**Python code** for performing a **Cohort Analysis on Customer Retention** over the last 2 months, tailored for a **lighting sub-rental company**, with visualization included:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from datetime import datetime, timedelta

# Step 1: Generate Sample Data

np.random.seed(0)

num\_customers = 200

start\_date = datetime.today() - timedelta(days=60)

end\_date = datetime.today()

customer\_ids = [f"CUST\_{i:04d}" for i in range(num\_customers)]

data = []

for cust in customer\_ids:

first\_purchase = start\_date + timedelta(days=np.random.randint(0, 30))

num\_rentals = np.random.randint(1, 5)

for \_ in range(num\_rentals):

rental\_date = first\_purchase + timedelta(days=np.random.randint(0, 30))

if rental\_date < end\_date:

data.append([cust, rental\_date])

df = pd.DataFrame(data, columns=['CustomerID', 'RentalDate'])

# Step 2: Preprocess for Cohort Analysis

df['RentalDate'] = pd.to\_datetime(df['RentalDate'])

df['RentalMonth'] = df['RentalDate'].dt.to\_period('M')

df['CohortMonth'] = df.groupby('CustomerID')['RentalDate'].transform('min').dt.to\_period('M')

# Step 3: Calculate Cohort Index

def get\_cohort\_index(df):

year\_diff = df['RentalMonth'].dt.year - df['CohortMonth'].dt.year

month\_diff = df['RentalMonth'].dt.month - df['CohortMonth'].dt.month

return year\_diff \* 12 + month\_diff + 1

df['CohortIndex'] = get\_cohort\_index(df)

# Step 4: Create Cohort Table

cohort\_data = df.groupby(['CohortMonth', 'CohortIndex'])['CustomerID'].nunique().unstack(0)

# Step 5: Visualize with Heatmap

plt.figure(figsize=(12, 6))

plt.title('Customer Retention Cohort Analysis\n(Lighting Sub-Rental Company)', fontsize=14)

plt.imshow(cohort\_data.T, cmap='Blues', aspect='auto')

plt.yticks(range(len(cohort\_data.columns)), cohort\_data.columns.astype(str))

plt.xticks(range(len(cohort\_data.index)), cohort\_data.index.astype(str), rotation=45)

plt.xlabel('Cohort Index (Months Since First Rental)')

plt.ylabel('Cohort Month')

plt.colorbar(label='Number of Active Customers')

plt.tight\_layout()

plt.show()

A blue and white chart

AI-generated content may be incorrect.

**How the Analysis and Plot Works:**

1. **Sample Data Generation**:
   * Simulated 200 customers.
   * Each customer made between 1 to 4 rentals in the last 2 months.
   * Each rental was randomly assigned within the date range.
2. **Preprocessing**:
   * **RentalMonth**: Month of each rental.
   * **CohortMonth**: The month a customer made their first rental.
   * **CohortIndex**: Number of months since the first rental (starts at 1).
3. **Cohort Table**:
   * Rows represent how many months since the cohort started.
   * Columns represent each cohort's starting month.
   * Values are the number of active customers in each period.
4. **Heatmap**:
   * Darker blues = higher customer counts.
   * Easy to visually compare retention over time by cohort.

A blue and white chart

AI-generated content may be incorrect.

**Interpretation:**

* Each column is a **cohort** (month of first rental).
* Each row shows the **percentage of retained customers** in subsequent months.
* Lighter colors represent **lower retention**, and darker colors represent **higher retention**.

This gives you a clear view of **how long customers stay active** after their first interaction.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from datetime import datetime, timedelta

# Step 1: Generate Sample Data

np.random.seed(0)

num\_customers = 200

start\_date = datetime.today() - timedelta(days=60)

end\_date = datetime.today()

customer\_ids = [f"CUST\_{i:04d}" for i in range(num\_customers)]

data = []

for cust in customer\_ids:

first\_purchase = start\_date + timedelta(days=np.random.randint(0, 30))

num\_rentals = np.random.randint(1, 5)

for \_ in range(num\_rentals):

rental\_date = first\_purchase + timedelta(days=np.random.randint(0, 30))

if rental\_date < end\_date:

data.append([cust, rental\_date])

df = pd.DataFrame(data, columns=['CustomerID', 'RentalDate'])

# Step 2: Preprocess for Cohort Analysis

df['RentalDate'] = pd.to\_datetime(df['RentalDate'])

df['RentalMonth'] = df['RentalDate'].dt.to\_period('M')

df['CohortMonth'] = df.groupby('CustomerID')['RentalDate'].transform('min').dt.to\_period('M')

# Step 3: Calculate Cohort Index

def get\_cohort\_index(df):

year\_diff = df['RentalMonth'].dt.year - df['CohortMonth'].dt.year

month\_diff = df['RentalMonth'].dt.month - df['CohortMonth'].dt.month

return year\_diff \* 12 + month\_diff + 1

df['CohortIndex'] = get\_cohort\_index(df)

# Step 4: Create Cohort Table with Retention Rates

cohort\_counts = df.groupby(['CohortMonth', 'CohortIndex'])['CustomerID'].nunique().unstack(0)

cohort\_sizes = cohort\_counts.loc[1] # Month 1 = Cohort Size

retention\_rates = cohort\_counts.divide(cohort\_sizes, axis=1).round(3) \* 100 # As percent

# Step 5: Visualize Retention Rates with Heatmap

plt.figure(figsize=(12, 6))

plt.title('Customer Retention Rates (%) Cohort Analysis\n(Lighting Sub-Rental Company)', fontsize=14)

plt.imshow(retention\_rates.T, cmap='Blues', aspect='auto')

plt.yticks(range(len(retention\_rates.columns)), retention\_rates.columns.astype(str))

plt.xticks(range(len(retention\_rates.index)), retention\_rates.index.astype(str), rotation=45)

plt.xlabel('Cohort Index (Months Since First Rental)')

plt.ylabel('Cohort Month')

plt.colorbar(label='Retention Rate (%)')

plt.tight\_layout()

plt.show()