

AlphaRank_DE

February 17, 2025

```
[22]: import requests
import json
import requests
import pandas as pd
import psycpg2

from psycpg2.extensions import ISOLATION_LEVEL_AUTOCOMMIT

from bs4 import BeautifulSoup
from datetime import datetime
from IPython import get_ipython
```

0.0.1 Check Data Lastest Date

```
[23]: # Get the needed API endpoint

summary_url = "https://banks.data.fdic.gov/api/summary?
↳fields=STNAME%2CYEAR%2CINTINC%2CEINTEXP%2CNIM%2CNONII%2CNONIX%2CELNATR%2CITAXR%2CIGLSEC%2CI

location_url = "https://banks.data.fdic.gov/api/locations?
↳fields=NAME%2CUNINUM%2CFI_UNINUM%2CCITY%2CSTNAME%2CZIP%2CCOUNTY&sort_by=NAME&sort_order=ASC

institution_url = "https://banks.data.fdic.gov/api/institutions?
↳filters=ACTIVE%3A1&fields=UNINUM%2CZIP%2CCITY%2CCOUNTY%2CSTNAME%2CSTALP%2CNAME%2CACTIVE%2CA
```

```
[24]: # Get request to the API

response_summary = requests.get(summary_url)
response_location = requests.get(location_url)
response_institution = requests.get(institution_url) # Active flag is 1
```

```
[25]: # Check if the requests are successful and parse the dates

for i in [response_summary, response_location, response_institution]:
    if i.status_code == 200:
        data = i.json()
```

```

        create_timestamp = data["meta"]["index"]["createTimestamp"]
        print(f"Successfully fetched the data for the date {create_timestamp}")
    else:
        print(f"Failed to fetch data.")

```

Successfully fetched the data for the date 2024-04-03T12:28:28Z

Successfully fetched the data for the date 2025-02-14T12:17:14Z

Successfully fetched the data for the date 2025-02-14T12:22:17Z

```

[26]: last_processed_date = datetime(2025, 2, 12)

location_date = datetime.strptime(response_location.
    ↳ json()["meta"]["index"]["createTimestamp"], "%Y-%m-%dT%H:%M:%SZ")
institution_date = datetime.strptime(response_institution.
    ↳ json()["meta"]["index"]["createTimestamp"], "%Y-%m-%dT%H:%M:%SZ")

```

0.0.2 Extract Banks Data

```

[27]: def fetch_all_data(api_url, params, limit=10000):

    all_data = []
    params["limit"] = limit
    params["offset"] = 0

    while True:
        # Make the request
        response = requests.get(api_url, params=params)
        response.raise_for_status() # Raise an error if the request fails

        data = response.json()

        # Check if there are records in the response
        if "data" in data and data["data"]:
            all_data.extend(data["data"]) # Append the new data
            print(f"Retrieved {len(data['data'])} records. Total so far:↳
↳ {len(all_data)}")

            # Break the loop if fewer than the limit of records were returned
            if len(data["data"]) < limit:
                break

            # Increment the offset for the next batch
            params["offset"] += limit
        else:
            print("No more data to fetch.")
            break

```

```

# Convert the list of JSON data to a pandas DataFrame
df = pd.json_normalize(all_data, sep="_")
return df

```

Locations

```

[28]: # If last processed date falls behind, need to re-fetch the data

# if (location_date > last_processed_date) or (institution_date >
↳ last_processed_date):
#     print("Running all cells below...")
#     shell = get_ipython()
#     shell.run_line_magic("run", "-i AllCellsBelow")

```

```

[29]: # API
api_url = "https://banks.data.fdic.gov/api/locations"
params = {
    "fields": "NAME,UNINUM,FI_UNINUM,CITY,STNAME,ZIP,COUNTY",
    "sort_by": "NAME",
    "sort_order": "ASC",
    "format": "json"
}

# Fetch all data and convert to DataFrame
df_locations = fetch_all_data(api_url, params)

# Display the first few rows of the DataFrame
print(df_locations.shape)
df_locations.head()

```

```

Retrieved 10000 records. Total so far: 10000
Retrieved 10000 records. Total so far: 20000
Retrieved 10000 records. Total so far: 30000
Retrieved 10000 records. Total so far: 40000
Retrieved 10000 records. Total so far: 50000
Retrieved 10000 records. Total so far: 60000
Retrieved 10000 records. Total so far: 70000
Retrieved 8868 records. Total so far: 78868
(78868, 9)

```

```

[29]:
  score data_ZIP    data_CITY data_FI_UNINUM data_STNAME data_COUNTY \
0      1   62230      Breese           9231   Illinois   Clinton
1      1   62231      Carlyle           9231   Illinois   Clinton
2      1   62216      Aviston           9231   Illinois   Clinton
3      1   62231      Carlyle           9231   Illinois   Clinton
4      1   63376  Saint Peters          429739  Missouri  St. Charles

```

	data_NAME	data_UNINUM	data_ID
0	1NB Bank	223055	223055
1	1NB Bank	232078	232078
2	1NB Bank	466427	466427
3	1NB Bank	9231	9231
4	1st Advantage Bank	429739	429739

Institution

```
[30]: # API parameters for the new API
api_url = "https://banks.data.fdic.gov/api/institutions"
params = {
    "filters": "ACTIVE:1",
    "fields": [
        "UNINUM,REPDTE,ZIP,CITY,COUNTY,STNAME,STALP,NAME,ACTIVE,ASSET,DEP",
    ],
    "sort_by": "REPDTE",
    "sort_order": "DESC",
    "format": "json"
}

# Fetch all data and convert to DataFrame
df_institutions = fetch_all_data(api_url, params)

# df_institutions["UNINUM"] = df_institutions["UNINUM"].astype(int)

# Display the first few rows of the DataFrame
print(df_institutions.shape)
df_institutions.head()
```

Retrieved 4490 records. Total so far: 4490
(4490, 13)

```
[30]: score data_ZIP    data_CITY  data_ACTIVE data_REPDTE data_STNAME \
0      0    53946    Markesan           1  09/30/2024  Wisconsin
1      0    53566    Monroe            1  09/30/2024  Wisconsin
2      0    54909    Almond             1  09/30/2024  Wisconsin
3      0    54757    New Auburn          1  09/30/2024  Wisconsin
4      0    54935    Fond Du Lac         1  09/30/2024  Wisconsin

data_ASSET data_STALP    data_DEP  data_COUNTY \
0    242674.0         WI    202952.0  Green Lake
1    452264.0         WI    344870.0    Green
2    210139.0         WI    173856.0  Portage
3    204089.0         WI    172964.0  Chippewa
4    2865827.0        WI    2323421.0  Fond Du Lac
```

	data_NAME	data_UNINUM	data_ID
0	Ergo Bank	6394	10004
1	Woodford State Bank	6400	10011
2	The Portage County Bank	6401	10012
3	Security Bank	6404	10015
4	National Exchange Bank and Trust	6419	10044

```
[31]: # print(df_summary.shape)
      # df_summary.head()
```

```
[32]: # print(df_location.shape)
      # df_location.head()
```

```
[33]: # df_institution["UNINUM"] = df_institution["UNINUM"].astype(int)
      # df_institution["ID"] = df_institution["ID"].astype(int)

      # print(df_institution.shape)
      # df_institution.head()
```

```
[34]: # # Merge three dataframes
      # merged_bank_0 = pd.merge(df_institution, df_location, left_on="UNINUM",
      #                           ↪right_on="FI_UNINUM", how="inner")

      # # Add a column to flag "Bank"
      # merged_bank_0["Type"] = "Bank"

      # print(merged_bank_0.shape)
      # merged_bank_0.head()
```

0.0.3 Extract Credit Union Data

```
[70]: # data_dec = "accountDescription_December2024.xlsx"
      # csv_df = pd.read_csv(data_dec, encoding='ISO-8859-1', delimiter=';')

      # # Display first few rows
      # print(csv_df.head())
```

```
[71]: # csv_path = "5310-All Charters(Dec2024-Dec2024).csv"
      # with open(csv_path, "r", encoding="utf-8") as file:
      #     for i, line in enumerate(file):
      #         if i == 214: # Line 215 (0-based index)
      #             print(f"Problematic line: {line}")
      # df = pd.read_csv(csv_path, on_bad_lines='skip')
      # df.head()
```

```
[69]: # with open(data_dec, "r", encoding="utf-8") as file:
#       for i, line in enumerate(file):
#           if i == 214: # Line 215 (0-based index)
#               print(f"Problematic line: {line}")
# df x= pd.read_csv(data_dec, on_bad_lines='skip')
# df.head()
```

```
# with open(data_dec, 'r', encoding='ISO-8859-1') as f:
#     for i in range(10): # Print first 10 lines
#         print(f"Line {i+1}: {f.readline()}")
```

```
[85]: # List of file paths
file_paths = ["522_Mar2023.xlsx", "897_Jun2023.xlsx", "372_Sep2023.xlsx",
↪ "495_Dec2023.xlsx", "515_Mar2024.xlsx", "585_Jun2024.xlsx", "566_Sep2024.
↪ xlsx"]

# Define sheets and columns
sheets_info = {
    "Total Accounts": ["Charter", "010"],
    "Shares and Deposits": ["Charter", "018"],
    "ProfileGenInfo": ["CUNumber", "CUName", "City", "State"]
}

# List to store merged dataframes
merged_dfs = []

# Iterate over each file
for file_path in file_paths:
    # Read data from each sheet
    dfs = {sheet: pd.read_excel(file_path, sheet_name=sheet, usecols=columns)
            for sheet, columns in sheets_info.items()}

    # Rename columns
    dfs["Total Accounts"].rename(columns={"010": "Total Assets"}, inplace=True)
    dfs["Shares and Deposits"].rename(columns={"018": "Total Shares and
↪ Deposits"}, inplace=True)

    # Merge dataframes
    merged_df_0 = pd.merge(dfs["Total Accounts"], dfs["Shares and Deposits"],
↪ on="Charter", how="inner")
    merged_df = pd.merge(merged_df_0, dfs["ProfileGenInfo"], left_on="Charter",
↪ right_on="CUNumber", how="inner")

    # Add a column to flag "CU"
    merged_df["Type"] = "CU"

    # Extract part after the last underscore and before the extension
```

```

report_quarter = file_path.rsplit("_", 1)[-1].split(".")[0]

# Add a column to mark report quarter
merged_df["report_quarter"] = report_quarter

# Store the merged dataframe
merged_dfs.append(merged_df)

# Combine all merged dataframes
final_df = pd.concat(merged_dfs, ignore_index=True)

# Display the first few rows
print(final_df.head())
print(final_df.shape)

```

	Charter	Total Assets	Total Shares and Deposits	CUNumber	\
0	1	12054517	10840893	1	
1	6	246363604	216235673	6	
2	12	62324176	58144271	12	
3	13	939319681	824618453	13	
4	16	10205662	8941200	16	

	CUName	State	City	Type	report_quarter	
0	MORRIS SHEPPARD	TEXARKANA	TX	TEXARKANA	CU	Mar2023
1	THE NEW ORLEANS FIREMEN'S	LA	Metairie	CU	Mar2023	
2	FRANKLIN TRUST	CT	Hartford	CU	Mar2023	
3	EFCU FINANCIAL	LA	Baton Rouge	CU	Mar2023	
4	WOODMEN	NE	OMAHA	CU	Mar2023	

(23662, 9)

[38]: # Merge and compare the change over last 2 quarters

```

merged_df_dep = pd.merge(merged_df_june, merged_df_sep, on="CUNumber",
    how="inner")
merged_df_dep

```

[38]:

	Charter_x	Total Assets_x	Total Shares and Deposits_x	CUNumber	\
0	566	3442694640	2714094872	566	
1	594	374754091	334915398	594	
2	1034	63640761	49291011	1034	
3	1074	1503029445	1256306639	1074	
4	1204	107666139	99193332	1204	
..	
238	68549	37038781	31548662	68549	
239	68579	9586191891	8063704676	68579	
240	68668	834000401	684989575	68668	
241	68712	3885613740	3340557844	68712	

242	68741	1140455731		1001307225	68741
	CUName_x	State_x	City_x	Type_x	Charter_y \
0	NUVISION	CA	HUNTINGTON BEAC	CU	566
1	PASADENA	CA	Pasadena	CU	594
2	OLIVE VIEW EMPLOYEES	CA	SYLMAR	CU	1034
3	FARMERS INSURANCE	CA	Burbank	CU	1074
4	RANCHO	CA	DOWNEY	CU	1204
..
238	MEDIA CITY	CA	BURBANK	CU	68549
239	PATELCO	CA	Dublin	CU	68579
240	1ST NORTHERN CALIFORNIA	CA	Martinez	CU	68668
241	VALLEY STRONG	CA	BAKERSFIELD	CU	68712
242	SESLOC	CA	SAN LUIS OBISPO	CU	68741

	Total Assets_y	Total Shares and Deposits_y	CUName_y \
0	3469113230	2731790560	NUVISION
1	364712289	324358927	PASADENA
2	63443173	48931060	OLIVE VIEW EMPLOYEES
3	1461148571	1229322187	FARMERS INSURANCE
4	108016124	99260328	RANCHO
..
238	38649552	32401070	MEDIA CITY
239	9525458343	7848994798	PATELCO
240	823837786	674249910	1ST NORTHERN CALIFORNIA
241	3925605060	3415504790	VALLEY STRONG
242	1132016926	1014397150	SESLOC

	City_y	State_y	Type_y
0	HUNTINGTON BEAC	CA	CU
1	Pasadena	CA	CU
2	SYLMAR	CA	CU
3	Burbank	CA	CU
4	DOWNEY	CA	CU
..
238	BURBANK	CA	CU
239	Dublin	CA	CU
240	Martinez	CA	CU
241	BAKERSFIELD	CA	CU
242	SAN LUIS OBISPO	CA	CU

[243 rows x 15 columns]

```
[86]: final_df.head()
```

```
[86]: Charter  Total Assets  Total Shares and Deposits  CUNumber  \
0         1         12054517         10840893         1
```


1	6	246363604	216235673	6
2	12	62324176	58144271	12
3	13	939319681	824618453	13
4	16	10205662	8941200	16

	CUName	State	City	Type	report_quarter	
0	MORRIS SHEPPARD	TEXARKANA	TX	TEXARKANA	CU	Mar2023
1	THE NEW ORLEANS FIREMEN'S	LA	Metairie	CU	Mar2023	
2	FRANKLIN TRUST	CT	Hartford	CU	Mar2023	
3	EFCU FINANCIAL	LA	Baton Rouge	CU	Mar2023	
4	WOODMEN	NE	OMAHA	CU	Mar2023	

0.0.4 1. How many banks and credit unions are active by asset tier (between \$500 M and \$1B)

```
[39]: df_institutions.head()
```

```
[39]:
```

	score	data_ZIP	data_CITY	data_ACTIVE	data_REPDTE	data_STNAME	\
0	0	53946	Markesan	1	09/30/2024	Wisconsin	
1	0	53566	Monroe	1	09/30/2024	Wisconsin	
2	0	54909	Almond	1	09/30/2024	Wisconsin	
3	0	54757	New Auburn	1	09/30/2024	Wisconsin	
4	0	54935	Fond Du Lac	1	09/30/2024	Wisconsin	

	data_ASSET	data_STALP	data_DEP	data_COUNTY	\
0	242674.0	WI	202952.0	Green Lake	
1	452264.0	WI	344870.0	Green	
2	210139.0	WI	173856.0	Portage	
3	204089.0	WI	172964.0	Chippewa	
4	2865827.0	WI	2323421.0	Fond Du Lac	

	data_NAME	data_UNINUM	data_ID
0	Ergo Bank	6394	10004
1	Woodford State Bank	6400	10011
2	The Portage County Bank	6401	10012
3	Security Bank	6404	10015
4	National Exchange Bank and Trust	6419	10044

```
[79]: # Bank Count

filtered_institutions = df_institutions[(df_institutions['data_ASSET'] >= 500000) & (df_institutions['data_ASSET'] <= 1000000)]
num_banks = filtered_institutions['data_UNINUM'].nunique()

# Credit Union Count
```

```

filtered_cu = final_df[(final_df['Total Assets'] >= 500000) & (final_df['Total_
↳Assets'] <= 1000000)]
num_cu = filtered_cu['CUNumber'].nunique()

print(f"Number of Active Banks That Have Total Assets Between $500M and $1B Is: ")
↳{num_banks}")
print(f"Number of Active Credit Unions That Have Total Assets Between $500M and
↳$1B Is: {num_cu}")

```

Number of Active Banks That Have Total Assets Between \$500M and \$1B Is: 774
Number of Active Credit Unions That Have Total Assets Between \$500M and \$1B Is:
93

0.0.5 2. Which banks and credit unions experienced >5% decline in deposits last quarter?

```

[41]: # merged_df_dep = pd.merge(merged_df_june, merged_df_sep, on="CUNumber",
↳how="inner")

# merged_df_dep = merged_df_dep.rename(columns={"Total Shares and Deposits_x":
↳"TotDep_June", "Total Shares and Deposits_y": "TotDep_Sep",})

# merged_df_dep["Diff_TotDep"] = merged_df_dep["TotDep_Sep"] -
↳merged_df_dep["TotDep_June"]

# merged_df_dep["%_Diff"] = 100 * round((merged_df_dep["Diff_TotDep"] /
↳merged_df_dep["TotDep_June"]), 2)

```

```

[42]: # merged_df_dep["%_Diff"] < -5.0

```

```

[42]: 0      False
      1      False
      2      False
      3      False
      4      False
      ...
      238    False
      239    False
      240    False
      241    False
      242    False
      Name: %_Diff, Length: 243, dtype: bool

```

```

[97]: final_df.head()

```

```
[97]: Charter Total Assets Total Shares and Deposits CUNumber \
0      1      12054517      10840893      1
1      6      246363604      216235673      6
2     12      62324176      58144271     12
3     13      939319681      824618453     13
4     16      10205662      8941200      16
```

	CUName	State	City	Type	report_quarter	
0	MORRIS SHEPPARD	TEXARKANA	TX	TEXARKANA	CU	Mar2023
1	THE NEW ORLEANS FIREMEN'S	LA	Metairie	CU	Mar2023	Mar2023
2	FRANKLIN TRUST	CT	Hartford	CU	Mar2023	Mar2023
3	EFCU FINANCIAL	LA	Baton Rouge	CU	Mar2023	Mar2023
4	WOODMEN	NE	OMAHA	CU	Mar2023	Mar2023

```
[43]: print(f"Banks and Credit Unions that have experienced > 5% decline in deposits_
↳last quarter are")

merged_df_dep[(merged_df_dep["%_Diff"] < -5.0)][["CUName_x", "City_x",
↳"State_x", "Total Assets_x", "TotDep_June", "TotDep_Sep"]]
```

Banks and Credit Unions that have experienced > 5% decline in deposits last quarter are

```
[43]: CUName_x City_x State_x Total Assets_x \
33      SKYONE Hawthorne CA 1012092407
43      MATTEL El Segundo CA 29889507
72 SANTA MARIA ASSOCIATED EMPLOYEES Santa Maria CA 5263378
80      ANTIOCH COMMUNITY Antioch CA 36096818
106     BOURNS EMPLOYEES Riverside CA 64680135
116     DELANCEY STREET SAN FRANCISCO CA 447905
151     BLUPEAK San Diego CA 1379011591
171     VISION ONE Sacramento CA 93985327
188     JONES METHODIST CHURCH SAN FRANCISCO CA 259339
235     FRONTWAVE Oceanside CA 1528672239
```

	TotDep_June	TotDep_Sep
33	856722400	747984533
43	26696295	24545843
72	4843103	4569806
80	31920212	29835770
106	58737627	55017027
116	321719	301488
151	1289449081	1186825650
171	78890443	74022595
188	206974	186873
235	1185761260	1118486290

```
[99]: # Keep the first occurrence of "Charter Name" for each "Charter"
df_name = final_df[["Charter", "CUName"]].drop_duplicates()

# Pivot the data to make "report_quarter" columns for easier calculations
df_pivot = final_df.pivot(index="Charter", columns="report_quarter",
    ↪values="Total Shares and Deposits")

# Calculate percentage change from Jun2024 to Sep2024
df_pivot["pct_change"] = ((df_pivot["Sep2024"] - df_pivot["Jun2024"]) /
    ↪df_pivot["Jun2024"]) * 100

# Reset index to make it a regular DataFrame
df_result = df_pivot.reset_index()

# Merge back with "Charter Name"
df_final = df_result.merge(df_name, on="Charter", how="left")
```

```
[100]: print(df_final)
```

	Charter	Dec2023	Jun2023	Jun2024	Mar2023	\
0	1	9.751785e+06	1.036751e+07	9.758484e+06	1.084089e+07	
1	6	2.341666e+08	2.098941e+08	2.392612e+08	2.162357e+08	
2	12	5.234440e+07	5.590718e+07	5.513259e+07	5.814427e+07	
3	13	8.856754e+08	8.369244e+08	9.267104e+08	8.246185e+08	
4	16	8.031106e+06	8.459457e+06	8.334581e+06	8.941200e+06	
...	
4712	68739	1.381355e+08	1.445525e+08	1.418589e+08	1.414363e+08	
4713	68740	NaN	NaN	0.000000e+00	NaN	
4714	68741	1.021962e+09	1.071368e+09	1.001307e+09	1.087556e+09	
4715	68742	6.013489e+07	6.117131e+07	6.293899e+07	6.201644e+07	
4716	68743	5.710965e+07	5.211237e+07	6.088248e+07	5.090896e+07	
	Mar2024	Sep2023	Sep2024	pct_change	\	
0	NaN	1.002802e+07	NaN	NaN		
1	NaN	2.223211e+08	NaN	NaN		
2	NaN	5.314996e+07	NaN	NaN		
3	NaN	8.527613e+08	NaN	NaN		
4	NaN	8.296547e+06	NaN	NaN		
...		
4712	NaN	1.427720e+08	NaN	NaN		
4713	NaN	NaN	NaN	NaN		
4714	1.018944e+09	1.043665e+09	1.014397e+09	1.307284		
4715	NaN	6.111434e+07	NaN	NaN		
4716	NaN	5.634592e+07	NaN	NaN		
	CUName					
0	MORRIS SHEPPARD TEXARKANA					

```

1      THE NEW ORLEANS FIREMEN'S
2          FRANKLIN TRUST
3          EFCU FINANCIAL
4          WOODMEN
...
4712          44 NORTH
4713          ARISE COMMUNITY
4714          SESLOC
4715      TEAMSTERS COUNCIL #37
4716      LIMESTONE FINANCIAL

```

[4717 rows x 10 columns]

```
[101]: df_final.columns
```

```
[101]: Index(['Charter', 'Dec2023', 'Jun2023', 'Jun2024', 'Mar2023', 'Mar2024',
            'Sep2023', 'Sep2024', 'pct_change', 'CUName'],
            dtype='object')
```

```
[103]: df_final[(df_final["pct_change"] < -5.0)][["CUName", "Jun2024", "Sep2024",
            ↪ "pct_change"]]
```

```
[103]:
```

	CUName	Jun2024	Sep2024	pct_change
776	SKYONE	8.567224e+08	7.479845e+08	-12.692310
1023	MATTEL	2.669630e+07	2.454584e+07	-8.055245
1534	SANTA MARIA ASSOCIATED EMPLOYEES	4.843103e+06	4.569806e+06	-5.643014
1618	ANTIOCH COMMUNITY	3.192021e+07	2.983577e+07	-6.530163
2024	FOX	4.069407e+07	3.845795e+07	-5.494952
2068	BOURNS EMPLOYEES	5.873763e+07	5.501703e+07	-6.334270
2396	DELANCEY STREET	3.217190e+05	3.014880e+05	-6.288407
3122	BLUPEAK	1.289449e+09	1.186826e+09	-7.958704
3516	VISION ONE	7.889044e+07	7.402260e+07	-6.170390
3744	JONES METHODIST CHURCH	2.069740e+05	1.868730e+05	-9.711848
4561	FRONTWAVE	1.185761e+09	1.118486e+09	-5.673568

```
[ ]:
```

```
[107]: db_connection = psycopg2.connect(host='127.0.0.1', dbname="postgres",
            ↪ user="postgres" , password="root")
```

```
[108]: db_connection.set_isolation_level(ISOLATION_LEVEL_AUTOCOMMIT)

        cursor = db_connection.cursor()
```

```
[109]: cursor.execute("CREATE DATABASE alpharank_creditunion;")
        db_connection.commit()
```

```
[110]: db_connection = psycopg2.connect(host='127.0.0.
↳1',dbname="alpharank_creditunion", user="postgres" , password="root")

cursor = db_connection.cursor()
```

```
[113]: # Comment the following line if you already created the postgis extension

cursor.execute("CREATE EXTENSION postgis;")
```

```
[ ]:
```