**Структура модуля:**

[**Юнит 10. Введение** 2](#_Toc148017724)

[**Юнит 11. Модель ARMA (общий подход)** 3](#_Toc148017725)

[**Юнит 12 – о выборе начальных значений пордяков модели. Юнит Опциональный** 6](#_Toc148017726)

[Юнит 13 - Оптимизация ARMA моделей 9](#_Toc148017727)

[Юнит 14. Модели ARIMA 15](#_Toc148017729)

[Юнит 15. Модель SARIMA 20](#_Toc148017732)

[**~~Юнит 15.1 Алгоритм SARIMA модели~~** 24](#_Toc148017733)

[Юнит 16 – Другие модели предсказания. 33](#_Toc148017734)

[Юнит 17. SKTime\_ARIMA 39](#_Toc148017737)

[**Использование моделей АРСС** 39](#_Toc148017738)

[**SARIMA Forecasting** 42](#_Toc148017739)

[Словарь терминов и определений 69](#_Toc148017741)

[Заключение 71](#_Toc148017742)

# **Юнит 10. Введение**

При решении задач анализа и предсказания временных рядов обычно сначала рассматривается применение простых алгоритмов машинного обучения (например, линейной регрессии), так как они менее требовательны к вычислительным ресурсам и для определенного спектра задач позволяют достичь значения показателя точности не хуже, чем при использовании ансамблей, трансформеров и других сложных моделей. Данный модуль познакомит вас с ними.

**В этом модуле вы изучите следующие вопросы:**

* базовые методы предварительного анализа ВР,
* типы решаемых задач анализа ВР,
* модели ВР,
* основные статистические свойства ВР,
* Понятие о особых свойствах ВР, например,:
  + АКФ,
  + Гетеросекдостичность
  + Белый Гауссов Шум,
  + особенности стационарного вида ВР.
* Виды простых предсказаний ВР, в т.ч.:
  + Наивные методы предсказаний,
  + Методы на основе скользящего среднего и экспоненциального сглаживания,
  + Особенности использования методов линейной регрессии в анализе ВР,
  + Особенности использования методов нелинейной регрессии в анализе ВР,
  + Особенности базовых методов разложения ВР.

Также вы познакомитесь с **методами работы с ВР в рамках языка программирования Python**, включая особенности:

* визуализации ВР,
* определения базовых свойств ВР,
* моделирования ВР,
* работы с фреймворком SKTime,
* выбора методов предсказания ВР.

**В результате прохождения модуля вы будете:**

* знать основные понятия анализ временных рядов, основные свойства ВР;
* уметь предложить подход для решения той или иной задачи анализа временных рядов;
* владеть методами предварительного анализа временных рядов.

Умение работать с временными рядами является отличным навыком, который откроет перед вами множество дверей: финансовая отрасль, маркетинг и продажи, анализ производственных показателей и другие сферы.

# **Юнит 11. Модель ARMA (общий подход)**

В данном юните мы рассмотрим модель авторегрессии скользящего среднего. Подход к её построению был предложен в 1970-х годах Джорджем Боксом и Гвилимом Дженкинсом. Она предназначена для анализа стационарных временных рядов на основе оценки линейной зависимости прогнозируемых значений от исторических. Начнем с того, что такое авторегрессия.

Авторегрессия — это составляющая модели временного ряда, в которой его прогнозируемое значение может быть выражено в виде линейной комбинации значений этого же ряда и случайной ошибки.

Рассмотрим уравнение взвешенного скользящего сглаживания (**Авторегрессия, АR**) как

где – это оценка (предсказанное значение); – набор наблюдений, например значения энергопотребления в заданные моменты времени; – коэффициенты при наблюдениях; – порядок модели авторегрессии. В данной модели порядок – как некоторый гиперпараметр, а набор коэффициентов оценивается.

Данная модель строится на предположении что будущие значения (или их оценки) являются некоторой линейной комбинацией ее предыдущих значений. Данное предположение можно обосновать некоторой инертностью процессов, лежащих в основе наших наблюдений.

Также переменная может быть также представлена как набор процессов, подобных случайному блужданию при помощи **скользящего среднего, ​​MA** как

где – последовательность случайных возмущений;– набор коэффициентов при значениях случайных блужданий; – порядок модели случайного среднего. Часто .

Описание временного ряда скользящим следует из теоремы Вольда. Эта теорема описывает недетерминированный стационарный в широком смысле процесс (почти стационарный) может быть представлен как стохастический процесс *МА*(q), в предельном случае бесконечного порядка.

где «почти стационарный» значит стационарный процесс плюс простой тренд, а в предельном случае «» будет бесконечным,

– постоянная (дрейф, глобальный тренд), тогда – стационарный ряд; как правило белый шум. Как следствие, пусть q – конечно, тогда каким будет остаток вида Попытаемся объединить процесс MA, но ограниченного порядка и AR ограниченного порядка.

Получаем процесс Авторегрессии-Скользящего среднего (ARCC / **ARMA**):

где и - весовые коэффициенты модели, а и порядки АРМА модели. В некоторой интерпретации можно сказать, что, что из модели следует, что **​​MA** это остаток, не объясненный авторегрессией (или наоборот):. Другими словами, процедура также можно записать **ARMA как**

И так, модель **авторегрессии - скользящего среднего (АРСС, ARMA) –** это модель которую можно проинтерпретировать как: текущее значение ВР зависит от прошлых значений до лага (**AR часть**) и от текущего и прошлых внешних «возмущений» (флуктуаций) до лага (**MA часть**). То есть АРМА - Это параметрическая модель c двумя параметрами ().

Задача **ARMA**-аппроксимации (построения АРСС модели) - найти весовые коэффициенты и и оценить их порядки (), приближающие лучше всего к исходному процессу . На самом деле эту задачу можно решить двумя способами. Можно искать значения коэффициентов и как решения некоторой общей задачи оптимизации. Это наиболее популярный способ. Но можно также сказать, что мы найдем сначала аппроксимацию ВР как значений авторегрессии (коэффициенты . Затем мы можем построить **АР** аппроксимацию **ВР** и ошибку этой аппроксимации. Эту ошибку мы будем считать некоторыми возмущениями относительно модели **АР**. Но тогда мы можем сказать, что это и есть модель **МА**. То есть аппроксимация остатка от **АР** разложения даст коэффициенты **МА** модели (, которые мы используем в общей модели.

Повторим еще раз. Задача ARMA-аппроксимации (построения АРСС модели) - найти весовые коэффициенты и и оценить их порядки (), приближающие лучше всего к исходному процессу . При этом порядки модели оцениваются вне процедуры оценки коэффициентов, то есть это гипепараметры. Поэтому модель называется параметрическая. Четких алгоритмов оценки параметров нет. Их нужно выбирать экспертно. Для этого мы будем выбирать начальные порядки модели, а потом их корректировать.

# **Юнит 12 – о выборе начальных значений пордяков модели. Юнит Опциональный**

**Предупреждение следующие абзацы про начальные значения порядков ARMA – Это Опциональный материал, если не хотите, не разбирайтесь.**

**Начальные значения порядков ARMA моделей** могут быть выбраны из анализа графиков выборочной автокорреляционной функции (АКФ, ACF) и частичной АКФ (ЧАКФ, PACF). Напомним, что автокорреляционная функция показывает связанность выборки ВР с его сдвинутыми копиями. То есть если мы возьмем копию ВР и будем сдвигать ее и каждый раз считать корреляцию, то мы получим АКФ. Значения запаздываний, при которых мы считаем АКФ мы называем лагами или временными задержками.

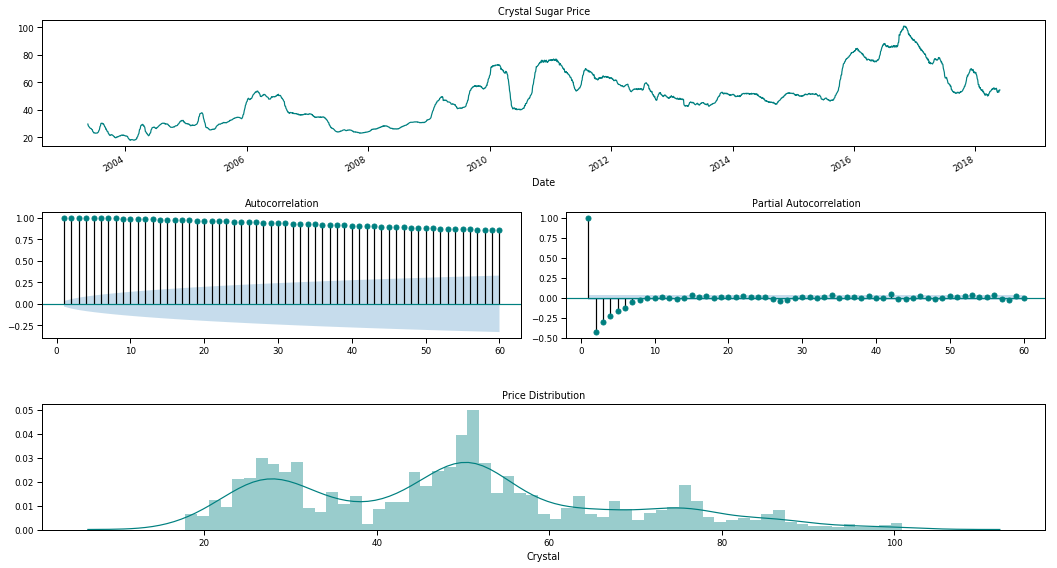
Напомним, что фактически АКФ показывает нам зависимость ВР от своих задержанных копий. Для стационарного ВР мы ожидаем что такая зависимость быстро пойдет на спад. Если этого не происходит, то мы можем сказать, что ВР не стационарен.

Причина, по которой мы ожидаем такой резкий спад заключается в следующем. Для стационарного временного ряда не важно с какого момента взять выборку заданной длины. В любом случае ее статистические моменты будут одинаковы. Тогда можно сказать что АКФ зависит не от момента взятия выборки, а только от лага. А чем дальше удален один ВР от другого тем интуитивно меньше они коррелируют. Отмечается, что такой спад должен быть экспоненциальным. Для нестационарного временного ряда выборки, взятые в каждый момент времени будут разными и соответственно поведения АКФ будет не предсказуемым.

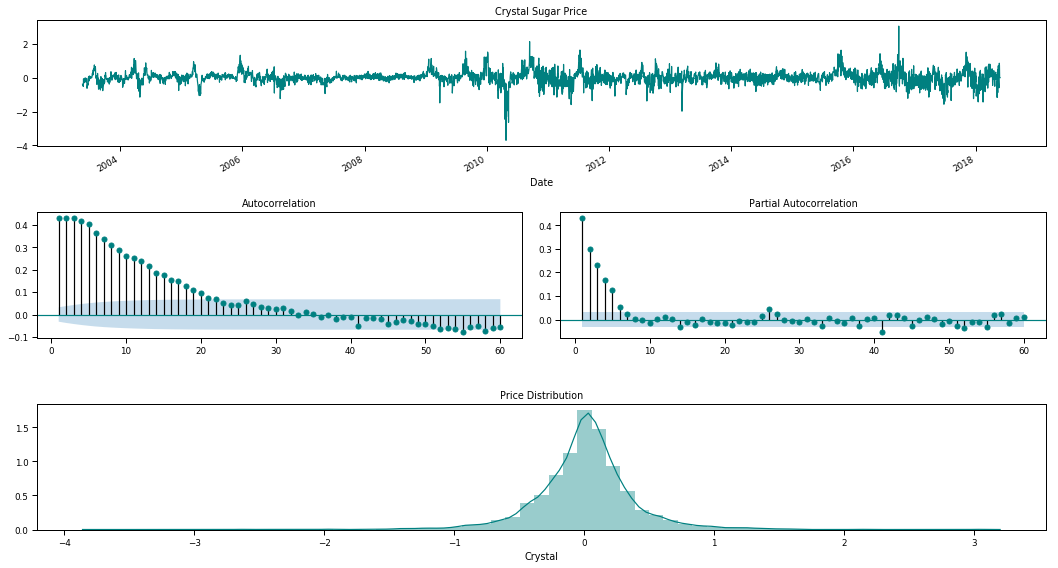
Если АКФ включается все задержанные копии, то частичная АКФ (ЧАКФ) показывает зависимость для каждого лага по отдельности без учета промежуточных лагов.

Ожидается, что для стационарного процесса ЧАКФ должна быть равна нулю для всех лагов выше действительного порядка авторегрессии. Другими словами, по ЧАКФ можно определить порядок авторегрессии как номер последнего не нулевого лага. Отметим, что порядок скользящего среднего можно аналогично определить как как последний лаг АКФ перед резким спадом (в силу экспоненциального спада лаги АКФ не будут полностью нулевыми).

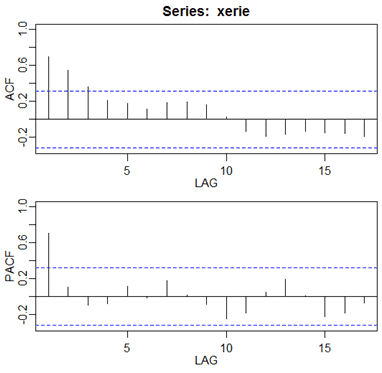
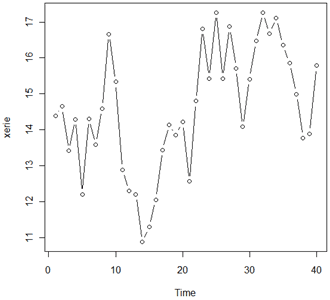
Также отметим, что в силу шумов значения лагов АКФ и ЧАКФ никогда не будут нулевыми. В определенный момент они начнут случайно себя вести (флуктуировать) в районе нуля. Тогда тестирование Лагов АКФ на значимость выполняется при помощи статистических тестов. На графиках уровень выше которого значения лагов точно не порождены шумом (уровень значимости) обычно показывается. Ниже этого уровня значения лагов считаются случайными.



Пример нестационарного временного ряда, АКФ у него не затухает. Также не стационарность видна на самом графике – это как минимум тренд процесса.



Пример процесса, который можно считать стационарным. Этот процесс получен как численная производная временного ряда на графике выше. Для данного процесса мы видим Резкий, хоть и не экспоненциальный спад АКФ и резкий спад ЧАКФ.



Пример процесса сгенерированного как ARMA(1,2). Для этого процесса мы видим один лаг ЧАКФ (PACF) выше уровня значимости (синяя линия) – значит порядок AR первый и видим два лага АКФ выше уровня значимости – значит порядок МА второй.

Как правило, эту задачу нахождения коэффициентов модели при заданных порядках можно решить регрессионными методами. Для этого необходимо выбрать часть ряда для аппроксимации (тренировочную выборку) и проверить результат (тестовая выборка). Если мы знаем коэффициенты ARMA, мы можем предсказать будущие значения с использованием уравнения ARMA.

Отметим, что указанные правила выбора начальных порядков в той или иной мере эвристические. То есть выбор таких значений не гарантирует что модель будет лучшей. После выбора таких начальных значений порядков их можно поварьировать чтобы понять какая модель лучше. Также отметим важное эвристическое правило. Чем меньше суммарный порядок модели (чем меньше параметров), тем меньше вероятность переобучения модели! То есть тем больше обобщающая способоность.

# Юнит 13 - Оптимизация ARMA моделей

Обучение модели ARMA состоит из нескольких частей. А именно: выбор начальных значений порядков модели (), определение коэффициентов модели, проверка модели на точность и варьирование порядков модели с целью выбора лучших для имеющегося временного ряда. Другими словами, при аппроксимации ВР моделью ARMA следует проводить процедуру оптимизации гиперпараметров (порядков модели). Эту процедуру назовем Оптимизация ARMA моделей.

**Оптимизация** **ARMA моделей** осуществляется путем выбора порядков модели при помощи информационных критериев и последующего тестирования модели. При этом напомним, что чем ниже общий порядок модели – тем ниже вероятность переобучения.

**Информационный критерий** – критерий, учитывающий, как среднеквадратичную ошибку ее работы (дисперсию остатка), так и порядки модели. Критерий имеет исключительно смысл в ранжировании (используется только для сравнения моделей, чем меньше значение, тем лучше). Критерий позволяет задать мини-максуную задачу поиска параметров моделей: минимум ошибки при максимально-допустимом числе параметров модели.

Отметим следующее. Чем выше порядки модели, тем ожидаемо точность модели на тренировочной выборке будет больше (можно более точно объяснить поведение ВР). Однако, в поведение модели могут быть ошибки – шумы, то есть нерегулярные события. Поэтому мы хотели бы достичь большей точности при как можно меньшем числе параметров. Поэтому мы «как бы ищем баланс точность – число параметров». То есть нужен компромисс между точностью и сложностью моделей. Таким образом Информационный критерии используются для выбора модели (сравнения между статистическими моделями). Чем ниже значение критериев – тем лучше. Модель «штрафуется» за слишком большое число параметров. Также важно понимать абсолютные значение критерия не имеет смысла.

Как правило при этом используют Информационный критерий Акайке

где оценка правдоподобия модели; – корень суммы квадратов ошибки, и число параметров, - объем выборки. Для ARMA модели или с константой .

Отметим, что на самом деле формула справедлива только если остаток – белый шум. *Для других случаев формулы будут иметь другой вид! Общий вид критерия : , где*  -логарифм функции правдоподобия.

На самом деле есть несколько популярных критериев:

* Информационный критерий Акайке, он Для ARMA процесса может несколько завышать число параметров.
* Байесовский Информационный критерий, он штрафует сильнее, чем AIC (оценка снизу), но работает для больших выборок.
* , модифицированный Информационный критерий Акайке. Он рекомендуется при небольших выборках.

Отметим что то, что такое “Большая выборка” - определяется по теореме Котельникова. То есть можно искусственно сделать выборку большой, но от этого она больше информации содержать не будет. Можно децимировать (проредить) выборку, но только если теорема Котельникова позволяет. Небольшая выборка в этом смысле становится той, в которой присутствует от 1 до нескольких периодов основной сезонности.

И так, Алгоритм оптимизации ARMA модели

1. Проверка, что модель стационарна. Модель работает для слабо-стационарного ряда (без тренда). Напомним, что чем менее стационарен ряд, тем меньше горизонт точного прогнозирования. То есть тем ниже обобщающая способность.

3.Выбор тренировочной и валидационной выборки.

4.Выбор изначальных параметров модели (например, по АКФ и ЧАКФ или как ).

5.Варьирование параметров моделей с целью минимизации BIC или AICc критериев.

6.Проверка остатков для модели (на предмет стационарного БГШ).

7.Валидация, проверка точности по известным методикам.

Отметим наше замечание про горизонт прогнозирования. Небольшим горизонтом прогноза является длительность, которая соответствует размеру выбору от долей значащего периода сезонности до единиц таких периодов. Другими словами, это горизонт, который мог бы легко достроить глаз человека. Если во ВР нет явной сезонности, то величина горизонта была бы связана со скоростью изменения тренда ВР.

Позже мы рассмотрим пример выбора порядка модели.

# Юнит 14. Модели ARIMA

Основным недостатком процесса ARMA в ранее описанной форме является внутреннее требование стационарности, которое лежит в процедуре поиска коэффициентов. Действительно, если нам требуется получить наилучшее приближение (наименьшую ошибку) для одной части ряда - ряд должен быть аналогичным (или иметь аналогичное поведение) для своих будущих частей, чтобы гарантировать ту же ошибку. Чем менее стационарный ряд – тем больше ошибка и тем больше она растет при увеличении горизонта прогнозирования.

При анализе временных рядов для уменьшения (или устранения) требования повторяемости в первую очередь мы должны достичь стационарности. Можно свести задачу к стационарной, попробовав некоторые обратимые преобразования. Среди методов приведения ряда к стационарному такие как:

•Численная обычная производная – тренд.

•Численная дробная производная – сложный, циклический тренд.

•Численная сезонная производная – высокое влияние сезонности.

•Преобразования Бокса-Кокса – устранения волатильности (гетероскедастичности, в частности ).

•Приведение к виду - устранения волатильности (гетероскедастичности).

•Декомпозиция модели (напр. использование разных подходов для разных частей).

Как правило, устранить нестационарность тренда можно, взяв численную производную. Такая модель называется интегрированной авторегрессионной скользящей средней (ARIMA). В ряде случаев помимо обычной производной берут т.н. сезонную производную – такая модель называется SARIMA. В некоторых специфических случаях вместо обычной производной берут так называемую дробную производную, тогда модель будет называться ARIFMA или FARIMA.

Обозначим производную как d, численная производная может быть вычислена как

и т.д.

В случае рассмотрения вместо мы делаем предположение, что производная имеет стационарное поведение (как правило разности значений рядов более стационарны, чем сами ряды).

Таким образом можно задать модель ARIMA () где помимо уже известных коэффициентов () мы вводим еще один новый коэффициент численную производную. Не вдаваясь в подробности формул заметим, что производную мы вводим нативно. То есть модель по-прежнему остается «end2end». При этом, однако, производная берется только по части авторегрессии.

Отметим, что если , имеет линейный тренд; если , парабалический тренд и т. д. если не имеет тренда. При этом реально мы редко, когда знаем, что тренд именно линейный и тд. Но всегда можем разложить тренд в ряд Тейлора. На практике не рекомендуется брать производную слишком большого порядка. Как правило выбирают порядки.

Кроме того недодиффренцированный (слегка не стационарный) временной ряд – это плохо только на очень длинном горизонте прогнозирования. Поэтому лучше выбрать порядок производной поменьше. Такой эффект часто можно компенсировать увелечением порядков авторегрессии.

Другими словами Обычная численная производная убирает тренд во временном ряду. Модель с одним порядком дифференцирования предполагает, что исходный ряд имеет постоянный средний тренд (например, случайное блуждание или модель типа простого скользящего среднего). Модель с двумя порядками дифференцирования предполагает, что исходный ряд имеет изменяющийся во времени тренд (например, полиномиальный тренд или модель типа EMA).

В противовес сказанному избыточная дифференцированность приводит к необходимости добавления MA составляющих. Также отметим, что избыточная дифференцированность может привести к появлению ложных составляющих во ВР. То есть порядок дифференцирования тоже должен аккуратно варьироваться исследователем.

Также заметим, что помимо нативного встраивания производной мы могли бы перед процедурой ARIMA просто продифференцировать ряд, получив в итоге схожий эффект. Однако, «end2end» процедура выбора моделей для нас предпочительней с точки зрения ее оптимизации.

Для интересующихся (ЭТО НЕ НУЖНО ЧИТАТЬ ВСЕМ!) модель выглядит как , где – лаговый оператор (оператор запаздывания), то есть – оператор численной производной  
 . Если взглянуть на модель в лаговой форме, то важно понять, что дифференцированию подлежит только AR часть.

Частные случаи моделей ARIMA:

ARIMA (0,1,0) случайное блуждание - простейшая модель .

ARIMA (0,1,1) простое экспоненциальное сглаживание

Таким образом, первое, что нужно сделать при выборе начальных порядков модели ARIMA - это определить степень дифференцирования. То есть исключить тренд и сделать данные более стационарными.

Другими словами дифференцируемость это метод приведения к стационарности, который мы используем в нужном объем. Для проверки стационарности можно использовать множество статистических и визуальных методов, среди которых:

**Статистика скользящего окна** (**Rolling Statistics**): Постройте скользящее среднее и скользящее стандартное отклонение. Временные ряды являются стационарными, если они остаются постоянными во времени (посмотрите, являются ли линии прямыми и параллельными оси x). Этот метод соответствует слабому определению стационарного.

**АКФ анализ** - для нестационарного процесса вы увидите медленное убывание коэфициентоа АКФ. Для стационарной последовательной автокорреляционной функции (АКФ) график довольно быстро затухает до нуля или ниже.

**Статистические тесты, например расширенный тест Дики-Фуллера (Augmented Dickey-Fuller Test, ADF):** Временной ряд считается стационарным по определенному критерию, связанному с единичной окружностью на z плоскости. При расчете критерия также вычисляется p-значение (статистическая значимость). Если значение p низкое (согласно нулевой гипотезе) и критические значения при доверительных интервалах 1%, 5%, 10% максимально близки к статистике ADF.

Таким образом модель **ARIMA** имеет три порядка (), где  **– порядок численной производной** (разности). В первую очередь такая модель может компенсировать нестационарность тренда или другие медленные изменения. При выборе порядков модели сначала должен быть найден порядок дифференцирования. Позже мы увидим, как прием дифференцирования модифицирует задачу АРМА анализа.

# Юнит 15. Модель SARIMA

**Модель сезонной интегрированной авторегрессии - скользящего среднего (SARIMA).**

Если модель имеет интенсивную сезонную составляющую, то она может быть компенсирована так называемой сезонной производной, имеющей вид

Отметим, но читать это нужно, что в операторной форме это можно представить как

Таким образом, может быть введена так называемая сезонная авторегрессионная интегрированная скользящая средняя (SARIMA). Невникая в подробности уравнений скажем что полная модель записывается как **SARIMA(p,d,q)×(P,D,Q)s** или иногда как SARIMA(p,d,q)×(P,D,Q,s). Модель включает как обычные порядки , так и сезонные составляющие для сезонности с периодом , где – порядок т.н. сезонной авторегрессии, – порядок сезонного скользящего среднего, а – порядок сезонной производной.

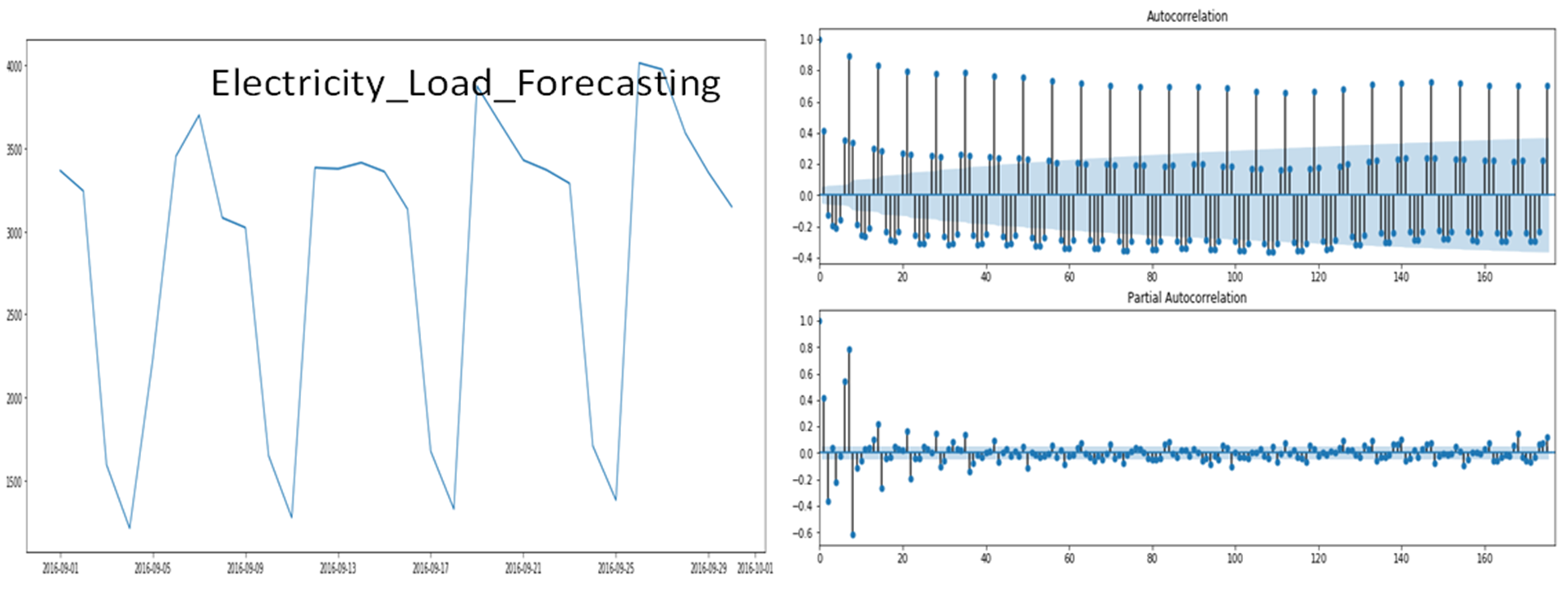
НЕЧИТАЕМ. Термины сезонных порядков мы не будем подробно рассматривать. Но скажем, что такое, например, сезонная авторегрессия. Сезонная авторегрессия могла бы быть задана как где – это коэффициенты сезонной авторегрессии. То есть вместо последовательных отсчетов мы возьмем только те, которые имеют тоже положение (фазу) в сезонности. Аналогично мы бы задали МА часть если бы это было нужно. Точно также мы могли бы записать уравнения для сезонной авторегрессии если речь шла бы о сезонных производных.

НЕЧИТАЕМ. Для интересующихся полное уравнение SARIMA можно представить как где – это оператор сезонного запаздывания, – оператор сезонного скользящего среднего. **Обратите внимание, что обе производные берутся только по части AR.**

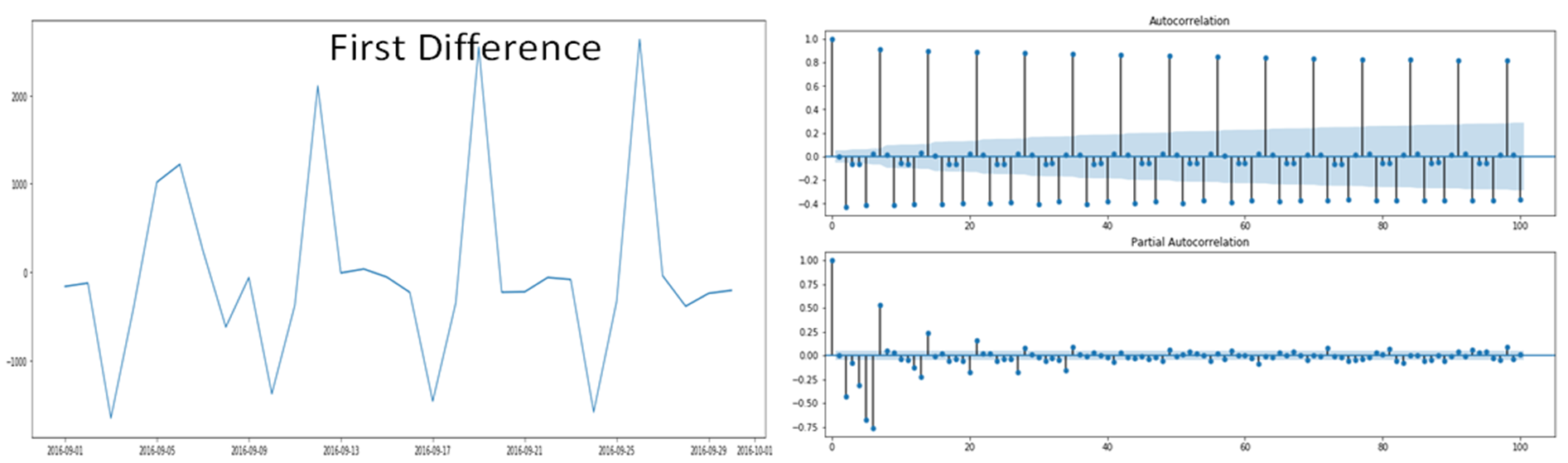
Таким образом **Модель сезонной интегрированной авторегрессии** скользящего среднего (SARIMA) - это модель нативно учитывающая даже сравнительно быстрые и интенсивные, но регулярные нестационарности типа сезонность. Для этого используется прием сезонного дифференцирования.

Важно отметить, что при выборе начальных значений порядков модели следует сначала определить порядки обычной и сезонной производной, приводящие ВР к стационарному виду.

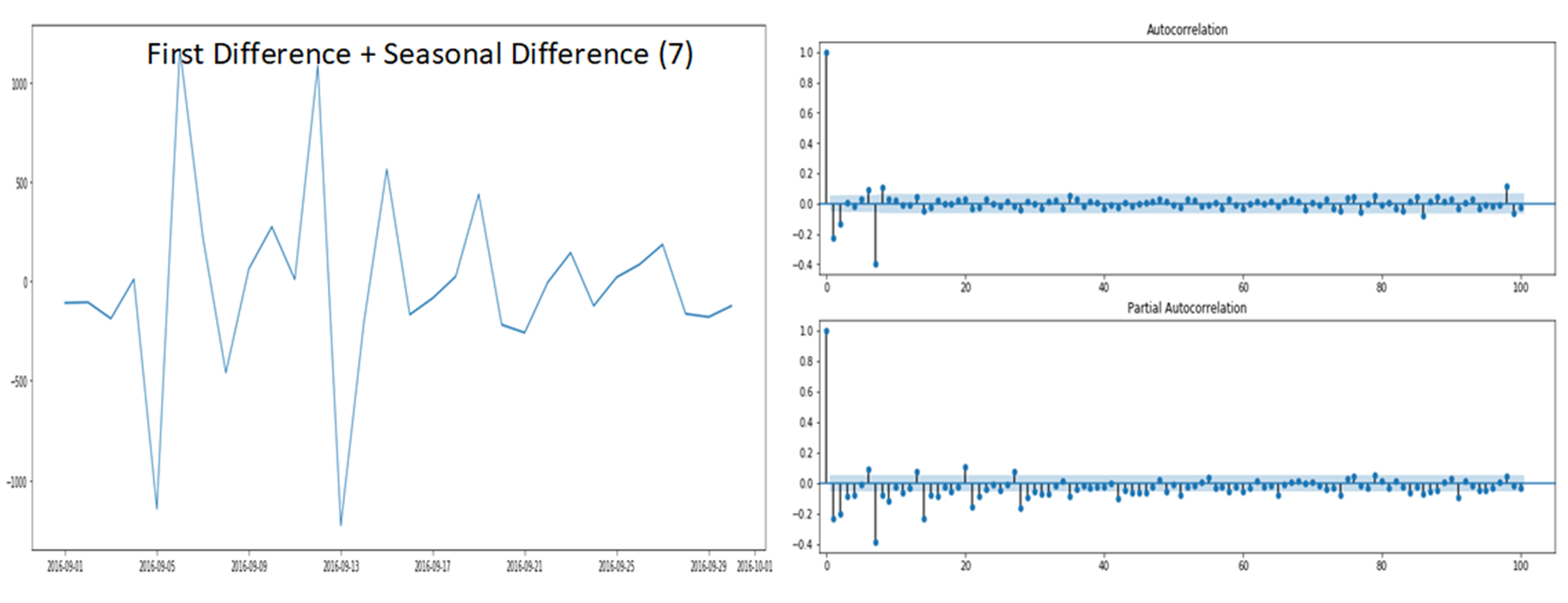
Рассмотрим пример ряда и его АКФ и ЧАКФ



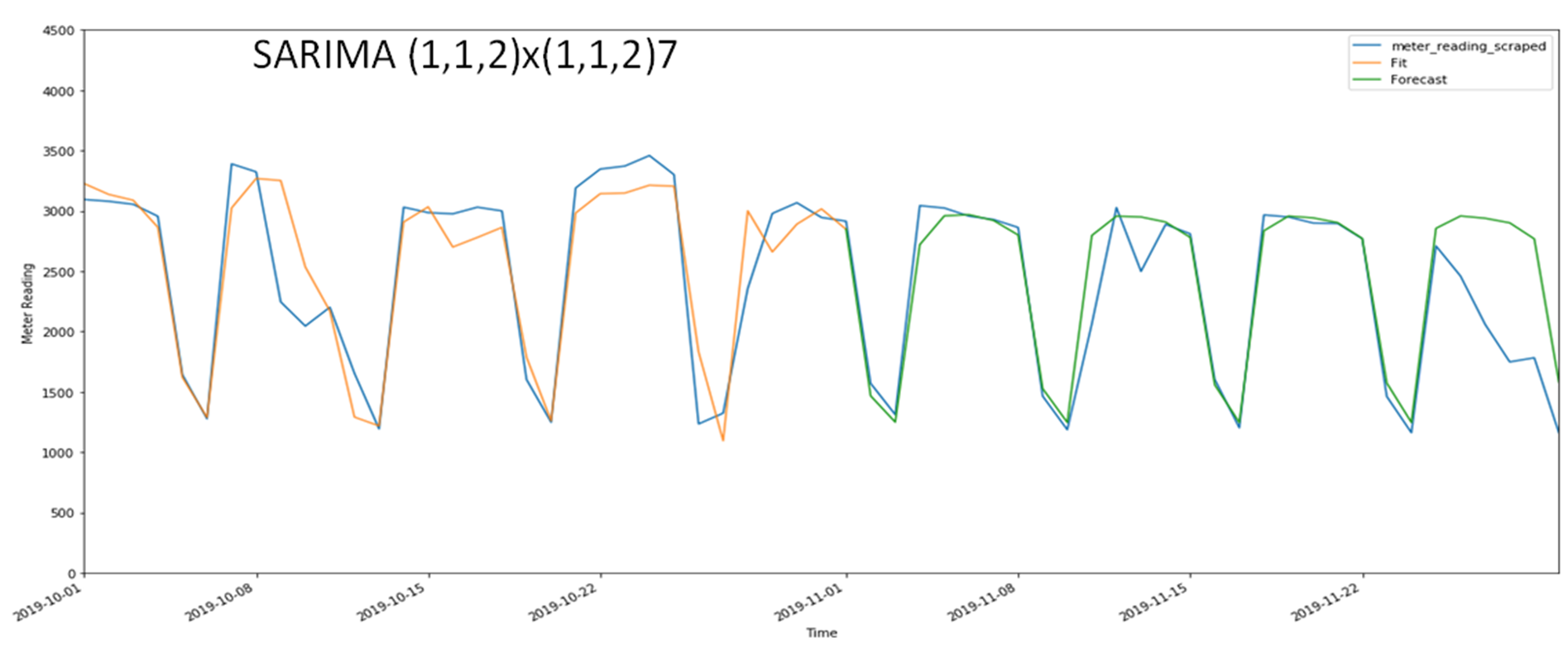
Ряд не стационарен, возможно, если убрать среднее, может и можно использовать ARMA, но слишком большого порядка - это плохо. Посмотрим первую простую производную ряда.



Видно явную сезонность, которая вызывает нестационарность. Посмотрим на сезонную производную.



Для сезонной производной ряд куда более стационарен. Видно, что есть AR 2 порядка, видно, что есть какое-то MA, допустим 1 порядка. Однако есть выбросы на АКФ по ним можно сказать о необходимости введения SAR составляющей (эмпирически так). Точно также по выбросам ЧАКФ можно сказать о необходимости SMA составляющих. Мы редко вводим такие составляющие высоких порядков. Не вдаваясь в подробности посмотрим как модель с такими компонентами аппроксимирует ВР.



Результат достаточно неплохой, хотя также видно, что какуюто нестационарность мы не перекрыли.

# **~~Юнит 15.1 Алгоритм SARIMA модели~~**

Таким образом общий Алгоритм подбора модели SARIMA

* Определение порядка обычного дифференцирования,
  + - необходимо для приведения ряда к стационарноcти (чтобы исключить тренд).
* Определение порядка основной сезонности.
* Проверка влияния сезонной составляющей и, возможно, устранение сильной сезонности путем сезонного дифференцирования.
* Определение порядков начальных значении AR и MA.
* Определение порядков начальных значений сезонных SAR и SMA.
* Расчет коэффициентов.
* Варьирование порядков модели
* Проверка результатов на валидационной выборке.

*Для справки укажем правила отбора исходных значений порядка*

1.**Правильный порядок -** это порядок разности, который делает временной ряд максимально шумоподобным т.е. значения колеблется около четко определенного среднего и имеют почти постоянный разброс.

Используйте сезонную производную только в случае сильного сезонного влияния.

Следует проверить дифференцированный ряд на стационарность по критериям, указанным ранее.

Часто оптимальный порядок дифференцирования - это тот порядок разности, при котором дисперсия наименьшая (вы должны стараться избегать чрезмерного дифференцирования).

Если ряд имеет устойчивый и последовательный сезонный шаблон, рекомендуется повысить порядок сезонной разницы.

*2. Опционально! Читать не нужно.* ***После дифференцирования выберите рассмотрите ACF и PACF для выбора основных порядков p,q****.*

*Количество AR слагаемых определяется как последний лаг PACF перед быстрым спадом (от положительных значений до нуля).*

*Количество MA слагаемых определяется как последний лаг ACF перед быстрым ростом (от отрицательных значений до нуля).*

*Если ваш ряд недостаточно дифференцирован, добавьте дополнительное слагаемое к AR.*

*Если PACF ряда показывает резкий спады и/или автокорреляция лаг-1 положительна, то задержка, при которой PACF спадает - это необходимое количество членов AR.*

*Если ваш ряд избыточно дифференцирован, добавьте дополнительное слагаемое к MA. Если ACF ряда показывает резкое падение и/или автокорреляция лаг-1 отрицательна, то запаздывание, при котором АКФ спадает, является указанным количеством слагаемых MA.*

*Если модель полная (AR и MA), член AR и член MA могут нейтрализовать эффекты друг друга. В этом случае попробуйте уменьшить порядки модели, особенно если оценки параметров в исходной модели требуют более 10 итераций для схождения.*

*Если сумма коэффициентов AR почти равна 1 – рекомендуется уменьшить количество членов AR на один и увеличить порядок дифференцирования на единицу.*

*Если сумма коэффициентов МА почти равна 1 - рекомендуется уменьшить количество членов МА на один и уменьшить порядок дифференцирования на единицу.*

*3.****После рассмотрите ACF и PACF для выбора сезонных порядков P,Q****.*

*Добавить слагаемое SAR, Если ACF периодически положительный.*

*Также это можно оценить если PACF периодически отрицателен .*

*Посмотрите на количество значительных лагов, которые кратны периоду сезона.*

*Например, если период равен 24, и мы видим, что 24-е и 48-е запаздывания являются значительными в PACF, это означает, что начальное значение P должно быть равно 2.*

*Добавить слагаемое SMA, Если ACF периодически отрицательный.*

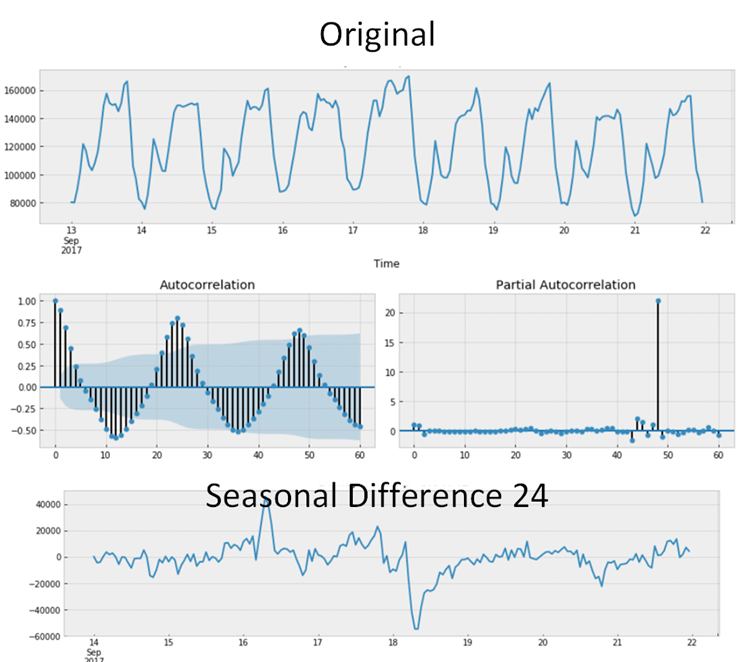
*Используйте те же правила для определения количества лага что касается SAR*

*Также это можно оценить если PACF периодически положительный .*

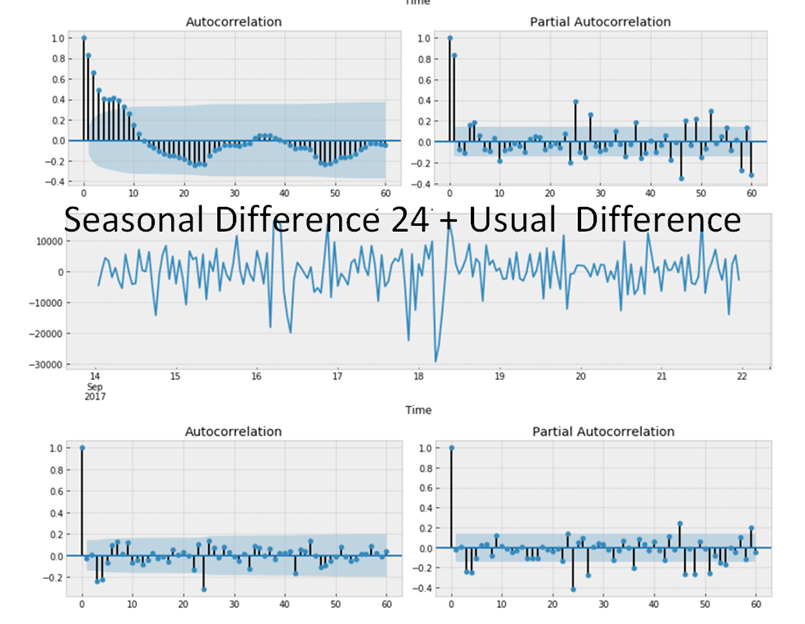
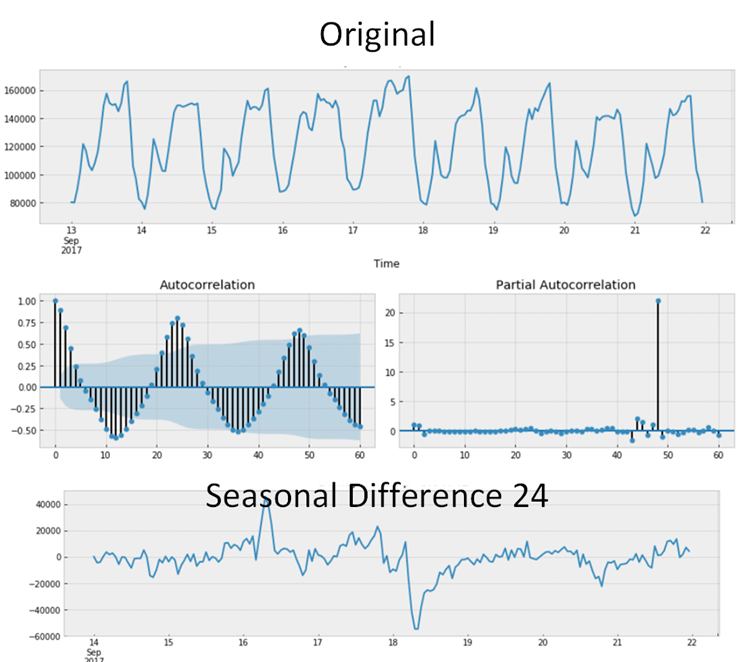
*Старайтесь избегать использования более одного или двух сезонных слагаемых (SAR + SMA) в одной модели, так как это может привести к переобучению данных и / или проблемам в оценке.*

*Если в ряде есть положительные значения ACF с большим запаздыванием, добавьте разностный порядок.*

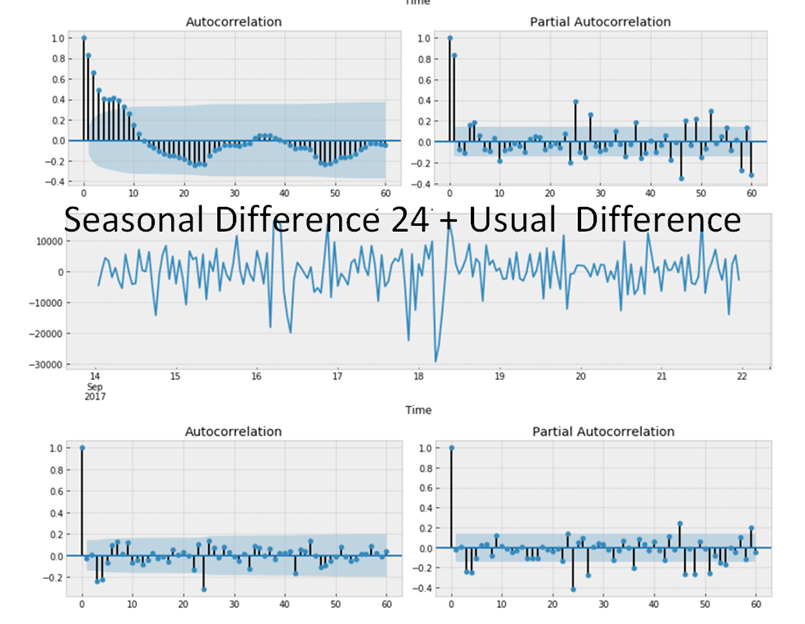
**Рассмотрим пример.**

****

**Процесс явно нестационарный. Посмотрим на производную**

****

**На глаз видим, что ВР все равно не стационарный.**

****

В данном примере , скорее всего, равно 2, это последний существенный лаг PACF, после которого большинство других не являются значительными. Также равно 1, должно быть где-то около 4, как видно на ACF. Также может быть 2, так как 24-е и 48-е лагы значительны на PACF. При этом равно 1, потому что мы провели сезонную дифференциацию, , вероятно, равен 1 - 24-е отставание по ACF значительно, а 48-е - нет.

Как говорилось ранее, помимо выбора начальных порядков, который вообще говоря является достаточно приблизительной оценкой реально подходящего порядка модели, необходим подбор более релевантных порядков. Также напомним, что в этом случае порядок выбирается как минимум информационного критерия. Наиболее популярные информационные критерии следующие:

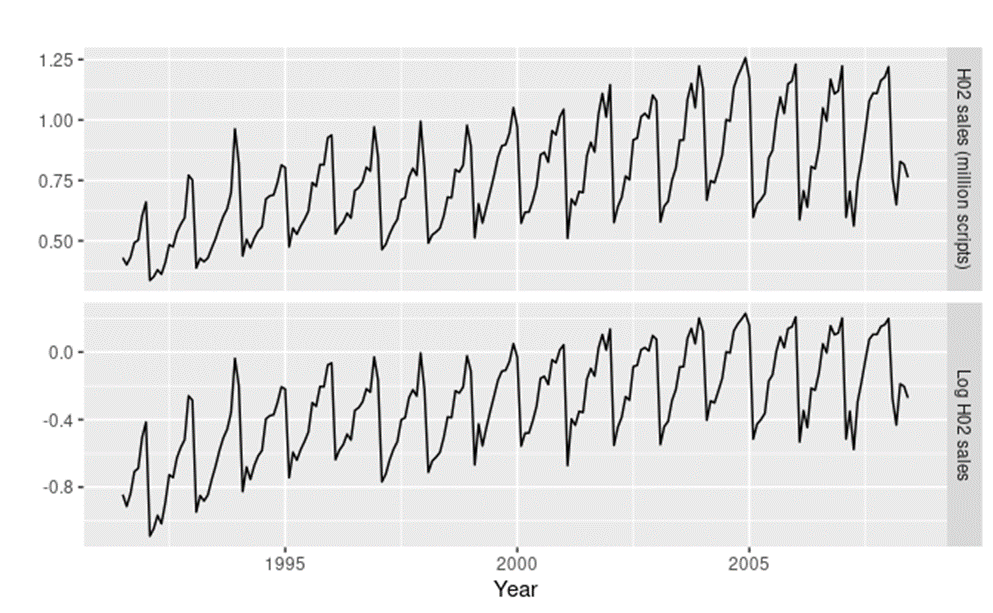
* ,
* ,
* ,

где - полное количество параметров: для ARMA без константы в другом случае ; для ARIMA ;   
для SARIMA .

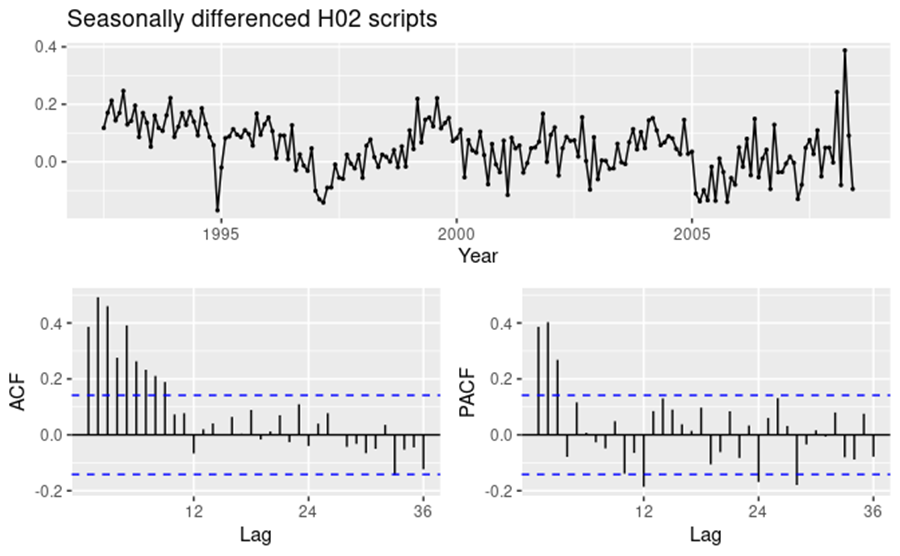
Специфика этих критериев заключается в совместной оптимизации количества параметров и значения RSS. По этим критериям AIC является наиболее популярным, но предполагается AICc, что обеспечит лучший результат. Указанные критерии могут быть применены для автоматизации параметрического поиска. Автоматизированный поиск будет рассмотрен в практических юнитых.

**Пример оценки коэффициента ARMA**

**Пример:** прогноз продаж. На исходном графике наблюдается небольшое увеличение дисперсии с уровнем, поэтому мы берем логарифмы, чтобы стабилизировать дисперсию.

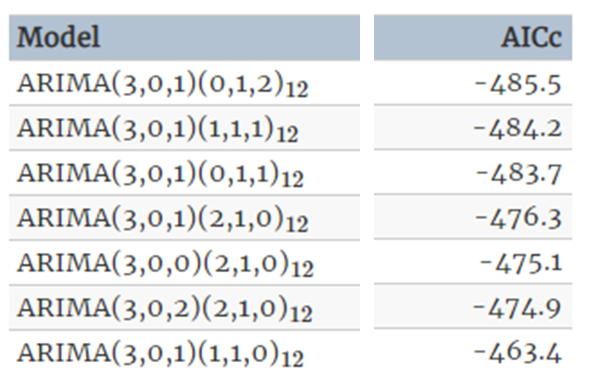
****

Данные сильно сезонны и явно нестационарные, поэтому используется сезонная разница (с сезонностью ).

****

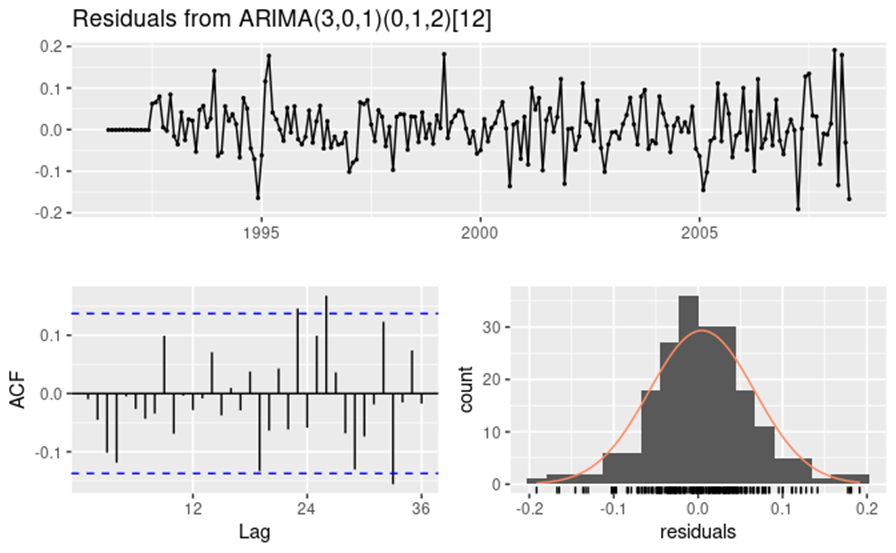
На данный момент неясно, следует ли нам делать еще одно дифференцирование или нет. Решаем не делать этого, но выбор не очевиден. На полученном графике есть всплески в PACF при лагах 12 и 24, но нет всплесков на при сезонных лагах в ACF. Это может указывать на порядок SAR. В несезонных лагах есть 3 значительных всплеска PACF, что указывает на возможный порядок AR (3). ACF не указывает на какую-либо простую модель.

Этот первоначальный анализ предполагает, что возможной моделью для этих данных является . Теперь посмотрим на вариацию параметров:

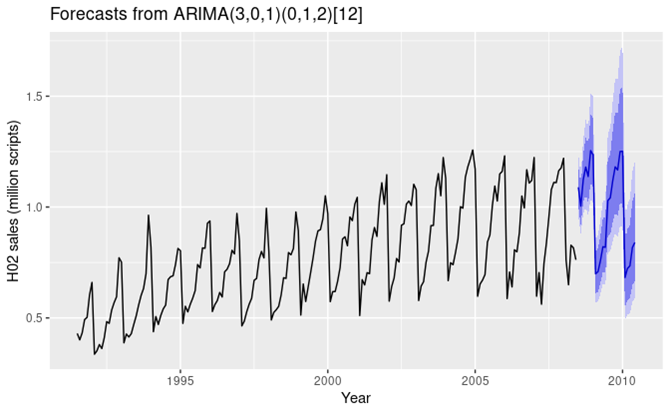


Автоподбор параметров опровергает первое предположение - лучше использовать модель .

Теперь посмотрим на остаток выбранного процесса.



В ACF остатков есть несколько значительных всплесков, поэтому модель не соответствует требованиям Ljung-Box test (критерию стат. значимости). Это значит, что модель конечно можно использовать для прогнозирования, но интервалы прогнозирования могут быть неточными из-за коррелированных остатков.



# Юнит 16 – Другие модели предсказания.

Отметим, что помимо рассмотренных есть и другие модели класса SARIMA. Помимо обсуждаемой выше модели SARIMA, она обычно рассматривается как несколько других:

1.**SARIMAX** - Модель SARIMA с экзогенными добавочными факторами.

2.**ARIFMA**- Модель ARIMA с использованием дробных производных (и дробного интегрирования).

3.**VAR, VMA, VARMA** - многомерные модели. То есть модели для многомерных ВР, где все компоненты учитываются одновременно.

Среди указанных модификаций важно отметитьмодели **SARIMAX**. В отличии от уже изученных моделей эта модификация включает понятие **Экзогенные факторы** (exdogs, экзогенные **ковариаты**) или набор факторов (). Экзогенные факторы являются дополнительными факторам к целевой переменной , которые статистически не зависят от нее, но влияют на нее. Например, средняя цена на нефть на предыдущей неделе , влияет на обменный курс валют на этой неделе . Вы можете использовать модель SARIMAX, чтобы проверить, влияет ли набор экзогенных переменных на основной временной ряд. Другими словами, Экзогенная переменная - это переменная, значение которой определяется вне модели и накладывается на нее. В отличии от экзогенных, эндогенная переменная - это переменная, значение которой определяется моделью. Здесь основной прогнозируемый ряд является эндогенной переменной.

Во временных рядах экзогенная переменная - это параллельные временные ряды, которые не моделируются напрямую, но используются в качестве взвешенных входных данных для модели. При этом значения экзогенных (внешних) факторов должны соответствовать выбранному временному промежутку.

Отметим то, что читать нужно! Для ARMAX экзогенные факторы могут быть введены как , где – это коэффициент влияния экзогенной переменной на результат.

Также отметим, что факторы, очевидно коррелирующие с рядом(входные), можно рассматривать как эндогенные (эндоги). Факторы, не имеющие линейной корреляции с временным рядом можно рассматривать как экзогенные. Примером эндогенных факторов является влияние осадков на рост растений - факторов коррелируют и их изучают вместе.

Другими модификациями моделей класса ARIMA являются ее расширения, например, ансамблирование или конвейеризация. Однако, эти термины мы употребляем в более широком смысле, чем это известно читателю. Типичные приемы тут могут быть следующие:

* **Предварительное преобразование ВР**, например, приведение ВР к стационарному виду при помощи вычитания среднего значения или преобразования Бокса-Кокса. А также такие приемы, как очистка данных от аномалий или разделение на сегменты, каждый из которых по мнению исследователя лучше исследовать отдельно.
* **Разложение временного ряда на составляющие**. Например, построение отдельных моделей для тренда и сезонных составляющих. Таких составляющих может быть несколько, и каждую лучше учитывать отдельно. Этот прием можно **ансамблированием** в некотором смысле.
* Использование последовательности нескольких моделей таким образом, что каждая следующая обрабатывает остаток от работы предыдущей. Этот прием можно назвать **конвейеризацией** в некотором смысле.
* **Расширения понятия модели** (например) ARIMA путем использования альтернативных оптимизаторов. Например, замена решения регрессии методом наименьших квадратов на другие оптимизацию методом градиентного спуска для других функций потерь.

Тут приведен далеко не полный перечень приемов.

Типичными примерами конвейеризации является аппроксимация ВР регрессией низкого порядка (фактически аппроксимация тренда ВР регрессией), затем вычитание тренда и аппроксимация остатка при помощи значительно более простой модели ARMA. Такой подход сложнее в плане его общей оптимизации, однако может быть более устойчивым – то есть давать лучшую обобщающую способность.

Другой пример такого подхода — это модели семейства «**B**ox-Cox transformation, **A**RMA errors, **T**rend, and **S**easonal components» или BATS. Общий подход таких моделей — это комбинация идей подхода Холта-Винтера и моделей ARMA для преобразованного временного ряда. Общая модель записывается как

где – это долгоиграющий тренд (уровень ) и его изменения, – это набор сезонных составляющих до порядка ; – шумы. Сезонные составляющие могу быть описаны как = , где – это период сезонности или другим способом. Если сезонности разложить в ряд Фурье, предположив, что они не стационарны, то модель будет называется Тригонометрический BATS = TBATS.

Примером расширения понятия модели (например) ARMA является модели AR-Net. В этих моделях авторегрессия рассматривается как некоторый многослойный персептрон без скрытых слоев и функций активации. Тогда к авторегрессии можно применить весь уже известный читателю аппарат оптимизации и регуляризации. В частности, регуляризация позволяет снизить влияние неверно выбранного порядка модели. А можно и ввести дополнительный скрытый слой. Если в модель добавить нелинейности, то модель будет называться NAR (non-lineaer autoregression). Могут быть и другие модификации подхода. Такой подход AR-Net дополняет, например, модель Prophet создавая из нее т.н. модель Neural Prophet. Могут быть и другие модификации ARMA моделей, например, для аппроксимации не только самой модели, но и ее дисперсии (если дисперсия волотильна – гетеросекдастичная). Такая модификация называется GARCH.

# Юнит 17. SKTime\_ARIMA

# **Использование моделей АРСС**

Один из методов, доступных моделирования и прогнозирования временных рядов, известен как SARIMAX, что означает сезонное авторегрессионное интегрированное скользящие средние с экзогенными регрессорами. В частности могут быть такие подвиды модели, как:

* AR (AutoRegressive);
* MA (Moving Average);
* SAR (Seasonal AutoRegressive);
* SMA (Seasonal Moving Average);
* ARMA (AutoRegressive-Moving Average);
* ARIMA (AutoRegressive-Integrated-Moving Average);
* SARIMA (Seasonal AutoRegressive-Integrated-Moving Average);
* SARIMAX (Seasonal AutoRegressive-Integrated-Moving Average with eXogenous factors).

Мы рассмотрим вариант SARIMA.

Семейство моделей SARIMAX относится к методам параметрической статистики. То есть свойства модели, в том числе, точность ее приближения к данным задаются набором параметров модели. В данном случае параметры называются порядком модели. Могу быть заданы отдельно порядок авторегресионной части модели; порядок скользящего среднего; порядок интегрирования (дифференцирования) и т.д.

Подход SARIMAX был предложен Боксом-Дженкинсом. Методология подхода включает следующие пункты:

* Идентификация модели: используйте графический метод и методы сводной статистики для определения характера тренда и сезонности. А также, чтобы получить представление о порядке производной (d) и порядках авторегрессии p и q скользящего среднего.
* Оценка модели: оценка коэффициентов регрессионной модели.
* Диагностика модели максимального правдоподобия: используйте графический метод и статистические тесты остаточных ошибок (разности), чтобы определить особенности данных, не охваченной моделью.

Импорт библиотек

**import** pandas **as** pd  
**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt

**try**:  
 **import** sktime  
**except**:  
 !pip install sktime --user  
 !pip install pmdarima  
 !pip install statsmodels  
 !pip install prophet  
**import** sktime

**import** statsmodels.api **as** sm  
**from** statsmodels.tsa.stattools **import** adfuller  
**import** pmdarima **as** pm

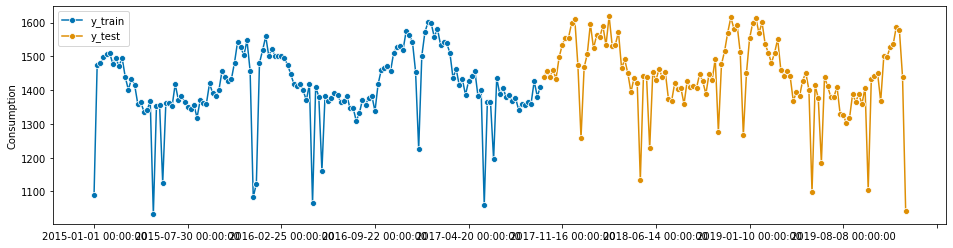
**from** sktime.utils.plotting **import** plot\_series  
**from** sktime.forecasting.sarimax **import** SARIMAX  
**from** sktime.forecasting.arima **import** AutoARIMA  
**from** sktime.forecasting.base **import** ForecastingHorizon  
**from** sktime.forecasting.model\_selection **import** temporal\_train\_test\_split  
  
**from** sktime.performance\_metrics.forecasting **import** MeanAbsolutePercentageError, MeanSquaredError  
smape = MeanAbsolutePercentageError(symmetric = True)  
rmse = MeanSquaredError(square\_root=True)  
r2\_score = **lambda** y\_pred, y\_test: 1-np.sum(np.square(y\_pred - y\_test))/np.sum(np.square(y\_test - np.mean(y\_test)))

**import** warnings  
**from** statsmodels.tools.sm\_exceptions **import** ConvergenceWarning  
warnings.simplefilter('ignore', ConvergenceWarning)

Импорт данных аналогично предыдущего урока.

path\_ts = 'de\_data.csv'  
  
df = pd.read\_csv(path\_ts, parse\_dates=['Date'], index\_col="Date")  
df=df.fillna(df.mean())  
  
y = df.Consumption.asfreq('7d')  
  
TEST\_SIZE = int(0.45\*y.size)  
  
y\_train, y\_test = temporal\_train\_test\_split(y, test\_size=TEST\_SIZE)  
  
print(f'Check splitted data size: Train: {y\_train.shape[0]}, Test: {y\_test.shape[0]}')  
  
sktime.utils.plotting.plot\_series(y\_train, y\_test, labels=["y\_train", "y\_test"]);

Check splitted data size: Train: 144, Test: 117



## **SARIMA Forecasting**

Для модели AutoregRessive Integrated Moving Average (ARIMA) существует три параметра (порядка) (p, d, q):

* - **авторегрессивная часть модели**. Этот параметр позволяет учесть влияние прошлых значений на текущее для модели. Прошлые значения здесь называются запаздывающими наблюдениями (также известными как «запаздывание» или «лаг»). Интуитивно это похоже на утверждение, что завтра, вероятно, будет тепло, если в последние 3 дня было тепло. Другими словами, здесь мы можем сказать, что текущее значение температуры зависит от последних трех значений.
* – **интегрирование модели**. Этот параметр включает в себя степень различия лагов (то есть количество прошлых временных точек, которые нужно вычесть из текущего значения), чтобы сделать временной ряд стационарным (чтобы исключить часть тренда). Интуитивно это было бы похоже на утверждение о том, что, вероятно, будет одно и то же повышение температуры каждый день (или одно и то же ускорение для второй производной и т.д.).
* - **скользящая средняя часть модели**. Этот параметр позволяет представить остаточную часть (шум, ошибку) модели как линейную комбинацию остаточных значений, наблюдаемых в предыдущие моменты времени.

Если в модели видно достаточно сильное влияние сезонной составляющей, то следует перейти к модели ARIMA с сезонными эффектами (Seasonal ARIMA - SARIMA). Как правило такая модель обозначается как

Где

* () - несезонные параметрами, описанными выше;
* () - сезонные порядки, которые следуют тому же порядку определений как и описанные выше, но применяются к сезонной составляющей временного ряда;
* - это периодичность временного ряда (например, 4 для квартальных периодов, 12 для годовых периодов и т.д.).

Рассмотрим процедуру ручного выбора порядков модели. Как правило, начинать следует с дифференцирования для достижения стационарности. Как правило, это 1-3 порядок, реже - больше.

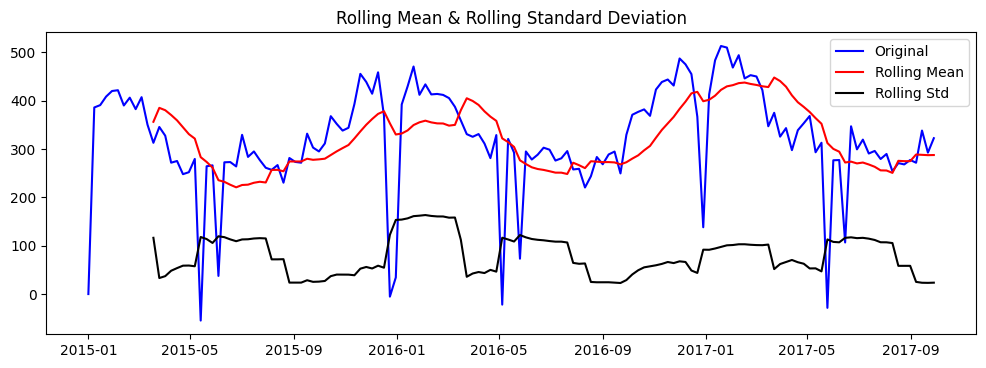
Правильный порядок дифференцирования () это порядок разности, который делает временной ряд шумоподобным т.е. значения колеблется около четко определенного среднего и имеют почти постоянный разброс значений. При этом рекомендуется использовать сезонную производную только в случае сильного сезонного влияния.

Для проверки стационарности после дифференцирования можно использовать несколько методов, в том числе:

* **Скользящая статистика**: построение скользящего среднего и скользящего стандартного отклонения. Идея этого метода в том, что временные ряды являются стационарными, если они остаются неизменными во времени. Скользящая статистика визуально показывает стационарность среднего значения.
* **Статистические тесты, в т.ч. Расширенный тест Дики-Фуллера**: временной ряд считается стационарным, если значение низкое (в соответствии с нулевой гипотезой), а критические значения с доверительными интервалами максимально близки к табличному значению параметра ADF (такой параметр принято называть статистика ADF). Если тест ADF показывает, что статистика ADF далека от критических значений, а значение превышает пороговое значение (например (), то ряд является нестационарным. В обратном случае ряд следует признать стационарным.

Проверим стационарность для нашего ряда.

rolling\_mean = y\_train.rolling(window = 12).mean()  
rolling\_std = y\_train.rolling(window = 12).std()  
  
plt.figure(figsize=(12,4), dpi=100)  
  
plt.plot(y\_train-y\_train[0], color = 'blue', label = 'Original')  
plt.plot(rolling\_mean-y\_train[0], color = 'red', label = 'Rolling Mean')  
plt.plot(rolling\_std, color = 'black', label = 'Rolling Std')  
  
plt.legend(loc = 'best')  
plt.title('Rolling Mean & Rolling Standard Deviation')  
plt.show()

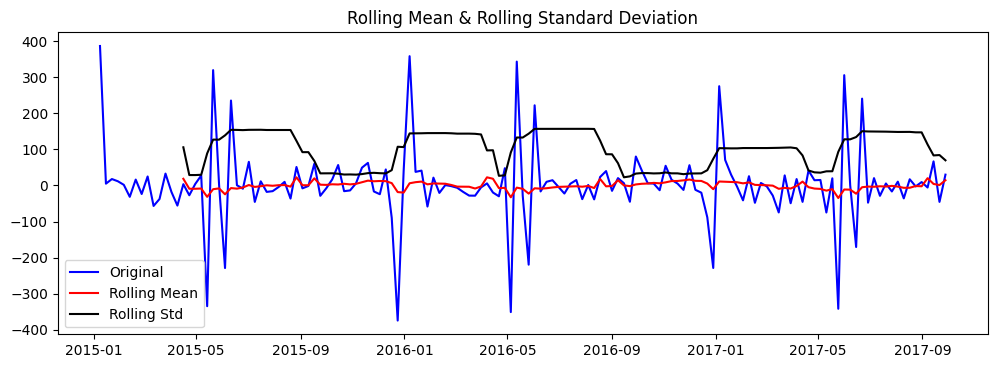


P\_THRESHOLD = 0.05  
**def** check\_ADF(y, p\_threshold = P\_THRESHOLD):  
 result = adfuller(y)  
 adf\_value = result[0]  
 p\_value = result[1]  
 print('ADF Statistic: {:.4f}'.format(adf\_value))  
 print('p-value: {:.4f}'.format(p\_value))  
 print('Critical Values:')  
 **for** key, value **in** result[4].items():  
 print('\t{}: {:.4f}, {}'.format(key, value, 'outperformed' **if** adf\_value>value **else** ""))   
 print(f'Result: The series is {"not " **if** p\_value < p\_threshold **else** ""}stationary')  
 **return** result  
  
check\_ADF(y\_train, p\_threshold = P\_THRESHOLD);

ADF Statistic: -3.3857  
p-value: 0.0115  
Critical Values:  
 1%: -3.4783, outperformed  
 5%: -2.8826,   
 10%: -2.5780,   
Result: The series is not stationary

В данной ситуации оказалось так, что ряд не стационарен. В этом случае стационарности можно добиться дифференцированием. Рассмотрим классической дифференцирование.

y\_diff = y\_train[:].diff(1).dropna()  
  
rolling\_mean = y\_diff.rolling(window = 15).mean()  
rolling\_std = y\_diff.rolling(window = 15).std()  
  
plt.figure(figsize=(12,4), dpi=100)  
  
plt.plot(y\_diff, color = 'blue', label = 'Original')  
plt.plot(rolling\_mean, color = 'red', label = 'Rolling Mean')  
plt.plot(rolling\_std, color = 'black', label = 'Rolling Std')  
  
plt.legend(loc = 'best')  
plt.title('Rolling Mean & Rolling Standard Deviation')  
plt.show()  
  
result = check\_ADF(y\_diff)

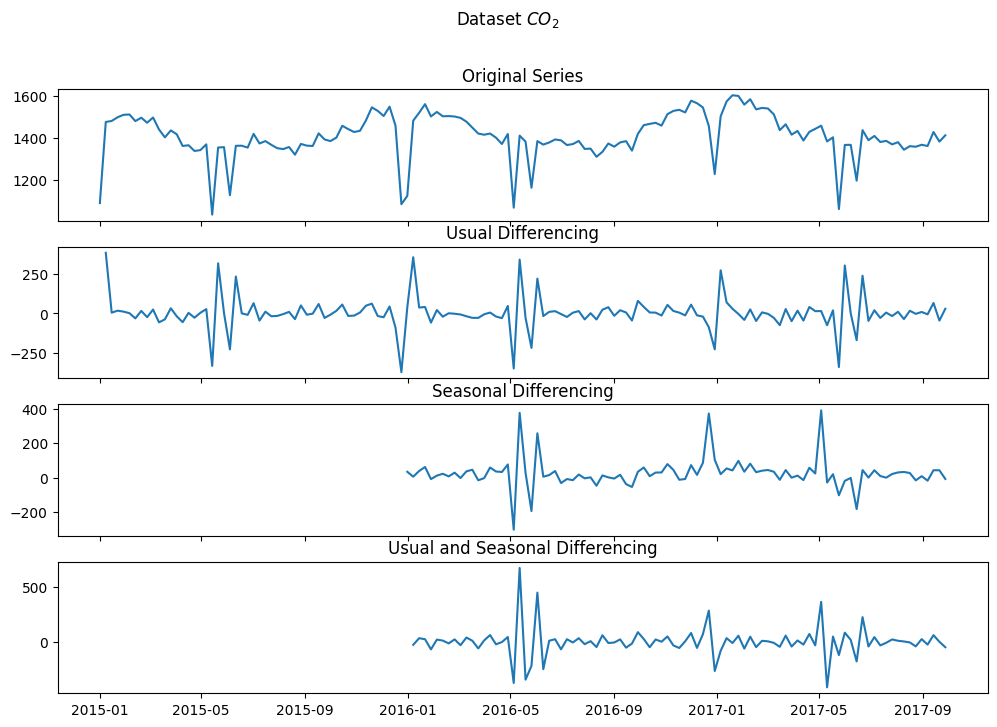


ADF Statistic: -8.5914  
p-value: 0.0000  
Critical Values:  
 1%: -3.4783,   
 5%: -2.8826,   
 10%: -2.5780,   
Result: The series is not stationary

Как и следовало ожидать (в ряду преобладает сезонная составляющая) ряд по прежнему не стационарен.

Попробуем теперь модель с сезонным дифференцированием.

SEASON = 52  
  
*# Plot*  
fig, axes = plt.subplots(4, 1, figsize=(12,8), dpi=100, sharex=True)  
  
*# Original Series*  
axes[0].plot(y\_train[:])  
axes[0].set\_title('Original Series')  
  
*# Usual Differencing*  
axes[1].plot(y\_train[:].diff(1))  
axes[1].set\_title('Usual Differencing')  
  
*# Seasinal Differencing*  
axes[2].plot(y\_train[:].diff(SEASON))  
axes[2].set\_title('Seasonal Differencing')  
  
*# Seasinal and Usual Differencing*  
axes[3].plot(y\_train[:].diff(1).diff(SEASON))  
axes[3].set\_title('Usual and Seasonal Differencing')  
  
plt.suptitle('Dataset $CO\_2$', fontsize=12)  
plt.show()



В последнем случае мы имеем более не менее стационарный случай. Проверим это при помощи теста.

y\_sdif = y\_train[:].diff(1).diff(SEASON).dropna()  
results = check\_ADF(y\_sdif);

ADF Statistic: -8.0074  
p-value: 0.0000  
Critical Values:  
 1%: -3.5079,   
 5%: -2.8954,   
 10%: -2.5848,   
Result: The series is not stationary

Хотя тест дал не лучшие результаты, мы можем заключить что предварительной оценкой порядков дифференцирования (интегрирования) являются: .

После выбора порядков дифференцирования следует перейти к оценке порядков авторегрессии и скользящего среднего. Как правило, вручную, эти параметры выбираются при помощи автокорреляционной и частичной автокорреляционной функций (ACF и PACF соответственно).

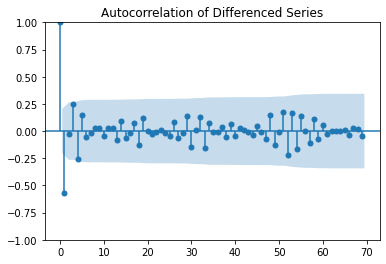
Как правило, графики ACF и PACF состоят из точек, соответствующих значениям этих функций для т.н. "лагов" - то есть значений, вычисленных для т.н. "запаздывания" одной из копий функции относительно другой (см. определение ACF и PACF). Также на графиках обычно отображают доверительные интервалы, которые имеют вид конуса в районе нуля. Эти интервалы показывают порог проверки гипотезы о белом шуме. Другими словами, все значение ниже уровень доверительного интервала скорее всего не имеют статистической значимости. По умолчанию установлен доверительный интервал 95%, что предполагает, что значения корреляции за пределами этого интервала, скорее всего, являются корреляцией, а не статистической случайностью.

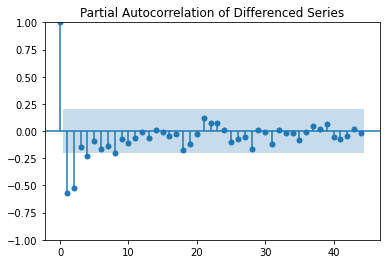
При ручном выборе порядков модели рекомендуются следующие правила (**ИЗУЧЕНИЕ ПРАВИЛ ОПЦИОНАЛЬНО**):

* **Количество слагаемых AR (AR порядок)** определяется как последнее значение лага PACF перед быстрым уменьшением от положительных значений до нуля.
* **Количество слагаемых скользящего среднего (MA)** определяется как последнее значение лага ACF перед быстрым увеличением от отрицательных значений до нуля.
* **Добавьте слагаемое SAR**, если значения PACF периодически положительны.
* **Порядок SAR** может быть оценен из PACF. Посмотрите на количество значений лагов выше уровня шума, которые кратны периоду сезона. Например, если период равен 24, и мы видим, что 24-е и 48-е запаздывания значительны в PACF, это означает, что начальное P должно быть 2.
* **Добавьте член SMA**, если значения ACF периодически отрицательный.
* **Порядок SMA** может быть оценен из ACF. Посмотрите на количество отрицательных значений лагов выходящих за уровень шума, которые кратны периоду сезона.
* Если временной ряд немного недодифференцирован, добавьте дополнительное слагаемое в AR.
* Если ваши ряды немного передифференцирован, добавьте дополнительные слагаемое в MA.
* Старайтесь избегать использования более одного или двух сезонных порядков (SAR + SMA) в одной модели, так как это может привести к переобучению данных и/или проблемам в точности оценок.
* Если все значения ACF и PACF кроме 0 не выходят за доверительный интервал следует считать модель белым шумом.

Рассмотрим графики ACF и PACF.

**from** statsmodels.graphics.tsaplots **import** plot\_acf  
**from** statsmodels.graphics.tsaplots **import** plot\_pacf  
  
*# Usual Differencing*  
plot\_acf(y\_sdif, title='Autocorrelation of Differenced Series', lags=np.arange(70) );plt.show()  
*# Usual Differencing*  
plot\_pacf(y\_sdif, title='Partial Autocorrelation of Differenced Series', method='ywm', lags=np.arange(45));plt.show()  
plt.show();





Из графиков выше следует следующий выбор параметров.

* порядка AR (1 и 2 лаги PACF ниже нуля);
* порядок MA (на ACF видна некоторая нестационарность, но 1 и 2 лаги не превысили доверительный интервал);
* порядок SAR (на PACF нет значимых лагов выше нуля);
* порядок SMA (на ACF нет значимых лагов ниже нуля);
* Ряд немного недодифференцирован, поэтому установим порядок AR;

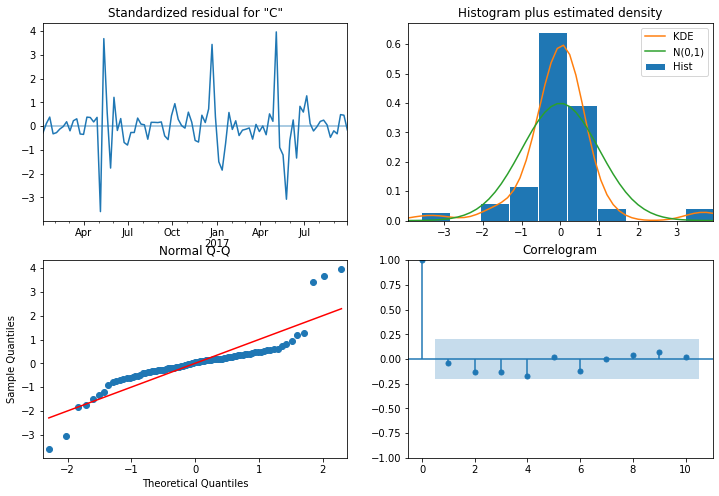
Теперь давайте проверим модель Напомним, что после оценки предварительных параметров модели, чаще всего, необходимо провести некоторую до настройку (подбор) значений параметров. Это делается по нескольким критериям, в том числе особым критериям выбора порядка ARMA моделей, например BIC, AIC и т.д.  
Отметим, что в данном случае мы будем рассматривать класс SARIMAX из пакета SKTime. Данный пакет основан на функциях из библиотеки [statsmodels](https://www.statsmodels.org/stable/index.html).

forecaster = SARIMAX(order=(3, 1, 0), seasonal\_order=(0, 1, 0, 52))  
forecaster.fit(y\_train)  
print(forecaster.summary())

SARIMAX Results   
==========================================================================================  
Dep. Variable: Consumption No. Observations: 144  
Model: SARIMAX(3, 1, 0)x(0, 1, 0, 52) Log Likelihood -544.223  
Date: Tue, 26 Jul 2022 AIC 1098.445  
Time: 16:39:06 BIC 1110.999  
Sample: 01-01-2015 HQIC 1103.510  
 - 09-28-2017   
Covariance Type: opg   
==============================================================================  
 coef std err z P>|z| [0.025 0.975]  
------------------------------------------------------------------------------  
intercept -0.3869 11.340 -0.034 0.973 -22.612 21.839  
ar.L1 -0.9386 0.072 -13.096 0.000 -1.079 -0.798  
ar.L2 -0.6374 0.107 -5.936 0.000 -0.848 -0.427  
ar.L3 -0.1381 0.117 -1.183 0.237 -0.367 0.091  
sigma2 9056.4590 709.265 12.769 0.000 7666.324 1.04e+04  
===================================================================================  
Ljung-Box (L1) (Q): 0.16 Jarque-Bera (JB): 167.25  
Prob(Q): 0.69 Prob(JB): 0.00  
Heteroskedasticity (H): 1.00 Skew: 0.67  
Prob(H) (two-sided): 1.00 Kurtosis: 9.50  
===================================================================================  
  
Выведенные результаты аппроксимации модели раскрывает достаточно много информации. В первой таблице представлена общая информация, включая критерии качества (AIC, BIC и HQIC). Таблица посередине - это таблица коэффициентов, где значения столбца coef - это веса соответствующих слагаемых. Значение sigma2 – это RSS (средне квадратическая) ошибка модели. В последней таблице представлены результаты различных статистических тестов для полученных остатков.

Помимо табличного представления, мы можем проводить диагностику остатков графическим способом. Напомним, что остатки в данном контексте означают разность модели, аппроксимированной при помощи SARIMA и тренировочных данных. Отметим, что данный способ не является стандартным методом класса SARIMAX, однако присутствует в нем.

forecaster.\_fitted\_forecaster.plot\_diagnostics(figsize=(12,8));



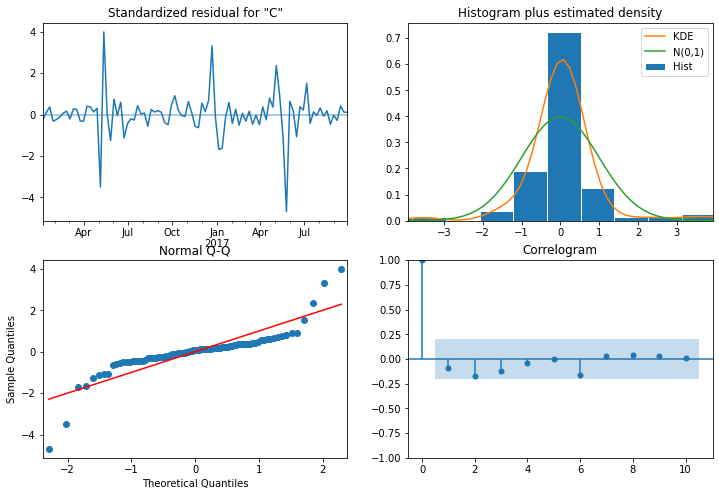
На графиках выше мы видим:

* *верхний левый график*: остаточные ошибки колеблются около нулевого среднего, однако имеют несколько не равномерную дисперсию, возможно модель недодифференцирована.
* *верхний правый график*: остаток имеет распределение, похожее на нормальное распределение, но с большими "хвостами".
* *нижний правый график*: график Q-Q показывает отклонения от нормального распределения.
* *нижний левый график*: автокорреляционная функция не показывает значительных (статистически значимых) отличий от нормального распределения.

Проведенный анализ показывает, что мы можем улучшить нашу модель. При ручном поборе порядка модели следует отдавать предпочтения моделям с наименьшим значением критериев качества и с наименьшей ошибкой RSS.

В качестве первого предположения мы можем попытаться увеличить порядок дифференцирования модели. После можно попробовать несколько значений разных порядков.

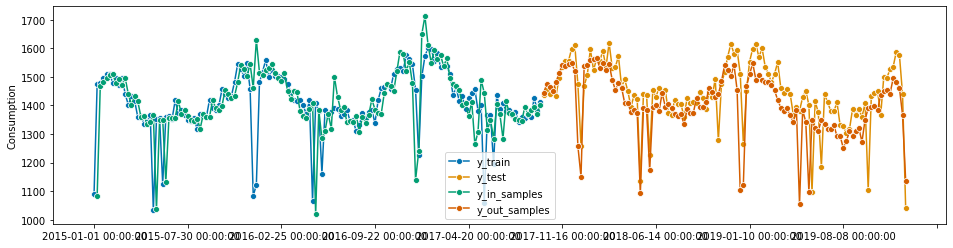
forecaster = SARIMAX(order=(2, 1, 0), seasonal\_order=(2, 1, 0, 52))  
forecaster.fit(y\_train)  
print(forecaster.summary())  
forecaster.\_fitted\_forecaster.plot\_diagnostics(figsize=(12,8));

SARIMAX Results   
==========================================================================================  
Dep. Variable: Consumption No. Observations: 144  
Model: SARIMAX(2, 1, 0)x(2, 1, 0, 52) Log Likelihood -539.528  
Date: Tue, 26 Jul 2022 AIC 1091.057  
Time: 16:39:44 BIC 1106.122  
Sample: 01-01-2015 HQIC 1097.135  
 - 09-28-2017   
Covariance Type: opg   
==============================================================================  
 coef std err z P>|z| [0.025 0.975]  
------------------------------------------------------------------------------  
intercept -3.5024 32.034 -0.109 0.913 -66.288 59.284  
ar.L1 -0.8226 0.071 -11.530 0.000 -0.962 -0.683  
ar.L2 -0.5662 0.079 -7.154 0.000 -0.721 -0.411  
ar.S.L52 -1.0492 4.042 -0.260 0.795 -8.972 6.873  
ar.S.L104 -0.9479 7.669 -0.124 0.902 -15.978 14.082  
sigma2 689.4358 9.86e+04 0.007 0.994 -1.93e+05 1.94e+05  
===================================================================================  
Ljung-Box (L1) (Q): 0.79 Jarque-Bera (JB): 296.03  
Prob(Q): 0.37 Prob(JB): 0.00  
Heteroskedasticity (H): 1.07 Skew: -0.46  
Prob(H) (two-sided): 0.86 Kurtosis: 11.79  
===================================================================================  
  


Теперь давайте попробуем визуализировать предсказания нашей модели.

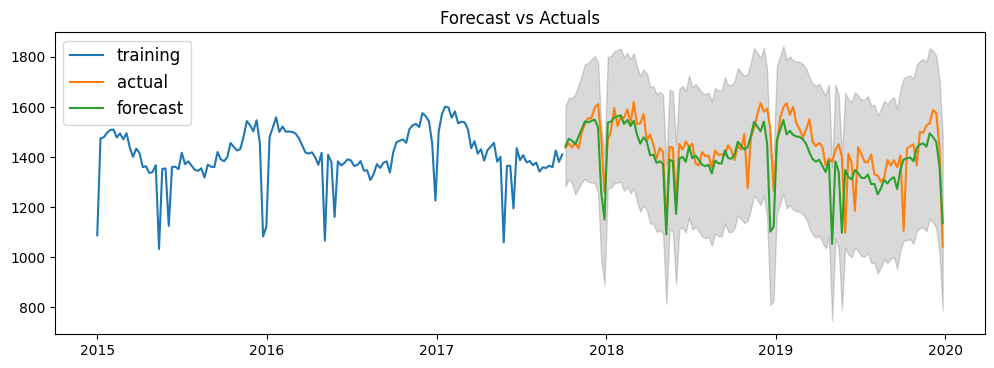
*# forecaster = SARIMAX(order=(2, 1, 0), seasonal\_order=(2, 1, 0, 52))*  
*# forecaster.fit(y\_train)*  
  
fhin = ForecastingHorizon(y\_train.index[1:], is\_relative=False)  
y\_in\_samples = forecaster.predict(fhin)  
  
fhout = ForecastingHorizon(y\_test.index, is\_relative=False)  
y\_out = forecaster.predict(fhout)  
  
plot\_series(y\_train, y\_test, y\_in\_samples, y\_out, labels=["y\_train", "y\_test", "y\_in\_samples", "y\_out\_samples"])  
  
print(f'sMAPE = {smape(y\_out.values, y\_test.values):.3f}')

sMAPE = 0.049



Полученный результаты достаточно хорошо описывают модель, однако, рискнем предположить, что можно достичь и лучшей аппроксимации. Однако, также отметим что помимо полученных значений предсказаний важно и предсказание их дисперсии. Давайте попробуем построить такие интервалы.

*# Forecast*  
forecast\_res = forecaster.\_fitted\_forecaster.get\_forecast(y\_test.size, alpha=0.01, dynamic=False) *# 95% conf*  
  
forecast = forecast\_res.predicted\_mean  
  
*# Make as pandas series*  
fc\_series = pd.Series(forecast.values, index=y\_test.index)  
  
lower\_series = pd.Series(forecast\_res.conf\_int()['lower Consumption'], index=y\_test.index)  
upper\_series = pd.Series(forecast\_res.conf\_int()['upper Consumption'], index=y\_test.index)  
  
*# Plot*  
plt.figure(figsize=(12,4), dpi=100)  
  
plt.plot(y\_train, label='training')  
plt.plot(y\_test, label='actual')  
plt.plot(fc\_series, label='forecast')  
  
plt.fill\_between(lower\_series.index,   
 lower\_series,   
 upper\_series,   
 color='k',   
 alpha=0.15)  
  
plt.title('Forecast vs Actuals')  
plt.legend(loc='upper left', fontsize=12)  
plt.show()



Во многих случаях, помимо ручного поиска параметров ARMA могут быть использованы инструменты автопоиска. Мы рассмотрим один из таких инструментов в рамках пакета SKTime это AutoARIMA. Данный класс основан функциях из библиотеки [pmdarima](http://alkaline-ml.com/pmdarima/).

Объект класса AutoARIMA позволяет задать границы поиска параметров моделей, а также метод тестирования и ряд других параметров. В качестве примера давайте попробуем найти автоматическое предложение параметров.

model = AutoARIMA(start\_p=1, *# начальный порядок AR*  
 d=1, *# Порядок производной*  
 start\_q=0, *# начальный порядок MA*  
 max\_p=5, *# конечный порядок AR*  
 max\_q=5, *# конечный порядок MA*   
 seasonal=True, *# Использовать SARIMA*   
 start\_P=0, *# начальный порядок SAR*  
 start\_Q=0, *# начальный порядок SMA*   
 D=1, *# Порядок сезонной производной*  
 sp=52, *# Период сезонности*  
 max\_order = 7, *# Максимальный порядок p+q+P+Q*   
 trace = True, *# отчет он-лайн*  
 stepwise = True, *# метод ускоренного выбора параметров.*  
 n\_jobs = 1, *# для stepwise парралелизм не доступен.*  
 error\_action='ignore',   
 suppress\_warnings=True)  
  
model.fit(y\_train)  
  
model.summary()

Performing stepwise search to minimize aic  
 ARIMA(1,1,0)(0,1,0)[52] intercept : AIC=1125.065, Time=0.44 sec  
 ARIMA(0,1,0)(0,1,0)[52] intercept : AIC=1158.761, Time=0.33 sec  
 ARIMA(1,1,0)(1,1,0)[52] intercept : AIC=1121.163, Time=5.92 sec  
 ARIMA(0,1,1)(0,1,1)[52] intercept : AIC=inf, Time=5.78 sec  
 ARIMA(0,1,0)(0,1,0)[52] : AIC=1156.762, Time=0.43 sec  
 ARIMA(1,1,0)(2,1,0)[52] intercept : AIC=inf, Time=30.73 sec  
 ARIMA(1,1,0)(1,1,1)[52] intercept : AIC=inf, Time=16.04 sec  
 ARIMA(1,1,0)(0,1,1)[52] intercept : AIC=1121.129, Time=5.61 sec  
 ARIMA(1,1,0)(0,1,2)[52] intercept : AIC=inf, Time=26.87 sec  
 ARIMA(1,1,0)(1,1,2)[52] intercept : AIC=inf, Time=29.83 sec  
 ARIMA(0,1,0)(0,1,1)[52] intercept : AIC=inf, Time=7.44 sec  
 ARIMA(2,1,0)(0,1,1)[52] intercept : AIC=inf, Time=13.07 sec  
 ARIMA(1,1,1)(0,1,1)[52] intercept : AIC=inf, Time=6.32 sec  
 ARIMA(2,1,1)(0,1,1)[52] intercept : AIC=inf, Time=23.75 sec  
 ARIMA(1,1,0)(0,1,1)[52] : AIC=1119.137, Time=5.97 sec  
 ARIMA(1,1,0)(0,1,0)[52] : AIC=1123.066, Time=0.28 sec  
 ARIMA(1,1,0)(1,1,1)[52] : AIC=inf, Time=15.92 sec  
 ARIMA(1,1,0)(0,1,2)[52] : AIC=inf, Time=25.07 sec  
 ARIMA(1,1,0)(1,1,0)[52] : AIC=1119.171, Time=3.58 sec  
 ARIMA(1,1,0)(1,1,2)[52] : AIC=inf, Time=31.34 sec  
 ARIMA(0,1,0)(0,1,1)[52] : AIC=inf, Time=7.13 sec  
 ARIMA(2,1,0)(0,1,1)[52] : AIC=inf, Time=13.10 sec  
 ARIMA(1,1,1)(0,1,1)[52] : AIC=inf, Time=5.77 sec  
 ARIMA(0,1,1)(0,1,1)[52] : AIC=inf, Time=4.61 sec  
 ARIMA(2,1,1)(0,1,1)[52] : AIC=inf, Time=19.15 sec  
  
Best model: ARIMA(1,1,0)(0,1,1)[52]   
Total fit time: 304.797 seconds

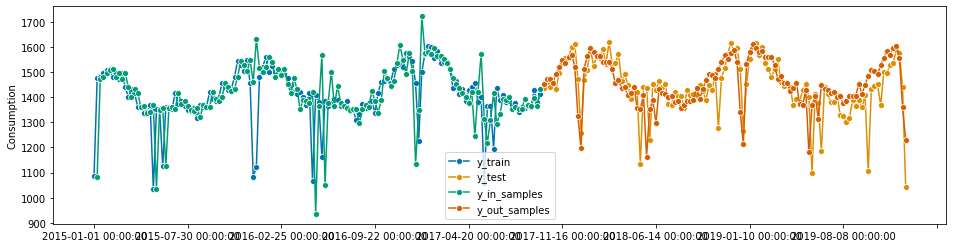
<class 'statsmodels.iolib.summary.Summary'>  
"""  
 SARIMAX Results   
============================================================================================  
Dep. Variable: y No. Observations: 144  
Model: SARIMAX(1, 1, 0)x(0, 1, [1], 52) Log Likelihood -556.568  
Date: Tue, 26 Jul 2022 AIC 1119.137  
Time: 16:44:51 BIC 1126.669  
Sample: 0 HQIC 1122.175  
 - 144   
Covariance Type: opg   
==============================================================================  
 coef std err z P>|z| [0.025 0.975]  
------------------------------------------------------------------------------  
ar.L1 -0.5346 0.051 -10.555 0.000 -0.634 -0.435  
ma.S.L52 -0.4727 0.133 -3.541 0.000 -0.734 -0.211  
sigma2 1.054e+04 1221.051 8.628 0.000 8142.195 1.29e+04  
===================================================================================  
Ljung-Box (L1) (Q): 8.38 Jarque-Bera (JB): 133.68  
Prob(Q): 0.00 Prob(JB): 0.00  
Heteroskedasticity (H): 0.43 Skew: 0.23  
Prob(H) (two-sided): 0.02 Kurtosis: 8.92  
===================================================================================  
  
"""

И так протестируем модель, выбранную автоматически.

fhin = ForecastingHorizon(y\_train.index[1:], is\_relative=False)  
y\_in\_samples = model.predict(fhin)  
  
fhout = ForecastingHorizon(y\_test.index, is\_relative=False)  
y\_out = model.predict(fhout)  
  
plot\_series(y\_train, y\_test, y\_in\_samples, y\_out, labels=["y\_train", "y\_test", "y\_in\_samples", "y\_out\_samples"])  
  
print(f'sMAPE = {smape(y\_out.values, y\_test.values):.3f}')

c:\users\ronkin\appdata\local\programs\python\python37\lib\site-packages\pmdarima\arima\arima.py:692: UserWarning: As of version 1.5.0 'typ' is no longer a valid arg for predict. In future versions this will raise a TypeError.  
 warnings.warn("As of version 1.5.0 'typ' is no longer a valid "

sMAPE = 0.042

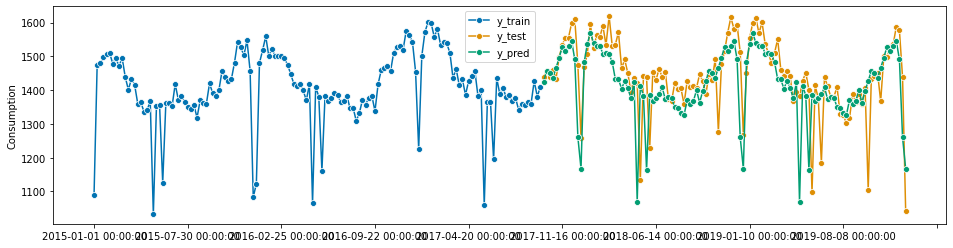


Полученная модель оказалась лучше найденной вручную. Однако, подчеркнем, что на практике такое не всегда бывает. В частности, можно заметить, что критерий AIC для модели, подобранной вручную был несколько ниже, чем для проверенных автоматически. Часто, авто поиск модели можно использовать лишь как некоторое очень хорошее начальной предположение.

Также отметим, что в рамках пакета SKTime доступны и другие методы работы с ARMA моделями. Например, вместо поиска производных можно провести предварительное удаление сезонности данных.

**from** sktime.forecasting.arima **import** ARIMA  
**from** sktime.forecasting.compose **import** TransformedTargetForecaster  
**from** sktime.transformations.series.detrend **import** Deseasonalizer  
  
forecaster = TransformedTargetForecaster(  
 [  
 ("deseasonalize", Deseasonalizer(model="multiplicative", sp=52)),  
 ("forecast", ARIMA( order=(2, 0, 0), seasonal\_order=(0, 0, 0, 0), )),  
 ]  
)  
  
forecaster.fit(y\_train)  
  
fh = ForecastingHorizon(y\_test.index, is\_relative=False)  
  
y\_pred = forecaster.predict(fh)  
  
plot\_series(y\_train, y\_test, y\_pred, labels=["y\_train", "y\_test", "y\_pred"])  
  
print(f'sMAPE = {smape(y\_pred.values, y\_test.values):.3f}')

sMAPE = 0.047



Точность такого результата оказывается несколько ниже, однако, скорость обучения модели значительно увеличивается. В дополнение к этому такой подход позволяет проводить раздельно учитывать несколько сезонных составляющих в данных.

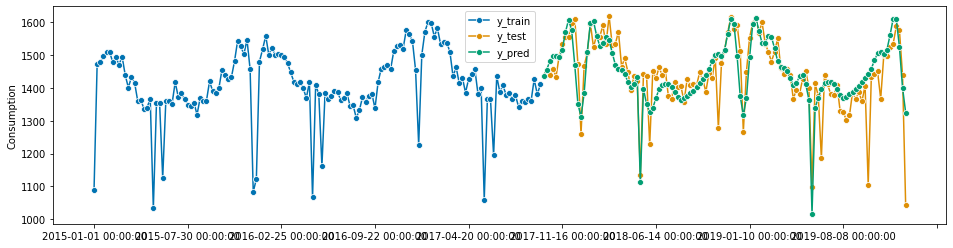
На сегодняшний день семейство моделей ARIMA является одним из основных методов предсказания временных рядов. Метод отлично подходит для однопеменных временных рядов. Метод хорошо работает как для рядов небольшой длительности, так и для достаточно длинных рядов. При этом метод подходит как для детерминированных, так и для случайных типа тренд. А также, само по себе описание ряда при помощи коэффициентов может быть важным признаком в ряде методов машинного обучения.

Также отметим, что часто метод ARIMA сравнивается c алгоритмом Prophet, который в свою очередь представляет собой некоторую адоптацию обобщенной регрессии к бизнесс процессом. Алгоритм Prophet в некоторых случаях может давать более точные предсказания, но не всегда. Давайте сравним результаты, полученные методами ARIMA и Prophet для нашего примера.

**from** sktime.forecasting.fbprophet **import** Prophet  
  
forecaster = Prophet(freq='1w',  
 seasonality\_mode='additive',  
 add\_country\_holidays={'country\_name': 'Germany'},  
 yearly\_seasonality=True)  
  
forecaster.fit(y\_train)  
fh = ForecastingHorizon(y\_test.index, is\_relative=False)  
y\_pred = forecaster.predict(fh)  
plot\_series(y\_train, y\_test, y\_pred, labels=["y\_train", "y\_test", "y\_pred"])  
print(f'sMAPE = {smape(y\_pred.values, y\_test.values):.3f}')

16:44:57 - cmdstanpy - INFO - Chain [1] start processing  
16:45:00 - cmdstanpy - INFO - Chain [1] done processing

sMAPE = 0.033



После правильной настройки метод алгоритм Prophet оказался лучше. Еще раз отметим, что в каждом случае следует выбирать свой вариант предсказателя.

# Словарь терминов и определений

**Модель авторегрессии** - скользящего среднего (АРСС, ARMA) – это модель которую можно проинтерпретировать как: текущее значение ВР зависит от прошлых значений до лага p (AR часть) и от текущего и прошлых внешних «возмущений» (флуктуаций) до лага q (MA часть). Это параметрическая модель c двумя параметрами (p,q).

**Задача ARMA-аппроксимации** (построения АРСС модели) - найти весовые коэффициенты и и оценить их порядки (p,q), приближающие лучше всего к исходному процессу

**Лаг модели** – это значение задержки рассматриваемых отсчетов относительно заданного 0.

**Начальные значения порядков ARMA моделей** могут быть выбраны из анализа графиков автокорреляционной функции (АКФ, ACF) – MA часть и частичной АКФ (ЧАКФ, PACF) – AR часть как значения лагов перед резким спадом к нулю в обоих случаях. Однако это справедливо только для стационарной в слабом смысле модели. Тестирование Лагов АКФ на значимость выполняется при помощистатистических тестов.

**Оптимизация ARMA** **моделей** заключается в настройке их порядков и соответствующих им коэффициентов. Оно осуществляется при помощи информационных критериев и последующего тестирования модели. При этом чем ниже общий порядок модели – тем ниже вероятность переобучения.

**Информационный критерий** – критерий, учитывающий, как дисперсию остаточной части модели, так и порядки модели. Критерий имеет исключительно смысл в ранжировании (используется только для сравнения моделей, чем меньше значение, тем лучше). Критерий позволяет задать мини-максуную задачу поиска параметров моделей: минимум ошибки при максимально-допустимом числе параметров модели.

Модель называется **интегрированной авторегрессионной скользящей средней** (ARIMA) – модель нативно учитывающая численное дифференцирование для достижения стационарности результат моделирования.

**Модель ARIMA имеет три порядка ()**, где **d** – порядок численной производной (разности). Модель удобно записать в лаговой форме. Важно понимать, что дифференцированию подлежит только AR часть. В первую очередь такая модель может компенсировать нестационарность тренда или другие медленные изменения. При выборе порядков модели сначала должен быть найден порядок дифференцирования.

**Для проверки модели на стационарность используются ряд визуальных и статистических тестов**. Среди них:

· Статистика скользящего окна (Rolling Statistics)

· ACF анализ

· Статистические тесты, например Расширенный тест Дики-Фуллера (Augmented Dickey-Fuller Test, ADF)

**Модель сезонной интегрированной авторегресси - скользящего среднего (SARIMA)** - это модель нативно учитывающая даже сравнительно быстрые и интенсивные, но регулярные нестационарности типа сезонность. Для этого испльзуется прием сезонного дифференцирования.

**Модель имеет задание порядков** , включающая как обычные , так и сезонные составляющие для сезонности с периодом

При выборе начальных значений порядков модели следует сначала определить порядки обычной и сезонной производной, приводящие ВР к стационарному виду.

**Экзогенные факторы** (exdogs, экзогенные ковариаты) или набор факторов являются дополнительными факторам к целевой переменной , которые статистически не зависят от , но влияют на нее. Учет таких факторов может быть произведен в моделях SARIMA, такая форма называется SARIMAX.

Другими модификациями моделей класса ARIMA являются ее расширения, например, ансамблирование или конвейеризация. Однако, эти термины мы употребляем в более широком смысле, чем это известно читателю. Типичные приемы тут могут быть следующие:

* **Предварительное преобразование ВР**, например, приведение ВР к стационарному виду при помощи вычитания среднего значения или преобразования Бокса-Кокса. А также такие приемы, как очистка данных от аномалий или разделение на сегменты, каждый из которых по мнению исследователя лучше исследовать отдельно.
* **Разложение временного ряда на составляющие**. Например, построение отдельных моделей для тренда и сезонных составляющих. Таких составляющих может быть несколько, и каждую лучше учитывать отдельно. Этот прием можно **ансамблированием** в некотором смысле.
* Использование последовательности нескольких моделей таким образом, что каждая следующая обрабатывает остаток от работы предыдущей. Этот прием можно назвать **конвейеризацией** в некотором смысле.
* **Расширения понятия модели** (например) ARIMA путем использования альтернативных оптимизаторов. Например, замена решения регрессии методом наименьших квадратов на другие оптимизацию методом градиентного спуска для других функций потерь.

# Заключение

Изучен модуль продвинутых методов предсказания. В модуле рассмотрен пример одного из наиболее популярных методов предсказания моделей класса авторегрессия-скользящее среднее. Данный метод част является базовым (т.н. baseline) для решения проблем анализа временных рядов. Показаны, подходы к модификации данного метода и варианты расширенной работы с ним. Также приведены примеры работы с методом в пакете SKTime. Метод часто позволяет достичь сравнительно высоких точностей на рабочих ВР, однако требует тщательно подбора своих параметров. Подбор может быть частично автоматизирован, однако всегда стоит оставлять время на «докрутку» параметров под тестовые данные. Также мы показали, что в ряде случаев, когда известна непараметрическая модель данных – она может быть более точным методом работы. Но в целом такая ситуация далеко не всегда имеет место.

**Дополнительная литература**

1. A Gentle Introduction to Autocorrelation and Partial Autocorrelation / machinelearningmastery.com — [Электронный ресурс]. URL: https://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/ (дата обращения: 10.06.2023).
2. <https://otexts.com/fpp3/> - Forecasting: Principles and Practice. Полезная практико-ориентированная книга по предсказаниям ВР (2ая версия книги тут <https://otexts.com/fpp2/>)
3. <https://forecasting.svetunkov.ru/> - он лайн учебник по прогнозированию на R
4. <https://github.com/jiwidi/time-series-forecasting-with-python> - примеры методов предсказания временных рядов
5. <https://phdinds-aim.github.io/time_series_handbook/> - справочник по анализу временных рядов
6. <https://www.sktime.net/en/stable/ Welcome to sktime — sktime documentation>.