1. Поработаем с данными о поездках на такси в Нью-Йорке, где также имеется информация о погодных условиях и выходных днях.

pickup\_dt – период с точностью до часа

pickup\_month – месяц

borough – район Нью-Йорка, из которого был сделан заказ (5 районов + аэропорт)

pickups – число поездок за период (час)

hday – является ли день праздничным/выходным; Y - да, N - нет

spd – скорость ветра в милях в час

vsb – видимость

temp – температура в градусах Фаренгейта

dewp – точка росы по Фаренгейту

slp – давление

pcp\_01 – количество осадков за час

pcp\_06 – количество осадков за 6 часов

pcp\_24 – количество осадков за 24 часа

sd – глубина снега в дюймах

Но перед тем, как приступать к работе, давайте загрузим данные. Импортируйте библиотеку pandas, а затем загрузите данные.

Результат сохраните в переменную taxi.

1. Проверьте, сколько всего строк и столбцов имеется в датасете.

shape возвращает кортеж (упрощённый список, относитесь к нему, как к списку), где сначала идёт число строк, а потом число колонок.

df.shape

1. Давайте посмотрим на типы колонок. Все ли из них считались правильно? Какой тип, преобладает в датасете?

df.dtypes

1. В названиях столбцов pcp 01, pcp 06, pcp 24 встречается пробел. Это не очень удобно, если в дальнейшем вы планируете обращаться к колонкам через точку, без использования кавычек и скобочек.

Замените пробел в названиях на знак нижнего подчеркивания.

Данные сохранены в переменную taxi. Данные должны быть изменены в этом же датафрейме.

1. Вас попросили узнать, сколько записей (строк) в датафрейме относятся к району Манхэттен (Manhattan). Получить ответ на данный вопрос можно было бы, например, с помощью следующей команды:

А если мы хотим посмотреть, сколько раз встречается каждый из районов? Неужели придется использовать подобную конструкцию для каждого уровня переменной? Здесь на помощь приходит метод value\_counts(),

Сколько раз в данных встречается район Бруклин (Brooklyn)?

1. Следующая задача: выяснить, из какого района было совершено наибольшее количество поездок за весь период. Для начала, посчитайте общее количество поездок (pickups), без группировки.
2. А теперь сгруппируйте данные по району (borough) и укажите, из какого пункта было совершено больше всего поездок.

использовать метод sum

1. А теперь сгруппируйте данные по району (borough) и укажите, из какого пункта было совершено больше всего поездок.
2. Еще несколько полезных методов в pandas! Возможно, в предыдущем степе вы просто посмотрели на данные или отсортировали значения. Ускорить данный процесс в дальнейшем помогут методы idxmin() и idxmax(), которые возвращают индекс минимального или максимального значения.

idxmin – индекс минимального значения

idxmax – индекс максимального значения

Сохраните название района с наименьшим числом поездок в переменную min\_pickups, применив подходящий метод.

Исходные данные сохранены в переменную taxi.

1. Продолжим изучение данных и посмотрим на число поездок в выходные дни. Сгруппируйте данные по двум признакам: району города и является ли день выходным (колонки borough и hday). Сравните среднее число поездок, и выберите районы, из которых по праздникам в среднем поступает больше заказов, чем в обычные дни.
2. Для каждого района посчитайте число поездок по месяцам. Отсортируйте полученные значения по убыванию и сохраните результирующий датафрейм в pickups\_by\_mon\_bor.

Обратите внимание, что итоговый датасет должен состоять из 3-х колонок - pickup\_month, borough, pickups.

Данные сохранены в переменную taxi.

1. Сложное задание!

Попрактикуемся в написании собственных функций и их применении к датафрейму. Поскольку данные о поездках в Нью-Йорке, температура представлена в градусах Фаренгейта.

Напишите функцию temp\_to\_celcius, которая получает на вход колонку с температурой в °F, и возвращает значения, переведенные в градусы Цельсия.

Формула:

Celsius =( Fahrenheit-32)\*5.0/9.0

Пример работы функции:

> taxi['temp'][:5]

0 30.0

1 30.0

2 30.0

3 30.0

4 30.0

> taxi['temp\_C'] = temp\_to\_celcius(taxi['temp'])

> taxi['temp\_C'][:5]

0 -1.111111

1 -1.111111

2 -1.111111

3 -1.111111

4 -1.111111

всё содержимое функции должно быть выполнено средствами Pandas - убедитесь, что на выходе она даёт pandas Series.