Explanatory Data Analysis (EDA) of regulated banks

Group 1
Lauren Bassett
Dima Mikhaylov
Cullan Bedwell
Casey Nguyen

FDIC
FEDERAL DEPOSIT INSURANCE CORPORATION

Introduction

Background information of topic

- Banking industry is going through a rapid consolidation as a number of regulated entities declined from 8,500 to roughly 5,000 in the last 20 years.
- From a preliminary survey, only 500 banks actually failed during these years. Consequently, 3,500 disappeared due to other reasons, such as mergers, charter changes and voluntary liquidation.

Main objectives of project

- Provide explanatory data analysis of banking industry's historical evolution
- Provide visual explanations of the consolidation
- Foster better understanding of historical trends and drivers

Data Set

- 4 files from FDIC representing 4 types of events (New Institutions, Liquidations, Business Combinations, Business Combinations – Failure) that can change the amount of active banks
- 2 summary tables with event counts by year for commercial banks and savings institutions
  - Aggregate up counts from the individual files to match the summary table counts
  - Extract information for each bank rather than simply the overall year counts

Data Preprocessing

- Deep dive into costs of failed banks
- Clean data (in csv format) using pandas library
- Missing values were imputed for NET FIRST NATIONAL BANK using 23% ‘avg_CostsToAssets’ ratio.
- Column ‘FAILDATE’ was converted to ‘datetime’ type and split into years, quarters, and months of failure.
- Created a function to clean the 4 files
  - Find the primary date column, convert it to datetime, and filter to only the years 2000-2020
  - Find the class types column and filter to only Commercial Banks or Savings Institutions
  - Allow for additional filtering on another column. This is used with Liquidations where there are banks pending sale that eventually show in the Business Combinations - Failure dataset
  - Filter to only the above columns + certificate id

Data Analysis

Data Processing for Retrieving all Regulatory Reports

- All regulated banks submit quarterly reports to the Federal Financial Institutions Examination Council (FFIEC). These statutory reports are aggregated and made available to the general public via SOAP APIs: **https://cdr.ffiec.gov/public/PWS/PWSPage.aspx**
- API client is stored in a separate folder and requires the zeep library to run:

  ../fdic_banks_eda/tree/main/notebooks/soap_client/ffipy
- Validating connection to the server can be performed by executing the following:

  'client = FFIEC_Client()'

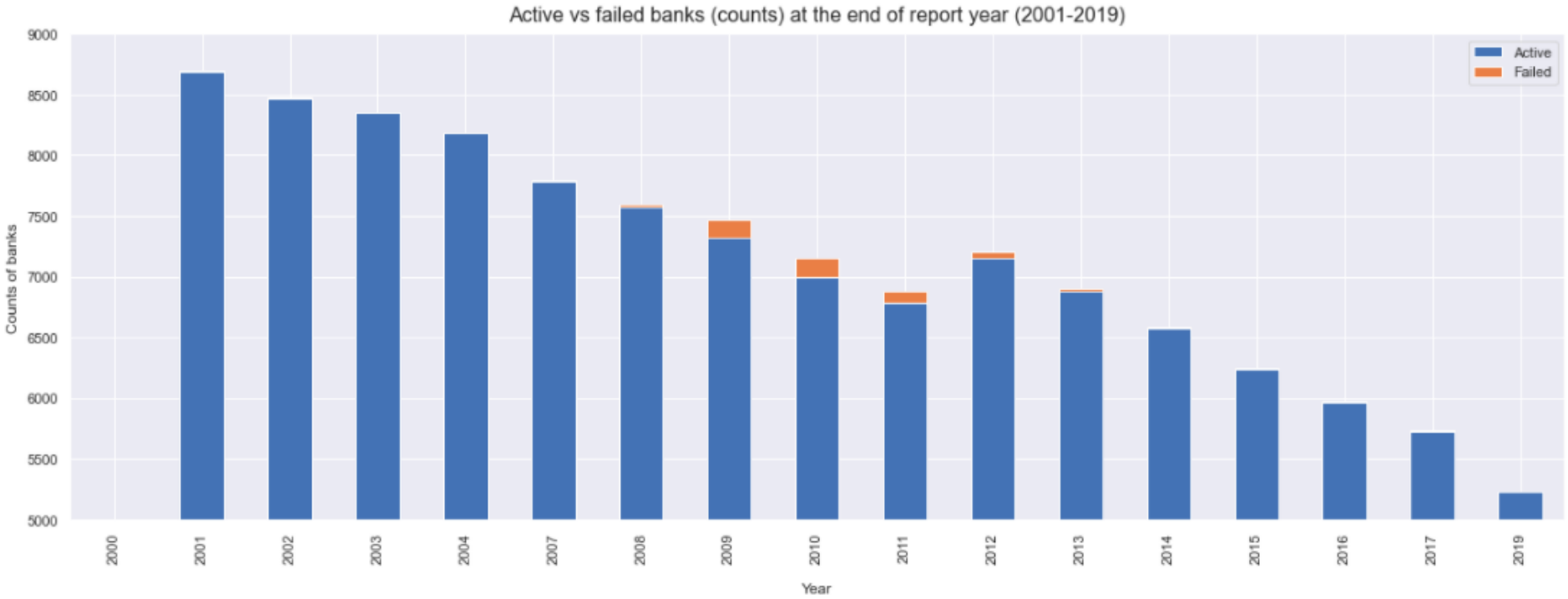
  'client.test_user_access()'

```python
class API_client_rest(unittest.TestCase):

# Test if still connected
    def test_connected(self):
        self.assertTrue(client.test_user_access())

# Test end of the period count
    def test_end_period_count(self, date='12/31/2019', count=5227):
        self.assertEqual(len(client.retrieve_panel_of_reporters(
            ds_name='Call', reporting_pd_end=date)), count)

# Test start of the period count
    def test_start_period_count(self, date='12/31/2001', count=8689):
        self.assertEqual(len(client.retrieve_panel_of_reporters(
            ds_name='Call', reporting_pd_end=date)), count)

# Test name of the bank
    def test_reporting_periods(self, report_type='Call', periods=83):
        self.assertEqual(len(client.retrieve_reporting_periods(ds_name=report_type)), periods)

if __name__ == "__main__":

  unittest.main(argv=['''], exit=False)
```

```
.../Users/dmitrymikhaylov/opt/anaconda3/lib/python3.8/site-packages/zeep/xsd/elements/indicators.py:617: ResourceWarn
ing: unclosed <ssl.SSLSocket fd=63, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('192.1
68.1.163', 57857), raddr=('192.59.35.198', 443)>
  item_subresult = element.parse_xmlelements(
ResourceWarning: Enable tracemalloc to get the object allocation traceback
.
----------------------------------------------------------------------
Ran 4 tests in 8.741s

OK
```

Data Analysis (cont.)

Summary plot that shows annual counts of failed banks stacked on top of active banks:



Data Analysis (cont.)

Data Processing for the Waterfall Chart

- Process the data to get counts by year - after running all four individual files through the cleaning function
- Created a function:
  - Aggregates the counts of the four cleaned files by year
  - Looks for years with no commercial banks or savings institutions to add a count of 0

Data Visualization - Histograms

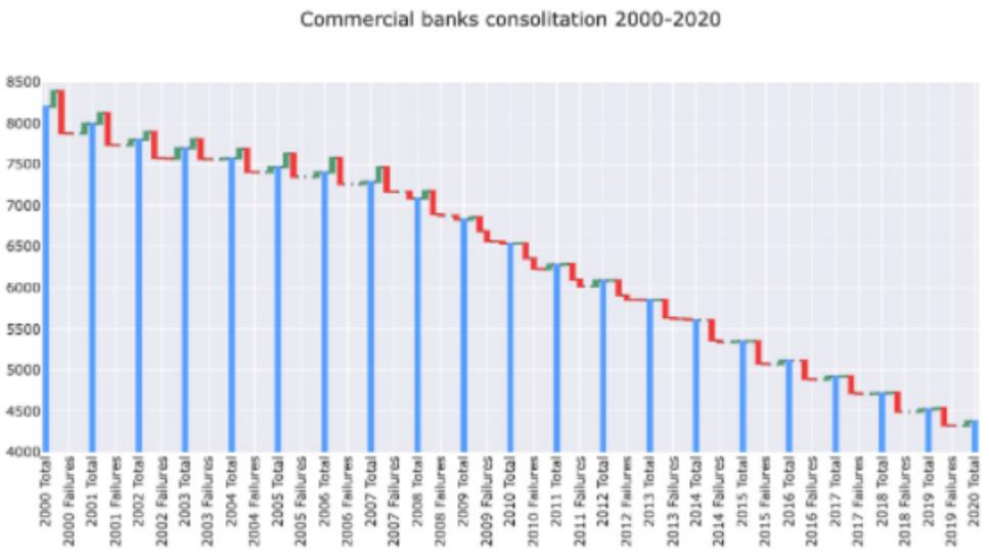
- Visualize the counts by year after getting all of the counts by year for the 4 files Display the counts as individual histograms
- Put into a function to configure the plots in a uniform way

Data Processing - Part 2

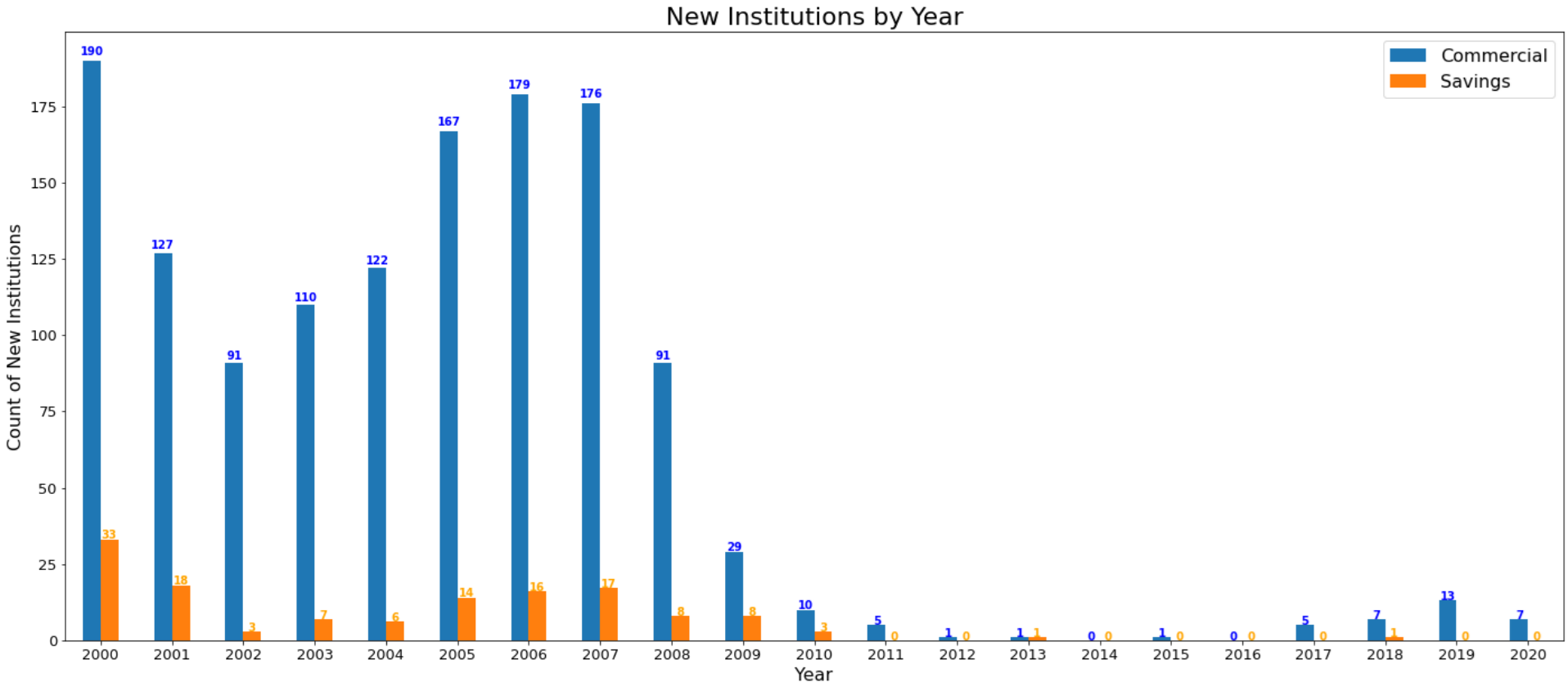
- Combine these counts to get an overall view of consolidation from 2000-2020
- Process the counts by year into a single data file
- Built a function that takes in the individual counts and the counts from the summary tables and combines them into a single DataFrame for plotting purposes

Data Visualization - Waterfall Chart

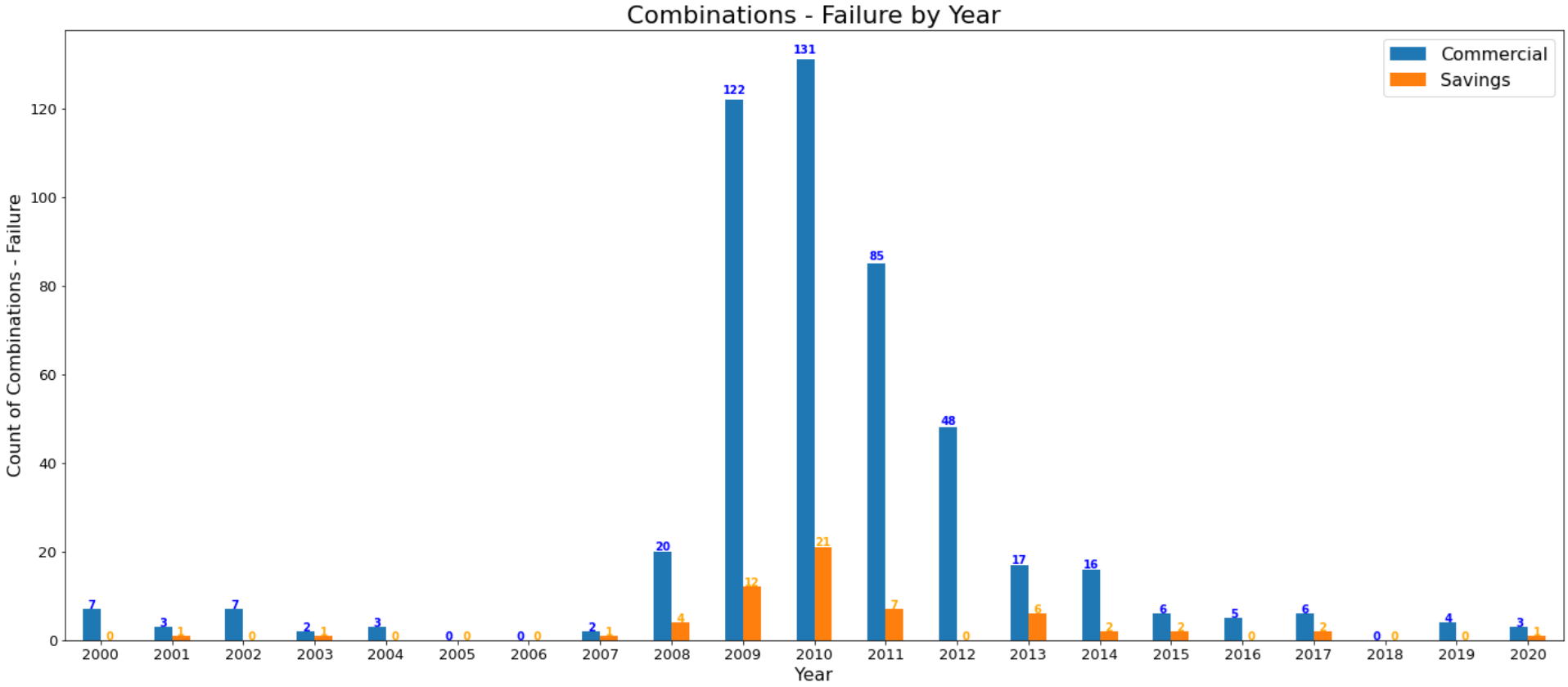
- Built an interactive plot from 2000 until 2020 that shows the overall consolidation of the industry using the four different types of events that change bank counts using Plotly



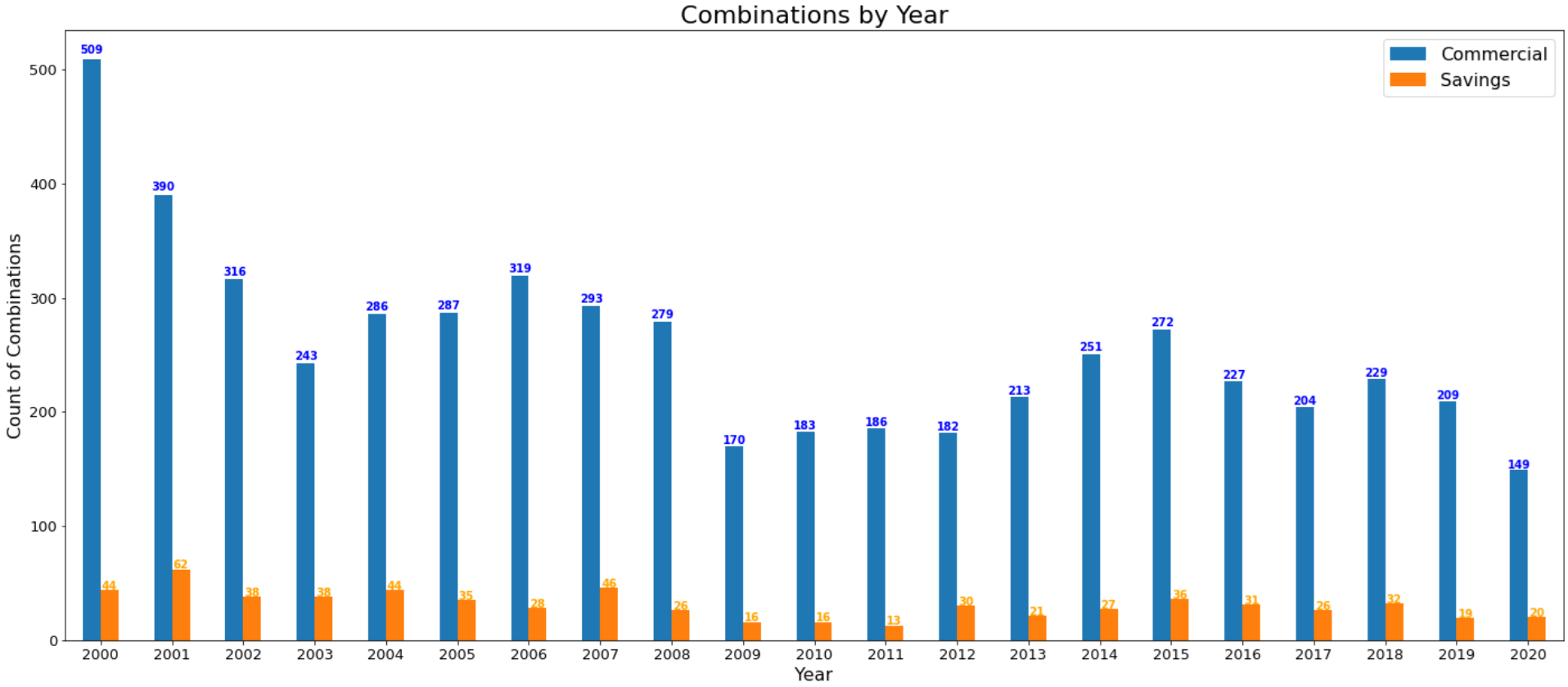
Results


New Institutions by Year
Commercial
Savings
Count of New Institutions
Year
0
25
50
75
100
125
150
175
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
190
33
127
18
91
3
110
7
122
6
167
14
179
16
176
17
91
8
29
8
10
3
5
0
1
0
1
1
0
0
1
0
0
0
5
0
7
1
13
0
7
0

Results


Combinations - Failure by Year
Commercial
Savings
Count of Combinations - Failure
Year
0
20
40
60
80
100
120
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
7
0
3
1
7
0
2
1
3
0
0
0
0
0
2
1
20
4
122
12
131
21
85
7
48
0
17
6
16
2
6
2
5
0
6
2
0
0
4
0
3
1

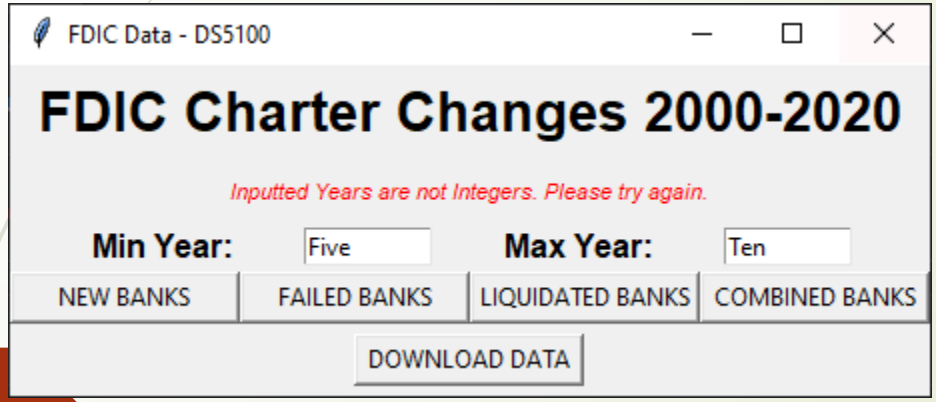
Results


Combinations by Year
Commercial
Savings
Count of Combinations
Year
0
100
200
300
400
500
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
509
390
316
243
286
287
319
293
279
170
183
186
182
213
251
272
227
204
229
209
149
44
62
38
38
44
35
28
46
26
16
16
13
30
21
27
36
31
26
32
19
20

Testing

GUI Error Handling

- Type Error



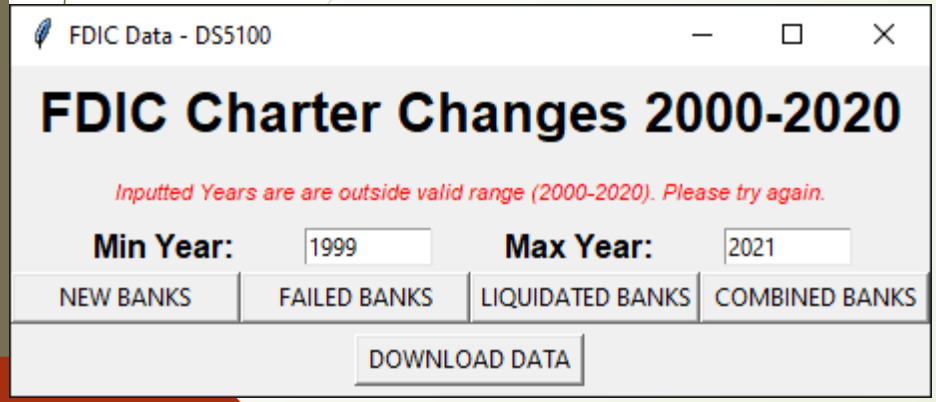

```python
class UserInputsTest(unittest.TestCase):

    def test_nan_inputs(self):

        # Looking for Type Errors
        with self.assertRaises(TypeError): ValidateInputYears('', '')
        with self.assertRaises(TypeError): ValidateInputYears('Five', 'Two')
        with self.assertRaises(TypeError): ValidateInputYears('2010', '')
        with self.assertRaises(TypeError): ValidateInputYears('', '2010')
        with self.assertRaises(TypeError): ValidateInputYears('abc', 'def')

    def test_out_of_range_inputs(self):

        # Looking for Value Errors
        with self.assertRaises(ValueError): ValidateInputYears('1999', '2010')
        with self.assertRaises(ValueError): ValidateInputYears('2000', '2021')
        with self.assertRaises(ValueError): ValidateInputYears('1800', '2200')

    def test_min_max_combo_inputs(self):

        # Looking for Value Error
        with self.assertRaises(ValueError): ValidateInputYears('2010', '2000')

    def test_valid_inputs(self):

        # Set Up
        min1, max1 = ValidateInputYears('2000', '2020')
        min2, max2 = ValidateInputYears('2005', '2008')
        min3, max3 = ValidateInputYears('2010', '2010')

        # Test Valid Inputs
        self.assertEqual(min1, 2000)
        self.assertEqual(min2, 2005)
        self.assertEqual(min3, 2010)
        self.assertEqual(max1, 2020)
        self.assertEqual(max2, 2008)
        self.assertEqual(max3, 2010)


unittest.main(argv=[''], exit = False)
```

```
....
----------------------------------------------------------------------
Ran 4 tests in 0.004s

OK
```

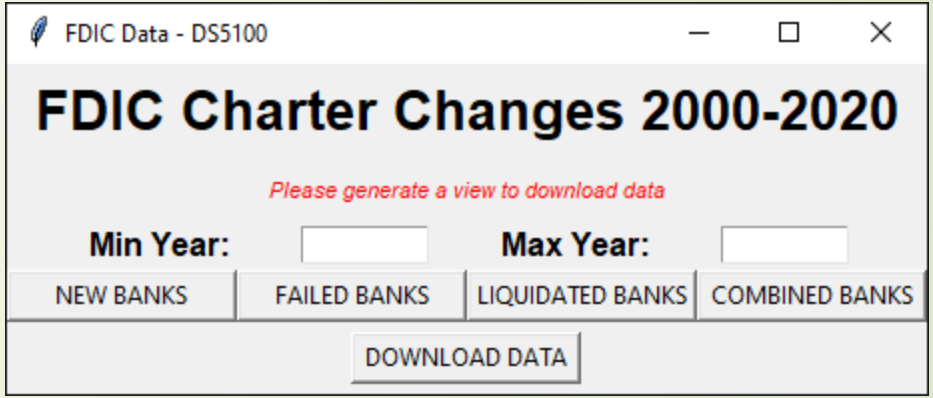
Testing (cont.)

GUI Error Handling

- Value Error



- Downloading Data with No Data Generated



Date Validation Function

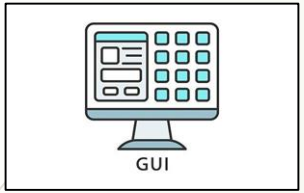
- Throws a TypeError if they're not integers
- Throws a ValueError if they're outside the valid range (2000-2020)
- Throws a ValueError if the maximum data is less than the minimum date

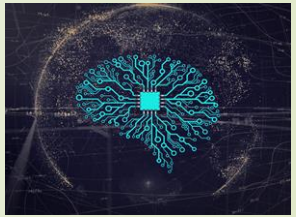
Conclusions

Consolidation over the last 20 years shows:

- The Great Recession caused lasting impact
  - There was a significant and lasting drop in new institutions
  - A large amount of banks failed in a short time period
- There have been a large number of business combinations (acquisitions) relative to new institutions

Future Opportunities





- Better combine multiple elements of our project together and drive some insights
- Query data using the GUI, download it, use the certificate ids and effective dates to pull financial call reports from the FDIC API, then create a model to predict bank failure from the finances

Thank you!