

## Libraries and dataset

```
In [1]: import numpy as np
import pandas as pd
pd.options.mode.chained_assignment = None # default='warn'
#libraries used in visualization
import plotly.express as px
from ipywidgets import interact, interactive, fixed, interact_manual
import ipywidgets as widgets
```

```
In [2]: train_users = pd.read_csv('train_users_2.csv')
```

```
In [3]: print("We have", train_users.shape[0], "users")
```

We have 213451 users

## Check Duplication

```
In [4]: train_users.duplicated().sum()
```

Out[4]: 0

## Check Missing values

```
In [5]: train_users.isna().sum()
```

```
Out[5]: id                                0
date_account_created                     0
timestamp_first_active                   0
date_first_booking                      124543
gender                                   0
age                                      87990
signup_method                           0
signup_flow                             0
language                                 0
affiliate_channel                        0
affiliate_provider                       0
first_affiliate_tracked                  6065
signup_app                              0
first_device_type                        0
first_browser                           0
country_destination                     0
dtype: int64
```

```
In [6]: print((train_users['date_first_booking'].isna().sum()*100 ) /train_users.shape[0])
print((train_users['age'].isna().sum()*100 ) /train_users.shape[0])
print((train_users['first_affiliate_tracked'].isna().sum()*100 ) /train_users.shape[0] )
```

```
58.347349040294965
41.222575673105304
2.84140153946339
```

## first\_affiliate\_tracked

```
In [7]: train_users['first_affiliate_tracked'].describe()
```

```
Out[7]: count          207386
unique              7
top      untracked
freq          109232
Name: first_affiliate_tracked, dtype: object
```

**Fill the null values in this column by #untracked because it is the most repeated (Mode)**

```
In [8]: train_users['first_affiliate_tracked'].fillna("untracked", inplace = True)
```

## Age

**Almost 41% from data has null values**

```
In [9]: train_users.age.value_counts()
```

```
Out[9]: 30.0      6124
        31.0      6016
        29.0      5963
        28.0      5939
        32.0      5855
        ...
        1925.0      1
        2008.0      1
        1995.0      1
        1952.0      1
        1942.0      1
        Name: age, Length: 127, dtype: int64
```

**alot of users put enter the year of birth instead of their age ==> change by their age now**

```
In [10]: train_users['age'] = train_users['age'].apply(lambda x: 2022-x if x > 122 else x)
```

```
In [11]: train_users['age'].value_counts()
```

```
Out[11]: 30.0      6124
          31.0      6016
          29.0      5963
          28.0      5939
          32.0      5855
          ...
          1.0         2
        1872.0         1
        1890.0         1
          14.0         1
         112.0         1
          Name: age, Length: 109, dtype: int64
```

```
In [12]: train_users.age.isna().sum()
```

```
Out[12]: 87990
```

## Fill the null values by the mean age based on gender kind

```
In [13]: male_data=train_users[(train_users['gender']=='MALE')]
          train_users['age'].loc[train_users['age'].isnull() & (train_users['gender']=='MALE')]=male_data['age'].mean()
          male_data['age'].mean()
```

```
Out[13]: 37.27762101150423
```

```
In [14]: female_data=train_users[(train_users['gender']=='FEMALE')]
          train_users['age'].loc[train_users['age'].isnull() & (train_users['gender']=='FEMALE')]=female_data['age'].mean()
          female_data['age'].mean()
```

```
Out[14]: 36.65766811636893
```

## change any value of gender to unknown except male and female

```
In [15]: train_users.gender.replace('OTHER', '-unknown-', inplace=True)
train_users.gender.value_counts()
```

```
Out[15]: -unknown-    95970
FEMALE    63041
MALE      54440
Name: gender, dtype: int64
```

```
In [16]: unknown_data=train_users[(train_users['gender'] == '-unknown-')]
train_users['age'].loc[train_users['age'].isnull() & (train_users['gender']=='-unknown-')]=unknown_data['age'].mean()
unknown_data['age'].mean()
```

```
Out[16]: 39.366469451115925
```

## Calculate Z- Score to detect the outliers of ages and remove them

```
In [17]: train_users['age_z-score'] = (train_users['age'] - train_users['age'].mean())/train_users['age'].std()
```

```
In [18]: train_users.head(3)
```

```
Out[18]:
```

	id	date_account_created	timestamp_first_active	date_first_booking	gender	age	signup_method	signup_flow	language
0	gxn3p5htnn	2010-06-28	20090319043255	NaN	unknown-	39.366469	facebook	0	
1	820tgsjxq7	2011-05-25	20090523174809	NaN	MALE	38.000000	facebook	0	
2	4ft3gnwmtx	2010-09-28	20090609231247	2010-08-02	FEMALE	56.000000	basic	3	

```
In [19]: clean_df =train_users[(train_users['age_z-score'] <=3) & (train_users['age_z-score'] >=-3)]
clean_df.shape[0]
```

```
Out[19]: 211054
```

```
In [20]: clean_df.isna().sum()
```

```
Out[20]: id                                0
date_account_created                      0
timestamp_first_active                    0
date_first_booking                        123402
gender                                    0
age                                        0
signup_method                             0
signup_flow                               0
language                                  0
affiliate_channel                         0
affiliate_provider                        0
first_affiliate_tracked                   0
signup_app                                0
first_device_type                         0
first_browser                             0
country_destination                      0
age_z-score                              0
dtype: int64
```

## Split date\_first\_booking column then fill nul values by the mode of each column

```
In [21]: clean_df['date_first_booking']
```

```
Out[21]: 0          NaN
1          NaN
2    2010-08-02
3    2012-09-08
4    2010-02-18
...
213446      NaN
213447      NaN
213448      NaN
213449      NaN
213450      NaN
Name: date_first_booking, Length: 211054, dtype: object
```

```
In [22]: clean_df['dfb_day'] = clean_df['date_first_booking'].str.split('-').str[-1]
clean_df['dfb_month'] = clean_df['date_first_booking'].str.split('-').str[-2]
clean_df['dfb_year'] = clean_df['date_first_booking'].str.split('-').str[-3]
```

```
In [23]: clean_df = clean_df.drop(['date_first_booking'], axis=1)
```

```
In [24]: clean_df['dfb_day'].fillna(clean_df['dfb_day'].mode()[0], inplace = True)
clean_df['dfb_month'].fillna(clean_df['dfb_month'].mode()[0], inplace = True)
clean_df['dfb_year'].fillna(clean_df['dfb_year'].mode()[0], inplace = True)
```

```
In [25]: clean_df.isna().sum()
```

```
Out[25]: id                0
date_account_created      0
timestamp_first_active    0
gender                    0
age                       0
signup_method             0
signup_flow               0
language                  0
affiliate_channel         0
affiliate_provider        0
first_affiliate_tracked   0
signup_app                0
first_device_type         0
first_browser             0
country_destination       0
age_z-score               0
dfb_day                   0
dfb_month                 0
dfb_year                  0
dtype: int64
```

```
In [26]: clean_df['timestamp_first_active'] = pd.to_datetime(clean_df['timestamp_first_active'], format='%Y%m%d%H%M%S')
clean_df.head(3)
```

Out[26]:

	id	date_account_created	timestamp_first_active	gender	age	signup_method	signup_flow	language	affiliate_channel
0	gxn3p5htnn	2010-06-28	2009-03-19 04:32:55	unknown-	39.366469	facebook	0	en	direct
1	820tgsjq7	2011-05-25	2009-05-23 17:48:09	MALE	38.000000	facebook	0	en	seo
2	4ft3gnwmtx	2010-09-28	2009-06-09 23:12:47	FEMALE	56.000000	basic	3	en	direct



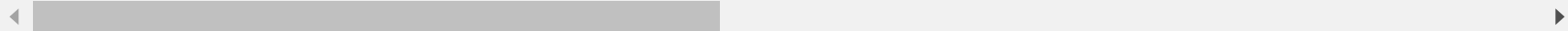


```
In [27]: clean_df['first_active_date'] = pd.to_datetime(clean_df['timestamp_first_active']).dt.date
clean_df['first_active_time'] = pd.to_datetime(clean_df['timestamp_first_active']).dt.time
clean_df.drop('timestamp_first_active',axis=1, inplace=True)
clean_df
```

Out[27]:

	id	date_account_created	gender	age	signup_method	signup_flow	language	affiliate_channel	affiliate_provider
0	gxn3p5htnn	2010-06-28	unknown-	39.366469	facebook	0	en	direct	direct
1	820tgsjxq7	2011-05-25	MALE	38.000000	facebook	0	en	seo	google
2	4ft3gnwmtx	2010-09-28	FEMALE	56.000000	basic	3	en	direct	direct
3	bijt8pjhuk	2011-12-05	FEMALE	42.000000	facebook	0	en	direct	direct
4	87mebub9p4	2010-09-14	unknown-	41.000000	basic	0	en	direct	direct
...	...	...	...	...	...	...	...	...	...
213446	zxodksqep	2014-06-30	MALE	32.000000	basic	0	en	sem-brand	google
213447	mhewnxesx9	2014-06-30	unknown-	39.366469	basic	0	en	direct	direct
213448	6o3arsjbb4	2014-06-30	unknown-	32.000000	basic	0	en	direct	direct
213449	jh95kwisub	2014-06-30	unknown-	39.366469	basic	25	en	other	other
213450	nw9fwlyb5f	2014-06-30	unknown-	39.366469	basic	25	en	direct	direct

211054 rows × 20 columns



```
In [28]: ## save this data to time series
t_df = clean_df.to_csv('t_df.csv')
```

```
In [29]: #date_account_created
dac = np.vstack(clean_df.date_account_created.astype(str).apply(lambda x: list(map(int, x.split('-')))).values)
clean_df['dac_year'] = dac[:,0]
clean_df['dac_month'] = dac[:,1]
clean_df['dac_day'] = dac[:,2]
clean_df = clean_df.drop(['date_account_created'], axis=1)

tfa = np.vstack(clean_df.first_active_date.astype(str).apply(lambda x: list(map(int, x.split('-')))).values)
clean_df['tfa_year'] = tfa[:,0]
clean_df['tfa_month'] = tfa[:,1]
clean_df['tfa_day'] = tfa[:,2]
clean_df = clean_df.drop(['first_active_date'], axis=1)
```

## Change datatypes

In [30]: `clean_df.info()`

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 211054 entries, 0 to 213450
Data columns (total 24 columns):
#   Column                      Non-Null Count  Dtype
---  -
0   id                          211054 non-null object
1   gender                      211054 non-null object
2   age                         211054 non-null float64
3   signup_method              211054 non-null object
4   signup_flow                211054 non-null int64
5   language                   211054 non-null object
6   affiliate_channel          211054 non-null object
7   affiliate_provider          211054 non-null object
8   first_affiliate_tracked    211054 non-null object
9   signup_app                  211054 non-null object
10  first_device_type           211054 non-null object
11  first_browser               211054 non-null object
12  country_destination         211054 non-null object
13  age_z-score                 211054 non-null float64
14  dfb_day                     211054 non-null object
15  dfb_month                   211054 non-null object
16  dfb_year                    211054 non-null object
17  first_active_time           211054 non-null object
18  dac_year                    211054 non-null int32
19  dac_month                   211054 non-null int32
20  dac_day                     211054 non-null int32
21  tfa_year                    211054 non-null int32
22  tfa_month                   211054 non-null int32
23  tfa_day                     211054 non-null int32
dtypes: float64(2), int32(6), int64(1), object(15)
memory usage: 35.4+ MB
```

```
In [31]: clean_df['age'] = clean_df['age'].astype(int)
clean_df['dfb_day'] = clean_df['dfb_day'].astype(int)
clean_df['dfb_month'] = clean_df['dfb_month'].astype(int)
clean_df['dfb_year'] = clean_df['dfb_year'].astype(int)

for col in ['gender', 'signup_method', 'language', 'affiliate_channel', 'affiliate_provider', 'first_affiliate_trav',
            'signup_app', 'first_device_type', 'country_destination']:
    clean_df[col] = clean_df[col].astype('category')
```

```
In [32]: clean_df.drop('id',axis=1, inplace=True)
clean_df.drop('first_active_time',axis=1, inplace=True)
clean_df.drop('age_z-score',axis=1, inplace=True)
```

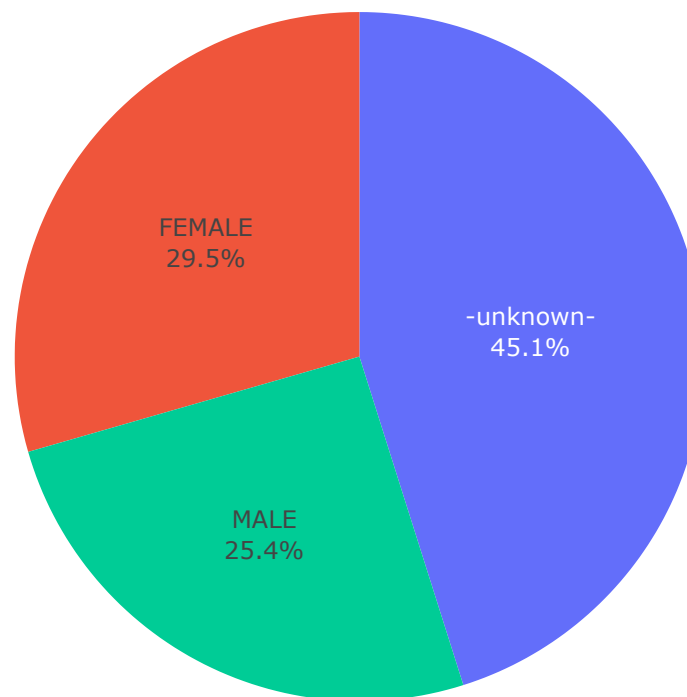
In [33]: `clean_df.info()`

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 211054 entries, 0 to 213450
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   gender                211054 non-null  category
1   age                  211054 non-null  int32
2   signup_method        211054 non-null  category
3   signup_flow          211054 non-null  int64
4   language             211054 non-null  category
5   affiliate_channel     211054 non-null  category
6   affiliate_provider    211054 non-null  category
7   first_affiliate_tracked 211054 non-null  category
8   signup_app           211054 non-null  category
9   first_device_type     211054 non-null  category
10  first_browser         211054 non-null  object
11  country_destination   211054 non-null  category
12  dfb_day              211054 non-null  int32
13  dfb_month            211054 non-null  int32
14  dfb_year             211054 non-null  int32
15  dac_year             211054 non-null  int32
16  dac_month            211054 non-null  int32
17  dac_day              211054 non-null  int32
18  tfa_year             211054 non-null  int32
19  tfa_month            211054 non-null  int32
20  tfa_day              211054 non-null  int32
dtypes: category(9), int32(10), int64(1), object(1)
memory usage: 14.7+ MB
```

## Visualization

```
In [34]: gender_index = clean_df['gender'].value_counts().index  
gender_value = clean_df['gender'].value_counts().values  
fig = px.pie(values=gender_value, names=gender_index,  
             title="Popular Genders")  
fig.update_traces(textposition='inside', textinfo='percent+label')  
fig.show()
```

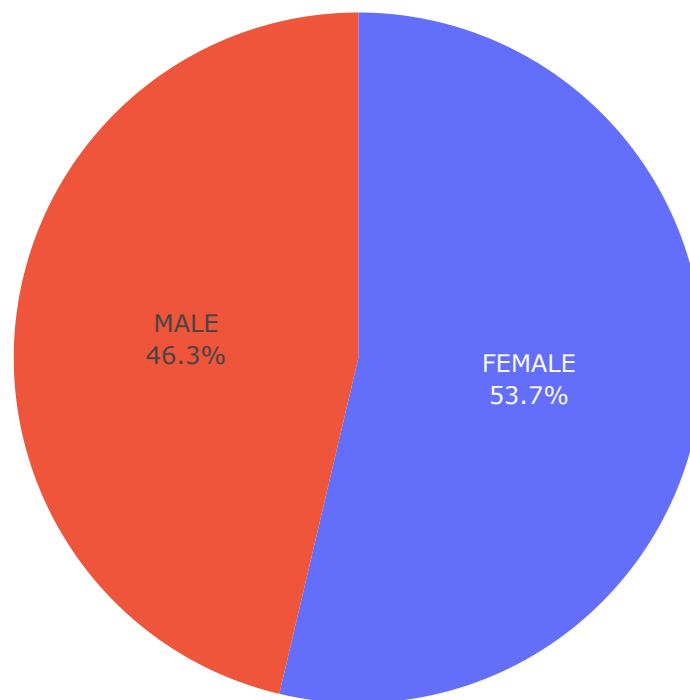
## Popular Genders



## M/F Genders

```
In [35]: gen_index=clean_df['gender'].value_counts().index[1:3]
gen_value=clean_df['gender'].value_counts().values[1:3]
fig = px.pie(values=gen_value, names=gen_index,
              title="M/F Genderes")
fig.update_traces(textposition='inside', textinfo='percent+label')
fig.show()
```

M/F Genderes

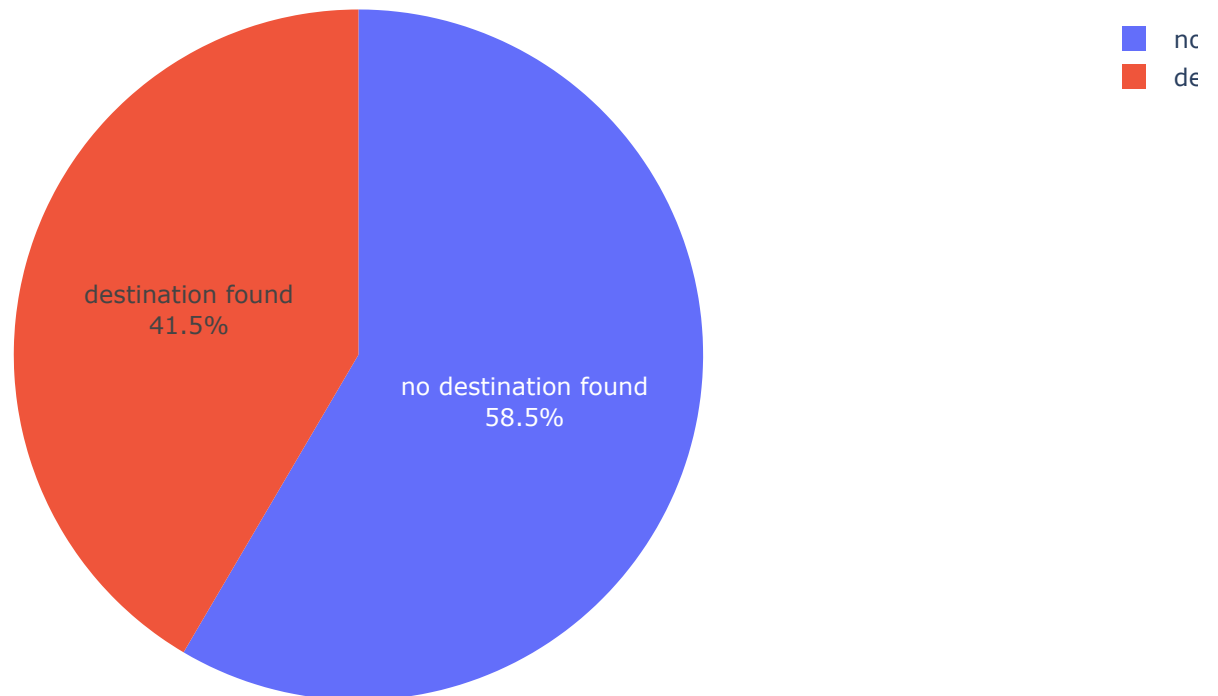


**Booking or Not ?**

```
In [36]: destination = pd.Series(['no destination found' if x == 'NDF' else 'destination found' for x in clean_df['country']])
```

```
In [37]: x=destination.value_counts().index  
y=destination.value_counts()  
fig = px.pie(names=x, values=y, title='Destination', hover_name=x)  
fig.update_traces(textposition='inside', textinfo='percent+label')  
fig.show()
```

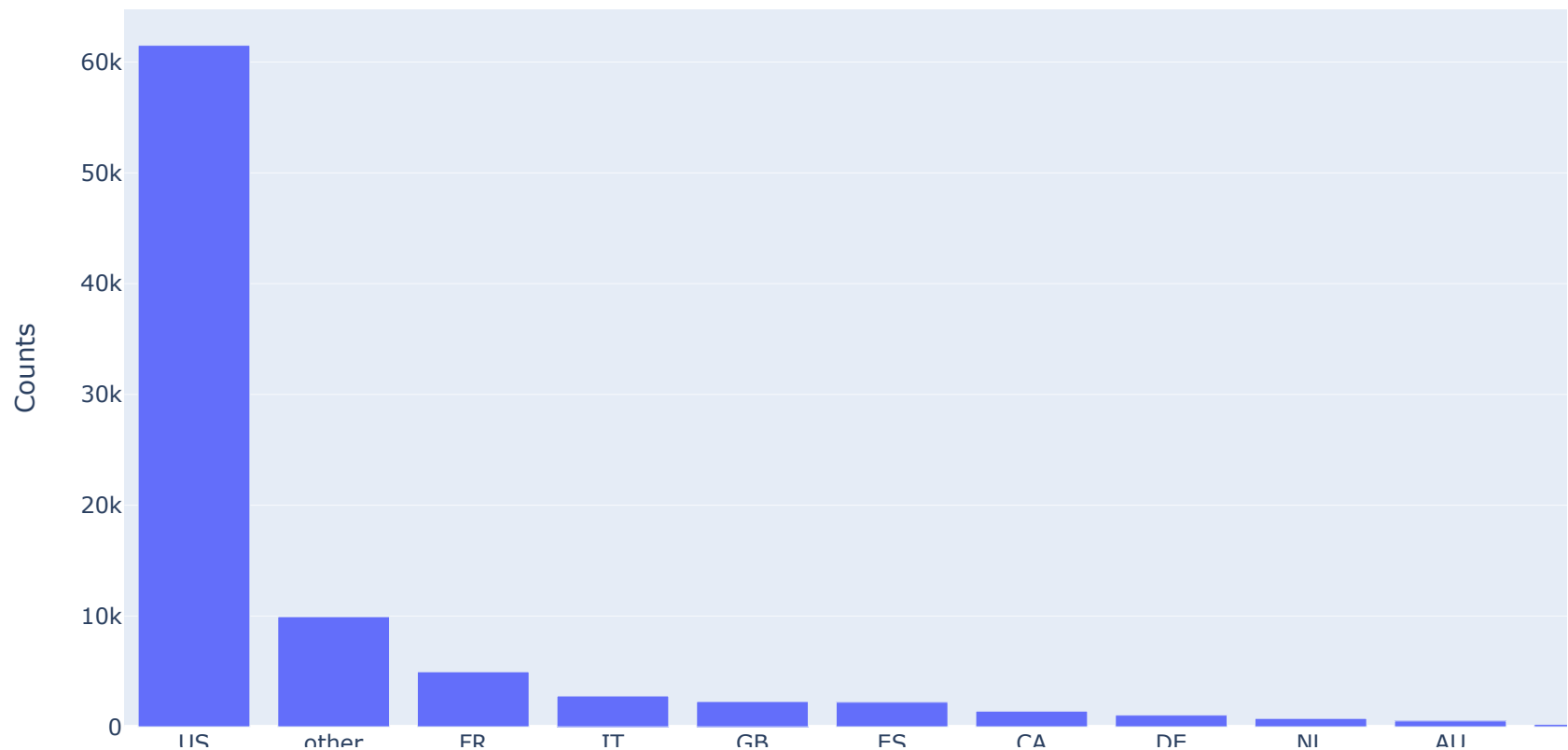
## Destination





## The most percentage of the people did not book a ticket when they first visit the website , but most of those who booked made the booking to the United States(US)

```
In [38]: x=clean_df['country_destination'].value_counts().index[1:]  
y=clean_df['country_destination'].value_counts()[1:]  
fig = px.bar(x=x, y=y, labels={  
    'x': "Countries", 'y': 'Counts'})  
fig.show()
```



```
In [39]: df_browsers = clean_df['first_browser']
df_browsers
```

```
Out[39]: 0          Chrome
1          Chrome
2           IE
3         Firefox
4          Chrome
...
213446      Safari
213447      Chrome
213448      Firefox
213449  Mobile Safari
213450    -unknown-
Name: first_browser, Length: 211054, dtype: object
```

```
In [40]: df_browsers.value_counts().index
```

```
Out[40]: Index(['Chrome', 'Safari', 'Firefox', '-unknown-', 'IE', 'Mobile Safari',
'Chrome Mobile', 'Android Browser', 'AOL Explorer', 'Opera', 'Silk',
'Chromium', 'BlackBerry Browser', 'Maxthon', 'IE Mobile', 'Apple Mail',
'Sogou Explorer', 'Mobile Firefox', 'RockMelt', 'SiteKiosk', 'Iron',
'Pale Moon', 'IceWeasel', 'Yandex.Browser', 'SeaMonkey', 'CometBird',
'Camino', 'TenFourFox', 'wOSBrowser', 'CoolNovo', 'Avant Browser',
'Opera Mini', 'Mozilla', 'OmniWeb', 'TheWorld Browser', 'Opera Mobile',
'Flock', 'SlimBrowser', 'Crazy Browser', 'Comodo Dragon', 'NetNewsWire',
'IceDragon', 'Stainless', 'Arora', 'Googlebot', 'Outlook 2007', 'Epic',
'Google Earth', 'Kindle Browser', 'PS Vita browser',
'Palm Pre web browser', 'Conkeror'],
dtype='object')
```

```
In [41]: df_browsers.value_counts().values
```

```
Out[41]: array([63172, 44563, 33226, 27117, 20757, 19086, 1259, 845, 234,
185, 120, 71, 53, 46, 36, 35, 33, 29,
24, 23, 17, 12, 12, 11, 11, 11, 9,
8, 6, 6, 4, 4, 3, 2, 2, 2,
2, 2, 2, 2, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1], dtype=int64)
```

```
In [42]: df_browsers.value_counts().values[6:]
```

```
Out[42]: array([[1259, 845, 234, 185, 120, 71, 53, 46, 36, 35, 33,
                29, 24, 23, 17, 12, 12, 11, 11, 11, 9, 8,
                6, 6, 4, 4, 3, 2, 2, 2, 2, 2, 2, 2,
                2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1], dtype=int64)
```

```
In [43]: b = df_browsers.value_counts().values[6:]
#add num of unknowns
sb = sum(b)+27117
sb
```

```
Out[43]: 30250
```

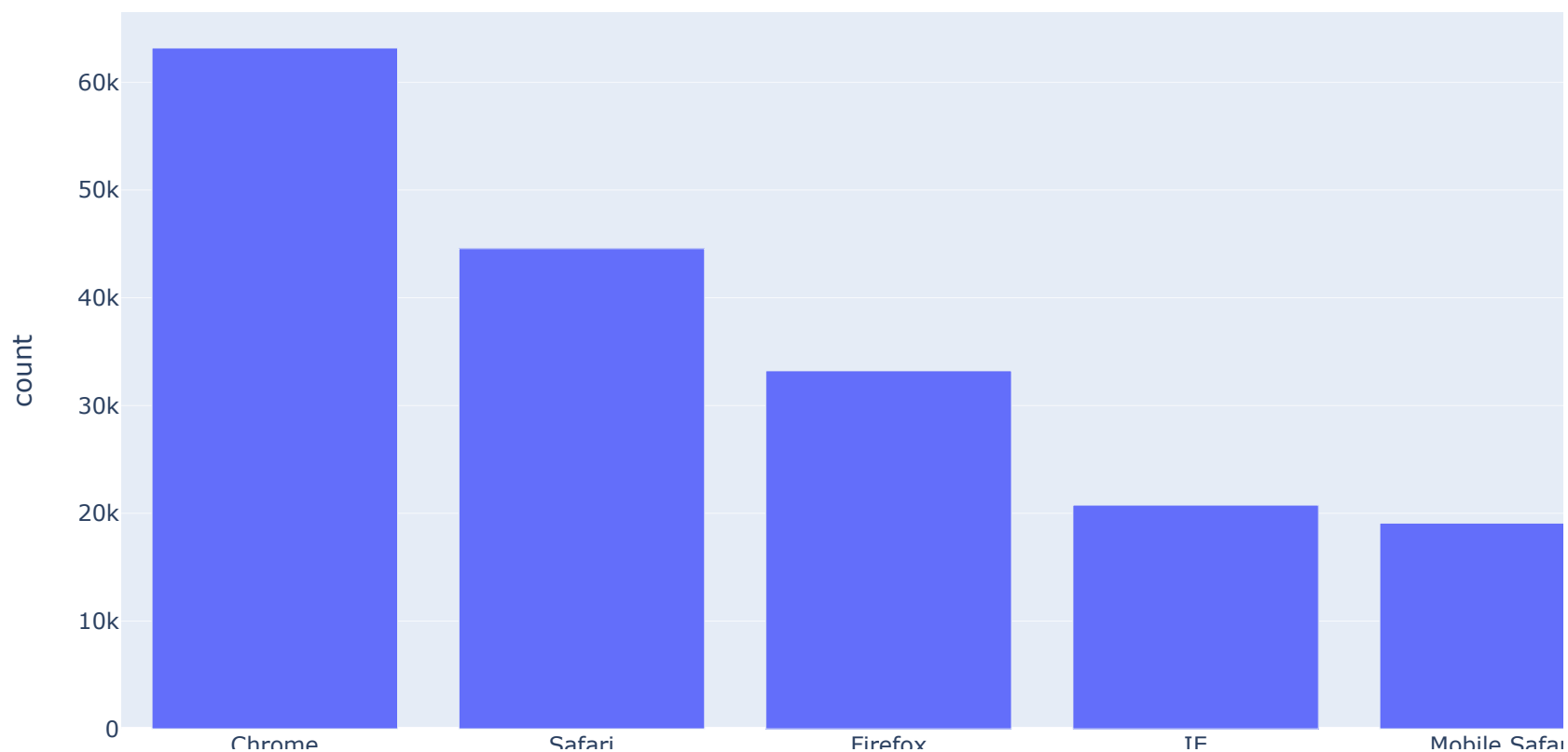
```
In [44]: browsers = df_browsers.value_counts()
browsers = browsers.drop(
    labels=['Chrome Mobile', 'Android Browser', 'Opera', 'AOL Explorer', 'Silk',
            'Chromium', 'BlackBerry Browser', 'Maxthon', 'Apple Mail', 'RockMelt',
            'Mobile Firefox', 'IE Mobile', 'Sogou Explorer', 'Iron', 'SiteKiosk',
            'IceWeasel', 'Yandex.Browser', 'Pale Moon', 'TenFourFox', 'Camino',
            'CoolNovo', 'Avant Browser', 'SeaMonkey', 'Opera Mobile', 'Opera Mini',
            'TheWorld Browser', 'wOSBrowser', 'SlimBrowser', 'Mozilla',
            'PS Vita browser', 'Stainless', 'NetNewsWire', 'Kindle Browser',
            'Comodo Dragon', 'CometBird', '-unknown-'])
```

```
In [45]: browsers['Others_browsers']=sb  
browsers
```

```
Out[45]: Chrome                63172  
Safari                44563  
Firefox               33226  
IE                   20757  
Mobile Safari        19086  
OmniWeb               2  
Flock                 2  
Crazy Browser         2  
IceDragon             1  
Arora                 1  
Googlebot             1  
Outlook 2007          1  
Epic                  1  
Google Earth          1  
Palm Pre web browser  1  
Conkeror              1  
Others_browsers       30250  
Name: first_browser, dtype: int64
```

## Top 5 Beowsers

```
In [46]: x=browsers.index[:5]  
y=browsers.values[:5]  
  
fig = px.bar(x=x, y=y, labels={  
            'x': 'browsers', 'y': 'count'})  
fig.show()
```

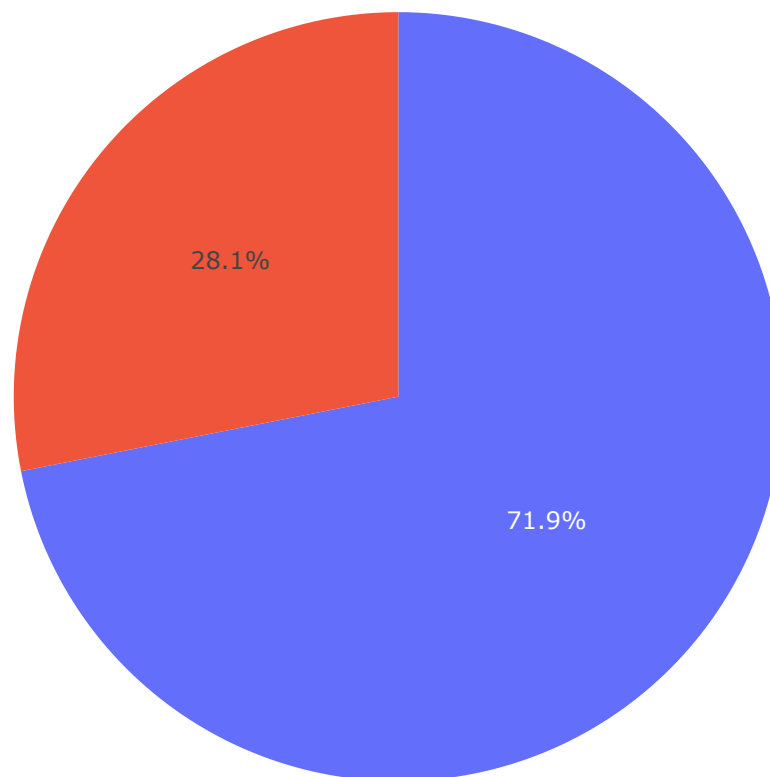


## The Top 2 Signing Up Methods

```
In [47]: clean_df['signup_method'].value_counts()
```

```
Out[47]: basic      151345  
facebook    59164  
google       545  
Name: signup_method, dtype: int64
```

```
In [48]: x=clean_df['signup_method'].value_counts().index[:2]  
y=clean_df['signup_method'].value_counts().values[:2]  
  
fig = px.pie(names=x, values=y)  
fig.show()
```



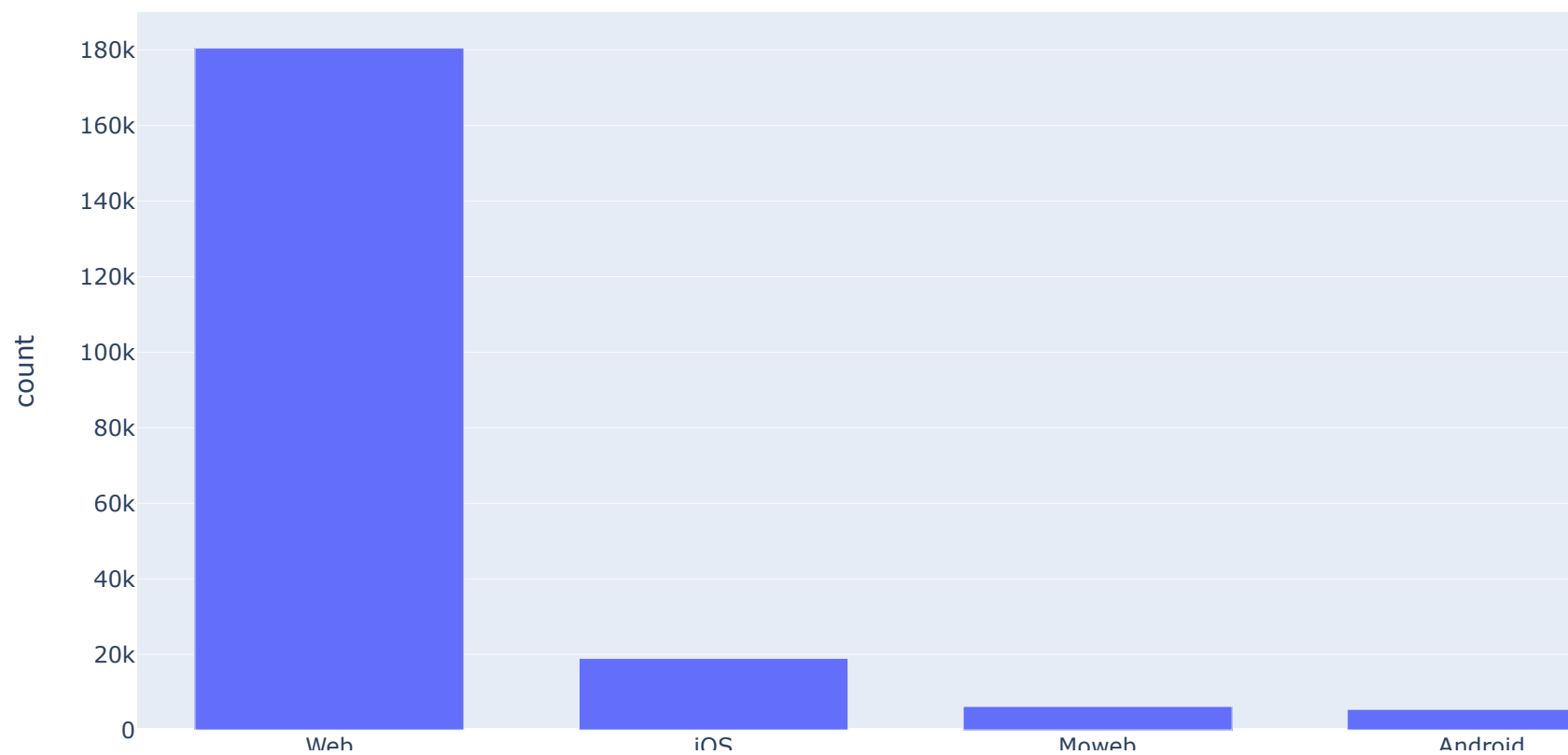
**signup app column**

```
In [49]: clean_df['signup_app'].value_counts()
```

```
Out[49]: Web          180508  
        iOS           18916  
        Moweb         6204  
        Android       5426  
        Name: signup_app, dtype: int64
```

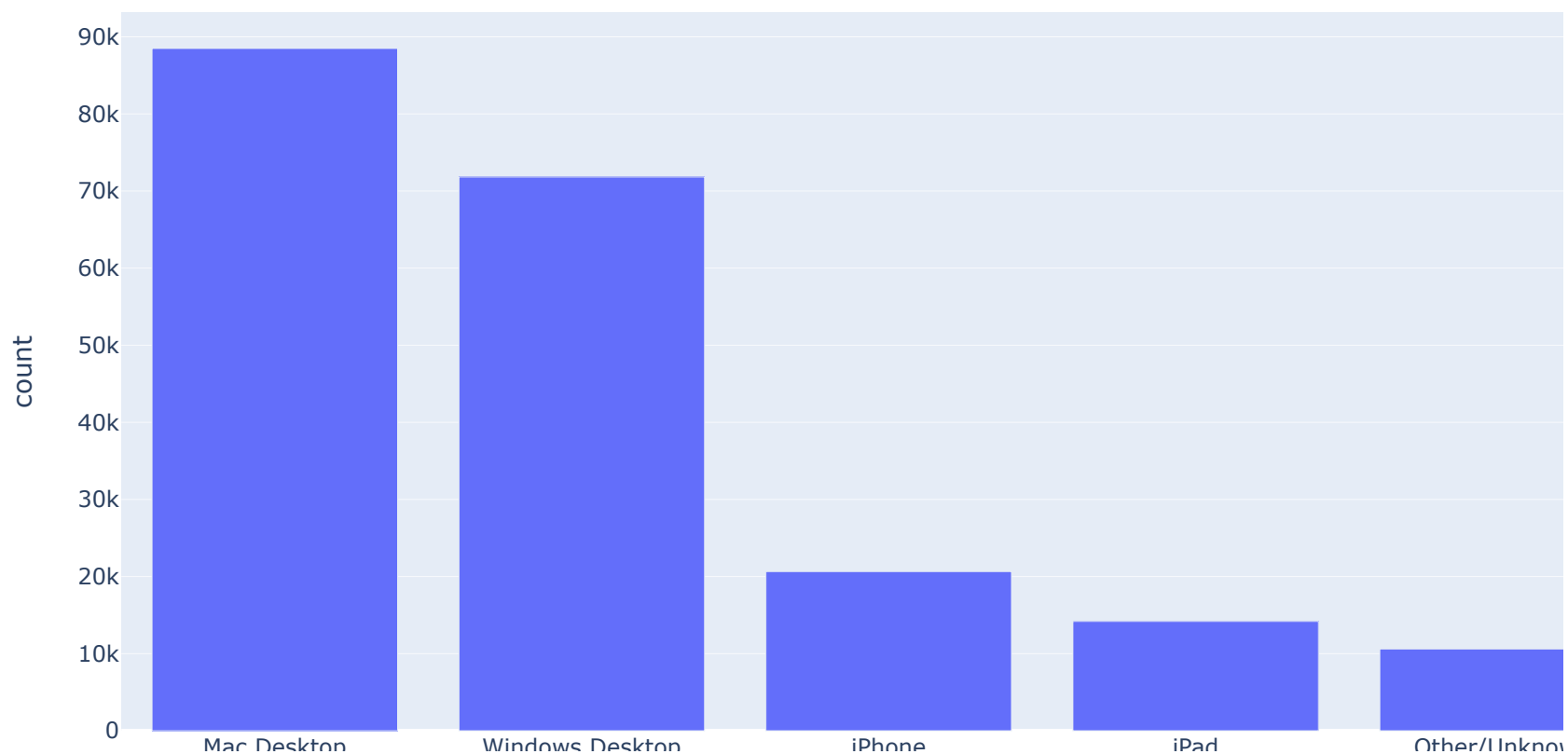


```
In [50]: x=clean_df['signup_app'].value_counts().index  
y=clean_df['signup_app'].value_counts().values  
  
fig = px.bar(x=x, y=y, labels={  
            'x': 'Apps', 'y': 'count'})  
fig.update_traces(width=0.7)  
fig.show()
```



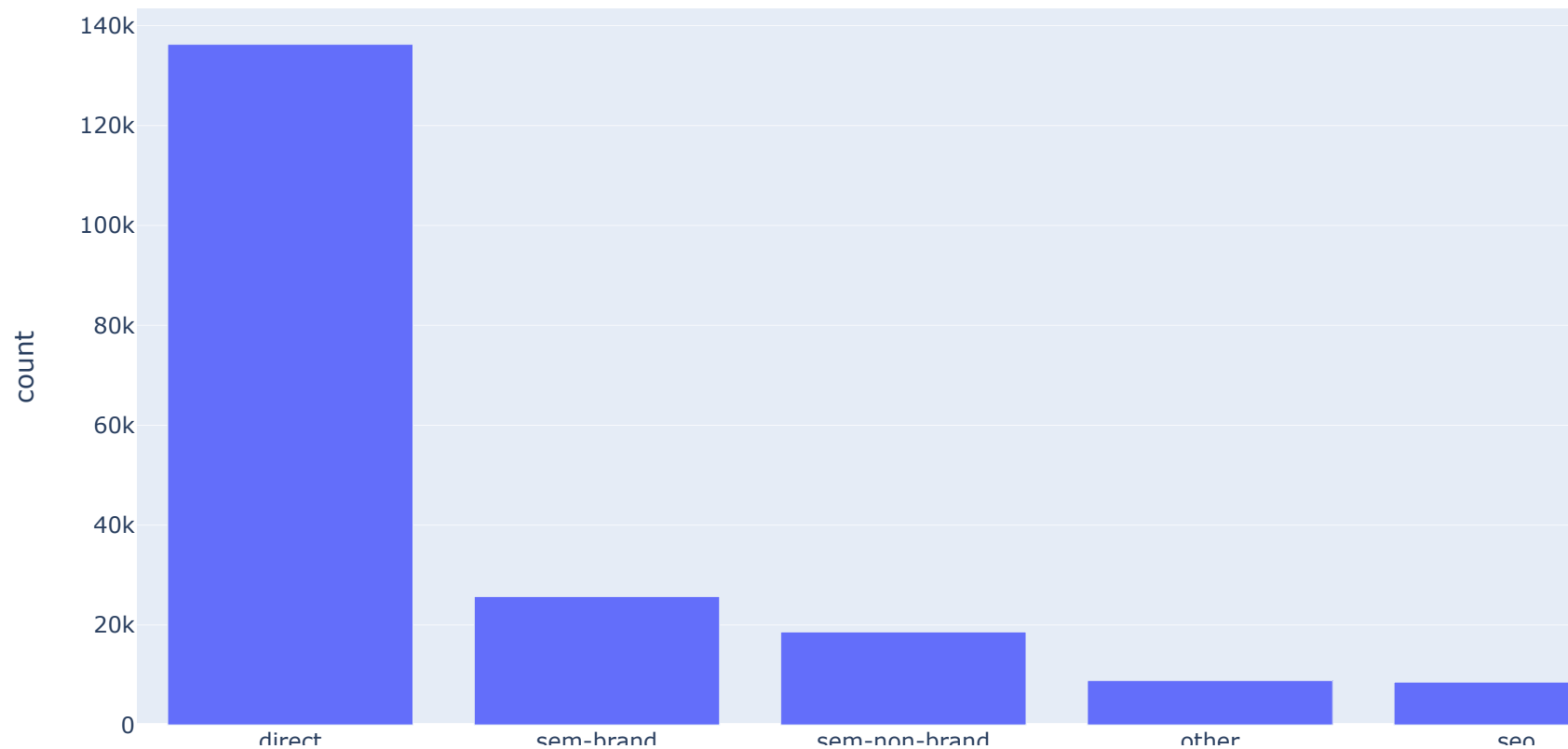
## Top 5 used devices

```
In [51]: x=clean_df['first_device_type'].value_counts().index[:5]  
y=clean_df['first_device_type'].value_counts().values[:5]  
fig = px.bar(x=x, y=y,labels={'x':'Devices','y':'count'})  
fig.show()
```



## Top 5 affiliate\_channels

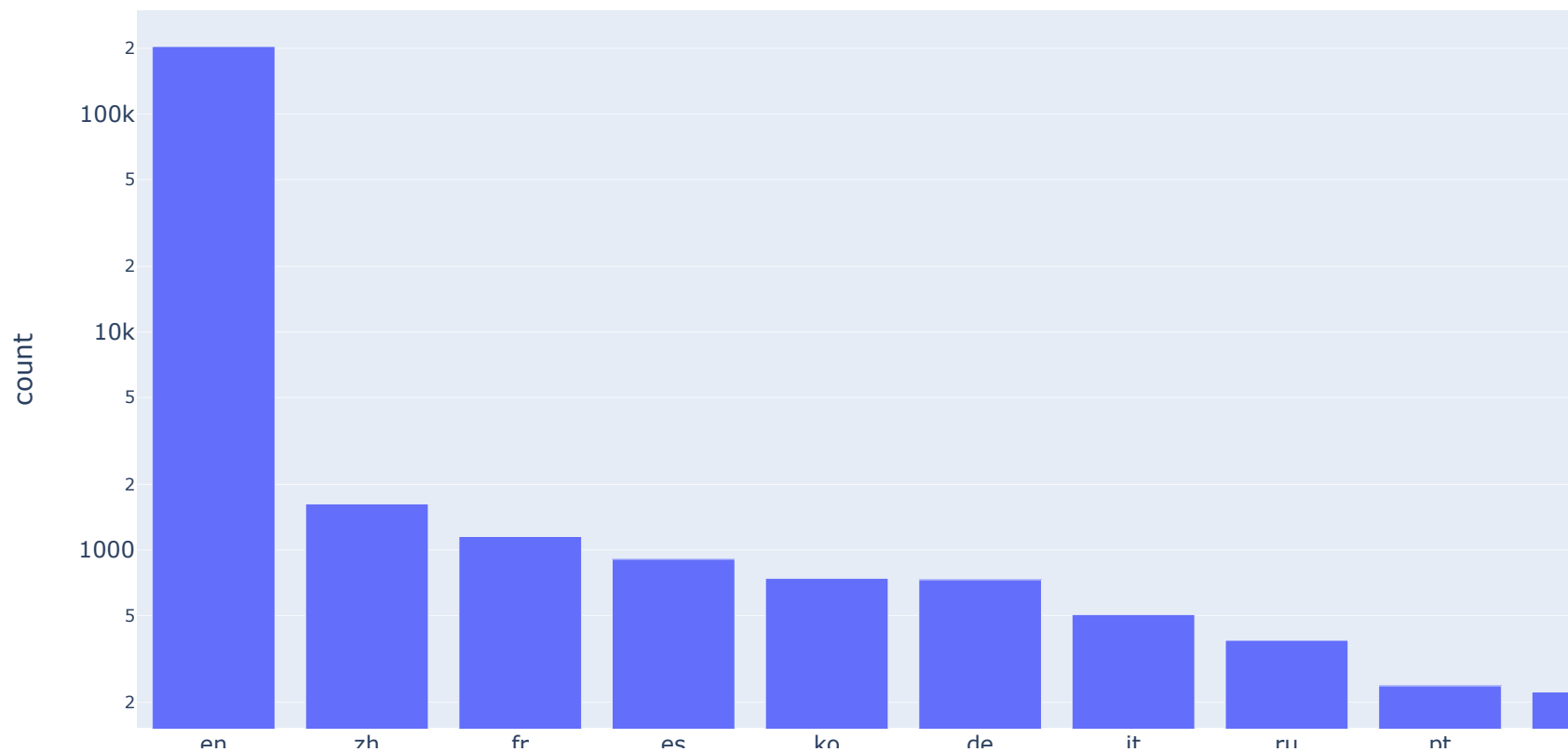
```
In [52]: x=clean_df['affiliate_channel'].value_counts().index[:5]  
y=clean_df['affiliate_channel'].value_counts().values[:5]  
fig = px.bar(x=x, y=y,labels={'x':'affiliate_channels','y':'count'})  
fig.show()
```



## Top 10 languages

```
In [53]: x=clean_df['language'].value_counts().index[:10]
y=clean_df['language'].value_counts().values[:10]
fig = px.bar(x=x, y=y, labels={'x':'language', 'y':'count'})
fig.update_yaxes(type="log")

fig.show()
```



**We Can see that the most of users has a English language then Chinese the**

# French

In [54]: `clean_df.head(2)`

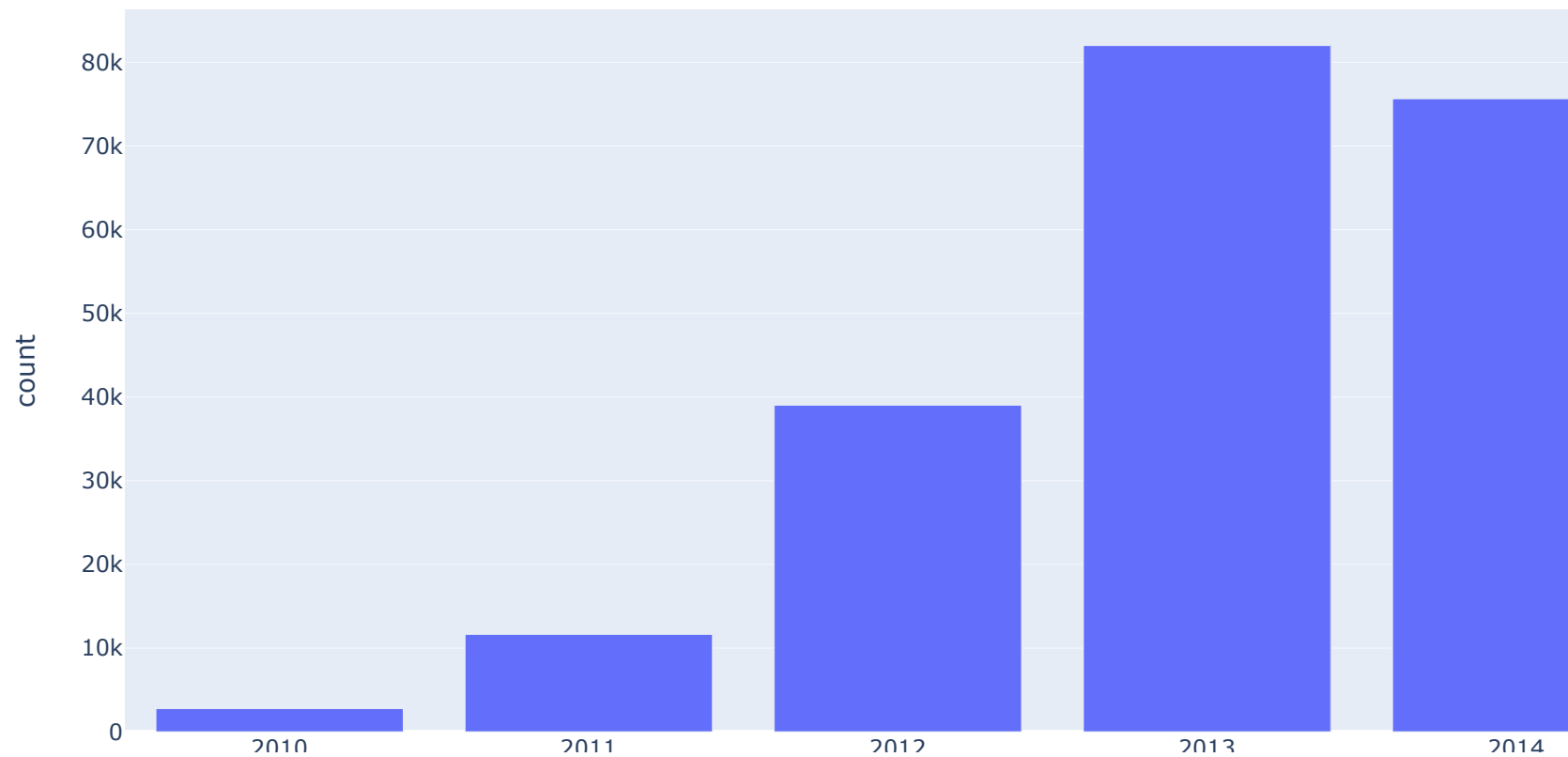
Out[54]:

	gender	age	signup_method	signup_flow	language	affiliate_channel	affiliate_provider	first_affiliate_tracked	signup_app	first_device
0	unknown-	39	facebook	0	en	direct	direct	untracked	Web	Mac I
1	MALE	38	facebook	0	en	seo	google	untracked	Web	Mac I

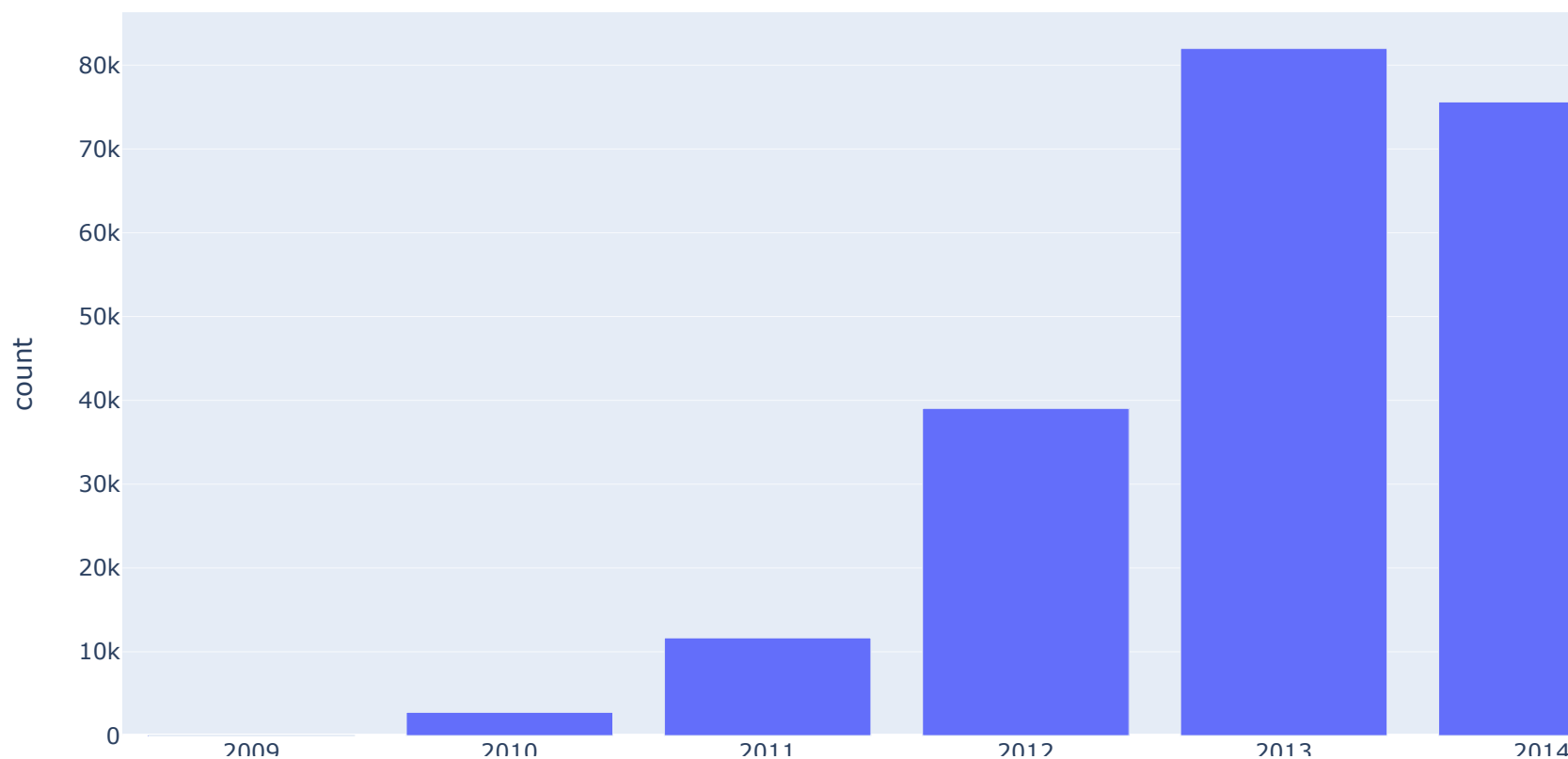
2 rows × 21 columns



```
In [55]: x = clean_df['dac_year'].value_counts().index  
y = clean_df['dac_year'].value_counts().values  
fig = px.bar(x=x, y=y, labels={'x': 'Acc Creating Date', 'y': 'count'})  
  
fig.show()
```



```
In [56]: x=clean_df['tfa_year'].value_counts().index  
y=clean_df['tfa_year'].value_counts().values  
fig = px.bar(x=x, y=y, labels={'x':'The First Activate time stamp','y':'count'})  
fig.show()
```



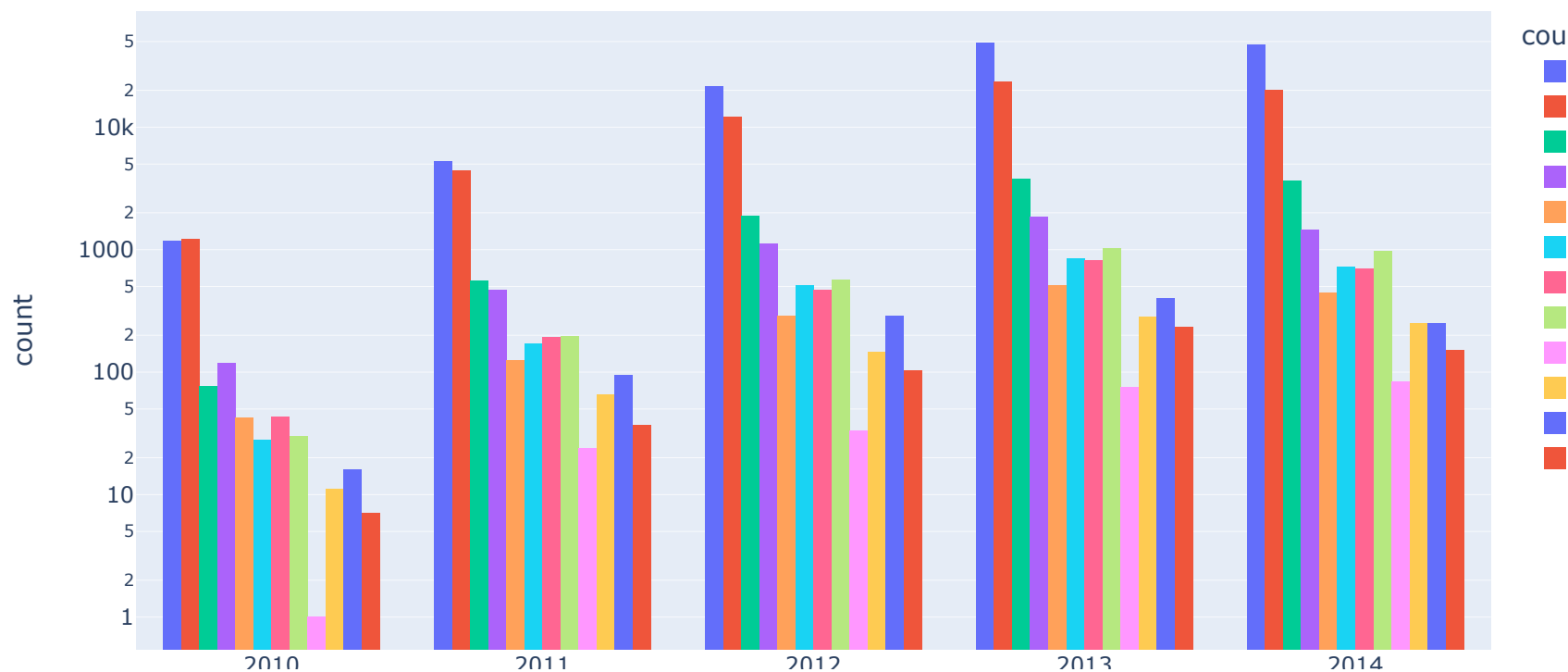
**The First Activate time stamp is earlier than Acc creating date by one year because users can search before signing up**

```
In [57]: #X = first active date  
X = clean_df['dac_year']
```



```
In [58]: fig = px.histogram(clean_df, x=X, y=X,
                             color = 'country_destination', histfunc='count', barmode='group', labels = {'x':"Country Destination",
                                                                                                     'y':'Count'},
                             title='Country destination across diff years')
fig.update_yaxes(type="log")
fig.update_layout(xaxis={'categoryorder':'category ascending'})
fig.show()
```

Country destination across diff years



**2013 and 2014 are the most years for creating accounts**

In [59]: *#Dividing member ages into various bins*

```
clean_df['users-bin']=clean_df['age'].apply(lambda x: '-10' if x<=10
                                             else '10 - 20' if 10<x<=20
                                             else '20 - 30' if 20<x<=30
                                             else '30 - 40' if 30<x<=40
                                             else '40 - 50' if 40<x<=50
                                             else '50 - 60' if 50<x<=60
                                             else '60-70' if 60<x<=70
                                             else '70+' if x>70
                                             else np.nan)

clean_df.head()
```

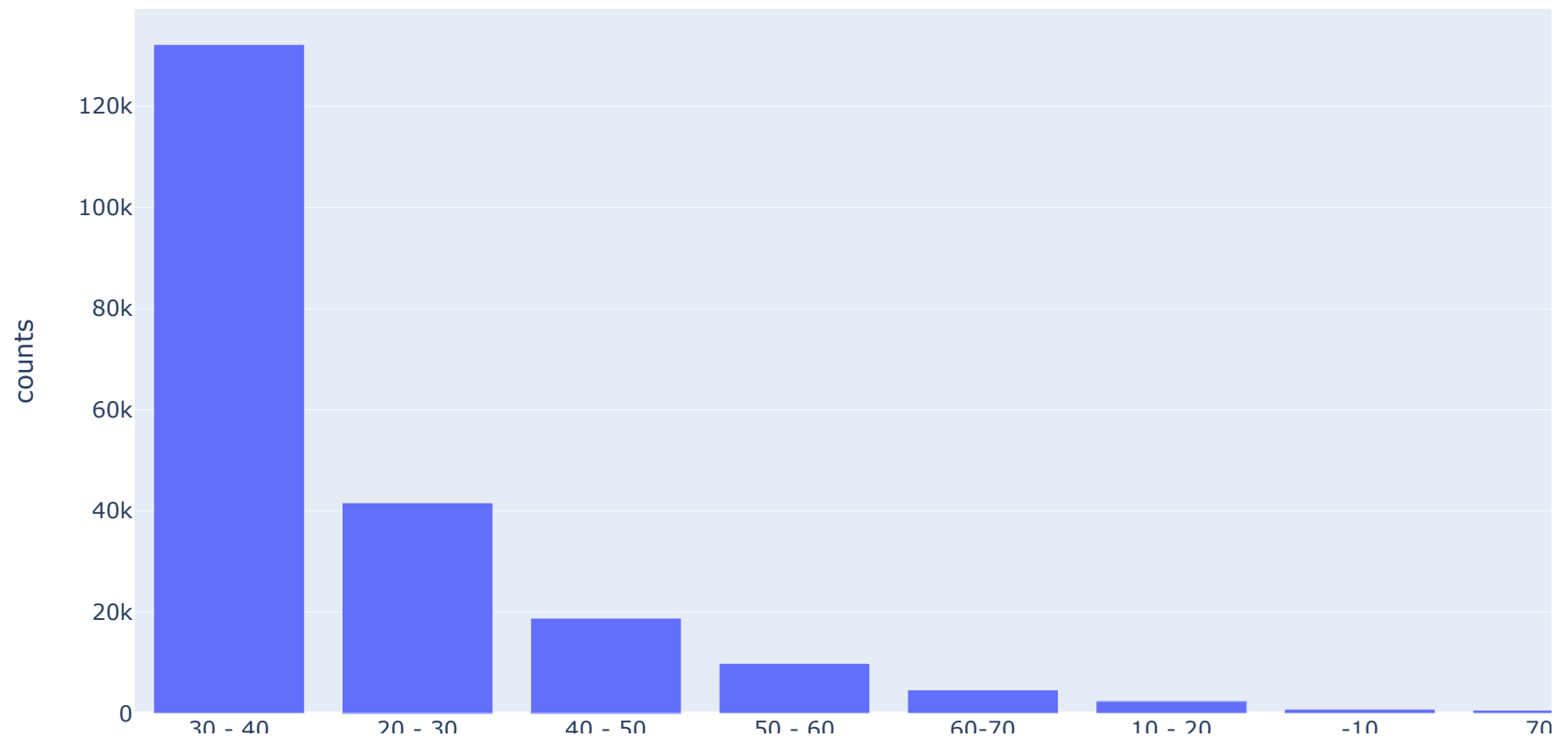
Out[59]:

	gender	age	signup_method	signup_flow	language	affiliate_channel	affiliate_provider	first_affiliate_tracked	signup_app	first_device
0	unknown-	39	facebook	0	en	direct	direct	untracked	Web	Mac I
1	MALE	38	facebook	0	en	seo	google	untracked	Web	Mac I
2	FEMALE	56	basic	3	en	direct	direct	untracked	Web	Windows I
3	FEMALE	42	facebook	0	en	direct	direct	untracked	Web	Mac I
4	unknown-	41	basic	0	en	direct	direct	untracked	Web	Mac I

5 rows × 22 columns



```
In [60]: x=clean_df['users-bin'].value_counts().index  
y=clean_df['users-bin'].value_counts().values  
fig = px.bar(x=x, y=y, labels={'x': 'Users_Bins', 'y': 'counts'})  
fig.show()
```



**The most interactive users are in period from the twenties to the forties**

```
In [61]: # Save dataset to the next phase  
c_df = clean_df.to_csv('clean_df.csv')
```

## Conclusions 🥰

**We can summarize our findings in those few points:**

- 1. some users prefer not to tell their gender, so if it is important info to booking operation , the 'gender must be mandatory field..**
- 2. Males near to females but females are the largest count**
- 3. The most of the people did not book a ticket when they first visit the website , but most of those who booked made the booking to the United States(US)**
- 4. The Top 5 used browsers are used:**
  - 'Chrome', 'Safari' , 'Firefox', 'Mobile Safari', 'IE',
- 5. The Top 2 Signing Up Methods are**
  - Basic , - Facebook
- 6. The Top 4 Used Types of Devices are :**

Mac Desktop , Windows Desktop , iPhone ,iPad
- 7. Top 4 affiliate\_channels are :**

**direct , sem-brand ,sem-non-brand , seo**

**8. 2013 and 2014 are the most popular years for the site**

**9. We Can see that the most of users has a English language then Chinese the French**

**10. The First Activate time stamp is earlier than Acc creating date by one year because users can search before signing up**

**11. The most interactive users are in period from the twenties to the forties**