E-commerce:
Brazilian (Olist) Orders Dataset

- Bernard Bernabeo
- King Eluard Camota
- Patrick Salazar
- Trizzia Ellaine Singson

Brazilian E-Commerce Dataset by Olist

- This **unstructured** dataset focuses on the **Brazilian E-Commerce Public Dataset by Olist**, from Kaggle, which provides information about approximately 100k+ orders placed from 2016 to 2018.
- It is made to provide various aspects of the customer purchasing experience, from order placement to fulfillment and post-delivery reviews.

olist_orders_dataset		
ukey	order_id	object
fkey	customer_id	object
	order_status	object
	order_purchase_timestamp	dtime
	order_approved_at	dtime
	order_delivered_carrier_date	dtime
	order_delivered_customer_date	dtime
	order_estimated_delivery_date	dtime

olist_order_items_dataset		
fkey	product_id	object
	order_item_id	int
fkey	order_id	object
fkey	seller_id	object
	shipping_limit_date	dtime
	price	float
	freight_value	float

olist_order_reviews_dataset		
ukey	review_id	object
fkey	order_id	object
	review_score	int
	review_comment_title	object
	review_comment_message	object
	review_creation_date	dtime
	review_answer_timestamp	dtime

olist_order_payments_dataset		
fkey	order_id	object
	payment_sequential	int
	payment_type	object
	payment_installments	int
	payment_value	float

olist_products_dataset		
ukey	product_id	object
	product_category_name	object
	product_name_lenght	int
	product_description_lenght	int
	product_photos_qty	int
	product_weight_g	int
	product_length_cm	int
	product_height_cm	int
	product_width_cm	int

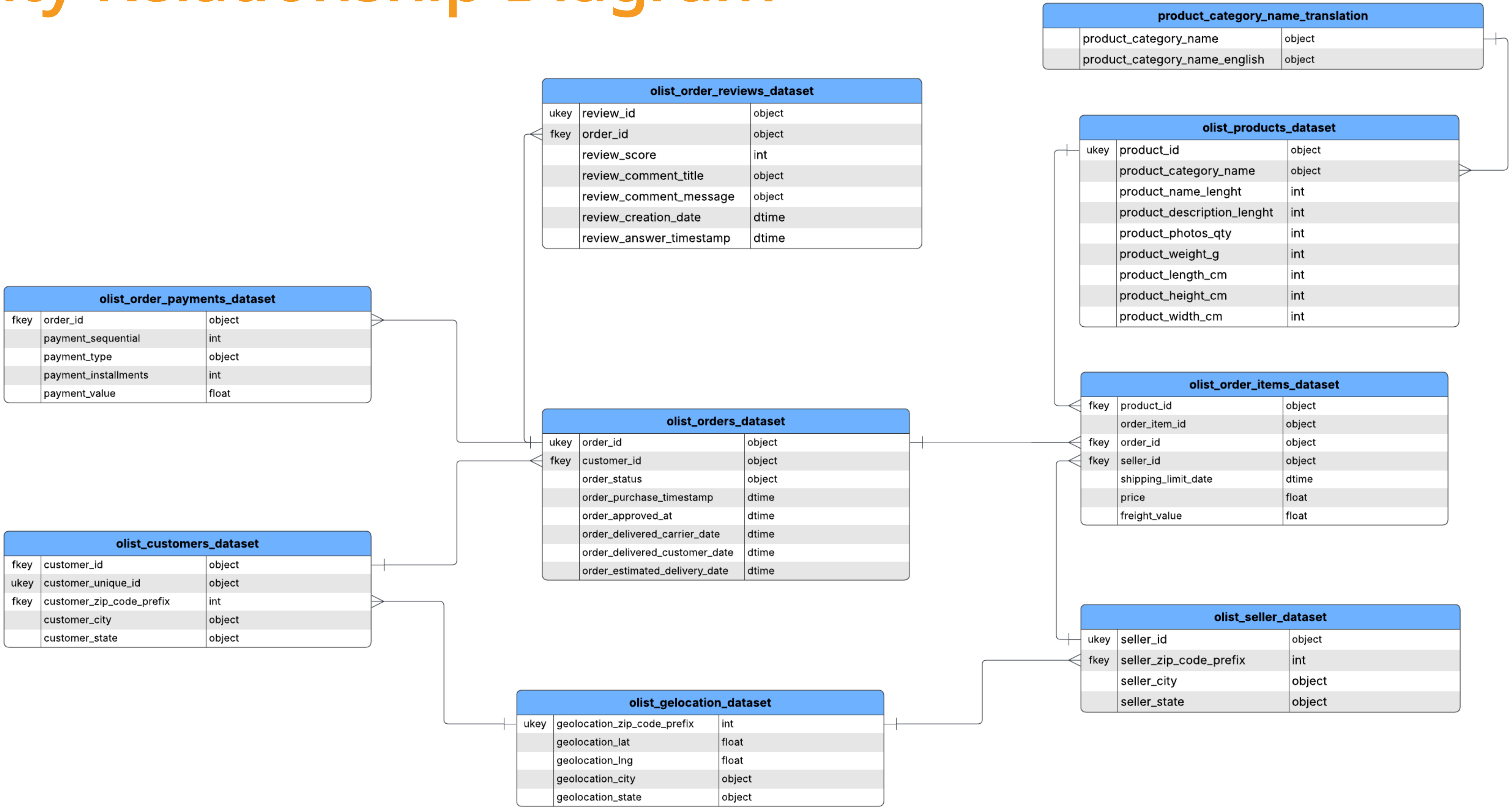
olist_customers_dataset		
fkey	customer_id	object
ukey	customer_unique_id	object
fkey	customer_zip_code_prefix	int
	customer_city	object
	customer_state	object

olist_seller_dataset		
ukey	seller_id	object
fkey	seller_zip_code_prefix	int
	seller_city	object
	seller_state	object

olist_gelocation_dataset		
ukey	geolocation_zip_code_prefix	int
	geolocation_lat	float
	geolocation_lng	float
	geolocation_city	object
	geolocation_state	object

product_category_name_translation		
	product_category_name	object
	product_category_name_english	object

Entity Relationship Diagram



Objectives

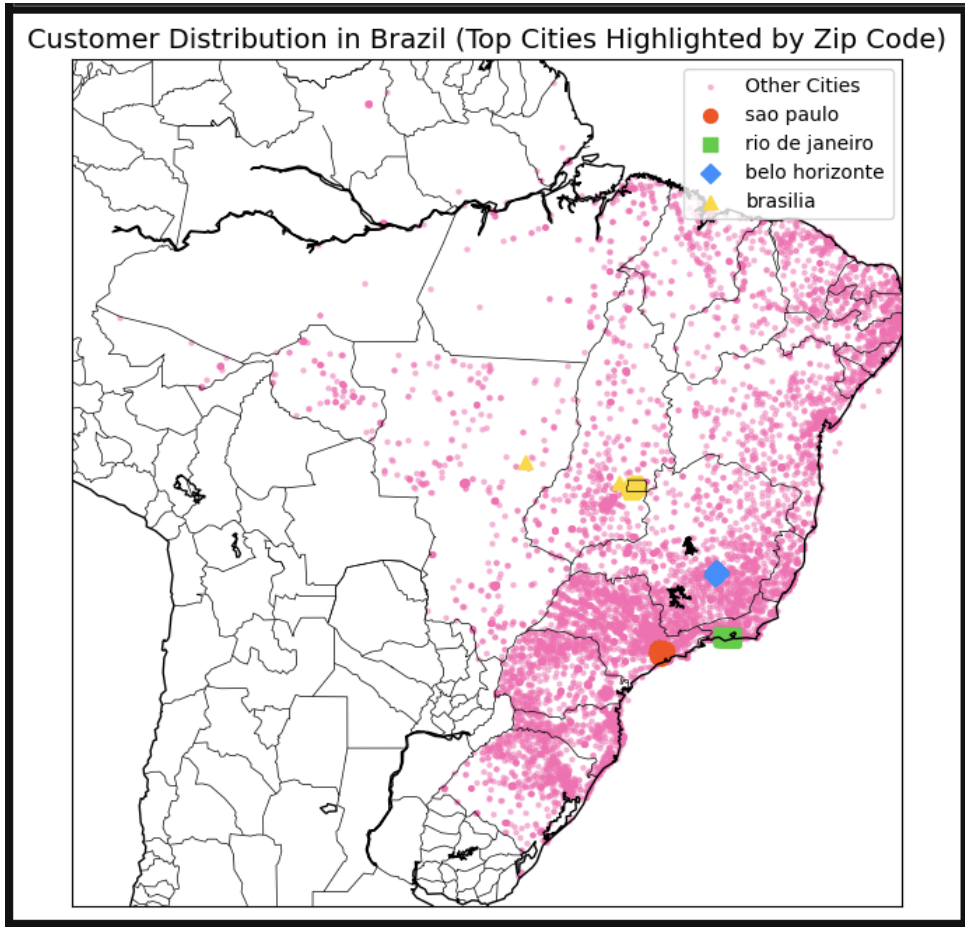
Identify customer distribution and sales performance by location

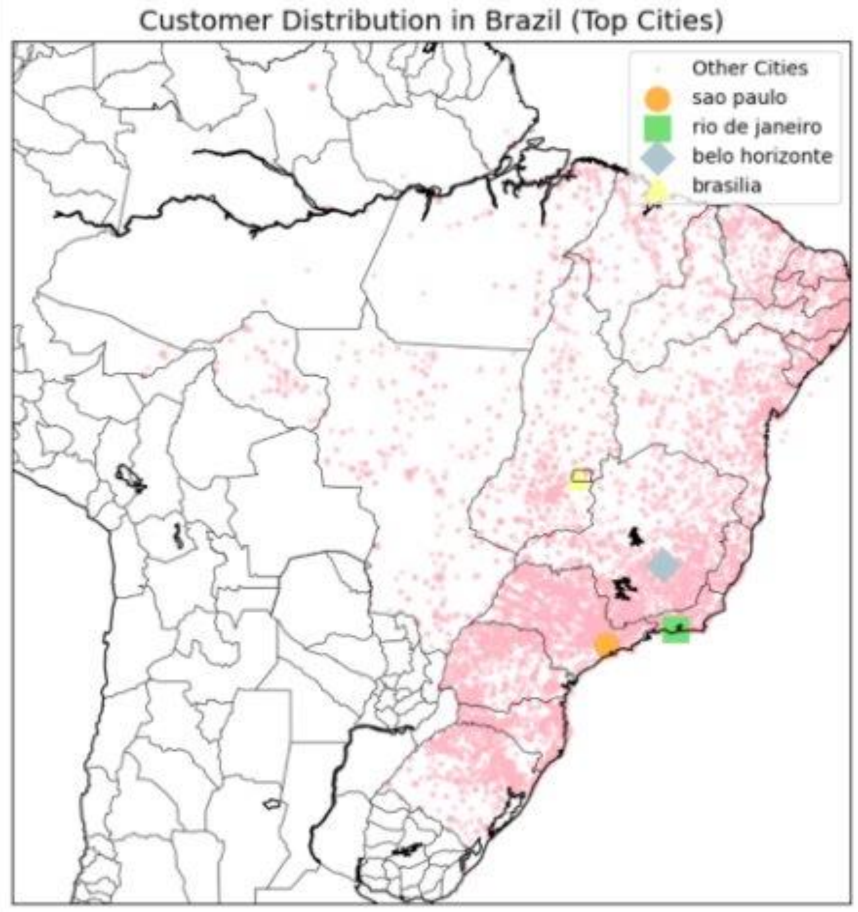
Analyze performance of products by categories and revenue

How Have Sales and Orders Changed Over Time?

Develop a model that predicts a more robust estimation of delivery date

Objective 1: Identify customer distribution and sales performance by location

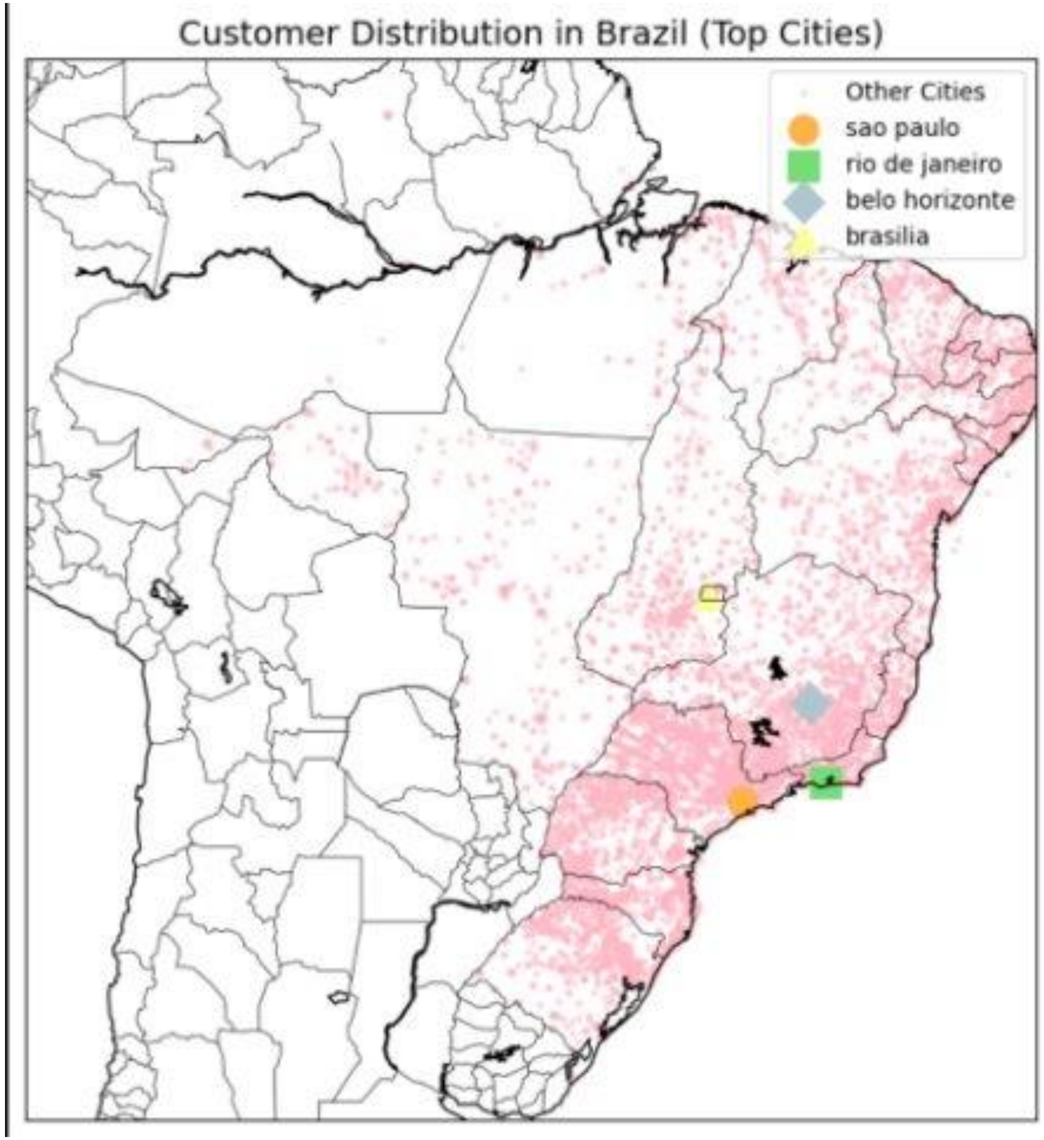




- **Scattered Zip Codes**
- **Lack of Data Grouping**
- **Outlier Points Included**
- **Inconsistent Representation**

Customer Distribution

Objective 1: Identify customer distribution and sales performance by location



Inconsistency	Solution (How It Was Fixed)
Scattered Zip Codes:	Grouping by zip_code_prefix: Averaged customer locations by zip codes, reducing noise and merging duplicated points.
- Cities like brasilia had hundreds of zip codes scattered over a wide area.	- Grouping points by zip codes provided a cleaner visualization.
Lack of Data Grouping:	Central Point Calculation: Averaged all zip code entries per city to produce a single reliable location.
- Multiple zip codes per city caused overlapping points.	- Creating a central point eliminated scattered points, especially for brasilia.
Outlier Points Included:	Outlier Handling: Averaging entries reduced the impact of irrelevant or incorrect points.
- Incorrect or miscategorized zip codes contributed to cluttered visuals.	- Averaging provided a consistent, accurate representation of each city.
Inconsistent Representation:	Uniform Representation: Using central points ensured fair comparison between cities.

Customer Distribution

Objective 1: Identify customer distribution and sales performance by location

Cleaning (How Outliers Were Handled):

Cleaning Step	What We Did	Why This Was Necessary
Group by zip_code_prefix	Averaged customer points per zip_code_prefix.	Reduced noise and clutter from raw data points.
Calculate Central Points	Consolidated entries to a single point per city.	Improved consistency and accuracy of city representation.
Visual Clarity Enhancement	Used pastel colors for better visibility.	Made comparison of top cities easier and more effective.

Objective 1: Identify customer distribution and sales performance by location

Handling Missing Values:

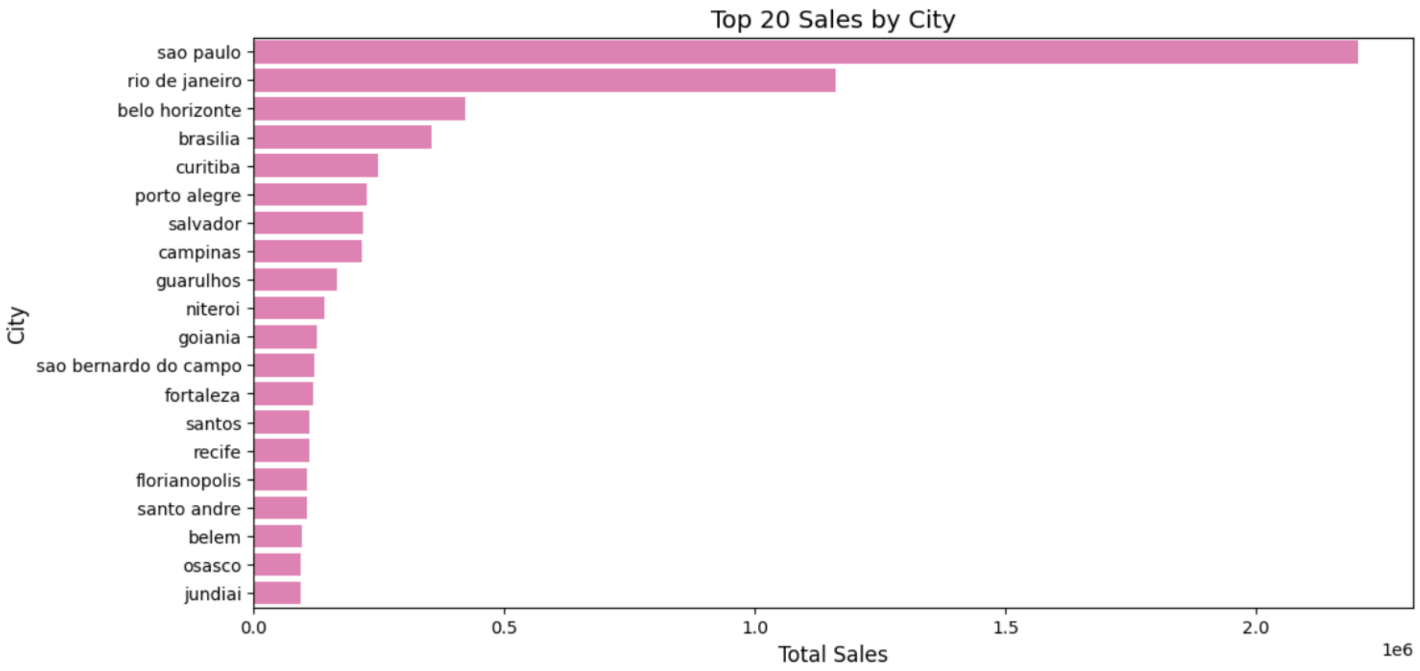
- Dropped rows with missing entries in critical columns (geolocation_lat, geolocation_lng, geolocation_zip_code_prefix, customer_id, payment_value).

Data Transformation:

- Merged datasets (olist_customers, olist_orders, olist_order_payments) to prepare sales_by_location DataFrame.
- Aggregated sales data per city to provide a reliable summary.

Outlier Handling:

- Removed cities with unusually low sales values (payment_value < 1000) - threshold set.

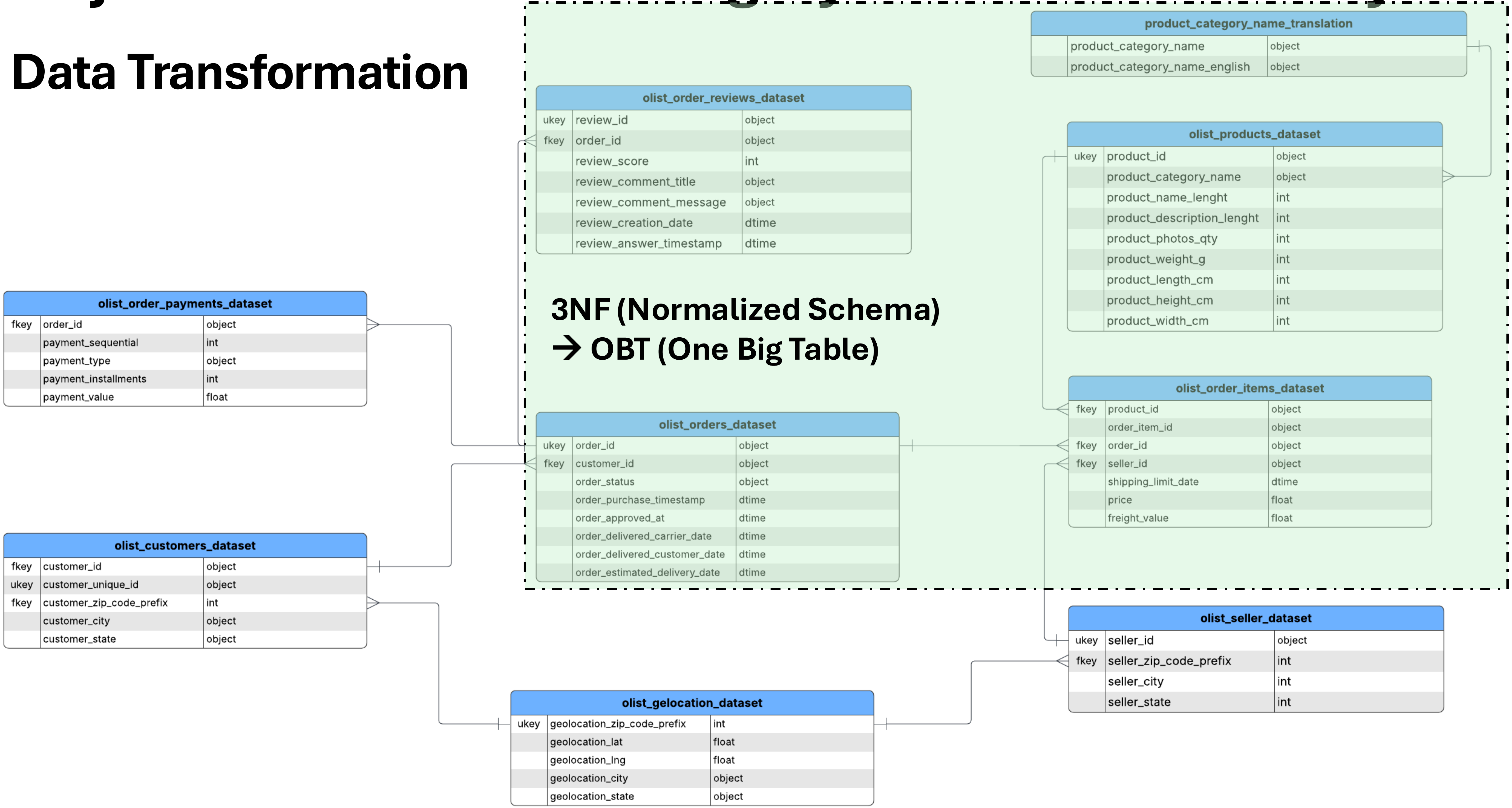


Sales Performance

Olist Product Category Performance Analysis

Objective 2: Product Category Performance Analysis

Data Transformation



Objective 2: Product Category Performance Analysis

Data Transformation

	order_id	price	product_category_name	generalized_product_category	review_score	order_status
0	00010242fe8c5a6d1ba2dd792cb16214	58.90	cool_stuff	Travel & Lifestyle	5.0	delivered
1	00018f77f2f0320c557190d7a144bdd3	239.90	pet_shop	Baby & Pet Supplies	4.0	delivered
2	000229ec398224ef6ca0657da4fc703e	199.00	moveis_decoracao	Home & Living	5.0	delivered
3	00024acbcdf0a6daa1e931b038114c75	12.99	perfumaria	Health & Beauty	4.0	delivered
4	00042b26cf59d7ce69dfabb4e55b4fd9	199.90	ferramentas_jardim	Home & Living	5.0	delivered
...	...	...	...	...	...	...
113309	fffc94f6ce00a00581880bf54a75a037	299.99	utilidades_domesticas	Home & Living	5.0	delivered
113310	fffcd46ef2263f404302a634eb57f7eb	350.00	informatica_acessorios	Electronics & Technology	5.0	delivered
113311	fffce4705a9662cd70adb13d4a31832d	99.90	esporte_lazer	Sports, Leisure & Hobbies	5.0	delivered
113312	fffe18544ffabc95dfada21779c9644f	55.99	informatica_acessorios	Electronics & Technology	5.0	delivered
113313	fffe41c64501cc87c801fd61db3f6244	43.00	cama_mesa_banho	Home & Living	5.0	delivered

110840 rows × 6 columns

Product Category Performance Analysis

Data Transformation

- Generalizing product categories

```
Number of unique product categories: 71
Unique product categories: ['health_beauty' 'computers_accessories' 'auto' 'bed_bath_table'
 'furniture_decor' 'sports_leisure' 'perfumery' 'housewares' 'telephony'
 'watches_gifts' 'food_drink' 'baby' 'stationery' 'tablets_printing_image'
 'toys' 'fixed_telephony' 'garden_tools' 'fashion_bags_accessories'
 'small_appliances' 'consoles_games' 'audio' 'fashion_shoes' 'cool_stuff'
 'luggage_accessories' 'air_conditioning'
 'construction_tools_construction'
 'kitchen_dining_laundry_garden_furniture' 'costruction_tools_garden'
 'fashion_male_clothing' 'pet_shop' 'office_furniture' 'market_place'
 'electronics' 'home_appliances' 'party_supplies' 'home_confort'
 'costruction_tools_tools' 'agro_industry_and_commerce'
 'furniture_mattress_and_upholstery' 'books_technical' 'home_construction'
 'musical_instruments' 'furniture_living_room' 'construction_tools_lights'
 'industry_commerce_and_business' 'food' 'art' 'furniture_bedroom'
 'books_general_interest' 'construction_tools_safety'
 'fashion_underwear_beach' 'fashion_sport' 'signaling_and_security'
 'computers' 'christmas_supplies' 'fashio_female_clothing'
 'home_appliances_2' 'books_imported' 'drinks' 'cine_photo' 'la_cuisine'
 'music' 'home_comfort_2' 'small_appliances_home_oven_and_coffee'
 'cds_dvds_musicals' 'dvds_blu_ray' 'flowers' 'arts_and_craftmanship'
 'diapers_and_hygiene' 'fashion_childrens_clothes' 'security_and_services']
Category counts (original):
product_category_name_english
health_beauty                1
food                         1
fashion_sport                1
fashion_underwear_beach      1
construction_tools_safety    1
                            ..
luggage_accessories          1
cool_stuff                   1
fashion_shoes                1
audio                        1
security_and_services        1
Name: count, Length: 71, dtype: int64
```

Product Categories (from Original dataset)

```
Number of unique generalized categories: 13
Unique generalized categories: ['Health & Beauty' 'Electronics & Technology' 'Uncategorized'
 'Home & Living' 'Sports, Leisure & Hobbies' 'Fashion & Accessories'
 'Food & Beverages' 'Baby & Pet Supplies' 'Travel & Lifestyle'
 'Construction & Tools' 'Industry & Business' 'Party & Seasonal Supplies'
 'Security & Services']
Category counts:
generalized_product_category
Home & Living                15
Sports, Leisure & Hobbies    11
Electronics & Technology      9
Uncategorized                 7
Fashion & Accessories         7
Construction & Tools          4
Health & Beauty               3
Food & Beverages              3
Industry & Business           3
Party & Seasonal Supplies     3
Baby & Pet Supplies           2
Travel & Lifestyle            2
Security & Services           2
Name: count, dtype: int64
```

Generalized Product Categories

Product Category Performance Analysis

Data Cleaning

Cleaning Step	What We Did	Justification
Filter data	Only select "delivered" orders in `order_status` column	Only purchases with “delivered” status were considered as successful transaction
Dropping Null Values	Drop null values on columns `review_score` and ‘generalized_product_category	Null value means no product rating from the customer. Keeping null values will affect the review score for a specific product. Untraceable product category.

Product Category Performance Analysis

Feature Engineering

olist_products_dataset		
ukey	product_id	object
	product_category_name	object
	product_name_lenght	int
	product_description_lenght	int
	product_photos_qty	int
	product_weight_g	int
	product_length_cm	int
	product_height_cm	int
	product_width_cm	int

product_category_name
cool_stuff
pet_shop
moveis_decoracao
perfumaria
ferramentas_jardim
...
utilidades_domesticas
informatica_acessorios
esporte_lazer

product_category_name_translation		
	product_category_name	object
	product_category_name_english	object

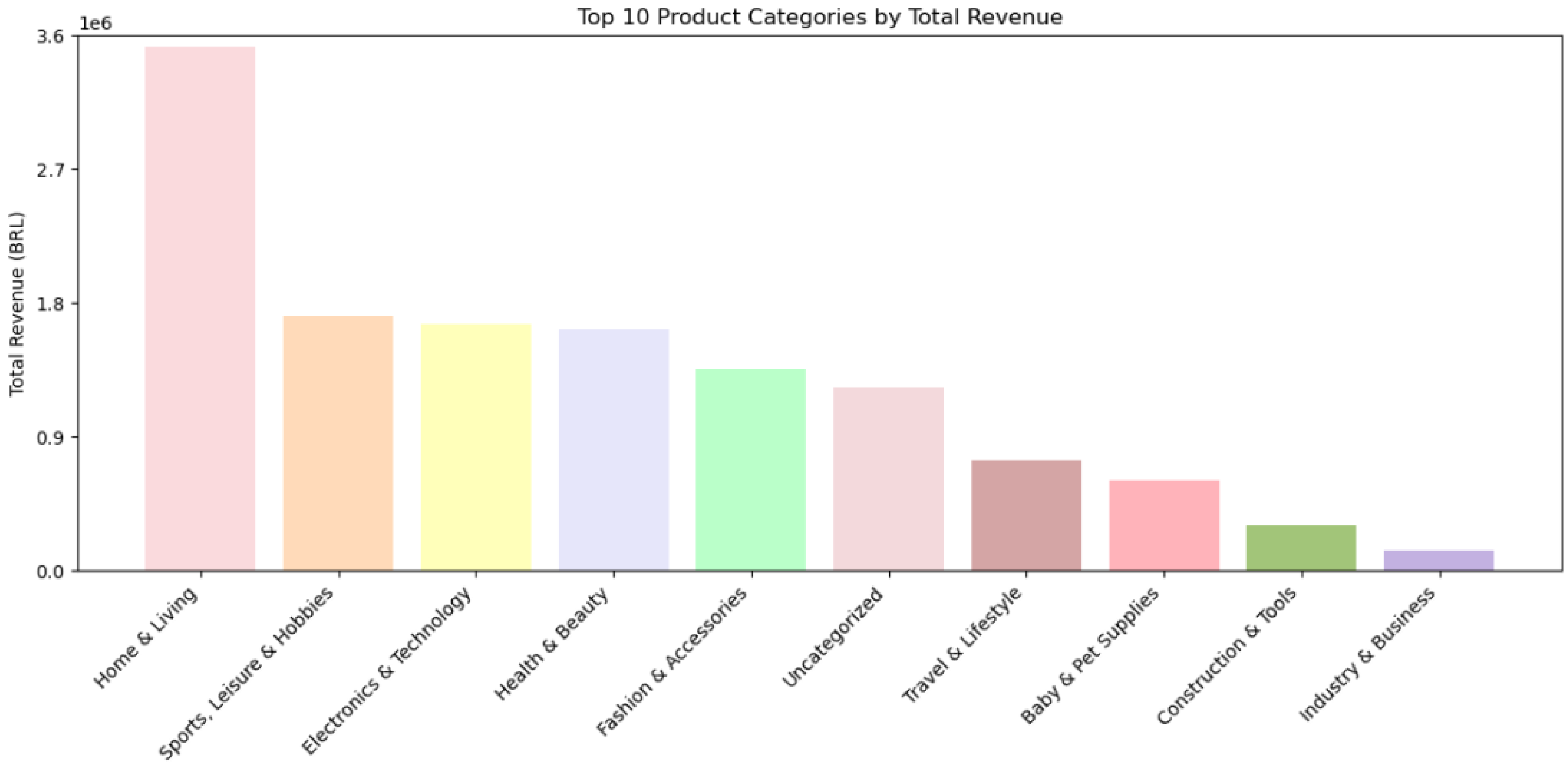
product_category_name_english
cool_stuff
pet_shop
furniture_decor
perfumery
garden_tools
...
housewares
computers_accessories
sports_leisure

product_category_name_english	generalized_product_category
cool_stuff	Travel & Lifestyle
pet_shop	Baby & Pet Supplies
furniture_decor	Home & Living
perfumery	Health & Beauty
garden_tools	Home & Living
...	...
housewares	Home & Living
computers_accessories	Electronics & Technology
sports_leisure	Sports, Leisure & Hobbies

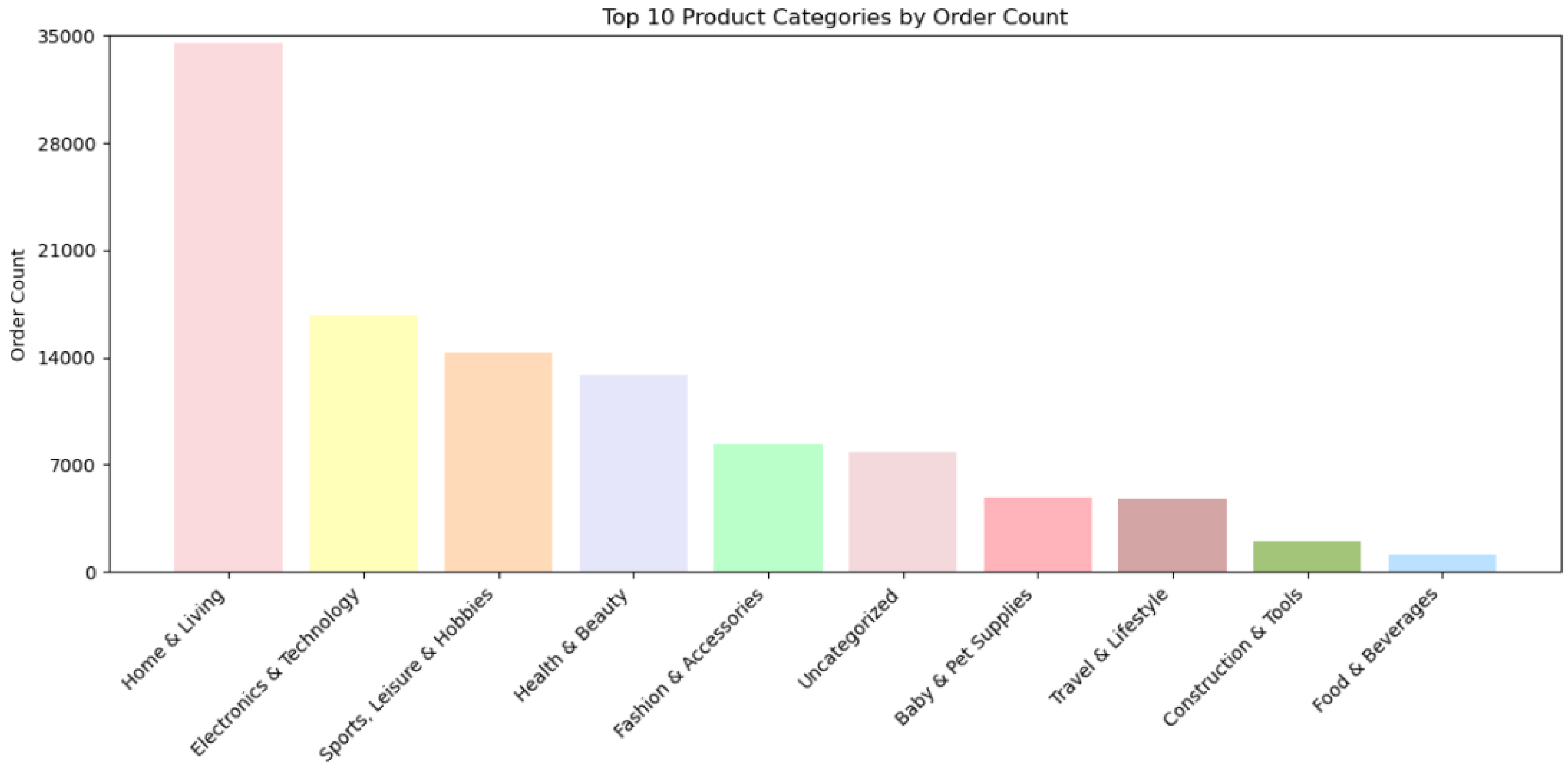
Before Feature Engineering

After Feature Engineering

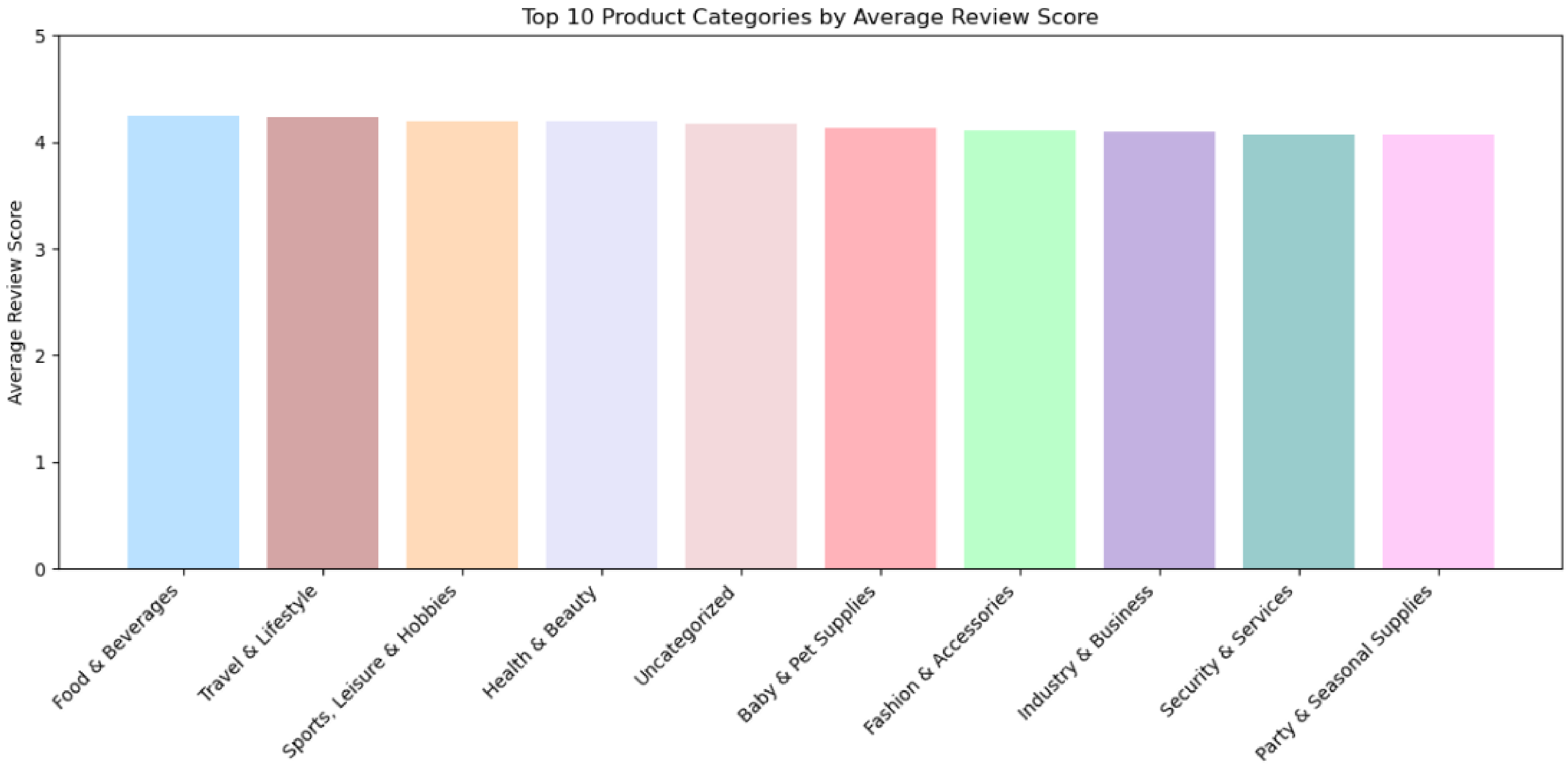
Product Category Performance Analysis



Product Category Performance Analysis

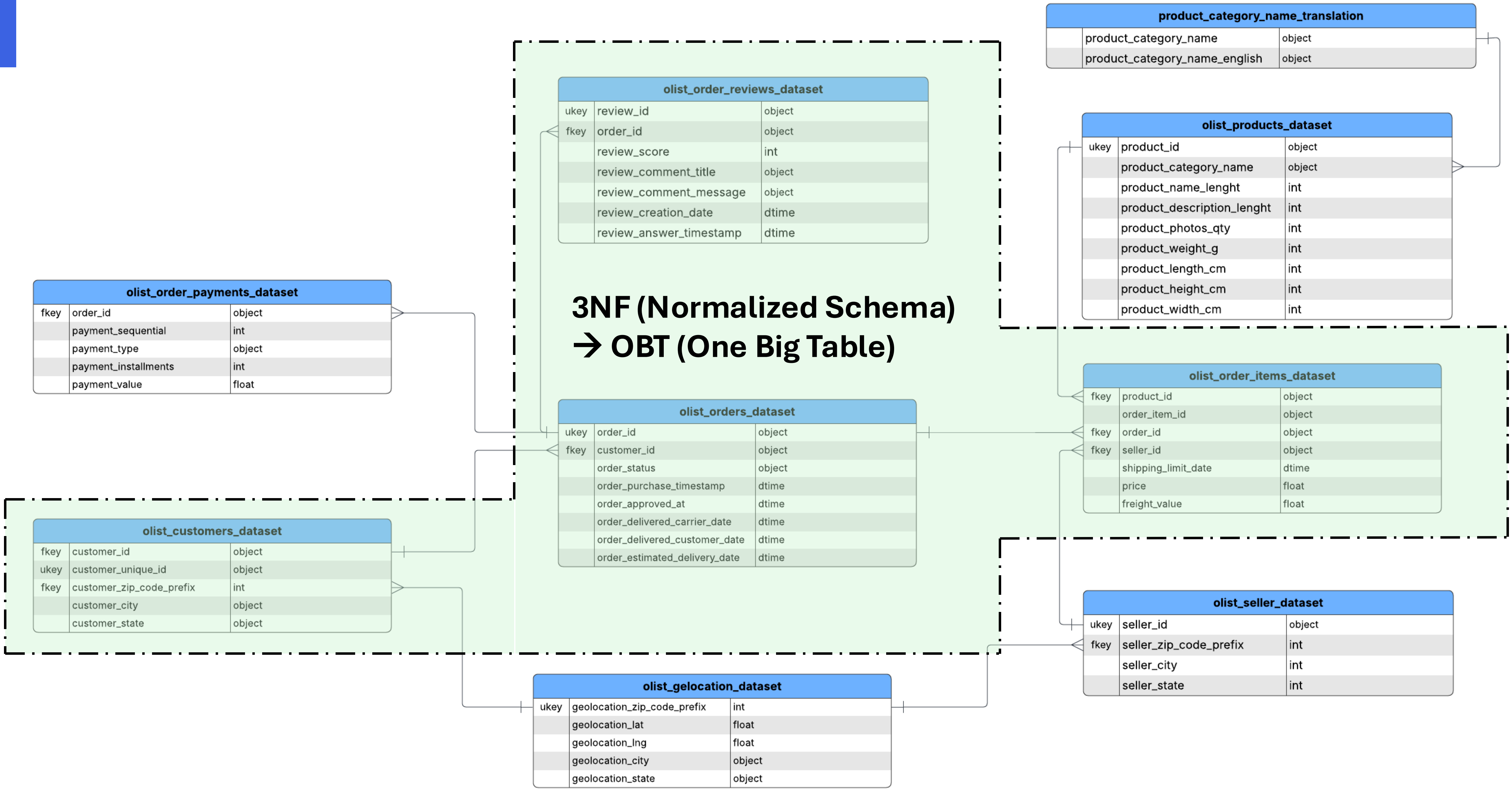


Product Category Performance Analysis



How Have Sales and Orders Changed Over Time?

Objective 3: How Have Sales and Orders Changed Over Time?



Objective 3: How Have Sales and Orders Changed Over Time?

Dataset	Missing Values	Duplicated Rows
reviews	145903	0
orders	4908	0
order_items	0	0
customers	0	0

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99224 entries, 0 to 99223
Data columns (total 7 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   review_id                99224 non-null  object
 1   order_id                 99224 non-null  object
 2   review_score             99224 non-null  int64
 3   review_comment_title     11568 non-null  object
 4   review_comment_message   40977 non-null  object
 5   review_creation_date     99224 non-null  datetime64[ns]
 6   review_answer_timestamp  99224 non-null  object
dtypes: datetime64[ns](1), int64(1), object(5)
memory usage: 5.3+ MB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99441 entries, 0 to 99440
Data columns (total 8 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   order_id                       99441 non-null  object
 1   customer_id                    99441 non-null  object
 2   order_status                   99441 non-null  object
 3   order_purchase_timestamp       99441 non-null  datetime64[ns]
 4   order_approved_at              99281 non-null  object
 5   order_delivered_carrier_date   97658 non-null  object
 6   order_delivered_customer_date  96476 non-null  object
 7   order_estimated_delivery_date  99441 non-null  object
dtypes: datetime64[ns](1), object(7)
memory usage: 6.1+ MB
```

Data Cleaning

Dataset	Missing Values	Duplicated Rows
reviews	145903	0
orders	4908	0
order_items	0	0
customers	0	0

```
 6   review_answer_timestamp  99224 non-null  object
dtypes: datetime64[ns](1), int64(1), object(5)
memory usage: 5.3+ MB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99441 entries, 0 to 99440
Data columns (total 8 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   order_id                       99441 non-null  object
 1   customer_id                    99441 non-null  object
 2   order_status                   99441 non-null  object
 3   order_purchase_timestamp       99441 non-null  datetime64[ns]
 4   order_approved_at              99281 non-null  object
 5   order_delivered_carrier_date   97658 non-null  object
 6   order_delivered_customer_date  96476 non-null  object
 7   order_estimated_delivery_date  99441 non-null  object
dtypes: datetime64[ns](1), object(7)
memory usage: 6.1+ MB
```

order_status	order_purchase_timestamp	order_approved_at	order_delivered_carrier_date	order_delivered_customer_date	order_estimated_delivery_date
invoiced	2017-04-11 12:22:08	2017-04-13 13:25:17	NaN	NaN	2017-05-09 00:00:00
shipped	2018-06-04 16:44:48	2018-06-05 04:31:18	2018-06-05 14:32:00	NaN	2018-06-28 00:00:00
invoiced	2018-08-03 17:44:42	2018-08-07 06:15:14	NaN	NaN	2018-08-21 00:00:00
processing	2017-09-03 14:22:03	2017-09-03 14:30:09	NaN	NaN	2017-10-03 00:00:00

Objective 3: How Have Sales and Orders Changed Over Time?

olist_order_reviews_dataset		
ukey	review_id	object
fkey	order_id	object
	review_score	int

olist_orders_dataset		
ukey	order_id	object
fkey	customer_id	object
	order_status	object
	order_purchase_timestamp	dtime
	order_approved_at	dtime
	order_delivered_carrier_date	dtime
	order_delivered_customer_date	dtime
	order_estimated_delivery_date	dtime

olist_order_items_dataset		
fkey	product_id	object
	order_item_id	int
fkey	order_id	object
fkey	seller_id	object
	shipping_limit_date	dtime
	price	float
	freight_value	float

olist_customers_dataset		
fkey	customer_id	object
ukey	customer_unique_id	object
fkey	customer_zip_code_prefix	int
	customer_city	object

Objective 3: How Have Sales and Orders Changed Over Time?

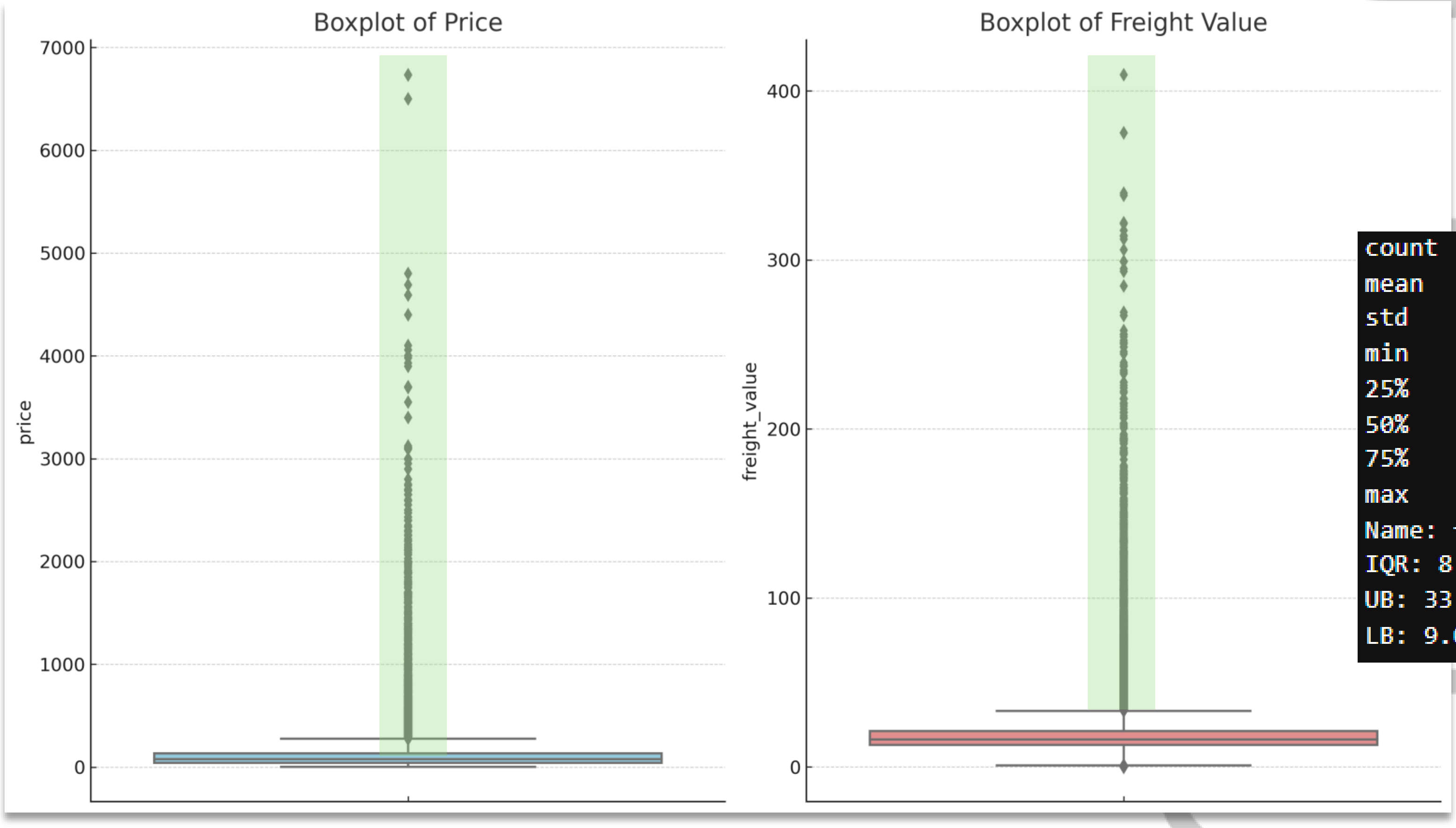
```
RangeIndex: 110839 entries, 0 to 110838
Data columns (total 10 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   order_id                  110839 non-null  object
 1   order_item_id             110839 non-null  int64
 2   product_id                110839 non-null  object
 3   seller_id                 110839 non-null  object
 4   shipping_limit_date       110839 non-null  object
 5   price                     110839 non-null  float64
 6   freight_value             110839 non-null  float64
 7   customer_id               110839 non-null  object
 8   order_purchase_timestamp  110839 non-null  datetime64[ns]
 9   review_score              110012 non-null  float64
```

Cleaning	What We Did	Why This Was Necessary
Drop Null Values	Drop “review_score” null values	To reduce noise and ensure charts reflect real score trend and not missing data dips.

Transformation	What We Did	Why This Was Necessary
Create new column	Quarter/Year	To fit 2016 to 2018 monthly trend into a readable line chart without losing too much insight/s.
Create a subset	Group by Quarter	To get the summation of Price per quarter giving us the total sales.
Transform Datatype	Object to datetime	To get a sequential trend line from 2016 to 2018.

Objective 3: How Have Sales and Orders Changed Over Time?

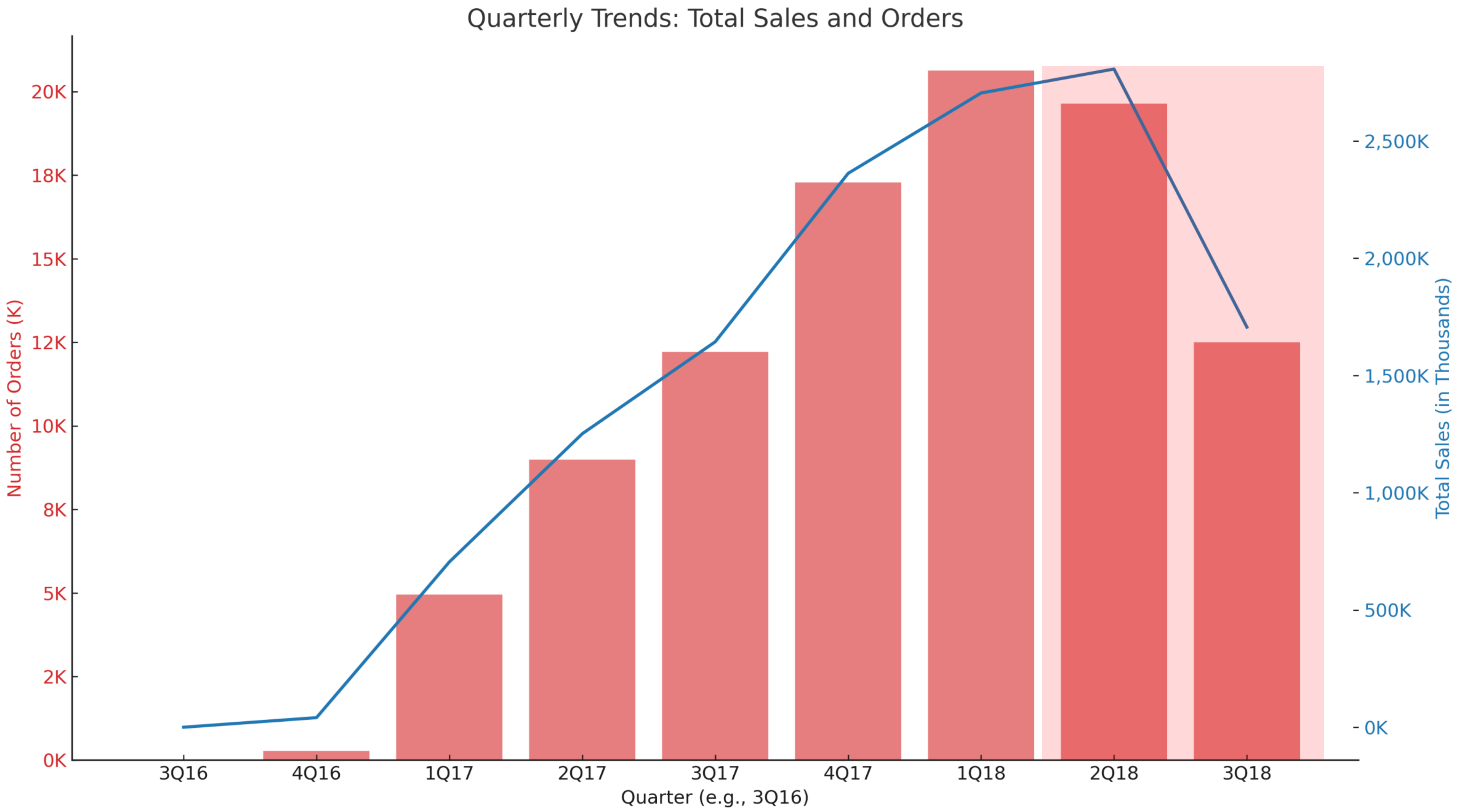
```
count    110012.000000
mean        119.693571
std         180.751636
min           0.850000
25%          39.900000
50%          74.900000
75%         133.900000
max        6735.000000
Name: price, dtype: float64
IQR: 95.0
UB: 277.4
LB: -7.599999999999994
```



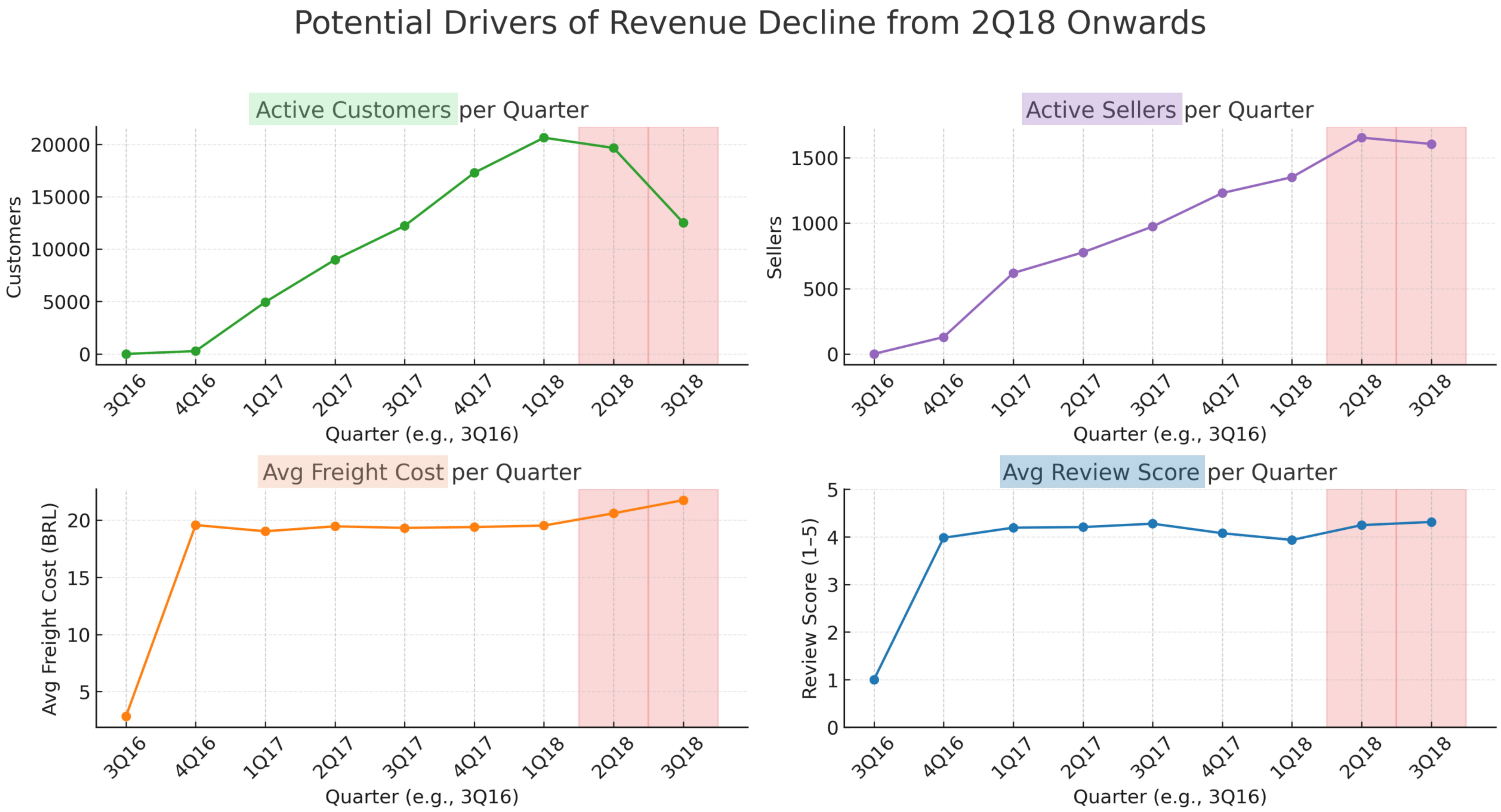

```
count    110012.000000
mean         19.933150
std          15.666202
min           0.000000
25%          13.070000
50%          16.250000
75%          21.150000
max         409.680000
Name: freight_value, dtype: float64
IQR: 8.069999999999999
UB: 33.254999999999995
LB: 9.045000000000002
```

Data Integrity & Outlier/s Handling

Objective 3: How Have Sales and Orders Changed Over Time?



Objective 3: How Have Sales and Orders Changed Over Time?



Objective 4: Develop a model that predicts a more robust estimation of delivery date

Data Transformation

- Converting to date time datatype

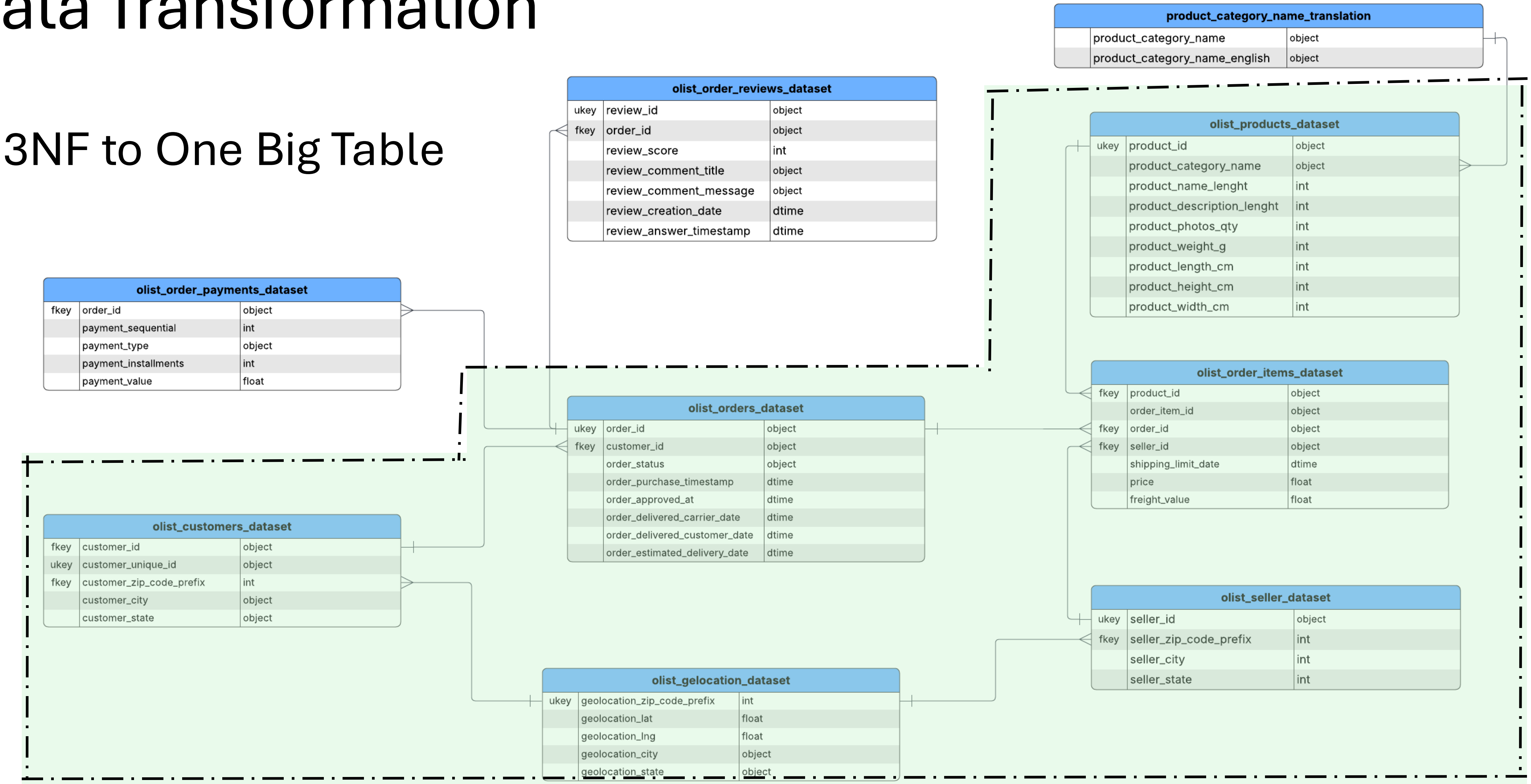
olist_orders_dataset		
ukey	order_id	object
fkey	customer_id	object
	order_status	object
	order_purchase_timestamp	object
	order_approved_at	object
	order_delivered_carrier_date	object
	order_delivered_customer_date	object
	order_estimated_delivery_date	object

Transform

olist_orders_dataset		
ukey	order_id	object
fkey	customer_id	object
	order_status	object
	order_purchase_timestamp	dtime
	order_approved_at	dtime
	order_delivered_carrier_date	dtime
	order_delivered_customer_date	dtime
	order_estimated_delivery_date	dtime

Data Transformation

- 3NF to One Big Table



Data Transformation

- Aggregating zip code prefix by the longitude and latitude average

geolocation_zip_code_prefix	geolocation_lat	geolocation_lng	geolocation_city	geolocation_state
1037	-23.545621	-46.639292	sao paulo	SP
1037	-23.545187	-46.637855	são paulo	SP
1037	-23.546705	-46.640336	são paulo	SP
1037	-23.543883	-46.638075	são paulo	SP
1037	-23.546157	-46.639885	sao paulo	SP
1037	-23.543883	-46.638075	sao paulo	SP
1037	-23.545199	-46.637916	sao paulo	SP
1037	-23.545187	-46.637855	sao paulo	SP
1037	-23.546723	-46.640281	sao paulo	SP
1037	-23.546463	-46.640145	sao paulo	SP
1037	-23.545621	-46.639292	sao paulo	SP

Data Transformation

	seller_customer_distance(km)	product_weight_g	freight_value
count	94273.000000	94273.000000	94273.000000
mean	397.006521	2090.764450	17.895652
std	285.021654	3735.928286	13.002029
min	0.000000	0.000000	0.000000
25%	133.046676	300.000000	12.600000
50%	363.465746	700.000000	15.370000
75%	581.855713	1800.000000	18.700000
max	1093.313840	40425.000000	375.280000

- Standardization of 'seller_customer_distance(km)', 'freight_value', 'product_weight_g'

Does Olist Store meet delivery commitments?

Data Cleaning

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 112650 entries, 0 to 112649
Data columns (total 13 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   order_id                       112650 non-null  object
 1   customer_id                    112650 non-null  object
 2   order_status                   112650 non-null  object
 3   order_purchase_timestamp       112650 non-null  datetime64[ns]
 4   order_approved_at              112635 non-null  datetime64[ns]
 5   order_delivered_customer_date  110196 non-null  datetime64[ns]
 6   order_delivered_carrier_date   111456 non-null  datetime64[ns]
 7   order_estimated_delivery_date  112650 non-null  datetime64[ns]
 8   price                          112650 non-null  float64
 9   freight_value                  112650 non-null  float64
 10  product_weight_g               112632 non-null  float64
 11  seller_zip_code_prefix         112650 non-null  int64
 12  customer_zip_code_prefix       112650 non-null  int64
dtypes: datetime64[ns](5), float64(3), int64(2), object(3)
memory usage: 11.2+ MB
```

Raw Dataset

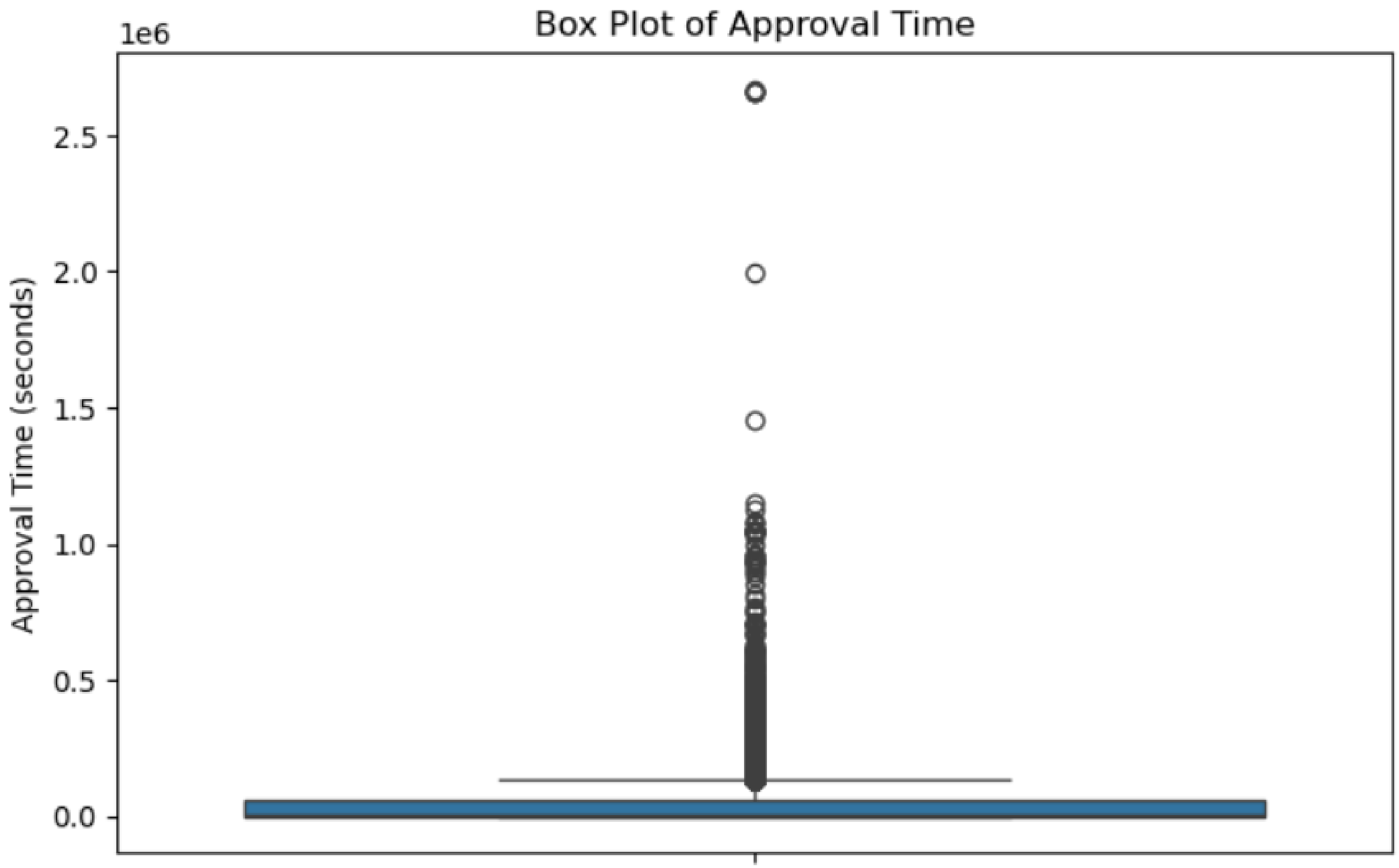
```
<class 'pandas.core.frame.DataFrame'>
Index: 110170 entries, 0 to 112649
Data columns (total 13 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   order_id                       110170 non-null  object
 1   customer_id                    110170 non-null  object
 2   order_status                   110170 non-null  object
 3   order_purchase_timestamp       110170 non-null  datetime64[ns]
 4   order_approved_at              110170 non-null  datetime64[ns]
 5   order_delivered_customer_date  110170 non-null  datetime64[ns]
 6   order_delivered_carrier_date   110170 non-null  datetime64[ns]
 7   order_estimated_delivery_date  110170 non-null  datetime64[ns]
 8   price                          110170 non-null  float64
 9   freight_value                  110170 non-null  float64
 10  product_weight_g               110170 non-null  float64
 11  seller_zip_code_prefix         110170 non-null  int64
 12  customer_zip_code_prefix       110170 non-null  int64
dtypes: datetime64[ns](5), float64(3), int64(2), object(3)
memory usage: 11.8+ MB
```

Cleaned Dataset

Data Cleaning

Cleaning Step	What We Did	Why This Was Necessary
Filter data	Only select "delivered" orders in `order_status` column	Since only delivered status has data for when order was received by the customer.
Dropping Null Values	Drop null values on columns `order_delivered_customer_date`, `product_weight_g`, and 'order_delivered_carrier_date'	Because each data differs from so many factors.
Imputation	Fill null values in `order_approved_at` by the median time it takes for an order to get approved	Outliers are present within the data

Data Cleaning



	approval_time_seconds
count	1.101550e+05
mean	3.786828e+04
std	7.555498e+04
min	0.000000e+00
25%	7.790000e+02
50%	1.261000e+03
75%	5.459400e+04
max	2.669197e+06

Feature Engineering

```
<class 'pandas.core.frame.DataFrame'>
Index: 110170 entries, 0 to 112649
Data columns (total 13 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   order_id                       110170 non-null  object
 1   customer_id                    110170 non-null  object
 2   order_status                   110170 non-null  object
 3   order_purchase_timestamp       110170 non-null  datetime64[ns]
 4   order_approved_at              110170 non-null  datetime64[ns]
 5   order_delivered_customer_date  110170 non-null  datetime64[ns]
 6   order_delivered_carrier_date   110170 non-null  datetime64[ns]
 7   order_estimated_delivery_date  110170 non-null  datetime64[ns]
 8   price                          110170 non-null  float64
 9   freight_value                  110170 non-null  float64
 10  product_weight_g               110170 non-null  float64
 11  seller_zip_code_prefix         110170 non-null  int64
 12  customer_zip_code_prefix       110170 non-null  int64
dtypes: datetime64[ns](5), float64(3), int64(2), object(3)
memory usage: 11.8+ MB
```

Before Feature Engineering

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 109634 entries, 0 to 109633
Data columns (total 24 columns):
 #   Column                                  Non-Null Count   Dtype
---  ------                                  --------------   -----
 0   order_id                                109634 non-null  object
 1   customer_id                             109634 non-null  object
 2   order_status                            109634 non-null  object
 3   order_purchase_timestamp                109634 non-null  datetime64[ns]
 4   order_approved_at                       109634 non-null  datetime64[ns]
 5   order_delivered_customer_date           109634 non-null  datetime64[ns]
 6   order_delivered_carrier_date            109634 non-null  datetime64[ns]
 7   order_estimated_delivery_date           109634 non-null  datetime64[ns]
 8   price                                   109634 non-null  float64
 9   freight_value                           109634 non-null  float64
 10  product_weight_g                        109634 non-null  float64
 11  seller_zip_code_prefix                  109634 non-null  int64
 12  customer_zip_code_prefix                109634 non-null  int64
 13  geolocation_zip_code_prefix_seller      109634 non-null  int64
 14  geolocation_lat_seller                  109634 non-null  float64
 15  geolocation_lng_seller                  109634 non-null  float64
 16  geolocation_zip_code_prefix_customer    109634 non-null  int64
 17  geolocation_lat_customer                109634 non-null  float64
 18  geolocation_lng_customer                109634 non-null  float64
 19  seller_customer_distance(km)            109634 non-null  float64
 20  delivery_time_days                      109634 non-null  float64
 21  carrier_received_time_days              109634 non-null  float64
 22  delivery_difference                     109634 non-null  int64
 23  delivery_status                         109634 non-null  object
dtypes: datetime64[ns](5), float64(10), int64(5), object(4)
memory usage: 20.1+ MB
```

After Feature Engineering

Feature Engineering

Get the distance in kilometers of the seller and customer.

```
from geopy.distance import geodesic
```

```
11   seller_zip_code_prefix                109634 non-null  int64
12   customer_zip_code_prefix              109634 non-null  int64
13   geolocation_zip_code_prefix_seller    109634 non-null  int64
14   geolocation_lat_seller                109634 non-null  float64
15   geolocation_lng_seller                109634 non-null  float64
16   geolocation_zip_code_prefix_customer  109634 non-null  int64
17   geolocation_lat_customer              109634 non-null  float64
18   geolocation_lng_customer              109634 non-null  float64
```

```
19   seller_customer_distance(km)          109634 non-null  float64
```

```
geodesic((lat1, lon1), (lat2, lon2)).kilometers
```

`seller_customer_distance(km)` =

(customer longitude and latitude data) & (seller longitude and latitude data)

Feature Engineering

Calculating the time it took:

- to receive the package by the carrier
- to deliver the package to customer; and
- The difference of the estimated delivery date and delivered date

```
3   order_purchase_timestamp        109634 non-null  datetime64[ns]
4   order_approved_at               109634 non-null  datetime64[ns]
5   order_delivered_customer_date   109634 non-null  datetime64[ns]
6   order_delivered_carrier_date    109634 non-null  datetime64[ns]
7   order_estimated_delivery_date   109634 non-null  datetime64[ns]
```

```
20  delivery_time_days              109634 non-null  float64
21  carrier_received_time_days      109634 non-null  float64
22  delivery_difference             109634 non-null  int64
```

`delivery_time_days` =
`order_delivered_customer_date` - `order_purchase_timestamp`

`carrier_received_time_days` =
`order_delivered_carrier_date` - `order_approved_at`

`delivery_difference` =
`order_estimated_delivery_date` - `order_delivered_customer_date`

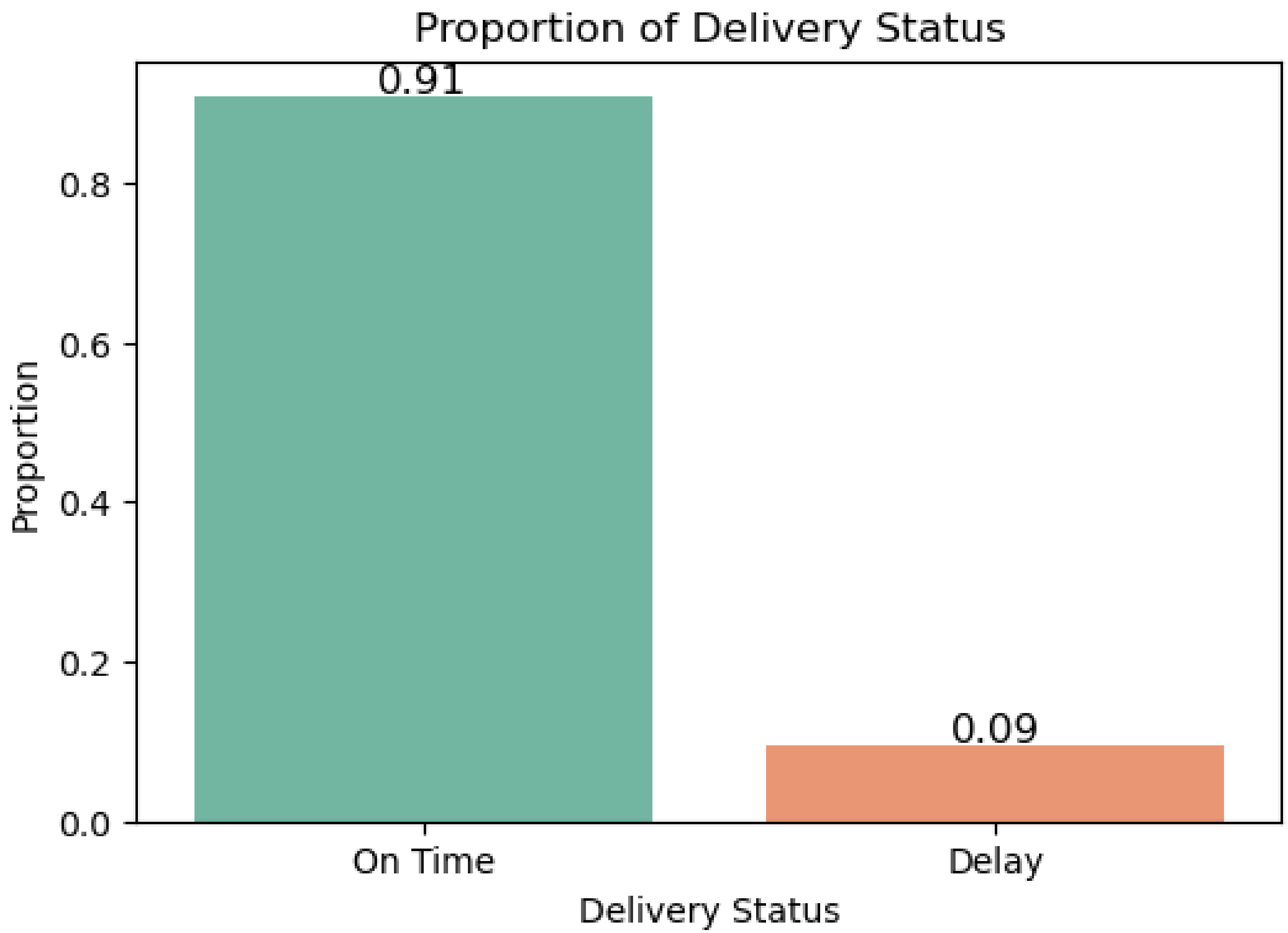
Feature Engineering

Identify whether the order was received on time or was delayed.

```
22  delivery_difference              109634 non-null  int64
23  delivery_status                  109634 non-null  object
```

If delivery difference is positive, then delivery status is "On Time".
While delivery difference is negative, then the order was delivered "Delay".

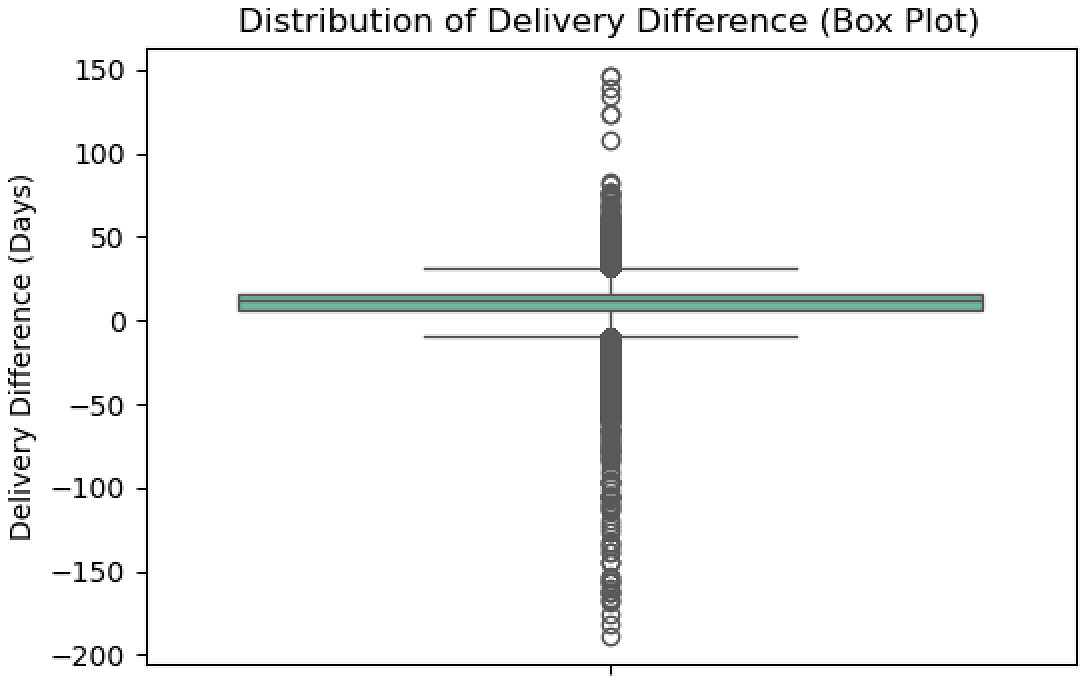
Exploratory Data Analysis

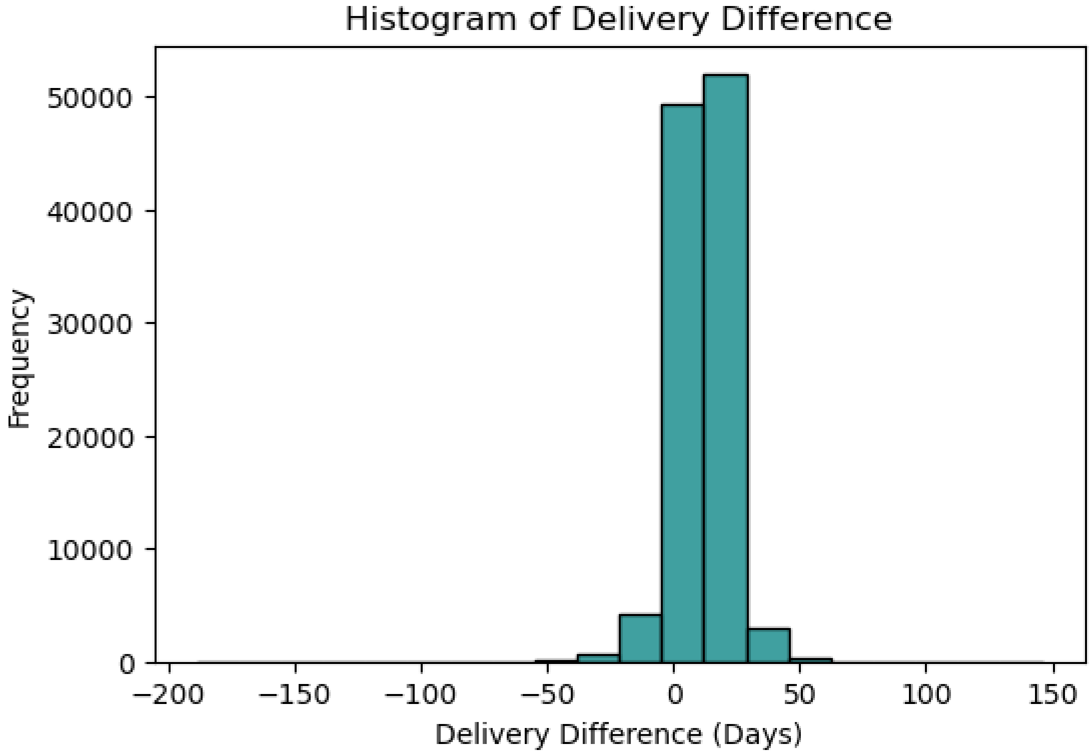


- On Time: When customer received the order on or before the estimated delivery date
- Delay: When customer received the order after the estimated delivery date

Conclusion:
91% of the order was delivered on time.

Exploratory Data Analysis

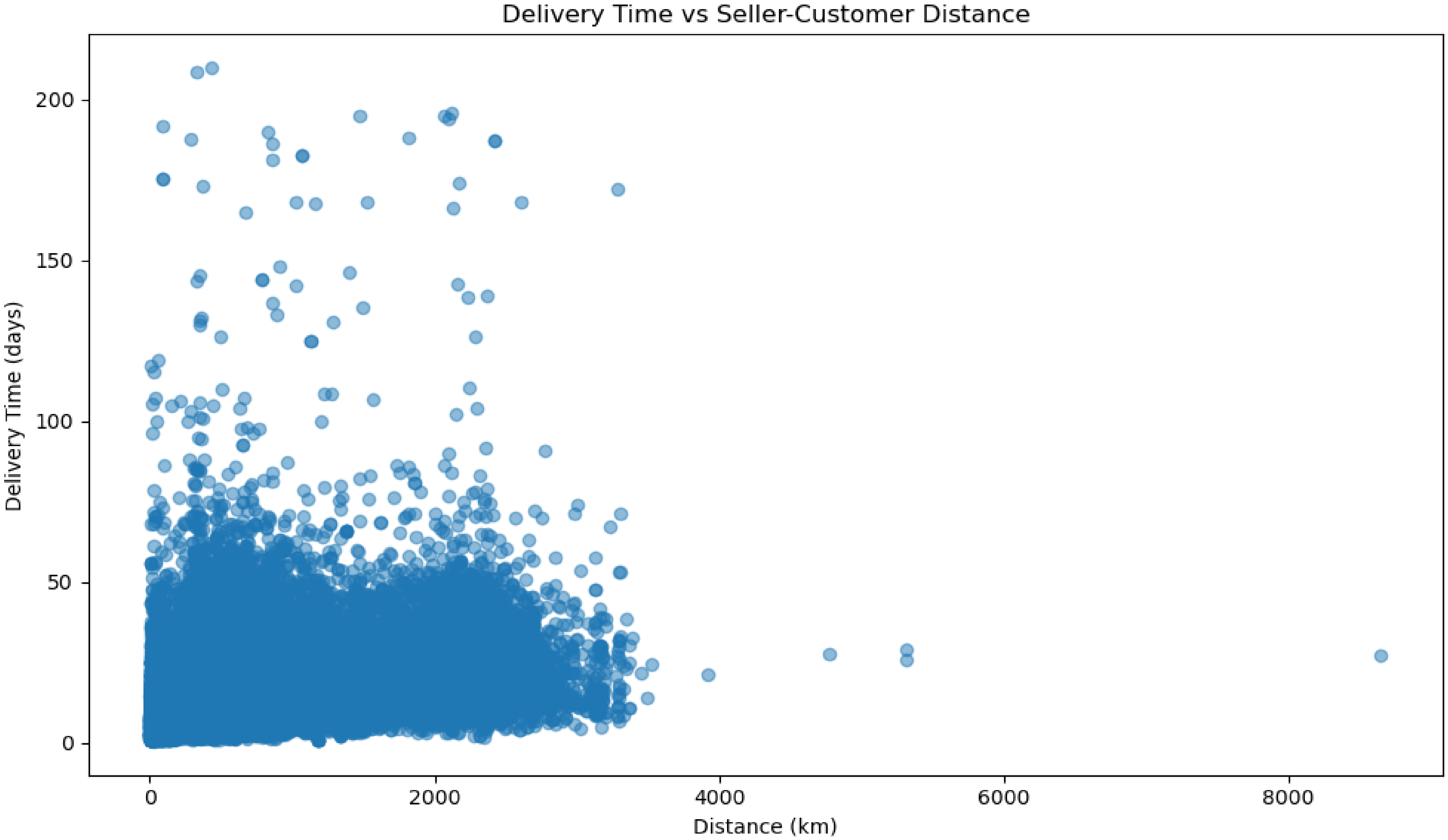




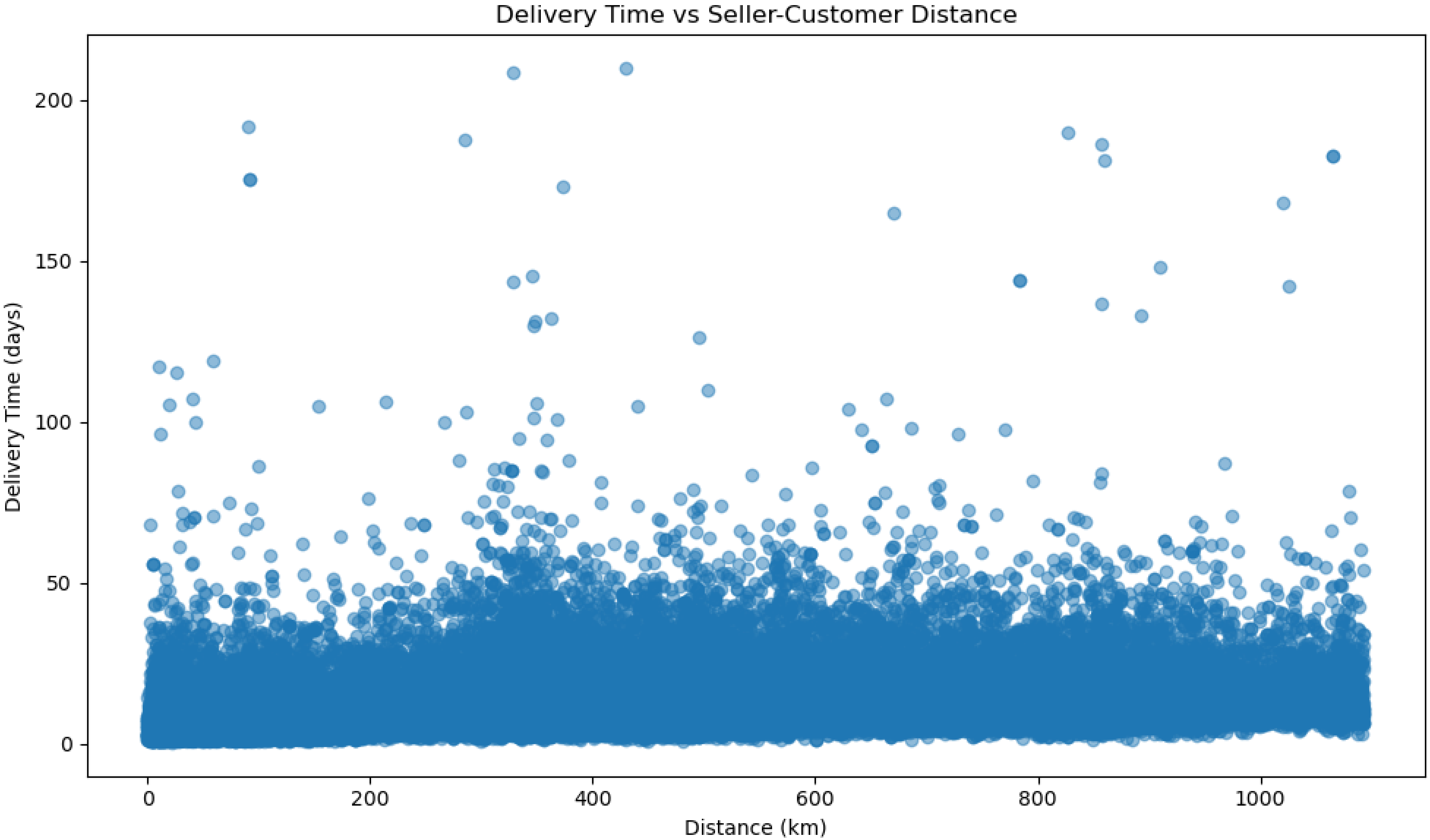
- Interquratile range (IQR) is somehow small: 10 days difference
  Q1 (25th percentile): 6 days difference
  Q3 (75th percentile): 16 days difference

Are the remaining data outliers?

Handling Outliers



Handling Outliers



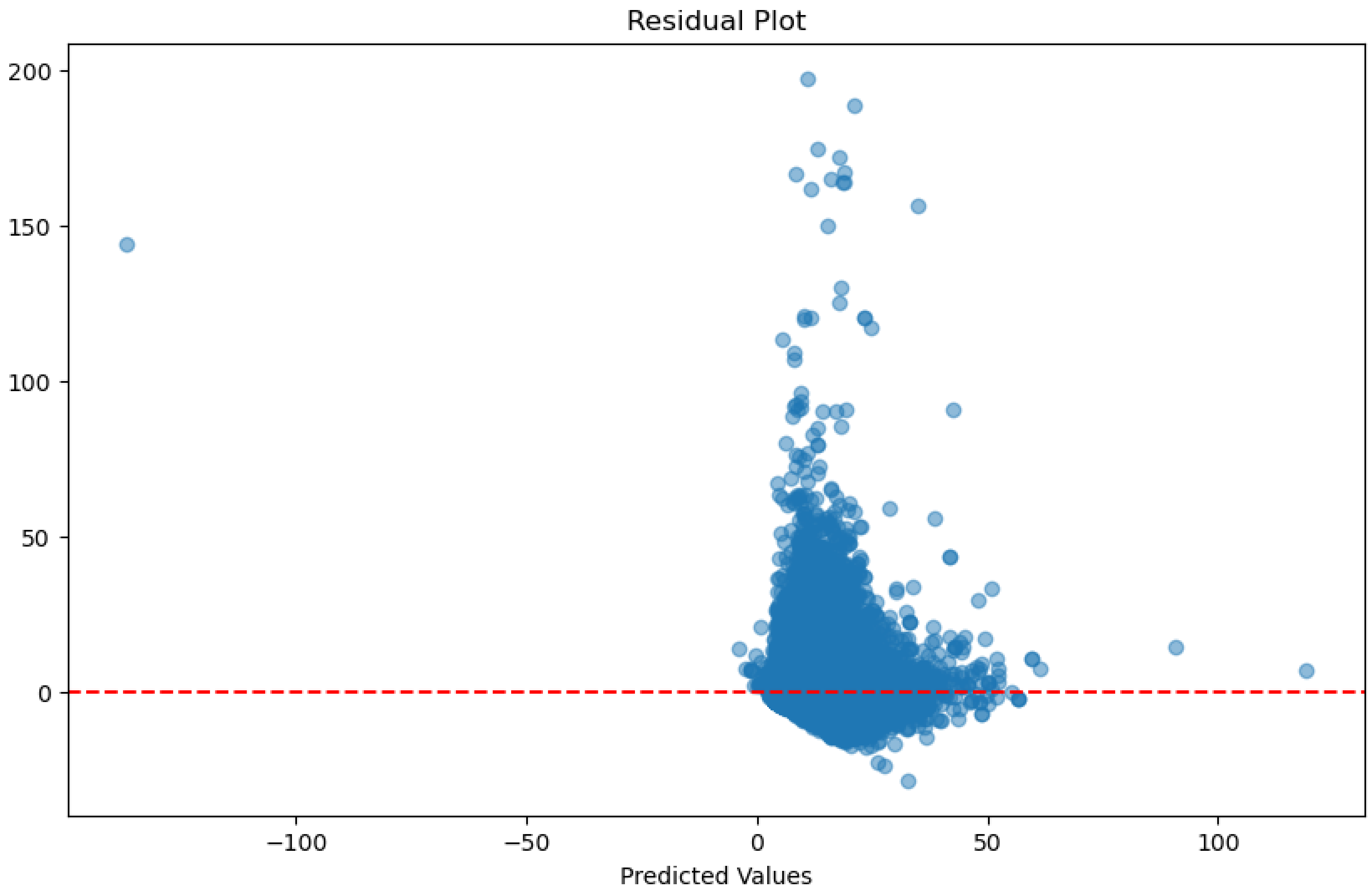
Model 1

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     delivery_time_days   R-squared:                       0.156
Model:                            OLS   Adj. R-squared:                  0.156
Method:                 Least Squares   F-statistic:                 1.622e+04
Date:                Thu, 27 Mar 2025   Prob (F-statistic):               0.00
Time:                        01:29:55   Log-Likelihood:            -3.1401e+05
No. Observations:               87707   AIC:                         6.280e+05
Df Residuals:                   87705   BIC:                         6.280e+05
Df Model:                           1
Covariance Type:            nonrobust
================================================================================================
                                  coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                           8.6733      0.042    207.750      0.000       8.592       8.755
seller_customer_distance(km)    0.0064      5e-05    127.350      0.000       0.006       0.006
==============================================================================
Omnibus:                    84846.194   Durbin-Watson:                   1.992
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          9991917.839
Skew:                           4.411   Prob(JB):                         0.00
Kurtosis:                      54.540   Cond. No.                     1.19e+03
==============================================================================
```

Model 2

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     delivery_time_days   R-squared:                       0.293
Model:                            OLS   Adj. R-squared:                  0.293
Method:                 Least Squares   F-statistic:                     7813.
Date:                Thu, 27 Mar 2025   Prob (F-statistic):               0.00
Time:                        01:30:59   Log-Likelihood:            -2.5563e+05
No. Observations:               75418   AIC:                         5.113e+05
Df Residuals:                   75413   BIC:                         5.113e+05
Df Model:                           4
Covariance Type:            nonrobust
==================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------------------
const                            8.4817      0.034    252.532      0.000       8.416       8.548
seller_customer_distance(km)     2.7219      0.028     97.057      0.000       2.667       2.777
carrier_received_time_days       1.0146      0.007    135.903      0.000       1.000       1.029
freight_value                    0.3281      0.038      8.676      0.000       0.254       0.402
product_weight_g                 0.0467      0.037      1.272      0.204      -0.025       0.119
==============================================================================
Omnibus:                    86310.374   Durbin-Watson:                   1.998
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         21339152.029
Skew:                           5.660   Prob(JB):                         0.00
Kurtosis:                      84.625   Cond. No.                         8.73
==============================================================================
```

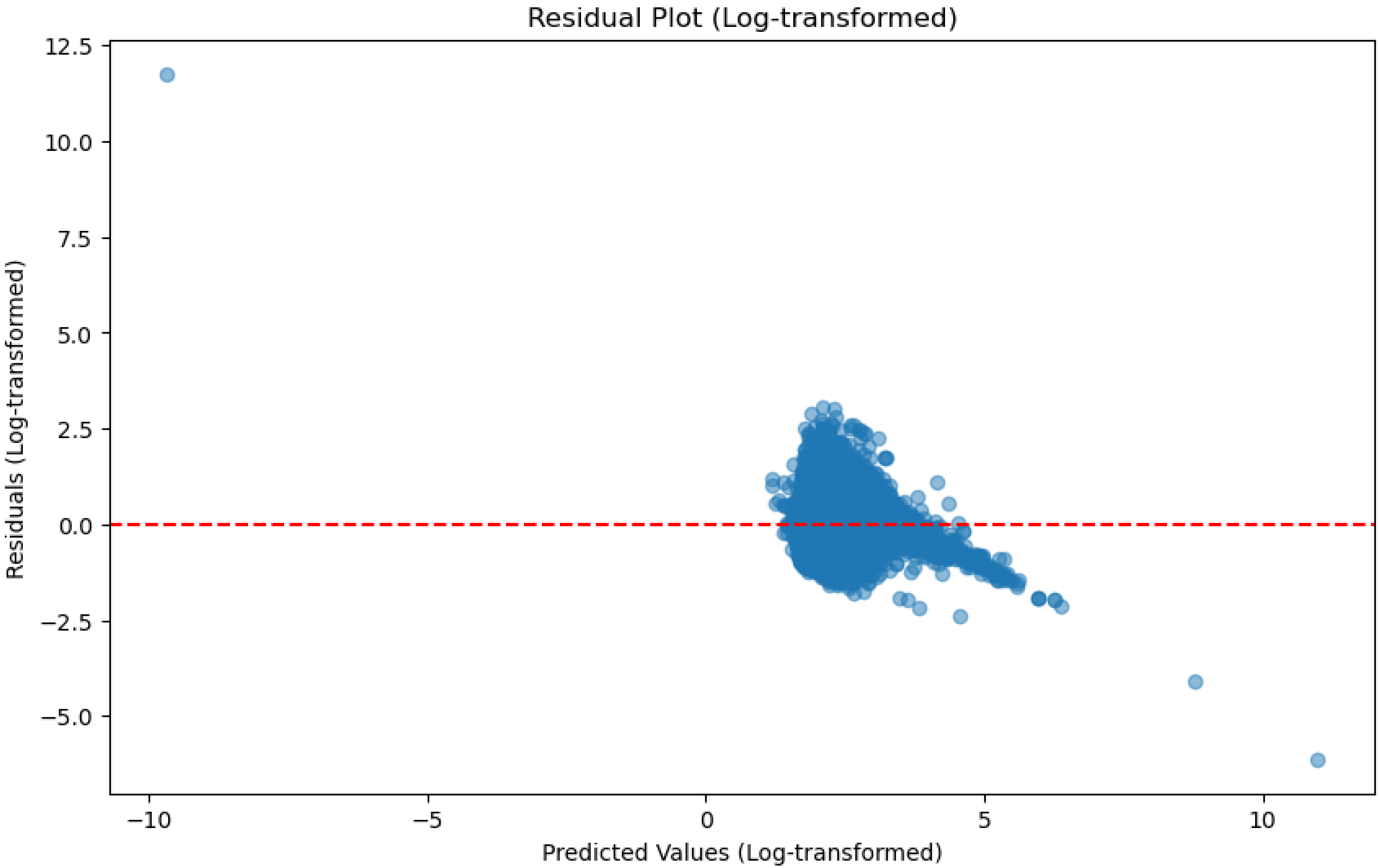
Model 2


Residual Plot
200
150
100
50
0
−100
−50
0
50
100
Predicted Values

Model 3

```
Log Transformation Model Summary:
                            OLS Regression Results
==============================================================================
Dep. Variable:     delivery_time_days   R-squared:                       0.388
Model:                            OLS   Adj. R-squared:                  0.388
Method:                 Least Squares   F-statistic:                 1.194e+04
Date:                Thu, 27 Mar 2025   Prob (F-statistic):               0.00
Time:                        01:32:48   Log-Likelihood:                -48025.
No. Observations:               75418   AIC:                         9.606e+04
Df Residuals:                   75413   BIC:                         9.611e+04
Df Model:                           4
Covariance Type:            nonrobust
================================================================================================
                                  coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                           2.1449      0.002   1001.756      0.000       2.141       2.149
seller_customer_distance(km)    0.2561      0.002    143.252      0.000       0.253       0.260
carrier_received_time_days      0.0685      0.000    144.023      0.000       0.068       0.069
freight_value                   0.0313      0.002     12.995      0.000       0.027       0.036
product_weight_g                0.0033      0.002      1.396      0.163      -0.001       0.008
==============================================================================
Omnibus:                    16066.966   Durbin-Watson:                   1.999
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           169318.019
Skew:                           0.725   Prob(JB):                         0.00
Kurtosis:                      10.196   Cond. No.                         8.73
==============================================================================
```

Model 4


Residual Plot (Log-transformed)
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
Residuals (Log-transformed)
−10
−5
0
5
10
Predicted Values (Log-transformed)

Model 4

```
Log Transformation Model Summary:
                            OLS Regression Results
==============================================================================
Dep. Variable:     delivery_time_days   R-squared:                       0.400
Model:                            OLS   Adj. R-squared:                  0.400
Method:                 Least Squares   F-statistic:                     8382.
Date:                Thu, 27 Mar 2025   Prob (F-statistic):               0.00
Time:                        02:17:06   Log-Likelihood:                -47262.
No. Observations:               75418   AIC:                         9.454e+04
Df Residuals:                   75411   BIC:                         9.460e+04
Df Model:                           6
Covariance Type:            nonrobust
===============================================================================================
                                  coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------------
const                           2.1635      0.002    991.689      0.000       2.159       2.168
seller_customer_distance(km)    0.2468      0.002    138.201      0.000       0.243       0.250
carrier_received_time_days      0.0688      0.000    141.823      0.000       0.068       0.070
freight_value                   0.0675      0.003     26.374      0.000       0.062       0.073
product_weight_g               -0.0074      0.003     -2.715      0.007      -0.013      -0.002
inter1                         -0.0016      0.000     -5.048      0.000      -0.002      -0.001
inter3                         -0.0687      0.002    -39.078      0.000      -0.072      -0.065
==============================================================================
Omnibus:                    16836.606   Durbin-Watson:                   2.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           197631.165
Skew:                           0.741   Prob(JB):                         0.00
Kurtosis:                      10.791   Cond. No.                         14.9
==============================================================================
```

Interaction terms:

inter1 = product_weight_g * carrier_received_time_days

inter3 = seller_customer_distance(km) * freight_value

What if you have a big dataset,(10,000 times in size) of your current dataset?

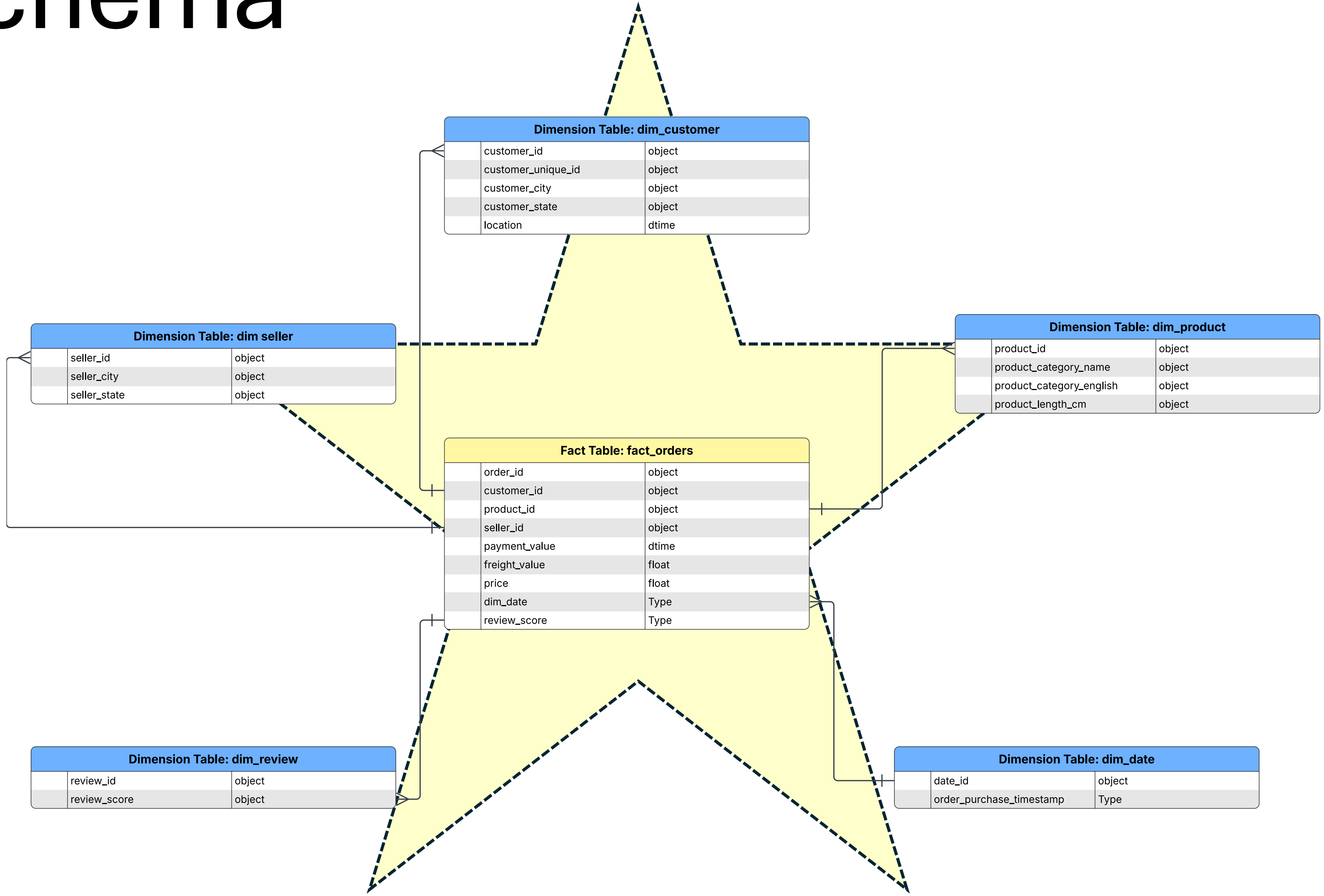
What database, or what software, or what you will do?

Recommended Tool / Stack	Why?
PostgreSQL/ MySQL (Self-Hosted / Budget-Friendly)	Structured data handling (customer IDs, order IDs, payment details).
Hadoop (Self-Hosted / Budget-Friendly)	Distributed storage for massive datasets (10,000x).
MongoDB + BI Tools (PowerBI, Tableau)	Semi-structured data handling (JSON, reviews).
GCP or AWS S3 + Athena + Pandas (Startup / Prototype)	Cloud solutions for fast prototyping and scalability.
Google BigQuery (Mid-Scale)	Scalable, serverless solutions for analytical workloads.
Amazon Redshift (Enterprise)	High-performance data warehousing with cloud integration.
Apache Spark / Dask (In-Memory Processing)	Fast, distributed computing for large datasets.

Final Choice: Amazon Redshift

- **Handling Big Data Efficiently**
- **Distributed & Scalable**
- **Optimized for Complex Queries**
- **Integration with AWS Ecosystem**
- **Cost-Effective**

Star Schema



Dimension Table: dim_customer
customer_id object
customer_unique_id object
customer_city object
customer_state object
location dtime
Dimension Table: dim seller
seller_id object
seller_city object
seller_state object
Dimension Table: dim_product
product_id object
product_category_name object
product_category_english object
product_length_cm object
Fact Table: fact_orders
order_id object
customer_id object
product_id object
seller_id object
payment_value dtime
freight_value float
price float
dim_date Type
review_score Type
Dimension Table: dim_review
review_id object
review_score object
Dimension Table: dim_date
date_id object
order_purchase_timestamp Type

ETL

ELT