ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

**РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА И ГОСУДАРСТВЕННОЙ СЛУЖБЫ ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ЛИПЕЦКИЙ ФИЛИАЛ

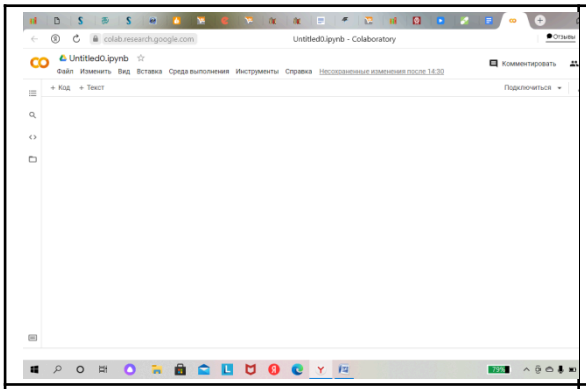
КАФЕДРА ГУМАНИТАРНЫХ И ЕСТЕСТВЕННОНАУЧНЫХ ДИСЦИПЛИН

# Лабораторная работа №6
по дисциплине
«Информационные технологии в ГМУ»

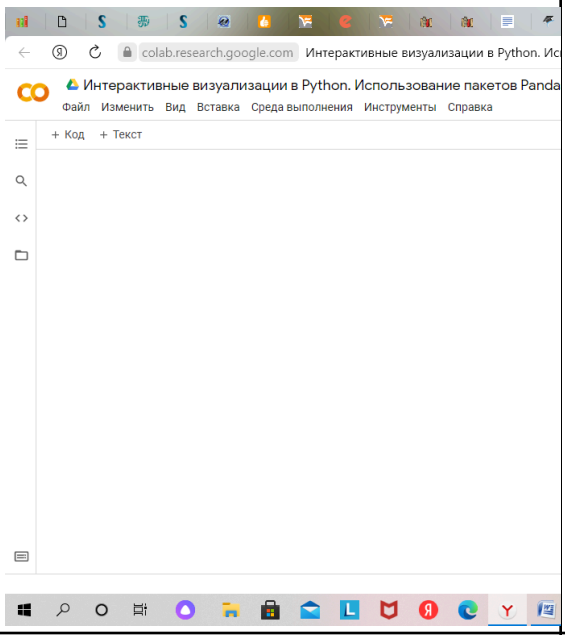
## ВАРИАНТ №1

Выполнил:
студент 1 курса
группы У-20-1
Фомина Алина

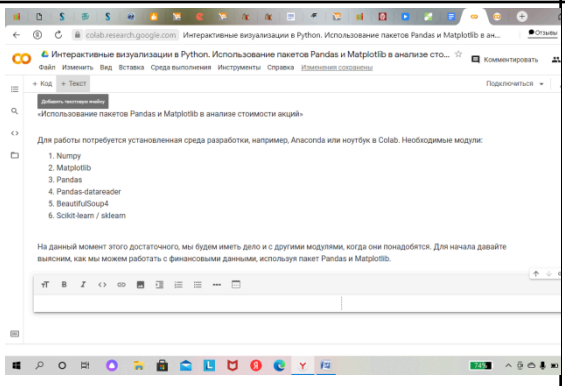
Проверил:
доцент
Яворский В.М.



1.Перейти на сайт по ссылке https://colab.research.google.com/ и создать новый блокнот, не забывая поделиться им с преподавателем в режиме онлайн.

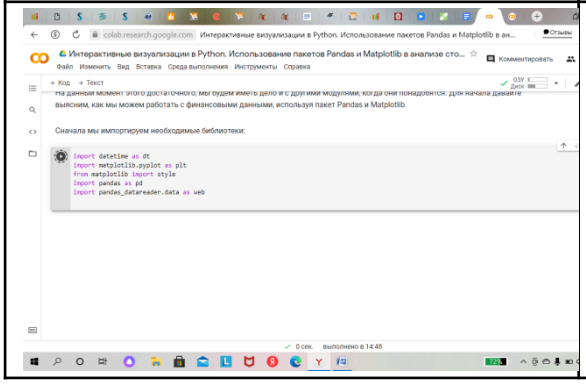


2.Назвать его «Интерактивные визуализации в Python. Использовать пакеты Pandas и Matplotlib в анализе стоимости акций»

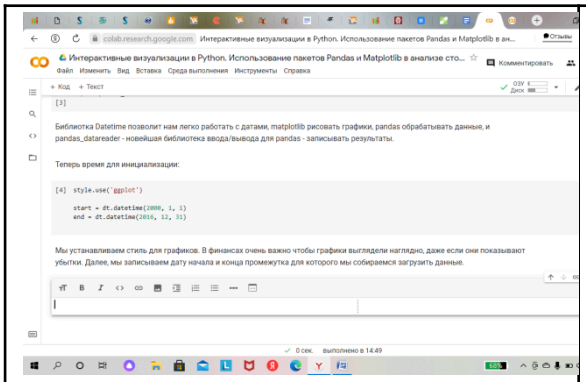


3.В работе с блокнотом можно использовать теоретические сведения и инструкции, вставляя их в текстовые ячейки, которые добавляются через нажатие на «+ Текст». Сначала нужно импортировать необходимые библиотеки, используя следующий код:

```
import datetime as dt
import matplotlib.pyplot as plt
from matplotlib import style
import pandas as pd
import pandas_datareader.data as web
```

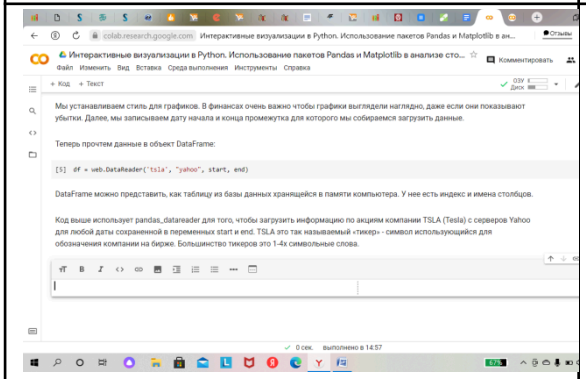


4.Необходимые коды нужно вставить в кодовые ячейки, нажимая на «+ Код». Также нужно прогрузить коды, нажимая на знак старта, это можно и делать через комбинацию Ctrl+Enter.

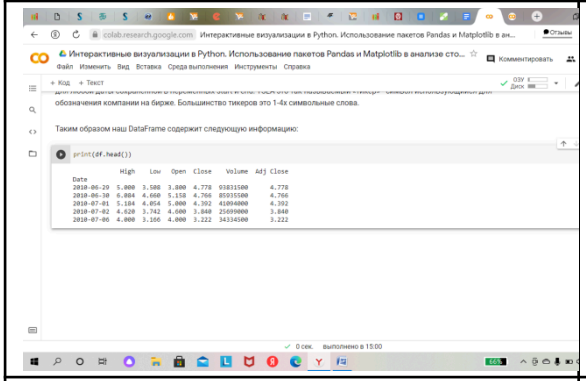


5.Теперь время для инициализации:
style.use('ggplot')

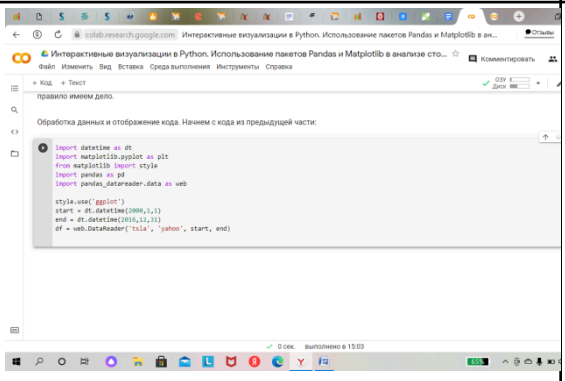
start = dt.datetime(2000, 1, 1)
end = dt.datetime(2016, 12, 31)
Установить стиль для графиков. В финансах очень важно чтобы графики выглядели наглядно, даже если они показывают убытки. Далее записать дату начала и конца промежутка для которого нужно загрузить данные.



6.Теперь прочтем данные в объект DataFrame:
df = web.DataReader('tsla', "yahoo", start, en)
DataFrame можно представить, как таблицу из базы данных хранящейся в памяти компьютера. У нее есть индекс и имена столбцов.
Код выше использует pandas_datareader для того, чтобы загрузить информацию по акциям компании TSLA (Tesla) с серверов Yahoo для любой даты сохраненной в переменных start и end. TSLA это так называемый «тикер» - символ использующийся для обозначения компании на бирже. Большинство тикеров это 1-4х символьные слова.

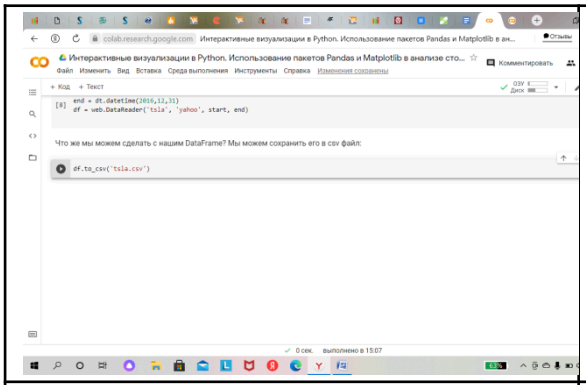
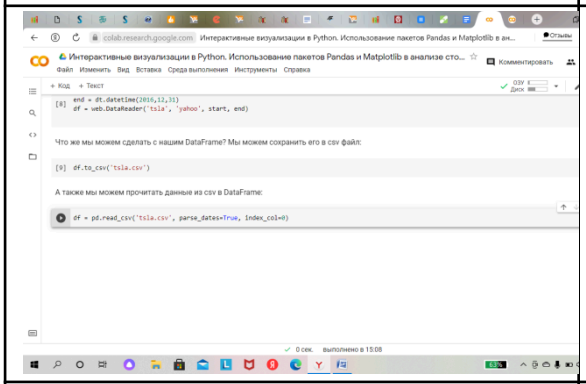
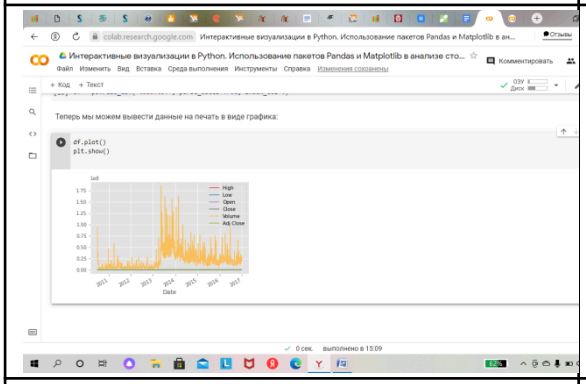
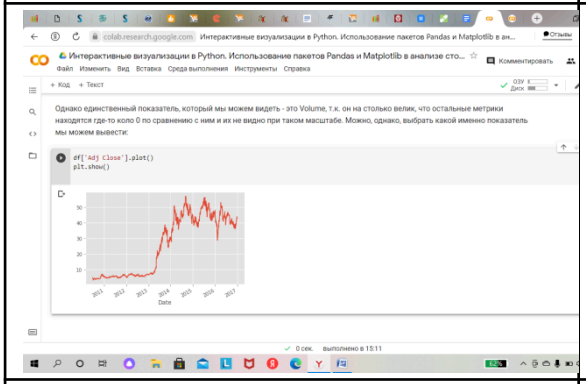
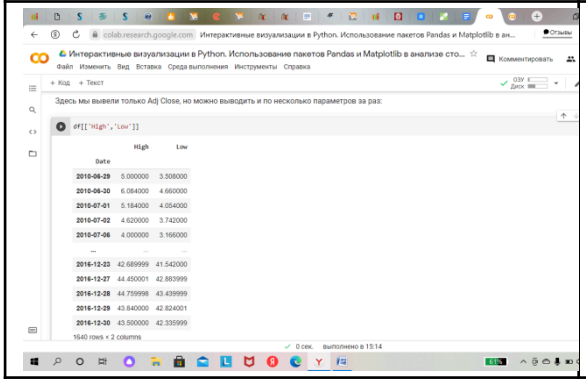


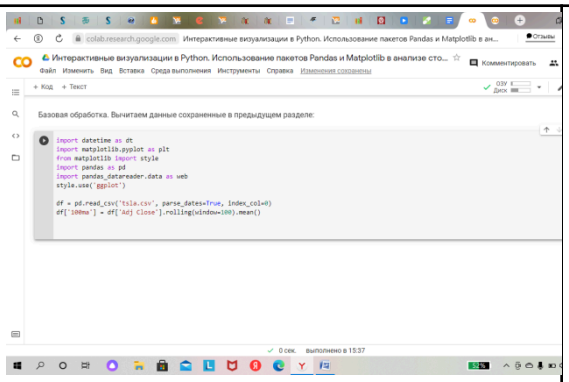
7.Таким образом, DataFrame содержит следующую информацию:
print(df.head())



8.Приступить к обработке данных и отображению кода.
Начнем с кода из предыдущей части:

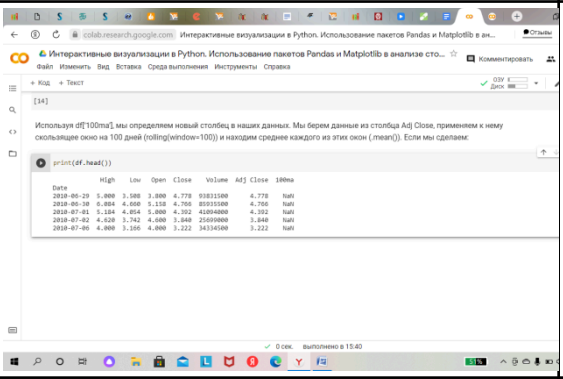
```
import datetime as dt
import matplotlib.pyplot as plt
from matplotlib import style
import pandas as pd
import pandas_datareader.data as web

style.use('ggplot')
start = dt.datetime(2000,1,1)
end = dt.datetime(2016,12,31)
df = web.DataReader('tsla', 'yahoo', start, end)
```

| | |
|---|---|
|  | 9.Можно сохранить DataFrame в csv файл:<br>df.to_csv ('tsla.csv') |
|  | 10.А также можно прочитать данные из csv в DataFrame:<br>df = pd.read_csv('tsla.csv', parse_dates=True, index_col=0) |
|  | 11.Теперь мы можем вывести данные на печать в виде графика:<br>df.plot()<br>plt.show() |
|  | 12.Однако единственный показатель, который мы можем видеть - это Volume, т.к. он на столько велик, что остальные метрики находятся где-то коло 0 по сравнению с ним и их не видно при таком масштабе. Можно, однако, выбрать какой именно показатель мы можем вывести:<br>df['Adj Close'].plot()<br>plt.show() |
|  | 13.Мы вывели только Adj Close, но можно выводить и по несколько параметров за раз:<br>df[['High','Low']]<br>Запускаем код. |



14.Переходим к базовой обработке.
Вычитаем данные сохраненные в предыдущем разделе:

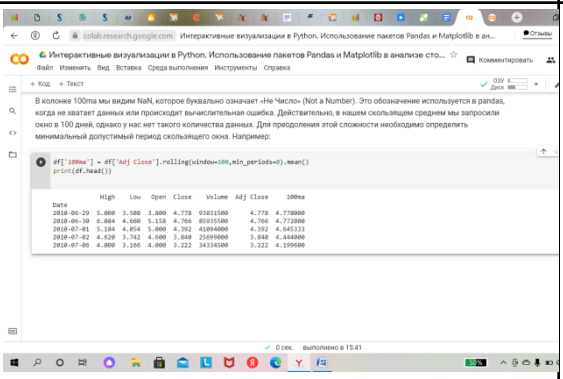
```
import datetime as dt
import matplotlib.pyplot as plt
from matplotlib import style
import pandas as pd
import pandas_datareader.data as web
style.use('ggplot')

df = pd.read_csv('tsla.csv', parse_dates=True, index_col=0)
df['100ma'] = df['Adj Close'].rolling(window=100).mean()
```



15.Используя df[‘100ma’], мы определяем новый столбец в наших данных. Мы берем данные из столбца Adj Close, применяем к нему скользящее окно на 100 дней (rolling(window=100)) и находим среднее каждого из этих окон (.mean()). Если мы сделаем:

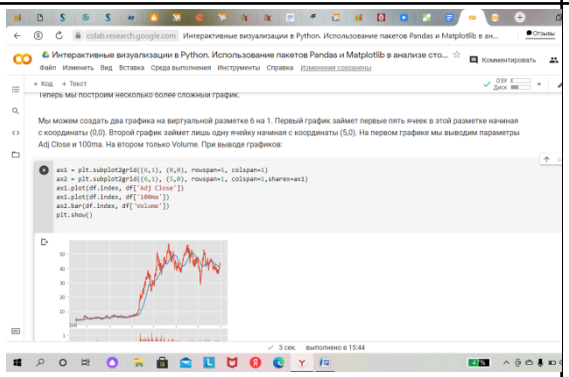
```
print(df.head())
```

, то получим: «результат приведён в виде скриншота».



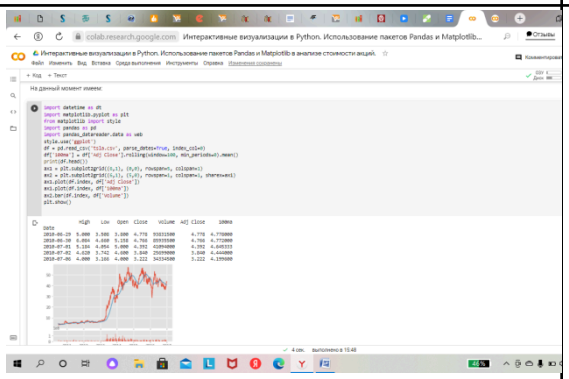
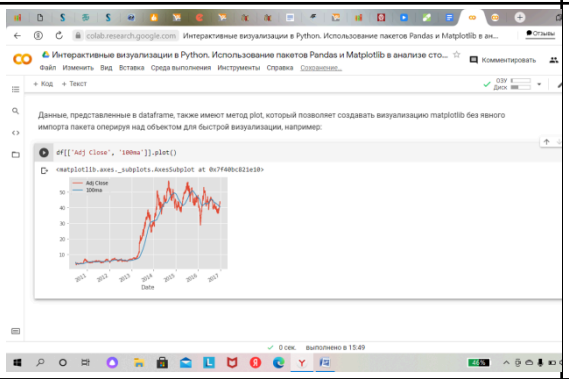
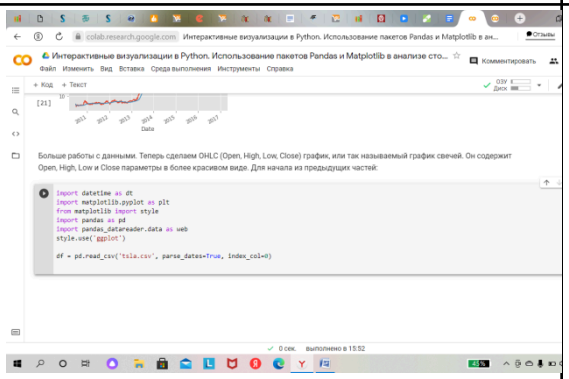
16.В колонке 100ma мы видим NaN, которое буквально означает «Не Число» (Not a Number). Это обозначение используется в pandas, когда не хватает данных или происходит вычислительная ошибка. Действительно, в нашем скользящем среднем мы запросили окно в 100 дней, однако у нас нет такого количества данных. Для преодоления этой сложности необходимо определить минимальный допустимый период скользящего окна. Например:

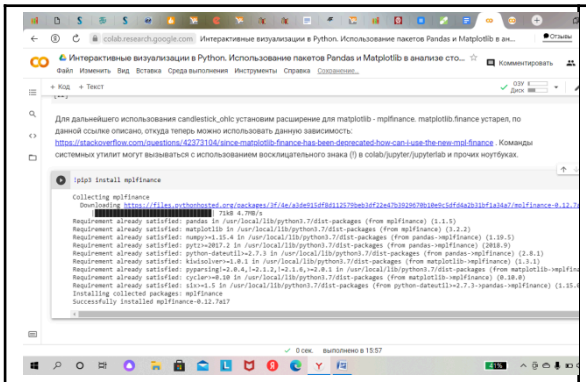
```
df['100ma'] = df['Adj Close'].rolling(window=100,min_periods=0).mean()
print(df.head())
```



17.Теперь мы построим несколько более сложный график. Мы можем создать два графика на виртуальной разметке 6 на 1. Первый график займет первые пять ячеек в этой разметке начиная с координаты (0,0). Второй график займет лишь одну ячейку начиная с координаты (5,0). На первом графике мы выводим параметры Adj Close и 100ma. На втором только Volume. При выводе графиков:

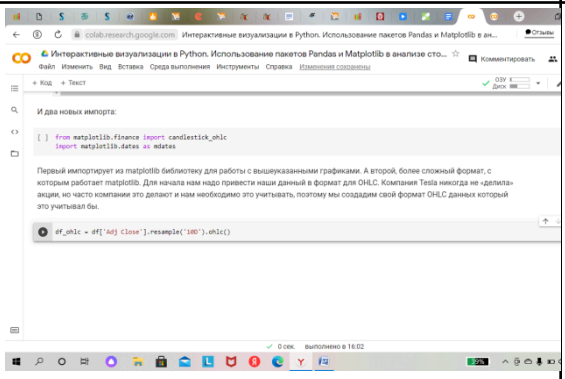
```
ax1 = plt.subplot2grid((6,1), (0,0), rowspan=5, colspan=1)
ax2 = plt.subplot2grid((6,1), (5,0), rowspan=1, colspan=1,sharex=ax1)
```

| | |
|---|---|
| | ax1.plot(df.index, df['Adj Close'])<br>ax1.plot(df.index, df['100ma'])<br>ax2.bar(df.index, df['Volume'])<br>plt.show() |
|  | 18.На данный момент имеем:<br>import datetime as dt<br>import matplotlib.pyplot as plt<br>from matplotlib import style<br>import pandas as pd<br>import pandas_datareader.data as web<br>style.use('ggplot')<br>df = pd.read_csv('tsla.csv', parse_dates=True, index_col=0)<br>df['100ma'] = df['Adj Close'].rolling(window=100, min_periods=0).mean()<br>print(df.head())<br>ax1 = plt.subplot2grid((6,1), (0,0), rowspan=5, colspan=1)<br>ax2 = plt.subplot2grid((6,1), (5,0), rowspan=1, colspan=1, sharex=ax1)<br>ax1.plot(df.index, df['Adj Close'])<br>ax1.plot(df.index, df['100ma'])<br>ax2.bar(df.index, df['Volume'])<br>plt.show()<br>Получим таблицу и диаграммы. |
|  | 19.Данные, представленные в dataframe, также имеют метод plot, который позволяет создавать визуализацию matplotlib без явного импорта пакета оперируя над объектом для быстрой визуализации, например:<br>df[['Adj Close', '100ma']].plot() |
|  | 20.Больше работы с данными.<br>Теперь сделаем OHLC (Open, High, Low, Close) график, или так называемый график свечей. Он содержит Open, High, Low и Close параметры в более красивом виде.<br>Для начала из предыдущих частей:<br>import datetime as dt<br>import matplotlib.pyplot as plt<br>from matplotlib import style<br>import pandas as pd<br>import pandas_datareader.data as web<br>style.use('ggplot')<br>df = pd.read_csv('tsla.csv', parse_dates=True, index_col=0) |



21.Для дальнейшего использования candlestick_ohlc установим расширение для matplotlib - mplfinance. matplotlib.finance устарел, по данной ссылке описано, откуда теперь можно использовать данную зависимость: https://stackoverflow.com/questions/42373104/since-matplotlib-finance-has-been-deprecated-how-can-i-use-the-new-mpl-finance
Команды системных утилит могут вызываться с использованием восклицательного знака (!) в colab/jupyter/jupyterlab и прочих ноутбуках.

```
!pip3 install mplfinance
```

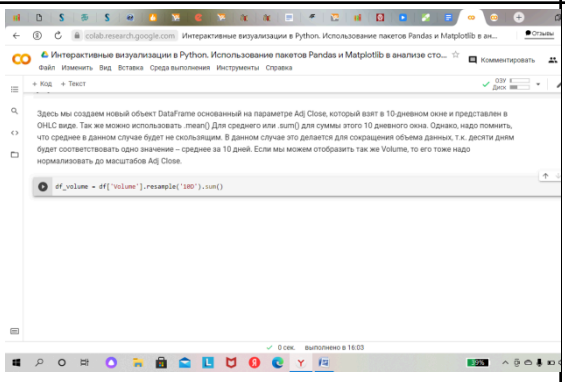


22.И два новых импорта:

```
from matplotlib.finance import candlestick_ohlc
import matplotlib.dates as mdates
```

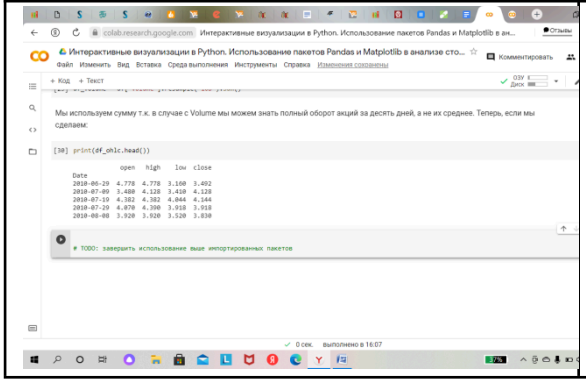
Первый импортирует из matplotlib библиотеку для работы с вышеуказанными графиками. А второй, более сложный формат, с которым работает matplotlib. Для начала нам надо привести наши данный в формат для OHLC. Компания Tesla никогда не «делила» акции, но часто компании это делают и нам необходимо это учитывать, поэтому мы создадим свой формат OHLC данных который это учитывал бы.

```
df_ohlc = df['Adj Close'].resample('10D').ohlc()
```



23.Здесь мы создаем новый объект DataFrame основанный на параметре Adj Close, который взят в 10-дневном окне и представлен в OHLC виде. Так же можно использовать .mean() Для среднего или .sum() для суммы этого 10 дневного окна. Однако, надо помнить, что среднее в данном случае будет не скользящим. В данном случае это делается для сокращения объема данных, т.к. десяти дням будет соответствовать одно значение - среднее за 10 дней. Если мы можем отобразить так же Volume, то его тоже надо нормализовать до масштабов Adj Close.

```
df_volume = df['Volume'].resample('10D').sum()
```



24.Мы используем сумму т.к. в случае с Volume мы можем знать полный оборот акций за десять дней, а не их среднее. Теперь, если мы сделаем:

```
print(df_ohlc.head())
```

то получим: «смотреть скриншот».
Пишем:

```
# TODO: завершить использование выше импортированных пакетов
```

25.Интерактивные визуализации в Python.
В Python есть множество пакетов, создающих интерактивные визуализации в Python, при этом встраивая Javascript код прямо в notebook / страницу в браузере. Одна из таких библиотек для визуализации - Plotly, уже предустановленная в окружении Colab.

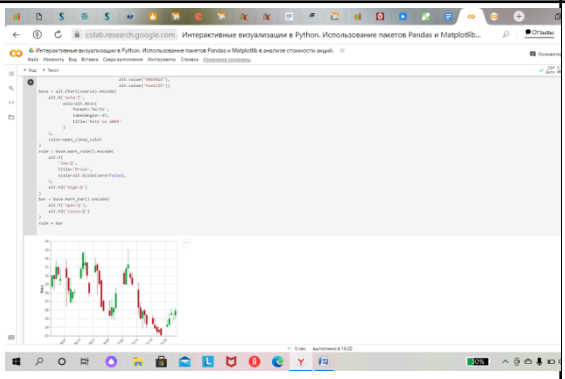
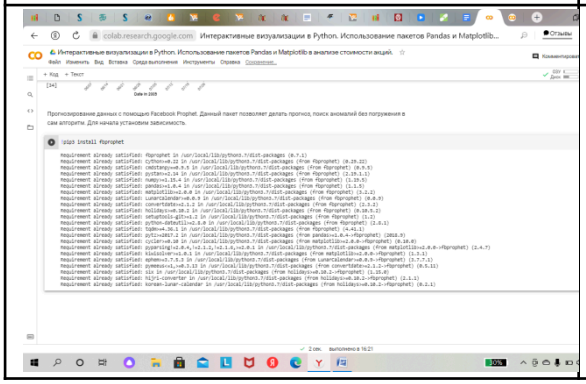
```
import plotly.graph_objects as go
import pandas as pd

fig = go.Figure(
   data=go.Ohlc(
      x=df_ohlc.index,
      open=df_ohlc["open"],
      high=df_ohlc["high"],
      low=df_ohlc["low"],
      close=df_ohlc["close"]
   )
)
fig.show()
```

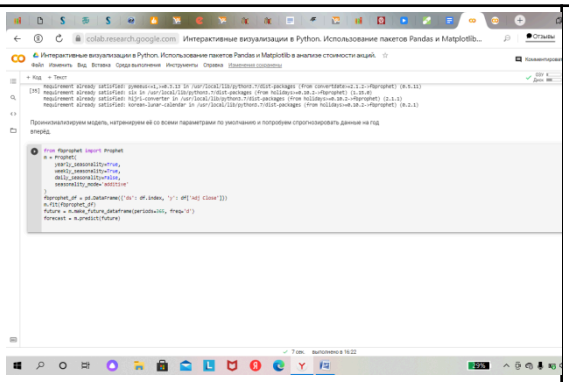
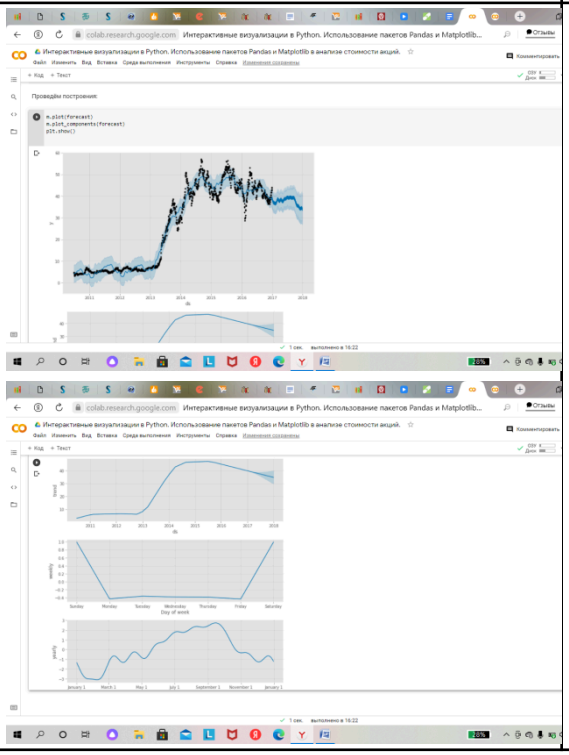
Получившийся график полностью интерактивный и имеет свой набор инструментов в правом верхнем углу для исследования данных.

26.Другая библиотека - Bokeh, также предустановлена.

```
# Для того, чтобы производить вывод графиков в notebook - colab,
# вызываем output_notebook()
from bokeh.io import output_notebook
output_notebook()
from math import pi
from bokeh.plotting import figure, output_file, show
inc = df_ohlc.close > df_ohlc.open
dec = df_ohlc.open > df_ohlc.close
w = 12*60*60*1000 # half day in ms
TOOLS = "pan,wheel_zoom,box_zoom,reset,save"
p = figure(x_axis_type="datetime", tools=TOOLS, plot_width=1000, title = "MSFT Candlestick")
p.xaxis.major_label_orientation = pi/4
p.grid.grid_line_alpha=0.3
p.segment(df_ohlc.index, df_ohlc.high, df_ohlc.index, df_ohlc.low, color="black")
p.vbar(df_ohlc.index[inc], w, df_ohlc.open[inc], df_ohlc.close[inc], fill_color="#D5E1DD", line_color="black")
p.vbar(df_ohlc.index[dec], w, df_ohlc.open[dec], df_ohlc.close[dec], fill_color="#F2583E", line_color="black")
output_file("candlestick.html", title="candlestick.py plot example")
show(p)  # open a browser
```

Получившийся график также имеет набор инструментов для исследования, например,

| | |
|---|---|
| | можно использовать "Box Zoom" для приближения к отдельному участку графика. |
|  | 27.Другой пакет для визуализаций, доступный в colab - Altair (Alt).<br>import altair as alt<br>from vega_datasets import data<br>source = data.ohlc()<br>open_close_color = alt.condition("datum.open <= datum.close",<br>alt.value("#06982d"),<br>alt.value("#ae1325"))<br>base = alt.Chart(source).encode(<br>alt.X('date:T',<br>axis=alt.Axis(<br>format='%m/%d',<br>labelAngle=-45,<br>title='Date in 2009'<br>)<br>),<br>color=open_close_color<br>)<br>rule = base.mark_rule().encode(<br>alt.Y(<br>'low:Q',<br>title='Price',<br>scale=alt.Scale(zero=False),<br>),<br>alt.Y2('high:Q')<br>)<br>bar = base.mark_bar().encode(<br>alt.Y('open:Q'),<br>alt.Y2('close:Q')<br>)<br>rule + bar |
|  | 28.Прогнозирование данных с помощью Facebook Prophet.<br>Данный пакет позволяет делать прогноз, поиск аномалий без погружения в сам алгоритм. Для начала установим зависимость.<br>!pip3 install fbprophet |

<table>
<tr>
<td>  </td>
<td>29.Проинизиализируем модель, натренируем её со всеми параметрами по умолчанию и попробуем спрогнозировать данные на год вперёд.<br>
<pre>
from fbprophet import Prophet
m = Prophet(
  yearly_seasonality=True,
  weekly_seasonality=True,
  daily_seasonality=False,
  seasonality_mode='additive'
)
fbprophet_df = pd.DataFrame({'ds': df.index, 'y': df['Adj Close']})
m.fit(fbprophet_df)
future = m.make_future_dataframe(periods=365, freq='d')
forecast = m.predict(future)
</pre>
</td>
</tr>
<tr>
<td>  </td>
<td>30.Проведём построения:<br>
<pre>
m.plot(forecast)
m.plot_components(forecast)
plt.show()
</pre>
Были получены графики прогноза с извлеченной компонентой тренда и доверительным интервалом изменений значений, а также графики изменений данных по различным сезонам.</td>
</tr>
<tr>
<td colspan="2">Задание выполнено. Отчёт сохраняем на Гугл Диске и прикрепляем в ментальной карте.</td>
</tr>
</table>