

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРНІВЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА**

**Факультет математики та інформатики
кафедра математичного моделювання**

ПОРІВНЯННЯ МЕТОДІВ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

Кваліфікаційна робота

Рівень вищої освіти – другий (магістерський)

Виконав:

студент 6 курсу, 607 групи

Мунтян Костянтин

Керівник:

кандидат фізико-математичних наук,
доцент Дорошенко І. В.

*До захисту допущено
на засіданні кафедри
протокол № 8 від 6 грудня 2022 р.
Зав. кафедрою _____ проф. Черевко І.М.*

Чернівці – 2022

Анотація

У даній роботі розглянуто порівняння методів прогнозування часових рядів: наївний метод, просте експоненціальне згладжування, метод тренду Холта, ARIMA, TBATS. Для порівняння методів прогнозування використано середню абсолютну відсоткову помилку (Mean Absolute Percentage Error, MAPE). Найкращою вважається модель із найменшим значенням MAPE. Для побудови моделей використано дані про рівень безробіття у США. Всі статистичні обчислення, які присутні у роботі, здійснено у середовищі R Programming із застосуванням пакету RStudio.

Ключові слова: прогнозування, часові ряди, ARIMA, TBATS, Exponential Smoothing, Naïve, Holt's trend, декомпозиція часового ряду, MAPE.

Annotation

This work deals with the comparison of time series forecasting methods: naive method, simple exponential smoothing, Holt trend method, ARIMA, TBATS. Mean Absolute Percentage Error (MAPE) was used to compare forecasting methods. The model with the lowest MAPE value is considered the best. Data on the unemployment rate in the USA were used to build the models. All statistical calculations, which are present in the work, were carried out in the R Programming environment using the RStudio package.

Key words: forecasting, time series, ARIMA, TBATS, Exponential Smoothing, Naive, Holt's trend, time series decomposition.

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів наукових досліджень інших авторів мають посилання на відповідне джерело.

_____ Костянтин Мунтян

ЗМІСТ

Вступ	4
РОЗДІЛ I. ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ.....	6
1.1. Задачі прогнозування.....	6
1.2. Основні означення.....	8
1.3. Підходи до аналізу часових рядів.....	10
1.4. Стаціонарність часових рядів	11
РОЗДІЛ II. ПОБУДОВА МОДЕЛЕЙ ЧАСОВИХ РЯДІВ.....	15
2.1. Модель випадкового блукання	15
2.2. Наївний метод прогнозування	16
2.3. Метод експоненціального згладжування.....	17
2.4. Модель TBATS.....	18
2.5. ARIMA	21
РОЗДІЛ III. ДЕКОМПОЗИЦІЯ ЧАСОВОГО РЯДУ	23
3.1. Оцінка та видалення тренду та сезонних компонентів.....	23
3.2. Оцінка та видалення тренду за відсутності сезонності	24
3.2.1. Оцінка тренду	24
3.2.2. Видалення тренду за допомогою диференціації (взяття перших різниць).....	27
3.3. Оцінка та видалення тренду та сезонності	28
РОЗДІЛ IV. ПРОГРАМНА РЕАЛІЗАЦІЯ.....	31
4.1. Опис вхідних даних.....	31
4.2. Аналіз результатів.....	32
4.2.1. Naive Forecasting Method	33
4.2.2. Simple Exponential Smoothing.....	33
4.2.3. Holt's Trend Method.....	34
4.2.4. ARIMA	34
4.2.5. TBATS	35
Висновки.....	37
Список використаних джерел	38
Додатки	41

Вступ

Прогнозування є одним з найбільш складних і затребуваних завдань, що цікавлять людство. Людям завжди було цікаво знати, що ж чекає їх у майбутньому. Однак, не тільки інтерес робить це завдання настільки затребуваним – в умовах глобалізації та прискорення ритму життя, коли за долю секунди відбуваються тисячі угод, можливість як-небудь передбачити розвиток ринку хоча б на найближчий час може стати результатом мільйонних прибутків. З іншого боку, дії, засновані на помилковому прогнозі, можуть призвести до ще більших втрат. Усе це свідчить про високу *актуальність задачі прогнозування*.

З точки зору математики, завдання прогнозування зводиться до задачі ідентифікації об'єкта – де об'єктом є прогнозований процес. Тобто передбачається, що прогнозований процес має деяку «модель поведінки», яка взаємодіє із зовнішнім світом. Маючи цю модель поведінки, ми б змогли оцінити виходи моделі в майбутньому. Тут важливим є слово «оцінити» – воно спочатку попереджає, що ми можемо лише передбачити деяку оцінку майбутніх виходів моделі [1].

У наш час розроблений ґрунтовний математичний апарат для вирішення задачі прогнозування. Однак навряд чи можна назвати єдиний метод прогнозування, який варто застосовувати при вирішенні усього розмаїття можливих практичних завдань прогнозування – кожен метод найкраще підходить для вирішення деякого кола завдань із певної сфери застосування, що володіють певними властивостями. Найбільш універсальними виявилися методи, що використовують елементи штучного інтелекту – інтелектуальні системи прогнозування. Найвідоміший представник цієї групи методів – штучні нейронні мережі (ШНМ), що є системою взаємозалежних штучних нейронів. У даній дипломній роботі розглянуто кілька методів прогнозування реальних даних. Кожен із підходів є описаним теоретично. Практичні результати роботи кожної побудованої моделі є обґрунтованими.

Метою роботи є дослідження, практичне застосування та порівняння існуючих інтелектуальних методів та систем прогнозування.

Об'єктом дослідження є системи прогнозування в цілому.

Предмет дослідження – застосування існуючих методів та систем прогнозування до реальних даних.

Основні завдання роботи:

- ознайомитися з існуючими методами прогнозування часових рядів;
- порівняти кілька сучасних інтелектуальних методів прогнозування часових рядів на прикладі реальних даних;
- навчитися будувати моделі часових рядів у середовищі R Programming.

РОЗДІЛ І. ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

1.1. Задачі прогнозування

Наука про дані розвивається дуже швидко. Штучний інтелект та машинне навчання проникають у всі сфери життєдіяльності людини, здійснюючи прогнозування із застосуванням аналізу часових рядів. Завдяки прогнозуванню в торгівлі, електронній комерції, прогнозуванню погоди та астрономії життя стає простішим для конкретної людини, а використання та збереження ресурсів оптимізуються у глобальному плані. Завдяки аналізу часових рядів можливо використовувати досвід та приховані взаємозалежності в даних для одержання майбутніх тенденцій та оцінок спостережуваної величини. Опишемо основні чотири напрямки застосування теорії часових рядів та прогнозування.

- **Аналіз часових рядів у сфері торгівлі та інвестицій**

Сьогодні розроблено багато додатків, які зосереджуються виключно на цілях криптовалюти та торгівлі. Вони дають нам точну інформацію про торгівлю, фінанси та курси валют, підключаючись до центрального сервера. Клієнти можуть зробити прогноз за допомогою прогнозування ковзного середнього чи свічкового аналізу та інвестувати лише кількома клацаннями на своєму мобільному телефоні. У даний час активно розвивається торгівля та інвестиційний сектор на індійському ринку. Оскільки банки дають низькі відсотки, люди зацікавлені в торгівлі та інвестиціях, створюючи надійний портфель. за допомогою аналізу часових рядів користувачі легко передбачають правильний час для інвестування та виходу з меншими ризиками, щоб отримати максимальний прибуток від своїх інвестицій – і все це сидячи вдома, працюючи в офісі чи подорожуючи.

- **Аналіз часових рядів у секторі електронної комерції**

Коли більшість компаній закривалися під час пандемії корона вірусу, сектор електронної комерції продовжував активно розвиватися. Таким чином

транспортувалися продукти з одного місця в інше з підвищеною безпекою, дотримуючись інструкцій щодо COVID та необхідних запобіжних заходів. Завдяки неперервним продажам ці платформи електронної комерції змогли передбачити продукти, які мають вищий попит за кількістю продажів за останній час. І якщо клієнт купує продукт, які ще продукти він може придбати, пов'язані з ним – моделі та алгоритми можуть передбачити багато деталей на основі минулих даних. Тому, коли настають нестандартні події у світі, Amazon, Flipkart, Snapdeal, Myntra проводять свої кампанії з продажу з величезними знижками. Таким чином компанії привертають ще більше уваги до власної продукції. За усім цим працюють спеціалісти з обробки даних, які роблять прогнози, дотримуючись минулих даних і поточних тенденцій на ринку.

- **Аналіз часових рядів для прогнозування погоди**

Сьогодні ми всі користуємося смартфонами, ноутбуками, планшетами з інтернетом 4G і оптоволоконним кабелем, завдяки чому ми отримуємо найвищу швидкість і завжди залишаємося на зв'язку з усім світом. Ще кілька років тому, щоб дізнатися прогноз погоди, нам доводилося чекати на новини. Такі погодні «форсмажори», як циклон або торнадо, ми навряд чи могли передбачити. Але сьогодні – достатньо просто спитати у Google. Прогнозування погодних умов також здійснюється за допомогою методів прогнозування часових рядів. Володіючи наперед інформацією про сильні дощі, циклони або торнадо, можна легко вжити необхідних заходів і уникнути жертв. Отже, на основі прогнозу погоди тепер можливо приймати рішення – подорожувати чи ні.

- **Аналіз часових рядів в астрономії**

21 століття – це століття нових технологій, які залишаться актуальними для штучного інтелекту та машинного навчання. У нас час інформація про схід та захід сонця, дату та час сонячного затемнення, в якій частині світу буде повне затемнення і де його можна буде побачити – доступні нам лише за кілька кліків на смартфоні. Якщо астероїд наближається до Землі або супутники

запускаються до Місяця чи Марса, о котрій годині вони там приземляться – усе це прогнози аналізу часових рядів, які дають дуже точні дані.

1.2. Основні означення

Часовий ряд – це сукупність значень випадкового процесу, взятих через рівні проміжки часу t . Позначимо цю сукупність символом z_t . Фактично z_t – це є дискретний у часі випадковий процес. Задача прогнозування полягає в тому, що, знаючи значення процесу у момент t , необхідно спрогнозувати його значення в момент $t + l$, де l – час упередження. Щоб відрізнити прогнозне значення процесу від дійсного, дійсне значення часового ряду у момент $t + 1$ позначають символом z_{t+l} , а прогнозне значення – символом $z_t(l)$.

Зрозуміло, що точно спрогнозувати значення випадкового процесу, яким є часовий ряд, неможливо в принципі, а тому прогноз здійснюють, домагаючись мінімуму якогось критерію адекватності прогнозованої моделі. Якщо значення t є малим, то одним із таких критеріїв може бути дисперсія відхилення $z_t(l)$ від z_{t+l} , яка повинна бути для оптимальної моделі прогнозу мінімальною, тобто

$$E\{(z_{t+l} - z_t(l))^2\} \rightarrow \min$$

Означення 1. Модель часових рядів для спостережуваних даних $\{x_t\}$ – це специфікація спільних розподілів (або, можливо, лише середніх значень і коваріацій) послідовності випадкових величин $\{X_t\}$, реалізацією яких вважається $\{x_t\}$.

Зауваження. Ми будемо часто використовувати термін часовий ряд, щоб позначати як дані, так і процес, реалізацією якого вони є.

Повна ймовірнісна модель часового ряду для послідовності випадкових величин $\{X_1, X_2, \dots\}$ визначала б усі спільні розподіли випадкових векторів

$(X_1, \dots, X_n)'$, $n = 1, 2, \dots$, або, еквівалентно, всі ймовірності

$$P[X_1 \leq x_1, \dots, X_n \leq x_n], -\infty < x_1, \dots, x_n < \infty, n = 1, 2, \dots$$

Означення 2. Нехай $\{X_t\}$ — часовий ряд із $EX_t^2 < \infty$. Середня функція $\{X_t\}$ є

$$\mu_x(t) = E(X_t).$$

Коваріаційна функція $\{X_t\}$ є

$$\gamma_x(r, s) = \text{Cov}(X_r, X_s) = E[(X_r - \mu_x(r))(X_s - \mu_x(s))]$$

для всіх цілих r і s .

Означення 3. $\{X_t\}$ є (слабко) стаціонарним, якщо

(i) $\mu_x(t)$ не залежить від t ,

(ii) $\gamma_x(t + h, t)$

не залежить від t для кожного h .

Зауваження 1. Суворе стаціонарність часового ряду $\{X_t, t = 0, \pm 1, \dots\}$ визначається умовою, що X_1, \dots, X_n і $(X_{1+h}, \dots, X_{n+h})$ мають однакові спільні розподіли для всіх цілих чисел h і $n > 0$. Легко перевірити, що якщо X_t строго стаціонарний і $EX_t^2 < \infty$ для всіх t , то $\{X_t\}$ також є слабко стаціонарним. Щоразу, коли ми використовуємо термін стаціонарний, ми маємо на увазі слабко стаціонарний, як у означенні 3, якщо ми спеціально не вказуємо інше.

Зауваження 2. З огляду на умову (ii), щоразу, коли ми використовуємо термін коваріаційна функція стосовно стаціонарного часового ряду $\{X_t\}$, ми матимемо на увазі функцію γ_x однієї змінної, визначено

$$\gamma_x(h) := \gamma_x(h, 0) = \gamma_x(t + h, t).$$

Функцію $\gamma_x(\cdot)$ будемо називати функцією автоковаріації, а $\gamma_x(h)$ — її значенням на *lag* h .

Означення 4. Нехай $\{X_t\}$ — стаціонарний часовий ряд. Автоковаріаційна функція (ACVF) $\{X_t\}$ із затримкою h є

$$\gamma_x(h) = \text{Cov}(X_{t+h}, X_t).$$

Функція автокореляції (ACF) $\{X_t\}$ із затримкою h є

$$\rho_x(h) \equiv \frac{\gamma_x(h)}{\gamma_x(0)} = \text{Cor}(X_{t+h}, X_t).$$

1.3. Підходи до аналізу часових рядів

Часовий ряд можна визначити як упорядковану послідовність значень змінної через однакові проміжки часу. Мотивація для дослідження моделей часових рядів є подвійною:

1. Отримати розуміння «підводних» структур, які створили дані спостереження.
2. Побудувати модель і перейти до прогнозування, моніторингу або навіть зворотного зв'язку та прямого керування.

Аналіз часових рядів можна розділити на дві основні категорії залежно від типу моделі, яка може бути встановлена [3]. Це дві категорії:

- *Кінетична модель*

$$x_t = f(t).$$

Вимірювання або спостереження розглядаються як функція часу.

- *Динамічна модель*

$$x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, \dots).$$

Класичні процедури аналізу часових рядів розкладають функцію часових рядів $x_t = f(t)$ на чотири компоненти [6]:

1. Тренд – довгострокова монотонна зміна середнього рівня часового ряду.
2. Циклічна компонента – довга хвиля в часовому ряді.
3. Сезонна компонента – коливання часових рядів, які повторюються протягом певних періодів часу.
4. Залишкова компонента, яка відображає всі впливи на часовий ряд, які не пояснюються трьома іншими компонентами.

Тренд і циклічна компонента відповідають фактору згладжування, а сезонна і залишкова компоненти вносять свій внесок у циклічний фактор. Часто перед застосуванням моделей часових рядів дані потрібно перевірити та, якщо необхідно, трансформувати, щоб можна було краще інтерпретувати ряди. Це робиться для стабілізації дисперсії. Наприклад, якщо в ряді є тенденція, а стандартне відхилення прямо пропорційне середньому, то пропонується логарифмічне перетворення. І для того, щоб сезонний вплив викликав звикання, якщо в ряду є тенденція і розмір сезонного ефекту має тенденцію до збільшення із середнім значенням, тоді можна порадити трансформувати дані так, щоб сезонний ефект був постійним з року в рік [5]. Іноді також застосовується перетворення, щоб зробити дані нормально розподіленими.

Підгонка моделей часових рядів може бути амбітною справою. Існує багато методів моделювання. Ці моделі добре обговорені в [7, 8]. Вибір відповідної техніки залежить від додатка та переваг користувача.

1.4. Стаціонарність часових рядів

Стаціонарний часовий ряд – це ряд, властивості якого не залежать від часу спостереження за рядом [4]. Таким чином, часові ряди з трендами або сезонністю не є стаціонарними – тренд і сезонність впливатимуть на значення часового ряду в різні часи. З іншого боку, білий шум є стаціонарним часовим рядом – не має значення, коли ви її спостерігаєте, вона має виглядати майже однаково в будь-який момент часу.

Деякі випадки можуть вводити в оману: часовий ряд із циклічною складовою (але без тренду чи сезонності) є стаціонарним. Це пояснюється тим, що цикли не мають фіксованої довжини, тому до того, як ми почнемо спостерігати за серіями, ми не можемо бути впевнені, де будуть піки та спади циклів.

Загалом, стаціонарний часовий ряд не матиме передбачуваних патернів на тривалий період. Графіки, наведені нижче, пояснюють, що стаціонарний ряд буде приблизно горизонтальним (хоча можлива певна циклічна поведінка) із постійною дисперсією [12].

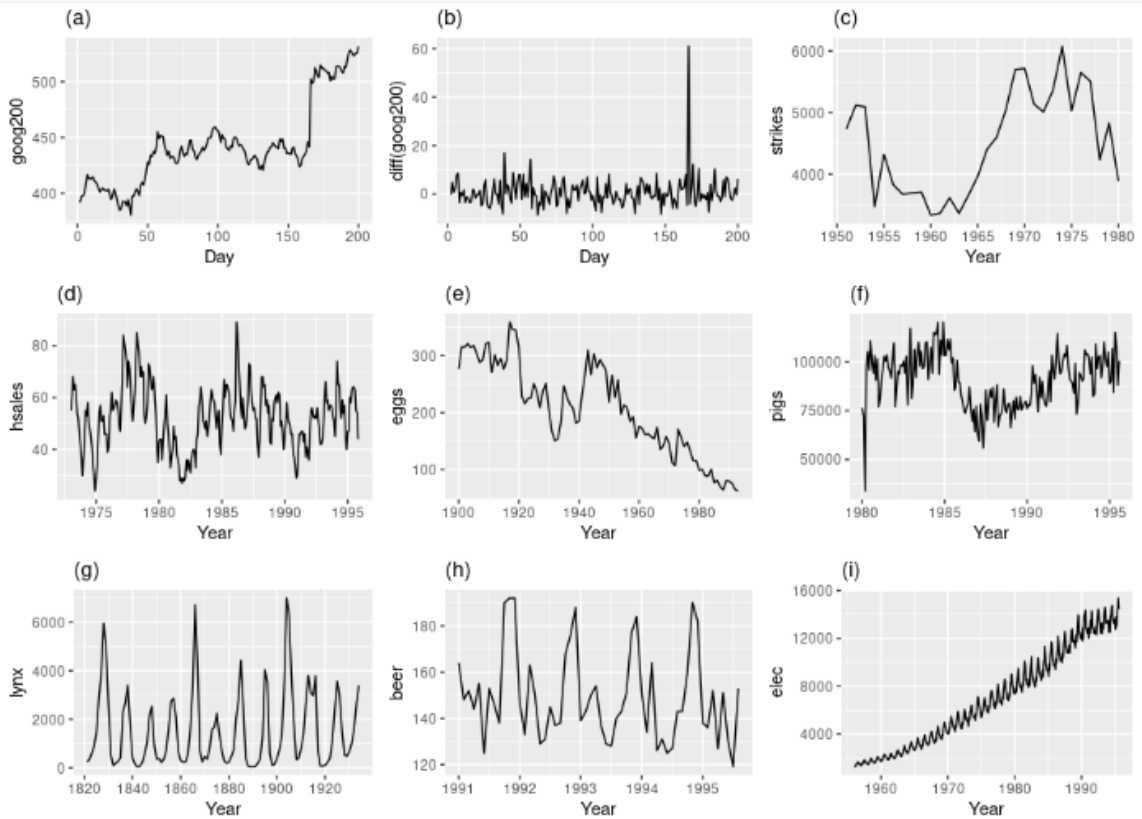


Рис. 1. Приклади стаціонарних та нестаціонарних часових рядів

На Рис. 1 зображено: (a) Курс акцій Google протягом 200 днів; (b) Щоденна зміна курсу акцій Google протягом 200 днів; (c) Річна кількість страйків у США; (d) Щомісячні продажі нових односімейних будинків, проданих у США; (e) Річна ціна дюжини яєць у США; (f) Загальна кількість свиней, забитих за місяць у Вікторії, Австралія; (g) Річна загальна кількість рисі, яка потрапила в пастку в районі річки Маккензі на північному заході Канади; (h) Щомісячне виробництво пива в Австралії; (i) Щомісячне виробництво електроенергії в Австралії.

Серед розглянутих графіків сезонна компонента присутня на зображеннях (d), (h) та (i). Тренд та зміна рівнів присутні у рядах (a), (c), (e), (f) та (i).

Збільшення дисперсії спостерігається для ряду (i). Отже, стаціонарними рядами є (b) і (g). На перший погляд може здатися, що великі цикли для ряду (g) роблять його нестационарним. Але ці цикли є аперіодичними — вони виникають, коли популяція рисі стає занадто великою для доступного корму, так що вони припиняють розмноження, а популяція падає до низького рівня, тоді відновлення їхніх джерел їжі дозволяє популяції знову зростати. Отже, ряди (b) та (g) – стаціонарні.

Варто зауважити, що ціна акцій Google була нестационарною на рисунку 1 (a), але щоденні зміни були стаціонарними на рисунку 1 (b). Це показує один спосіб зробити нестационарний часовий ряд стаціонарним – обчислити різницю між послідовними спостереженнями. Це відомо як differencing – взяття перших різниць у часовому ряді.

Такі перетворення, як логарифми, можуть допомогти стабілізувати дисперсію часового ряду. Взяття різниць може допомогти стабілізувати середнє значення часового ряду, усуваючи зміни в рівні часового ряду, а отже, усуваючи (або зменшуючи) тренд та сезонність. Окрім перегляду графіка даних у часі, графік авто кореляційної функції (ACF) також корисний для виявлення нестационарних часових рядів [14]. Для стаціонарного часового ряду ACF відносно швидко впаде до нуля, тоді як ACF нестационарних даних зменшується повільно. Крім того, для нестационарних даних значення r_1 часто велике і додатне. На рисунку 2 показано приклади ACF для нестационарного (ліворуч) та стаціонарного (праворуч) часового ряду.

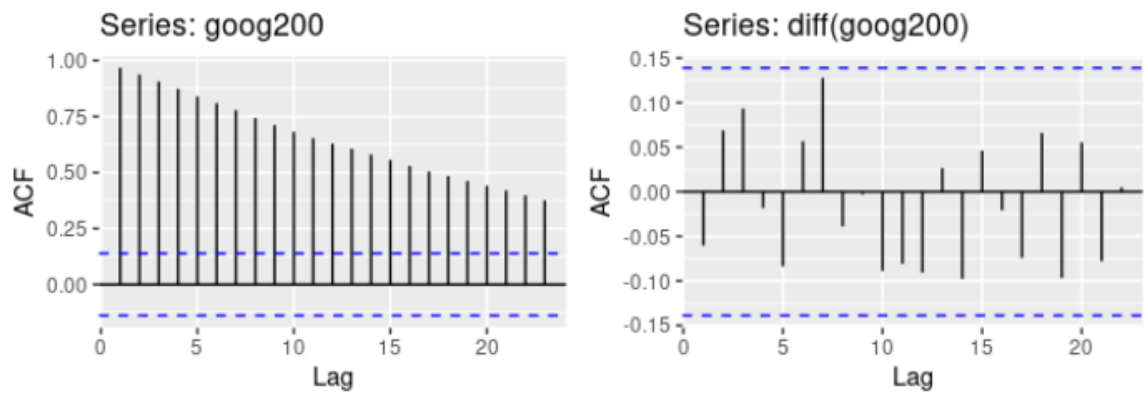


Рис. 2. Приклади ACF для стаціонарних та нестаціонарних часових рядів
 Це свідчить про те, що щоденна зміна ціни акцій Google є, по суті, випадковою величиною, яка не корелює з показниками попередніх днів.

РОЗДІЛ II. ПОБУДОВА МОДЕЛЕЙ ЧАСОВИХ РЯДІВ

2.1. Модель випадкового блукання

Модель випадкового блукання (Random walk) ще називають задачею про знаходження людини під впливом алкоголю, яку випустили у поле. Задача полягає у тому, щоб з'ясувати місце знаходження п'яного через деякий проміжок часу, якого залишили у полі. Ця модель є нестационарним часовим рядом. Забігаючи наперед, скажемо, що після усіх математичних розрахунків виявляється, що п'яний залишиться на тому ж місці. Перевіримо все вище сказане. Нехай y_t – часовий ряд, $t = 1, 2, \dots, T$. Тоді взяття першої різниці від часового ряду y_t можна подати у вигляді:

$$y'_t = y_t - y_{t-1}.$$

Ряд y'_t матиме $T - 1$ значень, бо неможливо обчислити першу різницю для початкового спостереження y'_1 . У випадку, коли y_t є білим шумом, то має місце наступна рівність:

$$y_t - y_{t-1} = \varepsilon_t,$$

де ε_t – білий шум. Переписавши рівність вище, отримуємо модель випадкового блукання

$$y_t = y_{t-1} + \varepsilon_t.$$

Моделі випадкового блукання широко використовуються для нестационарних даних, зокрема фінансових та економічних даних. Випадкові блукання зазвичай мають:

- тривалі періоди з трендами зростання або спадання;
- раптові та непередбачувані зміни напрямку.

Прогнози моделі випадкового блукання дорівнюють останньому спостереженню, оскільки майбутні рухи є непередбачуваними та з однаковою

ймовірністю будуть як вгору, так і вниз. Таким чином, модель випадкового блукання лежить в основі **наївних** прогнозів.

2.2. Наївний метод прогнозування

Розглянемо у цьому пункті також метод середніх. Цей метод прогнозування часових рядів є одним із найпростіших, проте він має очевидні недоліки. Для методу середніх прогнози всіх майбутніх значень дорівнюють середньому значенню історичних даних. Позначимо дані у вигляді y_1, \dots, y_T . Тоді прогноз можна подати у вигляді:

$$\hat{y}_{T+h|T} = \bar{y} = \frac{(y_1 + \dots + y_T)}{T}.$$

Позначення $\hat{y}_{T+h|T}$ вказує на необхідність оцінки y_{T+h} , ґрунтуючись на даних y_1, \dots, y_T .

Для наївного методу припускається, що усі прогнози є рівними останньому значенню часового ряду. Тобто

$$\hat{y}_{T+h|T} = y_T.$$

Такий наївний підхід і дав назву цьому методу. Прогнозування часових рядів за допомогою наївного методу є оптимальним, коли дані близькі до процесу випадкового блукання. Варто також описати сезонний наївний метод прогнозування. Сезонний наївний метод прогнозування використовують для даних, яким характерна сезонність. У такому випадку кожне прогнозне значення дорівнює останньому спостереженню з аналогічного сезону (попередній місяць, півріччя, рік). Формально прогноз $T + h$ можна записати у наступному вигляді:

$$\hat{y}_{T+h|T} = y_{T+h-m(k+1)},$$

де m – період сезонності, k є цілою частиною від $(h - 1)/m$, тобто кількість повних років у періоді прогнозування до моменту $T + h$.

2.3. Метод експоненціального згладжування

Найпростіший із групи методів експоненціального згладжування природно називається простим експоненціальним згладжуванням (Simple Exponential Smoothing, SES). Цей метод підходить для прогнозування даних без чіткої тенденції (тренду) або сезонної закономірності. Наприклад, дані на Рис. 3 не мають чітко вираженого тренду чи сезонності. (За останні кілька років спостерігається зростання, що може свідчити про певну тенденцію (тренд)).

Рис. 3. Виробництво палива у млн. тонн у Саудівській Аравії з 1996 по 2013

Наївний метод передбачає, що останнє спостереження є єдино важливим, а всі попередні спостереження не дають інформації на майбутнє. Це можна вважати зваженим підходом, де вся вага надається останньому спостереженню. Під «вагою» мається на увазі ваговий коефіцієнт, тобто число, яке становить частину цілого. Використовуючи метод середнього, усі майбутні прогнози дорівнюють простому середньому спостережених даних. Таким чином, середній метод припускає, що всі спостереження мають однакову важливість (однакові вагові коефіцієнти), і надає їм однакову вагу при створенні прогнозів.

Найчастіше потрібно отримати щось середнє між цими двома крайнощами. Наприклад, може бути доцільним надавати більшу вагу недавнім спостереженням, ніж спостереженням із далекого минулого. Саме ця концепція і є ключовою для простого експоненціального згладжування. Прогнози розраховуються за допомогою зважених середніх значень, де ваги зменшуються експоненціально, оскільки спостереження надходять із більш далекого минулого – найменші ваги пов'язані з найдавнішими спостереженнями:

$$y_{T+h|T} = \alpha y_T + \alpha(1 - \alpha)y_{T-1} + \alpha(1 - \alpha)^2 y_{T-2} + \dots,$$

де $0 \leq \alpha \leq 1$ є параметром згладжування. Прогнозне значення на один крок вперед $T + 1$ – це зважене середнє значення всіх спостережень у часовому ряді y_1, \dots, y_T . Швидкість зменшення ваг контролюється параметром α . У Таб. 1 нижче показано вагові коефіцієнти, додані до спостережень для чотирьох різних значень α при прогнозуванні з використанням простого експоненціального згладжування. Варто відзначити, що сума ваг навіть для невеликого значення α буде приблизно однаковою для будь-якого прийняттого розміру вибірки.

Таблиця 1. Коефіцієнти експоненціального згладжування

	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
y_T	0.2000	0.4000	0.6000	0.8000
y_{T-1}	0.1600	0.2400	0.2400	0.1600
y_{T-2}	0.1280	0.1440	0.0960	0.0320
y_{T-3}	0.1024	0.0864	0.0384	0.0064
y_{T-4}	0.0819	0.0518	0.0154	0.0013
y_{T-5}	0.0655	0.0311	0.0061	0.0003

Для будь-якого α між 0 і 1 ваги, додані до спостережень, експоненціально зменшуються, коли рухатися назад у часі, звідси й назва «експоненціальне згладжування». Якщо α є малим (тобто близьким до 0), то більша вага надається спостереженням з більш далекого минулого. Якщо α є великим (тобто близьким до 1), більші ваги надаються останнім спостереженням. У випадку, коли $\alpha = 1$, $y_{T+1|T} = y_T$, а прогнози дорівнюють наївним прогнозам.

2.4. Модель TBATS

Для часових рядів із комплексною сезонністю загальна модель типу ETS, ARIMA тощо, не може працювати. Під комплексною сезонністю мається на

увазі *нецілочисельна сезонність*. Наприклад, тижневий часовий ряд з річною сезонністю $365,25/7 \approx 52,18$. До комплексної сезонності відносять також *множинну сезонність*. Наприклад, погодинний часовий ряд. Справа у тому, що забагато сезонних компонентів призводить до надмірної параметризації.

Ідея методу TBATS полягає у тому, щоб модифікувати моделі простору станів для експоненціального згладжування:

1. Перетворення Бокса-Кокса дозволяють «нелінійність».
2. Використання процесу ARMA для моделювання помилки.
3. Тригонометричне представлення сезонних компонент на основі рядів Фур'є.

Модель BATS, тобто та, яка не допускає тригонометричної сезонності, має наступний вигляд:

- Перетворення Бокса-Кокса

$$y_t^{(\omega)} = \begin{cases} \frac{y_t^\omega}{\omega}, & \omega \neq 0, \\ \log y_t, & \omega = 0, \end{cases}$$

- Структура State Space Model

$$y_t^{(\omega)} = l_{t-1} + \phi b_{t-1} + \sum_{i=1}^T s_{t-m_i}^{(i)} + d_t,$$

- Компонент рівня

$$l_t = l_{t-1} + \phi b_{t-1} + \alpha d_t,$$

- Трендова складова

$$b_t = (1 - \phi)b + \phi b_{t-1} + \beta d_t,$$

- Сезонна складова

$$s_t^{(i)} = s_{t-m_i}^{(i)} + \gamma_i d_t,$$

- $ARMA(p, q)$ процес для залишків

$$d_t = \sum_{i=1}^p \varphi_i d_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t,$$

де α, β, γ – параметри згладжування, ϕ – параметр затухання для затухаючого тренду, $\varepsilon_t \sim \text{Gaussian} - WN(0, \sigma^2)$.

Модель BATS розглядає сезонний компонент так само, як і модель ETS. Це означає, що BATS може моделювати лише цілі періоди. Підхід, використаний у BATS, вимагає початкових станів m_i для сезону i , якщо цей сезон довгий, модель може стати нерозв'язною. Тому модель BATS не розглядає комплексну проблему сезонності.

Різниця між моделями BATS та TBATS полягає лише у одній букві «Т». Ця буква позначає тригонометричне представлення сезонних компонент, які використовують ряди Фур'є. TBATS модель має наступний вигляд:

- Часовий ряд у момент часу t (Box-Cox transformed)

$$y_t^{(\lambda)} = l_{t-1} + \phi b_{t-1} + \sum_{i=1}^T s_{t-m_i}^{(i)} + d_t,$$

- Компонент рівня

$$l_t = l_{t-1} + \phi b_{t-1} + \alpha d_t,$$

- Трендова складова

$$b_t = \phi b_{t-1} + \beta d_t,$$

- $ARMA(p, q)$ процес для залишків

$$d_t = \sum_{i=1}^p \varphi_i d_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t,$$

- Сезонна складова

$$s_t^{(i)} = \sum_{j=1}^{(k_i)} s_{j,t}^{(i)},$$

$$s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \cos(\omega_i) + s_{j,t-1}^{*(i)} \sin(\omega_i) + \gamma_1^{(i)} d_t,$$

$$s_{j,t}^{*(i)} = -s_{j,t-1}^{(i)} \sin(\omega_i) + s_{j,t-1}^{*(i)} \cos(\omega_i) + \gamma_2^{(i)} d_t,$$

$$\omega_i = 2\pi j/m_i,$$

де T – кількість сезонів, m_i – довжина i -го сезонного періоду, k_i – кількість «harmonics» для i -го сезонного періоду, λ – трансформація Бокса-Кокса, α, β – параметри згладжування, ϕ – параметр згладжування тренду, φ_i, θ_i – $ARMA(p, q)$ коефіцієнти, $\gamma_1^{(i)}, \gamma_2^{(i)}$ – сезонне згладжування (два параметри для кожного періоду). Модель TBATS обирає найкращу модель за допомогою інформаційного критерію Акаїке.

2.5. ARIMA

Модель ARIMA (Autoregressive Integrated Moving Average Model) являє собою комбінацію авторегресії та моделі ковзного середнього. Це зрозуміло із назви моделі, де перші дві літери AR вказують авто регресійну частину, а останні дві літери MA – частину ковзного середнього. Буква «I» позначає інтеграцію, тобто кількість диференціацій або взяття перших різниць, які потрібно взяти від вхідного часового ряду, щоб перетворити його у стаціонарний. На практиці кількість диференціацій зазвичай дорівнює 1 або 2. Тобто, модель $ARIMA(p, d, q)$ є моделлю для нестационарних часових рядів. Тут p – кількість доданків у авторегресійній частині, q – кількість доданків у моделі ковзного середнього, d – кількість диференціацій. У випадку, коли $d =$

0, то вхідний часовий ряд є стаціонарним. Таким чином модель ARIMA має наступний вигляд:

$$y_t = c + \alpha_1 y_{t-1} + \dots + \alpha_p y_{t-p} + \beta_1 \varepsilon_{t-1} + \beta_2 \varepsilon_{t-2} + \dots + \beta_q \varepsilon_{t-q} + \varepsilon_t,$$

де $\varepsilon \sim \text{Gaussian WN}(0, \sigma^2)$, $c = \text{const}$, α_i, β_i – коефіцієнти моделі.

РОЗДІЛ III. ДЕКОМПОЗИЦІЯ ЧАСОВОГО РЯДУ

3.1. Оцінка та видалення тренду та сезонних компонентів

Першим кроком в аналізі будь-якого часового ряду є побудова графіка даних. Якщо в ряді є будь-які очевидні розриви, такі як раптова зміна рівня, може бути доцільним проаналізувати ряд, спочатку розбивши його на однорідні сегменти. Якщо є сторонні спостереження, їх слід ретельно вивчити, щоб перевірити, чи є якесь обґрунтування для їх відхилення (наприклад, якщо спостереження було неправильно записано). Перевірка графіка може також припустити можливість представлення даних як реалізації процесу (класична модель декомпозиції):

$$X_1 = m_1 + s_1 + Y_1, \quad (1)$$

де m_1 – функція, що повільно змінюється, відома як тренд, s_1 – функція з відомим періодом d під назвою сезонна складова, і Y_1 – компонент випадкового шуму. Якщо сезонні коливання та шумові коливання збільшуються з рівнем процесу, то часто використовується попереднє перетворення даних, щоб зробити перетворені дані більш сумісними з моделлю (1).

Трансформовані дані не виявляють зростаючих коливань зі збільшенням рівня, які були очевидними у вихідних даних. Це говорить про те, що модель (1) більше підходить для трансформованої серії, ніж для оригінальної серії. Далі розглянемо деякі методи оцінки компонентів m_1 і s_1 у часовому ряді.

Наша мета полягає в тому, щоб оцінити та виокремити детерміновані компоненти m_1 та s_1 з вхідного часового ряду. Якщо залишки або випадкова компонента Y_1 виявиться стаціонарним часовим рядом, то прогнозування часового ряду у такому випадку здійснюється доволі просто. Далі можна використати теорію випадкових процесів, для знаходження задовільної імовірнісної моделі для процесу Y_1 , проаналізувати його властивості та

використовувати цю ймовірнісну модель разом із компонентами m_1 та s_1 для прогнозування та моделювання часового ряду $\{X_t\}$ [17].

Інший підхід, розроблений Боксом і Дженкінсом (1976), полягає в повторному застосуванні операторів різниць або зсуву до ряду, $\{X_t\}$ поки відповідні спостереження не стануть схожими на реалізацію деякого стаціонарного часового ряду $\{W_t\}$. Тоді ми можемо використовувати теорію стаціонарних процесів для моделювання, аналізу та прогнозування, $\{W_t\}$, а отже, і вихідного процесу. Опишемо далі у наступних двох підпунктах методи видалення та оцінки тренду за відсутності сезонності.

3.2. Оцінка та видалення тренду за відсутності сезонності

Розглянемо випадок, коли у моделі (1) немає сезонної компоненти. Тоді модель (1) є несезонною моделлю із трендом та має наступний вигляд:

$$X_t = m_t + Y_t, \quad t = 1, \dots, n, \quad (2)$$

де $EY_t = 0$. Якщо $EY_t \neq 0$, то ми можемо замінити m_t і Y_t на $m_t + EY_t$ і $Y_t - EY_t$, відповідно.

3.2.1. Оцінка тренду

Ковзне середнє та спектральне згладжування по суті є непараметричними методами для оцінки тренду (тенденції або сигналу), а не для побудови моделі. Спеціальні згладжувальні фільтри також можуть бути розроблені для видалення періодичних компонентів. Вибір фільтра згладжування вимагає певної частки суб'єктивного судження, тому рекомендується спробувати різноманітні фільтри, щоб отримати гарне уявлення про основну тенденцію. Експоненціальне згладжування, оскільки воно базується лише на ковзному середньому *минулих* значень, часто використовується для прогнозування, а

згладжене поточне значення використовується як прогноз наступного значення.

Для побудови *моделі* даних (без сезонності) є два загальні підходи. Перший полягає в тому, щоб підібрати поліноміальний тренд (за методом найменших квадратів), потім відняти підігнаний тренд із даних і знайти відповідну модель стаціонарного часового ряду для залишків. Інший – полягає в тому, щоб усунути тренд шляхом розрізнення, а потім знайти відповідну стаціонарну модель для різницевого ряду. Перевага останнього методу полягає в тому, що він зазвичай вимагає оцінки меншої кількості параметрів і не ґрунтується на припущенні тенденції, яка залишається фіксованою протягом усього періоду спостереження.

(а) *Згладжування за допомогою скінченного фільтра ковзного середнього.*

Нехай q є невід'ємним цілим числом і розглянемо двобічне ковзне середнє

$$W_t = (2q + 1)^{-1} \sum_{j=-q}^q X_{t-j} \quad (3)$$

Тоді для $q + 1 \leq t \leq n - q$,

$$W_t = (2q + 1)^{-1} \sum_{j=-q}^q m_{t-j} + (2q + 1)^{-1} \sum_{j=-q}^q Y_{t-j} \approx m_t, \quad (4)$$

припускаючи, що m_t є приблизно лінійним на інтервалі $[t - q, t + q]$ і що середнє значення похибок у цьому інтервалі є близьким до нуля.

Таким чином, ковзне середнє дає нам оцінки

$$\hat{m}_t = (2q + 1)^{-1} \sum_{j=-q}^q X_{t-j}, \quad q + 1 \leq t \leq n - q. \quad (5)$$

Оскільки X_t не спостерігається для $t \leq 0$ або $t > n$, ми не можемо використовувати модель (5) для $t \leq q$ або $t > n - q$. Рівняння (5) є лише одним із багатьох, які можна використовувати для згладжування. Для великих q

за умови, що $(2q + 1)^{-1} \sum_{j=-q}^q Y Y_{t-j} \approx 0$, зменшується шум, і в той же час у такому випадку лінійні функції тренду $m_t = c_0 + c_1 t$ проходять без спотворень. Однак ми повинні остерігатися вибору q занадто великим, оскільки якщо m_t не є лінійним, то відфільтрований процес, хоч і плавний, не буде хорошою оцінкою m_t . Завдяки розумному вибору вагових коефіцієнтів $\{a_j\}$ можна розробити фільтр, який не тільки буде ефективним для послаблення шуму в даних, але також дозволить більшому класу трендових функцій проходити без спотворень. Середнє – це фільтр, який пропускає поліноми третього ступеня без спотворень. Його ваги

$$a_j = 0, |j| > 7,$$

з

$$a_j = a_{-j}, |j| \leq 7,$$

і

$$[a_0, a_1, \dots, a_7] = \frac{1}{320} [74, 67, 46, 21, 3, -5, -6, -3].$$

Застосовуючи до процесу (2) з $m_t = c_0 + c_1 t + c_2 t^2 + c_3 t^3$, отримуємо

$$\sum_{j=-7}^7 a_j X_{t-j} = \sum_{j=-7}^7 a_j m_{t-j} + \sum_{j=-7}^7 a_j Y_{t-j} \approx \sum_{j=-7}^7 a_j m_{t-j} = m_t,$$

де останній крок залежить від передбачуваної форми m_t . Подальші відомості щодо цього та інших фільтрів згладжування можна знайти в **Kendall and Stuart (1976)**.

(b) *Експоненціальне згладжування.* Для будь-яких фіксованих $\alpha \in [0, 1]$

односторонніх ковзних середніх, \hat{m}_t , $t = 1, \dots, n$ визначених рекурсіями

$$\hat{m}_t = \alpha X_t + (1 - \alpha) \hat{m}_{t-1}, \quad t = 2, \dots, n,$$

$$\hat{m}_1 = X_1$$

можна обчислити за допомогою пакетів прикладних програм для статистичних розрахунків. Останні два рівняння часто називають

експоненціальним згладжуванням, оскільки рекурсії означають, що для $t \geq 2$, $\hat{m}_t = \sum_{j=0}^{t-2} \alpha(1-\alpha)^j X_{t-j} + (1-\alpha)^{t-1} X_1$, зважене ковзне середнє X_t, X_{t-1}, \dots , з вагами, що зменшуються експоненціально.

(с) *Згладжування шляхом усунення високочастотних компонентів.*

Дана опція є широко застосовною у пакетах прикладних програм. Таким чином можна згладити довільний ряд шляхом усунення високочастотних компонентів за допомогою розкладання в ряд Фур'є.

(d) *Поліноміальне згладжування.* Тренд форми $m_t = a_0 + a_1 t + a_2 t^2$ можна підігнати до даних $\{x_1, \dots, x_n\}$, вибравши параметри a_0, a_1 та a_2 таким чином, щоб мінімізувати суму квадратів, $\sum_{t=1}^n (x_t - m_t)^2$. Метод найменших квадратів також можна використовувати для оцінки трендів поліномів вищого порядку аналогічним способом же чином. Також можливо провести узагальнену оцінку за методом найменших квадратів, у якій враховується кореляція між залишками.

3.2.2. Видалення тренду за допомогою диференціації (взяття перших різниць)

Замість того, щоб намагатися усунути шум шляхом згладжування, як у першому описаному підході, тепер ми намагаємося усунути тренд шляхом диференціації або знаходження перших різниць ряду. Визначаємо різницевий оператор із зсувом на 1 крок (лаг рівний 1) через ∇ наступним чином

$$\nabla X_t = X_t - X_{t-1} = (1 - B)X_t, \quad (6)$$

де B – оператор зсуву назад,

$$BX_t = X_{t-1}.$$

Потужності операторів B і ∇ визначаються очевидним чином, тобто $B^j(X_t) = X_{t-j}$ і $\nabla^j(X_t) = \nabla(\nabla^{j-1}(X_t))$, $j \geq 1$, з $\nabla^0(X_t) = X_t$. Поліноми в B і ∇ обробляються точно так само, як і поліноміальні функції дійсних змінних.

Наприклад,

$$\begin{aligned}\nabla^2 X_t &= \nabla(\nabla(X_t)) = (1 - B)(1 - B)X_t = \\ &= (1 - 2B + B^2)X_t = X_t - 2X_{t-1} + X_{t-2}.\end{aligned}$$

Якщо оператор ∇ застосувати до функції лінійного тренду, $m_t = c_0 + c_1 t$, то ми отримаємо сталу функцію $\nabla m_t = m_t - m_{t-1} = c_0 + c_1 t - (c_0 + c_1(t-1)) = c_1$. Таким же чином будь-який поліноміальний тренд степеня k можна звести до константи за допомогою оператора ∇^k . Наприклад, якщо $X_t = m_t + Y_t$, де $m_t = \sum_{j=0}^k c_j t^j$ і Y_t є стаціонарним із середнім нулем, застосування ∇^k дає

$$\nabla^k X_t = k! c_k + \nabla^k Y_t,$$

стаціонарний процес із середнім $k! c_k$. Ці міркування свідчать про можливість, враховуючи будь-яку послідовність $\{x_t\}$ даних, багаторазового застосування оператора ∇ доки ми не знайдемо послідовність $\{\nabla^k x_t\}$, яку правдоподібно можна змоделювати як реалізацію стаціонарного процесу. На практиці часто виявляється, що порядок k необхідної різниці досить малий, часто один або два. (Це ґрунтується на тому факті, що багато функцій можна добре апроксимувати на інтервалі скінченної довжини поліномом досить низького степеня).

3.3. Оцінка та видалення тренду та сезонності

Методи, описані для оцінки та усунення тренду, можуть бути адаптовані природним чином для усунення як тренду, так і сезонності в загальній моделі, визначеній наступним чином. Класична модель декомпозиції має наступний вигляд:

$$X_t = m_t + s_t + Y_t, t = 1, \dots, n, \quad (7)$$

де

$$EY_t = 0, \quad s_{t+d} = s_t, \quad \sum_{j=1}^d s_j = 0.$$

Підхід 1. Оцінка тренду та сезонної компоненти

Припустимо, що у нас є спостереження. $\{x_1, \dots, x_n\}$. Тренд спочатку оцінюється шляхом застосування фільтра ковзного середнього, спеціально вибраного для усунення сезонної складової та послаблення шуму. Якщо період d є парним, скажімо $d = 2q$, тоді ми використовуємо

$$\hat{m}_t = \frac{(0.5x_{t-q} + x_{t-q+1} + \dots + x_{t+q-1} + 0.5x_{t+q})}{d}, \quad q < t \leq n - q.$$

Якщо період – непарний, скажімо $d = 2q + 1$, ми використовуємо просте ковзне середнє (5). Другим кроком є оцінка сезонної складової. Для кожного $k = 1, \dots, d$, ми обчислюємо середнє значення w_k відхилень. $\{(x_{k+jd} - \hat{m}_{k+jd}), q < k + jd \leq n - q\}$. Оскільки сума цих середніх відхилень не обов'язково дорівнює нулю, ми оцінюємо сезонний компонент s_k як

$$\hat{s}_k = w_k - d^{-1} \sum_{i=1}^d w_i, \quad k = 1, \dots, d,$$

$$\hat{s}_k = \hat{s}_{k-d}, \quad k > d.$$

Десезонні дані визначаються як вихідний ряд із вилученою оцінкою сезонної компоненти, *тобто*

$$d_t = x_t - \hat{s}_k, \quad t = 1, \dots, n.$$

Нарешті, ми повторно оцінюємо тренд на основі десезонних даних $\{d_t\}$ за допомогою одного з уже описаних методів. Статистичні програми дозволяють підганяти поліноміальний тренд методу найменших квадратів \hat{m} до десезонізованого ряду. З огляду на цей оцінений тренд та оцінку сезонної складової, оцінка шуму визначається наступним чином

$$\hat{Y}_t = x_t - \hat{m}_t - \hat{s}_t, \quad t = 1, \dots, n.$$

Повторна оцінка тренду виконується для того, щоб мати параметричну

форму для тренду, яку можна екстраполювати для цілей прогнозування та моделювання.

Підхід 2. Усунення тренду і сезонних компонентів шляхом диференціації

Техніка розрізнення, яку ми застосували раніше до несезонних даних, може бути адаптована для роботи з сезонністю періоду d шляхом введення оператора розрізнення $lag - d \nabla_d$ визначеного наступним чином

$$\nabla_d X_t = X_t - X_{t-d} = (1 - B^d)X_t.$$

(Цей оператор не слід плутати з оператором $\nabla^d = (1 - B)^d$, який визначено раніше у роботі).

Застосування оператора ∇_d до моделі

$$X_t = m_t + s_t + Y_t,$$

де $\{s_t\}$ має період d , отримуємо

$$\nabla_d X_t = m_t - m_{t-d} + Y_t - Y_{t-d},$$

що дає розкладання різниці $\nabla_d X_t$ на тренд $(m_t - m_{t-d})$ та шумовий термін $(Y_t - Y_{t-d})$. Тренд $m_t - m_{t-d}$, можна видалити за допомогою вже описаних методів, зокрема шляхом застосування потужності оператора ∇ .

Отже, у цьому розділі ми описали різноманітні методи оцінки та/або усунення тенденції та сезонності. Конкретний метод, обраний для будь-якого заданого набору даних, залежатиме від ряду факторів, у тому числі від того, чи потрібні оцінки компонентів ряду, а також від того, чи здається, що дані містять сезонну компоненту, яка не змінюється з часом.

РОЗДІЛ IV. ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1. Опис вхідних даних

Набір даних, використаний у даному дослідженні, отримано з сайту федеральної системи економічних даних США <https://fred.stlouisfed.org/>. Крім того даний набір даних є доступним у пакеті RStudio у бібліотеці ggplot2 під назвою “economics” у форматі “wide” та під назвою “economics_long” у форматі “long”. У даному дослідженні використано набір даних “economics”, який містить 574 рядки та 6 змінних:

Назва змінної в наборі даних	Опис значення змінної	Посилання на сайт, звідки вказана змінна була отримана
date	перше число кожного місяця, починаючи з січня 1968 року	
pce	витрати на особисте споживання, у мільярдах доларів	https://fred.stlouisfed.org/series/PCE
pop	загальна чисельність населення, тис. чол.	https://fred.stlouisfed.org/series/POP
psavert	норма особистих заощаджень	https://fred.stlouisfed.org/series/PSAVERT/
uempmed	середня тривалість безробіття, тижні	https://fred.stlouisfed.org/series/UEMPMED
unemploy	кількість безробітних у тисячах (залежна змінна)	https://fred.stlouisfed.org/series/UNEMPLOY

Незважаючи на те, що набір даних містить шість змінних, основна увага у дослідженні зосереджена на стовпцях «date» та «unemploy», оскільки у роботі розглядається однофакторне прогнозування часових рядів. Дані містять інформацію про безробіття з липня 1967 року по квітень 2015 року. Всі записи проведено 1-го числа кожного місяця.

4.2. Аналіз результатів

У цьому пункті покроково опишемо роботу кожного з методів прогнозування часових рядів за допомогою мови статистичного програмування R Programming.

Безробіття є основною соціально-економічною та політичною проблемою для будь-якої країни, а отже, боротьба з нею є головним завданням будь-якого уряду. Спрогнозуємо рівень безробіття на дванадцять місяців. Для початку розіб'ємо дані на навчальну та тестову вибірки для перевірки отриманої моделі. Важливо відмітити, що тестовий набір даних повинен складатися із 12 рядків, оскільки ми створюватимемо моделі для прогнозування на 12 місяців вперед. Кожна із п'яти моделей будується на навчальному наборі даних та перевіряється на тестовому наборі.

Для того, щоб прогнозування майбутніх значень часового ряду стало доступною опцією необхідно, щоб вхідні дані були часовим рядом. У середовищі R це можливо зробити з використанням функції `ts`:

```
dat_ts <- ts(dat_train[, 6], start = c(1968, 1), end = c(2013, 12), frequency = 12)
```

Аргументи «start» і «end» визначають час першого і останнього спостереження відповідно. Аргумент «frequency» визначає кількість спостережень за одиницю часу.

Для обчислення середньої абсолютної відсоткової помилки (MAPE) створимо допоміжну функцію, яка використовуватиметься для оцінки ефективності моделей прогнозування. Чим нижче значення MAPE, тим кращою є модель прогнозування.

Підготувавши дані та функцію MAPE, перейдемо до методів прогнозування.

4.2.1. Naive Forecasting Method

Найпростішим методом прогнозування є використання останнього спостереження у якості прогнозованого значення для наступного спостереження. Такий підхід називається наївним прогнозом і може бути реалізованим за допомогою функції “naive()”. Цей метод може бути не найкращим методом прогнозування, але він часто дає корисний орієнтир для інших, більш поширених методів прогнозування.

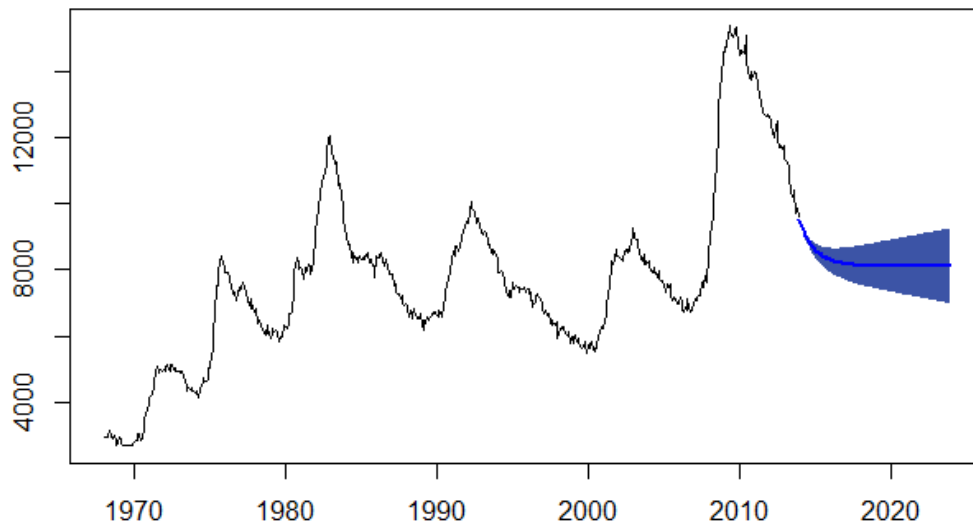
Перший рядок коду нижче зчитує об’єкт часового ряду «dat_ts» і створює наївну модель прогнозування. Другий аргумент «h» визначає кількість значень, які потрібно спрогнозувати, у нашому випадку – 12. У другому рядку друкується підсумок моделі, а також прогнозоване значення на наступні 12 місяців. Наведені результати показали, що наївний метод передбачає однакове значення для всіх 12 місяців прогнозування. Для оцінки продуктивності моделі на тестових даних використаємо це прогнозоване значення. Помилка MAPE при прогнозуванні наївним методом становить 8,5%.

4.2.2. Simple Exponential Smoothing

Методи експоненціального згладжування є розширенням простого методу, за якого прогнози створюються з використанням зважених середніх значень минулих спостережень, причому вагові коефіцієнти зменшуються експоненціально, коли спостереження стають «старшими». Простіше кажучи, більшу вагу надають останнім спостереженням. Значення параметра згладжування для кожного рівня визначається параметром «альфа».

Результати показують, що просте експоненціальне згладжування дає однакове значення для всіх прогнозів. Оскільки значення альфа близько до 1, прогнози ближчі до останніх спостережень. Оцінка продуктивності моделі на тестових даних становить 8,5%. Результати прогнозування з допомогою методу простого експоненціального згладжування можна побачити на малюнку нижче.

Forecasts from ETS(M,Ad,N)



4.2.3. Holt's Trend Method

Метод прогнозування Хольта є розширенням методу простого експоненціального згладжування, який враховує компонент тенденції під час створення прогнозів. Цей метод включає два рівняння згладжування, одне – для компонента рівня, а інше – для тренду.

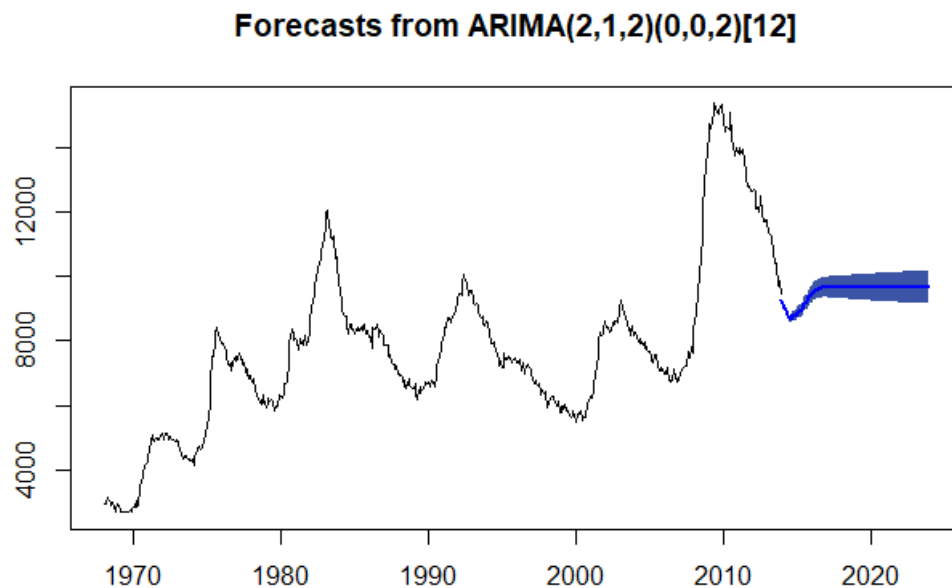
Результат помилки MAPE для побудованої моделі Хольта на навчальній вибірці становить 2,1%. Для тестових даних помилка MAPE становить 6,6%, що є кращим показником у порівнянні з попередніми двома моделями.

4.2.4. ARIMA

Моделювання ARIMA є одним із найпопулярніших підходів до прогнозування часових рядів. У той час як моделі експоненціального згладжування базуються на описі тенденції та сезонності в даних, моделі ARIMA спрямовані на опис автокореляції в даних. Функція «`auto.arima()`» у R

використовується для побудови моделей ARIMA за допомогою варіанту алгоритму Хайндмана-Кхандакара, який поєднує тести на одиничний корінь, мінімізацію AIC і MLE для отримання моделі ARIMA.

Для початку створюємо модель ARIMA та зберігаємо її в об'єкті «arima_model». Далі робимо прогнози на наступні 12 місяців. Отримані результати показали, що MAPE для навчальної вибірки становить 1,94%. Оцінка продуктивності моделі на тестових даних показує, що помилка MAPE становить 2,1%, що є кращим показником у порівнянні з усіма попередніми моделями.



4.2.5. TBATS

Модель TBATS поєднує кілька компонентів з різних методів, які вже описувалися у даному дослідженні. Таке поєднання робить TBATS дуже хорошим вибором для прогнозування. TBATS складається з наступних елементів:

T: Тригонометричні умови для сезонності

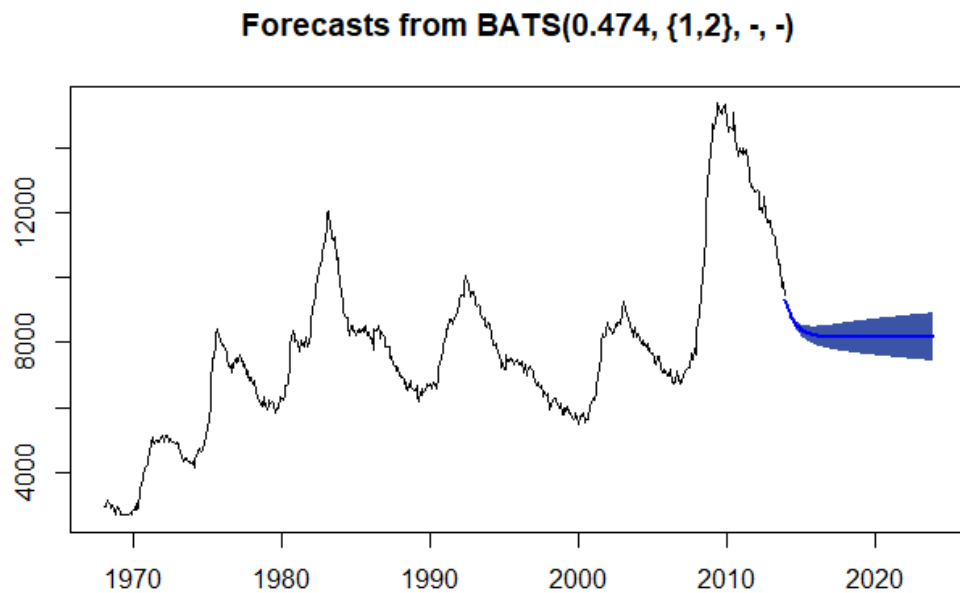
B: Перетворення Бокса-Кокса для неоднорідності

A: Помилки ARMA для короткострокової динаміки

T: Тренд

S: Сезонний (включаючи множинні та нецілі періоди)

Спочатку відбувається створення моделі TBATS та її збереження в об'єкті 'model_tbats'. На наступному кроці друкуються зведення та прогнози на наступні 12 місяців. Похибка MAPE на даних тестування для моделі TBATS становить 2,2%, що близько до продуктивності моделі ARIMA.



Висновки

У даній дипломній роботі розглянуто 5 методів прогнозування часових рядів: найвний метод, просте експоненціальне згладжування, метод Холта, ARIMA, TBATS. Ефективність прогнозування кожної із моделей визначено за допомогою середньої абсолютної відсоткової помилки – MAPE. MAPE обчислено у вигляді різниці між даними, отриманими за допомогою прогнозування та реальними даними. Найкращі результати прогнозування показали моделі ARIMA (MAPE = 2.1%) та TBATS (MAPE = 2.2%). Методу Холта (MAPE = 6,6%) належить третє місце серед п'яти розглянутих у прогнозуванні часового ряду даних про рівень безробіття. Для найвного методу та простого експоненціальне згладжування MAPE співпадає та становить 8,5%.

Протягом роботи над даним дипломним проектом я навчився працювати з реальними даними – опрацьовувати пропущені значення, визначати аутлаєри, виводити описові статистики тощо; будувати моделі часових рядів – визначати тренд, циклічні, сезонні та випадкові компоненти, проводити декомпозицію часового ряду, прогнозувати майбутні значення ряду на основі попередніх значень.

Список використаних джерел

- [1] An algorithm for solving the problem of forecasting / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatiuk // Aviation. -2013. №1(17), p. 9-13.
- [2] Victor Sineglazov, Elena Chumachenko & Vladyslav Gorbatiuk (2014) Using a mixture of experts' approach to solve the forecasting task, Aviation, 18:3, p. 129-133.
- [3] Gorbatiuk, Vladyslav; Sineglazov, Victor; Chumachenko, Olena. A method for building a forecasting model with dynamic weights. Eastern-European Journal of Enterprise Technologies, [S.l.], v. 2, n. 4(68), p. 4-8, apr. 2014. ISSN 1729-4061.
- [4] Алгоритм решения задачи прогнозирования / Е.И. Чумаченко, В.С. Горбатюк // Штучний інтелект. – 2012. – № 2., С. 24-31.
- [5] Applying Different Neural Network's Topologies to the Forecasting Task – Gorbatiuk, Vladyslav; Sineglazov, Victor; Chumachenko, Olena. – International Conference in Inductive Modelling ICIM' 2013, С. 217-220.
- [6] One approach for the forecasting task solution – Gorbatiuk, Vladyslav; Sineglazov, Victor; Chumachenko, Olena. – V Всесвітній конгрес «Авіація у XXI столітті» – «Безпека в авіації та космічні технології», р. 3.5.49-3.5.53.
- [7] Method for predicting failure risk of UAV navigation systems (2012) – Gorbatiuk, Vladyslav; Chumachenko, Olena. – Methods and systems of navigation and motion control (MSNMC 2012), p. 63-65.
- [8] Francis X. Diebold (2007), Elements of Forecasting, 4th edition.
- [9] Box, George; Jenkins, Gwilym (1970). Time Series Analysis: Forecasting and Control. San Francisco: Holden-Day.
- [10] Fisher, R. A. (1922). On the mathematical foundations of theoretical statistics. Philos. Trans. Roy. Soc. London Ser. A 222 309–368.
- [11] Kalman, R.E. (1960). "A new approach to linear filtering and prediction problems". Journal of Basic Engineering 82 (1): pp. 35–45.
- [12] Vogel, L., Owens, A., and Walsh, M., Iskustvennyj Intellekt i Evolyutsyonnoe Modelirovanie. (Artificial Intelligence and Evolutionary Modeling). Moscow: Nauka, 1969.
- [13] Klevecka Irina. Pre-Processing of Input Data of Neural Networks: The Case of Forecasting Telecommunication Network Traffic / Irina Klevecka, Janis // Telektronikk 3/4.2008. – С. 168-178.

- [14] Addison P.S. The Illustrated Wavelet Transform Handbook. – IOP, 2002.
- [15] Wong F. S. Time series forecasting using backpropagation neural networks // Neurocomputing. – 1990/91. – 2. – P. 147-159.
- [16] Groot de C., Wuertz D. Analysis of univariate time series with connectionist nets: a case study of two classical examples // Neurocomputing. – 1991. – 3. – P. 177-192.
- [17] Connor J. T., Martin R. D., Atlas L. E. Recurrent neural networks and robust time series prediction // IEEE Trans. on Neural Networks. – 1994. – 5. – P. 240–254.
- [18] Saxen H. Nonlinear time series analysis by neural networks. A case study // Int. J. Neural Systems. – 1996. – 7. – P. 195-201.
- [19] Madhavan P. G. A new recurrent neural network learning algorithm for time series prediction // J. of Intelligent Systems. – 1997. – 7. – P. 103 – 116.
- [20] Conway A. J., Macpherson K. P., Brown J. C. Delayed time series predictions with neural networks // Neurocomputing. – 1998. – 18. – P. 81 – 89.
- [21] Bodyanskiy Ye., Kolodyazniy V., Kulishova N. Generalized forecasting Sigma-Pi neural network // In “Intelligent Technologies – Theory and Applications”.–Amsterdam: IOS Press, 2002. – P.29-33.
- [22] Packard N., Crutchfield J., Farmer J., Shaw R. Geometry from a time series // Phys. Rev. Lett. – 1980. – 45. – P. 712-716.
- [23] Rosenblatt, Frank. x. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961.
- [24] Rumelhart, David E.; Hinton, Geoffrey E.; Williams, Ronald J. (8 October 1986). "Learning representations by back-propagating errors". Nature 323 (6088): 533–536.
- [25] Wan E. Temporal backpropagation for FIR neural networks // Int. Joint Conf. on Neural Networks. – V.1. – San Diego, 1990. – P. 575 – 580.
- [26] Wan E. A. Time series prediction by using a connectionist network with integral delay lines / Eds. by A. Weigend, N. Gershenfeld “Time Series Prediction. Forecasting the Future and Understanding the Past”. – SFI Studies in the Sciences of Complexity. – V. XVII. – Reading: Addison-Wesley, 1994. – P. 195 – 218.
- [27] Back A. D., Wan E. A., Lawrence S., Tsoi A. C. A unifying view of some training algorithms for multilayer perceptrons with FIR filter synapses / Eds. by J. Vlontzos, J. Hwang, E. Wilson “Neural Networks for Signal Processing 4”. – N.Y.: IEEE Press, 1994. – P. 146 – 154.

- [28] Yu H.-Y., Bang S.-Y. An improved time series prediction by applying the layer-by-layer learning method to FIR neural networks // *Neural Networks*. – 1997. – 10. – P. 1717 – 1729.
- [29] Elman, J.L. Finding structure in time. // *Cognitive Science*. – 1990. – C. 179-211.
- [30] Amir F. A. A comparison between neural-network forecasting techniques / F. A. Amir, I. S. Samir – case study: river flow forecasting. // *IEEE Transactions on neural networks*. –Vol. 10, No. 2. – 1999. – p. 402–409.
- [31] Mohsen H. Artificial neural network approach for short term load forecasting for Illam region / H. Mohsen, S. Yazdan // *World Academy of Science, Engineering and Technology* 28 2007. – p. 280–284.
- [32] Short term load forecasting with multilayer perceptron and recurrent neural networks / Muhammad Riaz Khan, Cestm Ondrusek // *Journal of ELECTRICAL ENGINEERING*, VOL. 53, NO. 1-2, 2002, p. 17-23.
- [33] Jerome T. C. Recurrent neural networks and robust time series prediction / T. C. Jerome, R. M. Douglas, L. E. Atlas // *IEEE transactions on neural networks*. – Vol. 5, No. 2. –1994. – p. 240–254.
- [34] Kenneth Levenberg (1944). «A Method for the Solution of Certain Non-Linear Problems in Least Squares». *Quarterly of Applied Mathematics* 2: p. 164–168.
- [35] U.S. General Aviation Aircraft Shipments and Sales 2012: <http://www.bga-aeroweb.com/database/Data3/US-General-Aviation-AircraftSales-and-Shipments.xls>.
- [36] Friedman, J. H. (1991). "Multivariate Adaptive Regression Splines". *The Annals of Statistics* 19: 1.
- [37] MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability* 1. University of California Press. pp. 281–297.
- [38] Data Sets for Time-Series Analysis [online] 2005. Available from Internet: <http://tracer.uc3m.es/tws/TimeSeriesWeb/repo.html>.
- [39] Lendasse, A., Oja, E., Simula, O., Verleysen, M. (2004). Time Series Prediction Competition: The CATS Benchmark. *International Joint Conference on Neural Networks*, Budapest (Hungary), IEEE, 1615-1620.

Додатки

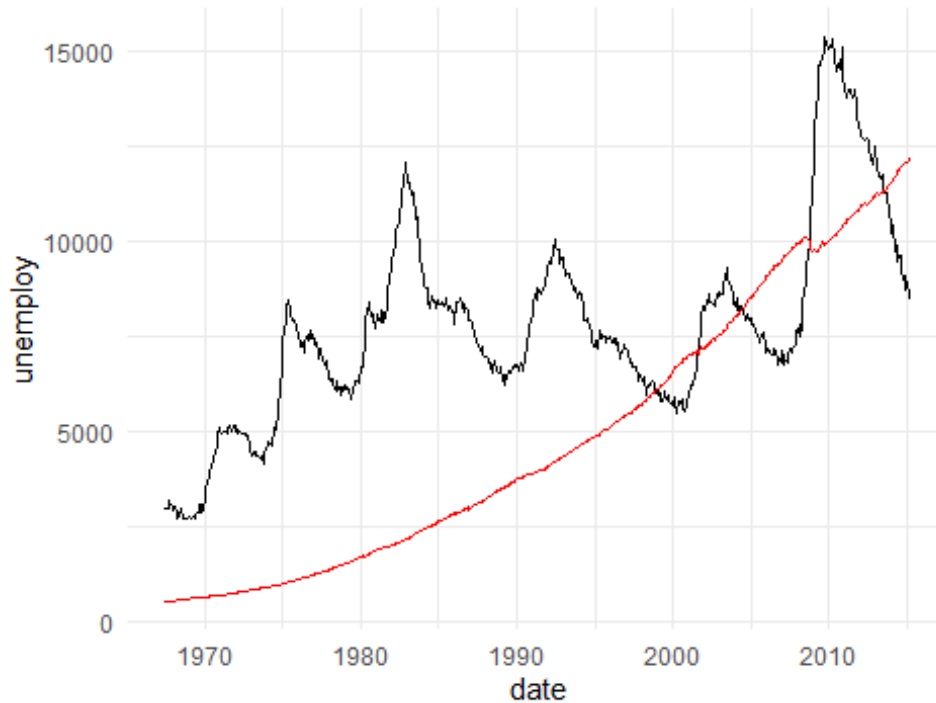
```
setwd("C:/Users/Dell/Desktop/ML_TS")
library(readr)
library(ggplot2)
library(forecast)
library(fpp2)
library(TTR)
library(dplyr)

dat <- economics
glimpse(dat)

## Rows: 574
## Columns: 6
## $ date   [3m[38;5;246m<date>[39m[23m 1967-07-01, 1967-08-01, 1967-09-01, 1967-10-01,
1967-11-0~
## $ pce    [3m[38;5;246m<dbl>[39m[23m 506.7, 509.8, 515.6, 512.2, 517.4, 525.1, 530.9, 533.6,
54~
## $ pop    [3m[38;5;246m<dbl>[39m[23m 198712, 198911, 199113, 199311, 199498, 199657,
199808, 19~
## $ psavert [3m[38;5;246m<dbl>[39m[23m 12.6, 12.6, 11.9, 12.9, 12.8, 11.8, 11.7, 12.3, 11.7,
12.3~
## $ uempmed [3m[38;5;246m<dbl>[39m[23m 4.5, 4.7, 4.6, 4.9, 4.7, 4.8, 5.1, 4.5, 4.1, 4.6, 4.4,
4.4~
## $ unemploy [3m[38;5;246m<dbl>[39m[23m 2944, 2945, 2958, 3143, 3066, 3018, 2878, 3001,
2877, 2709~

ggplot(dat) +
  geom_line(aes(x=date, y = unemploy))+
  geom_line(data = dat, aes(x=date, y = pce), col = "red")+
  theme_minimal() + labs(title = "Вхідні дані")
```

Вхідні дані



View

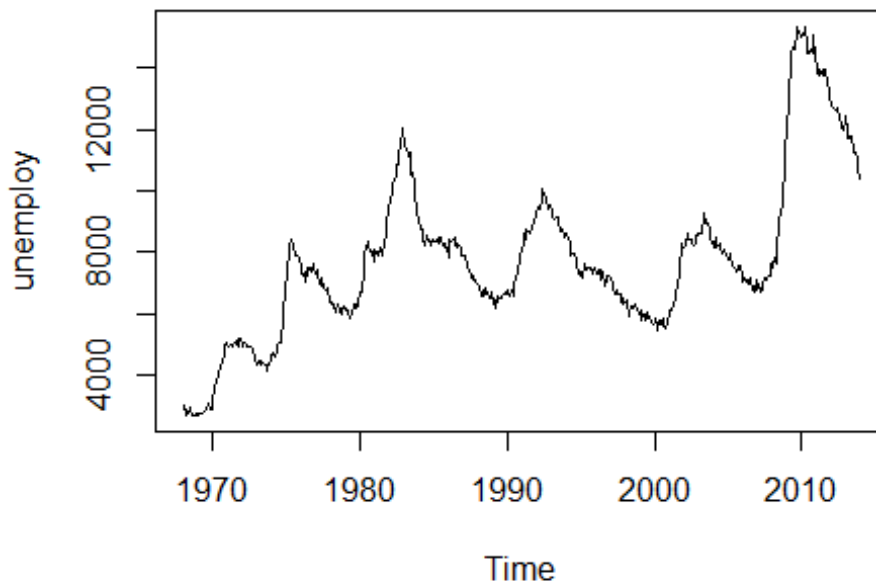
```
## function (...)  
## .rs.callAs(name, hook, original, ...)  
## <environment: 0x00000000aa74500>
```

summary(dat)

```
##   date      pce      pop      psavert  
## Min. :1967-07-01 Min. : 506.7 Min. :198712 Min. : 2.200  
## 1st Qu.:1979-06-08 1st Qu.:1578.3 1st Qu.:224896 1st Qu.: 6.400  
## Median :1991-05-16 Median :3936.8 Median :253060 Median : 8.400  
## Mean :1991-05-17 Mean : 4820.1 Mean :257160 Mean : 8.567  
## 3rd Qu.:2003-04-23 3rd Qu.: 7626.3 3rd Qu.:290291 3rd Qu.:11.100  
## Max. :2015-04-01 Max. :12193.8 Max. :320402 Max. :17.300  
##   uempmed   unemploy  
## Min. : 4.000 Min. : 2685  
## 1st Qu.: 6.000 1st Qu.: 6284  
## Median : 7.500 Median : 7494  
## Mean : 8.609 Mean : 7771  
## 3rd Qu.: 9.100 3rd Qu.: 8686  
## Max. :25.200 Max. :15352
```

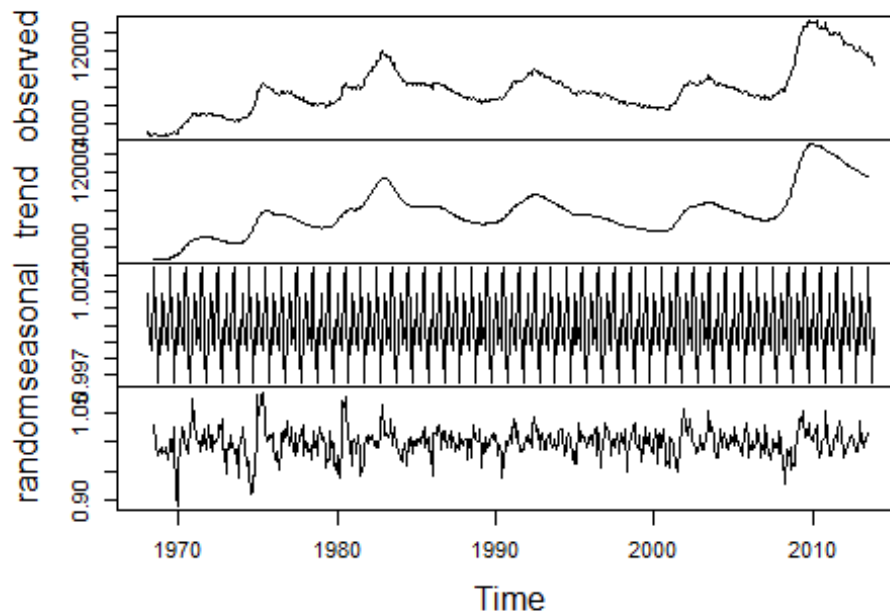
```
dat_train <- dat[7:558, 6]  
dat_test <- dat[559:570, 6]
```

```
nrow(dat_train)
## [1] 552
nrow(dat_test)
## [1] 12
dat_ts <- ts(dat_train, start = c(1968, 1), end = c(2013, 12), frequency
= 12)
plot(dat_ts)
```



```
#Decompose the Data Into Four Components
ddata <- decompose(dat_ts, "multiplicative")
plot(ddata)
```

Decomposition of multiplicative time series



```

#lines 2 to 4
mape <- function(actual,pred){
  mape <- mean(abs((actual - pred)/actual))*100
  return (mape)
}

##### Naive Forecasting Method #####

naive_mod <- naive(dat_ts, h = 12)
summary(naive_mod)

##
## Forecast method: Naive method
##
## Model Information:
## Call: naive(y = dat_ts, h = 12)
##
## Residual sd: 214.911
##
## Error measures:
##      ME  RMSE  MAE   MPE  MAPE  MASE  ACF1
## Training set 13.6588 214.911 161.0127 0.1936749 2.14771 0.1688895 0.2006429
##
## Forecasts:
##      Point Forecast  Lo 80  Hi 80  Lo 95  Hi 95
## Jan 2014          10404 10128.580 10679.42 9982.782 10825.22

```

```

## Feb 2014      10404 10014.498 10793.50 9808.308 10999.69
## Mar 2014      10404 9926.959 10881.04 9674.429 11133.57
## Apr 2014      10404 9853.161 10954.84 9561.564 11246.44
## May 2014      10404 9788.143 11019.86 9462.128 11345.87
## Jun 2014      10404 9729.363 11078.64 9372.231 11435.77
## Jul 2014      10404 9675.308 11132.69 9289.562 11518.44
## Aug 2014      10404 9624.996 11183.00 9212.616 11595.38
## Sep 2014      10404 9577.741 11230.26 9140.346 11667.65
## Oct 2014      10404 9533.047 11274.95 9071.992 11736.01
## Nov 2014      10404 9490.537 11317.46 9006.978 11801.02
## Dec 2014      10404 9449.919 11358.08 8944.859 11863.14

dat_test$naive = 10404
mape(dat_test$unemploy, dat_test$naive) ## 8.5%

## [1] 8.668593

#myforecast <- forecast(naive_mod$method, Level=c(10), h=10*12)

#plot(myforecast)

##### Simple Exponential Smoothing #####

se_model <- ses(dat_ts, h = 12)
summary(se_model)

##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
## ses(y = dat_ts, h = 12)
##
## Smoothing parameters:
##  alpha = 0.9999
##
## Initial states:
##  l = 2849.6943
##
## sigma: 215.1141
##
##  AIC  AICc  BIC
## 9418.845 9418.889 9431.786
##
## Error measures:

```

```

##           ME  RMSE  MAE  MPE  MAPE  MASE
## Training set 13.68677 214.724 160.7739 0.1951249 2.145614 0.1686391
##           ACF1
## Training set 0.2008243
##
## Forecasts:
##      Point Forecast  Lo 80  Hi 80  Lo 95  Hi 95
## Jan 2014    10404.04 10128.359 10679.72 9982.422 10825.65
## Feb 2014    10404.04 10014.188 10793.89 9807.813 11000.26
## Mar 2014    10404.04 9926.579 10881.50 9673.827 11134.25
## Apr 2014    10404.04 9852.720 10955.36 9560.870 11247.21
## May 2014    10404.04 9787.649 11020.43 9461.352 11346.72
## Jun 2014    10404.04 9728.820 11079.26 9371.381 11436.70
## Jul 2014    10404.04 9674.721 11133.36 9288.643 11519.43
## Aug 2014    10404.04 9624.366 11183.71 9211.633 11596.44
## Sep 2014    10404.04 9577.072 11231.00 9139.303 11668.77
## Oct 2014    10404.04 9532.341 11275.74 9070.892 11737.18
## Nov 2014    10404.04 9489.795 11318.28 9005.824 11802.25
## Dec 2014    10404.04 9449.143 11358.93 8943.652 11864.42

df_fc = as.data.frame(se_model)
dat_test$simplexp = df_fc$`Point Forecast`
mape(dat_test$unemploy, dat_test$simplexp)

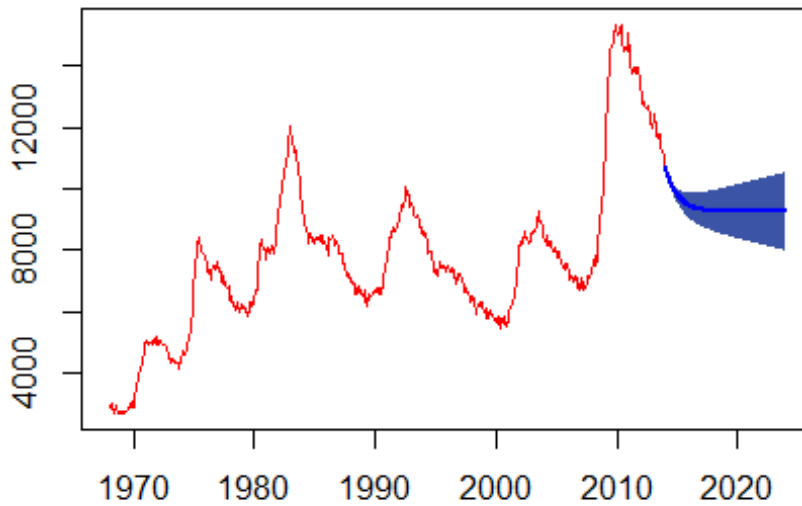
## [1] 8.668993

myforecast <- forecast(se_model$fitted, level=c(10), h=10*12)

plot(myforecast, col = "red")

```

Forecasts from ETS(M,Ad,N)



```
##### Holt's Trend Method #####
```

```
holt_model <- holt(dat_ts, h = 12)  
summary(holt_model)
```

```
##  
## Forecast method: Holt's method  
##  
## Model Information:  
## Holt's method  
##  
## Call:  
## holt(y = dat_ts, h = 12)  
##  
## Smoothing parameters:  
## alpha = 0.8728  
## beta = 0.2255  
##  
## Initial states:  
## l = 2941.2231  
## b = -0.8991  
##  
## sigma: 200.6435  
##  
## AIC AICc BIC
```

```

## 9343.953 9344.062 9365.520
##
## Error measures:
##      ME  RMSE  MAE   MPE  MAPE  MASE
## Training set -1.724098 199.9152 152.7096 0.04108105 2.054525 0.1601802
##      ACF1
## Training set -0.001071726
##
## Forecasts:
##      Point Forecast  Lo 80  Hi 80  Lo 95  Hi 95
## Jan 2014    10221.287 9964.152 10478.422 9828.033 10614.54
## Feb 2014    10005.807 9623.881 10387.733 9421.701 10589.91
## Mar 2014     9790.328 9278.739 10301.916 9007.920 10572.74
## Apr 2014     9574.848 8926.460 10223.236 8583.224 10566.47
## May 2014     9359.369 8566.499 10152.238 8146.779 10571.96
## Jun 2014     9143.889 8198.822 10088.956 7698.534 10589.24
## Jul 2014     8928.410 7823.577 10033.243 7238.713 10618.11
## Aug 2014     8712.930 7440.971 9984.889 6767.637 10658.22
## Sep 2014     8497.451 7051.234 9943.668 6285.652 10709.25
## Oct 2014     8281.971 6654.589 9909.353 5793.105 10770.84
## Nov 2014     8066.492 6251.255 9881.729 5290.326 10842.66
## Dec 2014     7851.012 5841.433 9860.592 4777.626 10924.40

df_holt = as.data.frame(holt_model)
dat_test$holt = df_holt$`Point Forecast`
mape(dat_test$unemploy, dat_test$holt)

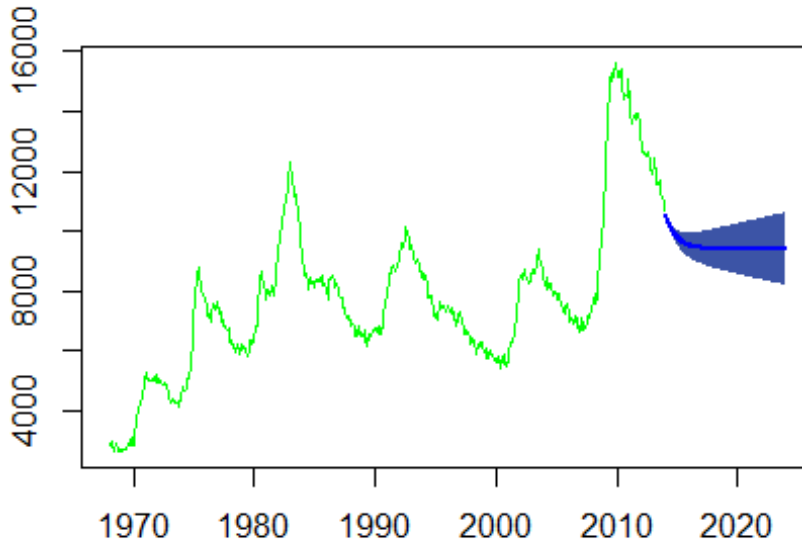
## [1] 6.044552

myforecast <- forecast(holt_model$fitted, level=c(10), h=10*12)

plot(myforecast, col = "green")

```

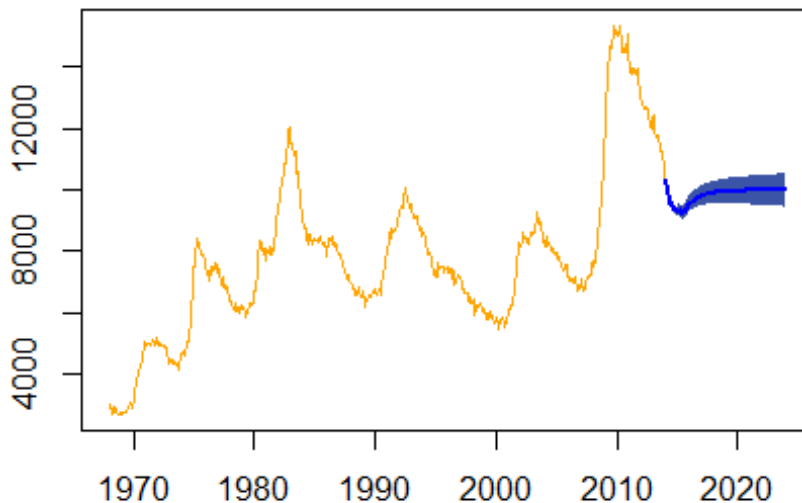

Forecasts from ETS(M,Ad,N)



```
##### ARIMA #####  
  
arima_model <- auto.arima(dat_ts)  
summary(arima_model)  
  
## Series: dat_ts  
## ARIMA(1,1,2)(0,0,2)[12]  
##  
## Coefficients:  
##   ar1  ma1  ma2  sma1  sma2  
##  0.9275 -0.8839 0.1541 -0.2311 -0.2441  
## s.e. 0.0249 0.0494 0.0444 0.0429 0.0391  
##  
## sigma^2 estimated as 35589: log likelihood=-3667.98  
## AIC=7347.96 AICc=7348.11 BIC=7373.83  
##  
## Training set error measures:  
##           ME  RMSE  MAE  MPE  MAPE  MASE  
## Training set 6.549444 187.6229 143.2256 0.1100776 1.938577 0.1502323  
##           ACF1  
## Training set 0.005164711  
  
fore_arima = forecast::forecast(arima_model, h=12)  
df_arima = as.data.frame(fore_arima)  
dat_test$arima = df_arima$`Point Forecast`  
mape(dat_test$unemploy, dat_test$arima) ## 2.1%
```

```
## [1] 2.251609
myforecast <- forecast(arima_model, level=c(10), h=10*12)
plot(myforecast, col = "orange")
```

Forecasts from ARIMA(1,1,2)(0,0,2)[12]



```
##### TBATS #####
```

```
model_tbats <- tbats(dat_ts)
summary(model_tbats)

##          Length Class Mode
## lambda          1 -none- numeric
## alpha           1 -none- numeric
## beta            1 -none- numeric
## damping.parameter 1 -none- numeric
## gamma.values    0 -none- NULL
## ar.coefficients 0 -none- NULL
## ma.coefficients 0 -none- NULL
## likelihood      1 -none- numeric
## optim.return.code 1 -none- numeric
## variance        1 -none- numeric
## AIC             1 -none- numeric
## parameters      2 -none- list
## seed.states     2 -none- numeric
## fitted.values   552 ts    numeric
```

```

## errors      552 ts numeric
## x           1104 -none- numeric
## seasonal.periods 0 -none- NULL
## y           552 ts numeric
## call        2 -none- call
## series      1 -none- character
## method      1 -none- character

for_tbats <- forecast::forecast(model_tbats, h = 12)
df_tbats = as.data.frame(for_tbats)
dat_test$tbats = df_tbats$`Point Forecast`
mape(dat_test$unemploy, dat_test$tbats)

## [1] 2.369709

myforecast <- forecast(model_tbats, level=c(10), h=10*12)

plot(myforecast)

```

Forecasts from BATS(0.45, {0,0}, 0.891, -)

