Capstone Visualizations

Zhora Stepanyan

2024-02-07

Refinancing Rate over time from 2003 to 2023

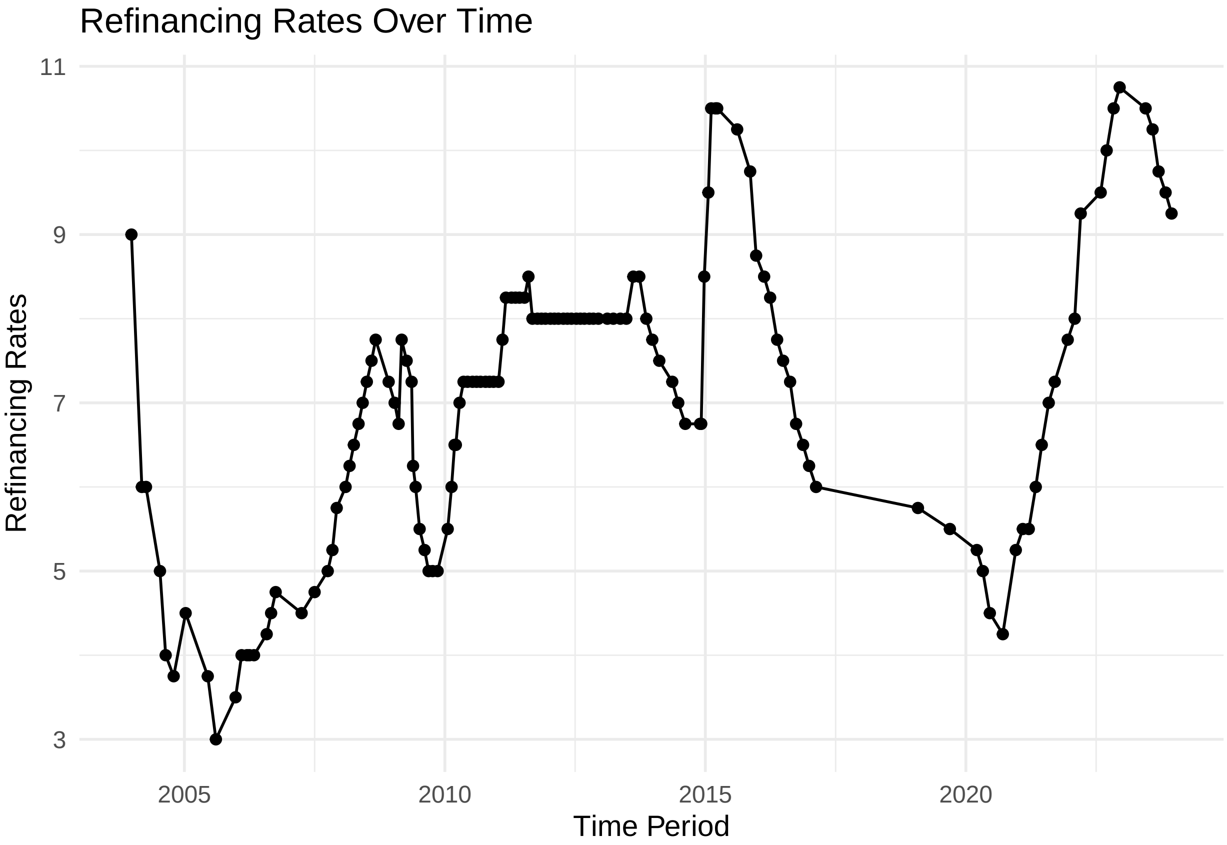
```
# Read the Excel file directly into a data frame
refinancing_rate <- read_xlsx('/Users/zhorastepanyan/Desktop/AUA/Spring_2024/Capstone/Data/8_policy_rat
```

```
## New names:
## * `` -> `...2`
## * `` -> `...3`
## * `` -> `...4`
```

```
# Remove the first two rows
refinancing_rate <- refinancing_rate[-(1:2), ]

# Set proper column names
colnames(refinancing_rate) <- c("Time_Period", "Refinancing_Rates", "Deposits_Attracted", "Lombard_Repo

# Convert Time_Period to Date type
refinancing_rate$Time_Period <- as.Date(refinancing_rate$Time_Period, format = "%d.%m.%Y")

# Remove rows with any missing values
refinancing_rate <- na.omit(refinancing_rate)

# Convert Refinancing_Rates to numeric
refinancing_rate$Refinancing_Rates <- as.numeric(refinancing_rate$Refinancing_Rates)

head(refinancing_rate)
```

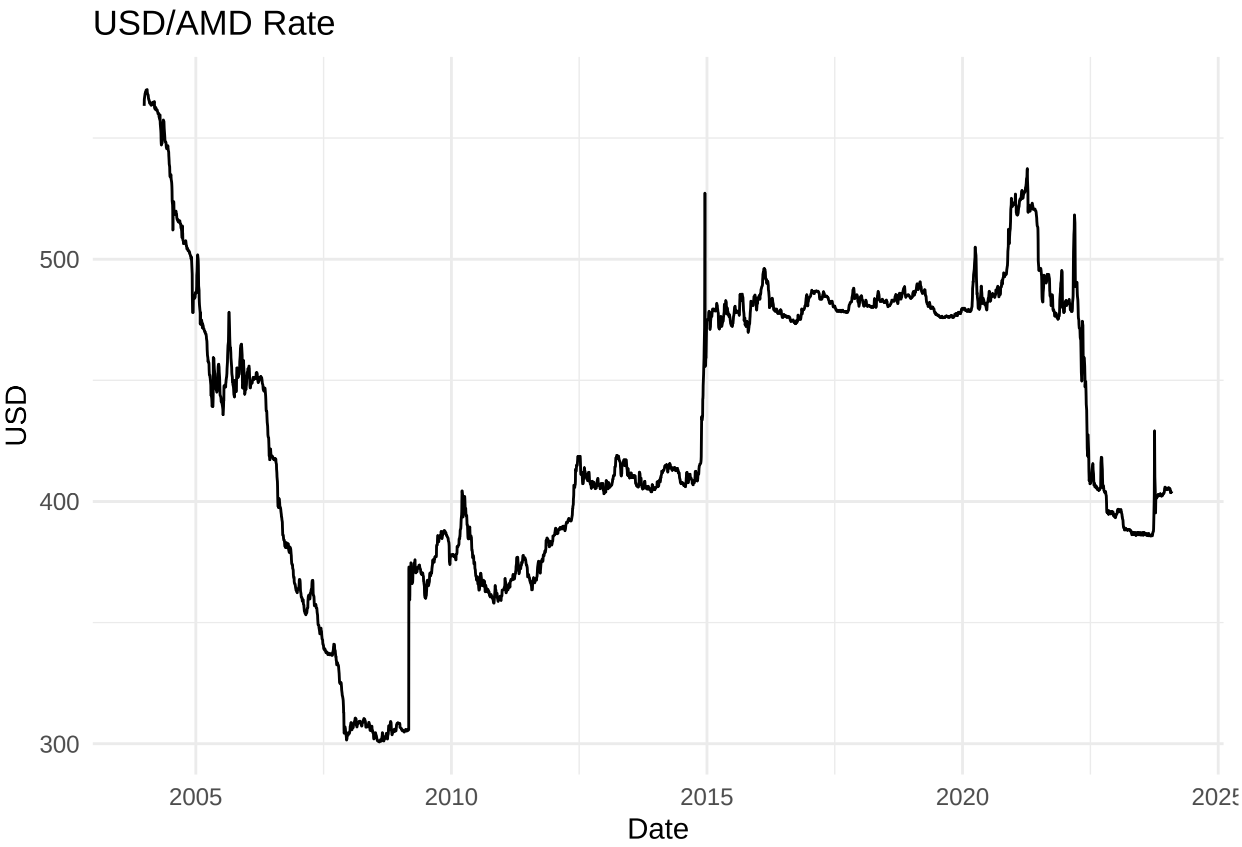
```
## # A tibble: 6 x 4
##   Time_Period Refinancing_Rates Deposits_Attracted Lombard_Repo1
##   <date>                  <dbl> <chr>              <chr>
## 1 2003-12-27               9    5                  20
## 2 2004-03-09               6    3                  20
## 3 2004-04-07               6    1                  20
## 4 2004-07-14               5    1                  20
## 5 2004-08-23               4    1                  16
## 6 2004-10-18               3.75 1                  16
```

```
# Create a line plot with points
ggplot(refinancing_rate, aes(x = Time_Period, y = Refinancing_Rates)) +
  geom_point() +
  geom_line() +
  labs(title = "Refinancing Rates Over Time",
       x = "Time Period",
       y = "Refinancing Rates") +
  theme_minimal()
```



USD/AMD Rate

```
usd_amd <- read_xlsx('/Users/zhorastepanyan/Desktop/AUA/Spring_2024/Capstone/Data/Rates-3.xlsx')


colnames(usd_amd) <- c("Date", "USD")



# Convert Time_Period to Date type
usd_amd$Date <- as.Date(usd_amd$Date, format = "%Y-%m-%d")

# Remove rows with any missing values
usd_amd <- na.omit(usd_amd)

usd_amd$USD <- as.numeric(usd_amd$USD)

ggplot(usd_amd, aes(x = Date, y = USD)) +
  geom_line() +
  labs(title = "USD/AMD Rate",
       x = "Date",
       y = "USD") +
  theme_minimal()
```



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.

*D1-D5 are compiled based on the indicators of the Central Bank of Armenia, commercial banks and credit organizations. D1 is the ratio of residents' foreign currency deposits and loans to the money supply D2 is the ratio of foreign currency deposits and loans of residents to total deposits and loans of residents D3 is the weight of foreign currency demand deposits and loans of resident natural persons in total demand deposits and loans of resident natural persons D4 is the weight of foreign currency time deposits and loans of resident individuals in total time deposits and loans of resident individuals D5 is the ratio of foreign currency loans to residents to total loans

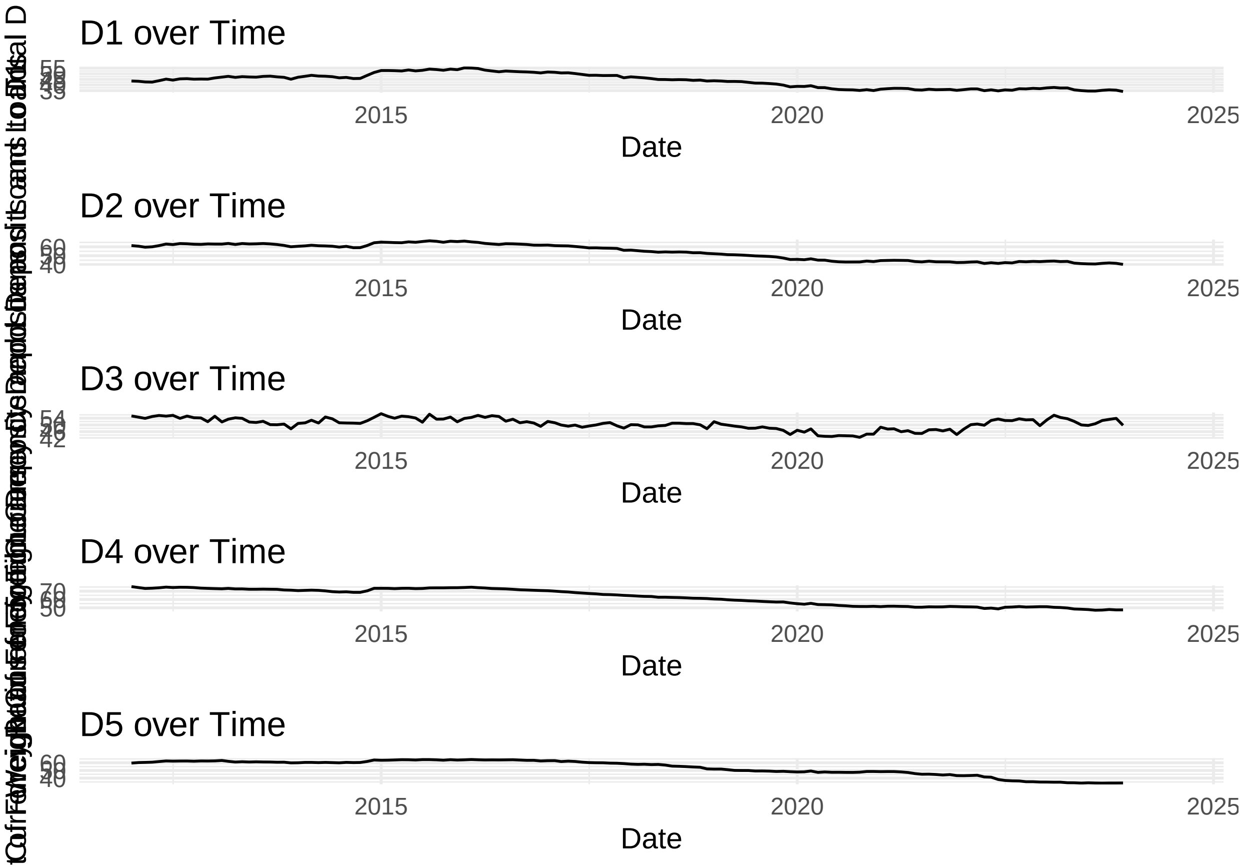
```
# Adjust 'skip' as needed based on your Excel file
dolarization <- read_xlsx('/Users/zhorastepanyan/Desktop/AUA/Spring_2024/Capstone/Data/16_Dollarization

dolarization$Date <- as.numeric(as.character(dolarization$Date))
dolarization$Date <- as.Date(dolarization$Date, origin = "2012-01-01")

# Number of rows in your data
n <- nrow(dolarization)

# Create a monthly sequence starting from January 2012
dolarization$Date <- seq(from = as.Date("2012-01-01"), by = "month", length.out = n)


head(dolarization)
```

```
## # A tibble: 6 x 6
##   Date          D1    D2    D3    D4    D5
##   <date>     <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2012-01-01  43.5  61.4  55.3  75.5  59.6
## 2 2012-02-01  43.3  60.7  54.6  74.3  60.3
## 3 2012-03-01  42.7  59.6  53.9  73.3  60.5
## 4 2012-04-01  42.6  60.0  55.0  73.6  60.8
## 5 2012-05-01  43.8  61.3  55.6  74.1  61.6
## 6 2012-06-01  45.2  63.1  55.2  74.9  62.4
```

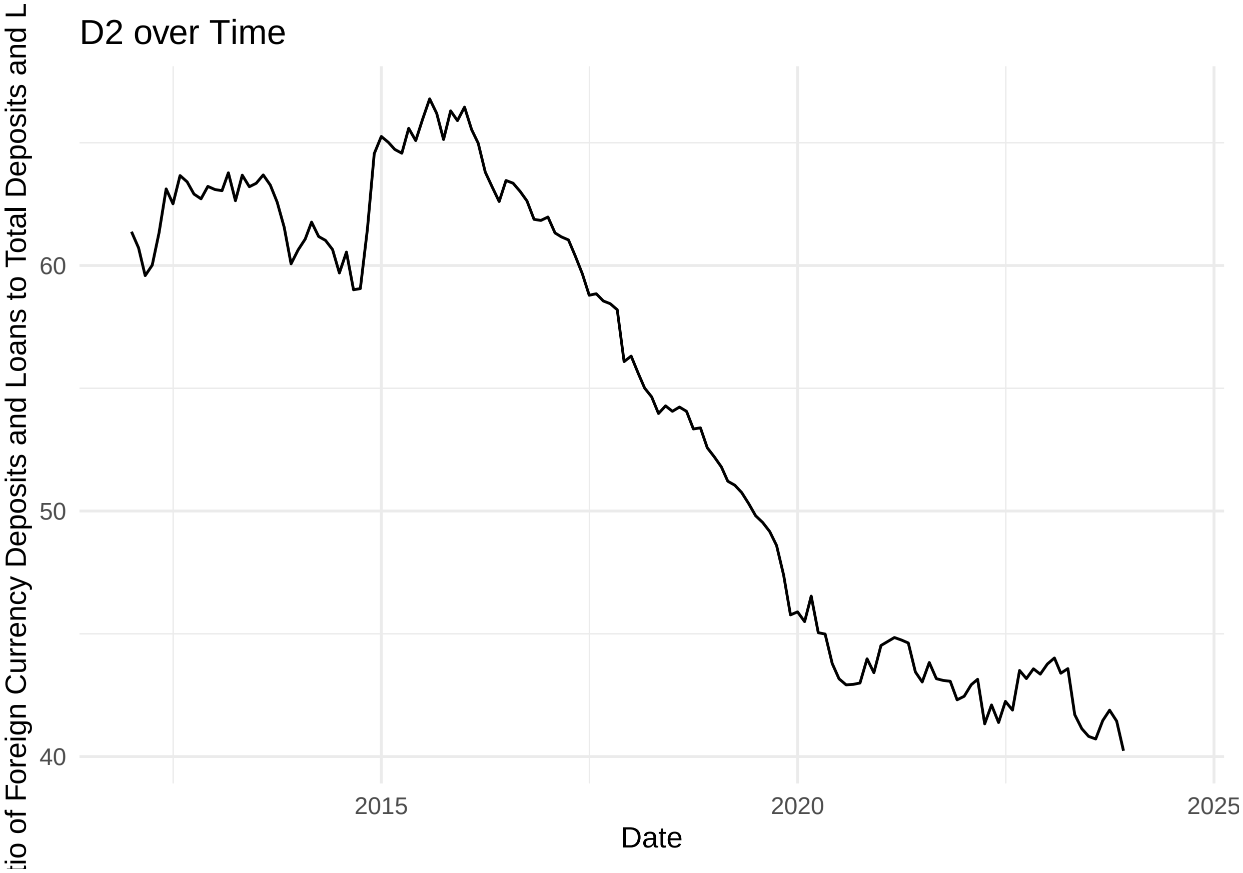
```r
library(ggplot2)

# Assuming your data is in a dataframe called dolarization and the Date column is properly formatted as



# Plot for D2
p2 <- ggplot(dolarization, aes(x = Date, y = D2)) +
  geom_line() +
  labs(title = "D2 over Time",
       y = "Ratio of Foreign Currency Deposits and Loans to Total Deposits and Loans") +
  theme_minimal()

# Plot for D3
p3 <- ggplot(dolarization, aes(x = Date, y = D3)) +
  geom_line() +
  labs(title = "D3 over Time",
       y = "Weight of Foreign Currency Demand Deposits and Loans") +
  theme_minimal()

# Plot for D4
p4 <- ggplot(dolarization, aes(x = Date, y = D4)) +
  geom_line() +
  labs(title = "D4 over Time",
       y = "Weight of Foreign Currency Time Deposits and Loans") +
  theme_minimal()

# Plot for D5
p5 <- ggplot(dolarization, aes(x = Date, y = D5)) +
  geom_line() +
  labs(title = "D5 over Time",
       x = "Date",
       y = "Ratio of Foreign Currency Loans to Total Loans") +
  theme_minimal()


library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
# Create the plots (but don't display them)
p1 <- ggplot(dolarization, aes(x = Date, y = D1)) +
  geom_line() +
  labs(title = "D1 over Time") +
  theme_minimal()

# ... create p2, p3, p4, p5 similarly ...

# Arrange the plots in a grid
grid.arrange(p1, p2, p3, p4, p5, nrow = 5)
```



```
print(p2)
```



1 Includes indicators of the Central Bank and commercial banks. Since 2008 includes also indicators of credit organizations. 2 M1 includes currency in circulation and demand deposits (including accounts) and borrowings in drams. 3 M2 is dram broad money, includes M1 and time deposits and borrowings in drams. 4 M2X is broad money, includes M2 and deposits (including accounts) and borrowings in foreign currency.

```
aggregates <- read_xlsx('/Users/zhorastepanyan/Desktop/AUA/Spring_2024/Capstone/Data/4_Aggregates_eng.xl
```

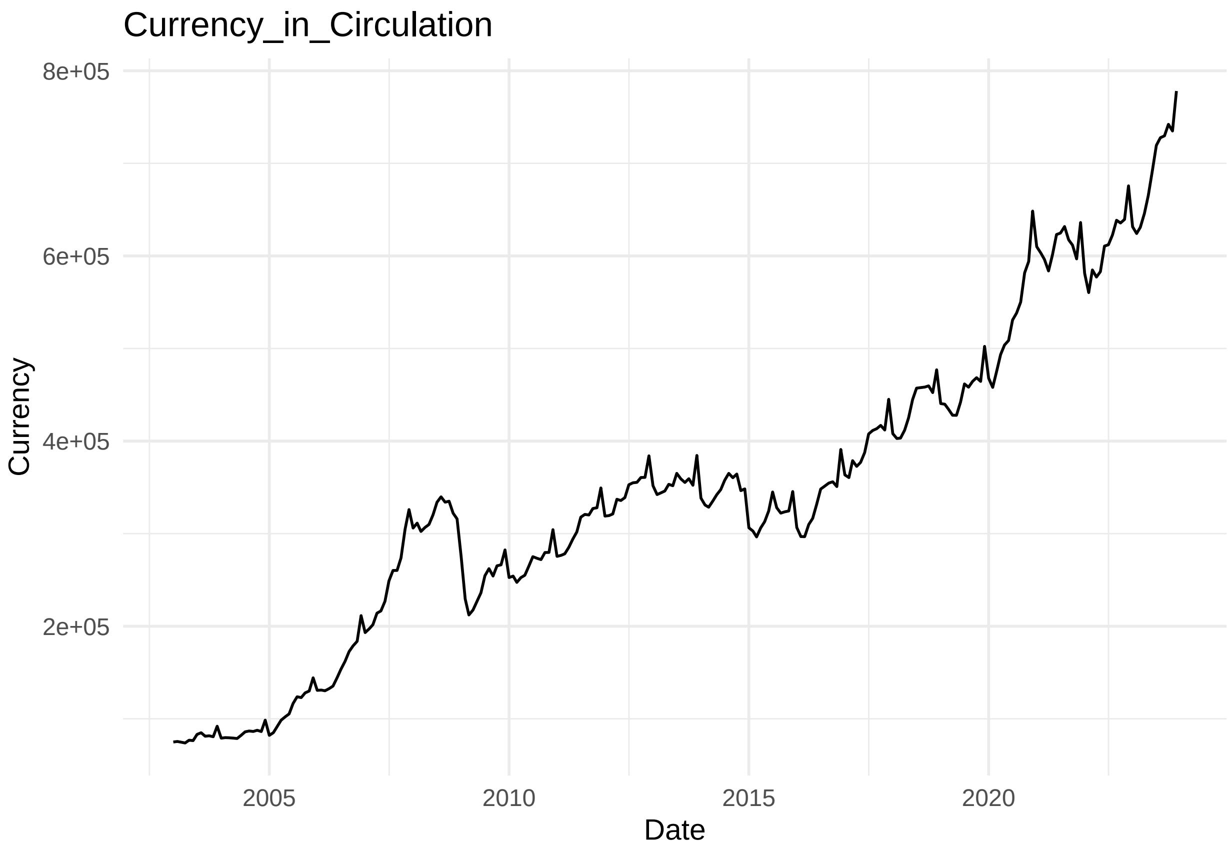
```
## New names:
## * `` -> `...2`
## * `` -> `...3`
## * `` -> `...4`
## * `` -> `...5`
## * `` -> `...6`
## * `` -> `...7`
## * `` -> `...8`
```

```
# Remove rows that are entirely NA
aggregates <- na.omit(aggregates)

# Set the column names as provided in the image
col_names <- c("Date","Currency_in_Circulation", "Demand_Deposits_in_Drams", "M1^2", "Time_Deposits_in_
names(aggregates) <- col_names

aggregates$Date <- as.numeric(as.character(aggregates$Date))
aggregates$Date <- as.Date(aggregates$Date, origin = "2003-01-01")

# Number of rows in your data
n <- nrow(aggregates)
```

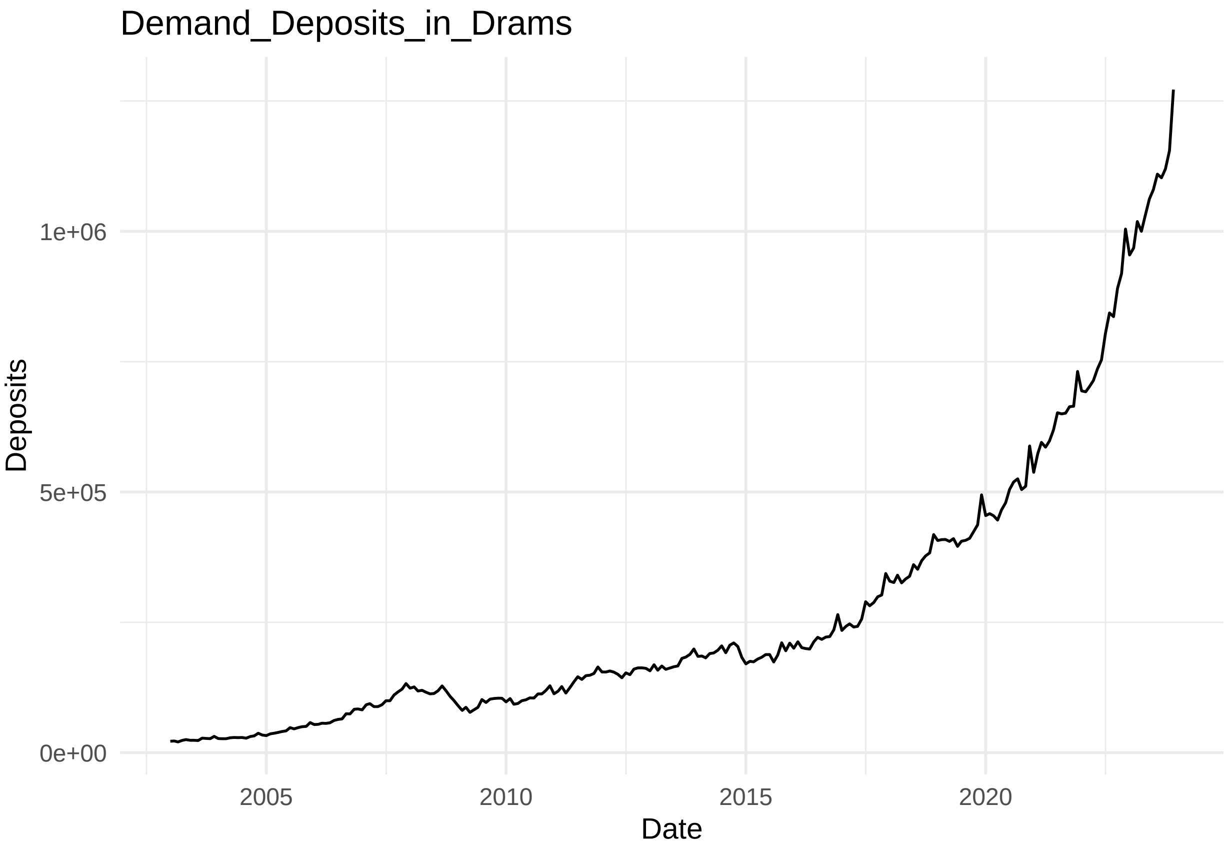
```r
# Create a monthly sequence starting from January 2003
aggregates$Date <- seq(from = as.Date("2003-01-01"), by = "month", length.out = n)

aggregates$Currency_in_Circulation <- as.numeric(aggregates$Currency_in_Circulation)
aggregates$Demand_Deposits_in_Drams <- as.numeric(aggregates$Demand_Deposits_in_Drams)
aggregates$`M1^2` <- as.numeric(aggregates$`M1^2`)
aggregates$Time_Deposits_in_Drams <- as.numeric(aggregates$Time_Deposits_in_Drams)
aggregates$`M2^3` <- as.numeric(aggregates$`M2^3`)
aggregates$`M2X^4` <- as.numeric(aggregates$`M2X^4`)


head(aggregates)
```

```
## # A tibble: 6 x 8
##   Date       Currency_in_Circulation Demand_Deposits_in_Drams  `M1^2`
##   <date>                       <dbl>                    <dbl>   <dbl>
## 1 2003-01-01                  74980.                   21954.  96935.
## 2 2003-02-01                  75523                    22351   97874
## 3 2003-03-01                  74801                    20608.  95409.
## 4 2003-04-01                  73890                    23499.  97389.
## 5 2003-05-01                  76892                    25103. 101995.
## 6 2003-06-01                  76479.                   23734. 100214.
## # i 4 more variables: Time_Deposits_in_Drams <dbl>, `M2^3` <dbl>,
## #   Deposits_in_Foreign_Currency <chr>, `M2X^4` <dbl>
```

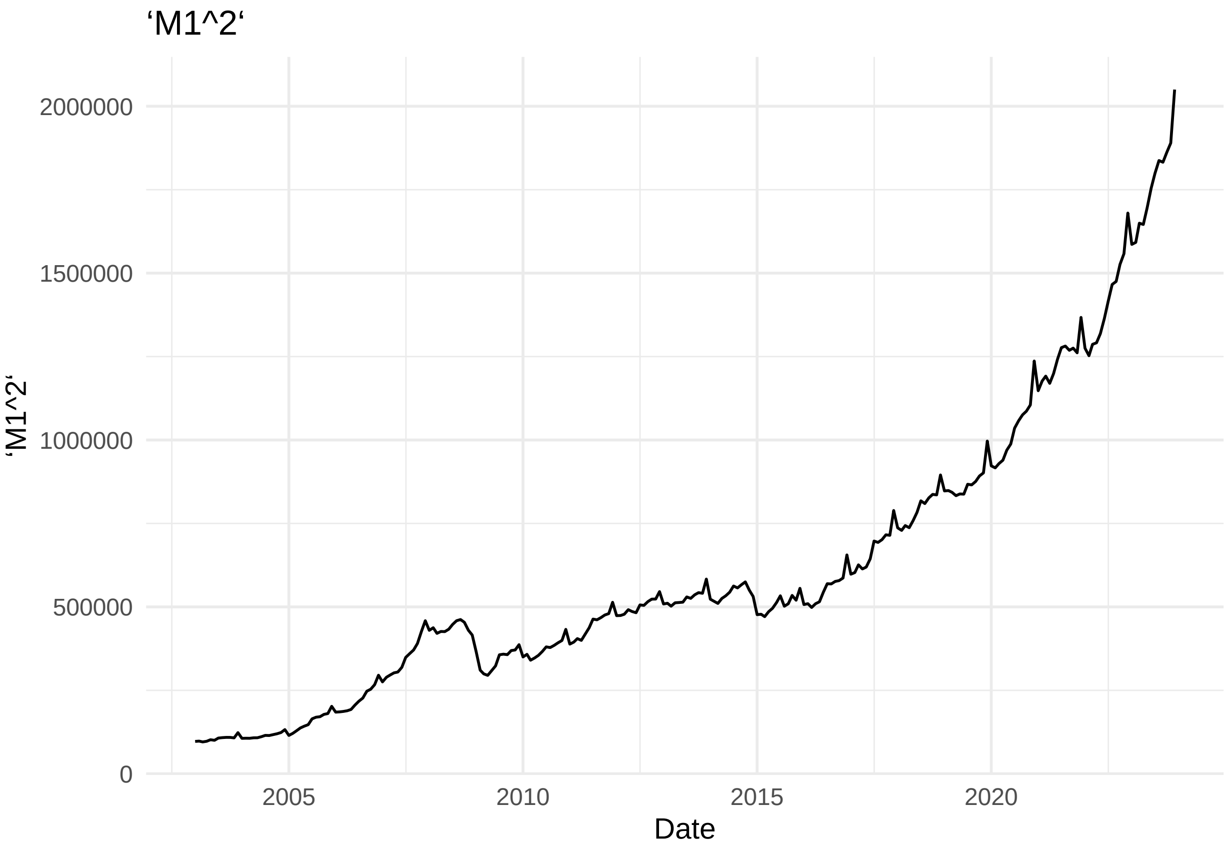
```r
ggplot(aggregates, aes(x = Date, y = Currency_in_Circulation)) +
  geom_line() +
  labs(title = "Currency_in_Circulation",
       x = "Date",
       y = "Currency") +
  theme_minimal()
```



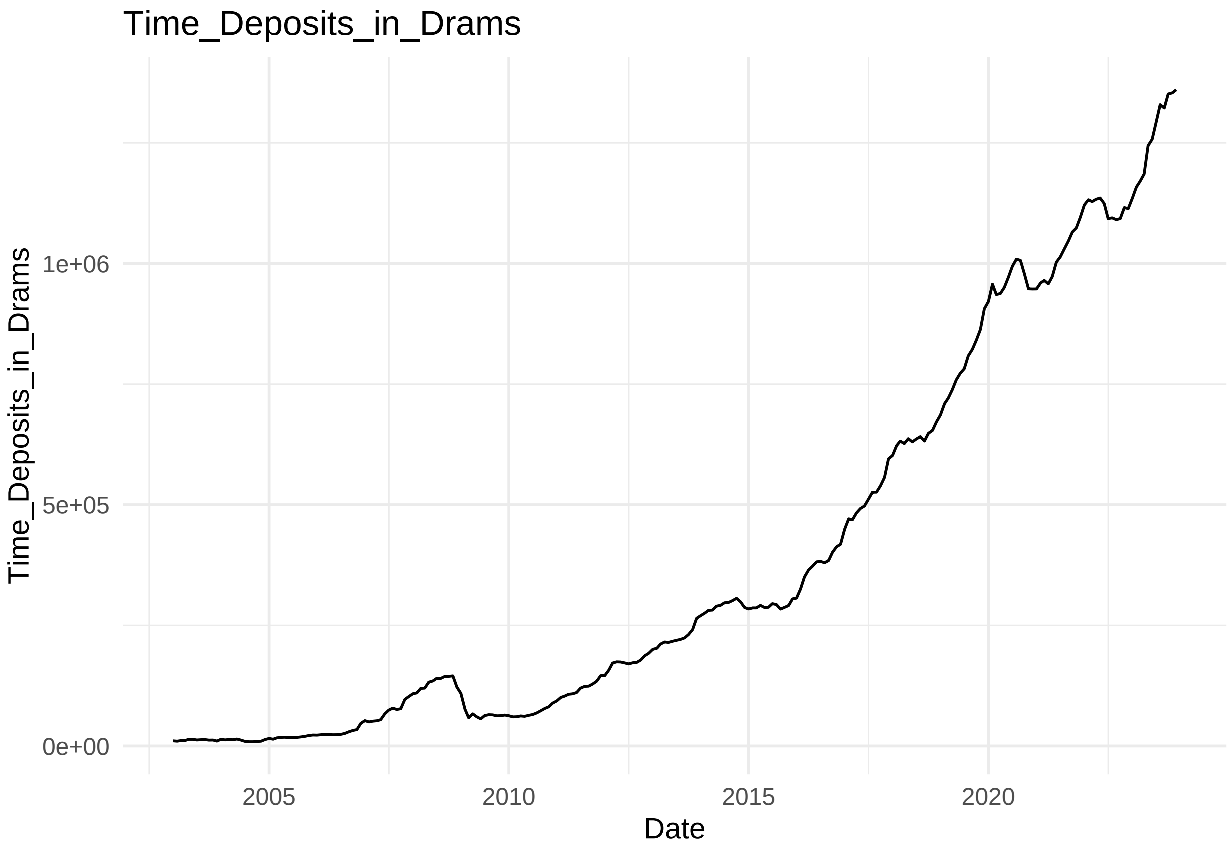
```
ggplot(aggregates, aes(x = Date, y = Demand_Deposits_in_Drams)) +
  geom_line() +
  labs(title = "Demand_Deposits_in_Drams",
       x = "Date",
       y = "Deposits") +
  theme_minimal()
```



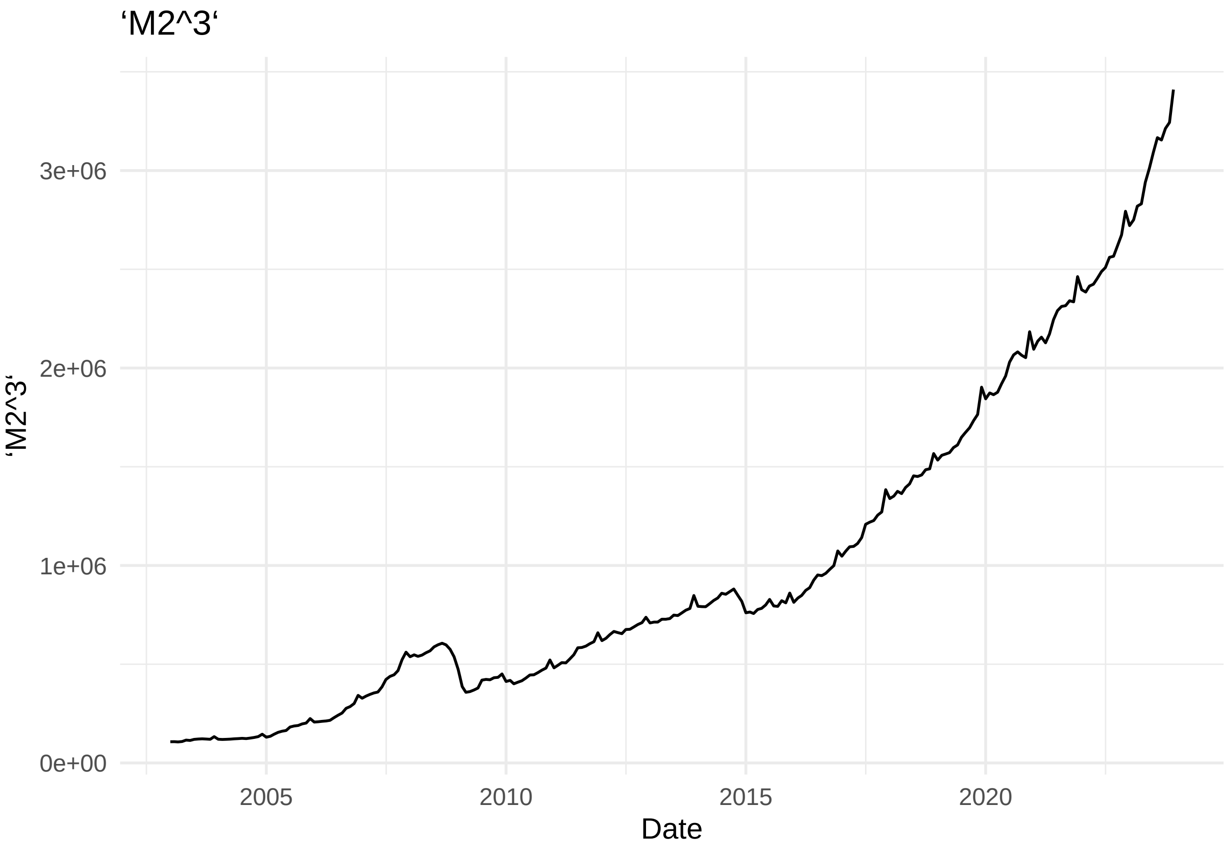
```r
ggplot(aggregates, aes(x = Date, y = `M1^2`)) +
  geom_line() +
  labs(title = "`M1^2`",
       x = "Date",
       y = "`M1^2`") +
  theme_minimal()
```



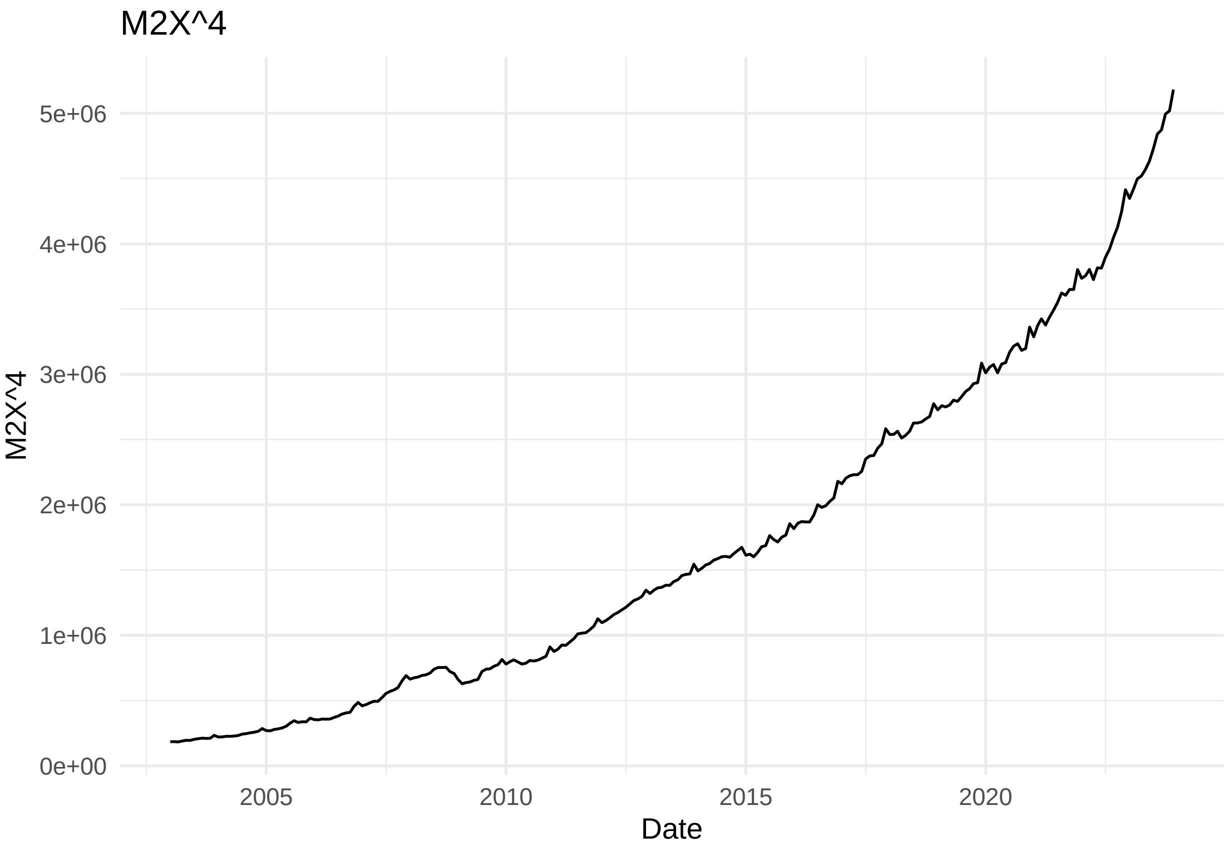
```
ggplot(aggregates, aes(x = Date, y = Time_Deposits_in_Drams)) +
  geom_line() +
  labs(title = "Time_Deposits_in_Drams",
       x = "Date",
       y = "Time_Deposits_in_Drams") +
  theme_minimal()
```



```
ggplot(aggregates, aes(x = Date, y = `M2^3`)) +
  geom_line() +
  labs(title = "`M2^3`",
       x = "Date",
       y = "`M2^3`") +
  theme_minimal()
```



```r
ggplot(aggregates, aes(x = Date, y = `M2X^4`)) +
  geom_line() +
  labs(title = "M2X^4",
       x = "Date",
       y = "M2X^4") +
  theme_minimal()
```



```
# Step 0: Install and load necessary package

# Step 1: Read the data from Excel file
core_inflation <- read_excel("/Users/zhorastepanyan/Desktop/AUA/Spring_2024/Capstone/Data/core_inflatio
```

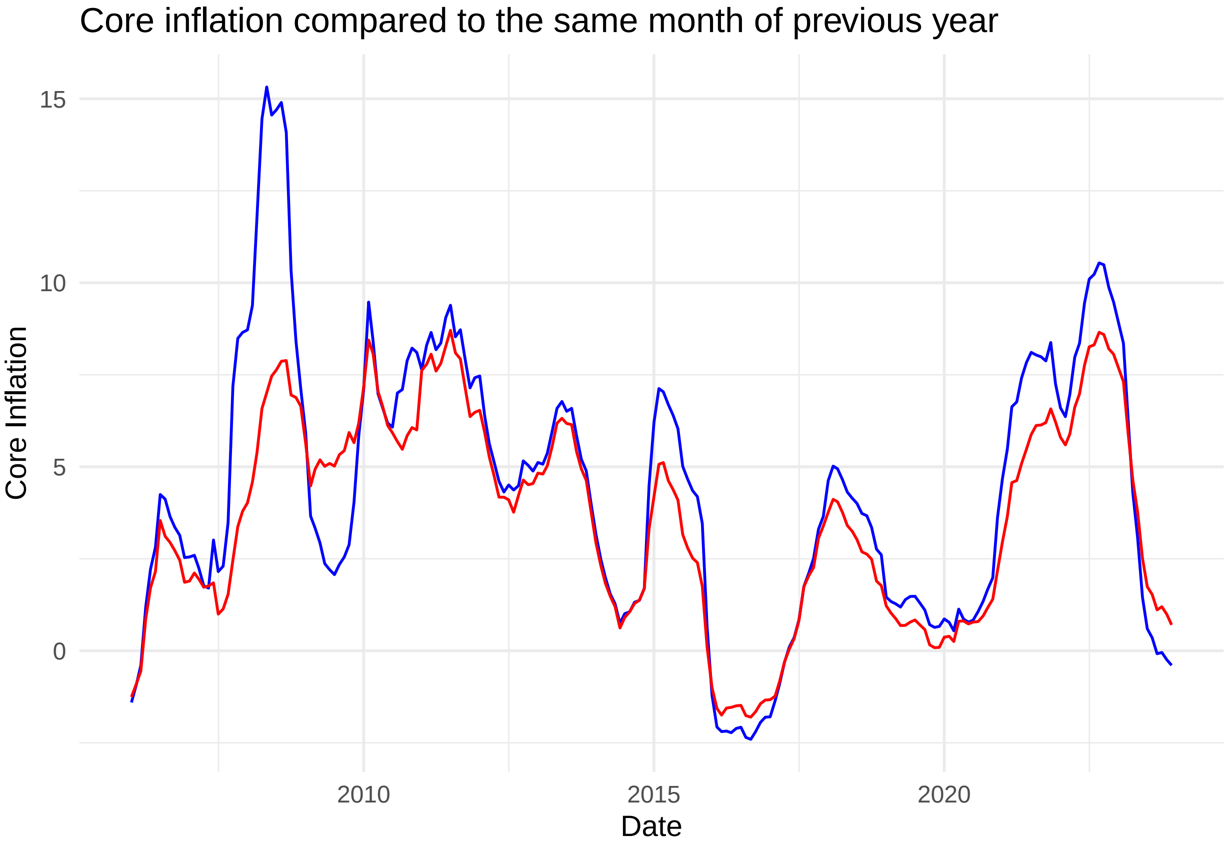
```
## New names:
## * `` -> `...1`
## * `compared to the same month of the previous year` -> `compared to the same
##   month of the previous year...2`
## * `compared to the previous month` -> `compared to the previous month...3`
## * `compared to the same month of the previous year` -> `compared to the same
##   month of the previous year...4`
## * `compared to the previous month` -> `compared to the previous month...5`
## * `compared to the same month of the previous year` -> `compared to the same
##   month of the previous year...6`
## * `compared to the previous month` -> `compared to the previous month...7`
```

```
core_inflation <- core_inflation[,c(1,2,4)]

col_names <- c("Date","Core_Inflation_1", "Core_Inflation_2")
names(core_inflation) <- col_names

core_inflation$Date <- as.Date(paste0(core_inflation$Date, "/1"), format="%Y/%m/%d")

core_inflation$Core_Inflation_1 <- as.numeric(core_inflation$Core_Inflation_1)
core_inflation$Core_Inflation_2 <- as.numeric(core_inflation$Core_Inflation_2)

core_inflation$Core_Inflation_1 <- core_inflation$Core_Inflation_1 - 100
```

```
core_inflation$Core_Inflation_2 <- core_inflation$Core_Inflation_2 - 100



# Now, the data is in a more structured and clean format
head(core_inflation)
```

```
## # A tibble: 6 x 3
##   Date       Core_Inflation_1 Core_Inflation_2
##   <date>                <dbl>            <dbl>
## 1 2006-01-01           -1.41            -1.26
## 2 2006-02-01           -0.917           -0.898
## 3 2006-03-01           -0.396           -0.552
## 4 2006-04-01            1.19             0.872
## 5 2006-05-01            2.22             1.71
## 6 2006-06-01            2.81             2.15
```

```
ggplot(core_inflation, aes(x = Date)) +
  geom_line(aes(y = Core_Inflation_1), color = "blue") + # First line for Core_Inflation_1
  geom_line(aes(y = Core_Inflation_2), color = "red") +  # Second line for Core_Inflation_2
  labs(title = "Core inflation compared to the same month of previous year",
       x = "Date",
       y = "Core Inflation") +
  theme_minimal()
```



```
# Step 0: Install and load necessary package

# Step 1: Read the data from Excel file
```

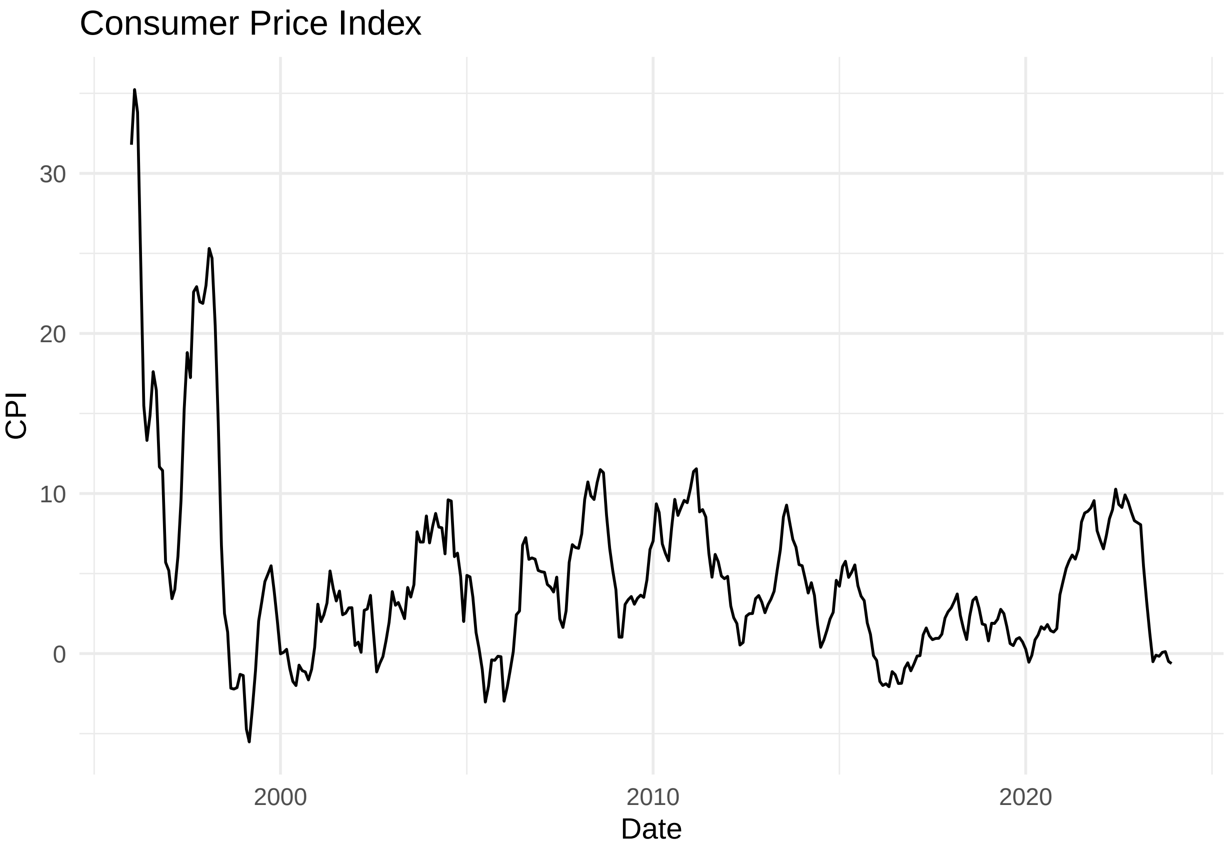
```
cpi <- read_excel("/Users/zhorastepanyan/Desktop/AUA/Spring_2024/Capstone/Data/6_CPI_eng.xls", skip = 3
```

```
## New names:
## * `` -> `...1`
## * `` -> `...6`
```

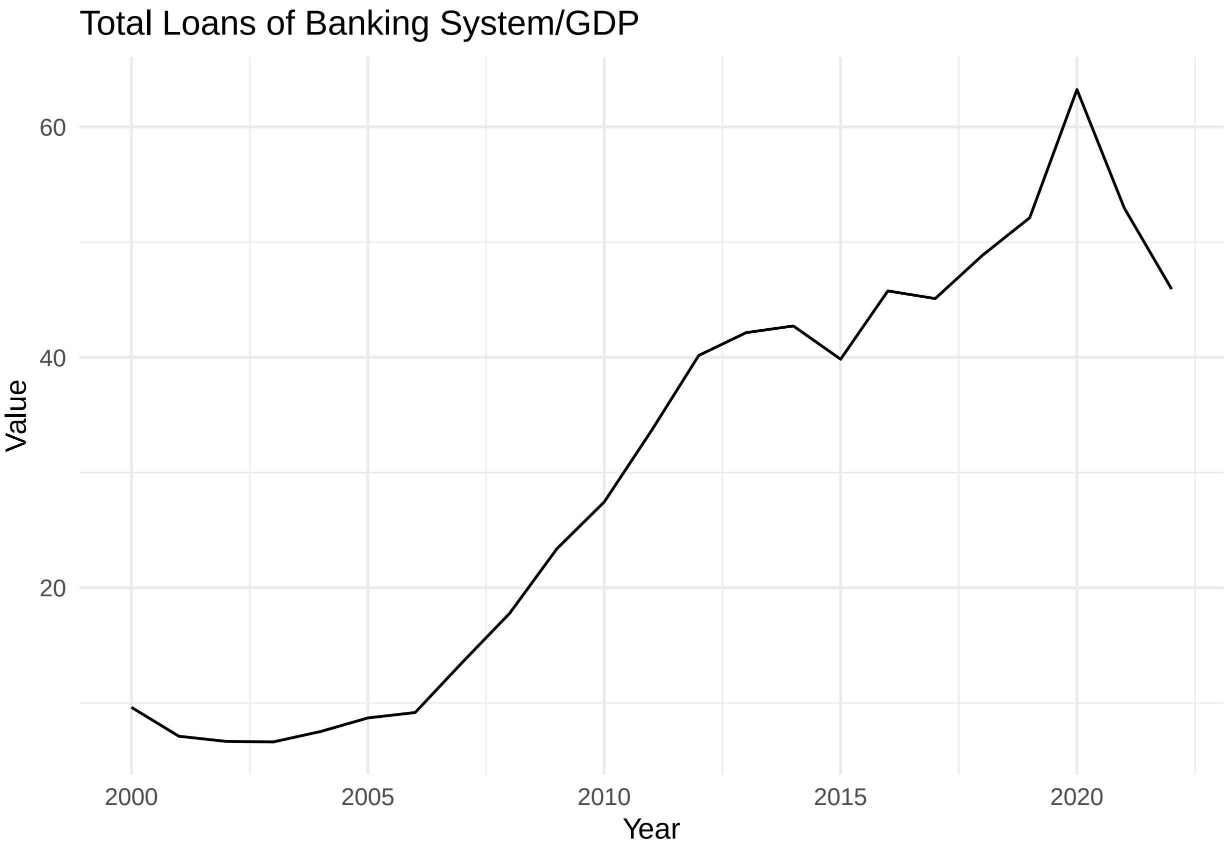
```
cpi <- cpi[,c(1,8)]

col_names <- c("Date","CPI")
names(cpi) <- col_names

cpi$Date <- as.Date(paste0(cpi$Date, "/1"), format="%Y/%m/%d")

cpi$CPI <- cpi$CPI - 100

head(cpi)
```

```
## # A tibble: 6 x 2
##   Date         CPI
##   <date>     <dbl>
## 1 1996-01-01  31.8
## 2 1996-02-01  35.2
## 3 1996-03-01  33.8
## 4 1996-04-01  24.5
## 5 1996-05-01  15.5
## 6 1996-06-01  13.3
```

```
ggplot(cpi, aes(x = Date)) +
  geom_line(aes(y = CPI)) + # First line for Core_Inflation_1
  labs(title = "Consumer Price Index",
       x = "Date",
       y = "CPI") +
  theme_minimal()
```



```r
# Step 0: Install and load necessary package

# Step 1: Read the data from Excel file
bank_gdp <- read_excel("/Users/zhorastepanyan/Desktop/AUA/Spring_2024/Capstone/Data/Banks-GDP-Eng.xls",

# Reshape the data to long format for the second indicator only
long_data <- bank_gdp %>%
  filter(row_number() == 2) %>% # Select only the second row (Total loans of banking system /GDP)
  pivot_longer(
    cols = -1, # Exclude the first column (Indicator names)
    names_to = "Year",
    values_to = "Value",
    names_transform = list(Year = as.integer) # Transform the year names to integer
  )

# The names_transform argument converts the year names from characters to integers

bank_gdp <- long_data %>% select(-1)

bank_gdp$Year <- as.Date(paste0(bank_gdp$Year, "-01-01"))

ggplot(bank_gdp, aes(x = Year)) +
  geom_line(aes(y = Value)) + # First line for Core_Inflation_1
  labs(title = "Total Loans of Banking System/GDP",
       x = "Year",
       y = "Value") +
  theme_minimal()
```



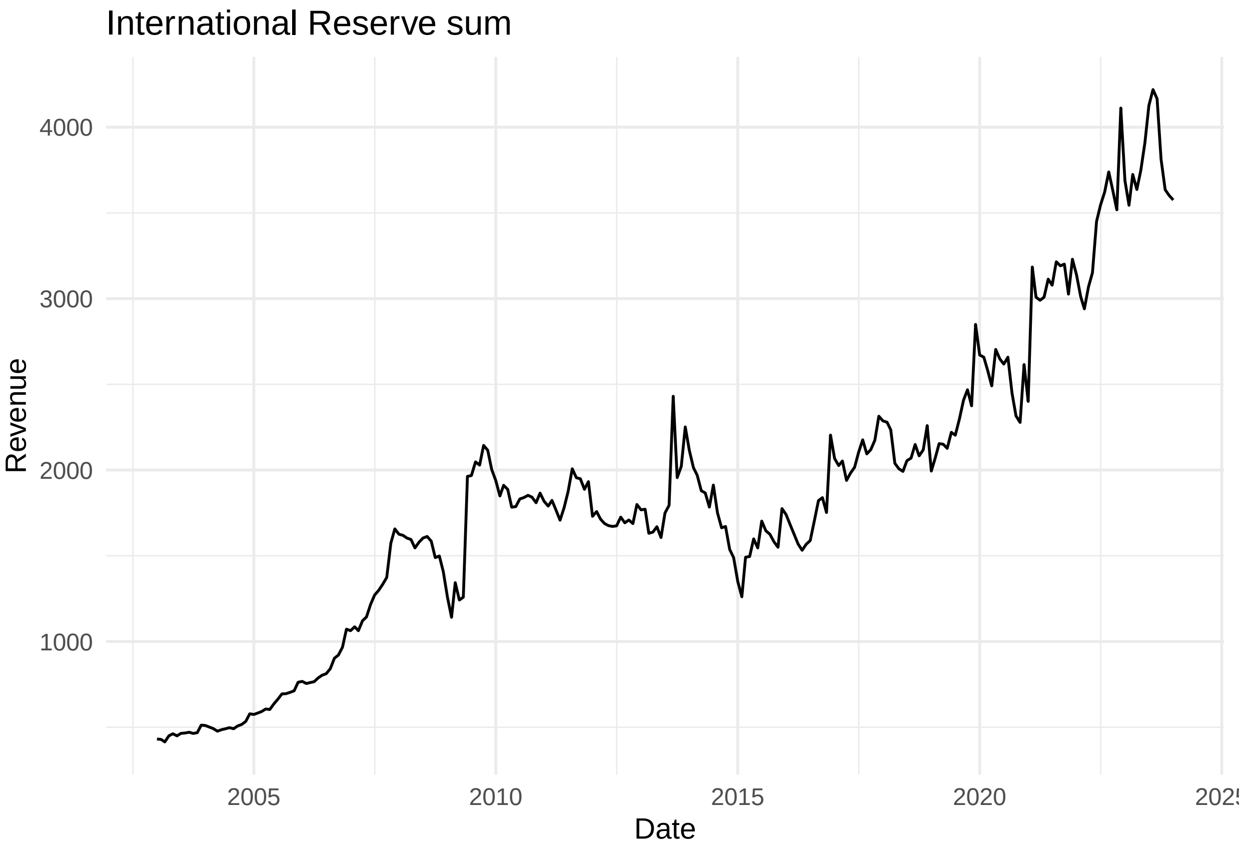
```r
# View the reshaped data
print(bank_gdp)
```

```
## # A tibble: 23 x 2
##    Year       Value
##    <date>     <dbl>
##  1 2000-01-01  9.62
##  2 2001-01-01  7.11
##  3 2002-01-01  6.67
##  4 2003-01-01  6.62
##  5 2004-01-01  7.52
##  6 2005-01-01  8.70
##  7 2006-01-01  9.17
##  8 2007-01-01 13.5
##  9 2008-01-01 17.8
## 10 2009-01-01 23.4
## # i 13 more rows
```

```r
bop <- read_excel("/Users/zhorastepanyan/Desktop/AUA/Spring_2024/Capstone/Data/7.International reserves_
```

```
## New names:
## * `` -> `...1`
```

```r
# Reshape the data to long format for the second indicator only
long_data <- bop %>%
  filter(row_number() == 2) %>% # Select only the second row (Total loans of banking system /GDP)
  pivot_longer(
    cols = -1, # Exclude the first column (Indicator names)
    names_to = "Date",
    values_to = "Gross_International_Revenue",
    names_transform = list(Year = as.integer) # Transform the year names to integer
  )

bop <- long_data %>% select(-1)


n <- nrow(bop)
bop$Date <- seq(from = as.Date("2003-01-01"), by = "month", length.out = n)

ggplot(bop, aes(x = Date)) +
  geom_line(aes(y = Gross_International_Revenue)) + # First line for Core_Inflation_1
  labs(title = "International Reserve sum",
       x = "Date",
       y = "Revenue") +
  theme_minimal()
```



```
print(bop)
```

```
## # A tibble: 253 x 2
##    Date       Gross_International_Revenue
##    <date>                           <dbl>
##  1 2003-01-01                        431.
##  2 2003-02-01                        429.
##  3 2003-03-01                        414.
##  4 2003-04-01                        450.
##  5 2003-05-01                        462.
##  6 2003-06-01                        450.
##  7 2003-07-01                        464.
##  8 2003-08-01                        466.
##  9 2003-09-01                        471.
## 10 2003-10-01                        464.
## # i 243 more rows
```