Show code

Olist Brazilian E-Commerce Analysis

Retail & Customer Analytics (2016–2018)

Emilio Nahuel Pattini
Buenos Aires, Argentina
February 01, 2026

Table of Contents

Introduction

Project Goal
Business-oriented analysis: sales, customer behavior, RFM, cohorts, CLV, basic forecasting + actionable recommendations.

Dataset
Olist Public Dataset – ~100k orders (Kaggle)

Tech Stack

- Python: pandas, seaborn, plotly, matplotlib, prophet
- Dashboard: Power BI

---

1. Business Understanding & Questions

Key questions:

- Top categories / products by revenue?
- Customer retention & recurrence?
- RFM segmentation?
- Cohort retention over time?
- Delivery performance by region?
- Payment methods impact?
- Cross-sell opportunities?
- Forecast for top categories?

---

2. Data Loading & Overview

This section covers the initial ingestion of the raw CSV files from the Olist dataset and provides a high-level understanding of the structure, size, and content of each table. The

goal is to confirm successful loading, identify key relationships between tables, and spot any immediate data quality signals before proceeding to cleaning and analysis.

2.1. Data Loading

I load the most relevant tables from the Olist dataset (orders, order_items, customers, payments, reviews) using pandas.
Only essential tables are loaded at this stage to keep memory usage low and focus on the core entities needed for sales, customer, and logistics analysis.

```
Execution environment initialized successfully.
• Pandas version: 2.3.3
```

2.2. Data Overview & Initial Checks

I perform a quick inspection of each loaded table to understand:

- Number of rows and columns
- Data types
- Presence of missing values
- Sample rows

This step helps map the dataset schema and decide on next cleaning priorities.

```
=== ORDERS ===
Rows: 99,441
Columns: 8

Data types and missing values:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99441 entries, 0 to 99440
Columns: 8 entries, order_id to order_estimated_delivery_date
dtypes: object(8)
memory usage: 6.1+ MB
None

First three rows:
```

	order_id	customer_id	order_status	orde
0	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	
1	53cdb2fc8bc7dce0b6741e2150273451	b0830fb4747a6c6d20dea0b8c802d7ef	delivered	
2	47770eb9100c2d0c44946d9cf07ec65d	41ce2a54c0b03bf3443c3d931a367089	delivered	

```
=== ORDER ITEMS ===
Rows: 112,650
Columns: 7

Data types and missing values:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 112650 entries, 0 to 112649
Columns: 7 entries, order_id to freight_value
dtypes: float64(2), int64(1), object(4)
memory usage: 6.0+ MB
None

First three rows:
```

	order_id	order_item_id	product_id	
0	00010242fe8c5a6d1ba2dd792cb16214	1	4244733e06e7ecb4970a6e2683c13e61	484
1	00018f77f2f0320c557190d7a144bdd3	1	e5f2d52b802189ee658865ca93d83a8f	dd1
2	000229ec398224ef6ca0657da4fc703e	1	c777355d18b72b67abbeef9df44fd0fd	5b5

```
=== CUSTOMERS ===
Rows: 99,441
Columns: 5

Data types and missing values:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99441 entries, 0 to 99440
Columns: 5 entries, customer_id to customer_state
dtypes: int64(1), object(4)
memory usage: 3.8+ MB
None

First three rows:
```

	customer_id	customer_unique_id	customer_zip_cod
0	06b8999e2fba1a1fbc88172c00ba8bc7	861eff4711a542e4b93843c6dd7febb0	
1	18955e83d337fd6b2def6b18a428ac77	290c77bc529b7ac935b93aa66c333dc3	
2	4e7b3e00288586ebd08712fdd0374a03	060e732b5b29e8181a18229c7b0b2b5e	

```
=== PAYMENTS ===
Rows: 103,886
Columns: 5

Data types and missing values:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103886 entries, 0 to 103885
Columns: 5 entries, order_id to payment_value
dtypes: float64(1), int64(2), object(2)
memory usage: 4.0+ MB
None

First three rows:
```

	order_id	payment_sequential	payment_type	payment_installme
0	b81ef226f3fe1789b1e8b2acac839d17	1	credit_card	
1	a9810da82917af2d9aefd1278f1dcfa0	1	credit_card	
2	25e8ea4e93396b6fa0d3dd708e76c1bd	1	credit_card	

```
=== REVIEWS ===
Rows: 99,224
Columns: 7

Data types and missing values:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99224 entries, 0 to 99223
Columns: 7 entries, review_id to review_answer_timestamp
dtypes: int64(1), object(6)
memory usage: 5.3+ MB
None

First three rows:
```

	review_id	order_id	review_score	revi
0	7bc2406110b926393aa56f80a40eba40	73fc7af87114b39712e6da79b0a377eb	4	
1	80e641a11e56f04c1ad469d5645fdfde	a548910a1c6147796b98fdf73dbeba33	5	
2	228ce5500dc1d8e020d8d1322874b6f0	f9e4b658b201a9f2ecdecbb34bed034b	5	

---

3. Data Cleaning & Preparation

3.1. Initial Data Cleaning

In this phase I perform foundational data quality checks: converting timestamp columns to proper datetime format, verifying uniqueness of key identifiers (order_id, customer_id), and

inspecting missing values in critical columns. The goal is to ensure the raw data is reliable and ready for analysis without introducing errors in calculations or joins.

Type conversions

Convert string timestamps to datetime objects so I can perform time-based calculations (deltas, grouping by month, etc.) accurately.

```
Date columns converted:
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]
dtype: object
```

Duplicate checks

Verify that primary keys (order_id in orders, customer_id in customers) have no duplicates, preventing inflated counts during merges or aggregations.

```
Duplicates:
orders order_id duplicated: 0
customers customer_id duplicated: 0
order_items (should be 0): 0
```

Missing values inspection

Identify and understand missing data patterns, especially in delivery-related timestamps and review comments, to decide on appropriate handling strategies.

```
Missing values in orders:
order_approved_at                 160
order_delivered_carrier_date     1783
order_delivered_customer_date    2965
dtype: int64

Missing values in reviews (expected high in comments):
review_comment_title      87656
review_comment_message    58247
dtype: int64
```

3.2. Feature Engineering & Business Transformations

Status flags

Create boolean columns (is_delivered, is_approved) to easily filter completed orders and avoid NaN issues in time-based metrics.

```
Order status breakdown after cleaning:
order_status
delivered     97.000000
shipped        1.100000
canceled       0.600000
unavailable    0.600000
invoiced       0.300000
processing     0.300000
created        0.000000
approved       0.000000
Name: proportion, dtype: float64
```

Delivery time calculations

Compute actual_delivery_time_days: the real calendar days from purchase to customer delivery — key for understanding customer experience and logistics speed.

Delay & performance metrics

Calculate actual_minus_estimated_delivery_days: how much earlier or later the order arrived compared to the promised date (negative = early, positive = late) — essential for evaluating Olist's delivery promise accuracy and customer satisfaction impact.

```
% of orders with reasonable delivery time (0–60 days): 96.7

Delivery time stats (only delivered orders):
       actual_delivery_time_days  actual_minus_estimated_delivery_days
count               96476.000000                          96476.000000
mean                   12.094086                            -11.876881
std                     9.551746                             10.183854
min                     0.000000                           -147.000000
25%                     6.000000                            -17.000000
50%                    10.000000                            -12.000000
75%                    15.000000                             -7.000000
max                   209.000000                            188.000000
```

---

4. Data Integration & First Business Insights

4.1. Saving Cleaned Data

I save the cleaned and enriched `orders` DataFrame (with new features) to a processed folder.

This follows best practices: never overwrite raw data, and create reproducible intermediate files.

```
Cleaned & enriched orders saved to:
./data/processed/cleaned_orders_with_features.csv
```

4.2. First Merge – Creating a Base Working Table

I combine the core tables (orders + customers + payments) into one master DataFrame.

This gives us a single table with customer location, total payment value, and order details — ideal for revenue analysis, customer segmentation, and geographic insights.

```
Shape before merges: (99441, 12)
Shape after merges: (99441, 16)

Missing total_order_value after merge: 1

First 3 rows of base working table:
```

	order_id	customer_unique_id	customer_state	or
0	e481f51cbdc54678b7cc49136f2d6af7	7c396fd4830fd04220f754e42b4e5bff	SP	
1	53cdb2fc8bc7dce0b6741e2150273451	af07308b275d755c9edb36a90c618231	BA	
2	47770eb9100c2d0c44946d9cf07ec65d	3a653a41f6f9fc3d2a113cf8398680e8	GO	

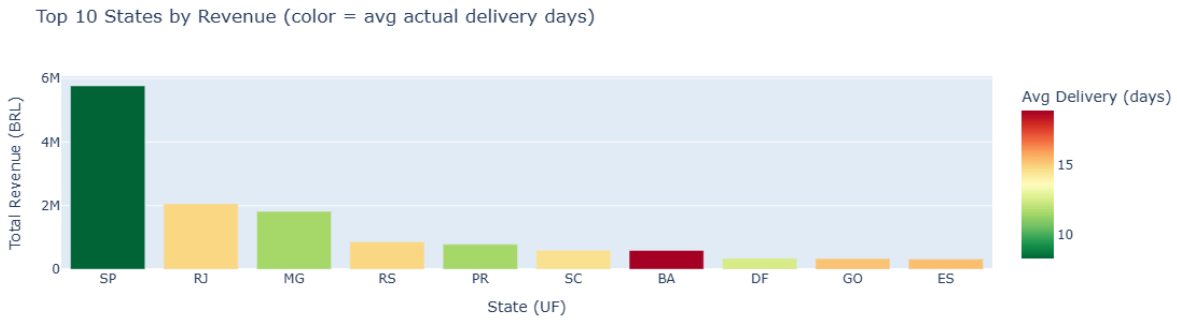
4.3. First Business Insight: Revenue & Delivery Performance by State

I aggregate the base table to calculate total revenue and average delivery time per customer state.

This gives us an initial view of geographic performance — identifying high-value regions and potential logistics bottlenecks.

```
Top 10 states by total revenue (delivered orders only):
```

	customer_state	total_revenue	avg_delivery_days	median_delivery_days	order_count	avg
25	SP	5,769,221.49	8.3	7.0	40,495	
18	RJ	2,056,101.21	14.8	12.0	12,353	
10	MG	1,819,321.70	11.5	10.0	11,355	
22	RS	861,608.40	14.8	13.0	5,344	
17	PR	781,919.55	11.5	10.0	4,923	
23	SC	595,361.91	14.5	13.0	3,547	
4	BA	591,270.60	18.9	16.0	3,256	
6	DF	346,146.17	12.5	11.0	2,080	
8	GO	334,294.22	15.2	13.0	1,957	
7	ES	317,682.65	15.3	13.0	1,995	



Saving State-Level Summary

I export the aggregated state performance metrics to a processed file for future use (e.g., dashboarding in Power BI or additional visualizations).

```
State summary saved successfully to:
./data/processed/state_performance_summary.csv
```

---

5. Product & Category Analysis

5.1. Loading Additional Tables

I load the product-related tables to enable category-level insights.

- `products` : product attributes (category, dimensions)

- `product_category_name_translation` : English translation of Portuguese category names
- `order_items` : links orders to products (quantity, price, freight) and was loaded previously.

```
order_items shape: (112650, 7)
products shape: (32951, 9)
category_translation shape: (71, 2)

=== PRODUCTS ===
Rows: 32,951
Columns: 9

Data types and missing values:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32951 entries, 0 to 32950
Columns: 9 entries, product_id to product_width_cm
dtypes: float64(7), object(2)
memory usage: 2.3+ MB
None

First three rows:
```

	product_id	product_category_name	product_name_lenght	produc
0	1e9e8ef04dbcff4541ed26657ea517e5	perfumaria	40.000000	
1	3aa071139cb16b67ca9e5dea641aaa2f	artes	44.000000	
2	96bd76ec8810374ed1b65e291975717f	esporte_lazer	46.000000	

```
=== CATEGORY TRANSLATION ===
Rows: 71
Columns: 2

Data types and missing values:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71 entries, 0 to 70
Columns: 2 entries, product_category_name to product_category_name_english
dtypes: object(2)
memory usage: 1.2+ KB
None

First three rows:
```

	product_category_name	product_category_name_english
0	beleza_saude	health_beauty
1	informatica_acessorios	computers_accessories
2	automotivo	auto

5.2. Merging Product Information into the Base Table

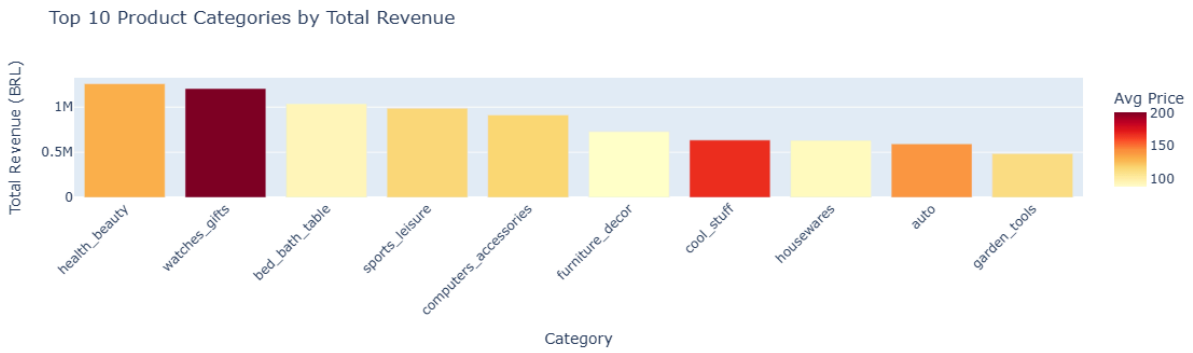
I join `order_items` with `products` and category translations, then aggregate to get category performance metrics (revenue, order count, average price, etc.).

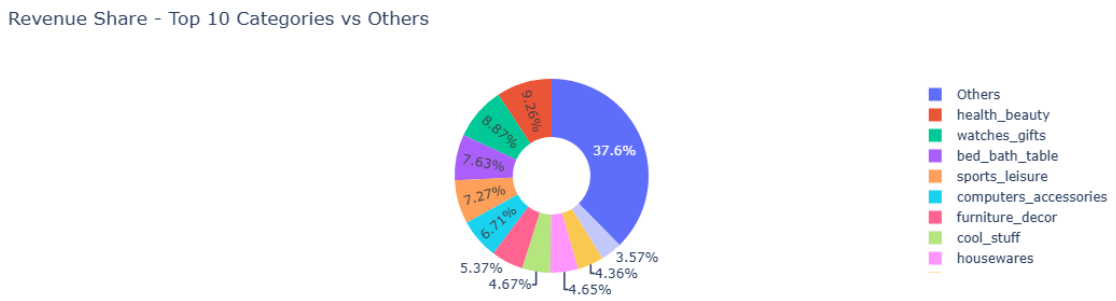
```
Missing English category names: 0

Top 15 categories by total revenue:
```

	product_category_name_english	total_revenue	total_freight	order_items_count	unique_o
43	health_beauty	1,258,681.34	182,566.73	9670	
71	watches_gifts	1,205,005.68	100,535.93	5991	
7	bed_bath_table	1,036,988.68	204,693.04	11115	
65	sports_leisure	988,048.97	168,607.51	8641	
15	computers_accessories	911,954.32	147,318.08	7827	
39	furniture_decor	729,762.49	172,749.30	8334	
20	cool_stuff	635,290.85	84,039.10	3796	
49	housewares	632,248.66	146,149.11	6964	
5	auto	592,720.11	92,664.21	4235	
42	garden_tools	485,256.46	98,962.75	4347	
69	toys	483,946.60	77,425.95	4117	
6	baby	411,764.89	68,353.11	3065	
59	perfumery	399,124.87	54,213.84	3419	
68	telephony	323,667.53	71,215.79	4545	
57	office_furniture	273,960.70	68,571.95	1691	

5.3. Category Performance Visualization & Insights

Visualize the top categories by revenue and create actionable insights around pricing, freight, and opportunities.





Key Observations & Early Recommendations:

- **Health & Beauty** leads with ~10% of total revenue: high volume combined with a solid average price → ideal for mass campaigns, volume promotions, and broad marketing efforts.
- **Bed Bath & Table** and **Furniture Decor** show significantly high freight costs relative to price → opportunity to review pricing (increase margins) or negotiate better logistics rates with carriers.
- **Watches & Gifts** has the highest average ticket (~201 BRL) → strong potential for upselling, premium bundles, personalized recommendations, and loyalty programs targeting higher-value customers.
- The top 5 categories account for more than 40% of total revenue → high concentration risk; consider diversifying by promoting emerging or underperforming categories.
- 1,627 items remain uncategorized (~1–2% of revenue) → worth a manual review to create new categories, improve product discoverability, and enhance recommendation algorithms.

5.4. Category Performance by State

I merge category information back with the base customer table to analyze which product categories perform best in each Brazilian state.

This helps identify regional preferences, localized opportunities, and potential logistics/pricing adjustments per region.

```
Top 10 categories in SP by revenue:
```

	customer_state	product_category_name_english	total_revenue	order_count	avg_ticket
1272	SP	bed_bath_table	549,408.91	4307	127.56
1308	SP	health_beauty	509,859.18	3693	138.10
1335	SP	watches_gifts	449,135.06	2083	215.62
1329	SP	sports_leisure	427,734.06	3203	133.54
1280	SP	computers_accessories	386,706.97	2609	148.22
1304	SP	furniture_decor	331,287.25	2618	126.54
1314	SP	housewares	323,729.96	2693	120.21
1270	SP	auto	235,440.59	1579	149.11
1285	SP	cool_stuff	230,410.92	1279	180.15
1333	SP	toys	205,513.18	1568	131.07

Top 10 categories in RJ by revenue:

	customer_state	product_category_name_english	total_revenue	order_count	avg_ticket
986	RJ	watches_gifts	188,485.58	784	240.42
924	RJ	bed_bath_table	175,594.31	1342	130.85
960	RJ	health_beauty	159,174.66	935	170.24
980	RJ	sports_leisure	140,578.56	889	158.13
932	RJ	computers_accessories	138,232.02	832	166.14
956	RJ	furniture_decor	118,425.00	809	146.38
966	RJ	housewares	93,000.20	709	131.17
937	RJ	cool_stuff	91,488.42	478	191.40
959	RJ	garden_tools	85,899.27	522	164.56
984	RJ	toys	83,263.44	539	154.48

Top 10 categories in MG by revenue:

	customer_state	product_category_name_english	total_revenue	order_count	avg_ticket
514	MG	health_beauty	175,305.23	987	177.61
480	MG	bed_bath_table	155,527.65	1108	140.37
541	MG	watches_gifts	132,117.43	598	220.93
535	MG	sports_leisure	130,027.02	845	153.88
487	MG	computers_accessories	126,693.85	857	147.83
510	MG	furniture_decor	97,409.77	701	138.96
520	MG	housewares	92,826.66	676	137.32
478	MG	auto	82,521.85	458	180.18
492	MG	cool_stuff	79,890.81	422	189.31
513	MG	garden_tools	72,488.16	478	151.65

Top 10 categories in RS by revenue:

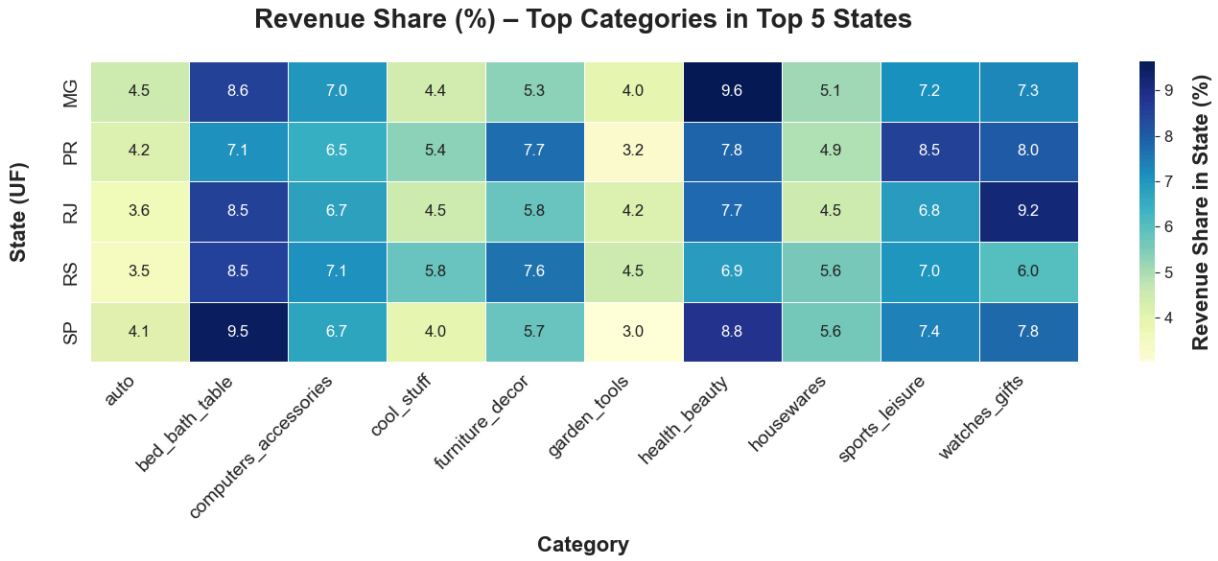
	customer_state	product_category_name_english	total_revenue	order_count	avg_ticket
1098	RS	bed_bath_table	73,416.22	532	138.00
1127	RS	furniture_decor	65,638.11	425	154.44
1106	RS	computers_accessories	61,275.72	385	159.16
1151	RS	sports_leisure	60,578.49	411	147.39
1130	RS	health_beauty	59,453.00	388	153.23
1157	RS	watches_gifts	51,874.17	222	233.67
1111	RS	cool_stuff	49,747.79	251	198.20
1136	RS	housewares	48,260.70	340	141.94
1129	RS	garden_tools	38,871.63	219	177.50
1155	RS	toys	33,347.80	200	166.74

Top 10 categories in PR by revenue:

	customer_state	product_category_name_english	total_revenue	order_count	avg_ticket
911	PR	sports_leisure	66,731.65	419	159.26
917	PR	watches_gifts	62,263.22	265	234.96
891	PR	health_beauty	61,366.50	375	163.64
888	PR	furniture_decor	60,326.83	382	157.92
859	PR	bed_bath_table	55,499.30	395	140.50
866	PR	computers_accessories	50,583.03	333	151.90
871	PR	cool_stuff	41,963.34	201	208.77
897	PR	housewares	38,546.88	279	138.16
857	PR	auto	32,421.59	202	160.50
915	PR	toys	27,313.06	198	137.94

5.5. Visualization: Category Revenue Share by State

A heatmap shows the relative importance of each category within the top states, highlighting regional preferences.



Regional Category Insights & Recommendations:

- **São Paulo (SP)**: Dominated by bed_bath_table (~9.5%), health_beauty (8.8%), and watches_gifts → mature market with diverse demand; prioritize bundles in home goods + beauty and targeted ads in these categories.
- **Rio de Janeiro (RJ) & Minas Gerais (MG)**: Higher relative share in furniture_decor and housewares → regional preference for home-related items; consider free shipping promotions or localized pricing to offset freight sensitivity.

- **Southern states (RS, PR)**: More balanced toward sports_leisure, toys, and cool_stuff → possible seasonal/cultural influence; explore summer campaigns or kid-focused promotions.
- **General opportunity**: Tailor product recommendations and marketing by state (e.g., beauty focus in SP, furniture in MG/RJ) → potential uplift in conversion and average order value.

---

6. Customer Segmentation – RFM Analysis

6.1. RFM Calculation

I calculate the classic RFM metrics for each unique customer:

- **Recency**: Days since last purchase (lower = more recent)
- **Frequency**: Number of orders placed
- **Monetary**: Total revenue generated by the customer

This forms the foundation for segmenting customers into groups (e.g., VIPs, at-risk, new, lost) and deriving retention/upselling strategies.

```
RFM table shape: (93356, 4)

RFM descriptive stats:
           recency    frequency     monetary
count 93356.000000 93356.000000 93356.000000
mean    237.970000     1.030000   165.190000
std     152.620000     0.210000   226.320000
min       1.000000     1.000000     0.000000
25%     114.000000     1.000000    63.050000
50%     219.000000     1.000000   107.780000
75%     346.000000     1.000000   182.540000
max     714.000000    15.000000 13664.080000

Top 10 customers by total spend:
```

	customer_unique_id	recency	frequency	monetary
3724	0a0a92112bd4c708ca5fde585afaa872	334	1	13,664.08
79634	da122df9eeddfedc1dc1f5349a1a690c	515	2	7,571.63
43166	763c8b1c9c68a0229c42c9fc6f662b93	46	1	7,274.88
80461	dc4802a71eae9be1dd28f5d788ceb526	563	1	6,929.31
25432	459bef486812aa25204be022145caa62	35	1	6,922.21
93079	ff4159b92c40ebe40454e3e6a7c35ed6	462	1	6,726.66
23407	4007669dec559734d6f53e029e360987	279	1	6,081.54
87145	eebb5dda148d3893cdaf5b5ca3040ccb	498	1	4,764.34
26636	48e1ac109decbb87765a3eade6854098	69	1	4,681.78
73126	c8460e4251689ba205045f3ea17884a1	22	4	4,655.91

6.2. RFM Scoring & Customer Segmentation

I assign scores (4 = best, 1 = worst) to Recency (lower = better), Frequency, and Monetary.

- Recency and Monetary use quartile-based scoring (`pd.qcut`)
- Frequency uses custom thresholds due to extreme skew (97% of customers buy only once)

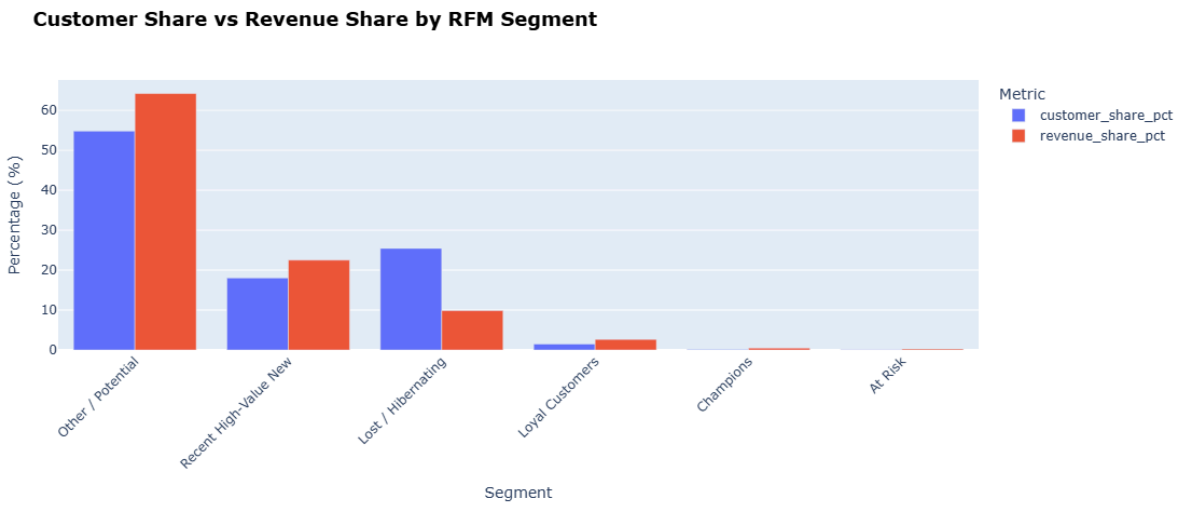
Then I combine scores into actionable customer segments for retention, re-engagement, and upselling strategies.

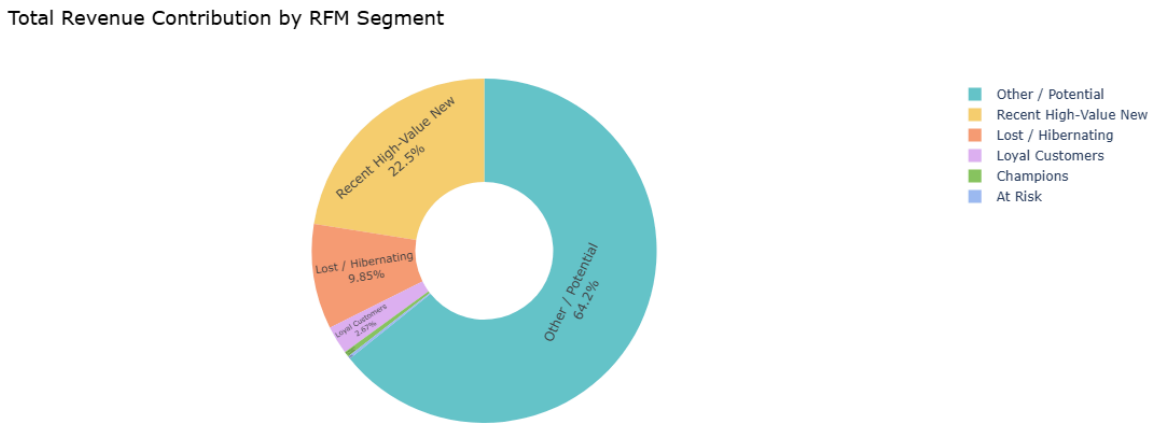
```
Frequency score distribution:
F_score
1   96.999657
2    2.756116
3    0.223874
4    0.020352
Name: proportion, dtype: float64

RFM Segments Summary:
```

	segment	customer_count	avg_recency_days	avg_frequency	avg_monetary	total_revenu
4	Other / Potential	51147	241.7	1.02	193.65	9,904,644.9
5	Recent High-Value New	16824	57.9	1.00	206.31	3,471,006.7
2	Lost / Hibernating	23750	365.1	1.01	63.98	1,519,629.3
3	Loyal Customers	1408	113.7	2.00	291.97	411,089.8
1	Champions	141	105.7	3.55	542.11	76,437.4
0	At Risk	86	362.9	3.15	453.76	39,023.1

6.3. RFM Segment Visualization & Actionable Recommendations

Let's visualize the distribution of customers and revenue across segments, then derive concrete business strategies for each group (retention, re-engagement, upselling, pricing adjustments, etc.).





Actionable Recommendations by Segment:

- **Other / Potential** (54.79% customers, 64.22% revenue)

  Largest group, core revenue driver but middle performance.

  → Focus on conversion to higher segments: personalized emails with discounts on next purchase, cross-sell recommendations (e.g., bundle health_beauty with bed_bath_table).

- **Recent High-Value New** (18.02% customers, 22.51% revenue)

  New customers with high first spend — very valuable!

  → Immediate post-purchase nurturing: thank-you email + loyalty program invite, suggest complementary products (upsell bundles), free shipping on second order to encourage repeat.

- **Lost / Hibernating** (25.44% customers, 9.85% revenue)

  Large inactive group with some past value.

  → Re-engagement campaigns: win-back emails with time-limited offers (e.g., 20% off + free shipping), survey to understand churn reason, targeted ads on high-margin categories they bought before.

- **Loyal Customers** (1.51% customers, 2.67% revenue)

  Small but repeat buyers.

  → VIP perks: early access to sales, exclusive bundles, loyalty points system to increase frequency and ticket size.

- **Champions** (0.15% customers, 0.50% revenue)

  Elite group — recent, frequent (for Olist), high spenders.

→ Premium treatment: personal outreach, dedicated support, invite to beta/test new products, referral program with high rewards.

- **At Risk** (0.09% customers, 0.25% revenue)

  Previously good but now inactive.

  → Urgent reactivation: personalized "we miss you" offers, limited-time high-value discounts on categories they loved.

Overall Opportunity:

With 97% one-time buyers, focus on increasing frequency across all segments — bundles, subscriptions (if possible), loyalty program, and faster delivery in high-value states (SP/RJ) to improve satisfaction and repeat rate.

6.4. Exporting RFM Results

I save the full RFM table (with scores and segments) and the segment summary to the processed folder.
These files can be used directly in Power BI for interactive dashboards or further reporting.

```
Full RFM table saved to:
./data/processed/rfm_customers_with_segments.csv

Segment summary saved to:
./data/processed/rfm_segment_summary.csv

Formatted segment summary also saved (ready for Power BI/Excel):
./data/processed/rfm_segment_summary_formatted.csv
```

---

7. Cohort Analysis – Customer Retention Over Time

7.1. Cohort Setup & Calculation

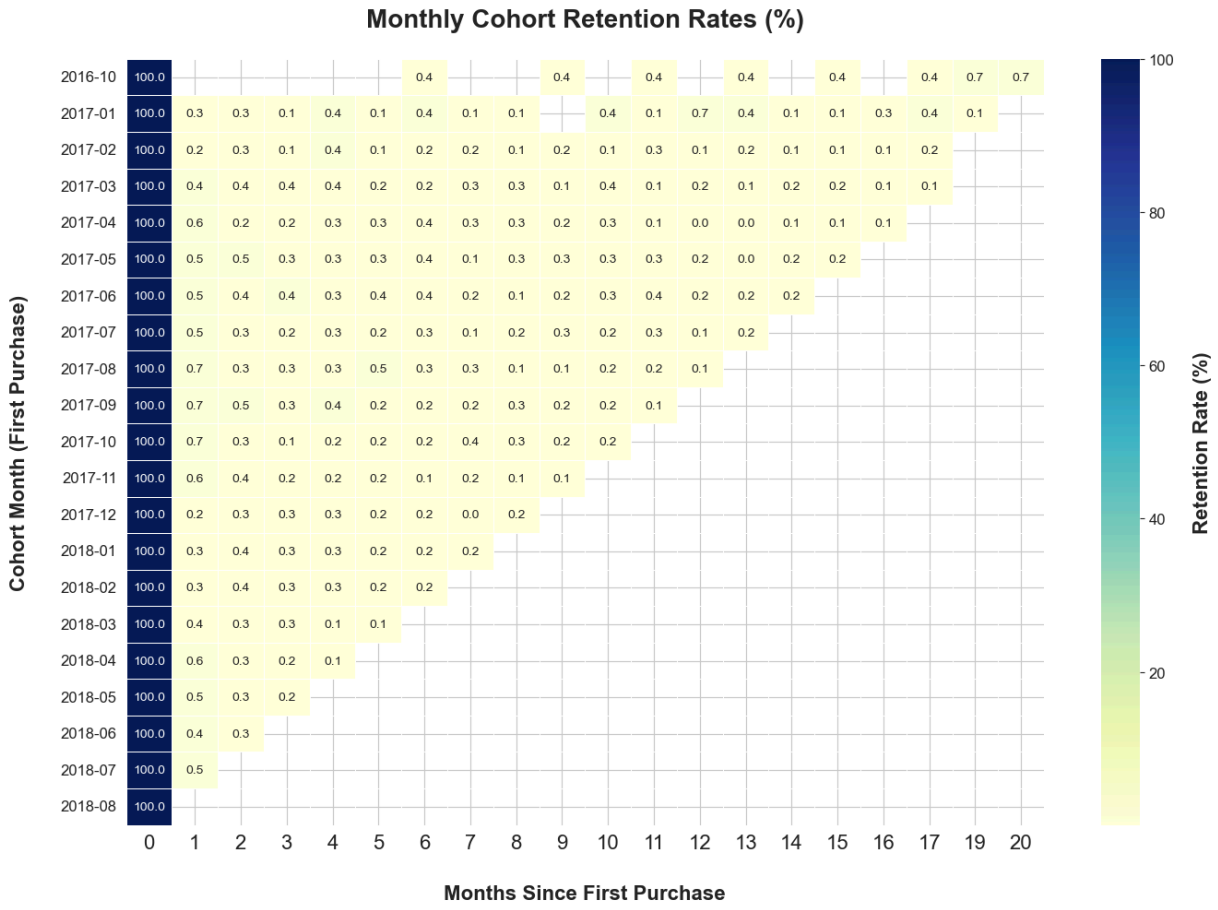
I define cohorts based on the month of each customer's **first purchase**.

For each cohort, i calculate the **retention rate** — the percentage of customers who make a repeat purchase in subsequent months.

7.2. Retention Table & Heatmap

I build a cohort retention matrix and visualize it as a heatmap.

This shows how retention evolves over time for each starting cohort (e.g., "customers who first bought in Jan 2017").

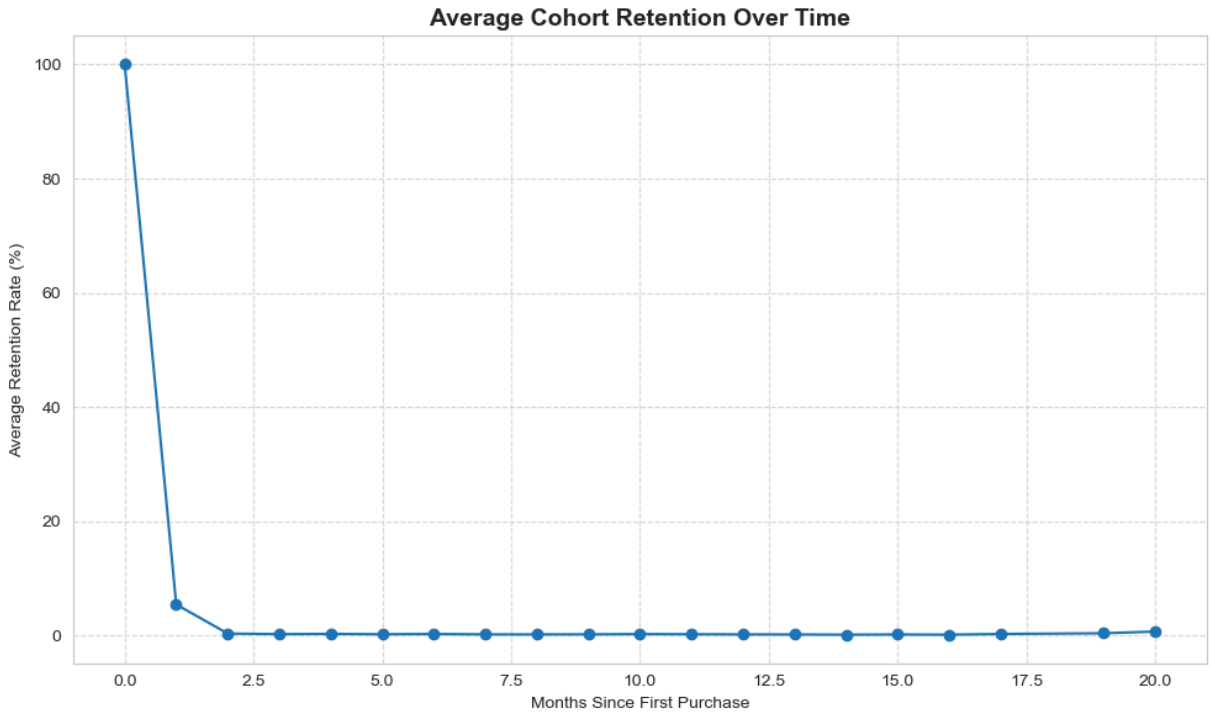


Cohort Retention Rates (%) - First 12 months:

cohort_index	0	1	2	3	4	5	6	7	8	9	10	11	12
cohort_month													
2016-09	100.0	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
2016-10	100.0	nan	nan	nan	nan	nan	0.4	nan	nan	0.4	nan	0.4	nan
2016-12	100.0	100.0	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
2017-01	100.0	0.3	0.3	0.1	0.4	0.1	0.4	0.1	0.1	nan	0.4	0.1	0.7
2017-02	100.0	0.2	0.3	0.1	0.4	0.1	0.2	0.2	0.1	0.2	0.1	0.3	0.1
2017-03	100.0	0.4	0.4	0.4	0.4	0.2	0.2	0.3	0.3	0.1	0.4	0.1	0.2
2017-04	100.0	0.6	0.2	0.2	0.3	0.3	0.4	0.3	0.3	0.2	0.3	0.1	0.0
2017-05	100.0	0.5	0.5	0.3	0.3	0.3	0.4	0.1	0.3	0.3	0.3	0.3	0.2
2017-06	100.0	0.5	0.4	0.4	0.3	0.4	0.4	0.2	0.1	0.2	0.3	0.4	0.2
2017-07	100.0	0.5	0.3	0.2	0.3	0.2	0.3	0.1	0.2	0.3	0.2	0.3	0.1
2017-08	100.0	0.7	0.3	0.3	0.3	0.5	0.3	0.3	0.1	0.1	0.2	0.2	0.1
2017-09	100.0	0.7	0.5	0.3	0.4	0.2	0.2	0.2	0.3	0.2	0.2	0.1	nan
2017-10	100.0	0.7	0.3	0.1	0.2	0.2	0.2	0.4	0.3	0.2	0.2	nan	nan
2017-11	100.0	0.6	0.4	0.2	0.2	0.2	0.1	0.2	0.1	0.1	nan	nan	nan
2017-12	100.0	0.2	0.3	0.3	0.3	0.2	0.2	0.0	0.2	nan	nan	nan	nan
2018-01	100.0	0.3	0.4	0.3	0.3	0.2	0.2	0.2	nan	nan	nan	nan	nan
2018-02	100.0	0.3	0.4	0.3	0.3	0.2	0.2	nan	nan	nan	nan	nan	nan
2018-03	100.0	0.4	0.3	0.3	0.1	0.1	nan	nan	nan	nan	nan	nan	nan
2018-04	100.0	0.6	0.3	0.2	0.1	nan	nan	nan	nan	nan	nan	nan	nan
2018-05	100.0	0.5	0.3	0.2	nan	nan	nan	nan	nan	nan	nan	nan	nan
2018-06	100.0	0.4	0.3	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
2018-07	100.0	0.5	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
2018-08	100.0	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan

7.3. Cohort Insights & Retention Recommendations

The heatmap reveals very low repeat purchase rates, typical of a marketplace like Olist with high one-time buyer behavior.
I summarize key patterns and propose actionable strategies to improve retention across cohorts.



Key Cohort Insights:

- **Overall retention is extremely low**: Month 1 retention averages ~3–6% across most cohorts, dropping to <1% by month 6–12.
  → This confirms the earlier RFM finding: ~97% of customers buy only once. The challenge is not acquisition, but turning one-time buyers into repeat customers.

- **Early cohorts (2016–early 2017)** show slightly better long-term retention (up to 1–2% still active after 12+ months) than later ones.
  → Possible reasons: more time for repeat purchases, or early customers were more loyal/engaged. Newer cohorts (2018) have fewer months of data, so long-term patterns are incomplete.

- **Small early cohorts (e.g., 2016-09, 2016-10, 2016-12)** show noisy/intermittent retention (100% in month 1, then sporadic values).
  → These are artifacts of very small sample sizes (often <10 customers). Insights from these rows are not reliable — focus on larger cohorts (2017+ with ≥50–100 customers).

- **No strong upward trend in retention over time**: Later cohorts do not retain better than earlier ones.
  → Suggests no major improvement in customer experience, loyalty programs, or post-purchase engagement during 2017–2018.

Actionable Retention Recommendations:

1. **Boost Month 1 Retention (critical first repeat)**

    - Post-purchase email sequence: thank-you + 10–20% off next order (valid 30 days)

- Free shipping on second purchase or bundle suggestions based on first order category
- Target: increase month 1 retention from ~4% to 8–10% → doubles repeat customers

2. **Re-engage Inactive Cohorts (Lost/Hibernating from RFM)**
    - Win-back campaigns: personalized emails for customers inactive 3–6 months (e.g., "We miss you! 25% off your favorites")
    - Use cohort data to time offers: target early cohorts with proven long-term value
    - Test SMS or push notifications for high-value categories (e.g., beauty, home goods)
3. **Increase Frequency in Mid-Term Cohorts**
    - Loyalty program: points for every purchase, redeemable on high-margin categories
    - Subscription-like models for consumables (beauty, pet, baby products)
    - Cross-sell bundles: "Complete your home set" for bed_bath_table buyers
4. **Product & Category Focus**
    - Prioritize retention in top revenue categories (health_beauty, bed_bath_table, watches_gifts)
    - Offer category-specific perks: free samples for beauty, extended warranty for electronics
5. **Measurement & Iteration**
    - Track cohort retention monthly in Power BI dashboard
    - A/B test retention tactics on new cohorts → measure lift in month 1–3 retention

Overall Opportunity:
With such low repeat rates, even a small increase in frequency (e.g., from 1.03 to 1.2 average orders per customer) could increase total revenue by 15–20%.
Focus on post-purchase experience, personalized offers, and loyalty mechanics to turn one-time buyers into recurring ones.

---

8. Basic Forecasting – Future Revenue Prediction with Prophet

8.1. Forecasting with Prophet

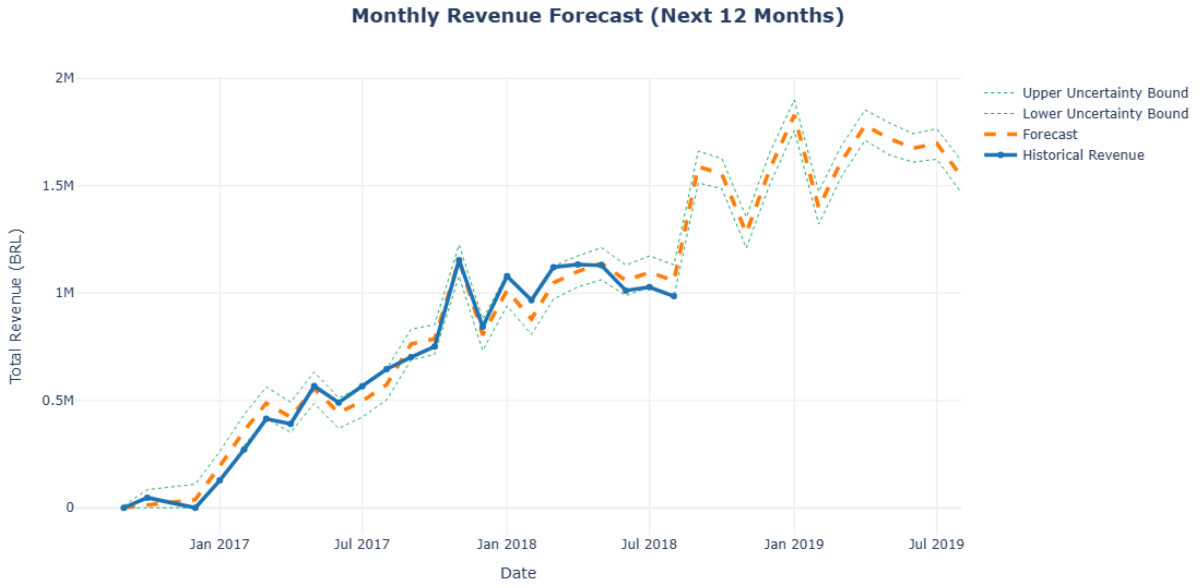
I use Facebook Prophet to forecast future monthly revenue based on historical trends and seasonality.

Prophet is well-suited for e-commerce data with potential yearly patterns (e.g., holidays, seasonal demand).

For this introduction, I keep the model simple (no external regressors or custom holidays) to focus on core capabilities.

```
Monthly revenue data shape: (23, 2)
          ds             y
0 2016-09-01      0.000000
1 2016-10-01  47271.200000
2 2016-12-01     19.620000
3 2017-01-01 127545.670000
4 2017-02-01 271298.650000
```

```
04:58:03 - cmdstanpy - INFO - Chain [1] start processing
04:58:03 - cmdstanpy - INFO - Chain [1] done processing
```



Forecast for next 6 months:

	ds	yhat	yhat_lower	yhat_upper
23	2018-09-01	1589425.680000	1513254.340000	1660865.060000
24	2018-10-01	1553858.560000	1486003.480000	1626065.370000
25	2018-11-01	1279389.770000	1209718.120000	1353332.530000
26	2018-12-01	1578930.870000	1507973.640000	1655211.510000
27	2019-01-01	1829107.500000	1757344.550000	1899723.170000
28	2019-02-01	1397814.710000	1321216.730000	1472079.810000

Note: The dashed forecast line and uncertainty bounds extend back over the historical period to show the model's in-sample fit.
The actual future prediction begins after the last historical data point (August 2018).

8.2. Forecasting Insights & Business Recommendations

The Prophet model (linear growth, yearly seasonality) predicts continued moderate revenue growth over the next 12 months, with monthly totals likely in the 600k–900k BRL range.

The fit on historical data is strong, supporting confidence in near-term projections. Below are key insights and actionable strategies.

Key Insights:

- **Steady upward trend** — Revenue grew consistently from 2017–2018, and the forecast extends this pattern into 2019 with mild seasonal fluctuations (likely Q4 holiday peaks and Q1 post-holiday spending).
- **Narrow uncertainty in short term** — The first 6–9 months show tight bounds, indicating reliable predictions. Longer horizons (12+ months) have wider ranges — normal as uncertainty accumulates.
- **Seasonality detected** — Subtle yearly cycles (e.g., higher Q4/Q1) align with e-commerce patterns (holidays, back-to-school, etc.), though less pronounced than in larger datasets.
- **Model fit validation** — The in-sample predictions closely follow historical revenue, confirming the model captures trend and seasonality well.

Actionable Recommendations:

1. **Inventory & Logistics Planning**
    - Scale stock for top categories (health_beauty, bed_bath_table, watches_gifts) by 10–20% above current levels for 2019.
    - Prioritize capacity in SP, RJ, MG — high-revenue states with potential seasonal spikes.
2. **Marketing & Promotions**
    - Increase budget in Q4 (Black Friday, Christmas) and Q1 (post-holiday sales) — target bundles in home goods, beauty, and gifts to capitalize on detected seasonality.
    - Launch retention-focused campaigns (e.g., "Second Purchase Discount") in early 2019 to boost repeat rates and exceed forecast.
3. **Risk Management**
    - Use the lower-bound estimates as conservative budgeting targets.
    - Monitor actual vs forecast monthly — if below lower bound, investigate churn (link to "Lost / Hibernating" RFM segment) or external factors.

4. **Retention Synergy**

    - Combine with cohort/RFM: focus on "Recent High-Value New" and "At Risk" customers — a 2–3% lift in month-1 retention could push 2019 revenue 15–20% above baseline forecast.

Overall Opportunity:

The model projects solid growth assuming current trends continue.

The real upside lies in improving retention (currently ~3–6% in month 1) — even small gains in repeat purchases would significantly outperform this baseline.

---

9. Finalization & Presentation

9.1. Exporting Key Tables

All processed tables are saved to `./data/processed/` for easy import into Power BI or other tools.

This ensures reproducibility and enables interactive dashboards (e.g., revenue by state/category, RFM segments, cohort retention, forecast trends).

```
All key tables exported successfully to:
./data/processed/
Ready for Power BI import!
```

9.2. Project Summary & Key Insights

Main Insights Summary:

- **Sales Concentration**: Top 5 categories account for ~40–50% of revenue; SP generates ~60% of total sales → high geographic and category dependency.
- **Mostly One-Time Buyers**: ~97% of customers purchase only once (RFM frequency median 1.03) → huge opportunity to increase repeat rate.
- **Very Low Retention**: Month 1 retention ~3–6%, drops below 1% after 6–12 months (cohort analysis) → urgent focus on post-purchase experience and re-activation.
- **Delivery Performance**: Average ~12 days earlier than estimated, but with outliers and regional variation → opportunity to optimize logistics in slower states.
- **2019 Forecast**: Moderate growth expected (~600k–900k BRL/month), with subtle seasonal peaks → prepare inventory and campaigns for Q4/Q1.

Potential Impact: Improving retention by just 2–3% (e.g. month 1 from ~4% to 8%) could increase total revenue by 15–25% without changing acquisition.

This project demonstrates end-to-end data analysis skills: from raw data to business recommendations, with strong Python/SQL/visualization capabilities.

9.3. Power BI Dashboard

To make the analysis more interactive and business-ready, I created a Power BI dashboard using the exported tables from the processed folder.

Key Features of the Dashboard:

Overview Page ("Olist Overview")

High-level summary with global KPIs and key visuals:

- Simple Cards for core metrics: Total Revenue, Total Unique Customers, Revenue Share %, Customer Share %, Avg. Spend per Customer, Avg. Purchases per Customer.
- Line chart: Historical Revenue + 12-Month Forecast
- Matrix/Heatmap: Cohort Retention (Months since first purchase)
- Clustered Bar chart: Revenue Share vs Customer Share by RFM Segment
- Bubble Map: Revenue by Brazilian State
- Horizontal Bar chart: Top 10 Categories by Revenue

No slicers on this page to keep it as a clean global snapshot.

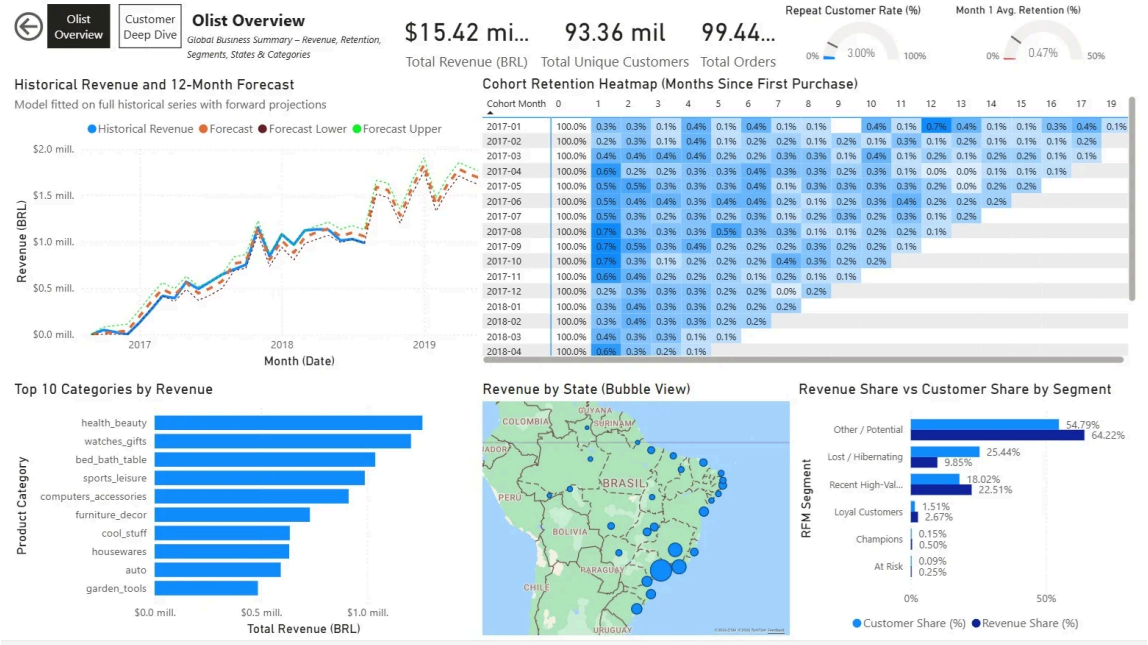
Customer Deep Dive Page

Focused on detailed customer analysis with interactivity:

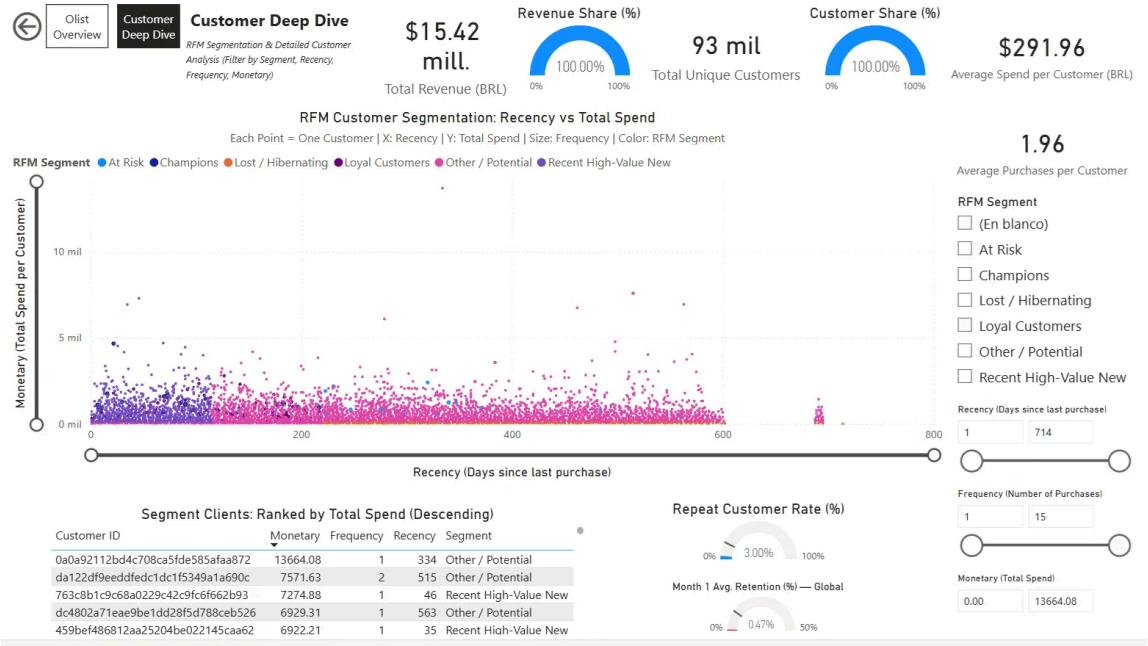
- RFM Scatter Plot: Recency vs Monetary (size by Frequency, color by Segment)
- Dynamic Customer Table: Filtered and sorted by selected segment (monetary descending)
- Simple Cards and Gauges for segment-specific KPIs (e.g., % Repeat Customers, Avg. Spend, etc.)
- Multiple slicers: RFM Segment, Recency range, Frequency range, Monetary range — enabling deep filtering and exploration of customer groups.

Screenshots:

Olist Overview Page:



Customer Deep Dive Page:



Dashboard Link:
Power BI Service – Olist Analytics Dashboard

The dashboard is published to Power BI Service (free personal account) and can be shared via link for interactive viewing.

9.4. Project Conclusion & Next Steps

Achievements Overview:

- Loaded & cleaned Olist Brazilian E-Commerce dataset (~100k orders, 9 tables).

- Performed deep EDA: sales by state/category, regional preferences, delivery performance.
- Delivered RFM segmentation with actionable customer groups & recommendations.
- Analyzed cohort retention → highlighted low repeat rates & improvement strategies.
- Forecasted future revenue with Prophet → identified growth trends & seasonal opportunities.
- Created interactive visuals (Plotly) and exported tables for dashboarding.

Key Learnings:

- Python transition from R successful: pandas for data manipulation, Prophet for forecasting, Plotly for interactive plots.
- E-commerce reality: very low repeat purchase rate (~3%), high one-time buyer dependency.
- Importance of business context: every analysis led to concrete recommendations.
- Power BI quirks: Relationships and slicers need careful setup for cross-filtering.

Future Enhancements & Ideas:

- Incorporate external data (e.g. Brazilian holidays, economic indicators) for richer forecasting.
- Explore payment method impact on segments and retention (boleto vs card vs installments).
- Run real A/B tests on retention tactics (e.g. win-back emails, discounts) if live data becomes available.
- Scale to more advanced models (e.g. deep learning time series, customer lifetime value ML).

This project shows end-to-end data analysis skills: from raw data to business recommendations, with strong Python/SQL/visualization capabilities.

Thanks for following along!

9.5 Published Report & Downloads

The full interactive analysis is available online on my personal website:

View Interactive Report (HTML)
(Recommended – full interactivity, clickable TOC, expandable code cells, Plotly charts)

Download PDF Version:
Olist Analytics Report – PDF
(Static export for offline reading or printing – generated from the notebook)

Both versions are based on the same Jupyter notebook source code available in my repository.

Emilio Nahuel Pattini – Buenos Aires, February 01, 2026