Week 3 Assignment Data 608

Mubashira Qari

Quarto Document

Load Libraries

```
#install.packages("tinytex")
#tinytex::install_tinytex()  # Install TinyTeX (if not installed)
#tinytex::tlmgr_install("koma-script")  # Install the missing KOMA-Script package
```

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
library(readxl)
library(ggplot2)
library(dplyr)
library(stringr)
library(tools)
library(stringdist)
```

```
Attaching package: 'stringdist'

The following object is masked from 'package:tidyr':

    extract
```

```r
library(broom)
library(gridExtra)
```

```
Attaching package: 'gridExtra'

The following object is masked from 'package:dplyr':

    combine
```

```r
library(gclus)
```

```
Loading required package: cluster
```

```r
library(car)
```

```
Loading required package: carData

Attaching package: 'car'

The following object is masked from 'package:dplyr':

    recode

The following object is masked from 'package:purrr':

    some
```

```r
library(VGAM)
```

```
Loading required package: stats4
Loading required package: splines
```

```
Attaching package: 'VGAM'

The following object is masked from 'package:car':

    logit
```

```r
library(MASS)
```

```
Attaching package: 'MASS'

The following object is masked from 'package:dplyr':

    select
```

```r
library(rpart.plot)
```

```
Loading required package: rpart
```

```r
library(ggfortify)
library(gridExtra)
library(forecast)
```

```
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
Registered S3 methods overwritten by 'forecast':
  method                 from
  autoplot.Arima         ggfortify
  autoplot.acf           ggfortify
  autoplot.ar            ggfortify
  autoplot.bats          ggfortify
  autoplot.decomposed.ts ggfortify
  autoplot.ets           ggfortify
  autoplot.forecast      ggfortify
  autoplot.stl           ggfortify
  autoplot.ts            ggfortify
  fitted.ar              ggfortify
  fortify.ts             ggfortify
  residuals.ar           ggfortify
```

```r
library(fpp2)
```

```
-- Attaching packages ---------------------------------------------- fpp2 2.5 --
v fma       2.5     v expsmooth 2.3
-- Conflicts ------------------------------------------------- fpp2_conflicts --
x car::some() masks purrr::some()
```

```r
library(fma)
library(kableExtra)
```

```

Attaching package: 'kableExtra'

The following object is masked from 'package:dplyr':

    group_rows
```

```r
library(e1071)
library(mlbench)
library(ggcorrplot)
library(DataExplorer)
library(timeDate)
```

```

Attaching package: 'timeDate'

The following objects are masked from 'package:e1071':

    kurtosis, skewness
```

```r
library(caret)
```

```
Loading required package: lattice

Attaching package: 'caret'

The following object is masked from 'package:VGAM':

    predictors
```

```
The following object is masked from 'package:purrr':

    lift
```

```r
library(GGally)
```

```
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2

Attaching package: 'GGally'

The following object is masked from 'package:fma':

    pigs
```

```r
library(corrplot)
```

```
corrplot 0.92 loaded
```

```r
library(RColorBrewer)
library(tibble)
library(tidyr)
library(reshape2)
```

```
Attaching package: 'reshape2'

The following object is masked from 'package:tidyr':

    smiths
```

```r
library(mixtools)
```

```
mixtools package, version 2.0.0, Released 2022-12-04
This package is based upon work supported by the National Science Foundation under Grant No.
```

```
Attaching package: 'mixtools'

The following object is masked from 'package:car':

    ellipse
```

```
library(skimr)
```

Loading Datasets

```
unemployment_data <- read.csv("https://raw.githubusercontent.com/uzmabb182/Data_608/refs/head

fed_data <- read.csv("https://raw.githubusercontent.com/uzmabb182/Data_608/refs/heads/main/We

cpi_data <- read.csv("https://raw.githubusercontent.com/uzmabb182/Data_608/refs/heads/main/We

#print(unemployment_data)
#print(fed_data)
#print(cpi_data)
```

Remove the HALF1 and HALF2 columns

```
cpi_df <- cpi_data[, !(names(cpi_data) %in% c("HALF1", "HALF2"))]
```

Convert from wide format to long format using `reshape()`

```
cpi_long <- reshape(cpi_df,
                    varying = list(2:ncol(cpi_df)),  # All columns except "Year"
                    v.names = "CPI",  # New column name for CPI values
                    timevar = "Month",  # New column for Month names
                    times = names(cpi_df)[2:ncol(cpi_df)],  # Month names from column names
                    idvar = "Year",  # Keep Year column as identifier
                    direction = "long")
#cpi_long
```

Saving as CSV

```r
# Define the file path with filename and extension
#file_path <- "C:/Users/Uzma/Downloads/new_df.csv"
file_path <- "C:/Users/Uzma/CUNY-SPS-Assignments/Data_608/output_csv/processed_interest_rates
# Write dataframe to CSV
write.csv(cpi_long, file = file_path, row.names = FALSE)

# Confirm that the file was saved
print("File saved successfully!")
```

```
[1] "File saved successfully!"
```

```r
#cpi_long
```

Calculate percentage change and round to 2 decimal places

```r
cpi_df <- cpi_long %>%
  group_by(Month) %>%  # Group by month to ensure comparisons are within the same month
  mutate(`inflation_rate` = round(((CPI - lag(CPI)) / lag(CPI)) * 100, 2)) %>%
  ungroup()

#cpi_df
```

Create a new column 'Status' based on 'inflation_rate'

```r
cpi_df <- cpi_df %>%
  mutate(inflation_criteria = ifelse(inflation_rate > 2, "Not Achieved", "Achieved"))

#cpi_df
```

Remove rows where Year is 1999

```
cpi_df <- cpi_df %>%
  filter(Year != 1999)  # Keep only rows where Year is NOT 1999

#cpi_df
```

Saving as CSV

```
# Define the file path with filename and extension
#file_path <- "C:/Users/Uzma/Downloads/new_df.csv"
file_path <- "C:/Users/Uzma/CUNY-SPS-Assignments/Data_608/output_csv/inflation_rates.csv"
# Write dataframe to CSV
write.csv(cpi_df, file = file_path, row.names = FALSE)

# Confirm that the file was saved
print("File saved successfully!")
```

```
[1] "File saved successfully!"
```

```
#cpi_df
```

Preparing FED Dataset

```
#fed_data
```

Convert "observation_date" to a Date format

```
fed_data <- fed_data %>%
  mutate(observation_date = mdy(observation_date))  # Converts MM/DD/YYYY format to Date typ

fed_df <- fed_data
#fed_df
```

Extract Year and Month

```r
fed_df <- fed_df %>%
  mutate(Year = year(observation_date),
         Month = month(observation_date, label = TRUE, abbr = TRUE))  # Extract month name (

#fed_df
```

Saving as CSV

```r
# Define the file path with filename and extension
#file_path <- "C:/Users/Uzma/Downloads/new_df.csv"
file_path <- "C:/Users/Uzma/CUNY-SPS-Assignments/Data_608/output_csv/fed_rates.csv"
# Write dataframe to CSV
write.csv(fed_df, file = file_path, row.names = FALSE)

# Confirm that the file was saved
#print("File saved successfully!")
```

Creating Dataframe for Unemployment Rate dataset

```r
unemp_df <- unemployment_data

#unemp_df
```

Extract the Month from the Label column

```r
unemp_df <- unemp_df %>%
  mutate(Month = word(Label, 2))  # Extract the second word (month name) from "1999 Jan"

#unemp_df
```

Arrange dataset in chronological order

```r
unemp_df <- unemp_df %>%
  arrange(Year, Month, unemployment_rate)

#unemp_df
```

Create a new column 'Status' based on 'unemployment_rate'

```r
unemp_df <- unemp_df %>%
  mutate(unemp_criteria = ifelse(unemployment_rate > 6, "Not Achieved", "Achieved"))

#unemp_df
```

Saving as CSV

```r
# Define the file path with filename and extension
#file_path <- "C:/Users/Uzma/Downloads/new_df.csv"
file_path <- "C:/Users/Uzma/CUNY-SPS-Assignments/Data_608/output_csv/unemp_rates.csv"
# Write dataframe to CSV
write.csv(unemp_df, file = file_path, row.names = FALSE)

# Confirm that the file was saved
#print("File saved successfully!")
```

Visualizing FED's Mandate Fulfillment

The Federal Reserve (FED) has a dual mandate from Congress:

Stable Prices (Low Inflation) → Inflation rate around 2%

Maximum Employment (Low Unemployment) → Low unemployment (~5% or lower)

Load Libraries

```r
library(ggplot2)
library(dplyr)
library(readr)
library(scales)
```

```
Warning: package 'scales' was built under R version 4.3.3


Attaching package: 'scales'

The following object is masked from 'package:purrr':

    discard

The following object is masked from 'package:readr':

    col_factor
```

Remove Year 1999

```r
unemp_df <- unemp_df %>% filter(Year != 1999)
cpi_df <- cpi_df %>% filter(Year != 1999)
fed_df <- fed_df %>% filter(Year != 1999)
```

Data Preparation: Merge All Three Datasets

```r
merged_df <- unemp_df %>%
  inner_join(cpi_df, by = c("Year", "Month")) %>%
  inner_join(fed_df, by = c("Year", "Month"))

#merged_df
```

Creating a Status Field

```r
merged_df <- merged_df %>%
  mutate(
    status = case_when(
      unemp_criteria == "Achieved" & inflation_criteria == "Achieved" ~ "Yes",
      unemp_criteria == "Not Achieved" & inflation_criteria == "Not Achieved" ~ "No",
      TRUE ~ "Partial Achieved"
    )
  )

#merged_df
```

Saving as CSV

```
# Define the file path with filename and extension
#file_path <- "C:/Users/Uzma/Downloads/new_df.csv"
file_path <- "C:/Users/Uzma/CUNY-SPS-Assignments/Data_608/output_csv/merged_data.csv"
# Write dataframe to CSV
#write.csv(merged_df, file = file_path, row.names = FALSE)

# Confirm that the file was saved
#print("File saved successfully!")
```

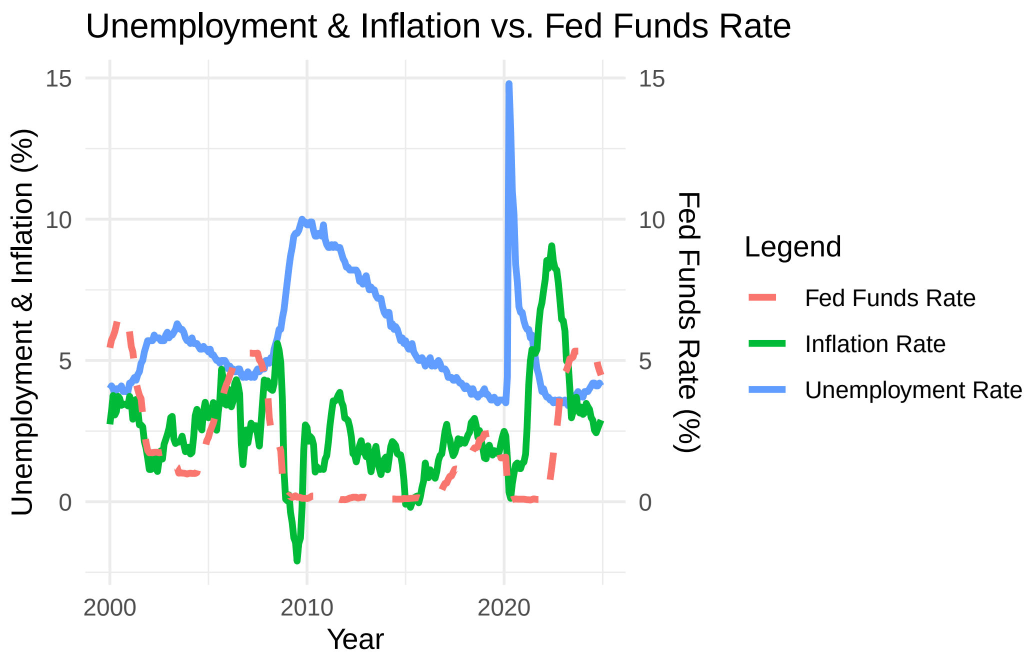
Visualization: Has the FED Fulfilled its Mandate?

Unemployment, Inflation & Fed Funds Rate Trend

This line chart shows how the unemployment rate, inflation rate, and Fed Funds rate have changed over time.

```
# Create a combined plot with dual y-axes
ggplot(merged_df, aes(x = as.Date(paste(Year, Month, "1", sep = "-"), "%Y-%b-%d"))) +
  geom_line(aes(y = unemployment_rate, color = "Unemployment Rate"), size = 1.2) +
  geom_line(aes(y = inflation_rate, color = "Inflation Rate"), size = 1.2) +
  geom_line(aes(y = fed_fund_rate, color = "Fed Funds Rate"), size = 1.2, linetype = "dashed
  scale_y_continuous(sec.axis = sec_axis(~., name = "Fed Funds Rate (%)")) +
  labs(title = "Unemployment & Inflation vs. Fed Funds Rate",
       x = "Year",
       y = "Unemployment & Inflation (%)",
       color = "Legend") +
  theme_minimal()
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

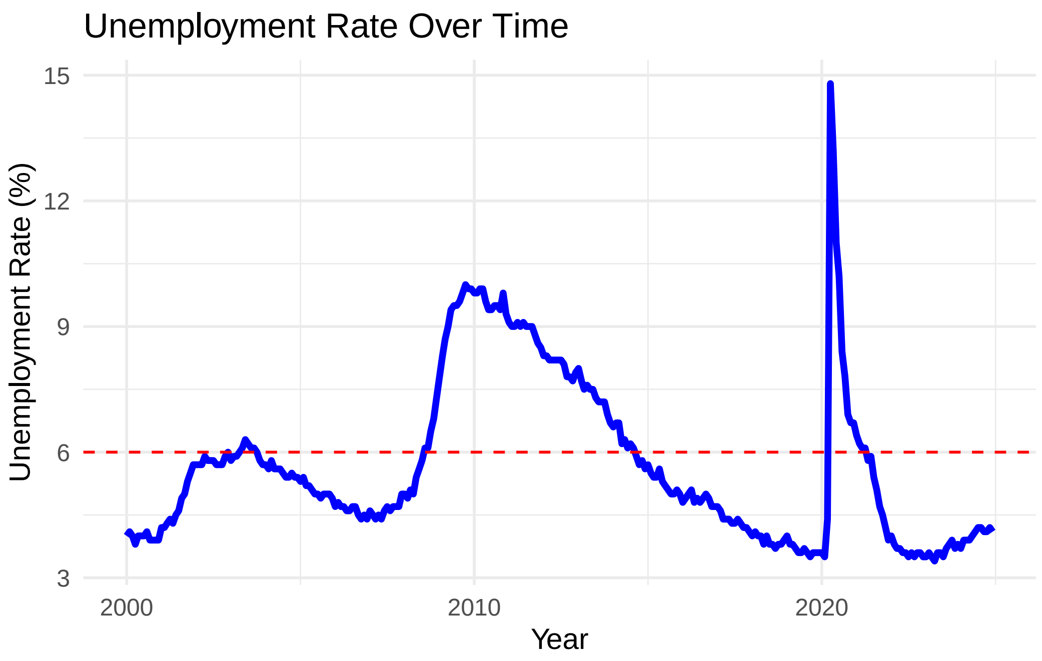


The Federal Reserve System has been given a dual mandate of pursuing the economic goals of maximum employment and price stability with a inflation rate of 2% over time and unemployment rate between 4% and 6% over time

"Has the FED been able to fulfill the mandate given to it by Congress?"

Unemployment Rate Over Time

```r
ggplot(merged_df, aes(x = as.Date(paste(Year, Month, "1", sep = "-"), "%Y-%b-%d"), y = unempl
  geom_line(color = "blue", size = 1.2) +
  geom_hline(yintercept = 6, linetype = "dashed", color = "red") +
  labs(title = "Unemployment Rate Over Time",
       x = "Year",
       y = "Unemployment Rate (%)") +
  theme_minimal()
```

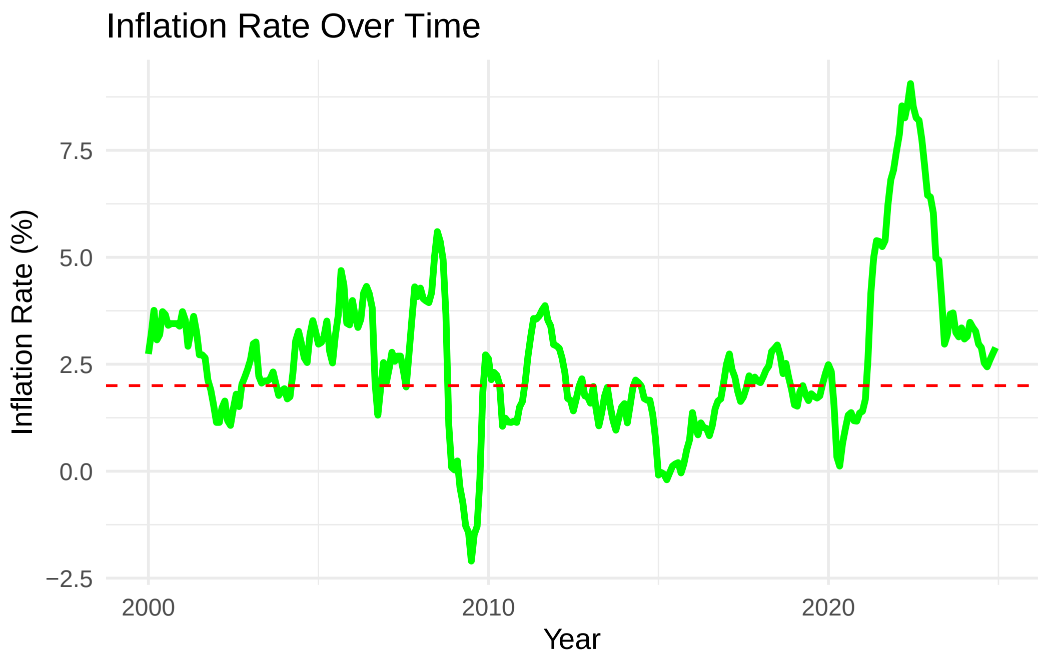


Interpretation:

A dashed red line at 6% shows the employment mandate If unemployment is below 6%, Fed is meeting its employment goal

Inflation Rate Over Time

```
ggplot(merged_df, aes(x = as.Date(paste(Year, Month, "1", sep = "-"), "%Y-%b-%d"), y = infla
  geom_line(color = "green", size = 1.2) +
  geom_hline(yintercept = 2, linetype = "dashed", color = "red") +
  labs(title = "Inflation Rate Over Time",
       x = "Year",
       y = "Inflation Rate (%)") +
  theme_minimal()
```



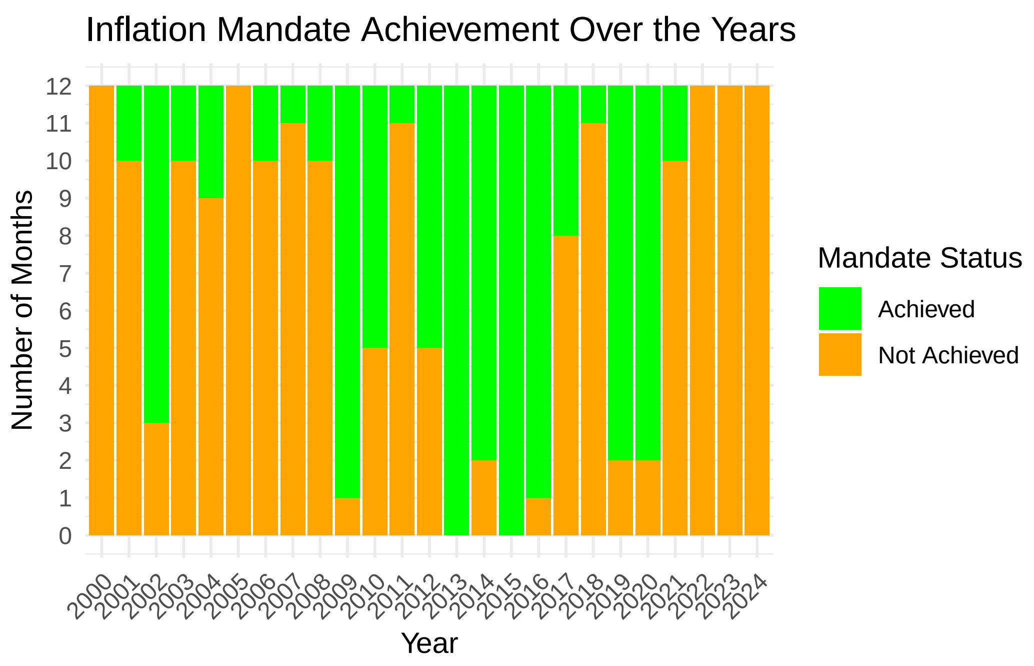
Interpretation:

A dashed red line at 2% shows the price stability mandate If inflation stays near 2%, Fed is achieving price stability

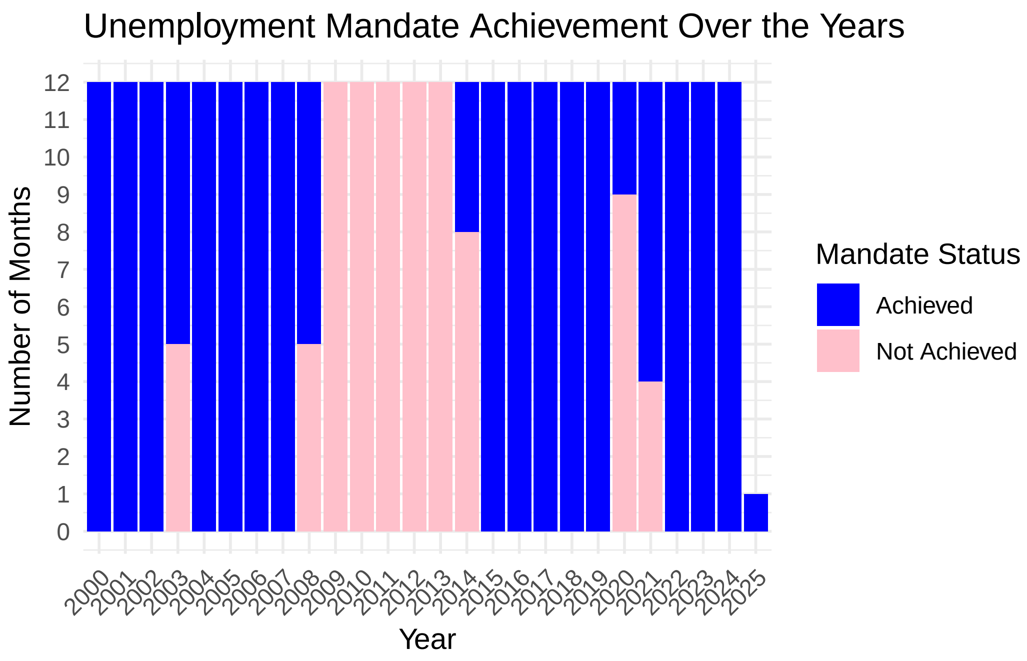
To evaluate whether the Federal Reserve (FED) has met its unemployment mandate, we need to visualize unemployment trends over time and compare them with the "Achieved" vs. "Not Achieved" status.

```
# Convert Month to numeric and create Date column
unemp_df <- unemp_df %>%
  mutate(
    Month = match(Month, month.abb),  # Convert "Jan" -> 1, "Feb" -> 2
    Date = as.Date(paste(Year, Month, "1", sep = "-"), format = "%Y-%m-%d"),
    unemployment_rate = as.numeric(unemployment_rate)  # Ensure numeric
  )
```

```
ggplot(cpi_df, aes(x = factor(Year), fill = inflation_criteria)) +
  geom_bar(position = "stack") +
  scale_fill_manual(values = c("Achieved" = "green", "Not Achieved" = "orange")) +
  scale_y_continuous(breaks = seq(0, 12, by = 1)) +  # Force whole numbers for months
  labs(title = "Inflation Mandate Achievement Over the Years",
       x = "Year",
       y = "Number of Months",
       fill = "Mandate Status") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for readal
```



```
ggplot(unemp_df, aes(x = factor(Year), fill = unemp_criteria)) +
  geom_bar(position = "stack") +
  scale_fill_manual(values = c("Achieved" = "blue", "Not Achieved" = "pink")) +
  scale_y_continuous(breaks = seq(0, 12, by = 1)) +  # Ensure y-axis is in whole numbers (0 
  labs(title = "Unemployment Mandate Achievement Over the Years",
       x = "Year",
       y = "Number of Months",
       fill = "Mandate Status") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels
```



Ensure merged_df is Properly Structured

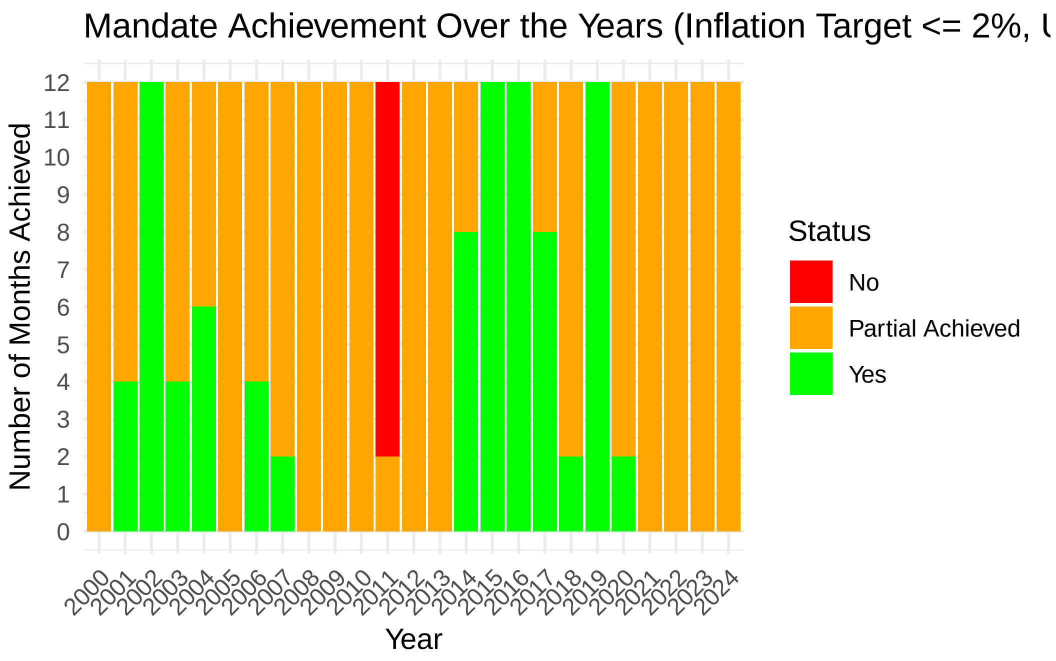
```
# Load necessary libraries
library(dplyr)
library(ggplot2)
library(tidyr)

# Ensure the Month column is a character type
merged_df <- merged_df %>%
  mutate(Month = as.character(Month))

# Convert to long format for better visualization
long_df <- merged_df %>%
  pivot_longer(cols = c(inflation_criteria, unemp_criteria),
               names_to = "Mandate",
               values_to = "Status")

# Rename "Mandate" column for clarity
long_df <- long_df %>%
  mutate(Mandate = ifelse(Mandate == "inflation_criteria", "Inflation Mandate", "Unemploymen
```

Create the Stacked Bar Chart

Now visualizing combined Inflation & Unemployment Mandate Achievements over the years.

```
library(dplyr)
library(ggplot2)

# Ensure Year is numeric
long_df$Year <- as.numeric(long_df$Year)

# Count unique months per year (avoid double-counting Inflation & Unemployment)
aggregated_df <- long_df %>%
  group_by(Year, Month, status) %>%
  summarise(count = n(), .groups = "drop")  # Count unique months per year

# Define custom colors
colors <- c("Yes" = "green",
            "No" = "red",
            "Partial Achieved" = "orange")

# Create the stacked bar chart
ggplot(aggregated_df, aes(x = factor(Year), y = count, fill = status)) +
  geom_col(position = "stack") +  # Use geom_col for pre-aggregated data
  scale_fill_manual(values = colors) +
  scale_y_continuous(limits = c(0, 12), breaks = 0:12) +  # Ensure Y-axis is 0-12 months
  labs(title = "Mandate Achievement Over the Years (Inflation Target <= 2%, Unemplyment Targ
       x = "Year",
       y = "Number of Months Achieved",
       fill = "Status") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for readal
```

```
Warning: Removed 150 rows containing missing values or values outside the scale range
(`geom_col()`).
```



Interpreting the Chart: Has the Federal Reserve (FED) Fulfilled Its Mandate?

The stacked bar chart represents the FED's mandate achievement over time (2000–2024), tracking whether it met its dual mandate of:

Stable Prices (Inflation Control) Maximum Employment The chart categorizes each month per year into:

Yes means Both inflation & employment criteria were achieved.

No means Neither was achieved. Partial Achieved – One was achieved, the other was not.