Анализ временных рядов

- В разных отраслях организации используют данные временных рядов, что означает любую информацию, собранную за регулярные промежутки времени, в своей деятельности. Примеры включают ежедневные цены на акции, уровни энергопотребления, показатели вовлеченности в социальных сетях, розничный спрос и другие. Анализ данных временных рядов дает такие сведения, как тенденции, сезонные закономерности и прогнозы будущих событий, которые могут помочь в получении прибыли. Например, понимая сезонные тенденции спроса на розничные товары, компании могут планировать рекламные акции, чтобы максимизировать продажи в течение всего года.
- При анализе данных временных рядов необходимо выполнить ряд шагов. Во-первых, вам нужно проверить на стационарность и автокорреляцию. Стационарность — это способ измерить, имеют ли данные структурные закономерности, такие как сезонные тренды. Автокорреляция возникает, когда будущие значения во временном ряду линейно зависят от прошлых значений. Необходимо проверить оба этих параметра в данных временных рядов, потому что они являются предположениями, которые делаются многими широко используемыми методами анализа временных рядов. Например, метод авторегрессионного интегрированного скользящего среднего (ARIMA) для прогнозирования временных рядов предполагает стационарность. Кроме того, линейная регрессия для прогнозирования временных рядов предполагает, что данные не имеют автокорреляции. Прежде чем проводить эти процессы, необходимо знать, пригодны ли данные для анализа.
- Во время анализа временных рядов также необходимо выполнить декомпозицию тренда и прогнозировать будущие значения. Декомпозиция позволяет визуализировать тенденции в данных, что является отличным способом четкого объяснения их поведения. Наконец, прогнозирование позволяет предвидеть будущие события, которые могут помочь в принятии решений.

Декомпозиция временных рядов

Данные временных рядов могут демонстрировать различные шаблоны, и часто полезно разделить временной ряд на несколько компонентов, каждый из которых представляет базовую категорию шаблонов (pattern).

Существуют три типа паттернов временных рядов: тренд, сезонность и циклы. Когда мы разлагаем временной ряд на компоненты, мы обычно объединяем тренд и цикл в один компонент тренд-цикл (иногда для простоты называемый трендом). Таким образом, мы рассматриваем временной ряд как состоящий из трех компонентов: компонента **тренда, сезонная компонента и остаточная компонента** (содержащего все остальное во временном ряду).

Используются методы извлечения этих компонент из временного ряда. Часто это делается для улучшения понимания временных рядов, но также может быть использовано для повышения точности прогноза.

- Если предположить аддитивное разложение, то можно написать

$$y_t = S_t + T_t + R_t,$$

- В качестве альтернативы мультипликативное разложение

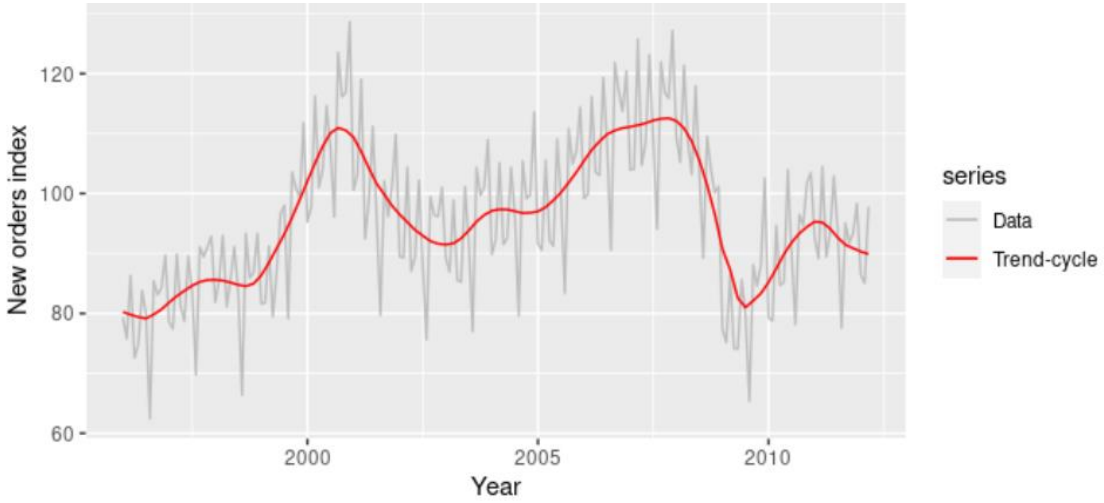
$$y_t = S_t \times T_t \times R_t.$$

- Аддитивная декомпозиция является наиболее подходящей, если величина сезонных колебаний или вариация вокруг цикла тренда не зависит от уровня временного ряда. Когда вариация сезонной модели или вариация вокруг цикла тренда пропорциональна уровню временного ряда, тогда более подходящим является мультипликативное разложение. Мультипликативные разложения распространены в экономических временных рядах.
- Альтернативой использованию мультипликативной декомпозиции является сначала преобразование данных до тех пор, пока изменение ряда не станет стабильным во времени, а затем использование аддитивной декомпозиции. Когда использовалось логарифмическое преобразование, это эквивалентно использованию мультипликативной декомпозиции, потому что

$$y_t = S_t \times T_t \times R_t \quad \text{is equivalent to} \quad \log y_t = \log S_t + \log T_t + \log R_t.$$

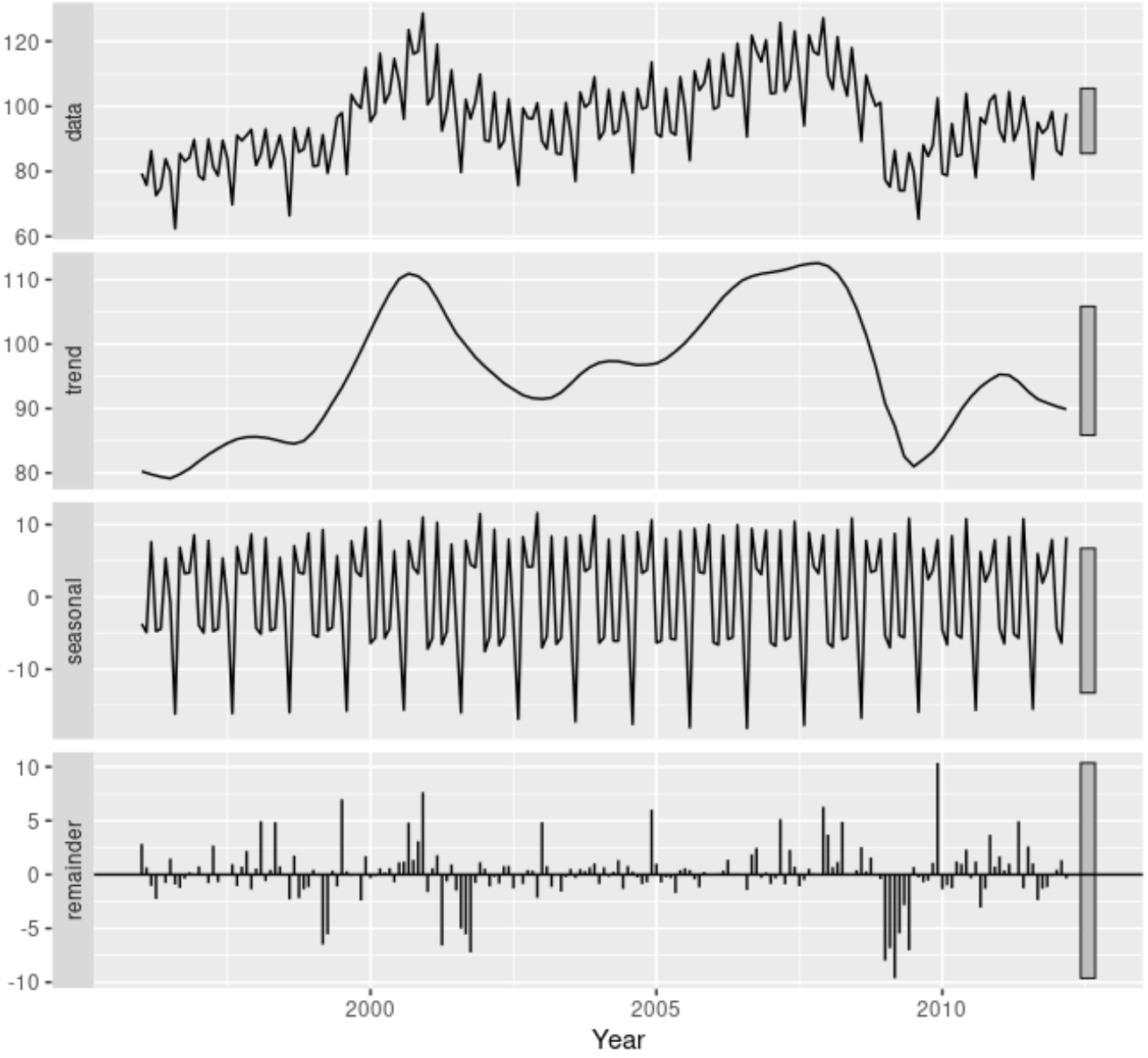
Производство электрооборудования

Данные показывают количество новых заказов на электрооборудование (компьютерная, электронная и оптическая продукция) в зоне евро (16 стран). Данные скорректированы по рабочим дням и нормализованы таким образом, что значение 100 соответствует 2005 году.



Заказы на электрооборудование: компонент тренд-цикл (красный) и исходные данные (серый).

Тренд-цикл показывает общее движение в ряду, игнорируя сезонность и любые небольшие случайные колебания.



На рисунке показано аддитивное разложение этих данных

Стационарность

- Стационарность является ключевой частью анализа временных рядов. Проще говоря, стационарность означает, что способ изменения данных временного ряда является постоянным. Стационарный временной ряд не будет иметь трендов или сезонных моделей. Вы должны проверить стационарность, потому что это не только упрощает моделирование временных рядов, но и является основным предположением во многих методах временных рядов. В частности, стационарность предполагается для широкого спектра методов прогнозирования временных рядов, включая авторегрессионное скользящее среднее (ARMA), ARIMA и сезонное ARIMA (SARIMA).

Автокорреляция

- Проверка автокорреляции в данных временных рядов — еще одна важная часть аналитического процесса. Это мера того, насколько данные временных рядов коррелируют в данный момент времени с прошлыми значениями, что имеет огромное значение для многих отраслей. Например, если наши данные о пассажирах имеют сильную автокорреляцию, мы можем предположить, что высокое количество пассажиров сегодня предполагает высокую вероятность того, что оно будет высоким и завтра.

Декомпозиция

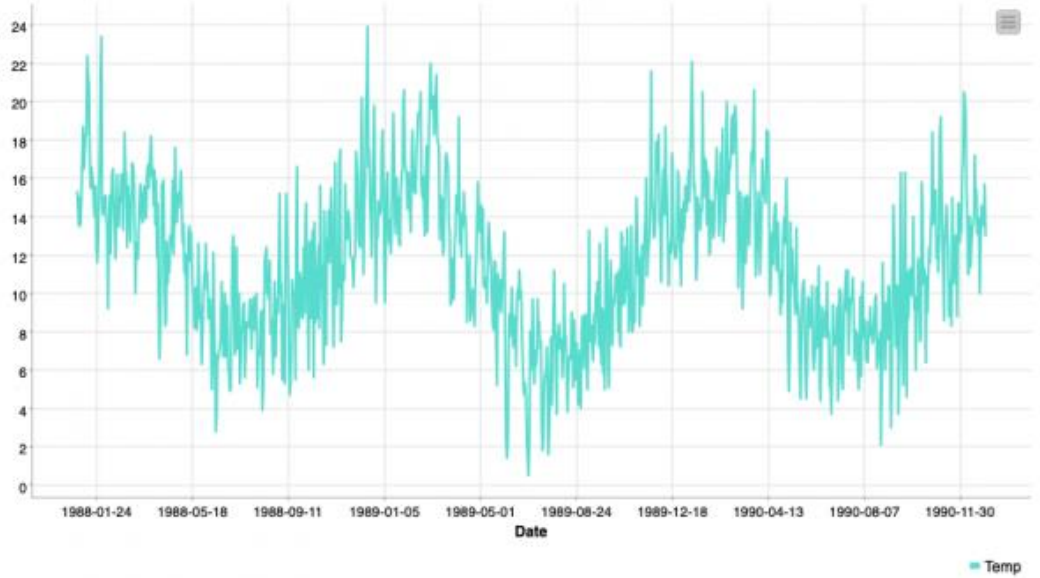
- Декомпозиция тренда — еще один полезный способ визуализации трендов в данных временных рядов.

Прогнозирование

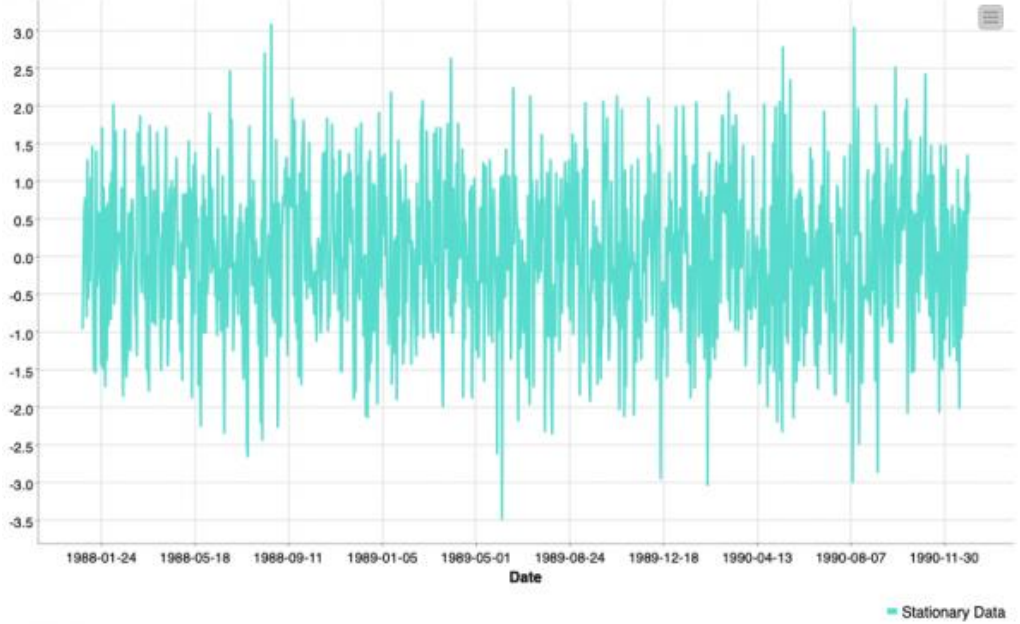
- Прогнозирование временных рядов позволяет прогнозировать будущие значения во временном ряду с учетом текущих и прошлых данных.

- Многие алгоритмы прогнозирования временных рядов требуют, чтобы данные временных рядов были стационарными. Первое и наиболее важное предположение модели ARIMA (и других моделей) заключается в том, что лежащие в ее основе данные являются *стационарными* . В частности, требуется **слабая** стационарность, хотя мы будем просто называть ее стационарностью, как это принято.

- Так что же такое стационарность? Это просто идея о том, что независимо от того, **когда** мы смотрим на данные в нашем временном ряду, они имеют одни и те же «особенности». Когда мы говорим о слабой стационарности, этими функциями являются **среднее значение** и **дисперсия** . Например, в нашем примере данных о температуре для этого потребуется, чтобы средняя температура была одинаковой как в декабре, так и в июле. Если это не так, нужно решить эту проблему.



а) Нестационарные данные, обратите внимание, как среднее колеблется в зависимости от времени года

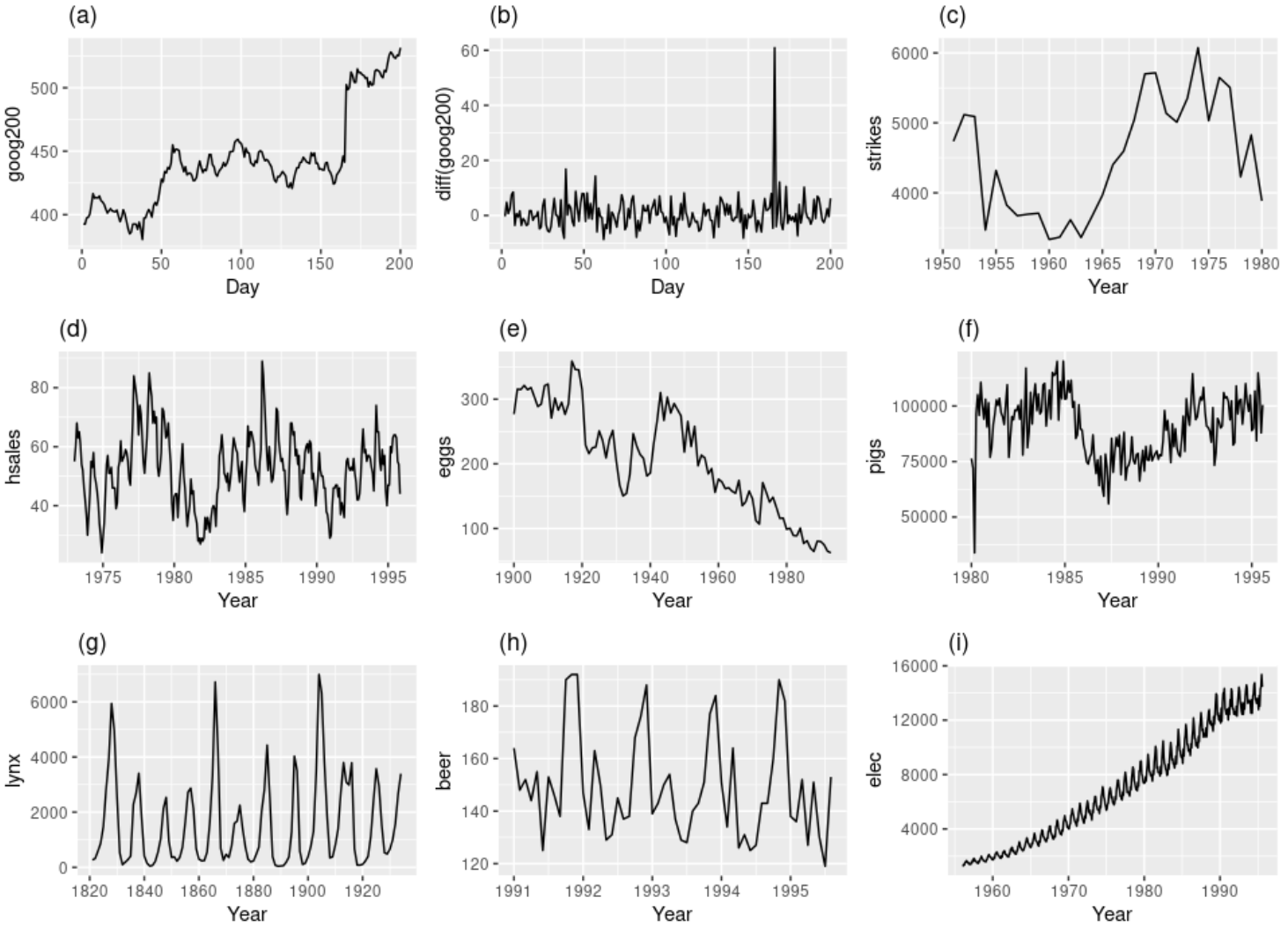


б) Стационарные данные. Обратите внимание, что и среднее значение, и дисперсия кажутся одинаковыми

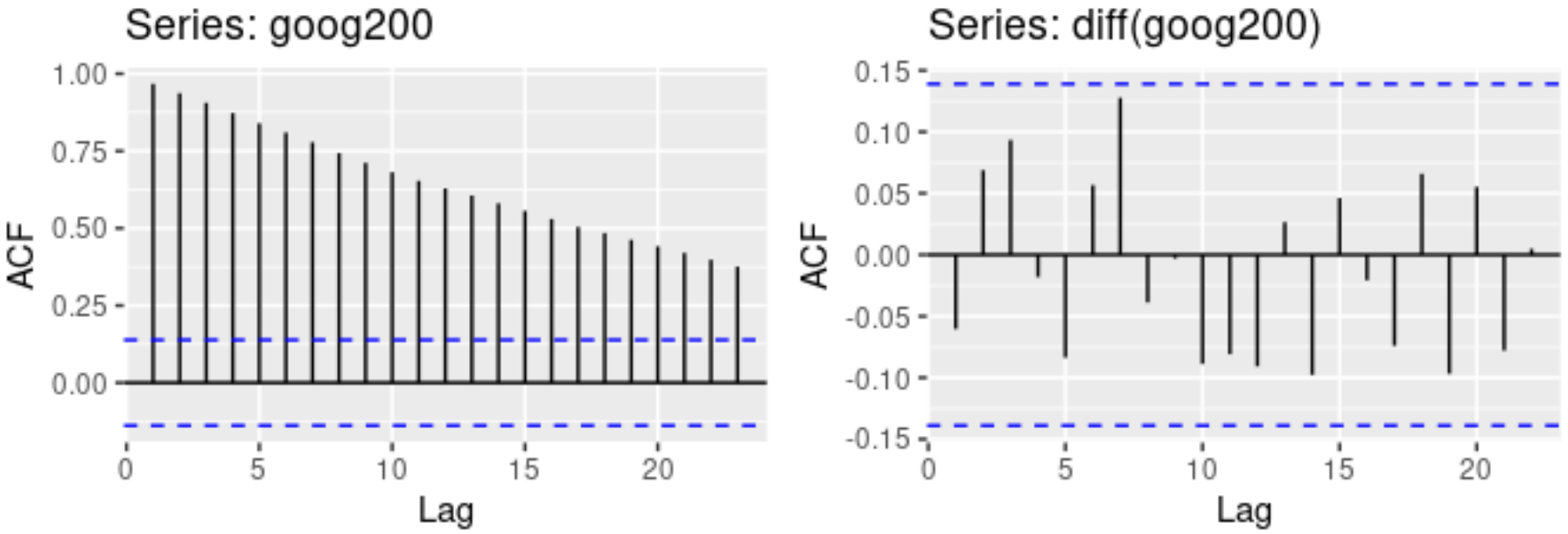
Стационарность

Стационарный временной ряд — это ряд, свойства которого не зависят от времени наблюдения ряда. Таким образом, временные ряды с трендами или с сезонностью не являются стационарными — тренд и сезонность будут влиять на значение временного ряда в разное время. С другой стороны, серия белого шума стационарна — не имеет значения, когда вы ее наблюдаете, она должна выглядеть примерно одинаково в любой момент времени.

- Некоторые случаи могут сбивать с толку — временной ряд с циклическим поведением (но без тренда или сезонности) является стационарным. Это связано с тем, что циклы не имеют фиксированной длины, поэтому, прежде чем мы рассмотрим ряд, мы не можем быть уверены, где будут пики и впадины циклов.
- Как правило, стационарный временной ряд не будет иметь предсказуемых закономерностей в долгосрочной перспективе. Графики времени покажут ряды примерно горизонтальными (хотя возможно некоторое циклическое поведение) с постоянной дисперсией.



Какие из этих рядов являются стационарными? (a) курс акций Google за 200 дней подряд; (b) Ежедневное изменение курса акций Google в течение 200 дней подряд; (c) Ежегодное количество забастовок в США; (d) Ежемесячные продажи новых односемейных домов, проданных в США; (e) годовая цена дюжины яиц в США (доллары в постоянных ценах); (f) Ежемесячное количество свиней, забитых в Виктории, Австралия; g) общее годовое количество рысей, отловленных в районе реки Маккензи на северо-западе Канады; h) ежемесячное производство пива в Австралии; (i) Ежемесячное производство электроэнергии в Австралии.



Дифференциация (Differencing)

Обратите внимание, что на рисунке цена акций Google была нестационарной на панели (а), но ежедневные изменения были стационарными на панели (б). Это показывает один из способов сделать нестационарный временной ряд стационарным — вычислить различия между последовательными наблюдениями. Это известно как дифференцирование .

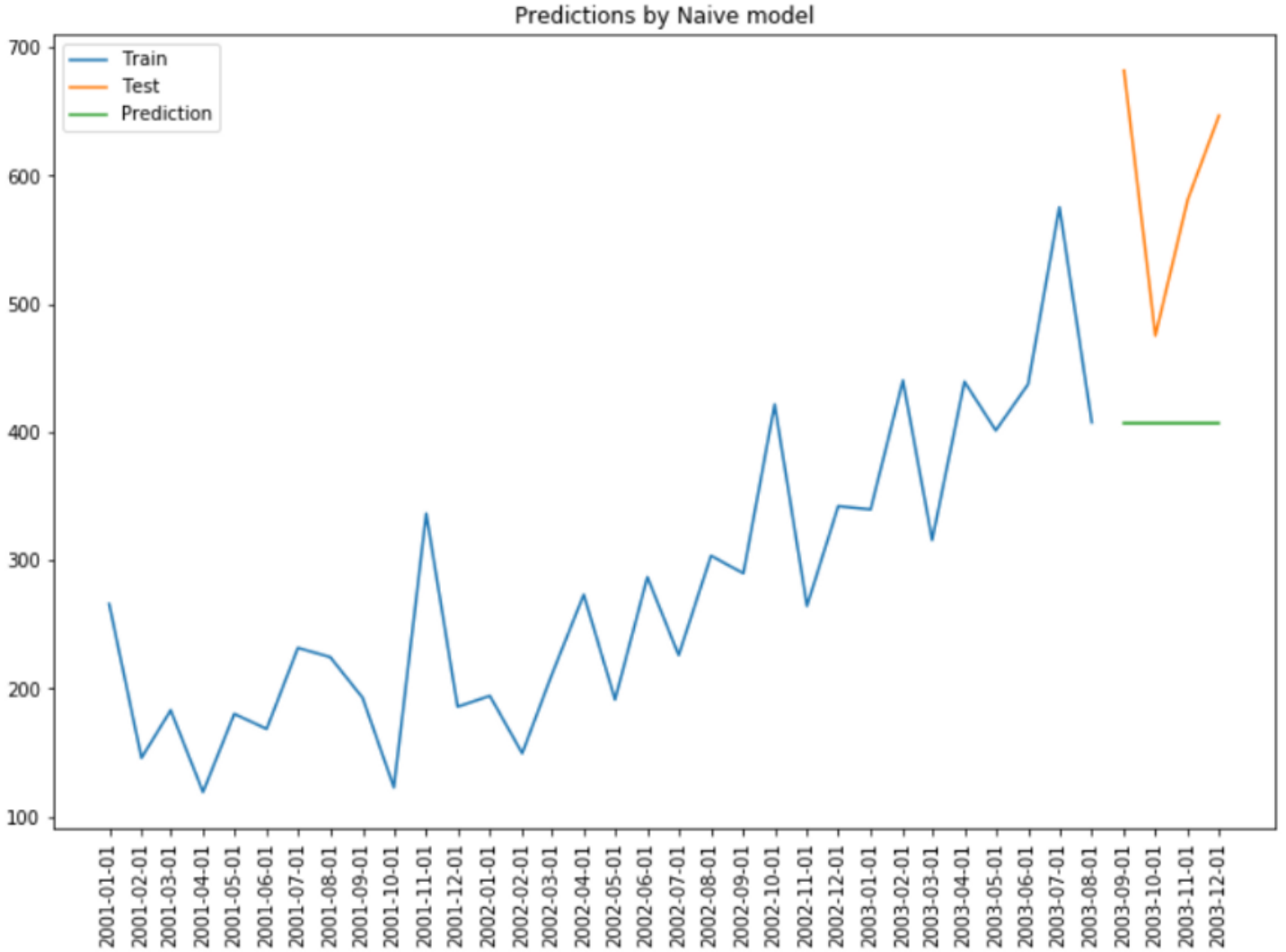
Преобразования, такие как логарифмирование, могут помочь стабилизировать дисперсию временного ряда. **Дифференциация может помочь стабилизировать среднее значение временного ряда, удаляя изменения уровня временного ряда и, следовательно, устраняя (или уменьшая) тренд и сезонность.**

Помимо просмотра временного графика данных, график ACF (функция автокорреляции - при лаге *k* это корреляция между рядами значений, отстоящих друг от друга на *k* интервалов) также полезен для выявления нестационарных временных рядов. Для стационарных временных рядов ACF упадет до нуля относительно быстро, в то время как ACF нестационарных данных убывает медленно. Кроме того, для нестационарных данных значение часто бывает большим и положительным.

Методы прогнозирования временных рядов

1. Наивный подход

- Это один из самых простых способов. В нем говорится, что прогноз на любой период равен последнему наблюдаемому значению. Если данные временного ряда содержат сезонность, лучше брать прогнозы, равные значению прошлого сезона. Это часто используется для целей сравнительного анализа.
- Метод *NaiveDrift()* не требует параметров. Он просто берет первое и последнее значения из ряда и рисует между ними прямую линию тренда.
- Его близкий родственник, *NaiveSeasonal()* , запрашивает параметр сезонности



синий – train, оранжевый – test, зеленый - predict

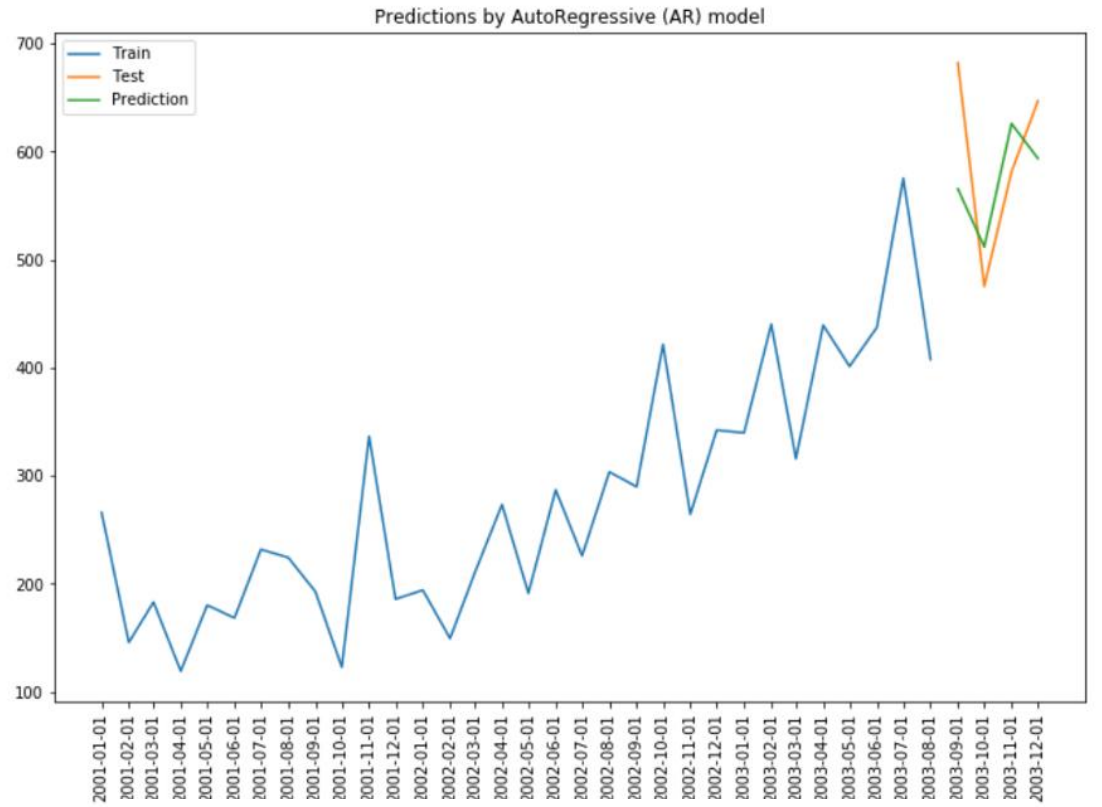
Методы прогнозирования временных рядов

2. AR (авторегрессионная)

- Авторегрессионная модель AR-X(p) следует модели линейной регрессии. Она делает один прогноз за раз и возвращает результат обратно в модель. Здесь *p* указывает порядок модели, например, *AR(1), т.е.* модель авторегрессии первого порядка. Выходная переменная линейно зависит от своих предыдущих значений (называемых лагами или порядками) на предыдущих временных шагах, т.е. регрессия с собственными значениями. Длина лага должна быть указана при создании модели.

```
Coefficients:
intercept   -33.906470
Sales.L1     -0.319220
Sales.L2      0.212906
Sales.L3      0.189463
Sales.L4      0.505078
Sales.L5      0.496669
Sales.L6      0.170001
Sales.L7      0.112577
dtype: float64
```

```
from statsmodels.tsa.ar_model import AutoReg
model_ag = AutoReg(endog = train["Sales"], \
                   lags = 7, \
                   trend='c', \
                   seasonal = False, \
                   exog = None, \
                   hold_back = None, \
                   period = None, \
                   missing = 'none')
# endog: dependent variable, response variable or y (endogenous)
# exog: independent variable, explanatory variable or x (exogenous)
# lags: the no. of lags to include in the model
#       [1, 4] will only include lags 1 and 4
#       while lags=4 will include lags 1, 2, 3, and 4
# trend: trend to include in the model
#        {'n', 'c', 't', 'ct'}
#        'n' - No trend.
#        'c' - Constant only.
#        't' - Time trend only.
#        'ct' - Constant and time trend.
# seasonal: whether to include seasonal dummies in the model
fit_ag = model_ag.fit()
print("Coefficients:\n%s" % fit_ag.params)
```



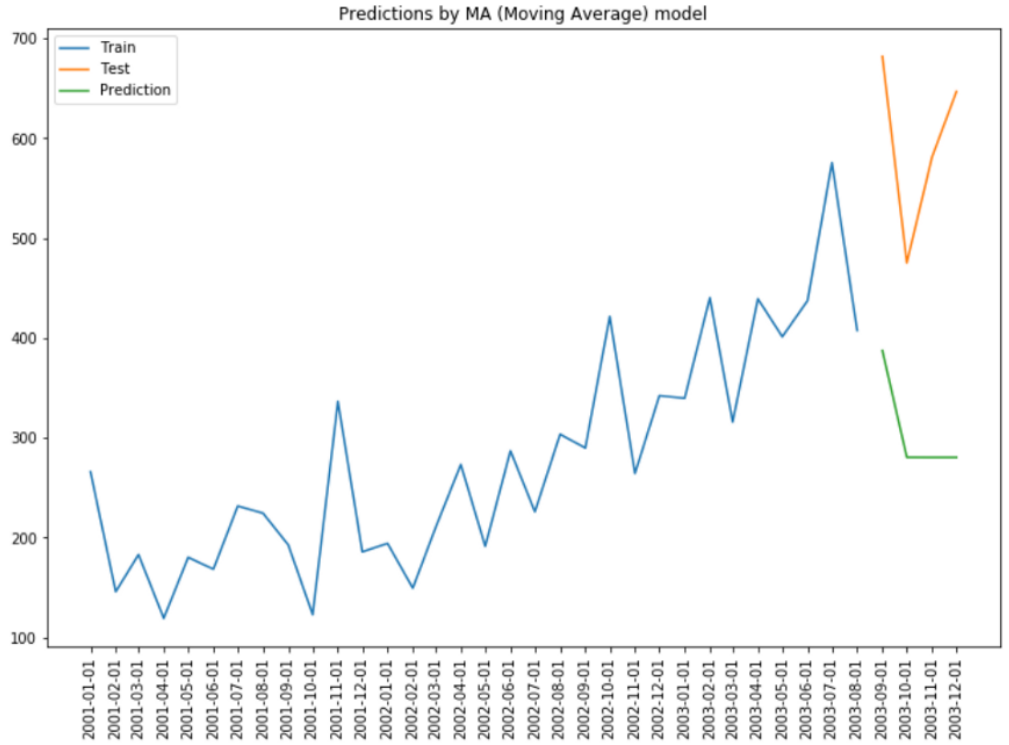
AR - Root Mean Square Error (RMSE): 70.227

3. MA (скользящее среднее)

- Метод скользящего среднего — это подход к моделированию одномерных временных рядов. Это используется для удаления любой сезонной тенденции во временном ряду, чтобы мы могли видеть любую тенденцию в данных. Это представлено как MA(q), где *q* определяет порядок модели, например, *MA(2), т.е.* модель скользящего среднего второго порядка.

```
from statsmodels.tsa.arima.model import ARIMA
model_ma = ARIMA(endog = train["Sales"], \
                 order=(0, 0, 2))
# endog: dependent variable, response variable or y (endogenous)
# order: order of the model for the autoregressive,
#        differences & moving average components.
fit_ma = model_ma.fit()
print("Coefficients:\n%s" % fit_ma.params)
```

```
Coefficients:
const      280.454325
ma.L1        0.435421
ma.L2        0.466753
sigma2    7062.007187
dtype: float64
```



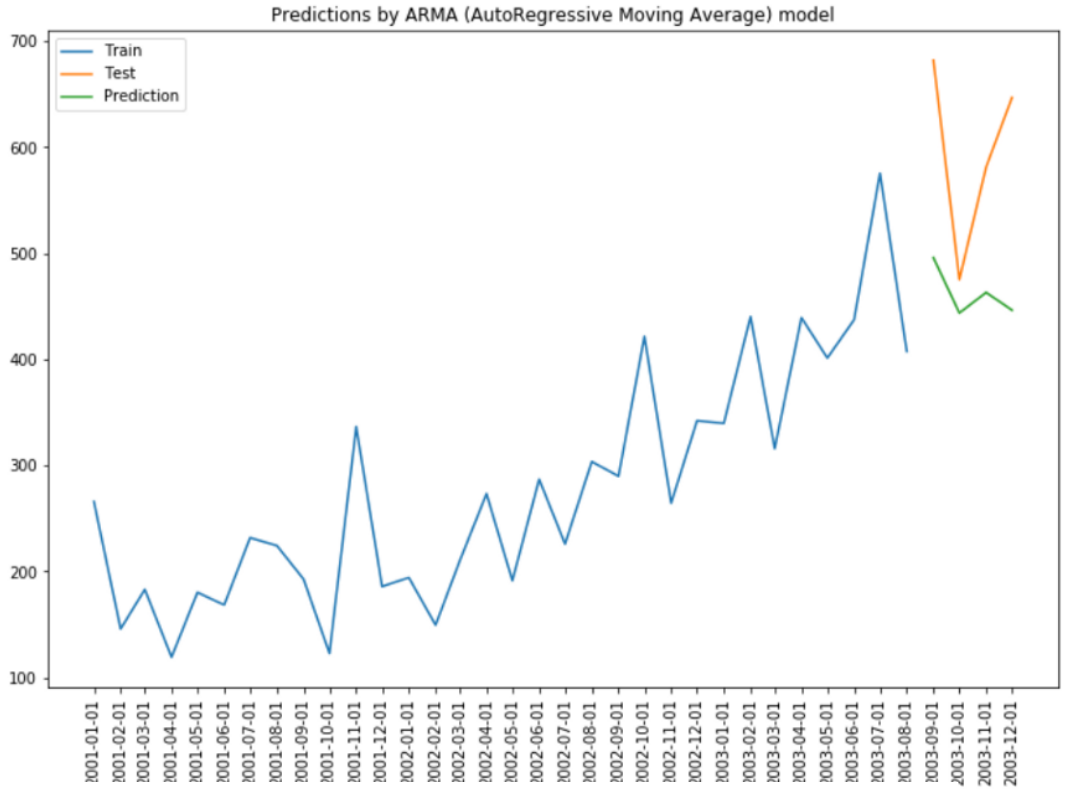
MA - Root Mean Square Error (RMSE): 295.602

4. AutoRegressive Moving Average (ARMA- авторегрессионная скользящая средняя)

Метод ARMA сочетает в себе модели авторегрессии (AR) и скользящего среднего (MA). Обычно это называется моделью *ARMA(p,q)* , где *p* — порядок части AR, а *q* — порядок части MA.

```
from statsmodels.tsa.arima.model import ARIMA
model_arma = ARIMA(endog = train["Sales"], \
                   order = (2, 0, 1))
# endog: dependent variable, response variable or y (endogenous)
# order: order of the model for the autoregressive,
#        differences & moving average components.
fit_arma = model_arma.fit()
print("Coefficients:\n%s" % fit_arma.params)
```

```
Coefficients:
const      316.928647
ar.L1        0.460559
ar.L2        0.490606
ma.L1       -0.321346
sigma2    4644.851441
dtype: float64
```



ARMA - Root Mean Square Error (RMSE): 149.844

Обозначения:

X_t : мы будем использовать X_t для обозначения точки данных в нашем временном ряду в момент времени t, $X_{t\text{-}1}$ будет обозначать предыдущую точку данных.

Y_t : мы будем использовать Y_t для обозначения значения прогноза или прогноза нашей модели в момент времени t, что эквивалентно $Y_{t\text{-}1}$ будет обозначать предыдущий прогноз.

ε_t : мы будем использовать ε_t (эпсилон т) для обозначения ошибки в нашем прогнозе в момент времени t, что эквивалентно $\varepsilon_{t\text{-}1}$ будет обозначать предыдущий член ошибки. Это: $\varepsilon_t = Y_t - X_t$

$\boldsymbol{\mu}$: мы будем использовать $\boldsymbol{\mu}$ для обозначения среднего значения нашего ряда

AR:

Первая часть (S)ARIMA представляет собой авторегрессионную модель. Авторегрессионная модель основана на прошлых (называемых лаговыми) значениях целевого объекта. Используя нашу нотацию выше, это означает, что мы строим модель для прогнозирования X_t , которая выглядит так:

$$Y_m = a_1 \cdot X_{t\text{-}1} + a_2 \cdot X_{t\text{-}2}$$

где a_1, a_2 — параметры, которые мы подбираем во время обучения. Очень похоже на линейную регрессию , верно?!

I: Integrated

- В отличие от частей AR и MA модели ARIMA, I не ссылается на какие-либо дополнительные термины, которые можно добавить в нашу модель. Интегрированный относится к дифференцированию, применяемому к данным до того, как будут применены какие-либо термины авторегрессии или скользящего среднего.

- Так чем же тогда отличается? Различие в этом случае означает, что мы берем ***разницу*** между X_t and X_{t-1} и соответствующим образом обновляем нашу серию

- Теперь вместо :

$$X_t \ , \ X_{t+1} \ , \ X_{t+2} \ldots$$

у нас есть:

$$X_t - X_{t-1} \ , X_{t+1} - X_t \ , X_{t+2} - X_{t+1} \ , \ldots$$

- Это наиболее широко используемый метод для индуцирования стационарности данных временных рядов. Это различие может произойти один или несколько раз. Однократное дифференцирование может помочь удалить линейный тренд, два раза — квадратичный тренд и т.д. Эти различия могут быть автоматически отменены во время прогноза.

MA: скользящая средняя

Третья часть (S)ARIMA — это модель скользящего среднего. Будьте осторожны, чтобы не перепутать этот термин с методом скользящего среднего для сглаживания. Как и в авторегрессионной модели, здесь мы добавляем в регрессию больше терминов. Однако они немного отличаются, вместо моделирования с использованием запаздывающих значений нашей цели мы моделируем прошлые ошибки прогноза.

$$Y_t = b_1 . \varepsilon_{t-1} + b_2 . \varepsilon_{t-2} + \mu$$

где: $\varepsilon_t = Y_t - X_t$

S: Сезонный период

- В базовой версии ARIMA представлено обновление до Seasonal ARIMA, или SARIMA — это простое расширение. С полной моделью ARIMA мы применяем некоторые различия к нашим данным, а затем моделируем с помощью составной модели авторегрессионного скользящего среднего:

$$Y_t = a_1 . X_{t-1} + a_2 . X_{t-2} + b_1 . \varepsilon_{t-1} + b_2 . \varepsilon_{t-2} + \mu$$

SARIMA добавляет дополнительные члены точно так же, как части AR и MA, за исключением того, что вместо добавления t-1, t-2, t-3 и т. д. мы увеличиваем некоторый сезонный параметр. Обычно это могут быть 24 часа, 7 дней или что-то другое, что соответствует шаблону, который вы наблюдаете в своей серии.

Итак мы получаем еще более длинное уравнение, но все же **регрессию**:

$$Y_t = a_1 . X_{t-1} + a_2 . X_{t-2} + b_1 . \varepsilon_{t-1} + b_2 . \varepsilon_{t-2} + A_1 . X_{t-24} + A_2 . X_{t-48} + B_1 . \varepsilon_{m-24} + B_2 . \varepsilon_{m-48} + \boldsymbol{\mu}$$

Гиперпараметры

Как и большинство методов моделирования, SARIMA имеет несколько гиперпараметров, которые описывают специфику модели, которую вы создаете, эти параметры часто называют порядками модели SARIMA. Обычно вы увидите их написанными с названием модели следующим образом: ARIMA(p,d,q) ARIMA(2,1,2). SARIMA(p,d,q)(P,D,Q) SARIMA(2,1,2)(2,0,2)7

Терминами являются **p**, **d** и **q** в стандартном ARIMA и дополнительно **P**, **D** и **Q** в SARIMA:

- p (авторегрессионный порядок): обозначает количество лаговых стоимостных терминов, которые мы использовали в стандартном авторегрессионном компоненте нашей модели.
- d (порядок дифференциации): обозначает, сколько раз мы применяли дифференциацию к нашему ряду до моделирования.
- q (порядок скользящего среднего): обозначает количество прошлых ошибок, которые мы использовали в стандартном компоненте скользящего среднего нашей модели.

Тогда P, D и Q являются сезонными аналогами. Просто количество сезонных терминов или сезонных различий, которые мы применяем к нашим данным (интуитивное расширение).

Обратите внимание, что сезонные различия D можно использовать для учета сезонных закономерностей в наших данных и обеспечения стационарности, необходимой для нашей модели. Подобно тому, как стандартное дифференцирование d может учитывать тенденции в данных.

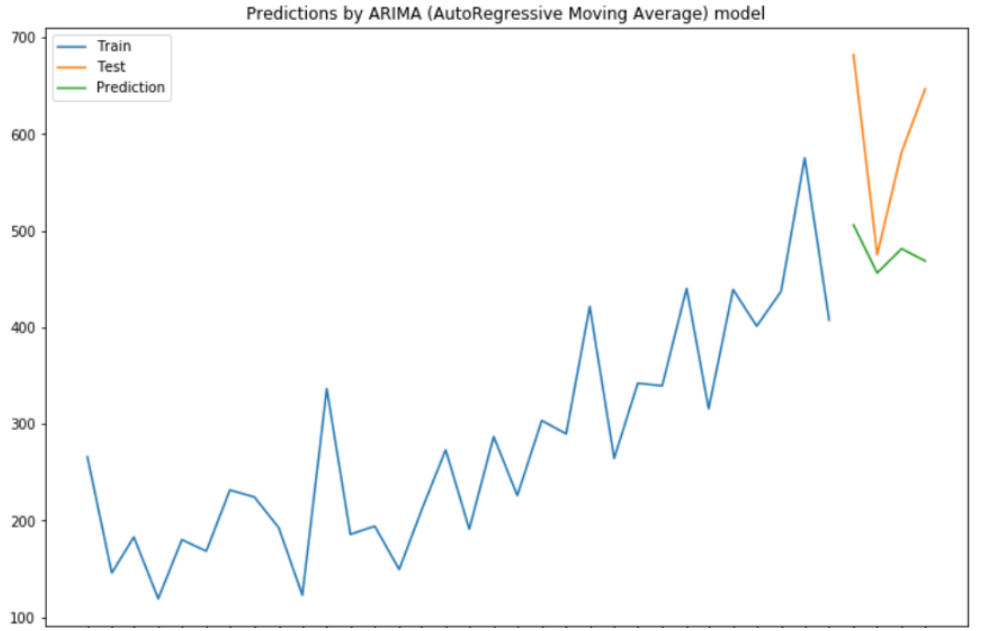
S (сезонный период): это длина сезонного цикла, основанная на количестве точек данных. Например, ежечасные данные сделают дневную сезонность 24. Ежедневные данные сделают недельную сезонность 7. Будьте осторожны, чтобы не использовать очень длинные сезонности, так как они требуют больше данных и времени обучению

5. AutoRegressive Integrated Moving Average (авторегрессионное интегрированное скользящее среднее - ARIMA)

- Модель ARIMA состоит из трех компонентов: (i) Компонент авторегрессии, AR(p), т.е. линейная регрессия по своим предыдущим значениям или лагам (p). (ii) Интегрированный компонент (I) указывает на то, что данные были заменены разницей между текущим наблюдением и предыдущим временным шагом. (iii) Скользящее среднее, MA(q), т.е. рассмотрим скользящее среднее с порядком q.
- Эта модель представлена как *ARIMA(p, d, q),* где *p* , *d* и *q* определяют порядок моделей *AR(p)* , *I(d)* и *MA(q)* соответственно.

```
from statsmodels.tsa.arima.model import ARIMA
model_arima = ARIMA(endog = train["Sales"], \
                    order = (1, 1, 1))
# endog: dependent variable, response variable or y (endogenous)
# order: order of the model for the autoregressive,
#        differences & moving average components.
fit_arima = model_arima.fit()
print("Coefficients:\n%s" % fit_arima.params)
```

```
Coefficients:
ar.L1       -0.504409
ma.L1       -0.355807
sigma2    4691.948058
dtype: float64
```



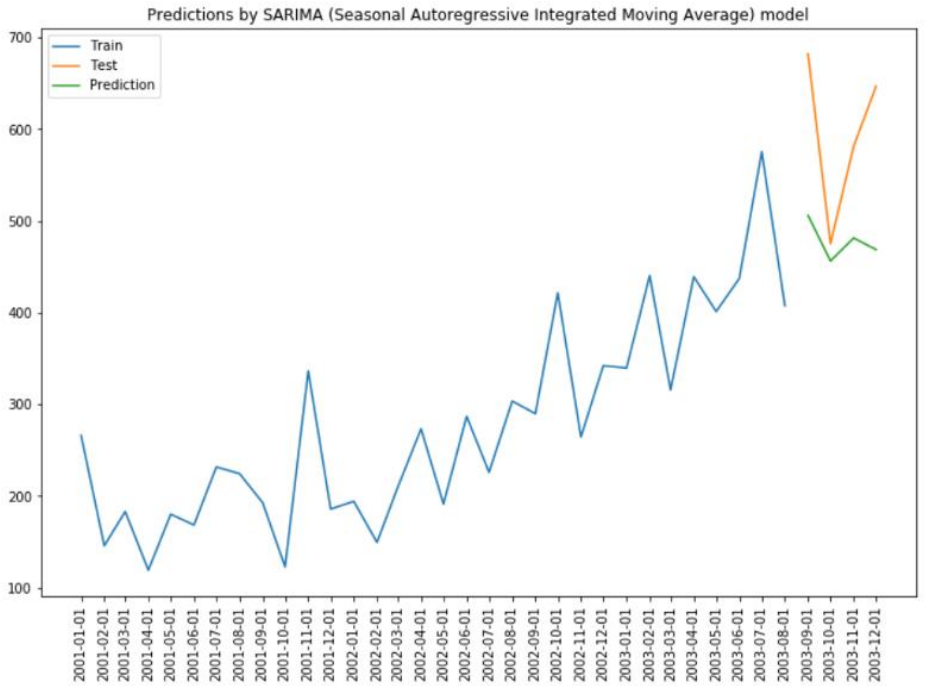
ARIMA - Root Mean Square Error (RMSE): 135.112

6. Seasonal AutoRegressive Integrated Moving Average (SARIMA)

- SARIMA или Seasonal ARIMA, является расширением ARIMA, которое явно поддерживает одномерные данные временных рядов с сезонным компонентом. Она добавляет три новых гиперпараметра для указания авторегрессии (AR), разности (I) и скользящего среднего (MA) для сезонной составляющей ряда, а также дополнительный параметр для периода сезонности.
- Сезонные модели ARIMA обозначаются как *ARIMA(p,d,q)(P,D,Q)m* , где *m* относится к количеству периодов в каждом сезоне, а *P, D, Q* (в верхнем регистре) относятся к авторегрессионному, разностному, и условия скользящего среднего для сезонной части модели ARIMA соответственно.

```
from statsmodels.tsa.statespace.sarimax import SARIMAX
model_sarima = SARIMAX(endog = train["Sales"], \
                       order = (1, 1, 1), \
                       seasonal_order=(0, 0, 0, 0))
# endog: dependent variable, response variable or y (endogenous)
# order: order of the model for the autoregressive,
#        differences & moving average components.
# seasonal_order: (P,D,Q,s) order of the seasonal component of
#     the model for the AR parameters, differences,
#     MA parameters, and periodicity. s is the periodicity
#     (number of periods in season), often it is 4 for
#     quarterly data or 12 for monthly data (default, no
#     seasonal effect).
fit_sarima = model_sarima.fit()
print("Coefficients:\n%s" % fit_sarima.params)
```

```
Coefficients:
ar.L1       -0.504409
ma.L1       -0.355807
sigma2    4691.948058
dtype: float64
```



SARIMA - Root Mean Square Error (RMSE): 135.112

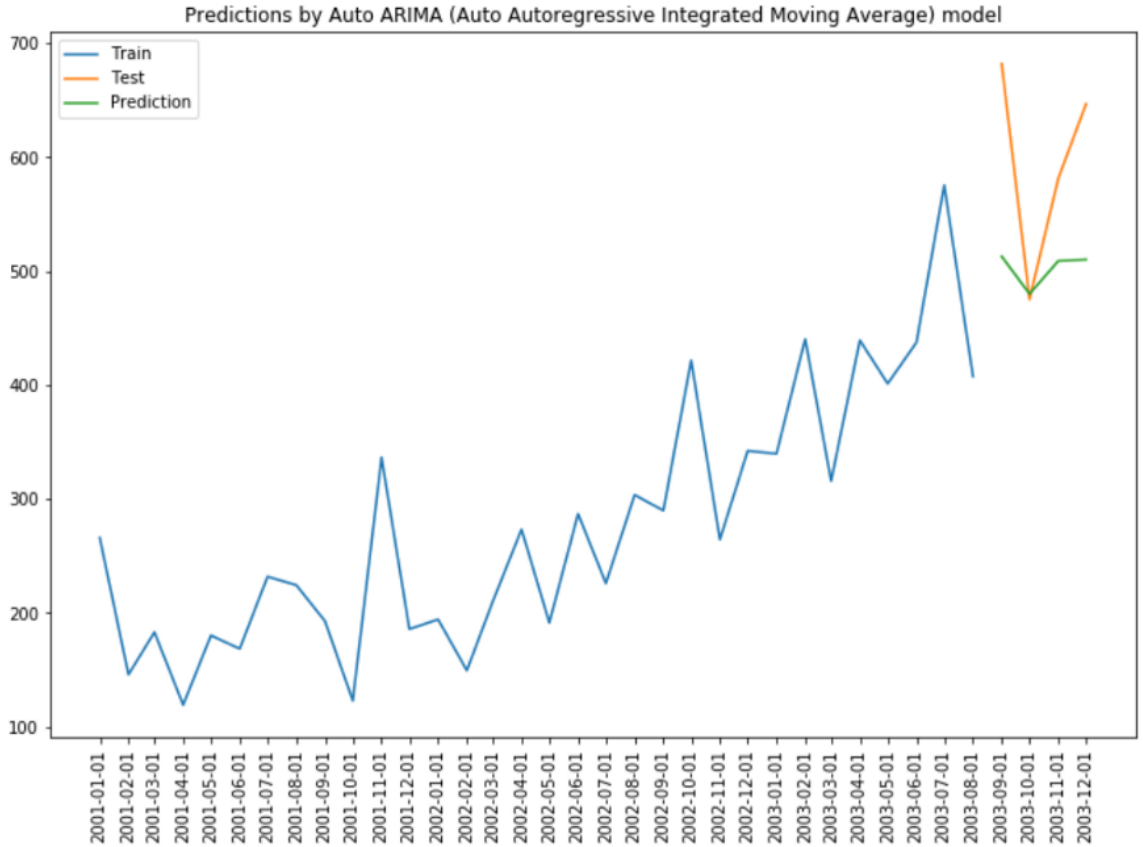
7. Auto ARIMA

- Автоматически обнаруживайте оптимальный порядок для модели ARIMA. Процесс автоматического ARIMA направлен на определение наиболее оптимальных параметров ARIMA модели, останавливаясь на одной подогнанной модели ARIMA. Этот процесс основан на обычно используемой функции R, forecast :: auto.arima

```
from pmdarima.arima import auto_arima
model_aarima = auto_arima (y = train["Sales"], \
                           seasonal=False, \
                           stepwise=True)


# seasonal : default=True, whether to fit
#            a seasonal ARIMA.
# stepwise : default=True, the auto_arima
#            function has two modes: stepwise
#            & parallelized (slower)
```

```
Auto ARIMA - Root Mean Square Error (RMSE): 114.585
```



8. Prophet (пророк)

Prophet — это процедура прогнозирования временных рядов, разработанная Facebook и может применяться в следующих сценариях:

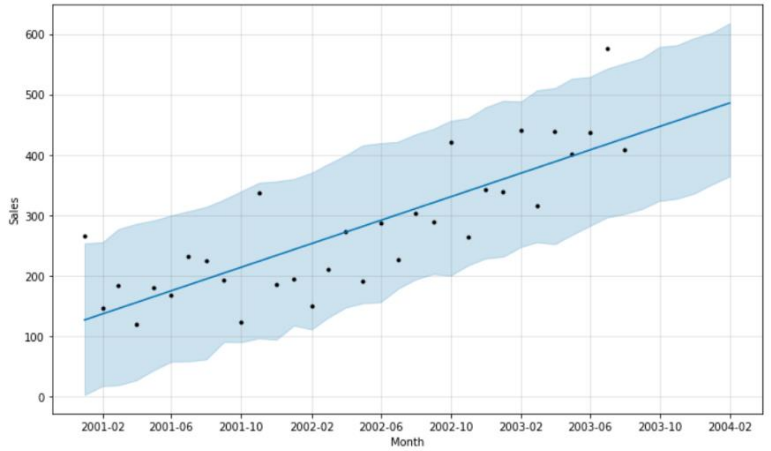
- ежечасные, ежедневные или еженедельные наблюдения как минимум за несколько месяцев (предпочтительно год) в анамнезе
- сильная множественная сезонность в «человеческом масштабе»: день недели и время года
- важные праздники, которые происходят через нерегулярные промежутки времени, о которых известно заранее (например, Суперкубок)
- разумное количество пропущенных наблюдений или больших выбросов
- исторические изменения тенденций, например, из-за запуска продукта или регистрации изменений
- тренды, представляющие собой нелинейные кривые роста, когда тренд достигает естественного предела или достигает насыщения

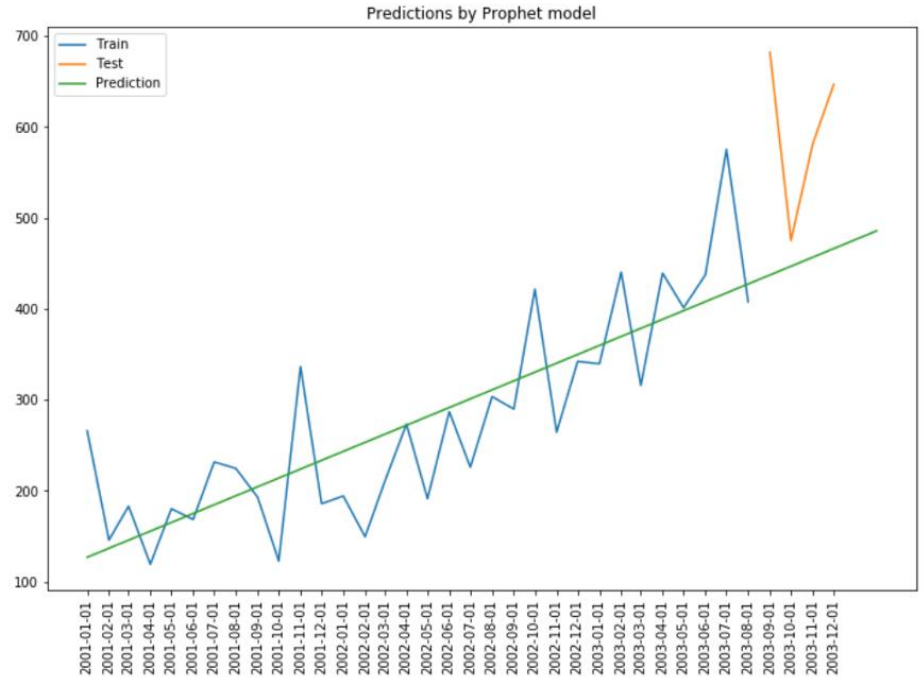
Входные данные для Prophet всегда представляют dataframe с двумя столбцами: *ds* (отметка даты, формат ГГГГ-ММ-ДД или ГГГГ-ММ-ДД ЧЧ:ММ:СС) и *y* (числовой, представляет измерение, которое мы хотим спрогнозировать).

```
from fbprophet import Prophet
# instantiate the model and set parameters
model_fb = Prophet( \
                    interval_width = 0.95, \
                    growth = "linear", \
                    daily_seasonality = False, \
                    weekly_seasonality = False, \
                    yearly_seasonality = False, \
                    seasonality_mode = "multiplicative"
)


train_fb = train.copy()
train_fb.columns = ["y", "ds"]


# fit the model to historical data
model_fb.fit(train_fb)
```

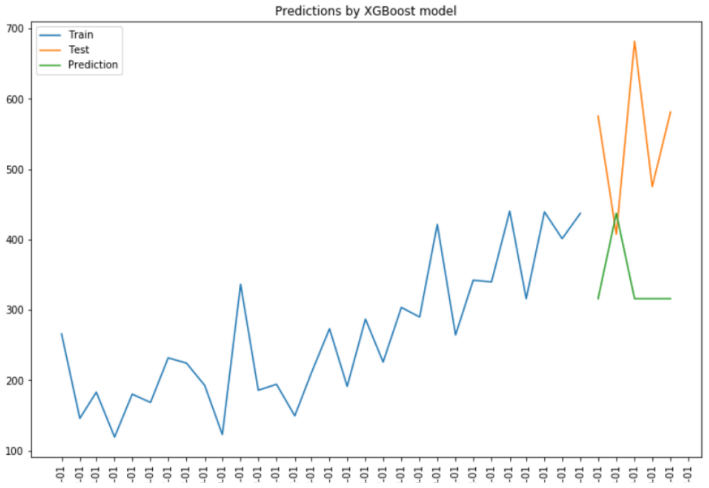




9. XGBoost

- XGBoost (E**x**treme **G**radient **Boosting**) — это реализация градиентного бустинга для задач классификации и регрессии. XGBoost можно использовать для прогнозирования временных рядов путем реструктуризации набора входных данных, чтобы она выглядела как задача обучения с учителем.

	Sales	Month	Target
1	266	2001-01-01	145.9
2	145.9	2001-02-01	183.1
3	183.1	2001-03-01	119.3
4	119.3	2001-04-01	180.3
5	180.3	2001-05-01	168.5
6	168.5	2001-06-01	231.8
7	231.8	2001-07-01	224.5
8	224.5	2001-08-01	192.8




```
dataXGB = dataset.copy()
# Restructure the data
dataXGB["Target"] = dataXGB.Sales.shift(-1)


# Drop the last null column because of shifting
dataXGB.dropna(inplace=True)


# Extract features & labels
X = dataXGB.iloc[:,0:1].values
y = dataXGB.iloc[:, -1].values


from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = \
      train_test_split(X, y, test_size = 0.12, \
                       random_state = 0, shuffle=False)


import xgboost
reg = xgboost.XGBRegressor(objective='reg:squarederror', \
                           n_estimators=1000)


reg.fit(X_train, y_train)
```

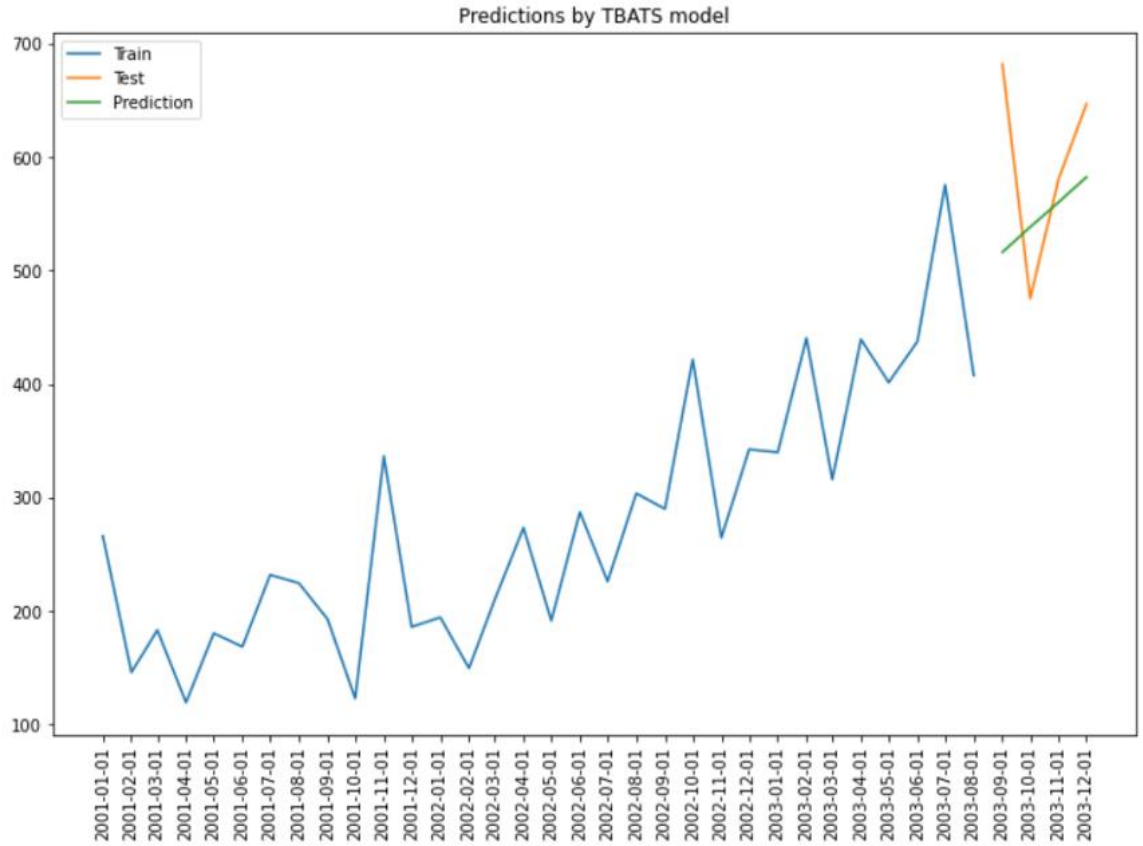
```
XGBoost - Root Mean Square Error (RMSE): 233.942
```

10. TBATS

- BATS и TBATS — это алгоритмы прогнозирования временных рядов, которые хорошо работают с несколькими сезонными периодами. TBATS предпочтительнее, когда сезонность сложна. ТБАТ — это аббревиатура от:
- **Тригонометрическая** сезонность
- **Преобразование** Бокса-Кокса (для приближения закона распределения к нормальному)
- **ARMA error**
- **Trend (Т**ренд)
- **Seasonal components (Сезонные** компоненты)

```
from tbats import TBATS
# /databricks/python/bin/pip install tbats==1.1.0


model_tbats = TBATS(seasonal_periods=(12, 28),\
              use_arma_errors=False,\
              use_box_cox=False,\
              n_jobs=1,\
              use_trend=None,\
              use_damped_trend=None)\
          .fit(train.Sales)
```




```
TBATS - Root Mean Square Error (RMSE): 94.925
```

11. ETS

- ETS (**E** rror, **T** rend , **S** easonality) представляют собой модели пространства состояний экспоненциального сглаживания (общее семейство моделей прогнозирования) для одномерного анализа временных рядов. В отличие от простого скользящего среднего, где прошлые наблюдения имеют одинаковый вес, экспоненциальные функции используются в моделях ETS для присвоения экспоненциально уменьшающихся весов с течением времени.
- Прогнозы, полученные с использованием методов экспоненциального сглаживания, представляют собой средневзвешенные значения прошлых наблюдений, при этом веса экспоненциально уменьшаются по мере старения наблюдений. Другими словами, чем позднее наблюдение, тем выше соответствующий вес. Эта структура быстро генерирует надежные прогнозы для широкого диапазона временных рядов, что является большим преимуществом и имеет большое значение для приложений в промышленности.
- *statsmodels* реализует все комбинации аддитивной и мультипликативной модели ошибок, аддитивной и мультипликативной модели тренда, возможно ослабленной — аддитивной и мультипликативной сезонности. Здесь мы будем использовать сезонный метод Холта-Уинтерса, который может включать тренд и сезонный компонент.

Помимо вышеперечисленного, есть еще несколько алгоритмов, которые мы можем использовать:

- ARFIMA: Авторегрессионное частично интегрированное скользящее среднее;
- LSTM (долговременная краткосрочная память): Tensorflow также можно использовать для прогнозирования временных рядов с использованием сверточных и рекуррентных нейронных сетей (CNN и RNN).
- NNETAR: прогнозы временных рядов нейронной сети, нейронные сети прямой связи с одним скрытым слоем и запаздывающими входными данными для прогнозирования одномерных временных рядов.
- RWF_Drift: эквивалентно модели ARIMA(0,1,0) с дополнительным коэффициентом дрейфа.
- SplineF: возвращает локальные линейные прогнозы и интервалы прогнозирования с использованием сплайнов кубического сглаживания.
- STL (разложение сезонного тренда с использованием LOESS): используется для разложения данных временных рядов и выделения влияния сезонности и тренда. Это можно использовать с другими моделями, такими как ARIMA или ETS.
- ThetaF: прогноз методом тета.
- TSLM: используется для подгонки линейных моделей к временным рядам, включая компоненты тренда и сезонности.

Итак, хотя данные с компонентами времени очень распространены, часто бывает сложно обрабатывать эти компоненты и анализировать закономерность. Как только закономерность будет понята, мы сможем использовать несколько методов для предсказания будущего и выбрать окончательное прогнозирование на основе наименьшей ошибки прогнозирования.