Code Written by Suchibrata Patra

```
rm(list=ls())

# Loading The necessary libraries
library(tidyverse)
library(lubridate)
library(ggplot2)
library(dplyr)
library(scales)
library(zoo)
library(car)

# ---------------------------- #
# Reading Data Sets
misc_data = read.csv("misc.csv")
clothing_data = read.csv("clothing.csv")
housing_data = read.csv("housing.csv")
food_and_beverages_data = read.csv("food_and_beverages.csv") %>% filter(State == "ALL
India")
fuel_and_light_data = read.csv("fuel_and_light.csv")
cereals_and_products_data = read.csv("cereals_and_products.csv")
tobacco_data = read.csv("tobacco.csv")
general_index_data = read.csv("general_index.csv")
inflation_data = read.csv("inflation.csv")

# List of all Data Sets
datasets = list(
  misc_data = misc_data,
  clothing_data = clothing_data,
  housing_data = housing_data,
  fuel_and_light_data = fuel_and_light_data,
  tobacco_data = tobacco_data,
  general_index_data = general_index_data,
  food_and_beverages_data = food_and_beverages_data,
  cereals_and_products_data = cereals_and_products_data,
  inflation_data = inflation_data
)

# Function to remove columns with all NA or NULL values
remove_NA_Columns = function(df) {
  # Replace blank values with NA
  df[df == ""] = NA
  # Remove columns with all NA's
  df[, colSums(is.na(df)) != nrow(df)]
}
```

```
# Apply NA removal
datasets = lapply(datasets, remove_NA_Columns)


# Ensure all datasets have a Date column before interpolation
add_date_column = function(df) {
  if ("Year" %in% colnames(df) & "Month" %in% colnames(df)) {
    df$Date = make_date(df$Year, match(df$Month, month.name), 1)
  }
  return(df)
}

# Apply date column creation to all datasets
datasets = lapply(datasets, add_date_column)

# Interpolate missing values in the Housing dataset AFTER Date creation
housing_data = datasets$housing_data %>%
  arrange(Date) %>%
  mutate(Combined = na.approx(Combined, na.rm = FALSE))

datasets$housing_data = housing_data

# Create a data frame with dataset names and their dimensions
datasets = lapply(datasets, remove_NA_Columns)
dimensions_table = data.frame(
  Dataset = names(datasets),
  Rows = sapply(datasets, nrow),
  Columns = sapply(datasets, ncol),
  row.names = NULL
)
print(dimensions_table)


# ------------------------------------------------ #
#  Plot : CPI Inflation Trends in India (2013-2025)
# ------------------------------------------------ #

combined_data = un(
  list(
    "Clothing" = datasets$clothing_data,
    "Food & Beverages" = datasets$food_and_beverages_data,
    "Housing" = datasets$housing_data,         # Include fixed Housing data
    "Fuel & Light" = datasets$fuel_and_light_data
  ), .id = "Dataset"
)

slected_colors = c(
```

```
  "Clothing" = "#55A868",
  "Housing" = "#117A65",
  "Food & Beverages" = "#C44E52",
  "Fuel & Light" = "#8172B2"
)

# CPI Inflation Trends in India (2013-2025)
ggplot(combined_data, aes(x = Date, y = Combined, color = Dataset)) +
  geom_line(size = 1, lineend = "round") +  # Smooth, thick lines without points
  labs(
    title = "CPI Inflation Trends in India (2013-2025)",
    subtitle = "Data Source: Time Series data from : https://www.cpi.mospi.in",
    x = "Year",
    y = "CPI Index (Base Year: 2012)",
    caption = "Data Source: Time Series data from : https://www.cpi.mospi.in"
  ) +
  scale_x_date(date_labels = "%Y", date_breaks = "1 year") +  # Yearly intervals
  scale_y_continuous(labels = comma) +  # Format Y-axis with commas
  scale_color_manual(values = slected_colors) +
  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(face = "bold", size = 22, hjust = 0.5, color = "#2C3E50"),
    plot.subtitle = element_text(size = 14, color = "gray40"),
    plot.caption = element_text(size = 10, color = "gray50"),
    axis.title = element_text(size = 14),
    axis.text = element_text(size = 12),
    legend.position = "top",
    legend.title = element_blank(),
    panel.grid.major.y = element_line(color = "gray85", linetype = "dashed"),
    panel.grid.major.x = element_line(color = "gray90", linetype = "dashed"),
    panel.grid.minor = element_blank()
  )




# ------------------------------------------------------------#
# How did food and beverage inflation change over the year?
# ------------------------------------------------------------ #
food_inflation = datasets$food_and_beverages_data %>%
  filter(Date >= as.Date("2024-01-01") & Date <= as.Date("2025-01-31")) %>%
  arrange(Date)

ggplot(food_inflation, aes(x = Date, y = Combined)) +
  geom_line(color = "#27AE60", size = 1.5) +  # Green line for food inflation
  geom_point(color = "#145A32", size = 2) +   # Highlight data points
  labs(
```

```
    title = "Food & Beverage Inflation Trend in India (2024)",
    subtitle = "Monthly changes in food and beverage inflation rates",
    x = "Month",
    y = "CPI Index (Base Year: 2012)",
    caption = "Source: Food & Beverages CPI Dataset"
  ) +
  scale_x_date(date_breaks = "1 month", date_labels = "%b %Y") +
  scale_y_continuous(labels = comma) +
  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(face = "bold", size = 20, hjust = 0.5),
    plot.subtitle = element_text(size = 14, color = "gray40"),
    plot.caption = element_text(size = 10, color = "gray60"),
    axis.text.x = element_text(angle = 45, hjust = 1),
    panel.grid.major.x = element_line(color = "gray85", linetype = "dashed")
  )


# ------------------------------------------------------------#
# Impact of Fuel Prices on Overall CPI Inflation in India
# ------------------------------------------------------------#
# Merge Fuel and CPI Inflation Data by Date
inflation_fuel_data = datasets$general_index_data %>%
  select(Date, CPI_Combined = Combined) %>%
  inner_join(datasets$fuel_and_light_data %>%
               select(Date, Fuel_Inflation = Combined), by = "Date") %>%
  arrange(Date)

# Compute Correlation
correlation_value = cor(inflation_fuel_data$Fuel_Inflation,
inflation_fuel_data$CPI_Combined, use = "complete.obs")
print(paste("Correlation between Fuel Inflation and CPI:", round(correlation_value, 3)))

# Visualizing CPI Inflation vs Fuel Inflation
ggplot(inflation_fuel_data, aes(x = Date)) +
  geom_line(aes(y = CPI_Combined, color = "CPI Inflation"), size = 1.2, linetype = "solid") +
  geom_line(aes(y = Fuel_Inflation, color = "Fuel Inflation"), size = 1.2, linetype = "solid") +
  labs(
    title = "Impact of Fuel Prices on Overall CPI Inflation",
    subtitle = "Comparing Fuel Inflation with CPI Inflation Over Time",
    x = "Year",
    y = "Index Value",
    caption = "Source: CPI & Fuel Inflation Dataset"
  ) +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  scale_y_continuous(labels = comma) +
  scale_color_manual(values = c("CPI Inflation" = "#1F618D", "Fuel Inflation" = "#E74C3C"))
+
```

```
  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(face = "bold", size = 20, hjust = 0.5),
    plot.subtitle = element_text(size = 14, color = "gray40"),
    plot.caption = element_text(size = 10, color = "gray60"),
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.title = element_blank(),
    legend.position = "top"
  )

# Regression Analysis: Impact of Fuel Inflation on CPI Inflation
regression_model = lm(CPI_Combined ~ Fuel_Inflation, data = inflation_fuel_data)
summary(regression_model)




# ------------------------------------------------------------------------ #
# Analyzing Inflation Disparity Between Rural and Urban Areas in India
# ------------------------------------------------------------------------ #
# Merge Rural and Urban Inflation Data by Date
inflation_rural_urban = datasets$general_index_data %>%
  select(Date, Rural_Inflation = Rural, Urban_Inflation = Urban) %>%
  arrange(Date)

# Calculate Inflation Disparity (Gap)
inflation_rural_urban = inflation_rural_urban %>%
  mutate(Disparity = Rural_Inflation - Urban_Inflation)

# Visualization: Rural vs Urban Inflation with Disparity Shading
ggplot(inflation_rural_urban, aes(x = Date)) +
  # Rural Inflation Line
  geom_line(aes(y = Rural_Inflation, color = "Rural Inflation"), size = 1) +
  # Urban Inflation Line
  geom_line(aes(y = Urban_Inflation, color = "Urban Inflation"), size = 1, linetype = "dashed")
+

  # Disparity Shading
  geom_ribbon(aes(ymin = pmin(Rural_Inflation, Urban_Inflation),
                  ymax = pmax(Rural_Inflation, Urban_Inflation)),
              fill = "#FAD7A0", alpha = 0.5) +  # Shaded area between the lines

  labs(
    title = "Inflation Disparity Between Rural and Urban Areas in India",
    subtitle = "Visualizing the gap between rural and urban CPI trends over time",
    x = "Year",
    y = "CPI Index (Base Year: 2012)",
```

```
    caption = "Source: Rural and Urban CPI Dataset"
  ) +

  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  scale_y_continuous(labels = comma) +
  scale_color_manual(values = c("Rural Inflation" = "#2E86C1", "Urban Inflation" =
"#E74C3C")) +

  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(face = "bold", size = 20, hjust = 0.5),
    plot.subtitle = element_text(size = 14, color = "gray40"),
    plot.caption = element_text(size = 10, color = "gray60"),
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.title = element_blank(),
    legend.position = "top"
  )

# ---------------------------- #
# Summary Statistics: Average Disparity
disparity_stats = inflation_rural_urban %>%
  summarize(
    Avg_Disparity = abs(mean(Disparity, na.rm = TRUE)),
    Max_Disparity = abs(max(Disparity, na.rm = TRUE)),
    Min_Disparity = abs(min(Disparity, na.rm = TRUE))
  )

print(disparity_stats)

# ---------------------------------------------------------------------- #
# Impact of Housing Costs on Overall CPI Inflation in India
# ---------------------------------------------------------------------- #
# Correlation Calculation
correlation_value = cor(inflation_housing_data$Housing_Inflation,
                        inflation_housing_data$CPI_Combined,
                        use = "complete.obs")
print(paste("Correlation between Housing Inflation and CPI:", round(correlation_value, 3)))

# Regression Analysis: Impact of Housing Inflation on CPI Inflation
regression_model = lm(CPI_Combined ~ Housing_Inflation, data = inflation_housing_data)
summary(regression_model)




# ---------------------------------------------------------------------- #
# How frequently did inflation breach the RBI’s target range?
# ---------------------------------------------------------------------- #
```

```
ggplot(cpi_data, aes(x = Date)) +

  # CPI Percentage Inflation Line
  geom_line(aes(y = CPI_Percentage), color = "#1F618D", size = 1.5) +

  # Shaded Area for Breaches
  geom_ribbon(aes(ymin = ifelse(Breach, CPI_Percentage, NA),
                  ymax = ifelse(Breach, 10, NA)),  # Highlight breach area
              fill = "#E74C3C", alpha = 0.3) +

  # RBI Target Range Lines
  geom_hline(yintercept = 2, linetype = "dashed", color = "#27AE60", size = 1) +
  geom_hline(yintercept = 6, linetype = "dashed", color = "#27AE60", size = 1) +

  # Annotation to show compliance rate
  annotate("text", x = min(cpi_data$Date) + months(6),
           y = max(cpi_data$CPI_Percentage) - 0.5,
           label = paste0("Within Target Range: 2.05% ",
                          "\nBreach Rate: 97.95% "),
           size = 5, color = "black", hjust = 0) +

  labs(
    title = "Frequency of Inflation Breaches Beyond RBI's Target Range",
    subtitle = "Target Range: 2% - 6% (Breaches highlighted in red)",
    x = "Year",
    y = "Inflation (%)",
    caption = "Source: RBI & CPI Inflation Dataset"
  ) +

  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  scale_y_continuous(labels = scales::comma) +

  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(face = "bold", size = 20, hjust = 0.5),
    plot.subtitle = element_text(size = 14, color = "gray40"),
    plot.caption = element_text(size = 10, color = "gray60"),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )



#-----------------------------------------#
# ARIMA Model to predict next Inflation Rates
#-----------------------------------------#
# Prepare the Time Series Data
cpi_ts = ts(cpi_data$CPI_Combined,
            start = c(year(min(cpi_data$Date)), month(min(cpi_data$Date))),
```

```
            frequency = 12)

# Fit ARIMA Model
auto_arima_model = auto.arima(cpi_ts)  # Automatically select best parameters
summary(auto_arima_model)

# Forecast for the Next 12 Months
forecast_next_year = forecast(auto_arima_model, h = 12)

# Format the Forecast Table with Year & Month Separately
forecast_table = data.frame(
  Date = seq.Date(from = max(cpi_data$Date) + months(1),
                  by = "month", length.out = 12),
  Forecast = round(forecast_next_year$mean, 2),
  Lower_80 = round(forecast_next_year$lower[, 1], 2),
  Upper_80 = round(forecast_next_year$upper[, 1], 2),
  Lower_95 = round(forecast_next_year$lower[, 2], 2),
  Upper_95 = round(forecast_next_year$upper[, 2], 2)
) %>%
  # Add separate Year and Month columns
  mutate(
    Year = year(Date),
    Month = month(Date, label = TRUE, abbr = TRUE)
  ) %>%
  select(Year, Month, Forecast, Lower_80, Upper_80, Lower_95, Upper_95)

# Display the formatted forecast table
print("CPI Inflation Forecast for the Next 12 Months (Year & Month Format):")
print(forecast_table)

# Step 5: Visualize the Forecast
autoplot(forecast_next_year) +
  labs(
    title = "CPI Inflation Forecast for 2025-2026",
    subtitle = "ARIMA model-based forecast with confidence intervals",
    x = "Year",
    y = "CPI Index (Base Year: 2012)",
    caption = "Source: CPI Inflation Dataset, Forecast by ARIMA Model"
  ) +
  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(face = "bold", size = 20, hjust = 0.5),
    plot.subtitle = element_text(size = 14, color = "gray40"),
    plot.caption = element_text(size = 10, color = "gray60"),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```

```
#-----------------------------------------#
# Analyzing Inflation Fluctuations During the COVID-19 Pandemic
#-----------------------------------------#

# Define COVID Periods
covid_periods = c("Pre-Pandemic", "Pandemic", "Post-Pandemic")

# Assign periods based on dates
cpi_data = cpi_data %>%
  mutate(
    Period = case_when(
      Date < as.Date("2020-03-01") ~ "Pre-Pandemic",
      Date >= as.Date("2020-03-01") & Date <= as.Date("2022-06-30") ~ "Pandemic",
      Date > as.Date("2022-06-30") ~ "Post-Pandemic"
    )
  )
# Summary Statistics by Period
inflation_by_period = cpi_data %>%
  group_by(Period) %>%
  summarise(
    Avg_Inflation = mean(CPI_Combined, na.rm = TRUE),
    Max_Inflation = max(CPI_Combined, na.rm = TRUE),
    Min_Inflation = min(CPI_Combined, na.rm = TRUE),
    SD_Inflation = sd(CPI_Combined, na.rm = TRUE),
    N = n()
  )

# Display the statistics
print("Inflation Statistics by COVID Period:")
print(inflation_by_period)

# Visualizing CPI Inflation Across Periods
ggplot(cpi_data, aes(x = Date, y = CPI_Combined, color = Period)) +
  geom_line(size = 1.2) +

  geom_vline(xintercept = as.numeric(as.Date("2020-03-01")), linetype = "dashed", color =
"red") +
  geom_vline(xintercept = as.numeric(as.Date("2022-06-30")), linetype = "dashed", color =
"green") +

  labs(
    title = "CPI Inflation Trends During the COVID-19 Pandemic",
    subtitle = "Comparison of Pre-Pandemic, Pandemic, and Post-Pandemic Periods",
    x = "Year",
    y = "CPI Index (Base Year: 2012)",
    caption = "Source: CPI Inflation Dataset"
  ) +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
```

```
  scale_y_continuous(labels = scales::comma) +
  scale_color_manual(values = c("Pre-Pandemic" = "#1F618D",
                                "Pandemic" = "#E74C3C",
                                "Post-Pandemic" = "#27AE60")) +

  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(face = "bold", size = 20, hjust = 0.5),
    plot.subtitle = element_text(size = 14, color = "gray40"),
    plot.caption = element_text(size = 10, color = "gray60"),
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "top"
  )



#  ------------------------------------------------------------ #
## Testing For Hypothesis for Checking Impact of COVID 19
#  ------------------------------------------------------------ #
# Check for Normality (Shapiro-Wilk Test)
shapiro_test = cpi_data %>%
  group_by(Period) %>%
  summarise(p_value = shapiro.test(CPI_Combined)$p.value)

print("Normality Test (Shapiro-Wilk):")
print(shapiro_test)

# Check for Homogeneity of Variance (Levene's Test)
levene_test = leveneTest(CPI_Combined ~ Period, data = cpi_data)
print("Levene's Test for Homogeneity of Variance:")
print(levene_test)

# ANOVA Test (if assumptions hold)
anova_model = aov(CPI_Combined ~ Period, data = cpi_data)
anova_summary = summary(anova_model)
print("ANOVA Results:")
print(anova_summary)

# Post-hoc Tukey HSD Test (if ANOVA is significant)
if (anova_summary[[1]]$`Pr(>F)`[1] < 0.05) {
  tukey_test = TukeyHSD(anova_model)
  print("Tukey's HSD Post-hoc Test:")
  print(tukey_test)
} else {
  print("ANOVA is not significant. No need for post-hoc test.")
}

# Kruskal-Wallis Test (if ANOVA assumptions fail)
kruskal_test = kruskal.test(CPI_Combined ~ Period, data = cpi_data)
```

```
print("Kruskal-Wallis Test:")
print(kruskal_test)

# Effect Size (Eta Squared)
eta_squared = sum(anova_model$residuals^2) / sum((cpi_data$CPI_Combined -
mean(cpi_data$CPI_Combined))^2)
print(paste("Effect Size (Eta Squared):", round(eta_squared, 4)))
```

```
#------------------------------------------------------------------------#
# Urban vs. Rural Cost-of-Living Gap Analysis
#------------------------------------------------------------------------#
urban_rural_gap_data = datasets$general_index_data %>%
  select(Date, CPI_Combined = Combined) %>%
  inner_join(datasets$clothing_data %>%
               select(Date, Rural_Inflation = Combined), by = "Date") %>%
  inner_join(datasets$misc_data %>%
               select(Date, Urban_Inflation = Combined), by = "Date") %>%
  arrange(Date)

# Calculate the Rural-Urban Gap
urban_rural_gap_data = urban_rural_gap_data %>%
  mutate(Gap = Urban_Inflation - Rural_Inflation)

# Summary Statistics
gap_stats = urban_rural_gap_data %>%
  summarize(
    Avg_Gap = mean(Gap, na.rm = TRUE),
    SD_Gap = sd(Gap, na.rm = TRUE),
    Max_Gap = max(Gap, na.rm = TRUE),
    Min_Gap = min(Gap, na.rm = TRUE),
    Stability = SD_Gap / Avg_Gap  # Measure of stability
  )
cat("\n Urban vs. Rural Inflation Gap Statistics:\n")
print(gap_stats)

# Visualize the Urban-Rural Inflation Gap Over Time
ggplot(urban_rural_gap_data, aes(x = Date)) +

    # Urban Inflation Line
  geom_line(aes(y = Urban_Inflation, color = "Urban Inflation"), size = 1.2) +
```

```
  # Rural Inflation Line
  geom_line(aes(y = Rural_Inflation, color = "Rural Inflation"), size = 1.2, linetype = "dashed")
+

  # Gap Band (Shaded area between rural and urban inflation)
  geom_ribbon(aes(ymin = pmin(Urban_Inflation, Rural_Inflation),
                  ymax = pmax(Urban_Inflation, Rural_Inflation)),
              fill = "lightgray", alpha = 0.4) +

  labs(
    title = "Urban vs. Rural Cost-of-Living Gap in India",
    subtitle = "Analyzing the inflation disparity over time",
    x = "Year",
    y = "CPI Index",
    caption = "Source: CPI Inflation Dataset"
  ) +

  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  scale_y_continuous(labels = comma) +

  scale_color_manual(values = c("Urban Inflation" = "#1F618D", "Rural Inflation" =
"#E74C3C")) +

  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(face = "bold", size = 20, hjust = 0.5),
    plot.subtitle = element_text(size = 14, color = "gray40"),
    plot.caption = element_text(size = 10, color = "gray60"),
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.title = element_blank(),
    legend.position = "top"
  )




#-----------------------------------------------------------------------#
# Rising Cost of Common Basket Over Time
#-----------------------------------------------------------------------#
# Filter data for the items in the basket
household_basket_data = inflation_data %>%
  filter(Description %in% unique(inflation_data$Description))  # Unique items for household
basket

# Aggregate the inflation data by Year and Month
basket_inflation_trend = household_basket_data %>%
  group_by(Year, Month) %>%
```

```
  summarise(Avg_Basket_Inflation = mean(`Combined.Inflation`, na.rm = TRUE)) %>%
  ungroup() %>%
  arrange(Year, Month) %>%
  mutate(Date = as.Date(paste(Year, Month, "01", sep = "-"), format = "%Y-%B-%d"))

# Plot the inflation trend for the basket
ggplot(basket_inflation_trend, aes(x = Date, y = Avg_Basket_Inflation)) +

  # Main Inflation Trend Line
  geom_line(color = "#1F77B4", size = 1) +  # Thicker line for prominence

  # Smooth Trend Line (Loess) with dashed style
  geom_smooth(method = "loess", color = "#E74C3C", linetype = "dashed", size = 1) +  #
Red smooth line

  labs(
    title = "Rising Cost of Common Household Basket Over Time",
    subtitle = "Analysis of Inflation Trends Across Years",
    x = "Year-Month",
    y = "Average Inflation (%)"
  ) +

  theme_minimal(base_size = 16) +

  theme(
    plot.title = element_text(hjust = 0.5, size = 20, face = "bold", color = "#333333"),
    plot.subtitle = element_text(hjust = 0.5, size = 16, color = "#666666"),
    axis.title.x = element_text(size = 14, face = "bold", color = "#333333"),
    axis.title.y = element_text(size = 14, face = "bold", color = "#333333"),
    axis.text = element_text(size = 12, color = "#333333"),
    axis.text.x = element_text(angle = 45, hjust = 1),
    panel.grid.major = element_line(color = "#DDDDDD", size = 0.8),
    panel.grid.minor = element_blank(),
    plot.background = element_rect(fill = "white"),
    plot.margin = margin(10, 20, 10, 20)
  )


# ANOVA Test (Check if inflation is significantly different over years)
anova_model = aov(Avg_Basket_Inflation ~ factor(Year), data = basket_inflation_trend)
anova_summary = summary(anova_model)

# Print ANOVA results
print("ANOVA Results:")
print(anova_summary)
```