In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

1. Data Wrangling ¶

Check each table for values outside of range and missing values.

In [16]:
```
df_trans = pd.read_csv('transactions_v2.csv')
```

In [22]:
```
df_trans.isnull().values.any()
```

Out[22]:
```
msno                    0
payment_method_id       0
payment_plan_days       0
plan_list_price         0
actual_amount_paid      0
is_auto_renew           0
transaction_date        0
membership_expire_date  0
is_cancel               0
dtype: int64
```

In [23]:
```
df_trans.describe()
```

Out[23]:

	payment_method_id	payment_plan_days	plan_list_price	actual_amount_paid
count	1.431009e+06	1.431009e+06	1.431009e+06	1.431009e+06
mean	3.791835e+01	6.601770e+01	2.817870e+02	2.813172e+02
std	4.964805e+00	1.024864e+02	4.351861e+02	4.354200e+02
min	2.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	3.600000e+01	3.000000e+01	9.900000e+01	9.900000e+01
50%	4.000000e+01	3.000000e+01	1.490000e+02	1.490000e+02
75%	4.100000e+01	3.000000e+01	1.490000e+02	1.490000e+02
max	4.100000e+01	4.500000e+02	2.000000e+03	2.000000e+03

Transactions data table doesn't have any missing values and the ranges of the features look good.

In [17]:
```
df_train = pd.read_csv('train_v2.csv')
```

In [18]:
```
df_train.isnull().values.any()
```

Out[18]:
```
False
```

In [20]:
```
df_train.describe()
```

Out[20]:

	is_churn
count	970960.000000
mean	0.089942
std	0.286099
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	1.000000

In [21]:
```
df_members = pd.read_csv('members_v3.csv')
```

In [22]:
```
df_members.isnull().sum()
```

Out[22]:
```
msno                          0
city                          0
bd                            0
gender                  4429505
registered_via                0
registration_init_time        0
dtype: int64
```

The members table has missing gender values.

In [23]:
```
df_members['gender'].isnull().sum()/len(df_members)
```

Out[23]:
```
0.65433527838873129
```

There is a large percentage of users whose gener is unknown, 65%. Since the percentage is so large, I will keep the data points with the missing values.

In [5]:
```
df_members.describe()
```

Out[5]:

	city	bd	registered_via	registration_init_time
count	6.769473e+06	6.769473e+06	6.769473e+06	6.769473e+06
mean	3.847358e+00	9.795794e+00	5.253069e+00	2.014518e+07
std	5.478359e+00	1.792590e+01	2.361398e+00	2.318601e+04
min	1.000000e+00	-7.168000e+03	-1.000000e+00	2.004033e+07
25%	1.000000e+00	0.000000e+00	4.000000e+00	2.014042e+07
50%	1.000000e+00	0.000000e+00	4.000000e+00	2.015101e+07
75%	4.000000e+00	2.100000e+01	7.000000e+00	2.016060e+07
max	2.200000e+01	2.016000e+03	1.900000e+01	2.017043e+07

The feature bd gives the users age, and there are some values outside of a range that makes sense. The min age is -7168 and the max age is 2016. The proportion of bad ages is:

In [26]:
```
len(df_members[(df_members['bd']<0) | (df_members['bd']>100)])/len(df_members)
```

Out[26]: 0.0008347769464476777

The max value of registered_via should be 16, and the min value should be 3. However, there are values that lie outside of this range. The proportion of bad values is:

In [33]:
```
len(df_members[(df_members['registered_via']<3) | (df_members['registered_via']>16)])/len(df_members)
```

Out[33]: 0.000586308564935557

Since the number of users with bad ages is a small percentage of the total users, I will remove these rows.

In [27]:
```
df_logs = pd.read_csv('user_logs_v2.csv')
```

In [28]:
```
df_logs.describe()
```

Out[28]:

	date	num_25	num_50	num_75	num_985	nu
count	1.839636e+07	1.839636e+07	1.839636e+07	1.839636e+07	1.839636e+07	1.83963
mean	2.017032e+07	6.191401e+00	1.508789e+00	9.413759e-01	1.079905e+00	3.02824
std	8.916720e+00	1.342827e+01	3.908539e+00	1.924840e+00	3.518409e+00	4.20364
min	2.017030e+07	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.00000
25%	2.017031e+07	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	7.00000
50%	2.017032e+07	2.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	1.70000
75%	2.017032e+07	7.000000e+00	2.000000e+00	1.000000e+00	1.000000e+00	3.70000
max	2.017033e+07	5.639000e+03	9.120000e+02	5.080000e+02	1.561000e+03	4.11070

In [30]:
```
df_logs.isnull().values.any()
```

Out[30]: False

The logs table doesn't have any null values, and the ranges for the features seems sensible.

2. Data Storytelling

In [1]:
```
sns.hist('is_autorenew',df=df_trans)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call
 last)
<ipython-input-1-9984a562c7ab> in <module>()
----> 1 sns.hist('is_autorenew',df=df_trans)

NameError: name 'sns' is not defined
```