# Модели регрессии Пуассона для временных рядов

<https://towardsdatascience.com/poisson-regression-models-for-time-series-data-sets-54114e68c46d>

Как, используя Python и Statsmodels, построить модель регрессии Пуассона для наборов данных, представляющих собой счетные временные ряды.

[[Дата Sachin](https://sachin-date.medium.com/?source=post_page-----54114e68c46d--------------------------------)](https://sachin-date.medium.com/?source=post_page-----54114e68c46d--------------------------------)

[Sachin Date](https://sachin-date.medium.com/?source=post_page-----54114e68c46d--------------------------------)

[25.04.2021](https://towardsdatascience.com/poisson-regression-models-for-time-series-data-sets-54114e68c46d?source=post_page-----54114e68c46d--------------------------------)

Оглавление

[Набор данных MANUFACTURING STRIKES 1](#_Toc75616152)

[Степень соответствия 5](#_Toc75616153)

[Построение авторегрессионной модели Пуассона 6](#_Toc75616154)

[Степень соответствия 8](#_Toc75616155)

[Значимость переменных 8](#_Toc75616156)

[Прогнозирование 8](#_Toc75616157)

[Следующие шаги 9](#_Toc75616158)

[Литература 10](#_Toc75616159)

[Пуассоновские](https://towardsdatascience.com/an-illustrated-guide-to-the-poisson-regression-model-50cccba15958) и [пуассон](https://towardsdatascience.com/generalized-poisson-regression-for-real-world-datasets-d1ff32607d79)-подобные регрессионные модели часто используются для счетных наборов данных, а именно для данных, имеющих целые нумерованные значения. Например, количество людей, приходящих каждый час в отделение неотложной помощи, является одним из примеров подобного набора.

Обычные регрессионные линейные модели на основе наименьших квадратов или нелинейные модели, основанные, например, на техниках нейронных сетей, не работают для таких наборов данных, поскольку указанные модели могут предсказать отрицательные значения.

Если набор данных является **счетной временной последовательностью, дополнительную** сложность может привнести возможная автокорреляция данных. Предыдущие измерения влияют на значения будущих. Если регрессионная модель не может адекватно уловить информацию, содержащуюся в этих корреляциях, то необъясненная информация просочится в остаточные ошибки (отклонения) модели в виде автокоррелированных ошибок. В этом случае низкой окажется [**степень соответствия**](https://towardsdatascience.com/the-complete-guide-to-r-squared-adjusted-r-squared-and-pseudo-r-squared-4136650fc06c) модели.

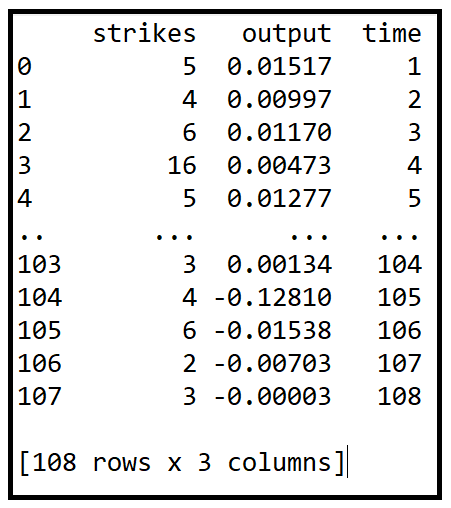
Общие способы решения этой проблемы:

1. Прежде чем подбирать регрессионную модель, проверьте, демонстрирует ли временной ряд сезонность, и, если да, [**выполните корректировку сезонности**](https://towardsdatascience.com/what-is-time-series-decomposition-and-how-does-it-work-9b67e007ae90). Это объяснит сезонные автокорреляции, если таковые имеются.
2. **Рассчитайте первую разность временного ряда,** то есть y(t) - y (t-1) для всех t, и [**выполните тест белого шума**](https://towardsdatascience.com/the-white-noise-model-1388dbd0a7d) для разностного временного ряда. Если разностный временной ряд представляет собой [белый шум](https://towardsdatascience.com/the-white-noise-model-1388dbd0a7d), то исходный временной ряд можно считать [случайным блужданием](https://towardsdatascience.com/the-white-noise-model-1388dbd0a7d). В этом случае дальнейшее моделирование не имеет смысла.
3. Настройте пуассоновскую (или родственную) регрессионную модель для сезонно скорректированного временного ряда, но добавьте запаздывающие (смещенные) копии зависимой переменной **y** в качестве независимых переменных регрессии.

В этой статье мы объясним, как настроить пуассоновскую или пуассон-подобную модель к временному ряду, используя третий **способ**.

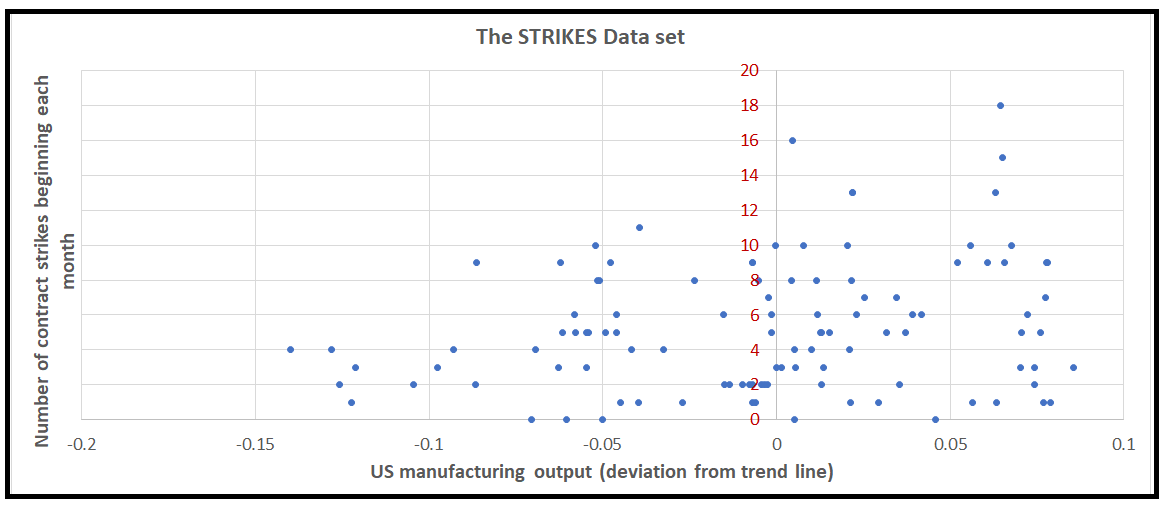
## Набор данных MANUFACTURING STRIKES

Чтобы проиллюстрировать процедуру подбора модели, будем использовать открытый набор данных, который широко используется в литературе по регрессионному моделированию.



Забастовки на производстве (Источник данных: [US BLS](https://www.bls.gov/) в [наборах данных R](https://github.com/vincentarelbundock/Rdatasets/blob/master/datasets.csv#L609) )

Набор данных представляет собой ежемесячный временной ряд, показывающий взаимосвязь между производственной активностью в США, измеряемой как отклонение от линии тренда (departure from the trend line), и количеством легальных забастовок (contract strikes) в обрабатывающей промышленности США с 1968 по 1976 год.



Набор данных STRIKES (Источник: [наборы данных R](https://github.com/vincentarelbundock/Rdatasets/blob/master/datasets.csv#L609) ) (Изображение [автора](https://sachin-date.medium.com/) )

Этот набор данных доступен в R, его можно получить с помощью [пакета statsmodels Datasets](https://www.statsmodels.org/devel/datasets/index.html) .

Зависимая переменная **y** – srikes *(*забастовки)*.*

Начнем с импорта необходимых пакетов.

import statsmodels.api as sm

import statsmodels.discrete.discrete\_model as dm

import numpy as np

from patsy import dmatrices

import statsmodels.graphics.tsaplots as tsa

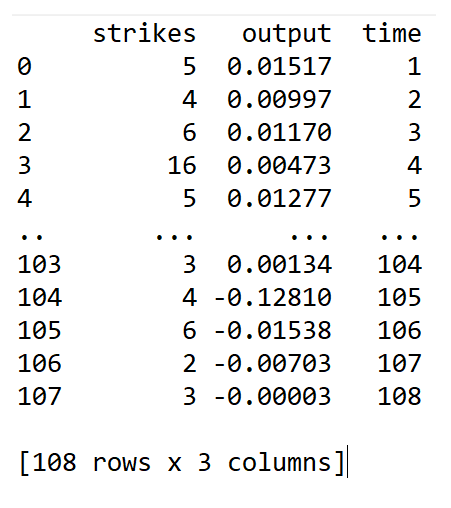
from matplotlib import pyplot as plt

Загрузим набор данных с помощью statsmodels.

strikes\_dataset = sm.datasets.get\_rdataset(dataname='StrikeNb', package='Ecdat')

Выведем набор.

print (strikes\_dataset.data)



Будем рассматривать первые 92 точки как обучающий набор, а оставшиеся 16 как тестовый.

strikes\_data = strikes\_dataset.data. copy ()   
strikes\_data\_train = strikes\_data.query(‘time<=92' )strikes\_data\_test = strike\_data.query(‘time>92').reset\_index (). drop (‘index’, axis=1)

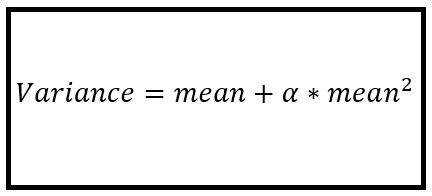
Распечатаем статистику для зависимой переменной.

print('Mean='+str(np.mean(strikes\_data\_train['strikes'])) + ' Variance='+str(np.var(strikes\_data\_train['strikes'])))

Получим.

Mean=5.5 Variance=14.728260869565217

Видим, что **y** чрезмерно изменчива, нарушается предположение модели Пуассона среднее значение = дисперсия. Чтобы учесть избыточную дисперсию, [**выберем модель отрицательной биномиальной регрессии,**](https://towardsdatascience.com/negative-binomial-regression-f99031bb25b4) имеющую следующую функцию дисперсии NB2:



Функция дисперсии модели NB2 (Изображение [автора](https://sachin-date.medium.com/) )

Ниже наше выражение регрессии. strikes - зависимая переменная, output - независимая. Предполагается, что присутствует свободный член регрессии (intercept of regression).

expr = 'strikes ~ output'

Будем использовать [Patsy](https://patsy.readthedocs.io/en/latest/quickstart.html) для чтобы выделить матрицу **X** и столбец ***y***.Patsy автоматически добавит столбец свободных членов в **X**.

y\_train, X\_train = dmatrices(expr, strikes\_data\_train, return\_type='dataframe')  
print(y\_train)  
print(X\_train)  
  
y\_test, X\_test = dmatrices(expr, strikes\_data\_test, return\_type='dataframe')  
print(y\_test)  
print(X\_test)

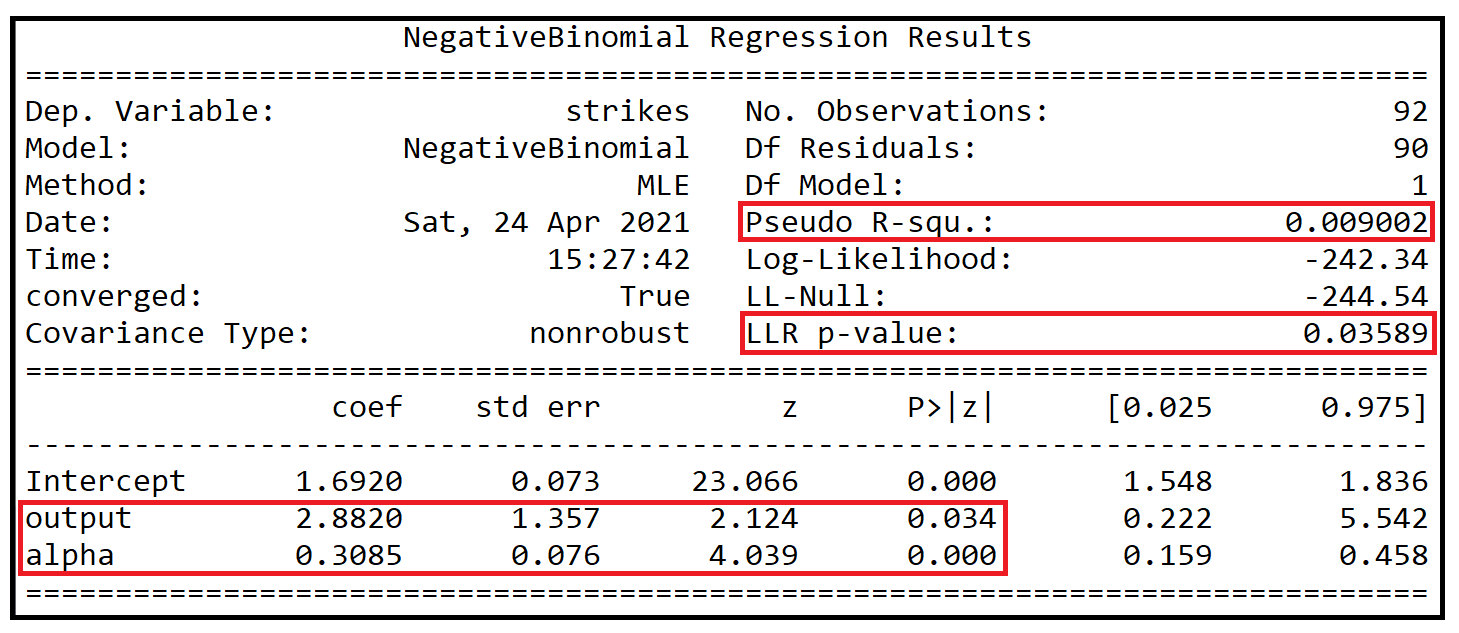
Построим и обучим модель отрицательной биномиальной регрессии с функцией дисперсии NB2.

nb2\_model = dm.NegativeBinomial(endog=y\_train, exog=X\_train, loglike\_method='nb2')

nb2\_model\_results = nb2\_model.fit(maxiter=100)

print(nb2\_model\_results.summary())

Получим следующее.



Результат моделирования посредством NB2 (Изображение [автора](https://sachin-date.medium.com/) )

Как активность, так и параметр дисперсии альфа являются статистически значимыми при уровне доверия (доверительной вероятности, надежности) 95%, на что указывают p-значения (0,034 и 0,000) их коэффициентов.

## Степень соответствия

[**Псевдо R**](https://towardsdatascience.com/the-complete-guide-to-r-squared-adjusted-r-squared-and-pseudo-r-squared-4136650fc06c)**2** равен 0,9%, что говорит об **очень низком качестве настройки модели** на множестве обучающих данных.

p – значение для теста отношения логарифмического правдоподобия составляет 0,03589, что указывает лучшую работу модели по сравнению с моделью Intercept Only (Null Model, модель без предикторов) с уровнем доверия 95%, но не с уровнем 99% или выше.

Давайте посмотрим на график автокорреляции остаточных ошибок настроенной модели.

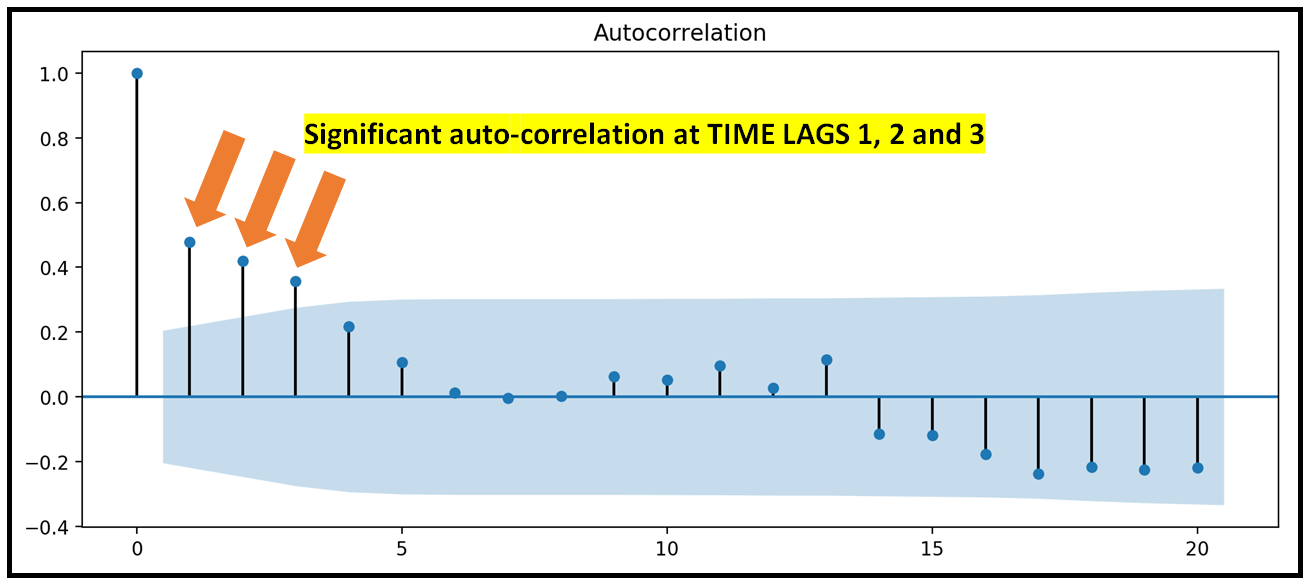


График автокорреляции остаточных ошибок модели NB2 (Изображение [автора](https://sachin-date.medium.com/) )

Видим, что остаточные ошибки, которые значимо автокоррелируются на временных лагах 1, 2 и 3, показывают, что существует автокорреляция в зависимой переменной strikes, которую модель NB2 не в состоянии полностью объяснить, заставляя автокорреляцию просочиться в отклонения модели.

В целом качество данной модели очень низкое.

## Построение авторегрессионной модели Пуассона

Чтобы исправить ситуацию с автокоррелированными остатками (отклонениями), введем запаздывающие (смещенные) копии **y**, а именно, y (t-1), y (t-2) и y (t-3) в качестве регрессионных переменных в дополнение к переменной *output*.

Но вместо того, чтобы напрямую вводить y (t-k) в качестве переменных регрессии, будем использовать ln [y (t-k)], чтобы решить проблему «взрыва модели», когда коэффициент при y(t-k) положителен.

При использовании преобразования ln () возникает вопрос, как поступать с нулевыми значениями y(t), для которых логарифм не определен.

Решаем эту проблему, используя прием, описанный Кэмероном и Триведи в их книге *«*Регрессионный анализ счетных данных» (см. раздел 7.5. Авторегрессионые модели).

Определим новую индикаторную переменную d(t) для каждого интересующего нас временного лага следующим образом:

* если y(t) = 0, то d(t) = 1 и зададим y(t) равной 0;
* если y(t)> 0. то d(t) = 0 и оставим y(t) как есть.

Давайте внесем эти изменения во фрейм данных.

Создадим новый столбец strikes\_adj, который устанавливается в 1, если strikes <1, иначе равным strikes.

strikes\_data['strikes\_adj'] = np.maximum(1, strikes\_data['strikes'])

Определим функцию, которая будет устанавливать значение индикаторной переменной d(t).

def indicator\_func (x):   
 if x == 0:   
 return 1   
 else:   
 return 0

И применим эту функцию для создания нового столбца индикаторной переменной.

strikes\_data['d'] = strikes\_data['strikes'].apply(indicator\_func)

Создадим переменные с лагом для strikes\_adj и d.

strikes\_data['ln\_strikes\_adj\_lag1'] = strikes\_data['strikes\_adj'].shift(1)  
strikes\_data['ln\_strikes\_adj\_lag2'] = strikes\_data['strikes\_adj'].shift(2)  
strikes\_data['ln\_strikes\_adj\_lag3'] = strikes\_data['strikes\_adj'].shift(3)  
  
strikes\_data['d\_lag1'] = strikes\_data['d'].shift(1)  
strikes\_data['d\_lag2'] = strikes\_data['d'].shift(2)  
strikes\_data['d\_lag3'] = strikes\_data['d'].shift(3)

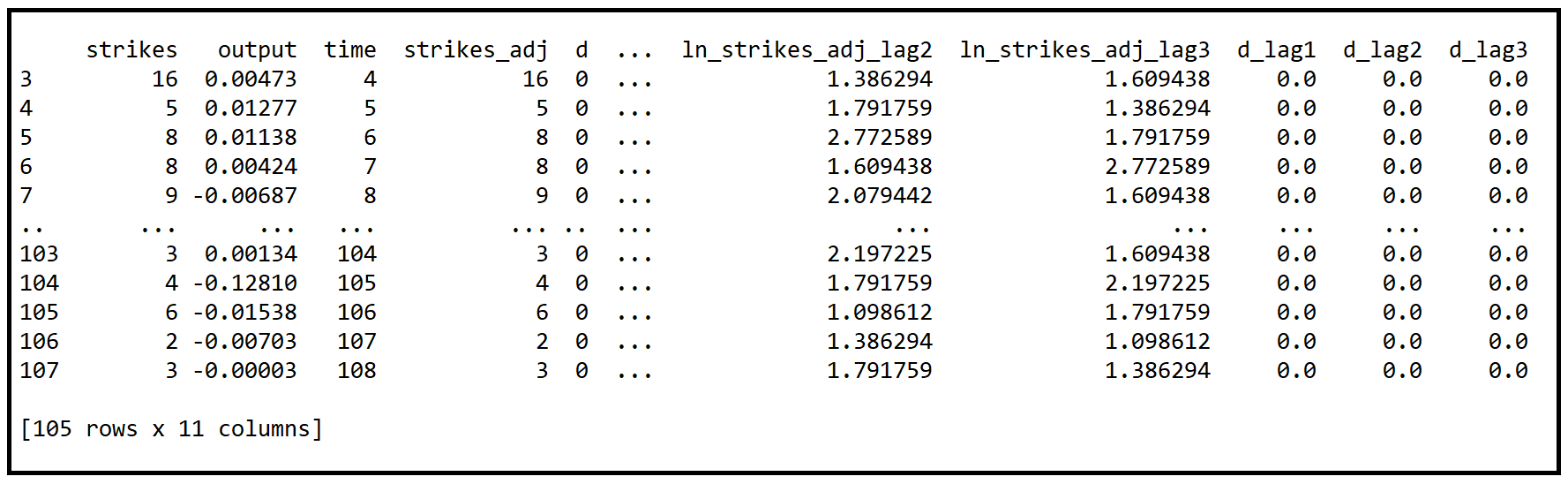
Удалим все строки с пустыми ячейками.

strikes\_data = strikes\_data.dropna()

Наконец, возьмем натуральный логарифм ln\_strikes\_adj\_lag1*[[1]](#footnote-1)*. Помним, что мы хотим добавить натуральный логарифм запаздывающих переменных y (t-1), y(t-2) и y (t-3).

Посмотрим, как теперь выглядит наш набор данных.

print(strikes\_data)



Фрейм данных о забастовках с добавленными переменными с запаздыванием

Давайте вновь разделим набор данных на обучающий и тестовый наборы.

strikes\_data\_train=strikes\_data.query('time<=92')

strikes\_data\_test=strikes\_data.query('time>92').reset\_index().drop('index', axis=1)

Обновим регрессию, включив в нее запаздывающие переменные.

expr = 'strikes ~ output + ln\_strikes\_adj\_lag1 + ln\_strikes\_adj\_lag2 + ln\_strikes\_adj\_lag3 + d\_lag1 + d\_lag2 + d\_lag3'

Применяем Patsy для создания **y** и **X**.

y\_train, X\_train = dmatrices(expr, strikes\_data\_train, return\_type='dataframe')print(y\_train)print(X\_train)

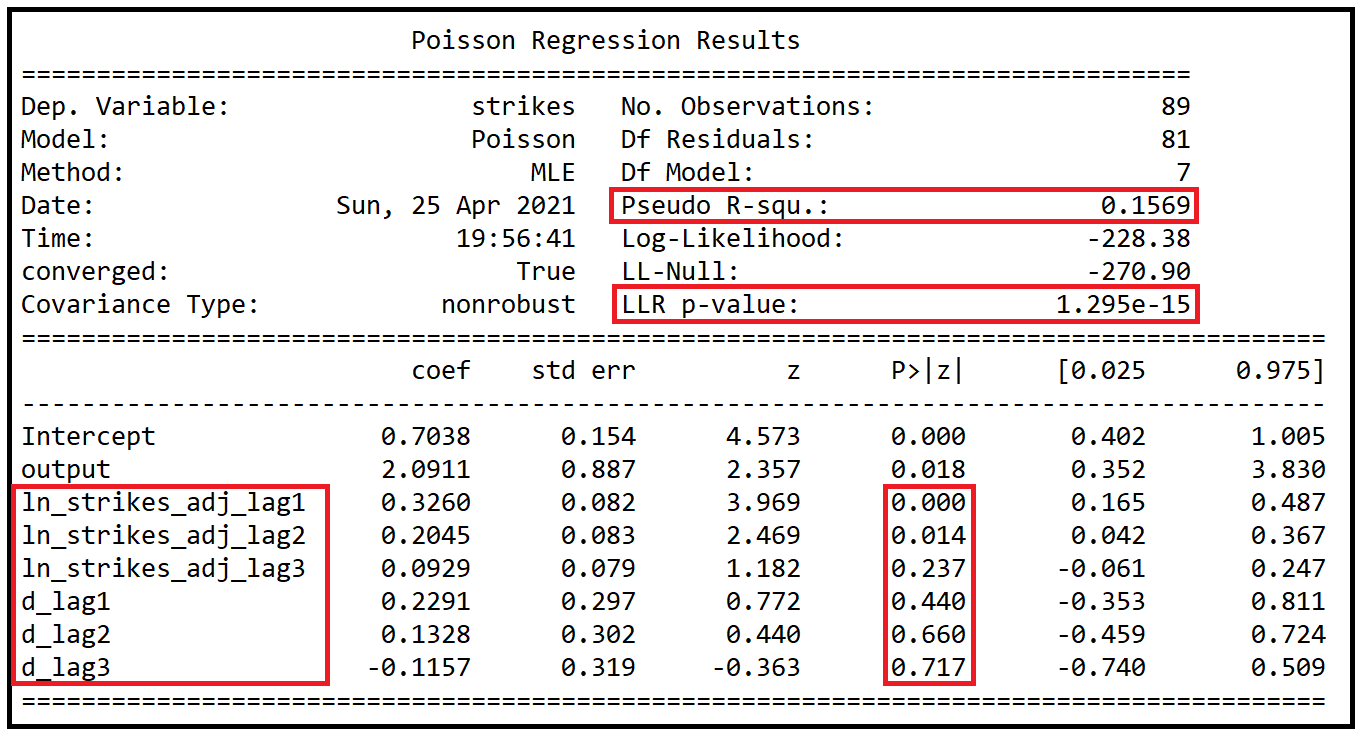
Наконец, создадим и настроим регрессионную модель на (**y\_train**, **X\_train**). На этот раз используем прямую модель регрессии Пуассона.

poisson\_model = dm.Poisson(endog=y\_train, exog=X\_train)

poisson\_model\_results = poisson\_model.fit(maxiter=100)

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

Получим следующее.



Результаты моделирования регрессии Пуассона с запаздывающими выходными переменными

### Степень соответствия

Первое, что следует отметить, это что степень соответствия, измеренная посредством [псевдо-R2](https://towardsdatascience.com/the-complete-guide-to-r-squared-adjusted-r-squared-and-pseudo-r-squared-4136650fc06c) , улучшилась по сравнению с моделью NB2 с 0,9%до15,69% . Это значительное улучшение. На этот раз p-значение теста LLR также исчезающе мало 1,295e-15. Последнее означает, что можно почти со 100% уверенностью сказать, что модель Пуассона с запаздывающими переменными лучше, чем модель Intercept Only. Вспомним, что ранее мы могли сказать это с уровнем доверия всего 95%.

Давайте посмотрим на график автокорреляции остатков этой модели Пуассона с лаговой переменной.

tsa.plot\_acf(poisson\_model\_results.resid, alpha=0.05)plt.show()

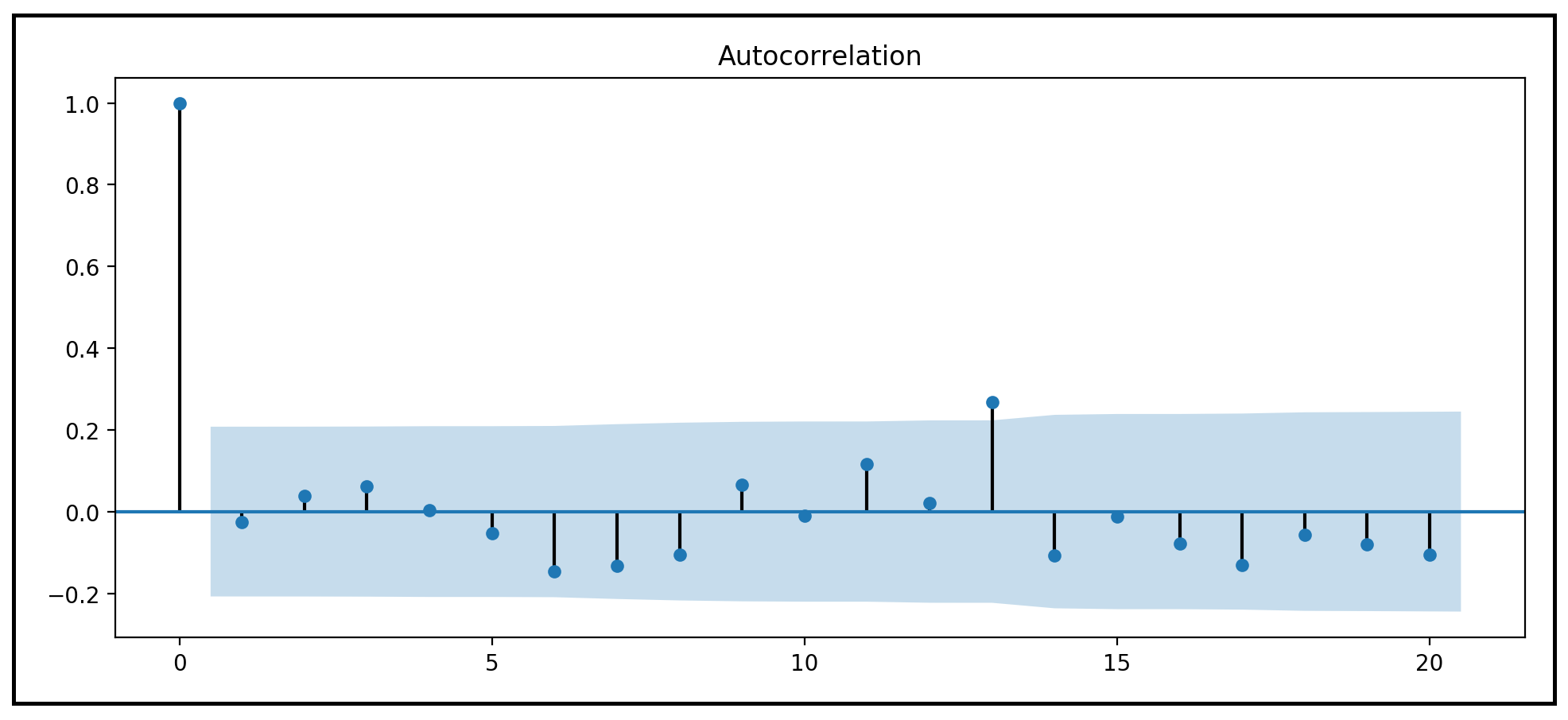


График автокорреляции остаточных ошибок модели Пуассона с запаздыванием (Изображение [автора](https://sachin-date.medium.com/) )

За исключением очень умеренно значимой корреляции на LAG 13, корреляция остатков по всем остальным лагам находится в пределах установленных альфа-границ.

Наша стратегия добавления запаздывающих копий strikes к регрессионным переменным модели Пуассона, по-видимому, объяснила большую часть автокорреляции в переменной strikes.

### Значимость переменных

Отметим в результатах моделирования по Пуассону с запаздывающими переменными, что в то время как коэффициенты output, ln\_strikes\_adj\_lag1 и ln\_strikes\_adj\_lag2 имеют уровень доверия 95%, коэффициент переменной третьего лага ln\_strikes\_adj\_lag3 значим только на 75% как указано в его р-значении равном 0.237. Более того, все три лаговые индикаторные переменные d\_lag1, d\_lag2 и d\_lag3 оказались статистически недостоверными при уровне доверия 95%.

### Прогнозирование

Воспользуемся настроенной моделью Пуассона с запаздывающими переменными, чтобы спрогнозировать количество забастовок на тестовом наборе. Не стоит возлагать слишком большие надежды на качество прогноза. Помним, что хотя эта модель работает много лучше, чем предыдущая модель NB2, псевдо-R2 составляет всего 16%.

Используем Patsy для конструирования **y\_test**, **X\_test**).

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

print(y\_test)

print(X\_test)

Сделаем прогноз на **X\_test**.

poisson\_predictions = poisson\_model\_results.predict(X\_test)

Построим прогноз и факт.

predicted\_counts=poisson\_predictions

actual\_counts = y\_test['strikes']

fig = plt.figure()fig.suptitle('Predicted versus actual strike counts')

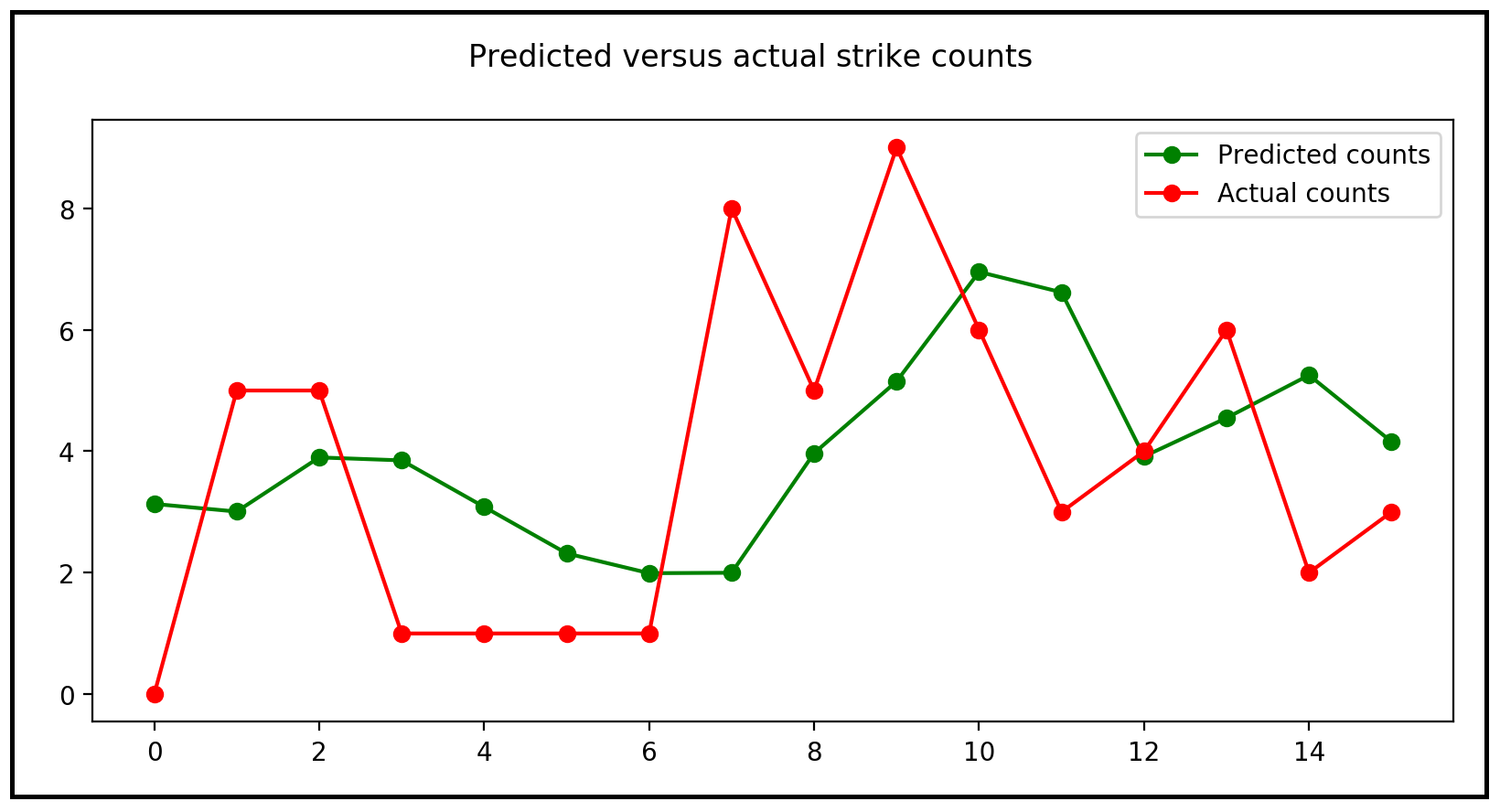
predicted, = plt.plot(X\_test.index, predicted\_counts, 'go-', label='Predicted counts')

actual, = plt.plot(X\_test.index, actual\_counts, 'ro-',

label='Actual counts')

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

plt.show()



Прогнозируемое и фактическое количество забастовок с использованием модели Пуассона с запаздывающими переменными (Изображение [автора](https://sachin-date.medium.com/) )

## Следующие шаги

Можно попробовать улучшить качество настройки нашей модели с переменными с запаздыванием, экспериментируя со следующим:

* включить первые три временных лага (запаздывания) переменной *output* в качестве переменных регрессии в дополнение к переменной output;
* включить оба значения временных лагов переменных *output* и *strikes* в качестве регрессионных переменных;
* вместо использования модели Пуассона применить отрицательную биномиальную модель (с использованием функции дисперсии NB1 или NB2) с указанными выше видами переменных с запаздыванием в качестве переменных регрессии.

Кроме того, было бы интересно использовать структуру [обобщенной линейной модели ([Generalized Linear Model](https://www.statsmodels.org/stable/glm.html)),](https://www.statsmodels.org/stable/glm.html) существующую в statsmodels, для построения и обучения модели Пуассона или отрицательной биномиальной модели. См. ссылки ниже о том, как создавать и обучать GLM модели.

## Литература

Cameron A. Colin, Trivedi Pravin K., [*Regression Analysis of Count Data*](http://cameron.econ.ucdavis.edu/racd/count.html), Econometric Society Monograph №30, Cambridge University Press, 1998. ISBN: 0521635675

McCullagh P., Nelder John A., [*Generalized Linear Models*](https://www.routledge.com/Generalized-Linear-Models/McCullagh-Nelder/p/book/9780412317606), 2nd Ed., CRC Press, 1989, ISBN 0412317605, 9780412317606

Kennan J., [*The duration of contract strikes in U.S. manufacturing, Journal of Econometrics*](https://www.sciencedirect.com/science/article/pii/0304407685900648), Volume 28, Issue 1, 1985, Pages 5–28, ISSN 0304–4076, <https://doi.org/10.1016/0304-4076(85)90064-8.> [PDF download link](https://www.ssc.wisc.edu/~jkennan/research/JEM85.pdf)

Cameron C. A., Trivedi P. K., [*Regression-based tests for overdispersion in the Poisson model*](https://www.sciencedirect.com/science/article/pii/030440769090014K), Journal of Econometrics, Volume 46, Issue 3, 1990, Pages 347–364, ISSN 0304–4076, <https://doi.org/10.1016/0304-4076(90)90014-K.>

1. Прим. перев. По-видимому, должно быть «… ln\_strikes\_adj\_lag1, ln\_strikes\_adj\_lag2, ln\_strikes\_adj\_lag3.» [↑](#footnote-ref-1)