Correlation Analysis – Lab Solutions Guide

Overview

Welcome to the Splunk Education lab environment. These lab exercises will test your knowledge of using Splunk commands to analyze and correlate events.

Scenario

You will use data from the international video game company, Buttercup Games. A list of source types is provided below.

NOTE: This is a lab environment driven by data generators with obvious limitations. This is not a production environment. Screenshots approximate what you should see, not the **exact** output.

Index	Type	Sourcetype	Interesting Fields
web	Online sales	**access_combined**	action, bytes, categoryId, clientip, itemId, JSESSIONID, price, productId, product_name, referer, referer_domain, sale_price, status, user, useragent
network	Web security appliance data	**cisco_wsa_squid**	action, cs_method, cs_mime_type, cs_url, cs_username, sc_bytes, sc_http_status, sc_result_code, severity, src_ip, status, url, usage, x_mcafee_virus_name, x_wbrs_score, x_webcat_code_abbr

©2023 Splunk Inc. All rights reserved.

splunk>

Common Commands and Functions

These commands and statistical functions are commonly used in searches but may not have been explicitly discussed in the course. Please use this table for quick reference. Click on the hyperlinked SPL (Search Processing Language) to be taken to the Search Manual for that command or function.

SPL	Type	Description	Example
sort	command	Sorts results in descending or ascending order by a specified field. Can limit results to a specific number.	*Sort the first 100 `src_ip` values in descending order* `| sort 100 -src_ip`
where	command	Filters search results using eval-expressions.	*Return events with a `count` value greater than 30* `| where count > 30`
rename	command	Renames one or more fields.	*Rename `SESSIONID` to 'The session ID'* `| rename SESSIONID as "The session ID"`
fields	command	Keeps (+) or removes (-) fields from search results.	*Remove the `host` field from the results* `| fields - host`
stats	command	Calculates aggregate statistics over the results set.	*Calculate the total sales, i.e. the sum of `price` values.* `| stats sum(price)`
eval	command	Calculates an expression and puts the resulting value into a new or existing field.	*Concatenate `first_name` and `last_name` values with a space to create a field called "`full_name`"* `| eval full_name=first_name." ".last_name`
table	command	Returns a table.	*Output `vendorCountry`, `vendor`, and `sales` values to a table* `| table vendorCountry, vendor, sales`
sum()	statistical function	Returns the sum of the values of a field. Can be used with **`stats`**, **`timechart`**, and **`chart`** commands.	*Calculate the sum of the `bytes` field* `| stats sum(bytes)`
count or count()	statistical function	Returns the number of occurrences of all events or a specific field. Can be used with **`stats`**, **`timechart`**, and **`chart`** commands.	*Count all events as "events" and count all events that contain a value for `action` as "action"* `| stats count as events, count(action) as action`

Refer to the Search Reference Manual for a full list of commands and functions.

©2023 Splunk Inc. All rights reserved.

Lab Exercise 1 – Calculate Co-Occurrence Between Fields

Description

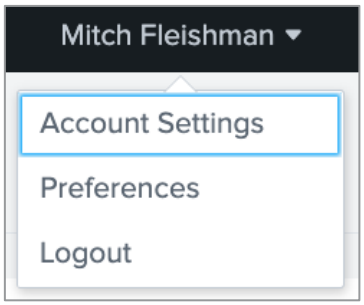
Configure the lab environment user account. Then, use the **`transaction`** command to correlate events.

Steps

Task 1: Log into Splunk and change the account name and time zone.

Set up your lab environment to fit your time zone. This also allows the instructor to track your progress and assist you if necessary.

1. Log into your Splunk lab environment using the username and password provided to you.
2. You may see a pop-up window welcoming you to the lab environment. You can click **Continue to Tour** but this is not required. Click **Skip** to dismiss the window.
3. Click on the username you logged in with (at the top of the screen) and then choose **Account Settings** from the drop-down menu.
4. In the **Full name** box, enter your first and last name.
5. Click **Save**.
6. Reload your browser to reflect the recent changes to the interface. (This area of the web interface will be referred to as ***user name***.)



After you complete step 6, you will see your name in the web interface.

NOTE: Sometimes there can be delays in executing an action like saving in the UI or returning results of a search. If you are experiencing a delay, please allow the UI a few minutes to execute your action.

7. Navigate to ***user name*** **> Preferences.**
8. Choose your local time zone from the **Time zone** drop-down menu.
9. Click **Apply**.
10. (Optional) Navigate to ***user name*** **> Preferences > SPL Editor > Search auto-format** and click on the toggle to activate auto-formatting. Then click **Apply**. When the pipe character is used in search, the SPL Editor will automatically begin the pipe on a new line.

Search auto-format disabled (default)

Search auto-format enabled

Scenario: **Sales wants a report of all purchase events correlated with a unique `JSESSIONID` over the last 60 minutes. The report should include information about the time of the event, the actions performed during the session, and the client IP.**

Task 2: **Correlate events based on `JSESSIONID` that involve a value for `action`. Then, filter results to show only events that involved a purchase.**

11. Search for all events in the online store (**`index=web sourcetype=access_combined`**) during the **last 60 minutes**.

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

12. Display a table that shows the **`_time`**, **`clientip`**, **`JSESSIONID`**, and **`action`** fields. Note that the actions are listed in reverse chronological order (most to least recent.) (**Hint**: Use the **`table`** command.)

```
index=web sourcetype=access_combined
| table _time, clientip, JSESSIONID, action
```

Events Patterns **Statistics (357)** Visualization

20 Per Page ▾ Format Preview ▾ < Prev 1 2 3 4 5 6 7 8 ...

_time	clientip	JSESSIONID	action
2020-04-06 03:10:21	130.253.37.97	SD1SL7FF4ADFF4957	purchase
2020-04-06 03:10:21	130.253.37.97	SD1SL7FF4ADFF4957	purchase
2020-04-06 03:10:19	130.253.37.97	SD1SL7FF4ADFF4957	addtocart
2020-04-06 03:10:12	130.253.37.97	SD1SL7FF4ADFF4957	
2020-04-06 03:10:04	130.253.37.97	SD1SL7FF4ADFF4957	purchase

13. Modify your search to only include events with a value in the **`action`** field.

```
index=web sourcetype=access_combined action=*
| table _time, clientip, JSESSIONID, action
```

Events Patterns **Statistics (231)** Visualization

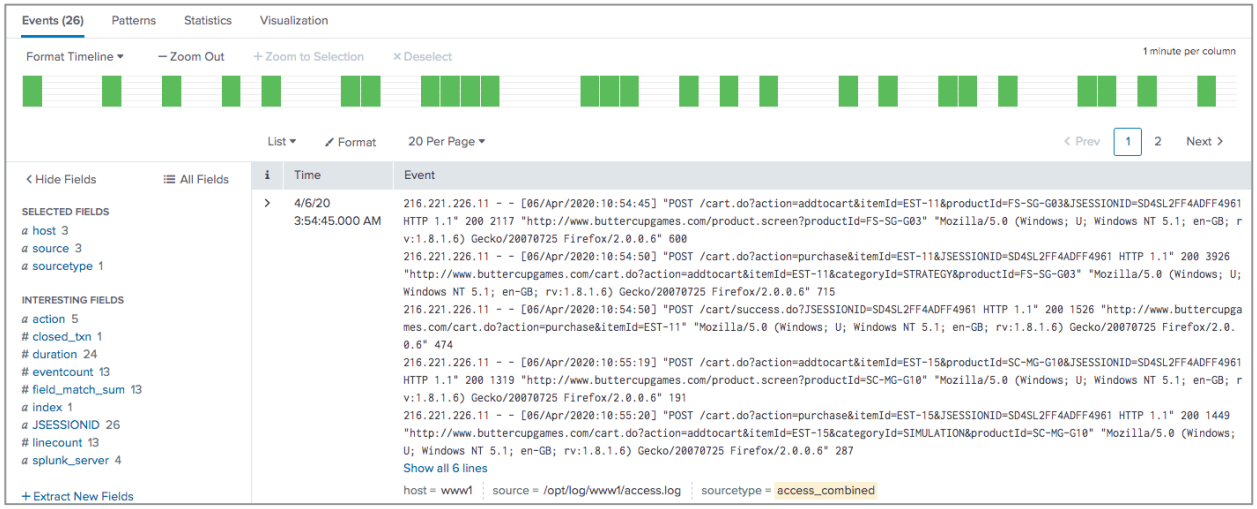
20 Per Page ▾ Format Preview ▾ < Prev 1 2 3 4 5 6 7 8 ...

_time	clientip	JSESSIONID	action
2020-04-06 03:11:34	91.217.178.210	SD5SL5FF8ADFF4966	purchase
2020-04-06 03:11:34	91.217.178.210	SD5SL5FF8ADFF4966	purchase
2020-04-06 03:11:32	91.217.178.210	SD5SL5FF8ADFF4966	addtocart
2020-04-06 03:11:19	91.217.178.210	SD5SL5FF8ADFF4966	view
2020-04-06 03:11:14	91.217.178.210	SD5SL5FF8ADFF4966	purchase

14. Remove the **`table`** command and all the arguments being passed to it. Use the **`transaction`** command to create groups of transactions based on the **`JSESSIONID`** field.

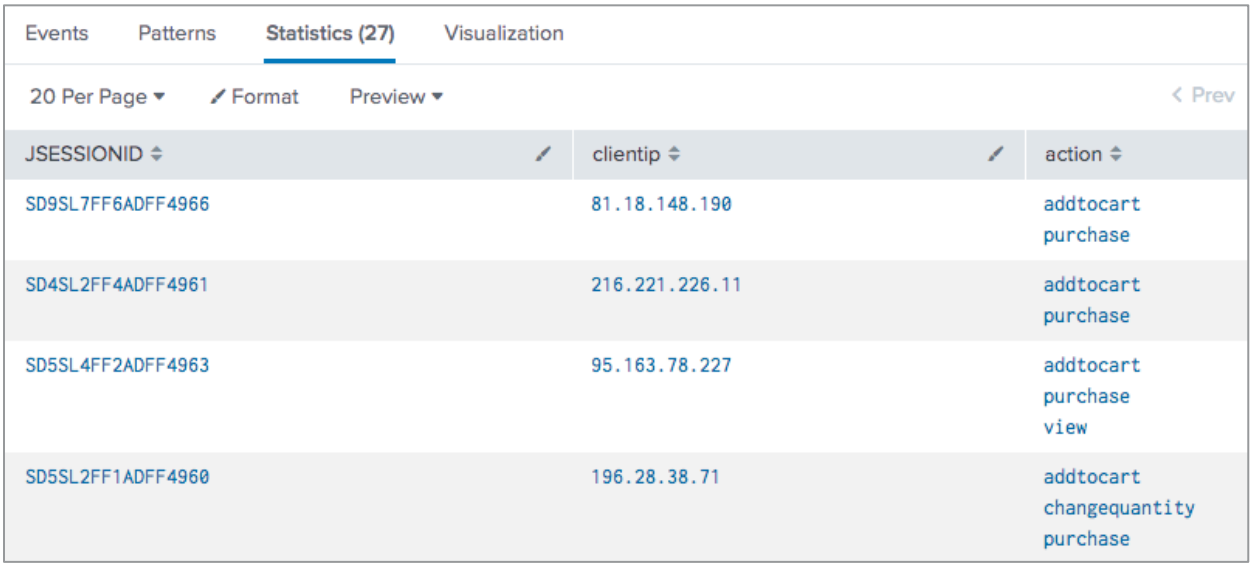
```
index=web sourcetype=access_combined action=*
| transaction JSESSIONID
```

©2023 Splunk Inc. All rights reserved.



15. Modify your search to display the transactions in a table.

```
index=web sourcetype=access_combined action=*
| transaction JSESSIONID
| table JSESSIONID, clientip, action
```

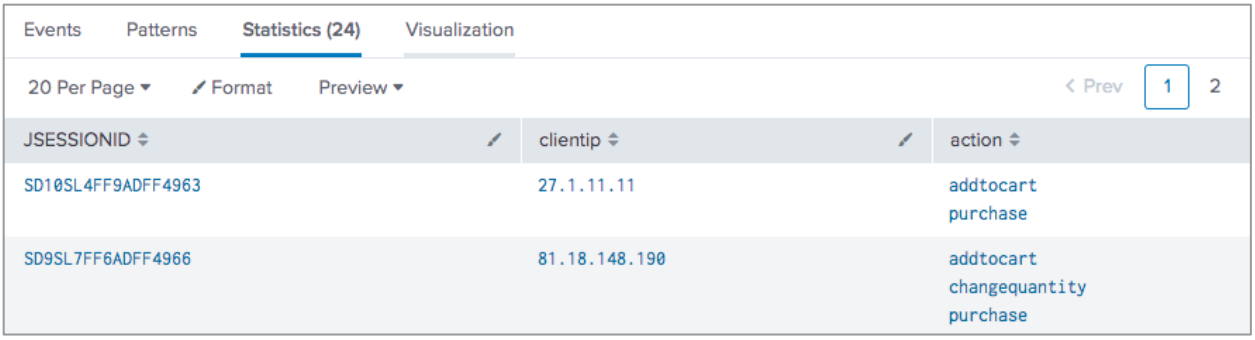


NOTE: By default, the values in the action column are ordered alphabetically, ignoring duplicates.

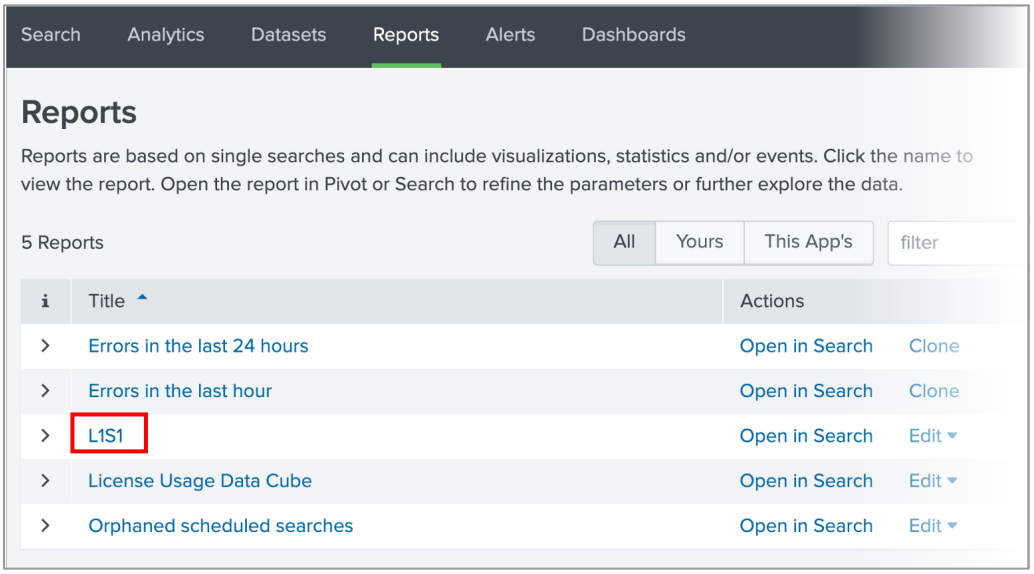
16. View only transactions that contain at least one purchase event. Use the **`search`** command to find transactions containing a purchase.

NOTE: The search command must be downstream from the **`transaction`** command.

```
index=web sourcetype=access_combined action=*
| transaction JSESSIONID
| table JSESSIONID, clientip, action
| search action=purchase
```



17. Save your search as a report with the name **L1S1**.
    a. Click **Save As** > **Report**
    b. For **Title**, enter L1S1.
    c. **Save**.
    d. You can **View** your report or exit out of the **Your Report Has Been Created** window by clicking the **X** in the upper-right corner.
    e. You can access your saved reports using the **Reports** tab in the application bar.



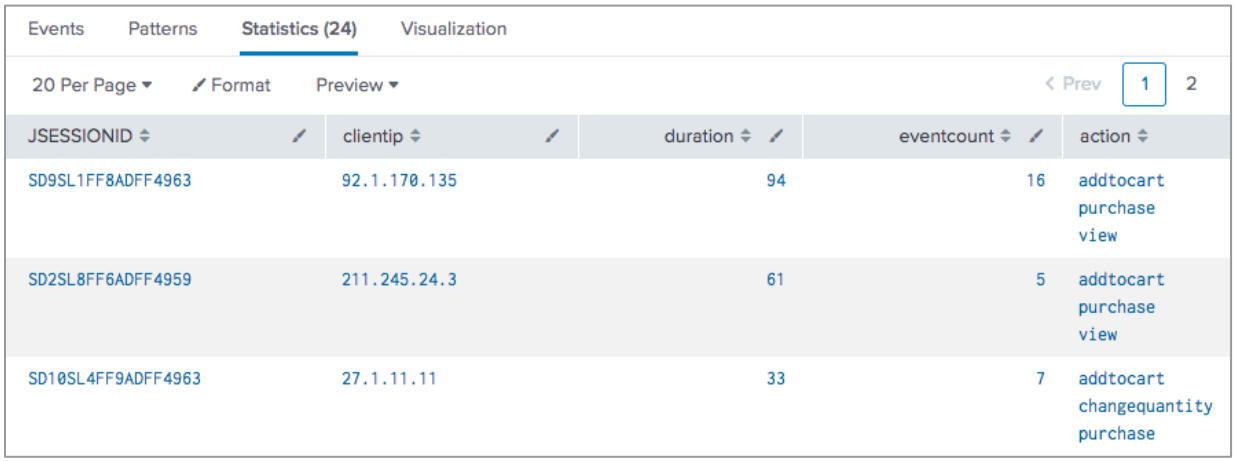
*Your recently saved **L1S1** report will be visible in the **Reports** tab.*

Scenario: **Sales needs a report of online store transactions that lasted longer than one minute and involved the `purchase` action.**

Task 3: Edit the previous search so that the `duration` field is available to manipulate. Then, use this field to filter results to only show events longer than 1 minute.

18. If not already displayed, run your **L1S1** search again.
    a. Set the search mode to **Verbose Mode**, which will re-execute your search.
    b. Click the **Events** tab. Notice the new fields generated by the **`transaction`** command: **`duration`** and **`eventcount`**.
19. Modify your search to add the `duration` and **`eventcount`** fields to your table after the **`clientip`** field. Run your search in **Smart Mode**.

```
index=web sourcetype=access_combined action=*
| transaction JSESSIONID
| table JSESSIONID, clientip, duration, eventcount, action
| search action=purchase
```



JSESSIONID	clientip	duration	eventcount	action
SD9SL1FF8ADFF4963	92.1.170.135	94	16	addtocart purchase view
SD2SL8FF6ADFF4959	211.245.24.3	61	5	addtocart purchase view
SD10SL4FF9ADFF4963	27.1.11.11	33	7	addtocart changequantity purchase

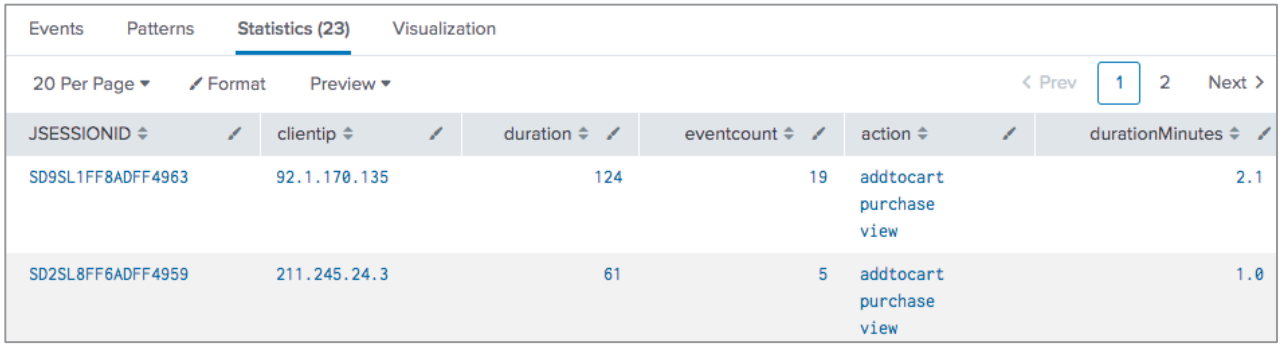
20. Pipe results to the following **`eval`** command.

```
| eval durationMinutes=round(duration/60,1)
```

The **`eval`** command creates a new field called **`durationMinutes`** and populates this field with the value of **`duration`** divided by 60 rounded to 1 decimal place.

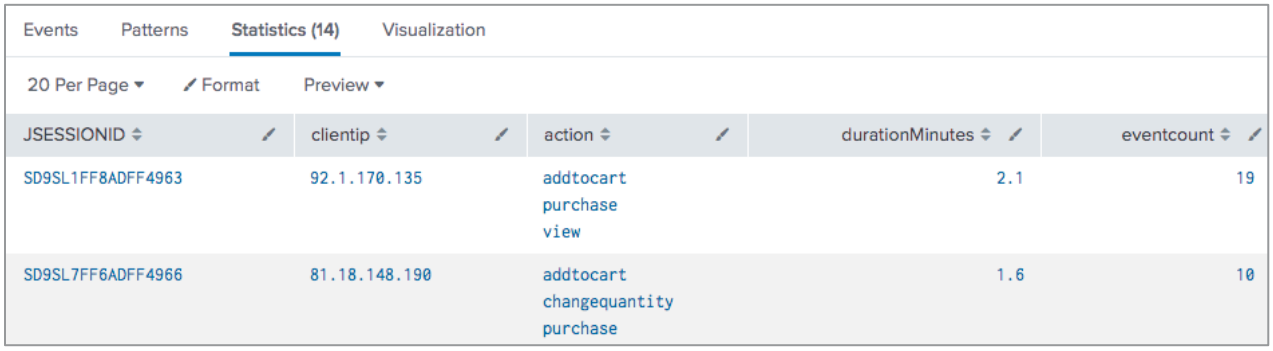
```
index=web sourcetype=access_combined action=*
| transaction JSESSIONID
| table JSESSIONID, clientip, duration, eventcount, action
| search action=purchase
| eval durationMinutes=round(duration/60,1)
```

©2023 Splunk Inc. All rights reserved.



21. Modify your search to find data where the **durationMinutes** is greater than one minute. Adjust the table to display only **JSESSIONID**, **clientip**, **action**, **durationMinutes**, and **eventcount**, in that order. (**Hint** Refer to the Common Commands and Functions table at the beginning of this document to find a command that filters search results.)

```
index=web sourcetype=access_combined action=*
| transaction JSESSIONID
| search action=purchase
| eval durationMinutes=round(duration/60,1)
| table JSESSIONID, clientip, action, durationMinutes, eventcount
| where durationMinutes > 1
```



22. Save your search as report, **L1S2**.

Scenario: Sales wants a report of all events correlated with a unique `clientip` over the last 60 minutes that began with the `addtocart` action and ended with the `purchase` action.

Task 4: Use the `transaction` command with the `startswith` and `endswith` options to group events by `clientip` that started with `action=addtocart` and ended with `action=purchase`.

23. Search for all events from the online store (**`index=web sourcetype=access_combined`**) in the **last 60 minutes** and correlate the events based on **`clientip`**.

```
index=web sourcetype=access_combined
| transaction clientip
```

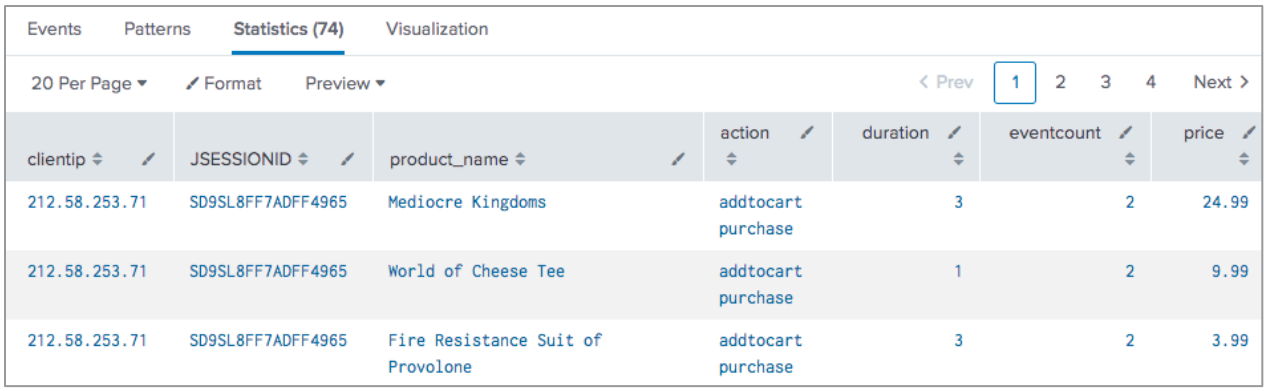
24. Use the **`startswith`** and **`endswith`** options of the **`transaction`** command to display transactions that begin with an **`addtocart`** action and end with a **`purchase`** action.

```
index=web sourcetype=access_combined
| transaction clientip startswith=action=addtocart endswith=action=purchase
```

25. Display **`clientip`**, **`JSESSIONID`**, **`product_name`**, **`action`**, **`duration`**, **`eventcount`**, and **`price`** in a table.

©2023 Splunk Inc. All rights reserved.

```
index=web sourcetype=access_combined
| transaction clientip startswith=action=addtocart endswith=action=purchase
| table clientip, JSESSIONID, product_name, action, duration, eventcount, price
```



Events Patterns Statistics (74) Visualization

20 Per Page ▾ Format Preview ▾ < Prev 1 2 3 4 Next >

clientip	JSESSIONID	product_name	action	duration	eventcount	price
212.58.253.71	SD9SL8FF7ADFF4965	Mediocre Kingdoms	addtocart purchase	3	2	24.99
212.58.253.71	SD9SL8FF7ADFF4965	World of Cheese Tee	addtocart purchase	1	2	9.99
212.58.253.71	SD9SL8FF7ADFF4965	Fire Resistance Suit of Provolone	addtocart purchase	3	2	3.99

26. Save your search as report, **L1S3**.

CHALLENGE Exercise: Report the most common HTTP status errors that occurred during the last 30 days on the online sales web servers and the internal web appliance within a proximity of 5 minutes or less. Only include days with more than 5 of these frequent errors.

27. Search HTTP status error events (**`status>399`**) from the online sales web servers (**`index=web sourcetype=access_combined`**) and the web appliance (**`index=network sourcetype=cisco_wsa_squid`**) during the **last 30 days**. For best performance, use the **`fields`** command to limit extracted fields to only **`sourcetype`** and **`status`**. (**Hint**: See the Common Commands and Functions table for information on how to use **`fields`**.)

```
(index=network sourcetype=cisco_wsa_squid) OR
(index=web sourcetype=access_combined) status>399
| fields sourcetype, status
```

28. Create transactions based on **`status`** field values and limit the span to 5 minutes.

NOTE: If you do not see results, increase the **`maxspan`** value.

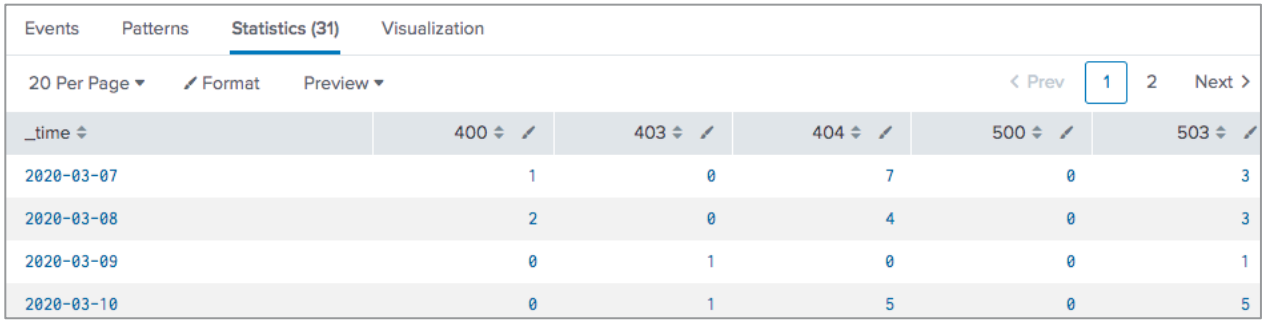
```
(index=network sourcetype=cisco_wsa_squid) OR
(index=web sourcetype=access_combined) status>399
| fields sourcetype, status
| transaction status maxspan=5m
```

29. Limit the results to only transactions that contain at least one event from each sourcetype.

```
(index=network sourcetype=cisco_wsa_squid) OR
(index=web sourcetype=access_combined) status>399
| fields sourcetype, status
| transaction status maxspan=5m
| search sourcetype=access_combined AND sourcetype=cisco_wsa_squid
```

30. Use **`timechart`** to count events by **`status`**.

```
(index=network sourcetype=cisco_wsa_squid) OR
(index=web sourcetype=access_combined) status>399
| fields sourcetype, status
| transaction status maxspan=5m
| search sourcetype=access_combined AND sourcetype=cisco_wsa_squid
| timechart count by status
```



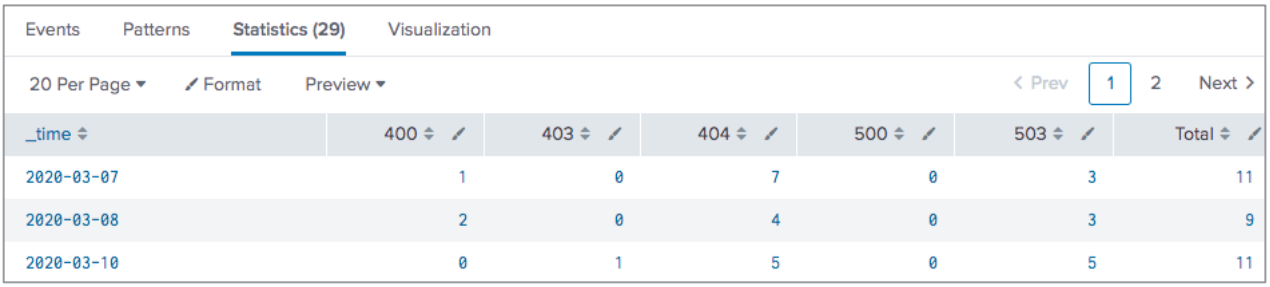
Events Patterns **Statistics (31)** Visualization

20 Per Page ▾ Format Preview ▾ < Prev 1 2 Next >

_time	400	403	404	500	503
2020-03-07	1	0	7	0	3
2020-03-08	2	0	4	0	3
2020-03-09	0	1	0	0	1
2020-03-10	0	1	5	0	5

31. Discard rows that have fewer than 5 errors for all **`status`** values. (**Hint:** Use the **`addtotals`** command without additional arguments.)

```
(index=network sourcetype=cisco_wsa_squid) OR
(index=web sourcetype=access_combined) status>399
| fields sourcetype, status
| transaction status maxspan=5m
| search sourcetype=access_combined AND sourcetype=cisco_wsa_squid
| timechart count by status
| addtotals
| search Total>4
```



Events Patterns **Statistics (29)** Visualization

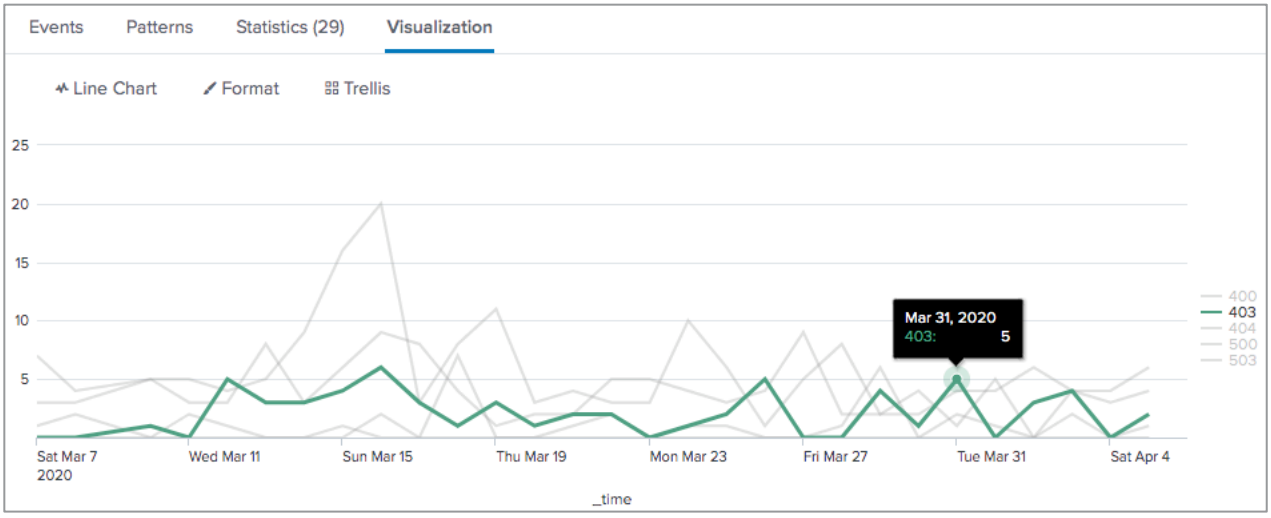
20 Per Page ▾ Format Preview ▾ < Prev 1 2 Next >

_time	400	403	404	500	503	Total
2020-03-07	1	0	7	0	3	11
2020-03-08	2	0	4	0	3	9
2020-03-10	0	1	5	0	5	11

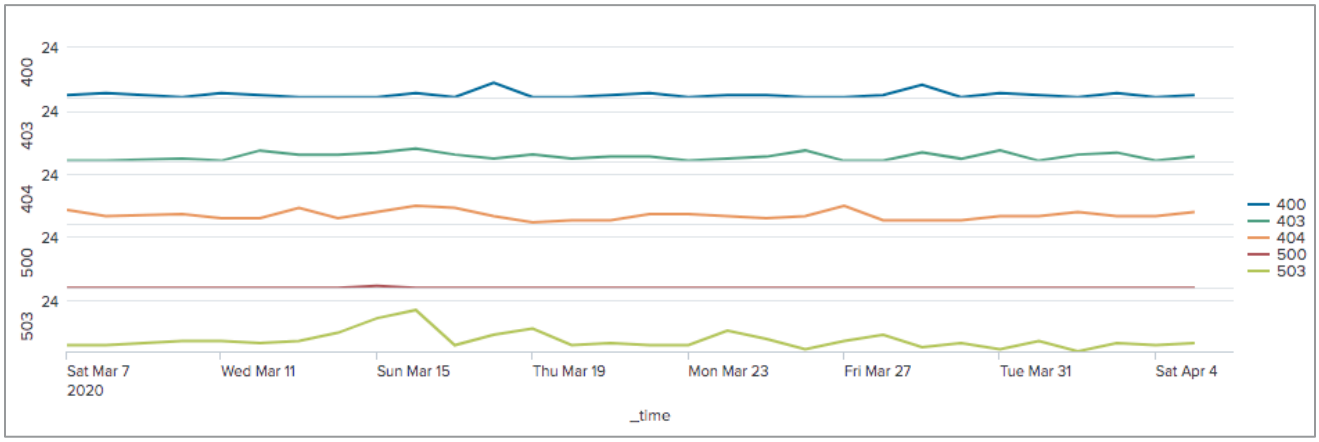
32. Remove the **`Total`** column and display the data as a **Line chart**.

```
(index=network sourcetype=cisco_wsa_squid) OR
(index=web sourcetype=access_combined) status>399
| fields sourcetype, status
| transaction status maxspan=5m
| search sourcetype=access_combined AND sourcetype=cisco_wsa_squid
| timechart count by status
| addtotals
| search Total>4
| fields - Total
```

©2023 Splunk Inc. All rights reserved.



33. Save your search as report, **L1X**.
34. Optionally, for this line chart, set **Multi-series Mode** to **Yes**. Observe the change in how the lines are represented. (**Hint**: It's one of the **Format** options on the **General** tab.)



Lab Exercise 2 – Analyze Multiple Data Sources

Description

Use the **append** command to analyze dissimilar data sources into one search.

Steps

Scenario: **The Sales department would like to see a list of sales by `productid` for the last hour as well as the previous hour.**

Task 1: **Use the `append` command to create a search that displays results from two different time ranges. Then, align results using the `first` function.**

1. Search for successful purchase events in the online store that involve a value for **`productId`** (**`index=web sourcetype=access* productId=* action=purchase status=200`**) over the **last 24 hours** from the previous hour. (**Hint**: Include the following time modifiers in your basic search: **`earliest=-1h@h latest=@h`**.)

   ```
   index=web sourcetype=access* productId=* action=purchase status=200 earliest=-1h@h
   latest=@h
   ```

2. Pipe results to the following **`stats`** command.

   ```
   | stats sum(price) as lastHourSales by productId
   ```

   The **`stats`** command calculates the sum of **`price`** values for each **`productId`**. The values are listed under a column called **`lastHourSales`**.

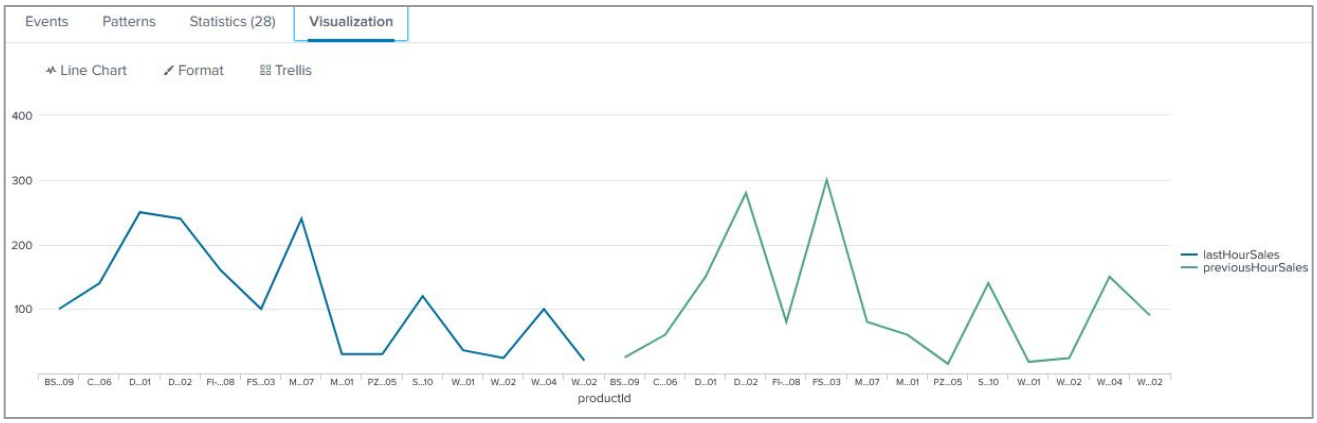
   ```
   index=web sourcetype=access* productId=* action=purchase status=200 earliest=-1h@h
   latest=@h
   | stats sum(price) as lastHourSales by productId
   ```

3. Use the **`append`** command to add an additional search of the previous hour. This search will look similar to the first search with the following differences:

   - The time modifiers should capture the previous hour: **`earliest=-2h@h latest=-1h@h`**
   - The results of the calculation performed by **`stats`** should be named "previousHourSales".

   ```
   index=web sourcetype=access* productId=* action=purchase status=200 earliest=-1h@h
   latest=@h
   | stats sum(price) as lastHourSales by productId
   | append
     [search index=web sourcetype=access* productId=* action=purchase status=200
     earliest=-2h@h latest=-1h@h
     | stats sum(price) as previousHourSales by productId]
   ```

4. Select **Visualization**, then select the **Line Chart**.



5. Your results should look misaligned and less meaningful. Use the **`first`** function to overlay the two searches into one clean line chart.

```
index=web sourcetype=access* productId=* action=purchase status=200 earliest=-1h@h
latest=@h
| stats sum(price) as lastHourSales by productId
  | append
  [search index=web sourcetype=access* productId=* action=purchase status=200
  earliest=-2h@h latest=-1h@h
  | stats sum(price) as previousHourSales by productId]
| stats first(*) as * by productId
| fillnull
```

6. Save your results as a report named **L2S1.**