Unit 9 Live Session

Time Series Analysis Part 4: Mixed Autoregressive Moving Average (ARMA) Models Autoregressive Integrated Moving Average (ARIMA) Models Seasonal ARIMA (SARIMA) Models

Figure 1: South Hall

Class Announcements

- No HW this week

Roadmap

Rearview Mirror

- Mathematical formulation and key properties of AR and MA process
- Comparing AR models and MA models using simulated series
- Discuss the steps to build a time series model: Box-Jenkins' approach

Today

- Mixed Autoregressive Moving Average (ARMA) Models
- An introduction to the non-stationary time series model
- Random walk and integrated processes
- Autoregressive Integrated Moving Average (ARIMA) Models
- Seasonal ARIMA (SARIMA) Models

Looking Ahead

- Regression with multiple trending time series
- Cointegration
- Multivariate Time Series Models: Vector Autoregressive (VAR) model

Start-up Code

```
# Load required libraries
## Load a set of packages inclusing: broom, cli, crayon, dbplyr , dplyr, dtplyr, forcats,
#googledrive, googlesheets4, ggplot2, haven, hms, httr, jsonlite, lubridate , magrittr,
#modelr, pillar, purrr, readr, readxl, reprex, rlang, rstudioapi, rvest, stringr, tibble,
#tidyr, xml2
library(tidyverse)
# To create and work with tidy temporal data
library(tsibble)
# To work with date-times and time-spans
library(lubridate)
# Provides a collection of commonly used univariate and multivariate time series models
library(fable )
## To interact directly with the Quandl API and download data
library(Quandl)
# For analysing tidy time series data.
library(feasts)
```

An introduction to ARMA, ARIMA, and Seasonal ARIMA (SARIMA) Models

1. Last week, we were introduced to AR, MA, ARMA, and ARIMA models. AR, MA, and ARMA models are only appropriate when a time series is stationary.
2. We are often confronted with time series that are not stationary in the mean and variance (and other forms, such as seasonality and volatility clustering). Luckily, many non-stationary series can be simply transformed into stationary series using straightforward transformations such as differencing.
3. Here are some common reasons why time series might not be stationary in the mean:
    a. The series has a trend
    b. The series contains seasonal elements
    c. The series contains time-varying variance
4. We can take care of some of these problems by de-trending the data or by differencing the data. In the context of ARIMA modeling, first difference the data and then fit an ARMA model to the differenced series. If the data are seasonal, we use additional seasonal parameters, resulting in a SARIMA model - an ARIMA model with seasonal components.
5. Remember, here are the steps to building an ARIMA model (assuming that you already have your questions well-defined and data collected and cleansed):
    i. Conduct an EDA to determine if you need to transform the series to make it stationary.
    ii. Transform the series if needed.
    iii. Estimate several `ARIMA(p, d, q)x(P, D, Q)s` models, with starting values coming from examining time series plot, ACF, and PACF.
    iv. Evaluate the residuals of models with the lowest AIC and/or BIC values and more parsimonious models. Select the model where the residuals resemble a white nose.
    v. Answer your question / generate forecasts!

Seasonal Autoregressive Integrated Moving Average SARIMA(p,d,q)(P,D,Q)s Models

SARIMA models combine ARIMA models with additional parameters to capture seasonal components in time series data. We basically specify an "additional" ARIMA model for the seasonal components of the process.

This means there are seven parameters we need to estimate in the data. There are the normal ARIMA parameters: p the number of non seasonal AR lags, q the number of non seasonal MA lags, and d the non seasonal order of differencing. But then there are also the same components for the seasonal part of the process: P the number of seasonal AR lags, Q the number of seasonal MA lags, and D the seasonal differencing. There is also s the number of observations per year or seasonal period we care about. For monthly data we usually have $s = 12$ and for quarterly data we usually set $s = 4$ and so on.

We can express SARIMA(p,d,q)(P,D,Q)s models using the backshift notation as follows:

$$\phi(B)\Phi(B^s)(1-B)^d(1-B^s)^D x_t = \mu + \theta(B)\Theta(B^s)\epsilon_t$$

This shows that SARIMA models just have additional differencing and terms at seasonal lags in the time series process to capture the seasonal effects. Typically we don't want the total differencing $d + D$ in SARIMA models to be much more than 2 and generally do not set $D > 1$. Otherwise we risk overfitting the data.

Parameter Redundancy in ARIMA/SARIMA Models

One thing that can happen in SARIMA (and ARIMA) models is parameter redundancy. For example, suppose we have a random walk process:

$$x_t = \epsilon_t$$

We could also write that as:

$$x_t = x_{t-1} + \epsilon_t - \epsilon_{t-1}$$

The first formulation is an $ARIMA(0,0,0)$ model while the second formulation is an $ARIMA(0,1,1)$ model even though they describe the exact same process.

Hence the same time series can be described by multiple time series models, but there is also a model that represents the most parimonious expression of a time series process. We want to ensure we are estimating the most parismonious version and so prefer models that have less parameters.

To see that a time series process has parameter redundancy and can be reduced, we can use the backshift notational form.

For the process above, we could write it as:

$$x_t = x_{t-1} + \epsilon_t - \epsilon_{t-1}$$

$$x_t - x_{t-1} = \epsilon_t - \epsilon_{t-1}$$

$$(1 - B)x_t = (1 - B)\epsilon_t$$

So we can simplify the backshift notation form by dividing through by $(1 - B)$ to remove the parameter redundancy. This is a general fact about backshift notation and can help in simplifying time series.

An Analytical Analysis

- For following specifications:

1- $y_t = y_{t-1} - \frac{1}{4}y_{t-2} + w_t - \frac{1}{2}w_{t-1}$

2- $y_t = \frac{4}{3}y_{t-1} - \frac{1}{3}y_{t-2} + w_t - w_{t-1} + \frac{1}{4}w_{t-2}$

3- $y_t = \frac{1}{4}y_{t-1} + \frac{1}{10}y_{t-12} - \frac{1}{40}y_{t-13} + w_t + \frac{1}{2}w_{t-1}$

4- $y_t = y_{t-1} + y_{t-4} - y_{t-5} + w_t - \frac{1}{2}w_{t-1} - \frac{1}{2}w_{t-4} + \frac{1}{4}w_{t-5}$

- Where $\{w_t\}$ is white noise with zero means and standard deviation 1.

a) Rewriting the processes in `Seasonal ARIMA(p,d,q)(P,D,Q)` notation.

b) Check for parameter redundancy

c) Check the stationarity and invertibility of these processes

d) Identify each series

e) Compute the Mean of each process (If it is stationary)

f) Simulate each process separately

g) Check ACF and PACF plots of each simulation and discuss the pattern in these two plots

Is the efficient market hypothesis true in cryptocurrency market?

Bitcoin has become a fad among investors despite the ambiguity surrounding its nature and characteristics. Investors can speculate on bitcoin prices, and buy when they think prices are going up and sell when they think prices are going down. This would be profitable for investors if they can make a accurate predictions of bitcoin prices. However, the predictability of bitcoin prices has been questioned. According to the efficient market hypothesis, asset prices are unpredictable, so one can not beat the market.

The efficient market hypothesis is associated with the "random walk hypothesis," which states that asset prices are random walk, like the steps a drunk takes and, therefore, unpredictable.

$$P_t = P_{t-1} + w_t$$

- where $\{w_t\}$ is a Gaussian white noise $w_t \sim N(0, \sigma^2)$.

The current consensus is that the random walk is explained by the efficient market hypothesis, that the markets quickly and efficiently react to new information about assets, so most of the fluctuations in prices are explained by the changes in the instantaneous demand and supply of any given asset, causing the random walk-in prices. An implication is that technical analysis is undependable.

This exercise aims to investigate whether the cryptocurrency market is predictable and if the efficient market hypothesis is true

To test the null hypothesis of random walk or efficient market, we estimate an ARIMA and a random walk model for the period 01/2017 to 01/2022 and compare the six-month a head forecast performances of these two models. If the ARIMA model outperforms the random walk model, it suggests that the cryptocurrency market is not efficient during the sample period.

What does unpredictable price mean?

To better understate the random walk hypothesis, answer the following questions.

a) If the crypto price P_t follow a random walk process, what are the forecasts of P_{t+1}, P_{t+2} and P_{t+h}

b) If the crypto price P_t follow a random walk process, what are the variances of forecast of P_{t+1}, P_{t+2}, and P_{t+h}

Data Description and wrangling

This study only incorporates Bitcoin (BTC) prices but can be replicated for other cryptocurrencies to check for the robustness of its results.

We use a Quandl package to download the bitcoin prices from Quandl(). Quandl is a platform that provides its users with economic, financial, and alternative datasets. Users can download free data, buy paid data or sell data to Quandl. So the first step is to create an account that allows you to access their API, libraries, and tools.

```
## Downloading from Quandl
# Quandl.api_key("Your API KEY")
# bitcoin_daily = Quandl("BCHAIN/MKPRU", start_date="2017-01-01") %>%
#   as_tsibble(index=Date)

## Importing form the file
# bitcoin_daily <- read_csv("./data/Bitcoin_prices.csv") %>%
#   as_tsibble(index=Date)

# class(bitcoin_daily)
#
# head(bitcoin_daily)
# tail(bitcoin_daily)
```

Training and test sets

In order to evaluate forecast accuracy, we separate the data set into training and test data. The training data is used to estimate model parameters, and it is for 01/2017-04/2022. The test data is used to evaluate its accuracy, and it is for 01/2022-current.

```
# bitcoin_training <-bitcoin_daily %>%
#   filter( Date < ymd(20220101))
#
# tail(bitcoin_training)
#
# bitcoin_test <-bitcoin_daily %>%
#   filter( Date >= ymd(20220101))
#
# head(bitcoin_test)
# tail(bitcoin_test)
```

Model Development

a) Plot graphs of the training data set and the ACF/PACF. What do you notice? Is there any transformation neccessary?

```
# uncomment and replace with your code
```

b) Use a unit root test to determine whether differencing is required more objectively. If data is not stationary, difference the data, and apply the test again until it becomes stationary? How many differences are needed to make data stationary?

```
# uncomment and replace with your code
```

c) Plot graphs of the differenced data and try to identify an appropriate ARIMA model.

```
# uncomment and replace with your code
```

d) Use ARIMA() to estimate a random walk model and find an appropriate ARIMA model. What model was selected? Check the in-sample performance of the models using information criteria. Which one performs better?

```
#bit_fit <- # uncomment and replace with your code
```

e) Do residual diagnostic checking of random walk and ARIMA model. Are the residuals white noise? Use the ljung box test to check if the residuals are white noise.

```
# uncomment and replace with your code
```

f) Use the random walk models and chosen ARIMA model to forecast the next month and evaluate the forecast accuracy of these models.

- Recall that the two most commonly used measures of forecast accuracy are MAE(Mean absolute error) and RMSE(Root mean squared error) given by:

$$MAE = mean(|e_t|)$$

$$RMSE = \sqrt{mean(e_t^2)}$$

-Where e_t is forecast "error."

```
# uncomment and replace with your code
```

Reminders

1. Complete all videos and reading for unit 10