DATA WRANGLING PROJECT REPORT December 2022

TABLE OF CONTENTS

A. Project Overview

"Which Property Types can be a main focus for a Holder to recover before it is deemed as unclaimed?"

We picked the topic Unclaimed Property because it will always be a big business problem for financial and accounting institutes as they are the ones who deal with a huge volume of unclaimed property. Analysis of the information received from the two states (California & Massachusetts) and answers derived from this project will be of use to financial institutions, like banks, holding enormous funds for individuals and businesses. This can be used to help them audit these funds and report them to the state complying with the State's Unclaimed Property Law.

B. Introduction

What is an unclaimed property?

Unclaimed property is a personal asset separated from its original owner for several reasons [1]. It has had no activity or contact with the original owner for more than a year due to the individual's change of address or the fact it is there, is unknown to him/her.

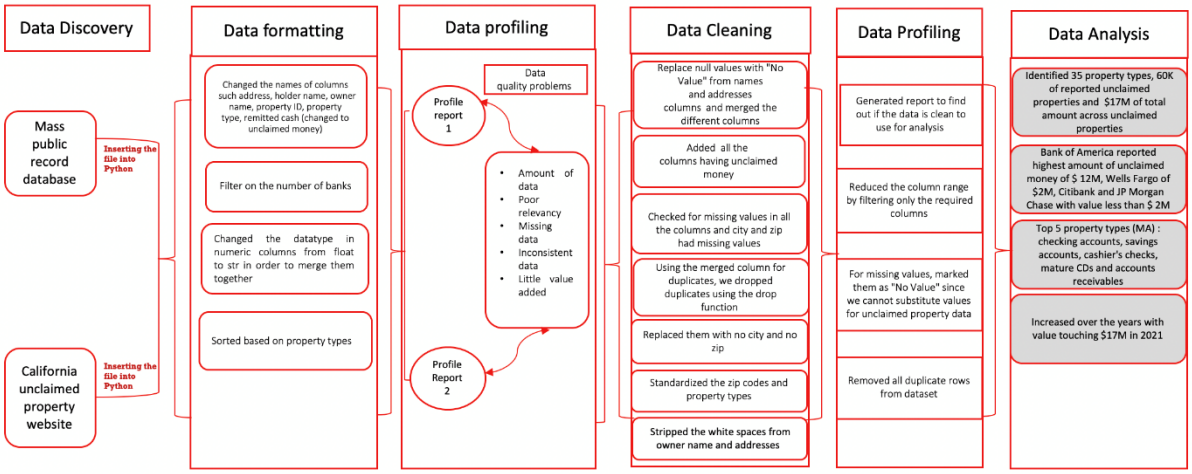
In simple words, if a business, government office, or any other source owes an individual or business money that they do not collect for a certain period due to relocation or absence of knowledge it is called Unclaimed property [2]

Some examples of unclaimed property may include – uncashed checks, savings or checking account deposits, pay checks, traveler's checks, money orders, tax refunds, and refund stocks or dividends.

Business Context

After formatting and cleaning the data, we identified 4 banks in each dataset that were estimated to have many unclaimed properties (1)

C. Data Wrangling Process



These steps for the data wrangling process are covered in more detail in the following paragraphs.

1. Data Discovery

Using the Unclaimed Property datasets from two states (for example unclaimed property data from the State of Massachusetts and the State of California), we have answered the questions of how we can derive the analysis for ease of unclaimed property recovery for holders. To collect the unclaimed property information from the State of Massachusetts, we will be using the Public Records law and to obtain the information and from the State of California, we have the data updated on their website for Unclaimed property information.

State	Source
California	Download Unclaimed Property Records (ca.gov)
Massachusetts	Public Records Request | Mass.gov

The variables in the data set are as follows:

- Property ID
- Property Type
- Date Reported
- Owner Name
- Co-Owner
- Address
- City

- State
- Zip
- Reported By - Holder

- **Challenges faced**

We received 2 .txt files from the Massachusetts public record. Both text files were huge (1.7 million rows), so we were only able to use 1 of them. We converted the Massachusetts text file into a .csv file in python so that we can view all the columns in a tableau format. For the California unclaimed dataset, we received 3 csv files (3 million rows). We merged the 3 csv files using python then proceeded into the data profiling step to see the quality of both our datasets.

2. Data Formatting

The data we received from the State of Massachusetts Unclaimed Property was in the form of multiple .txt files and from the State of California Unclaimed Property was received in multiple .csv files format.

- To read and understand the data in a table, the data for the state of MA was converted into and saved as a csv file through the following code. These two files with all the same columns were once again read on the jupyter notebook and merged to continue with further data wrangling. Whereas the data for CA was received as .csv files and the three files were read on jupyter notebook with the encoding as latin1 and were merged as they all had similar columns (2)(3)
  - The data received from both the states is in a structured format and as each record represents a single unclaimed property transaction for a single owner from a particular holder, the data follows a Fine granularity and all the records in the datasets are homogeneous.
  - To check which columns to pick from both the datasets, a function column was run which displayed all the column names in the output and helped us choose the right columns for us to proceed with further data wrangling and analysis. (4)(5)
  - For the MA State File, a subset was created by selecting the columns - 'PropertyID', ' ReportYear', ' NameLast', ' NameFirst', ' NameMiddle', ' Address1', ' Address2',' City', ' ZipCode', ' PropertyTypeCodeDescription', ' HolderName', ' RemittedCash', ' OriginalSecurityCashAmount', ' OriginalShares', ' RemainingShares', ' RemainingSecuritiesCash', which can be useful to get a desired output for the Massachusetts Unclaimed Property. Likewise, for the CA State File, a subset was created by selecting the columns - 'PROPERTY_ID', 'PROPERTY_TYPE', 'DATE_REPORTED','OWNER_NAME','ADDRESS','OWNER_CITY','OWNER_ZIP','CURRENT_ CASH_BALANCE', 'HOLDER_NAME', which can be useful to get a desired output for the California Unclaimed Property. (6)(7)
  - The formatting step to apply filters to our datasets based on the holder with the selected banks and subset it to make a clear agenda for our analysis. The banks selected and filtered to create a subset from the MA dataset were JP Morgan Chase Bank, Bank of America, Wells Fargo, Citibank and the same was applied in our code to filter holder names. Likewise, the banks selected and filtered to create a subset from the MA dataset

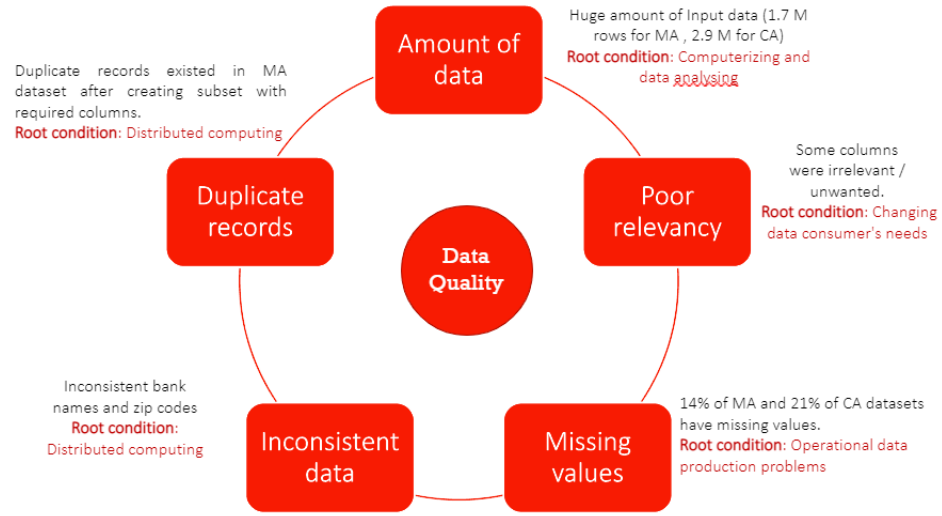
were JP Morgan Chase Bank, Bank of America, Wells Fargo, Bank of New York Mellon Corporate Trust and the same was applied in our code to filter holder names. (8)(9)

- A new column ADDRESS was inserted by merging the separated address1 and address2 columns in the MA dataset whereas in the CA dataset, the separated columns named OWNER_STREET_1, OWNER_STREET_2, OWNER_STREET_3 were merged to obtain the ADDRESS column (10)(11)
- In the MA dataset, the Owner names were split into First name, Last name and Middle name which were merged in an order to form the Owner Name. Also, the Unclaimed Amount was split into different columns – RemittedCash, OriginalSecurityCashAmount, OriginalShares, RemainingShares, RemainingSecuritiesCash which were numeric, and these were merged by adding up to a single value under Unclaimed Money. (12)(13)
- The last formatting step applied for both the datasets MA and CA were that they were sort based on the Property ID in ascending order (14)(15)

3. Data Profiling before Preprocessing

For California and Mass, we created separate data frames in Python and profiled them using pandas before cleaning them. Both datasets were analyzed and summarized information in the data profiling report.

Information Quality of the data



1. **Amount of data:** The MA and CA datasets received from the sources were huge datasets. We chose to use only one CSV file for MA as the tools (Python) we used were accepting only that. We chose to use 3 CSV files for CA. The final files used 3M rows for CA and 1.7 M rows for MA. The root condition for this problem was **computerizing and data analyzing** as large volumes of data made it difficult to access information in a reasonable amount of time.

2. **Poor relevancy:** Some columns in the MA and CA dataset were irrelevant to keep in the dataset. So, we had to subset them based on our business questions and context. Before data preprocessing, we had 28 columns in CA dataset and 27 columns for MA dataset. After Data preprocessing, we had 10 columns in CA dataset and 9 columns for MA dataset. The root conditions of this problem are **changing data consumer's needs** as consumer needs change all the time and once it changes the data becomes irrelevant. (16)(17)
3. **Missing values:** We found out that 14% of MA and 21% of CA datasets had missing values. The root condition of incomplete data is **operational data production problems**. This is because Null/NAN values can arise from operational problems.
4. **Inconsistent data:** The data had inconsistent bank names and zip codes. Information with different values may have been generated from multiple sources or created by inconsistent updating of multiple copies leading to inconsistent formats or values. Thus, **distributed computing** may have been the root condition to cause inconsistent representation (data quality pattern).
5. **Duplicate records:** The MA dataset has duplicate records. The root condition of duplicate records is **distributed computing**. In distributed computing, we could have a lot of different resources, which means that we have many records that could potentially be repeated.

Data Validation Rules

1. Precheck Zipcode - In the datasets there were issues with zip code: some were invalid, and some had more and less than 5 numbers. We must set a business rule for the zip code verification. The data collector must precheck the number of zip code and if it is invalid or not when collecting data. This reduces the number of inconsistent data in the dataset.
2. Data Type Check - The Unclaimed Money must be a number with the data type set as float. This can be set as a system validation check. For example, when the amount under unclaimed money is being entered it cannot be '$400' or 'four hundred dollars', it can just be 400 or 400.00. This check makes sure that we do not have inconsistencies within the amount entered under unclaimed money.
3. Subset creation: The type of validation rule for subset creation is Business. It is decided that the columns Property_ID, Property_Type, Year_Reported or Date_Reported, Owner_Name, Address, City, Zip, Holder_Name, Unclaimed_Money from the MA dataset were relevant for us. This solves the problem of poor relevancy and root condition changing data consumer's needs.
4. Check owner name and address: The type of validation rule to check for missing values in owner name and addresses is a system check. Since the name and address field cannot

be blank the Null/NAN values need to be masked. This solves the problem of incomplete data and root condition is operational data production problems.

4. Data Cleaning

The raw data received from both, the State of Massachusetts and the State of California had redundant, inessential, and bad data. To remove these inconsistencies, the data had to undergo through the cleansing stage by implementing certain cleaning steps or functions which helped improve the data quality and made it possible to read the data and understand the information.

The following data transformation steps were necessary to overcome the challenges faced to understand the data and its profile separately for both Massachusetts Unclaimed Property and California Unclaimed Property.

Steps followed to clean data received for MA Unclaimed Property:

- Size of data: As the data we received was huge i.e., more than 1.5 million rows and 26 columns. As the holder column had the information for the entity's holding the unclaimed property, we made a choice to go ahead with the banks which had more repetitions as the holder for multiple properties
- So, we chose the 4 banks or holders for each dataset which had reported the greatest number of unclaimed properties as this would direct and make our purpose clear for analysis.
- Poor relevancy: As a few columns were found irrelevant to be included in the output, a subset was created with relevant columns after removal of redundant columns (18).
- Missing data – The dataset had multiple null values and NaN columns which were cleaned during data wrangling process. To handle these missing and null values, they were replaced with blanks for columns Last name, first name and Middle name and were merged (19)
- Likewise, the null values from columns Address1, Address2 were replaced with blanks and were merged (20)
- The above step was applied to get rid of the missing values majorly from the Owner Names and Addresses.
- A function was run to check if there are any missing values in the dataset within other columns and we found out that City and Zip code columns had missing values, and these were handled by filling the respective column's fields with 'No City' and 'No Zip code' (21)
- Inconsistent data: We found the data in Zipcode to be inconsistent. The formats that were available were: 21276589, 2127 6589, 2127 | 6589, 2127, 6589. All these inconsistent values were corrected by stripping out the special characters between them and limiting the number of characters which had to be included in a zip code using the following code (22).
- The bank names under the Holder Names were inconsistent across the dataset such as BANK OF AMERICA - NORTH CAROLINA, BANK OF AMERICA-MASSACHUSETTS, BANK OF AMERICA CORPORATION, BANK OF AMERICA – FLORIDA, BANK OF AMERICA – TEXAS. BANK OF AMERICA - PRE-PAID CARD, BANK OF AMERICA - HOME LOANS, Bank of America N.A. F/K/A FIA Card Services NA. We have standardized all the Holder names into common and simpler terms, like Bank of America, Citibank and likewise for other banks with the code below (23).
- Duplicates: To get rid of the duplicate records from the dataset, we had created a new column which had combined data from the remaining columns. We used this new column to check for

duplicates and drop the duplicate records/rows from the complete dataset and to achieve this we had used the code shown below (24)

5. Data Profiling after preprocessing

- **How did we address it?**

We have addressed all the issues spotted in the profiling report generated before cleaning the data and generated a profiling report after all the cleaning process to verify if the data is clean and perfect to use for analysis. The report included the summary, visualization of data, and the range of unclaimed property values. We reduced the column range by filtering only the required columns. For the missing values we have marked them as 'No Value' since we cannot substitute values for the unclaimed property data. We have Removed all the duplicate rows from the

datasets. As per the data profiling report, after preprocessing our dataset is ready to process and analyze it. (25).

6. Data Analysis

We have created a dashboard with the number of properties, ranking by property types, heatmap across regions and time-graph for unclaimed properties.

Below is the data analysis for CA Dataset:

California Dashboard:(26)

- **Number of Unclaimed Property**

  While looking at the unclaimed property types across CA dataset --> we have identified 102 property types, 0.3 million number of reported unclaimed properties and $ 1.4 B of the total amount across unclaimed properties.

- **Ranking by banks**

  For CA dataset, the listing of unclaimed properties across banks provided Bank of America with the highest amount of unclaimed money of $0.8 B, followed by Wells Fargo with $0.4 B, then JP Morgan Chase with $0.17 B and followed by Bank of New York Mellon Corporation with $ 1.6 M.

- **Ranking by Unclaimed Property**

  For CA dataset, unclaimed property consists of the top 5 property types of namely cashier's checks, checking accounts (demand deposits), time deposits (certificate of deposit accounts), customer overpayments, mutual funds, and others.

- **Unclaimed Property Reported over the Years**

  For CA dataset, the unclaimed property has increased over the years from 1975-2021. We see a drop in unclaimed money to $ 48.6 M in the year 2007, thereafter, increasing at a CAGR of ~9% to touch $ 160 M in 2021.

- **Number of Property Types across Banks**

  For CA dataset, the number of property types across banks is split into cashier's checks, checking accounts, checking accounts (with demand deposits), mature CD, and savings accounts.

MA Dashboard:(27)

Number of Unclaimed Property

While looking at the unclaimed property types across MA dataset --> we have identified 35 property types, 60K number of reported unclaimed properties and $ 17M of the total amount across unclaimed properties.

- **Ranking by the banks**

  For MA dataset, the listing of unclaimed properties across banks shows Bank of America with the highest amount of unclaimed money of $12 M, followed by Wells Fargo with $2M, then Citibank and JP Morgan Chase each with value less than $ 2 M.

- **Ranking by Unclaimed Property**

  For MA dataset, unclaimed property consists of top 5 property types namely checking accounts, savings accounts, cashier's checks, mature CDs, and accounts receivables.

- **Number of Property Types across Banks**

For MA dataset, the number of property types across banks is split into cashier's checks, checking accounts, Any other outstanding check, mature CD, and savings accounts.

D. What we could have done

- Looked at other unclaimed property holders like insurance companies and see the number of property types they have.
- Enrich the datasets by adding another dataset that contains the address of holders.

Conclusion
We have concluded that the banks should focus on checking and saving accounts checking accoE. unts, savings accounts, cashier's checks, mature CDs, and accounts receivables because they have the highest amount of unclaimed property.

E. Methodology & Tools

In this project, we used the below software and programming language:

- **Python**: Much of the data wrangling process was performed on Python. We ran specific codes on Python to transform and clean the data for better understanding.
- **Microsoft Excel**: Excel was used to filter out 'banks' from the list of financial institutions and standardize the bank names. For example, Bank of America NA was modified to Bank of America
- **Tableau**: After the data was cleaned, formatted, and profiled, the final datasets were exported to Tableau to create a visual dashboard for the top 3 banks identified in both MA and CA datasets namely, Bank of America, JP Morgan Chase & Wells Fargo.
    - Using the variables – property type, unclaimed property value, bank name, year – we created multiple graphs for a visual representation of the final datasets.

F. Articles

a. Unclaimed property is reported to the state, and it is responsible for contacting the owner. Most Laws regarding unclaimed properties are set up by the Supreme court and most states choose to adopt these laws. The Revised Uniform Unclaimed law (RUUPA) was introduced in 2016 to include new digital property types to be considered as Unclaimed property. The aim of this law being set is to make state authority increase there to search for unclaimed property owners. In May 2022 it was reported by NBC news that Massachusetts has about $3.4 billion in unclaimed money. The Massachusetts treasury is responsible for handling unclaimed money. California is the state with the most unclaimed property rate in the United States with over $9 billion in Unclaimed property.

b. Most individuals in the public are not aware of what Unclaimed property is and that is one of the reasons there is an increase in the unclaimed property rate. In the article "Why Unclaimed property risk management is on the menu in 2022" the author explains how It is especially important for people to understand how unclaimed property is and what are the laws centered around it. Some businesses do not even claim their properties because

they want to avoid being taxed by the government. Unclaimed properties are the liability for businesses which can cause auditing issues if it is not stated. The state needs to create awareness about unclaimed property.

c. All 54 states in the United States have information on their website on how individuals and businesses can claim their Unclaimed property. Some websites like the National Association of State Treasurers help people find unclaimed property. Some unclaimed property law differs by state. Knowing the several types of Unclaimed properties will give us a brief understanding of the type of data we are dealing with. Reading the various articles, we have better knowledge and understanding of how to approach our project.

Some related data has been gathered on examples of how property can become "unclaimed":

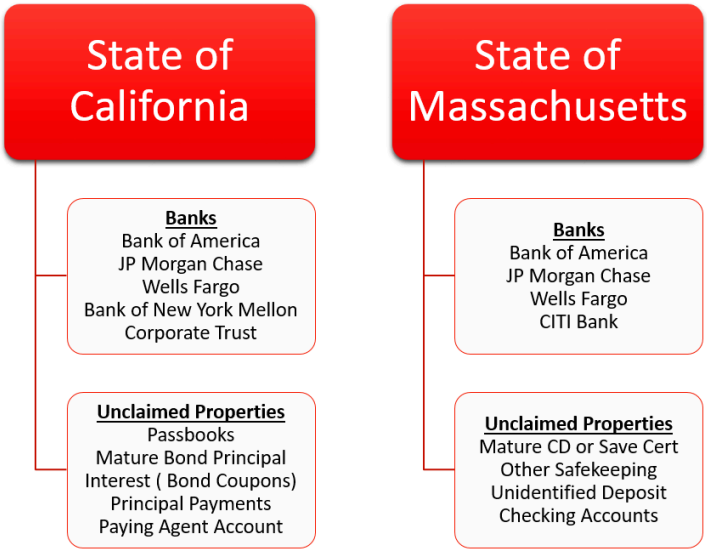
a. Life insurance policy: Over a decade ago, some insurers, including MetLife, Prudential, and John Hancock, went public and converted their companies into a stock form of ownership. As a result, this turned their policyholders into stockholders. At that time, they were supposed to pay out a certain amount to their customers. But despite their efforts to locate and notify all policyholders, they still have not been able to track down thousands of them.
b. Tax refund: The IRS claimed that it was holding on to over $1 billion in tax refunds for folks who failed to file their 2009 tax returns. Even if someone did not earn money in a particular year, the person may still be owed a refund due to certain tax credits and rebates.
c. Savings Bonds: Millions of savings bonds now worth billions of dollars have stopped earning interest but have not been redeemed. Since most bonds stop earning interest after 30 years, it gets overlooked to be cashed in.
d. Beneficiary of a pension with a company that went bankrupt: If someone once worked for a firm that went out of business and did not receive a pension that was earned under their plan, then the person may have retirement funds waiting to be retrieved.
e. Forgotten money: It can include unclaimed property left in safe deposit boxes, uncashed payroll checks, bank account monies, utility deposits or life insurance proceeds that have been forgotten about.

- **Sources:**

1. *https://unclaimed.org/what-is-unclaimed-property/*
2. *https://www.forbes.com/sites/forbesfinancecouncil/2022/04/06/unclaimed-property-in-a-digital-world/?sh=50e3540e10e6*
3. *https://www.nbcboston.com/investigations/consumer/mass-has-3-4b-in-unclaimed-money-and-these-people-want-to-help-give-it-back/2722208/*
4. *https://www.collibra.com/us/en/blog/the-7-most-common-data-quality-issues*
5. *https://www.techtarget.com/searchdatamanagement/feature/Data-quality-for-big-data-Why-its-a-must-and-how-to-improve-it*
6. *https://www.forbes.com/sites/forbesfinancecouncil/2022/04/06/unclaimed-property-in-a-digital-world/?sh=50e3540e10e6*
7. *https://taxexecutive.org/why-unclaimed-property-risk-management-is-on-the-menu-in-2022/*
8. *https://ucpi.sco.ca.gov/*
9. *How to find unclaimed money from the IRS and others (winknews.com)*
10. *Unclaimed Money from the Government | USAGov*
11. *Massachusetts Unclaimed Funds (unclaimed-funds.org)*
12. *Unclaimed Insurance Policies Over $350 million in California | UnclaimedMoney.org*
13. *6 tips for finding unclaimed money (onemilitary.net)*
14. *Americans owed billions in unclaimed money | wgad.com*
15. *Shouldn't Regulators Be Accountable for Returning Rs82,000 Crore of Unclaimed Money to Savers? | NewsClick*

G. Appendix

(1)



(2)

```
In [2]: #Importing the Datasets for Massachusetts Unclaimed Porperty
MA2 = pd.read_csv (r'/Users/jonathangabriel/Documents/Data Wrangling Project/Datasets/MA2.txt')
MA2.to_csv (r'/Users/jonathangabriel/Documents/Data Wrangling Project/Datasets/Mass1.csv', index=None)

MA3 = pd.read_csv (r'/Users/jonathangabriel/Documents/Data Wrangling Project/Datasets/MA3.txt')
MA3.to_csv (r'/Users/jonathangabriel/Documents/Data Wrangling Project/Datasets/Mass2.csv', index=None)
```

```
In [3]: #Reading the file to be saved as a csv file.

ab1 = pd.read_csv('Mass1.csv', low_memory = False )
ab2 = pd.read_csv('Mass2.csv', low_memory = False )
```

```
In [4]: #Merge the datasets

result = [ab1, ab2]
df_MA = pd.concat(result)
```

(3)

```
In [2]: #Import Datasets

df1 = pd.read_csv('CA1.csv', encoding='latin1', low_memory = False)
df2 = pd.read_csv('CA2.csv', encoding='latin1', low_memory = False)
df3 = pd.read_csv('CA3.csv', encoding='latin1', low_memory = False)
```

```
In [3]: #Merge the datasets

result = [df1, df2, df3]
df_CA = pd.concat(result)
```

(4)

```
In [159]: df_MA.columns

Out[159]: Index(['PropertyID', ' ReportYear', ' NameLast', ' NameFirst', ' NameMiddle',
                 ' Address1', ' Address2', ' City', ' ZipCode', ' OwnerCount',
                 ' OwnerType', ' RelationshipType', ' PropertyTypeCD',
                 ' PropertyTypeCodeDescription', ' HolderName', ' CUSIPNumber',
                 ' SecurityName', ' RemittedCash', ' OriginalSecurityCashAmount',
                 ' OriginalShares', ' RemainingShares', ' RemainingSecuritiesCash',
                 ' CheckNo', ' HasSecurities', ' HasCash', ' HasTangible '],
                dtype='object')
```

(5)

```
In [6]: df_CA.columns

Out[6]: Index(['PROPERTY_ID', 'PROPERTY_TYPE', 'CASH_REPORTED', 'SHARES_REPORTED',
               'NAME_OF_SECURITIES_REPORTED', 'DATE_REPORTED', 'NO_OF_OWNERS',
               'OWNER_NAME', 'OWNER_STREET_1', 'OWNER_STREET_2', 'OWNER_STREET_3',
               'OWNER_CITY', 'OWNER_STATE', 'OWNER_ZIP', 'OWNER_COUNTRY_CODE',
               'CURRENT_CASH_BALANCE', 'NUMBER_OF_PENDING_CLAIMS',
               'NUMBER_OF_PAID_CLAIMS', 'DATE_OF_LAST_CONTACT', 'HOLDER_NAME',
               'HOLDER_STREET_1', 'HOLDER_STREET_2', 'HOLDER_STREET_3', 'HOLDER_CITY',
               'HOLDER_STATE', 'HOLDER_ZIP', 'CUSIP'],
              dtype='object')
```

(6)

```
In [160]: #Selecting certain useful columns.

          df_MA = df_MA[['PropertyID', ' ReportYear', ' NameLast', ' NameFirst', ' NameMiddle', ' Address1',
                         ' Address2',' City', ' ZipCode', ' PropertyTypeCodeDescription', ' HolderName', ' RemittedCash',
                          ' OriginalSecurityCashAmount', ' OriginalShares', ' RemainingShares', ' RemainingSecuritiesCash']]
```

(7)

```
In [10]: #Subsetting with the required columns
         df_CA_subset = df_CA[['PROPERTY_ID', 'PROPERTY_TYPE', 'DATE_REPORTED','OWNER_NAME', 'ADDRESS','OWNER_CITY',
                               'OWNER_ZIP', 'CURRENT_CASH_BALANCE', 'HOLDER_NAME']]
         df_CA_subset
```

(8)

```
In [164]: #Filtering out chosen bank names

          df_MA = df_MA[df_MA[' HolderName'].str.contains("JP MORGAN CHASE BANK|BANK OF AMERICA|WELLS FARGO|CITIBANK")]

          df_MA
```

Out[164]:

st	NameFirst	NameMiddle	Address1	Address2	City	ZipCode	PropertyTypeCodeDescription	HolderName	RemittedCash	OriginalSecurityCashA
CZ	MARYNIA	E	52 TROWBRIDGE ST APT 8	NaN	CAMBRIDGE	02138	Accounts Receivable Credit Balances	CITIBANK, N.A.	3.48	
IN	APRIL	M	56 EASTLAND RD	NaN	JAMAICA PLAIN	02130	Accounts Receivable Credit Balances	CITIBANK, N.A.	0.66	

(9)

```
In [13]: #Subset with only JPMORGAN CHASE BANK , BANK OF AMERICA , WELLS FARGO , BANK OF NEW YORK MELLON CORPORATE TRUST
         df_CA_Banksubset = df_CA_subset[df_CA_subset['HOLDER_NAME'].str.contains
                                         ("JPMORGAN CHASE BANK|BANK OF AMERICA|WELLS FARGO|BANK OF NEW YORK MELLON CORPORATE TRUST")]
         df_CA_Banksubset
```

Out[13]:

	PROPERTY_ID	PROPERTY_TYPE	DATE_REPORTED	OWNER_NAME	ADDRESS	CITY	ZIPCODE	UNCLAIMED MONEY	HOLDER_NAME
0	189993	AC51: Bank of America Passbooks	1989-02-05	ROGERS ROBERT	No ADDRESS	No City	No Zipcode	500.00	BANK OF AMERICA - PASSBOOK ACCOUNTS
11	341109	56 : Checking accounts, dmnd depsit	1976-05-07	MOORE MABEL	No ADDRESS	No City	No Zipcode	500.00	WELLS FARGO BANK N A

(10)

In [170]:
```
#Replacing null values from columns Address1, Address1 and merging them together.
df_MA[' Address1'] = df_MA[' Address1'].fillna('')
df_MA[' Address2'] = df_MA[' Address2'].fillna('')
df_MA.insert(loc = 3, column = 'Address', value = df_MA[' Address1'] + " " + df_MA[' Address2'].map(str))

del df_MA[' Address1']
del df_MA[' Address2']
```

(11)

In [8]:
```
#Created a new column Owner's address by merging 'OWNER_STREET_1', 'OWNER_STREET_2', 'OWNER_STREET_3'
df_CA.insert(loc = 8, column = 'ADDRESS', value = df_CA['OWNER_STREET_1'].map(str) + ' ' +
             df_CA['OWNER_STREET_2'].map(str) + ' ' +  df_CA['OWNER_STREET_3'].map(str))
df_CA['ADDRESS']
```

(12)

In [166]:
```
#Replacing null values from columns Last name, First name and Middle name and merging them together.
df_MA[' NameLast'] = df_MA[' NameLast'].fillna('')
df_MA[' NameFirst'] = df_MA[' NameFirst'].fillna('')
df_MA[' NameMiddle'] = df_MA[' NameMiddle'].fillna('')

df_MA.insert(loc = 2, column = 'Owner_Name', value = df_MA[' NameFirst']
             + " " + df_MA[' NameMiddle'] + " " + df_MA[' NameMiddle'].map(str))
```

(13)

In [172]:
```
#Merging the amount columns.
df_MA.insert(loc = 8, column = 'Unclaimed_Money', value = df_MA[' RemittedCash'] +
             df_MA[' OriginalSecurityCashAmount'] +df_MA[' OriginalShares']
             + df_MA[' RemainingShares'] + df_MA[' RemainingSecuritiesCash'])

del df_MA[' RemittedCash']
del df_MA[' OriginalSecurityCashAmount']
del df_MA[' OriginalShares']
del df_MA[' RemainingShares']
del df_MA[' RemainingSecuritiesCash']

df_MA
```

Out[172]:

	PropertyID	ReportYear	Owner_Name	Address	City	Zipcode	PropertyTypeCodeDescription	HolderName	Unclaimed_Money
0	26212501	2020	MARYNIA E E	52 TROWBRIDGE ST APT 8	CAMBRIDGE	02138	Accounts Receivable Credit Balances	CITIBANK	3.48

(14)

In [189]:
```
#Sorting the data based on Property ID.
df_MASS = df_MA.sort_values(by=['PROPERTY_ID'])
df_MASS = df_MA.reset_index(drop=True)
df_MASS
```

Out[189]:

	PROPERTY_ID	YEAR_REPORTED	OWNER_NAME	ADDRESS	CITY	ZIP	PROPERTY_TYPE	HOLDER_NAME	UNCLAIMED_MONEY
0	25361511	2019	LISE	CO TEF TUFF NA 39 DODGE ST	BEVERLY	01915	Underlying shares	JP MORGAN CHASE BANK	26.000
1	25361512	2019	GREGORY P P	11 ROYCE RD APT 32	ALLSTON	02134	Underlying shares	JP MORGAN CHASE BANK	2.000

(15)

```
In [17]: Output_df_CA = df_CA_Banksubset.sort_values(by='PROPERTY_ID')
         Output_df_CA
```

Out[17]:

	PROPERTY_ID	PROPERTY_TYPE	DATE_REPORTED	OWNER_NAME	ADDRESS	CITY	ZIPCODE	UNCLAIMED MONEY	HOLDER_NAME
942745	7242	AC51: Bank of America Passbooks	1989-02-05	WILLIAMS AILEEN G	1237 FRANCIS ST APT 15	LOGMONT	80501	1495.81	BANK OF AMERICA
470151	8473	AC51: Bank of America Passbooks	1989-02-05	SIU FRANCIS	948 GUERRERO	SAN FRANCISCO	94110	590.67	BANK OF AMERICA

(16)

Dataset statistics

Number of variables	28
Number of observations	2987149
Missing cells	17641896
Missing cells (%)	21.1%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	638.1 MiB
Average record size in memory	224.0 B

(17)

Dataset statistics

Number of variables	27
Number of observations	1763565
Missing cells	6864482
Missing cells (%)	14.4%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	363.3 MiB
Average record size in memory	216.0 B

(18)

```
In [79]: df_MA.columns
```

```
Out[79]: Index(['PropertyID', ' ReportYear', ' NameLast', ' NameFirst', ' NameMiddle',
                ' Address1', ' Address2', ' City', ' ZipCode', ' OwnerCount',
                ' OwnerType', ' RelationshipType', ' PropertyTypeCD',
                ' PropertyTypeCodeDescription', ' HolderName', ' CUSIPNumber',
                ' SecurityName', ' RemittedCash', ' OriginalSecurityCashAmount',
                ' OriginalShares', ' RemainingShares', ' RemainingSecuritiesCash',
                ' CheckNo', ' HasSecurities', ' HasCash', ' HasTangible '],
               dtype='object')
```

```
In [80]: #Selecting certain useful columns.

         df_MA = df_MA[['PropertyID', ' ReportYear', ' NameLast', ' NameFirst', ' NameMiddle', ' Address1',
                        ' Address2',' City', ' ZipCode', ' PropertyTypeCodeDescription', ' HolderName', ' RemittedCash',
                        ' OriginalSecurityCashAmount', ' OriginalShares', ' RemainingShares', ' RemainingSecuritiesCash',
                        ' CheckNo', ' HasSecurities']]
```

(19)

In [83]:
```
#Replacing null values from columns Last name, First name and Middle name and merging them together.
df_MA[' NameLast'] = df_MA[' NameLast'].fillna('')
df_MA[' NameFirst'] = df_MA[' NameFirst'].fillna('')
df_MA[' NameMiddle'] = df_MA[' NameMiddle'].fillna('')

df_MA.insert(loc = 2, column = 'Owner_Name', value = df_MA[' NameFirst'] + " " + df_MA[' NameMiddle'] + " " + df_MA[' NameMiddle'
```

In [84]:
```
del df_MA[' NameLast']
del df_MA[' NameFirst']
del df_MA[' NameMiddle']

df_MA
```

Out[84]:

	PropertyID	ReportYear	Owner_Name	Address1	Address2	City	ZipCode	PropertyTypeCodeDescription	HolderName	RemittedCash	
0	26212501	2020	MARYNIA E E	52 TROWBRIDGE ST APT 8	NaN	CAMBRIDGE	02138	Accounts Receivable Credit Balances	CITIBANK, N.A.	3.48	
1	26212502	2020	APRIL M M	56 EASTLAND RD	NaN	JAMAICA PLAIN	02130	Accounts Receivable Credit Balances	CITIBANK, N.A.	0.66	
2	26212503	2020	KEVIN C C	43 BOLAS RD	NaN	DUXBURY	02332	Accounts Receivable Credit Balances	CITIBANK, N.A.	0.88	
				13							

(20)

In [87]:
```
#Replacing null values from columns Address1, Address1 and merging them together.
df_MA[' Address1'] = df_MA[' Address1'].fillna('')
df_MA[' Address2'] = df_MA[' Address2'].fillna('')
df_MA.insert(loc = 3, column = 'Address', value = df_MA[' Address1'] + " " + df_MA[' Address2'].map(str))

del df_MA[' Address1']
del df_MA[' Address2']
```

(21)

In [174]:
```
for x in df_MA.columns:
    missing_val= df_MA[x].isnull().sum()

    if missing_val>0:
        print("{} has {} missing value(s)".format(x,missing_val))
    else:
        print("{} has NO missing value!".format(x))
```

```
PROPERTY_ID has NO missing value!
YEAR_REPORTED has NO missing value!
OWNER_NAME has NO missing value!
ADDRESS has NO missing value!
CITY has 557 missing value(s)
ZIP has 1447 missing value(s)
PROPERTY_TYPE has NO missing value!
HOLDER_NAME has NO missing value!
UNCLAIMED_MONEY has NO missing value!
```

In [175]:
```
#Filling Missing values for No Cities and No Zip
df_MA.CITY = df_MA.CITY.fillna('No City')
df_MA.ZIP = df_MA.ZIP.fillna('No ZIP')
```

(22)

```
In [82]: #Cleaning and giving the zipcode a proper format.

df_MA['Zip'] = df_MA[' ZipCode'].str.replace(' |-','')
df_MA['Zip'] = df_MA['Zip'].str.slice(stop=5)
df_MA
```

```
/var/folders/6_/g9lywzxj5hb6vln3ckbxczlr0000gn/T/ipykernel_65355/3371020668.py:3: FutureWarning: The default value of regex wil
l change from True to False in a future version.
  df_MA['Zip'] = df_MA[' ZipCode'].str.replace(' |-','')
```

Out[82]:

	PropertyID	ReportYear	NameLast	NameFirst	NameMiddle	Address1	Address2	City	ZipCode	PropertyTypeCodeDescription	Hold
0	26212501	2020	MACKIEWICZ	MARYNIA	E	52 TROWBRIDGE ST APT 8	NaN	CAMBRIDGE	02138	Accounts Receivable Credit Balances	CI
1	26212502	2020	MACKIN	APRIL	M	56 EASTLAND RD	NaN	JAMAICA PLAIN	02130	Accounts Receivable Credit Balances	CI
2	26212503	2020	MACKIN	KEVIN	C	43 BOLAS RD	NaN	DUXBURY	02332	Accounts Receivable Credit Balances	CI
3	26212504	2020	MACKIN	SCOTT	NaN	13 PROSPECT ST	NaN	NEWBURYPORT	01950	Accounts Receivable Credit Balances	CI
4	26212505	2020	MACKINNON	BRIAN	J	7 STRONG ST	APT 2	NEWBURYPORT	01950	Accounts Receivable Credit Balances	CI
...	...	...	...	...	...	...	...	...	...	...	
548090	26214873	2020	MISNER	BETH	J	4 FORGE HILL RD	APT 313	FRANKLIN	02038	Accounts Receivable Credit Balances	CI

(23)

```
In [85]: #Standardizing bank names.
df_MA.loc[df_MA[' HolderName'].str.contains("BANK OF AMERICA"), ' HolderName']="BANK OF AMERICA"
df_MA.loc[df_MA[' HolderName'].str.contains("JP MORGAN CHASE"), ' HolderName']="JP MORGAN CHASE BANK"
df_MA.loc[df_MA[' HolderName'].str.contains("WELLS FARGO"), ' HolderName']="WELLS FARGO BANK"
df_MA.loc[df_MA[' HolderName'].str.contains("CITIBANK"), ' HolderName']="CITIBANK"
```

```
In [86]: df_MA
```

Out[86]:

	PropertyID	ReportYear	Owner_Name	Address1	Address2	City	ZipCode	PropertyTypeCodeDescription	HolderName	RemittedCash
0	26212501	2020	MARYNIA E E	52 TROWBRIDGE ST APT 8	NaN	CAMBRIDGE	02138	Accounts Receivable Credit Balances	CITIBANK	3.48
1	26212502	2020	APRIL M M	56 EASTLAND RD	NaN	JAMAICA PLAIN	02130	Accounts Receivable Credit Balances	CITIBANK	0.66
2	26212503	2020	KEVIN C C	43 BOLAS RD	NaN	DUXBURY	02332	Accounts Receivable Credit Balances	CITIBANK	0.88
3	26212504	2020	SCOTT	13 PROSPECT ST	NaN	NEWBURYPORT	01950	Accounts Receivable Credit Balances	CITIBANK	91.56
4	26212505	2020	BRIAN J J	7 STRONG ST	APT 2	NEWBURYPORT	01950	Accounts Receivable Credit Balances	CITIBANK	30.45
...	...	...	...	...	...	...	...	...	...	...
548090	26214873	2020	BETH J J	4 FORGE HILL RD	APT 313	FRANKLIN	02038	Accounts Receivable Credit Balances	CITIBANK	150.89

(24)

```
In [178]: df_MA.insert(loc = 9, column = 'for_duplicates', value = df_MA['PROPERTY_ID'].map(str) + df_MA['YEAR_REPORTED'].map(str
                    +df_MA['OWNER_NAME'] + df_MA['ADDRESS'] + df_MA['CITY'] + df_MA['ZIP'].map(str) + df_MA['PROPERTY_TYPE']
                    + df_MA['HOLDER_NAME'] + df_MA['UNCLAIMED_MONEY'].map(str))
df_MA
```

	PROPERTY_ID	YEAR_REPORTED	OWNER_NAME	ADDRESS	CITY	ZIP	PROPERTY_TYPE	HOLDER_NAME	UNCLAIMED_MONEY	for_duplicates
0	26212501	2020	MARYNIA E E	52 TROWBRIDGE ST APT 8	CAMBRIDGE	02138	Accounts Receivable Credit Balances	CITIBANK	3.48	262125012020MARYNIA E E52 TROWBRIDGE ST APT 8 ...
1	26212502	2020	APRIL M M	56 EASTLAND RD	JAMAICA PLAIN	02130	Accounts Receivable Credit Balances	CITIBANK	0.66	262125022020APRIL M M56 EASTLAND RD JAMAICA PL...
2	26212503	2020	KEVIN C C	43 BOLAS RD	DUXBURY	02332	Accounts Receivable Credit Balances	CITIBANK	0.88	262125032020KEVIN C C43 BOLAS RD DUXBURY02332A...
3	26212504	2020	SCOTT	13 PROSPECT ST	NEWBURYPORT	01950	Accounts Receivable Credit Balances	CITIBANK	91.56	262125042020SCOTT13 PROSPECT ST NEWBURYPORT0...
4	26212505	2020	BRIAN J J	7 STRONG ST APT 2	NEWBURYPORT	01950	Accounts Receivable Credit Balances	CITIBANK	30.45	262125052020BRIAN J J7 STRONG ST APT 2NEWBURYP...
...	...	...	...	...	...	...	...	...	...	...

```
In [179]: df_MA = df_MA.drop_duplicates(subset=['for_duplicates'])
df_MA
```

```
...
```

```
In [180]: #Dropping Duplicates from the dataframe
df_MA = df_MA.drop(['for_duplicates'], axis=1)
```

(25)

CA Dataset:

Dataset statistics

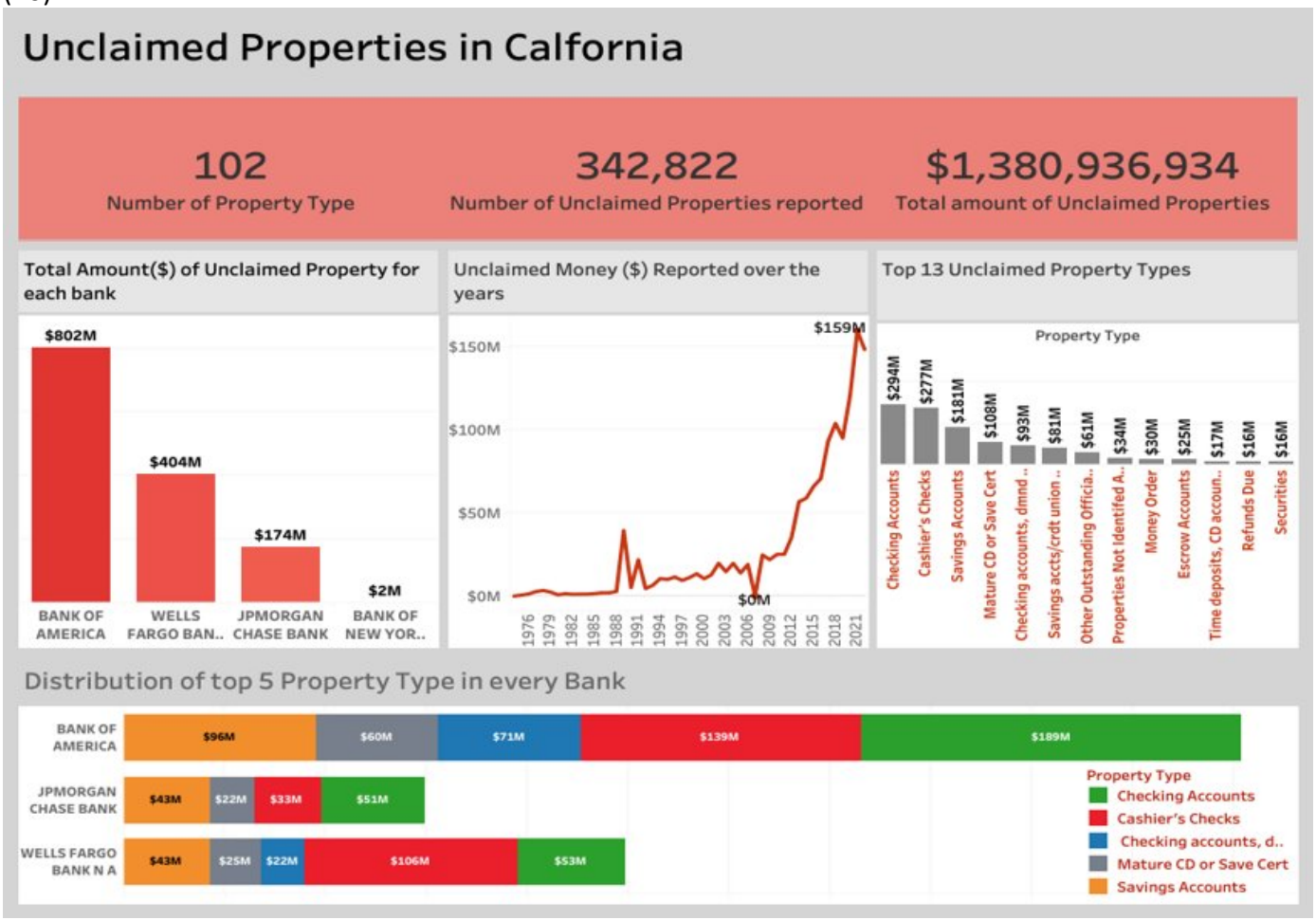
Number of variables	10
Number of observations	396058
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	30.2 MiB
Average record size in memory	80.0 B

MA Dataset:

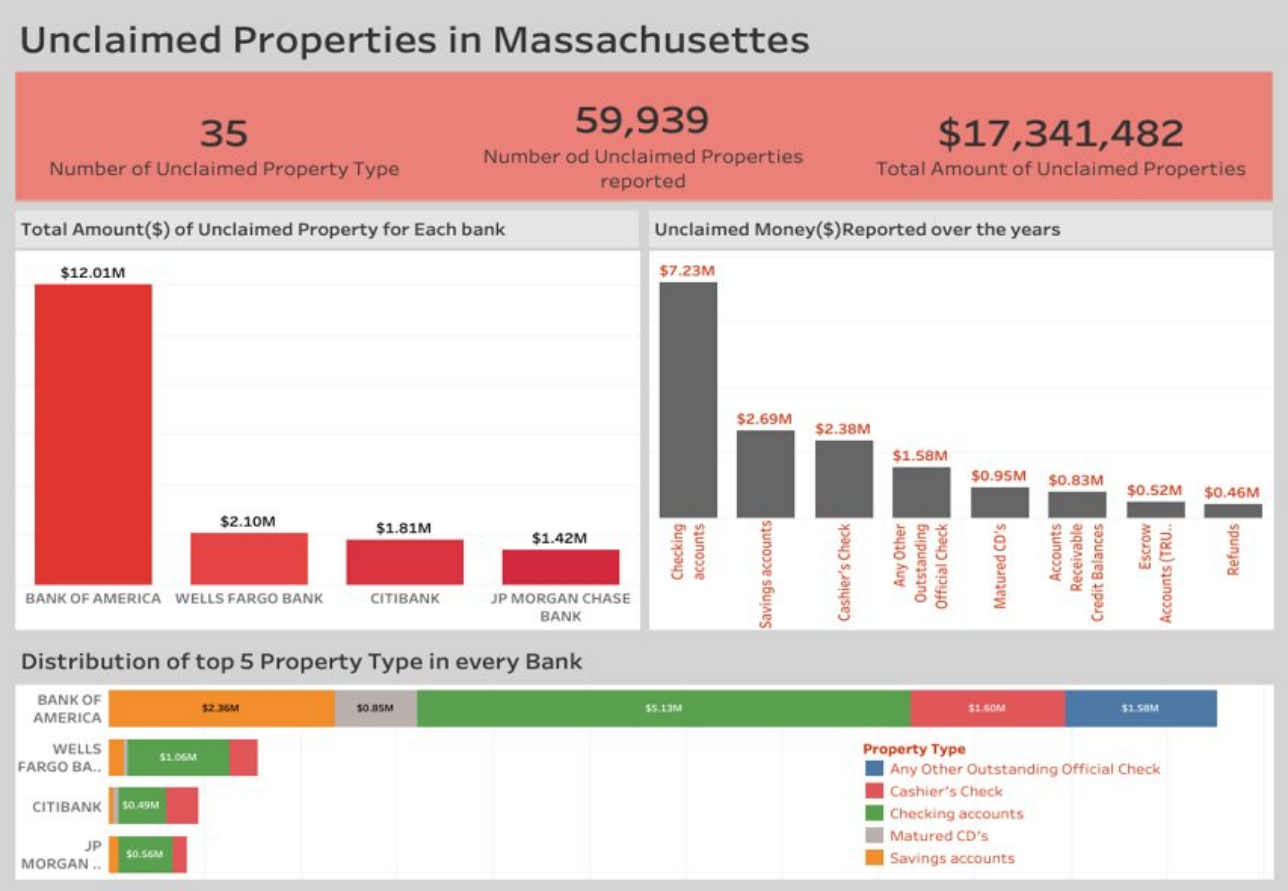
Dataset statistics

Number of variables	9
Number of observations	61854
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	4.2 MiB
Average record size in memory	72.0 B

(26)



(27)


Unclaimed Properties in Massachusettes
35
Number of Unclaimed Property Type
59,939
Number od Unclaimed Properties reported
$17,341,482
Total Amount of Unclaimed Properties
Total Amount($) of Unclaimed Property for Each bank
$12.01M
$2.10M
$1.81M
$1.42M
BANK OF AMERICA
WELLS FARGO BANK
CITIBANK
JP MORGAN CHASE BANK
Unclaimed Money($)Reported over the years
$7.23M
$2.69M
$2.38M
$1.58M
$0.95M
$0.83M
$0.52M
$0.46M
Checking accounts
Savings accounts
Cashier's Check
Any Other Outstanding Official Check
Matured CD's
Accounts Receivable Credit Balances
Escrow Accounts (TRU..
Refunds
Distribution of top 5 Property Type in every Bank
BANK OF AMERICA
WELLS FARGO BA..
CITIBANK
JP MORGAN ..
$2.36M
$0.85M
$5.13M
$1.60M
$1.58M
$1.06M
$0.49M
$0.56M
Property Type
Any Other Outstanding Official Check
Cashier's Check
Checking accounts
Matured CD's
Savings accounts