E-COMMERCE DATA ANALYSIS

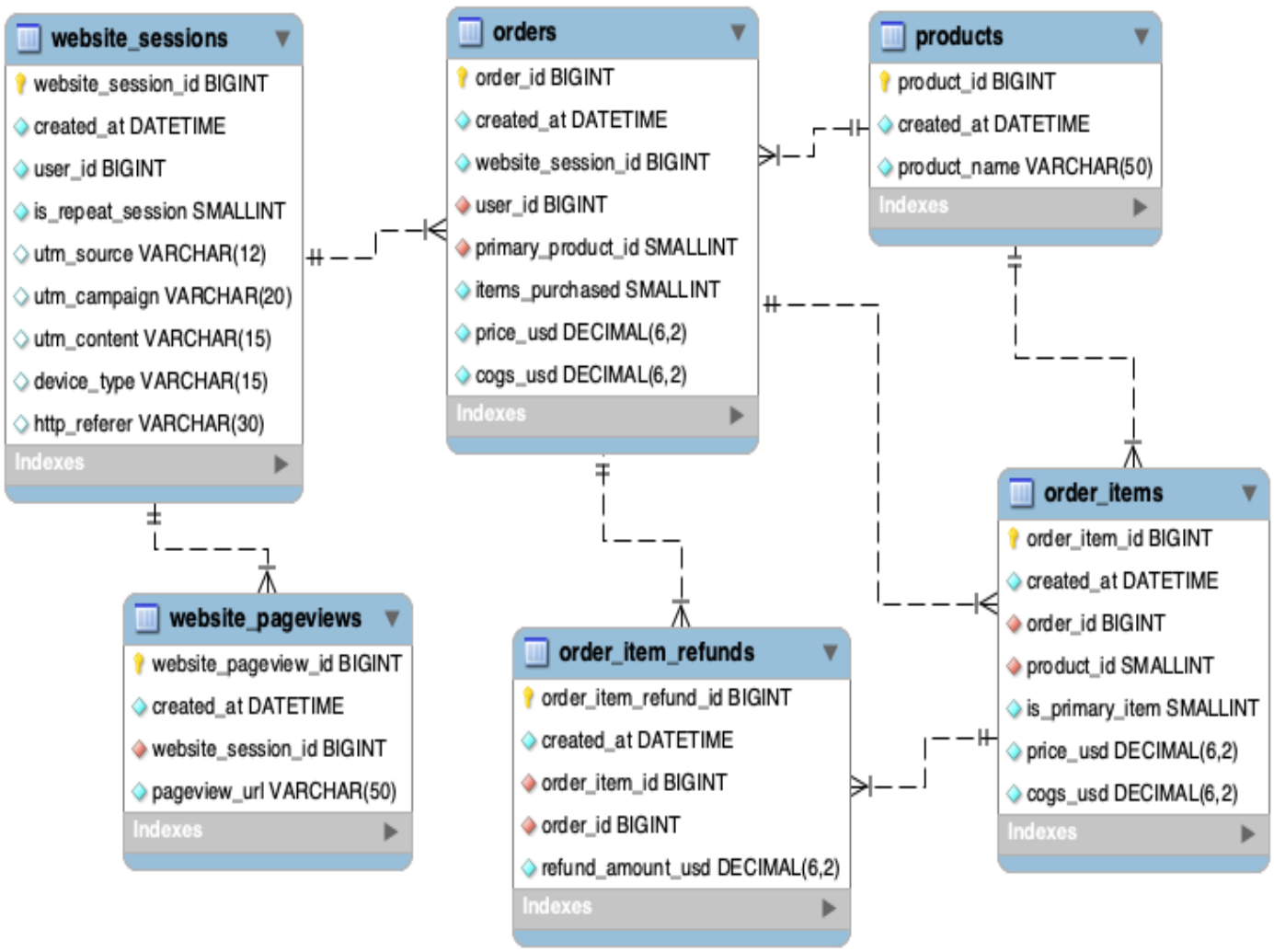
About Project:

Maven fuzzy factory, an online retailer, launched their first product in March 2012. The company has been live for roughly 8 months now and your CEO is due to present company performance metrics to the board next week. As an Analyst, you are tasked with preparing relevant metrics to show the company's promising growth.
Your objective is to extract and analyze website traffic and performance data from the Maven Fuzzy Factory database to quantify the company's growth, and to tell the story of how you have been able to generate that growth.

Tech-Stack:

- MySQL has been used to analyze the data.
- Tableau has been used for visualisation.

Entity Relationship Diagram:



Data Tables:

a. **Website_sessions:**

website_session_id	created_at	user_id	is_repeat_session	utm_source	utm_campaign	utm_content	device_...	http_referer
1	2012-03-19 08:04:16	1	0	gsearch	nonbrand	g_ad_1	mobile	https://www.gsearch.com
2	2012-03-19 08:16:49	2	0	gsearch	nonbrand	g_ad_1	desktop	https://www.gsearch.com
3	2012-03-19 08:26:55	3	0	gsearch	nonbrand	g_ad_1	desktop	https://www.gsearch.com
4	2012-03-19 08:37:33	4	0	gsearch	nonbrand	g_ad_1	desktop	https://www.gsearch.com
5	2012-03-19 09:00:55	5	0	gsearch	nonbrand	g_ad_1	mobile	https://www.gsearch.com

b. Website_pageviews:

website_pageview_id	created_at	website_session_id	pageview_url
7	2012-03-19 09:06:27	7	/home
8	2012-03-19 09:10:08	6	/products
9	2012-03-19 09:10:52	6	/the-original-mr-fuzzy
10	2012-03-19 09:14:02	6	/cart
11	2012-03-19 09:16:52	6	/shipping

c. Products:

product_id	created_at	product_name
1	2012-03-19 08:00:00	The Original Mr. Fuzzy
2	2013-01-06 13:00:00	The Forever Love Bear
3	2013-12-12 09:00:00	The Birthday Sugar Panda
4	2014-02-05 10:00:00	The Hudson River Mini bear

`

d. Orders:

order_id	created_at	website_session_id	user_id	primary_product_id	items_purchased	price_usd	cogs_usd
1	2012-03-19 10:42:46	20	20	1	1	49.99	19.49
2	2012-03-19 19:27:37	104	104	1	1	49.99	19.49
3	2012-03-20 06:44:45	147	147	1	1	49.99	19.49
4	2012-03-20 09:41:45	160	160	1	1	49.99	19.49
5	2012-03-20 11:28:15	177	177	1	1	49.99	19.49

e. Order_items:

order_item_id	created_at	order_id	product_id	is_primary_item	price_usd	cogs_usd
1	2012-03-19 10:42:46	1	1	1	49.99	19.49
2	2012-03-19 19:27:37	2	1	1	49.99	19.49
3	2012-03-20 06:44:45	3	1	1	49.99	19.49
4	2012-03-20 09:41:45	4	1	1	49.99	19.49
5	2012-03-20 11:28:15	5	1	1	49.99	19.49

f. Order_item_refunds:

order_item_refund_id	created_at	order_item_id	order_id	refund_amount_usd
1	2012-04-06 11:32:43	57	57	49.99
2	2012-04-13 01:09:43	74	74	49.99
3	2012-04-15 07:03:48	71	71	49.99
4	2012-04-17 20:00:37	118	118	49.99
5	2012-04-22 20:53:49	116	116	49.99

Project Taks:

1. **Pull the monthly trends for all our channels to understand how each of our channels are performing.**

First, finding out various utm sources and referers to see the traffic we are getting.

[UTM (Urchin Tracking Module) parameters are used by marketers to track the effectiveness of online marketing campaigns across traffic sources. You add them to URLs to help you track the performance of a webpage or a campaign]

```
SELECT DISTINCT
    utm_source,
    utm_campaign,
    http_referer
FROM website_sessions
WHERE website_sessions.created_at < '2012-11-27';
```

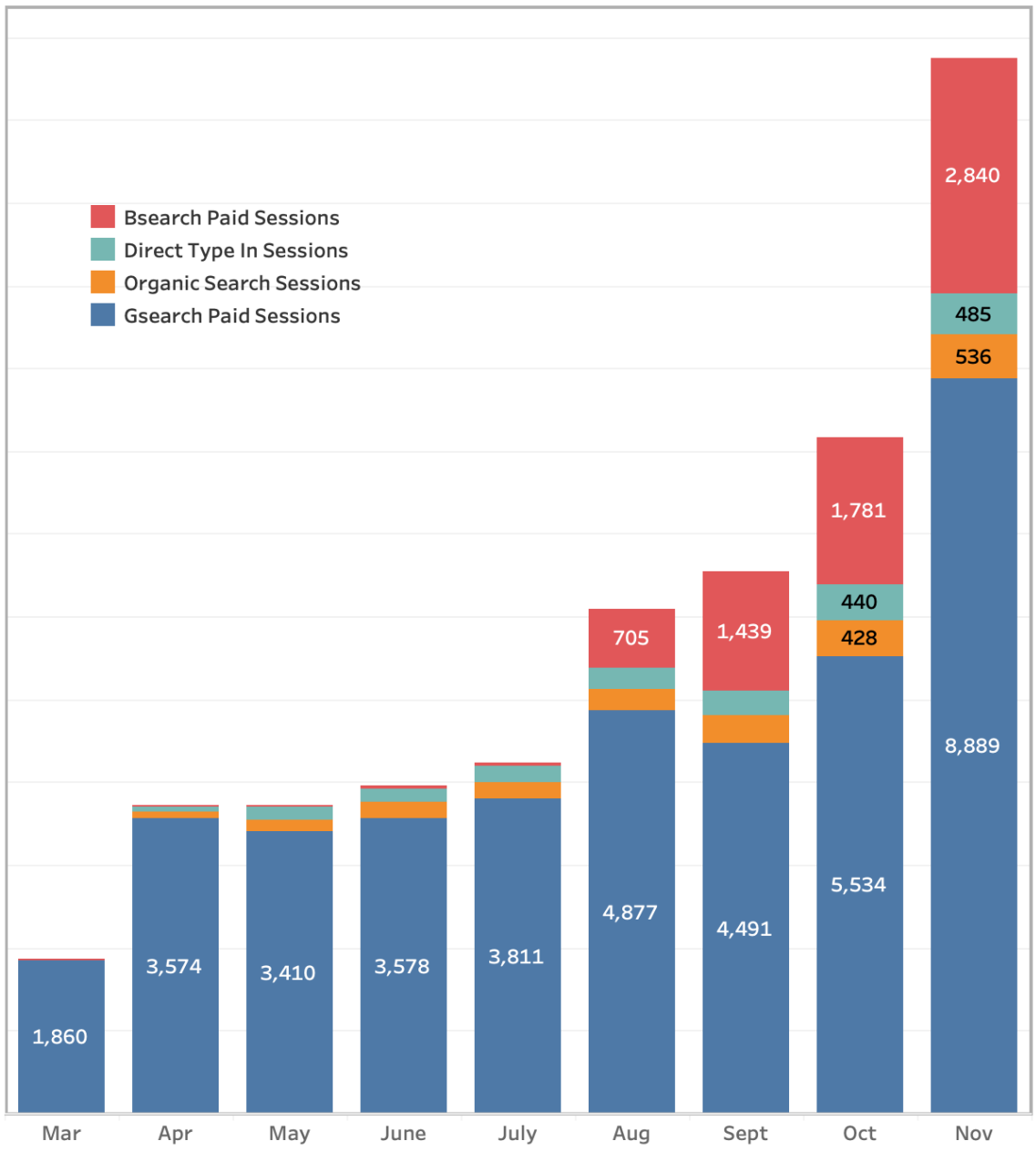
utm_source	utm_campaign	http_referer
gsearch	nonbrand	https://www.gsearch.com
NULL	NULL	NULL
gsearch	brand	https://www.gsearch.com
NULL	NULL	https://www.gsearch.com
bsearch	brand	https://www.bsearch.com
NULL	NULL	https://www.bsearch.com
bsearch	nonbrand	https://www.bsearch.com

Now, checking the performance of each of our channels.

```
SELECT
    YEAR(WS.created_at) AS yr,
    MONTH(WS.created_at) AS mo,
    COUNT(DISTINCT CASE WHEN utm_source = 'gsearch' THEN WS.website_session_id ELSE NULL END) AS gsearch_paid_sessions,
    COUNT(DISTINCT CASE WHEN utm_source = 'bsearch' THEN WS.website_session_id ELSE NULL END) AS bsearch_paid_sessions,
    COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NOT NULL
        THEN WS.website_session_id ELSE NULL END) AS organic_search_sessions,
    COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NULL
        THEN WS.website_session_id ELSE NULL END) AS direct_type_in_sessions
FROM website_sessions AS WS
LEFT JOIN orders
ON orders.website_session_id = WS.website_session_id
WHERE WS.created_at < '2012-11-27'
GROUP BY 1,2;
```

yr	mo	gsearch_paid_sessions	bsearch_paid_sessions	organic_search_sessions	direct_type_in_sessions
2012	3	1860	2	8	9
2012	4	3574	11	78	71
2012	5	3410	25	150	151
2012	6	3578	25	190	170
2012	7	3811	44	207	187
2012	8	4877	705	265	250
2012	9	4491	1439	331	285
2012	10	5534	1781	428	440
2012	11	8889	2840	536	485

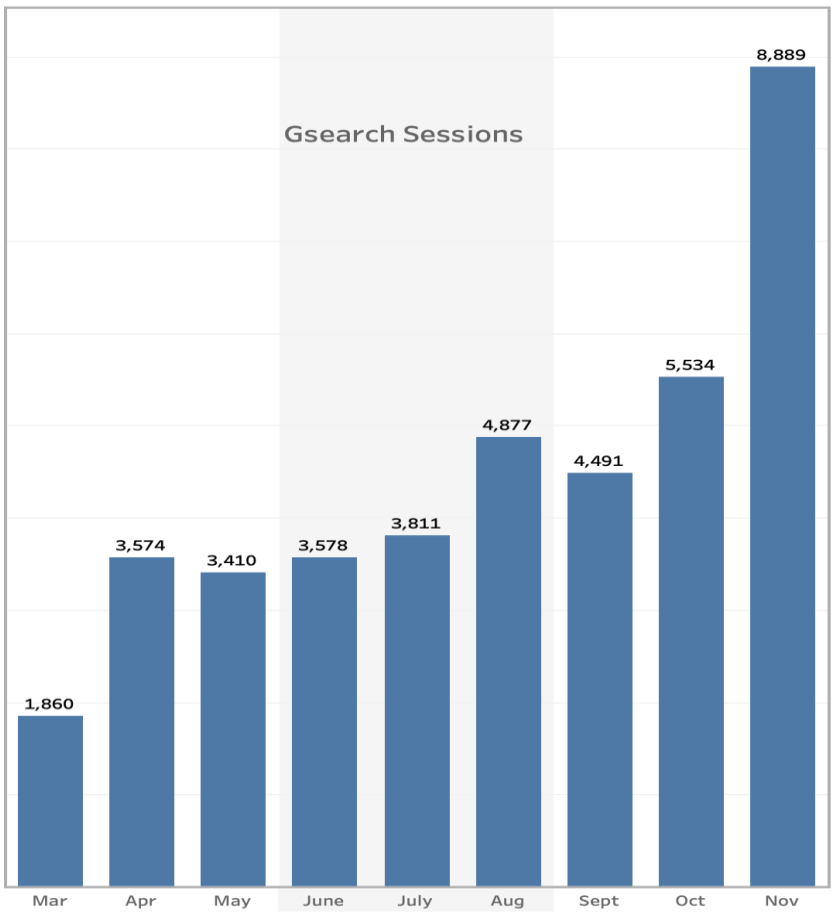
[Organic & direct_type_in sessions are sessions which are not paid for.
<u>Organic sessions</u>: When a user lands on your website through a link found on search engine result and not through paid advertisement.
<u>Direct type in sessions</u>: When a user lands on your website by directly typing website URL in search engine.]

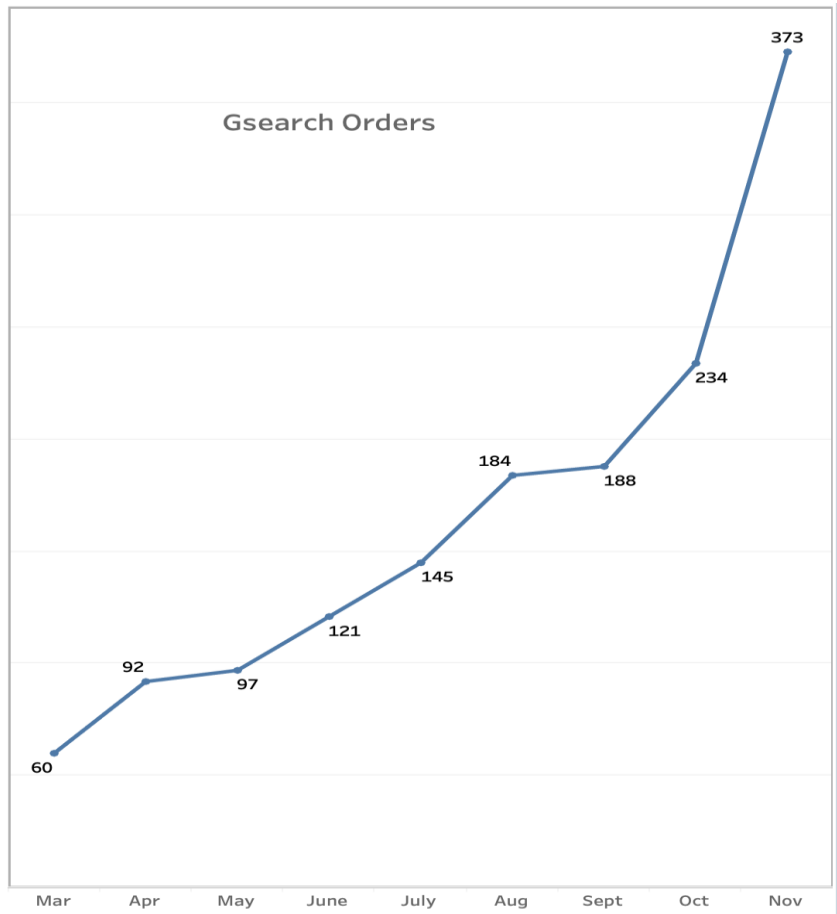


2. **Gsearch Seems to be the biggest driver of our business. Could you pull monthly trends for gsearch sessions and orders so that we can showcase the growth here?**

```sql
SELECT
    YEAR(website_sessions.created_at) AS yr,
    MONTH(website_sessions.created_at) AS mo,
    COUNT(DISTINCT website_sessions.website_session_id) AS sessions,
    COUNT(DISTINCT orders.order_id) AS orders,
    COUNT(DISTINCT orders.order_id)/COUNT(DISTINCT website_sessions.website_session_id) AS conv_rate
FROM website_sessions
LEFT JOIN orders
ON orders.website_session_id = website_sessions.website_session_id
WHERE website_sessions.created_at < '2012-11-27'
    AND website_sessions.utm_source = 'gsearch'
GROUP BY 1,2;
```

yr	mo	sessions	orders	conv_rate
2012	3	1860	60	0.0323
2012	4	3574	92	0.0257
2012	5	3410	97	0.0284
2012	6	3578	121	0.0338
2012	7	3811	145	0.0380
2012	8	4877	184	0.0377
2012	9	4491	188	0.0419
2012	10	5534	234	0.0423
2012	11	8889	373	0.0420



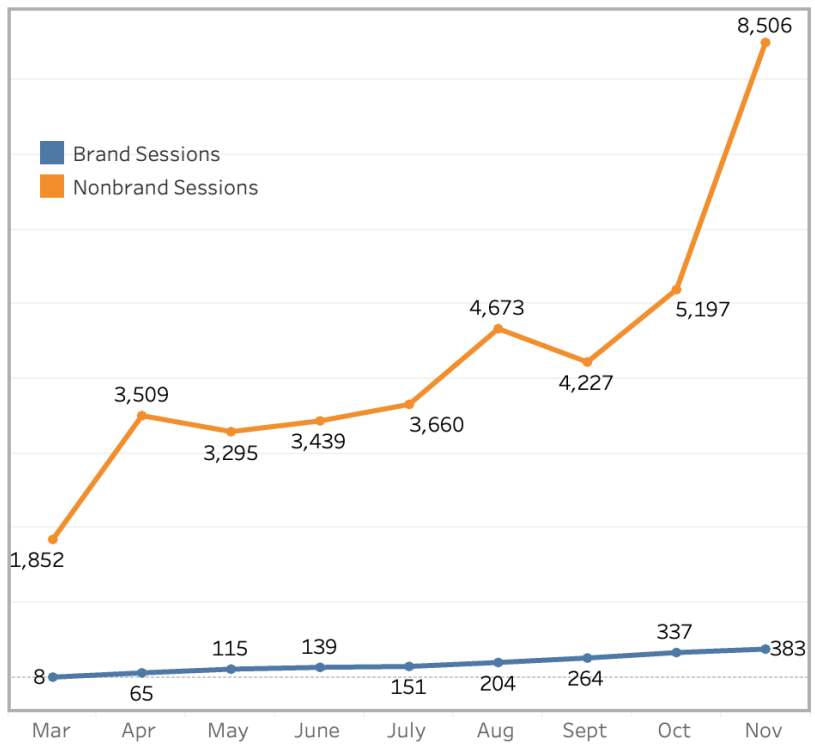


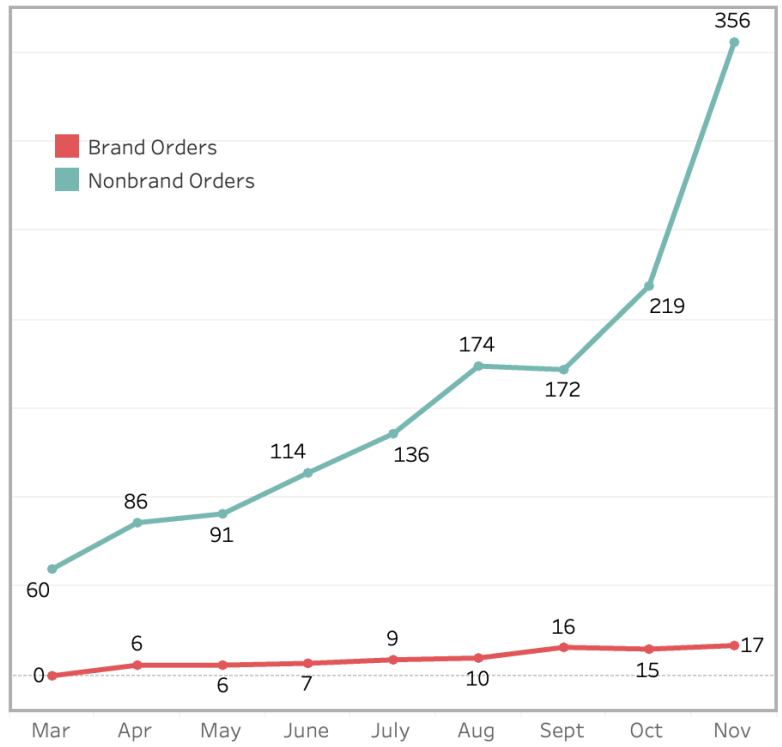
3. **Next, it would be great to see a similar monthly trend for Gsearch, but this time splitting out nonbrand and brand campaigns separately. I am wondering if brand is picking up at all. If so, this is a good story to tell.**

[<u>Brand campaign</u> : Such campaigns include brand keywords such as name of the company, products etc. in order to increase the brand awareness.
<u>Non brand campaign</u> : Such campaigns don't' specifically uses company name but reach customers that could have otherwise not been include in company's market funnel]

```sql
SELECT
    YEAR(website_sessions.created_at) AS yr,
    MONTH(website_sessions.created_at) AS mo,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' THEN website_sessions.website_session_id ELSE NULL END) AS nonbrand_sessions
    COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' THEN orders.order_id ELSE NULL END) AS nonbrand_orders,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN website_sessions.website_session_id ELSE NULL END) AS brand_sessions,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN orders.order_id ELSE NULL END) AS brand_orders
FROM website_sessions
    LEFT JOIN orders
        ON orders.website_session_id = website_sessions.website_session_id
WHERE website_sessions.created_at < '2012-11-27'
    AND website_sessions.utm_source = 'gsearch'
GROUP BY 1,2;
```

yr	mo	nonbrand_sessions	nonbrand_orders	brand_sessions	brand_orders
2012	3	1852	60	8	0
2012	4	3509	86	65	6
2012	5	3295	91	115	6
2012	6	3439	114	139	7
2012	7	3660	136	151	9
2012	8	4673	174	204	10
2012	9	4227	172	264	16
2012	10	5197	219	337	15
2012	11	8506	356	383	17

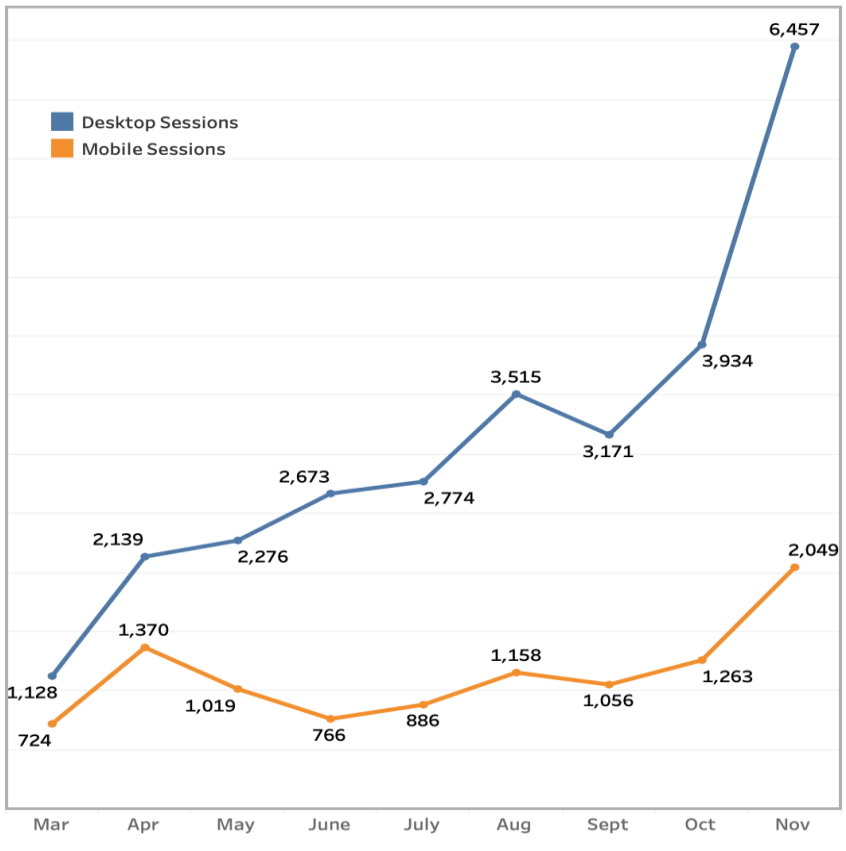


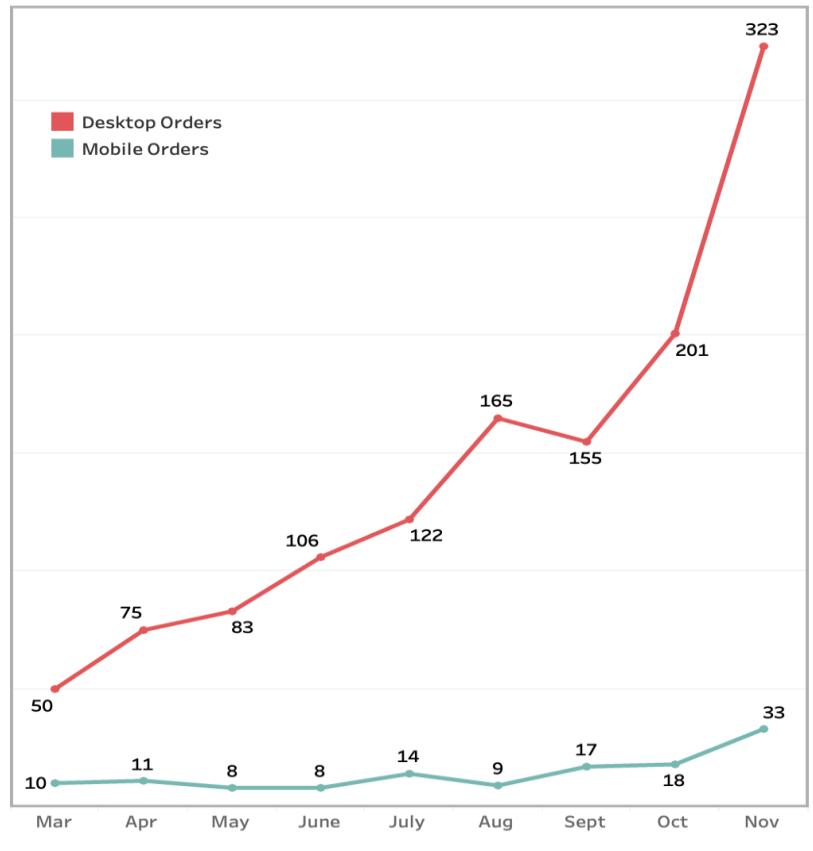


4. **While we are on gsearch, could you dive into nonbrand, and pull monthly sessions and orders split by device type? I want to flex our analytical muscles a little and show the board we really know our traffic sources.**

```sql
SELECT
    YEAR(website_sessions.created_at) AS yr,
    MONTH(website_sessions.created_at) AS mo,
    COUNT(DISTINCT CASE WHEN device_type = 'desktop' THEN website_sessions.website_session_id ELSE NULL END) AS desktop_sessions,
    COUNT(DISTINCT CASE WHEN device_type = 'desktop' THEN orders.order_id ELSE NULL END) AS desktop_orders,
    COUNT(DISTINCT CASE WHEN device_type = 'mobile' THEN website_sessions.website_session_id ELSE NULL END) AS mobile_sessions,
    COUNT(DISTINCT CASE WHEN device_type = 'mobile' THEN orders.order_id ELSE NULL END) AS mobile_orders
FROM website_sessions
LEFT JOIN orders
ON orders.website_session_id = website_sessions.website_session_id
WHERE website_sessions.created_at < '2012-11-27'
    AND website_sessions.utm_source = 'gsearch'
    AND website_sessions.utm_campaign = 'nonbrand'
GROUP BY 1,2;
```

yr	mo	desktop_sessions	desktop_orders	mobile_sessions	mobile_orders
2012	3	1128	50	724	10
2012	4	2139	75	1370	11
2012	5	2276	83	1019	8
2012	6	2673	106	766	8
2012	7	2774	122	886	14
2012	8	3515	165	1158	9
2012	9	3171	155	1056	17
2012	10	3934	201	1263	18
2012	11	6457	323	2049	33

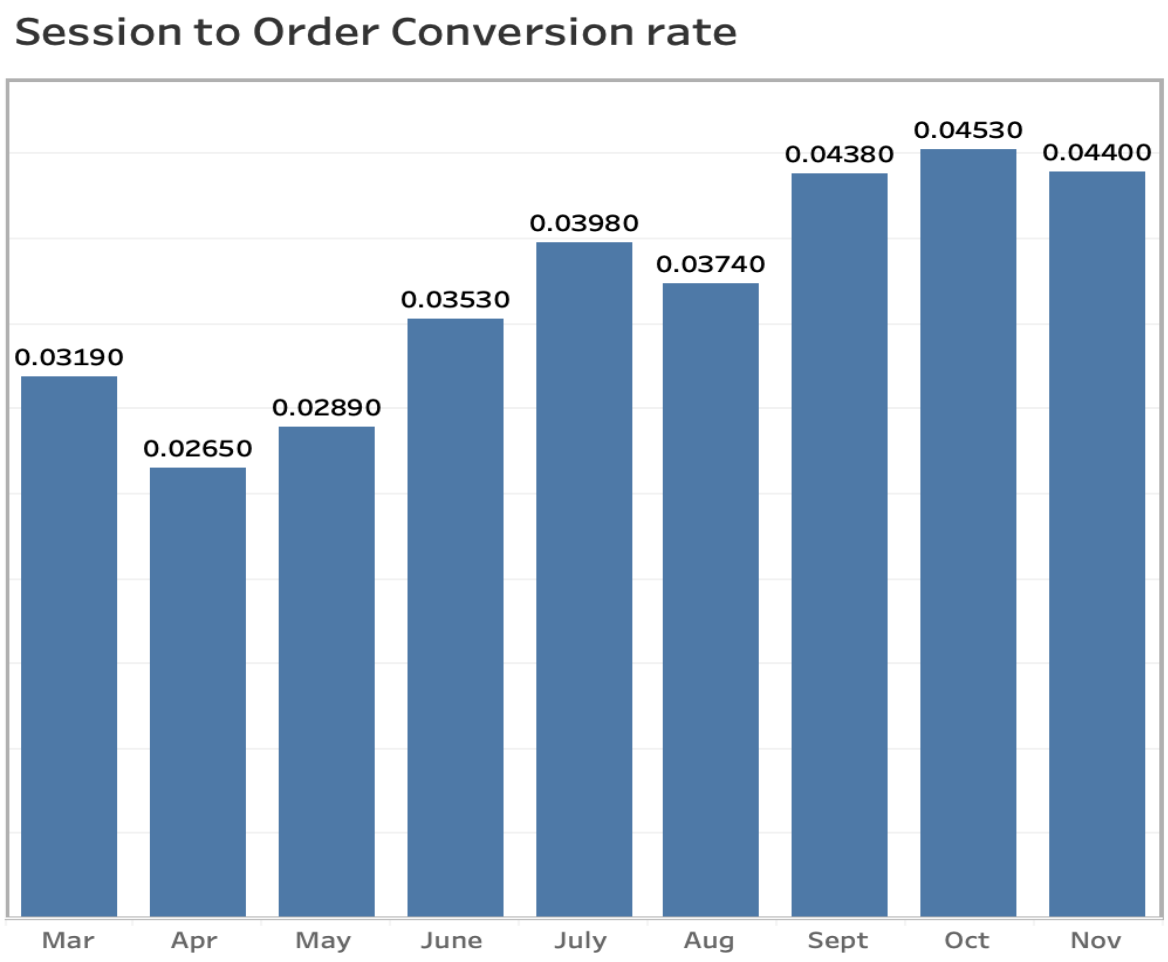




5. I'd like to tell the story of our website performance improvements over the course of the first 8 months. Could you pull session to order conversion rates, by month?

```sql
SELECT
    YEAR(website_sessions.created_at) AS yr,
    MONTH(website_sessions.created_at) AS mo,
    COUNT(DISTINCT website_sessions.website_session_id) AS sessions,
    COUNT(DISTINCT orders.order_id) AS orders,
    COUNT(DISTINCT orders.order_id)/COUNT(DISTINCT website_sessions.website_session_id) AS conversion_rate
FROM website_sessions
    LEFT JOIN orders
        ON orders.website_session_id = website_sessions.website_session_id
WHERE website_sessions.created_at < '2012-11-27'
GROUP BY 1,2;
```

yr	mo	sessions	orders	conversion_rate
2012	3	1879	60	0.0319
2012	4	3734	99	0.0265
2012	5	3736	108	0.0289
2012	6	3963	140	0.0353
2012	7	4249	169	0.0398
2012	8	6097	228	0.0374
2012	9	6546	287	0.0438
2012	10	8183	371	0.0453
2012	11	12750	561	0.0440



6. For the gsearch lander test, please estimate the revenue that test earned us.
(Hint: Look at the increase in CVR from the test (Jun 19 – Jul 28), and use nonbrand sessions and revenue since then to calculate incremental value)

First, finding out the first website pageview id when the landing page was '/lander-1'.

```
SELECT
    MIN(website_pageview_id) AS first_test_pv
FROM website_pageviews
WHERE pageview_url = '/lander-1';
```

First website_pageview_id = 23504
Now, getting session to order conversion rate for both landing pages i.e. 'home' & 'lander-1'

```sql
SELECT
    pageview_url AS landing_page,
    COUNT(DISTINCT website_sessions.website_session_id) sessions,
    COUNT(DISTINCT orders.order_id) orders,
    COUNT(DISTINCT orders.order_id)/COUNT(DISTINCT website_sessions.website_session_id) cnv_rate
FROM website_sessions
INNER JOIN website_pageviews
ON website_pageviews.website_session_id = website_sessions.website_session_id
LEFT JOIN orders
ON orders.website_session_id = website_sessions.website_session_id
WHERE website_pageview_id >= 23504
AND website_sessions.created_at < '2012-07-28'
AND utm_source = 'gsearch' AND utm_campaign = 'nonbrand'
AND pageview_url IN ('/home','/lander-1')
GROUP BY 1
```

landing_page	sessions	orders	cnv_rate
/home	2261	72	0.0318
/lander-1	2316	94	0.0406

Sessions to orders conversion rate for '/home' landing page = 0.0318
Sessions to orders conversion rate for '/lander-1' landing page = 0.0406
Extra orders per session for '/lander-1' landing page = 0.0406 - 0.0318 = 0.0088

Now, finding the most recent pageview id for gsearch nonbrand when traffic was sent to '/home' landing page

```sql
SELECT MAX(website_sessions.website_session_id) most_recent_session_with_home_landing_page
FROM website_sessions
INNER JOIN website_pageviews
ON website_pageviews.website_session_id = website_sessions.website_session_id
WHERE website_sessions.created_at < '2012-11-27'
AND utm_source = 'gsearch'
AND utm_campaign = 'nonbrand'
AND pageview_url = '/home';
```

most_recent_session_with_home_landing_page
17145

Most recent session id with '/home' landing page = 17145. After this session, all the sessions are rerouted to '/lander-1'

Now, finding total number of sessions we have had since rerouting to '/lander-1'

```
SELECT COUNT(DISTINCT(website_session_id)) total_sessions
FROM website_sessions
WHERE website_session_id > 17145
AND created_at < '2012-11-27'
AND utm_source = 'gsearch'
AND utm_campaign = 'nonbrand';
```

total_sessions
22972

Total number of sessions since rerouting = 22972
Extra/incremental orders per session for '/lander-1' = 0.0088
Total extra orders for '/lander-1' from '2012-07-28' to '2012-11-27' = 0.0088 *22972 = 202 orders
So, we have roughly 50 extra orders per month for '/lander-1'

7. **For the landing page test you analyzed previously, it would be great to show a full conversion funnel from each of the two pages to orders. You can use the same time period you analyzed last time (Jun 19 – Jul 28).**

```sql
-- getting landing page for all gsearch nonbrand sessions
WITH sessions_with_landing_page AS(
SELECT
     website_sessions.website_session_id,
     pageview_url AS landing_page
FROM website_sessions
INNER JOIN website_pageviews
ON website_sessions.website_session_id = website_pageviews.website_session_id
WHERE utm_source = 'gsearch' AND utm_campaign = 'nonbrand'
    AND website_sessions.created_at BETWEEN '2012-06-19' AND '2012-07-28'
    AND pageview_url IN ('/home','/lander-1'))
,
-- flagging to know all the pages viewed within a session after landing page
pageview_level_session_flag AS(
SELECT
    website_session_id,
    landing_page,
    MAX(product_page) to_product ,
    MAX(mr_fuzzy_page) to_fuzzy,
    MAX(cart_page) to_cart,
    MAX(shipping_page) to_shipping,
    MAX(billing_page) to_billing,
    MAX(thank_you_page) to_thankyou
FROM
(SELECT
    sessions_with_landing_page.website_session_id,
    sessions_with_landing_page.landing_page,
    CASE WHEN pageview_url = '/products' THEN 1 ELSE 0 END product_page,
    CASE WHEN pageview_url = '/the-original-mr-fuzzy' THEN 1 ELSE 0 END mr_fuzzy_page,
    CASE WHEN pageview_url = '/cart' THEN 1 ELSE 0 END cart_page,
    CASE WHEN pageview_url = '/shipping' THEN 1 ELSE 0 END shipping_page,
    CASE WHEN pageview_url = '/billing' THEN 1 ELSE 0 END billing_page,
    CASE WHEN pageview_url = '/thank-you-for-your-order' THEN 1 ELSE 0 END thank_you_page
FROM sessions_with_landing_page
    INNER JOIN website_pageviews
        ON sessions_with_landing_page.website_session_id = website_pageviews.website_session_id
) AS pageview_level
GROUP BY 1,2)
```

```sql
-- finally getting count of sessions for each pageview_url
SELECT
    landing_page,
    COUNT(DISTINCT website_session_id) sessions,
    COUNT(DISTINCT CASE WHEN to_product = 1 THEN website_session_id ELSE NULL END) made_to_product,
    COUNT(DISTINCT CASE WHEN to_fuzzy = 1 THEN website_session_id ELSE NULL END) made_to_fuzzy,
    COUNT(DISTINCT CASE WHEN to_cart = 1 THEN website_session_id ELSE NULL END) made_to_cart,
    COUNT(DISTINCT CASE WHEN to_shipping = 1 THEN website_session_id ELSE NULL END) made_to_shipping,
    COUNT(DISTINCT CASE WHEN to_billing = 1 THEN website_session_id ELSE NULL END) made_to_billing,
    COUNT(DISTINCT CASE WHEN to_thankyou = 1 THEN website_session_id ELSE NULL END) made_to_thankyou
FROM pageview_level_session_flag
GROUP BY 1
```

landing_page	sessions	made_to_product	made_to_fuzzy	made_to_cart	made_to_shipping	made_to_billing	made_to_thankyou
/home	2261	942	684	296	200	168	72
/lander-1	2316	1083	772	348	231	197	94

Total number of sessions landing on 'Home' page = 2261
Total number of sessions landing on 'lander-1' page = 2316
made_to_product = number of sessions which made it to 'Product' page after clicking through landing_page.
made_to_fuzzy = number of sessions which made it to 'Mr. Fuzzy' page after clicking through product page.
made_to_shipping = number of sessions which made it to 'Shipping' page after clicking through fuzzy page.
made_to_billing = number of sessions which made it to Billing page after clicking through shipping page.
made_to_thankyou = number of sessions in which an order was placed after clicking through billing page.

8. **I'd love for you to quantify the impact of our billing test, as well. Please analyze the lift generated from the test (Sep 10 – Nov 10), in terms of revenue per billing page session, and then pull the number of billing page sessions for the past month to understand monthly impact.**

Old billing page = '/billing'
New billing page = '/billing-2'
LIFT generated = difference between both the billing pages in terms of revenue generated per billing session

First, checking when new billing page '/billing-2' went live

```
SELECT MIN(created_at)
FROM website_pageviews
WHERE pageview_url = '/billing-2';
```

Date when '/billing-2' went live comes as 10 Sept

Now, checking whether the old billing page is live till 10 Nov just to ensure the fair comparison between both the billing pages.

```
SELECT MAX(created_at)
FROM website_pageviews
WHERE pageview_url = '/billing';
```

Seems like old billing page is active till 5 Jan, 2013. Therefore, the comparison between these two billing pages from 10 Sept,2012 to 10 Nov,2012 is fair as both these billing pages are active in this duration.

Now, getting the revenue per billing page session for both these billing pages.

```
SELECT
    billing_version_seen,
    COUNT(DISTINCT website_session_id) AS sessions,
    SUM(price_usd)/COUNT(DISTINCT website_session_id) AS revenue_per_billing_page_seen
 FROM
 (
SELECT
    website_pageviews.website_session_id,
    website_pageviews.pageview_url AS billing_version_seen,
    orders.order_id,
    orders.price_usd
FROM website_pageviews
LEFT JOIN orders
ON orders.website_session_id = website_pageviews.website_session_id
WHERE website_pageviews.created_at > '2012-09-10' -- prescribed in assignment
    AND website_pageviews.created_at < '2012-11-10' -- prescribed in assignment
    AND website_pageviews.pageview_url IN ('/billing','/billing-2')
) AS billing_pageviews_and_order_data
GROUP BY 1;
```

billing_version_seen	sessions	revenue_per_billing_page_seen
/billing	657	22.826484
/billing-2	654	31.339297

Revenue generated per billing page session for old billing page = $ 22.83
Revenue generated per billing page session for new billing page = $ 31.34
LiIFT generated = $ 8.51 per billing session

Finally, getting the number of billing page session for past month ('2012-10-27' to '2012-11-27) to understand the impact

```sql
SELECT
    pageview_url AS billing_page_seen,
    COUNT(website_session_id) AS billing_session_past_month
FROM website_pageviews W
WHERE pageview_url IN ('/billing', '/billing-2')
AND created_at BETWEEN '2012-10-27' AND '2012-11-27'
GROUP BY 1;
```

billing_page_seen	billing_session_past_month
/billing-2	583
/billing	610

1193 sessions last month (583 with billing & 610 with billing-2)
Total revenue = (583*22.83 + 610*31.34) = $ 32427.29
LFT: $8.52 per billing session
Extra revenue generated = billing-2 sessions * LIFT = 610*8.51 = $ 5191.10