Codes

2023-03-21

# importing the necessary libraries  
library(pacman)  
p\_load(tsfe, ggplot2, tseries, forecast, xts, ModelMetrics, knitr, kableExtra, quantmod,ggfortify, dplyr )  
  
# Attaching the data set  
  
data("indices")  
  
# Obtaining the DJIA Indicies and the dates  
  
data <- data.frame(indices$date, indices$`DOW JONES INDUSTRIALS - PRICE INDEX`)  
  
# Renaming the variables for easier calling  
  
colnames(data) <- c("date", "DJI")  
  
summary <- summary(data$DJI)  
print(summary)

# Plotting the time series  
data %>%   
 ggplot(aes(x = date)) +  
 geom\_line(aes(y = DJI), col = "royalblue") +  
 theme\_minimal() +  
 theme(axis.line = element\_line(colour = "black"),  
 axis.ticks = element\_line(colour = "black"),  
 plot.title = element\_text(hjust = .5),  
 axis.text = element\_text(colour = "black"))+  
 labs(title = "Time series plot for DJIA Stock Price Indicies", col = "black")

returns <- Delt(data$DJI) %>%   
 as.data.frame()  
  
# Getting the dates  
  
returns$date <- data$date  
  
# renaming the columns  
  
colnames(returns) <- c("Returns", "date")  
  
# Piloting the returns  
  
returns <- returns %>%   
 na.omit() # Because of the formula for returns the first row had an empty value hence we drop it  
  
returns %>%   
 ggplot(aes(x = date)) +  
 geom\_line(aes(y = Returns), col = "royalblue") +  
 theme\_minimal() +  
 theme(axis.line = element\_line(colour = "black"),  
 axis.ticks = element\_line(colour = "black"),  
 plot.title = element\_text(hjust = .5),  
 axis.text = element\_text(colour = "black")) +  
 labs(title = "Time series plot for DJIA Equity Return Indicies",col = "black")

# The adf-test  
  
adf.test(returns$Returns, alternative = "stationary")

# Plotting the ACF  
  
acf(ts(returns$Returns))

# PACF  
pacf(ts(returns$Returns))

# Removing the last ten values  
  
series\_train <- window(returns$Returns, end = c(length(returns$Returns))-10)  
  
# Fitting the ARIMA model  
model <- auto.arima(series\_train)  
model

# Obtaining the actual values   
actual\_values <- tail(returns$Returns, 10)  
  
# Forecasting  
forecasted <- forecast(model, h = 10)  
  
# Obtaining the accuracy  
accuracy(forecasted, actual\_values)

# Grouping the daily returns to monthly  
data$date <- as.Date(data$date, format = "%Y-%m-%d")  
monthindex <- data[2:length(data$DJI),] %>%   
 group\_by(YearMonth = format(date, "%Y-%m")) %>%   
 group\_by(YearMonth) %>%   
 summarise(returns = sum(DJI))  
  
  
# Pbtaining the monthly returns  
  
monthlyreturns <- Delt(monthindex$returns) %>%   
 as.data.frame()  
  
# Adding the year month column  
  
monthlyreturns$date <- monthindex$YearMonth   
  
# Renaming the columns  
  
colnames(monthlyreturns) <- c("Monthly Return Indicies", "YearMonth")  
  
# Removing the one empty value  
  
monthlyreturns <- monthlyreturns %>%   
 na.omit()  
  
# Converting the Year Month variable to date  
  
monthlyreturns$YearMonth <- as.Date(paste0(monthlyreturns$YearMonth, "-01"), format = "%Y-%m-%d")  
  
# Ploting  
  
monthlyreturns %>%   
 ggplot(aes(x = YearMonth)) +  
 geom\_line(aes(y = `Monthly Return Indicies`), col = "royalblue") +  
 theme\_minimal() +  
 theme(axis.line = element\_line(colour = "black"),  
 axis.ticks = element\_line(colour = "black"),  
 plot.title = element\_text(hjust = .5),  
 axis.text = element\_text(colour = "black")) +  
 labs(title = "Time series plot for Monhtly DJIA Equity Return Indicies",col = "black") +  
 xlab("Year")

adf.test(monthlyreturns$`Monthly Return Indicies`)

# Removing the last ten values  
  
series\_train2 <- window(monthlyreturns$`Monthly Return Indicies`, end = c(length(returns$Returns))-10)  
  
# Fitting the ARIMA model  
model2 <- auto.arima(series\_train2)  
model2

# Obtaining the actual values   
actual\_values2 <- tail(monthlyreturns$`Monthly Return Indicies`, 10)  
  
# Forecasting  
forecasted2 <- forecast(model2, h = 10)  
  
# Obtaining the accuracy  
accuracy(forecasted2, actual\_values2)