Instacart EDA 1 Assignment [ANSWER]

In this assignment you have to replace the missing code on the following code blocks.
First you will explore the available data on products.csv, and then you will proceed with a visualization on data of orders.csv.
Finally you will answer to the following questions:

Business Insights

- How many days pass since a prior order?
- How many orders do customers make? (frequency distribution)

Import Packages

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 graphics
```

Explore products data frame

- Now, Import products.csv and save it as a data frame. Use the appropiate function from pandas package.

In [2]:
```
products = pd.read_csv('../input/products.csv')
```

- And explore the data of products DataFrame. With which method can you retrieve the first rows of a dataframe?

In [3]:
```
products.head()
```

Out[3]:

	product_id	product_name	aisle_id	department_id
0	1	Chocolate Sandwich Cookies	61	19
1	2	All-Seasons Salt	104	13
2	3	Robust Golden Unsweetened Oolong Tea	94	7
3	4	Smart Ones Classic Favorites Mini Rigatoni Wit...	38	1
4	5	Green Chile Anytime Sauce	5	13

- As every row describes a single product, how many products does Instacart sell? You need to use the keyword that returns the number of rows and columns

In [4]:
```
products.shape
```

Out[4]:
```
(49688, 4)
```

- What is the actual size of the DataFrame (in MB)? You can answer on this question, by using the appropiate method that gives summary info for a data frame

In [5]:
```
products.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49688 entries, 0 to 49687
Data columns (total 4 columns):
product_id       49688 non-null int64
product_name     49688 non-null object
aisle_id         49688 non-null int64
department_id    49688 non-null int64
dtypes: int64(3), object(1)
memory usage: 1.5+ MB
```

How many days pass since a prior order?

Explore further orders data frame

- Now Import the orders.csv and save it as a dataframe

In [6]:
```
orders = pd.read_csv('../input/orders.csv' )
```

- Get the first rows of DataFrame orders. Have a look at **days_since_prior_order** column; what does it represent?

In [7]:
```
orders.head()
```

Out[7]:

	order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
0	2539329	1	prior	1	2	8	NaN
1	2398795	1	prior	2	3	7	15.0
2	473747	1	prior	3	3	12	21.0
3	2254736	1	prior	4	4	7	29.0
4	431534	1	prior	5	4	15	28.0

The **days_since_prior_order** column indicates how many days have passed since a previous order of a customer. The first order of each customer has NaN (Not a Number) value as there is no previous order.

- Use the appropiate method to find on **days_since_prior_order** how many orders were placed for each distinct period of days. Actually, you want to count how many orders were placed after 1 day, 2 days etc.

In [8]:
```
orders.days_since_prior_order.value_counts()
```

Out[8]:
```
30.0    369323
7.0     328668
6.0     240813
4.0     221696
5.0     217985
8.0     214583
3.0     193296
9.0     181717
1.0     145247
2.0     139188
14.0    100239
10.0     93186
13.0     83214
11.0     80978
12.0     76146
0.0      67755
15.0     66579
16.0     46941
21.0     45478
17.0     39245
20.0     38527
18.0     35881
19.0     34384
22.0     32812
28.0     26777
23.0     23885
27.0     22913
24.0     20712
25.0     19234
29.0     19101
26.0     19816
Name: days_since_prior_order, dtype: int64
```

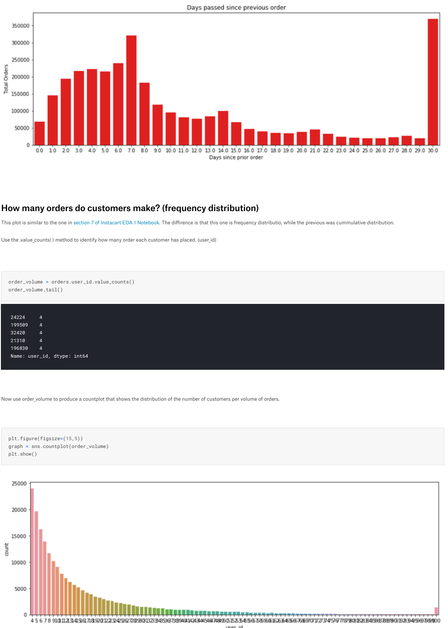
- Use the 'days_since_prior_order' column from orders data frame and create a countplot. Use appropriate titles, labels and a proper size.

In [9]:
```
plt.figure(figsize=(15,5))

sns.countplot(x='days_since_prior_order', data=orders, color='red')

plt.ylabel('Total Orders')
plt.xlabel('Days since prior order')
plt.title('Days passed since previous order')

plt.show()
```



How many orders do customers make? (frequency distribution)

This plot is similar to the one in section 7 of Instacart EDA 1 Notebook. The difference is that this one is frequency distributio, while the previous was cummulative distribution.

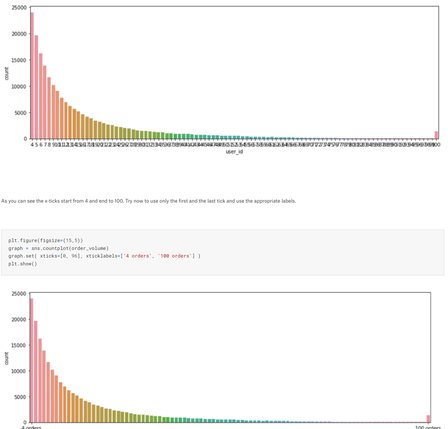
Use the value_counts() method to identify how many order each customer has placed. (user_id)

In [10]:
```
order_volume = orders.user_id.value_counts()
order_volume.tail()
```

Out[10]:
```
24224    4
199589   4
32439    4
21318    4
196838   4
Name: user_id, dtype: int64
```

Now use order_volume to produce a countplot that shows the distribution of the number of customers per volume of orders.

In [11]:
```
plt.figure(figsize=(15,5))
graph = sns.countplot(order_volume)
plt.show()
```



As you can see the x-ticks start from 4 and end to 100. Try now to use only the first and the last tick and use the appropiate labels.

In [12]:
```
plt.figure(figsize=(15,5))
graph = sns.countplot(order_volume)
graph.set( xticks=[0, 96], xticklabels=['4 orders', '100 orders'] )
plt.show()
```