

# Analyzing Trends in Bank Assets, Liabilities, and Monetary Aggregates in Canada

Nastaran Khorram

2025-01-24

```
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

library(tidyverse)

## — Attaching core tidyverse packages ————— tidyverse
## 2.0.0 —
## ✓ forcats   1.0.0   ✓ readr      2.1.5
## ✓ ggplot2    3.5.1   ✓ stringr    1.5.1
## ✓ lubridate  1.9.4   ✓ tibble     3.2.1
## ✓ purrr      1.0.2   ✓ tidyr      1.3.1

## — Conflicts —————
tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()
## ⓘ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
## conflicts to become errors

library(ggplot2)
library(magrittr)

##
## Attaching package: 'magrittr'
##
## The following object is masked from 'package:purrr':
##
##   set_names
##
## The following object is masked from 'package:tidyr':
```

```
##
##      extract

library(readxl)
library(forecast)

## Registered S3 method overwritten by 'quantmod':
##   method      from
##   as.zoo.data.frame zoo

library(zoo)

##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##      as.Date, as.Date.numeric

data <- read_xlsx("C:/Users/98910/Desktop/Chartered bank assets and
liabilities and monetary aggregates.xlsx", sheet=1)
head(data)

## # A tibble: 6 × 3
##   Row                                Column
Value                                <dtm>
##   <chr>                                <dtm>
<dbl>
## 1 Canada Savings Bonds and other retail instruments 2024-10-01 00:00:00
349
## 2 Canada Savings Bonds and other retail instruments 2024-11-01 00:00:00
409
## 3 Canada Savings Bonds and other retail instruments 2024-07-01 00:00:00
469
## 4 Canada Savings Bonds and other retail instruments 2024-09-01 00:00:00
469
## 5 Canada Savings Bonds and other retail instruments 2024-08-01 00:00:00
470
## 6 Currency outside banks                          2024-07-01 00:00:00
116449

colnames(data)

## [1] "Row"      "Column"   "Value"

summary(data)

##      Row                                Column                                Value
## Length:100                      Min.      :2024-07-01 00:00:00      Min.      : 349
## Class :character                1st Qu.:2024-08-01 00:00:00      1st Qu.: 707331
## Mode  :character                Median :2024-09-01 00:00:00      Median :1486836
##                                     Mean   :2024-08-31 14:24:00      Mean    :1686512
```

```
##          3rd Qu.:2024-10-01 00:00:00    3rd Qu.:2726503
##          Max.      :2024-11-01 00:00:00    Max.      :4827363
```

```
data <- data %>%
  rename(
    Category = `Row`,
    Date = `Column`,
    Value = `Value`
  ) %>%
  mutate(
    Date = as.Date(Date, format = "%Y-%m-%d"),
    Value = as.numeric(Value)
  )
data <- data %>%
  mutate(Short_Category = substr(Category, 1, 30)) # Use first 30 characters

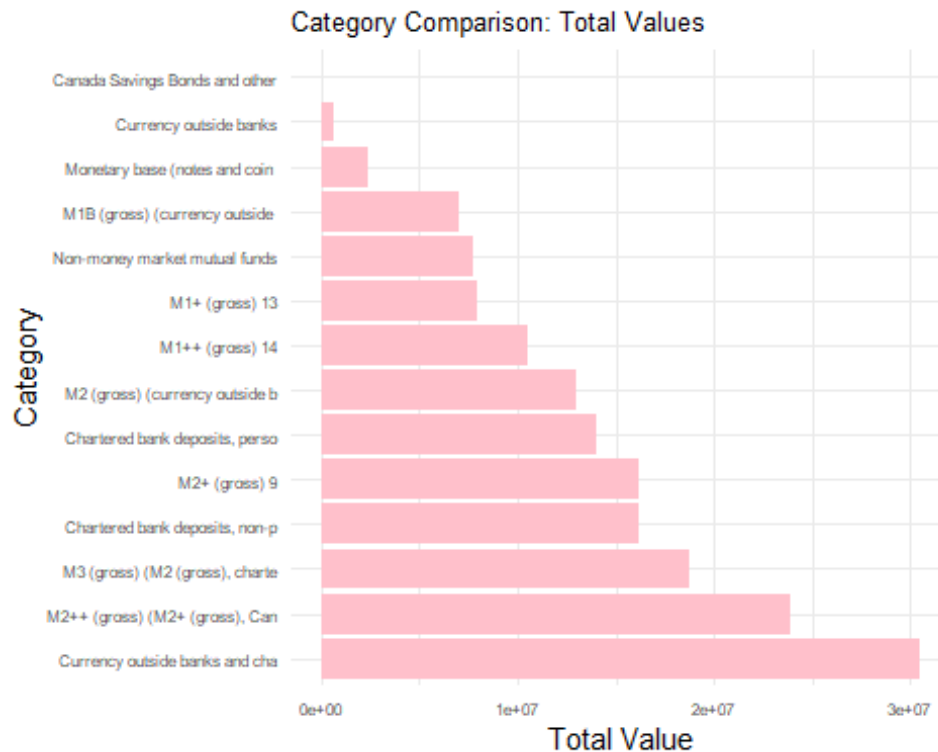
category_totals <- data %>%
  group_by(Short_Category) %>%
  summarize(Total_Value = sum(Value, na.rm = TRUE)) %>%
  arrange(desc(Total_Value))
```

```
print(category_totals)
```

```
## # A tibble: 14 × 2
##   Short_Category      Total_Value
##   <chr>              <dbl>
## 1 "Currency outside banks and cha" 30511374
## 2 "M2++ (gross) (M2+ (gross), Can" 23896879
## 3 "M3 (gross) (M2 (gross), charte" 18716522
## 4 "Chartered bank deposits, non-p" 16191243
## 5 "M2+ (gross) 9"                16124836
## 6 "Chartered bank deposits, perso" 13981836
## 7 "M2 (gross) (currency outside b" 12992769
## 8 "M1++ (gross) 14"              10515868
## 9 "M1+ (gross) 13"               7912973
## 10 "Non-money market mutual funds" 7769877
## 11 "M1B (gross) (currency outside " 7016680
## 12 "Monetary base (notes and coin " 2432230
## 13 "Currency outside banks"        585902
## 14 "Canada Savings Bonds and other" 2166
```

```
ggplot(category_totals, aes(x = reorder(Short_Category, -Total_Value), y =
Total_Value)) +
  geom_bar(stat = "identity", fill = "pink") +
  coord_flip() +
  labs(
    title = "Category Comparison: Total Values",
    x = "Category",
    y = "Total Value"
  ) +
```

```
theme_minimal() +
theme(
  axis.text.y = element_text(size = 6),
  axis.text.x = element_text(size = 6),
  plot.title = element_text(size = 10),
  legend.position = "none"
)
```



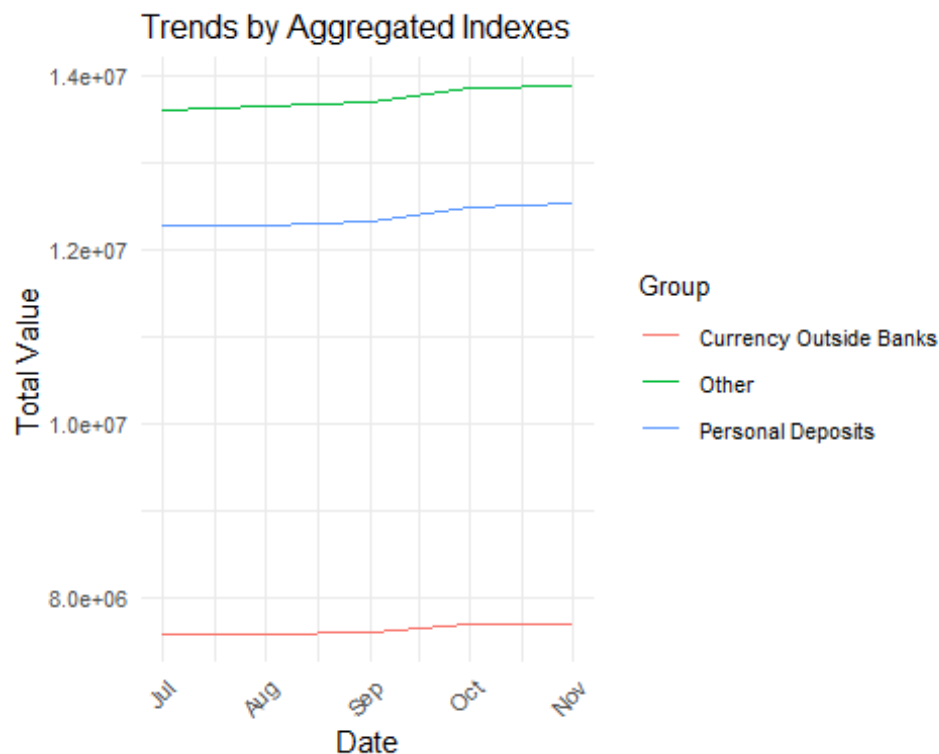
```
aggregated_data <- data %>%
  mutate(
    Group = case_when(
      grepl("personal", Category, ignore.case = TRUE) ~ "Personal Deposits",
      grepl("non-personal", Category, ignore.case = TRUE) ~ "Non-Personal
Deposits",
      grepl("Currency outside banks", Category, ignore.case = TRUE) ~
"Currency Outside Banks",
      TRUE ~ "Other"
    )
  ) %>%
  group_by(Group, Date) %>%
  summarize(Total_Value = sum(Value, na.rm = TRUE))

## `summarise()` has grouped output by 'Group'. You can override using the
## `.groups` argument.

head(aggregated_data)
```

```
## # A tibble: 6 × 3
## # Groups:   Group [2]
##   Group          Date      Total_Value
##   <chr>         <date>         <dbl>
## 1 Currency Outside Banks 2024-07-01      7565092
## 2 Currency Outside Banks 2024-08-01      7572345
## 3 Currency Outside Banks 2024-09-01      7588357
## 4 Currency Outside Banks 2024-10-01      7691598
## 5 Currency Outside Banks 2024-11-01      7696564
## 6 Other                2024-07-01     13591290

ggplot(aggregated_data, aes(x = Date, y = Total_Value, color = Group)) +
  geom_line() +
  labs(
    title = "Trends by Aggregated Indexes",
    x = "Date",
    y = "Total Value",
    color = "Group"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(size = 8, angle = 45, hjust = 1),
    axis.text.y = element_text(size = 8),
    legend.text = element_text(size = 8),
    legend.title = element_text(size = 10),
    plot.title = element_text(size = 12)
  )
)
```

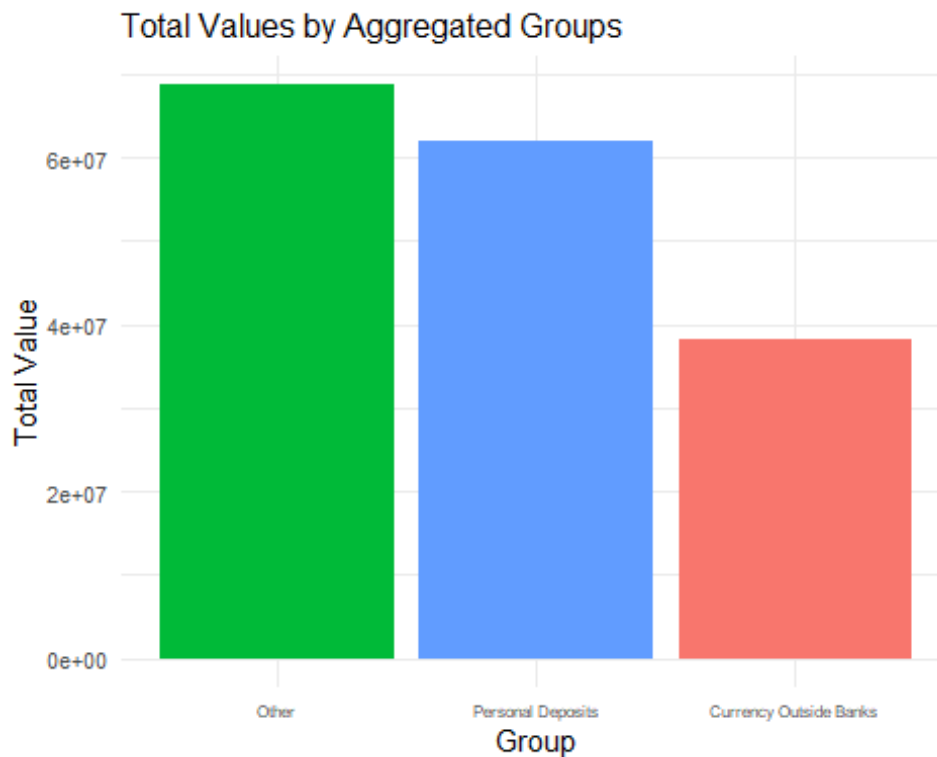


```

group_totals <- aggregated_data %>%
  group_by(Group) %>%
  summarize(Total_Value = sum(Total_Value))

ggplot(group_totals, aes(x = reorder(Group, -Total_Value), y = Total_Value,
fill = Group)) +
  geom_bar(stat = "identity") +
  labs(
    title = "Total Values by Aggregated Groups",
    x = "Group",
    y = "Total Value"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(size = 6),
    axis.text.y = element_text(size = 8),
    plot.title = element_text(size = 12),
    legend.position = "none" # Hide Legend (optional)
  )

```



```

summary_stats <- aggregated_data %>%
  group_by(Group) %>%
  summarize(
    Mean_Value = mean(Total_Value, na.rm = TRUE),
    Median_Value = median(Total_Value, na.rm = TRUE),
    SD_Value = sd(Total_Value, na.rm = TRUE),

```

```

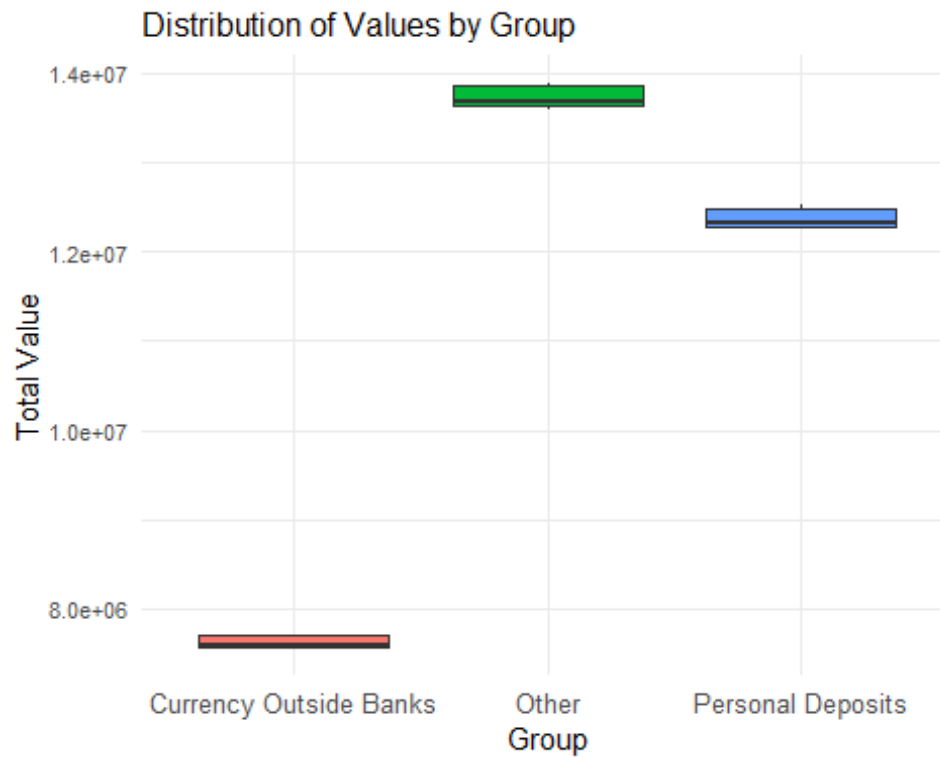
    Min_Value = min(Total_Value, na.rm = TRUE),
    Max_Value = max(Total_Value, na.rm = TRUE)
  )

print(summary_stats)

## # A tibble: 3 × 6
##   Group          Mean_Value Median_Value SD_Value Min_Value
##   <chr>          <dbl>         <dbl>    <dbl>    <dbl>
## 1 Currency Outside Banks  7622791.      7588357    65644.    7565092
##   7696564
## 2 Other          13730966.    13685271   130833.   13591290
##   13883097
## 3 Personal Deposits    12376474    12320066   121203.   12273411
##   12523467

ggplot(aggregated_data, aes(x = Group, y = Total_Value, fill = Group)) +
  geom_boxplot() +
  labs(
    title = "Distribution of Values by Group",
    x = "Group",
    y = "Total Value"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(size = 10),
    axis.text.y = element_text(size = 8),
    plot.title = element_text(size = 12),
    legend.position = "none"
  )

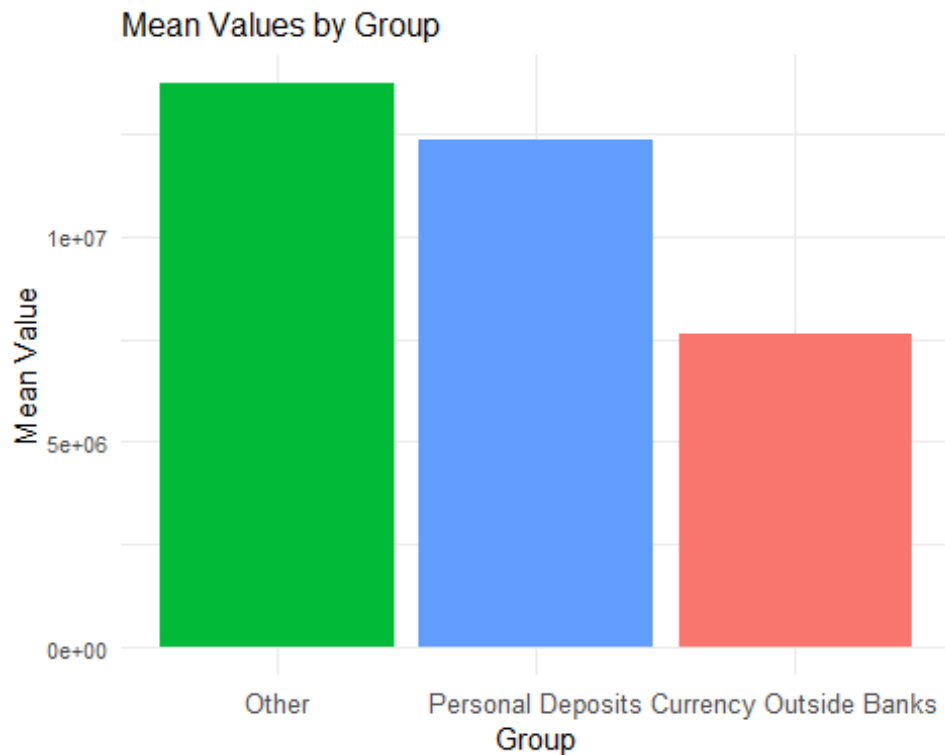
```



```
mean_values <- summary_stats %>%
  select(Group, Mean_Value)

ggplot(mean_values, aes(x = reorder(Group, -Mean_Value), y = Mean_Value, fill
= Group)) +
  geom_bar(stat = "identity") +
  labs(
    title = "Mean Values by Group",
    x = "Group",
    y = "Mean Value"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(size = 10),
    axis.text.y = element_text(size = 8),
    plot.title = element_text(size = 12),
    legend.position = "none"
  )
```





### ##ARIMA Forecast

```
personal_deposits <- aggregated_data %>%
  filter(Group == "Personal Deposits") %>%
  arrange(Date)

all_dates <- seq(min(personal_deposits$Date), max(personal_deposits$Date), by
= "month")
missing_dates <- setdiff(all_dates, personal_deposits$Date)

if (length(missing_dates) > 0) {
  cat("Missing dates detected:\n")
  print(missing_dates)
} else {
  cat("No missing dates.\n")
}

## No missing dates.

personal_deposits <- personal_deposits %>%
  complete(Date = seq(min(Date), max(Date), by = "month")) %>%
  arrange(Date)

personal_deposits$Total_Value <-
zoo::na.approx(personal_deposits$Total_Value, na.rm = FALSE)
```

```

personal_ts <- ts(personal_deposits$Total_Value, frequency = 12, start =
c(2024, 7))

personal_arma <- auto.arima(personal_ts)
summary(personal_arma)

## Series: personal_ts
## ARIMA(0,1,0)
##
## sigma^2 = 8.089e+09: log likelihood = -51.29
## AIC=104.59 AICc=106.59 BIC=103.97
##
## Training set error measures:
##
##           ME      RMSE      MAE      MPE      MAPE  MASE
ACF1
## Training set 52465.88 80443.24 52465.88 0.4212955 0.4212955 NaN -
0.04011962

forecast_personal <- forecast(personal_arma, h = 12)

forecast_df <- data.frame(
  Date = seq(as.Date("2024-08-01"), by = "month", length.out = 12),
  Forecast = as.numeric(forecast_personal$mean),
  Lower = as.numeric(forecast_personal$lower[, 2]),
  Upper = as.numeric(forecast_personal$upper[, 2])
)

ggplot(forecast_df, aes(x = Date, y = Forecast)) +
  geom_line(color = "maroon") +
  geom_ribbon(aes(ymin = Lower, ymax = Upper), alpha = 0.2, fill = "green") +
  labs(
    title = "ARIMA Forecast for Personal Deposits",
    x = "Date",
    y = "Total Value"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.text.y = element_text(size = 8),
    plot.title = element_text(size = 12)
  )

```

ARIMA Forecast for Personal Deposits

