



# Time Series Analysis

*Kalbe Digital University*

# Outline

**1** What is Time Series Data

**2** Framework and Toolkit

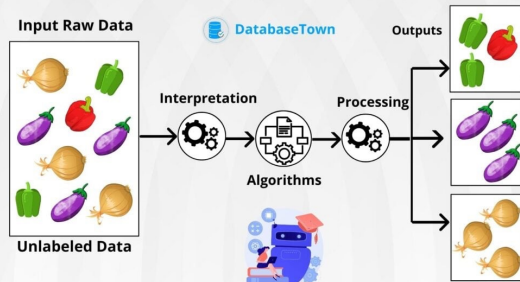
**3** Visualization

**4** Linear Regression

# Kinds of Data We learn

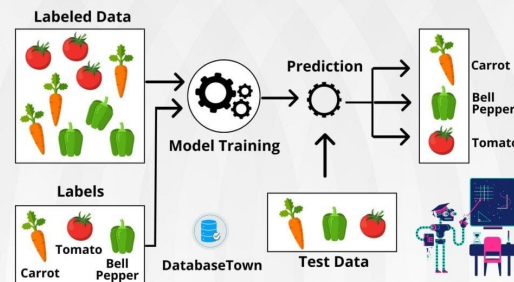
## UNSUPERVISED LEARNING

Unsupervised learning is a type of machine learning where the algorithm learns from unlabeled data without any predefined outputs or target variables.



## SUPERVISED LEARNING

Supervised machine learning is a branch of artificial intelligence that focuses on training models to make predictions or decisions based on labeled training data.



### Supervised

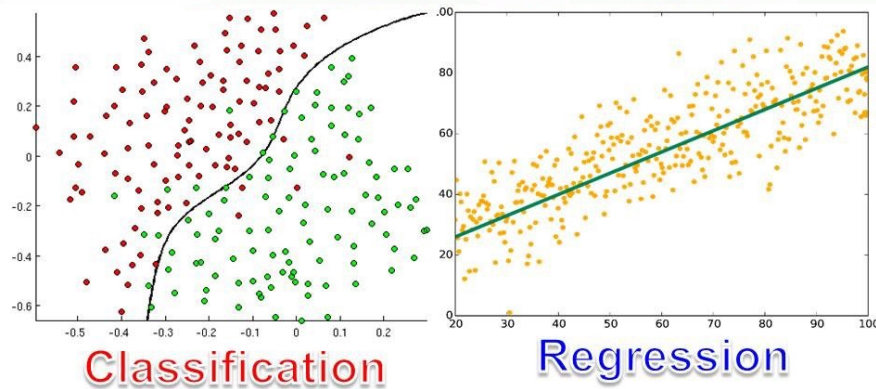
$X_1$	$X_2$	$X_p$	$Y$

Target

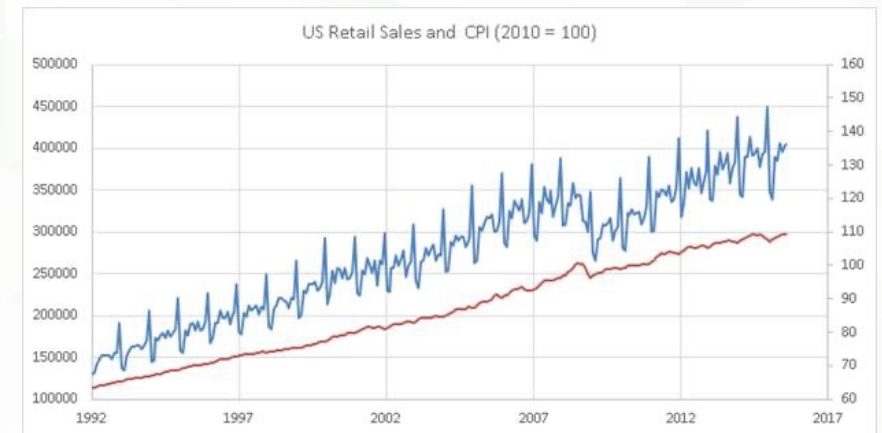
### Un-Supervised

$X_1$	$X_2$	$X_p$	$Y$

No Target



No	Tahun	Jumlah Penduduk
1	2000	174706
2	2001	178125
3	2002	191990
4	2003	201263
5	2004	210984
6	2005	219351
7	2006	225249
8	2007	231121
9	2008	240553
10	2009	250367
11	2010	253178
12	2011	259913
13	2012	268022
14	2013	274089
15	2014	280109
16	2015	285967



## Time Series

# What is Time Series Data

Time series is a sequence of data points collected at successive and uniformly spaced points in time.

## Format:

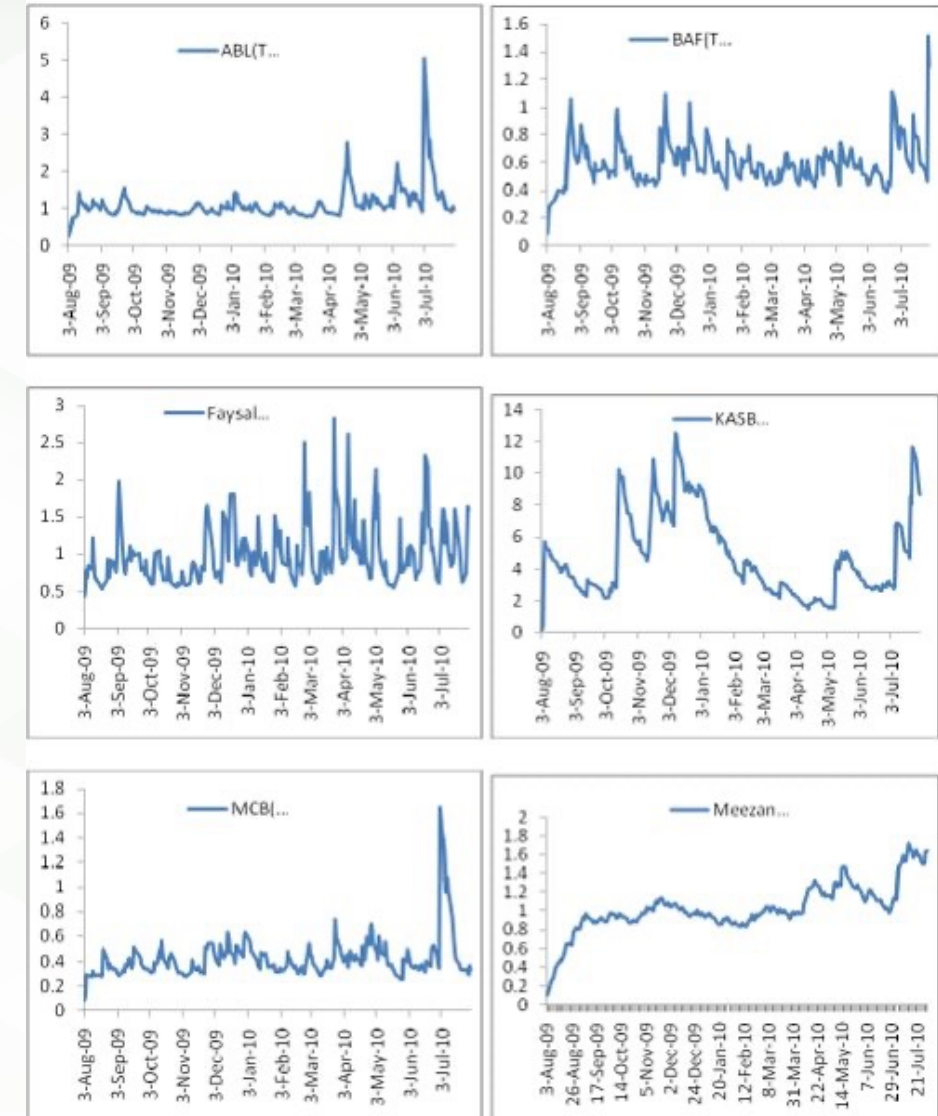
- Daily
- Monthly
- Yearly

## Patterns:

- Seasonality
- Cyclic Patterns
- Noise

## Domains:

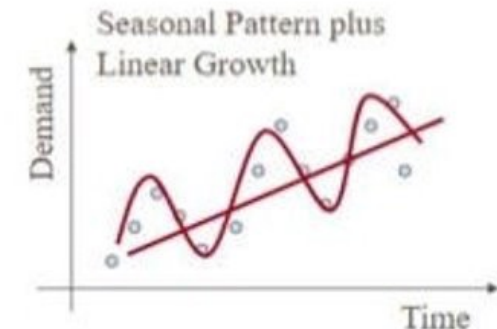
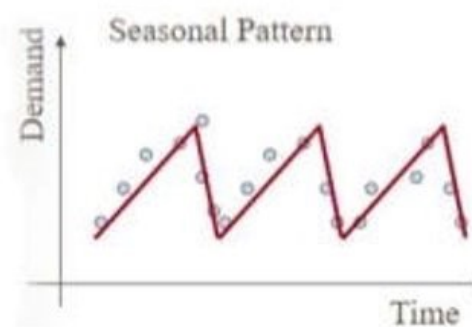
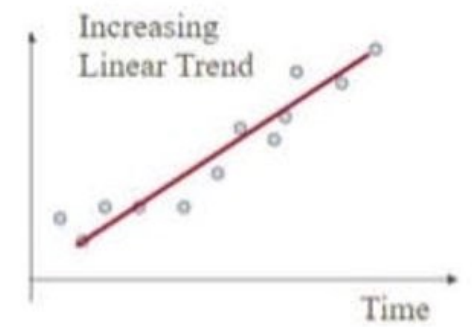
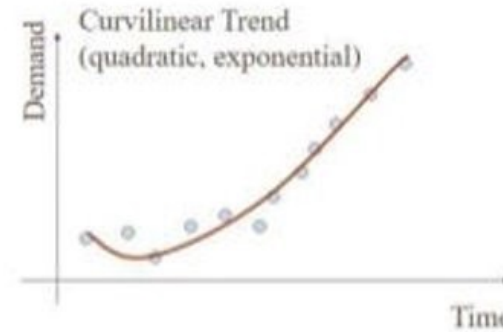
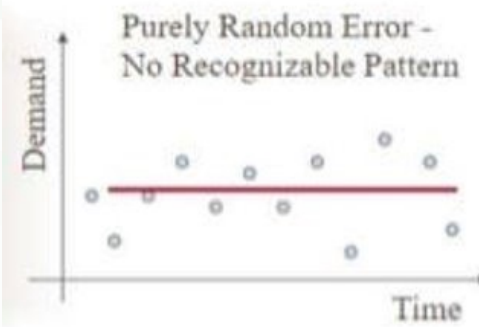
- Economics
- Medicine
- Environment
- Retail and Sales
- Technology



# Patterns Inside Time Series Data

## Patterns

- Trend
- Seasonality
- Cycles
- Noise
- Level
- Structural Breaks
- Autocorrelation
- Stationarity





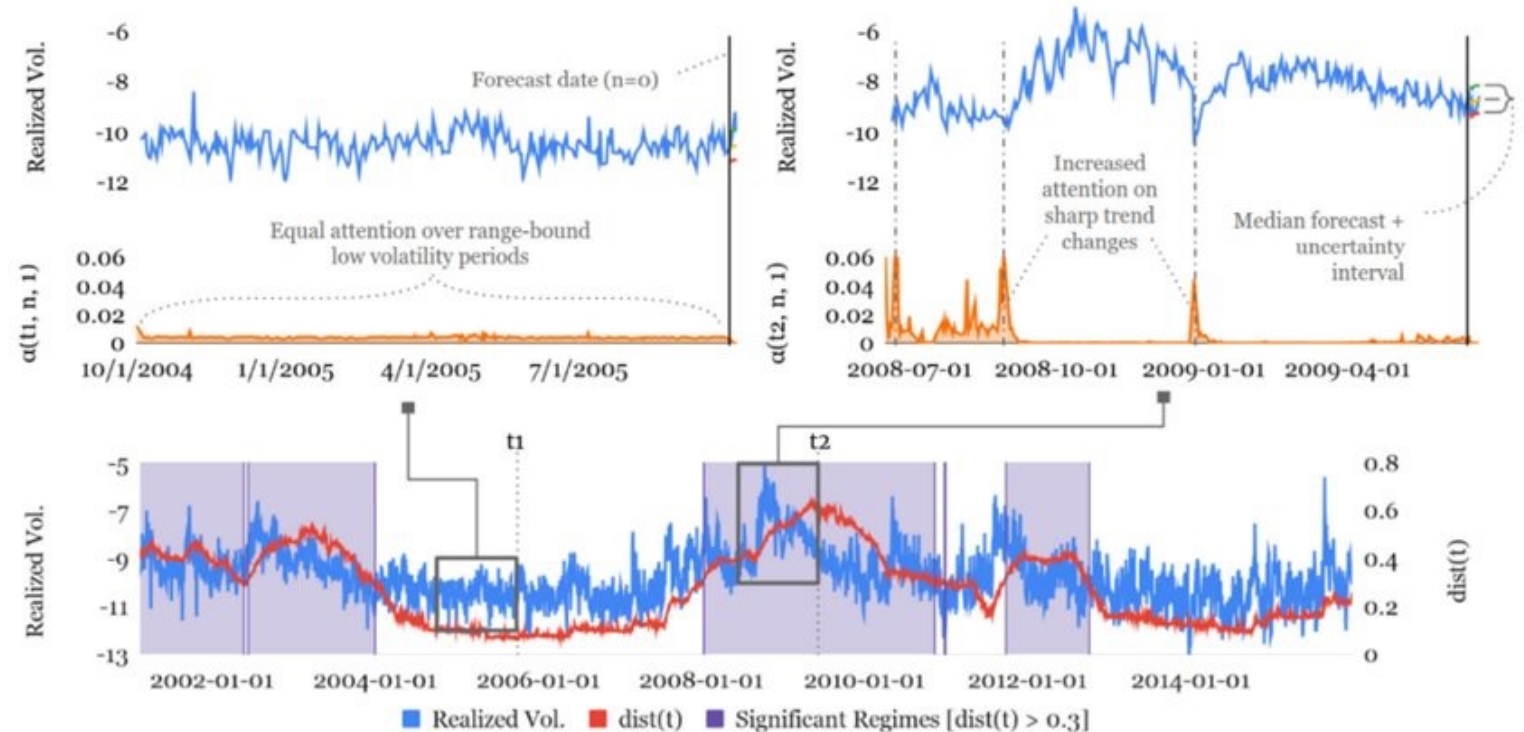
# Examples in Manufacturing

## Data Sources

- Production Output Data
- Energy Consumption Data
- Employee Productivity Data

## The importance:

- Identification of production trends
- Identification of energy consumption patterns that can provide insights into efficiency and potential energy savings.
- Evaluation of employee performance to identify the need for training or skill development.



# Time Series Analysis Scopes in EDA

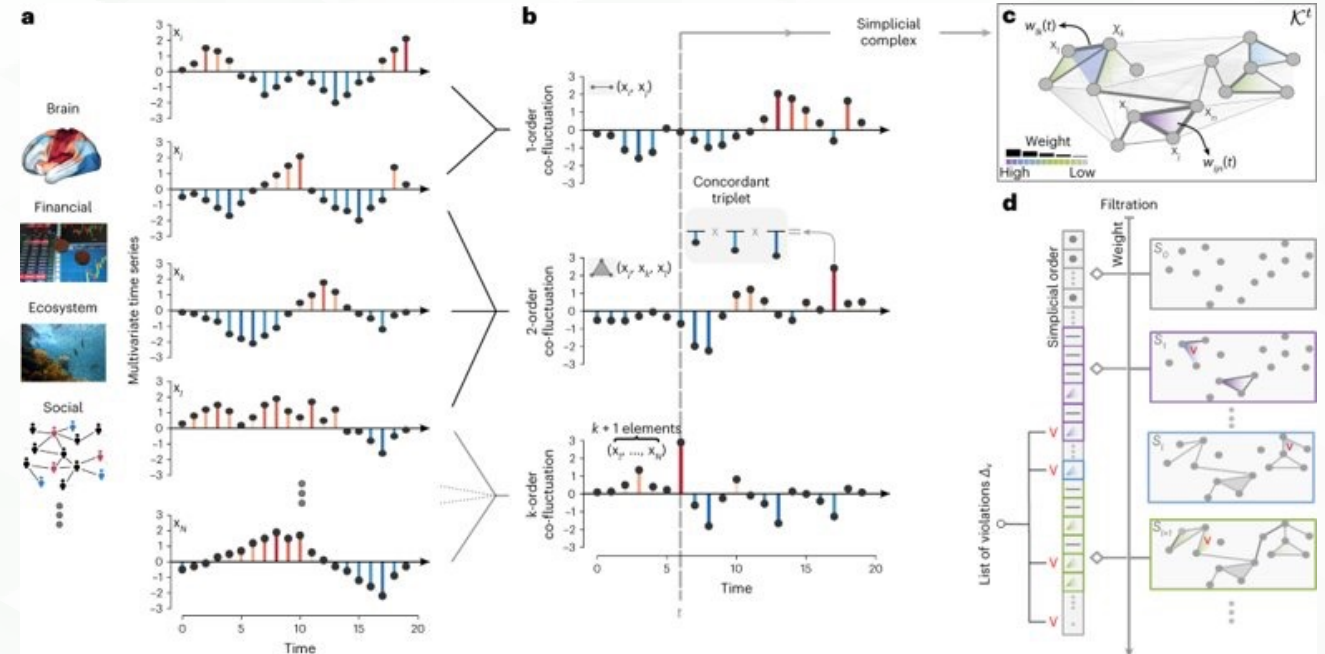
To provide a comprehensive approach to understanding chronological data, ensuring that subsequent analyses or predictions are based on solid foundational knowledge.

## Purpose in EDA:

- Understanding Patterns
- Detect Anomalies
- Inform Assumptions
- Data Quality Assessment

## Methods and Techniques:

- Visualization: Histogram and Density Plots
- Decomposition
- Autocorrelation
- Partial Autocorrelation
- Rolling Statistics
- Stationarity Testing



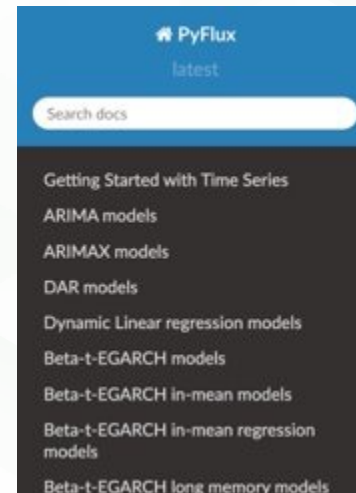
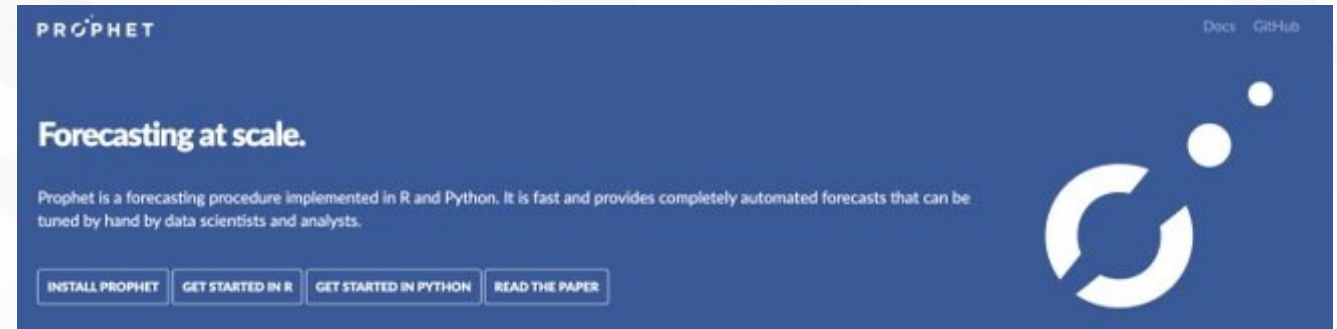
# Framework and Toolkit

## Methods and Techniques:

- Data Importing (CSV, Database, APIs)
- Handling Missing Values
- Time Zone Handling
- Resampling
- Deduplication
- Outlier Detection

## Frameworks and Toolkits:

- Pandas
- StatsModels
- NumPy
- Scikit-Learn
- Prophet
- PyFlux



[Docs](#) » [Introduction](#)

[Edit on GitHub](#)

## Introduction

### What is PyFlux?

PyFlux is a library for time series analysis and prediction. Users can choose from a flexible range of modelling and inference options, and use the output for forecasting and retrospection. Users can build a full probabilistic model where the data  $y$  and latent variables (parameters)  $z$  are treated as random variables through a joint probability  $p(y, z)$ . The advantage of a probabilistic approach is that it gives a more complete picture of uncertainty, which is important for time series tasks such as forecasting. Alternatively, for speed, users can simply use Maximum Likelihood estimation for speed within the same unified API.

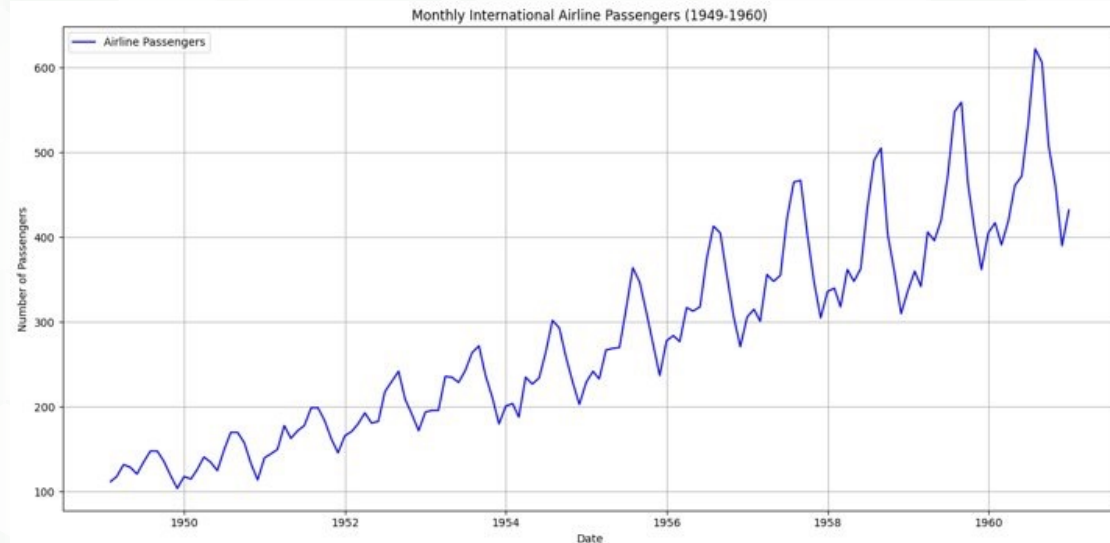


# Visualization

## Common Visualization

- Line Plots
- Seasonal Plots
- Scatter Plots
- Histogram and Density Plots
- Box and Whisker Plots
- Heat Maps
- Lag Plots
- ACF and PACF
- Decomposition Plots
- Rolling Statistics

**ChatGPT Prompt:** Use publicly available Airline Passenger dataset, write Python codes to visualize time series data using Line Plot.



# Visualization

**ChatGPT Prompt:** Use publicly available Airline Passenger dataset, write Python codes to visualize time series data using Scatter, Seasonal, and Lag Plots.

```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from pandas.plotting import lag_plot

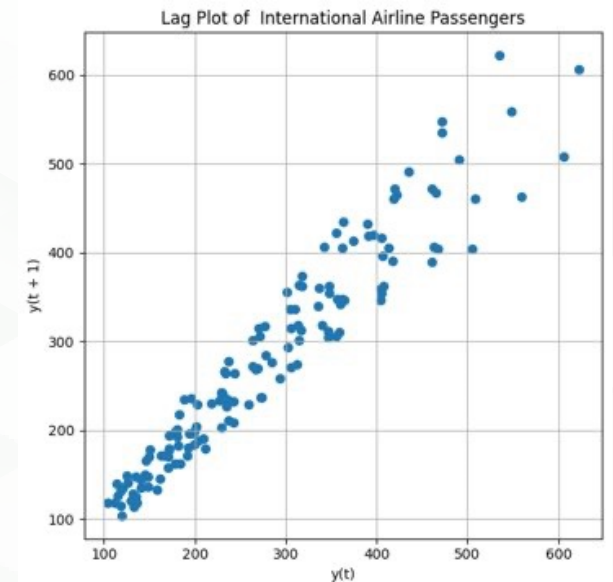
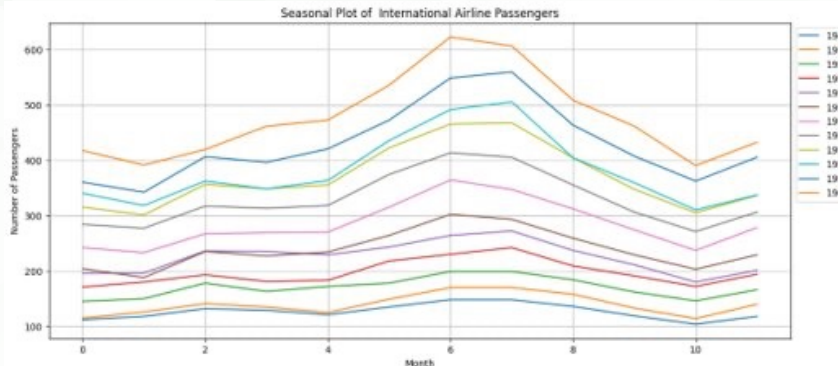
# Load the AirPassengers dataset from statsmodels
dataset = sm.datasets.get_rdataset("AirPassengers").data

# Convert the 'time' column to a datetime format for better visualization
dataset['Date'] = pd.date_range(start="1949-01", periods=len(dataset), freq='M')
dataset.set_index('Date', inplace=True)

# Scatter Plot
plt.figure(figsize=(14, 6))
plt.scatter(dataset.index, dataset['value'], color='blue', facecolors='none')
plt.title('Scatter Plot of International Airline Passengers (1949-1960)')
plt.xlabel('Date')
plt.ylabel('Number of Passengers')
plt.grid(True)
plt.show()

# Seasonal Plot
years = dataset.index.year.unique()
plt.figure(figsize=(14, 6))
for year in years:
    plt.plot(dataset['value'][dataset.index.year == year].values,
             label=str(year))
plt.title('Seasonal Plot of International Airline Passengers')
plt.xlabel('Month')
plt.ylabel('Number of Passengers')
plt.legend(loc="upper left", bbox_to_anchor=(1, 1))
plt.grid(True)
plt.show()

# Lag Plot
plt.figure(figsize=(6, 6))
lag_plot(dataset['value'])
plt.title('Lag Plot of International Airline Passengers')
plt.grid(True)
plt.tight_layout()
plt.show()
```



# Linear Regression

## Applications of Linear Regression:

- Trend Estimation
- Seasonal Adjustment
- Forecasting
- Relation to External Factors
- Intervention Analysis
- Error Analysis

## Special Concerns:

- Auto Correlation
- Non-Stationarity
- Overfitting

**ChatGPT Prompt:** Use publicly available Airline Passenger dataset, write Python codes to demonstrate linear regression of time series data and visualize it using line plot.

