In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
```

데이터 불러오기

In [2]:
```
cc_df = pd.read_csv('~/data/fraud.csv')
```

In [3]:
```
cc_df
```

Out[3]:

	trans_date_trans_time	cc_num	merchant	category	amt	f
0	2019-01-01 00:00:44	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.23	Steph
1	2019-01-01 00:12:34	4956828990005111019	fraud_Schultz, Simonis and Little	grocery_pos	44.71	Kenr
2	2019-01-01 00:17:16	180048185037117	fraud_Kling-Grant	grocery_net	46.28	N
3	2019-01-01 00:20:15	374930071163758	fraud_Deckow-O'Conner	grocery_pos	64.09	Da
4	2019-01-01 00:23:41	2712209726293386	fraud_Balistreri-Nader	misc_pos	25.58	Je
...	...	...	...	...	...	
491129	2020-12-31 23:56:48	6011109736646996	fraud_Botsford and Sons	home	134.26	Rebe
491130	2020-12-31 23:56:57	213112402583773	fraud_Baumbach, Hodkiewicz and Walsh	shopping_pos	25.49	
491131	2020-12-31 23:59:09	3556613125071656	fraud_Hoppe-Parisian	kids_pets	111.84	
491132	2020-12-31 23:59:15	6011724471098086	fraud_Rau-Robel	kids_pets	86.88	
491133	2020-12-31 23:59:34	4170689372027579	fraud_Dare-Marvin	entertainment	38.13	San

491134 rows × 22 columns

In [4]:
```
pd.set_option('display.max_columns', 50)
```

In [5]:
```
cc_df.head()
```

Out[5]:

	trans_date_trans_time	cc_num	merchant	category	amt	first	
0	2019-01-01 00:00:44	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.23	Stephanie	
1	2019-01-01 00:12:34	4956828990005111019	fraud_Schultz, Simonis and Little	grocery_pos	44.71	Kenneth	Rob
2	2019-01-01 00:17:16	180048185037117	fraud_Kling-Grant	grocery_net	46.28	Mary	
3	2019-01-01 00:20:15	374930071163758	fraud_Deckow-O'Conner	grocery_pos	64.09	Daniel	Es
4	2019-01-01 00:23:41	2712209726293386	fraud_Balistreri-Nader	misc_pos	25.58	Jenna	B

In [6]:
```
cc_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 491134 entries, 0 to 491133
Data columns (total 22 columns):
 #   Column                 Non-Null Count   Dtype  
---  ------                 --------------   -----  
 0   trans_date_trans_time  491134 non-null  object 
 1   cc_num                 491134 non-null  int64  
 2   merchant               491134 non-null  object 
 3   category               491134 non-null  object 
 4   amt                    491134 non-null  float64
 5   first                  491134 non-null  object 
 6   last                   491134 non-null  object 
 7   gender                 491134 non-null  object 
 8   street                 491134 non-null  object 
 9   city                   491134 non-null  object 
 10  state                  491134 non-null  object 
 11  zip                    491134 non-null  int64  
 12  lat                    491134 non-null  float64
 13  long                   491134 non-null  float64
 14  city_pop               491134 non-null  int64  
 15  job                    491134 non-null  object 
 16  dob                    491134 non-null  object 
 17  trans_num              491134 non-null  object 
 18  unix_time              491134 non-null  int64  
 19  merch_lat              491134 non-null  float64
 20  merch_long             491134 non-null  float64
 21  is_fraud               491134 non-null  int64  
dtypes: float64(5), int64(5), object(12)
memory usage: 82.4+ MB
```

In [7]:
```
cc_df.describe()
```

Out[7]:

	cc_num	amt	zip	lat	long	city_pop
count	4.911340e+05	491134.000000	491134.000000	491134.000000	491134.000000	4.911340e+05
mean	3.706013e+17	69.050120	50770.532384	37.931230	-90.495619	1.213922e+05
std	1.260229e+18	160.322867	26854.947965	5.341193	12.990732	3.725751e+05
min	5.038744e+11	1.000000	1843.000000	24.655700	-122.345600	4.600000e+01
25%	2.131124e+14	8.960000	28405.000000	33.746700	-97.235100	1.228000e+03
50%	3.531130e+15	42.170000	49628.000000	38.507200	-87.591700	5.760000e+03
75%	4.653879e+15	80.330000	75048.000000	41.520500	-80.731000	5.083500e+04
max	4.956829e+18	25086.940000	99323.000000	48.887800	-69.965600	2.906700e+06

불필요한 컬럼 제거하기

In [8]:
```
cc_df.head(2)
```

Out[8]:

	trans_date_trans_time	cc_num	merchant	category	amt	first	
0	2019-01-01 00:00:44	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.23	Stephanie	
1	2019-01-01 00:12:34	4956828990005111019	fraud_Schultz, Simonis and Little	grocery_pos	44.71	Kenneth	Robin

In [9]:
```
cc_df.columns.unique()
```

Out[9]:
```
Index(['trans_date_trans_time', 'cc_num', 'merchant', 'category', 'amt',
       'first', 'last', 'gender', 'street', 'city', 'state', 'zip', 'lat',
       'long', 'city_pop', 'job', 'dob', 'trans_num', 'unix_time', 'merch_lat',
       'merch_long', 'is_fraud'],
      dtype='object')
```

In [10]:
```
cc_df['merchant'].nunique()
```

Out[10]: 693

In [11]:
```
cc_df['job'].nunique()
```

Out[11]: 110

In [12]:
```
cc_df['cc_num'].nunique()
```

Out[12]: 124

In [13]:
```
cc_df = cc_df.drop(['merchant', 'first', 'last', 'street', 'city', 'state', 'zip', '
```

In [14]:
```
cc_df.sort_values('cc_num')
```

Out[14]:

	trans_date_trans_time	cc_num	category	amt	gender	lat	lor
378075	2020-08-05 17:03:19	503874407318	shopping_pos	7.77	M	29.5894	-98.52(
230588	2019-12-20 22:21:36	503874407318	health_fitness	72.06	M	29.5894	-98.52(
421413	2020-10-10 12:39:32	503874407318	misc_pos	4.78	M	29.5894	-98.52(
468378	2020-12-13 15:55:44	503874407318	kids_pets	84.56	M	29.5894	-98.52(
345085	2020-06-22 23:52:06	503874407318	entertainment	24.33	M	29.5894	-98.52(
...	...	...	...	...	...	...	
53631	2019-04-14 16:57:31	4956828990005111019	entertainment	27.41	M	40.6747	-74.22:
485223	2020-12-27 14:14:40	4956828990005111019	home	28.52	M	40.6747	-74.22:
264968	2020-02-15 13:32:48	4956828990005111019	shopping_pos	2.95	M	40.6747	-74.22:
63475	2019-04-30 17:45:09	4956828990005111019	shopping_pos	7.28	M	40.6747	-74.22:
450904	2020-11-28 12:12:27	4956828990005111019	food_dining	16.22	M	40.6747	-74.22:

491134 rows × 13 columns

구매금액의 z-score계산하기

In [15]:
```
temp = pd.DataFrame({'a': [10,20,30,40,50,5000], 'b':[100,200,300,250,150,310], 'c':
```

In [16]:
```
temp
```

Out[16]:

	a	b	c
0	10	100	10
1	20	200	30
2	30	300	50000
3	40	250	80
4	50	150	40
5	5000	310	70

In [17]:
```
temp.mean()
```

Out[17]:
```
a    858.333333
b    218.333333
c   8371.666667
dtype: float64
```

In [18]:
```
temp.std()
```

Out[18]:
```
a    2029.043289
b      83.765546
c   20393.651381
dtype: float64
```

In [19]:
```
(temp['a']-858.33)/2029.04
```

Out[19]:
```
0   -0.418094
1   -0.413166
2   -0.408237
3   -0.403309
4   -0.398381
5    2.041197
Name: a, dtype: float64
```

In [20]:
```
(temp['b']-218.33)/83.76
```

Out[20]:
```
0   -1.412727
1   -0.218840
2    0.975048
3    0.378104
4   -0.815783
5    1.094436
Name: b, dtype: float64
```

In [21]:
```
(temp['c']-8371.66)/20393.65
```

Out[21]:
```
0   -0.410013
1   -0.409032
2    2.041240
3   -0.406580
4   -0.408542
5   -0.407071
Name: c, dtype: float64
```

In [22]:
```
cc_df['cc_num'].value_counts().sort_values(ascending=True)
```

Out[22]:
```
3511378610369890    3628
4681601008538160    3638
4005676619255478    3638
30551643947183      3638
36913587729122      3641
                    ...
6538891242532018    4386
4642255475285942    4386
4364010865167176    4386
30270432095985      4392
6538441737335434    4392
Name: cc_num, Length: 124, dtype: int64
```

In [23]:
```
amt_info = cc_df.groupby('cc_num')['amt'].agg(['mean','std']).reset_index()
```

In [24]:
```
amt_info
```

Out[24]:

	cc_num	mean	std
0	503874407318	60.253406	127.265783
1	567868110212	83.442558	117.303828
2	571365235126	59.392974	134.289959
3	581686439828	58.578675	149.804992
4	630423337322	56.078113	159.201852
...	...	...	...
119	4792627764422477317	84.135134	107.316736
120	4797297220948468262	56.313583	247.931817
121	4861310130652566408	85.805306	130.998089
122	4906628655840914250	54.243453	154.767184
123	4956828990005111019	59.858059	132.138802

124 rows × 3 columns

로컬로 파일 저장

In [25]:
```
amt_info.to_pickle('./amt_info.pkl')
```

In [26]:
```
amt_info
```

Out[26]:

	cc_num	mean	std
0	503874407318	60.253406	127.265783
1	567868110212	83.442558	117.303828
2	571365235126	59.392974	134.289959
3	581686439828	58.578675	149.804992
4	630423337322	56.078113	159.201852
...	...	...	...
119	4792627764422477317	84.135134	107.316736
120	4797297220948468262	56.313583	247.931817
121	4861310130652566408	85.805306	130.998089
122	4906628655840914250	54.243453	154.767184
123	4956828990005111019	59.858059	132.138802

124 rows × 3 columns

In [27]:
```
cc_df
```

Out[27]:

	trans_date_trans_time	cc_num	category	amt	gender	lat	l
0	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.
1	2019-01-01 00:12:34	4956828990005111019	grocery_pos	44.71	M	40.6747	-74.
2	2019-01-01 00:17:16	180048185037117	grocery_net	46.28	F	40.6152	-74.
3	2019-01-01 00:20:15	374930071163758	grocery_pos	64.09	M	42.2203	-83.
4	2019-01-01 00:23:41	2712209726293386	misc_pos	25.58	F	30.4066	-91.
...	...	...	...	...	...	...	
491129	2020-12-31 23:56:48	6011109736646996	home	134.26	F	34.2651	-77.
491130	2020-12-31 23:56:57	213112402583773	shopping_pos	25.49	F	34.0326	-82.
491131	2020-12-31 23:59:09	3556613125071656	kids_pets	111.84	M	29.0393	-95.
491132	2020-12-31 23:59:15	6011724471098086	kids_pets	86.88	F	46.1966	-118.
491133	2020-12-31 23:59:34	4170689372027579	entertainment	38.13	M	35.6665	-97.

491134 rows × 13 columns

In [28]:
```
cc_df.merge(amt_info, on = 'cc_num', how = 'left')
```

Out[28]:

	trans_date_trans_time	cc_num	category	amt	gender	lat	l
0	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.
1	2019-01-01 00:12:34	4956828990005111019	grocery_pos	44.71	M	40.6747	-74.
2	2019-01-01 00:17:16	180048185037117	grocery_net	46.28	F	40.6152	-74.
3	2019-01-01 00:20:15	374930071163758	grocery_pos	64.09	M	42.2203	-83.
4	2019-01-01 00:23:41	2712209726293386	misc_pos	25.58	F	30.4066	-91.
...	...	...	...	...	...	...	
491129	2020-12-31 23:56:48	6011109736646996	home	134.26	F	34.2651	-77.
491130	2020-12-31 23:56:57	213112402583773	shopping_pos	25.49	F	34.0326	-82.
491131	2020-12-31 23:59:09	3556613125071656	kids_pets	111.84	M	29.0393	-95.
491132	2020-12-31 23:59:15	6011724471098086	kids_pets	86.88	F	46.1966	-118.
491133	2020-12-31 23:59:34	4170689372027579	entertainment	38.13	M	35.6665	-97.

491134 rows × 15 columns

In [29]:
```
cc_df = cc_df.merge(amt_info, on = 'cc_num', how = 'left')
```

In [30]:
```
cc_df.head()
```

Out[30]:

	trans_date_trans_time	cc_num	category	amt	gender	lat	long	ci
0	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	
1	2019-01-01 00:12:34	4956828990005111019	grocery_pos	44.71	M	40.6747	-74.2239	
2	2019-01-01 00:17:16	180048185037117	grocery_net	46.28	F	40.6152	-74.4150	
3	2019-01-01 00:20:15	374930071163758	grocery_pos	64.09	M	42.2203	-83.3583	
4	2019-01-01 00:23:41	2712209726293386	misc_pos	25.58	F	30.4066	-91.1468	

In [31]:
```
cc_df['amt_z'] = (cc_df['amt']-cc_df['mean'])/cc_df['std']
```

In [32]:
```
cc_df
```

Out[32]:

	trans_date_trans_time	cc_num	category	amt	gender	lat	l
0	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.
1	2019-01-01 00:12:34	4956828990005111019	grocery_pos	44.71	M	40.6747	-74.
2	2019-01-01 00:17:16	180048185037117	grocery_net	46.28	F	40.6152	-74.
3	2019-01-01 00:20:15	374930071163758	grocery_pos	64.09	M	42.2203	-83.
4	2019-01-01 00:23:41	2712209726293386	misc_pos	25.58	F	30.4066	-91.
...	...	...	...	...	...	...	
491129	2020-12-31 23:56:48	6011109736646996	home	134.26	F	34.2651	-77.
491130	2020-12-31 23:56:57	213112402583773	shopping_pos	25.49	F	34.0326	-82.
491131	2020-12-31 23:59:09	3556613125071656	kids_pets	111.84	M	29.0393	-95.
491132	2020-12-31 23:59:15	6011724471098086	kids_pets	86.88	F	46.1966	-118.
491133	2020-12-31 23:59:34	4170689372027579	entertainment	38.13	M	35.6665	-97.

491134 rows × 16 columns

In [33]:
```
cc_df.tail()
```

Out[33]:

	trans_date_trans_time	cc_num	category	amt	gender	lat	long
491129	2020-12-31 23:56:48	6011109736646996	home	134.26	F	34.2651	-77.867(
491130	2020-12-31 23:56:57	213112402583773	shopping_pos	25.49	F	34.0326	-82.202
491131	2020-12-31 23:59:09	3556613125071656	kids_pets	111.84	M	29.0393	-95.440
491132	2020-12-31 23:59:15	6011724471098086	kids_pets	86.88	F	46.1966	-118.901
491133	2020-12-31 23:59:34	4170689372027579	entertainment	38.13	M	35.6665	-97.479

In [34]:
```
cc_df['amt_z'].describe()
```

Out[34]:
```
count    4.911340e+05
mean     1.793956e-18
std      9.998748e-01
min     -9.227851e-01
25%     -3.609422e-01
50%     -1.774582e-01
75%      9.572879e-02
max      6.342123e+01
Name: amt_z, dtype: float64
```

In [35]:
```
cc_df
```

Out[35]:

	trans_date_trans_time	cc_num	category	amt	gender	lat	l
0	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.
1	2019-01-01 00:12:34	4956828990005111019	grocery_pos	44.71	M	40.6747	-74.
2	2019-01-01 00:17:16	180048185037117	grocery_net	46.28	F	40.6152	-74.
3	2019-01-01 00:20:15	374930071163758	grocery_pos	64.09	M	42.2203	-83.
4	2019-01-01 00:23:41	2712209726293386	misc_pos	25.58	F	30.4066	-91.
...	...	...	...	...	...	...	
491129	2020-12-31 23:56:48	6011109736646996	home	134.26	F	34.2651	-77.
491130	2020-12-31 23:56:57	213112402583773	shopping_pos	25.49	F	34.0326	-82.
491131	2020-12-31 23:59:09	3556613125071656	kids_pets	111.84	M	29.0393	-95.
491132	2020-12-31 23:59:15	6011724471098086	kids_pets	86.88	F	46.1966	-118.
491133	2020-12-31 23:59:34	4170689372027579	entertainment	38.13	M	35.6665	-97.

491134 rows × 16 columns

In [36]:
```
cc_df.head()
```

Out[36]:

	trans_date_trans_time	cc_num	category	amt	gender	lat	long	ci
0	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	
1	2019-01-01 00:12:34	4956828990005111019	grocery_pos	44.71	M	40.6747	-74.2239	
2	2019-01-01 00:17:16	180048185037117	grocery_net	46.28	F	40.6152	-74.4150	
3	2019-01-01 00:20:15	374930071163758	grocery_pos	64.09	M	42.2203	-83.3583	
4	2019-01-01 00:23:41	2712209726293386	misc_pos	25.58	F	30.4066	-91.1468	

In [37]:
```
cc_df['amt_z'].describe()
```

Out[37]:
```
count    4.911340e+05
mean     1.793956e-18
std      9.998748e-01
min     -9.227851e-01
25%     -3.609422e-01
50%     -1.774582e-01
75%      9.572879e-02
max      6.342123e+01
Name: amt_z, dtype: float64
```

In [38]:
```
cc_df.drop(['mean','std'], axis=1, inplace = True)
```

In [39]:
```
cc_df.head()
```

Out[39]:

	trans_date_trans_time	cc_num	category	amt	gender	lat	long	ci
0	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	
1	2019-01-01 00:12:34	4956828990005111019	grocery_pos	44.71	M	40.6747	-74.2239	
2	2019-01-01 00:17:16	180048185037117	grocery_net	46.28	F	40.6152	-74.4150	
3	2019-01-01 00:20:15	374930071163758	grocery_pos	64.09	M	42.2203	-83.3583	
4	2019-01-01 00:23:41	2712209726293386	misc_pos	25.58	F	30.4066	-91.1468	

In [40]:
```
cc_df.groupby(['cc_num', 'category'])['amt'].agg(['mean', 'std']).reset_index()
```

Out[40]:

	cc_num	category	mean	std
0	503874407318	entertainment	73.282418	103.050402
1	503874407318	food_dining	38.712305	46.548436
2	503874407318	gas_transport	68.457820	14.730440
3	503874407318	grocery_net	48.931302	18.736252
4	503874407318	grocery_pos	61.987806	23.449569
...	...	...	...	...
1731	4956828990005111019	misc_pos	74.177012	168.341518
1732	4956828990005111019	personal_care	35.379382	44.082579
1733	4956828990005111019	shopping_net	70.019115	239.350164
1734	4956828990005111019	shopping_pos	45.988976	174.986921
1735	4956828990005111019	travel	72.701961	456.554619

1736 rows × 4 columns

In [41]:
```
cat_info = cc_df.groupby(['cc_num', 'category'])['amt'].agg(['mean', 'std']).reset_i
```

In [42]:
```
cat_info
```

Out[42]:

	cc_num	category	mean	std
0	503874407318	entertainment	73.282418	103.050402
1	503874407318	food_dining	38.712305	46.548436
2	503874407318	gas_transport	68.457820	14.730440
3	503874407318	grocery_net	48.931302	18.736252
4	503874407318	grocery_pos	61.987806	23.449569
...	...	...	...	...
1731	4956828990005111019	misc_pos	74.177012	168.341518
1732	4956828990005111019	personal_care	35.379382	44.082579
1733	4956828990005111019	shopping_net	70.019115	239.350164
1734	4956828990005111019	shopping_pos	45.988976	174.986921
1735	4956828990005111019	travel	72.701961	456.554619

1736 rows × 4 columns

파일 로컬 저장 2_cat_info

In [43]:
```
cat_info.to_pickle('./cat_info.pkl')
```

In [44]:
```
cc_df = cc_df.merge(cat_info, on='cc_num', how ='left')
```

In [45]:
```
cc_df.head()
```

Out[45]:

	trans_date_trans_time	cc_num	category_x	amt	gender	lat	long	city_pop
0	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149
1	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149
2	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149
3	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149
4	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149

?? 카테고리가 두개가 생겨버리는데.. 어떻게 구별하지?

In [46]:
```
(cc_df['amt'] - cc_df['mean']) / cc_df['std']
```

Out[46]:
```
0          1.021883
1          2.048113
2          3.110189
3          2.722909
4          0.317631
             ...
6875871   -0.172634
6875872    0.163646
6875873   -0.130802
6875874   -0.097054
6875875   -0.055682
Length: 6875876, dtype: float64
```

In [47]:
```
cc_df['cat_amt_z'] = (cc_df['amt'] - cc_df['mean']) / cc_df['std']
```

In [48]:
```
cc_df.head()
```

Out[48]:

	trans_date_trans_time	cc_num	category_x	amt	gender	lat	long	city_pop
0	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149
1	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149
2	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149
3	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149
4	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149

In [49]:
```
cc_df.drop(['mean','std'], axis=1, inplace = True)
```

In [50]:
```
cc_df.head()
```

Out[50]:

	trans_date_trans_time	cc_num	category_x	amt	gender	lat	long	city_pop
0	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149
1	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149
2	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149
3	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149
4	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149

In [54]:
```
cc_df['trans_date_trans_time'] = pd.to_datetime(cc_df['trans_date_trans_time'])
```

In [55]:
```
cc_df['trans_date_trans_time'].dt.hour
```

Out[55]:
```
0          0
1          0
2          0
3          0
4          0
          ..
6875871    23
6875872    23
6875873    23
6875874    23
6875875    23
Name: trans_date_trans_time, Length: 6875876, dtype: int64
```

In [56]:
```
cc_df['trans_date_trans_time'] = pd.to_datetime(cc_df['trans_date_trans_time'])
```

In [57]:
```
cc_df['trans_date_trans_time'].dt.hour
```

Out[57]:
```
0          0
1          0
2          0
3          0
4          0
          ..
6875871    23
6875872    23
6875873    23
6875874    23
6875875    23
Name: trans_date_trans_time, Length: 6875876, dtype: int64
```

In [58]:
```
cc_df['hour'] = cc_df['trans_date_trans_time'].dt.hour
```

In [59]:
```
cc_df.head()
```

Out[59]:

	trans_date_trans_time	cc_num	category_x	amt	gender	lat	long	city_pop
0	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149
1	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149
2	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149
3	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149
4	2019-01-01 00:00:44	630423337322	grocery_pos	107.23	F	48.8878	-118.2105	149

In [60]:
```
def hour_func(x):
    if (x >= 6) and (x < 12):
```

```
        return 'morning'
    elif (x >= 12) and (x < 18):
        return 'afternoon'
    elif (x >= 18) and ( x < 23):
        return 'evening'
    else:
        return 'night'
```

In [61]:
```
cc_df['hour_cat'] = cc_df['hour'].apply(hour_func)
```

In [62]:
```
cc_df['hour_cat'].unique()
```

Out[62]:
```
array(['night', 'morning', 'afternoon', 'evening'], dtype=object)
```

In [67]:
```
# cc_df['hour_cat'].groupby.sum()
cc_df.groupby('hour_cat')['amt'].mean()
```

Out[67]:
```
hour_cat
afternoon    61.714859
evening      62.846885
morning      84.206051
night        80.822763
Name: amt, dtype: float64
```

In [79]:
```
cc_df.groupby('hour_cat').size()
```

Out[79]:
```
hour_cat
afternoon    2475214
evening      2053758
morning       965636
night        1381268
dtype: int64
```

In [82]:
```
cc_df.groupby(['cc_num','hour_cat'])['amt'].count().reset_index()
```

Out[82]:

	cc_num	hour_cat	amt
0	503874407318	afternoon	17920
1	503874407318	evening	15120
2	503874407318	morning	7812
3	503874407318	night	10318
4	567868110212	afternoon	17192
...	...	...	...
491	4906628655840914250	night	11200
492	4956828990005111019	afternoon	17878
493	4956828990005111019	evening	14532
494	4956828990005111019	morning	7812
495	4956828990005111019	night	10976

496 rows × 3 columns

In [83]:
```
hour_cnt = cc_df.groupby(['cc_num','hour_cat'])['amt'].count().reset_index()
```

In [85]:
```
hour_cnt
```

Out[85]:

	cc_num	hour_cat	amt
0	503874407318	afternoon	17920
1	503874407318	evening	15120
2	503874407318	morning	7812
3	503874407318	night	10318
4	567868110212	afternoon	17192
...	...	...	...
491	4906628655840914250	night	11200
492	4956828990005111019	afternoon	17878
493	4956828990005111019	evening	14532
494	4956828990005111019	morning	7812
495	4956828990005111019	night	10976

496 rows × 3 columns

In [86]:
```
cc_df.groupby('cc_num')['amt'].count().reset_index
```

Out[86]:
```
<bound method Series.reset_index of cc_num
503874407318              51170
567868110212              51016
571365235126              61236
581686439828              51142
630423337322              61068
                          ...
4792627764422477317       60998
4797297220948468262       51142
4861310130652566408       51002
4906628655840914250       51170
4956828990005111019       51198
Name: amt, Length: 124, dtype: int64>
```

In [100…
```
all_cnt = cc_df.groupby('cc_num')['amt'].count().reset_index()
```

In [102…
```
all_cnt.head()
```

Out[102]:

	cc_num	amt
0	503874407318	51170
1	567868110212	51016
2	571365235126	61236
3	581686439828	51142
4	630423337322	61068

In [103…
```
hour_cnt.head()
```

Out[103]:

	cc_num	hour_cat	amt
0	503874407318	afternoon	17920
1	503874407318	evening	15120
2	503874407318	morning	7812
3	503874407318	night	10318
4	567868110212	afternoon	17192

In [107…
```
hour_cnt1 = hour_cnt.merge(all_cnt, on='cc_num', how='left')
```

In [112…
```
hour_cnt1
```

Out[112]:

	cc_num	hour_cat	amt_x	amt_y
0	503874407318	afternoon	17920	51170
1	503874407318	evening	15120	51170
2	503874407318	morning	7812	51170
3	503874407318	night	10318	51170
4	567868110212	afternoon	17192	51016
…	…	…	…	…
491	4906628655840914250	night	11200	51170
492	4956828990005111019	afternoon	17878	51198
493	4956828990005111019	evening	14532	51198
494	4956828990005111019	morning	7812	51198
495	4956828990005111019	night	10976	51198

496 rows × 4 columns

In [117…
```
hour_cnt1.rename(columns={'amt_x':'hour_cnt', 'amt_y': 'total_cnt'}, inplace=True)
```

In [119…
```
hour_cnt1['hour_cnt'] / hour_cnt1['total_cnt']
```

Out[119]:
```
0      0.350205
1      0.295486
2      0.152668
3      0.201642
4      0.336992
         ...
491    0.218878
492    0.349193
493    0.283839
494    0.152584
495    0.214383
Length: 496, dtype: float64
```

In [122…
```
hour_cnt1['hour_perc'] = hour_cnt1['hour_cnt'] / hour_cnt1['total_cnt']
```

In [126…
```
hour_cnt1.head(20)
```

Out[126]:

	cc_num	hour_cat	hour_cnt	total_cnt	hour_perc
0	503874407318	afternoon	17920	51170	0.350205
1	503874407318	evening	15120	51170	0.295486
2	503874407318	morning	7812	51170	0.152668
3	503874407318	night	10318	51170	0.201642
4	567868110212	afternoon	17192	51016	0.336992
5	567868110212	evening	14938	51016	0.292810
6	567868110212	morning	7406	51016	0.145170
7	567868110212	night	11480	51016	0.225027
8	571365235126	afternoon	21322	61236	0.348194
9	571365235126	evening	17598	61236	0.287380
10	571365235126	morning	9114	61236	0.148834
11	571365235126	night	13202	61236	0.215592
12	581686439828	afternoon	17192	51142	0.336162
13	581686439828	evening	15736	51142	0.307692
14	581686439828	morning	7602	51142	0.148645
15	581686439828	night	10612	51142	0.207501
16	630423337322	afternoon	22064	61068	0.361302
17	630423337322	evening	17976	61068	0.294360
18	630423337322	morning	9030	61068	0.147868
19	630423337322	night	11998	61068	0.196470

In [133…
```
hour_cnt1.loc[16:19]['hour_perc'].sum()
```

Out[133]:
```
1.0
```

In [ ]:
```
```