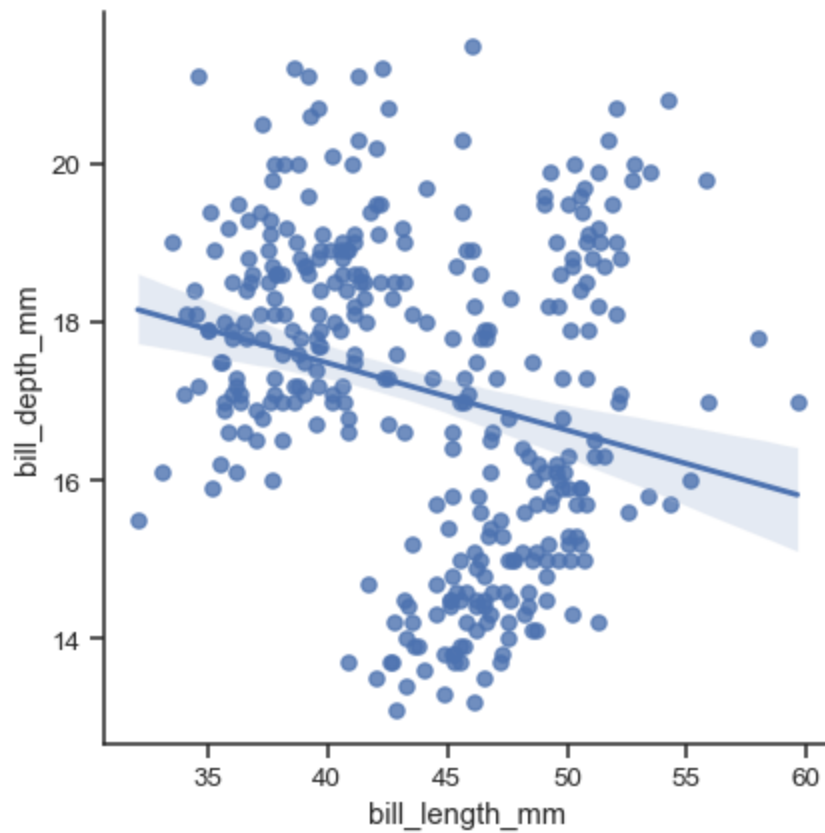


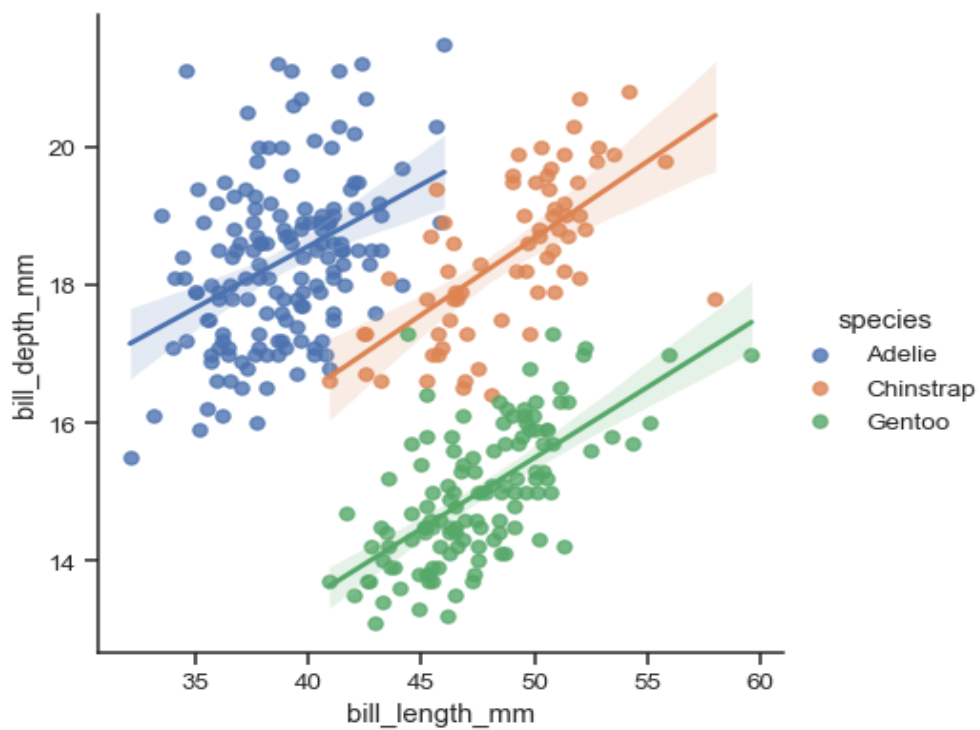
Seaborn : Lmplots

Параметр	Описание
data : DataFrame	Таблица с данными
x, y : strings	Входные переменные; это должны быть имена столбцов в данных
hue : string	Разбиение по цветам уникальных значений или категорий
col : string	Дополнительное разбиение по какой-то колонке + разбивает на subplots
row : string	Также разбиение + разбивает на subplots, снизу будут примеры
palette : list or dict	Цвета, используемые для различных уровней переменной hue. Должно быть что-то, что можно интерпретировать с помощью color_palette() или словарь, отображающий уровни оттенков в цветах matplotlib.

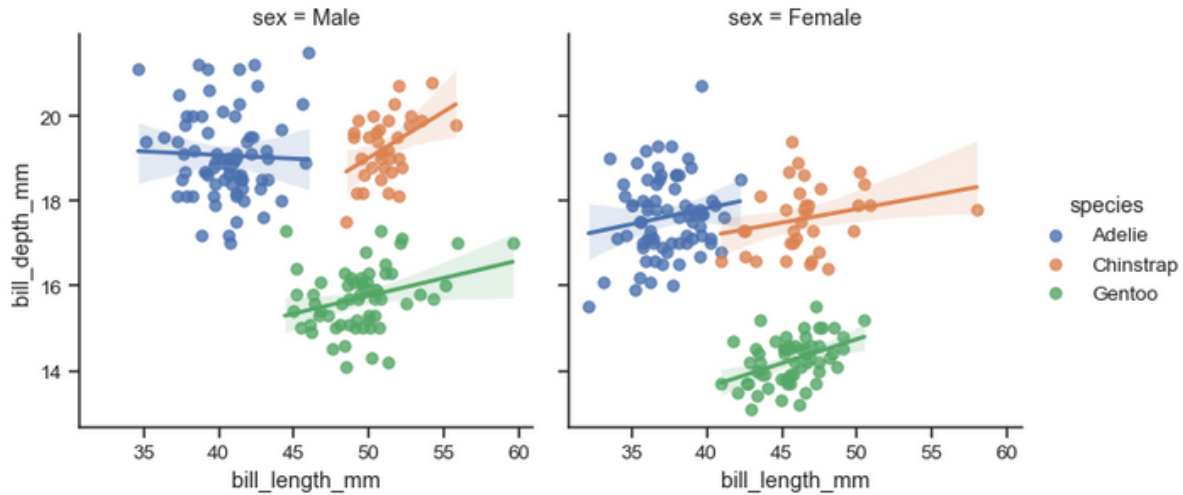
```
sns.lmplot(data=penguins, x="bill_length_mm", y="bill_depth_mm")
```



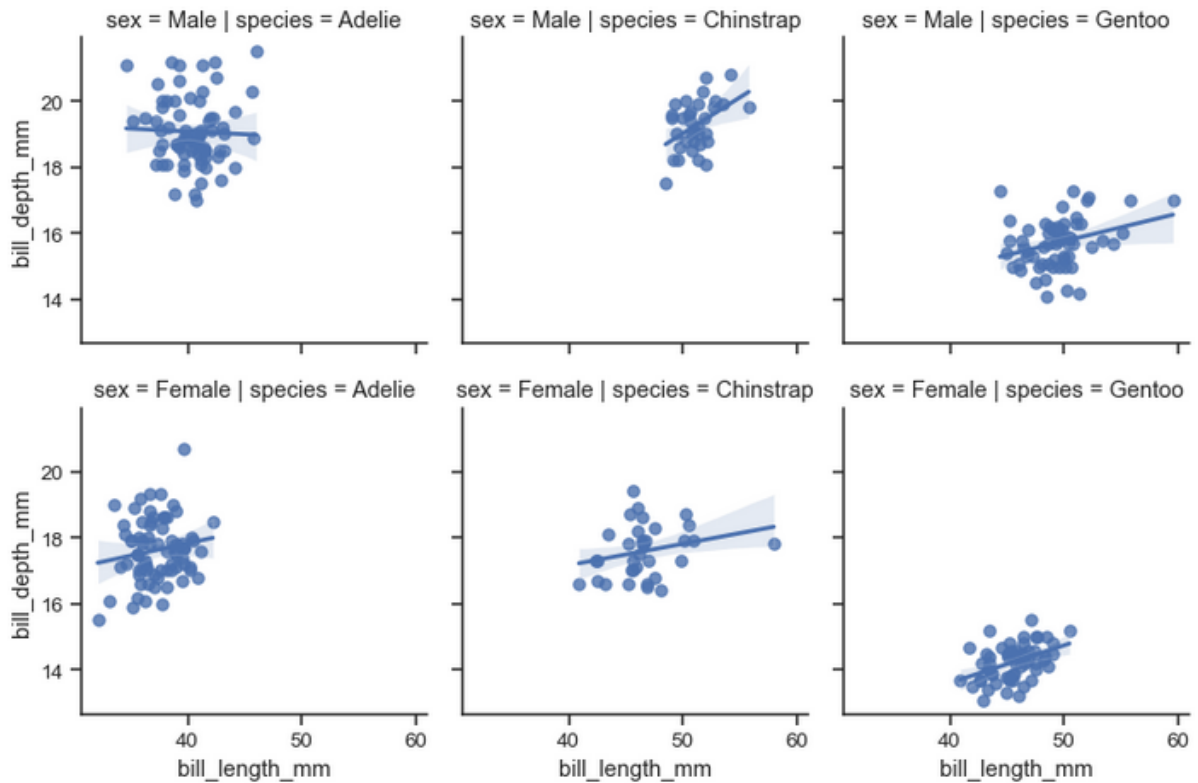
```
sns.lmplot(data=penguins, x="bill_length_mm", y="bill_depth_mm", hue="species")
```



```
sns.lmplot(
    data=penguins, x="bill_length_mm", y="bill_depth_mm",
    hue="species", col="sex", height=4,
)
```



```
sns.lmplot(
    data=penguins, x="bill_length_mm", y="bill_depth_mm",
    col="species", row="sex", height=3,
)
```

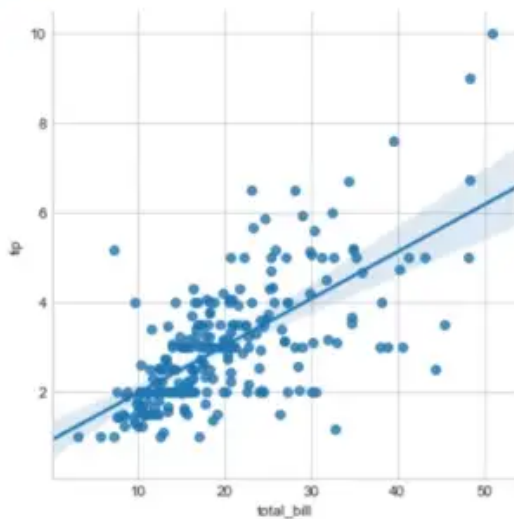


Пример

Каждый график в Seaborn имеет набор фиксированных параметров. Для `sns.lmplot()` у нас есть три обязательных параметра, а остальные являются необязательными, которые мы можем использовать в соответствии с нашими требованиями. Эти 3 параметра представляют собой значения для оси X, значения для оси Y и ссылку на набор данных. Эти 3 в основном видны почти на всех графиках Seaborn, и, кроме того, есть необязательный параметр, который я хотел бы, чтобы вы запомнили, так как он очень удобен. Это параметр оттенка, он принимает категориальные столбцы и помогает нам сгруппировать наш график данных в соответствии со значениями параметра оттенка. Давайте посмотрим, как это работает:

```
# Basic lmplot visualization:  
sns.lmplot(x="total_bill", y="tip", data=tips)
```

```
<seaborn.axisgrid.FacetGrid at 0x22f2e961a58>
```



Давайте теперь разберемся, что мы видим на экране, прежде чем мы перейдем к добавлению параметров. Эта линейная линия на нашем графике лучше всего соответствует тенденции чаевых, которые обычно дают клиенты, относительно общего счета, который они получают. А точки данных, которые мы видим в крайнем правом верхнем углу и которые находятся далеко от этой

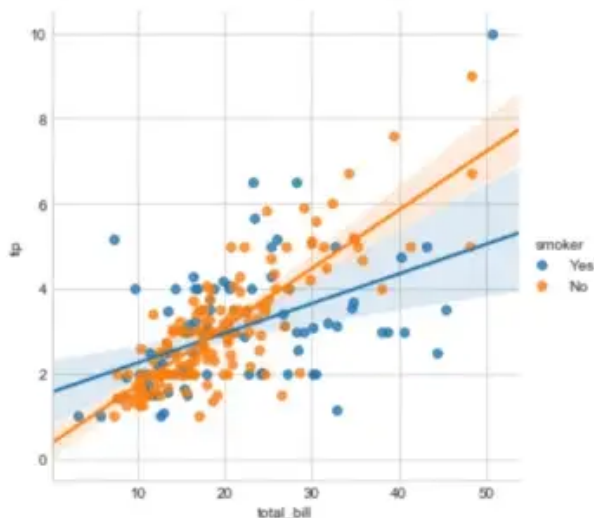
линии, называются выбросами в наборе данных. Мы можем думать о выбросах как об исключениях.

Цель науки о данных состоит в том, чтобы предсказать наилучшее соответствие для понимания тенденций в поведении посетителей, и наш алгоритм всегда должен разрабатываться соответствующим образом. Мы можем найти это распространенным сценарием при применении обобщенных линейных моделей в машинном обучении. Если мы очень внимательно заметим, эта тень сходится в центре, где находится часть наших данных. Эта точка схождения на самом деле является средним статистическим значением или, проще говоря, обобщенным прогнозом стоимости чаевых в этом ресторане на ежедневной основе.

В этом случае, глядя на этот график, мы можем сказать, что если общий счет составляет около 20 долларов, то он получит чаевые в размере около 3 долларов. Давайте еще больше улучшим эту визуализацию, добавив в сюжет больше функций, и для этой цели попробуем понять, дает ли Курильщик в целом чаевые больше или меньше:

```
sns.lmplot(x="total_bill", y="tip", hue="smoker", data=tips)
```

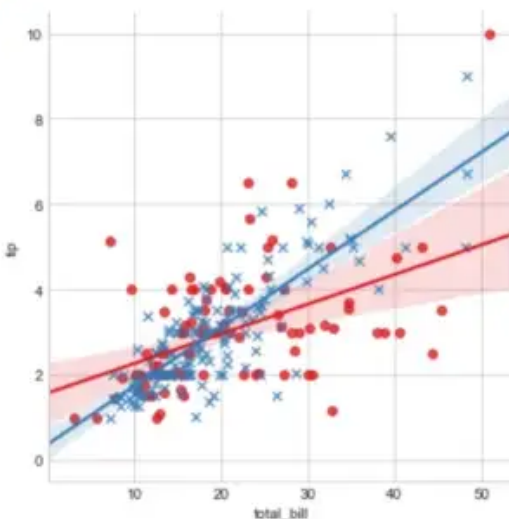
<seaborn.axisgrid.FacetGrid at 0x22f2e838da0>



Отражает, что курильщики, которых мы видим в синем цвете, немного более щедры, но не так соответствуют их привычке давать чаевые, поскольку точки данных довольно расплывчаты. Итак, добавление 3-го параметра `hue` помогло нам визуализировать эту разницу в отдельной цветовой диаграмме, а также добавило легенду с Yes, No для удобства интерпретации. Давайте рассмотрим другие часто используемые параметры для дальнейшей настройки этого графика:

```
sns.lmplot(x="total_bill", y="tip", hue="smoker", data=tips, markers=["o", "x"], palette="Set1", legend=False)
```

<seaborn.axisgrid.FacetGrid at 0x22f2ea64240>



Здесь мы установили стиль маркера точки данных, изменили цвет и решили удалить легенду, которая по умолчанию присутствует всегда. Прямо сейчас, будь то для курильщика или для некурящего, представление находится на одном и том же сюжете, поэтому давайте разберем его на отдельных гранях:

```
sns.lmplot(x="total_bill", y="tip", col="smoker", data=tips)
```

```
<seaborn.axisgrid.FacetGrid at 0x22f2f2560f0>
```

