Task A Data Exploration and Auditing

A1. Dataset size

How many data instances and variables exist in the given dataset as indicated by the rows and columns?

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
data = pd.read_csv('monthly_smartcard_replacements.csv',encoding='utf-8')
# read  'monthly_smartcard_replacements.csv'
#A1
data.shape
#.shape function returns the rows and columns of the data
print('The data has rows number:',data.shape[0])
print('The data has columns number:',data.shape[1])
```

```
The data has rows number: 5792
The data has columns number: 5
```

A2.Missing values in the dataset

Are there any null values in the dataset? Report the number of null values in each column.

In [2]:

```
data.isna().sum()# total NA
```

Out[2]:

```
Month                    0
Transaction              0
Smartcard.Type           0
Action.Reason            0
Number.of.transactions   0
dtype: int64
```

No, there are no null values in the dataset

A3.Data Types

What are the different data types for each column?

In [3]:

```
data.dtypes # return the data type of each column.
```

Out[3]:

```
Month                       object
Transaction                 object
Smartcard.Type              object
Action.Reason               object
Number.of.transactions       int64
dtype: object
```

The first four columns are of type object and the fifth column is of type int.

A4. Convert Data Type

Convert data type of column ‘Month’ to a datetime format.

In [4]:

```
data['Month'] = pd.to_datetime(data['Month']) # Convert argument to datetime.
```

In [5]:

```
data.dtypes # data.dtypes # return the data type of each column.
```

Out[5]:

```
Month                       datetime64[ns]
Transaction                         object
Smartcard.Type                      object
Action.Reason                       object
Number.of.transactions               int64
dtype: object
```

The type of ‘Month’ column is datetime format.

A5. Descriptive Statistics

Calculate summary statistics for the Number.of.Transactions column. What does it tell you? Discuss at least two observations.

In [6]:

```
data.info() # Print a concise summary of a DataFrame.
data.describe() # Generate descriptive statistics.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5792 entries, 0 to 5791
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype         
---  ------                  --------------  -----         
 0   Month                   5792 non-null   datetime64[ns]
 1   Transaction             5792 non-null   object        
 2   Smartcard.Type          5792 non-null   object        
 3   Action.Reason           5792 non-null   object        
 4   Number.of.transactions  5792 non-null   int64         
dtypes: datetime64[ns](1), int64(1), object(3)
memory usage: 226.4+ KB
```

Out[6]:

	Number.of.transactions
count	5792.000000
mean	303.241540
std	845.056684
min	1.000000
25%	5.000000
50%	18.000000
75%	84.000000
max	9097.000000

In the data, there was at least one transaction per day, with 9,097 transactions on the most traded day. That's an average of 303 transactions per day.

A6. Exploring Smartcard Types

1. How many different (unique) smartcard types are recorded in the 'Smartcard.Type'column? What are those different smartcard types and how many instances recorded for each type

In [7]:

```
print('There are',data['Smartcard.Type'].nunique(),'different types')
name = data['Smartcard.Type'].unique() # Compute the ExtensionArray of unique values.
num = pd.value_counts(data['Smartcard.Type'],sort = False) #Return a Series or DataFrame containing
print(name[0],'number is',num[0])
print(name[1],'number is',num[1])
print(name[2],'number is',num[2])
print(name[3],'number is',num[3])
```

```
There are 4 different types
Photo Identification Card number is 1631
Driver Licence Card number is 1896
Industry Authority Card number is 1218
Marine Licence Ind Card number is 1047
```

2. What is the percentage of Driver Licence Card records as one of the smartcard types in 'Smartcard.Type' column?

In [8]:

```
percentage =  data[data['Smartcard.Type']=='Driver Licence Card'].shape[0]/data.shape[0] # .shape[0]
print('The of Driver Licence Card records as one of the smartcard types in  'Smartcard.Type'  is',pe
```

```
The of Driver Licence Card records as one of the smartcard types in  'Smartcard.Typ
e'  is 32.73480662983425 %
```

A7. Exploring Reasons for Smartcard Replacement

1. What are the different reasons for smartcard replacements in the given data and how many instances are observed for each reason? Hint: Check the 'Action.Reason' column.

In [9]:

```
pd.value_counts(data['Action.Reason'],sort = False)
# Return a Series or DataFrame containing counts of unique rows.
```

Out[9]:

```
Change Customer Details                  521
Destroyed                                379
Lost In Mail - Imu                       519
Managers Approval                        532
Disaster Relief                           48
Lost                                     539
Merged                                   200
Stolen                                   471
Damaged                                  342
Facial Image Is Not A True Likeness      304
Transition Laminate To Smartcard         256
Condition Change                         364
Expired                                  133
Product Exists Othr Surrend Void Cancel  321
Da/dgd Smartcard Replacement Fee Exempt  134
Faulty                                   344
Court Order Issued X3 Or X4 Condition    137
Marine Licence Transition                132
Defective                                 88
Remove Gender From Smartcard              28
Name: Action.Reason, dtype: int64
```

The left is the different causes, and the right is the number of times each cause occurs

2. What is the total number of months in which 100 or more smartcard replacements are reported due to being "Lost"?

In [10]:

```
temp = data[['Action.Reason','Smartcard.Type','Month','Number.of.transactions']]
temp = temp[temp['Action.Reason']=='Lost']
temp['month'] = temp['Month'].dt.month # The month of the datetime.
temp['year'] = temp['Month'].dt.year   # The year of the datetime.
temp = temp.groupby([ 'month','year'])['Number.of.transactions'].sum().reset_index()
# Groupby adds the numbers together for each month of the year, and reset_index resets the data
temp[temp['Number.of.transactions']>=100]
# Returns the amount of data greater than 100
```

Out[10]:

	month	year	Number.of.transactions
1	1	2012	5210
2	1	2013	4774
3	1	2014	4637
4	1	2015	4601
5	1	2016	4824
...	...	...	...
132	12	2017	4819
133	12	2018	5012
134	12	2019	5610
135	12	2020	6654
136	12	2021	7250

133 rows × 3 columns

In [11]:

```
print('There were ',temp[temp['Number.of.transactions']>=100].shape[0] ,' months in which 100 or mor
```

```
There were  133  months in which 100 or more pieces of smart caine were "lost" and r
eported stolen
```

Task B: Group Level Analysis and Visualisation

B1. Investigating Annual Smartcard Replacements

1. Create a new column named 'Year' extracting the year from the 'Month' column.

In [12]:

```
# B1
data['Year'] = data['Month'].dt.year # Returns the year of the date time
data
```

Out[12]:

	Month	Transaction	Smartcard.Type	Action.Reason	Number.of.transactions	Year
0	2019-03-01	Replace Smartcard	Photo Identification Card	Change Customer Details	156	2019
1	2019-03-01	Replace Smartcard	Driver Licence Card	Destroyed	110	2019
2	2019-03-01	Replace Smartcard	Industry Authority Card	Lost In Mail - Imu	48	2019
3	2019-03-01	Replace Smartcard	Marine Licence Ind Card	Managers Approval	8	2019
4	2019-03-01	Replace Smartcard	Marine Licence Ind Card	Lost In Mail - Imu	7	2019
...	...	...	...	...	...	...
5787	2020-11-01	Replace Smartcard	Photo Identification Card	Remove Gender From Smartcard	1	2020
5788	2020-12-01	Replace Smartcard	Marine Licence Ind Card	Stolen	1	2020
5789	2021-07-01	Replace Smartcard	Marine Licence Ind Card	Stolen	1	2021
5790	2021-07-01	Replace Smartcard	Photo Identification Card	Merged	1	2021
5791	2021-12-01	Replace Smartcard	Driver Licence Card	Transition Laminate To Smartcard	2	2021

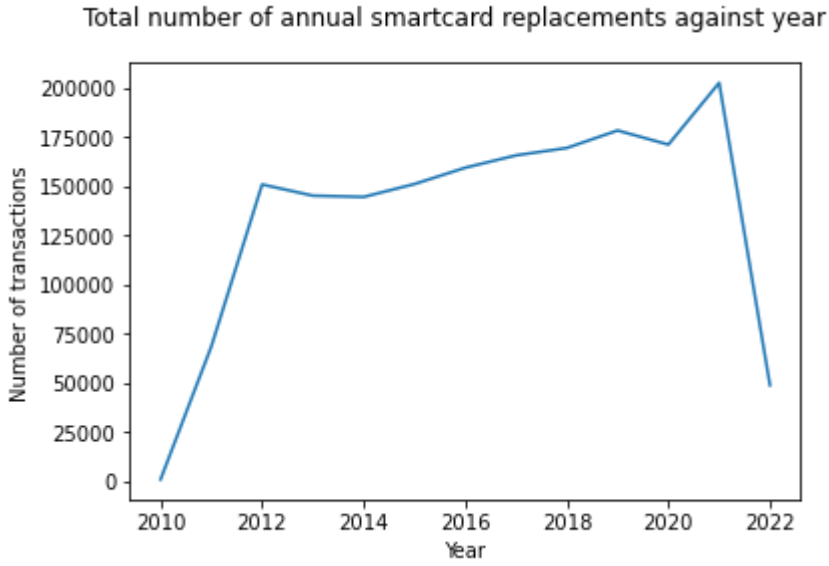
5792 rows × 6 columns

As you can see, a new row 'Year' is created on the far right of the data.

2. Create a line plot showing total number of annual smartcard replacements (number of transactions) against year.

In [13]:

```
data.groupby(['Year'])['Number.of.transactions'].sum().plot()
# Group The number of each year and add it together to draw a line chart
plt.xlabel('Year')
plt.ylabel('Number of transactions')
plt.suptitle('Total number of annual smartcard replacements against year')
plt.show()
```



3. Explain the trend as observed from the chart. Are there any years that are different than others and if so, what is the reason behind it?

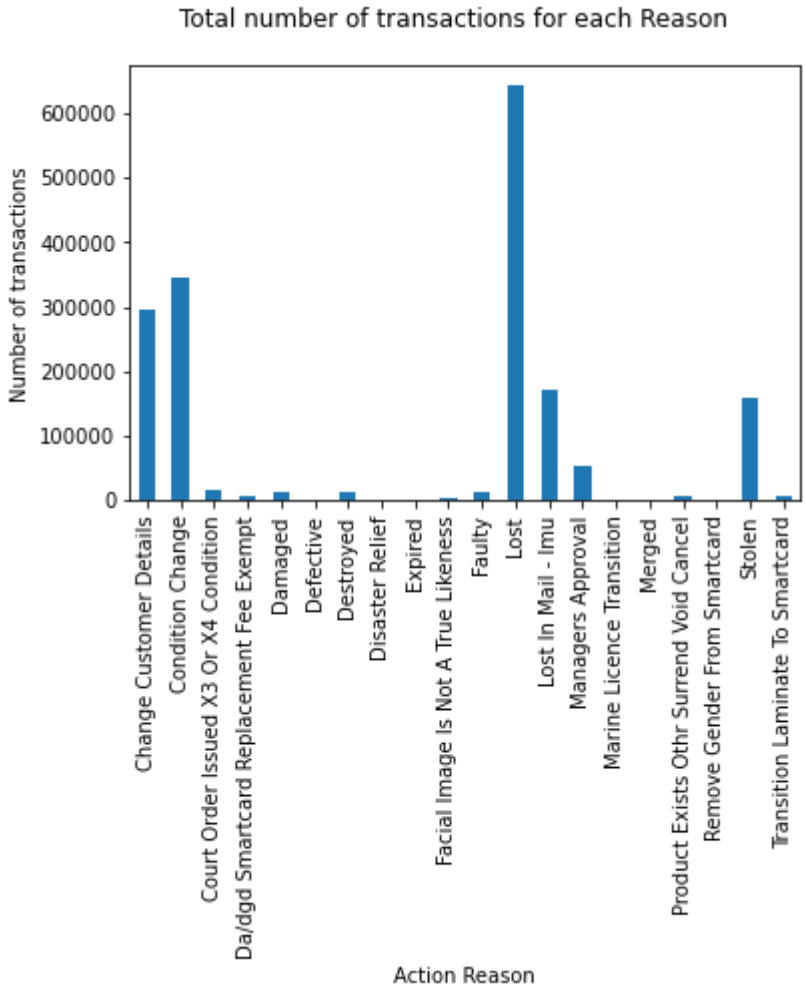
2010-2012 and 2021-2022 are significantly different from other years. In my opinion, in 2010-2012, smartphones were just launched and quickly became popular, so the number of smartphones increased rapidly. And in 2021-22, because of COVID-19, people weren't going out, so there was a big drop.

B2. Investigating Reasons for Smartcard Replacement

1. Create a barchart showing the total number of transactions for each 'Action.Reason' using the available data.

In [14]:

```
x = data.groupby(['Action.Reason'])['Number.of.transactions'].sum() # Group the quantities according
x.plot.bar() # Draw by bar chart
plt.xlabel('Action Reason')
plt.ylabel('Number of transactions')
plt.suptitle('Total number of transactions for each Reason')
plt.show()
```



2. What are the top three reasons for smartcard replacement?

In [15]:

```
x.sort_values(ascending=False).head(3)
# Reset the indexes and have them sorted in descending order, returning the first three
```

Out[15]:

```
Action.Reason
Lost                      642749
Condition Change          344905
Change Customer Details   294435
Name: Number.of.transactions, dtype: int64
```

The top three reasons are 'Lost','Condition Change' and 'Change Customer Details'.

3. Total number of transactions of which 'Action.Reason' is between 1000 and 2000?

In [16]:

```
temp = data.groupby(data['Action.Reason'])['Number.of.transactions'].sum()
# Categorize the data for each of the different reasons and add up the quantities
temp = pd.DataFrame(temp)# Make a list of each reason and their number
temp[(temp['Number.of.transactions']>=1000)&(temp['Number.of.transactions']<2000)]
# Look for data greater than 1000 and less than 2000
```

Out[16]:

Action.Reason	Number.of.transactions
Marine Licence Transition	1822

The number of transactions with a "Action.Reason" of between 1,000 and 2,000 was one('Marine Licence Transition').

B3. Investigating Reasons over Annual Smartcard Replacement

1. Find out the annual number of transactions for each 'Action.Reason' over different years that data is available.

In [17]:

```
temp = pd.DataFrame(data.groupby(['Action.Reason','Year'])['Number.of.transactions'].sum())
# Group by reason and year, and add their number to the list
temp
```

Out[17]:

		Number.of.transactions
Action.Reason	**Year**	
Change Customer Details	**2010**	84
	2011	12265
	2012	28446
	2013	28603
	2014	28188
...	**...**	...
Transition Laminate To Smartcard	**2018**	410
	2019	453
	2020	576
	2021	738
	2022	156

236 rows × 1 columns

On the left is each 'Action.Reason'. The right-hand side is the number of transactions per year for each reason.

2. For each action reason calculate the number of years that the number of annual transactions exceed 10000.

In [18]:

```
temp = pd.DataFrame(data.groupby(['Year','Action.Reason'])['Number.of.transactions'].sum()).reset_in
# Group by year and reason, add the quantities to the list and reset the index
temp = temp[temp['Number.of.transactions']>10000]
# List the number of transactions greater than 10000
x = pd.DataFrame(temp.groupby('Action.Reason')['Year'].count())
# Group each 'action.reason' to list the number of years in which the number of transactions exceeds
x
```

Out[18]:

Action.Reason	Year
Change Customer Details	11
Condition Change	11
Lost	12
Lost In Mail - Imu	8
Stolen	10

In the top table, on the left are the reasons why the number of trades exceeded 10,000 per year, and on the right are the number of years when they exceeded 10,000

3. Which action reasons have at least one year with the number of annual transactions exceeding 10000?

In [19]:

```
temp = pd.DataFrame(data.groupby(['Year','Action.Reason'])['Number.of.transactions'].sum()).reset_in
# Group by year and reason, add the quantities to the list and reset the index
x = temp[temp['Number.of.transactions']>10000]
# List the number of transactions greater than 10000
x['Action.Reason'].unique()
# Return every 'action.reason' that exceeds 10000 for at least one year
```

Out[19]:

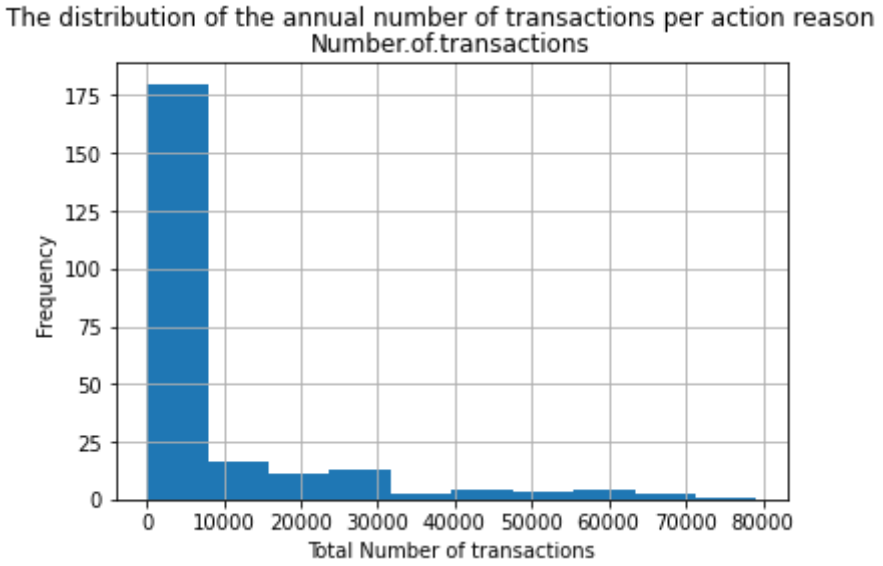
```
array(['Change Customer Details', 'Lost', 'Condition Change', 'Stolen',
       'Lost In Mail - Imu'], dtype=object)
```

The 'Action.Reason' is 'Change Customer Details', 'Lost', 'Condition Change', 'Stolen'and 'Lost In Mail - Imu' with the number of transactions exceeding 10000 for at least one year.

4. Create a histogram to analyze the distribution of the annual number of transactions per action reason as calculated in B3.1.

In [20]:

```
temp = pd.DataFrame(data.groupby(['Action.Reason','Year'])['Number.of.transactions'].sum())
# B3.1 Group by reason and year, and add their number to the list
temp.hist() # Plot the result as a histogram
plt.xlabel('Total Number of transactions')
plt.ylabel('Frequency')
plt.suptitle('The distribution of the annual number of transactions per action reason')
plt.show()
```



5. Explain any observations and comment on the distribution.

It is clear in the histogram that this is a left-leaning skew distribution, which means that the number of transactions per action reason occurs most frequently per year below 10,000. The higher the number, the fewer occurrences.