The client

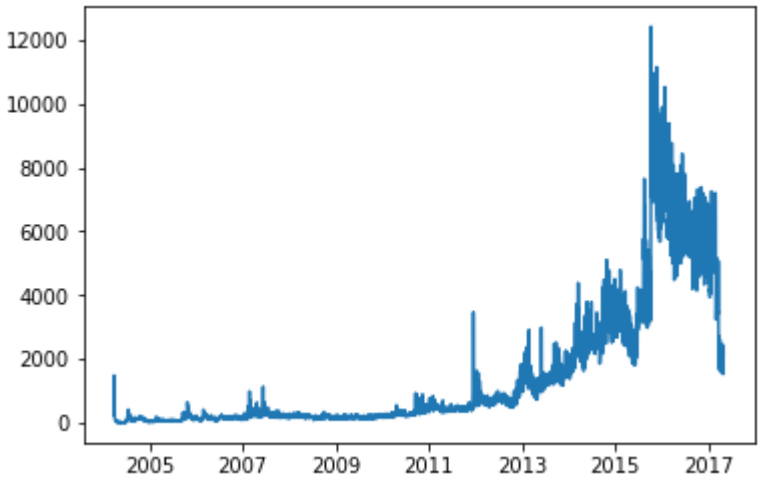
KKBox is a streaming music service in Asia. They have both paid and free subscriptions. The paid subscriptions subsidize other services that they offer, so KKBox must predict the churn of their paid users so that they can continue offering the free subscriptions.

If KKBox can learn why users leave, they can use these insights to try and keep the customers subscribing.

KKBox started in 2005 and 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. This makes keeping existing users even more important.

```
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
```

and only a small percentage of users in the data set canceled their subscriptions, 2.5%.

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

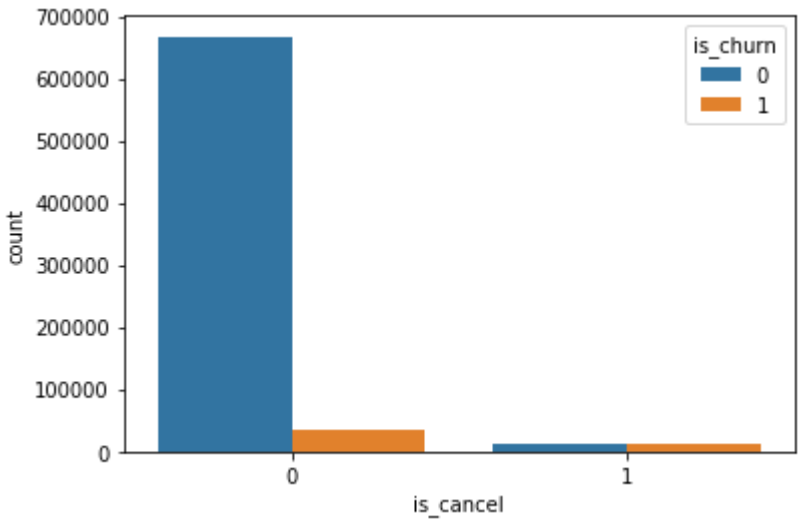
```
Out[52]: 0.02455120827332323
```

People who haven't canceled are much less likely to churn than those who have canceled, so KKBox can work on preventing users from canceling their subscriptions.

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])
```

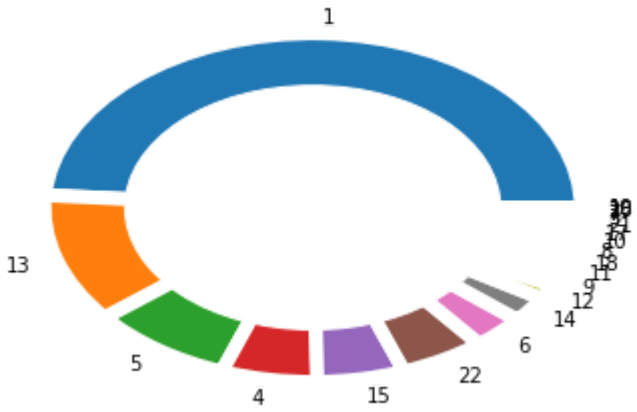
Out[83]: `<matplotlib.axes._subplots.AxesSubplot at 0x7f846ae888d0>`



In addition to whether or not a user cancels their subscription at some point, demographic factors might influence whether a user will churn.

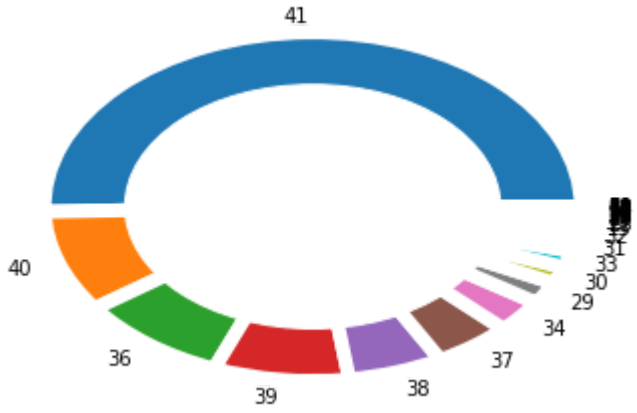
There are roughly 20 cities that users of KKBox live in. Users mostly come from city 1. Using a one way ANOVA, at a confidence level of 0.01, we reject the null hypothesis that the proportion of churners who are in cities 1, 3, and 4 are the same.

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()
```



The users can use different methods (encoded as numbers) to pay for their subscriptions. Half of the users use payment method id 41. Payment method 32 is the fifth most highly correlated feature with whether or not a user churns.

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()
```

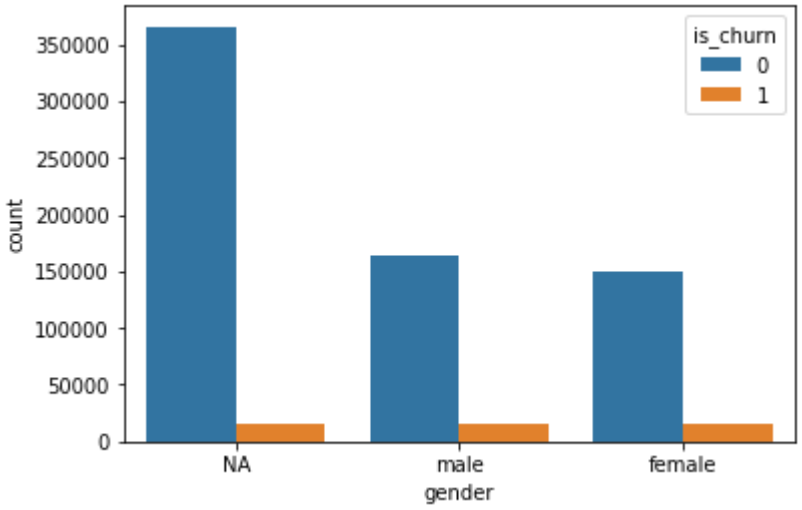


About half of the users in the data set did not provide their gender. However, people who don't list their gender are less likely to churn. Additionally, using a t-test at a confidence level of 0.01, we reject the null hypothesis that the proportion of female and male churners is the same.

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])
```

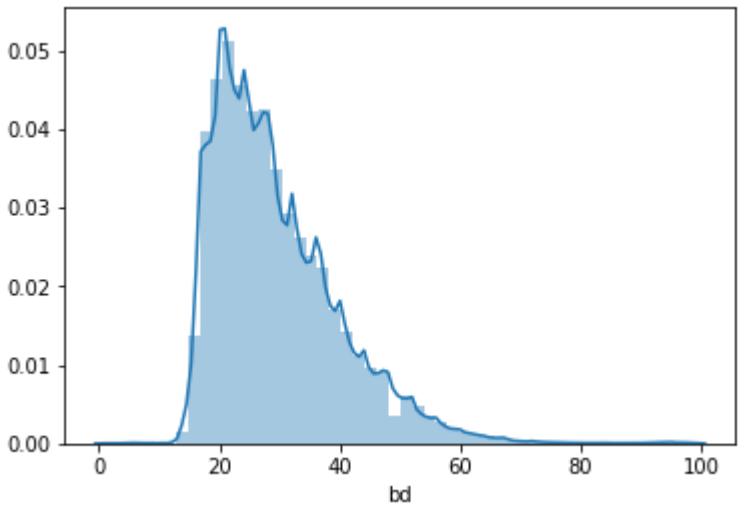
Out[41]: <matplotlib.axes._subplots.AxesSubplot at 0x7f741f030470>



Plotting only sensible values for the age, shows that most of the users are teenagers and young adults. The distribution of ages peaks around 25, and then decreases. We might expect that age is a good predictor of churners. Since most of the users are teenagers/young adults, they might churn more since they usually have less money and might be unable to afford the subscription, or change to other music streaming services. However, age does not correlate very highly with churn.

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

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



Looking at the mean of is_auto_renew, we see that the majority of users, 78%, have their plans set to automatically renew. This seems like a feature that would be predictive of whether a customer churns. If a customer needs to renew their subscription manually, they might reflect upon whether or not they actually need the subscription. Indeed, is_auto_renew is the fifth most highly correlated feature with churn.

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

```
Out[26]: 0.7853025382789347
```

Fianlly, the majority of users pay the plan's list price. Customers might be incentivized not to churn if they are given a discount.

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

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

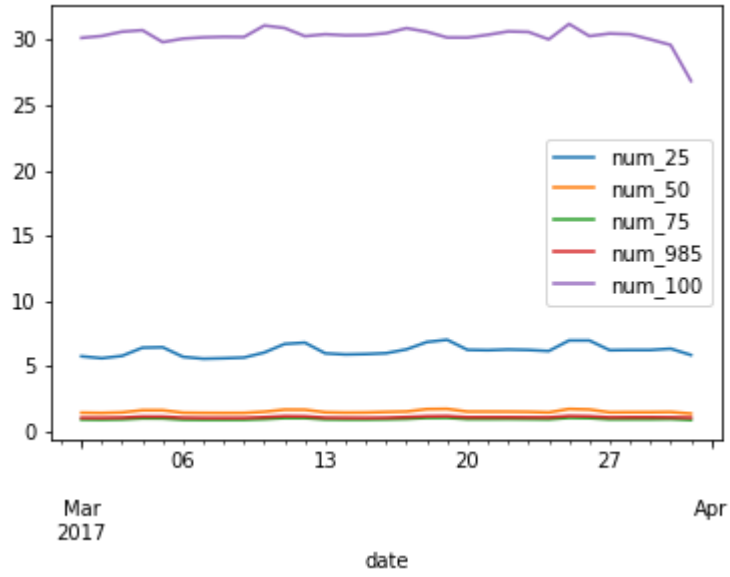
Customers' listening behavior could also affect whether or not they churn. If a customer isn't using the service, they will probably cancel their subscription.

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. This suggests that users are liking the songs that they listen to.

```
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>



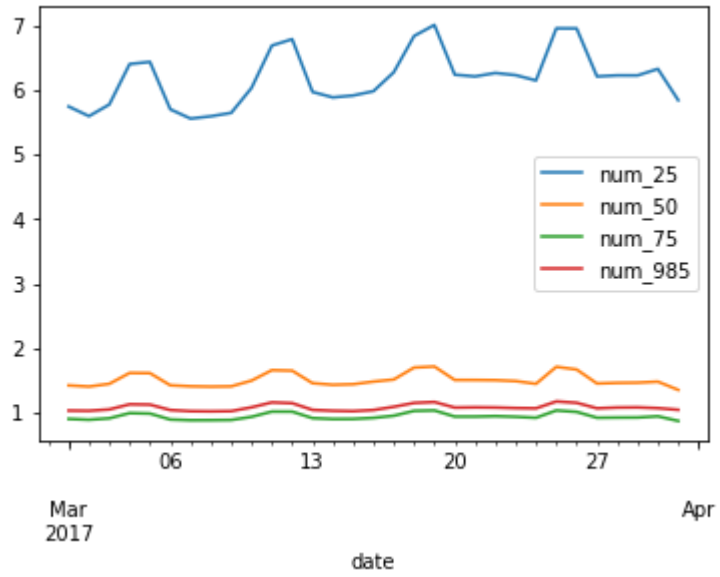
After songs played up to 100% of the song length, on average, users listen daily to more songs played less than 25% of the song length, than songs played at other lengths.

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, around 1-2 songs.

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>

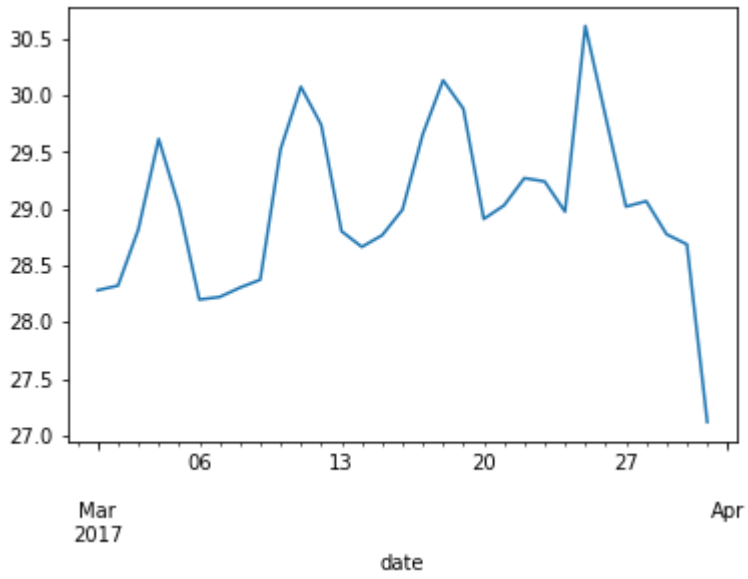


The daily average number of unique songs listened to fluctuates throughout the month, peaking on the weekends, and is around 28-30 songs. If a song's length is 3 minutes long, this means that users are listening to about 90 minutes of new music a day, so it seems that they are finding new music that they like.

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 [46]:
```
# 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
```

all_data = one_hot('payment_method_id',all_data) all_data = one_hot('city',all_data) all_data = one_hot('gender',all_data) all_data = one_hot('registered_via',all_data) all_data = one_hot('registration_init_time',all_data) all_data = one_hot('year',all_data)