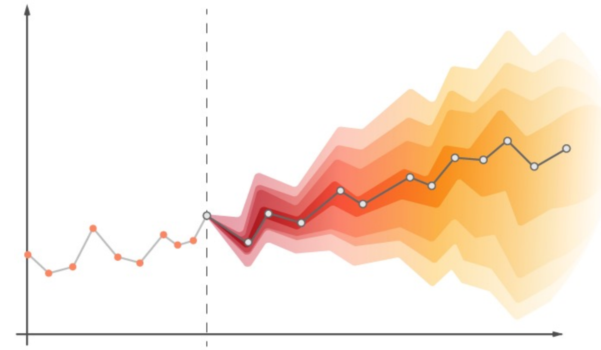
TIME SERIES PREDICTION

2143488 BIG DATA AND ARTIFICIAL INTELLIGENCE

DR. JING TANG

TIME SERIES DATA



- Any data which involved **time component**
- Examples:
  - Meteorology: temperature…
  - Economy and Finance: GDP, spread…
  - Marketing: sales…
  - Industry: power consumption…
  - Web: clicks
  - Genomics: gene expression during cell cycle..

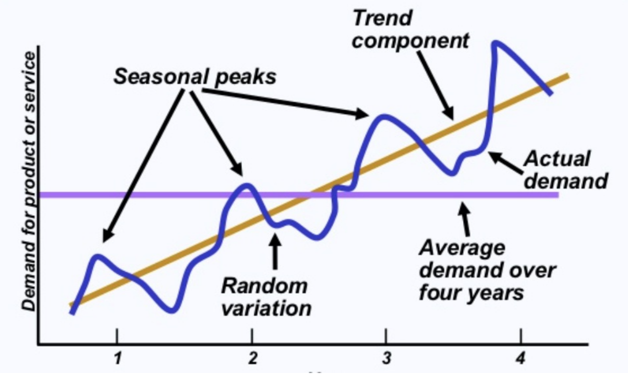
Time-series data has autocorrela
↳ ex: GDP stock price

TIME RELATED CONCEPTS

Concept	Def,	Scalar Class	pandas Data Type	Primary Creation Method
Date times	specific date & time with timezone	Timestamp	datetime64[ns] datetime64[ns, tz]	to_datetime date_range
Time deltas	absolute time duration	Timedelta	timedelta64[ns]	to_timedelta timedelta_range
Time spans	defined by a point in time and its associated frequency	Period	period[freq]	Period period_range
Date offsets	relative time duration that respects calendar arithmetic	DateOffset	None	DateOffset

WHY TIME SERIES?

- **Description** of its salient features
  ↳ it's own features
- **Understanding** of the mechanism or getting meaningful insights from it
- **Control** of the process producing of it
- **Forecasting** its future looking at past data behavior



TYPES OF VARIATION 1

- **Long-term movement or Trend**:
  - Increase or decrease or remain stable during a prolonged time interval
  - Common to change direction
- **Seasonal short-term movement**:
  - Periodic temporal fluctuations that show the same variation
  - Recur over a fixed period < 1 year
  - In hourly, daily, weekly, quarterly, or monthly pattern
  - Different in social conventions as holidays and festivities, weather season, and climate conditions

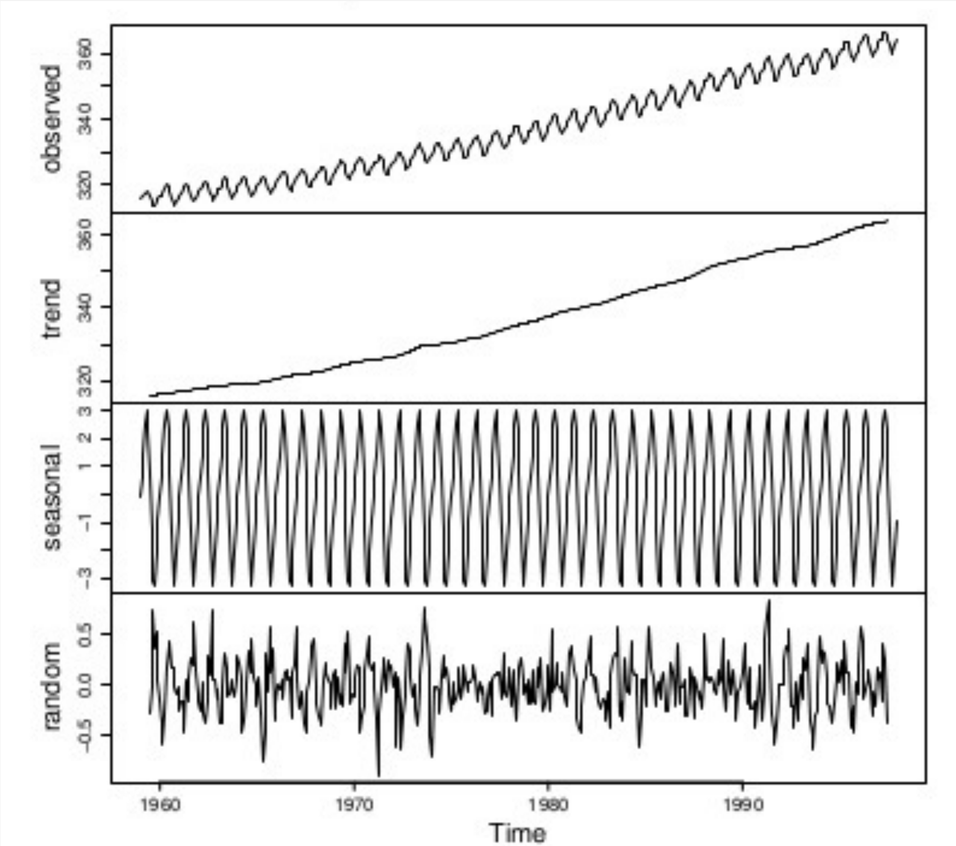
TYPES OF VARIATION 2

- **Cyclic short-term movement** :
  - Rises/falls that not a fixed period (>1 year)
  - Without a specific predetermined length of time
  - E.g., economic cycles
- **Random or Irregular fluctuations**:
  - Uncontrollable, unpredictable and erratic
  - E.g., earthquakes, wars, flood…

- First 3 components are predictatable signals
- Last component is not predicatable (Noise)

DECOMPOSITION OF TIME SERIES

- statsmodels.tsa.seasonal.seasonal_decompose



trend seasonalality

$e.g., x_t = \alpha + \beta t + \epsilon_t$
Differencing to remove the trend

$e.g., x_t = m_t + s_t + \epsilon_t$
Differencing to remove the seasonal impact

DEFINE YOUR FORCASTING MODEL

history future

- Input and output
- Granularity level: average or sum in daily, weekly, or monthly, etc.
- Horizon: short-term vs long-term
- Endogenous and exogenous features
- Univariable or multivariable
- Single-step or multi-step structure

  ↳ predict 1 step in the fture ↳ multiple steps
- Contiguous or noncontigous (missing) time series values
  - Fixed interval between obervations

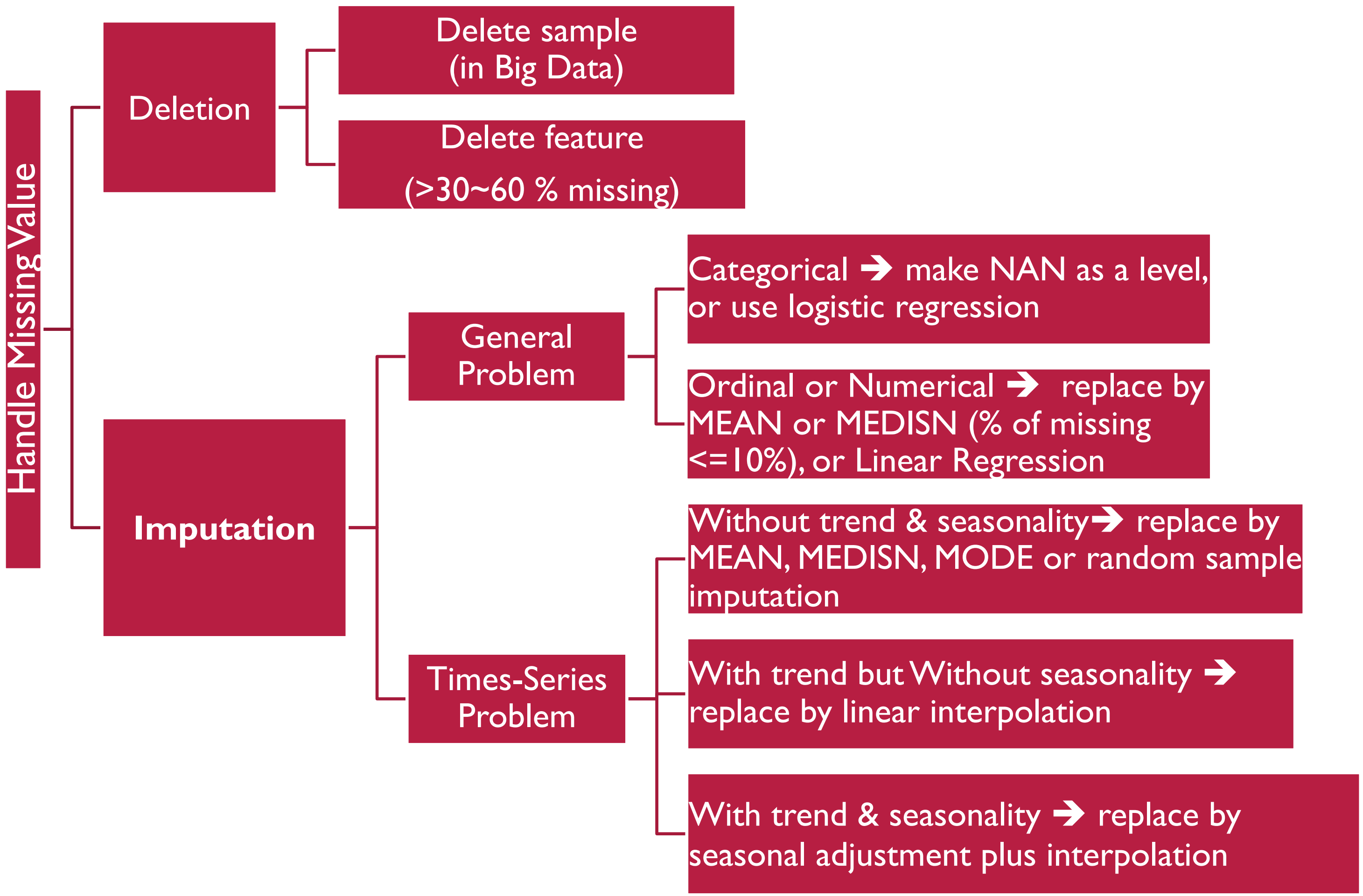
UNIVARIATE VS. MULTIVARIATE TIME SERIES

- **Univariate time series**:
  - A single variable
  - No causes or relationships
  - Lag values of itself as independent variables
- **Multivariate time series**:
  - Several related time series are observed simultaneously
  - E.g.,
    - how sea level is affected by temperature and pressure
    - how sales are affected by price and economic conditions

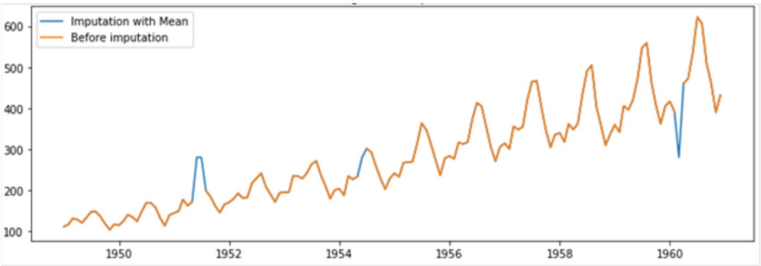
SINGLE-STEP VS. MULTI-STEP TIME SERIES FORECASTING

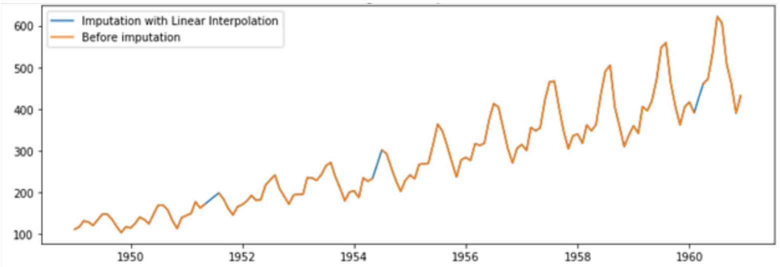
- **Single-step**:
  - Predict the observation at the next time step
- **Multi-step**:
  - Predict a sequence of values in a times series
  - E.g., stock prices, traffic volume, electricity consumption…
  - 4 common strategies
    - Direct multi-step
    - Recursive multi-step
    - Direct-recursive hybrid multi-step
    - Multiple output

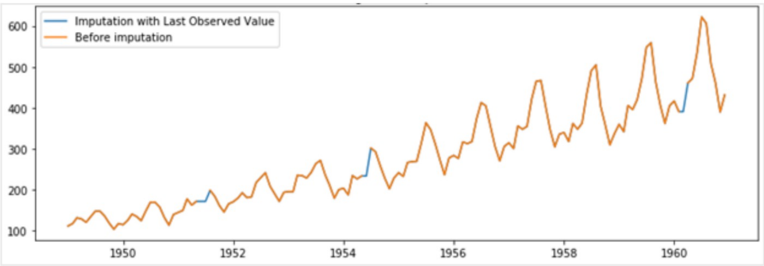
HANDLE MISSING VALUE 1

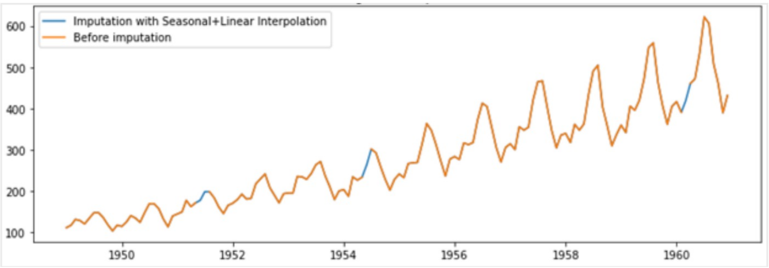


HANDLE MISSING VALUE 2









PROCEDURE 1

- Load data: Batch/Real-time
- Explore data:
  - Size (col & row)
  - Quality
    - Missing value
    - Measurement accuracy
    - Time of measurement: actual time-stamp of the data collection
    - Synchronization: <10 seconds between the collected time stamp from any 2 independent data sources
    - Latency: actual measurement time vs value loading time

PROCEDURE 2

- Preprocess data:
  - Split train-test
  - Time driven features
    - Time of day: 0-23
    - Day of week: 1 (sun) -7 (Sat)
    - Day of month: 1-28/29/30/31
    - Month of year: 1-12
    - Weekend: 0 (weekday) -1 (weekend)
    - Holiday: 0 (regular day) -1 (holiday)
    - Fourier terms:
      - yearly, weekly and daily seasons → 3 Fourier terms

PROCEDURE 3

- Preprocess data:
  - Independent features:
    - Lag feature:
      - time-shifted values of y
      - Lag 1, 2, 3…
    - Long-term trending
      - Linear growth of y between years
- Model:
  - Debug a Time Series model to make the model work for train dataset
  - Preprocess (same as train dataset) and predict test dataset
  - Evaluate regression result of test dataset

TIME SERIES FORECASTING MODELING TECHNIQUES 1

- Simple Moving Average (SMA):
  - assigns an equal weighting to all values

$$SMA = \frac{A_1 + A_2 + \ldots + A_n}{n}$$

where:

$A =$ Average in period n

$n =$ Number of time periods

- Exponential Moving Average (SMA):
  - gives a higher weighting to recent prices

$$EMA_t = \left[V_t \times \left(\frac{s}{1+d}\right)\right] + EMA_y \times \left[1 - \left(\frac{s}{1+d}\right)\right]$$

where:

$EMA_t =$ EMA today

$V_t =$ Value today

$EMA_y =$ EMA yesterday

$s =$ Smoothing

$d =$ Number of days

TIME SERIES FORECASTING MODELING TECHNIQUES 2

- Autoregreation Integrated Moving Average (ARIMA(p,d,q))
  - p: AR
  - d: nonseasonal differences needed for stationary
  - q: lagged forecast errors

↳ d=0 → no differentiation
d=1 → diff 1 time

- General Multiple Regression

If d=0: $y_t = Y_t$

If d=1: $y_t = Y_t - Y_{t-1}$

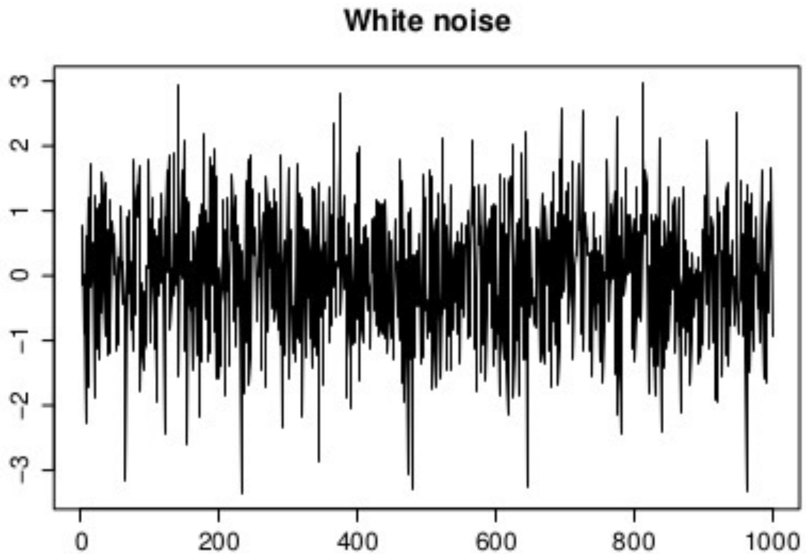
★ If d=2: $y_t = \underbrace{(Y_t - Y_{t-1})}_{\text{diff}} - \underbrace{(Y_{t-1} - Y_{t-2})}_{\text{previous diff}} = Y_t - 2Y_{t-1} + Y_{t-2}$

$$\hat{y}_t = \mu + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \ldots - \theta_q e_{t-q}$$

ACF VS. PACF

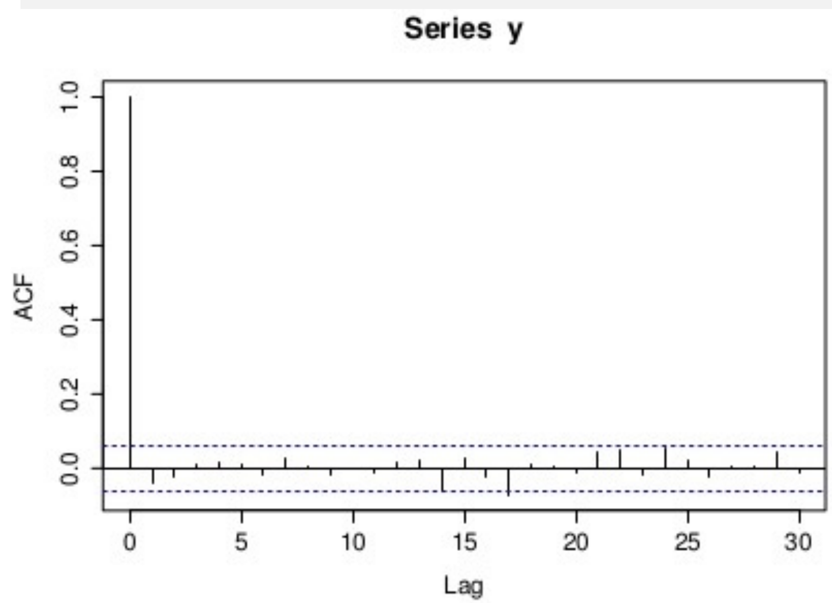
- **ACF:** an (complete) auto-correlation function
    - Find auto-correlation of any series with its lagged values
- **PACF**: a partial auto-correlation function
    - Conditional correlation
    - With an assumption that some other variables are consider
    - Find correlation of the residuals with the next lag value hence ‘partial’

ACF WITH GAUSSIAN RANDOM ϵ_t

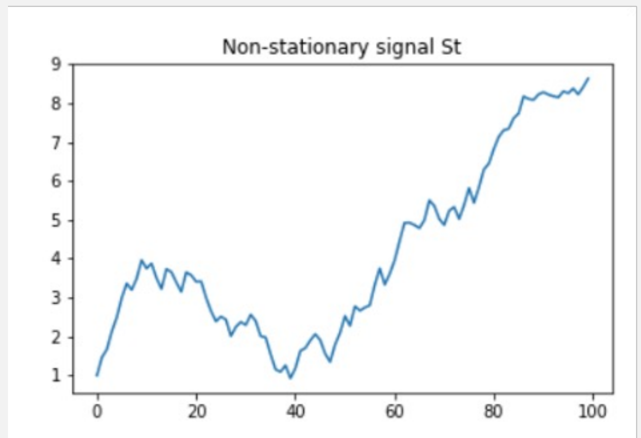


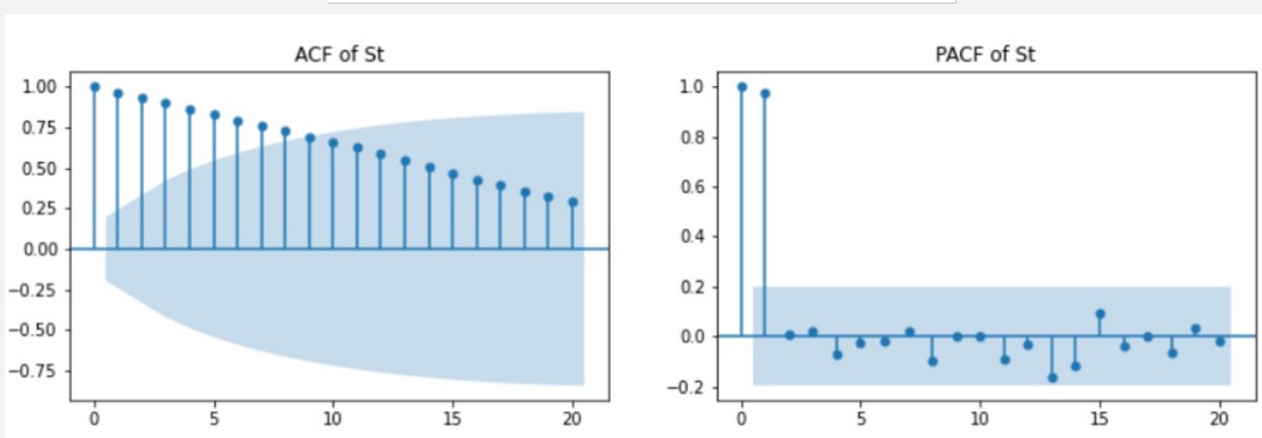
Autocorrelation Function

$$Corr(\epsilon_t, \epsilon_{t+k}) = \frac{\sum_{t=1}^{N-k}(\epsilon_t - \bar{\epsilon})(\epsilon_{t+k} - \bar{\epsilon})}{\sum_{t=1}^{N-k}(\epsilon_t - \bar{\epsilon})^2}$$



ACF AND PACF WITH NON-STATIONARY SIGNAL





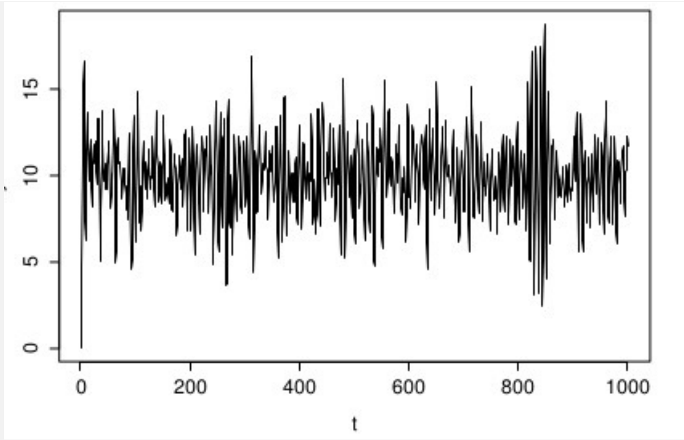
AR MODEL

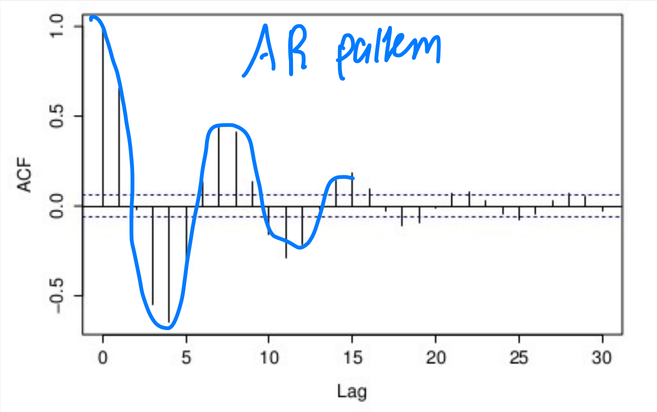
- A process x_t is said to be an **autoregressive process** of order p, $AR(p)$, if

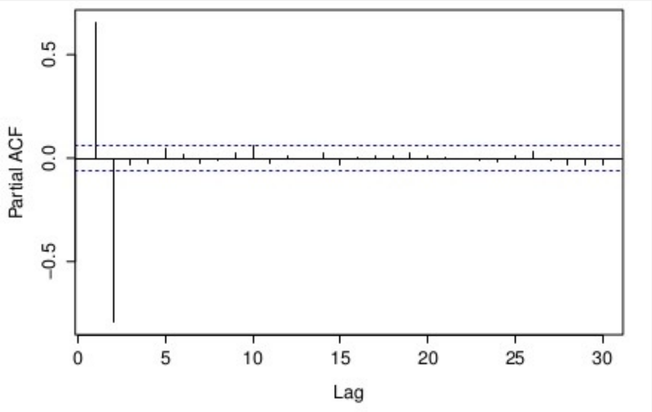
$$x = a_0 + a_1 x_{t-1} + \cdots + a_p x_{t-p} + \epsilon_t$$

- **Auto**: like a linear regression, but not on independent varies but on its **past values**
- Properties of **stationarity** depends on the coefficiencies $a_i, 1 \ldots n$
- **PACF plot tell us the order of the AR model**

AR(2)


15
10
5
0
0
200
400
600
800
1000
t


AR pattern
ACF
1.0
0.5
0.0
-0.5
0
5
10
15
20
25
30
Lag


Partial ACF
0.5
0.0
-0.5
0
5
10
15
20
25
30
Lag

GOODNESS-OF-FIT MEASURES 1

- Akaike Information Criterion (AIC)

$$AIC = -\frac{2}{N} \times LL + 2 \times \frac{k}{N}$$

  - *N* is the number of examples in the training dataset
  - *LL* is the log-likelihood of the model on the training dataset
  - *k* is the number of parameters in the model
- To use AIC for model selection, we simply choose the model giving **smallest AIC** over the set of models considered.

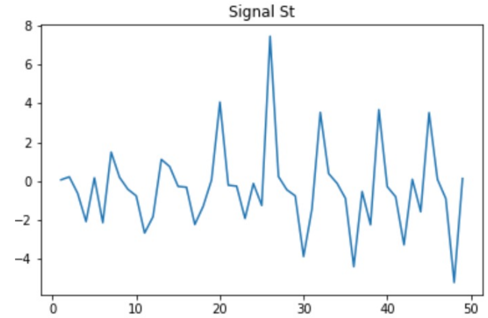
GOODNESS-OF-FIT MEASURES 2

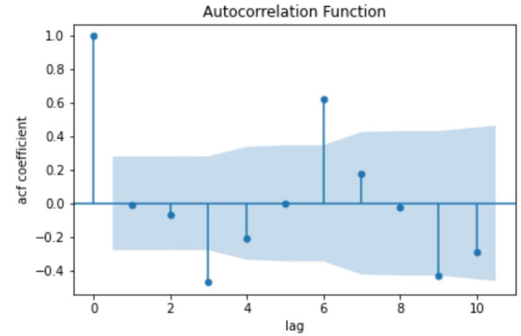
- Bayesian Information Criterion (BIC)

$$BIC = -2 \times LL + \log N \times k$$

  - *N* is the number of examples in the training dataset
  - *LL* is the log-likelihood of the model on the training dataset
  - *k* is the number of parameters in the model
- To use BIC for model selection, we simply choose the model giving **smallest BIC** over the set of models considered
- Unlike the AIC, the BIC penalizes the model more for its complexity

ARMA MODEL





- Autoregreation Moving Average $ARMA(p,q)$
  - p: AR
  - q: lagged forecast errors

$$x_t = a_0 + a_1 x_{t-1} + \cdots + a_p x_{t-p} + \epsilon_t + \gamma_1 \epsilon_{t-1} + \cdots + \gamma_q \epsilon_{t-q}$$

- **PACF plot tell us the order of the AR model**
- **ACF plot tell us the order of the MA model, if it has a sharp cut-off after lag q**

ARIMA MODEL

- Autoregreation Integrated Moving Average $ARIMA(p, d, q)$
  - p: AR
  - d: nonseasonal differences needed for stationary
  - q: lagged forecast error
- x are not put into the model directly, but the difference terms. When d=1

$$\Delta x_t = x_t - x_{t-1}$$

$$\Delta x_t = a_0 + a_1 \Delta x_{t-1} + \cdots + a_p \Delta x_{t-p} + \epsilon_t + \gamma_1 \epsilon_{t-1} + \cdots + \gamma_q \epsilon_{t-q}$$

- GridSearchCV for best model

PROCEDURE

1. Visulize the time series
2. Seasonal_decompose
3. Plot ACF/PACF charts and find optimal parameters
4. Build the ARIMA model
5. Make Predictions (smallest AIC)

HW5: FORECAST DAILY AVERAGE PRESSURE (PRES) IN TIANTAN, BEIJING IN 2017 MARCH

1. Select attributes (univariable model PRES)
2. Group data
   - Set time feature as index

     ```
     df['datetime']=df['year'].astype(str).str.cat([df['month'].astype(str),df['day'].astype(str),df['hour'].astype(str)], sep='-')
     df['datetime']=pd.to_datetime(df['datetime'],format='%Y-%m-%d-%H')
     ```

   - Aggregate pressure daily by average

     ```
     df.resample('D').mean().round(2)
     ```
3. Visualize data (the trend, seasonal pattern of pres)
4. Calculate ACF and PACF
5. Apply ARIMA
6. Evaluate the result is ok or not? How to improve it?

- Throw everything, use pres
- y is currently hour-based
  ↳ change to daily base