Forecasting Monthly Inflation in Pakistan: An Academic Guide to ARIMA and Penalized Regression Techniques in R

Current Date: Tuesday, May 6, 2025, 8:49 PM Asia/Karachi
Location: Rawalpindi, Punjab

Part I: Foundations and Data Acquisition for Pakistan Inflation Forecasting

Section 1: Contextualizing Inflation in Pakistan

Overview of Pakistan's Inflation

Inflation in Pakistan has historically demonstrated significant volatility. This instability is a result of a complex interplay between domestic and international factors. Domestically, monetary policy decisions, persistent fiscal deficits, disruptions in supply chains, and the performance of the agricultural sector play crucial roles.[1] Externally, Pakistan's inflation is sensitive to global commodity price shocks, particularly oil, and fluctuations in the exchange rate.[4] This inherent volatility presents a considerable challenge for accurate inflation forecasting, yet simultaneously underscores its critical importance for economic stability.

Emerging economies, such as Pakistan, often grapple with distinct inflationary pressures stemming from their structural characteristics. These include ongoing institutional adjustments and the significant weight of food and energy components within the Consumer Price Index (CPI) basket.[4] Understanding this specific context is paramount when selecting appropriate forecasting models and interpreting their results. For instance, the State Bank of Pakistan (SBP), the nation's central bank, actively targets inflation, making precise and reliable inflation forecasts an indispensable input for effective monetary policy formulation and implementation.[1]

The Significance of Accurate Inflation Forecasting

The ability to accurately forecast inflation holds profound significance for various economic agents. It directly influences the economic decisions made by households, guides the strategic planning of businesses, and informs the policy measures enacted by governmental bodies.[4] Accurate forecasts contribute to a reduction in economic uncertainty, facilitate better long-term investment planning, and help mitigate arbitrary wealth transfers between debtors and creditors that can arise from unanticipated price level changes.[1] Furthermore, robust inflation forecasts are crucial for sound budget forecasting and the effective management of fiscal policy.[9]

Specific Challenges Related to Forecasting Inflation in Pakistan

Forecasting inflation in Pakistan is compounded by several specific challenges:

- **Data Quality and Availability:** While data provision through national sources like the State Bank of Pakistan's EasyData portal [10] and the Pakistan Bureau of Statistics (PBS) [12] has been progressively improving, issues regarding the consistency of historical series, varying base years, and the overall depth of available data can pose challenges for time series analysis.
- **Structural Breaks:** The Pakistani economy has experienced numerous policy shifts, economic reforms, and significant external shocks over the years. These events can induce structural breaks in economic time series, including inflation, thereby affecting the stability of model parameters and predictive accuracy.[13]
- **Inherent Volatility:** As previously noted, Pakistan's inflation series exhibits high intrinsic volatility. This makes it more difficult for forecasting models to discern underlying patterns from noise, potentially leading to less precise forecasts.[13]
- **Calendar Effects:** Standard methods for seasonal adjustment in time series models are typically based on the Gregorian calendar. However, certain economic activities and consumption patterns in Pakistan may be influenced by the Islamic calendar (e.g., Ramadan, Eid festivals), which shifts relative to the Gregorian calendar each year. While this guide will focus on standard ARIMA/SARIMA models which handle fixed seasonality, accounting for such moving holiday effects represents an advanced consideration for further research and could potentially improve forecast accuracy.[3]

The economic context of Pakistan, particularly its vulnerability to supply-side shocks such as fluctuations in food and energy prices, and the significant pass-through effects from the exchange rate to domestic prices, suggests that forecasting models should ideally be capable of incorporating these external influences.[1] Univariate ARIMA models, which rely solely on the past values of inflation [1], might be outperformed by models that can explicitly include these exogenous variables. Consequently, the research design for forecasting inflation in Pakistan should prioritize the evaluation of ARIMAX (ARIMA with exogenous variables) models and penalized regression techniques, as the latter are well-suited to handling multiple predictors.

Furthermore, the literature on inflation forecasting, particularly for developed economies like the U.S., has noted the "episodic" nature of the Phillips curve's forecast performance.[18] This implies that the predictive power of models based on relationships like the Phillips curve (which links inflation to economic activity) can vary over time. It is plausible that similar instability affects inflation forecasting models in

Pakistan. This suggests that no single model is likely to be universally superior across all time periods. Therefore, in practical applications, periodic re-evaluation of model performance, and potentially the use of model averaging techniques or dynamic model selection strategies, could prove beneficial for maintaining forecast accuracy. The research should, if data permits, assess model performance over different sub-periods and acknowledge that the best-performing model identified in the study might not retain its superiority indefinitely due to evolving economic structures and conditions.[4]

Section 2: Assembling the Dataset: Core Inflation and Predictor Variables

A robust forecasting model relies heavily on the quality and relevance of the underlying data. This section details the primary dependent variable, the Consumer Price Index (CPI) for Pakistan, and a selection of key exogenous variables commonly considered in inflation forecasting.

Dependent Variable: Monthly Consumer Price Index (CPI) for Pakistan

The primary variable for this research project is the monthly inflation rate, derived from the Consumer Price Index (CPI) of Pakistan.

- **Primary National Sources:**
  - **Pakistan Bureau of Statistics (PBS):** The PBS is the official agency responsible for compiling and disseminating CPI data in Pakistan. Recent publications, such as the "Monthly Price Indices for April 2025 (Base Year 2015-16)," are typically available on their website.[12] The PBS also provides historical data, which may have different base years (e.g., 2007-08 [19]). Monthly reviews and detailed annexures are often available in PDF and Excel formats.[19]
  - **State Bank of Pakistan (SBP) EasyData Portal:** This is a crucial resource for Pakistani macroeconomic time series.[10] The portal hosts the "National CPI, an Inflation Measure (Year-on-Year basis)" under the Series Key: TS_GP_PT_CPI_M.P00011516. This series, based on the 2015-16 base year, provides data from July 2016 to March 2025.[21] Older CPI series with different base years, such as 1980-81 (covering August 1964 - December 1973, and then March 1986 - December 1994) [24], and an Urban CPI series (Base 2015-16, from July 2016 to March 2025) [26], are also accessible.
- **International Sources (for supplementary data or cross-validation):**
  - **International Monetary Fund (IMF) Data Portal:** The IMF's "Consumer Price Index (CPI)" dataset includes national CPIs for many countries, including Pakistan.[27] Researchers can typically query for Pakistan, select indicators like

"Consumer Price Index, All items, Percentage change, Previous year," and download the data in CSV format.[33] The IMF DataMapper also presents "Inflation rate, average consumer prices" and "Inflation rate, end of period consumer prices" for Pakistan, though these are often annual figures or projections.[34]

- **World Bank Data:**
  - The World Bank's *Global Inflation Database* covers the period 1970-2025 and includes headline CPI inflation. Data is downloadable in Excel and Stata formats.[35]
  - The *World Development Indicators (WDI)* database provides "Inflation, consumer prices (annual %)" under the series code FP.CPI.TOTL.ZG.[36]
- **Food and Agriculture Organization (FAO) FAOSTAT:** This database provides General and Food CPI data from January 2000 onwards, downloadable in CSV format. FAOSTAT employs statistical estimation procedures to fill data gaps.[39]

- Choice of CPI Series for Modeling:
  For consistency and relevance to current economic conditions, the PBS National CPI (Base Year 2015-16), accessible via SBP EasyData (Series Key: TS_GP_PT_CPI_M.P00011516), is recommended as the primary target variable for this project.21
  The monthly inflation rate (πt) should be calculated as the year-on-year percentage change in the CPI:
  πt=(CPIt−12CPIt−CPIt−12)×100
- Core Inflation Measures:
  To understand underlying inflation trends by excluding volatile components, core inflation measures are essential:
  - **PBS Non-Food Non-Energy (NFNE) Core Inflation:** The PBS calculates and publishes NFNE core inflation rates for both Urban and Rural areas.[20] For instance, in March 2025, the Urban NFNE YoY inflation was reported at 8.2%, and Rural NFNE YoY inflation was 10.2%.[20] These series provide valuable insights into persistent inflationary pressures.
  - **World Bank Core CPI Inflation:** The World Bank's Global Inflation Database also includes a series for Core CPI inflation (FP.CPI.CORE.ZG).[35]

Exogenous Variables (Predictors) for Inflation Forecasting

The selection of appropriate exogenous variables is crucial for building robust ARIMAX and penalized regression models. Based on economic theory and data availability for Pakistan, the following categories and specific variables are recommended:

- **Monetary Variables:**
  - **Money Supply (M1, M2):**
    - *Source:* SBP EasyData is the primary source, typically found under "Monetary Statistics".[10] Data is available monthly in PKR Million. Historical data for M1 (Narrow Money) and M2 (Broad Money) often extends back to January 1988.[46] CEIC and TradingEconomics also report this data, often sourcing it from the SBP.[46]
    - *IMF Source:* The Monetary and Financial Statistics (MFS) dataset on the IMF Data Portal is another potential source.[28]
  - **SBP Policy Rate:**
    - *Source:* SBP EasyData provides the "State Bank of Pakistan's Policy (Target) Rate" (Series Key: TS_GP_IR_SIRPR_AH.SBPOL0030).[10] Although the frequency is listed as "As-Needed," changes occur at discrete intervals, effectively creating a step function when sampled monthly. The current "Policy (Target) Rate" definition was introduced on May 25, 2015.[59]
    - *Other Sources:* CEIC reports the SBP Reverse Repo Rate as the policy rate monthly from February 1992.[60] TradingEconomics shows daily interest rate data from 1992 [61], with recent reports indicating a rate cut to 11% in May 2025.[5] OpenData Pakistan hosts an "Interest Rates 2011-2019" dataset from SBP.[62] The IMF's IFS or MFS datasets may also contain relevant policy rates.[33]
  - **KIBOR (Karachi Interbank Offered Rate):**
    - *Source:* SBP EasyData provides daily KIBOR data; for example, 1-Month KIBOR is available from June 2005.[65] This daily data will require aggregation (e.g., monthly average) for use in this project.
    - *Other Sources:* The SBP website publishes daily KIBOR rates for 3-Month, 6-Month, and 12-Month tenors.[67] The Karandaaz Data Portal also lists daily 1-month KIBOR.[68] Khistocks.com is another platform showing KIBOR rates.[69]
- **External Sector Variables:**
  - **PKR/USD Exchange Rate:**
    - *Source:* SBP EasyData offers "Average Exchange rate of Pak Rupees per U.S. Dollar" (Series Key: TS_GP_ER_FAERPKR_M.E00220) and "Month end Exchange rate of Pak Rupees per U.S. Dollar" (Series Key: TS_GP_ER_FMEERPKR_M.E00220), both available monthly from August 1947.[23]
    - *IMF Source:* The Exchange Rate (ER) dataset on the IMF Data Portal.[27] The

International Financial Statistics (IFS) typically includes monthly average and end-of-period exchange rates.[27]
        - *World Bank WDI:* Provides annual official exchange rates.[83]
    - **Exports, Imports, and Trade Balance (Goods, USD Million):**
        - *Source:* SBP EasyData's "Export and Import of Goods and Services" dataset (Series Key: TS_GP_BOP_XMGS_M) contains monthly data from July 2005 in USD Million.[23]
        - *IMF Source:* The International Trade in Goods (ITG) dataset, formerly Direction of Trade Statistics (DOTS), on the IMF Data Portal offers monthly data.[27]
        - *World Bank WITS:* The World Integrated Trade Solution platform provides trade data, potentially including monthly figures, with download options in CSV/Excel.[97]
        - *Other Sources:* The Pakistan Ministry of Commerce publishes monthly trade statistics.[104] CEIC also reports monthly Trade Balance, Total Exports, and Total Imports.[105]
- **Real Sector Variables:**
    - **Industrial Production Index (Quantum Index of Large Scale Manufacturing - QIM):**
        - *Source:* SBP EasyData offers the "Quantum Index Series of Selected Large-scale Manufacturing Items (base 2015-16)" (Series Key: TS_GP_RL_LSM1516_M), available monthly from July 2016.[23] Older series with base year 2005-06 (July 2007 - June 2022) are also available.[106]
        - *PBS Source:* The Pakistan Bureau of Statistics publishes QIM data. Information for February 2025 (base 2015-16) is noted.[12] Archive tables for base years 2015-16 (from 2015-16) and 2005-06 (from 2005-06) are available, though often in PDF format.[107] Opendata.com.pk may host some PBS manufacturing data in CSV, but these might be for specific items rather than the overall QIM.[113]
        - *IMF Source:* The Production Indexes (formerly IPI) dataset on the IMF Data Portal is a potential source.[27] The IFS may also contain a seasonally adjusted Industrial Production Index.[116]
        - *Other Sources:* CEIC reports the Manufacturing Quantum Index (base 2015-16) and Industrial Production Index Growth YoY (from May 1998).[105] TradingEconomics reports Industrial Production and Manufacturing Production monthly (percent change, 2015-2016=100 NSA) from PBS data.[118]
- **Global Factors:**
    - **International Oil Prices (e.g., Brent Crude):**

- *IMF Source:* The IMF Primary Commodity Prices (PCP) database provides monthly data. This can be accessed via the Commodity Data Portal or an Excel Database.[119] Historical data often starts from 1980 or 1992. YCharts shows the IMF Brent Crude Oil Spot Price (Indicator: I:BCOSPNM) monthly, with data for March 2025 updated on May 2, 2025.[125]
- *FRED Source:* The Federal Reserve Economic Data (FRED) offers "Crude Oil Prices: Brent - Europe (DCOILBRENTEU)" on a daily basis, which can be aggregated to monthly.[126]
- *Other Sources:* DataHub.io provides Brent monthly data (file: brent-month.csv).[130] The World Bank Commodity Prices "Pink Sheet" also includes monthly data for Brent crude.[131]

- **Global Food Price Index:**
  - *FAO Source:* The FAO Food Price Index (FFPI) offers monthly data from 1990, downloadable in CSV format directly from the FAOSTAT website.[41]
  - *IMF Source:* The IMF Primary Commodity Prices (PCP) database includes various food price indices.[119] YCharts displays the IMF Food and Beverage Price Index (Indicator: I:FBPI) monthly, with data for March 2025 updated on May 2, 2025.[136]
  - *World Bank Source:* The World Bank Commodity Prices "Pink Sheet" also includes food price indices.[131]

Table: Recommended Data Sources for Inflation Forecasting in Pakistan

To streamline the data collection process, the following table summarizes recommended sources for key variables. This table is critical as it centralizes diverse information, saving significant research time and directly addressing the user's need for data source identification. It provides a clear, actionable starting point, highlights potential data limitations (like varying start dates or base years), and supports reproducibility.

Variable	Category	Primary Source (Pakistan)	Series Name/Key (Primary)	Secondary Source (International)	Series Name/Key (Secondary)	Typical Frequency	Historical Start (Primary)	Historical Start (Secondary)	Download Format(s)	Direct Link/Portal
Infla	Depe	SBP	TS_G	IMF	CPI,	Mont	Jul-2	Varie	CSV,	(http

tion Rate (YoY %)	ndent Var.	Easy Data / PBS	P_PT_CPI_M.PO0011516 (Base 2015-16)	Data Portal	All items, % change, Previous year	hly	016	s	Excel	s://easydata.sbp.org.pk/), [IMF Data]()
Core Inflation (NFNE YoY %)	Dependent Var.	PBS	Monthly Review of Price Indices	World Bank	FP.CPI.CORE.ZG (Global Inflation DB)	Monthly	Varies	1970	PDF, Excel	(https://www.pbs.gov.pk/price-statistics).(https://www.worldbank.org/en/research/brief/inflation-database,(https://www.worldbank.org/en/research/brief/inflation-database))

Money Supply M1 (PKR Million)	Monetary	SBP Easy Data	e.g., from Monetary Statistics	IMF Data Portal (MFS)	Narrow Money	Monthly	Jan-1988 (CEIC/SBP)	Varies	CSV, Excel	(https://easydata.sbp.org.pk/), (https://data.imf.org/MFS, (https://data.imf.org/MFS))
Money Supply M2 (PKR Million)	Monetary	SBP Easy Data	e.g., from Monetary Statistics	IMF Data Portal (MFS)	Broad Money	Monthly	Jan-1988 (CEIC/SBP)	Varies	CSV, Excel	(https://easydata.sbp.org.pk/), (https://data.imf.org/MFS, (https://data.imf.org/MFS))
SBP Policy Rate (%)	Monetary	SBP Easy Data	TS_GP_IR_SIRR_AH.SBPOL0030	CEIC	State Bank of Pakistan: Reverse Repo Rate	As Needed/Monthly	May-2015 (current def.)	Feb-1992	CSV, Excel	(https://easydata.sbp.org.pk/), CEIC

KIBOR (e.g., 3-Month Avg %)	Monetary	SBP Easy Data / SBP	Daily KIBOR (needs aggregation)	-	-	Daily/Monthly	Jun-2005 (SBP Easy Data)	-	CSV, Excel	(https://easydata.sbp.org.pk/),(https://www.sbp.org.pk/ecodata/kibor_index.asp,(https://www.sbp.org.pk/ecodata/kibor_index.asp))
PKR/USD Exchange Rate (Average)	External	SBP Easy Data	TS_GP_ER_FAERPKR_M.E00220	IMF Data Portal (ER/IFS)	Exchange Rate, PKR per USD, Period Average	Monthly	Aug-1947	Varies	CSV, Excel	(https://easydata.sbp.org.pk/),(https://data.imf.org/ER,(https://data.imf.org/ER))
Exports of	External	SBP Easy	Part of	IMF Data	Goods,	Monthly	Jul-2	Varies	CSV,	(https://ea

Goods (USD Million, FOB)	nal	Data	TS_GP_BOP_XMGS_M	Portal (ITG/DOTS)	Value of Exports, FOB, US Dollars	hly	005	s	Excel	sydata.sbp.org.pk/),(https://data.imf.org/dot,(https://data.imf.org/dot))
Imports of Goods (USD Million, CIF/FOB)	External	SBP Easy Data	Part of TS_GP_BOP_XMGS_M (FOB)	IMF Data Portal (ITG/DOTS)	Goods, Value of Imports, CIF, US Dollars	Monthly	Jul-2005	Varies	CSV, Excel	(https://easydata.sbp.org.pk/),(https://data.imf.org/dot,(https://data.imf.org/dot))
Industrial Production (QIM, Base 2015-16)	Real Sector	SBP Easy Data / PBS	TS_GP_RL_LSM1516_M	IMF Data Portal (IFS)	Industrial Production, Index	Monthly	Jul-2016	Varies	CSV, Excel, PDF	(https://easydata.sbp.org.pk/),(https://www.pbs.gov.pk/content/qim,(h

										ttps://www.pbs.gov.pk/content/qim))
Brent Crude Oil Price (USD/bbl)	Global	-	-	IMF PCP / FRED	I:BCOSPNM (IMF via YCharts) / DCOILBRENTEU (FRED)	Monthly	-	1980s	CSV, Excel	IMF PCP, (https://fred.stlouisfed.org/)
Global Food Price Index (2014-16=100)	Global	-	-	FAOSTAT	FAO Food Price Index	Monthly	-	Jan-1990	CSV	(https://www.fao.org/faostat/en/#data/CP)

A significant practical consideration when assembling the dataset is the need to reconcile data from different base years, particularly for the CPI and QIM series.[19] Time series analysis necessitates consistent data definitions over the study period. Using data from different base years directly would introduce artificial shifts or breaks in the series. To address this, the researcher must either splice the series by rebasing them to a common year or, more commonly for stationarity purposes in modeling, convert all relevant series into growth rates (e.g., year-on-year percentage change for inflation, and year-on-year percentage change or differenced logarithms for QIM). The latter approach is generally preferred as it also helps in achieving stationarity, a prerequisite for ARIMA modeling.

Another important point is that KIBOR data is typically available on a daily basis.[65] Since the inflation data and most other predictors will be on a monthly frequency, the daily KIBOR series will need to be aggregated to a monthly frequency. This usually involves calculating the monthly average of the daily rates. The specific method of aggregation (e.g., average of all daily rates in a month, end-of-month rate, or start-of-month rate) should be clearly documented and applied consistently. This choice could also form part of a sensitivity analysis if deemed critical.

Section 3: Data Acquisition and Initial Handling in R

Once the data sources are identified, the next step is to acquire the data and perform initial handling using R. This involves loading the data into R, cleaning variable names, parsing dates, creating time series objects, and merging different datasets into a cohesive analytical dataset.

Essential R Packages for Data Tasks

A suite of R packages facilitates these data management tasks efficiently:

- **readr**: For fast and friendly reading of delimited text files, especially CSV files.[137]
- **readxl**: For importing data from Microsoft Excel files (.xls and .xlsx formats).[139]
- **dplyr**: A powerful package for data manipulation, including filtering rows, selecting columns, creating new variables, and joining datasets.[140]
- **lubridate**: Simplifies working with dates and times, offering robust functions for parsing various date formats and performing date arithmetic.[141]
- **zoo**: Provides infrastructure for regular and irregular time series, with useful functions like as.yearmon for monthly data and na.locf or na.approx for time series imputation.[142]
- **httr**: If accessing data via APIs (e.g., from IMF or FRED), this package is essential for making HTTP requests.
- **jsonlite**: Useful for parsing JSON data, which is a common format for API responses.
- **janitor**: Contains helpful functions for cleaning data, notably clean_names() for standardizing column names.[144]
- **purrr**: Enhances R's functional programming capabilities; its reduce() function is particularly useful for iteratively joining multiple data frames.[145]

Step-by-Step R Code Examples

The following R code snippets illustrate common data acquisition and initial handling tasks. It's crucial to install these packages first if they are not already available in the R environment (e.g., install.packages("readr")).

- **Loading Packages:**
  Code snippet
  ```
  library(readr)
  library(readxl)
  library(dplyr)
  library(lubridate)
  library(zoo)
  library(janitor)
  library(purrr)
  ```

- **Reading Data:**
  - From a CSV file:
    Code snippet
    ```
    # Example: Reading monthly CPI data
    # Assume 'cpi_data.csv' has columns 'Date_String' and 'CPI_Value'
    # readr::read_csv automatically prints guessed column types, which should be checked.
    raw_cpi_data <- readr::read_csv("path/to/your/cpi_data.csv")
    # [137, 138]
    ```

  - From an Excel file:
    Code snippet
    ```
    # Example: Reading exchange rate data from an Excel sheet
    # readxl::read_excel arguments like col_names, skip, sheet are important.
    raw_exchange_rate_data <-
    readxl::read_excel("path/to/your/exchange_rate_data.xlsx", sheet = "MonthlyRates")
    # [139]
    ```

- Cleaning Column Names:
  Standardizing column names (e.g., to snake_case) improves code readability and consistency.
  Code snippet
  ```
  cpi_data <- janitor::clean_names(raw_cpi_data)
  exchange_rate_data <- janitor::clean_names(raw_exchange_rate_data)
  # [144]
  ```

- Date Parsing and Conversion to yearmon:
  Economic time series data often requires careful handling of date columns. The lubridate package offers flexible parsing functions, and zoo::as.yearmon is ideal for monthly data.
  Code snippet
  ```
  # Assuming cpi_data has a column 'date_string' like "Jan-2020" or "2020-01-15"
  # If date_string is like "Jan-2020", parse with mdy or other appropriate lubridate
  function
  # For "YYYY-MM-DD" format:
  # cpi_data$date_column <- lubridate::ymd(cpi_data$date_string)
  # For "Mon-YYYY" format (e.g. "Jan-2020"):
  cpi_data$date_column <- lubridate::my(cpi_data$date_string) # Or dmy, ymd etc.
  as per actual format [141]

  # Convert to yearmon object for monthly aggregation/joining
  cpi_data$year_month <- zoo::as.yearmon(cpi_data$date_column)
  # [142]

  # Example for exchange_rate_data, assuming a similar date column
  exchange_rate_data$date_column <-
  lubridate::ymd(exchange_rate_data$date_string_excel) # Adjust parser as
  needed
  exchange_rate_data$year_month <-
  zoo::as.yearmon(exchange_rate_data$date_column)
  ```

- Creating Time Series (ts) Objects:
  For use with the forecast package, data needs to be in ts format.
  Code snippet
  ```
  # Ensure data is sorted by date first
  cpi_data <- cpi_data %>% dplyr::arrange(year_month)

  # Calculate year-on-year inflation if CPI is in levels
  # cpi_data <- cpi_data %>%
  #   dplyr::mutate(inflation_yoy = (cpi_value / dplyr::lag(cpi_value, 12) - 1) * 100)

  # Create a ts object for the inflation rate
  # Determine the start year and month from your data
  start_year <- lubridate::year(min(cpi_data$date_column, na.rm = TRUE))
  start_month <- lubridate::month(min(cpi_data$date_column, na.rm = TRUE))
  ```

```
inflation_ts <- ts(cpi_data$inflation_yoy, # Or the column containing already calculated inflation
            start = c(start_year, start_month),
            frequency = 12) # frequency=12 for monthly data
# [146]
```

- Merging Multiple Datasets:
  Often, predictor variables will be in separate files and need to be merged into a single analytical dataset.
  Code snippet
  ```
  # Assume 'm1_data', 'oil_price_data' are already loaded and have 'year_month' column
  # Ensure all data frames are prepared similarly with a 'year_month' column

  list_of_dfs <- list(cpi_data, exchange_rate_data, m1_data, oil_price_data) # Add all relevant dataframes

  # Use purrr::reduce with dplyr::left_join (or inner_join) to merge all dataframes
  # by the common 'year_month' column
  analytical_data <- list_of_dfs %>%
    purrr::reduce(dplyr::left_join, by = "year_month")
  # [140, 145]

  # Ensure the final dataset is sorted by time
  analytical_data <- analytical_data %>% dplyr::arrange(year_month)
  ```

Initial Data Integrity Checks

After loading and merging, perform basic checks:

- summary(analytical_data): Provides descriptive statistics for each variable, including NA counts.
- str(analytical_data): Shows the structure of the data frame, including data types of columns.
- Visually inspect the first and last few rows (head(analytical_data), tail(analytical_data)).
- Verify that date ranges are consistent and that merging did not create unexpected duplicate rows or an excessive number of NAs due to misaligned

dates.

The process of acquiring data from diverse national and international sources, each with potentially different reporting lags, base years, and file formats, represents a significant practical undertaking in any econometric project. Automating this process as much as possible using R scripts is crucial for ensuring reproducibility and efficiency. A well-documented R script, perhaps modularized for downloading and processing each key variable if the steps are complex, is highly recommended. Furthermore, employing version control systems like Git for both the data (if feasible, e.g., for smaller CSVs) and the R scripts is a best practice in academic research to track changes and collaborate effectively.

When working with time series in R, researchers often choose between the base ts objects and more flexible structures like those provided by the zoo or xts packages. While ts objects are fundamental and directly compatible with many functions in the forecast package [146], zoo and xts offer enhanced capabilities for handling irregular dates, more complex indexing, and merging operations.[142] For this project, which involves monthly data, it is often practical to perform initial data cleaning, date parsing (e.g., using lubridate::ymd then zoo::as.yearmon), and merging using dplyr and zoo's yearmon class for robust date alignment. Once the analytical dataset is prepared, relevant columns can be converted into ts objects for use with ARIMA modeling functions in the forecast package. This approach leverages the strengths of different packages at appropriate stages of the workflow.

Part II: Data Pre-processing and Exploratory Analysis in R

Section 4: Addressing Data Imperfections: Missing Values and Outliers

Economic time series are frequently imperfect, containing missing observations or outliers that can distort analysis and model estimation if not appropriately handled.

Detecting Missing Data Patterns

The first step is to identify the extent and pattern of missing data.

- A quick overview can be obtained using summary(your_data_frame), which will list NA counts for each variable.
- For a more visual understanding, packages like VIM offer functions to plot missing data patterns. Custom plots using ggplot2 can also be created to show missingness over time or across variables.
- The imputeTS package provides specialized tools for time series, such as plotNA.distribution(your_ts_object) to visualize the distribution of NAs, and

statsNA(your_ts_object) to get detailed statistics on missing values, including the longest run of NAs.[147]

Time-Series Specific Imputation Methods in R

Once missing values are identified, they must be imputed. The choice of method should be guided by the nature of the time series and the extent of missingness.

- **Last Observation Carried Forward (LOCF) / Next Observation Carried Backward (NOCB):**
  - Implemented using zoo::na.locf(object, na.rm = TRUE/FALSE, fromLast = FALSE/TRUE).[143]
  - na.rm = TRUE removes leading NAs (or trailing NAs if fromLast = TRUE). If na.rm = FALSE (default for na.locf0), leading/trailing NAs are not filled.
  - fromLast = TRUE carries the next observation backward.
  - The maxgap argument can limit imputation to gaps of a certain size.
  - na.locf0 is a faster, more limited version, implicitly using na.rm=FALSE.
  - This method is simple but assumes the value remains constant, which may not hold for volatile series.

  Code snippet
  ```
  # Example using zoo::na.locf
  # Assuming 'inflation_ts' is a ts object with NAs
  # Carry forward last observation
  # inflation_ts_locf <- zoo::na.locf(inflation_ts, na.rm = FALSE)
  # Carry backward next observation
  # inflation_ts_nocb <- zoo::na.locf(inflation_ts, na.rm = FALSE, fromLast = TRUE)
  ```

- **Linear Interpolation:**
  - Implemented using zoo::na.approx(object, na.rm = TRUE) or imputeTS::na_interpolation(x).[147]
  - This method fills NAs by drawing a straight line between the last observed point before the NA and the first observed point after it. It is suitable for variables that exhibit relatively smooth trends.

  Code snippet
  ```
  # Example using imputeTS::na_interpolation
  # inflation_ts_interp <- imputeTS::na_interpolation(inflation_ts)
  ```

- **Seasonal Decomposition based Imputation:**
  - For time series with clear seasonality, methods like imputeTS::na_seadec(x, algorithm = "interpolation") or imputeTS::na_seasplit(x, algorithm = "interpolation") can be very effective.[147] These methods first decompose the series into seasonal, trend, and remainder components, impute missing values

in the components (often the seasonally adjusted series), and then recombine them.

- **Kalman Smoothing:**
    - A more sophisticated approach using state-space models, implemented as imputeTS::na_kalman(x, model = "auto.arima") or imputeTS::na_kalman(x, model = "StructTS").[147] This method uses the entire observed series to estimate the missing values based on the underlying ARIMA or structural model structure.
- **Multiple Imputation (mice package):**
    - The mice package in R provides robust methods for multiple imputation.[148] However, it is primarily designed for cross-sectional data. Applying mice directly to time series data requires careful consideration to preserve temporal dependencies and autocorrelation structures. While powerful, for this specific project focusing on univariate inflation forecasting with standard time series techniques, simpler time-series specific methods from zoo or imputeTS are likely more appropriate and straightforward to implement correctly. mice could be mentioned as an advanced option if predictors also have substantial missing data and a multivariate imputation strategy is deemed necessary, but with a caution about its standard application to time series.

The choice of imputation method is not trivial and can influence subsequent modeling results. For economic time series, simple methods like LOCF or linear interpolation are often used due to their simplicity and interpretability. However, for series with strong trends or seasonality, more advanced methods like those available in the imputeTS package may yield more accurate imputations.[147] It is good practice to justify the chosen imputation method and, if missing data is substantial, to perform sensitivity analysis by comparing results using different imputation techniques. Documenting the extent of missing data and the specific imputation method employed is crucial for academic transparency and reproducibility.

Outlier Detection and Treatment

Outliers, or extreme observations, can disproportionately influence model estimation and forecast accuracy.

- **Detection:**
    - Visual inspection of time series plots and boxplots can often reveal potential outliers.
    - Statistical methods can also be employed.
- **Treatment in R:**

- The forecast::tsclean(series) function is a convenient tool that identifies and replaces outliers (and missing values) in a time series using robust statistical methods, typically by fitting a robust STL decomposition and replacing extreme residuals.[149]

  Code snippet
  ```
  # Example using forecast::tsclean
  # cleaned_inflation_ts <- forecast::tsclean(inflation_ts_with_NAs_or_outliers)
  ```

- It is important to document which observations were identified as outliers and how they were treated.

Outliers can significantly affect ARIMA model parameter estimates and error variance, and can also unduly influence the loss function minimized by penalized regressions.[14] Therefore, identifying and appropriately handling outliers is a critical pre-processing step. The forecast::tsclean() function offers a practical approach, but the researcher should be aware of the underlying mechanism and consider the economic context of any identified outliers before automatic replacement.

Section 5: Preparing Time Series for Modeling: Transformations and Stationarity

Before fitting time series models, especially ARIMA models, it is essential to visualize the data, stabilize its variance if necessary, and ensure stationarity.

Visualizing Time Series Data

Visual inspection is the first step in understanding the characteristics of the time series.

- Base R's plot() function can be used for a simple time series plot.
- The forecast package offers enhanced visualization tools:
  - forecast::ggtsdisplay(series) is highly recommended as it conveniently displays the time series plot, Autocorrelation Function (ACF) plot, and Partial Autocorrelation Function (PACF) plot in a single figure.[149]
  - forecast::ggseasonplot(series) creates a seasonal plot, overlaying cycles.
  - forecast::ggsubseriesplot(series) plots the data for each season separately. These plots help identify trends, seasonality, changes in variance, and potential outliers.[151]

Variance Stabilization

If the variance of the time series changes with its level (heteroscedasticity), a transformation may be necessary.

- **Log Transformation:** log_series <- log(series). This is commonly used when the variance appears to increase proportionally with the mean.[151] This is only applicable if all values are positive.
- **Box-Cox Transformation:** forecast::BoxCox(series, lambda) provides a more general family of power transformations, where lambda is the transformation parameter. forecast::BoxCox.lambda(series) can be used to find an optimal lambda.

The Concept of Stationarity

A time series is stationary if its statistical properties—such as mean, variance, and autocorrelation structure—are constant over time.[13] This is a fundamental assumption for ARIMA modeling. Non-stationary series often exhibit clear trends (systematic upward or downward movements) or seasonality (patterns that repeat over a fixed period).[152]

Stationarity Testing in R

Formal statistical tests are used to assess stationarity:

- **Augmented Dickey-Fuller (ADF) Test:**
  - *Packages:* tseries (function: adf.test) [156] or urca (function: ur.df).
  - *Null Hypothesis (H0):* The series has a unit root (i.e., it is non-stationary).[152]
  - *Alternative Hypothesis (H1):* The series is stationary.[152]
  - *Interpretation:* A small p-value (typically < 0.05) leads to the rejection of H0, suggesting the series is stationary.[153]
  - *R Implementation (tseries):*

    Code snippet
    ```
    # adf_result <- tseries::adf.test(your_series, alternative = "stationary", k = trunc((length(your_series)-1)^(1/3)))
    # 'k' is the number of lags; a common rule of thumb is trunc((T-1)^(1/3))
    # Or k can be selected based on AIC/BIC from auxiliary regressions.
    ```

- **Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test:**
  - *Package:* tseries (function: kpss.test).[156]
  - *Null Hypothesis (H0):* The series is level stationary or trend stationary (depending on the null argument).[156]
  - *Alternative Hypothesis (H1):* The series has a unit root (is non-stationary).[156]
  - *Interpretation:* A small p-value (typically < 0.05) leads to the rejection of H0, suggesting the series is non-stationary.[158]
  - *R Implementation (tseries):*

    Code snippet

```
# kpss_level_result <- tseries::kpss.test(your_series, null = "Level") # Tests for level stationarity
# kpss_trend_result <- tseries::kpss.test(your_series, null = "Trend") # Tests for trend stationarity
# The 'lshort' argument (TRUE/FALSE) controls the truncation lag parameter.[158]
```

- **Interpreting Conflicting Results:** Sometimes, ADF and KPSS tests yield conflicting results.[157] For example, if ADF suggests stationarity (rejects unit root) while KPSS suggests non-stationarity (rejects stationarity around a trend), the series might be trend-stationary, and detrending (or differencing if the trend is stochastic) would be needed. If ADF fails to reject non-stationarity and KPSS fails to reject stationarity, the evidence is ambiguous, but differencing is often a safer course for ARIMA modeling.
- **Automated Differencing Order Selection:**
  - forecast::ndiffs(series, test="kpss", alpha=0.05): Suggests the number of regular (non-seasonal) differences required to make the series stationary.[149]
  - forecast::nsdiffs(series, m=12, test="ocsb"): Suggests the number of seasonal differences needed for a seasonally stationary series (for monthly data, m=12).[149]

The choice of k (number of lags) in the ADF test and the specification of the null hypothesis (level or trend stationarity) in the KPSS test can influence the test outcomes.[156] It is important to understand these parameters. For ADF lags, information criteria (AIC/BIC) applied to the test regression can guide selection, or common rules of thumb can be used. Functions like forecast::ndiffs incorporate logic for selecting these test parameters.

Achieving Stationarity Through Differencing

If a series is found to be non-stationary, differencing is applied to make it stationary.

- **Non-seasonal Differencing (Order d):** This removes trends.
  Code snippet
  ```
  # differenced_series <- diff(your_series, lag = 1, differences = d)
  # 'd' is typically 1 or 2.
  ```
  151
- **Seasonal Differencing (Order D):** This removes seasonality. For a seasonal period m (e.g., 12 for monthly data):
  Code snippet
  ```
  # seasonally_differenced_series <- diff(your_series, lag = m, differences = D)
  ```

'D' is typically 0 or 1.
[154] It is uncommon to require more than two non-seasonal differences (d=2) or more than one seasonal difference (D=1).[151] Over-differencing can introduce spurious correlations and complicate model identification.[151] After differencing, the ADF and KPSS tests should be reapplied to confirm stationarity.

While differencing is crucial for preparing data for ARIMA models, it's important to note that penalized regression models (Lasso, Ridge, Elastic Net) are typically applied to the original (or log-transformed) levels of the series, along with their lagged values. The explicit requirement for stationarity is primarily associated with the ARIMA framework. Highly trended predictors in regression can sometimes lead to issues if not carefully managed (e.g., by including trend terms or ensuring cointegrating relationships if theoretically appropriate), but differencing is not a standard prerequisite for these models as it is for ARIMA. The data preparation pipeline will thus diverge: differenced series for ARIMA, and level (or log-transformed and then scaled) series for constructing the predictor matrix for penalized regressions.

Table: Stationarity Test Results and Differencing Orders

This table provides a systematic record of the stationarity testing process, justifying the choice of differencing orders (d and D) for the ARIMA models.

Variable Name	**Test on Original Series (ADF: stat, p-val, lags)**	**Test on Original Series (KPSS: stat, p-val, type)**	**Chosen d**	**Test on d-Differenced (ADF)**	**Test on d-Differenced (KPSS)**	**Chosen D**	**Test on d,D-Differenced (ADF)**	**Test on d,D-Differenced (KPSS)**	**Final Stationarity Assessment**
Pakistan CPI Inflation	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*
Predictor 1 (e.g.,	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*

M2)									
... *(other predictors)*	...	...	...	...	...	...	...	...	...

(Note: This table should be populated with actual results from R analysis.)

Section 6: Feature Scaling for Regularized Regression Models

Penalized regression techniques like Lasso, Ridge, and Elastic Net require careful preprocessing of predictor variables, with feature scaling being a particularly important step.

Rationale for Feature Scaling

The core idea of penalized regression is to add a penalty term to the ordinary least squares (OLS) loss function, where this penalty is a function of the magnitude of the regression coefficients.[161]

- **Uniform Penalty Application:** If predictor variables are measured on vastly different scales (e.g., one variable ranging from 0-1 and another from 1,000-100,000), the penalty term will disproportionately affect the coefficients of variables with larger absolute scales. This can lead to a situation where the model shrinks the coefficients of large-scale variables more aggressively, irrespective of their true predictive power, simply because their coefficients need to be smaller to achieve the same effect on the response variable. Scaling ensures that the penalty is applied more uniformly, allowing the regularization process to identify important variables based on their relationship with the response, not their native scale.[163]
- **Model Stability and Convergence:** Optimization algorithms used to fit penalized regression models (like coordinate descent in glmnet) can converge faster and more reliably when features are on a similar scale. Large disparities in feature scales can lead to ill-conditioned optimization problems, slowing down convergence or even preventing the algorithm from finding an optimal solution.[164]

Standardization (Z-score Scaling)

The most common method for feature scaling is standardization. This process transforms each predictor variable to have a mean of 0 and a standard deviation of 1.164 The formula for standardizing a value x of a predictor is:

$z = \sigma x(x - \mu x)$
where μx is the mean of the predictor and σx is its standard deviation.

Implementing Standardization in R

- **Base R scale() function:** The scale(x, center = TRUE, scale = TRUE) function in base R performs standardization. It returns a matrix with the scaled values.
  Code snippet
  ```
  # Assuming 'predictor_matrix' contains only numeric predictors
  # scaled_predictor_matrix <- scale(predictor_matrix)
  ```
- **Using glmnet's Internal Standardization:** The glmnet package provides a standardize = TRUE argument within the glmnet() and cv.glmnet() functions.[161] When set to TRUE (the default for predictors), glmnet internally standardizes the predictor variables before fitting the model. The coefficients are then automatically converted back to the original scale of the predictors when reported or used for prediction. This is generally the recommended approach as it correctly handles standardization within cross-validation loops, preventing data leakage from the test folds into the scaling parameters of the training folds.
  Code snippet
  ```
  # library(glmnet)
  # cv_fit <- cv.glmnet(x_matrix, y_vector, family = "gaussian", alpha = 0.5,
  standardize = TRUE)
  ```
- **Important Note on Training and Test Sets:** If manual scaling is performed (i.e., not using glmnet's internal standardization), it is crucial to calculate the mean and standard deviation *only from the training data*. These training set parameters must then be used to scale both the training data and any subsequent test data. Applying scale() separately to the whole dataset before splitting or to the test set using its own mean/sd would lead to data leakage and biased performance estimates.

While glmnet conveniently handles predictor standardization internally, it's important to recognize that this occurs after the predictor matrix x (which may include lagged variables, interaction terms, etc.) has been constructed. The response variable y (inflation, in this case) is generally not standardized by glmnet by default when family="gaussian", though it can be scaled by the user beforehand if desired. glmnet always returns coefficients on the original scale of y.[161] For most applications, relying on glmnet(..., standardize = TRUE) for predictor scaling is appropriate and robust.

The decision to transform the target variable (inflation) itself is a separate consideration from scaling predictors for regularization. If the inflation series exhibits strong skewness or extreme outliers that might violate assumptions of downstream models (even if glmnet is robust to some deviation from normality for residuals), transformations like a logarithm or Box-Cox (if the series is strictly positive) might be applied to the inflation series *before* any modeling. This is a general data pre-processing step that would apply to all models being compared, not just the penalized regressions.

Part III: Univariate and Exogenous Forecasting: ARIMA, SARIMA, and ARIMAX Models

Section 7: The Box-Jenkins Methodology for ARIMA/SARIMA/ARIMAX

The Box-Jenkins methodology provides a systematic framework for identifying, estimating, and diagnosing Autoregressive Integrated Moving Average (ARIMA) models and their seasonal counterparts (SARIMA) for time series forecasting.[1] These models can also be extended to include exogenous predictor variables (ARIMAX/SARIMAX).

ARIMA(p,d,q) Components

A non-seasonal ARIMA model is characterized by three orders: (p, d, q).

- **AR(p) - Autoregressive component:** This component models the dependency of the current value of the series on its own p previous (lagged) values. Mathematically, an AR(p) process can be represented as: $y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t$ where y_t is the value of the series at time t, c is a constant, ϕ_i are the autoregressive coefficients, and ε_t is a white noise error term.[154]
- **I(d) - Integrated component:** This refers to the number of non-seasonal differences required to make the time series stationary. Differencing helps to remove trends and stabilize the mean of the series.[152] A first difference is $\Delta y_t = y_t - y_{t-1}$.
- **MA(q) - Moving Average component:** This component models the dependency of the current value of the series on q past error terms (residuals). An MA(q) process can be represented as: $y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q}$ where θ_i are the moving average coefficients.[154]

SARIMA(p,d,q)(P,D,Q)m Components

For time series exhibiting seasonality, the SARIMA model extends the ARIMA framework by adding seasonal components.[146] The notation is SARIMA(p,d,q)(P,D,Q)m:

- (p,d,q): These are the non-seasonal AR, differencing, and MA orders, as defined above.
- (P,D,Q): These are the seasonal counterparts:
  - **P (Seasonal AR order):** Number of seasonal autoregressive terms (lags from the same season in previous years).
  - **D (Seasonal Differencing order):** Number of seasonal differences required to remove seasonal patterns. For a seasonal period m, a seasonal difference is $\nabla_m y_t = y_t - y_{t-m}$.
  - **Q (Seasonal MA order):** Number of seasonal moving average terms (past errors from the same season in previous years). [154]
- m: The seasonal period or the number of observations per seasonal cycle (e.g., m=12 for monthly data with annual seasonality).[146]

ARIMAX/SARIMAX Models

ARIMA and SARIMA models can be further extended to include the impact of exogenous variables (predictors), leading to ARIMAX or SARIMAX models. In R's forecast package, this is achieved by supplying a matrix of exogenous regressors to the xreg argument in the Arima() or auto.arima() functions.159 The model then becomes:
$ y'_t = \beta X_t + n_t $
where y_t' is the (possibly differenced) original series, X_t is the vector of exogenous variables at time t, β is the vector of regression coefficients for these variables, and n_t is an ARIMA/SARIMA process for the error term.

Model Assumptions

The validity and effectiveness of ARIMA-type models hinge on several key assumptions [13]:

1. **Stationarity:** The time series (after any necessary differencing) must be stationary. This means its mean, variance, and autocorrelation structure should be constant over time.
2. **Linearity:** ARIMA models assume linear relationships between the current value and past values/past errors.
3. **Residuals as White Noise:** The residuals (the differences between observed values and model fits) should be white noise. This implies they are independent and identically distributed (i.i.d.) with a mean of zero and constant variance.
4. **Normality of Residuals (Approximate):** While not strictly required for parameter estimation, normally distributed residuals are assumed when constructing prediction intervals based on standard statistical theory.

Strengths of ARIMA/SARIMA Models

These models are widely used due to several advantages [17]:

- **Well-Established Methodology:** The Box-Jenkins approach provides a structured process for model building.
- **Flexibility:** Capable of modeling a diverse range of time series patterns, including trends and seasonality.
- **Short-Term Forecasting Performance:** Often provide accurate short-term forecasts.
- **Interpretability:** Model parameters (AR, MA coefficients) can often be interpreted in the context of the time series dynamics.
- **Strong Benchmark:** Frequently serve as a robust benchmark against which more complex models are compared.

Weaknesses/Challenges of ARIMA/SARIMA Models

Despite their strengths, ARIMA-type models have limitations [13]:

- **Stationarity Requirement:** Data must be made stationary, which can sometimes be challenging or involve subjective choices in differencing.
- **Linearity Assumption:** May struggle to capture complex non-linear relationships present in many economic time series.
- **Subjectivity in Identification:** Manual identification of model orders (p,q,P,Q) using ACF/PACF plots can be subjective, though automated functions like auto.arima mitigate this.
- **Sensitivity to Outliers:** Extreme values can disproportionately affect parameter estimation and forecasts.
- **Predicting Turning Points:** Generally poor at forecasting sharp turning points unless these represent a return to a long-run equilibrium.
- **Long-Horizon Forecasts:** Predictive accuracy can degrade significantly as the forecast horizon increases.
- **Estimation Complexity:** Parameter estimation can become complex, especially for models with many parameters or high-order seasonal components.

Given that inflation in Pakistan is likely influenced by identifiable exogenous factors such as international oil prices, exchange rate movements, and domestic monetary conditions (as discussed in Section 2), ARIMAX or SARIMAX models should be strongly considered. These models offer a way to incorporate the predictive information from these external variables directly into the ARIMA framework, potentially leading to more accurate forecasts than purely univariate ARIMA or SARIMA models.[159] A critical aspect of using ARIMAX models for out-of-sample forecasting is the need for future values of the exogenous variables (xreg). If these are not known, they must themselves be forecasted, which introduces an additional layer of uncertainty, or the model must be restricted to using only lagged values of the exogenous predictors.

Seasonality in monthly inflation data can also be complex. Standard SARIMA models with frequency=12 are effective at capturing fixed calendar seasonality (e.g., price changes consistently occurring in specific months each year). However, in Pakistan, significant economic activity and consumption patterns are tied to events like Ramadan and Eid, which follow the lunar calendar and thus "move" across Gregorian calendar months from year to year.[3] Standard SARIMA models may not adequately capture these moving holiday effects. While a full treatment of such effects is advanced, if diagnostic checks on SARIMA residuals reveal patterns correlated with these moving holidays, a potential refinement could involve including dummy variables representing these holiday periods as part of the xreg matrix in a SARIMAX model.

Section 8: Model Identification: ACF and PACF Analysis in R

The model identification stage of the Box-Jenkins methodology aims to determine the appropriate orders for the autoregressive (AR) and moving average (MA) components (i.e., p and q for non-seasonal models, and P and Q for seasonal models). This is primarily done by examining the patterns in the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots of the *stationary* (differenced) time series.[17]

Generating ACF and PACF Plots in R

After ensuring the time series (e.g., inflation rate) has been made stationary through appropriate differencing (as discussed in Section 5), ACF and PACF plots can be generated.

- The forecast::ggtsdisplay(stationary_series, main="ACF/PACF of Stationary Inflation") function is highly recommended. It produces a single figure containing the time plot of the series, its ACF plot, and its PACF plot, facilitating a comprehensive visual analysis.[149]
- Alternatively, base R functions can be used:
  Code snippet
  ```r
  # Assuming 'stationary_inflation' is your stationary time series object
  # acf_plot <- acf(stationary_inflation, lag.max = 36, main = "ACF of Stationary Inflation")
  # pacf_plot <- pacf(stationary_inflation, lag.max = 36, main = "PACF of Stationary Inflation")
  # It's common to examine up to lag 36 or 48 for monthly data to capture seasonal patterns.
  ```

Interpreting ACF and PACF Patterns

The characteristic patterns in the ACF and PACF plots provide clues for the orders of AR and MA terms [151]:

- **Pure Autoregressive (AR(p)) Process:**
  - **PACF:** Shows a significant spike at lag p and cuts off to (statistically insignificant) zero for lags greater than p.
  - **ACF:** Tails off gradually towards zero, often in an exponential decay or damped sinusoidal pattern.
- **Pure Moving Average (MA(q)) Process:**
  - **ACF:** Shows a significant spike at lag q and cuts off to zero for lags greater than q.
  - **PACF:** Tails off gradually towards zero.
- **Mixed Autoregressive Moving Average (ARMA(p,q)) Process:**
  - **ACF:** Tails off gradually towards zero.
  - **PACF:** Tails off gradually towards zero. Identifying the exact orders p and q for ARMA models can be more challenging from the plots alone and often involves some trial and error or reliance on information criteria during model estimation.
- **Seasonal Patterns (SAR(P) or SMA(Q)):**
  - If the data has been seasonally differenced (e.g., diff(series, lag=12)), examine the ACF and PACF plots of this seasonally differenced series.
  - Significant spikes at seasonal lags (e.g., 12, 24, 36 for monthly data) in the PACF suggest a seasonal AR (SAR) component of order P.
  - Significant spikes at seasonal lags in the ACF suggest a seasonal MA (SMA) component of order Q.

Practical Considerations

- **Principle of Parsimony:** Start by considering models with small orders of p, q, P, Q (e.g., 0, 1, or 2). Simpler models are often more robust and easier to interpret.
- **Iterative Process:** Model identification is not a one-off step. Initial guesses for model orders based on ACF/PACF plots may need to be revised after model estimation and diagnostic checking.[17]
- **Automated Identification:** The forecast::auto.arima() function in R automates the process of identifying p, d, q, P, D, Q by minimizing an information criterion (typically AICc) across a range of models, also incorporating unit root tests to determine differencing orders.[159]

While the skill of interpreting ACF and PACF plots is a cornerstone of classical time series analysis, it can become ambiguous for complex series, especially those with mixed ARMA components or strong seasonality.[151] In such cases, auto.arima() provides a more objective and often more efficient starting point for model selection. Nevertheless, understanding the patterns in ACF and PACF plots remains valuable for validating the suggestions from auto.arima() or for manually refining model choices. The recommended workflow is to use ggtsdisplay() on the appropriately differenced series to gain an initial understanding of its correlation structure, then employ auto.arima() as the primary tool for selecting candidate models, and finally, use ACF/PACF plots of the *residuals* from the fitted model for crucial diagnostic checking (see Section 10).

Section 9: Model Estimation and Selection in R

Once candidate ARIMA/SARIMA/ARIMAX model structures have been identified (either manually via ACF/PACF analysis or using auto.arima), the next step is to estimate the model parameters and select the best model based on statistical criteria.

Fitting Models in R

The forecast package in R provides two main functions for fitting ARIMA-type models:

- **forecast::Arima():** This function fits a specified ARIMA model.

  Code snippet

  ```
  # Example for a SARIMA(1,1,1)(1,1,0)12 model with an exogenous regressor
  # Assume 'inflation_ts' is the target time series (already differenced if d/D > 0 in manual specification)
  # Assume 'exog_regressors' is a matrix of exogenous variables aligned with inflation_ts
  # For Arima(), if d or D > 0, the function handles differencing internally based on order argument.
  # fit_manual <- forecast::Arima(inflation_ts,
  #                    order = c(1,1,1), # Non-seasonal (p,d,q)
  #                    seasonal = list(order = c(1,1,0), period = 12), # Seasonal (P,D,Q)m
  #                    include.mean = TRUE, # Or FALSE, Arima often decides based on d
  #                    include.drift = FALSE, # Consider if d=1 and a trend is present
  #                    xreg = exog_regressors)
  # [146, 170]
  ```

Key arguments for Arima():

- order = c(p,d,q): Specifies the non-seasonal orders.
- seasonal = list(order = c(P,D,Q), period = m): Specifies seasonal orders and period.
- include.mean: Controls inclusion of a constant term c. Arima() often determines this automatically based on the order of differencing d.[170] If d>0, a constant implies a polynomial trend.
- include.drift: If d=1, allows for a linear trend term (drift).[170] If d>1, constants/drifts are generally omitted to avoid inducing high-order polynomial trends.
- xreg: A numeric matrix or vector of exogenous regressors. If used, it must have the same number of rows as the series. For forecasting, future values of xreg must be provided to the forecast() function.

* **forecast::auto.arima():** This function automatically selects the "best" ARIMA model according to an information criterion.

Code snippet

```
# Example using auto.arima with exogenous regressors
# fit_auto <- forecast::auto.arima(inflation_ts,
#                          d = NA, # Allows auto.arima to choose d using KPSS test
#                          D = NA, # Allows auto.arima to choose D using OCSB test
#                          max.p = 5, max.q = 5, # Max non-seasonal orders
#                          max.P = 2, max.Q = 2, # Max seasonal orders
#                          stepwise = TRUE,     # Faster search; set to FALSE for more
exhaustive
#                          approximation = FALSE, # Use for more accurate AICc in
exhaustive search
#                          xreg = exog_regressors,
#                          ic = "aicc",        # Information criterion to minimize (aic, aicc,
bic)
#                          test = "kpss",       # Unit root test for d
#                          seasonal.test = "ocsb") # Unit root test for D
# [159, 160, 170]
```

auto.arima uses the Hyndman-Khandakar algorithm, which combines unit root tests (like KPSS for d and OCSB for D) with minimization of an information criterion (default AICc) to select the orders (p,q,P,Q) and estimate parameters.[170] Setting stepwise=FALSE and approximation=FALSE leads to a more comprehensive search but is computationally more intensive.

Maximum Likelihood Estimation (MLE)

Both Arima() and auto.arima() use Maximum Likelihood Estimation (MLE) to estimate the model parameters ($\phi_i, \theta_j, \Phi_k, \Theta_l$, coefficients for xreg, and the residual variance σ^2). MLE finds the parameter values that maximize the likelihood function, which represents the probability of observing the given data under the assumed model structure.[172] For ARIMA models, MLE is numerically similar to minimizing the sum of squared residuals.[172] R reports the log-likelihood of the data for the fitted model.

Information Criteria for Model Selection

When comparing different ARIMA model specifications (i.e., different orders of p, q, P, Q, and inclusion of constant/drift or xreg), information criteria are used to balance model fit with parsimony (simplicity). Lower values generally indicate a better model.

- **AIC (Akaike Information Criterion):** $AIC = -2\log(L) + 2k_{params}$.[17] L is the maximized likelihood, and k_{params} is the total number of estimated parameters (including AR, MA, seasonal AR/MA, xreg coefficients, and σ^2).
- **AICc (Corrected AIC):** A modification of AIC for smaller sample sizes, generally preferred. $AICc = AIC + \frac{2k_{params}(k_{params}+1)}{T - k_{params} - 1}$, where T is the sample size.[172] This is the default for auto.arima.
- **BIC (Bayesian Information Criterion) / SIC (Schwarz Criterion):** $BIC = AIC + (\log(T) - 2)k_{params}$ (or $BIC = -2\log(L) + k_{params}\log(T)$).[17] BIC tends to penalize model complexity more heavily than AIC, often leading to simpler models.
- **HQ (Hannan-Quinn Criterion):** Another criterion sometimes used.[17]

These criteria are primarily used for selecting the orders p, q, P, Q *after* the orders of differencing d and D have been determined, as differencing changes the data used for likelihood computation, making AIC/BIC values non-comparable across different d or D values.[172]

Table: Candidate ARIMA/SARIMA/ARIMAX Model Specifications and Selection Metrics

This table facilitates a systematic comparison of various candidate models, aiding in an evidence-based selection of the final model.

Model Specification (p,d,q)(P,D,Q)m +	Log-Likelihood	AIC	AICc	BIC	Residual Variance (σ^2)	Notes (e.g., Significant Coefficie

xreg?						**nts)**
e.g., ARIMA(1,1,1)	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*
e.g., SARIMA(0,1,1)(0,1,1)12	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*
e.g., auto.arima selected model	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*
e.g., ARIMAX with oil price lag	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*	*(fill)*
... (other candidates)	...	...	...	...	...	...

(Note: This table should be populated with actual results from R analysis using summary(fit_object) or accessing components like fit_object$aic.)

While auto.arima is a powerful and convenient tool, it's not infallible.[170] Its stepwise search (if stepwise=TRUE) might not find the global optimum AICc value. Therefore, it is good practice to use auto.arima as a strong starting point but also to consider fitting a few other theoretically plausible models (e.g., based on ACF/PACF insights or economic theory) and compare them using information criteria and thorough diagnostic checks. For final candidate models, running auto.arima with stepwise=FALSE, approximation=FALSE can provide a more robust selection if computationally feasible.

Section 10: Diagnostic Checking of ARIMA/SARIMA/ARIMAX Models in R

After a candidate ARIMA-type model has been estimated, rigorous diagnostic checking is essential to ensure its adequacy. The primary goal is to verify that the model has captured all systematic information in the time series, leaving behind residuals that resemble white noise.[17]

Analyzing Model Residuals

The residuals of a fitted model are obtained in R using residuals(fit_object).

- **Time Plot of Residuals:** A plot of the residuals over time should show no discernible patterns, trends, or obvious changes in variance. The residuals should appear to be randomly scattered around zero.[176]
- **ACF/PACF of Residuals:** The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) of the residuals should show no significant spikes. Any remaining significant autocorrelations suggest that the model has not fully captured the temporal dependencies in the data.[151]
  - The forecast::ggtsdisplay(residuals(fit_object), main="Residual Diagnostics") function is excellent for this, as it plots the residuals, their ACF, and their PACF.
- **Ljung-Box Test:** This is a formal statistical test for autocorrelation in the residuals.
  - *Function:* Box.test(residuals(fit_object), lag = h, fitdf = k, type = "Ljung-Box")
  - *Null Hypothesis (H0):* The residuals are independently distributed (i.e., they are white noise).[175]
  - lag: The number of lags up to which autocorrelation is tested. Common choices include ln(T) (where T is the number of observations) or, for seasonal data, 2m (where m is the seasonal period, e.g., $2\times12=24$ for monthly data) [[175] suggests $h\approx\max(10,n)$].
  - fitdf: This is a **critical** argument. It specifies the number of parameters estimated in the ARIMA model (sum of AR, MA, seasonal AR, and seasonal MA parameters: $p+q+P+Q$). If exogenous regressors (xreg) were included, their count should also be included if the test version accounts for them, though standard Box.test fitdf is for ARMA parameters. The degrees of freedom for the test statistic are $h-\text{fitdf}$.[170]
  - *Interpretation:* A high p-value (typically > 0.05) means we fail to reject H0, indicating that there is no significant evidence of autocorrelation in the residuals, and they appear to be white noise.[175]
- **forecast::checkresiduals(fit_object):** This is a very convenient wrapper function. It produces a time plot of residuals, their ACF plot, a histogram of residuals with an overlaid normal curve, and performs the Ljung-Box test (correctly setting fitdf).[149]

  Code snippet
  ```
  # Assuming 'fit_arima' is your fitted Arima object
  # forecast::checkresiduals(fit_arima)
  ```

It is crucial to correctly specify the fitdf argument when using Box.test directly on residuals from an ARIMA model. If fitdf is omitted or set incorrectly (e.g., to 0, which is the default for a raw series), the p-values from the Ljung-Box test will be inaccurate, potentially leading to erroneous conclusions about the adequacy of the model.[175] The forecast::checkresiduals() function automatically handles the correct specification of degrees of freedom for the Ljung-Box test based on the fitted Arima object.

Checking for Normality of Residuals

While not strictly necessary for the validity of point forecasts from ARIMA models, the assumption of normally distributed residuals is important for constructing prediction intervals that rely on this assumption.

- **Histogram of Residuals:** hist(residuals(fit_object)) or forecast::gghistogram(residuals(fit_object)) can provide a visual check for a bell-shaped distribution.
- **Normal Q-Q Plot:** qqnorm(residuals(fit_object)); qqline(residuals(fit_object)) helps assess if the quantiles of the residuals match the quantiles of a normal distribution (points should fall approximately on a straight line).
- **Formal Normality Tests:** Tests like the Jarque-Bera test (e.g., from tseries::jarque.bera.test) can be used, though they can be sensitive in large samples.[17] checkresiduals() often reports a normality test result.

If residuals are found to be non-normal, the point forecasts from the ARIMA model are still generally considered consistent. However, the prediction intervals generated by default (which often assume normality) may not have the stated coverage probability (e.g., a 95% interval may not actually contain the true future value 95% of the time).[154] In such cases, bootstrap-based prediction intervals, which can be generated by setting bootstrap=TRUE in the forecast() function, might provide more reliable interval estimates, albeit at a higher computational cost.

Iterative Model Refinement

Diagnostic checking is part of an iterative model-building process. If the diagnostics indicate problems (e.g., significant autocorrelation in residuals, patterns in the residual plot), the model specification should be revised. This might involve changing the orders (p,d,q,P,D,Q), considering different transformations of the data, or adding/modifying exogenous variables in an ARIMAX framework. The cycle of identification, estimation, and diagnostic checking is repeated until a satisfactory model is found.[17]

Section 11: Generating Forecasts with ARIMA/SARIMA/ARIMAX in R

Once a satisfactory ARIMA-type model has been identified, estimated, and diagnostically checked, it can be used to generate forecasts for future periods.

Using forecast::forecast()

The primary function for generating forecasts from models fitted with Arima() or auto.arima() is forecast::forecast().[146]

Code snippet

```
# Assuming 'final_arima_model' is the selected and validated Arima object
# And 'h_periods' is the number of months to forecast ahead
# And 'future_exog_regressors' is a matrix of future values for xreg variables (if used in the model)

# forecast_results <- forecast::forecast(final_arima_model,
#                                        h = h_periods,
#                                        xreg = future_exog_regressors, # Omit if no xreg in model
#                                        level = c(80, 95)) # Desired confidence levels for prediction intervals
```

Key arguments:

- object: The fitted model object (e.g., from Arima() or auto.arima()).
- h: The number of periods ahead to forecast.
- xreg: If the model included exogenous variables (xreg), this argument must be supplied with a matrix containing the future values of those exogenous variables for the forecast horizon. The matrix must have h rows and the same number of columns as the original xreg matrix used for estimation.
- level: A vector of confidence levels (e.g., c(80, 95)) for which prediction intervals should be calculated.

Output Object

The forecast() function returns an object of class forecast, which is a list containing various useful components:

- mean: The point forecasts for each period in the forecast horizon.
- lower: A matrix containing the lower bounds of the prediction intervals for each specified confidence level.
- upper: A matrix containing the upper bounds of the prediction intervals.
- level: The confidence levels used for the prediction intervals.
- x: The original time series data.
- residuals: The residuals from the fitted model on the training data.
- fitted: The fitted values on the training data.
- model: Details of the fitted ARIMA model.

Visualizing Forecasts

The forecast package provides convenient plotting functions:

- plot(forecast_results): Generates a base R plot of the historical data, point forecasts, and prediction intervals.
- autoplot(forecast_results): Generates a similar plot using ggplot2 for enhanced aesthetics.[146]
  Code snippet
  ```
  # library(ggplot2)
  # autoplot(forecast_results) +
  #   ggtitle("Inflation Forecast for Pakistan") +
  #   xlab("Year") +
  #   ylab("Inflation Rate (%)") +
  #   theme_minimal()
  ```

A critical practical challenge when forecasting with ARIMAX models is the requirement for future values of the exogenous variables (xreg).[146] If the model includes, for example, future oil prices or exchange rates as predictors, these future values must be known or forecasted separately to generate the inflation forecast. If actual future values of xreg are unavailable at the time of forecasting (which is typical), several strategies can be employed:

1. **Use Lagged Exogenous Variables Only:** Restrict the xreg matrix to only include lagged values of exogenous variables. This means their impact on inflation is only through their past values, and no future values are needed for forecasting inflation.
2. **Scenario-Based Forecasts:** Assume specific future paths for the exogenous variables (e.g., oil price remains constant, increases by X%, or follows a path from

an external institution's forecast). The inflation forecast will then be conditional on these scenarios.

3. **Forecast Exogenous Variables Separately:** Develop separate time series models (e.g., ARIMA or other methods) for each exogenous variable in xreg. Use these models to forecast the future values of xreg, and then plug these forecasts into the ARIMAX model for inflation. This approach introduces additional forecast uncertainty from the xreg forecasts into the final inflation forecast. Ideally, this uncertainty should be incorporated into the prediction intervals for inflation, though this is a complex task often requiring simulation methods.

The choice of strategy for handling future_xreg should be clearly documented, as it significantly impacts the nature and interpretation of the out-of-sample forecasts.

Part IV: Forecasting with Penalized Regression Techniques

Penalized regression methods, including Lasso, Ridge, and Elastic Net, offer an alternative approach to time series forecasting, particularly when dealing with a potentially large number of predictor variables and issues like multicollinearity.

Section 12: Introduction to Lasso, Ridge, and Elastic Net Regression

These methods are extensions of ordinary least squares (OLS) linear regression, designed to improve model performance by adding a penalty term to the sum of squared residuals (RSS).[178] This penalty shrinks the regression coefficients, helping to prevent overfitting and manage multicollinearity.

Linear Regression Foundation and Its Challenges

Standard linear regression aims to find coefficients (β) that minimize the RSS:
$RSS = \sum_{i=1}^{N} (y_i - X_i\beta)^2$
However, when the number of predictors (p) is large relative to the number of observations (N), or when predictors are highly correlated (multicollinearity), OLS estimates can have high variance, leading to poor out-of-sample predictive performance (overfitting).162

The Principle of Regularization

Penalized regression addresses these issues by adding a penalty term to the RSS, which discourages overly complex models by shrinking the coefficient estimates.161 The objective function becomes:
$\min_{\beta} \{ RSS + \lambda \cdot P(\beta) \}$
where $\lambda \geq 0$ is a tuning parameter controlling the strength of the penalty $P(\beta)$.

Ridge Regression (L2 Penalty)

Ridge regression uses an L2 penalty, which is the sum of the squares of the coefficients:
$P(\beta) = \sum_{j=1}^{p} \beta_j^2 = ||\beta||_2^2$

The objective function is: $\min_{\beta}\{\sum_{i=1}^{N}(y_i - X_i\beta)^2 + \lambda\sum_{j=1}^{p}\beta_j^2\}$.[161]

- **Shrinkage:** Ridge regression shrinks coefficients towards zero as λ increases. However, it rarely sets coefficients exactly to zero, meaning all predictors are typically retained in the model.[161]
- **Multicollinearity:** It is particularly effective when predictors are highly correlated, as it tends to shrink the coefficients of correlated variables towards each other, stabilizing their estimates.[161]

Lasso Regression (L1 Penalty)

Lasso (Least Absolute Shrinkage and Selection Operator) regression uses an L1 penalty, which is the sum of the absolute values of the coefficients:
$P(\beta) = \sum_{j=1}^{p} |\beta_j| = ||\beta||_1$
The objective function is: $\min_{\beta}\{\sum_{i=1}^{N}(y_i - X_i\beta)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\}$.[161]

- **Shrinkage and Variable Selection:** Lasso performs both shrinkage and automatic variable selection. As λ increases, some coefficients are shrunk to exactly zero, effectively removing those predictors from the model.[2] This makes Lasso useful for high-dimensional datasets where many predictors might be irrelevant or redundant.
- **Correlated Predictors:** Lasso can be somewhat unstable when predictors are highly correlated; it tends to arbitrarily select one variable from a group of correlated predictors and zero out the others.[162]

Elastic Net Regression (Combination of L1 and L2 Penalties)

Elastic Net regression combines both L1 and L2 penalties to leverage the strengths of both Ridge and Lasso. The penalty term is:
$P(\beta) = \alpha \sum_{j=1}^{p} |\beta_j| + \frac{(1-\alpha)}{2} \sum_{j=1}^{p} \beta_j^2$
The objective function is: $\min_{\beta}\{\sum_{i=1}^{N}(y_i - X_i\beta)^2 + \lambda(\alpha\sum_{j=1}^{p}|\beta_j| + \frac{(1-\alpha)}{2}\sum_{j=1}^{p}\beta_j^2)\}$.[161]

- **Mixing Parameter α:** The parameter $\alpha \in$ controls the mix between Ridge (α=0) and Lasso (α=1).[161]
- **Benefits:** Elastic Net can perform variable selection like Lasso while retaining the stability of Ridge regression in the presence of highly correlated predictors. It often exhibits a "grouping effect," where strongly correlated predictors tend to be selected or removed together.[161]

Benefits of Penalized Regression

These methods offer several advantages, especially in macroeconomic forecasting where datasets can be rich with predictors [161]:

- Improved prediction accuracy by reducing overfitting.
- Ability to handle multicollinearity among predictors.

- Automatic variable selection (Lasso and Elastic Net), leading to more parsimonious and interpretable models.
- Capability to work in high-dimensional settings where the number of predictors (p) may exceed the number of observations (N).

Assumptions and Considerations

- **Linearity:** Like OLS, these methods fundamentally assume a linear relationship between predictors and the response. This can be relaxed by including non-linear transformations of predictors (e.g., polynomials, interaction terms) in the predictor matrix.
- **Feature Scaling:** As discussed in Section 6, standardizing predictors is crucial for penalized regression methods to ensure that the penalty is applied fairly across all coefficients.[182]

Given the potential for numerous predictors in macroeconomic forecasting (including various lags of inflation and exogenous variables), Elastic Net often emerges as a good default choice. It provides a balance between the variable selection capability of Lasso and the multicollinearity handling of Ridge.[161] The process of tuning both λ (overall penalty strength) and α (Lasso/Ridge mix) via cross-validation becomes a key part of the modeling workflow.

It is also important to recognize that while Lasso and Elastic Net perform variable selection, this selection is data-driven and aimed at optimizing predictive performance under the given penalty structure. The set of selected variables may not always perfectly align with the "true" causal drivers of inflation, especially if predictors are highly correlated or the sample size is limited relative to the number of potential predictors.[162] Therefore, while the selected variables can offer valuable insights, their importance should be interpreted with caution and ideally corroborated by economic theory or further analysis.

Section 13: Implementing Penalized Regressions with glmnet in R

The glmnet package in R is a powerful and efficient tool for fitting Lasso, Ridge, and Elastic Net models.[161] It uses a cyclical coordinate descent algorithm that is particularly fast for large datasets and can handle sparse predictor matrices.

Data Preparation

1. **Predictor Matrix (x):** This matrix will contain the predictor variables. For time series forecasting of inflation, x will typically include:
   - Lagged values of the inflation series itself (to capture autoregressive

dynamics).
    - Current and/or lagged values of relevant exogenous variables (e.g., oil prices, exchange rates, money supply, industrial production index). The model.matrix() function can be useful for creating dummy variables from any categorical predictors and for generating interaction terms, if desired. However, the glmnetUtils package provides a formula interface that can simplify this step.[188]
2. **Response Vector (y):** This vector will contain the inflation series that is being predicted. It should be aligned with the predictor matrix x (i.e., y_t is regressed on predictors available at or before time t-1 for one-step-ahead forecasting).
3. Ensure that the rows in x and y used for training have no missing values.

Using glmnet::glmnet()

The core function for fitting the models is glmnet():

Code snippet

```
# library(glmnet)

# Assume 'x_train' is the training predictor matrix and 'y_train' is the training response vector
# For Lasso (alpha = 1)
# lasso_fit <- glmnet::glmnet(x_train, y_train, family = "gaussian", alpha = 1, standardize = TRUE)

# For Ridge (alpha = 0)
# ridge_fit <- glmnet::glmnet(x_train, y_train, family = "gaussian", alpha = 0, standardize = TRUE)

# For Elastic Net (e.g., alpha = 0.5)
# enet_fit <- glmnet::glmnet(x_train, y_train, family = "gaussian", alpha = 0.5, standardize = TRUE)
```

161

- family = "gaussian": Specifies a linear regression model for a continuous response variable like inflation.

- alpha: Controls the elastic net mixing parameter (1 for Lasso, 0 for Ridge, 0-1 for Elastic Net).
- lambda: An optional sequence of lambda (penalty strength) values. If not provided, glmnet generates a sequence of nlambda (default 100) values based on the data.[161]
- standardize = TRUE: Default and generally recommended. It standardizes the predictor variables before fitting. Coefficients are always returned on their original scale.

Cross-Validation with glmnet::cv.glmnet()

To select the optimal tuning parameter λ, k-fold cross-validation is used via cv.glmnet():

Code snippet

```
# For Lasso
# cv_lasso_fit <- glmnet::cv.glmnet(x_train, y_train, family = "gaussian", alpha = 1,
#                               nfolds = 10, type.measure = "mse")

# For Ridge
# cv_ridge_fit <- glmnet::cv.glmnet(x_train, y_train, family = "gaussian", alpha = 0,
#                               nfolds = 10, type.measure = "mse")

# For Elastic Net (fixed alpha, e.g., 0.5)
# cv_enet_fit_alpha0.5 <- glmnet::cv.glmnet(x_train, y_train, family = "gaussian", alpha
= 0.5,
#                                    nfolds = 10, type.measure = "mse")
```

161

- nfolds: Number of cross-validation folds (e.g., 10 is common).
- type.measure: The loss function to use for evaluating performance during cross-validation (e.g., "mse" for mean squared error, "mae" for mean absolute error). The function returns a cv.glmnet object. Plotting this object (plot(cv_fit_object)) visualizes the cross-validated error (e.g., MSE) against log(λ), helping to identify optimal λ values.[161]

Selecting Optimal λ

The cv.glmnet object stores two important λ values:

- lambda.min: The value of λ that gives the minimum mean cross-validated error.[161]
- lambda.1se: The largest value of λ such that the cross-validated error is within one standard error of the minimum. This often results in a more parsimonious (simpler) model with similar predictive performance to the lambda.min model and is frequently preferred.[161]

Strategy for Tuning α in Elastic Net

cv.glmnet() tunes λ for a *fixed* value of α. To find the optimal combination of both α and λ for an Elastic Net model, one typically performs a grid search:

1. Define a sequence of α values to test (e.g., alphas_to_try <- seq(0, 1, by = 0.1)).
2. Loop through each α in this sequence.
3. Inside the loop, run cv.glmnet() with the current α.
4. Store the minimum cross-validated error (e.g., cv_model$cvm[which(cv_model$lambda == cv_model$lambda.1se)]) and the corresponding lambda.1se (or lambda.min) for each α.
5. Select the (α,λ) pair that yields the overall minimum cross-validated error. [185] This two-dimensional tuning can also be facilitated by packages like caret.

Extracting Coefficients

Coefficients for a specific λ can be extracted using the coef() method:

Code snippet

```
# For a model fit with glmnet()
# coef_at_lambda <- coef(lasso_fit, s = 0.05) # s is the lambda value

# For a model fit with cv.glmnet()
# coef_lambda_min <- coef(cv_lasso_fit, s = "lambda.min")
# coef_lambda_1se <- coef(cv_lasso_fit, s = "lambda.1se")
```

161

Making Predictions

Predictions on new data are made using the predict() method:

Code snippet

```
# Assume 'x_test' is the predictor matrix for the test set
# For a model fit with glmnet()
# predictions_glmnet <- predict(lasso_fit, newx = x_test, s = 0.05)

# For a model fit with cv.glmnet()
# predictions_cv_min <- predict(cv_lasso_fit, newx = x_test, s = "lambda.min")
# predictions_cv_1se <- predict(cv_lasso_fit, newx = x_test, s = "lambda.1se")
```

161

Table: Penalized Regression Hyperparameter Tuning Summary

This table is crucial for outlining the optimal hyperparameters found for each penalized regression model, ensuring reproducibility and providing insight into model complexity.

Model Type	**Alpha (α) (for Elastic Net)**	**Optimal Lambda (λ) (e.g., lambda.1se)**	**CV Error (MSE/MAE) at Optimal Lambda**	**Number of Non-Zero Predictors**
Lasso	1 (fixed)	*(fill with result)*	*(fill with result)*	*(fill with result)*
Ridge	0 (fixed)	*(fill with result)*	*(fill with result)*	*(fill with result)*
Elastic Net	*(fill with best alpha)*	*(fill with corresponding lambda)*	*(fill with result)*	*(fill with result)*

(Note: This table should be populated with actual results from R analysis after performing cross-validation for each model type and, for Elastic Net, after the grid search for alpha.)

The construction of the predictor matrix x is a critical step. For time series forecasting, this matrix will typically include lagged values of the target variable (inflation) and current and/or lagged values of exogenous predictors. The number of lags to include for each variable is a modeling decision that can be guided by economic theory, domain knowledge, or preliminary analyses like cross-correlation functions. R functions such as dplyr::lag(), base R's embed(), or functions from specialized packages like tsDyn (e.g., lagmatrix) can be used to create these lagged predictor columns. The choice of the maximum lag length can significantly impact model complexity and performance, and may itself be a parameter to tune or determine based on information criteria in a preliminary step.

The glmnetUtils package offers a formula interface for glmnet and cv.glmnet (e.g., glmnetUtils::cv.glmnet(inflation ~ lag1_inflation + lag1_oil_price, data = my_ts_df, alpha = 1)), which can simplify the creation of the predictor matrix, especially if interaction terms or simple transformations are desired directly within the formula.[188] This can be a convenient alternative for users more familiar with R's standard formula syntax.

Section 14: Time Series Forecasting with Penalized Regressions

After fitting and tuning penalized regression models, they can be used to generate out-of-sample forecasts for inflation.

Forecasting Approaches

When generating a sequence of forecasts over a test period, two common schemes are used:

- **Recursive (Expanding Window) Forecasting:** At each step in the test period, the model is re-estimated using all available historical data up to that point. The training window expands as more data becomes available. This approach is suitable if the underlying data generating process is believed to be relatively stable or evolving slowly.
- **Rolling Window (Fixed Window) Forecasting:** At each step, the model is re-estimated using only the most recent W observations (a fixed-size window of data). This approach is more adaptive to potential structural changes or parameter instability in the data, as it gives less weight to distant past observations.

The choice between recursive and rolling schemes depends on assumptions about the stability of the relationships being modeled. For this academic project, a recursive scheme is often a good starting point unless there's strong evidence of significant structural breaks that would make older data less relevant.

Generating Out-of-Sample Forecasts

- **One-Step-Ahead Forecasts:** To forecast inflation for time T+1, the model is trained using data up to time T. The values of the predictors for time T+1 (which would be based on information available at time T, e.g., yT, xregT) are then fed into the fitted model to predict yT+1.
- **Multi-Step-Ahead Forecasts:** For forecasting h steps ahead (e.g., yT+h):
  - **Direct Forecasting:** This approach involves building a separate model for each forecast horizon h. For example, to predict yT+h, the response variable is yt+h and predictors are values known at time t. This is often preferred for penalized regression models as it avoids the issue of iterating on forecasted values.
  - **Iterated Forecasting:** A one-step-ahead model is used. To forecast yT+2, the forecast y^T+1 is used as an input if lagged values of y are predictors. This process is repeated for yT+3, and so on. While common for ARIMA models, iterated forecasting can accumulate errors, especially over longer horizons.

For penalized regression models in this project, the **direct forecasting** approach is generally more straightforward to implement. For each desired forecast horizon h (e.g., 1-month ahead, 3-months ahead, 6-months ahead, 12-months ahead), a separate model would be estimated where the target variable is inflationt+h and the predictors are lagged values of inflation and other exogenous variables known up to time t.

Updating Predictor Matrix for Future Predictions

When making genuine out-of-sample forecasts (i.e., predicting beyond the available data), the predictor matrix newx fed to the predict() function must contain the values of the predictors corresponding to the forecast period.

- If predictors include lags of inflation, these will be known up to the forecast origin. For multi-step iterated forecasts, forecasted inflation values would be used. For direct forecasts of yt+h using predictors from time t, this is straightforward.
- If predictors include exogenous variables (e.g., oil prices, M2), their future values for the forecast horizon must be available. If these are not known (which is typical), they must either be:
  - Forecasted using separate models (e.g., an ARIMA model for oil prices).
  - Based on assumed scenarios (e.g., oil price remains constant or follows a specific path).
  - Or, the penalized regression models must be restricted to using only lagged values of these exogenous variables such that their values are known at the

time of forecasting.

The direct forecasting approach is generally simpler for implementing multi-step forecasts with penalized regression, as it directly models the h-step ahead target. However, it does require fitting h separate models if forecasts for multiple horizons are needed from the same origin.

The computational cost of re-estimating glmnet models in a rolling or recursive forecasting scheme can be considerable, particularly if a two-dimensional cross-validation (for both α and λ) is performed at each forecasting step. For a long test set, this can be time-consuming. Strategies to manage this include updating the hyperparameters (α and optimal λ) less frequently (e.g., once a year in data time, rather than at every monthly step) or using efficient coding practices and parallelization if available. This computational trade-off should be acknowledged.

Part V: Comparative Evaluation and Conclusion

Section 15: Framework for Forecast Evaluation

A robust framework is essential for evaluating and comparing the forecasting performance of the different models developed. This involves splitting the data, defining error metrics, and using appropriate statistical tests.

Training and Testing (Hold-out) Samples

To obtain an unbiased assessment of forecast accuracy, the available data should be divided into two distinct sets [190]:

1. **Training Set:** This portion of the data is used to estimate model parameters and tune any hyperparameters (e.g., selecting λ and α for penalized regressions, or ARIMA orders if not using auto.arima exclusively).
2. **Test Set (Hold-out Set):** This portion of the data is kept separate and is used *only* for evaluating the out-of-sample forecast performance of the finalized models. The models are not exposed to this data during the estimation or tuning phases.

For time series data, it is crucial that the test set consists of observations that occur *after* the training set. A common split is to use approximately 70-80% of the data for training and the remaining 20-30% for testing.[190] The test set should ideally be at least as long as the maximum forecast horizon being considered.

In R, time series data can be split using the window() function for ts objects, or by simple indexing if using data frames with a date column.[190]

Code snippet

```
# Assuming 'full_inflation_ts' is the complete ts object
# Determine the split point
# For example, if data runs from Jan 2000 to Dec 2024, and we want 3 years for testing (36 months)
# train_end_year <- 2021
# train_end_month <- 12
# test_start_year <- 2022
# test_start_month <- 1

# training_ts <- window(full_inflation_ts, end = c(train_end_year, train_end_month))
# test_ts <- window(full_inflation_ts, start = c(test_start_year, test_start_month))
```

Forecast Errors

The forecast error (et) for a given period t is the difference between the actual observed value (yt) and the forecasted value (y^t):
$ e_t = y_t - \hat{y}_t $
190
These errors are calculated for each period in the test set.

Key Forecast Accuracy Metrics

Several metrics are commonly used to summarize forecast errors and evaluate accuracy [190]:

- **Scale-Dependent Errors:** These metrics are in the same units as the data.
  - **Mean Absolute Error (MAE):** Calculates the average of the absolute forecast errors. $ MAE = \text{mean}(|e_t|) $ MAE is easily interpretable as the average magnitude of the forecast error. Minimizing MAE leads to forecasts of the median.
  - **Root Mean Squared Error (RMSE):** Calculates the square root of the average of the squared forecast errors. $ RMSE = \sqrt{\text{mean}(e_t^2)} $ RMSE gives a higher weight to larger errors due to the squaring term. It is a very common metric, and minimizing RMSE leads to forecasts of the mean.
- **Percentage Errors:** These metrics are unit-free, allowing for comparisons across different time series or models.

- **Mean Absolute Percentage Error (MAPE):** Calculates the average of the absolute percentage errors. $ MAPE = \text{mean}\left(\left|\frac{100 \cdot e_t}{y_t}\right|\right) $ MAPE expresses the error as a percentage of the actual value. However, it has limitations: it can be infinite or undefined if yt=0, and it can produce extremely large values if yt is close to zero. It also asymmetrically penalizes positive and negative errors of the same magnitude.

- Scaled Errors (e.g., Mean Absolute Scaled Error - MASE):
  MASE scales the errors based on the in-sample, one-step-ahead MAE of a naïve (e.g., random walk) forecast method. An MASE < 1 suggests the forecast is better than the average one-step naïve forecast in the training set. While not explicitly requested by the user, MASE is a robust metric.

No single accuracy metric is universally superior. RMSE heavily penalizes large errors, MAE provides a more direct interpretation of average error magnitude, and MAPE is useful for relative comparisons but must be used cautiously if actual values can be zero or near-zero.[190] Reporting multiple metrics provides a more comprehensive assessment of forecast performance. For inflation forecasting, where the target variable (inflation rate) is typically non-zero and positive, MAPE can be informative, but RMSE and MAE are standard.

R Function for Accuracy Calculation

The forecast package provides the accuracy() function, which is very convenient for calculating these metrics:

Code snippet

```
# Assuming 'forecast_object' is an object returned by forecast()
# and 'actual_test_values' is a ts object or vector of the true values from the test set

# accuracy_metrics <- forecast::accuracy(forecast_object, actual_test_values)
# print(accuracy_metrics)
```

This function will return a table containing RMSE, MAE, MAPE, MASE (if in-sample data is available in forecast_object), and other metrics for both the training set (residuals) and the test set (forecast errors).[149] For model comparison, the metrics on the test set are of primary interest.

The choice of a single training/test split point can influence the evaluation results, as performance might vary depending on the specific period chosen for testing. For more robust evaluation, especially if the time series is long enough, techniques like time series cross-validation (e.g., using a rolling forecast origin, implemented in R via forecast::tsCV()) can provide multiple out-of-sample error estimates, leading to a more reliable assessment of generalization performance.[149] However, for this project, a single, well-chosen hold-out set is a standard and acceptable approach.

Section 16: Empirical Comparison of Model Performance

This section outlines how to compare the forecasting performance of the ARIMA-family models and the penalized regression models using the metrics and tests discussed previously.

Presenting Accuracy Metrics

A clear and concise table summarizing the key accuracy metrics (MAE, RMSE, MAPE) for all developed models on the common test set is essential. This allows for a direct comparison of their out-of-sample predictive capabilities.

Table: Comparative Forecast Accuracy Metrics

Model	MAE (Test Set)	RMSE (Test Set)	MAPE (Test Set) (%)
ARIMA/SARIMA Models			
Baseline ARIMA (e.g., from auto.arima)	*(fill)*	*(fill)*	*(fill)*
Selected SARIMA (if seasonality present)	*(fill)*	*(fill)*	*(fill)*
Selected ARIMAX/SARIMAX (with chosen xreg)	*(fill)*	*(fill)*	*(fill)*
Penalized Regression Models			
Lasso Regression (lambda.1se)	*(fill)*	*(fill)*	*(fill)*

Ridge Regression (lambda.1se)	*(fill)*	*(fill)*	*(fill)*
Elastic Net Regression (alpha & lambda.1se)	*(fill)*	*(fill)*	*(fill)*
(Optional: Lasso/Elastic Net lambda.min)	*(fill)*	*(fill)*	*(fill)*

(Note: This table should be populated with actual results from applying forecast::accuracy() to the forecasts generated by each model on the test set.)

Statistical Comparison of Forecast Accuracy: Diebold-Mariano Test

While the table above shows which model has lower error metrics, it does not indicate whether these differences are statistically significant. The Diebold-Mariano (DM) test is used to formally compare the predictive accuracy of two competing forecast models.[149]

- **Null Hypothesis (H0):** The two forecast methods have the same forecast accuracy.
- **Alternative Hypothesis (H1):**
  - "two.sided": The two methods have different levels of accuracy (default).
  - "less": Method 2 is less accurate than method 1.
  - "greater": Method 2 is more accurate than method 1.
- **Implementation in R:** forecast::dm.test(e1, e2, alternative = "two.sided", h = 1, power = 2)
  - e1, e2: Numeric vectors of forecast errors from model 1 and model 2, respectively, for the same test set periods.
  - h: The forecast horizon. For one-step-ahead forecasts, h=1.
  - power: The power used in the loss function. power=1 corresponds to an MAE-based comparison, while power=2 (default) corresponds to an MSE-based comparison. The dm.test function in the forecast package implements a modified version of the test proposed by Harvey, Leybourne, and Newbold (1997), which is often preferred for better small-sample properties.[193]

Table: Diebold-Mariano Test Results (Example Pairings)

Model 1	Model 2	Forecast Horizon (h)	Loss (power)	DM Statistic	p-value	Conclusion (at α=0.05)
Best ARIMA/SARIMAX	Best Penalized Reg. (e.g., Elastic Net)	1	2 (MSE)	*(fill)*	*(fill)*	*(e.g., Model 2 significantly better / No significant diff.)*
Best ARIMA/SARIMAX	Best Penalized Reg. (e.g., Elastic Net)	1	1 (MAE)	*(fill)*	*(fill)*	*(fill)*
Lasso	Elastic Net	1	2 (MSE)	*(fill)*	*(fill)*	*(fill)*
(Other relevant pairwise comparisons)						

(Note: This table should be populated with actual results from applying forecast::dm.test() to the forecast errors of the selected models.)

Discussion of Results

The discussion should address:

- Which model(s) performed best overall based on the chosen accuracy metrics (MAE, RMSE, MAPE)?
- Were the observed differences in forecast accuracy statistically significant according to the Diebold-Mariano tests?
- How do these findings relate to the characteristics of the models? For example:
    - If penalized regressions outperformed ARIMA-type models, was it due to their ability to incorporate a larger set of predictors effectively?
    - If SARIMA improved over ARIMA, it indicates the presence and importance of seasonality in Pakistan's inflation.
    - If ARIMAX/SARIMAX models were superior to their univariate counterparts,

which exogenous variables proved to be important (based on coefficient significance in ARIMA or selection in penalized models)?

- Comparison with existing literature on inflation forecasting in Pakistan or similar emerging economies.[2] Some studies suggest machine learning methods like Lasso, Ridge, or Elastic Net can outperform traditional econometric models like ARIMA for inflation forecasting, especially when dealing with large datasets or forecasting over longer horizons.[2] Conversely, other studies find ARIMA models to be highly competitive or even superior in specific contexts, particularly for short-term forecasting or when data is limited.[168]

The relative performance of ARIMA versus penalized regression models is often contingent on several factors: the number and quality of available predictor variables, the length and characteristics of the time series (e.g., degree of non-linearity, presence of structural breaks), the stability of underlying economic relationships, and the specific forecast horizon. Penalized regressions are designed to leverage information from a large set of predictors [2], whereas ARIMA models are more parsimonious, primarily relying on the past values of the series itself and past errors.[17] If strong, stable linear or seasonal patterns dominate the inflation series and good exogenous predictors are few or noisy, ARIMA models might perform very well. However, if numerous predictors collectively contain useful information, or if non-linear relationships are important (which can be approximated to some extent by including many transformed predictors or interaction terms in a penalized regression framework), then penalized methods might offer superior performance. The discussion should attempt to analyze *why*

Works cited

1. Forecasting Inflation Rate Using the ARIMA Model: Zambia's Perspective from 2023 to 2043 - International Journal of Research and Scientific Innovation (IJRSI), accessed May 6, 2025, https://rsisinternational.org/journals/ijrsi/articles/forecasting-inflation-rate-using-the-arima-model-zambias-perspective-from-2023-to-2043/
2. A Machine Learning Guide for Econometricians – A Hands-on Approach - IBA Karachi, accessed May 6, 2025, https://www.iba.edu.pk/sess/labs/egf/a-machine-learning-guide-econometricians-hands-on-approach.php
3. Three Attempts at Inflation Forecasting in Pakistan -- Madhavi Bokil and Axel Schimmelpfennig -- IMF Working Paper 05/105, May 1, accessed May 6, 2025, https://www.imf.org/external/pubs/ft/wp/2005/wp05105.pdf
4. Forecasting Inflation in Kenya Using ARIMA Model - African Journals Online (AJOL), accessed May 6, 2025, https://www.ajol.info/index.php/asarev/article/download/273155/257919/642872

5. State Bank of Pakistan lowers policy rate to 12 per cent with 100bps reduction, accessed May 6, 2025, https://tribune.com.pk/story/2524829/sbp-slashes-interest-rate-by-100bps-sets-policy-rate-at-12-precent
6. PAKISTAN - Public Documents | The World Bank, accessed May 6, 2025, https://thedocs.worldbank.org/en/doc/e414b36ae736660edf8f0f3cb597b1e9-0310012025/original/Pakistan-Development-Update-Report-April-2025-FINAL.pdf
7. FORECASTING INFLATION IN A DATA-RICH ENVIRONMENT: THE BENEFITS OF MACHINE LEARNING METHODS Marcelo C. Medeiros, accessed May 6, 2025, https://apps.unive.it/server/eventi/17435/Medeiros_paper.pdf
8. Nuts and Bolts of a Forecasting and Policy Analysis System in - IMF eLibrary, accessed May 6, 2025, https://www.elibrary.imf.org/display/book/9781484325940/ch004.xml
9. (PDF) Forecasting the Inflation for Budget Forecasters: An Analysis of ANN Model Performance in Türkiye - ResearchGate, accessed May 6, 2025, https://www.researchgate.net/publication/390271229_Forecasting_the_Inflation_for_Budget_Forecasters_An_Analysis_of_ANN_Model_Performance_in_Turkiye/download
10. SBP EasyData - Home, accessed May 6, 2025, https://easydata.sbp.org.pk/apex/f?p=10:1:0:
11. State Bank of Pakistan, accessed May 6, 2025, https://www.sbp.org.pk/
12. Pakistan Bureau of Statistics: Home, accessed May 6, 2025, https://www.pbs.gov.pk/
13. The Ultimate ARIMA Model Guide in Economic Theory - Number Analytics, accessed May 6, 2025, https://www.numberanalytics.com/blog/ultimate-arima-model-guide-economic-theory
14. Challenges with ARIMA (AutoRegressive Integrated Moving Average) - BytePlus, accessed May 6, 2025, https://www.byteplus.com/en/topic/402185
15. Forecasting Inflation in Mongolia Using Machine Learning - Taylor & Francis Online, accessed May 6, 2025, https://www.tandfonline.com/doi/full/10.1080/10168737.2025.2491386?src=
16. Understanding Inflation Dynamics: The Role of Global Shocks in CEMAC, WP/24/55, March 2024, accessed May 6, 2025, https://www.imf.org/-/media/Files/Publications/WP/2024/English/wpiea2024055-print-pdf.ashx
17. store.ectap.ro, accessed May 6, 2025, https://store.ectap.ro/articole/1641.pdf
18. Phillips Curve Inflation Forecasts - National Bureau of Economic Research, accessed May 6, 2025, https://www.nber.org/system/files/working_papers/w14322/w14322.pdf
19. Monthly Price Indices (Base Year 2015-16) | Pakistan Bureau of Statistics, accessed May 6, 2025, https://www.pbs.gov.pk/cpi
20. March 2025 - Monthly Review on Price Indices, accessed May 6, 2025, https://www.pbs.gov.pk/sites/default/files/price_statistics/cpi/Monthly_Review_March_2025.pdf

21. SBP EasyData - Dataset Series Chart - State Bank of Pakistan, accessed May 6, 2025, https://easydata.sbp.org.pk/apex/f?p=10:120:4512460874690::NO:RP:P120_SERIES_KEY,P120_ACTION,P120_PAGE_ID:TS_GP_PT_CPI_M.P00011516,New,100&cs=18EC341732156D8DE4D3B806D027D8D32
22. Dataset Series Chart - SBP EasyData, accessed May 6, 2025, https://easydata.sbp.org.pk/apex/f?p=10:220:2887724916300::NO:RP:P220_SERIES_KEY,P220_PAGE_ID:TS_GP_PT_CPI_M.P00011516,1&cs=1B7AE140DE77E33D3A36D8D63D0C32CE7
23. SBP EasyData - Home, accessed May 6, 2025, https://easydata.sbp.org.pk/
24. General CPI, an Inflation Measure (Month-on-Month basis) - SBP EasyData - Dataset Series Chart, accessed May 6, 2025, https://easydata.sbp.org.pk/apex/f?p=10:220:16930734117915::NO::P220_SERIES_KEY,P220_ACTION,P220_PAGE_ID:TS_GP_RLS_CPI5960_M.P00025960,New,211&cs=1965CD1E7B0E4FE07CB42C8DAE16F8325
25. SBP EasyData - Dataset Series Chart, accessed May 6, 2025, https://easydata.sbp.org.pk/apex/f?p=10:220:13861811600645::NO::P220_SERIES_KEY,P220_ACTION,P220_PAGE_ID:TS_GP_RLS_CPI8081_M.P00018081,New,211&cs=1785579E694146083F063A310BE7CEEFF
26. Urban CPI, an Inflation Measure (Year-on-Year basis) - SBP EasyData - Dataset Series Chart, accessed May 6, 2025, https://easydata.sbp.org.pk/apex/f?p=10:220:::NO::P220_SERIES_KEY,P220_ACTION,P220_PAGE_ID:TS_GP_PT_CPI_M.P00021516,New,211&cs=1810A431F08C49135959339E4D33758E1
27. Accessing International Financial Statistics (IFS) - IMF Data, accessed May 6, 2025, https://data.imf.org/ifs
28. IMF Data, accessed May 6, 2025, https://www.imf.org/en/Data
29. Data Home - IMF Data - International Monetary Fund (IMF), accessed May 6, 2025, https://data.imf.org/?sk=4BE0C9CB-272A-4667-8892-34B582B21BA6
30. Consumer Price Index - IMF Data, accessed May 6, 2025, https://data.imf.org/?sk=4ffb52b2-3653-409a-b471-d47b46d904b5
31. CPI - IMF Data, accessed May 6, 2025, https://data.imf.org/CPI
32. Consumer Price Index (CPI) - IMF Data, accessed May 6, 2025, https://data.imf.org/en/datasets/IMF.STA:CPI
33. Data Home, accessed May 6, 2025, https://data.imf.org/
34. Pakistan - IMF DataMapper, accessed May 6, 2025, https://www.imf.org/external/datamapper/profile/PAK
35. A Global Database of Inflation - World Bank, accessed May 6, 2025, https://www.worldbank.org/en/research/brief/inflation-database
36. World Development Indicators | DataBank, accessed May 6, 2025, https://databank.worldbank.org/reports.aspx?source=2&series=FP.CPI.TOTL.ZG&country=PAK
37. Inflation, consumer prices (annual %) - Pakistan - World Bank Data, accessed May 6, 2025, https://data.worldbank.org/indicator/FP.CPI.TOTL.ZG?locations=PK
38. DataBank | The World Bank, accessed May 6, 2025,

https://databank.worldbank.org/
39. Consumer Price Indices data for Pakistan | Pakistan - Food Prices ..., accessed May 6, 2025, https://data.humdata.org/dataset/faostat-food-prices-for-pakistan/resource/5811d7c8-d6b3-4abb-9fbf-3f393bb7e832
40. Pakistan - Food Prices | Humanitarian Dataset | HDX, accessed May 6, 2025, https://data.humdata.org/dataset/faostat-food-prices-for-pakistan
41. General and food consumer price indices inflation rates - FAO Open Knowledge, accessed May 6, 2025, https://openknowledge.fao.org/bitstreams/3db28299-8bd6-491a-98f1-add50241a42a/download
42. Consumer Price Indices | World Bank Data360, accessed May 6, 2025, https://data360.worldbank.org/en/dataset/FAO_CP
43. FAOSTAT database - Food and Agriculture Organization of the ..., accessed May 6, 2025, https://www.fao.org/faostat/en/#data/CP
44. January 2025 - Monthly Review on Price Indices, accessed May 6, 2025, https://www.pbs.gov.pk/sites/default/files/price_statistics/cpi/Monthly_Review_January_2025.pdf
45. Weekly Broad Money M2 - SBP EasyData - Dataset List, accessed May 6, 2025, https://easydata.sbp.org.pk/apex/f?p=10:211:7859714457485::NO:RP:P211_DATASET_TYPE_CODE,P211_PAGE_ID:TS_GP_BAM_M2_W,1&cs=1CBAECABBD50D9A7D395C32000B2748C3
46. Pakistan Money Supply M1, 1988 – 2025 | CEIC Data, accessed May 6, 2025, https://www.ceicdata.com/en/indicator/pakistan/money-supply-m1
47. Pakistan Money Supply M2 - Trading Economics, accessed May 6, 2025, https://tradingeconomics.com/pakistan/money-supply-m2
48. Pakistan Money Supply M2, 1988 – 2025 | CEIC Data, accessed May 6, 2025, https://www.ceicdata.com/en/indicator/pakistan/money-supply-m2
49. Pakistan Money Supply M1 - Trading Economics, accessed May 6, 2025, https://tradingeconomics.com/pakistan/money-supply-m1
50. Pakistan PK: Money Supply: M2 | Economic Indicators - CEIC, accessed May 6, 2025, https://www.ceicdata.com/en/pakistan/money-supply-annual/pk-money-supply-m2
51. Monetary and Financial Statistics - IMF Data, accessed May 6, 2025, https://data.imf.org/?sk=B83F71E8-61E3-4CF1-8CF3-6D7FE04D0930&sId=1503344291552
52. Accessing International Financial Statistics (IFS) - IMF Data, accessed May 6, 2025, https://data.imf.org/en/news/accessing%20international%20financial%20statistics
53. Monetary and Financial Statistics (MFS) - IMF Data, accessed May 6, 2025, https://data.imf.org/?sk=B83F71E8-61E3-4CF1-8CF3-6D7FE04D0930&sId=1390030341854
54. Datasets - IMF Data, accessed May 6, 2025, https://data.imf.org/datasets
55. Dataset Portal - IMF DATA Help, accessed May 6, 2025, https://datahelp.imf.org/knowledgebase/articles/505304-dataset-portal

56. Monetary and Financial Statistics - IMF Data, accessed May 6, 2025, https://data.imf.org/MFS
57. International Monetary Fund data - European University Institute, accessed May 6, 2025, https://www.eui.eu/Research/Library/ResearchGuides/Economics/Statistics/DataPortal/IMFdata
58. IMF/MFS | DBnomics, accessed May 6, 2025, https://db.nomics.world/IMF/MFS
59. State Bank of Pakistan's Policy (Target) Rate - SBP EasyData - Dataset Series Chart, accessed May 6, 2025, https://easydata.sbp.org.pk/apex/f?p=10:220:2887724916300::NO:RP:P220_SERIES_KEY,P220_PAGE_ID:TS_GP_IR_SIRPR_AH.SBPOL0030,1&cs=1A3D7A14EA3341E09890956075FAF562E
60. Pakistan Policy Rate, 1992 – 2025 | CEIC Data, accessed May 6, 2025, https://www.ceicdata.com/en/indicator/pakistan/policy-rate
61. Pakistan Interest Rate - Trading Economics, accessed May 6, 2025, https://tradingeconomics.com/pakistan/interest-rate
62. Interest Rates 2011-2019 - Datasets - Open Data Pakistan, accessed May 6, 2025, https://opendata.com.pk/dataset/interest-rates
63. IMF Cross Country Macroeconomic Statistics - Nasdaq Data Link, accessed May 6, 2025, https://data.nasdaq.com/databases/ODA
64. IMF International Financial Statistics, 1920-2024 - CESSDA Data Catalogue, accessed May 6, 2025, https://datacatalogue.cessda.eu/detail?q=2cfe735ffe8985165c027e97462cebfe4dc60b98e271f4ff3ee917bc9f13f0f2&lang=en
65. One-Month Karachi Interbank Offer Rate - SBP EasyData - Dataset Series Chart, accessed May 6, 2025, https://easydata.sbp.org.pk/apex/f?p=10:220:12368599763918::NO::P220_SERIES_KEY,P220_ACTION,P220_PAGE_ID:TS_GP_BAM_SIRKIBOR_D.KIBOR0010,New,211&cs=1F061071725587DA810C3889C242626F6
66. Structure of Interest Rates: KIBORs and KIBIDs - SBP EasyData - Dataset List - State Bank of Pakistan, accessed May 6, 2025, https://easydata.sbp.org.pk/apex/f?p=10:211:8461527090896::NO:RP:P211_DATASET_TYPE_CODE,P211_PAGE_ID:TS_GP_BAM_SIRKIBOR_D,1&cs=1883CA5742C889BB27CD0C1C818F1AB8B
67. Kibor Rates - State Bank of Pakistan, accessed May 6, 2025, https://www.sbp.org.pk/ecodata/kibor_index.asp
68. Kibor Rates - 1 Month - Karandaaz Data Portal, accessed May 6, 2025, https://portal.karandaaz.com.pk/dataset/daily-kibor-rates-1-month/1193
69. Kibor Rates | Karachi Stocks, accessed May 6, 2025, https://www.khistocks.com/market-data/kibor-rates.html
70. Monthly Avg. Ex. Rates in PKR - State Bank of Pakistan, accessed May 6, 2025, https://www.sbp.org.pk/ecodata/IBF_Arch.xls
71. Bank Floating Month End Exchange Rates (PKR per National Currency) - SBP EasyData - Dataset List, accessed May 6, 2025, https://easydata.sbp.org.pk/apex/f?p=10:211:::NO:RP:P211_DATASET_TYPE_CODE,P

211_PAGE_ID:TS_GP_ER_FMEERPKR_M,210&cs=1BA317C13763D8D441DBCE0D2ED286E34
72. Month End Exchange rate of US Dollar per PKR - SBP EasyData - Dataset Series Chart - State Bank of Pakistan, accessed May 6, 2025, https://easydata.sbp.org.pk/apex/f?p=10:220:::NO::P220_SERIES_KEY,P220_ACTION,P220_PAGE_ID:TS_GP_ER_FMEERUSD_M.E00250,New,211&cs=19E3B6BA47F33A48BCD476E745B2BB338
73. Average Exchange rate of Pak Rupees per US Dollar - SBP EasyData - Dataset Series Chart, accessed May 6, 2025, https://easydata.sbp.org.pk/apex/f?p=10:120:::NO:RP:P120_SERIES_KEY,P120_ACTION,P120_PAGE_ID:TS_GP_ER_FAERPKR_M.E00220,New,100&cs=12ED11FDA940A8DEFB86DB69D06F2FEBB
74. ER - IMF Data, accessed May 6, 2025, https://data.imf.org/Datasets/ER
75. request for a stand-by arrangement—press release; staff report; staff statement; and statement by the executive director for pakistan - Finance Division, accessed May 6, 2025, https://www.finance.gov.pk/mefp/IMF_21072023.pdf
76. Exchange Rate Archives by Month - International Monetary Fund (IMF), accessed May 6, 2025, https://www.imf.org/external/np/fin/data/param_rms_mth.aspx
77. IMF International Financial Statistics - Wikipedia, accessed May 6, 2025, https://en.wikipedia.org/wiki/IMF_International_Financial_Statistics
78. Dataset | International Financial Statistics (IFS) | World Bank Data360, accessed May 6, 2025, https://data360.worldbank.org/en/dataset/IMF_IFS
79. International Financial Statistics - IMF Data, accessed May 6, 2025, https://data.imf.org/?sk=4c514d48-b6ba-49ed-8ab9-52b0c1a0179b&sid=1438865921483
80. International Financial Statistics (IFS) - IMF Data, accessed May 6, 2025, https://data.imf.org/?sk=4c514d48-b6ba-49ed-8ab9-52b0c1a0179b&sId=1390030341854
81. International Financial Statistics - IMF Data, accessed May 6, 2025, https://data.imf.org/?sk=4c514d48-b6ba-49ed-8ab9-52b0c1a0179b
82. Accessing data from International Financial Statistics - IMF data portal download tutorial, accessed May 6, 2025, https://www.youtube.com/watch?v=Ap_kDEQrgqI
83. Exchange rates and prices - World Development Indicators | The World Bank, accessed May 6, 2025, https://wdi.worldbank.org/table/4.16
84. Exchange Rates and Real Effective Exchange Rates - Pakistan - CEIC, accessed May 6, 2025, https://www.ceicdata.com/en/pakistan/exchange-rates-and-real-effective-exchange-rates
85. Export and Import of Goods and Services - SBP EasyData - Dataset List - State Bank of Pakistan, accessed May 6, 2025, https://easydata.sbp.org.pk/apex/f?p=10:211:4278280750635::NO:RP:P211_DATASET_TYPE_CODE,P211_PAGE_ID:TS_GP_BOP_XMGS_M,210&cs=1634998159D5D38CEBAF9A6F379FBA0CA
86. Export and Import of Goods and Services - SBP EasyData - Dataset List, accessed May 6, 2025,

https://easydata.sbp.org.pk/apex/f?p=10:211:8265319688353::NO:RP:P211_DATASET_TYPE_CODE,P211_PAGE_ID:TS_GP_BOP_XMGS_M,1&cs=147AD264D44AECBF09E7B5D24ED2B4637
87. SBP EasyData - Dataset Series Chart - State Bank of Pakistan, accessed May 6, 2025, https://easydata.sbp.org.pk/apex/f?p=10:220:::NO::P220_SERIES_KEY,P220_ACTION,P220_PAGE_ID:TS_GP_BOP_XMGS_M.P00130,New,211&cs=17794E975B1B913A9D080A406D23DE43B
88. International Trade in Goods (by partner country) (IMTS) - IMF Data, accessed May 6, 2025, https://data.imf.org/dot
89. Indicator | External Trade, Goods, Value | World Bank Data360, accessed May 6, 2025, https://data360.worldbank.org/en/indicator/IMF_IFS_TG
90. International Trade in Goods (by partner country) (IMTS) - IMF Data, accessed May 6, 2025, https://data.imf.org/en/datasets/IMF.STA:IMTS
91. Direction of Trade Statistics (DOTS) - IMF Data, accessed May 6, 2025, https://data.imf.org/?sk=9d6028d4f14a464ca2f259b2cd424b85&sid=1514498212031
92. Direction of Trade Statistics (DOTS) - IMF Data, accessed May 6, 2025, https://legacydata.imf.org/?sk=9D6028D4-F14A-464C-A2F2-59B2CD424B85
93. Direction of Trade Statistics (DOTS) | World Bank Data360, accessed May 6, 2025, https://data360.worldbank.org/en/dataset/IMF_DOT
94. Direction of Trade Statistics - IMF Data, accessed May 6, 2025, https://data.imf.org/?sk=9d6028d4f14a464ca2f259b2cd424b85&sid=1409151240976
95. IMF Direction of Trade Statistics, 1948-2021 - UK Data Service › Study, accessed May 6, 2025, https://beta.ukdataservice.ac.uk/datacatalogue/studies/study?id=4745
96. Can I access the IMF's Direction of Trade Statistics online? - NYU Libraries' FAQs, accessed May 6, 2025, https://library.answers.nyu.edu/faq/100508
97. World Integrated Trade Solution (WITS) | Data on Export, Import, Tariff, NTM - World Bank, accessed May 6, 2025, https://wits.worldbank.org/
98. Pakistan Trade Statistics | WITS, accessed May 6, 2025, https://wits.worldbank.org/CountryProfile/en/PAK
99. Pakistan Trade | WITS Data - World Integrated Trade Solution (WITS), accessed May 6, 2025, https://wits.worldbank.org/countrysnapshot/en/PAK
100. Pakistan Trade | WITS Data - World Integrated Trade Solution (WITS), accessed May 6, 2025, https://wits.worldbank.org/countrysnapshot/PAK
101. Pakistan Trade | WITS | Text, accessed May 6, 2025, https://wits.worldbank.org/countrysnapshot/PAK/textview
102. Data Download - World Integrated Trade Solution (WITS), accessed May 6, 2025, https://wits.worldbank.org/datadownload.aspx?lang=en
103. World Bank's World Integrated Trade Solution (WITS) - Tax Management India. Com, accessed May 6, 2025, https://www.taxmanagementindia.com/visitor/detail_article.asp?ArticleID=13690
104. Trade Statistics - Ministry of Commerce | Government of Pakistan, accessed

May 6, 2025, https://www.commerce.gov.pk/trade-statistics/
105. Pakistan Industrial Production Index Growth, 1998 – 2025 | CEIC Data, accessed May 6, 2025, https://www.ceicdata.com/en/indicator/pakistan/industrial-production-index-growth
106. Quantum Index Series of Selected Large-scale Manufacturing Items (base 2015-16) - SBP EasyData - Dataset List, accessed May 6, 2025, https://easydata.sbp.org.pk/apex/f?p=10:211:::NO:RP:P211_DATASET_TYPE_CODE,P211_PAGE_ID:TS_GP_RL_LSM1516_M,1&cs=1DA6E8D7541EAE101E5A836B95D9DF1F4
107. Quantum Index of Large Scale Manufacturing Industries (QIM ..., accessed May 6, 2025, https://www.pbs.gov.pk/content/qim
108. Quantum Index of Large Scale Manufacturing Industries (QIM) for February 2025 | Pakistan Bureau of Statistics, accessed May 6, 2025, https://www.pbs.gov.pk/node/1322
109. Quantum Index Series of Selected Large-scale Manufacturing Items (base 2005-06) - SBP EasyData - Dataset List, accessed May 6, 2025, https://easydata.sbp.org.pk/apex/f?p=10:211:6194229339942::NO::P211_DATASET_TYPE_CODE,P211_PAGE_ID:TS_GP_RL_LSM_M,210&cs=1FEAE0BF14F3140EAABE144C97358FD78
110. Industry - Pakistan Bureau of Statistics, accessed May 6, 2025, https://www.pbs.gov.pk/content/industry
111. Quantum Index of Large Scale Manufacturing Industries (QIM) - Pakistan Bureau of Statistics, accessed May 6, 2025, https://www.pbs.gov.pk/qim
112. Enhanced General Data Dissemination System - IMF -GDDS DQAF Base (DSBB), accessed May 6, 2025, https://dsbb.imf.org/e-gdds/dqaf-base/country/PAK/category/IND00
113. Pakistan Bureau of Statistics (PBS) - Organizations - Open Data Pakistan, accessed May 6, 2025, https://opendata.com.pk/organization/b30ca4b2-5602-4ed7-9451-ca60ec58c83a?category=Manufacturing
114. Datasets - Open Data Pakistan, accessed May 6, 2025, https://www.opendata.com.pk/dataset?tags=Censuses&organization_type=Federal+government&organization=pakistan-bureau-of-statistics-pbs&_vocab_DATA_LOCATIONS_limit=0
115. cerp-analytics/pbs2017: This repository contains digitized data from Pakistan Bureau of Statistics's 2017 Census results. We converted them to csv format to help analysts and researchers make use of the data in research projects. - GitHub, accessed May 6, 2025, https://github.com/cerp-analytics/pbs2017
116. International Financial Statistics (IFS) - IMF Data, accessed May 6, 2025, https://data.imf.org/?sk=0828146b-47d2-49cf-ba7a-aebed8b5434d&hide_uv=1
117. Manufacturing Quantum Index - Pakistan - CEIC, accessed May 6, 2025, https://www.ceicdata.com/en/pakistan/manufacturing-quantum-index
118. Pakistan Manufacturing Production - Trading Economics, accessed May 6, 2025, https://tradingeconomics.com/pakistan/manufacturing-production

119. IMF Primary Commodity Prices - International Monetary Fund (IMF), accessed May 6, 2025, https://www.imf.org/en/Research/commodity-prices
120. Commodity prices - DataHub, accessed May 6, 2025, https://datahub.io/core/commodity-prices
121. Primary Commodity Price System - IMF Data, accessed May 6, 2025, https://data.imf.org/?sk=471dddf8-d8a7-499a-81ba-5b332c01f8b9&sid=1390030341854
122. Primary Commodity Price System - IMF Data, accessed May 6, 2025, https://data.imf.org/?sk=471DDDF8-D8A7-499A-81BA-5B332C01F8B9
123. IMF Primary Commodity Prices: Frequently Asked Questions (FAQs), accessed May 6, 2025, https://www.imf.org/external/np/res/commod/
124. IMF Primary Commodity Prices 1980-2013 - CESSDA Data Catalogue, accessed May 6, 2025, https://datacatalogue.cessda.eu/detail?lang=en&q=0d110f1cc5443560be4c953bae966c6c095ec226489a9cf419713602b38be761
125. Brent Crude Oil Spot Price Monthly Insights: IMF Primary Commodity Prices | YCharts, accessed May 6, 2025, https://ycharts.com/indicators/brent_crude_oil_spot_price_imf
126. Crude Oil Prices: West Texas Intermediate (WTI) - Cushing, Oklahoma (DCOILWTICO) | FRED | St. Louis Fed, accessed May 6, 2025, https://fred.stlouisfed.org/series/DCOILWTICO
127. Time Series Analysis with Oil Price Data in Python - Deepnote, accessed May 6, 2025, https://deepnote.com/app/kokinoda/Time-Series-Analysis-with-Oil-Price-Data-in-Python-71c3bedd-3c97-446e-9a7f-867f6ec5c3e7
128. Crude Oil Prices: Brent - Europe (DCOILBRENTEU) | FRED | St. Louis Fed, accessed May 6, 2025, https://fred.stlouisfed.org/series/DCOILBRENTEU/downlo
129. Crude Oil Prices: Brent - Europe (DCOILBRENTEU) | FRED | St ..., accessed May 6, 2025, https://fred.stlouisfed.org/series/DCOILBRENTEU
130. Brent and WTI Spot Prices - DataHub, accessed May 6, 2025, https://datahub.io/core/oil-prices
131. World Bank Commodities Price Data (The Pink Sheet), accessed May 6, 2025, https://thedocs.worldbank.org/en/doc/18675f1d1639c7a34d463f59263ba0a2-0050012025/related/CMO-Pink-Sheet-May-2025.pdf
132. World Food Price Index - Trading Economics, accessed May 6, 2025, https://tradingeconomics.com/world/food-price-index
133. FAO Food Price Index increases in April, accessed May 6, 2025, https://www.fao.org/newsroom/detail/fao-food-price-index-increases-in-april/en
134. FAO Food Price Index | Corn | Collection - MacroMicro, accessed May 6, 2025, https://en.macromicro.me/collections/53/agri-corn/48065/fao-food-price-index
135. FAO Food Price Index | Food and Agriculture Organization of the ..., accessed May 6, 2025, https://www.fao.org/worldfoodsituation/foodpricesindex/en/
136. Food and Beverage Price Index Monthly Trends: IMF Primary Commodity Prices | YCharts, accessed May 6, 2025, https://ycharts.com/indicators/food_and_beverage_price_index

137. Read Rectangular Text Data • readr, accessed May 6, 2025, https://readr.tidyverse.org/
138. Data Science for Practicing Clinicians - The Carpentries Incubator, accessed May 6, 2025, https://carpentries-incubator.github.io/Data-Science-for-Docs/aio/index.html
139. Pakistan Federal Public Expenditure Review 2023 - Open Knowledge Repository, accessed May 6, 2025, https://openknowledge.worldbank.org/entities/publication/3041989a-32fd-4906-9479-b8b18cf4e6b9
140. Manipulating Data with dplyr - sparklyr, accessed May 6, 2025, https://spark.posit.co/dplyr/
141. Dates and times made easy with lubridate - Tidyverse, accessed May 6, 2025, https://lubridate.tidyverse.org/reference/lubridate-package.html
142. yearmon: An Index Class for Monthly Data - RDocumentation, accessed May 6, 2025, https://www.rdocumentation.org/packages/zoo/versions/1.8-12/topics/yearmon
143. na.locf function - Last Observation Carried Forward - RDocumentation, accessed May 6, 2025, https://www.rdocumentation.org/packages/zoo/versions/1.8-12/topics/na.locf
144. Overview of janitor functions - CRAN, accessed May 6, 2025, https://cran.r-project.org/web/packages/janitor/vignettes/janitor.html
145. R Join Multiple Data Frames - Spark By {Examples}, accessed May 6, 2025, https://sparkbyexamples.com/r-programming/r-join-multiple-data-frames/
146. SARIMA in R | GeeksforGeeks, accessed May 6, 2025, https://www.geeksforgeeks.org/sarima-in-r/
147. imputeTS: Time Series Missing Value Imputation in R - DigitalCommons@UNL, accessed May 6, 2025, https://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1585&context=r-journal
148. mice: Multivariate Imputation by Chained Equations - RDocumentation, accessed May 6, 2025, https://www.rdocumentation.org/packages/mice/versions/3.17.0/topics/mice
149. forecast package - RDocumentation, accessed May 6, 2025, https://www.rdocumentation.org/packages/forecast/versions/8.24.0
150. ARIMA & SARIMA: Real-World Time Series Forecasting - neptune.ai, accessed May 6, 2025, https://neptune.ai/blog/arima-sarima-real-world-time-series-forecasting-guide
151. ARIMA and Seasonal ARIMA Models | Saylor Academy, accessed May 6, 2025, https://learn.saylor.org/mod/book/tool/print/index.php?id=55330
152. A Hybrid of Box-Jenkins ARIMA Model and Neural Networks for ..., accessed May 6, 2025, https://www.mdpi.com/2227-7072/12/4/118
153. Time Series: Interpreting ACF and PACF - Kaggle, accessed May 6, 2025, https://www.kaggle.com/code/iamleonie/time-series-interpreting-acf-and-pacf
154. Guide to Seasonal ARIMA: Forecasting with SARIMA Models, accessed May 6, 2025,

https://www.numberanalytics.com/blog/guide-to-seasonal-arima-forecasting-with-sarima-models
155. Comparative Analysis of ARIMA, SARIMA and Prophet Model in Forecasting, accessed May 6, 2025, https://www.sciencepublishinggroup.com/article/10.11648/j.rd.20240504.13
156. Applied Practical Guide to Unit Root Testing Techniques - Number Analytics, accessed May 6, 2025, https://www.numberanalytics.com/blog/applied-unit-root-testing-guide
157. Time Series Analysis - RPubs, accessed May 6, 2025, https://rpubs.com/rabbiamin/1179973
158. kpss.test function - RDocumentation, accessed May 6, 2025, https://www.rdocumentation.org/packages/tseries/versions/0.10-58/topics/kpss.test
159. AutoARIMA — sktime documentation, accessed May 6, 2025, https://www.sktime.net/en/v0.19.2/api_reference/auto_generated/sktime.forecasting.arima.AutoARIMA.html
160. AutoARIMA — sktime documentation, accessed May 6, 2025, https://www.sktime.net/en/stable/api_reference/auto_generated/sktime.forecasting.arima.AutoARIMA.html
161. An Introduction to `glmnet` • glmnet, accessed May 6, 2025, https://glmnet.stanford.edu/articles/glmnet.html
162. Chapter 4 Lasso Regression and Elasticnet | Prediction and Feature ..., accessed May 6, 2025, https://bookdown.org/staedler_n/highdimstats/lasso-regression-and-elasticnet.html
163. BINF301 Machine Learning for Systems Biology - Ridge, lasso and ..., accessed May 6, 2025, https://lab.michoel.info/BINF301-ML-for-sysbio/regularized-regression/ridge-lasso-elnet.html
164. Scaling to Success: Implementing and Optimizing Penalized Models ..., accessed May 6, 2025, https://machinelearningmastery.com/scaling-to-success-implementing-and-optimizing-penalized-models/
165. Feature Engineering: Scaling, Normalization, and Standardization | GeeksforGeeks, accessed May 6, 2025, https://www.geeksforgeeks.org/ml-feature-scaling-part-2/
166. glmnet fit a GLM with lasso or elasticnet regularization - RDocumentation, accessed May 6, 2025, https://www.rdocumentation.org/packages/glmnet/versions/4.1-8/topics/glmnet
167. (PDF) Seasonal Time Series Analysis: Why SARIMA Outshines ARIMA - ResearchGate, accessed May 6, 2025, https://www.researchgate.net/publication/384196885_Seasonal_Time_Series_Analysis_Why_SARIMA_Outshines_ARIMA
168. Inflation forecasting in the Western Balkans and EU: A comparison of Holt-Winters, ARIMA and NNAR models - EconStor, accessed May 6, 2025,

https://www.econstor.eu/bitstream/10419/281585/1/1795437685.pdf
169. 3/RT/98 - Forecasting Irish Inflation Using ARIMA Models - Central Bank of Ireland, accessed May 6, 2025, https://www.centralbank.ie/docs/default-source/publications/research-technical-papers/3rt98---forecasting-irish-inflation-using-arima-models-(kenny-meyler-and-quinn).pdf
170. 8.7 ARIMA modelling in R | Forecasting: Principles and Practice (2nd ed) - OTexts, accessed May 6, 2025, https://otexts.com/fpp2/arima-r.html
171. (Stata13): ARIMA Models (Identification) #arima #arma #boxjenkins #financialeconometrics, accessed May 6, 2025, https://www.youtube.com/watch?v=Rd67Tin8igA&pp=0gcJCfcAhR29_xXO
172. 8.6 Estimation and order selection | Forecasting: Principles and ..., accessed May 6, 2025, https://otexts.com/fpp2/arima-estimation.html
173. Lesson 12: Estimation of the parameters of an ARMA model, accessed May 6, 2025, https://www.lem.sssup.it/phd/documents/Lesson12.pdf
174. ARIMA - Parameter Estimation, accessed May 6, 2025, https://docs.tibco.com/pub/stat/14.2.0/doc/html/UserGuide/user-guide/arima-parameter-estimation.htm
175. Master Ljung-Box Test: Key Steps for Time Series, accessed May 6, 2025, https://www.numberanalytics.com/blog/master-ljung-box-test-steps
176. 6.6.2.4. Model Validation - Information Technology Laboratory, accessed May 6, 2025, https://www.itl.nist.gov/div898/handbook/pmc/section6/pmc624.htm
177. Ljung-Box Statistics for ARIMA residuals in R: confusing test results - Cross Validated, accessed May 6, 2025, https://stats.stackexchange.com/questions/64711/ljung-box-statistics-for-arima-residuals-in-r-confusing-test-results
178. Chapter 19: Ridge Regression, Lasso, Elastic Net - World Scientific Publishing, accessed May 6, 2025, https://www.worldscientific.com/doi/10.1142/9789811216572_0019
179. Mitigating Multicollinearity in Data Analysis for Predictive Accuracy - Number Analytics, accessed May 6, 2025, https://www.numberanalytics.com/blog/mitigating-multicollinearity-predictive-accuracy
180. (PDF) Effect of Multicollinearity on Variable Selection in Multiple Regression - ResearchGate, accessed May 6, 2025, https://www.researchgate.net/publication/356977729_Effect_of_Multicollinearity_on_Variable_Selection_in_Multiple_Regression
181. Multicollinearity in Regression Analysis: Problems, Detection, and Solutions, accessed May 6, 2025, https://statisticsbyjim.com/regression/multicollinearity-in-regression-analysis/
182. Ridge and Lasso Regression in Python - Analytics Vidhya, accessed May 6, 2025, https://www.analyticsvidhya.com/blog/2016/01/ridge-lasso-regression-python-complete-tutorial/
183. Lasso vs Ridge vs Elastic Net | ML - GeeksforGeeks, accessed May 6, 2025,

https://www.geeksforgeeks.org/lasso-vs-ridge-vs-elastic-net-ml/
184. Lasso and Ridge Regression in Python & R Tutorial - Analytics Vidhya, accessed May 6, 2025, https://www.analyticsvidhya.com/blog/2017/06/a-comprehensive-guide-for-linear-ridge-and-lasso-regression/
185. The Elastic Net with the Simulator - CRAN, accessed May 6, 2025, https://cran.r-project.org/web/packages/simulator/vignettes/elastic-net.html
186. Elastic/ridge/lasso analysis, what then? - Cross Validated - Stats Stackexchange, accessed May 6, 2025, https://stats.stackexchange.com/questions/137798/elastic-ridge-lasso-analysis-what-then
187. What is the Glmnet package in R? - GeeksforGeeks, accessed May 6, 2025, https://www.geeksforgeeks.org/what-is-the-glmnet-package-in-r/
188. Formula interface for elastic net cross-validation with cv.glmnet - RDocumentation, accessed May 6, 2025, https://www.rdocumentation.org/packages/glmnetUtils/versions/1.1.9/topics/cv.glmnet
189. make predictions from a "cv.glmnet" object., accessed May 6, 2025, https://glmnet.stanford.edu/reference/predict.cv.glmnet.html
190. 3.4 Evaluating forecast accuracy | Forecasting: Principles and ..., accessed May 6, 2025, https://otexts.com/fpp2/accuracy.html
191. Evaluating Forecast Accuracy | GeeksforGeeks, accessed May 6, 2025, https://www.geeksforgeeks.org/evaluating-forecast-accuracy/
192. Top 10 Machine Learning Evaluation Metrics for Regression - Implemented in R - Appsilon, accessed May 6, 2025, https://www.appsilon.com/post/machine-learning-evaluation-metrics-regression
193. dm.test function - RDocumentation, accessed May 6, 2025, https://www.rdocumentation.org/packages/forecast/versions/8.23.0/topics/dm.test
194. forecast/R/DM2.R at master · robjhyndman/forecast - GitHub, accessed May 6, 2025, https://github.com/robjhyndman/forecast/blob/master/R/DM2.R
195. Diebold-Mariano Test | Real Statistics Using Excel, accessed May 6, 2025, https://real-statistics.com/time-series-analysis/forecasting-accuracy/diebold-mariano-test/
196. ceur-ws.org, accessed May 6, 2025, https://ceur-ws.org/Vol-3619/AISD_Paper_2.pdf
197. (PDF) Comparative Analysis of Stock Price Prediction Models: Generalized Linear Model (GLM), Ridge Regression, Lasso Regression, Elasticnet Regression, and Random Forest -A Case Study on Netflix - ResearchGate, accessed May 6, 2025, https://www.researchgate.net/publication/374950281_Comparative_Analysis_of_Stock_Price_Prediction_Models_Generalized_Linear_Model_GLM_Ridge_Regression_Lasso_Regression_Elasticnet_Regression_and_Random_Forest_-A_Case_Study_on_Netflix
198. The benefits of forecasting inflation with machine learning: New evidence -

Queen Mary University of London, accessed May 6, 2025, https://qmro.qmul.ac.uk/xmlui/bitstream/handle/123456789/94801/J%20of%20Applied%20Econometrics%20-%202024%20-%20Naghi%20-%20The%20benefits%20of%20forecasting%20inflation%20with%20machine%20learning%20%20New%20evidence.pdf?sequence=22
199. Inflation forecasting with machine learning methods: A case of Mongolia, accessed May 6, 2025, https://www.pp.u-tokyo.ac.jp/wp-content/uploads/2016/02/R3_Batbold-Narmandakh.pdf
200. forecasting inflation rate with algorithmic arima modeling - ARF India, accessed May 6, 2025, https://www.arfjournals.com/image/catalog/Journals%20Papers/IJEFI/2024/No%202%20(2024)/2_Rushad%20Faridi.pdf
201. forecasting inflation by using the sub-groups of both cpi and wpi: evidence from auto regression (ar) and arima models, accessed May 6, 2025, https://ipe.ro/rjef/rjef2_21/rjef2_2021p144-161.pdf
202. Inflation Forecasting Models and Forecasting Combination Analysis: The Case of Ethiopia, accessed May 6, 2025, https://www.researchgate.net/publication/373659339_Inflation_Forecasting_Models_and_Forecasting_Combination_Analysis_The_Case_of_Ethiopia
203. FORECASTING OF THE INFLATION RATES IN UGANDA: A COMPARISON OF ARIMA, SARIMA AND VECM MODELS - DiVA portal, accessed May 6, 2025, http://www.diva-portal.org/smash/get/diva2:912462/FULLTEXT02
204. Forecasting and Analyzing Predictors of Inflation Rate: Using Machine Learning Approach, accessed May 6, 2025, https://ideas.repec.org/a/spr/jqecon/v22y2024i2d10.1007_s40953-024-00384-z.html
205. Mending the Crystal Ball: Enhanced Inflation Forecasts with Machine Learning, accessed May 6, 2025, https://www.imf.org/en/Publications/WP/Issues/2024/09/27/Mending-the-Crystal-Ball-Enhanced-Inflation-Forecasts-with-Machine-Learning-552848
206. Mending the Crystal Ball: Enhanced Inflation Forecasts with Machine Learning - International Monetary Fund (IMF), accessed May 6, 2025, https://www.imf.org/-/media/Files/Publications/WP/2024/English/wpiea2024206-print-pdf.ashx
207. Machine Learning Methods for Inflation Forecasting in Brazil: new contenders versus classical models - Banco Central do Brasil, accessed May 6, 2025, https://cdn-www.bcb.gov.br/content/publicacoes/WorkingPaperSeries/wps561.pdf
208. nber working paper series - efficient inflation forecasts, accessed May 6, 2025, https://www.nber.org/system/files/working_papers/w1542/w1542.pdf
209. NBER WORKING PAPER SERIES FORECASTING INFLATION James H. Stock NATIONAL BUREAU OF ECONOMIC RESEARCH March 1999, accessed May 6, 2025, https://www.nber.org/system/files/working_papers/w7023/w7023.pdf
210. Real-Time Inflation Forecasting in a Changing World, accessed May 6, 2025,

https://www.newyorkfed.org/medialibrary/media/research/staff_reports/sr388.pdf