## Time series and its importance

1. Time-Series(TS) is a sequence of measurements on the same variable collected over time.
2. It is a set of observations, each one being recorded at equally spaced time interval.
3. We extrapolate a past trend to predict future such as movement of short-term interest rate, capacity demand in airline and other sectors.

## Multivariate-time series analysis and prediction using incomplete data

1. TS data is collected for multiple variables over the same period of time. For example, a country's unemployment, GDP and inflation data over an interval period.
2. The TS data measures the changes over time. The time column in the data structure is used to sort the data.
3. TS analysis has played major role in areas like fraud detection, spam email filtering, finance, weather, healthcare, space exploration, manufacturing et al.

## How to predict future using TS?

It is difficult to make predictions, especially about the future : NEILS BOHR, Danish physicist

1. Simple deterministic model such as linear extrapolation

2. Complex deep learning approaches

## STATISTICS BACKGROUND FOR FORECASTING

## Multivariate Time Series (MTS)

A Multivariate time series has more than one time series variable. Each variable depends not only on its past values but also has some dependency on other variables. This dependency is used for forecasting future values. Sounds complicated? Let me explain.

Suppose our dataset includes perspiration percent, dew point, wind speed, cloud cover percentage, etc., and the temperature value for the past two years.

In this case, multiple variables must be considered to predict temperature optimally. A series like this would fall under the category of multivariate time series. Below is an illustration of this:

## TS data intervals

The interval depends on the nature of forecasting. For example, for a nation's GDP, we use yearly interval while sales data can be monthly and air-quality index being monitored on hourly basis.

## Challenges with time-series analysis

1. Data is not independent.

1. Data order is important to keep the data structure intact

1. Order is also important as data is list of observations

## Understanding the ARIMA model

AR - Auto Regression. I - Integrated. MA - Moving average.

Auto-Regressive Integrated Moving Average (ARIMA) is a time series model that identifies hidden patterns in time series values and makes predictions. For example, an ARIMA model can predict future stock prices after analyzing previous stock prices.

Also, an ARIMA model assumes that the time series data is [stationary](https://towardsdatascience.com/stationarity-in-time-series-analysis-90c94f27322). Before implementing the ARIMA model, we will remove the non-stationarity components in the time series.

A non-stationary time series is a series whose properties change over time. A non-stationary time series has trends and seasonality components. Removing the non-stationarity in a time series will make it stationary and apply the ARIMA model.

The properties of time series that should remain constant are variance and mean. Allowing these properties to remain constant will remove the trend and seasonal components. We remove non-stationarity in a time series through differencing.

The differencing technique subtracts the present time series values from the past time series values. We may have to repeat the process of differencing multiple times until we output a stationary time series.

AR - Auto Regression. I - Integrated. MA - Moving average.

They have the following functionalities:

* Auto Regression sub-model - This sub-model uses past values to make future predictions.
* Integrated sub-model - This sub-model performs differencing to remove any non-stationarity in the time series.
* Moving Average sub-model. - It uses past errors to make a prediction.

These sub-models are parameters of the overall ARIMA model. We initialize the parameters using unique notations as follows:

* p: It is the order of the Auto Regression (AR) sub-model. It refers to the number of past values that the model uses to make predictions.
* d: It is the number of differencing done to remove non-stationary components.
* q: It is the order of the Moving Average (MA) sub-model. It refers to the number of past errors that an ARIMA Model can have when making predictions.

## Why do we use Auto ARIMA?

Before we build an ARIMA model, we pass the p,d, and q values. We use statistical plots and techniques to find the optimal values of these parameters.

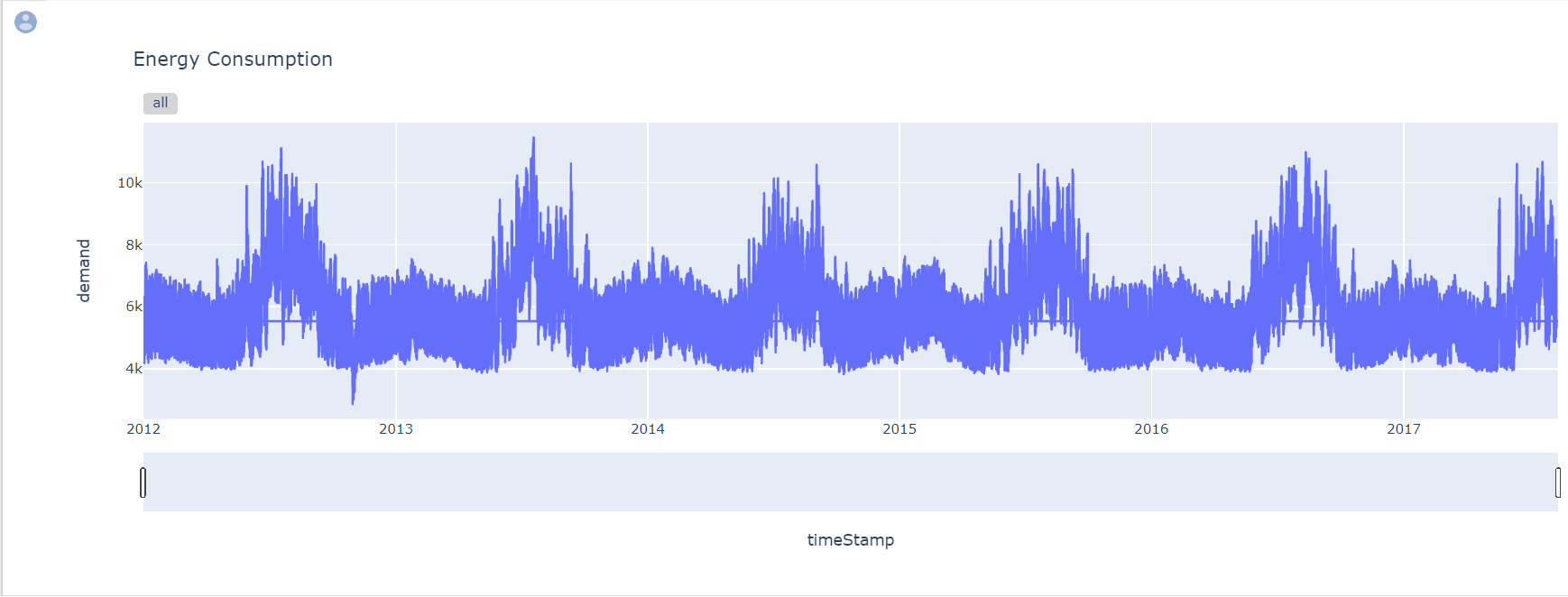
We also use statistical plots such as [Partial Autocorrelation Function plots](https://online.stat.psu.edu/stat510/lesson/2/2.2) and [AutoCorrelation Function plot](https://www.dummies.com/article/technology/information-technology/data-science/big-data/autocorrelation-plots-graphical-technique-for-statistical-data-141241/).

The process of using statistical plots is usually hectic and time-consuming. Many people have difficulties interpreting these plots to find the optimal parameter values. Wrong interpretation leads to people not getting the best/optimal p,d, and q values. It affects the ARIMA model’s overall performance.

Auto ARIMA automatically generates the optimal parameter values (p,d, and q). The generated values are the best, and the model will give accurate forecast results.

Auto ARIMA simplifies the process of building a time series model using the ARIMA model. Now we know how an ARIMA works and how Auto ARIMA applies its concepts. We will start exploring the time series dataset.

Data Set - https://drive.google.com/file/d/1l5MhAnlBYdp5Dk7EvcxzfGJFpvrwbbuw/view



From the output above, the dataset has seasonality (repetitive cycles). Since the dataset has seasonality, we can say it is non-stationary. But still, we need to perform a statistical check using the [Augmented Dickey-Fuller (ADF) test](https://www.machinelearningplus.com/time-series/augmented-dickey-fuller-test/) to assess stationarity in our dataset. The test is more accurate.

## Why DL approach has performed not so well for univariate time series forecasting compared to naive and classical forecasting methods?

Deep learning (DL) approaches have shown remarkable success in various fields, including computer vision, natural language processing, and speech recognition. However, when it comes to univariate time series forecasting, DL models have not always outperformed classical and naive forecasting methods.

There are several reasons why DL approaches may not perform as well for univariate time series forecasting:

Lack of data: DL models require a large amount of data to train effectively. In some cases, univariate time series data may not be abundant, making it challenging for DL models to learn patterns and make accurate predictions.

Overfitting: DL models are prone to overfitting when the training data is not representative of the test data. This is especially problematic when working with time series data, as there may be hidden patterns and trends that are not captured in the training data.

Complexity: DL models can be very complex and have many parameters to tune. This can make it challenging to find the optimal architecture and hyperparameters for a particular time series problem.

Interpretability: DL models can be challenging to interpret, making it difficult to understand why they are making certain predictions. This can be a significant drawback in some applications, where interpretability is crucial.

Performance metric: DL models are often evaluated using mean squared error (MSE) or mean absolute error (MAE). While these metrics are useful for measuring the accuracy of point forecasts, they may not be the best metrics for evaluating the overall performance of a time series forecasting model.

In contrast, classical and naive forecasting methods, such as ARIMA and exponential smoothing, have been developed specifically for time series forecasting and have been shown to perform well in many cases. These methods are often simpler to implement and interpret than DL models and can be effective when working with smaller datasets. Additionally, classical and naive methods often have well-established performance metrics, such as AIC or BIC, that can be used to evaluate the overall performance of a model.

Overall, DL approaches can be a powerful tool for time series forecasting, but they may not always outperform classical and naive methods, especially when working with small datasets or when interpretability