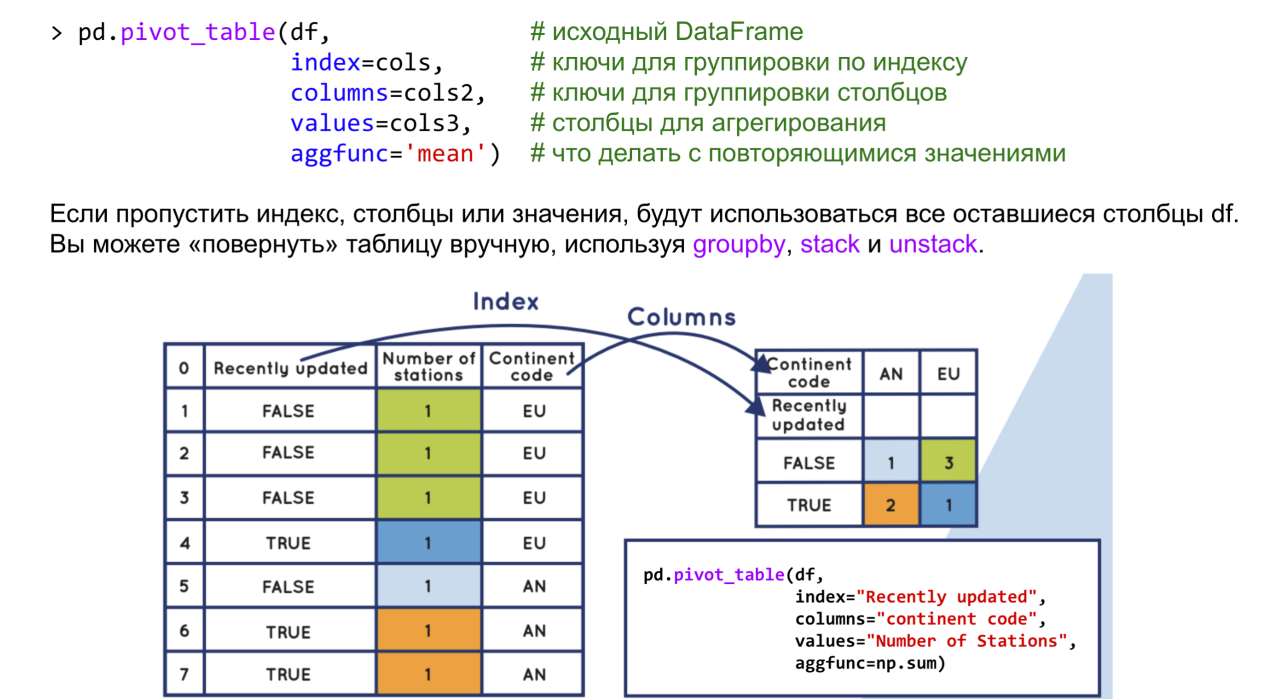
1. Есть два файла с данными турагенства: email.csv и username.csv. C ними нужно проделать все манипуляции, указанные в лекции 2, а именно:
   1. Группировка и агрегирование ( сгруппировать набор данных по значениям в столбце, а затем вычислить среднее значение для каждой группы)
   2. Обработка отсутствующих данных (заполнение отсутствующих значений определенным значением или интерполяция отсутствующих значений)
   3. Слияние и объединение данных (соединить два DataFrames в определенном столбце)

Или можно использовать файл sales.csv, который понадобится позже.

1. Преобразование данных (pivot):



1. Нужно создать сводную таблицу так, чтобы в index были столбцы “Rep”, “Manager” и “Product”, а в values “Price” и “Quantity”. Также нужно использовать функцию aggfunc=[numpy.sum] и заполнить отсутствующие значения нулями. В итоге можно будет увидеть количество проданных продуктов и их стоимость, отсортированные по имени менеджеров и директоров.
2. Учебный файл (data.csv) + практика Dataframe.pivot. Поворот фрейма данных и суммирование повторяющихся значений.
3. Визуализация данных (можно использовать любой из учебных csv-файлов).
4. Необходимо создать простой линейный график из файла csv (два любых столбца, в которых есть зависимость)
5. Создание визуализации распределения набора данных. Создать произвольный датасет через np.random.normal или использовать датасет из csv-файлов, потом построить гистограмму.
6. Сравнение нескольких наборов данных на одном графике. Создать два набора данных с нормальным распределением или использовать данные из файлов. Оба датасета отразить на одной оси, добавить легенду и название.
7. Построение математической функции. Создать данные для x и y (где x это numpy.linspace, а y - заданная в условии варианта математическая функция). Добавить легенду и название графика.
8. Вариант 1 - функция sin
9. Вариант 2 - функция cos
10. Моделирование простой анимации. Создать данные для x и y (где x это numpy.linspace, а y - математическая функция). Запустить объект line, ввести функцию animate(i) c методом line.set\_ydata() и создать анимированный объект FuncAnimation.
    1. Шаг 1: смоделировать график sin(x) (или cos(x)) в движении.
    2. Шаг 2: добавить к нему график cos(x) (или sin(x)) так, чтобы движение шло одновременно и оба графика отображались на одной оси.
11. **Загрузка CSV-файла в DataFrame.** Используя pandas, напишите скрипт, который загружает CSV-файл в DataFrame и отображает первые 5 строк df.
12. **Выбор столбцов из DataFrame.** 
    1. Используя pandas, напишите сценарий, который из DataFrame файла sales.csv выбирает только те строки, в которых Status = presented, и сортирует их по цене от меньшего к большему.
    2. Из файла climate.csv отображает в виде двух столбцов названия и коды (rw\_country\_code) тех стран, у которых cri\_score больше 100, а fatalities\_total не более 10.
    3. Из файла cars.csv отображает названия 50 первых американских машин, у которых расход топлива MPG не менее 25, а частное внутреннего объема (Displacement) и количества цилиндров не более 40. Названия машин нужно вывести в алфавитном порядке.
13. **Вычисление статистики для массива numpy**

Используя numpy, напишите скрипт, который загружает файл CSV в массив numpy и вычисляет среднее значение, стандартное отклонение и максимальное значение массива. Для тренировки используйте файл data.csv, а потом любой другой csv-файл от 20 строк.

1. **Операции с матрицами:** Используя numpy, напишите сценарий, который создает матрицу и выполняет основные математические операции, такие как сложение, вычитание, умножение и транспонирование матрицы.