# Иллюстрированное руководство по модели регрессии Пуассона

An Illustrated Guide to the Poisson Regression Model

<https://towardsdatascience.com/an-illustrated-guide-to-the-poisson-regression-model-50cccba15958>

[Sachin Date](https://sachin-date.medium.com/?source=post_page-----50cccba15958--------------------------------)

[22 сент.2019 г.](https://towardsdatascience.com/an-illustrated-guide-to-the-poisson-regression-model-50cccba15958?source=post_page-----50cccba15958--------------------------------)

[Несколько советов перед началом ... 2](#_Toc77861729)

[Что такое счетные данные? 2](#_Toc77861730)

[Набор данных из реального мира 3](#_Toc77861731)

[Модели регрессии для счетных данных 4](#_Toc77861732)

[Стратегия регрессии 4](#_Toc77861733)

[Введение в распределение Пуассона 4](#_Toc77861734)

[Модель регрессии Пуассона при постоянном λ 5](#_Toc77861735)

[Модель регрессии Пуассона при изменяющемся λ 6](#_Toc77861736)

[Формальное описание (спецификация) модели регрессии Пуассона 9](#_Toc77861737)

[Обучение модели регрессии Пуассона 11](#_Toc77861738)

[Понимание оценки максимального правдоподобия (MLE) 11](#_Toc77861739)

[Все шаги выполнения регрессии Пуассона 14](#_Toc77861740)

[Создание регрессии Пуассона в Python 15](#_Toc77861741)

[Степень соответствия модели регрессии Пуассона 21](#_Toc77861742)

[Заключение и следующие шаги 22](#_Toc77861743)

Рассмотрим в статье следующие темы:

1. Особенности счетного наборов данных (count based dataset). Счетные наборы – одни из наиболее часто встречающихся в мире данных. Посмотрим, в чем отличие подобных наборов.
2. Модели регрессии для прогнозирования количества событий (зависимой переменной в счетном наборе). Подробно рассмотрим модель регрессии Пуассона. Еще одна популярная модель для работы со счетными наборами – это модель отрицательной биномиальной регрессии (NB), о которой расскажем в следующей статье.
3. Py thon – руководство по регрессии Пуассона. Рассмотрим пошаговое руководство о разработке модели регрессии Пуассона в Py thon с использованием GLM класса статистических моделей и ее обучении на реальном наборе данных.

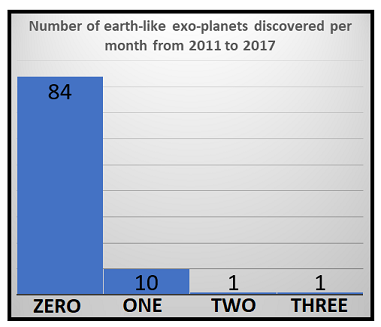
# Несколько советов перед началом ...

* Python - руководство представлено в двух последних разделах статьи.
* Код в Github [здесь](https://gist.github.com/sachinsdate/d5535d5489178e6271f4c0c7a444da1e) .
* Используемый в статье набор данных о подсчете велосипедистов [здесь](https://gist.github.com/sachinsdate/c17931a3f000492c1c42cf78bf4ce9fe) .
* Базовые знания по случайным величинам, процессу Пуассона и программе Python для моделирования процесса Пуассона представлены в статье [The Poisson Process: Everything you need to know](https://towardsdatascience.com/the-poisson-process-everything-you-need-to-know-322aa0ab9e9a) (Пуассоновский [процесс: все, что вам нужно знать](https://towardsdatascience.com/the-poisson-process-everything-you-need-to-know-322aa0ab9e9a))

Приступаем!

# Что такое счетные данные?

Счетные данные содержат события, которые происходят с определенной частотой. Частота события может меняться со временем или от одного наблюдения к другому. Несколько примеров:



Источник данных: Википедия, [потенциально обитаемые экзопланеты](https://en.wikipedia.org/wiki/List_of_potentially_habitable_exoplanets) (изображение [автора](https://sachin-date.medium.com/))

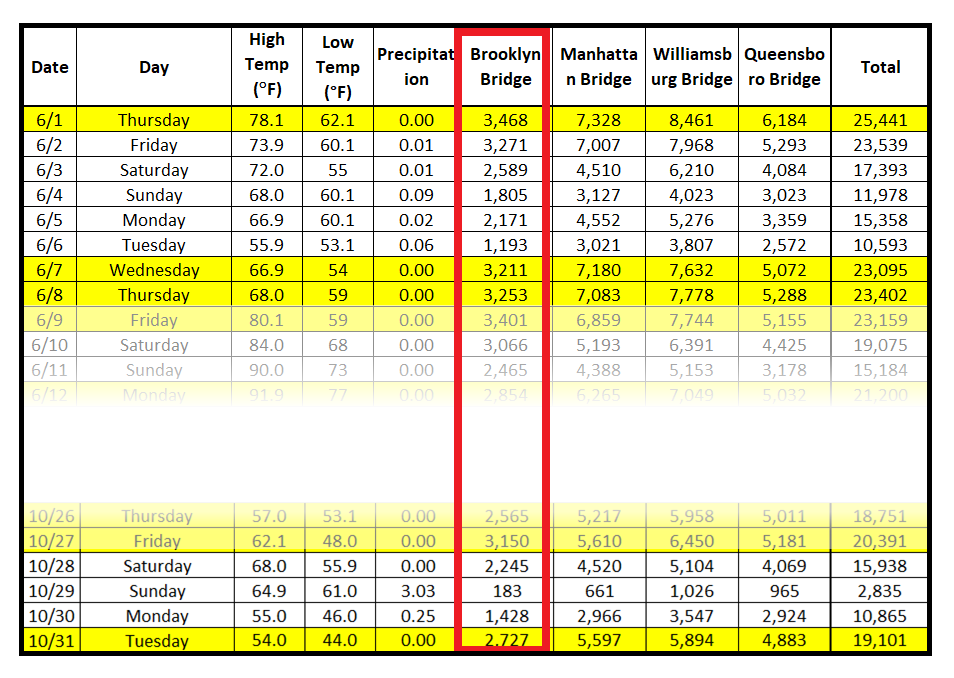
* количество транспортных средств, пересекающих перекресток в час;
* количество человек, посещающих врача в месяц;
* количество обнаруженных за месяц экзопланет.

Набор счетных данных характеризуется следующим:

* Целочисленность, неотрицательность данных. Данные состоят из неотрицательных целых чисел: [0… ∞]. Такие методы регрессии, как регрессия посредством наименьших квадратов (Ordinary Least Squares Regression, OLSR), могут не подходить для моделирования счетных данных, поскольку OLSR лучше всего работает с действительными числами, такими как -656.0, -0.00000345, 13786.1 и т.п.
* Смещенное распределение. Данные могут содержать большое количество точек всего для нескольких значений, что приводит к смещению частотного распределения. См., например, вышеприведенную гистограмму для обнаружения экзопланет.
* Немногочисленность, разреженность. Данные могут отражать редкие события, как например, вспышки гамма-излучения.
* Частота возникновения. Для моделирования можно предположить, что существует заданная частота возникновения событий λ, определяющая генерацию данных. Частота событий может изменяться со временем.

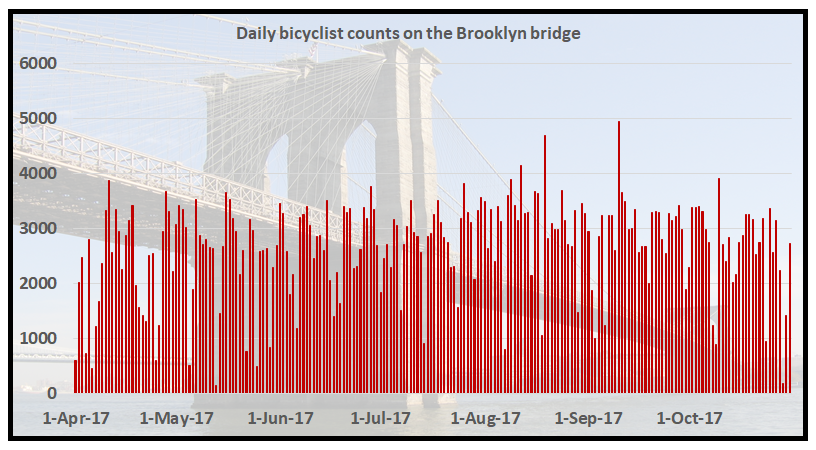
# Набор данных из реального мира

В следующей таблице представлено количество велосипедистов, проезжающих по различным мостам Нью-Йорка. Учет проводился ежедневно с 1 апреля по 31 октября 2017 года.



Источник: Количество [велосипедистов на мостах Ист-Ривер](https://data.cityofnewyork.us/Transportation/Bicycle-Counts-for-East-River-Bridges/gua4-p9wg) (Источник данных: NY C OpenData) (Изображение [автора](https://sachin-date.medium.com/))

Временной график появления велосипедистов на Бруклинском мосту.



Фон – [вид на Бруклинский мост с острова Манхэттен](https://en.wikipedia.org/wiki/Brooklyn_Bridge#/media/File:Brooklyn_Bridge_Postdlf.jpg)

# Модели регрессии для счетных данных

Две наиболее популярных модели регрессии для счетных данных – это модель регрессии Пуассона и модель отрицательной биномиальной (Negative Binomial) регрессии. Другие варианты – это модели с [упорядоченным логитом](https://en.wikipedia.org/wiki/Ordered_logit) (ordered logit), [упорядоченным пробитом](https://en.wikipedia.org/wiki/Ordered_probit) (ordered probit) и [нелинейными методами наименьших квадратов](https://en.wikipedia.org/wiki/Non-linear_least_squares) (nonlinear least squares).

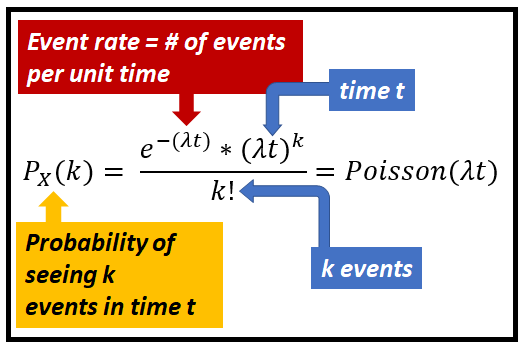
# Стратегия регрессии

Хорошей практикой является начало работы с модели регрессии Пуассона и применение ее в качестве «точки отсчета» для более сложных моделей или моделей с менее жесткими ограничениями. В своей книге « [Регрессионный анализ счетных данных»](http://faculty.econ.ucdavis.edu/faculty/cameron/racd2/) (Regression Analy sis of Count Data) Кэмерон и Триведи говорят: «Хорошая практика - оценивать как пуассоновские, так и отрицательные биномиальные модели».

В статье будем использовать модель регрессии Пуассона для оценки количества велосипедистов на Бруклинском мосту.

# Введение в распределение Пуассона

Распределение Пуассона имеет следующую функцию вероятности (Probability Mass Function, PMF).



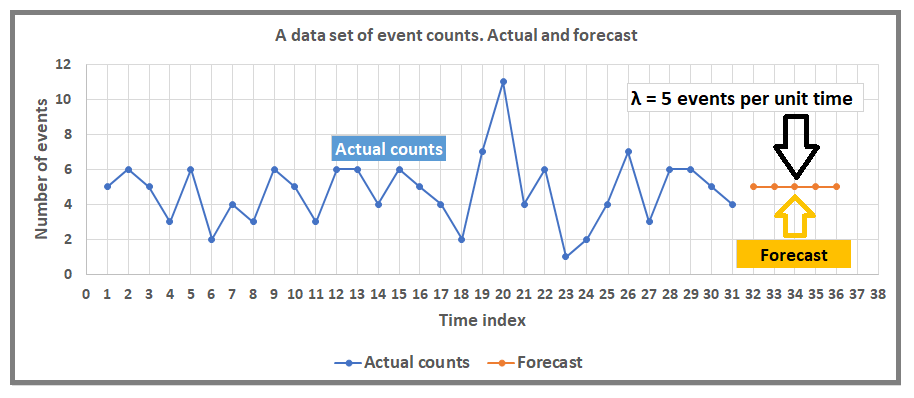
Вероятность обнаружения *k* событий за время t, учитывая λ событий, происходящих в единицу времени (Изображение [автора](https://sachin-date.medium.com/))

Ожидаемое (среднее) значение для распределения Пуассона равно λ. Это означает, что при отсутствии другой информации ожидается появление λ событий в единичном временном интервале, например, 1 час, 1 день и т. д. Для любого интервала t предполагается появление λt событий.

# Модель регрессии Пуассона при постоянном λ

Если частота событий λ постоянна, можно просто использовать модель среднего значения для прогнозирования будущего количества событий. В этом случае можно установить все предсказания просто равными λ.

На следующем рисунке показано решение на основе постоянного λ.



Фактические и прогнозируемые подсчеты для модели с постоянной скоростью (изображение [автора](https://sachin-date.medium.com/))

Представленный ниже код Python использовался для генерации синих точек (фактическое количество событий в прошлом) с использованием процесса Пуассона с λ = 5. Оранжевые точки (прогноз) имеют одинаковое значение 5.

import random

import math

\_lambda = 5

\_num\_total\_arrivals = 150

\_num\_arrivals = 0

\_arrival\_time = 0

\_num\_arrivals\_in\_unit\_time = []

\_time\_tick = 1

print('RANDOM\_N,INTER\_ARRIVAL\_TIME,EVENT\_ARRIVAL\_TIME')

for i in range(\_num\_total\_arrivals):

#Get the next probability value from Uniform(0,1)

p = random.random()

#Plug it into the inverse of the CDF of Exponential(\_lamnbda)

\_inter\_arrival\_time = -math.log(1.0 - p)/\_lambda

#Add the inter-arrival time to the running sum

\_arrival\_time = \_arrival\_time + \_inter\_arrival\_time

#Increment the number of arrival per unit time

\_num\_arrivals = \_num\_arrivals + 1

if \_arrival\_time > \_time\_tick:

\_num\_arrivals\_in\_unit\_time.append(\_num\_arrivals)

\_num\_arrivals = 0

\_time\_tick = \_time\_tick + 1

#print it all out

print(str(p)+','+str(\_inter\_arrival\_time)+','+str(\_arrival\_time))

print('\nNumber of arrivals in successive unit length intervals ===>')

print(\_num\_arrivals\_in\_unit\_time)

print('Mean arrival rate for sample:' + str(sum(\_num\_arrivals\_in\_unit\_time)/len(\_num\_arrivals\_in\_unit\_time)))

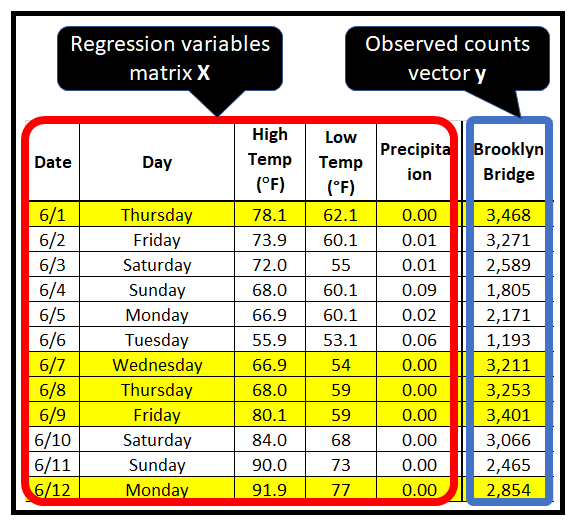
Python-код для генерации счетчиков событий с использованием процесса Пуассона.

# Модель регрессии Пуассона при изменяющемся λ

Теперь перейдем к самому интересному. Рассмотрим более распространенную ситуацию, когда λ может меняться от одного наблюдения к другому. В этом случае предполагаем, что на значение **λ** влияет вектор независимых переменных, также известных как предикторы, регрессионные переменные или регрессоры. Будем обозначать матрицу регрессионных переменных **X**.

Работа модели регрессии состоит в настройке количества наблюдаемых значений у на матрицу **X**.

В наборе данных велосипедистов Нью-Йорка переменными регрессии являются Дата, День недели, Высокая температура, Низкая температура и Осадки. Можно ввести дополнительные регрессоры, например, «Месяц» и «День месяца», являющиеся производными от Дата, и соответственно отказаться от каких-либо существующих регрессоров, в данном случае, от Дата.



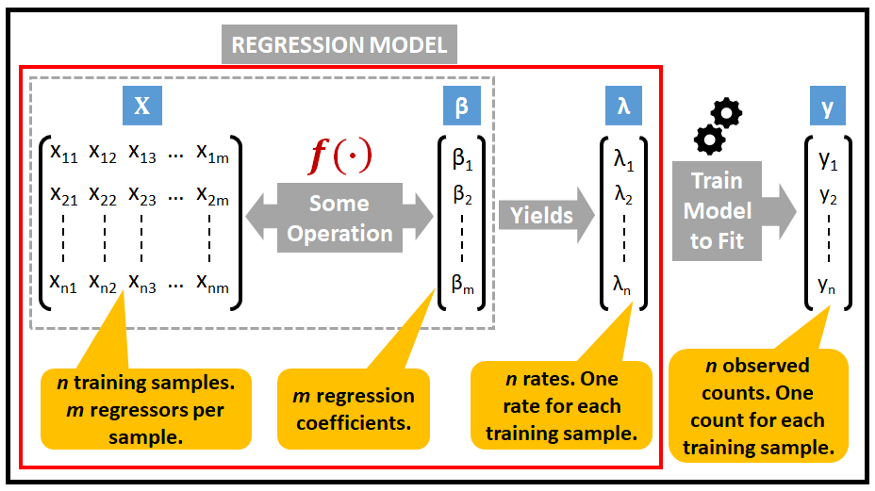
(Изображение [автора](https://sachin-date.medium.com/))

Настройка **y** к **X** происходит путем определения значений вектора коэффициентов регрессии **β**.

В модели регрессии Пуассона предполагается, что количество событий **y** имеет распределение Пуассона, т.е. вероятность наблюдения **y** является функцией вектора частоты событий **λ**.

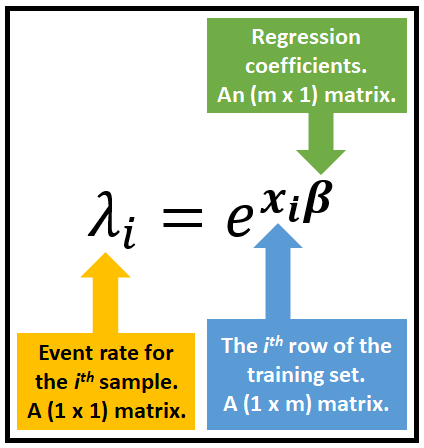
Моделирование регрессии Пуассона состоит в том, чтобы настроить наблюдаемые значения **y** к матрице регрессии **X** с помощью функции связи, выражающую **λ** через коэффициенты регрессии **β** и матрицу регрессии **X**.

На следующем рисунке показана структура модели регрессии Пуассона.



Смотрим слева направо. Структура модели регрессии Пуассона (изображение [автора](https://sachin-date.medium.com/))

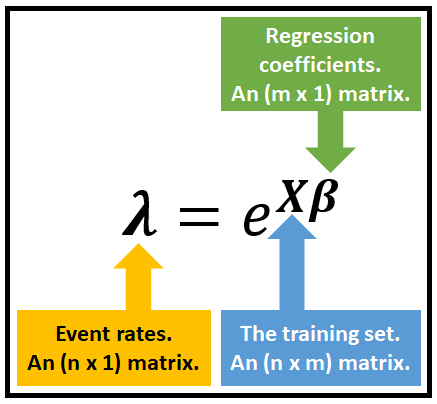
Какой могла бы быть хорошая функция связи ***f*** (.), представляющая **λ** через **X**? Оказывается, отлично работает следующая экспоненциальная функция.



Функция экспоненциальной связи в модели регрессии Пуассона (Изображение [автора](https://sachin-date.medium.com/))

Эта функция связи сохраняет значения **λ** неотрицательными даже в случае отрицательных регрессоров **X** или коэффициентов регрессии **β**. Это требование обязательно для счетных данных.

В векторно-матричной записи имеем.

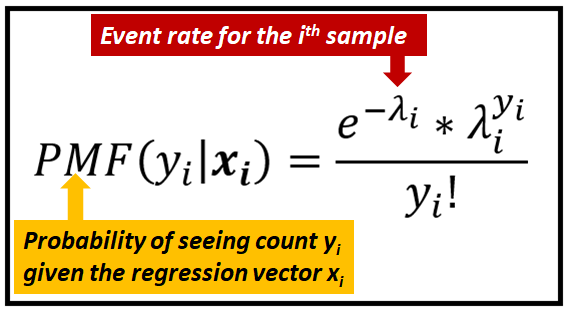


Функция экспоненциальной связи в модели регрессии Пуассона (Изображение [автора](https://sachin-date.medium.com/) )

# Формальное описание (спецификация) модели регрессии Пуассона

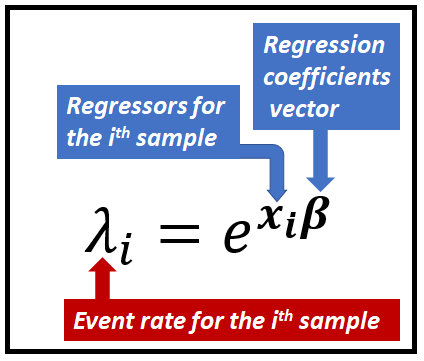
В целом модель регрессии Пуассона для счетных данных определяется следующим образом:

в наборе данных для *i-го* наблюдения, обозначенного *y(i)* и соответствующего строке регрессионных переменных ***x(i)***, вероятность наблюдения значения *y(i)* распределена по Пуассону согласно следующей функции вероятности (PMF).



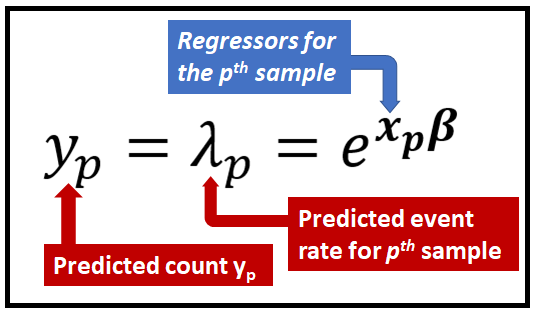
Вероятность наблюдения количества событий y(i) при заданном x(i) (согласно PMF Пуассона) (Изображение [автора](https://sachin-date.medium.com/))

Средняя скорость *λ(i)* для i-й выборки задается рассмотренной выше экспоненциальной функцией связи. Повторим эту функцию.



Функция связи модели регрессии Пуассона (Изображение [автора](https://sachin-date.medium.com/))

После того, как модель обучена, становятся известны коэффициенты регрессии **β**, и модель может давать прогнозы. Для прогноза количества событий *y(p),* соответствующее наблюдаемой входной строке регрессоров ***x(p)***, используется следующая формула.



Выражение прогноза для модели регрессии Пуассона (Изображение [автора](https://sachin-date.medium.com/))

Все сказанное связано с возможностью обучения модели и получения вектора коэффициентов регрессии **β*.***

Посмотрим, как происходит это обучение.

# Обучение модели регрессии Пуассона

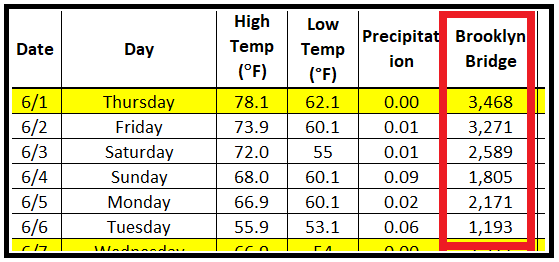
Обучение модели регрессии Пуассона включает в себя поиск значений коэффициентов регрессии **β,** которые сделают вектор наблюдаемых значений **y** наиболее правдоподобным.

Метод определения коэффициентов **β** называется методом оценки максимального правдоподобия (**М**aXimum **L**ikelihood **Е**stimation, MLE).

Познакомимся с **MLE.**

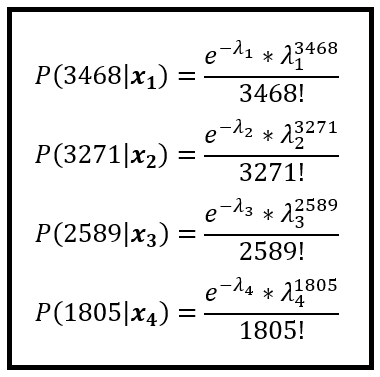
# Понимание оценки максимального правдоподобия (MLE)

Проиллюстрируем **MLE**, используя наш набор данных о велосипедистах. Посмотрим первые несколько строк этого набора.



Ежедневное количество велосипедистов на Бруклинском мосту (изображение [автора](https://sachin-date.medium.com/) )

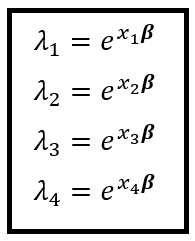
Предполагаем, что количество велосипедистов, показанное в красном прямоугольнике, является результатом пуассоновского процесса. Следовательно, можно сказать, что вероятность таких значений определяется PMF Пуассона. Вот вероятности первых 4 случаев.



Вероятности появления велосипедистов для нескольких первых наблюдений с учетом соответствующих векторов регрессии (Изображение [автора](https://sachin-date.medium.com/))

Аналогичным образом можем вычислить вероятности для всех ***n*** наблюдений, присутствующих в обучающей выборке.

Обратим внимание, что в приведенных выше формулах *λ(1), λ(2), λ(3),…, λ(n)* вычисляются посредством функции связи.

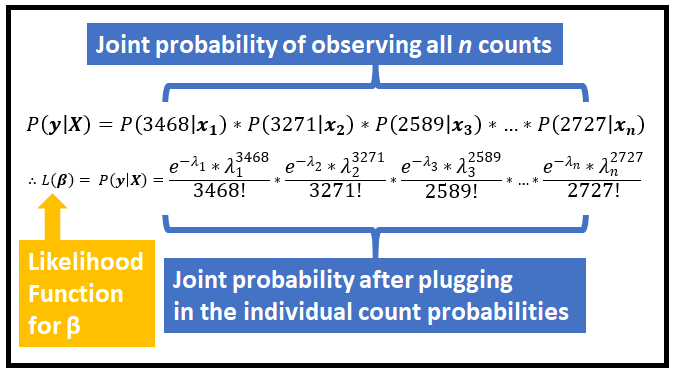


Частота событий, соответствующая наблюдениям за первые несколько дней (Изображение [автора](https://sachin-date.medium.com/))

Где ***x(1), x(2), x(3), x(4)*** – первые 4 строки матрицы регрессии.

Вероятность появления всего набора *y(1), y(2),…, y(n)* в обучающей выборке – совместная вероятность появления отдельных значений *y(i).*

Значения ***у*** подчиняются распределению Пуассона, *y(1), y(2), ..., y(n)* – независимые случайные величины, соответствующие ***x(1), x(2), ..., x(n)*** . Следовательно, совместная вероятность появления *y(1), y(2),…, y(n)* может быть выражена как произведение отдельных вероятностей. Вот как выглядит совместная вероятность для всей обучающей выборки.



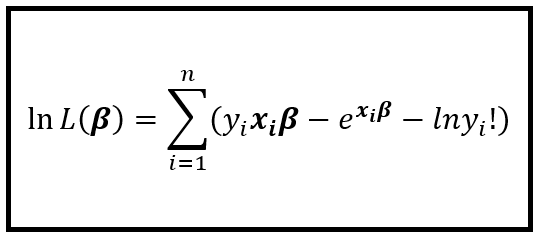
Функция правдоподобия L(β), выраженная как совместная функция вероятности (Изображение [автора](https://sachin-date.medium.com/) )

Напомним*,* что *λ(1), λ(2), λ(3),…, λ(n)* связаны с векторами регрессии **x(1), x(2), x(3),…, x(n)** через коэффициенты регрессии **β*.***

Какое значение **β** сделает данный набор наблюдаемых значений **y** наиболее вероятным? Это значение **β*,*** при котором совместная вероятность, показанная в приведенном выше выражении, достигает максимального значения. Другими словами, это значение **β*,*** для которого скорость изменения совместной функции вероятности относительно **β** равна **0.** Т.е., это решение уравнения, полученного путем дифференцирования уравнения совместной вероятности относительно ***β*** и приравнивания этого уравнения к 0.

Легче дифференцировать логарифм совместной вероятности, чем исходное выражение. Решение соответствующего уравнения дает такое же оптимальное значение ***β***.

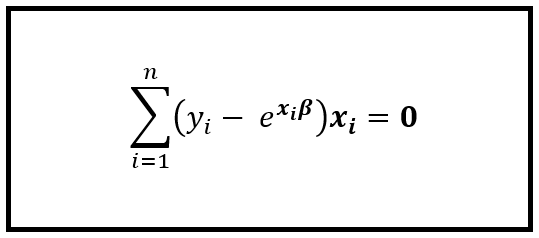
Это логарифмическое выражение называется функцией логарифмического правдоподобия. Для регрессии Пуассона функция логарифмического правдоподобия выглядит следующим образом.



Функция логарифмического правдоподобия для модели регрессии Пуассона (Изображение [автора](https://sachin-date.medium.com/) )

Данное выражение получается путем замены *λ(i)* на *exp* (x**(i)** \* ***β)*** и взятия натурального логарифма от обеих частей совместной вероятности.

Дифференцируем это выражение относительно ***β*** и приравниваем его к нулю. Получаем.



MLE Пуассона для *β является решением этого уравнения* (Изображение [автора](https://sachin-date.medium.com/))

Решение этого уравнения относительно коэффициентов регрессии ***β*** даст оценку максимального правдоподобия (MLE) для ***β.***

Для решения полученного уравнения, используется [итеративный метод наименьших квадратов с повторным изменением весовых коэффициентов (Iteratively Reweighted Least Squares, IRLS)](https://en.wikipedia.org/wiki/Iteratively_reweighted_least_squares). При этом применяется соответствующее статистическое программное обеспечение, как, например, пакет Python statsmodels , который выполняет все вычисления при обучении модели регрессии Пуассона.

# Все шаги выполнения регрессии Пуассона

Сведем вместе шаги выполнения регрессии Пуассона для счетного набора данных.

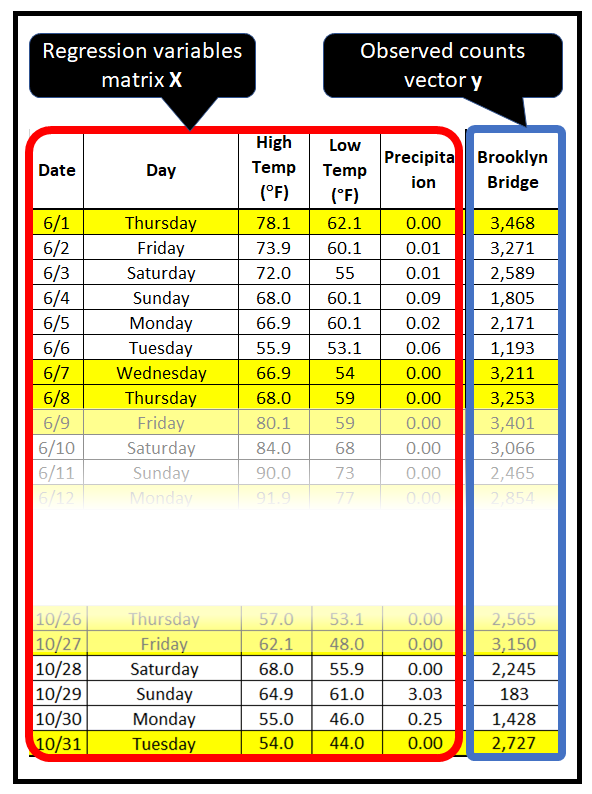
1. Во-первых, убедимся, что наш набор данных действительно является счетным. Один из основных способов подтверждения данного тезиса – это наличие в наборе только неотрицательных целых чисел, которые представляют количество появлений некоторого события в течение некоторого интервала времени. В наборе данных о велосипедистах – это количество велосипедистов, пересекающих Бруклинский мост за день.
2. Найдем (или предположим) переменные регрессии, которые будут влиять на наблюдаемые значения. В наборе данных о велосипедистах переменными регрессии являются Д*ень недели, Минимальная температура, Максимальная температура, Осадки* и т. д.
3. Выделим набор обучающих данных, на котором будет обучаться регрессионная модель, и набор тестовых данных, который пока следует оставить в стороне. Мы не тренируем модель на тестовых данных.
4. Используем подходящее статистическое программное обеспечение, например, пакет Python statsmodels, чтобы настроить модель регрессии Пуассона к набору обучающих данных.
5. Проверим качество модели, запустив ее на тестовых данных, чтобы получить прогнозные значения. Сравним их с фактическими значениями в тестовых данных.
6. Используем степень соответствия (goodness of fit), чтобы определить, насколько хорошо модель обучена на наборе обучающих данных.

# Создание регрессии Пуассона в Python

Применим знания на практике. Пакет Python statmodels предоставляет широкие возможности для выполнения регрессии Пуассона.

Воспользуемся набором данных о велосипедистах на Бруклинском мосту. Этот набор можно найти [**здесь**](https://gist.github.com/sachinsdate/c17931a3f000492c1c42cf78bf4ce9fe)**.**

Цель – построить модель регрессии Пуассона для наблюдаемого количества велосипедистов ***y.*** Обучим, а затем протестируем модель путем прогнозирования ежедневного количества велосипедистов на Бруклинском мосту.



Ежедневное количество велосипедистов на Бруклинском мосту (изображение [автора](https://sachin-date.medium.com/))

Начнем с импорта всех необходимых пакетов.

import pandas as pd  
from patsy import dmatrices  
import numpy as np  
import statsmodels.api as sm  
import matplotlib.pyplot as plt

Создадим DataFrame для набора данных.

df = pd.read\_csv('nyc\_bb\_bicyclist\_counts.csv', header=0, infer\_datetime\_format=True, parse\_dates=[0], indeX\_col=[0])

Добавим несколько регрессионных переменных – производных от переменной Date в матрицу X.

ds = df.index.to\_series()

df['MONTH'] = ds.dt.month

df['DAY\_OF\_WEEK'] = ds.dt.dayofweek

df['DAY'] = ds.dt.day

В дальнейшем не будем использовать переменную Date в качестве регрессора, т.к. она содержит абсолютное значение даты. Но при этом нет необходимости удаления Date, поскольку она задействована в качестве индекса фрейма данных. Таким образом, Date не будет присутствовать в матрице X.

Создадим наборы данных для обучения и тестирования.

mask = np.random.rand(len(df)) < 0.8

df\_train = df[mask]

df\_test = df[~mask]

print('Training data set length='+str(len(df\_train)))

print('Testing data set length='+str(len(df\_test)))

Зададим выражение регрессии в нотации patsy. Указываем patsy, что BB\_COUNT – зависимая переменная, а переменные регрессии: DAY, DAY\_OF\_WEEK, MONTH, HIGH\_T, LOW\_T и PRECIP.

expr = """BB\_COUNT ~ DAY + DAY\_OF\_WEEK + MONTH + HIGH\_T + LOW\_T + PRECIP"""

Создадим матрицы **X** и **y** для наборов обучения и тестирования. patsy делает это очень просто.

y\_train, X\_train = dmatrices(expr, df\_train, return\_type='dataframe')

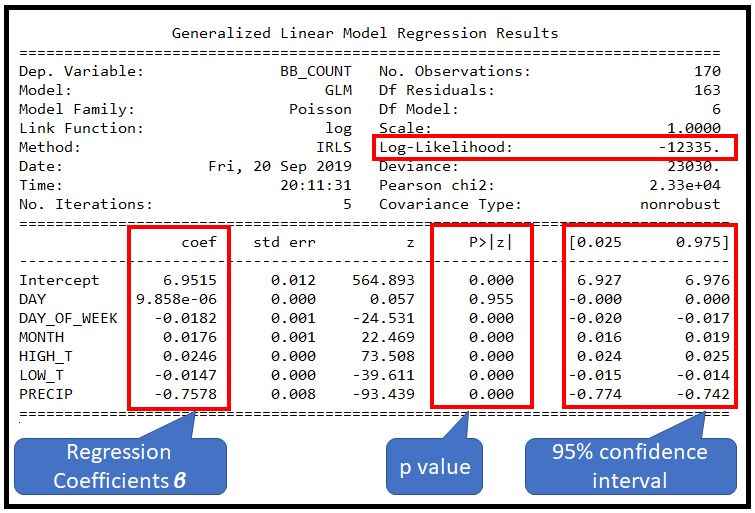
y\_test, X\_test = dmatrices(expr, df\_test, return\_type='dataframe')

Используя класс GLM из statsmodels, обучим модель.

poisson\_training\_results = sm.GLM(y\_train, X\_train, family=sm.families.Poisson()).fit()

Посмотрим результат обучения.

print(poisson\_training\_results.summary())



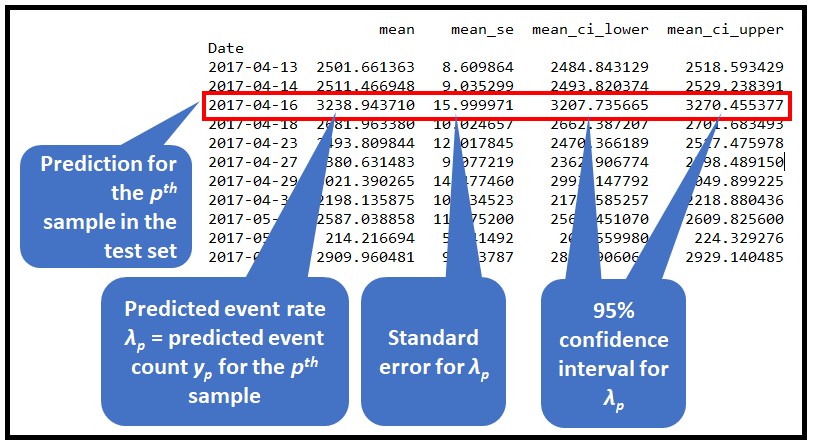
Результат обучения для модели регрессии Пуассона (Изображение [автора](https://sachin-date.medium.com/))

Насколько хорошо обучилась наша модель? Сделаем некоторые прогнозы на тестовом наборе данных.

poisson\_predictions = poisson\_training\_results.get\_prediction(X\_test) #.summary\_frame() returns a pandas DataFrame  
predictions\_summary\_frame = poisson\_predictions.summary\_frame()

print(predictions\_summary\_frame)

Несколько первых строк прогноза.



Первые несколько строк прогноза poisson\_predictions.summary\_frame () (Изображение [автора](https://sachin-date.medium.com/))

Построим прогноз в сравнении с фактом для тестовых данных.

predicted\_counts=predictions\_summary\_frame['mean']

actual\_counts = y\_test['BB\_COUNT']

fig = plt.figure()

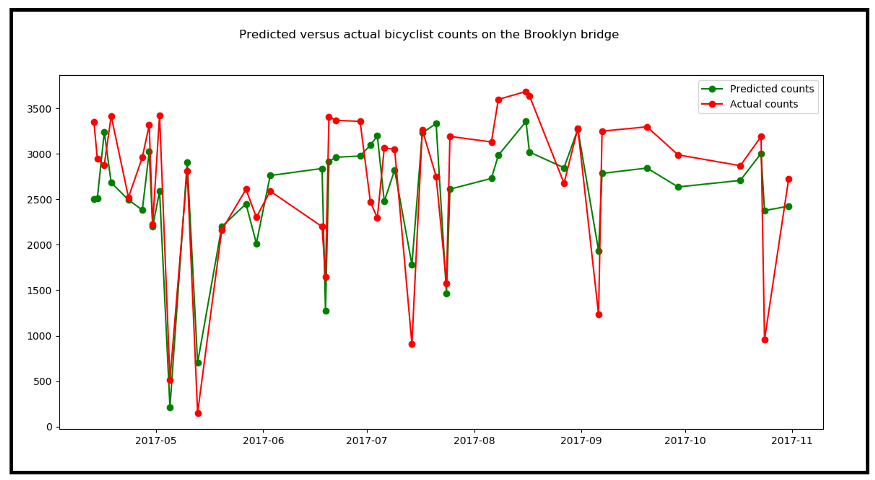
fig.suptitle('Predicted versus actual bicyclist counts on the Brooklyn bridge')

predicted, = plt.plot(X\_test.indeX, predicted\_counts, 'go-', label='Predicted counts')

actual, = plt.plot(X\_test.indeX, actual\_counts, 'ro-', label='Actual counts')

plt.legend(handles=[predicted, actual])

plt.show()



Прогнозируемое и фактическое количество велосипедистов на Бруклинском мосту (изображение [автора](https://sachin-date.medium.com/))

Модель более или менее отражает общий тренд фактических данных, хотя часто присутствуют и существенные отклонения.

Построим график рассеяния факта и прогноза.

plt.clf()

fig = plt.figure()

fig.suptitle('Scatter plot of Actual versus Predicted counts')

plt.scatter(X=predicted\_counts, y=actual\_counts, marker='.')

plt.Xlabel('Predicted counts')

plt.ylabel('Actual counts')

plt.show()

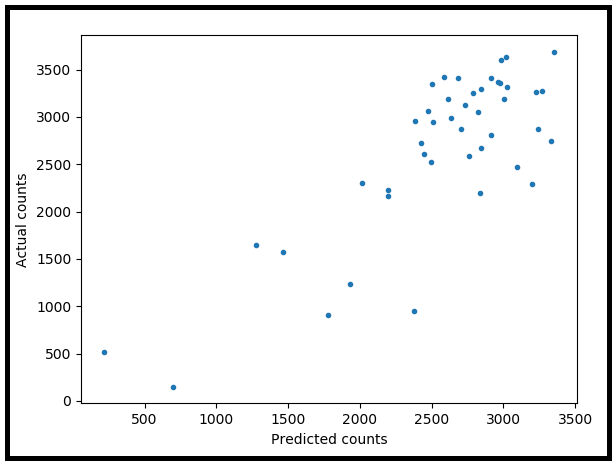


График рассеяния факта и прогноза (изображение [автора](https://sachin-date.medium.com/))

Полный Python код для реализации регрессии Пуассона

import pandas as pd

from patsy import dmatrices

import numpy as np

import statsmodels.api as sm

import matplotlib.pyplot as plt

#Create a pandas DataFrame for the counts data set.

df = pd.read\_csv('nyc\_bb\_bicyclist\_counts.csv', header=0, infer\_datetime\_format=True, parse\_dates=[0], indeX\_col=[0])

#Add a few derived regression variables.

ds = df.index.to\_series()

df['MONTH'] = ds.dt.month

df['DAY\_OF\_WEEK'] = ds.dt.dayofweek

df['DAY'] = ds.dt.day

#Create the training and testing data sets.

mask = np.random.rand(len(df)) < 0.8

df\_train = df[mask]

df\_test = df[~mask]

print('Training data set length='+str(len(df\_train)))

print('Testing data set length='+str(len(df\_test)))

#Setup the regression expression in patsy notation. We are telling patsy that BB\_COUNT is our dependent variable and

# it depends on the regression variables: DAY, DAY\_OF\_WEEK, MONTH, HIGH\_T, LOW\_T and PRECIP.

expr = """BB\_COUNT ~ DAY + DAY\_OF\_WEEK + MONTH + HIGH\_T + LOW\_T + PRECIP"""

#Set up the X and y matrices

y\_train, X\_train = dmatrices(expr, df\_train, return\_type='dataframe')

y\_test, X\_test = dmatrices(expr, df\_test, return\_type='dataframe')

#Using the statsmodels GLM class, train the Poisson regression model on the training data set.

poisson\_training\_results = sm.GLM(y\_train, X\_train, family=sm.families.Poisson()).fit()

#Print the training summary.

print(poisson\_training\_results.summary())

#Make some predictions on the test data set.

poisson\_predictions = poisson\_training\_results.get\_prediction(X\_test)

#.summary\_frame() returns a pandas DataFrame

predictions\_summary\_frame = poisson\_predictions.summary\_frame()

print(predictions\_summary\_frame)

predicted\_counts=predictions\_summary\_frame['mean']

actual\_counts = y\_test['BB\_COUNT']

#Mlot the predicted counts versus the actual counts for the test data.

fig = plt.figure()

fig.suptitle('Predicted versus actual bicyclist counts on the Brooklyn bridge')

predicted, = plt.plot(X\_test.indeX, predicted\_counts, 'go-', label='Predicted counts')

actual, = plt.plot(X\_test.indeX, actual\_counts, 'ro-', label='Actual counts')

plt.legend(handles=[predicted, actual])

plt.show()

#Show scatter plot of Actual versus Predicted counts

plt.clf()

fig = plt.figure()

fig.suptitle('Scatter plot of Actual versus Predicted counts')

plt.scatter(X=predicted\_counts, y=actual\_counts, marker='.')

plt.Xlabel('Predicted counts')

plt.ylabel('Actual counts')

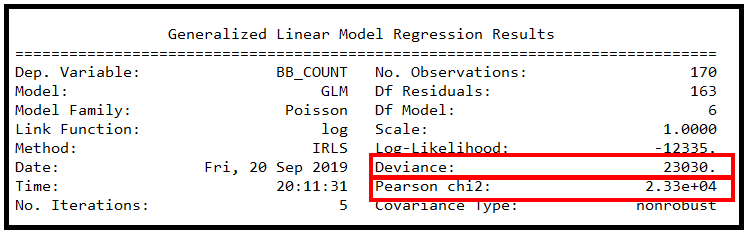
plt.show()

# Степень соответствия модели регрессии Пуассона

Вспомним, что как ожидаемое (среднее) значение, так и дисперсия распределения Пуассона должны быть равны λ. Это довольно строгое условие часто нарушается в реальных наборах данных.

Несоответствие данных критерию среднее = дисперсия, налагаемому распределением Пуассона, является распространенным источником неудач моделирования посредством регрессии Пуассона.

Метод summary() в классе [GLMResult](https://www.statsmodels.org/devel/generated/statsmodels.genmod.generalized_linear_model.GLMResults.html)s из statsmodels показывает пару полезных статистик степени соответствия для оценки успешности модели регрессии Пуассона при обучении. Посмотрим на эти статистики.



Результат обучения для модели регрессии Пуассона (Изображение [автора](https://sachin-date.medium.com/))

Полученные значения дисперсии и хи-квадрат Пирсона очень велики. При таких значениях практически невозможно качественно настроить модель. Чтобы количественно определить степень соответствия на некотором уровне достоверности, например, 95% (p = 0,05), ищем значение в таблице χ2 для p = 0,05 и степеней свободы остатков (Degrees of freedom of residuals, DF Residuals) = 163. (DF Residuals = количество наблюдений минус степени свободы модели). Сравниваем значение хи-квадрат с наблюдаемой статистикой, в данном случае с дисперсией или значением хи-квадрат Пирсона, указанным в GLMResults. Получили, что при p = 0,05 и DF Residuals = 163 значение хи-квадрат из [стандартной таблицы хи-квадрат](https://www.medcalc.org/manual/chi-square-table.php) составляет 193,791, что много меньше, чем заявленные статистические данные 23030 и 23300. Следовательно, согласно этому тесту, модель регрессии Пуассона, несмотря на хорошее визуальное соответствие для тестовых данных, довольно плохо соответствовала обучающему набору.

# Заключение и следующие шаги

Для счетных данных разумной отправной точкой является модель регрессии Пуассона. Далее результат ее работы можно сравнивать с другими популярными моделями для счетных данных, например:

* [нуль-инфляционная модель регрессии Пуассона](https://towardsdatascience.com/an-illustrated-guide-to-the-zero-inflated-poisson-model-b22833343057) (модель регрессии Пуассона с избыточным количеством нулей), если предполагается, что данные содержат много больше нулей, чем может объяснить обычная модель Пуассона.
* модель [отрицательной биномиальной регрессии,](https://towardsdatascience.com/negative-binomial-regression-f99031bb25b4) которая не требует предположения среднее значение = дисперсия.
* [обобщенные модели регрессии Пуассона,](https://towardsdatascience.com/generalized-poisson-regression-for-real-world-datasets-d1ff32607d79) которые хорошо работают с чрезмерно или недостаточно рассредоточенными данными.

Удачного моделирования!

*Спасибо за прочтение! Я пишу на темы науки о данных, уделяя особое внимание регрессии и анализу временных рядов.*

*Если вам понравилась эта статья, подпишитесь на меня на* [*Sachin Date,*](https://sachin-date.medium.com/) *чтобы получать рекомендации, инструкции и советы по программированию по темам, посвященным регрессии и анализу временных рядов.*

[Sachin Date](https://sachin-date.medium.com/?source=post_sidebar--------------------------post_sidebar-----------)