

Python Data Manipulation Assignment

Created by : Bhuvnesh

Website : statinfer.com (<https://statinfer.com/>)

Dataset Information

This is a Brazilian ecommerce public dataset of orders made at Olist Store. The dataset has information of 100k orders from 2016 to 2018 made at multiple marketplaces in Brazil. Its features allows viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers. We also released a geolocation dataset that relates Brazilian zip codes to lat/lng coordinates.

This is real commercial data, it has been anonymised, and references to the companies and partners in the review text have been replaced

In [3]:

```
#from google.colab import drive
#drive.mount('/content/drive')
```

In [4]:

```
input = "/content/drive/My Drive/Training/ML_Full_Semester/Assignments/Python_Data_Manipuat
# input="/content/drive/My Drive/Training/ML_Full_Semister/Assignments/Python_Data_Manipuat
```

In [5]:

```
import pandas as pd
import numpy as np
```

1.Create a dictionary object

1.Create a dictionary object with two keys with 5 values each, say Name and Age. You can use any arbitrary 5 name and their age. Then create a Pandas DataFrame from this dictionary.

In [6]:

```
# creating panda data frame from dictionary
data = {'name': ['Stan', 'Kyle', 'Eric', 'Kenny'],
        'age': [9, 9, 11, 10]}
df = pd.DataFrame(data)
df
```

Out[6]:

	name	age
0	Stan	9
1	Kyle	9
2	Eric	11
3	Kenny	10

2.Download the datasets

2.Download the following datasets, and unzip it within this notebook with code. And Read all the files as pandas dataframes.Display basic information

In [7]:

```
df_item = pd.read_csv(input+"/olist_order_items_dataset.csv")
df_reviews = pd.read_csv(input+"/olist_order_reviews_dataset.csv")
df_orders = pd.read_csv(input+"/olist_orders_dataset.csv")
df_products = pd.read_csv(input+"/olist_products_dataset.csv")
df_geolocation = pd.read_csv(input+"/olist_geolocation_dataset.csv")
df_sellers = pd.read_csv(input+"/olist_sellers_dataset.csv")
df_order_pay = pd.read_csv(input+"/olist_order_payments_dataset.csv")
df_customers = pd.read_csv(input+"/olist_customers_dataset.csv")
df_category = pd.read_csv(input+"/product_category_name_translation.csv")
```

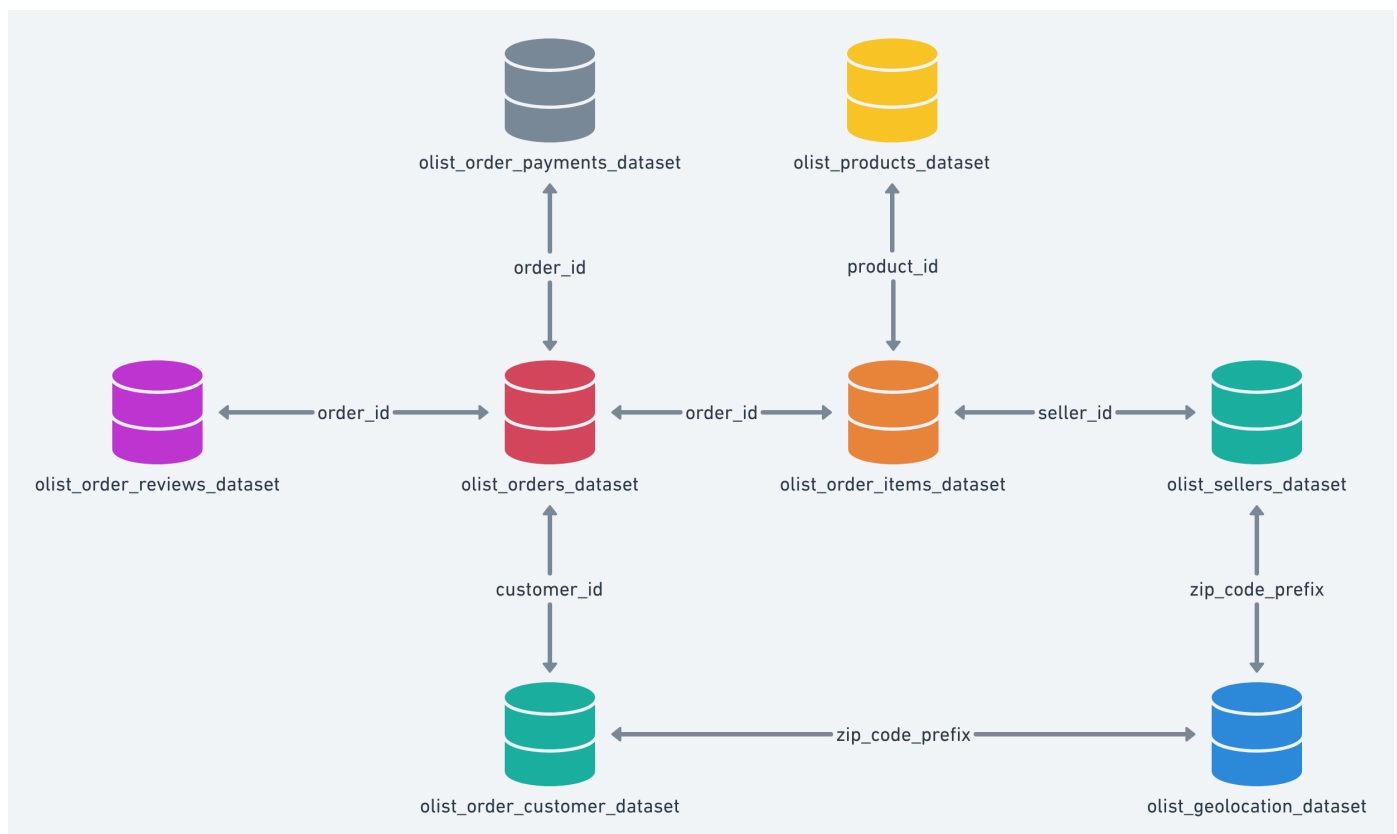
In [8]:

```
datasets_list=[df_item,df_reviews,df_orders,df_products,df_geolocation,df_sellers,df_order_
for data_name in datasets_list:
    name =[x for x in globals() if globals()[x] is data_name][0]
    print("Dataframe Name is: %s" % name)
    print(data_name.info())
    print("=====\n")
```

```
Dataframe Name is: df_item
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 112650 entries, 0 to 112649
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  -
0   order_id               112650 non-null object
1   order_item_id          112650 non-null int64
2   product_id             112650 non-null object
3   seller_id              112650 non-null object
4   shipping_limit_date    112650 non-null object
5   price                  112650 non-null float64
6   freight_value          112650 non-null float64
dtypes: float64(2), int64(1), object(4)
memory usage: 6.0+ MB
None
=====
```

```
Dataframe Name is: df_reviews
```

3.Look the Schema below and Create the Full dataframe using pandas merge. Skip merging olist_geolocation_dataset.csv



In [9]:

```
print("df_orders", df_orders.shape)
df_full = pd.merge(df_orders, df_item, on='order_id', how='left')
print("items added", df_full.shape)
# df_full = df_full.merge(df_order_pay, on='order_id', how='outer', validate='m:m')
df_full = df_full.merge(df_order_pay, on='order_id', how='left')
print("df_order_pay added", df_full.shape)
df_full = df_full.merge(df_reviews, on='order_id', how='left')
print("df_reviews added", df_full.shape)
df_full = df_full.merge(df_products, on='product_id', how='left')
print("df_products added", df_full.shape)
df_full = df_full.merge(df_customers, on='customer_id', how='left')
print("df_customers added", df_full.shape)
df_full = df_full.merge(df_sellers, on='seller_id', how='left')
print("df_sellers added", df_full.shape)
```

```
df_orders (99441, 8)
items added (113425, 14)
df_order_pay added (118434, 18)
df_reviews added (119151, 24)
df_products added (119151, 32)
df_customers added (119151, 36)
df_sellers added (119151, 39)
```

In [10]:

```
df_full.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 119151 entries, 0 to 119150
Data columns (total 39 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   order_id                             119151 non-null  object
 1   customer_id                           119151 non-null  object
 2   order_status                           119151 non-null  object
 3   order_purchase_timestamp              119151 non-null  object
 4   order_approved_at                     118974 non-null  object
 5   order_delivered_carrier_date          117065 non-null  object
 6   order_delivered_customer_date         115730 non-null  object
 7   order_estimated_delivery_date         119151 non-null  object
 8   order_item_id                         118318 non-null  float64
 9   product_id                           118318 non-null  object
10   seller_id                             118318 non-null  object
11   shipping_limit_date                   118318 non-null  object
12   price                                 118318 non-null  float64
13   freight_value                         118318 non-null  float64
14   payment_sequential                    119151 non-null  float64
```

4.Show the head() of the dataframe and print the transpose; in such a way that We can Scroll through all the column names.

In [11]:

```
df_full.head(20).T
```

Out[11]:

0		
order_id	e481f51cbdc54678b7cc49136f2d6af7	e481f51cbdc54678b7cc49136f2d6af7
customer_id	9ef432eb6251297304e76186b10a928d	9ef432eb6251297304e76186b10a928d
order_status	delivered	delivered
order_purchase_timestamp	2017-10-02 10:56:33	2017-10-02 10:56:33
order_approved_at	2017-10-02 11:07:15	2017-10-02 11:07:15
order_delivered_carrier_date	2017-10-04 19:55:00	2017-10-04 19:55:00
order_delivered_customer_date	2017-10-10 21:25:13	2017-10-10 21:25:13
order_estimated_delivery_date	2017-10-18 00:00:00	2017-10-18 00:00:00
order_item_id	1	1
product_id	87285b34884572647811a353c7ac498a	87285b34884572647811a353c7ac498a
seller_id	3504c0cb71d7fa48d967e0e4c94d59d9	3504c0cb71d7fa48d967e0e4c94d59d9
shipping_limit_date	2017-10-06 11:07:15	2017-10-06 11:07:15
price	29.99	29.99
freight_value	8.72	8.72
payment_sequential	1	1
payment_type	credit_card	credit_card
payment_installments	1	1
payment_value	18.12	18.12
review_id	a54f0611adc9ed256b57ede6b6eb5114	a54f0611adc9ed256b57ede6b6eb5114
review_score	4	4
review_comment_title	NaN	NaN
review_comment_message	Não testei o produto ainda, mas ele veio corre...	Não testei o produto ainda, mas ele veio corre...
review_creation_date	2017-10-11 00:00:00	2017-10-11 00:00:00
review_answer_timestamp	2017-10-12 03:43:48	2017-10-12 03:43:48
product_category_name	utilidades_domesticas	utilidades_domesticas
product_name_lenght	40	40
product_description_lenght	268	268
product_photos_qty	4	4
product_weight_g	500	500
product_length_cm	19	19
product_height_cm	8	8
product_width_cm	13	13
customer_unique_id	7c396fd4830fd04220f754e42b4e5bff	7c396fd4830fd04220f754e42b4e5bff
customer_zip_code_prefix	3149	3149

0

customer_city	sao paulo	sa
customer_state	SP	
seller_zip_code_prefix	9350	
seller_city	maua	
seller_state	SP	

In [12]:

```
df_full.shape
```

Out[12]:

(119151, 39)

5 Map the column product_category_name from Portuguese to English using product_category_name_translation.csv

Hint: create a map dictionary from product_category_name_translation.csv , and use the dictionary to map or replace the Portuguese names to English

In [13]:

```
df_category.head()
```

Out[13]:

	product_category_name	product_category_name_english
0	beleza_saude	health_beauty
1	informatica_acessorios	computers_accessories
2	automotivo	auto
3	cama_mesa_banho	bed_bath_table
4	moveis_decoracao	furniture_decor

In [14]:

```
rename_dict = df_category.set_index('product_category_name').to_dict()['product_category_name_english']
df_full['product_category_name'] = df_full['product_category_name'].replace(rename_dict)
```

In [15]:

```
df_full.head().T
```

Out[15]:

0		
order_id	e481f51cbdc54678b7cc49136f2d6af7	e481f51cbdc54678b7cc49136f2d6af7
customer_id	9ef432eb6251297304e76186b10a928d	9ef432eb6251297304e76186b10a928d
order_status	delivered	delivered
order_purchase_timestamp	2017-10-02 10:56:33	2017-10-02 10:56:33
order_approved_at	2017-10-02 11:07:15	2017-10-02 11:07:15
order_delivered_carrier_date	2017-10-04 19:55:00	2017-10-04 19:55:00
order_delivered_customer_date	2017-10-10 21:25:13	2017-10-10 21:25:13
order_estimated_delivery_date	2017-10-18 00:00:00	2017-10-18 00:00:00
order_item_id	1	1
product_id	87285b34884572647811a353c7ac498a	87285b34884572647811a353c7ac498a
seller_id	3504c0cb71d7fa48d967e0e4c94d59d9	3504c0cb71d7fa48d967e0e4c94d59d9
shipping_limit_date	2017-10-06 11:07:15	2017-10-06 11:07:15
price	29.99	29.99
freight_value	8.72	8.72
payment_sequential	1	1
payment_type	credit_card	credit_card
payment_installments	1	1
payment_value	18.12	18.12
review_id	a54f0611adc9ed256b57ede6b6eb5114	a54f0611adc9ed256b57ede6b6eb5114
review_score	4	4
review_comment_title	NaN	NaN
review_comment_message	Não testei o produto ainda, mas ele veio corre...	Não testei o produto ainda, mas ele veio corre...
review_creation_date	2017-10-11 00:00:00	2017-10-11 00:00:00
review_answer_timestamp	2017-10-12 03:43:48	2017-10-12 03:43:48
product_category_name	housewares	housewares
product_name_lenght	40	40
product_description_lenght	268	268
product_photos_qty	4	4
product_weight_g	500	500
product_length_cm	19	19
product_height_cm	8	8
product_width_cm	13	13
customer_unique_id	7c396fd4830fd04220f754e42b4e5bff	7c396fd4830fd04220f754e42b4e5bff
customer_zip_code_prefix	3149	3149

0

customer_city	sao paulo	sa
customer_state	SP	
seller_zip_code_prefix	9350	
seller_city	maua	
seller_state	SP	

6. Use single line code to print missing values and number of unique values in each column.

In [16]:

```
# Single line code for missing values in each column
df_full.isnull().sum()
```

Out[16]:

```
order_id                0
customer_id             0
order_status            0
order_purchase_timestamp 0
order_approved_at       177
order_delivered_carrier_date 2086
order_delivered_customer_date 3421
order_estimated_delivery_date 0
order_item_id           833
product_id              833
seller_id               833
shipping_limit_date     833
price                   833
freight_value           833
payment_sequential       3
payment_type             3
payment_installments     3
payment_value            3
review_id                0
review_score             0
review_comment_title     104962
review_comment_message   67901
review_creation_date     0
review_answer_timestamp  0
product_category_name    2542
product_name_lenght      2542
product_description_lenght 2542
product_photos_qty       2542
product_weight_g         853
product_length_cm        853
product_height_cm        853
product_width_cm         853
customer_unique_id       0
customer_zip_code_prefix 0
customer_city            0
customer_state           0
seller_zip_code_prefix   833
seller_city              833
seller_state             833
dtype: int64
```

In [17]:

```
# Single line code for unique values in each column
df_full.nunique()
```

Out[17]:

```
order_id          99441
customer_id       99441
order_status       8
order_purchase_timestamp    98875
order_approved_at    90733
order_delivered_carrier_date    81018
order_delivered_customer_date    95664
order_estimated_delivery_date    459
order_item_id      21
product_id        32951
seller_id         3095
shipping_limit_date    93318
price            5968
freight_value      6999
payment_sequential    29
payment_type         5
payment_installments    24
payment_value      29077
review_id         99173
review_score        5
review_comment_title    4600
review_comment_message    36921
review_creation_date    637
review_answer_timestamp    99010
product_category_name    73
product_name_lenght     66
product_description_lenght    2960
product_photos_qty     19
product_weight_g      2204
product_length_cm      99
product_height_cm     102
product_width_cm       95
customer_unique_id    96096
customer_zip_code_prefix    14994
customer_city         4119
customer_state        27
seller_zip_code_prefix    2246
seller_city           611
seller_state          23
dtype: int64
```

7. Use `.iloc[]` to select First 8 columns and row indices from 10000-12000. Write shortest code.

In [18]:

```
df_full.iloc[10000:12000, :8]
```

Out[18]:

	order_id	customer_id	order_status	or
10000	cdc1504eb9521a2941c00363ee8288d2	87e3b43896edfc6786a0d8ab36a14202	delivered	
10001	f49b0d1c77d2f8b37c7579f4cbc8264e	60084b7d470df61a877bc270cae7f70f	delivered	
10002	60cea07de0865aeead1984c1c97dee57	e31cc7a1dbedf7b8118bf23028475df4	delivered	
10003	fecadea4b60522702bd322933da20c9e	52949927b7bca40124954b070e2ccd44	delivered	
10004	9694aa09499321709cdb542840ebbbb2	18cf90f4d4f765bd1ca02c2af5e214ea	delivered	
...	
11995	94b9dffc1d9b49273c91598fc0dc5e6	6236a1fd73ab92114378e88fe16698ea	delivered	
11996	0c986e5002868d137f566b34fb183827	56a65a21f846976ecbc7b7958ff32f74	delivered	
11997	487cdbe6d900d62275117d1fd45674bf	1dc9dd1db9aebd0a3c03eefeb3f7ee1f	delivered	
11998	809a282bbd5dbcabb6f2f724fca862ec	622e13439d6b5a0b486c435618b2679e	canceled	
11999	c0cb4eb4f8b433eb1d73091276e12ebd	793206f0990570a0101ebb9558de79c3	delivered	

2000 rows × 8 columns



8.Now use .loc[] to select First 8 columns and row indices from 10000-12000. Write shortest code.

In [19]:

```
df_full.loc[10000:12000, : 'order_purchase_timestamp']
```

Out[19]:

	order_id	customer_id	order_status	or
10000	cdc1504eb9521a2941c00363ee8288d2	87e3b43896edfc6786a0d8ab36a14202	delivered	
10001	f49b0d1c77d2f8b37c7579f4cbc8264e	60084b7d470df61a877bc270cae7f70f	delivered	
10002	60cea07de0865aeead1984c1c97dee57	e31cc7a1dbedf7b8118bf23028475df4	delivered	
10003	fecadea4b60522702bd322933da20c9e	52949927b7bca40124954b070e2ccd44	delivered	
10004	9694aa09499321709cdb542840ebbbb2	18cf90f4d4f765bd1ca02c2af5e214ea	delivered	
...	
11996	0c986e5002868d137f566b34fb183827	56a65a21f846976ecbc7b7958ff32f74	delivered	
11997	487cdbe6d900d62275117d1fd45674bf	1dc9dd1db9aebd0a3c03eefeb3f7ee1f	delivered	
11998	809a282bbd5dbcabb6f2f724fca862ec	622e13439d6b5a0b486c435618b2679e	canceled	
11999	c0cb4eb4f8b433eb1d73091276e12ebd	793206f0990570a0101ebb9558de79c3	delivered	
12000	164f22d152f646ca5ebb5a2049e3adfd	b22ca1cb1efb9b7b3dadf51aae18d481	delivered	

2001 rows × 4 columns



9.Convert Date columns to datetime objects

In [20]:

```
df_full.dtypes
```

Out[20]:

order_id	object
customer_id	object
order_status	object
order_purchase_timestamp	object
order_approved_at	object
order_delivered_carrier_date	object
order_delivered_customer_date	object
order_estimated_delivery_date	object
order_item_id	float64
product_id	object
seller_id	object
shipping_limit_date	object
price	float64
freight_value	float64
payment_sequential	float64
payment_type	object
payment_installments	float64
payment_value	float64
review_id	object
review_score	int64
review_comment_title	object
review_comment_message	object
review_creation_date	object
review_answer_timestamp	object
product_category_name	object
product_name_lenght	float64
product_description_lenght	float64
product_photos_qty	float64
product_weight_g	float64
product_length_cm	float64
product_height_cm	float64
product_width_cm	float64
customer_unique_id	object
customer_zip_code_prefix	int64
customer_city	object
customer_state	object
seller_zip_code_prefix	float64
seller_city	object
seller_state	object
dtype:	object

In [21]:

```
date_cols = ['order_purchase_timestamp',
             'order_approved_at',
             'order_delivered_carrier_date',
             'order_delivered_customer_date',
             'order_estimated_delivery_date',
             'shipping_limit_date',
             'review_creation_date',
             'review_answer_timestamp']

df_full.loc[:, date_cols] = df_full.loc[:, date_cols].apply(pd.to_datetime, errors='coerce')
# df_full[date_cols] = df_full[date_cols].apply(pd.to_datetime, errors='coerce')
```

In [22]:

```
df_full.dtypes
```

Out[22]:

order_id	object
customer_id	object
order_status	object
order_purchase_timestamp	datetime64[ns]
order_approved_at	datetime64[ns]
order_delivered_carrier_date	datetime64[ns]
order_delivered_customer_date	datetime64[ns]
order_estimated_delivery_date	datetime64[ns]
order_item_id	float64
product_id	object
seller_id	object
shipping_limit_date	datetime64[ns]
price	float64
freight_value	float64
payment_sequential	float64
payment_type	object
payment_installments	float64
payment_value	float64
review_id	object
review_score	int64
review_comment_title	object
review_comment_message	object
review_creation_date	datetime64[ns]
review_answer_timestamp	datetime64[ns]
product_category_name	object
product_name_lenght	float64
product_description_lenght	float64
product_photos_qty	float64
product_weight_g	float64
product_length_cm	float64
product_height_cm	float64
product_width_cm	float64
customer_unique_id	object
customer_zip_code_prefix	int64
customer_city	object
customer_state	object
seller_zip_code_prefix	float64
seller_city	object
seller_state	object
dtype:	object

10. Create a subset with condition(s), using least lines of code possible:

1. Delivery was made within 5 days after order. Use: `order_purchase_timestamp` , `order_delivered_customer_date`
2. Review Score was less than or equal to 2. Use: `review_score` column
3. price was more than 199
4. and select only these columns: 'order_id' , 'customer_id', 'product_id', 'price', 'review_score', 'product_category_name', 'product_weight_g'

In [23]:

```
mask = (((df_full["order_purchase_timestamp"] - df_full["order_delivered_customer_date"]).dt.days < 5) && df_full["review_score"] <= 2 && df_full["price"] > 199)
df_full.loc[mask, ['order_id', 'customer_id', 'product_id', 'price', 'review_score', 'product_category_name', 'product_weight_g']]
```

Out[23]:

	order_id	customer_id
37	f70a0aff17df5a6cdd9a7196128bd354	456dc10730fbdba34615447ea195d643
48	6ea2f835b4556291ffdc53fa0b3b95e8	c7340080e394356141681bd4c9b8fe31
96	634e8f4c0f6744a626f77f39770ac6aa	05e996469a2bf9559c7122b87e156724
97	634e8f4c0f6744a626f77f39770ac6aa	05e996469a2bf9559c7122b87e156724
130	e1da8361c76cab67aa3588a1fbf1af54	dd854e24b40e3bc2b306946dee252015
...
118950	4cf09d9e5ebbe0f91ddd7bf9aae891cd	07b6b4fe5fefb948fc76b6d2bdba77d8
118968	87d30b7fd5316c576377b79548be7d84	f6707ad9321dba0eba6285da8c77b7e4
119004	3c042ee4b8b597c3d265a93a21bbf99f	d71a0d0cf6bbacec526203263382501b
119087	aa07fc0f496d65986abc9044683b8800	39256804b05cde32ac8f5ed003645f6b
119101	0fa1fab1d7c1211c824596ed5e111e3c	7f3bd6c94d2daf7b6462d1a894a775b4

2149 rows × 7 columns

11. Find the subset where `review_comment_message` Doesn't have word `ruim`.

in Portuguese `ruim` means bad or ruined

In [24]:

```
mask = (df_full['review_comment_message'].astype(str).str.contains(' ruim'))
df_full.loc[~mask].shape
```

Out[24]:

(118888, 39)

In [25]:

```
# We have 263 rows without ruim
df_full.loc[~mask].shape[0]-df_full.shape[0]
```

Out[25]:

-263

12.Create a new column `review_length` using `.apply()` on the column `review_comment_message`

In [26]:

```
def sentLenght(x):
    return len(x.split())

df_full['review_length'] = df_full['review_comment_message'].fillna('').apply(sentLenght)
```

13.Create a new column `day_of_order` , representing day of the week based on `order_purchase_timestamp` column

In [27]:

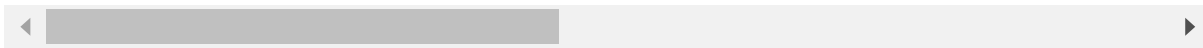
```
df_full['day_of_order'] = df_full["order_purchase_timestamp"].dt.dayofweek
df_full.head().T
```

Out[27]:

0		
order_id	e481f51cbdc54678b7cc49136f2d6af7	e481f51cbdc54678b7cc49136f2d6af7
customer_id	9ef432eb6251297304e76186b10a928d	9ef432eb6251297304e76186b10a928d
order_status	delivered	delivered
order_purchase_timestamp	2017-10-02 10:56:33	2017-10-02 10:56:33
order_approved_at	2017-10-02 11:07:15	2017-10-02 11:07:15
order_delivered_carrier_date	2017-10-04 19:55:00	2017-10-04 19:55:00
order_delivered_customer_date	2017-10-10 21:25:13	2017-10-10 21:25:13
order_estimated_delivery_date	2017-10-18 00:00:00	2017-10-18 00:00:00
order_item_id	1	1
product_id	87285b34884572647811a353c7ac498a	87285b34884572647811a353c7ac498a
seller_id	3504c0cb71d7fa48d967e0e4c94d59d9	3504c0cb71d7fa48d967e0e4c94d59d9
shipping_limit_date	2017-10-06 11:07:15	2017-10-06 11:07:15
price	29.99	29.99
freight_value	8.72	8.72
payment_sequential	1	1
payment_type	credit_card	credit_card
payment_installments	1	1
payment_value	18.12	18.12
review_id	a54f0611adc9ed256b57ede6b6eb5114	a54f0611adc9ed256b57ede6b6eb5114
review_score	4	4
review_comment_title	NaN	NaN
review_comment_message	Não testei o produto ainda, mas ele veio corre...	Não testei o produto ainda, mas ele veio corre...
review_creation_date	2017-10-11 00:00:00	2017-10-11 00:00:00
review_answer_timestamp	2017-10-12 03:43:48	2017-10-12 03:43:48
product_category_name	housewares	housewares
product_name_lenght	40	40
product_description_lenght	268	268
product_photos_qty	4	4
product_weight_g	500	500
product_length_cm	19	19
product_height_cm	8	8
product_width_cm	13	13
customer_unique_id	7c396fd4830fd04220f754e42b4e5bff	7c396fd4830fd04220f754e42b4e5bff
customer_zip_code_prefix	3149	3149

0

customer_city	sao paulo	sa
customer_state	SP	
seller_zip_code_prefix	9350	
seller_city	maua	
seller_state	SP	
review_length	32	
day_of_order	0	



14.Rename product_category_name to product_category

In [28]:

```
df_full=df_full.rename(columns = {'product_category_name': 'product_category'})
df_full.head().T
```

Out[28]:

0		
order_id	e481f51cbdc54678b7cc49136f2d6af7	e481f51cbdc54678b7cc49136f2d6af7
customer_id	9ef432eb6251297304e76186b10a928d	9ef432eb6251297304e76186b10a928d
order_status	delivered	delivered
order_purchase_timestamp	2017-10-02 10:56:33	2017-10-02 10:56:33
order_approved_at	2017-10-02 11:07:15	2017-10-02 11:07:15
order_delivered_carrier_date	2017-10-04 19:55:00	2017-10-04 19:55:00
order_delivered_customer_date	2017-10-10 21:25:13	2017-10-10 21:25:13
order_estimated_delivery_date	2017-10-18 00:00:00	2017-10-18 00:00:00
order_item_id	1	1
product_id	87285b34884572647811a353c7ac498a	87285b34884572647811a353c7ac498a
seller_id	3504c0cb71d7fa48d967e0e4c94d59d9	3504c0cb71d7fa48d967e0e4c94d59d9
shipping_limit_date	2017-10-06 11:07:15	2017-10-06 11:07:15
price	29.99	29.99
freight_value	8.72	8.72
payment_sequential	1	1
payment_type	credit_card	credit_card
payment_installments	1	1
payment_value	18.12	18.12
review_id	a54f0611adc9ed256b57ede6b6eb5114	a54f0611adc9ed256b57ede6b6eb5114
review_score	4	4
review_comment_title	NaN	NaN
review_comment_message	Não testei o produto ainda, mas ele veio corre...	Não testei o produto ainda, mas ele veio corre...
review_creation_date	2017-10-11 00:00:00	2017-10-11 00:00:00
review_answer_timestamp	2017-10-12 03:43:48	2017-10-12 03:43:48
product_category	housewares	housewares
product_name_lenght	40	40
product_description_lenght	268	268
product_photos_qty	4	4
product_weight_g	500	500
product_length_cm	19	19
product_height_cm	8	8
product_width_cm	13	13
customer_unique_id	7c396fd4830fd04220f754e42b4e5bff	7c396fd4830fd04220f754e42b4e5bff
customer_zip_code_prefix	3149	3149

0

customer_city	sao paulo	sa
customer_state	SP	
seller_zip_code_prefix	9350	
seller_city	maua	
seller_state	SP	
review_length	32	
day_of_order	0	

15.Drop all rows where product_weight_g was <= 1000.

In [29]:

```
mask = df_full['product_weight_g'] <=1000
# df_train.drop(index = mask, inplace=True)
df_full.drop(index = df_full[mask].index)
```

Out[29]:

	order_id	customer_id	order_status	oi
7	a4591c265e18cb1dcee52889e2d8acc3	503740e9ca751ccdda7ba28e9ab8f608	delivered	
11	e69bfb5eb88e0ed6a785585b27e16dbf	31ad1d1b63eb9962463f764d4e6e0c9d	delivered	
12	e69bfb5eb88e0ed6a785585b27e16dbf	31ad1d1b63eb9962463f764d4e6e0c9d	delivered	
13	e6ce16cb79ec1d90b1da9085a6118aeb	494dded5b201313c64ed7f100595b95c	delivered	
14	e6ce16cb79ec1d90b1da9085a6118aeb	494dded5b201313c64ed7f100595b95c	delivered	
...	
119145	9c5dedf39a927c1b2549525ed64a053c	39bd1228ee8140590ac3aca26f2dfe00	delivered	
119146	63943bddc261676b46f01ca7ac2f7bd8	1fca14ff2861355f6e5f14306ff977a7	delivered	
119147	83c1379a015df1e13d02aae0204711ab	1aa71eb042121263aafbe80c1b562c9c	delivered	
119148	11c177c8e97725db2631073c19f07b62	b331b74b18dc79bcd6532d51e1637c1	delivered	
119149	11c177c8e97725db2631073c19f07b62	b331b74b18dc79bcd6532d51e1637c1	delivered	

48063 rows × 41 columns

16.Drop all the columns if the missing values are more than

50%

In [30]:

```
to_drop = df_full.columns[(df_full.isnull().sum()/df_full.shape[0])>0.5]
df_full.drop(columns = to_drop).head().T
```

Out[30]:

	0	
order_id	e481f51cbdc54678b7cc49136f2d6af7	e481f51cbdc54678b7cc49136f2d6af7
customer_id	9ef432eb6251297304e76186b10a928d	9ef432eb6251297304e76186b10a928d
order_status	delivered	
order_purchase_timestamp	2017-10-02 10:56:33	2017-10-02 10:56:33
order_approved_at	2017-10-02 11:07:15	2017-10-02 11:07:15
order_delivered_carrier_date	2017-10-04 19:55:00	2017-10-04 19:55:00
order_delivered_customer_date	2017-10-10 21:25:13	2017-10-10 21:25:13
order_estimated_delivery_date	2017-10-18 00:00:00	2017-10-18 00:00:00
order_item_id	1	
product_id	87285b34884572647811a353c7ac498a	87285b34884572647811a353c7ac498a
seller_id	3504c0cb71d7fa48d967e0e4c94d59d9	3504c0cb71d7fa48d967e0e4c94d59d9
shipping_limit_date	2017-10-06 11:07:15	2017-10-06 11:07:15
price	29.99	
freight_value	8.72	
payment_sequential	1	
payment_type	credit_card	
payment_installments	1	
payment_value	18.12	
review_id	a54f0611adc9ed256b57ede6b6eb5114	a54f0611adc9ed256b57ede6b6eb5114
review_score	4	
review_creation_date	2017-10-11 00:00:00	2017-10-11 00:00:00
review_answer_timestamp	2017-10-12 03:43:48	2017-10-12 03:43:48
product_category	housewares	
product_name_lenght	40	
product_description_lenght	268	
product_photos_qty	4	
product_weight_g	500	
product_length_cm	19	
product_height_cm	8	
product_width_cm	13	
customer_unique_id	7c396fd4830fd04220f754e42b4e5bff	7c396fd4830fd04220f754e42b4e5bff
customer_zip_code_prefix	3149	
customer_city	sao paulo	
customer_state	SP	

	0
seller_zip_code_prefix	9350
seller_city	maua
seller_state	SP
review_length	32
day_of_order	0

17. Finally merge the geo-location data from olist_geolocation_dataset.csv to the dataframe.

Make sure that the primary key geolocation_zip_code_prefix has no duplicates

In [31]:

```
print("df_geolocation shape", df_geolocation.shape)
df_geolocation = df_geolocation.iloc[:, :3]
print("Duplicates in zip code", sum(df_geolocation.duplicated(["geolocation_zip_code_prefix"])
df_geolocation_uniq=df_geolocation.drop_duplicates(["geolocation_zip_code_prefix"])
print("df_geolocation_uniq shape", df_geolocation_uniq.shape)
```

```
df_geolocation shape (1000163, 5)
Duplicates in zip code 981148
df_geolocation_uniq shape (19015, 3)
```

In [32]:

```
print(df_full.shape)
```

```
(119151, 41)
```

In [33]:

```
# Merge for customer geo Locations
df_geolocation_uniq.columns = ['customer_zip_code_prefix', 'cust_geolocation_lat', 'cust_geolocation_lng']
df_full1 = pd.merge(df_full, df_geolocation_uniq, on='customer_zip_code_prefix', how='left')
```

In [34]:

```
print(df_full.shape)
```

```
(119151, 41)
```

In [35]:

```
# Merge for Seller geo Locations
df_geolocation_uniq.columns = ['seller_zip_code_prefix', 'seller_geolocation_lat', 'seller_geolocation_lng']
df_full12 = pd.merge(df_full1, df_geolocation_uniq, on='seller_zip_code_prefix', how='left')
```

In [36]:

```
print(df_full.shape)
print(df_full1.shape)
print(df_full2.shape)
```

(119151, 41)

(119151, 43)

(119151, 45)

In [37]:

```
df_full12.head().T
```

Out[37]:

	0	
order_id	e481f51cbdc54678b7cc49136f2d6af7	e481f51cbdc54678b7cc49
customer_id	9ef432eb6251297304e76186b10a928d	9ef432eb6251297304e7618
order_status	delivered	
order_purchase_timestamp	2017-10-02 10:56:33	2017-10-02 10:56:33
order_approved_at	2017-10-02 11:07:15	2017-10-02 11:07:15
order_delivered_carrier_date	2017-10-04 19:55:00	2017-10-04 19:55:00
order_delivered_customer_date	2017-10-10 21:25:13	2017-10-10 21:25:13
order_estimated_delivery_date	2017-10-18 00:00:00	2017-10-18 00:00:00
order_item_id	1	
product_id	87285b34884572647811a353c7ac498a	87285b34884572647811a35
seller_id	3504c0cb71d7fa48d967e0e4c94d59d9	3504c0cb71d7fa48d967e0e
shipping_limit_date	2017-10-06 11:07:15	2017-10-06 11:07:15
price	29.99	
freight_value	8.72	
payment_sequential	1	
payment_type	credit_card	
payment_installments	1	
payment_value	18.12	
review_id	a54f0611adc9ed256b57ede6b6eb5114	a54f0611adc9ed256b57ede
review_score	4	
review_comment_title	NaN	
review_comment_message	Não testei o produto ainda, mas ele veio corre...	Não testei o produto ain...
review_creation_date	2017-10-11 00:00:00	2017-10-11 00:00:00
review_answer_timestamp	2017-10-12 03:43:48	2017-10-12 03:43:48
product_category	housewares	
product_name_lenght	40	
product_description_lenght	268	
product_photos_qty	4	
product_weight_g	500	
product_length_cm	19	
product_height_cm	8	
product_width_cm	13	
customer_unique_id	7c396fd4830fd04220f754e42b4e5bff	7c396fd4830fd04220f754
customer_zip_code_prefix	3149	

0

customer_city	sao paulo
customer_state	SP
seller_zip_code_prefix	9350
seller_city	maua
seller_state	SP
review_length	32
day_of_order	0
cust_geolocation_lat	-23.5748
cust_geolocation_lng	-46.5875
seller_geolocation_lat	-23.6801
seller_geolocation_lng	-46.4525

18. Create a new column, Euclidean Distance (L2) between customer's and seller's geo-locations.

Hint use package called `geopy`

In [38]:

```
# !pip install geopy
from geopy import distance
```

In [39]:

```
#First we need to fix the missing values
```

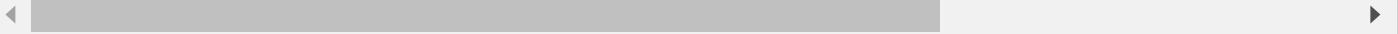
```
geo_loc_columns = ['cust_geolocation_lat', 'cust_geolocation_lng', 'seller_geolocation_lat']  
df_full2[geo_loc_columns] = df_full2[geo_loc_columns].fillna(0)  
df_full2.isnull().sum()
```

Out[39]:

```
order_id                0  
customer_id             0  
order_status            0  
order_purchase_timestamp 0  
order_approved_at       177  
order_delivered_carrier_date 2086  
order_delivered_customer_date 3421  
order_estimated_delivery_date 0  
order_item_id           833  
product_id              833  
seller_id               833  
shipping_limit_date     833  
price                   833  
freight_value           833  
payment_sequential      3  
payment_type            3  
payment_installments    3  
payment_value           3  
review_id               0  
review_score            0  
review_comment_title    104962  
review_comment_message  67901  
review_creation_date     0  
review_answer_timestamp  0  
product_category        2542  
product_name_lenght     2542  
product_description_lenght 2542  
product_photos_qty      2542  
product_weight_g        853  
product_length_cm       853  
product_height_cm       853  
product_width_cm        853  
customer_unique_id      0  
customer_zip_code_prefix 0  
customer_city           0  
customer_state          0  
seller_zip_code_prefix  833  
seller_city             833  
seller_state            833  
review_length           0  
day_of_order            0  
cust_geolocation_lat    0  
cust_geolocation_lng    0  
seller_geolocation_lat  0  
seller_geolocation_lng  0  
dtype: int64
```

In [40]:

```
df_full12['l2_distance'] = df_full12.apply(lambda x: distance.distance((x['cust_geolocation_l  
                                                                    (x['seller_geolocation  
                                                                    axis=1)
```

A horizontal scrollbar is located below the code input area. It consists of a grey track with a darker grey slider bar. The slider bar is positioned approximately one-third of the way from the left. Small black arrows are visible at both ends of the track.

In [41]:

```
df_full12.head().T
```

Out[41]:

	0	
order_id	e481f51cbdc54678b7cc49136f2d6af7	e481f51cbdc54678b7cc49
customer_id	9ef432eb6251297304e76186b10a928d	9ef432eb6251297304e7618
order_status	delivered	
order_purchase_timestamp	2017-10-02 10:56:33	2017-10-02 10:56:33
order_approved_at	2017-10-02 11:07:15	2017-10-02 11:07:15
order_delivered_carrier_date	2017-10-04 19:55:00	2017-10-04 19:55:00
order_delivered_customer_date	2017-10-10 21:25:13	2017-10-10 21:25:13
order_estimated_delivery_date	2017-10-18 00:00:00	2017-10-18 00:00:00
order_item_id	1	
product_id	87285b34884572647811a353c7ac498a	87285b34884572647811a35
seller_id	3504c0cb71d7fa48d967e0e4c94d59d9	3504c0cb71d7fa48d967e0e
shipping_limit_date	2017-10-06 11:07:15	2017-10-06 11:07:15
price	29.99	
freight_value	8.72	
payment_sequential	1	
payment_type	credit_card	
payment_installments	1	
payment_value	18.12	
review_id	a54f0611adc9ed256b57ede6b6eb5114	a54f0611adc9ed256b57ede
review_score	4	
review_comment_title	NaN	
review_comment_message	Não testei o produto ainda, mas ele veio corre...	Não testei o produto ain...
review_creation_date	2017-10-11 00:00:00	2017-10-11 00:00:00
review_answer_timestamp	2017-10-12 03:43:48	2017-10-12 03:43:48
product_category	housewares	
product_name_lenght	40	
product_description_lenght	268	
product_photos_qty	4	
product_weight_g	500	
product_length_cm	19	
product_height_cm	8	
product_width_cm	13	
customer_unique_id	7c396fd4830fd04220f754e42b4e5bff	7c396fd4830fd04220f754
customer_zip_code_prefix	3149	

0

customer_city	sao paulo
customer_state	SP
seller_zip_code_prefix	9350
seller_city	maua
seller_state	SP
review_length	32
day_of_order	0
cust_geolocation_lat	-23.5748
cust_geolocation_lng	-46.5875
seller_geolocation_lat	-23.6801
seller_geolocation_lng	-46.4525
l2_distance	18.0511

In [42]:

```
df_full12.shape
```

Out[42]:

```
(119151, 46)
```

19. Use groupby to find the Statewise number of unique orders. And which state accounts for most orders?

In [43]:

```
df_full12.groupby('customer_state')['order_id'].nunique().sort_values(ascending=False)
```

Out[43]:

```
customer_state
SP      41746
RJ      12852
MG      11635
RS       5466
PR       5045
SC       3637
BA       3380
DF       2140
ES       2033
GO       2020
PE       1652
CE       1336
PA        975
MT        907
MA        747
MS        715
PB        536
PI        495
RN        485
AL        413
SE        350
TO        280
RO        253
AM        148
AC         81
AP         68
RR         46
Name: order_id, dtype: int64
```

20.Export the file as .csv

In [44]:

```
df_full12.to_csv("final.csv", index=False)
```

In [44]: