

**Федеральное государственное автономное образовательное  
учреждение высшего образования “Национальный  
исследовательский университет  
“Высшая школа экономики”  
Московский институт электроники и математики им. А.Н. Тихонова  
НИУ ВШЭ  
Департамент компьютерной инженерии**

**Курс: «Проектный семинар «Python в науке о данных»**

**Руководство разработчика  
по проекту “Political Coordinates”**

**Группа:**

БИВ225

**Номер бригады:**

\_\_\_\_\_

**Директор бригады:**

Носов Иван Александрович

+7 (924) 301-27-51

ianosov@edu.hse.ru

**Состав бригады:**

Носов Иван Александрович,

Пантякова Дарья Евгеньевна,

Лифановский Дмитрий Валентинович

**Руководитель:**

Полякова Марина Васильевна

**МОСКВА 2023**

## Содержание

1. Требования к характеристикам компьютера и операционной системе
2. Версии интерпретатора и используемых библиотек
3. Инструкция по установке приложения
4. Инструкция по запуску и настройке приложения
5. Описание структуры БД
6. Структура каталогов
7. Архитектура приложения
8. Листинг основного скрипта

## 1. Требования к характеристикам компьютера и операционной системе

Для корректной работы программы на компьютере должны быть установлены интерпретатор "Python" (вне зависимости от среды разработки) и следующие библиотеки: numpy, pandas, python-dateutil, pytz, six и tzdata.

Также необходимым условием для работы программы является операционная система. Подходящими версиями являются Windows 10+.

## 2. Версии интерпретатора и используемых библиотек

Интерпретатор – Python 3.7 и выше.

Используемые библиотеки:

Библиотека	Версия
numpy	1.24.3
pandas	2.0.1
python-dateutil	2.8.2
pytz	2023.3
six	1.16.0
tzdata	2023.3

## 3. Инструкция по установке приложения

Если установлена Anaconda, то пользователь уже имеет большинство необходимых библиотек, таких как NumPy, pandas, matplotlib, и многие другие. В случае, если установлен только интерпретатор Python3, пользователь должен установить дополнительные библиотеки. Для этого необходимо воспользоваться менеджером пакетов pip, который уже установлен в большинстве компьютеров под управлением ОС Windows. Чтобы установить необходимые библиотеки, пользователь должен запустить терминал и в нужной директории ввести команду "pip install <имя библиотеки>". Список всех необходимых библиотек и их версий находится в таблице в предыдущем разделе, а также в файле "requirements.txt", находящемся в поставляемом архиве.

## 4. Инструкция по запуску и настройке приложения

Запуск программы производится в необходимой директории ...work/main.py путем вызова в этой директории следующей команды:

`python main.py`

Дополнительных настроек после запуска программы и полноценной работы в ней производить не надо, в противном случае стоит обратиться к руководству пользователя к разделу "5.11 Настройки программы".

## 5. Описание структуры БД

Все базы данных имеют поля:

1. id (индивидуальный номер записи) - натуральное число;
2. gender (гендер респондента) - строка;
3. field (научное направление) - строка;
4. university (учебное заведение) - строка;
5. course - курс обучения респондента;
6. x (политическая координата по оси X) - вещественное число;
7. y (политическая координата по оси Y) - вещественное число;
8. z (политическая координата по оси Z) - вещественное число.

## 6. Структура каталогов

Первый уровень	Второй уровень	Объяснение
Work		Основной каталог с подкаталогами
	main.py	Основной исполняемый файл
	library	Содержит библиотеку стандартных функций
	scripts	Содержит специализированный модуль и файл с определением параметров настройки приложения
	data	Содержит изначальную базу данных
	notes	Содержит всю документацию по данному проекту

## 7. Архитектура приложения

Модуль	Местонахождение	Функция
main.py	work/main.py	Основной модуль для запуска приложения
library.py	work/library/library.py	Модуль, содержащий функции, методы и классы, необходимые для корректной работы и функционирования модуля scripts.py
script.py	work/scripts/script.py	Главный модуль, содержащий все функции, классы и методы, необходимые для функционирования приложения

## 8. Листинг основного скрипта

Модуль	Функции с докстрингами
main.py <pre> """ Основной модуль для запуска приложения Автор: Носов Иван Александрович """ </pre>	
library.py <pre> """ Модуль, необходимый для работы модуля script.py """ </pre>	<pre> def read_config():     """ Читает конфигурационный файл и возвращает     его содержимое в виде словаря     Автор: Пантякова Дарья Евгеньевна     Вход: отсутствует     Выход: dict - содержимое конфигурационного файла     """  def read_csv():     """ Читает csv-файл и возвращает его содержимое в     виде DataFrame </pre>

	<p>Автор: Пантякова Дарья Евгеньевна  Вход: отсутствует  Выход: pd.DataFrame - содержимое csv-файла  """</p> <p>def delete_row(df, row_index):  """" Удаляет строку из DataFrame по индексу, перенумеровывает индексы и сохраняет изменения в csv-файл  Автор: Носов Иван Александрович  Вход: df (pd.DataFrame) - исходный DataFrame, из которого нужно удалить строку; row_index (int) - индекс строки, которую нужно удалить  Выход: pd.DataFrame - DataFrame с обновленным содержимым  """</p> <p>def save_to_file(df):  """" Сохраняет DataFrame в csv-файл  Автор: Носов Иван Александрович  Вход: df (pd.DataFrame) - DataFrame, который нужно сохранить в csv-файл  Выход: отсутствует  """</p> <p>def pivot(data, values, column, index, aggfunc):  """" Создает и выводит сводную таблицу из DataFrame на основе указанных аргументов  Автор: Пантякова Дарья Евгеньевна  Вход: data (pd.DataFrame) - исходный DataFrame, из которого будет создана сводная таблица; values (str) - имя столбца, который используется для вычисления агрегированных значений; column (str) - имя столбца, который будет использоваться для создания столбцов в сводной таблице; index (str) - имя столбца, который будет использоваться для создания индекса в сводной таблице; aggfunc (function) - функция, которую нужно применить к значениям в сводной таблице</p>
--	---

	<p>Выход: pd.DataFrame / None - сводная таблица, созданная из исходного DataFrame или ничего, если указанный столбец не найден</p> <pre> """ def qual_var_text_report(df, qualitative_var): """ Создает текстовый отчет для качественной переменной из переданного DataFrame Автор: Носов Иван Александрович Вход: df (pandas.DataFrame) - DataFrame, который содержит качественную переменную; qualitative_var (str) - имя столбца в df, который представляет качественную переменную Выход: table (pandas.DataFrame) - DataFrame, который содержит три столбца - 'Значение', 'Частоты' и 'Процент' """  def quantitative_text_report(df, quantitative_vars): """ Создает текстовый отчет для количественных переменных из переданного DataFrame Автор: Носов Иван Александрович Вход: df (pandas.DataFrame) - DataFrame, который содержит количественные переменные; quantitative_vars (list) - список имен столбцов в df, которые представляют количественные переменные Выход: table (pandas.DataFrame) - DataFrame, который содержит статистические меры для каждой из количественных переменных """ </pre>
<p>script.py</p> <pre> """ </pre> <p>Главный модуль, содержащий все функции, классы и методы, необходимые для функционирования приложения</p> <pre> """ </pre>	<pre> def __init__(self): """ Инициализирует графическое приложение 'Политические координаты', настраивает начальную конфигурацию окна приложения. Он устанавливает заголовок окна, тему и позицию, а также запрещает изменение размеров окна. Затем вызывается метод create_widgets для создания необходимых виджетов для приложения. Автор: Носов Иван Александрович Вход: отсутствует </pre>

	<p>Выход: отсутствует        """</p> <p>def set_styles(self):        """ Настраивает стили виджетов для приложения.        Автор: Лифановский Дмитрий Валентинович        Вход: отсутствует        Выход: отсутствует        """</p> <p>def delete_update(self, index, table, frame):        """ Удаляет строку с заданным индексом из таблицы        и обновляет таблицу.        Автор: Лифановский Дмитрий Валентинович        Вход: index (str) - индекс строки, которую нужно        удалить; table (ttk.Treeview) - таблица, из которой        нужно удалить строку; frame (tk.Frame) - рамка, в        которой расположена таблица.        Выход: отсутствует        """</p> <p>def save_settings(self):        """ Сохраняет настройки, введенные пользователем.        Автор: Лифановский Дмитрий Валентинович        Вход: отсутствует        Выход: отсутствует        """</p> <p>def next_question(self, last_answer, question_label, sex,        direction, university, course):        """ Переходит к следующему вопросу в тесте и        обновляет результаты ответов.        Автор: Носов Иван Александрович        Вход: last_answer (str) - последний выбранный ответ;        question_label (ttk.Label) - метка с текущим вопросом;        sex (str) - пол пользователя; direction (str) - выбранное        направление; university (str) - выбранный университет;        course (str) - выбранный курс.        Выход: отсутствует        """</p>
--	--



	<pre>def create_report_from_dataframe(self, df, f3_output):     """ Создает отчет из данных DataFrame и отображает     его в виджете Treeview.     Автор: Пантякова Дарья Евгеньевна     Вход: df (pandas.DataFrame) - исходные данные;     f3_output (ttk.Treeview) - виджет Treeview для     отображения отчета.     Выход: отсутствует     """  def bar(self, frame, data, x_column, y_column, row, column, columnspan):     """ Функция создает кластеризованную столбчатую     диаграмму (bar plot) с использованием библиотеки     matplotlib.pyplot.     Автор: Пантякова Дарья Евгеньевна     Вход: frame (tkinter.Frame) - фрейм tkinter, в котором     нужно разместить столбчатую диаграмму; data     (pandas.DataFrame) - DataFrame, содержащий данные;     x_column (str) - имя столбца в data, представляющего     категории (ось x) на столбчатой диаграмме; y_column     (str) - имя столбца в data, представляющего     подкатегории (ось y) на столбчатой диаграмме; row     (int) - номер строки в frame, в которой будет     размещена столбчатая диаграмма; column (int) -     номер столбца в frame, в котором будет размещена     столбчатая диаграмма; columnspan (int) - ширина     столбчатой диаграммы в frame (в количестве     столбцов).     Выход: отсутствует     """  def plot_hist(self, frame, df, qual_attr, quant_attr, row, column, columnspan):     """ Функция создает гистограмму для пары     "количественный атрибут - качественный атрибут" с     использованием библиотеки matplotlib.     Автор: Пантякова Дарья Евгеньевна</pre>
--	---

	<p>Вход: frame (tkinter.Frame) - фрейм tkinter, в котором нужно разместить диаграмму; df (pandas.DataFrame) - DataFrame, содержащий данные; qual_attr (str) - имя столбца в df, представляющего качественный атрибут; quant_attr (str) - имя столбца в df, представляющего количественный атрибут; row (int) - номер строки в окне tkinter, в которой будет размещена гистограмма; column (int) - номер столбца в окне tkinter, в котором будет размещена гистограмма; columnspan (int) - ширина гистограммы в окне tkinter (в количестве столбцов).</p> <p>Выход: отсутствует</p> <p>"""</p> <pre>def create_boxplot(self, frame, df, qual_attr, quant_attr, row, column, columnspan):</pre> <p>""" Функция создает ящиковую диаграмму (boxplot) для пары "количественный атрибут - качественный атрибут" с использованием библиотеки matplotlib.pyplot.</p> <p>Автор: Лифановский Дмитрий Валентинович</p> <p>Вход: frame (tkinter.Frame) - фрейм tkinter, в котором нужно разместить точечную диаграмму; df (pandas.DataFrame) - DataFrame, содержащий данные; qual_attr (str) - имя столбца в df, представляющего качественный атрибут; quant_attr1 (str) - имя первого столбца в df, представляющего количественный атрибут; quant_attr2 (str) - имя второго столбца в df, представляющего количественный атрибут; row (int) - номер строки в frame, в которой будет размещена точечная диаграмма; column (int) - номер столбца в frame, в котором будет размещена точечная диаграмма; columnspan (int) - ширина точечной диаграммы в frame (в количестве столбцов).</p> <p>Выход: отсутствует</p> <p>"""</p>
--	--