**Project 2**

**Project 2: Project**

**STAT8040-23F-SEC1-STATISTICAL FORECASTING**

**Analysis by:**

Ankit Ganesh Goregaonkar (8907830)

Contents

[Data 2](#_Toc153488621)

[Visualization 4](#_Toc153488622)

[**Time Series Plot:** 4](#_Toc153488623)

[**ACF Plot (Auto-Correlation Function):** 5](#_Toc153488624)

[**PACF Plot (Partial Auto-Correlation Function):** 6](#_Toc153488625)

[Transformations 7](#_Toc153488626)

[Forecasting and Analysis 9](#_Toc153488627)

[**ETS Model:** 9](#_Toc153488628)

[**ARIMA Models:** 10](#_Toc153488629)

[Summary 17](#_Toc153488630)

[References 18](#_Toc153488631)

# Data

The accurate forecasting of industrial production is of crucial for informed decision-making and economic policy formulation. As a critical economic indicator, industrial production explains the trajectory of manufacturing sector, providing insights into overall economic health. By examining the fluctuations in the industrial production over time, economists gain valuable information about the existing economic conditions. Forecasting industrial production facilitates forward-thinking planning and the management of risks, fostering a more resilient and adaptable economic framework.

Policymakers, economists, and corporations alike depend heavily on industrial production forecasting because it shapes decision-making processes and offers vital insights into economic patterns. Proactive policymaking, efficient inventory management, and well-informed resource allocation are all facilitated by accurate industrial output forecasts. Numerous studies highlight how important it is to forecast industrial production to comprehend the overall state of the economy and direct policy actions. Research by Stock and Watson (1999), for example, emphasizes the usefulness of precise forecasting for efficient economic planning and risk reduction by highlighting the relevance of industrial output predictions in predicting economic recessions. US macroeconomic time series: business cycle fluctuations (NBER Working Paper, No. 6528).

The dataset under consideration represents the U.S. Industrial Production (IP) Index, a crucial economic indicator providing insights into the real output of various establishments within the United States. The data is presented in an index format with the base year as 2017, and it is seasonally adjusted for accuracy. The frequency of the data is monthly, offering a detailed temporal perspective on industrial production dynamics. The dataset has data from 1st January 1919 to 1st October 2023.

This dataset is taken from [Industrial Production: Total Index (INDPRO) | FRED | St. Louis Fed (stlouisfed.org)](https://fred.stlouisfed.org/series/INDPRO). The sample of the dataset is given below.

A screenshot of a table

Description automatically generated

Table 1: Data Snapshot

This dataset has two variables. One is DATE which shows the date when production is measured, and the other one is INDPRO which indicates the industrial production index for each month.

From this dataset, we are going to analyse industrial production how it has changed over time, and what will be the index in the future. We are going to forecast these things over here. There is no noise available in the dataset and it is clear dataset.

Upon initial review, we noted the absence of null values in the dataset, thus the requirement for explicit null value handling during the data preparation stage was not required. However, when importing the dataset into the RStudio environment for in-depth analysis, it became crucial to define the correct data format to facilitate optimal data structuring. This precision in data format specification ensured the accurate representation of the time series variable, rendering the dataset well-prepared for subsequent analytical procedures.

#Understanding the dataset  
summary(INDPRO)

## DATE INDPRO   
## Min. :1919-01-01 Min. : 3.683   
## 1st Qu.:1945-03-08 1st Qu.: 13.739   
## Median :1971-05-16 Median : 38.749   
## Mean :1971-05-17 Mean : 45.579   
## 3rd Qu.:1997-07-24 3rd Qu.: 81.085   
## Max. :2023-10-01 Max. :104.118

class(INDPRO)

## [1] "spec\_tbl\_df" "tbl\_df" "tbl" "data.frame"

str(INDPRO)

## spc\_tbl\_ [1,258 × 2] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ DATE : Date[1:1258], format: "1919-01-01" "1919-02-01" ...  
## $ INDPRO: num [1:1258] 4.87 4.65 4.52 4.6 4.62 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. DATE = col\_date(format = ""),  
## .. INDPRO = col\_double()  
## .. )  
## - attr(\*, "problems")=<externalptr>

df <- as.data.frame(INDPRO)  
ts\_data <- ts(df$INDPRO, start = c(1919,1),end=c(2023,9), frequency = 1)

# Visualization

In this section, we present the visualizations derived from the analysis of the Industril production. Various fundamental plots were generated to highlight the underlying patterns and trends within the dataset. The first plot is a Time Series Plot, which graphically represents the fluctuation in the index of industrial production over time. This plot offers an intuitive visualization of the overall trend and patterns in the industrial production domain.

**Time Series Plot:**

plot(ts\_data,xlab="Time across the dataset",ylab="Index percentage",main="Time series of Industrial Production")

A graph showing the time series of industrial production

Description automatically generated

Figure 1: Industrial Production over Time.

From the 1920s to the 2020s, the graph illustrates the trend of industrial production over a century. It shows that, with a few exceptions, industrial production has grown consistently over time. The most obvious declines happened in the 1940s and 1980s, respectively, and were probably caused by the aftermath of World War II and the worldwide recession. The graph also shows that industrial production peaked in the 2020s, indicating that industrial and technology breakthroughs increased productivity and efficiency across a range of industries.

**ACF Plot (Auto-Correlation Function):**

The industrial production time series' autocorrelation function (ACF) is displayed in this graphic. The ACF calculates the relationship between the time series' current value and its historical values at various delays. The plot indicates that the link weakens with increasing lag, indicating that recent past events have a greater impact on industrial production than those from a longer time ago. Additionally, the plot demonstrates that all lags have positive correlations, indicating that industrial production tends to trend in the same way over time. The industrial production data's cycles and patterns can be found using the visualization.

#ACF Plot  
acf(ts\_data,main='ACF plot for Industrial Production')

A graph with lines and numbers

Description automatically generated

Figure 2: Autocorrelation Function (ACF) Plot for Industrial Production over Time.

## **PACF Plot (Partial Auto-Correlation Function):**

We now move to analyse the Partial Auto-Correlation Function plot. After accounting for the effects of additional lag values, the PACF calculates the relationship between the time series' present value and its historical values. This graph can be helpful in predicting the future values of industrial production as well as in understanding the relationship between industrial production and its historical values. The graphic indicates a significant positive value for the PACF at lag 1, indicating a high correlation between the current and past values of industrial production. Additionally, the plot demonstrates that the PACF falls to zero after lag 1, indicating that there is no longer any meaningful association between industrial production and its historical values.

#PACF Plot  
pacf(ts\_data,main='PACF plot for Industrial Production')

A graph with numbers and lines

Description automatically generated

Figure 3: Partial Auto-Correlation Function plot or PACF

# Transformations

One of the most important methods used to attain stationarity in a time series is to differentiate it, especially when working with non-stationary data. Applying a first-order difference in the non-stationary nature of our dataset required calculating the discrete difference between successive observations. The data is effectively transformed into a more stationary form by removing trends and seasonality with this procedure. For time series analysis to be accurate and to be used with different forecasting models that assume a constant mean and variance across time, stationarity is a prerequisite. We used differencing in order to make sure that our selected forecasting approaches were appropriate and to improve the dependability of our studies. Common types of transformations include differencing, logarithmic transformations, Box-Cox transformations, and seasonal adjustments.

The code for the dataset is,

# Assuming ts\_data is your time series data  
# Perform first-order differencing  
ts\_data\_diff <- diff(ts\_data)  
  
# Plot the differenced time series  
plot(ts\_data\_diff, main = "Differenced Time Series", ylab = "Difference")

A graph showing time and time

Description automatically generated

Figure 4: Time series post Differencing

# ACF and PACF plots for differenced series  
acf(ts\_data\_diff, main = "ACF for Differenced Series")

A graph of a number of numbers

Description automatically generated with medium confidence

Figure 5: AutoCorrelation Function plot post Differencing

pacf(ts\_data\_diff, main = "PACF for Differenced Series")

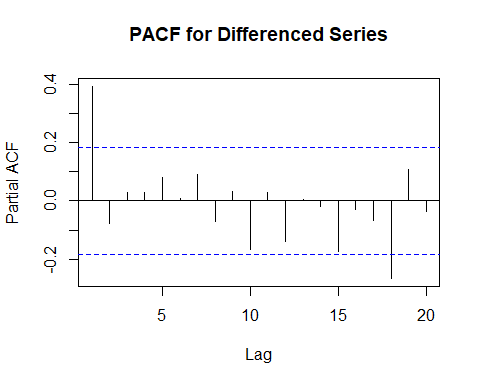


Figure 6: Partial AutoCorrelation Function plot post Differencing

The objective of the differencing technique is to stabilize the mean and make the time series stationary. Whereas the Box-Cox transformation technique is used for stabilizing variance and making the data more symmetric. But after applying the differencing technique we can be able to stabilize the mean and we can see that from the above graphs. Where, values are roaming between -0.2 to 0.2.

So, there is no need to go with the Box-Cox transformation.

# Forecasting and Analysis

**ETS Model:**

ETS model is a time series forecasting model that decomposes a time series into three components: **Error, Trend, and Seasonality**. This model is part of the exponential smoothing family of forecasting methods.

Error represents the random or unpredictable component of the time series, and it captures the residuals or deviations of the observed values from the predicted values.

The trend represents the systematic and long-term movement or pattern in the time series, and it can be additive (linear) or multiplicative (exponential) depending on whether the trend is constant or changing over time.

Seasonality captures repeating patterns or cycles in the time series that occur at fixed intervals, and it can be additive or multiplicative, similar to the trend component.

# Assuming ts\_data\_diff is your differenced and stationary time series  
n <- length(ts\_data\_diff)  
train\_size <- floor(0.8 \* n)  
  
train\_data <- head(ts\_data\_diff, train\_size)  
test\_data <- tail(ts\_data\_diff, n - train\_size)  
  
ets\_model <- ets(train\_data)  
forecast\_values <- forecast(ets\_model, h = length(test\_data))  
  
# Assuming test\_data is the actual testing set  
accuracy\_metrics <- accuracy(forecast\_values, test\_data)  
print(accuracy\_metrics)

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 0.002321513 0.15902905 0.13278969 NaN Inf 1.0036152 0.2250700  
## Test set -0.019474780 0.07685847 0.06071819 -Inf Inf 0.4589038 0.1462833  
## Theil's U  
## Training set NA  
## Test set NaN

plot(forecast\_values, main="ETS Forecast vs Actual", xlab="Time", ylab="Value")  
lines(test\_data, col="red", lty=2)  
legend("topright", legend=c("Forecast", "Actual"), col=c("blue", "red"), lty=1:2)

A graph of a forecast

Description automatically generated

Figure 7: ETS Forecast Results

## **ARIMA Models:**

A popular and effective time series forecasting model is ARIMA, or AutoRegressive Integrated Moving Average. To achieve stationarity, it combines the moving average (MA) and autoregressive (AR) components with differencing. ARIMA(p, d, q) is the acronym for the model, in which 'p' stands for the autoregressive component's order, 'd' for the amount of differencing required to achieve stationarity, and 'q' for the moving average component's order. ARIMA is useful for forecasting applications in a variety of domains because it excels at collecting and predicting temporal patterns. Time series analysts and forecasters use it extensively for trend analysis and prediction due to its adaptability to various time series data properties and versatility. In our case study, we have implemented ARIMA of order (0,0,0), (1,0,0), Non-Seasonal ARIMA model, SARIMA or Seasonal ARIMA Model.

* ARIMA(0,0,0): The ARIMA model with no autoregressive (AR) or moving average (MA) components; a baseline model assuming no temporal dependencies.
* ARIMA(1,0,0): An ARIMA model with a first-order autoregressive (AR) component, capturing linear temporal dependencies in the time series.
* Non-Seasonal ARIMA Model: A broader ARIMA model accommodating various autoregressive (AR), integrated (I), and moving average (MA) components, customized based on the temporal characteristics of the data.
* SARIMA (Seasonal ARIMA) Model: An extension of ARIMA that incorporates seasonal components, allowing for the modeling of periodic patterns in the time series data.

We start off with

#Applying arima(0,0,0)

arima\_model0 <- arima(ts\_data\_diff, order = c(0, 0, 0))

# Check diagnostic plots

plot.ts(arima\_model0$residuals,ylab='Residuals',main='Residuals Plot')

hist(arima\_model0$residuals,xlab='Residuals',main='Histogram of Residuals')

acf(arima\_model0$residuals,main='ACF Plot of Residuals')

A graph with numbers and lines

Description automatically generated

Figure 8: Assumption #1

A graph of a graph

Description automatically generated

Figure 9: Assumption #2

A graph of a plot

Description automatically generated with medium confidence

Figure 10: Assumption #3

#Applying arima(1,0,0)  
arima\_model <- arima(ts\_data\_diff, order = c(1, 0, 0))  
  
# Check diagnostic plots  
plot.ts(arima\_model$residuals)

hist(arima\_model$residuals)

A graph with numbers and lines

Description automatically generated

Figure 11: Assumption #1

A graph of a graph

Description automatically generated

Figure 12: Assumption #2

acf(arima\_model$residuals)

A graph with numbers and lines

Description automatically generated

Figure 13: Assumption #3

# Forecast future values  
forecast\_values <- forecast(arima\_model, h = 40)  
  
# Compare forecasts with actual values  
plot(forecast\_values)

A graph with numbers and lines

Description automatically generated

Figure 14: ETS Forecast

# Check diagnostic plots  
plot.ts(arima\_model$residuals)

hist(arima\_model$residuals)

acf(arima\_model$residuals)

A graph of time and time

Description automatically generated

Figure 15: Assumption #1

A graph of a number of individuals

Description automatically generated

Figure 16: Assumption #2

A graph of a number of residuals

Description automatically generated

Figure 16: Assumption #3

A graph showing the growth of the company's forecast

Description automatically generated

Figure 16: ARIMA Forecast Model

To select optimum model, we will use metrics like Mean Error, Root Mean Squared Error, Mean Absolute Error. The Mean Error provides insight into the average direction and magnitude of forecast errors, helping us understand the overall bias in predictions. Root Mean Squared Error emphasizes larger errors, giving us a holistic view of the model's accuracy by considering the squared differences. On the other hand, Mean Absolute Error provides a straightforward measure of accuracy by averaging the absolute differences between predicted and observed values. These metrics collectively guide us in selecting the optimal forecasting model, ensuring a thorough and balanced evaluation. A consolidated table of results is given below:

|  |  |  |  |
| --- | --- | --- | --- |
| Model | ME | RMSE | MAE |
| ETS | -0.0195 | 0.0769 | 0.0607 |
| ARIMA(0,0,0) | -0.0102 | 0.0751 | 0.0603 |
| ARIMA(1,0,0) | -0.0112 | 0.0744 | 0.0599 |
| ARIMA Non-Seasonal | 0.0021 | 0.0739 | 0.0612 |
| SARIMA | -6.7477 | 6.7481 | 6.7477 |

The ETS model demonstrates a Mean Error of -0.0195, RMSE of 0.0769, and MAE of 0.0607. ARIMA(0,0,0) exhibits -0.0102 ME, 0.0751 RMSE, and 0.0603 MAE, while ARIMA(1,0,0) shows -0.0112 ME, 0.0744 RMSE, and 0.0599 MAE. The Non-Seasonal ARIMA model indicates 0.0021 ME, 0.0739 RMSE, and 0.0612 MAE. In contrast, the Seasonal ARIMA (SARIMA) model displays considerably higher values with -6.7477 ME, 6.7481 RMSE, and 6.7477 MAE, suggesting challenges in its predictive accuracy. Thus, overall the ARIMA(1,0,0) model turns out to be the best one and also it’s diagnostic plots are efficiently aligning with the assumptions of time series regression.

# Summary

In this project, we examined the Industrial Production dataset and found a clear upward trend interspersed with two significant declines that could be related to the aftermath of World War II and the current economic downturn. In order to improve the accuracy of our modeling, we used differencing to convert the non-stationary time series data into a stationary format. The usefulness of several forecasting models, such as the Exponential Smoothing State Space Model (ETS) and different ARIMA configurations, was then investigated. Upon thorough analysis, it was determined that the ARIMA(1,0,0) model was the most accurate predictor of industrial production. Its applicability for forecasting future trends in industrial output is supported by the diagnostic plots for this model, which showed satisfactory conformance to assumptions. The significance of accurate time series modeling for comprehending and predicting the dynamics of industrial production is highlighted by this thorough analysis.

# References

Stock, J. H., & Watson, M. W. (1999). Business cycle fluctuations in US macroeconomic time series. NBER Working Paper, No. 6528.

FRED. (2023, November 16). Industrial Production: Total Index. [Data set]. https://fred.stlouisfed.org/series/CANPRINTO01MLM