Introduction

The goal of this study is to see the emerging markets in the Retail Shop Datasets through cluster analysis.

1. Cleaning the Datasets

There are 3 datasets for this study: customers, product categories and transactions. Some columns of the datasets were transformed to lowercase and renamed for easier understanding. Rows with null values were also dropped since they would give no value at all.

At the end of this section, these three are merged into one dataset using SQL syntax enabled by the pandasql library.

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

from IPython.display import Image
from IPython.core.display import HTML

sns.set(rc={'figure.figsize':(10,10)})

%matplotlib inline
```

In [2]:
```
# Load the csv datasets into pandas dataframes
dataset_dir = os.path.join(os.getcwd(), 'dataset')
```

In [3]:
```
# Create a function to drop rows from a dataframe. This will be used throughout this noteb
ook.
# rows_to_drop = transact_df[transact_df['qty'] < 1].index
# transact_df.drop(rows_to_drop, axis=0, inplace=True)
# transact_df.reset_index(drop=True, inplace=True)

def drop_rows_reset_index_inplace(rows_index, dataframe):
    dataframe.drop(rows_index, axis=0, inplace=True)
    dataframe.reset_index(drop=True, inplace=True)
```

1.1. The Customers Dataset

In [4]:
```
customer_csv = os.path.join(dataset_dir, 'Customer.csv')
customer_df = pd.read_csv(customer_csv)
```

In [5]:
```
customer_df_col_map = {
    'customer_Id': 'customer_id',
    'DOB': 'birth_year',
    'Gender': 'gender'
}
customer_df.rename(columns=customer_df_col_map, inplace=True)
```

In [6]:
```
# Check for and drop the rows with null values
customer_df.isna().sum()
```

Out[6]:
```
customer_id    0
birth_year     0
gender         2
city_code      2
dtype: int64
```

In [7]:
```
customer_df.dropna(inplace=True)
customer_df.reset_index(drop=True, inplace=True)
```

In [8]:
```
# Transform the former DOB column to Year only
customer_df['birth_year'] = customer_df['birth_year'].apply(lambda x: datetime.strptime(x,
'%d-%m-%Y').year)
```

In [9]:
```
customer_df.head()
```

Out[9]:

	customer_id	birth_year	gender	city_code
0	268408	1970	M	4.0
1	269696	1970	F	8.0
2	268159	1970	F	8.0
3	270181	1970	F	2.0
4	268073	1970	M	1.0

1.2. Product Categories Dataset

In [10]:
```
prodcat_csv = os.path.join(dataset_dir, 'prod_cat_info.csv')
prodcat_df = pd.read_csv(prodcat_csv)
```

In [11]:
```
# Rename column prod_subcat to prod_sub_cat for consistency
prodcat_df.rename(columns={'prod_subcat': 'prod_sub_cat'}, inplace=True)
```

Notice that "prod_cat" column with values "Clothing", "Footwear" and "Bags" all have the same "prod_sub_cat" values. To avoid confusion later on in the clustering, the "prod_sub_cat" values must be unique.

In [12]:
```
prodcat_df[prodcat_df['prod_cat'].isin(["Clothing", "Footwear", "Bags"])]
```

Out[12]:

	prod_cat_code	prod_cat	prod_sub_cat_code	prod_sub_cat
0	1	Clothing	4	Mens
1	1	Clothing	1	Women
2	1	Clothing	3	Kids
3	2	Footwear	1	Mens
4	2	Footwear	3	Women
5	2	Footwear	4	Kids
11	4	Bags	1	Mens
12	4	Bags	4	Women

Function rename_prod_sub_cat is created to transform the "prod_sub_cat" values.

In [13]:
```
def rename_prod_sub_cat(prod_cat, old_sub_cat, new_sub_cat):
    # Locate the row that corresponds to parm prod_cat and old_sub_cat
    # Replace the prod_sub_cat of that row with the new_sub_cat
    prodcat_df.loc[(prodcat_df['prod_cat'] == prod_cat) &
                   (prodcat_df['prod_sub_cat'] == old_sub_cat),
                   'prod_sub_cat'] = new_sub_cat
```

The dictionary 'prod_sub_cat_names' contains the constants to use for the function. Loop its key's and values and pass them to the function.

In [14]:
```
prod_sub_cat_names = {
    # new_sub_cat : [prod_cat, old_sub_cat]
    'clothing_men': ['Clothing', 'Mens'],
    'clothing_women': ['Clothing', 'Women'],
    'clothing_kids': ['Clothing', 'Kids'],
    'footwear_men': ['Footwear', 'Mens'],
    'footwear_women': ['Footwear', 'Women'],
    'footwear_kids': ['Footwear', 'Kids'],
    'bags_men': ['Bags', 'Mens'],
    'bags_women': ['Bags', 'Women'],
}

for key, val_list in prod_sub_cat_names.items():
    rename_prod_sub_cat(prod_cat=val_list[0], old_sub_cat=val_list[1], new_sub_cat=key)
```

In [15]:
```
# Convert the rest of prod_sub_cat values to lowercase
prodcat_df['prod_sub_cat'] = prodcat_df['prod_sub_cat'].str.lower()
```

In [16]:
```
print(prodcat_df)
```

```
    prod_cat_code          prod_cat  prod_sub_cat_code         prod_sub_cat
0               1          Clothing                  4         clothing_men
1               1          Clothing                  1       clothing_women
2               1          Clothing                  3        clothing_kids
3               2          Footwear                  1         footwear_men
4               2          Footwear                  3       footwear_women
5               2          Footwear                  4        footwear_kids
6               3       Electronics                  4              mobiles
7               3       Electronics                  5            computers
8               3       Electronics                  8  personal appliances
9               3       Electronics                  9              cameras
10              3       Electronics                 10      audio and video
11              4              Bags                  1             bags_men
12              4              Bags                  4           bags_women
13              5             Books                  7              fiction
14              5             Books                 12             academic
15              5             Books                 10          non-fiction
16              5             Books                 11             children
17              5             Books                  3               comics
18              5             Books                  6                  diy
19              6  Home and kitchen                  2           furnishing
20              6  Home and kitchen                 10              kitchen
21              6  Home and kitchen                 11                 bath
22              6  Home and kitchen                 12                tools
```

1.3. Transactions Dataset

In [17]:
```
transact_csv = os.path.join(dataset_dir, 'Transactions.csv')
transact_df = pd.read_csv(transact_csv)
```

In [18]:
```
# Set to lowercase these columns of transact_df for easier reference
transact_df_col_map = {
    'cust_id': 'customer_id',
    'prod_subcat_code': 'prod_sub_cat_code',
    'Qty': 'qty',
    'Rate': 'rate',
    'Tax': 'tax',
    'Store_type': 'store_type',
    'tran_date': 'transact_date'
}
transact_df.rename(columns=transact_df_col_map, inplace=True)
```

There are some transactions where the "qty" columns is less than 1. These can be interpreted as returns or refunds. However, since this study aims to know the emerging markets, these refund rows will be dropped.

In [19]:
```
transact_df[transact_df['qty'] < 1].head()
```

Out[19]:

	transaction_id	customer_id	transact_date	prod_sub_cat_code	prod_cat_code	qty	rate	tax	total_am
0	80712190438	270351	28-02-2014	1	1	-5	-772	405.300	-4265.30
1	29258453508	270384	27-02-2014	5	3	-5	-1497	785.925	-8270.92
2	51750724947	273420	24-02-2014	6	5	-2	-791	166.110	-1748.11
3	93274880719	271509	24-02-2014	11	6	-3	-1363	429.345	-4518.34
4	51750724947	273420	23-02-2014	6	5	-2	-791	166.110	-1748.11

In [20]:
```
# Remove the rows where qty is less than 1
rows_to_drop = transact_df[transact_df['qty'] < 1].index
drop_rows_reset_index_inplace(rows_to_drop, transact_df)
```

Convert "transact_date" to a datetime object. This column has a mix of dates formatted in DD-MM-YYYY and DD/MM/YYYY, so pandas.to_datetime with dayfirst=True param is used

In [21]:
```
transact_df['transact_date'] = pd.to_datetime(transact_df['transact_date'], dayfirst=True)
```

Create new columns transact_year and transact_month that are derived from transact_date.

In [22]:
```
transact_df['transact_year'] = transact_df['transact_date'].dt.year
transact_df['transact_month'] = transact_df['transact_date'].dt.month
```

In [23]:
```
transact_df.head()
```

Out[23]:

	transaction_id	customer_id	transact_date	prod_sub_cat_code	prod_cat_code	qty	rate	tax	total_amt
0	29258453508	270384	2014-02-20	5	3	5	1497	785.925	8270.925
1	25455265351	267750	2014-02-20	12	6	3	1360	428.400	4508.400
2	1571002198	275023	2014-02-20	6	5	4	587	246.540	2594.540
3	36554696014	269345	2014-02-20	3	5	3	1253	394.695	4153.695
4	56814940239	268799	2014-02-20	7	5	5	368	193.200	2033.200

1.4. Merging the Datasets

The library pandasql is used to join the three datasets together using SQL. This is more precise and easier to read, because the SQL Joins allow more control than pandas.

In [24]:
```
# Use pandasql to easily merge the three datasets together
from pandasql import sqldf
```

In [25]:
```
merged_df = lambda query: sqldf(query, globals())
query = """
    SELECT
        t.*,
        c.birth_year,
        c.gender,
        c.city_code,
        p.prod_cat,
        p.prod_sub_cat
    FROM transact_df t
    INNER JOIN customer_df c
        on c.customer_id = t.customer_id
    INNER JOIN prodcat_df p
        ON p.prod_cat_code = t.prod_cat_code
        AND p.prod_sub_cat_code = t.prod_sub_cat_code
"""
merged_df = merged_df(query)
```

In [26]:
```
merged_df.head()
```

Out[26]:

	transaction_id	customer_id	transact_date	prod_sub_cat_code	prod_cat_code	qty	rate	tax	total_a
0	29258453508	270384	2014-02-20 00:00:00.000000	5	3	5	1497	785.925	8270.9
1	25455265351	267750	2014-02-20 00:00:00.000000	12	6	3	1360	428.400	4508.4
2	1571002198	275023	2014-02-20 00:00:00.000000	6	5	4	587	246.540	2594.5
3	36554696014	269345	2014-02-20 00:00:00.000000	3	5	3	1253	394.695	4153.6
4	56814940239	268799	2014-02-20 00:00:00.000000	7	5	5	368	193.200	2033.2

2. Exploring the Data

The goal is to find out which emerging market to look out for. As I narrowed the criteria, I trimmed the merged dataset until I got the ideal dataset for clustering. Aside from having a defined scope, this section also helped me run cluster analysis in R successfully with the hardware specifications of my machine.

2.1 Exploring the Merged Dataset

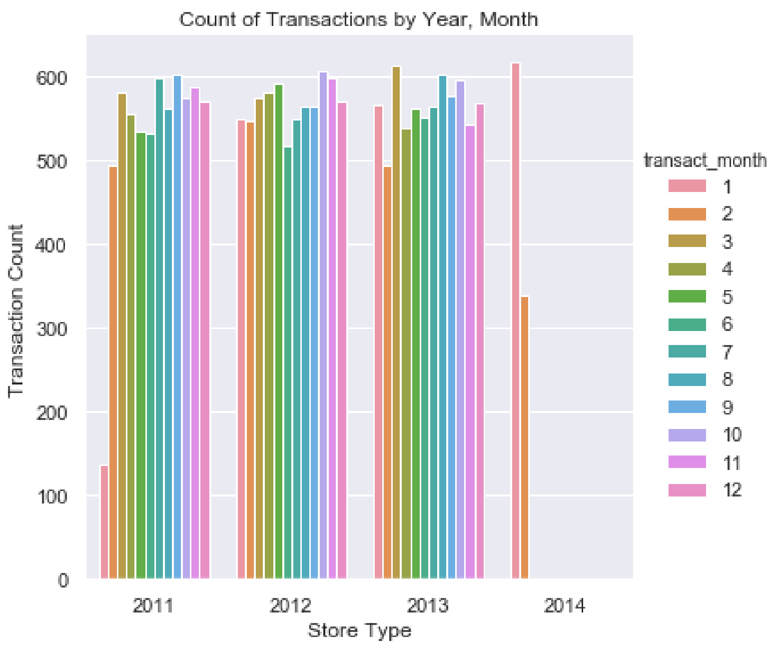
The merged_df has 19,905 rows. This needs to be narrowed down as mentioned above.

In [27]:
```
merged_df.shape
```

Out[27]:
```
(20860, 17)
```

As the 2014 transactions are up to Feburary only, remove all 2014 transactions from the dataframe.

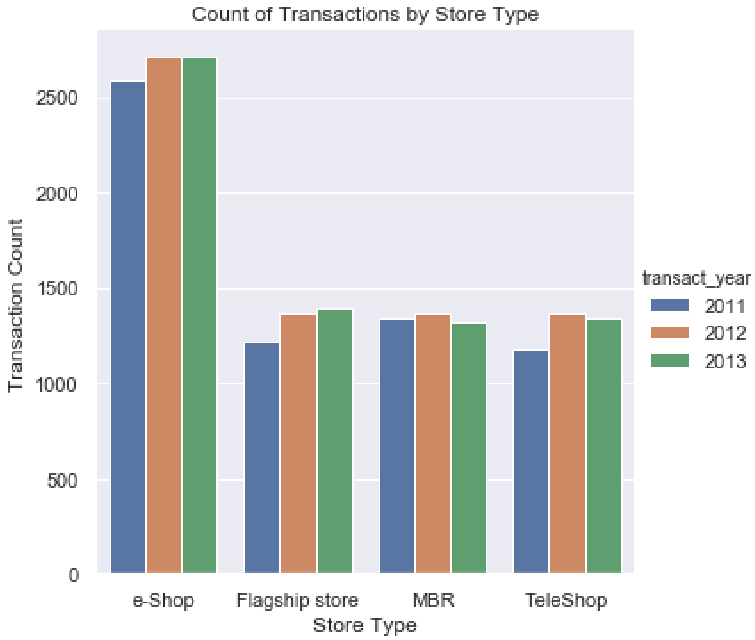
In [28]:
```
fig = sns.catplot(x='transact_year', kind='count', hue='transact_month' ,data=merged_df )
fig.set(xlabel='Store Type', ylabel="Transaction Count", title="Count of Transactions by Y
ear, Month");
```



In [29]:
```
rows_to_drop = merged_df[merged_df['transact_year'] == 2014].index
drop_rows_reset_index_inplace(rows_to_drop, merged_df)
```

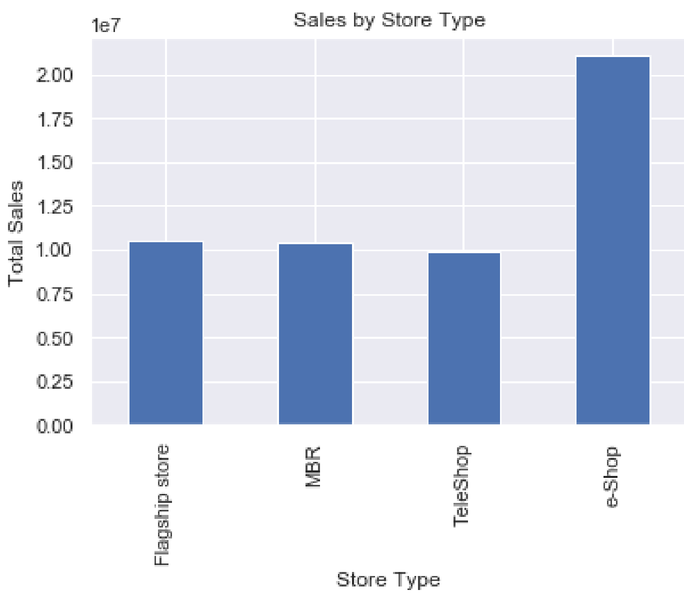
Majority of the transactions come from the 'e-Shop'. The number of rows to analyze is significant enough to focus only on the e-Shop.

In [30]:
```
fig = sns.catplot(x='store_type', kind='count', hue='transact_year', data=merged_df )
fig.set(xlabel='Store Type', ylabel="Transaction Count",title="Count of Transactions by St
ore Type");
```



In addition, the total sales from the e-Shop is the highest among the store types.

In [31]:
```
bar = merged_df.groupby('store_type').total_amt.sum().plot(kind='bar');
bar.set_title('Sales by Store Type')
bar.set_ylabel('Total Sales')
bar.set_xlabel('Store Type');
```



With these, a new dataframe called **"eshop_df"** is created. It is a copy of the "merged_df", but it will only contain rows where "store_type" is equal to "e_Shop".

In [32]:
```
eshop_df = merged_df.copy(deep=True)
rows_to_drop = eshop_df[eshop_df['store_type'] != 'e-Shop'].index
drop_rows_reset_index_inplace(rows_to_drop, eshop_df)
eshop_df.head()
```

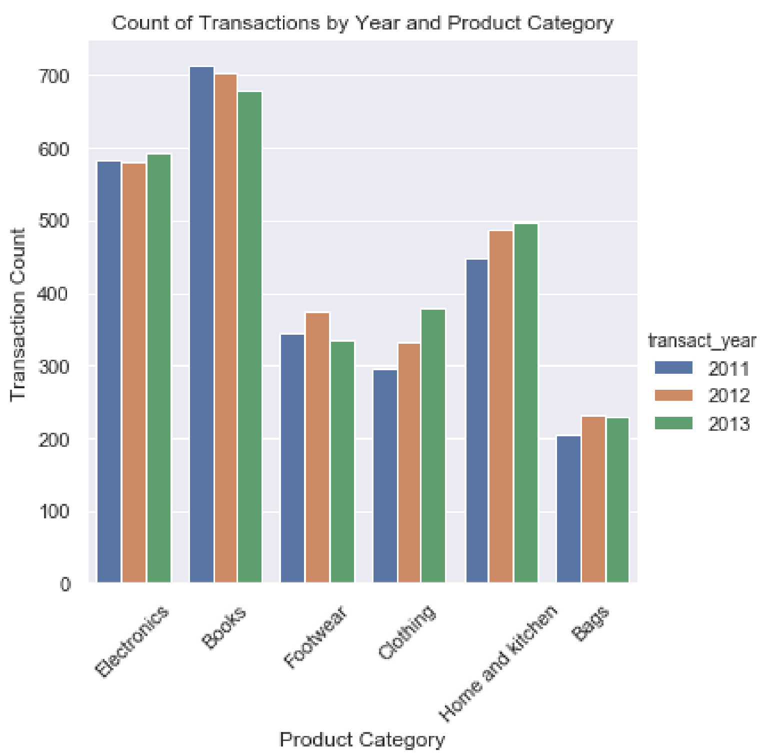
Out[32]:

	transaction_id	customer_id	transact_date	prod_sub_cat_code	prod_cat_code	qty	rate	tax	total_an
0	58387181112	275068	2013-12-31 00:00:00.000000	8	3	5	792	415.800	4375.8(
1	26100869804	273836	2013-12-31 00:00:00.000000	9	3	3	843	265.545	2794.5⁄
2	4116412179	269788	2013-12-31 00:00:00.000000	10	3	3	984	309.960	3261.9(
3	51849180620	273963	2013-12-31 00:00:00.000000	9	3	3	617	194.355	2045.3!
4	73514951834	269518	2013-12-31 00:00:00.000000	6	5	2	582	122.220	1286.2:

2.2 Exploring the E-Shop Dataset

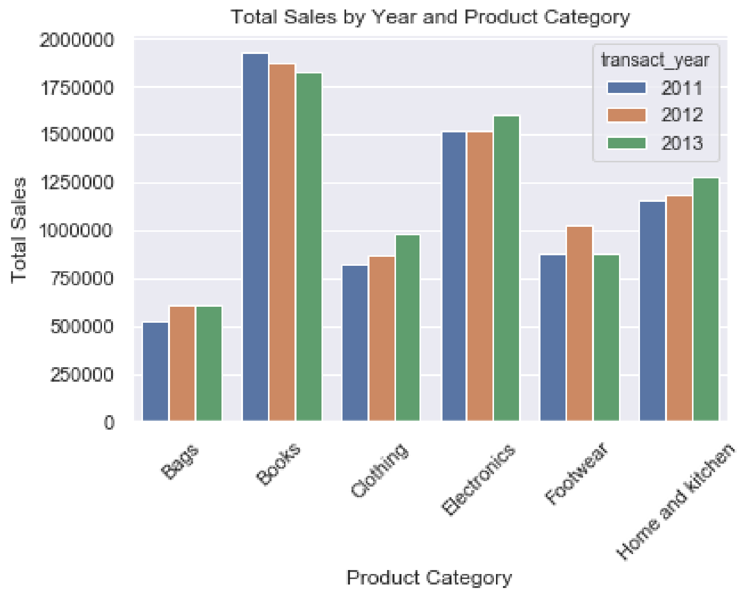
The chart below shows the Books category having the highest transaction count followed by Electronics, Home and Kitchen.

In [33]:
```
fig = sns.catplot(x='prod_cat', kind='count', hue='transact_year', data=eshop_df)
fig.set(xlabel='Product Category', ylabel="Transaction Count", title="Count of Transaction
s by Year and Product Category");
fig.set_xticklabels(rotation=45);
```



The same can be said when checking the store's total sales by year per product category.

In [34]:
```
grouped_df = eshop_df.groupby(['prod_cat', 'transact_year']).total_amt.sum().reset_index()
fig = sns.barplot(x='prod_cat', y='total_amt', hue='transact_year', data=grouped_df)
fig.set(xlabel='Product Category', ylabel="Total Sales", title="Total Sales by Year and Pr
oduct Category")
fig.set_xticklabels(fig.get_xticklabels(), rotation=45);
```



For Bags, Clothing and Home and Kitchen, they are the only product categories whose sales are increasing year by year. While their cumulative sales are lower than that of Books and Electronics, they are gaining traction in the E-Shop.

It is of interest to know the customers who buy from the Bags, Clothing and Home and Kitchen categories. There is a potential for these categories to grow further in sales, and knowing who the right customers are can drive this.

A new dataframe, **'rising_market_df'**, is created and is derived from the eshop_df.

In [35]:
```
rising_market_df = eshop_df.copy(deep=True)
rows_to_drop = rising_market_df[~rising_market_df['prod_cat'].isin(['Clothing', 'Home and
 kitchen', 'Bags'])].index
drop_rows_reset_index_inplace(rows_to_drop, rising_market_df)
rising_market_df.head()
```

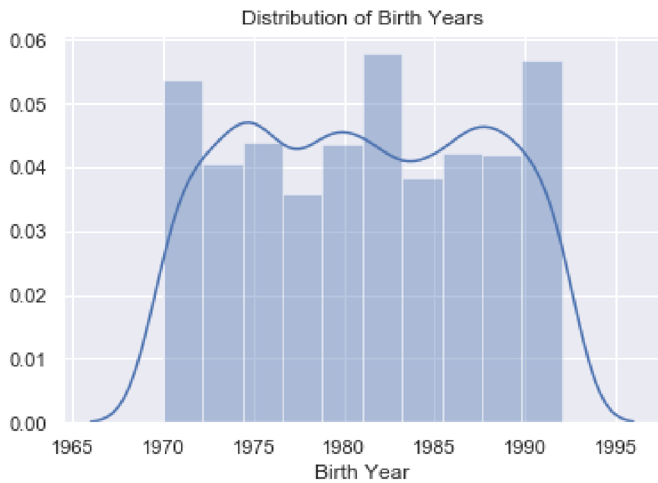
Out[35]:

	transaction_id	customer_id	transact_date	prod_sub_cat_code	prod_cat_code	qty	rate	tax	total_a
0	83963970126	274655	2013-12-31 00:00:00.000000	3	1	5	213	111.825	1176.8
1	51514545410	270709	2013-12-30 00:00:00.000000	1	1	5	1304	684.600	7204.6
2	83941716509	273771	2013-12-30 00:00:00.000000	4	1	5	725	380.625	4005.6
3	33215457342	272081	2013-12-29 00:00:00.000000	12	6	4	1079	453.180	4769.1
4	9488888491	267446	2013-12-29 00:00:00.000000	1	4	2	1336	280.560	2952.5

2.3. Exploring the Customers of the Rising Markets Dataset

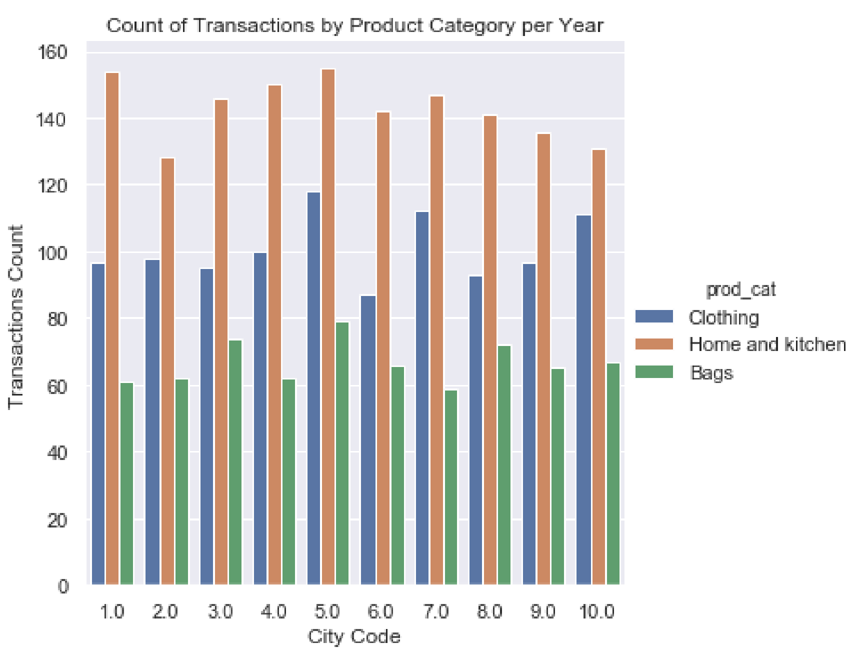
Below is the distribution of birth years of the customers. This is not normally distributed.

In [36]:
```
dist = sns.distplot(rising_market_df['birth_year'], bins=10)
dist.set(xlabel='Birth Year', title='Distribution of Birth Years');
```



All city codes follow the same trend. Home and Kitchen are the most bought category followd by Clothing and Bags.

In [37]:
```
fig = sns.catplot(x='city_code', kind='count', hue='prod_cat', data=rising_market_df)
fig.set(xlabel='City Code', ylabel="Transactions Count", title="Count of Transactions by P
roduct Category per Year");
```



Conclusion

At this point, no significant data can be gathered from the rising market dataset, since it was the result of the transactions dataset merged with products and customers.

Since each customer can have more than one transaction, the **customer must be profiled** based on his transaction history.

3. Customer Profiling

3.1. Pivoting the Data by Total Sales per Subcategory

Some customers have multiple transactions, and they have bought either from the same or from different subcategories. In this section, the rising_market_df is pivoted by total sales for each customer per subcategory.

In [38]:
```
pivot = pd.pivot_table(rising_market_df, values='total_amt', index=['customer_id'], column
s=['prod_sub_cat'],
                       aggfunc=np.sum, fill_value=0)
pivot.head()
```

Out[38]:

prod_sub_cat customer_id	bags_men	bags_women	bath	clothing_kids	clothing_men	clothing_women	furnishing	kitchen
266783	0.00	0.0	0.0	0.00	960.245	0.0	0.0	0.0
266794	2948.14	0.0	0.0	1533.74	0.000	0.0	0.0	0.0
266806	0.00	0.0	0.0	0.00	923.780	0.0	0.0	0.0
266807	0.00	0.0	0.0	0.00	495.040	0.0	0.0	0.0
266810	0.00	0.0	0.0	0.00	7542.730	0.0	0.0	0.0

3.2. Joining Customer Dataset Columns to the Pivot Table

Using `pandasql` , join the `customer_df` to the pivot table using the column `customer_id` . This creates the new dataframe, `profile_df`

In [39]:
```
profile_df = lambda query: sqldf(query, globals())
query = """
    SELECT
        p.*,
        c.gender,
        c.city_code,
        c.birth_year
    FROM pivot p
    INNER JOIN customer_df c
        ON c.customer_id = p.customer_id

"""
profile_df = profile_df(query)
```

Rename the subcategories columns of `profile_df` by adding the 'total_' prefix.

In [40]:
```
col_mapping = {
    'bags_men': 'total_bags_men',
    'bags_women': 'total_bags_women',
    'bath': 'total_bath',
    'clothing_kids': 'total_clothing_kids',
    'clothing_men': 'total_clothing_men',
    'clothing_women': 'total_clothing_women',
    'furnishing': 'total_furnishing',
    'kitchen': 'total_kitchen',
    'tools': 'total_tools'
}
profile_df.rename(columns=col_mapping, inplace=True)
profile_df.head()
```

Out[40]:

	customer_id	total_bags_men	total_bags_women	total_bath	total_clothing_kids	total_clothing_men	total_clo
0	268159	779.025	0.0	327.08	8141.64	0.0	
1	270181	0.000	0.0	0.00	0.00	0.0	
2	275152	0.000	0.0	0.00	0.00	1701.7	
3	270829	0.000	0.0	7602.40	0.00	0.0	
4	274593	0.000	0.0	0.00	0.00	0.0	

In [41]:
```
profile_df.describe(include='all')
```

Out[41]:

	customer_id	total_bags_men	total_bags_women	total_bath	total_clothing_kids	total_clothing_mer
count	2376.000000	2376.000000	2376.000000	2376.000000	2376.000000	2376.00000
unique	NaN	NaN	NaN	NaN	NaN	Na
top	NaN	NaN	NaN	NaN	NaN	Na
freq	NaN	NaN	NaN	NaN	NaN	Na
mean	270950.167929	368.784914	361.544280	364.604423	369.708072	406.41118
std	2459.351938	1202.264941	1206.009963	1156.046928	1206.322403	1278.38958
min	266783.000000	0.000000	0.000000	0.000000	0.000000	0.00000
25%	268805.750000	0.000000	0.000000	0.000000	0.000000	0.00000
50%	270880.500000	0.000000	0.000000	0.000000	0.000000	0.00000
75%	273139.500000	0.000000	0.000000	0.000000	0.000000	0.00000
max	275264.000000	8370.375000	11417.965000	10290.865000	9632.285000	8323.96500

3.3. Exporting the Profile to CSV

Export the `profile_df` to CSV format. This will be fed to the R Notebook where the clustering happens.

In [42]:
```
exported_dataset_dir = os.path.join(dataset_dir, 'exported')
if not os.path.exists(exported_dataset_dir):
    os.mkdir(exported_dataset_dir)

def export_df_to_csv(df, dir, filename):
    filepath = os.path.join(dir, filename)
    df.to_csv(filepath, index=False)
```

In [43]:
```
export_df_to_csv(profile_df, exported_dataset_dir, 'profile')
```

4. Cluster Analysis

4.1. Mixed Data Clustering in R

Rationale

R is the programming language used to cluster the two datasets. While python is robust and has many libraries, it simply cannot cluster mixed-data types. After researching, I used a method in R to discover the market segments of this retail store.

4.2. Viewing the R Notebook and R Scripts

R Notebook

This notebook has the steps I took for clustering. It is where I figured out the columns and the number of clusters to use. Please view this at:

4.3. Joining the Cluster Results with the Datasets

In this section, the results from the R Notebook are exported in CSV format and are then imported as pandas dataframes. They will be joined to the existing `profile_df` using pandasql.

In [44]:
```
profile_tsne_data_file = os.path.join(exported_dataset_dir, 'profile_tsne_data')
profile_tsne_df = pd.read_csv(profile_tsne_data_file)
profile_tsne_df.head()
```

Out[44]:

	Unnamed: 0	X	Y	cluster	customer_id
0	1	-8.097424	40.178476	1	268159
1	2	-2.231335	4.000221	1	270181
2	3	10.922439	-32.672245	2	275152
3	4	-7.468783	40.917174	1	270829
4	5	-13.048357	1.410031	3	274593

In [45]:
```
profile_cluster_df = lambda query: sqldf(query, globals())
query = """
    SELECT
        p.*,
        t.X,
        t.Y,
        t.cluster
    FROM profile_df p
    INNER JOIN profile_tsne_df t
        ON p.customer_id = t.customer_id
"""
profile_cluster_df = profile_cluster_df(query)
```

Add a new column, 'overlap_cluster' to easily identify customers that are in overlapping clusters.

In [46]:
```
profile_cluster_df['overlap_cluster'] = np.nan
```

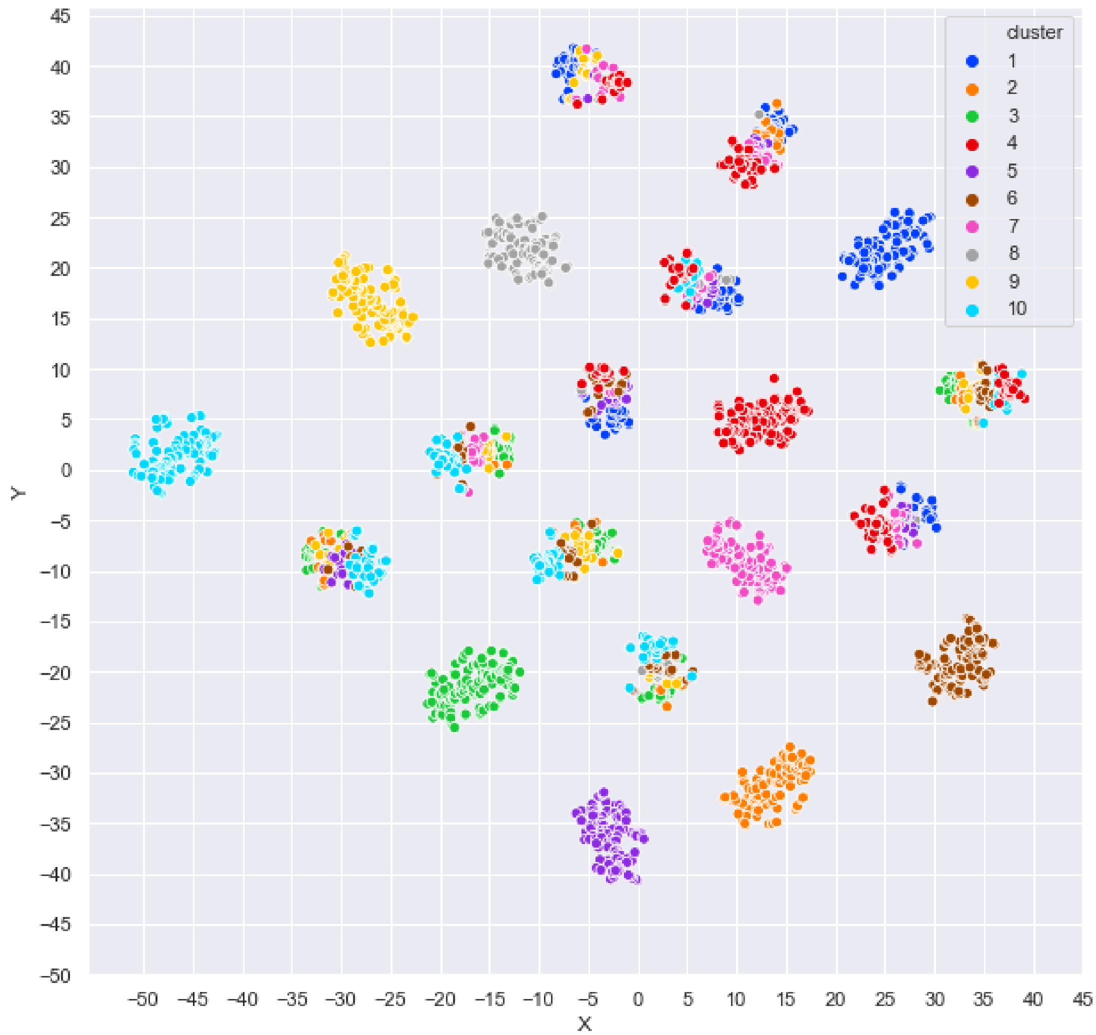
Change data type of columns 'city_code' and 'customer_id' to characters.

In [47]:
```
profile_cluster_df['city_code'].apply(str);
profile_cluster_df['customer_id'].apply(str);
```

5. Clustering Results Analysis

Now that the `profile_cluster_df` now has the x and y coordinates from rTSNE, use the scatterplot in seaborn's design to further differentiate the clusters.

In [48]:
```
fig, ax = plt.subplots(figsize=(10, 10))
repeat_scatter = sns.scatterplot(x='X', y='Y', hue='cluster', data=profile_cluster_df, leg
end='full', palette=sns.color_palette(palette='bright'), ax=ax)
repeat_scatter.set_xticks(np.arange(-50,50,5));
repeat_scatter.set_yticks(np.arange(-50,50,5));
```



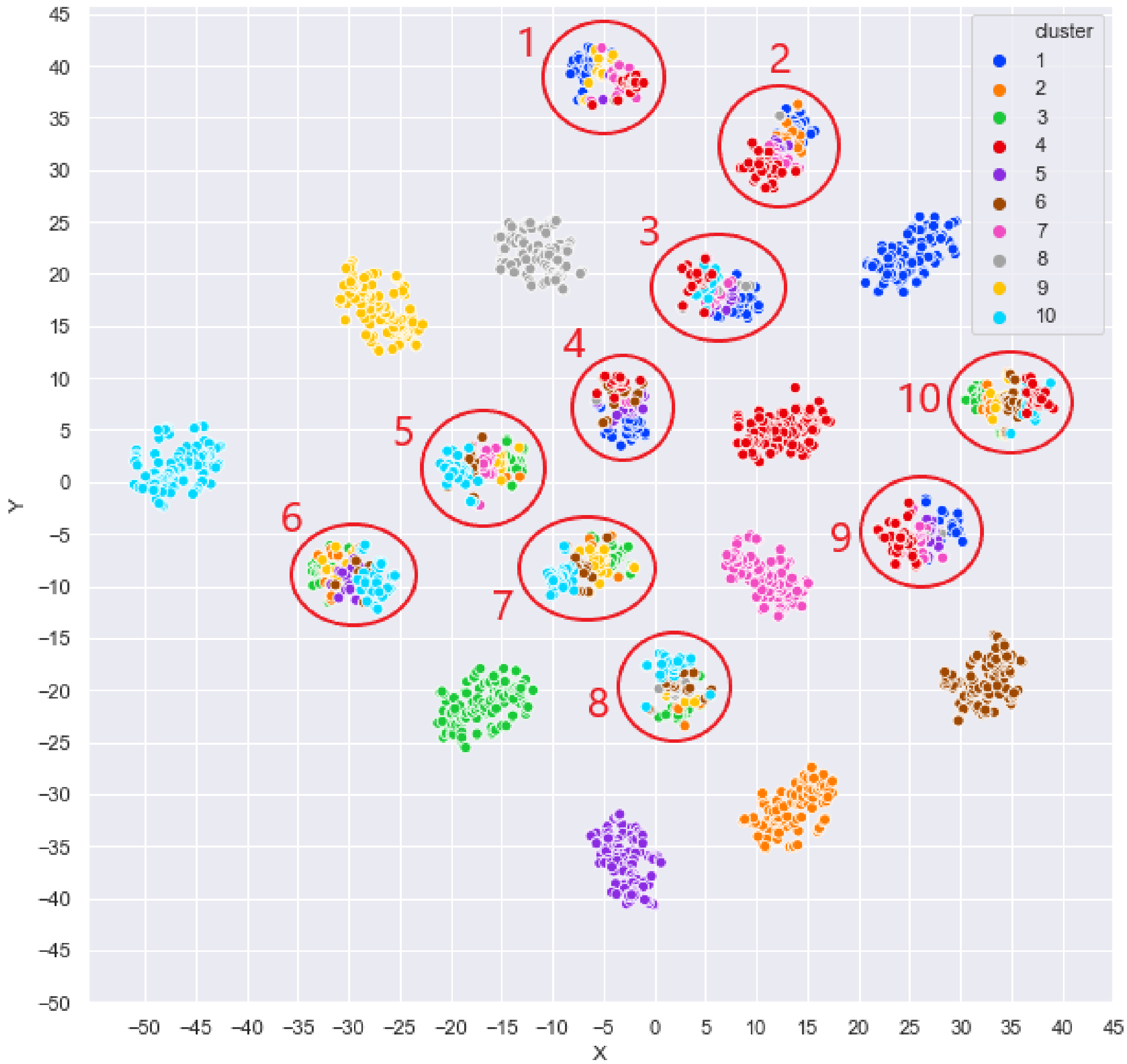
5.1. Overlapping Clusters

The image below is the same scatterplot but with the overlapping clusters encircled and labelled. Overlapping clusters are extracted by their coordinates on the scatter plot.

There are different clusters that have overlapped or have grouped with each other. We need to find out what they have in common to form a cluster different from their originally assigned clusters.

In [49]:
```
PATH = os.path.join(os.getcwd(), "cluster_profile_overlap.png")
Image(filename=PATH, width=650, height=650)
```

Out[49]:



For each labelled overlapping cluster, assign the ranges of their X and Y coordinates. The dictionary `overlap_df` initially contains `None` , but their values will be filled out later on.

```
In [50]: coordinates = {
    'overlap_1_df': {'X': [-10, 3], 'Y': [35, 44]},
    'overlap_2_df': {'X': [7, 17], 'Y': [27, 37]},
    'overlap_3_df': {'X': [1, 12], 'Y': [14, 24]},
    'overlap_4_df': {'X': [-8, 2], 'Y': [3, 13]},
    'overlap_5_df': {'X': [-23, -12], 'Y': [-4, 8]},
    'overlap_6_df': {'X': [-37, -23], 'Y': [-14, -4]},
    'overlap_7_df': {'X': [-13, 0], 'Y': [-13, -3]},
    'overlap_8_df': {'X': [-4, 8], 'Y': [-25, -14]},
    'overlap_9_df': {'X': [20, 33], 'Y': [-9, 1]},
    'overlap_10_df': {'X': [28, 42], 'Y': [3, 13]},
}
overlap_df = {
    'overlap_1_df': None,
    'overlap_2_df': None,
    'overlap_3_df': None,
    'overlap_4_df': None,
    'overlap_5_df': None,
    'overlap_6_df': None,
    'overlap_7_df': None,
    'overlap_8_df': None,
}
```

The lists `describe_columns` and `sub_cat_columns` are constant variables to easily refer to them in the iteration functions of both overlapping and solid clusters. They will be used in printing out the summaries.

```
In [51]: describe_columns = ['gender', 'city_code', 'birth_year', 'cluster', 'overlap_cluster']
sub_cat_columns = ['total_bags_men', 'total_bags_women', 'total_bath',
                   'total_clothing_kids', 'total_clothing_men', 'total_clothing_women',
                   'total_furnishing', 'total_kitchen', 'total_tools']
```

Each loop for the solid and overlapping clusters would have their summaries printed out. This function formats how they will be displayed in the output.

```
In [52]: def print_cluster_summary(cluster_name, describe_df, categ_summary_df):
    print('-------------------------------------------------')
    print(cluster_name.upper(), ' ', 'DESCRIPTION')
    print('-------------------------------------------------')
    print(describe_df)
    print('-------------------------------------------------')
    print(cluster_name.upper(), ' ', 'SUB CATEGORY SUMMARY')
    print('-------------------------------------------------')
    print(categ_summary_df)
    print('-------------------------------------------------')
    print('\n', '+++++++++++++++++++++++++++++++++++++++++++++++++++', '\n')
```

From the `profile_cluster_df` , the customers who belong to the overlapping clusters are extracted by using the mapped X and Y coordinates ranges. This subset will be the dataframe value of the dictionary key `overlap_df[cluster]` .

In the `profile_cluster_df` , customer_ids that are in the `overlap_df[cluster]` are assigned with their overlapping cluster names .

Finally, print the description and the total sales per subcategory of the `overlap_df[cluster]` .

In [53]:
```
for cluster, coord in coordinates.items():
    overlap_df[cluster] = profile_cluster_df[profile_cluster_df.columns]
    overlap_df[cluster] = overlap_df[cluster][(overlap_df[cluster]['X'].between(coord['X']
[0], coord['X'][1])) &
                                              (overlap_df[cluster]['Y'].between(coord['Y']
[0], coord['Y'][1]))
                                             ]

    # Assign the overlapping cluster number in the profile_cluster_df
    ol_cluster = 'OL' + str(cluster[8])
    profile_cluster_df.loc[profile_cluster_df['customer_id'].isin(overlap_df[cluster]['cus
tomer_id']),
                          'overlap_cluster'] = ol_cluster

    describe_cluster_df = overlap_df[cluster][describe_columns].describe(include='all')

    totals_list = overlap_df[cluster][sub_cat_columns].sum()
    mean_list = overlap_df[cluster][sub_cat_columns].mean()
    median_list = overlap_df[cluster][sub_cat_columns].median()

    categ_summary_data = zip(sub_cat_columns, totals_list, mean_list, median_list)
    overlap_categ_summary_df = pd.DataFrame(categ_summary_data, columns=['prod_sub_cat',
'total_sales', 'mean', 'median'])
    overlap_categ_summary_df.sort_values(by=['total_sales'], ascending=False, inplace=True
)

    print_cluster_summary(cluster, describe_cluster_df, overlap_categ_summary_df)
```

```
----------------------------------------------
OVERLAP_1_DF   DESCRIPTION
----------------------------------------------
       gender  city_code   birth_year    cluster  overlap_cluster
count      99       99.0    99.000000  99.000000              0.0
unique      1        NaN          NaN        NaN              NaN
top         F        NaN          NaN        NaN              NaN
freq       99        NaN          NaN        NaN              NaN
mean      NaN        8.0  1979.646465   4.464646              NaN
std       NaN        0.0     6.751135   3.134121              NaN
min       NaN        8.0  1970.000000   1.000000              NaN
25%       NaN        8.0  1974.000000   1.000000              NaN
50%       NaN        8.0  1979.000000   4.000000              NaN
75%       NaN        8.0  1985.000000   7.000000              NaN
max       NaN        8.0  1992.000000   9.000000              NaN
----------------------------------------------
OVERLAP_1_DF   SUB CATEGORY SUMMARY
----------------------------------------------
           prod_sub_cat  total_sales        mean  median
7         total_kitchen    60022.495  606.287828     0.0
2            total_bath    48565.855  490.564192     0.0
1      total_bags_women    47097.310  475.730404     0.0
4     total_clothing_men   41513.745  419.330758     0.0
8           total_tools    39408.720  398.067879     0.0
0        total_bags_men    39060.645  394.551970     0.0
3    total_clothing_kids   24894.545  251.460051     0.0
6      total_furnishing    24805.040  250.555960     0.0
5  total_clothing_women    20737.535  209.470051     0.0
----------------------------------------------

 ++++++++++++++++++++++++++++++++++++++++++++++++++++

----------------------------------------------
OVERLAP_2_DF   DESCRIPTION
----------------------------------------------
       gender  city_code   birth_year     cluster overlap_cluster
count     111      111.0   111.000000  111.000000               0
unique      1        NaN          NaN         NaN               0
top         F        NaN          NaN         NaN             NaN
freq      111        NaN          NaN         NaN             NaN
mean      NaN        4.0  1981.234234    3.639640             NaN
std       NaN        0.0     6.883386    2.177542             NaN
min       NaN        4.0  1970.000000    1.000000             NaN
25%       NaN        4.0  1975.000000    1.000000             NaN
50%       NaN        4.0  1982.000000    4.000000             NaN
75%       NaN        4.0  1987.000000    5.000000             NaN
max       NaN        4.0  1992.000000    8.000000             NaN
----------------------------------------------
OVERLAP_2_DF   SUB CATEGORY SUMMARY
----------------------------------------------
           prod_sub_cat  total_sales        mean  median
8           total_tools    58562.790  527.592703     0.0
1      total_bags_women    55235.635  497.618333     0.0
7         total_kitchen    48431.045  436.315721     0.0
4     total_clothing_men   47947.055  431.955450     0.0
3    total_clothing_kids   46934.875  422.836712     0.0
0        total_bags_men    40144.650  361.663514     0.0
2            total_bath    38802.075  349.568243     0.0
6      total_furnishing    32822.920  295.701982     0.0
5  total_clothing_women    26816.140  241.586847     0.0
----------------------------------------------

 ++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
------------------------------------------------
OVERLAP_3_DF   DESCRIPTION
------------------------------------------------
       gender  city_code   birth_year     cluster overlap_cluster
count     108      108.0   108.000000  108.000000               0
unique      1        NaN          NaN         NaN               0
top         F        NaN          NaN         NaN             NaN
freq      108        NaN          NaN         NaN             NaN
mean      NaN        3.0  1982.398148    4.972222             NaN
std       NaN        0.0     6.409601    3.268251             NaN
min       NaN        3.0  1970.000000    1.000000             NaN
25%       NaN        3.0  1977.000000    1.000000             NaN
50%       NaN        3.0  1983.000000    4.000000             NaN
75%       NaN        3.0  1987.250000    7.000000             NaN
max       NaN        3.0  1992.000000   10.000000             NaN
------------------------------------------------
OVERLAP_3_DF   SUB CATEGORY SUMMARY
------------------------------------------------
           prod_sub_cat  total_sales        mean  median
3   total_clothing_kids    49681.905  460.017639     0.0
8           total_tools    49070.840  454.359630     0.0
4     total_clothing_men    45551.415  421.772361     0.0
5  total_clothing_women    43651.920  404.184444     0.0
6       total_furnishing    40699.360  376.845926     0.0
1       total_bags_women    39414.245  364.946713     0.0
7          total_kitchen    36249.525  335.643750     0.0
0        total_bags_men    28062.580  259.838704     0.0
2            total_bath    27947.660  258.774630     0.0
------------------------------------------------

 ++++++++++++++++++++++++++++++++++++++++++++++++++++

------------------------------------------------
OVERLAP_4_DF   DESCRIPTION
------------------------------------------------
       gender  city_code   birth_year    cluster overlap_cluster
count      96       96.0    96.000000  96.000000               0
unique      1        NaN          NaN        NaN               0
top         F        NaN          NaN        NaN             NaN
freq       96        NaN          NaN        NaN             NaN
mean      NaN        2.0  1980.364583   3.593750             NaN
std       NaN        0.0     6.430388   2.231724             NaN
min       NaN        2.0  1970.000000   1.000000             NaN
25%       NaN        2.0  1975.000000   1.000000             NaN
50%       NaN        2.0  1980.500000   4.000000             NaN
75%       NaN        2.0  1985.000000   5.250000             NaN
max       NaN        2.0  1992.000000   8.000000             NaN
------------------------------------------------
OVERLAP_4_DF   SUB CATEGORY SUMMARY
------------------------------------------------
           prod_sub_cat  total_sales        mean  median
4     total_clothing_men    52456.560  546.422500     0.0
6       total_furnishing    49557.040  516.219167     0.0
3   total_clothing_kids    46596.745  485.382760     0.0
8           total_tools    35062.755  365.237031     0.0
5  total_clothing_women    33157.735  345.393073     0.0
7          total_kitchen    31432.830  327.425313     0.0
0        total_bags_men    26234.910  273.280312     0.0
1       total_bags_women    22294.480  232.234167     0.0
2            total_bath    18307.640  190.704583     0.0
------------------------------------------------

 ++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
------------------------------------------------
OVERLAP_5_DF   DESCRIPTION
------------------------------------------------
       gender  city_code   birth_year     cluster overlap_cluster
count     117      117.0   117.000000  117.000000               0
unique      1        NaN          NaN         NaN               0
top         M        NaN          NaN         NaN             NaN
freq      117        NaN          NaN         NaN             NaN
mean      NaN       10.0  1981.675214    7.042735             NaN
std       NaN        0.0     6.695121    2.859929             NaN
min       NaN       10.0  1970.000000    2.000000             NaN
25%       NaN       10.0  1977.000000    3.000000             NaN
50%       NaN       10.0  1982.000000    7.000000             NaN
75%       NaN       10.0  1987.000000   10.000000             NaN
max       NaN       10.0  1992.000000   10.000000             NaN
------------------------------------------------
OVERLAP_5_DF   SUB CATEGORY SUMMARY
------------------------------------------------
           prod_sub_cat  total_sales        mean  median
4    total_clothing_men    54103.010  462.418889     0.0
6      total_furnishing    52042.185  444.805000     0.0
5  total_clothing_women    47643.180  407.206667     0.0
8           total_tools    42599.960  364.102222     0.0
2            total_bath    38431.900  328.477778     0.0
3   total_clothing_kids    35320.220  301.882222     0.0
7         total_kitchen    33109.115  282.983889     0.0
0        total_bags_men    29493.555  252.081667     0.0
1      total_bags_women    28181.920  240.871111     0.0
------------------------------------------------

 ++++++++++++++++++++++++++++++++++++++++++++++++++++

------------------------------------------------
OVERLAP_6_DF   DESCRIPTION
------------------------------------------------
       gender  city_code   birth_year     cluster overlap_cluster
count     122      122.0   122.000000  122.000000               0
unique      1        NaN          NaN         NaN               0
top         M        NaN          NaN         NaN             NaN
freq      122        NaN          NaN         NaN             NaN
mean      NaN        1.0  1981.614754    6.696721             NaN
std       NaN        0.0     6.638788    3.203535             NaN
min       NaN        1.0  1970.000000    2.000000             NaN
25%       NaN        1.0  1976.000000    3.000000             NaN
50%       NaN        1.0  1982.000000    6.000000             NaN
75%       NaN        1.0  1987.000000   10.000000             NaN
max       NaN        1.0  1992.000000   10.000000             NaN
------------------------------------------------
OVERLAP_6_DF   SUB CATEGORY SUMMARY
------------------------------------------------
           prod_sub_cat  total_sales        mean  median
4    total_clothing_men    73110.115  599.263238     0.0
8           total_tools    72775.300  596.518852     0.0
3   total_clothing_kids    52711.815  432.064057     0.0
2            total_bath    43353.570  355.357131     0.0
6      total_furnishing    40952.405  335.675451     0.0
1      total_bags_women    39777.790  326.047459     0.0
7         total_kitchen    32955.520  270.127213     0.0
5  total_clothing_women    28548.780  234.006393     0.0
0        total_bags_men    28002.910  229.532049     0.0
------------------------------------------------

 ++++++++++++++++++++++++++++++++++++++++++++++++++++
```

OVERLAP_7_DF   DESCRIPTION

----

	gender	city_code	birth_year	cluster	overlap_cluster
count	113	113.0	113.000000	113.000000	0
unique	1	NaN	NaN	NaN	0
top	M	NaN	NaN	NaN	NaN
freq	113	NaN	NaN	NaN	NaN
mean	NaN	6.0	1981.044248	6.964602	NaN
std	NaN	0.0	6.751176	3.102215	NaN
min	NaN	6.0	1970.000000	2.000000	NaN
25%	NaN	6.0	1975.000000	3.000000	NaN
50%	NaN	6.0	1979.000000	9.000000	NaN
75%	NaN	6.0	1988.000000	10.000000	NaN
max	NaN	6.0	1992.000000	10.000000	NaN

OVERLAP_7_DF   SUB CATEGORY SUMMARY

----

	prod_sub_cat	total_sales	mean	median
8	total_tools	51620.075	456.814823	0.0
5	total_clothing_women	48032.140	425.063186	0.0
4	total_clothing_men	45671.860	404.175752	0.0
2	total_bath	42521.505	376.296504	0.0
0	total_bags_men	42423.160	375.426195	0.0
6	total_furnishing	33401.940	295.592389	0.0
3	total_clothing_kids	29737.760	263.166018	0.0
1	total_bags_women	29078.075	257.328097	0.0
7	total_kitchen	19166.225	169.612611	0.0

++

OVERLAP_8_DF   DESCRIPTION

----

	gender	city_code	birth_year	cluster	overlap_cluster
count	103	103.0	103.000000	103.000000	0
unique	1	NaN	NaN	NaN	0
top	M	NaN	NaN	NaN	NaN
freq	103	NaN	NaN	NaN	NaN
mean	NaN	9.0	1981.349515	6.815534	NaN
std	NaN	0.0	6.007689	2.933069	NaN
min	NaN	9.0	1970.000000	2.000000	NaN
25%	NaN	9.0	1975.500000	3.000000	NaN
50%	NaN	9.0	1981.000000	8.000000	NaN
75%	NaN	9.0	1986.500000	10.000000	NaN
max	NaN	9.0	1992.000000	10.000000	NaN

OVERLAP_8_DF   SUB CATEGORY SUMMARY

----

	prod_sub_cat	total_sales	mean	median
0	total_bags_men	61775.025	599.757524	0.0
4	total_clothing_men	53518.465	519.596748	0.0
6	total_furnishing	53496.365	519.382184	0.0
7	total_kitchen	51870.910	503.601068	0.0
2	total_bath	44930.405	436.217524	0.0
5	total_clothing_women	35619.675	345.822087	0.0
1	total_bags_women	32028.425	310.955583	0.0
8	total_tools	29121.170	282.729806	0.0
3	total_clothing_kids	18735.275	181.895874	0.0

++

```
-------------------------------------------------
OVERLAP_9_DF   DESCRIPTION
-------------------------------------------------
       gender  city_code   birth_year     cluster overlap_cluster
count     114      114.0   114.000000  114.000000               0
unique      1        NaN          NaN         NaN               0
top         F        NaN          NaN         NaN             NaN
freq      114        NaN          NaN         NaN             NaN
mean      NaN        6.0  1981.982456    4.096491             NaN
std       NaN        0.0     6.471151    2.120149             NaN
min       NaN        6.0  1970.000000    1.000000             NaN
25%       NaN        6.0  1977.250000    4.000000             NaN
50%       NaN        6.0  1982.000000    4.000000             NaN
75%       NaN        6.0  1988.000000    5.000000             NaN
max       NaN        6.0  1992.000000    8.000000             NaN
-------------------------------------------------
OVERLAP_9_DF   SUB CATEGORY SUMMARY
-------------------------------------------------
           prod_sub_cat  total_sales        mean  median
2            total_bath    81806.465  717.600570     0.0
0        total_bags_men    58977.165  517.343553     0.0
6       total_furnishing   55698.630  488.584474     0.0
8           total_tools    52791.375  463.082237     0.0
3   total_clothing_kids    39544.635  346.882763     0.0
4    total_clothing_men    35868.300  314.634211     0.0
7         total_kitchen    35217.455  308.925044     0.0
5  total_clothing_women    34674.900  304.165789     0.0
1      total_bags_women    24342.045  213.526711     0.0
-------------------------------------------------

 ++++++++++++++++++++++++++++++++++++++++++++++++++++

-------------------------------------------------
OVERLAP_10_DF   DESCRIPTION
-------------------------------------------------
       gender  city_code   birth_year     cluster overlap_cluster
count     111      111.0   111.000000  111.000000               0
unique      1        NaN          NaN         NaN               0
top         M        NaN          NaN         NaN             NaN
freq      111        NaN          NaN         NaN             NaN
mean      NaN        7.0  1981.558559    5.612613             NaN
std       NaN        0.0     6.590192    2.744155             NaN
min       NaN        7.0  1970.000000    2.000000             NaN
25%       NaN        7.0  1976.000000    3.000000             NaN
50%       NaN        7.0  1981.000000    4.000000             NaN
75%       NaN        7.0  1987.500000    9.000000             NaN
max       NaN        7.0  1992.000000   10.000000             NaN
-------------------------------------------------
OVERLAP_10_DF   SUB CATEGORY SUMMARY
-------------------------------------------------
           prod_sub_cat  total_sales        mean  median
4    total_clothing_men    68426.020  616.450631     0.0
5  total_clothing_women    59992.660  540.474414     0.0
8           total_tools    44121.545  397.491396     0.0
1      total_bags_women    34377.655  309.708604     0.0
7         total_kitchen    34136.765  307.538423     0.0
3   total_clothing_kids    29794.115  268.415450     0.0
0        total_bags_men    29620.630  266.852523     0.0
6      total_furnishing    27531.075  248.027703     0.0
2            total_bath    13950.625  125.681306     0.0
-------------------------------------------------

 ++++++++++++++++++++++++++++++++++++++++++++++++++++
```

This table summarizes the printed results. The **Cluster Details** describe the profiles of the overlapping customers, while the **Top Selling Subcategories** are the top 3 selling subcategories.

Cluster Details							Top Selling Subcategories		
Overlapping Cluster	Count	Mean Birth Year	Mean Age (2013 - birth year)	Median Birth Year	Gender	City Code	1st Subcategory	2nd Subcategory	3rd Subcategory
1	99	1980	33	1979	F	8	kitchen	bath	bags_women
2	111	1981	32	1982	F	4	tools	bags_women	kitchen
3	108	1982	31	1983	F	3	clothing_kids	tools	clothing_men
4	96	1980	33	1981	F	2	clothing_men	furnishing	clothing_kids
5	117	1982	31	1982	M	10	clothing_men	furnishing	clothing_women
6	122	1982	31	1982	M	1	clothing_men	tools	clothing_kids
7	113	1981	32	1979	M	6	tools	clothing_women	clothing_men
8	103	1981	32	1981	M	9	bags_men	clothing_women	furnishing
9	114	1982	31	1982	F	6	bath	bags_men	furnishing
10	111	1982	31	1981	M	7	clothing_men	clothing_women	tools

5.2. Solid Clusters

Solid clusters are the clusters of the same color that have stuck together. The cell does the same as the one for overlapping clusters. However, this time, customers belonging to solid clusters are extracted by specifying their `cluster` number and if their `overlap_cluster` is `Null` .

In [54]:
```
for i in range(1, 11):
    solid_cluster = profile_cluster_df[(profile_cluster_df['cluster'] == i) &
                                       (profile_cluster_df['overlap_cluster'].isnull())
                                      ]

    describe_cluster_df = solid_cluster[describe_columns].describe(include='all')

    totals_list = solid_cluster[sub_cat_columns].sum()
    mean_list = solid_cluster[sub_cat_columns].mean()
    median_list = solid_cluster[sub_cat_columns].median()

    categ_summary_data = zip(sub_cat_columns, totals_list, mean_list, median_list)
    solid_categ_summary_df = pd.DataFrame(categ_summary_data, columns=['prod_sub_cat', 'to
tal_sales', 'mean', 'median'])
    solid_categ_summary_df.sort_values(by=['total_sales'], ascending=False, inplace=True)

    cluster_name = 'SOLID CLUSTER # {}'.format(str(i))

    print_cluster_summary(cluster_name, describe_cluster_df, solid_categ_summary_df)
```

```
----------------------------------------------
SOLID CLUSTER # 1   DESCRIPTION
----------------------------------------------
       gender  city_code   birth_year  cluster overlap_cluster
count     127      127.0   127.000000    127.0               0
unique      1        NaN          NaN      NaN               0
top         F        NaN          NaN      NaN             NaN
freq      127        NaN          NaN      NaN             NaN
mean      NaN        5.0  1981.330709      1.0             NaN
std       NaN        0.0     6.795254      0.0             NaN
min       NaN        5.0  1970.000000      1.0             NaN
25%       NaN        5.0  1975.000000      1.0             NaN
50%       NaN        5.0  1982.000000      1.0             NaN
75%       NaN        5.0  1987.000000      1.0             NaN
max       NaN        5.0  1992.000000      1.0             NaN
----------------------------------------------
SOLID CLUSTER # 1   SUB CATEGORY SUMMARY
----------------------------------------------
           prod_sub_cat  total_sales        mean  median
8           total_tools    82650.685  650.792795     0.0
5  total_clothing_women    61191.585  481.823504     0.0
4    total_clothing_men    56680.975  446.306890     0.0
7         total_kitchen    53569.295  421.805472     0.0
0        total_bags_men    48625.525  382.878150     0.0
6      total_furnishing    47019.960  370.235906     0.0
1      total_bags_women    45361.355  357.176024     0.0
3   total_clothing_kids    41570.100  327.323622     0.0
2            total_bath    38856.220  305.954488     0.0
----------------------------------------------

 ++++++++++++++++++++++++++++++++++++++++++++++++++++

----------------------------------------------
SOLID CLUSTER # 2   DESCRIPTION
----------------------------------------------
       gender  city_code   birth_year  cluster overlap_cluster
count     130      130.0   130.000000    130.0               0
unique      1        NaN          NaN      NaN               0
top         M        NaN          NaN      NaN             NaN
freq      130        NaN          NaN      NaN             NaN
mean      NaN        4.0  1979.861538      2.0             NaN
std       NaN        0.0     6.705030      0.0             NaN
min       NaN        4.0  1970.000000      2.0             NaN
25%       NaN        4.0  1974.000000      2.0             NaN
50%       NaN        4.0  1980.000000      2.0             NaN
75%       NaN        4.0  1985.750000      2.0             NaN
max       NaN        4.0  1992.000000      2.0             NaN
----------------------------------------------
SOLID CLUSTER # 2   SUB CATEGORY SUMMARY
----------------------------------------------
           prod_sub_cat  total_sales      mean  median
6      total_furnishing    61996.025  476.8925     0.0
5  total_clothing_women    56299.750  433.0750     0.0
7         total_kitchen    55962.725  430.4825     0.0
4    total_clothing_men    46989.020  361.4540     0.0
2            total_bath    41309.320  317.7640     0.0
1      total_bags_women    38760.085  298.1545     0.0
8           total_tools    35383.205  272.1785     0.0
0        total_bags_men    33617.415  258.5955     0.0
3   total_clothing_kids    32601.920  250.7840     0.0
----------------------------------------------

 ++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
-------------------------------------------------
SOLID CLUSTER # 3   DESCRIPTION
-------------------------------------------------
       gender  city_code   birth_year  cluster overlap_cluster
count     142      142.0   142.000000    142.0               0
unique      1        NaN          NaN      NaN               0
top         M        NaN          NaN      NaN             NaN
freq      142        NaN          NaN      NaN             NaN
mean      NaN        5.0  1979.901408      3.0             NaN
std       NaN        0.0     6.914175      0.0             NaN
min       NaN        5.0  1970.000000      3.0             NaN
25%       NaN        5.0  1974.000000      3.0             NaN
50%       NaN        5.0  1979.000000      3.0             NaN
75%       NaN        5.0  1986.000000      3.0             NaN
max       NaN        5.0  1992.000000      3.0             NaN
-------------------------------------------------
SOLID CLUSTER # 3   SUB CATEGORY SUMMARY
-------------------------------------------------
           prod_sub_cat  total_sales        mean  median
2            total_bath    77826.255  548.072218     0.0
3   total_clothing_kids    67645.890  476.379507     0.0
1      total_bags_women    65477.880  461.111831     0.0
8           total_tools    57259.995  403.239401     0.0
6      total_furnishing    45359.145  319.430599     0.0
4    total_clothing_men    41951.325  295.431866     0.0
7         total_kitchen    41114.840  289.541127     0.0
0        total_bags_men    36052.835  253.893204     0.0
5  total_clothing_women    34697.000  244.345070     0.0
-------------------------------------------------

 ++++++++++++++++++++++++++++++++++++++++++++++++++++

-------------------------------------------------
SOLID CLUSTER # 4   DESCRIPTION
-------------------------------------------------
       gender  city_code   birth_year  cluster overlap_cluster
count     126      126.0   126.000000    126.0               0
unique      1        NaN          NaN      NaN               0
top         F        NaN          NaN      NaN             NaN
freq      126        NaN          NaN      NaN             NaN
mean      NaN        7.0  1981.238095      4.0             NaN
std       NaN        0.0     6.663847      0.0             NaN
min       NaN        7.0  1970.000000      4.0             NaN
25%       NaN        7.0  1975.250000      4.0             NaN
50%       NaN        7.0  1981.000000      4.0             NaN
75%       NaN        7.0  1987.000000      4.0             NaN
max       NaN        7.0  1992.000000      4.0             NaN
-------------------------------------------------
SOLID CLUSTER # 4   SUB CATEGORY SUMMARY
-------------------------------------------------
           prod_sub_cat  total_sales        mean  median
3   total_clothing_kids    60979.425  483.963690     0.0
6      total_furnishing    59727.460  474.027460     0.0
2            total_bath    59471.100  471.992857     0.0
1      total_bags_women    57372.705  455.338929     0.0
5  total_clothing_women    46007.780  365.141111     0.0
7         total_kitchen    45454.175  360.747421     0.0
4    total_clothing_men    41996.630  333.306587     0.0
8           total_tools    40809.860  323.887778     0.0
0        total_bags_men    38539.085  305.865754     0.0
-------------------------------------------------

 ++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
---------------------------------------------------
SOLID CLUSTER # 5   DESCRIPTION
---------------------------------------------------
       gender  city_code  birth_year  cluster overlap_cluster
count     124      124.0   124.00000    124.0               0
unique      1        NaN         NaN      NaN               0
top         F        NaN         NaN      NaN             NaN
freq      124        NaN         NaN      NaN             NaN
mean      NaN        1.0  1981.50000      5.0             NaN
std       NaN        0.0     6.37513      0.0             NaN
min       NaN        1.0  1970.00000      5.0             NaN
25%       NaN        1.0  1977.00000      5.0             NaN
50%       NaN        1.0  1981.00000      5.0             NaN
75%       NaN        1.0  1987.00000      5.0             NaN
max       NaN        1.0  1992.00000      5.0             NaN
---------------------------------------------------
SOLID CLUSTER # 5   SUB CATEGORY SUMMARY
---------------------------------------------------
           prod_sub_cat  total_sales        mean  median
8           total_tools    57189.275  461.203831     0.0
3   total_clothing_kids    56773.795  457.853185     0.0
6       total_furnishing   51526.150  415.533468     0.0
0        total_bags_men    49902.905  402.442782     0.0
7         total_kitchen    47254.220  381.082419     0.0
1       total_bags_women   41985.580  338.593387     0.0
4     total_clothing_men   37646.245  303.598750     0.0
2             total_bath   34229.585  276.045040     0.0
5   total_clothing_women   31724.550  255.843145     0.0
---------------------------------------------------

 ++++++++++++++++++++++++++++++++++++++++++++++++++++

---------------------------------------------------
SOLID CLUSTER # 6   DESCRIPTION
---------------------------------------------------
       gender  city_code   birth_year  cluster overlap_cluster
count     120      120.0   120.000000    120.0               0
unique      1        NaN          NaN      NaN               0
top         M        NaN          NaN      NaN             NaN
freq      120        NaN          NaN      NaN             NaN
mean      NaN        2.0  1981.150000      6.0             NaN
std       NaN        0.0     6.654309      0.0             NaN
min       NaN        2.0  1970.000000      6.0             NaN
25%       NaN        2.0  1976.000000      6.0             NaN
50%       NaN        2.0  1980.500000      6.0             NaN
75%       NaN        2.0  1986.250000      6.0             NaN
max       NaN        2.0  1992.000000      6.0             NaN
---------------------------------------------------
SOLID CLUSTER # 6   SUB CATEGORY SUMMARY
---------------------------------------------------
           prod_sub_cat  total_sales        mean  median
7         total_kitchen    62990.525  524.921042     0.0
1       total_bags_women   62882.235  524.018625     0.0
3   total_clothing_kids    62205.975  518.383125     0.0
0        total_bags_men    60240.180  502.001500     0.0
4     total_clothing_men   41733.640  347.780333     0.0
6       total_furnishing   41245.230  343.710250     0.0
2             total_bath   38012.000  316.766667     0.0
5   total_clothing_women   27601.795  230.014958     0.0
8           total_tools    24850.345  207.086208     0.0
---------------------------------------------------

 ++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
-----------------------------------------------
SOLID CLUSTER # 7   DESCRIPTION
-----------------------------------------------
       gender  city_code   birth_year  cluster overlap_cluster
count     124      124.0   124.000000    124.0               0
unique      1        NaN          NaN      NaN               0
top         F        NaN          NaN      NaN             NaN
freq      124        NaN          NaN      NaN             NaN
mean      NaN       10.0  1980.975806      7.0             NaN
std       NaN        0.0     6.399268      0.0             NaN
min       NaN       10.0  1970.000000      7.0             NaN
25%       NaN       10.0  1976.000000      7.0             NaN
50%       NaN       10.0  1981.000000      7.0             NaN
75%       NaN       10.0  1986.250000      7.0             NaN
max       NaN       10.0  1992.000000      7.0             NaN
-----------------------------------------------
SOLID CLUSTER # 7   SUB CATEGORY SUMMARY
-----------------------------------------------
           prod_sub_cat  total_sales        mean  median
3   total_clothing_kids    56423.510  455.028306     0.0
1      total_bags_women    55246.685  445.537782     0.0
5  total_clothing_women    52422.305  422.760524     0.0
8           total_tools    50475.295  407.058831     0.0
2            total_bath    40219.790  324.353145     0.0
4    total_clothing_men    40045.200  322.945161     0.0
6      total_furnishing    36894.845  297.539073     0.0
0        total_bags_men    31714.605  255.762944     0.0
7         total_kitchen    30301.310  244.365403     0.0
-----------------------------------------------

 ++++++++++++++++++++++++++++++++++++++++++++++++++++

-----------------------------------------------
SOLID CLUSTER # 8   DESCRIPTION
-----------------------------------------------
       gender  city_code   birth_year  cluster overlap_cluster
count     125      125.0   125.000000    125.0               0
unique      1        NaN          NaN      NaN               0
top         F        NaN          NaN      NaN             NaN
freq      125        NaN          NaN      NaN             NaN
mean      NaN        9.0  1980.688000      8.0             NaN
std       NaN        0.0     6.197575      0.0             NaN
min       NaN        9.0  1970.000000      8.0             NaN
25%       NaN        9.0  1975.000000      8.0             NaN
50%       NaN        9.0  1981.000000      8.0             NaN
75%       NaN        9.0  1985.000000      8.0             NaN
max       NaN        9.0  1992.000000      8.0             NaN
-----------------------------------------------
SOLID CLUSTER # 8   SUB CATEGORY SUMMARY
-----------------------------------------------
           prod_sub_cat  total_sales       mean  median
0        total_bags_men    66059.110  528.47288     0.0
2            total_bath    64217.075  513.73660     0.0
4    total_clothing_men    52152.685  417.22148     0.0
8           total_tools    48510.605  388.08484     0.0
3   total_clothing_kids    45526.000  364.20800     0.0
6      total_furnishing    44858.580  358.86864     0.0
1      total_bags_women    43249.700  345.99760     0.0
7         total_kitchen    30682.535  245.46028     0.0
5  total_clothing_women    29833.895  238.67116     0.0
-----------------------------------------------

 ++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
---------------------------------------------
SOLID CLUSTER # 9   DESCRIPTION
---------------------------------------------
       gender  city_code   birth_year  cluster overlap_cluster
count     135      135.0   135.000000    135.0               0
unique      1        NaN          NaN      NaN               0
top         M        NaN          NaN      NaN             NaN
freq      135        NaN          NaN      NaN             NaN
mean      NaN        8.0  1981.274074      9.0             NaN
std       NaN        0.0     6.825577      0.0             NaN
min       NaN        8.0  1970.000000      9.0             NaN
25%       NaN        8.0  1975.000000      9.0             NaN
50%       NaN        8.0  1982.000000      9.0             NaN
75%       NaN        8.0  1987.500000      9.0             NaN
max       NaN        8.0  1992.000000      9.0             NaN
---------------------------------------------
SOLID CLUSTER # 9   SUB CATEGORY SUMMARY
---------------------------------------------
           prod_sub_cat  total_sales        mean  median
7         total_kitchen    69339.855  513.628556     0.0
4    total_clothing_men    64604.930  478.555037     0.0
5  total_clothing_women    55851.120  413.712000     0.0
1      total_bags_women    55850.015  413.703815     0.0
8           total_tools    49098.465  363.692333     0.0
3   total_clothing_kids    45522.685  337.205074     0.0
0        total_bags_men    41682.810  308.761556     0.0
2            total_bath    37011.975  274.162778     0.0
6      total_furnishing    31074.810  230.183778     0.0
---------------------------------------------

 +++++++++++++++++++++++++++++++++++++++++++++++++++

---------------------------------------------
SOLID CLUSTER # 10   DESCRIPTION
---------------------------------------------
       gender  city_code   birth_year  cluster overlap_cluster
count     129      129.0   129.000000    129.0               0
unique      1        NaN          NaN      NaN               0
top         M        NaN          NaN      NaN             NaN
freq      129        NaN          NaN      NaN             NaN
mean      NaN        3.0  1981.992248     10.0             NaN
std       NaN        0.0     6.320226      0.0             NaN
min       NaN        3.0  1970.000000     10.0             NaN
25%       NaN        3.0  1977.000000     10.0             NaN
50%       NaN        3.0  1982.000000     10.0             NaN
75%       NaN        3.0  1987.000000     10.0             NaN
max       NaN        3.0  1992.000000     10.0             NaN
---------------------------------------------
SOLID CLUSTER # 10   SUB CATEGORY SUMMARY
---------------------------------------------
           prod_sub_cat  total_sales        mean  median
0        total_bags_men    86003.255  666.691899     0.0
8           total_tools    83343.520  646.073798     0.0
7         total_kitchen    55716.310  431.909380     0.0
5  total_clothing_women    55644.485  431.352597     0.0
6      total_furnishing    45019.910  348.991550     0.0
1      total_bags_women    41015.390  317.948760     0.0
2            total_bath    36529.090  283.171240     0.0
3   total_clothing_kids    35225.190  273.063488     0.0
4    total_clothing_men    23665.785  183.455698     0.0
---------------------------------------------

 +++++++++++++++++++++++++++++++++++++++++++++++++++
```

5.2.1. Solid Clusters Summary

This table summarizes the customer details and top selling subcategories of the solid clusters.

Cluster Details							Top Selling Subcategories		
Solid Cluster	Count	Mean Birth Year	Mean Age (2013 - birth year)	Median Birth Year	Gender	City Code	1st Subcategory	2nd Subcategory	3rd Subcategory
1	127	1981	33	1982	F	5	tools	clothing_women	clothing_men
2	130	1980	32	1980	M	4	furnishing	clothing_women	kitchen
3	142	1980	31	1979	M	5	bath	clothing_kids	bags_women
4	126	1981	33	1981	F	7	clothing_kids	furnishing	bath
5	124	1982	31	1981	F	1	tools	clothing_kids	furnishing
6	120	1981	31	1981	M	2	kitchen	bags_women	clothing_kids
7	124	1981	32	1981	F	10	clothing_kids	bags_women	clothing_women
8	125	1981	32	1981	F	9	bags_men	bath	clothing_men
9	135	1981	31	1982	M	8	kitchen	clothing_men	clothing_women
10	129	1982	31	1982	M	3	bags_men	tools	kitchen

6. Recommendations

The clustering analysis of this study was initialized with 10 clusters, but based on the scatterplot, there are 10 additional overlapping clusters. **20 market segments have then been identified**.

The customers have been grouped into 20 market segments. It would be ideal to learn more about their behavior and their buying patterns. After studying them, create promotions and bundles to drive sales for each target market. Since these customers bought from the e-shop, put out recommendations based on their clusters.

All customers in the 20 clusters are in their early thirties (30-33) but each cluster has different buying priorities. Here are some clusters that I was able to describe.

Cluster Details							Top Selling Subcategories				
Overlapping Cluster	Count	Mean Birth Year	Mean Age (2013 - birth year)	Median Birth Year	Gender	City Code	1st Subcategory	2nd Subcategory	3rd Subcategory	Description	Suggestion
1	99	1980	33	1979	F	8	kitchen	bath	bags_women	Women who are furnishing their kitchen and bath. They might occasionally buys some bags.	Bundle kitchen and bath items. Recommend items that can be used for both rooms (shelves, towels, mats).
4	96	1980	33	1981	F	2	clothing_men	furnishing	clothing_kids	Women who buy items for their family members.	Bundle matching father and son clothes. Recommend some home improvement furniture.

			Cluster Details					Top Selling Subcategories			
Solid Cluster	**Count**	**Mean Birth Year**	**Mean Age (2013 - birth year)**	**Median Birth Year**	**Gender**	**City Code**	**1st Subcategory**	**2nd Subcategory**	**3rd Subcategory**	**Description**	**Suggestion**
4	126	1981	33	1981	F	7	clothing_kids	furnishing	bath	Mothers who have kids at home and are furnishing their house.	Recommend kid's clothes and kid-friendly furniture.
7	124	1981	32	1981	F	10	clothing_kids	bags_women	clothing_women	Mothers who buy clothes and accessories for them and their child.	Bundle mother and child clothes and accessories. Recommend matching items so both persons would have the same style.
8	125	1981	32	1981	F	9	bags_men	bath	clothing_men	Women who are buying items that will improve the style and grooming of men.	Recommend gift-sets for men.

7. Further Improvements

This study is only focused on e-shop customers who have bought items from the clothes, home and kitchen and bags categories. If machine specifications can carry the workload, it would be good to cluster the whole dataset. This way, we can understand the the behavior of customers from different sales channels.

To effectively understand customer behavior, the dataset can have the following data:

1. Marital Status - To know if a customer is buying it for himself or for someone.
2. Gender Identity - To understand why some customers are buying items of the opposite sex. Are they going to use it for themselves or give it to someone?
3. Purpose - What exactly are they going to use the items for? Were the items boughts as gifts, personal collections, hobbies or on a whim?
4. The items themselves - It is not enough that the dataset has subcategories. The products must be identified to understand why they are bought.

```
In [ ]:
```