude
- "stars":star rating, rounded to half-stars
- "review_count":number of reviews
- "is_open":0/1 (closed/open)
- "attributes":["an array of strings: each array element is an attribute"]
- "categories":["an array of strings of business categories"]
- "hours":["an array of strings of business hours"]
- "type": "business"

In [ ]:

```
# Load the data about Yelp businesses in an RDD
# each RDD element is a Python dictionary parsed from JSON using json.loads()
businesses_rdd = get_rdd_from_path('/ocean/projects/cie170025p/shared/data/yelp_academi
print (businesses_rdd.count())
print (businesses_rdd.take(2))
```

```
61184
[{'business_id': 'vcNAWiLM4dR7D2nwwJ7nCA', 'full_address': '4840 E Indian School Rd\nSte
101\nPhoenix, AZ 85018', 'hours': {'Tuesday': {'close': '17:00', 'open': '08:00'}, 'Frid
ay': {'close': '17:00', 'open': '08:00'}, 'Monday': {'close': '17:00', 'open': '08:00'},
'Wednesday': {'close': '17:00', 'open': '08:00'}, 'Thursday': {'close': '17:00', 'open':
'08:00'}}, 'open': True, 'categories': ['Doctors', 'Health & Medical'], 'city': 'Phoeni
x', 'review_count': 9, 'name': 'Eric Goldberg, MD', 'neighborhoods': [], 'longitude': -1
11.983758, 'state': 'AZ', 'stars': 3.5, 'latitude': 33.499313, 'attributes': {'By Appoin
tment Only': True}, 'type': 'business'}, {'business_id': 'UsFtqoBl7naz8AVUBZMjQQ', 'full
_address': '202 McClure St\nDravosburg, PA 15034', 'hours': {}, 'open': True, 'categorie
s': ['Nightlife'], 'city': 'Dravosburg', 'review_count': 4, 'name': "Clancy's Pub", 'nei
ghborhoods': [], 'longitude': -79.88693, 'state': 'PA', 'stars': 3.5, 'latitude': 40.350
519, 'attributes': {'Happy Hour': True, 'Accepts Credit Cards': True, 'Good For Groups':
True, 'Outdoor Seating': False, 'Price Range': 1}, 'type': 'business'}]
```

The second dataset we are going to load is information about Yelp users. Each user's information will be stored as a Python dictionary within an RDD. The dictionary consists of the following fields:

- "user_id":"encrypted user id"

- "name":"first name"
- "review_count":number of reviews
- "yelping_since": date formatted like "2009-12-19"
- "friends":["an array of encrypted ids of friends"]
- "useful":"number of useful votes sent by the user"
- "funny":"number of funny votes sent by the user"
- "cool":"number of cool votes sent by the user"
- "fans":"number of fans the user has"
- "elite":["an array of years the user was elite"]
- "average_stars":floating point average like 4.31
- "compliment_hot":number of hot compliments received by the user
- "compliment_more":number of more compliments received by the user
- "compliment_profile": number of profile compliments received by the user
- "compliment_cute": number of cute compliments received by the user
- "compliment_list": number of list compliments received by the user
- "compliment_note": number of note compliments received by the user
- "compliment_plain": number of plain compliments received by the user
- "compliment_cool": number of cool compliments received by the user
- "compliment_funny": number of funny compliments received by the user
- "compliment_writer": number of writer compliments received by the user
- "compliment_photos": number of photo compliments received by the user
- "type":"user"

In [ ]:
```
# load the data about Yelp users in an RDD
# each RDD element is a Python dictionary parsed from JSON using json.loads()
users_rdd = get_rdd_from_path('/ocean/projects/cie170025p/shared/data/yelp_academic_dat
print (users_rdd.count())
print (users_rdd.take(2))
```

```
366715
[{'yelping_since': '2004-10', 'votes': {'funny': 166, 'useful': 278, 'cool': 245}, 'revi
ew_count': 108, 'name': 'Russel', 'user_id': '18kPq7GPye-YQ3LyKyAZPw', 'friends': ['rpOy
qD_893cqmDAtJLbdog', '4U9kSBLuBDU391x6bxU-YA', 'fHtTaujcyKvXglE33Z5yIw', '8J4IIYcqBlFch8
T90N923A', 'wy6l_zUo7SN0qrvNRWgySw', 'HDQixQ-WZEV0LVPJlIGQeQ', 'T4kuUr_iJiywOPdyM7gTHQ',
'z_5D4XEIlGAPjG3Os9ix5A', 'i63u3SdbrLsP4FxiSKP0Zw', 'pnrGw4ciBXJ6U5QB2m0F5g', 'ytjCBxosV
SqCOQ62c4KAxg', 'r5uiIxwJ-I-oHBkNY2Ha3Q', 'niWoSKswEbooJC_M7HMbGw', 'kwoxiKMyoYjB1wTCYAj
YRg', '9A8OuP6XwLwnNb9ov3_Ncw', '27MmRg8LfbZXNEHkEnKSdA', 'Bn4sJUTtKFZQt0FKHF2Adw', 'ugu
XfIEpI65jSCH5MgUDgA', '6VZNGc2h2Bn-uyuEXgOt5g', 'AZ8CTtwr-4sGM2kZqF6qig', 'S742m-AuQicMS
LDdErrLZQ', 'uGmQ6ab4iVpWn5m61VFhkQ', 'GJYJX4SujVj3BR8v2F9PDQ', '3shjifK-vZkIHciyy_KbY
A', '4lc_H2Cf7CO0tCgyA3aSVQ', 'Tunkp_F1R_uFBJQTsDxD4g', 'B9pKfr27czBbCoAIircZdQ', 'pePGM
O6EbDpbaZ7D2m6HIg', 'XRM8W6HUoXbrYKR3BCj9Rg', '8DqIWXsKXOipfduYEfFpNw', 'dvRVX54Z9f7Om51
NsTRX1w', 'CM0saLQmk4oAB17UmQTV-g', 'HANb8-8InWnju-XzBQQSBw', 'JuJeZeQJgv7bUreY7a1SlQ',
'2NVVEEYhhoVELdaPILFrDQ', 'e4M9_S-ASmRys3DvKQfotw', 'XA109o963exKgoVGcg9z7w', 'tVAKdax3J
bf24R7OJB99JQ', 'OMWT-Z6OnJLcg44lCuDuhw', 'd82F_FFtowYtjGtxRySehA', 'pW91HUnVz6ssLZ4dY-z
tyQ', 'ojQYtstkGXtcryk5I9GTYA', '4hxVql33ldY_gkf3rG8_3w', '0arQ82n4mvrc42U8KuoE6g', 'w6V
v-kldGpmvSGqXvTbAdQ', 'NZeCINmoOJ8vsQvYkZAp9w', 'n9ltC5DxMjefffeMfBgcXQ', 'QBj2AL66bEAmK
3ULkrn1Gw', 'ayZlWyV1M2k_UWx1dreIDg', 'AG1KRDkIa8QCCPsMnINEOg', 'P-lfV8cdAgEOZJkw2dJps
w', '7Pef9EA21szXXOPJo6Gb5w', 'YvtRBqCqhSIhGGREqcqnQw', '8M7I2-s5nQ8LSkuCvN0Xqg', '-ETxy
7f37BBQXhw5zJfkrQ', 'TjPt2hD56jfpmtoa6kVVRg', 'uZgFEPEUIWJMaaqvMlxYoA', 'YKppbE7ogNmbtVr
luxEF4A', 'D7uXj1_0pE_Fa2YeLkOkyw', 'eVV99D3EZvkuswIvIySKiw', 'PhwgVPqpJlxu40AxAOUBbg',
'gWcz4QFTW76q8VtpIlZEHQ', 'wpcoiQNbO5x0xkZwKFVUFw', 'Ot-QjYpIdVsluuxcqidjkA', 'E3C_gEhkt
ux9Ca-Z2fmNTw', 'ClaGLOr72DPJlQhRc9HHyQ', 'sc6wS6YOFIgtN2XuJI8_WQ', 'dD5mA6kwVlBHvrNtX9E
yZQ', 'qwvgycPC5u4JZv6DfawPqw', 'As_oVE27fiE-0CbDJwUDiA', 'SuzSrMWoycf4DXL_DMMdNg', 'y0S
```

RKlzVuvj-mbQtbsWx-A', 'qbfQRHLvZk5WSkKY0l_lMw', 'D6A0rn5MIg7AVRrVeouABg', 'KUqpJ9eKREuvl
nlitJJRYg', 'gOvKWYXobX_Pgzsz-K8auA', 'xzdZTsR2KKcQfjF4OjJeOw', 'DyfA7aPzRQ81kJnbkuIiJ
Q', 'tA2suyZG_qWIuNB_HR-YRA', 'R5QmtpSsrQ9mqIxjZIv3uw', '5T8SdBwiH0jdwatUJ7KFUg', '1Gfn4
nlqPEIvRX4cnctJlg', 'A4_fxqSj4tgYet8OE3jTyA', 'oPD_t6BVpfhvu1-TS4FNzA', 'gn3Y0mWduWrvT1
hvu7fv4A', 'NUeLSLukiPPCSK6EVdUo_Q', 'rtrh46NJfksxR4TgljYg9w', '5zpVMWD-q6nEYapiWrSyX
w', 'ooZ7TV6IuxuqbFgnqaoSsQ', '8phGuOZ0NIfeJiTX-ZDZLA', '8L1p_zlmAB_uWyaW0QTaCg', 's7ZS
ag-bHo0C4fSBDJeBDw', '5ANDwUyQqsRhEl7QBKRFiw', 'KMGF6KAnPRoOTFVXBy9jtQ', 'yk-lz4JUKREUV
w9hxk-VZw', '6uM5lx0DCX4ctQlebtxraw', 'Hc6agWZ9jGCxsPAqQwlf2w', 'z39ulrgtvEMB2oeokx0-m
w', 'wLt28qh-46LFhKOeByWZ0w', 'R9FFFqYOUmvWjaHqODfzyw', 'p4Jg_rO_AsCWZGMd_w8xjw', 'pJpp
mQyW74ejj8xCa2SHuA', 'hzaqnSxfRYv_Wbp6CMKtQw', '0QwV8PJk9PJDFaMNw1Fcsw', '6pC5j-hfP4xLZ
B2W0Wi1vg', '8nAj60xsBjcHXLLSs7cPqw', 'NhbV8AbaQiKNzEU11cbxcQ', '5Szwd1CSepRXwAFH12nPz
Q', 'ifD7o8Mi67yNYwgWL5CbKg', 'AaZdXn0I6F5bdIVwGpxdDA', 'YwcacxA7WCB9fryDHDxF8w', 'HBFt
zjRarXxDH8xZRSNsPA', 'zi0kxtr5gq0Flg7jYdIlMA', 'oWEdmTYl13VHh1A_Mkhh2g', 'n1GSQ7VT9sTL5
MlCWolbNw', 'PO8XNxGUizzzKGHmn24fQQ', 'JTRxzA1AZ8Zfwj7FpH2CnA', 'AixLPxra_4HirfchjPkiW
A', 'xOK3i0wgmkpGu54BuIcwQA', 'F9T6m1YdRFreyKDufcyoOQ', '-cscgA-lVPipPwEX4RmcgQ', '1UJb
Ol-jKNIUjroeC4xWxg', 'ACLuEsNHV7lfq1O7CHuFaA', 'UyOtYjnavsr-Dtjo6Xy91g', 'J11gYVC1u9S5j
qEg3oh-gQ', '9LZP7Ll7N9zkuWfA5DUAYw', 'SbM1m5nejHDD1pJePsar9g', 'wF7_-zf7R8Rri7Kq-a0tl
w', 'B5o7JC4iRJ7bKPRnoYEIHg', 'SAPz30NHFJ95SBlvz5cTqw', 'fzkO1lGkx8MHhT8S4Rbg3w', 'tlSS
QwfHYJany7wPoTH46A', '2VfhMrTJ9bZ0O4PZmRmxKA', 'odeoZZHRQp6TkeoEQk3eeA', '2XCsVpAMsGUyk
rM-BBFEXA', 'qVePixO2kuK0Dr03upCWFg', 'D9CIW04wrcQzZNdCEoASAg', 'G-iuXcbJnX3sA3lhB6nFh
Q', '3utEqyNzFcxH8E8pm7ByPA', 'T5TZfn9_9X-LTbRtaMAgUw', 'fczQCSmaWF78toLEmb0Zsw', 'Hyhm
pb3UnbjtSEXvgviCow', 'FI-Fjxr3fUBx2q7icnqcfA', 'S8ZY1HEDoECqKrIUY34JOg', '3JtdBvnY6Wzp2
2gt5zFwDg', '470dZswdTHz5aaGPvIYNpg', 'JT5kM6wRwh7cNt8IAuX_sw', 'r7V9Nhes-Eg-lH0cqfpFF
w', '2m3lE8p8T2nFRDxIUaRXew', 'bZFRqP7s0Vszxeu8_IwYow', 'snd45oedwXxF9tsHUX_6dg', '8jEK
8gxHYg9w6sINujO0MQ', 'LTWY2Ee-SYu8ItVOebD3CQ', '7zpDhrRZRTGCkAh3SHbEww', 'NSg7k2UFt_HqM
2mvsw-n0w', 'egdj4-wXAzYzT5bmTAHYqQ', 'wxhf-RSExCGECZuz_iVV7w', 'C6IOtaaYdLIT5fWd7ZYIu
A', 'jntSRFnN_O7QfFzZo24rvA', 'yfQwWLYtqwHKiky1hiTBew', '6SxnRYFR3GUzGREKSttxJg', 'WZgo
M8UuzxjZf1AaWx3qZQ', 'cNj8rheuldPDX9zF6q2yaA', 'OqOCMRpVNvvdPDMhoG_DCA', '9ZonzxW0DAqIU
O_5tDqZig', 'rLQNlHnIKdtoqoU77CbEUQ', 'LbgQK5B_5IkN77FgRJHhrg', 'it23EnutJ6fOjeYQBDKJu
g', '1XWzXgFvcu5pU2wj1ONQ_Q', '92j9xiUMOsM3HviIGoR8Yg', 'AP6udBIStvWLY0T5rS_24A', 'G_PC
zMHcOLoMyyEsxW1DFQ', 'ZuQ87LQYWRw3AHj-jEiQ1Q', 'zdutsCEMdvc0B8s3_XN3nQ', 'fRbbPPUdMCmL9
o39yshi2g', 'pjgE1d3csesAkgn6rQI_RQ', 'C7nUczj6n0ouxWqp1I7cNQ', 'D8EE4gbAiXS3DF_HNRJfA
w', 'Milv8k4zZsoCaNFZugKtJw', 'GALRNo-AbcAHxEdmqKl--w', 'XOe-I25-W_FnbYfQyafylw', '-weT
B55ulUzv0E_4laXsqQ', 'kuP0AMnSGgkLfNSt0wOAgA', 'EwMgdjuPXLAuxxu3IOUyKA', 'Mx-vxv_V-SQCe
76w4RmUfA', '6xxVdJBdMREj1fCqjm6UtA', 'm61SjiXAIlOY1qAXOHEk4Q', 'R4sYowLqQQEpxWjQbqaFv
g', 'OCkge7lzOw33bqMs1244Yw', 'rjJJ8QQg-eqb7mgMxArgBA', 'T_wjLgPOPXry7Bea4MzoVQ', 'hBwb
BaBqofCgvkJ3PatuVw', 'epTMlecM8SZNjDhBsd64Jg', 'PUMROYfbVgKypcfJNxs1mw', '82ZqgHOfUMV1x
tpG2KfisA', 'IN2HcBun5PpyfQK8OC68iw', 'yQZL8MTulNh3d3de-N0oug', 'nZFcWvgPU8T4JCjOVvJU8
Q', 'FDPjfyyHNfy9tqyxo27QeQ', 'jIuc3nM9R0KWX3v-8OyLsQ', 'F1oGgK3NtTYnyRMxupiiuA', 'nGYa
T5sAuqKkd48dqX_2Kw', 'NzP9OoNP8vBMRUSs0UXBNw', 'ZTs-go-xYRQOp3PSw_XGDg', 'gc4rNAgbGydNM
APJ85FZag', 'KYuljD4Tovti3J1pdBnT4g'], 'fans': 69, 'average_stars': 4.14, 'type': 'use
r', 'compliments': {'profile': 8, 'cute': 15, 'funny': 11, 'plain': 25, 'writer': 9, 'n
ote': 20, 'photos': 15, 'hot': 48, 'cool': 78, 'more': 3}, 'elite': [2005, 2006]}, {'ye
lping_since': '2004-10', 'votes': {'funny': 6849, 'useful': 12642, 'cool': 9837}, 'revi
ew_count': 1233, 'name': 'Jeremy', 'user_id': 'rpOyqD_893cqmDAtJLbdog', 'friends': ['18
kPq7GPye-YQ3LyKyAZPw', '4U9kSBLuBDU391x6bxU-YA', 'fHtTaujcyKvXglE33Z5yIw', 'SIBCL7HBkrP
4llolm4SC2A', '8J4IIYcqBlFch8T90N923A', 'ysYmC-ufbdmVEX9yAv-VEQ', 'UTS9XcT14H2ZscRIf0MY
HQ', '1blidZhgxDVSBuJ_dbTU6g', 'HDQixQ-WZEV0LVPJlIGQeQ', '91ovYNfGfPiqe6GbTcPwTg', '8VJ
UJbrfPYJC0vMAtXGR_g', 'T4kuUr_iJiywOPdyM7gTHQ', 'anTtulQ9G2b3SipCEwGVHg', 'Re447krbp0VQ
VX3Dk4Tsog', 'XejvgD4CrkR7FeIkjwAjtQ', 'Ym2-9Rdeb_PPher2F6qksQ', 'i63u3SdbrLsP4FxiSKP0Z
w', 'T7t5UdadpHua-YnmiFpcZw', 'pnrGw4ciBXJ6U5QB2m0F5g', 'ytjCBxosVSqCOQ62c4KAxg', 'K4FA
ia2Iy5MVnmBLfS-mCg', 'o-6ZHG2iUO3oooMXQawd9g', 'r5uiIxwJ-I-oHBkNY2Ha3Q', 'niWoSKswEbooJ
C_M7HMbGw', 'kwoxiKMyoYjB1wTCYAjYRg', 'F4sfffJ8-VCNFLUT0q6Gtg', 'TNJRTBrl0yjtpAACr1Bth
g', 'W0M8csYKD2ryUsr6OsACyA', 'POQZrdj2otEnb0Le1oLWjg', 'HKvXA96cbqAawKCU7xyGSw', 'EoTo
oNBcxu0Hy0-XeZI8aw', '9A8OuP6XwLwnNb9ov3_Ncw', '27MmRg8LfbZXNEHkEnKSdA', 'WsPgO3ycZjRp7
wFh18JdMQ', 'Bn4sJUTtKFZQt0FKHF2Adw', 'AC84oF7Ciuw7M52RszgWUA', 'uguXfIEpI65jSCH5MgUDg
A', '6VZNGc2h2Bn-uyuEXgOt5g', 'AZl5xknQ2QzJGUfd0V_mjQ', 'pz97SxRe1Vk-5_K6EB9OSA', 'AZ8C
Ttwr-4sGM2kZqF6qig', 'xaPwzbtkselbxI472WIjsA', '4ohJEFpIk99NGGNdSa2Uig', 'q1dyK6mo5QkHI
0ZBPXeCQQ', 'xRpSYEOS4MN_fnfjBPy5dw', 'Gcb3QKjGjvH_GdzLpM-fAg', 'uGmQ6ab4iVpWn5m61VFhk
Q', 'GJYJX4SujVj3BR8v2F9PDQ', 'Eh9SSFnh7xOF8N3X6ESPOA', 'DglSIRnl69GUqHvVnG3QaA', '3shj
ifK-vZkIHciyy_KbYA', 'xafscyoSZPnchPkzGyDAWw', '4lc_H2Cf7CO0tCgyA3aSVQ', 'Tunkp_F1R_uFB
JQTsDxD4g', 'yAIA9J3FQmg6XwAfXMLh-g', 'IsVIVpgV7VralvYD4DEQYg', '-6k_vUBna1kzXBjyBOCeq
g', 'B9pKfr27czBbCoAIircZdQ', '7WGcHwBkXKRCTL2AG--3CA', 'ORXFUXpvHloPYVyQ4C5oqA', 'Rir-
YRPPClKXDFQbc3BsVw', 'wj-UBmM6p-TkfD4Xg_zY7A', 'G4LfUOgAglHfZLILESSb0w', '-1Q1s_NMGjBLB
ULA8z_npg', 'UDIR9Aojno4FzqYfOuwunQ', 'nw-QVHwrhhwZpVuR5DWdpg', 'qaeKxVt-KnFKKzKxKVrKD

A', 'uyf9AeK95dh4oxs36bMQkw', 'lxZSVeJz6KEBW1nlA3JKJg', 'zTWH9b_ItSdLOK9ypeFOIw', 'pePG
MO6EbDpbaZ7D2m6HIg', 'MKkaAk3EZumQVvnMjd_avQ', 'XRM8W6HUoXbrYKR3BCj9Rg', '7FuLnS_-b79GG
-33mwLaMg', 'o5KEicAZaO72mHgSYVZ8bw', '8DqIWXsKXOipfduYEfFpNw', 'GZ9kx3IQe15JWrVGrAQZ8
A', '7uxXuCcpw9-mUS3OJVw8aQ', 'p64irD26M0-tgKj0z1jgKw', 'dvRVX54Z9f7Om51NsTRX1w', 'K2PD
I4_GJLUTwjq0jA2viQ', 'WF4Bz2Z37AGdFZSVqOObOg', 'CM0saLQmk4oAB17UmQTV-g', 'jIJXDhJ08iCh5
KxZjIuRPQ', 'HANb8-8InWnju-XzBQQSBw', 'CvQhlr_VtwGqQBbiFOjOaw', 'frz1uzgKdxjOt8bWZCMmV
Q', '-jXrmFgORGmiTEfxaKlA6w', 'S6sAyi61H7ecAj6qfNMKNQ', 't51FxYOLqwbTSBeyPbHHVQ', 'e4M9
_S-ASmRys3DvKQfotw', 'weLT87ID6D0yPWg5d3atcQ', 'OMWT-Z6OnJLcg44lCuDuhw', 'd82F_FFtowYtj
GtxRySehA', 'pW91HUnVz6ssLZ4dY-ztyQ', 'ojQYtstkGXtcryk5I9GTYA', 'n0vqc-DbRK84XOLKHp5wj
w', '5StJwPvmlJj0yMHBj1PATw', '8w23JX96mElm3ViwtS_WKQ', '4hxVql33ldY_gkf3rG8_3w', '9V14
_mnEi28kNnN_aVhuSA', 'kxkSjuIHutMTS88y1-b_Ew', '0arQ82n4mvrc42U8KuoE6g', 'CvWhDNP2pLVYd
lHg_e4DXQ', 'WGf64PeXIZtdoqkBj0-_zA', 'YA6LcxRxXEyMN-jpFSBarA', 'ZZ5hV2jcxchiZo6C9AXq_
Q', 'lDnXkR-Hc5FyPZ7T5oPyRQ', '3_VAhzONbrThOlZr_6crpQ', 'NZeCINmoOJ8vsQvYkZAp9w', 'hfdh
6M4M8cOqG4McE1aoYA', 'LroJKi7_BWqDT7lbHvt6kg', '0FqVxW9PVqh66vMbNJ4Y1A', 'eB_WpbBxBTABk
zes-ZQTRw', 'QBj2AL66bEAmK3ULkrn1Gw', 'ayZlWyV1M2k_UWx1dreIDg', '1--cOFkdo345uvepQwfGE
A', 'hGyKkRaJkOpl6jagEZAWcg', 'JguVDOCjcNicW5zCwuqehg', 'ms9P6qj-mKUGkBJRY0yAHA', 'I38k
TTMpIQfm6CZQIeZKgg', 'mj2X5F2xh2xxPTaw2HsnGg', 'qq57IKwroZux85aC6hM38w', 'KUtzHIs-Y2Va3
STDlWa0iA', 'Gi62agSW34oYrCcEdB9UGg', 'ZQf-UQL8ChAmTfN9SrUBfA', 'P-lfV8cdAgEOZJkw2dJps
w', '7Pef9EA21szXXOPJo6Gb5w', 'Qcn-UlswGM8JmrYbf4WMLA', 'Q4rDqPKe1Iko7Kldj9ADBw', 'MfSP
SNoF2iWWt7ZdCA8T8Q', '4RK2egxIKfC55TGg2ToulA', 'HHCn5JGkJc5Lp3EaGw6baA', 'hlyE3S6w-949M
A-u8TXsjA', 'hDuhc9RQOf3ydg1PxLkRxQ', 'ZQduSjrBcIYjO1d8IaXxVw', 'PuuRW29sMLIN0wuoZAllF
A', 'KLIgcDSCque_hMrtjnUcow', '8dwO6Y8Vsss1hGEeXm1acw', 'Q1lOqbip5AspTHHBJ4VDlA', 'ARe8
Nr_YehB2ubsGJhZ-hg', 'oDYf8_HV9WpSxrLmJ1j5JA', '_YI2xCPY90YqTazh-xL_Pg', '-ETxy7f37BBQX
hw5zJfkrQ', 'oUa5MYdA6FdEJO-HBB0DaA', 'jDCblyKGE1UrulT7Dyzb0Q', '8wQ8fugCcUEMO7lXg8yDG
Q', 'BDx8yOOvNE5UWs_xsIOXwA', 'Wwl_HlxMyfnPhH4iKbzdOA', '49pqdhT4su49QnwmXNdyIA', 'uZgF
EPEUIWJMaaqvMlxYoA', 'gvqh39oy-m10H7aNyVBL8g', 'YKppbE7ogNmbtVrluxEF4A', 'qB_WxOFVPKlK8
DmbhPXGhw', 'Gwj-__sLYJUXO3S2ajHUDw', 'MofhftznXiuBxID1GNOSYA', '9dmIMMQT6XJ19RSsWioU5
Q', 'D7uXj1_0pE_Fa2YeLkOkyw', 'zzWUzYAj8SXwohidVLax9Q', 'eVV99D3EZvkuswIvIySKiw', 'Phwg
VPqpJlxu40AxAOUBbg', 'LiAq4ejWXt9ooo9X5wHQRg', 'ajCb2CmRkd5V0OvzsjYC5w', 'CkY48UaWOmLhs
dc_NTfFBA', 'Z78lp4o03tZbGRMwLZw0nA', 'J4xFxIXDH6nXdF5rTYiBCw', 'pbUs7oReAS4lhMvEzwNVh
A', 'AtFGH7Spf7QMNm-BFcshWQ', '_cDnFQHMrwo5Xg_1Z8QQNA', '-e0zsU6ootfv9ImTZ7wb2A', 'wpco
iQNbO5x0xkZwKFVUFw', 'eSAmN-5RAu8y1igSvz_yJw', 'Upt8QGZF8LFWRSw3LUopIg', 'wOSwzy3p0cuwb
v5xjZW1bA', 'Ot-QjYpIdVsluuxcqidjkA', 'E3C_gEhktux9Ca-Z2fmNTw', 'HpuonncxvaKSixI3bM5Of
w', 'mGSyq3buf9AggxuXuZRkyQ', 'ClaGLOr72DPJlQhRc9HHyQ', 'YX-O9N2rg1DNH-IbyD1GLw', 'wHr9
izcKk8uLc6tucnXskQ', 'sc6wS6YOFIgtN2XuJI8_WQ', 'PMgE5Yqv7QL_cTjjInyYIg', 'zdhIXcofp-N9W
fC0tiQ5jg', '6HJzAxgP9XB-JhyjeOEFaA', 'tn5AIW1NVjTI4oNdrpqwiA', 'dThnQfCZCpbLXwqRqnQhF
g', 'bUjVCdMEdlHUZTj-uiBLUQ', 'yWeouzBJp_kwplPtwcl_ig', 'hO4UkN8RniS204O1PdxoBg', 'X1YV
sT7fadktjP0WvraqtA', 'zlta0Zhv3qrlPT4wgmx1zQ', '0BxWPocgVIAzZUBQ_Nrimw', 'Cz6cUmulaEii_
_wUcQ6QVw', 't1h9wl-Q4zGnt3JX860Ztw', '4YhwHzvNj-Rg8IhLntmhCw', 'OwcvQW8hbLjcU8eW-1xCI
w', '-QjeoLnIeKStNfGIq5KGDw', 'iR1pShxMP5Aiwo1DZzGJlg', 'Mg0G8_Wa_FoNQO3Kv7RDNw', 'G5Ps
RwFSjF4Sql7ME0ENsw', 'SuzSrMWoycf4DXL_DMMdNg', 'niQLnZVEz-E9KtbqFRKWcA', 'dNBfuG6I4nadi
bbjFX7SiA', 'y0SRKlzVuvj-mbQtbsWx-A', '2hBQmM99bJWBc4TLXyFCxg', '-E6yYsngVZqIUHlX856sb
w', 'hpAFdPtU6Am_agXJT856gw', 'n7WUEZeNKCYgC950r3rbvA', 'BuaNITlWZwJskEhBbz-H6w', 'qbfQ
RHLvZk5WSkKY0l_lMw', '6kJ9OXgotJzYBdWWTOjpCg', '10O3dYXmF3R_OMcXBnTkWg', 'D6A0rn5MIg7AV
RrVeouABg', 'KUqpJ9eKREuvlnlitJJRYg', 'a0pG_Dl4sKb3yNps7I_hMQ', 'x7_Wyt9kjqn3itW9y0DHb
A', 'tAAi5iUrQY1OKYnuChDJGQ', '5axKCCwwpcfPTzv7fS819A', 'BgqGtOskxyKzP2oGR8uQ8g', 'dTye
HgCDwu9binCwjw7d5g', 'MMCiQSRLdhyUzkHyksST_w', 'YqXkPv4dittwSxctmZbqkw', 'gOvKWYXobX_Pg
zsz-K8auA', '8qoEnuFvMAQWv7qmhKyulw', '6JXh-O4gFMm-ugGad6PvQQ', 'brRr5SRQyzYa6S623Kvd5
Q', 'b0bhYMBU9vk_smKjby-u_A', 'muSuq2Ep_PXjvLYXHNStig', '77s2qg6JOqXy69stUku6ew', 'y4F3
0s5qp4VBInqmt4Upyg', 'WTvD6w7arH4S74rAlefC9A', '2176sNPv5tchjgJcV-2yiA', 'DyfA7aPzRQ81k
JnbkuIiJQ', 'NhP0IQNpZkw6tQ0drfd-qQ', 'fA1-M_FoudCU9xxpauJBaw', 'tA2suyZG_qWIuNB_HR-YR
A', 'C3edE64xBOfWC6HJbxRJxQ', 'PzYWzSKJ_HcwNKPD_f1cXw', 'VPon9k-yAGmeSw3sUfMyvw', 'VJjl
Ru2Hg2VgaN1P2GVdbA', 'SCw9iirijegjhSfK26ZheQ', 'eeRhnURn8ebE84S1xymc7A', '9VedeYzgqzq7M
XXOWhbQcg', '5T8SdBwiH0jdwatUJ7KFUg', 'OMB8Y_DOaB21vCXa-cruwg', 'r6usm4GcaZbuB1iIbkJzS
g', 'inEHrQz45dQ_bRCW7HxgQw', 'faoGu7ngJ2t4I9DO5cxG_w', 'eTeQp28_mMoImh9c4l2xxA', '1Gfn
4nlqPEIvRX4cnctJlg', 'PlQSyQNujQkd-ewRLZye_Q', 'aERWRgfZJGSuPoCdhCRUoQ', 'Q2e6q3_5DbF10
ZKJCPEuOg', 'tgxgeEvYUA_Gq8Ekc5X1FA', 'ZbWP0L2Ni_icKOJboC4SbA', 'j22WZq4GjIhdp1cyGNhJr
w', 'kygBLEEZ7lqLi6CnM7gLyw', 'E1zOJdKblkxFEfBGtX-hUw', 'Z7nr-LLKaPG7bwKuk1PWVQ', 'uePT
Lkka-CXvuoIk1d1Ucw', 'oPD_t6BVpfhvu1-TS4FNzA', 'BbWSGazo6_fUCKxNLJ4Z2Q', 'FHaG51Fyfj8xq
4CNsixWwQ', 'gn3Y0mWduWrvT1hvu7fv4A', 'pzHB2lJov4XgmfGgkh6N3w', 'cAVLTaT4C5Zdz9lzTVYUL
Q', 'Bq-fS1NXfBJiaBfLB2vwEA', 'aPQuGzqs4D8my3K0vXjb8Q', 'AGHesuIAE2hJ4yPGZTzO0g', 'q6hi
ZCOlvZ-JWY5alt4aiw', 'NUeLSLukiPPCSK6EVdUo_Q', 'rtrh46NJfksxR4TgljYg9w', '52hYzmoL4yUFy
c9Ct1Zukg', '94N07GeXPUyC8baIz0l3UA', '2Qo8EXnxxfTtUWIqPPih6g', 'DrKQzBFAvxhyjLgbPSW2Q
w', 'v6giEKBrh7uLR3npuqda9Q', 'DH3X4Qj14nIsdeHMFf4xSw', 'wZtfgvBJ7VihuRyq9Nqepw', 'Swqh
vXTAVUrGMxQBlyn8vQ', '8phGuOZ0NIfeJiTX-ZDZLA', 'dAcYRfvKOVw94_U4gyz8cw', 'TiftBBMFaQIMK

-vJaRbZgw', 'XBHJCDzNh--mBhJkzyM_iA', '_YNKf7vrI08r2eI0tM6zqw', 'xUhW65cmXuAjIaVVpZHpT
w', 'HHij7LRNfzWru9TKEIsWgw', '3Fvga8TuSEOdsN_xYQZR3g', '8L1p_zlmAB_uWyaW0QTaCg', 'g0LF
QoGn9TNDpKlFjmFWwQ', 'u3xXsUp-01Tw6IF5eqHbbA', 'TFD4dqa9BXvDbtbDx8JwuQ', 'E1FiD1I3K7YEV
Rd00EkPsw', 'BSI42r_Q2NmDeYU-rf12nA', 'DzTMG-LBSfMeFsVdkzoKVg', 'KMGF6KAnPRoOTFVXBy9jt
Q', 'PFtbOIt89VdJB6OLLNaEGw', 'UbLmrkrIga-o2v1IS8mLHA', 'M35ddM0p0VtAb-n3DZtoQQ', 'm6Pd
Ls4FnQn4MHFd7i_wsg', 'rp82QG3ewXv0XCeXqT2ojA', 'Q9n39MgkWbvf6ohzQCDO7Q', '77NzO7uhl59ZK
-EYoi7n9g', 'at9rwt_tidIDDHfFeLFF8w', 'htatikeBPPYOmbiPP0lXbw', 'xQWD-K9GJKt25oFV0VXcS
Q', 'A69Q7QPRXs4rvbbpuDsm6g', 'O_mvwBwSz8KzZkozRPywdA', 'pMlK6piV8k45lhMTRomV5g', 'gOsp
pOXgtB8S7Cmsqc8Skg', 'i5FE5KHUWxsTgaOkmnyBnw', 'vyfsQo-estP8EfiIFMsL6g', 'NOBR1-mzfrvDJ
JEMRxc8KA', 'n7vjwXPlfG9kmC89IhFiWw', 'AufULLEHyH35-1Px95f97Q', 'ErsBXW-E-mWNjMc62TPQN
w', 'JM71F4zQwVDDSzqSkiVUhA', 'RiHqU26mvzK1kRhg5KfzWw', 'ctfrZmE6YZ9veDGcQSbfuw', 'p55K
b76HHeATutpPN68sdg', 'xbSPOPOq57wxCxzg3eJG7A', 'KLOs8RAHj14iuM70dMfUGQ', '6uM5lx0DCX4ct
Qlebtxraw', 'itXz3aLcWHzw5ccZXpXHjw', 'Hc6agWZ9jGCxsPAqQwlf2w', 'TK_fTr_VHwS2Ec2ZxUUKL
Q', '1-DcXw8_o_oibpBlnXgdEg', 'z39ulrgtvEMB2oeokx0-mw', 'xpMHUEbvHet50XDg98fJOQ', 'YiuR
V_Xy8lpLndMOWi0uAQ', 'lL0oNTVtsFhkGwILqmiYoQ', 'wLt28qh-46LFhKOeByWZ0w', 'R9FFFqYOUmvWj
aHqODfzyw', 'p4Jg_rO_AsCWZGMd_w8xjw', 'yyvwMoMifs1p1cp9B68XIQ', 'MTVsyoL3dajP7VS4iuLP4
Q', 'pJppmQyW74ejj8xCa2SHuA', '5MvIg39L6FmBrGZG5C2jjA', 'hzaqnSxfRYv_Wbp6CMKtQw', 'VFTd
1QaNvVVucagkq8ZJAw', '0QwV8PJk9PJDFaMNw1Fcsw', 'lGrDfsaWl9IDUcEeUWyn6A', 'yTAOBwi29nTPn
hu9wH75Cw', '6pC5j-hfP4xLZB2W0Wi1vg', 'JcJwC-NxaVJ6xUTYpC9AMw', '8nAj60xsBjcHXLLSs7cPq
w', '7WotaQG7M4JJqaqqDraaMQ', 'wiE8owjhxXYpOxCcYKhbiA', 'wNTZMYPQRZAx645BuhdTyQ', 'sAA1
n6_M6qllKlT-XuWkQA', '7u4J3l42DPIK6ghUu7fs5A', 'qnnieCU8EXZyojWuKg1iVQ', '5h_P5sVbGxtw9
ldH4sWZkw', 'HsKPpfcATrQ5Bb0ab9VQNg', '3uPesra2sEBRrO8kDwO1qA', 'Hm6495EXdwjKqhzvhLgsy
g', '5Szwd1CSepRXwAFH12nPzQ', 'GoTkP8VOSk6rqGXS3uQ-bQ', '8S-UrKLKRsJhv6mfpCV7lQ', 'WMA-
mhjrNJVZSkMoBD5ZWA', '4ulkvNf-T_UVM-e4dd0lbw', '-T4sYhtArnj2XNj7xoEeiQ', 'sU9pCk9IvfZtM
d0AmrRhvg', 'gcCs-oFd0veNsb10nX_fmA', 'gt03nuYv2BJxuq20d7OGkw', 'ZH6rqgDyfVEb5yZxRD63-
A', 'ZR232ZiiDJ3j3dCyMygA2w', 'JReQtKy2yfrwPXl8efFBKA', 'd54DWwCsmiZLQ3rXfTQldw', 'uBrI
2sD8oEHegtiz6g3k2A', 'KWBbXYiNvwGjEFrhitRwPw', 'sl44rRFiL-my03C0oywKmA', 'AonCa5s2nOfNf
ZYecxgIDA', 'HBFtzjRarXxDH8xZRSNsPA', 'DVANCbItlJLt3Eqi7CaciQ', '-EAx_2wa-h4YyNzYyWFfV
g', 'nb2wZQ-DL7Bn72N_h9NMXw', 'qjBt7pr7CcQsX7040RYYsA', 'n4QTIJq0uXVcEls_No9DyQ', '__Az
EBHrVp19B5RQUUg0vQ', 'zch7L6wKTPUQI2sVMxUqcg', 'kXuA2ffm0Uw46U3O0mcmtQ', 'zi0kxtr5gq0Fl
g7jYdIlMA', 'UkYIuhD8kw2y-h1BHNo44A', 'oWEdmTYl13VHh1A_Mkhh2g', 'Y-9yQMPlnRWgD40xAHl1K
A', 'BJUXMLNyQKjQq57BDWy1Ow', 'qFHOh1DMqJ2Zb_pvqR1Muw', 'PO8XNxGUizzzKGHmn24fQQ', 'Ldjj
wHSHCeqSvm7a73wmXA', '46ASXV3uv_x2QAWe6ocplQ', 'DzVKlwlnqwSSiGGzLmAp_g', 'ztpTP8QJIR3wq
j-xCw6Mow', '3Np7xOb7SFhgAUbdtvxOyA', 'mxDC5k-GFhdqZLttPM1dZg', 'AzV-8kuqyXJqRc-MNOmOl
w', 'Zs5TpdcmCAFs3SKsftnX6Q', 'KTO2pnfnbqclrNm0HqoyOQ', 'N9yGtVv_uvWmxt9XY7zPUg', 'DB0O
fzpdQQw-Q-8udINCSQ', 'gzvltlt2n97tZMnEXihbBg', 'HKHAr2pPrN8rVzUdYXxVZA', 'XpD6HNw4LoGDB
CGkMqxrnw', 'WgnmE0dd3lB71lRP_OdyBw', 'DDxueGylgwBGa7-hR2toZA', 'szLodDvllHAwYqkUVopqH
A', 'YYz9sM0_98ksQJUeHs-SJw', 'ac-LWMXfGE0ZMChLhYqnnA', 'tXxlJ-tDbFn-52sT8WOdSA', 'Tj8X
3qpdjiJpFAEqWILPEg', 'JTRxzA1AZ8Zfwj7FpH2CnA', 'AixLPxra_4HirfchjPkiWA', '_hyY8DGw9kNpd
XLVSn8OeA', 'xOK3i0wgmkpGu54BuIcwQA', 'qm24FjqzuhQqAWGm7VRSYA', 'LCEAg894v5-08hx6av0p4
A', '2YHVTrs6EY7Me9NwxiQVpQ', 'BpmYi6K_0jXj0gO4_GmVHw', 'fPHLPrymsyb6WSFFKoMrTQ', 'F9T6
m1YdRFreyKDufcyoOQ', 'eib_gPmmdNMZ5LLZs18p8w', 'kGgAARL2UmvCcTRfiscjug', 'LmsezECOd8IqQ
UYjTzWNSg', 'DVZf6SX19FVnzPyJeZ9-eg', 'fd68FgmkzCi_Tud41ex2dQ', 'rX_t4oJNwMTLMleZ7s4zt
A', 'wJuqi0qRDVOjSkyWhf4PIg', '1UJbOl-jKNIUjroeC4xWxg', 'J_uRQSKIxN0MCItOnJzfBA', 'ACLu
EsNHV7lfq1O7CHuFaA', '63rmu8axehcaFxQ3TyIFkQ', 'zhV2H02tC04MGbgch5sqNw', 'DJEnhCkQlQ2E2
z--aO9j0A', 'Rhb81Ivo0XQOXuE5-5JPQw', '0ypTVVwcyusnxx4Zbws_bg', 'UyOtYjnavsr-Dtjo6Xy91
g', 'EFg-ew3x8j2XPJn4e5t45g', 'IGkoathSlJmIOBDGo9vUZg', '4cI8eBbkYJ5vBaDSZ00RFA', 'FRBX
U5mAa9MCAXLCjBTi4g', 'J3wHuqConRziCBkqCxY4UA', 'Lw2ix3qoY7V15se33597Zg', 'k0jwdPwmeR3QX
HICqSVL-g', 'HHbe1svajOb3FOXdoAoGeA', 'eNrtdrGazNbdiQRmOr5EXw', 'n9g51fYo-4mjkvu6wXLrc
Q', '3AIZkvXQaThvVODrL8jEeA', 'J5ptmoQJiHzm-XqyNW5d-A', 'gLkQlGtokKIyWyNG040JvA', 'US2O
XYlR4EvFQyTSKSNMnQ', 'ECGa0INRH7Tr3qGDU8I60A', 'ygronrTGPlQHxUn8NCmaIw', 'd8cenZMDAAUzE
tfbIyfwrw', 'wWSRhqRAez5TF6ZWR4od6g', 'C8ZTiwa7qWoPSMIivTeSfw', 'NgS8s99qQIiHOszKd-m87
g', '71vvbcu0wgwu8VcJazSZZA', '9LZP7Ll7N9zkuWfA5DUAYw', 'B6jSXFsQkKyoh9E2u2ynfg', 'vEhO
UPzqO4KJL2dF32I8Qg', 'SNBJb9xiaRJ0ohHhugwn_g', 'x73VgCg2HVjo_SUDBQ4plg', '8W6N_Vj1trKUv
tPya9h1xw', 'SbM1m5nejHDD1pJePsar9g', 'N50csIgIytyVZ8VUsHxZug', 'u6sai1T3fSXsyF-fnMToP
Q', 'zQ_FpDX-GDPuai4GOldCJQ', 'bLjKUuAvdx7wbp6arixUfg', 'wF7_-zf7R8Rri7Kq-a0tlw', 'LkP6
rtLtC26GrKtgR-COhQ', '_1mBBkphbaKO3EQHnpBQJA', 'nN6rr9Z6DRYg2J4mXagWZA', 'PNcKplCkU7150
DmX0aDylA', 'vrU_kYfmp0Mmh2S5-FY-sg', '2iqG5g2iP83GvtBiqMRQoA', '_2TQOwIQMaWPjjY0bZxg8
g', '_HcZLN9uoS8urIDGyzew8g', 'pXhkHt7mAqhrKLebB6Um8g', 'mAuBHENS_DeFCQqrJC_UxA', 'qbiZy
hbuhTMENzs4XH-gzg', '0ZY6rM7Fm9AbHfG40HCLfw', 'IPcbHkP9eB20ur2oKsLxlw', '-Ll2lk5MuVsJIBS
BBnM3Tg', 'TF0wvOmYBOCRYInIlDnulg', 'jlpbR7IurKHCnSrGxELVlg', 'h2j2B7Jif_0lW0H6FgqGow',
'CLz9JgzYmyGNP5p5EERRdg', 'uCWHw0AcKwTjvxDl8RpHKQ', 'WYWjFF01WzwQNL_whGEkKQ', 'IrmsjAqxw
7UgZfA9imDfEA', '3P9Wnn0Z9TSe08bA8IIh_A', 'N5JPIH_0fUge9suPoyp-9A', 'OhqXsFeqCVVnxbWuZln
Gvg', 'oPG5VgtbV0DaDqYjBIgyXw', 'J3NDKW-yjdDXkboig0hAHw', 'WmAyExqSWoiYZ5XEqpk_Uw', 'm79
wRpkxwlxqCwd5e37vWw', 'RpC1jwOttGJcM5TIgscMTQ', 'mhFF84q9CZbmb5PqKzfGww', 'JqaFfOsOFM_nx

-dpKnFp8A', '_iXng1V68InwgFaiy-50eg', 'tlSSQwfHYJany7wPoTH46A', 'FBheGN7IBrFkGSNdv3cbG
Q', '7m4-lgi0etB6KvIQ2RBBCw', '9BApDAgV8POMQ6VffTz3aA', 'y9NvtHUzizvWFYSZ5FV5_g', '2VfhM
rTJ9bZ0O4PZmRmxKA', '5pAk-cfZQy2CuhdZrqYCEg', 'TMsX0mrf_wQaxcZXKgIh5g', 'yueiAo4E5vtAqWa
qSyVo4g', 'C9tCsaYVHdMcQeACECXDCA', 'bxEeVkcFEkaUn4cfgDGS-A', 'hQvmGmxWOhoVB7hbDRQb0g',
'-2pjspSYwR_0mtJmQcxzQw', 'kCfwfPSOiC59qMu5EkOggA', 'Wnpi3iUqULVG8GG3fdXWMQ', 'u2mMLkTFO
4qEEL9QFwUjRg', 'YD2trEiFhj_ehUNCpFtfsA', 'GoqZDWwCvsgwiJRZJ27CiQ', '-heP_e4m3mEiI75Vozr
GwA', 'eFuJA-G5a_j4j3qB9p2UCQ', 'mFOZOsPQOacWIMVSyXbEbg', 'wgVQ-y7Q7ymQ25VVIgZEmw', 'qK6
_sysqplVrEkq-MxkpTw', 'A4DM2Piz-hXs-o9X2dSQZw', 'Gf-2KwbHuIzp2YdZXvC3sQ', 'YcmbgKpGDhMu4
w2AWuIgWA', '2XCsVpAMsGUykrM-BBFEXA', 'qVePixO2kuK0Dr03upCWFg', '-ANkfLbDf8aiBQ7vywIL6
w', 'u2lanPBcORxbwJzsEbAExg', 'HQJ_2pzqz9DuPHNpTOmrWQ', 'D9CIW04wrcQzZNdCEoASAg', 'LNvw5
VDA9b2PXkD-zn3-OQ', 'Nu0oythh2v7FSfJb30d12g', 'Aq_fbQ4aMVvTCiskOz876Q', 'A0puRx8MGQsnk8u
Xtpw4hg', 'TLj3XaclA7V4ldJ5yNP-9Q', '0XRFyY8-dxmrDheFPpe8Sg', '8E0DGec8LNn6oDmPHmj-mg',
'LQWlBt00QZrvn7McdS_Z_g', 'y8ZCNq8HSGDSx7Vm-NYEOw', '270isWtFFYFtlBR5RZUjtw', 'NGaa4JBPX
l8GzdDWWbxb7Q', 'fMzU7cCSgv44IKN5Crs1mQ', '3utEqyNzFcxH8E8pm7ByPA', '7SVxpEnu760QstKGa3x
2Mw', 'zRiz9k2brNAW-AgM_TLk6Q', 'LRBeqLo1FhG_9-Wtm1irGw', 'B4KTa9To2jrqr-NK7c7kwA', 'fcz
QCSmaWF78toLEmb0Zsw', 'ZNpAVHTehFYp0rObGT4teQ', '_VtoKtLvDAjWttQ9wVD5kw', 'US2UWum3MaQLt
FGiONNIvA', '1F8RulRCjaQTN61TX9Hd7A', '6hRKotVFPYL7Ci0hipN9Eg', '1krwpAcRTPy6lrjAGQQ1l
g', 'rvY6h_FqTuB7YCu_TrWRkQ', 'UUfnmyEBi60Qw9tjuQ-1sQ', 'NhXOaROBjlU-IuQrmWrQYQ', 'WeX6U
jZSTeMfFBQGDfu1jw', 'yPy13LY2idzQ_VpoavbKuw', 'RTwuP9OtmJvMm91wuzxGSQ', 'AGOmxrO-YUnltlj
d0SI08g', 'q9XgOylNsSbqZqF_SO3-OQ', 'XKilChOBIMq0hp2kF8KwHw', 'aYpO4xX9yWRGfF09Co4SXw',
'bnVjfyPIsh1dQ8iC1zXNkw', '44R1syuo0et8sLtRnCojkw', '6o0NNvPxXHvNSs3NT0WkJQ', '8N6y30aGV
wtIYFraswbJeA', 'LdrGYCwF8-mc272RdBQSaw', 'D1HMFu5IMz2n49N8ga-IVw', 'o_LCYay4uo5N4eq3U5p
brQ', '1cR5-LZcSYcACxhGe3DEVA', 'yoQAZWDdBW3kQH7geFEoWA', 'BYFWt4p_KLgINKvSqpEojg', 'y2z
wsT-arc0H2rR5yKR2XQ', '6wveyznYgxA0sPKEfe4wOQ', 'dwrLwOylx8KvPers5hAahg', 'MlcL_KuqYyvyo
3U00MAVwA', '4NX8crn5WLSVjWH1fhLS3A', 'nWry09tpTwD9peA3ERhMJQ', '0Ypfll3kJiL5OrxBBat_Q
w', 'nFO4MFFynxIiO1mOY9W8sQ', 'sjG0UdiJaEwin_pvWuyxVg', 'BSTOKq5iEIuwzbe-qNzbkQ', 'VLWiX
2gzkfDwNAR3fwjtlA', 'h2lZY6mojN2ENQdU9qazGA', 'h_IuHfAeOzE8KT5FT04PVA', 'ZWi5gzaMROxG3oA
KikHFEg', 'AX5N-yKV0w-lVNTHyb9AXg', 'fGMaltkx1S55k3GpeLpTyQ', '08vW8jwsZcNXc9k1_H7H4Q',
'2z9yZ6yWiRlZfF_Lshc6Zg', 'zunSYtFT1h20xnQH_mTsZw', 'CJCvh6WSLd4Rx-VnQGsSuw', 'pSOpnpuqH
aT0gM9aizQsmw', '5YySKNsPmTeG69nT-dS6Hg', '0dTkjUHgc6OEsSk8JnRzsw', 'S8ZY1HEDoECqKrIUY34
JOg', 'euadoc7gbju3zFjB37x3Cw', 'kJ44AlQeQIaqbtoGHAQeNQ', 'mTmHkMOdv-VQdE8VCJ8GlQ', 'nQV
J_9f7xVtlozrFu4xmqg', '0bNXP9quoJEgyVZu9ipGgQ', '3JtdBvnY6Wzp22gt5zFwDg', '1VyqxZ4LawfHi
RghQryVWw', 'FEDX7jt7NmAusbDUTXdArQ', '3WsDt1vFkLyCQqb_qUtFVw', 'i2u6i8HWiVvMjcG5nwTAE
A', '470dZswdTHz5aaGPvIYNpg', '19lLsmz8dx97X1APkVNlMQ', 'JT5kM6wRwh7cNt8IAuX_sw', 'cp9eU
Ll_Nn8Mo5hC9obSOQ', 'cbc2PFuBzBAg7TXRPgXwtg', 'xRf8sUczJqHgfgA_Dns9gQ', '25uBXnNZgHDaovC
1VTENrQ', 'Z55xeJuUqA20PL2oY4nMcA', '_E9-6-AYmN0V-l6HsQk7hQ', '0qIsBt4EzBDCKrIviV55Ew',
'lLZfboZB3B0LquJNR6GmkA', 'ca_Q9OGfyjwOnfvXqmhdIg', 'GrIpU3rvhW1Ydi5DlKQQjQ', 'd42T37_3i
MNRrIbi1gq2_Q', 'f5gM-S0atZaG7tsW1pi2NA', 'v1se4CVd74gBqTc0iMcK0A', 'TQ5eVGOFr_qB5_BbEd3
Sow', '_pV81WTJRyppkqj9B7GF2w', 'r7V9Nhes-Eg-lH0cqfpFFw', '8ma1-FGZD63wjmrxyeajUg', 'whk
zCRr6zbjjA3O0tVEKow', '8N481iGgsUYA1-TtESOntw', '2nM6phE0Emudtw-feBLWtg', 'enVSF2CIt_vMm
sq665LSSQ', 'sdH7B1S-PSngGY5Hi44OqQ', 'DPwh-CgEIleGiZIe0Wqo6g', 'ZTC-ln1GjUB44g7wtfJVH
w', 'ciIdgFaC_gSBHtkU2m7ETQ', '97rWQ31yB_JnjEDIxUI3Og', 'cAsYfm9UkDFDjPRLR5zfWg', 'uF4lw
uRTFvh-qrbIDVW32g', 'PMNbT3Y8KDt7E979AJMyng', 'BpIXi537n43BFg8Q64a-3A', 'x6HA_v9WhLKMAsR
m1n12sA', 'LJ3nfy4DLLfLy6wA8-64Qg', 'Ig7yd58fPKZYFhH2bDYDQQ', 'x8Xx1-_hbjkuH8xXV-BrwA',
'7TRprrHXkVpjhE5WFggkWg', '4qe2MoR7r_UyaDrGe-20AQ', '2m3lE8p8T2nFRDxIUaRXew', 'gxoqD1Y9l
Rs2CFZMUvxbxw', 'CEfUkE9nqGsJeIpFgAv1yw', 'DzY5lpsa9tBcJtoRbzCEPQ', 'AaBwzccT3c7vAZUQS4q
ZeA', '0qfftO54idymAc-6LPVTbA', 'VkOKJOX1gqarNssLxg9LOQ', 'J92aUGPKStpBIwx__yncfA', '-MZ
OtQxxYX9CljlnHbqWpA', 'bZFRqP7s0Vszxeu8_IwYow', 'snd45oedwXxF9tsHUX_6dg', '8jEK8gxHYg9w6
sINujO0MQ', '3AWY6ilx4acBR08xNJnWew', '8JC-Yb3UDUv2FUl5ym1nVg', '8UBBkdxYDCnSsOVc1YOEc
g', 'qu3uqXJvrQJSqz1XHcN0zw', 'hMMVN5kSxReODk2ofxic7Q', 'CReqkya7EKcp5G_PIBgQrg', 'HW-cL
93tvU7qng1sZueKWw', 'Dg560aU9ymlloG-TOPx0Vg', 'ZmSKVX7q0GVXxmVFUcqGDQ', 'kYi1-z_eErTLozH
G9gBYGw', 'ehfZRCVBEFYQzAYx7NLOHw', 'bvQw4Bp68zR7r_-aLmgq9g', 'Ow2FFWLAHDIM3Lp3q2vwBQ',
'2hK0GUNB76HeH3nOUUMesw', 'ZVyQudI56A_5hg-2B_wjbQ', 'uE7jv7qItkIMCCItQRSIbQ', 'yhByTmJls
wtYvSq9qKrskQ', 'LTWY2Ee-SYu8ItVOebD3CQ', 'f0_QSzyQUvuuWpmJCVv_WA', '6nK8JYOfMJBZ2lpN2S0
-cQ', 'SeUINpGxGUr9pd7gWACU4A', 'frLdxQKCtofva9b0dA8Eqg', 'qiEqWWU2CMpOHr1tCIdIVg', 'uHq
9T4xEvdldiRI5k1WidQ', '_A_2A-bjADJTKBoWd_B5qg', 'pkXq4ZbKLRlGlMiEJzm0Sg', 'k0iLX341x3mm9
golh9wejQ', 'u7cUruVV6_B7bisu-SYIKQ', '7zpDhrRZRTGCkAh3SHbEww', '50ofEHGBhs1HGQYvCRnFO
Q', 'xBvqbRdQlIJcw1DT5Hz_kQ', 'GO11d7wyai9DVtHGV52OcQ', 'qbVAB3Xxxtc_OzsDNCmUBg', 'ItqL2
488GCmuFcKxHDF7Nw', 'DFr0LYHHTokLFCSniznohQ', 'uQgfF-fHN4vhH5sx2W4jaQ', 'NSg7k2UFt_HqM2m
vsw-n0w', 'hMhGD_Z6AkG9jC4URiHByA', '5yBkoRI9DSKH9aMV3Vadpw', 'li148FKNnjlwmC5Mq-qFXg',
'O2Sepw53kNHuf1VqJ85cLg', '1cuKOGuLDfD240LInAlbwA', 'DFlMzHxJOQ2yggfYqnORoA', 'dS02pxg9F
MgcJoQEohsMCA', 'EcFydWvO9D9jCBUgLSMzKA', 'egdj4-wXAzYzT5bmTAHYqQ', 'VESaDn1MpEIAC7C-SvG
O2w', '7z6uuJ3fuaurG4dmdGidJw', 'MNOW-LkZx_Ns7AdTyFIYYA', 'nUHHk3cmr4ia7u36YMZTtw', 'Am0
M-nmJZ9fIU-NBFCzSSQ', 'Hv-U6M-gVXf8jqzgWbYLXA', 'PmLa7Pu9Ba32fhvCOLOr5A', 'C6IOtaaYdLIT5
fWd7ZYIuA', 'ToNSCfbIbW0zKKZUl4cpHA', 'AXHAksv-Y84Ww5uVw8RdHA', '4J4J72Xr67GJYxxGt8rHw

A', 'eg0AARmcCCsnyS1WtokWkg', 'EgXJoFdQlcokmIiqgMXd1w', '4CJGQajGdc8JwFneADetcQ', '6UUlG
Uc_VjcFslmRle1vPg', '-egXQfAzhRvq3BAgug10fA', 'BqdywQzVmBdjymq7BfdnmA', 'RPn0Avnq-1syMwE
Yz7kPrQ', 'GTxtOC-bNmfnBrCfRUC9Mg', 'GMSSg-Ts1XfWvy5kGHTZlw', '6cZuyvWcccCrJeqWF5WPoA',
'3x175sXzgadiYMlLErUMoA', 'i1iaLhsO26OefrP-6Y41WA', 'Hdb6HUDha7XG9X5NnmitgQ', 'qYGB7R1vc
O2rDJOeJNDfXQ', 'qaBXZe4O96SDjnxyl6xmuw', '2t9wSXQ3Luadi_eW7O6Tag', 'S3DqpJzGe0iZtDVdgrL
Rlw', '7h7IlYzn2j2b7AH-801EtQ', 'ec5NsXiWB0C7yeoxVlhxKQ', 'KQonJ09FfP7SA-ChNDgXdg', 'yfQ
wWLYtqwHKiky1hiTBew', 'lmDxv2cNKO_bKrD25-jAGg', 'xm_DzmDnhtNyNnnxhSDncw', 'mDqWHJG-NQq9y
cwV07wjkg', '3OxcNCuHxHUSEToYXo4LIA', 'AKqC8waFcFS7iYhL-02pgw', 'uo5LWFJTOsla9j4a5fMii
g', 'VSRmWUjNkSSeV7PLBOuh1A', 'ltYDFHSVSc_ztIiJrndmeQ', 'gGFthU9GCGkcSLNo1UyNDQ', 'pqlRa
KvhnyrNoxT8yeadWg', '3gBhA4bNoPWQYeONL5NPQQ', '6SxnRYFR3GUzGREKSttxJg', 'Tl7FatOzF33ciAc
A1GdTcQ', 'Pponzfdq5fYm5Za7Oj1BAw', 'xt8iR1kTQKxbBc6tvYZxqg', 'ru8p3RTlk8f9LB_3zLXURQ',
'WZgoM8UuzxjZf1AaWx3qZQ', '086zTSYPSTZs0220Teh5vA', 'GUrPmYze-oF6NFDJusiSNw', 'eqi3jBRRv
CGQ6zpXkzlJYw', 'q2M1kDqxwSlQxKMyPX7Ttw', 'Prsk7SiPZNfgcPyygOdGHg', '49r9640dX3NPnq3bQia
h2A', 'XBdkBQD5GkyCFwLnEIhEJA', 'dPcoYTCV53X9Wy-n2lxWhg', 's74Gi6-mS5J5WnwGHKORfg', 'T0v
VdZebDAsYN-UttJKJow', 'mRncQJkKWVALsflOVvO2kg', 'ChN9DVAAoNHUY6JljBcPqg', 'eBUE5o2msLFOj
JvFLm17Ig', 'fk9sG-0ayVV7dGsar3vy0g', 'iDs0mr1LAb7m3K2L-KcAAA', '0U9fwlPYRV5cy9G9mb6Q6
A', 'zI0j9SN1aYoQSPOlL4E4rA', 'QO3DkQtPKgZu6G4MXPmnYQ', 'f7Ez84QWvsJPXwVpG9-AjQ', 'FEkrE
E6gKYudG4WC2gXPVw', 'eDv4o8Pg3dNLZwX5prO1aA', 'saiZmxzBIowJbMgPBW1Now', 'cJmDizVwKXChEXR
rNaCUDQ', 'VQn4csaGtX0mhr-VaZzREg', 'CQVYCo4OP_TbnZ3DFMi2Kw', 'olT2J66OB2HFlRymvqkxZA',
'Q8KFcuPwUh2v6BMTyFFyBw', 'GiMKnA1NjkMgLYipd-qNSQ', 'DfxlZ3t6CxbVyKMA-JUSmA', 'IWBD0ipfm
AZVKYaUUArIoA', 'RnrVGMWolNXYWzgHg1hyVg', 'aYsjSXX3m6UQUPcVk0yiBw', 'UTGqubcmDv1cNe5-k8-
OeA', '10nYcaEv8zxiPumJECLsDg', 'gBID8jIwsQIE5W_Zr0Vhjg', '2W2kRe_ewEBz_jCGmfdMIw', 'yj4
lNvmxAt42NvBCxB3WTw', 'E-E50dUzNkeCGNLqXQ4PpQ', 'LbgQK5B_5IkN77FgRJHhrg', '_3f44aWnibPGT
pmiiNLxQw', '9zW8DTJuCNAN-Y2XqG-xZA', 'vhxh6yKNWA5IJntxjREqMg', 'it23EnutJ6fOjeYQBDKJu
g', 'd1w5vqi4Dz-Gc-vFefpOTQ', 'FaX9JLeacAteTlNKgeGNRw', 'cYJyOG1ea1gYhsR_xWUNvg', 'Uf_Pr
KEpYGWBsL81lY6NSA', 'KqaH0TMidJb6AMAeLXmu4A', 'j5f7ERREO65Qvcrs7KQTwQ', '0wUw3vODMYJ9IM5
8ND5gAA', 'l68a-NyKoVJeoMVGk0PMTw', '4mtCpNizYNdXuJk5IpP5JA', 'r7KlsQ0TYCYCBH1s-b2wnQ',
'g3dUh9lpWTVIKxzNmkFp4w', 'UZysWMlFjuMWGTC2qYmIZw', '0IAOkW3KD1Dsx2hnwb0CSA', 'u00EOkApq
yRP9YLlI90TKA', 'i2KwDZV-Bs_BrZjGaBn4YA', 'jVQoCoLm8JHqbQI5jrB0LA', 'k5zGEHC3C7AC6gxIGOd
Smw', 'xHAlPHFluxyPWnH95KSJLA', 'Doa1UjBxGVTmcvTxjEDHUQ', 'kl0zgp60RM09pySxA5NubA', '1XW
zXgFvcu5pU2wj1ONQ_Q', 'w7YkAgJ2t5Q9zkfq58tpJg', 'AP6udBIStvWLY0T5rS_24A', 'wNqwKWaRjClmc
KsoJJRFow', 'q7Zn9s5wXAjhhSJMf7GiVg', 'V0WOxlZrIhBjcvlpszAgXg', '-UP82TPPpG_NpcsbDbBlS
w', 'G_PCzMHcOLoMyyEsxW1DFQ', '1I3vCdFcsSW3KAppt2sXoQ', 'ZuQ87LQYWRw3AHj-jEiQ1Q', 'j95Hl
oXgQiHWDdkkHwytlQ', '558AjR8esTKOKNAUvwlhxw', '1tQZrkIweF1iv2RrU_dxJg', 'mKBqGkfk7NqwUoA
9P77Qig', '687b_5uUBlitPUac9T_u2g', '5-ZF673BNm9aY6VQQtc97g', 'tQV5mTcDx5r-3JS4BpByaA',
'IHUXwBK3pjsCxzuS_g9jCw', 'ZZ43etAB2n_T53YBYtf8Dw', 'iJvrW5HPprsTlsABAdYeGQ', 'XPGCWhBzJ
M0bQT-f6wzfRQ', 'z9JL8bX2OkDrigPqXpyERg', 'wwdRyRFoB-zBjXmJ5ClKrg', '50EcfFlkuvc_T6kf9Kb
ZFA', 'VDHUtEnPh54OJiYow91d3A', 'lsYq1eKzU0X77LN6PfH1Ng', 'HGWIasjq8h6GGNPRXhn3cw', 'syg
EVJoT1zWGgv_OjTnGbA', '-B32HlBjDRafJpD_eW5pxw', 'Ig-WArvZ39LcRvH0kidHNg', 'r8xfTpifZ7gP5
Cd5CSQaIQ', 'VVvmWzmjmkasz0AqwQHSEg', 'xAKoiSjFt_k9tWh9RIsj3g', 'aTi0NVrcPJWbN6jAsJVcA
w', 'RoSIhgbHiQFzsgavFFaaeQ', 'ZnOdDbndR0b5p6eF-014vA', 'hU0Vgl_hkJaWM5uUq_9W-g', 'h2xSl
FE8yfPn4DMD-3hbxg', '7_Tx5bj6Ylf90j328qH44A', 'QNlNjDHxj1uonlkxuaJVOg', 'ToJ8ZZITRnU2NoB
TP80nHw', 'QmXXmqX7Fdszdd3qDq3-Jw', 'd1B7lZdPgF1M61E1zROjqQ', 'weYT-xJwz8o7mLNWIhD6HA',
'fwsJGulnozT2U6FefsLiFw', 'KHxnsmfpt_Oo_0iqfToZuA', 'r0wUI4JVDBw9WyRimmyVMg', 'agBpuKvBx
rgTUTOydISZRg', 'WvCV79VfjM3rLWqH-k_Lfg', 'aXwa1hVyCgYn3PUlrkadyA', 'NkNs-f6BzQ2f2m2hJAA
c0A', 'fIRFQ3cYna9UK3CwFmYKfw', 'EN7M-RJPfzbUxP7dtKvA-Q', 'pjgE1d3csesAkgn6rQI_RQ', 'd1i
H2EPxYMQLbKHN03Y7XQ', 'gSXgEuv2wf4QBwzu8ySbig', 'IENeTl3e9W4diDMwW72gfA', 'a56VaJVGI00za
JVG2bF60w', 'AZuSSh8xMwLtoapGa02WZQ', '6AUX1beuHGiaT-EvD_iL7Q', 'g9Q7pHMQJEctwBCi3umr8
w', 'WP8A4MKs_LdRZ3TG45QC1A', 'kDZUGqP3j8eiLKIExBWI8Q', 'rMC4XjcyxQtpx4ZejzPibA', 'WxHQX
5O24QWJkV3_5EdFtg', 'wXQvpgxHM-13dulOMO43Pg', 'RjJGTNH1LAL1pJJhqJnvtg', 'D8EE4gbAiXS3DF_
HNRJfAw', 'ToY2qQZYBIGXqKGqgx4QkQ', 'EOffNH1xPOh6WMaPQrVF7Q', 'h_TlJAgBfSXG4Yw7rkC1Kg',
'_wY1dHBXKN4NjB5IsIk4eQ', 'Qp1pHOQUssuX2MH_M8CXng', 'Milv8k4zZsoCaNFZugKtJw', 'cD5_lX613
zUyG13MT_wQnw', 'LASSKMzO-XpNvKDlfTcCIQ', 'U2zN6K22bXa9EgfAmGIEbw', 'rg6MxcrOUNqsvUFl9RW
PUA', 'VysCH2SU35qN5WlWJxujsw', 'DORWRp1AfwJwdu0Nfzuz3Q', 'o88vNyL8qw2q9M1rGgRSzQ', 'ped
2oFDSX-0JZOOxDJQEoA', 'RT6qe2ZAANWsBBZLhbTxKw', 'fR3LvglfLdiTXPrqsGaJLQ', 'GALRNo-AbcAHx
EdmqKl--w', '1noVqERUUdFh1iHVxPnllA', '8JDeA__M765evfPv-xO2pw', 'p7PjqH8_0KFe0ZKHOlwz1
w', '81XpyQp_ObJMCQ85KFMRKQ', 'Fm1kLWF2t8DdvIMs0zW2aA', 'iFHk1IeIeMxL30QrH14EtQ', 'xl4yl
_IFXmU_9bFL0zB--g', '9VSegbNRzNlGKfBUv23vNQ', 'CRywkebZtGq0lKcbRlOPqQ', 'XOe-I25-W_FnbYf
Qyafylw', '_46Sn7OmqF-odwJgfyPESw', 'bA4OlnXLTPTZ6GN0QRmw8A', 'pb0MADLnmfv5SzCVciVBGQ',
'UmHvuFL7DiEA7JkyskbFjQ', 'Op77M7U4GdkksPz12FikeQ', 'TEoY60poY_5k82q_w9zt5A', 'tgRwVGIhq
N3TrF6clvRf0g', 'LF8U2She0pRYl5A479rPPA', '7vEqbLlIaBCvcoJUz67yzw', 'sY3vAX43yUeZVk4d99c
jYQ', 't-27_YZKsk9tMbAiwQKCwQ', 'Sv5kY5jq0Quljf5fydJY-3g', 'H23KoR8A_iQ1RgDZhVqwCw', 'JM2
6JogHgfsBEFuN41YGhw', 'ifFTbGgwjXALO1PgMHzrTQ', '2dskMMZLzaTjXmJ4ZVi4wQ', 'jRteemWmPzYuT
xFe0L0xJQ', 'Aaafbip-U2aeTXbvinBLiA', 'R2Pt2kZTrONyedI0-65FWg', 'nRNhtY7ow_mXte9wmrtH4
g', 'bt6PwCE8Lc3YBFdx_qITzA', 'RU4VlU9LgbAEDMU3GfoldQ', 'SB9L3xDwYybGrYNhQPRC9Q', 'p2UKi

Ld0jVx17b4nVNOeKg', 'kuP0AMnSGgkLfNSt0wOAgA', '2Vv78ycxr9m2ZktiwTbkWQ', 'Mx-vxv_V-SQCe76
w4RmUfA', '5QIRfDmbYhVv7xDeiw4p3w', 'PEw095uxUWPYB-vsugwynQ', 'trhLX2fxdqjuv_hCVm4N0A',
'sLMUvV5osYtLYU3JVBiLgA', 'DD3fOuL8bo6jrEsSeQHx9A', 'IXPibgPATsJo1uVaevcH6g', '1I3Hd2TlG
_BX3xIPXsttyg', 'AR9wLh3lEzFxitOaDFWUiQ', 'wq7PyqXT3QZ1UsOP92ipng', 'cTSSA0PFK6_xJAlj_uh
N-A', '6NXn9bd7TEbmiS7e8yf5Zw', 'socNNfwbqsKoTvfSDA7uVw', 'uh-AusMLL3_83q0ywp_2XQ', '-gA
QMwp5b4x0XBqCtwNVtA', 'akjaV3M3qumD5HYkjU9dtw', '_f_6B-oiWSXklz9F4C0Khw', 'bCLFdVfkAiZ7t
1xDskb66Q', 'JGJ4FgFihTApMrjNn9kfDA', 'HIilP51OIzfmgS23vvFEfg', 'pUUmwguzGEAFXDrIkvzua
g', 'r8nd1J5aXG1bL5ctt8ucCQ', 'CInFos81Gf4dkBJQL-1jGg', 'LGg1MNLAYWmHfXzp6hC4ng', 'lG08m
1dziucV47k4olWjEA', 'dfkRRR7vHoF_0ApWjHsOHA', 'oDO8TUHqP5b1Dq60xITLyA', 'VIOt1BAZ_9Xig0B
f1EtIEA', '0mVScF9nfPVIkS2OlbvpNw', 'KYOwoMnBtNhwBRFbw0fn5g', '39KYyHuggTbIceHwpujX1g',
'ZDKi3qsO8lSSogrIdbLJKg', '14v766ba7jx8JV6BC1lJxQ', 'TqkBwRmoEgECElxeY0OgKA', 'RPuCnMVKN
aKsSJ_INclzaA', 'NfHxV49mtVRq8enjdq1CeQ', 'tM9nH90xuXsucJ8qXcuUTg', '7rSnmi6siHu6KJU3990
oAQ', 'OpjQft3XugK05MtWegHFGg', 'C-DyhE2OMYRsahCVQFIvxg', 'LXg0qZbRViO02GezPRe67Q', 'f2l
2Trs6c0DbWGMMllmJHg', 'ab5LMDx_uraOLlIiwfKI2Q', 'QpVbA2JJvXzEW7S6wDHV7Q', 'CUUpi7gtqx1zo
jEGJ6Z5ew', '6NfOfSxsYkWSgYB-oWRoZQ', 'hhlABhXouwdWF5YJ6-tXOw', 'JnEpFNLnj0TWOO9ZZ_Q4-
w', 'r9YT-wfk8DlMyq_q4a_saQ', 'ORRz8KqeiLFYYQuEYWdmUg', 'hf8wMckM9DMg_gp5Vj6PrA', 'PEmmE
IagYJ5EfA8xfjWbOQ', '7mqhV16NyehMAJlG6kX_WQ', 'LkCoTvna2lV6MPG4Xzoc8w', 'CBqhQFyUA2fVMF1
NmEEgtQ', '_BWhSAjVNmVSNl7pTG4w2g', 'BCU34iLQ9kHqslWU9fFYzA', '_RN13rQ1c77hAi9LU7nwnA',
'IyEdBo884XBPmNLhIp8dig', 'JRv7TrR012humEezHENUjQ', 'qM5w7JwlEQGYlGtrOlEDRQ', 'z_umk5Ue-
tUcnmcwejPCmA', 'mdqDcnfQfTIcxM_zaXWaig', 'l8q-Mj_CuEKgvUgnG7IkPw', '3NEZvGl12UVuJcUzAfh
Anw', 'vDHtHxgzZqL-qebSa53e9A', 'dFC0MPQQNu89a9LpfL4PhA', 'iK4GPz61Yj_cZCMKC2C9NA', 'GzM
h3V8oVBoII38x1GpajQ', 'T463_lixRwQCRjS0TEe1Zw', '_cdM8vgagBmo3S2PcAycYQ', 'w1y7rcJAokqFy
ySEhd4r_A', 'HQJmocHzUhcfnfAKKI82qQ', 'bM2OTIopnFoaQGLxK2PxPg', 'g98c3p2XC1ePPDMUODT15
Q', '2uk4R74YwMeBQFFnWj5HQQ', 'AAX8oeoIq2UvMDLghGBclQ', 'OCkge7lzOw33bqMs1244Yw', 'rhxTE
DhN6D-uSW2D7weUAg', 'vEdlJ7dtXRQgjxTdjf-j4w', 'rnuiIVuM6-wT7SJt_z80pg', 'FE3lnTyP4j54Xlm
HmxNHlg', 'JvylLkohihV4I6b24UCL_Q', '9tELOBy_iGPNpNjXzkekHw', 'nBFR96CJqRtm5pfQ3xMn0g',
'PWXJpyVyAKtVehw4suOimg', '1mzDvYWjU9lFKzOMHZFlVw', 'D8wdpaUA7cK73VEZ-Y6omA', 'ysB_jaJVw
GzgTa3s_KyHnQ', 'wTz9pYPDgg64PFM2v9gScw', 'k8Y5dMaWp8N71OO8jsl5AA', '19ZqbAxrTdEJbvwjJSF
sRg', 'zs_GJibScPz5Cbz1hoNanQ', 'GEUTx3re-HmI4fhc47klAw', 'jbd8Uv0r16Kwj3Ayd9vvuA', 'AHz
5oFLWuWvDNWcnooM5kw', 'dk5zOa6PtED2eAMExoRB3Q', 'T_wjLgPOPXry7Bea4MzoVQ', 'IgMhM28GtVdkg
7qwM-auLQ', 'Ke9WsSOUOdQaTRwaYibYpQ', 'bLmp1BzhbHJ8adLQ86loHQ', 'B5LOrNT4RdtJ-kFEXs7qS
g', 'HKMcpGwNg1zi2ghARQdykg', 'GBPEA07x_XnTi366qTrjjw', 'rJW1NZGbDbqtI4BcygndAA', 'o8CJ1
4Iz7g9A5N8CNfqWiQ', 'POmVbGJ5nd9aZTxTJ_FVfQ', 'IlQkSpwXtYygzHXRCoveHQ', 'Y9dC6RSXUtPC5o9
0ZnC3Kg', 'rLtl8ZkDX5vH5nAx9C3q5Q', 'F4Eigu2W0lpF4oEwa6lxDw', '8tbXmjYGsYFZXk6ppuwRWQ',
'dU4aXFeWOBoZvgmBSDdqKw', 'hBwbBaBqofCgvkJ3PatuVw', 'WNfna4-XsIZbCrpCF0ZfPw', 'MdCZyRN4H
frfnmJlvTugEg', '7b134wSLX8gN7G2sbLPZoQ', 'SIjLiu746VXQ2eI_mQnhWQ', 'o5Uot_QoTnTJB8CTKnP
ltA', 'epTMlecM8SZNjDhBsd64Jg', 'VFtBm5d3lZLNSnkZCSZgGw', 'zUZIhg3UPaLsWGex5EWdOg', 'QPS
q3SwbTZbTB1dx6Ms2Lw', 'uWak0g1TXVr8axztY8Bdkw', 'ObAgEif9mNN4VodOi7rEIA', 'DOJM58OkGSsId
k2qCUZnLQ', 'rlTc-qZ5qERiXFRUd4XbMw', 'w9EJaYDPKYBg8dkHxwNgHg', '6R80N0O5E3FJ42f94O5RP
g', 'SYoux78T6d-AVxa7FhD9LA', 'g3nLX8WMsmwH6gSJiPU9Hg', 'PUMROYfbVgKypcfJNxs1mw', 'WI7NL
rNZ90cj1vPpFOOHUg', 'Qd8hqPpawzkezbOZ0kXJGw', 'N9n6XgHfiaaP1axKw0cyyA', 'yaBTkNRszgPLez9
6qo3lew', 'l-1Ti_hHb8QIJFG0CbINIw', 'LuQD-Pm0zqorZDv1V4VSWw', 'D_-iaWZwRj5X6Fl6Oca-8A',
'82ZqgHOfUMV1xtpG2KfisA', 'u7fkBhl2yGR1ZGZMue2B5w', 'pkW2moNGANM2W50yqwaaOg', 'VhI6xyylc
Axi0wOy2HOX3w', 'n71jnaDu0mVNW_UJgGKPFg', 'BFZD0XQ4C3n9Uc7TmppaDg', 'TpH2dPNn5ehrluF1190
BJA', 'KVaCXKlrKSyqoLcB8WF_2w', 'GBgY4j4mZjB-eoRb359Nnw', 'o1IKKMxnl5lDR3-z4SucgA', 'ia-
Bk0oHTcutM7_Kq6Jrlw', 'D8GD--YyjXVtlYjDGZX1yA', 'xTZwuLbQtNtKl2rd5woIQQ', 'kbXND7Y3udL7s
ZA2786nCQ', 'EuWhvhW90cg3HLIHCnXzvQ', 'ghPh5QV6PV7J3Jl4MH8euA', 'LiuKdou7xBsV-29MTlb97
w', 'UHOFZkgIvb8sn3RSPMXMSQ', '8OZ3TKist9Ekuw5T1LKRkQ', 'QCKH_57b2Hi146J85ns6Og', 'uxEe0
lQ4XqncfDMMhZGQBQ', 'gVR0LJ5Dbfa3_bGOt-_6VA', 'q3kcZKVht4z6lLOaxRgn3g', 'qJQtWo10SSxifcX
w-5RGWg', 'jykemU4-CaDr-8Lnei-_ZA', 'QYES_c-F13OhsFGB4twmgg', 'wZougLAnwunlZ92tHbyZpg',
'3DmA6rp1NHqrq938cnRCmw', 'uklc05zRRZuYzB2X015Vgg', 'bxciP2Aw-DZr9fVkMJtgOA', 'HhE4bvGlK
B5nuN1odc9GjA', 'Bmk46CeiMrga526ZX0eXsQ', '7H82rkP108uQ_wXf_uMeRQ', 'IN2HcBun5PpyfQK8OC6
8iw', 'T3-UHagV42o4NpgqKJ2K6g', 'zmPzZYdZqP9MUfzhn8QTfw', '9tngnaDXfVeKtXP84pXSjw', '5k3
g8CGc4bXSN3mcjFzWIw', '_9Sj8L_XbVgSmExcvDLHiQ', 'yxHnHENAwM1wlwd5SdS-aA', 'CNw6OzOjme50i
7CF8Bwntw', 'Wb_kZ07Zsj8PLFcsiMeixw', 'lXhOAP6ncg9ENgA9ln_ijA', 'Les7_9h8xQDZuZbHhjVYk
w', 'fm5-O3_otnzaAFHYH4pYRw', 'IkkXHuygD4y1tquSLhnpyQ', 'ytDfLMUaaHsGbeJMIzWKsQ', 'NZUB8
rNpZmAuGtiFfIac9Q', 'k3esMpsFS6lG2oRI6rfnLw', 'blimeRPMCJggra7NOCpErA', 'mGDjwSebGqtbBRR
EgLaE8g', 'rGXK7xWyvJLjAG9pHH4brw', '9UtNYyrWnpMbpshMrZSRGQ', 'I5ab7grI_zRFieGVvDUPEA',
'LdEq_hOTltUD_PX_9_qlRg', 'CatKFlIPqxChN1Tu0LkYmw', '9NMPC1bfwcJ2IZu2k0OAhA', 'glRXVWWD6
x1EZKfjJawTOg', 'h-EjNgw2t9V6tiGd_Qxf3w', 'nVQa8qgHbdosO-ZDY9-d_g', 'VqSCQCva71Q-ZVcMsYf
Quw', 'BZRGP49I5pK2YxZokJWhHw', 'yw7_Ba6yRFkK92_srFj97w', 'Nd-bx8nCwoA5c14ugyVI_A', 'V2A
RgOj_0v92gSxlgfx4Jw', '6nxt7bPWPRJHwUKI2Leabg', 'xofFo7WFfHOkzbtoG6SgPg', 'DR6qPrRgeSwmB
KAap41Xfw', '56i3ZvUITU5MmP9tjhRCmA', 'W1NZ33Kh4BEZrjHqP1FD-w', 'BVJinahzVxJFLMA9bZY8B
g', 'tx54uB3v9-WNoKxWoJFgGA', 'le5mkF5BevJ49WnumKYZ3Q', 'OrVtmaML9UI99bsYrV6IQg', 'bcUjJ
rpo6OeQxWK9hY-gMQ', '1hBqG6Fk5OCU8IMxCqN3Vw', 'yQZL8MTulNh3d3de-N0oug', 'spJUPXI7QaIctU0

FO5c42w', 'wHfvDGojhXsB5mmAu8ulKQ', 'zCzIbD_gs9hoAkg0_R_Q_Q', 'IMZhBBEG-7b0EaW84Sttmw',
'E_KAdo61uMDvIvqFcOX1Gw', '809QEhbAKPf2drYxrbjM_g', 'M_Nhgosc63jD-nu3XgpETg', 'nZFcWvgPU
8T4JCjOVvJU8Q', 'KOIWgIoBEEIuAnUJm3HXOQ', 'KMpj7TQf2gGFwpL-EK5c7g', '_Fxrn9P_nvvIY4JCvD5
tOw', 'JZvaa_v0__hzdrwvMb1Rfg', 'xv_XC8Ey8c1Agc7La1pwAg', '8Ggm4j6gnRSKo7k0sBmdyg', 'p3Z
oNUziIevbpqVMoN-fDQ', 'TTP7l5g4QUVCpQZ9NTAWqw', '-MyqAKY4nNPM9PDv2wrxRw', '1Qs8pJAnTEdHv
0mTcKny5Q', 'HqXxZd-lapvR0bvVTlKnNg', '98f92DVyKhoQZ94DbTLVBQ', '0MRlubaLu5v3sI4vu_XqD
g', 'pXHSbmRWtYjXF21jqol2ug', 'oWKBHNGxTyp7B0Yr1dRxpw', 'jmv7eDUgx3cn1y-Qw8G-Og', 'NN17W
rE5W9oN5UnvSNT8xw', 'LNKdYODHuGqah_nUaDJx_g', 'PHrPDWUCtCmkwx_gzTMGCQ', 'Bfo9NUm-uYdNzL
MyNfy_pA', 'V0WEZ2PQtJGX0sWNe89A8g', 'v5k-__wHo2xPI25WO-dcQA', 'z1_SJGzJhhXxR4ftohOK6
g', '2biTJfc25MuCOXD6XCo-zA', 'oz98YFoHRrJ9u-BIUFvwKw', 'JZ7N_ugLCrooDzkiDYXLdw', 'lhPH
OrHafkiAtUeTDee9dA', 'lWcklENPjAi78Miv9EG87A', 'b8vpDCrjDwjXSr92JVp-PQ', 'BKo0QW3LGBzAZ
ug4doJrgA', 'Au3Qs-AAZEWu2_4gIMwRgw', 'GX0dUsYmVzLgb0aiA73-rw', 'Sptc2IOKAfM3CDd5QTCAA
w', '9v8BtiGLKWdVcCN5WK4s3g', 'RQalwji2z28KIPq4DwLXmw', 'xLbSUpVLC6AQHRZTwJvxkA', '1LfU
XEdm0O3qhEqkKZjztA', 'LP5MjcVLtG6TmXRykg3Tzg', '2EDBVBqeQ0fuDvt9m4P8Gg', 'FDPjfyyHNfy9t
qyxo27QeQ', '640yIMkALhRVE3JYu13OsQ', 'cD9d9XFoC_bETPzjpnRj9g', 'b-Girgjp4FnLdfwfLDIHx
w', 'D4KvR-cqveAMH8tLOIks1w', 'rxGMXiiZyxwnYUrWP1OJTQ', '-XLQKak-GsDt_oTw5MutEQ', 'oXxV
7lleT2lF2AwGZM6QWg', 'NzOrlc4LJJLRtfzoxkJFsw', '7V6RYBSmpoHsVkohGD_yGg', 'WpAMN4ebd28Il
4yamLv0zg', 'jIuc3nM9R0KWX3v-8OyLsQ', 'TT5WOWluwF0elhzJPrNnvQ', 'Cs6JRqxOADfeqli20F5Yh
g', 'k49tKSAka0GyvbL4HFYDzg', 'F1oGgK3NtTYnyRMxupiiuA', '_3js68YksSedX6bGn4-2Tg', 'y6M7
TDkbEID5urBb_2oQ2A', '4NY9iOBSwI_AKZi1cHPMUw', 'ts5QpC5ymz1m1EeumIm7Hg', '7yWgjGRb02JcU
tphd4te8w', 'c45AFDoQz5OJcZISlZ1y0A', 'OyGx_yDm6XoVDs_MzYjftQ', 'ZPolhetd6Od5_VhXPbFIx
w', 'w4dgIzMwwZvKsHaC47Ox3A', 'utcN2FtmIymOprcfFS-Tfg', '8p4at4zdzCpueAmSBaorZA', 'Vo7o
omW3yovftNleHAX9Dg', 'ZW7oLADt3iMhjXloUHJ7mA', '8ncc2JviL8dOl9MLPpYMvQ', 'nh3mYZ0cIXNVX
lQAoOt-uQ', 'mtfkAgyunxat9fcCvDLVRw', '5wnW-qphVh5fa6qfMBzDpw', '5SLqeubMzxW7ifBwmqhH1
A', 'ewix80iggo6HsqG1zQ2ZdQ', 'GtV0RPZLK2fLBbGQsm-rGQ', '8Xnd_YqFUIPg0Nflq5wH_Q', '6VCz
wT57Vgr3JA4TIeF_MA', '790JhUvCFexsA5KFReMhig', 'FOfUxCjxD5xpKM_7mphLJg', 'MoVzMz9uqrDQe
Qeazh3eJQ', 'y_iOYwgsEQ4RYfRQ6tRGwA', '9fTTesCwn9EO-d82ODJAnw', 'Lb0yp9YlXSh_6miFJCLqZ
w', 'qhooXf7Uh8NP3mX6YG3Drw', '9UNGbiE9k0fvMpCMJp0Gww', 'nGYaT5sAuqKkd48dqX_2Kw', 'wM-4
uxHkuThPWfSRaKXPdQ', 'J1ATkrYS0IUkV4u-bkltgA', 'ODhV53XhesCBLJVfGsJGNA', 'dLqqNMVnbF6Nn
gz4R60_Hg', 'dtE9hHw1P2NANIBMCp2cBg', 'VT-4bP-o0aUzRiosIj04dQ', 'nrOCJCQUgXwdUIwg8QHir
w', 'g3z_RHT1oBH8e8vKcAUhTQ', 'CVLN1uZgGUky9Brlr8JJSQ', 'JjKJznvd5f7Io8hFJMGbzA', 'WS9j
QRCTMN_TL7W_065G7g', 'dJyY3aXKL34vtCF3OP3FQw', 'oZWx-Udu1bV_jbR_MuXaAw', 'l5eF4e5dBrB9p
XtAJn3hBQ', 'huKYBfeFtYCOWs4I-GESSw', 'SfiNyNLmW7alXejl-m_iYw', 'pwFXtVzH_So9f3iqHCEhH
A', 'u6VZAszfuylriytDU66gxw', '-jgGsPzZ4W2TyGh7KOwl9Q', 'IpjuqXEJFa-tciHOE5NN8Q', 'baQV
zpnc1f9JmMpyGAHn4A', 'AODbZttUrVN79H_mzekhYA', 'ydaYTc4HzEbByz4HhNzC0w', 'mYHfwCjpImZ2h
fEn1pkFLQ', '209Pltw3TJ7Zy31RcO8XXQ', '0LoXMgGgnL4c3P8ldx5KVA', 'eAq_2RBtk_zmRy1_tEvnP
w', 'fCHvzdXkXfI2ORPFk1U9-A', 'z5EHulCyTTYqHRDWoyrnKw', 'R6R2WBSbeRRftN-vXETapA', '8zfD
UTELhJczCHhq0QuDHA', 'zEggXKgiDxiSTzRo07GFGg', 'NzP9OoNP8vBMRUSs0UXBNw', 'pZSjYxdRazshC
PbSI1Eu6w', 'EpkFuQVu3XBigTU_PnjzLQ', 'M0G3zi2UH1-ESD1rtomXkQ', 'dDdinrAqDdeR-LVhAuiOy
g', 'UpPqFMrWGClk9YYfM9Ck5Q', '_jpghfHskfOsN53gYZtHzw', '3Zi1udoY2iT9s8Q3X6tgZQ', 'UDWN
MHV1UqQAVgPzzWvZKA', 'rSbqoimVUWPEbNSGwyjI-Q', 'XEJKtrG2LR_wFjH05s6Qcg', 'Iay4jEsj8GHzf
1UA40ailg', 'Qelj8uPxf0Ub5CFyB37kgw', '9zZNilcQoiJicpenGM4-Cg', 'lIIoaWG_J96-cQyh0af55
w', 's2FbJNdU8vIOEk7n48HJ9w', '1Lu8MAjA8byyrwHQoYpiVQ', 'WRODj3cwZSRD0QKfapmyiw', 'Roaj
VinaapQgQxXqCg6KCw', 'Xgwpxdj4dSnKpYYd_07vBA', 'nGvolzdWXhC-U8U0ZLydWQ', '-YVgvwt1Jp4XZ
7QFmd-XQw', 'ZTs-go-xYRQOp3PSw_XGDg', 'KVm5lSaNK-fL1OsbukgyuQ', 'C1NCPmIP-sXfYm5Ywhjj9
A', 'HCrmPNeV1QzMESHHRBhOkw', 'SpyUjfaUXdhgAaVuhzNVTw', '26Ow8ovoXuCdduVBwOwX_w', '-K3Z
jROK0ml2P-Rk7ttHzA', 'OIT6MBzr-yxw2qtokoSXIQ', 'yC7VOAAvoam0rxEjTUd3UA', 'XJwiv5u7h1gia
xgYAOz8jQ', 'h1BJNxB6jPc6xGMACpqPoA', 'JF1FEhOHxPLgc9dJ_qOXdw', 'GsmmRjg6c059dhxsLf0YD
g', 'A4PEFU6xorg3Gr8MssKi-Q', 'akjw1U2vzDrNP7ybg8Rj8A', 'wFweIWhv2fREZV_dYkz_1g', 'nKaR
5Z9Qmqc4RsakLLX_7w', '2fmj5ng3irLWTr7CKrSxxA', 'BDKaQY0N4NQxTUaMdoxKFw', 'pVYLj3qq6-zdE
WFUiC2CQw', 'v-FX2WyDYOpA6wqpzudZ9w', '9TJD_b6WWp2l4NT-qZaH8w', '2ShjrKMCIcLRLCtafbqJm
g', 'v4ejNYOneqVyFU-yXO3wEw', 'N18SNshdyaB2vItkP5zG1Q', 'NkaFJUKzzQfMcFekrZMwnw', 'gIWh
LgPNT1TFWBZtShNKnw', 'f3cFc4_r0pdY5Wqb8W9w-Q', 'MOdILSTOFnq0iXHN9l8_zA', 'TWIg1Jx2jGiC8
UqaKpNorw', 'sFQsKwx1iTPSAGYOo1yTdw', 'Xo4tysJw1XdbYF6P44vbWQ', 'hKU9eYeo3gJnduUpWDCCo
Q', 'wgcZ6blNmIpIREuJN1vhXw', 'pk2kKnjUhl5-Hv2Gygoctw', 'bKhM7KzqdM0j2O4SOLmSLA', 'tqCK
gb6HTj2Qs9y04p5ZMg', 'QS6NNvy-k132mSImi4kIjA', 'uGWoaSdH2gOtl_Gjo9G40g', 'CvMVd31cnTfzM
UsHDXm4zQ', 'bNrr3RJGMO_gOrKpDudP9Q', 'mvRps_pPV9aw18qGRJ626g', 'NBO1pzogTGH8wejRcdyC0
A', 'XlYbPATxPQRJR1FOZ81yog', 'eelWS7zIYMM7ci7dEd7MCg', 'UoS4bUrlIvs1gqmHCoB_Hg', 'm1wg
M90q_3Geq8LywXcU6Q', 'neOOSMNWcVL0o2Xwb0goVg', 'ww2K-1OgVGJXNUd4WtfRGw', 'cTyEgBRwcGhWi
q6TZPMFHQ', 'QS5c0pnSNWbv5D9YtuT2vw', 'xwcvsSC8Virgl7CveVvB4w', '17xLGvnf-fSATC5fuk5BC
g', 'VgjgoQDmWTDXrWj3MqJRRw', '3Y27vBOZzbkBhelwqu0rIQ', 'I5tOYH0rnjDxW8ieL3TYyw', 'l0iz
zNMbD2lL9xcOnnMEag', '5hptLoN75O1O8_xEgQip9Q', '-XPEaiJ9djZ5FugRVauwsg', '8v9q-mpjdna5X
2VYCEniew', 'FbdB9KXyuvT8rC_zmqRSuQ', 'ZFaem6KoZxMy3pgZSet2rg', 'UOppDzNrQokBg--Br_9wO
g', 'W98MlFKogX6oGoO026bTvg', 'gc4rNAgbGydNMAPJ85FZag', 'SRSxFmtaaePlmmebx2QRgw', 'XfkC
a6Ma32-KZOpMSvQ0mw', 'UUHbKus9GdbL-bFervUoCw', 'QB2h8hRFcVokhZuajF7Wiw', '9zlwzi6s2nm-q

2fy7hNmtA', 't3rZpAy4CrPi3kuJGfuCTA', 'ngR1Y009kpsuqWRdlKp7hg', 'njQ8Bn6HtqD0iMI5ABX38
g', 'JQGSDe-kap-IuUSH5q487w', 'EuYX6_YkzKYmazHmQGH_Gg', 'zdZfTTdKbWgTsce5v1HvSQ', 'TQ4v
RT9ETlxaL2Bvoj0fMg', 'b-u85eePnSX5XtfEzIz7gg', '3U-MZ5ydj-s8oOJW7lk-Kg', 'tiPH-oA8Upuls
0__lZ6XqQ', 'cduTmD2AKlX6p2pIVsTzpQ', 'ZpYfzg_GZa2CwWflRR2b5A', 'rSBYu1pV5AF4WGJUWIdjn
Q', 'V4VW69mBsRzfa3mLmyBqLQ', 'PyeEPahaIMGQo0v3F2tplQ', 'Ikf6HMvFlXMEJLW4HH-tFw', 'gyBs
-PcHasLx7IF9B6OBZA', 'TmrLXwR7eTyy0nzxXhdcnQ', 'inkv3_Ja07wnzXvrc_-4eQ', '7pfs-dhgcye8n
E1ktpVJmg', 'jdP_qK6pEfc1XCRnEhlIPA', '2dcPh9j30SrIYO9tB2YXLw', 'vHNtj4-I_H6nTRPUaoU5t
Q', 'Ve05sG1S5QLSm5Yuo-96Ew', 'YEyx8t2Okqy8JDWHcF8C9Q', 'vasHsAZEgLZGJDTlIweUYQ', 'r4i7
uL7zx4u0M-WmWdZZgg', '6uRIFyibHWozuRjZ10-jiA', 'sQBMbqZWNvqII8QwW_6RPQ', 'r46fg3yYJganU
aF9mCzf-w', '8dbRf1UsWp2ktXHZ6Zv06w', '2VhcOYqEFiZKlFlPrp2VJQ', '_X6TvNAG26owDEKOcPe75
Q', 'BzsUOXhZ_i-fjigkudOLcw', 'o1Esx88FqeolMqajzjKZpg', 'rWTgooIaB6vmD5bKCJgkLw', 'vY1k
ebWNJBBLPGZZ9qbl3Q', 'vWZsmtNF2O5W_hcBRFK3lg', 'hLuKgo7G6us4wNQoFb7N8A', '0A2phOr1tuw5x
fc5Psc94w', 'aaton9Cy93Ae-7Kr-OEr9g', '9qUYm17IHOA5AgRfSoKySw', 'URcLQkuZWiAAvVnqAoNfR
g', '_W4Pgs4js74DZCMQ4d4jLA', 'rzhjzUy34KZA2cL-gJaDuQ', 'TM6LLYhkkV1VwTDkHE2tiw', 'yIo8
1MlQWemhrXOPv-qFdA', 'Lw99LHT54yb7ED0qlgujeg', 'U8l76MT2Ej32cFKfFI24fA', 'Laoc1ZOWoeNjR
7CyWGDX-A', '_-W0caBilETBPJIlIqbKSQ', 'mzM1y8_2nIdU52BHnCNrXQ', 'UlkTmq_VLqEYwAnh5196B
g', 'BJcN-MemVWH6vPdwbCzE-g', 'LYBDUrVgaeMuD62ADMdqJg', '-ebYDOeqkWCuFCL7JO5I9A', 'H1r9
_Qxbh0zGMii0ax1C_g', '9P4YzvP6My5ktU6s0cA7Yg', 'XY_bQqhZfyyEi_W5n4kQNg', 'XrfkmMwiaPwNN
3ww-T_MLA', 'QGRNWxVU-pbd3piNFWU4Wg', 'WBQrry43eaTOIPoJjwczyw', 'HY53AFJzVN8DMHMn6Cd6j
A', 'yuVXHT4DzKgWSz7_1RvDWw', 'oPjmenCi0xOw1WnXmSh_MA', 'DdEm24t9KK_C6kP5z1osDw', '6IHc
b7PC6qiYR9qEsBieVQ', 'lpZtfD-vCqUER1yGKEvUPw', 'zWli0lD5TNKpTSHjXWdHxA', 'Z82HLQBeNdrGK
GEzbI_zPw', 'JVnGX-4Rn_jMWB9picw-Vw', 'po_IfliSJIPzzvd6UfU-iQ', '-K8xcVpQN7jJLwNDDuj1x
A', 'RKDSCRbN17rlE4ayYiyEqw', 'v-b34wAC0rJVWk4dCtgkXw', 'uc_kCbu0N9hztmtSkSHEtg', '2t8z
2r3KYbBpZUH_uq9yFw', 'C5SNlxImCgHrtu3B3faSlQ', '9U1PH64XqbfjC75mDnXbsA', 'NduGDKc6hLcb1
S6OtW62XQ', 'wx12_24dFiL1Pc0H_PygLw', 'd0PHPwf_MZs-ZZ_d0iSapw', 'aqcxaC1SwdL9XjWuZ4Mxo
A', '08tQjvwbh5hEoTq_mJjtDw', 'RF74LWUXaoZjNPpt4782pQ', 'Q7SjcvCvxqsbqVnqU7EN1g', 'ugMQ
jVWUnW_3An7fJEMLbw', 'NtqtKtuFNLXFF8gqb9td-g', 'HpjppzvCJpYHzPVtjevACA', 'P5VfQpNis_p_v
Ah0m8xzsw', 'OXzYyZSHo9Dl6Qd0s_o2Ig', 'DJWC4idmjq_FmcOCbTljLw', 'HPxji3cX5H460ECCgtnGH
w', 'S3HZF5aANmhZoMkFkPMdqQ', 'NBWUrKMUgC6xh5yf10-gzA', '9YVJtEzVfjfNujAfy0ysHg', 'vn3l
ecMsL3kXVe-7hg3gLg', 'YouW_cDsluvPxV9rlDYvMA', 'nPDUuuX_2C_GZpjGifWRYg', 't5AWRnd4Eb4cd
jtBgBCasg', 'FUId2sKCoanj_P4JRgQn-Q', 'ke7LgKD9XV7tDw9HER-Ggw', 'LdhwY7eudLlx1KFQQe9Gw
A', 'bvJvqYTSnStE8QL5fSJCkw', 'gxOZMidTXSm2z_0AecgbGw', 'BMQdK1kdXCiDTaeaIYxmew', '-OXN
Fm3dC0UYT-yA4IkrsA', '5HWf00VduXm8olEyR59_Wg', '_AVSpDL3qhiZD6N2YdhpJA', 'SxHFDNhmPp7xg
a6ooQ1eBQ', 'lC0KGXmIhyjzghBUlVnkhQ', 'GAeM4pken2drf7CpuKHuLg', 'AQnk7fS3XdXYM2ZWSF_Wq
w', 'JnFI2B1H5DkhVWM5NehsDQ', '4XrpU7kpxbFqMNdDEsj-xQ', 'Ksh-pjTVoRC8EGMR7G-boA', 'JmyF
xjiPLygW5ToW775pPQ', '3LaLg5Zg7d3FW7hyq8AQPA', 'OvXnk1S60tKzH5n_CrwyIQ', 'DLP3XfRNhTVxX
lCRJpGLNA', 'ibEK-BLVXBl1cW2lq5FvRA', 'uJVv3nSqscK0_c8cZSlJ_A', 'x-JcwIoL-b3FmJ6I0G8Q7
Q', 'deAYF4xLVB9n-Y-fxGyhNw', 'WpG-g-bcIoi9s0n41gP85g', 'ZNIxRh-iCCynnUk5PFHJnw', '7ACT
IN1VopEKbmKXog2DEQ', '1BW2HC851fJKPfJeQxjkTA', 'Sn5VmGYLYkcEMVjvze1k1A', 'M2R5lDfU72VkR
Wsd_0uICQ', 'enXVISXPVpRv3InUhexQNg', 'I9mIKgo8cvSwLC4kIe5MMQ', 'ZkuPKOz3tN9iTUZrGzj3n
Q', 'Yh6IPYTO2YsunUl9aeyU1g', 'JXOVmy5n0aANd9TEKvMoCw', 'HRLkhkYHiM-awcj2k0FgJA', '8T7V
E9bwNoEUs_0RervT0w', 'dr2Z50uFoCvYkifzxa9IQQ', 'B8pmNy3pUAeJWg0rEtf9dQ', 'l2FAzcmrBNhmu
F7G7XdQpg', 'l10Z3-aGfOokEutif0niag', 'yhDgEDo-u9NlGJynZK8HsQ', '232eTYWlI4pnKRz9nh4nQ
w', 'IN49cEfDQTbhr_SaL9y1OQ', 'fGBQXv5-9eRBjwhm_EL2jQ', 'qN6SH4CKMG46LNkl8zL8sg', 'chPr
kj4zYNNXig5sI0Udew', 'beFBVr42X5PdzazW6AuthA', 'yUUnTJ4mGs7CxMIWqUSfog', 'C1SiUu-sGXL6l
HlRkJxfFg', 'ZcNOSxKpSw3Xd3ZL4ex0Fg', 'IJFzDrRikDtYXUMqvHGhXw', '1pF_YG6foh3aJqjVZn8Qm
w', 'AHgV8ONL8do0Dmlx9-tM0Q', '4dkA654ss5dMR9wG_IrVYw', 'UpdAuxn_6oxYpSsMowmgbA', 'pQZ-
yK9zEnvtg5NRqTowsw', 'C5YT9IID8c-x20EXsxHFXQ', 'ylGYTVXYss5dk4SRqQsu3w', 'uV5fELkeuJflf
8slZMhd5A', '3ZQsNZRW-OhGl29ff2QGjA', 'UCOssThZufcqehMMe6OA0g', 'OXL9-5XdktfJQ3WdEOPSV
g', 'ao0KOJbU8qoKz72TO5QRNA', 'ZwtRT0gc4WZ4Q563ifJQQg', 'u_wXnVPQ4d7yAvz7xaoqsg', 'Ju09
FUAava_EWx_6szpnug', 'j68jx3DUkW2_6txDzqTY-g', 'k9NXn2KCQ2k8wteCqH5cSw', 'Ed-2su2WyQRSh
WrLFdY_Xg', 'Y5B2-04eA_F5zGyHUb7OpQ', 'tRy3UZHBPT926UgAE9O8_w', '5Q4Xz6QEoVGLlEN_STTtH
Q', 'Fky6k88rN4_XN-wicz_9Vg', 'W2qM9vaFHiAonmogmX58AQ', 'fpRXn7pFleWb0sgOHn1npQ', 'ZP2P
RJ8DAxDamxe82CuOCQ', 'BT3AQK9b5oIAdVve3_Sh1Q', '3ceE2d7-nRLASFIALsnorg', '_tdK3xZ-xf1Y_
7rL4V5xMQ', 'HIlrU4ShA10pNwwhc4xSaA', 'RenOpB9KRRx3_MFWIzI3UQ', '5z7yBm-fIyfCF3CQJkuar
A', '7EJJUCl3lHM5mYlFX0bLEQ', '1-CrNyLusSNYdOhWCKHnlA', 'EmFOHQzNewGtdLWlvvosFA', 'PCny
1cwjCCu19VudKeCanA', 'CDTNFwOXygHDxxaiIyzzBA', 'FwzXiPL8nY3PmmHjRP1h_g', 'uO8iD4P9kHpzk
SXLLK27bA', 'qOVcYwNIOMtPnve8a16tGQ', 'CZAw9nxnBWr-AQHdRLvU6w', 'JF05ffi0qdheB9Ti1kACO
g', 'wKV3xCxJcnBnQ03eGp2bPg', 'Sj4ruod2J8tUc2PKkc0IoQ', 'ml8-oQjWKT5YBCkT9p-yhg', 'en_l
BIEoyYdByLHNGdWF7Q', 'dGfMkK_fdj6nfiSeEmtzrw', 'GlAL3I-vsN5dbnOP_vQV_g', '6rs7VCPIFNsM1
bx60qKqvw', 'TFtyj0Cx_gpBi1yFDKccmA', 'aLnH0qJhmI-KVA_rfR-8QA', 'Bxypvn3rP1fwamC8cU4PY
g', 'MTQ7_0ZDOd_vS4SjczQcbw', 'T8N53fAhYESlrJUMEsBenw', 'jcoXg7IlOR1LBCgtMLdUsQ', '-Do-
a2vqE9EoXtnV-TZEBg', 'VRsMYNzt0FQ__gUAOuRqcw', 'ESNUEr2Ex3eU3diQvQWILQ', '2bAbL28lhrnyO
lYiZ8yMdg', 'zFW9Mp34CBuAWGYZOt1o5g', 'nG_zliKHG2PUttJa8z2APQ', '4jz83Df1emVCgifPOfuoF
Q', '9GJHHkT8x6NW3x_nXyvPMA', 'gsdSJTKtaLtjynSD6hgQVw', 'QWkXgWM7jjIruQ4Q4wzZWg', 'ZJDo

H1d6mPtsdMFDFV7wsw', '49kzkutc5_UFa2hjLEFYGQ', '85uSaCejFtK-8oVyqN9HHQ', '9OZH1Ecw-qUkC
W5MS0NefA', 'o-B090oRMU1Wej-o3KGT9Q', '5-zDa3uehYmGqrzc7esK6A', 'zSsucoCEGxnw4vxmSQXOC
g', '4BWGVJEqqOnihPxnZbuEzQ', 'dYWzKUPzpuCEJJHzOeVqxQ', 'mEgD2AGWGKD_gs-Jf3Ypew', 'YzKz
F-cYKJnnZtIj7fe3_g', '9BALHxACEqnLtLHLY1cNlQ', 'mUznWCN0tScMnMcMFUqmpg', 'yl3YZddzKKGdm
0a71su3Xg', 'IX743aJcFbmqcmne_ov_2A', 'kNtuMa19HAvQzrAFjXc0RQ', '9FDMUyJStI86B07bKQleo
w', 'fjGY-tlAkyPbIqVWmgBsEw', 'sZUwd8g_crDamsZNvw2E_Q', 'aGpFWlu6vOsV1GRUmpWllA', '34gJ
_KlP3RM6jotNT6TcDQ', 'KQP4Xl3kKNALIQse1BeCfA', 'NJB8VnfJZdEnIQxawIh5BA', 'Sx2OM_pi-NsqN
iZk1fMNXw', 'OWwNgrFIzXnUcqPpTx_53g', 'LMJ-eXo_GEKUJwuwOEHzwQ', 'bRLPoRvnbJbh8SiLjQRBl
Q', '60lHfhyzSwACS1TRErqUjg', 'prt4Pvu69WGv5pc87RuLMQ', 'BkxbkiTa9_9A7j-0BRGEhQ', '8muA
Bd0tSq9xrjCa9TltZQ', 'Q3oLto-17WmWK5-iazpZXQ', 'dbzRMMi4ZWiH37rt7RNizg', 'sQBk81qRtXTx2
8SHGqoX6A', '1D5xuAHi9TXN-G5h9tPntg', '_AJekjxvLJlt7i9jXGvumA', 'Pce6723ATKGKuI4SG8JFY
w', 'Q1fpW3qiL3oOe3b0si-0nA', 'd_KgMQIH2ekU789WghwKvg', '9LwyHtKt5iRqlcFH-897aQ', 'We0B
Lw5csPSuodlGXdTXfA', 'CWqDJ99YEXmqrbXNy7NHLQ', 'PkPyd4PZvlZ-3O2ARq8Oqw', 'R5yR3y6Nlvm0T
svnHXkDSA', 'V0if1DQ0MaTY-shWddXCsA', 'jqlPckTumnpCSvQ9jkiXwQ', 'AeucYo8J-rZjcq09Wuqsj
w', 'LhcGcnw4eHaxzg58y2YYUA', 'Pr2bD8IxmJPOt3MPcwJbJA', 'sPEFql7fLp0oouFxnooGpQ', 'PUh7
fqdsIcLwA5LxD-WCSg', 'rzHZ3iYVQe_8h2e42DbUsw', 'lT5B7K8p57jQ9KlkwMkPmw', 'jdyV9MtV4_94V
DP9aYqC0Q', 'yA77gvXdxFViK0mZcSiveA', 'K6MrbZeqKG35ZiucFfucvw', 'UD7Y1CqfY6mDmRwIuCf6n
A', 'zo_32oay-tjJ6ckos-vVCw', 'xzAHL7X8LDiSUdFNtB_APA', 'Rf4ldVXX_zHwRauT6ICk2A', '5OOu
iG_basiJFfAln8UtAA', 'voqhxAies6n_95l8usebsg', 'uRhxqIdyu_jjoDKm8lSEDg', 'AiuxEpAxjPN7c
iIrNupyWA', 'CxqXgJ672n08peXURbyUAw', '1DB-S0AKdjkUhUBzSuwB6A', '6VFHoeOaYCwZMgosAf08X
A', 'r3wE-UMb-ooPsn0805v-3w', 'eVTcstm1qKf0U7oppUZQFA', 'SgDWDjBr8fV7id1UnFaAFg', 'GJrG
PKF2xxB06Es6aH1VWg', 'Z2plM53IU6v6rVgBSgkopg', 'mqyH3CqjMufsm5FccDrjLA', 'C0F_DP9yO18mL
F73IKhcGg', 'NrgIE-qEt-qnGWdZJP_BuA', '74oc3KCvVhvmo73Gk2gv3Q', 'VrpOQV5Q5K_oIX-pqW6_O
g', 'lmiDCrmas8TxRsbIGZX9Pg', 'gFQDzGzufke0NWzb_AJ4oQ', 'yY1MgFvfDVDhotp3xWJLlQ', 'Nf_8
RYY4dTnvwaz96KMq-w', 'jKn-ehjxG6l3o8gy1p3ZOQ', 'QiSSo2Kg-znygvxJz_67sA', 'h0UyMagnO4COV
H5KDH3hZA', 'q6QBo2RC67GIHhDLG7ssDA', 'DHZ-kZ9sPDMmpnyFGXQ4hg', 'JXLIHoA6NGmOl2LznYH3P
A', 'hjvsUcztfGB2u130R9WCug', 'llzUppvv0t4noKubpIWCqw', 'DblIKwFdkkTNAeft7J4Taw', 'h4tf
HGODErFCFrg6KEZ3xw', 'rnMXl9whqnQSZtBFtlNy0g', '80c4xyemPV3XaMwlFtzVvw', 'lbg82t_mmEvlk
u3uVXn5TQ', 'PTqGpKNODTChfyGTRskFzg', 'ja7jzAqT7purefjl6wBxRQ', 'uvoJwcQxHfMfqDIX3a4xK
g', 'mRVLpO33EE6qHJqCfB48MA', 'TPracyTuh0oJikeyTA3w6A', 'wPjdB0viLu-i7YKGG3LJqA', 'jvhV
RQ--8liQcdJvqDXwbw', 'hNqEqYaRD_wLgShfn0Kv4Q', 'YaDoz22xUMKYKgrtK5cZ5Q', 'VaBcoKQHmRnwt
VkUypsvwA', 'G-jipHtQxKH5CLa47mn51A', '23ih9Fdu4v2KpBE1JDoVJw', 'WapiDiv7IjzfpIeczYwMI
w', 'HcRkWWnGJ_7iYJs30zXrxg', 'IT0sp0lM1ETIPQO1Z0c4oQ', '0CC4mLhyxpiq6O4nQpnQqg', 'v4Iu
HLQsLuJ5vVve6atgzQ', 'cRyNICH0mhjxagvSyVr60Q', 'Ti86eeLrIhpNHDpjbla_fA', 'Ze4YIsiTXq2yr
NM87Ue31A', 'N1EMQkSiTLcWHyG-VweCXg', '8n4SarsuI9Km6lEtydiE0w', 'HKTWp-uHABEbNIFK3i4LC
w', 'amr0FHCs4EBYmy8_D2UGGw', 'CdrOG7bBwAgHfkskPlA8xA', 'V4ND65sPYjWqUbLvpy7C-Q', 'NjOY
Sgr2LNMoSPi4e140Ig', 'ekfSera3-_wJrAiRaoKJ1A', 'hp6eRTTck1IQ1PwHwJx4cQ', 'z3Hf0snMcvVTR
--nC4uOLg', 'zZlHqWiCrCj0WKSNI1Nxlw', 'h5c9zka_FuMAIpDZ2X2Wew', 'I37ejvLDijHdqv6YOoVmI
g', 'rKMJaCa-hUjeZDXEa6rKMg', '1sDu-W8uXNZZLuWGcohy2g', 'U9j_xFnRVTXJGe2hNUZX7g', 'Wmhv
ywSbr4B-JRcdoHwwVw', 'kV5vDL9e0HxN1FuxxSCBIQ', 'WTEcPmeEf5Pqv7diNm4wuA', 'y87DXWU4qy_OK
tPJgnQFgg', 'L0Xrx4zdjPtPDcLcE0dJMg', 'NObfIhH5Lrdrw6PALek9LQ', 'lClvoNr2cNYM6hC_rG7yE
g', 'taQoRWH9dr0jYPMIJC8e8w', '9hPoAujvzKe-pxeCaHchgA', 'LsWb7sIq5Ywi01GIYOZzmQ', 'vLFk
6nslFe35NKvlU2Tz9Q', 'nkhJYqK24Plr9OHciX5FUw', 'gRxcP_zpGnZ5PFeGgFM3UQ', 'e6f6ETqM6c-SF
369w0wqDQ', 'AVMN_xKLGaUW99lYK4XSHQ', 'bIzXegnjDpfWJazSh6Fu8g', 'BRydUrmdK4pZsbUvZ8tBA
w', '4q3czSXkHBPkW21RdzpJ2A', 'Z-TW_TppnxT4TkFeavNaQw', '_L1NbjxnLRJMmmjgVMdX7A', 'DqQf
_U39vTcxd4nwNSbZ1w', 'c1J3fJozDVHeICaYc3t2ew', '4FJKLPkl-mvUD0iV9Q2p4Q', 'r3diS8VPXKbRC
srbPtXFkA', '_VeWBpintcoqjIcJdJ0_AQ', 'KyIPG6zBQj9Jqgl69tHtNQ', '4JMLe_6FHVUD2ZRR1r3vl
Q', 'gqkKPS92L_qG03gHaEtIsQ', '7s9BkBonSqGHdVkpC_qdbw', '85ZDaEjICAF0-6OnvbeFag', 'E4Hb
TIHd9PVjUnEKpysaLw', 'ANetBXp0CdITDGhp17huGQ', '7lAr0ypf1jEA4qJGCWYaTg', '8qBnLmI1xd5Zr
HeV7fBJZw', 'bAyT4kIMky9Gn8eFeZlSnw', 'D7iY_stJhsdLxbiOR3gOYw', '6wX7uHlJMl_kaGvhNpNTn
A', 'bN-oNETOC8RPVRWdBfyMPA', 'VFL4LY37bEvyDPobGoxvXg', 'fsSnng8XoT0sOKW472RQBA', 'Kx_I
T3LiBivsuoaKnOdNOQ', 'TdjT7TgOWhBxhd_lgGygCA', 'xBOjmYlcTheqrvH-CveHwQ', 'IHt7toZZ4RlGv
vz7aaKI6A', 'sVqdFnlxtZxYB80ZzZH8kw', 'FvqYM_zHngZZ5JFG1hLrtg', '6Jh0nU3Xey6J0YfpSJLUw
g', 'Z4Kn84VryzeJK-cGOfm6Pg', 'AbO18pf8bXG1qF9bb9L6wQ', 'QFj64b_eNwQaeJTvgkcvQA', 'fFmL
NPsvu-mT0ewk-sG37w', 'zanDVZZ6u4k3DaGbOa_pKA', 'DODrRvjHBklw-W8ueb1CtA', 'FTSmBX-WvzxFp
H_vOiRIkg', 'x7twQq2PxhdPuKTOOhWo7Q', 'hG7tKFN-REC22Zm48IP-0A', 'mdJZA5p391bCOZnvAxHV3
w', 'IDHrwv_RCildFvmfWTkj5Q', 'y_jn16HpJkfW4NfnKjfXLQ', 'x3GYZy8dMMw4j7eq-JN7Bg', '3DJL
AIE0soSxYdvF50L5jQ', 'dinZ3pxeWAG9ouMutcH63g', 'M1EQRRC91pofvyUucOEmsw', 'IyKosl07zfd4M
1bMGNpxJA', 'e3RgGXBrk9BZ4utixycubA', 'HU4k7g73kIZHAtp6Ajv2-Q', 'XAAs2Y77fUqV26C2ec5-7
g', 'iZkGpLxDooIvqj96CS9BNg', '43WQN_FXtzMDYHUqMeaoTA', 'p7xa33mWZkhV5HECw0Kk-w', 'uuy_
NVUeF1Kfc1E6R68MfA', 'mtlNpVemf0u2lIIMgqV8gw', 'x1E4U9wXGElDDXIBzp4b3w', 'VYVxxvbc1g8aH
7kFCP16Bg', 'RTnNGiTz2_EtTUtHGw7IYg', '1UXOwFKujaGNCNCijrdbsA', 'fwkidAyUXAHyXWtMxH0Wn
w', 'xcmFjlhZj1Wq-oeYEd8gfQ', 'WhAYw3hYWFA647Lrf2OJNg', 'n-0JFhD3V0FfN7sVmQs5lg', '7wSi
6CwaxhR5rbphALIg5g', 'utIPtVx2gryj5cWJzb1iYg', 'FFluq9BpoG4ADhi9gI_LzQ', '7E5cTuA0Zt3xF
bKNKaU9-Q', 'x3G-ZnbYzcFK3n2-8havGw', 'CK_Hg-YIpS_WSkN6iSWu5g', '-3wX7RLYheuW65F-IOF8K

A', 'pKh4Pc96YbOg-6hBOOajAA', 'sKtmrRvPMn5GRTnHYd41CA', '3nKW07v5Ot1WkDESYHOgJA', 'L5cv
v2zsyP6AOZ1GjCP_dw', 'QGFLP49noPFZtFePsCcT4Q', 'yvATWMLzso1M0MQaqxq5Rw', 'oOayyTrVO-PbV
eXXsh_JEQ', 'aBs5NRDe8q3Di7Bkreg92A', 'So32N7bSbUd1RwhFtI6jTQ', 'bjw9ZpD3nwlvt7osANXLF
g', 'BiZvuNKi0bs_YwttHhhq4w', 'DjuPUzqoUTYi4ecFxXHbZA', 'v2bkiiHE_hJbKcyvlCk0_Q', 'f5ud
yzZy4SX-vphL_gUcBA', 'j1ndTaipSt1ATuY3kynR0A', 'DDPwyrdmtcQVkscBtn83Ow', 'pfITP0e6VRrZi
JphzxUjug', 'f3r3W8YPUcgRB41J8JE5EA', 'yvws3aMGz76FiQKIaatqqA', 'fwEPqYiyZCaB7QKY6a7h5
g', 'VeBhqVUGbFbjgD08mEewyQ', 'pnDIFiXeU9tm3KtIISe64Q', 'DvpUCL8h6QVU-Cb258cCeQ', 'U6OF
bI210w6QZlsCbXZZtA', 'LLZu1gmhfEgCgffnF1kZZQ', 'GpPKhPQ3mMX1_NJRLYcVrQ', '7kZM6JmGp0K-k
fMWI-Cu_w', '3AgV08454BMMO08vqBUJRw', 'tPTQ7dOR63nSkG4bza-lEA', 'SntjekOZ1a2PoTPyMnYWT
w', 'v6rb-4YhADpuD99f3RKmgg', 'k5yMFDnQrBbc82qxRr_a8g', 'M5jekfwRaw52RWpKNOkbqg', 'Mfho
YW96OfXcjF4TgvxawQ', 'DZm20Z2tH25wKfyVkpmF4g', 'B_2NDDzcZELoxFbAPXwJLQ', 'fy8u9MsHxUk1h
78r4AlmGg', 'kRLZKani2Te8eSNqQaz5rg', 'dHVdhgc-9Wf4246uCkuAmg', 'N7wNlo3_EaHUWHiFDyGh5
g', 'e0XJXCxhA8zTHG3_dAxpeQ', 'cSwEein5F83c1VP7n69C4A', '9WRfZHW7jYoj1JyYebA0_Q', '9H2k
QNvpuHpK3YK4-r2R0Q', 'U4wiRY7GgB_SqREcXpewoQ', 'BGLDcowP44WPtsNbrP5wtQ', 'VjZZ_TKemzg1Z
4yaJM6J7w', '2ELMxD4wdjzy9HdjwG0JzQ', 'RsUNADZTrJ5xFXVMXJC1MQ', '6urFO5AiJCIwmJVATmWCy
A', 'OA9I8dKT9Uims5rp7YFKzA', 'zHJelsCN3Iknvc-0dRoktA', 'UnGPr6ZnGe9YOSZRopnS5g', 'Wzaa
orVCmUTQvu4mScunNg', 'CdXusNhIEfwM2nJaNxGqNw', 'bRFynDPNsgFTEOLU2Rz7Nw', 'yuB1BSvEuPjWm
BG7PIGsOA', 'Qew7qNGE8ksnrZmwrOvAQg', 'UzsKmJRX6x8E1PX6ClSw1Q', 'ZilAKKl3F4DqWSZPSnZfh
w', 'RYkCobCYYZ2HqL6TdEqoqQ', 'yC1giQ-YlFaF2m0pgLRcow', 'ZJAYnSbP6-38rKCxcNZIGg', 'w8pZ
YayZQY5PmywZQ83KRw', 'ygv557H4_4VEqlDdPFr9Sw', 'rpJXWxl83G-b3yQ3u9rCTQ', '5ltY6j-HCw9MV
er8puq65Q', 'absZ2-rmeH4NKo2VtWWDUQ', 'JLYqE7ITnkTupHgyOM6gvg', 'ADhJUB2UfJ88aRNAIuzXl
A', 'ihUHxlE5erKlMaxl52fTDw', 'L-cexm4-3KDqtL6kGfjzBA', '1K50PvXa6l5PBXZXhzkB-g', 'd2Ha
Q8x-3kF6DEg_tkOI7Q', 'GdXDLrMruYKSsEcb9Lvg6Q', 'icf8Tr75xv-QOnI7wZX9Xw', 'B9UvDrXUpZTNh
jrV1BO0jA', 'ClioP8ShELrgCyxoD4wK2w', 'PX3bqC0n1syKPulnpe9NFw', '4qRz_FoKPwHNvPdESPa8x
A', '03vp3tenQ0aQOr0AE5j8Uw', '5mEtUgrWR1epdj2VFzwMaA', '-gg-WvyzPXVjmbqMIVBP4w', '2mBi
n9oV_a3QnbjidmbVaA', 'PR-EPp7Wru_Igb-ir4Yh4Q', 'ZHLEWpQkQ35bY4QnTCT51Q', '4YfoQcMx1CFOZ
ghBnzS_6g', 'ofEx4LotDyaxIjWB9zbQiQ', 's4iBNiw0iy3YZU3R7Os-pw', 'dkmeYJYEGhAV0kijSHpef
A', '2CjqjRENwrrcEPtM5fAujQ', '46QnJ8IGQBxL0sQ9cq2f9Q', 'NgnwzZ9YLaKj0HnRKhUH6g', 'h-3H
tyPSh_1270egSeak7Q', 'XZ_484uR0Q_YHf61D5RO_A'], 'fans': 1012, 'average_stars': 3.64, 't
ype': 'user', 'compliments': {'profile': 110, 'cute': 209, 'funny': 561, 'plain': 921,
'writer': 290, 'list': 37, 'note': 589, 'photos': 287, 'hot': 1032, 'cool': 1521, 'mor
e': 129}, 'elite': [2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015]}]

The third dataset we are going to load is information about business checkins reported by users on Yelp. Each checkin's information will be stored as a Python dictionary within an RDD. The dictionary consists of the following fields:

- "checkin_info":["an array of check ins with the format day-hour:number of check ins from hour to hour+1"]
- "business_id":"encrypted business id"
- "type":"checkin"

In [ ]:
```
# load the data about business checkins reported by users on Yelp in an RDD
# each RDD element is a Python dictionary parsed from JSON using json.loads()
checkins_rdd = get_rdd_from_path('/ocean/projects/cie170025p/shared/data/yelp_academic_
print (checkins_rdd.count())
print (checkins_rdd.take(2))
```

```
45166
[{'checkin_info': {'9-5': 1, '7-5': 1, '13-3': 1, '17-6': 1, '13-0': 1, '17-3': 1, '10-
0': 1, '18-4': 1, '14-6': 1}, 'type': 'checkin', 'business_id': 'cE27W9VPgO88Qxe4ol6y_
g'}, {'checkin_info': {'22-5': 1, '9-5': 1, '15-1': 1, '15-4': 1, '16-2': 1, '21-4': 1,
'13-1': 1, '14-4': 1, '12-5': 1, '12-1': 1}, 'type': 'checkin', 'business_id': 'mVHrayjG
3uZ_RLHkLj-AMg'}]
```

The fourth dataset we are going to load is information about business reviews written by users on Yelp. Each review's data will be stored as a Python dictionary within an RDD. The dictionary consists of the following fields:

- "review_id":"encrypted review id"
- "user_id":"encrypted user id"
- "business_id":"encrypted business id"
- "stars":star rating rounded to half-stars
- "date":"date formatted like 2009-12-19"
- "text":"review text"
- "useful":number of useful votes received
- "funny":number of funny votes received
- "cool": number of cool review votes received
- "type": "review"

In [ ]:
```
# Load the data about business reviews written by users on Yelp in an RDD, limited to b
# each RDD element is a Python dictionary parsed from JSON using json.loads()
reviews_rdd = get_rdd_from_path('/ocean/projects/cie170025p/shared/data/yelp_academic_d
print (reviews_rdd.count())
print (reviews_rdd.take(2))
```

```
62608
[{'votes': {'funny': 3, 'useful': 7, 'cool': 7}, 'user_id': 'JbAeIYc89Sk8SWmrBCJs9g', 'r
eview_id': 'fBQ69-NU9ZyTjjS7Tb5tww', 'stars': 5, 'date': '2013-06-10', 'text': "THANK YO
U ROB! i truly appreciated all the help i received from this agent today who was able to
removed the extra charges on my bill that the Pasadena Verizon Store on Lake was chargin
g me on my bill for upgrading my phone.   When i went in i was having problems with my B
lacberry and had to switch to the Iphone last week. Rob from the Pennsylvania store who
i was connected today was able to look at my bill and all the notes and correct the prob
lem immediately. Great Customer Service! He even set up a FOLLOW UP Phone Call with me O
n July 5th to make sure the credit goes through on my bill...I can't thank him enoug
h!!!!", 'type': 'review', 'business_id': 'HZdLhv6COCleJMo7nPl-RA'}, {'votes': {'funny':
1, 'useful': 1, 'cool': 1}, 'user_id': 'l_szjd-ken3ma6oHDkTYXg', 'review_id': 'CFiLh7WvH
7dM3qVZvNiacQ', 'stars': 2, 'date': '2013-12-23', 'text': "After waiting for almost 30 m
inutes to trade in an old phone part of the buy back program, our customer service rep i
ncorrectly processed the transaction. This led to us waiting another 30 minutes for him
to correct it. Don't visit this store if you want pleasant or good service.", 'type': 'r
eview', 'business_id': 'HZdLhv6COCleJMo7nPl-RA'}]
```

Finally, we will load two lists. The first list consists of male names, and the second list consists of female names, to map Yelp user names to gender.

In [ ]:
```
# helper function to load a list of names from a publicly accessible url
def get_names_from_path(path):
    file_reader = open(path, 'r')
    str_contents = file_reader.readlines()
    str_contents = [x.strip() for x in str_contents]
    result = str_contents[6:]
    return result

male_names = get_names_from_path('/ocean/projects/cie170025p/shared/data/male.txt')
print('First five male names: ', male_names[:5])
print('Number of male names: ', len(male_names))

female_names = get_names_from_path('/ocean/projects/cie170025p/shared/data/female.txt')
print('First five female names: ', female_names[:5])
print('Number of female names: ', len(female_names))
```

```
First five male names:  ['Aamir', 'Aaron', 'Abbey', 'Abbie', 'Abbot']
Number of male names:  2943
```

```
First five female names:  ['Abagael', 'Abagail', 'Abbe', 'Abbey', 'Abbi']
Number of female names:  5001
```

Part 1: Exploratory Data Analysis

Performing some exploratory analysis is a great step toward understanding the data before building any statistical machine learning models on it.

In [ ]:
```
print ('Number of businesses: ', businesses_rdd.count())
print ('Number of users: ', users_rdd.count())
print ('Number of checkins: ', checkins_rdd.count())
print ('Number of reviews: ', reviews_rdd.count())
```

```
Number of businesses:  61184
Number of users:  366715
Number of checkins:  45166
Number of reviews:  62608
```

Question1: Print the top 5 business categories by frequency and the number of times they appear in the businesses data.

In [ ]:
```
businesses = businesses_rdd.flatMap(lambda x:x['categories']).map(lambda x: (x,1))
business_count = businesses.reduceByKey(lambda x,y: x+y).map(lambda x: (x[1],x[0])).sor
print(business_count.take(5))
```

```
[(21892, 'Restaurants'), (8919, 'Shopping'), (7862, 'Food'), (4738, 'Beauty & Spas'), (4
340, 'Nightlife')]
```

Question2: Print the top 5 cities by frequency and the number of times they appear in the businesses data.

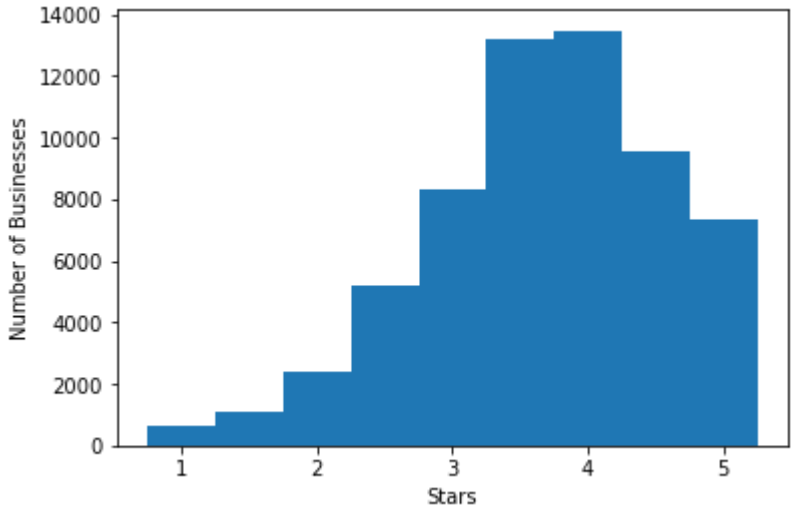
In [ ]:
```
cities = businesses_rdd.map(lambda x:x['city']).map(lambda x:(x,1))
city_count = cities.reduceByKey(lambda x,y:x+y).map(lambda x:(x[1],x[0])).sortByKey(Fal
print(city_count.take(5))
```

```
[(13601, 'Las Vegas'), (8410, 'Phoenix'), (4224, 'Charlotte'), (4039, 'Scottsdale'), (30
31, 'Edinburgh')]
```

Question3: Plot the histogram of stars received by businesses.

In [ ]:
```
businesses_stars_counts = businesses_rdd.map(lambda x:(x['stars'])).collect()
plt.hist(businesses_stars_counts, bins=[x/2-0.25 for x in range(2, 12)])
plt.xlabel('Stars')
plt.ylabel('Number of Businesses')
```

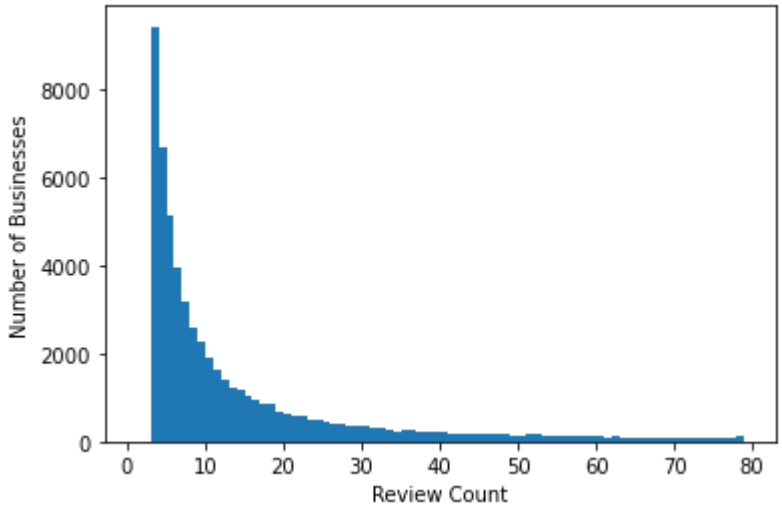
Out[ ]:
```
Text(0, 0.5, 'Number of Businesses')
```



Question4: Plot the histogram of number of reviews received by businesses.

In [ ]:
```
businesses_review_counts = businesses_rdd.map(lambda x:(x['review_count'])).collect()

plt.hist(businesses_review_counts, bins=range(1,80))
plt.xlabel('Review Count')
plt.ylabel('Number of Businesses')
```

Out[ ]:
```
Text(0, 0.5, 'Number of Businesses')
```

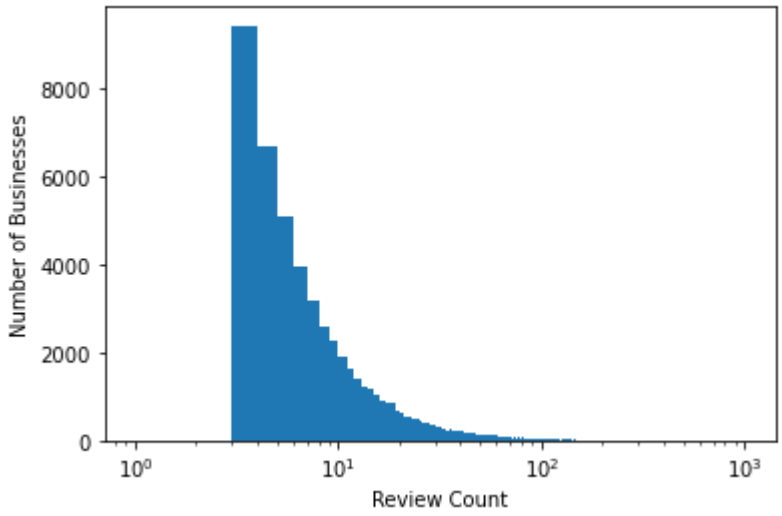


Question5: Plot the above histogram but now on a log-log scale using `bins=range(1,1000)` . Do you see a Power Law relationship in the plot? Explain your answer.

Answer: Yes, we observe a power law relationship in the plot below due to the long-tail. This indicates that reviews for bigger businesses are less in number while for smaller businesses they're higher in number since there are more number of smaller businesses than the big, established ones.

In [ ]:
```
plt.hist(businesses_review_counts, bins=range(1,1000))
plt.xscale("log")
plt.xlabel('Review Count')
plt.ylabel('Number of Businesses')
```

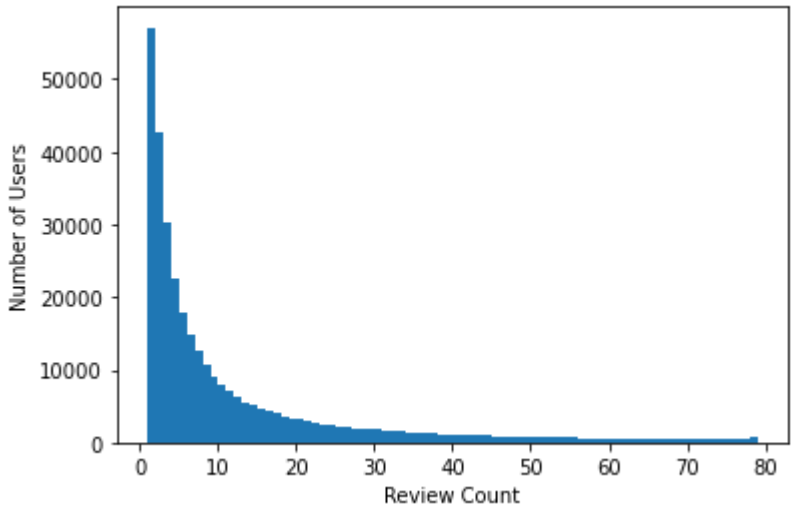
Out[ ]: Text(0, 0.5, 'Number of Businesses')



Question6: Plot the histogram of number of reviews written by users.

In [ ]:
```
users_review_counts = users_rdd.map(lambda x:x['review_count']).collect()

plt.hist(users_review_counts, bins=range(1,80))
plt.xlabel('Review Count')
plt.ylabel('Number of Users')
```

Out[ ]: Text(0, 0.5, 'Number of Users')

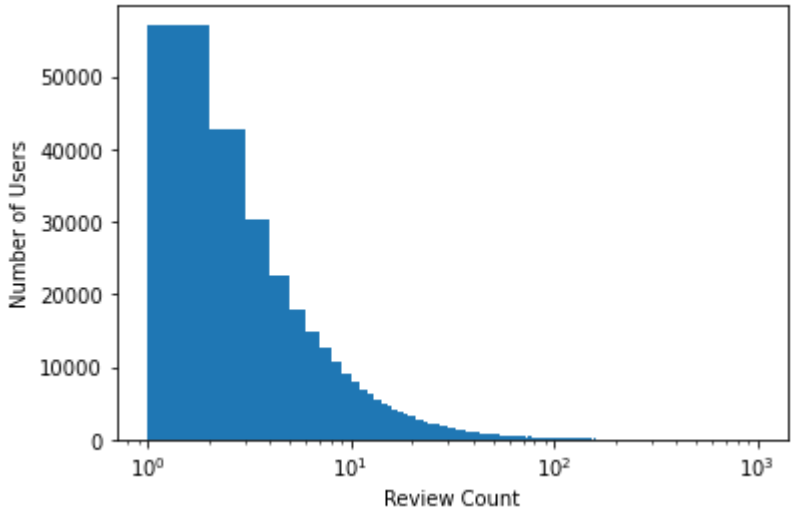


Question7: Plot the above histogram but now on a log-log scale using `bins=range(1,1000)` . Do you see a Power Law relationship in the plot? Explain your answer.

Answer: Yes, the plot below follows a power law relationship as we can observe a long-tail. We can infer that users who will give more reviews will be lesser than users who will give lesser reviews for the businesses.

In [ ]:
```
plt.hist(users_review_counts, bins=range(1,1000))
plt.xscale("log")
plt.xlabel('Review Count')
plt.ylabel('Number of Users')
```

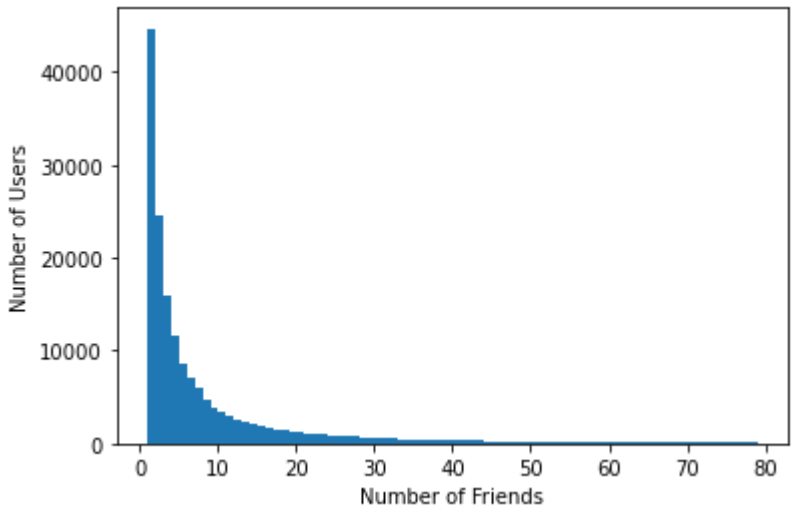
Out[ ]:
```
Text(0, 0.5, 'Number of Users')
```



Question8: Plot the histogram of number of friends a Yelp user has.

In [ ]:
```
user_friend_counts = users_rdd.map(lambda x: len(x['friends'])).collect()

plt.hist(user_friend_counts, bins=range(1,80))
plt.xlabel('Number of Friends')
plt.ylabel('Number of Users')
```

Out[ ]:
```
Text(0, 0.5, 'Number of Users')
```

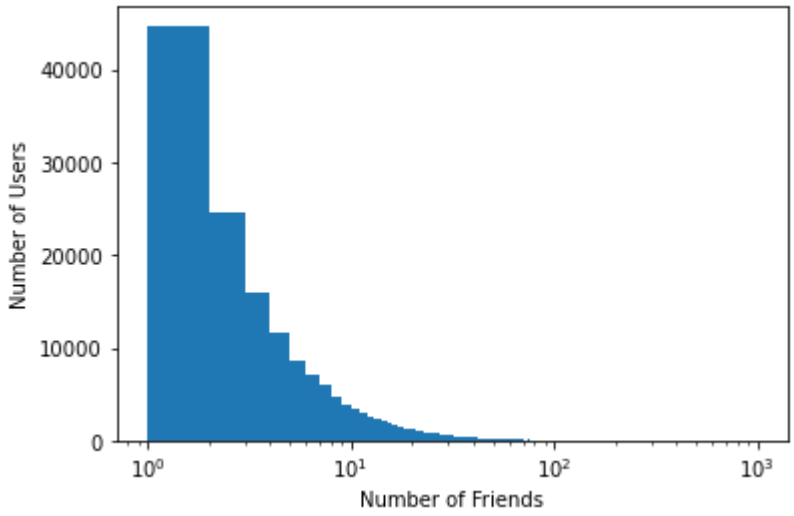


Question9: Plot the above histogram but now on a log-log scale. Do you see a Power Law relationship in the plot? Explain your answer.

Answer: Yes, we observe a Power Law in the histogram below but it's tail is small compared to other long-tailed graphs we observed. We can infer that very small number of users have higher number of friends while most of them have lesser friends on Yelp.

In [ ]:
```
plt.hist(user_friend_counts, bins=range(1,1000))
plt.xscale("log")
plt.xlabel('Number of Friends')
plt.ylabel('Number of Users')
```

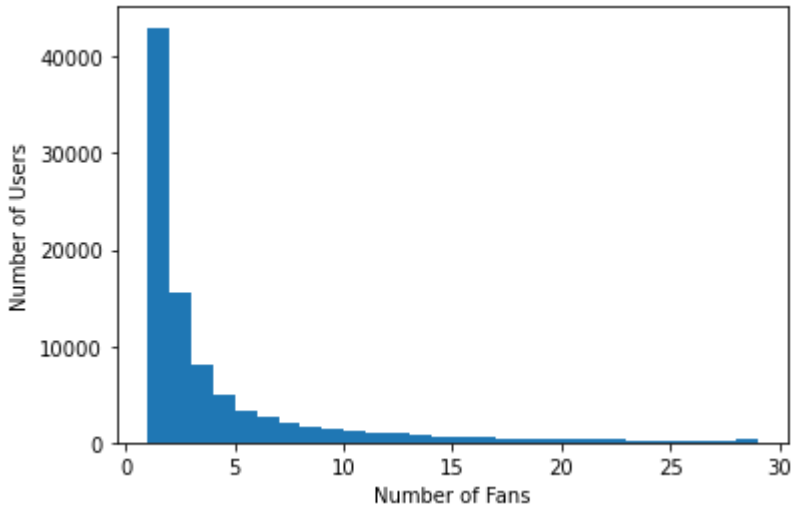
Out[ ]: Text(0, 0.5, 'Number of Users')



Question10: Plot the histogram of number of fans a Yelp user has.

In [ ]:
```
users_fan_counts = users_rdd.map(lambda x: x['fans']).collect()

plt.hist(users_fan_counts, bins=range(1,30))
plt.xlabel('Number of Fans')
plt.ylabel('Number of Users')
```

Out[ ]: Text(0, 0.5, 'Number of Users')



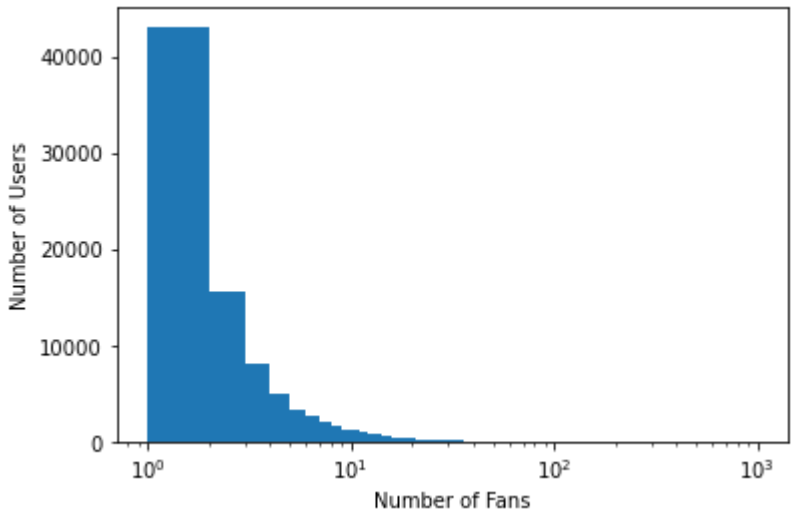
Question11: Plot the above histogram but now on a log-log scale. Do you see a Power Law relationship in the plot? Explain your answer.

Answer: Yes, we observe a Power Law relationship but not that strong as it's tail isn't that long compared to an ideal Power Law graph. We can infer that very less number of users have extremely high fan count like famous celebrities while most of the users have lesser fans.

In [ ]:
```
plt.hist(users_fan_counts, bins=range(1,1000))
plt.xscale("log")
```

```
plt.xlabel('Number of Fans')
```

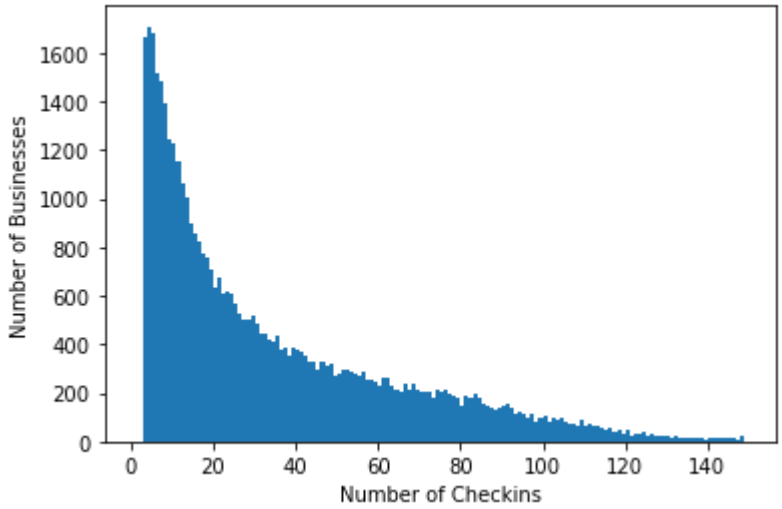
Out[ ]: Text(0, 0.5, 'Number of Users')



Question12: Plot the histogram of number of checkins per Yelp business.

In [ ]:
```
business_checkin_counts = checkins_rdd.map(lambda x:len(x['checkin_info'])).collect()

plt.hist(business_checkin_counts, bins=range(1,150))
plt.xlabel('Number of Checkins')
plt.ylabel('Number of Businesses')
```

Out[ ]: Text(0, 0.5, 'Number of Businesses')

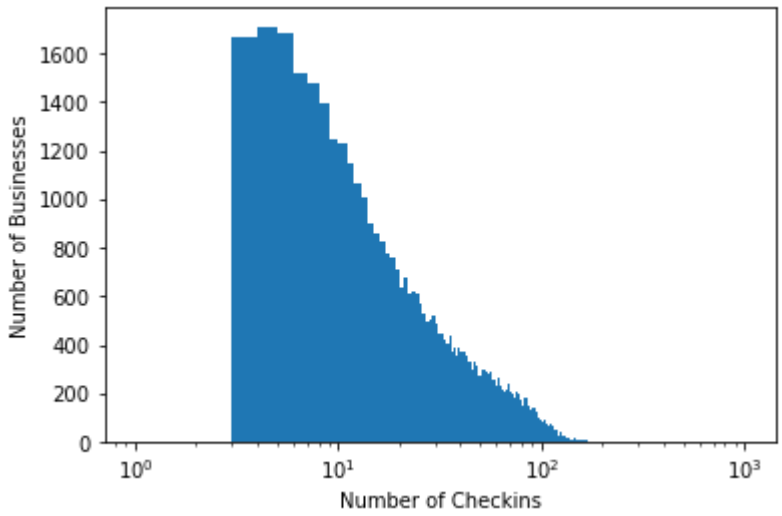


Question13: Plot the above histogram but now on a log-log scale using `bins=range(3,200)` . Do you see a Power Law relationship in the plot? Explain your answer.

Answer: No, we do not observe a Power Law relationship in the following graph as there is no long-tail towards the end in this graph. An ideal Power Law graph has 95% of the data clustered in the start of the graph followed by a long tail indicating the number of Yelp businesses and checkins do not follow a Power Law distribution. The number of checkins are in the range of 0 to 200 only without any long tail extension.

In [ ]:
```
plt.hist(business_checkin_counts, bins=range(1,1000))
plt.xscale("log")
plt.xlabel('Number of Checkins')
plt.ylabel('Number of Businesses')
```

Out[ ]:
```
Text(0, 0.5, 'Number of Businesses')
```



Question14: Find the maximum value of checkins per business. Filter to obtain business IDs of businesses that had these maximum number of checkins. Fill in the code required to carry out these steps.

In [ ]:
```
max_checkin_count = checkins_rdd.map(lambda x: len(x['checkin_info'])).max()
business_ids_with_max_checkins = checkins_rdd \
    .filter(lambda x: len(x['checkin_info'])==max_checkin_count) \
    .map(lambda x: x['business_id']).collect()
len(business_ids_with_max_checkins)
```

Out[ ]:
```
41
```

In [ ]:
```
business_names_with_max_checkins = businesses_rdd \
    .filter(lambda x: x['business_id'] in business_ids_with_max_checkins) \
    .map(lambda x: (x['name'], x['city'])).collect()
business_names_with_max_checkins
```

Out[ ]:
```
[('Charlotte Douglas International Airport', 'Charlotte'),
 ('Phoenix Sky Harbor International Airport', 'Phoenix'),
 ('Pho Kim Long', 'Las Vegas'),
 ('McCarran International Airport', 'Las Vegas'),
 ('The California Hotel & Casino', 'Las Vegas'),
 ('Golden Nugget Hotel & Casino', 'Las Vegas'),
 ('Rio All Suites Hotel & Casino', 'Las Vegas'),
 ('24 Hour Fitness', 'Las Vegas'),
 ('Excalibur Hotel', 'Las Vegas'),
 ('Ellis Island Casino & Brewery', 'Las Vegas'),
 ('Las Vegas Athletic Club', 'Las Vegas'),
 ('Orleans Hotel & Casino', 'Las Vegas'),
 ('Palms Casino Resort', 'Las Vegas'),
 ('Bellagio Hotel', 'Las Vegas'),
 ('New York - New York', 'Las Vegas'),
```

```
('Paris Las Vegas Hotel & Casino', 'Las Vegas'),
('Monte Carlo Hotel And Casino', 'Las Vegas'),
('The Venetian Resort Hotel Casino', 'Las Vegas'),
('The Mirage', 'Las Vegas'),
('MGM Grand Hotel', 'Las Vegas'),
('Treasure Island, LLC', 'Las Vegas'),
('The Peppermill Restaurant & Fireside Lounge', 'Las Vegas'),
('Flamingo Las Vegas Hotel & Casino', 'Las Vegas'),
('Luxor Hotel And Casino Las Vegas', 'Las Vegas'),
('Mandalay Bay Resort & Casino', 'Las Vegas'),
('Grand Lux Cafe', 'Las Vegas'),
('Market Street Cafe', 'Las Vegas'),
('Caesars Palace Las Vegas Hotel & Casino', 'Las Vegas'),
('Las Vegas Athletic Club', 'Las Vegas'),
('South Point Hotel, Casino and Spa', 'Las Vegas'),
('Starbucks', 'Las Vegas'),
('Planet Hollywood Las Vegas Resort & Casino', 'Las Vegas'),
('Earl of Sandwich', 'Las Vegas'),
('InterContinental Alliance Resorts THE PALAZZO', 'Las Vegas'),
('Las Vegas Athletic Club', 'Las Vegas'),
('Encore', 'Las Vegas'),
('ARIA Hotel & Casino', 'Las Vegas'),
('Vdara Hotel', 'Las Vegas'),
('The Cosmopolitan of Las Vegas', 'Las Vegas'),
("Bally's Las Vegas Hotel & Casino", 'Las Vegas'),
('Hard Rock Hotel & Casino', 'Las Vegas')]
```

Question15: Why do you think the above list sees much higher checkins than other businesses in the dataset?

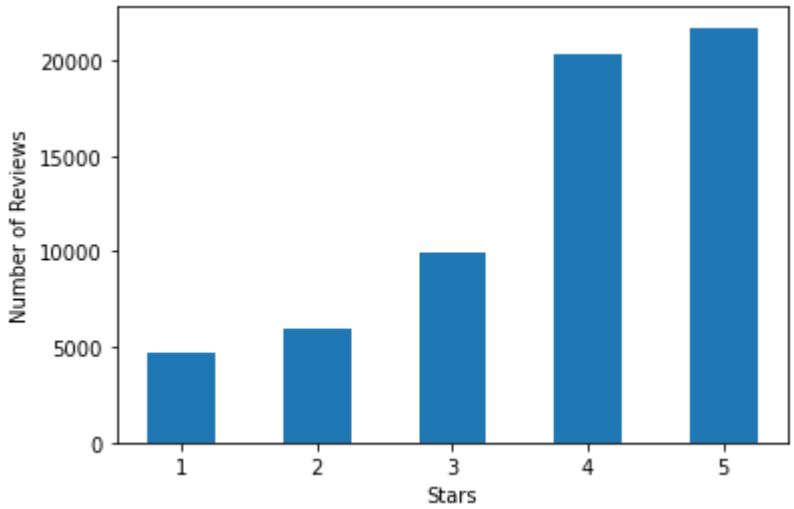
Answer: The above list sees much higher checkins than other businesses could be due to their location. Most of them are located in Las Vegas which is one of the popular tourist spot. They could also be offering some good food with deals or have other attractive activities like casinos, shows etc in them.

Question16: Plot a histogram of the stars associated with business reviews.

In [ ]:

```
review_stars_counts = reviews_rdd.map(lambda x: x['stars']).collect()
plt.hist(review_stars_counts, bins=[x/2-0.25 for x in range(2, 12)])
plt.xlabel('Stars')
plt.ylabel('Number of Reviews')
```

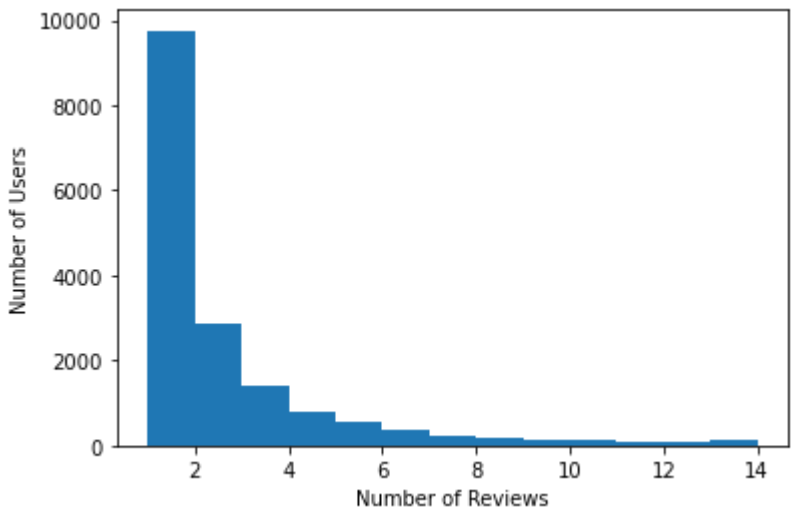
Out[ ]: Text(0, 0.5, 'Number of Reviews')



Question17: Plot a histogram of the number of reviews written per Yelp user.

In [ ]:
```
user_review_counts = list(reviews_rdd.map(lambda x: x['user_id']).countByValue().values

plt.hist(user_review_counts, bins=[x for x in range(1, 15)])
plt.xlabel('Number of Reviews')
plt.ylabel('Number of Users')
```

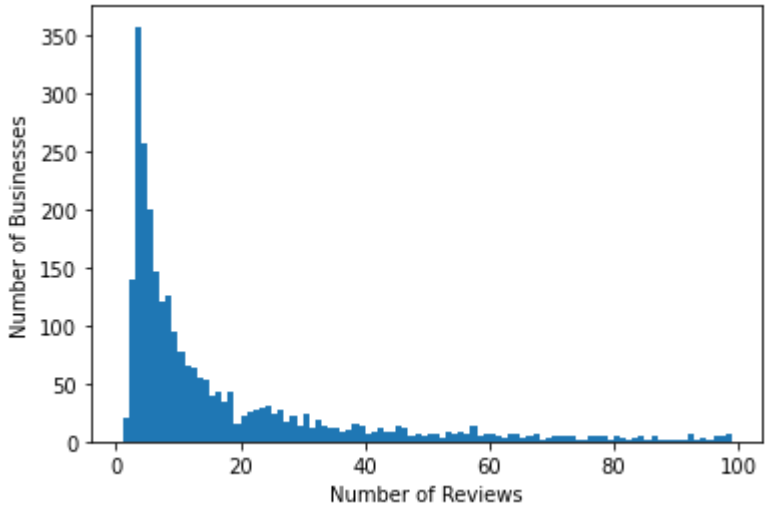
Out[ ]: Text(0, 0.5, 'Number of Users')



Question18: Plot a histogram of the number of reviews written per Yelp business.

In [ ]:
```
business_review_counts = list(reviews_rdd.map(lambda x: x['business_id']).countByValue(

plt.hist(business_review_counts, bins=[x for x in range(1, 100)])
plt.xlabel('Number of Reviews')
plt.ylabel('Number of Businesses')
```

Out[ ]: Text(0, 0.5, 'Number of Businesses')



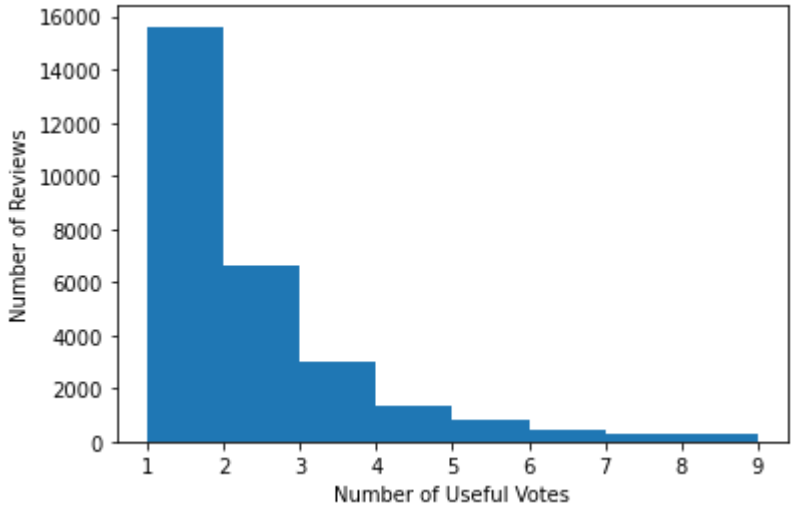
In [ ]:
```
reviews_rdd.take(1)
```

Out[ ]:
```
[{'votes': {'funny': 3, 'useful': 7, 'cool': 7},
  'user_id': 'JbAeIYc89Sk8SWmrBCJs9g',
  'review_id': 'fBQ69-NU9ZyTjjS7Tb5tww',
  'stars': 5,
  'date': '2013-06-10',
  'text': "THANK YOU ROB! i truly appreciated all the help i received from this agent to
day who was able to removed the extra charges on my bill that the Pasadena Verizon Store
on Lake was charging me on my bill for upgrading my phone.   When i went in i was having
problems with my Blacberry and had to switch to the Iphone last week. Rob from the Penns
ylvania store who i was connected today was able to look at my bill and all the notes an
d correct the problem immediately. Great Customer Service! He even set up a FOLLOW UP Ph
one Call with me On July 5th to make sure the credit goes through on my bill...I can't t
hank him enough!!!!",
  'type': 'review',
  'business_id': 'HZdLhv6COCleJMo7nPl-RA'}]
```

Question19: Plot a histogram of the number of useful votes received by Yelp reviews.

In [ ]:
```
review_useful_counts = reviews_rdd.map(lambda x:x['votes']).map(lambda x:x['useful']).c

plt.hist(review_useful_counts, bins=[x for x in range(1, 10)])
plt.xlabel('Number of Useful Votes')
plt.ylabel('Number of Reviews')
```

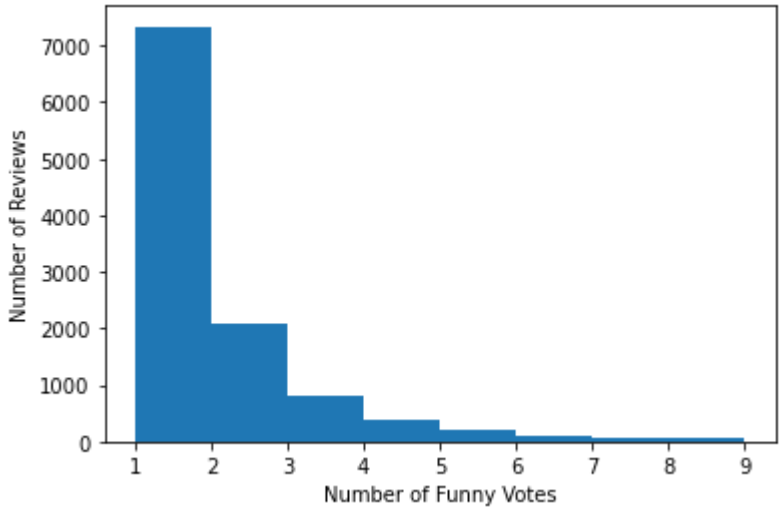
Out[ ]:
```
Text(0, 0.5, 'Number of Reviews')
```



Question20: Plot a histogram of the number of funny votes received by Yelp reviews.

In [ ]:
```
review_funny_counts = reviews_rdd.map(lambda x:x['votes']).map(lambda x:x['funny']).col

plt.hist(review_funny_counts, bins=[x for x in range(1, 10)])
plt.xlabel('Number of Funny Votes')
plt.ylabel('Number of Reviews')
```

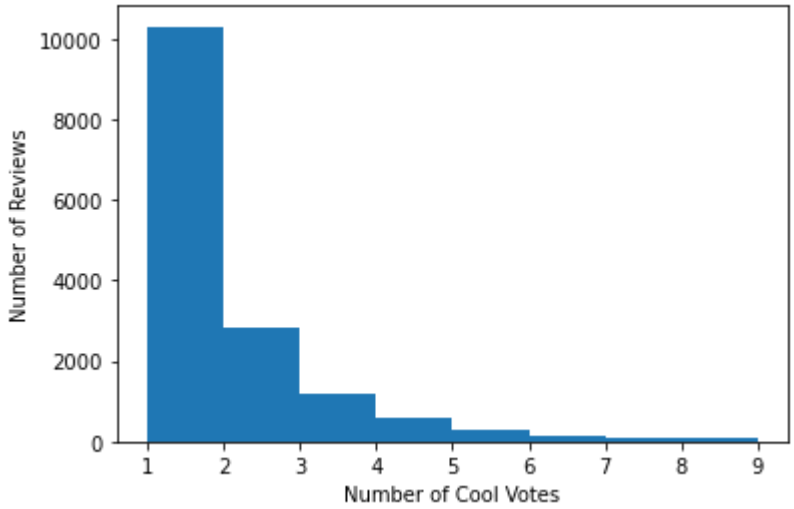
Out[ ]:
```
Text(0, 0.5, 'Number of Reviews')
```



Question21: Plot a histogram of the number of cool votes received by Yelp reviews.

In [ ]:
```
review_cool_counts = reviews_rdd.map(lambda x:x['votes']).map(lambda x:x['cool']).colle

plt.hist(review_cool_counts, bins=[x for x in range(1, 10)])
plt.xlabel('Number of Cool Votes')
plt.ylabel('Number of Reviews')
```

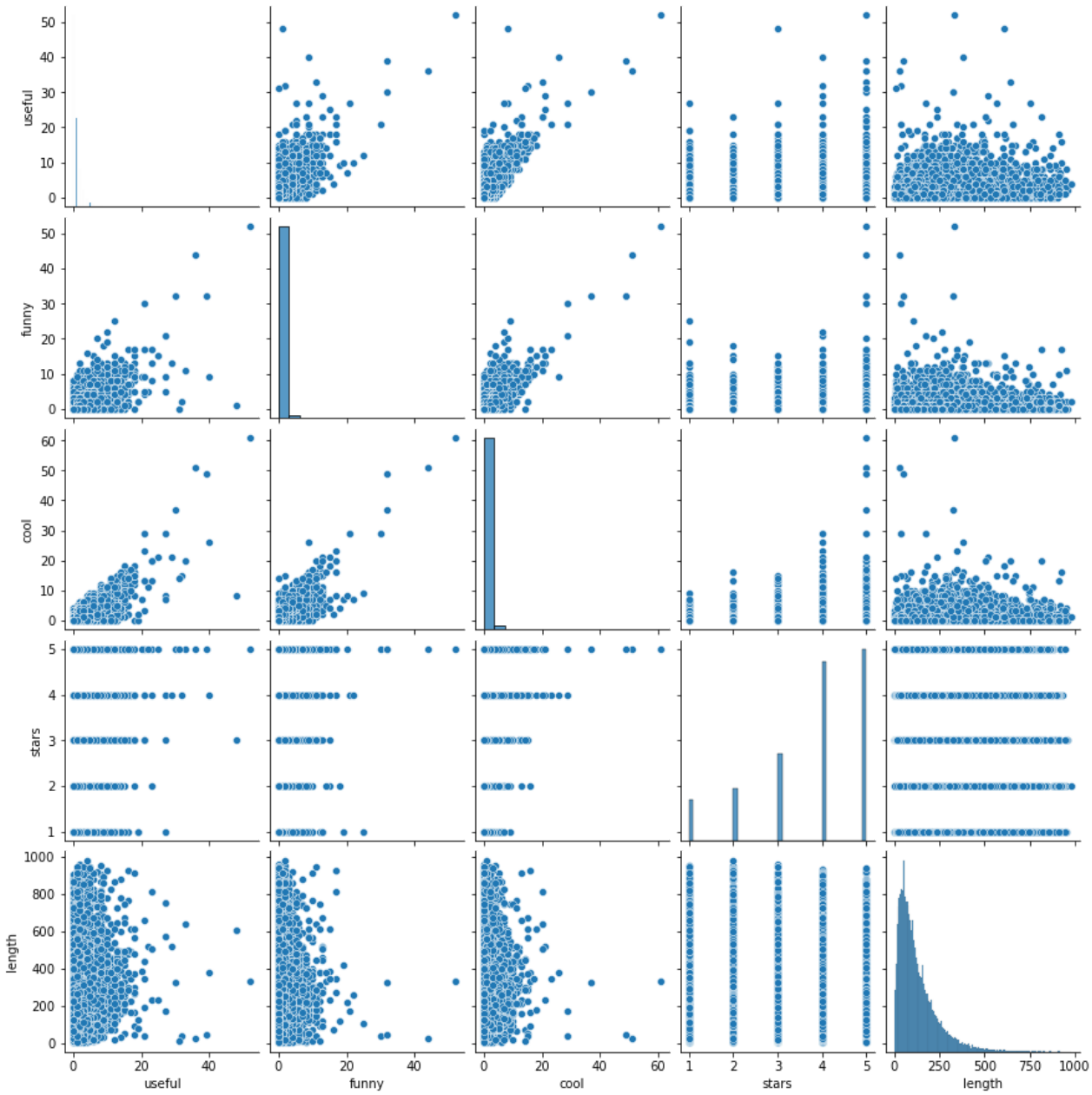
Out[ ]:
```
Text(0, 0.5, 'Number of Reviews')
```



Question22: Plot a pair-plot of the number of useful, funny, and cool votes received by Yelp reviews alongwith the stars associated with the review and the length of the review.

In [ ]:
```
review_votes_length = reviews_rdd.map(lambda x: (x['votes']['useful'],x['votes']['funny
review_votes_length_df = pd.DataFrame(review_votes_length, columns=['useful', 'funny',
sns.pairplot(review_votes_length_df)
```

Out[ ]:
```
<seaborn.axisgrid.PairGrid at 0x14ead16e3250>
```



Question23: Let us plot the distribution of the number of words used by males and females in their reviews. We will use the lists "male_names" and "female_names" we had created earlier for this purpose. Let's first find the user IDs associated with males and females.

In [ ]:

```
male_users = users_rdd.filter(lambda x: x['name'] in male_names)
female_users = users_rdd.filter(lambda x: x['name'] in female_names)

male_user_ids = male_users.map(lambda x: x['user_id']).collect()
female_user_ids = female_users.map(lambda x:x['user_id']).collect()

print (len(male_user_ids))
print (len(female_user_ids))
print (users_rdd.count())
```

```
166682
174869
366715
```

Question24: We can now use the user ID lists to separate the reviews into those by males and females and calculate the length of each review.

In [ ]:
```
male_reviews = reviews_rdd.filter(lambda x: x['user_id'] in male_user_ids).map(lambda x
female_reviews = reviews_rdd.filter(lambda x: x['user_id'] in female_user_ids).map(lamb

male_word_count = male_reviews.map(lambda x:len(x.split()))
female_word_count = female_reviews.map(lambda x: len(x.split()))

print ('Male and female review length averages: ', male_word_count.mean(), female_word_
```

```
Male and female review length averages:  127.3470787972136 139.21062276751053
```

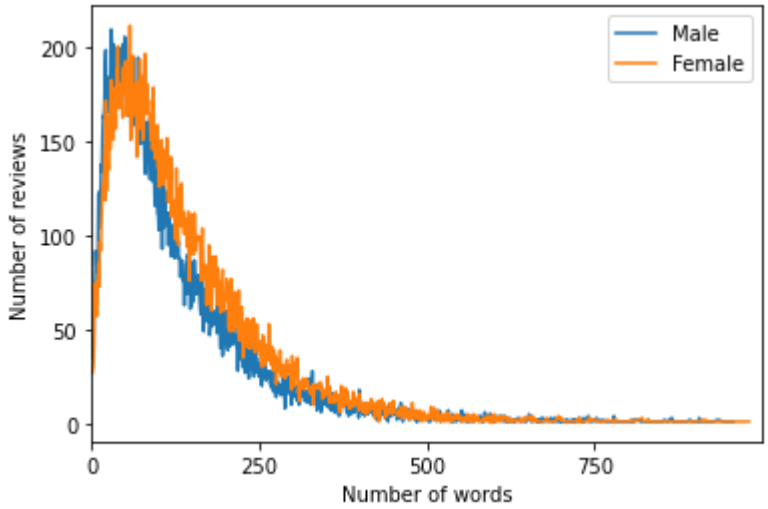
Question25: The code below calculates the distributions of review lengths for males and female reviewers and plots them. Do you see a marked difference between the average review length of male and female reviewers? Are there any major trends or differences between the distributions of review length of male and female reviewers?

Answer: The number of reviewers for review word count upto 375 has more contribution from female reviewers as observed from the graph. For initial reviews (90 average word length), male reviewers tend to write higher number of reviews(as seen from the blue lines above orange). Though after 90 word length unitl 375, female reviewers tend to write more. After 375 word length, the word count of the reviews are almost similar. Additionally, the curve tends to follow Power Law as it has a long tail after a peak in the start indicating there are lesser reviewers who write long reviews and more reviewers who write short reviews.

In [ ]:
```
male_word_distribution = list(male_word_count.map(lambda x : (x,1)).countByKey().items(
female_word_distribution = list(female_word_count.map(lambda x : (x,1)).countByKey().it

male_word_distribution = sorted(male_word_distribution, key=lambda x: x[0])
female_word_distribution = sorted(female_word_distribution, key=lambda x: x[0])
```

In [ ]:
```
fig, ax = plt.subplots()
ax.plot([x[0] for x in male_word_distribution], [x[1] for x in male_word_distribution],
ax.plot([x[0] for x in female_word_distribution], [x[1] for x in female_word_distributi

ax.set_xlim((0, 1000))
ax.set_xticks([0, 250, 500, 750])
ax.set_xticklabels(['0', '250','500','750'])

plt.xlabel('Number of words')
plt.ylabel('Number of reviews')
plt.legend()
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x1553c7b6d670>
```



Part 2: Classification using tree ensemble methods

In this section, we will predict the number of funny votes that a review has earned, indicating how funny readers found the review.

In [ ]:
```
from pyspark.mllib.tree import DecisionTree, DecisionTreeModel
from pyspark.mllib.tree import RandomForest, RandomForestModel
from pyspark.mllib.tree import GradientBoostedTrees, GradientBoostedTreesModel
from pyspark.mllib.linalg import Vectors, DenseVector, SparseVector
from pyspark.mllib.regression import LabeledPoint
```

Question1: Fill in the necessary code to calculate word counts from text reviews below.

In [ ]:
```
max_words = 30000

all_reviews = reviews_rdd.map(lambda x : (x['text'], x['votes']['funny']))
word_counts = list(all_reviews.flatMap(lambda x:x[0].split()).map(lambda x: (x,1)).coun
word_counts = sorted(word_counts, key=lambda x: -x[1])

unique_words = [x[0] for x in word_counts[:max_words]]
num_unique_words = len(unique_words)
print('Number of unique words: ', num_unique_words)
```

```
Number of unique words:  30000
```

Question2: We will now construct two dictionaries - one which maps from each word to a unique integer index and the second one which maps back from the index to the word. Write the code required to do this.

In [ ]:
```
word_to_index_dict = {unique_words[i]:i for i in range(len(unique_words))}
index_to_word_dict = {i:unique_words[i] for i in range(len(unique_words))}
```

Question3: Fill in the required code below to obtain a LabeledPoint RDD that can be used to train an mllib classifier/regressor.

In [ ]:
```
from collections import Counter
doc_vectors = all_reviews.map(lambda x: (x[1], x[0].split())).map(lambda x: (x[0], [wor
doc_vectors = doc_vectors.map(lambda x: LabeledPoint(x[0],SparseVector(num_unique_words
print(doc_vectors.count())
print(doc_vectors.take(2))
```

```
62608
[LabeledPoint(3.0, (30000,[0,1,2,4,6,8,9,12,13,15,16,19,20,25,38,42,49,58,86,107,114,12
5,164,168,189,204,208,233,297,301,344,352,364,365,465,471,490,669,684,767,781,942,1003,1
014,1106,1148,1360,1912,2283,2414,2960,3413,3958,4179,4356,4423,4779,4806,4968,5327,574
3,6021,6022,6057,6268,9594,14463,14848,15307,16821,16822,17427,26734],[8.0,3.0,1.0,5.0,
5.0,1.0,1.0,1.0,2.0,4.0,6.0,1.0,1.0,1.0,2.0,2.0,2.0,1.0,1.0,1.0,1.0,2.0,1.0,1.0,5.0,1.0,
1.0,1.0,1.0,1.0,1.0,1.0,2.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,3.0,2.0,1.0,1.0,1.0,
1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,
1.0,1.0,1.0])), LabeledPoint(1.0, (30000,[0,4,5,8,9,17,19,35,36,41,45,55,71,77,87,99,10
1,132,201,239,306,316,326,331,364,367,377,446,471,607,610,649,914,939,994,2960,3542,658
7,6709,10548,15761,18035,20358],[2.0,3.0,1.0,1.0,2.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.
0,1.0,1.0,1.0,1.0,1.0,2.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,2.0,1.0,1.0,1.0,1.0,2.0,1.0,1.0,1.
0,1.0,1.0,1.0,1.0,1.0,1.0,1.0]))]
```

Question4: Randomly split the doc_vectors RDD into 80% training and 20% validation data.

In [ ]:
```
doc_vectors_train, doc_vectors_val = doc_vectors.randomSplit(weights=[0.8,0.2])
```

Question5: Let us implement the baseline predictor which always outputs the most common value of funny votes. Fill in appropriate code.

In [ ]:
```
most_common_prediction = doc_vectors_train.map(lambda x: (x.label,1)).reduceByKey(lambd
                         .sortByKey(False).map(lambda x:x[1]).take(1)[0]

labels_and_predictions = doc_vectors_val.map(lambda lp: (lp.label, most_common_predicti
val_mse = labels_and_predictions.map(lambda lp: (lp[0] - lp[1]) * (lp[0] - lp[1])).sum(
    float(doc_vectors_val.count())
print('Validation Root Mean Squared Error (Baseline) = ' + str(val_mse))
print('Learned baseline prediction: ', most_common_prediction)
```

```
Validation Root Mean Squared Error (Baseline) = 1.2262937962070073
Learned baseline prediction:  0.0
```

Question6: Let us now use a Decision Tree to predict the number of funny votes. Set the maximum depth of the tree to 5 and use an appropriate impurity metric for regression.

In [ ]:
```
dt_model = DecisionTree.trainRegressor(doc_vectors_train,maxDepth=5,categoricalFeatures

predictions = dt_model.predict(doc_vectors_val.map(lambda x: x.features))
labels_and_predictions = doc_vectors_val.map(lambda lp: lp.label).zip(predictions)
val_mse = labels_and_predictions.map(lambda lp: (lp[0] - lp[1]) * (lp[0] - lp[1])).sum(
    float(doc_vectors_val.count())
print('Validation Root Mean Squared Error (Decision Tree) = ' + str(val_mse))
print('Learned regression tree model:')
print(dt_model.toDebugString())
```

```
Validation Root Mean Squared Error (Decision Tree) = 0.7459508019068113
Learned regression tree model:
DecisionTreeModel regressor of depth 5 with 35 nodes
  If (feature 24056 <= 0.5)
   If (feature 4 <= 4.5)
```

```
  If (feature 2138 <= 1.5)
   If (feature 5 <= 2.5)
    If (feature 2570 <= 1.5)
     Predict: 0.1921811619662078
    Else (feature 2570 > 1.5)
     Predict: 12.5
   Else (feature 5 > 2.5)
    If (feature 33 <= 6.5)
     Predict: 0.38892377244757004
    Else (feature 33 > 6.5)
     Predict: 9.333333333333334
  Else (feature 2138 > 1.5)
   If (feature 44 <= 0.5)
    If (feature 3 <= 1.5)
     Predict: 0.0
    Else (feature 3 > 1.5)
     Predict: 3.0
   Else (feature 44 > 0.5)
    Predict: 44.0
 Else (feature 4 > 4.5)
  If (feature 7214 <= 1.5)
   If (feature 29916 <= 0.5)
    If (feature 146 <= 0.5)
     Predict: 0.5192457016084304
    Else (feature 146 > 0.5)
     Predict: 0.953168044077135
   Else (feature 29916 > 0.5)
    If (feature 0 <= 15.5)
     Predict: 12.0
    Else (feature 0 > 15.5)
     Predict: 13.0
  Else (feature 7214 > 1.5)
   Predict: 32.0
Else (feature 24056 > 0.5)
 If (feature 2677 <= 0.5)
  If (feature 36 <= 0.5)
   If (feature 17 <= 3.5)
    If (feature 0 <= 22.5)
     Predict: 8.0
    Else (feature 0 > 22.5)
     Predict: 6.0
   Else (feature 17 > 3.5)
    Predict: 2.0
  Else (feature 36 > 0.5)
   Predict: 0.0
 Else (feature 2677 > 0.5)
  If (feature 0 <= 2.5)
   Predict: 32.0
  Else (feature 0 > 2.5)
   Predict: 52.0
```

Question7: Let us now use a Random Forest ensemble to predict the number of funny votes. Set the maximum depth of the tree to 5 and use an appropriate impurity metric for regression. Build a random forest regressor with 10 trees.

In [ ]:

```
rf_model = RandomForest.trainRegressor(doc_vectors_train,numTrees = 10, maxDepth=5,cate

predictions = rf_model.predict(doc_vectors_val.map(lambda x: x.features))
labels_and_predictions = doc_vectors_val.map(lambda lp: lp.label).zip(predictions)
val_mse = labels_and_predictions.map(lambda lp: (lp[0] - lp[1]) * (lp[0] - lp[1])).sum(
    float(doc_vectors_val.count())
print('Validation Root Mean Squared Error (Random Forest) = ' + str(val_mse))
```

```
print('Learned regression RF model:')
print(rf_model.toDebugString())
```

```
Validation Root Mean Squared Error (Random Forest) = 0.7403807049182749
Learned regression RF model:
TreeEnsembleModel regressor with 10 trees

  Tree 0:
    If (feature 11343 <= 1.5)
     If (feature 2 <= 5.5)
      If (feature 940 <= 1.5)
       If (feature 9675 <= 1.5)
        If (feature 24544 <= 0.5)
         Predict: 0.23074558819762825
        Else (feature 24544 > 0.5)
         Predict: 7.25
       Else (feature 9675 > 1.5)
        Predict: 15.0
      Else (feature 940 > 1.5)
       If (feature 0 <= 4.5)
        Predict: 0.0
       Else (feature 0 > 4.5)
        Predict: 20.0
     Else (feature 2 > 5.5)
      If (feature 1791 <= 3.5)
       If (feature 23252 <= 0.5)
        If (feature 23778 <= 0.5)
         Predict: 0.5915582067968185
        Else (feature 23778 > 0.5)
         Predict: 13.0
       Else (feature 23252 > 0.5)
        Predict: 14.0
      Else (feature 1791 > 3.5)
       Predict: 32.0
    Else (feature 11343 > 1.5)
     Predict: 52.0
  Tree 1:
    If (feature 4 <= 4.5)
     If (feature 24056 <= 0.5)
      If (feature 1 <= 21.5)
       If (feature 5321 <= 1.5)
        If (feature 29282 <= 0.5)
         Predict: 0.2324881060836117
        Else (feature 29282 > 0.5)
         Predict: 8.0
       Else (feature 5321 > 1.5)
        If (feature 2 <= 2.5)
         Predict: 0.0
        Else (feature 2 > 2.5)
         Predict: 17.0
      Else (feature 1 > 21.5)
       If (feature 0 <= 19.5)
        Predict: 1.0
       Else (feature 0 > 19.5)
        If (feature 0 <= 29.5)
         Predict: 13.0
        Else (feature 0 > 29.5)
         Predict: 15.0
     Else (feature 24056 > 0.5)
      If (feature 0 <= 19.5)
       If (feature 2 <= 1.5)
        Predict: 32.0
       Else (feature 2 > 1.5)
        Predict: 52.0
      Else (feature 0 > 19.5)
```

```
      If (feature 17 <= 3.5)
       Predict: 8.0
      Else (feature 17 > 3.5)
       Predict: 2.0
    Else (feature 4 > 4.5)
     If (feature 15614 <= 0.5)
      If (feature 1791 <= 3.5)
       If (feature 1513 <= 1.5)
        If (feature 5431 <= 4.5)
         Predict: 0.5760335093789838
        Else (feature 5431 > 4.5)
         Predict: 11.0
       Else (feature 1513 > 1.5)
        If (feature 1 <= 5.5)
         Predict: 20.0
        Else (feature 1 > 5.5)
         Predict: 5.25
      Else (feature 1791 > 3.5)
       Predict: 32.0
     Else (feature 15614 > 0.5)
      If (feature 133 <= 0.5)
       If (feature 101 <= 0.5)
        If (feature 1 <= 7.5)
         Predict: 3.0
        Else (feature 1 > 7.5)
         Predict: 2.0
       Else (feature 101 > 0.5)
        If (feature 0 <= 13.5)
         Predict: 0.0
        Else (feature 0 > 13.5)
         Predict: 1.0
      Else (feature 133 > 0.5)
       If (feature 18 <= 0.5)
        If (feature 0 <= 6.5)
         Predict: 11.0
        Else (feature 0 > 6.5)
         Predict: 13.0
       Else (feature 18 > 0.5)
        Predict: 9.0
  Tree 2:
    If (feature 11343 <= 1.5)
     If (feature 2 <= 5.5)
      If (feature 22762 <= 0.5)
       If (feature 3 <= 4.5)
        If (feature 13900 <= 0.5)
         Predict: 0.19331072749691738
        Else (feature 13900 > 0.5)
         Predict: 9.0
       Else (feature 3 > 4.5)
        If (feature 17677 <= 0.5)
         Predict: 0.4022878932316492
        Else (feature 17677 > 0.5)
         Predict: 18.0
      Else (feature 22762 > 0.5)
       If (feature 2 <= 0.5)
        Predict: 30.0
       Else (feature 2 > 0.5)
        If (feature 1 <= 2.5)
         Predict: 1.0
        Else (feature 1 > 2.5)
         Predict: 0.0
     Else (feature 2 > 5.5)
      If (feature 24762 <= 0.5)
       If (feature 14439 <= 0.5)
        If (feature 29916 <= 0.5)
```

```
          Predict: 0.564151284835324
         Else (feature 29916 > 0.5)
          Predict: 12.666666666666666
        Else (feature 14439 > 0.5)
         If (feature 108 <= 0.5)
          Predict: 0.75
         Else (feature 108 > 0.5)
          Predict: 11.0
       Else (feature 24762 > 0.5)
        If (feature 3 <= 6.5)
         Predict: 1.0
        Else (feature 3 > 6.5)
         Predict: 17.0
    Else (feature 11343 > 1.5)
     If (feature 6 <= 1.5)
      Predict: 4.0
     Else (feature 6 > 1.5)
      Predict: 52.0
  Tree 3:
    If (feature 418 <= 3.5)
     If (feature 21655 <= 1.0)
      If (feature 2 <= 5.5)
       If (feature 3 <= 5.5)
        If (feature 16439 <= 0.5)
         Predict: 0.19995396743195812
        Else (feature 16439 > 0.5)
         Predict: 6.5
       Else (feature 3 > 5.5)
        If (feature 24593 <= 0.5)
         Predict: 0.4380313199105145
        Else (feature 24593 > 0.5)
         Predict: 8.75
      Else (feature 2 > 5.5)
       If (feature 28487 <= 0.5)
        If (feature 23952 <= 0.5)
         Predict: 0.5725586470267321
        Else (feature 23952 > 0.5)
         Predict: 11.666666666666666
       Else (feature 28487 > 0.5)
        If (feature 2 <= 7.5)
         Predict: 21.0
        Else (feature 2 > 7.5)
         Predict: 0.5
     Else (feature 21655 > 1.0)
      Predict: 32.0
    Else (feature 418 > 3.5)
     Predict: 32.0
  Tree 4:
    If (feature 5 <= 4.5)
     If (feature 22762 <= 0.5)
      If (feature 3 <= 6.5)
       If (feature 3686 <= 0.5)
        If (feature 5444 <= 1.5)
         Predict: 0.21358918721079187
        Else (feature 5444 > 1.5)
         Predict: 6.6
       Else (feature 3686 > 0.5)
        If (feature 7758 <= 0.5)
         Predict: 0.44285714285714284
        Else (feature 7758 > 0.5)
         Predict: 26.4
      Else (feature 3 > 6.5)
       If (feature 1513 <= 1.5)
        If (feature 20949 <= 0.5)
         Predict: 0.5362819626814098
```

```
      Else (feature 20949 > 0.5)
       Predict: 11.0
     Else (feature 1513 > 1.5)
      Predict: 20.0
   Else (feature 22762 > 0.5)
    If (feature 3 <= 0.5)
     Predict: 30.0
    Else (feature 3 > 0.5)
     If (feature 3 <= 2.5)
      Predict: 0.0
     Else (feature 3 > 2.5)
      If (feature 3 <= 4.5)
       Predict: 1.0
      Else (feature 3 > 4.5)
       Predict: 2.0
  Else (feature 5 > 4.5)
   If (feature 19697 <= 0.5)
    If (feature 4778 <= 1.5)
     If (feature 11156 <= 0.5)
      If (feature 24084 <= 0.5)
       Predict: 0.6754324286316974
      Else (feature 24084 > 0.5)
       Predict: 15.0
     Else (feature 11156 > 0.5)
      If (feature 49 <= 1.5)
       Predict: 3.5714285714285716
      Else (feature 49 > 1.5)
       Predict: 13.0
    Else (feature 4778 > 1.5)
     If (feature 3 <= 2.5)
      Predict: 21.0
     Else (feature 3 > 2.5)
      If (feature 3 <= 6.5)
       Predict: 2.0
      Else (feature 3 > 6.5)
       Predict: 0.0
   Else (feature 19697 > 0.5)
    If (feature 0 <= 11.5)
     Predict: 0.0
    Else (feature 0 > 11.5)
     Predict: 32.0
Tree 5:
  If (feature 4 <= 5.5)
   If (feature 5 <= 16.5)
    If (feature 7758 <= 0.5)
     If (feature 27218 <= 0.5)
      If (feature 3 <= 4.5)
       Predict: 0.20616598480420806
      Else (feature 3 > 4.5)
       Predict: 0.43745924090256944
     Else (feature 27218 > 0.5)
      If (feature 0 <= 2.5)
       Predict: 32.0
      Else (feature 0 > 2.5)
       Predict: 0.0
    Else (feature 7758 > 0.5)
     If (feature 3686 <= 0.5)
      If (feature 42 <= 0.5)
       Predict: 0.4782608695652174
      Else (feature 42 > 0.5)
       Predict: 2.857142857142857
     Else (feature 3686 > 0.5)
      If (feature 1 <= 0.5)
       Predict: 44.0
      Else (feature 1 > 0.5)
```

```
        Predict: 0.0
      Else (feature 5 > 16.5)
       If (feature 4 <= 4.5)
        Predict: 15.0
       Else (feature 4 > 4.5)
        Predict: 10.0
    Else (feature 4 > 5.5)
     If (feature 17541 <= 0.5)
      If (feature 235 <= 7.0)
       If (feature 1791 <= 3.5)
        If (feature 23255 <= 0.5)
         Predict: 0.6462860030318343
        Else (feature 23255 > 0.5)
         Predict: 12.75
       Else (feature 1791 > 3.5)
        Predict: 32.0
      Else (feature 235 > 7.0)
       Predict: 14.0
     Else (feature 17541 > 0.5)
      If (feature 1 <= 9.5)
       Predict: 7.0
      Else (feature 1 > 9.5)
       Predict: 13.0
  Tree 6:
    If (feature 11343 <= 1.5)
     If (feature 5 <= 3.5)
      If (feature 5797 <= 0.5)
       If (feature 3 <= 4.5)
        If (feature 25985 <= 0.5)
         Predict: 0.19384596524564038
        Else (feature 25985 > 0.5)
         Predict: 13.0
       Else (feature 3 > 4.5)
        If (feature 189 <= 10.0)
         Predict: 0.42029604326786224
        Else (feature 189 > 10.0)
         Predict: 10.0
      Else (feature 5797 > 0.5)
       If (feature 22762 <= 0.5)
        If (feature 137 <= 0.5)
         Predict: 0.36363636363636365
        Else (feature 137 > 0.5)
         Predict: 3.0
       Else (feature 22762 > 0.5)
        Predict: 30.0
     Else (feature 5 > 3.5)
      If (feature 54 <= 5.5)
       If (feature 9675 <= 1.5)
        If (feature 11156 <= 0.5)
         Predict: 0.6027293404094011
        Else (feature 11156 > 0.5)
         Predict: 5.25
       Else (feature 9675 > 1.5)
        Predict: 15.0
      Else (feature 54 > 5.5)
       If (feature 0 <= 10.5)
        Predict: 2.0
       Else (feature 0 > 10.5)
        Predict: 13.0
    Else (feature 11343 > 1.5)
     If (feature 0 <= 9.5)
      Predict: 4.0
     Else (feature 0 > 9.5)
      Predict: 52.0
  Tree 7:
```

```
  If (feature 24056 <= 0.5)
   If (feature 5 <= 4.5)
    If (feature 2557 <= 1.5)
     If (feature 3 <= 4.5)
      If (feature 9172 <= 0.5)
       Predict: 0.2058165548098434
      Else (feature 9172 > 0.5)
       Predict: 12.0
     Else (feature 3 > 4.5)
      If (feature 22313 <= 0.5)
       Predict: 0.44150450878025627
      Else (feature 22313 > 0.5)
       Predict: 11.0
    Else (feature 2557 > 1.5)
     If (feature 2 <= 4.5)
      Predict: 17.0
     Else (feature 2 > 4.5)
      Predict: 0.0
   Else (feature 5 > 4.5)
    If (feature 28487 <= 0.5)
     If (feature 1575 <= 1.5)
      If (feature 27115 <= 0.5)
       Predict: 0.6922319166901211
      Else (feature 27115 > 0.5)
       Predict: 9.666666666666666
     Else (feature 1575 > 1.5)
      If (feature 12307 <= 0.5)
       Predict: 0.0
      Else (feature 12307 > 0.5)
       Predict: 17.0
    Else (feature 28487 > 0.5)
     Predict: 21.0
  Else (feature 24056 > 0.5)
   If (feature 0 <= 19.5)
    Predict: 52.0
   Else (feature 0 > 19.5)
    If (feature 8 <= 3.5)
     Predict: 6.0
    Else (feature 8 > 3.5)
     Predict: 0.0
Tree 8:
  If (feature 11343 <= 1.5)
   If (feature 2 <= 5.5)
    If (feature 27412 <= 0.5)
     If (feature 1449 <= 1.5)
      If (feature 4 <= 4.5)
       Predict: 0.206332317248519
      Else (feature 4 > 4.5)
       Predict: 0.4676814988290398
     Else (feature 1449 > 1.5)
      Predict: 17.0
    Else (feature 27412 > 0.5)
     Predict: 13.0
   Else (feature 2 > 5.5)
    If (feature 19697 <= 0.5)
     If (feature 27625 <= 0.5)
      If (feature 11402 <= 0.5)
       Predict: 0.5738242799854174
      Else (feature 11402 > 0.5)
       Predict: 6.769230769230769
     Else (feature 27625 > 0.5)
      Predict: 11.0
    Else (feature 19697 > 0.5)
     If (feature 1 <= 6.5)
      Predict: 32.0
```

```
      Else (feature 1 > 6.5)
       Predict: 0.0
    Else (feature 11343 > 1.5)
     If (feature 11 <= 2.5)
      Predict: 4.0
     Else (feature 11 > 2.5)
      Predict: 52.0
  Tree 9:
    If (feature 11343 <= 1.5)
     If (feature 21655 <= 1.0)
      If (feature 4 <= 4.5)
       If (feature 246 <= 3.5)
        If (feature 1823 <= 1.5)
         Predict: 0.23125205638935992
        Else (feature 1823 > 1.5)
         Predict: 10.0
       Else (feature 246 > 3.5)
        Predict: 17.0
      Else (feature 4 > 4.5)
       If (feature 23160 <= 0.5)
        If (feature 5737 <= 2.0)
         Predict: 0.5815026767583533
        Else (feature 5737 > 2.0)
         Predict: 14.0
       Else (feature 23160 > 0.5)
        If (feature 24269 <= 0.5)
         Predict: 10.428571428571429
        Else (feature 24269 > 0.5)
         Predict: 0.0
     Else (feature 21655 > 1.0)
      Predict: 32.0
    Else (feature 11343 > 1.5)
     If (feature 6 <= 1.5)
      Predict: 4.0
     Else (feature 6 > 1.5)
      Predict: 52.0
```

Question8: Let us now use a Gradient Boosting Trees (GBT) ensemble to predict the number of funny votes. Set the maximum number of iterations to 10. Does this affect the number of trees in the ensemble? Do we need to set the maximum depth of trees in the ensemble? Why or why not?

Answer: By setting the number of iterations to 10, this means the Gradient Bossting Trees will run 10 times. This doesn't affect the number of trees in the ensemble. Yes, we need to set the maximum depth of the trees in the ensemble otherwise it will overfit and perform poorly on the unseen the data. Without a maximum depth size, the base learners will not be weak as needed. Though, the default max_depth of Gradient boosted trees is 3 so it works without explicitly specifying it.

In [ ]:

```
gb_model = GradientBoostedTrees.trainRegressor(doc_vectors_train, numIterations=10,cate

predictions = gb_model.predict(doc_vectors_val.map(lambda x: x.features))
labels_and_predictions = doc_vectors_val.map(lambda lp: lp.label).zip(predictions)
val_mse = labels_and_predictions.map(lambda lp: (lp[0] - lp[1]) * (lp[0] - lp[1])).sum(
    float(doc_vectors_val.count())
print('Validation Root Mean Squared Error (Gradient Boosting Trees) = ' + str(val_mse))
print('Learned regression GBT model:')
print(gb_model.toDebugString())
```

```
Validation Root Mean Squared Error (Gradient Boosting Trees) = 0.7358175384949524
Learned regression GBT model:
```

```
TreeEnsembleModel regressor with 10 trees

  Tree 0:
    If (feature 24056 <= 0.5)
     If (feature 4 <= 4.5)
      If (feature 2138 <= 1.5)
       Predict: 0.23339358912340555
      Else (feature 2138 > 1.5)
       Predict: 10.0
     Else (feature 4 > 4.5)
      If (feature 7214 <= 1.5)
       Predict: 0.5941654357459379
      Else (feature 7214 > 1.5)
       Predict: 32.0
    Else (feature 24056 > 0.5)
     If (feature 2677 <= 0.5)
      If (feature 36 <= 0.5)
       Predict: 5.333333333333333
      Else (feature 36 > 0.5)
       Predict: 0.0
     Else (feature 2677 > 0.5)
      If (feature 0 <= 2.5)
       Predict: 32.0
      Else (feature 0 > 2.5)
       Predict: 52.0
  Tree 1:
    If (feature 27770 <= 0.5)
     If (feature 146 <= 0.5)
      If (feature 8 <= 3.5)
       Predict: -0.09535203928987485
      Else (feature 8 > 3.5)
       Predict: 0.32063953806008694
     Else (feature 146 > 0.5)
      If (feature 26440 <= 0.5)
       Predict: 0.5617213214724145
      Else (feature 26440 > 0.5)
       Predict: 42.81166912850813
    Else (feature 27770 > 0.5)
     If (feature 38 <= 0.5)
      If (feature 3 <= 2.5)
       Predict: 25.29269825733817
      Else (feature 3 > 2.5)
       Predict: 20.81166912850813
     Else (feature 38 > 0.5)
      If (feature 7 <= 2.5)
       Predict: 12.811669128508123
      Else (feature 7 > 2.5)
       Predict: 2.172440975130656
  Tree 2:
    If (feature 27770 <= 0.5)
     If (feature 17 <= 2.5)
      If (feature 54 <= 7.5)
       Predict: -0.052536955658568384
      Else (feature 54 > 7.5)
       Predict: 32.69932486421385
     Else (feature 17 > 2.5)
      If (feature 21692 <= 0.5)
       Predict: 0.4433073133791754
      Else (feature 21692 > 0.5)
       Predict: 34.249335302806685
    Else (feature 27770 > 0.5)
     If (feature 7 <= 1.5)
      If (feature 1 <= 12.5)
       Predict: 20.234158605870533
      Else (feature 1 > 12.5)
```

```
      Predict: 16.64933530280649
    Else (feature 7 > 1.5)
     If (feature 1 <= 12.5)
      Predict: 1.7379527801045254
     Else (feature 1 > 12.5)
      Predict: 10.2493353028065
Tree 3:
  If (feature 2581 <= 2.5)
   If (feature 138 <= 0.5)
    If (feature 1958 <= 2.5)
     Predict: -0.04445951715624452
    Else (feature 1958 > 2.5)
     Predict: 25.380423451465276
   Else (feature 138 > 0.5)
    If (feature 17523 <= 0.5)
     Predict: 0.4605739588996646
    Else (feature 17523 > 0.5)
     Predict: 18.53712340680219
  Else (feature 2581 > 2.5)
   Predict: 32.610663401537806
Tree 4:
  If (feature 5 <= 3.5)
   If (feature 9864 <= 1.5)
    If (feature 6203 <= 1.5)
     Predict: -0.07175196101231436
    Else (feature 6203 > 1.5)
     Predict: 31.57168252417432
   Else (feature 9864 > 1.5)
    Predict: 38.7491321357179
  Else (feature 5 > 3.5)
   If (feature 29916 <= 0.5)
    If (feature 11402 <= 0.5)
     Predict: 0.2499744133995401
    Else (feature 11402 > 0.5)
     Predict: 8.486738791974753
   Else (feature 29916 > 0.5)
    If (feature 0 <= 15.5)
     Predict: 22.850138830929062
    Else (feature 0 > 15.5)
     Predict: 24.76694051545907
Tree 5:
  If (feature 186 <= 0.5)
   If (feature 2581 <= 2.5)
    If (feature 27115 <= 0.5)
     Predict: -0.03693027102695246
    Else (feature 27115 > 0.5)
     Predict: 9.4825760113774
   Else (feature 2581 > 2.5)
    Predict: 26.038535838550388
  Else (feature 186 > 0.5)
   If (feature 9698 <= 0.5)
    If (feature 1513 <= 1.5)
     Predict: 0.44611908739839845
    Else (feature 1513 > 1.5)
     Predict: 30.99930570857441
   Else (feature 9698 > 0.5)
    If (feature 11 <= 1.5)
     Predict: 4.151489274577876
    Else (feature 11 > 1.5)
     Predict: 21.285355492452485
Tree 6:
  If (feature 27770 <= 0.5)
   If (feature 146 <= 0.5)
    If (feature 13900 <= 0.5)
     Predict: -0.03909876728819629
```

```
    Else (feature 13900 > 0.5)
     Predict: 7.244114549333896
   Else (feature 146 > 0.5)
    If (feature 9957 <= 0.5)
     Predict: 0.38880664526335307
    Else (feature 9957 > 0.5)
     Predict: 30.77221407382126
  Else (feature 27770 > 0.5)
   If (feature 7 <= 1.5)
    If (feature 1 <= 12.5)
     Predict: 16.11994106124943
    Else (feature 1 > 12.5)
     Predict: 13.18914144551686
   Else (feature 7 > 1.5)
    If (feature 19 <= 2.5)
     Predict: 5.9798056895986935
    Else (feature 19 > 2.5)
     Predict: -1.2182375794547458
Tree 7:
  If (feature 32 <= 0.5)
   If (feature 158 <= 5.5)
    If (feature 22762 <= 0.5)
     Predict: -0.07529498103365705
    Else (feature 22762 > 0.5)
     Predict: 15.813763182548655
   Else (feature 158 > 5.5)
    Predict: 40.81534975591239
  Else (feature 32 > 0.5)
   If (feature 11402 <= 0.5)
    If (feature 10654 <= 1.5)
     Predict: 0.17125429807397466
    Else (feature 10654 > 1.5)
     Predict: 24.617771259057008
   Else (feature 11402 > 0.5)
    If (feature 124 <= 1.5)
     Predict: 3.6510788050413754
    Else (feature 124 > 1.5)
     Predict: 19.007996127767864
Tree 8:
  If (feature 17 <= 0.5)
   If (feature 2570 <= 1.5)
    If (feature 28453 <= 0.5)
     Predict: -0.08961733430045397
    Else (feature 28453 > 0.5)
     Predict: 20.09356580707731
   Else (feature 2570 > 1.5)
    Predict: 48.15965505692202
  Else (feature 17 > 0.5)
   If (feature 29916 <= 0.5)
    If (feature 15662 <= 0.5)
     Predict: 0.13071548966065213
    Else (feature 15662 > 0.5)
     Predict: 8.36127229252965
   Else (feature 29916 > 0.5)
    If (feature 0 <= 15.5)
     Predict: 18.31037586861301
    Else (feature 0 > 15.5)
     Predict: 19.79450736041549
Tree 9:
  If (feature 1846 <= 0.5)
   If (feature 27770 <= 0.5)
    If (feature 390 <= 9.0)
     Predict: -0.010362179444218881
    Else (feature 390 > 9.0)
     Predict: 12.204685850001738
```

```
    Else (feature 27770 > 0.5)
     If (feature 425 <= 0.5)
      Predict: 0.8484520306214995
     Else (feature 425 > 0.5)
      Predict: 11.208409619359506
   Else (feature 1846 > 0.5)
    If (feature 21510 <= 0.5)
     If (feature 6916 <= 0.5)
      Predict: 1.1151749141566545
     Else (feature 6916 > 0.5)
      Predict: 15.953555916702726
    Else (feature 21510 > 0.5)
     If (feature 0 <= 6.5)
      Predict: 20.829479178391384
     Else (feature 0 > 6.5)
      Predict: 15.343228199596442
```

Question9: Which of the four methods we tried gave the best validation RMSE results?

Answer: Gradient Boosting Trees give the best validation RMSE results - 0.735 (the least amongst all). Hence, it is able to predict the number of funny votes on the Yelp reviews more accurately.

Part 3: Collaborative filtering for recommendation

In this section, we will tackle a collaborative filtering task which can be used to recommend businesses to users based on the ratings they have already assigned to some businesses they have visited.

In [ ]:
```
from pyspark.mllib.recommendation import ALS, MatrixFactorizationModel, Rating
```

Question1: Let us first determine the unique user and business IDs that appear in the reviews data. This will help us build dictionaries for mapping the user/business IDs to unique integer indices. Fill in the required code to build these dictionaries below.

In [ ]:
```
review_user_ids = reviews_rdd.map(lambda x: x['user_id']).distinct().collect()
review_business_ids = reviews_rdd.map(lambda x: x['business_id']).distinct().collect()

user_to_index_dict = {review_user_ids[i]:i for i in range(len(review_user_ids))}
business_to_index_dict = {review_business_ids[i]:i for i in range(len(review_business_i
```

Question2: Next, transform each review into a rating. The Rating object takes a unique user index, a unique business index, and float-valued rating.

In [ ]:
```
ratings_rdd = reviews_rdd.map(lambda x: Rating(user_to_index_dict[x['user_id']], busine
print(ratings_rdd.take(2))
```

```
[Rating(user=15903, product=910, rating=5.0), Rating(user=11610, product=910, rating=2.
0)]
```

Question3: Let us randomly split data into 80% train and 20% validation set.

In [ ]:
```
ratings_rdd_train, ratings_rdd_val = ratings_rdd.randomSplit([0.8,0.2])
```

Question4: For a succession of ranks, we will now build an collaborative filtering algorithm using ALS (Alternating Least Squares). We will use the model to obtain train as well as validation RMSE for each rank. In the cell below, you can fill in the code to carry out the model-building, prediction, and RMSE calculation.

In [ ]:

```
numIterations=10
ranks = list(range(1,20)) + list(range(20, 201, 20))
train_rmses = []
val_rmses = []

for rank in ranks:
    cf_model = ALS.train(ratings_rdd_train,rank,iterations=numIterations)

    train_data = ratings_rdd_train.map(lambda p: (p[0], p[1]))
    predictions = cf_model.predictAll(train_data).map(lambda r: ((r[0], r[1]), r[2]))
    rates_and_preds = ratings_rdd_train.map(lambda r: ((r[0], r[1]), r[2])).join(predic
    train_rmse = np.sqrt(rates_and_preds.map(lambda r: (r[1][0] - r[1][1])**2).mean())
    train_rmses.append(train_rmse)

    val_data = ratings_rdd_val.map(lambda p: (p[0], p[1]))
    predictions = cf_model.predictAll(val_data).map(lambda r: ((r[0], r[1]), r[2]))
    rates_and_preds = ratings_rdd_val.map(lambda r: ((r[0], r[1]), r[2])).join(predicti
    val_rmse = np.sqrt(rates_and_preds.map(lambda r: (r[1][0] - r[1][1])**2).mean())
    val_rmses.append(val_rmse)

    print("Root Mean Squared Error (rank={}) = Train {}, Validation {}".format(rank, tr
```

```
Root Mean Squared Error (rank=1) = Train 2.4118501787688444, Validation 10.8878487831795
98
Root Mean Squared Error (rank=2) = Train 0.7307324074692422, Validation 2.90922983508792
13
Root Mean Squared Error (rank=3) = Train 0.6255490690471001, Validation 2.58093477419481
63
Root Mean Squared Error (rank=4) = Train 0.531465954696533, Validation 2.313501106860287
8
Root Mean Squared Error (rank=5) = Train 0.512365289423873, Validation 2.48793065730135
Root Mean Squared Error (rank=6) = Train 0.44872528516731247, Validation 2.2827710957147
18
Root Mean Squared Error (rank=7) = Train 0.4071248879719336, Validation 2.20590682176311
15
Root Mean Squared Error (rank=8) = Train 0.3775727238222933, Validation 2.18680205651154
7
Root Mean Squared Error (rank=9) = Train 0.3548101849692834, Validation 2.21235467411604
7
Root Mean Squared Error (rank=10) = Train 0.3256252031819768, Validation 2.1796442359147
594
Root Mean Squared Error (rank=11) = Train 0.30944374896395865, Validation 2.145398388497
2043
Root Mean Squared Error (rank=12) = Train 0.29313548436389913, Validation 2.124945694118
6082
Root Mean Squared Error (rank=13) = Train 0.27659883420627235, Validation 2.136902539961
3886
Root Mean Squared Error (rank=14) = Train 0.2579444873217719, Validation 2.0406596012283
39
Root Mean Squared Error (rank=15) = Train 0.2511450132320192, Validation 2.0791207173680
992
Root Mean Squared Error (rank=16) = Train 0.2417916874676037, Validation 2.0554875125656
693
Root Mean Squared Error (rank=17) = Train 0.2351185922389638, Validation 2.0620568912926
```

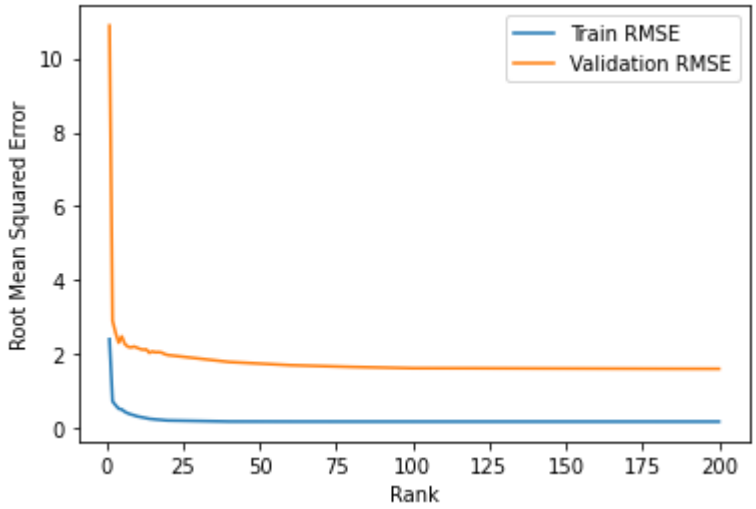
```
8
Root Mean Squared Error (rank=18) = Train 0.22836129671397132, Validation 2.050538767726
722
Root Mean Squared Error (rank=19) = Train 0.22013440851867205, Validation 2.007594031622
3417
Root Mean Squared Error (rank=20) = Train 0.2137891231399327, Validation 1.9824762268907
Root Mean Squared Error (rank=40) = Train 0.18490646328327245, Validation 1.794577767641
7987
Root Mean Squared Error (rank=60) = Train 0.18292794931622686, Validation 1.707773065523
8699
Root Mean Squared Error (rank=80) = Train 0.1823727882133683, Validation 1.6605764911460
832
Root Mean Squared Error (rank=100) = Train 0.1821831035448625, Validation 1.625950857311
9124
Root Mean Squared Error (rank=120) = Train 0.18204466054875182, Validation 1.62427893923
1615
Root Mean Squared Error (rank=140) = Train 0.1820187071924242, Validation 1.620044130526
5274
Root Mean Squared Error (rank=160) = Train 0.1820197760903513, Validation 1.616359877584
0464
Root Mean Squared Error (rank=180) = Train 0.18196990591787873, Validation 1.61235009260
58547
Root Mean Squared Error (rank=200) = Train 0.181970266395981, Validation 1.6094432864086
996
```

Question5: Let us plot the train and validation RMSE versus the rank. The code below does this for you. Based on this plot, what would your choice of the rank hyperparameter be? Is this choice conclusive or do we need to conduct a more extensive hyperparameter search at larger ranks than the ones we have evaluated?

Answer: The choice of the rank hyperparameter according to the plot below is around 24 and onwards as that's when the validation RMSE curve plateaus. Yes, this choice is conclusive as we can observe that after 24th rank, the curve plateaus for the higher ranks. From the values printed above, we can see that from rank 80 onwards there's change in hundredth decimal place of RMSE which is very minute and changes between rank 20, 40 and 60 are little too.

In [ ]:
```
fig, ax = plt.subplots()
ax.plot(ranks, train_rmses, label='Train RMSE')
ax.plot(ranks, val_rmses, label='Validation RMSE')

plt.xlabel('Rank')
plt.ylabel('Root Mean Squared Error')
plt.legend()
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x14cfe45ebcd0>
```



Part 4: Topic modeling for text reviews

In this section, we will build and examine a Bayesian topic model named Latent Dirichlet Allocation (LDA). The goal of textual topic modeling is to discover latent topics whose mixtures generate textual documents through a stylized probabilistic generatve model. The topics often have semantic meaning. They may be associated with various aspects discussed in the text corpus such as politics, health, education, etc. Topic models are unsupervised machine learning algorithms. Hence, the nature of discovered topics is entirely dependent of the context of your dataset.

In [ ]:
```
from pyspark.mllib.linalg import Vectors, DenseVector, SparseVector
from pyspark.mllib.clustering import LDA, LDAModel
```

Question1: Let us create a new RDD of just textual reviews from reviews_rdd, obtain word counts, and build a list of unique words that do not include stop words. Use num_stop_words as a measure of how many of the most frequent words to filter out.

In [ ]:
```
# number of stopwords i.e. most frequent words to remove
# removal of stopwords such as a, the, from, etc. that occur across a vast majority of
num_stop_words = 1000

all_reviews = reviews_rdd.map(lambda x: x['text'])
word_counts = list(all_reviews.flatMap(lambda x: x.lower().split()).map(lambda x: (x,1)
# sort words in descending order of frequency
word_counts = sorted(word_counts, key=lambda x: -x[1])

# remove stopwords
unique_words = [x[0] for x in word_counts[num_stop_words:]]
num_unique_words = len(unique_words)
print('Number of unique words: ', num_unique_words)
```

```
Number of unique words:  192742
```

Question2: We will now construct two dictionaries - one which maps from each word to a unique integer index and the second one which maps back from the index to the word. Write the code required to do this.

In [ ]:
```
word_to_index_dict = {unique_words[i]:i for i in range(len(unique_words))}
index_to_word_dict = {i:unique_words[i] for i in range(len(unique_words))}
```

Question3: Construct an RDD of SparseVectors. Each SparseVector is built using the word counts of a review. Hence, the RDD of SparseVectors should be obtained as a map from the RDD of document word counts.

In [ ]:
```
from collections import Counter

doc_vectors = all_reviews.map(lambda x: x.lower().split()).map(lambda x: [word_to_index
doc_vectors = doc_vectors.map(lambda x: SparseVector(num_unique_words,dict(Counter(x)))

# zipWithIndex result needs a minor transform to be acceptable to the LDA training proc
doc_vectors = doc_vectors.zipWithIndex().map(lambda x:[x[1],x[0]])
print(doc_vectors.count())
print(doc_vectors.take(2))
```

```
62608
[[0, SparseVector(192742, {8: 2.0, 114: 1.0, 311: 1.0, 1209: 1.0, 1315: 1.0, 1444: 1.0,
1805: 1.0, 1987: 1.0, 2249: 1.0, 2738: 1.0, 2965: 1.0, 3148: 1.0, 3166: 1.0, 3530: 1.0,
3614: 1.0, 3977: 1.0, 4305: 1.0, 4391: 1.0, 4700: 1.0, 6041: 1.0, 8947: 1.0, 11064: 1.0,
13456: 1.0, 14371: 1.0, 23207: 1.0, 54812: 1.0, 77509: 1.0, 77510: 1.0, 77511: 1.0, 7751
2: 1.0})], [1, SparseVector(192742, {1805: 1.0, 2231: 1.0, 4799: 1.0, 5281: 1.0, 8776:
1.0, 12656: 1.0, 15459: 1.0, 17524: 1.0})]]
```

Question4: Train an LDA model with a 100 topics and the random seed set to 42.

In [ ]:
```
lda_model = LDA.train(doc_vectors,k=100,seed=42)
```

Question5: Display the LDA model vocabulary size.

In [ ]:
```
print('Model vocabulary size: ', lda_model.vocabSize())
```

```
Model vocabulary size:  192742
```

Question6: Display 5 learned topics and the top 100 terms that appear in each of these topics. Assign a semantic label/meaning to each of them (e.g. food, ambience, drinks, service, etc.) You can access the topic matrix using the function topicsMatrix on the model. Do the topics learned from Yelp reviews look representative of the corpus?

In [ ]:
```
topicMat = lda_model.topicsMatrix()[:,:5]
sorted_mat = np.argsort(-topicMat,axis=0)[:100,:]
```

In [ ]:
```
top_5 = []

for i in range(sorted_mat.shape[1]):
    top_100 = []
    for j in range(sorted_mat.shape[0]):
        top_100.append(index_to_word_dict[sorted_mat[j,i]])
    top_5.append(top_100)
```

In [ ]:
```
pd.set_option('display.max_rows', 1000)
data = pd.DataFrame(top_5)
data = data.T
data.columns=(['Topic 1','Topic 2','Topic 3','Topic 4','Topic 5'])
data.set_index(np.arange(1,101))
```

Out[ ]:

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
1	charge	choices	choices	noodle	thank
2	up,	covered	bean	bean	pancakes
3	bottom	general	burrito	sour	you'd
4	true	pepper	limited	pepper	up,
5	pancakes	thick	covered	is,	times.
6	you'd	bean	somewhere	generally	general
7	great!	bottom	you'd	best.	museum
8	thick	charge	great!	paying	limited
9	covered	finished	thick	dressing	choices
10	choices	room.	thin	up,	,
11	limited	up,	paying	thick	bottom
12	paying	times.	finished	share	sour
13	best.	sign	up,	wall	somewhere
14	generally	tasty,	true	burrito	...
15	times.	wall	general	finished	wait.
16	orders	best.	charge	except	finished
17	stayed	you'd	share	tasty,	best.
18	sign	true	pepper	inside.	sign
19	thinking	paying	sour	choices	pizza,
20	forget	somewhere	best.	pancakes	tasty,
21	wall	side,	pancakes	thank	please
22	thank	generally	wall	avoid	atmosphere,
23	room.	forget	noodle	creamy	great!
24	pizza,	sour	oil	special.	pepper
25	sour	except	bottom	we're	dressing
26	pepper	clearly	please	charge	burrito
27	general	limited	atmosphere,	forget	area,
28	card	pricey	generally	great!	personal
29	somewhere	please	mine	bottom	card

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
30	life	pancakes	fresh.	times.	noodle
31	recommended	personal	tasty,	side,	is,
32	please	seriously	dressing	apple	pricey
33	meal,	apple	creamy	hand	bars
34	hand	thank	burger.	somewhere	hand
35	penn	dressing	pizza,	chili	cook
36	area,	they've	forget	true	generally
37	right.	meal,	side,	sign	we're
38	location.	great!	orders	visit.	except
39	tasty,	small,	perfect.	thinking	room.
40	atmosphere,	creamy	seriously	you'd	mine
41	is,	mine	times.	oil	rooms
42	personal	cook	recommended	covered	thick
43	chili	oil	wait.	small,	recommended
44	mine	atmosphere,	except	pizza,	share
45	minute	employees	hand	atmosphere,	charge
46	pricey	means	inside.	cook	side,
47	seriously	selection.	thank	room.	minute
48	avoid	hand	smaller	clearly	recent
49	terrible	amazing!	meal,	bread.	they've
50	clearly	special.	toast	mine	greasy
51	finished	chili	is,	,	paying
52	noodle	perfect.	greasy	please	clearly
53	employees	honestly	la	area,	means
54	share	minutes.	chili	toast	inside.
55	inside.	thin	apple	and,	longer
56	town.	share	sign	seriously	forget
57	greasy	recent	special.	recommended	avoid
58	small,	bread.	clearly	orders	special.
59	visit.	penn	small,	amazing!	normally
60	means	you,	thinking	pumpkin	life
61	except	greasy	them,	terrible	selection.
62	they've	longer	bread.	drink.	creamy

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
63	else.	terrible	drink.	pricey	seriously
64	you,	recommended	whatever	limited	honestly
65	dogs	visit.	life	spinach	orders
66	dressing	supposed	else.	means	wall
67	we're	noodle	area,	meal,	else.
68	apple	okay,	supposed	general	back!
69	side,	thinking	honestly	employees	covered
70	station	life	they've	okay,	them,
71	rooms	byob	and,	picked	bread.
72	bread.	toast	...	dogs	you,
73	original	total	cook	dinner,	and,
74	total	pizza,	options.	selection.	fat
75	oakland	fresh.	bunch	wait.	toast
76	special.	and,	employees	greasy	hostess
77	works	else.	byob	else.	works
78	okay,	delivery	waffle	nearly	terrible
79	...	town.	banana	supposed	small,
80	cook	is,	delivery	life	bean
81	picked	area,	nearly	minute	picked
82	them,	rooms	back!	foods	drink.
83	waffle	,	yummy	smaller	la
84	kinda	we're	minute	fat	apple
85	(not	hostess	total	byob	thinking
86	and,	la	means	personal	employees
87	perfect.	location.	we're	8	true
88	fresh.	credit	hostess	penn	visit.
89	nearly	nearly	minutes.	suggest	whatever
90	amazing!	fat	terrible	perfect.	supposed
91	selection.	corner	avoid	you,	location.
92	play	minute	you,	whatever	wide
93	,	oh,	penn	honestly	kinda
94	bean	orders	slice	card	hill
95	oil	right.	personal	thin	times,

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
96	thin	inside.	slices	dip	burger.
97	longer	(not	amazing!	weeks	fresh.
98	bunch	whatever	pricey	they've	center
99	minutes.	picked	barely	fresh.	bunch
100	recent	burrito	recent	town.	yummy

By looking at the top hundred words given by our LDA model in five topics, we can infer the following:

Topic 1- Reviews focussing on ambience

- Words like "perfect", "longer", "amazing!", "visit", "selection", "recommended", "best!" etc are commonly observed words in any reviews focussing on ambience of a cafe/restaurant.

Topic 2- Reviews focussing on menu

- Words like "choices", "dressing" , "creamy","pancakes", "pricey", "covered", "picked", "right", "general", "best" etc are commonly observed words in an reviews focussing on food menu of a cafe/restaurant.

Topic 3- Reviews on order instructions

- Words like "thick", "thin", "creamy", "drink", "slices", "options", "smaller", "share" etc are words that we use while giving instructions with our order. Say for example, we want square cut or pie cut pizza slices or coffee with cream.

Topic 4- Reviews focussing on food

- Words like "noodle", "bean", "sour", "pepper", "burrito", "pancakes", "pizza", "bread" etc are commonly observed in reviews describing food of any place.

Topic 5- Reviews on service

- Words like "thank", "best", "wait", "employees", "hostess", "terrible", "forget", "they've", "we're", "hand", "cook", "visit" etc are words than we commonly see when a service of employees is being described.

Part 5: Word2Vec for text reviews

In this section, we will fit a Word2Vec model to the Yelp reviews text. Word2Vec is a popular model for embedding words in Euclidean space so they can be analyzed similar to real-valued vectors. Contrary to popular belief, Word2Vec models are not deep neural models. Inspite of being shallow neural networks, they capture word associations and analogies remarkably well.

In [ ]:
```
from pyspark.mllib.feature import Word2Vec
import re
pattern = re.compile('[\W_]+')
```

In [ ]:
```
review_docs = reviews_rdd.map(lambda x : x['text'].lower().split())
review_docs = review_docs.map(lambda x : [pattern.sub('', w) for w in x])
print(review_docs.take(2))
```

```
[['thank', 'you', 'rob', 'i', 'truly', 'appreciated', 'all', 'the', 'help', 'i', 'receiv
ed', 'from', 'this', 'agent', 'today', 'who', 'was', 'able', 'to', 'removed', 'the', 'ex
tra', 'charges', 'on', 'my', 'bill', 'that', 'the', 'pasadena', 'verizon', 'store', 'o
n', 'lake', 'was', 'charging', 'me', 'on', 'my', 'bill', 'for', 'upgrading', 'my', 'phon
e', 'when', 'i', 'went', 'in', 'i', 'was', 'having', 'problems', 'with', 'my', 'blacberr
y', 'and', 'had', 'to', 'switch', 'to', 'the', 'iphone', 'last', 'week', 'rob', 'from',
'the', 'pennsylvania', 'store', 'who', 'i', 'was', 'connected', 'today', 'was', 'able',
'to', 'look', 'at', 'my', 'bill', 'and', 'all', 'the', 'notes', 'and', 'correct', 'the',
'problem', 'immediately', 'great', 'customer', 'service', 'he', 'even', 'set', 'up',
'a', 'follow', 'up', 'phone', 'call', 'with', 'me', 'on', 'july', '5th', 'to', 'make',
'sure', 'the', 'credit', 'goes', 'through', 'on', 'my', 'billi', 'cant', 'thank', 'him',
'enough'], ['after', 'waiting', 'for', 'almost', '30', 'minutes', 'to', 'trade', 'in',
'an', 'old', 'phone', 'part', 'of', 'the', 'buy', 'back', 'program', 'our', 'customer',
'service', 'rep', 'incorrectly', 'processed', 'the', 'transaction', 'this', 'led', 'to',
'us', 'waiting', 'another', '30', 'minutes', 'for', 'him', 'to', 'correct', 'it', 'don
t', 'visit', 'this', 'store', 'if', 'you', 'want', 'pleasant', 'or', 'good', 'service']]
```

Question1: Fit a Word2Vec model to the review_docs RDD. Set the size of embedding vectors to 10, the random seed to 42, and the number of iterations to 10.

In [ ]:
```
word2vec_model = Word2Vec().setSeed(42).setVectorSize(10).setNumIterations(10).fit(revi
```

Let's us examine what words are closely associated with some example words. Run the cell below to see word associations. Feel free to add any additional words whose results you find interesting, but do not delete any of the words already in the list.

In [ ]:
```
for word in ['salt', 'pepper', 'restaurant', 'italian', 'indian', 'chinese', 'direction
    syms = word2vec_model.findSynonyms(word, 5)
    print('Words most similar to ', word, ' : ', [s[0] for s in syms])
```

```
Words most similar to  salt  :  ['cholula', 'fluff', 'burst', 'smear', 'coarse']
Words most similar to  pepper  :  ['stewed', 'garlic', 'deep', 'sriracha', 'ranch']
Words most similar to  restaurant  :  ['resturant', 'restaraunt', 'establishment', 'loca
tion', 'soho']
Words most similar to  italian  :  ['style', 'shacks', 'greek', 'neopolitan', 'tradition
al']
Words most similar to  indian  :  ['chinese', 'chinesetaiwanese', 'mexican', 'japanese',
'cuisine']
Words most similar to  chinese  :  ['indian', 'chinesetaiwanese', 'mexican', 'japanese',
'korean']
Words most similar to  direction  :  ['driveway', 'greyhound', 'post', 'tripadvisor', 'a
partment']
Words most similar to  pittsburgh  :  ['pgh', 'connecticut', 'burgh', 'bloomfield', 'ne
w']
Words most similar to  burgh  :  ['lebanon', 'city', 'berlin', 'lebo', 'burg']
Words most similar to  city  :  ['robinson', 'hiptique', 'burg', 'bloomfield', 'northsid
e']
Words most similar to  location  :  ['toonseum', 'neighborhood', 'scene', 'timebomb', 'p
burgh']
```

```
Words most similar to  cmu  :  ['college', 'undergrad', 'philadelphian', 'campus', 'scho
ol']
Words most similar to  pizza  :  ['sub', 'barbecue', 'bbq', 'superfresh', 'hoagie']
```

Question2: What "synonyms" in the result above give rise to perfect analogies? Are there words in the result that are spurious and not good substitutes for the originally supplied word?

Answer: The "synonyms" in the result above do not give rise to perfect analogies for some of them. The words aren't exact synonyms but describe a characteristic of it. For pittsburhg, the synonym "pgh", "burgh" make sense. Synonyms for "cmu" give a good analogy of an academic instition. Same is the case for "pizza" too, as sub, barbecue,bbq help in guessing it a fast food type.

Some of the words are spurious and not good subtitutes for the originally supplied word. For example, the synonyms for salt do not really make sense. The closest characteristic trait of salt is representated by coarse but this still cannot be used a it's synonym. Even for pepper, not exact synonym but sriracha alines with it. For italian, indian and chinese, the synonyms generated are other cuisine, which makes a good analogy of different cuisines but it isn't synchronous with italian, indian and chinese cuisine directly.

Part 6: Frequent pattern mining using FP-Growth algorithm

In this section, we will mine frequent subsets of items that appear together in datapoints. This type of analysis is also known as frequent itemset mining or market basket analysis. Since the tags associated with Yelp businesses are sets, we can use them to carry out the frequent item set mining by employing the FP-Growth algorithm available in Spark.

In [ ]:
```
from pyspark.mllib.fpm import FPGrowth
```

Question1: Fill in the required code to perform itemset mining on business categories represented as an RDD of sets. Train the FP-Growth algorithm with a minimum support parameter of 0.01 and 10 partitions.

In [ ]:
```
business_categories = businesses_rdd.map(lambda x: x['categories'])

fpgrowth_model = FPGrowth.train(business_categories, minSupport=0.01, numPartitions=10)
result = sorted(fpgrowth_model.freqItemsets().collect(), key=lambda x: -x[1])
for fi in result:
    if len(fi[0]) > 1:
        print(fi)
```

```
FreqItemset(items=['Bars', 'Nightlife'], freq=3628)
FreqItemset(items=['Fashion', 'Shopping'], freq=2566)
FreqItemset(items=['Fast Food', 'Restaurants'], freq=2383)
FreqItemset(items=['Pizza', 'Restaurants'], freq=2223)
FreqItemset(items=['Mexican', 'Restaurants'], freq=2208)
FreqItemset(items=['American (Traditional)', 'Restaurants'], freq=2113)
FreqItemset(items=['Nightlife', 'Restaurants'], freq=2045)
FreqItemset(items=['Sandwiches', 'Restaurants'], freq=1981)
FreqItemset(items=['Bars', 'Nightlife', 'Restaurants'], freq=1934)
FreqItemset(items=['Bars', 'Restaurants'], freq=1934)
```

```
FreqItemset(items=['Coffee & Tea', 'Food'], freq=1890)
FreqItemset(items=['Food', 'Restaurants'], freq=1807)
FreqItemset(items=['Italian', 'Restaurants'], freq=1633)
FreqItemset(items=['Chinese', 'Restaurants'], freq=1496)
FreqItemset(items=['American (New)', 'Restaurants'], freq=1494)
FreqItemset(items=['Burgers', 'Restaurants'], freq=1481)
FreqItemset(items=['Hair Salons', 'Beauty & Spas'], freq=1388)
FreqItemset(items=['Hotels & Travel', 'Event Planning & Services'], freq=1339)
FreqItemset(items=['Hotels', 'Event Planning & Services'], freq=1307)
FreqItemset(items=['Hotels', 'Hotels & Travel'], freq=1307)
FreqItemset(items=['Hotels', 'Hotels & Travel', 'Event Planning & Services'], freq=1307)
FreqItemset(items=['Nail Salons', 'Beauty & Spas'], freq=1256)
FreqItemset(items=['Grocery', 'Food'], freq=1233)
FreqItemset(items=['Auto Repair', 'Automotive'], freq=1220)
FreqItemset(items=['Home & Garden', 'Shopping'], freq=1173)
FreqItemset(items=['Breakfast & Brunch', 'Restaurants'], freq=1116)
FreqItemset(items=['Doctors', 'Health & Medical'], freq=1077)
FreqItemset(items=['Fitness & Instruction', 'Active Life'], freq=1068)
FreqItemset(items=['Specialty Food', 'Food'], freq=1001)
FreqItemset(items=['Bakeries', 'Food'], freq=941)
FreqItemset(items=["Women's Clothing", 'Shopping'], freq=916)
FreqItemset(items=["Women's Clothing", 'Fashion'], freq=916)
FreqItemset(items=["Women's Clothing", 'Fashion', 'Shopping'], freq=916)
FreqItemset(items=['Ice Cream & Frozen Yogurt', 'Food'], freq=867)
FreqItemset(items=['Real Estate', 'Home Services'], freq=850)
FreqItemset(items=['Pubs', 'Bars'], freq=784)
FreqItemset(items=['Pubs', 'Bars', 'Nightlife'], freq=784)
FreqItemset(items=['Pubs', 'Nightlife'], freq=784)
FreqItemset(items=['Cafes', 'Restaurants'], freq=776)
FreqItemset(items=['Dentists', 'Health & Medical'], freq=752)
FreqItemset(items=['Japanese', 'Restaurants'], freq=746)
FreqItemset(items=['Sports Bars', 'Bars'], freq=713)
FreqItemset(items=['Sports Bars', 'Bars', 'Nightlife'], freq=713)
FreqItemset(items=['Sports Bars', 'Nightlife'], freq=713)
FreqItemset(items=['Sushi Bars', 'Restaurants'], freq=671)
FreqItemset(items=['Burgers', 'Fast Food'], freq=654)
FreqItemset(items=['Burgers', 'Fast Food', 'Restaurants'], freq=654)
FreqItemset(items=['Delis', 'Restaurants'], freq=649)
FreqItemset(items=['Italian', 'Pizza'], freq=641)
FreqItemset(items=['Italian', 'Pizza', 'Restaurants'], freq=641)
FreqItemset(items=['Pet Services', 'Pets'], freq=634)
FreqItemset(items=['American (Traditional)', 'Nightlife'], freq=617)
FreqItemset(items=['American (Traditional)', 'Nightlife', 'Restaurants'], freq=617)
FreqItemset(items=['American (Traditional)', 'Bars'], freq=612)
FreqItemset(items=['American (Traditional)', 'Bars', 'Nightlife'], freq=612)
FreqItemset(items=['American (Traditional)', 'Bars', 'Nightlife', 'Restaurants'], freq=6
12)
FreqItemset(items=['American (Traditional)', 'Bars', 'Restaurants'], freq=612)
```

Question2: Fill in the required code to perform itemset mining on business categories represented as an RDD of sets. Train the FP-Growth algorithm with a minimum support parameter of 0.001 and 10 partitions.

In [ ]:

```
fpgrowth_model = FPGrowth.train(business_categories, minSupport=0.001, numPartitions=10
result = sorted(fpgrowth_model.freqItemsets().collect(), key=lambda x: -x[1])
for fi in result:
    if len(fi[0]) > 1:
        print(fi)
```

```
FreqItemset(items=['Bars', 'Nightlife'], freq=3628)
FreqItemset(items=['Fashion', 'Shopping'], freq=2566)
FreqItemset(items=['Fast Food', 'Restaurants'], freq=2383)
FreqItemset(items=['Pizza', 'Restaurants'], freq=2223)
```

```
FreqItemset(items=['Mexican', 'Restaurants'], freq=2208)
FreqItemset(items=['American (Traditional)', 'Restaurants'], freq=2113)
FreqItemset(items=['Nightlife', 'Restaurants'], freq=2045)
FreqItemset(items=['Sandwiches', 'Restaurants'], freq=1981)
FreqItemset(items=['Bars', 'Nightlife', 'Restaurants'], freq=1934)
FreqItemset(items=['Bars', 'Restaurants'], freq=1934)
FreqItemset(items=['Coffee & Tea', 'Food'], freq=1890)
FreqItemset(items=['Food', 'Restaurants'], freq=1807)
FreqItemset(items=['Italian', 'Restaurants'], freq=1633)
FreqItemset(items=['Chinese', 'Restaurants'], freq=1496)
FreqItemset(items=['American (New)', 'Restaurants'], freq=1494)
FreqItemset(items=['Burgers', 'Restaurants'], freq=1481)
FreqItemset(items=['Hair Salons', 'Beauty & Spas'], freq=1388)
FreqItemset(items=['Hotels & Travel', 'Event Planning & Services'], freq=1339)
FreqItemset(items=['Hotels', 'Event Planning & Services'], freq=1307)
FreqItemset(items=['Hotels', 'Hotels & Travel'], freq=1307)
FreqItemset(items=['Hotels', 'Hotels & Travel', 'Event Planning & Services'], freq=1307)
FreqItemset(items=['Nail Salons', 'Beauty & Spas'], freq=1256)
FreqItemset(items=['Grocery', 'Food'], freq=1233)
FreqItemset(items=['Auto Repair', 'Automotive'], freq=1220)
FreqItemset(items=['Home & Garden', 'Shopping'], freq=1173)
FreqItemset(items=['Breakfast & Brunch', 'Restaurants'], freq=1116)
FreqItemset(items=['Doctors', 'Health & Medical'], freq=1077)
FreqItemset(items=['Fitness & Instruction', 'Active Life'], freq=1068)
FreqItemset(items=['Specialty Food', 'Food'], freq=1001)
FreqItemset(items=['Bakeries', 'Food'], freq=941)
FreqItemset(items=["Women's Clothing", 'Shopping'], freq=916)
FreqItemset(items=["Women's Clothing", 'Fashion'], freq=916)
FreqItemset(items=["Women's Clothing", 'Fashion', 'Shopping'], freq=916)
FreqItemset(items=['Ice Cream & Frozen Yogurt', 'Food'], freq=867)
FreqItemset(items=['Real Estate', 'Home Services'], freq=850)
FreqItemset(items=['Pubs', 'Bars'], freq=784)
FreqItemset(items=['Pubs', 'Bars', 'Nightlife'], freq=784)
FreqItemset(items=['Pubs', 'Nightlife'], freq=784)
FreqItemset(items=['Cafes', 'Restaurants'], freq=776)
FreqItemset(items=['Dentists', 'Health & Medical'], freq=752)
FreqItemset(items=['Japanese', 'Restaurants'], freq=746)
FreqItemset(items=['Sports Bars', 'Bars'], freq=713)
FreqItemset(items=['Sports Bars', 'Bars', 'Nightlife'], freq=713)
FreqItemset(items=['Sports Bars', 'Nightlife'], freq=713)
FreqItemset(items=['Sushi Bars', 'Restaurants'], freq=671)
FreqItemset(items=['Burgers', 'Fast Food'], freq=654)
FreqItemset(items=['Burgers', 'Fast Food', 'Restaurants'], freq=654)
FreqItemset(items=['Delis', 'Restaurants'], freq=649)
FreqItemset(items=['Italian', 'Pizza'], freq=641)
FreqItemset(items=['Italian', 'Pizza', 'Restaurants'], freq=641)
FreqItemset(items=['Pet Services', 'Pets'], freq=634)
FreqItemset(items=['American (Traditional)', 'Nightlife'], freq=617)
FreqItemset(items=['American (Traditional)', 'Nightlife', 'Restaurants'], freq=617)
FreqItemset(items=['American (Traditional)', 'Bars'], freq=612)
FreqItemset(items=['American (Traditional)', 'Bars', 'Nightlife'], freq=612)
FreqItemset(items=['American (Traditional)', 'Bars', 'Nightlife', 'Restaurants'], freq=6
12)
FreqItemset(items=['American (Traditional)', 'Bars', 'Restaurants'], freq=612)
FreqItemset(items=['Food', 'Shopping'], freq=604)
FreqItemset(items=['Beauty & Spas', 'Shopping'], freq=595)
FreqItemset(items=['Sporting Goods', 'Shopping'], freq=585)
FreqItemset(items=['Convenience Stores', 'Food'], freq=578)
FreqItemset(items=['Desserts', 'Food'], freq=576)
FreqItemset(items=['Department Stores', 'Shopping'], freq=573)
FreqItemset(items=['Department Stores', 'Fashion'], freq=573)
FreqItemset(items=['Department Stores', 'Fashion', 'Shopping'], freq=573)
FreqItemset(items=['Cosmetics & Beauty Supply', 'Shopping'], freq=563)
FreqItemset(items=['Cosmetics & Beauty Supply', 'Beauty & Spas'], freq=563)
FreqItemset(items=['Cosmetics & Beauty Supply', 'Beauty & Spas', 'Shopping'], freq=563)
```

```
FreqItemset(items=['Drugstores', 'Shopping'], freq=561)
FreqItemset(items=['Gyms', 'Fitness & Instruction'], freq=560)
FreqItemset(items=['Gyms', 'Fitness & Instruction', 'Active Life'], freq=560)
FreqItemset(items=['Gyms', 'Active Life'], freq=560)
FreqItemset(items=['Steakhouses', 'Restaurants'], freq=554)
FreqItemset(items=['Seafood', 'Restaurants'], freq=554)
FreqItemset(items=['Lounges', 'Bars'], freq=536)
FreqItemset(items=['Lounges', 'Bars', 'Nightlife'], freq=536)
FreqItemset(items=['Lounges', 'Nightlife'], freq=536)
FreqItemset(items=['Day Spas', 'Beauty & Spas'], freq=519)
FreqItemset(items=['Chicken Wings', 'Restaurants'], freq=516)
FreqItemset(items=['Coffee & Tea', 'Food', 'Restaurants'], freq=513)
FreqItemset(items=['Coffee & Tea', 'Restaurants'], freq=513)
FreqItemset(items=['Sports Bars', 'Bars', 'Nightlife', 'Restaurants'], freq=512)
FreqItemset(items=['Sports Bars', 'Bars', 'Restaurants'], freq=512)
FreqItemset(items=['Sports Bars', 'Nightlife', 'Restaurants'], freq=512)
FreqItemset(items=['Sports Bars', 'Restaurants'], freq=512)
FreqItemset(items=['Accessories', 'Shopping'], freq=510)
FreqItemset(items=['Accessories', 'Fashion'], freq=510)
FreqItemset(items=['Accessories', 'Fashion', 'Shopping'], freq=510)
FreqItemset(items=['Books, Mags, Music & Video', 'Shopping'], freq=505)
FreqItemset(items=['General Dentistry', 'Dentists'], freq=496)
FreqItemset(items=['General Dentistry', 'Dentists', 'Health & Medical'], freq=496)
FreqItemset(items=['General Dentistry', 'Health & Medical'], freq=496)
FreqItemset(items=["Men's Clothing", 'Shopping'], freq=494)
FreqItemset(items=["Men's Clothing", 'Fashion'], freq=494)
FreqItemset(items=["Men's Clothing", 'Fashion', 'Shopping'], freq=494)
FreqItemset(items=['Flowers & Gifts', 'Shopping'], freq=492)
FreqItemset(items=['Arts & Crafts', 'Shopping'], freq=492)
FreqItemset(items=['Mediterranean', 'Restaurants'], freq=490)
FreqItemset(items=['Massage', 'Beauty & Spas'], freq=489)
FreqItemset(items=['Tires', 'Automotive'], freq=487)
FreqItemset(items=['Arts & Entertainment', 'Nightlife'], freq=479)
FreqItemset(items=['Barbeque', 'Restaurants'], freq=477)
FreqItemset(items=['Skin Care', 'Beauty & Spas'], freq=475)
FreqItemset(items=['Thai', 'Restaurants'], freq=472)
FreqItemset(items=['Apartments', 'Home Services'], freq=469)
FreqItemset(items=['Apartments', 'Real Estate'], freq=469)
FreqItemset(items=['Apartments', 'Real Estate', 'Home Services'], freq=469)
FreqItemset(items=['Beer, Wine & Spirits', 'Food'], freq=465)
FreqItemset(items=['Jewelry', 'Shopping'], freq=454)
FreqItemset(items=['American (New)', 'Nightlife'], freq=446)
FreqItemset(items=['American (New)', 'Nightlife', 'Restaurants'], freq=446)
FreqItemset(items=['Asian Fusion', 'Restaurants'], freq=446)
FreqItemset(items=['Hair Removal', 'Beauty & Spas'], freq=439)
FreqItemset(items=['French', 'Restaurants'], freq=433)
FreqItemset(items=['Oil Change Stations', 'Automotive'], freq=433)
FreqItemset(items=['Sandwiches', 'Fast Food'], freq=428)
FreqItemset(items=['Sandwiches', 'Fast Food', 'Restaurants'], freq=428)
FreqItemset(items=['Dry Cleaning & Laundry', 'Local Services'], freq=425)
FreqItemset(items=['American (New)', 'Bars'], freq=425)
FreqItemset(items=['American (New)', 'Bars', 'Nightlife'], freq=425)
FreqItemset(items=['American (New)', 'Bars', 'Nightlife', 'Restaurants'], freq=425)
FreqItemset(items=['American (New)', 'Bars', 'Restaurants'], freq=425)
FreqItemset(items=['Buffets', 'Restaurants'], freq=420)
FreqItemset(items=['Gas & Service Stations', 'Automotive'], freq=412)
FreqItemset(items=['Veterinarians', 'Pets'], freq=401)
FreqItemset(items=['Performing Arts', 'Arts & Entertainment'], freq=401)
FreqItemset(items=['Venues & Event Spaces', 'Event Planning & Services'], freq=397)
FreqItemset(items=['Furniture Stores', 'Shopping'], freq=397)
FreqItemset(items=['Furniture Stores', 'Home & Garden'], freq=397)
FreqItemset(items=['Furniture Stores', 'Home & Garden', 'Shopping'], freq=397)
FreqItemset(items=['Barbers', 'Beauty & Spas'], freq=379)
FreqItemset(items=['Sandwiches', 'Food'], freq=379)
FreqItemset(items=['Sandwiches', 'Food', 'Restaurants'], freq=379)
```

```
FreqItemset(items=['Pubs', 'Bars', 'Nightlife', 'Restaurants'], freq=376)
FreqItemset(items=['Pubs', 'Bars', 'Restaurants'], freq=376)
FreqItemset(items=['Pubs', 'Nightlife', 'Restaurants'], freq=376)
FreqItemset(items=['Pubs', 'Restaurants'], freq=376)
FreqItemset(items=['Indian', 'Restaurants'], freq=376)
FreqItemset(items=['Pet Groomers', 'Pets'], freq=373)
FreqItemset(items=['Pet Groomers', 'Pet Services'], freq=373)
FreqItemset(items=['Pet Groomers', 'Pet Services', 'Pets'], freq=373)
FreqItemset(items=['Mexican', 'Fast Food'], freq=366)
FreqItemset(items=['Mexican', 'Fast Food', 'Restaurants'], freq=366)
FreqItemset(items=['Greek', 'Restaurants'], freq=363)
FreqItemset(items=['Banks & Credit Unions', 'Financial Services'], freq=355)
FreqItemset(items=['Trainers', 'Fitness & Instruction'], freq=353)
FreqItemset(items=['Trainers', 'Fitness & Instruction', 'Active Life'], freq=353)
FreqItemset(items=['Trainers', 'Active Life'], freq=353)
FreqItemset(items=['Parks', 'Active Life'], freq=352)
FreqItemset(items=['Sushi Bars', 'Japanese'], freq=345)
FreqItemset(items=['Sushi Bars', 'Japanese', 'Restaurants'], freq=345)
FreqItemset(items=["Men's Clothing", "Women's Clothing"], freq=343)
FreqItemset(items=["Men's Clothing", "Women's Clothing", 'Shopping'], freq=343)
FreqItemset(items=["Men's Clothing", "Women's Clothing", 'Fashion'], freq=343)
FreqItemset(items=["Men's Clothing", "Women's Clothing", 'Fashion', 'Shopping'], freq=34
3)
FreqItemset(items=['Music Venues', 'Nightlife'], freq=343)
FreqItemset(items=['Music Venues', 'Arts & Entertainment'], freq=343)
FreqItemset(items=['Music Venues', 'Arts & Entertainment', 'Nightlife'], freq=343)
FreqItemset(items=['Auto Parts & Supplies', 'Automotive'], freq=334)
FreqItemset(items=['Shoe Stores', 'Shopping'], freq=331)
FreqItemset(items=['Shoe Stores', 'Fashion'], freq=331)
FreqItemset(items=['Shoe Stores', 'Fashion', 'Shopping'], freq=331)
FreqItemset(items=['Dance Clubs', 'Nightlife'], freq=330)
FreqItemset(items=['Juice Bars & Smoothies', 'Food'], freq=325)
FreqItemset(items=['Optometrists', 'Health & Medical'], freq=324)
FreqItemset(items=['Accessories', "Women's Clothing"], freq=323)
FreqItemset(items=['Accessories', "Women's Clothing", 'Shopping'], freq=323)
FreqItemset(items=['Accessories', "Women's Clothing", 'Fashion'], freq=323)
FreqItemset(items=['Accessories', "Women's Clothing", 'Fashion', 'Shopping'], freq=323)
FreqItemset(items=['Car Wash', 'Automotive'], freq=323)
FreqItemset(items=['Oil Change Stations', 'Auto Repair'], freq=322)
FreqItemset(items=['Oil Change Stations', 'Auto Repair', 'Automotive'], freq=322)
FreqItemset(items=['Pet Stores', 'Pets'], freq=319)
FreqItemset(items=['Drugstores', 'Food'], freq=317)
FreqItemset(items=['Drugstores', 'Food', 'Shopping'], freq=317)
FreqItemset(items=['Home Decor', 'Shopping'], freq=304)
FreqItemset(items=['Home Decor', 'Home & Garden'], freq=304)
FreqItemset(items=['Home Decor', 'Home & Garden', 'Shopping'], freq=304)
FreqItemset(items=['Delis', 'Sandwiches'], freq=304)
FreqItemset(items=['Delis', 'Sandwiches', 'Restaurants'], freq=304)
FreqItemset(items=['Tires', 'Auto Repair'], freq=303)
FreqItemset(items=['Tires', 'Auto Repair', 'Automotive'], freq=303)
FreqItemset(items=['Diners', 'Restaurants'], freq=302)
FreqItemset(items=['Printing Services', 'Local Services'], freq=299)
FreqItemset(items=['Mobile Phones', 'Shopping'], freq=287)
FreqItemset(items=['Bakeries', 'Food', 'Restaurants'], freq=284)
FreqItemset(items=['Bakeries', 'Restaurants'], freq=284)
FreqItemset(items=['Grocery', 'Food', 'Shopping'], freq=284)
FreqItemset(items=['Grocery', 'Shopping'], freq=284)
FreqItemset(items=['Sports Bars', 'American (Traditional)'], freq=283)
FreqItemset(items=['Sports Bars', 'American (Traditional)', 'Bars'], freq=283)
FreqItemset(items=['Sports Bars', 'American (Traditional)', 'Bars', 'Nightlife'], freq=2
83)
FreqItemset(items=['Sports Bars', 'American (Traditional)', 'Bars', 'Nightlife', 'Restau
rants'], freq=283)
FreqItemset(items=['Sports Bars', 'American (Traditional)', 'Bars', 'Restaurants'], freq
=283)
```

```
FreqItemset(items=['Sports Bars', 'American (Traditional)', 'Nightlife'], freq=283)
FreqItemset(items=['Sports Bars', 'American (Traditional)', 'Nightlife', 'Restaurants'],
freq=283)
FreqItemset(items=['Sports Bars', 'American (Traditional)', 'Restaurants'], freq=283)
FreqItemset(items=['Vietnamese', 'Restaurants'], freq=283)
FreqItemset(items=['Health & Medical', 'Beauty & Spas'], freq=282)
FreqItemset(items=['Cafes', 'Food'], freq=281)
FreqItemset(items=['Cafes', 'Food', 'Restaurants'], freq=281)
FreqItemset(items=['Transportation', 'Hotels & Travel'], freq=271)
FreqItemset(items=['Electronics', 'Shopping'], freq=271)
FreqItemset(items=['Cosmetic Dentists', 'Dentists'], freq=265)
FreqItemset(items=['Cosmetic Dentists', 'Dentists', 'Health & Medical'], freq=265)
FreqItemset(items=['Cosmetic Dentists', 'Health & Medical'], freq=265)
FreqItemset(items=['Home Services', 'Shopping'], freq=257)
FreqItemset(items=['Pet Boarding/Pet Sitting', 'Pets'], freq=255)
FreqItemset(items=['Pet Boarding/Pet Sitting', 'Pet Services'], freq=255)
FreqItemset(items=['Pet Boarding/Pet Sitting', 'Pet Services', 'Pets'], freq=255)
FreqItemset(items=['Trainers', 'Gyms'], freq=251)
FreqItemset(items=['Trainers', 'Gyms', 'Fitness & Instruction'], freq=251)
FreqItemset(items=['Trainers', 'Gyms', 'Fitness & Instruction', 'Active Life'], freq=25
1)
FreqItemset(items=['Trainers', 'Gyms', 'Active Life'], freq=251)
FreqItemset(items=['Tex-Mex', 'Restaurants'], freq=250)
FreqItemset(items=['Local Services', 'Shopping'], freq=250)
FreqItemset(items=['Chiropractors', 'Health & Medical'], freq=249)
FreqItemset(items=['Breakfast & Brunch', 'Food'], freq=248)
FreqItemset(items=['Breakfast & Brunch', 'Food', 'Restaurants'], freq=248)
FreqItemset(items=['Arts & Entertainment', 'Shopping'], freq=245)
FreqItemset(items=['Vegetarian', 'Restaurants'], freq=244)
FreqItemset(items=['Donuts', 'Food'], freq=243)
FreqItemset(items=['Thrift Stores', 'Shopping'], freq=243)
FreqItemset(items=['Eyewear & Opticians', 'Shopping'], freq=241)
FreqItemset(items=['Wine Bars', 'Bars'], freq=236)
FreqItemset(items=['Wine Bars', 'Bars', 'Nightlife'], freq=236)
FreqItemset(items=['Wine Bars', 'Nightlife'], freq=236)
FreqItemset(items=['Burgers', 'American (Traditional)'], freq=236)
FreqItemset(items=['Burgers', 'American (Traditional)', 'Restaurants'], freq=236)
FreqItemset(items=['Heating & Air Conditioning/HVAC', 'Home Services'], freq=234)
FreqItemset(items=['Bookstores', 'Books, Mags, Music & Video'], freq=233)
FreqItemset(items=['Bookstores', 'Books, Mags, Music & Video', 'Shopping'], freq=233)
FreqItemset(items=['Bookstores', 'Shopping'], freq=233)
FreqItemset(items=['Florists', 'Flowers & Gifts'], freq=233)
FreqItemset(items=['Florists', 'Flowers & Gifts', 'Shopping'], freq=233)
FreqItemset(items=['Florists', 'Shopping'], freq=233)
FreqItemset(items=['Cosmetic Dentists', 'General Dentistry'], freq=230)
FreqItemset(items=['Cosmetic Dentists', 'General Dentistry', 'Dentists'], freq=230)
FreqItemset(items=['Cosmetic Dentists', 'General Dentistry', 'Dentists', 'Health & Medic
al'], freq=230)
FreqItemset(items=['Cosmetic Dentists', 'General Dentistry', 'Health & Medical'], freq=2
30)
FreqItemset(items=['Car Dealers', 'Automotive'], freq=230)
FreqItemset(items=['Caterers', 'Event Planning & Services'], freq=229)
FreqItemset(items=['Salad', 'Restaurants'], freq=228)
FreqItemset(items=['Medical Centers', 'Health & Medical'], freq=227)
FreqItemset(items=['Sandwiches', 'Pizza'], freq=226)
FreqItemset(items=['Sandwiches', 'Pizza', 'Restaurants'], freq=226)
FreqItemset(items=['Casinos', 'Arts & Entertainment'], freq=225)
FreqItemset(items=['Hot Dogs', 'Restaurants'], freq=223)
FreqItemset(items=['Used, Vintage & Consignment', 'Shopping'], freq=219)
FreqItemset(items=['Used, Vintage & Consignment', 'Fashion'], freq=219)
FreqItemset(items=['Used, Vintage & Consignment', 'Fashion', 'Shopping'], freq=219)
FreqItemset(items=['Shipping Centers', 'Local Services'], freq=216)
FreqItemset(items=['Golf', 'Active Life'], freq=215)
FreqItemset(items=['Specialty Schools', 'Education'], freq=212)
FreqItemset(items=['Hardware Stores', 'Shopping'], freq=211)
```

```
FreqItemset(items=['Hardware Stores', 'Home & Garden'], freq=211)
FreqItemset(items=['Hardware Stores', 'Home & Garden', 'Shopping'], freq=211)
FreqItemset(items=['Sporting Goods', 'Fashion'], freq=211)
FreqItemset(items=['Sporting Goods', 'Fashion', 'Shopping'], freq=211)
FreqItemset(items=['Tanning', 'Beauty & Spas'], freq=210)
FreqItemset(items=['Party & Event Planning', 'Event Planning & Services'], freq=210)
FreqItemset(items=['Middle Eastern', 'Restaurants'], freq=209)
FreqItemset(items=['Event Planning & Services', 'Shopping'], freq=208)
FreqItemset(items=['Health & Medical', 'Shopping'], freq=208)
FreqItemset(items=['Hair Stylists', 'Hair Salons'], freq=207)
FreqItemset(items=['Hair Stylists', 'Hair Salons', 'Beauty & Spas'], freq=207)
FreqItemset(items=['Hair Stylists', 'Beauty & Spas'], freq=207)
FreqItemset(items=['Yoga', 'Fitness & Instruction'], freq=206)
FreqItemset(items=['Yoga', 'Fitness & Instruction', 'Active Life'], freq=206)
FreqItemset(items=['Yoga', 'Active Life'], freq=206)
FreqItemset(items=['Event Planning & Services', 'Restaurants'], freq=205)
FreqItemset(items=['Tex-Mex', 'Mexican'], freq=205)
FreqItemset(items=['Tex-Mex', 'Mexican', 'Restaurants'], freq=205)
FreqItemset(items=['Art Galleries', 'Shopping'], freq=204)
FreqItemset(items=['Art Galleries', 'Arts & Entertainment'], freq=204)
FreqItemset(items=['Art Galleries', 'Arts & Entertainment', 'Shopping'], freq=204)
FreqItemset(items=['Dive Bars', 'Bars'], freq=203)
FreqItemset(items=['Dive Bars', 'Bars', 'Nightlife'], freq=203)
FreqItemset(items=['Dive Bars', 'Nightlife'], freq=203)
FreqItemset(items=['Contractors', 'Home Services'], freq=202)
FreqItemset(items=['Hobby Shops', 'Shopping'], freq=197)
FreqItemset(items=['Tobacco Shops', 'Shopping'], freq=195)
FreqItemset(items=['Convenience Stores', 'Food', 'Shopping'], freq=194)
FreqItemset(items=['Convenience Stores', 'Shopping'], freq=194)
FreqItemset(items=['Specialty Food', 'Food', 'Restaurants'], freq=194)
FreqItemset(items=['Specialty Food', 'Restaurants'], freq=194)
FreqItemset(items=['Delis', 'Food'], freq=192)
FreqItemset(items=['Delis', 'Food', 'Restaurants'], freq=192)
FreqItemset(items=['Chicken Wings', 'Pizza'], freq=191)
FreqItemset(items=['Chicken Wings', 'Pizza', 'Restaurants'], freq=191)
FreqItemset(items=['Ethnic Food', 'Specialty Food'], freq=190)
FreqItemset(items=['Ethnic Food', 'Specialty Food', 'Food'], freq=190)
FreqItemset(items=['Ethnic Food', 'Food'], freq=190)
FreqItemset(items=['Eyelash Service', 'Beauty & Spas'], freq=190)
FreqItemset(items=['Lounges', 'Bars', 'Nightlife', 'Restaurants'], freq=190)
FreqItemset(items=['Lounges', 'Bars', 'Restaurants'], freq=190)
FreqItemset(items=['Lounges', 'Nightlife', 'Restaurants'], freq=190)
FreqItemset(items=['Lounges', 'Restaurants'], freq=190)
FreqItemset(items=['Arts & Entertainment', 'Bars'], freq=190)
FreqItemset(items=['Arts & Entertainment', 'Bars', 'Nightlife'], freq=190)
FreqItemset(items=['Plumbing', 'Home Services'], freq=189)
FreqItemset(items=['Sports Wear', 'Sporting Goods'], freq=188)
FreqItemset(items=['Sports Wear', 'Sporting Goods', 'Shopping'], freq=188)
FreqItemset(items=['Sports Wear', 'Sporting Goods', 'Fashion'], freq=188)
FreqItemset(items=['Sports Wear', 'Sporting Goods', 'Fashion', 'Shopping'], freq=188)
FreqItemset(items=['Sports Wear', 'Shopping'], freq=188)
FreqItemset(items=['Sports Wear', 'Fashion'], freq=188)
FreqItemset(items=['Sports Wear', 'Fashion', 'Shopping'], freq=188)
FreqItemset(items=['Korean', 'Restaurants'], freq=186)
FreqItemset(items=['Bagels', 'Food'], freq=185)
FreqItemset(items=['Arts & Entertainment', 'Event Planning & Services'], freq=184)
FreqItemset(items=['Oil Change Stations', 'Tires'], freq=183)
FreqItemset(items=['Oil Change Stations', 'Tires', 'Automotive'], freq=183)
FreqItemset(items=['Canadian (New)', 'Restaurants'], freq=183)
FreqItemset(items=['Shipping Centers', 'Printing Services'], freq=181)
FreqItemset(items=['Shipping Centers', 'Printing Services', 'Local Services'], freq=181)
FreqItemset(items=['Oil Change Stations', 'Tires', 'Auto Repair'], freq=181)
FreqItemset(items=['Oil Change Stations', 'Tires', 'Auto Repair', 'Automotive'], freq=18
1)
FreqItemset(items=['Body Shops', 'Automotive'], freq=181)
```

```
FreqItemset(items=['Greek', 'Mediterranean'], freq=181)
FreqItemset(items=['Greek', 'Mediterranean', 'Restaurants'], freq=181)
FreqItemset(items=['Bikes', 'Sporting Goods'], freq=180)
FreqItemset(items=['Bikes', 'Sporting Goods', 'Shopping'], freq=180)
FreqItemset(items=['Bikes', 'Shopping'], freq=180)
FreqItemset(items=['Tours', 'Hotels & Travel'], freq=179)
FreqItemset(items=['Sewing & Alterations', 'Local Services'], freq=178)
FreqItemset(items=['Food Trucks', 'Food'], freq=178)
FreqItemset(items=['Cinema', 'Arts & Entertainment'], freq=178)
FreqItemset(items=['Nightlife', 'Food'], freq=177)
FreqItemset(items=['Drugstores', 'Convenience Stores'], freq=177)
FreqItemset(items=['Drugstores', 'Convenience Stores', 'Food'], freq=177)
FreqItemset(items=['Drugstores', 'Convenience Stores', 'Food', 'Shopping'], freq=177)
FreqItemset(items=['Drugstores', 'Convenience Stores', 'Shopping'], freq=177)
FreqItemset(items=['Arts & Entertainment', 'Restaurants'], freq=176)
FreqItemset(items=['Car Rental', 'Hotels & Travel'], freq=176)
FreqItemset(items=['Automotive', 'Food'], freq=175)
FreqItemset(items=['IT Services & Computer Repair', 'Local Services'], freq=174)
FreqItemset(items=['Breakfast & Brunch', 'American (Traditional)'], freq=174)
FreqItemset(items=['Breakfast & Brunch', 'American (Traditional)', 'Restaurants'], freq=
174)
FreqItemset(items=['Eyewear & Opticians', 'Health & Medical'], freq=173)
FreqItemset(items=['Eyewear & Opticians', 'Health & Medical', 'Shopping'], freq=173)
FreqItemset(items=['Optometrists', 'Shopping'], freq=171)
FreqItemset(items=['Optometrists', 'Health & Medical', 'Shopping'], freq=171)
FreqItemset(items=['Drugstores', 'Beauty & Spas'], freq=170)
FreqItemset(items=['Drugstores', 'Beauty & Spas', 'Shopping'], freq=170)
FreqItemset(items=['Beauty & Spas', 'Food'], freq=169)
FreqItemset(items=['Drugstores', 'Cosmetics & Beauty Supply'], freq=169)
FreqItemset(items=['Drugstores', 'Cosmetics & Beauty Supply', 'Shopping'], freq=169)
FreqItemset(items=['Drugstores', 'Cosmetics & Beauty Supply', 'Beauty & Spas'], freq=16
9)
FreqItemset(items=['Drugstores', 'Cosmetics & Beauty Supply', 'Beauty & Spas', 'Shoppin
g'], freq=169)
FreqItemset(items=['Wine Bars', 'Bars', 'Nightlife', 'Restaurants'], freq=167)
FreqItemset(items=['Wine Bars', 'Bars', 'Restaurants'], freq=167)
FreqItemset(items=['Wine Bars', 'Nightlife', 'Restaurants'], freq=167)
FreqItemset(items=['Wine Bars', 'Restaurants'], freq=167)
FreqItemset(items=['Eyewear & Opticians', 'Optometrists'], freq=167)
FreqItemset(items=['Eyewear & Opticians', 'Optometrists', 'Shopping'], freq=167)
FreqItemset(items=['Eyewear & Opticians', 'Optometrists', 'Health & Medical'], freq=167)
FreqItemset(items=['Eyewear & Opticians', 'Optometrists', 'Health & Medical', 'Shoppin
g'], freq=167)
FreqItemset(items=['Gas & Service Stations', 'Food'], freq=166)
FreqItemset(items=['Gas & Service Stations', 'Automotive', 'Food'], freq=166)
FreqItemset(items=['Photographers', 'Event Planning & Services'], freq=165)
FreqItemset(items=['Discount Store', 'Shopping'], freq=164)
FreqItemset(items=['Cosmetics & Beauty Supply', 'Food'], freq=163)
FreqItemset(items=['Cosmetics & Beauty Supply', 'Food', 'Shopping'], freq=163)
FreqItemset(items=['Cosmetics & Beauty Supply', 'Beauty & Spas', 'Food'], freq=163)
FreqItemset(items=['Cosmetics & Beauty Supply', 'Beauty & Spas', 'Food', 'Shopping'], fr
eq=163)
FreqItemset(items=['Beauty & Spas', 'Food', 'Shopping'], freq=163)
FreqItemset(items=['Antiques', 'Shopping'], freq=162)
FreqItemset(items=['Convenience Stores', 'Automotive'], freq=161)
FreqItemset(items=['Convenience Stores', 'Automotive', 'Food'], freq=161)
FreqItemset(items=['Home & Garden', 'Home Services'], freq=161)
FreqItemset(items=['Home & Garden', 'Home Services', 'Shopping'], freq=161)
FreqItemset(items=['Makeup Artists', 'Beauty & Spas'], freq=161)
FreqItemset(items=['Convenience Stores', 'Beauty & Spas'], freq=160)
FreqItemset(items=['Convenience Stores', 'Beauty & Spas', 'Food'], freq=160)
FreqItemset(items=['Convenience Stores', 'Beauty & Spas', 'Food', 'Shopping'], freq=160)
FreqItemset(items=['Convenience Stores', 'Beauty & Spas', 'Shopping'], freq=160)
FreqItemset(items=['Cafes', 'Coffee & Tea'], freq=160)
FreqItemset(items=['Cafes', 'Coffee & Tea', 'Food'], freq=160)
```

```
FreqItemset(items=['Cafes', 'Coffee & Tea', 'Food', 'Restaurants'], freq=160)
FreqItemset(items=['Cafes', 'Coffee & Tea', 'Restaurants'], freq=160)
FreqItemset(items=['Cosmetics & Beauty Supply', 'Convenience Stores'], freq=160)
FreqItemset(items=['Cosmetics & Beauty Supply', 'Convenience Stores', 'Food'], freq=160)
FreqItemset(items=['Cosmetics & Beauty Supply', 'Convenience Stores', 'Food', 'Shoppin
g'], freq=160)
FreqItemset(items=['Cosmetics & Beauty Supply', 'Convenience Stores', 'Shopping'], freq=
160)
FreqItemset(items=['Cosmetics & Beauty Supply', 'Convenience Stores', 'Beauty & Spas'],
freq=160)
FreqItemset(items=['Cosmetics & Beauty Supply', 'Convenience Stores', 'Beauty & Spas',
'Food'], freq=160)
FreqItemset(items=['Cosmetics & Beauty Supply', 'Convenience Stores', 'Beauty & Spas',
'Food', 'Shopping'], freq=160)
FreqItemset(items=['Cosmetics & Beauty Supply', 'Convenience Stores', 'Beauty & Spas',
'Shopping'], freq=160)
FreqItemset(items=['Drugstores', 'Convenience Stores', 'Beauty & Spas'], freq=160)
FreqItemset(items=['Drugstores', 'Convenience Stores', 'Beauty & Spas', 'Food'], freq=16
0)
FreqItemset(items=['Drugstores', 'Convenience Stores', 'Beauty & Spas', 'Food', 'Shoppin
g'], freq=160)
FreqItemset(items=['Drugstores', 'Convenience Stores', 'Beauty & Spas', 'Shopping'], fre
q=160)
FreqItemset(items=['Drugstores', 'Cosmetics & Beauty Supply', 'Convenience Stores'], fre
q=160)
FreqItemset(items=['Drugstores', 'Cosmetics & Beauty Supply', 'Convenience Stores', 'Foo
d'], freq=160)
FreqItemset(items=['Drugstores', 'Cosmetics & Beauty Supply', 'Convenience Stores', 'Foo
d', 'Shopping'], freq=160)
FreqItemset(items=['Drugstores', 'Cosmetics & Beauty Supply', 'Convenience Stores', 'Sho
pping'], freq=160)
FreqItemset(items=['Drugstores', 'Cosmetics & Beauty Supply', 'Convenience Stores', 'Bea
uty & Spas'], freq=160)
FreqItemset(items=['Drugstores', 'Cosmetics & Beauty Supply', 'Convenience Stores', 'Bea
uty & Spas', 'Food'], freq=160)
FreqItemset(items=['Drugstores', 'Cosmetics & Beauty Supply', 'Convenience Stores', 'Bea
uty & Spas', 'Food', 'Shopping'], freq=160)
FreqItemset(items=['Drugstores', 'Cosmetics & Beauty Supply', 'Convenience Stores', 'Bea
uty & Spas', 'Shopping'], freq=160)
FreqItemset(items=['Drugstores', 'Cosmetics & Beauty Supply', 'Food'], freq=160)
FreqItemset(items=['Drugstores', 'Cosmetics & Beauty Supply', 'Food', 'Shopping'], freq=
160)
FreqItemset(items=['Drugstores', 'Cosmetics & Beauty Supply', 'Beauty & Spas', 'Food'],
freq=160)
FreqItemset(items=['Drugstores', 'Cosmetics & Beauty Supply', 'Beauty & Spas', 'Food',
'Shopping'], freq=160)
FreqItemset(items=['Drugstores', 'Beauty & Spas', 'Food'], freq=160)
FreqItemset(items=['Drugstores', 'Beauty & Spas', 'Food', 'Shopping'], freq=160)
FreqItemset(items=['Health Markets', 'Specialty Food'], freq=160)
FreqItemset(items=['Health Markets', 'Specialty Food', 'Food'], freq=160)
FreqItemset(items=['Health Markets', 'Food'], freq=160)
FreqItemset(items=['Festivals', 'Arts & Entertainment'], freq=159)
FreqItemset(items=['Desserts', 'Bakeries'], freq=158)
FreqItemset(items=['Desserts', 'Bakeries', 'Food'], freq=158)
FreqItemset(items=['Gas & Service Stations', 'Convenience Stores'], freq=157)
FreqItemset(items=['Gas & Service Stations', 'Convenience Stores', 'Food'], freq=157)
FreqItemset(items=['Gas & Service Stations', 'Convenience Stores', 'Automotive'], freq=1
57)
FreqItemset(items=['Gas & Service Stations', 'Convenience Stores', 'Automotive', 'Foo
d'], freq=157)
FreqItemset(items=['Museums', 'Arts & Entertainment'], freq=157)
FreqItemset(items=['Nurseries & Gardening', 'Shopping'], freq=156)
FreqItemset(items=['Nurseries & Gardening', 'Home & Garden'], freq=156)
FreqItemset(items=['Nurseries & Gardening', 'Home & Garden', 'Shopping'], freq=156)
FreqItemset(items=['Gluten-Free', 'Restaurants'], freq=156)
```

```
FreqItemset(items=['Auto Glass Services', 'Automotive'], freq=155)
FreqItemset(items=['Home Cleaning', 'Home Services'], freq=155)
FreqItemset(items=['Shopping Centers', 'Shopping'], freq=154)
FreqItemset(items=['Pet Stores', 'Pet Services'], freq=153)
FreqItemset(items=['Pet Stores', 'Pet Services', 'Pets'], freq=153)
FreqItemset(items=['Massage Therapy', 'Health & Medical'], freq=152)
FreqItemset(items=['Family Practice', 'Doctors'], freq=152)
FreqItemset(items=['Family Practice', 'Doctors', 'Health & Medical'], freq=152)
FreqItemset(items=['Family Practice', 'Health & Medical'], freq=152)
FreqItemset(items=['Local Services', 'Home Services'], freq=152)
FreqItemset(items=['Latin American', 'Restaurants'], freq=151)
FreqItemset(items=['Professional Services', 'Home Services'], freq=151)
FreqItemset(items=['Toy Stores', 'Shopping'], freq=151)
FreqItemset(items=['Tattoo', 'Beauty & Spas'], freq=151)
FreqItemset(items=['Appliances', 'Shopping'], freq=150)
FreqItemset(items=['Appliances', 'Home & Garden'], freq=150)
FreqItemset(items=['Appliances', 'Home & Garden', 'Shopping'], freq=150)
FreqItemset(items=['Dance Clubs', 'Bars'], freq=149)
FreqItemset(items=['Dance Clubs', 'Bars', 'Nightlife'], freq=149)
FreqItemset(items=['Massage', 'Day Spas'], freq=149)
FreqItemset(items=['Massage', 'Day Spas', 'Beauty & Spas'], freq=149)
FreqItemset(items=['Tex-Mex', 'Fast Food'], freq=148)
FreqItemset(items=['Tex-Mex', 'Fast Food', 'Restaurants'], freq=148)
FreqItemset(items=['Tex-Mex', 'Mexican', 'Fast Food'], freq=146)
FreqItemset(items=['Tex-Mex', 'Mexican', 'Fast Food', 'Restaurants'], freq=146)
FreqItemset(items=['Nail Salons', 'Hair Salons'], freq=146)
FreqItemset(items=['Nail Salons', 'Hair Salons', 'Beauty & Spas'], freq=146)
FreqItemset(items=['Carpet Cleaning', 'Local Services'], freq=145)
FreqItemset(items=['Breakfast & Brunch', 'Sandwiches'], freq=145)
FreqItemset(items=['Breakfast & Brunch', 'Sandwiches', 'Restaurants'], freq=145)
FreqItemset(items=['Bars', 'Food'], freq=145)
FreqItemset(items=['Bars', 'Nightlife', 'Food'], freq=145)
FreqItemset(items=['British', 'Restaurants'], freq=143)
FreqItemset(items=['Professional Services', 'Local Services'], freq=141)
FreqItemset(items=['Landscaping', 'Home Services'], freq=140)
FreqItemset(items=['Urgent Care', 'Health & Medical'], freq=140)
FreqItemset(items=['Gastropubs', 'Restaurants'], freq=139)
FreqItemset(items=['Auto Detailing', 'Automotive'], freq=139)
FreqItemset(items=['Coffee & Tea', 'Sandwiches'], freq=138)
FreqItemset(items=['Coffee & Tea', 'Sandwiches', 'Food'], freq=138)
FreqItemset(items=['Coffee & Tea', 'Sandwiches', 'Food', 'Restaurants'], freq=138)
FreqItemset(items=['Coffee & Tea', 'Sandwiches', 'Restaurants'], freq=138)
FreqItemset(items=['Real Estate Services', 'Home Services'], freq=138)
FreqItemset(items=['Real Estate Services', 'Real Estate'], freq=138)
FreqItemset(items=['Real Estate Services', 'Real Estate', 'Home Services'], freq=138)
FreqItemset(items=['Drugstores', 'Grocery'], freq=137)
FreqItemset(items=['Drugstores', 'Grocery', 'Food'], freq=137)
FreqItemset(items=['Drugstores', 'Grocery', 'Food', 'Shopping'], freq=137)
FreqItemset(items=['Drugstores', 'Grocery', 'Shopping'], freq=137)
FreqItemset(items=['Hair Removal', 'Skin Care'], freq=137)
FreqItemset(items=['Hair Removal', 'Skin Care', 'Beauty & Spas'], freq=137)
FreqItemset(items=['Chocolatiers & Shops', 'Specialty Food'], freq=136)
FreqItemset(items=['Chocolatiers & Shops', 'Specialty Food', 'Food'], freq=136)
FreqItemset(items=['Chocolatiers & Shops', 'Food'], freq=136)
FreqItemset(items=['Ice Cream & Frozen Yogurt', 'Food', 'Restaurants'], freq=136)
FreqItemset(items=['Ice Cream & Frozen Yogurt', 'Restaurants'], freq=136)
FreqItemset(items=['Arts & Crafts', 'Event Planning & Services'], freq=135)
FreqItemset(items=['Arts & Crafts', 'Event Planning & Services', 'Shopping'], freq=135)
FreqItemset(items=['Southern', 'Restaurants'], freq=134)
FreqItemset(items=['Donuts', 'Coffee & Tea'], freq=133)
FreqItemset(items=['Donuts', 'Coffee & Tea', 'Food'], freq=133)
FreqItemset(items=['Breweries', 'Food'], freq=132)
FreqItemset(items=['Notaries', 'Local Services'], freq=132)
FreqItemset(items=['Sports Bars', 'American (New)'], freq=132)
FreqItemset(items=['Sports Bars', 'American (New)', 'Bars'], freq=132)
```

```
FreqItemset(items=['Sports Bars', 'American (New)', 'Bars', 'Nightlife'], freq=132)
FreqItemset(items=['Sports Bars', 'American (New)', 'Bars', 'Nightlife', 'Restaurants'],
freq=132)
FreqItemset(items=['Sports Bars', 'American (New)', 'Bars', 'Restaurants'], freq=132)
FreqItemset(items=['Sports Bars', 'American (New)', 'Nightlife'], freq=132)
FreqItemset(items=['Sports Bars', 'American (New)', 'Nightlife', 'Restaurants'], freq=13
2)
FreqItemset(items=['Sports Bars', 'American (New)', 'Restaurants'], freq=132)
FreqItemset(items=['Specialty Food', 'Grocery'], freq=132)
FreqItemset(items=['Specialty Food', 'Grocery', 'Food'], freq=132)
FreqItemset(items=['Medical Spas', 'Beauty & Spas'], freq=132)
FreqItemset(items=['Medical Spas', 'Health & Medical'], freq=132)
FreqItemset(items=['Medical Spas', 'Health & Medical', 'Beauty & Spas'], freq=132)
FreqItemset(items=['Movers', 'Home Services'], freq=131)
FreqItemset(items=['Farmers Market', 'Food'], freq=130)
FreqItemset(items=['Burgers', 'Nightlife'], freq=130)
FreqItemset(items=['Burgers', 'Nightlife', 'Restaurants'], freq=130)
FreqItemset(items=['Day Spas', 'Hair Salons'], freq=129)
FreqItemset(items=['Day Spas', 'Hair Salons', 'Beauty & Spas'], freq=129)
FreqItemset(items=['Pet Stores', 'Pet Groomers'], freq=129)
FreqItemset(items=['Pet Stores', 'Pet Groomers', 'Pets'], freq=129)
FreqItemset(items=['Pet Stores', 'Pet Groomers', 'Pet Services'], freq=129)
FreqItemset(items=['Pet Stores', 'Pet Groomers', 'Pet Services', 'Pets'], freq=129)
FreqItemset(items=['Obstetricians & Gynecologists', 'Doctors'], freq=129)
FreqItemset(items=['Obstetricians & Gynecologists', 'Doctors', 'Health & Medical'], freq
=129)
FreqItemset(items=['Obstetricians & Gynecologists', 'Health & Medical'], freq=129)
FreqItemset(items=['Music Venues', 'Bars'], freq=129)
FreqItemset(items=['Music Venues', 'Bars', 'Nightlife'], freq=129)
FreqItemset(items=['Music Venues', 'Arts & Entertainment', 'Bars'], freq=129)
FreqItemset(items=['Music Venues', 'Arts & Entertainment', 'Bars', 'Nightlife'], freq=12
9)
FreqItemset(items=['Active Life', 'Shopping'], freq=128)
FreqItemset(items=['Self Storage', 'Local Services'], freq=128)
FreqItemset(items=['Pet Training', 'Pets'], freq=128)
FreqItemset(items=['Pet Training', 'Pet Services'], freq=128)
FreqItemset(items=['Pet Training', 'Pet Services', 'Pets'], freq=128)
FreqItemset(items=['Flowers & Gifts', 'Event Planning & Services'], freq=128)
FreqItemset(items=['Flowers & Gifts', 'Event Planning & Services', 'Shopping'], freq=12
8)
FreqItemset(items=['Asian Fusion', 'Chinese'], freq=128)
FreqItemset(items=['Asian Fusion', 'Chinese', 'Restaurants'], freq=128)
FreqItemset(items=['Desserts', 'Ice Cream & Frozen Yogurt'], freq=127)
FreqItemset(items=['Desserts', 'Ice Cream & Frozen Yogurt', 'Food'], freq=127)
FreqItemset(items=['Auto Parts & Supplies', 'Auto Repair'], freq=127)
FreqItemset(items=['Auto Parts & Supplies', 'Auto Repair', 'Automotive'], freq=127)
FreqItemset(items=['Vape Shops', 'Shopping'], freq=126)
FreqItemset(items=['Vegan', 'Restaurants'], freq=126)
FreqItemset(items=['Cocktail Bars', 'Bars'], freq=126)
FreqItemset(items=['Cocktail Bars', 'Bars', 'Nightlife'], freq=126)
FreqItemset(items=['Cocktail Bars', 'Nightlife'], freq=126)
FreqItemset(items=['Burgers', 'Bars'], freq=125)
FreqItemset(items=['Burgers', 'Bars', 'Nightlife'], freq=125)
FreqItemset(items=['Burgers', 'Bars', 'Nightlife', 'Restaurants'], freq=125)
FreqItemset(items=['Burgers', 'Bars', 'Restaurants'], freq=125)
FreqItemset(items=['Desserts', 'Food', 'Restaurants'], freq=124)
FreqItemset(items=['Desserts', 'Restaurants'], freq=124)
FreqItemset(items=['Churches', 'Religious Organizations'], freq=123)
FreqItemset(items=['Hawaiian', 'Restaurants'], freq=122)
FreqItemset(items=["Children's Clothing", 'Shopping'], freq=122)
FreqItemset(items=["Children's Clothing", 'Fashion'], freq=122)
FreqItemset(items=["Children's Clothing", 'Fashion', 'Shopping'], freq=122)
FreqItemset(items=['Car Dealers', 'Auto Repair'], freq=121)
FreqItemset(items=['Car Dealers', 'Auto Repair', 'Automotive'], freq=121)
FreqItemset(items=['Salad', 'Sandwiches'], freq=121)
```

```
FreqItemset(items=['Salad', 'Sandwiches', 'Restaurants'], freq=121)
FreqItemset(items=['Notaries', 'Shipping Centers'], freq=120)
FreqItemset(items=['Notaries', 'Shipping Centers', 'Local Services'], freq=120)
FreqItemset(items=['German', 'Restaurants'], freq=120)
FreqItemset(items=["Men's Clothing", 'Accessories'], freq=120)
FreqItemset(items=["Men's Clothing", 'Accessories', 'Shopping'], freq=120)
FreqItemset(items=["Men's Clothing", 'Accessories', 'Fashion'], freq=120)
FreqItemset(items=["Men's Clothing", 'Accessories', 'Fashion', 'Shopping'], freq=120)
FreqItemset(items=['Kitchen & Bath', 'Shopping'], freq=119)
FreqItemset(items=['Kitchen & Bath', 'Home & Garden'], freq=119)
FreqItemset(items=['Kitchen & Bath', 'Home & Garden', 'Shopping'], freq=119)
FreqItemset(items=['Physical Therapy', 'Health & Medical'], freq=119)
FreqItemset(items=['Steakhouses', 'American (Traditional)'], freq=119)
FreqItemset(items=['Steakhouses', 'American (Traditional)', 'Restaurants'], freq=119)
FreqItemset(items=['Seafood', 'Steakhouses'], freq=118)
FreqItemset(items=['Seafood', 'Steakhouses', 'Restaurants'], freq=118)
FreqItemset(items=['Pest Control', 'Local Services'], freq=117)
FreqItemset(items=['Office Equipment', 'Shopping'], freq=116)
FreqItemset(items=['Hair Extensions', 'Hair Salons'], freq=116)
FreqItemset(items=['Hair Extensions', 'Hair Salons', 'Beauty & Spas'], freq=116)
FreqItemset(items=['Hair Extensions', 'Beauty & Spas'], freq=116)
FreqItemset(items=['Arts & Entertainment', 'Nightlife', 'Restaurants'], freq=115)
FreqItemset(items=['Bagels', 'Food', 'Restaurants'], freq=114)
FreqItemset(items=['Bagels', 'Restaurants'], freq=114)
FreqItemset(items=['Arts & Crafts', 'Flowers & Gifts'], freq=114)
FreqItemset(items=['Arts & Crafts', 'Flowers & Gifts', 'Shopping'], freq=114)
FreqItemset(items=['Music & DVDs', 'Books, Mags, Music & Video'], freq=114)
FreqItemset(items=['Music & DVDs', 'Books, Mags, Music & Video', 'Shopping'], freq=114)
FreqItemset(items=['Music & DVDs', 'Shopping'], freq=114)
FreqItemset(items=['Orthodontists', 'Dentists'], freq=113)
FreqItemset(items=['Orthodontists', 'Dentists', 'Health & Medical'], freq=113)
FreqItemset(items=['Orthodontists', 'Health & Medical'], freq=113)
FreqItemset(items=['Notaries', 'Printing Services'], freq=113)
FreqItemset(items=['Notaries', 'Printing Services', 'Local Services'], freq=113)
FreqItemset(items=['American (Traditional)', 'Fast Food'], freq=113)
FreqItemset(items=['American (Traditional)', 'Fast Food', 'Restaurants'], freq=113)
FreqItemset(items=['Venues & Event Spaces', 'Hotels & Travel'], freq=113)
FreqItemset(items=['Venues & Event Spaces', 'Hotels & Travel', 'Event Planning & Service
s'], freq=113)
FreqItemset(items=['Hair Removal', 'Nail Salons'], freq=113)
FreqItemset(items=['Hair Removal', 'Nail Salons', 'Beauty & Spas'], freq=113)
FreqItemset(items=['Notaries', 'Shipping Centers', 'Printing Services'], freq=112)
FreqItemset(items=['Notaries', 'Shipping Centers', 'Printing Services', 'Local Service
s'], freq=112)
FreqItemset(items=['Internet Service Providers', 'Home Services'], freq=112)
FreqItemset(items=['Internet Service Providers', 'Professional Services'], freq=112)
FreqItemset(items=['Internet Service Providers', 'Professional Services', 'Home Service
s'], freq=112)
FreqItemset(items=['Appliances & Repair', 'Local Services'], freq=111)
FreqItemset(items=['Caterers', 'Event Planning & Services', 'Restaurants'], freq=111)
FreqItemset(items=['Caterers', 'Restaurants'], freq=111)
FreqItemset(items=['Real Estate Agents', 'Home Services'], freq=110)
FreqItemset(items=['Real Estate Agents', 'Real Estate'], freq=110)
FreqItemset(items=['Real Estate Agents', 'Real Estate', 'Home Services'], freq=110)
FreqItemset(items=['Printing Services', 'Professional Services'], freq=110)
FreqItemset(items=['Printing Services', 'Professional Services', 'Local Services'], freq
=110)
FreqItemset(items=['Cards & Stationery', 'Arts & Crafts'], freq=109)
FreqItemset(items=['Cards & Stationery', 'Arts & Crafts', 'Event Planning & Services'],
freq=109)
FreqItemset(items=['Cards & Stationery', 'Arts & Crafts', 'Event Planning & Services',
'Shopping'], freq=109)
FreqItemset(items=['Cards & Stationery', 'Arts & Crafts', 'Flowers & Gifts'], freq=109)
FreqItemset(items=['Cards & Stationery', 'Arts & Crafts', 'Flowers & Gifts', 'Event Plan
ning & Services'], freq=109)
```

```
FreqItemset(items=['Cards & Stationery', 'Arts & Crafts', 'Flowers & Gifts', 'Event Plan
ning & Services', 'Shopping'], freq=109)
FreqItemset(items=['Cards & Stationery', 'Arts & Crafts', 'Flowers & Gifts', 'Shoppin
g'], freq=109)
FreqItemset(items=['Cards & Stationery', 'Arts & Crafts', 'Shopping'], freq=109)
FreqItemset(items=['Cards & Stationery', 'Event Planning & Services'], freq=109)
FreqItemset(items=['Cards & Stationery', 'Event Planning & Services', 'Shopping'], freq=
109)
FreqItemset(items=['Cards & Stationery', 'Flowers & Gifts'], freq=109)
FreqItemset(items=['Cards & Stationery', 'Flowers & Gifts', 'Event Planning & Service
s'], freq=109)
FreqItemset(items=['Cards & Stationery', 'Flowers & Gifts', 'Event Planning & Services',
'Shopping'], freq=109)
FreqItemset(items=['Cards & Stationery', 'Flowers & Gifts', 'Shopping'], freq=109)
FreqItemset(items=['Cards & Stationery', 'Shopping'], freq=109)
FreqItemset(items=["Men's Clothing", 'Accessories', "Women's Clothing"], freq=109)
FreqItemset(items=["Men's Clothing", 'Accessories', "Women's Clothing", 'Shopping'], fre
q=109)
FreqItemset(items=["Men's Clothing", 'Accessories', "Women's Clothing", 'Fashion'], freq
=109)
FreqItemset(items=["Men's Clothing", 'Accessories', "Women's Clothing", 'Fashion', 'Shop
ping'], freq=109)
FreqItemset(items=['Arts & Crafts', 'Flowers & Gifts', 'Event Planning & Services'], fre
q=109)
FreqItemset(items=['Arts & Crafts', 'Flowers & Gifts', 'Event Planning & Services', 'Sho
pping'], freq=109)
FreqItemset(items=['Skin Care', 'Day Spas'], freq=108)
FreqItemset(items=['Skin Care', 'Day Spas', 'Beauty & Spas'], freq=108)
FreqItemset(items=['Mortgage Brokers', 'Home Services'], freq=108)
FreqItemset(items=['Mortgage Brokers', 'Real Estate'], freq=108)
FreqItemset(items=['Mortgage Brokers', 'Real Estate', 'Home Services'], freq=108)
FreqItemset(items=['Skin Care', 'Massage'], freq=107)
FreqItemset(items=['Skin Care', 'Massage', 'Beauty & Spas'], freq=107)
FreqItemset(items=['Swimming Pools', 'Active Life'], freq=107)
FreqItemset(items=['Insurance', 'Financial Services'], freq=107)
FreqItemset(items=['Community Service/Non-Profit', 'Local Services'], freq=107)
FreqItemset(items=['Post Offices', 'Public Services & Government'], freq=107)
FreqItemset(items=['Fashion', 'Food'], freq=107)
FreqItemset(items=['Fashion', 'Food', 'Shopping'], freq=107)
FreqItemset(items=['Wedding Planning', 'Event Planning & Services'], freq=106)
FreqItemset(items=['Food Delivery Services', 'Food'], freq=106)
FreqItemset(items=['Buffets', 'Chinese'], freq=106)
FreqItemset(items=['Buffets', 'Chinese', 'Restaurants'], freq=106)
FreqItemset(items=['Arts & Entertainment', 'Hotels & Travel'], freq=106)
FreqItemset(items=['Property Management', 'Home Services'], freq=104)
FreqItemset(items=['Property Management', 'Real Estate'], freq=104)
FreqItemset(items=['Property Management', 'Real Estate', 'Home Services'], freq=104)
FreqItemset(items=['Meat Shops', 'Specialty Food'], freq=104)
FreqItemset(items=['Meat Shops', 'Specialty Food', 'Food'], freq=104)
FreqItemset(items=['Meat Shops', 'Food'], freq=104)
FreqItemset(items=['Adult Entertainment', 'Nightlife'], freq=104)
FreqItemset(items=['Venues & Event Spaces', 'Hotels'], freq=104)
FreqItemset(items=['Venues & Event Spaces', 'Hotels', 'Event Planning & Services'], freq
=104)
FreqItemset(items=['Venues & Event Spaces', 'Hotels', 'Hotels & Travel'], freq=104)
FreqItemset(items=['Venues & Event Spaces', 'Hotels', 'Hotels & Travel', 'Event Planning
& Services'], freq=104)
FreqItemset(items=['Massage', 'Health & Medical'], freq=103)
FreqItemset(items=['Massage', 'Health & Medical', 'Beauty & Spas'], freq=103)
FreqItemset(items=['Chicken Wings', 'Sandwiches'], freq=103)
FreqItemset(items=['Chicken Wings', 'Sandwiches', 'Restaurants'], freq=103)
FreqItemset(items=['Breakfast & Brunch', 'Coffee & Tea'], freq=103)
FreqItemset(items=['Breakfast & Brunch', 'Coffee & Tea', 'Food'], freq=103)
FreqItemset(items=['Breakfast & Brunch', 'Coffee & Tea', 'Food', 'Restaurants'], freq=10
3)
```

```
FreqItemset(items=['Breakfast & Brunch', 'Coffee & Tea', 'Restaurants'], freq=103)
FreqItemset(items=['Candy Stores', 'Specialty Food'], freq=103)
FreqItemset(items=['Candy Stores', 'Specialty Food', 'Food'], freq=103)
FreqItemset(items=['Candy Stores', 'Food'], freq=103)
FreqItemset(items=['Department Stores', 'Food'], freq=102)
FreqItemset(items=['Department Stores', 'Food', 'Shopping'], freq=102)
FreqItemset(items=['Department Stores', 'Fashion', 'Food'], freq=102)
FreqItemset(items=['Department Stores', 'Fashion', 'Food', 'Shopping'], freq=102)
FreqItemset(items=['Bakeries', 'Coffee & Tea'], freq=102)
FreqItemset(items=['Bakeries', 'Coffee & Tea', 'Food'], freq=102)
FreqItemset(items=['Sporting Goods', 'Active Life'], freq=102)
FreqItemset(items=['Sporting Goods', 'Active Life', 'Shopping'], freq=102)
FreqItemset(items=['Arts & Entertainment', 'Active Life'], freq=102)
FreqItemset(items=['Juice Bars & Smoothies', 'Food', 'Restaurants'], freq=101)
FreqItemset(items=['Juice Bars & Smoothies', 'Restaurants'], freq=101)
FreqItemset(items=['Department Stores', 'Grocery'], freq=101)
FreqItemset(items=['Department Stores', 'Grocery', 'Food'], freq=101)
FreqItemset(items=['Department Stores', 'Grocery', 'Food', 'Shopping'], freq=101)
FreqItemset(items=['Department Stores', 'Grocery', 'Shopping'], freq=101)
FreqItemset(items=['Department Stores', 'Grocery', 'Fashion'], freq=101)
FreqItemset(items=['Department Stores', 'Grocery', 'Fashion', 'Food'], freq=101)
FreqItemset(items=['Department Stores', 'Grocery', 'Fashion', 'Food', 'Shopping'], freq=
101)
FreqItemset(items=['Department Stores', 'Grocery', 'Fashion', 'Shopping'], freq=101)
FreqItemset(items=['Fast Food', 'Food'], freq=101)
FreqItemset(items=['Fast Food', 'Food', 'Restaurants'], freq=101)
FreqItemset(items=['Lawyers', 'Professional Services'], freq=101)
FreqItemset(items=['Libraries', 'Public Services & Government'], freq=101)
FreqItemset(items=['Grocery', 'Fashion'], freq=101)
FreqItemset(items=['Grocery', 'Fashion', 'Food'], freq=101)
FreqItemset(items=['Grocery', 'Fashion', 'Food', 'Shopping'], freq=101)
FreqItemset(items=['Grocery', 'Fashion', 'Shopping'], freq=101)
FreqItemset(items=['Day Spas', 'Nail Salons'], freq=100)
FreqItemset(items=['Day Spas', 'Nail Salons', 'Beauty & Spas'], freq=100)
FreqItemset(items=['Breakfast & Brunch', 'American (New)'], freq=100)
FreqItemset(items=['Breakfast & Brunch', 'American (New)', 'Restaurants'], freq=100)
FreqItemset(items=['Hotels', 'Arts & Entertainment'], freq=100)
FreqItemset(items=['Hotels', 'Arts & Entertainment', 'Event Planning & Services'], freq=
100)
FreqItemset(items=['Hotels', 'Arts & Entertainment', 'Hotels & Travel'], freq=100)
FreqItemset(items=['Hotels', 'Arts & Entertainment', 'Hotels & Travel', 'Event Planning
& Services'], freq=100)
FreqItemset(items=['Karaoke', 'Nightlife'], freq=100)
FreqItemset(items=['Arts & Entertainment', 'Hotels & Travel', 'Event Planning & Service
s'], freq=100)
FreqItemset(items=['Computers', 'Shopping'], freq=99)
FreqItemset(items=['American (Traditional)', 'Food'], freq=99)
FreqItemset(items=['American (Traditional)', 'Food', 'Restaurants'], freq=99)
FreqItemset(items=['Casinos', 'Event Planning & Services'], freq=99)
FreqItemset(items=['Casinos', 'Arts & Entertainment', 'Event Planning & Services'], freq
=99)
FreqItemset(items=['Fish & Chips', 'Restaurants'], freq=99)
FreqItemset(items=['Amusement Parks', 'Active Life'], freq=98)
FreqItemset(items=['Colleges & Universities', 'Education'], freq=98)
FreqItemset(items=['Hospitals', 'Health & Medical'], freq=98)
FreqItemset(items=['Martial Arts', 'Fitness & Instruction'], freq=98)
FreqItemset(items=['Martial Arts', 'Fitness & Instruction', 'Active Life'], freq=98)
FreqItemset(items=['Martial Arts', 'Active Life'], freq=98)
FreqItemset(items=['Home Decor', 'Furniture Stores'], freq=98)
FreqItemset(items=['Home Decor', 'Furniture Stores', 'Shopping'], freq=98)
FreqItemset(items=['Home Decor', 'Furniture Stores', 'Home & Garden'], freq=98)
FreqItemset(items=['Home Decor', 'Furniture Stores', 'Home & Garden', 'Shopping'], freq=
98)
FreqItemset(items=['Casinos', 'Hotels & Travel'], freq=98)
FreqItemset(items=['Casinos', 'Arts & Entertainment', 'Hotels & Travel'], freq=98)
```

```
FreqItemset(items=['Videos & Video Game Rental', 'Books, Mags, Music & Video'], freq=97)
FreqItemset(items=['Videos & Video Game Rental', 'Books, Mags, Music & Video', 'Shoppin
g'], freq=97)
FreqItemset(items=['Videos & Video Game Rental', 'Shopping'], freq=97)
FreqItemset(items=['Casinos', 'Hotels'], freq=97)
FreqItemset(items=['Casinos', 'Hotels', 'Event Planning & Services'], freq=97)
FreqItemset(items=['Casinos', 'Hotels', 'Hotels & Travel'], freq=97)
FreqItemset(items=['Casinos', 'Hotels', 'Hotels & Travel', 'Event Planning & Services'],
freq=97)
FreqItemset(items=['Casinos', 'Hotels', 'Arts & Entertainment'], freq=97)
FreqItemset(items=['Casinos', 'Hotels', 'Arts & Entertainment', 'Event Planning & Servic
es'], freq=97)
FreqItemset(items=['Casinos', 'Hotels', 'Arts & Entertainment', 'Hotels & Travel'], freq
=97)
FreqItemset(items=['Casinos', 'Hotels', 'Arts & Entertainment', 'Hotels & Travel', 'Even
t Planning & Services'], freq=97)
FreqItemset(items=['Casinos', 'Hotels & Travel', 'Event Planning & Services'], freq=97)
FreqItemset(items=['Casinos', 'Arts & Entertainment', 'Hotels & Travel', 'Event Planning
& Services'], freq=97)
FreqItemset(items=['Electronics', 'Mobile Phones'], freq=96)
FreqItemset(items=['Electronics', 'Mobile Phones', 'Shopping'], freq=96)
FreqItemset(items=['Bridal', 'Shopping'], freq=96)
FreqItemset(items=['Preschools', 'Education'], freq=96)
FreqItemset(items=['Baby Gear & Furniture', 'Shopping'], freq=95)
FreqItemset(items=['Event Planning & Services', 'Active Life'], freq=95)
FreqItemset(items=['Building Supplies', 'Home Services'], freq=95)
FreqItemset(items=['Delis', 'Fast Food'], freq=95)
FreqItemset(items=['Delis', 'Fast Food', 'Restaurants'], freq=95)
FreqItemset(items=['Jewelry', 'Fashion'], freq=95)
FreqItemset(items=['Jewelry', 'Fashion', 'Shopping'], freq=95)
FreqItemset(items=['Outdoor Gear', 'Sporting Goods'], freq=95)
FreqItemset(items=['Outdoor Gear', 'Sporting Goods', 'Shopping'], freq=95)
FreqItemset(items=['Outdoor Gear', 'Shopping'], freq=95)
FreqItemset(items=['Chicken Wings', 'American (Traditional)'], freq=94)
FreqItemset(items=['Chicken Wings', 'American (Traditional)', 'Restaurants'], freq=94)
FreqItemset(items=['Pediatricians', 'Doctors'], freq=94)
FreqItemset(items=['Pediatricians', 'Doctors', 'Health & Medical'], freq=94)
FreqItemset(items=['Pediatricians', 'Health & Medical'], freq=94)
FreqItemset(items=['American (New)', 'Food'], freq=94)
FreqItemset(items=['American (New)', 'Food', 'Restaurants'], freq=94)
FreqItemset(items=['Pet Boarding/Pet Sitting', 'Pet Groomers'], freq=94)
FreqItemset(items=['Pet Boarding/Pet Sitting', 'Pet Groomers', 'Pets'], freq=94)
FreqItemset(items=['Pet Boarding/Pet Sitting', 'Pet Groomers', 'Pet Services'], freq=94)
FreqItemset(items=['Pet Boarding/Pet Sitting', 'Pet Groomers', 'Pet Services', 'Pets'],
freq=94)
FreqItemset(items=['Ethnic Food', 'Specialty Food', 'Food', 'Restaurants'], freq=94)
FreqItemset(items=['Ethnic Food', 'Specialty Food', 'Restaurants'], freq=94)
FreqItemset(items=['Ethnic Food', 'Food', 'Restaurants'], freq=94)
FreqItemset(items=['Ethnic Food', 'Restaurants'], freq=94)
FreqItemset(items=['Tapas Bars', 'Restaurants'], freq=93)
FreqItemset(items=['Landmarks & Historical Buildings', 'Public Services & Government'],
freq=93)
FreqItemset(items=['Child Care & Day Care', 'Local Services'], freq=93)
FreqItemset(items=["Men's Hair Salons", 'Hair Salons'], freq=93)
FreqItemset(items=["Men's Hair Salons", 'Hair Salons', 'Beauty & Spas'], freq=93)
FreqItemset(items=["Men's Hair Salons", 'Beauty & Spas'], freq=93)
FreqItemset(items=['Gay Bars', 'Bars'], freq=93)
FreqItemset(items=['Gay Bars', 'Bars', 'Nightlife'], freq=93)
FreqItemset(items=['Gay Bars', 'Nightlife'], freq=93)
FreqItemset(items=['Pool Cleaners', 'Home Services'], freq=92)
FreqItemset(items=['Delis', 'Sandwiches', 'Fast Food'], freq=92)
FreqItemset(items=['Delis', 'Sandwiches', 'Fast Food', 'Restaurants'], freq=92)
FreqItemset(items=['Musical Instruments & Teachers', 'Shopping'], freq=92)
FreqItemset(items=['Gift Shops', 'Flowers & Gifts'], freq=92)
FreqItemset(items=['Gift Shops', 'Flowers & Gifts', 'Shopping'], freq=92)
```

```
FreqItemset(items=['Gift Shops', 'Shopping'], freq=92)
FreqItemset(items=['Dermatologists', 'Doctors'], freq=91)
FreqItemset(items=['Dermatologists', 'Doctors', 'Health & Medical'], freq=91)
FreqItemset(items=['Dermatologists', 'Health & Medical'], freq=91)
FreqItemset(items=['Mattresses', 'Shopping'], freq=91)
FreqItemset(items=['Mattresses', 'Home & Garden'], freq=91)
FreqItemset(items=['Mattresses', 'Home & Garden', 'Shopping'], freq=91)
FreqItemset(items=['Italian', 'Nightlife'], freq=90)
FreqItemset(items=['Italian', 'Nightlife', 'Restaurants'], freq=90)
FreqItemset(items=['Soup', 'Restaurants'], freq=90)
FreqItemset(items=['Auto Detailing', 'Car Wash'], freq=90)
FreqItemset(items=['Auto Detailing', 'Car Wash', 'Automotive'], freq=90)
FreqItemset(items=['Pubs', 'American (Traditional)'], freq=89)
FreqItemset(items=['Pubs', 'American (Traditional)', 'Bars'], freq=89)
FreqItemset(items=['Pubs', 'American (Traditional)', 'Bars', 'Nightlife'], freq=89)
FreqItemset(items=['Pubs', 'American (Traditional)', 'Bars', 'Nightlife', 'Restaurant
s'], freq=89)
FreqItemset(items=['Pubs', 'American (Traditional)', 'Bars', 'Restaurants'], freq=89)
FreqItemset(items=['Pubs', 'American (Traditional)', 'Nightlife'], freq=89)
FreqItemset(items=['Pubs', 'American (Traditional)', 'Nightlife', 'Restaurants'], freq=8
9)
FreqItemset(items=['Pubs', 'American (Traditional)', 'Restaurants'], freq=89)
FreqItemset(items=['Chicken Wings', 'Bars'], freq=89)
FreqItemset(items=['Chicken Wings', 'Bars', 'Nightlife'], freq=89)
FreqItemset(items=['Chicken Wings', 'Bars', 'Nightlife', 'Restaurants'], freq=89)
FreqItemset(items=['Chicken Wings', 'Bars', 'Restaurants'], freq=89)
FreqItemset(items=['Chicken Wings', 'Nightlife'], freq=89)
FreqItemset(items=['Chicken Wings', 'Nightlife', 'Restaurants'], freq=89)
FreqItemset(items=['Pediatric Dentists', 'Dentists'], freq=89)
FreqItemset(items=['Pediatric Dentists', 'Dentists', 'Health & Medical'], freq=89)
FreqItemset(items=['Pediatric Dentists', 'Health & Medical'], freq=89)
FreqItemset(items=['Pet Training', 'Pet Groomers'], freq=89)
FreqItemset(items=['Pet Training', 'Pet Groomers', 'Pets'], freq=89)
FreqItemset(items=['Pet Training', 'Pet Groomers', 'Pet Services'], freq=89)
FreqItemset(items=['Pet Training', 'Pet Groomers', 'Pet Services', 'Pets'], freq=89)
FreqItemset(items=['Event Planning & Services', 'Food'], freq=88)
FreqItemset(items=['Burgers', 'Sandwiches'], freq=88)
FreqItemset(items=['Burgers', 'Sandwiches', 'Restaurants'], freq=88)
FreqItemset(items=['Pakistani', 'Restaurants'], freq=88)
FreqItemset(items=['Chicken Wings', 'Sandwiches', 'Pizza'], freq=87)
FreqItemset(items=['Chicken Wings', 'Sandwiches', 'Pizza', 'Restaurants'], freq=87)
FreqItemset(items=['Pizza', 'Nightlife'], freq=87)
FreqItemset(items=['Pizza', 'Nightlife', 'Restaurants'], freq=87)
FreqItemset(items=['Ophthalmologists', 'Doctors'], freq=87)
FreqItemset(items=['Ophthalmologists', 'Doctors', 'Health & Medical'], freq=87)
FreqItemset(items=['Ophthalmologists', 'Health & Medical'], freq=87)
FreqItemset(items=['Italian', 'Bars'], freq=86)
FreqItemset(items=['Italian', 'Bars', 'Nightlife'], freq=86)
FreqItemset(items=['Italian', 'Bars', 'Nightlife', 'Restaurants'], freq=86)
FreqItemset(items=['Italian', 'Bars', 'Restaurants'], freq=86)
FreqItemset(items=['Pizza', 'Bars'], freq=86)
FreqItemset(items=['Pizza', 'Bars', 'Nightlife'], freq=86)
FreqItemset(items=['Pizza', 'Bars', 'Nightlife', 'Restaurants'], freq=86)
FreqItemset(items=['Pizza', 'Bars', 'Restaurants'], freq=86)
FreqItemset(items=['Tapas/Small Plates', 'Restaurants'], freq=86)
FreqItemset(items=['Auto Parts & Supplies', 'Tires'], freq=86)
FreqItemset(items=['Auto Parts & Supplies', 'Tires', 'Automotive'], freq=86)
FreqItemset(items=['Street Vendors', 'Food'], freq=86)
FreqItemset(items=['Mexican', 'Nightlife'], freq=85)
FreqItemset(items=['Mexican', 'Nightlife', 'Restaurants'], freq=85)
FreqItemset(items=['Art Supplies', 'Arts & Crafts'], freq=85)
FreqItemset(items=['Art Supplies', 'Arts & Crafts', 'Shopping'], freq=85)
FreqItemset(items=['Art Supplies', 'Shopping'], freq=85)
FreqItemset(items=['Pubs', 'American (New)'], freq=84)
FreqItemset(items=['Pubs', 'American (New)', 'Bars'], freq=84)
```

```
FreqItemset(items=['Pubs', 'American (New)', 'Bars', 'Nightlife'], freq=84)
FreqItemset(items=['Pubs', 'American (New)', 'Bars', 'Nightlife', 'Restaurants'], freq=8
4)
FreqItemset(items=['Pubs', 'American (New)', 'Bars', 'Restaurants'], freq=84)
FreqItemset(items=['Pubs', 'American (New)', 'Nightlife'], freq=84)
FreqItemset(items=['Pubs', 'American (New)', 'Nightlife', 'Restaurants'], freq=84)
FreqItemset(items=['Pubs', 'American (New)', 'Restaurants'], freq=84)
FreqItemset(items=['Lingerie', 'Shopping'], freq=84)
FreqItemset(items=['Lingerie', 'Fashion'], freq=84)
FreqItemset(items=['Lingerie', 'Fashion', 'Shopping'], freq=84)
FreqItemset(items=['Waxing', 'Hair Removal'], freq=84)
FreqItemset(items=['Waxing', 'Hair Removal', 'Beauty & Spas'], freq=84)
FreqItemset(items=['Waxing', 'Beauty & Spas'], freq=84)
FreqItemset(items=['Sports Bars', 'Pubs'], freq=84)
FreqItemset(items=['Sports Bars', 'Pubs', 'Bars'], freq=84)
FreqItemset(items=['Sports Bars', 'Pubs', 'Bars', 'Nightlife'], freq=84)
FreqItemset(items=['Sports Bars', 'Pubs', 'Nightlife'], freq=84)
FreqItemset(items=['Arts & Entertainment', 'Food'], freq=84)
FreqItemset(items=['Graphic Design', 'Professional Services'], freq=83)
FreqItemset(items=['Chicken Wings', 'Italian'], freq=82)
FreqItemset(items=['Chicken Wings', 'Italian', 'Pizza'], freq=82)
FreqItemset(items=['Chicken Wings', 'Italian', 'Pizza', 'Restaurants'], freq=82)
FreqItemset(items=['Chicken Wings', 'Italian', 'Restaurants'], freq=82)
FreqItemset(items=['Diners', 'Breakfast & Brunch'], freq=82)
FreqItemset(items=['Diners', 'Breakfast & Brunch', 'Restaurants'], freq=82)
FreqItemset(items=['Convenience Stores', 'Grocery'], freq=81)
FreqItemset(items=['Convenience Stores', 'Grocery', 'Food'], freq=81)
FreqItemset(items=['Desserts', 'Coffee & Tea'], freq=81)
FreqItemset(items=['Desserts', 'Coffee & Tea', 'Food'], freq=81)
FreqItemset(items=['Mexican', 'Bars'], freq=81)
FreqItemset(items=['Mexican', 'Bars', 'Nightlife'], freq=81)
FreqItemset(items=['Mexican', 'Bars', 'Nightlife', 'Restaurants'], freq=81)
FreqItemset(items=['Mexican', 'Bars', 'Restaurants'], freq=81)
FreqItemset(items=['Food Trucks', 'Food', 'Restaurants'], freq=81)
FreqItemset(items=['Food Trucks', 'Restaurants'], freq=81)
FreqItemset(items=['Sandwiches', 'American (Traditional)'], freq=81)
FreqItemset(items=['Sandwiches', 'American (Traditional)', 'Restaurants'], freq=81)
FreqItemset(items=['Veterinarians', 'Pet Services'], freq=80)
FreqItemset(items=['Veterinarians', 'Pet Services', 'Pets'], freq=80)
FreqItemset(items=['Steakhouses', 'Nightlife'], freq=80)
FreqItemset(items=['Steakhouses', 'Nightlife', 'Restaurants'], freq=80)
FreqItemset(items=['Pet Training', 'Pet Stores'], freq=80)
FreqItemset(items=['Pet Training', 'Pet Stores', 'Pets'], freq=80)
FreqItemset(items=['Pet Training', 'Pet Stores', 'Pet Services'], freq=80)
FreqItemset(items=['Pet Training', 'Pet Stores', 'Pet Services', 'Pets'], freq=80)
FreqItemset(items=['Medical Centers', 'Doctors'], freq=80)
FreqItemset(items=['Medical Centers', 'Doctors', 'Health & Medical'], freq=80)
FreqItemset(items=['Seafood', 'American (Traditional)'], freq=80)
FreqItemset(items=['Seafood', 'American (Traditional)', 'Restaurants'], freq=80)
FreqItemset(items=['Italian', 'Sandwiches'], freq=79)
FreqItemset(items=['Italian', 'Sandwiches', 'Restaurants'], freq=79)
FreqItemset(items=['Shoe Repair', 'Local Services'], freq=79)
FreqItemset(items=['Makeup Artists', 'Hair Salons'], freq=79)
FreqItemset(items=['Makeup Artists', 'Hair Salons', 'Beauty & Spas'], freq=79)
FreqItemset(items=['Nurseries & Gardening', 'Hardware Stores'], freq=78)
FreqItemset(items=['Nurseries & Gardening', 'Hardware Stores', 'Shopping'], freq=78)
FreqItemset(items=['Nurseries & Gardening', 'Hardware Stores', 'Home & Garden'], freq=7
8)
FreqItemset(items=['Nurseries & Gardening', 'Hardware Stores', 'Home & Garden', 'Shoppin
g'], freq=78)
FreqItemset(items=['Cafes', 'Breakfast & Brunch'], freq=78)
FreqItemset(items=['Cafes', 'Breakfast & Brunch', 'Restaurants'], freq=78)
FreqItemset(items=['Optometrists', 'Doctors'], freq=78)
FreqItemset(items=['Optometrists', 'Doctors', 'Health & Medical'], freq=78)
FreqItemset(items=['Taxis', 'Transportation'], freq=78)
```

```
FreqItemset(items=['Taxis', 'Transportation', 'Hotels & Travel'], freq=78)
FreqItemset(items=['Taxis', 'Hotels & Travel'], freq=78)
FreqItemset(items=['Soul Food', 'Restaurants'], freq=78)
FreqItemset(items=['Pakistani', 'Indian'], freq=78)
FreqItemset(items=['Pakistani', 'Indian', 'Restaurants'], freq=78)
FreqItemset(items=['Caribbean', 'Restaurants'], freq=78)
FreqItemset(items=['Weight Loss Centers', 'Health & Medical'], freq=78)
FreqItemset(items=['Asian Fusion', 'Sushi Bars'], freq=78)
FreqItemset(items=['Asian Fusion', 'Sushi Bars', 'Restaurants'], freq=78)
FreqItemset(items=['Motorcycle Dealers', 'Automotive'], freq=77)
FreqItemset(items=['Home & Garden', 'Fashion'], freq=77)
FreqItemset(items=['Home & Garden', 'Fashion', 'Shopping'], freq=77)
FreqItemset(items=['Cafes', 'Bakeries'], freq=77)
FreqItemset(items=['Cafes', 'Bakeries', 'Food'], freq=77)
FreqItemset(items=['Cafes', 'Bakeries', 'Food', 'Restaurants'], freq=77)
FreqItemset(items=['Cafes', 'Bakeries', 'Restaurants'], freq=77)
FreqItemset(items=['Event Planning & Services', 'Nightlife'], freq=77)
FreqItemset(items=['Pet Training', 'Pet Stores', 'Pet Groomers'], freq=77)
FreqItemset(items=['Pet Training', 'Pet Stores', 'Pet Groomers', 'Pets'], freq=77)
FreqItemset(items=['Pet Training', 'Pet Stores', 'Pet Groomers', 'Pet Services'], freq=7
7)
FreqItemset(items=['Pet Training', 'Pet Stores', 'Pet Groomers', 'Pet Services', 'Pet
s'], freq=77)
FreqItemset(items=['Beer, Wine & Spirits', 'Food', 'Restaurants'], freq=77)
FreqItemset(items=['Beer, Wine & Spirits', 'Restaurants'], freq=77)
FreqItemset(items=['Music Venues', 'Nightlife', 'Restaurants'], freq=77)
FreqItemset(items=['Music Venues', 'Arts & Entertainment', 'Nightlife', 'Restaurants'],
freq=77)
FreqItemset(items=['Music Venues', 'Arts & Entertainment', 'Restaurants'], freq=77)
FreqItemset(items=['Music Venues', 'Restaurants'], freq=77)
FreqItemset(items=['Irish', 'Restaurants'], freq=77)
FreqItemset(items=['Bowling', 'Active Life'], freq=77)
FreqItemset(items=['Wheel & Rim Repair', 'Automotive'], freq=76)
FreqItemset(items=['Chicken Wings', 'Sports Bars'], freq=76)
FreqItemset(items=['Chicken Wings', 'Sports Bars', 'Bars'], freq=76)
FreqItemset(items=['Chicken Wings', 'Sports Bars', 'Bars', 'Nightlife'], freq=76)
FreqItemset(items=['Chicken Wings', 'Sports Bars', 'Bars', 'Nightlife', 'Restaurants'],
freq=76)
FreqItemset(items=['Chicken Wings', 'Sports Bars', 'Bars', 'Restaurants'], freq=76)
FreqItemset(items=['Chicken Wings', 'Sports Bars', 'Nightlife'], freq=76)
FreqItemset(items=['Chicken Wings', 'Sports Bars', 'Nightlife', 'Restaurants'], freq=76)
FreqItemset(items=['Chicken Wings', 'Sports Bars', 'Restaurants'], freq=76)
FreqItemset(items=['Hiking', 'Active Life'], freq=76)
FreqItemset(items=['Airport Shuttles', 'Transportation'], freq=76)
FreqItemset(items=['Airport Shuttles', 'Transportation', 'Hotels & Travel'], freq=76)
FreqItemset(items=['Airport Shuttles', 'Hotels & Travel'], freq=76)
FreqItemset(items=['Body Shops', 'Auto Repair'], freq=76)
FreqItemset(items=['Body Shops', 'Auto Repair', 'Automotive'], freq=76)
FreqItemset(items=['Food Delivery Services', 'Food', 'Restaurants'], freq=75)
FreqItemset(items=['Food Delivery Services', 'Restaurants'], freq=75)
FreqItemset(items=['Watches', 'Shopping'], freq=75)
FreqItemset(items=['Flooring', 'Home Services'], freq=75)
FreqItemset(items=['Steakhouses', 'Bars'], freq=75)
FreqItemset(items=['Steakhouses', 'Bars', 'Nightlife'], freq=75)
FreqItemset(items=['Steakhouses', 'Bars', 'Nightlife', 'Restaurants'], freq=75)
FreqItemset(items=['Steakhouses', 'Bars', 'Restaurants'], freq=75)
FreqItemset(items=['Ophthalmologists', 'Optometrists'], freq=75)
FreqItemset(items=['Ophthalmologists', 'Optometrists', 'Doctors'], freq=75)
FreqItemset(items=['Ophthalmologists', 'Optometrists', 'Doctors', 'Health & Medical'], f
req=75)
FreqItemset(items=['Ophthalmologists', 'Optometrists', 'Health & Medical'], freq=75)
FreqItemset(items=['Department Stores', "Women's Clothing"], freq=74)
FreqItemset(items=['Department Stores', "Women's Clothing", 'Shopping'], freq=74)
FreqItemset(items=['Department Stores', "Women's Clothing", 'Fashion'], freq=74)
FreqItemset(items=['Department Stores', "Women's Clothing", 'Fashion', 'Shopping'], freq
```

```
=74)
FreqItemset(items=['Nightlife', 'Food', 'Restaurants'], freq=74)
FreqItemset(items=['Food Stands', 'Restaurants'], freq=74)
FreqItemset(items=['Professional Services', 'Shopping'], freq=74)
FreqItemset(items=['Tea Rooms', 'Food'], freq=74)
FreqItemset(items=['Grocery', 'Food', 'Restaurants'], freq=74)
FreqItemset(items=['Grocery', 'Restaurants'], freq=74)
FreqItemset(items=['Stadiums & Arenas', 'Arts & Entertainment'], freq=73)
FreqItemset(items=['Bagels', 'Breakfast & Brunch'], freq=73)
FreqItemset(items=['Bagels', 'Breakfast & Brunch', 'Food'], freq=73)
FreqItemset(items=['Bagels', 'Breakfast & Brunch', 'Food', 'Restaurants'], freq=73)
FreqItemset(items=['Bagels', 'Breakfast & Brunch', 'Restaurants'], freq=73)
FreqItemset(items=['Financial Services', 'Home Services'], freq=73)
FreqItemset(items=['Financial Services', 'Real Estate'], freq=73)
FreqItemset(items=['Financial Services', 'Real Estate', 'Home Services'], freq=73)
FreqItemset(items=['Diagnostic Services', 'Health & Medical'], freq=73)
FreqItemset(items=['Hookah Bars', 'Bars'], freq=73)
FreqItemset(items=['Hookah Bars', 'Bars', 'Nightlife'], freq=73)
FreqItemset(items=['Hookah Bars', 'Nightlife'], freq=73)
FreqItemset(items=['Active Life', 'Health & Medical'], freq=72)
FreqItemset(items=['Wheel & Rim Repair', 'Tires'], freq=72)
FreqItemset(items=['Wheel & Rim Repair', 'Tires', 'Automotive'], freq=72)
FreqItemset(items=['Comfort Food', 'Restaurants'], freq=72)
FreqItemset(items=['Electronics', 'Home Services'], freq=72)
FreqItemset(items=['Electronics', 'Home Services', 'Shopping'], freq=72)
FreqItemset(items=['Playgrounds', 'Active Life'], freq=72)
FreqItemset(items=['Cajun/Creole', 'Restaurants'], freq=72)
FreqItemset(items=['Pizza', 'Food'], freq=72)
FreqItemset(items=['Pizza', 'Food', 'Restaurants'], freq=72)
FreqItemset(items=['Eyelash Service', 'Hair Removal'], freq=72)
FreqItemset(items=['Eyelash Service', 'Hair Removal', 'Beauty & Spas'], freq=72)
FreqItemset(items=['Hair Removal', 'Hair Salons'], freq=72)
FreqItemset(items=['Hair Removal', 'Hair Salons', 'Beauty & Spas'], freq=72)
FreqItemset(items=['Outlet Stores', 'Shopping'], freq=71)
FreqItemset(items=['Mortgage Brokers', 'Financial Services'], freq=71)
FreqItemset(items=['Mortgage Brokers', 'Financial Services', 'Home Services'], freq=71)
FreqItemset(items=['Mortgage Brokers', 'Financial Services', 'Real Estate'], freq=71)
FreqItemset(items=['Mortgage Brokers', 'Financial Services', 'Real Estate', 'Home Servic
es'], freq=71)
FreqItemset(items=['IT Services & Computer Repair', 'Shopping'], freq=70)
FreqItemset(items=['IT Services & Computer Repair', 'Local Services', 'Shopping'], freq=
70)
FreqItemset(items=['Dance Studios', 'Fitness & Instruction'], freq=70)
FreqItemset(items=['Dance Studios', 'Fitness & Instruction', 'Active Life'], freq=70)
FreqItemset(items=['Dance Studios', 'Active Life'], freq=70)
FreqItemset(items=['Keys & Locksmiths', 'Home Services'], freq=70)
FreqItemset(items=['Interior Design', 'Home Services'], freq=70)
FreqItemset(items=['Limos', 'Transportation'], freq=70)
FreqItemset(items=['Limos', 'Transportation', 'Hotels & Travel'], freq=70)
FreqItemset(items=['Limos', 'Hotels & Travel'], freq=70)
FreqItemset(items=['Hot Dogs', 'Burgers'], freq=70)
FreqItemset(items=['Hot Dogs', 'Burgers', 'Restaurants'], freq=70)
FreqItemset(items=['Wholesale Stores', 'Shopping'], freq=69)
FreqItemset(items=['Elementary Schools', 'Education'], freq=69)
FreqItemset(items=['Spanish', 'Restaurants'], freq=69)
FreqItemset(items=['Fitness & Instruction', 'Active Life', 'Health & Medical'], freq=69)
FreqItemset(items=['Fitness & Instruction', 'Health & Medical'], freq=69)
FreqItemset(items=['Middle Eastern', 'Mediterranean'], freq=69)
FreqItemset(items=['Middle Eastern', 'Mediterranean', 'Restaurants'], freq=69)
FreqItemset(items=['Flowers & Gifts', 'Food'], freq=69)
FreqItemset(items=['Flowers & Gifts', 'Food', 'Shopping'], freq=69)
FreqItemset(items=['Electricians', 'Home Services'], freq=69)
FreqItemset(items=['American (New)', 'American (Traditional)'], freq=68)
FreqItemset(items=['American (New)', 'American (Traditional)', 'Restaurants'], freq=68)
FreqItemset(items=['Guns & Ammo', 'Shopping'], freq=68)
```

```
FreqItemset(items=['Gastropubs', 'Nightlife'], freq=68)
FreqItemset(items=['Gastropubs', 'Nightlife', 'Restaurants'], freq=68)
FreqItemset(items=['Smog Check Stations', 'Automotive'], freq=68)
FreqItemset(items=['Performing Arts', 'Nightlife'], freq=68)
FreqItemset(items=['Performing Arts', 'Arts & Entertainment', 'Nightlife'], freq=68)
FreqItemset(items=['Karaoke', 'Bars'], freq=68)
FreqItemset(items=['Karaoke', 'Bars', 'Nightlife'], freq=68)
FreqItemset(items=['Carpet Cleaning', 'Home Services'], freq=67)
FreqItemset(items=['Carpet Cleaning', 'Local Services', 'Home Services'], freq=67)
FreqItemset(items=['Chicken Wings', 'Fast Food'], freq=67)
FreqItemset(items=['Chicken Wings', 'Fast Food', 'Restaurants'], freq=67)
FreqItemset(items=['Skin Care', 'Nail Salons'], freq=67)
FreqItemset(items=['Skin Care', 'Nail Salons', 'Beauty & Spas'], freq=67)
FreqItemset(items=['Sports Bars', 'Pubs', 'Bars', 'Nightlife', 'Restaurants'], freq=67)
FreqItemset(items=['Sports Bars', 'Pubs', 'Bars', 'Restaurants'], freq=67)
FreqItemset(items=['Sports Bars', 'Pubs', 'Nightlife', 'Restaurants'], freq=67)
FreqItemset(items=['Sports Bars', 'Pubs', 'Restaurants'], freq=67)
FreqItemset(items=['Pilates', 'Fitness & Instruction'], freq=67)
FreqItemset(items=['Pilates', 'Fitness & Instruction', 'Active Life'], freq=67)
FreqItemset(items=['Pilates', 'Active Life'], freq=67)
FreqItemset(items=['Fabric Stores', 'Arts & Crafts'], freq=67)
FreqItemset(items=['Fabric Stores', 'Arts & Crafts', 'Shopping'], freq=67)
FreqItemset(items=['Fabric Stores', 'Shopping'], freq=67)
FreqItemset(items=['Electronics Repair', 'Local Services'], freq=67)
FreqItemset(items=['Acupuncture', 'Health & Medical'], freq=67)
FreqItemset(items=['Oral Surgeons', 'Dentists'], freq=66)
FreqItemset(items=['Oral Surgeons', 'Dentists', 'Health & Medical'], freq=66)
FreqItemset(items=['Oral Surgeons', 'Health & Medical'], freq=66)
FreqItemset(items=['Skin Care', 'Hair Salons'], freq=66)
FreqItemset(items=['Skin Care', 'Hair Salons', 'Beauty & Spas'], freq=66)
FreqItemset(items=['Bars', 'Food', 'Restaurants'], freq=66)
FreqItemset(items=['Bars', 'Nightlife', 'Food', 'Restaurants'], freq=66)
FreqItemset(items=['Diners', 'American (Traditional)'], freq=66)
FreqItemset(items=['Diners', 'American (Traditional)', 'Restaurants'], freq=66)
FreqItemset(items=['Shaved Ice', 'Food'], freq=66)
FreqItemset(items=['Resorts', 'Hotels & Travel'], freq=65)
FreqItemset(items=['Towing', 'Automotive'], freq=65)
FreqItemset(items=['Soup', 'Sandwiches'], freq=65)
FreqItemset(items=['Soup', 'Sandwiches', 'Restaurants'], freq=65)
FreqItemset(items=['Doctors', 'Beauty & Spas'], freq=65)
FreqItemset(items=['Doctors', 'Health & Medical', 'Beauty & Spas'], freq=65)
FreqItemset(items=['Sports Bars', 'Burgers'], freq=65)
FreqItemset(items=['Sports Bars', 'Burgers', 'Bars'], freq=65)
FreqItemset(items=['Sports Bars', 'Burgers', 'Bars', 'Nightlife'], freq=65)
FreqItemset(items=['Sports Bars', 'Burgers', 'Bars', 'Nightlife', 'Restaurants'], freq=6
5)
FreqItemset(items=['Sports Bars', 'Burgers', 'Bars', 'Restaurants'], freq=65)
FreqItemset(items=['Sports Bars', 'Burgers', 'Nightlife'], freq=65)
FreqItemset(items=['Sports Bars', 'Burgers', 'Nightlife', 'Restaurants'], freq=65)
FreqItemset(items=['Sports Bars', 'Burgers', 'Restaurants'], freq=65)
FreqItemset(items=['Gastropubs', 'Bars'], freq=65)
FreqItemset(items=['Gastropubs', 'Bars', 'Nightlife'], freq=65)
FreqItemset(items=['Gastropubs', 'Bars', 'Nightlife', 'Restaurants'], freq=65)
FreqItemset(items=['Gastropubs', 'Bars', 'Restaurants'], freq=65)
FreqItemset(items=['Banks & Credit Unions', 'Home Services'], freq=65)
FreqItemset(items=['Banks & Credit Unions', 'Financial Services', 'Home Services'], freq
=65)
FreqItemset(items=['Banks & Credit Unions', 'Financial Services', 'Real Estate'], freq=6
5)
FreqItemset(items=['Banks & Credit Unions', 'Financial Services', 'Real Estate', 'Home S
ervices'], freq=65)
FreqItemset(items=['Banks & Credit Unions', 'Real Estate'], freq=65)
FreqItemset(items=['Banks & Credit Unions', 'Real Estate', 'Home Services'], freq=65)
FreqItemset(items=['Caterers', 'Event Planning & Services', 'Food'], freq=65)
FreqItemset(items=['Caterers', 'Food'], freq=65)
```

```
FreqItemset(items=['Mortgage Brokers', 'Banks & Credit Unions'], freq=65)
FreqItemset(items=['Mortgage Brokers', 'Banks & Credit Unions', 'Home Services'], freq=6
5)
FreqItemset(items=['Mortgage Brokers', 'Banks & Credit Unions', 'Financial Services'], f
req=65)
FreqItemset(items=['Mortgage Brokers', 'Banks & Credit Unions', 'Financial Services', 'H
ome Services'], freq=65)
FreqItemset(items=['Mortgage Brokers', 'Banks & Credit Unions', 'Financial Services', 'R
eal Estate'], freq=65)
FreqItemset(items=['Mortgage Brokers', 'Banks & Credit Unions', 'Financial Services', 'R
eal Estate', 'Home Services'], freq=65)
FreqItemset(items=['Mortgage Brokers', 'Banks & Credit Unions', 'Real Estate'], freq=65)
FreqItemset(items=['Mortgage Brokers', 'Banks & Credit Unions', 'Real Estate', 'Home Ser
vices'], freq=65)
FreqItemset(items=['Beer, Wine & Spirits', 'Nightlife'], freq=65)
FreqItemset(items=['Beer, Wine & Spirits', 'Nightlife', 'Food'], freq=65)
FreqItemset(items=['Internet Service Providers', 'Home Services', 'Shopping'], freq=64)
FreqItemset(items=['Internet Service Providers', 'Professional Services', 'Home Service
s', 'Shopping'], freq=64)
FreqItemset(items=['Internet Service Providers', 'Professional Services', 'Shopping'], f
req=64)
FreqItemset(items=['Internet Service Providers', 'Shopping'], freq=64)
FreqItemset(items=['Skin Care', 'Health & Medical'], freq=64)
FreqItemset(items=['Skin Care', 'Health & Medical', 'Beauty & Spas'], freq=64)
FreqItemset(items=['Laser Hair Removal', 'Hair Removal'], freq=64)
FreqItemset(items=['Laser Hair Removal', 'Hair Removal', 'Beauty & Spas'], freq=64)
FreqItemset(items=['Laser Hair Removal', 'Beauty & Spas'], freq=64)
FreqItemset(items=['Comic Books', 'Books, Mags, Music & Video'], freq=64)
FreqItemset(items=['Comic Books', 'Books, Mags, Music & Video', 'Shopping'], freq=64)
FreqItemset(items=['Comic Books', 'Shopping'], freq=64)
FreqItemset(items=['Hobby Shops', 'Arts & Crafts'], freq=64)
FreqItemset(items=['Hobby Shops', 'Arts & Crafts', 'Shopping'], freq=64)
FreqItemset(items=['Windshield Installation & Repair', 'Automotive'], freq=64)
FreqItemset(items=['Turkish', 'Restaurants'], freq=64)
FreqItemset(items=['Professional Services', 'Home Services', 'Shopping'], freq=64)
FreqItemset(items=['Irish', 'Bars'], freq=64)
FreqItemset(items=['Irish', 'Bars', 'Nightlife'], freq=64)
FreqItemset(items=['Irish', 'Bars', 'Nightlife', 'Restaurants'], freq=64)
FreqItemset(items=['Irish', 'Bars', 'Restaurants'], freq=64)
FreqItemset(items=['Irish', 'Nightlife'], freq=64)
FreqItemset(items=['Irish', 'Nightlife', 'Restaurants'], freq=64)
FreqItemset(items=['Breakfast & Brunch', 'Sandwiches', 'Food'], freq=63)
FreqItemset(items=['Breakfast & Brunch', 'Sandwiches', 'Food', 'Restaurants'], freq=63)
FreqItemset(items=['Brasseries', 'Restaurants'], freq=63)
FreqItemset(items=['Bike Rentals', 'Active Life'], freq=63)
FreqItemset(items=['Session Photography', 'Event Planning & Services'], freq=63)
FreqItemset(items=['Session Photography', 'Photographers'], freq=63)
FreqItemset(items=['Session Photography', 'Photographers', 'Event Planning & Services'],
freq=63)
FreqItemset(items=['Arts & Entertainment', 'Bars', 'Nightlife', 'Restaurants'], freq=63)
FreqItemset(items=['Arts & Entertainment', 'Bars', 'Restaurants'], freq=63)
FreqItemset(items=['Signmaking', 'Professional Services'], freq=62)
FreqItemset(items=['Internet Service Providers', 'Electronics'], freq=62)
FreqItemset(items=['Internet Service Providers', 'Electronics', 'Home Services'], freq=6
2)
FreqItemset(items=['Internet Service Providers', 'Electronics', 'Home Services', 'Shoppi
ng'], freq=62)
FreqItemset(items=['Internet Service Providers', 'Electronics', 'Professional Service
s'], freq=62)
FreqItemset(items=['Internet Service Providers', 'Electronics', 'Professional Services',
'Home Services'], freq=62)
FreqItemset(items=['Internet Service Providers', 'Electronics', 'Professional Services',
'Home Services', 'Shopping'], freq=62)
FreqItemset(items=['Internet Service Providers', 'Electronics', 'Professional Services',
'Shopping'], freq=62)
```

```
FreqItemset(items=['Internet Service Providers', 'Electronics', 'Shopping'], freq=62)
FreqItemset(items=['Italian', 'Sandwiches', 'Pizza'], freq=62)
FreqItemset(items=['Italian', 'Sandwiches', 'Pizza', 'Restaurants'], freq=62)
FreqItemset(items=['Electronics', 'Professional Services'], freq=62)
FreqItemset(items=['Electronics', 'Professional Services', 'Home Services'], freq=62)
FreqItemset(items=['Electronics', 'Professional Services', 'Home Services', 'Shopping'],
freq=62)
FreqItemset(items=['Electronics', 'Professional Services', 'Shopping'], freq=62)
FreqItemset(items=['Hair Salons', 'Shopping'], freq=62)
FreqItemset(items=['Hair Salons', 'Beauty & Spas', 'Shopping'], freq=62)
FreqItemset(items=['Asian Fusion', 'Japanese'], freq=62)
FreqItemset(items=['Asian Fusion', 'Japanese', 'Restaurants'], freq=62)
```

Question3: Are all the itemsets obtained by setting minimum support 0.01 included in the itemsets obtained when we set the minimum support to 0.001?

Answer: Yes, all the itemsets obtaiend by setting minimum support 0.01 are included in the itemsets obtained by minimum support 0.001. The itemset obtained by minimum support 0.001 is very large compared to the first one (where minimum support is 0.01).

Part 7: Bonus Analysis (if any)

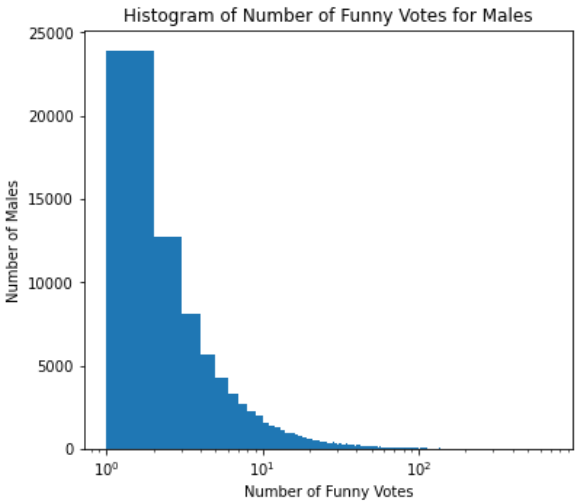
Here, you can include any additional and insightful exploratory data analysis or machine learning tasks you have carried out in addition to the guided exploration of the dataset above. Feel free to add code/markdown cells here to present your analysis.

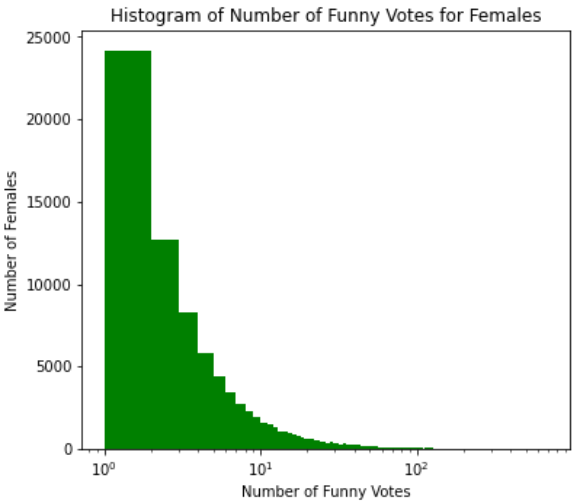
Analyzing the Funny & Useful Votes Given, by Gender

In [ ]:
```
funny_male=male_users.map(lambda x: x['votes']['funny'])
funny_female=female_users.map(lambda x: x['votes']['funny'])
```

In [ ]:
```
useful_male=male_users.map(lambda x: x['votes']['useful'])
useful_female=female_users.map(lambda x: x['votes']['useful'])
```

In [ ]:
```
plt.figure(figsize=(14,5))

plt.subplot(1, 2, 1)
plt.hist(funny_male.collect(),bins=range(1,700))
plt.title("Histogram of Number of Funny Votes for Males")
plt.xscale("log")
plt.xlabel('Number of Funny Votes')
plt.ylabel('Number of Males')

plt.subplot(1, 2, 2)
plt.hist(funny_female.collect(),bins=range(1,700),color='g')
plt.title("Histogram of Number of Funny Votes for Females")
plt.xscale("log")
plt.xlabel('Number of Funny Votes')
plt.ylabel('Number of Females')

plt.subplots_adjust(left=0.1,bottom=0.1, right=0.9, top=0.9, wspace=0.4,hspace=0.4)
```

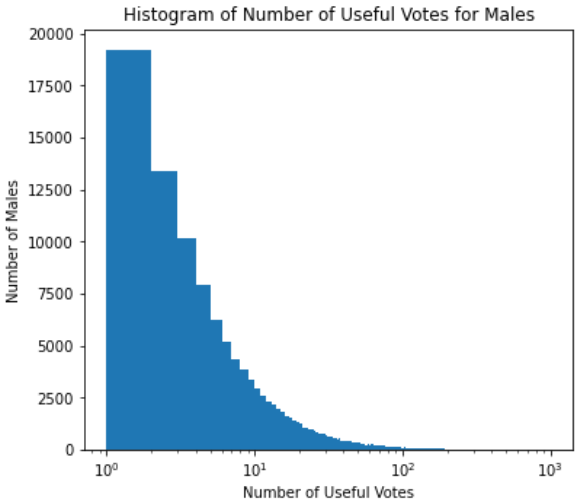


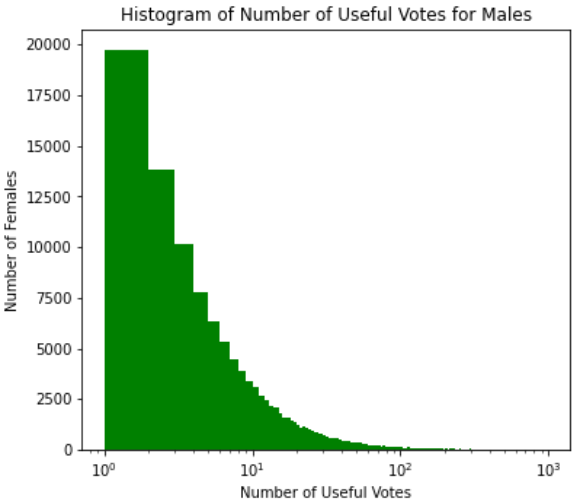


Both the plots (Number of Funny Votes for Men and Females) tend to follow the Power-Law relationship.

In [ ]:

```
plt.figure(figsize=(14,5))

plt.subplot(1, 2, 1)
plt.hist(useful_male.collect(),bins=range(1,1000))
plt.title("Histogram of Number of Useful Votes for Males")
plt.xscale("log")
plt.xlabel('Number of Useful Votes')
plt.ylabel('Number of Males')

plt.subplot(1, 2, 2)
plt.hist(useful_female.collect(),bins=range(1,1000),color='g')
plt.title("Histogram of Number of Useful Votes for Males")
plt.xscale("log")
plt.xlabel('Number of Useful Votes')
plt.ylabel('Number of Females')

plt.subplots_adjust(left=0.1,bottom=0.1, right=0.9, top=0.9, wspace=0.4,hspace=0.4)
```

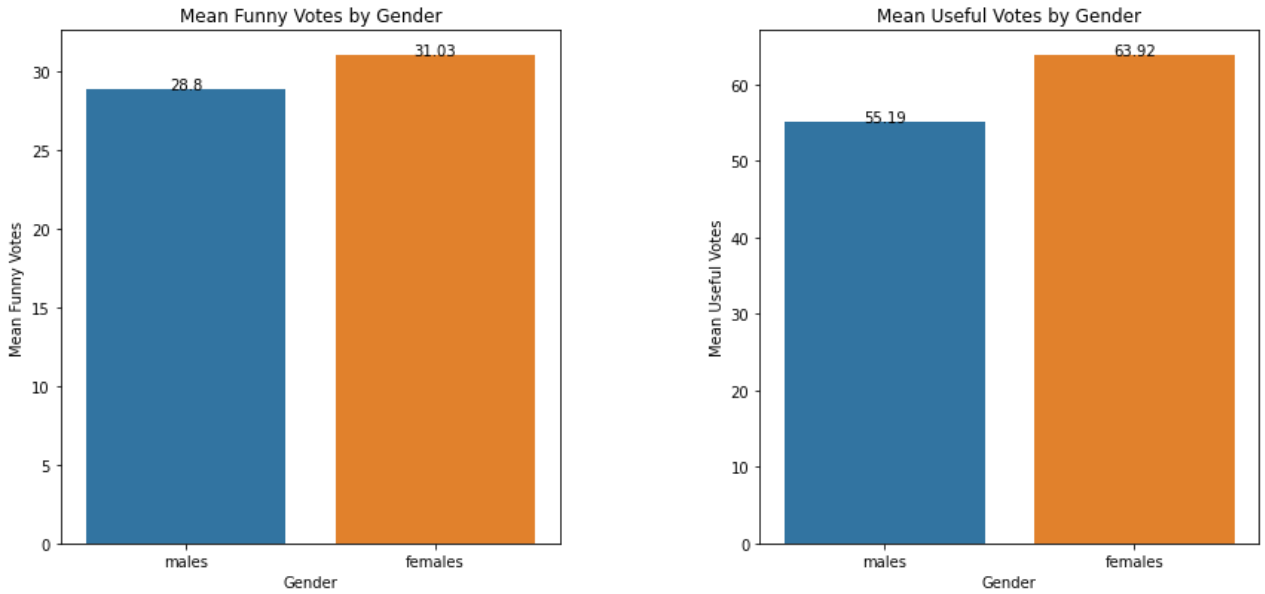




Both the plots (Number of Useful Votes for Men and Females) tend to follow the Power-Law relationship.

In [ ]:
```
males_funny_mean= funny_male.mean()
females_funny_mean= funny_female.mean()

males_useful_mean= useful_male.mean()
females_useful_mean= useful_female.mean()

d = {'Gender': ['males', 'females'], 'Mean Funny Votes': [males_funny_mean,females_funn
df = pd.DataFrame(data=d)
```

In [ ]:
```
plt.figure(figsize=(14,6))

plt.subplot(1, 2, 1)
g=sns.barplot(x='Gender',y='Mean Funny Votes',data=df)
plt.title("Mean Funny Votes by Gender")


for index, row in df.iterrows():
    g.text(row.name,row['Mean Funny Votes'], str(round(row['Mean Funny Votes'],2)), col


plt.subplot(1, 2, 2)
g=sns.barplot(x='Gender',y='Mean Useful Votes',data=df)
plt.title("Mean Useful Votes by Gender")

for index, row in df.iterrows():
    g.text(row.name,row['Mean Useful Votes'], str(round(row['Mean Useful Votes'],2)), c


plt.subplots_adjust(left=0.1,bottom=0.1, right=0.9, top=0.9, wspace=0.4,hspace=0.4)
```



Hence, we can say that females tend to increase the credibility of reviews on Yelp, which can help other users be more trustful of certain reviews while selecting amongst various restaurants, hotels, cafes etc.

More reviews that are voted as "useful" also reinforces the authenticity of a review, and enhances the general popularity of a brand/restaurant on Yelp.