Instacart EDA 2 Assignment [ANSWER]

Business Insights

- Which aisle has the most products?
- What is the average position of a product in an order?

Introduction

For your second assignment you have to execute code and fill out the missing code.

- the code blocks that you need to execute (without writing something else) are marked with single hashtag (#)
- the code blocks that you need to fill out the missing part are marked with a double hashtag (##)
- the code blocks that you need to write on your own are marked with triple hashtags (###)

For this assignment you will answer two business insights; one with data from products.csv, and one with data from order_products__prior.csv.

Before we start, Import the required packages for this assignment.

In [1]:
```
import pandas as pd                     # for data manipulation
import matplotlib.pyplot as plt         # for plotting
import seaborn as sns                   # an extension of matplotlib for statistical gr
aphics
```

Assignment I:

Which aisle has the most products?

To answer this question you have to:

1. Import the products.csv from directory *../input/products.csv*
2. .groupby() (all available products (from products data frame) by their 'aisle_id', select the appropriate column and use aggregation function count())
3. Rename the column of the produced data frame as: 'total_products'
4. Sort the values so to get the aisles with most products first.
5. Select the first 10 rows of the data frame.
6. Visualize the results.

In [2]:
```
## step 0 - import products.csv from directory '../input/products.csv'
products = pd.read_csv('../input/products.csv')
```

In [3]:
```
## step 1 - .groupby() ( all available products (from products data frame) by their
 'aisle_id', then select to find the size of each group
aisle_top = products.groupby('aisle_id')[['product_id']].count()
```

In [4]:
```
### step 2 - Rename the column of aisle_top as: 'total_products'
aisle_top.columns = ['total_products']
```

In [5]:
```
# Before you move on to step 3, have a look at your produced results so far.
# Check the results below
aisle_top.head()
```

Out[5]:

aisle_id	total_products
1	146
2	271
3	832
4	543
5	409

In [6]:
```
## step 3 - Sort the values of total_products so to get the aisles with most produ
cts first.
aisle_top_sort = aisle_top.sort_values(by='total_products', ascending=False)

## step 4 - Select the first 10 rows of the data frame. Remember that index in Pyt
hon starts from 0
aisle_top_sort = aisle_top_sort.iloc[0:10]
```

In [7]:
```
### Before you move on to the final step, how can you ensure that the aisle_top ha
s only 10 aisles?
aisle_top_sort.shape
```

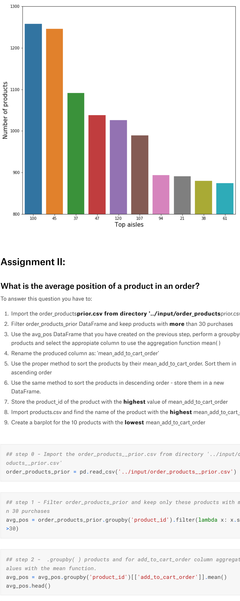
Out[7]:
```
(10, 1)
```

In [8]:
```
# Have a look at the produced data frame before you plot it (visualize it).
# Are your results fine?
aisle_top_sort.head()
```

Out[8]:

aisle_id	total_products
100	1258
45	1246
37	1091
47	1038
120	1026

In [9]:
```
## step 5 - Visualize the results. Place index on x-axis
plt.figure(figsize=(10,10))
sns.barplot(aisle_top_sort.index, aisle_top_sort.total_products  , order=aisle_t
op_sort.index)
plt.xlabel('Top aisles', size=15)
plt.ylabel('Number of products', size=15)
plt.ylim(800,1300)
plt.show()
```



Assignment II:

What is the average position of a product in an order?

To answer this question you have to:

1. Import the order_products**prior.csv from directory '../input/order_products**priorcsv'
2. Filter order_products_prior DataFrame and keep products with **more** than 30 purchases
3. Use the avg_pos DataFrame that you have created on the previous step, perform a groupby() on products and select the appropiate column to use the aggregation function mean ()
4. Rename the produced column as: 'mean_add_to_cart_order'
5. Use the proper method to sort the products by their mean_add_to_cart_order. Sort them in ascending order
6. Use the same method to sort the products in descending order - store them in a new DataFrame.
7. Store the product_id of the product with the **highest** value of mean_add_to_cart_order
8. Import products.csv and find the name of the product with the **highest** mean_add_to_cart_order
9. Create a barplot for the 10 products with the **lowest** mean_add_to_cart_order

In [10]:
```
## step 0 - Import the order_products__prior.csv from directory '../input/order_pr
oducts__prior.csv'
order_products_prior = pd.read_csv('../input/order_products__prior.csv')
```

In [11]:
```
## step 1 - Filter order_products_prior and keep only these products with more tha
n 30 purchases.
avg_pos = order_products_prior.groupby('product_id').filter(lambda x: x.shape[0]
>30)
```

In [12]:
```
## step 2 -  .groupby( ) products and for add_to_cart_order column aggregate the v
alues with the mean function.
avg_pos = avg_pos.groupby('product_id')[['add_to_cart_order']].mean()
avg_pos.head()
```

Out[12]:

product_id	add_to_cart_order
1	5.801836
2	9.888889
3	6.415162
4	9.507599
8	8.418182

In [13]:
```
### step 3 - Rename column of avg_pos as: 'mean_add_to_cart_order'
avg_pos.columns = ['mean_add_to_cart_order']
```

In [14]:
```
## step 4 -  Use the proper method to sort the products by their mean_add_to_cart_
order. Sort them in ascending order
avg_pos_asc = avg_pos.sort_values(by='mean_add_to_cart_order', ascending=True)
avg_pos_asc.head()
```

Out[14]:

product_id	mean_add_to_cart_order
14609	1.514286
25524	1.627907
4212	2.000000
15511	2.083333
45328	2.197183

In [15]:
```
## step 5 - And now use again the same method to sort the products in descending o
rder (store the results in a new DataFrame)
avg_pos_des = avg_pos.sort_values(by='mean_add_to_cart_order', ascending=False)
avg_pos_des.head()
```

Out[15]:

product_id	mean_add_to_cart_order
7816	22.674419
2959	22.285714
20103	19.423529
29228	18.402597
4431	18.000000

In [16]:
```
## step 6 - Store the product_id of the product with the highest mean_add_to_cart_
order
id_low = avg_pos_des.index[0]
id_low
```

Out[16]:
```
7816
```

In [17]:
```
### step 7 -  Import products.csv and find the name of the product with the highest
 mean_add_to_cart_order
products = pd.read_csv('../input/products.csv')
products[products.product_id== id_low ]
```

Out[17]:

	product_id	product_name	aisle_id	department_id
7815	7816	Madagascar Chocolate Bar	45	19

In [18]:
```
### step 8 - Create a sns.barplot for the 10 products with the lowest mean_add_to_
cart_order
avg_pos_asc_10 = avg_pos_asc.iloc[0:10]
sns.barplot(avg_pos_asc_10.index, avg_pos_asc_10.mean_add_to_cart_order, order=a
vg_pos_asc_10.index)
```

Out[18]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f42253a8208>
```