Python как инструмент анализа данных

Курс «Аналитическое мышление»

Алексей Кузьмин
Директор разработки ДомКлик.ру

Алексей Кузьмин

Директор разработки
ДомКлик.ру

О спикере

- Руковожу направлением работы с данными и Data Science
- Преподаю в Нетологии
- Работаю в IT с 2010 года (ABBYY, ДомКлик)
- Окончил МехМат МГУ в 2012 году

В слаке

@Alexey Kuzmin

Структура курса

План занятия

1. Знакомство с Python
2. Знакомство с Google Colab
3. Базовые инструменты и конструкции в Python
4. Знакомство с Pandas — библиотекой для анализа данных

Python

Что это и зачем?

1

Python

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода.

Основной язык для изучения данных и построения моделей машинного обучения.

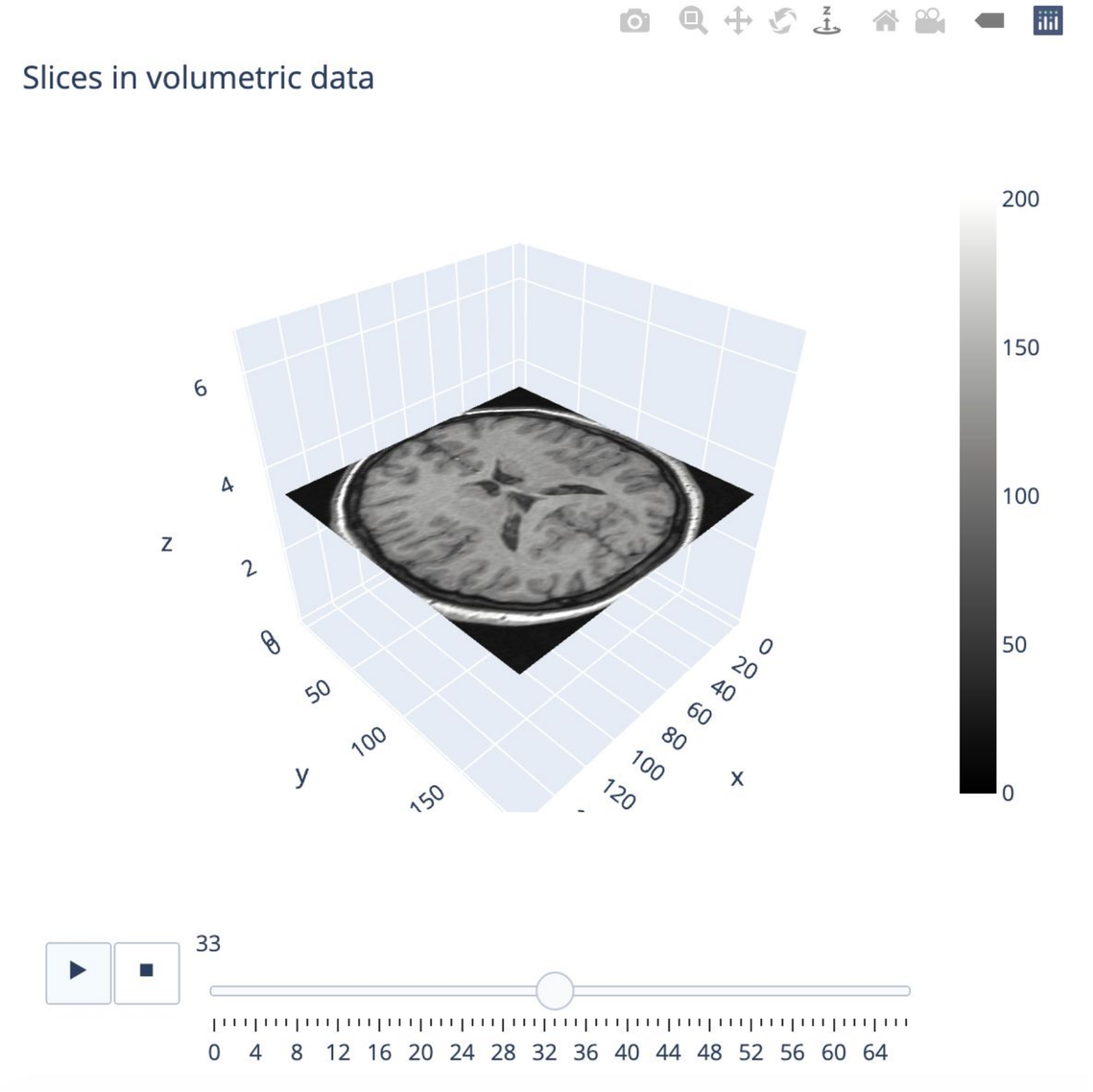
Почему

Python

Простой синтаксис

Python

```
print('Hello world')
```



Библиотеки на все случаи жизни

Обработка данных
Отчеты и презентации
Интерактивные дашборды
Excel, PowerPoint, финансы
Научные вычисления

https://plotly.com/python/visualizing-mri-volume-slices

Работа с большими данными

Самые свежие версии библиотек
Хорошая документация
Большое сообщество

Colab

Что это и зачем нужно

1

2

Google Collaboratory

- Облачная среда для работы с Python
- Бесплатная
- Построчное выполнение кода
- Удобное отображение таблиц и промежуточных результатов

Google Collaboratory

- Ноутбук — файл с кодом
- Состоит из ячеек (код или текст)
- Код — ячейки с кодом на языке Python.
  Можно выполнять в произвольном порядке
- Текст — текстовые комментарии (в формате markdown)

Практика

- Запустим Google Collaboratory
- Создадим новый ноутбук
- Выполним команду print

Синтаксис Python

Учимся программировать

1 2 3

Комментарии

Это текст, который присутствует в коде программы, но игнорируется интерпретатором.

Используются для того, чтобы добавить объяснение для определенного блока кода.

Однострочный комментарий начинается с символа #.

Арифметические операции

сложение (+)

вычитание (-)

умножение (*)

деление (/)

целочисленное деление (//)

возведение в степень (**)

взятие остатка от деления (%)

Переменные

Данные хранятся в ячейках памяти компьютера. Когда мы вводим число, оно помещается в какую-то ячейку памяти.

Переменная — именованная ячейка памяти, хранящая определенное значение.

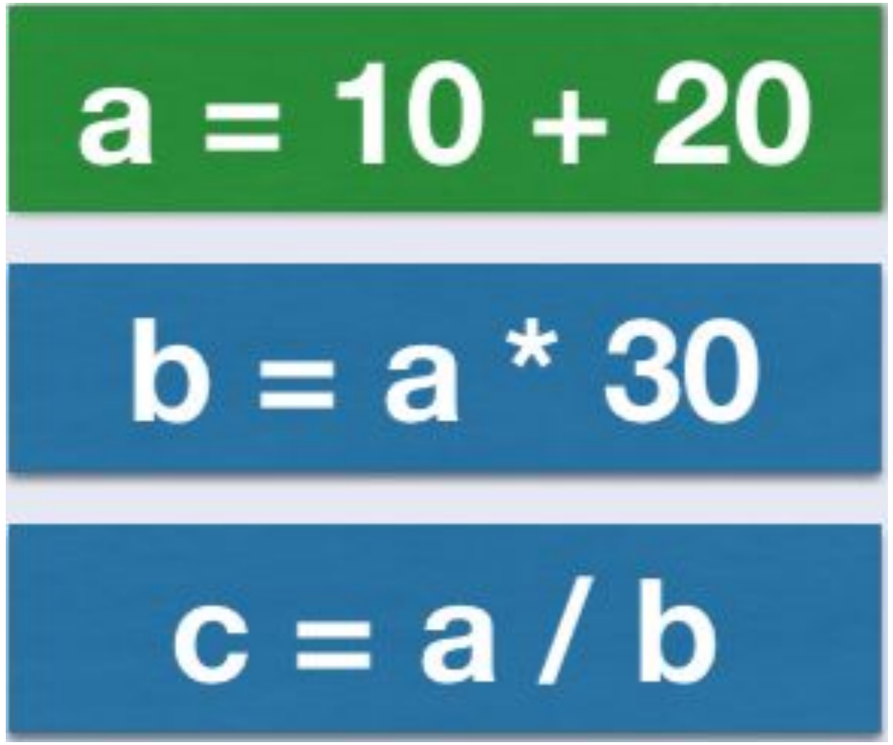
В Python связь между данными и переменными устанавливается с помощью знака = (оператор присваивания).

Переменные

Переменная — это объект, которому дано имя.

В переменных хранятся данные,
промежуточные результаты вычислений.

Объект — это:
- число,
- строка,
- практически что угодно в Python.



Как называть переменные

1. имя переменной может состоять только из цифр, букв и знаков подчеркивания

2. имя переменной не может начинаться с цифры

3. имя переменной должно описывать ее суть

Операторы сравнения

- >
- <
- == (не путать с =)
- >=
- <=
- !=

В результате операций сравнения возвращается булево значение (True / False).

Сравнения могут быть записаны в цепочку.

Операторы сравнения

```
2 > 1
```

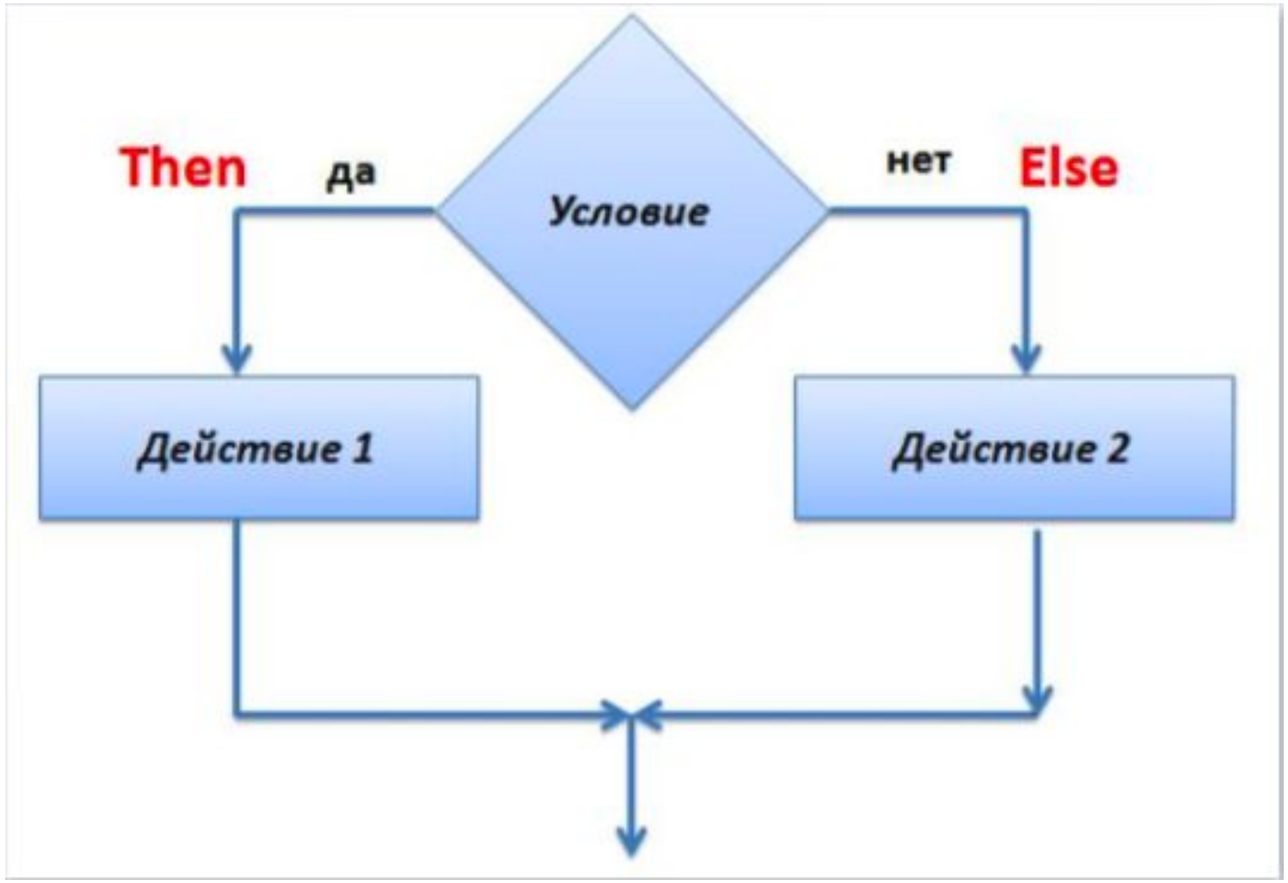
```
True
```

аналог ИСТИНА и ЛОЖЬ в Excel

```
10 < 9
```

```
False
```

Условные конструкции



- способ управлять выполнением программы;
- способ запрограммировать принятие решений;
- логическое выражение: если условие истинно / ложно, после него выполняются те или иные команды.

Условные конструкции

```
a = 1
b = 2

if a < b:
    print('Значение переменной a меньше b')
```

```
Значение переменной a меньше b
```

Множественное ветвление

С помощью if-elif-else можно реализовать несколько отдельных ветвей выполнения в зависимости от условий.

```
age = 6
if  age < 12:
    print("Ребенок")
elif age < 18:
    print("Подросток")
elif age < 50:
    print("Взрослый")
else:
    print("Пожилой")
```

Что куда относится?

В случае истинности условия будет выполнен **«блок»** кода.

Блок определяется наличием отступов.

```
age = 13
print(“начало”)
if  age < 12:
    print("Ребенок")
print(“конец”) <- будет ли напечатано?
```

Что куда относится?

Блок:

```
if  age < 12:
    print("Ребенок")
    print("Конец")
```

<- Будет напечатано, только если возраст меньше 12 лет

Не блок:

```
if  age < 12:
    print("Ребенок")
print(“конец”)
```

<- Будет напечатано в любом случае

Print

Команда print выводит данные на экран.

Она принимает значения, которые будут напечатаны на экране, через запятую.

```
my_name = “Вася”

print(“Мое имя”, my_name)

>>> Мое имя Вася
```

Практика

2

- Введем две переменные: ширина и длина
- Рассчитаем площадь прямоугольника
- Выведем сообщение, если площадь прямоугольника больше 5

Pandas

Варим данные

1 2 3 **4**

Модули

Встроенные в язык программирования функции доступны сразу. Чтобы их вызвать, не надо выполнять никаких дополнительных действий.

Доступ к дополнительным возможностям языка возможен через т.н. модули.

Каждый модуль содержит коллекцию функций и классов, предназначенных для решения задач из определенной области.

Импорт модуля

По умолчанию дополнительные модули не доступны в программе. Для хранения их функций и классов нужно выделять память и т.п.

Для работы с модулем его нужно импортировать в программу. После импорта Python узнает о существовании его функций и классов.

Разные способы импорта:

import pandas

import pandas as pd

from pandas import *

from pandas import DataFrame

Pandas

Модуль Python’a, предназначенный для работы с табличными данными, полученными из различных источников

Загружаем данные

Подключаем модуль pandas

import **pandas** as **pd**

Загружаем данные

```
df = pd.read_csv(
    ‘iris.csv’, sep=’,’
)
```

Загружаем данные

Сюда сохранить результат команды

```
df = pd.read_csv(
    ‘iris.csv’, sep=’,’
)
```

Загружаем данные

Сюда сохранить результат команды

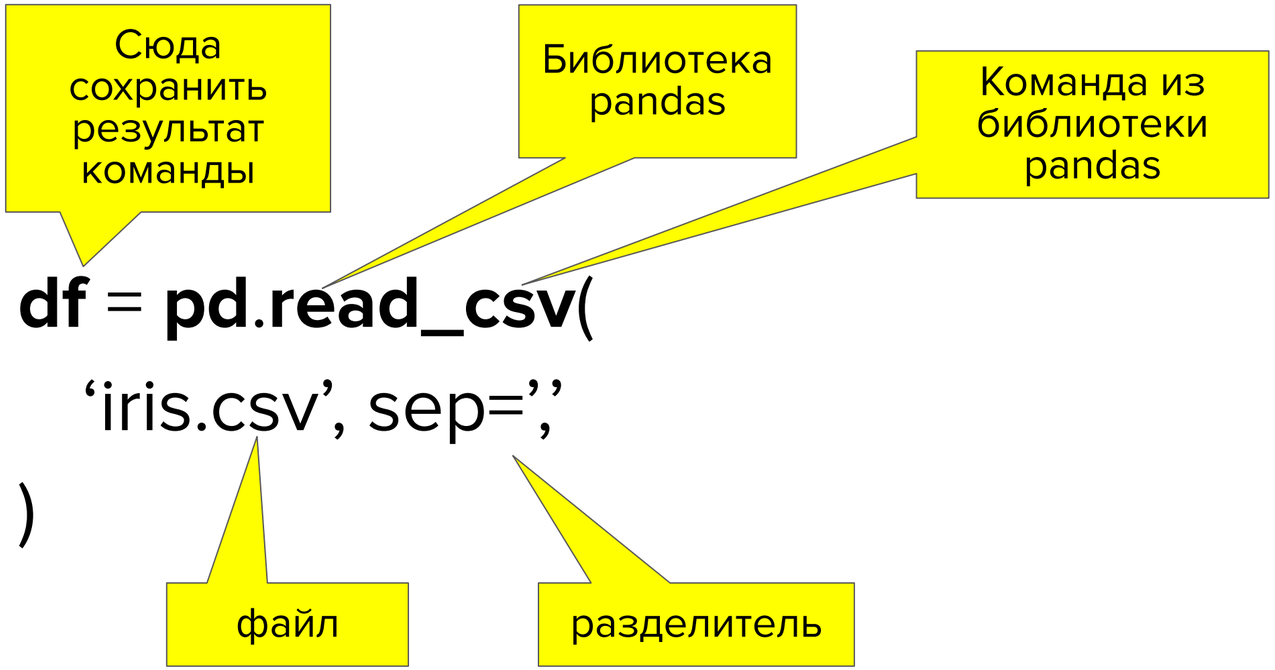
Библиотека pandas

```
df = pd.read_csv(
    ‘iris.csv’, sep=’,’
)
```

Загружаем данные


```
df = pd.read_csv(
    ‘iris.csv’, sep=’,’
)
```

Загружаем данные



Посмотрим, что загрузили

df.head()

Практика

3

- Загрузим приложенный файл iris.csv в google colab
- Подключим библиотеку pandas
- Загрузим в нее данные из приложенного файла
- Посмотрим, что получилось

Что можно делать с DataFrame?

1. Вывести первые несколько строчек
   a. df.head()

2. Узнать количество строк
   a. len(df)

3. Вывести общую статистику
   a. df.describe()
   b. df.info()

Практика

4

- Посмотрим, что как устроен наш dataframe, применив к нему функции с прошлого слайда

Что можно делать с DataFrame?

1. Вывести список колонок
   a. `df.columns`

2. Оставить только некоторые из них
   a. `df[[‘column 1’, ‘column 2’]]` <- в результате получается новый dataframe

3. Сгруппировать данные и вывести сводную информацию
   a.
```
df[['variety',
'sepal_width']].groupby(['variety']).mean()
```
   b. Другие функции агрегации: sum, min, max, count

Практика

5

- Применим функции с прошлого слайда

Что можно python?

Пример анализа данных на Python

Экспорт данных из Google таблиц

1 2 3 4 5

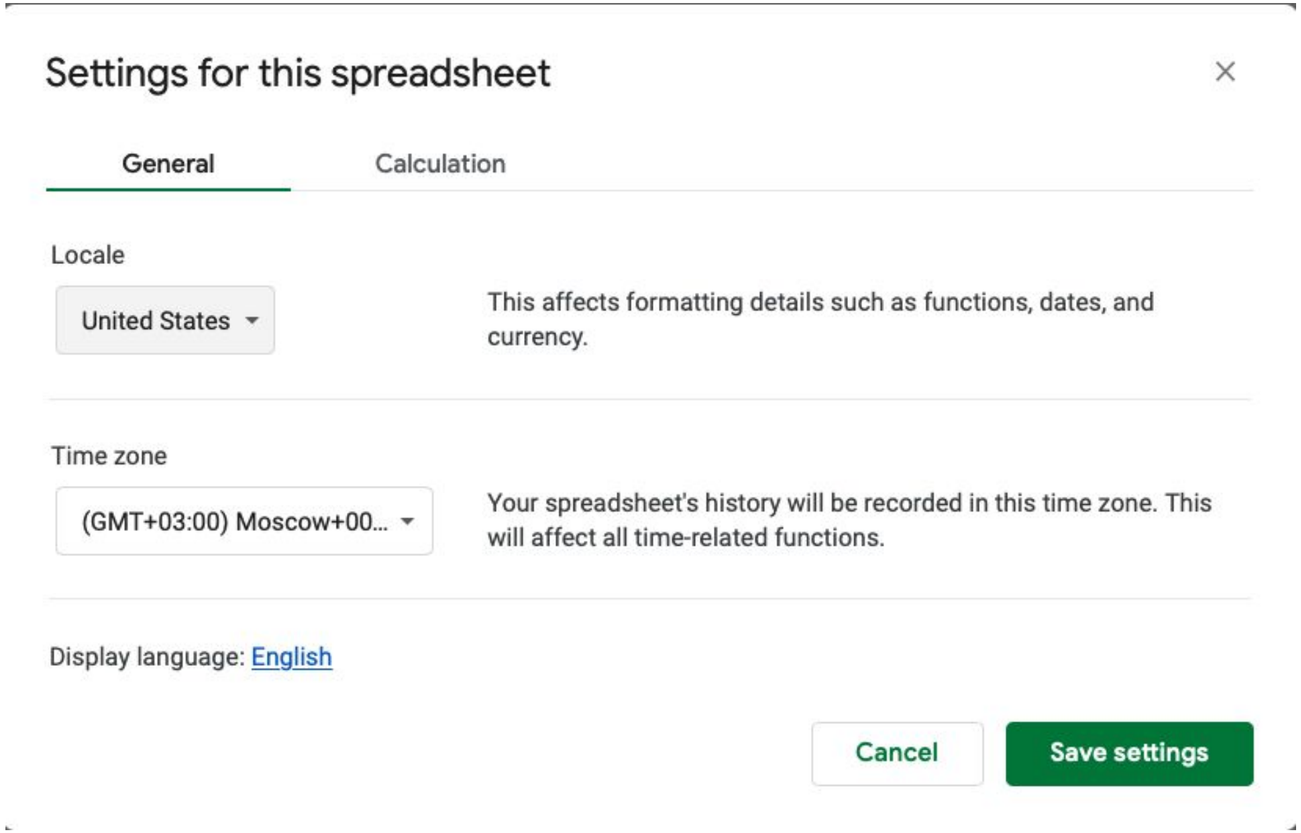
Вспоминаем про локаль

Числовые колонки в Google-таблицах отформатированы с учетом локали.

Формат сохраняется при экспорте в csv. Из-за этого Pandas может некорректно импортировать данные (импортировать числа как строки).

Исправляем форматирование колонки с числами

- Меняем локаль на United States
- File -> Spreadsheet settings
- Locale -> выбрать United States -> Save Settings



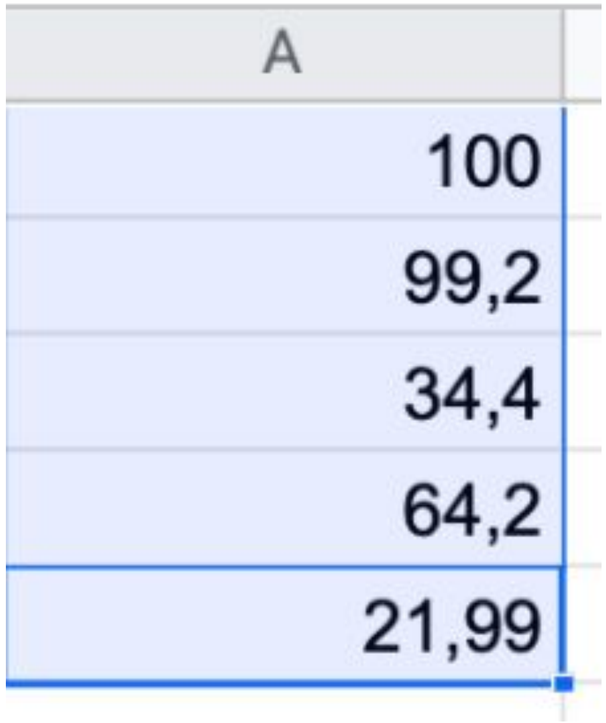
Исправляем форматирование колонки с числами

- Берем столбец с числами

A
100
99,2
34,4
64,2
21,99

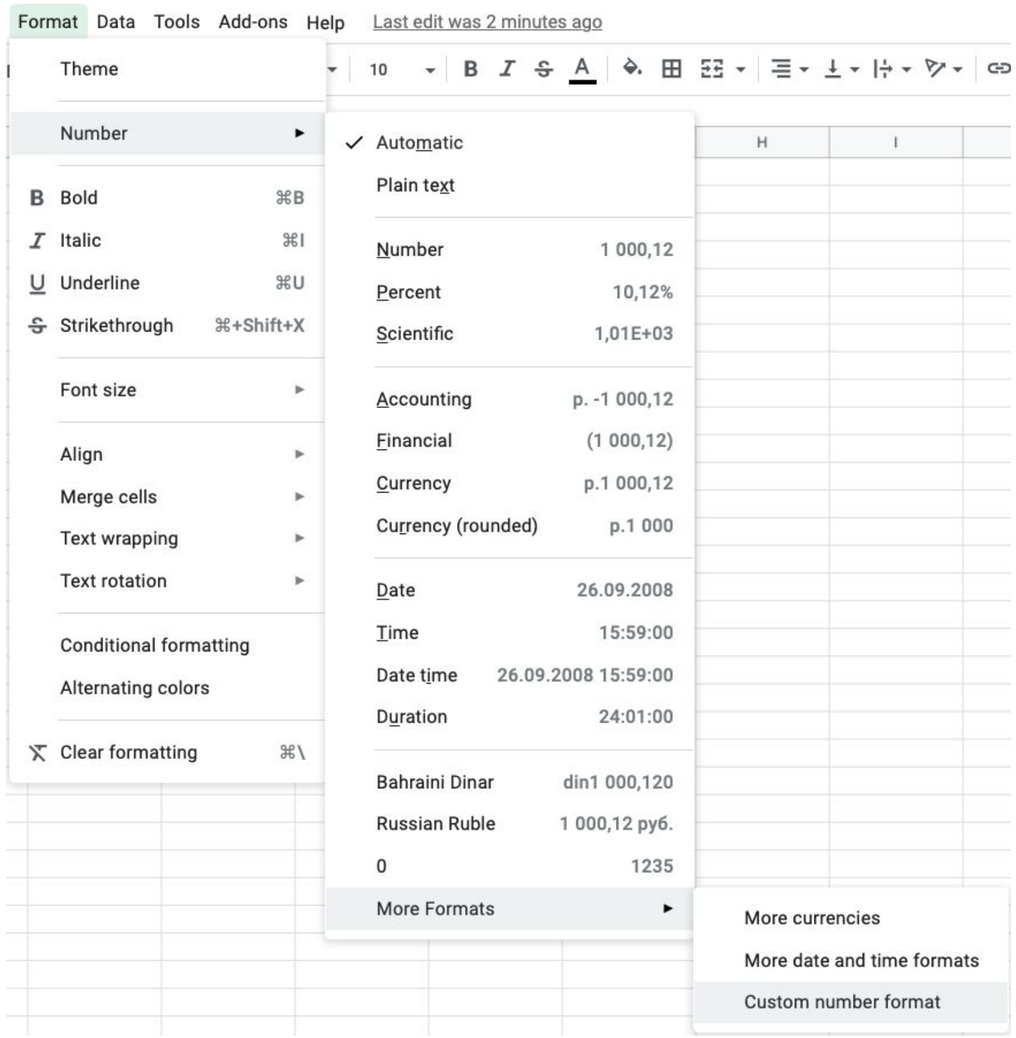
Исправляем форматирование колонки с числами

- Выделяем диапазон ячеек



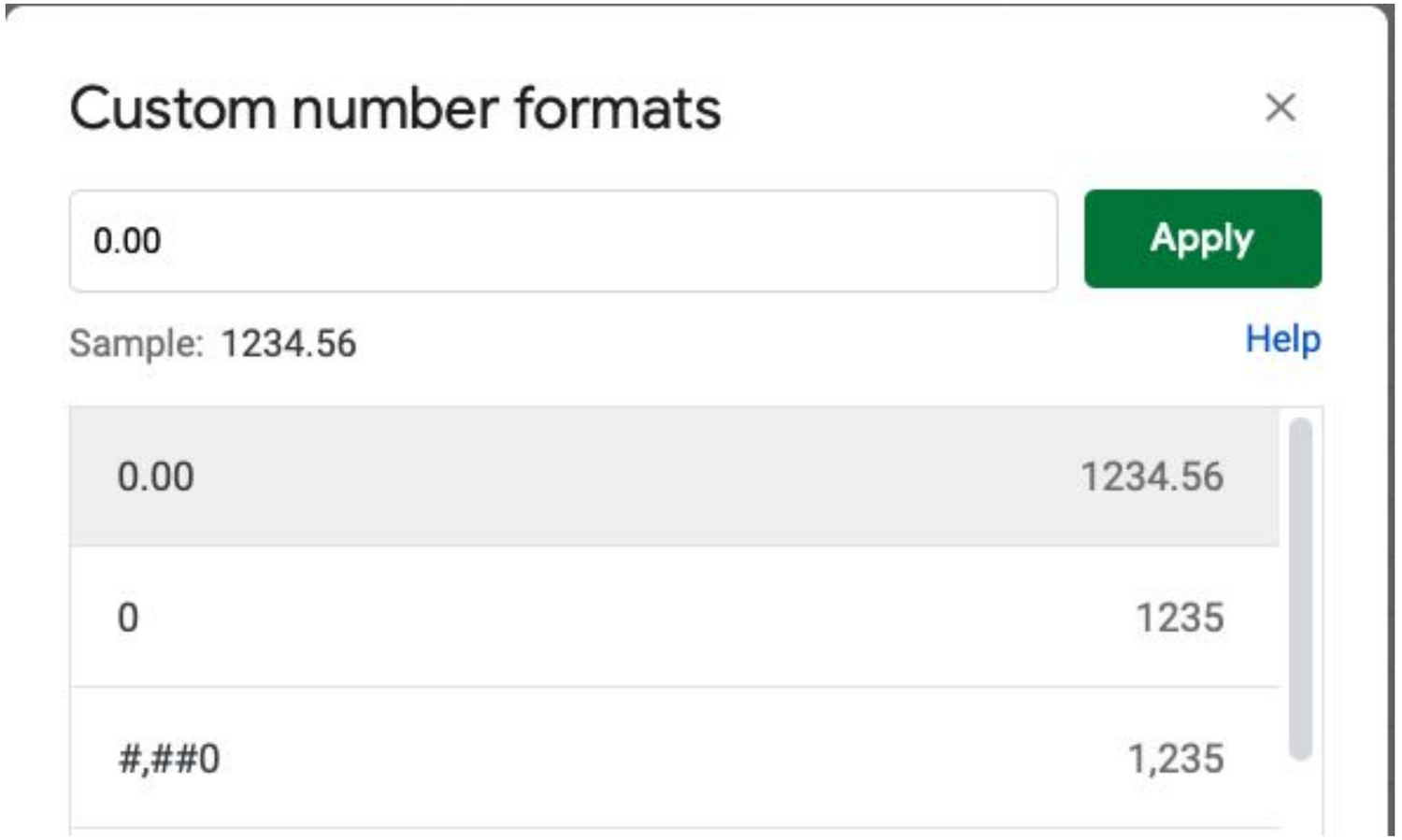
Исправляем форматирование колонки с числами

- Идем в меню Format -> Number -> More Formats -> Custom Number Format



Исправляем форматирование колонки с числами

- Выбираем формат 0.00
- Нажимаем apply



Итоги

Алексей Кузьмин

Аналитическое мышление

Что мы узнали сегодня

- Познакомились с языком программирования Python
- Научились работать с ним в облачной среде Google Colab
- Подключили библиотеку pandas и загрузили в нее данные

Домашнее задание

7

Алексей Кузьмин

Аналитическое мышление

нетология

Домашняя работа #5:

В этом задании научитесь проводить первичный анализ данных с помощью языка python.

Перед началом работы:

1. Поменяйте локаль на United States и установите формат для столбца с суммами затрат в 0.00 (используйте материал лекции).
2. Скачайте данные из Google-таблиц в виде csv-файла. Для скачивания нужно: открыть раздел File >>> Download >>> CSV-файл (текущий лист).

(**Внимание!** Сохраняется текущий, активный лист. Убедитесь, что вы скачиваете лист с данными)

Задание:

1. Запустите Google Colaboratory (https://colab.research.google.com)
2. На боковой панели нажмите на значок с изображением папки (при наведении он будет подписан как «файлы»). Ничего не открывая и не меняя в панели, перетащите скачанный csv-файл с данными в эту область.
3. Подключите библиотеку pandas, выполнив ячейку с кодом из лекции: «import pandas as pd».
4. Загрузите csv-файл в pandas-датафрейм, с помощью команды df = pd.read_csv(‘...’) указав вместо «...» необходимые параметры.
5. Выведите общую статистику: минимальное, среднее и максимальное значение трат, без учёта категорий. Для этого воспользуйтесь командой describe().
6. Сгруппируйте данные по категориям и вычислите суммарные затраты в каждой из них. Для этого воспользуйтесь командами groupby() и sum() по аналогии с тем, как мы делали на лекции.

Результат выполненной работы:

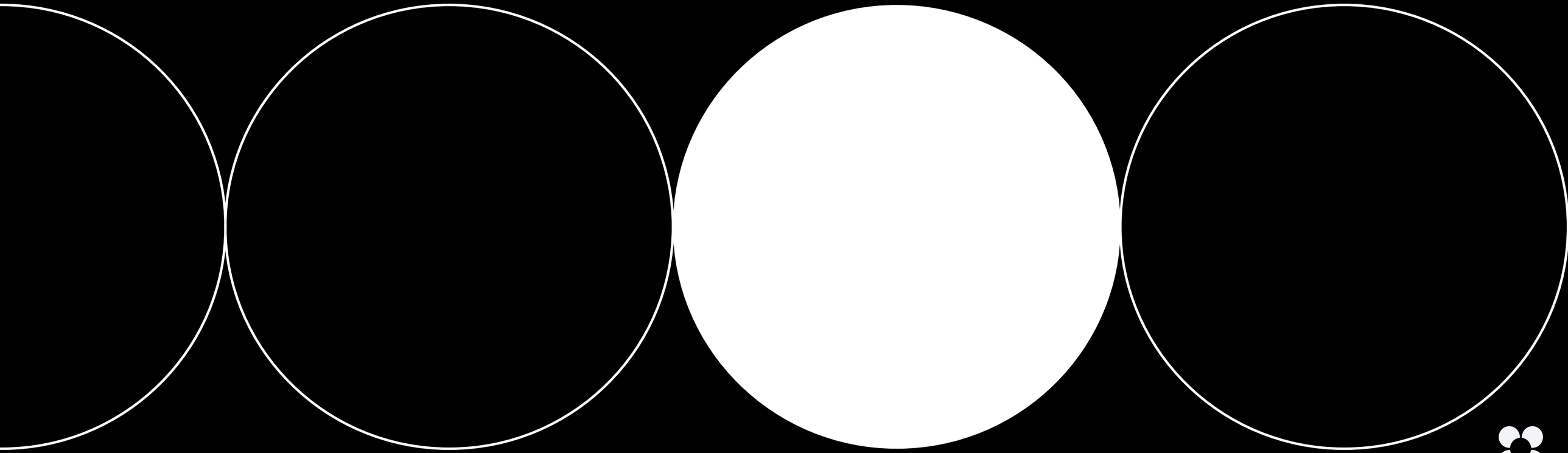
- Ссылка на ноутбук в google colaboratory. К ноутбуку должен быть открыт доступ на комментирование.

Убедитесь, что к ноутбуку предоставлен доступ по ссылке, иначе преподаватель не сможет проверить работу. Для этого можно открыть браузер в режиме инкогнито и убедиться, что ссылки открываются корректно.

Как запустить chrome в режиме инкогнито

Как запустить Safari в режиме инкогнито

Дополнительные ресурсы



Что почитать

- «Изучаем Python. Программирование игр, визуализация данных, веб-приложения» Эрик Мэтиз
- «Изучаем программирование на Python» Пол Бэрри
- «Изучаем pandas» Артем Груздев, Майкл Хейдт

- 10 минут в Pandas
- С чего начать изучение Python

Спасибо за внимание!

@Alexey Kuzmin

Алексей Кузьмин
Директор разработки
ДомКлик.ру