```
In [116]: all_data = pd.merge(all_data,df_train,on='msno',how='inner')
```

```
---------------------------------------------------------------------
------
ValueError                                Traceback (most recent call
 last)
ValueError: Buffer dtype mismatch, expected 'Python object' but got
 'long'

Exception ignored in: 'pandas._libs.lib.is_bool_array'
ValueError: Buffer dtype mismatch, expected 'Python object' but got
'long'

---------------------------------------------------------------------
------
ValueError                                Traceback (most recent call
 last)
ValueError: Buffer dtype mismatch, expected 'Python object' but got
 'long'

Exception ignored in: 'pandas._libs.lib.is_bool_array'
ValueError: Buffer dtype mismatch, expected 'Python object' but got
'long'

---------------------------------------------------------------------
------
ValueError                                Traceback (most recent call
 last)
ValueError: Buffer dtype mismatch, expected 'Python object' but got
 'long'

Exception ignored in: 'pandas._libs.lib.is_bool_array'
ValueError: Buffer dtype mismatch, expected 'Python object' but got
'long'
```

```
In [117]: all_data.to_csv('all_merged.csv')
```

```
In [38]: all_data = pd.read_csv('all_merged.csv')
```

```
In [81]: all_data['city'] = all_data['city'].astype('category')
all_data['gender'] = all_data['gender'].astype('category')
all_data['registered_via'] = all_data['registered_via'].astype('categ
ory')
all_data['registration_init_time'] = all_data['registration_init_tim
e'].astype('category')
all_data['payment_method_id'] = all_data['payment_method_id'].astype(
'category')
```

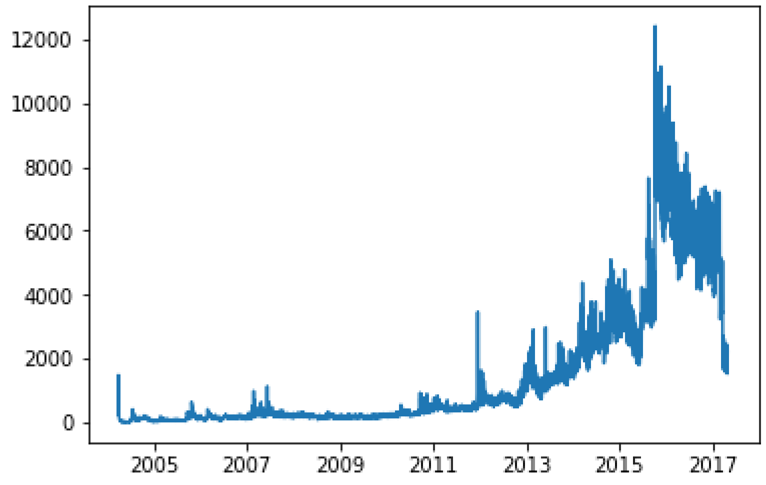
Data Storytelling

Who are the listeners?

The service wasn't very popular until 2010, at which point signups began to slowly increase. After 2015, signups increased strongly. However, recently new registrations have been declining.

```
In [14]: plt.plot(pd.to_datetime(df_members['registration_init_time'], format=
         '%Y%m%d').value_counts().sort_index())

Out[14]: [<matplotlib.lines.Line2D at 0x7f036ac904e0>]
```



About 10% of the users have churned.

```
In [51]: len(df_train[df_train.is_churn==1])/len(df_train)

Out[51]: 0.08994191315811156
```

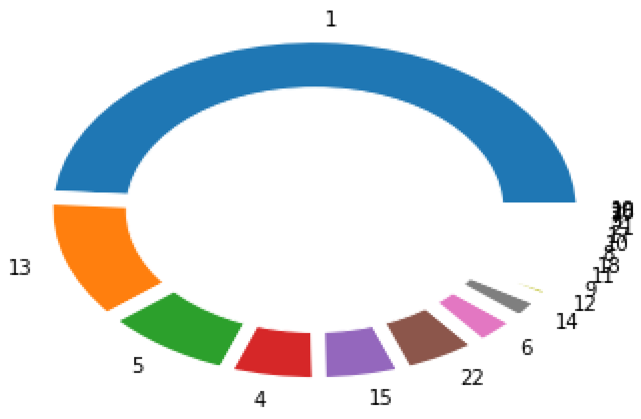
Only a small proportion of users in the data set canceled their subscriptions. The fraction that cancelled is:

```
In [52]: len(df_trans[df_trans.is_cancel==1])/len(df_trans)

Out[52]: 0.02455120827332323
```

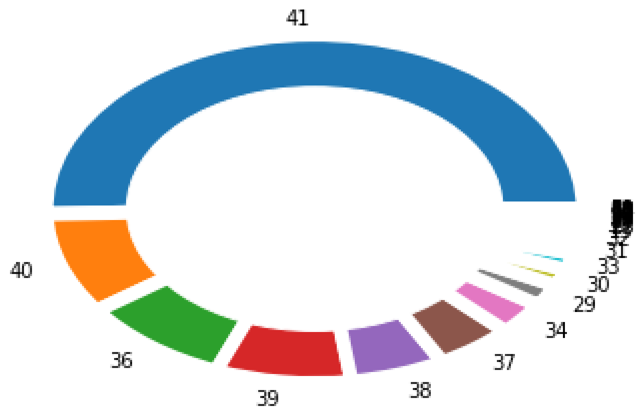
Users mostly come from city 1.

```
In [125]: plt.pie(all_data.city.value_counts(),labels=all_data.city.value_count
          s().index,wedgeprops = { 'linewidth' : 7, 'edgecolor' : 'white' })
          my_circle=plt.Circle( (0,0),0.7, color='white')
          p=plt.gcf()
          p.gca().add_artist(my_circle)
          plt.show()
```



Half of the users use payment method id 41.

```
In [40]: plt.pie(all_data.payment_method_id.value_counts(),labels=all_data.pay
         ment_method_id.value_counts().index,wedgeprops = { 'linewidth' : 7,
         'edgecolor' : 'white' })
         my_circle=plt.Circle( (0,0),0.7, color='white')
         p=plt.gcf()
         p.gca().add_artist(my_circle)
         plt.show()
```

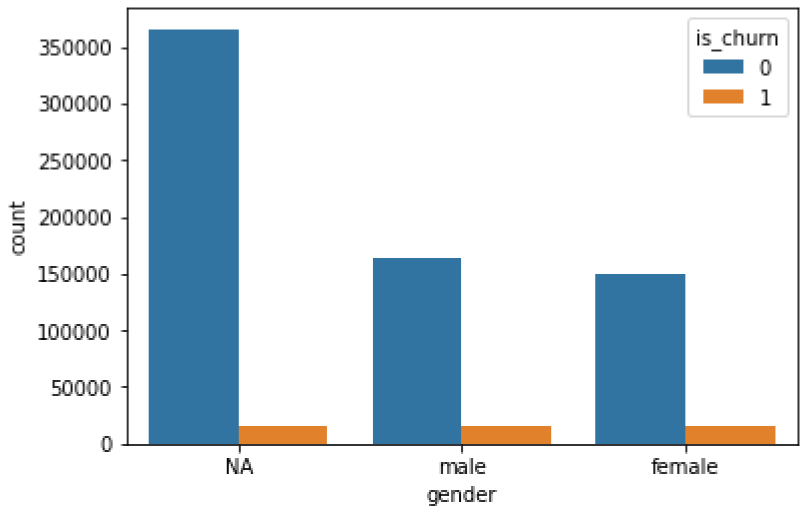


The distributions for gender are very similar for churners and non-churners. However, people who don't list their gender are less likely to churn.

```
In [41]: sns.countplot(x="gender",data=all_data.fillna('NA'),hue='is_churn')
```

```
/home/rebecca/anaconda3/envs/my_projects_env/lib/python3.6/site-packa
ges/seaborn/categorical.py:1508: FutureWarning: remove_na is deprecat
ed and is a private function. Do not use.
  stat_data = remove_na(group_data[hue_mask])
```

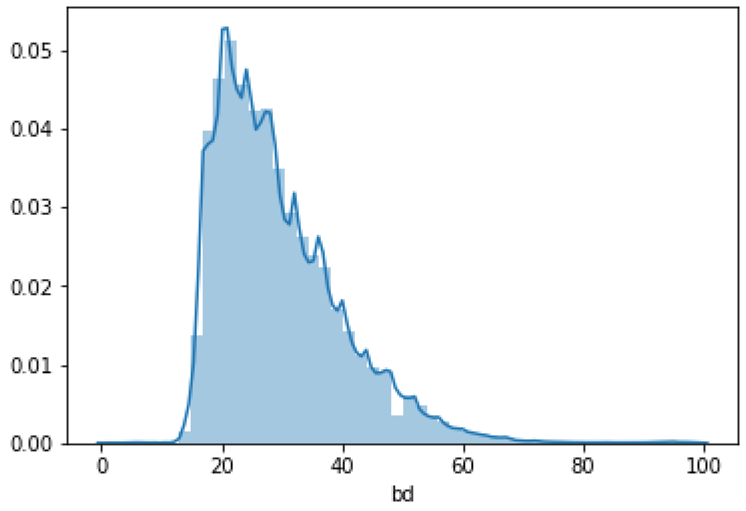
```
Out[41]: <matplotlib.axes._subplots.AxesSubplot at 0x7f741f030470>
```



Plotting only sensible values for the age, shows that most of the users are teenageers and young adults. The distribution of ages peaks around 25, and then decreases.

```
In [26]: sns.distplot(df_members['bd'][(df_members.bd>0) & (df_members.bd<100
         )])
```

```
Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x7f0ffdcd3cc0>
```

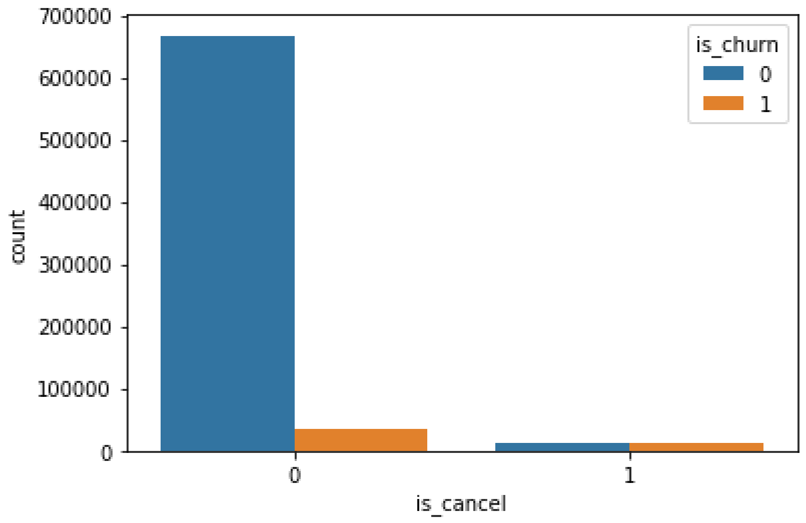


People who haven't canceled are much less likely to churn than those who have canceled.

```
In [83]: sns.countplot(x="is_cancel",data=all_data,hue='is_churn')

/home/rebecca/anaconda3/envs/my_projects_env/lib/python3.6/site-packa
ges/seaborn/categorical.py:1508: FutureWarning: remove_na is deprecat
ed and is a private function. Do not use.
  stat_data = remove_na(group_data[hue_mask])

Out[83]: <matplotlib.axes._subplots.AxesSubplot at 0x7f846ae888d0>
```



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
```

The majority of users pay the plan's list price.

```
In [63]: (df_trans['plan_list_price']==df_trans['actual_amount_paid']).value_c
ounts()

Out[63]: True     1419106
         False      11903
         dtype: int64
```

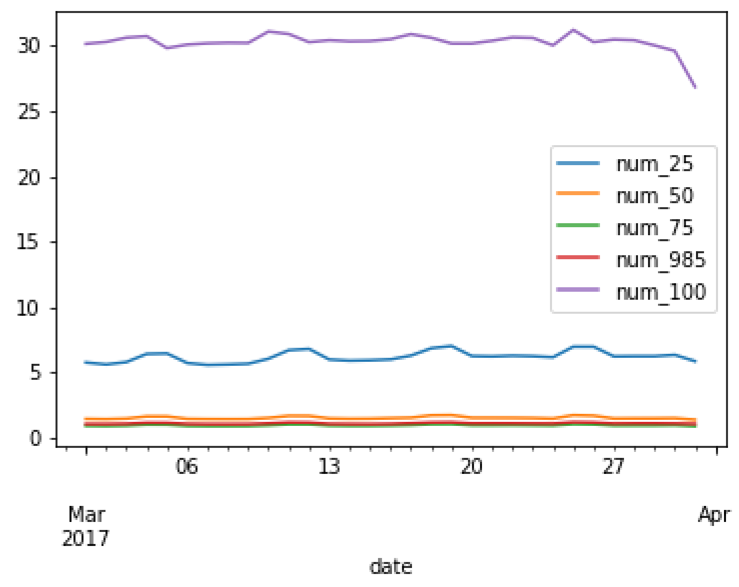
What is their listening behavior?

On average, users listen to more songs played over 100% of the song length than to songs played less than 100% of the song's length.

```
In [31]: num_listened_plot = df_logs.groupby('date')['num_25','num_50','num_7
5','num_985','num_100'].agg('mean')
```

```
In [32]: num_listened_plot.plot()
```

```
Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x7f73eb169908>
```



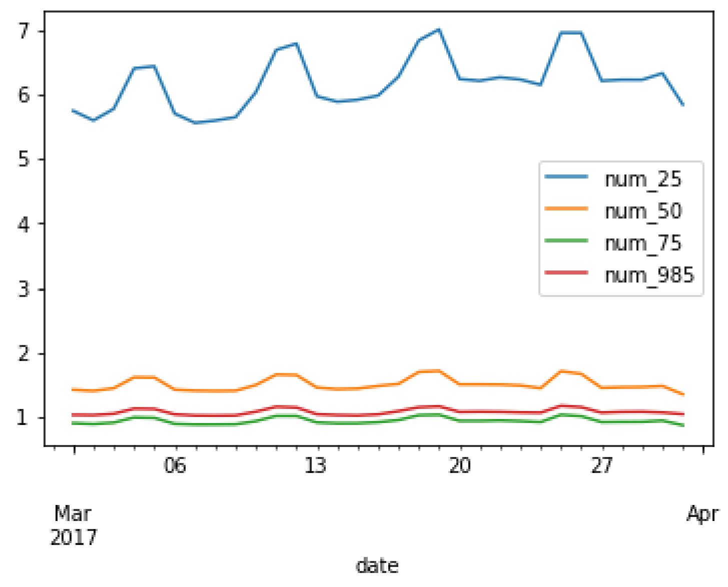
After songs played up to 100% of the song length, on average, users listen daily to more songs played less than 25% of the song length, than songs played at other lengths.

This could be because users are interested in trying out songs that they are not familiar with. The daily average number of songs played between 25% and 50%, between 50% and 75%, and between 75% and 98.5% of the length are very similar.

```
In [33]: num_listened_plot_2 = df_logs.groupby('date')['num_25','num_50','num_
         75','num_985'].agg('mean')
```

```
In [34]: num_listened_plot_2.plot()
```

```
Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x7f741f12b2b0>
```

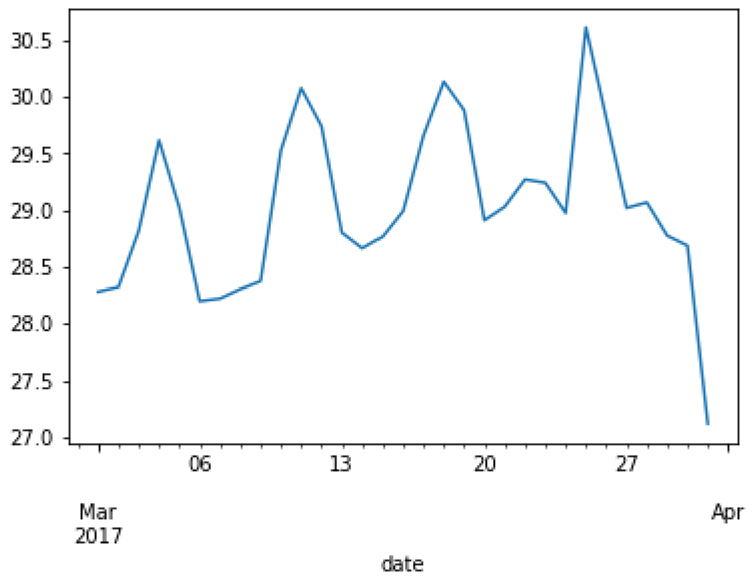


The daily average number of unique songs listened to fluctuates throughout the month, peaking on the weekends.

```
In [14]: num_unq_plot = df_logs.groupby('date')['num_unq'].agg('mean')
```

```
In [27]: num_unq_plot.plot()
```

```
Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x7f73bce777f0>
```



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

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

```
Out[32]: 29.036145516162382
```

Inferential Statistics

```
In [119]: all_data.set_index('msno',inplace=True)
```

```
In [29]: def one_hot(column,df):
             df_dummies = pd.get_dummies(df[column])
             del df_dummies[df_dummies.columns[-1]]
             df_new = pd.concat([df, df_dummies], axis=1)
             del df_new[column]
             return df_new
```

```
In [28]: all_data.drop('Unnamed: 0',axis=1,inplace=True)
```