Time Series Analysis - Assignment 2: Statistical Analysis

Objective: The goal of this assignment is to analyze the statistical properties of a time series, determine its stationarity, and apply forecasting techniques such as ARIMA, SARIMA, and other statistical models.

Steps:

1. **Dataset Loading and Initial Analysis:**
    - Load the given time series dataset.
    - Ensure the dataset has a date column set as the index.
    - Visualize the dataset to understand its behavior over time.
2. **Statistical Analysis:**
    - Compute and report the **mean** and **variance** of the time series.
    - Check if the mean and variance change over time (indicative of non-stationarity).
    - Perform **Augumented Dickey-Fuller test.** What conclusions can you extract?
3. **Correlation Testing:**
    - Perform the **Durbin-Watson test**.
    - Plot the **Autocorrelation Function (ACF)** and **Partial Autocorrelation Function (PACF)** to examine dependencies and lags. Is the time series correlated?
4. **Frequency Domain Decomposition:**
    - Use **spectral analysis** to decompose the signal in frequency domain (**fft** + **periodogram** approach)
    - Plot the resulting signal
5. **Time Series Forecasting:**
    - Fit an **ARMA** and **ARIMA** model to the time series and evaluate its performance. Check multiple configurations and see the results. Do experiments for both univariate and multivariate forecasting. For forecasting use a window of at least 30 data points.
    - Apply a **SARIMA (Seasonal ARIMA)** model too. Does it have seasonal components? Compare the results to ARIMA.
6. **Evaluation of Forecasting Models:**
    - Use **error measuring metrics** to evaluate model accuracy.
    - Plot actual vs. predicted values for visualization.
7. **Final Submission:**
    - Ensure that all visualizations and results are well-documented.
    - Submit the final Jupyter Notebook containing the implementation and analysis.

Deliverables:

- A Jupyter Notebook (`.ipynb` file) containing:
    - Code implementation for all the steps mentioned above.
    - Clear and concise visualizations.

    - Comments, explanations and conclusions for each step.

Grading Criteria:

- Correctness of the implementation.
- Clarity of visualizations and explanations.
- Proper handling of stationarity testing and transformations.
- Accuracy and evaluation of forecasting models.
- Code readability and explanations.

Deadline:

- Submit your notebook on moodle until march 30, 23:59 for full grade
- Allowed delay: april 6, 23:59 with 2 points penalisation

Notes:

- Use Python with libraries such as `pandas`, `matplotlib`, `seaborn`, `statsmodels`, and `scipy`.
- Ensure proper preprocessing before applying forecasting models.
- Experiment with different ARMA/ARIMA/SARIMA parameters for best results.

Dataset:

- https://www.kaggle.com/datasets/sumanthvrao/daily-climate-time-series-data
- https://www.kaggle.com/datasets/kingki19/semarang-daily-climate-data-2020-2023
- https://www.kaggle.com/datasets/decide-soluciones/air-quality-madrid/data