**Week 1**

Time Series

Stationary vs. non-stationary time series: behavior does/does not change over time

Define: Data gathered sequentially in time.

Univariate: Single value at each time step

Eg. Moore’s law chart over time

Multivariate: Multiple value at each time step

Eg. Birth and death rate of Japan over time

By using machine learning we can 1) predict future values and 2) project back into the past.

Imputation: Sometimes there are no data in certain time stamp, but you can apply imputation to replace those missing data.

Memory: Steps dependent on previous one

Innovation: Occasional spike pike, cannot be predicted in the past values

Common Patterns:

1. Trend  
   Time series have specific direction (Eg. Moore’s law: facing upward direction)
2. Seasonal  
   Repeats at predictable interval (Eg. Chart that shows active users on a website will have a regular dip on weekends)
3. Both trend and seasonality
4. Autocorrelated  
   When there is similarity with a lagged (delayed copy) version of itself
5. White noise

Train, Validation, and Test Sets

Fixed Partitioning: Splitting the time series in train, validation and test period. (Eg. If the data has some seasonality, you want to have the whole season in your period)

Roll-Forward Partitioning: Initially splitting at a short training period, but gradually increase and train per day/week (iteration). We use them for forecast the following day/week.

Metrics:

errors = forecasts – actual

mse = np.square(errors).mean()

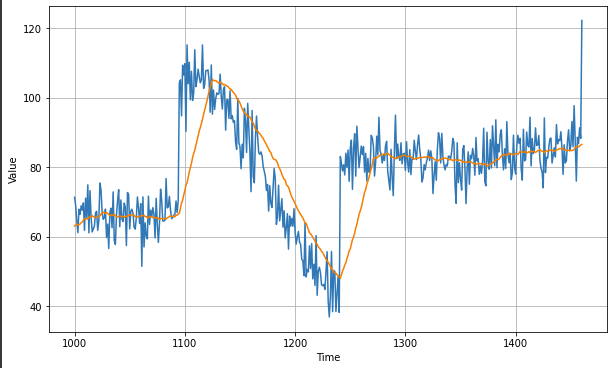
rmse = np.sqrt(mse) // mean to be same scale as original error.

mae = np.abs(errors).mean() // mean absolute error: does not penalize large error as much as mse.

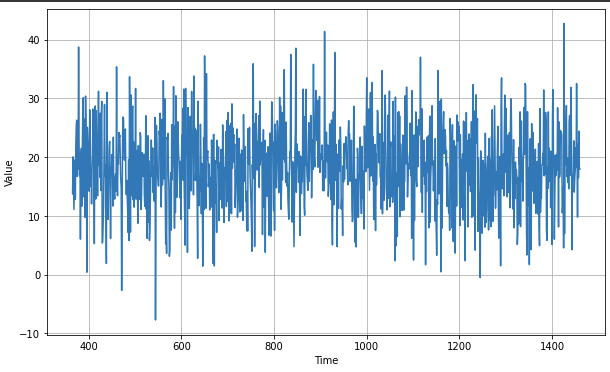
mape = np.abs(errors / x\_valid).mean() // mean absolute percentage error : mean ratio between absolute error and value

Forecasting method : Moving Average and Differencing

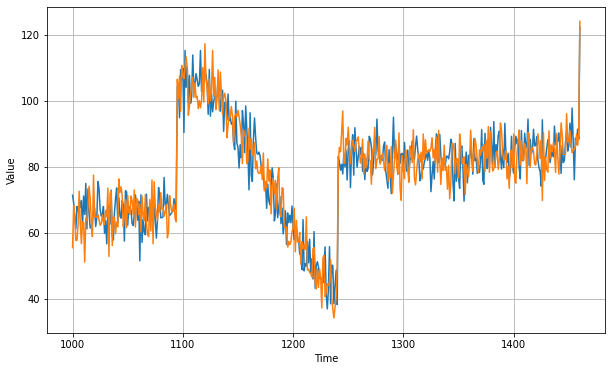
Moving average takes the average of over a fixed period.  
 Cons: it does not anticipate trend / seasonality.



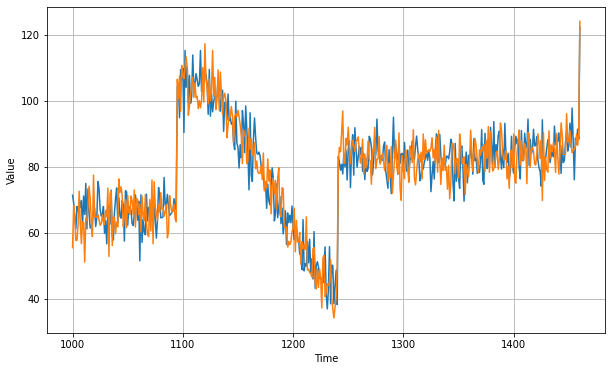
To improve, use differencing to remove trend / seasonality by getting difference between value at time T and value at an earlier period (Eg. T-365days)

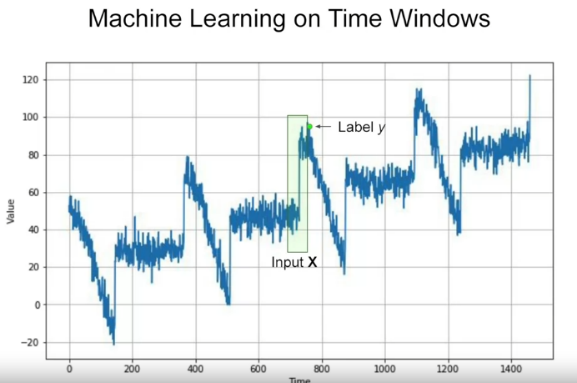


Add the moving average:



Then bring band the trend + seasonality:



**Week 2**

**Sequence bias:** when sequence (order of things) can impact selection of things.

Input X: window set with previous n values

Diagram

Description automatically generatedLabel y: current value with any time stamp

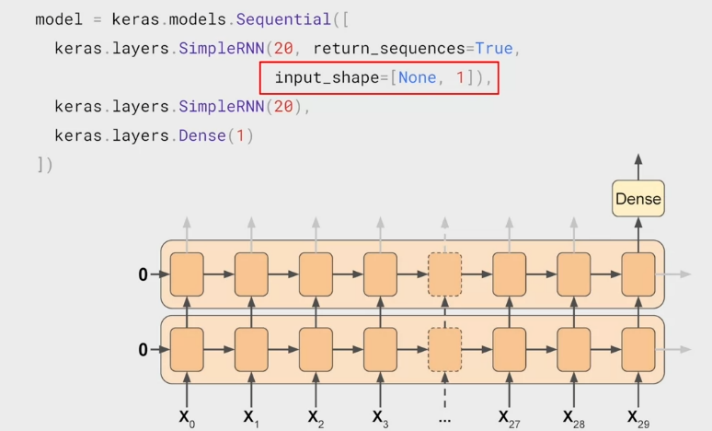
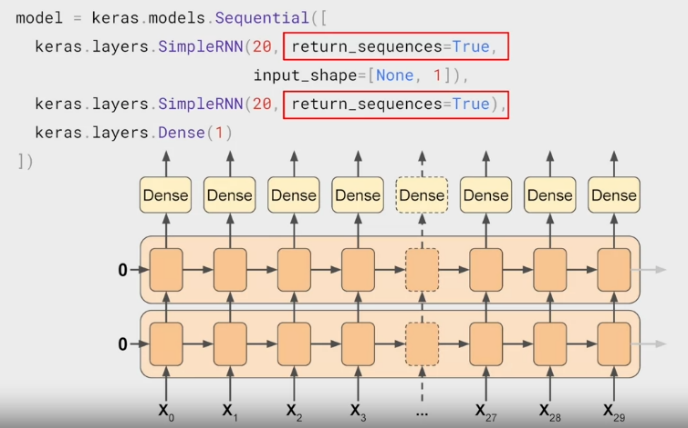
3D

Chart, box and whisker chart

Description automatically generated

Shape = 4 (batch size) x 30 (timestamps) x 1 (univariate)

If memory cell = 3 neurons -> output Y = 4 x 3 matrices. 4 (batch size) x 30 (Y\_0 – Y\_29) x 3 (# of units)

Sequence-Vector / Sequence-Sequence

LSTM (adding Cell State : can also be bi-directional)

