

Национальный исследовательский университет
”ВЫСШАЯ ШКОЛА ЭКОНОМИКИ”
Факультет экономических наук

Курсовая работа

Прогнозирование временных рядов
с помощью модели состояние-наблюдение

Выполнила
студентка группы БЭК 151
Гнилова О.А.

Научный руководитель
старший преподаватель
департамента прикладной экономики
Демешев Б.Б.

Москва, 2017.

Содержание

1	Введение	2
2	Описание данных	3
3	Построение моделей одномерных рядов	6
3.1	Моделирование ошибкой – Irregular	6
3.2	Моделирование константой – Deterministic constant	11
3.3	Модель локального уровня – Local level	12
3.4	Случайное блуждание – Random walk	14
3.5	Постоянный тренд – Deterministic trend	15
3.6	Локальный линейный постоянный тренд – Local linear deterministic trend	16
3.7	Случайное блуждание со смещением – Random walk with drift	16
3.8	Локальный линейный тренд – Local linear trend	17
3.9	Сглаженный тренд – Smoothed trend	18
3.10	Случайный тренд – Random trend	19
3.11	Локальный линейный тренд с сезонностью	19
3.12	Локальный линейный тренд с сезонностью и регрессорами	21
4	Построение модели многомерных рядов	24
4.1	Описание модели	24
4.2	Пример: модель локального линейного уровня с регрессорами	24
4.3	Общие уровни и наклоны	26
5	Заключение	27
6	Список источников	28

1 Введение

В данной работе исследуется модель наблюдение-состояние (state space model), используемая в прогнозировании временных рядов. Её суть состоит в представлении наблюдаемого временного ряда в виде нескольких ненаблюдаемых компонент: тренда, сезонности и ошибки наблюдений, каждая из которых, в свою очередь, может моделироваться не единственным образом. Как правило, в модели предполагается ещё одна составляющая – цикличность, однако из-за специфики выбранных мною данных она рассмотрена не будет.

Цель работы – подробно описать возможные способы моделирования временного ряда, начиная с простых прогнозов, когда в уравнение наблюдаемой компоненты входит только погрешность наблюдений, до более сложных, когда в модель входят все перечисленные ненаблюдаемые компоненты и регрессоры, и реализовать их на языке Python с использованием библиотеки `statsmodels`. Также в работе описана модель локального линейного тренда с регрессорами в форме состояние-наблюдение для случая многомерных временных рядов.

2 Описание данных

Временные ряды для этой работы были взяты из соревнования [Rossmann Store Sales](#) на Kaggle и содержат информацию по продажам 1115 магазинов сети Rossmann за период 01-01-2013 по 31-07-2015, то есть за два с половиной года. Поскольку наиболее подробно мною изучались модели прогнозирования одномерных временных рядов, подопытным рядом стали продажи магазина с `id=1`. Посмотрим, как он выглядит:

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

data = pd.read_csv('train.csv')
data.index = pd.to_datetime(data.Date)
data[data.Store==1].sort_values('Date').Sales.plot(marker='.',
                                                    figsize=(20,6))

plt.title('Rossmann Sales, Store 1')
plt.ylabel('Sales')
```

```
Out[1]: <matplotlib.text.Text at 0x1115185f8>
```

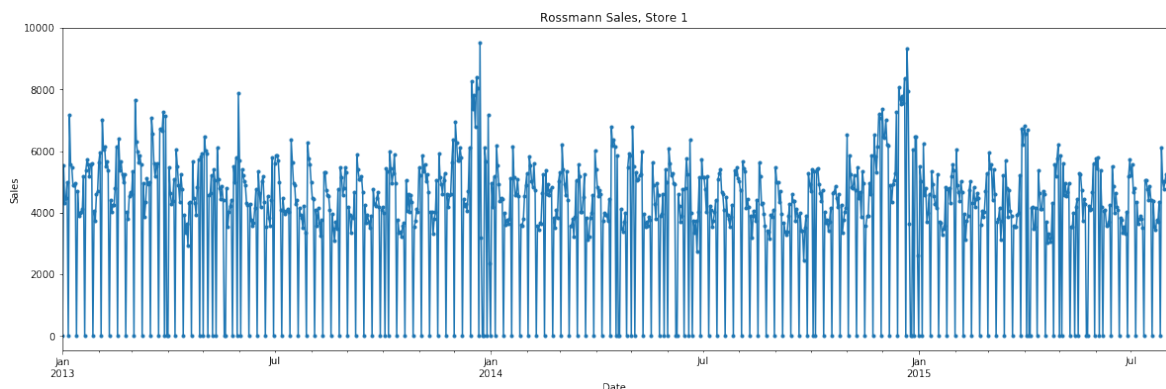


Рис. 1: Продажи в магазине 1

Из-за выходных и праздников, когда магазин закрыт и продаж нет, график выглядит очень зашумлённым. Более того, эти выбросы сильно влияют на результаты моделей, описанных мной далее, делают их менее наглядными. По этой причине я решила на время убрать нулевые значения из ряда, тем более, что они подчиняются понятному паттерну и их легко можно будет восстановить. Однако убрав их, я всё же не стану сохранять исходную последовательность дат, поскольку мне важны качественные результаты моделей, а не сами предсказанные значения. Работать же с исходной последовательностью дат не получается из-за того, что библиотека `statsmodels` не может давать прогноз на будущее для "необычных" дат: функция `get_forecast` не берёт

аргумент `freq`, задающий периодичность дней, которая здесь нестандартна: исключается каждое седьмое значение (те, что соответствуют воскресеньям), а также праздники.

Осуществим задуманное и посмотрим на график снова:

```
In [2]: dta = data[(data.Store==1)
                & (data.Open==1)].sort_values('Date').Sales

fig, ax = plt.subplots(figsize=(18,4))

ax.plot(dta)
ax.set_xlabel('Date')
ax.set_ylabel('Sales')
ax.set_title('Rossmann Sales, Store 1')
```

```
Out[2]: <matplotlib.text.Text at 0x112c54828>
```

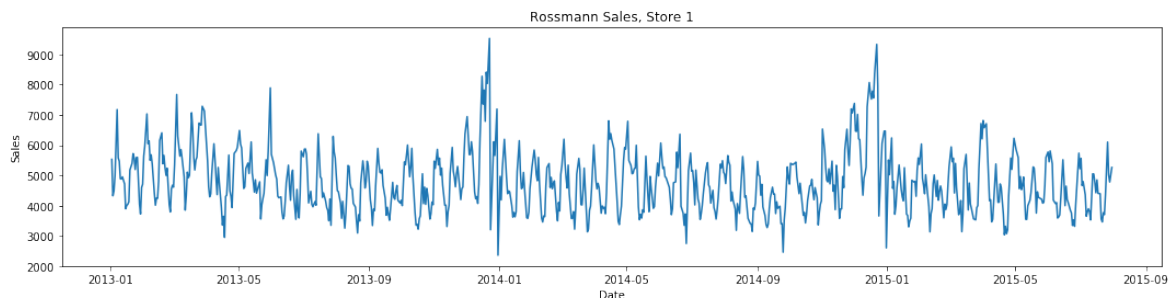


Рис. 2: Ненулевые продажи в магазине 1

Теперь, когда ряд стал выглядеть лучше, можно сделать первые выводы о нём: в данных нет явного тренда, в декабре каждого года продажи резко увеличиваются, а амплитуда колебаний приблизительно постоянна.

Предположения о тренде и колебаниях можно сразу проверить: для этого необходимо установить, является ли ряд стационарным. Ответ будет положительным, если ни математическое ожидание, ни дисперсия его не зависят от времени. Проверка осуществляется с помощью теста Дики-Фулера, в котором нулевая гипотеза соответствует предположению о том, что ряд не стационарен.

```
In [3]: import statsmodels.api as sm

result = sm.tsa.stattools.adfuller(dta)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))
```

ADF Statistic: -5.399623
p-value: 0.000003
Critical Values:
10%: -2.569
5%: -2.865
1%: -3.439

Выходит, что на любом разумном уровне значимости, гипотеза отвергается, а значит, ряд является стационарным. И теперь, когда про ряд стало известно больше, можно переходить к построению моделей.

3 Построение моделей одномерных рядов

Под моделью временного ряда для наблюдений $\{y_t\}$ будем понимать описание совместных распределений последовательности случайных величин $\{Y_t\}$, которые принимают значения $\{y_t\}$

3.1 Моделирование ошибкой – Irregular

Самая простая модель для прогнозирования временных рядов: в ней нет ни тренда, ни сезонности, а наблюдения считаются независимыми одинаково распределёнными случайными величинами Y_1, Y_2, \dots с нулевым математическим ожиданием:

$$y_t = \varepsilon_t$$
$$\varepsilon_t \text{ iid: } \varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2)$$

Выясним, какой прогноз будет давать эта модель. По определению, можем записать:

$$P[Y_1 \leq y_1, \dots, Y_n \leq y_n] = P[Y_1 \leq y_1] \cdot \dots \cdot P[Y_n \leq y_n] = F(y_1) \cdot \dots \cdot F(y_n)$$

где $F(\cdot)$ – функция распределения случайных величин Y_1, Y_2, \dots . Поскольку в модели предполагается, что наблюдения независимы, то для всех $h \geq 1$ и для всех y_1, \dots, y_n :

$$P[Y_{n+h} \leq y \mid Y_1 = y_1, \dots, Y_n = y_n] = P[Y_{n+h} \leq y]$$

Другими словами, предыдущие наблюдения не несут никакой информации о следующем наблюдении Y_{n+h} . При известных значениях Y_1, Y_2, \dots функция f , которая минимизирует среднеквадратичную ошибку $E[(Y_{n+h} - f(Y_1, \dots, Y_n))^2]$, равна нулю, покажем:

$$\begin{aligned} E[(Y_{n+h} - f(Y_1, \dots, Y_n))^2] &= E[Y_{n+h}^2 - 2 \cdot Y_{n+h} \cdot f(Y_1, \dots, Y_n) + f(Y_1, \dots, Y_n)^2] = \\ &= E[Y_{n+h}^2] - 2 \cdot f(Y_1, \dots, Y_n) \cdot E[Y_{n+h}] + f(Y_1, \dots, Y_n)^2 \rightarrow \min_f \end{aligned}$$

Так как $Y_{n+h} \sim \mathcal{N}(0, \sigma_\varepsilon^2)$, то $E[Y_{n+h}] = 0$ и задача сводится к минимизации следующей функции: $E[Y_{n+h}^2] + f(Y_1, \dots, Y_n)^2 \rightarrow \min_f$, а её минимум достигается, очевидно, при $f(Y_1, \dots, Y_n) = 0$.

Обучим модель и убедимся в том, что она будет давать только нулевые значения. В модуле `UnobservedComponents` библиотеки `statsmodels` необходимо указать все параметры модели: в данном примере единственным параметром является ошибка, поэтому прописываем аргумент `irregular=True`. Остальные модели можно настраивать схожим образом, или же передавать параметру `level` сокращённое название модели, указанное в [документации](#). В таком случае все остальные параметры из-

меняться на нужные автоматически, в чём можно будет убедиться, вызвав функцию `model_name.fit().specification`.

```
In [4]: model_1 = statsmodels.tsa.statespace.structural. \
        UnobservedComponents(endog=dta, level=False,
                               trend=False, seasonal=None,
                               cycle=False, autoregressive=None,
                               exog=None, irregular=True,
                               stochastic_level=False,
                               stochastic_trend=False,
                               stochastic_seasonal=False,
                               stochastic_cycle=False,
                               damped_cycle=False,
                               cycle_period_bounds=None)
```

Выведем результаты модели:

```
In [5]: res_model_1 = model_1.fit()
        res_model_1.summary()
```

Out[5]: Unobserved components results

Dep. Variable:	Sales	No. Observations:	781
Model:	irregular	Log Likelihood	-7739
Date:	Sun, 21 May 2017	AIC	15480
Time:	14:11:26	BIC	15485
Sample:	01-01-2013	HQIC	15482
	- 02-20-2015		
Covariance Type:	opg		

Таблица 1: Описание модели Irregular

	coef	std err	z	P> z 	[0.025	0.975]
sigma2.irregular	$2.30 \cdot 10^7$	$3.53 \cdot 10^6$	6.52	0	$1.61 \cdot 10^7$	$2.99 \cdot 10^7$

Таблица 2: Оценка параметров модели Irregular

Ljung-Box (Q):	1255	Jarque-Bera (JB):	190.5
Prob(Q):	0.00	Prob(JB):	0.00
Heteroskedasticity (H):	0.98	Skew:	0.93
Prob(H) (two-sided):	0.88	Kurtosis:	4.54

Таблица 3: Информация о данных в модели Irregular

Как видно из таблиц, библиотека `statsmodels` выдаёт очень подробную информацию о модели. В частности, таблица 1 позволяет определять качество модели и сравнивать их между собой. AIC – информационный критерий Акаике – устроен на основе функции максимального правдоподобия и записывается следующим образом:

$$AIC = 2k - 2 \ln(L)$$

где k – количество параметров в модели, а L – максимум функции правдоподобия. Критерий вознаграждает за сходство результатов с истинными данными и штрафует за большое количество параметров, так что чем меньше его значение, тем модель лучше.

BIC – байесовский информационный критерий – модификация предыдущего критерия, основанная на байесовском подходе:

$$BIC = k \ln n - 2 \ln(L)$$

где n – объём выборки, а остальные параметры те же. Этот критерий сильнее штрафует за большое количество параметров.

И последний подсчитываемый критерий HQIC – это информационный критерий Ханнана-Куинна.

$$HQIC = 2k \ln \ln n/n - 2 \ln(L)/n$$

В таблице 3 находится информация о случайности данных. Так, в ней есть значение статистики Q-теста Льюнг-Бокса, формула для которого имеет вид:

$$\tilde{Q} = n(n+2) \sum_{k=1}^m \frac{\hat{\rho}_k^2}{n-k} \sim \chi_m^2$$

где n – число наблюдений, $\hat{\rho}_k^2$ – автокорреляция k -ого порядка, m – количество лагов. В основной гипотезе предполагается, что данные являются случайными (белым шумом).

В таблице также представлены результаты теста Харке-Бера, который проверяет ошибки наблюдений на нормальность, сравнивая выборочные третий (S от skew, асимметрия) и четвёртый (K от kurtosis, эксцесс) моменты с соответствующими моментами нормального распределения, для которого $S = 0$, $K = 3$. Нулевая гипотеза в тесте так и формулируется: $\mathcal{H}_0 : S = 0, K = 3$, а статистика имеет вид:

$$JB = n \left(\frac{S^2}{6} + \frac{(K-3)^2}{24} \right) \sim \chi_2^2$$

где $S = \frac{\sum e_i^3}{n \hat{\sigma}_{ML}^3}$, $K = \frac{\sum e_i^4}{\hat{\sigma}_{ML}^4}$, e_i – остатки модели, n – количество наблюдений, $\hat{\sigma}_{ML}^2 =$

$\frac{\sum e_i^2}{n}$. Чем ближе значение статистики к нулю, тем меньше отличается распределение ошибок от нормального распределения.

И последний значимый результат из таблицы 3 – это гетероскедастичность, то есть степень неоднородности данных, которая проявляется в непостоянстве дисперсий случайных ошибок.

Но всё же самая значимая таблица для данного исследования – это таблица 2, поскольку она содержит оценки неизвестных параметров модели, полученные методом максимального правдоподобия. В рассматриваемой модели параметр один, и это дисперсия ошибки, оценка которой: $\hat{\sigma}_{ML} = 2.302 \cdot 10^7$.

Осталось построить график:

```
In [6]: def plot(res_model, title):
        predict = res_model.get_prediction(start='2014-11-01',
                                           end='2015-02-20')
        forecast = res_model.get_forecast('2015-03-20')

        fig, ax = plt.subplots(figsize=(12,6))

        dta.ix['2014-11-01:'].plot(ax=ax, style='k',
                                  label='Observations')
        predict.predicted_mean.plot(ax=ax,
                                    label='One-step-ahead Prediction')

        predict_ci = predict.conf_int(alpha=0.05)
        ax.fill_between(predict_ci.index[2:], predict_ci.iloc[2:, 0],
                        predict_ci.iloc[2:, 1], alpha=0.1,
                        label='95% confidence interval')

        forecast.predicted_mean.plot(ax=ax, style='--',
                                     label='Forecast')
        forecast_ci = forecast.conf_int()
        ax.fill_between(forecast_ci.index, forecast_ci.iloc[:, 0],
                        forecast_ci.iloc[:, 1], alpha=0.1,
                        label='95% confidence interval')

        legend = ax.legend(loc='lower left')
        ax.set_xlabel('Date')
        ax.set_ylabel('Sales')
        title = title + ' results'
        ax.set_title(title)
```

```
In [7]: plot(res_model_1, 'Irregular model')
```

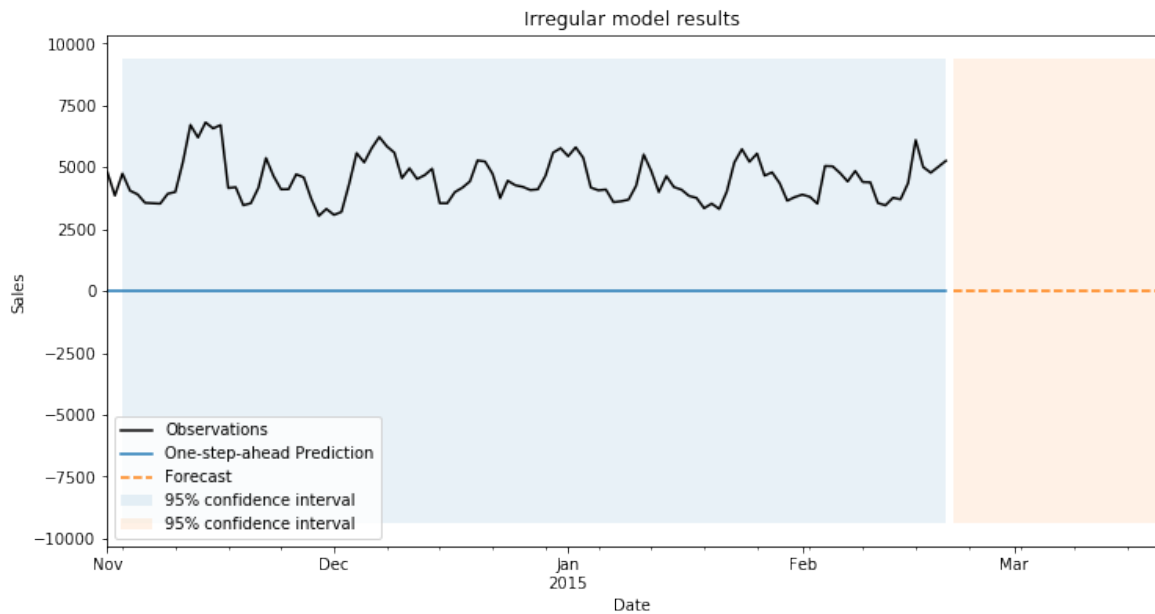


Рис. 3: Прогноз в модели Irregular

Информацию из таблиц частично можно наблюдать на графиках, к чему мы и будем прибегать в дальнейшем. Для этой модели анализ остатков (а тут больше ничего и нет) выглядит так:

```
In [8]: res_model_1.plot_diagnostics(figsize=(8,6))
plt.tight_layout()
```

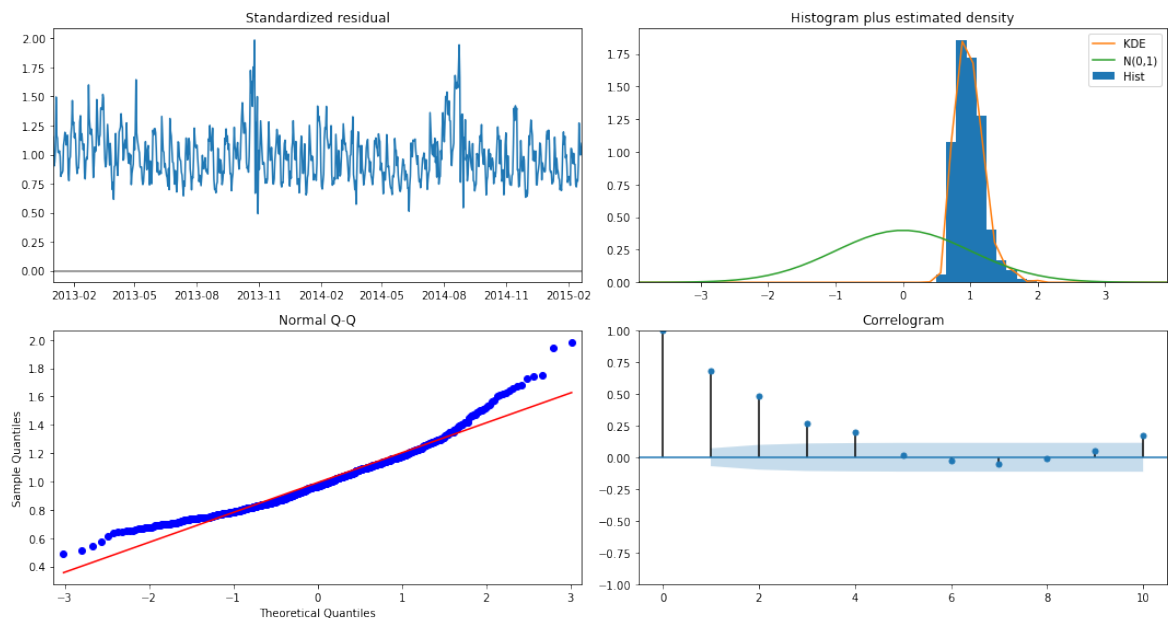


Рис. 4: Анализ остатков в модели Irregular

3.2 Моделирование константой – Deterministic constant

Ещё одна простая модель, в которой наблюдения предполагаются независимыми и одинаково распределёнными случайными величинами равными сумме константы и ошибки:

$$y_t = \mu + \varepsilon_t, \varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2)$$

Сама константа получается из задачи минимизации среднеквадратичной ошибки. Как и в предыдущей модели, наблюдения независимы, поэтому и предсказываемое значение не зависит от предыдущих и задачу можно записать так:

$$\begin{aligned} MSE &= E[(Y_{n+h} - c)^2] = E[Y_{n+h}^2 - 2 \cdot c \cdot Y_{n+h} + c^2] = \\ &= E(Y_{n+h}^2) - 2 \cdot c \cdot E(Y_{n+h}) + c^2 = E(Y_{n+h}^2) - 2 \cdot c \cdot \mu + c^2 \rightarrow \min_c \\ \frac{\partial MSE}{\partial c} &= -2 \cdot \mu + 2 \cdot c \Rightarrow c = \mu \end{aligned}$$

Получили, что искомая константа – это математическое ожидание наблюдений. Значит, прогнозные значения должны быть близки к выборочному среднему в силу большого числа наблюдений. Обучив модель, можем убедиться в этом для значений y_{t+h} , $h \geq 1$:

```
In [9]: print('Выборочное среднее: ', np.mean(dta))
        print('Значение прогноза: ',
              res_model_2.predict(start='2015-02-20').values)
```

Выборочное среднее: 4 759

Значение прогноза: 4 752

Посмотрим, каковы оценки параметров, и визуализируем результат.

	coef	std err	z	P> z	[0.025	0.975]
sigma2.irregular	$1.01 \cdot 10^6$	$4.01 \cdot 10^4$	25.2	0	$9.33 \cdot 10^5$	1.0910^6

Таблица 4: Оценки параметров в константной модели

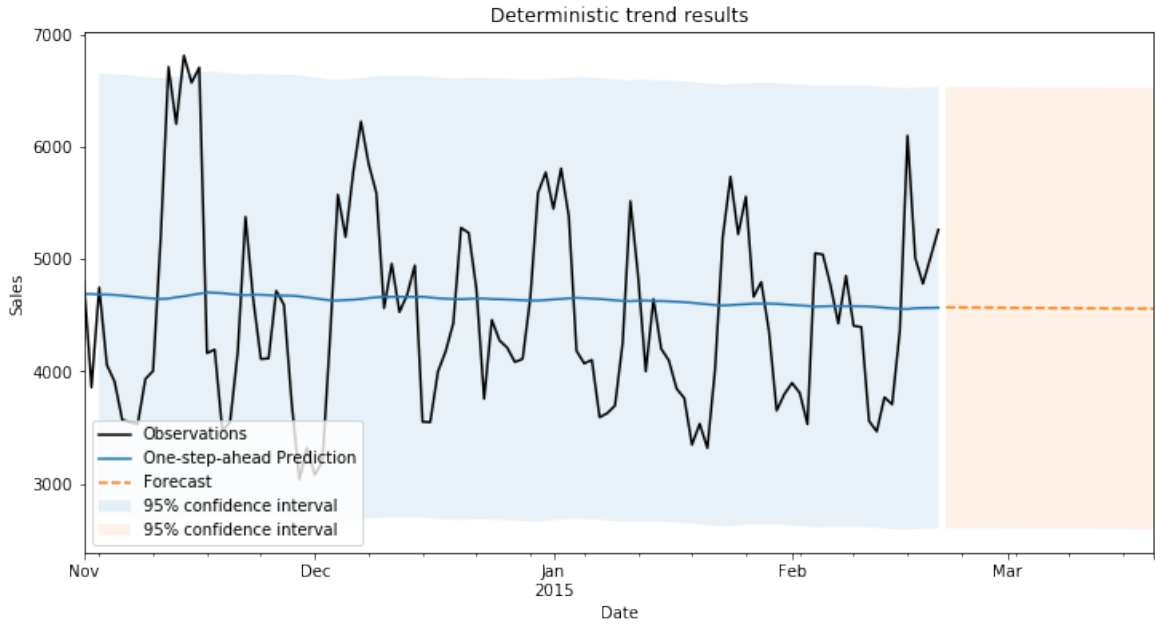


Рис. 5: Прогноз в константной модели

Каждую из составляющих частей модели тоже можно визуализировать.

```
In [10]: res_model_2.plot_components(figsize=(12,6))
plt.tight_layout()
```

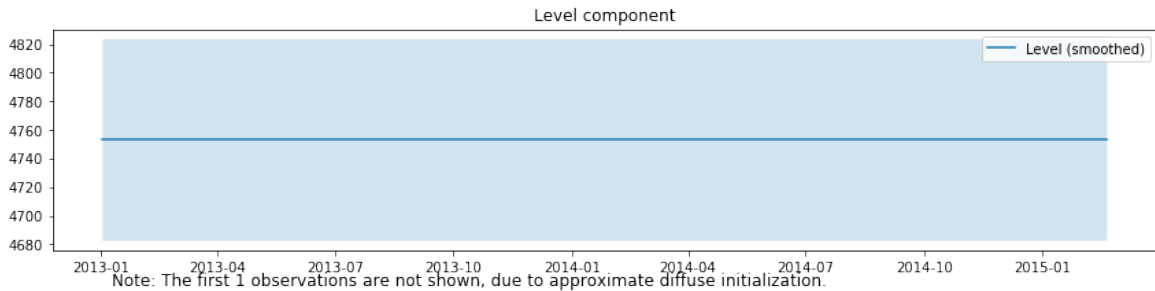


Рис. 6: Декомпозиция в константной модели

3.3 Модель локального уровня – Local level

Эта модель, в отличие от предыдущих, содержит два параметра, так как в ней тренд μ_t предполагается зависящим от времени и содержащим ошибку η_t . В терминах модели состояние-наблюдение тренд – это состояние, то есть часть, которую мы не наблюдаем, а y_t – напротив, видимые данные, то есть наблюдения.

$$y_t = \mu_t + \varepsilon_t, \varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2)$$

$$\mu_{t+1} = \mu_t + \eta_t, \eta_t \sim \mathcal{N}(0, \sigma_\eta^2)$$

$$\mu_1 \sim \mathcal{N}(a_1, P_1)$$

В модели локального уровня распределения случайных величин y_t и μ_t зависят от времени t , и дополнительно предполагается, что в начальный момент времени распределение μ_1 таково: $\mu_1 \sim \mathcal{N}(a_1, P_1)$, где a_1, P_1 , а также $\sigma_\varepsilon^2, \sigma_\eta^2$ известны. Это необходимо, потому что значения ненаблюдаемой переменной μ_t находятся с помощью фильтра Калмана, который позволяет посчитать математическое ожидание и дисперсию ненаблюдаемого состояния с учётом наблюдений. Фильтр Калмана – это рекурсивный алгоритм, и текущая оценка параметров обновляется, как только появляется новое наблюдение, иначе говоря, повторяются два этапа: предсказание и корректировка.

Формальное описание фильтра Калмана следующее:

- $Y_{t-1} = (y_1, \dots, y_{t-1})'$ – вектор наблюдений
- $\mu_t \mid Y_{t-1} \sim \mathcal{N}(a_t, P_t)$ – условное распределение μ_t при известном Y_{t-1} , где a_t, P_t также известны
- $\mu_t \mid Y_t \sim \mathcal{N}(a_{t|t}, P_{t|t})$ – условное распределение μ_t при известном Y_t
- $\mu_{t+1} \mid Y_t \sim \mathcal{N}(a_{t+1}, P_{t+1})$ – условное распределение μ_{t+1} при известном Y_t

Задача – найти $a_{t|t}, P_{t|t}, a_{t+1}$ и P_{t+1} , когда появляется новое наблюдение y_t , где $a_{t|t}$ – оценка фильтра для μ_t , a_{t+1} – прогноз для μ_t на один шаг вперёд, а $P_{t|t}$ и P_{t+1} – соответствующие дисперсии. Также обозначим за $v_t = y_t - a_t$ величину ошибки прогноза.

Пользуясь нормальностью распределений, можно показать¹, что

$$\begin{aligned} a_{t|t} &= a_t + \frac{P_t}{P_t + \sigma_\varepsilon^2} v_t & a_{t+1} &= a_t + \frac{P_t}{P_t + \sigma_\varepsilon^2} v_t \\ P_{t|t} &= \frac{P_t \sigma_\varepsilon^2}{P_t + \sigma_\varepsilon^2} & P_{t+1} &= \frac{P_t \sigma_\varepsilon^2}{P_t + \sigma_\varepsilon^2} + \sigma_\eta^2 \end{aligned}$$

Далее необходимо посчитать дисперсию ошибки прогноза: $F_t = \text{Var}(v_t) = P_t + \sigma_\varepsilon^2$, и усиление Калмана: $K_t = P_t / F_t$, после чего прогноз обновляется.

Все шаги обновления:

$$\begin{aligned} v_t &= y_t - a_t & F_t &= P_t + \sigma_\varepsilon^2 \\ a_{t|t} &= a_t + K_t v_t & P_{t|t} &= P_t (1 - K_t) \\ a_{t+1} &= a_t + K_t v_t & P_{t+1} &= P_t (1 - K_t) + \sigma_\eta^2 \end{aligned}$$

¹Подробный вывод соотношений содержится в [3] на стр. 11

Оценки параметров будут следующими:

	coef	std err	z	P> z	[0.025	0.975]
sigma2.irregular	$1.31 \cdot 10^5$	$1.61 \cdot 10^4$	8.17	0	$9.97 \cdot 10^4$	$1.63 \cdot 10^5$
sigma2.level	$3.91 \cdot 10^5$	$3.08 \cdot 10^4$	12.7	0	$3.30 \cdot 10^5$	$4.51 \cdot 10^5$

Таблица 5: Оценки параметров в модели локального уровня

Наглядные же результаты модели таковы:

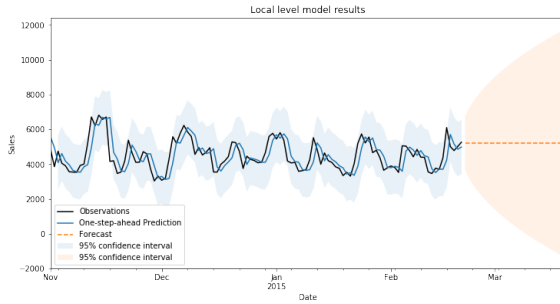


Рис. 7: Прогноз в модели локального уровня

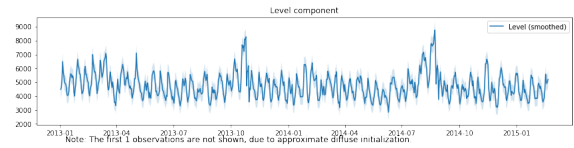


Рис. 8: Декомпозиция в модели локального уровня

3.4 Случайное блуждание – Random walk

В модели случайного блуждания каждое следующее значение предполагается равным сумме предыдущего значения и ошибки.

$$y_t = \mu_t$$

$$\mu_t = \mu_{t-1} + \eta_t, \eta_t \sim \mathcal{N}(0, \sigma_\eta^2)$$

При прогнозе на известных наблюдениях на один шаг вперёд получится этот же исходный ряд, но с лагом в один период, а прогнозирование на большее количество шагов вперёд будет равно последнему известному значению. Чтобы доказать это, нужно взять условное математическое ожидание от прогнозируемого значения при всех известных предыдущих значениях. Для μ_{t+k} и для любого $k \in \mathbb{N}$ уравнение ненаблюдаемой компоненты можно привести к виду $\mu_{t+k} = \mu_t + \sum_{i=2}^{k+1} \eta_i$, последовательно подставляя выражения для μ_i до первого известного. Тогда прогноз будет таким:

$$\hat{\mu}_{t+k} = E(\mu_t + \sum_{i=2}^{k+1} \eta_i \mid \mu_t, \mu_{t-1} \dots) = E(\mu_t \mid \mu_t, \mu_{t-1} \dots) = \mu_t$$

что и есть последнее известное значение.

	coef	std err	z	P> z	[0.025	0.975]
sigma2.level	$6.45 \cdot 10^5$	$1.43 \cdot 10^4$	45	0	$6.17 \cdot 10^5$	$6.73 \cdot 10^5$

Таблица 6: Оценки параметров в модели случайного блуждания

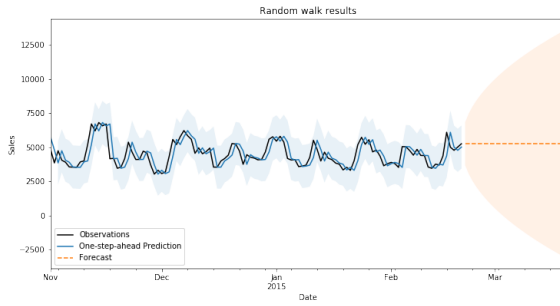


Рис. 9: Прогноз модели случайного блуждания

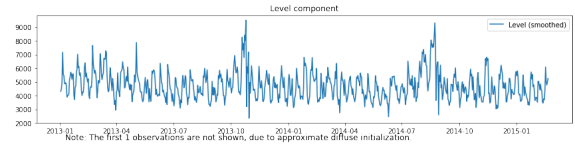


Рис. 10: Декомпозиция в модели случайного блуждания

3.5 Постоянный тренд – Deterministic trend

В этой модели задаётся линейный тренд:

$$y_t = \mu_t + \varepsilon_t, \varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2)$$

$$\mu_t = \mu_{t-1} + \beta$$

И переменная, отвечающая за тренд, меняется на константу каждый период, покажем:

$$y_t = \mu_{t-1} + \beta + \varepsilon_t$$

$$y_{t-1} = \mu_{t-2} + \beta + \varepsilon_{t-1}$$

$$\begin{aligned} E(y_t - y_{t-1}) &= E[(\mu_{t-1} + \beta + \varepsilon_t) - (\mu_{t-2} + \beta + \varepsilon_{t-1})] = E(\mu_{t-1} + \varepsilon_t - \mu_{t-2} - \varepsilon_{t-1}) = \\ &= E(\mu_{t-2} + \beta - \mu_{t-2}) + E(\varepsilon_t) - E(\varepsilon_{t-1}) = \beta \end{aligned}$$

	coef	std err	z	P> z	[0.025	0.975]
sigma2.irregular	$1.00 \cdot 10^6$	$3.62 \cdot 10^4$	27.7	0	$9.31 \cdot 10^5$	$1.07 \cdot 10^6$
β_{t+h}	-0.47					

Таблица 7: Оценки параметров в модели постоянного тренда

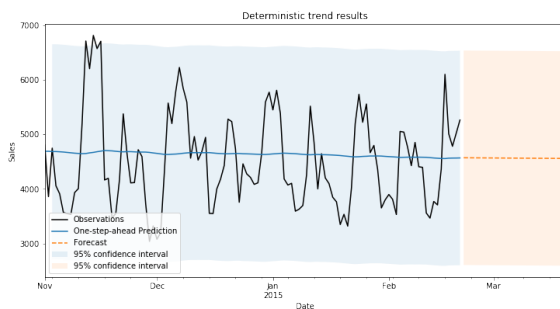


Рис. 11: Прогноз в модели постоянного тренда

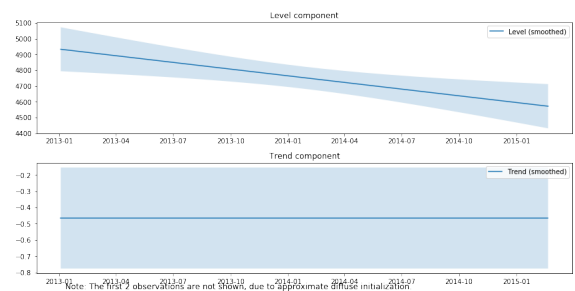


Рис. 12: Декомпозиция в модели постоянного тренда

3.6 Локальный линейный постоянный тренд – Local linear deterministic trend

Отличие этой модели от предыдущей только в добавлении ошибки η_t в тренд:

$$y_t = \mu_t + \varepsilon_t, \varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2)$$

$$\mu_t = \mu_{t-1} + \beta + \eta_t, \eta_t \sim \mathcal{N}(0, \sigma_\eta^2)$$

Однако прогноз на известных данных больше напоминает результаты модели случайного блуждания: предыдущее известное значение + ошибка прогноза, смещённое на фиксированную величину. Прогноз неизвестных значений отличается от каждого предыдущего на константу β – рассуждения такие же, как и в предыдущей модели. Тем не менее, из-за добавленной ошибки доверительный интервал будет расширяться со временем.

	coef	std err	z	P> z	[0.025	0.975]
sigma2.irregular	$1.29 \cdot 10^5$	$1.62 \cdot 10^4$	7.99	0	$9.76 \cdot 10^4$	$1.61 \cdot 10^5$
sigma2.level	$3.94 \cdot 10^5$	$3.13 \cdot 10^4$	12.6	0	$3.33 \cdot 10^5$	$4.56 \cdot 10^5$
$\hat{\beta}_{t+h}$	0.52					

Таблица 8: оценки параметров в модели локального линейного постоянного тренда

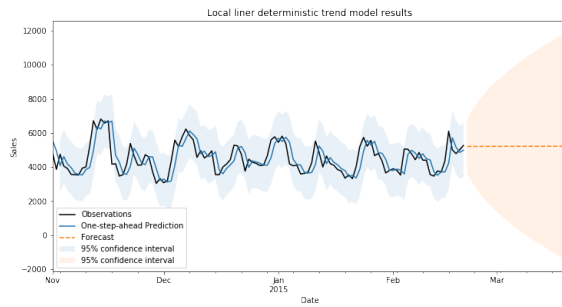


Рис. 13: Прогноз в модели локального линейного постоянного тренда

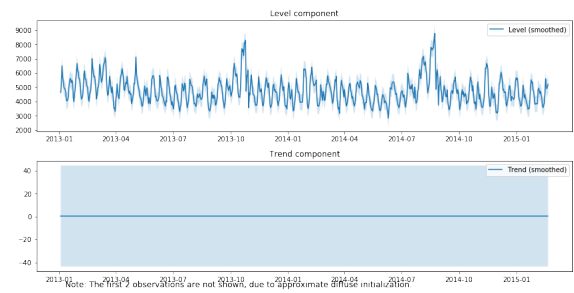


Рис. 14: Декомпозиция в модели локального линейного постоянного тренда

3.7 Случайное блуждание со смещением – Random walk with drift

Модель случайного блуждания со смещением записывается следующим образом:

$$y_t = \mu_t$$

$$\mu_t = \mu_{t-1} + \beta + \eta_t, \eta_t \sim \mathcal{N}(0, \sigma_\eta^2)$$

Если предположить начальное значение $\mu_0 = a$ известным, то модель можно записать в виде: $\mu_t = a + t \cdot \beta + \sum_{j=0}^{t-1} \varepsilon_{t-j}$. Найдём математическое ожидание и дисперсию:

$$E(\mu_t) = E(y_t) = t \cdot \beta \rightarrow \infty$$

$$Var(\mu_t) = Var(y_t) = t \cdot \sigma_\varepsilon^2 \rightarrow \infty$$

что объясняет вид графика с прогнозом: прогнозируемые значения – это прямая с постоянным наклоном, а доверительный интервал расширяется со временем.

	coef	std err	z	P> z	[0.025	0.975]
sigma2.level	$6.46 \cdot 10^5$	$1.44 \cdot 10^4$	44.9	0	$6.18 \cdot 10^5$	$6.74 \cdot 10^5$
$\hat{\beta}_{t+h}$	-0.34					

Таблица 9: Оценки параметров в модели случайного блуждания со смещением

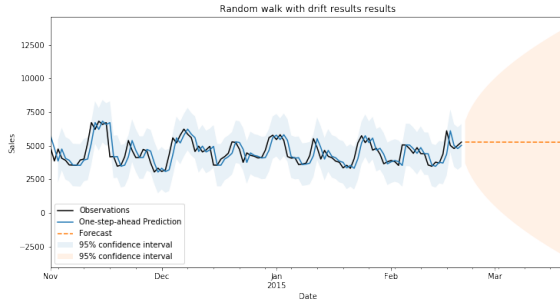


Рис 15: Прогноз в модели случайного блуждания со смещением

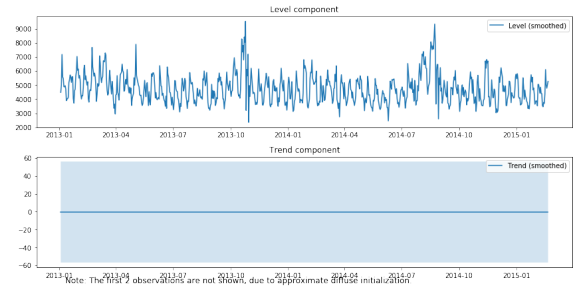


Рис. 16: Декомпозиция в модели случайного блуждания со смещением

3.8 Локальный линейный тренд – Local linear trend

В этой модели добавляется ещё одна случайная компонента: наклон тренда β , значения которого моделируются случайным блужданием.

$$y_t = \mu_t + \varepsilon_t, \varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2)$$

$$\mu_t = \mu_{t-1} + \beta_{t-1} + \eta_t, \eta_t \sim \mathcal{N}(0, \sigma_\eta^2)$$

$$\beta_t = \beta_{t-1} + \zeta_t, \zeta_t \sim \mathcal{N}(0, \sigma_\zeta^2)$$

Или она же, но в форме модели состояние-наблюдение:

$$y_t = \begin{pmatrix} 1 & 0 \end{pmatrix} + \begin{pmatrix} \mu_t \\ \beta_t \end{pmatrix} + \varepsilon_t$$

$$\begin{pmatrix} \mu_{t+1} \\ \beta_{t+1} \end{pmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \mu_t \\ \beta_t \end{pmatrix} + \begin{pmatrix} \eta_t \\ \zeta_t \end{pmatrix}$$

В модели остается только найти ковариационные матрицы:

$$H_t = \begin{bmatrix} \sigma_\varepsilon^2 \end{bmatrix}$$

$$Q_t = \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\zeta^2 \end{bmatrix}$$

	coef	std err	z	P> z	[0.025	0.975]
sigma2.irregular	$1.30 \cdot 10^5$	$1.68 \cdot 10^4$	7.74	0	$9.68 \cdot 10^4$	$1.62 \cdot 10^5$
sigma2.level	$3.96 \cdot 10^5$	$3.15 \cdot 10^4$	12.6	0	$3.34 \cdot 10^5$	$4.57 \cdot 10^5$
sigma2.trend	$0.04 \cdot 10^{-2}$	63.8	$6.26 \cdot 10^{-6}$	1	-125	125

Таблица 10: Оценки параметров в модели локального линейного тренда

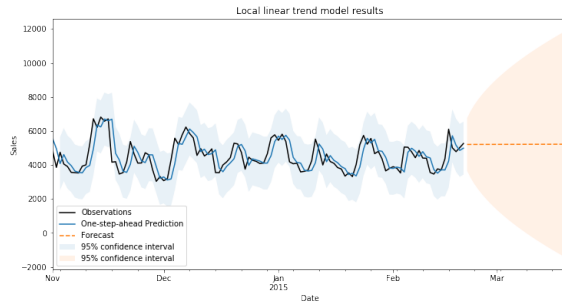


Рис. 17: Прогноз в модели локального линейного тренда

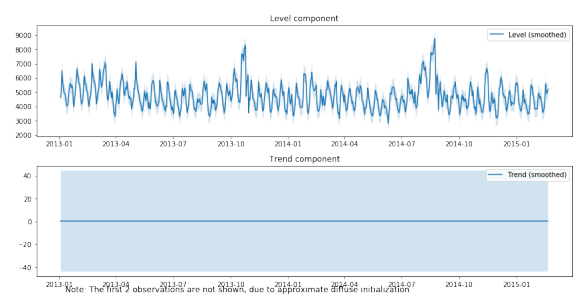


Рис. 18: Декомпозиция в модели локального линейного тренда

3.9 Сглаженный тренд – Smoothed trend

Отличие этой модели от предыдущей в том, что в уравнении, задающем тренд, отсутствует компонента ошибки η_t , отчего модель получила название ”сглаженный тренд”. Сам же тренд по-прежнему моделируется случайным блужданием.

$$y_t = \mu_t + \varepsilon_t, \varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2)$$

$$\mu_t = \mu_{t-1} + \beta_{t-1}$$

$$\beta_t = \beta_{t-1} + \zeta_t, \zeta_t \sim \mathcal{N}(0, \sigma_\zeta^2)$$

	coef	std err	z	P> z	[0.025	0.975]
sigma2.irregular	$2.98 \cdot 10^5$	$1.09 \cdot 10^4$	27.2	0	$2.77 \cdot 10^5$	$3.19 \cdot 10^5$
sigma2.trend	$8.25 \cdot 10^4$	7917	10.4	0	$6.70 \cdot 10^4$	$9.80 \cdot 10^4$

Таблица 11: Оценки параметров в модели сглаженного тренда

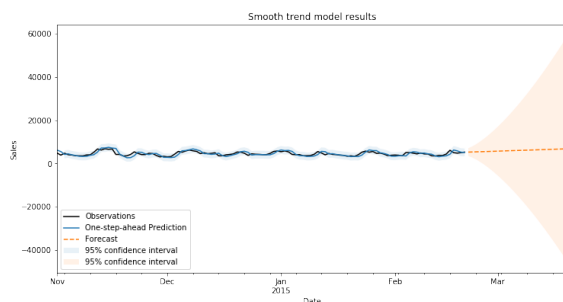


Рис. 19: Прогноз в модели сглаженного тренда

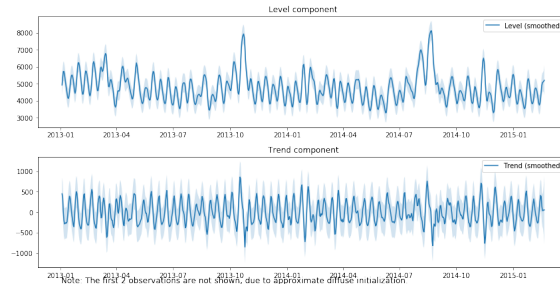


Рис. 20: Декомпозиция в модели сглаженного тренда

3.10 Случайный тренд – Random trend

В этой модели отсутствует ещё одна переменная, та, что отвечает за погрешность наблюдения, ε_t . В таком случае наблюдения моделируются предыдущими значением тренда и шумовым значением из уравнения, задающего наклон.

$$y_t = \mu_t$$

$$\mu_t = \mu_{t-1} + \beta_{t-1}$$

$$\beta_t = \beta_{t-1} + \zeta_t, \zeta_t \sim \mathcal{N}(0, \sigma_\zeta^2)$$

	coef	std err	z	P> z	[0.025	0.975]
sigma2.trend	$1.53 \cdot 10^6$	$3.15 \cdot 10^4$	48.6	0	$1.47 \cdot 10^6$	$1.59 \cdot 10^6$

Таблица 12: Оценки параметров в модели случайного тренда

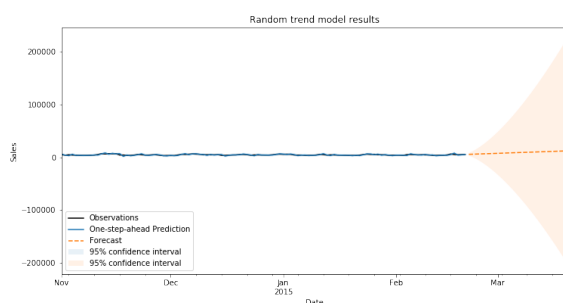


Рис. 21: Прогноз в модели случайного тренда

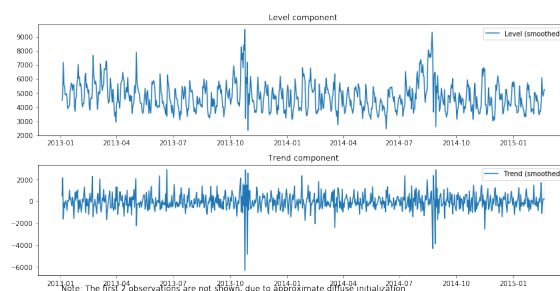


Рис. 22: Декомпозиция в модели случайного тренда

3.11 Локальный линейный тренд с сезонностью

В данной модели добавляется качественно новый компонент: сезонность, что позволяет учитывать колебания, повторяющиеся с некоторой фиксированной периодичностью.

$$\begin{aligned}
y_t &= \mu_t + \gamma_t + \varepsilon_t, \varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2) \\
\mu_t &= \mu_{t-1} + \beta_{t-1} + \eta_t, \eta_t \sim \mathcal{N}(0, \sigma_\eta^2) \\
\beta_t &= \beta_{t-1} + \zeta_t, \zeta_t \sim \mathcal{N}(0, \sigma_\zeta^2) \\
\mu_1 &\sim \mathcal{N}(a_1, P_1) \\
\gamma_t &= - \sum_{j=1}^{s-1} \gamma_{t+1-j} + \omega_t, \omega_t \sim \mathcal{N}(0, \sigma_\omega^2), s - \text{периодичность}
\end{aligned}$$

По определению, детерминистическая сезонность с периодичностью s – это последовательность чисел, которые повторяются каждые s последовательных наблюдений: $\gamma_t = \gamma_{t-s}$. Более того, сумма их значений равна нулю через s последовательных наблюдений, поскольку в данных за весь год сезонности нет: $\sum_{j=0}^{s-1} \gamma_{t-j} = 0$. Таким образом, детерминистическая сезонность – это периодическая функция с нулевым средним значением, и при известных первых s значениях (которые должны давать в сумме ноль), можно получить все остальные значения с помощью рекурсии: $\gamma_t = - \sum_{j=1}^{s-1} \gamma_{t-j}$. А для того чтобы компонента могла изменяться с течением времени, необходимо добавить случайный шок ω_t .

Зададим годовую сезонность, передав в описание модели параметр `seasonal=12`. Заметим также, что модель с сезонностью наиболее точно улавливает наблюдения среди всех рассмотренных в этой работе моделей: её показатели AIC, BIC и HQIC наименьшие. Тем не менее на рисунке 25 видно, что остатки в модели не полностью соответствуют нормальному распределению.

	coef	std err	z	P> z	[0.025	0.975]
sigma2.irregular	$1.36 \cdot 10^5$	$1.71 \cdot 10^4$	7.93	0	$1.02 \cdot 10^5$	$1.69 \cdot 10^5$
sigma2.level	$3.36 \cdot 10^5$	$2.86 \cdot 10^4$	11.8	0	$2.80 \cdot 10^5$	$3.92 \cdot 10^5$
sigma2.trend	$1.82 \cdot 10^{-6}$	45.8	$3.97 \cdot 10^{-8}$	1	-89.9	89.9
sigma2.seasonal	1014	504	2.01	0.04	25.8	2002

Таблица 13: Оценки параметров в модели с сезонностью

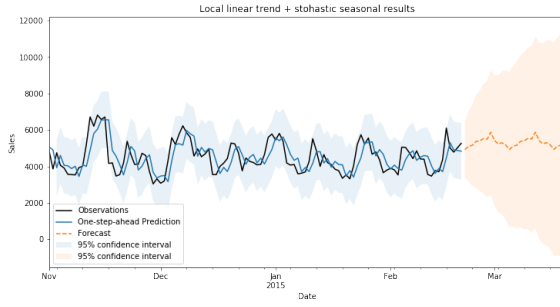


Рис.23: Прогноз в модели с сезонностью

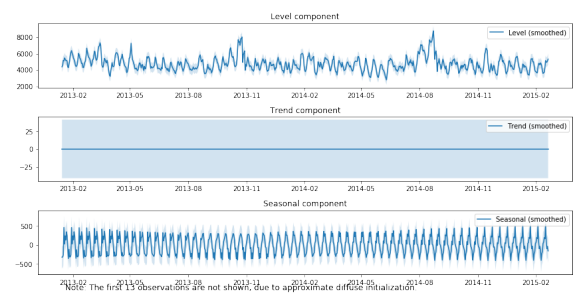


Рис. 24: Декомпозиция в модели с сезонностью

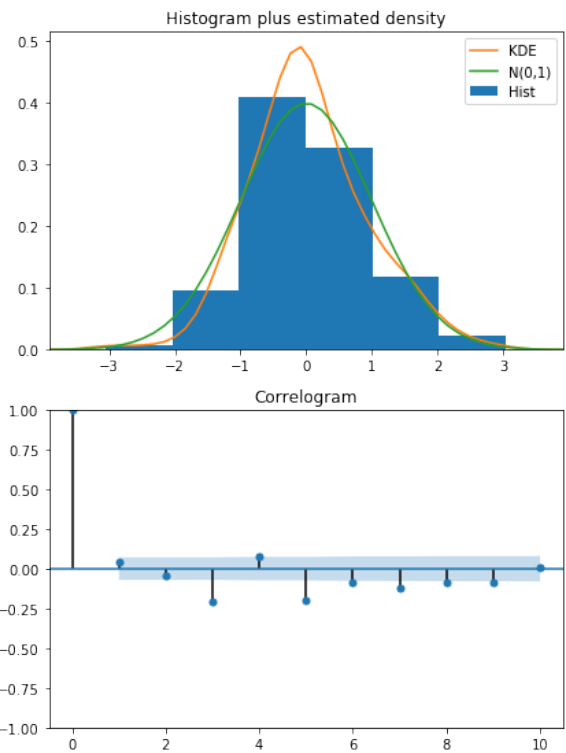
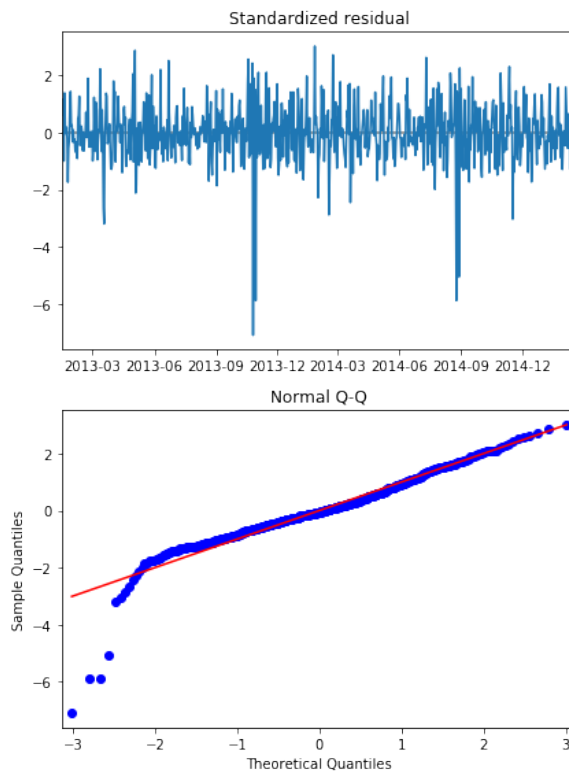


Рис. 25: Анализ остатков в модели с сезонностью

3.12 Локальный линейный тренд с сезонностью и регрессорами

Предыдущую версию модели можно уточнить, добавив наблюдаемый объясняющие переменные, или регрессоры. Добавив их в уравнение, связывающее наблюдения с ненаблюдаемыми компонентами, получим статическую регрессию (как в данном случае), а при добавлении регрессора в одно из уравнений ненаблюдаемой компоненты – динамическую. Например, если определить μ_t как $\mu_t = \mu_{t-1} + \beta_{t-1} + \delta^T x_t + \eta_t$, то получим, что μ_{t+1} зависит от μ_t , которая в свою очередь зависит от x_t . Эти рассуждения верны и для $t+2, t+3, \dots$. Таким образом, регрессор, относящийся к моменту времени t

влияет на настоящее и будущее значение μ , а следовательно, и на все соответствующие значения y_t .

Но мы рассмотрим статическую регрессию с двумя регрессорами, которая записывается следующим образом:

$$\begin{aligned} y_t &= \mu_t + \gamma_t + \delta_1^T x_t + \delta_2^T z_t + \varepsilon_t, \varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2) \\ \mu_t &= \mu_{t-1} + \beta_{t-1} + \eta_t, \eta_t \sim \mathcal{N}(0, \sigma_\eta^2) \\ \beta_t &= \beta_{t-1} + \zeta_t, \zeta_t \sim \mathcal{N}(0, \sigma_\zeta^2) \\ \mu_1 &\sim \mathcal{N}(a_1, P_1) \\ \gamma_t &= - \sum_{j=1}^{s-1} \gamma_{t+1-j} + \omega_t, \omega_t \sim \mathcal{N}(0, \sigma_\omega^2) \end{aligned}$$

Значения x_t положим равными единице, если день t – воскресенье, и нулю иначе, а z_t будут единицами только в случае праздника в день t

```
In
[11]: data['isSunday'] = (data.DayOfWeek==7).astype('int')
      data['isHoliday'] = (data['StateHoliday']=='a').astype('int')
      sd_dta = data[data.Store==1][['Sales', 'isSunday',
                                   'isHoliday']].sort_index()
```

Обучаем модель, добавляя регрессоры в описание параметров:

```
In
[12]: model_12_1 = statsmodels.tsa.statespace.structural.Unobserved\
      Components(endog=sd_dta['Sales'], level='l1trend',
                  trend=True, seasonal=12, cycle=False,
                  autoregressive=None,
                  exog=sd_dta[['isSunday', 'isHoliday']],
                  irregular=True, stochastic_level=True,
                  stochastic_trend=True,
                  stochastic_seasonal=True,
                  stochastic_cycle=False,
                  damped_cycle=False,
                  cycle_period_bounds=None)
```

	coef	std err	z	P> z	[0.025	0.975]
sigma2.irregular	$5.56 \cdot 10^5$	$1.80 \cdot 10^4$	30.9	0	$5.21 \cdot 10^5$	$5.92 \cdot 10^5$
sigma2.level	$1.89 \cdot 10^5$	$1.45 \cdot 10^4$	13.1	0	$1.61 \cdot 10^5$	$2.17 \cdot 10^5$
sigma2.trend	$7.91 \cdot 10^{-5}$	16.7	$4.75 \cdot 10^{-6}$	1	-32.7	32.7
sigma2.seasonal	0,01	154	$7.30 \cdot 10^{-5}$	1	-302	302
beta.isSunday	-4768	89.2	-53.5	0	-4942	-4593
beta.isHoliday	-5030	313	-16.1	0	-5644	-4417

Таблица 14: Оценки параметров для модели с регрессорами

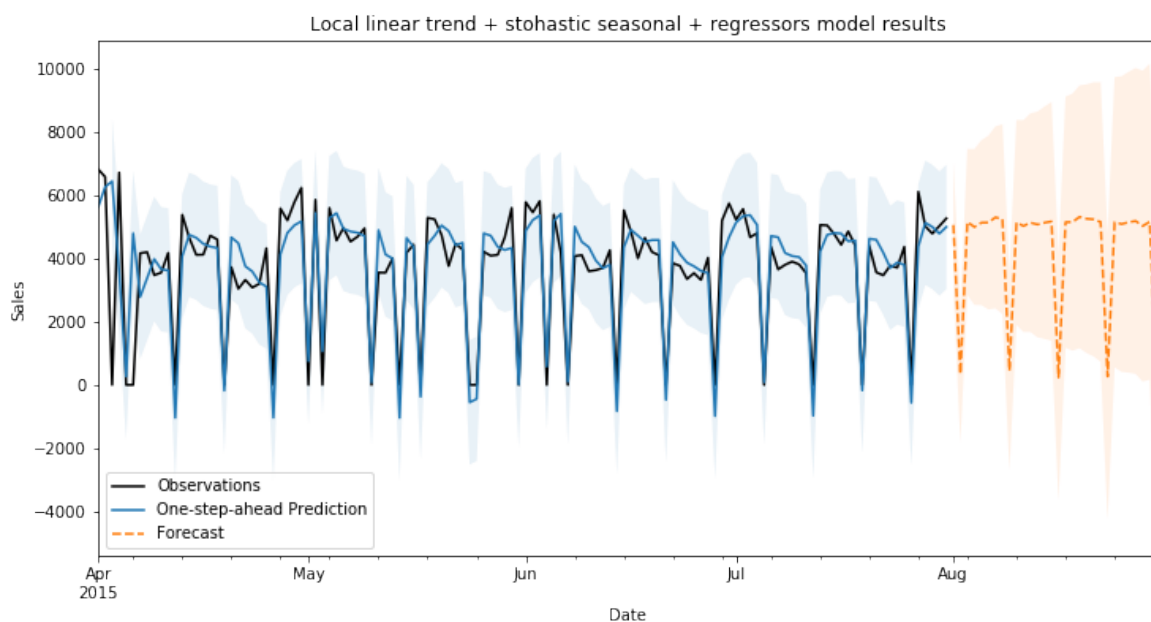


Рис. 26: Прогноз в модели с регрессорами

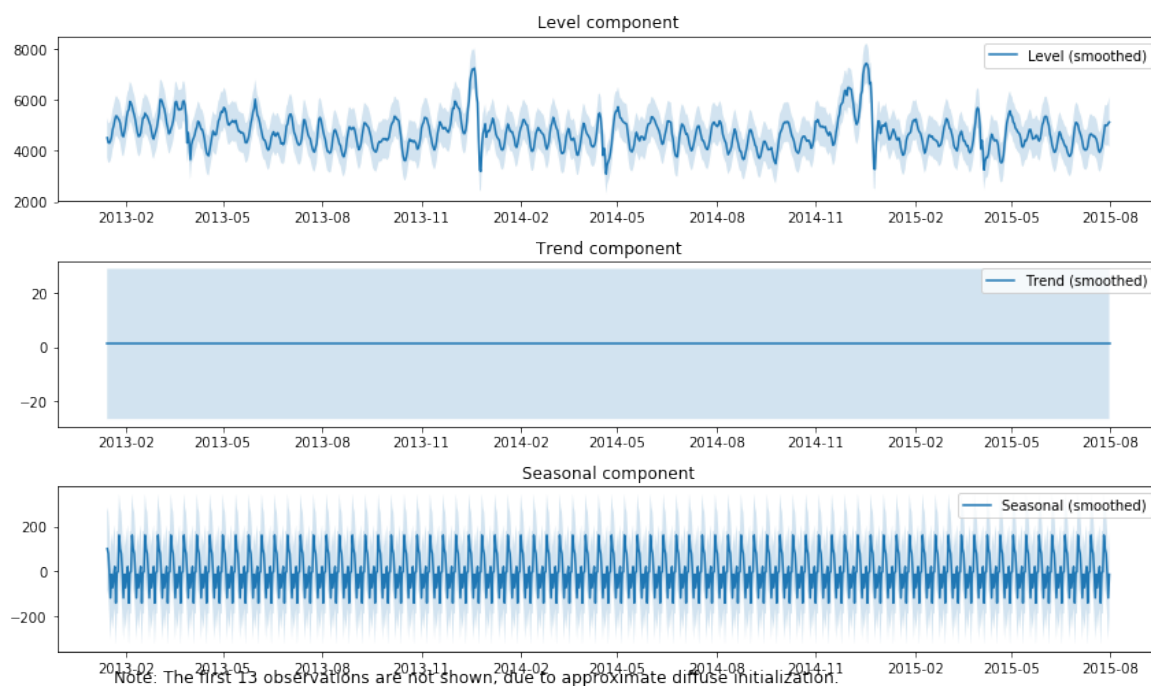


Рис. 27: Декомпозиция в модели с регрессорами

По рисунку 26 видно, что модель прогнозирует даже отрицательные продажи, когда магазин закрыт, другими словами, она очень неустойчива к выбросам. Так что, если задаться целью прогнозировать максимально точно, стоит обучаться на данных с ненулевыми значениями и восстанавливать их потом.

4 Построение модели многомерных рядов

Модели, описанные в предыдущей главе, могут быть обобщены для многомерного случая, когда нужно одновременно анализировать p временных рядов. В частности, в данных, которые я использовала, содержалась информация о продажах в более чем ста магазинах сети, и для каждого такого ряда необходимо сделать прогноз.

4.1 Описание модели

Модель состояние-наблюдения для нескольких временных рядов записывается следующим образом:

$$y_t = Z_t \alpha_t + \varepsilon_t \quad \varepsilon_t \sim \mathcal{N}(0, H_t) \quad (1)$$

$$\alpha_{t+1} = T_t \alpha_t + R_t \eta_t \quad \eta_t \sim \mathcal{N}(0, Q_t) \quad (2)$$

для $t = 1, 2, \dots$. Уравнение наблюдений y_t — это вектор размера $p \times 1$, который содержит значения всех p временных рядов в момент времени t . Вектор ε_t размерности $p \times 1$ содержит все шоки для соответствующих временных рядов. Эти ошибки распределены нормально с нулевым математическим ожиданием и ковариационной матрицей H_t размерности $p \times p$.

Вектор $m \times 1$ состояний α_t содержит все ненаблюдаемые компоненты и неизвестные постоянные значения. Матрица Z_t , размерность которой $p \times m$ связывает эти компоненты и регрессоры с вектором наблюдений. Матрица T_t — называется матрицей перехода и имеет размерность $m \times m$. Вектор $r \times 1$ η_t содержит шоки, которые распределены нормально с нулевым математическим ожиданием и ковариационной матрицей Q_t , $r \times r$. Часто $r = m$ и матрица R_t — это единичная матрица I_m . Но поскольку R_t задаётся произвольно, то её размерность может быть $m \times r$, где $r < m$.

4.2 Пример: модель локального линейного уровня с регрессорами

Запишем модель локального линейного уровня в форме состояние-наблюдение для двух временных рядов ($p = 2$). Выпишем все вектора и матрицы:

$$\alpha_t = \begin{pmatrix} \mu_t^{(1)} \\ \nu_t^{(1)} \\ \beta_t^{(1)} \\ \mu_t^{(2)} \\ \nu_t^{(2)} \\ \beta_t^{(2)} \end{pmatrix} \quad \eta_t = \begin{pmatrix} \xi_t^{(1)} \\ \zeta_t^{(1)} \\ \xi_t^{(2)} \\ \zeta_t^{(2)} \end{pmatrix} \quad T_t = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad R_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Z_t = \begin{bmatrix} 1 & 0 & x_t & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & x_t \end{bmatrix} \quad H_t = \begin{bmatrix} \sigma_{\varepsilon^{(1)}}^2 & \text{cov}(\varepsilon^{(1)}, \varepsilon^{(2)}) \\ \text{cov}(\varepsilon^{(1)}, \varepsilon^{(2)}) & \sigma_{\varepsilon^{(2)}}^2 \end{bmatrix}$$

$$Q_t = \begin{bmatrix} \sigma_{\xi^{(1)}}^2 & 0 & \text{cov}(\xi^{(1)}, \xi^{(2)}) & 0 \\ 0 & \sigma_{\zeta^{(1)}}^2 & 0 & \text{cov}(\zeta^{(1)}, \zeta^{(2)}) \\ \text{cov}(\xi^{(1)}, \xi^{(2)}) & 0 & \sigma_{\xi^{(2)}}^2 & 0 \\ 0 & \text{cov}(\zeta^{(1)}, \zeta^{(2)}) & 0 & \sigma_{\zeta^{(2)}}^2 \end{bmatrix}$$

В данном случае регрессор x_t применяется к обоим рядам. Надстрочные индексы (1) и (2) выделяют компоненты, относящиеся к первому и второму ряду соответственно.

Уравнения для наблюдений можно получить, подставив соответствующие матрицы в уравнение 1:

$$\begin{aligned} y_t^{(1)} &= \mu_t^{(1)} + \beta_t^{(1)} + x_t + \varepsilon_t^{(1)} \\ y_t^{(2)} &= \mu_t^{(2)} + \beta_t^{(2)} + x_t + \varepsilon_t^{(2)} \end{aligned}$$

Аналогично для уравнений, задающих состояния, из уравнения 2 получим:

$$\begin{aligned} \mu_{t+1}^{(1)} &= \mu_t^{(1)} + \nu_t^{(1)} + \xi_t^{(1)} \\ \nu_{t+1}^{(1)} &= \nu_t^{(1)} + \zeta_t^{(1)} \\ \beta_{t+1}^{(1)} &= \beta_t^{(1)} \\ \mu_{t+1}^{(2)} &= \mu_t^{(2)} + \nu_t^{(2)} + \xi_t^{(2)} \\ \nu_{t+1}^{(2)} &= \nu_t^{(2)} + \zeta_t^{(2)} \\ \beta_{t+1}^{(2)} &= \beta_t^{(2)} \end{aligned}$$

4.3 Общие уровни и наклоны

В случае многомерных временных рядов уравнения наблюдений и состояний включают в себя ошибки, относящиеся к определённым компонентам. Например, $\zeta_t^{(1)}$ и $\zeta_t^{(2)}$ связаны с наклонами $\nu_t^{(1)}$ и $\nu_t^{(2)}$. Когда эти ошибки не коррелированы, наклоны не зависимы. И, наоборот, зависимы, когда $\text{cov}(\zeta^{(1)}, \zeta^{(2)}) \neq 0$. Модели многомерных временных рядов, которые содержат векторы ненаблюдаемых компонент, зависящих от коррелирующих ошибок, называются *по-видимому несвязанными уравнениями временных рядов* (*seemingly unrelated time series equations*), то есть подразумевается, что уравнения независимы, несмотря на то что ошибки могут быть коррелированы.

Степень зависимости измеряется коэффициентом корреляции между ошибками. В случае $-1 < \text{corr}(\zeta^{(1)}, \zeta^{(2)}) < 1$ ранг матрицы

$$\begin{bmatrix} \sigma_{\zeta^{(1)}}^2 & \text{cov}(\zeta^{(1)}, \zeta^{(2)}) \\ \text{cov}(\zeta^{(1)}, \zeta^{(2)}) & \sigma_{\zeta^{(2)}}^2 \end{bmatrix}$$

равен двум. Когда же $\text{corr}(\zeta^{(1)}, \zeta^{(2)}) = \pm 1$, ранг равен единице.

Таким образом, исходя из информации о ранге ковариационной матрицы, можно сделать вывод, являются ли компоненты рядов общими, что важно, поскольку даёт дополнительную информацию о взаимосвязях между временными рядами.

5 Заключение

В работе описан ряд одномерных моделей с ненаблюдаемыми компонентами, для каждой из которых представлен прогноз на реальных данных и объяснение вида прогноза. Более того, в исследовании разобрана модель для многомерных временных рядов, указаны её особенности, приведён пример для случая двух временных рядов. В работе также содержится код, позволяющий описывать модели на языке Python.

Исследование может быть продолжено за счёт использования байесовского подхода к оценке параметров. Помимо этого, существенным вкладом может стать код, с помощью которого можно моделировать многомерные временные ряды.

6 Список источников

- [1] Brockwell, P. J. and R. A. Davis (2002). Introduction to Time Series and Forecasting (2nd edition). Springer. pp. 1–40.
- [2] Commandeur, J. J. F. and S. J. Koopman (2007). An Introduction to State Space Time Series Analysis. Oxford University Press.
- [3] Durbin, J. and S. J. Koopman (2001). Time Series Analysis by State Space Methods. Oxford University Press. pp. 9–16, 43–52.
- [4] Pelagatti, M. M. (2016). Time Series Modelling with Unobserved Components. CRC Press.