Lab Exercise 5 – Filtering and Formatting Data

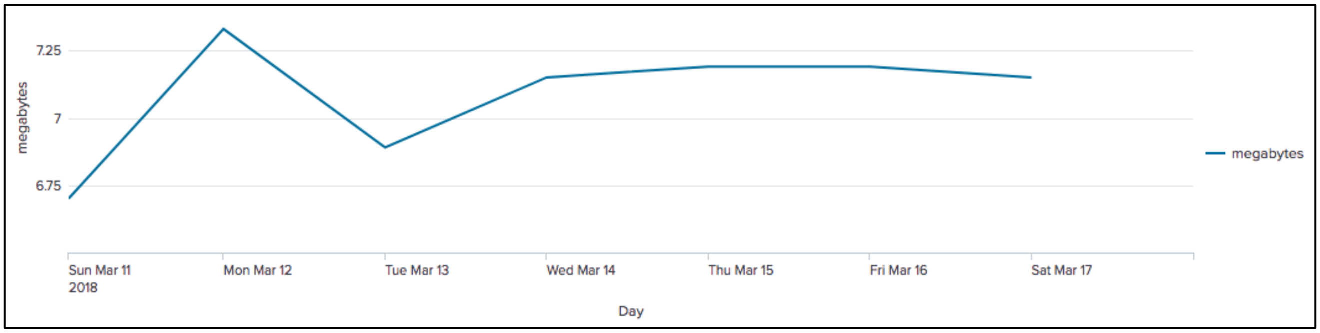
Description

In this lab exercise, you use `eval`, `search`, and `where` commands.

Steps

Task 1: Chart the total daily volume (in MB) of the web servers during the previous week.

Final Results Example:



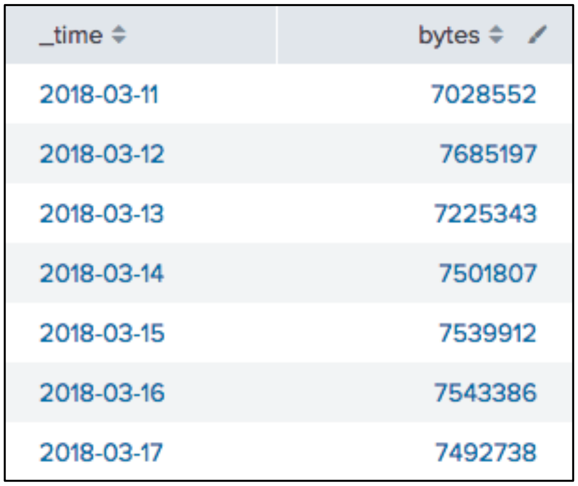
1. Search online sales [`access_combined`] during the **previous week**.

   index=web sourcetype=access_combined

2. Use `timechart` to calculate the total `bytes` and name the field: `bytes`

   index=web sourcetype=access_combined
   | timechart sum(bytes) as bytes

Results Example:

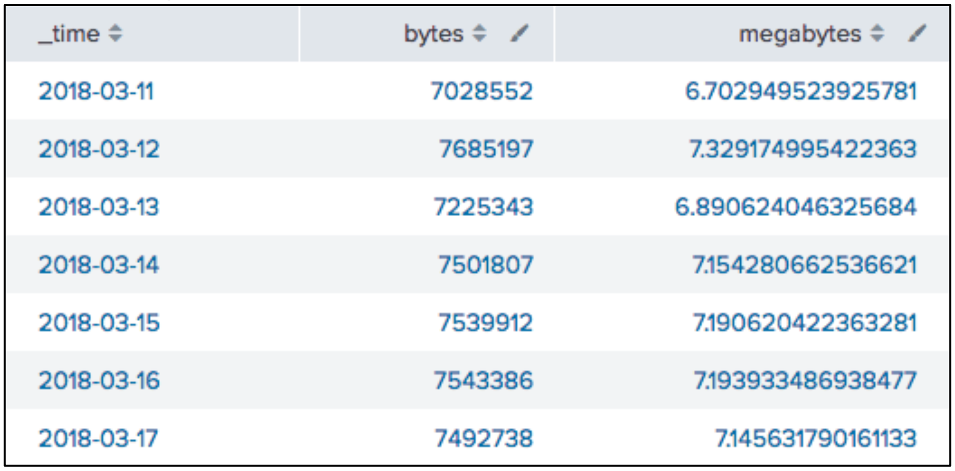


_time	bytes
2018-03-11	7028552
2018-03-12	7685197
2018-03-13	7225343
2018-03-14	7501807
2018-03-15	7539912
2018-03-16	7543386
2018-03-17	7492738

3. Use `eval` to convert the `bytes` field to `megabytes`.

   sourcetype=access_combined
   | timechart sum(bytes) as bytes
   | eval megabytes=bytes/(1024*1024)

Results Example:

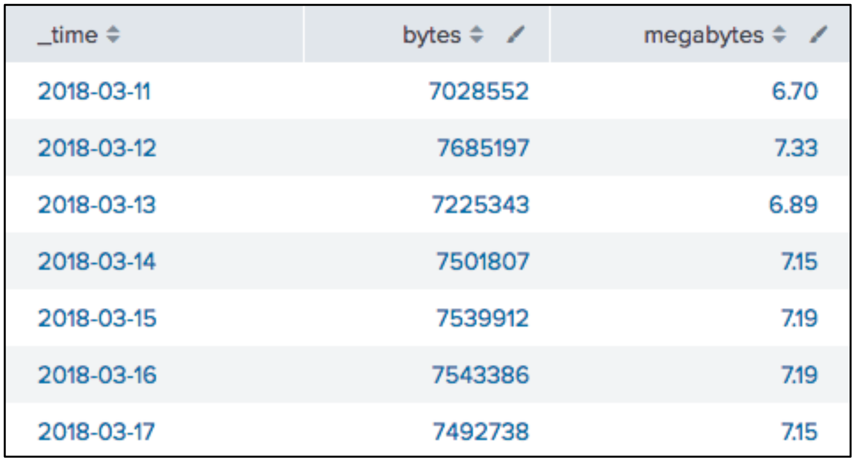


_time	bytes	megabytes
2018-03-11	7028552	6.702949523925781
2018-03-12	7685197	7.329174995422363
2018-03-13	7225343	6.890624046325684
2018-03-14	7501807	7.154280662536621
2018-03-15	7539912	7.190620422363281
2018-03-16	7543386	7.193933486938477
2018-03-17	7492738	7.145631790161133

4. Use the `round` function to round the `megabytes` field values to two decimal places.

```
index=web sourcetype=access_combined
| timechart sum(bytes) as bytes
| eval megabytes=round(bytes/(1024*1024),2)
```

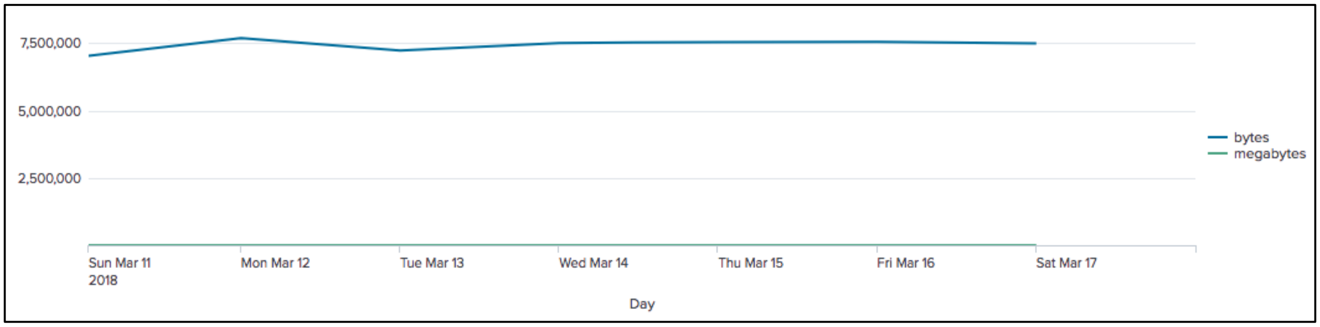
Results Example:



_time	bytes	megabytes
2018-03-11	7028552	6.70
2018-03-12	7685197	7.33
2018-03-13	7225343	6.89
2018-03-14	7501807	7.15
2018-03-15	7539912	7.19
2018-03-16	7543386	7.19
2018-03-17	7492738	7.15

5. Switch to the **Visualization** tab and display the data as a **Line Chart**. Set the X-axis label to **Day**. Notice that the `bytes` field still displays.

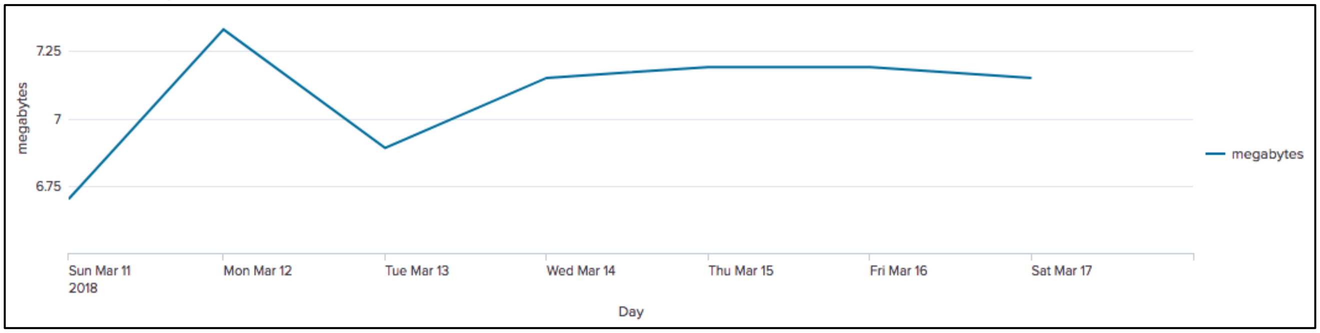
Results Example:



6. Use the `fields` command to remove the `bytes` field.

```
index=web sourcetype=access_combined
| timechart sum(bytes) as bytes
| eval megabytes=round(bytes/(1024*1024),2)
| fields - bytes
```

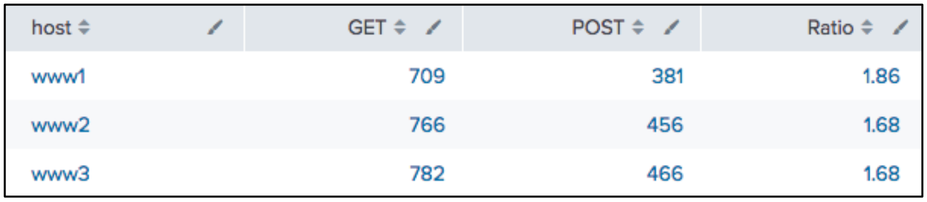
Results Example:



7. Save your search as report, **L4S1**.

Task 2: Calculate the ratio of GET requests to POST requests for each web server.

Final Results Example:



host	GET	POST	Ratio
www1	709	381	1.86
www2	766	456	1.68
www3	782	466	1.68

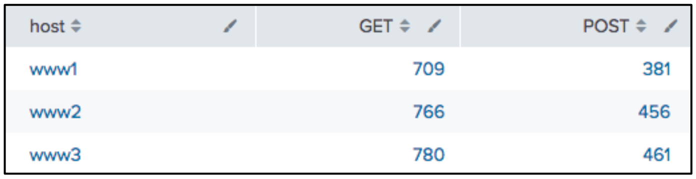
8. Search for all events in the online store [`access_combined`] during the **last 24 hours**.

```
index=web sourcetype=access_combined
```

9. Use `chart` to count events over `host` by `method`.

```
index=web sourcetype=access_combined
| chart count over host by method
```

Results Example:

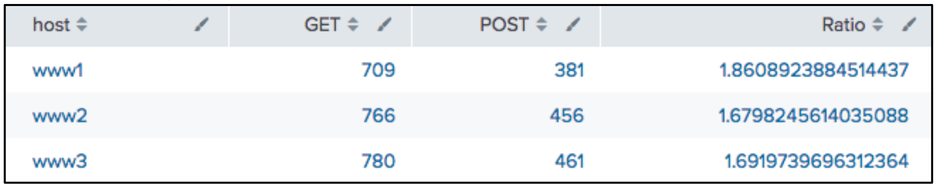


host	GET	POST
www1	709	381
www2	766	456
www3	780	461

10. Use `eval` to create a new column called `Ratio`, which divides `GET` by `POST`.

```
index=web sourcetype=access_combined
| chart count over host by method
| eval Ratio=GET/POST
```

Results Example:

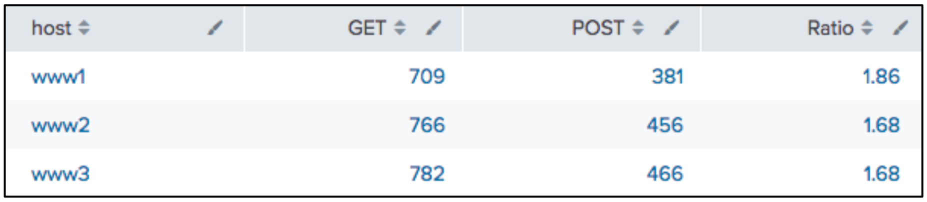


host	GET	POST	Ratio
www1	709	381	1.8608923884514437
www2	766	456	1.6798245614035088
www3	780	461	1.6919739696312364

11. Round the `Ratio` field to two decimal places.

```
index=web sourcetype=access_combined
| chart count over host by method
| eval Ratio=round(GET/POST,2)
```

Results Example:

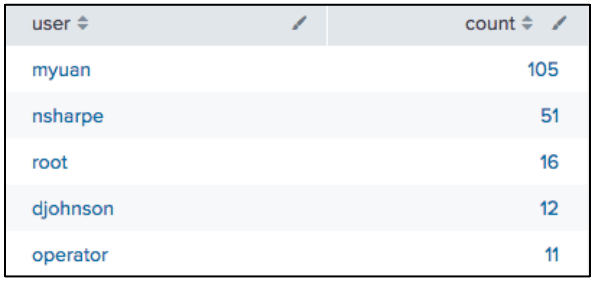


host	GET	POST	Ratio
www1	709	381	1.86
www2	766	456	1.68
www3	782	466	1.68

12. Save your search as report, **L4S2**.

Task 3: Identify users with more than 3 failed logins during the last 60 minutes and sort in descending order.

Final Results Example:

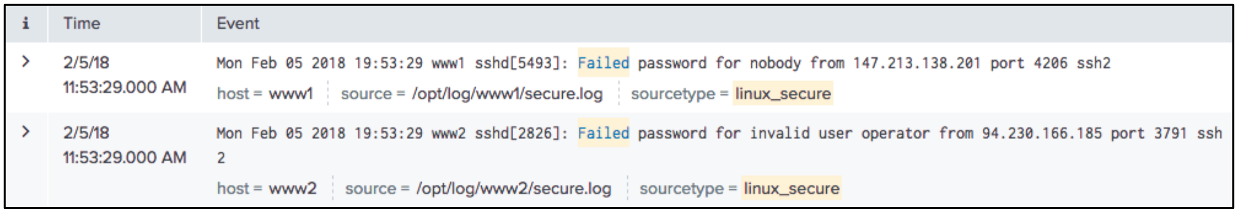


user	count
myuan	105
nsharpe	51
root	16
djohnson	12
operator	11

13. Search the web server [linux_secure] for failures during the **last 60 minutes**.

```
index=security sourcetype=linux_secure fail*
```

Results Example:

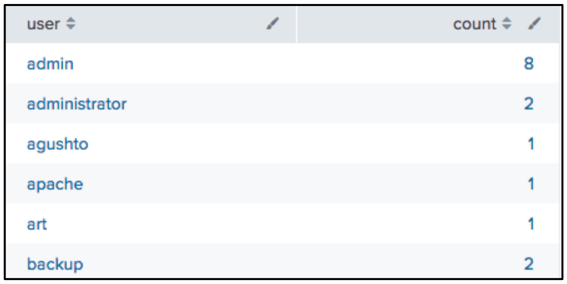


i	Time	Event
>	2/5/18 11:53:29.000 AM	Mon Feb 05 2018 19:53:29 www1 sshd[5493]: Failed password for nobody from 147.213.138.201 port 4206 ssh2 host = www1 source = /opt/log/www1/secure.log sourcetype = linux_secure
>	2/5/18 11:53:29.000 AM	Mon Feb 05 2018 19:53:29 www2 sshd[2826]: Failed password for invalid user operator from 94.230.166.185 port 3791 ssh2 host = www2 source = /opt/log/www2/secure.log sourcetype = linux_secure

14. Use `stats` to count the number of failures by user.

```
index=security sourcetype=linux_secure fail*
| stats count by user
```

Results Example:

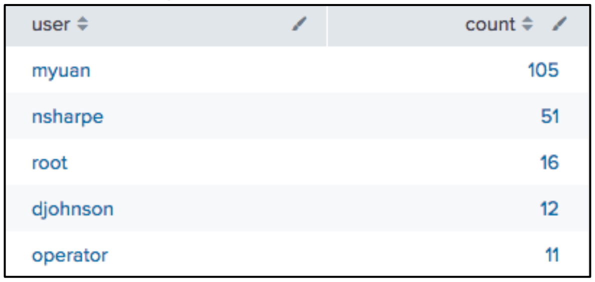


user	count
admin	8
administrator	2
agushto	1
apache	1
art	1
backup	2

15. Using the `search` command, filter the results to include only users with more than three failures and sort in descending order.

```
index=security sourcetype=linux_secure fail*
| stats count by user
| search count>3
| sort -count
```

Results Example:

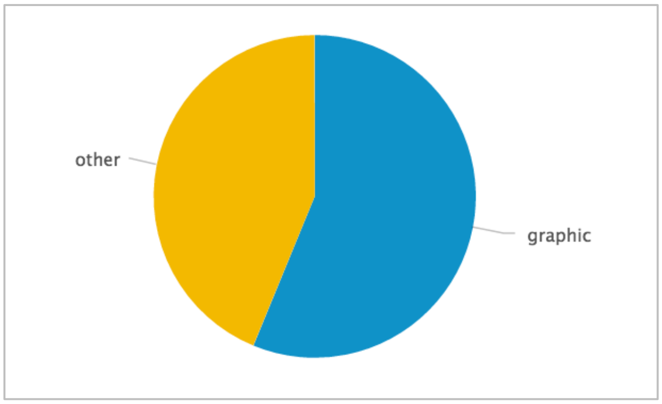


user	count
myuan	105
nsharpe	51
root	16
djohnson	12
operator	11

16. Save your search as report, **L4S3**.

Task 4: Classify and report employee web traffic by content type during the previous business week..

Final Results Example:



17. Search web appliance data [`cisco_wsa_squid`] during the **previous business week**.

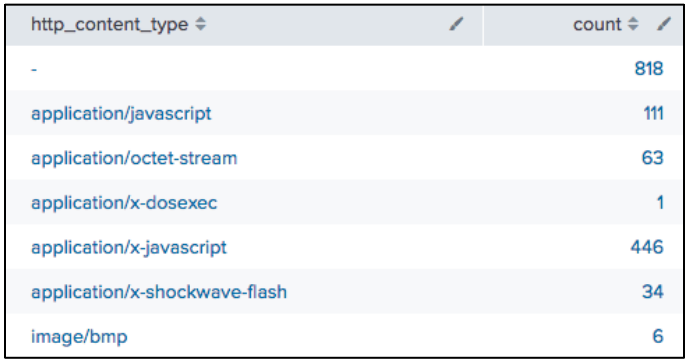
```
index=network sourcetype=cisco_wsa_squid
```

18. Use stats or chart to count events by the http_content_type field.

```
index=network sourcetype=cisco_wsa_squid
| stats count by http_content_type
```

NOTE: In this case, stats and chart are interchangeable—they use the same syntax and return the same results.

Results Example:



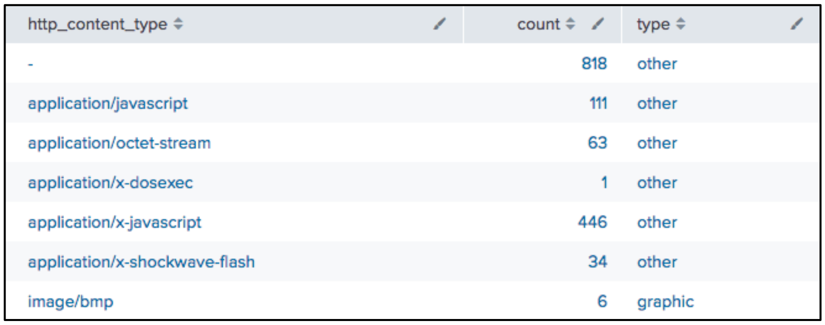
http_content_type	count
-	818
application/javascript	111
application/octet-stream	63
application/x-dosexec	1
application/x-javascript	446
application/x-shockwave-flash	34
image/bmp	6

19. Use the if function of eval to create a new column named type. If the http_content_type value begins with "image", set the type field to "graphic". Otherwise, set the value to "other".

Hint: Use the LIKE operator and the % wildcard to define the expression as follows:
`http_content_type LIKE "image%"`

```
index=network sourcetype=cisco_wsa_squid
| stats count by http_content_type
| eval type=if(http_content_type LIKE "image%","graphic","other")
```

Results Example:

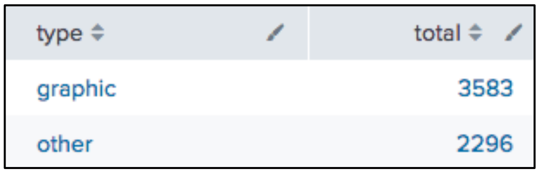


http_content_type	count	type
-	818	other
application/javascript	111	other
application/octet-stream	63	other
application/x-dosexec	1	other
application/x-javascript	446	other
application/x-shockwave-flash	34	other
image/bmp	6	graphic

20. Use another stats or chart command to sum the count column by the type field. Rename the sum of the count calculation to total.

```
index=network sourcetype=cisco_wsa_squid
| stats count by http_content_type
| eval type=if(http_content_type LIKE "image%","graphic","other")
| stats sum(count) as total by type
```

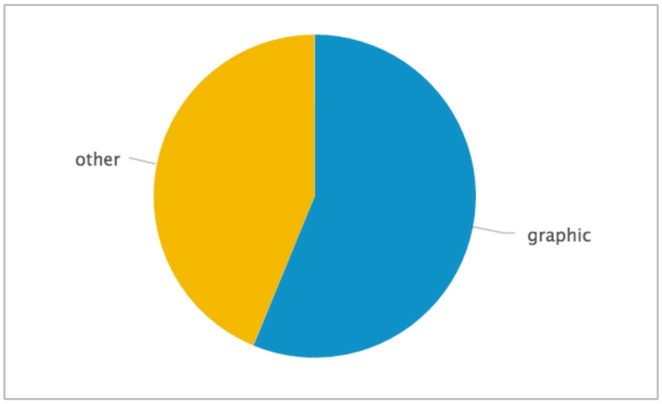
Results Example:



type	total
graphic	3583
other	2296

21. Change the visualization to a **Pie Chart**.

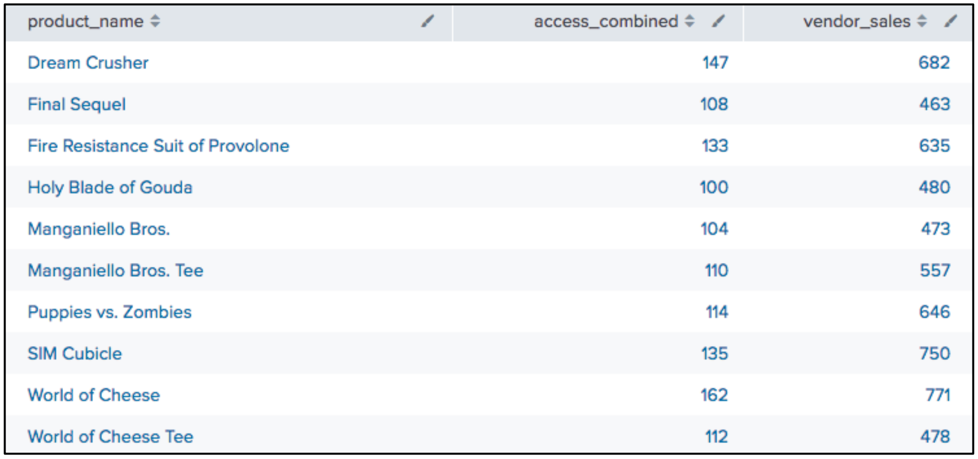
Results Example:



22. Save your search as report, **L4C1**.

Task 5: Report which products sold twice as much in the Buttercup Games online store than in the retail store during the previous week. Show the name of each of these products, as well as the number of units sold online and in the retail store.

Final Results Example:



product_name	access_combined	vendor_sales
Dream Crusher	147	682
Final Sequel	108	463
Fire Resistance Suit of Provolone	133	635
Holy Blade of Gouda	100	480
Manganiello Bros.	104	473
Manganiello Bros. Tee	110	557
Puppies vs. Zombies	114	646
SIM Cubicle	135	750
World of Cheese	162	771
World of Cheese Tee	112	478

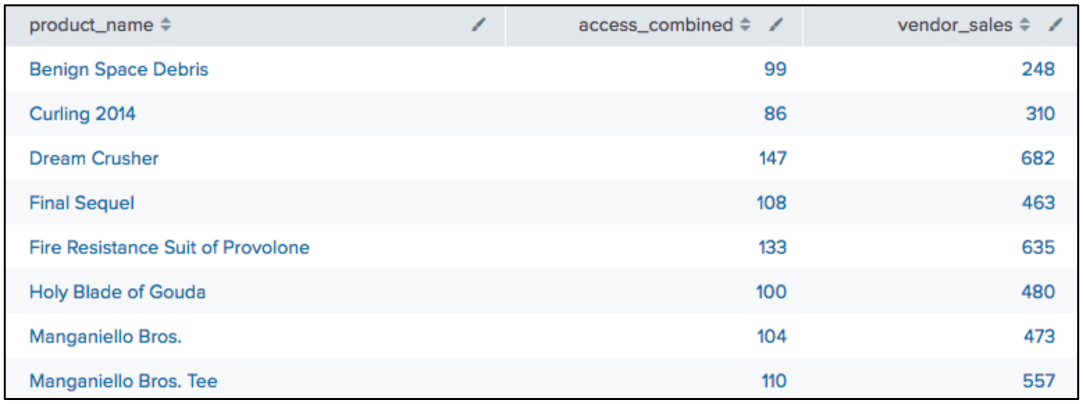
23. Search online sales data [`access_combined`] and retail sales data [`vendor_sales`] for successful purchases during the **previous week**.

    (index=web sourcetype=access* action=purchase status=200) OR (index=sales sourcetype=vendor_sales)

24. Chart a count of productId over product_name by sourcetype.

```
(index=web sourcetype=access* action=purchase status=200) OR (index=sales
sourcetype=vendor_sales)
| chart count(productId) as Count over product_name by sourcetype
```

Results Example:

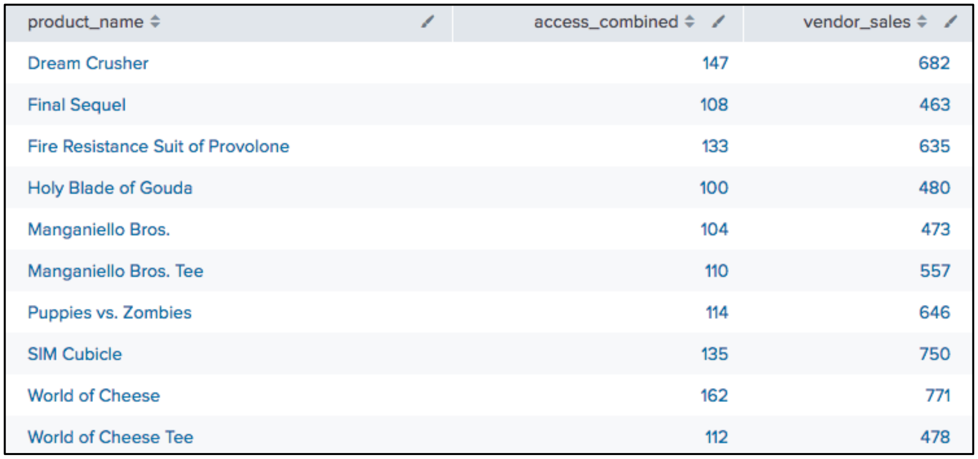


product_name	access_combined	vendor_sales
Benign Space Debris	99	248
Curling 2014	86	310
Dream Crusher	147	682
Final Sequel	108	463
Fire Resistance Suit of Provolone	133	635
Holy Blade of Gouda	100	480
Manganiello Bros.	104	473
Manganiello Bros. Tee	110	557

25. Use a `where` command to keep only rows where the value in `access_combined` is greater than two times the value in `vendor_sales`.

```
(index=web sourcetype=access* action=purchase status=200) OR (index=sales
sourcetype=vendor_sales)
| chart count(productId) as Count over product_name by sourcetype
| where access_combined > vendor_sales*2
```

Results Example:



product_name	access_combined	vendor_sales
Dream Crusher	147	682
Final Sequel	108	463
Fire Resistance Suit of Provolone	133	635
Holy Blade of Gouda	100	480
Manganiello Bros.	104	473
Manganiello Bros. Tee	110	557
Puppies vs. Zombies	114	646
SIM Cubicle	135	750
World of Cheese	162	771
World of Cheese Tee	112	478

26. Save your search as report, **L4C2**.

27. Modify your previous search to use `search` instead of `where`. Observe that the search produces no results. Why does this search produce no results?

```
(index=web sourcetype=access* action=purchase status=200) OR (index=sales
sourcetype=vendor_sales)
| chart count(productId) as Count over product_name by sourcetype
| search access_combined > vendor_sales*2
```

No results are found because the `search` command cannot compare values from two different fields. (As you saw earlier, the `where` command can do this.)