

Пропуски, аномалии и структурные сдвиги

Заполнение пропусков

Заполнение пропусков: план

- Линейная интерполяция.

Заполнение пропусков: план

- Линейная интерполяция.
- Модели для заполнения пропусков.

Заполнение пропусков: план

- Линейная интерполяция.
- Модели для заполнения пропусков.
- Использование STL-разложения.

Линейная интерполяция

Идея

Заполним пропуски так, чтобы восстановленные значения идеально ложились на прямую (образовывали арифметическую прогрессию),

$$\Delta y_t^{imp} = const.$$

Линейная интерполяция

Идея

Заполним пропуски так, чтобы восстановленные значения идеально ложились на прямую (образовывали арифметическую прогрессию),

$$\Delta y_t^{imp} = const.$$

Пример:

10, NA, NA, 100.

Линейная интерполяция

Идея

Заполним пропуски так, чтобы восстановленные значения идеально ложились на прямую (образовывали арифметическую прогрессию),

$$\Delta y_t^{imp} = const.$$

Пример:

10, NA, NA, 100.

10, 40, 70, 100

Модели для заполнения пропусков

1. Оцениваем модель, **допускающую** пропуски в данных.
ARIMA подходит! И автоматическая ARIMA тоже!

Модели для заполнения пропусков

1. Оцениваем модель, **допускающую** пропуски в данных. ARIMA подходит! И автоматическая ARIMA тоже!
2. Пропущенные значения y_t заменяем на условное математическое ожидание, полагая оценённые параметры модели равным истинными,

$$y_t^{imp} = \mathbb{E}(y_t \mid \text{данные}).$$

Используется **фильтр Калмана**.

Модели для заполнения пропусков

1. Оцениваем модель, **допускающую** пропуски в данных. ARIMA подходит! И автоматическая ARIMA тоже!
2. Пропущенные значения y_t заменяем на условное математическое ожидание, полагая оценённые параметры модели равным истинными,

$$y_t^{imp} = \mathbb{E}(y_t \mid \text{данные}).$$

Используется **фильтр Калмана**.

Модели для заполнения пропусков

1. Оцениваем модель, **допускающую** пропуски в данных. ARIMA подходит! И автоматическая ARIMA тоже!
2. Пропущенные значения y_t заменяем на условное математическое ожидание, полагая оценённые параметры модели равным истинными,

$$y_t^{imp} = \mathbb{E}(y_t \mid \text{данные}).$$

Используется **фильтр Калмана**.

Возможность оценивать модель на данных с пропусками сильно зависит от **реализации**.

Использование STL-разложения

1. Раскладываем ряд с пропусками на составляющие:

$$y_t = \text{trend}_t + \text{seasonal}_t + \text{remainder}_t = \text{seasonal}_t + \text{deseason}_t.$$

STL восстанавливает **сезонную компоненту** без пропусков!

Использование STL-разложения

1. Раскладываем ряд с пропусками на составляющие:

$$y_t = \text{trend}_t + \text{seasonal}_t + \text{remainder}_t = \text{seasonal}_t + \text{deseason}_t.$$

STL восстанавливает **сезонную компоненту** без пропусков!

2. Восстанавливаем пропущенные значения десезонированного ряда **линейной** интерполяцией.

Использование STL-разложения

1. Раскладываем ряд с пропусками на составляющие:

$$y_t = \text{trend}_t + \text{seasonal}_t + \text{remainder}_t = \text{seasonal}_t + \text{deseason}_t.$$

STL восстанавливает **сезонную компоненту** без пропусков!

2. Восстанавливаем пропущенные значения десезонированного ряда **линейной** интерполяцией.
3. Пропущенные значения y_t заменяем на сумму восстановленных десезонированных значений и сезонной составляющей,

$$y_t^{imp} = \text{seasonal}_t + \text{deseason}_t^{imp}.$$

Зачем заполнять пропуски?

- Иногда заполнение пропусков — основная задача.

Зачем заполнять пропуски?

- Иногда заполнение пропусков — основная задача.
- Возможность использовать больше алгоритмов прогнозирования для восстановленного ряда.

Зачем заполнять пропуски?

- Иногда заполнение пропусков — основная задача.
- Возможность использовать больше алгоритмов прогнозирования для восстановленного ряда.
- Возможность использовать восстановленный ряд как предиктор.

Заполнение пропусков: итоги

- Линейная интерполяция: просто и быстро!

Заполнение пропусков: итоги

- Линейная интерполяция: просто и быстро!
- Использование ARIMA или более сложных моделей.

Заполнение пропусков: итоги

- Линейная **интерполяция**: просто и быстро!
- Использование **ARIMA** или более сложных моделей.
- **STL-разложение** и восстановление компонент.

Заполнение пропусков: итоги

- Линейная **интерполяция**: просто и быстро!
- Использование **ARIMA** или более сложных моделей.
- **STL-разложение** и восстановление компонент.
- **Вариации** у каждого алгоритма.