АКАДЕМИЯ АЙТИ

День 24

academyit.ru

Работа с файлами и каталогами. Основные операции с путями к файлам. Импорт пакета. Важнейшие стандартные пакеты. Подсистема pip . Установка стороннего модуля. Создание собственных модулей.

Турашова Анна Николаевна
Преподаватель
anna1turashova@gmail.com
Telegram: @anna1tur

Проверка домашнего задания

Задание 1.

С помощью модуля os создайте папку в директории вашего проекта. Затем добавьте в неё несколько файлов и папок. Попробуйте переименовать файлы, скопировать.
Заархивируйте созданную папку.

В конце удалите всё созданное с помощью модуля os.

Pandas

```
import pandas as pd
```

```
l = [['Anna', 23, 3],
     ['Sam', 36, 10],
     ['Bill', 33, 10],
     ['Moica', 23, 7],
     ['Anna', 27, 7],
     ['Peter', 32, None]]
```

```
df = pd.DataFrame(l)
```

```
df
```

	0	1	2
0	Anna	23	3.0
1	Sam	36	10.0
2	Bill	33	10.0
3	Moica	23	7.0
4	Anna	27	7.0
5	Peter	32	NaN

```
type(df)
```

```
pandas.core.frame.DataFrame
```

```
df[1]
```

```
0    23
1    36
2    33
3    23
4    27
5    32
Name: 1, dtype: int64
```

```
type(df[1])
```

```
pandas.core.series.Series
```

Изменение наименований колонок и столбцов

```
df.columns = ['name', 'age', 'expr']
```

```
df.index = df["name"]
```

Обращение по индексу df.iloc[...]

```
df.iloc[1, 2]
```

```
df.iloc[1:3]
```

```
df.iloc[1:3, 1]
```

```
df.iloc[:, 0]
```

```
df.iloc[1:3, :]
```

Обращение по имени df.loc[...]

```
df.loc['Bill', 'age']
```

```
df.loc['Sam':'Moica', 'age']
```

```
df.loc['Anna', 'age']
```

```
df.loc['Sam':'Moica', 'age':'expr']
```

```
df.loc[['Sam','Moica'], ['name','expr']]
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 6 entries, Anna to Peter
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   name    6 non-null      object
 1   age     6 non-null      int64
 2   expr    5 non-null      float64
dtypes: float64(1), int64(1), object(1)
memory usage: 364.0+ bytes
```

```
df.shape
```

```
(6, 3)
```

```
df.describe()
```

	age	expr
count	6.000000	5.000000
mean	29.000000	7.400000
std	5.477226	2.880972
min	23.000000	3.000000
25%	24.000000	7.000000
50%	29.500000	7.000000
75%	32.750000	10.000000
max	36.000000	10.000000

mean - среднее
std - стандартное отклонение –
отвечает за разброс, всегда > 0
25% 50% 75% - квантили
50% - медиана

Переименование столбцов

```
df2 = df.rename(columns={'age': 'X'})
```

```
df2.rename({'X': 'xxxxx'}, axis=1, inplace=True)
```

Переименование колонок

```
df2.rename({'Anna': 'xxxxx'}, axis=0, inplace=True)
```

	name	xxxxx	exp
name			
xxxxx	Anna	23	3.0
Sam	Sam	36	10.0
Bill	Bill	33	10.0
Moica	Moica	23	7.0
xxxxx	Anna	27	7.0
Peter	Peter	32	NaN

```
df['age'] > 30
```

```
name
Anna      False
Sam        True
Bill       True
Moica     False
Anna      False
Name: age, dtype: bool
```

```
df[df['age'] > 25]
```

name	name	age	exp
Sam	Sam	36	10.0
Bill	Bill	33	10.0
Anna	Anna	27	7.0

```
df['age'] ** 2
```

```
name
Anna       529
Sam       1296
Bill      1089
Moica      529
Anna       729
Name: age, dtype: int64
```

```
df['age'] + 2
```

```
name
Anna      25
Sam       38
Bill      35
Moica     25
Anna      29
Name: age, dtype: int64
```

Создание новых столбцов

```
df['gender'] = [0, 1, 1, 0, 0]
```

```
df['z'] = df['age'] ** 2
```

```
df['W'] = "W"
```

```
df["-"] = df['age'] - df['exp']
```

```
df
```

	name	age	exp	gender	z	W	-
name							
Anna	Anna	23	3.0	0	529	W	20.0
Sam	Sam	36	10.0	1	1296	W	26.0
Bill	Bill	33	10.0	1	1089	W	23.0
Moica	Moica	23	7.0	0	529	W	16.0
Anna	Anna	27	7.0	0	729	W	20.0

Что такое формат CSV

CSV (comma-separated values; значения, разделенные запятыми) – текстовый формат, позволяющий хранить табличные данные

Почему CSV – наиболее популярный формат табличных данных

- Легко читается людьми
- Содержит структурированные данные
- Поддерживается почти всеми системами хранения данных

import pandas as pd

df = pd.read_csv(“name.csv”) – чтение файла

pandas.series.apply – принимает функцию и применяет её к series:
df.loc[:, ‘w1’, ‘w2’].apply(np.mean, axis=1)

df.groupby(‘group’).agg(‘sum’) – группирование, применение sum к группе

df.sort_values(‘total’) – сортировка
df.sort_index() – сортировка по индексу

df.isnull() – маска нулевых значений
df.dropna() – удалить строки с отсутствующими данными
df.fillna(0) – все nan заменяется на 0

df.set_index([‘date’, ‘lang’]) – установить два столбца как индекс

pd.concat([df5, df]) – объединить df

df.to_csv(‘name.csv’) – сохранение df в csv

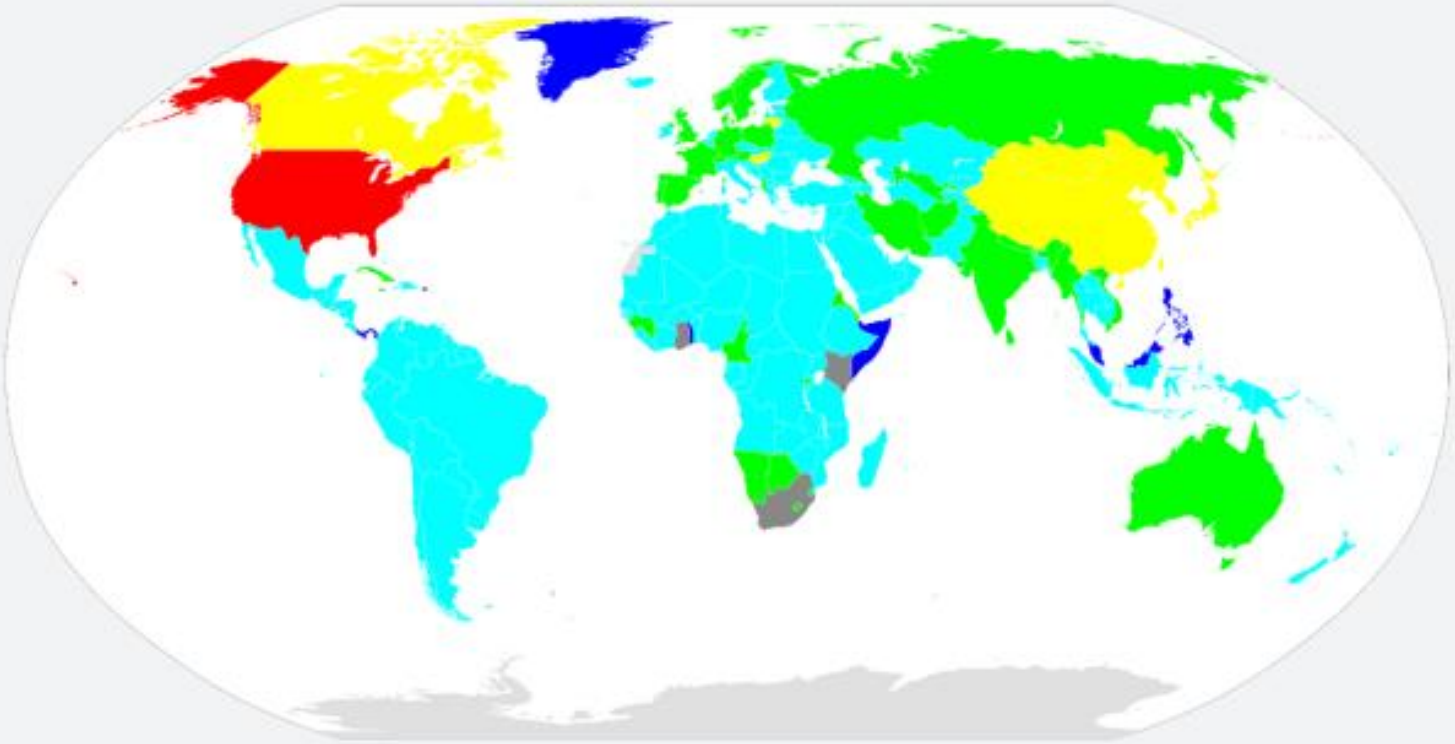
Проблема форматирования дат

22.04.2006

2006.04.22

04/22/2006

04.22.2006



Стандартный формат дат

11/08/12 – ?

Стандарт **ISO 8601:** предписывает записывать даты в следующем виде: **2012-08-11**

Что такое формат XLSX

XLSX – бинарный формат хранения данных Excel

Некоторые особенности формата XLSX:

- Несколько таблиц в одном файле
- Форматирование и объединение ячеек
- Формулы для автоматического вычисления значений ячеек

df = pd.read_excel('name.xlsx') – чтение

DataFrame может содержать только одну таблицу.
Используйте sheet_name='name', чтобы указать имя листа.

df.to_excel('name.xlsx', index_label='index') – запись

Домашнее задание

Задание 1.

Файл Fishing.csv содержит результаты опроса о рыбалке: респонденты, заполняя опросник, подробно описывали свою недавнюю рыбалку.

Описание переменных в датафрейме:
* mode: выбранный тип рыбалки: на берегу (beach), на пирсе (pier), в своей лодке (boat) и в арендованной лодке (charter);
* price: стоимость выбранного типа рыбалки;
* catch: коэффициент улова при выбранном типе рыбалки;
* pbeach: стоимость рыбалки на берегу;
* ppier: стоимость рыбалки на пирсе;
* pboat: стоимость рыбалки на своей лодке;
* pcharter: стоимость рыбалки на арендованной лодке;
* cbeach: коэффициент улова на рыбалке на берегу;
* cpier: коэффициент улова на рыбалке на пирсе;
* cboat: коэффициент улова на рыбалке на своей лодке;
* ccharter: коэффициент улова на рыбалке на арендованной лодке;
* income: доход в месяц.

Подробнее об опросе и исследовании можно почитать в [статье](https://core.ac.uk/download/pdf/38934845.pdf)
J.Herriges, C.Kling "*Nonlinear Income Effects in Random Utility Models*" (1999).

Задание 1.

1) Загрузить таблицу из файла Fishing.csv и сохранить её в датафрейм dat.
Вывести на экран первые 8 строк загруженного датафрейма.

2) Добавить, используя метод .apply(), столбец log_income, содержащий натуральный логарифм доходов респондентов.

3) Посчитать для каждого респондента абсолютное значение отклонения price от pbeach и сохранить результат в столбец pdiff.
Подсказка 1: для нахождения абсолютного значения числа используется функция abs(). Пример:

```
abs(-8)
8
```

Подсказка 2: пример с lamda-функцией в первом уроке этого модуля.

4) Сгруппировать наблюдения в таблице по признаку тип рыбалки (mode) и вывести для каждого типа среднюю цену (price), которую респонденты заплатили за рыбалку.

5) Сгруппировать наблюдения в таблице по признаку тип рыбалки (mode) и вывести для каждого типа разницу между медианным и средним значением цены (price), которую респонденты заплатили за рыбалку.
Посказка: можно написать свою lambda-функцию для подсчёта разницы между медианой и средним и применить её внутри метода для агрегирования. Внимание: название самостоятельно написанной функции будет уже вводиться без кавычек.

Задание 1.

6) Сгруппировать наблюдения в таблице по признаку тип рыбалки (mode) и сохранить полученные датафреймы (один для каждого типа рыбалки) в отдельные csv-файлы. В итоге должно получиться четыре разных csv-файла.
Подсказка: можно запустить следующий код и посмотреть, что получится:

```
for name, data in dat.groupby("mode"):
    print(name, data)
```

7) Отсортировать строки в датафрейме в соответствии со значениями income в порядке убывания таким образом, чтобы результаты сортировки сохранились в исходном датафрейме.

8) Отсортировать строки в датафрейме в соответствии со значениями price и income в порядке возрастания. Можно ли сказать, что люди с более низким доходом и выбравшие более дешёвый тип рыбалки, в целом, предпочитают один тип рыбалки, а люди с более высоким доходом и более дорогой рыбалкой – другой? Ответ записать в виде текстовой ячейки или в виде комментария.

9) Любым известным способом проверить, есть ли в датафрейме пропущенные значения. Если есть, удалить строки с пропущенными значениями. Если нет, написать комментарий, что таких нет.

АПЛАНА
ГРУППА КОМПАНИЙ

Входит в ГК Аплана

АКАДЕМИЯ АЙТИ

Основана в 1995 г.

E-learning и очное обучение

Направления обучения:

- Информационные технологии
- Информационная безопасность
- ИТ-менеджмент и управление проектами
- Разработка и тестирование ПО
- Гос. и муниципальное управление

Филиалы:

Санкт-Петербург, Казань, Уфа, Челябинск, Хабаровск, Красноярск, Тюмень, Нижний Новгород, Краснодар, Волгоград, Ростов-на-Дону

Головной офис в Москве

Разработка программного обеспечения и информационных систем

Ежегодные награды Microsoft, Huawei, Cisco и другие

Программы по импортозамещению

Ресурсы более 400 высококлассных экспертов и преподавателей

Сеть региональных учебных центров по всей России

Крупные заказчики

СБЕР

РЖД

Билайн

TATNEFT

НОРНИКЕЛЬ

100+
сотрудников

АКАДЕМИЯ АЙТИ

Спасибо за внимание!

Центральный офис:
Москва, Варшавское шоссе 47, корп. 4, 7 этаж
Тел: +7 (495) 150-96-00
academy@it.ru
academyit.ru