Cohort Analysis with online retail dashboard personal

December 22, 2021

```python
[16]: import pandas as pd
      import numpy as np
      import matplotlib.pyplot as plt
      import matplotlib as mpl
      get_ipython().magic(u'matplotlib inline')
      from datetime import datetime, timedelta
      %matplotlib inline
```

```python
[17]: df = pd.read_excel('Online Retail.xlsx')
```

```python
[18]: df.head()
```

```
[18]:    InvoiceNo                          Description  Quantity InvoiceDate  \
      0   536365.0    WHITE HANGING HEART T-LIGHT HOLDER       6.0  2010-12-01
      1   536365.0                   WHITE METAL LANTERN       6.0  2010-12-01
      2   536365.0        CREAM CUPID HEARTS COAT HANGER       8.0  2010-12-01
      3   536365.0   KNITTED UNION FLAG HOT WATER BOTTLE       6.0  2010-12-01
      4   536365.0           RED WOOLLY HOTTIE WHITE HEART.      6.0  2010-12-01

         UnitPrice  CustomerID         Country  Sales
      0       2.55     17850.0  United Kingdom  15.30
      1       3.39     17850.0  United Kingdom  20.34
      2       2.75     17850.0  United Kingdom  22.00
      3       3.39     17850.0  United Kingdom  20.34
      4       3.39     17850.0  United Kingdom  20.34
```

```python
[19]: # converting date columns to datetime
      date_columns = ['InvoiceDate']
      for col in date_columns:
          df[col] = pd.to_datetime(df[col], format='%Y-%m-%d %H:%M:%S')
```

```python
[20]: df.set_index('CustomerID', inplace=True)

      df['CohortGroup'] = df.groupby(level=0)['InvoiceDate'].min().apply(lambda x: x.
       ↪strftime('%Y-%m'))
      df.reset_index(inplace=True)
      df.head()
```

```
[20]:    CustomerID  InvoiceNo                          Description  Quantity \
      0     17850.0   536365.0   WHITE HANGING HEART T-LIGHT HOLDER       6.0
      1     17850.0   536365.0                  WHITE METAL LANTERN       6.0
      2     17850.0   536365.0       CREAM CUPID HEARTS COAT HANGER       8.0
      3     17850.0   536365.0  KNITTED UNION FLAG HOT WATER BOTTLE       6.0
      4     17850.0   536365.0         RED WOOLLY HOTTIE WHITE HEART.      6.0

        InvoiceDate  UnitPrice         Country   Sales CohortGroup
      0  2010-12-01       2.55  United Kingdom   15.30     2010-12
      1  2010-12-01       3.39  United Kingdom   20.34     2010-12
      2  2010-12-01       2.75  United Kingdom   22.00     2010-12
      3  2010-12-01       3.39  United Kingdom   20.34     2010-12
      4  2010-12-01       3.39  United Kingdom   20.34     2010-12
```

```python
[21]: df.set_index('CustomerID', inplace=True)

df['OrderPeriod'] = df.groupby(level=0)['InvoiceDate'].min().apply(lambda x: x.
 ↪strftime('%Y-%m'))
df.reset_index(inplace=True)
df.head()
```

```
[21]:    CustomerID  InvoiceNo                          Description  Quantity \
      0     17850.0   536365.0   WHITE HANGING HEART T-LIGHT HOLDER       6.0
      1     17850.0   536365.0                  WHITE METAL LANTERN       6.0
      2     17850.0   536365.0       CREAM CUPID HEARTS COAT HANGER       8.0
      3     17850.0   536365.0  KNITTED UNION FLAG HOT WATER BOTTLE       6.0
      4     17850.0   536365.0         RED WOOLLY HOTTIE WHITE HEART.      6.0

        InvoiceDate  UnitPrice         Country   Sales CohortGroup OrderPeriod
      0  2010-12-01       2.55  United Kingdom   15.30     2010-12     2010-12
      1  2010-12-01       3.39  United Kingdom   20.34     2010-12     2010-12
      2  2010-12-01       2.75  United Kingdom   22.00     2010-12     2010-12
      3  2010-12-01       3.39  United Kingdom   20.34     2010-12     2010-12
      4  2010-12-01       3.39  United Kingdom   20.34     2010-12     2010-12
```

```python
[22]: df['OrderMonth'] = df['InvoiceDate'].dt.to_period('M')
df['CohortGroup'] = df.groupby('CustomerID')['InvoiceDate'] \
                .transform('min') \
                .dt.to_period('M')
```

```python
[23]: df.head()
```

```
[23]:    CustomerID  InvoiceNo                          Description  Quantity \
      0     17850.0   536365.0   WHITE HANGING HEART T-LIGHT HOLDER       6.0
      1     17850.0   536365.0                  WHITE METAL LANTERN       6.0
      2     17850.0   536365.0       CREAM CUPID HEARTS COAT HANGER       8.0
      3     17850.0   536365.0  KNITTED UNION FLAG HOT WATER BOTTLE       6.0
```

```
4    17850.0   536365.0        RED WOOLLY HOTTIE WHITE HEART.       6.0

   InvoiceDate  UnitPrice        Country  Sales CohortGroup OrderPeriod  \
0   2010-12-01       2.55  United Kingdom  15.30     2010-12     2010-12
1   2010-12-01       3.39  United Kingdom  20.34     2010-12     2010-12
2   2010-12-01       2.75  United Kingdom  22.00     2010-12     2010-12
3   2010-12-01       3.39  United Kingdom  20.34     2010-12     2010-12
4   2010-12-01       3.39  United Kingdom  20.34     2010-12     2010-12

   OrderMonth
0     2010-12
1     2010-12
2     2010-12
3     2010-12
4     2010-12
```

[24]:
```python
from operator import attrgetter
import matplotlib.colors as mcolors
df_cohort = df.groupby(['CohortGroup', 'OrderMonth']) \
              .agg(n_customers=('CustomerID', 'count')) \
              .reset_index(drop=False)
df_cohort['PeriodIndex'] = (df_cohort.OrderMonth - df_cohort.CohortGroup).
 ↪apply(attrgetter('n'))

def get_data(df_cohort):
    cohort_pivot = df_cohort.pivot_table(index = 'CohortGroup',
                                         columns = 'PeriodIndex',
                                         values = 'n_customers')
    #cohort_pivot.iloc[:,0]= (cohort_pivot.iloc[:,0].divide(25, axis = 0)).
 ↪astype(int)
    #cohort_pivot.iloc[:,1:] = cohort_pivot.iloc[:,0:].divide(2, axis = 0)
    cohort_pivot = cohort_pivot.iloc[4:-1,:-5]
    return cohort_pivot

cohort_pivot_ = get_data(df_cohort)
cohort_pivot_
```

[24]:
```
PeriodIndex        0       1       2       3       4       5       6       7
CohortGroup
2011-04       7320.0  1435.0  1329.0  1299.0  1395.0  2088.0  1935.0  2215.0
2011-05       6087.0  1123.0   890.0   920.0  1494.0  2345.0  1946.0   802.0
2011-06       5652.0   939.0   733.0  1513.0  1723.0  2504.0   622.0     NaN
2011-07       5024.0   525.0  1353.0  1387.0  1839.0   529.0     NaN     NaN
2011-08       4822.0  1608.0  2872.0  2780.0   905.0     NaN     NaN     NaN
2011-09       8343.0  2483.0  2719.0   874.0     NaN     NaN     NaN     NaN
2011-10      11878.0  3643.0   907.0     NaN     NaN     NaN     NaN     NaN
2011-11      11249.0  1152.0     NaN     NaN     NaN     NaN     NaN     NaN
```

```python
[25]: cohort_size = cohort_pivot_.iloc[:,0]
      retention_matrix = cohort_pivot_.divide(cohort_size, axis = 0)
```

```python
[27]: retention_matrix
```

```
[27]: PeriodIndex     0         1         2         3         4         5         6  \
      CohortGroup
      2011-04       1.0  0.196038  0.181557  0.177459  0.190574  0.285246  0.264344
      2011-05       1.0  0.184492  0.146213  0.151142  0.245441  0.385247  0.319698
      2011-06       1.0  0.166136  0.129689  0.267693  0.304848  0.443029  0.110050
      2011-07       1.0  0.104498  0.269307  0.276075  0.366043  0.105295       NaN
      2011-08       1.0  0.333472  0.595603  0.576524  0.187681       NaN       NaN
      2011-09       1.0  0.297615  0.325902  0.104758       NaN       NaN       NaN
      2011-10       1.0  0.306701  0.076360       NaN       NaN       NaN       NaN
      2011-11       1.0  0.102409       NaN       NaN       NaN       NaN       NaN

      PeriodIndex          7
      CohortGroup
      2011-04       0.302596
      2011-05       0.131756
      2011-06            NaN
      2011-07            NaN
      2011-08            NaN
      2011-09            NaN
      2011-10            NaN
      2011-11            NaN
```

```python
[28]: import seaborn as sns
```

```python
[29]: with sns.axes_style("white"):
          fig, ax = plt.subplots(1, 2, figsize=(12, 8), sharey=True,
      ↪gridspec_kw={'width_ratios': [1, 20]})

          # retention matrix
          sns.heatmap(retention_matrix,
                      mask=retention_matrix.isnull(),
                      annot=True,
                      fmt='.0%',
                      cmap='RdYlGn',
                      ax=ax[1])
          ax[1].set_title('Monthly Cohorts: Customer Retention', fontsize=16)
          ax[1].set(xlabel='# of periods',
                    ylabel='')

          # cohort size
          cohort_size_df = pd.DataFrame(cohort_size).rename(columns={0:
      ↪'cohort_size'})
```
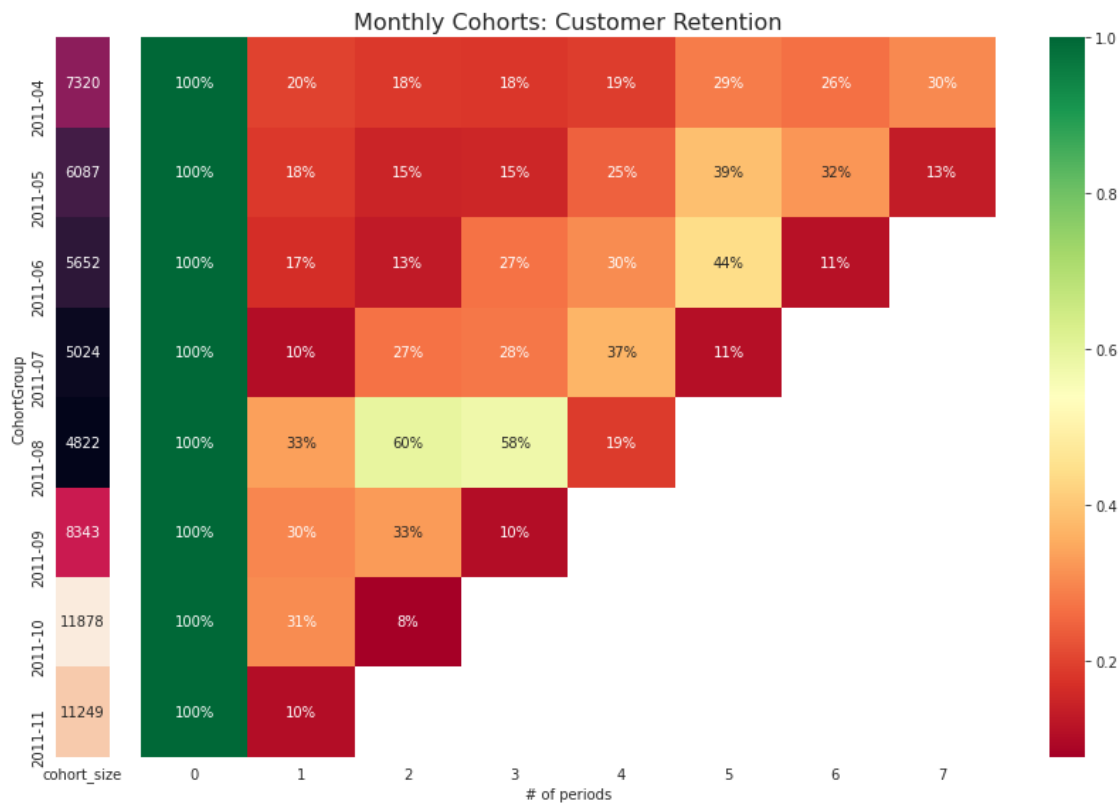
```
white_cmap = mcolors.ListedColormap(['white'])
sns.heatmap(cohort_size_df,
            annot=True,
            cbar=False,
            fmt='g',
            #cmap=white_cmap,
            ax=ax[0])

fig.tight_layout()
```


Monthly Cohorts: Customer Retention

[30]:
```
df['PurchaseSize'] = df['Quantity']*df['UnitPrice']
```

[31]:
```
def get_date_int(df, col):
    '''Helper function to extract year and month from the column (col) from a
    →dataframe (df)'''

    year = df[col].map(lambda x: x.year)
    month = df[col].map(lambda x: x.month)

    return year, month
```

```
[32]:  def get_cohort_index(df, transaction_col, cohort_col):
           '''
           Calculates Cohort Index based on monthly data

           Input:
               df: dataframe
               transaction_col: column with which time offset will be calculated
               cohort_col: cohort column (the acquisition year-month for each customer)
           Output:
               cohort_index: how many months elapsed since the acquisition
           '''

           transaction_year, transaction_month = get_date_int(df, transaction_col)
           cohort_year, cohort_month = get_date_int(df, cohort_col)

           year_diff = transaction_year - cohort_year
           month_diff = transaction_month - cohort_month

           # adding 1 so that our index starts with 1 instead of 0 for easier␣
       ↪interpretation
           cohort_index = year_diff * 12 + month_diff + 1

           return cohort_index
```

```
[42]:  df['CohortIndex'] = get_cohort_index(df, 'OrderMonth', 'CohortGroup')
```

```
[43]:  df.tail()
```

```
[43]:          CustomerID  InvoiceNo                        Description  Quantity  \
       541905     12680.0   581587.0      CHILDREN'S APRON DOLLY GIRL       6.0
       541906     12680.0   581587.0      CHILDRENS CUTLERY DOLLY GIRL      4.0
       541907     12680.0   581587.0   CHILDRENS CUTLERY CIRCUS PARADE      4.0
       541908     12680.0   581587.0      BAKING SET 9 PIECE RETROSPOT      3.0
       541909         NaN        NaN                               NaN      NaN

              InvoiceDate  UnitPrice Country  Sales CohortGroup OrderPeriod  \
       541905  2011-12-09       2.10  France  12.60     2011-08     2011-08
       541906  2011-12-09       4.15  France  16.60     2011-08     2011-08
       541907  2011-12-09       4.15  France  16.60     2011-08     2011-08
       541908  2011-12-09       4.95  France  14.85     2011-08     2011-08
       541909         NaT        NaN     NaN   0.00         NaT         NaN

              OrderMonth  PurchaseSize  CohortIndex
       541905    2011-12         12.60          5.0
       541906    2011-12         16.60          5.0
       541907    2011-12         16.60          5.0
       541908    2011-12         14.85          5.0
```

```
541909          NaT          NaN          NaN
```

```
[44]: consumers_in_cohorts = df.groupby(['CohortGroup', 'CohortIndex'])['CustomerID'].
      ↪nunique()
```

```
[45]: consumers_in_cohorts = consumers_in_cohorts.reset_index()
      consumers_in_cohorts.head()
```

```
[45]:    CohortGroup  CohortIndex  CustomerID
      0      2010-12          1.0         948
      1      2010-12          2.0         362
      2      2010-12          3.0         317
      3      2010-12          4.0         367
      4      2010-12          5.0         341
```

```
[46]: cohort_fin = consumers_in_cohorts.pivot(index='CohortGroup',␣
      ↪columns='CohortIndex', values='CustomerID')
```

```
[47]: cohort_fin
```

```
[47]: CohortIndex   1.0    2.0    3.0    4.0    5.0    6.0    7.0    8.0    9.0   \
      CohortGroup
      2010-12      948.0  362.0  317.0  367.0  341.0  376.0  360.0  336.0  336.0
      2011-01      421.0  101.0  119.0  102.0  138.0  126.0  110.0  108.0  131.0
      2011-02      380.0   94.0   73.0  106.0  102.0   94.0   97.0  107.0   98.0
      2011-03      440.0   84.0  112.0   96.0  102.0   78.0  116.0  105.0  127.0
      2011-04      299.0   68.0   66.0   63.0   62.0   71.0   69.0   78.0   25.0
      2011-05      279.0   66.0   48.0   48.0   60.0   68.0   74.0   29.0    NaN
      2011-06      235.0   49.0   44.0   64.0   58.0   79.0   24.0    NaN    NaN
      2011-07      191.0   40.0   39.0   44.0   52.0   22.0    NaN    NaN    NaN
      2011-08      167.0   42.0   42.0   42.0   23.0    NaN    NaN    NaN    NaN
      2011-09      298.0   89.0   97.0   36.0    NaN    NaN    NaN    NaN    NaN
      2011-10      352.0   93.0   46.0    NaN    NaN    NaN    NaN    NaN    NaN
      2011-11      321.0   43.0    NaN    NaN    NaN    NaN    NaN    NaN    NaN
      2011-12       41.0    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN

      CohortIndex   10.0   11.0   12.0   13.0
      CohortGroup
      2010-12      374.0  354.0  474.0  260.0
      2011-01      146.0  155.0   63.0    NaN
      2011-02      119.0   35.0    NaN    NaN
      2011-03       39.0    NaN    NaN    NaN
      2011-04        NaN    NaN    NaN    NaN
      2011-05        NaN    NaN    NaN    NaN
      2011-06        NaN    NaN    NaN    NaN
      2011-07        NaN    NaN    NaN    NaN
      2011-08        NaN    NaN    NaN    NaN
```

```
2011-09        NaN     NaN     NaN     NaN
2011-10        NaN     NaN     NaN     NaN
2011-11        NaN     NaN     NaN     NaN
2011-12        NaN     NaN     NaN     NaN
```

[48]:
```python
# access the first cohort values
cohort_sizes = cohort_fin[1]
cohort_sizes[:10]
```

[48]:
```
CohortGroup
2010-12    948.0
2011-01    421.0
2011-02    380.0
2011-03    440.0
2011-04    299.0
2011-05    279.0
2011-06    235.0
2011-07    191.0
2011-08    167.0
2011-09    298.0
Freq: M, Name: 1.0, dtype: float64
```

[49]:
```python
# calc retention by dividing all values by respective cohort_size
# axis=0 so that we divide along the row axis
# round to 3 and mutiply by 100 to get percentage for easy interpretation
retention = cohort_fin.divide(cohort_sizes, axis=0).round(3)
retention
```

[49]:
```
CohortIndex  1.0    2.0    3.0    4.0    5.0    6.0    7.0    8.0    9.0   \
CohortGroup
2010-12      1.0  0.382  0.334  0.387  0.360  0.397  0.380  0.354  0.354
2011-01      1.0  0.240  0.283  0.242  0.328  0.299  0.261  0.257  0.311
2011-02      1.0  0.247  0.192  0.279  0.268  0.247  0.255  0.282  0.258
2011-03      1.0  0.191  0.255  0.218  0.232  0.177  0.264  0.239  0.289
2011-04      1.0  0.227  0.221  0.211  0.207  0.237  0.231  0.261  0.084
2011-05      1.0  0.237  0.172  0.172  0.215  0.244  0.265  0.104    NaN
2011-06      1.0  0.209  0.187  0.272  0.247  0.336  0.102    NaN    NaN
2011-07      1.0  0.209  0.204  0.230  0.272  0.115    NaN    NaN    NaN
2011-08      1.0  0.251  0.251  0.251  0.138    NaN    NaN    NaN    NaN
2011-09      1.0  0.299  0.326  0.121    NaN    NaN    NaN    NaN    NaN
2011-10      1.0  0.264  0.131    NaN    NaN    NaN    NaN    NaN    NaN
2011-11      1.0  0.134    NaN    NaN    NaN    NaN    NaN    NaN    NaN
2011-12      1.0    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN

CohortIndex  10.0   11.0  12.0   13.0
CohortGroup
2010-12      0.395  0.373  0.50  0.274
```

```
2011-01    0.347   0.368   0.15    NaN
2011-02    0.313   0.092   NaN     NaN
2011-03    0.089   NaN     NaN     NaN
2011-04    NaN     NaN     NaN     NaN
2011-05    NaN     NaN     NaN     NaN
2011-06    NaN     NaN     NaN     NaN
2011-07    NaN     NaN     NaN     NaN
2011-08    NaN     NaN     NaN     NaN
2011-09    NaN     NaN     NaN     NaN
2011-10    NaN     NaN     NaN     NaN
2011-11    NaN     NaN     NaN     NaN
2011-12    NaN     NaN     NaN     NaN
```

[50]:
```python
df['PurchaseSize'] = df['Quantity']*df['UnitPrice']
```

[51]:
```python
average_purchase_size = df.groupby(['CohortGroup', 'CohortIndex']).PurchaseSize.
 ↪mean()
average_purchase_size = average_purchase_size.reset_index()
```

[52]:
```python
average_purchase_size.head()
```

[52]:
```
   CohortGroup  CohortIndex  PurchaseSize
0    2010-12           1.0     20.655643
1    2010-12           2.0     24.962459
2    2010-12           3.0     25.139521
3    2010-12           4.0     24.978069
4    2010-12           5.0     19.909629
```

[53]:
```python
average_purchase_size_fin = average_purchase_size.pivot(index='CohortGroup',␣
 ↪columns='CohortIndex', values='PurchaseSize').round(1)
```
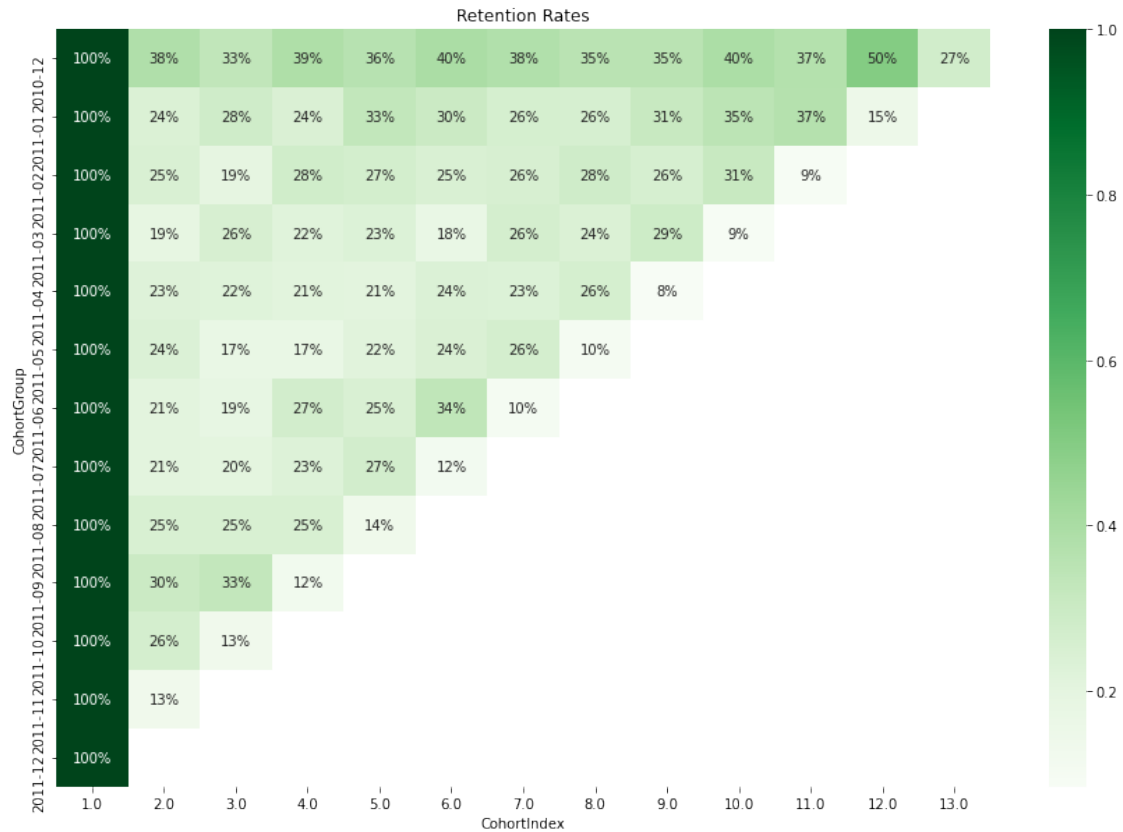
[54]:
```python
average_purchase_size_fin
```

[54]:

CohortIndex	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0 \
CohortGroup											
2010-12	20.7	25.0	25.1	25.0	19.9	25.5	26.5	25.4	26.0	31.1	30.6
2011-01	18.4	23.5	20.3	17.5	21.5	25.4	24.4	24.5	18.7	20.1	18.7
2011-02	17.0	17.0	19.3	18.8	16.1	15.1	21.6	20.9	17.5	20.6	21.5
2011-03	17.0	17.9	21.5	17.0	19.2	18.1	21.7	17.3	15.4	11.4	NaN
2011-04	16.4	20.2	18.8	18.4	18.6	14.2	14.6	15.3	11.8	NaN	NaN
2011-05	19.0	15.7	21.2	19.4	17.8	14.1	16.1	13.3	NaN	NaN	NaN
2011-06	16.4	14.5	19.0	19.8	15.0	15.9	12.7	NaN	NaN	NaN	NaN
2011-07	13.1	21.2	11.4	12.4	10.5	11.4	NaN	NaN	NaN	NaN	NaN
2011-08	16.1	11.9	11.5	14.4	15.6	NaN	NaN	NaN	NaN	NaN	NaN
2011-09	18.4	10.4	13.2	14.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011-10	13.0	10.8	13.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011-11	11.9	12.9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
     2011-12       26.9   NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN

     CohortIndex  12.0   13.0
     CohortGroup
     2010-12       22.2   24.7
     2011-01       18.4   NaN
     2011-02        NaN   NaN
     2011-03        NaN   NaN
     2011-04        NaN   NaN
     2011-05        NaN   NaN
     2011-06        NaN   NaN
     2011-07        NaN   NaN
     2011-08        NaN   NaN
     2011-09        NaN   NaN
     2011-10        NaN   NaN
     2011-11        NaN   NaN
     2011-12        NaN   NaN
```
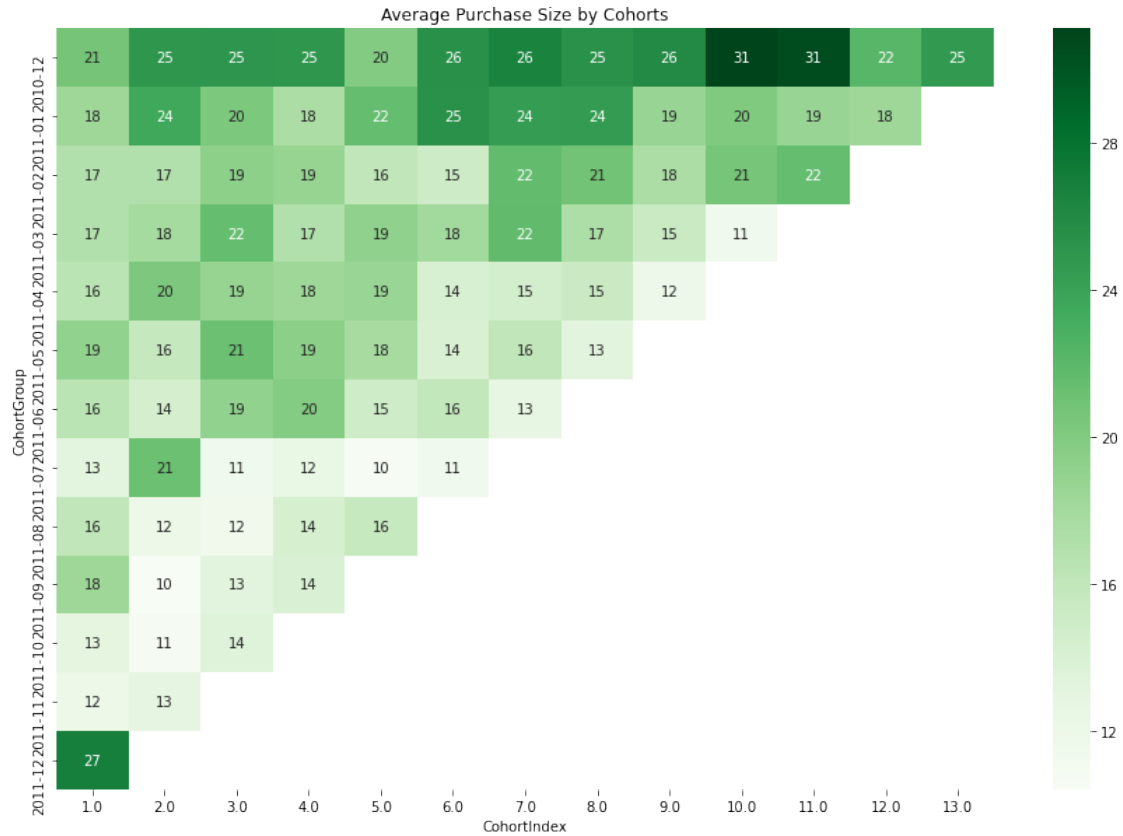
[55]:
```python
retention.index = retention.index.strftime('%Y-%m')
```

[56]:
```python
plt.figure(figsize=(15, 10))
plt.title('Retention Rates')
sns.heatmap(data=retention, fmt='.0%', annot=True, cmap='Greens')
plt.show()
```

Retention Rates

CohortGroup	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0
2010-12	100%	38%	33%	39%	36%	40%	38%	35%	35%	40%	37%	50%	27%
2011-01	100%	24%	28%	24%	33%	30%	26%	26%	31%	35%	37%	15%	
2011-02	100%	25%	19%	28%	27%	25%	26%	28%	26%	31%	9%		
2011-03	100%	19%	26%	22%	23%	18%	26%	24%	29%	9%			
2011-04	100%	23%	22%	21%	21%	24%	23%	26%	8%				
2011-05	100%	24%	17%	17%	22%	24%	26%	10%					
2011-06	100%	21%	19%	27%	25%	34%	10%						
2011-07	100%	21%	20%	23%	27%	12%							
2011-08	100%	25%	25%	25%	14%								
2011-09	100%	30%	33%	12%									
2011-10	100%	26%	13%										
2011-11	100%	13%											
2011-12	100%												

CohortIndex

[57]:
```python
average_purchase_size_fin.index = average_purchase_size_fin.index.
↪strftime('%Y-%m')
```

[58]:
```python
plt.figure(figsize=(15, 10))
plt.title('Average Purchase Size by Cohorts')
sns.heatmap(data=average_purchase_size_fin, annot=True, cmap='Greens')
plt.show()
```

Average Purchase Size by Cohorts

```
[59]: cohort_fin.to_csv('cohort_fin.csv', index=False)
```

```
[60]: retention.to_csv('retention.csv', index=False)
```