# EX: 2 IMPLEMENT PROGRAMS FOR VISUALIZING DATE : TIME SERIES DATA

**AIM:**

The program visualizes and analyzes time series data using five techniques to identify trends, seasonality, distribution, and pattern.

# ALGORITHM:

1. Load the dataset and parse the date column.
2. Set the date column as the index for time series analysis.
3. Plot a line graph to visualize trends over time.
4. Compute and plot the rolling mean for trend smoothing.
5. Plot a histogram with KDE to analyze data distribution.
6. Create a box plot by year to examine seasonal variations.
7. Generate an autocorrelation plot to identify repeating patterns.
8. Display all visualizations for interpretation.

# PROGRAM:

# Install required libraries (if not already installed) import pandas as pd

import numpy as np

import matplotlib.pyplot as plt import seaborn as sns

from pandas.plotting import autocorrelation\_plot

# Load the uploaded dataset

file\_path = "/content/airline-passengers (1).csv"

df = pd.read\_csv(file\_path, parse\_dates=["Month"], index\_col="Month")

# Display the first few rows print("Dataset Preview:")

print(df.head())

# Set Seaborn style

sns.set(style="darkgrid")

# 1. LINE PLOT: Basic time series visualization plt.figure(figsize=(12, 6))

sns.lineplot(x=df.index, y=df["Passengers"], marker="o", linestyle="-", color="blue")

plt.xlabel("Date") plt.ylabel("Passengers")

plt.title("Airline Passengers - Line Plot") plt.xticks(rotation=45)

plt.show()

# 2. ROLLING MEAN PLOT: Trend smoothing using moving average

df["Rolling\_Mean"] = df["Passengers"].rolling(window=12).mean() # 12-month moving average

plt.figure(figsize=(12, 6))

sns.lineplot(x=df.index, y=df["Passengers"], label="Original", alpha=0.5)

sns.lineplot(x=df.index, y=df["Rolling\_Mean"], label="12-Month Rolling Mean", color="red")

plt.xlabel("Date") plt.ylabel("Passengers")

plt.title("Airline Passengers - Rolling Mean (Trend Analysis)") plt.legend()

plt.xticks(rotation=45) plt.show()

# 3. HISTOGRAM & KDE PLOT: Distribution of values plt.figure(figsize=(10, 5))

sns.histplot(df["Passengers"], kde=True, bins=20, color="green") plt.xlabel("Passengers")

plt.ylabel("Frequency")

plt.title("Distribution of Airline Passengers Data") plt.show()

# 4. BOX PLOT BY YEAR: Checking seasonality

df["Year"] = df.index.year # Extract year from date plt.figure(figsize=(12, 6))

sns.boxplot(x=df["Year"], y=df["Passengers"], palette="coolwarm") plt.xlabel("Year")

plt.ylabel("Passengers")

plt.title("Seasonality Check - Box Plot by Year") plt.xticks(rotation=45)

plt.show()

# 5. AUTOCORRELATION PLOT: Identifying repeating patterns plt.figure(figsize=(12, 6))

autocorrelation\_plot(df["Passengers"])

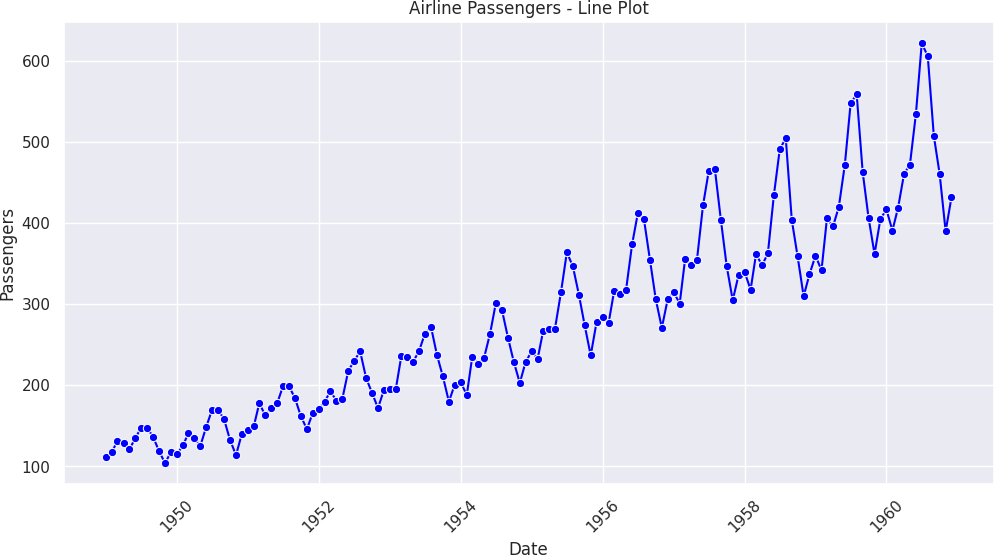
plt.title("Autocorrelation Plot of Airline Passengers Data") plt.show()

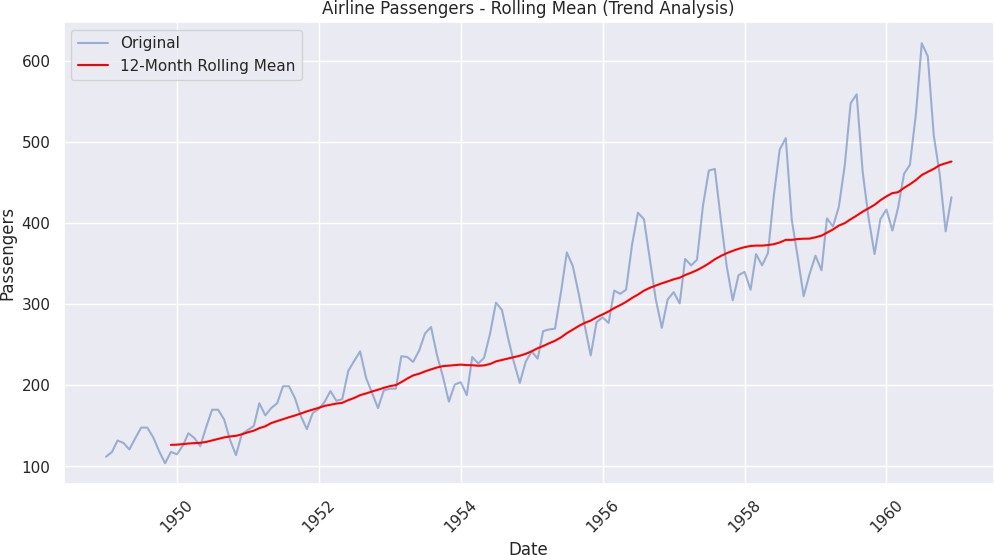
# OUTPUT:

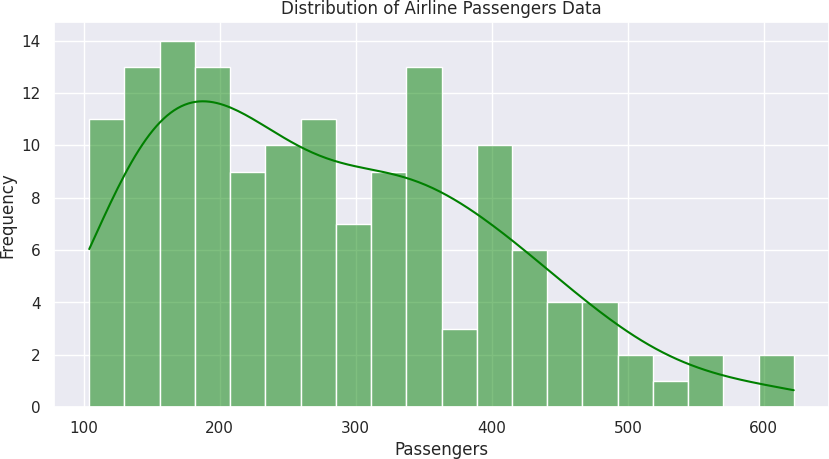
Dataset Preview:

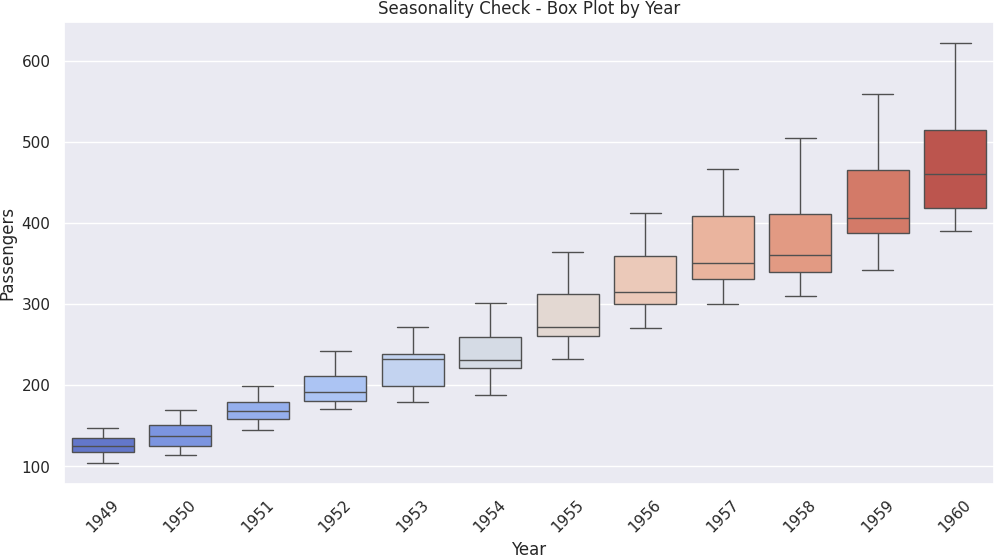
Passengers

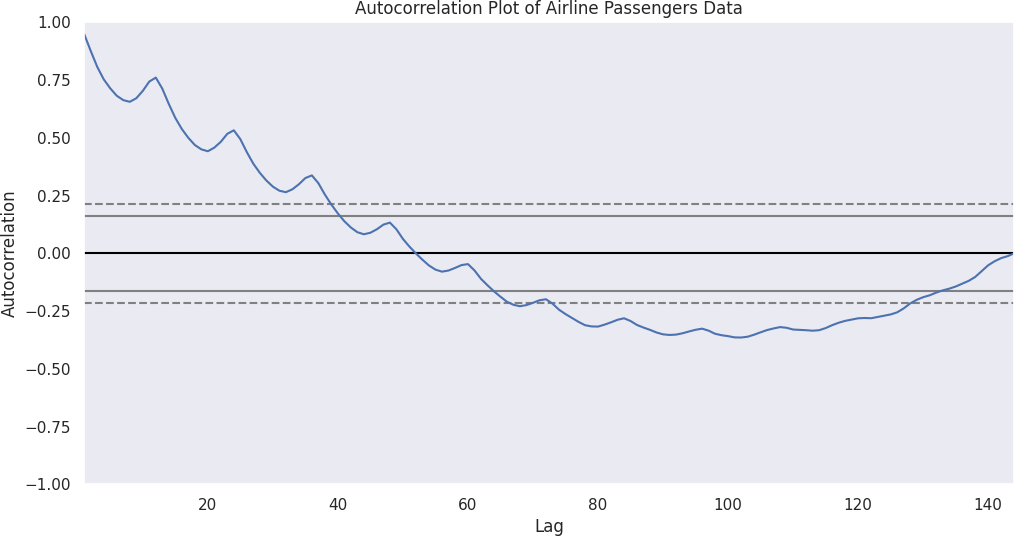
|  |  |
| --- | --- |
| **Month** |  |
| **1949-01-01** | **112** |
| **1949-02-01** | **118** |
| **1949-03-01** | **132** |
| **1949-04-01** | **129** |
| **1949-05-01** | **121** |











**RESULT:**

Thus, the visualization and analysis of the Airline Passengers Time Series Dataset were successfully completed.