Assignment 2 Submission

1. Cleaning/Imputation Logic

Performing exclusions here. We are only keeping transactions with transaction type of P (Purchase) and Amount <= 3000000 to remove the outlier transaction that was incorrectly recorded in Mexican pesos and hence very large when compared to USD transactions. We know that it was in fact an actual transaction and not fraudulent.

```
In [6]: data = data[data['Transtype'] == 'P']
        data = data[data['Amount'] <= 3000000]
        data.shape

Out[6]: (96397, 10)
```

Here we first check the count of missing values in each column to further perform imputation on columns that have missing values (Merchnum, Merch state and Merch zip). In the next cell we store a copy of the cleaned data with the exclusions before we perform imputation and add variables as a reference point to go back to

```
In [7]: data.isna().sum()

Out[7]: Recnum                  0
        Cardnum                 0
        Date                    0
        Merchnum             3198
        Merch description       0
        Merch state          1020
        Merch zip            4300
        Transtype               0
        Amount                  0
        Fraud                   0
        dtype: int64

In [8]: data_orig = data.copy()
```

Explanation for merchant number

1. First we replace merchant number 0 with null values as it is highly unlikely that a merchant number would be 0.
2. We see that total null values are now 3251

3. Next we create a data dictionary mapping merchant descriptions to merchant numbers
4. We fill in the missing merchant numbers that have merchant descriptions that using the above dictionary
5. Null values are now 2094
6. Next we assign 'unknown' for transactions that have merchant description as 'Retail Credit Adjustment' and 'Retail Debit Adjustment' as these seem to be adjustment transactions with no merchant records
7. Null values are now 1403
8. Next we count the total number of unique merchant descriptions in the remaining null values, it's 508.
9. Then we create a new merchant number for each unique merchant description and add it to our data by mapping to merchant description, each new merchant number is max(merchnum) + 1
10. Our merchant numbers are all populated now with 0 null values

Explanation for Merchant State

1. Our total null values for Merchant state are 1020
2. Next we create a data dictionary mapping zip codes that exist in the data that have no merchant state assigned to their real world values
3. We create two more data dictionaries, mapping merchant numbers and merchant descriptions to their states.
4. We use the above 3 data dictionaries to impute the values of merchant states.
5. Next we assign 'unknown' for merchant states that have merchant description as 'Retail Credit Adjustment' and 'Retail Debit Adjustment' as these seem to be adjustment transactions with no merchant records
6. The null values are now 346.
7. Next, if we have states outside of U.S. we change their merchant state to 'foreign', this could be useful as foreign transactions could be fraudulent
8. Finally we impute all remaining null values with 'unknown'
9. Our merchant state is now all populated with 0 null values

Explanation for Merchant Zip

1. Our total null values for Merchant Zip are 4300
2. We create a data dictionary mapping merchant numbers to merchant zip codes
3. We create another data dictionary mapping merchant descriptions to merchant zip codes
4. We use the above dictionaries to map missing values of merchant zips using merchant number and descriptions
5. Our null values are now 2658
6. Next we assign 'unknown' for merchant zips that have merchant description as 'Retail Credit Adjustment' and 'Retail Debit Adjustment' as these seem to be adjustment transactions with no merchant records
7. We fill the remaining zip codes as unknown
8. Our zipcode is now completely populated with 0 null values

Explanation for Target encoded variable "Day of the week"

1. We create a categorical variable dow (day of the week) using the date record for each transaction
2. Next, to convert this variable to a numerical one using target encoding we use training data and remove the last 2 months of transactions (Out of time validation). This is to replicate a real life scenario where we use past data to train our model and then use it to make predictions on current or future transactions and this prevents overfitting.
3. We use a smoothing formula to target encode this variable (this also helps prevent overfitting).
4. In the end we plot a graph with days of the week against fraud rates with a baseline for average fraud rate for the population (average of the now numerical dow variable

Explanation for Target encoded variable "Month of the transaction" (Created by me)

1. I created a categorical variable month of the transaction using the date record for each transaction
2. Next, to convert this variable to a numerical one using target encoding I used a smoothing formula to target encode this variable (this helps prevent overfitting). Note that using OOT training data here was not suitable as values for month of november and december would not be calculated in that case
4. In the end we plot a graph with month of the transaction against fraud rates with a baseline for average fraud rate for the population (average of the now numerical month variable)

Explanation for Merchant state, Card Number, and Merchant number

1. Next we target encode merchant state using the smoothing formula for target encoding and plot the top 15 fraud merchant states against fraud rate for the population and plot baseline fraud average (using mean of the now numerical merchant state)
2. We calculate the target encoded values for card number using the smoothing formula for target encoding but we don't include it in our data as it overfits (in the plot we can see that the baseline is close to 0 and all values for card numbers lie above it). A possible reason for this could be that we don't have statistically significant samples in each category of card numbers to prevent overfitting and develop good smoothing values.
3. We calculate the target encoded values for merchant numbers using the smoothing formula for target encoding but we don't include it in our data as it overfits. A possible reason for this could be that we don't have statistically significant samples in each category of card numbers to prevent overfitting and develop good smoothing values.

2. Summary table of created variables

Description	#Variables_ Created
Day of the week: day of the week of the particular transaction	1
Day of the week target encoded: average fraud percentage of that day	1

Month: month of the particular transaction	1
Month target encoded: average fraud percentage of that month	1
State_risk target encoded: average fraud percentage of that state	1
Days Since: Number of days since a transaction with that entity was last seen	18
Velocity: [Number, average, maximum, median, total amount, actual amount/average, actual amount/max, actual amount/median, actual amount/ total amount] of transactions with the same entity over the last [0,1,3,7,14,30,60] days	1134
Relative Velocity: Number/amount of transactions with the same entities seen in the past [0,1] day divided by the number/amount of transactions with those same entities seen in the last [7,14,30,60] days	288
Velocity Density: Number of transactions with the same entities seen in the past [0,1] day divided by the number of days since a transaction with those same entities in the last [7,14,30,60] days	144
Transaction Amount Variability: Maximum, median, and mean of amount differences between current transaction and transaction seen [0,1,3,7,14,30] days ago while grouping transactions by each entity.	324
Counts by entity: Number of unique entities for a particular field over the last [1,3,7,14,30,60] days	1836
Relative Velocity (square divided): Number of transactions with the same entities seen in the past [0,1] day divided by the number of transactions with those same entities seen in the last [7,14,30,60] days. This result is further divided by the square of [7,14,30,60] days.	144
Amount category: Divide amount column into 5 equal sized bins and assign a label (1,5) to each bin	1
Foreign: Boolean field indicating if the merchant is located outside of the U.S. (True) or within the U.S. (False)	1

New variables created by me in the above table:
Month
Month target encoded

transactions clean make variables (1)

April 16, 2023

[3]:
```
import pandas as pd
import numpy as np
import datetime
import calendar
import timeit
import datetime as dt
import re
from math import exp
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
start_time = datetime.datetime.now()
```

[4]:
```
data = pd.read_csv('card transactions.csv')
data.shape
```

[4]:
```
(96753, 18)
```

[5]:
```
data.head()
```

[5]:
```
   Recnum     Cardnum    Date         Merchnum         Merch description  \
0       1  5142190439  1/1/10  5509006296254   FEDEX SHP 12/23/09 AB#
1       2  5142183973  1/1/10    61003026333  SERVICE MERCHANDISE #81
2       3  5142131721  1/1/10  4503082993600         OFFICE DEPOT #191
3       4  5142148452  1/1/10  5509006296254   FEDEX SHP 12/28/09 AB#
4       5  5142190439  1/1/10  5509006296254   FEDEX SHP 12/23/09 AB#

  Merch state  Merch zip Transtype  Amount  Fraud  Unnamed: 10  Unnamed: 11  \
0          TN    38118.0         P    3.62      0          NaN          NaN
1          MA     1803.0         P   31.42      0          NaN          NaN
2          MD    20706.0         P  178.49      0          NaN          NaN
3          TN    38118.0         P    3.62      0          NaN          NaN
4          TN    38118.0         P    3.62      0          NaN          NaN

   Unnamed: 12  Unnamed: 13  Unnamed: 14  Unnamed: 15  Unnamed: 16  \
0          NaN          NaN          NaN          NaN          NaN
1          NaN          NaN          NaN          NaN          NaN
```

```
2          NaN            NaN            NaN            NaN            NaN
3          NaN            NaN            NaN            NaN            NaN
4          NaN            NaN            NaN            NaN            NaN

   Unnamed: 17
0          NaN
1          NaN
2          NaN
3          NaN
4          NaN
```

[6]:
```
data.dropna(how='all', axis=1, inplace=True)
data['Date'] = pd.to_datetime(data['Date'])
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 96753 entries, 0 to 96752
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Recnum             96753 non-null  int64
 1   Cardnum            96753 non-null  int64
 2   Date               96753 non-null  datetime64[ns]
 3   Merchnum           93378 non-null  object
 4   Merch description  96753 non-null  object
 5   Merch state        95558 non-null  object
 6   Merch zip          92097 non-null  float64
 7   Transtype          96753 non-null  object
 8   Amount             96753 non-null  float64
 9   Fraud              96753 non-null  int64
dtypes: datetime64[ns](1), float64(2), int64(3), object(4)
memory usage: 7.4+ MB
```

[7]:
```
data['Transtype'].value_counts()
```

[7]:
```
P    96398
A      181
D      173
Y        1
Name: Transtype, dtype: int64
```

Performing exclusions here. We are only keeping transactions with transaction type of P (Purchase) and Amount <= 3000000 to remove the outlier transaction that was incorrectly recorded in mexican pesos and hence very large when compared to USD transactions. We know that it was in fact an actual transaction and not fraudulent

[8]:
```
data = data[data['Transtype'] == 'P']
data = data[data['Amount'] <= 3000000]
data.shape
```

[8]: (96397, 10)

Here we first check the count of missing values in each column to further perform imputation on columns that have missing values (Merchnum, Merch state and Merch zip). In the next cell we store a copy of the cleaned data before we perform imputation and add variables as a reference point to go back to

[9]:
```
data.isna().sum()
```

[9]:
```
Recnum                  0
Cardnum                 0
Date                    0
Merchnum             3198
Merch description       0
Merch state          1020
Merch zip            4300
Transtype               0
Amount                  0
Fraud                   0
dtype: int64
```

[10]:
```
data_orig = data.copy()
```

0.1 Clean and impute merchnum

Explanation for merchant number

1. First we replace merchant number 0 with null values as it is highly unlikely that a merchant number would be 0.
2. We see that total null values are now 3251
3. Next we create a data dictionary mapping merchant descriptions to merchant numbers
4. We fill in the missing merchant numbers that have merchant descriptions that using the above dictionary
5. Null values are now 2094
6. Next we assign 'unknown' for transactions that have merchant description as 'Retail Credit Adjustment' and 'Retail Debit Adjustment' as these seem to be adjustment transactions with no merchant records
7. Null values are now 1403
8. Next we count the total number of unique merchant descriptions in the remaining null values, it's 508.
9. Then we create a new merchant number for each unique merchant description and add it to our data by mapping to merchant description, each new merchant number is max(merchnum) + 1
10. Our merchant numbers are all populated now with 0 null values

[11]:
```python
data['Merchnum'] = data['Merchnum'].replace({'0':np.nan})
```

[12]:
```python
data['Merchnum'].isnull().sum()
```

[12]: 3251

[13]:
```python
merchdes_merchnum = {}
for index, merchdes in data[data['Merch description'].
 ↪notnull()][data['Merchnum'].notnull()]['Merch description'].items():
    if pd.isnull(merchdes) == True:
        continue
    elif merchdes not in merchdes_merchnum:
        merchdes_merchnum[merchdes] = data.loc[index, 'Merchnum']
```

[14]:
```python
# fill in by mapping with Merch description
data['Merchnum'] = data['Merchnum'].fillna(data['Merch description'].
 ↪map(merchdes_merchnum))
```

[15]:
```python
data['Merchnum'].isnull().sum()
```

[15]: 2094

[16]:
```python
# assign unknown for adjustments transactions
data['Merchnum'] = data['Merchnum'].mask(data['Merch description'] == 'RETAIL␣
 ↪CREDIT ADJUSTMENT', 'unknown')
data['Merchnum'] = data['Merchnum'].mask(data['Merch description'] == 'RETAIL␣
 ↪DEBIT ADJUSTMENT', 'unknown')
```

[17]:
```python
data['Merchnum'].isnull().sum()
```

[17]: 1403

[18]:
```python
data.loc[data.Merchnum.isna(), 'Merch description'].unique()[:20]
```

[18]:
```
array(['MONTGOMERY COLLEGE-PHONE', 'PACKAGE PLACE  THE',
       'CUBIX CORPORATION', 'SIGNAL GRAPHICS PRINTING',
       'C & M OFFICE EQUIPMENT', "TOMMY'S TRAILERS",
       'Z WORLD/RABBIT SEMICONDUC', 'IMPAC/TRI-COUNTY/FREED',
       'REPROGRPHC TECHNLGIES INC', 'STP SPECIALITY TECH',
       'VANGARD INTERNAITONAL', 'BLACKWELL SCIENCE', 'CDN ISOTOPES INC',
       'INTERACTIVE SOFTWARE S', 'H R WILLIAMS MILL SUPP',
       'ELSEVIER SCIENCE BV', 'COLORADO GARDEN SHOW',
       'PEARSON EDUCATION CANADA', 'PONTOTOC AREA VO-TECH',
       'NATIONAL BAG COMPANY'], dtype=object)
```

[19]:
```python
# 1403 NULL Merchnums with 508 unique Descriptions
data.loc[data.Merchnum.isna(), 'Merch description'].nunique()
```

[19]: 508

0.1.1 Create new Merchnums using the description field

[20]:
```
# adding new merchnums
# each new unique merchnum will be max(merchnum) + 1
merchnum_create = {}
max_merchnum = pd.to_numeric(data.Merchnum, errors='coerce').max()
for merch_desc in data.loc[data.Merchnum.isna(), 'Merch description'].unique():
    merchnum_create[merch_desc] = str(int(max_merchnum + 1))
    max_merchnum += 1
```

[21]:
```
# fill in by mapping with Merch description (newly created merchnums)
data['Merchnum'] = data['Merchnum'].fillna(data['Merch description'].
 ↪map(merchnum_create))
```

[22]:
```
for i in data.columns:
    print(i, data[i].isnull().sum())
```

```
Recnum 0
Cardnum 0
Date 0
Merchnum 0
Merch description 0
Merch state 1020
Merch zip 4300
Transtype 0
Amount 0
Fraud 0
```

0.2 Clean and impute State

Explanation for Merchant State

1. Our total null values for Merchant state are 1020
2. Next we create a data dictionary mapping zipcodes that exist in the data that have no merchant state assigned to their real world values
3. We create two more data dictionaries, mapping merchant numbers and merchant descriptions to their states.
4. We use the above 3 data dictionaries to impute the values of merchant states.
5. Next we assign 'unknown' for merchant states that have merchant description as 'Retail Credit Adjustment' and 'Retail Debit Adjustment' as these seem to be adjustment transactions with no merchant records
6. The null values are now 346.
7. Next, if we have states outside of U.S. we change their merchant state to 'foreign', this could be useful as foreign transactions could be fraudulent
8. Finally we impute all remaining null values with 'unknown'
9. Our merchant state is now all populated with 0 null values

[23]:
```
data['Merch state'].isnull().sum()
```

[23]: 1020

[24]:
```
data[(data['Merch state'].isnull()) & (data['Merch zip'].notnull())]['Merch␣
↪zip'].unique()
```

[24]:
```
array([9.2600e+02, 9.2900e+02, 1.4000e+03, 6.5132e+04, 8.6899e+04,
       2.3080e+04, 6.0528e+04, 9.3400e+02, 9.0200e+02, 7.3800e+02,
       9.0805e+04, 7.6302e+04, 9.0000e+00, 9.1400e+02, 6.0000e+00,
       9.5461e+04, 5.0823e+04, 2.0000e+00, 4.8700e+04, 6.8000e+02,
       1.0000e+00, 6.8100e+02, 6.2300e+02, 7.2600e+02, 9.3600e+02,
       1.2108e+04, 7.9100e+02, 9.0700e+02, 9.2200e+02, 9.2000e+02,
       3.0000e+00, 8.0100e+02, 8.0000e+00, 3.1040e+04, 3.8117e+04,
       4.1160e+04])
```

[25]:
```
# dict for mapping
zip_state = {}
for index, zip5 in data[data['Merch zip'].notnull()]['Merch zip'].items():
    if zip5 not in zip_state:
        zip_state[zip5] = data.loc[index, 'Merch state']

zip_state['00926'] = 'PR'
zip_state['00929'] = 'PR'
zip_state['00934'] = 'PR'
zip_state['00902'] = 'PR'
zip_state['00738'] = 'PR'
zip_state['90805'] = 'CA'
zip_state['76302'] = 'TX'
zip_state['00914'] = 'PR'
zip_state['95461'] = 'CA'
zip_state['00680'] = 'PR'
zip_state['00623'] = 'PR'
zip_state['00726'] = 'PR'
zip_state['00936'] = 'PR'
zip_state['12108'] = 'NY'
zip_state['00791'] = 'PR'
zip_state['00907'] = 'PR'
zip_state['00922'] = 'PR'
zip_state['00920'] = 'PR'
zip_state['00801'] = 'VI'
zip_state['31040'] = 'GA'
zip_state['41160'] = 'KY'
zip_state['00681'] = 'PR'
```

[26]:
```
merchnum_state = {}
for index, merchnum in data[data['Merchnum'].notnull()]['Merchnum'].items():
```

```
    if merchnum not in merchnum_state :
        merchnum_state [merchnum] = data.loc[index, 'Merch state']
```

[27]:
```
merchdes_state = {}
for index, merchdes in data[data['Merch description'].notnull()]['Merch␣
 ↪description'].items():
    if merchdes not in merchdes_state :
        merchdes_state [merchdes] = data.loc[index, 'Merch state']
```

[28]:
```
# fill in by mapping with zip, merchnum and merch description
data['Merch state'] = data['Merch state'].fillna(data['Merch zip'].
 ↪map(zip_state))
data['Merch state'] = data['Merch state'].fillna(data['Merchnum'].
 ↪map(merchnum_state))
data['Merch state'] = data['Merch state'].fillna(data['Merch description'].
 ↪map(merchdes_state))
```

[29]:
```
# assign unknown for adjustments transactions
data['Merch state'] = data['Merch state'].mask(data['Merch description'] ==␣
 ↪'RETAIL CREDIT ADJUSTMENT', 'unknown')
data['Merch state'] = data['Merch state'].mask(data['Merch description'] ==␣
 ↪'RETAIL DEBIT ADJUSTMENT', 'unknown')
```

[30]:
```
data['Merch state'].isnull().sum()
```

[30]: 346

[31]:
```
# change non-US states
# might actually be useful cus fraud could be foreign transactions
# maybe put a 'foreign' tag or just leave them as is

states = ["AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DC", "DE", "FL", "GA",
          "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA", "ME", "MD",
          "MA", "MI", "MN", "MS", "MO", "MT", "NE", "NV", "NH", "NJ",
          "NM", "NY", "NC", "ND", "OH", "OK", "OR", "PA", "RI", "SC",
          "SD", "TN", "TX", "UT", "VT", "VA", "WA", "WV", "WI", "WY",
          'VI', 'PR', np.nan, 'unknown']

for index, state in data['Merch state'].items():
    if state not in states:
        data.loc[index, 'Merch state'] = 'foreign'
```

[32]:
```
data['Merch state'].fillna('unknown',inplace=True)
```

[33]:
```
data['Merch state'].isnull().sum()
```

[33]: 0

0.3 Clean and impute zip

Explanation for Merchant Zip

1. Our total null values for Merchant Zip are 4300
2. We create a data dictionary mapping merchant numbers to merchant zip codes
3. We create another data dictionary mapping merchant desriptions to merchant zip codes
4. We use the above dictionaries to map missing values of merchant zips using merchant number and descriptions
5. Our null values are now 2658
6. Next we assign 'unknown' for merchant zips that have merchant description as 'Retail Credit Adjustment' and 'Retail Debit Adjustment' as these seem to be adjustment transactions with no merchant records
7. We fill the remaining zipcodes as unknown
8. Our zipcode is now completely populated with 0 null values

[34]:
```
data['Merch zip'].isnull().sum()
```

[34]: 4300

[35]:
```
merchnum_zip = {}
for index, merchnum in data[data['Merchnum'].notnull()]['Merchnum'].items():
    if merchnum not in merchnum_zip :
        merchnum_zip [merchnum] = data.loc[index, 'Merch zip']
```

[36]:
```
merchdes_zip = {}
for index, merchdes in data[data['Merch description'].notnull()]['Merch␣
↪description'].items():
    if merchdes not in merchdes_zip :
        merchdes_zip [merchdes] = data.loc[index, 'Merch zip']
```

[37]:
```
# fill in by mapping with merchnum and merch description
data['Merch zip'] = data['Merch zip'].fillna(data['Merchnum'].map(merchnum_zip))
data['Merch zip'] = data['Merch zip'].fillna(data['Merch description'].
↪map(merchdes_zip))
```

[38]:
```
data['Merch zip'].isnull().sum()
```

[38]: 2658

[39]:
```
# assign unknown for adjustments transactions
data['Merch zip'] = data['Merch zip'].mask(data['Merch zip'] == 'RETAIL CREDIT␣
↪ADJUSTMENT', 'unknown')
data['Merch zip'] = data['Merch zip'].mask(data['Merch zip'] == 'RETAIL DEBIT␣
↪ADJUSTMENT', 'unknown')
```

[40]:
```
data['Merch zip'].isnull().sum()
```

[40]: 2658

[41]:
```
temp = data[data['Merch zip'].isna()]
temp.head(50)
```

[41]:

	Recnum	Cardnum	Date	Merchnum	Merch description	\
51	52	5142204384	2010-01-02	5000006000095	IBM INTERNET 01000025	
54	55	5142146340	2010-01-02	5000006000095	IBM INTERNET 01000025	
55	56	5142260984	2010-01-02	5000006000095	IBM INTERNET 01000025	
58	59	5142204384	2010-01-02	5000006000095	IBM INTERNET 01000025	
59	60	5142204384	2010-01-02	5000006000095	IBM INTERNET 01000025	
60	61	5142204384	2010-01-02	5000006000095	IBM INTERNET 01000025	
61	62	5142204384	2010-01-02	5000006000095	IBM INTERNET 01000025	
62	63	5142253356	2010-01-02	5000006000095	IBM INTERNET 01000025	
64	65	5142204384	2010-01-02	5000006000095	IBM INTERNET 01000025	
65	66	5142204384	2010-01-02	5000006000095	IBM INTERNET 01000025	
66	67	5142260984	2010-01-02	5000006000095	IBM INTERNET 01000025	
68	69	5142260984	2010-01-02	5000006000095	IBM INTERNET 01000025	
69	70	5142260984	2010-01-02	5000006000095	IBM INTERNET 01000025	
71	72	5142204384	2010-01-02	5000006000095	IBM INTERNET 01000025	
72	73	5142204384	2010-01-02	5000006000095	IBM INTERNET 01000025	
75	76	5142253356	2010-01-02	5000006000095	IBM INTERNET 01000025	
77	78	5142204384	2010-01-02	5000006000095	IBM INTERNET 01000025	
78	79	5142149994	2010-01-02	5000006000095	IBM INTERNET 01000025	
79	80	5142153201	2010-01-02	5000006000095	IBM INTERNET 01000025	
87	88	5142255416	2010-01-03	8053478940091	MCGHEE & COMPANY INC	
199	200	5142257356	2010-01-03	2000049710067	LASER ACCESS 42760017	
218	219	5142172995	2010-01-03	6700046420068	MARYS GIFTS 41480013	
230	231	5142221571	2010-01-03	6005030600003	FORMA SCIENTIFIC	
258	259	5142171582	2010-01-04	4900000004673	WALGREEN 00004179	
262	263	5142257575	2010-01-04	unknown	RETAIL DEBIT ADJUSTMENT	
272	273	5142124791	2010-01-04	unknown	RETAIL DEBIT ADJUSTMENT	
293	294	5142171582	2010-01-04	4900000004673	WALGREEN 00004179	
379	380	5142183904	2010-01-04	1700000096481	AMES DEPT STOR 0021436	
400	401	5142276099	2010-01-04	unknown	RETAIL DEBIT ADJUSTMENT	
416	417	5142173617	2010-01-04	6100020004006	USGPO SUPT DOCS-SUB/PU	
476	477	5142267793	2010-01-05	unknown	RETAIL DEBIT ADJUSTMENT	
482	483	5142257356	2010-01-05	2000049710067	LASER ACCESS 42760017	
487	488	5142267793	2010-01-05	unknown	RETAIL CREDIT ADJUSTMENT	
515	516	5142267793	2010-01-05	unknown	RETAIL CREDIT ADJUSTMENT	
544	545	5142193730	2010-01-05	unknown	RETAIL CREDIT ADJUSTMENT	
557	558	5142234471	2010-01-05	unknown	RETAIL CREDIT ADJUSTMENT	
737	738	5142267793	2010-01-05	unknown	RETAIL CREDIT ADJUSTMENT	
739	740	5142267793	2010-01-05	unknown	RETAIL CREDIT ADJUSTMENT	
744	745	5142267793	2010-01-05	unknown	RETAIL CREDIT ADJUSTMENT	
758	759	5142267793	2010-01-05	unknown	RETAIL CREDIT ADJUSTMENT	
817	818	5142230669	2010-01-06	9996060597906	TOMMY'S TRAILERS	

852	853	5142174305	2010-01-06	unknown	RETAIL CREDIT ADJUSTMENT
868	869	5142205500	2010-01-06	5000006000095	IBM INTERNET 01000025
906	907	5142288897	2010-01-06	9996060597908	IMPAC/TRI-COUNTY/FREED
931	932	5142214551	2010-01-06	6176269	MUNKSGAARDS FORLAG
1040	1041	5142230181	2010-01-06	9996060597910	STP SPECIALITY TECH
1082	1083	5142132574	2010-01-06	9900020008506	UNICOR FED PRISON IND
1162	1163	5142179159	2010-01-07	674906173338	G B SCIENTIFIC
1174	1175	5142139011	2010-01-07	7300020006306	GENERAL SERVICES ADMIN
1195	1196	5142260689	2010-01-07	679960185332	BUSINESS WIRE

	Merch state	Merch zip	Transtype	Amount	Fraud
51	NY	NaN	P	20.15	0
54	NY	NaN	P	23.90	0
55	NY	NaN	P	19.95	0
58	NY	NaN	P	20.15	0
59	NY	NaN	P	20.15	0
60	NY	NaN	P	20.15	0
61	NY	NaN	P	20.15	0
62	NY	NaN	P	27.41	0
64	NY	NaN	P	20.15	0
65	NY	NaN	P	20.15	0
66	NY	NaN	P	19.95	0
68	NY	NaN	P	37.51	0
69	NY	NaN	P	19.95	0
71	NY	NaN	P	28.13	0
72	NY	NaN	P	20.15	0
75	NY	NaN	P	12.50	0
77	NY	NaN	P	20.15	0
78	NY	NaN	P	101.40	0
79	NY	NaN	P	19.95	0
87	WV	NaN	P	55.80	0
199	GA	NaN	P	1940.00	0
218	IL	NaN	P	17.97	0
230	OH	NaN	P	72.00	0
258	IL	NaN	P	21.28	0
262	unknown	NaN	P	320.00	0
272	unknown	NaN	P	970.00	0
293	IL	NaN	P	6.76	0
379	MA	NaN	P	13.00	0
400	unknown	NaN	P	82.59	0
416	DC	NaN	P	98.00	0
476	unknown	NaN	P	17.59	0
482	GA	NaN	P	600.00	0
487	unknown	NaN	P	19.69	0
515	unknown	NaN	P	17.59	0
544	unknown	NaN	P	105.00	0
557	unknown	NaN	P	1149.97	0

```
737       unknown         NaN          P     18.64      0
739       unknown         NaN          P     17.00      0
744       unknown         NaN          P     17.59      0
758       unknown         NaN          P     24.68      0
817            OK         NaN          P     48.97      0
852       unknown         NaN          P    379.42      0
868            NY         NaN          P     14.09      0
906            MD         NaN          P    467.58      0
931       unknown         NaN          P   1790.00      0
1040      foreign         NaN          P    486.18      0
1082           KY         NaN          P   1090.00      0
1162           CA         NaN          P    849.89      0
1174           MD         NaN          P    609.67      0
1195           CA         NaN          P    532.00      0
```

[42]:
```
data['Merch zip'].fillna('unknown', inplace=True)
data['Merch zip'].isnull().sum()
```

[42]: 0

[43]:
```
df = data.copy()
```

[44]:
```
data.to_csv('transactions_clean.csv')
```

[45]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 96397 entries, 0 to 96752
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Recnum             96397 non-null  int64
 1   Cardnum            96397 non-null  int64
 2   Date               96397 non-null  datetime64[ns]
 3   Merchnum           96397 non-null  object
 4   Merch description  96397 non-null  object
 5   Merch state        96397 non-null  object
 6   Merch zip          96397 non-null  object
 7   Transtype          96397 non-null  object
 8   Amount             96397 non-null  float64
 9   Fraud              96397 non-null  int64
dtypes: datetime64[ns](1), float64(1), int64(3), object(5)
memory usage: 10.1+ MB
```

0.4 Target encoded variables

[46]:
```
## to be safe, check the data type of dates first
df.Date = pd.to_datetime(df.Date)
df.Date.dtypes
## all good
```

[46]:
```
dtype('<M8[ns]')
```

Explanation for Target encoded variable "Day of the week"

1. We create a categorical variable dow (day of the week) using the date record for each transaction
2. Next, to convert this variable to a numerical one using target encoding we use training data and remove last 2 months of transactions (Out of time validation). This is to replicate a real life scenario where we use past data to train our model and then use it to make predictions on current or future transactions and this prevents overfitting.
3. We use smoothing formula to target encode this variable (this also helps prevent overfitting).
4. In the end we plot a graph with days of the week against fraud rates with a baseline for average fraud rate for the population (average of the now numerical dow variable

[47]:
```
## find the day of the week
df['Dow'] = df.Date.apply(lambda x: calendar.day_name[x.weekday()])
```

[48]:
```
## we want to not use the oot for target encoding variables
train_test = df[df.Date < '2010-11-01']
c = 4; nmid = 20; y_avg = train_test['Fraud'].mean()
y_dow = train_test.groupby('Dow')['Fraud'].mean()
num = train_test.groupby('Dow').size()
y_dow_smooth = y_avg + (y_dow - y_avg)/(1 + np.exp(-(num - nmid)/c))
df['Dow_Risk'] = df.Dow.map(y_dow_smooth)
```

[49]:
```
y_dow=y_dow.reset_index()
cats=['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday',␣
↪'Sunday']
y_dow['Dow']=pd.Categorical(y_dow['Dow'], categories=cats, ordered=True)
y_dow=y_dow.sort_values('Dow')
y_dow=y_dow.set_index('Dow')
y_dow
```

[49]:
```
              Fraud
Dow
Monday     0.008711
Tuesday    0.007127
Wednesday  0.009788
Thursday   0.018626
Friday     0.025994
```

```
Saturday    0.010095
Sunday      0.009630
```

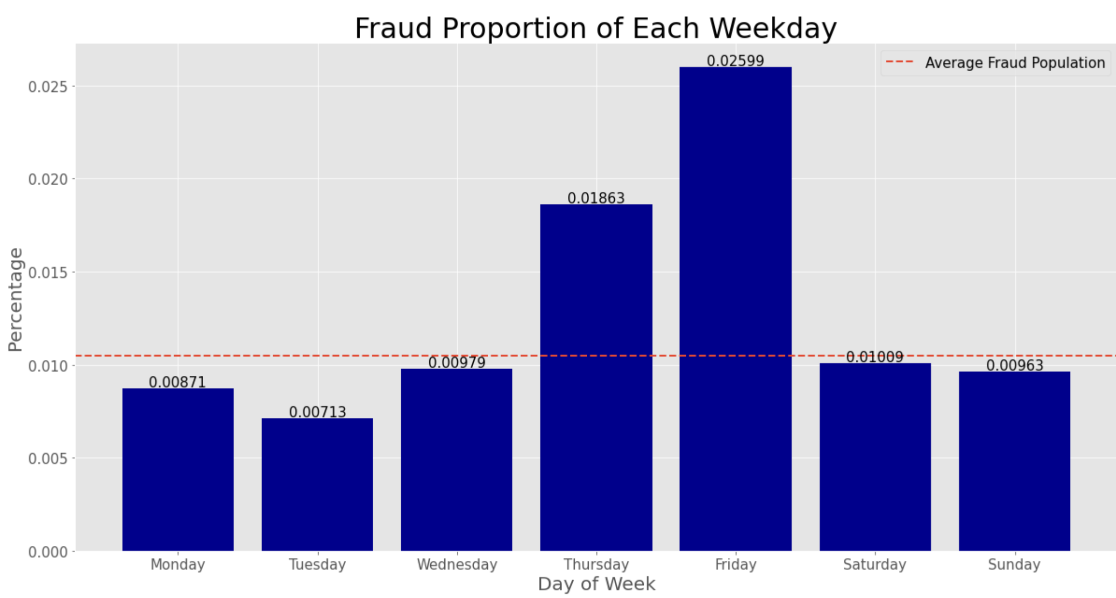
[50]:

```
plt.style.use('ggplot')
fig, ax = plt.subplots(figsize=(20,10))
plt.bar(data = y_dow,
        x = y_dow.index,
        height = 'Fraud',
        color = 'darkblue'
        )
#ax.set_ylim(bottom = 0.013)
ax.axhline(y = y_avg, ls = '--', lw = 2, label="Average Fraud Population")

for i, v in enumerate(y_dow.index):
    ax.text(v,y_dow.loc[v,'Fraud']+0.0001,round(y_dow.
 ↪loc[v,'Fraud'],5),horizontalalignment='center',fontsize=15)

plt.legend(['Average Fraud Population'], fontsize=15)
plt.xlabel("Day of Week",fontsize=20)
plt.ylabel("Percentage",fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.title("Fraud Proportion of Each Weekday", fontsize=30)

plt.show()
```



Explanation for Target encoded variable "Month of the transaction" (Created by me)

1. I created a categorical variable month of the transaciton using the date record for each transaction
2. Next, to convert this variable to a numerical one using target encoding I used a smoothing formula to target encode this variable (this helps prevent overfitting). Note that using OOT training data here was not suitable as values for month of november and december would not be calculated in that case
3. In the end we plot a graph with month of the transaction against fraud rates with a baseline for average fraud rate for the population (average of the now numerical month variable)

[51]:
```
## find the month of the transaction
df['Month'] = df.Date.apply(lambda x: datetime.datetime.strftime(x, '%B'))
```

[52]:
```
df['Month'].value_counts()
```

[52]:
```
August       10998
September     9821
March         9386
June          9206
May           8938
July          8271
February      7746
April         7700
January       6801
December      6642
November      5785
October       5103
Name: Month, dtype: int64
```

[53]:
```
c = 4; nmid = 20; y_avg_1 = df['Fraud'].mean()
y_month = df.groupby('Month')['Fraud'].mean()
num_1 = df.groupby('Month').size()
y_month_smooth = y_avg_1 + (y_month - y_avg_1)/(1 + np.exp(-(num_1 - nmid)/c))
df['Month_Risk'] = df.Month.map(y_month_smooth)
```

[54]:
```
y_month=y_month.reset_index()
cats=['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',␣
 ↪'September', 'October', 'November', 'December']
y_month['Month']=pd.Categorical(y_month['Month'], categories=cats, ordered=True)
y_month=y_month.sort_values('Month')
y_month=y_month.set_index('Month')
y_month
```

[54]:
```
              Fraud
Month
January    0.003235
February   0.002195
March      0.005753
```

```
April        0.008961
May          0.016335
June         0.007821
July         0.010881
August       0.021186
September    0.008859
October      0.017637
November     0.012792
December     0.015808
```

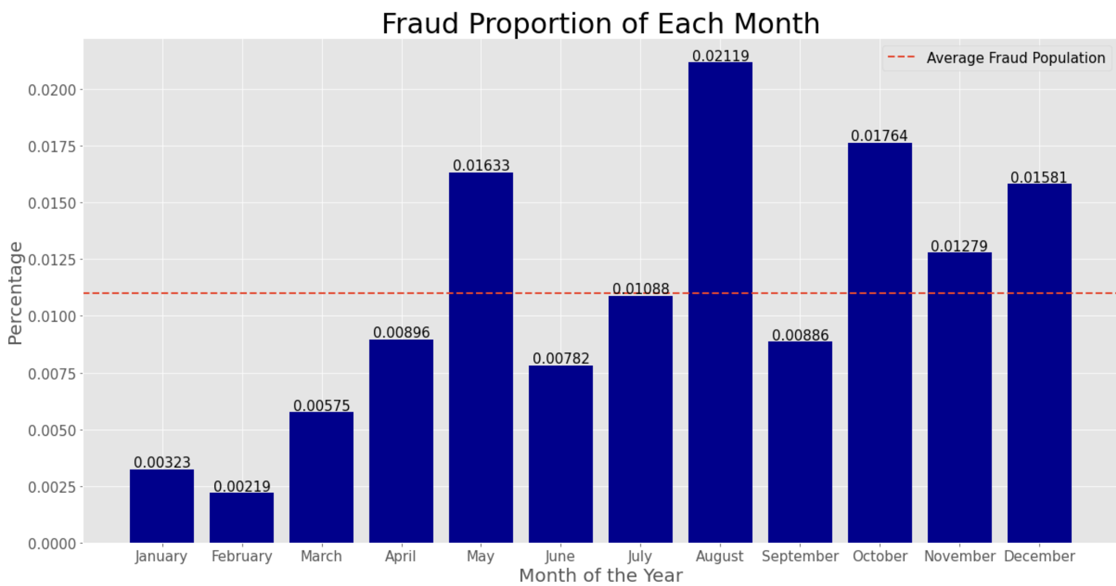
[55]:

```
plt.style.use('ggplot')
fig, ax = plt.subplots(figsize=(20,10))
plt.bar(data = y_month,
        x = y_month.index,
        height = 'Fraud',
        color = 'darkblue'
        )
#ax.set_ylim(bottom = 0.013)
ax.axhline(y = y_avg_1, ls = '--', lw = 2, label="Average Fraud Population")

for i, v in enumerate(y_month.index):
    ax.text(v,y_month.loc[v,'Fraud']+0.0001,round(y_month.
 ↪loc[v,'Fraud'],5),horizontalalignment='center',fontsize=15)

plt.legend(['Average Fraud Population'], fontsize=15)
plt.xlabel("Month of the Year",fontsize=20)
plt.ylabel("Percentage",fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.title("Fraud Proportion of Each Month", fontsize=30)

plt.show()
```



[56]:
```
df.head()
```

[56]:
```
   Recnum     Cardnum        Date       Merchnum       Merch description  \
0       1  5142190439  2010-01-01  5509006296254   FEDEX SHP 12/23/09 AB#
1       2  5142183973  2010-01-01    61003026333  SERVICE MERCHANDISE #81
2       3  5142131721  2010-01-01  4503082993600        OFFICE DEPOT #191
3       4  5142148452  2010-01-01  5509006296254   FEDEX SHP 12/28/09 AB#
4       5  5142190439  2010-01-01  5509006296254   FEDEX SHP 12/23/09 AB#

  Merch state Merch zip Transtype  Amount  Fraud     Dow  Dow_Risk    Month  \
0          TN   38118.0         P    3.62      0  Friday  0.025994  January
1          MA    1803.0         P   31.42      0  Friday  0.025994  January
2          MD   20706.0         P  178.49      0  Friday  0.025994  January
3          TN   38118.0         P    3.62      0  Friday  0.025994  January
4          TN   38118.0         P    3.62      0  Friday  0.025994  January

   Month_Risk
0    0.003235
1    0.003235
2    0.003235
3    0.003235
4    0.003235
```

Explanation for Merchant state, Card Number, and Merchant number

1. Next we target encode merchant state using the smoothing formula for target encoding and plot the top 15 fraud merchant states against fraud rate for the population and plot baseline fraud average (using mean of the now numerical merchant state)

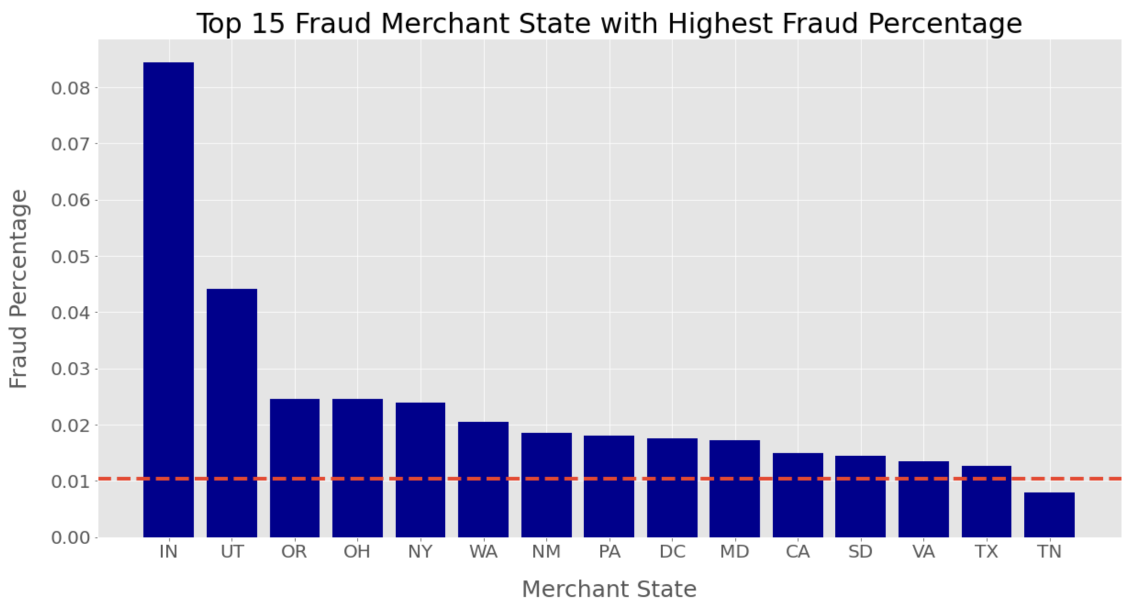
2. We calculate the target encoded values for card number using the smoothing formula for target encoding but we don't include it in our data as it overfits (in the plot we can see that the baseline is close to 0 and all values for card numbers lie above it). A possible reason for this could be that we don't have statistically significant samples in each category of card numbers to prevent overfitting and develop good smoothing values.
3. We calculate the target encoded values for merchant number using the smoothing formula for target encoding but we don't include it in our data as it overfits. A possible reason for this could be that we don't have statistically significant samples in each category of card numbers to prevent overfitting and develop good smoothing values.

[57]:
```
# statistical smoothing
c = 4
nmid = 20
y_avg = train_test['Fraud'].mean()
y_state = train_test.groupby('Merch state')['Fraud'].mean()
num = train_test.groupby('Merch state').size()
y_state_smooth = y_avg + (y_state - y_avg)/(1 + np.exp(-(num-nmid)/c))
df['state_risk'] = df['Merch state'].map(y_state_smooth)
top15_states = pd.DataFrame(y_state.sort_values(ascending=False).head(15))
plt.style.use('ggplot')
fig, ax = plt.subplots(figsize=(20,10))
plt.bar(data=top15_states, x=top15_states.index, height='Fraud',␣
 ↪color='darkblue')

plt.title('Top 15 Fraud Merchant State with Highest Fraud Percentage',␣
 ↪fontsize=30)

plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.xlabel('Merchant State',fontsize=25, labelpad=20)
plt.ylabel('Fraud Percentage',fontsize=25, labelpad=20)

ax.axhline(y=y_avg, lw = 4, ls='--')
plt.show()
```



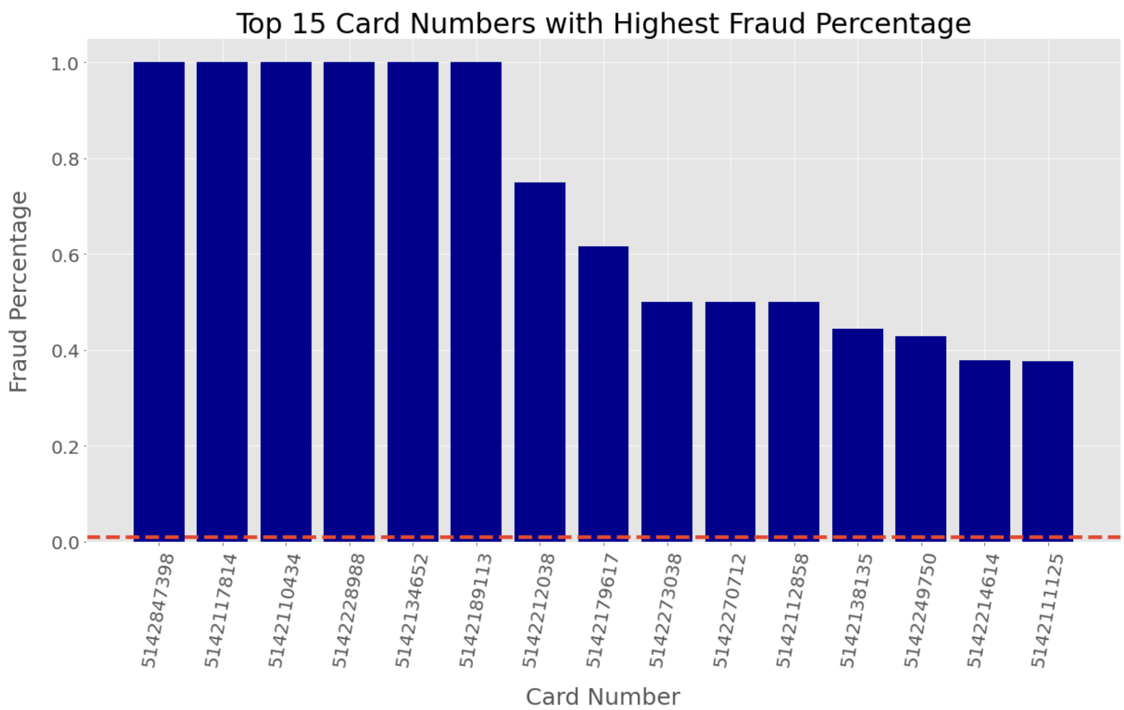
[58]:
```
# statistical smoothing
c = 4
nmid = 20
y_avg = train_test['Fraud'].mean()
y_cardnum = train_test.groupby('Cardnum')['Fraud'].mean()
num = train_test.groupby('Cardnum').size()
y_cardnum_smooth = y_avg + (y_cardnum - y_avg)/(1 + np.exp(-(num-nmid)/c))

# comment this out so we don't include this variable because it overfits
# df['cardnum_risk'] = df['Cardnum'].map(y_cardnum_smooth)
top15_cardnum = pd.DataFrame(y_cardnum\
                             .sort_values(ascending=False).head(15))
plt.style.use('ggplot')
fig, ax = plt.subplots(figsize=(20,10))
plt.bar(data=top15_cardnum, x=top15_cardnum.index.astype(str), height='Fraud',␣
 ↪color='darkblue')
plt.title('Top 15 Card Numbers with Highest Fraud Percentage', fontsize=30)

plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.xlabel('Card Number',fontsize=25, labelpad=20)
plt.ylabel('Fraud Percentage',fontsize=25, labelpad=20)
plt.xticks(rotation = 80)

ax.axhline(y=y_avg, lw = 4, ls='--')

plt.show()
```



[59]:
```
# statistical smoothing
c = 4
nmid = 20
y_avg = train_test['Fraud'].mean()
y_merchnum = train_test.groupby('Merchnum')['Fraud'].mean()
num = train_test.groupby('Merchnum').size()
y_merchnum_smooth = y_avg + (y_merchnum - y_avg)/(1 + np.exp(-(num-nmid)/c))

# comment this out so we don't include this variable because it overfits
# data['merchnum_risk'] = data['Merchnum'].map(y_merchnum_smooth)
top15_merchnum = pd.DataFrame(y_merchnum\
                             .sort_values(ascending=False).head(15))
top15_merchnum.head(20)

# plt.style.use('ggplot')
# fig, ax = plt.subplots(figsize=(20,10))
# plt.bar(data=top15_merchnum, x=top15_merchnum.index.astype(str),␣
 ↪height='Fraud', color='darkblue')
# plt.title('Top 15 Merchant Numbers with Highest Fraud Percentage',␣
 ↪fontsize=30)

# plt.xticks(fontsize=20)
# plt.yticks(fontsize=20)
# plt.xlabel('Merchant Number',fontsize=25, labelpad=20)
```

```
# plt.ylabel('Fraud Percentage',fontsize=25, labelpad=20)
# plt.xticks(rotation = 80)

# ax.axhline(y=y_avg, lw = 4, ls='--')

# plt.show()
```

[59]:

```
                     Fraud
Merchnum
450730006NOTO     1.000000
6006333528866     1.000000
4503738417400     1.000000
600660007477      1.000000
19908503337       1.000000
7000330100777     1.000000
8834000695423     1.000000
7593860080752     1.000000
92891948003       1.000000
6070095870009     0.931034
938909877224      0.780488
8292309000040     0.689655
6929              0.666667
6005030600003     0.615385
3831009006589     0.500000
```

0.5 Other variables

0.5.1 2.1. Data types

[61]:

```
df['Cardnum'] = df['Cardnum'].apply(str)
df['Merchnum'] = df['Merchnum'].apply(str)
df['Merch zip'] = df['Merch zip'].apply(str)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 96397 entries, 0 to 96752
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Recnum             96397 non-null  int64
 1   Cardnum            96397 non-null  object
 2   Date               96397 non-null  datetime64[ns]
 3   Merchnum           96397 non-null  object
 4   Merch description  96397 non-null  object
 5   Merch state        96397 non-null  object
 6   Merch zip          96397 non-null  object
 7   Transtype          96397 non-null  object
```

```
 8   Amount             96397 non-null  float64
 9   Fraud              96397 non-null  int64
 10  Dow                96397 non-null  object
 11  Dow_Risk           96397 non-null  float64
 12  Month              96397 non-null  object
 13  Month_Risk         96397 non-null  float64
 14  state_risk         96397 non-null  float64
dtypes: datetime64[ns](1), float64(4), int64(2), object(8)
memory usage: 13.8+ MB
```

[62]:
```
### add leading 0 to zips
### note: there are some zips that are state abbrv. as we imputed them ealier,␣
↪so pandas read the column as str

def leading_0(x):

    if '.0' in x:
        x = x[:-2]
        if len(x) == 5:
            return x
        else:
            return '0'*(5-len(x)) + x
    else:
        return '0'*(5-len(x)) + x

# df['Merch zip'] = df['Merch zip'].apply(leading_0)
```

[63]:
```
### delete white spaces in merch description
df['Merch description'] = df['Merch description'].str.replace(r'\s', '')
```

0.5.2 Create entities

[64]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 96397 entries, 0 to 96752
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Recnum             96397 non-null  int64
 1   Cardnum            96397 non-null  object
 2   Date               96397 non-null  datetime64[ns]
 3   Merchnum           96397 non-null  object
 4   Merch description  96397 non-null  object
 5   Merch state        96397 non-null  object
 6   Merch zip          96397 non-null  object
 7   Transtype          96397 non-null  object
```

```
 8   Amount             96397 non-null  float64
 9   Fraud              96397 non-null  int64
 10  Dow                96397 non-null  object
 11  Dow_Risk           96397 non-null  float64
 12  Month              96397 non-null  object
 13  Month_Risk         96397 non-null  float64
 14  state_risk         96397 non-null  float64
dtypes: datetime64[ns](1), float64(4), int64(2), object(8)
memory usage: 13.8+ MB
```

[65]:

```
df['card_merch'] = df['Cardnum'] + df['Merchnum']
df['card_zip'] = df['Cardnum'] + df['Merch zip']
df['card_state'] = df['Cardnum'] + df['Merch state']
df['merch_zip'] = df['Merchnum'] + df['Merch zip']
df['merch_state'] = df['Merchnum'] + df['Merch state']
df['state_des'] = df['Merch state'] + df['Merch description']

# these next entity take a long time to calculate the variables for, and I␣
↪don't know why
# df['state_zip'] = df['Merch state'] + df['Merch zip']

df['zip3'] = df['Merch zip'].str[:3]
df['card_zip3'] = df.Cardnum + df['zip3']
# df['merchnum_zip'] = df.Merchnum + df['Merch zip']
# df['merchnum_zip3'] = df.Merchnum + df['zip3']
df['Card_Merchdesc'] = df['Cardnum'] + df['Merch description']
df['Card_dow'] = df['Cardnum'] + df['Dow']
df['Merchnum_desc'] = df['Merchnum'] + df['Merch description']
df['Merchnum_dow'] = df['Merchnum'] + df['Dow']
# df['Merchdesc_State'] = df['Merch description'] + df['Merch state']
# df['Merchdesc_Zip'] = df['Merch description'] + df['Merch zip']
df['Merchdesc_dow'] = df['Merch description'] + df['Dow']
df['Card_Merchnum_desc'] = df['Cardnum'] + df['Merchnum'] + df['Merch␣
↪description']
# df['Card_Merchnum_State'] = df['Cardnum'] + df['Merchnum'] + df['Merch state']
df['Card_Merchnum_Zip'] = df['Cardnum'] + df['Merchnum'] + df['Merch zip']
# df['Card_Merchdesc_State'] = df['Cardnum'] + df['Merch description'] +␣
↪df['Merch state']
df['Card_Merchdesc_Zip'] = df['Cardnum'] + df['Merch description'] + df['Merch␣
↪zip']
df['Merchnum_desc_State'] = df['Merchnum'] + df['Merch description'] +␣
↪df['Merch state']
# df['Merchnum_desc_Zip'] = df['Merchnum'] + df['Merch description'] +␣
↪df['Merch zip']
```

[66]:

```
df.columns
```

[66]:
```
Index(['Recnum', 'Cardnum', 'Date', 'Merchnum', 'Merch description',
       'Merch state', 'Merch zip', 'Transtype', 'Amount', 'Fraud', 'Dow',
       'Dow_Risk', 'Month', 'Month_Risk', 'state_risk', 'card_merch',
       'card_zip', 'card_state', 'merch_zip', 'merch_state', 'state_des',
       'zip3', 'card_zip3', 'Card_Merchdesc', 'Card_dow', 'Merchnum_desc',
       'Merchnum_dow', 'Merchdesc_dow', 'Card_Merchnum_desc',
       'Card_Merchnum_Zip', 'Card_Merchdesc_Zip', 'Merchnum_desc_State'],
      dtype='object')
```

[67]:
```
entities = list(df.iloc[:, np.r_[1, 3, 12:len(df.columns)]].columns)
```

[68]:
```
entities
```

[68]:
```
['Cardnum',
 'Merchnum',
 'Month',
 'Month_Risk',
 'state_risk',
 'card_merch',
 'card_zip',
 'card_state',
 'merch_zip',
 'merch_state',
 'state_des',
 'zip3',
 'card_zip3',
 'Card_Merchdesc',
 'Card_dow',
 'Merchnum_desc',
 'Merchnum_dow',
 'Merchdesc_dow',
 'Card_Merchnum_desc',
 'Card_Merchnum_Zip',
 'Card_Merchdesc_Zip',
 'Merchnum_desc_State']
```

0.5.3 Variables

[69]:
```
df.Date = pd.to_datetime(df.Date)
df1 = df.copy()
final = df.copy()
df1['check_date'] = df1.Date
df1['check_record'] = df1.Recnum
```

0.6 Make the Benford's law top 40 tables and variables

[70]:
```python
# another way to get the first digit
bf = data.copy()
bf['amount_100'] = (bf['Amount'] * 100).astype(str)
bf['first_digit'] = bf['amount_100'].str[0]
bf['first_digit'].value_counts()
```

[70]:
```
1    26603
3    18670
2    16178
4     8278
5     6955
6     6017
7     5027
8     4534
9     4135
Name: first_digit, dtype: int64
```

[71]:
```python
dropfedex = bf[bf['Merch description'].str.contains('FEDEX')]
dropfedex.head()
```

[71]:
```
   Recnum     Cardnum       Date      Merchnum       Merch description  \
0       1  5142190439 2010-01-01  5509006296254  FEDEX SHP 12/23/09 AB#
3       4  5142148452 2010-01-01  5509006296254  FEDEX SHP 12/28/09 AB#
4       5  5142190439 2010-01-01  5509006296254  FEDEX SHP 12/23/09 AB#
5       6  5142149874 2010-01-01  5509006296254  FEDEX SHP 12/22/09 AB#
6       7  5142189277 2010-01-01  5509006296254  FEDEX SHP 12/28/09 AB#

  Merch state Merch zip Transtype  Amount  Fraud amount_100 first_digit
0          TN   38118.0         P    3.62      0      362.0           3
3          TN   38118.0         P    3.62      0      362.0           3
4          TN   38118.0         P    3.62      0      362.0           3
5          TN   38118.0         P    3.67      0      367.0           3
6          TN   38118.0         P    3.62      0      362.0           3
```

[72]:
```python
droplist = dropfedex.index.tolist()
droplist[:10]
```

[72]:
```
[0, 3, 4, 5, 6, 9, 10, 11, 12, 15]
```

[73]:
```python
droplist[-10:]
```

[73]:
```
[96246, 96291, 96292, 96319, 96397, 96415, 96426, 96433, 96459, 96727]
```

[74]:
```python
len(droplist)
```

[74]: 11775

[75]:
```
bf.head()
```

[75]:
```
   Recnum     Cardnum       Date       Merchnum        Merch description  \
0       1  5142190439 2010-01-01  5509006296254   FEDEX SHP 12/23/09 AB#
1       2  5142183973 2010-01-01    61003026333  SERVICE MERCHANDISE #81
2       3  5142131721 2010-01-01  4503082993600        OFFICE DEPOT #191
3       4  5142148452 2010-01-01  5509006296254   FEDEX SHP 12/28/09 AB#
4       5  5142190439 2010-01-01  5509006296254   FEDEX SHP 12/23/09 AB#

  Merch state Merch zip Transtype  Amount  Fraud amount_100 first_digit
0          TN   38118.0         P    3.62      0      362.0           3
1          MA    1803.0         P   31.42      0     3142.0           3
2          MD   20706.0         P  178.49      0    17849.0           1
3          TN   38118.0         P    3.62      0      362.0           3
4          TN   38118.0         P    3.62      0      362.0           3
```

[76]:
```
bf.shape
```

[76]: (96397, 12)

[77]:
```
bf1 = bf.drop(droplist)
bf1.shape
```

[77]: (84622, 12)

[78]:
```
# datefilter = datetime.datetime(2010,11,1)
# bf1 = bf1[bf1['Date'] < datefilter]
# bf1.shape
```

[79]:
```
bf1['bin']= bf1['first_digit'].apply(lambda x: "low" if x == "1" else ("low" if␣
 ↪x == "2" else "high"))
bf1.head(5)
```

[79]:
```
    Recnum     Cardnum       Date       Merchnum        Merch description  \
1        2  5142183973 2010-01-01    61003026333  SERVICE MERCHANDISE #81
2        3  5142131721 2010-01-01  4503082993600        OFFICE DEPOT #191
7        8  5142191182 2010-01-01  6098208200062    MIAMI COMPUTER SUPPLY
8        9  5142258629 2010-01-01   602608969534           FISHER SCI ATL
13      14  5142124791 2010-01-01  5725000466504      CDW*GOVERNMENT INC

   Merch state Merch zip Transtype  Amount  Fraud amount_100 first_digit   bin
1           MA    1803.0         P   31.42      0     3142.0           3  high
2           MD   20706.0         P  178.49      0    17849.0           1   low
7           OH   45429.0         P  230.32      0    23032.0           2   low
8           GA   30091.0         P   62.11      0     6211.0           6  high
```

```
13           IL    60061.0          P   106.89       0     10689.0              1    low
```

[80]:
```
bf1['first_digit'].value_counts()
```

[80]:
```
1    25697
2    15827
3    10297
4     7686
5     6749
6     5469
7     4699
8     4152
9     4046
Name: first_digit, dtype: int64
```

[81]:
```
# calculating n_low and n_high
card_bf =  bf1.groupby(['Cardnum','bin']).agg({'bin': ['count']}).reset_index()
card_bf.columns=['Cardnum', 'bin', 'count']
card_bf
```

[81]:
```
         Cardnum   bin  count
0     5142110002   low      1
1     5142110081  high      4
2     5142110313  high      1
3     5142110313   low      2
4     5142110402  high      8
…            …    …     …
3128  5142310598   low      2
3129  5142310768  high      2
3130  5142310768   low      2
3131  5142847398  high     35
3132  5142847398   low     10

[3133 rows x 3 columns]
```

[82]:
```
card_bf =  card_bf.
 ↪pivot_table(index='Cardnum',columns='bin',values='count',aggfunc='sum').
 ↪reset_index()
card_bf.columns=['Cardnum', 'n_high', 'n_low']
card_bf
```

[82]:
```
         Cardnum  n_high  n_low
0     5142110002     NaN    1.0
1     5142110081     4.0    NaN
2     5142110313     1.0    2.0
3     5142110402     8.0    3.0
4     5142110434     NaN    1.0
```

```
…              …       …      …
1635   5142310397     1.0    NaN
1636   5142310525     3.0    1.0
1637   5142310598     NaN    2.0
1638   5142310768     2.0    2.0
1639   5142847398    35.0   10.0

[1640 rows x 3 columns]
```

[83]:

```python
# if either n_low or n_high is zero, set it to 1
card_bf = card_bf.fillna(1)
card_bf
```

[83]:

```
         Cardnum  n_high  n_low
0     5142110002     1.0    1.0
1     5142110081     4.0    1.0
2     5142110313     1.0    2.0
3     5142110402     8.0    3.0
4     5142110434     1.0    1.0
…            …       …      …
1635  5142310397     1.0    1.0
1636  5142310525     3.0    1.0
1637  5142310598     1.0    2.0
1638  5142310768     2.0    2.0
1639  5142847398    35.0   10.0

[1640 rows x 3 columns]
```

[84]:

```python
# calclating R, 1/R, U, n, t U_smoothed
c=3
n_mid=15
card_bf['R'] = (1.096 * card_bf['n_low']/card_bf['n_high'])
card_bf['1/R'] = (1/card_bf['R'])
card_bf['U'] = list(map(lambda x, y : max(x,y),card_bf['R'],card_bf['1/R']))
card_bf['n'] = card_bf['n_high'] + card_bf['n_low']
card_bf['t'] = ((card_bf['n']-n_mid)/c)
card_bf['U_smoothed']= list(map(lambda x, y : (1 + (x-1)/
↪(1+exp(-y))),card_bf['U'],card_bf['t']))
```

[85]:

```python
top40_card_bf = card_bf.sort_values(['U_smoothed'], ascending = False).head(40).
↪reset_index(drop = True)
top40_card_bf.head(40)
```

[85]:

```
      Cardnum  n_high  n_low          R       1/R          U      n  \
0  5142253356     5.0   61.0  13.371200  0.074788  13.371200   66.0
1  5142299705     3.0   25.0   9.133333  0.109489   9.133333   28.0
2  5142197563   134.0   15.0   0.122687  8.150852   8.150852  149.0
```

3	5142194617	33.0	5.0	0.166061	6.021898	6.021898	38.0
4	5142288241	13.0	1.0	0.084308	11.861314	11.861314	14.0
5	5142239140	3.0	16.0	5.845333	0.171077	5.845333	19.0
6	5142144931	30.0	6.0	0.219200	4.562044	4.562044	36.0
7	5142192606	2.0	13.0	7.124000	0.140371	7.124000	15.0
8	5142204384	54.0	199.0	4.038963	0.247588	4.038963	253.0
9	5142284940	6.0	21.0	3.836000	0.260688	3.836000	27.0
10	5142189113	24.0	6.0	0.274000	3.649635	3.649635	30.0
11	5142225308	17.0	4.0	0.257882	3.877737	3.877737	21.0
12	5142116864	18.0	58.0	3.531556	0.283161	3.531556	76.0
13	5142293257	13.0	2.0	0.168615	5.930657	5.930657	15.0
14	5142173286	13.0	2.0	0.168615	5.930657	5.930657	15.0
15	5142246929	25.0	79.0	3.463360	0.288737	3.463360	104.0
16	5142224699	25.0	7.0	0.306880	3.258603	3.258603	32.0
17	5142847398	35.0	10.0	0.313143	3.193431	3.193431	45.0
18	5142273608	21.0	6.0	0.313143	3.193431	3.193431	27.0
19	5142147267	76.0	22.0	0.317263	3.151958	3.151958	98.0
20	5142224769	5.0	15.0	3.288000	0.304136	3.288000	20.0
21	5142242241	51.0	16.0	0.343843	2.908303	2.908303	67.0
22	5142260984	101.0	265.0	2.875644	0.347748	2.875644	366.0
23	5142113192	12.0	2.0	0.182667	5.474453	5.474453	14.0
24	5142191416	7.0	18.0	2.818286	0.354826	2.818286	25.0
25	5142194228	2.0	11.0	6.028000	0.165893	6.028000	13.0
26	5142308889	2.0	11.0	6.028000	0.165893	6.028000	13.0
27	5142212038	3.0	12.0	4.384000	0.228102	4.384000	15.0
28	5142195887	3.0	12.0	4.384000	0.228102	4.384000	15.0
29	5142225184	11.0	27.0	2.690182	0.371722	2.690182	38.0
30	5142257356	58.0	142.0	2.683310	0.372674	2.683310	200.0
31	5142216493	5.0	14.0	3.068800	0.325860	3.068800	19.0
32	5142239106	23.0	8.0	0.381217	2.623175	2.623175	31.0
33	5142144593	14.0	4.0	0.313143	3.193431	3.193431	18.0
34	5142126842	16.0	38.0	2.603000	0.384172	2.603000	54.0
35	5142117315	20.0	7.0	0.383600	2.606882	2.606882	27.0
36	5142218798	9.0	21.0	2.557333	0.391032	2.557333	30.0
37	5142180432	25.0	58.0	2.542720	0.393280	2.542720	83.0
38	5142264155	12.0	27.0	2.466000	0.405515	2.466000	39.0
39	5142294614	15.0	5.0	0.365333	2.737226	2.737226	20.0

	t	U_smoothed
0	17.000000	13.371199
1	4.333333	9.027976
2	44.666667	8.150852
3	7.666667	6.019548
4	-0.333333	5.533836
5	1.333333	4.834555
6	7.000000	4.558799
7	0.000000	4.062000

```
8     79.333333    4.038963
9      4.000000    3.784991
10     5.000000    3.631901
11     2.000000    3.534703
12    20.333333    3.531556
13     0.000000    3.465328
14     0.000000    3.465328
15    29.666667    3.463360
16     5.666667    3.250816
17    10.000000    3.193331
18     4.000000    3.153979
19    27.666667    3.151958
20     1.666667    2.924507
21    17.333333    2.908303
22   117.000000    2.875644
23    -0.333333    2.867770
24     3.333333    2.755655
25    -0.666667    2.705717
26    -0.666667    2.705717
27     0.000000    2.692000
28     0.000000    2.692000
29     7.666667    2.689391
30    61.666667    2.683310
31     1.333333    2.637231
32     5.333333    2.615376
33     1.000000    2.603526
34    13.000000    2.602996
35     4.000000    2.577980
36     5.000000    2.546910
37    22.666667    2.542720
38     8.000000    2.465508
39     1.666667    2.461235
```

[86]:

```
# calculating n_low and n_high
merch_bf =  bf1.groupby(['Merchnum','bin']).agg({'bin': ['count']}).
 ↪reset_index()
merch_bf.columns=['Merchnum', 'bin', 'count']
merch_bf =  merch_bf.
 ↪pivot_table(index='Merchnum',columns='bin',values='count',aggfunc='sum').
 ↪reset_index()
merch_bf.columns=['Merchnum', 'n_high', 'n_low']
merch_bf.head()
```

[86]:

```
        Merchnum  n_high  n_low
0  003100006NOT6     1.0    NaN
1  004740006ABC6     NaN    1.0
2  005590006PNB6     1.0    NaN
```

```
3  014430619  14      NaN     1.0
4  014938913  51      1.0     NaN
```

[87]:
```
# if either n_low or n_high is zero, set it to 1
merch_bf = merch_bf.fillna(1)
merch_bf
```

[87]:
```
             Merchnum  n_high  n_low
0      003100006NOT6     1.0    1.0
1      004740006ABC6     1.0    1.0
2      005590006PNB6     1.0    1.0
3      014430619   14     1.0    1.0
4      014938913   51     1.0    1.0
…                 …      …      …
13586  DU49038320006     1.0    1.0
13587     JCPENNE9 CO     2.0    1.0
13588  PENNE9 CO #05     1.0    1.0
13589  PENNE9 CO #68     1.0    1.0
13590        unknown   417.0  274.0

[13591 rows x 3 columns]
```

[88]:
```
# calclating R, 1/R, U, n, t U_smoothed
merch_bf['R'] = (1.096 * merch_bf['n_low']/merch_bf['n_high'])
merch_bf['1/R'] = (1/merch_bf['R'])
merch_bf['U'] = list(map(lambda x, y : max(x,y),merch_bf['R'],merch_bf['1/R']))
merch_bf['n'] = merch_bf['n_high'] + merch_bf['n_low']
merch_bf['t'] = ((merch_bf['n']-n_mid)/c)
merch_bf['U_smoothed']= list(map(lambda x, y : (1 + (x-1)/
  ↪(1+exp(-y))),merch_bf['U'],merch_bf['t']))
```

[89]:
```
top40_merch_bf = merch_bf.sort_values(['U_smoothed'], ascending = False).
  ↪head(40).reset_index(drop = True)
top40_merch_bf.head(40)
```

[89]:
```
         Merchnum  n_high  n_low          R         1/R           U      n  \
0    991808369338   181.0    1.0   0.006055  165.145985  165.145985  182.0
1   8078200641472     1.0   59.0  64.664000    0.015465   64.664000   60.0
2    308904389335    53.0    1.0   0.020679   48.357664   48.357664   54.0
3   3523000628102     1.0   34.0  37.264000    0.026836   37.264000   35.0
4    808998385332    36.0    1.0   0.030444   32.846715   32.846715   37.0
5        55158027     1.0   27.0  29.592000    0.033793   29.592000   28.0
6   8916500620062    31.0    1.0   0.035355   28.284672   28.284672   32.0
7   3910694900001     1.0   25.0  27.400000    0.036496   27.400000   26.0
8       881145544     1.0   24.0  26.304000    0.038017   26.304000   25.0
9      8889817332     1.0   24.0  26.304000    0.038017   26.304000   25.0
10  5600900060992    27.0    1.0   0.040593   24.635036   24.635036   28.0
```

11	6844000608436	1.0	23.0	25.208000	0.039670	25.208000	24.0
12	92891948003	24.0	1.0	0.045667	21.897810	21.897810	25.0
13	5803301245621	1.0	21.0	23.016000	0.043448	23.016000	22.0
14	3433000017263	3.0	53.0	19.362667	0.051646	19.362667	56.0
15	467615916337	22.0	1.0	0.049818	20.072993	20.072993	23.0
16	817004638227	1.0	19.0	20.824000	0.048022	20.824000	20.0
17	2376700063599	2.0	30.0	16.440000	0.060827	16.440000	32.0
18	993620816222	19.0	1.0	0.057684	17.335766	17.335766	20.0
19	993620810220	76.0	5.0	0.072105	13.868613	13.868613	81.0
20	465614140337	18.0	1.0	0.060889	16.423358	16.423358	19.0
21	8999000079657	18.0	1.0	0.060889	16.423358	16.423358	19.0
22	8317600900099	2.0	24.0	13.152000	0.076034	13.152000	26.0
23	5000006000095	23.0	253.0	12.056000	0.082946	12.056000	276.0
24	9420966064460	17.0	1.0	0.064471	15.510949	15.510949	18.0
25	5186264200136	17.0	1.0	0.064471	15.510949	15.510949	18.0
26	600000201284	50.0	4.0	0.087680	11.405109	11.405109	54.0
27	5600000060302	16.0	1.0	0.068500	14.598540	14.598540	17.0
28	7080606900600	16.0	1.0	0.068500	14.598540	14.598540	17.0
29	6070095870009	3.0	26.0	9.498667	0.105278	9.498667	29.0
30	999960264339	28.0	3.0	0.117429	8.515815	8.515815	31.0
31	555400670006	15.0	1.0	0.073067	13.686131	13.686131	16.0
32	881894855	15.0	1.0	0.073067	13.686131	13.686131	16.0
33	1960400470068	3.0	23.0	8.402667	0.119010	8.402667	26.0
34	993620559229	43.0	5.0	0.127442	7.846715	7.846715	48.0
35	2586000448258	14.0	1.0	0.078286	12.773723	12.773723	15.0
36	8100544800098	14.0	1.0	0.078286	12.773723	12.773723	15.0
37	604901367333	14.0	1.0	0.078286	12.773723	12.773723	15.0
38	6000330043193	1.0	13.0	14.248000	0.070185	14.248000	14.0
39	6005300190068	1.0	13.0	14.248000	0.070185	14.248000	14.0

	t	U_smoothed
0	55.666667	165.145985
1	15.000000	64.663981
2	13.000000	48.357557
3	6.666667	37.217908
4	7.333333	32.825921
5	4.333333	29.221627
6	5.666667	28.190609
7	3.666667	26.741995
8	3.333333	25.432399
9	3.333333	25.432399
10	4.333333	24.328875
11	3.000000	24.059914
12	3.333333	21.177981
13	2.333333	21.069793
14	13.666667	19.362645
15	2.666667	18.833836

```
16   1.666667   17.674579
17   5.666667   16.386771
18   1.666667   14.740518
19  22.000000   13.868613
20   1.333333   13.205914
21   1.333333   13.205914
22   3.666667   12.849118
23  87.000000   12.056000
24   1.000000   11.608354
25   1.000000   11.608354
26  13.000000   11.405086
27   0.666667    9.985322
28   0.666667    9.985322
29   4.666667    9.419493
30   5.333333    8.479703
31   0.333333    8.390562
32   0.333333    8.390562
33   3.666667    8.218159
34  11.000000    7.846601
35   0.000000    6.886861
36   0.000000    6.886861
37   0.000000    6.886861
38  -0.333333    6.530110
39  -0.333333    6.530110
```

[90]:

```
# Here are the tables for the Benford's law. They would be useful for a␣
 ↪forensic analysis
top40_card_bf.to_csv('Benford top cards.csv')
top40_merch_bf.to_csv('Benford top merchs.csv')
```

[91]:

```
card_bf['Cardnum'] = card_bf['Cardnum'].apply(str)
merch_bf['Merchnum'] = merch_bf['Merchnum'].apply(str)
card_bf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1640 entries, 0 to 1639
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Cardnum     1640 non-null   object
 1   n_high      1640 non-null   float64
 2   n_low       1640 non-null   float64
 3   R           1640 non-null   float64
 4   1/R         1640 non-null   float64
 5   U           1640 non-null   float64
 6   n           1640 non-null   float64
 7   t           1640 non-null   float64
```

```
 8   U_smoothed  1640 non-null   float64
dtypes: float64(8), object(1)
memory usage: 115.4+ KB
```

[92]:
```
card_bf.set_index('Cardnum',inplace=True)
```

[93]:
```
card_Ustar = pd.DataFrame(card_bf['U_smoothed'])
card_Ustar.sort_values(['U_smoothed'], ascending = False).head(10)
```

[93]:
```
            U_smoothed
Cardnum
5142253356   13.371199
5142299705    9.027976
5142197563    8.150852
5142194617    6.019548
5142288241    5.533836
5142239140    4.834555
5142144931    4.558799
5142192606    4.062000
5142204384    4.038963
5142284940    3.784991
```

[94]:
```
merch_bf.set_index('Merchnum',inplace=True)
```

[95]:
```
merch_Ustar = pd.DataFrame(merch_bf['U_smoothed'])
merch_Ustar.sort_values(['U_smoothed'], ascending = False).head(10)
```

[95]:
```
               U_smoothed
Merchnum
991808369338   165.145985
8078200641472   64.663981
308904389335    48.357557
3523000628102   37.217908
808998385332    32.825921
55158027        29.221627
8916500620062   28.190609
3910694900001   26.741995
881145544       25.432399
8889817332      25.432399
```

[96]:
```
final = final.merge(card_Ustar, how =␣
 ↪'left',left_on='Cardnum',right_on=card_Ustar.index)
final = final.rename(columns={'U_smoothed':'U*_cardnum'})
final = final.merge(merch_Ustar, how =␣
 ↪'left',left_on='Merchnum',right_on=merch_Ustar.index)
final = final.rename(columns={'U_smoothed':'U*_merchnum'})
```

[97]:
```
final
```

[97]:
```
       Recnum     Cardnum       Date       Merchnum     Merch description  \
0           1  5142190439 2010-01-01  5509006296254    FEDEXSHP12/23/09AB#
1           2  5142183973 2010-01-01    61003026333  SERVICEMERCHANDISE#81
2           3  5142131721 2010-01-01  4503082993600        OFFICEDEPOT#191
3           4  5142148452 2010-01-01  5509006296254    FEDEXSHP12/28/09AB#
4           5  5142190439 2010-01-01  5509006296254    FEDEXSHP12/23/09AB#
…         …         …          …            …                      …
96392   96749  5142276053 2010-12-31  3500000006160        BESTBUY00001610
96393   96750  5142225701 2010-12-31  8090710030950  MARKUSOFFICESUPPLIES
96394   96751  5142226486 2010-12-31  4503057341100           TECHPAC,INC
96395   96752  5142244619 2010-12-31  8834000695412               BUY.COM
96396   96753  5142243247 2010-12-31  9108347680006   STAPLESNATIONAL#471

      Merch state Merch zip Transtype   Amount  Fraud  …          Card_dow  \
0              TN   38118.0         P     3.62      0  …  5142190439Friday
1              MA    1803.0         P    31.42      0  …  5142183973Friday
2              MD   20706.0         P   178.49      0  …  5142131721Friday
3              TN   38118.0         P     3.62      0  …  5142148452Friday
4              TN   38118.0         P     3.62      0  …  5142190439Friday
…             …         …         …        …    …  …                …
96392          KY   41042.0         P    84.79      0  …  5142276053Friday
96393          OH   45248.0         P   118.75      0  …  5142225701Friday
96394          OH   45150.0         P   363.56      0  …  5142226486Friday
96395          CA   92656.0         P  2202.03      0  …  5142244619Friday
96396          NJ    7606.0         P   554.64      0  …  5142243247Friday

                             Merchnum_desc         Merchnum_dow  \
0         5509006296254FEDEXSHP12/23/09AB#  5509006296254Friday
1         61003026333SERVICEMERCHANDISE#81    61003026333Friday
2             4503082993600OFFICEDEPOT#191  4503082993600Friday
3         5509006296254FEDEXSHP12/28/09AB#  5509006296254Friday
4         5509006296254FEDEXSHP12/23/09AB#  5509006296254Friday
…                                      …                  …
96392         3500000006160BESTBUY00001610  3500000006160Friday
96393  8090710030950MARKUSOFFICESUPPLIES  8090710030950Friday
96394             4503057341100TECHPAC,INC  4503057341100Friday
96395                 8834000695412BUY.COM  8834000695412Friday
96396     9108347680006STAPLESNATIONAL#471  9108347680006Friday

                    Merchdesc_dow  \
0         FEDEXSHP12/23/09AB#Friday
1       SERVICEMERCHANDISE#81Friday
2             OFFICEDEPOT#191Friday
3         FEDEXSHP12/28/09AB#Friday
4         FEDEXSHP12/23/09AB#Friday
```

```
…                            …
96392       BESTBUY00001610Friday
96393   MARKUSOFFICESUPPLIESFriday
96394            TECHPAC,INCFriday
96395                BUY.COMFriday
96396    STAPLESNATIONAL#471Friday

                                Card_Merchnum_desc  \
0        51421904395509006296254FEDEXSHP12/23/09AB#
1        514218397361003026333SERVICEMERCHANDISE#81
2            514213172145030829936000FFICEDEPOT#191
3        51421484525509006296254FEDEXSHP12/28/09AB#
4        51421904395509006296254FEDEXSHP12/23/09AB#
…                                              …
96392        514227605335000000006160BESTBUY00001610
96393   514222570180907100309500MARKUSOFFICESUPPLIES
96394            51422264864503057341100TECHPAC,INC
96395                51422446198834000695412BUY.COM
96396    514224324791083476800006STAPLESNATIONAL#471

                  Card_Merchnum_Zip                       Card_Merchdesc_Zip  \
0       5142190439550900629625438118.0   5142190439FEDEXSHP12/23/09AB#38118.0
1         5142183973610030263331803.0   5142183973SERVICEMERCHANDISE#811803.0
2       5142131721450308299360020706.0        5142131721OFFICEDEPOT#19120706.0
3       5142148452550900629625438118.0   5142148452FEDEXSHP12/28/09AB#38118.0
4       5142190439550900629625438118.0   5142190439FEDEXSHP12/23/09AB#38118.0
…                                 …                                        …
96392   5142276053350000000616041042.0        5142276053BESTBUY0000161041042.0
96393   5142225701809071003095045248.0  5142225701MARKUSOFFICESUPPLIES45248.0
96394   5142226486450305734110045150.0            5142226486TECHPAC,INC45150.0
96395   5142244619883400069541292656.0                5142244619BUY.COM92656.0
96396    514224324791083476800067606.0    5142243247STAPLESNATIONAL#4717606.0

                    Merchnum_desc_State U*_cardnum U*_merchnum
0        5509006296254FEDEXSHP12/23/09AB#TN   2.178008         NaN
1        61003026333SERVICEMERCHANDISE#81MA   1.604857    1.001244
2            450308299360OFFICEDEPOT#191MD   2.368143    1.025818
3        5509006296254FEDEXSHP12/28/09AB#TN   1.044105         NaN
4        5509006296254FEDEXSHP12/23/09AB#TN   2.178008         NaN
…                                        …        …         …
96392        3500000006160BESTBUY00001610KY   1.002393    1.001244
96393  8090710030950MARKUSOFFICESUPPLIESOH   1.137948    1.288057
96394           4503057341100TECHPAC,INCOH   1.201338    1.106055
96395               8834000695412BUY.COMCA   1.499767    1.029441
96396    9108347680006STAPLESNATIONAL#471NJ   1.233441    1.113448

[96397 rows x 34 columns]
```

[98]:
```
final['U*_cardnum'].isna().sum()
```

[98]: 72

[99]:
```
final['U*_merchnum'].isna().sum()
```

[99]: 11775

[100]:
```
final['U*_cardnum'].fillna(1,inplace=True)
final['U*_merchnum'].fillna(1,inplace=True)
```

[101]:
```
final['U*_cardnum'].isna().sum()
```

[101]: 0

[102]:
```
final['U*_merchnum'].isna().sum()
```

[102]: 0

[103]:
```
print(final.shape)
final.drop(columns=['U*_cardnum','U*_merchnum'],inplace=True)
print(final.shape)
```

```
(96397, 34)
(96397, 32)
```

[104]:
```
final
```

[104]:
```
       Recnum     Cardnum       Date       Merchnum      Merch description  \
0           1  5142190439 2010-01-01  5509006296254    FEDEXSHP12/23/09AB#
1           2  5142183973 2010-01-01    61003026333  SERVICEMERCHANDISE#81
2           3  5142131721 2010-01-01  4503082993600        OFFICEDEPOT#191
3           4  5142148452 2010-01-01  5509006296254    FEDEXSHP12/28/09AB#
4           5  5142190439 2010-01-01  5509006296254    FEDEXSHP12/23/09AB#
…         …         …         …          …                    …
96392   96749  5142276053 2010-12-31  3500000006160        BESTBUY00001610
96393   96750  5142225701 2010-12-31  8090710030950   MARKUSOFFICESUPPLIES
96394   96751  5142226486 2010-12-31  4503057341100            TECHPAC,INC
96395   96752  5142244619 2010-12-31  8834000695412                BUY.COM
96396   96753  5142243247 2010-12-31  9108347680006    STAPLESNATIONAL#471

      Merch state Merch zip Transtype  Amount  Fraud  …       card_zip3  \
0              TN   38118.0         P    3.62      0  …  5142190439381
1              MA    1803.0         P   31.42      0  …  5142183973180
2              MD   20706.0         P  178.49      0  …  5142131721207
3              TN   38118.0         P    3.62      0  …  5142148452381
4              TN   38118.0         P    3.62      0  …  5142190439381
```

```
…          …     …        …      …    …  …          …
96392      KY  41042.0      P   84.79     0  …  5142276053410
96393      OH  45248.0      P  118.75     0  …  5142225701452
96394      OH  45150.0      P  363.56     0  …  5142226486451
96395      CA  92656.0      P 2202.03     0  …  5142244619926
96396      NJ   7606.0      P  554.64     0  …  5142243247760

                    Card_Merchdesc          Card_dow  \
0        5142190439FEDEXSHP12/23/09AB#  5142190439Friday
1      5142183973SERVICEMERCHANDISE#81  5142183973Friday
2          5142131721OFFICEDEPOT#191  5142131721Friday
3        5142148452FEDEXSHP12/28/09AB#  5142148452Friday
4        5142190439FEDEXSHP12/23/09AB#  5142190439Friday
…                                    …                …
96392      5142276053BESTBUY00001610  5142276053Friday
96393  5142225701MARKUSOFFICESUPPLIES  5142225701Friday
96394             5142226486TECHPAC,INC  5142226486Friday
96395                 5142244619BUY.COM  5142244619Friday
96396    5142243247STAPLESNATIONAL#471  5142243247Friday

                       Merchnum_desc          Merchnum_dow  \
0      5509006296254FEDEXSHP12/23/09AB#  5509006296254Friday
1      61003026333SERVICEMERCHANDISE#81    61003026333Friday
2          4503082993600OFFICEDEPOT#191  4503082993600Friday
3      5509006296254FEDEXSHP12/28/09AB#  5509006296254Friday
4      5509006296254FEDEXSHP12/23/09AB#  5509006296254Friday
…                                     …                    …
96392      3500000006160BESTBUY00001610  3500000006160Friday
96393 8090710030950MARKUSOFFICESUPPLIES  8090710030950Friday
96394             4503057341100TECHPAC,INC  4503057341100Friday
96395                 8834000695412BUY.COM  8834000695412Friday
96396   9108347680006STAPLESNATIONAL#471  9108347680006Friday

                   Merchdesc_dow  \
0        FEDEXSHP12/23/09AB#Friday
1      SERVICEMERCHANDISE#81Friday
2            OFFICEDEPOT#191Friday
3        FEDEXSHP12/28/09AB#Friday
4        FEDEXSHP12/23/09AB#Friday
…                                …
96392          BESTBUY00001610Friday
96393    MARKUSOFFICESUPPLIESFriday
96394              TECHPAC,INCFriday
96395                  BUY.COMFriday
96396      STAPLESNATIONAL#471Friday

                               Card_Merchnum_desc  \
```

```
0         514219043955090062962544FEDEXSHP12/23/09AB#
1         5142183973610030263333SERVICEMERCHANDISE#81
2             514213172145030829936000FFICEDEPOT#191
3         514214845255090062962544FEDEXSHP12/28/09AB#
4         514219043955090062962544FEDEXSHP12/23/09AB#
…                                                   …
96392         5142276053350000000616OBESTBUY00001610
96393   5142225701809071003095OMARKUSOFFICESUPPLIES
96394             5142226486450305734110OTECHPAC,INC
96395                 514224461988340006954 12BUY.COM
96396     514224324791083476800006STAPLESNATIONAL#471

                        Card_Merchnum_Zip                     Card_Merchdesc_Zip  \
0       51421904395509006296254 38118.0   5142190439FEDEXSHP12/23/09AB#38118.0
1          51421839736100302633318O3.0  5142183973SERVICEMERCHANDISE#811803.0
2       51421317214503082993600 20706.0       5142131721OFFICEDEPOT#19120706.0
3       51421484525509006296254 38118.0   5142148452FEDEXSHP12/28/09AB#38118.0
4       51421904395509006296254 38118.0   5142190439FEDEXSHP12/23/09AB#38118.0
…                                     …                                      …
96392   514227605335000000061604 1042.0       5142276053BESTBUY0000161041042.0
96393   51422257018090710030950 45248.0  5142225701MARKUSOFFICESUPPLIES45248.0
96394   51422264864503057341100 45150.0           5142226486TECHPAC,INC45150.0
96395   51422446198834000695412 92656.0               5142244619BUY.COM92656.0
96396    5142243247910834768000 67606.0     5142243247STAPLESNATIONAL#4717606.0

                          Merchnum_desc_State
0         5509006296254FEDEXSHP12/23/09AB#TN
1         61003026333SERVICEMERCHANDISE#81MA
2             4503082993600OFFICEDEPOT#191MD
3         5509006296254FEDEXSHP12/28/09AB#TN
4         5509006296254FEDEXSHP12/23/09AB#TN
…                                          …
96392         3500000006160BESTBUY00001610KY
96393   8090710030950MARKUSOFFICESUPPLIESOH
96394             4503057341100TECHPAC,INCOH
96395                 8834000695412BUY.COMCA
96396     9108347680006STAPLESNATIONAL#471NJ

[96397 rows x 32 columns]
```

[105]:
```
entities
```

[105]:
```
['Cardnum',
 'Merchnum',
 'Month',
 'Month_Risk',
 'state_risk',
```

```
 'card_merch',
 'card_zip',
 'card_state',
 'merch_zip',
 'merch_state',
 'state_des',
 'zip3',
 'card_zip3',
 'Card_Merchdesc',
 'Card_dow',
 'Merchnum_desc',
 'Merchnum_dow',
 'Merchdesc_dow',
 'Card_Merchnum_desc',
 'Card_Merchnum_Zip',
 'Card_Merchdesc_Zip',
 'Merchnum_desc_State']
```

[106]:

```
# If you want, remove some entities that take a long time
# these take a long time and don't add much
entities.remove('state_risk')
entities.remove('zip3')
entities.remove('Month')
entities.remove('Month_Risk')
entities
```

[106]:

```
['Cardnum',
 'Merchnum',
 'card_merch',
 'card_zip',
 'card_state',
 'merch_zip',
 'merch_state',
 'state_des',
 'card_zip3',
 'Card_Merchdesc',
 'Card_dow',
 'Merchnum_desc',
 'Merchnum_dow',
 'Merchdesc_dow',
 'Card_Merchnum_desc',
 'Card_Merchnum_Zip',
 'Card_Merchdesc_Zip',
 'Merchnum_desc_State']
```

[107]:

```
final.shape
numstart = len(final.columns)
```

```
[108]: %%time
start = timeit.default_timer()
for entity in entities:
    try: print(entity,'Run time for the this entity ----------------- {}s'.
 ↪format(timeit.default_timer() - st))
    except: print('')
    st = timeit.default_timer()

# Day-since variables:
    df_l = df1[['Recnum', 'Date', entity]]
    df_r = df1[['check_record', 'check_date', entity, 'Amount']]
    temp = pd.merge(df_l, df_r, left_on = entity, right_on = entity)
    temp1 = temp[temp.Recnum > temp.
 ↪check_record][['Recnum','Date','check_date']]\
                                                    .groupby('Recnum')[['Date',␣
 ↪'check_date']].last()
    mapper = (temp1.Date - temp1.check_date).dt.days
    final[entity + '_day_since'] = final.Recnum.map(mapper)
    final[entity + '_day_since'].fillna((final.Date - pd.
 ↪to_datetime('2006-01-01')).dt.days, inplace = True)
    print('\n' + entity + '_day_since ---> Done')

# Frequency & Amount variables:
    for time in [0,1,3,7,14,30,60]:
        temp2 = temp[(temp.check_date >= (temp.Date - dt.timedelta(time))) &\
                               (temp.Recnum >= temp.check_record)][['Recnum', entity,␣
 ↪'Amount']]
        col_name = entity + '_count_' + str(time)
        mapper2 = temp2.groupby('Recnum')[entity].count()
        final[col_name] = final.Recnum.map(mapper2)
        print(col_name + ' ---> Done')
        final[entity + '_avg_' + str(time)] = final.Recnum.map(temp2.
 ↪groupby('Recnum')['Amount'].mean())
        final[entity + '_max_' + str(time)] = final.Recnum.map(temp2.
 ↪groupby('Recnum')['Amount'].max())
        final[entity + '_med_' + str(time)] = final.Recnum.map(temp2.
 ↪groupby('Recnum')['Amount'].median())
        final[entity + '_total_' + str(time)] = final.Recnum.map(temp2.
 ↪groupby('Recnum')['Amount'].sum())
        final[entity + '_actual/avg_' + str(time)] = final['Amount'] /␣
 ↪final[entity + '_avg_' + str(time)]
        final[entity + '_actual/max_' + str(time)] = final['Amount'] /␣
 ↪final[entity + '_max_' + str(time)]
        final[entity + '_actual/med_' + str(time)] = final['Amount'] /␣
 ↪final[entity + '_med_' + str(time)]
```

```
        final[entity + '_actual/toal_' + str(time)] = final['Amount'] /␣
↪final[entity + '_total_' + str(time)]
        print(entity + ' amount variables over past ' + str(time) + ' --->␣
↪Done')
    del df_l
    del df_r
    del temp
    del temp1
    del temp2
    del mapper2

print('Total run time: {}mins'.format((timeit.default_timer() - start)/60))
```

```
Cardnum_day_since ---> Done
Cardnum_count_0 ---> Done
Cardnum amount variables over past 0 ---> Done
Cardnum_count_1 ---> Done
Cardnum amount variables over past 1 ---> Done
Cardnum_count_3 ---> Done
Cardnum amount variables over past 3 ---> Done
Cardnum_count_7 ---> Done
Cardnum amount variables over past 7 ---> Done
Cardnum_count_14 ---> Done
Cardnum amount variables over past 14 ---> Done
Cardnum_count_30 ---> Done
Cardnum amount variables over past 30 ---> Done
Cardnum_count_60 ---> Done
Cardnum amount variables over past 60 ---> Done
Merchnum Run time for the this entity ----------------- 4.729067958000016s

Merchnum_day_since ---> Done
Merchnum_count_0 ---> Done
Merchnum amount variables over past 0 ---> Done
Merchnum_count_1 ---> Done
Merchnum amount variables over past 1 ---> Done
Merchnum_count_3 ---> Done
Merchnum amount variables over past 3 ---> Done
Merchnum_count_7 ---> Done
Merchnum amount variables over past 7 ---> Done
Merchnum_count_14 ---> Done
Merchnum amount variables over past 14 ---> Done
Merchnum_count_30 ---> Done
Merchnum amount variables over past 30 ---> Done
Merchnum_count_60 ---> Done
Merchnum amount variables over past 60 ---> Done
```

```
card_merch Run time for the this entity ----------------- 28.796616333000003s

card_merch_day_since ---> Done
card_merch_count_0 ---> Done
card_merch amount variables over past 0 ---> Done
card_merch_count_1 ---> Done
card_merch amount variables over past 1 ---> Done
card_merch_count_3 ---> Done
card_merch amount variables over past 3 ---> Done
card_merch_count_7 ---> Done
card_merch amount variables over past 7 ---> Done
card_merch_count_14 ---> Done
card_merch amount variables over past 14 ---> Done
card_merch_count_30 ---> Done
card_merch amount variables over past 30 ---> Done
card_merch_count_60 ---> Done
card_merch amount variables over past 60 ---> Done
card_zip Run time for the this entity ----------------- 1.7529169160000038s

card_zip_day_since ---> Done
card_zip_count_0 ---> Done
card_zip amount variables over past 0 ---> Done
card_zip_count_1 ---> Done
card_zip amount variables over past 1 ---> Done
card_zip_count_3 ---> Done
card_zip amount variables over past 3 ---> Done
card_zip_count_7 ---> Done
card_zip amount variables over past 7 ---> Done
card_zip_count_14 ---> Done
card_zip amount variables over past 14 ---> Done
card_zip_count_30 ---> Done
card_zip amount variables over past 30 ---> Done
card_zip_count_60 ---> Done
card_zip amount variables over past 60 ---> Done
card_state Run time for the this entity ----------------- 1.9156079170000169s

card_state_day_since ---> Done
card_state_count_0 ---> Done
card_state amount variables over past 0 ---> Done
card_state_count_1 ---> Done
card_state amount variables over past 1 ---> Done
card_state_count_3 ---> Done
card_state amount variables over past 3 ---> Done
card_state_count_7 ---> Done
card_state amount variables over past 7 ---> Done
card_state_count_14 ---> Done
card_state amount variables over past 14 ---> Done
card_state_count_30 ---> Done
```

```
card_state amount variables over past 30 ---> Done
card_state_count_60 ---> Done
card_state amount variables over past 60 ---> Done
merch_zip Run time for the this entity ----------------- 2.2823695000000157s

merch_zip_day_since ---> Done
merch_zip_count_0 ---> Done
merch_zip amount variables over past 0 ---> Done
merch_zip_count_1 ---> Done
merch_zip amount variables over past 1 ---> Done
merch_zip_count_3 ---> Done
merch_zip amount variables over past 3 ---> Done
merch_zip_count_7 ---> Done
merch_zip amount variables over past 7 ---> Done
merch_zip_count_14 ---> Done
merch_zip amount variables over past 14 ---> Done
merch_zip_count_30 ---> Done
merch_zip amount variables over past 30 ---> Done
merch_zip_count_60 ---> Done
merch_zip amount variables over past 60 ---> Done
merch_state Run time for the this entity ----------------- 30.321649624999992s

merch_state_day_since ---> Done
merch_state_count_0 ---> Done
merch_state amount variables over past 0 ---> Done
merch_state_count_1 ---> Done
merch_state amount variables over past 1 ---> Done
merch_state_count_3 ---> Done
merch_state amount variables over past 3 ---> Done
merch_state_count_7 ---> Done
merch_state amount variables over past 7 ---> Done
merch_state_count_14 ---> Done
merch_state amount variables over past 14 ---> Done
merch_state_count_30 ---> Done
merch_state amount variables over past 30 ---> Done
merch_state_count_60 ---> Done
merch_state amount variables over past 60 ---> Done
state_des Run time for the this entity ----------------- 30.394130165999997s

state_des_day_since ---> Done
state_des_count_0 ---> Done
state_des amount variables over past 0 ---> Done
state_des_count_1 ---> Done
state_des amount variables over past 1 ---> Done
state_des_count_3 ---> Done
state_des amount variables over past 3 ---> Done
state_des_count_7 ---> Done
state_des amount variables over past 7 ---> Done
```

```
state_des_count_14 ---> Done
state_des amount variables over past 14 ---> Done
state_des_count_30 ---> Done
state_des amount variables over past 30 ---> Done
state_des_count_60 ---> Done
state_des amount variables over past 60 ---> Done
card_zip3 Run time for the this entity ----------------- 5.569443250000006s

card_zip3_day_since ---> Done
card_zip3_count_0 ---> Done
card_zip3 amount variables over past 0 ---> Done
card_zip3_count_1 ---> Done
card_zip3 amount variables over past 1 ---> Done
card_zip3_count_3 ---> Done
card_zip3 amount variables over past 3 ---> Done
card_zip3_count_7 ---> Done
card_zip3 amount variables over past 7 ---> Done
card_zip3_count_14 ---> Done
card_zip3 amount variables over past 14 ---> Done
card_zip3_count_30 ---> Done
card_zip3 amount variables over past 30 ---> Done
card_zip3_count_60 ---> Done
card_zip3 amount variables over past 60 ---> Done
Card_Merchdesc Run time for the this entity ----------------- 2.185508249999998s

Card_Merchdesc_day_since ---> Done
Card_Merchdesc_count_0 ---> Done
Card_Merchdesc amount variables over past 0 ---> Done
Card_Merchdesc_count_1 ---> Done
Card_Merchdesc amount variables over past 1 ---> Done
Card_Merchdesc_count_3 ---> Done
Card_Merchdesc amount variables over past 3 ---> Done
Card_Merchdesc_count_7 ---> Done
Card_Merchdesc amount variables over past 7 ---> Done
Card_Merchdesc_count_14 ---> Done
Card_Merchdesc amount variables over past 14 ---> Done
Card_Merchdesc_count_30 ---> Done
Card_Merchdesc amount variables over past 30 ---> Done
Card_Merchdesc_count_60 ---> Done
Card_Merchdesc amount variables over past 60 ---> Done
Card_dow Run time for the this entity ----------------- 1.219429166999987s

Card_dow_day_since ---> Done
Card_dow_count_0 ---> Done
Card_dow amount variables over past 0 ---> Done
Card_dow_count_1 ---> Done
Card_dow amount variables over past 1 ---> Done
Card_dow_count_3 ---> Done
```

```
Card_dow amount variables over past 3 ---> Done
Card_dow_count_7 ---> Done
Card_dow amount variables over past 7 ---> Done
Card_dow_count_14 ---> Done
Card_dow amount variables over past 14 ---> Done
Card_dow_count_30 ---> Done
Card_dow amount variables over past 30 ---> Done
Card_dow_count_60 ---> Done
Card_dow amount variables over past 60 ---> Done
Merchnum_desc Run time for the this entity ----------------- 1.7811040420000097s

Merchnum_desc_day_since ---> Done
Merchnum_desc_count_0 ---> Done
Merchnum_desc amount variables over past 0 ---> Done
Merchnum_desc_count_1 ---> Done
Merchnum_desc amount variables over past 1 ---> Done
Merchnum_desc_count_3 ---> Done
Merchnum_desc amount variables over past 3 ---> Done
Merchnum_desc_count_7 ---> Done
Merchnum_desc amount variables over past 7 ---> Done
Merchnum_desc_count_14 ---> Done
Merchnum_desc amount variables over past 14 ---> Done
Merchnum_desc_count_30 ---> Done
Merchnum_desc amount variables over past 30 ---> Done
Merchnum_desc_count_60 ---> Done
Merchnum_desc amount variables over past 60 ---> Done
Merchnum_dow Run time for the this entity ----------------- 4.393769208000009s

Merchnum_dow_day_since ---> Done
Merchnum_dow_count_0 ---> Done
Merchnum_dow amount variables over past 0 ---> Done
Merchnum_dow_count_1 ---> Done
Merchnum_dow amount variables over past 1 ---> Done
Merchnum_dow_count_3 ---> Done
Merchnum_dow amount variables over past 3 ---> Done
Merchnum_dow_count_7 ---> Done
Merchnum_dow amount variables over past 7 ---> Done
Merchnum_dow_count_14 ---> Done
Merchnum_dow amount variables over past 14 ---> Done
Merchnum_dow_count_30 ---> Done
Merchnum_dow amount variables over past 30 ---> Done
Merchnum_dow_count_60 ---> Done
Merchnum_dow amount variables over past 60 ---> Done
Merchdesc_dow Run time for the this entity ----------------- 6.163328000000007s

Merchdesc_dow_day_since ---> Done
Merchdesc_dow_count_0 ---> Done
Merchdesc_dow amount variables over past 0 ---> Done
```

```
Merchdesc_dow_count_1 ---> Done
Merchdesc_dow amount variables over past 1 ---> Done
Merchdesc_dow_count_3 ---> Done
Merchdesc_dow amount variables over past 3 ---> Done
Merchdesc_dow_count_7 ---> Done
Merchdesc_dow amount variables over past 7 ---> Done
Merchdesc_dow_count_14 ---> Done
Merchdesc_dow amount variables over past 14 ---> Done
Merchdesc_dow_count_30 ---> Done
Merchdesc_dow amount variables over past 30 ---> Done
Merchdesc_dow_count_60 ---> Done
Merchdesc_dow amount variables over past 60 ---> Done
Card_Merchnum_desc Run time for the this entity -----------------
1.8828737089999947s

Card_Merchnum_desc_day_since ---> Done
Card_Merchnum_desc_count_0 ---> Done
Card_Merchnum_desc amount variables over past 0 ---> Done
Card_Merchnum_desc_count_1 ---> Done
Card_Merchnum_desc amount variables over past 1 ---> Done
Card_Merchnum_desc_count_3 ---> Done
Card_Merchnum_desc amount variables over past 3 ---> Done
Card_Merchnum_desc_count_7 ---> Done
Card_Merchnum_desc amount variables over past 7 ---> Done
Card_Merchnum_desc_count_14 ---> Done
Card_Merchnum_desc amount variables over past 14 ---> Done
Card_Merchnum_desc_count_30 ---> Done
Card_Merchnum_desc amount variables over past 30 ---> Done
Card_Merchnum_desc_count_60 ---> Done
Card_Merchnum_desc amount variables over past 60 ---> Done
Card_Merchnum_Zip Run time for the this entity -----------------
1.2736649579999835s

Card_Merchnum_Zip_day_since ---> Done
Card_Merchnum_Zip_count_0 ---> Done
Card_Merchnum_Zip amount variables over past 0 ---> Done
Card_Merchnum_Zip_count_1 ---> Done
Card_Merchnum_Zip amount variables over past 1 ---> Done
Card_Merchnum_Zip_count_3 ---> Done
Card_Merchnum_Zip amount variables over past 3 ---> Done
Card_Merchnum_Zip_count_7 ---> Done
Card_Merchnum_Zip amount variables over past 7 ---> Done
Card_Merchnum_Zip_count_14 ---> Done
Card_Merchnum_Zip amount variables over past 14 ---> Done
Card_Merchnum_Zip_count_30 ---> Done
Card_Merchnum_Zip amount variables over past 30 ---> Done
Card_Merchnum_Zip_count_60 ---> Done
Card_Merchnum_Zip amount variables over past 60 ---> Done
```

```
Card_Merchdesc_Zip Run time for the this entity -----------------
1.7015918339999985s

Card_Merchdesc_Zip_day_since ---> Done
Card_Merchdesc_Zip_count_0 ---> Done
Card_Merchdesc_Zip amount variables over past 0 ---> Done
Card_Merchdesc_Zip_count_1 ---> Done
Card_Merchdesc_Zip amount variables over past 1 ---> Done
Card_Merchdesc_Zip_count_3 ---> Done
Card_Merchdesc_Zip amount variables over past 3 ---> Done
Card_Merchdesc_Zip_count_7 ---> Done
Card_Merchdesc_Zip amount variables over past 7 ---> Done
Card_Merchdesc_Zip_count_14 ---> Done
Card_Merchdesc_Zip amount variables over past 14 ---> Done
Card_Merchdesc_Zip_count_30 ---> Done
Card_Merchdesc_Zip amount variables over past 30 ---> Done
Card_Merchdesc_Zip_count_60 ---> Done
Card_Merchdesc_Zip amount variables over past 60 ---> Done
Merchnum_desc_State Run time for the this entity -----------------
1.194342292000016s

Merchnum_desc_State_day_since ---> Done
Merchnum_desc_State_count_0 ---> Done
Merchnum_desc_State amount variables over past 0 ---> Done
Merchnum_desc_State_count_1 ---> Done
Merchnum_desc_State amount variables over past 1 ---> Done
Merchnum_desc_State_count_3 ---> Done
Merchnum_desc_State amount variables over past 3 ---> Done
Merchnum_desc_State_count_7 ---> Done
Merchnum_desc_State amount variables over past 7 ---> Done
Merchnum_desc_State_count_14 ---> Done
Merchnum_desc_State amount variables over past 14 ---> Done
Merchnum_desc_State_count_30 ---> Done
Merchnum_desc_State amount variables over past 30 ---> Done
Merchnum_desc_State_count_60 ---> Done
Merchnum_desc_State amount variables over past 60 ---> Done
Total run time: 2.2022122451333333mins
CPU times: user 1min 42s, sys: 36.4 s, total: 2min 18s
Wall time: 2min 12s
```

[109]:
```
print(final.shape)
print('# new variables is ',len(final.columns) - numstart)
numstart = len(final.columns)
```

```
(96397, 1184)
# new variables is  1152
```

[110]:
```
%%time
start = timeit.default_timer()
for ent in entities:
    for d in ['0', '1']:
        for dd in ['7', '14', '30', '60']:
            final[ent + '_count_' + d + '_by_' + dd] =\
            final[ent + '_count_' + d]/(final[ent + '_count_' + dd])/float(dd)
            final[ent + '_total_amount_'+d+'_by_' + dd]=\
            final[ent +'_total_'+d]/(final[ent+'_total_'+dd])/float(dd)

print('run time: {}s'.format(timeit.default_timer() - start))
```

```
run time: 1.7363042919999998s
CPU times: user 1.51 s, sys: 222 ms, total: 1.73 s
Wall time: 1.74 s
```

[111]:
```
final.shape
```

[111]: (96397, 1472)

[112]:
```
print(final.shape)
print('# new variables is ',len(final.columns) - numstart)
numstart = len(final.columns)
```

```
(96397, 1472)
# new variables is  288
```

[113]:
```
start = timeit.default_timer()
for ent in entities:
    for d in ['0', '1']:
        for dd in ['7', '14', '30', '60']:
            final[ent + '_vdratio_' + d +'by' + dd] =\
            final[ent + '_count_' + d + '_by_' + dd]/(final[ent +␣
 ↪'_day_since']+1)

print('run time: {}s'.format(timeit.default_timer() - start))
```

```
run time: 0.5857774579999955s
```

[114]:
```
final.shape
```

[114]: (96397, 1616)

[115]:
```
print(final.shape)
print('# new variables is ',len(final.columns) - numstart)
numstart = len(final.columns)
```

```
(96397, 1616)
# new variables is  144
```

[116]:
```
# start = timeit.default_timer()
# # Cross entity uniqueness variables
# for entity in entities:
#     for field in entities:
#         st = timeit.default_timer()
#         if entity != field:
#             new_attributes = f'{entity}_{field}_nunique'
#             if new_attributes not in list(final.columns):
#                 mapper3 = final.groupby(entity)[field].nunique()
#                 final[new_attributes] = final[entity].map(mapper3)
#         print(f'Run time for entity {entity} in field {field}'+ ' ---> Done')
# print('Total run time: {}mins'.format((timeit.default_timer() - start)/60))
```

[117]:
```
final.shape
```

[117]:
```
(96397, 1616)
```

[118]:
```
print(final.shape)
print('# new variables is ',len(final.columns) - numstart)
numstart = len(final.columns)
```

```
(96397, 1616)
# new variables is  0
```

[119]:
```
# %%time
# print(final.shape)
# final = final.T.drop_duplicates().T
# final.shape
```

[120]:
```
# df2 = data.copy()
# df2['check_date'] = df2.Date
# df2['check_recnum'] = df2.Recnum
# df_2 = df2[['Recnum', 'Date', 'Amount', 'Cardnum', 'Merchnum']]
# df_s = df2[['check_recnum', 'check_date', 'Amount', 'Cardnum', 'Merchnum']]
# temp2 = pd.merge(df_2, df_s, left_on = 'Cardnum', right_on = 'Cardnum')

# #Frequency Mappers
# # groupers = ['Cardnum', 'Merchnum']
# groupers = ['Cardnum']
# for grouper in groupers:
#     for d in [0,1]:
#         for dd in [3,7,14,30]:
#             numerator_df = temp2[(temp2.check_date >= (temp2.Date - dt.
↪timedelta(d))
```

```
#                               & (temp2.Recnum >= temp2.check_recnum)]
#             denominator_df = temp2[(temp2.check_date >= (temp2.Date - dt.
↪timedelta(dd)))
#                               & (temp2.Recnum >= temp2.check_recnum)]

#             numerator = numerator_df.groupby(grouper)['Recnum'].count()
#             denominator = denominator_df.groupby(grouper)['Recnum'].count()/dd

#             colname = 'relative_velocity_count_by_' + grouper + '_' + str(d)␣
↪+ '_days_over_' + str(dd)

#             final[colname] = final[grouper].map(numerator)/final[grouper].
↪map(denominator)
```

[121]:
```
print(final.shape)
print('# new variables is ',len(final.columns) - numstart)
numstart = len(final.columns)
```

```
(96397, 1616)
# new variables is  0
```

[122]:
```
start = timeit.default_timer()
for entity in entities:
    try: print('Run time for the last entity ----------------- {}s'.
↪format(timeit.default_timer() - st))
    except:
        print('')
    st = timeit.default_timer()
    df_l = df1[['Recnum', 'Date', entity,'Amount']]
    df_r = df1[['check_record', 'check_date', entity, 'Amount']]
    temp = pd.merge(df_l, df_r, left_on = entity, right_on = entity)

    for time in [0,1,3,7,14,30]:
        temp2 = temp[(temp.check_date >= (temp.Date - dt.timedelta(time))) &\
                       (temp.Recnum >= temp.check_record)][['Recnum',␣
↪'check_record',entity, 'Amount_x','Amount_y']]
        temp2['Amount_diff']=temp2['Amount_y']-temp2['Amount_x']

        col_name = entity + '_variability_avg_' + str(time)
        mapper2 = temp2.groupby('Recnum')['Amount_diff'].mean()
        final[col_name] = final.Recnum.map(mapper2)
        print(col_name + ' ---> Done')

        col_name = entity + '_variability_max_' + str(time)
        mapper2 = temp2.groupby('Recnum')['Amount_diff'].max()
        final[col_name] = final.Recnum.map(mapper2)
        print(col_name + ' ---> Done')
```

```python
        col_name = entity + '_variability_med_' + str(time)
        mapper2 = temp2.groupby('Recnum')['Amount_diff'].median()
        final[col_name] = final.Recnum.map(mapper2)
        print(col_name + ' ---> Done')

        print(entity + ' amount variables over past ' + str(time) + ' --->␣
 ↪Done')
    del df_l
    del df_r
    del temp
    del temp2

print('Total run time: {}mins'.format((timeit.default_timer() - start)/60))
```

```
Run time for the last entity ----------------- 33.43841124999997s
Cardnum_variability_avg_0 ---> Done
Cardnum_variability_max_0 ---> Done
Cardnum_variability_med_0 ---> Done
Cardnum amount variables over past 0 ---> Done
Cardnum_variability_avg_1 ---> Done
Cardnum_variability_max_1 ---> Done
Cardnum_variability_med_1 ---> Done
Cardnum amount variables over past 1 ---> Done
Cardnum_variability_avg_3 ---> Done
Cardnum_variability_max_3 ---> Done
Cardnum_variability_med_3 ---> Done
Cardnum amount variables over past 3 ---> Done
Cardnum_variability_avg_7 ---> Done
Cardnum_variability_max_7 ---> Done
Cardnum_variability_med_7 ---> Done
Cardnum amount variables over past 7 ---> Done
Cardnum_variability_avg_14 ---> Done
Cardnum_variability_max_14 ---> Done
Cardnum_variability_med_14 ---> Done
Cardnum amount variables over past 14 ---> Done
Cardnum_variability_avg_30 ---> Done
Cardnum_variability_max_30 ---> Done
Cardnum_variability_med_30 ---> Done
Cardnum amount variables over past 30 ---> Done
Run time for the last entity ----------------- 3.4803078329999835s
Merchnum_variability_avg_0 ---> Done
Merchnum_variability_max_0 ---> Done
Merchnum_variability_med_0 ---> Done
Merchnum amount variables over past 0 ---> Done
Merchnum_variability_avg_1 ---> Done
Merchnum_variability_max_1 ---> Done
```

```
Merchnum_variability_med_1 ---> Done
Merchnum amount variables over past 1 ---> Done
Merchnum_variability_avg_3 ---> Done
Merchnum_variability_max_3 ---> Done
Merchnum_variability_med_3 ---> Done
Merchnum amount variables over past 3 ---> Done
Merchnum_variability_avg_7 ---> Done
Merchnum_variability_max_7 ---> Done
Merchnum_variability_med_7 ---> Done
Merchnum amount variables over past 7 ---> Done
Merchnum_variability_avg_14 ---> Done
Merchnum_variability_max_14 ---> Done
Merchnum_variability_med_14 ---> Done
Merchnum amount variables over past 14 ---> Done
Merchnum_variability_avg_30 ---> Done
Merchnum_variability_max_30 ---> Done
Merchnum_variability_med_30 ---> Done
Merchnum amount variables over past 30 ---> Done
Run time for the last entity ----------------- 22.674994750000053s
card_merch_variability_avg_0 ---> Done
card_merch_variability_max_0 ---> Done
card_merch_variability_med_0 ---> Done
card_merch amount variables over past 0 ---> Done
card_merch_variability_avg_1 ---> Done
card_merch_variability_max_1 ---> Done
card_merch_variability_med_1 ---> Done
card_merch amount variables over past 1 ---> Done
card_merch_variability_avg_3 ---> Done
card_merch_variability_max_3 ---> Done
card_merch_variability_med_3 ---> Done
card_merch amount variables over past 3 ---> Done
card_merch_variability_avg_7 ---> Done
card_merch_variability_max_7 ---> Done
card_merch_variability_med_7 ---> Done
card_merch amount variables over past 7 ---> Done
card_merch_variability_avg_14 ---> Done
card_merch_variability_max_14 ---> Done
card_merch_variability_med_14 ---> Done
card_merch amount variables over past 14 ---> Done
card_merch_variability_avg_30 ---> Done
card_merch_variability_max_30 ---> Done
card_merch_variability_med_30 ---> Done
card_merch amount variables over past 30 ---> Done
Run time for the last entity ----------------- 1.042115166999963s
card_zip_variability_avg_0 ---> Done
card_zip_variability_max_0 ---> Done
card_zip_variability_med_0 ---> Done
card_zip amount variables over past 0 ---> Done
```

```
card_zip_variability_avg_1 ---> Done
card_zip_variability_max_1 ---> Done
card_zip_variability_med_1 ---> Done
card_zip amount variables over past 1 ---> Done
card_zip_variability_avg_3 ---> Done
card_zip_variability_max_3 ---> Done
card_zip_variability_med_3 ---> Done
card_zip amount variables over past 3 ---> Done
card_zip_variability_avg_7 ---> Done
card_zip_variability_max_7 ---> Done
card_zip_variability_med_7 ---> Done
card_zip amount variables over past 7 ---> Done
card_zip_variability_avg_14 ---> Done
card_zip_variability_max_14 ---> Done
card_zip_variability_med_14 ---> Done
card_zip amount variables over past 14 ---> Done
card_zip_variability_avg_30 ---> Done
card_zip_variability_max_30 ---> Done
card_zip_variability_med_30 ---> Done
card_zip amount variables over past 30 ---> Done
Run time for the last entity ---------------- 1.123003208s
card_state_variability_avg_0 ---> Done
card_state_variability_max_0 ---> Done
card_state_variability_med_0 ---> Done
card_state amount variables over past 0 ---> Done
card_state_variability_avg_1 ---> Done
card_state_variability_max_1 ---> Done
card_state_variability_med_1 ---> Done
card_state amount variables over past 1 ---> Done
card_state_variability_avg_3 ---> Done
card_state_variability_max_3 ---> Done
card_state_variability_med_3 ---> Done
card_state amount variables over past 3 ---> Done
card_state_variability_avg_7 ---> Done
card_state_variability_max_7 ---> Done
card_state_variability_med_7 ---> Done
card_state amount variables over past 7 ---> Done
card_state_variability_avg_14 ---> Done
card_state_variability_max_14 ---> Done
card_state_variability_med_14 ---> Done
card_state amount variables over past 14 ---> Done
card_state_variability_avg_30 ---> Done
card_state_variability_max_30 ---> Done
card_state_variability_med_30 ---> Done
card_state amount variables over past 30 ---> Done
Run time for the last entity ---------------- 1.3126604170000178s
merch_zip_variability_avg_0 ---> Done
merch_zip_variability_max_0 ---> Done
```

```
merch_zip_variability_med_0 ---> Done
merch_zip amount variables over past 0 ---> Done
merch_zip_variability_avg_1 ---> Done
merch_zip_variability_max_1 ---> Done
merch_zip_variability_med_1 ---> Done
merch_zip amount variables over past 1 ---> Done
merch_zip_variability_avg_3 ---> Done
merch_zip_variability_max_3 ---> Done
merch_zip_variability_med_3 ---> Done
merch_zip amount variables over past 3 ---> Done
merch_zip_variability_avg_7 ---> Done
merch_zip_variability_max_7 ---> Done
merch_zip_variability_med_7 ---> Done
merch_zip amount variables over past 7 ---> Done
merch_zip_variability_avg_14 ---> Done
merch_zip_variability_max_14 ---> Done
merch_zip_variability_med_14 ---> Done
merch_zip amount variables over past 14 ---> Done
merch_zip_variability_avg_30 ---> Done
merch_zip_variability_max_30 ---> Done
merch_zip_variability_med_30 ---> Done
merch_zip amount variables over past 30 ---> Done
Run time for the last entity ----------------- 22.962556542000016s
merch_state_variability_avg_0 ---> Done
merch_state_variability_max_0 ---> Done
merch_state_variability_med_0 ---> Done
merch_state amount variables over past 0 ---> Done
merch_state_variability_avg_1 ---> Done
merch_state_variability_max_1 ---> Done
merch_state_variability_med_1 ---> Done
merch_state amount variables over past 1 ---> Done
merch_state_variability_avg_3 ---> Done
merch_state_variability_max_3 ---> Done
merch_state_variability_med_3 ---> Done
merch_state amount variables over past 3 ---> Done
merch_state_variability_avg_7 ---> Done
merch_state_variability_max_7 ---> Done
merch_state_variability_med_7 ---> Done
merch_state amount variables over past 7 ---> Done
merch_state_variability_avg_14 ---> Done
merch_state_variability_max_14 ---> Done
merch_state_variability_med_14 ---> Done
merch_state amount variables over past 14 ---> Done
merch_state_variability_avg_30 ---> Done
merch_state_variability_max_30 ---> Done
merch_state_variability_med_30 ---> Done
merch_state amount variables over past 30 ---> Done
Run time for the last entity ----------------- 24.289851958000042s
```

```
state_des_variability_avg_0 ---> Done
state_des_variability_max_0 ---> Done
state_des_variability_med_0 ---> Done
state_des amount variables over past 0 ---> Done
state_des_variability_avg_1 ---> Done
state_des_variability_max_1 ---> Done
state_des_variability_med_1 ---> Done
state_des amount variables over past 1 ---> Done
state_des_variability_avg_3 ---> Done
state_des_variability_max_3 ---> Done
state_des_variability_med_3 ---> Done
state_des amount variables over past 3 ---> Done
state_des_variability_avg_7 ---> Done
state_des_variability_max_7 ---> Done
state_des_variability_med_7 ---> Done
state_des amount variables over past 7 ---> Done
state_des_variability_avg_14 ---> Done
state_des_variability_max_14 ---> Done
state_des_variability_med_14 ---> Done
state_des amount variables over past 14 ---> Done
state_des_variability_avg_30 ---> Done
state_des_variability_max_30 ---> Done
state_des_variability_med_30 ---> Done
state_des amount variables over past 30 ---> Done
Run time for the last entity ---------------- 3.7878034169999637s
card_zip3_variability_avg_0 ---> Done
card_zip3_variability_max_0 ---> Done
card_zip3_variability_med_0 ---> Done
card_zip3 amount variables over past 0 ---> Done
card_zip3_variability_avg_1 ---> Done
card_zip3_variability_max_1 ---> Done
card_zip3_variability_med_1 ---> Done
card_zip3 amount variables over past 1 ---> Done
card_zip3_variability_avg_3 ---> Done
card_zip3_variability_max_3 ---> Done
card_zip3_variability_med_3 ---> Done
card_zip3 amount variables over past 3 ---> Done
card_zip3_variability_avg_7 ---> Done
card_zip3_variability_max_7 ---> Done
card_zip3_variability_med_7 ---> Done
card_zip3 amount variables over past 7 ---> Done
card_zip3_variability_avg_14 ---> Done
card_zip3_variability_max_14 ---> Done
card_zip3_variability_med_14 ---> Done
card_zip3 amount variables over past 14 ---> Done
card_zip3_variability_avg_30 ---> Done
card_zip3_variability_max_30 ---> Done
card_zip3_variability_med_30 ---> Done
```

```
card_zip3 amount variables over past 30 ---> Done
Run time for the last entity ----------------- 1.2735787909999772s
Card_Merchdesc_variability_avg_0 ---> Done
Card_Merchdesc_variability_max_0 ---> Done
Card_Merchdesc_variability_med_0 ---> Done
Card_Merchdesc amount variables over past 0 ---> Done
Card_Merchdesc_variability_avg_1 ---> Done
Card_Merchdesc_variability_max_1 ---> Done
Card_Merchdesc_variability_med_1 ---> Done
Card_Merchdesc amount variables over past 1 ---> Done
Card_Merchdesc_variability_avg_3 ---> Done
Card_Merchdesc_variability_max_3 ---> Done
Card_Merchdesc_variability_med_3 ---> Done
Card_Merchdesc amount variables over past 3 ---> Done
Card_Merchdesc_variability_avg_7 ---> Done
Card_Merchdesc_variability_max_7 ---> Done
Card_Merchdesc_variability_med_7 ---> Done
Card_Merchdesc amount variables over past 7 ---> Done
Card_Merchdesc_variability_avg_14 ---> Done
Card_Merchdesc_variability_max_14 ---> Done
Card_Merchdesc_variability_med_14 ---> Done
Card_Merchdesc amount variables over past 14 ---> Done
Card_Merchdesc_variability_avg_30 ---> Done
Card_Merchdesc_variability_max_30 ---> Done
Card_Merchdesc_variability_med_30 ---> Done
Card_Merchdesc amount variables over past 30 ---> Done
Run time for the last entity ----------------- 0.5981405830000313s
Card_dow_variability_avg_0 ---> Done
Card_dow_variability_max_0 ---> Done
Card_dow_variability_med_0 ---> Done
Card_dow amount variables over past 0 ---> Done
Card_dow_variability_avg_1 ---> Done
Card_dow_variability_max_1 ---> Done
Card_dow_variability_med_1 ---> Done
Card_dow amount variables over past 1 ---> Done
Card_dow_variability_avg_3 ---> Done
Card_dow_variability_max_3 ---> Done
Card_dow_variability_med_3 ---> Done
Card_dow amount variables over past 3 ---> Done
Card_dow_variability_avg_7 ---> Done
Card_dow_variability_max_7 ---> Done
Card_dow_variability_med_7 ---> Done
Card_dow amount variables over past 7 ---> Done
Card_dow_variability_avg_14 ---> Done
Card_dow_variability_max_14 ---> Done
Card_dow_variability_med_14 ---> Done
Card_dow amount variables over past 14 ---> Done
Card_dow_variability_avg_30 ---> Done
```

```
Card_dow_variability_max_30 ---> Done
Card_dow_variability_med_30 ---> Done
Card_dow amount variables over past 30 ---> Done
Run time for the last entity ----------------- 0.91397975000001s
Merchnum_desc_variability_avg_0 ---> Done
Merchnum_desc_variability_max_0 ---> Done
Merchnum_desc_variability_med_0 ---> Done
Merchnum_desc amount variables over past 0 ---> Done
Merchnum_desc_variability_avg_1 ---> Done
Merchnum_desc_variability_max_1 ---> Done
Merchnum_desc_variability_med_1 ---> Done
Merchnum_desc amount variables over past 1 ---> Done
Merchnum_desc_variability_avg_3 ---> Done
Merchnum_desc_variability_max_3 ---> Done
Merchnum_desc_variability_med_3 ---> Done
Merchnum_desc amount variables over past 3 ---> Done
Merchnum_desc_variability_avg_7 ---> Done
Merchnum_desc_variability_max_7 ---> Done
Merchnum_desc_variability_med_7 ---> Done
Merchnum_desc amount variables over past 7 ---> Done
Merchnum_desc_variability_avg_14 ---> Done
Merchnum_desc_variability_max_14 ---> Done
Merchnum_desc_variability_med_14 ---> Done
Merchnum_desc amount variables over past 14 ---> Done
Merchnum_desc_variability_avg_30 ---> Done
Merchnum_desc_variability_max_30 ---> Done
Merchnum_desc_variability_med_30 ---> Done
Merchnum_desc amount variables over past 30 ---> Done
Run time for the last entity ----------------- 2.998627499999998s
Merchnum_dow_variability_avg_0 ---> Done
Merchnum_dow_variability_max_0 ---> Done
Merchnum_dow_variability_med_0 ---> Done
Merchnum_dow amount variables over past 0 ---> Done
Merchnum_dow_variability_avg_1 ---> Done
Merchnum_dow_variability_max_1 ---> Done
Merchnum_dow_variability_med_1 ---> Done
Merchnum_dow amount variables over past 1 ---> Done
Merchnum_dow_variability_avg_3 ---> Done
Merchnum_dow_variability_max_3 ---> Done
Merchnum_dow_variability_med_3 ---> Done
Merchnum_dow amount variables over past 3 ---> Done
Merchnum_dow_variability_avg_7 ---> Done
Merchnum_dow_variability_max_7 ---> Done
Merchnum_dow_variability_med_7 ---> Done
Merchnum_dow amount variables over past 7 ---> Done
Merchnum_dow_variability_avg_14 ---> Done
Merchnum_dow_variability_max_14 ---> Done
Merchnum_dow_variability_med_14 ---> Done
```

```
Merchnum_dow amount variables over past 14 ---> Done
Merchnum_dow_variability_avg_30 ---> Done
Merchnum_dow_variability_max_30 ---> Done
Merchnum_dow_variability_med_30 ---> Done
Merchnum_dow amount variables over past 30 ---> Done
Run time for the last entity ----------------- 3.9143187089999856s
Merchdesc_dow_variability_avg_0 ---> Done
Merchdesc_dow_variability_max_0 ---> Done
Merchdesc_dow_variability_med_0 ---> Done
Merchdesc_dow amount variables over past 0 ---> Done
Merchdesc_dow_variability_avg_1 ---> Done
Merchdesc_dow_variability_max_1 ---> Done
Merchdesc_dow_variability_med_1 ---> Done
Merchdesc_dow amount variables over past 1 ---> Done
Merchdesc_dow_variability_avg_3 ---> Done
Merchdesc_dow_variability_max_3 ---> Done
Merchdesc_dow_variability_med_3 ---> Done
Merchdesc_dow amount variables over past 3 ---> Done
Merchdesc_dow_variability_avg_7 ---> Done
Merchdesc_dow_variability_max_7 ---> Done
Merchdesc_dow_variability_med_7 ---> Done
Merchdesc_dow amount variables over past 7 ---> Done
Merchdesc_dow_variability_avg_14 ---> Done
Merchdesc_dow_variability_max_14 ---> Done
Merchdesc_dow_variability_med_14 ---> Done
Merchdesc_dow amount variables over past 14 ---> Done
Merchdesc_dow_variability_avg_30 ---> Done
Merchdesc_dow_variability_max_30 ---> Done
Merchdesc_dow_variability_med_30 ---> Done
Merchdesc_dow amount variables over past 30 ---> Done
Run time for the last entity ----------------- 1.023237292000033s
Card_Merchnum_desc_variability_avg_0 ---> Done
Card_Merchnum_desc_variability_max_0 ---> Done
Card_Merchnum_desc_variability_med_0 ---> Done
Card_Merchnum_desc amount variables over past 0 ---> Done
Card_Merchnum_desc_variability_avg_1 ---> Done
Card_Merchnum_desc_variability_max_1 ---> Done
Card_Merchnum_desc_variability_med_1 ---> Done
Card_Merchnum_desc amount variables over past 1 ---> Done
Card_Merchnum_desc_variability_avg_3 ---> Done
Card_Merchnum_desc_variability_max_3 ---> Done
Card_Merchnum_desc_variability_med_3 ---> Done
Card_Merchnum_desc amount variables over past 3 ---> Done
Card_Merchnum_desc_variability_avg_7 ---> Done
Card_Merchnum_desc_variability_max_7 ---> Done
Card_Merchnum_desc_variability_med_7 ---> Done
Card_Merchnum_desc amount variables over past 7 ---> Done
Card_Merchnum_desc_variability_avg_14 ---> Done
```

```
Card_Merchnum_desc_variability_max_14 ---> Done
Card_Merchnum_desc_variability_med_14 ---> Done
Card_Merchnum_desc amount variables over past 14 ---> Done
Card_Merchnum_desc_variability_avg_30 ---> Done
Card_Merchnum_desc_variability_max_30 ---> Done
Card_Merchnum_desc_variability_med_30 ---> Done
Card_Merchnum_desc amount variables over past 30 ---> Done
Run time for the last entity ----------------- 0.6129164170000081s
Card_Merchnum_Zip_variability_avg_0 ---> Done
Card_Merchnum_Zip_variability_max_0 ---> Done
Card_Merchnum_Zip_variability_med_0 ---> Done
Card_Merchnum_Zip amount variables over past 0 ---> Done
Card_Merchnum_Zip_variability_avg_1 ---> Done
Card_Merchnum_Zip_variability_max_1 ---> Done
Card_Merchnum_Zip_variability_med_1 ---> Done
Card_Merchnum_Zip amount variables over past 1 ---> Done
Card_Merchnum_Zip_variability_avg_3 ---> Done
Card_Merchnum_Zip_variability_max_3 ---> Done
Card_Merchnum_Zip_variability_med_3 ---> Done
Card_Merchnum_Zip amount variables over past 3 ---> Done
Card_Merchnum_Zip_variability_avg_7 ---> Done
Card_Merchnum_Zip_variability_max_7 ---> Done
Card_Merchnum_Zip_variability_med_7 ---> Done
Card_Merchnum_Zip amount variables over past 7 ---> Done
Card_Merchnum_Zip_variability_avg_14 ---> Done
Card_Merchnum_Zip_variability_max_14 ---> Done
Card_Merchnum_Zip_variability_med_14 ---> Done
Card_Merchnum_Zip amount variables over past 14 ---> Done
Card_Merchnum_Zip_variability_avg_30 ---> Done
Card_Merchnum_Zip_variability_max_30 ---> Done
Card_Merchnum_Zip_variability_med_30 ---> Done
Card_Merchnum_Zip amount variables over past 30 ---> Done
Run time for the last entity ----------------- 0.8989931659999684s
Card_Merchdesc_Zip_variability_avg_0 ---> Done
Card_Merchdesc_Zip_variability_max_0 ---> Done
Card_Merchdesc_Zip_variability_med_0 ---> Done
Card_Merchdesc_Zip amount variables over past 0 ---> Done
Card_Merchdesc_Zip_variability_avg_1 ---> Done
Card_Merchdesc_Zip_variability_max_1 ---> Done
Card_Merchdesc_Zip_variability_med_1 ---> Done
Card_Merchdesc_Zip amount variables over past 1 ---> Done
Card_Merchdesc_Zip_variability_avg_3 ---> Done
Card_Merchdesc_Zip_variability_max_3 ---> Done
Card_Merchdesc_Zip_variability_med_3 ---> Done
Card_Merchdesc_Zip amount variables over past 3 ---> Done
Card_Merchdesc_Zip_variability_avg_7 ---> Done
Card_Merchdesc_Zip_variability_max_7 ---> Done
Card_Merchdesc_Zip_variability_med_7 ---> Done
```

```
Card_Merchdesc_Zip amount variables over past 7 ---> Done
Card_Merchdesc_Zip_variability_avg_14 ---> Done
Card_Merchdesc_Zip_variability_max_14 ---> Done
Card_Merchdesc_Zip_variability_med_14 ---> Done
Card_Merchdesc_Zip amount variables over past 14 ---> Done
Card_Merchdesc_Zip_variability_avg_30 ---> Done
Card_Merchdesc_Zip_variability_max_30 ---> Done
Card_Merchdesc_Zip_variability_med_30 ---> Done
Card_Merchdesc_Zip amount variables over past 30 ---> Done
Run time for the last entity ----------------- 0.6224040000000173s
Merchnum_desc_State_variability_avg_0 ---> Done
Merchnum_desc_State_variability_max_0 ---> Done
Merchnum_desc_State_variability_med_0 ---> Done
Merchnum_desc_State amount variables over past 0 ---> Done
Merchnum_desc_State_variability_avg_1 ---> Done
Merchnum_desc_State_variability_max_1 ---> Done
Merchnum_desc_State_variability_med_1 ---> Done
Merchnum_desc_State amount variables over past 1 ---> Done
Merchnum_desc_State_variability_avg_3 ---> Done
Merchnum_desc_State_variability_max_3 ---> Done
Merchnum_desc_State_variability_med_3 ---> Done
Merchnum_desc_State amount variables over past 3 ---> Done
Merchnum_desc_State_variability_avg_7 ---> Done
Merchnum_desc_State_variability_max_7 ---> Done
Merchnum_desc_State_variability_med_7 ---> Done
Merchnum_desc_State amount variables over past 7 ---> Done
Merchnum_desc_State_variability_avg_14 ---> Done
Merchnum_desc_State_variability_max_14 ---> Done
Merchnum_desc_State_variability_med_14 ---> Done
Merchnum_desc_State amount variables over past 14 ---> Done
Merchnum_desc_State_variability_avg_30 ---> Done
Merchnum_desc_State_variability_max_30 ---> Done
Merchnum_desc_State_variability_med_30 ---> Done
Merchnum_desc_State amount variables over past 30 ---> Done
Total run time: 1.6062613722166665mins
```

[123]:
```
final.shape
```

[123]: (96397, 1940)

[124]:
```
print(final.shape)
print('# new variables is ',len(final.columns) - numstart)
numstart = len(final.columns)
```

```
(96397, 1940)
# new variables is  324
```

[125]:

```
%%time
# this cell can take a long time.
start = timeit.default_timer()
for i in entities:
    for v in entities:
        if i==v:
            continue
        else:
            df_c=df1[['Recnum','Date',i]]
            df_d=df1[['check_record','check_date',i,v]]
            temp=pd.merge(df_c,df_d,left_on=i,right_on=i)

        for t in [1,3,7,14,30,60]:
            count_day_df=temp[(temp.check_date>=(temp.Date-dt.
 ↪timedelta(t)))&(temp.Recnum>=temp.check_record)]
            col_name=f'{i}_unique_count_for_{v}_{t}'
            mapper=count_day_df.groupby(['Recnum'])[v].nunique()
            final[col_name]=final.Recnum.map(mapper)

print('Total run time: {}mins'.format((timeit.default_timer() - start)/60))
```

```
Total run time: 34.933968690283336mins
CPU times: user 24min 33s, sys: 13min, total: 37min 33s
Wall time: 34min 56s
```

[126]:

```
final.shape
```

[126]: (96397, 3776)

[127]:

```
start = timeit.default_timer()
for ent in entities:
    print(ent)
    for d in ['0', '1']:
        for dd in ['7', '14', '30', '60']:
            final[ent + '_count_' + d + '_by_' + dd + "_sq"] =\
            final[ent + '_count_' + d]/(final[ent + '_count_' + dd])/
 ↪pow(float(dd),2)
print('run time: {}s'.format(timeit.default_timer() - start))
```

```
Cardnum
Merchnum
card_merch
card_zip
card_state
merch_zip
merch_state
state_des
```

```
card_zip3
Card_Merchdesc
Card_dow
Merchnum_desc
Merchnum_dow
Merchdesc_dow
Card_Merchnum_desc
Card_Merchnum_Zip
Card_Merchdesc_Zip
Merchnum_desc_State
run time: 0.6368081659998097s
```

[128]:
```
final.shape
```

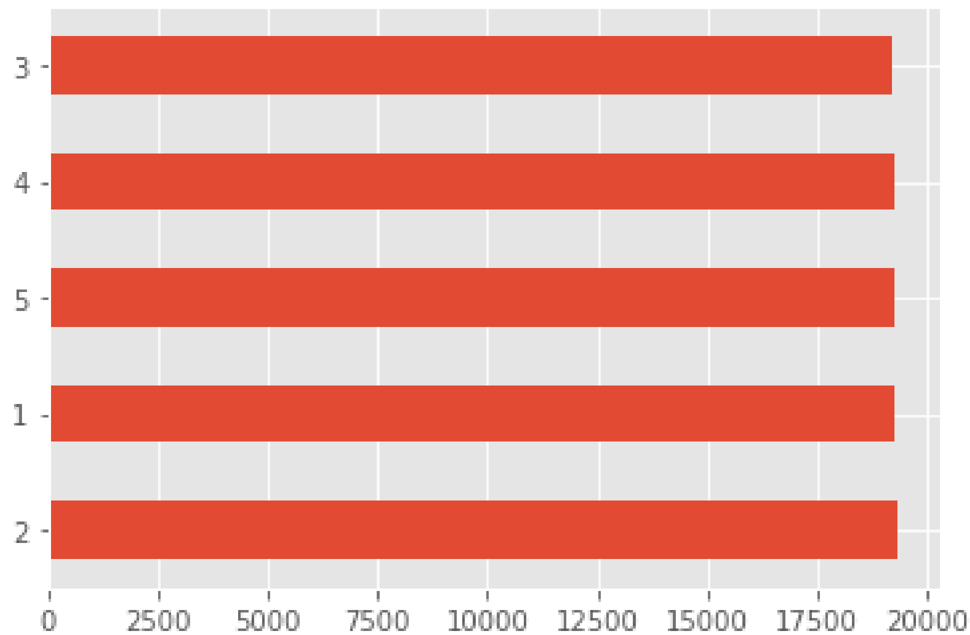
[128]: (96397, 3920)

0.7 Binning Amounts

[132]:
```
# Amount bins
AMOUNT = True
if AMOUNT:
    final['amount_cat'] = pd.qcut(final.Amount, q=5,labels=[1,2,3,4,5])
    final['amount_cat'].value_counts().plot(kind='barh')
    plt.show()

    qcut_series, qcut_intervals = pd.qcut(final.Amount,␣
  ↪q=5,labels=[1,2,3,4,5],retbins=True)

    qcut_series.value_counts()
```




```
[133]: bins = [1,2,3,4,5]
       for bin, interval in zip(bins, qcut_intervals):
           print(bin, round(interval,2))
```

```
1 0.01
2 21.74
3 85.0
4 216.0
5 550.57
```

```
[134]: if AMOUNT:
           final[['Amount', 'amount_cat']].head(10)
```

```
[135]: if AMOUNT:
           final['amount_cat'] = final['amount_cat'].astype(str)
```

```
[136]: # Foreign zipcode
       zip_state = pd.read_csv('zip_code_database.csv')[['zip','state']]
       zip_state.sample(5)
       # Check if the zipcode of merchant is in the US
       zip_state = pd.read_csv('zip_code_database.csv')['zip'].astype(float).
        ↪astype(str).values
       zip_state
```

```
[136]: array(['501.0', '544.0', '601.0', …, '99928.0', '99929.0', '99950.0'],
             dtype=object)
```

[137]:
```
mapping = list(map(lambda x: x not in zip_state, final['Merch zip']))
final = pd.concat([final, pd.DataFrame({'foreign': mapping})], axis = 1)
```

[138]:
```
final.fillna(0,inplace=True)
```

[139]:
```
final.head()
```

[139]:
```
   Recnum     Cardnum       Date        Merchnum      Merch description  \
0       1  5142190439 2010-01-01  5509006296254    FEDEXSHP12/23/09AB#
1       2  5142183973 2010-01-01    61003026333  SERVICEMERCHANDISE#81
2       3  5142131721 2010-01-01  4503082993600        OFFICEDEPOT#191
3       4  5142148452 2010-01-01  5509006296254    FEDEXSHP12/28/09AB#
4       5  5142190439 2010-01-01  5509006296254    FEDEXSHP12/23/09AB#

  Merch state Merch zip Transtype  Amount  Fraud  …  \
0          TN   38118.0         P    3.62      0  …
1          MA    1803.0         P   31.42      0  …
2          MD   20706.0         P  178.49      0  …
3          TN   38118.0         P    3.62      0  …
4          TN   38118.0         P    3.62      0  …

  Merchnum_desc_State_count_0_by_7_sq  Merchnum_desc_State_count_0_by_14_sq  \
0                            0.020408                              0.005102
1                            0.020408                              0.005102
2                            0.020408                              0.005102
3                            0.020408                              0.005102
4                            0.020408                              0.005102

  Merchnum_desc_State_count_0_by_30_sq  Merchnum_desc_State_count_0_by_60_sq  \
0                             0.001111                              0.000278
1                             0.001111                              0.000278
2                             0.001111                              0.000278
3                             0.001111                              0.000278
4                             0.001111                              0.000278

   Merchnum_desc_State_count_1_by_7_sq Merchnum_desc_State_count_1_by_14_sq  \
0                             0.020408                              0.005102
1                             0.020408                              0.005102
2                             0.020408                              0.005102
3                             0.020408                              0.005102
4                             0.020408                              0.005102

  Merchnum_desc_State_count_1_by_30_sq Merchnum_desc_State_count_1_by_60_sq  \
0                             0.001111                              0.000278
1                             0.001111                              0.000278
2                             0.001111                              0.000278
3                             0.001111                              0.000278
```

```
4                              0.001111                              0.000278

   amount_cat foreign
0           1   False
1           2   False
2           3   False
3           1   False
4           1   False

[5 rows x 3922 columns]
```

[140]:
```
final.shape
```

[140]:
```
(96397, 3922)
```

[142]:
```
final.to_csv('final.csv')
```

0.7.1 Remove any redundant columns

[141]:
```
final.set_index('Recnum', inplace = True)
```

[143]:
```
%%time
# if the kernel dies in this cell it's likely due to memory problems.
# In that case, just write out the data file as is and you can read it in␣
↪another notebook that just does deduping
print(final.shape)
final = final.T.drop_duplicates().T
final.shape
```

```
(96397, 3921)
CPU times: user 1min 26s, sys: 53.6 s, total: 2min 19s
Wall time: 2min 41s
```

[143]:
```
(96397, 2885)
```

[144]:
```
final.columns.values.tolist()
```

[144]:
```
['Cardnum',
 'Date',
 'Merchnum',
 'Merch description',
 'Merch state',
 'Merch zip',
 'Transtype',
 'Amount',
 'Fraud',
 'Dow',
```

```
'Dow_Risk',
'Month',
'Month_Risk',
'state_risk',
'card_merch',
'card_zip',
'card_state',
'merch_zip',
'merch_state',
'state_des',
'zip3',
'card_zip3',
'Card_Merchdesc',
'Card_dow',
'Merchnum_desc',
'Merchnum_dow',
'Merchdesc_dow',
'Card_Merchnum_desc',
'Card_Merchnum_Zip',
'Card_Merchdesc_Zip',
'Merchnum_desc_State',
'Cardnum_day_since',
'Cardnum_count_0',
'Cardnum_avg_0',
'Cardnum_max_0',
'Cardnum_med_0',
'Cardnum_total_0',
'Cardnum_actual/avg_0',
'Cardnum_actual/max_0',
'Cardnum_actual/med_0',
'Cardnum_actual/toal_0',
'Cardnum_count_1',
'Cardnum_avg_1',
'Cardnum_max_1',
'Cardnum_med_1',
'Cardnum_total_1',
'Cardnum_actual/avg_1',
'Cardnum_actual/max_1',
'Cardnum_actual/med_1',
'Cardnum_actual/toal_1',
'Cardnum_count_3',
'Cardnum_avg_3',
'Cardnum_max_3',
'Cardnum_med_3',
'Cardnum_total_3',
'Cardnum_actual/avg_3',
'Cardnum_actual/max_3',
```

```
'Cardnum_actual/med_3',
'Cardnum_actual/toal_3',
'Cardnum_count_7',
'Cardnum_avg_7',
'Cardnum_max_7',
'Cardnum_med_7',
'Cardnum_total_7',
'Cardnum_actual/avg_7',
'Cardnum_actual/max_7',
'Cardnum_actual/med_7',
'Cardnum_actual/toal_7',
'Cardnum_count_14',
'Cardnum_avg_14',
'Cardnum_max_14',
'Cardnum_med_14',
'Cardnum_total_14',
'Cardnum_actual/avg_14',
'Cardnum_actual/max_14',
'Cardnum_actual/med_14',
'Cardnum_actual/toal_14',
'Cardnum_count_30',
'Cardnum_avg_30',
'Cardnum_max_30',
'Cardnum_med_30',
'Cardnum_total_30',
'Cardnum_actual/avg_30',
'Cardnum_actual/max_30',
'Cardnum_actual/med_30',
'Cardnum_actual/toal_30',
'Cardnum_count_60',
'Cardnum_avg_60',
'Cardnum_max_60',
'Cardnum_med_60',
'Cardnum_total_60',
'Cardnum_actual/avg_60',
'Cardnum_actual/max_60',
'Cardnum_actual/med_60',
'Cardnum_actual/toal_60',
'Merchnum_day_since',
'Merchnum_count_0',
'Merchnum_avg_0',
'Merchnum_max_0',
'Merchnum_med_0',
'Merchnum_total_0',
'Merchnum_actual/avg_0',
'Merchnum_actual/max_0',
'Merchnum_actual/med_0',
```

```
'Merchnum_actual/toal_0',
'Merchnum_count_1',
'Merchnum_avg_1',
'Merchnum_max_1',
'Merchnum_med_1',
'Merchnum_total_1',
'Merchnum_actual/avg_1',
'Merchnum_actual/max_1',
'Merchnum_actual/med_1',
'Merchnum_actual/toal_1',
'Merchnum_count_3',
'Merchnum_avg_3',
'Merchnum_max_3',
'Merchnum_med_3',
'Merchnum_total_3',
'Merchnum_actual/avg_3',
'Merchnum_actual/max_3',
'Merchnum_actual/med_3',
'Merchnum_actual/toal_3',
'Merchnum_count_7',
'Merchnum_avg_7',
'Merchnum_max_7',
'Merchnum_med_7',
'Merchnum_total_7',
'Merchnum_actual/avg_7',
'Merchnum_actual/max_7',
'Merchnum_actual/med_7',
'Merchnum_actual/toal_7',
'Merchnum_count_14',
'Merchnum_avg_14',
'Merchnum_max_14',
'Merchnum_med_14',
'Merchnum_total_14',
'Merchnum_actual/avg_14',
'Merchnum_actual/max_14',
'Merchnum_actual/med_14',
'Merchnum_actual/toal_14',
'Merchnum_count_30',
'Merchnum_avg_30',
'Merchnum_max_30',
'Merchnum_med_30',
'Merchnum_total_30',
'Merchnum_actual/avg_30',
'Merchnum_actual/max_30',
'Merchnum_actual/med_30',
'Merchnum_actual/toal_30',
'Merchnum_count_60',
```

```
'Merchnum_avg_60',
'Merchnum_max_60',
'Merchnum_med_60',
'Merchnum_total_60',
'Merchnum_actual/avg_60',
'Merchnum_actual/max_60',
'Merchnum_actual/med_60',
'Merchnum_actual/toal_60',
'card_merch_day_since',
'card_merch_count_0',
'card_merch_avg_0',
'card_merch_max_0',
'card_merch_med_0',
'card_merch_total_0',
'card_merch_actual/avg_0',
'card_merch_actual/max_0',
'card_merch_actual/med_0',
'card_merch_actual/toal_0',
'card_merch_count_1',
'card_merch_avg_1',
'card_merch_max_1',
'card_merch_med_1',
'card_merch_total_1',
'card_merch_actual/avg_1',
'card_merch_actual/max_1',
'card_merch_actual/med_1',
'card_merch_actual/toal_1',
'card_merch_count_3',
'card_merch_avg_3',
'card_merch_max_3',
'card_merch_med_3',
'card_merch_total_3',
'card_merch_actual/avg_3',
'card_merch_actual/max_3',
'card_merch_actual/med_3',
'card_merch_actual/toal_3',
'card_merch_count_7',
'card_merch_avg_7',
'card_merch_max_7',
'card_merch_med_7',
'card_merch_total_7',
'card_merch_actual/avg_7',
'card_merch_actual/max_7',
'card_merch_actual/med_7',
'card_merch_actual/toal_7',
'card_merch_count_14',
'card_merch_avg_14',
```

```
'card_merch_max_14',
'card_merch_med_14',
'card_merch_total_14',
'card_merch_actual/avg_14',
'card_merch_actual/max_14',
'card_merch_actual/med_14',
'card_merch_actual/toal_14',
'card_merch_count_30',
'card_merch_avg_30',
'card_merch_max_30',
'card_merch_med_30',
'card_merch_total_30',
'card_merch_actual/avg_30',
'card_merch_actual/max_30',
'card_merch_actual/med_30',
'card_merch_actual/toal_30',
'card_merch_count_60',
'card_merch_avg_60',
'card_merch_max_60',
'card_merch_med_60',
'card_merch_total_60',
'card_merch_actual/avg_60',
'card_merch_actual/max_60',
'card_merch_actual/med_60',
'card_merch_actual/toal_60',
'card_zip_day_since',
'card_zip_count_0',
'card_zip_avg_0',
'card_zip_max_0',
'card_zip_med_0',
'card_zip_total_0',
'card_zip_actual/avg_0',
'card_zip_actual/max_0',
'card_zip_actual/med_0',
'card_zip_actual/toal_0',
'card_zip_count_1',
'card_zip_avg_1',
'card_zip_max_1',
'card_zip_med_1',
'card_zip_total_1',
'card_zip_actual/avg_1',
'card_zip_actual/max_1',
'card_zip_actual/med_1',
'card_zip_actual/toal_1',
'card_zip_count_3',
'card_zip_avg_3',
'card_zip_max_3',
```

```
'card_zip_med_3',
'card_zip_total_3',
'card_zip_actual/avg_3',
'card_zip_actual/max_3',
'card_zip_actual/med_3',
'card_zip_actual/toal_3',
'card_zip_count_7',
'card_zip_avg_7',
'card_zip_max_7',
'card_zip_med_7',
'card_zip_total_7',
'card_zip_actual/avg_7',
'card_zip_actual/max_7',
'card_zip_actual/med_7',
'card_zip_actual/toal_7',
'card_zip_count_14',
'card_zip_avg_14',
'card_zip_max_14',
'card_zip_med_14',
'card_zip_total_14',
'card_zip_actual/avg_14',
'card_zip_actual/max_14',
'card_zip_actual/med_14',
'card_zip_actual/toal_14',
'card_zip_count_30',
'card_zip_avg_30',
'card_zip_max_30',
'card_zip_med_30',
'card_zip_total_30',
'card_zip_actual/avg_30',
'card_zip_actual/max_30',
'card_zip_actual/med_30',
'card_zip_actual/toal_30',
'card_zip_count_60',
'card_zip_avg_60',
'card_zip_max_60',
'card_zip_med_60',
'card_zip_total_60',
'card_zip_actual/avg_60',
'card_zip_actual/max_60',
'card_zip_actual/med_60',
'card_zip_actual/toal_60',
'card_state_day_since',
'card_state_count_0',
'card_state_avg_0',
'card_state_max_0',
'card_state_med_0',
```

```
'card_state_total_0',
'card_state_actual/avg_0',
'card_state_actual/max_0',
'card_state_actual/med_0',
'card_state_actual/toal_0',
'card_state_count_1',
'card_state_avg_1',
'card_state_max_1',
'card_state_med_1',
'card_state_total_1',
'card_state_actual/avg_1',
'card_state_actual/max_1',
'card_state_actual/med_1',
'card_state_actual/toal_1',
'card_state_count_3',
'card_state_avg_3',
'card_state_max_3',
'card_state_med_3',
'card_state_total_3',
'card_state_actual/avg_3',
'card_state_actual/max_3',
'card_state_actual/med_3',
'card_state_actual/toal_3',
'card_state_count_7',
'card_state_avg_7',
'card_state_max_7',
'card_state_med_7',
'card_state_total_7',
'card_state_actual/avg_7',
'card_state_actual/max_7',
'card_state_actual/med_7',
'card_state_actual/toal_7',
'card_state_count_14',
'card_state_avg_14',
'card_state_max_14',
'card_state_med_14',
'card_state_total_14',
'card_state_actual/avg_14',
'card_state_actual/max_14',
'card_state_actual/med_14',
'card_state_actual/toal_14',
'card_state_count_30',
'card_state_avg_30',
'card_state_max_30',
'card_state_med_30',
'card_state_total_30',
'card_state_actual/avg_30',
```

```
'card_state_actual/max_30',
'card_state_actual/med_30',
'card_state_actual/toal_30',
'card_state_count_60',
'card_state_avg_60',
'card_state_max_60',
'card_state_med_60',
'card_state_total_60',
'card_state_actual/avg_60',
'card_state_actual/max_60',
'card_state_actual/med_60',
'card_state_actual/toal_60',
'merch_zip_day_since',
'merch_zip_count_0',
'merch_zip_avg_0',
'merch_zip_max_0',
'merch_zip_med_0',
'merch_zip_total_0',
'merch_zip_actual/avg_0',
'merch_zip_actual/max_0',
'merch_zip_actual/med_0',
'merch_zip_actual/toal_0',
'merch_zip_count_1',
'merch_zip_avg_1',
'merch_zip_max_1',
'merch_zip_med_1',
'merch_zip_total_1',
'merch_zip_actual/avg_1',
'merch_zip_actual/max_1',
'merch_zip_actual/med_1',
'merch_zip_actual/toal_1',
'merch_zip_count_3',
'merch_zip_avg_3',
'merch_zip_max_3',
'merch_zip_med_3',
'merch_zip_total_3',
'merch_zip_actual/avg_3',
'merch_zip_actual/max_3',
'merch_zip_actual/med_3',
'merch_zip_actual/toal_3',
'merch_zip_count_7',
'merch_zip_avg_7',
'merch_zip_max_7',
'merch_zip_med_7',
'merch_zip_total_7',
'merch_zip_actual/avg_7',
'merch_zip_actual/max_7',
```

```
'merch_zip_actual/med_7',
'merch_zip_actual/toal_7',
'merch_zip_count_14',
'merch_zip_avg_14',
'merch_zip_max_14',
'merch_zip_med_14',
'merch_zip_total_14',
'merch_zip_actual/avg_14',
'merch_zip_actual/max_14',
'merch_zip_actual/med_14',
'merch_zip_actual/toal_14',
'merch_zip_count_30',
'merch_zip_avg_30',
'merch_zip_max_30',
'merch_zip_med_30',
'merch_zip_total_30',
'merch_zip_actual/avg_30',
'merch_zip_actual/max_30',
'merch_zip_actual/med_30',
'merch_zip_actual/toal_30',
'merch_zip_count_60',
'merch_zip_avg_60',
'merch_zip_max_60',
'merch_zip_med_60',
'merch_zip_total_60',
'merch_zip_actual/avg_60',
'merch_zip_actual/max_60',
'merch_zip_actual/med_60',
'merch_zip_actual/toal_60',
'merch_state_day_since',
'merch_state_count_0',
'merch_state_avg_0',
'merch_state_max_0',
'merch_state_med_0',
'merch_state_total_0',
'merch_state_actual/avg_0',
'merch_state_actual/max_0',
'merch_state_actual/med_0',
'merch_state_actual/toal_0',
'merch_state_count_1',
'merch_state_avg_1',
'merch_state_max_1',
'merch_state_med_1',
'merch_state_total_1',
'merch_state_actual/avg_1',
'merch_state_actual/max_1',
'merch_state_actual/med_1',
```

```
'merch_state_actual/toal_1',
'merch_state_count_3',
'merch_state_avg_3',
'merch_state_max_3',
'merch_state_med_3',
'merch_state_total_3',
'merch_state_actual/avg_3',
'merch_state_actual/max_3',
'merch_state_actual/med_3',
'merch_state_actual/toal_3',
'merch_state_count_7',
'merch_state_avg_7',
'merch_state_max_7',
'merch_state_med_7',
'merch_state_total_7',
'merch_state_actual/avg_7',
'merch_state_actual/max_7',
'merch_state_actual/med_7',
'merch_state_actual/toal_7',
'merch_state_count_14',
'merch_state_avg_14',
'merch_state_max_14',
'merch_state_med_14',
'merch_state_total_14',
'merch_state_actual/avg_14',
'merch_state_actual/max_14',
'merch_state_actual/med_14',
'merch_state_actual/toal_14',
'merch_state_count_30',
'merch_state_avg_30',
'merch_state_max_30',
'merch_state_med_30',
'merch_state_total_30',
'merch_state_actual/avg_30',
'merch_state_actual/max_30',
'merch_state_actual/med_30',
'merch_state_actual/toal_30',
'merch_state_count_60',
'merch_state_avg_60',
'merch_state_max_60',
'merch_state_med_60',
'merch_state_total_60',
'merch_state_actual/avg_60',
'merch_state_actual/max_60',
'merch_state_actual/med_60',
'merch_state_actual/toal_60',
'state_des_day_since',
```

```
'state_des_count_0',
'state_des_avg_0',
'state_des_max_0',
'state_des_med_0',
'state_des_total_0',
'state_des_actual/avg_0',
'state_des_actual/max_0',
'state_des_actual/med_0',
'state_des_actual/toal_0',
'state_des_count_1',
'state_des_avg_1',
'state_des_max_1',
'state_des_med_1',
'state_des_total_1',
'state_des_actual/avg_1',
'state_des_actual/max_1',
'state_des_actual/med_1',
'state_des_actual/toal_1',
'state_des_count_3',
'state_des_avg_3',
'state_des_max_3',
'state_des_med_3',
'state_des_total_3',
'state_des_actual/avg_3',
'state_des_actual/max_3',
'state_des_actual/med_3',
'state_des_actual/toal_3',
'state_des_count_7',
'state_des_avg_7',
'state_des_max_7',
'state_des_med_7',
'state_des_total_7',
'state_des_actual/avg_7',
'state_des_actual/max_7',
'state_des_actual/med_7',
'state_des_actual/toal_7',
'state_des_count_14',
'state_des_avg_14',
'state_des_max_14',
'state_des_med_14',
'state_des_total_14',
'state_des_actual/avg_14',
'state_des_actual/max_14',
'state_des_actual/med_14',
'state_des_actual/toal_14',
'state_des_count_30',
'state_des_avg_30',
```

```
'state_des_max_30',
'state_des_med_30',
'state_des_total_30',
'state_des_actual/avg_30',
'state_des_actual/max_30',
'state_des_actual/med_30',
'state_des_actual/toal_30',
'state_des_count_60',
'state_des_avg_60',
'state_des_max_60',
'state_des_med_60',
'state_des_total_60',
'state_des_actual/avg_60',
'state_des_actual/max_60',
'state_des_actual/med_60',
'state_des_actual/toal_60',
'card_zip3_day_since',
'card_zip3_count_0',
'card_zip3_avg_0',
'card_zip3_max_0',
'card_zip3_med_0',
'card_zip3_total_0',
'card_zip3_actual/avg_0',
'card_zip3_actual/max_0',
'card_zip3_actual/med_0',
'card_zip3_actual/toal_0',
'card_zip3_count_1',
'card_zip3_avg_1',
'card_zip3_max_1',
'card_zip3_med_1',
'card_zip3_total_1',
'card_zip3_actual/avg_1',
'card_zip3_actual/max_1',
'card_zip3_actual/med_1',
'card_zip3_actual/toal_1',
'card_zip3_count_3',
'card_zip3_avg_3',
'card_zip3_max_3',
'card_zip3_med_3',
'card_zip3_total_3',
'card_zip3_actual/avg_3',
'card_zip3_actual/max_3',
'card_zip3_actual/med_3',
'card_zip3_actual/toal_3',
'card_zip3_count_7',
'card_zip3_avg_7',
'card_zip3_max_7',
```

```
'card_zip3_med_7',
'card_zip3_total_7',
'card_zip3_actual/avg_7',
'card_zip3_actual/max_7',
'card_zip3_actual/med_7',
'card_zip3_actual/toal_7',
'card_zip3_count_14',
'card_zip3_avg_14',
'card_zip3_max_14',
'card_zip3_med_14',
'card_zip3_total_14',
'card_zip3_actual/avg_14',
'card_zip3_actual/max_14',
'card_zip3_actual/med_14',
'card_zip3_actual/toal_14',
'card_zip3_count_30',
'card_zip3_avg_30',
'card_zip3_max_30',
'card_zip3_med_30',
'card_zip3_total_30',
'card_zip3_actual/avg_30',
'card_zip3_actual/max_30',
'card_zip3_actual/med_30',
'card_zip3_actual/toal_30',
'card_zip3_count_60',
'card_zip3_avg_60',
'card_zip3_max_60',
'card_zip3_med_60',
'card_zip3_total_60',
'card_zip3_actual/avg_60',
'card_zip3_actual/max_60',
'card_zip3_actual/med_60',
'card_zip3_actual/toal_60',
'Card_Merchdesc_day_since',
'Card_Merchdesc_count_0',
'Card_Merchdesc_avg_0',
'Card_Merchdesc_max_0',
'Card_Merchdesc_med_0',
'Card_Merchdesc_total_0',
'Card_Merchdesc_actual/avg_0',
'Card_Merchdesc_actual/max_0',
'Card_Merchdesc_actual/med_0',
'Card_Merchdesc_actual/toal_0',
'Card_Merchdesc_count_1',
'Card_Merchdesc_avg_1',
'Card_Merchdesc_max_1',
'Card_Merchdesc_med_1',
```

```
'Card_Merchdesc_total_1',
'Card_Merchdesc_actual/avg_1',
'Card_Merchdesc_actual/max_1',
'Card_Merchdesc_actual/med_1',
'Card_Merchdesc_actual/toal_1',
'Card_Merchdesc_count_3',
'Card_Merchdesc_avg_3',
'Card_Merchdesc_max_3',
'Card_Merchdesc_med_3',
'Card_Merchdesc_total_3',
'Card_Merchdesc_actual/avg_3',
'Card_Merchdesc_actual/max_3',
'Card_Merchdesc_actual/med_3',
'Card_Merchdesc_actual/toal_3',
'Card_Merchdesc_count_7',
'Card_Merchdesc_avg_7',
'Card_Merchdesc_max_7',
'Card_Merchdesc_med_7',
'Card_Merchdesc_total_7',
'Card_Merchdesc_actual/avg_7',
'Card_Merchdesc_actual/max_7',
'Card_Merchdesc_actual/med_7',
'Card_Merchdesc_actual/toal_7',
'Card_Merchdesc_count_14',
'Card_Merchdesc_avg_14',
'Card_Merchdesc_max_14',
'Card_Merchdesc_med_14',
'Card_Merchdesc_total_14',
'Card_Merchdesc_actual/avg_14',
'Card_Merchdesc_actual/max_14',
'Card_Merchdesc_actual/med_14',
'Card_Merchdesc_actual/toal_14',
'Card_Merchdesc_count_30',
'Card_Merchdesc_avg_30',
'Card_Merchdesc_max_30',
'Card_Merchdesc_med_30',
'Card_Merchdesc_total_30',
'Card_Merchdesc_actual/avg_30',
'Card_Merchdesc_actual/max_30',
'Card_Merchdesc_actual/med_30',
'Card_Merchdesc_actual/toal_30',
'Card_Merchdesc_count_60',
'Card_Merchdesc_avg_60',
'Card_Merchdesc_max_60',
'Card_Merchdesc_med_60',
'Card_Merchdesc_total_60',
'Card_Merchdesc_actual/avg_60',
```

```
'Card_Merchdesc_actual/max_60',
'Card_Merchdesc_actual/med_60',
'Card_Merchdesc_actual/toal_60',
'Card_dow_day_since',
'Card_dow_count_7',
'Card_dow_avg_7',
'Card_dow_max_7',
'Card_dow_med_7',
'Card_dow_total_7',
'Card_dow_actual/avg_7',
'Card_dow_actual/max_7',
'Card_dow_actual/med_7',
'Card_dow_actual/toal_7',
'Card_dow_count_14',
'Card_dow_avg_14',
'Card_dow_max_14',
'Card_dow_med_14',
'Card_dow_total_14',
'Card_dow_actual/avg_14',
'Card_dow_actual/max_14',
'Card_dow_actual/med_14',
'Card_dow_actual/toal_14',
'Card_dow_count_30',
'Card_dow_avg_30',
'Card_dow_max_30',
'Card_dow_med_30',
'Card_dow_total_30',
'Card_dow_actual/avg_30',
'Card_dow_actual/max_30',
'Card_dow_actual/med_30',
'Card_dow_actual/toal_30',
'Card_dow_count_60',
'Card_dow_avg_60',
'Card_dow_max_60',
'Card_dow_med_60',
'Card_dow_total_60',
'Card_dow_actual/avg_60',
'Card_dow_actual/max_60',
'Card_dow_actual/med_60',
'Card_dow_actual/toal_60',
'Merchnum_desc_day_since',
'Merchnum_desc_count_0',
'Merchnum_desc_avg_0',
'Merchnum_desc_max_0',
'Merchnum_desc_med_0',
'Merchnum_desc_total_0',
'Merchnum_desc_actual/avg_0',
```

```
'Merchnum_desc_actual/max_0',
'Merchnum_desc_actual/med_0',
'Merchnum_desc_actual/toal_0',
'Merchnum_desc_count_1',
'Merchnum_desc_avg_1',
'Merchnum_desc_max_1',
'Merchnum_desc_med_1',
'Merchnum_desc_total_1',
'Merchnum_desc_actual/avg_1',
'Merchnum_desc_actual/max_1',
'Merchnum_desc_actual/med_1',
'Merchnum_desc_actual/toal_1',
'Merchnum_desc_count_3',
'Merchnum_desc_avg_3',
'Merchnum_desc_max_3',
'Merchnum_desc_med_3',
'Merchnum_desc_total_3',
'Merchnum_desc_actual/avg_3',
'Merchnum_desc_actual/max_3',
'Merchnum_desc_actual/med_3',
'Merchnum_desc_actual/toal_3',
'Merchnum_desc_count_7',
'Merchnum_desc_avg_7',
'Merchnum_desc_max_7',
'Merchnum_desc_med_7',
'Merchnum_desc_total_7',
'Merchnum_desc_actual/avg_7',
'Merchnum_desc_actual/max_7',
'Merchnum_desc_actual/med_7',
'Merchnum_desc_actual/toal_7',
'Merchnum_desc_count_14',
'Merchnum_desc_avg_14',
'Merchnum_desc_max_14',
'Merchnum_desc_med_14',
'Merchnum_desc_total_14',
'Merchnum_desc_actual/avg_14',
'Merchnum_desc_actual/max_14',
'Merchnum_desc_actual/med_14',
'Merchnum_desc_actual/toal_14',
'Merchnum_desc_count_30',
'Merchnum_desc_avg_30',
'Merchnum_desc_max_30',
'Merchnum_desc_med_30',
'Merchnum_desc_total_30',
'Merchnum_desc_actual/avg_30',
'Merchnum_desc_actual/max_30',
'Merchnum_desc_actual/med_30',
```

```
'Merchnum_desc_actual/toal_30',
'Merchnum_desc_count_60',
'Merchnum_desc_avg_60',
'Merchnum_desc_max_60',
'Merchnum_desc_med_60',
'Merchnum_desc_total_60',
'Merchnum_desc_actual/avg_60',
'Merchnum_desc_actual/max_60',
'Merchnum_desc_actual/med_60',
'Merchnum_desc_actual/toal_60',
'Merchnum_dow_day_since',
'Merchnum_dow_count_7',
'Merchnum_dow_avg_7',
'Merchnum_dow_max_7',
'Merchnum_dow_med_7',
'Merchnum_dow_total_7',
'Merchnum_dow_actual/avg_7',
'Merchnum_dow_actual/max_7',
'Merchnum_dow_actual/med_7',
'Merchnum_dow_actual/toal_7',
'Merchnum_dow_count_14',
'Merchnum_dow_avg_14',
'Merchnum_dow_max_14',
'Merchnum_dow_med_14',
'Merchnum_dow_total_14',
'Merchnum_dow_actual/avg_14',
'Merchnum_dow_actual/max_14',
'Merchnum_dow_actual/med_14',
'Merchnum_dow_actual/toal_14',
'Merchnum_dow_count_30',
'Merchnum_dow_avg_30',
'Merchnum_dow_max_30',
'Merchnum_dow_med_30',
'Merchnum_dow_total_30',
'Merchnum_dow_actual/avg_30',
'Merchnum_dow_actual/max_30',
'Merchnum_dow_actual/med_30',
'Merchnum_dow_actual/toal_30',
'Merchnum_dow_count_60',
'Merchnum_dow_avg_60',
'Merchnum_dow_max_60',
'Merchnum_dow_med_60',
'Merchnum_dow_total_60',
'Merchnum_dow_actual/avg_60',
'Merchnum_dow_actual/max_60',
'Merchnum_dow_actual/med_60',
'Merchnum_dow_actual/toal_60',
```

```
'Merchdesc_dow_day_since',
'Merchdesc_dow_count_0',
'Merchdesc_dow_avg_0',
'Merchdesc_dow_max_0',
'Merchdesc_dow_med_0',
'Merchdesc_dow_total_0',
'Merchdesc_dow_actual/avg_0',
'Merchdesc_dow_actual/max_0',
'Merchdesc_dow_actual/med_0',
'Merchdesc_dow_actual/toal_0',
'Merchdesc_dow_count_7',
'Merchdesc_dow_avg_7',
'Merchdesc_dow_max_7',
'Merchdesc_dow_med_7',
'Merchdesc_dow_total_7',
'Merchdesc_dow_actual/avg_7',
'Merchdesc_dow_actual/max_7',
'Merchdesc_dow_actual/med_7',
'Merchdesc_dow_actual/toal_7',
'Merchdesc_dow_count_14',
'Merchdesc_dow_avg_14',
'Merchdesc_dow_max_14',
'Merchdesc_dow_med_14',
'Merchdesc_dow_total_14',
'Merchdesc_dow_actual/avg_14',
'Merchdesc_dow_actual/max_14',
'Merchdesc_dow_actual/med_14',
'Merchdesc_dow_actual/toal_14',
'Merchdesc_dow_count_30',
'Merchdesc_dow_avg_30',
'Merchdesc_dow_max_30',
'Merchdesc_dow_med_30',
'Merchdesc_dow_total_30',
'Merchdesc_dow_actual/avg_30',
'Merchdesc_dow_actual/max_30',
'Merchdesc_dow_actual/med_30',
'Merchdesc_dow_actual/toal_30',
'Merchdesc_dow_count_60',
'Merchdesc_dow_avg_60',
'Merchdesc_dow_max_60',
'Merchdesc_dow_med_60',
'Merchdesc_dow_total_60',
'Merchdesc_dow_actual/avg_60',
'Merchdesc_dow_actual/max_60',
'Merchdesc_dow_actual/med_60',
'Merchdesc_dow_actual/toal_60',
'Card_Merchnum_desc_day_since',
```

```
'Card_Merchnum_desc_count_0',
'Card_Merchnum_desc_avg_0',
'Card_Merchnum_desc_max_0',
'Card_Merchnum_desc_med_0',
'Card_Merchnum_desc_total_0',
'Card_Merchnum_desc_actual/avg_0',
'Card_Merchnum_desc_actual/max_0',
'Card_Merchnum_desc_actual/med_0',
'Card_Merchnum_desc_actual/toal_0',
'Card_Merchnum_desc_count_1',
'Card_Merchnum_desc_avg_1',
'Card_Merchnum_desc_max_1',
'Card_Merchnum_desc_med_1',
'Card_Merchnum_desc_total_1',
'Card_Merchnum_desc_actual/avg_1',
'Card_Merchnum_desc_actual/max_1',
'Card_Merchnum_desc_actual/med_1',
'Card_Merchnum_desc_actual/toal_1',
'Card_Merchnum_desc_count_3',
'Card_Merchnum_desc_avg_3',
'Card_Merchnum_desc_max_3',
'Card_Merchnum_desc_med_3',
'Card_Merchnum_desc_total_3',
'Card_Merchnum_desc_actual/avg_3',
'Card_Merchnum_desc_actual/max_3',
'Card_Merchnum_desc_actual/med_3',
'Card_Merchnum_desc_actual/toal_3',
'Card_Merchnum_desc_count_7',
'Card_Merchnum_desc_avg_7',
'Card_Merchnum_desc_max_7',
'Card_Merchnum_desc_med_7',
'Card_Merchnum_desc_total_7',
'Card_Merchnum_desc_actual/avg_7',
'Card_Merchnum_desc_actual/max_7',
'Card_Merchnum_desc_actual/med_7',
'Card_Merchnum_desc_actual/toal_7',
'Card_Merchnum_desc_count_14',
'Card_Merchnum_desc_avg_14',
'Card_Merchnum_desc_max_14',
'Card_Merchnum_desc_med_14',
'Card_Merchnum_desc_total_14',
'Card_Merchnum_desc_actual/avg_14',
'Card_Merchnum_desc_actual/max_14',
'Card_Merchnum_desc_actual/med_14',
'Card_Merchnum_desc_actual/toal_14',
'Card_Merchnum_desc_count_30',
'Card_Merchnum_desc_avg_30',
```

```
'Card_Merchnum_desc_max_30',
'Card_Merchnum_desc_med_30',
'Card_Merchnum_desc_total_30',
'Card_Merchnum_desc_actual/avg_30',
'Card_Merchnum_desc_actual/max_30',
'Card_Merchnum_desc_actual/med_30',
'Card_Merchnum_desc_actual/toal_30',
'Card_Merchnum_desc_count_60',
'Card_Merchnum_desc_avg_60',
'Card_Merchnum_desc_max_60',
'Card_Merchnum_desc_med_60',
'Card_Merchnum_desc_total_60',
'Card_Merchnum_desc_actual/avg_60',
'Card_Merchnum_desc_actual/max_60',
'Card_Merchnum_desc_actual/med_60',
'Card_Merchnum_desc_actual/toal_60',
'Card_Merchnum_Zip_day_since',
'Card_Merchnum_Zip_count_0',
'Card_Merchnum_Zip_avg_0',
'Card_Merchnum_Zip_max_0',
'Card_Merchnum_Zip_med_0',
'Card_Merchnum_Zip_total_0',
'Card_Merchnum_Zip_actual/avg_0',
'Card_Merchnum_Zip_actual/max_0',
'Card_Merchnum_Zip_actual/med_0',
'Card_Merchnum_Zip_actual/toal_0',
'Card_Merchnum_Zip_count_1',
'Card_Merchnum_Zip_avg_1',
'Card_Merchnum_Zip_max_1',
'Card_Merchnum_Zip_med_1',
'Card_Merchnum_Zip_total_1',
'Card_Merchnum_Zip_actual/avg_1',
'Card_Merchnum_Zip_actual/max_1',
'Card_Merchnum_Zip_actual/med_1',
'Card_Merchnum_Zip_actual/toal_1',
'Card_Merchnum_Zip_count_3',
'Card_Merchnum_Zip_avg_3',
'Card_Merchnum_Zip_max_3',
'Card_Merchnum_Zip_med_3',
'Card_Merchnum_Zip_total_3',
'Card_Merchnum_Zip_actual/avg_3',
'Card_Merchnum_Zip_actual/max_3',
'Card_Merchnum_Zip_actual/med_3',
'Card_Merchnum_Zip_actual/toal_3',
'Card_Merchnum_Zip_count_7',
'Card_Merchnum_Zip_avg_7',
'Card_Merchnum_Zip_max_7',
```

```
'Card_Merchnum_Zip_med_7',
'Card_Merchnum_Zip_total_7',
'Card_Merchnum_Zip_actual/avg_7',
'Card_Merchnum_Zip_actual/max_7',
'Card_Merchnum_Zip_actual/med_7',
'Card_Merchnum_Zip_actual/toal_7',
'Card_Merchnum_Zip_count_14',
'Card_Merchnum_Zip_avg_14',
'Card_Merchnum_Zip_max_14',
'Card_Merchnum_Zip_med_14',
'Card_Merchnum_Zip_total_14',
'Card_Merchnum_Zip_actual/avg_14',
'Card_Merchnum_Zip_actual/max_14',
'Card_Merchnum_Zip_actual/med_14',
'Card_Merchnum_Zip_actual/toal_14',
'Card_Merchnum_Zip_count_30',
'Card_Merchnum_Zip_avg_30',
'Card_Merchnum_Zip_max_30',
'Card_Merchnum_Zip_med_30',
'Card_Merchnum_Zip_total_30',
'Card_Merchnum_Zip_actual/avg_30',
'Card_Merchnum_Zip_actual/max_30',
'Card_Merchnum_Zip_actual/med_30',
'Card_Merchnum_Zip_actual/toal_30',
'Card_Merchnum_Zip_count_60',
'Card_Merchnum_Zip_avg_60',
'Card_Merchnum_Zip_max_60',
'Card_Merchnum_Zip_med_60',
'Card_Merchnum_Zip_total_60',
'Card_Merchnum_Zip_actual/avg_60',
'Card_Merchnum_Zip_actual/max_60',
'Card_Merchnum_Zip_actual/med_60',
'Card_Merchnum_Zip_actual/toal_60',
'Card_Merchdesc_Zip_day_since',
'Card_Merchdesc_Zip_count_0',
'Card_Merchdesc_Zip_avg_0',
'Card_Merchdesc_Zip_max_0',
'Card_Merchdesc_Zip_med_0',
'Card_Merchdesc_Zip_total_0',
'Card_Merchdesc_Zip_actual/avg_0',
'Card_Merchdesc_Zip_actual/max_0',
'Card_Merchdesc_Zip_actual/med_0',
'Card_Merchdesc_Zip_actual/toal_0',
'Card_Merchdesc_Zip_count_1',
'Card_Merchdesc_Zip_avg_1',
'Card_Merchdesc_Zip_max_1',
'Card_Merchdesc_Zip_med_1',
```

```
 'Card_Merchdesc_Zip_total_1',
 'Card_Merchdesc_Zip_actual/avg_1',
 'Card_Merchdesc_Zip_actual/max_1',
 …]
```

[145]:
```
# careful about this line. Modify it so you only keep the variables (including
↪the record # and dependent variable)
final_vars = final.iloc[:, np.r_[8, 10, 11, len(entities)+10:len(final.
↪columns)]]
```

[146]:
```
final_vars.head()
```

[146]:
```
        Fraud  Dow_Risk    Month               Card_Merchnum_Zip  \
Recnum
1           0  0.025994  January  5142190439550900629625438118.0
2           0  0.025994  January     5142183973610030263331803.0
3           0  0.025994  January  5142131721450308299360020706.0
4           0  0.025994  January  5142148452550900629625438118.0
5           0  0.025994  January  5142190439550900629625438118.0

                           Card_Merchdesc_Zip  \
Recnum
1          5142190439FEDEXSHP12/23/09AB#38118.0
2         5142183973SERVICEMERCHANDISE#811803.0
3              5142131721OFFICEDEPOT#19120706.0
4          5142148452FEDEXSHP12/28/09AB#38118.0
5          5142190439FEDEXSHP12/23/09AB#38118.0

                       Merchnum_desc_State Cardnum_day_since Cardnum_count_0  \
Recnum
1         5509006296254FEDEXSHP12/23/09AB#TN            1461.0               1
2         61003026333SERVICEMERCHANDISE#81MA            1461.0               1
3             4503082993600OFFICEDEPOT#191MD            1461.0               1
4         5509006296254FEDEXSHP12/28/09AB#TN            1461.0               1
5         5509006296254FEDEXSHP12/23/09AB#TN               0.0               2

        Cardnum_avg_0 Cardnum_max_0  … Merchnum_desc_State_count_0_by_7_sq  \
Recnum                               …
1                3.62          3.62  …                            0.020408
2               31.42         31.42  …                            0.020408
3              178.49        178.49  …                            0.020408
4                3.62          3.62  …                            0.020408
5                3.62          3.62  …                            0.020408

        Merchnum_desc_State_count_0_by_14_sq  \
Recnum
1                                   0.005102
```

```
2                                 0.005102
3                                 0.005102
4                                 0.005102
5                                 0.005102

       Merchnum_desc_State_count_0_by_30_sq  \
Recnum
1                                 0.001111
2                                 0.001111
3                                 0.001111
4                                 0.001111
5                                 0.001111

       Merchnum_desc_State_count_0_by_60_sq  \
Recnum
1                                 0.000278
2                                 0.000278
3                                 0.000278
4                                 0.000278
5                                 0.000278

       Merchnum_desc_State_count_1_by_7_sq  \
Recnum
1                                0.020408
2                                0.020408
3                                0.020408
4                                0.020408
5                                0.020408

       Merchnum_desc_State_count_1_by_14_sq  \
Recnum
1                                 0.005102
2                                 0.005102
3                                 0.005102
4                                 0.005102
5                                 0.005102

       Merchnum_desc_State_count_1_by_30_sq  \
Recnum
1                                 0.001111
2                                 0.001111
3                                 0.001111
4                                 0.001111
5                                 0.001111

       Merchnum_desc_State_count_1_by_60_sq amount_cat foreign
Recnum
```

```
1                                   0.000278          1   False
2                                   0.000278          2   False
3                                   0.000278          3   False
4                                   0.000278          1   False
5                                   0.000278          1   False

[5 rows x 2860 columns]
```

[147]:
```
final_vars.shape
```

[147]:
```
(96397, 2860)
```

[148]:
```
final_vars.to_csv('candidate_variables.csv')
```

[149]:
```
print('Duration: ', pd.datetime.now() - start_time)
```

```
Duration:  1:02:44.875442
```

[ ]:
```

```