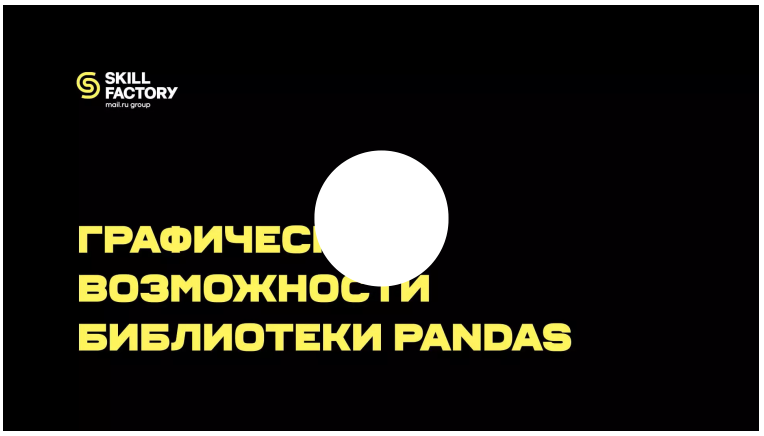
SkillFactory

4. Графические возможности библиотеки Pandas



→ Скачать ноутбук из скринкаста

✍ Дальнейшая работа будет проводиться с таблицей covid_df — результатом объединения таблиц covid_data и vaccinations_data.

Оказывается, за способами визуализации данных не нужно далеко ходить. Базовую (примитивную) визуализацию можно выполнить с помощью уже знакомой нам библиотеки *Pandas*. Функционал для создания основных

типов графиков уже включён в библиотеку и устанавливается вместе с пакетом *Anaconda* по умолчанию. С изучения возможностей *Pandas* мы и начнём наше погружение в программную визуализацию.

БАЗОВАЯ ВИЗУАЛИЗАЦИЯ В PANDAS

Основным методом для создания простейших графиков в *Pandas* является plot().

Кликните на плашку, чтобы увидеть информацию ↓

Основные параметры метода plot()

Давайте попрактикуемся в использовании метода plot().

Начнём с исследования заболеваемости коронавирусом во всём мире. Для этого первым делом отобразим, как менялось ежедневное число заболевших (*daily_confirmed*) во всём мире во времени. Далее сгруппируем таблицу по датам и подсчитаем суммарное число зафиксированных случаев по дням.

Теперь мы наконец можем построить график с помощью метода plot(). Будем использовать **линейный график** размером 12x4 (*попробуйте взять другие числа, чтобы увидеть разницу*). Подпишем график и отобразим сетку. Параметр lw *(line width*) отвечает за ширину линии для линейного графика.

```
grouped_cases = covid_df.groupby('date')['daily_confirmed'].sum()
grouped_cases.plot(
    kind='line',
    figsize=(12, 4),
    title='Ежедневная заболеваемость во времени',
    grid = True,
    lw=3
);
```

▶ Открыть примечание

На графике выше отчётливо виден умеренный начальный рост заболеваемости, после чего наблюдается её резкое повышение в середине октября 2020 года, а в декабре 2020 года — аномальная вспышка коронавируса (зафиксировано более 1.4 млн. заболевших в день). Такой резкий максимум, возможно, является ошибкой в данных и требует более детального разбора. Далее заметно постепенное уменьшение числа ежедневно фиксируемых случаев и наступление второй волны в марте 2021 года. Наконец, начиная с мая 2021 года наблюдается очередной спад.

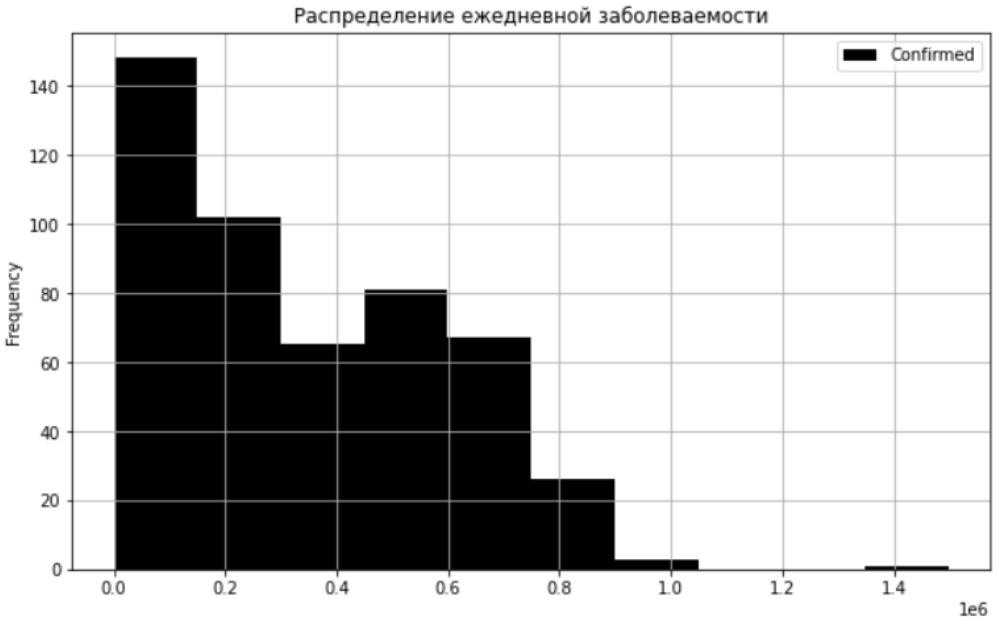
Пилообразность графика (подъёмы и спады с периодом в 7 дней) может быть связана с рабочими и выходными днями.

Нам неизвестно, как устроен во времени процесс постановки диагноза и сбора статистики и отличается ли он в разных странах. Можно предположить, что:

- спад в выходные обусловлен меньшей интенсивностью работы медицинских служб;
- в тех случаях, когда диагноз ставится по результатам анализа, если в выходные берётся/проводится меньше тестов, к понедельнику они ещё не готовы и диагноз ещё не поставлен;
- минимум в понедельник может быть запаздыванием подсчёта статистики, т.е. на самом деле данные за понедельник — это данные за воскресенье.

Теперь построим **гистограмму**, которая покажет распределение ежедневной заболеваемости во всём мире. Для этого параметр kind выставляем на значение 'hist'. Параметр bins (корзины) отвечает за число прямоугольников в гистограмме — пусть их будет 10 (*попробуйте использовать другие числа, чтобы увидеть разницу*).

```
grouped_cases.plot(
    kind='hist',
    figsize=(10, 6),
    title='Распределение ежедневной заболеваемости',
    grid = True,
    color = 'black',
    bins=10
);
```



По гистограмме можно судить о частоте попадания ежедневной заболеваемости в определённый интервал. На оси абсцисс отложен диапазон ежедневной заболеваемости (в млн человек), разбитый на десять равных интервалов, на оси ординат — число наблюдений, попавших в этот интервал.

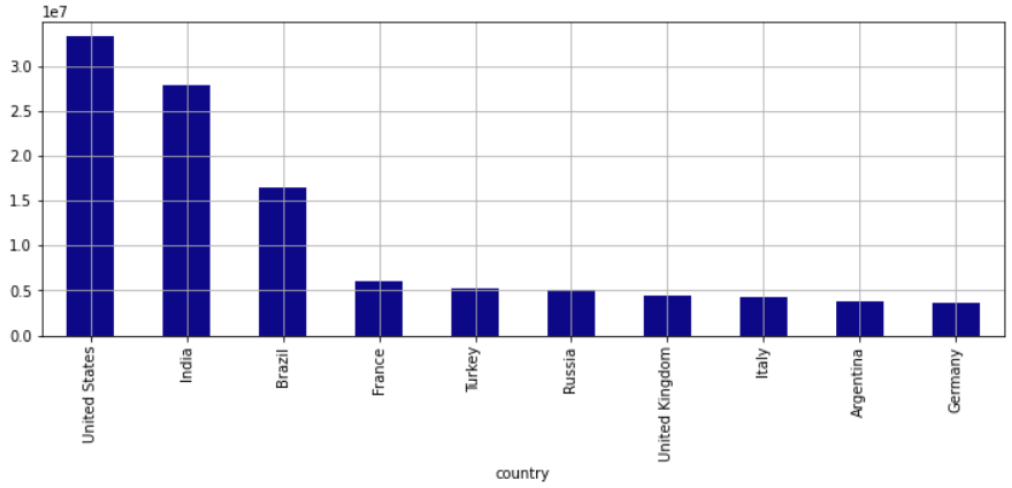
1. Мы видим модальное значение около нуля (от 0 до примерно 150 тыс. заболеваний в день), то есть большинство наблюдений сосредоточено в первом интервале, далее частота постепенно падает. Это связано с тем, что долгое время распространение вируса было довольно слабым.
2. В глаза бросается «пенёк», соответствующий резкой вспышке заболеваемости, которую мы видели ранее. Его высота очень мала, ведь такое наблюдение единственное. Даже на гистограмме кажется, что данное наблюдение является аномальным и, скорее всего, представляет собой выброс.

Давайте построим **столбчатую диаграмму**, которая покажет ТОП-10 стран по суммарной заболеваемости.

Для этого сгруппируем данные по странам и вычислим последний зафиксированный показатель с помощью агрегирующего метода last() — он возвращает последнее значение в столбце *DataFrame*.

Для построения столбчатой диаграммы значение параметра kind выставляем на 'bar'. Параметр colormap отвечает за цветовую гамму графика.

```
grouped_country = covid_df.groupby(['country'])['confirmed'].last()
grouped_country = grouped_country.nlargest(10)
grouped_country.plot(
    kind='bar',
    grid=True,
    figsize=(12, 4),
    colormap='plasma'
);
```

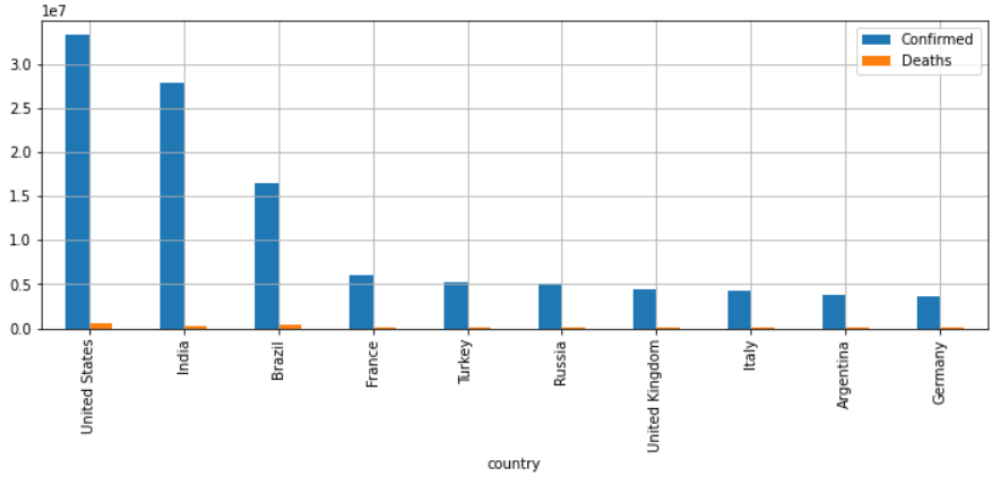


На диаграмме видно, что лидерами по числу заболевших являются Соединённые Штаты, Индия и Бразилия. Соотношение зафиксированных случаев заражения в остальных странах примерно одинаковое.

А теперь посмотрим, как в этих десяти странах соотносится количество заболевших и умерших от вируса. Для этого отобразим сразу два показателя на **столбчатой диаграмме**:

```
grouped_country = covid_df.groupby(['country'])[['confirmed',
'deaths']].last()
grouped_country = grouped_country.nlargest(10, columns=['confirmed'])
```

```
grouped_country.plot(
    kind='bar',
    grid=True,
    figsize=(12, 4),
);
```



Этот график является небольшим усовершенствованием предыдущего. Теперь на нём можно увидеть соотношение зафиксированных случаев заражения и смертей.

Очевидно, что отношение числа умерших к числу заболевших весьма низкое. Также это может косвенно говорить о разных методиках учёта заболевших (например, какие-то страны могут учитывать заболевших только по мазку, в то время как другие — по клинической картине, учитывающей и другие показатели). То есть с методической точки зрения учёт по числу заражений может быть не совсем корректным. Лучшим показателем будет являться число смертей (хотя и этот метод не идеален).

Более того, если построить график с сортировкой не по числу заболевших, а по числу умерших, поменяются и места, и страны в рейтинге. *Попробуйте построить такой график сами!*

Визуализация с помощью *Pandas* является удобным инструментом, когда графики необходимо построить «здесь и сейчас», не сильно заботясь об их внешнем виде. Однако такой подход имеет значительный минус по сравнению с использованием специализированных библиотек для визуализации — довольно ограниченный функционал:

- С помощью *Pandas* можно построить лишь базовый набор диаграмм. Для построения более сложных видов визуализации *Pandas* не подходит.
- Трудно настроить визуализацию нескольких видов графиков одновременно в разных масштабах.
- Сложно (иногда и вовсе невозможно) корректировать внешний вид графика.
- Отсутствует *3D*-визуализация.

✍ Предлагаем вам ответить на **несколько вопросов**, чтобы закрепить знания по визуализации в *Pandas* ↓

Задание 4.1

1/1 point (graded)

За выбор типа визуализации в методе `plot()` библиотеки *Pandas* отвечает параметр:

- ○ `type`
- ○ `diagram`
- ◉ `kind` ✔
- ○ `bins`

Отправить

Задание 4.2

1/1 point (graded)

Загляните в документацию по методу plot() и найдите параметр, который отвечает за установку названия оси ординат:

отвечает за установку названия оси ординат.

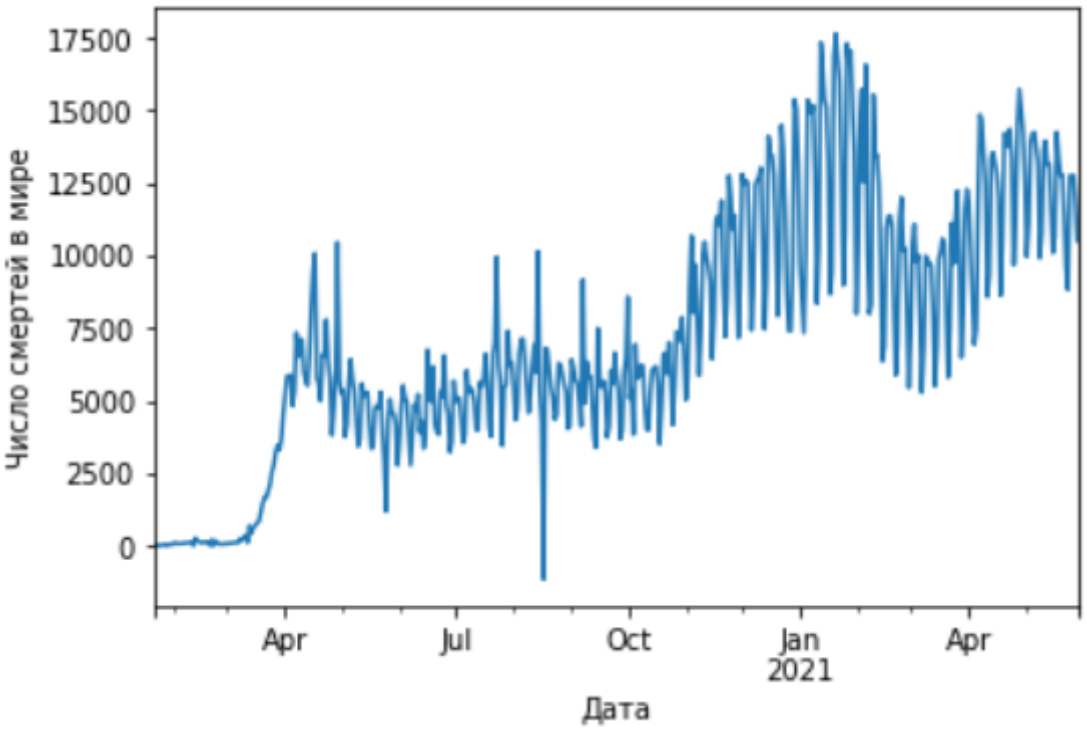
- `title`
- `xlabel`
- `yaxis`
- `ylabel` ✔

Отправить

Задание 4.3

1/1 point (graded)

С помощью какого из перечисленных ниже вариантов кода можно получить такой график?



Прежде чем заносить все варианты кода в ноутбук, попробуйте определить верный вариант ответа логически.

- ```
  covid_df.plot(x='daily_deaths', kind='line',xlabel='Дата', ylabel='Число смертей в мире');
  ```

- ```
  grouped_cases = covid_df.groupby('date')['daily_deaths'].sum()
  grouped_cases.plot(kind='line', xaxis='Дата', yaxis='Число смертей в мире');
  ```

- ```
  grouped_cases = covid_df.groupby('date')['daily_deaths'].sum()
  grouped_cases.plot(kind='bar',xlabel='Дата', ylabel='Число смертей в мире');
  ```

- ```
  grouped_cases = covid_df.groupby('date')['daily_deaths'].sum()
  grouped_cases.plot(kind='line',xlabel='Дата', ylabel='Число смертей в мире');
  ```
  ✔

Отправить

Задание 4.4

1/1 point (graded)

С помощью какого из перечисленных ниже вариантов кода можно построить столбчатую диаграмму для пяти стран с наименьшим общим числом вакцинаций на последний день рассматриваемого периода (*total_vaccinations*)?

В данном задании мы используем метод nsmallest(), который позволяет выбрать `n` наименьших значений в *Series*, а также метод last(), с помощью которого можно получить первое непустое значение в группах.

- ```
  covid_df.groupby(['country'])
  ['total_vaccinations'].last().nsmallest(5).plot(kind='bar');
  ```
  ✔

- ```
  covid_df.groupby(['country'])
  ['total_vaccinations'].first().nsmallest(5).plot(kind='bar');
  ```

- ```
  covid_df.groupby(['country'])
  ['total_vaccinations'].last().nlargest(5).plot(kind='box');
  ```

- ```
  covid_df.groupby(['date'])
  ['total_vaccinations'].mean().nsmallest(5).plot(kind='bar');
  ```

Отправить

Задание 4.5

1/1 point (graded)

Постройте график из задания 4.4. В какой стране число вакцинированных

наименьшее?

- ○ Россия
- ○ Гвинея-Бисау
- ○ Южный Судан
- ◉ Центральноафриканская Республика ✔
- ○ Китай

Отправить