# Flow of Funds - Final Report

September 7, 2024

University of California Los Angeles
Master of Quantitative Economics -MQE-
ECON 451 - Financial Institutions and Monetary Policy

Nikolaos Papadatos

Flow of Funds

In this project, we focus on analyzing the U.S. commercial banking sector (L.111 U.S.-Chartered Depository Institutions) as documented in the Flow of Funds Levels. This sector plays a pivotal role in the economy, making it a compelling subject for analysis. Commercial banks serve as essential financial intermediaries, channeling funds between depositors and borrowers, thereby fueling investment, consumption, and business growth. Through the practice of fractional reserve banking, they possess the unique ability to expand the money supply, directly influencing economic activity and growth.

Moreover, their central role in the financial markets gives commercial banks substantial influence over market conditions and stability. Fluctuations in lending practices and interest rates can have profound effects on asset prices, liquidity, and investor sentiment. Banks also act as the primary channels for implementing monetary policy, adjusting interest rates and credit availability to steer economic activity. With a strong presence in the credit and mortgage markets, commercial banks significantly impact consumer spending, housing dynamics, and business investments. This centrality makes them critical players in both economic growth and stability.

[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter
```

### Importing datasets

- Assets
- Liabilities
- Equity

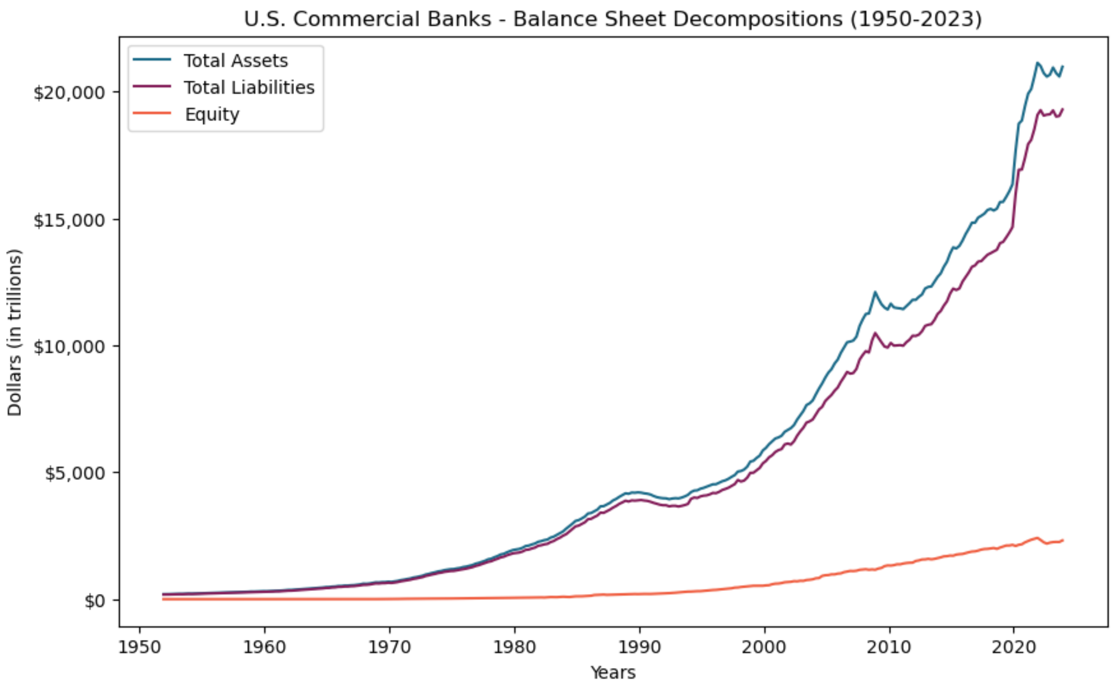
[ ]:
```
# Importing the datasets
df = pd.read_excel('L.111 Banks_assets.xlsx', sheet_name='assets_clean',␣
 ↪index_col=0, parse_dates=True)
```

```
assets_terms = pd.read_excel('L.111 Banks_assets.xlsx', sheet_name='Term',␣
 ↪index_col=0)


# Assets dataframe without 'total financial assets'
assets = df.drop(['total financial assets'], axis=1)
assets = assets/1000
```

```
[ ]: df2 = pd.read_excel('L.111 Banks_liabilities.xlsx', sheet_name='Clean␣
      ↪Liabilities', index_col=0, parse_dates=True)
     liabilities_terms = pd.read_excel('L.111 Banks_liabilities.xlsx',␣
      ↪sheet_name='Term', index_col=0)

     liab = df2 / 1000
```

```
[ ]: # Obtain the total assets, total liabilities and equity capital
     Total_Assets = assets.sum(axis=1)
     Total_Liabilities = liab.sum(axis=1) - liab[' capital market equity and␣
      ↪investment fund shares ']
     Equity_Capital = liab[' capital market equity and investment fund shares ']
     Time = Total_Assets.index
```

```
[ ]: # Set different color palettes for each trend
     palette1 = sns.color_palette("crest_r", 2)  # Trend 1 palette
     palette2 = sns.color_palette("rocket", 2)  # Trend 2 palette
     palette3 = sns.color_palette("rocket_r", 2)  # Trend 3 palette

     # Plot the trends
     plt.figure(figsize=(10, 6))

     # Plot each trend separately and manually set the line color
     sns.lineplot(x=Time, y=Total_Assets, label='Total Assets', color=palette1[0])
     sns.lineplot(x=Time, y=Total_Liabilities, label='Total Liabilities',␣
      ↪color=palette2[0])
     sns.lineplot(x=Time, y=Equity_Capital, label='Equity', color=palette3[0])

     # Define formatter function
     formatter = FuncFormatter(lambda x, _: '${:,.0f}'.format(x))
     plt.gca().yaxis.set_major_formatter(formatter)

     plt.title('U.S. Commercial Banks - Balance Sheet Decomposition (1950-2023)')
     plt.xlabel('Years')
     plt.ylabel('Dollars (in trillions)')
     plt.legend()
     plt.show()
```



### 0.0.1 Assets

Figure X illustrates the historical composition of assets held by commercial banks in the U.S. The primary component among these assets is depository institution loans, which represent [explain briefly what depository institution loans are]. Agency-backed securities also constitute a significant portion of the assets.

Notably, there has been a notable uptick in the account of depository institution reserves since 2020. This increase in liquidity may be attributed to the implementation of economic COVID programs aimed at stabilizing the economy.

A pertinent observation is that the assets of commercial banks typically decrease following major economic recessions (80's, 2008 and 2020). As expected, the majority of these assets are long-term, comprising 81.5% of the total financial assets. This trend aligns with the evolution of maturity transformation within the banking industry.

```
[ ]: # Sort variables based on their last observation (highest to lowest value)
     df_sorted = assets.iloc[:, assets.iloc[-1].argsort()[::-1]]

     # Define labels with percentages
     #labels = [f"{col}\n({(df_sorted[col].iloc[-1] / df_sorted.iloc[-1].sum() *␣
      ↪100):.1f}%)" for col in df_sorted.columns]
     labels = [f"{col} ({(df_sorted[col].iloc[-1] / df_sorted.iloc[-1].sum() * 100):.
      ↪1f}%)" for col in df_sorted.columns]

     # Get the color palette
```

```
palette = sns.color_palette("crest_r", len(labels))  # Rocket is a good␣
↪alternative for liability accounts

# Plot stackplot
fig, ax = plt.subplots(figsize=(12, 8))
ax.stackplot(df_sorted.index, df_sorted.values.T, labels=labels, colors=palette)

# Add legend
ax.legend(loc='upper left', shadow=False, ncol=1, title = 'Assets (% of total␣
↪assets, Q4 2023)')

# Add x and y labels
ax.set_xlabel("Year")
ax.set_ylabel("Trillions USD")

# Rotate x-axis labels
plt.xticks(rotation=40)

# Format y-axis ticks as currency
formatter = FuncFormatter(lambda x, _: '${:,.0f}'.format(x))
ax.yaxis.set_major_formatter(formatter)

# Remove the frame
#ax.set_frame_on(False)
plt.title("Figure X: Assets of U.S. Banks (1950-2023)", fontsize=16)
plt.show()
```

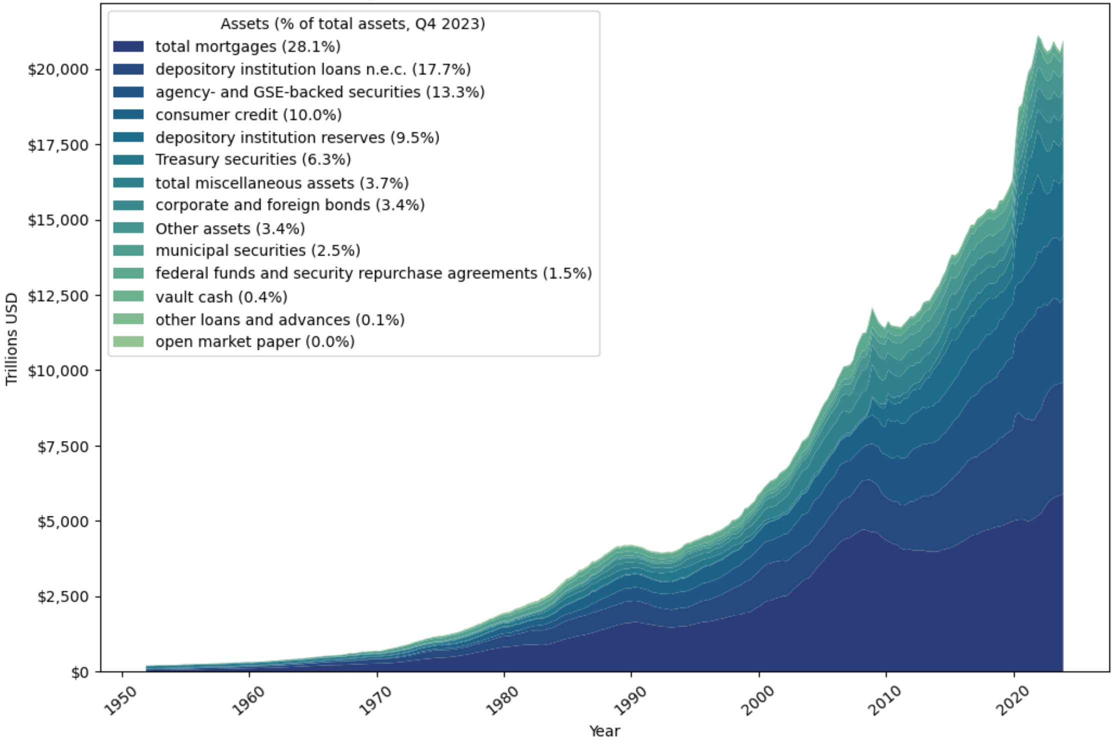


Figure X: Assets of U.S. Banks (1950-2023)

```
[ ]: # Order values in assets_terms, biggest to smallest
     assets_terms = assets_terms.sort_values(by= 'Value', ascending=False)
     assets_terms['Value'] = assets_terms.Value/1000

     # Proportion of each asset term
     assets_terms['Proportion'] = assets_terms['Value']/assets_terms['Value'].sum()
     assets_terms
```

```
[ ]:                                          Account     Value        Term  \
     23                                 total mortgages  5895.092   Long-term
     21             depository institution loans n.e.c.  3713.624   Long-term
     7                agency- and GSE-backed securities  2782.457   Long-term
     24                                 consumer credit  2100.672   Long-term
     2                  depository institution reserves  1987.050  Short-term
     6                             Treasury securities  1329.206   Long-term
     32                     total miscellaneous assets   777.390  Short-term
     14                     corporate and foreign bonds   718.202   Long-term
     25                                    Other assets   706.168  Short-term
     13                            municipal securities   531.851   Long-term
     3   federal funds and security repurchase agreements   321.586  Short-term
     1                                       vault cash    83.802  Short-term
     22                         other loans and advances    19.957   Long-term
```

```
5                                   open market paper    0.000   Long-term

    Proportion
23    0.281160
21    0.177117
7     0.132706
24    0.100189
2     0.094770
6     0.063395
32    0.037077
14    0.034254
25    0.033680
13    0.025366
3     0.015338
1     0.003997
22    0.000952
5     0.000000
```

[ ]:

```python
# Group by terms, sum the Proportion
assets_terms_grouped = assets_terms.groupby('Term').sum()

# Estimate the proportion of the terms
assets_terms_grouped['Proportion'] = assets_terms_grouped['Value']/
 ↪assets_terms_grouped['Value'].sum()

# Bar plot of terms in assets
fig, ax = plt.subplots(figsize=(12, 8))
palette2 = sns.color_palette("crest_r", 2)
ax.bar(assets_terms_grouped.index, assets_terms_grouped['Value'],␣
 ↪color=palette2)
ax.set_xlabel("Term")
ax.set_ylabel("Trillions USD")
formatter = FuncFormatter(lambda x, _: '${:,.0f}'.format(x))
ax.yaxis.set_major_formatter(formatter)

# Add text labels
for i, v in enumerate(assets_terms_grouped['Value']):
    ax.text(i, v + 0.5, f"{v:.1f} ({assets_terms_grouped['Proportion'].
 ↪iloc[i]*100:.1f}%)", color='black', ha='center')
plt.title("Figure X: Assets of U.S. Banks by Term (Q4 2023)", fontsize=16)
plt.show()
```

```
C:\Users\nikpa\AppData\Local\Temp\ipykernel_30940\568526362.py:2: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a
future version, numeric_only will default to False. Either specify numeric_only
or select only columns which should be valid for the function.
  assets_terms_grouped = assets_terms.groupby('Term').sum()
```

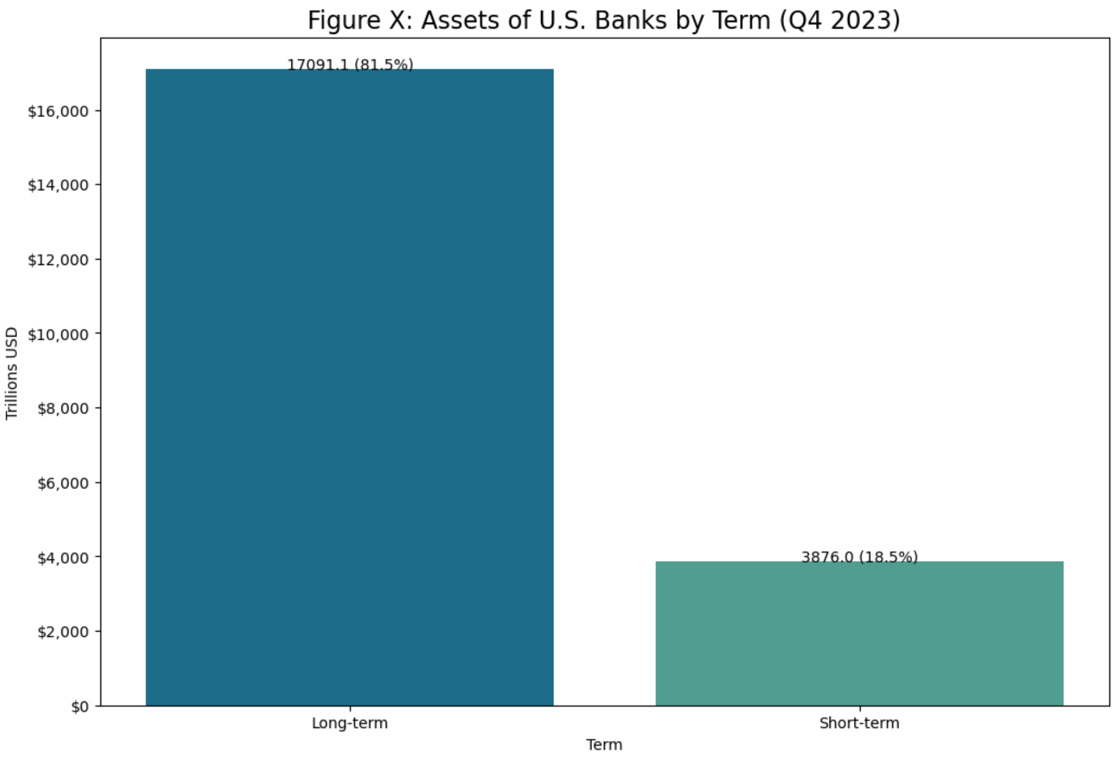


Figure X: Assets of U.S. Banks by Term (Q4 2023)

### 0.0.2 Liabilities

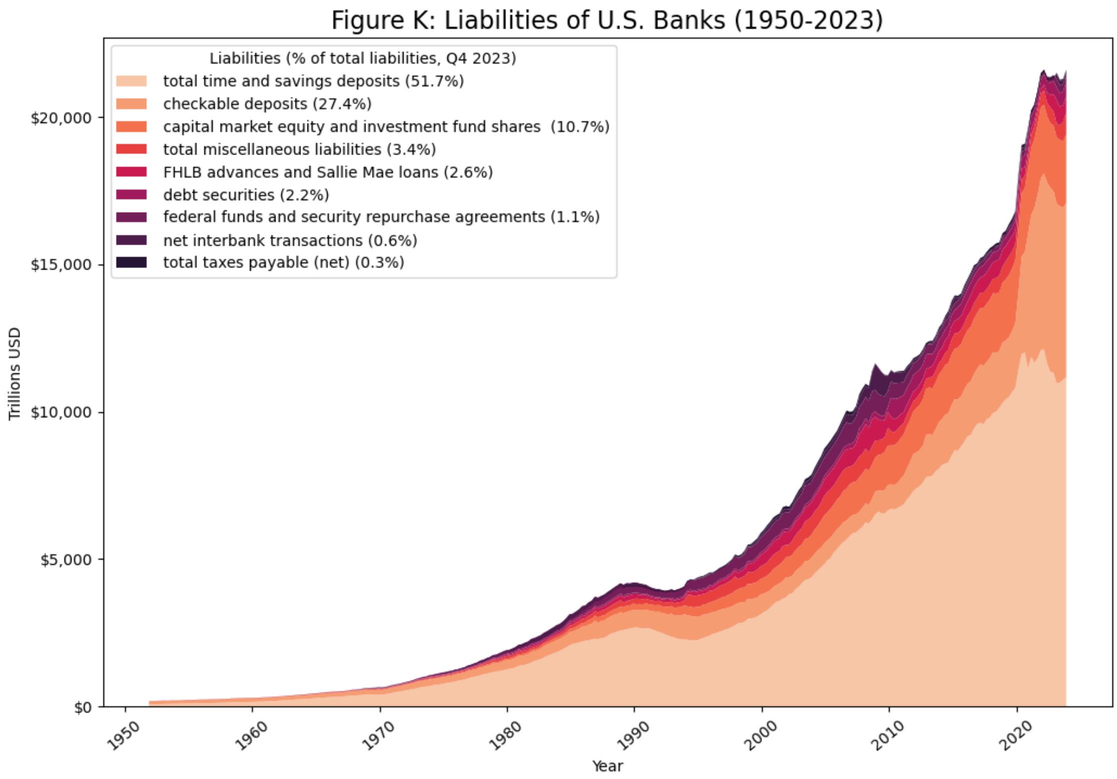
### 0.0.3 Liabilities Composition Analysis (Q4 2023)

Figure K offers insights into the historical composition of liabilities among commercial banks in the U.S. Notably, time and savings deposits emerge as the primary component, constituting approximately 52% of the total liabilities. Additionally, checkable deposits contribute significantly, collectively representing around 80% of the total figure. This dominance of deposit-based liabilities underscores the fundamental role of customer deposits in bank financing.

Over the decades since the 1950s, there has been a consistent upward trajectory in bank liabilities, with a remarkable aspect being the predominance of long-term liabilities. Despite the conventional expectation of a higher proportion of short-term liabilities in banking operations, we see that long-term liabilities hold substantial significance, representing approximately 69% of the total liabilities. This enduring trend highlights banks' adeptness in attracting stable funding sources over extended periods, facilitating long-term lending and investment activities.

[ ]:
```
# Sort variables based on their last observation (highest to lowest value)
df_sorted2 = liab.iloc[:, liab.iloc[-1].argsort()[::-1]]

# Define labels with percentages
#labels = [f"{col}\n({(df_sorted[col].iloc[-1] / df_sorted.iloc[-1].sum() *␣
 ↪100):.1f}%)" for col in df_sorted.columns]
```

```python
labels2 = [f"{col} ({(df_sorted2[col].iloc[-1] / df_sorted2.iloc[-1].sum() * 100):.1f}%)" for col in df_sorted2.columns]

# Get the color palette
palette = sns.color_palette("rocket_r", len(labels2))  # Rocket is a good alternative for liability accounts

# Plot stackplot
fig, ax = plt.subplots(figsize=(12, 8))
ax.stackplot(df_sorted2.index, df_sorted2.values.T, labels=labels2, colors=palette)

# Add legend
ax.legend(loc='upper left', shadow=False, ncol=1, title = 'Liabilities (% of total liabilities, Q4 2023)')

# Add x and y labels
ax.set_xlabel("Year")
ax.set_ylabel("Trillions USD")

# Rotate x-axis labels
plt.xticks(rotation=40)

# Format y-axis ticks as currency
formatter = FuncFormatter(lambda x, _: '${:,.0f}'.format(x))
ax.yaxis.set_major_formatter(formatter)

# Remove the frame
#ax.set_frame_on(False)
plt.title("Figure K: Liabilities of U.S. Banks (1950-2023)", fontsize=16)
plt.show()
```



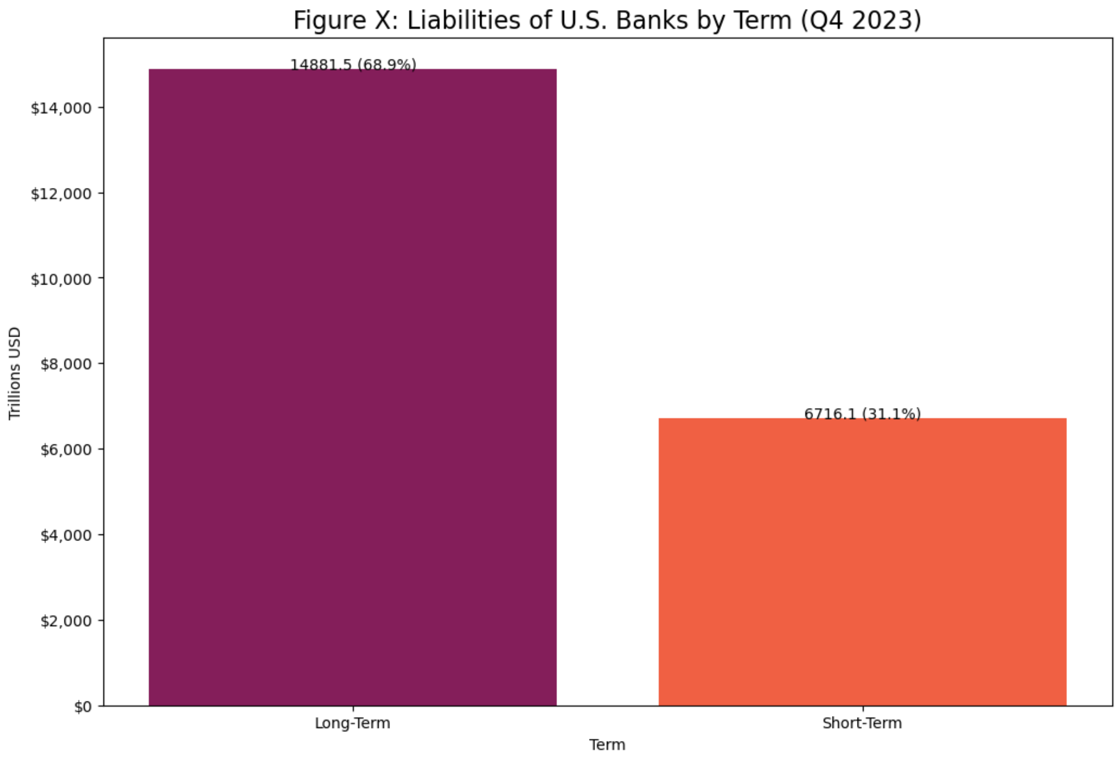
```
[ ]: # Order values in assets_terms, biggest to smallest
     liabilities_terms = liabilities_terms.sort_values(by= 'Value', ascending=False)
     liabilities_terms['Value'] = liabilities_terms.Value/1000

     # Proportion of each asset term
     liabilities_terms['Proportion'] = liabilities_terms['Value']/
      ↪assets_terms['Value'].sum()
     liabilities_terms
```

```
[ ]:                                                        Value        Term  \
     Account
      total time and savings deposits                     11173.773   Long-Term
      checkable deposits                                   5918.590  Short-Term
      capital market equity and investment fund shares     2314.001   Long-Term
      total miscellaneous liabilities                       738.259   Long-Term
      FHLB advances and Sallie Mae loans                    552.618  Short-Term
      debt securities                                       471.213   Long-Term
      federal funds and security repurchase agreements      244.922  Short-Term
      net interbank transactions                            127.528   Long-Term
      total taxes payable (net)                              56.747   Long-Term

                                                        Proportion
```

```
Account
 total time and savings deposits                     0.532920
 checkable deposits                                  0.282280
 capital market equity and investment fund shares    0.110364
 total miscellaneous liabilities                     0.035210
 FHLB advances and Sallie Mae loans                  0.026356
 debt securities                                     0.022474
 federal funds and security repurchase agreements    0.011681
 net interbank transactions                          0.006082
 total taxes payable (net)                           0.002706
```

[ ]:

```
# liabilities_terms# Group by terms, sum the Proportion
liabilities_terms_grouped = liabilities_terms.groupby('Term').sum()

# Estimate the proportion of the terms
liabilities_terms_grouped['Proportion'] = liabilities_terms_grouped['Value']/
 ↪liabilities_terms_grouped['Value'].sum()

# Bar plot of terms in assets
fig, ax = plt.subplots(figsize=(12, 8))
palette2 = sns.color_palette("rocket", 2)
ax.bar(liabilities_terms_grouped.index, liabilities_terms_grouped['Value'],␣
 ↪color=palette2)
ax.set_xlabel("Term")
ax.set_ylabel("Trillions USD")
formatter = FuncFormatter(lambda x, _: '${:,.0f}'.format(x))
ax.yaxis.set_major_formatter(formatter)

# Add text labels
for i, v in enumerate(liabilities_terms_grouped['Value']):
    ax.text(i, v + 0.5, f"{v:.1f} ({liabilities_terms_grouped['Proportion'].
 ↪iloc[i]*100:.1f}%)", color='black', ha='center')
plt.title("Figure X: Liabilities of U.S. Banks by Term (Q4 2023)", fontsize=16)
plt.show()
```

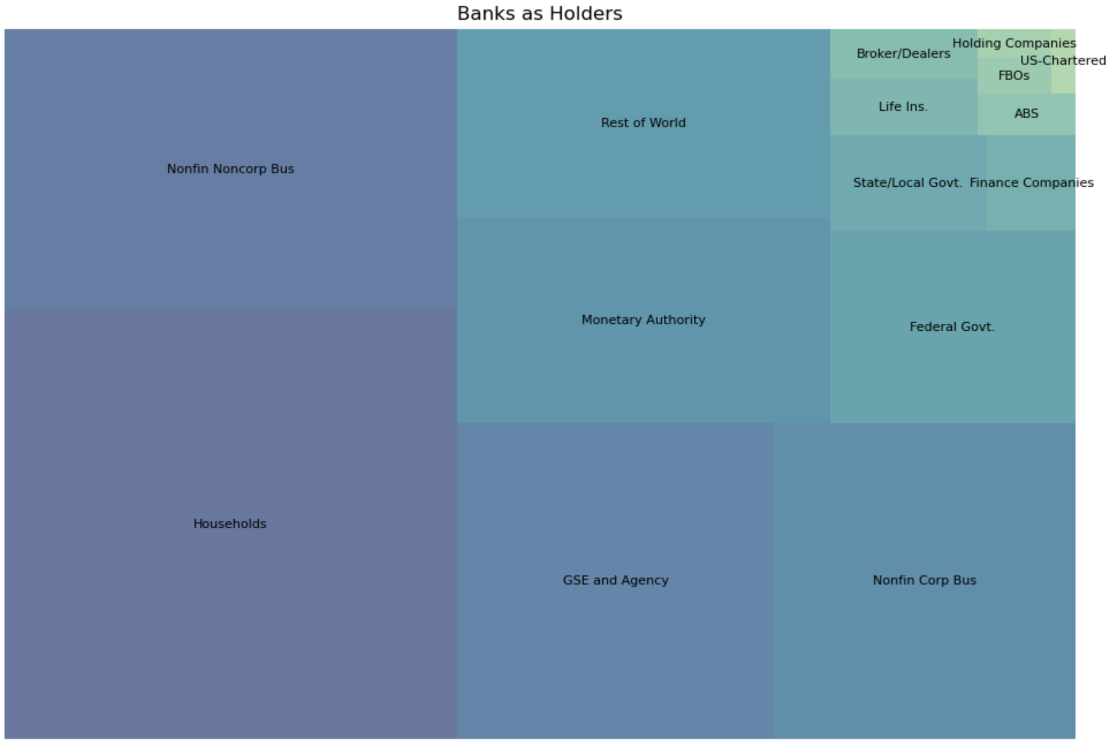


```
[ ]: holder = pd.read_excel('Issuer-to-Holder.xlsx', sheet_name='Banks as Holders')
     holder = holder.head(15)
     holder
```

```
[ ]:              Row Labels  Sum of Level  Proportions
     0            Households     5472483.2     0.254461
     1   Nonfin Noncorp Bus     3565507.4     0.165790
     2        GSE and Agency     2819574.9     0.131105
     3      Nonfin Corp Bus     2675612.5     0.124411
     4   Monetary Authority     2158609.4     0.100372
     5         Rest of World     1978394.2     0.091992
     6         Federal Govt.     1333900.1     0.062024
     7     State/Local Govt.      423665.1     0.019700
     8     Finance Companies      236954.9     0.011018
     9            Life Ins.      234791.6     0.010917
     10       Broker/Dealers      207244.5     0.009637
     11                  ABS      115823.0     0.005386
     12                 FBOs       71833.3     0.003340
     13   Holding Companies       60102.9     0.002795
     14          US-Chartered       44147.2     0.002053
```

[ ]:

```
# Plot treemap
plt.figure(figsize=(12, 8))

# Define a palette of colors
custom_palette = sns.color_palette("crest_r", n_colors=len(holder))

# Plot treemap using squarify
squarify.plot(sizes=holder['Sum of Level'], label=holder['Row Labels'],␣
 ↪color=custom_palette, alpha=0.7, text_kwargs={'fontsize': 8})

# Remove axis
plt.axis('off')
plt.title('Banks as Holders')
plt.show()
```



In the case of Banks as the Holder of the Financial Instruments, acting as creditor, the main counterparties are households (25%), non-financial companies (17%), GSE (13%) and Non-Corporate Businesses (12%). We also observe a notable amount of 10% with the Monetary authority.

[ ]:

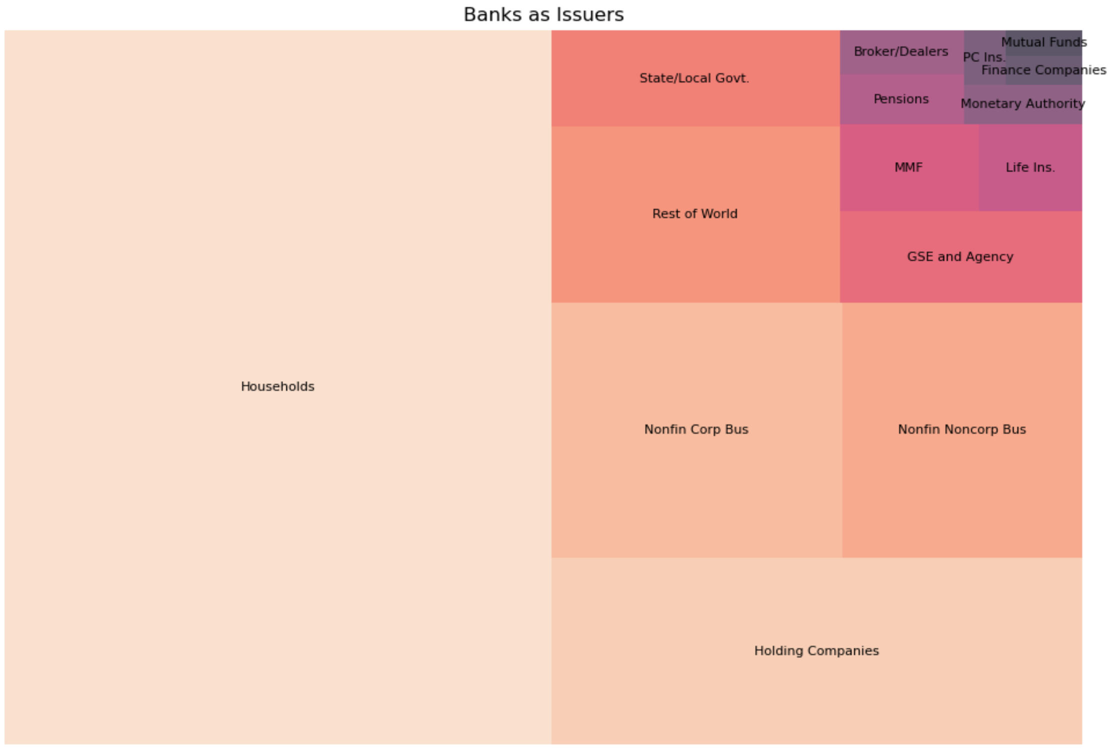
```
issuer = pd.read_excel('Issuer-to-Holder.xlsx', sheet_name='Banks as Issuers')
issuer = issuer.head(15)
issuer
```

[ ]:

```
            Row Labels  Sum of Level  Proportions
0           Households    11192019.5     0.516596
1    Holding Companies     2846932.4     0.131407
2      Nonfin Corp Bus     2130217.0     0.098325
3   Nonfin Noncorp Bus     1758380.4     0.081162
4        Rest of World     1450067.2     0.066931
5    State/Local Govt.      799943.6     0.036923
6       GSE and Agency      629848.1     0.029072
7                  MMF      349837.0     0.016148
8            Life Ins.      259192.7     0.011964
9             Pensions      178190.3     0.008225
10      Broker/Dealers      155515.1     0.007178
11  Monetary Authority      134839.8     0.006224
12             PC Ins.       64781.0     0.002990
13   Finance Companies       64627.6     0.002983
14        Mutual Funds       55170.3     0.002547
```

[ ]:

```python
# Plot treemap
plt.figure(figsize=(12, 8))

# Define a palette of colors
custom_palette = sns.color_palette("rocket_r", n_colors=len(issuer))

# Plot treemap using squarify
squarify.plot(sizes=issuer['Sum of Level'], label=issuer['Row Labels'],␣
 ↪color=custom_palette, alpha=0.7, text_kwargs={'fontsize': 8})

# Remove axis
plt.axis('off')
plt.title('Banks as Issuers')
plt.show()
```



In the case of Banks as the Issuer of the Financial Instruments, acting as lender, the main counterparties are households (52%), Holding Companies (13%), Non-Fiancial Corporate Businesses (9%) and Non-Fiancial Non-Corporate Businesses.