«Визуальная аналитика», Киреев В.С.

Лабораторная работа № 2

Тема: «Визуализация данных на языке Python с помощью библиотеки matplotlib»

Цель работы: более глубоко изучить настройки matplotlib и отдельные виды графиков применительно к реальным данным.

Теоретическая справка

Чтобы настроить цвет, можно использовать ключевое слово color, которое принимает строковый аргумент, представляющий практически любой цвет. Цвет можно указать разными способами:

- color='blue'
- color='g'
- color='0.75'
- color='#FFDD44'
- color=(1.0,0.2,0.3)
- color='chartreuse'

В графиках, связанных с линиями (например, plt.plot) можно указать тип линии, либо с помощью названия, либо с помощью кода:

- linestyle='-' solid
- linestyle='--' dashed
- linestyle='-.' dashdot
- linestyle=':' dotted

В matplotlib имеется выбор пределов осей по умолчанию для вашего графика, но иногда лучше иметь более точный контроль. Самый простой способ настроить пределы оси — использовать методы plt.xlim() и plt.ylim().

- plt.xlim(-1, 11)
- plt.ylim(-1.5, 1.5);

Заголовки и метки осей — самые простые из таких меток — есть методы, которые можно использовать для их быстрой установки:

- plt.title("синус")
- plt.xlabel("x")
- plt.ylabel("sin(x)");

Каждый раз, когда matplotlib загружается, он определяет конфигурацию времени выполнения (rc), содержащую стили по умолчанию для каждого создаваемого вами элемента графика. Можно изменить эту конфигурацию в любое время с помощью удобной процедуры plt.rc. Далее представлены некоторые параметры с примерами

- plt.rc('axes', facecolor='#E6E6E6', edgecolor='none', axisbelow=True, grid=True, prop_cycle=colors)
- plt.rc('grid', color='w', linestyle='solid')
- plt.rc('xtick', direction='out', color='gray')
- plt.rc('ytick', direction='out', color='gray')
- plt.rc('patch', edgecolor='#E6E6E6')
- plt.rc('lines', linewidth=2)

В matplotlib версии 1.4 в августе 2014 года был добавлен очень удобный модуль стилей, который включает ряд новых таблиц стилей по умолчанию, а также возможность создавать и упаковывать собственные стили. Далее представлены некоторые команды работы со стилями:

- plt.style.available (показать доступные стили)
- with plt.style.context('fivethirtyeight'):
  какой-то график

Круговая диаграмма (pie chart) строится с помощью pyplot (псевдоним plt) командой plt.pie и обладает в том числе следующими параметрами:

- explode – массив зазоров между клиньями,
- labels – массив подписей клиньев,
- autopct='%.2f%%', Параметр autopct отображает процентное значение срезов. Если autopct имеет значение %.2f, то для каждого сегмента круговой диаграммы строка формата имеет вид %.2f, где % — это специальный символ, указывающий, когда вводить значение, f задает результат как тип с плавающей запятой, а . 2 устанавливает ограничение только на 2 цифры после точки.
- colors –массив цветов окраски клиньев. Пример цветов - 'wheat', 'crimson', 'lightgrey'.

Кольцевая диаграмма очень похожа на круговую диаграмму. Однако, поскольку кольцевая диаграмма имеет отверстие в центре, срезы больше похожи на столбцы. К сожалению, в библиотеке matplotlib нет специального метода построения кольцевой диаграммы. Но можно использовать параметр wedgeprops для определения ширины клиньев:

- wedgeprops={'width': 0.2}

В matplotlib есть несколько функций, которые можно использовать для создания тепловой карты. Среди прочих имеется метод plt.imshow(). Единственный требуемый аргумент — это X — набор данных для графика:

- plt.imshow(df.corr())

По умолчанию цветная полоса (plt.colorbar()) представляет собой вертикальную линию с правой стороны графика:

- plt.imshow(df.corr(), cmap="Spectral")
- plt.colorbar(orientation='horizontal')

Чтобы создать столбиковую гистограмму с помощью matplotlib, просто нужно вызвать функцию bar(). Синтаксис этого метода следующий:

plt.bar(x, высота, ширина, низ, выравнивание), где:

- x это категория
- высота является соответствующим значением.
- Ширина — это ширина полос (значение по умолчанию — 0,8).
- нижняя — основание координаты y; другими словами, это точка, где начинаются ваши полосы. (по умолчанию 0)
- align — это место, где вы хотите разместить названия категорий. По умолчанию они располагаются в центре полосы.

Гистограмма — это графическое отображение данных, в котором группы точек данных организованы в диапазоны. Эти диапазоны представлены барами. Это похоже на гистограмму, но это не совсем то же самое. Ключевое отличие состоит в том, что вы используете столбчатую диаграмму для представления данных по категориям, а гистограмма отображает только числовые данные.

По умолчанию столбцы располагаются рядом. Альтернативным способом является наложение значений друг на друга. Вы можете сделать это, установив для аргумента stacked значение True. Мы также добавляем цвет края для лучшей читабельности:

- plt.hist([data1, data2], bins=bins, label=names, stacked=True, edgecolor='white')

Другим часто используемым типом графика является простой точечный график, близкий родственник линейного графика. Вместо того, чтобы соединять точки отрезками, здесь точки представлены по отдельности в виде точки, круга или другой формы:

- plt.scatter(x, y, c=colors, s=sizes, alpha=0.3,cmap='viridis')

Самостоятельное задание

1. Импортировать данные lab2.csv в соответствии с кодом:

```
import pandas as pd
df=pd.read_csv('/content/lab2.csv')
```

Должна получиться таблица следующего вида:

id	name	area.name	salary.from	salary.to	salary.currency	salary.gross	employer.name	snippet.requirement	snippet.responsibility	schedule.name
77448260	Lead Data Engineer	Москва	NaN	500000.0	RUR	false	Wanted	С прошлого года активно развиваем новую область компетенций - рекомендательные ML системы, уже успешно реализовали проект для крупного заказчика в сфере...	...обучения и участвовать в их продукционализации совместно с <highlighttext>Data</highlighttext> Science командой. Разрабатывать продукт в команде и развивать отдел <highlighttext>Data</highlighttext> <highlighttext>Engineer</highlighttext>.	Гибкий график
77015802	Data Engineer	Москва	200000.0	280000.0	RUR	true	HR Prime	Хороший уровень работы с БД. Опыт с Python. Опыт работы с ClickHouse.	Поддержание работоспособности платформы. Разработка новых инструментов. Создание новых витрин данных.	Полный день
75921771	Data engineer	Москва	200000.0	250000.0	RUR	false	РВИ	Высшее техническое образование, предпочтительные профили: экономика/финансы/математика/физика. Знание специализированных программных средств для работы с базами данных.	Системный анализ, инженерная аналитика. Разработке архитектуры данных и структур баз данных. Работа с базами данных, формирование логики наполнения и хранения...	Полный день

2. Построить столбиковую диаграмму по числу пропущенных значений в каждом поле загруженных данных. Предварительно отсортировать частоты по убыванию.
   2.1. Добавить данные по числу уникальных значений в каждом поле
   2.2. Заменить фон диаграммы на серый ('#E6E6E6')
   2.3. Заменить ширину зазоров между столбцами на
3. Построить столбиковую диаграмму по числу аномальных значений (слишком больших, слишком маленьких, по значению, по числу символов) по полям salary.from, salary.to, snippet.requirement, snippet.responsibility
   3.1. Применить к диаграмме стиль “ggplot”.
4. Построить круговую диаграмму по количеству вакансий в городах, используя поле area.name.
   4.1. Добавить название графика, подписи клиньев с названием городов, доли, абсолютного значения.
   4.2. Сделать выделяющимся клин с самой большой частотой (использовать параметр разброса – explode).
5. Построить гистограмму по полю salary.to
   5.1. Добавить поле город и сделать гистограмму группированной (stacked) по городу
6. Построить тепловую карту средней зарплаты по полю salary.to в разрезе – расписания работы и города.
7. Построить ящичковую диаграмму по полю salary.from в разрезе города.
   7.1. Добавить свой цвет для города внутри диаграммы
8. Построить пончиковую диаграмму по частоте технологий - систем, фреймворков, библиотек (выделить из требований к кандидату с помощью строковых функций или модуля регулярных выражений - re)
9. Построить диаграмму рассеяния - число лет опыта (выделить из требований к кандидату кандидату с помощью строковых функций или модуля регулярных выражений - re) – зарплата (поле salary.from)
   9.1. Добавить сетку
   9.2. Окрасить точки в цвета, связанный с городом
   9.3. Добавить размер точки – число вакансий, приходящихся на координаты точки
10. Построить горизонтальную столбиковую диаграмму описывающую вклад каждого фактора в зарплату, если под факторами понимать город и расписание работы. Использовать линейный регрессионный анализ.