import time

from datetime import datetime

import pandas as pd

import matplotlib.pyplot as plt

#Задание 1

#Функция, возвращающая регион, если перечисленные города есть в keyword

def region(data):

for i in geo\_data:

for j in geo\_data[i]:

if j in data:

return i

return 'undefined'

data=pd.read\_csv('C:/Users/Katya/Desktop/keywords.csv')

geo\_data = {

'Центр': ['москва', 'тула', 'ярославль'],

'Северо-Запад': ['петербург', 'псков', 'мурманск'],

'Дальний Восток': ['владивосток', 'сахалин', 'хабаровск']}

#разделение keyword на слова через пробел

data['keyword']=data.apply(lambda x: x['keyword'].split(' '), axis=1)

#вызов функции опредения региона для кажой строки в keyword

data['region']=data['keyword'].apply(lambda x: region(x))

#Задание 2

#Функция, возвращающая класс фильма по среднему рейтингу

def get\_class(rating):

if rating<=2: return 'Низкий рейтинг'

if 2<rating<=4: return 'Средний рейтинг'

if rating==4.5 or rating==5 : return 'Высокий ретинг'

else: return 'undefined'

rating=pd.read\_csv('C:/Users/Katya/Desktop/ratings.csv')

#Группировка по фильмам с передачей в функцию get\_class среднего рейтинга

rating.groupby('movieId').mean()['rating'].apply(lambda x: get\_class(x))

#Задание 3

rating=pd.read\_csv('C:/Users/Katya/Desktop/ratings.csv')

#Перевод timespamp в формат datetime

rating['timestamp'] = rating['timestamp'].apply(lambda x: datetime.fromtimestamp(x))

#Подсчет максимального и минимального времени и количества проставленных рейтингов в группировке по юзерам

rating\_grouped=rating.groupby('userId').agg({'timestamp': ['min', 'max'], 'rating': 'count'}).head()

#Селекция киноманов: пользователи, проставившие 100 и более оценок

rating\_grouped=rating\_grouped[rating\_grouped['rating']['count']>=100]

#Подсчет Lifetime для каждого пользователя

rating\_grouped['diff'] = rating\_grouped['timestamp']['max'] - rating\_grouped['timestamp']['min']

#Перевод Lifetime в дни

rating\_grouped['diff']=rating\_grouped['diff'].apply(lambda x: str(x).split()[0])

#Задание 4

years=[x+1 for x in range(1950,2010)]

def production\_year(data):

d=datetime.fromtimestamp(data)

if d.year in years: return d.year

else: return 1900

rating['year']=rating['timestamp'].apply(lambda x: production\_year(x))

#Можно ли столбец с рейтингом при группировке как-нибудь назвать, чтобы потом к нему обратиться?

rating.groupby('year').mean().sort\_values('rating',ascending=False)['rating']

#%matplotlib inline

#plt.plot(rating['year'], rating['rating'])