Язык Python. Часть 2

Лекция 8

Визуализация

- Когда много данных, хочется их визуализировать
- Есть отдельные библиотеки
- Классическая - matplotlib
- Более стильная - seaborn
- В pandas встроена поддержка matplotlib

Скачаем dataset

- https://openaq.org
- OpenAir quality
- Нам нужен один csv-файл
- https://raw.githubusercontent.com/pandas-dev/pandas/main/doc/data/air_quality_no2.csv

Определенная последовательность действий

- Устанавливаем matplotlib (один раз)
- Импортируем `import matplotlib.pyplot as plt`
- Создаем DataFrame
- Вызываем метод plot
- Он готовит визуализацию
- Но не рисует
- Для прорисовки - plt.show()

Пример на визуализацию

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 air_quality = pd.read_csv("air_quality_no2.csv", index_col=0
5 air_quality.plot()
6 plt.show()
```

scatterplot

- Популярная форма визуализации
- Берем два столбца
- С числовыми значениями
- Каждую строчку изображаем как точку
- Один столбец - левая координата
- Другой - правая

Пример на scatterplot

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 air_quality = pd.read_csv("air_quality_no2.csv", index_col=0
5 air_quality.plot.scatter(x="station_london", y="station_pari
6 plt.show()
```

"Ящики с усами"

- Визуализицация числовых показателей
- Показывает диапазон основных значений
- Среднее
- И выбросы (аномальные значения)

Пример на "ящик с усами"

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 air_quality = pd.read_csv("air_quality_no2.csv", index_col=0
5 air_quality.boxplot(x="station_london", y="station_paris", a
6 plt.show()
```

Отдельная визуализация по параметрам

- Мы вызывали plot и получали визуализацию всего сразу
- Можно вызвать area над свойством plot
- И настроить другой режим
- С отдельным графиком для каждого показателя

Пример на отдельную визуализацию

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 air_quality = pd.read_csv("air_quality_no2.csv", index_col=0
5 air_quality.plot.area(figsize=(12, 4), subplots=True)
6 plt.show()
```

Побольше настроек

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 air_quality = pd.read_csv("air_quality_no2.csv", index_col=0
5 fig, axs = plt.subplots(figsize=(12, 4))
6 air_quality.plot.area(ax=axs)
7 axs.set_ylabel("NO$_2$ concentration")
8 fig.savefig("no2_concentrations.png")
9 plt.show()
```

Создание новых колонок

- Можно взять существующую колонку
- Применить к ней операцию
- Например, умножить на число
- Или поэлементно вычесть другую колонку
- И присвоить результат новой колонке

Пример кода

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 air_quality = pd.read_csv("air_quality_no2.csv", index_col=0
5 air_quality["ratio_paris_antwerp"] = air_quality["station_pa
6 air_quality["london_mg_per_cubic"] = air_quality["station_lo
7 air_quality.head()
```

Статистика

- Помимо базовой, которая уже была
- С группировкой по полю
- Группируем все строки по значению одного поля
- С одинаковым значением по данному полю
- И считаем среднюю, медиану и т.п.

Пример кода

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 titanic = pd.read_csv("titanic.csv")
5 print(titanic[["Sex", "Age"]].groupby("Sex").mean())
6 print(titanic.groupby("Sex").mean(numeric_only=True))
7 print(titanic.groupby("Sex")["Age"].mean())
8 print(titanic.groupby(["Sex", "Pclass"])["Fare"].mean())
```

Количество значений по категориям

- value_counts
- Фактически это сокращение
- Для группировки по полю
- С последующей вырезкой этого поля
- И вызовом count

Пример кода

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 titanic = pd.read_csv("titanic.csv")
5 print(titanic["Pclass"].value_counts())
6 print(titanic.groupby("Pclass")["Pclass"].count())
```

Сортировка

- DataFrame можно сортировать
- По разным полям
- По одному или по нескольким
- По возрастанию или по убыванию

Пример на сортировку

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 titanic = pd.read_csv("titanic.csv")
5 print(titanic.sort_values(by="Age").head())
6 print(titanic.sort_values(by=['Pclass', 'Age'], ascending=Fa
```

Еще пример

- Хотим взять данные по NO2
- И хотим взять первые два измерения по каждой локации
- Используем понятие индекс
- https://raw.githubusercontent.com/pandas-dev/pandas/main/doc/data/air_quality_long.csv

Чтобы два раза не вставать

Еще пример

- https://raw.githubusercontent.com/pandas-dev/pandas/main/doc/data/air_quality_no2_long.cs
- https://raw.githubusercontent.com/pandas-dev/pandas/main/doc/data/air_quality_pm25_long.
- https://raw.githubusercontent.com/pandas-dev/pandas/main/doc/data/air_quality_stations.csv
- https://raw.githubusercontent.com/pandas-dev/pandas/main/doc/data/air_quality_parameters.

Пример на сортировку

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 air_quality = pd.read_csv("air_quality_long.csv", index_col=
5 no2 = air_quality[air_quality["parameter"] == "no2"]
6 no2_subset = no2.sort_index().groupby(["location"]).head(2)
7 print(no2_subset)
```

Разбивка по колонке

- Есть таблица с индексной колонкой
- Например, момент измерения
- И есть колонка с повторяющимся значением
- Например, место измерения
- И в разных местах мы делаем много измерений
- Преимущественно, в одно время

Разбивка по колонке

- Хотим получить сводную табличку
- Чтобы слева в колонку шли моменты измерения
- И было по колонке на каждое место
- И в ячейках - значения
- NaN, если значения нет

Пример на pivot

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 air_quality = pd.read_csv("air_quality_long.csv", index_col=
5 no2 = air_quality[air_quality["parameter"] == "no2"]
6 no2_subset = no2.sort_index().groupby(["location"]).head(2)
7 print(no2_subset.pivot(columns="location", values="value"))
```

Пример на pivot с визуализацией

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 air_quality = pd.read_csv("air_quality_long.csv", index_col=
5 no2 = air_quality[air_quality["parameter"] == "no2"]
6 no2.pivot(columns="location", values="value").plot()
7 plt.show()
```

pivot_table

- Иногда при разбивке на одну ячейку приходится много значений
- Например, если индекс - место измерения
- А колонки - измеряемый показатель
- Значений много - по разным моментам
- Часто ожидается агрерирование (min, max, average, ...)

Пример на pivot_table

```
 1 import pandas as pd
 2 import matplotlib.pyplot as plt
 3
 4 air_quality = pd.read_csv("air_quality_long.csv", index_col
 5 print(air_quality.pivot_table(
 6     values="value", index="location", columns="parameter",
 7 ))
 8 print(air_quality.pivot_table(
 9     values="value",
10     index="location",
11     columns="parameter",
12     aggfunc="mean",
13     margins=True,
14 ))
```