

# customers who order every 60 days

```
SELECT customers, COUNT(customers) +1 AS `count_orders`, MIN(first_order_date) AS `first_order_date`, AVG(days_difference) AS `avg_days_difference`,
(DATE_DIFF(CURRENT_DATE(), MIN(last_order_date), DAY)) AS days_since_last_order, MIN(last_order_date) as `last_order_date`,
SUM(qty) AS `sum_qty`, SUM(weft) AS `sum_weft`, SUM(poly) AS `poly`, SUM(IONIX) AS `IONIX`, SUM(cylinder) AS `sum_cylinder`, SUM(amt) AS `sum_amount`
FROM (
SELECT *, DATE_DIFF(CAST(DATE(full_date) AS DATE), CAST(DATE(previous_date) AS DATE), DAY) AS `days_difference`,
CASE WHEN DATE_DIFF(CAST(DATE(full_date) AS DATE), CAST(DATE(previous_date) AS DATE), DAY) <= 60 THEN 0 ELSE 1 END AS `fit`
FROM (
--
SELECT customers, invoice_id, MIN(year) AS `year`, MIN(month) AS `month`, MIN(day) AS `day`, MIN(last_order_date) AS `last_order_date`,
SUM(amount) AS `amount`, SUM(qty) AS `qty`, SUM(weft) AS `weft`, SUM(poly) AS `poly`, SUM(cylinder) AS `cylinder`, SUM(IONIX) AS `IONIX`,
MIN(full_date) AS `full_date`, MIN(first_order_date) AS `first_order_date`, LAG(MIN(full_date)) over (partition by customers order by
MIN(full_date) ASC) as `previous_date`

FROM (
SELECT EXTRACT(YEAR FROM invoice.txndate) AS year,
EXTRACT(MONTH FROM invoice.txndate) AS month,
EXTRACT(DAY FROM invoice.txndate) AS day,
(invoice.customerref_name) AS customers,
MAX(CAST(DATE(invoice.txndate) AS DATE)) over (partition by customerref_name) last_order_date,
(invoice.totalamt - invoice.txntaxdetail_totaltax) AS sales,
row_number() over (partition by id order by blendo_imported_at) row_number,
MIN(CAST(DATE(invoice.txndate) AS DATE)) over (partition by customerref_name) first_order_date,
txndate as `full_date`,

r.description, r.amount, r.invoice_id, r.qty,
CASE WHEN REGEXP_CONTAINS(LOWER(r.description), "weft") THEN 1*r.qty ELSE 0 END AS `weft`,
CASE WHEN REGEXP_CONTAINS(LOWER(r.description), "poly") THEN 1*r.qty ELSE 0 END AS `poly`,
CASE WHEN REGEXP_CONTAINS(LOWER(r.description), "cylinder") THEN 1*r.qty ELSE 0 END AS `cylinder`,
CASE WHEN REGEXP_CONTAINS(LOWER(r.description), "ionix") THEN 1*r.qty ELSE 0 END AS `IONIX`

FROM `blendo.hc_invoice` as invoice

JOIN
(
SELECT description, amount, invoice_id, salesitemlinedetail_qty as `qty`

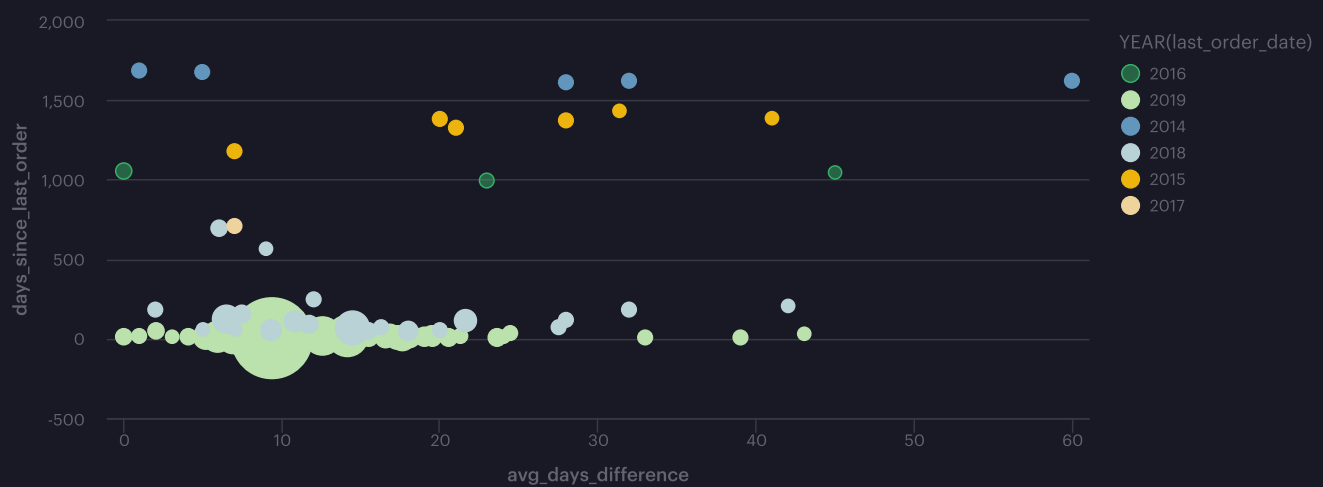
FROM `blendo.hc_invoice_lines`
-- color ring?
WHERE --description <> "Professional Color Ring"
--AND
description <> "Shipping Loss Damage Insurance"
AND amount > 0
AND description IS NOT NULL AND amount IS NOT NULL
)
r
```

```

ON invoice.id = r.invoice_id
)
GROUP BY invoice_id, customers
ORDER BY customers
--
)
)
WHERE previous_date IS NOT NULL
GROUP BY customers
HAVING SUM(fit) < 1

```

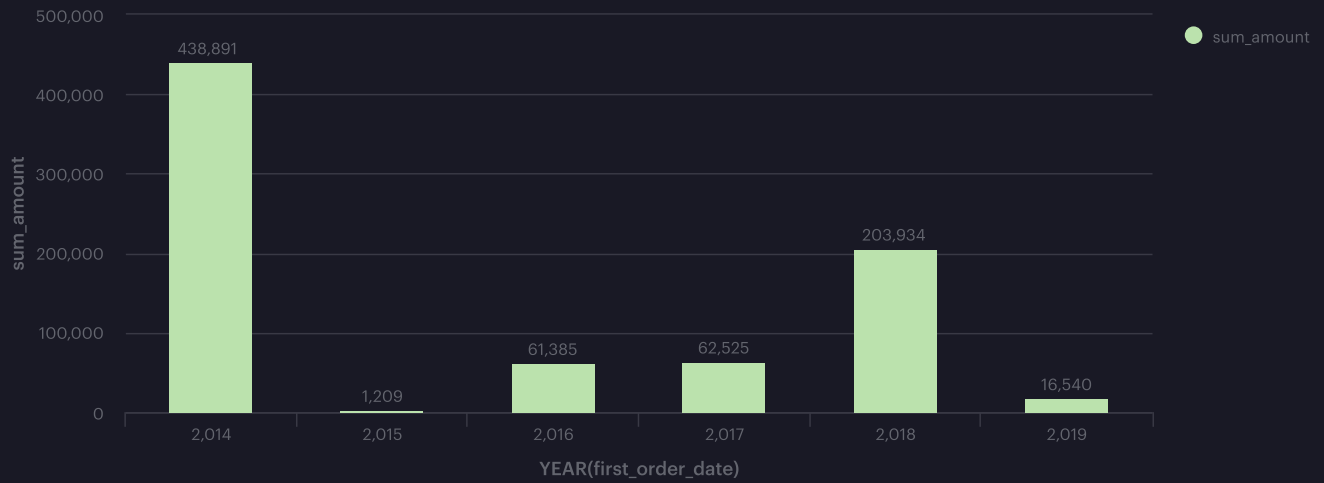
Average order frequency vs. days since last order (We are interested in the segment that ordered freq...



Frequently ordering customers cohort by year vs. total Cylinder items purchased



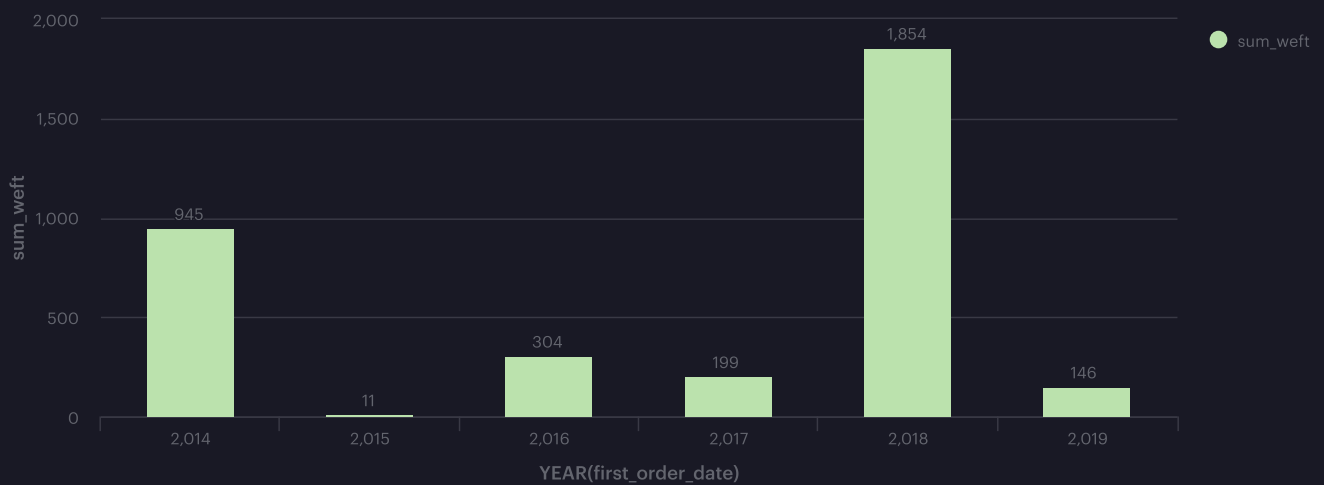
Frequently ordering customers cohort by year vs. total amount spent



Frequently ordering customers cohort by year vs. Poly items



Frequently ordering customers cohort by year vs. Weft items



Churned customers cohort (stopped ordering) by year vs. Weft items



Churned customers cohort (stopped ordering) by year vs. Polymer items



Count of orders v/s days since last ordered

