SkillFactory

6. Графические возможности библиотеки Seaborn

→ **Скачать ноутбук из скринкаста**

✍ Если вы подумали, что *Matplotlib* — это потолок визуализации в *Python*, то спешим сообщить вам, что это не так, и представляем **Seaborn**.

НЕМНОГО О БИБЛИОТЕКЕ

> Seaborn — надстройка над библиотекой *Matplotlib*, которая значительно расширяет её возможности, позволяя создавать графики более высокого уровня с эстетичным оформлением. Библиотека предоставляет большое количество дополнительных опций для творчества при визуализации данных.

Установка библиотеки стандартна. В командной строке (или командной строке *Anaconda*) выполните следующее:

```
pip install seaborn
```

Традиционно *Seaborn* импортируется под псевдонимом sns:

```
import seaborn as sns
print(sns.__version__)
```

Если импорт прошёл успешно, вы увидите на экране вашу версию библиотеки. Теперь можно начинать работу!

В данном разделе мы будем сравнивать несколько стран: Россию, Австралию, Германию, Канаду и Великобританию. Создадим специальный *DataFrame* croped_covid_df для этих данных.

Для фильтрации по списку значений используем метод isin(), который проверяет, есть ли запись в столбце в переданном в метод списке. В результате возвращается привычная нам маска.

⭐ А теперь снова немного магии *Feature Engineering*, чтобы показатели по странам стали сопоставимыми: добавим информацию о населении стран, чтобы рассчитать ежедневную заболеваемость на 100 человек — заболеваемость в процентах от общего количества населения (*daily_confirmed_per_hundred*).

```
countries = ['Russia', 'Australia', 'Germany', 'Canada', 'United Kingdom']
croped_covid_df = covid_df[covid_df['country'].isin(countries)]

populations = pd.DataFrame([
    ['Canada', 37664517],
    ['Germany', 83721496],
    ['Russia', 145975300],
    ['Australia', 25726900],
    ['United Kingdom', 67802690]
    ],
    columns=['country', 'population']
)
croped_covid_df = croped_covid_df.merge(populations, on=['country'])
croped_covid_df['daily_confirmed_per_hundred'] =
croped_covid_df['daily_confirmed'] / croped_covid_df['population'] * 100
croped_covid_df.head()
```

Начнём с **гистограммы**. Для визуализации гистограмм в библиотеке *Seaborn* используется метод histplot().

У данного метода (как и у всех методов библиотеки *Seaborn*) огромнейшее количество параметров. Мы приведём лишь основные.

Кликните на плашку, чтобы увидеть информацию ↓

Основные параметры метода histplot()

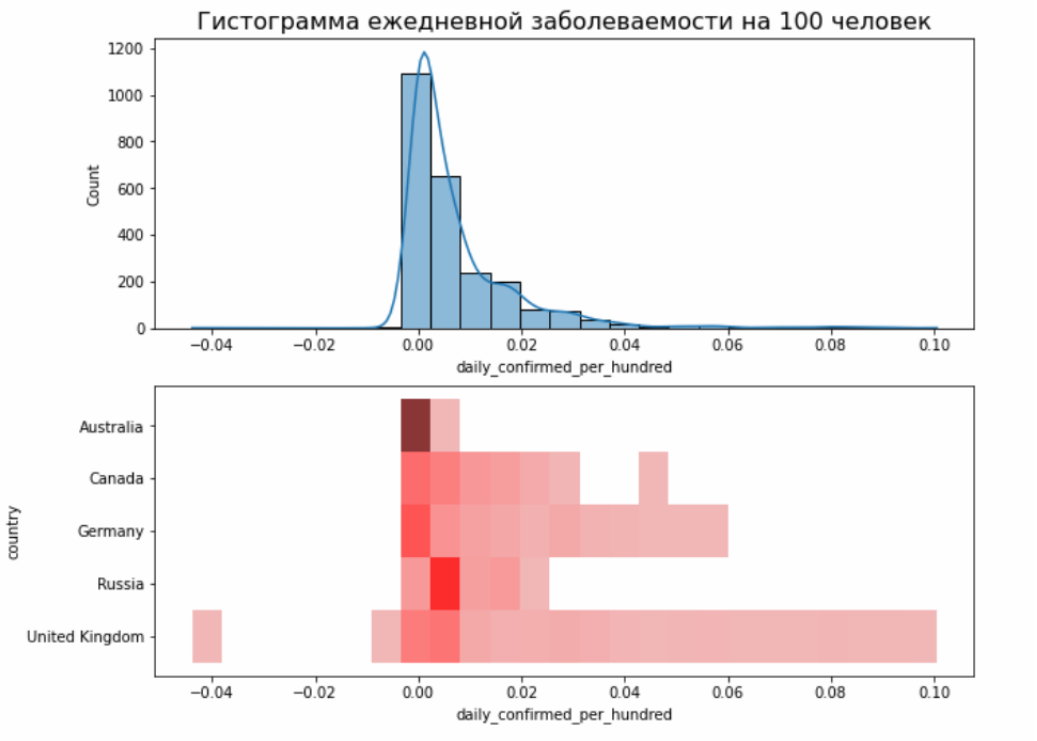
Построим сразу две гистограммы: одна будет иллюстрировать общее распределение ежедневной заболеваемости (*daily_confirmed*), а вторая — то же распределение в разрезе стран.

Для этого создаём две координатные плоскости с помощью метода subplots().

На первой координатной плоскости рисуем простую гистограмму с 25 столбцами, а также добавим на неё сглаживающую кривую.

На второй гистограмме добавляем параметр названия страны по оси *y*. В таком случае количество наблюдений будет обозначаться на диаграмме яркостью цвета (чем темнее полоса, тем больше наблюдений находится в интервале).

```
fig, axes = plt.subplots(nrows=2, ncols=1, figsize=(10, 8))
sns.histplot(
    data=croped_covid_df,
    x='daily_confirmed_per_hundred',
    bins=25,
    kde=True,
    ax=axes[0]
);
axes[0].set_title('Гистограмма ежедневной заболеваемости на 100 человек', fontsize=16)
sns.histplot(
    data=croped_covid_df,
    x='daily_confirmed_per_hundred',
    y='country',
    bins=25,
    color='red',
    ax=axes[1]
);
```



Общая гистограмма показывает, что ежедневная заболеваемость в выбранных странах не превышает 0.1 % от общего количества населения, причём большая часть наблюдений сосредоточена около 0.

Также отчётливо видны аномалии — маленькие «пеньки», где заболеваемость отрицательная.

Гистограмма по странам показывает, какой вклад в общее распределение вносит заболеваемость в каждой из стран по отдельности. Например, ясно, что пик около нуля на общей гистограмме в основном задаётся Австралией, так как в ней ежедневная заболеваемость не превышала 0.005 % от общего числа населения (около 1.5 тыс. человек в день) и все наблюдения сосредоточены в двух интервалах. Чуть больший разброс по числу фиксируемых в день случаев имеет Россия, затем идут Канада, Германия и Великобритания.

Отличительной особенностью распределения для России и Великобритании является то, что для них характерен больший процент заболевших (самая тёмная отметка находится правее, чем у других стран).

Наконец, видно, что аномальная отрицательная заболеваемость принадлежит Великобритании. Об аномалиях, их поиске как с помощью визуализации, так и иными методами мы ещё будем говорить в модуле по очистке данных.

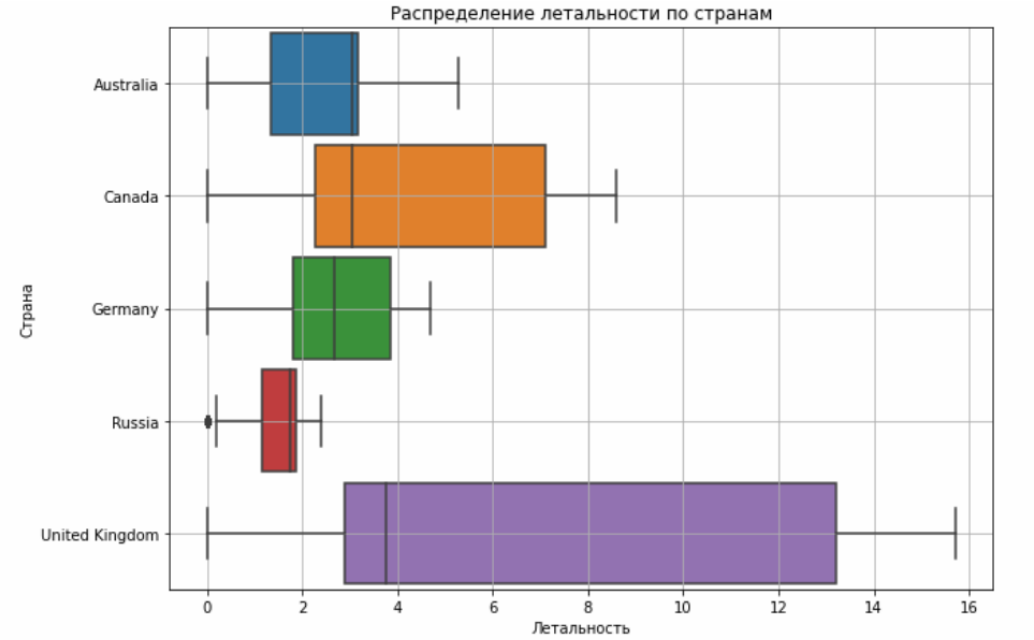
На самом деле при использовании *Seaborn* необязательно передавать координатные плоскости в аргументы функций. Графики в библиотеке вполне себе самодостаточны: функции для построения графика возвращают объект *Axes* из библиотеки *Matplotlib*, с которыми мы уже умеем работать. Достаточно просто занести результат выполнения функции в переменную и использовать её для настройки графика.

Рассмотрим пример — построим **коробчатые диаграммы** признака летальности (*death_rate*), который вы создавали ранее в задании 3.3.

Коробчатые диаграммы в *Seaborn* строятся с помощью метода boxplot().

Ящики отразим горизонтально (для этого по оси *x* отложим признак летальности, а по оси *y* — страны), параметр orient отвечает за ориентацию диаграммы, а width — за ширину коробок:

```
fig = plt.figure(figsize=(10, 7))
boxplot = sns.boxplot(
    data=croped_covid_df,
    y='country',
    x='death_rate',
    orient='h',
    width=0.9
)
boxplot.set_title('Распределение летальности по странам');
boxplot.set_xlabel('Летальность');
boxplot.set_ylabel('Страна');
boxplot.grid()
```



Из выделенных стран наиболее стабильная во времени летальность от коронавируса — в России (ширина ящика наименьшая), она же является самой низкой (наименьшая медиана). Наибольший разброс имеет процент смертей в Великобритании, что объясняется вирусологами и британскими СМИ как неподготовленность страны к эпидемии в её начале, что приводило к высокой летальности. Однако, судя по тому что медианное значение летальности в стране практически совпадает со всеми остальными, можно сказать, что со временем обстановка стабилизировалась.

Теперь рассмотрим пример **многоуровневой столбчатой диаграммы**. С помощью неё мы можем, например, посмотреть на средний ежедневный процент заболевших в странах по кварталам.

Для построения столбчатых диаграмм в *Seaborn* используется метод barplot(). По умолчанию метод отображает среднее по столбцу, который указан в параметре x (вместо среднего можно вычислить и любую другую статистическую характеристику, наименование которой задаётся в параметре estimator). Для добавления многоуровневости используется параметр hue, который позволяет группировать данные по признаку:

```
fig = plt.figure(figsize=(10, 7))
croped_covid_df['quarter'] = croped_covid_df['date'].dt.quarter
barplot = sns.barplot(
    data=croped_covid_df,
    x='country',
    y='daily_confirmed_per_hundred',
    hue='quarter',
)
barplot.set_title('Средний процент болеющего населения по кварталам');
```

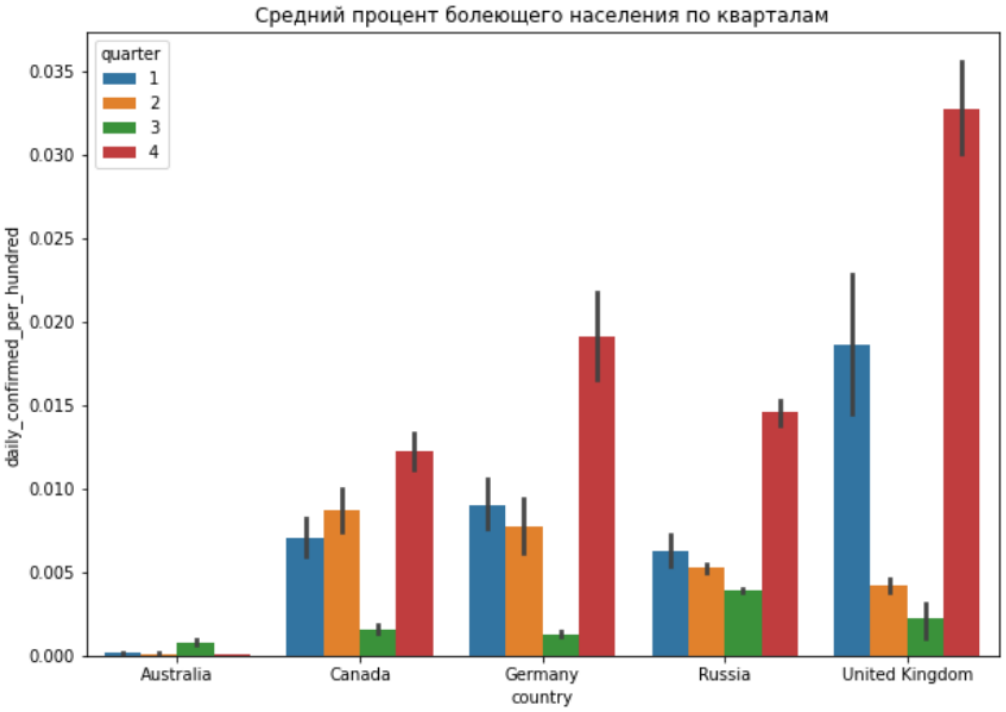


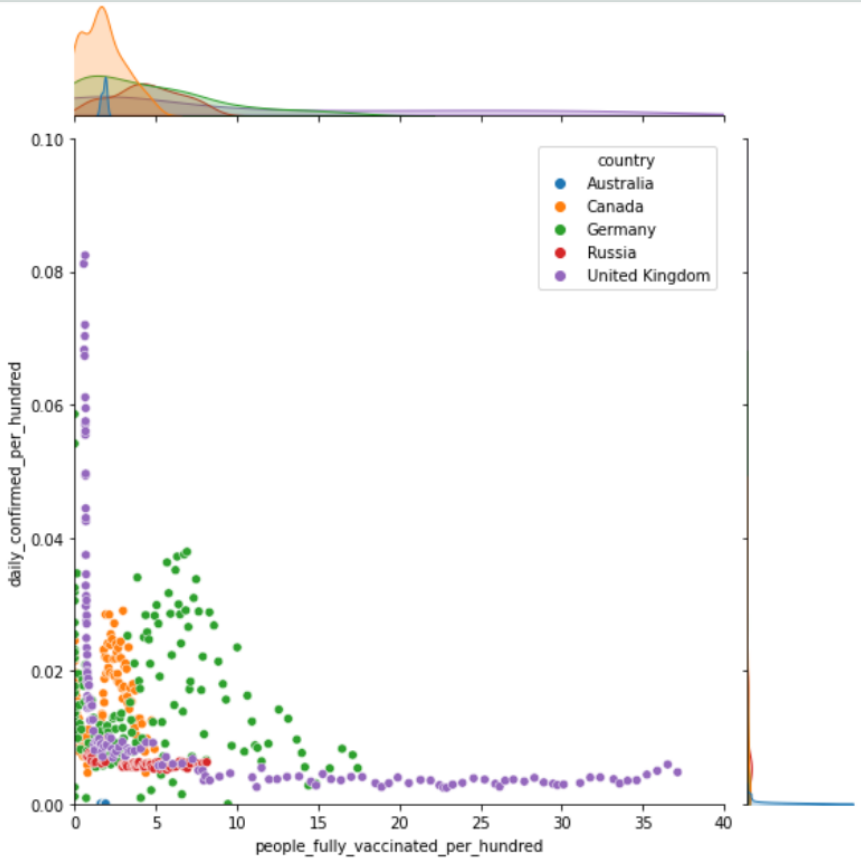
Диаграмма показывает, как зависит средний ежедневный процент заболевших от страны и квартала. Отчётливо видно, что во всех странах (кроме Австралии), большинство людей заболевают в четвёртом квартале (октябрь, ноябрь, декабрь), когда иммунитет особенно ослаблен, а наименьшее число заболевших соответствует третьему кварталу (июль, август, сентябрь).

Построим один из самых любимых дата-сайентистами графиков — jointplot() — в котором совмещены диаграмма рассеяния и гистограмма. Это довольно удобный и полезный инструмент, когда мы хотим одновременно посмотреть и на распределения переменных, и сразу оценить их взаимосвязь.

Построим *jointplot* зависимости ежедневной заболеваемости в странах от общей численности населения в процентах (*daily_confirmed_per_hundred*) от числа полностью вакцинированных в процентах (*people_fully_vaccinated_per_hundred*).

Параметры xlim и ylim определяют диапазон отображения осей *x* и *y*. Параметр height отвечает за высоту и ширину графика (он квадратный).

```
jointplot = sns.jointplot(
    data=croped_covid_df,
    x='people_fully_vaccinated_per_hundred',
    y='daily_confirmed_per_hundred',
    hue='country',
    xlim = (0, 40),
    ylim = (0, 0.1),
    height=8,
)
```



Из графика для Великобритании и России наблюдается следующая тенденция: с увеличением числа полностью привитых людей уменьшается ежедневное число заболевших. Для Канады и Германии такая же тенденция наблюдается только после достижения отметки в 7 % полностью привитого населения. Однако это может быть стечением обстоятельств, так как вирусологи говорят о необходимости полного вакцинирования 60 % населения в стране для снижения заболеваемости.

Допустим, мы хотим сравнить темпы вакцинации по странам во времени. Вы, скорее всего, сразу подумали о линейном графике. Но давайте мыслить шире. Когда мы хотим сравнить скорость изменения показателей по малому количеству категорий (в данном случае — по странам, а их у нас их всего пять), нагляднее всего будет **тепловая карта**.

Предварительно создадим сводную таблицу: по столбцам отложим признак даты, а по строкам — страны. В ячейках таблицы будет находиться процент вакцинированных (первым компонентом) людей в стране на определённую дату. Чтобы даты отображались на тепловой карте верно, их необходимо привести к типу string.

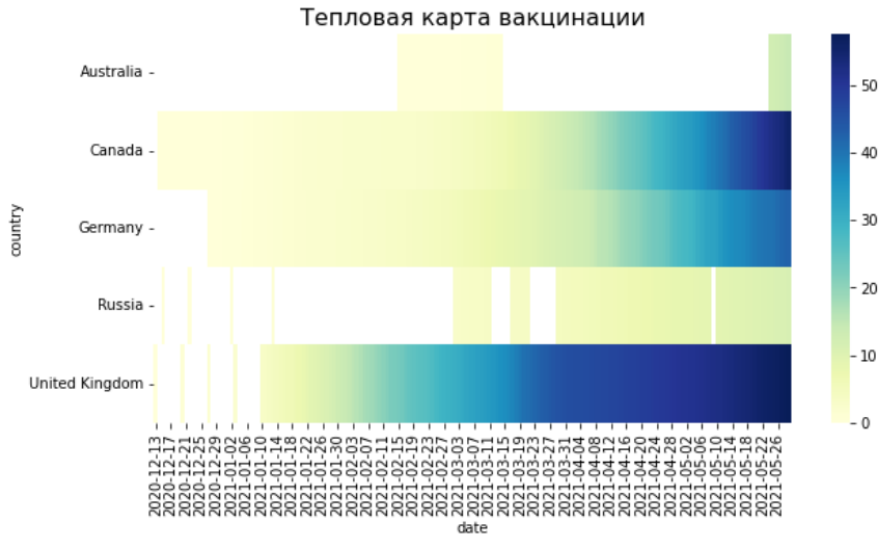
```
pivot = croped_covid_df.pivot_table(
    values='people_vaccinated_per_hundred',
    columns='date',
    index='country',
)
pivot.columns = pivot.columns.astype('string')
display(pivot)
```

date	2020-12-13	2020-12-14	2020-12-15	2020-12-16	2020-12-17	2020-12-18	2020-12-19	2020-12-20	2020-12-21	2020-12-22	...	2021-05-20	202 05-2
country													
Australia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	Na
Canada	NaN	0.0	0.00	0.01	0.02	0.03	0.03	0.03	0.05	0.07	...	48.10	49.2
Germany	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	39.18	39.6
Russia	NaN	NaN	0.02	NaN	NaN	NaN	NaN	NaN	NaN	0.04	...	10.20	10.3
United Kingdom	0.13	NaN	NaN	NaN	NaN	NaN	NaN	0.99	NaN	NaN	...	55.01	55.3

5 rows × 168 columns

Для построения тепловой карты в *Seaborn* используется метод heatmap(). Данный метод работает с табличными данными и визуализирует все ячейки таблицы с помощью цвета. Параметр annot отвечает за отображение легенды (аннотаци параметр cmap — за цветовую гамму графика.

```
heatmap = sns.heatmap(data=pivot, cmap='YlGnBu')
heatmap.set_title('Тепловая карта вакцинации', fontsize=16);
```



По тепловой карте легко можно понять, в каких странах темпы вакцинации выше, а в каких — ниже. Согласно легенде справа, чем ближе цвет полосы к синему, тем больше процент вакцинированных людей. Чем быстрее полоса переходит от блёклого жёлтого к насыщенному синему, тем выше темп вакцинации. Белые полосы обозначают отсутствие информации за данный период.

Так, можно судить, что наиболее активно кампания по вакцинации проходила в Великобритании, и на конец периода число вакцинированных первым компонентом людей в стране превысило отметку в 50 % от общего числа населения. В Канаде вакцинация населения вначале проходила медленнее, однако к концу периода наблюдений общий процент вакцинированных первым компонентом сравнялся с Великобританией.

Темпы вакцинации в России и Австралии гораздо ниже: здесь число привитых на конец периода составляет около 10 % от общего числа населения.

При этом с помощью тепловой карты мы смогли увидеть, что в данных о вакцинации в России, Великобритании и Австралии содержатся пропуски, и мы даже можем узнать, за какие периоды, посмотрев на ось абсцисс.

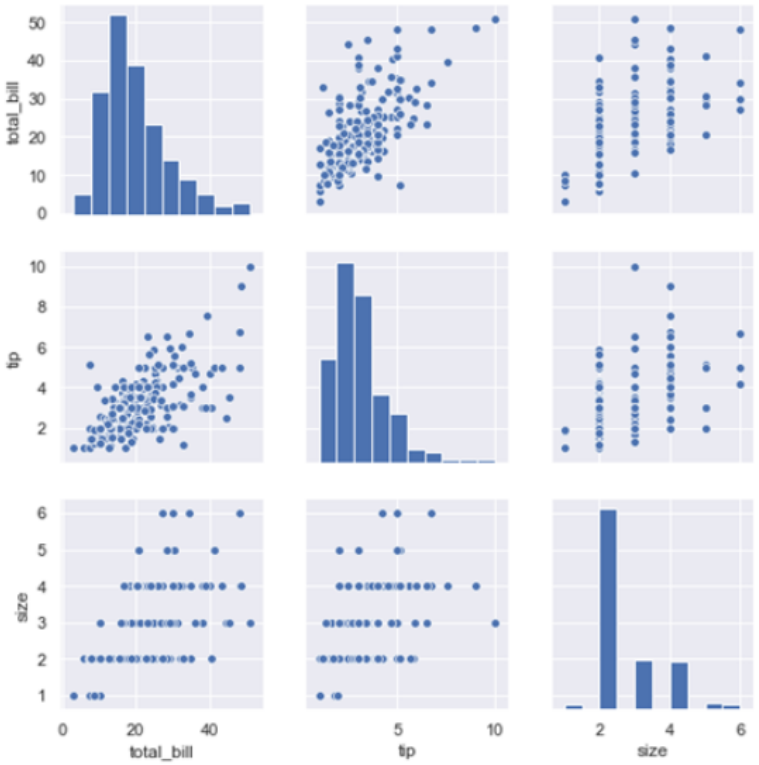
✍ Как вы уже успели понять, *Seaborn* расширяет границы визуализации, делает её более информативной и понятной. Графики становятся более приятными визуально, и их вполне можно использовать для отчёта по анализу рынка, демонстрации результатов моделирования и т. д.

Предлагаем вам **потренироваться в использовании Seaborn** ↓

Задание 6.1

1/1 point (graded)

Найдите в документации по библиотеке Seaborn название графика, который позволяет строить матрицу из диаграмм рассеяния для всех пар числовых признаков, а на диагонали этой матрицы отображаются гистограммы. Например:



- ○ `histplot()`
- ○ `kdeplot()`
- ○ `contour()`
- ◉ `pairplot()` ✔

Ответ

Верно:
Данный метод предназначен для построения отображения попарной зависимости между всеми числовыми признаками в виде диаграмм рассеяния.

Отправить

Задание 6.2

1/1 point (graded)

С помощью какого параметра в графиках *Seaborn* можно группировать данные по признаку и отображать каждую категорию разным цветом?

- `color`
- `ax`
- `palette`
- `hue` ✔

Ответ
Верно:
Данный параметр позволяет строить графики для каждой уникальной категории признака, который указан в этом параметре (группировка данных по цветам).

Отправить

Задание 6.3

1/1 point (graded)

Какой график строит код ниже?

```
sns.barplot(
    data=croped_covid_df,
    x='country',
    y='total_vaccinations_per_hundred',
    estimator=max
)
```

- ○ Гистограмму распределения ежедневной вакцинации в разрезе стран
- ◉ Столбчатую диаграмму, отображающую процентное отношение вакцинированных людей к общей численности населения страны ✔
- ○ Столбчатую диаграмму, отображающую среднее число вакцинированных в день людей по странам
- ○ Столбчатую диаграмму, отображающую максимальное число вакцинированных в день людей по странам

Отправить

Задание 6.4

1/1 point (graded)

Создайте новый признак `confirmed_per_hundred`, который покажет процентное отношение заболевших вирусом к общему числу населения в странах ($confirmed/population * 100$).

Постройте тепловую карту, которая покажет, как росло число заболевших в процентах от общего числа населения (`confirmed_per_hundred`) в странах из таблицы `croped_covid_df` .

Выберите верные выводы по построенному графику:

- ☑ **A** Из представленных стран самые быстрые темпы роста относительной заболеваемости — в Великобритании.

- [ ] **B** Суммарное число заболевших в Австралии превышает 2 % от общего числа населения.

- [ ] **C** Интенсивность относительной заболеваемости в России выше, чем в Германии.

- [x] **D** Из представленных стран самая низкая скорость распространения вируса — в Австралии.

✔

Ответ
Верно:
A Верно.
D Верно.

Отправить

Задание 6.5

1/1 point (graded)
Постройте коробчатую диаграмму для признака `recover_rate` (отношение выздоровлений к числу зафиксированных случаев заболевания в процентах).

Выберите верные выводы по данному графику:

- [x] **A** График для Великобритании имеет «сплюснутый» в нуле вид, что указывает либо на практически полное отсутствие случаев выздоровления, либо, с точки зрения здравого смысла, на неверные данные о числе выздоровевших пациентов в этой стране.

- [x] **B** Наибольший разброс по проценту ежедневных выздоровлений — в Канаде.

- [ ] **C** Наименьшая медиана — у Германии.

- [ ] **D** В четырёх из пяти стран на графике медианный процент ежедневных выздоровлений превышает 80 %.

Ответ
Верно:
A Верно.
B Верно.

Отправить