|  |
| --- |
|  |

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский ядерный университет «МИФИ»

**Обнинский институт атомной энергетики –**

филиал федерального государственного автономного образовательного учреждения высшего

образования "Национальный исследовательский ядерный университет «МИФИ»

(ИАТЭ НИЯУ МИФИ)

Отделение интеллектуальных кибернетических систем

**Лабораторная работа № 3**

**”Средства визуализации данных в языках программирования”**

Выполнил:

студент гр. ИС-М18 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кастыря Н. А.

Принял:

Д.т.н., профессор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сальников Н.Л.

Обнинск, 2018 г

**Цель работы:** выбрать интересующий язык программирования и изучить средства визуализации данных.

**Выполнение работы:**

Для выполнения лабораторной работы была выбрана библиотека Seaborn. Seaborn - по сути более высокоуровневое API на базе библиотеки matplotlib. В качестве языка программирования выступает Python, для которого собственно и написана библиотека.

Для начала импортируем библиотеку в проект командой *import seaborn as sns.* При отсутствии библиотеки установить в Python через терминал командой *pip install seaborn.*

Перейдём непосредственно к рассмотрению возможностей Seaborn.

1. Построение тепловой карты.

Построим тепловую карту по типовым данным Python о полётах.

Загружаем набор данных:

*flights = sns.load\_dataset("flights")*

Реорганизуем данные в виде кластеризованной таблицы:

*flights = flights.pivot("month", "year", "passengers")*

Создаём тепловую карту:

*ax = sns.heatmap(flights)*

Отображаем на экране:

*plt.show()*

Можно подписать в каждую ячейку количество пассажиров: ax = *sns.heatmap(flights, annot=****True****, fmt="d")*. Добавить строки между промежутками: *ax = sns.heatmap(flights, linewidths=.5).* Использовать другую цветовую схему: *ax = sns.heatmap(flights, cmap=****"YlGnBu"****)*

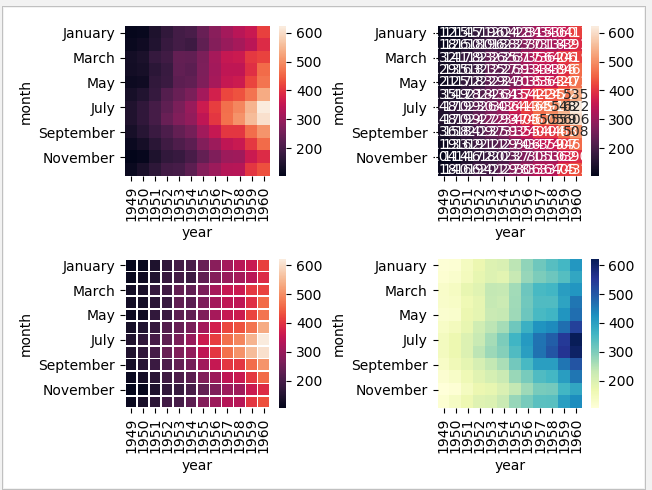


Рисунок 1. Тепловая карта

2.Построение гистограммы.

По рандомному набору значений постоим гистограмму c помощью функции *distplot*:

*sns.set(); np.random.seed(0)  
 x = np.random.randn(100)  
 ax = sns.distplot(x)*

Можно добавить название осей:

*x = pd.Series(x, name=****"x variable"****)  
 ax = sns.distplot(x)*

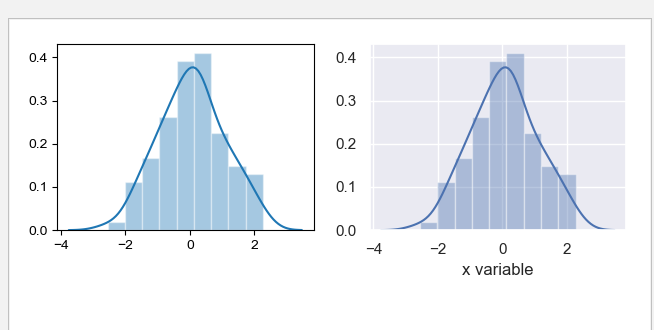


Рисунок 2. Гистограмма

3. Визуализация временного ряда

С помощь функции *lineplot*  по тестовым данным отобразим значение сигнала в разных регионам и с разным типом событий:

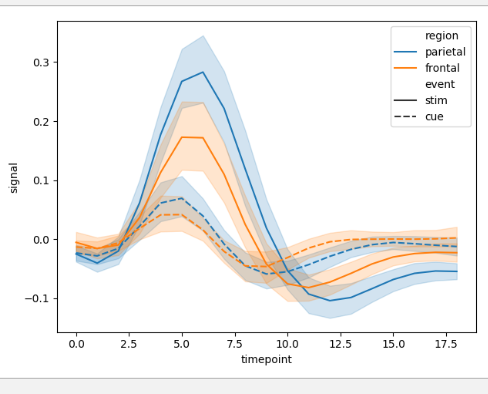
*sns.lineplot(x="timepoint", y="signal", hue="region", style="event"**,* data=fmri)

Рисунок 3. Визуализация временного ряда

4. Диаграмма размаха.

С помощью функции *boxplot* построим диаграммы размаха, то есть диаграмму, показывающую максимальные и минимальные значение, а также медиану.

Загружаем тестовый набор данных по чаевым:

*tips = sns.load\_dataset("tips")*

Диаграмма размаха для одной переменной (итоговый счёт):

*ax = sns.boxplot(x=tips["total\_bill"])*

Диаграмма размаха для двух переменной (итоговый счёт в зависимости от дня):

*ax = sns.boxplot(x="day", y="total\_bill", data=tips)*

Диаграмма размаха для двух переменной c разбиением на категории (итоговый счёт в зависимости от дня по курящим):

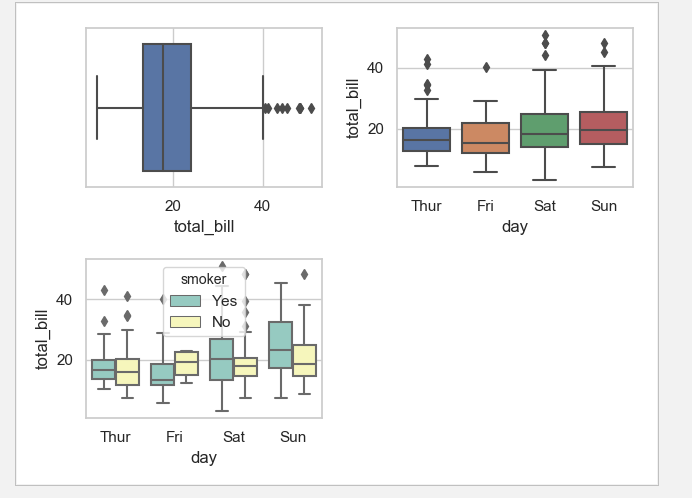
*ax = sns.boxplot(x="day", y="total\_bill", hue="smoker",data=tips, palette="Set3")*

Рисунок 4. Диаграммы размаха

5. Диаграмма рассеивания

Строятся через команду *scatterplot.*

Диаграммы размаха для двух переменной :

*ax = sns.scatterplot(x="day", y="total\_bill", data=tips)*

*ax = sns.scatterplot(x="tip", y="total\_bill", data=tips)*

Диаграмма размаха для двух переменной c разбиением на категории:

*ax = sns.scatterplot(x="total\_bill", y="tip", hue="time",data=tips)*

Диаграмма размаха для двух переменной c разбиением на 2 категории:

*ax = sns.scatterplot(x="total\_bill", y="tip", hue="day", style="time", data=tips)*

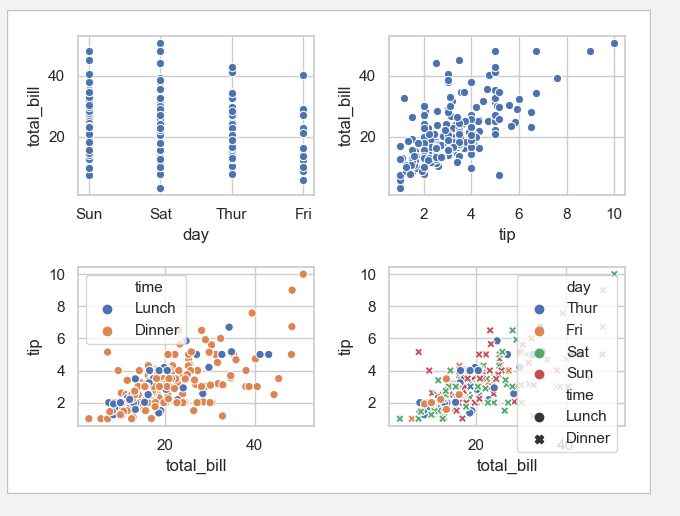


Рисунок 5. Диаграммы рассеивания

**Вывод:**

В результате выполнения работы была изучена часть возможностей библиотеки Seaborn. Seaborn отлично подойдёт для отображения статистических графиков. На мой взгляд достойною конкуренцию Seaborn сможет составить язык R.

**Список использованных источников**

1. Документация по Seaborn [Электронный ресурс]. URL: https://seaborn.pydata.org [Доступ от 13.12.2018]