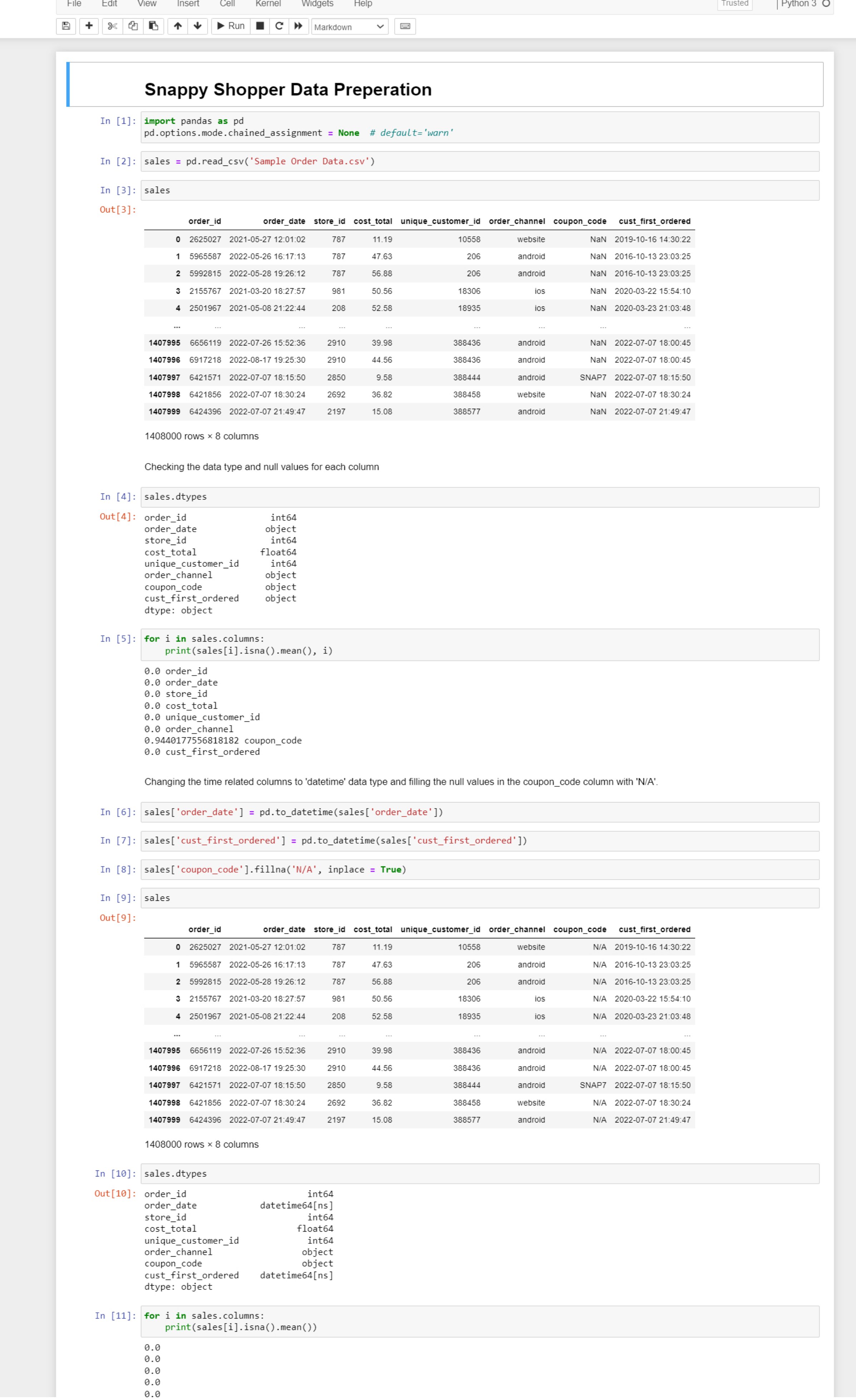

Snappy Shopper Data Preperation

In [1]:
```
import pandas as pd
pd.options.mode.chained_assignment = None  # default='warn'
```

In [2]:
```
sales = pd.read_csv('Sample Order Data.csv')
```

In [3]:
```
sales
```

Out[3]:

	order_id	order_date	store_id	cost_total	unique_customer_id	order_channel	coupon_code	cust_first_ordered
0	2625027	2021-05-27 12:01:02	787	11.19	10558	website	NaN	2019-10-16 14:30:22
1	5965587	2022-05-26 16:17:13	787	47.63	206	android	NaN	2016-10-13 23:03:25
2	5992815	2022-05-28 19:26:12	787	56.88	206	android	NaN	2016-10-13 23:03:25
3	2155767	2021-03-20 18:27:57	981	50.56	18306	ios	NaN	2020-03-22 15:54:10
4	2501967	2021-05-08 21:22:44	208	52.58	18935	ios	NaN	2020-03-23 21:03:48
...	...	...	...	...	...	...	...	...
1407995	6656119	2022-07-26 15:52:36	2910	39.98	388436	android	NaN	2022-07-07 18:00:45
1407996	6917218	2022-08-17 19:25:30	2910	44.56	388436	android	NaN	2022-07-07 18:00:45
1407997	6421571	2022-07-07 18:15:50	2850	9.58	388444	android	SNAP7	2022-07-07 18:15:50
1407998	6421856	2022-07-07 18:30:24	2692	36.82	388458	website	NaN	2022-07-07 18:30:24
1407999	6424396	2022-07-07 21:49:47	2197	15.08	388577	android	NaN	2022-07-07 21:49:47

1408000 rows × 8 columns

Checking the data type and null values for each column

In [4]:
```
sales.dtypes
```

Out[4]:
```
order_id                int64
order_date             object
store_id                int64
cost_total            float64
unique_customer_id      int64
order_channel          object
coupon_code            object
cust_first_ordered     object
dtype: object
```

In [5]:
```
for i in sales.columns:
    print(sales[i].isna().mean(), i)
```

```
0.0 order_id
0.0 order_date
0.0 store_id
0.0 cost_total
0.0 unique_customer_id
0.0 order_channel
0.9440177556818182 coupon_code
0.0 cust_first_ordered
```

Changing the time related columns to 'datetime' data type and filling the null values in the coupon_code column with 'N/A'.

In [6]:
```
sales['order_date'] = pd.to_datetime(sales['order_date'])
```

In [7]:
```
sales['cust_first_ordered'] = pd.to_datetime(sales['cust_first_ordered'])
```

In [8]:
```
sales['coupon_code'].fillna('N/A', inplace = True)
```

In [9]:
```
sales
```

Out[9]:

	order_id	order_date	store_id	cost_total	unique_customer_id	order_channel	coupon_code	cust_first_ordered
0	2625027	2021-05-27 12:01:02	787	11.19	10558	website	N/A	2019-10-16 14:30:22
1	5965587	2022-05-26 16:17:13	787	47.63	206	android	N/A	2016-10-13 23:03:25
2	5992815	2022-05-28 19:26:12	787	56.88	206	android	N/A	2016-10-13 23:03:25
3	2155767	2021-03-20 18:27:57	981	50.56	18306	ios	N/A	2020-03-22 15:54:10
4	2501967	2021-05-08 21:22:44	208	52.58	18935	ios	N/A	2020-03-23 21:03:48
...	...	...	...	...	...	...	...	...
1407995	6656119	2022-07-26 15:52:36	2910	39.98	388436	android	N/A	2022-07-07 18:00:45
1407996	6917218	2022-08-17 19:25:30	2910	44.56	388436	android	N/A	2022-07-07 18:00:45
1407997	6421571	2022-07-07 18:15:50	2850	9.58	388444	android	SNAP7	2022-07-07 18:15:50
1407998	6421856	2022-07-07 18:30:24	2692	36.82	388458	website	N/A	2022-07-07 18:30:24
1407999	6424396	2022-07-07 21:49:47	2197	15.08	388577	android	N/A	2022-07-07 21:49:47

1408000 rows × 8 columns

In [10]:
```
sales.dtypes
```

Out[10]:
```
order_id                       int64
order_date            datetime64[ns]
store_id                       int64
cost_total                   float64
unique_customer_id             int64
order_channel                 object
coupon_code                   object
cust_first_ordered    datetime64[ns]
dtype: object
```

In [11]:
```
for i in sales.columns:
    print(sales[i].isna().mean())
```

```
0.0
0.0
0.0
0.0
0.0
```

Checking the data type and null values for each column in the 'Stores' dataset.

```
In [12]: stores = pd.read_csv('Stores.csv')
```

```
In [13]: stores
```

Out[13]:

	id	region_1	region_2	is_hub	retail_id
0	26	Scotland	Dundee	1.0	0.0
1	30	Scotland	Dundee	1.0	582.0
2	64	Scotland	Aberdeen	1.0	14.0
3	69	Scotland	Dundee	1.0	0.0
4	70	Scotland	Angus	1.0	0.0
...	...	...	...	...	...
1821	3179	Isle of Man	Isle of Man	1.0	NaN
1822	3191	England	Barnsley	1.0	NaN
1823	3196	Wales	Neath Port Talbot	1.0	NaN
1824	3197	England	Somerset	1.0	NaN
1825	3200	England	Wakefield	1.0	NaN

1826 rows × 5 columns

```
In [14]: stores.dtypes
```

```
Out[14]: id               int64
         region_1       object
         region_2       object
         is_hub        float64
         retail_id     float64
         dtype: object
```

```
In [15]: for i in stores.columns:
             print(stores[i].isna().mean(), i)

         0.0 id
         0.002190580503833516 region_1
         0.002190580503833516 region_2
         0.001095290251916758 is_hub
         0.009309967141292442 retail_id
```

```
In [16]: stores[stores['region_1'].isna()]
```

Out[16]:

	id	region_1	region_2	is_hub	retail_id
42	990	NaN	NaN	1.0	0.0
43	991	NaN	NaN	1.0	0.0
1807	3262	NaN	NaN	NaN	325.0
1808	3264	NaN	NaN	NaN	1420.0

Saving the store ids that corresponds to the null values in 'region_1' to a list. These id values in the list are then compared to the 'customer orders' dataset, to see if these stores are relevant to this analysis.

```
In [17]: l1 = stores[stores['region_1'].isna()]['id'].tolist()
```

```
In [18]: l1
```

```
Out[18]: [990, 991, 3262, 3264]
```

```
In [19]: for i in l1:
             print(sales[sales['store_id'] == i])

         Empty DataFrame
         Columns: [order_id, order_date, store_id, cost_total, unique_customer_id, order_channel, coupon_code, cust_first_ordered]
         Index: []
         Empty DataFrame
         Columns: [order_id, order_date, store_id, cost_total, unique_customer_id, order_channel, coupon_code, cust_first_ordered]
         Index: []
         Empty DataFrame
         Columns: [order_id, order_date, store_id, cost_total, unique_customer_id, order_channel, coupon_code, cust_first_ordered]
         Index: []
         Empty DataFrame
         Columns: [order_id, order_date, store_id, cost_total, unique_customer_id, order_channel, coupon_code, cust_first_ordered]
         Index: []
```

Since these store ids are not relavant to the analysis, they are discarded.

```
In [20]: stores = stores[stores['region_1'].notna()]
```

```
In [21]: for i in stores.columns:
             print(stores[i].isna().mean(), i)

         0.0 id
         0.0 region_1
         0.0 region_2
         0.0 is_hub
         0.009330406147091108 retail_id
```

Performing the same methods for the 'retail_id', to see if store ids that correspond to the null values in the 'retail_id' are relavant to the analysis.

```
In [22]: stores[stores['retail_id'].isna()]
```

Out[22]:

	id	region_1	region_2	is_hub	retail_id
1809	1154	England	Lincolnshire	1.0	NaN
1810	2722	England	Stockton-on-Tees	1.0	NaN
1811	2949	Scotland	South Lanarkshire	1.0	NaN
1812	3116	England	Walsall	1.0	NaN
1813	3117	England	Lincolnshire	1.0	NaN
1814	3125	England	Walsall	1.0	NaN
1815	3158	England	Leeds	1.0	NaN
1816	3159	England	Lancashire	1.0	NaN

1820	3171	England	Cambridgeshire	1.0	NaN
1821	3179	Isle of Man	Isle of Man	1.0	NaN
1822	3191	England	Barnsley	1.0	NaN
1823	3196	Wales	Neath Port Talbot	1.0	NaN
1824	3197	England	Somerset	1.0	NaN
1825	3200	England	Wakefield	1.0	NaN

In [23]:
```
l2 = stores[stores['retail_id'].isna()]['id'].tolist()
```

In [24]:
```
l2
```

Out[24]:
```
[1154,
 2722,
 2949,
 3116,
 3117,
 3125,
 3158,
 3159,
 3166,
 3168,
 3169,
 3171,
 3179,
 3191,
 3196,
 3197,
 3200]
```

In [25]:
```
for i in l2:
    print(sales[sales['store_id'] == i])
```

```
         order_id          order_date  store_id  cost_total  \
634395    6888590 2022-08-15 11:09:10      1154       24.54
638884    6861650 2022-08-13 08:48:55      1154       26.94
821856    7005771 2022-08-25 18:21:48      1154       43.96
821881    7009411 2022-08-26 09:40:06      1154       49.59
896061    6882284 2022-08-14 16:49:19      1154       16.13
896062    6974140 2022-08-22 16:49:10      1154       18.45
1114446   6904820 2022-08-16 16:32:34      1154       15.26
1304351   7034274 2022-08-28 09:06:19      1154       13.63
1357442   6831389 2022-08-10 15:42:53      1154       33.39
1357443   6920677 2022-08-18 11:47:47      1154       26.25
1357898   6921754 2022-08-18 13:15:39      1154       19.72
1357901   6921770 2022-08-18 13:15:53      1154       16.92
1368756   6893936 2022-08-15 16:58:58      1154       12.14
1368757   7006007 2022-08-25 18:33:43      1154       18.25

         unique_customer_id order_channel coupon_code  cust_first_ordered
634395               408214       android     WISHAW7 2022-08-15 11:09:10
638884               406883           ios      PB4QXX 2022-08-13 08:48:55
```

In [26]:
```
stores
```

Out[26]:

	id	region_1	region_2	is_hub	retail_id
0	26	Scotland	Dundee	1.0	0.0
1	30	Scotland	Dundee	1.0	582.0
2	64	Scotland	Aberdeen	1.0	14.0
3	69	Scotland	Dundee	1.0	0.0
4	70	Scotland	Angus	1.0	0.0
...	...	...	...	...	...
1821	3179	Isle of Man	Isle of Man	1.0	NaN
1822	3191	England	Barnsley	1.0	NaN
1823	3196	Wales	Neath Port Talbot	1.0	NaN
1824	3197	England	Somerset	1.0	NaN
1825	3200	England	Wakefield	1.0	NaN

1822 rows × 5 columns

Changing the 'is_hub' column to boolean data type and the 'retail_id' to Int64 data type.

In [27]:
```
stores.groupby('is_hub').count()
```

Out[27]:

	id	region_1	region_2	retail_id
is_hub				
0.0	474	474	474	474
1.0	1348	1348	1348	1331

In [28]:
```
stores['is_hub'] = stores['is_hub'].astype(bool)
```

In [29]:
```
stores['retail_id'] = stores['retail_id'].astype('Int64')
```

In [30]:
```
stores.dtypes
```

Out[30]:
```
id              int64
region_1       object
region_2       object
is_hub           bool
retail_id       Int64
dtype: object
```

In [31]:
```
for i in stores.columns:
    print(stores[i].isna().mean())
```

```
0.0
0.0
0.0
0.0
0.009330406147091108
```

Renaming the 'id' column in store dataset to 'store_id' and merging the store and sales_order dataset into one, on the column of store_id in both of the datasets.

In [33]: stores

Out[33]:

	store_id	region_1	region_2	is_hub	retail_id
0	26	Scotland	Dundee	True	0
1	30	Scotland	Dundee	True	582
2	64	Scotland	Aberdeen	True	14
3	69	Scotland	Dundee	True	0
4	70	Scotland	Angus	True	0
...	...	...	...	...	...
1821	3179	Isle of Man	Isle of Man	True	<NA>
1822	3191	England	Barnsley	True	<NA>
1823	3196	Wales	Neath Port Talbot	True	<NA>
1824	3197	England	Somerset	True	<NA>
1825	3200	England	Wakefield	True	<NA>

1822 rows × 5 columns

In [34]:

```
combined = pd.merge(sales,stores, on = 'store_id', how = 'left')
```

In [35]:

```
combined
```

Out[35]:

	order_id	order_date	store_id	cost_total	unique_customer_id	order_channel	coupon_code	cust_first_ordered	region_1	region_2	is_hub	retail_id
0	2625027	2021-05-27 12:01:02	787	11.19	10558	website	N/A	2019-10-16 14:30:22	Scotland	Dundee	True	10
1	5965587	2022-05-26 16:17:13	787	47.63	206	android	N/A	2016-10-13 23:03:25	Scotland	Dundee	True	10
2	5992815	2022-05-28 19:26:12	787	56.88	206	android	N/A	2016-10-13 23:03:25	Scotland	Dundee	True	10
3	2155767	2021-03-20 18:27:57	981	50.56	18306	ios	N/A	2020-03-22 15:54:10	Scotland	Fife	True	33
4	2501967	2021-05-08 21:22:44	208	52.58	18935	ios	N/A	2020-03-23 21:03:48	Scotland	Angus	True	309
...	...	...	...	...	...	...	...	...	...	...	...	...
1407995	6656119	2022-07-26 15:52:36	2910	39.98	388436	android	N/A	2022-07-07 18:00:45	Scotland	Highland	True	11
1407996	6917218	2022-08-17 19:25:30	2910	44.56	388436	android	N/A	2022-07-07 18:00:45	Scotland	Highland	True	11
1407997	6421571	2022-07-07 18:15:50	2850	9.58	388444	android	SNAP7	2022-07-07 18:15:50	England	Wirral	True	1077
1407998	6421856	2022-07-07 18:30:24	2692	36.82	388458	website	N/A	2022-07-07 18:30:24	Wales	Neath Port Talbot	True	940
1407999	6424396	2022-07-07 21:49:47	2197	15.08	388577	android	N/A	2022-07-07 21:49:47	Scotland	North Lanarkshire	True	560

1408000 rows × 12 columns

In [36]:

```
combined.dtypes
```

Out[36]:

```
order_id                       int64
order_date            datetime64[ns]
store_id                       int64
cost_total                   float64
unique_customer_id             int64
order_channel                 object
coupon_code                   object
cust_first_ordered    datetime64[ns]
region_1                      object
region_2                      object
is_hub                        object
retail_id                      Int64
dtype: object
```

In [37]:

```
for i in combined.columns:
    print(combined[i].isna().mean(), i)
```

```
0.0 order_id
0.0 order_date
0.0 store_id
0.0 cost_total
0.0 unique_customer_id
0.0 order_channel
0.0 coupon_code
0.0 cust_first_ordered
0.0005553977272727273 region_1
0.0005553977272727273 region_2
0.0005553977272727273 is_hub
0.0009069602272727273 retail_id
```

In [38]:

```
for i in stores.columns:
    print(stores[i].isna().mean(), i)
```

```
0.0 store_id
0.0 region_1
0.0 region_2
0.0 is_hub
0.009330406147091108 retail_id
```

In [39]:

```
combined[combined['is_hub'].isna()]
```

Out[39]:

	order_id	order_date	store_id	cost_total	unique_customer_id	order_channel	coupon_code	cust_first_ordered	region_1	region_2	is_hub	retail_id
4542	1687863	2021-01-11 11:48:28	1344	24.64	66527	ios	N/A	2020-07-30 20:07:05	NaN	NaN	NaN	<NA>
7007	2010302	2021-02-27 10:05:25	1344	14.78	93985	ios	N/A	2020-11-12 10:49:29	NaN	NaN	NaN	<NA>
15007	1707489	2021-01-14 14:47:34	1344	9.23	113947	android	N/A	2021-01-04 10:47:52	NaN	NaN	NaN	<NA>
18574	1756033	2021-01-22 09:30:53	1344	27.58	93300	website	N/A	2020-11-09 16:06:47	NaN	NaN	NaN	<NA>
18594	1664866	2021-01-07 15:05:44	1344	19.39	77713	android	N/A	2020-09-15 15:40:03	NaN	NaN	NaN	<NA>
...	...	...	...	...	...	...	...	...	...	...	...	...
1346918	7064123	2022-08-30 19:59:41	2876	18.91	218086	website	N/A	2021-07-27 18:32:55	NaN	NaN	NaN	<NA>
1357971	6923330	2022-08-18	2876	35.64	409542	android	N/A	2022-08-18	NaN	NaN	NaN	<NA>

1404784	6500505	2022-07-14 17:36:12	2876	24.49	391728	ios	N/A	2022-07-14 17:36:12	NaN	NaN	NaN	<NA>
1404785	6687138	2022-07-29 12:53:34	2876	27.07	391728	ios	N/A	2022-07-14 17:36:12	NaN	NaN	NaN	<NA>

782 rows × 12 columns

In [40]:
```
l3 = combined[combined['region_1'].isna()]['store_id'].tolist()
```

In [41]:
```
l3
```

Out[41]:
```
[1344,
 1344,
 1344,
 1344,
 1344,
 1344,
 1546,
 1025,
 1344,
 1344,
 1344,
 1344,
 1344,
 1344,
 1344,
 1344,
 1344,
 1344,
 1548,
```

In [42]:
```
for i in l3:
    print(stores[stores['store_id'] == i])
```

```
Empty DataFrame
Columns: [store_id, region_1, region_2, is_hub, retail_id]
Index: []
Empty DataFrame
Columns: [store_id, region_1, region_2, is_hub, retail_id]
Index: []
Empty DataFrame
Columns: [store_id, region_1, region_2, is_hub, retail_id]
Index: []
Empty DataFrame
Columns: [store_id, region_1, region_2, is_hub, retail_id]
Index: []
Empty DataFrame
Columns: [store_id, region_1, region_2, is_hub, retail_id]
Index: []
Empty DataFrame
Columns: [store_id, region_1, region_2, is_hub, retail_id]
Index: []
Empty DataFrame
```

In [43]:
```
combined
```

Out[43]:

	order_id	order_date	store_id	cost_total	unique_customer_id	order_channel	coupon_code	cust_first_ordered	region_1	region_2	is_hub	retail_id
0	2625027	2021-05-27 12:01:02	787	11.19	10558	website	N/A	2019-10-16 14:30:22	Scotland	Dundee	True	10
1	5965587	2022-05-26 16:17:13	787	47.63	206	android	N/A	2016-10-13 23:03:25	Scotland	Dundee	True	10
2	5992815	2022-05-28 19:26:12	787	56.88	206	android	N/A	2016-10-13 23:03:25	Scotland	Dundee	True	10
3	2155767	2021-03-20 18:27:57	981	50.56	18306	ios	N/A	2020-03-22 15:54:10	Scotland	Fife	True	33
4	2501967	2021-05-08 21:22:44	208	52.58	18935	ios	N/A	2020-03-23 21:03:48	Scotland	Angus	True	309
...	...	...	...	...	...	...	...	...	...	...	...	...
1407995	6656119	2022-07-26 15:52:36	2910	39.98	388436	android	N/A	2022-07-07 18:00:45	Scotland	Highland	True	11
1407996	6917218	2022-08-17 19:25:30	2910	44.56	388436	android	N/A	2022-07-07 18:00:45	Scotland	Highland	True	11
1407997	6421571	2022-07-07 18:15:50	2850	9.58	388444	android	SNAP7	2022-07-07 18:15:50	England	Wirral	True	1077
1407998	6421856	2022-07-07 18:30:24	2692	36.82	388458	website	N/A	2022-07-07 18:30:24	Wales	Neath Port Talbot	True	940
1407999	6424396	2022-07-07 21:49:47	2197	15.08	388577	android	N/A	2022-07-07 21:49:47	Scotland	North Lanarkshire	True	560

1408000 rows × 12 columns

In [44]:
```
combined.rename(columns={'order_id':'Order_ID','order_date':'Order_Date','store_id':'Store_ID','cost_total':'Order_Value', 'uniqu
```

In [45]:
```
combined
```

Out[45]:

	Order_ID	Order_Date	Store_ID	Order_Value	Customer_ID	Order_Channel	Coupon_Code	First_Order_Date	Region	City	Is_Hub	retail_id
0	2625027	2021-05-27 12:01:02	787	11.19	10558	website	N/A	2019-10-16 14:30:22	Scotland	Dundee	True	10
1	5965587	2022-05-26 16:17:13	787	47.63	206	android	N/A	2016-10-13 23:03:25	Scotland	Dundee	True	10
2	5992815	2022-05-28 19:26:12	787	56.88	206	android	N/A	2016-10-13 23:03:25	Scotland	Dundee	True	10
3	2155767	2021-03-20 18:27:57	981	50.56	18306	ios	N/A	2020-03-22 15:54:10	Scotland	Fife	True	33
4	2501967	2021-05-08 21:22:44	208	52.58	18935	ios	N/A	2020-03-23 21:03:48	Scotland	Angus	True	309
...	...	...	...	...	...	...	...	...	...	...	...	...
1407995	6656119	2022-07-26 15:52:36	2910	39.98	388436	android	N/A	2022-07-07 18:00:45	Scotland	Highland	True	11
1407996	6917218	2022-08-17 19:25:30	2910	44.56	388436	android	N/A	2022-07-07 18:00:45	Scotland	Highland	True	11
1407997	6421571	2022-07-07 18:15:50	2850	9.58	388444	android	SNAP7	2022-07-07 18:15:50	England	Wirral	True	1077
1407998	6421856	2022-07-07 18:30:24	2692	36.82	388458	website	N/A	2022-07-07 18:30:24	Wales	Neath Port Talbot	True	940
1407999	6424396	2022-07-07 21:49:47	2197	15.08	388577	android	N/A	2022-07-07 21:49:47	Scotland	North Lanarkshire	True	560

```
In [ ]: combined.to_csv(r'C:\Users\DELL\Desktop\Interview Doc\final_for_pres.csv', index = False)
```