**Лабораторная работа № 8**

**Тема: «Базовая визуализация на языке Python с помощью библиотеки bokeh»**

**Цель работы**: изучить базовые способы визуализации данных в библиотеке bokeh.

**Теоретическая справка**

Python Bokeh - это библиотека визуализации данных, которая предоставляет интерактивные диаграммы и графики графиков. Bokeh отображает свои графики с использованием HTML и JavaScript, которые используют современные веб-браузеры для представления элегантного, лаконичного построения новой графики с высоким уровнем интерактивности.

Bokeh.models

Этот класс представляет собой библиотеку Python для Bokeh, которая содержит классы моделей, обрабатывающие данные JSON, созданные библиотекой JavaScript Bokeh (BokehJS). Большинство моделей очень просты и состоят из очень небольшого количества атрибутов или вообще не содержат методов.

Bokeh.plotting

Это интерфейс среднего уровня, который предоставляет функции, подобные Matplotlib или MATLAB, для построения графиков. Он имеет дело с данными, которые должны быть нанесены на график, и создает допустимые оси, сетки и инструменты. Основным классом этого интерфейса является класс Figure.

**output\_file('filename.html')** запишет визуализацию в статический HTML-файл.

**output\_notebook()** отобразит визуализацию непосредственно в записной книжке Jupyter.

Объект figure() - это не только основа визуализации данных, но и объект, который открывает доступ ко всем доступным инструментам Bokeh для визуализации данных. Фигура Боке - это подкласс объекта Bokeh Plot, который предоставляет множество параметров, позволяющих настроить эстетические элементы этой области рисования.

Ниже пример параметров

from bokeh.io import output\_notebook

from bokeh.plotting import figure, show

output\_notebook()

fig = figure(background\_fill\_color='gray',

background\_fill\_alpha=0.5,

border\_fill\_color='blue',

border\_fill\_alpha=0.25,

plot\_height=400,

plot\_width=600,

h\_symmetry=True,

x\_axis\_label='X Label',

x\_axis\_type='datetime',

x\_axis\_location='above',

x\_range=('2022-01-01', '2022-12-31'),

y\_axis\_label='Y Label',

y\_axis\_type='linear',

y\_axis\_location='left',

y\_range=(0, 100),

title='Example Figure',

title\_location='right',

toolbar\_location='below',

tools='save')

Глиф - это векторизованная графическая фигура или маркер, который используется для представления данных, например, круг или квадрат.

Пример создания круговой диаграммы в bokeh

from bokeh.plotting import output\_notebook, Figure, show, figure

from bokeh.transform import cumsum

data['angle'] = data['value']/data['value'].sum() \* 2\*pi

data['color'] = Category20c[len(x)]

p = figure(height=350, title="Pie Chart", toolbar\_location=None,

tools="hover", tooltips="@status: @value", x\_range=(-0.5, 1.0))

p.wedge(x=0, y=1, radius=0.4, start\_angle=cumsum('angle', include\_zero=True), end\_angle=cumsum('angle'), line\_color="white", fill\_color='color', legend\_field='status', source=data)

show(p)

ColumnDataSource является основополагающим при передаче данных глифам, которые используются для визуализации. Его основная функциональность заключается в сопоставлении имен со столбцами ваших данных. Это упрощает обращение к элементам данных при построении визуализации.

ColumnDataSource может интерпретировать три типа объектов данных:

* Python dict: Ключи - это имена, связанные с соответствующими последовательностями значений (списками, массивами и так далее).
* Pandas DataFrame: Столбцы фрейма данных становятся ссылочными именами для ColumnDataSource.
* Pandas groupby: Столбцы ColumnDataSource ссылаются на столбцы, которые видны при вызове groupby.describe().

Ниже представлен пример использования ColumnDataSource для построения линейных графиков:

from bokeh.plotting import figure, show

from bokeh.io import output\_file

from bokeh.models import ColumnDataSource

output\_file('races.html',

title='гонки)

rockets\_data = west\_top\_2[west\_top\_2['teamAbbr'] == 'HOU']

warriors\_data = west\_top\_2[west\_top\_2['teamAbbr'] == 'GS']

rockets\_cds = ColumnDataSource(rockets\_data)

warriors\_cds = ColumnDataSource(warriors\_data)

fig = figure(x\_axis\_type='datetime',

plot\_height=300, plot\_width=600,

title='гонки, 2017-18',

x\_axis\_label='Дата', y\_axis\_label='Победы',

toolbar\_location=None)

fig.step('stDate', 'gameWon',

color='#CE1141', legend='Rockets',

source=rockets\_cds)

fig.step('stDate', 'gameWon',

color='#006BB6', legend='Warriors',

source=warriors\_cds)

fig.legend.location = 'top\_left'

show(fig)

Ниже представлен пример добавления подписи к точкам на диаграмме рассеяния с помощью инструмента Bokeh HoverTool

from bokeh.models import HoverTool

tooltips = [

('Player','@name'),

('Three-Pointers Made', '@play3PM'),

('Three-Pointers Attempted', '@play3PA'),

('Three-Point Percentage','@pct3PM{00.0%}'),

]

fig.add\_tools(HoverTool(tooltips=tooltips))

show(fig)

**Самостоятельное задание**

1. Импортировать данные lab8.xlsx в соответствии с кодом:

|  |
| --- |
| import pandas as pd  df=pd.read\_excel(' lab8.xlsx) |

1. Подготовить данные – создать столбец со средней зарплатой между salary.from и salary.to. Если в наличии только один предел – взять его в качестве оценки зп для этой вакансии.
2. Подготовить данные – выделить из названий вакансий новый столбец со значениями джун, мидл, сеньор, использовать либо встроенные строковые функции Python либо регулярные выражения. Учесть и русское и английское написание этих терминов.
3. Подготовить данные – оставить только вакансии с указанием средней зп. Очищенный таким образом датафрейм использовать в последующих заданиях.
4. Создать пузырьковую диаграмму в bokeh где по оси x-город, по оси y-(джун, мидл, сеньор), размер пузырька - число вакансий, цвет пузырька - (чем более светлый, тем больше медианная зп, чем более темный-тем меньше)
5. Создать горизонтальную столбиковую диаграмму, по оси ординат – город, по оси абсцисс – число вакансий в городе.
6. Создать круговую диаграмму – число вакансий с налогами и без (gross=True)
7. Создать ящичковую диаграмму – распределение сумм зарплат по городам.
8. Добавить названия диаграмм для 5-8. Добавить всплывающую подпись (город или наличие налога, либо квалификация специалиста) для каждого примитива на диаграммах 5-8.
9. Нарисовать в bokeh домик, с треугольной крышей и окном, , стены сделать белого цвета, крышу - красного. (делается без данных выборки)