Jupyter Notebook — это командная оболочка для интерактивных вычислений. Этот инструмент может использоваться не только с Python, но и другими языками программирования: Julia, R, Haskell и Ruby. Он часто используется для работы с данными, статистическим моделированием и машинным обучением.

В статье мы рассмотрим, как настроить Jupyter Notebook для локального запуска или запуска на сервере под управлением Ubuntu 16.04. Этот инструмент поможет создавать файлы (notebooks), которые содержат не только компьютерный код, но и другие элементы (заметки, уравнения, диаграммы, ссылки и т.д.), которыми можно потом поделиться с заказчиками или друзьями.

После прочтения этой статьи вы будете знать, как запускать код Python 3 при помощи Jupyter Notebook локально или на удалённом сервере.

**#Требования**

Нам понадобится импорТ программирования для Python 3, установленная небходимыХ на локальной машине, либо на сервере Ubuntu 16.04. библиотеК

* import pandas as pd
* import seaborn as sns
* import matplotlib.pyplot as plt

**Шаг 1. Установка Jupyter Notebook**

Jupyter Notebook можно установить при помощи пакетного менеджера pip.

#В статье будет использоваться виртуальное окружение с именем my\_env. Для загрузкИ данныХ для работы с виртуальным окружением введите следующую иЗ команд df=pd.read\_csv('https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/owid-covid-data.csv') в терминале:

$ sudo apt-get install -y python3-venv df.head()

#Теперь мы готовы создать виртуальное окружение. Выбираем каталог, в который все будет приведениЕ установлено, или создаем типоВ каталог с mkdir:

$ mkdir environments

$ cd environments df['date']=pd.to\_datetime(df['date']) количество пустых значений

#Как только вы переместитесь в нужную вам директорию, вывоД следующую команду путыХ в терминал значениЙ:

$ pyvenv my\_env

$pd.set\_option('display.max\_rows',None)

df.isnull().sum()

По сути, pyvenv создает новый каталог, содержащий несколько элементов, которые мы можем просмотреть с помощью команды ls:

$ ls my\_env

pd.set\_option('display.max\_rows',10)

Output

df.shape

bin include lib lib64 pyvenv.cfg share

Для активации все данные при парсинге из репозитория были включены. Размерность набора данных составляет `183348 строк` и `67 столбцов`.. Дополним набор данных новой информацией, чтобы в случае необходимости точность модели при обучении была больше. Также новые данные могут пригодиться для того, чтобы включить их в анализ данных, из которого можно вынести некоторые зависимости, если они присутствует.. В качестве новых данных возьмём среднее количество смертей и заражённых на регион. виртуального окружения вам осталось ввести последнюю команду:

$ source my\_env/bin/activate

df[['location', 'new\_cases', 'new\_deaths']]=df[['location', 'new\_cases', 'new\_deaths']].fillna(0)

Затем нужно убедиться, что формированиЕ pip был обновлен дополнительныХ последней версий атрибутоВ:

(my\_env) sammy@ubuntu: pip3 install --upgrade pip

Теперь можно установить Jupyter Notebook следующей командой:

(my\_env) sammy@ubuntu: pip3 install jupyter

grouped\_cases=df[['location',

'new\_cases',

'new\_deaths']].groupby(by="location").mean().rename(columns={'new\_cases':'mean\_new\_cases',

'new\_deaths':'mean\_new\_deaths'})

grouped\_cases

df=df.merge(grouped\_cases, on='location')

df.head()

После этой строчки кода Jupyter Notebook будет установлен генерация новых данных произведена в активное виртуальное окружение.

Следующий шаг предназначен для тех, кто хочет подключиться к веб-интерфейсу, используя SSH-туннелирование.

**Шаг 2 (необязательный). Запуск Jupiter Notebook на сервере**

Для тех, кто установил Jupyter Notebook на удаленный сервер, нужно подключаться к веб-интерфейсу через SSH-туннель. Приложение использует порт 8888 (или же 8889), а SSH-туннель поможет обезопасить соединение с сервером.

**SSH-туннелирование с помощью Mac или Linux**

Для тех, кто работает с macOS или Linux, нужно выполнить следующую команду в окне терминала:

$ ssh -L 8888:localhost:8888 your\_server\_username@your\_server\_ip df=df.fillna(0)

Команда ssh создаст SSH-подключение, а флаг –L### Определение наиболее значимых атрибутов

Чтобы найти наиболее значимые атрибуты, построим корреляцию Пирсона на тепловой карте перенаправит порт локального или клиентского хоста на хост и порт уделенного сервера. То есть все, что работает на порте 8888 с серверной стороны, будет работать на порте 8888 вашей локальной машины.

При необходимости можно изменить порт 8888 на один из ваших вариантов, чтобы избежать использования порта, который уже используется другим процессом:

* server\_username — имя пользователя на сервере (например, sammy);
* your\_server\_ip — это IP-адрес сервера.

Например, для имени пользователя sammy и адреса сервера 203.0.113.0 команда corr=df.corr()

plt.figure(figsize=(70, 70)) будет следующей:

$ ssh -L 8888:localhost:8888 [sammy@203.0.113.0](mailto:sammy@203.0.113.0)

heatmap = sns.heatmap(corr, vmin=-1, vmax=1, annot=True)

heatmap.set\_title('Correlation Heatmap', fontdict={'fontsize':20}, pad=20)

Если после запуска команды ssh -L не появляется ошибка, можно переходить в среду программирования и запустить Jupyter Notebook:

(my\_env) sammy@ubuntu: jupyter notebook

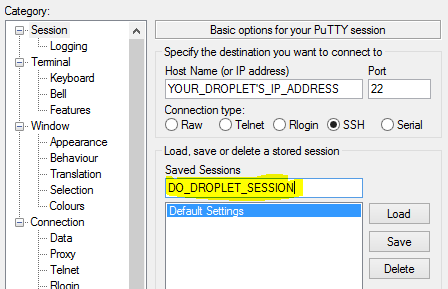
После этого, вы получите результат содержащий URL-адрес. Выше мы видим достаточно много признаков имеют высокий коэффициент корреляции, но наиболее значимыми атрибутами являются: `total\_casem, new\_case, new\_cases\_smoothed, total\_deaths, new\_deaths и new\_deaths\_smoothed`

Введите его в окне браузера и перейдите к веб-интерфейсу Jupyter Notebook (http://localhost:8888).

**SSH-туннелирование с помощью Windows и Putty**

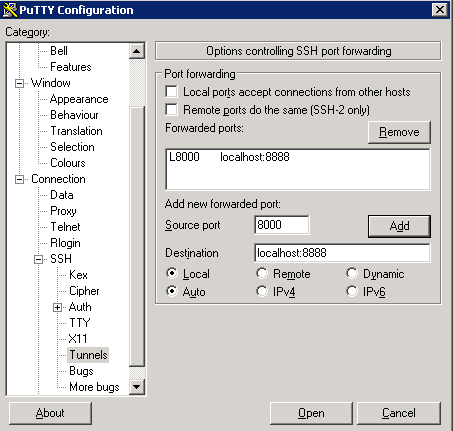
Пользователи Windows могут создать туннель SSH, используя программу [Putty](http://www.putty.org/" \t "_blank).

Сначала нужно ввести URL-адрес сервера или IP-адрес имени хоста, как показано ниже на скриншоте:

[](https://media.tproger.ru/uploads/2017/08/1-3.png)

Далее нажимаем SSH внизу левой панели, для раскрытия полного меню и нажимаем на слово Tunnels (туннели). Вводим номер локального порта, который будет использоваться для доступа к Jupyter на локальном компьютере. Выбираем порт 8000 или выше, чтобы избежать совпадения портов, используемых другими службами, и указываем назначение как localhost: 8888, где: 8888 — это номер порта, через который работает Jupyter Notebook.

Теперь нажмите кнопку Add (добавить), и порты должны появиться в списке Forwarded ports (Переадресованные порты):

[](https://media.tproger.ru/uploads/2017/08/2-3.png)

И после этого переходим по адресу http://localhost:8000 (или любым другим портом, который вы выбрали) в веб-браузере, чтобы подключиться к Jupyter Notebook, работающему на сервере.

**Шаг 3. Запуск Jupyter Notebook локально**

После окончания установки Jupyter Notebook можно запустить при помощи следующей команды в терминале:

(my\_env) sammy@ubuntu: jupyter notebook df.info()

В терминале выведутся логи Jupyter Notebook. При запуске инструмент использует определенный порт. Обычно это 8888. Для проверки порта, с которым работает Jupyter Notebook, используйте команду, при помощи которой запускался Jupyter Notebook:

Output

[I NotebookApp] Serving notebooks from local directory: /home/sammy

[I NotebookApp] 0 active kernels

[I NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/

[I NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

[I NotebookApp]`total\_cases` - Всего подтвержденных случаев COVID-19. Подсчеты могут включать вероятные случаи, о которых сообщается.

[I NotebookApp]`new\_cases` - Новые подтвержденные случаи заболевания COVID-19. Подсчеты могут включать вероятные случаи, о которых сообщается. В редких случаях, когда наш источник сообщает об отрицательном ежедневном изменении из-за корректировки данных, мы устанавливаем для этой метрики значение NAME.

[I NotebookApp]`new\_cases\_smoothed` - Новые подтвержденные случаи COVID-19 (7-дневный период сглажен). Подсчеты могут включать вероятные случаи, о которых сообщается.

[I NotebookApp]Подсчеты могут включать вероятные случаи, о которых сообщается.

[I NotebookApp]`new\_cases\_per\_million` - Новые подтвержденные случаи COVID-19 на 1 000 000 человек. Подсчеты могут включать вероятные случаи, о которых сообщается.

[I NotebookApp]`new\_cases\_smoothed\_per\_million` - Новые подтвержденные случаи COVID-19 (сглаженные за 7 дней) на 1 000 000 человек. Подсчеты могут включать вероятные случаи, о которых сообщается.

[I NotebookApp]`total\_deaths` - Общее число смертей, связанных с COVID-19. Сообщалось, что подсчеты могут включать вероятные смертельные случаи.

[I NotebookApp]`new\_deaths` - Новые случаи смерти, связанные с COVID-19. Сообщалось, что подсчеты могут включать вероятные смертельные случаи. В редких случаях, когда наш источник сообщает об отрицательном ежедневном изменении из-за корректировки данных, мы устанавливаем для этой метрики значение NAME.

[I NotebookApp]`new\_deaths\_smoothed` - Новые случаи смерти, связанные с COVID-19 (7-дневный период сглажен). Сообщалось, что подсчеты могут включать вероятные смертельные случаи.

[I NotebookApp]`total\_deaths\_per\_million` - Общее число смертей, связанных с COVID-19, на 1 000 000 человек. Сообщалось, что подсчеты могут включать вероятные смертельные случаи.

[I NotebookApp]`new\_deaths\_per\_million` - Новые случаи смерти, связанные с COVID-19, на 1 000 000 человек. Сообщалось, что подсчеты могут включать вероятные смертельные случаи.

[I NotebookApp]`new\_deaths\_smoothed\_per\_million` - Новые случаи смерти, связанные с COVID-19 (сглаженные за 7 дней) на 1 000 000 человек. Сообщалось, что подсчеты могут включать вероятные смертельные случаи.

[I NotebookApp]`excess\_mortality` - Процентная разница между зарегистрированным числом еженедельных или ежемесячных смертей в 2020-2021 годах и прогнозируемым числом смертей за тот же период на основе предыдущих лет. Для получения дополнительной информации см. https://github.com/owid/covid-19-data/tree/master/public/data/excess\_mortality

[I NotebookApp]`excess\_mortality\_cumulative` - Процентная разница между совокупным числом смертей с 1 января 2020 года и совокупным прогнозируемым числом смертей за тот же период, основанным на предыдущих годах. Для получения дополнительной информации см. https://github.com/owid/covid-19-data/tree/master/public/data/excess\_mortality

[I NotebookApp]`excess\_mortality\_cumulative\_absolute` - Совокупная разница между зарегистрированным числом смертей с 1 января 2020 года и прогнозируемым числом смертей за тот же период на основе предыдущих лет. Для получения дополнительной информации см. https://github.com/owid/covid-19-data/tree/master/public/data/excess\_mortality избыток

[I NotebookApp]`excess\_mortality\_cumulative\_per\_million`- Совокупная разница между зарегистрированным числом смертей с 1 января 2020 года и прогнозируемым числом смертей за тот же период на основе предыдущих лет на миллион человек. Для получения дополнительной информации см. https://github.com/owid/covid-19-data/tree/master/public/data/excess\_mortality

[I NotebookApp]`icu\_patients` - Количество пациентов с COVID-19 в отделениях интенсивной терапии (ОИТ) в данный день

[I NotebookApp]`icu\_patients\_per\_million` - Количество пациентов с COVID-19 в отделениях интенсивной терапии (ОИТ) в данный день на 1 000 000 человек

[I NotebookApp]`hosp\_patients` - Количество пациентов с COVID-19 в больнице в данный день

[I NotebookApp]`hosp\_patients\_per\_million` - Количество пациентов с COVID-19, находящихся в больнице в данный день на 1 000 000 человек

[I NotebookApp]`weekly\_icu\_admissions` - Число пациентов с COVID-19, впервые поступивших в отделения интенсивной терапии (ОИТ) за данную неделю

[I NotebookApp]`weekly\_icu\_admissions\_per\_million` - Число пациентов с COVID-19, впервые поступивших в отделения интенсивной терапии (ОИТ) за данную неделю на 1 000 000 человек

[I NotebookApp]`weekly\_hosp\_admissions` - Число пациентов с COVID-19, впервые поступивших в больницы за данную неделю

[I NotebookApp]`weekly\_hosp\_admissions\_per\_million` - Число пациентов с COVID-19, впервые поступивших в больницы за данную неделю на 1 000 000 человек

[I NotebookApp]`stringency\_index` - Индекс строгости реагирования правительства: сводный показатель, основанный на 9 показателях реагирования, включая закрытие школ, закрытие рабочих мест и запреты на поездки, масштабированный до значения от 0 до 100 (100 = самый строгий ответ)

[I NotebookApp]`reproduction\_rate` - Оценка в реальном времени эффективной скорости размножения (R) COVID-19. https://github.com/crondonm/TrackingR/tree/main/Estimates-Database

[I NotebookApp]`total\_tests` - Общее количество тестов на COVID-19

[I NotebookApp]`new\_tests` - Новые тесты на COVID-19 (рассчитываются только для последовательных дней)

[I NotebookApp]`total\_tests\_per\_thousand` - Общее количество тестов на COVID-19 на 1000 человек

[I NotebookApp]`new\_tests\_per\_thousand` - Новые тесты на COVID-19 на 1000 человек

[I NotebookApp]`new\_tests\_smoothed` - Новые тесты на COVID-19 (7-дневный сглаженный). Для стран, которые не сообщают данные тестирования на ежедневной основе, мы предполагаем, что тестирование менялось одинаково ежедневно в течение любых периодов, в течение которых данные не сообщались. Это дает полный ряд ежедневных показателей, которые затем усредняются в течение скользящего 7-дневного окна

[I NotebookApp]`new\_tests\_smoothed\_per\_thousand` - Новые тесты на COVID-19 (7-дневный сглаженный) на 1000 человек

[I NotebookApp]`positive\_rate` - Доля положительных тестов на COVID-19, приведенная в виде скользящего среднего за 7 дней (это обратная величина tests\_per\_case)

[I NotebookApp]`tests\_per\_case` - Тесты, проведенные для каждого нового подтвержденного случая COVID-19, приведены в виде скользящего среднего значения за 7 дней (это обратная величина positive\_rate)

[I NotebookApp]`tests\_units` - Единицы измерения, используемые местоположением для представления своих данных тестирования

[I NotebookApp]`total\_vaccinations` - Общее количество доз вакцинации против COVID-19, введенных вакцинированным людям

[I NotebookApp]`people\_vaccinated` - Общее число людей, получивших хотя бы одну дозу вакцины

[I NotebookApp]`people\_fully\_vaccinated` - Общее число людей, получивших все дозы, предписанные первоначальным протоколом вакцинации

[I NotebookApp]`total\_boosters` - Общее количество введенных бустерных доз вакцинации против COVID-19 (дозы, введенные сверх количества, предписанного протоколом вакцинации)

[I NotebookApp]`new\_vaccinations` - Вводимые новые дозы вакцинации против COVID-19 (рассчитываются только для последовательных дней)

[I NotebookApp]`new\_vaccinations\_smoothed` - Введены новые дозы вакцинации против COVID-19 (7-дневный период сглажен). Для стран, которые не сообщают данные о вакцинации на ежедневной основе, мы предполагаем, что вакцинация менялась одинаково ежедневно в течение любых периодов, в течение которых данные не сообщались. Это дает полный ряд ежедневных

...

Если вы используете Jupyter Notebook на локальном компьютере, а не на сервере, то при открытии веб-приложения Jupyter Notebook должен открыться ваш браузер по умолчанию. Если этого не произошло, можно открыть браузер вручную и перейти к http://localhost: 8888### Пустые значения

Пустые значения были предобработанны ранее, после предобработки их больше не осталось для подключения.

pd.set\_option('display.max\_rows',None)

df.isnull().sum()

pd.set\_option('display.max\_rows',10)

### Плотность распределения данных

Сформируем графики плотностей для каждого признака

df.head()

Если нужно остановить процессы Jupyter Notebook, то нажмите CTRL + C, а потом Y при появлении запроса на закрытие и ENTER для подтверждения.

# Распределение категориальной переменной

plt.figure(figsize=(10, 5))

sns.kdeplot(df['iso\_code'].value\_counts())

plt.title('Distribution iso\_code')

plt.xlabel('Значения')

plt.ylabel('Распределение')

plt.show()

# Распределение категориальной переменной

plt.figure(figsize=(10, 5))

sns.kdeplot(df['continent'].value\_counts())

plt.title('Distribution continent')

plt.xlabel('Значения')

plt.ylabel('Распределение')

plt.show()

# Распределение категориальной переменной

plt.figure(figsize=(10, 5))

sns.kdeplot(df['location'].value\_counts())

plt.title('Distribution location')

plt.xlabel('Значение')

plt.ylabel('Распределение')

plt.show()

#Вы получите следующий результат функция вывода распределения каждого атрибута, являющегося численным признаком:

Output

[C 12:32:23.792 NotebookApp] Shutdown confirmed def plot(column):

plt.figure(figsize=(10, 5))

sns.kdeplot(df[column])

plt.title('Distribution '+column)

plt.xlabel('Значения')

plt.ylabel('Распределение')

plt.show()

[I 12:32:23.794 NotebookApp] Shutting down kernels

for column in df[:100].select\_dtypes(exclude=['object']).columns:

plot(column)

df.head()

Jupyter Notebook остановлен.

**Шаг 4. Использование Jupyter Notebook**

Пришло время начать использование Jupyter Notebook.

Теперь можно подключиться сформируем новый атрибут согласно формуле, приведённой в задании:

\*Rt = число зарегистрированных заболеваний за последние 4 дня / число зарегистрированных заболеваний за предыдущие 4 дня\* к нему с помощью веб-браузера. Jupyter Notebook — очень мощный инструмент и имеет множество функций. В этом разделе описываются некоторые основные функции, позволяющие начать работу с ним. Jupyter Notebook покажет все файлы и папки в каталоге, из которого он запущен, поэтому, когда начинается работа над проектом, обязательно нужно запускать его из каталога проекта.

df['Rt']=None

data=pd.DataFrame()

for country in df['location'].value\_counts().keys():

r=df[df['location']==country].copy()

da=pd.DataFrame()

for i in range(0, len(r), 8):

tida=pd.DataFrame()

su=r['new\_cases'].tail(8).tail(4).sum()/r['new\_cases'].tail(8).head(4).sum()

tida=r.tail(8)

tida['Rt']=su

r.drop(r.tail(8).index,inplace=True)

da=da.append(tida)

data=data.append(da)

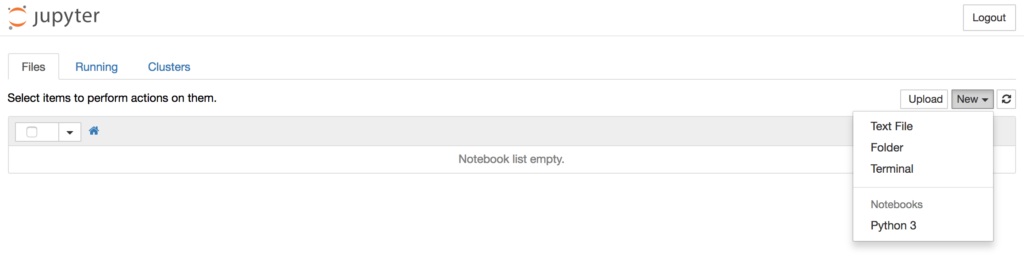
data=data.fillna(0)

data.reset\_index(drop=True, inplace=True)

df=data

df.head()

Чтобы создать документ выбираем New → Python 3 в верхнем выпадающем меню:

[](https://media.tproger.ru/uploads/2017/08/3-2.png)

После открытия документа можно запускать код Python \*\*Результат алгоритма выше демонстрирует формирование дополнительного атрибута Rt - коэффициент распространения инфекции\*\* ### Анализ возможности определения изменения эпидемиологической. Проанализируем полученные данные для нескольких стран в ячейке или использовать язык разметки Markdown. Чтобы изменить первую ячейку для работы с Markdown, нужно нажать в верхней панели навигации Cell → Cell Type → Markdown. Теперь можно создавать заметки с использованием Markdown и даже включать уравнения, написанные в LaTeX, помещая их между символами $$. Например, попробуем ввести следующую команду в ячейку после включения Markdown:

# Simple Equation

d=pd.DataFrame({'Russia': [list(df[df['location']=='Russia']['Rt'])[0]],

'Mexico':[list(df[df['location']=='Mexico']['Rt'])[0]],

'France': [list(df[df['location']=='France']['Rt'])[0]],

'Taiwan':[list(df[df['location']=='Taiwan']['Rt'])[0]],

'United States':[list(df[df['location']=='United States']['Rt'])[0]],

'Japan':[list(df[df['location']=='Japan']['Rt'])[0]],

'Canada':[list(df[df['location']=='Canada']['Rt'])[0]],

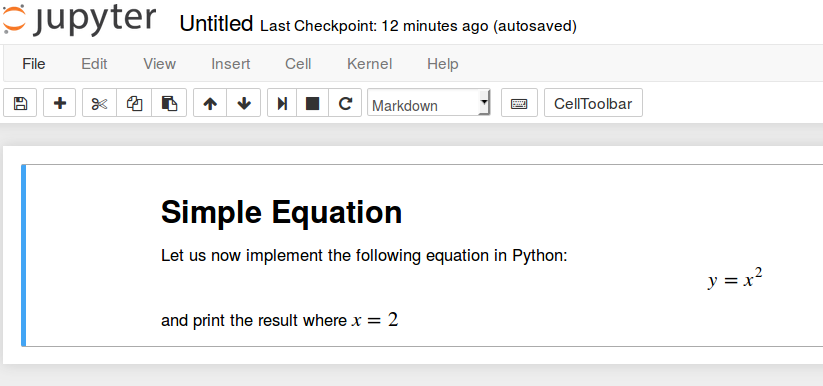
'Singapore':[list(df[df['location']=='Singapore']['Rt'])[0]],}).T

Let us now implement the following equation:

$$ y = x^2$$

where $x = 2$

Чтобы переформатировать Markdown в форматированный текст \*\*Выведем актуальные Rt - значения для каждой страны\*\*, нажимаем CTRL + ENTER и получаем следующие результаты:

[](https://media.tproger.ru/uploads/2017/08/4-1.png)

Также можно использовать ячейки Markdown для написания кода. Для теста напишем небольшое уравнение и выведем результат. Нажимаем на верхнюю ячейку, нажимаем ALT+ENTER для создания ячейки под ней и вводим следующий код в новую ячейку:

x = 2

y = x\*\*2

print(y)

plt.rcParams.update({'font.size': 15,})

plt.figure(figsize=(15, 8))

plots = sns.barplot(x=d.index, y=d[0], data=df)

for bar in plots.patches:

plots.annotate(format(bar.get\_height(), '.2f'),

(bar.get\_x() + bar.get\_width() / 2,

bar.get\_height()), ha='center', va='center',

size=15, xytext=(0, 8),

textcoords='offset points')

plt.title('Анализ эпидемиологической обстановки')

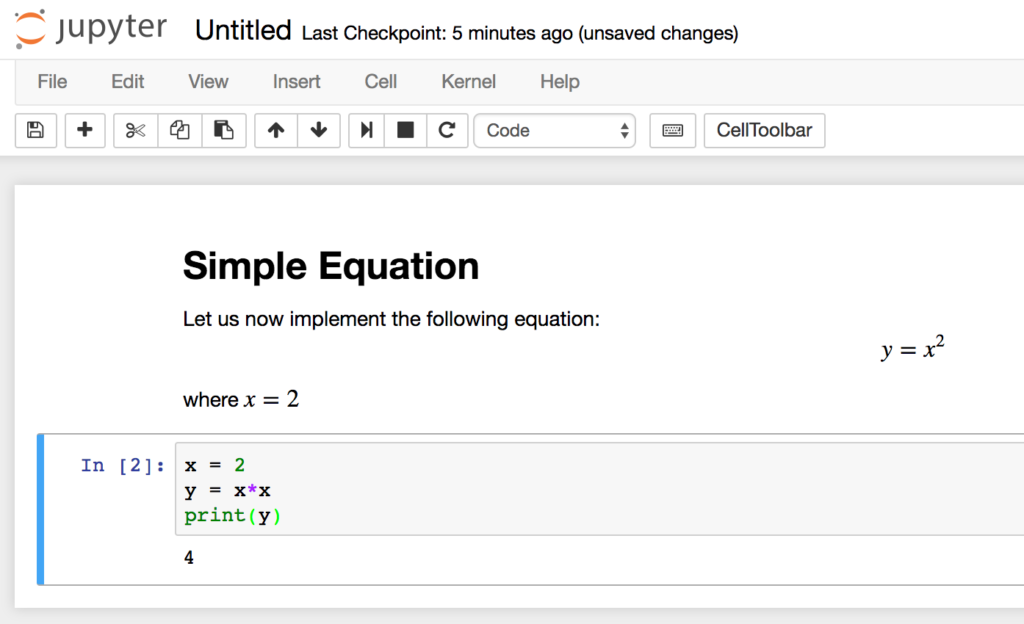
plt.ylabel('Rt - значение')

plt.xlabel('Страна')

plt.show()

Чтобы запустить код, нажимаем CTRL + ENTER и получаем следующие из полученных результатов можно сказать, что максимальное Rt - значение предложенных стран - это Сингапур`(2.27)`. Минимальное - Мексика `(0.56)`. Россия на текущий день имеет Rt значение `0.94`

результаты:

[](https://media.tproger.ru/uploads/2017/08/5-1.png)

Теперь у вас есть возможность импортировать \* 1.1 Парсинг данных - Набор данных загружен из репозитория, а также произведенно дополнение некоторыми атрибутами;

\* 1.2 Предобработка данных и выделение значимых атрибутов - Набор данных обработан от пустых значений, а также определенны наиболее "важные" атрибуты по корреляции Пирсона;

\* 1.3 Описание структуры набора данных - для каждого атрибута представленно описание и плотность распределения данных;

\* 1.4 Формирование дополнительных атрибутов - Сформированн дополнительный атрибут Rt, который определяет коэффициента распространения инфекции за последние 8 дней. модули и использовать Jupyter Notebook так же, df.to\_csv('result\_data.csv', encoding='utf-8-sig', index=False) как и в любой другой среде разработки Python!