Vizualizations for the presentation

Contents

```r
library(ggplot2)
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(grid)
library(png)
```

Colors

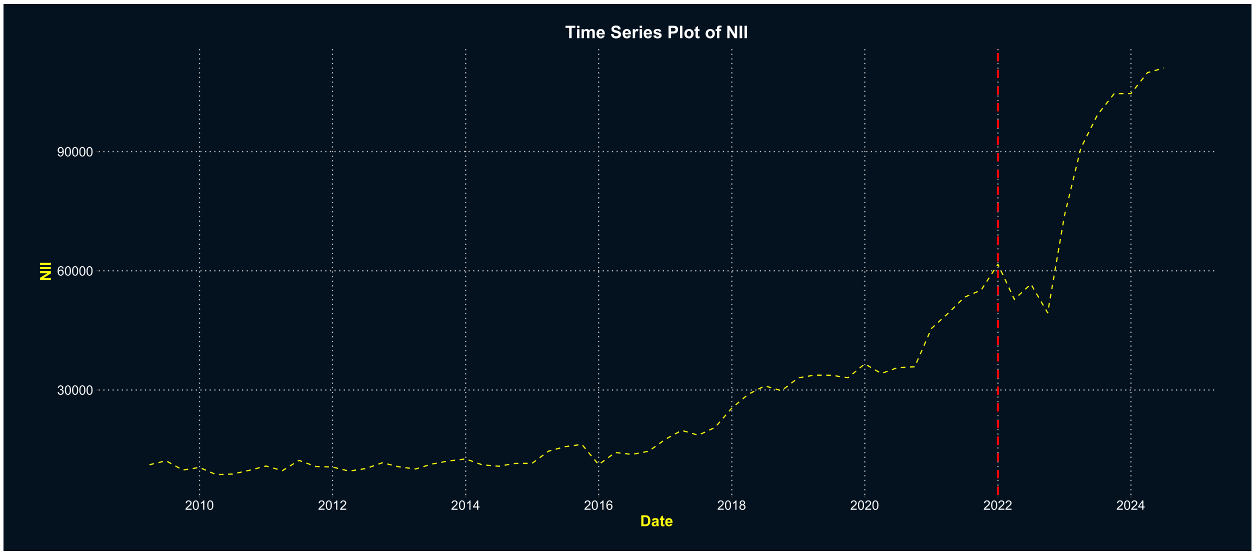
```r
background = "#03182bff"
complement = "yellow"
grey = "cyan"
fill_col = "grey50"
text_col = "white"
grid_major_col = "grey80"
grid_minor_col = background
width = 16
height = 7
dpi = 400

theme_custom <- function() {
  theme(
    # Set background color for the entire plot and panel
    plot.background = element_rect(fill = background, color = NA),  # Entire plot background
    panel.background = element_rect(fill = background, color = NA),  # Plot panel background
    # Set grid lines
    panel.grid.major = element_line(color = grid_major_col, linetype = "dotted"),
    panel.grid.minor = element_line(color = grid_minor_col, linetype = "dotted", linewidth = 0.6),
    # Customize axis titles and text
    axis.title.x = element_text(size = 14, face = "bold", color = complement),
    axis.title.y = element_text(size = 14, face = "bold", color = complement),
    axis.text = element_text(size = 12, color = text_col),
    # Customize plot title
    plot.title = element_text(size = 16, face = "bold", color = text_col, hjust = 0.5),
    # Customize legend (if applicable)
    legend.background = element_rect(fill = background, color = NA),  # Legend background
    legend.text = element_text(size = 12, color = text_col),
    legend.title = element_text(size = 12, color = text_col),
    plot.margin = margin(20, 20, 20, 20),
    # Set aspect ratio
    aspect.ratio = 0.4
  )
}
display_plot <- function(filepath) {
  img <- readPNG(filepath)  # Load the image
  grid.raster(img)          # Display the image
}
```

```
plot_median_quartiles <- function(file_path, name="~/Documents/GitHub/MonetaryPolicyEffectOnNetInterest
  # Read the CSV file
  df <- read.csv(file_path, stringsAsFactors = FALSE)

  # Convert Date column to Date type
  df$Date <- as.Date(df$Date)

  # Ensure all columns are numeric or NA, excluding the Date column
  df_clean <- df %>%
    mutate(across(-Date, ~ as.numeric(as.character(.)), .names = "numeric_{col}"))

  # Drop non-numeric columns and keep only the Date and numeric columns
  df_numeric <- df_clean %>%
    select(Date, starts_with("numeric_")) %>%
    rename_with(~ sub("numeric_", "", .))

  # Convert to long format
  df_long <- df_numeric %>%
    pivot_longer(cols = -Date, names_to = "Bank", values_to = "Value")

  # Compute median and quartiles by date
  summary_stats <- df_long %>%
    group_by(Date) %>%
    summarise(
      Median = median(Value, na.rm = TRUE),
      Q1 = quantile(Value, 0.25, na.rm = TRUE),
      Q3 = quantile(Value, 0.75, na.rm = TRUE),
      .groups = 'drop'
    )

  # Plot using ggplot2
  p <- ggplot(summary_stats, aes(x = Date)) +
    geom_line(aes(y = Median, color = "Median")) +
    geom_ribbon(aes(ymin = Q1, ymax = Q3), alpha = 0.2, fill = fill_col) +
    # geom_vline(xintercept = as.Date("2022-01-01"), color = text_col, linetype = "dashed") +
    labs(title = "Median and Quartiles of Values Over Time",
         x = "Date",
         y = "Value") +
    scale_x_date(date_breaks = "2 years", date_labels = "%Y") +
    theme_custom() +
    scale_color_manual(name = "Statistic", values = c("Median" = complement)) +
    theme(legend.position = "top")
  ggsave(name, plot = p, width = width, height = height, dpi = dpi)

  display_plot(name)
}
```

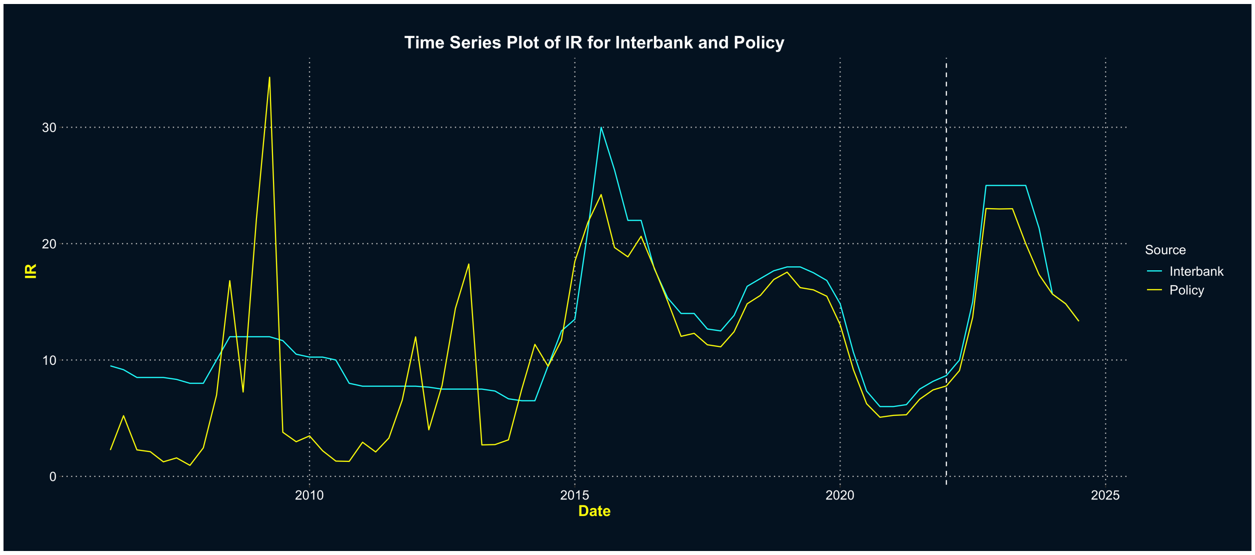
```
plot_column <- function(filename, chosen_column, name="~/Documents/GitHub/MonetaryPolicyEffectOnNetInte
  # Read the CSV file
  data <- read.csv(filename, stringsAsFactors = FALSE)

  # Convert the Date column to Date type
  data$Date <- as.Date(data$Date)

  # Check if the chosen column exists
  if (!(chosen_column %in% colnames(data))) {
    stop("Chosen column ", chosen_column, " does not exist in the CSV file.")
  }

  # Plot the chosen column
  p <- ggplot(data, aes(x = Date, y = get(chosen_column))) +
    geom_line(color = complement, linetype = "dashed") +
    geom_vline(xintercept = as.Date("2022-01-01"), color = "red", linetype = "dashed", linewidth = 0.9)
    labs(title = paste("Time Series Plot of", chosen_column),
         x = "Date",
         y = chosen_column) +
    scale_x_date(date_breaks = "2 years", date_labels = "%Y") +  # Labels and gridlines every 2 years
    theme_custom() # +
    # theme(panel.grid.major.x = element_line(colour = "red", linetype = "dashed"))  # Adding vertical

  ggsave(name, plot = p, width = width, height = height, dpi = dpi)

  display_plot(name)
}
```

Type any R code in the chunk, for example:

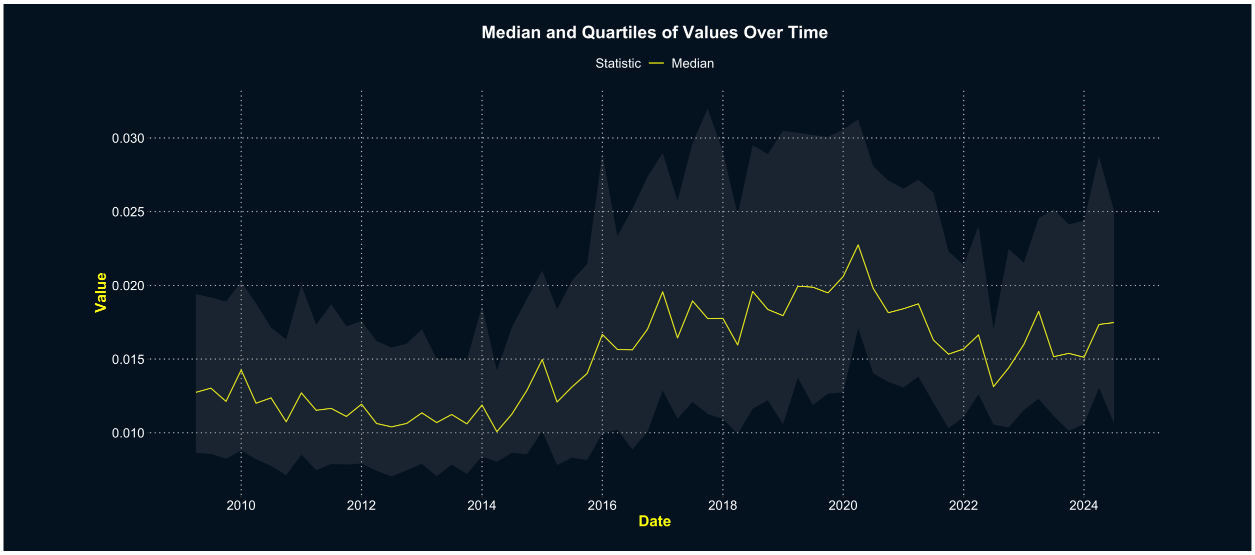
```
plot_column("~/Documents/GitHub/MonetaryPolicyEffectOnNetInterestMargins/data/relative/averaged/NII_ave
```



```r
# plot_column('~/Documents/GitHub/MonetaryPolicyEffectOnNetInterestMargins/data/relative/Interbank.csv'
#
# plot_column('~/Documents/GitHub/MonetaryPolicyEffectOnNetInterestMargins/data/relative/Policy.csv', '

# File paths
filename1 <- '~/Documents/GitHub/MonetaryPolicyEffectOnNetInterestMargins/data/relative/Interbank.csv'
filename2 <- '~/Documents/GitHub/MonetaryPolicyEffectOnNetInterestMargins/data/relative/Policy.csv'

# Column of interest
chosen_column <- 'IR'

# Read the two datasets
data1 <- read.csv(filename1, stringsAsFactors = FALSE)
data2 <- read.csv(filename2, stringsAsFactors = FALSE)

# Convert the Date column to Date type for both datasets
data1$Date <- as.Date(data1$Date)
data2$Date <- as.Date(data2$Date)

# Check if the chosen column exists in both datasets
if (!(chosen_column %in% colnames(data1)) | !(chosen_column %in% colnames(data2))) {
  stop("Chosen column ", chosen_column, " does not exist in one or both of the CSV files.")
}

# Create a new column to label the data source
data1$Source <- "Policy"
data2$Source <- "Interbank"

# Combine both datasets
combined_data <- bind_rows(
  data1 %>% select(Date, IR, Source),
  data2 %>% select(Date, IR, Source)
)

# Plot both datasets on the same plot
p <- ggplot(combined_data, aes(x = Date, y = IR, color = Source)) +
  geom_line() +
  geom_vline(xintercept = as.Date("2022-01-01"), color = text_col, linetype = "dashed") +
  labs(title = "Time Series Plot of IR for Interbank and Policy",
       x = "Date",
       y = chosen_column) +
  theme_custom() +
  scale_color_manual(values = c("cyan", complement))
name <- "~/Documents/GitHub/MonetaryPolicyEffectOnNetInterestMargins/plots/bothIR.png"
ggsave(name, plot = p, width = width, height = height, dpi = dpi)

display_plot(name)
```

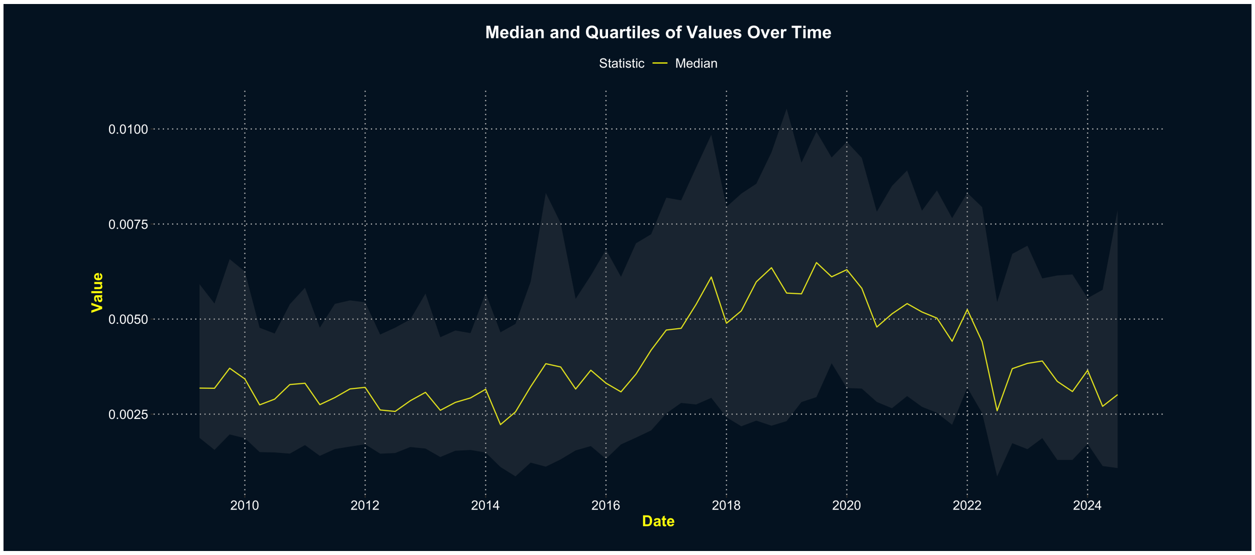


```
plot_median_quartiles("~/Documents/GitHub/MonetaryPolicyEffectOnNetInterestMargins/data/relative/admin_
```

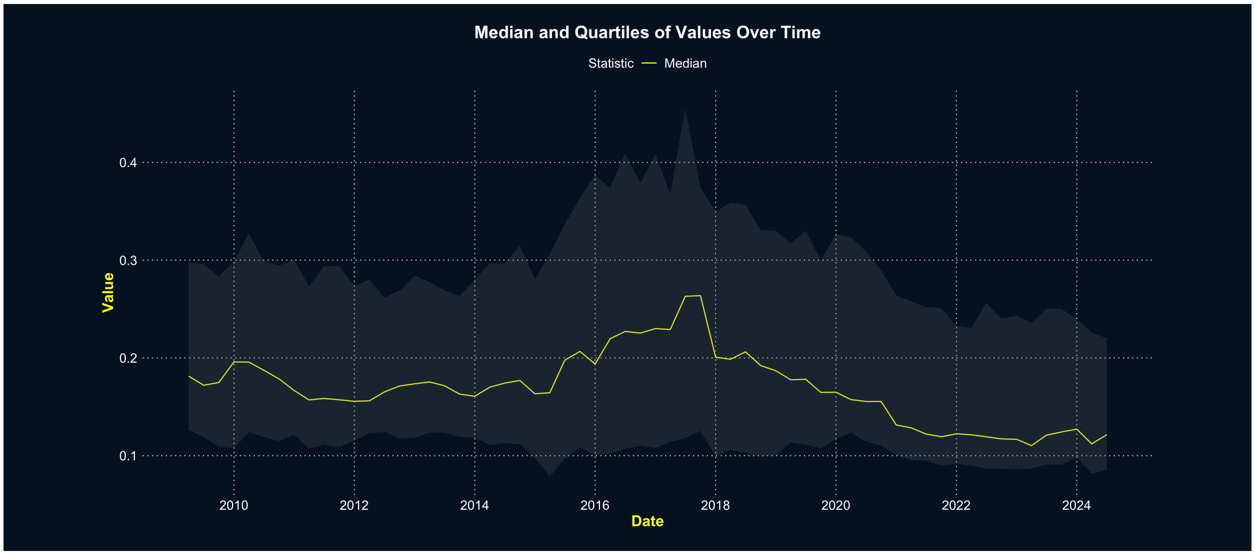


Now, click the **Run** button on the chunk toolbar to execute the chunk code. The result should be placed under the chunk. Click the **Knit and Open Document** to build and preview an output.

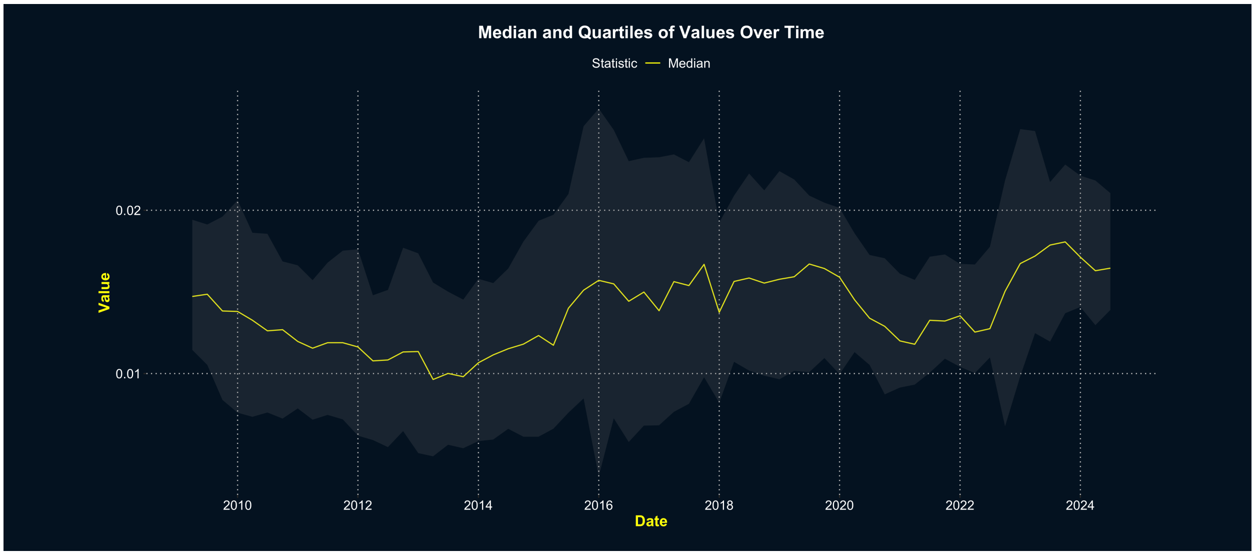
```
plot_median_quartiles("~/Documents/GitHub/MonetaryPolicyEffectOnNetInterestMargins/data/relative/net_co
```



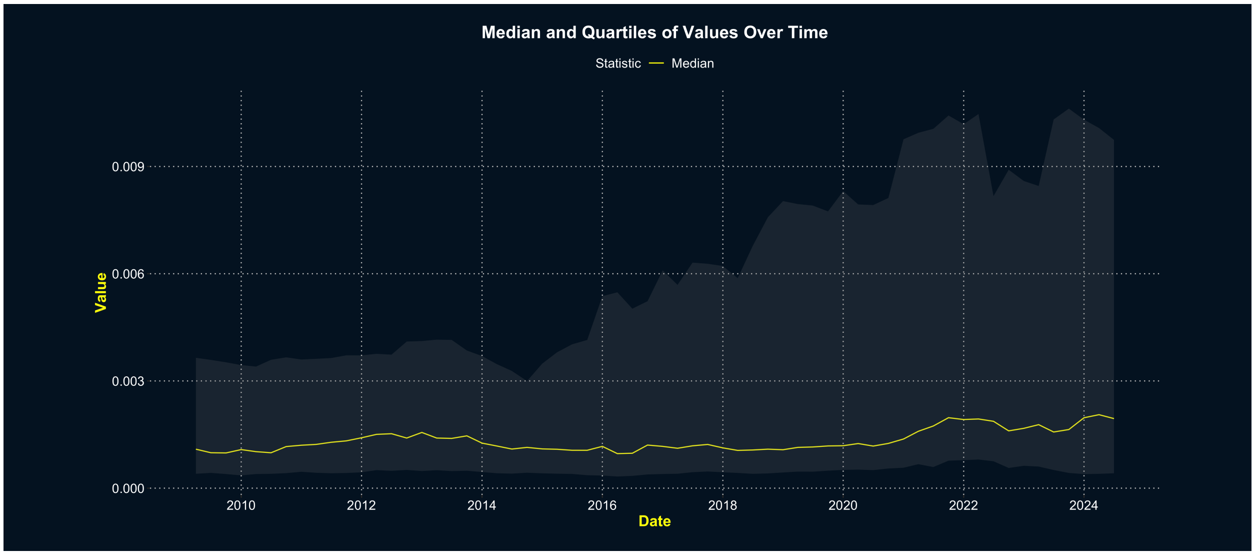
```
plot_median_quartiles("~/Documents/GitHub/MonetaryPolicyEffectOnNetInterestMargins/data/relative/capital
```



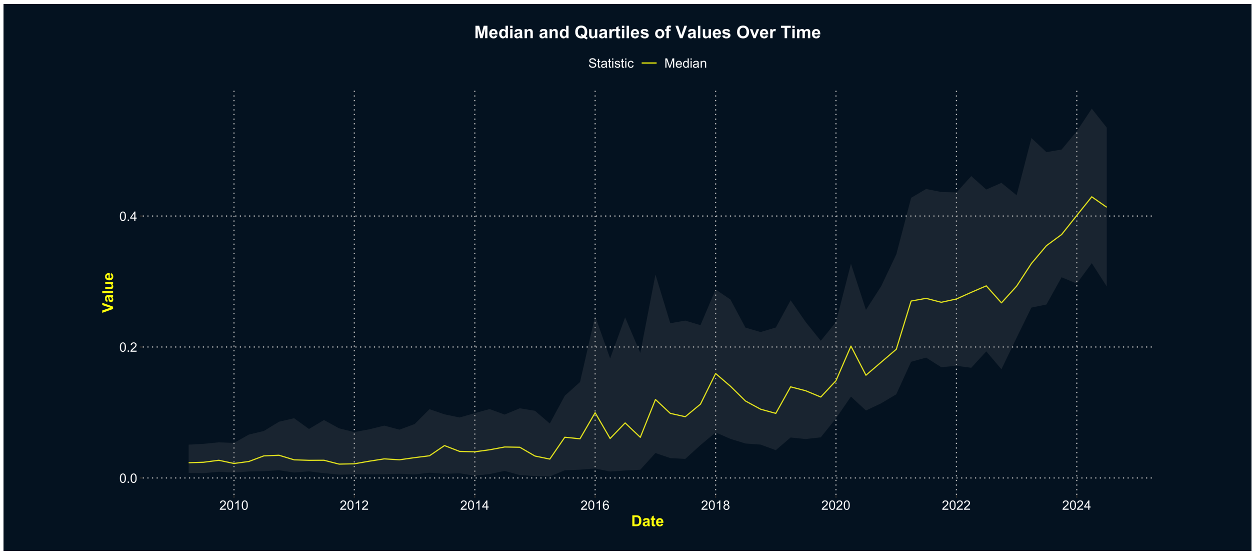
```
plot_median_quartiles("~/Documents/GitHub/MonetaryPolicyEffectOnNetInterestMargins/data/relative/net_in
```



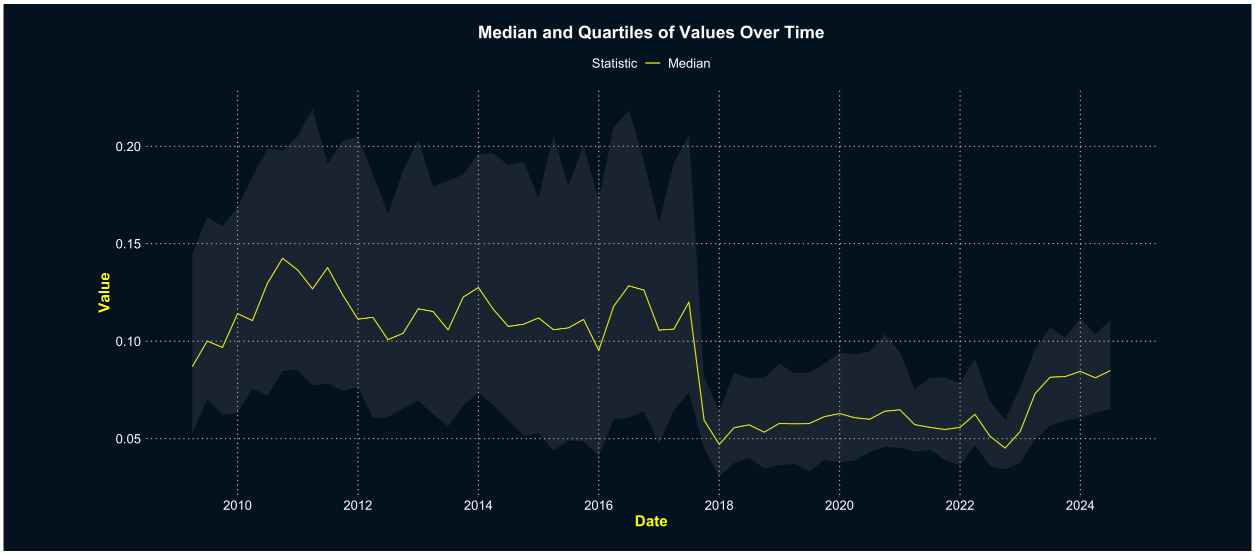
```
plot_median_quartiles("~/Documents/GitHub/MonetaryPolicyEffectOnNetInterestMargins/data/relative/total_a
```



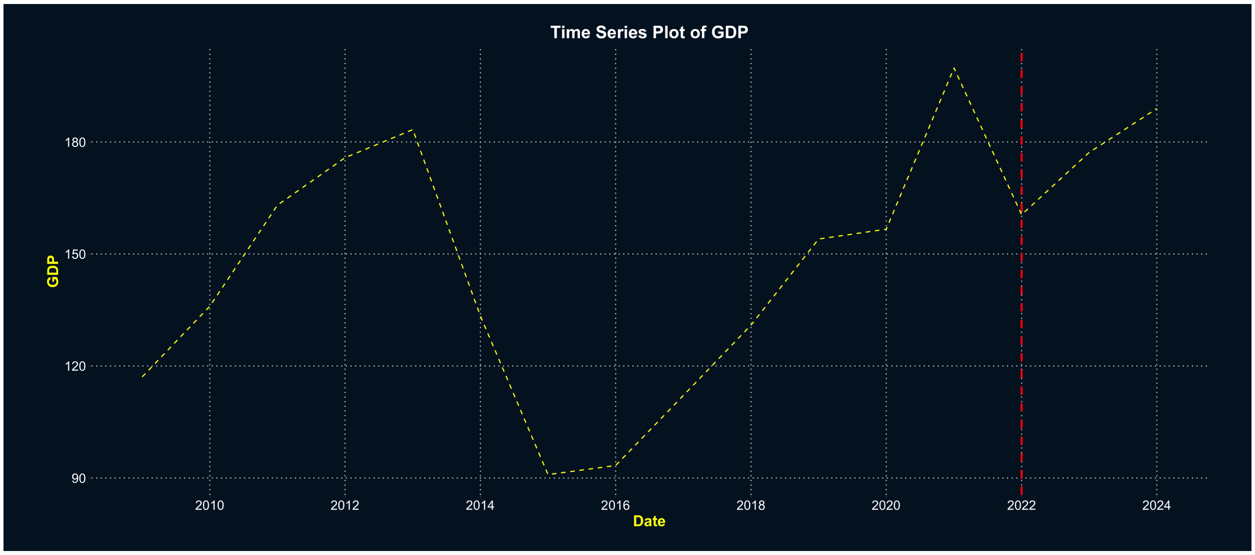
```
plot_median_quartiles("~/Documents/GitHub/MonetaryPolicyEffectOnNetInterestMargins/data/relative/securi
```



```
plot_median_quartiles("~/Documents/GitHub/MonetaryPolicyEffectOnNetInterestMargins/data/relative/cash_to
```



```
plot_column("~/Documents/GitHub/MonetaryPolicyEffectOnNetInterestMargins/data/original/statistic_id2961
```


Time Series Plot of GDP
GDP
180
150
120
90
2010
2012
2014
2016
2018
2020
2022
2024
Date