```
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

Q1: Count something

The mean number of unique songs listened to daily is about 29.

```
In [32]: df_logs.num_unq.mean()

Out[32]: 29.036145516162382
```

Looking at the mean of is_auto_renew, we see that the majority of users, 78%, have their plans set to automatically renew.

```
In [26]: df_trans.is_auto_renew.mean()

Out[26]: 0.7853025382789347
```

Q2: Find some trends

The registration of new members is increasing over time.

```
In [46]: pd.to_datetime(df_members['registration_init_time'], format='%Y%m%d')
         .dt.year.value_counts()
```

```
Out[46]: 2016    2246761
         2015    1620525
         2014     975776
         2013     524722
         2017     481684
         2012     283190
         2011     179051
         2010     115075
         2007      89830
         2008      67690
         2009      63633
         2006      53953
         2005      41349
         2004      26234
         Name: registration_init_time, dtype: int64
```

The month with the most signups is January, and the month with the fewest signups is May (TODO: START AND STOP SERIES AT THE SAME MONTH).

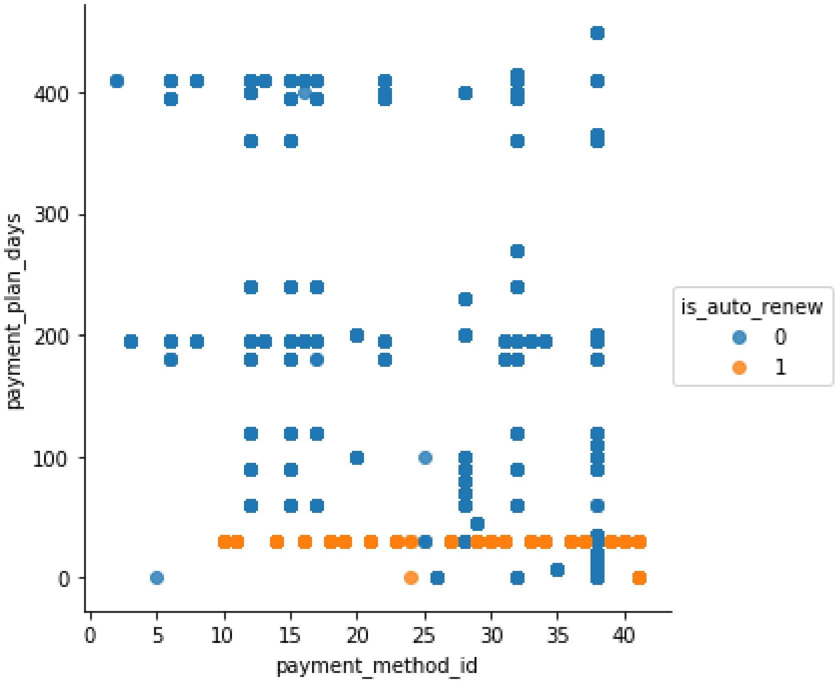
```
In [6]: pd.to_datetime(df_members['registration_init_time'], format='%Y%m%d')
        .dt.month.value_counts()
```

```
Out[6]: 1     688712
        12    652783
        2     647056
        10    645650
        11    643998
        3     591504
        8     540394
        7     492109
        9     489340
        4     477881
        6     465483
        5     434563
        Name: registration_init_time, dtype: int64
```

Q5. Make scatterplots

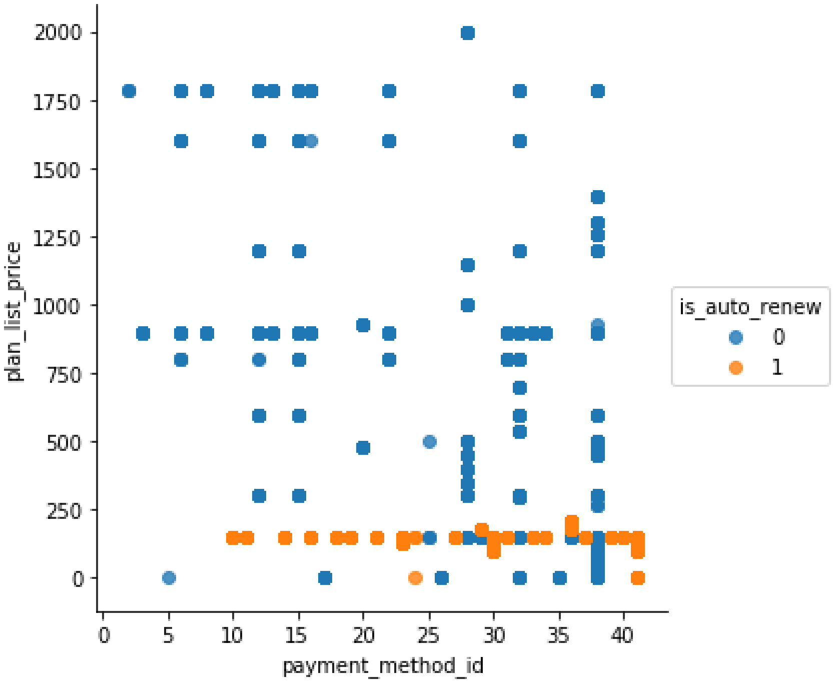
All of the auto_renew plans have only 2 lengths (payment_plan_days).

```
In [22]: sns.lmplot(x="payment_method_id", y="payment_plan_days", hue="is_auto
         _renew", data=df_trans,fit_reg=False)

Out[22]: <seaborn.axisgrid.FacetGrid at 0x7f70c6f5bd30>
```



These two auto_renew plans have a few different prices.

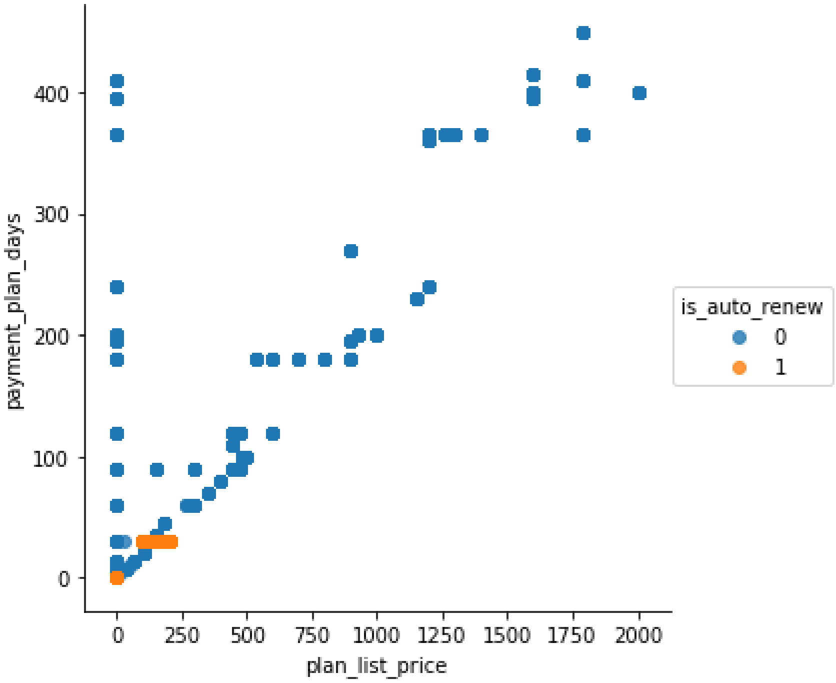
```
In [28]: sns.lmplot(x="payment_method_id", y="plan_list_price", hue="is_auto_r
         enew", data=df_trans,fit_reg=False)

Out[28]: <seaborn.axisgrid.FacetGrid at 0x7f70c6ec49e8>
```



From the scatterplot of payment_plan_days vs plan_list_price it looks like there is a max plan price for each plan duration.

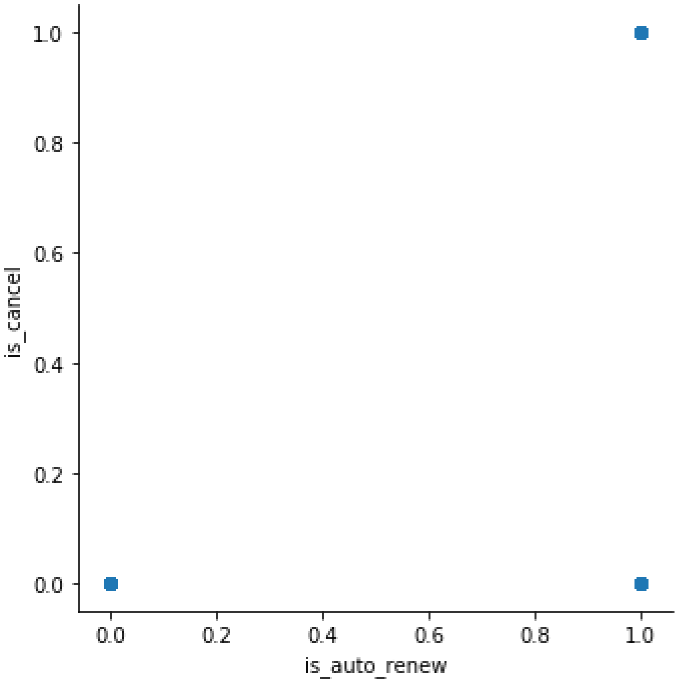
```
In [27]: sns.lmplot(x="plan_list_price", y="payment_plan_days", hue="is_auto_r
         enew", data=df_trans,fit_reg=False)
```

```
Out[27]: <seaborn.axisgrid.FacetGrid at 0x7f70c2c166d8>
```



Auto renew is never false when is_cancel is true.

```
In [30]: sns.lmplot(x="is_auto_renew", y="is_cancel", data=df_trans,fit_reg=False)

Out[30]: <seaborn.axisgrid.FacetGrid at 0x7f70c2baecc0>
```


is_cancel
is_auto_renew