SI 671/721:
Mining Time-Series Data (II): Forecasting

Lecture 10
Fall 2021

Instructor: Prof. Paramveer Dhillon
dhillonp@umich.edu
University of Michigan

UMSI

Announcements

- No class next week. We’ll meet again on 11/29
- Today’s the last lab for the course.
- We will make the solutions of the discussion labs available to the students.

Time Series Forecasting

Recap: Time series data

A list of timestamped values (measurements):

$$Y = \left\{ (y_1, t_1), (y_2, t_2), \ \ldots, (y_n, t_n) \right\}$$

Numerical measures

Timestamp matters

In time series forecast, we will use y (to indicate the targets)

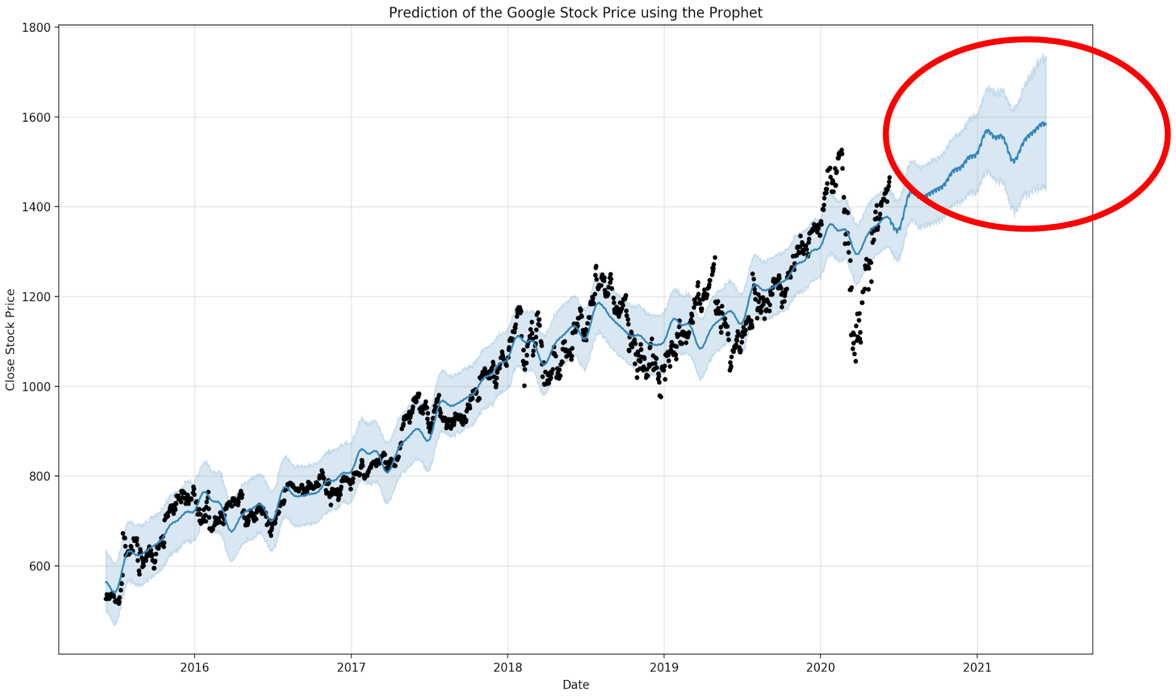
Prediction and forecasting

- Like for sequence data, prediction & forecasting are major data mining tasks for time series data

- Prediction: given any t, find y_t

- Forecasting: given t in the future ($t > t_n$), find y_t

Time series forecasting



https://towardsdatascience.com/time-series-forecasting-predicting-stock-prices-using-facebooks-prophet-model-9ee1657132b5

Applications of time series forecasting

- Financial market
- Weather
- Traffic load
- Business planning
- Enrollment
- Healthcare, epidemics
- Election
- …

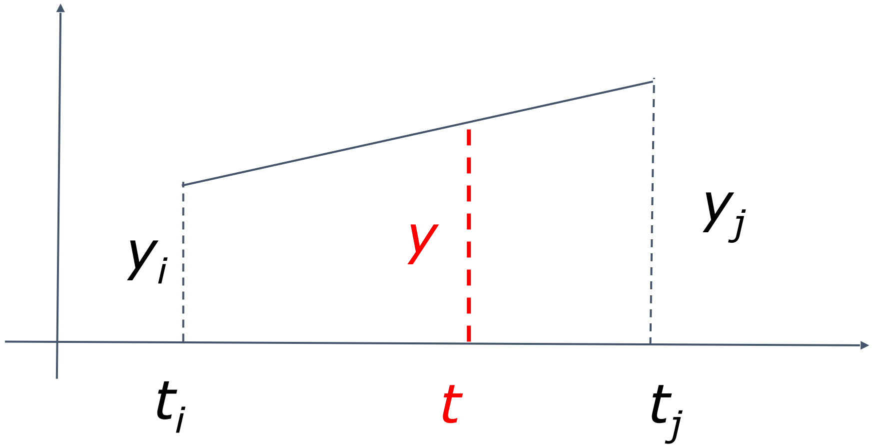
Predicting the “past” is also useful

- Fill missing values
- Understand correlation & causation
- Counterfactual analysis
- …
- It helps us learn how to predict for the future

Interpolation

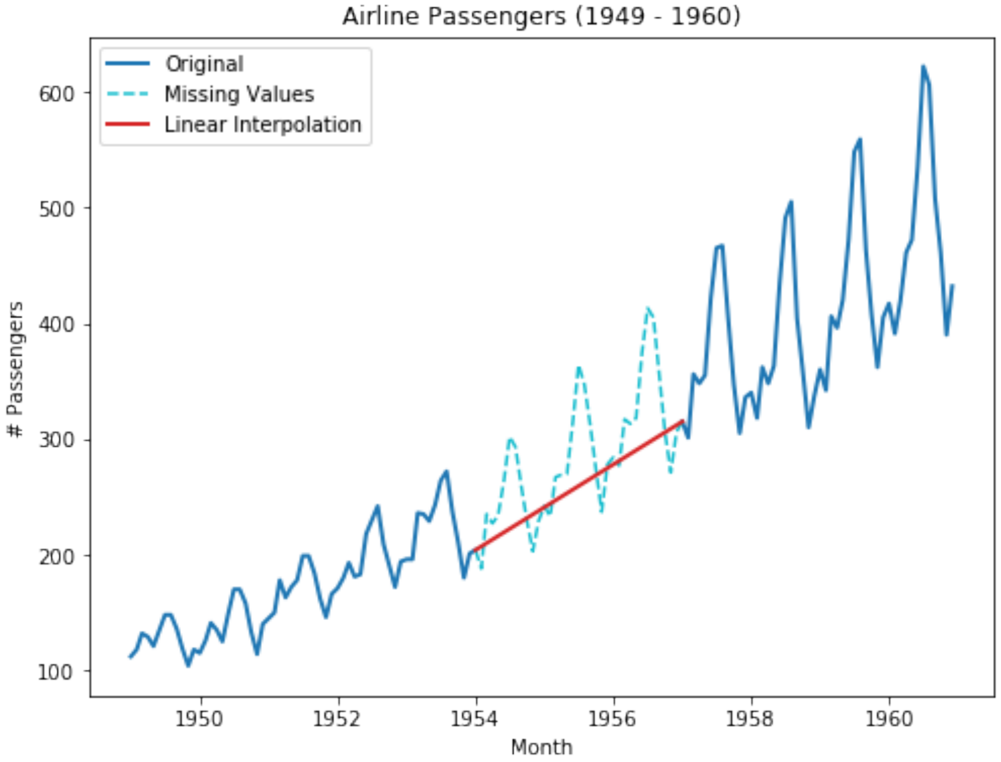
- Known: (y_i, t_i), (y_j, t_j), $t_i < t_j$
- Want to know: y_t, where $t_i < t < t_j$
- Often used to infer missing value based on values nearby

$$y_t = y_i + \frac{t - t_i}{t_j - t_i} \bullet \left(y_j - y_i\right)$$



Interpolation

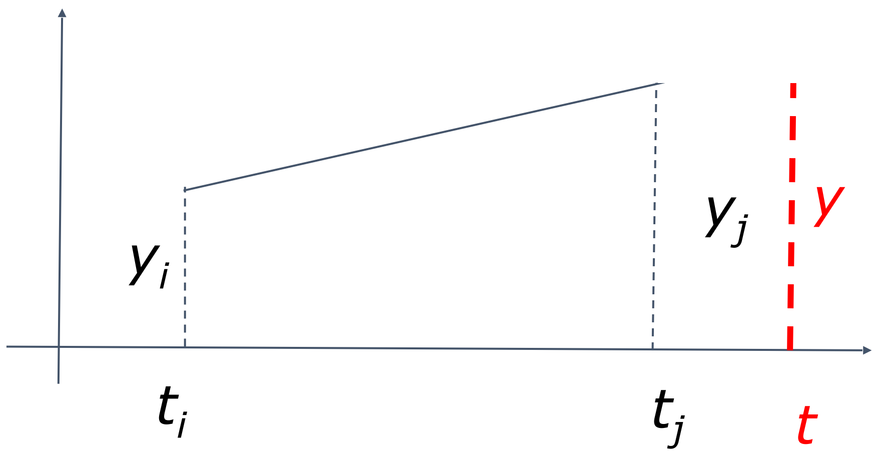
- Commonly used to resample a time series
- Transform an irregular time series into an equally spaced one



Extrapolation

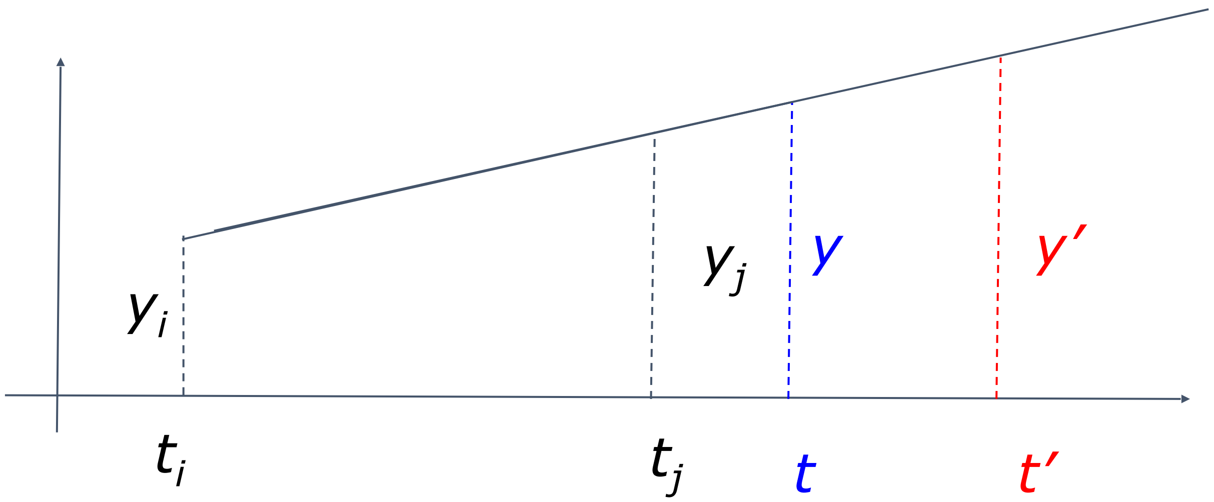
- Known: $(y_1, t_1), \ldots, (y_i, t_i), (y_j, t_j), t_i < t_j$
- Want to know: y_t, where $t_i < t_j < t$
- Simple solution (same as interpolation)

$$y_t = y_i + \frac{t - t_i}{t_j - t_i} \bullet \left(y_j - y_i\right)$$



Simple extrapolation for forecasting

- Only considers two existing data points
- Assumes **local trend** generalizes to the future
- Leads to inflation
- Doesn't explain variance
- Normally we don't do that in practice



Beyond simple extrapolation

- Better prediction methods should consider the statistical properties and patterns of the whole series:
  - Mean
  - Variance
  - Covariance
  - Trend
  - Seasonality
  - ...

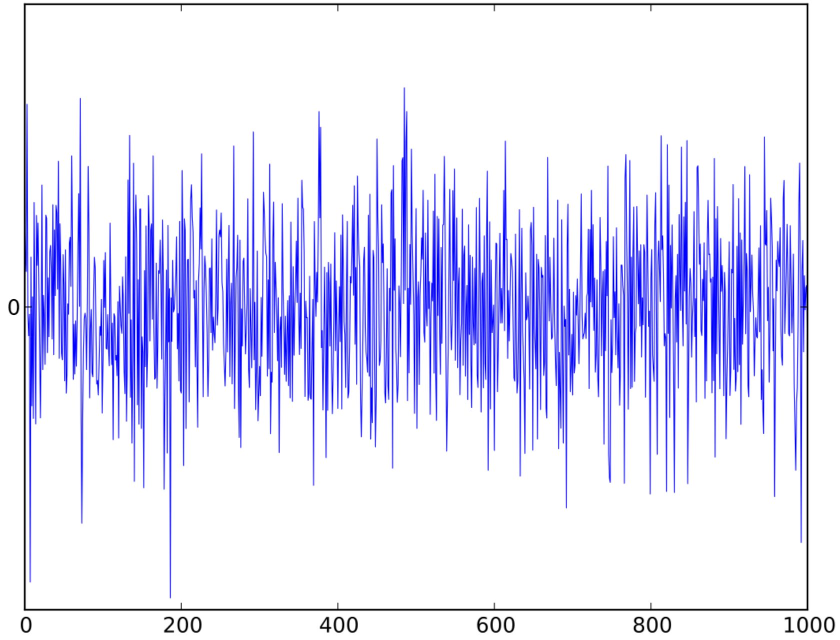
Stationary Time Series

Stationary time series

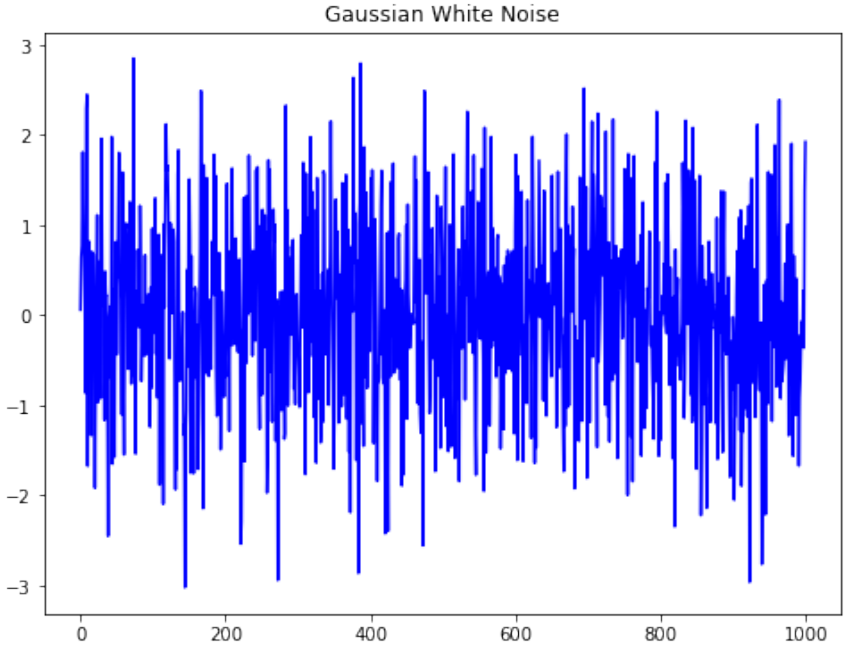
- If only the statistical properties (e.g., mean and variance) of the observations never change!
- Such a time series is known as a stationary time series:
  - flat trend (constant mean)
  - constant variance
  - zero covariance (over different timestamps)
- For example, white noise

Example of stationary time series

- Left: a white noise series ($y \sim WN(0, \sigma^2)$)
- Right: a Gaussian white noise series ($y \sim Gaussian(0, \sigma^2)$)

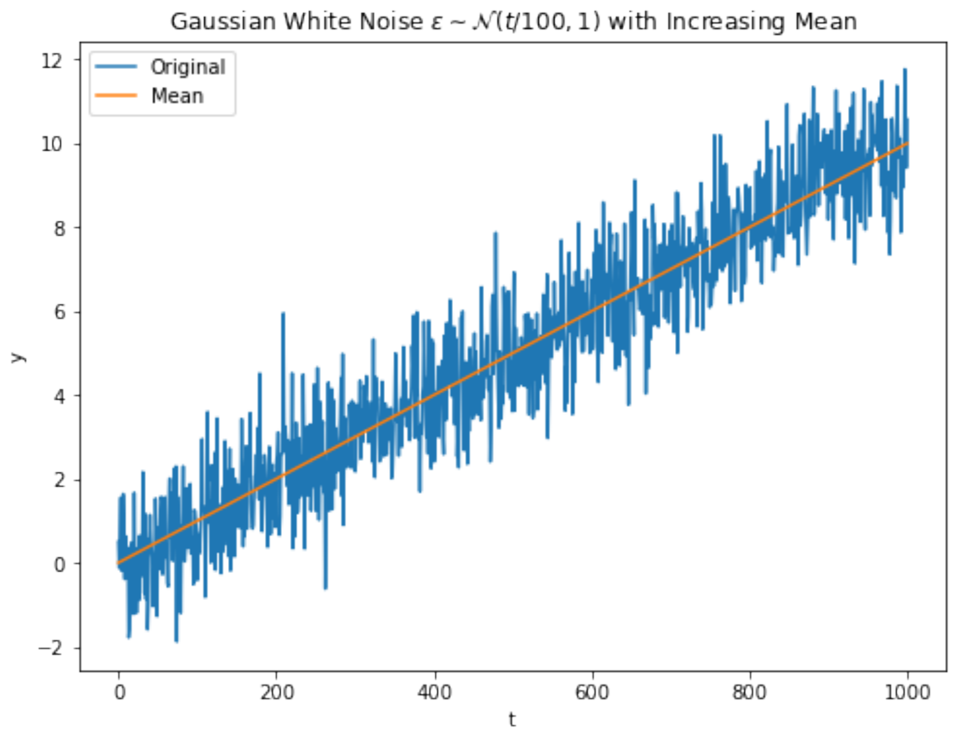


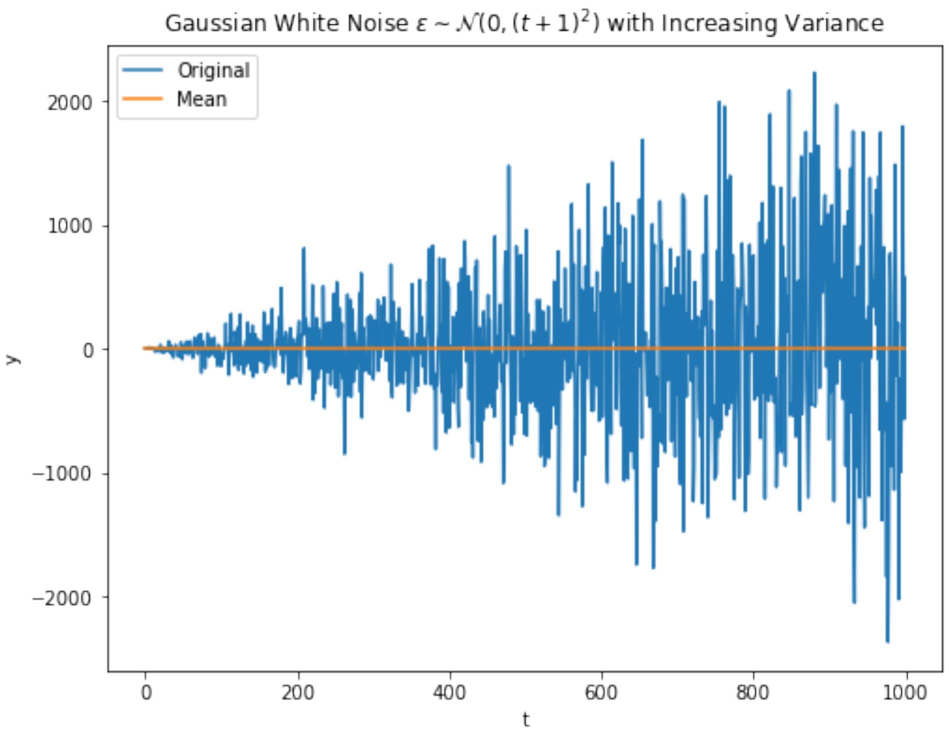
https://commons.wikimedia.org/wiki/File:White_noise.svg



Nonstationary time series

- Left: there is a trend (mean changes over time)
- Right: flat mean but variance changes over time





Prediction for stationary time series

- If a time series is stationary and has zero mean, then prediction is basically sampling from a white noise
- Unfortunately, that basically means y_t is completely random and there's not much signal
- Most real world series are not like that

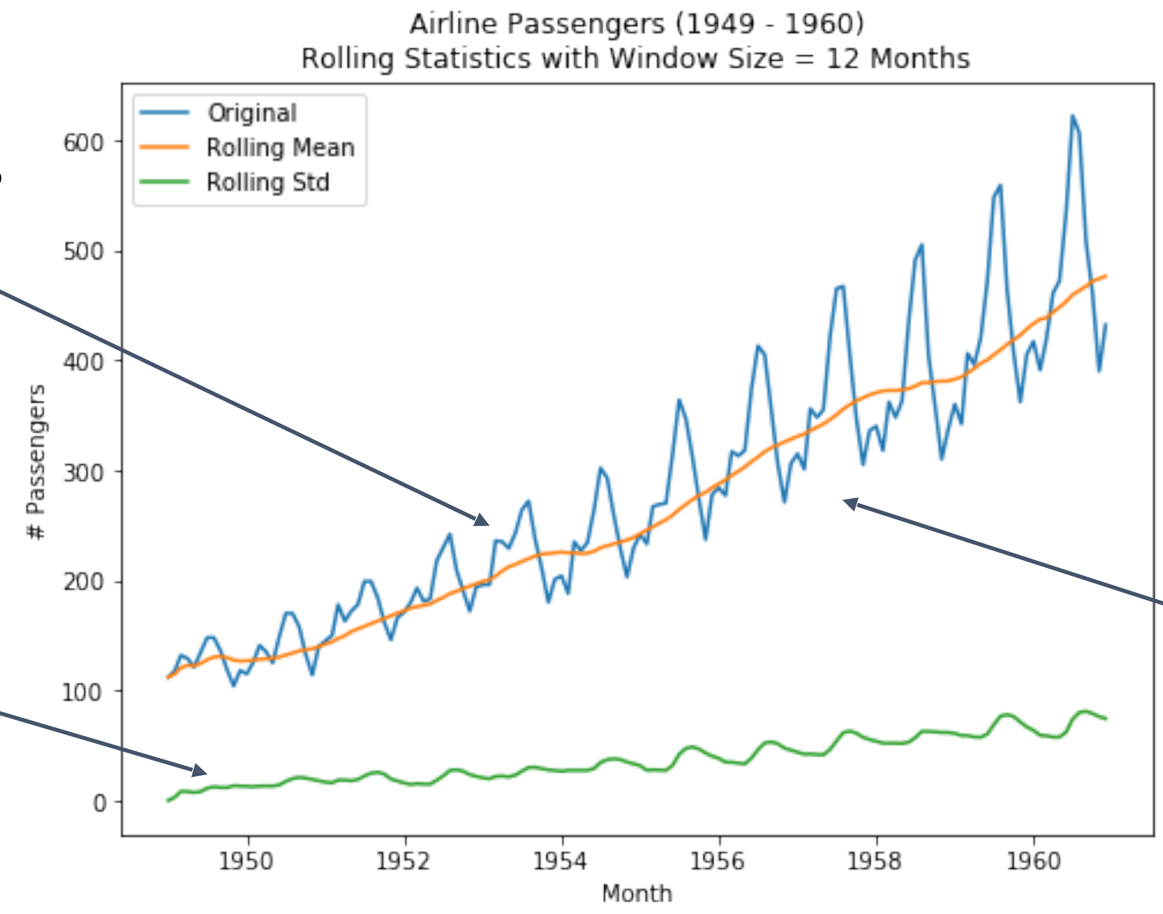
White noise

$$y_t = \boxed{\epsilon_t}$$

Airline passenger is not stationary

Mean increases

Variance increases



Seasonal patterns indicate covariance

Weakly stationary series

- Most real world time series data are not strictly stationary
- But some are weakly stationary
  - Covariance stationary: covariance between nearby values is a constant
  - Trend stationary: mean is a trend line
- Transform nonstationary series into (weakly) stationary series, so that forecasting is easier

Detrending

- Correct the observation y'_t by the empirical mean

$$y'_t = y_t - \boxed{\mu_t}$$

Estimated based on history or aligned series

Standardization

- Compute Z-values of the series
- Commonly used as normalization of vector data
- μ and σ are mean and standard deviation of the series

$$z_i = \frac{y_i - \mu}{\sigma}$$

Differencing

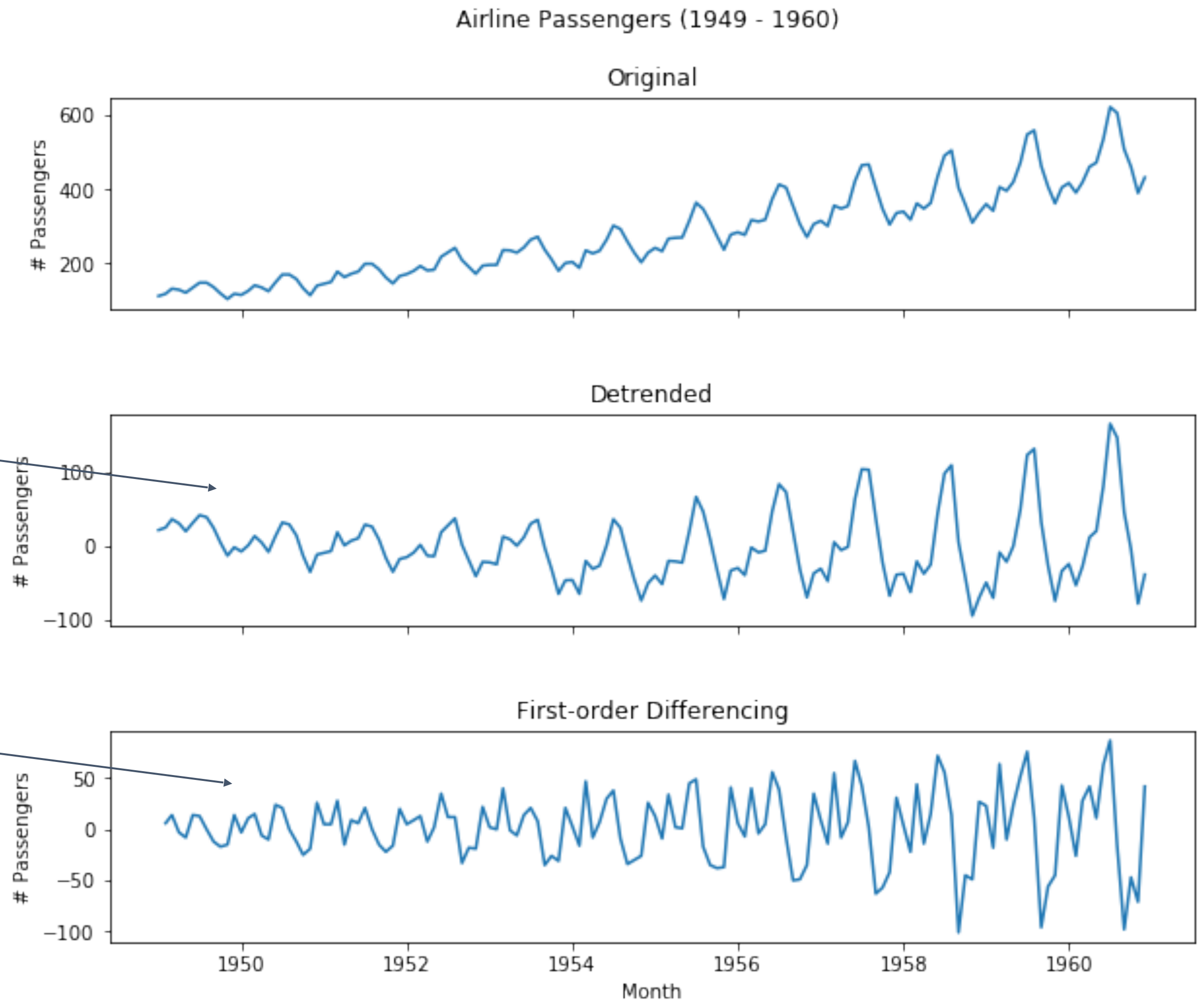
- Using the difference between values may remove level mean
- Note this is considered as the observation at t_j, not t_{j-1} (because we can't look into the future)

$$y'_j = y_j - y_{j-1}$$

Example of detrending and differencing

Detrending

Differencing



Longer Differences

- If *k* is selected properly, computing differences through a larger window can remove seasonal variations.

$$y'_j = y_j - y_{j-k}$$

Return

- In finance, if y is price, $y_j - y_{j-1}$ is the price difference at time t_j.

- (Raw) Return: rate of price difference (assuming $y > 0$)
- Can be interpreted as the profit and loss (P&L) of an investment

$$y'_j = \frac{y_j - y_{j-1}}{y_{j-1}}$$

Log return

- In stock trading, we usually use logarithmic difference.
- Assuming both y_j and y_{j-1} are positive:

$$y_j' = \log\left(y_j\right) - \log\left(y_{j-1}\right)$$

Log return vs. raw return

- Let $$\gamma = \frac{y_j - y_{j-1}}{y_{j-1}}$$
- We have:

$$\log(1+\gamma) = \log\left(\frac{y_j}{y_{j-1}}\right) = \log(y_j) - \log(y_{j-1})$$

- When r << 1 (mostly true in short period trading):

$$\log(1+\gamma) \approx \gamma$$

Why log return?

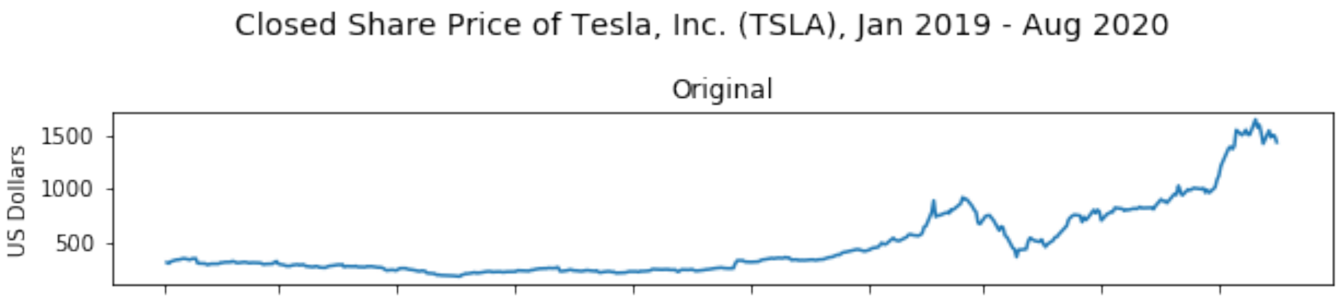
- If y is normally distributed → use raw return
- If y is log-normally distributed → use log return
- Price is usually log-normally distributed
  - Non-negative
  - Right skewed

Why log return?

- Raw return becomes computationally inefficient if you need to aggregate over time periods

- Use additions instead of products

Example



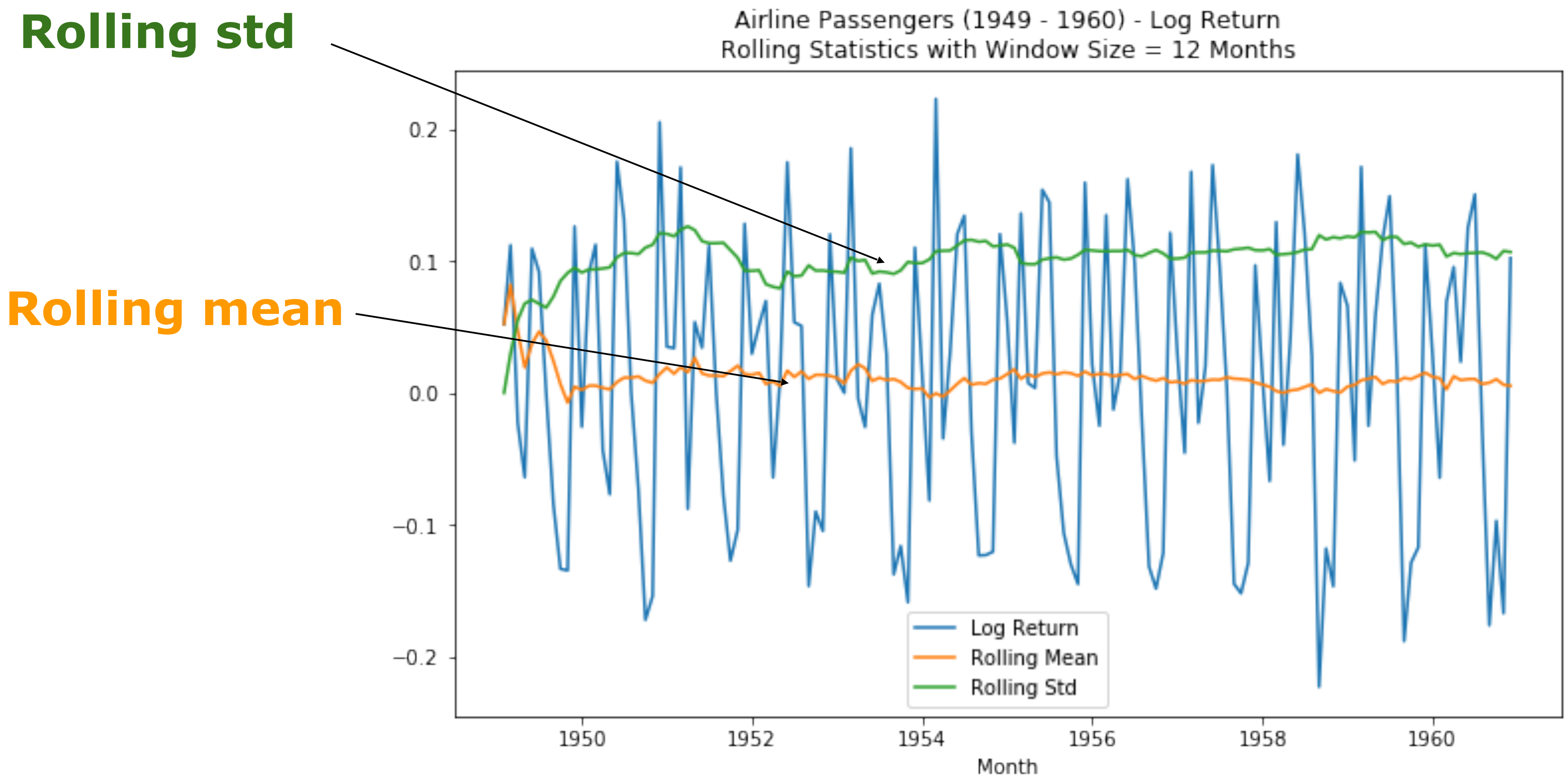
Price Difference

Price Return

Log Return

Generated on trading data from Yahoo! Fi
finance.yahoo.com/quote/TSLA?p=TSLA&.tsrc=fin-srch

Log return, airport passenger



What does difference tell us?

- We hope the difference, $y_j' = y_j - y_{j-1}$ is stationary
- If y' is stationary, we can make a guess of y_t', then add it back to y_{t-1}

$$y_t' = y_t - y_{t-1} = \epsilon_t$$

$$y_t = y_{t-1} + \epsilon_t$$

Higher order differences

- Higher level differences can remove trends

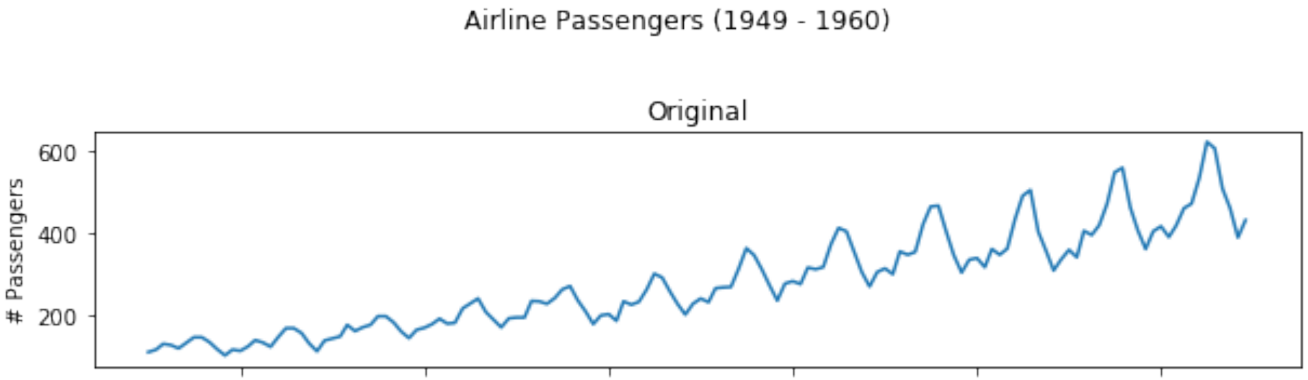
$$y_j'' = y_j' - y_{j-1}'$$
$$= y_j - y_{j-1} - (y_{j-1} - y_{j-2})$$

$$y_{j+1} = y_j + \boxed{c} + e_{j+1}$$

constant (linear)
trend mean

Example

First-order
Differencing



Second-order
differencing

Measuring the Predictability

Goodness of fit

- For every t_i in a time series, we have a predicted value $\hat{y}_i$
- y_i is the value we actually observed.
- $y_i - \hat{y}_i$ is called the <u>residual</u> (observed error)
- Residual Sum of Squares (RSS, or SS_{RES}):

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Root Mean Square Error (RMSE)

- Expectation of square error, then take the square root
- Interpretable (comparable to *y*)
- Usually computed based on hold-out test samples

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{n}}$$

R-squared

- RSS and RMSE are comparable on the same time series!
- Comparing RSS and RMSE for different prediction targets can be misleading (because one target can be intrinsically harder to predict)
- R^2 (R-Squared): explained variance / total variance

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2}$$

More on R-squared

- R-squared is between 0 and 1
- Good to describe the general predictive power of the model
- But hard to interpret the practical value of prediction
- Doesn’t describe the predictive power of individual features

Autocorrelation

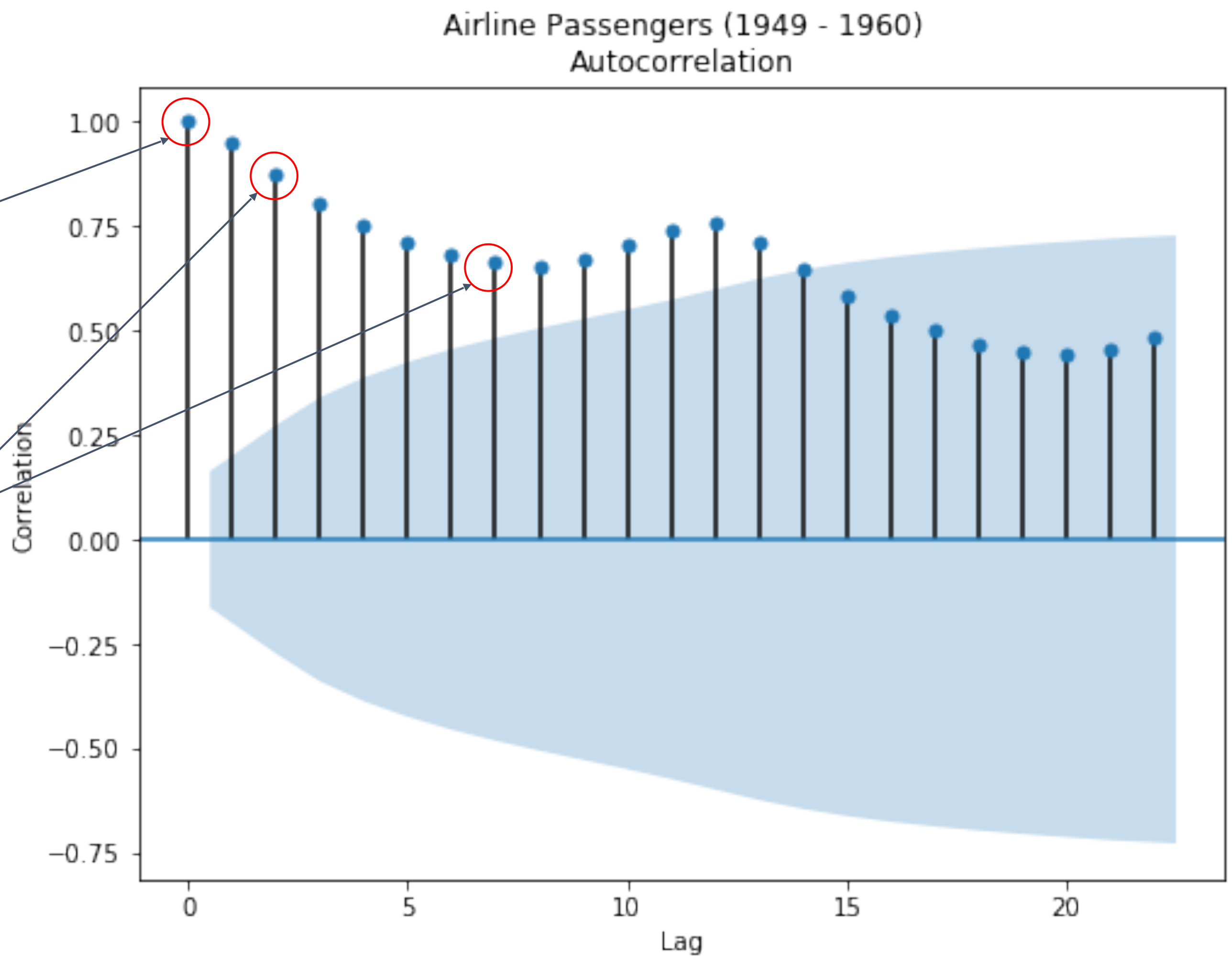
- Autocorrelation of a time series computes correlation over different lags: y_t and $y_{t\text{-}k}$

$$Autocorrelation(k) = \frac{Covariance_t\left(y_{t-k}, y_t\right)}{Variance_t\left(y_{t-k}\right)}$$

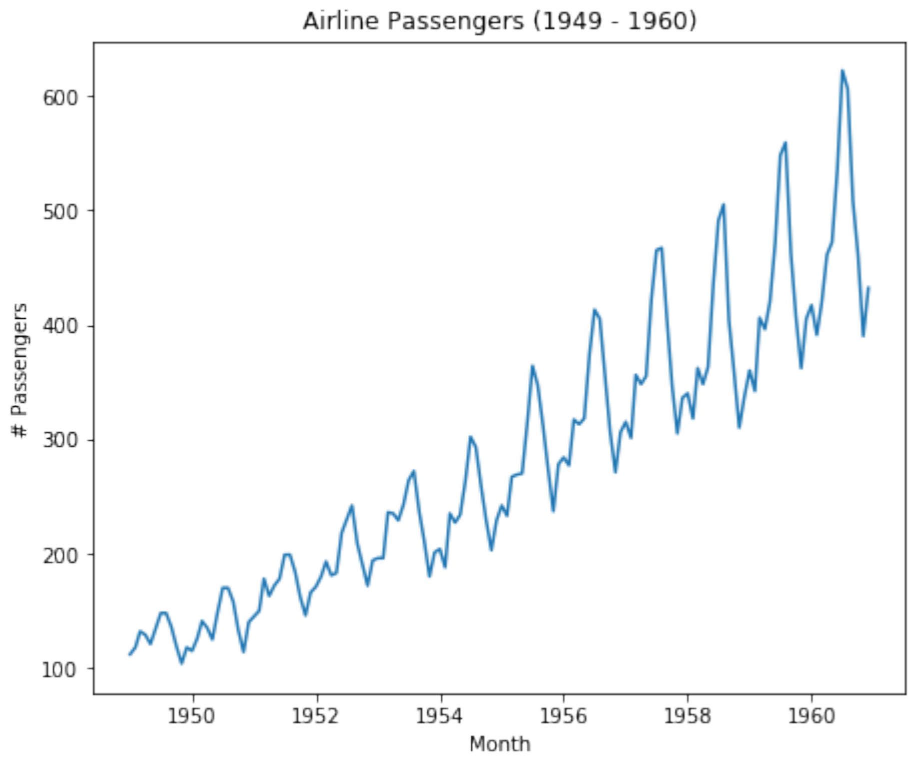
Autocorrelation function (ACF) plot

autocorrelation to itself: always 1

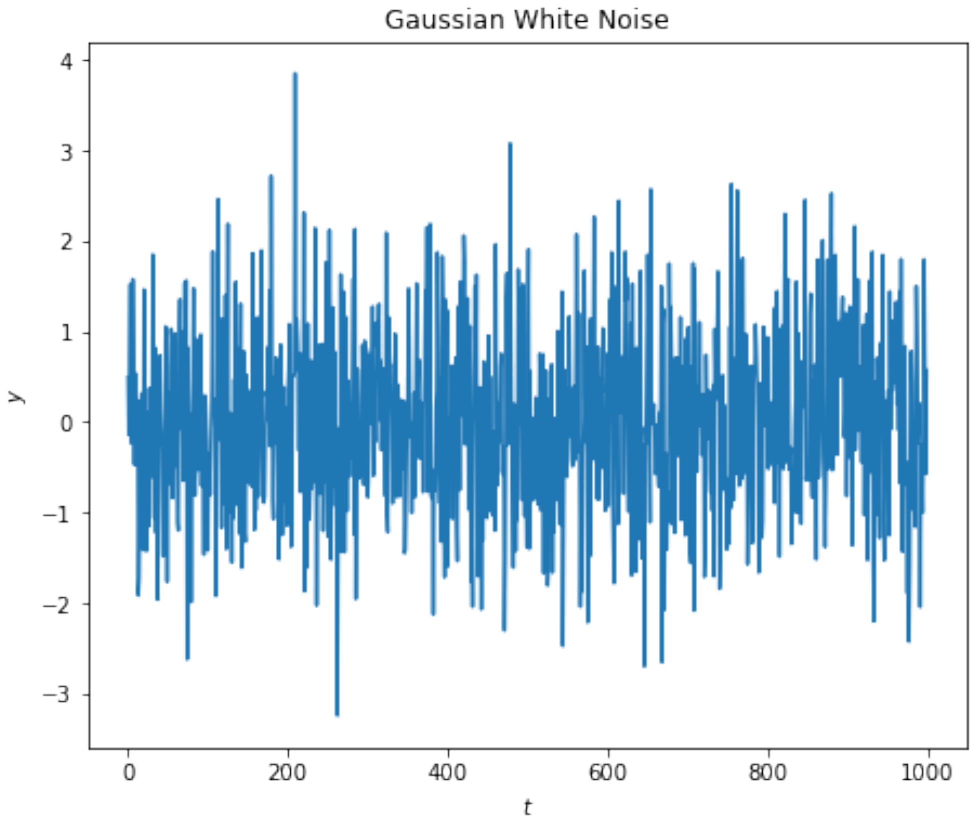
autocorrelation to lagged observations



ACF of the airline passenger series



ACF of the white noise series



Partial autocorrelation function plot

- If y_{t-1} is correlated with y_t, then y_{t-2} is likely too (because it is correlated with y_{t-1}).

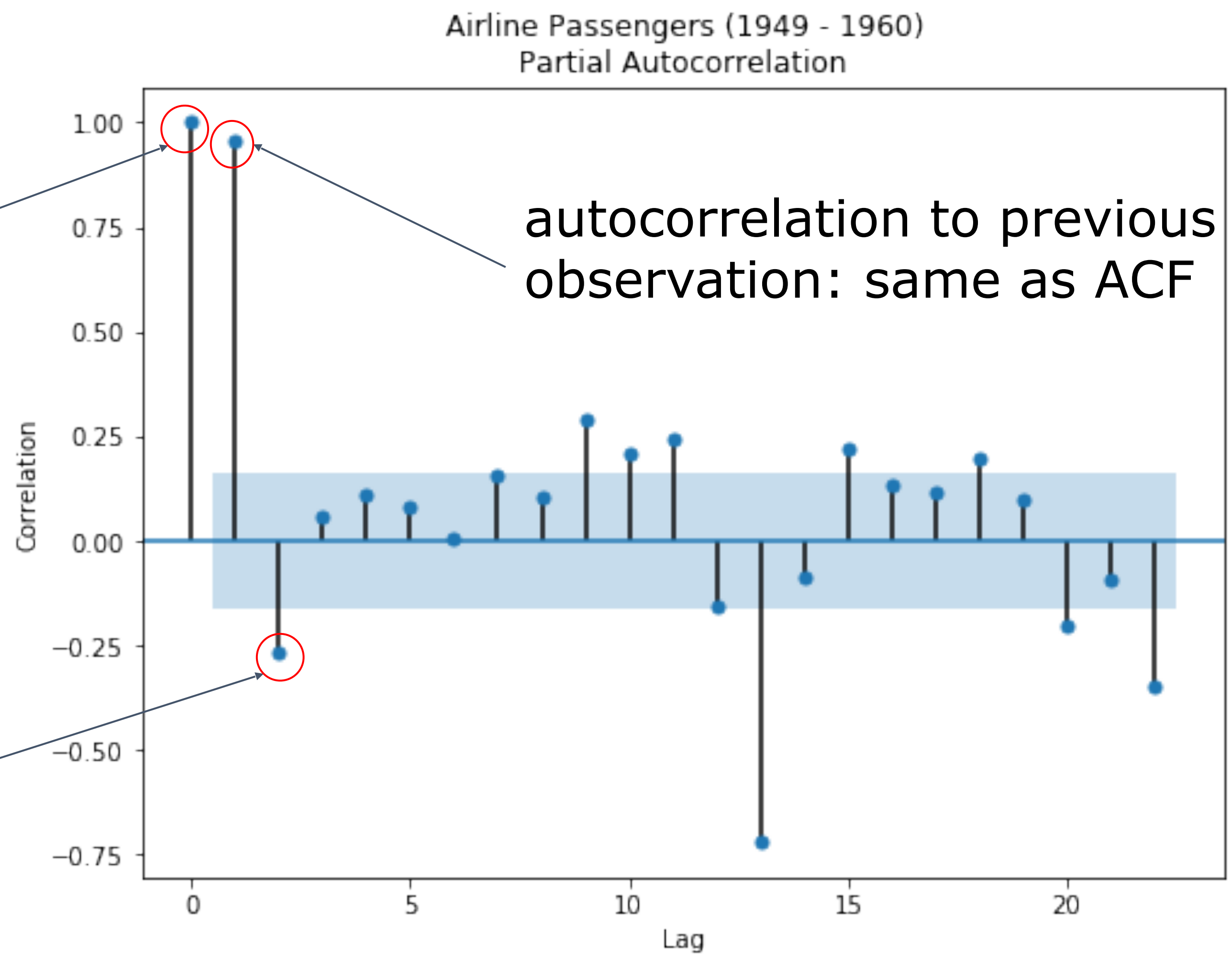
- Partial autocorrelation: remove the effect of a correlation with a longer lag (e.g., y_{t-i}) due to the observations with shorter lags (e.g., y_{t-i+1} … y_{t-1}).

PACF plot

autocorrelation to itself: still 1

autocorrelation to previous observation: same as ACF

given y_{t-1}, autocorrelation to y_{t-2} becomes smaller



What does autocorrelation tell us?

- $y_{t\text{-}i}$ may have a prediction power for y_t
- We may build a prediction model that uses $y_{t\text{-}i}$ to predict y_t
- There are multiple $y_{t\text{-}i}$ (with different i) that are more or less predictive
- Does this remind you of n-gram language models?
- Both y_t and $y_{t\text{-}i}$ are numerical, so we need to use regression

Autoregressions

Autoregression (AR)

- A regression model that uses multiple y_{t-i} to predict y_t
- AR(p): *p* is a critical parameter - max lag

White noise

$$y_t = \sum_{i=1}^{\rho} \boxed{\Phi}_i \bullet y_{t-i} + \boxed{\mu} + \boxed{\epsilon_t}$$

Regression coefficient of y_{t-i}, can be interpreted as the effect of a value at lag i on y_t

Intercept: constant, explaining linear trend

How does AR work?

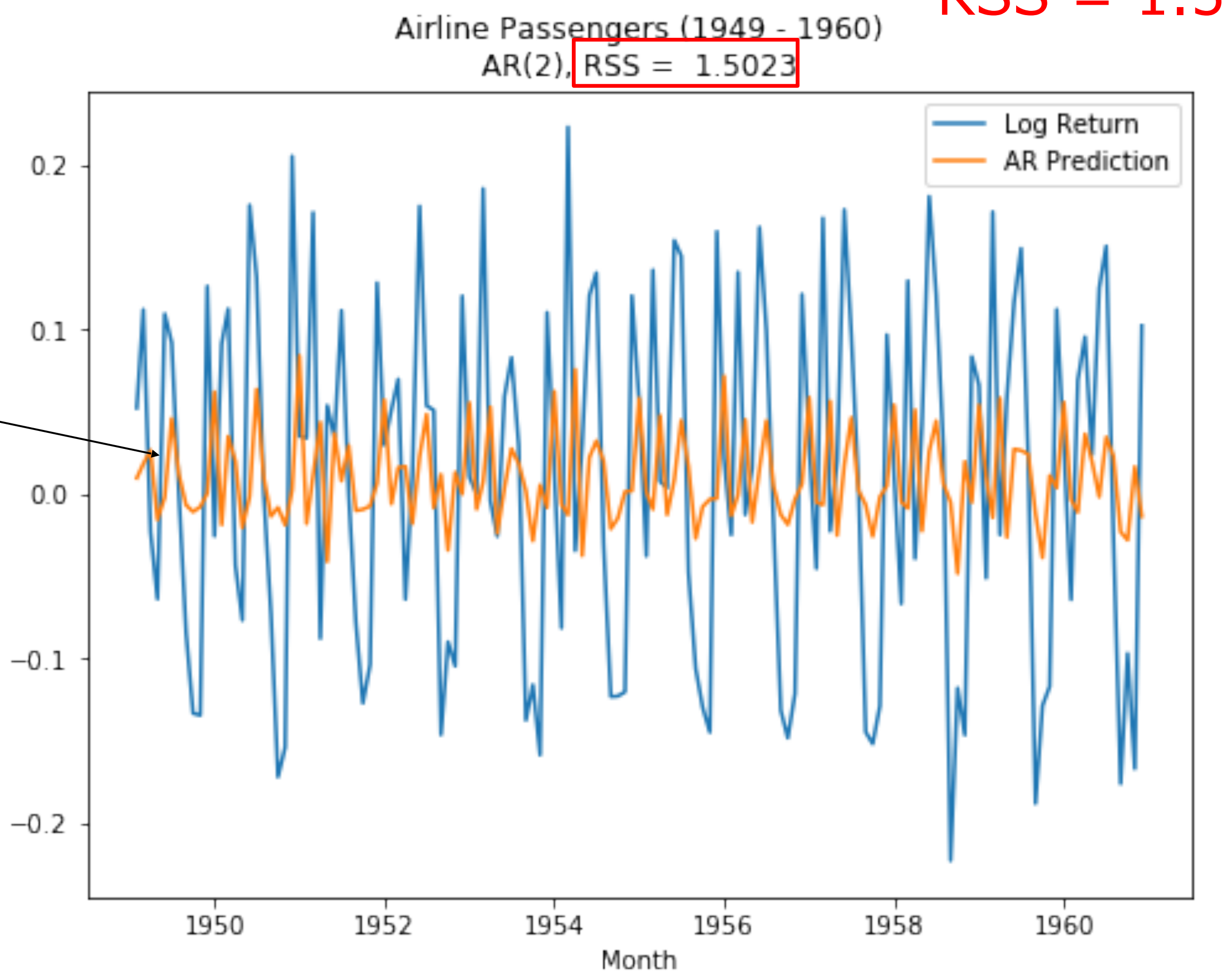
- Training: Learn μ, ϕ_1, …, ϕ_p from training data
- Usually through *least squares regression*
- Testing: for an out-sample y_t, make a prediction using

$$\hat{y}_t = \sum_{i=1}^{\rho} \Phi_i \bullet y_{t-i} + \mu$$

Example of AR

RSS = 1.5023

Predictions preserve the patterns of the series and explain part of the variance



Problem of AR

- AR does not explain all the variation
- E.g., it does not consider the impact of outliers (unexpected shocks) on future values
- What would help cascade the impact of shocks?
- Moving average

Moving average model

- Suppose all y_t are sampled from white noise with a flat mean, then after (weighted) moving average, we have:

max lag of moving average

$$y_t = \mu + \epsilon_t + \sum_{i=1}^{\boxed{q}} \boxed{\theta_i} \bullet \boxed{\epsilon_{t-i}}$$

Weights of moving average

White noise at time (t-i)

Moving average model

- MA(q): q is a critical parameter, max window size
- Comparing to moving average smoothing, EMA: the weights $\theta_1,\ldots,\theta_q$ are estimated through regression instead of predefined

Moving average model

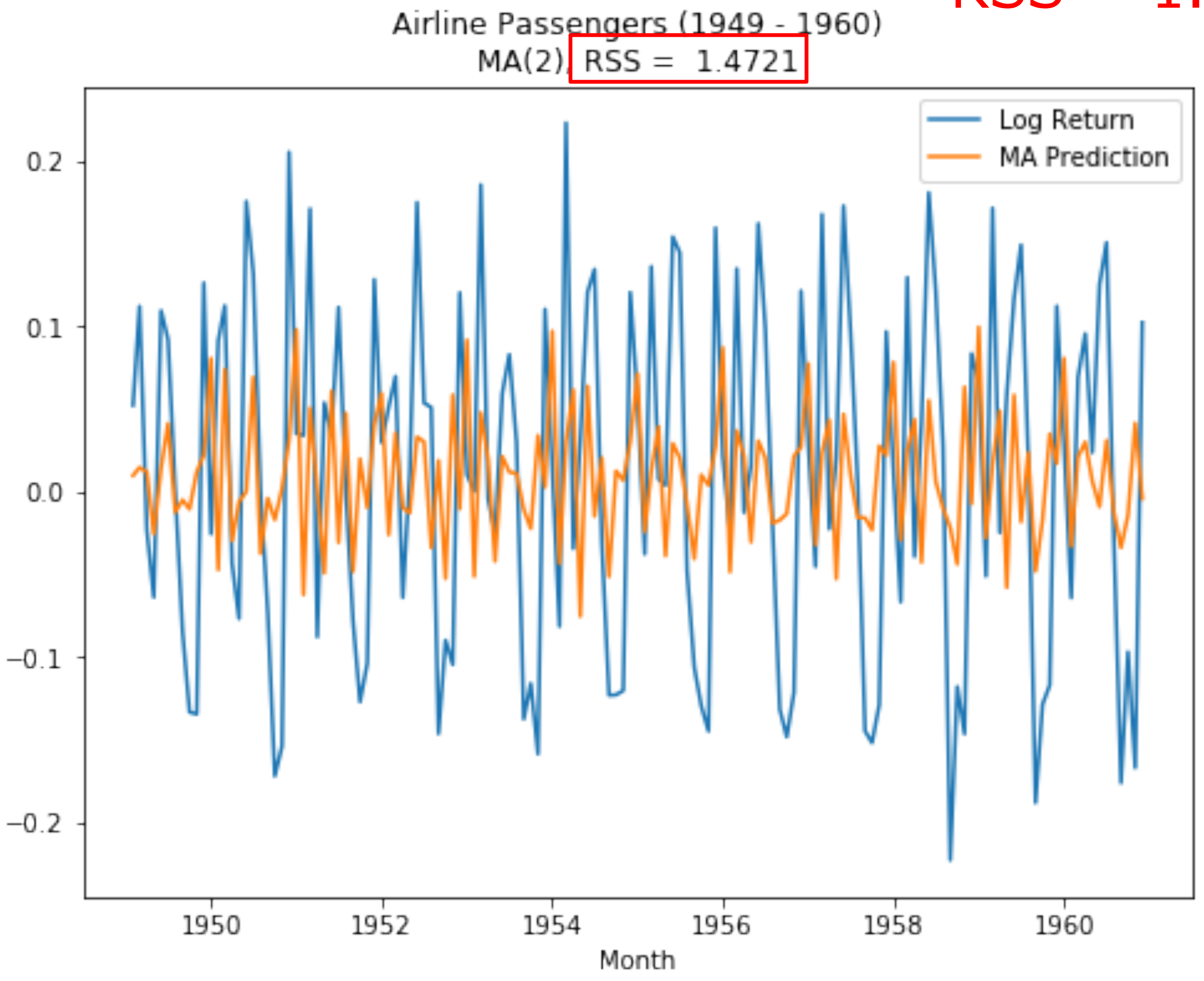
- MA(q) is always stationary, because it’s essentially a weighted sum of white noises

- But the error terms (ε) are not observable

- This makes the estimation of parameters harder than AR

Example of MA

RSS = 1.4721



Autoregressive moving averages

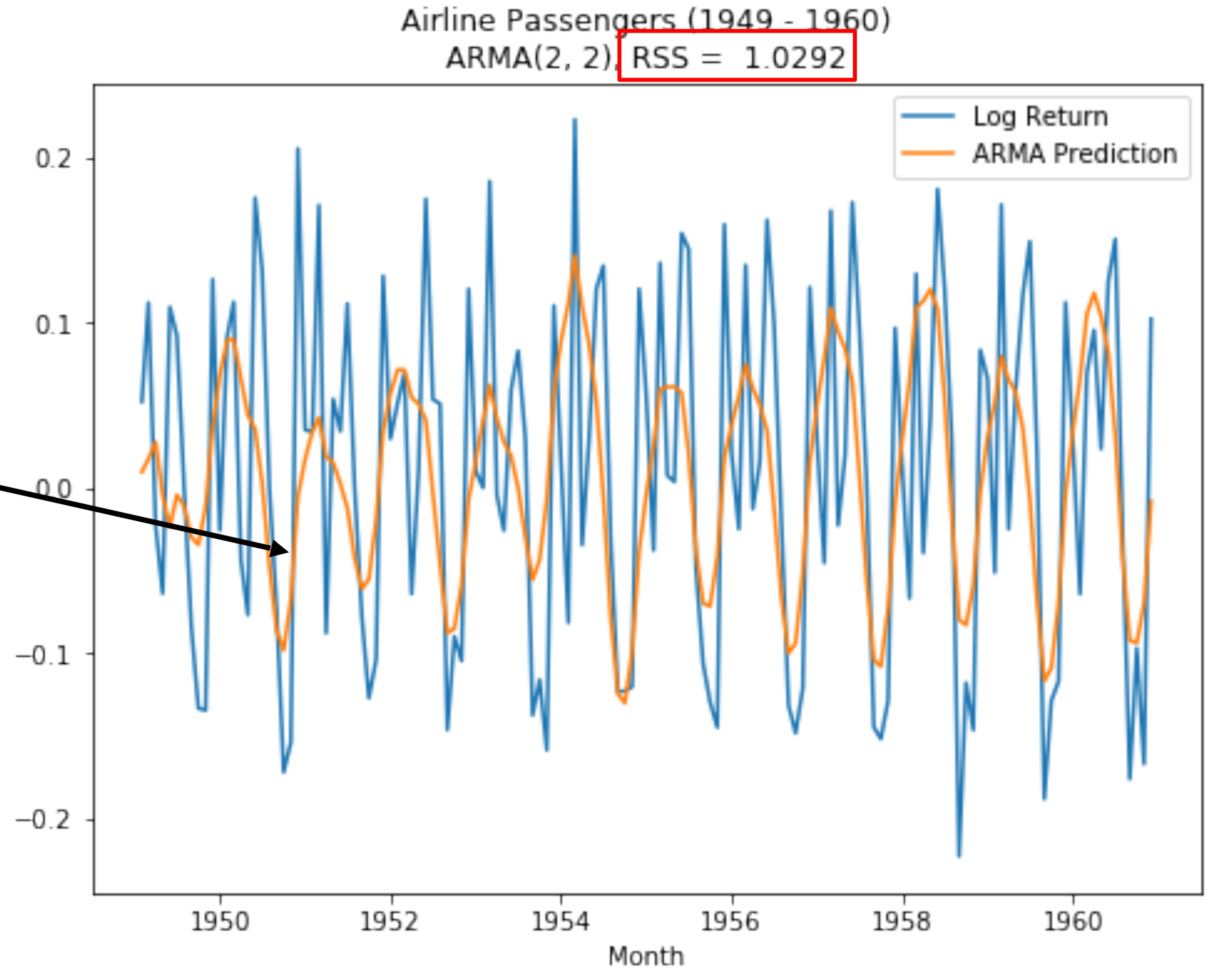
- Autoregressive moving average (ARMA)
- Combining autoregression (AR) and moving average (MA)
- ARMA(p, q)

$$y_t = \underbrace{\boxed{\sum_{i=1}^{p} \phi_i \bullet y_{t-i}}}_{\text{Autoregression}} + \mu + \epsilon_t + \underbrace{\boxed{\sum_{i=1}^{q} \theta_i \bullet \epsilon_{t-i}}}_{\text{Moving average}}$$

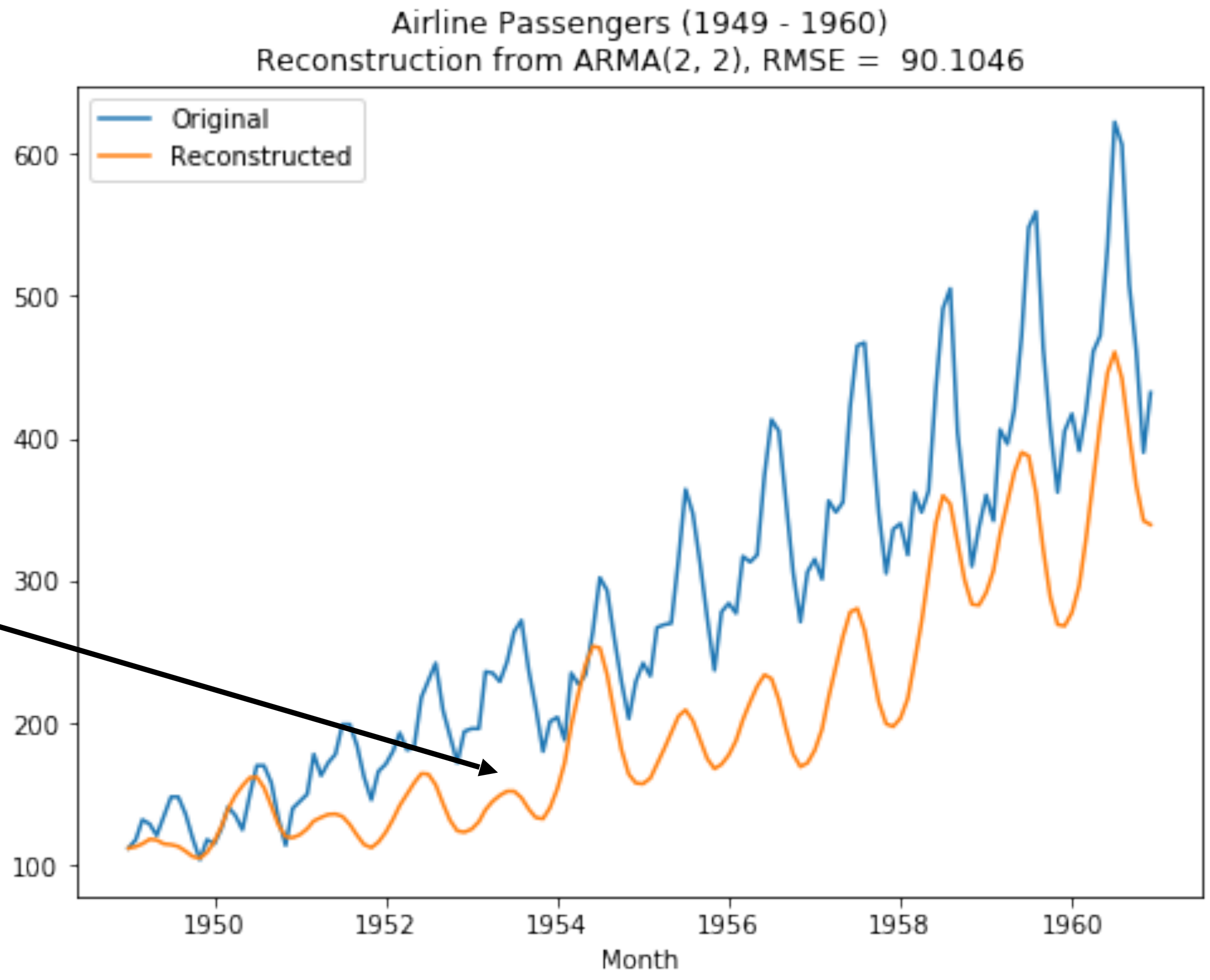
Example of ARMA

RSS = 1.0292: a much better fit than AR or MA

Smoothed predictions that preserve the patterns



Transform it back to original



Transform from log return back to raw y

Problem with ARMA

- Either AR or MA works on stationary time series (in theory)
- When time series are not stationary?
- Use differencing to make them (more) stationary

ARIMA

ARIMA

- Autoregressive Integrated Moving Average
- AR: autoregression means new observations depend on lagged observations
- I: use of differencing to make time series (more) stationary
- MA: residuals of previous observations propagate over time

ARIMA

- ARIMA(p, d, q)
- *p* (lag order): maximum lags for autoregression (whether the observation would still)
- *d* (difference order):
  - 0: raw value;
  - 1: $y' = y_t - y_{t-1}$;
  - 2: $y'' = y_t' - y_{t-1}'$
- *q* (order of MA): window size of moving average

$$\boxed{y_t'} = \sum_{i=1}^{p} \phi_i \bullet \boxed{y_{t-1}'} + \mu + \epsilon_t + \sum_{i=1}^{q} \theta_i \bullet \epsilon_{t-i}$$

Differencing

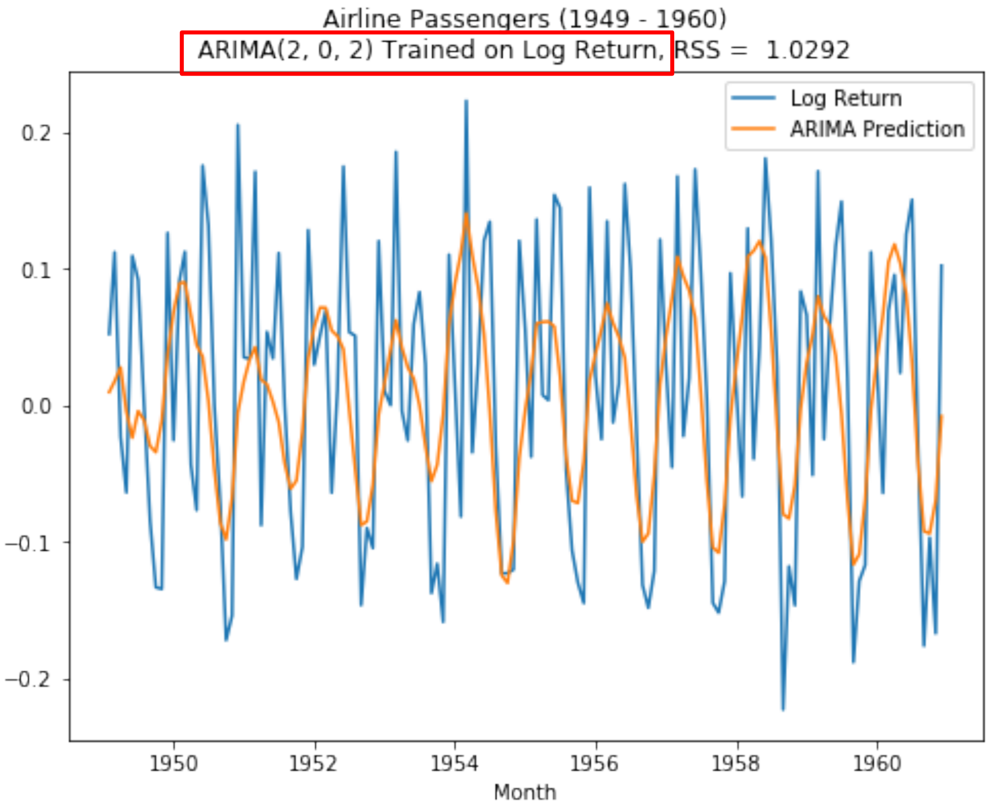
ARIMA integrates other models

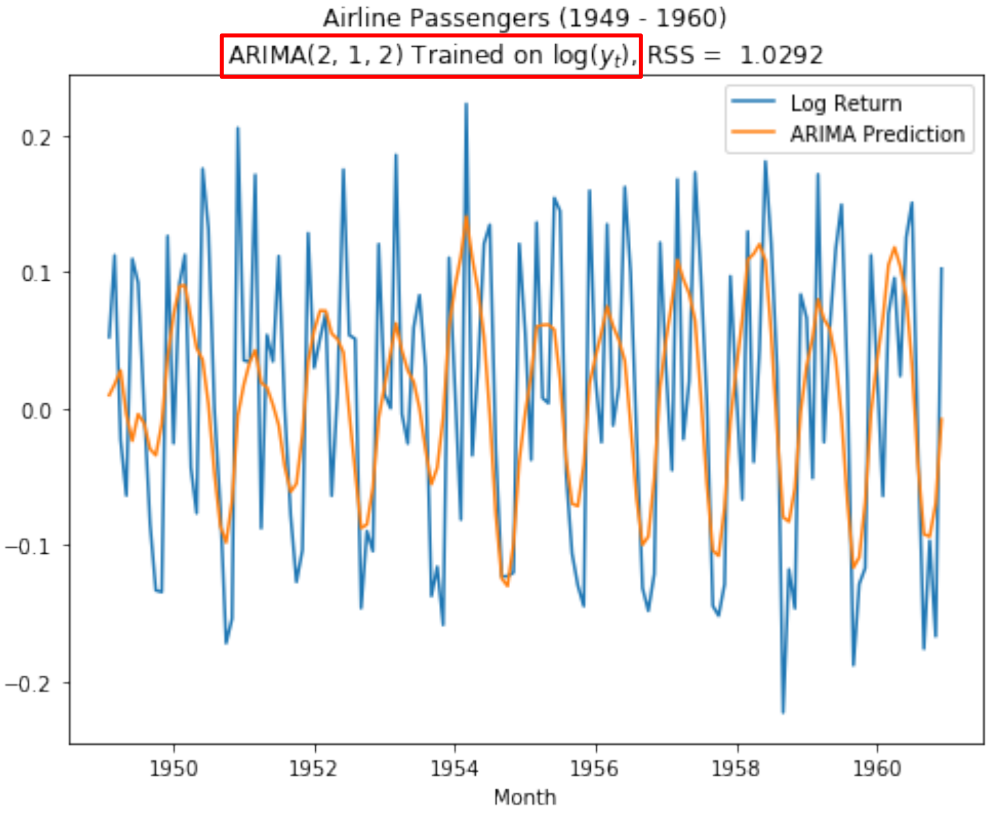
- ARIMA(p, d, q)

- AR is essentially ARIMA(p, 0, 0)
- MA is essentially ARIMA(0, 0, q)
- ARMA is essentially ARIMA(p, 0, q)

Example of ARIMA

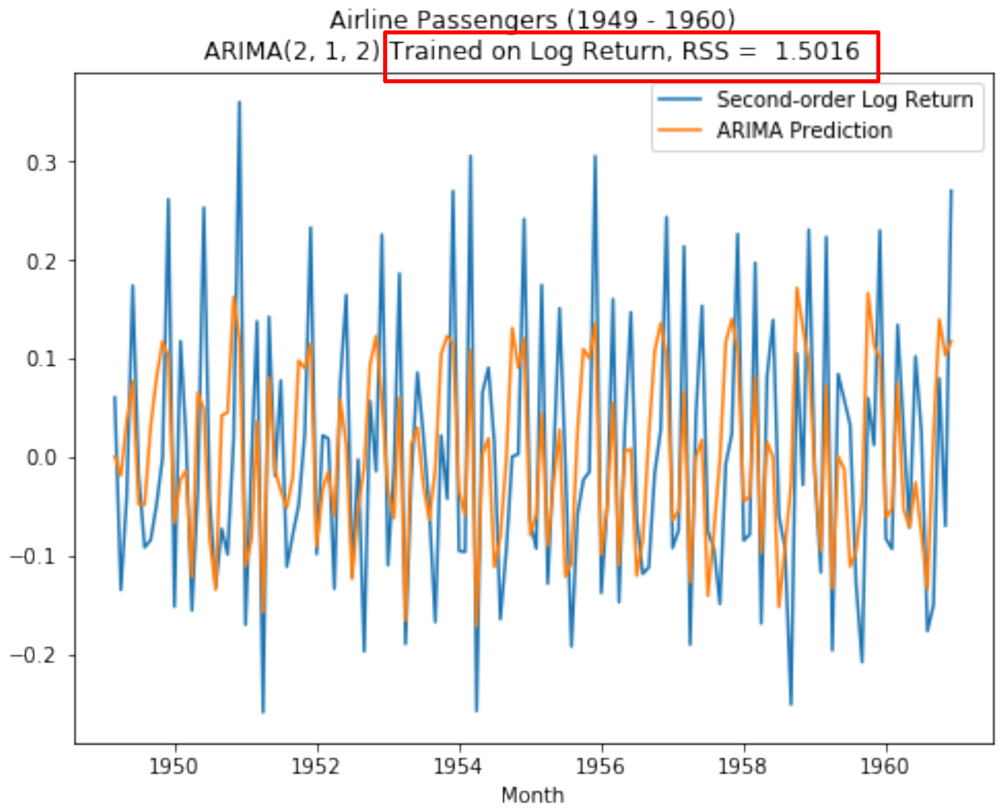
Verification: ARIMA(*, 1, *) on raw y is the same as ARIMA(*, 0, *) on differenced y'

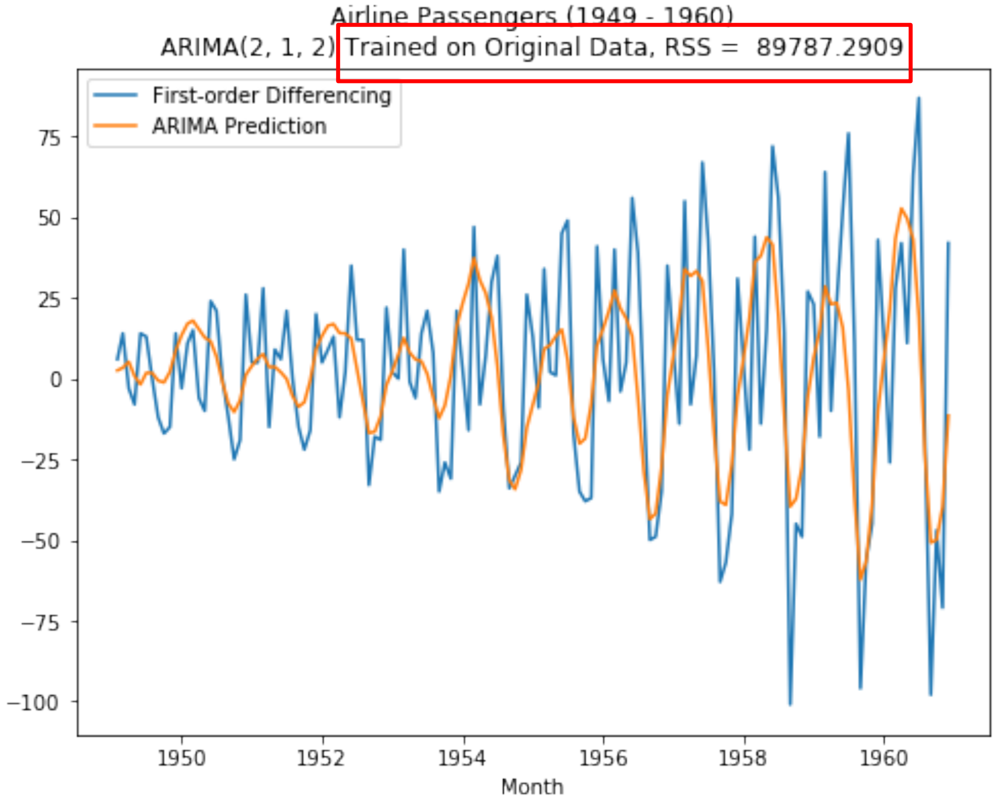




Example of ARIMA

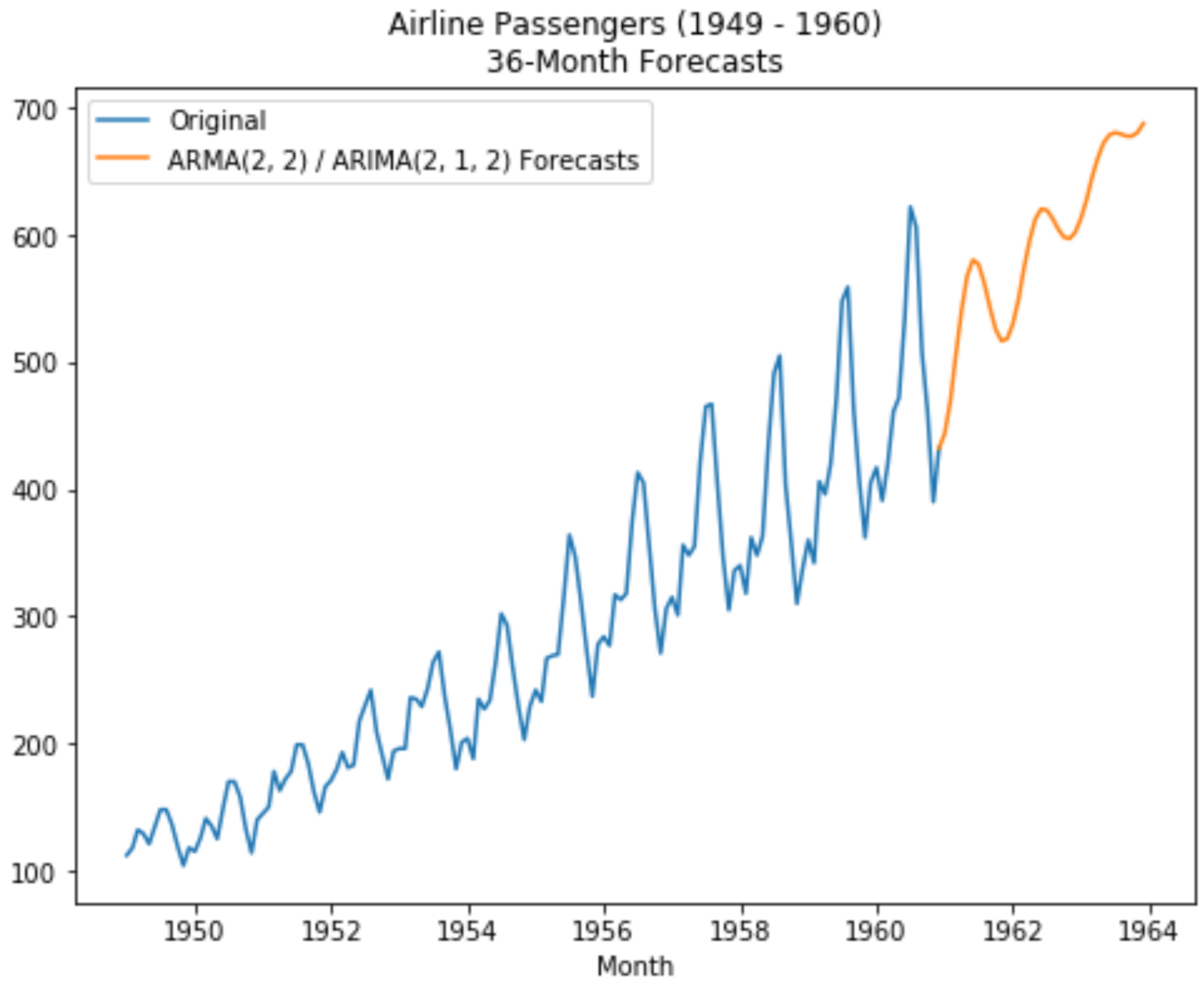
ARIMA works on stationarized series and original series; RSS is not comparable for different targets





Making forecasts

- Train the autoregression model with observed series
- Generate prediction of y_{n+1}: $\hat{y}_{n+1}$
- Use $\hat{y}_{n+1}$ (with previous observations) to predict y_{n+2}: $\hat{y}_{n+2}$
- Use $\hat{y}_{n+2}$ to generate $\hat{y}_{n+3}$ …



Summary

- Basic assumption: the observation y_t depends on the observations in the previous window
- AR assumes linear regression
- MA assumes smoothness of noise
- They can be combined, and then combined with differencing
- In more advanced models, regressions could consider complex patterns in the previous window (or even the entire history)
- In reality, often use more powerful machine learning predictors

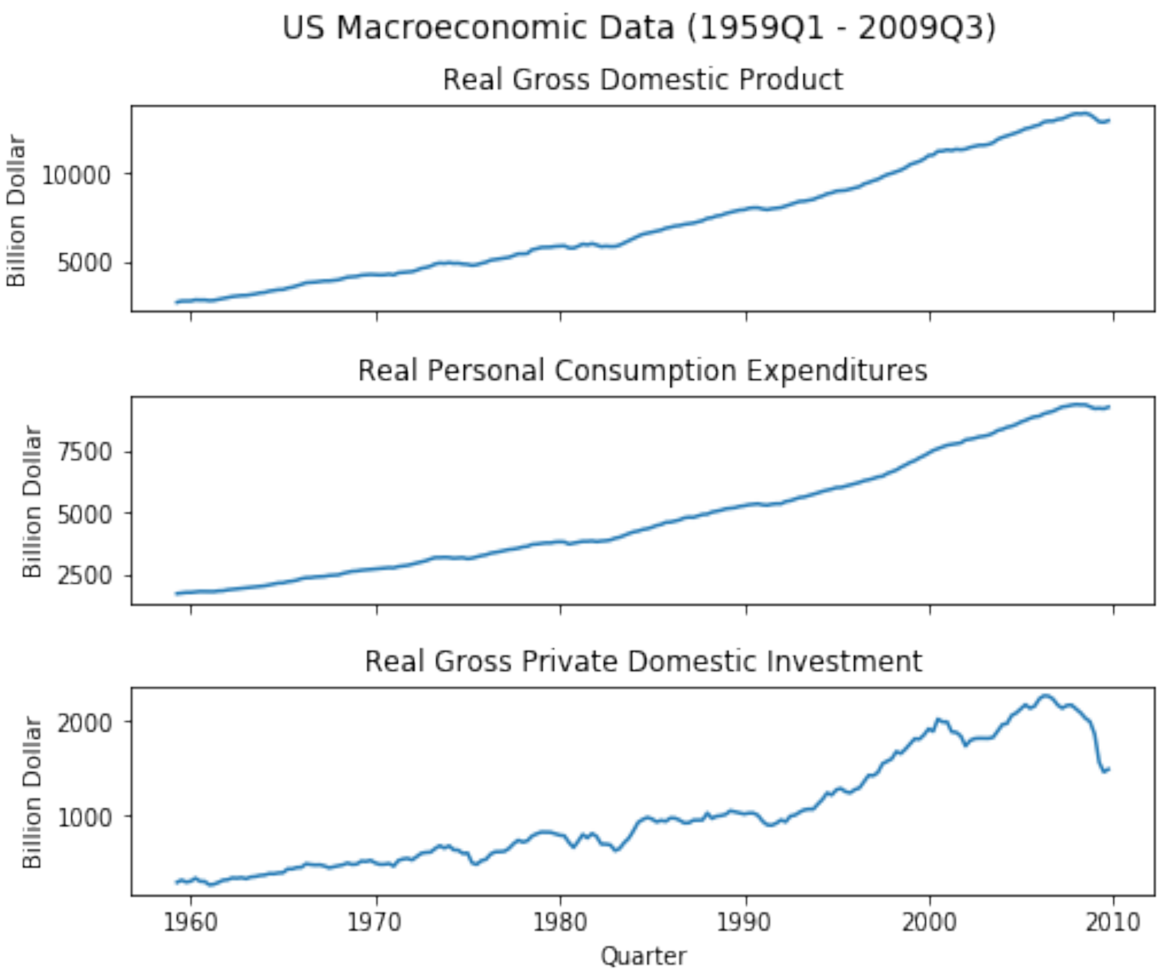
Vector Autoregression

Prediction of multivariate series

- In many cases we have multiple aligned time series
- Measurements of different variables
- We want to use one to make prediction for another

Example

- GDP
- Expenditures
- Domestic Investment



Vector autoregression (VAR)

- Input: multiple aligned time series (multi-dimensional series)
- Predict the future observation of <u>one</u> series using the past observations of <u>all</u> series
- Example: two series X, Y, VAR(p)

$$y_t = \boxed{\sum_{i=1}^{p} \phi_{i,\,1} \bullet y_{t-i}} + \boxed{\sum_{i=1}^{p} \varphi_{i,\,1} \bullet x_{t-i}} + \mu_1 + \epsilon_{t,\,1}$$

Autoregression within series

Autoregression with the other series

Vector autoregression (VAR)

- Input: multiple aligned time series (multi-dimensional series)
- Predict the future observation of <u>one</u> series using the past observations of <u>all</u> series
- Example: two series X, Y, VAR(p)

$$y_t = \sum_{i=1}^{p} \boxed{\phi_{i,\,1}} \bullet y_{t-i} + \sum_{i=1}^{p} \boxed{\varphi_{i,\,1}} \bullet x_{t-i} + \boxed{\mu_1} + \boxed{\epsilon_{t,\,1}}$$

One set of parameters
for prediction Y

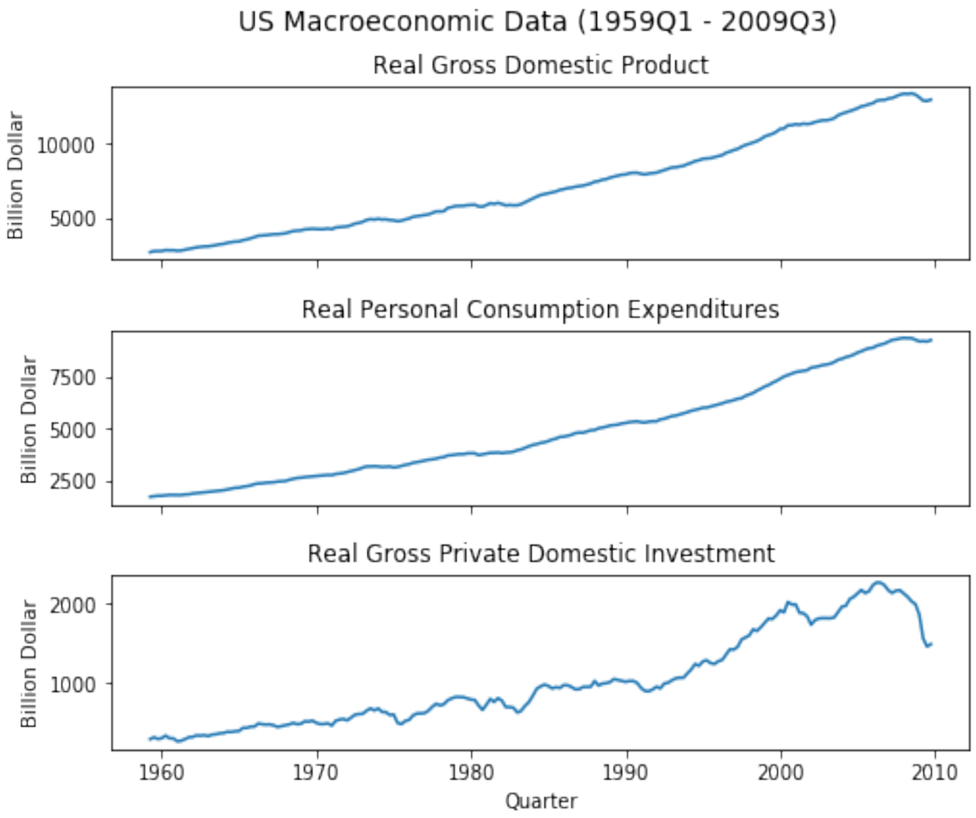
Vector autoregression (VAR)

- Input: multiple aligned time series (multi-dimensional series)
- Predict the future observation of <u>one</u> series using the past observations of <u>all</u> series
- Example: two series X, Y, VAR(p)

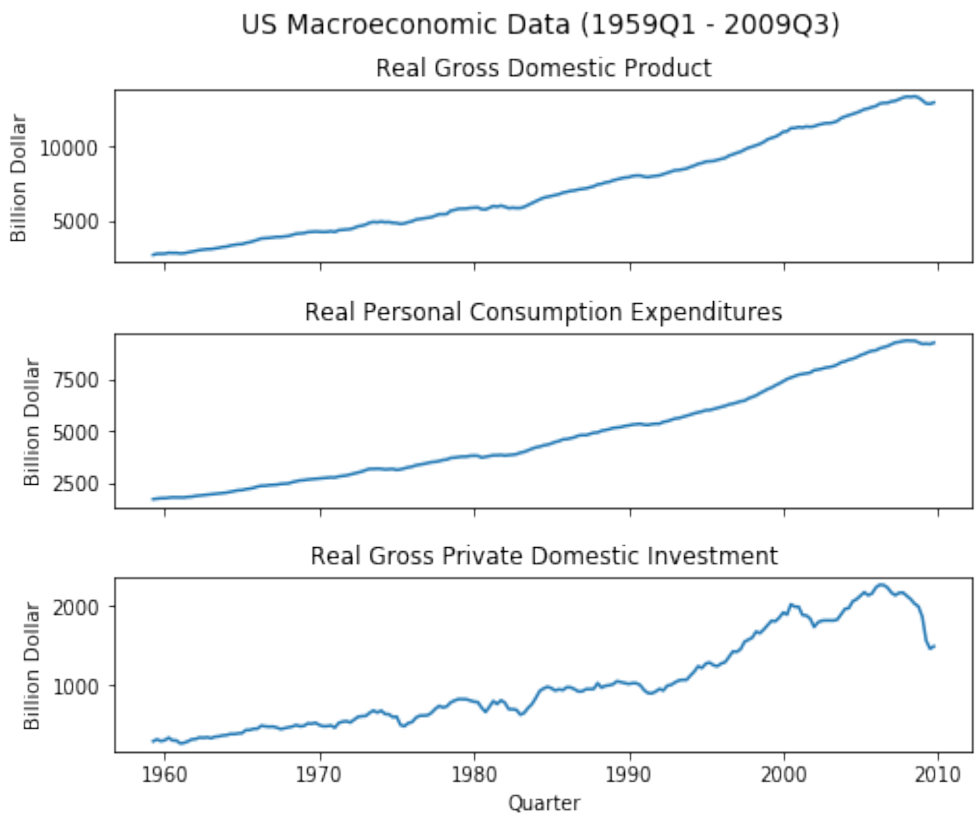
$$y_t = \sum_{i=1}^{p} \phi_{i,1} \bullet y_{t-i} + \sum_{i=1}^{p} \varphi_{i,1} \bullet x_{t-i} + \mu_1 + \epsilon_{t,1}$$

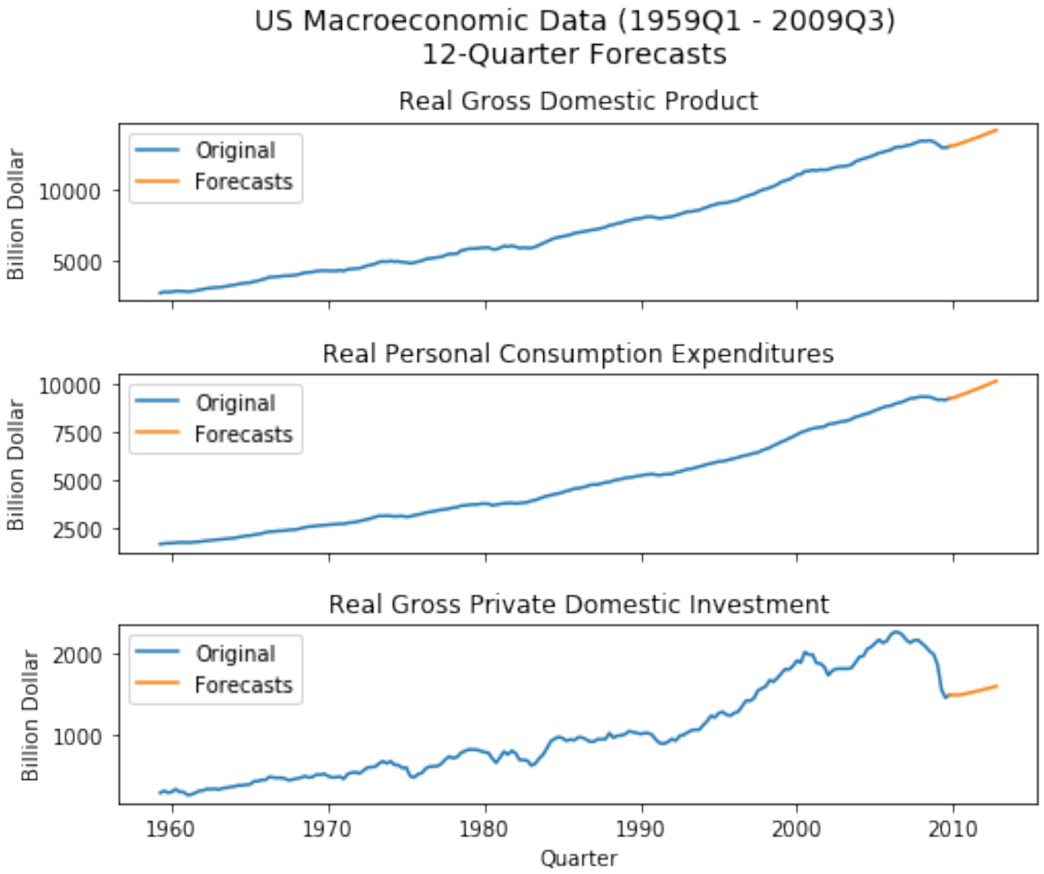
$$x_t = \boxed{\sum_{i=1}^{p} \phi_{i,2} \bullet y_{t-i}} + \boxed{\sum_{i=1}^{p} \varphi_{i,2} \bullet x_{t-i}} + \boxed{\mu_2} + \boxed{\epsilon_{t,2}}$$

Examples of VAR (ACF)



Examples of VAR (forecasts)





Extending VAR

- Easily generalizable to K dimensions
- Same idea also applies to MA, ARMA, ARIMA
- VMA
- VARMA
- VARIMA
- …

Granger Causality

Limitation of VAR

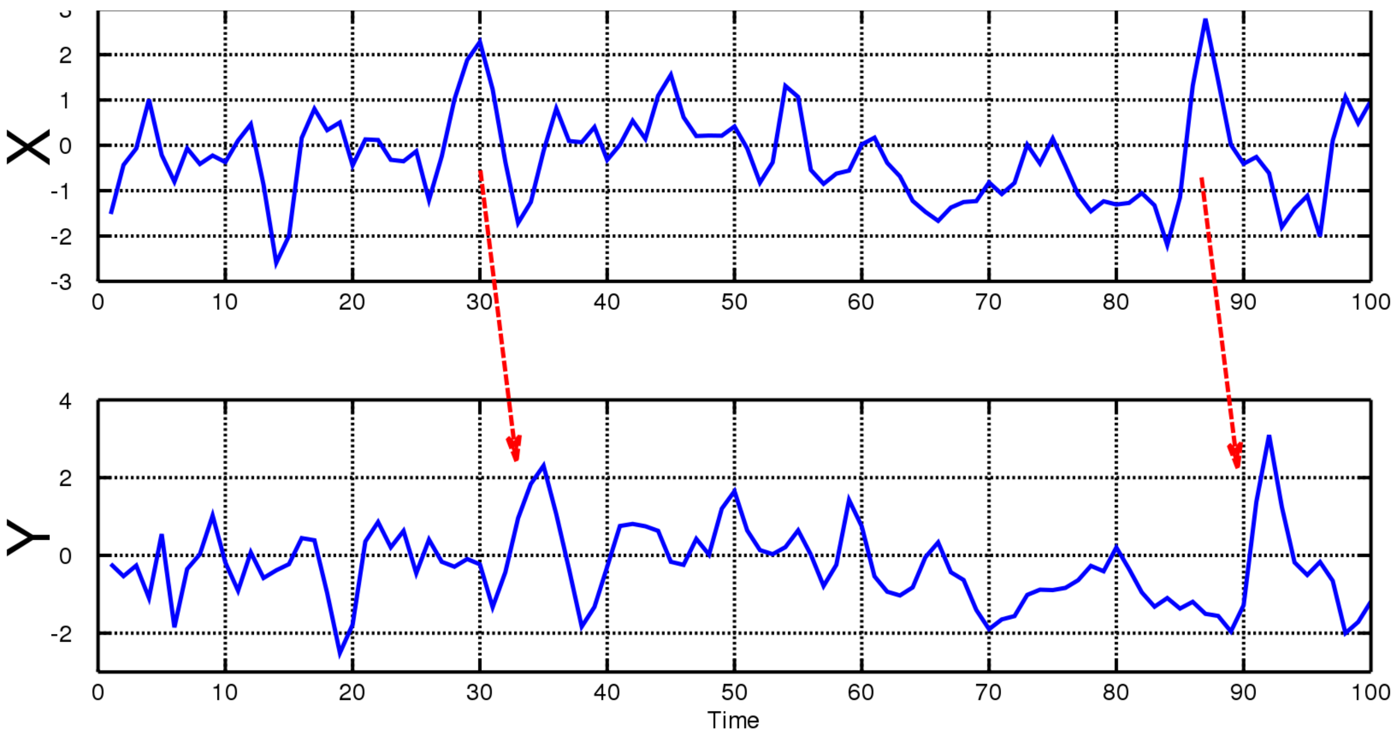
- VAR doesn’t tell us whether X is driving Y or the other way around
- All dimensions are treated equally
- In real world applications, we usually wanted to figure out the direction (news → market, or market → news)
- This is a “mild” notion of causality

Granger Causality

- When multiple series are present, we are usually interested in which one drives which one.
- Correlation can't answer this question: if X is correlated with Y, then Y is correlated with X.
- Test for causal relationship: if X is causal to Y, Y isn't necessarily causal to X.
- In time series, this is usually tested with Granger Causality

Granger Causality

- X "Granger" causes Y



https://en.wikipedia.org/wiki/Granger_causality#/media/File:GrangerCausalityIllustration.svg

Granger Causality Test

- Null hypothesis: X does <u>not</u> "Granger" cause Y
- Consider two AR models:

Predictive power of X "conditional on" Y's own history.

1 $$y_t = \mu + \sum_{i=1}^{p} \phi_i \bullet y_{t-i} + \epsilon_t$$

2 $$y_t = \mu + \sum_{i=1}^{p} \phi_i \bullet y_{t-i} + \boxed{\sum_{j=1}^{m} \varphi_j \bullet x_{t-j}} + \epsilon_t$$

- If 2 is significantly "better" than 1, then X adds extra predictive power to Y in addition to Y's own previous observations!

Granger Causality Test

- Null hypothesis: X does <u>not</u> “Granger” cause Y

- Compare two regressions using F-test

- Reject the null hypothesis if F-stats is high

- Use model selection to determine the best parameters (p, m), or try multiple values and watch for robustness

- Be careful: F-test might lose power if too many parameters (multiple-tests)

Granger Causality - Summary

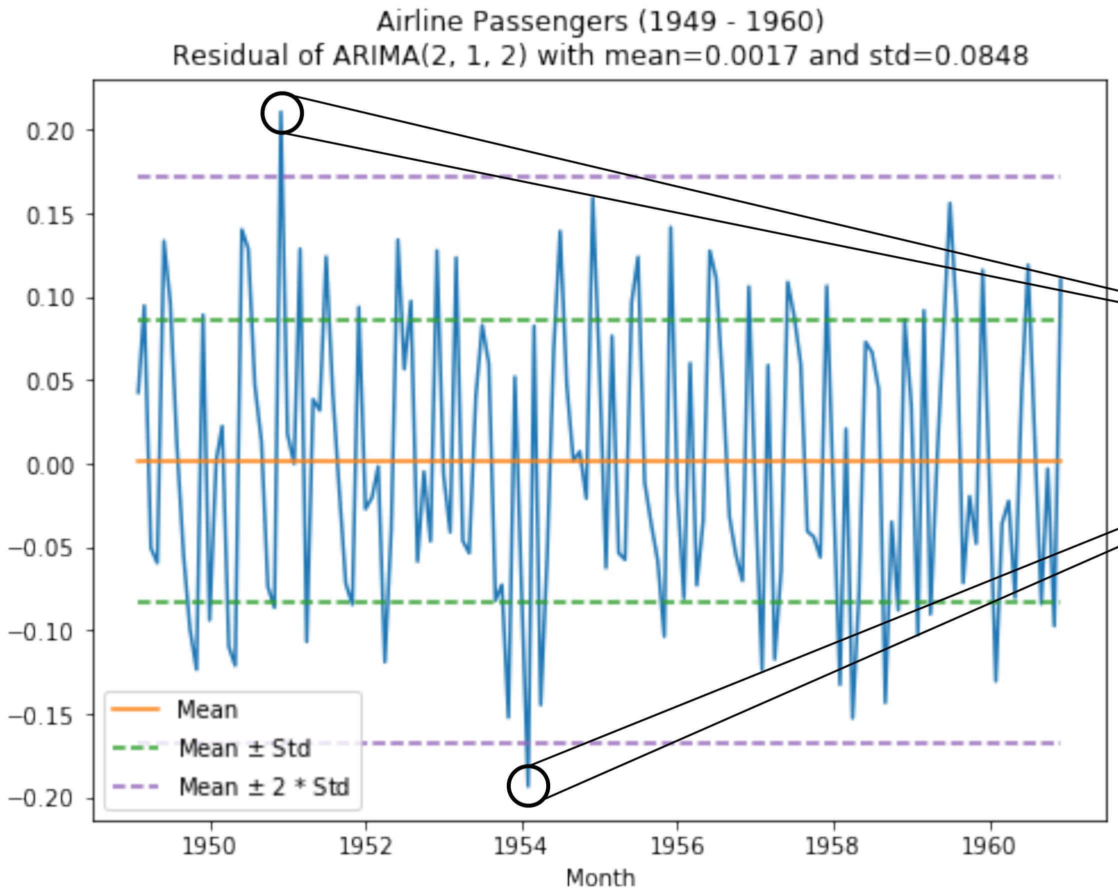
- It is often disputed whether Granger causality is truly causality
- Criticism: it works for pairs of variables (X, Y), but there may be Z that causes both X and Y!
- Also there are many people defending it
- Be cautious when you are making conclusions (refer to causal inference).

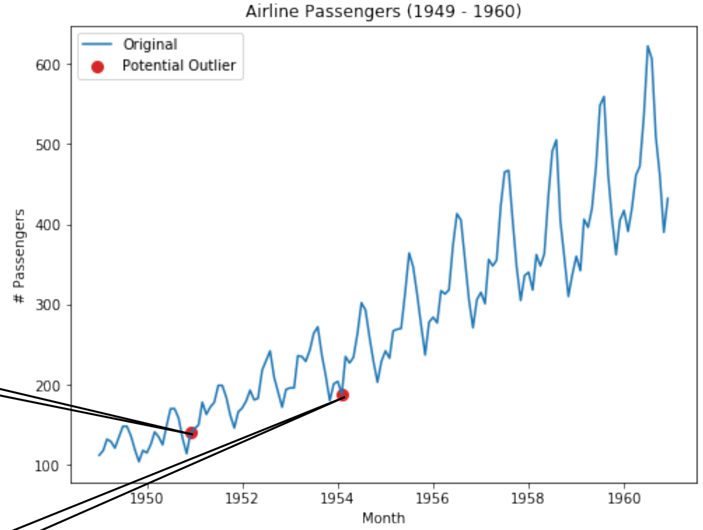
Outlier Detection

Outlier

- Build a reasonable prediction model (simple or complex)
- If $\hat{y}_i$ deviates significantly from the observed y_i, then y_i is likely to be an outlier
- Use confidence intervals to identify individual outliers
- In practice, individual outliers are not very interesting - analyze the residual series to identify patterns.

Example of residual series





Residual series: $y_i - \hat{y}_i$

What you should know

- The assumptions behind time series prediction and forecasting
- What autocorrelation plots tell us
- How autoregression models work
- Granger causality is built upon multivariate autoregressions
- Outliers can be detected through residual analysis

Thank You

Questions?