

Шпаргалка

Присоединение датафреймов

Тип присоединения	Аналог в SQL	Описание
inner	INNER JOIN	Оставляет только те строки, в которых совпадают значения столбцов, используемых для присоединения
left	LEFT JOIN	Сохраняет все строки из левого датафрейма и добавляет соответствующие строки из правого, а для несовпадающих значений будут пропуски
right	RIGHT JOIN	Похоже на левое присоединение, но в этом случае сохраняются все строки из правого датафрейма. Для несовпадающих значений также будут пропуски
outer	FULL JOIN	Присоединяет все соответствующие строки из обоих датафреймов, добавляя пропуски для несовпадающих значений

Методы присоединения

Метод присоединения	Описание	Синтаксис	Параметры
<code>merge()</code>	Позволяет присоединять данные по любым столбцам	<code>таблица1.merge(таблица2, on='столбец', how='тип_присоединения') pd.merge(таблица1, таблица2, on='столбец', how='тип_присоединения')</code>	<ul style="list-style-type: none"> <code>on</code> определяет, по какому столбцу или столбцам выполнять присоединение. <code>how</code> управляет типом присоединения, по умолчанию '<code>inner</code>'. <code>left_on</code> и <code>right_on</code> используются, когда имена столбцов для присоединения различаются в двух таблицах. Параметру <code>left_on</code> передают столбец из левой таблицы, а параметру <code>right_on</code> — из правой.
<code>join()</code>	Присоединяет данные с помощью индексов	<code>таблица1.join(таблица2, how='тип_присоединения')</code>	<ul style="list-style-type: none"> <code>how</code> управляет типом присоединения, по умолчанию '<code>inner</code>'. <code>lsuffix</code> и <code>rsuffix</code> — суффиксы, которые помогают избавиться от конфликтов в именах столбцов. Например, если в двух таблицах встречаются столбцы с одинаковыми названиями, то к каждому из них добавляется соответствующий суффикс. <code>on</code> — необязательный параметр, в отличие от <code>merge()</code>. Он используется, когда индексы соединяемых датафреймов не совпадают.
<code>concat()</code>	Аналогичен оператору <code>UNION ALL</code> в SQL, конкатенирует два датафрейма и более, соединяя их по строкам или столбцам	<code>pd.concat([таблица1, таблица2, ...])</code>	<code>objs</code> — последовательность датафреймов, которые нужно объединить, чаще всего список.

Описательная статистика

Генеральная совокупность — это полное множество объектов, относительно которых предполагается делать выводы при исследовании.

Выборка — это подмножество генеральной совокупности. Используется для сбора данных и подготовки выводов о генеральной совокупности.

Степень свободы — понятие, которое используют, чтобы оценить количество независимых данных, доступных для анализа. В случае выборки степень свободы рассчитывается так: размер выборки n минус один, то есть $n - 1$.

Название показателя	Описание	Метод в pandas
Меры центральной тенденции		
Среднее арифметическое	Сумма всех значений в наборе данных, делённая на количество значений	<code>df['column_name'].mean()</code>
Медиана	Значение, которое делит распределение на две равные части так, что половина значений находится выше медианы, а половина ниже	<code>df['column_name'].median()</code>
Мода	Значение, которое встречается в наборе данных наиболее часто	<code>df['column_name'].mode()</code>
Меры разброса		
Размах	Разница между максимальным и минимальным значениями выборки	<code>df['column_name'].max() - df['column_name'].min()</code>
Дисперсия	Среднее квадратичное отклонение значений выборки от среднего	<code>df['column_name'].var()</code>
Стандартное отклонение	Средняя величина изменчивости в наборе данных, корень из дисперсии	<code>df['column_name'].std()</code>

Шпаргалка

3/10

Визуализации в Python

Тип визуализации	Описание	Варианты кода	Настройки
Гистограмма	Визуализирует распределение числовых данных.	<p>Первый пример гистограммы в matplotlib:</p> <pre>plt.figure(figsize=(10, 6)) plt.hist(column='column_name', bins=8, alpha=1, color='salmon', edgecolor='black') plt.title('Название графика') plt.xlabel('Подпись оси X') plt.ylabel('Подпись оси Y') plt.savefig('название_картинки', dpi=300)</pre> <p>Второй пример гистограммы в matplotlib:</p> <pre>ax=df.hist(column='column_name', figsize=(12, 10)) ax.set_title('Название графика') ax.set_xlabel('Подпись оси X') ax.set_ylabel('Подпись оси Y')</pre> <p>Гистограмма в seaborn:</p> <pre>sns.histplot(data=df, x='score', bins=15, kde=True, hue='Churn') plt.title('Название графика') plt.ylabel('Подпись оси Y') plt.show()</pre>	<p>Настройки в matplotlib:</p> <ul style="list-style-type: none"> • <code>column</code> — столбец, на данных которого строится гистограмма; • <code>bins</code> — количество корзин в гистограмме; • <code>alpha</code> — устанавливает прозрачность столбцов на графике: <code>1</code> — полностью непрозрачные, <code>0</code> — полностью прозрачные, а <code>0.5</code> — полупрозрачные; • <code>color</code> — задаёт цвет столбцов; • <code>edgecolor</code> — применяется для создания обводки столбцов; • <code>figsize</code> — устанавливает размер графика; • <code>title</code> — задаёт название графика; • <code>xlabel</code> — задаёт название оси X; • <code>ylabel</code> задаёт название оси Y; • <code>savefig</code> — сохраняет картинку в определённом качестве, чем выше <code>dpi</code>, тем лучше качество. <p>Настройки в seaborn:</p> <ul style="list-style-type: none"> • <code>bins</code> — параметр, который отвечает за количество групп в распределении и принимает только целые числа. • <code>kde</code> — с помощью этого параметра можно построить линию KDE (с англ. «ядерная оценка плотности»). Его значения могут быть как <code>True</code>, так и <code>False</code> (по умолчанию используется <code>False</code>). • <code>hue</code> — этот параметр принимает столбец, по которому произойдёт группировка данных.

Шпаргалка

<p>«Ящик с усами», или диаграмма размаха</p>	<p>Позволяет быстро оценить симметричность распределения, его размах и наличие выбросов.</p>	<p>Первый пример:</p> <pre>df.boxplot(column='column_name', vert=False, grid=True figsize=(12, 10))</pre> <p>Второй пример:</p> <pre>plt.figure(figsize=(10, 6)) plt.boxplot('column_name', vert=False, flierprops=dict(markerfacecolor='red', marker='o')) plt.title('Название графика') plt.xlabel('Подпись оси X') plt.grid(True)</pre>	<ul style="list-style-type: none"> • <code>vert</code> — принимает значение <code>True</code> или <code>False</code>, которое указывает на ориентацию диаграммы. При <code>True</code> (значение по умолчанию) диаграмма рисуется вертикально. При <code>False</code> — горизонтально. • <code>grid</code> — принимает значение <code>True</code> или <code>False</code>, которое соответственно включает или отключает отображение сетки на графике. Можно также задать значения <code>axis='y'</code> или <code>axis='x'</code>, тогда построятся только вертикальные или горизонтальные линии соответственно указанной оси. • Параметр <code>flierprops</code> принимает словарь с настройками, которые задают стиль точек с выбросами. • <code>markerfacecolor='red'</code> устанавливает красный цвет заливки точек (<code>'red'</code>). Параметр <code>markerfacecolor</code> может принимать и другие цвета (<code>'green'</code>, <code>'blue'</code>, <code>'yellow'</code> и так далее). • <code>marker='o'</code> устанавливает форму маркера для точек как круг (<code>'o'</code>). Может принимать и другие значения: <code>'s'</code> — квадрат; <code>'^'</code> — треугольник, направленный вверх; <code>'v'</code> — треугольник, направленный вниз; <code>'D'</code> — ромб; <code>'x'</code> — крест.
<p>Диаграмма рассеяния</p>	<p>Помогает наглядно увидеть зависимости между переменными. На что стоит обращать внимание при анализе диаграммы рассеяния:</p> <ul style="list-style-type: none"> • Тип связи: линейная или нелинейная. • Направление: положительное или отрицательное. • Сила связи. Чем ближе точки располагаются к линии или кривой, тем сильнее связь. 	<pre>df.plot(kind='scatter', x='название_столбца_X', y='название_столбца_Y', alpha=0.7, color='blue', edgecolor='black', s=50, figsize=(10, 6), title='Заголовок графика')</pre>	<ul style="list-style-type: none"> • <code>x</code> — название столбца, значения которого будут отображены по оси X. Этот параметр обязателен для диаграммы, ведь он определяет, какую переменную использовать для горизонтальной оси. • <code>y</code> — название столбца, значения которого будут отображены по оси Y. Также обязательный параметр, который задаёт вертикальную ось графика. • <code>color</code> — задаёт цвет точек на графике. • <code>edgecolor</code> — определяет цвет контура точек. • <code>s</code> — размер точек на графике. Значение <code>s</code> — это число, которое определяет размер каждой точки, например <code>s=50</code> или <code>s=80</code>. Чем выше значение, тем крупнее точки на графике. Можно использовать этот параметр, если на графике много точек или их необходимо выделить.

Шпаргалка

Столбчатая диаграмма	<p>Предназначена для сравнения отдельных категорий или групп.</p>	<p>Первый пример:</p> <pre>df.plot() kind='bar', title='Заголовок графика', legend=False, ylabel='название_столбца_Y', xlabel='название_столбца_X', rot=0, color='skyblue', figsize=(10, 6)) plt.show()</pre> <p>Второй пример:</p> <pre>df.plot.bar(title='Заголовок графика', legend=False, x='название_столбца_X', y='название_столбца_Y', rot=0, color='skyblue', figsize=(10, 6)) plt.show()</pre> <p>Третий пример:</p> <pre>df.plot(kind='bar', color='skyblue', legend=False) plt.title('Заголовок графика') plt.ylabel('название_столбца_Y') plt.xlabel('название_столбца_X') plt.xticks(rotation=0), figsize=(10, 6)) plt.show()</pre>	<ul style="list-style-type: none"> • <code>legend</code> — значение, которое указывает на то, отображать легенду или нет (<code>True</code> или <code>False</code> соответственно). По умолчанию выставлен в <code>True</code>, то есть отображать легенду. • <code>rot</code> — угол поворота меток на осях X или Y, в зависимости от типа графика. В столбчатой диаграмме <code>bar</code> будут повёрнуты метки на оси X, а в горизонтальной столбчатой диаграмме <code>barh</code> — метки на оси Y.
----------------------	---	---	--

Шпаргалка

Линейный график	<p>Помогает увидеть тенденции и определить, где происходит увеличение или снижение значений, что особенно важно для анализа временных рядов.</p>	<p>Простой линейный график:</p> <pre>df.plot.line() x='название_столбца_X', y='название_столбца_Y', title='Заголовок графика')</pre> <p>Линейный график с двумя переменными:</p> <pre>df_agg = df.groupby('user_check_in_month').agg({'client id': 'count', 'price for the night': 'mean'}) df_agg.plot.line() plt.title('Линейный график количества посещений и средней цены номера за ночь') plt.xlabel('Месяц') plt.ylabel('Величина') plt.legend(loc='upper right') plt.show()</pre> <p>Подграфики:</p> <pre>df_agg.plot(kind='line', subplots=True, sharex=True, sharey=False, legend=False, title=['Количество гостей по месяцам', 'Средний чек за ночь']) plt.xlabel('Месяц') plt.show()</pre>	<ul style="list-style-type: none"> • <code>legend</code> — значение, которое указывает на то, отображать легенду или нет (<code>True</code> или <code>False</code> соответственно). По умолчанию выставлен в <code>True</code>, то есть отображать легенду. С помощью <code>loc='upper right'</code> можно указать местоположение легенды, например в правом верхнем углу. • <code>subplots</code> — позволяет создать несколько подграфиков в одном окне, принимает значение <code>True</code> или <code>False</code>. Если выставить <code>True</code>, тогда каждый столбец датафрейма станет отдельным подграфиком. • <code>sharex</code> — принимает значение <code>True</code> или <code>False</code>. Если выставить <code>True</code> — будет использоваться общая ось X для всех подграфиков. • <code>sharey</code> — принимает значение <code>True</code> или <code>False</code>. Если выставить <code>True</code> — будет использоваться общая ось Y для всех подграфиков.
-----------------	--	---	---

Шпаргалка

Тепловая карта	<p>Демонстрирует плотность распределения точек или значений.</p> <pre><code>plt.figure(figsize=(8, 6)) sns.heatmap(data=corr_matrix, annot=True, fmt='.2f', linewidths=0.5, cmap='viridis') plt.title('Тепловая карта матрицы корреляций') plt.show()</code></pre>	<ul style="list-style-type: none"> • <code>data</code> — основной параметр, принимает набор данных, который будет визуализирован как тепловая карта. Если передали датафрейм, его индексы и названия столбцов будут использоваться для подписей строк и столбцов. В примере кода это переменная <code>corr_matrix</code>, которая содержит корреляции между числовыми столбцами датафрейма. • <code>annot</code> — необязательный параметр, по умолчанию <code>None</code>. Если установлено <code>True</code>, то значения ячеек будут отображены на тепловой карте поверх цветовых градиентов. • <code>fmt</code> — необязательный параметр, по умолчанию <code>'.2g'</code>. Отвечает за формат строк для вывода чисел в случае, когда параметр <code>annot=True</code>. При <code>'.2f'</code> значения будут округляться до двух знаков после запятой (например: 0.85 вместо 0.84923). • <code>vmin</code> и <code>vmax</code> — определяют минимальные и максимальные значения для цветовой шкалы тепловой карты. Параметры позволяют управлять диапазоном значений, которые будут отображаться на карте. • <code>linewidths</code> — необязательный параметр, по умолчанию <code>0</code>. Отвечает за толщину линий, которые разделяют ячейки тепловой карты. В примере <code>linewidths=0.5</code> добавляет тонкие линии между ячейками для более чёткого восприятия. • <code>cmap</code> — необязательный параметр, по умолчанию <code>None</code>. Определяет цветовую схему тепловой карты. В коде выбрана <code>cmap='viridis'</code> — популярная цветовая схема, которая хорошо различается по контрасту. Наиболее часто применяются такие цветовые карты: <code>'viridis'</code>, <code>'coolwarm'</code>, <code>'Blues'</code> и <code>'rocket'</code>.
----------------	--	---

Шпаргалка

График совместного распределения	<p>Объединяет в себе два компонента:</p> <ul style="list-style-type: none"> диаграмма рассеяния; гистограммы для каждой переменной по осям <code>x</code> и <code>y</code>, что даёт возможность оценить их распределение. 	<pre><code>sns.jointplot(data=df, x='score', y='Balance', kind='hex', height=6) plt.suptitle('Совместное распределение данных в столбцах score и Balance') plt.tight_layout() plt.show()</code></pre>	<ul style="list-style-type: none"> • <code>data</code> — датафрейм, на основе которого будет строиться распределение. • <code>x</code> и <code>y</code> — названия столбцов данных, отображаемых на осях X и Y соответственно. • <code>plt.suptitle()</code> — располагает общий заголовок для графика по центру. • <code>plt.tight_layout()</code> — помогает настраивать отступы между элементами графика. • <code>kind</code> — указывает тип графика, который будет использован для отображения связей между двумя переменными. Наиболее популярные <code>'scatter'</code>, <code>'kde'</code>, <code>'hex'</code> и <code>'hist'</code>. По умолчанию используется <code>'scatter'</code>. • <code>height</code> — определяет размер графика. Указывается одно число, ведь график отрисовывается в квадрате.
Сводная таблица	<p>Используется при работе с данными, в которых столбцы и строки представляют параметры или категории, а ячейки содержат агрегированные показатели: сумму, среднее, максимум и минимум — по соответствующим группам.</p>	<pre><code>pivot_sales = pd.pivot_table(df, index='Категория товара', columns='Регион', values='Количество продаж', aggfunc='sum')</code></pre>	<ul style="list-style-type: none"> • <code>df</code> — данные. • <code>index</code> — названия столбцов исходных данных, по которым нужно сгруппировать информацию. В итоговой сводной таблице эти столбцы становятся строками, а данные группируются по заданным категориям. Без <code>index</code> данные будут агрегированы в общую сумму по столбцам. • <code>columns</code> — задаёт, как распределять данные по столбцам. Если не указывать здесь данные, то результат будет похож на <code>groupby()</code>, так как данные не будут распределяться по столбцам. • <code>values</code> — определяет, какие данные нужно агрегировать, чаще всего это числовые значения. Без этого параметра pandas попытается агрегировать все числовые столбцы. • <code>aggfunc</code> — определяет функцию агрегации: сумма <code>'sum'</code>, количество <code>'count'</code>, среднее <code>'mean'</code> или другая агрегирующая функция. Без указания <code>aggfunc</code> применяется метод <code>mean()</code>.

Шпаргалка

Коэффициенты корреляции

Название	Описание	Ограничения	Интерпретация	Синтаксис
Коэффициент корреляции Пирсона (r)	<p>Статистическая мера, которая оценивает силу и направление линейной зависимости между двумя количественными переменными. Показывает, насколько изменение одной переменной связано с изменением другой.</p>	<p>В каких случаях применяется:</p> <ul style="list-style-type: none"> Линейность. <p>Коэффициент Пирсона улавливает только линейные зависимости. Если связь между переменными нелинейная, коэффициент может быть близок к нулю, даже если переменные связаны.</p> <ul style="list-style-type: none"> Отсутствие выбросов. Выбросы могут сильно повлиять на значение коэффициента. Если какая-то точка данных будет находиться слишком далеко от других, это сильно исказит значение коэффициента, что может привести к неверным выводам. 	<p>• $r = 1$ — идеальная положительная линейная связь: если одна переменная увеличивается, другая увеличивается с той же скоростью.</p> <p>• $r = -1$ — идеальная отрицательная линейная связь: если одна переменная увеличивается, другая уменьшается с той же скоростью.</p> <p>• $r = 0$ — линейной связи между переменными нет.</p>	<p>Метод <code>corr()</code> позволяет быстро и удобно вычислять коэффициенты корреляции для числовых данных в датафреймах. Метод позволяет использовать разные коэффициенты корреляции, включая Пирсона и Спирмена.</p> <pre><code>df.corr(method='pearson', min_periods=1)</code></pre> <p>В коде метод вызван со значениями параметров по умолчанию.</p> <p>Параметры:</p> <ul style="list-style-type: none"> <code>method</code>: <ul style="list-style-type: none"> <code>'pearson'</code> — считает коэффициент Пирсона, <code>'spearman'</code> — считает коэффициент Спирмена. <code>min_periods</code> — этот параметр контролирует минимальное количество непустых значений, которые должны присутствовать в каждой паре данных, чтобы вычислить корреляцию между двумя переменными.
Коэффициент корреляции Спирмена (ρ)	<p>Статистическая мера, которая оценивает силу и направление монотонной зависимости между двумя переменными. Коэффициент Спирмена применяется, когда данные являются порядковыми, или ранговыми, или когда между переменными не наблюдается линейной связи, но есть тенденция к изменению в одном направлении.</p>	<p>В каких случаях применяется:</p> <ul style="list-style-type: none"> Монотонность, но не линейность. Спирмен определяет наличие монотонной зависимости, но не гарантирует, что связь между переменными линейна. Устойчивость к выбросам: Коэффициент Спирмена менее чувствителен к выбросам в данных по сравнению с коэффициентом Пирсона, ведь он оперирует ранговыми значениями. 	<ul style="list-style-type: none"> $\rho = 1$ — идеальная монотонная положительная зависимость: все ранги совпадают. $\rho = -1$ — идеальная монотонная отрицательная зависимость: все ранги обратны друг другу. $\rho = 0$ — монотонной зависимости между переменными нет. 	

Шпаргалка

Критерий хи-квадрат (χ^2)	С помощью критерия хи-квадрат можно проверить, отличается ли распределение предпочтений в зависимости от категории. Если значение критерия будет высоким, это укажет на наличие связи.	Для категориальных переменных.	
Коэффициент V Крамера	Коэффициент V Крамера поможет оценить, насколько сильно связаны две категориальные переменные. Пример: вы исследуете связь между уровнем образования (среднее, высшее, ученая степень) и предпочтением разных видов спорта (футбол, баскетбол, теннис). Чтобы оценить корреляцию, создаётся таблица размером 3x3x3.	Для категориальных переменных.	
Коэффициент сопряжённости и Пирсона (ϕ)	Используется для оценки взаимосвязи между двумя двоичными переменными. Создаётся таблица 2x2.	Для категориальных переменных.	
Коэффициент корреляции phi_k	Измеряет степень отличия связи между переменными от случая независимости. Анализирует, насколько сильно две переменные связаны, независимо от их природы: числовые, категориальные и смешанные.	Используя метод <code>phi_k</code> , можно определить как линейные, так и нелинейные зависимости.	<pre>from phik import phik_matrix correlation_matrix = df.phik_matrix()</pre>