In [1]:
```
# Airbnb data exploration
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt


# Draw inline
%matplotlib inline
```

In [20]:
```
# Set figure aesthetics
import seaborn as sns
sns.set_style("white", {'ytick.major.size': 10.0})
sns.set_context("poster", font_scale=1.1)
```

```
C:\Anaconda3\lib\site-packages\matplotlib\__init__.py:872: UserWarning: axes.col
or_cycle is deprecated and replaced with axes.prop_cycle; please use the latter.
  warnings.warn(self.msg_depr % (key, alt_key))
```

In [2]:
```
# Load the data into DataFrames
train_users = pd.read_csv('train_users.csv')
test_users = pd.read_csv('test_users.csv')
```

In [12]:
```
# Merge train and test users
users = pd.concat((train_users, test_users), axis=0, ignore_index=True)

# Remove ID's since now we are not interested in making predictions
users.drop('id',axis=1, inplace=True)

users.head()
```

Out[12]:

	affiliate_channel	affiliate_provider	age	country_destination	date_account_created	date_first_bo
0	direct	direct	NaN	NDF	2010-06-28	NaN
1	seo	google	38	NDF	2011-05-25	NaN
2	direct	direct	56	US	2010-09-28	2010-08-02
3	direct	direct	42	other	2011-12-05	2012-09-08
4	direct	direct	41	US	2010-09-14	2010-02-18

In [13]:
```
# Missing Data
users.gender.replace('-unknown-', np.nan, inplace=True)
```

In [14]:
```
users_nan = (users.isnull().sum() / users.shape[0]) * 100
users_nan[users_nan > 0].drop('country_destination')
```

Out[14]:
```
age                        42.412365
date_first_booking         67.733998
first_affiliate_tracked     2.208335
gender                     46.990169
dtype: float64
```

In [18]:
```
users.age.describe()
print(sum(users.age > 122))
print(sum(users.age < 18))
```

```
830
188
```

In [27]:
```
users[users.age < 14]["age"].describe()
```

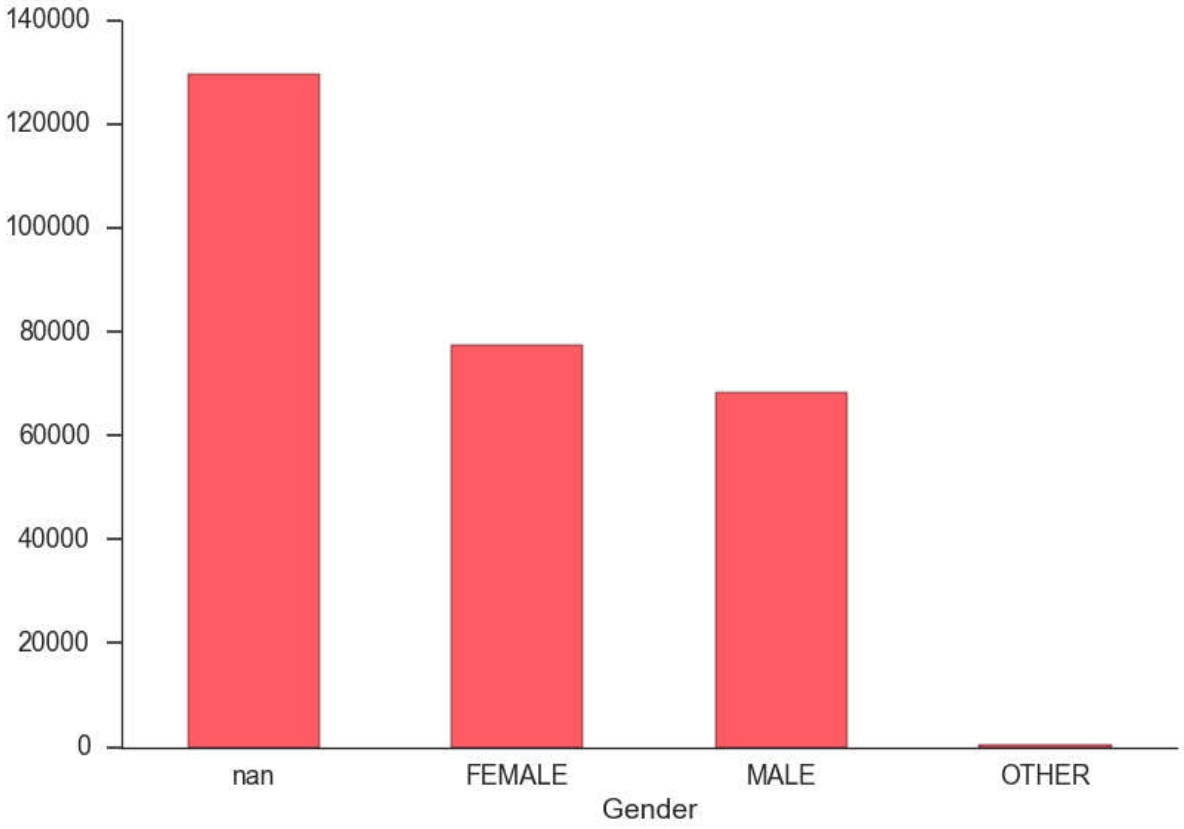
Out[27]:
```
count    59.000000
mean      4.322034
std       1.331847
min       1.000000
25%       5.000000
50%       5.000000
75%       5.000000
max       5.000000
Name: age, dtype: float64
```

In [28]:
```
users.loc[users.age > 95, 'age'] = np.nan
users.loc[users.age < 14, 'age'] = np.nan
```

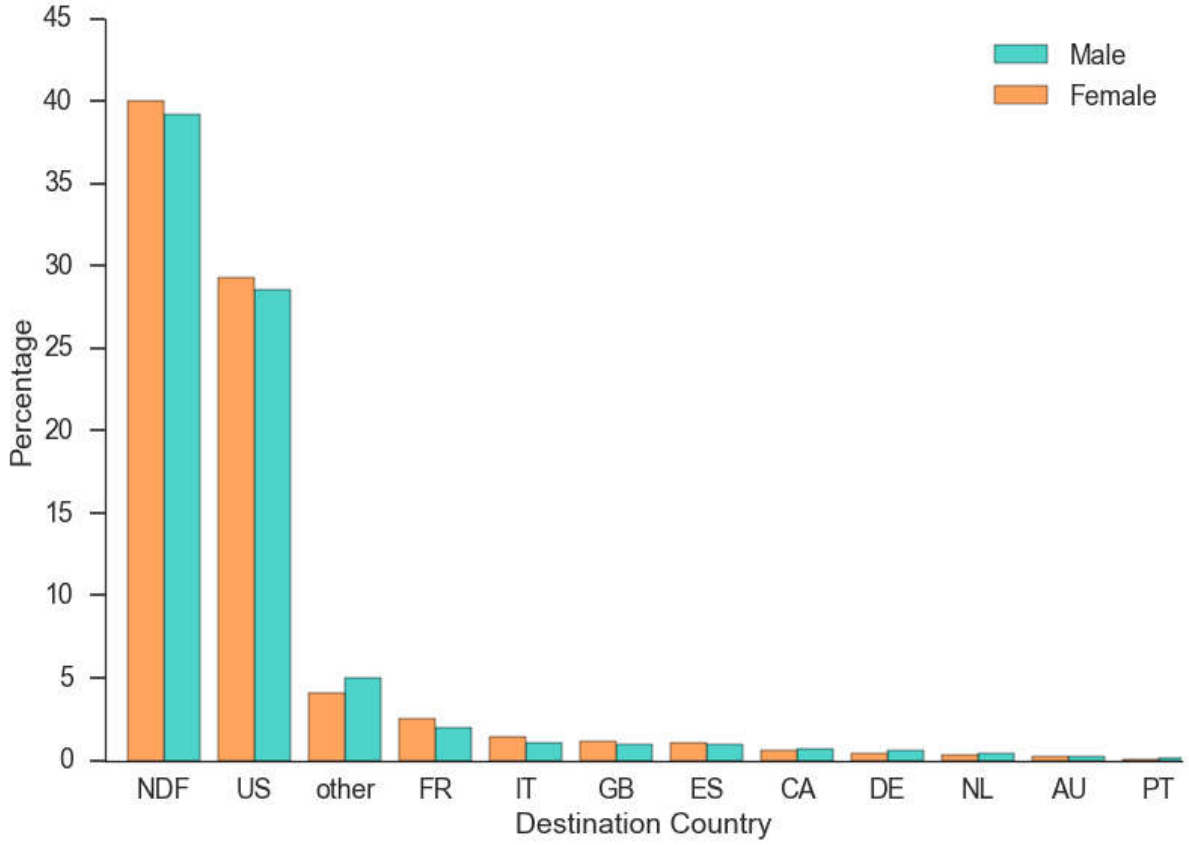
In [29]:
```
categorical_features = [
    'affiliate_channel',
    'affiliate_provider',
    'country_destination',
    'first_affiliate_tracked',
    'first_browser',
    'first_device_type',
    'gender',
    'language',
    'signup_app',
    'signup_method'
]

for categorical_feature in categorical_features:
    users[categorical_feature] = users[categorical_feature].astype('category')
```

In [30]:
```
# formatting date
users['date_account_created'] = pd.to_datetime(users['date_account_created'])
users['date_first_booking'] = pd.to_datetime(users['date_first_booking'])
users['date_first_active'] = pd.to_datetime((users.timestamp_first_active // 100000
0), format='%Y%m%d')
```

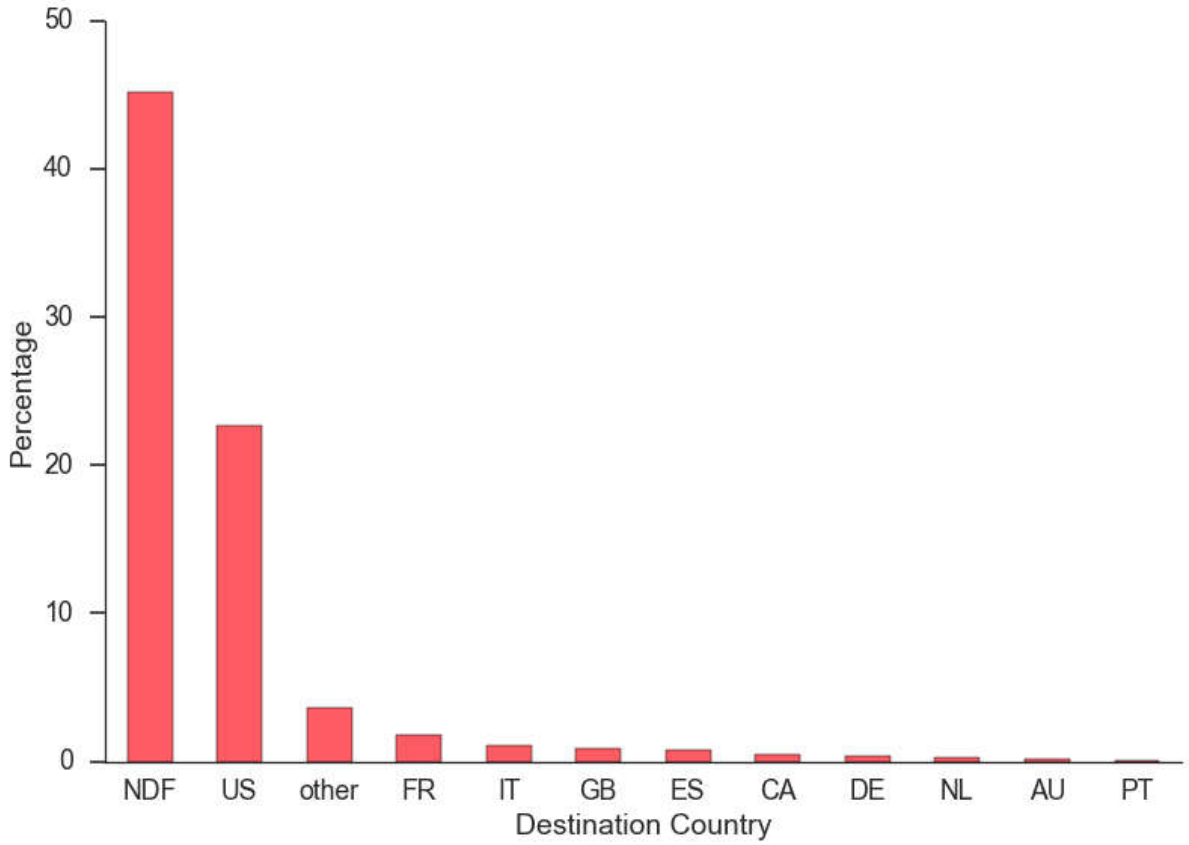
In [31]:
```
# Graph by Gender
users.gender.value_counts(dropna=False).plot(kind='bar', color='#FD5C64', rot=0)
plt.xlabel('Gender')
sns.despine()
```



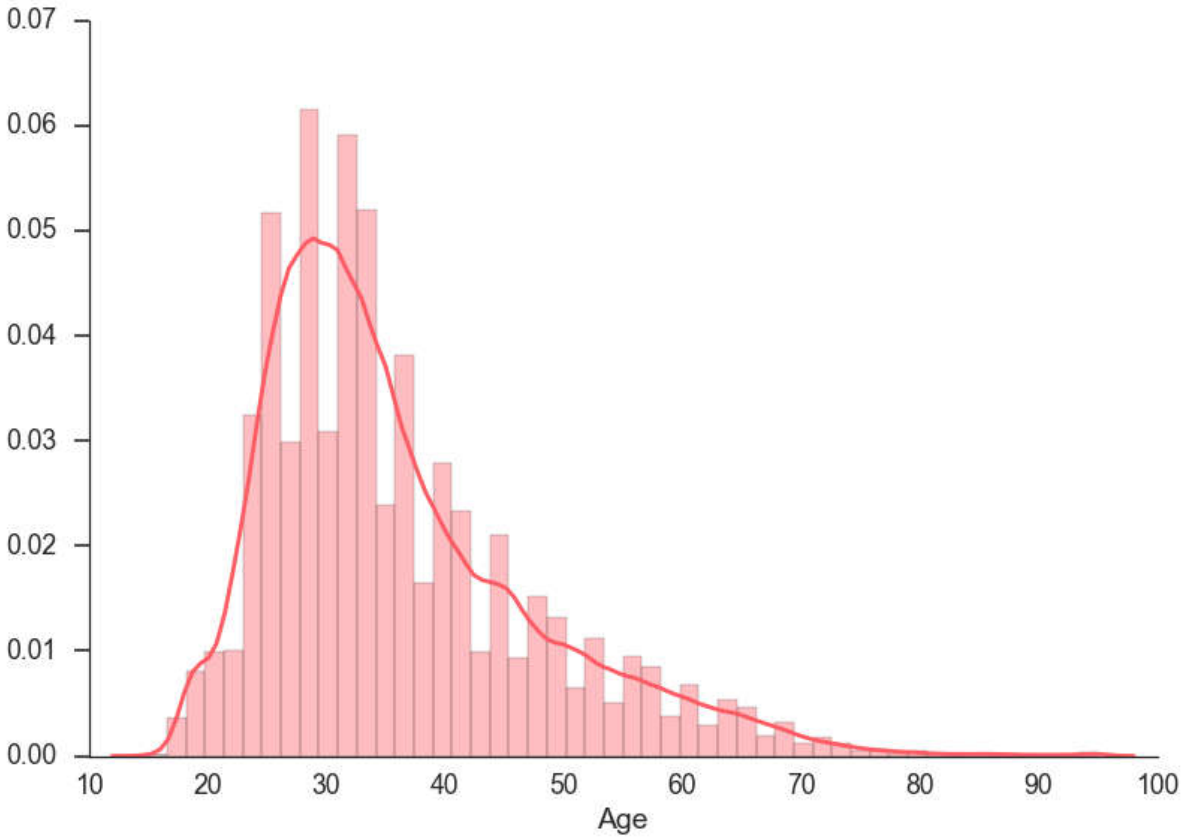
In [32]:
```
women = sum(users['gender'] == 'FEMALE')
men = sum(users['gender'] == 'MALE')

female_destinations = users.loc[users['gender'] == 'FEMALE', 'country_destination']
.value_counts() / women * 100
male_destinations = users.loc[users['gender'] == 'MALE', 'country_destination'].val
ue_counts() / men * 100

# Bar width
width = 0.4

male_destinations.plot(kind='bar', width=width, color='#4DD3C9', position=0, label=
'Male', rot=0)
female_destinations.plot(kind='bar', width=width, color='#FFA35D', position=1, labe
l='Female', rot=0)

plt.legend()
plt.xlabel('Destination Country')
plt.ylabel('Percentage')

sns.despine()
plt.show()
```



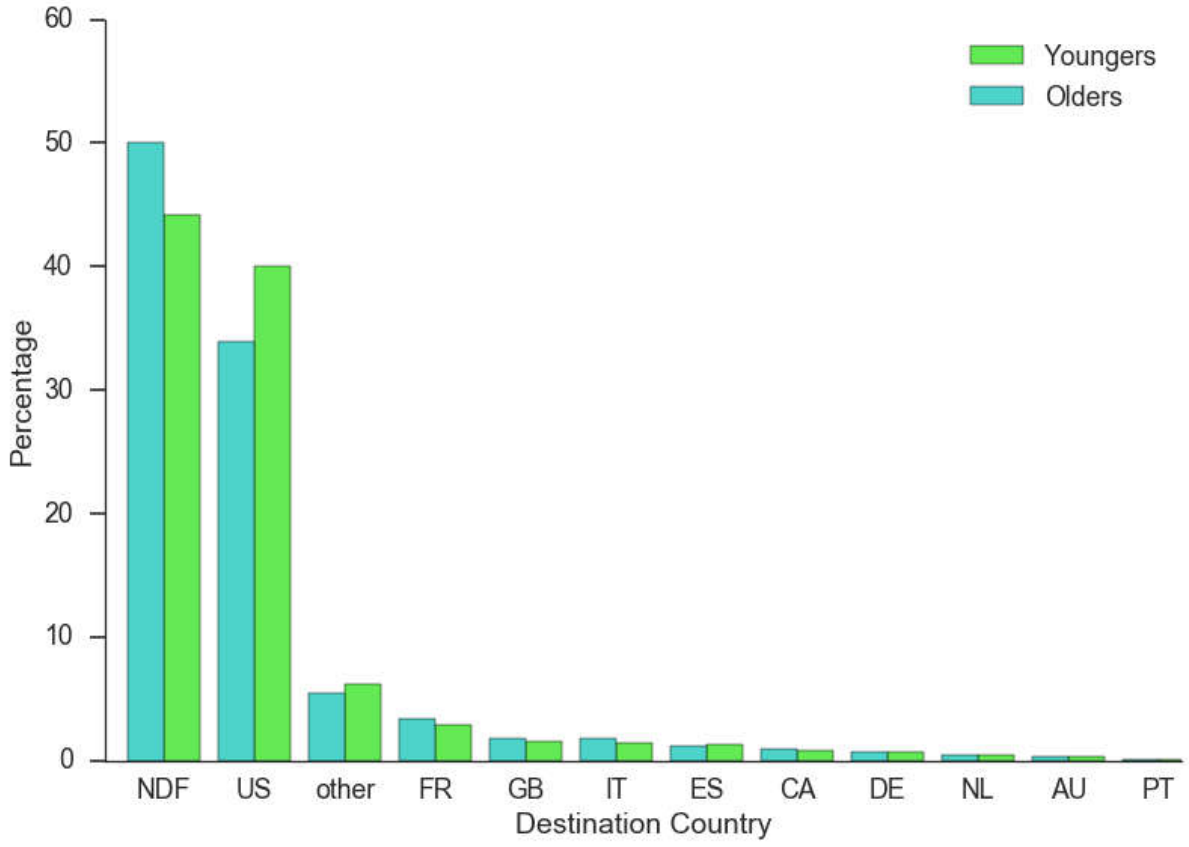
In [33]:
```
destination_percentage = users.country_destination.value_counts() / users.shape[0]
* 100
destination_percentage.plot(kind='bar',color='#FD5C64', rot=0)
# Using seaborn can also be plotted
# sns.countplot(x="country_destination", data=users, order=list(users.country_desti
nation.value_counts().keys()))
plt.xlabel('Destination Country')
plt.ylabel('Percentage')
sns.despine()
#The first thing we can see that if there is a reservation, it's likely to be insid
e the US.
#But there is a 45% of people that never did a reservation.
```



In [34]:
```
sns.distplot(users.age.dropna(), color='#FD5C64')
plt.xlabel('Age')
sns.despine()
# the common age to travel is between 25 and 40.
```



In [35]:
```
# Let's see if, for example, older people travel in a different way.
#Let's pick an arbitrary age to split into two groups. Maybe 45?
age = 45

younger = sum(users.loc[users['age'] < age, 'country_destination'].value_counts())
older = sum(users.loc[users['age'] > age, 'country_destination'].value_counts())

younger_destinations = users.loc[users['age'] < age, 'country_destination'].value_c
ounts() / younger * 100
older_destinations = users.loc[users['age'] > age, 'country_destination'].value_cou
nts() / older * 100

younger_destinations.plot(kind='bar', width=width, color='#63EA55', position=0, lab
el='Youngers', rot=0)
older_destinations.plot(kind='bar', width=width, color='#4DD3C9', position=1, label
='Olders', rot=0)

plt.legend()
plt.xlabel('Destination Country')
plt.ylabel('Percentage')

sns.despine()
plt.show()
#We can see that the young people tends to stay in the US, and the older people cho
ose to travel outside the country
```

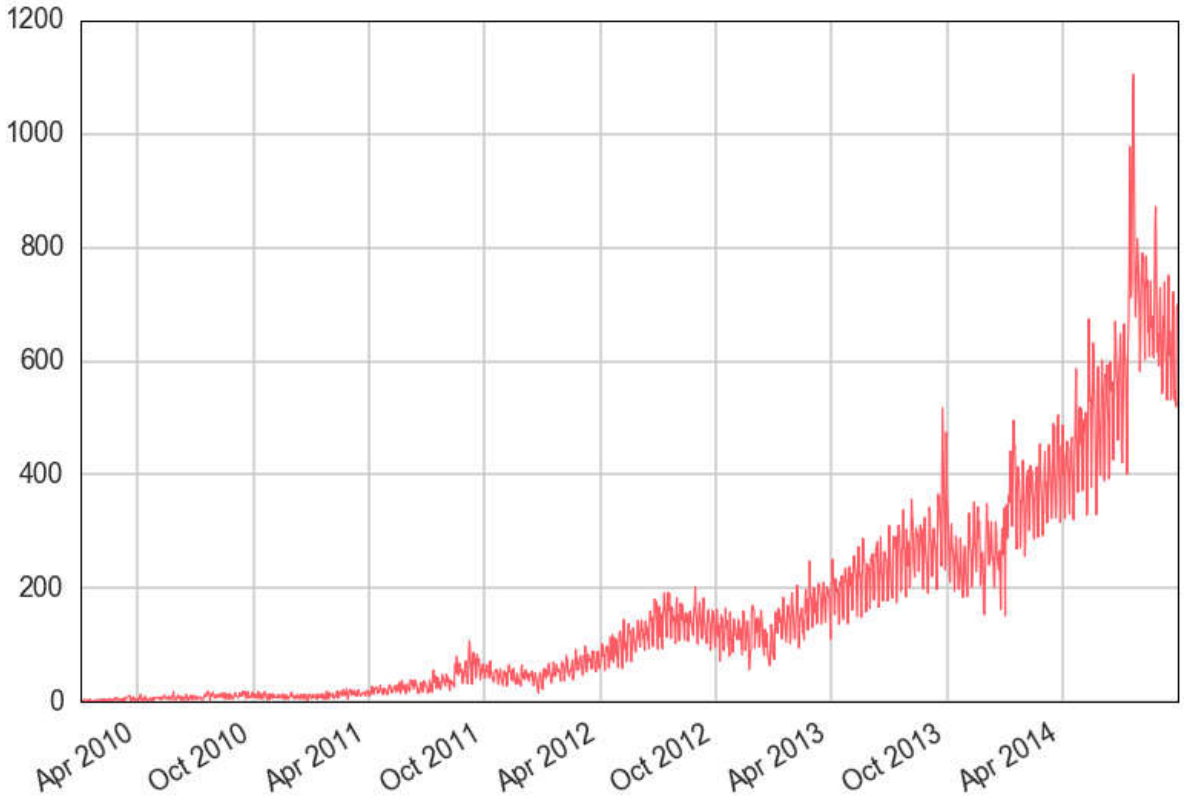


In [36]:
```
print((sum(users.language == 'en') / users.shape[0])*100)
# With the 96% of users using English as their language, it is understandable that
a lot of people stay in the US.
```

```
96.3675888324
```

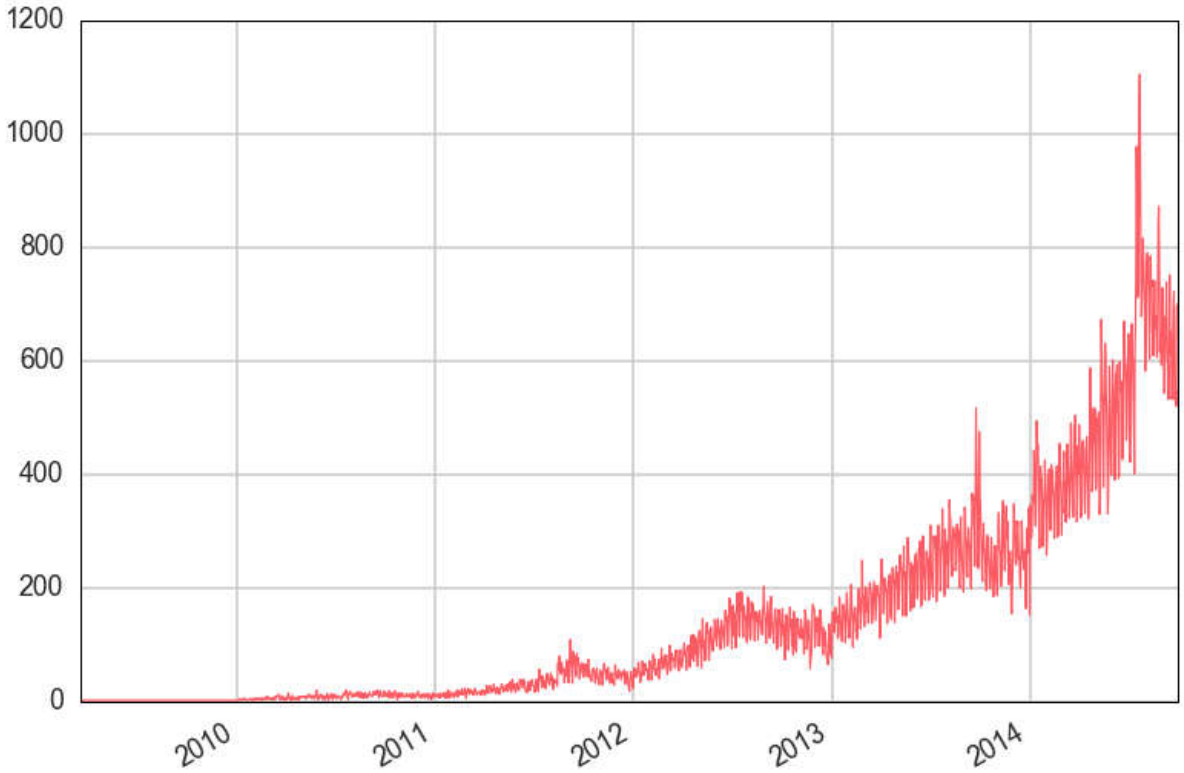
In [37]:
```
sns.set_style("whitegrid", {'axes.edgecolor': '0'})
sns.set_context("poster", font_scale=1.1)
users.date_account_created.value_counts().plot(kind='line', linewidth=1.2, color='#
FD5C64')
```

Out[37]: <matplotlib.axes._subplots.AxesSubplot at 0x1fef0de400>

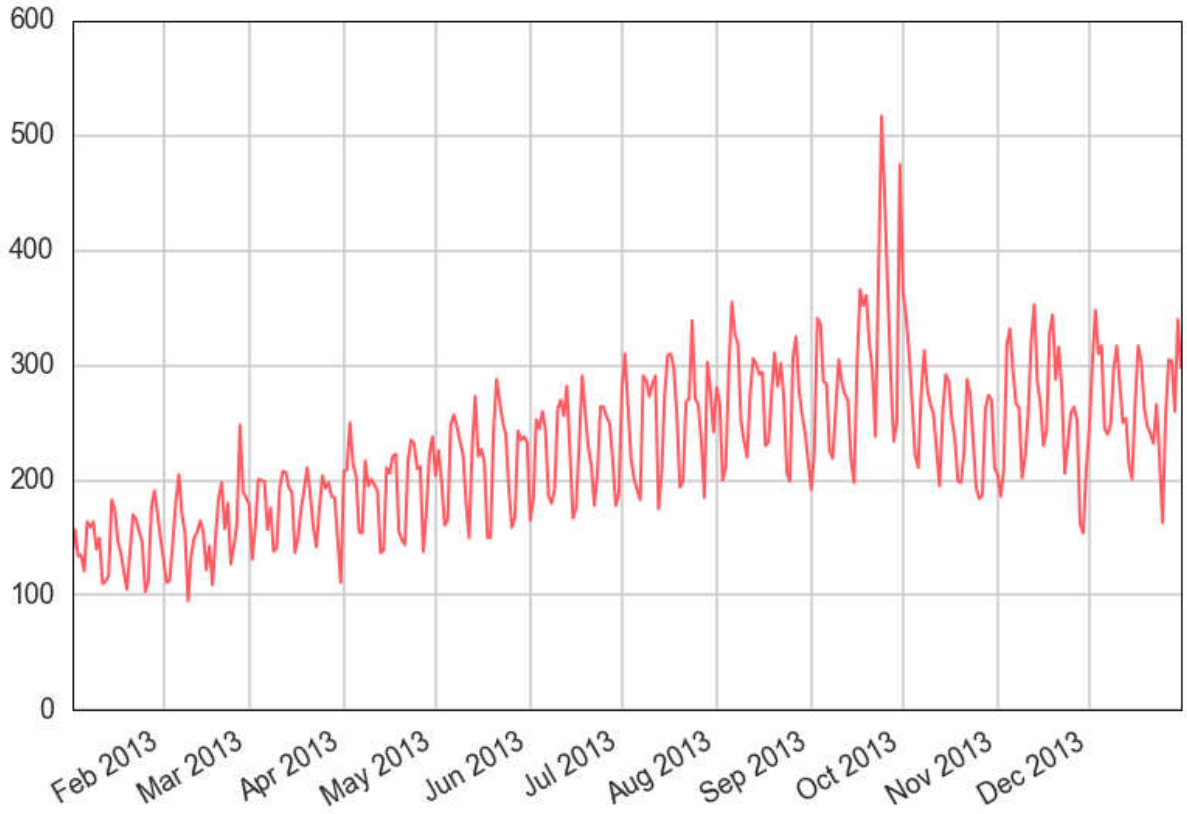


In [38]:
```
users.date_first_active.value_counts().plot(kind='line', linewidth=1.2, color='#FD5C64')
```

Out[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1ff3c68f98>
```

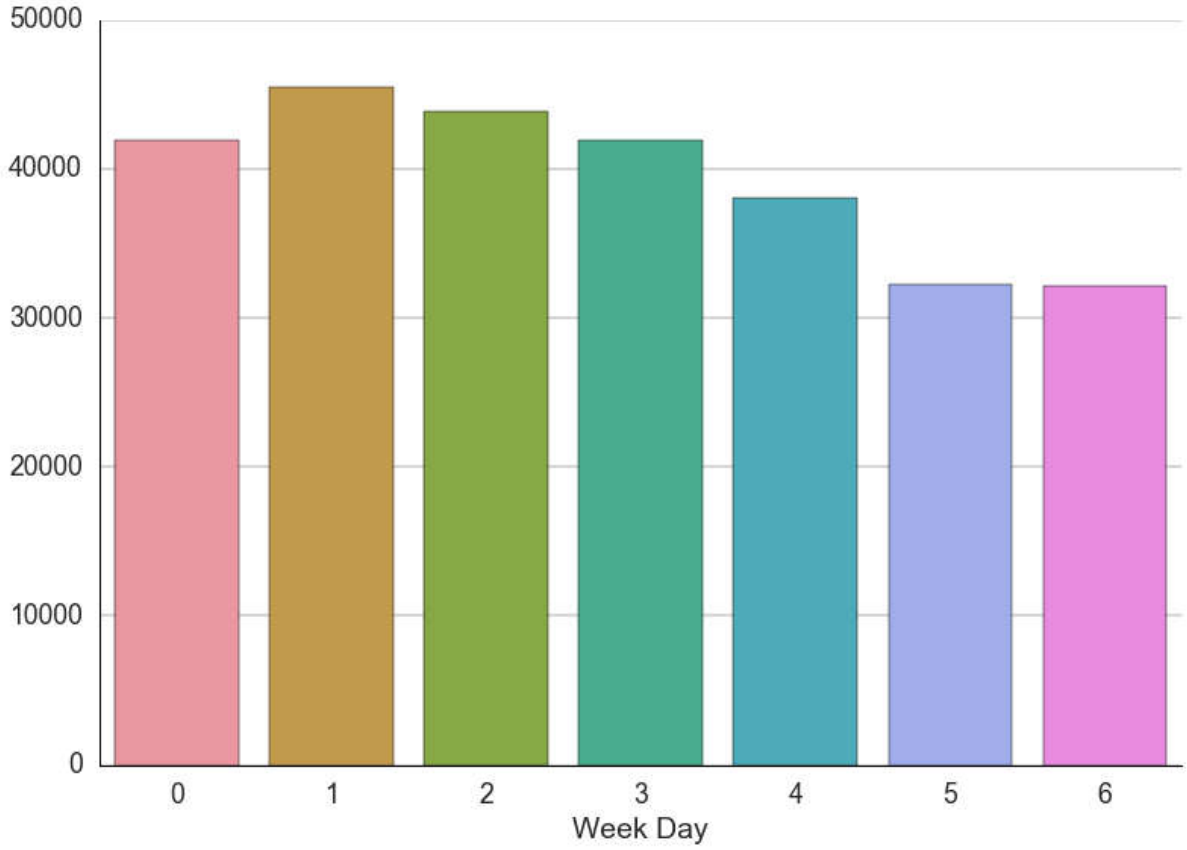


In [39]:
```
users_2013 = users[users['date_first_active'] > pd.to_datetime(20130101, format='%Y
%m%d')]
users_2013 = users_2013[users_2013['date_first_active'] < pd.to_datetime(20140101,
format='%Y%m%d')]
users_2013.date_first_active.value_counts().plot(kind='line', linewidth=2, color='#
FD5C64')
plt.show()
```



In [40]:
```
weekdays = []
for date in users.date_account_created:
    weekdays.append(date.weekday())
weekdays = pd.Series(weekdays)
sns.barplot(x = weekdays.value_counts().index, y=weekdays.value_counts().values, or
der=range(0,7))
plt.xlabel('Week Day')
sns.despine()
```

```
C:\Anaconda3\lib\site-packages\matplotlib\__init__.py:892: UserWarning: axes.col
or_cycle is deprecated and replaced with axes.prop_cycle; please use the latter.
  warnings.warn(self.msg_depr % (key, alt_key))
```



In [41]:
```
date = pd.to_datetime(20140101, format='%Y%m%d')

before = sum(users.loc[users['date_first_active'] < date, 'country_destination'].va
lue_counts())
after = sum(users.loc[users['date_first_active'] > date, 'country_destination'].val
ue_counts())
before_destinations = users.loc[users['date_first_active'] < date,
                                'country_destination'].value_counts() / before * 10
0
after_destinations = users.loc[users['date_first_active'] > date,
                               'country_destination'].value_counts() / after * 100
before_destinations.plot(kind='bar', width=width, color='#63EA55', position=0, labe
l='Before 2014', rot=0)
after_destinations.plot(kind='bar', width=width, color='#4DD3C9', position=1, label
='After 2014', rot=0)

plt.legend()
plt.xlabel('Destination Country')
plt.ylabel('Percentage')

sns.despine()
plt.show()
```