



Queries used in the Analysis

Created two views to facilitate the analysis of the data:

ic_agg_curr, ic_agg_prior

```
● ● ●  
CREATE VIEW ic_agg_curr AS  
SELECT product_id,  
       COUNT(*) AS total_units_sold,  
       ROUND(AVG(add_to_cart_order)) AS avg_cart_order,  
       SUM(reordered) AS total_reorders  
FROM ic_order_products_curr  
GROUP BY product_id;  
  
CREATE VIEW ic_agg_prior AS  
SELECT product_id,  
       COUNT(*) AS total_units_sold,  
       ROUND(AVG(add_to_cart_order)) AS avg_cart_order,  
       SUM(reordered) AS total_reorders  
FROM ic_order_products_prior  
GROUP BY product_id;
```

Department	Total Aisles	Total Products	Total Sold in Q3	Total Sold in Q2
produce	5	1684	3005	2913
dairy eggs	10	3449	1538	1647
snacks	11	6264	927	862
beverages	8	4365	770	801
frozen	11	4007	714	746
pantry	12	5371	641	598
canned goods	5	2092	356	362
dry goods pasta	5	1858	327	255
deli	5	1322	309	319
bakery	5	1516	308	354
household	10	3085	239	249
breakfast	4	1115	204	238
meat seafood	7	907	194	225
personal care	17	6563	144	125
babies	4	1081	101	131
missing	1	1258	77	15
international	4	1139	72	86
alcohol	5	1054	37	27
pets	2	972	19	28
other	1	548	10	13
bulk	2	38	8	6
Total	134	49688	10000	10000

Distribution of Products Sold



```

SELECT id.department AS "Department",
       COUNT(DISTINCT aisle_id) AS "Total Aisles",
       COUNT(DISTINCT product_id) AS "Total Products",
       SUM(iac.total_units_sold) AS "Total Sold in Q3",
       SUM(iap.total_units_sold) AS "Total Sold in Q2"
  FROM ic_products ip
 JOIN ic_departments id USING(department_id)
 LEFT JOIN ic_agg_curr iac USING(product_id)
 LEFT JOIN ic_agg_prior iap USING(product_id)
 GROUP BY id.department
 ORDER BY "Total Sold in Q3" DESC,
          "Total Sold in Q2" DESC;
    
```

Cart Size Analysis

```

WITH curr_items AS (SELECT order_id, COUNT(product_id) AS total_items
FROM ic_order_products_prior
GROUP BY 1),

prior_items AS (SELECT order_id, COUNT(product_id) AS total_items
FROM ic_order_products_curr
GROUP BY 1)

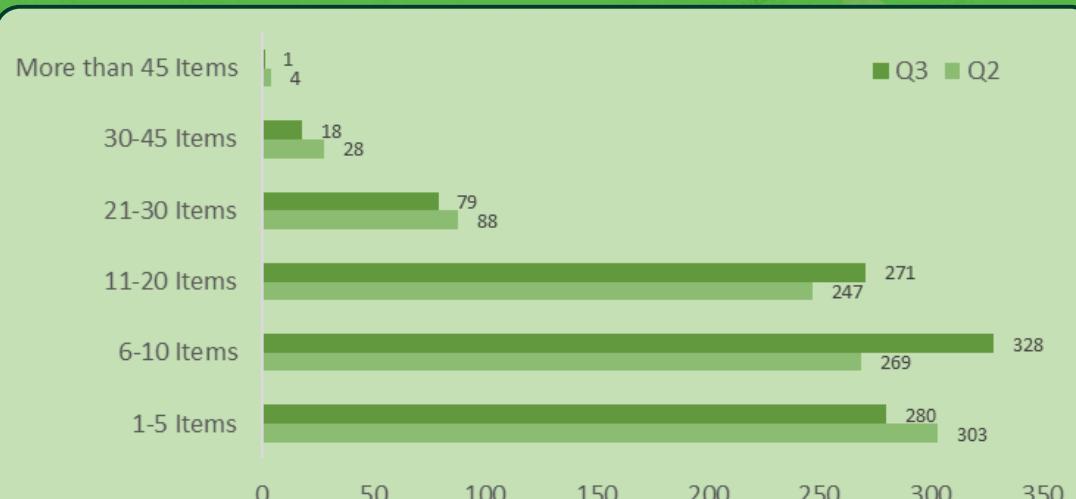
SELECT
    'Q3' AS "Quarter",
    COUNT(CASE WHEN total_items BETWEEN 1 AND 5 THEN TRUE ELSE NULL END) AS "1-5 Items",
    COUNT(CASE WHEN total_items BETWEEN 6 AND 10 THEN TRUE ELSE NULL END) AS "6-10 Items",
    COUNT(CASE WHEN total_items BETWEEN 11 AND 20 THEN TRUE ELSE NULL END) AS "11-20 Items",
    COUNT(CASE WHEN total_items BETWEEN 21 AND 30 THEN TRUE ELSE NULL END) AS "21-30 Items",
    COUNT(CASE WHEN total_items BETWEEN 31 AND 45 THEN TRUE ELSE NULL END) AS "30-45 Items",
    COUNT(CASE WHEN total_items > 45 THEN TRUE ELSE NULL END) AS "More than 45 Items"
FROM curr_items

UNION

SELECT
    'Q2' AS "Quarter",
    COUNT(CASE WHEN total_items BETWEEN 1 AND 5 THEN TRUE ELSE NULL END) AS "1-5 Items",
    COUNT(CASE WHEN total_items BETWEEN 6 AND 10 THEN TRUE ELSE NULL END) AS "6-10 Items",
    COUNT(CASE WHEN total_items BETWEEN 11 AND 20 THEN TRUE ELSE NULL END) AS "11-20 Items",
    COUNT(CASE WHEN total_items BETWEEN 21 AND 30 THEN TRUE ELSE NULL END) AS "21-30 Items",
    COUNT(CASE WHEN total_items BETWEEN 31 AND 45 THEN TRUE ELSE NULL END) AS "30-45 Items",
    COUNT(CASE WHEN total_items > 45 THEN TRUE ELSE NULL END) AS "More than 45 Items"
FROM prior_items;

```

Bin	Q2	Q3
1-5 Items	303	280
6-10 Items	269	328
11-20 Items	247	271
21-30 Items	88	79
30-45 Items	28	18
More than 45 Items	4	1



Top 10 Products (by Units Sold)

```
WITH curr_units_sold AS(
    SELECT *,
        DENSE_RANK() OVER(ORDER BY total_units_sold DESC) AS rank
    FROM ic_agg_curr),

prior_units_sold AS (
    SELECT *,
        DENSE_RANK() OVER(ORDER BY total_units_sold DESC) AS rank
    FROM ic_agg_prior)

SELECT ip.product_name AS "Product",
    id.department AS "Department",
    cus.total_units_sold AS "Total Orders (Q3)",
    cus.rank AS "Rank (Q3)",
    pus.total_units_sold AS "Total Orders (Q2)",
    pus.rank AS "Rank (Q2)"
FROM curr_units_sold cus
LEFT JOIN prior_units_sold pus USING(product_id)
LEFT JOIN ic_products ip USING(product_id)
LEFT JOIN ic_departments id USING(department_id)
WHERE cus.rank<=10
ORDER BY cus.rank;
```

Product	Department	Total Orders (Q3)	Rank (Q3)	Total Orders (Q2)	Rank (Q2)
Banana	produce	124	1	156	1
Organic Strawberries	produce	98	2	73	3
Bag of Organic Bananas	produce	94	3	119	2
Organic Baby Spinach	produce	70	4	70	4
Large Lemon	produce	66	5	46	8
Organic Hass Avocado	produce	54	6	68	5
Organic Avocado	produce	45	7	53	6
Organic Cucumber	produce	44	8	26	18
Limes	produce	43	9	33	12
Strawberries	produce	41	10	52	7

Top 10 Products (by Units Sold)

```
WITH units_sold AS (
    SELECT *,  
        DENSE_RANK() OVER(ORDER BY total_units_sold DESC) AS  
rank  
    FROM ic_agg_curr)  
  
SELECT ip.product_name AS "Product",  
    us.total_units_sold "Total Units Sold"  
FROM units_sold us  
LEFT JOIN ic_products ip USING(product_id)  
WHERE us.rank<=10  
ORDER BY us.total_units_sold DESC;
```

Q2		Q3	
Product	Total Units Sold	Product	Total Units Sold
Banana	156	Banana	124
Bag of Organic Bananas	119	Organic Strawberries	98
Organic Strawberries	73	Bag of Organic Bananas	94
Organic Baby Spinach	70	Organic Baby Spinach	70
Organic Hass Avocado	68	Large Lemon	66
Organic Avocado	53	Organic Hass Avocado	54
Strawberries	52	Organic Avocado	45
Large Lemon	46	Organic Cucumber	44
Organic Yellow Onion	41	Limes	43
Organic Raspberries	39	Strawberries	41

In what order, the top 10 products are being added to the Cart?

Product	Aisle	Total Units Sold	Avg Cart Order
Banana	fresh fruits	124	4
Organic Strawberries	fresh fruits	98	8
Bag of Organic Bananas	fresh fruits	94	5
Organic Baby Spinach	packaged vegetables fruits	70	7
Large Lemon	fresh fruits	66	9
Organic Hass Avocado	fresh fruits	54	7
Organic Avocado	fresh fruits	45	6
Organic Cucumber	fresh vegetables	44	11
Limes	fresh fruits	43	11
Strawberries	fresh fruits	41	7



```
WITH curr_units_sold AS (
    SELECT product_id, total_units_sold, avg_cart_order,
           DENSE_RANK() OVER(ORDER BY total_units_sold DESC) AS rank
    FROM ic_agg_curr)

SELECT ip.product_name AS "Product",
       ia.aisle AS "Aisle",
       cus.total_units_sold AS "Total Units Sold",
       cus.avg_cart_order AS "Avg Cart Order"
FROM curr_units_sold cus
LEFT JOIN ic_products ip USING(product_id)
LEFT JOIN ic_aisles ia USING(aisle_id)
LEFT JOIN ic_departments id USING(department_id)
WHERE cus.rank<=10
ORDER BY cus.rank;
```

Top sellers join carts late (4th+ position). So, what fills carts early?



```
SELECT ia.aisle AS "Aisle",
       COUNT(product_id) AS "Total Products"
  FROM ic_agg_curr c
 LEFT JOIN ic_products ip USING(product_id)
LEFT JOIN ic_departments id
USING(id)
 WHERE c.avg_cart_order=1
 GROUP BY ia.aisle
 ORDER BY COUNT(product_id) DESC;
```

Aisle	Total Products	Aisle	Total Products	Aisle	Total Products	Aisle	Total Products
yogurt	10	cold flu allergy	3	eggs	2	tofu meat alternatives	1
baking ingredients	10	condiments	3	other	2	other creams cheeses	1
juice nectars	9	diapers wipes	3	grains rice dried goods	2	digestion	1
tea	8	dog food care	3	pasta sauce	2	packaged meat	1
ice cream ice	8	dry pasta	3	cookies cakes	2	deodorants	1
packaged vegetables fruit	7	first aid	3	plates bowls cups flatware	2	trail mix snack mix	1
soft drinks	6	fresh vegetables	3	beers coolers	2	poultry counter	1
refrigerated	6	frozen appetizers sides	3	canned meals beans	1	trash bags liners	1
chips pretzels	6	frozen pizza	3	refrigerated pudding dess	1	preserved dips spreads	1
soy lactosefree	6	hot dogs bacon sausage	3	salad dressing toppings	1	protein meal replacement	1
fresh fruits	6	instant foods	3	skin care	1		
crackers	5	milk	3	soap	1		
packaged cheese	5	oils vinegars	3	fresh dips tapenades	1		
candy chocolate	5	oral hygiene	3	feminine care	1		
energy granola bars	5	paper goods	3	breakfast bars pastries	1		
frozen meals	5	prepared soups salads	3	specialty cheeses	1		
coffee	5	spices seasonings	3	frozen breakfast	1		
cream	4	spreads	3	facial care	1		
water seltzer sparkling wa	4	tortillas flat bread	3	bakery desserts	1		
laundry	4	vitamins supplements	3	fruit vegetable snacks	1		
pickled goods olives	4	butter	2	granola	1		
popcorn jerky	4	soup broth bouillon	2	spirits	1		
prepared meals	3	cleaning products	2	energy sports drinks	1		
baby food formula	3	breakfast bakery	2	frozen produce	1		
bread	3	asian foods	2	kitchen supplies	1		
buns rolls	3	nuts seeds dried fruit	2	doughs gelatins bake mix	1		
canned jarred vegetables	3	missing	2	lunch meat	1		
canned meat seafood	3	packaged produce	2	marinades meat preparat	1		
cereal	3	white wines	2	more household	1		

What about the rest of the Add to Cart Orders?



```
SELECT avg_cart_order AS "Avg Cart Order",
       COUNT(product_id) AS "Total Products",
       SUM(total_units_sold) AS "Total Units
Sold",SUM(total_reorders) AS "Total Reorders"
FROM ic_agg_curr
GROUP BY avg_cart_order
ORDER BY avg_cart_order;
```

Avg Cart Order	Total Units Sold	Total Reorders
1	296	173
2	331	217
3	489	304
4	662	438
5	690	427
6	737	441
7	1006	652
8	888	568
9	807	495
10	884	501
11	740	400
12	458	233
13	387	192
14	243	124
15	217	99
16	205	98
17	124	60
18	172	74
19	103	50
20	81	35
21	81	34
22	63	26
23	47	22
24	47	22
25	29	6
26	23	10

Reordering Behavior Change



```
WITH pct_change_reorders AS (
  SELECT ip.product_name,
    ia.aisle,
    iap.total_reorders as prior_reorders,
    iac.total_reorders as curr_reorders,
    ROUND((iac.total_reorders -
    iap.total_reorders)/iap.total_reorders) * 100 AS change_pct
  FROM ic_agg_curr iac
  LEFT JOIN ic_agg_prior iap USING(product_id)
  LEFT JOIN ic_products ip USING(product_id)
  LEFT JOIN ic_aisles ia USING(aisle_id)
  WHERE (iap.total_reorders >= 5 AND iac.total_reorders >= 5)
  ORDER BY 4 DESC)

SELECT product_name AS "Product",
  aisle AS "Aisle",
  prior_reorders AS "Total Reorders (Q2)",
  curr_reorders AS "Total Reorders (Q3)",
  change_pct || '%' AS "% Change"
FROM pct_change_reorders
WHERE ABS(change_pct) >= 50
ORDER BY change_pct DESC;
```

Product	Aisle	Total Reorders (Q2)	Total Reorders (Q3)	% Change
Small Hass Avocado	fresh fruits	5	21	320%
Organic Broccoli	fresh vegetables	8	16	100%
Red Vine Tomato	fresh vegetables	8	15	88%
Organic Spring Mix	packaged vegetables	5	9	80%
Organic Grade A Free Range Large Eggs	eggs	7	12	71%
Large Lemon	fresh fruits	26	44	69%
Organic Lacinato (Dinosaur) Kale	fresh vegetables	8	13	63%
Organic Ginger Root	fresh vegetables	8	13	63%
Lightly Salted Baked Snap Pea Chips	pretzels	5	8	60%
Organic Red Bell Pepper	fresh vegetables	11	17	55%
Yellow Onions	fresh vegetables	10	15	50%
Organic Black Beans	canned meals beans	8	12	50%
Pure Irish Butter	butter	6	9	50%
Original Hummus	fresh dips tapenades	22	11	-50%
Organic Blackberries	fresh fruits	18	9	-50%
Green Bell Pepper	fresh vegetables	14	6	-57%
Organic Whole Strawberries	frozen produce	14	6	-57%
Organic Whole String Cheese	packaged cheese	14	6	-57%
Blueberries	frozen produce	17	7	-59%
Roma Tomato	fresh vegetables	13	5	-62%
Half & Half	cream	16	5	-69%



Thank you for your Time!

 Explore more about the Case Study on [GitHub](#)

 Explore other projects on my [Portfolio](#)

 Wants to know more about me on [LinkedIn](#)