**ПРАКТИЧЕСКАЯ РАБОТА 6**

**Анализ данных**

**Цели:** Сбор и хранение данных скорости интернета на Python

**Задание:** В этой лаборатории, вы получите статистику скорости интернета и сохраните данные в реальном времени в файле CSV. Вы также загрузите сохраненные данные из файла CSV в структуру данных Python: Pandas DataFrame, и используете его функциональные возможности для изучения данных и управления ними для легкой читаемости.

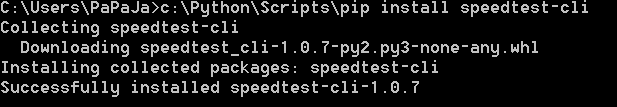
Необходимые библиотеки Python: datetime, csv, subprocess, pandas, numpy.

**Порядок выполнения работы**

**Часть 1. Сбор и хранение данных**

**Шаг 1. Установите Speedtest**

Speedtest-cli - это скрипт Python, который измеряет скорость загрузки и загрузки вашего интернет-соединения с помощью speedtest.net. Для получения дополнительной информации о speedtest-cli перейдите на страницу <https://github.com/sivel/speedtest-cli>. Для установки Speedtest-cli с помощью pip в командной строке выполните команду pip install speedtest-cli, указав путь к директории, где установлен Python.



**Шаг 2. Создайте новый файл в Python3.6 IDLE и импортируйте библиотеки Python.**

# Python library to manage date and time data

import datetime

# Python library to read and write csv files

import csv

# Python library to execute bash commands.

# If you want to know more about this, check this resource:

# http://www.pythonforbeginners.com/os/subprocess-for-system-administrators

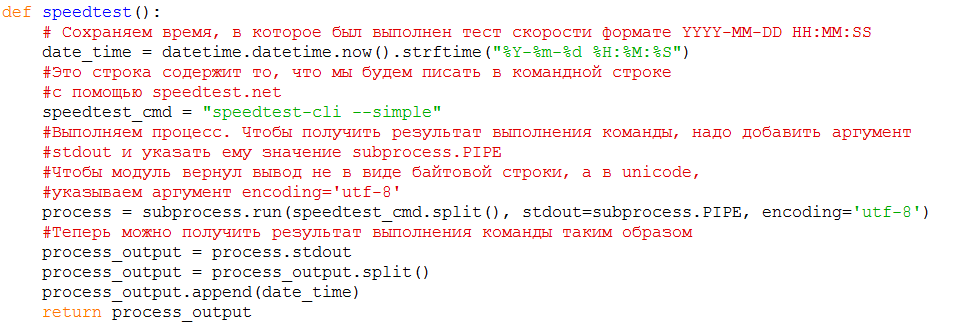
import subprocess

Модуль datetime позволяет работать со временем и датами в Python.

Модуль csv позволяет работать с файлами CSV.

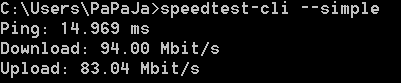
Модуль subprocess позволяет создавать новые процессы. При этом он может подключаться к стандартным потокам ввода/вывода/ошибок и получать код возврата. С помощью subprocess можно, например, выполнять любые команды из скрипта и, в зависимости от ситуации, получать вывод или только проверять, что команда выполнилась без ошибок.

**Шаг 3: Написать функцию speedtest, которая будет собирать сведения о скорости Интернета.**

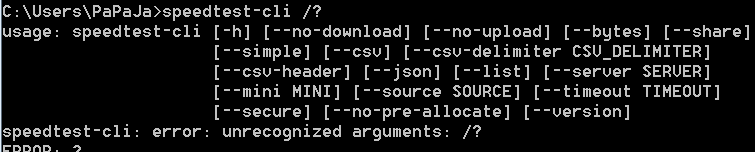


Чтобы создать метку времени используется функция datetime.now пакета datetime.

Команда speedtest-cli, если она запускается с терминала, возвращает строку со скоростью загрузки и выгрузки.



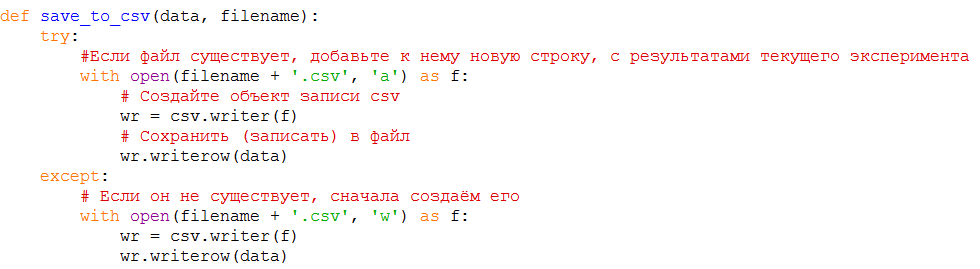
Дополнительные параметры команды speedtest-cli:



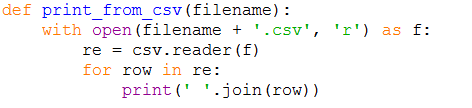
Чтобы выполнить команду из созданного файла Python, необходимо использовать подпроцесс модуля, который позволяет запускать процесс непосредственно из файла. Чтобы получить результат выполнения команды, надо добавить аргумент stdout и указать ему значение subprocess.PIPE.

**Шаг 4: Создание функции сохранения результата функции speedtest() в файл csv.**

Напишите функцию, чтобы создать файл csv для записи результата функции speedtest(). Для этого необходимо создать объект csv.writer (справка <https://docs.python.org/3.6/library/csv.html>).

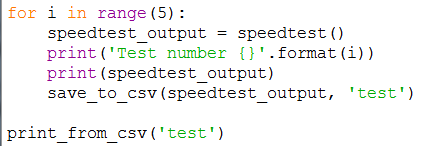


**Шаг 5: Напишите функцию, чтобы открыть файл csv и распечатать его содержимое на экране.**



**Шаг 6: Запустите Speedtest несколько раз и сохраните данные.**

Напишите цикл for, который вызывает 5 раз тест скорости, сохраняет данные в CSV-файле и распечатывает выходные данные тестов.

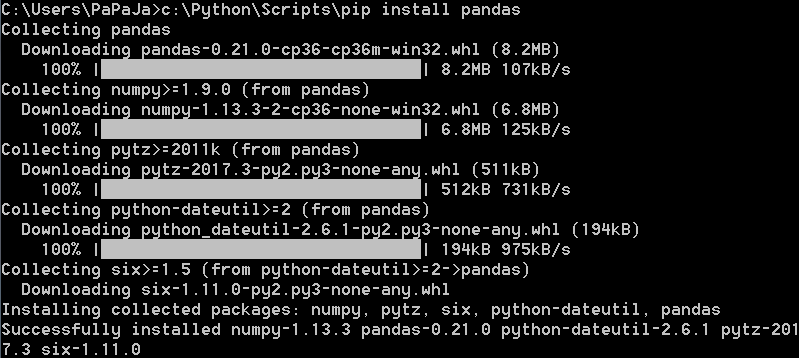


**Часть 2: Обработка данных**

Библиотека panda очень полезна для работы со структурированными данными. Полную документацию можно найти здесь: <http://pandas.pydata.org/pandas-docs/stable/>

Для этой части лаборатории будет использоваться более крупный набор данных, собранный заранее. Имя файла: rpi\_data\_long.csv.

**Шаг 1. Установите panda**



**Шаг 2. Создайте новый файл в Python3.6 IDLE и импортируйте необходимые библиотеки Python.**

**import** **datetime**

**import** **csv**

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

*# NumPy - это библиотека, которая добавляет поддержку больших многомерных*

*# массивов и матриц наряду с высокоуровневыми математическими функциями для*

*# работы с этими массивами*

**import** **numpy** **as** **np**

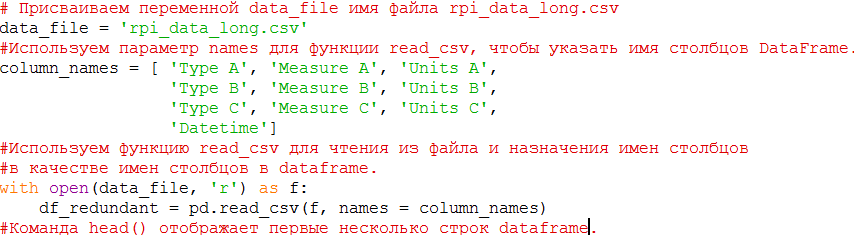
**Шаг 3. Загрузите файл csv в объект DataFrame с помощью panda.**

Чтобы эффективно работать с pandas, необходимо освоить самые главные структуры данных библиотеки: DataFrame и Series.

Объект DataFrame можно представить в виде таблицу, в которой всегда присутствуют строки и столбцы или таблицу SQL. Считать данные из csv-файла и превратить в DataFrame можно функцией read\_csv.

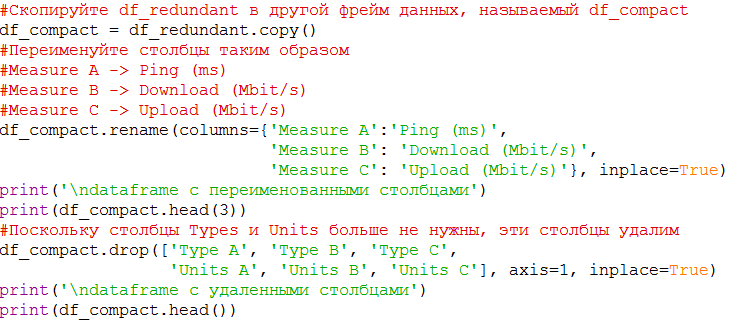
Прочитайте документацию о read\_csv в <http://pandas.pydata.org/pandas-docs/version/0.14.1/generated/pandas.read_csv.html>. Эта функция содержит множество параметров. Единственным обязательным является путь к файлу, то есть расположение файла csv. Все остальные параметры являются необязательными.

На этом этапе вы будете импортировать и просматривать содержимое файла rpi\_data\_long.csv. Этот файл csv находится в том же каталоге, что и установленный Python.



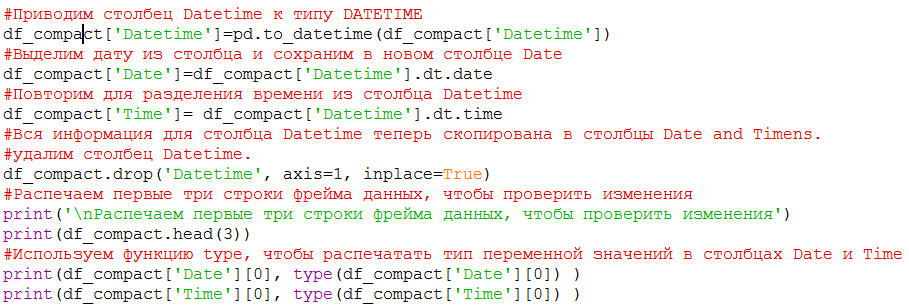
**Шаг 4: Создание краткого представления**

На этом этапе создадим более компактное представление, используя копию фрейма данных df\_redundant.



**Шаг 5: Разделение даты на две колонки.**

Будем использовать Pandas для генерации столбцов Date и Time из столбца Datetime, а затем удалим столбец Datetime.



**Шаг 6: Сохранение нового фрейма данных.**

df\_compact.to\_csv('rpi\_data\_compact.csv')