In [1]:
```
import pandas as pd
import numpy as np
import datetime as dt
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_excel(r"C:\Users\PC1\Desktop\1 April 2023 to 31 March 2024 Data File.xlsx")
df
```

Out[2]:

	Order ID	Shiprocket Created At	Status	Product Name	Product Name1	Product Quantity	Product HSN	Customer Name	Customer Mobile	Created By	Order Type	Crops	A
0	KI0101042324	2023-04-01	Delivered	Konig	Konig 200ml	1	12119099.0	Lakshuman Ganpat Dhumal	4.339279e+06	KI	Repeat order	Banana	
1	KI0101042324	2023-04-01	Delivered	Rich Charge Power Gel	Rich Charge Power Gel 1Kg	4	3105.0	Lakshuman Ganpat Dhumal	4.339279e+06	KI	Repeat order	Banana	
2	KI0101042324	2023-04-01	Delivered	Distributor (Sticker)	Distributor (Sticker) (100ml)	3	31808.0	Lakshuman Ganpat Dhumal	4.339279e+06	KI	Repeat order	Banana	
3	KI0102042324	2023-04-01	Returned	Rich Charge Power Gel	Rich Charge Power Gel 1Kg	4	3105.0	Vijay shankar madane	4.274835e+06	KI	Fresh order	Banana	
4	KI0102042324	2023-04-01	Returned	Distributor (Sticker)	Distributor (Sticker) (100ml)	3	31808.0	Vijay shankar madane	4.274835e+06	KI	Fresh order	Banana	
...	...	...	...	...	...	...	...	...	...	...	...	...	
34771	MA2906032324	2024-03-29	In-Transit	Rich Charge Power Gel	Rich Charge Power Gel 500gms	1	3105.0	Vyankatesh Dnyaneshwar Godase	5.679649e+06	MA	Fresh order	Banana	
34772	MA2906032324	2024-03-29	In-Transit	Caloxid	Caloxid 400ml	1	28332990.0	Vyankatesh Dnyaneshwar Godase	5.679649e+06	MA	Fresh order	Banana	
34773	SD2901032324	2024-03-29	Delivered	Prime 45	Prime 45 2kg	1	25081010.0	Yuvraj Bodke	5.198610e+06	SD	Fresh order	Sugarcane	
34774	VR3001032324	2024-03-30	In-Transit	BioBitter	Bio Bitter 1 Litre	1	38089199.0	Gajanan Hanumant Mane	5.630951e+06	VR	Fresh order	Maize	
34775	VR3002032324	2024-03-30	In-Transit	BioBitter	Bio Bitter 1 Litre	1	38089199.0	Gaurav Vasudev Dhalpe	4.559351e+06	VR	Fresh order	Maize	

34776 rows × 20 columns

In [3]:
```
uni_que = df['Customer Name'].unique()
len(uni_que)
```

Out[3]: 14614

In [4]:
```
unii_que = df['Customer Mobile'].unique()
len(unii_que)
```

Out[4]: 12518

In [5]:
```
uniii_que = df[df['Order Type']=='Fresh order']
len(uniii_que)
```

Out[5]: 18630

In [6]:
```
df.describe()
```

Out[6]:

	Product Quantity	Product HSN	Customer Mobile	Product Price	Product Total	Price including GST	Order Total	Tax	Tax %
count	34776.000000	3.477400e+04	3.463900e+04	34776.000000	34776.000000	34776.000000	34776.000000	34776.000000	34776.000000
mean	1.603606	2.318180e+07	5.151671e+06	601.537117	945.666022	1029.987660	2400.896084	85.638286	0.094709
std	2.447615	1.377927e+07	9.813386e+05	317.300334	1362.750231	1473.111104	4093.404169	130.329598	0.050802
min	1.000000	1.151000e+03	2.415311e+03	100.000000	100.000000	118.000000	180.000000	0.000000	0.050000
25%	1.000000	1.211910e+07	4.831193e+06	363.560000	379.460000	429.000800	800.000000	26.428500	0.050000
50%	1.000000	3.002903e+07	5.327114e+06	523.800000	566.670000	595.003500	1176.000000	52.857000	0.050000
75%	1.000000	3.101010e+07	5.552874e+06	714.280000	1119.040000	1174.992000	2290.000000	93.540000	0.120000
max	100.000000	3.808920e+07	5.248382e+07	10500.000000	41100.000000	43155.000000	80876.000000	4536.000000	0.180000

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34776 entries, 0 to 34775
Data columns (total 20 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Order ID               34776 non-null  object
 1   Shiprocket Created At  34776 non-null  datetime64[ns]
 2   Status                 34776 non-null  object
 3   Product Name           34776 non-null  object
 4   Product Name1          34776 non-null  object
 5   Product Quantity       34776 non-null  int64
 6   Product HSN            34774 non-null  float64
 7   Customer Name          34772 non-null  object
 8   Customer Mobile        34639 non-null  float64
 9   Created By             34776 non-null  object
 10  Order Type             34776 non-null  object
 11  Crops                  34775 non-null  object
 12  Address City           34775 non-null  object
 13  Address State          34776 non-null  object
 14  Product Price          34776 non-null  float64
 15  Product Total          34776 non-null  float64
 16  Price including GST    34776 non-null  float64
 17  Order Total            34776 non-null  float64
 18  Tax                    34776 non-null  float64
 19  Tax %                  34776 non-null  float64
dtypes: datetime64[ns](1), float64(8), int64(1), object(10)
memory usage: 5.3+ MB
```

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
Order ID                  0
Shiprocket Created At     0
Status                    0
Product Name              0
Product Name1             0
Product Quantity          0
Product HSN               2
Customer Name             4
Customer Mobile         137
Created By                0
Order Type                0
Crops                     1
Address City              1
Address State             0
Product Price             0
Product Total             0
Price including GST       0
Order Total               0
Tax                       0
Tax %                     0
dtype: int64
```

In [9]:
```
df.duplicated().sum()
```

Out[9]:
```
104
```

In [10]:
```
df.drop_duplicates()
```

Out[10]:

	Order ID	Shiprocket Created At	Status	Product Name	Product Name1	Product Quantity	Product HSN	Customer Name	Customer Mobile	Created By	Order Type	Crops	A
0	KI0101042324	2023-04-01	Delivered	Konig	Konig 200ml	1	12119099.0	Lakshuman Ganpat Dhumal	4.339279e+06	KI	Repeat order	Banana	
1	KI0101042324	2023-04-01	Delivered	Rich Charge Power Gel	Rich Charge Power Gel 1Kg	4	3105.0	Lakshuman Ganpat Dhumal	4.339279e+06	KI	Repeat order	Banana	
2	KI0101042324	2023-04-01	Delivered	Distributor (Sticker)	Distributor (Sticker) (100ml)	3	31808.0	Lakshuman Ganpat Dhumal	4.339279e+06	KI	Repeat order	Banana	
3	KI0102042324	2023-04-01	Returned	Rich Charge Power Gel	Rich Charge Power Gel 1Kg	4	3105.0	Vijay shankar madane	4.274835e+06	KI	Fresh order	Banana	
4	KI0102042324	2023-04-01	Returned	Distributor (Sticker)	Distributor (Sticker) (100ml)	3	31808.0	Vijay shankar madane	4.274835e+06	KI	Fresh order	Banana	
...	...	...	...	...	...	...	...	...	...	...	...	...	
34771	MA2906032324	2024-03-29	In-Transit	Rich Charge Power Gel	Rich Charge Power Gel 500gms	1	3105.0	Vyankatesh Dnyaneshwar Godase	5.679649e+06	MA	Fresh order	Banana	
34772	MA2906032324	2024-03-29	In-Transit	Caloxid	Caloxid 400ml	1	28332990.0	Vyankatesh Dnyaneshwar Godase	5.679649e+06	MA	Fresh order	Banana	
34773	SD2901032324	2024-03-29	Delivered	Prime 45	Prime 45 2kg	1	25081010.0	Yuvraj Bodke	5.198610e+06	SD	Fresh order	Sugarcane	
34774	VR3001032324	2024-03-30	In-Transit	BioBitter	Bio Bitter 1 Litre	1	38089199.0	Gajanan Hanumant Mane	5.630951e+06	VR	Fresh order	Maize	
34775	VR3002032324	2024-03-30	In-Transit	BioBitter	Bio Bitter 1 Litre	1	38089199.0	Gaurav Vasudev Dhalpe	4.559351e+06	VR	Fresh order	Maize	

34672 rows × 20 columns

In [11]:
```
df.rename(columns={'Shiprocket Created At':'order_date'}, inplace=True)
df['order_date']
```

Out[11]:
```
0       2023-04-01
1       2023-04-01
2       2023-04-01
3       2023-04-01
4       2023-04-01
           ...
34771   2024-03-29
34772   2024-03-29
34773   2024-03-29
34774   2024-03-30
34775   2024-03-30
Name: order_date, Length: 34776, dtype: datetime64[ns]
```

In [12]:

```
def get_month(x): return dt.datetime(x.year, x.month,1)
df['Invoice_month'] = df['order_date'].apply(get_month)
grouping = df.groupby('Customer Name')['Invoice_month']
df['Cohort_month'] = grouping.transform('min')
df
```

Out[12]:

	Order ID	order_date	Status	Product Name	Product Name1	Product Quantity	Product HSN	Customer Name	Customer Mobile	Created By	...	Address City	Address State
0	KI0101042324	2023-04-01	Delivered	Konig	Konig 200ml	1	12119099.0	Lakshuman Ganpat Dhumal	4.339279e+06	KI	...	AC1001	AZ100
1	KI0101042324	2023-04-01	Delivered	Rich Charge Power Gel	Rich Charge Power Gel 1Kg	4	3105.0	Lakshuman Ganpat Dhumal	4.339279e+06	KI	...	AC1001	AZ100
2	KI0101042324	2023-04-01	Delivered	Distributor (Sticker)	Distributor (Sticker) (100ml)	3	31808.0	Lakshuman Ganpat Dhumal	4.339279e+06	KI	...	AC1001	AZ100
3	KI0102042324	2023-04-01	Returned	Rich Charge Power Gel	Rich Charge Power Gel 1Kg	4	3105.0	Vijay shankar madane	4.274835e+06	KI	...	AC1001	AZ100
4	KI0102042324	2023-04-01	Returned	Distributor (Sticker)	Distributor (Sticker) (100ml)	3	31808.0	Vijay shankar madane	4.274835e+06	KI	...	AC1001	AZ100
...	...	...	...	...	...	...	...	...	...	...	...	...	..
34771	MA2906032324	2024-03-29	In-Transit	Rich Charge Power Gel	Rich Charge Power Gel 500gms	1	3105.0	Vyankatesh Dnyaneshwar Godase	5.679649e+06	MA	...	AC1087	AZ100
34772	MA2906032324	2024-03-29	In-Transit	Caloxid	Caloxid 400ml	1	28332990.0	Vyankatesh Dnyaneshwar Godase	5.679649e+06	MA	...	AC1087	AZ100
34773	SD2901032324	2024-03-29	Delivered	Prime 45	Prime 45 2kg	1	25081010.0	Yuvraj Bodke	5.198610e+06	SD	...	AC1462	AZ100
34774	VR3001032324	2024-03-30	In-Transit	BioBitter	Bio Bitter 1 Litre	1	38089199.0	Gajanan Hanumant Mane	5.630951e+06	VR	...	AC1098	AZ100
34775	VR3002032324	2024-03-30	In-Transit	BioBitter	Bio Bitter 1 Litre	1	38089199.0	Gaurav Vasudev Dhalpe	4.559351e+06	VR	...	AC1098	AZ100

34776 rows × 22 columns

In [13]:
```
df['Invoice_month'].equals(df['Cohort_month'])
```

Out[13]:
```
False
```

In [14]:
```
def get_month_int (dframe,column):
    year = dframe[column].dt.year
    month = dframe[column].dt.month
    day = dframe[column].dt.day
    return year, month , day

invoice_year,invoice_month,_ = get_month_int(df,'Invoice_month')
cohort_year,cohort_month,_ = get_month_int(df,'Cohort_month')

year_diff = invoice_year - cohort_year
month_diff = invoice_month - cohort_month

df['Cohort_index'] = year_diff * 12 + month_diff
```

In [15]:
```
#Count monthly active customers from each cohort
grouping = df.groupby(['Cohort_month', 'Cohort_index'])
cohort_data = grouping['Customer Name'].apply(pd.Series.nunique)
# Return number of unique elements in the object.
cohort_data = cohort_data.reset_index()
cohort_counts = cohort_data.pivot(index='Cohort_month',columns='Cohort_index',values='Customer Name')
cohort_counts
```

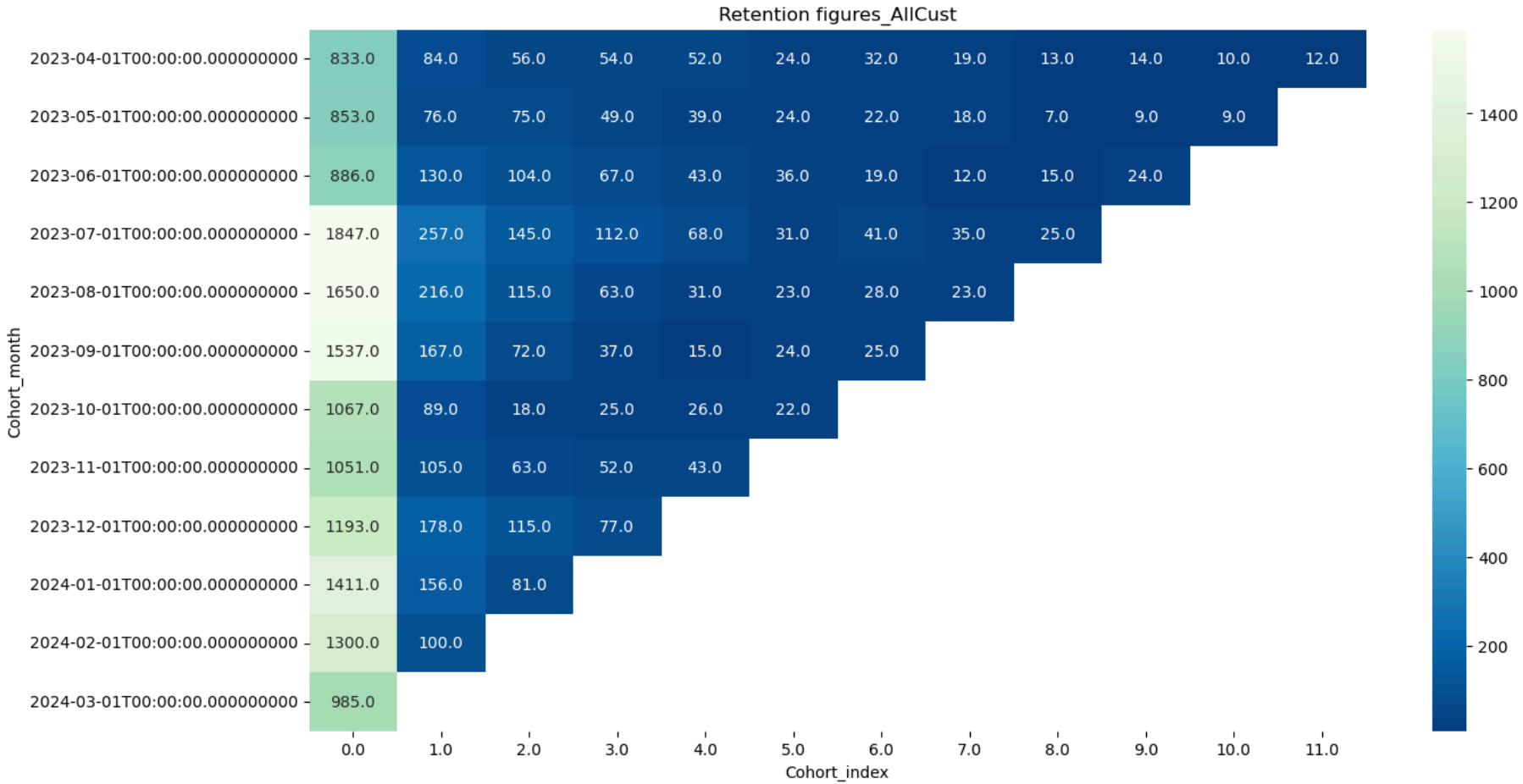
Out[15]:

Cohort_index	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0
Cohort_month												
2023-04-01	833.0	84.0	56.0	54.0	52.0	24.0	32.0	19.0	13.0	14.0	10.0	12.0
2023-05-01	853.0	76.0	75.0	49.0	39.0	24.0	22.0	18.0	7.0	9.0	9.0	NaN
2023-06-01	886.0	130.0	104.0	67.0	43.0	36.0	19.0	12.0	15.0	24.0	NaN	NaN
2023-07-01	1847.0	257.0	145.0	112.0	68.0	31.0	41.0	35.0	25.0	NaN	NaN	NaN
2023-08-01	1650.0	216.0	115.0	63.0	31.0	23.0	28.0	23.0	NaN	NaN	NaN	NaN
2023-09-01	1537.0	167.0	72.0	37.0	15.0	24.0	25.0	NaN	NaN	NaN	NaN	NaN
2023-10-01	1067.0	89.0	18.0	25.0	26.0	22.0	NaN	NaN	NaN	NaN	NaN	NaN
2023-11-01	1051.0	105.0	63.0	52.0	43.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2023-12-01	1193.0	178.0	115.0	77.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2024-01-01	1411.0	156.0	81.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2024-02-01	1300.0	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2024-03-01	985.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [16]:
```
cohort_counts[0.0].sum()
```

Out[16]:
```
14613.0
```

In [26]:
```
#Build the heatmap
plt.figure(figsize=(15, 8))
plt.title('Retention figures_AllCust')
sns.heatmap(data=cohort_counts,annot = True,fmt = '0',cmap="GnBu_r", robust=True)
plt.show()
```



In [18]:
```
# Retention table
cohort_size = cohort_counts.iloc[:,0]
retention = cohort_counts.divide(cohort_size,axis=0) #axis=0 to ensure the divide along the row axis
retention.round(3) * 100 #to show the number as percentage
```

Out[18]:

Cohort_index	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0
Cohort_month												
2023-04-01	100.0	10.1	6.7	6.5	6.2	2.9	3.8	2.3	1.6	1.7	1.2	1.4
2023-05-01	100.0	8.9	8.8	5.7	4.6	2.8	2.6	2.1	0.8	1.1	1.1	NaN
2023-06-01	100.0	14.7	11.7	7.6	4.9	4.1	2.1	1.4	1.7	2.7	NaN	NaN
2023-07-01	100.0	13.9	7.9	6.1	3.7	1.7	2.2	1.9	1.4	NaN	NaN	NaN
2023-08-01	100.0	13.1	7.0	3.8	1.9	1.4	1.7	1.4	NaN	NaN	NaN	NaN
2023-09-01	100.0	10.9	4.7	2.4	1.0	1.6	1.6	NaN	NaN	NaN	NaN	NaN
2023-10-01	100.0	8.3	1.7	2.3	2.4	2.1	NaN	NaN	NaN	NaN	NaN	NaN
2023-11-01	100.0	10.0	6.0	4.9	4.1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2023-12-01	100.0	14.9	9.6	6.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2024-01-01	100.0	11.1	5.7	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2024-02-01	100.0	7.7	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2024-03-01	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [24]:
```
#Build the heatmap
plt.figure(figsize=(15, 8))
plt.title('Retention %_AllCust')
sns.heatmap(data=retention,annot = True,fmt = '.0%',cmap="GnBu_r", robust=True, cbar=False)
plt.show()
```