

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**Федеральное государственное автономное образовательное учреждение высшего**  
**образования**  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ**  
**«МИФИ»**

---

**ИНСТИТУТ ФИНАНСОВОЙ И ЭКОНОМИЧЕСКОЙ БЕЗОПАСНОСТИ**  
**КАФЕДРА ФИНАНСОВОГО МОНИТОРИНГА**

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №4**  
**НА ТЕМУ:**  
**ИСПОЛЬЗОВАНИЕ БОЛЬШИХ ДАННЫХ В МАРКЕТИНГОВЫХ ИССЛЕДОВАНИЯХ**  
**ПО КУРСУ «РИАС»**

Выполнил: Нуритдинходжаева А. (С20-702)

Преподаватель: Прохоров И.В.

Оценка \_\_\_\_\_ Дата \_\_\_\_\_

Москва, 2024

**Цель:** освоить использование больших данных в маркетинговых исследованиях, используя в языке Python API ВКонтакте.

### Задание 1

Используя метод ВКонтакте `groups.getMembers`, вернуть информацию о пользователях (а именно пол, дата рождения, город), которые являются членами сообщества <https://vk.com/buh.mephi>.

Для выполнения данной лабораторной работы будем использовать `access.token`, полученный в лабораторной работе №3: `vk1.a.x3wkE8A5yrPoMDRoXtpA7gcCvHh-ybVaPk_YgNQx9VaJF5sA1cWtPqDg3AtTaAwtuDRas1kvizHskNjXtJLpjFXAALdB3dyL7ezkJHm2YmsfIoL0Q6J08Mw-__McGCT5XLSIB5pXnhICkgivUf_XT-dGzbPwRSdpADvVmbiN2cox6p0TpGsXh3QMbcgY1HWPhncJC3Yv1g1hYU3-fPL1CA`

По заданию лабораторной работы была написана программа:

```
pip install vk
pip install vk_api
import time
import requests
import pandas as pd
import vk
import vk_api
import datetime
from collections import Counter

vk_session = vk_api.VkApi(token="vk1.a.x3wkE8A5yrPoMDRoXtpA7gcCvHh-
ybVaPk_YgNQx9VaJF5sA1cWtPqDg3AtTaAwtuDRas1kvizHskNjXtJLpjFXAALdB3dyL7ezkJH
m2YmsfIoL0Q6J08Mw-__McGCT5XLSIB5pXnhICkgivUf_XT-
dGzbPwRSdpADvVmbiN2cox6p0TpGsXh3QMbcgY1HWPhncJC3Yv1g1hYU3-fPL1CA")
api = vk_session.get_api()

lst_of_users = []

offset = 0
group_id = "buh.mephi"
while True:
    response = api.groups.getMembers(group_id=group_id, offset=offset,
fields=["bdate", "sex", "city"])
    time.sleep(0.1)
    lst_of_users.extend(response["items"])
    offset += len(response["items"])
    if offset >= response["count"]:
        break

print("Количество участников в группе:", len(lst_of_users))
```

По результату работы программы был получен следующий результат о количестве пользователей в группе:

---

Количество участников в группе: 2305

## Задание 2

С помощью Python подсчитать число пользователей по полу (мужского, женского), возрасту (12 – 18, 18 – 30, 30 – 45), городу (Москва). Отобрать показатели, по которым наибольшее количество пользователей.

По заданию лабораторной работы была написана программа:

```
def validate_birth(birth):
    try:
        return int(birth.split(".")[2])
    except IndexError:
        return 0

def determine_sex(sex):
    if sex == 1:
        return "w"
    else:
        return "m"

def check_age(age):
    if 12 <= age <= 18:
        return "12-18"
    elif 18 <= age <= 30:
        return "18-30"
    elif 30 <= age <= 45:
        return "30-45"
    else:
        return "other age"

groups = {}
lst_of_ages = []
lst_of_cities = []
lst_of_sex = []
for i in lst_of_users:
    if ("bdate" in i) and (validate_birth(i["bdate"]) != 0) and ("sex" in i)
    and ("city" in i):
        age = check_age(datetime.datetime.now().year -
validate_birth(i["bdate"]))
        city = i["city"]["title"]
        sex = determine_sex(i["sex"])
        key = (age, sex, city)
        if key in groups:
            groups[key] += 1
        else:
            groups[key] = 1
groups
groups_counts = dict(Counter(groups).most_common(15))
df = pd.DataFrame.from_dict(groups_counts, orient = 'index')
df.plot(kind = 'bar', figsize = (20, 10))
```

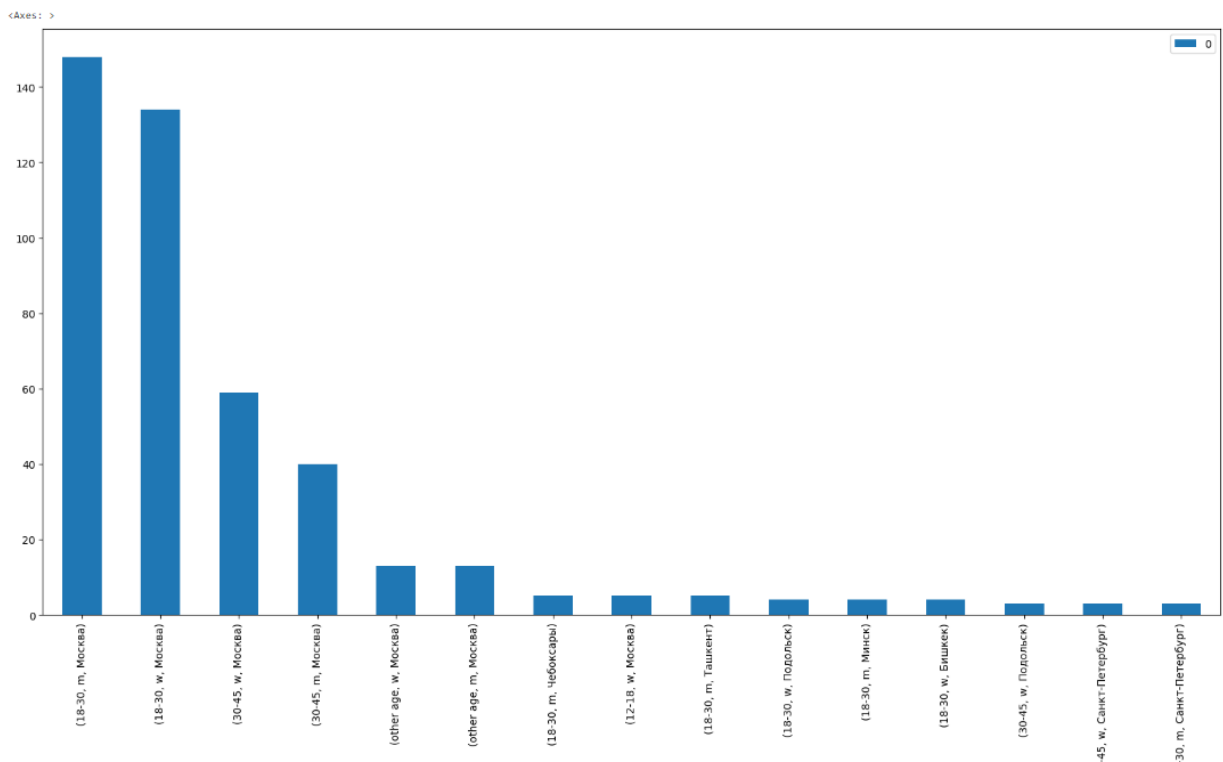
По результату работы программы был получен следующий результат:

{('30-45', 'm', 'Москва'): 40,	('other age', 'm', 'Кривой Рог'): 1,
('30-45', 'w', 'Москва'): 59,	('18-30', 'w', 'Тольятти'): 1,
('other age', 'w', 'Москва'): 13,	('18-30', 'w', 'Монино'): 1,
('other age', 'm', 'Москва'): 13,	('18-30', 'm', 'Пенза'): 2,
('18-30', 'w', 'Москва'): 134,	('other age', 'w', 'Коломна'): 1,
('18-30', 'w', 'Красноярск'): 2,	('18-30', 'm', 'Кашира'): 1,
('18-30', 'm', 'Москва'): 148,	('30-45', 'm', 'Чехов'): 1,
('30-45', 'w', 'Подольск'): 3,	('30-45', 'm', 'Ростов-на-Дону'): 1,
('30-45', 'w', 'Сочи'): 1,	('30-45', 'w', 'Анапа'): 1,
('30-45', 'm', 'Краснодар'): 2,	('18-30', 'm', 'Мытищи'): 2,
('30-45', 'w', 'Обнинск'): 1,	('18-30', 'm', 'Тула'): 1,
('18-30', 'm', 'Саранск'): 1,	('18-30', 'm', 'Астана'): 1,
('30-45', 'm', 'Ангарск'): 1,	('30-45', 'm', 'Ташкент'): 1,
('30-45', 'w', 'Серпухов'): 1,	('other age', 'm', 'Семей'): 1,
('18-30', 'w', 'Подольск'): 4,	('other age', 'w', 'Люберцы'): 1,
('30-45', 'w', 'Seattle'): 1,	('18-30', 'w', 'Владикавказ'): 1,
('18-30', 'm', 'Домодедово'): 1,	('18-30', 'm', 'Элиста'): 1,
('other age', 'w', 'Вологда'): 1,	('30-45', 'w', 'Воронеж'): 1,
('30-45', 'w', 'Сергиев Посад'): 1,	('other age', 'm', 'Актобе'): 1,
('30-45', 'w', 'Домодедово'): 1,	('18-30', 'm', 'Ашдоод'): 1,
('30-45', 'm', 'Praha'): 1,	('18-30', 'w', 'Людиново'): 1,
('30-45', 'w', 'Солнечногорск'): 1,	('18-30', 'm', 'Душанбе'): 2,
('18-30', 'w', 'Домодедово'): 2,	('18-30', 'w', 'Тамбов'): 1,
('30-45', 'w', 'Санкт-Петербург'): 3,	('18-30', 'w', 'Минеральные Воды'): 1,
('30-45', 'm', 'Воронеж'): 1,	('18-30', 'm', 'Рославль'): 1,
('18-30', 'm', 'Пичаево'): 1,	('18-30', 'm', 'Подольск'): 1,
('other age', 'w', 'Фурманов'): 1,	('12-18', 'w', 'Москва'): 5,
('18-30', 'm', 'Волжский'): 2,	('18-30', 'w', 'Краснодар'): 1,
('18-30', 'm', 'Санкт-Петербург'): 3,	('18-30', 'w', 'Бишкек'): 4,
('30-45', 'w', 'Новороссийск'): 1,	('18-30', 'm', 'Сергиев Посад'): 1,
('18-30', 'm', 'Хабаровск'): 1,	('18-30', 'm', 'Видное'): 1,
('other age', 'w', 'Нижегород'): 1,	('30-45', 'w', 'Алматы'): 1,
('18-30', 'm', 'Смоленск'): 1,	('18-30', 'm', 'Самара'): 1,
('12-18', 'w', 'Калуга'): 1,	('18-30', 'm', 'Ванадзор'): 1,
('other age', 'w', 'Калуга'): 1,	('12-18', 'm', 'Москва'): 2,
('18-30', 'm', 'Саров'): 1,	('18-30', 'm', 'Тбилиси'): 1,
('30-45', 'w', 'Киров'): 1,	('18-30', 'm', 'Одинцово'): 1,
('18-30', 'm', 'Электросталь'): 1,	('18-30', 'm', 'Калуга'): 1,
('18-30', 'm', 'Минск'): 4,	('18-30', 'm', 'Нефтекамск'): 1,
('other age', 'w', 'Limerick'): 1,	('18-30', 'w', 'Смоленск'): 1,
('18-30', 'w', 'Волгоград'): 2,	('18-30', 'w', 'Орёл'): 2,
('30-45', 'm', 'Саров'): 1,	('18-30', 'm', 'Караганда'): 1,
('30-45', 'w', 'Ликино-Дулево'): 1,	('18-30', 'w', 'Саяногорск'): 1,
('18-30', 'm', 'Воронеж'): 1,	('other age', 'w', 'Красногорск'): 1,
('18-30', 'w', 'Саров'): 1,	('18-30', 'm', 'Токуо'): 1,
('18-30', 'm', 'Северодвинск'): 1,	('other age', 'w', 'Челябинск'): 1,
('18-30', 'm', 'Казань'): 1,	('18-30', 'm', 'Владимир'): 2,
('30-45', 'w', 'Пенза'): 1,	('12-18', 'm', 'Сургут'): 1,
('18-30', 'w', 'Ташкент'): 1,	('18-30', 'm', 'Ташкент'): 5,
('18-30', 'm', 'Чебоксары'): 5,	('other age', 'm', 'Beograd'): 1,
('18-30', 'w', 'Тула'): 1,	('18-30', 'w', 'Хабаровск'): 1,
('18-30', 'w', 'Брянск'): 2,	('18-30', 'w', 'Старые Дороги'): 1,
('30-45', 'm', 'Los Angeles'): 1,	('other age', 'm', 'New York City'): 1,
('30-45', 'w', 'Нижегород'): 1,	('other age', 'w', 'Подольск'): 1,
('30-45', 'm', 'Chicago'): 1,	('18-30', 'm', 'Los Angeles'): 1,
('18-30', 'm', 'Дубна'): 1,	('other age', 'w', 'Ливны'): 1,
('other age', 'm', 'Бишкек'): 2,	('18-30', 'w', 'Худжанд'): 1,

('other age', 'w', 'Электросталь'): 1,  
 ('18-30', 'm', 'Киев'): 1,  
 ('30-45', 'w', 'Саранск'): 1,  
 ('18-30', 'm', 'Рязань'): 1,  
 ('18-30', 'w', 'Beograd'): 1,  
 ('30-45', 'w', 'Новомосковск'): 1,  
 ('18-30', 'w', 'Жуковский'): 1,  
 ('other age', 'w', 'Винница'): 1,  
 ('18-30', 'm', 'Елец'): 1,  
 ('30-45', 'w', 'Наро-Фоминск'): 1,  
 ('30-45', 'w', 'Верхний Уфалей'): 1,  
 ('18-30', 'w', 'Юрьев-Польский'): 1,  
 ('18-30', 'm', 'Иваново'): 1,  
 ('30-45', 'w', 'Гагарин'): 1,  
 ('other age', 'm', 'Ульяновск'): 1,  
 ('18-30', 'm', 'Брест'): 1,  
 ('18-30', 'w', 'Гагарин'): 1,  
 ('other age', 'w', 'Los Angeles'): 1,

('30-45', 'w', 'Балаково'): 1,  
 ('30-45', 'm', 'Челябинск'): 1,  
 ('30-45', 'm', 'Владивосток'): 1,  
 ('18-30', 'm', 'Хива'): 1,  
 ('18-30', 'm', 'Sidmouth'): 1,  
 ('other age', 'w', 'Новороссийск'): 1,  
 ('18-30', 'w', 'Королёв'): 1,  
 ('18-30', 'm', 'Бишкек'): 1,  
 ('other age', 'w', 'Белгород'): 1,  
 ('12-18', 'w', 'Тамбов'): 1,  
 ('other age', 'w', 'Снежинск'): 1,  
 ('18-30', 'w', 'Чехов'): 1,  
 ('18-30', 'm', 'Карши'): 1,  
 ('other age', 'm', 'Пенза'): 1,  
 ('18-30', 'm', 'Ulaanbaatar'): 1,  
 ('30-45', 'm', 'Alger'): 1,  
 ('other age', 'w', 'Кривандино'): 1}

В том числе получена гистограмма, по которой отобраны показатели с наибольшим количеством пользователей.



Москва, m, 18-30 - 148  
 Москва, w, 18-30 - 134  
 Москва, w, 30-45 - 59

### Задание 3

Создать рекламное объявление ВКонтакте по отобранным показателям и посмотреть размер аудитории.

Создано рекламное объявление ВКонтакте по отобранным показателям и получен размер аудитории:

Целевая аудитория ?

⚠ 81 941 000 чел.

#### Задание 4

##### Рассчитать по формулам:

Объём рекламной кампании = цена клика \* размер аудитории.

Прогнозируемый объём продаж = 1000 \* размер аудитории \* 0,05

Если Объём рекламной кампании > Прогнозируемый объём продаж, то открытие пиццерии не имеет смысла

Если Объём рекламной кампании < Прогнозируемый объём продаж, то открытие пиццерии имеет смысл

Для расчета написана программа:

```
audience = 81941000
click_price = 13

coverage = audience * click_price
volume_of_sales = 1000 * audience * 0.05
if coverage > volume_of_sales:
    print("Рекламная кампания не имеет смысла")
else:
    print("Рекламная кампания имеет смысл")
```

Получен следующий результат:

Рекламная кампания имеет смысл

## **Заключение**

По итогам проведенной лабораторной работы были успешно освоены методы работы с большими данными, использованы сервисы API ВКонтакте и рекламы ВКонтакте.

При работе с API ВКонтакте были реализованы запросы для получения данных о поле, возрасте и городе проживания участников одного из сообществ. По полученным данным были отобраны самые многочисленные категории участников, которые впоследствии использовались для настроек рекламной кампании ВКонтакте. По данным кампании были определены рамки, в которых она имеет смысл.