Definition of Terms

Term	Definition
SOW	Statement of Work; a contracted agreement of work to be performed by Contractor for Client.
MVP	Minimum Viable Product; a working baseline deliverable intended to provide necesary functionality to be performant of the requirements outlined within the SOW, also carrying the possibility to inspire future work or enhancement of the current work in later SOWs.
Topline Metrics	The top KPIs being monitored by Client in Looker dashboards. These KPIs consist of Net Sales, Total Orders, AOV, New Customer acquisition and CPA performance.

Scope of MVP

The overall scope of this MVP was to provide visibility into the company's topline metrics over different snapshots of time and at different levels of detail to drive actionable decisions. Moreover, this MVP aimed at providing a New Product Launch template dashboard to allow visibility into new product performance upon launch until maturity. Finally, Contractor agreed to provide a rendering of the Finance Cohort Exports within Looker. The dashboards that housed most of these deliverables are within the Company Report and the New Product Launch dashboards. The Company Report dashboard is a multi-tabbed dashboard designed to give visibility in inceasing detail into the topline metrics.

Table of contents

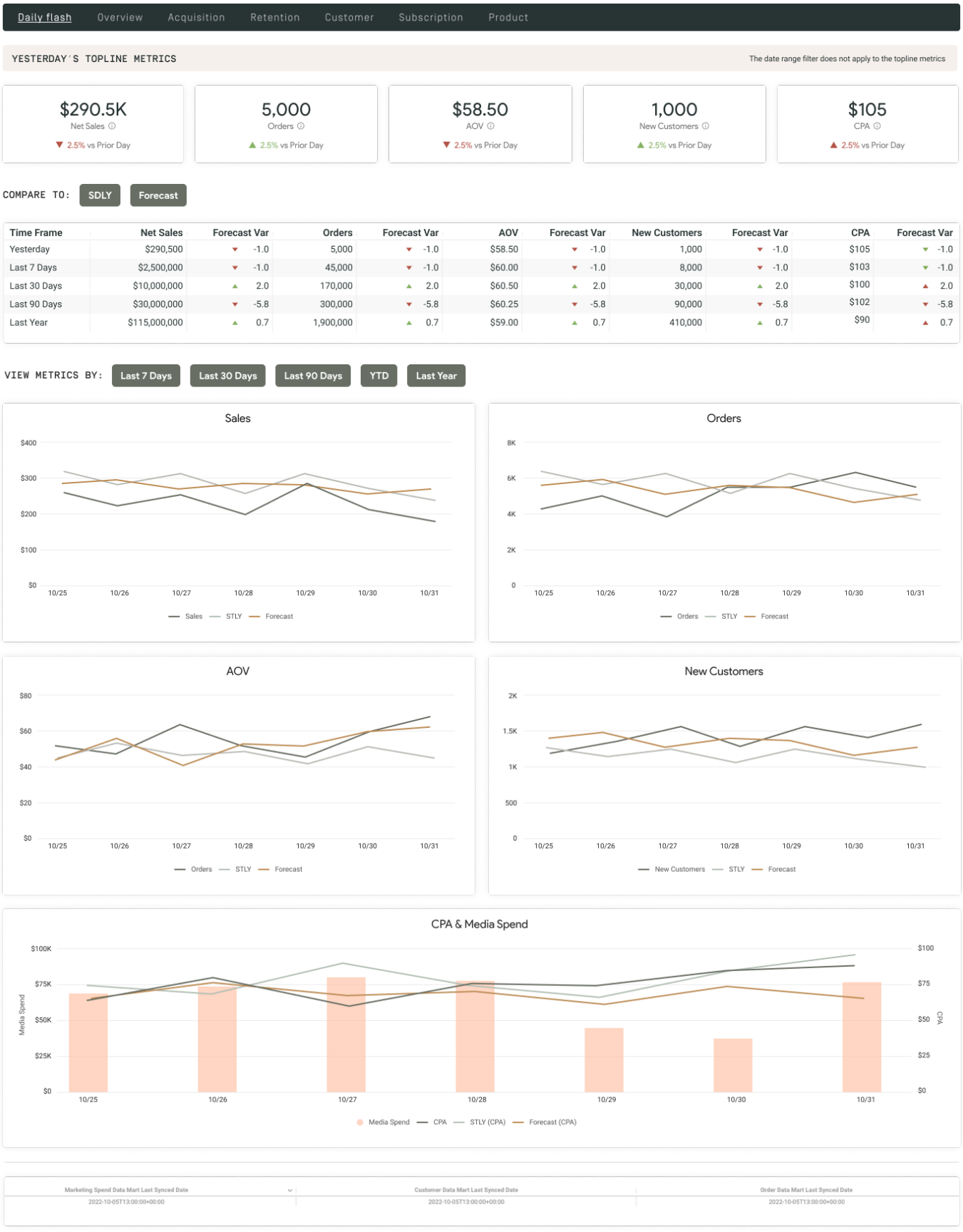
Plan for this SOW

During this SOW, Contractor agreed to deliver the two beginning tabs of the Company Report dashboard (Daily Flash & Overview), the New Product Launch dashboard and the Finance Cohort Export. Below are the wireframes for the dashboards, as well as the previous embodiment of the Finance Cohort Export:

High Fidelity Designs

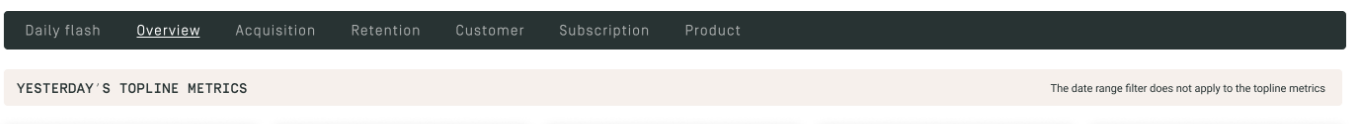
Return to TOC

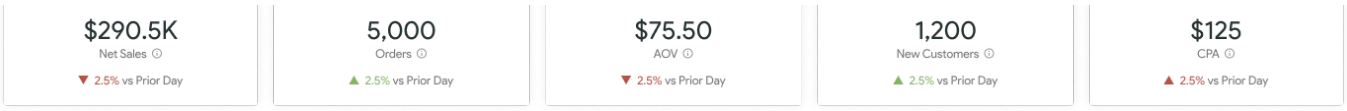
Daily Flash Tab High Fidelity Design



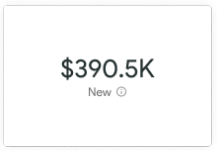
Return to TOC

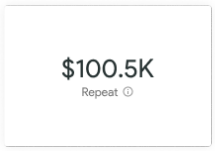
Overview Tab High Fidelity Design

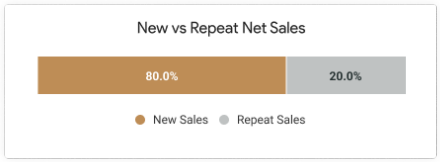


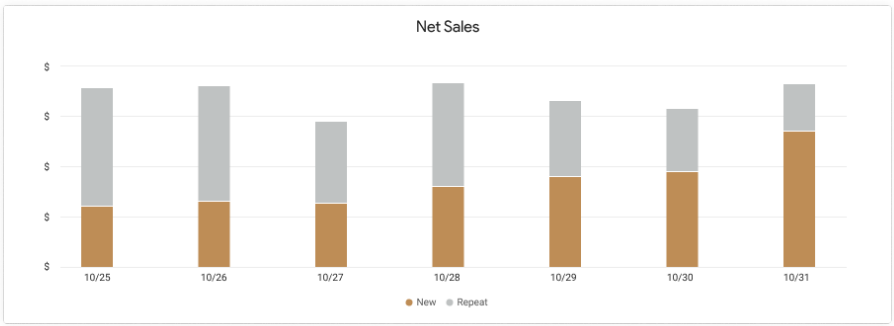


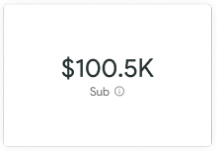
NET SALES

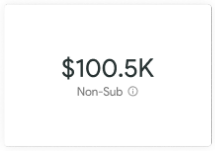


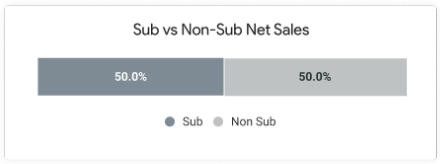


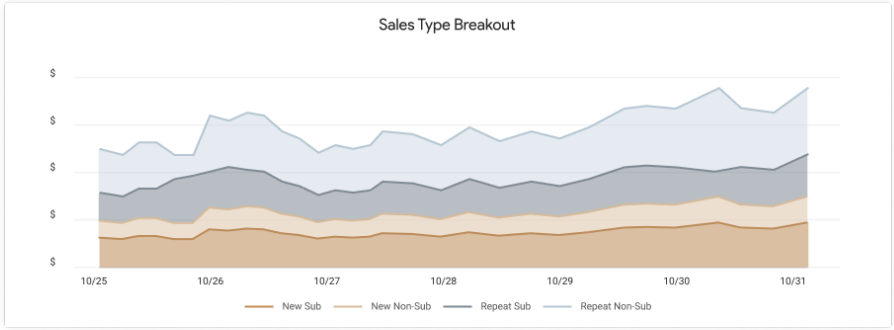


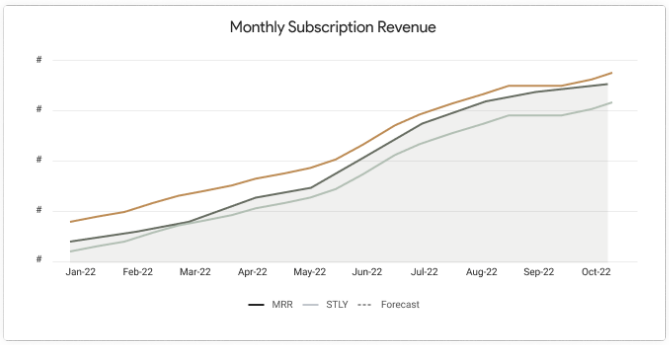


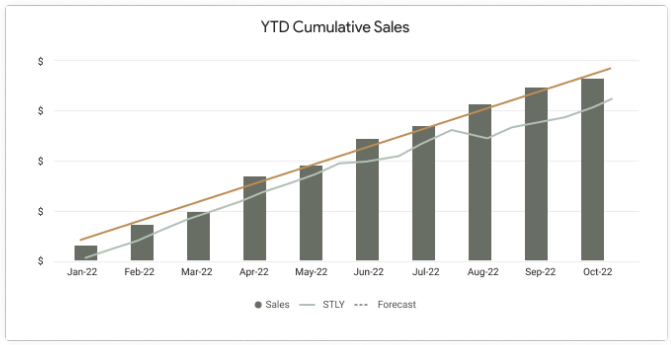




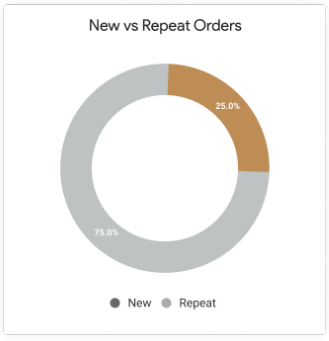


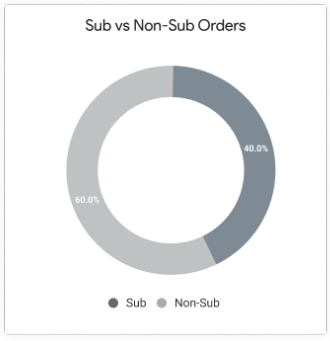


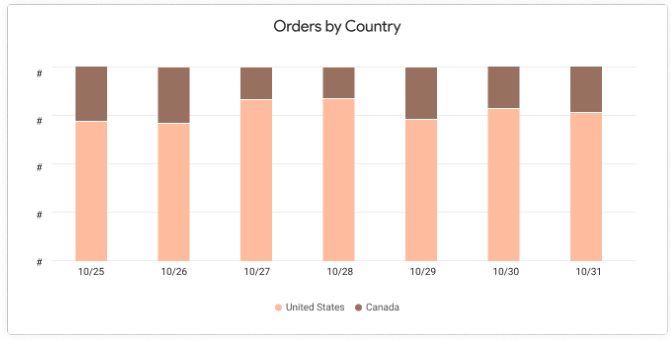


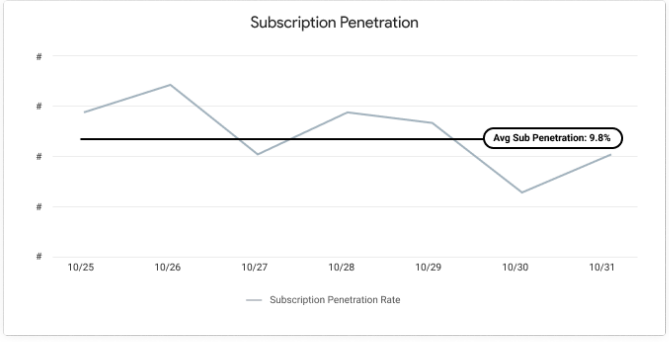


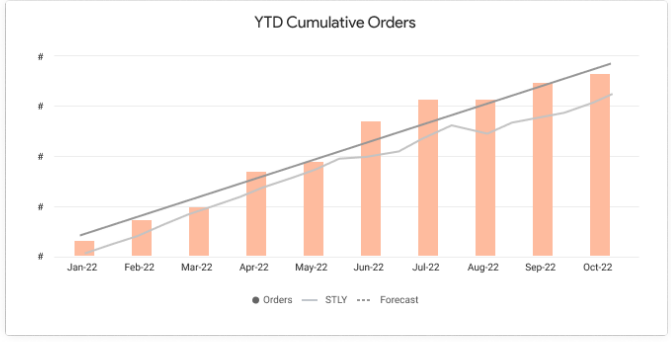
ORDERS

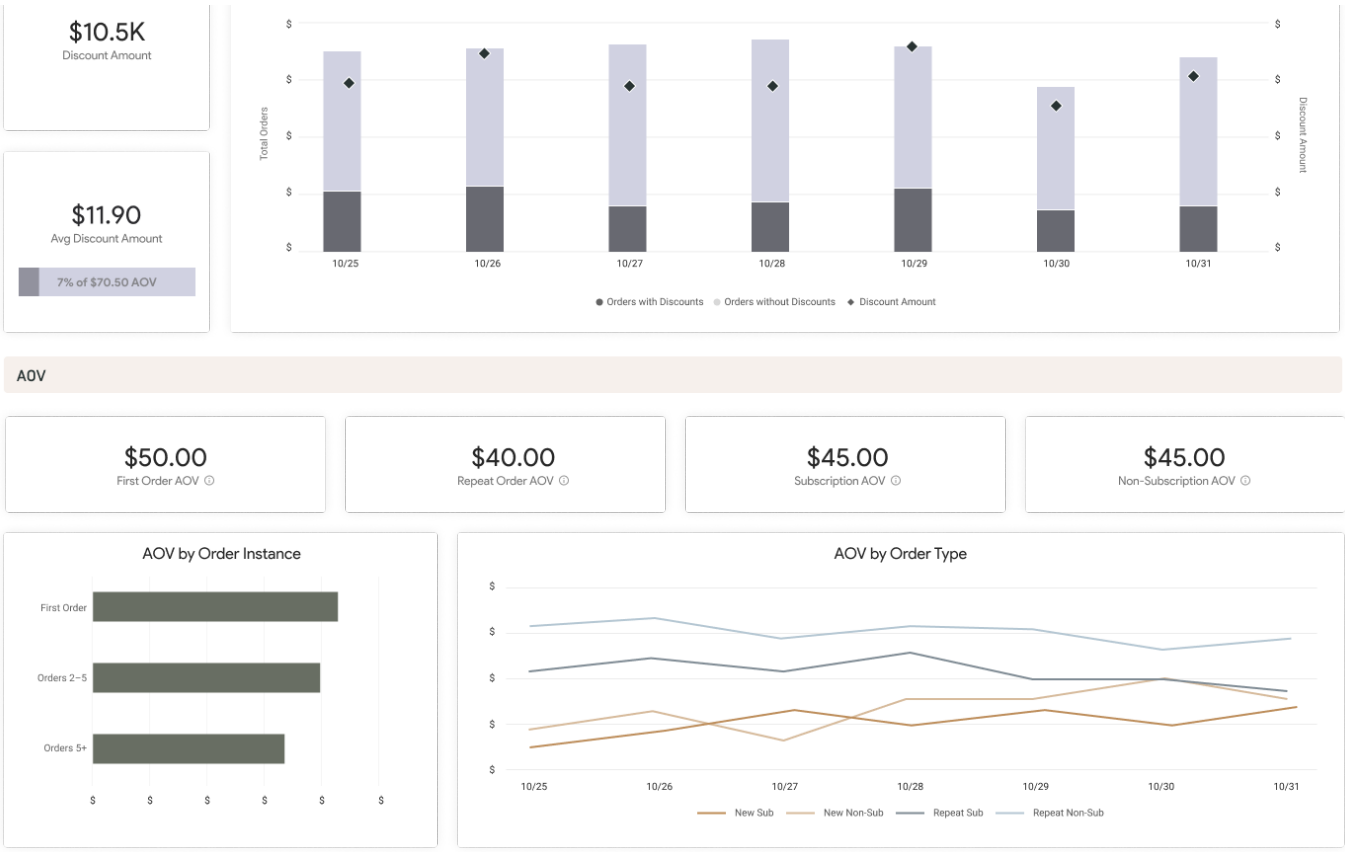






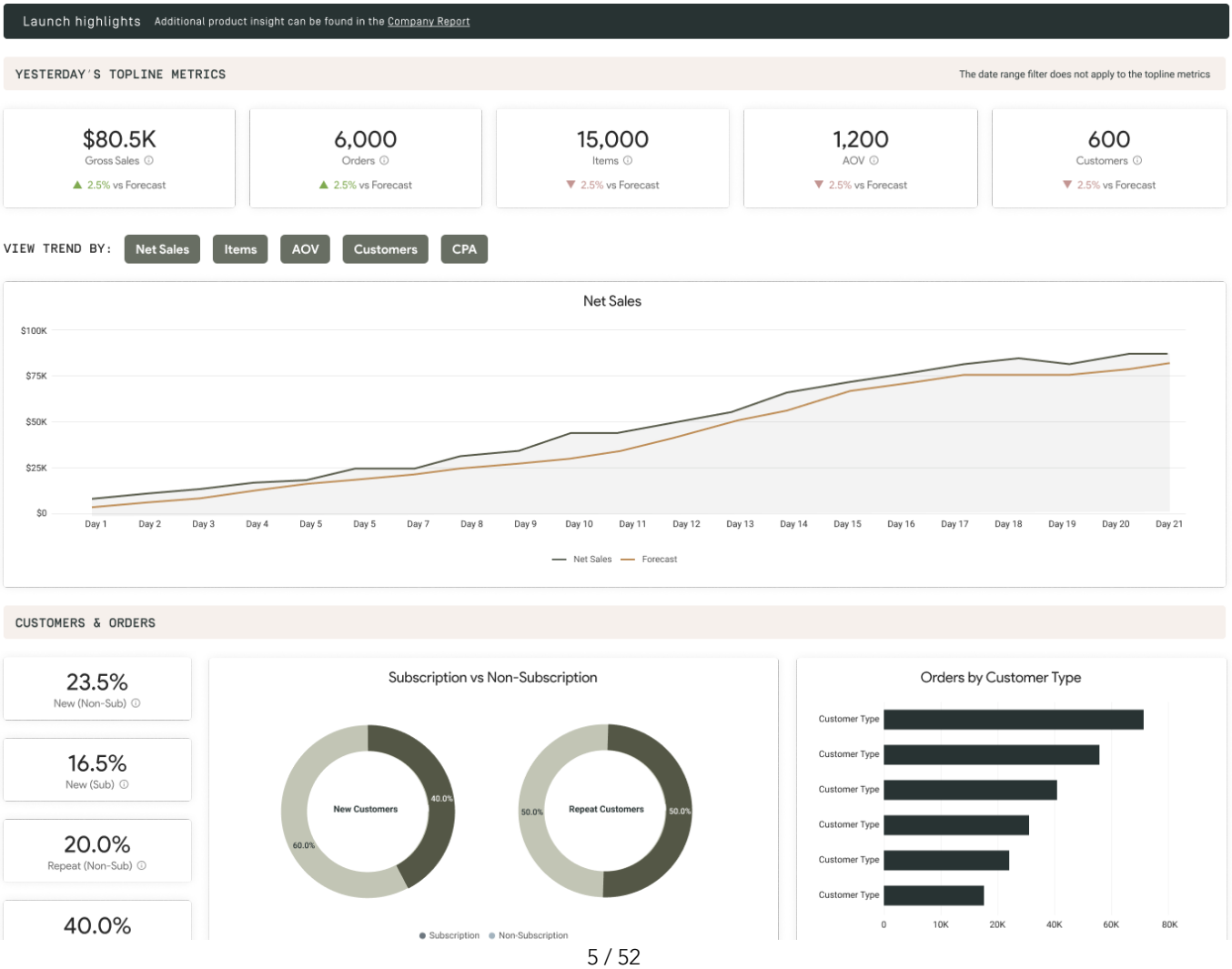


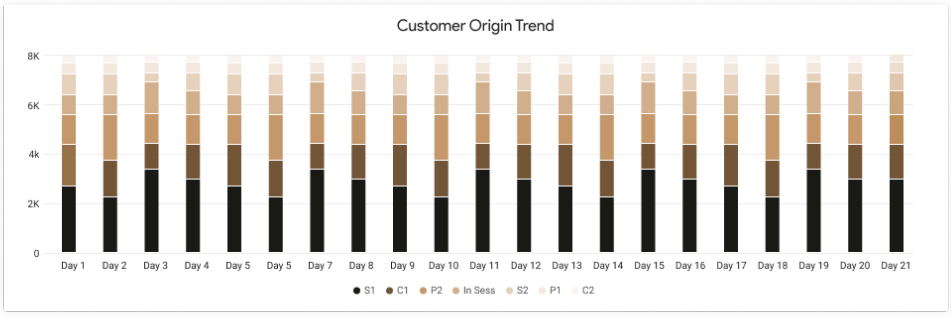


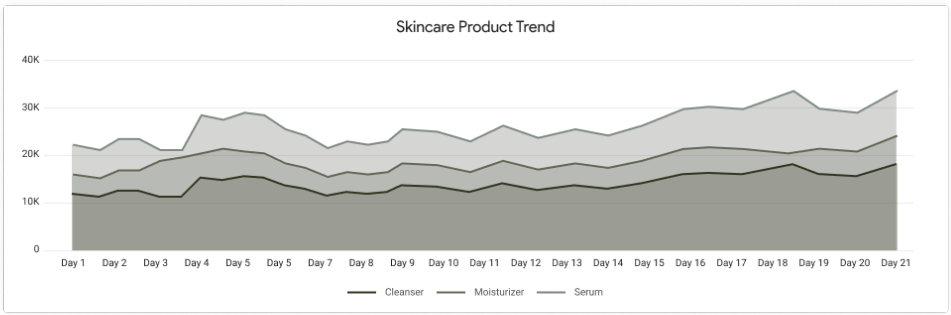


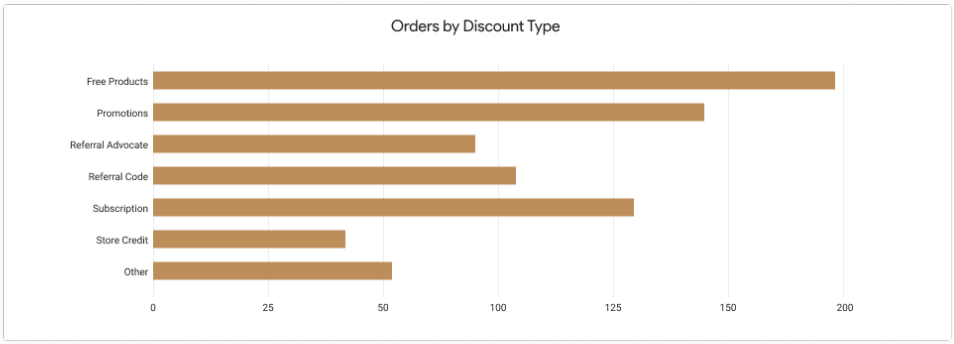
Return to TOC

New Product Launch High Fidelity Design

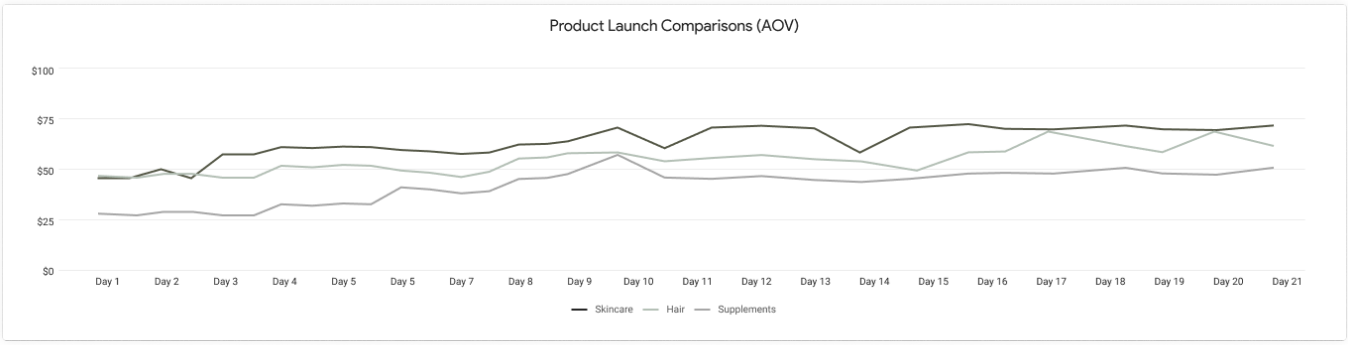


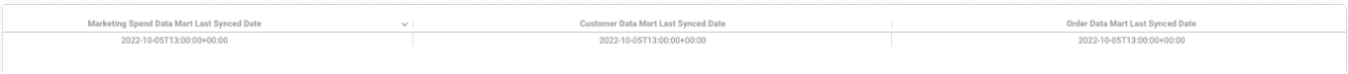






PRODUCT LAUNCH COMPARISON

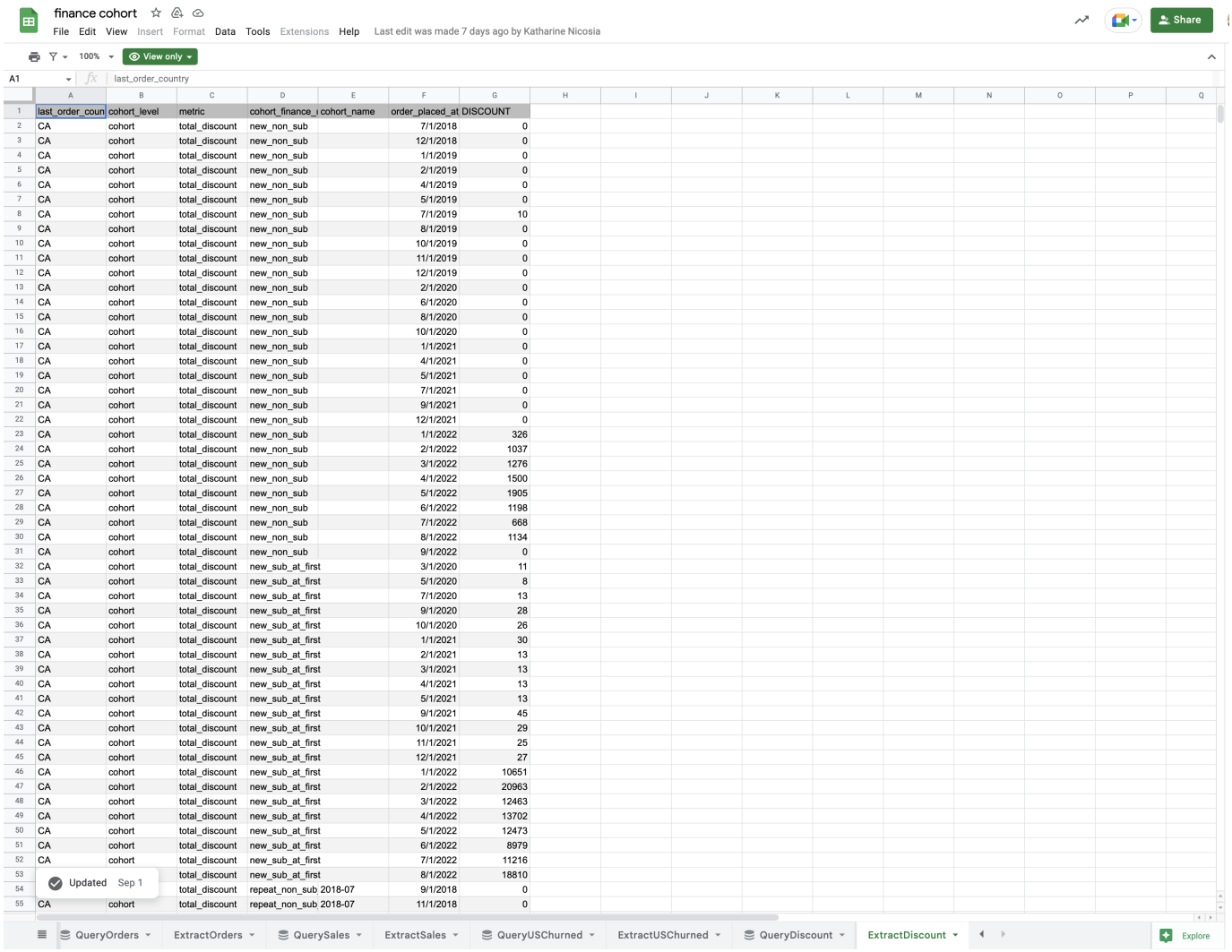




Return to TOC

Finance Cohort Export Original Report

The Finance Cohort Export was originally a multi-tabbed Connected Sheet in Google Sheets. This was refactored and hosted in Looker. Below is a snapshot of the original Connected Sheet:



finance cohort

File Edit View Insert Format Data Tools Extensions Help Last edit was made 7 days ago by Katharine Nicosia

Share

100% View only

A1 last_order_country

	A	B	C	D	E	F	G
1	last_order_coun	cohort_level	metric	cohort_finance_i	cohort_name	order_placed_at	DISCOUNT
2	CA	cohort	total_discount	new_non_sub		7/1/2018	0
3	CA	cohort	total_discount	new_non_sub		12/1/2018	0
4	CA	cohort	total_discount	new_non_sub		1/1/2019	0
5	CA	cohort	total_discount	new_non_sub		2/1/2019	0
6	CA	cohort	total_discount	new_non_sub		4/1/2019	0
7	CA	cohort	total_discount	new_non_sub		5/1/2019	0
8	CA	cohort	total_discount	new_non_sub		7/1/2019	10
9	CA	cohort	total_discount	new_non_sub		8/1/2019	0
10	CA	cohort	total_discount	new_non_sub		10/1/2019	0
11	CA	cohort	total_discount	new_non_sub		11/1/2019	0
12	CA	cohort	total_discount	new_non_sub		12/1/2019	0
13	CA	cohort	total_discount	new_non_sub		2/1/2020	0
14	CA	cohort	total_discount	new_non_sub		6/1/2020	0
15	CA	cohort	total_discount	new_non_sub		8/1/2020	0
16	CA	cohort	total_discount	new_non_sub		10/1/2020	0
17	CA	cohort	total_discount	new_non_sub		1/1/2021	0
18	CA	cohort	total_discount	new_non_sub		4/1/2021	0
19	CA	cohort	total_discount	new_non_sub		5/1/2021	0
20	CA	cohort	total_discount	new_non_sub		7/1/2021	0
21	CA	cohort	total_discount	new_non_sub		9/1/2021	0
22	CA	cohort	total_discount	new_non_sub		12/1/2021	0
23	CA	cohort	total_discount	new_non_sub		1/1/2022	326
24	CA	cohort	total_discount	new_non_sub		2/1/2022	1037
25	CA	cohort	total_discount	new_non_sub		3/1/2022	1276
26	CA	cohort	total_discount	new_non_sub		4/1/2022	1500
27	CA	cohort	total_discount	new_non_sub		5/1/2022	1905
28	CA	cohort	total_discount	new_non_sub		6/1/2022	1198
29	CA	cohort	total_discount	new_non_sub		7/1/2022	668
30	CA	cohort	total_discount	new_non_sub		8/1/2022	1134
31	CA	cohort	total_discount	new_non_sub		9/1/2022	0
32	CA	cohort	total_discount	new_sub_at_first		3/1/2020	11
33	CA	cohort	total_discount	new_sub_at_first		5/1/2020	8
34	CA	cohort	total_discount	new_sub_at_first		7/1/2020	13
35	CA	cohort	total_discount	new_sub_at_first		9/1/2020	28
36	CA	cohort	total_discount	new_sub_at_first		10/1/2020	26
37	CA	cohort	total_discount	new_sub_at_first		1/1/2021	30
38	CA	cohort	total_discount	new_sub_at_first		2/1/2021	13
39	CA	cohort	total_discount	new_sub_at_first		3/1/2021	13
40	CA	cohort	total_discount	new_sub_at_first		4/1/2021	13
41	CA	cohort	total_discount	new_sub_at_first		5/1/2021	13
42	CA	cohort	total_discount	new_sub_at_first		9/1/2021	45
43	CA	cohort	total_discount	new_sub_at_first		10/1/2021	29
44	CA	cohort	total_discount	new_sub_at_first		11/1/2021	25
45	CA	cohort	total_discount	new_sub_at_first		12/1/2021	27
46	CA	cohort	total_discount	new_sub_at_first		1/1/2022	10651
47	CA	cohort	total_discount	new_sub_at_first		2/1/2022	20963
48	CA	cohort	total_discount	new_sub_at_first		3/1/2022	12463
49	CA	cohort	total_discount	new_sub_at_first		4/1/2022	13702
50	CA	cohort	total_discount	new_sub_at_first		5/1/2022	12473
51	CA	cohort	total_discount	new_sub_at_first		6/1/2022	8979
52	CA	cohort	total_discount	new_sub_at_first		7/1/2022	11216
53			total_discount	new_sub_at_first		8/1/2022	18810
54			total_discount	repeat_non_sub	2018-07	9/1/2018	0
55	CA	cohort	total_discount	repeat_non_sub	2018-07	11/1/2018	0

Updated Sep 1

QueryOrders ExtractOrders QuerySales ExtractSales QueryUSChurned ExtractUSChurned QueryDiscount ExtractDiscount Explore

Return to TOC

Overview of Dashboards and Components

The goal of this section is to provide a high-level overview of each dashboard created during SOW#1, providing insight into functionality and origins of each Look.

During this SOW, Contractor agreed to deliver the two beginning tabs of the Company Report dashboard (Daily Flash & Overview), the New Product Launch dashboard and the Finance Cohort Export. Below will be a high-level, technical introduction to each deliverable:

Daily Flash

Return to TOC

The Daily Flash tab of the Company Report is intended to give high-level insight into how Client is performing across its topline metrics. Contents include 'Yesterday's Topline Metrics' and multiple visuals on Net Sales, Orders, AOV, New Customers, CPA, and Media Spend per the date range bucket (last 7/30/90 days, or ytd), shipping country and hair type.

Filters

Below are the current filters employed:

Filter	Scope

Filter	Scope
Trend Line Date Range	Affects all trendline tiles; thus, it does not affect the five topline metric tiles, KPI Performance over Time or Last Synced Dates.
Shipping Country	Affects all tiles except Last Synced Dates.

Explores and Tiles

Below are the involved explores in each Look within the dashboard:

Explore	Tiles
order	Tile #1 Net Sales - [Single Value] Tile #2 Customers - [Single Value] Tile #3 Orders - [Single Value] Tile #4 AOV - [Single Value] Tile #5 Sales - [looker_line] Tile #6 Orders - [looker_line] Tile #7 New Customers - [looker_line] Tile #8 AOV (2) - [looker_line]
marketing_spend_and_orders	Tile #1 CPA & Media Spend - [looker_column] Tile #2 CPA Change - [Single Value]
drv_pdt_daily_flash	Tile #1 KPI Performance over Time - [looker_grid]
last_synced	Tile #1 Last Synced Dates - [looker_grid]

Dashboard Components

Sub-Navigation Pane

At the top of the dashboard, one finds a sub-navigation panel to navigate across the different tabs within the Company Report:

YESTERDAY'S TOPLINE METRICS	The date range filter does not apply to the topline metrics

This tile consists of the following HTML code:

```
<div style="border:solid 0px #000000; border-radius: 4px; padding-top: 10px; padding: 12px;
              background: #283333; height: 100%">
<a href="https://Client.cloud.looker.com/dashboards/14"><img style="padding-left:10px;
padding-right: 36px" src="https://drive.google.com/uc?
export=view&id=1As5iwk2wDa6hSIt18StEazFhV6oPhYaC"  height=25px /></a><a
href="https://Client.cloud.looker.com/dashboards/23"><img style="padding-right:10px;"
src="https://drive.google.com/uc?export=view&id=1CsUTkAZmMBh158b5iiJgvtMzlLk2f44I"
height=25px/></a></div>
<div style="height:10px;"><br></div>
```

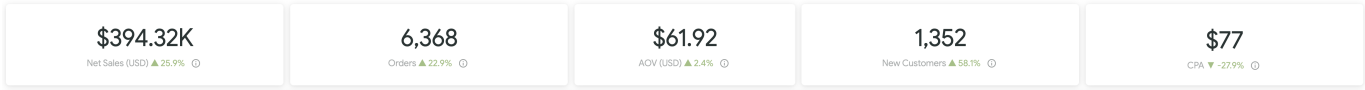
```
<div style="border:solid 0px #000000; border-radius: 4px;
              background: #FBF6F2; padding: 12px;  height: 100%">
<img style="padding-right:20px;" src="https://drive.google.com/uc?
export=view&id=1S2bX8XNdhgHorgUOHfYMWfgVKHul2AX5" height=20px/><span style="float:right;">
    The date range filter does not apply to the topline metrics </span>
```

Return to TOC

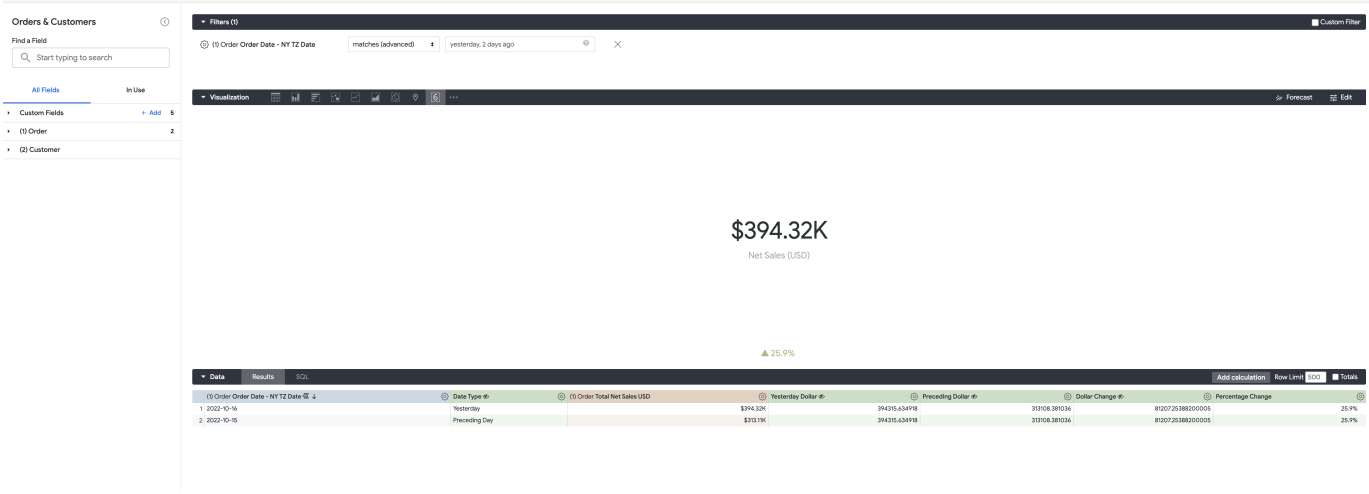
Directly below the sub-navigation tile, one finds the five topline metric single-value visualizations:

The tile renders custom assets housed within the `Client LOOKER STATIC ASSETS` shared Google Drive hosted by Client, as well as makes an HTML clarification of the limits of the date range filter.

Topline Metrics Snapshot



These topline metrics are aggregates of yesterday's performance; the comparisons directly below them compare to aggregates from the preceding day. Below exposes the filters and table calculations involved in generating these aggregates and comparisons:



KPI Performance over Time Crosstab

Below the topline metrics holds the KPI performance over time crosstab. See screenshot below:

KPI Performance over Time

Time Frame	Net Sales	Net Sales Δ	Orders	Orders Δ	Net AOV	Net AOV Δ	New Customers	New Customers Δ	CPA	CPA Δ
Yesterday	$394.32K	▲ 32.9%	6,368	▲ 31.5%	$61.92	▲ 2.1%	1,352	▲ 12.1%	$79	▼ 19.3%
Last 7 Days	$2.28M	▲ 22.0%	38,632	▲ 18.7%	$59.04	▲ 4.0%	6,429	▼ 27.5%	$105	▲ 8.7%
Last 30 Days	$10.17M	▲ 29.5%	169,437	▲ 25.4%	$60.05	▲ 5.5%	29,906	▼ 11.6%	$102	▲ 9.8%
Last 365 Days	$115.92M	▲ 32.5%	1,923,978	▲ 33.0%	$60.25	▼ 0.8%	410,629	▼ 3.0%	$91	▲ 11.9%

This crosstab addresses KPI performance in across different time bins (yesterday, last 7 days, last 30 days and last 365 days). Additionally, the delta between each KPI is calculated against the same KPI's performance in the preceding year. The following were the challenges face in the development of this rollup report:

- Binary limitations in `case when` statements when binning data
- Calculating deltas to compare against same day of the week in the preceding year instead of same date last year
- Performance speed

These challenges were cared for by creating separate dimensions for each time bin for each dimension/measure for the calculation of each field within the Look. Below is a table of each dimension/measure with its corresponding value to the Look:

Field	View	Description

`yesterday`
`last_7_days`
`last_30_days`
`last_365_days`
	`order` & `marketing_spend`	`yesno` dimensions flagging if a date is within the respective time range. These are utilized in fields described later below.

`yesterday_prior_year`
`last_7_days_prior_year`
`last_30_days_prior_year`
`last_365_days_prior_year`
	`order` & `marketing_spend`	`yesno` dimensions used to perform similar tasks to above, except for the same day last year. These are utilized in fields below to calculate the deltas.

`order_pubkey_yesterday`
`order_pubkey_last_7_days`
`order_pubkey_last_30_days`
`order_pubkey_last_365_days`
	`order`	These dimensions care for the count distinct data issue caused by the binary classification employed by a simple `case when` solution; a simple running total will not suffice. Thus, creating separate dimensions for each time bin, with `case when` statements within these dimensions to isolate orders within a given time frame, preserved order pubkeys to accurately perform a count distinct in a later measure.

`customer_pubkey_yesterday`
`customer_pubkey_last_7_days`
`customer_pubkey_last_30_days`
`customer_pubkey_last_365_days`
	`order`	These dimensions perform the same functions as above. They dedicate separate fields for different time bins to accurately care for distinct aggregations upon customer pubkey.

`order_date_yesterday`
`order_date_last_7_days`
`order_date_last_30_days`
`order_date_last_365_days`
	`order`	These dimensions isolate dates within the specified time bin. Within the dimension holds a `case when` statement, returning the order date if within the respective date range, and `null` if not.

`order_date_yesterday_prior_year`
`order_date_last_7_days_prior_year`
`order_date_last_30_days_prior_year`
`order_date_last_365_days_prior_year`
	`order`	Similar to above, these dimensions isolate dates within the specified time bin. Within the dimension holds a `case when` statement, returning the order date if within the respective date range, and `null` if not.

`daily_flash_times`	`order`	Returns 'Yesterday', 'Last 7 Days', 'Last 30 Days' or 'Last 365 Days' strings if the order date bins above fall within the respective time bins.

Field	View	Description
flash_rank	order	A simple case when statement that convertes daily_flash_times to integers to support chronological sort (since daily_flash_times is a string and would therefore sort incorrectly).
total_number_orders_daily_flash	order	count_distinct measure with a case when statement in the sql parameter. In this case when statement, it evaluates which bin type daily_flash_times is returning, then pulling the corresponding order_pubkey dimension to perform a count distinct. Thus, if daily_flash_times = 'Last 30 Days' then order_pubkey_last_30_days is returned.
total_number_orders_yesterday_py total_number_orders_last_7_days_py total_number_orders_last_30_days_py total_number_orders_last_365_days_py	order	count_distinct measures that calculate the prior year values of total orders. These had to be separated into separate measures because the primary dates were already assumed in the daily_flash_times; creating a daily_flash_times_prior_year would double the crosstab size and not allow side-by-side comparison. Thus, separate, filtered measures were created, later being aggregated via a table calculation in the final Look.
distinct_new_customer_count_daily_flash	order	Same logic for total_number_orders_daily_flash, different pubkey returns (customer_pubkey return instead of order_pubkey).
distinct_new_customer_count_yesterday_py distinct_new_customer_count_last_7_days_py distinct_new_customer_count_last_30_days_py distinct_new_customer_count_last_365_days_py	order	Same logic for total_number_orders_[time_bin]_py measures above, except calculating the total number of new distinct customers instead of distinct order counts.
total_net_sales_USD_running_total	order	A simple running total upon total_net_sales_USD. This cares for the binary classification issue originally grappled with the daily_flash_times.

Field	View	Description
total_net_sales_USD_yesterday_py total_net_sales_USD_last_7_days_py total_net_sales_USD_last_30_days_py total_net_sales_USD_last_365_days_py	order	sum_distinct measures that calculate the prior year values of total net sales. These had to be separated into separate measures because the primary dates were already assumed in the daily_flash_times; creating a daily_flash_times_prior_year would double the crosstab size and not allow side-by-side comparison. Thus, separate, filtered measures were created, later being aggregated via a table calculation in the final Look.
total_media_spend_running_total	marketing_spend	A simple running total upon total_media_spend. This cares for the binary classification issue originally grappled with the daily_flash_times.
total_media_spend_yesterday_py total_media_spend_last_7_days_py total_media_spend_last_30_days_py total_media_spend_last_365_days_py	marketing_spend	sum measures that calculate the prior year values of total marketing spend. These had to be separated into separate measures because the primary dates were already assumed in the daily_flash_times; creating a daily_flash_times_prior_year would double the crosstab size and not allow side-by-side comparison. Thus, separate, filtered measures were created, later being aggregated via a table calculation in the final Look.
cpa_yesterday_py cpa_last_7_days_py cpa_last_30_days_py cpa_last_365_days_py	marketing_spend	Measures that calculate the prior year values of CPA. These had to be separated into separate measures because the primary dates were already assumed in the daily_flash_times; creating a daily_flash_times_prior_year would double the crosstab size and not allow side-by-side comparison. Thus, separate, filtered measures were created, later being aggregated via a table calculation in the final Look.

Return to TOC

While these dimensions were validated and finalized, they were employed within the crosstab, along with some table calculations to calculate the deltas. However, performance issues were encountered that made the Look run upwards towards 10 minutes. To care for this, we created a PDT (drv_pdt_daily_flash) that is managed by the datagroup_orders datagroup, which has a max cache age of 24 hours and a sql_trigger of

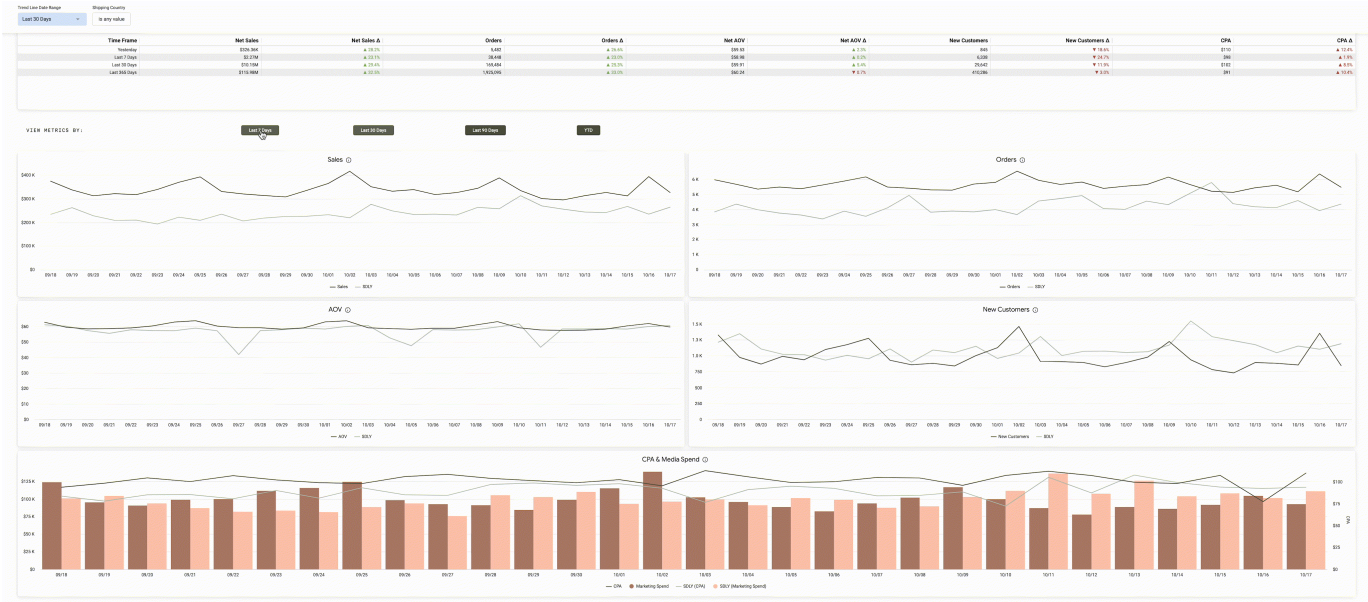
```
SELECT max(order_pubkey) FROM
`Clienthair.Client_bi_datamart_growth_customer_ltv_customer_order.order
```

with an `interval_trigger` of 12 hours.

After the PDT, the crosstab was refactored to point towards `drv_pdt_daily_flash`, and it is significantly more performant.

Topline Metric Trendlines

Below the KPI crosstab is the trend lines for the five topline metrics over time. The default filter for these is to show the last 30 days, and one may toggle between alternate views to view the trends by the last 7 days, the last 30 days, the last 90 days or YTD. Below is a demo of the UX functionality:



In order to allow for dynamic date range renderings in comparison with the prior year, a number of custom dimensions and measures had to be employed. Below is a list of all the involved fields, as well as descriptions of their utility:

Field	View	Description
`select_timeframe`	`order`	A parameter with values of 'Last 7 Days', 'Last 30 Days', 'Last 90 Days' and 'YTD'. This parameter is manipulated when buttons are clicked on the dashboard to change the date range and date formatting.
`current_vs_previous_period`	`order`	A dynamic dimension that allows for comparison between current year and last year. This dimension employs a warpper of liquid logic around a `case when` statement, and it is heavily dependent on the value of `select_timeframe`.
`selected_dynamic_day_of`	`order`	A dynamic dimension used to change the date formatting depending upon the value for `select_timeframe`.
`selected_dynamic_day_of_sort`	`order`	A dynamic dimension used to determine sort order based upon `select_timeframe` to enforce chronological sorting. This could be an opportunity for simplification.
`selected_dynamic_duration`	`order`	Returns the week index or month number based on the selection of `select_timeframe`; this could be an opportunity for deprecation.

Overview

Return to TOC

The Overview tab is intended to provide a level more detail than the Daily Flash tab. The date filter affects all trend lines similar to the dynamic trend line date filter in the Daily Flash tab. This dashboard displays Yesterday's Topline Metrics and multiple visuals on Net Sales, Orders, and AOV per the date range bucket (last 7/30/90 days, or ytd), shipping country and hair type.

The looks mainly employs simple aggregations that Client pre-supplied on view/data mart creation. However, each Look (besides the topline metrics) uses the `current_vs_previous_period` dimensions to filter by date. Some Looks incorporate prior year comparison; others do not. The ones that do not compare the metrics to prior year have "Previous" filtered out. is in the use of the Sales Type Breakout & Monthly Subscription Revenue area charts, as well as the Cumulative Sales column/line combination chart.

Filters

Below are the current filters employed:

Filter	Scope
Date Range	This filter applies to all tiles except the five 'Yesterday's Topline Metrics' tiles and the Last Synced Dates tile.
Shipping Country	Applies to all tiles except the Last Synced Dates tile.
Hair Type	Applies to all tiles except the Last Synced Dates tile.

Explores and Tiles

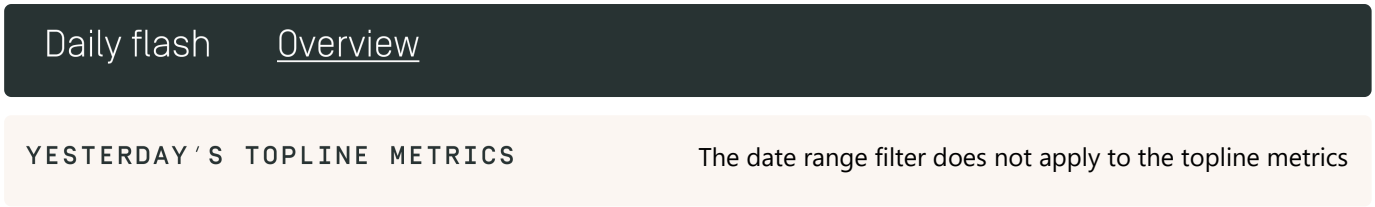
Below are the involved explores in each Look within the dashboard:

Explore	Tiles

Explore	Tiles
order	Tile #1 Net Sales - [Single Value] Tile #2 Customers - [Single Value] Tile #3 Orders - [Single Value] Tile #4 AOV - [Single Value] Tile #5 Net Sales (2) - [looker_column] Tile #6 Sales Type Breakout - [looker_area] Tile #7 New vs Repeat Net Sales - [looker_bar] Tile #8 Sub vs Non-Sub Net Sales - [looker_bar] Tile #9 Orders by Country - [looker_column] Tile #10 New vs Repeat - [looker_donut_multiples] Tile #11 Sub vs Non-Sub - [looker_donut_multiples] Tile #12 Total Discount Amount - [single_value] Tile #13 Average Discount Value - [single_value] Tile #14 Discount Trend - [looker_column] Tile #15 AOV by Order Type - [looker_line] Tile #16 Sub Sales - [single_value] Tile #17 First Order AOV (Copy 2) - [single_value] Tile #18 First Order AOV - [single_value] Tile #19 First Order AOV (Copy) - [single_value] Tile #20 First Order AOV (Copy 3) - [single_value] Tile #22 Cumulative Orders - [looker_column] Tile #24 Cumulative Sales - [looker_line] Tile #25 New Sales - [single_value] Tile #26 Repeat Sales - [single_value] Tile #27 Non-Sub Sales - [single_value Tile #28 AOV by Order Instance - [looker_bar]
marketing_spend_and_orders	Tile #1 CPA Change - [Single Value]
last_synced	Tile #1 Last Synced Dates - [looker_grid]

Sub-Navigation Pane

The Sub-Navigation pane is similar to the Daily Flash tab, with the exception being that the "Overview" is underlined to signal selection as opposed to "Daily Flash". The rendering and HTML are below:



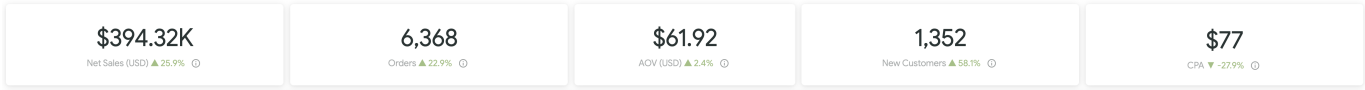
YESTERDAY'S TOPLINE METRICS The date range filter does not apply to the topline metrics

```
<div style="border:solid 0px #000000; border-radius: 4px; padding: 10px 12px;
              background: #283333; height: 100%">
<a href="https://Client.cloud.looker.com/dashboards/14"><img style="padding-right:36px;
padding-left:10px;" src="https://drive.google.com/uc?export=view&id=1v6NGGMwRHwbN1R1-
_4QkBEEFKMSVKWEG" height=25px /></a><a
href="https://Client.cloud.looker.com/dashboards/23"><img style="padding-right:36px; "
src="https://drive.google.com/uc?export=view&id=1FnSy-ZdqBDOJ552izgYdqVAn9yg6K_MC"
height=25px/></a></div>
```

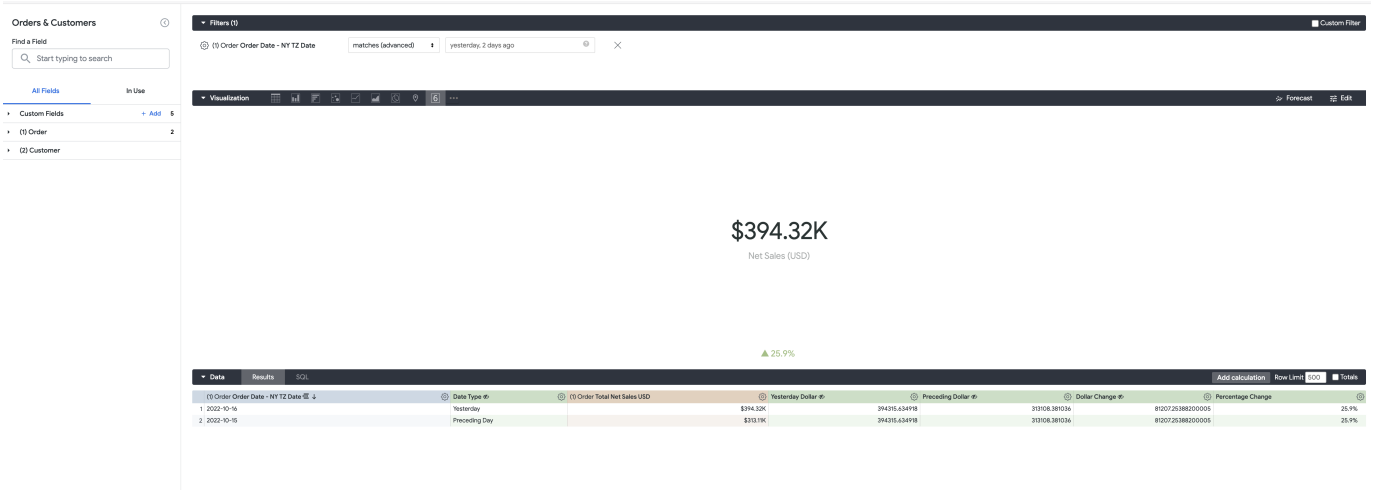
```
<div style="height:10px;"><br></div>
<div style="border:solid 0px #000000; border-radius: 4px;
            background: #FBF6F2; padding: 12px;  height: 100%">
<img style="padding-right:20px;" src="https://drive.google.com/uc?
export=view&id=1S2bX8XNdhgHorgUOHfYMWfgVKHul2AX5" height=20px/><span style="float:right;">
    The date range filter does not apply to the topline metrics </span>
```

Topline Metrics Snapshot

Directly below the sub-navigation tile, one finds the five topline metric single-value visualizations:



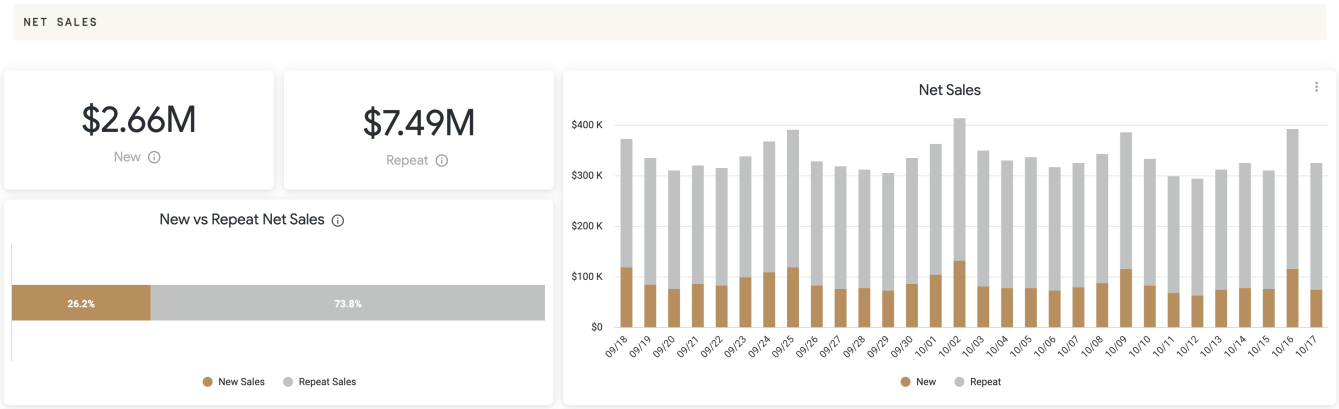
These topline metrics are aggregates of yesterday's performance; the comparisons directly below them compare to aggregates from the preceding day. Below exposes the filters and table calculations involved in generating these aggregates and comparisons:



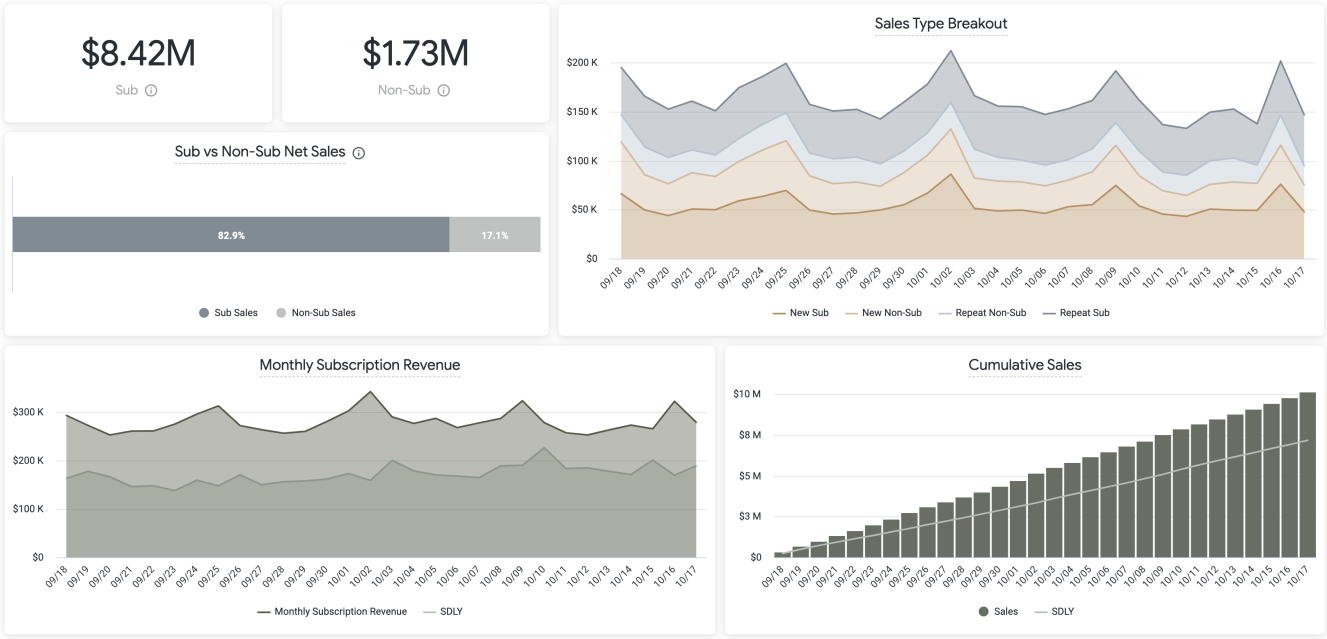
Net Sales Section

This section comprises of two perspectives of net sales data: new vs repeat & subscription vs non-subscription. The following is a snapshot and description of the new vs repeat section:

New vs Repeat

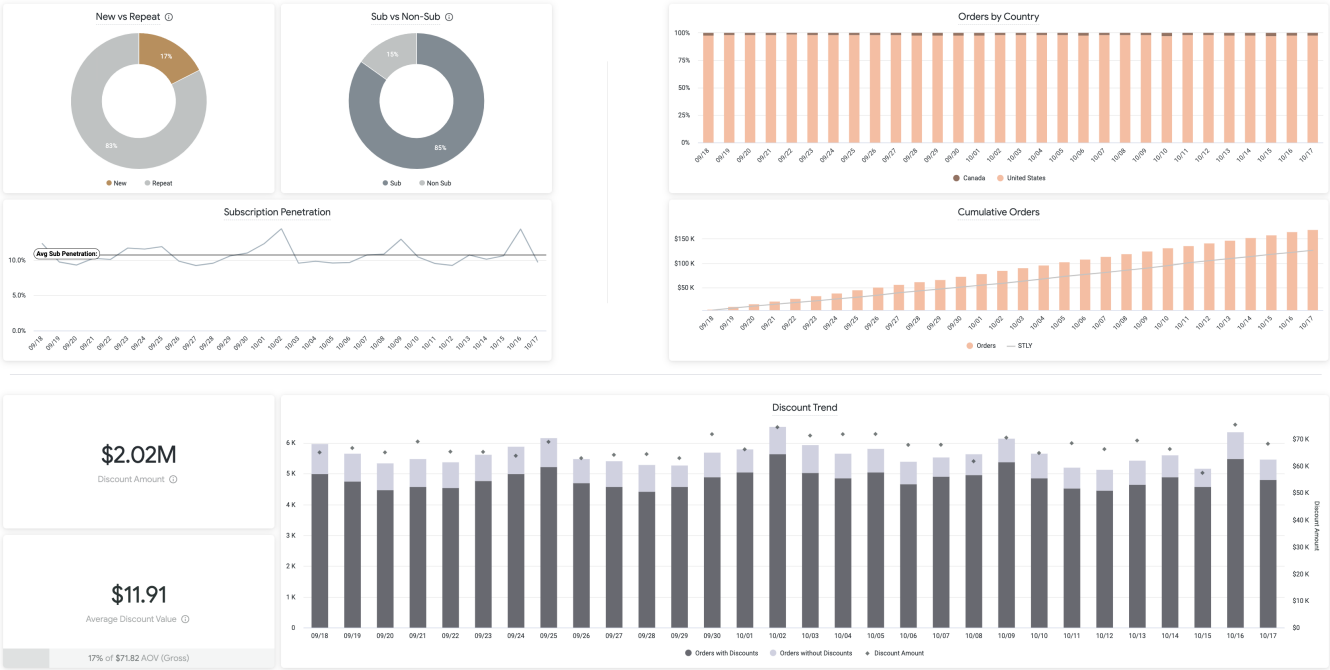


Subscription vs Non-Subscription



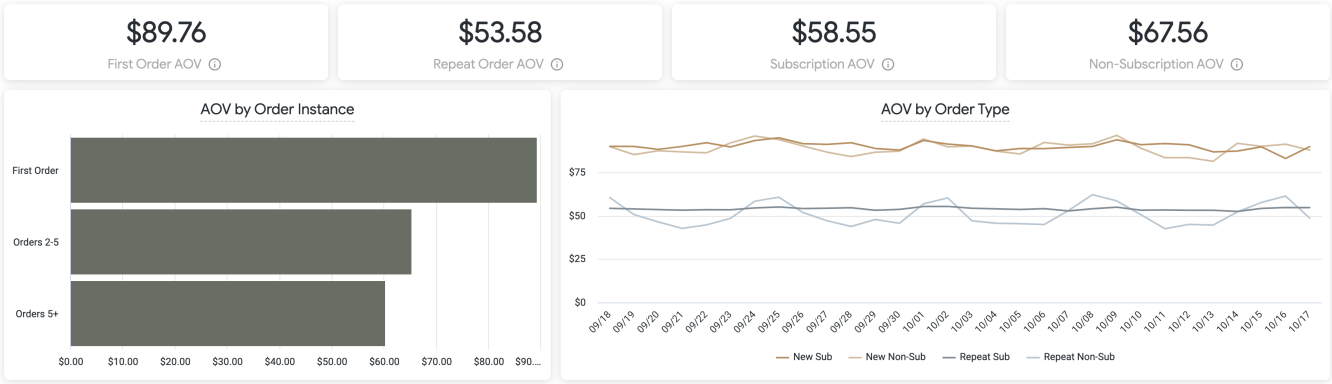
Orders

The Orders section addresses order driven metrics. It provides insight into number and types of orders, along with discount data. Below is a snapshot of the section:



AOV

The AOV section provides insight into various AOV contexts: first order, repeat order, subscription, non-subscription, as well as AOV by order tier.Below is a snapshot of the section:



New Product Launch

Return to TOC

The New Product Launch Dashboard provides a "Days Since Launch" perspective on the selected new product. The Days since launch parameter provides 30/60/90/1 yr/ and unlimited options for selection. Components required to support this parameter are detailed at the end of this section.

The looks mainly employs simple aggregations that Client pre-supplied on view/data mart creation. However, each Look (besides the topline metrics) uses the `current_vs_previous_period` dimensions to filter by date. Some Looks incorporate prior year comparison; others do not. The ones that do not compare the metrics to prior year have "Previous" filtered out. is in the use of the Sales Type Breakout & Monthly Subscription Revenue area charts, as well as the Cumulative Sales column/line combination chart.

Days Since Launch Components

In order to support the requirement for a variety of 'days since launch' options and correspondingly adjust the 'tick density' on the x-axis of the impacted trend charts, number of dimension were created to complement the 'select days since launch parameter. These fields are described in the table below, with hyperlinks on the fields to LookML snippets detailing their construction:

Field	View	Description
`launch_date`	`drv_launch_date_by_product`	A dimension group based on min(order_date) by product_name used to calculate launch duration.
`launch`	`order_items`	A duration dimension group surfacing number of days since launch based on launch_date as start and order_date as duration end.
`select_days_since_launch`	`order_items`	The parameter for providing options for 30,60,90, 1 yr, and 'no limit' time frame options for days since launch
`selected_dynamic_launch_days_label`	`order_items`	A dynamic dimension used to aggregate tick density from daily to 7 or 30 day increments based on the selected parameter value.

Field	View	Description
selected_dynamic_launch_days_duration	order_items	This dimension is used to translate the sring parameter value to a number for the custom filter below. Values returned are 31,61,91, 366, or 9999.
[custom filter](#custom filter)	NA - specific tiles	A custom filter was applied on the impacted tiles= to translate the parameter to the requisite filter.

Filters

Below are the current filters employed:

Filter	Scope
Days Since Launch	All Trend Tiles: Sales Trend (with measure substitution for Items, AOV, and Customers) Customer Origin Trend Product Trend - Gross Sales % Discount on Orders Orders by Discount Type Prod Launch Comparisons (AOV)
Shipping Country	All tiles except Last Synced Dates tile.
Hair Type	All tiles except Last Synced Dates tile.
Measure Selector	Sales Trend tile
New Products	All tiles except Last Synced Dates tile.

Explores and Tiles

Below are the involved explores in each Look within the dashboard:

Explore	Tiles

Explore	Tiles
order_items	Tile #1 Gross Sales - [Single Value] Tile #2 Orders - [Single Value] Tile #3 Items - [Single Value] Tile #4 AOV - [Single Value] Tile #5 New Customers - [Single Value] Tile #6 Sales Trend - [looker_line] Tile #7 New(Non-Sub) - [Single Value] Tile #8 New(Sub) - [Single Value] Tile #9 Repeat(Non-Sub) - [Single Value] Tile #10 Repeat(Sub) - [Single Value] Tile #11 Subscription vs Non-Subscription - [Donut Multiples] Tile #12 Customer Origin - [looker_bar] Tile #13 Customer Origin Trend - [looker_column_stacked] Tile #14 Subscription Order Composition - [Packed Bubble] Tile #15 Product Trend (Gross Sales) - [looker_line] Tile #16 Product Launch Comparison Trend - [looker_line]
order	Tile #1 Orders by Customer Type - [looker_bar] Tile #2 % of Discount on Orders - [Donut Multiples] Tile #3 Orders by Discount Type - [looker_bar]
customer	Tile #1 Customer Origin - [looker_bar] Tile #2 Customer Origin Trend - [looker_column_stacked]
cohort_finance_name	Tile #1 Orders by Customer Type - [looker_bar]
last_synced	Tile #1 Last Synced Dates - [looker_grid]

Sub-Navigation Pane

The Sub-Navigation pane provides a hyperlink to the Company Report.

Launch highlights Additional product insight can be found in the Company Report

YESTERDAY'S TOPLINE METRICS The date range filter does not apply to the topline metrics

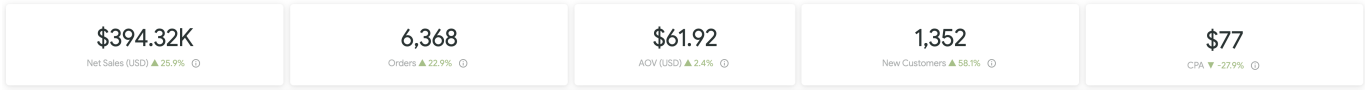
This tile consists of the following html code:

```
<div style="border:solid 0px #000000; border-radius: 4px; padding: 10px 12px;
              background: #283333; height: 100%">
<img style="padding-right:10px; padding-left:10px;" src="https://drive.google.com/uc?
export=view&id=16jfZ2JT6cqh5TaFLJR3NoKXgYfPih_UX" height=25px><img style="padding-
right:0px; " src="https://drive.google.com/uc?
export=view&id=1MZol1GYgLcQkQrKdSoH8myfdRCbpcLu7" height=25px/><a
href="https://Client.cloud.looker.com/dashboards/14"><img style="padding-left:0px; "
src="https://drive.google.com/uc?export=view&id=16eNl4T-I1f5423iKY2b-py2P04QDogMK"
height=25px/></a></div>
<div style="height:10px;"><br></div>
<div style="border:solid 0px #000000; border-radius: 4px;
              background: #FBF6F2; padding: 12px;  height: 100%">
<div><img style="padding-right:20px;" src="https://drive.google.com/uc?
```

```
export=view&id=1S2bX8XNdhgHorgUOHfYMWfgVKHul2AX5" height=20px/>
<span style="float:right;">     The date range filter does not apply to
the topline metrics </span> </div>
```

Topline Metrics Snapshot

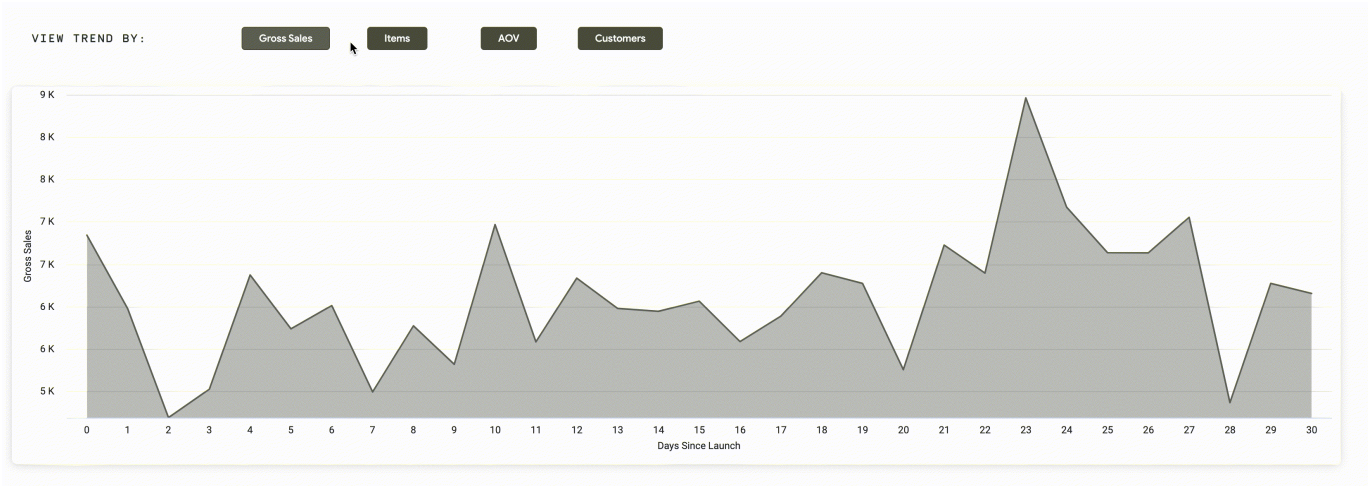
Directly below the sub-navigation tile, one finds the five topline metric single-value visualizations:



These topline metrics are aggregates of yesterday's performance.

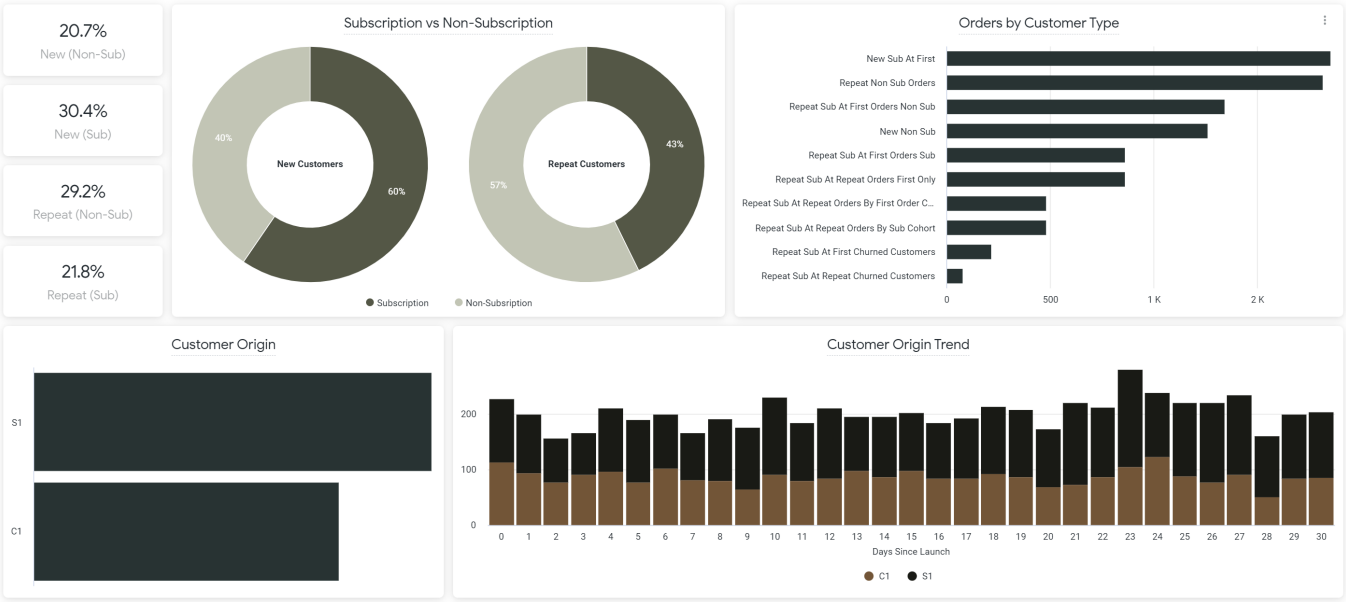
View Trend By:

This section provides the option to select a trend view of one of four measures determined by the button selector. The timespan is controlled by the days since launch parameter selected in the filter section. The x-axis tick density is dynamic, changing from daily, 7-day, and 30-day increments based on the selected timespan. The following is a snapshot of this section with Gross Sales selected:

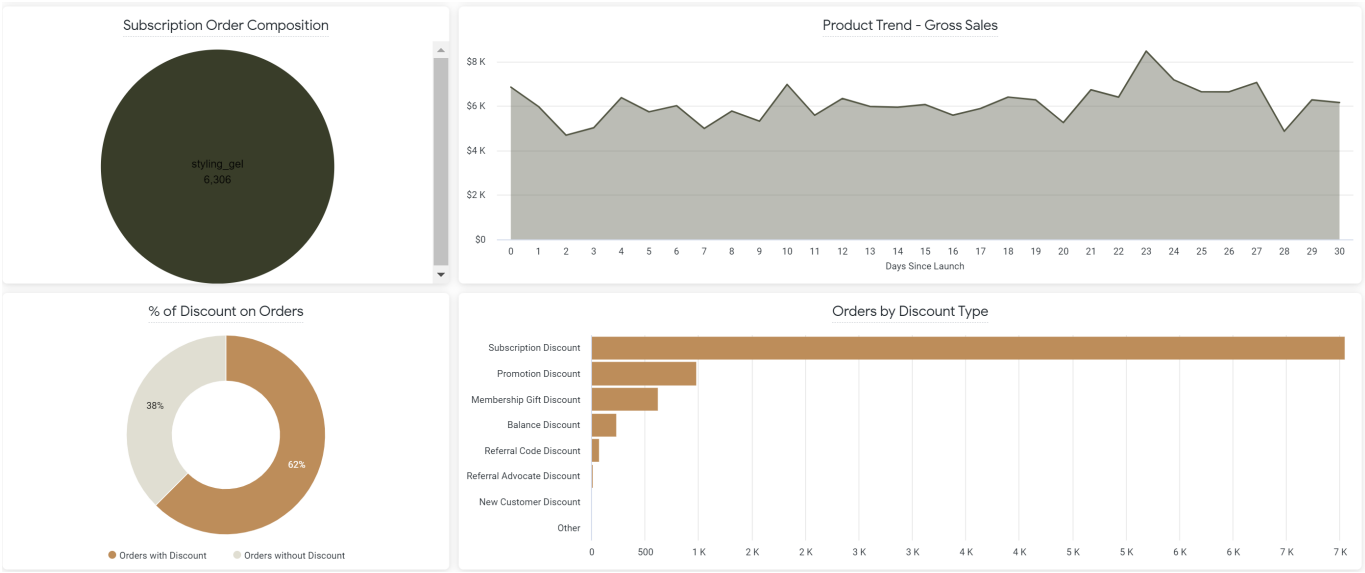


Customers & Orders

This section provides visualizations on New Subscribers vs New Non-Subscribers, a Subscription vs Non-Subscription breakout of new vs repeat customers, a visualization of order counts by customer type

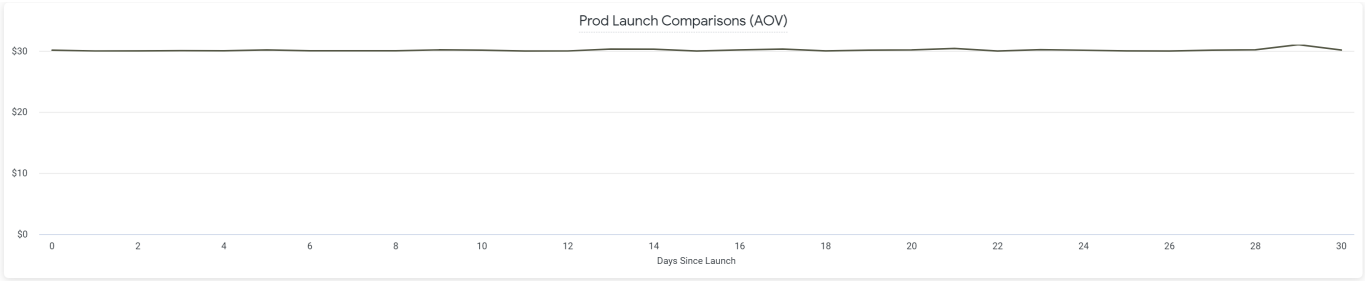


Customer orders second section



Product Launch Comparison

This section is to provide a product component comparison by AOV across the selected days since launch timespan:



Finance Cohort Looks

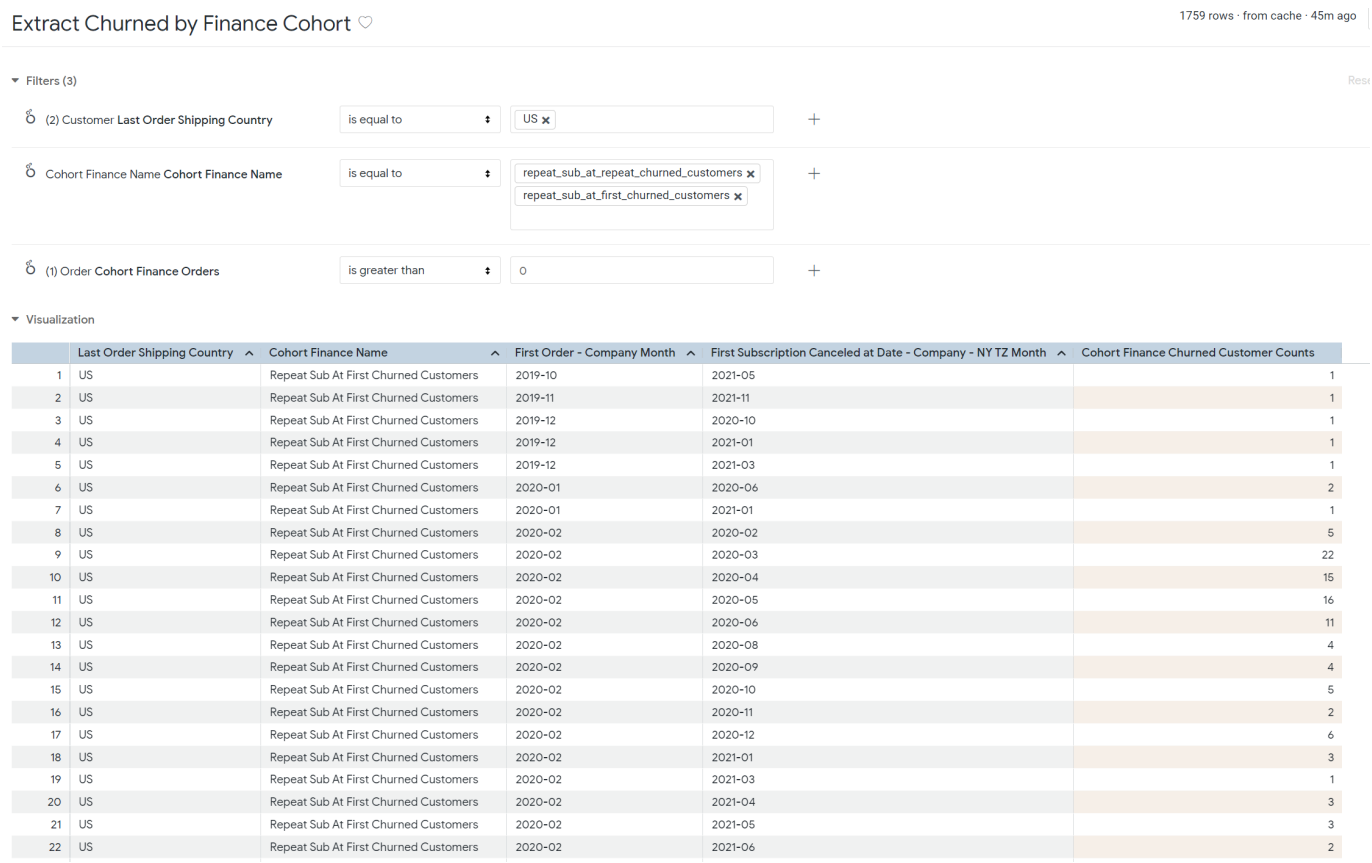
Return to TOC

Five Looks were produced to support the 'Finance Cohort Name' reporting requirement. The Looks themselves are straightforward, though some have large result sets that will require elevated user permissions for large record set downloads.

Three views were required to support these looks and are detailed in the view section of this document. Listed here for convenience/context.

View	Description
`cohort_finance_name`	This view is used to generate a list of 16 [`cohort_finance_names`](#Cohort Finance Name) as identified in the requirements from Finance. Corresponding filtered measures are mapped to these names via case statements in the order_cohort_extension and orer_items_cohort_extension views.
`order_cohort_extension`	This view extends the order view, adding cohort_finance_name from the required join to the cohort_finance_names view. This is required to accomplish the cohort_name mapping to the appropriate measure. Example of this mapping [`here.`](#Cohort Finance Units)The net sales and discount measures are redefined in this view, as sum_distinct is required as measure type due to the required cross join to cohort_finance_name.
`order_items_cohort_extension`	This view extends the order_items view, adding cohort_finance_name from the required join to the cohort_finance_names view required to accomplish the cohort_name mapping at product level to the appropriate measure. Measure duplication was not required for this view in that the measures are limited to type count_distinct.

Screen grabs of the five Looks are provided below:



Extract Churned by Finance Cohort ♡

1759 rows · from cache · 45m ago

Filters (3)

(2) Customer **Last Order Shipping Country** | is equal to | US

Cohort Finance Name **Cohort Finance Name** | is equal to | repeat_sub_at_repeat_churned_customers, repeat_sub_at_first_churned_customers

(1) Order **Cohort Finance Orders** | is greater than | 0

Visualization

	Last Order Shipping Country	Cohort Finance Name	First Order - Company Month	First Subscription Canceled at Date - Company - NY TZ Month	Cohort Finance Churned Customer Counts
1	US	Repeat Sub At First Churned Customers	2019-10	2021-05	1
2	US	Repeat Sub At First Churned Customers	2019-11	2021-11	1
3	US	Repeat Sub At First Churned Customers	2019-12	2020-10	1
4	US	Repeat Sub At First Churned Customers	2019-12	2021-01	1
5	US	Repeat Sub At First Churned Customers	2019-12	2021-03	1
6	US	Repeat Sub At First Churned Customers	2020-01	2020-06	2
7	US	Repeat Sub At First Churned Customers	2020-01	2021-01	1
8	US	Repeat Sub At First Churned Customers	2020-02	2020-02	5
9	US	Repeat Sub At First Churned Customers	2020-02	2020-03	22
10	US	Repeat Sub At First Churned Customers	2020-02	2020-04	15
11	US	Repeat Sub At First Churned Customers	2020-02	2020-05	16
12	US	Repeat Sub At First Churned Customers	2020-02	2020-06	11
13	US	Repeat Sub At First Churned Customers	2020-02	2020-08	4
14	US	Repeat Sub At First Churned Customers	2020-02	2020-09	4
15	US	Repeat Sub At First Churned Customers	2020-02	2020-10	5
16	US	Repeat Sub At First Churned Customers	2020-02	2020-11	2
17	US	Repeat Sub At First Churned Customers	2020-02	2020-12	6
18	US	Repeat Sub At First Churned Customers	2020-02	2021-01	3
19	US	Repeat Sub At First Churned Customers	2020-02	2021-03	1
20	US	Repeat Sub At First Churned Customers	2020-02	2021-04	3
21	US	Repeat Sub At First Churned Customers	2020-02	2021-05	3
22	US	Repeat Sub At First Churned Customers	2020-02	2021-06	2

Extract Discounts by Finance Cohort ♡

▾ Filters (3)

(2) Customer **Last Order Shipping Country**	is equal to	US ×	+
Cohort Finance Name **Cohort Finance Name**	is equal to		+
(1) Order **Cohort Finance Discounts**	is greater than	0	+

▾ Visualization

	Last Order Shipping Country	Cohort Finance Name	First Order - Company Month	Order Date - NY TZ Month	Cohort Finance Discounts
1	US	Repeat Sub At Repeat Orders First Only	2020-02	2022-08	$1,551
2	US	Repeat Sub At Repeat Orders First Only	2021-05	2022-05	$38,075
3	US	Repeat Sub At Repeat Orders First Only	2020-11	2021-10	$22,925
4	US	Repeat Sub At Repeat Orders First Only	2021-12	2022-07	$48,719
5	US	Repeat Sub At Repeat Orders First Only	2021-06	2022-08	$24,629
6	US	Repeat Sub At Repeat Orders First Only	2020-04	2020-09	$27,283
7	US	Repeat Sub At Repeat Orders First Only	2020-06	2020-06	$1,389
8	US	Repeat Sub At Repeat Orders First Only	2021-12	2022-03	$98,789
9	US	Repeat Sub At Repeat Orders First Only	2022-06	2022-06	$6,156
10	US	Repeat Sub At Repeat Orders First Only	2020-06	2021-05	$15,036
11	US	Repeat Sub At Repeat Orders First Only	2019-11	2020-05	$33
12	US	Repeat Sub At Repeat Orders First Only	2021-06	2021-07	$53,624
13	US	Repeat Sub At Repeat Orders First Only	2021-06	2022-06	$26,832
14	US	Repeat Sub At Repeat Orders First Only	2020-07	2021-07	$16,199
15	US	Repeat Sub At Repeat Orders First Only	2020-05	2021-09	$8,924
16	US	Repeat Sub At Repeat Orders First Only	2020-06	2021-10	$10,198
17	US	Repeat Sub At Repeat Orders First Only	2020-02	2021-05	$1,989
18	US	Repeat Sub At Repeat Orders First Only	2021-08	2022-04	$34,052
19	US	Repeat Sub At Repeat Orders First Only	2020-03	2021-07	$7,310
20	US	Repeat Sub At Repeat Orders First Only	2021-07	2021-10	$60,895
21	US	Repeat Sub At Repeat Orders First Only	2021-04	2021-07	$89,448
22	US	Repeat Sub At Repeat Orders First Only	2020-05	2022-10	$4,418
23	US	Repeat Sub At Repeat Orders First Only	2021-03	2022-05	$37,580
24	US	Repeat Sub At Repeat Orders First Only	2019-12	2021-04	$93

Extract Orders by Finance Cohort

Filters (3)

(2) Customer **Last Order Shipping Country**	is equal to	US, CA
Cohort Finance Name **Cohort Finance Name**	is equal to	
(1) Order **Cohort Finance Orders**	is greater than	0

Visualization

	Last Order Shipping Country	Cohort Finance Name	First Order - Company Month	Order Date - NY TZ Month	Cohort Finance Orders
1	CA	New Non Sub	2018-07	2018-07	1
2	CA	New Non Sub	2018-10	2018-10	2
3	CA	New Non Sub	2018-12	2018-12	1
4	CA	New Non Sub	2019-01	2019-01	1
5	CA	New Non Sub	2019-02	2019-02	1
6	CA	New Non Sub	2019-04	2019-04	1
7	CA	New Non Sub	2019-05	2019-05	1
8	CA	New Non Sub	2019-07	2019-07	1
9	CA	New Non Sub	2019-08	2019-08	1
10	CA	New Non Sub	2019-10	2019-10	2
11	CA	New Non Sub	2019-11	2019-11	1
12	CA	New Non Sub	2019-12	2019-12	1
13	CA	New Non Sub	2020-01	2020-01	1
14	CA	New Non Sub	2020-02	2020-02	2
15	CA	New Non Sub	2020-04	2020-04	1
16	CA	New Non Sub	2020-06	2020-06	1
17	CA	New Non Sub	2020-08	2020-08	1
18	CA	New Non Sub	2020-10	2020-10	1
19	CA	New Non Sub	2021-01	2021-01	1
20	CA	New Non Sub	2021-03	2021-03	1
21	CA	New Non Sub	2021-04	2021-04	4
22	CA	New Non Sub	2021-05	2021-05	2
23	CA	New Non Sub	2021-07	2021-07	2
24	CA	New Non Sub	2021-08	2021-08	1
25	CA	New Non Sub	2021-09	2021-09	1
26	CA	New Non Sub	2021-12	2021-12	2
27	CA	New Non Sub	2022-01	2022-01	740

Extract Units by Finance Cohort by Product

2536 rows · 1s · 22m ago Run Edit

Filters (3) Reset Look | Explore from Here

(2) Customer Last Order Shipping Country	is equal to	US
Cohort Finance Name Cohort Finance Name	is equal to	new_non_sub, new_sub_at_first, repeat_sub_at_repeat_orders_by_first_orde..., repeat_sub_at_repeat_orders_first_only
(1) Order Cohort Finance Units	is greater than	0

Visualization

	Last Order Shipping Country	Order Date - NY TZ Month	Cohort Finance Name	Product Name	Cohort Finance Units
1	US	2022-10	Repeat Sub At Repeat Orders First Only	Brush Agave	9
2	US	2022-10	Repeat Sub At Repeat Orders First Only	Brush Boar	19
3	US	2022-10	Repeat Sub At Repeat Orders First Only	Brush Detangling	24
4	US	2022-10	Repeat Sub At Repeat Orders First Only	Brush Mixed	69
5	US	2022-10	Repeat Sub At Repeat Orders First Only	Complete Set	13
6	US	2022-10	Repeat Sub At Repeat Orders First Only	Conditioner	41,188
7	US	2022-10	Repeat Sub At Repeat Orders First Only	Curl Cream	12,387
8	US	2022-10	Repeat Sub At Repeat Orders First Only	Dry Shampoo	2,594
9	US	2022-10	Repeat Sub At Repeat Orders First Only	Essentials Set	18
10	US	2022-10	Repeat Sub At Repeat Orders First Only	Filet Bag Gwp	4

LookML

Return to TOC

Below addresses custom LookML implementations throughout the dashboards. General dimensions/measures will not be discussed here; components that provided creative solutions to business-specific problems will be reserved for this section.

Manifest File

Return to TOC

The project's manifest file (`manifest.lkml`) was used to define many constants that are subsequently referenced throughout the projects LookML files using Looker's @{constant_name} conventions. Variables include a variety of SQL date range expressions/declarations, dynamic currency formatting, and even a variable for parsing in HTML to accomplish format conversions.

Below addresses the two general types of constants in the manifest file: Formatting Blocks and SQL Constants.

Formatting Blocks

Return to TOC

A number of liquid formatting and logic components are stored within the manifest file. Below details each component and its purpose:

Dynamic Currency Formatting

Return to TOC

Looker does not have a native functionality to dynamically format currency with a `value_format`; that is, a way to give currency summaries if something is in the millions or thousands with 'M' or 'K', respectively.

Thus, this necessitated utilizing Liquid for dynamic currency formatting via a dimension/measure's `html` parameter. Below is the constant that houses this Liquid logic:

```
constant: dynamic_currency_formatting_us {
value: "{% assign abs_value = value | abs %}
{% if abs_value >= 1000 and abs_value < 1000000 %}
${{value | divided_by: 1000 | round: 2 }}K
{% elsif abs_value >= 1000000 and abs_value <  1000000000 %}
${{value | divided_by: 1000000 | round: 2 }}M
{% elsif abs_value >= 1000000000 %}
${{value | divided_by: 1000000000 | round: 2 }}B
{% else %}
${{value | round: 2 }}
{% endif %}"}
```

Proper Casing of Snakecase Strings

Return to TOC

The below Liquid logic splits strings with the '_' delimiter and capitalizes each word in the returned split:

```
constant: proper_case_snakecase {
  value: "{% assign words = value | split: '_' %}
```

```
    {% for word in words %}
    {% assign down_word = word | downcase %}
    {% if down_word == 'gcs' %}
    {{word | upcase}}
    {% elsif down_word == 'ssn' %}
    {{word | upcase}}
    {% elsif down_word contains 'https' %}
    {{word | downcase }}
    {% elsif down_word contains '@' %}
    {{word | downcase }}
    {% elsif down_word == 'n/a' %}
    {{word | upcase }}
    {%else %}
    {{ word | capitalize }}
    {% endif %}
    {% endfor %}"
  }
```

SQL Date Expressions

Return to TOC

A number of date logic subqueries are stored and utilized from the manifest file. These were instantiated within the manifest file due to the reusability throughout the various dashboards.

Below tackles each date logic subquery and explains the reasoning behind its implementation:

Yesterday SQL

Return to TOC

The below constant houses SQL that grabs yesterday's date:

```
constant: yesterday_sql {
  value: "date_add(current_date, interval -1 day)"
}
```

Yesterday Prior Year SQL

Return to TOC

The below constant houses SQL that grabs yesterday's date in the prior year:

```
constant: yesterday_prior_year {
  value: "date(extract(year from @{yesterday_sql}) - 1, extract(month from
@{yesterday_sql}), extract(day from @{yesterday_sql}))"
}
```

Yesterday Same Day Prior Year SQL

Return to TOC

The below constant houses SQL that grabs the date of the same day of yesterday in the prior year. This is not the same date last year; this is the same day of the week ~365 days prior. Thus, this would not be 10/01/2021 for a comparison with 10/01/2022; the correct value would be 10/02/2021, since both 10/02/2021 and 10/01/2022 both fall on Saturdays.

Also note: the following SQL takes into account if the current or compared-to year are leap years.

```
constant: yesterday_same_day_prior_year {
  value:  "case
  when extract(day from (last_day(date(extract(year from
  @{yesterday_sql}),
  2, 1)))) = 29
  then date_add(date(extract(year from @{yesterday_sql}) - 1,
  extract(month from @{yesterday_sql}),
  extract(day from @{yesterday_sql})), interval 2 day)
  else
  date_add(date(extract(year from @{yesterday_sql}) - 1,
  extract(month from @{yesterday_sql}),
  extract(day from @{yesterday_sql})), interval 1 day)
  end"
}
```

Beginning of Last 7 days SQL

Return to TOC

The below constant houses SQL that grabs the date from 7 days ago:

```
constant: beginning_of_last_7_days {
  value: "date_add(current_date, interval -7 day)"
}
```

Beginning of Last 7 Days Prior Year SQL

Return to TOC

The below constant houses SQL that grabs the date from 7 days ago in the prior year.

```
constant: beginning_of_last_7_days_prior_year {
  value: "date(extract(year from @{beginning_of_last_7_days}) - 1, extract(month from
@{beginning_of_last_7_days}), extract(day from @{beginning_of_last_7_days}))"
}
```

Beginning of Last 7 Days Same Day Prior Year SQL

Return to TOC

The below constant houses SQL that grabs the date of the same day of 7 days ago in the prior year. This is not the same date last year; this is the same day of the week ~365 days prior. Thus, this would not be 10/01/2021 for a comparison with 10/01/2022; the correct value would be 10/02/2021, since both 10/02/2021 and 10/01/2022 both fall on Saturdays.

Also note: the following SQL takes into account if the current or compared-to year are leap years.

```
constant: beginning_of_last_7_days_same_day_prior_year {
  value:  "case
  when extract(day from (last_day(date(extract(year from
  @{beginning_of_last_7_days}),
  2, 1)))) = 29
  then date_add(date(extract(year from @{beginning_of_last_7_days}) - 1,
  extract(month from @{beginning_of_last_7_days}),
  extract(day from @{beginning_of_last_7_days})), interval 2 day)
  else
  date_add(date(extract(year from @{beginning_of_last_7_days}) - 1,
  extract(month from @{beginning_of_last_7_days}),
  extract(day from @{beginning_of_last_7_days})), interval 1 day)
  end"
}
```

Beginning of Last 30 Days Prior Year SQL

Return to TOC

The below constant houses SQL that grabs the date from 30 days ago in the prior year.

```
constant: beginning_of_last_30_days {
  value: "date_add(current_date, interval -30 day)"
}
```

Beginning of Last 30 Days Prior Year SQL

Return to TOC

The below constant houses SQL that grabs the date from 30 days ago in the prior year.

```
constant: beginning_of_last_30_days_prior_year {
  value: "date(extract(year from @{beginning_of_last_30_days}) - 1, extract(month from
@{beginning_of_last_30_days}), extract(day from @{beginning_of_last_30_days}))"
}
```

Beginning of Last 30 Days Same Day Prior Year SQL

Return to TOC

The below constant houses SQL that grabs the date of the same day of 30 days ago in the prior year. This is not the same date last year; this is the same day of the week ~365 days prior. Thus, this would not be 10/01/2021 for a comparison with 10/01/2022; the correct value would be 10/02/2021, since both 10/02/2021 and 10/01/2022 both fall on Saturdays.

Also note: the following SQL takes into account if the current or compared-to year are leap years.

```
constant: beginning_of_last_30_days_same_day_prior_year {
  value:  "case
  when extract(day from (last_day(date(extract(year from
  @{beginning_of_last_30_days}),
```

```
  2, 1)))) = 29
  then date_add(date(extract(year from @{beginning_of_last_30_days}) - 1,
  extract(month from @{beginning_of_last_30_days}),
  extract(day from @{beginning_of_last_30_days})), interval 2 day)
  else
  date_add(date(extract(year from @{beginning_of_last_30_days}) - 1,
  extract(month from @{beginning_of_last_30_days}),
  extract(day from @{beginning_of_last_30_days})), interval 1 day)
  end"
}
```

Beginning of Last 90 days SQL

Return to TOC

The below constant houses SQL that grabs the date from 90 days ago:

```
constant: beginning_of_last_90_days {
  value: "date_add(current_date, interval -90 day)"
}
```

Beginning of Last 90 Days Prior Year SQL

Return to TOC

The below constant houses SQL that grabs the date from 90 days ago in the prior year.

```
constant: beginning_of_last_90_days_prior_year {
  value: "date(extract(year from @{beginning_of_last_90_days}) - 1, extract(month from
@{beginning_of_last_90_days}), extract(day from @{beginning_of_last_90_days}))"
}
```

Beginning of Last 365 days SQL

Return to TOC

The below constant houses SQL that grabs the date from 365 days ago:

```
constant: beginning_of_last_365_days {
  value: "date_add(current_date, interval -365 day)"
}
```

Beginning of Last 365 Days Same Day Prior Year SQL

Return to TOC

The below constant houses SQL that grabs the date of the same day of 365 days ago in the prior year. This is not the same date last year; this is the same day of the week ~365 days prior. Thus, this would not be 10/01/2021 for a comparison with 10/01/2022; the correct value would be 10/02/2021, since both 10/02/2021 and 10/01/2022 both fall on Saturdays.

Also note: the following SQL takes into account if the current or compared-to year are leap years.

```
constant: beginning_of_last_365_days_same_day_prior_year {
  value:  "case
  when extract(day from (last_day(date(extract(year from
  @{beginning_of_last_365_days}),
  2, 1)))) = 29
  then date_add(date(extract(year from @{beginning_of_last_365_days}) - 1,
  extract(month from @{beginning_of_last_365_days}),
  extract(day from @{beginning_of_last_365_days})), interval 2 day)
  else
  date_add(date(extract(year from @{beginning_of_last_365_days}) - 1,
  extract(month from @{beginning_of_last_365_days}),
  extract(day from @{beginning_of_last_365_days})), interval 1 day)
  end"
}
```

Beginning of Last Year SQL

Return to TOC

The below constant houses SQL that grabs the date for the beginning of the year:

```
constant: beginning_of_last_year {
  value: "date(extract(year from (current_date())), 1, 1)"
}
```

Beginning of Last Year Prior Year SQL

Return to TOC

The below constant houses SQL that grabs the date for the beginning of last year.

```
constant: beginning_of_last_year_prior_year {
  value: "date(extract(year from @{beginning_of_last_year}) - 1, 1, 1)"
}
```

Custom Dimensions

Return to TOC

In this section, any dimensions developed that fall outside what would be considered standard dimensions will be addressed. Thus, any dimensions provided by Client upon view creation/dimensions that perform modest calculations will not be addressed in this section.

Yesterday YesNo Dimension

Return to TOC

The following is a simple `yesno` dimension that returns `true` if an order date is the same date as yesterday's date. Note that it references the `yesterday_sql` SQL constant stored in the manifest.

This dimension is referenced throughout other fields in the LookML as a foundational building block.

```
dimension: yesterday {
hidden: yes
type: yesno
sql: case when ${order_date_at_tz_ny_date} = @{yesterday_sql}
              then true
              else false
              end;;
group_label: "Relative to Today"
}
```

Last 7 Days YesNo Dimension

Return to TOC

The following is a simple `yesno` dimension that returns `true` if an order date is between the last 7 days and yesterday's dates. Note that it references the `beginning_of_last_7_days` & `yesterday_sql` SQL constants stored in the manifest.

This dimension is referenced throughout other fields in the LookML as a foundational building block.

```
dimension: last_7_days {
hidden: yes
type: yesno
sql: case when ${order_date_at_tz_ny_date} between @{beginning_of_last_7_days} and
@{yesterday_sql}
              then true
              else false
              end;;
group_label: "Relative to Today"
}
```

Last 30 Days YesNo Dimension

Return to TOC

The following is a simple `yesno` dimension that returns `true` if an order date is between the last 30 days and yesterday's dates. Note that it references the `beginning_of_last_30_days` & `yesterday_sql` SQL constants stored in the manifest.

This dimension is referenced throughout other fields in the LookML as a foundational building block.

```
dimension: last_30_days {
  hidden: yes
  type: yesno
  sql: case when ${order_date_at_tz_ny_date} between @{beginning_of_last_30_days} and
@{yesterday_sql}
              then true
              else false
              end;;
  group_label: "Relative to Today"
}
```

Last 365 Days YesNo Dimension

Return to TOC

The following is a simple `yesno` dimension that returns `true` if an order date is between the last 365 days and yesterday's dates. Note that it references the `beginning_of_last_365_days` & `yesterday_sql` SQL constants stored in the manifest.

This dimension is referenced throughout other fields in the LookML as a foundational building block.

```
dimension: last_365_days {
  hidden: yes
  type: yesno
  sql: case when ${order_date_at_tz_ny_date} between @{beginning_of_last_365_days} and
@{yesterday_sql}
              then true
              else false
              end;;
  group_label: "Relative to Today"
}
```

Yesterday Prior Year YesNo Dimension

Return to TOC

The following is a simple `yesno` dimension that returns `true` if an order date is the same day as yesterday in the prior year's date. Note that it references the `yesterday_same_day_prior_year` SQL constant stored in the manifest.

This dimension is referenced throughout other fields in the LookML as a foundational building block.

```
dimension: yesterday_prior_year {
hidden: yes
type: yesno
sql: case when ${order_date_at_tz_ny_date} = @{yesterday_same_day_prior_year}
              then true
              else false
              end;;
group_label: "Relative to Prior Year"
}
```

Last 7 Days Prior Year YesNo Dimension

Return to TOC

The following is a simple `yesno` dimension that returns `true` if an order date is between the same day as 7 days ago in the prior year and yesterday's same day prior year dates. Note that it references the `beginning_of_last_7_days_same_day_prior_year` & `yesterday_same_day_prior_year` SQL constants stored in the manifest.

This dimension is referenced throughout other fields in the LookML as a foundational building block.

```
dimension: last_7_days_prior_year {
  hidden: yes
  type: yesno
  sql: case when ${order_date_at_tz_ny_date} between
@{beginning_of_last_7_days_same_day_prior_year} and @{yesterday_same_day_prior_year}
              then true
              else false
              end;;
  group_label: "Relative to Prior Year"
}
```

Last 30 Days Prior Year YesNo Dimension

Return to TOC

The following is a simple `yesno` dimension that returns `true` if an order date is between the same day as 30 days ago in the prior year and yesterday's same day prior year dates. Note that it references the `beginning_of_last_30_days_same_day_prior_year` & `yesterday_same_day_prior_year` SQL constants stored in the manifest.

This dimension is referenced throughout other fields in the LookML as a foundational building block.

```
dimension: last_30_days_prior_year {
  hidden: yes
  type: yesno
  sql: case when ${order_date_at_tz_ny_date} between
@{beginning_of_last_30_days_same_day_prior_year} and @{yesterday_same_day_prior_year}
              then true
              else false
              end;;
  group_label: "Relative to Prior Year"
}
```

Last 365 Days Prior Year YesNo Dimension

Return to TOC

The following is a simple `yesno` dimension that returns `true` if an order date is between the same day as 365 days ago in the prior year and yesterday's same day prior year dates. Note that it references the `beginning_of_last_365_days_same_day_prior_year` & `yesterday_same_day_prior_year` SQL constants stored in the manifest. This dimension is referenced throughout other fields in the LookML as a foundational building block.

```
dimension: last_365_days_prior_year {
  hidden: yes
  type: yesno
  sql: case when ${order_date_at_tz_ny_date} between
@{beginning_of_last_365_days_same_day_prior_year} and @{yesterday_same_day_prior_year}
              then true
              else false
              end;;
  group_label: "Relative to Prior Year"
}
```

Order Pubkey (Yesterday) String Dimension

Return to TOC

The following string dimension isolates order pubkeys that involve orders placed yesterday. This is integral to allowing count distinct integrity across various time binnings in the Daily Flash dashboard.

```
dimension: order_pubkey_yesterday {
hidden: yes
type: string
sql: case when ${yesterday}
    then ${TABLE}.order_pubkey
    else null
    end;;
group_label: "ID Fields"
}
```

Order Pubkey (Last 7 Days) String Dimension

Return to TOC

The following string dimension isolates order pubkeys that involve orders placed within the last 7 days. This is integral to allowing count distinct integrity across various time binnings in the Daily Flash dashboard.

```
dimension: order_pubkey_last_7_days {
  hidden: yes
  type: string
  sql: case when ${yesterday} or ${last_7_days}
          then ${TABLE}.order_pubkey
          else null
          end;;
  group_label: "ID Fields"
}
```

Order Pubkey (Last 30 Days) String Dimension

Return to TOC

The following string dimension isolates order pubkeys that involve orders placed within the last 30 days. This is integral to allowing count distinct integrity across various time binnings in the Daily Flash dashboard.

```
dimension: order_pubkey_last_30_days {
  hidden: yes
  type: string
  sql: case when ${yesterday} or ${last_7_days} or ${last_30_days}
          then ${TABLE}.order_pubkey
          else null
          end;;
  group_label: "ID Fields"
}
```

Order Pubkey (Last 365 Days) String Dimension

Return to TOC

The following string dimension isolates order pubkeys that involve orders placed within the last 365 days. This is integral to allowing count distinct integrity across various time binnings in the Daily Flash dashboard.

```
dimension: order_pubkey_last_365_days {
  hidden: yes
  type: string
  sql: case when ${yesterday} or ${last_7_days} or ${last_30_days} or ${last_365_days}
          then ${TABLE}.order_pubkey
          else null
          end;;
  group_label: "ID Fields"
}
```

Customer Pubkey (Yesterday) String Dimension

Return to TOC

The following string dimension isolates customer pubkeys that involve orders placed yesterday. This is integral to allowing count distinct integrity across various time binnings in the Daily Flash dashboard.

```
dimension: customer_pubkey_yesterday {
  hidden: yes
  type: string
  sql: case when ${yesterday}
          then ${TABLE}.customer_pubkey
          else null
          end;;
  group_label: "ID Fields"
}
```

Customer Pubkey (Last 7 Days) String Dimension

Return to TOC

The following string dimension isolates customer pubkeys that involve orders placed within the last 7 days. This is integral to allowing count distinct integrity across various time binnings in the Daily Flash dashboard.

```
dimension: customer_pubkey_last_7_days {
  hidden: yes
  type: string
  sql: case when ${yesterday} or ${last_7_days}
          then ${TABLE}.customer_pubkey
          else null
          end;;
  group_label: "ID Fields"
}
```

Customer Pubkey (Last 30 Days) String Dimension

Return to TOC

The following string dimension isolates customer pubkeys that involve orders placed within the last 30 days. This is integral to allowing count distinct integrity across various time binnings in the Daily Flash dashboard.

```
dimension: customer_pubkey_last_30_days {
  hidden: yes
  type: string
  sql: case when ${yesterday} or ${last_7_days} or ${last_30_days}
          then ${TABLE}.customer_pubkey
          else null
          end;;
  group_label: "ID Fields"
}
```

Customer Pubkey (Last 365 Days) String Dimension

Return to TOC

The following string dimension isolates customer pubkeys that involve orders placed within the last 365 days. This is integral to allowing count distinct integrity across various time binnings in the Daily Flash dashboard.

```
dimension: customer_pubkey_last_365_days {
  hidden: yes
  type: string
  sql: case when ${yesterday} or ${last_7_days} or ${last_30_days} or ${last_365_days}
          then ${TABLE}.customer_pubkey
          else null
          end;;
  group_label: "ID Fields"
}
```

Order Date (Yesterday) String Dimension

Return to TOC

The following is a simple `string` dimension that returns `order_date_at_tz_ny_date` if an order date is the same as yesterday's date. Note that it uses the `yesterday yesno` dimension for logic.

```
dimension: order_date_yesterday {
hidden: yes
type: string
sql: case when ${yesterday}
          then ${order_date_at_tz_ny_date}
          else null
          end;;
group_label: "Daily Flash Calculation"

}
```

Order Date (Last 7 Days) String Dimension

Return to TOC

The following is a simple `string` dimension that returns `order_date_at_tz_ny_date` if an order date is between the last 7 days and yesterday's dates. Note that it uses the `last_7_days` & `yesterday yesno` dimensions for logic.

```
dimension: order_date_last_7_days {
  hidden: yes
  type: string
  sql: case when ${yesterday} or ${last_7_days}
          then ${order_date_at_tz_ny_date}
          else null
          end;;
  group_label: "Daily Flash Calculation"
}
```

Order Date (Last 30 Days) String Dimension

Return to TOC

The following is a simple `string` dimension that returns `order_date_at_tz_ny_date` if an order date is between the last 30 days and yesterday's dates. Note that it uses the `last_30_days`, `last_7_days` & `yesterday yesno` dimensions for logic.

```
dimension: order_date_last_30_days {
  hidden: yes
  type: string
  sql: case when ${yesterday} or ${last_7_days} or ${last_30_days}
          then ${order_date_at_tz_ny_date}
          else null
          end;;
  group_label: "Daily Flash Calculation"
}
```

Order Date (Last 365 Days) String Dimension

Return to TOC

The following is a simple `string` dimension that returns `order_date_at_tz_ny_date` if an order date is between the last 365 days and yesterday's dates. Note that it uses the `last_365_days`, `last_30_days`, `last_7_days` & `yesterday yesno` dimensions for logic.

```
dimension: order_date_last_365_days {
  hidden: yes
  type: string
  sql: case when ${yesterday} or ${last_7_days} or ${last_30_days} or ${last_365_days}
          then ${order_date_at_tz_ny_date}
          else null
          end;;
  group_label: "Daily Flash Calculation"
}
```

Order Date (Yesterday Prior Year) String Dimension

Return to TOC

The following is a simple `string` dimension that returns `order_date_at_tz_ny_date` if an order date as the same yesterday prior year's date. Note that it uses the `yesterday_prior_year` `yesno` dimension for logic.

```
dimension: order_date_yesterday_prior_year {
  hidden: yes
  type: string
  sql: case when ${yesterday_prior_year}
          then ${order_date_at_tz_ny_date}
          else null
          end;;
  group_label: "Daily Flash Calculation"
}
```

Order Date (Last 7 Days Prior Year) String Dimension

Return to TOC

The following is a simple `string` dimension that returns `order_date_at_tz_ny_date` if an order date is between the last 7 days in the prior year and yesterday in the prior year's dates. Note that it uses the `last_7_days_prior_year` & `yesterday_prior_year` `yesno` dimensions for logic.

```
dimension: order_date_last_7_days_prior_year {
  hidden: yes
  type: string
  sql: case when ${yesterday_prior_year} or ${last_7_days_prior_year}
          then ${order_date_at_tz_ny_date}
          else null
          end;;
  group_label: "Daily Flash Calculation"
}
```

Order Date (Last 30 Days Prior Year) String Dimension

Return to TOC

The following is a simple `string` dimension that returns `order_date_at_tz_ny_date` if an order date is between the last 30 days in the prior year and yesterday in the prior year's dates. Note that it uses the `last_30_days_prior_year`, `last_7_days_prior_year` & `yesterday_prior_year` `yesno` dimensions for logic.

```
dimension: order_date_last_30_days_prior_year {
  hidden: yes
  type: string
  sql: case when ${yesterday_prior_year} or ${last_7_days_prior_year} or
${last_30_days_prior_year}
          then ${order_date_at_tz_ny_date}
          else null
```

```
          end;;
  group_label: "Daily Flash Calculation"
}
```

Order Date (Last 365 Days Prior Year) String Dimension

Return to TOC

The following is a simple `string` dimension that returns `order_date_at_tz_ny_date` if an order date is between the last 365 days in the prior year and yesterday in the prior year's dates. Note that it uses the `last_365_days_prior_year`, `last_30_days_prior_year`, `last_7_days_prior_year` & `yesterday_prior_year` `yesno` dimensions for logic.

```
dimension: order_date_last_365_days_prior_year {
  hidden: yes
  type: string
  sql: case when ${yesterday_prior_year} or ${last_7_days_prior_year} or
${last_30_days_prior_year} or ${last_365_days_prior_year}
          then ${order_date_at_tz_ny_date}
          else null
          end;;
  group_label: "Daily Flash Calculation"
}
```

Daily Flash Times String Dimesion

Return to TOC

The following dimension bins time according to date criteria, returning a bin string if an order date meets the corresponding criteria. This dimension is used in the KPI Performance over Time Look on the Daily Flash tab of the Company Report.

Note that this dimension is ordered by the `flash_rank` dimension described directly after this dimension.

```
dimension: daily_flash_times {
description: "Buckets for different timeframes to compare by in the Daily Flash dashboard"
hidden: yes
sql: case when ${order_date_yesterday} is not null
    then 'Yesterday'
    when ${order_date_last_7_days} is not null
    then 'Last 7 Days'
    when ${order_date_last_30_days} is not null
    then 'Last 30 Days'
    when ${order_date_last_365_days} is not null
    then 'Last 365 Days'
    else null
    end;;
order_by_field: flash_rank
group_label: "Daily Flash Calculation"
}
```

Flash Rank Number Dimesion

Return to TOC

This dimension ensures chronological order sorting of the `daily_flash_times` dimension. This dimension recasts the logic in `daily_flash_times` into a ranking of integers.

```
dimension: flash_rank {
  description: "Hidden field used to rank the daily_flash_times dinmension
chronologically."
  hidden: yes
  sql: case when ${daily_flash_times} = 'Yesterday'
          then 1
          when ${daily_flash_times} = 'Last 7 Days'
          then 2
          when ${daily_flash_times} = 'Last 30 Days'
          then 3
          when ${daily_flash_times} = 'Last 365 Days'
          then 4
          when ${daily_flash_times} is null
          then 5
          end;;
}
```

Current vs Previous Period Dynamic Dimension

Return to TOC

This dynamic dimension employs a mixture of Liquid logic and SQL logic. This dimension is dependent upon the seletion of the `select_timeframe` parameter, which branches the Liquid logic accordingly. Within each Liquid block is a different `case when` criterion addressing order date logic.

This dimension returns either 'Current Year' or 'Prior Year' after the logic has been evaluated. This is a dynamic dimension; any measure stacked against it will be cast to compare the current vs prior year metrics.

```
dimension: current_vs_previous_period {
description: "Use this dimension along with the Period over Period Scope Filter for dynamic
Period of Period analysis"
hidden: yes
type: string
sql:
    case
      {% if select_timeframe._parameter_value == "'Last 7 Days'" %}
        when ${order_date_at_tz_ny_date}
        between @{beginning_of_last_7_days} and @{yesterday_sql}
      {% elsif select_timeframe._parameter_value == "'Last 30 Days'" %}
        when ${order_date_at_tz_ny_date}
        between @{beginning_of_last_30_days} and @{yesterday_sql}
      {% elsif select_timeframe._parameter_value == "'Last 90 Days'" %}
       when ${order_date_at_tz_ny_date}
        between @{beginning_of_last_90_days} and @{yesterday_sql}
       {% elsif select_timeframe._parameter_value == "'YTD'" %}
        when ${order_date_at_tz_ny_date}
        between @{beginning_of_last_year} and @{yesterday_sql}
        {% endif %}
        then 'Current Year'
      {% if select_timeframe._parameter_value == "'Last 7 Days'" %}
```

```
        when ${order_date_at_tz_ny_date}
        between @{beginning_of_last_7_days_prior_year} and @{yesterday_prior_year}
      {% elsif select_timeframe._parameter_value == "'Last 30 Days'" %}
        when ${order_date_at_tz_ny_date}
        between @{beginning_of_last_30_days_prior_year} and @{yesterday_prior_year}
      {% elsif select_timeframe._parameter_value == "'Last 90 Days'" %}
        when ${order_date_at_tz_ny_date}
        between @{beginning_of_last_90_days_prior_year} and @{yesterday_prior_year}
      {% elsif select_timeframe._parameter_value == "'YTD'" %}
        when ${order_date_at_tz_ny_date}
        between @{beginning_of_last_year_prior_year} and @{yesterday_prior_year}
      {% endif %}
      then 'Prior Year'
      else null
    end
    ;;
}
```

Selected Dynamic Duration Number Dynamic Dimension

Return to TOC

This dynamic dimension is Liquid dependent. The key evaluator is the `select_timeframe` parameter value. Depending on the value, a different sorting index is returned to aide in sorting.

```
dimension: selected_dynamic_duration  {
  description: "Dynamic date column for Period over Period analysis"
  hidden: yes
  type: number
  sql:
    {% if select_timeframe._parameter_value == "'Last 7 Days'" %}
      ${order_date_at_tz_ny_day_of_week_index}
    {% elsif select_timeframe._parameter_value == "'Last 30 Days'" %}
     ${order_date_at_tz_ny_month_num}
    {% elsif select_timeframe._parameter_value == "'Last 90 Days'" %}
      ${order_date_at_tz_ny_month_num}
    {% elsif select_timeframe._parameter_value == "'YTD'" %}
     ${order_date_at_tz_ny_month_num}
    {% endif %}
    ;;
}
```

Selected Dynamic Day Of String Dynamic Dimension

Return to TOC

This dynamic dimension is Liquid dependent. The key evaluator is the `select_timeframe` parameter value. Depending on the value, a different date formatting is returned. This formatting is what is rendered on the x-axis of dynamic trend visualizations.

Note in the formatting that year is formatted out. This is to allow comparison between current and previous periods simultaneously.

```
dimension: selected_dynamic_day_of  {
  label: "Trend Line Date"
  description: "Dynamic date formatting column for Period over Period analysis"
  hidden: yes
  order_by_field: selected_dynamic_day_of_sort
  type: string
  sql:
    {% if select_timeframe._parameter_value == "'Last 7 Days'" %}
      format_date("%m/%d", ${order_date_at_tz_ny_date})
    {% elsif select_timeframe._parameter_value == "'Last 30 Days'" %}
      format_date("%m/%d", ${order_date_at_tz_ny_date})
    {% elsif select_timeframe._parameter_value == "'Last 90 Days'" %}
      format_date("%m/%d", ${order_date_at_tz_ny_date})
    {% elsif select_timeframe._parameter_value == "'YTD'" %}
    ${order_date_at_tz_ny_month_name}
    {% endif %}
    ;;
}
```

Selected Dynamic Day Of Sort String Dynamic Dimension

Return to TOC

This dynamic dimension is Liquid dependent. The key evaluator is the `select_timeframe` parameter value. Depending on the value, a different date formatting is returned. This formatting is used to sort values chronologically.

Note in the formatting that year is formatted out. This is to allow comparison between current and previous periods simultaneously.

```
dimension: selected_dynamic_day_of_sort  {
  description: "Sorting column to dynamically sort Period over Period analysis in
chronological order"
  hidden:  yes
  type: string
  sql:
    {% if select_timeframe._parameter_value == "'Last 7 Days'" %}
      format_date("%m/%d",${order_date_at_tz_ny_date})
    {% elsif select_timeframe._parameter_value == "'Last 30 Days'" %}
    format_date("%m/%d",${order_date_at_tz_ny_date})
    {% elsif select_timeframe._parameter_value == "'Last 90 Days'" %}
      format_date("%m/%d",${order_date_at_tz_ny_date})
    {% elsif select_timeframe._parameter_value == "'YTD'" %}
    format_date("%m",${order_date_at_tz_ny_date})
    {% endif %};;
}
```

Parameters

Return to TOC

`select_timeframe` | `order` | A parameter with values of 'Last 7 Days', 'Last 30 Days', 'Last 90 Days' and 'YTD'. This parameter is manipulated when buttons are clicked on the dashboard to change the date range and date formatting.|

Custom Measures

Return to TOC

| total_number_orders_daily_flash | order | count_distinct measure with a case when statement in the sql parameter. In this case when statement, it evaluates which bin type daily_flash_times is returning, then pulling the corresponding order_pubkey dimension to perform a count distinct. Thus, if daily_flash_times = 'Last 30 Days' then order_pubkey_last_30_days is returned. | | total_number_orders_yesterday_py
total_number_orders_last_7_days_py
total_number_orders_last_30_days_py
total_number_orders_last_365_days_py | order | count_distinct measures that calculate the prior year values of total orders. These had to be separated into separate measures because the primary dates were already assumed in the daily_flash_times; creating a daily_flash_times_prior_year would double the crosstab size and not allow side-by-side comparison. Thus, separate, filtered measures were created, later being aggregated via a table calculation in the final Look. | | distinct_new_customer_count_daily_flash | order | Same logic for total_number_orders_daily_flash, different pubkey returns (customer_pubkey return instead of order_pubkey). | | distinct_new_customer_count_yesterday_py
distinct_new_customer_count_last_7_days_py
distinct_new_customer_count_last_30_days_py
distinct_new_customer_count_last_365_days_py | order | Same logic for total_number_orders_[time_bin]_py measures above, except calculating the total number of new distinct customers instead of distinct order counts. | | total_net_sales_USD_running_total | order | A simple running total upon total_net_sales_USD. This cares for the binary classification issue originally grappled with the daily_flash_times. | |
total_net_sales_USD_yesterday_py
total_net_sales_USD_last_7_days_py
total_net_sales_USD_last_30_days_py
total_net_sales_USD_last_365_days_py | order | sum_distinct measures that calculate the prior year values of total net sales. These had to be separated into separate measures because the primary dates were already assumed in the daily_flash_times; creating a daily_flash_times_prior_year would double the crosstab size and not allow side-by-side comparison. Thus, separate, filtered measures were created, later being aggregated via a table calculation in the final Look. | | total_media_spend_running_total | marketing_spend | A simple running total upon total_media_spend. This cares for the binary classification issue originally grappled with the daily_flash_times. | |
total_media_spend_yesterday_py
total_media_spend_last_7_days_py
total_media_spend_last_30_days_py
total_media_spend_last_365_days_py | marketing_spend | sum measures that calculate the prior year values of total marketing spend. These had to be separated into separate measures because the primary dates were already assumed in the daily_flash_times; creating a daily_flash_times_prior_year would double the crosstab size and not allow side-by-side comparison. Thus, separate, filtered measures were created, later being aggregated via a table calculation in the final Look.| | cpa_yesterday_py
cpa_last_7_days_py
cpa_last_30_days_py
cpa_last_365_days_py | marketing_spend | Measures that calculate the prior year values of CPA. These had to be separated into separate measures because the primary dates were already assumed in the daily_flash_times; creating a daily_flash_times_prior_year would double the crosstab size and not allow side-by-side comparison. Thus, separate, filtered measures were created, later being aggregated via a table calculation in the final Look.|

Product Launch Components Reference

Return to TOC

Launch Date

return

```
view: drv_launch_date_by_product {
    derived_table: {
      sql:
      select distinct product_name, product_target_sub_category, target_sub_category,
min(order_date_at_tz_ny) over (partition by product_name order by product_name) as
launch_date FROM
`Clienthair.Client_bi_datamart_growth_customer_ltv_customer_order.order_items`
      where item_type ='formula' ;;
    }
  dimension: product_name{
    primary_key: yes
    type: string
    sql: ${TABLE}.product_name ;;
  }
  dimension: target_sub_category {
    type: string
    sql: ${TABLE}.target_sub_category ;;
  }
    dimension: product_target_sub_category {
      type: string
      sql: ${TABLE}.product_target_sub_category ;;
    }
  dimension_group: launch_date {
    type: time
    timeframes: [
      raw,
      date,
      day_of_week,
      day_of_week_index,
      week,
      month,
      month_name,
      month_num,
      quarter,
      year
    ]
    convert_tz: no
    datatype: date
    sql: ${TABLE}.launch_date ;;
  }
  }
```

Launch

return

```
dimension_group: launch {
    type: duration
    sql_start: ${drv_launch_date_by_product.launch_date_raw} ;;
    sql_end: ${order_date_at_tz_ny_raw} ;;
  }
```

Select Days Since Launch

return

```
parameter: select_days_since_launch {
    hidden: no
    type: string
    default_value: "< 31"
    allowed_value: {
      label: "30 days"
      value: "< 31"
    }
    allowed_value: {
      label: "60 Days"
      value: "< 61"
    }
    allowed_value: {
      label: "90 Days"
      value: "< 91"
    }
    allowed_value: {
      label:"1 yr"
      value: "< 366"
    }
    allowed_value: {
      label: "no limit"
      value: "< 99999"
    }
  }
```

Selected Dynamic Launch Days Label

return

```
dimension: selected_dynamic_launch_days_label  {
  description: "Dynamic Number of Launch days grouping"
  label: "Days Since Launch"
  type: number
  sql:
    {% if select_days_since_launch._parameter_value == "'< 31'" %}
      ${days_launch}
    {% elsif select_days_since_launch._parameter_value == "'< 61'" %}
       ${days_launch}
    {% elsif select_days_since_launch._parameter_value == "'< 91'" %}
       DIV(${days_launch},7)* 7
    {% elsif select_days_since_launch._parameter_value == "'< 366'" %}
       DIV(${days_launch},7)* 7
    {% elsif select_days_since_launch._parameter_value == "'< 99999'" %}
       DIV(${days_launch},30)* 30
    {% else %}
       DIV(${days_launch},7)* 7
    {% endif %}
        ;;
    }
```

Selected Dynamic Launch Days Duration

return

```
dimension: selected_dynamic_launch_days_duration  {
      description: "Dynamic Number of Launch days grouping"
      type: number
      sql:
          {% if select_days_since_launch._parameter_value == "'< 31'" %}
           31
          {% elsif select_days_since_launch._parameter_value == "'< 61'" %}
            61
          {% elsif select_days_since_launch._parameter_value == "'< 91'" %}
            91
          {% elsif select_days_since_launch._parameter_value == "'< 366'" %}
           366
          {% elsif select_days_since_launch._parameter_value == "'< 99999'" %}
           9999
          {% else %}
           31
        {% endif %}
        ;;
    }
```

Custom Filter

[return](#custom filter_ref)

```
 ${order_items.days_launch}<${order_items.selected_dynamic_launch_days_duration}
```

Finance Cohort Components Reference

Return to TOC

Cohort Finance Name

[return](#Cohort Finance Name_ref)

```
view: cohort_finance_name {
  derived_table: {
    sql:
     Select "repeat_sub_at_first_orders_non_sub_supplements" as cohort_finance_name
     union all
     Select  "repeat_sub_at_first_orders_sub_haircare" as cohort_finance_name
     union all
     Select "repeat_sub_at_first_orders_non_sub_haircare" as cohort_finance_name
     union all
     Select "repeat_sub_at_first_orders_sub_supplements" as cohort_finance_name
     union all
     Select "repeat_sub_at_repeat_churned_customers" as cohort_finance_name
     union all
     Select "repeat_sub_at_first_churned_customers" as cohort_finance_name
     union all
     Select "new_non_sub" as cohort_finance_name
```

```
      union all
      Select "repeat_sub_at_repeat_orders_first_only" as cohort_finance_name
      union all
      Select "repeat_non_sub_orders" as cohort_finance_name
      union all
      Select "repeat_sub_at_repeat_orders_by_sub_cohort" as cohort_finance_name
      union all
      Select "repeat_sub_at_repeat_orders_by_sub_cohort_supplements" as cohort_finance_name
      union all
      Select "repeat_sub_at_repeat_orders_by_sub_cohort_haircare" as cohort_finance_name
      union all
      Select "repeat_sub_at_first_orders_non_sub" as cohort_finance_name
      union all
      Select "repeat_sub_at_first_orders_sub" as cohort_finance_name
      union all
      Select "repeat_sub_at_repeat_orders_by_first_order_cohort" as cohort_finance_name
      union all
      Select "new_sub_at_first" as cohort_finance_name
    ;;
  }
  dimension: cohort_finance_name {
    primary_key: yes
    type: string
    sql: ${TABLE}.cohort_finance_name ;;
    html: @{proper_case_snakecase} ;;
  }
}
```

Cohort Finance Units

[return](#Cohort Finance Units_ref)

```
 measure: cohort_finance_units {
      type: number
      sql: CASE
        WHEN ${cohort_finance_name}  = "new_non_sub"
        THEN ${new_non_subscription_at_first_units}

        WHEN ${cohort_finance_name}  = "new_sub_at_first"
        THEN ${new_subscription_at_first_units}

        WHEN ${cohort_finance_name}  = "repeat_non_sub_orders"
        THEN ${repeat_nonsub_units}

        WHEN ${cohort_finance_name}  ="repeat_sub_at_repeat_orders_by_first_order_cohort"
--same as above, grouping different on report??
        THEN ${repeat_subscription_units}

        WHEN ${cohort_finance_name}  = "repeat_sub_at_repeat_orders_first_only"
        THEN ${repeat_sub_at_first_units}

        WHEN ${cohort_finance_name}  = "repeat_sub_at_first_orders_non_sub"
        THEN ${repeat_sub_at_first_orders_non_sub_units}

        WHEN ${cohort_finance_name}  = "repeat_sub_at_first_orders_sub"
        THEN ${repeat_sub_at_first_sub_units}
```

```
        WHEN ${cohort_finance_name}  = "repeat_sub_at_repeat_churned_customers"  -- best
guess on this - need confirmation
        THEN ${repeat_subscription_units_churned}

        WHEN ${cohort_finance_name}  = "repeat_sub_at_first_churned_customers"
        THEN ${repeat_sub_at_first_sub_units_churned}

        else
        0 end ;;
    }
```

Explores

[Return to TOC](#TOC)

Explores used in the application are identified below along with their component views with additional commentary where applicable.

Explore Name: name	**Model File**	**Notes**
Explore #1 **customer:** Contained View File #1 customer	Test_model	Customer view, no joins
Explore #2 **customer_subscriber:** Contained View File #1 customer_subscriber	Test_model	Customer subscriber view, no joins
Explore #3 **order:** Contained View File #1 order Contained View File #2 customer	Test_model	Orders & Customers
Explore #4 **marketing_spend_and_orders:** Contained View File #1 order Contained View File #2 customer Contained View File #3 marketing_spend	Test_model	Orders & Customers
Explore #5 **order_items:** Contained View File #1 order_items Contained View File #2 customer Contained View File #3 order Contained View File #4 drv_launch_date_by_produc	Test_model	Created to support product level reporting by days since launch parameter

Explore Name: name	Model File	Notes
Explore #6 **last_synced:** Contained View File #1 order Contained View File #2 customer Contained View File #3 marketing_spend Contained View File #4 order_items	Test_model	Obtains last synced date from respective views
Explore #7 **finance_cohort:** Contained View File #1 order_cohort_extensions Contained View File #2 customer Contained View File #3 cohort_finance_name	Test_model	Created for Finance Cohorts reporting, supports reporting against derived cohort finance name dimension
Explore #8 **finance_cohort_products:** Contained View File #1 order_items_cohort_extension Contained View File #2 customer Contained View File #3 cohort_finance_name Contained View File #4 drv_launch_date_by_product	Test_model	Created for Finance Cohort product units reporting
Explore #9 **drv_pdt_daily_flash:**	Test_model	Single view explore on pdt for daily flash table visual
Explore #10 **subscription:**	Test_model	Single view explore on subscription view

View Files

[Return to TOC](#TOC)

The views required for the application are identified below, with a brief description of purpose and identification of referenced BigQuery tables.

The views are categorized as follows: -DB Table/view -Derived (Native) Table -Derived (SQL) Table -Extension

The lone persistent derived table, is identified as such, along with it's triggering datagroup. LookML objects with embedded logic in the SQL definition or LookML objects with unique parameters are listed with a brief description.

View File #1 customer (DB Table/view)

Only minor changes were required in the customer view, consisting of pulling in a couple of additional flags and correctinng a misspeling on leavin_conditioner, changed to leavein_conditioner.

BigQuery Tables Referenced: `Clienthair.Client_bi_datamart_growth_customer_ltv_customer_order.customer`

View File #2 customer_subscriber (Derived (SQL) Table)

This view is not used in the application.

BigQuery Tables Referenced:
`Clienthair.Client_bi_datamart_growth_customer_ltv_customer_order.order_items`

View File #3 marketing_spend (DB Table/view)

This view captures media spend by country, platform, channel group, and calendar date in which the media dollars were spent / recorded.n Filtered measures are used to "bin" prior year measures into 4 time period buckets (yesterday_py,last_7_days_prior_year, last_30_days_prior_year, and last_365_days_prior_year).)

BigQuery Tables Referenced: `Clienthair.Client_bi_datamart_marketing_spend.marketing_spend`

View File #4 order (DB Table/view)

The order view serves as one of the primary fact tables for the application, with heavy dependency on customer attributes from the customer view which is a required join to this view. A number of dimensions and measures were created to support "binning" of results by This year/last year grouping in to time period buckets of yesterday, last 7 days, last 30 days and last 365 days. In additon to the basic measure computation, dimensions were created to support dynamic labeling of the time period axis.

BigQuery Tables Referenced: `Clienthair.Client_bi_datamart_growth_customer_ltv_customer_order.order`

View File #5 order_items (DB Table/view)

This view pulls data from the order items data mart. There are many hidden fields within this view. This is to support efforts to provide visibility into rollup aggregation bucketed by timeframes.

BigQuery Tables Referenced:
`Clienthair.Client_bi_datamart_growth_customer_ltv_customer_order.order_items`

View File #6 subscription (DB Table/view)

No changes were made to this view

BigQuery Tables Referenced: `Clienthair.Client_bi_datamart_subscription.subscription`

View File #7 cohort_finance_name (Derived (SQL) Table)

This view is used to generate a list of 16 cohort_finance_names as identified in the requirements from Finance. Corresponding filtered measures are mapped to these names via case statements in the order_cohort_extension and orer_items_cohort_extension views.

BigQuery Tables Referenced: N/A

View File #8 order_cohort_extension (Extension)

This view extends the order view, adding cohort_finance_name from the required join to the cohort_finance_names view required to accomplish the cohort_name mapping to the appropriate measure. The measures themselves are redefined in this view, as sum_distinct is required as measure type due to the required cross join to cohort_finance_name.

BigQuery Tables Referenced: `Clienthair.Client_bi_datamart_growth_customer_ltv_customer_order.order`

View File #9 order_items_cohort_extension (Extension)

This view extends the order_items view, adding cohort_finance_name from the required join to the cohort_finance_names view required to accomplish the cohort_name mapping to the appropriate measure. Measure duplicaiton was not required for this view in that the measures are limited to type count_distinct.

BigQuery Tables Referenced:
`Clienthair.Client_bi_datamart_growth_customer_ltv_customer_order.order_items`

View File #10 drv_launch_date_by_product (Derived (SQL) Table)

This view was required to get a distinct launch date (min(order_date_at_tz_ny)) by product, which was used in launch duration dimension in order_items. The duration dimension is a filter in the Product Launch dashboard. This construction requires that drv_launch_date_by_product be joined in explores referencing order_items.

BigQuery Tables Referenced:
`Clienthair.Client_bi_datamart_growth_customer_ltv_customer_order.order_items`

View File #11 drv_pdt_daily_flash (Derived (Native) Table)

This persistent derived table was constructed to address performance issues with the daily flash table visual on the dashboard of the same name. The base lookml was generated from the non-pdt version of the visual, with additional measures added to support calculation of ratio (i.e. non-aggretable) measures in the view post aggregation.

Explore Source: `marketing_spend_and_orders`

Persistent Derived Table with datagroup_trigger: datagroup_orders Return to TOC