

Стаціонарність часового ряду

КУРС АНАЛІЗ ЧАСОВИХ РЯДІВ

ДОКТОР ФІЛОСОФІЇ

ЛЮДМИЛА БОРИСІВНА ЛЕВЕНЧУК



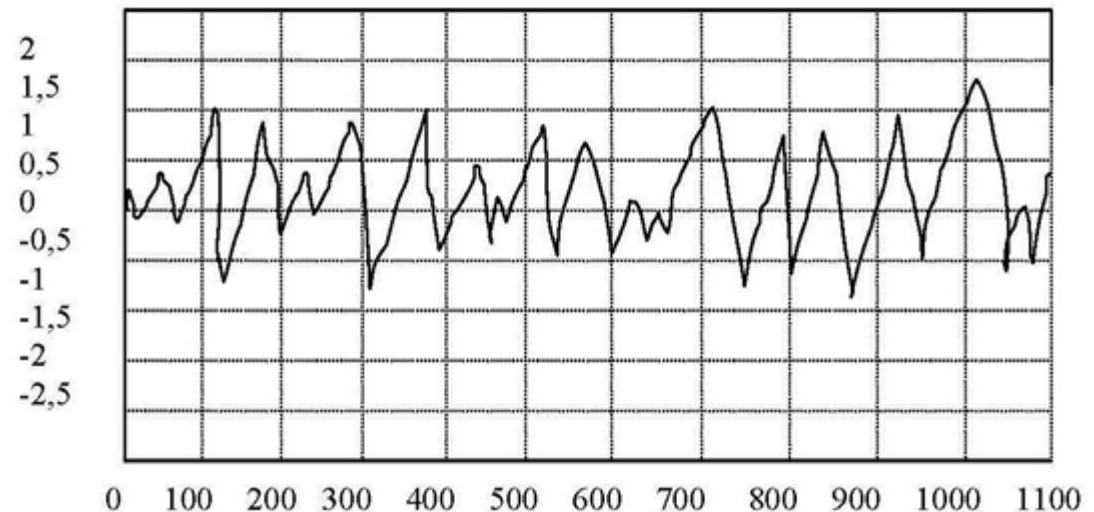
Стаціонарність часового ряду

Стаціонарний часовий ряд – це часовий ряд, у якого статистичні характеристики, такі як **середнє значення** і **дисперсія**, залишаються **постійними в часі**.

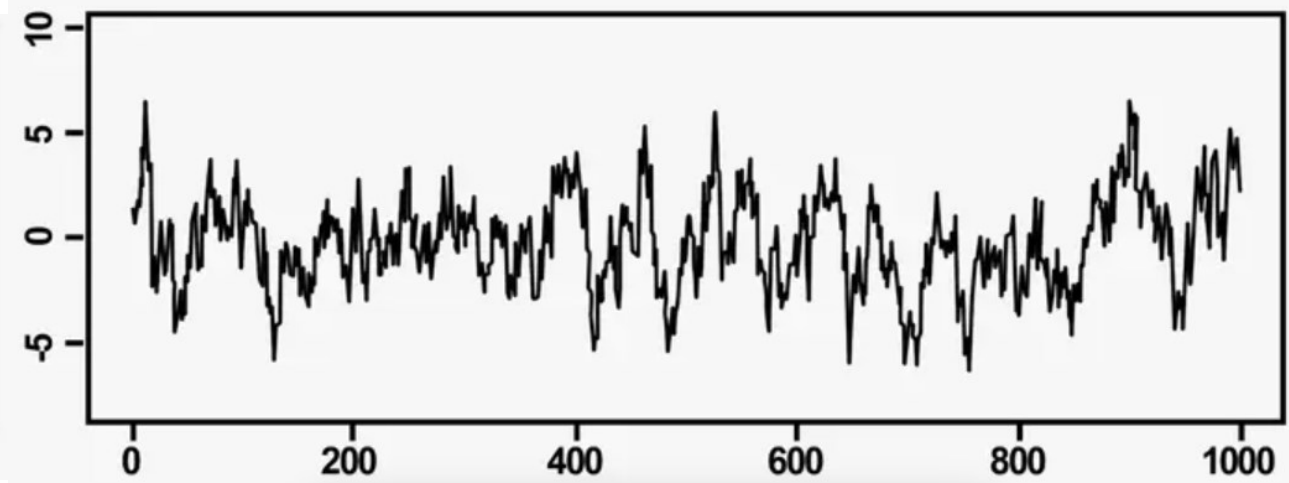
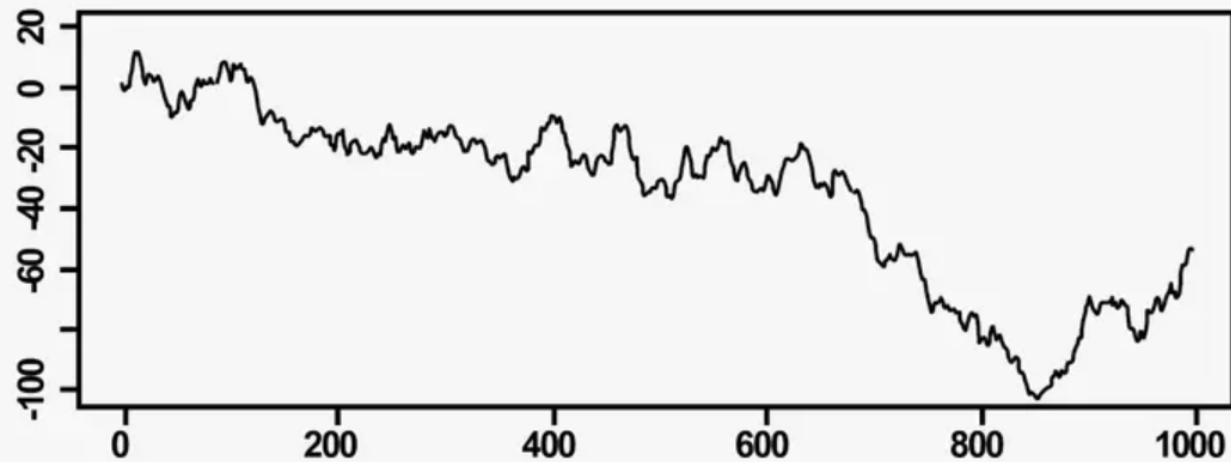
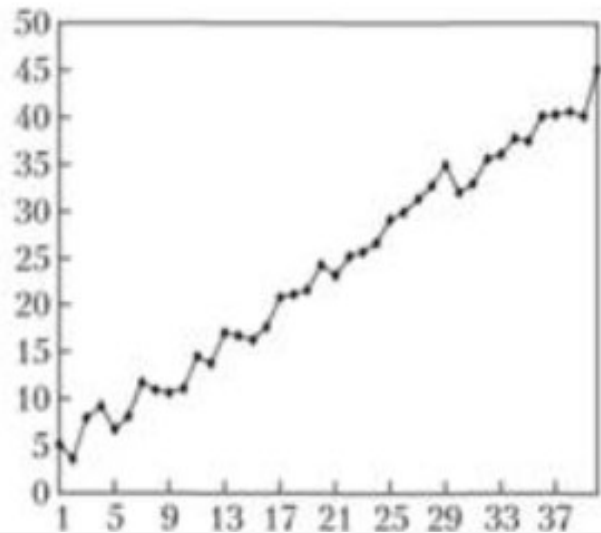
У стаціонарного часового ряду немає систематичних змін або сильно виражених трендів і його характеристики залишаються стабільними протягом всього спостереження.

Формально, для стаціонарного ряду повинні виконуватися наступні умови:

- постійне середнє значення;
- постійна дисперсія;
- сезонні ефекти відсутні.



Які ряди стаціонарні?



Чому важлива стаціонарність ряду

Стаціонарний часовий ряд **легше передбачити**, оскільки тренд є константою, на відміну від нестаціонарного часового ряду, чий тренд може змінити подія (*наприклад, акції Твіттера впали через висловлювання Ілона Маска*), передбачити яку дуже складно.

Стабільні статистичні характеристики, що значно спрощує аналіз, бо ми можемо не враховувати зміни в часі.

Ефективні моделі. Багато традиційних методів прогнозування, таких як моделі ARIMA (авторегресійно-інтегровані змінні середні) або згладжування експоненційним середнім, ефективніше працюють зі стаціонарними рядами.

Це не означає, що працюємо лише зі стаціонарним рядом. Існує великий клас моделей для моделювання та прогнозування саме нестаціонарних процесів.

Як перевірити ряд на нестационарність?

Графічний аналіз: Дивимося на графік. Шукаємо тренд, сезонність, гетероскедастичність.

Описові статистики: Розраховуємо середнє та дисперсію в різних частинах часовго ряду. Якщо вони суттєво відрізняються – можлива нестационарність ряду.

Тести: Використовуємо тести для перевірки на стаціонарність ряду.

Тест Дікі-Фуллера

<https://colab.research.google.com/drive/136ELxULnN0kuRO0y0uz9khNpOPdFGKQd?usp=sharing>



```
import pandas as pd

# Створення часового ряду (приклад)
data = {'Тиждень': range(1, 13),
        'Кількість продажів': [25, 28, 34, 31, 36, 29, 27, 32, 38, 41, 30, 35]}
df = pd.DataFrame(data)

# Розбивка часового ряду на три інтервали
interval_1 = df[df['Тиждень'].between(1, 4)]
interval_2 = df[df['Тиждень'].between(5, 8)]
interval_3 = df[df['Тиждень'].between(9, 12)]

# Обчислення статистичних показників для кожного інтервалу
seredne_1 = interval_1['Кількість продажів'].mean()
dispersiya_1 = interval_1['Кількість продажів'].var()

seredne_2 = interval_2['Кількість продажів'].mean()
dispersiya_2 = interval_2['Кількість продажів'].var()

seredne_3 = interval_3['Кількість продажів'].mean()
dispersiya_3 = interval_3['Кількість продажів'].var()

print("Інтервал 1:")
print(f"Середнє значення: {seredne_1}")
print(f"Дисперсія: {dispersiya_1}")
```



Інтервал 1:

Середнє значення: 29.5

Дисперсія: 15.0

Інтервал 2:

Середнє значення: 31.0

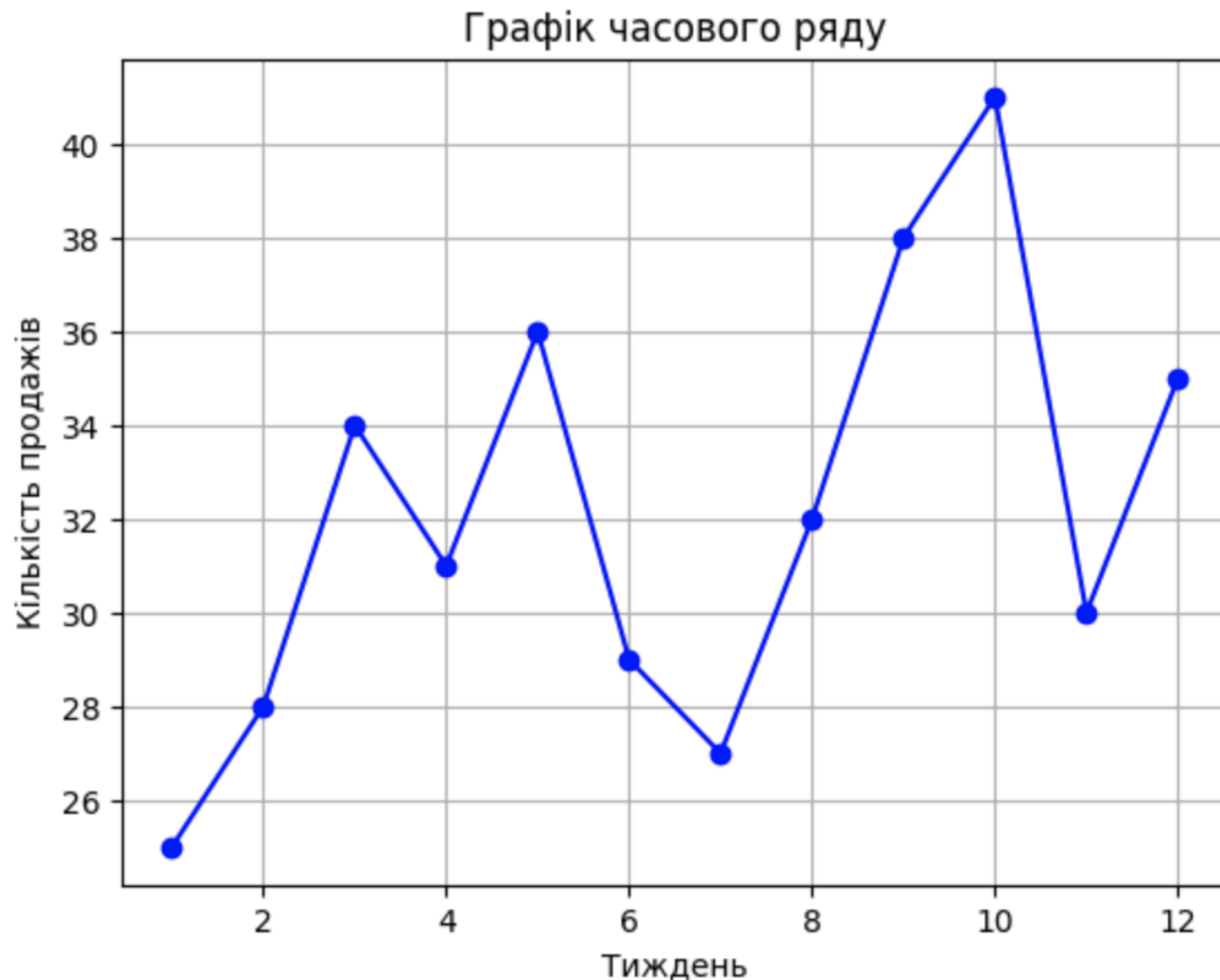
Дисперсія: 15.333333333333334

Інтервал 3:

Середнє значення: 36.0

Дисперсія: 22.0

https://colab.research.google.com/drive/1UqqEepiXZkJbOiwsM8lWIH_SGk6CS2ZD?usp=sharing



На основі наданих результатів можна сказати, що часовий ряд не є стаціонарним.

Існують кілька ознак нестационарності: Середнє значення змінюється між інтервалами: Середнє значення різних інтервалів відрізняється (**29.5, 31.0 і 36.0**), що свідчить про наявність тренду або інших складових змін.

Дисперсія не залишається сталою: Дисперсія також змінюється між інтервалами (**15.0, 15.33 і 22.0**), що свідчить про можливість тренду або гетероскедастичності.

https://colab.research.google.com/drive/1UqqEepiXZkJbOiws8IWIH_SGk6CS2ZD?usp=sharing

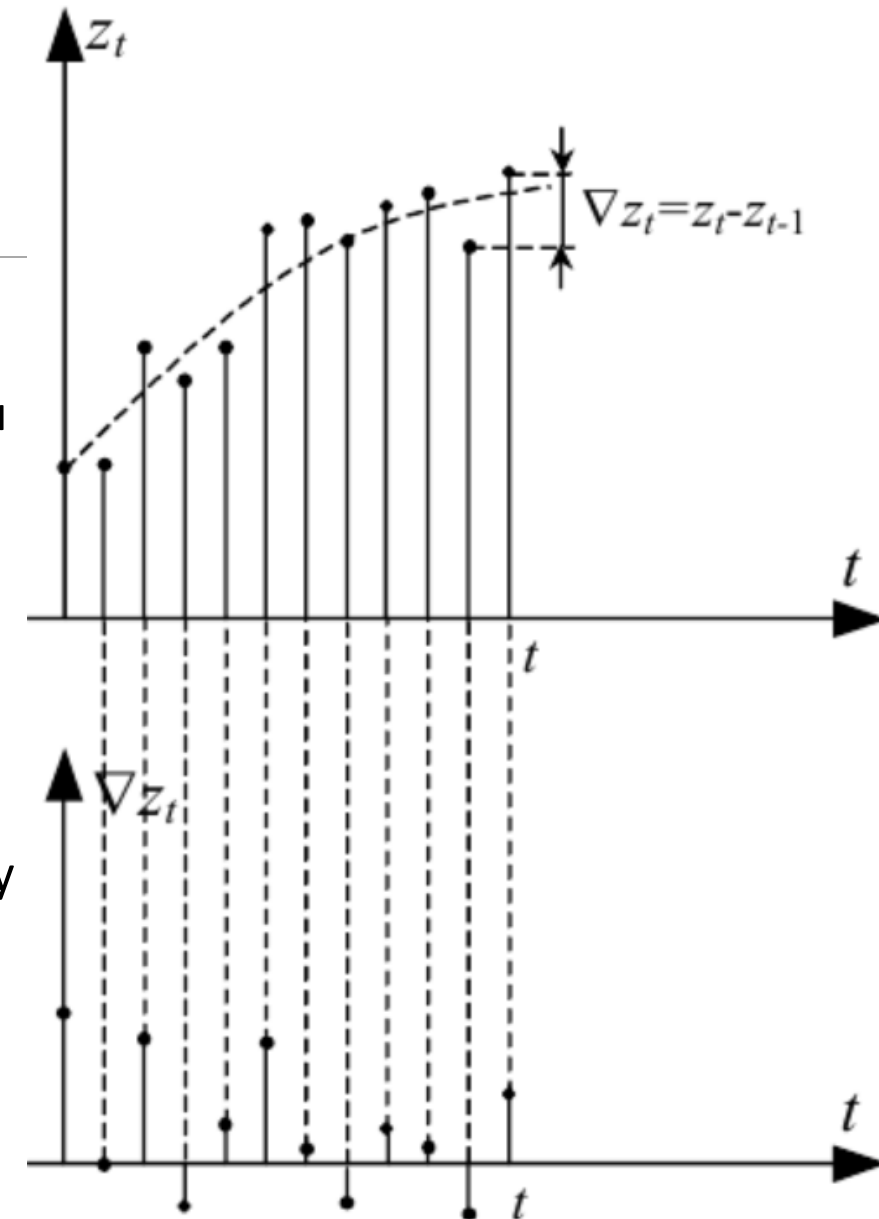
Приведення до стаціонарності

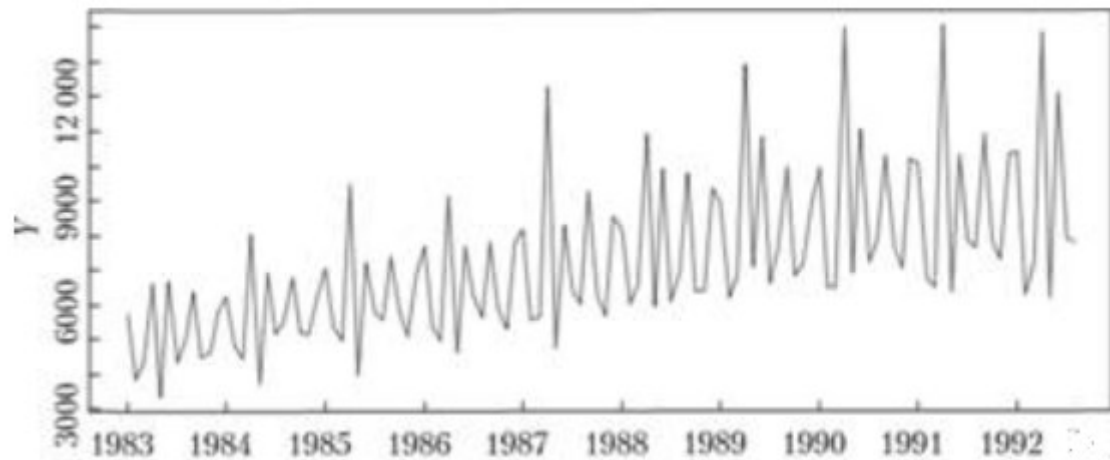
Видалення тренду за рахунок згладжування (Smoothing):

Використовуйте методи, такі як згладжування експоненційним середнім або ковзне середнє, щоб видалити тренд та зберегти загальну структуру ряду.

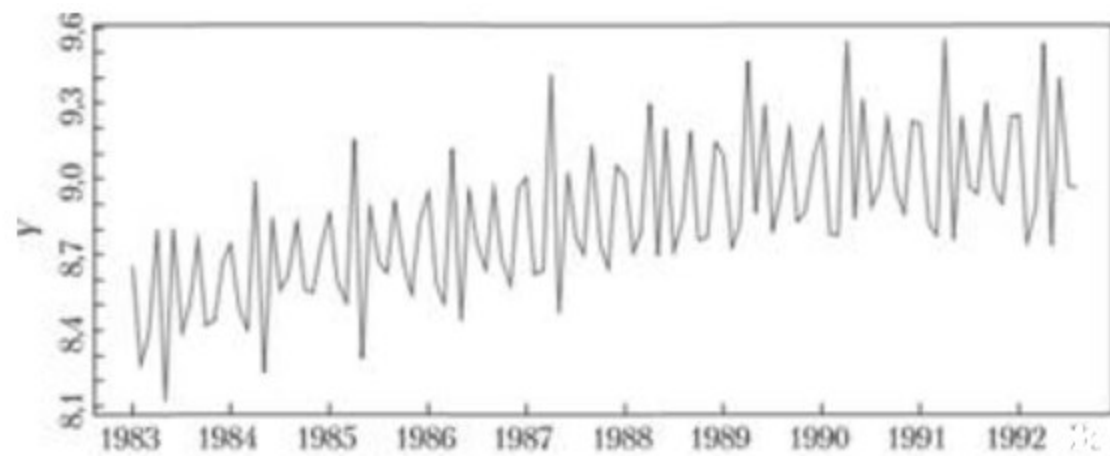
Нормалізація: Нормалізація початкового ряду може зменшити вплив зростання дисперсії з часом, зробити дисперсію більш константою.

Видалення тренду та сезонності: Сезонне диференціювання: Якщо у ряду присутня сезонність, використання сезонного диференціювання (різницю між значеннями на одному і тому ж сезоні в різні роки) може допомогти зменшити цю компоненту.

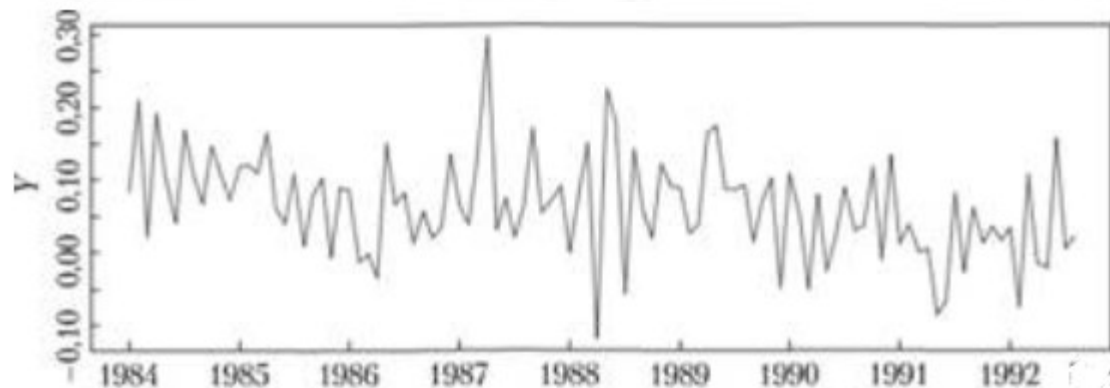




Наявність тренду і змінних коливань
(гетероскедастичності)

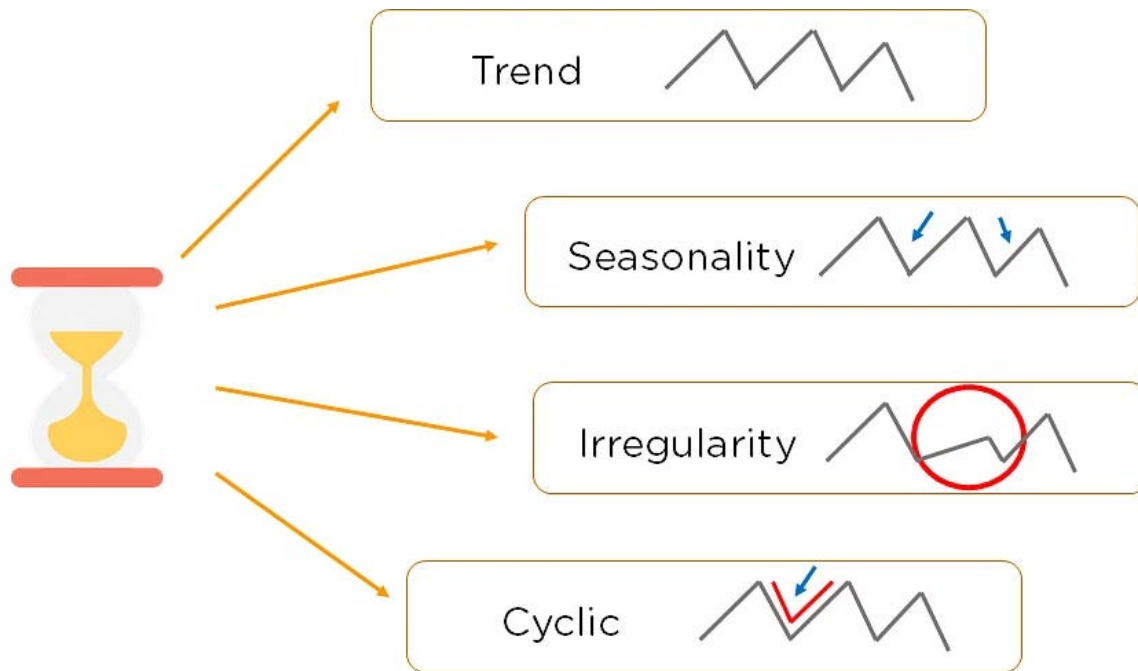


Прологарифмували – стабілізували
дисперсію



Для позбавлення сезонності застосуємо
сезонні різниці

Складові часового ряду

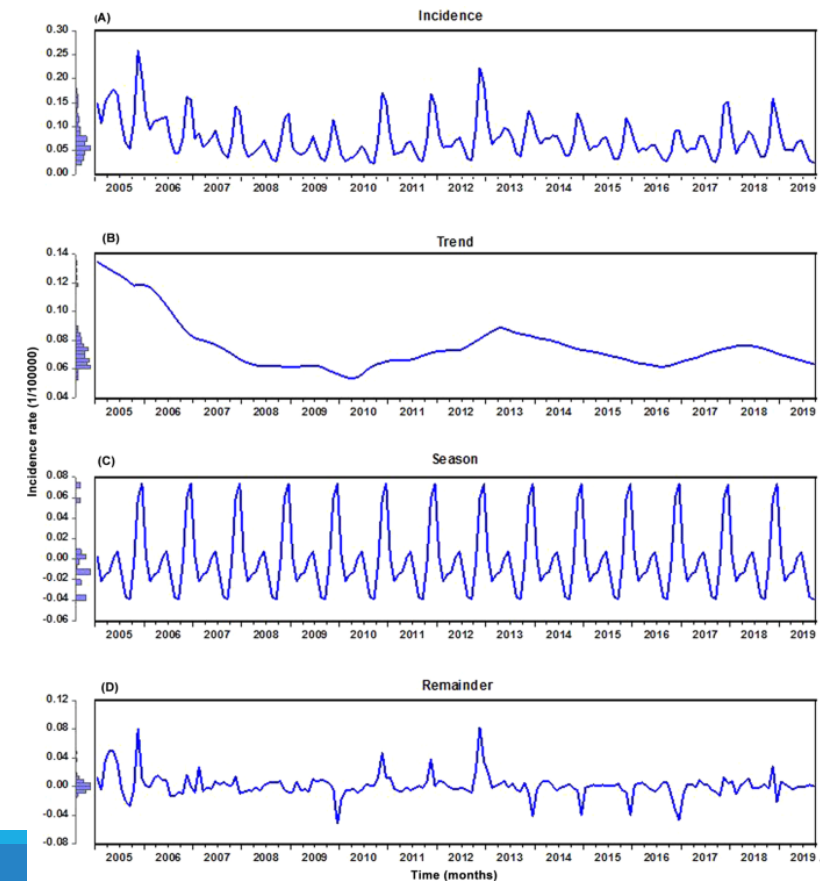
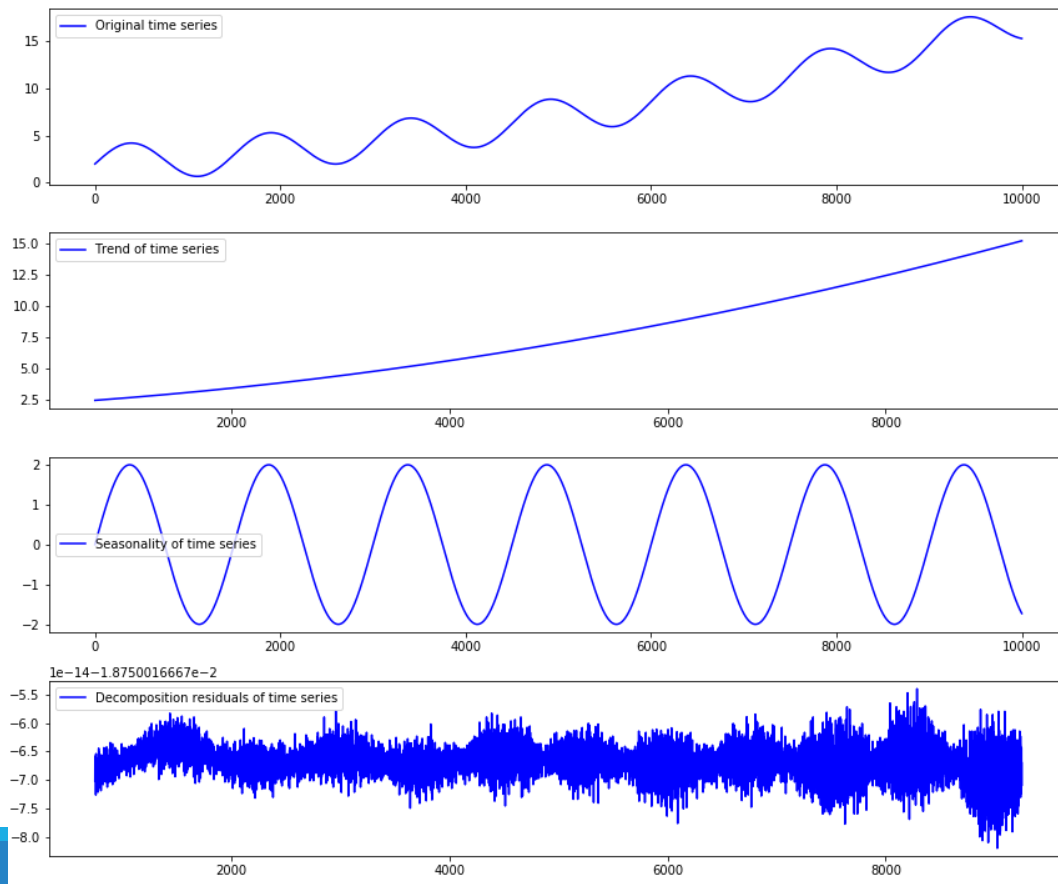
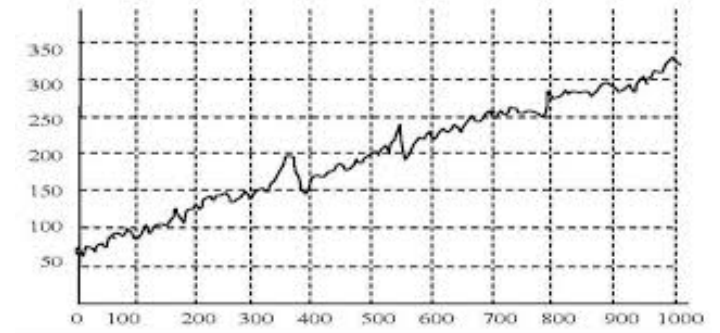


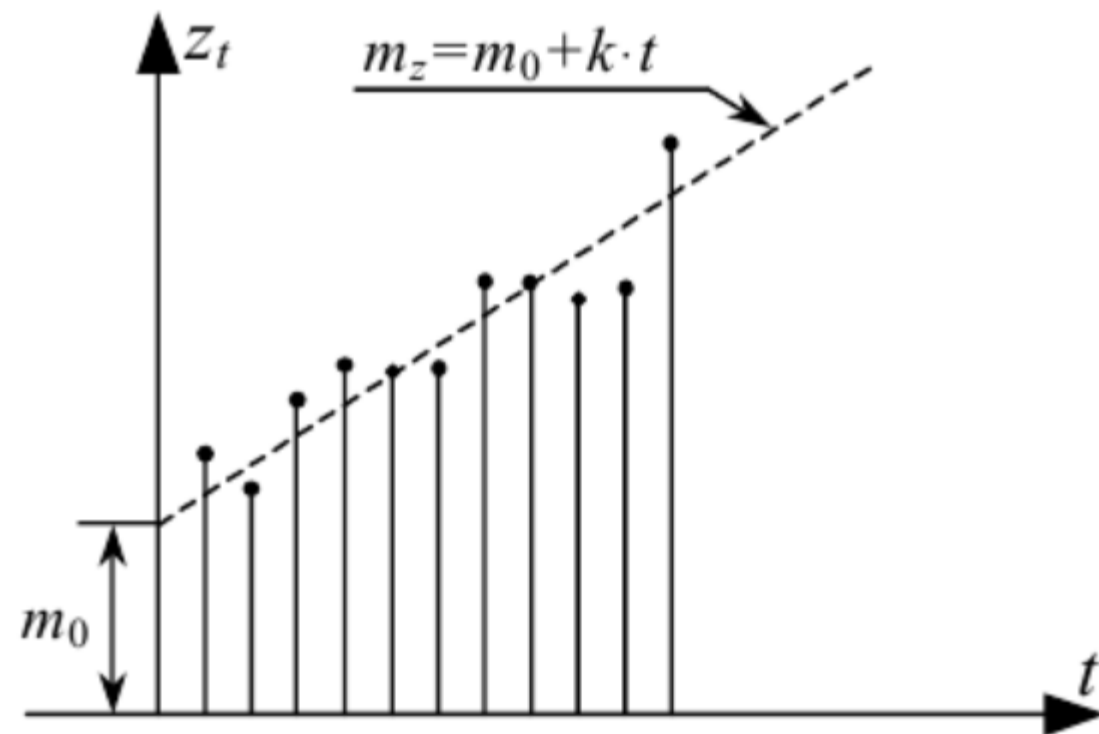
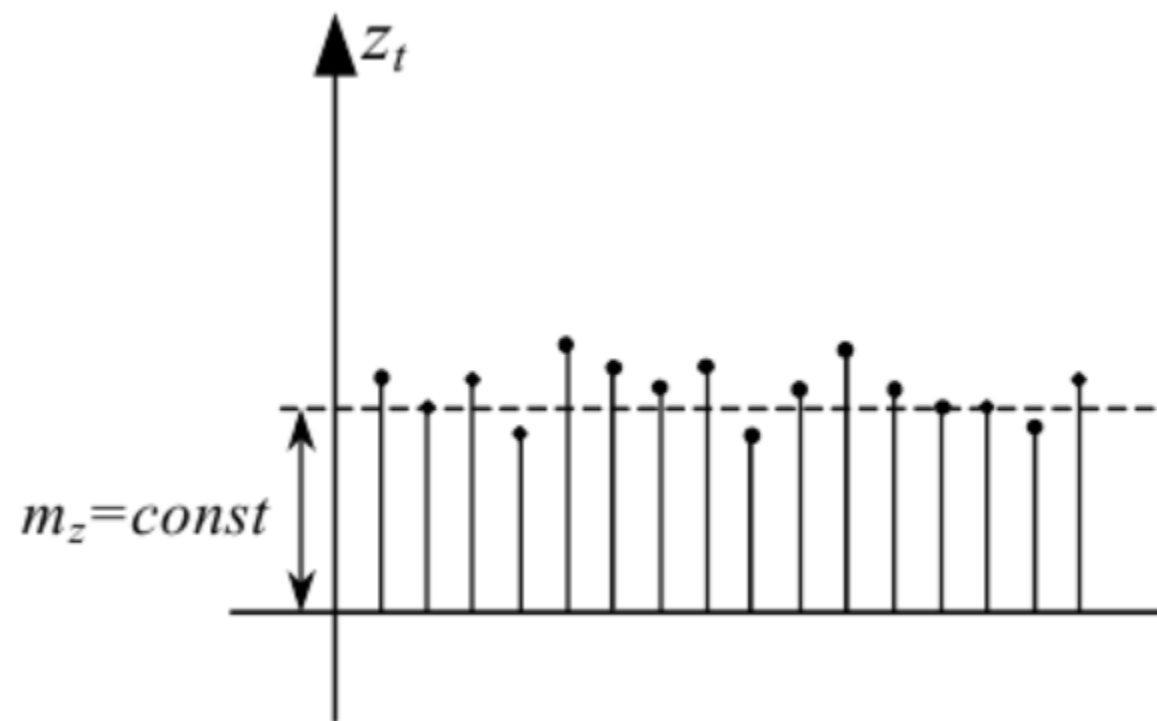
Якщо часовий ряд **не є стаціонарним**, то перше завдання полягає в визначенні того, чи існує в ряду **тренд**.

Ідентифікація тренду може бути складною задачею, і вона зазвичай виконується після визначення стаціонарності, щоб забезпечити правильний контекст для аналізу тренду.

Тренд

Тренд - загальна тенденція в зміні показників часового ряду, яка визначається загальним напрямком змін. Тип графіку встановлюється на основі даних часового ряду.





Типи трендів часових рядів

Upward trend

Upward trend: Bull Market— це період часу на фінансових ринках, коли ціна активу або цінного паперу постійно зростає. Загальноприйнятим визначенням бичачого ринку є те, що ціни на акції зростають на 20%.

Bull Market загалом розглядаються як ознака економічної міцності і можуть бути викликані факторами, такими як підвищений попит, зростання прибутків і сприятливі економічні умови.

Протилежністю бичачого ринку є bear market, коли ціни мають тенденцію до зниження.



Типи трендів часових рядів

Downward trend

Downward trend: Downward trend, також відомий як bear market, - це загальне падіння фондового ринку протягом певного періоду часу. Він включає перехід від високого оптимізму інвесторів до широко поширеного страху і песимізму інвесторів. Одним із загальноприйнятих показників ведмежого ринку є зниження ціни на 20 % або більше протягом принаймні двох місяців.

Downward trend загалом розглядаються як ознака економічної слабкості і можуть бути викликані факторами, такими як слабкий попит, зменшення прибутків і не вигідні економічні умови.



Типи трендів часових рядів

Sideways trend

Sideways trend: Sideways trend, також відомий як ринок у межах діапазону, - це період відносно стабільних цін на певний цінний папір або ринок.

Горизонтальні тренди можуть характеризуватися відсутністю чіткого напрямку, з цінами, які коливаються в межах відносно вузького діапазону.



Переваги та недоліки аналізу трендів



- + Допомагає визначити можливості для купівлі або продажу цінних паперів.
- + Може виявити потенційні ризики або попереджати про ознаки того, що цінні папери або ринок можуть рухатися вниз.
- + Надає уявлення про психологію ринку та його динаміку.



- Якщо ринки ефективні, аналіз трендів може бути менш корисним.
- Якщо дані неповні, неточні або інакше помилкові, аналіз також може вводити в оману або бути неточним.
- Може не враховувати зміни в управлінні компанії, зміни в галузевих правилах або інші зовнішні фактори, які можуть вплинути на продуктивність цінних паперів.
- Різні статистичні показники можуть давати різні результати.

Визначення тренду та робота з даними

Кроки аналізу тренду:

- ✓ Збір даних
- ✓ Візуалізація даних
- ✓ Згладжування даних
- ✓ Статистичний аналіз
- ✓ Тестування стаціонарності
- ✓ Декомпозиція часового ряду
- ✓ Прогнозування
- ✓ Моніторинг та аналіз помилок



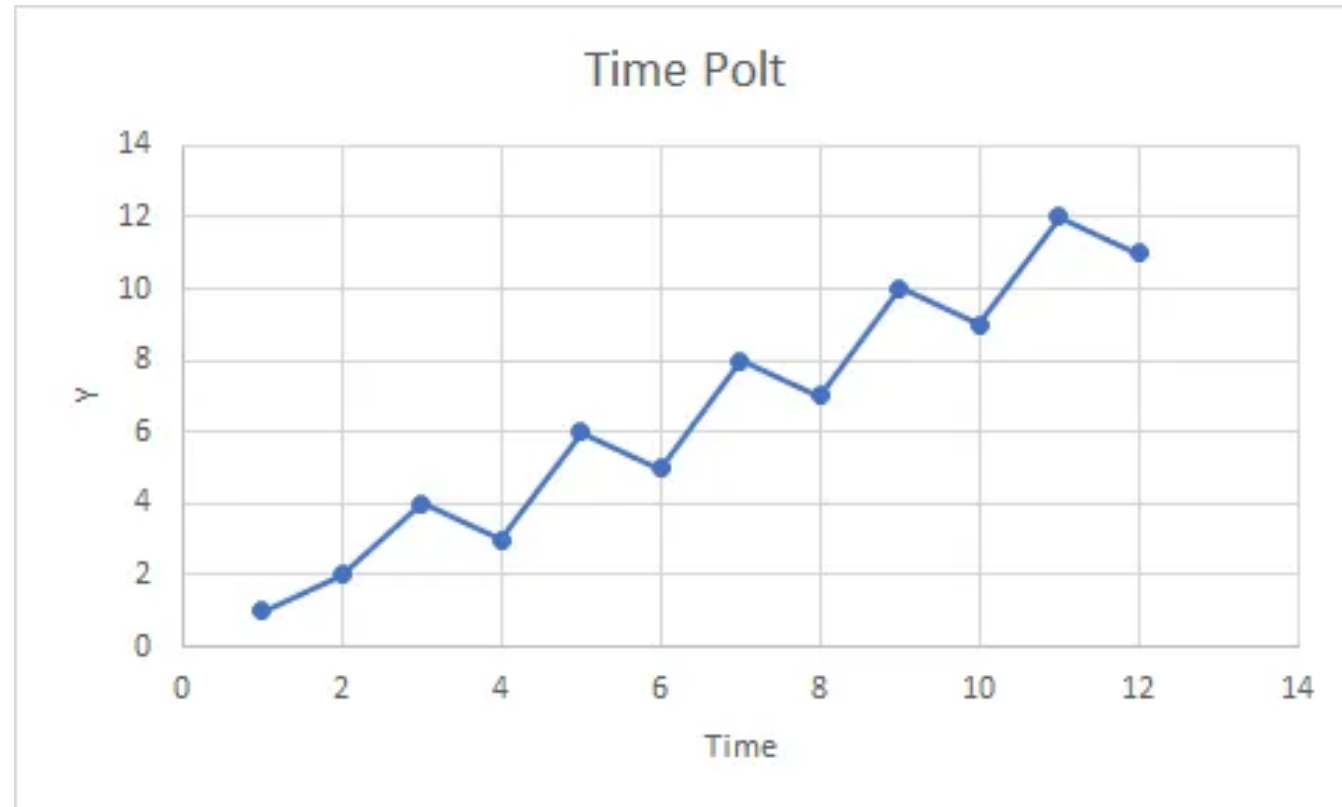
Як визначити тренд і що з ним робити

Time	Y
1	1
2	2
3	4
4	3
5	6
6	5
7	8
8	7
9	10
10	9
11	12
12	11

Постановка задачі:

Є одновимірний масив даних, представлений у вигляді часового ряду, для якого необхідно:

- Знайти тренд
- Визначити кроки роботи з цим трендом (видалення, опрацювання, аналіз)



Лінійна регресія

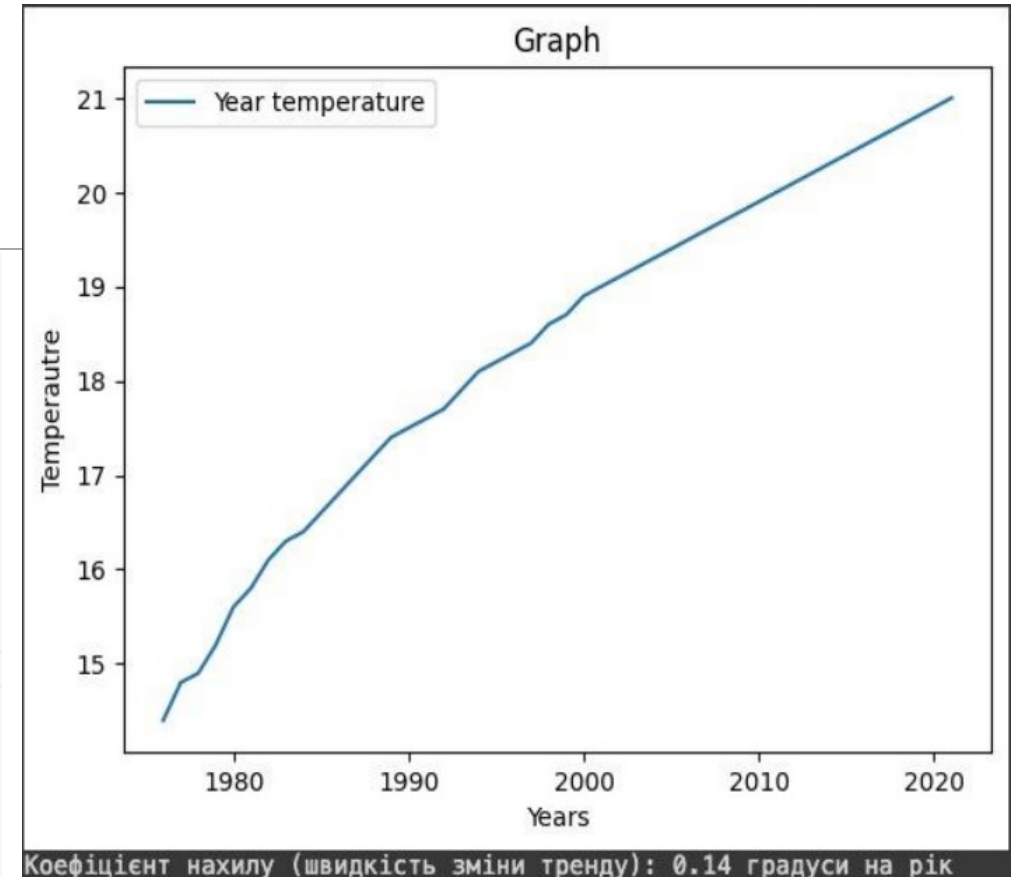
```
import numpy as np
import matplotlib.pyplot as plt

# Задайте дані в часовому ряді
years = np.array([i for i in range(1976, 2022)])
temperature = np.array([14.4, 14.8, 14.9, 15.2, 15.6, 15.8, 16.1, 16.3, 16.4,

# Використовуємо лінійну регресію
coefficients = np.polyfit(years, temperature, 1)
slope, intercept = coefficients

x = years
y1 = temperature
plt.plot(x, y1, label="Year temperature")
plt.plot()

plt.xlabel("Years")
plt.ylabel("Temperautre")
plt.title("Graph")
plt.legend()
plt.show()
```



https://colab.research.google.com/drive/1tMVLoVUT-PO9_UxHZNs_UaDUXB9w8yKt?usp=sharing

ADF-тест (Augmented Dickey-Fuller test)

H0: ряд нестационарний

H1: ряд стаціонарний

*Якщо p-значення менше або дорівнює **0.05**, то ми відхиляємо нульову гіпотезу про наявність тренда, інакше ми припускаємо наявність тренда в часовому ряді.*

➞ ADF Statistic: -1.1320384625097901
p-value: 0.7021277385898382

Critical Values: {'1%': -3.498198082189098, '5%': -2.891208211860468, '10%': -2.5825959973472097}

Lags Used: 0

Number of Observations Used: 99

Ряд не є стаціонарним. Є тренд.

```
import numpy as np
import pandas as pd
from statsmodels.tsa.stattools import adfuller

# Згенуємо приклад часового ряду з трендом
np.random.seed(0)
time_series = np.cumsum(np.random.randn(100)) # Сума нормально розподілених випадкових значень

# Проводимо ADF-тест
result = adfuller(time_series)

# Виведемо результати тесту
print('ADF Statistic:', result[0])
print('p-value:', result[1])
print('Critical Values:', result[4])
print('Lags Used:', result[2])
print('Number of Observations Used:', result[3])

# Інтерпретація результату
if result[1] <= 0.05:
    print("Ряд є стаціонарним. Немає тренда.")
else:
    print("Ряд не є стаціонарним. Є тренд.")
```

https://colab.research.google.com/drive/1Ycnf37KqkVEtBoTlggkrJet_W8TRQ1K0?usp=sharing

KPSS (Kwiatkowski-Phillips-Schmidt-Shin) test

H0: ряд стаціонарний

H1: ряд нестационарний

Якщо результат KPSS-тесту вказує на відхилення від нульової гіпотези, це свідчить про наявність тренда у часовому ряді. З іншого боку, якщо результат не відхиляє нульову гіпотезу, це може вказувати на відсутність тренда в часовому ряді, що робить його стаціонарним.

```
[2] import numpy as np
from statsmodels.tsa.stattools import kpss

# Згенеруємо приклад часового ряду з трендом
np.random.seed(0)
time_series = np.cumsum(np.random.randn(100)) # Сума нормально розподілених випадкових значень

# Проводимо KPSS-тест
result = kpss(time_series)

# Виведемо результати тесту
print('KPSS Statistic:', result[0])
print('p-value:', result[1])
print('Lags Used:', result[2])
print('Critical Values:', result[3])

# Інтерпретація результату
if result[1] <= 0.05:
    print("Ряд не є стаціонарним. Є тренд.")
else:
    print("Ряд є стаціонарним. Немає тренда.")
```

➞ KPSS Statistic: 1.1001161286471663

p-value: 0.01

Lags Used: 5

Critical Values: {'10%': 0.347, '5%': 0.463, '2.5%': 0.574, '1%': 0.739}

Ряд не є стаціонарним. Є тренд.

<ipython-input-2-cd9a3ca1d68e>:9: InterpolationWarning: The test statistic is outside of the range of p-values available in the look-up table. The actual p-value is smaller than the p-value returned.

```
result = kpss(time_series)
```

https://colab.research.google.com/drive/1Ycnf37KqkVEtBoTlggkrJet_W8TRQ1K0?usp=sharing

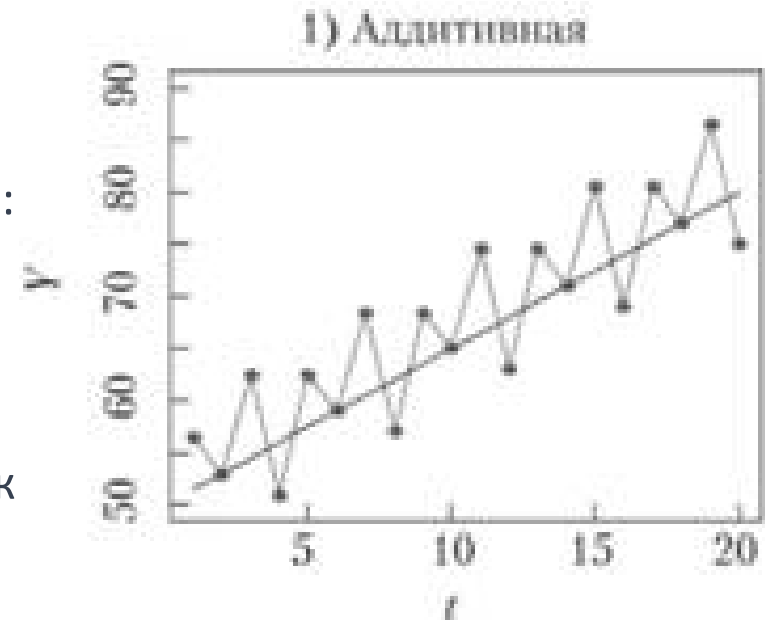
Адитивна декомпозиція ЧР

У цьому методі компоненти часового ряду розглядаються як "додавання", що означає, що вони додаються разом для формування часового ряду.

Формула адитивної декомпозиції виглядає наступним чином:
 $Y(t) = \text{Тренд}(t) + \text{Сезонність}(t) + \text{Залишкові члени}(t)$

$$y_t = T_t + C_t + S_t + I_t$$

У цьому випадку зміни в тренді та сезонності розглядаються як постійні абсолютні значення, які додаються до початкових даних.



Використовуємо: Якщо сезонні коливання в часовому ряді досить слабкі та не змінюються пропорційно до рівня даних, адитивна декомпозиція може бути ефективною.

Мультиплікативна декомпозиція ЧР

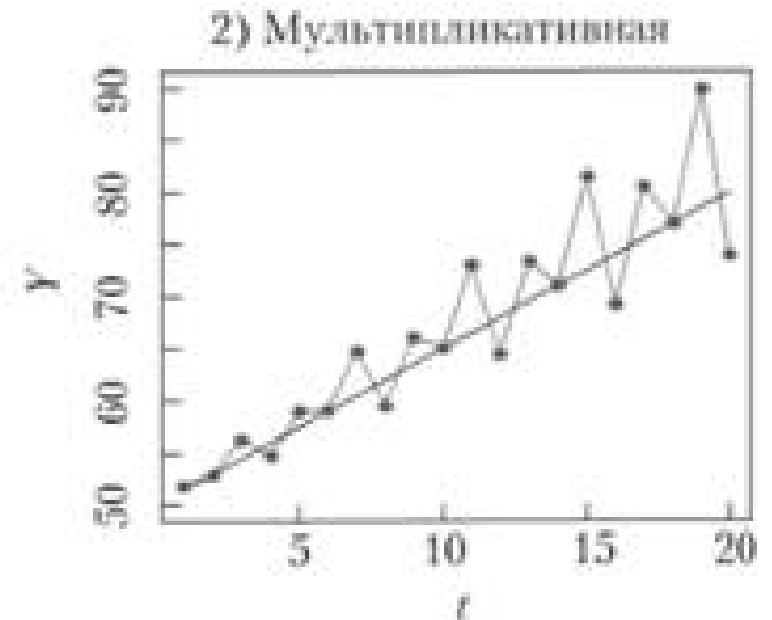
У методі мультиплікативної декомпозиції компоненти розглядаються як "множення", що означає, що вони множаться разом для формування часового ряду.

Формула мультиплікативної декомпозиції виглядає наступним чином: $Y(t) = \text{Тренд}(t) * \text{Сезонність}(t) * \text{Залишкові члени}(t)$

$$y_t = T_t \times C_t \times S_t \times I_t$$

У цьому випадку зміни в тренді та сезонності розглядаються як відсоткові зміни, які множаться на початкові дані.

Використовуємо: Якщо сезонні коливання в часовому ряді є значними та змінюються пропорційно до рівня даних, мультиплікативна декомпозиція може бути більш інформативною.



Приклад – адитивна декомпозиція

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose

# Згенуємо уявні дані про часовий ряд
np.random.seed(0)
date_rng = pd.date_range(start='2022-01-01', end='2023-12-31', freq='D')
data = np.random.randn(len(date_rng)) # Уявні дані

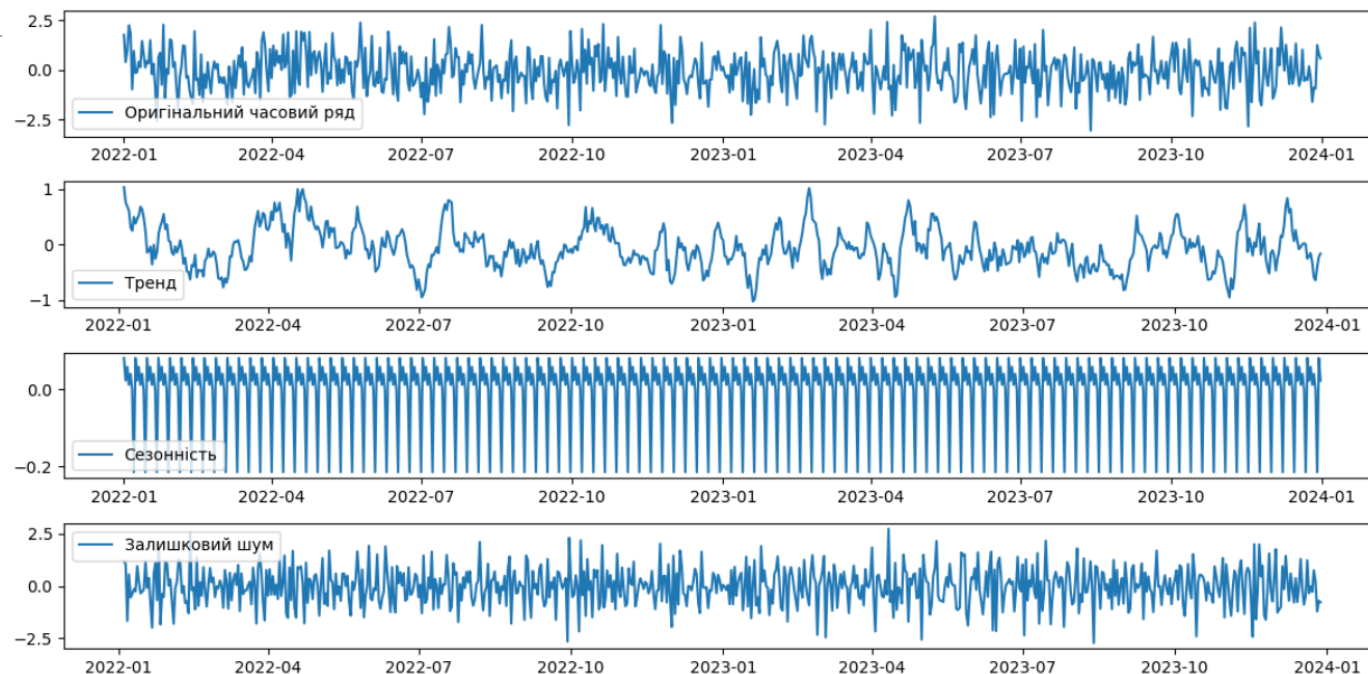
# Створення DataFrame з датою та даними
df = pd.DataFrame(data, index=date_rng, columns=['Значення'])

# Виконання адитивної декомпозиції
result = seasonal_decompose(df, model='additive')

# Візуалізація результатів декомпозиції
plt.figure(figsize=(12, 6))

# Оригінальний часовий ряд
plt.subplot(411)
plt.plot(df['Значення'], label='Оригінальний часовий ряд')
plt.legend()

# Тренд
plt.subplot(412)
plt.plot(result.trend, label='Тренд')
plt.legend()
```



<https://colab.research.google.com/drive/1kZ43JC1UZt5KzybpF--9rRKuWYDCE86?usp=sharing>

Приклад – мультиплікативна декомпозиція

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose

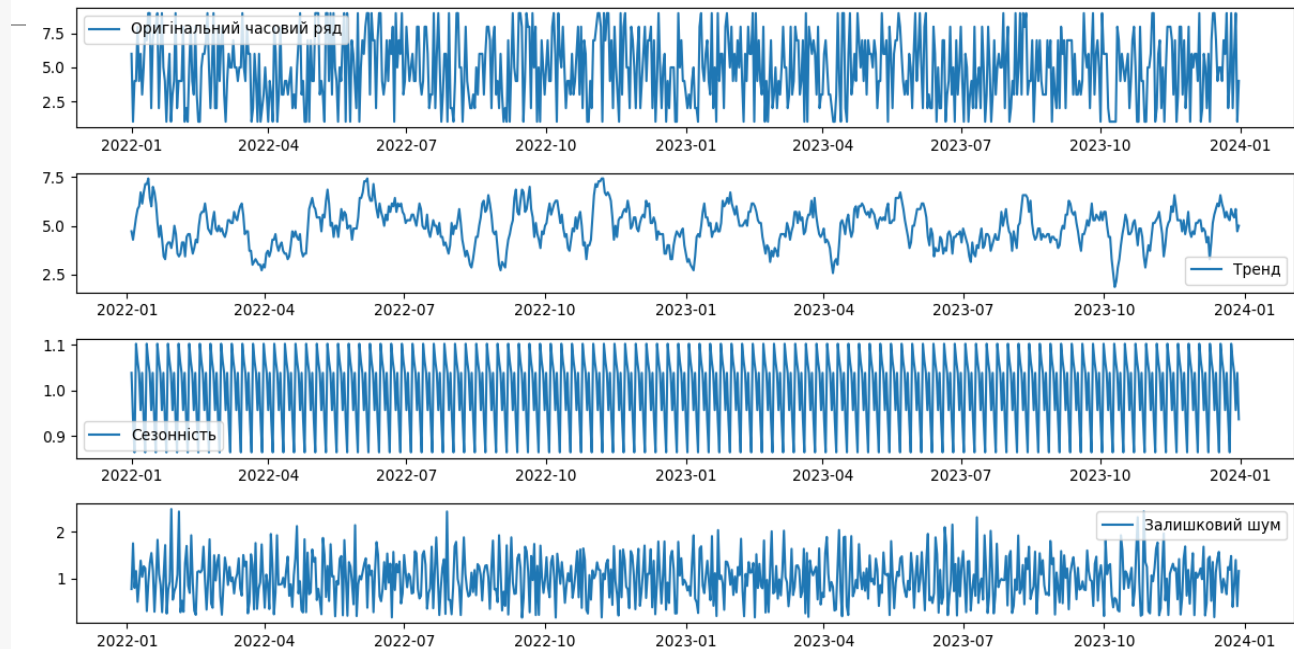
# Згенуємо уявні дані про часовий ряд
np.random.seed(0)
date_rng = pd.date_range(start='2022-01-01', end='2023-12-31', freq='D')
data = np.random.randint(1,10,(date_rng.size,1))

# Створення DataFrame з датою та даними
df = pd.DataFrame(data, index=date_rng, columns=['Значення'])

# Виконання мультиплікативної декомпозиції
result = seasonal_decompose(df, model='multiplicative')

# Візуалізація результатів декомпозиції
plt.figure(figsize=(12, 6))

# Оригінальний часовий ряд
plt.subplot(411)
plt.plot(df['Значення'], label='Оригінальний часовий ряд')
plt.legend()
```



https://colab.research.google.com/drive/1qGS1_C7_xkwRmEYH6KHxaWZIFbGVTHCb?usp=sharing

Диференціація. Після виконання диференціації, data міститиме дані без тренду.

```
import pandas as pd
import matplotlib.pyplot as plt

# Завантаження часового ряду
data = pd.read_csv('/content/sample_data/oil.csv')

# Видалення тренду за допомогою середньої скользящей
window = 7
data['MovAver'] = data['dcoilwtico'] - data['dcoilwtico'].rolling(window=window).mean()
plt.plot(data['MovAver'])
```

```
import pandas as pd
from scipy.signal import savgol_filter

# Загрузка часового ряда
data = pd.read_csv('/content/sample_data/oil.csv')

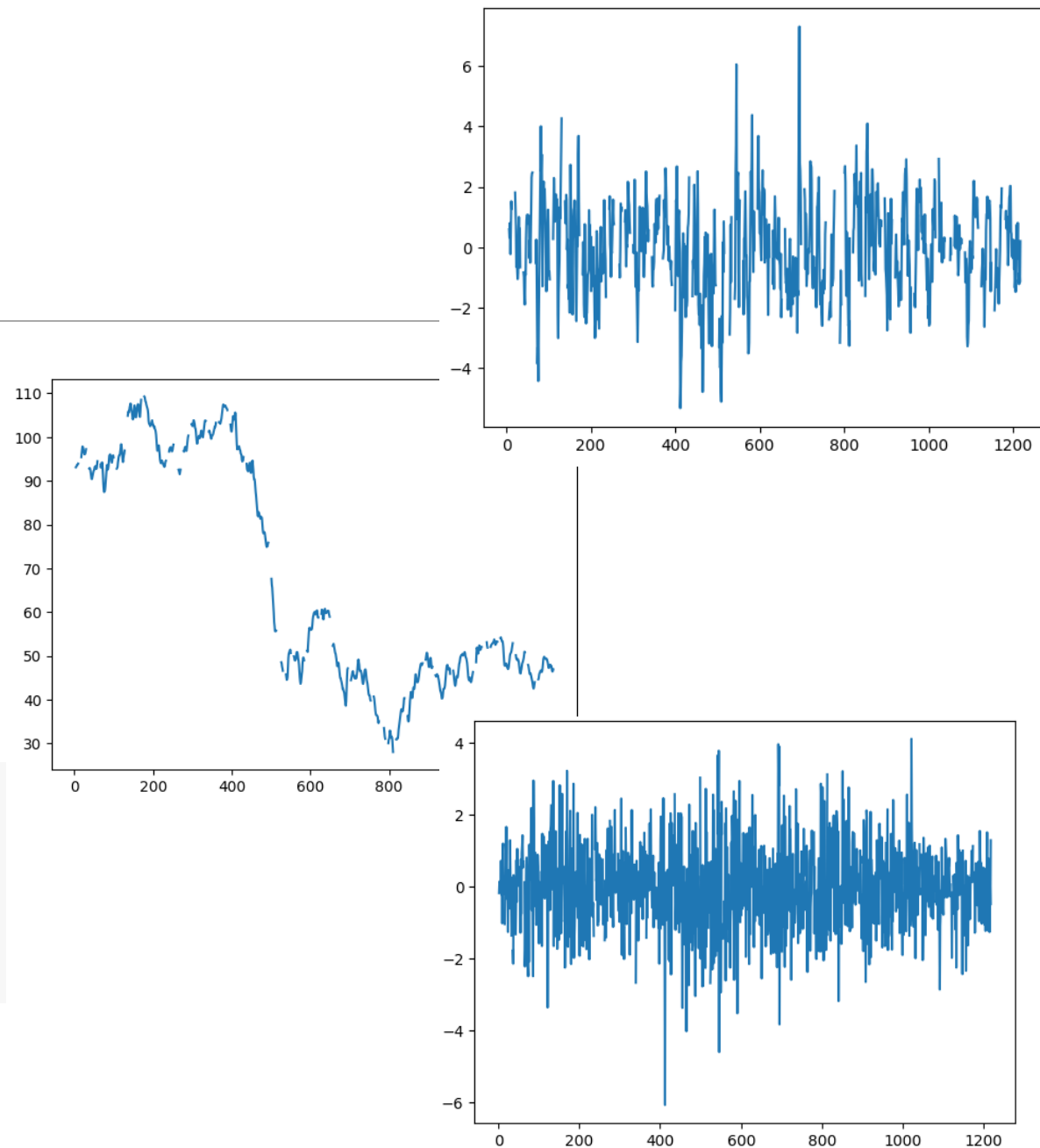
# Застосування фільтра Савіцького-Голея для видалення тренду
data['Filtered'] = savgol_filter(data['dcoilwtico'], window_length=7, polyorder=2)
plt.plot(data['Filtered'])
```

```
import pandas as pd

# Загрузка часового ряда
data = pd.read_csv('/content/sample_data/oil.csv')

# Виконання першого рівня диференціації
data['Differenced'] = data['dcoilwtico'] - data['dcoilwtico'].shift(1)
plt.plot(data['Differenced'])
```

https://colab.research.google.com/drive/1Zz-5xdt6jYGa3yuO_jWHp3kSXfU4RKwb?usp=sharing



Чи потрібно видаляти тренд?

Аналіз тренду: Іноді вас цікавить саме аналіз тренду. Наприклад, ви можете бажати вивчити зростання або спад показників в часовому ряді та визначити, чи існують загальні тенденції. У цьому випадку ви не видаляєте тренд, а намагаєтеся краще зрозуміти його характер.

Прогнозування без тренду: Якщо вам потрібно прогнозувати майбутні значення часового ряду, і ви знаєте, що тренд є нестаціонарним та змінюється з часом, видалення тренду може допомогти стабілізувати дисперсію та зробити ряд стаціонарним, що полегшує прогнозування.

Аналіз сезонності та шуму: У деяких випадках вас може цікавити аналіз сезонних коливань та шуму у часовому ряді, і тренд може заважати вам виділити ці компоненти. В цьому випадку видалення тренду може полегшити аналіз сезонності та шуму.

Дослідження і виокремлення складових: У деяких випадках ви можете видаляти тренд для подальшого дослідження сезонності або інших складових часового ряду. Видалення тренду може допомогти зосередити увагу на інших важливих аспектах даних.

Збереження тренду для окремого аналізу: Іноді ви можете зберегти тренд для подальшого аналізу, наприклад, як окремої змінної, і досліджувати його окремо від інших компонентів часового ряду.

Стратегії прогнозування з трендом

1. Прогнозування з використанням тренду: Якщо тренд сталої або прогресуючої форми, то його можна використовувати у моделі для прогнозування майбутніх значень. **Наприклад,** для лінійного тренду можна використовувати лінійну регресію для прогнозування.

Лінійна регресія: Якщо тренд є лінійним, ви можете використовувати просту лінійну регресію для прогнозування майбутніх значень тренду.

Авторегресійні моделі (AR): Якщо є автокореляція у тренді, можна використовувати авторегресійні моделі для прогнозування.

https://www.statsmodels.org/stable/generated/statsmodels.tsa.ar_model.AutoReg.html

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

https://colab.research.google.com/drive/1G0SqiW1822Q-20Q8NSjD6PrCLrlxewe_?usp=sharing

Стратегії прогнозування з трендом

2. Врахування тренду в моделі: В більш складних ситуаціях, де тренд може бути нелінійним або взаємодіяти з іншими факторами, можна використовувати більш складні моделі для прогнозування, такі як авторегресійні моделі (ARIMA), моделі експоненційного зростання (Holt-Winters) тощо. У таких моделях тренд інтегрується в саму модель.

Модель Хольта-Вінтерса (Holt-Winters): Ця модель включає тренд, сезонність і залишкову компоненту. Вона добре підходить для часових рядів із сезонністю.

ARIMA (Autoregressive Integrated Moving Average): може враховувати як тренд, так і сезонність, а також знаходити оптимальні параметри моделі.

<https://www.statsmodels.org/stable/generated/statsmodels.tsa.arima.model.ARIMA.html>

<https://www.statsmodels.org/dev/generated/statsmodels.tsa.holtwinters.ExponentialSmoothing.html>

https://colab.research.google.com/drive/1uRddaaSUSy0bx-FepR_tn2drRe2TvM-1?usp=sharing

Стратегії прогнозування з трендом

3. Прогнозування з урахуванням тренду та моделі ковзного середнього

Експоненційне згладжування (Exponential Smoothing): Ця модель враховує тренд і може включати сезонність.

Метод ковзного середнього: Простий ковзний середній або зважений ковзний середній можуть бути використані для прогнозування в залежності від специфіки даних.

https://colab.research.google.com/drive/17AoUMGKRv_4QgiS8Q_FyucTZNmlukbxC?usp=sharing

Підсумки



1. Тренд в часовому ряді представляє собою загальну напрямну тенденцію розвитку явища в часі.
2. Тренд може бути зростаючим, спадаючим або лінійним, і він може мати різні форми.
3. Аналіз тренду є важливими етапами у розумінні часових рядів та прогнозуванні.
4. Тренди важливі для виявлення довгострокових тенденцій та змін у часовому ряді.
5. Розуміння трендів допомагає приймати управлінські рішення, визначати планування та прогнозування.
6. Реальні приклади з різних сфер життя, такі як фінанси, економіка, погода, демографія, маркетинг і інші, можуть виглядати по-різному.

Розуміння трендів - це важлива навичка в області аналізу даних та прийняття рішень, і вона може бути корисною в різних сферах життя та бізнесу.

Додатково:

Книги:

1. "Time Series Analysis and Its Applications: With R Examples" by Robert H. Shumway and David S. Stoffer
2. "Forecasting: Principles and Practice" by Rob J. Hyndman and George Athanasopoulos
3. "Introduction to Time Series Analysis and Forecasting" by Douglas C. Montgomery, Cheryl L. Jennings, and Murat Kulahci

Статті та матеріали:

1. "A Gentle Introduction to Time Series Analysis" by Frederick Hartman
2. "Time Series Analysis" від Вікіпедії

Онлайн-курси:

1. Coursera: Курс "Practical Time Series Analysis" від State University of New York
2. edX: Курс "Time Series Analysis for Business Forecasting" від UC Berkeley School of Information
3. DataCamp: DataCamp пропонує кілька курсів про роботу з часовими рядами, включаючи "Introduction to Time Series Analysis in Python" та "Time Series Analysis with Python".

