ETW2001 Assignment 2

Sia Yi Bin

33363129

Section A

Question 1

After inner joining the datasets, I have access to all the attributes of both the orders and their corresponding items in a single dataset. This allows me performing comprehensive analysis of orders, including details such as customer information, order status, and delivery dates, alongside specific item details like product, price, and seller.

```
> glimpse(merged_dataset)
Rows: 112,650
Columns: 14
$ order_id                      <chr> "e481f51cbdc54678b7cc49136f2d6af7", "53cdb2fc8bc7dce0b6741e2…
$ customer_id                   <chr> "9ef432eb6251297304e76186b10a928d", "b0830fb4747a6c6d20dea0b…
$ order_status                  <chr> "delivered", "delivered", "delivered", "delivered", "deliver…
$ order_purchase_timestamp      <chr> "2017-10-02 10:56:33", "2018-07-24 20:41:37", "2018-08-08 08…
$ order_approved_at             <chr> "2017-10-02 11:07:15", "2018-07-26 03:24:27", "2018-08-08 08…
$ order_delivered_carrier_date  <chr> "2017-10-04 19:55:00", "2018-07-26 14:31:00", "2018-08-08 13…
$ order_delivered_customer_date <chr> "2017-10-10 21:25:13", "2018-08-07 15:27:45", "2018-08-17 18…
$ order_estimated_delivery_date <chr> "2017-10-18 00:00:00", "2018-08-13 00:00:00", "2018-09-04 00…
$ order_item_id                 <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, …
$ product_id                    <chr> "87285b34884572647811a353c7ac498a", "595fac2a385ac33a80bd511…
$ seller_id                     <chr> "3504c0cb71d7fa48d967e0e4c94d59d9", "289cdb325fb7e7f891c3860…
$ shipping_limit_date           <chr> "2017-10-06 11:07:15", "2018-07-30 03:24:27", "2018-08-13 08…
$ price                         <dbl> 29.99, 118.70, 159.90, 45.00, 19.90, 147.90, 49.90, 59.99, 1…
$ freight_value                 <dbl> 8.72, 22.76, 19.22, 27.20, 8.72, 27.36, 16.05, 15.17, 16.05,…
```

Question 2

Perform a left join between olist_order_dataset and olist_order_reviews_dataset ensures that all orders from the olist_orders_dataset are retained, but only those with matching reviews from the olist_order_reviews_dataset are merged. The merged dataset allows us to identify orders that have not received any reviews, which is 768 orders don't have any reviews.

```
> orders_without_reviews
  orders_without_reviews
1                    768
```

Question 3

Perform a right join will retain all rows from olist_products_dataset, matching them with rows from olist_order_items_dataset based on product_id. The result showing there are 0 unmatched product id from the right dataset, it means that every product has been sold at least once.

```
> unsold_products
  unsold_products
1               0
```

Question 4

Perform a full join will retain all rows from both datasets, matching them based on the order_id, which ensures that every customer and every order will be included in the resulting dataset. By filtering the merged dataset to include only rows where the order_id column is missing, finding customers without orders. Similarly, by filtering the merged dataset to include only rows where the customer_id column is missing, is used to find orders without corresponding customer details. But both the result shown 0 rows, which means there is no customers without orders or orders without customers detail in the dataset given.

```
> customers_without_orders
 [1] customer_id                 customer_unique_id            customer_zip_code_prefix
 [4] customer_city               customer_state                order_id
 [7] order_status                order_purchase_timestamp      order_approved_at
[10] order_delivered_carrier_date order_delivered_customer_date order_estimated_delivery_date
<0 rows> (or 0-length row.names)
> orders_without_customer_details
 [1] customer_id                 customer_unique_id            customer_zip_code_prefix
 [4] customer_city               customer_state                order_id
 [7] order_status                order_purchase_timestamp      order_approved_at
[10] order_delivered_carrier_date order_delivered_customer_date order_estimated_delivery_date
<0 rows> (or 0-length row.names)
```

Question 5

Perform a semi join in this case will returns only the rows from olist_sellers_dataset where there is a match with the olist_order_items_dataset based on seller_id column. I first determine the total number of unique sellers in each dataset by counting the number of unique seller_id values. Then find the number of common sellers between the datasets by merging them based on the seller_id column and counting the number of unique sellers. After comparing the common sellers and total olist sellers, I found that they are 100% similar, which means that there is no seller being inactive, all the sellers have at least made 1 sales.

```
> similarity_percentage
[1] 100
```

Question 6

Perform an anti-join will compares two datasets based on customer_id and returns rows from the olist_customers_dataset that do not have matching keys in olist_orders_dataset. The result shown that the matched row is 0, means every customer ID from the olist_customers_dataset has a corresponding entry in the olist_orders_dataset, so there is no customer haven't placed an order.

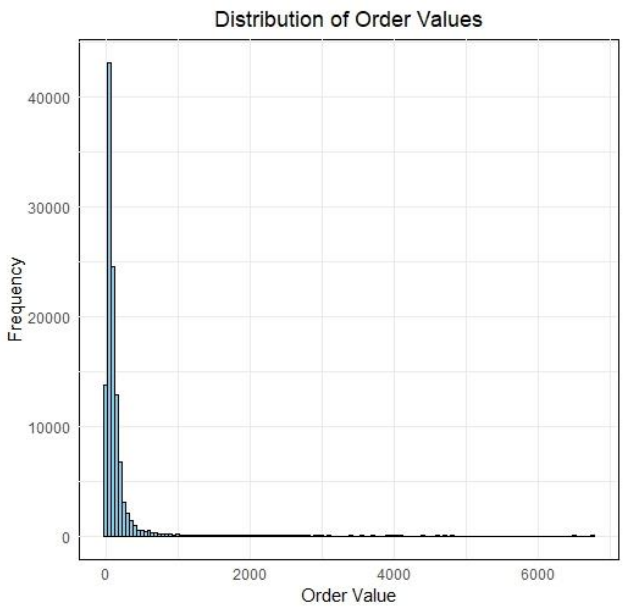
```
> customers_without_orders
[1] customer_id              customer_unique_id       customer_zip_code_prefix
[4] customer_city            customer_state
<0 rows> (or 0-length row.names)
```

Question 7

I first use an inner join to connect the olist_orders_dataset with the olist_order_items_dataset based on the common key order_id, which joins each order with its corresponding order items. Then use left join with olist_products_dataset to adds product details to each order item. Finally, I use right join with olist_sellers_dataset to add seller details to each order. By analysing the comprehensive dataset, I found that the flow between seller and product is there is 11.13 products sold per seller, which means that on average, one seller sells about 11 different products. The flow between products and order, is there is 1.04 products have been placed within an order, which indicates that the buyer usually only buy a single products per order, we can also observe this on the plot shown below, the order value at 1 having the highest frequency.

```
Average number of products sold per seller: 11.13021

Average number of products per order: 1.038098
```



Section B

Question 1-4

Chart generated in R studio.

Question 5

The bar chart depicts the global market trend. The United States leads the way with a staggering 1,000 sales, demonstrating its strong market presence and high level of consumption. In Europe, countries such as France, Germany, Spain, and the United Kingdom also have large consumer markets, indicating that these countries have stable economies and sizable consumer numbers. Japan and Singapore are the only two countries in the Asia-Pacific region that performed better, with sales lower than those of the countries, possibly due to their smaller populations. Conversely, countries such as Ireland and the Philippines have lower sales, possibly due to smaller populations or limited market coverage. However, countries such as Austria, Belgium, Denmark, and Norway are promising countries where companies can explore emerging markets and expand their sales channels.