**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение**

**высшего образования**

**«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК**

«Визуализация среднедушевого дохода населения России с помощью интерактивной карты»

Выполнил студент

Герасимов Константин Сергеевич

МОиАИС, 1 курс

**Содержание**

**Глава 1.      Введение**

1.1 Проблема проекта.

1.2 Актуальность проекта

1.3 Цель проекта.

1.4 Задачи проекта.

**Глава 2. Основная часть**

2.1 Используемые технологии.

2.2 Обоснование выбора технологий.

2.3 Анализ данных, с которыми взаимодействует проект.

2.4 Описание структуры и логики проекта.

2.5 Листинг проекта с комментариями.

2.6 Тестирование.

**Глава 3.       Вывод**

3.1 Результаты проекта.

3.2 Краткое описание скрипта.

3.3 Общая оценка результата проекта.

**4.     Список литературы**

**Глава 1. Введение**

**1.1 Проблема проекта.**

На сегодняшний день, доходы жителей России очень сильно рознятся в зависимости от региона. Однако рядовой пользователь может быть не в курсе того, насколько сильна эта дифференциация.

**1.2 Актуальность проекта.**

На сегодняшний день эта проблема очень актуальна, так как среднедушевые доходы населения – ключевой показатель благосостояния жителей той или иной страны.

**1.3 Цель проекта.**

Создание интерактивной карты, которая отображает дифференциацию доходов населения в зависимости от года, за который были проведены подсчеты официальными государственными службами статистики.

**1.4 Задачи проекта.**

1. Поиск необходимой информации на сайте витрины государственной статистики.
2. Проектирование базы данных на основе .geojson и .csv.
3. Разработка программы, создающей карты в реальном времени, используя стек данных на основе .geojson с помощью Python.
4. Разработка интерактивной карты на основе .geojson с помощью Python и HTML.
5. Тестирование интерактивной карты.
6. Представление проекта аудитории.

**Глава 2. Основная часть**

**2.1 Используемые технологии.**

1. В качестве основного языка программирования для реализации проекта был выбран язык Python 3.11. Python – язык, направленный на Data-аналитику и обладающий множеством библиотек, которые позволяют визуализировать данные, создавать интерактивные решения множества проблем. Он используется во множестве сфер науки и бизнеса, в том числе и при изучении благосостояния жизни. Именно поэтому выбор пал на язык программирования Python.
2. Geojson – открытый формат, предназначенный для хранения географических структур данных. С помощью него можно воспроизводить карты и фигуры. Также geojson, как и обычный json, имеет возможность хранить и другие типы данных, что и привлекло использование именно этого формата фалов, для хранения гео-структур и экономических данных по каждому региону России.
3. Folium – открытая библиотека для Python, позволяющая создавать интерактивные карты на основе Leaflet.js. С помощью этой библиотеки можно просто и быстро визуализировать данные из различных форматов: geojson, topotiff, geotiff и другие.
4. Matplotlib – открытая библиотека для Python, позволяющая визуализировать данные с помощью графиков, используя вышеуказанные форматы данных.
5. Pandas и Geopandas – две схожие библиотеки для Python, позволяющие удобно и легко редактировать, изучать и разрабатывать стеки данных на основе .csv, .json и .geojson файлов.

**2.2 Обоснование выбора технологий и структур данных.**

Выбор пал именно на вышеуказанные технологии и структуры данных из-за их простоты работы с географическими массивами данных. Python поддерживает высокий уровень разработки скриптов для Data-аналитики, на чем и завязан данный проект. Форматы .csv и .geojson очень удобны для работы, так как их редактирование может осуществляться на встроенном софте. В целом, выбор технологий был основан в первую очередь на удобности их использования.

**2.3 Анализ данных, с которыми взаимодействует проект.**

**Входные данные:**

* .csv файл, содержащий стек данных за 11 лет, отображающий доходы населения Российской Федерации в каждом регионе.
* .geojson, содержащий стек географических структур России. Он включает себя координаты точек, по которым необходимо строить сложные полилинии, составляющие географический рисунок РФ.

**Выходные данные:**

* .html файл, содержащий описание интерактивной карты
* .jpg файлы, наглядно отражающие дифференциацию доходов населения.

**Типы данных:**

* Data Frame – тип данных, позволяющий работать с .geojson и .csv
* Списки и словари
* Folium.Map – тип данных, позволяющий формировать интерактивную карту.
* Folium.Choropleth – тип данных, формирующий визуальный слой для карты.
* Coordinates System EPSG – системы координат, позволяющие лучше визуализировать данные для России.

**2.4 Описание структуры и логики проекта.**

Структура проекта состоит из двух программ:

рandas\_map.py – Python файл, в котором описан процесс слияния данных из .csv в .geojson и визуализации стека данных за 11 лет на картах.

maps.py – Python файл, в котором реализован скрипт создания интерактивной карты на основе .geojson, полученного в ходе слияния в программе pandas\_map.py.

[Стек географических данных](https://github.com/timurkanaz/Russia_geojson_OSM) обрабатывается с помощью библиотеки geopandas, далее он сливается со стеком экономических данных с помощью библиотеки pandas, слияние происходит по соответствию названий регионов.

Затем на основе этих данных библиотекой matplotlib создаются графики, отражающие дифференциацию доходов жителей РФ.

Затем файл, содержащий географический массив информации и экономический массив передается на вход программе maps.py, где в свою очередь, с помощью библиотеки Folium создается html файл, содержащий интерактивную карту со слоями-годами.

**2.5 Листинг проекта с комментариями.**

**Pandas\_map.py**

# импорт библиотек  
import pandas as pd  
import geopandas as gpd  
import matplotlib.pyplot as plt  
  
  
# функиця, создающая карты  
def make\_map():  
 data = pd.read\_csv('/Users/cuperuser/kostyA/Coding/Programming2Semester/data.csv')  
 shape = gpd.read\_file("/Users/cuperuser/kostyA/Coding/Programming2Semester/Russia\_geojson\_OSM-master/GeoJson's"  
 "/Countries/Russia\_regions.geojson")  
 # слияние двух стеков данных  
 shape = pd.merge(  
 left=shape,  
 right=data,  
 left\_on='region',  
 right\_on='Region',  
 how='left')  
  
 yrs = []  
 for i in range(3, len(shape.columns.tolist())):  
 yrs.append(shape.columns.tolist()[i])  
 # экспорт geojson, использующийся в программе maps.py  
 shape.to\_file(f"map.geojson")  
  
 for i in range(len(yrs)):  
 shape = shape.to\_crs(epsg=3576)  
 fig, ax = plt.subplots(figsize=(12.7, 7.2))  
 shape.plot(ax=ax,  
 edgecolor='black',  
 linewidth=0.5,  
 cmap='YlOrRd',  
 column=yrs[i],  
 legend=True,  
 legend\_kwds={'shrink': 0.3, 'orientation': 'horizontal'})  
  
 ax.get\_xaxis().set\_visible(False)  
 ax.get\_yaxis().set\_visible(False)  
 for edge in ['right', 'left', 'bottom', 'top']:  
 ax.spines[edge].set\_visible(False)  
 # экспорт jpg файлов-карт  
 ax.set\_title(f"Доходы населения Российской Федерации, {yrs[i][-5:-1]}", size=18, weight='bold')  
 plt.savefig(f"maps/{yrs[i][-5:-1]}.png")  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 make\_map()

**Maps.py**

# Импорт библиотек  
import folium  
import geopandas as gpd  
  
data = gpd.read\_file("map.geojson")  
# Создание css-стиля для оформления карт  
borderStyle = {  
 'color': 'red',  
 'weight': 1,  
 'fillOpacity': 0  
}  
# Создание основной карты  
russia\_map = folium.Map(  
 location=[64.6863136, 97.7453061], # широта и долгота России  
 zoom\_start=4,  
 min\_zoom=4,  
 max\_zoom=5,  
 zoom\_control=False,  
)  
# Создание слоев со стеком данных за год  
one = folium.Choropleth(geo\_data=data,  
 data=data,  
 columns=['Region', 'Personal Income (2010)'],  
 key\_on='feature.properties.Region',  
 fill\_color='YlOrRd',  
 name='2010',  
 overlay=False,  
 nan\_fill\_color='White').add\_to(russia\_map)  
  
two = folium.Choropleth(geo\_data=data,  
 data=data,  
 columns=['Region', 'Personal Income (2020)'],  
 key\_on='feature.properties.Region',  
 fill\_color='YlOrRd',  
 name='2020',  
 show=False,  
 overlay=False,  
 nan\_fill\_color='White').add\_to(russia\_map)  
  
one\_j = folium.features.GeoJson(data=data,  
 name='Russia',  
 style\_function=lambda x: borderStyle,  
 control=False,  
 overlay=False).add\_to(one)  
  
two\_j = folium.features.GeoJson(data=data,  
 name='Russia',  
 style\_function=lambda x: borderStyle,  
 control=False,  
 overlay=False).add\_to(two)  
# создание всплывающих интерактивных окон, для отображения информации по каждому региону  
folium.features.GeoJsonPopup(fields=['region', 'Personal Income (2010)'],  
 aliases=['Название региона: ',  
 'Доход жителя 2010г. : ']).add\_to(one\_j)  
  
folium.features.GeoJsonPopup(fields=['region', 'Personal Income (2020)'],  
 aliases=['Название региона: ',  
 'Доход жителя 2020г. : ']).add\_to(two\_j)  
# добавление возможности переключения между слоями  
folium.LayerControl().add\_to(russia\_map)  
# экспорт карты в html файл  
russia\_map.save('maps\_examples/Map.html')

**2.6 Тестирование.**

Чтобы проверить работоспособность программ, необходимо собрать данные. Для этого отправляемся на сайт <https://showdata.gks.ru> и делаем выгрузку за 11 лет по доходам населения РФ. Далее, на странице проекта <https://github.com/timurkanaz/Russia_geojson_OSM> выгружаем географический стек данных.

Затем указываем пути к данным и запускаем программы. Все сделается автоматически.

Pandas\_map.py

Изображение выглядит как снимок экрана, текст

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Maps.py

Изображение выглядит как текст, снимок экрана, черный

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, письмо, белый

Автоматически созданное описание

Изображение выглядит как карта, текст, атлас

Автоматически созданное описание

Изображение выглядит как карта, текст, атлас

Автоматически созданное описание

**Глава 3. Вывод.**

**3.1 Результаты проекта.**

Проект выполняет поставленные цели и решает проблему визуализации данных в виде интерактивной карты. Разработаны программы, которые отображают текущее состояние дифференциации доходов населения нашей страны. Результаты проекта возможно открыть на любом компьютере и интегрировать в иные проекты.

**3.2 Краткое описание программ.**

Программы написаны кратким, чистым, соответствующим РЕР-8 требованиям кодом. Они содержат пояснительные комментарии к ключевым моментам. Язык Python очень хорошо раскрыл представленные стеки данных и реализовал их в виде картинок и HTML файла.

**3.3 Общая оценка результата проекта.**

В ходе работы над проектом были изучены различные способы визуализации данных, проектирования и составления стеков данных. Был разработан скрипт, который создает HTML файл, содержащий интерактивную карту регионов и их экономические показатели. Были созданы изображения-графики, отражающие дифференциацию доходов за 11 лет.

**4. Список литературы**

1. Чистик О.Ф. Анализ уровня и дифференциации доходов населения в регионах Российской Федерации // Вестник Самарского государственного экономического университета. - 2015. - №1. - С. 6.
2. Гусейнова В. А., А. А. Богданова\*, Т. Н. Афанасьева ВЛИЯНИЕ ИНФЛЯЦИИ НА РЕАЛЬНЫЕ ДОХОДЫ НАСЕЛЕНИЯ // АКТУАЛЬНЫЕ ПРОБЛЕМЫ АВИАЦИИ И КОСМОНАВТИКИ Учредители: Сибирский государственный университет науки и технологий им. акад. М.Ф. Решетнева. - 2017. - №13. - С. 452-454.
3. Гуляева Т. И., Такмакова Е. В. // ОЦЕНКА УРОВНЯ ЖИЗНИ НАСЕЛЕНИЯ РЕГИОНОВ РОССИИ НА ОСНОВЕ ПРИМЕНЕНИЯ КЛАСТЕРНОГО АНАЛИЗА. Экономический анализ: теория и практика. Том 20, no 5, 2021.
4. Evsey Gurvich, Ilya Prilepskiy The impact of financial sanctions on the Russian economy // Russian Journal of Economics . – (2015) . - №1
5. Irving B. Kravis, Alan W. Heston and Robert Summers REAL GDP PER CAPITA FOR MORE THAN ONE HUNDRED COUNTRIES\* // The Economic Journal. – 1978. - №350. – С. 88.