Lab Exercise 4 – Filtering Results and Manipulating Data

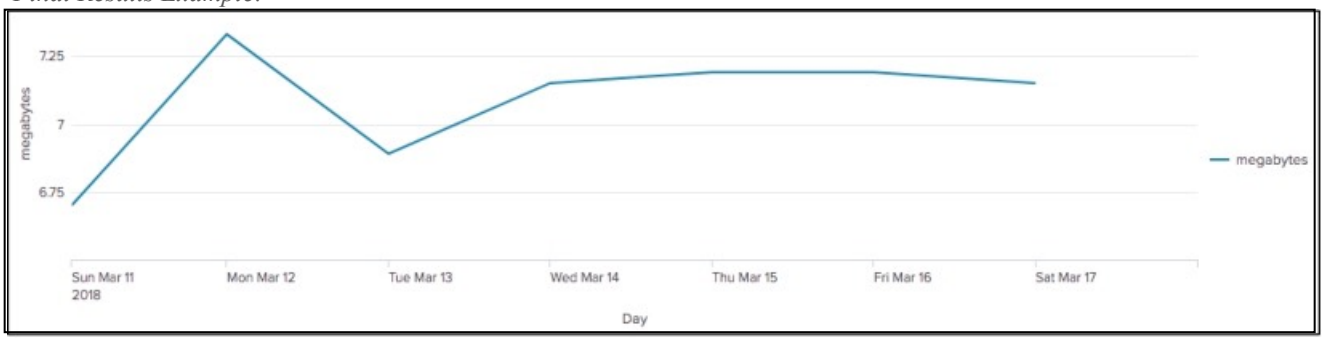
Description

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

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

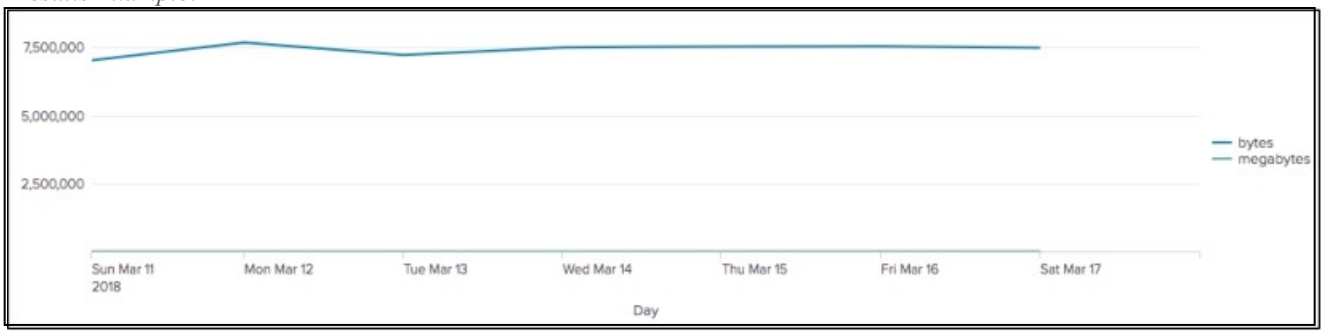
4. Use the `round` function to round the `megabytes` field values to two decimal places.

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

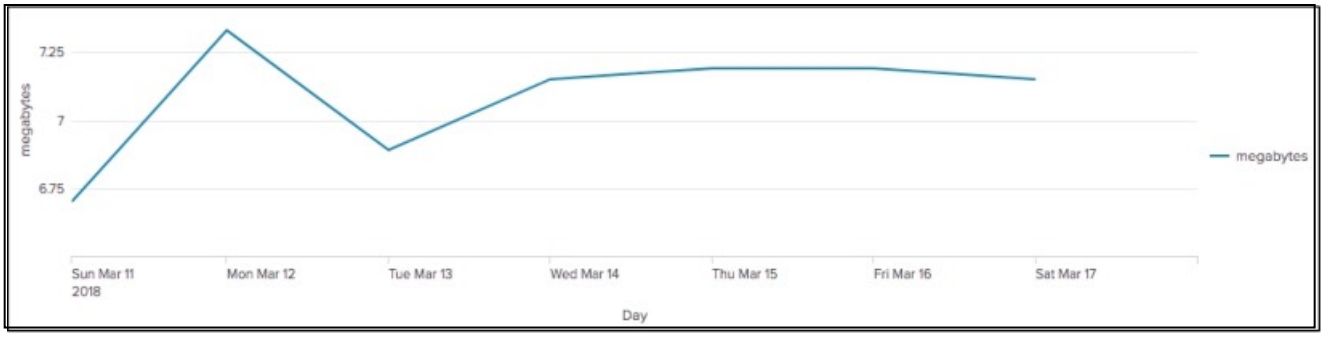
5. Switch to the **Visualization** tab and display the data as a **Line Chart**. Set the X-axis label to **Day**. Notice that the `bytes` field still displays.

Results Example:



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

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**.
9. Use `chart` to count events 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`.

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.

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 failed password attempts during the **last 60 minutes**.

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 ssh 2 host = www2 source = /opt/log/www2/secure.log sourcetype = linux_secure

14. Use `stats` to count the number of failed password attempts 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 .

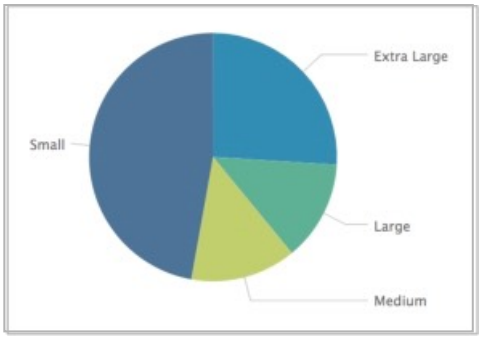
Results Example:

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

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

Scenario: **Evaluate and classify the number of bytes associated with each web server event during the last 24 hours as a pie chart. (Event sizes should be categorized as follows: Small, < 2000 bytes; Medium, from 2000 to 2500 bytes; Large, from 2500 to 3000 bytes; Extra Large, over 3000 bytes.)**

Example of final output:

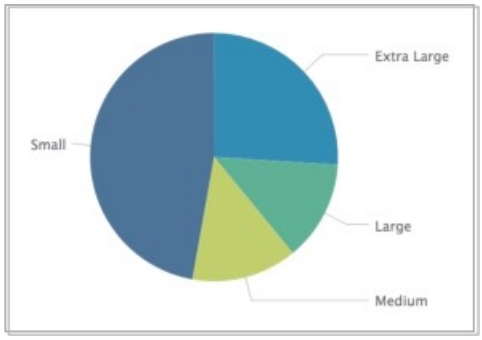


17. Search online transactions `[access_combined]` during the **last 24 hours** and—using the `case` function of the `eval` command—classify the size `(bytes)` of events into a field called `dataSize`. If the event is less than 2,000 bytes, classify it as Small; if 2,000 or more but less than 2,500 bytes, classify as Medium; finally, if 2,500 or more but less than 3,000 bytes, classify as Large. Include a default value of Extra Large for all events where the bytes value is 3,000 or greater .

Results example:

18. Using `chart` or `stats`, count the events by `dataSize` and display the results as a pie chart.

Results example:

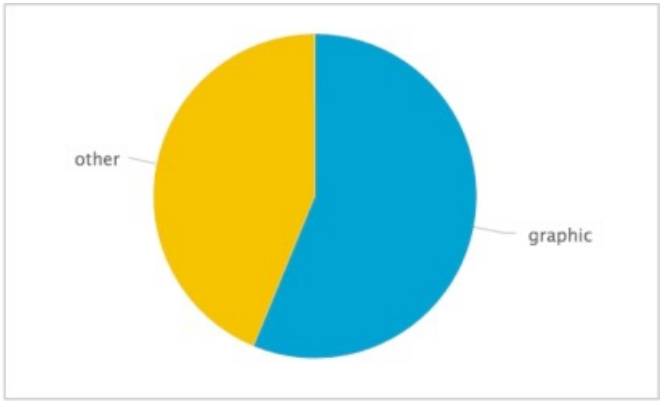


19. Save your search with the name **L4S4**.

CHALLENGE Exercise:

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

Final Results Example:



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

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

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

22. 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%"`

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

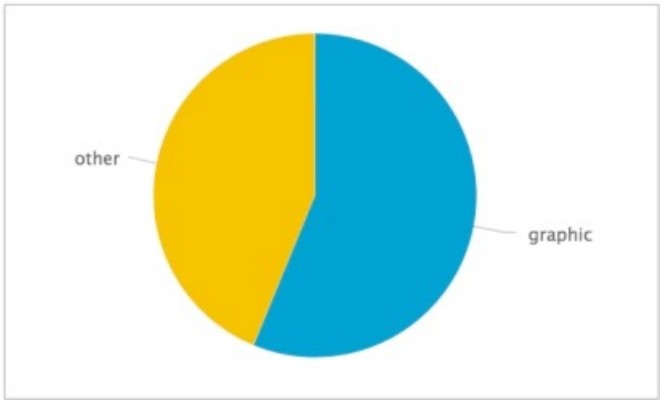
23. Use another `stats` or `chart` command to sum the `count` column by the `type` field. Rename the sum of the `count` calculation to `total`.

Results Example:

type	total
graphic	3583
other	2296

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

Results Example:



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

CHALLENGE Exercise:

Report which one-hour periods over the last 24 hours have seen the number of Buttercup Games online sales twice as numerous as the number of sales in retail stores.

Final Results Example:

_time	sales	web
2019-09-11 10:00	39	139
2019-09-11 11:00	40	122
2019-09-11 12:00	40	174
2019-09-11 13:00	36	145
2019-09-11 14:00	36	143
2019-09-11 15:00	39	142

26. Search online sales data [`access_combined`] and retail sales data [`vendor_sales`] for successful purchases during the **last 24 hours**.
27. Use `timechart` to count the sales events by `index` using a sampling interval of 1 hour.

Results Example:

_time	sales	web
2019-09-11 11:00	40	122
2019-09-11 12:00	40	174
2019-09-11 13:00	36	145
2019-09-11 14:00	36	143
2019-09-11 15:00	39	142
2019-09-11 16:00	39	159

28. Use a `where` command to keep only rows where the number of web sales are more than twice the number of retail sales.

Results Example:

_time	sales	web
2019-09-11 10:00	39	139
2019-09-11 11:00	40	122
2019-09-11 12:00	40	174
2019-09-11 13:00	36	145
2019-09-11 14:00	36	143
2019-09-11 15:00	39	142

29. Save your search as report, **L4C2**.
30. Modify your previous search to use search instead of where and observe the results. Why are the results different?