МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОННОЙ ТЕХНИКИ

Институт системной и программной инженерии

и информационных технологий (Институт СПИНТех)

**Лабораторный практикум**

Лабораторная работа 3

«Анализ данных в Jupyter Notebook»

На этом лабораторном занятии вы научитесь использовать на практике Jupyter Notebook. Эта работа должна дать вам более четкое представление о том, почему Jupyter Notebooks так популярны.

Итак, в этом компьютерном практикуме вы произведете анализ данных в блокноте Jupyter. Допустим, вы аналитик данных, и вам было поручено выяснить, как исторически менялась прибыль крупнейших компаний в США. У вас для этого есть набор данных о компаниях из списка Fortune 500, охватывающих более 50 лет с момента первой публикации списка в 1955 году, собранных из [открытого архива Fortune](http://archive.fortune.com/magazines/fortune/fortune500_archive/full/2005/" \t "_blank). CSV файл с этими данными для анализа включен в задание, вы можете найти его в директории с разработками лабораторных работ - *fortune500.csv*. Итак, ваша цель – узнать, как исторически менялась прибыль крупнейших компаний США.

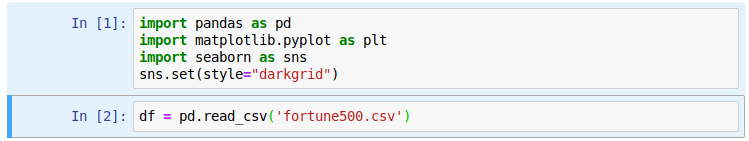
Прежде чем приступить к выполнению лабораторных заданий, настоятельно рекомендуется ознакомиться с языком программирования Python, пакетом для анализа данных [Pandas](http://pandas.pydata.org/), а также основами работы с Jupyter Notebook, подробно рассмотренными в первом компьютерном практикуме данного курса.

## **Настройка**

Обычно начинают с ячейки кода, предназначенной для импорта и настройки.

**Упражнение**

Запустите Jupyter Notebook и создайте новый блокнот, нажав кнопку «Создать» на панели инструментов в правом верхнем углу и выберите «Python 3». Дайте вашему блокноту соответствующее название из панели управления Jupyter, не забудьте, что для этого сперва необходимо закрыть ядро. Далее запустите заново ваш блокнот и напечатайте следующий код:



Здесь мы импортируем [pandas](https://pandas.pydata.org/" \t "_blank) для работы с нашими данными, [Matplotlib](https://matplotlib.org/" \t "_blank) для построения графиков и [Seaborn](https://seaborn.pydata.org/" \t "_blank) для улучшения внешнего вида наших графиков. Обычно также импортируется [NumPy](http://www.numpy.org/" \t "_blank), но в нашем случае, мы используем его через pandas. Последняя строка не является командой Python, она называется строковой магией, для инструктирования Jupyter захватывать графики Matplotlib и отображать их в выходных данных ячейки, это одна из расширенных функций, выходящих за рамки данной работы.

В следующей ячейке мы загружаем наш набор данных. Целесообразно делать это в отдельной ячейке на случай, если нам понадобится перезагрузить ее в любой момент.

## **Сохранение и контрольная точка**

Теперь, когда вы начали создавать свой первый проект, полезно будет регулярно сохраняться. Нажатие **Ctrl + S** сохранит вашу записную книжку, вызвав команду «Save and Checkpoint».

Каждый раз, когда вы создаете новую записную книжку, создается не только файл вашей записной книжки, но и файл контрольной точки, он также является файлом **.ipynb** и будет расположен в скрытом подкаталоге вашего места сохранения с именем **.ipynb\_checkpoints**. По умолчанию Jupyter каждые 120 секунд автоматически сохраняет ваш блокнот в этот файл контрольных точек, не изменяя основной файл блокнота. Когда вы сохраняете контрольную точку, файлы записной книжки и контрольной точки обновляются. Следовательно, контрольная точка позволяет вам восстановить несохраненную работу в случае непредвиденной проблемы. Вы можете вернуться к контрольной точке из меню через «*File> Revert to Checkpoint»*.

**Упражнение**

Сохраните ваш Jupyter Notebook.

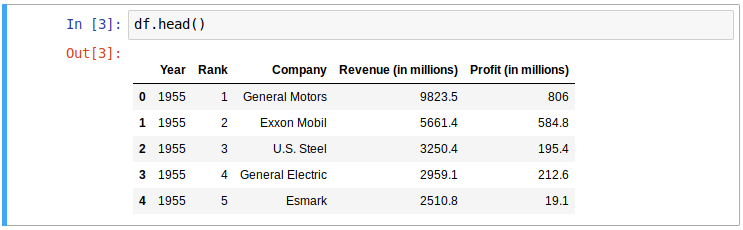
## **Изучение набора данных**

Ваш блокнот благополучно сохранен, и вы загрузили наш набор данных переменную **df** – в наиболее часто используемую структуру данных **pandas**, которая называется **DataFrame** и в основном выглядит как таблица. Давайте посмотрим, как выглядят эти данные?

**Упражнение**

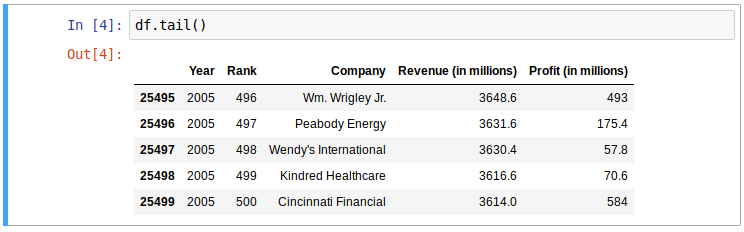
Создайте новую ячейку кода и введите команду df.head().

Результат должен выглядеть так:



Далее создайте ещё одну ячейку и введите команду df.tail().

Вы должны получить следующий результат:



Итак, у нас есть необходимые нам столбцы, и каждая строка соответствует истории каждой компании за один год.

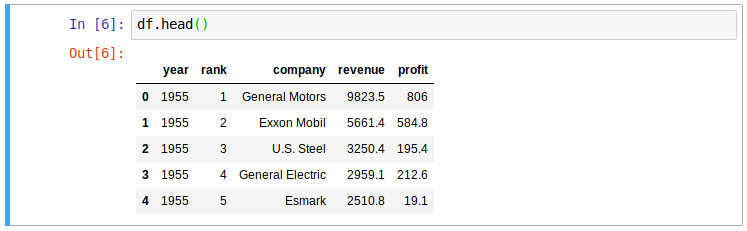
Давайте переименуем эти столбцы, чтобы мы могли обратиться к ним позже.

**Упражнение**

Создайте новую ячейку и выполните следующую команду:



Затем снова запустите ячейку с командой df.head(). Вы должны получить следующий результат:



## **Проверка набора данных**

Далее вам нужно изучить набор данных. Являются ли они завершенными? Распознало ли pandas их, как ожидалось? Отсутствуют ли в них какие-либо значения?

**Упражнение**

Создайте новую ячейку и выполните команду *len(df)*:

Вы должны получить следующий результат:



Итак, у нас есть 500 строк за каждый год с 1955 по 2005 год включительно.

Давайте проверим, был ли наш набор данных импортирован правильно. Простая проверка заключается в том, чтобы увидеть, были ли типы данных (или dtypes) правильно интерпретированы.

**Упражнение**

Создайте новую ячейку и выполните команду *df.dtypes:*

Вы должны получить следующий результат:

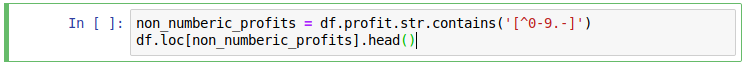


Проанализируйте полученные данные.

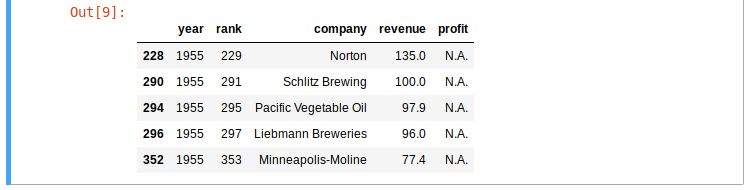
Внимательно изучите полученный результат, вы должны были заметить, что с колонкой profit что-то не так – мы ожидаем, что она должна иметь тип float64, как и колонка revenue. Данный результат указывает на то, что колонка profit, вероятно, содержит нецелые значения, давайте это проверим.

**Упражнение**

Создайте новую ячейку и выполните следующую команду*:*



Результат должен выглядеть так:

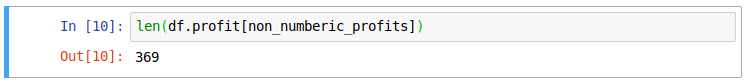


Посмотрите на колонку profit! Некоторые значения являются строками, которые использовались для указания отсутствующих данных - 'N.A.'. Что же нам с этим делать? Это зависит от того, сколько значений пропущено.

**Упражнение**

Создайте новую ячейку и выполните следующую команду *len(df.profit[non\_numberic\_profits]).*

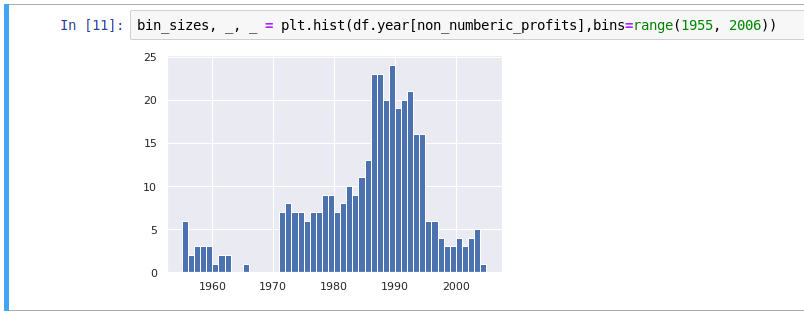
В результате вы должны получить:



Итак, мы видим, что это небольшая часть нашего набора данных, хотя и не совсем несущественная, поскольку составляет около 1,5%. Если строки, содержащие N.A., примерно одинаково распределены по годам, самым простым решением было бы просто удалить их. Давайте кратко рассмотрим их распределение.

**Упражнение**

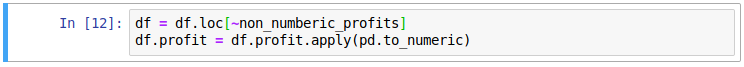
Создайте новую ячейку и выполните следующую команду*:*



На первый взгляд, мы видим, что недопустимые значения за один год составляют менее 25, а поскольку существует 500 точек данных в год, удаление этих значений будет составлять менее 4% данных для худших лет. Действительно, кроме всплеска около 90-х годов, большинство лет имеют менее половины недостающих значений пика. Допустим, что для наших целей это приемлемо, и просто удалим эти строки.

**Упражнение**

Создайте новую ячейку и выполните следующую команду*:*



Теперь давайте проверим, что у нас получилось.

**Упражнение**

Выполните заново следующие команды:





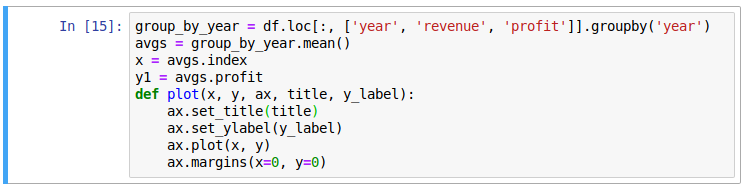
Если вы получили результат такой же как на последнем изображении, значит вы успешно завершили настройку набора данных!

## **Графики с matplotlib**

Теперь мы можем перейти к решению данного поставленной задачи, а именно, построить график средней прибыли за год и рассчитать доход. Но для начала определим некоторые необходимые переменные и метод.

**Упражнение**

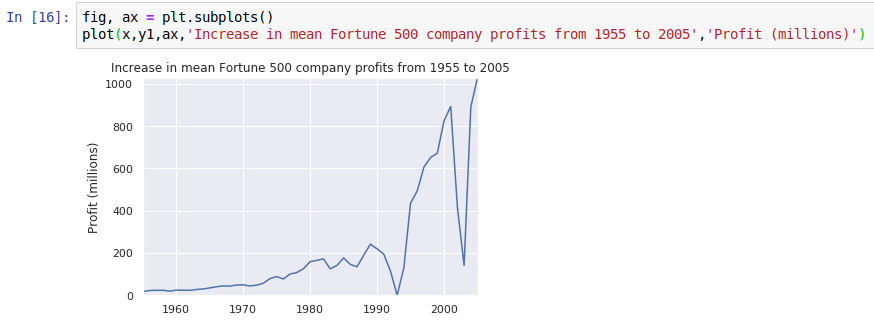
Выполните следующие команды и поясните, что они делают:



Теперь давайте наконец-то построим график прибыли!

**Упражнение**

Постройте график с помощью следующих команд:



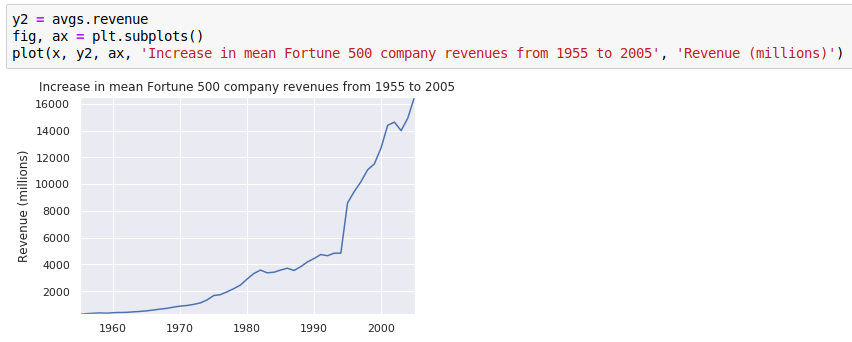
Проанализируйте полученный график.

Кажется, график нашей функции немного похож на экспоненту, но с огромными провалами. Они должны соответствовать [рецессии начала 1990-х](https://en.wikipedia.org/wiki/Early_1990s_recession" \t "_blank). Довольно интересно увидеть это в данных. Но почему прибыль возвращается к еще более высоким уровням после каждой рецессии?

Может быть, доходы могут рассказать нам больше?

**Упражнение**

Постройте график с помощью следующих команд:



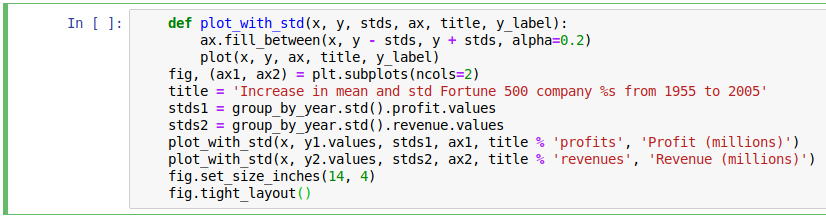
Проанализируйте полученный график.

Здесь мы видим другую сторону истории. Доходы отнюдь не так сильно пострадали!

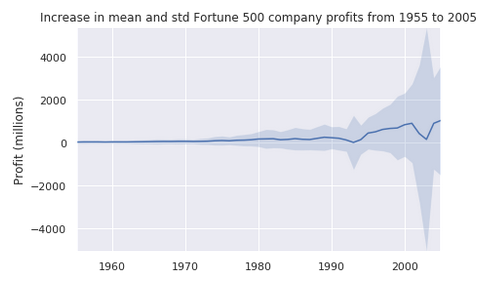
Давайте попробуем наложить эти графики с +/- их стандартными отклонениями.

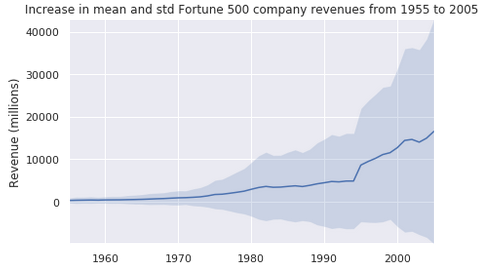
**Упражнение**

Создайте новую ячейку и выполните следующие команды*:*



Вы должны получить следующие графики:





Проанализируйте полученные результаты!

Как видите, стандартные отклонения огромны. Некоторые компании из списка Fortune 500 зарабатывают миллиарды, в то время как другие теряют миллиарды, и риск увеличивается вместе с ростом прибыли за последние годы. Возможно, некоторые компании работают лучше, чем другие.

Итак, данный блокнот помог нам легко исследовать наш набор данных в одном месте без переключения контекста между приложениями, и наша работа является доступной и воспроизводимой. Если бы мы хотели создать более краткий отчет для конкретной аудитории, мы могли бы быстро реорганизовать нашу работу, объединив ячейки и удалив промежуточный код.

**Задание к лабораторной работе**

1) Проведите анализ исторического изменения прибыли крупнейших компаний США с помощью Jupyter Notebook, последовательно выполнив все упражнения из данного практикума.

2) Представьте отчёт в виде файла **.ipynb** с результатами вашей работы.

3) Проанализируйте полученные результаты, что вы можете сказать об изменении прибыли рассмотренных крупнейших компаний? Проанализируйте последние, полученные графики, являются ли прибыли первых 10% более или менее волатильными, чем нижних 10%?