IPCA

Daniel Sayao

2024-08-23

#Understanding IPCA #The Índice de Preços ao Consumidor Amplo (IPCA) isthe official inflation measure in Brazil, calculated by the BrazilianInstitute of Geography and Statistics (IBGE). Established in 1979, IPCA monitors the price changes of a basket of goods and services representative of consumption patterns among Brazilian households with monthly incomes ranging from 1 to 40 minimum wages. It covers multiple categories, such as food, transportation, housing, and healthcare, and serves as a critical indicator for economic planning, monetary policy, and contract adjustments in the country.

#Code Explanation #This code provides a step-by-step guide to load, analyze, and forecast future values of the IPCA using R. It uses time series analysis techniques, specifically ARIMA (AutoRegressive Integrated Moving Average), to predict inflation trends.

Install and load necessary packages

install.packages(“forecast”) install.packages(“tseries”)

library(forecast) library(tseries)

install.packages(“tinytex”) tinytex::install_tinytex()

Step 1: Load the data

Assuming the data is stored in a CSV file with columns: Date and IPCA_Index

data <- read.csv(“ipca_data.csv”)

data

Step 2: Convert the Date column to Date type and set it as the index

data$Date < -as.Date(data$Date, format=“%d/%m/%Y”)

head(data$IPCA_Index)

sum(is.na(data$IPCA_Index))

Step 3: Create a time series object

```
ipca_ts <- ts(data$IPCA_Index, start=c(1994,1), frequency=12) # Monthly data starting from 1994
ipca_ts
```

Step 4: Plot the time series to visualize it

```
plot(ipca_ts, main="IPCA Time Series", ylab="IPCA Index", xlab="Year")
```

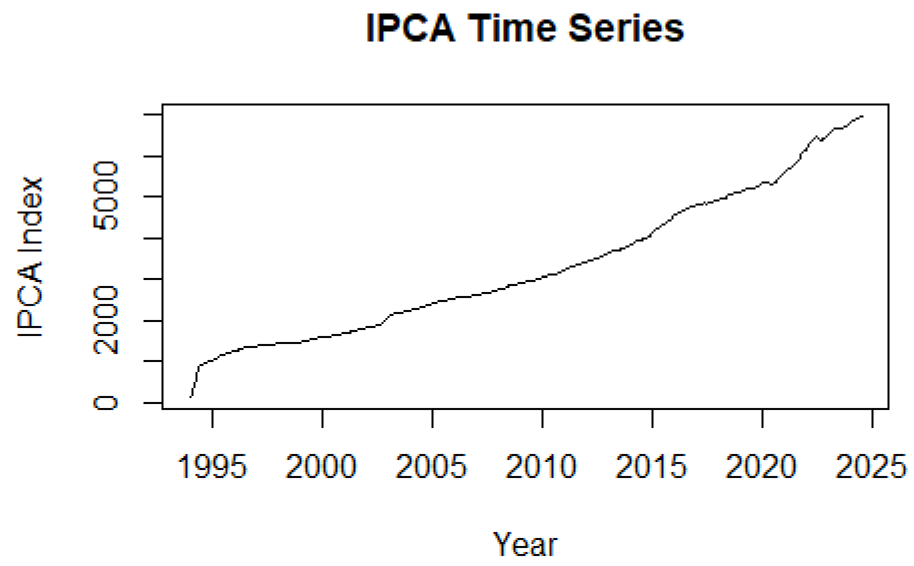

Figure 1: IPCA Time Series

Step 5: Check for stationarity using Augmented Dickey-Fuller test

```
adf_test <- adf.test(ipca_ts) print(adf_test)
```

Step 6: Differencing if necessary to make the series stationary

```
ipca_diff <- diff(ipca_ts, differences=1) plot(ipca_diff, main="Differenced IPCA Time Series",
ylab="Differenced IPCA Index", xlab="Year")
```

Step 7: Fit the ARIMA model

```
fit <- auto.arima(ipca_ts)
```

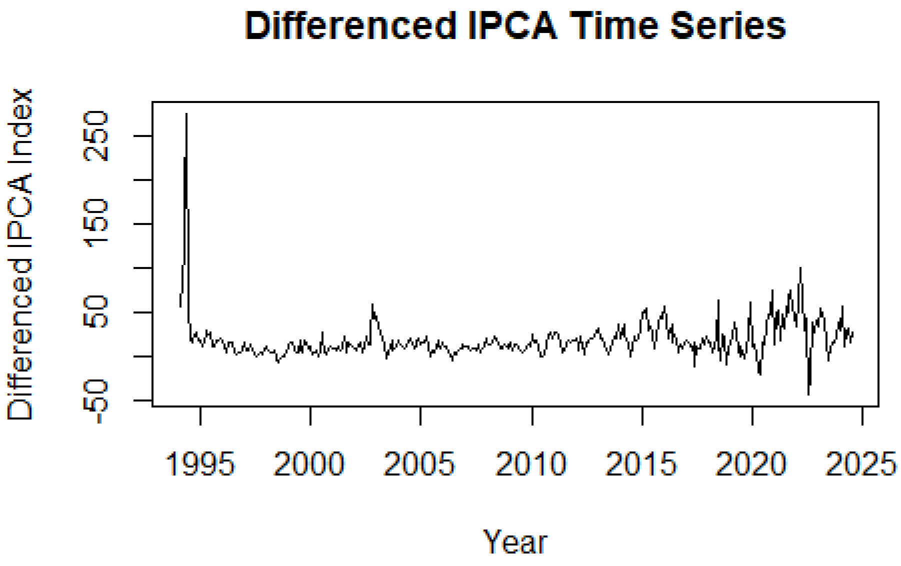

Figure 2: Differenced IPCA Time Series

Step 8: Summarize the model

summary(fit)

Step 9: Forecast future values

forecast_period <- 12 # Forecast for the next 12 months forecast_values <- forecast(fit, h=forecast_period)

Step 10: Plot the forecast

plot(forecast_values, main="IPCA Index Forecast", ylab="IPCA Index", xlab="Year")

Step 11: Print the forecasted values

print(forecast_values)

```
> print(forecast_values)          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95 Aug 2024       699
```

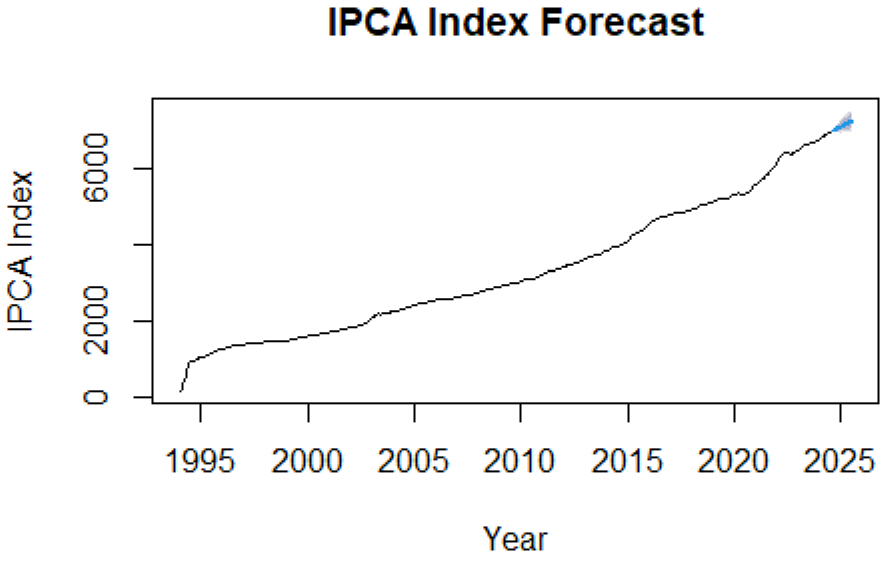

Figure 3: IPCA Index Forecast