«Визуальная аналитика», Киреев В.С.

Лабораторная работа № 10

Тема: «Создание базового дэшборда в Streamlit и plotly »

Цель работы: получить базовые навыки работы с фреймворком Streamlit и plotly.

Теоретическая справка

Plotly Express — это высокоуровневый API для создания диаграмм.

для импорта модуля

import plotly.express as pe

для создания графика двухмерной плотности, с дополнительными гистограммами по обоим осям

pe.density_contour(df, x="age", y="height", marginal_x="histogram", marginal_y="histogram")

Plotly.graph_objects – интерфейс, который содержит основные объекты и их методы, из которых состоят любые диаграммы plotly. Далее представлены некоторые типовые команды:

для импорта модуля

from plotly import graph_objects as go

создать область для рисования

fig=go.Figure()

добавить трассировку, содержащую диаграмму рассеяния

fig.add_trace(go.Scatter(x=df['age'], y=df['height'],mode = 'markers'))

изменить макет в области для рисования

fig.update_layout(xaxis_title='Возраст',yaxis_title='Рост')

показать результирующую диаграмму

fig.show()

Для создания сетки графиков

from plotly.subplots import make_subplots

specs – параметр описывающий диаграммы в сетке,

rowspan – параметр в спецификации, отвечающий за число строк, занимаемых диаграммой в сетке

Для замены графического движка в pandas

import pandas as pd

pd.options.plotting.backend = "plotly"

Streamlit

Streamlit — это фреймворк для языка программирования Python. Каждый раз, когда пользователь взаимодействует с получившимся веб-интерфейсом или разработчик меняет что-то в коде, Streamlit сам обновляет и перерисовывает нужные части страницы. Поэтому с помощью фреймворка можно делать интерактивные визуализации, дашборды или простые пользовательские сервисы. В Streamlit есть встроенные стандартные виджеты для частых действий, например ползунки или поля для ввода текста.

Streamlit позволяет отображать текст по-разному:

- st.title() – для установки заголовка;
- st.text() – для записи описания для конкретного графика;
- st.markdown() – для отображения текста в виде markdown;
- st.latex() – для отображения математических выражений на панели мониторинга;
- st.write() – помогает отображать все возможные детали, например, график, фрейм данных, функции, модель и т. д.;
- st.sidebar() – для отображения данных на боковой панели;
- st.dataframe() – для отображения фрейма данных;
- st.map() – для отображения карты в одной строке кода и т. д.
- st.sidebar.checkbox() - для отображения/скрытия данных;
- st.sidebar.selectbox() – для выбора для отображения параметров.

Пример настройки боковой панели

```
st.sidebar.title("About")
st.sidebar.info(
    """
    This app is Open Source dashboard.
    """
)
```

Пример настройки чек-бокса

```
show_data = st.sidebar.checkbox('Show raw data')
if show_data == True:
    st.subheader('Raw data')
    st.markdown("Пример")
    st.write(df)
```

Самостоятельное задание

1. В качестве источника данных для анализа вакансий, используйте данные парсинга сайта hh.ru по запросу «Аналитик», в 4-х городах (Москва, Санкт-Петербург, Екатеринбург, Новосибирск).
2. Создайте, используя pandas, вычисляемый столбец – «Зарплата» (среднее значение по столбцам “salary.from”, “salary.to”, умноженное на коэффициент 1 (“salary.gross” = False) или 0.87 (“salary.gross” = True))
3. Создайте, используя pandas, вычисляемый столбец – «Квалификация» («Сеньор» ((“experience.name ” = “Более 6 лет” или “От 3 до 6 лет”) и (“name” содержит “Senior” или “Ведущий” или “Старший”)), «Миддл» ((“experience.name ” = “От 1 года до 3

лет”), «Джун» ((“experience.name ” = “Нет опыта”) и (“name” содержит “начинающий” или “стажер” или “помощник”), «Другое», содержащее все остальные вакансии)

4. С помощью plotly.express построить круговую диаграмму, показывающую долю вакансий в разных городах.
5. С помощью plotly.express построить диаграмму рассеяния, показывающую зависимость зп от числа лет опыта.
6. С помощью plotly.express построить ящичковую диаграмму, показывающую зависимость зп от города.
7. С помощью plotly.express построить гистограмму, показывающую распределение зп
8. С помощью plotly.graph_objects построить тепловую карту, показывающую среднее зп для города и квалификации.
9. Создайте приложение streamlit, показывающее гистограмму распределения чистой зп.
10. Обновите предыдущее приложение, добавив чек-бокс, для скрытия и показа данных по отдельным городам.