In [1]:
```
import numpy as np # linear algebra, data manipulation
import pandas as pd # data processing, # Data Visualization
import seaborn as sns
import matplotlib.pyplot as plt
sns.set()
# Manipulating dates and time
from datetime import datetime
```

In [35]:
```
train_data= pd.read_csv('C:/Users/win10/Desktop/train_users_2.csv',index_col='date_
test_data= pd.read_csv('C:/Users/win10/Desktop/test_users.csv')
age_data = pd.read_csv('C:/Users/win10/Desktop/age_gender_bkts.csv')
train_data=train_data.dropna()
train_data.head()
```

Out[35]:

date_first_booking	id	date_account_created	timestamp_first_active	gender	age	sig
NaT	820tgsjxq7	2011-05-25	20090523174809	MALE	38.0	
2010-08-02	4ft3gnwmtx	2010-09-28	20090609231247	FEMALE	56.0	
2012-09-08	bjjt8pjhuk	2011-12-05	20091031060129	FEMALE	42.0	
2010-02-18	87mebub9p4	2010-09-14	20091208061105	-unknown-	41.0	
2010-01-05	lsw9q7uk0j	2010-01-02	20100102012558	FEMALE	46.0	

In [3]:
```
!pip install statsmodels --upgrade
```

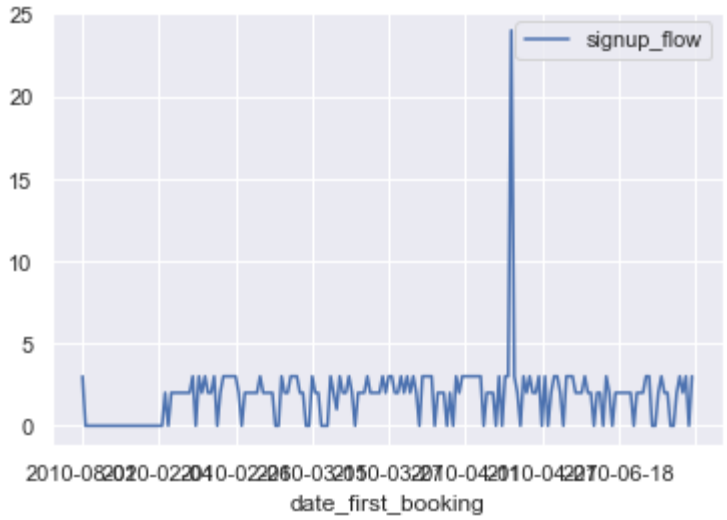
```
Requirement already satisfied: statsmodels in c:\users\win10\anaconda3\lib\site-pac
kages (0.13.2)
Requirement already satisfied: numpy>=1.17 in c:\users\win10\anaconda3\lib\site-pac
kages (from statsmodels) (1.21.5)
Requirement already satisfied: scipy>=1.3 in c:\users\win10\anaconda3\lib\site-pack
ages (from statsmodels) (1.7.3)
Requirement already satisfied: pandas>=0.25 in c:\users\win10\anaconda3\lib\site-pa
ckages (from statsmodels) (1.4.2)
Requirement already satisfied: patsy>=0.5.2 in c:\users\win10\anaconda3\lib\site-pa
ckages (from statsmodels) (0.5.2)
Requirement already satisfied: packaging>=21.3 in c:\users\win10\anaconda3\lib\site
-packages (from statsmodels) (21.3)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\win10\anaconda
3\lib\site-packages (from packaging>=21.3->statsmodels) (3.0.4)
Requirement already satisfied: pytz>=2020.1 in c:\users\win10\anaconda3\lib\site-pa
ckages (from pandas>=0.25->statsmodels) (2021.3)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\win10\anaconda3\l
ib\site-packages (from pandas>=0.25->statsmodels) (2.8.2)
Requirement already satisfied: six in c:\users\win10\anaconda3\lib\site-packages (f
rom patsy>=0.5.2->statsmodels) (1.16.0)
```

In [5]:
```
from statsmodels.tsa.ar_model import AutoReg
```

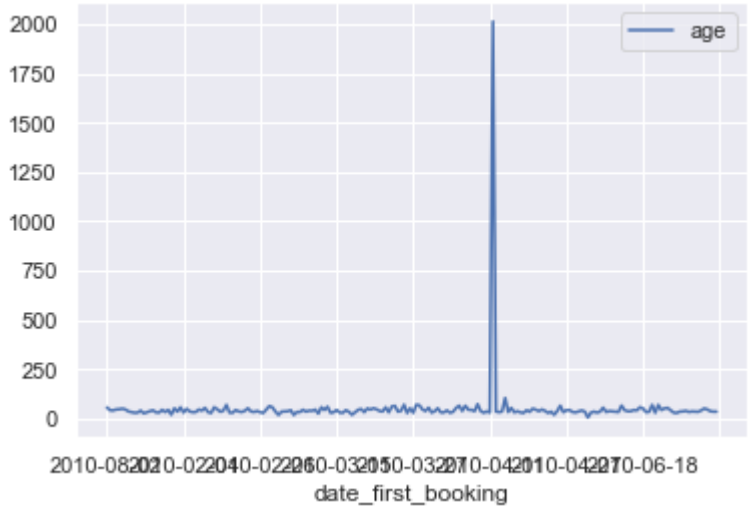
In [9]:
```
train_data=train_data.dropna()
X=train_data['signup_flow'].values
print('shape of data \t',train_data.shape)
print('original dataset :\n',train_data.head())
print('After extracting :\n',X)
```

```
shape of data    (68171, 15)
original dataset :
            date_account_created  timestamp_first_active date_first_booking  \
id
4ft3gnwmtx           2010-09-28          20090609231247         2010-08-02
bjjt8pjhuk           2011-12-05          20091031060129         2012-09-08
87mebub9p4           2010-09-14          20091208061105         2010-02-18
lsw9q7uk0j           2010-01-02          20100102012558         2010-01-05
0d01nltbrs           2010-01-03          20100103191905         2010-01-13

               gender   age signup_method  signup_flow language  \
id
4ft3gnwmtx     FEMALE  56.0         basic            3       en
bjjt8pjhuk     FEMALE  42.0      facebook            0       en
87mebub9p4  -unknown-  41.0         basic            0       en
lsw9q7uk0j     FEMALE  46.0         basic            0       en
0d01nltbrs     FEMALE  47.0         basic            0       en

           affiliate_channel affiliate_provider first_affiliate_tracked  \
id
4ft3gnwmtx            direct             direct               untracked
bjjt8pjhuk            direct             direct               untracked
87mebub9p4            direct             direct               untracked
lsw9q7uk0j             other         craigslist               untracked
0d01nltbrs            direct             direct                     omg

           signup_app first_device_type first_browser country_destination
id
4ft3gnwmtx        Web   Windows Desktop            IE                  US
bjjt8pjhuk        Web       Mac Desktop       Firefox               other
87mebub9p4        Web       Mac Desktop        Chrome                  US
lsw9q7uk0j        Web       Mac Desktop        Safari                  US
0d01nltbrs        Web       Mac Desktop        Safari                  US
After extracting :
 [3 0 0 ... 0 0 0]
```

In [22]:
```
x= train_data[:200].plot(x='date_first_booking',y='signup_flow')
```



In [23]:
```
x= train_data[:200].plot(x='date_first_booking',y='age')
```



In [21]:
```
from statsmodels.tsa.stattools import adfuller
def ad_test(dataset):
    dftest=adfuller(dataset, autolag='AIC')
    print("1.ADF ",dftest[0])
    print("2.p-value ",dftest[1])
    print("3.Number of Lags ",dftest[2])
    print("4.Number of observations & critical values ",dftest[3])
    print("5.Critical values ",dftest[4])
    for key,val in dftest[4].items():
        print("\t",key,":",val)
```

In [24]:
```
ad_test(train_data['signup_flow'])
```

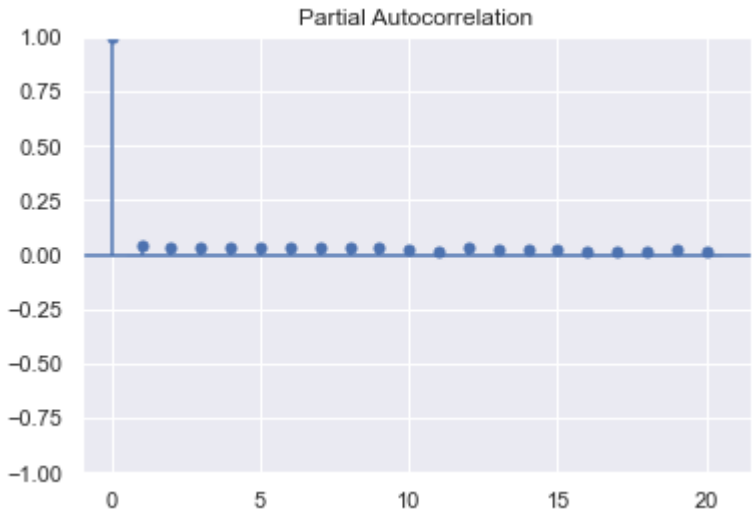
```
1.ADF  -18.992825584785283
2.p-value  0.0
3.Number of Lags  62
4.Number of observations & critical values  68108
5.Critical values  {'1%': -3.4304460173030864, '5%': -2.8615824379246875, '10%': -
2.5667925882604585}
         1% : -3.4304460173030864
         5% : -2.8615824379246875
         10% : -2.5667925882604585
```

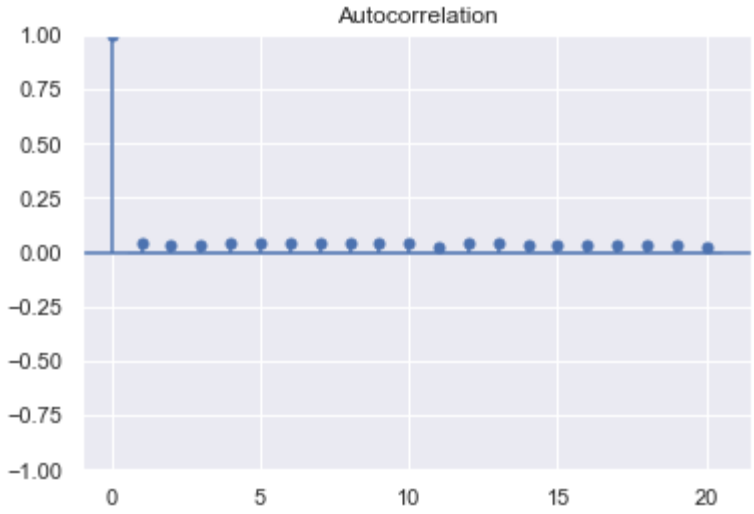
In [25]:
```
ad_test(train_data['age'])
```

```
1.ADF  -260.9881010375797
2.p-value  0.0
3.Number of Lags  0
4.Number of observations & critical values  68170
5.Critical values  {'1%': -3.430445929972927, '5%': -2.8615823993269482, '10%': -2.
566792567716089}
         1% : -3.430445929972927
         5% : -2.8615823993269482
         10% : -2.566792567716089
```

In [27]:
```
from statsmodels.graphics.tsaplots import plot_pacf,plot_acf
pacf=plot_pacf(train_data['signup_flow'],lags=20)
acf=plot_acf(train_data['signup_flow'],lags=20)
```

```
C:\Users\win10\anaconda3\lib\site-packages\statsmodels\graphics\tsaplots.py:348: Fu
tureWarning: The default method 'yw' can produce PACF values outside of the [-1,1]
interval. After 0.13, the default will change tounadjusted Yule-Walker ('ywm'). You
can use this method now by setting method='ywm'.
  warnings.warn(
```





Split the data into train and test to generate the model //last 7 days

In [31]:
```
train=X[:len(X)-7]
test=X[len(X)-7:]
model=AutoReg(train,lags=11).fit()
print(model.summary())
```

```
                            AutoReg Model Results
==============================================================================
Dep. Variable:                      y   No. Observations:                68164
Model:                  AutoReg(11)   Log Likelihood             -224248.721
Method:               Conditional MLE   S.D. of innovations              6.498
Date:                Wed, 07 Sep 2022   AIC                        448523.441
Time:                        04:00:46   BIC                        448642.125
Sample:                          11   HQIC                       448560.091
                              68164
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.7140      0.036     47.467      0.000       1.643       1.785
y.L1           0.0281      0.004      7.342      0.000       0.021       0.036
y.L2           0.0266      0.004      6.941      0.000       0.019       0.034
y.L3           0.0264      0.004      6.898      0.000       0.019       0.034
y.L4           0.0311      0.004      8.108      0.000       0.024       0.039
y.L5           0.0286      0.004      7.459      0.000       0.021       0.036
y.L6           0.0277      0.004      7.223      0.000       0.020       0.035
y.L7           0.0290      0.004      7.573      0.000       0.021       0.037
y.L8           0.0276      0.004      7.212      0.000       0.020       0.035
y.L9           0.0315      0.004      8.225      0.000       0.024       0.039
y.L10          0.0273      0.004      7.121      0.000       0.020       0.035
y.L11          0.0130      0.004      3.403      0.001       0.006       0.021
                                    Roots
==============================================================================
                  Real          Imaginary           Modulus         Frequency
------------------------------------------------------------------------------
AR.1            1.2022           -0.0000j            1.2022           -0.0000
AR.2            1.0565           -0.8436j            1.3519           -0.1072
AR.3            1.0565           +0.8436j            1.3519            0.1072
AR.4            0.4368           -1.3711j            1.4390           -0.2009
AR.5            0.4368           +1.3711j            1.4390            0.2009
AR.6           -0.3914           -1.4798j            1.5307           -0.2912
AR.7           -0.3914           +1.4798j            1.5307            0.2912
AR.8           -1.5855           -0.4344j            1.6440           -0.4574
AR.9           -1.5855           +0.4344j            1.6440            0.4574
AR.10          -1.1637           -1.1437j            1.6316           -0.3764
AR.11          -1.1637           +1.1437j            1.6316            0.3764
------------------------------------------------------------------------------
```

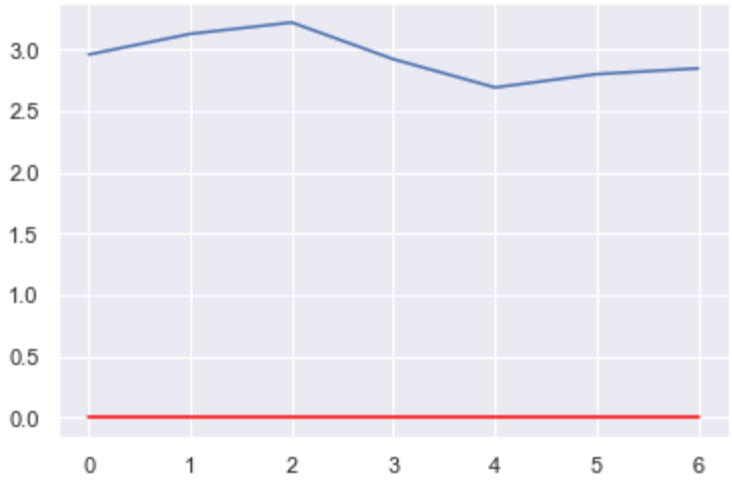
In [32]:
```
print(len(train))
```

```
68164
```

Making predictions to the test data

In [33]:
```
pred =model.predict(start=len(train),end=len(X)-1,dynamic=False)
plt.plot(pred)
plt.plot(test,color='red')
print(pred)
```

```
[2.9608364  3.12948439 3.22238525 2.9229977  2.69277886 2.80144863
 2.84830311]
```



In [37]:
```
from math import sqrt
from sklearn.metrics import mean_squared_error
rmse = sqrt(mean_squared_error(test,pred))
print(rmse)
```

```
2.9447267638487524
```

In [39]:
```
pred_future =model.predict(start=len(X)+1, end=len(X)+7,dynamic=False)
print("future predictions in next 7 days: ")
print(pred_future)
print("number of predictions made : ", len(pred_future))
```

```
future predictions in next 7 days:
[3.00491714 2.77186264 2.54774292 2.57570156 2.56219954 2.54387616
 2.52921902]
number of predictions made :  7
```

In [ ]: