

Возможности программного интерфейса API при проведении исследований по наличию вакансий в городах России в источнике открытых данных *hh.ru*

А.М. Патрусова^a, А.Н. Бильданов^b

Братский государственный университет, ул. Макаренко, 40, Братск, Россия

^a patrusova@mail.ru, ^b fossadoe@yandex.ru

^a <https://orcid.org/0000-0001-5433-9614>, ^b <https://orcid.org/0009-0004-7081-0145>

Статья поступила 25.11.2024, принята 12.12.2024

В статье представлены возможности программного интерфейса API как инструмента для получения данных о вакансиях на сайте hh.ru, что направлено на принятие решений по результатам поиска вакансий в источнике открытых данных hh.ru. Актуальность данной статьи обусловлена необходимостью изучения данных о вакансиях, представленных в источниках открытых данных, с целью автоматизации выборки данных в разрезе характеристик рынка труда для выявления различных факторов, характеризующих состояние рынка труда отдельного города или региона России. Формирование программного кода на языке Python позволило осуществить несколько этапов работы с данными: парсинг страниц на поиск вакансии «водитель»; изучение наименования столбцов полученного набора данных; вывод выбранных признаков; получение массива данных с минимальными значениями зарплат и списком городов с исследуемой вакансией; построение графика распределения зарплат водителей по городам России. Выполнение отдельных аспектов представлено на основе использования встроенных библиотек Python. Исследование направлено на анализ вакансий в городах России с применением протокола API, формирование алгоритма подготовки данных, полученных в результате запроса, к проведению дальнейшей аналитики. Рассмотрены способы получения открытых данных с применением протокола API – получение данных в формате csv. Представлены возможности использования инструментов открытия файлов формата CSV, полученных из открытых источников – библиотеки Requests, Pandas, NumPy языка программирования Python. Продемонстрирована визуализация обработанных данных в виде диаграммы, для построения которой используется библиотека Matplotlib, позволяющая визуализировать большие массивы данных. Статья демонстрирует использование различных библиотек в одном исследовании для получения конкретного результата.

Ключевые слова: данные; открытые данные; анализ данных; вакансии, протокол API.

The capabilities of the API software interface for conducting research on the availability of vacancies in Russian cities in the open data source *hh.ru*

A.M. Patrusova^a, A.N. Bildanov^b

Bratsk State University; 40, Makarenko St., Bratsk, Russia

^a patrusova@mail.ru, ^b fossadoe@yandex.ru

^a <https://orcid.org/0000-0001-5433-9614>, ^b <https://orcid.org/0009-0004-7081-0145>

Received 25.11.2024, accepted 12.12.2024

The article presents the capabilities of the API programming interface as a tool for obtaining data on vacancies on the site hh.ru, which is aimed at making decisions based on the results of a job search in an open data source hh.ru. The relevance of this article is due to the need to study data on vacancies presented in open data sources in order to automate data sampling in the context of market characteristics to identify various factors characterizing the state of the labor market of a particular city or region of Russia. The formation of the program code in Python makes it possible to carry out several stages of working with data: parsing pages to search for a "driver" vacancy; examine the names of columns of the resulting dataset; display selected features; obtain an array of data with minimum wage values and a list of cities with the vacancy under study; plot the distribution of driver salaries across Russian cities. The implementation of certain aspects is presented based on the use of built-in Python libraries. The research is aimed at analyzing vacancies in Russian cities using the API protocol, forming an algorithm for preparing the data obtained as a result of the request for further analysis. The methods of obtaining open data are considered: using the API protocol – obtaining data in csv format. The possibilities of using tools for opening CSV files obtained from open sources are presented: Requests, Pandas, NumPy libraries of the Python programming language.

The visualization of the processed data in the form of a diagram is demonstrated, for the construction of which the Matplotlib library is used, which allows one to visualize large amounts of data. The article demonstrates the use of different libraries in one study to obtain a specific result.

Keywords: data; open data; data analysis; vacancies, API protocol.

Введение. Открытые данные как инструмент для изучения, анализа и выработки управленческих решений представляет широкие возможности при использовании современных методов получения и обработки данных [1–4]. Размещение вакансий работодателями в открытых источниках данных – наиболее популярный ресурс для граждан, ищущих или планирующих сменить работу или место жительства. При этом основной целью поиска информации является изучение вакансий по интересующим профессиям и их отличительным характеристикам в зависимости от региона, города.

Вместе с этим частью ресурса открытых источников информации является размещение резюме, что представляет особый интерес для работодателей в целях реализации процесса найма персонала. Сегодня активно развиваются автоматизированные системы рекрутинга с применением технологий искусственного интеллекта [5].

Вакансии и резюме позволяют оценить возможности и дефицит трудовых ресурсов на рынке труда по статистическим данным, например, количество вакансий и частота их публикаций, уровень зарплаты, а также другие параметры. В работе [6] авторы предлагают разработанную модель, позволяющую проводить мониторинг и управление на рынке труда. В работе [7] на основе данных открытых источников формируется отчет о необходимых профессиональных компетенциях в зависимости от вида деятельности. В работе [8] разработана технология анализа актуальных вакансий, размещенных в источнике *hh.ru*. В статье [9] разработан телеграм-бот для автоматизации парсинга вакансий на сайте *hh.ru*.

Интерес российских ученых подтверждает высокую актуальность разработки инструментария автоматизированного анализа вакансий рынка труда в городах и регионах России [10–16].

Применение API для получения данных. Рассмотрим особенности применения программного интерфейса API источника открытых данных *hh.ru*, который предоставляет ряд возможностей для работы с данными:

- *вакансии*: получение списка вакансий по различным фильтрам (регион, зарплата, опыт работы и т. д.);

- *резюме*: можно запросить резюме по ключевым словам и региону;

- *статистика*: доступ к данным о рынке труда, например, средней зарплате по профессиям.

Рассмотрим, какие данные можно получить через интерфейс API и какие параметры доступны для фильтрации.

API *hh.ru* – открытый ресурс, не требующий регистрации для получения данных. URL-адрес источника вакансий: <https://api.hh.ru/vacancies>.

С помощью API *hh.ru* можно получить огромный массив данных о различных вакансиях. Работать с набором данных, представленных в «сыром» виде, неудобно, поэтому требуется дополнительная обработка данных.

Для проведения анализа данных использован высокоуровневый язык программирования *Python*. На начальном этапе требуется импортировать необходимые библиотеки, позволяющие формировать запросы (*Requests*) и работать с наборами структурированных данных (*Pandas*). Далее необходимо указать количество страниц (например 10) для поиска вакансий и название самих вакансий (водитель). Названия вакансий можно указывать через запятую на любом языке, так как будут искаяться совпадения (рис. 1).

```
import requests
import pandas as pd

# Тут указать количество страниц сайта hh.ru для
number_of_pages = 10

# Тут указать название профессии(или нескольких)
job_title = ["Водитель"]
```

Рис. 1. Загрузка библиотек *Requests* и *Pandas*, указание количества страниц на вакансию и наименование вакансии

Обработка данных. На следующем этапе происходит парсинг указанных страниц на поиск нужных вакансий (рис. 2). В качестве параметров указывается следующее:

- *text* – наименование профессии(-ий);
- *area* – регион поиска (113 – Россия);
- *per_page* – количество вакансий на страницу;
- *page* – номер страницы.

```
for job in job_title:
    data=[]
    for i in range(number_of_pages):
        # API сайта hh.ru
        url = 'https://api.hh.ru/vacancies'
        par = {'text': job, 'area':'113', 'per_page':'10', 'page':i}
        r = requests.get(url, params=par)
        e=r.json()
        data.append(e)
        vacancy_details = data[0]['items'][0].keys()
        df = pd.DataFrame(columns= list(vacancy_details))
        ind = 0
        for i in range(len(data)):
            for j in range(len(data[i]['items'])):
                df.loc[ind] = data[i]['items'][j]
                ind+=1
```

Рис. 2. Программный код для парсинга страниц на поиск вакансии «водитель»

Данные, полученные из открытого источника API *hh.ru*, для дальнейшей работы можно сохранить в csv-файл (рис. 3).

```
# Все сохраняется в .csv файл
csv_name = job+".csv"
df.to_csv(csv_name)

df = pd.read_csv(csv_name)
df
```

Рис. 3. Программный код для сохранения набора данных в csv-файл

Анализ наименований столбцов полученного набора данных показал, что некоторые столбцы не являются необходимыми (рис. 4), поэтому их не стоит использовать: из 37 столбцов оставим для проведения исследования 8. Столбцы выбирались по принципу значимости для конкретного исследования. Так как в статье демонстрируется исследование зарплат, определенной профессии по городам России, то столбцы, не имеющие связи с тематикой исследования, отбрасывались.

```
[3]: df.columns.tolist()

[3]: ['Unnamed: 0',
      'id',
      'premium',
      'name',
      'department',
      'has_test',
      'response_letter_required',
      'area',
      'salary',
      'type',
      'address',
      'response_url',
      'sort_point_distance',
      'published_at',
      'created_at',
      'archived',
      'apply_alternate_url',
      'show_logo_in_search',
```

Рис. 4. Программный код для изучения наименований столбцов полученного набора данных

Выведем полезные для исследования характеристики (признаки), такие как:

- id* – уникальный код вакансии;
- name* – наименование вакансии;
- area* – место работы;
- salary* – зарплата;
- schedule* – график;
- address* – адрес (если указан);
- employer* – наниматель;
- contacts* – контакты, если указаны (рис. 5).

Сформируем из словаря массив данных, где возьмем минимальные значения зарплат. В том случае, если зарплата не указана, будет записано значение «NaN» (рис. 6).

Некоторые работодатели указывают зарплату, условно, в 1 рубль. В таких случаях можно установить минимально допустимое значение или определять такие значения на графике, поскольку такие «аномалии» обычно заметны сразу.

```
[4]: df[['id', 'name', 'area', 'salary', 'schedule', 'address', 'employer', 'contacts']]

[4]:
```

	id	name	area	salary	schedule	address	employer	contacts
0	109958418	Менеджер чатов (в Яндекс)	{'id': '113', 'name': 'Россия', 'url': 'https://...'}	{'from': 30000, 'to': 44000, 'currency': 'RUR'...}	{'id': 'remote', 'name': 'Удаленная работа'}	NaN	{'id': '9498120', 'name': 'Яндекс', 'Команда для ...'}	NaN
1	109952643	Водитель в семью	{'id': '1', 'name': 'Москва', 'url': 'https://...'}	{'from': 275000, 'to': 309000, 'currency': 'RU'...}	{'id': 'fullDay', 'name': 'Полный день'}	NaN	{'id': '11565807', 'name': 'Интер', 'url': 'ht...'}	NaN

Рис. 5. Программный код для вывода выбранных признаков

```
import numpy as np

df_salary = []

for i in range(len(df['salary'].tolist())):
    try:
        salary_from = eval(str(df['salary'].tolist()[i]))['from']
        if eval(str(df['salary'].tolist()[i]))['currency'] == 'RUR':
            df_salary.append(eval(str(df['salary'].tolist()[i]))['fr
    else:
        df_salary.append('NaN')
    except:
        df_salary.append('NaN')

df_salary
```

Рис. 6. Программный код для формирования массива данных с минимальными значениями зарплат

Города, тоже представляют собой словарь. По аналогии сформируем список городов. Там, где города не указаны, внесем значения «NaN» (рис. 7).

```
import numpy as np

df_area = []

for i in range(len(df['area'].tolist())):
    try:
        if eval(str(df['area'].tolist()[i]))['name'] != 'Россия':
            df_area.append(eval(str(df['area'].tolist()[i]))['name']
    else:
        df_area.append('NaN')
    except:
        df_area.append('NaN')

df_area

['NaN',
'Moskva',
'Sankt-Peterburg',
'Moskva',
'Ufa',
'Tomsk',
'Omsk',
'Sankt-Peterburg',
'Odnitsovo (Moskovskaya oblast)',
'Moskva',
'Yaroslavl',
'Moskva',
'Chita',
'Moskva',
'Moskva',
'Moskva',
'Moskva',
'Moskva']
```

Рис. 7. Программный код для формирования списка городов

Построим график распределения зарплат водителей по городам России. По оси абсцисс укажем значения зарплат, по оси ординат – города России.

На этом шаге можно удалить «NaN»-значения, так как, удаляя пропуск в массиве зарплат, необходимо удалять пропуск в значении городов для того, чтобы данные не сдвигались, и наоборот (рис. 8).

После удаления пропусков визуализируем значения (рис. 9).

```
import matplotlib.pyplot as plt

x = df_salary
y = df_area

for i in range(len(df_salary)):
    if df_salary[i] == 'NaN':
        x.pop(i)
        y.pop(i)
    elif df_area[i] == 'NaN':
        x.pop(i)
        y.pop(i)

plt.title("Зарплаты водителей по городам России")
plt.grid() #
plt.scatter(x, y)
```

Рис. 8. Программный код для построения графика распределения зарплат водителей по городам России

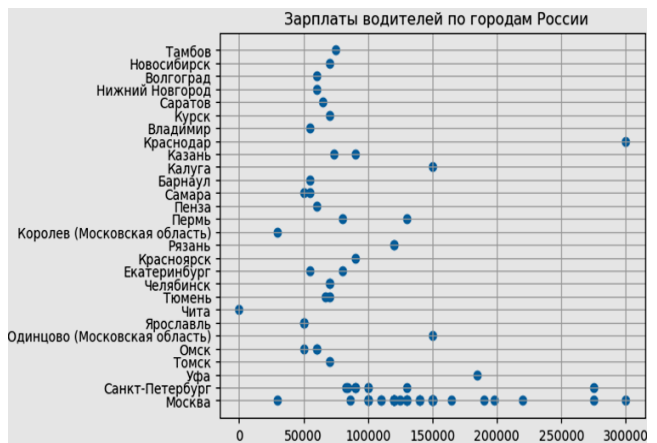


Рис. 9. График распределения зарплат водителей по городам России

На рис. 9 видно, что большая часть точек лежит в диапазоне от 50 тыс. до 100 тыс. р. Лишь в некоторых городах можно заметить значения до 150 тыс. р.

Заметим, что в Чите значение зарплаты лежит около нуля. Как говорилось ранее, скорее всего, работодатель указал зарплату в 1 р., поэтому такие данные учитывать при анализе не стоит.

Проведем аналогичный анализ для других профессий и признаков. Исследуем профессию IT-специалистов по различным городам России. В качестве исследуемого признака выберем тип графика работы.

Загрузим и сохраним в формате *csv* данные о вакансиях IT-специалистов (рис. 10).

```

number_of_pages = 10
job_title = ["'It-специалист'"]

for job in job_title:
    data=[]
    for i in range(number_of_pages):
        # API сайта hh.ru
        url = 'https://api.hh.ru/vacancies'
        par = {'text': job, 'area':'113','per_page':'10', 'page':i}
        r = requests.get(url, params=par)
        e=r.json()
        data.append(e)
        vacancy_details = data[0]['items'][0].keys()
        df = pd.DataFrame(columns= list(vacancy_details))
        ind = 0
        for i in range(len(data)):
            for j in range(len(data[i]['items'])):
                df.loc[ind] = data[i]['items'][j]
                ind+=1
    
```

Рис. 10. Программный код для парсинга страниц на поиск вакансии «IT-специалист»

Выберем, к примеру, признак *schedule*, в котором хранится информация о типе графика работы (рис. 11).

schedule
{'id': 'fullDay', 'name': 'Полный день'}
{'id': 'fullDay', 'name': 'Полный день'}
{'id': 'remote', 'name': 'Удаленная работа'}
{'id': 'fullDay', 'name': 'Полный день'}
{'id': 'remote', 'name': 'Удаленная работа'}
...

Рис. 11. Массив с данными о типе графика работы IT-специалистов

Элементы массива представляют собой словари. Обрабатываем массив, получив из словарей только значения (рис. 12).

```

import numpy as np

df_schedule = []

for i in range(len(df['schedule'].tolist())):
    try:
        salary_from = eval(str(df['schedule'].tolist()[i]))['name']
        if eval(str(df['schedule'].tolist()[i]))['id'] == 'fullDay':
            df_schedule.append(eval(str(df['schedule'].tolist()[i]))['name'])
        else:
            df_schedule.append('Удаленная работа')
    except:
        df_schedule.append('Удаленная работа')
df_schedule
    
```

Рис. 12. Программный код для формирования списка типов графика работы

С помощью библиотеки *Matplotlib* построим диаграмму распределения, чтобы наглядно увидеть и сравнить количество вакансий, предоставляющих возможность удаленной работы, и количество вакансий, которые предполагают работу полный день в офисе (рис. 13).

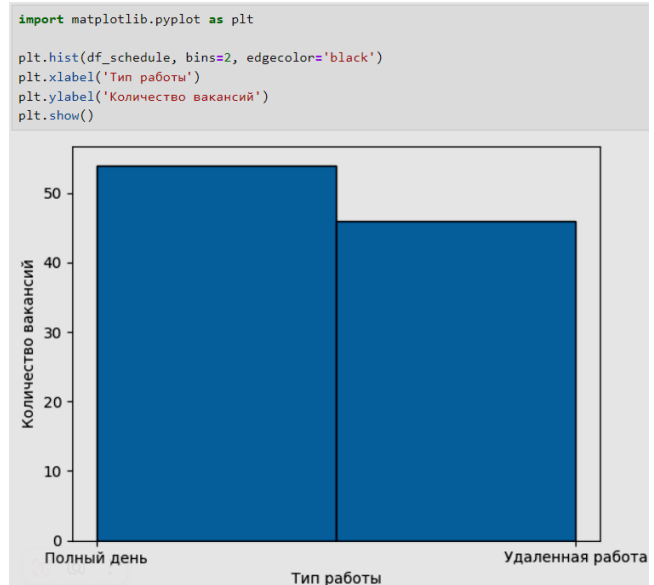


Рис. 12. Диаграмма распределения типа графика работы IT-специалистов

Обрабатываем массив данных, в котором содержится информация о городах с вакансиями IT-специалистов (рис. 13).

```

import numpy as np

df_area = []

for i in range(len(df['area'].tolist())):
    try:
        if eval(str(df['area'].tolist()[i]))['name'] != 'Россия':
            df_area.append(eval(str(df['area'].tolist()[i]))['name'])
        else:
            df_area.append('NaN')
    except:
        df_area.append('NaN')
df_area

['Новокузнецк',
 'Ижевск',
 'Москва',
 'Дзержинск (Нижегородская область)',
 'Новокузнецк',
 'Красноярск',
 'Ростов-на-Дону',
 'Москва',
 'Тамбов',
 'Ростов-на-Дону',
 'Москва',
 'Калининград',
 'Москва',
 'Ульяновск',
 'Москва',
 'Санкт-Петербург',
 'Москва',
    
```

Рис. 13. Программный код для формирования списка городов

Полученные данные можно использовать для построения диаграммы, показывающей, в каких городах вакансии IT-специалистов наиболее востребованы (рис. 14).

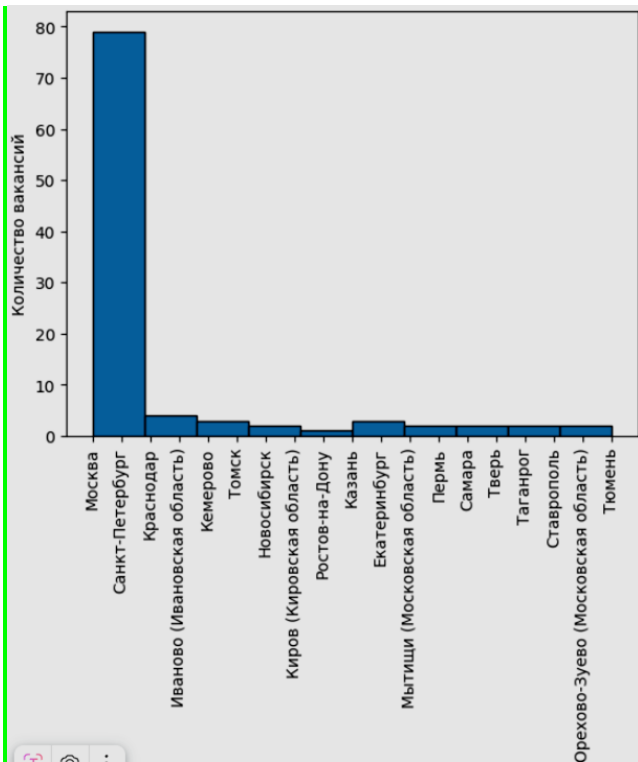


Рис. 14. Диаграмма распределения вакансий IT-специалистов по городам России

На диаграмме явно показан высокий спрос на IT-специалистов в Москве и Санкт-Петербурге. В остальных городах спрос оказался примерно на одном уровне.

Проведем аналогичный анализ для работников финансовой сферы. В качестве профессии укажем несколько возможных вариантов для того, чтобы парсинг данных охватил как можно больше подходящих объявлений (рис. 15).

```
number_of_pages = 10
job_title = ["финансист", "банк"]

for job in job_title:
    data=[]
    for i in range(number_of_pages):
        # API сайта hh.ru
        url = 'https://api.hh.ru/vacancies'
        par = {'text': job, 'area':'113','per_page':'10', 'page':i}
        r = requests.get(url, params=par)
        e=r.json()
        data.append(e)
        vacancy_details = data[0]['items'][0].keys()
        df = pd.DataFrame(columns= list(vacancy_details))
        ind = 0
        for i in range(len(data)):
            for j in range(len(data[i]['items'])):
                df.loc[ind] = data[i]['items'][j]
                ind+=1
```

Рис. 15. Программный код для парсинга страниц на поиск вакансий финансовой сферы

В качестве исследуемой переменной выберем опыт работы – *experience* (рис. 16).

experience
{'id': 'between1And3', 'name': 'От 1 года до 3...
{'id': 'between3And6', 'name': 'От 3 до 6 лет}
{'id': 'between3And6', 'name': 'От 3 до 6 лет}
{'id': 'noExperience', 'name': 'Нет опыта}
{'id': 'between3And6', 'name': 'От 3 до 6 лет}

Рис. 16. Массив с данными об опыте работы финансистов

Как и в предыдущих случаях, массив представляет собой набор словарей, которые необходимо обработать перед анализом данных (рис. 17).

```
import numpy as np

df_experience = []

for i in range(len(df['experience'].tolist())):
    try:
        salary_from = eval(str(df['experience'].tolist()[i]))['name']
        if eval(str(df['experience'].tolist()[i]))['id'] == 'noExperience':
            df_experience.append(eval(str(df['experience'].tolist()[i]))['name'])
        else:
            df_experience.append('Требуется опыт работы')
    except:
        df_experience.append('Требуется опыт работы')
df_experience
```

Рис. 17. Программный код для формирования списка требуемого опыта работы финансистов

Обработанные данные можно использовать для визуализации. Построим диаграмму распределения, чтобы наглядно увидеть разницу в количестве специалистов финансовой сферы с опытом и без опыта работы (рис. 18).

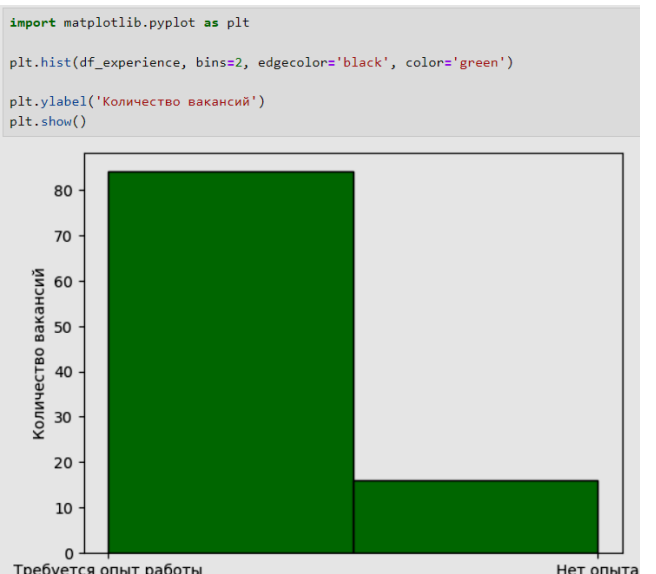


Рис. 18. Диаграмма распределения данных об опыте работы финансистов

Обработаем массив данных, в котором содержится информация о городах с вакансиями финансистов (рис. 19).

```
import numpy as np

df_area = []

for i in range(len(df['area'].tolist())):
    try:
        if eval(str(df['area'].tolist()[i])['name'] != 'Россия':
            df_area.append(eval(str(df['area'].tolist()[i])['name']))
        else:
            df_area.append('NaN')
    except:
        df_area.append('NaN')

df_area
```

```
['Новокузнецк',
 'Ижевск',
 'Москва',
 'Дзержинск (Нижегородская область)',
 'Новокузнецк',
 'Красноярск',
 'Ростов-на-Дону',
 'Москва',
 'Тамбов',
 'Ростов-на-Дону',
 'Москва',
 'Калининград',
 'Москва',
 'Ульяновск',
 'Москва',
 'Санкт-Петербург',
 'Москва',
```

Рис. 19. Программный код для формирования списка городов

Полученные данные можно использовать для построения диаграммы, показывающей, в каких городах вакансии финансистов наиболее востребованы (рис. 20).

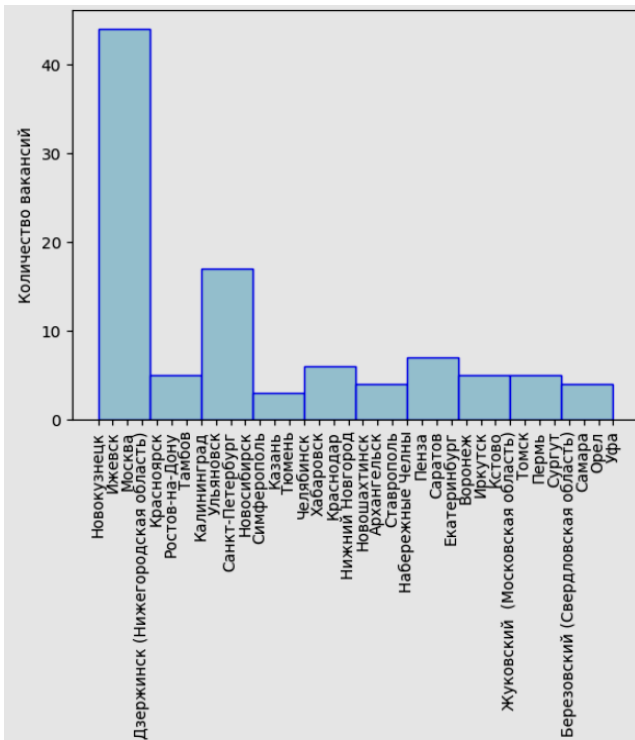


Рис. 20. Диаграмма распределения вакансий финансистов по городам России

Результаты распределения оказались иными, в отличие от данных о вакансиях в IT-сфере: финансисты оказались востребованы в Новокузнецке, Ижевске, Москве и Дзержинске.

Все представленные данные являются актуальными, так как были получены при помощи протокола API, который загружает, или обновляет, набор данных в момент выполнения кода.

Аналогичный анализ можно делать для других профессий и признаков. Можно также увеличить количество данных в наборе, указав большее количество страниц для поиска, при этом время на обработку операций будет увеличено.

Заключение. Авторские разработки, представленные в статьях российских авторов, являются перспективой подобных исследований: разработка модуля загрузки данных для обработки их в информационной системе [17]; автоматизированная обработка данных при отклике на отдельную вакансию [18]; автоматизация сбора данных для анализа рынка труда на базе API порталов интернет-рекрутмента [19]; разработка веб-приложений на основе REST API, разработка сервиса мониторинга вакансий, интернет-ресурсы и программное обеспечение для поиска информации при массовом подборе персонала [20–23].

Исследование, проведенное в статье, является универсальным решением для проведения анализа данных о вакансиях в России. Применение API-протокола на открытом источнике данных *hh.ru* позволяет получить актуальную информацию о вакансиях в момент обращения.

Обработка данных с помощью библиотек *Pandas* и *NumPy* позволяет проводить исследования по различным направлениям, отбирая информацию из полученного через протокол API набора данных.

Применение представленных в статье технологий обработки информации является актуальным решением в области исследования вакансий в городах России.

Литература

1. Стефогло С.Н., Томилова Н.И., Амиров А.Ж. Концепция открытых данных, основные принципы открытых данных // *Инновационная наука*. 2015. Т. 2, № 4. С. 53-55.
2. Парфенчик А.А. Открытые данные в контексте открытого Правительства и больших данных // *Информационные технологии и право: сб. материалов VII Междунар. науч.-практической конф.* (28 окт. 2021 г.). Минск, 2021. С. 292-297.
3. Боброва Е.В. Открытые данные и методы их изучения // *Информационный бюллетень ассоциации «История и компьютер»*. 2022. № 49. С. 199-201.
4. Лысова Н.В. Публикация официальной статистической информации в формате открытых данных // *Роль статистики в современном обществе и эффе*

- тивном управлении: материалы Всерос. науч.-практической конф., посвящ. 100-летию образования гос. статистики в Коми крае (18-28 янв. 2019 г.). Сыктывкар, 2019. С. 375-381.
5. Крамник М.С., Калиновская И.Н. Цифровые технологии в управлении человеческими ресурсами // *Материалы докладов 54-й Междунар. науч.-технической конф. препод. и студентов* (28 апр. 2021 г.). Витебск, 2021. С. 81-84.
 6. Гиоргашвили В.С., Бакаев М.А. Анализ и прогнозирование для рынков труда на основе онлайн-данных // *International Journal of Open Information Technologies*. 2018. Vol. 6, № 12. P. 12-20.
 7. Турков Е.С., Степанов Ю.А. Концептуальная модель модуля сбора данных о вакансиях для экспертной системы // *Междунар. науч.-исследовательский журнал*. 2020. № 2-1 (92). С. 75-78.
 8. Шухман А.Е., Гришина Л.С., Легашев Л.В., Парфенов Д.И. Какие компетенции ждёт региональный рынок ИТ-специалистов? Разработка и апробация технологии анализа актуальных вакансий // *Высш. образование в России*. 2022. Т. 31, № 8-9. С. 137-153.
 9. Ким А.Д., Смирнов Е.Е. Построение телеграм-бота для автоматизации парсинга вакансий на сайте HH.RU // *Инновационное развитие техники и технологий в промышленности: сб. материалов Всерос. науч. конф. молодых исследователей с междунар. участием* (1 апр. 2024 г.). М., 2024. С. 11-13.
 10. Локтев А.П. Исследовательские данные как ключевой элемент открытой науки // *Наука, технологии и информация в библиотеках (Libway-2019): сб. тез. докладов Междунар. науч.-практической конф.* (17-19 сент. 2019 г.). Иркутск, 2019. С. 129-130.
 11. Кузовлев В.И., Орлов А.О. Повышение качества данных с использованием методики поиска аномалий на примере портала открытых данных правительства Москвы // *Инженерный вестн.* 2014. № 8. С. 7.
 12. Miachin D.A., Pugach V.P., Avdeyuk S.S., Zunin V.V., Romanov A.Y. The open system for storing and processing of a dataset of combinational circuits // *Proceedings of the Institute for System Programming of the RAS*. 2023. Vol. 35, № 5. P. 81-90.
 13. Nesterov S.A., Smolina E.M. The assessment of the results of a massive open online course using data mining methods // *Computing, Telecommunications and Control*. 2020. Vol. 13, № 1. P. 65-78.
 14. Khayyat M., Aboulola O., Khayyat M. How open is open? A study of two Irish open government data websites // *Public Administration Issues*. 2022. № 5. P. 30-44.
 15. Бородин А.Ф. Сравнительный анализ систем, предоставляющих открытые данные // *Дневник науки*. 2019. № 5 (29). С. 32.
 16. Резников Д.Ю. Система сбора и анализа открытых данных: свид. о регистрации программы для ЭВМ RU 2020662024, 06.10.2020; заявл. 29.09.2020 № 2020661307.
 17. Широбокова С.Н., Жевакин Д.М. Об аспектах разработки инструментария поддержки принятия решений выбора и рейтингования вакансий из различных информационных источников // *Вестн. молодой науки России*. 2020. № 4. С. 17.
 18. Козлов Д.О. Приложение для автоматизированного формирования откликов на вакансии // *Актуальные проблемы авиации и космонавтики: сб. материалов IX Междунар. науч.-практической конф., посвящ. Дню космонавтики* (10-14 апр. 2023 г.). Красноярск, 2023. С. 191-193.
 19. Кузнецов В.Д., Хамитов Р.М., Князькина О.В. Автоматизация сбора данных для анализа рынка труда на базе API порталов интернет-рекрутмента // *Экономика и предпринимательство*. 2024. № 6 (167). С. 1430-1435.
 20. Воробьева Г.Р., Фарваев Э.Ф. Высокореактивное веб-приложение для интеграции разнородных пространственных данных из распределённых источников на основе REST API // *Вестн. Воронежского гос. ун-та. Сер.: Системный анализ и информационные технологии*. 2023. № 4. С. 152-166.
 21. Кудрявцев С.И., Яковина И.Н. Разработка сервиса мониторинга вакансий // *Наука. Технологии. Инновации: сб. науч. тр. XVII Всерос. науч. конф. молодых ученых* (4-8 дек. 2023 г.). Новосибирск, 2024. С. 78-81.
 22. Ливинская В.А., Гайчуков Е.И. О сборе данных для анализа рынка вакансий специалистов IT-отрасли // *Материалы, оборудование и ресурсосберегающие технологии: материалы Междунар. науч.-технической конф.* (22-23 апр. 2021 г.). Могилев, 2021. С. 365-366.
 23. Пимонов А.Г., Емельянов И.Д. Интернет-ресурсы и программное обеспечение для поиска информации при массовом подборе персонала // *Инновации в информационных технологиях, машиностроении и автотранспорте: сб. материалов II Междунар. науч.-практической конф.* (3-4 окт. 2018 г.). Кемерово, 2018. С. 21-23.

References

1. Stefoglo S.N., Tomilova N.I., Amirov A.Zh. The concept of open data, the basic principles of open data // *Innovation science*. 2015. V. 2, № 4. P. 53-55.
2. Parfenchik A.A. Open Data in the context of Open Government and Big Data // *Informacionnye tekhnologii i pravo: sb. materialov VII Mezhdunar. nauch.-prakticheskoy konf.* (28 okt. 2021 g.). Minsk, 2021. P. 292-297.
3. Bobrova E.V. Open data and methods of their study // *Informacionnyj byulleten' associacii «Istoriya i komp'yuter»*. 2022. № 49. P. 199-201.
4. Lysova N.V. Publication of official statistical information in open data format // *Rol' statistiki v sovremennom obshchestve i effektivnom upravlenii: materialy Vseros. nauch.-prakticheskoy konf., posvyashch. 100-letiyu obrazovaniya gos. statistiki v Komi krae* (18-28 yanv. 2019 g.). Syktyvkar, 2019. P. 375-381.
5. Kramnik M.S., Kalinovskaya I.N. Digital technologies in human resource management // *Materialy докладов 54-j Mezhdunar. nauch.-tekhnicheskoy konf. prepod. i studentov* (28 apr. 2021 g.). Vitebsk, 2021. P. 81-84.
6. Giorgashvili V.S., Bakaev M.A. Analysis and forecasting for labor markets based on online data // *International Journal of Open Information Technologies*. 2018. Vol. 6, № 12. P. 12-20.
7. Turkov E.S., Stepanov Yu.A. The conceptual model of the module for collecting data on vacancies for the ex

- pert system // Research Journal of International Studies. 2020. № 2-1 (92). P. 75-78.
8. Shuhman A.E., Grishina L.S., Legashev L.V., Parfenov D.I. What competencies does the regional market of IT specialists expect? Development and testing of technology for analyzing current vacancies // Higher Education in Russia. 2022. V. 31, № 8-9. P. 137-153.
 9. Kim A.D., Smirnov E.E. Building a telegram bot to automate job parsing on the site HH.RU // Innovacionnoe razvitie tekhniki i tekhnologij v promyshlennosti: sb. materialov Vseros. nauch. konf. molodyh issledovatelej s mezhdunar. uchastiem (1 apr. 2024 g.). M., 2024. P. 11-13.
 10. Loktev A.P. Research data as a key element of open science // Nauka, tekhnologii i informaciya v bibliotekah (Libway-2019): sb. tez. dokladov Mezhdunar. nauch.-prakticheskoy konf. (17-19 sent. 2019 g.). Irkutsk, 2019. P. 129-130.
 11. Kuzovlev V.I., Orlov A.O. Improving the quality of data using the anomaly search methodology using the example of the open data portal of the Government of Moscow // Inzhenernyj vestn. 2014. № 8. P. 7.
 12. Miachin D.A., Pugach V.P., Avdeyuk S.S., Zunin V.V., Romanov A.Y. The open system for storing and processing of a dataset of combinational circuits // Proceedings of the Institute for System Programming of the RAS. 2023. Vol. 35, № 5. P. 81-90.
 13. Nesterov S.A., Smolina E.M. The assessment of the results of a massive open online course using data mining methods // Computing, Telecommunications and Control. 2020. Vol. 13, № 1. P. 65-78.
 14. Khayyat M., Aboulola O., Khayyat M. How open is open? A study of two Irish open government data websites // Public Administration Issues. 2022. № 5. P. 30-44.
 15. Borodin A.F. Comparative analysis of systems providing open data // Dnevnik nauki. 2019. № 5 (29). P. 32.
 16. Reznikov D.Yu. Open data collection and analysis system: svid. o registracii programmy dlya EVM RU 2020662024, 06.10.2020; zayavl. 29.09.2020 № 2020661307.
 17. Shirobokova S.N., Zhevakin D.M. On the aspects of developing decision support tools for selecting and rating vacancies from various information sources // Vestn. molodyozhnoj nauki Rossii. 2020. № 4. P. 17.
 18. Kozlov D.O. An application for automated generation of job responses // Aktual'nye problemy aviatsii i kosmonavтики: sb. materialov IX Mezhdunar. nauch.-prakticheskoy konf., posvyashch. Dnyu kosmonavтики (10-14 apr. 2023 g.). Krasnoyarsk, 2023. P. 191-193.
 19. Kuznecov V.D., Hamitov R.M., Knyaz'kina O.V. Automation of data collection for labor market analysis based on the API of Internet recruitment portals // Economy and entrepreneurship. 2024. № 6 (167). P. 1430-1435.
 20. Voroby'ova G.R., Farvaev E.F. Highly responsive web application for integrating heterogeneous spatial data from distributed sources based on REST API // Proceedings of Voronezh State University. Series: Systems analysis and information technologies. 2023. № 4. P. 152-166.
 21. Kudryavcev S.I., Yakovina I.N. Development of a job monitoring service // Nauka. Tekhnologii. Innovacii: sb. nauch. tr. XVII Vseros. nauch. konf. molodyh uchenykh (4-8 dek. 2023 g.). Novosibirsk, 2024. P. 78-81.
 22. Livinskaya V.A., Gajchukov E.I. About data collection for the analysis of the market of vacancies of IT industry specialists // Materialy, oborudovanie i resursosberegayushchie tekhnologii: materialy Mezhdunar. nauch.-tekhnicheskoy konf. (22-23 apr. 2021 g.). Mogilev, 2021. P. 365-366.
 23. Pimonov A.G., Emel'yanov I.D. Online resources and software for information retrieval in mass recruitment // Innovacii v informacionnykh tekhnologiyah, mashinostroenii i avtotransporte: sb. materialov II Mezhdunar. nauch.-prakticheskoy konf. (3-4 okt. 2018 g.). Kemerovo, 2018. P. 21-23.