

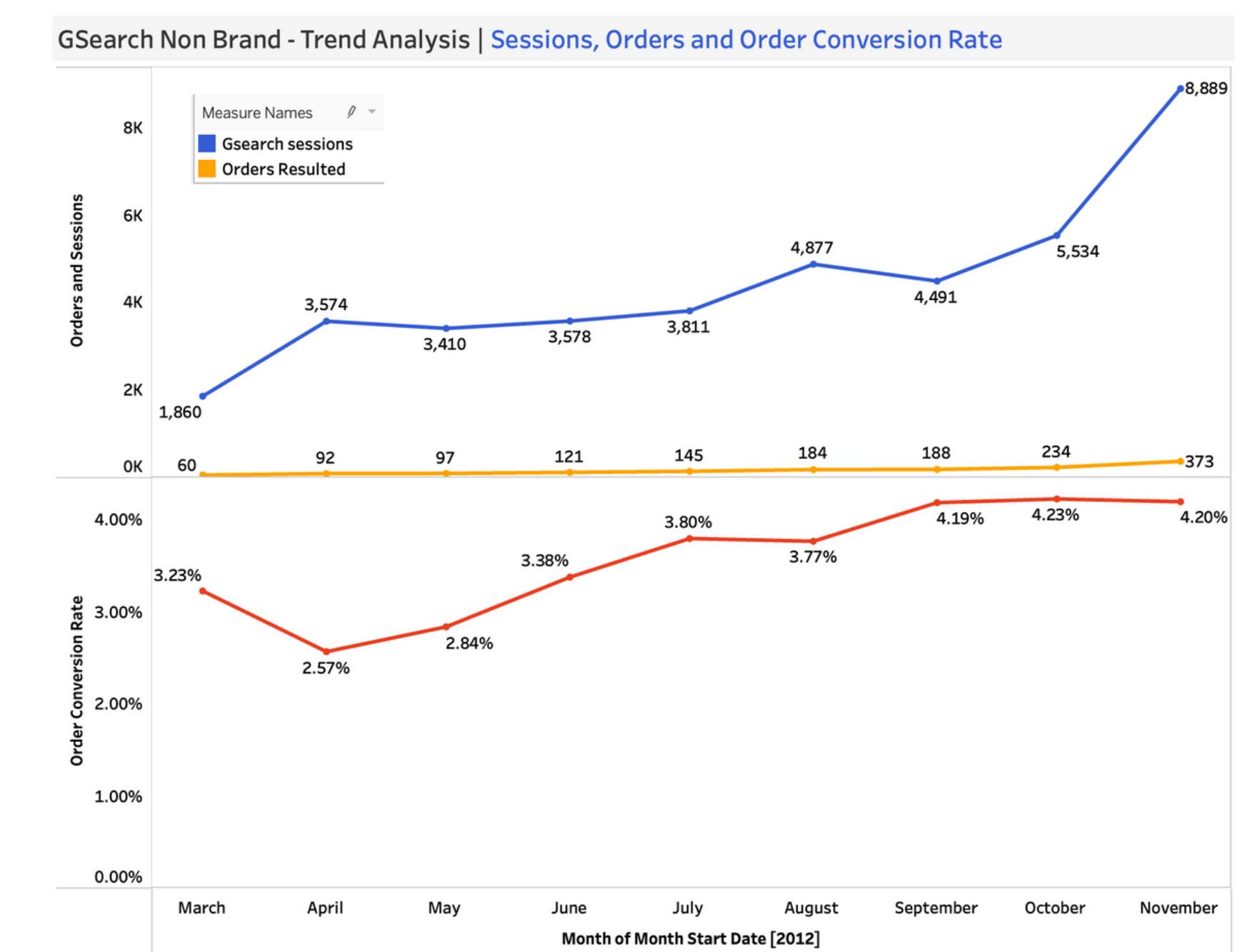
GSearch Non Brand - Main Driver of Growth. Lets look at the monthly trends

```
10
11 •  SELECT DISTINCT utm_source, utm_campaign FROM website_sessions;
12
13 •  SELECT|
14     MONTH(W.created_at) AS month,
15     MIN(DATE(W.created_at)) AS month_start_date,
16     COUNT(DISTINCT W.website_session_id) AS Gsearch_sessions,
17     COUNT(DISTINCT O.order_id) AS orders_resulted,
18     COUNT(DISTINCT O.order_id)/COUNT(DISTINCT W.website_session_id) AS conversion_rate
19   FROM website_sessions AS W
20     LEFT JOIN orders AS O ON W.website_session_id = O.website_session_id
21   WHERE w.created_at <'2012-11-27'
22   AND w.utm_source = 'gsearch'
23   GROUP BY 1;
24
25
```

100% 7:13

Result Grid Filter Rows: Search Export:

month	month_start_d...	Gsearch_sessions	orders_resulted	conversion_rate
3	2012-03-19	1860	60	0.0323
4	2012-04-01	3574	92	0.0257
5	2012-05-01	3410	97	0.0284
6	2012-06-01	3578	121	0.0338
7	2012-07-01	3811	145	0.0380
8	2012-08-01	4877	184	0.0377
9	2012-09-01	4491	188	0.0419
10	2012-10-01	5534	234	0.0423
11	2012-11-01	8889	373	0.0420

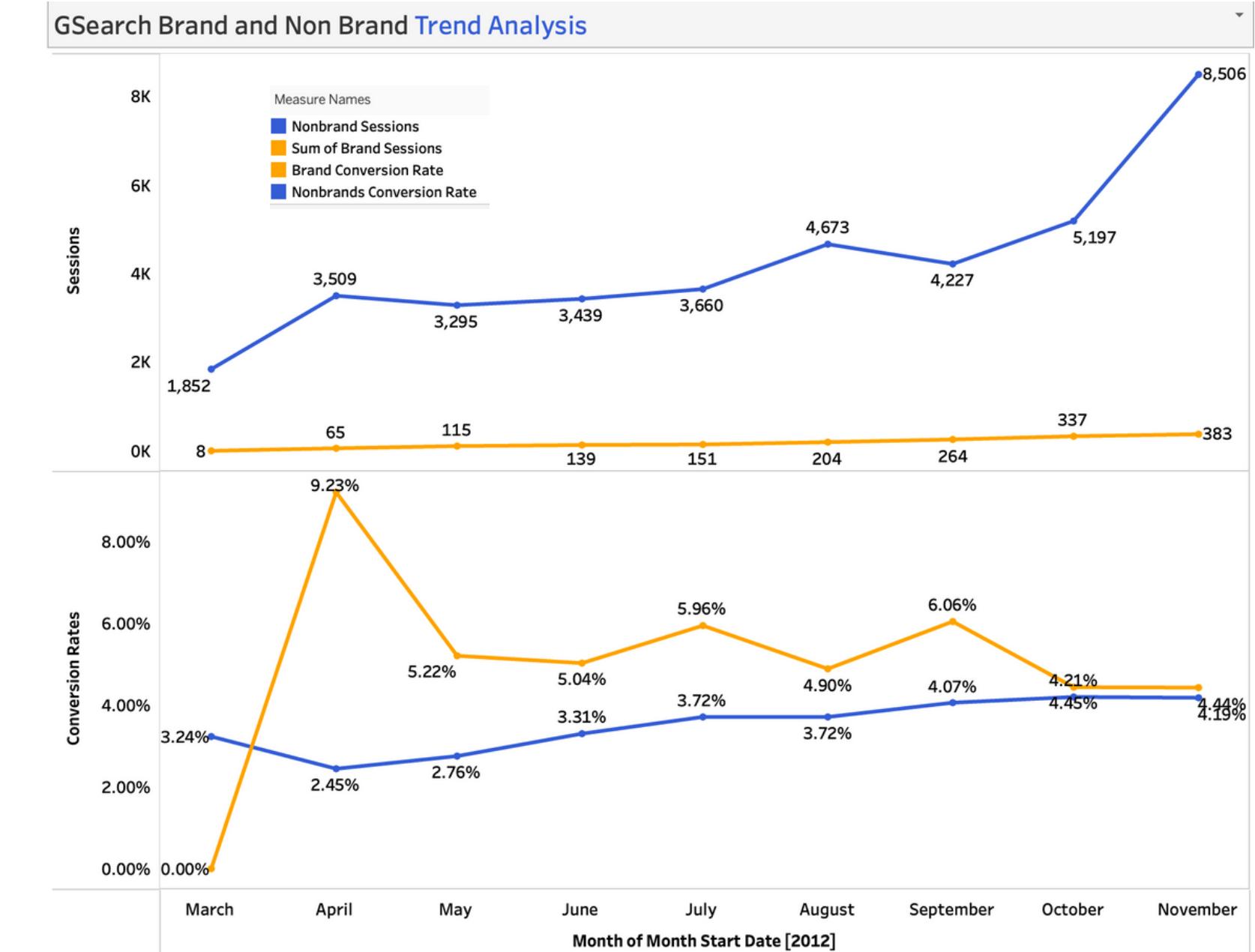


Findings:

- Sessions and Orders both seem to be growing substantially. If we look at the order volume from March or the first month of April at 92, we're already at about four times that order volume.
- Conversion rate starts in this range here of roughly 3 % over the first three months, and then by the final months we're looking at here, we're up over 4%.

GSearch Marketing Channels Trend Analysis

```
27 -- Processing on how different search marketing campaigns working
28 -- There are two gsearch campaigns running, brand and nonbrand
29
30 SELECT
31     MONTH(W.created_at) AS month,
32     MIN(DATE(W.created_at)) AS month_start_date,
33     COUNT(DISTINCT CASE WHEN W.utm_campaign = 'nonbrand' THEN W.website_session_id ELSE NULL END) AS nonbrand_sessions,
34     COUNT(DISTINCT CASE WHEN W.utm_campaign = 'nonbrand' THEN O.order_id ELSE NULL END) AS nonbrand_orders,
35     COUNT(DISTINCT CASE WHEN W.utm_campaign = 'nonbrand' THEN O.order_id ELSE NULL END)/COUNT(DISTINCT CASE WHEN W.utm_campaign = 'nonbrand' THEN W.website_session_id ELSE NULL END) AS nonbrands_conversion_rate,
36     COUNT(DISTINCT CASE WHEN W.utm_campaign = 'brand' THEN W.website_session_id ELSE NULL END) AS brand_sessions,
37     COUNT(DISTINCT CASE WHEN W.utm_campaign = 'brand' THEN O.order_id ELSE NULL END) AS brand_orders,
38     COUNT(DISTINCT CASE WHEN W.utm_campaign = 'brand' THEN O.order_id ELSE NULL END)/COUNT(DISTINCT CASE WHEN W.utm_campaign = 'brand' THEN W.website_session_id ELSE NULL END) AS brand_conversion_rate
39
40     FROM website_sessions AS W
41             LEFT JOIN orders AS O ON W.website_session_id = O.website_session_id
42     WHERE w.created_at < '2012-11-27'
43     AND w.utm_source = 'gsearch'
44     GROUP BY 1;
```



Findings:

- Brand campaigns represent someone going into search engines and explicitly looking for your business so the fact that this has increased dramatically is a good sign.

GSearch Non Brand Split by Device Type | SQL

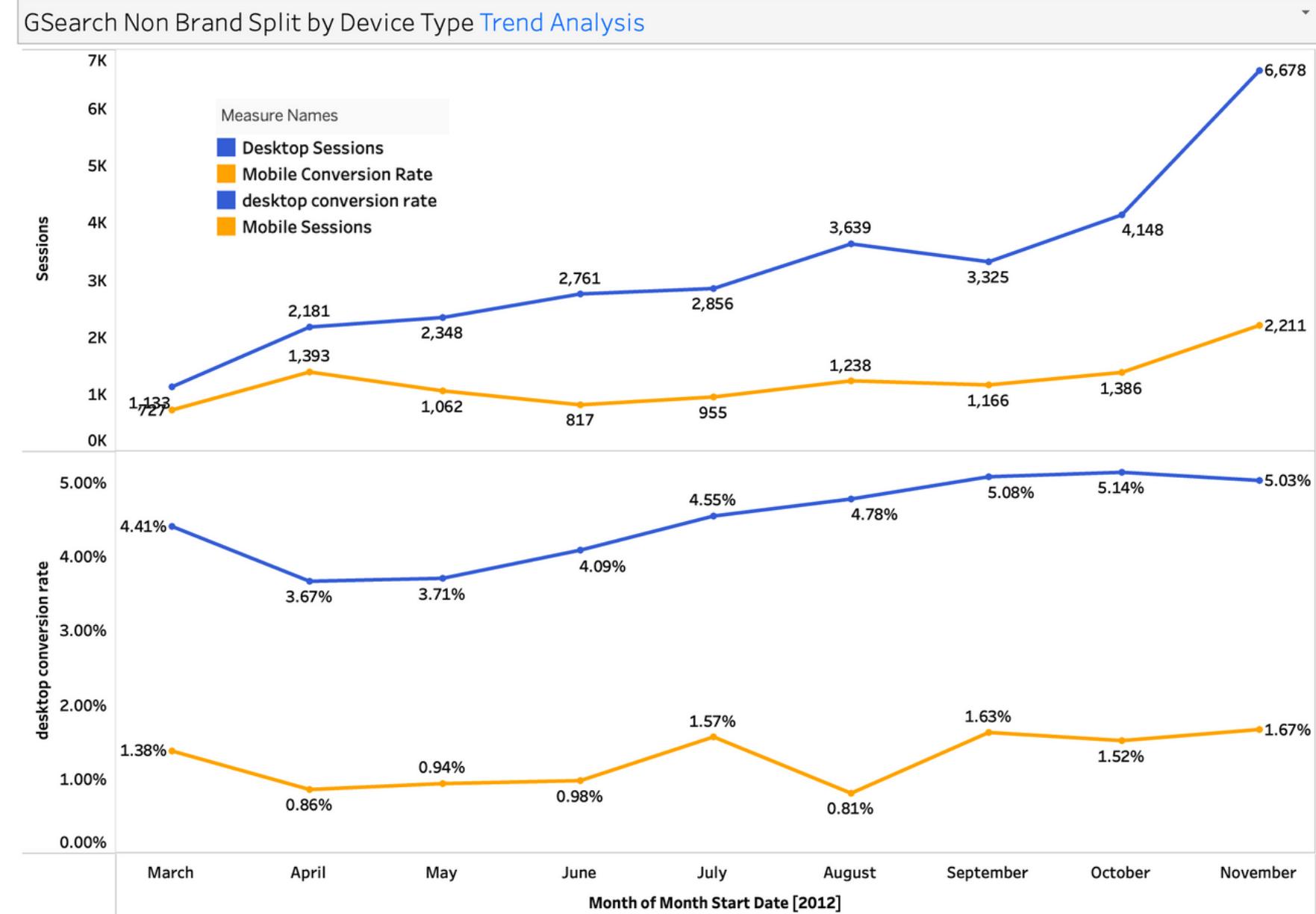
```

49   SELECT
50     MONTH(W.created_at) AS month,
51     MIN(DATE(W.created_at)) AS month_start_date,
52     COUNT(DISTINCT CASE WHEN W.device_type = 'mobile' THEN W.website_session_id ELSE NULL END) AS mobile_sessions,
53     COUNT(DISTINCT CASE WHEN W.device_type = 'mobile' THEN O.order_id ELSE NULL END) AS mobile_orders,
54     COUNT(DISTINCT CASE WHEN W.device_type = 'mobile' THEN O.order_id ELSE NULL END)/COUNT(DISTINCT CASE WHEN W.device_type = 'mobile' THEN W.website_session_id ELSE NULL END) AS mobile_conversion_rate,
55     COUNT(DISTINCT CASE WHEN W.device_type = 'desktop' THEN W.website_session_id ELSE NULL END) AS desktop_sessions,
56     COUNT(DISTINCT CASE WHEN W.device_type = 'desktop' THEN O.order_id ELSE NULL END) AS desktop_orders,
57     COUNT(DISTINCT CASE WHEN W.device_type = 'desktop' THEN O.order_id ELSE NULL END)/COUNT(DISTINCT CASE WHEN W.device_type = 'desktop' THEN W.website_session_id ELSE NULL END) AS desktop_conversion_rate
58   FROM website_sessions AS W
59   LEFT JOIN orders AS O ON W.website_session_id = O.website_session_id
60   WHERE w.created_at < '2012-11-27'
61   AND w.utm_source = 'gsearch'
62   GROUP BY 1;
63
64 -- How are each channels driving the traffic and order rate

```

Result Grid Filter Rows: Search Export:

month	month_start_d...	mobile_sessions	mobile_orders	mobile_conversion_r...	desktop_sessions	desktop_order...	desktop_conversion_r...
3	2012-03-19	727	10	0.0138	1133	50	0.0441
4	2012-04-01	1393	12	0.0086	2181	80	0.0367
5	2012-05-01	1062	10	0.0094	2348	87	0.0371
6	2012-06-01	817	8	0.0098	2761	113	0.0409
7	2012-07-01	955	15	0.0157	2856	130	0.0455
8	2012-08-01	1238	10	0.0081	3639	174	0.0478
9	2012-09-01	1166	19	0.0163	3325	169	0.0508
10	2012-10-01	1386	21	0.0152	4148	213	0.0514
11	2012-11-01	2211	37	0.0167	6678	336	0.0503



Findings:

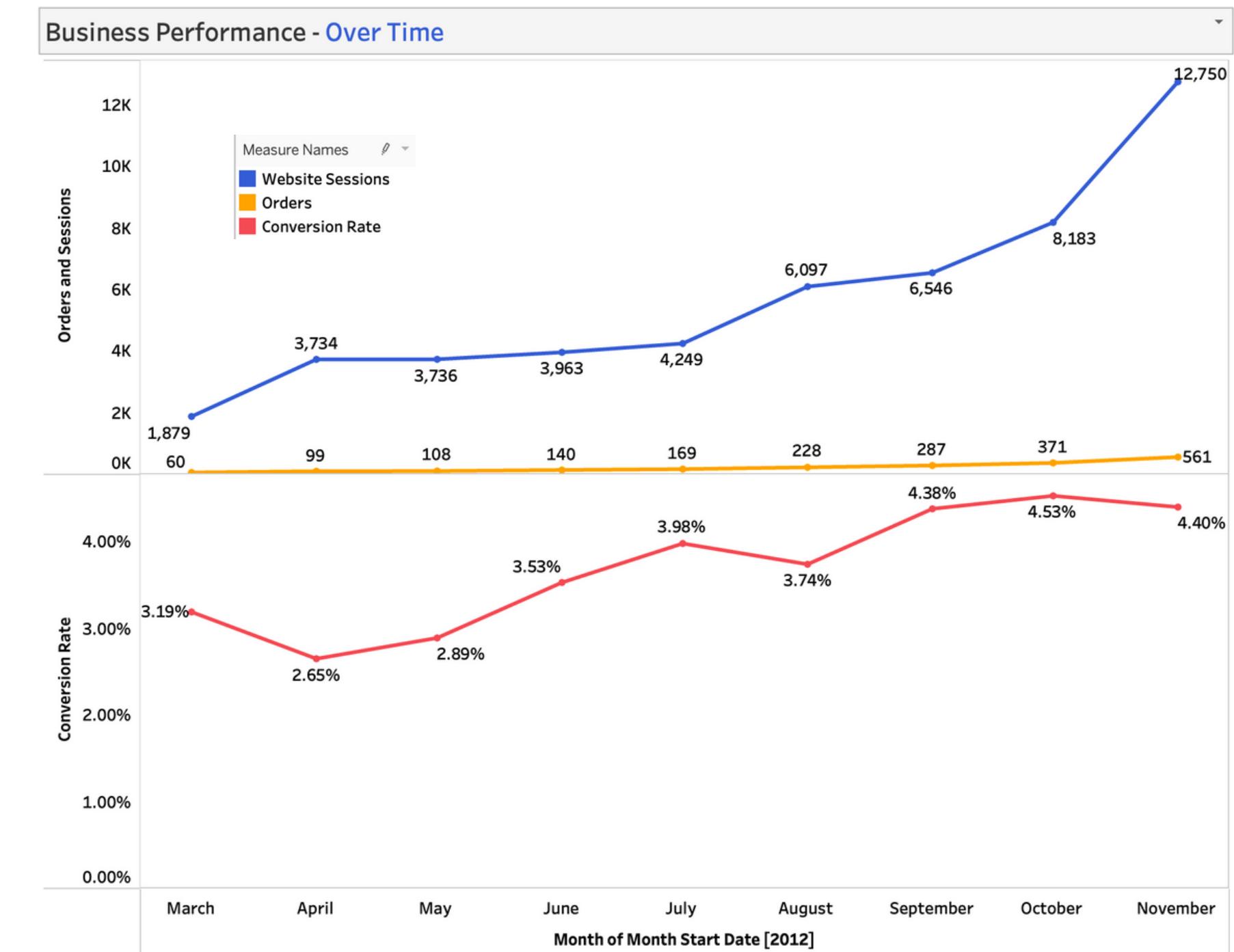
- Lot more desktop sessions so from the beginning it was a little less than a 2 to one ratio but here at the end of this time we've got more than a three to one ratio.

Website Performance Improvement | SQL

```
86
87 • SELECT
88     YEAR(W.created_at),
89     MONTH(W.created_at),
90     MIN(DATE(W.created_at)) AS month_start_date,
91     COUNT(DISTINCT W.website_session_id) AS website_sessions,
92     COUNT(DISTINCT O.order_id) AS orders,
93     COUNT(DISTINCT O.order_id)/COUNT(DISTINCT W.website_session_id) AS conversion_rate
94
95     FROM website_sessions AS W
96     LEFT JOIN orders AS O ON W.website_session_id = O.website_session_id
97     WHERE w.created_at <'2012-11-27'
98     GROUP BY 1,2;
```

Result Grid Filter Rows: Search Export:

YEAR(W.created_at)	MONTH(W.created_at)	month_start_d...	website_sessions	orders	conversion_rate
2012	3	2012-03-19	1879	60	0.0319
2012	4	2012-04-01	3734	99	0.0265
2012	5	2012-05-01	3736	108	0.0289
2012	6	2012-06-01	3963	140	0.0353
2012	7	2012-07-01	4249	169	0.0398
2012	8	2012-08-01	6097	228	0.0374
2012	9	2012-09-01	6546	287	0.0438
2012	10	2012-10-01	8183	371	0.0453
2012	11	2012-11-01	12750	561	0.0440



Findings:

- Lot more desktop sessions so from the beginning it was a little less than a 2 to one ratio but here at the end of this time we've got more than a three to one ratio.

GSearch Lander Test Analysis | SQL

```

101
102
103 -- Lander Test, how much is increment in the revenue?
104 -- Step 1 Identify the lander page view IDS
105 • SELECT
106     min(website_session_id),
107     min(website_pageview_id)
108 FROM
109     website_pageviews
110 WHERE pageview_url = '/lander-1';
111 # min(website_session_id), min(website_pageview_id)
112 # '11683', '23504'
113

```

Result Grid

	min(website_session_id)	min(website_pageview_id)
▶	11683	23504

```

7
8 -- Step 2 Identity the landing pageview_ids
9 • CREATE TEMPORARY TABLE landing_ids
10 SELECT
11     P.website_session_id,
12     MIN(P.website_pageview_id) AS website_pageview_id
13 FROM website_pageviews AS P
14     LEFT JOIN website_sessions AS S
15     ON P.website_session_id = S.website_session_id
16 WHERE
17     P.website_session_id >= 11633
18 AND P.created_at < '2012-07-28' -- A/B Testing was done in this time period
19 AND S.utm_source = 'gsearch'
20 AND S.utm_campaign = 'nonbrand'
21 GROUP BY 1;
22 • SELECT * FROM landing_ids;
23

```

Result Grid

website_session_id	website_pageview_id
11635	23376
11636	23384
11637	23385
11638	23386
11639	23387
11640	23390
11641	23391
11642	23392

```

133
134 -- Step 3
135 -- Joining it with URLs and the order_id table
136 • CREATE TEMPORARY TABLE landing_page
137 SELECT
138     L.website_session_id,
139     W.pageview_url AS landing_url
140 FROM landing_ids AS L
141     LEFT JOIN website_pageviews AS W
142     ON L.website_pageview_id = W.website_pageview_id;
143 • SELECT * FROM landing_page;
144

```

Result Grid

website_session_id	landing_url
▶ 11635	/home
11636	/home
11637	/home
11638	/home
11639	/home

```

144
145 -- Step 4
146 -- Now joining this with the orders table
147 -- to find the increment in order conversion rate
148 • CREATE TEMPORARY TABLE testing_orders
149 SELECT
150     L.website_session_id,
151     L.landing_url,
152     O.order_id
153 FROM landing_page AS L
154     LEFT JOIN orders AS O
155     ON L.website_session_id = O.website_session_id ;
156 • SELECT * FROM testing_orders;
157

```

Result Grid

website_session_id	landing_url	order_id
11672	/home	NULL
11673	/home	NULL
11674	/home	NULL
11675	/home	NULL
11677	/home	349
11678	/home	NULL
11679	/home	NULL
11680	/home	NULL

```

158 -- Step 5, Calculating Conversion Rate
159 • SELECT
160     landing_url,
161     COUNT(DISTINCT website_session_id) AS sessions,
162     COUNT(DISTINCT order_id) AS orders,
163     COUNT(DISTINCT order_id)/COUNT(DISTINCT website_session_id) AS conversion_rate
164 FROM testing_orders
165 GROUP BY 1;
166 -- increment in the conversion raate = 0.0405 - 0.0325 = 0.0084
167

```

Result Grid

landing_url	sessions	orders	conversion_rate
▶ /home	2300	74	0.0322
/lander-1	2316	94	0.0406

```

187 • SELECT
188     count(website_session_id) AS sessions_since_test
189 FROM website_sessions
190 WHERE utm_source = 'gsearch'
191 AND utm_campaign = 'nonbrand'
192 AND created_at < '2012-11-27'
193 AND website_session_id > 17145;
194
195 -- 22972 order since the test
196 -- lift in conversion rate 0.0084
197 -- 192 incremental orders since A/B test concluded
198

```

Result Grid

sessions_since_t...
▶ 22972

Findings:

- The A/B Lander Test increased the conversion rate by 0.0084%, which resulted in the lift of orders by 192. [4 Months Data]

GSearch Lander Test Conversion Funnel | SQL

```

210 -- home or lander1, products, fuzzy, cart, shipping, billing, thankyou
211 • SELECT
212     S.website_session_id,
213     P.pageview_url,
214     P.created_at,
215     CASE WHEN p.pageview_url = '/home' THEN 1 ELSE 0 END AS flag_home,
216     CASE WHEN p.pageview_url = '/lander-1' THEN 1 ELSE 0 END AS flag_lander1,
217     CASE WHEN p.pageview_url = '/products' THEN 1 ELSE 0 END AS flag_products,
218     CASE WHEN p.pageview_url = '/the-original-mr-fuzzy' THEN 1 ELSE 0 END AS flag_fuzzy,
219     CASE WHEN p.pageview_url = '/cart' THEN 1 ELSE 0 END AS flag_cart,
220     CASE WHEN p.pageview_url = '/shipping' THEN 1 ELSE 0 END AS flag_shipping,
221     CASE WHEN p.pageview_url = '/billing' THEN 1 ELSE 0 END AS flag_billing,
222     CASE WHEN p.pageview_url = '/thank-you-for-your-order' THEN 1 ELSE 0 END AS flag_thankyou
223 FROM website_sessions AS S
224     LEFT JOIN website_pageviews AS P ON S.website_session_id = P.website_session_id
225 WHERE S.created_at >= '2012-06-19'
226 AND S.created_at <= '2012-07-28'
227 AND S.utm_source = 'gsearch'
228 AND S.utm_campaign = 'nonbrand'
229 ORDER BY 1,3;
230

```

Result Grid Filter Rows: Export: Fetch rows:

website_session_id	pageview_url	created_at	flag_home	flag_lander1	flag_products	flag_fuzzy	flag_cart	flag_shipping	flag_billing	flag_thankyou
11683	/lander-1	2012-06-19 00:35:54	0	1	0	0	0	0	0	0
11684	/home	2012-06-19 00:48:25	1	0	0	0	0	0	0	0
11685	/lander-1	2012-06-19 01:02:38	0	1	0	0	0	0	0	0
11686	/lander-1	2012-06-19 01:31:57	0	1	0	0	0	0	0	0
11686	/products	2012-06-19 01:32:51	0	0	1	0	0	0	0	0
11687	/home	2012-06-19 01:36:39	1	0	0	0	0	0	0	0
11688	/home	2012-06-19 01:47:53	1	0	0	0	0	0	0	0
11689	/lander-1	2012-06-19 02:14:55	0	1	0	0	0	0	0	0
11689	/products	2012-06-19 02:18:38	0	0	1	0	0	0	0	0
11689	/the-original-mr-fuzzy	2012-06-19 02:20:28	0	0	0	1	0	0	0	0
11690	/home	2012-06-19 03:05:34	1	0	0	0	0	0	0	0
11691	/lander-1	2012-06-19 04:08:52	0	1	0	0	0	0	0	0

```

233
234 • CREATE TEMPORARY TABLE session_level
235     SELECT
236         website_session_id,
237         MAX(flag_home) AS saw_home,
238         MAX(flag_lander1) AS saw_lander1,
239         MAX(flag_products) AS saw_products,
240         MAX(flag_fuzzy) AS saw_fuzzy,
241         MAX(flag_cart) AS saw_cart,
242         MAX(flag_shipping) AS saw_shipping,
243         MAX(flag_billing) AS saw_billing,
244         MAX(flag_thankyou) AS saw_thankyou
245     FROM (
246         SELECT *
247         FROM pageview_level
248     GROUP BY 1;
249 • SELECT * FROM session_level;
250

```

Result Grid Filter Rows: Search: Export: Fetch rows:

website_session_id	saw_home	saw_lander1	saw_products	saw_fuzzy	saw_cart	saw_shipping	saw_billing	saw_thankyou
11683	0	1	0	0	0	0	0	0
11684	1	0	0	0	0	0	0	0
11685	0	1	0	0	0	0	0	0
11686	0	1	1	0	0	0	0	0
11687	1	0	0	0	0	0	0	0
11688	1	0	0	0	0	0	0	0
11689	0	1	1	0	0	0	0	0
11690	1	0	0	0	0	0	0	0
11691	0	1	1	0	0	0	0	0
11692	0	1	0	0	0	0	0	0
11693	0	1	1	0	0	0	0	0

```

268
269 • SELECT
270     CASE
271         WHEN saw_home = 1 THEN 'saw_homepage'
272         WHEN saw_lander1 = 1 THEN 'saw_lander1'
273         ELSE 'check_logic' END AS lander_type,
274     COUNT(website_session_id) AS sessions,
275     SUM(saw_products) AS to_products,
276     SUM(saw_fuzzy) AS to_fuzzy,
277     SUM(saw_cart) AS to_cart,
278     SUM(saw_shipping) AS to_shipping,
279     SUM(saw_billing) AS to_billing,
280     SUM(saw_thankyou) AS to_thankyou
281     FROM session_level
282     GROUP BY 1;
283

```

Result Grid Filter Rows: Export:

lander_type	sessions	to_products	to_fuzzy	to_cart	to_shipping	to_billing	to_thankyou
saw_lander1	2316	1083	772	348	231	197	94
saw_homepage	2261	942	684	296	200	168	72

```

284 • SELECT
285     CASE
286         WHEN saw_home = 1 THEN 'saw_homepage'
287         WHEN saw_lander1 = 1 THEN 'saw_lander1'
288         ELSE 'check_logic' END AS lander_type,
289     SUM(saw_products)/COUNT(website_session_id) AS clickrate_lander,
290     SUM(saw_fuzzy)/SUM(saw_products) AS clickrate_products,
291     SUM(saw_cart)/SUM(saw_fuzzy) AS clickrate_fuzzy,
292     SUM(saw_shipping)/SUM(saw_cart) AS clickrate_cart,
293     SUM(saw_billing)/SUM(saw_shipping) AS clickrate_shipping,
294     SUM(saw_thankyou)/SUM(saw_billing) AS clickrate_billing
295     FROM session_level
296     GROUP BY 1;
297
298 — Quantifying the billing test in terms revenue per billing session
299
300

```

Result Grid Filter Rows: Export:

lander_type	clickrate_lander	clickrate_products	clickrate_fuzzy	clickrate_cart	clickrate_shipping	clickrate_billing
saw_lander1	0.4676	0.7128	0.4508	0.6638	0.8528	0.4772
saw_homepage	0.4166	0.7261	0.4327	0.6757	0.8400	0.4286

Findings:

- Lander1 funnel generated ~5% higher orders compared to the home webpage.

Quantifying the impact of the Billing Test | SQL

```
302
303 -- Lets first identify the billing sessions, whether they resulted in a order or not
304 • SELECT
305     P.website_session_id,
306     P.pageview_url,
307     O.order_id,
308     O.price_usd
309 FROM website_pageviews AS P
310     LEFT JOIN orders AS O ON P.website_session_id = O.website_session_id
311 WHERE P.created_at > '2012-09-10'
312 AND P.created_at < '2012-11-10'
313 AND P.pageview_url IN ('/billing', '/billing-2');
314
%   86:303 |
```

Result Grid

website_session_id	pageview_url	order_id	price_usd
25353	/billing	873	49.99
25358	/billing-2	874	49.99
25368	/billing	875	49.99
25393	/billing	876	49.99
25411	/billing-2	877	49.99
25454	/billing-2	878	49.99
25459	/billing-2	NULL	NULL
25468	/billing	NULL	NULL
25488	/billing	NULL	NULL
25508	/billing	879	49.99
25516	/billing-2	880	49.99

```
316
317 -- finding the lift
318 • SELECT
319     pageview_url,
320     COUNT(website_session_id) AS billing_sessions,
321     SUM(price_usd)/COUNT(website_session_id) AS revenue_per_billingsessions
322 FROM (
323     SELECT
324         P.website_session_id,
325         P.pageview_url,
326         O.order_id,
327         O.price_usd
328     FROM website_pageviews AS P
329     LEFT JOIN orders AS O ON P.website_session_id = O.website_session_id
330 WHERE P.created_at > '2012-09-10'
331 AND P.created_at < '2012-11-10'
332 AND P.pageview_url IN ('/billing', '/billing-2') AS billing
333 GROUP BY 1;
334 -- There is a lift of 8.51 dollars per billing session seen
335
%   7:330 |
```

Result Grid

pageview_url	billing_sessions	revenue_per_billingsessions
/billing-2	654	31.339297
/billing	657	22.826484

```
335
336 -- Monthly Impact
337 • SELECT
338     count(website_session_id) AS billing_sessions
339 FROM website_pageviews
340 WHERE
341     created_at >= '2012-10-27'
342 AND created_at <= '2012-11-27'
343 AND pageview_url IN ('/billing', '/billing-2')
344
345 -- 1193*8.51 = 10152 additional billing generated
346
347
%   31:342 |
```

Result Grid

billing_sessions
1193