Section 1: Brand Analysis

In this lesson, the impact of a brand name on consumer purchasing behavior is discussed.

We'll cover the following

Exploring the data

The data files for both *October* &and*November* are very large in size so for this exercise, the data for only *October* will be selected and used. Let's review and analyze what data is stored in which format.

The following code reads the *CSV* data file and prints the `DataFrame`.

> **Note**: The code in this chapter is not runnable due to memory constraints. Please run them locally along with each lesson.

```
import pandas as pd

df = pd.read_csv("2019-Oct.csv") # Reading the data from file

print(df)
```

	event_time	event_type	product_id	category_id	category_code	brand	price	user_id	user_session
0	2019-10-01 00:00:00 UTC	view	44600062	2103807459595387724	NaN	shiseido	35.79	541312140	72d76fde-8bb3-4e00-8c23-a032dfed738c
1	2019-10-01 00:00:00 UTC	view	3900821	2053013552326770905	appliances.environment.water_heater	aqua	33.20	554748717	9333dfbd-b87a-4708-9857-6336556b0fcc
2	2019-10-01 00:00:01 UTC	view	17200506	2053013559792632471	furniture.living_room.sofa	NaN	543.10	519107250	566511c2-e2e3-422b-b695-cf8e6e792ca8
3	2019-10-01 00:00:01 UTC	view	1307067	2053013558920217191	computers.notebook	lenovo	251.74	550050854	7c90fc70-0e80-4590-96f3-13c02c18c713
4	2019-10-01 00:00:04 UTC	view	1004237	2053013555631882655	electronics.smartphone	apple	1081.98	535871217	c6bd7419-2748-4c56-95b4-8cec9ff8b80d
...	...	...	...	...	...	...	...	...	...
42448759	2019-10-31 23:59:58 UTC	view	2300275	2053013560530830019	electronics.camera.video	gopro	527.40	537931532	22c57267-da98-4f28-9a9c-18bb5b385193
42448760	2019-10-31 23:59:58 UTC	view	10800172	2053013554994348409	NaN	redmond	61.75	527322328	5054190a-46cb-4211-a8f1-16fc1a060ed8

According to the output, there are **nine** columns in the `DataFrame`, which are described below:

- `event_time`: The exact time when the activity occurred by a user
- `event_type`: The type of activity occurred; there are three types in our case, i.e, *view*, *cart*, and *purchase*
- `product_id`: The unique ID of a particular product
- `category_id`: The unique ID of the category to which the product belongs to
- `category_code`: The unique category code to which the product belongs to
- `brand`: The brand name of the selected product
- `price`: The price of the selected product
- `user_id`: The unique ID of the user
- `user_session`: The unique ID generated every time a user visits the site. It is different for every visit of a particular user

Brand analysis

A *brand* is a term that differentiates one product from another. In this analysis, we will review whether people like to purchase products with a popular brand or a product without a brand.

For this analysis, only the products actually bought by the users will be considered. In our dataset, the products which have no brand are given a `NaN` value.
This will be done in two steps:

1. Separate the original `DataFrame` into two DataFrames. One with all the products with brands and one with all the products without brands.

2. Fetch all those rows from the two `DataFrames` where the `event_type` value is `purchase`.

As a final result, two `Dataframes` will be obtained containing the brand products with and without, that was purchased.

```
import pandas as pd

df = pd.read_csv("2019-Oct.csv") # Reading the data from file




# Step 1

# Fetch rows with brand
with_brand = df[df['brand'].notna()]

# Fetch rows without brand
without_brand = df[df['brand'].isna()]

# Step 2

# Purchased products with brands
with_brand = with_brand[with_brand['event_type'] == 'purchase']
print(with_brand)

# Purchased products without brands
without_brand = without_brand[without_brand['event_type'] == 'purchase']
print(without_brand)
```

1	2019-10-01 00:00:00+00:00	view	3900821	2053013552326770905	appliances.environment.water_heater	aqua	33.20	554748717	9333dfbd-b87a-4708-9857-6336556b0fcc
3	2019-10-01 00:00:01+00:00	view	1307067	2053013558920217191	computers.notebook	lenovo	251.74	550050854	7c90fc70-0e80-4590-96f3-13c02c18c713
4	2019-10-01 00:00:04+00:00	view	1004237	2053013555631882655	electronics.smartphone	apple	1081.98	535871217	c6bd7419-2748-4c56-95b4-8cec9ff8b80d
5	2019-10-01 00:00:05+00:00	view	1480613	2053013561092866779	computers.desktop	pulser	908.62	512742880	0d0d91c2-c9c2-4e81-90a5-86594dec0db9
...	...	...	...	...	...	...	...	...	...
42448759	2019-10-31 23:59:58+00:00	view	2300275	2053013560530830019	electronics.camera.video	gopro	527.40	537931532	22c57267-da98-4f28-9a9c-18bb5b385193
42448760	2019-10-31 23:59:58+00:00	view	10800172	2053013554994348409	NaN	redmond	61.75	527322328	5054190a-46cb-4211-a8f1-16fc1a060ed8
42448761	2019-10-31 23:59:58+00:00	view	5701038	2053013553970938175	auto.accessories.player	kenwood	128.70	566280422	05b6c62b-992f-4e8e-91f7-961bcb4719cd
42448762	2019-10-31 23:59:59+00:00	view	21407424	2053013561579406073	electronics.clocks	tissot	689.85	513118352	4c14bf2a-2820-4504-929d-046356a5a204
42448763	2019-10-31 23:59:59+00:00	view	13300120	2053013557166998015	NaN	swisshome	155.73	525266378	6e57d2d7-6022-46e6-81d6-fa77f14cefd8

36331684 rows × 11 columns

With brand

2	2019-10-01 00:00:01+00:00	view	17200506	2053013559792632471	furniture.living_room.sofa	NaN	543.10	519107250	566511c2-e2e3-422b-b695-cf8e6e792ca8
15	2019-10-01 00:00:17+00:00	view	23100006	2053013561638126333	NaN	NaN	357.79	513642368	17566c27-0a8f-4506-9f30-c6a2ccbf583b
26	2019-10-01 00:00:24+00:00	view	34700031	2061717937420501730	NaN	NaN	151.87	539512263	f27a45f8-fb98-459a-96a6-45271f56a987
31	2019-10-01 00:00:26+00:00	view	13500046	2053013557099889147	furniture.bedroom.bed	NaN	60.75	555446365	7f0062d8-ead0-4e0a-96f6-43a0b79a2fc4
32	2019-10-01 00:00:27+00:00	view	31501072	2053013558031024687	NaN	NaN	165.64	550978835	6280d577-25c8-4147-99a7-abc6048498d6
...	...	...	...	...	...	...	...	...	...
42448746	2019-10-31 23:59:54+00:00	view	1002786	2053013555631882655	electronics.smartphone	NaN	391.26	512789086	cc782b99-88ab-4573-8311-c62e1d447757
42448747	2019-10-31 23:59:54+00:00	view	25600078	2053013559675191951	NaN	NaN	81.86	522031876	39d48518-9fca-4df3-9724-950cd6ec44eb
42448750	2019-10-31 23:59:55+00:00	view	42200036	2095518917320508073	NaN	NaN	17.50	515474976	222c370b-0fac-4287-982b-e340f5eaf3a1
42448753	2019-10-31 23:59:57+00:00	view	21408491	2053013561579406073	electronics.clocks	NaN	350.07	553802615	e09684bb-0c95-4f67-98d1-59fc593f3890
42448754	2019-10-31 23:59:57+00:00	view	44300011	2100825583029060150	apparel.jeans	NaN	50.45	545220871	f278cca0-e0f6-49a3-819a-d961998282d5

6117080 rows × 11 columns

Without brand

In the output, we can see that the products have been correctly filtered, and two `DataFrames` have been obtained: one with purchased branded products and the other with purchased non-branded products.

On **line 11**, those rows from `df` are selected where the `brand` value is not null or `NaN`. A new `DatFrame` is returned which only contains products that belong to a brand.

On **line 14**, the rows from `df` were selected where `brand` value is null or `NaN`. A new `DatFrame` will be returned which will only contain products that don't belong to a brand.

On **line 19**, those entries from the `with_brand` DataFrame where `event_type` is equal to `purchased` are filtered to get information for only those products that were bought.

On **line 23**, the entries from the `without_brand` DataFrame where `event_type` is equal to `purchased` are filtered to get information for only those products that were bought.

At the end of each output, the number of rows for each `DataFrame` is mentioned. Let's review how much percentage of branded and non-branded products were bought.

```
# Get length of original dataframe with purchased products
org = len(df[df['event_type'] == 'purchase'])

# Divide the length of with_brand dataframe with length org dataframe
brand_p = len(with_brand) / org
print(brand_p * 100)

# Divide the length of without_brand dataframe with length org dataframe
brand_a = len(without_brand) / org
print(brand_a * 100)
```

brand_p = 92.15116396468193 %

brand_a = 7.8488360353180795 %

On **line 5**, the length of products with the brand `DataFrame` is divided by all of the purchased products to get the percentage of their sales.

On **line 9**, the length of products without the brand `DataFrame` is divided by all of

On **line 9**, the length of products without the brand `DataFrame` is divided by all of the purchased products to get the percentage of their sales.

According to the above output, approximately **92%** of the purchased products were associated with a brand, and only **8%** of products without a brand were bought.

The hypothesis

A hypothesis can be drawn based on the above results.

- For *marketers*, most of the marketing budget should be allotted to the advertisement of branded products.
- For inventors or entrepreneurs, always introduce the product with a brand name because products without a brand have a very low probability of getting bought.

In the next lesson, we analyze the customer's activity on the website.