Лабораторная работа №28 (2 часа)

Тема работы: «Разработка, отладка и испытание программ, анализа данных над объектами библиотеки Pandas»

1 Цель работы

Закрепить навык работы с объектами Series, DataFrame, Index пакета Pandas

2 Задание

Поместите данные в объект Series (используйте ассоциированные метки) произведите поиск информации с помощью меток.

Сгруппируйте данные, которые находятся в объекте Series, и поместите их в объект DataFrame. Создайте таблицу. Выполните агрегирование данных.

3 Оснащение работы

Задание по варианту, ЭВМ, среда разработки **Python 3.7, IDLE**.

4 Основные теоретические сведения

Pandas – это высокоуровневая Python библиотека для анализа данных. Почему я её называю высокоуровневой, потому что построена она поверх более низкоуровневой библиотеки NumPy (написана на Си), что является большим плюсом в производительности. В экосистеме Python, pandas является наиболее продвинутой и быстроразвивающейся библиотекой для обработки и анализа данных. В своей работе мне приходится пользоваться ею практически каждый день, поэтому я пишу эту краткую заметку для того, чтобы в будущем ссылаться к ней, если вдруг что-то забуду. Также надеюсь, что читателям блога заметка поможет в решении их собственных задач с помощью pandas, и послужит небольшим введением в возможности этой библиотеки.

Группировка и агрегирование в pandas

Группировка данных один из самых часто используемых методов при анализе данных. В pandas за группировку отвечает метод *.groupby*. Стандартный набор данных (dataset), использующийся во всех курсах про анализ данных — данные о пассажирах Титаника. Скачать CSV файл можно тут.

```
>>> titanic_df = pd.read_csv('titanic.csv')
>>> print(titanic_df.head())
   PassengerID                                             Name PClass    Age  \
0            1                    Allen, Miss Elisabeth Walton    1st  29.00
1            2                   Allison, Miss Helen Loraine    1st   2.00
2            3            Allison, Mr Hudson Joshua Creighton    1st  30.00
3            4  Allison, Mrs Hudson JC (Bessie Waldo Daniels)    1st  25.00
4            5                  Allison, Master Hudson Trevor    1st   0.92
      Sex  Survived  SexCode
```

```
0  female       1       1
1  female       0       1
2    male       0       0
3  female       0       1
4    male       1       0
```

Необходимо подсчитать, сколько женщин и мужчин выжило, а сколько нет. В этом нам поможет метод *.groupby*.

```
>>> print(titanic_df.groupby(['Sex', 'Survived'])['PassengerID'].count())
Sex     Survived
female  0          154
        1          308
male    0          709
        1          142
Name: PassengerID, dtype: int64
```

А теперь проанализируем в разрезе класса кабины:

```
>>> print(titanic_df.groupby(['PClass', 'Survived'])['PassengerID'].count())
PClass  Survived
*       0            1
1st     0          129
        1          193
2nd     0          160
        1          119
3rd     0          573
        1          138
Name: PassengerID, dtype: int64
```

Сводные таблицы в pandas

Термин «сводная таблица» хорошо известен тем, кто не по наслышке знаком с инструментом Microsoft Excel или любым иным, предназначенным для обработки и анализа данных. В pandas сводные таблицы строятся через метод *.pivot_table*. За основу возьмём всё тот же пример с Титаником. Например, перед нами стоит задача посчитать сколько всего женщин и мужчин было в конкретном классе корабля:

```
>>> titanic_df = pd.read_csv('titanic.csv')
>>> pvt = titanic_df.pivot_table(index=['Sex'], columns=['PClass'], values='Name', aggfunc='count')
```

В качестве индекса теперь у нас будет пол человека, колонками станут значения из PClass, функцией агрегирования будет count (подсчёт количества записей) по колонке Name.

```
>>> print(pvt.loc['female', ['1st', '2nd', '3rd']])
PClass
1st    143.0
2nd    107.0
3rd    212.0
```

Name: female, dtype: float64
Всё очень просто.

Анализ временных рядов

В pandas очень удобно анализировать временные ряды. В качестве показательного примера я буду использовать цену на акции корпорации Apple за 5 лет по дням. Файл с данными можно скачать тут.

```
>>> import pandas as pd
>>> df = pd.read_csv('apple.csv', index_col='Date', parse_dates=True)
>>> df = df.sort_index()
>>> print(df.info())
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1258 entries, 2017-02-22 to 2012-02-23
Data columns (total 6 columns):
Open        1258 non-null float64
High        1258 non-null float64
Low         1258 non-null float64
Close       1258 non-null float64
Volume      1258 non-null int64
Adj Close   1258 non-null float64
dtypes: float64(5), int64(1)
memory usage: 68.8 KB
```

Здесь мы формируем DataFrame с DatetimeIndex по колонке Date и сортируем новый индекс в правильном порядке для работы с выборками. Если колонка имеет формат даты и времени отличный от ISO8601, то для правильного перевода строки в нужный тип, можно использовать метод pandas.to_datetime.

Давайте теперь узнаем среднюю цену акции (mean) на закрытии (Close):

```
>>> df.loc['2012-Feb', 'Close'].mean()
528.4820021999999
```

А если взять промежуток с февраля 2012 по февраль 2015 и посчитать среднее:

```
>>> df.loc['2012-Feb':'2015-Feb', 'Close'].mean()
430.43968317018414
```

А что если нам нужно узнать среднюю цену закрытия по неделям?!

```
>>> df.resample('W')['Close'].mean()
Date
2012-02-26    519.399979
2012-03-04    538.652008
2012-03-11    536.254004
2012-03-18    576.161993
2012-03-25    600.990001
2012-04-01    609.698003
2012-04-08    626.484993
2012-04-15    623.773999
```

2012-04-22	591.718002
2012-04-29	590.536005
2012-05-06	579.831995
2012-05-13	568.814001
2012-05-20	543.593996
2012-05-27	563.283995
2012-06-03	572.539994
2012-06-10	570.124002
2012-06-17	573.029991
2012-06-24	583.739993
2012-07-01	574.070004
2012-07-08	601.937489
2012-07-15	606.080008
2012-07-22	607.746011
2012-07-29	587.951999
2012-08-05	607.217999
2012-08-12	621.150003
2012-08-19	635.394003
2012-08-26	663.185999
2012-09-02	670.611995
2012-09-09	675.477503
2012-09-16	673.476007
...	
2016-08-07	105.934003
2016-08-14	108.258000
2016-08-21	109.304001
2016-08-28	107.980000
2016-09-04	106.676001
2016-09-11	106.177498
2016-09-18	111.129999
2016-09-25	113.606001
2016-10-02	113.029999
2016-10-09	113.303999
2016-10-16	116.860000
2016-10-23	117.160001
2016-10-30	115.938000
2016-11-06	111.057999
2016-11-13	109.714000
2016-11-20	108.563999
2016-11-27	111.637503
2016-12-04	110.587999
2016-12-11	111.231999
2016-12-18	115.094002
2016-12-25	116.691998

2017-01-01 116.642502
2017-01-08 116.672501
2017-01-15 119.228000
2017-01-22 119.942499
2017-01-29 121.164000
2017-02-05 125.867999
2017-02-12 131.679996
2017-02-19 134.978000
2017-02-26 136.904999
Freq: W-SUN, Name: Close, dtype: float64

Resampling мощный инструмент при работе с временными рядами (time series), помогающий переформировать выборку так, как удобно вам. Метод resample первым аргументом принимает строку rule. Все доступные значения можно найти в документации.

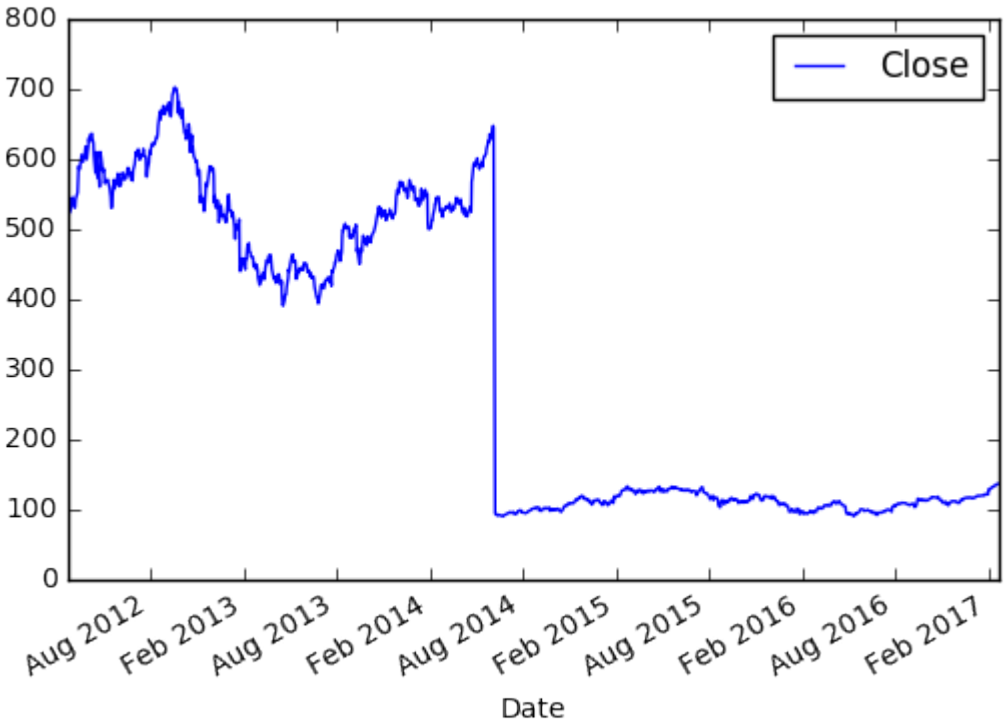
Визуализация данных в pandas

Для визуального анализа данных, pandas использует библиотеку matplotlib. Продемонстрирую простейший способ визуализации в pandas на примере с акциями Apple.

Берём цену закрытия в промежутке между 2012 и 2017.

```
>>> import matplotlib.pyplot as plt
>>> new_sample_df = df.loc['2012-Feb':'2017-Feb', ['Close']]
>>> new_sample_df.plot()
>>> plt.show()
```

И видим вот такую картину:



5 Порядок выполнения работы

1. Выделить ключевые моменты задачи.
2. Построить алгоритм решения задачи.
3. Запрограммировать полученный алгоритм.
4. Провести тестирование полученной программы.

6 Форма отчета о работе

Лабораторная работа № ___
Номер учебной группы _____
Фамилия, инициалы учащегося: _____
Дата выполнения работы: _____
Тема работы: _____
Цель работы: _____
Оснащение работы: _____
Результат выполнения работы: _____
_____

7 Контрольные вопросы и задания

1. Какое назначение у библиотеки Pandas?
2. Опишите инструменты визуализации библиотеки
3. Опишите процесс анализа временных рядов
4. Как создать таблицу?
5. Что такое агрегирование?

8 Рекомендуемая литература

Плас, Дж. В. Python для сложных задач. Наука о данных и машинное обучение / Дж.В. Плас. – СПб: Питер, 2018.

Прохоренок, Н.А. Python 3. Самое необходимое / Н.А Прохоренок, В.А. Дронов – СПб.: БВХ-Петербург, 2016.