import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import matplotlib as mpl  
  
pd.set\_option('max\_columns', 50)  
mpl.rcParams['lines.linewidth'] = 2  
  
%matplotlib inline

df = pd.read\_csv('Data Scientist Intern\_Data Set1.csv',  
 dtype={'customer\_id': str,  
 'order\_id': str},  
 parse\_dates=['month'],   
 infer\_datetime\_format=True)  
  
df.head()

#create order period column Y-m  
df['OrderPeriod'] = df.month.apply(lambda x: x.strftime('%Y-%m'))  
df.head()

#create corhort group, Y-m of first purchase  
df.set\_index('customer\_id', inplace=True)  
  
df['CohortGroup'] = df.groupby(level=0)['month'].min().apply(lambda x: x.strftime('%Y-%m'))  
df.reset\_index(inplace=True)  
df.head()

#group them by cohortgroup and orderperiod  
grouped = df.groupby(['CohortGroup', 'OrderPeriod'])  
  
# count the unique users, orders, and total revenue per Group + Period  
cohorts = grouped.agg({'customer\_id': pd.Series.nunique,  
 'order\_id': pd.Series.nunique})  
  
# make the column names more meaningful  
cohorts.rename(columns={'customer\_id': 'TotalUsers',  
 'order\_id': 'TotalOrders'}, inplace=True)  
cohorts

#Creates a cohortperiod column, which is the Nth period based on the user's first purchase.  
  
def cohort\_period(df):  
 df['CohortPeriod'] = np.arange(len(df)) + 1  
 return df  
  
cohorts = cohorts.groupby(level=0).apply(cohort\_period)  
cohorts.head()

# User retention by cohort group

# reindex the DataFrame  
cohorts.reset\_index(inplace=True)  
cohorts.set\_index(['CohortGroup', 'CohortPeriod'], inplace=True)  
  
# create a Series holding the total size of each CohortGroup. This is to be used in later calculations  
cohort\_group\_size = cohorts['TotalUsers'].groupby(level=0).first()  
cohort\_group\_size.head()

cohorts['TotalUsers'].head()

cohorts['TotalUsers'].unstack(0).head()

user\_retention = cohorts['TotalUsers'].unstack(0).divide(cohort\_group\_size, axis=1)  
user\_retention.head(10)

user\_retention.plot(figsize=(10,5))  
plt.title('Cohorts: User Retention')  
plt.xticks(np.arange(1, 12.1, 1))  
plt.xlim(1, 12)  
plt.ylabel('% of Cohort Purchasing');

import seaborn as sns  
sns.set(style='white')  
  
plt.figure(figsize=(12, 8))  
plt.title('Cohorts: User Retention')  
sns.heatmap(user\_retention.T, mask=user\_retention.T.isnull(), annot=True, fmt='.0%');

# Retention curve

# Unstack the TotalUsers  
unstacked = cohorts['TotalUsers'].unstack(0)  
unstacked.reset\_index()  
  
# Create a weighted data frame and reset the index  
weighted = unstacked.reset\_index()  
# Add a Total Subs column which sums up all of the subscribers within each Cohort Period.  
weighted['Total\_Subs'] = weighted.drop('CohortPeriod', axis=1).sum(axis=1)  
# Count non-NaN values in the row, call n  
# Add up first n values of the first row, n\_sum  
# Divide the value in the total subs column of that row by n\_sum  
weighted['num\_months'] = weighted['CohortPeriod'].count() - weighted.isnull().sum(axis=1)  
def calc\_sum(col\_end):  
 ans = 0   
 for i in range(1,int(col\_end)):  
 ans = ans + weighted.iloc[0, i]  
   
 return ans  
def calc\_ret\_pct(total\_subs, num\_months):  
 sum\_initial = calc\_sum(1 + num\_months)  
   
 return total\_subs / sum\_initial  
# Create a retention percentage column with use of a lambda function to apply calc ret pct for each row  
weighted['Ret\_Pct'] = weighted.apply(lambda row: calc\_ret\_pct(row['Total\_Subs'], row['num\_months']), axis=1)  
# weighted  
# Grab only the Cohort Period and Ret Pct columns  
weighted\_avg = weighted.filter(items=['CohortPeriod', 'Ret\_Pct'])  
weighted\_avg['Ret\_Pct'] = pd.Series(["{0:.2f}%".format(val \* 100) for val in weighted\_avg['Ret\_Pct']], index = weighted\_avg.index)  
weighted\_avg['CohortPeriod'] = weighted\_avg['CohortPeriod'].astype(int)  
weighted['Retention Percentage'] = (100 \* weighted['Ret\_Pct'].round(3))  
  
  
weighted\_avg[['CohortPeriod','Ret\_Pct']]

#Plotly  
import sys  
!conda install --yes --prefix {sys.prefix} plotly  
import plotly as py  
from plotly.offline import download\_plotlyjs, init\_notebook\_mode, iplot  
from plotly.graph\_objs import \*  
import plotly.graph\_objs as go  
import plotly  
import plotly.express as px  
  
layout = go.Layout(  
 title = 'Monthly Retention Curve',  
 yaxis=dict(  
# tickformat="%",  
 tickformat= '%',  
 categoryorder="category ascending",  
 ticksuffix='%'  
 )  
)  
fig = px.line(weighted, x="CohortPeriod", y="Retention Percentage", title='Cohorts: User Retention')  
fig.show(layout = layout)

# Cohort table and Retention Curve

plt.figure(figsize=(12, 8))  
plt.title('Cohorts: User Retention')  
sns.heatmap(user\_retention.T, mask=user\_retention.T.isnull(), annot=True, fmt='.0%');

In this version of the retention curve, I removed cohortperiod1(100%) as its an outlier and does not provide much data here.

import plotly.graph\_objects as go  
  
fig = go.Figure()  
  
fig.add\_trace(go.Scatter(  
 x = weighted\_avg.iloc[1:]['CohortPeriod'],  
 y = weighted\_avg.iloc[1:]['Ret\_Pct'],  
))  
  
fig.update\_layout(  
 yaxis=dict(  
 tickformat="%",  
 categoryorder="category ascending",  
 ),  
 title="Retention Curve (excluding cohort 1)",  
 xaxis\_title="Cohort Period",  
 yaxis\_title="Retention Percentage",  
 font=dict(  
 family="Courier New, monospace",  
 size=18,  
 color="RebeccaPurple"  
 )  
)  
  
fig.show()

# Reflections and suggestions

Looking at the Cohort table, with the exception of period 2 of the 2018-04 and period 4 of 2018-08, the overall retention is very low among users.

Similiarly on the retention curve, there isn't a clear trend after that sharp drop in rentention rate from cohort period 1-2

We do have enough data to properly conclude the root cause for the low retention rate. One broad suggestion I would recommend is to have users sign up for a news-letter. Sign-ups can be encouraged via a website pop-up that offers users a one time promo code in exchange for signing up with the news-letter. The process should be as easy as entering their email and hitting send. We do not want to deter customers with long and tedious steps.

With the news-letter service, exisiting customers will be notified pediodically of special sales and events. This ensures a higher customer recovery rate as well as the sustainbility of Skin brand awareness With this in mind, customers are more likely to make new purchases