Result Modification

Document Usage Guidelines

- Should be used only for enrolled students
- Not meant to be a self-paced document, an instructor is needed
- Do not distribute

Course Goals

- Manipulate fields and field values
- Modify results sets
- Normalize fields with **`eval`**

Course Outline

- Manipulating Output
- Modifying Result Sets
- Modifying Field Values
- Normalizing with **`eval`**

Manipulating Output

Topic Objectives

- Convert a 2-dimensional table into a flat table with `untable`
- Convert a flat table into a 2-dimensional table with `xyseries`

Comparing `xyseries` and `untable`

`xyseries` and `untable` commands perform opposite tasks

_time	www1	www2	www3
2020-09-15 18:00	317928	353688	305128
2020-09-15 19:00	309588	201784	256223
2020-09-15 20:00	516013	331965	145669
2020-09-15 21:00	381120	203596	312564
2020-09-15 22:00	439490	371449	70308
2020-09-15 23:00	312041	283474	251202
6 00:00	349520	322397	349439

tabular output

```
...| untable _time, host, totalBytes
```

_time	host	totalBytes
2020-09-15 18:00	www1	317928
2020-09-15 18:00	www2	353688
2020-09-15 18:00	www3	305128
2020-09-15 19:00	www1	309588
2020-09-15 19:00	www2	201784
2020-09-15 19:00	www3	256223
-15 20:00	www1	516013

stats-like output

_time	host	totalBytes
2020-09-06 00:00	www1	399861
2020-09-06 00:00	www2	361365
2020-09-06 00:00	www3	291661
2020-09-06 01:00	www1	368440
2020-09-06 01:00	www2	312665
2020-09-06 01:00	www3	385473
2020-09-06 02:00	www1	

stats-like output

```
...| xyseries _time host totalBytes
```

_time	www1	www2	www3
2020-09-06 00:00	399861	361365	291661
2020-09-06 01:00	368440	312665	385473
2020-09-06 02:00	232468	497763	353482
2020-09-06 03:00	388791	423780	414596
2020-09-06 04:00	435896	252771	210099
2020-09-06 05:00	276943	264455	225258
2020-09-06 06:00	316618	190335	

tabular output

`untable` Command

```
...| untable <x-field> <y-name-field> <y-data-field>
```

- Reformats 2-dimensional output as a flat table (**`stats`**-like output)
- Identify the **`<x-field>`** that will become the first column in the output
- Specify the data series labels with **`<y-name-field>`**
- Name the **`<y-data-field>`** that contains the chartable numeric data

```
...| untable product_name VendorCountry prod_count
```

product_name	Ethiopia	Morocco	Nigeria	South Africa
Benign Space Debris	0	5	5	10
Dream Crusher	5	3	4	3
Final Sequel	6	4	2	10
Fire Resistance Suit of Provolone	0	2	3	3
Holy Blade of Gouda	2	2	1	3
Manganiello Bros.	0	0	0	4
Manganiello Bros. Tee	0	3	1	7
Mediocre Kingdoms	3	2	2	12

y-name-field

y-data-field

product_name	VendorCountry	prod_count
Benign Space Debris	Ethiopia	0
Benign Space Debris	Morocco	5
Benign Space Debris	Nigeria	5
Benign Space Debris	South Africa	10
Benign Space Debris	Tanzania	3

`untable` Command Example

Scenario

Sales wants to know the top selling products for the 5 highest-performing African countries over the last week

These results display the top 5 most profitable countries but fails to limit results to the top product in each country

product_name	Ethiopia	Morocco	Nigeria	South Africa	Tanzania
Benign Space Debris	0	5	5	10	3
Dream Crusher	5	3	4	3	0
Final Sequel	6	4	2	10	0
Fire Resistance Suit of Provolone	0	2	3	3	0
Holy Blade of Gouda	2	2	1	3	1
Manganiello Bros.	0	0	0	4	0
Manganiello Bros. Tee	0	3	1	7	0
Mediocre Kingdoms	3	2	2	12	4
SIM Cubicle	0	3	1	1	4
World of Cheese	0	0	1	2	2
World of Cheese Tee	0	0	0	1	1

`untable` Command Example (cont.)

A. `untable` manipulates the results into chartable output

B. It is now easier to sort `VendorCountry` and `prod_count` to find the top product for each country

```
...
| untable product_name, VendorCountry, prod_count
```

A

product_name	Ethiopia	Morocco	Nigeria	South Africa	Tanzania
Benign Space Debris	0	5	5	10	3
Dream Crusher	5	3	4	3	0
Final Sequel	6	4	2	10	0
Fire Resistance Suit of Provolone	0	2	3	3	0
Holy Blade of Gouda	2	2	1	3	1
Manganiello Bros.	0	0	0	4	0
Manganiello Bros. Tee	0	3	1	7	0
Mediocre Kingdoms	3	2	2	12	4
SIM Cubicle	0	3	1	1	4
World of Cheese	0	0			
World of Cheese Tee	0	0			

yfield

B

product_name	VendorCountry	prod_count
Benign Space Debris	Ethiopia	0
Benign Space Debris	Morocco	5
Benign Space Debris	Nigeria	5
Benign Space Debris	South Africa	10
Benign Space Debris	Tanzania	3

`untable` Command Example (cont.)

- In this format, the data can be piped to other commands to fulfill the scenario
  - The **`where`** and **`stats`** commands are outside the scope of this course
  - The **`eventstats`** command will be discussed later in this course

```
...
| untable product_name, VendorCountry, prod_count
| eventstats max(prod_count) as max by VendorCountry
| where prod_count=max
| stats list(product_name), list(prod_count) by VendorCountry
```

VendorCountry	list(product_name)	list(prod_count)
Ethiopia	Final Sequel	6
Morocco	Benign Space Debris	5
Nigeria	Benign Space Debris	5
South Africa	Mediocre Kingdoms	12
Tanzania	Mediocre Kingdoms	4
	SIM Cubicle	4

`xyseries` Command

```
...| xyseries <x-field> <y-name-field> <y-data-field>
```

- Reformats **`stats`**-like output as chartable tabular output
- Define the x-axis labels with **`<x-field>`**
- Define y-axis data series labels with **`<y-name-field>`**
- Chose a **`<y-data-field>`** that contains the chartable numerical data

```
...| xyseries _time host MB
```

_time	host	totalBytes	MB
2019-03-24 00:00	www1	334243	0.32
2019-03-24 00:00	www2	240765	0.23
2019-03-24 00:00	www3	301519	0.29
2019-03-24 01:00	www1	345347	0.33
2019-03-24 01:00	www2	220408	0.21

y-name-field

y-data-field

_time	www1	www2	www3
2019-03-24 00:00	0.32	0.23	0.29
2019-03-24 01:00	0.33	0.21	0.25
2019-03-24 02:00	0.14	0.27	0.31
2019-03-24 03:00	0.17	0.18	0.36
2	0.16	0.31	0.22

`xyseries` Command Example

- A This `stats`-based search creates a meaningless visualization
- B Piping to the **`xyseries`** command generates chartable output that creates a meaningful visualization with distinct data series

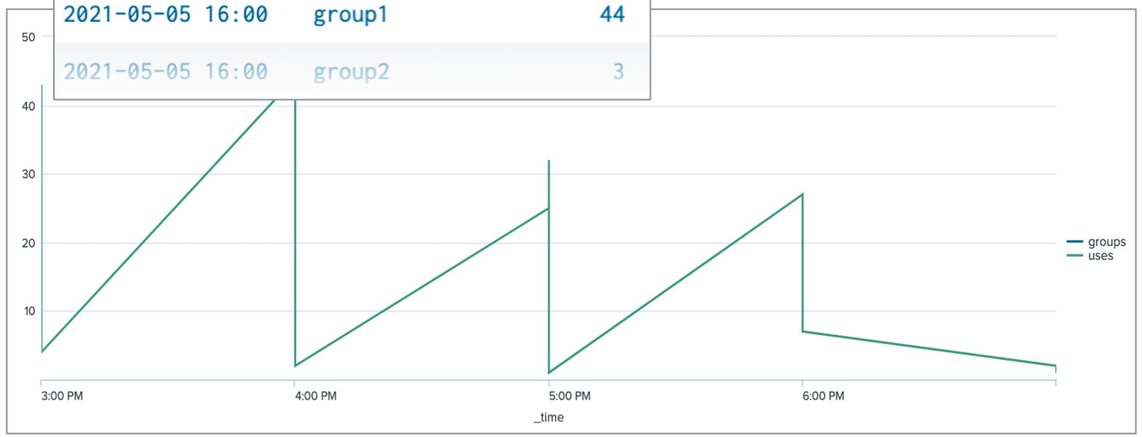
A

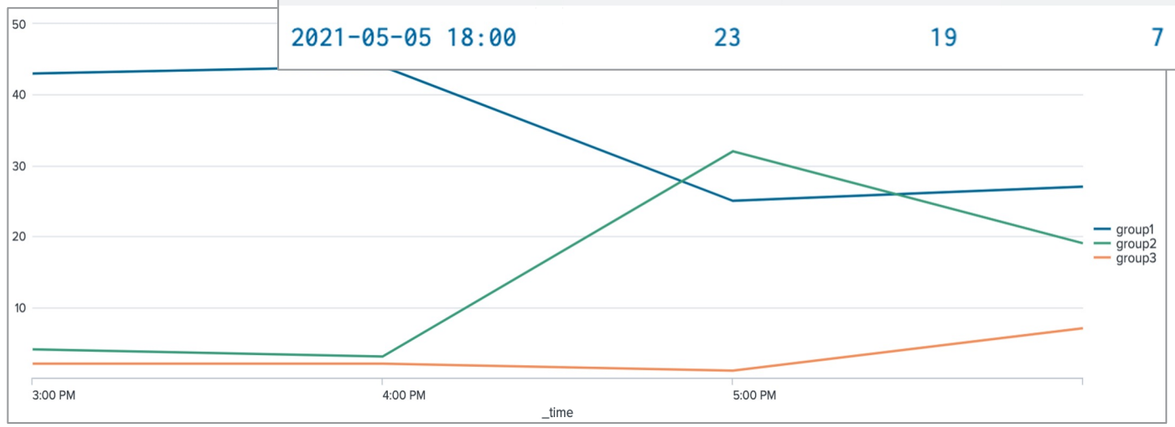
_time	groups	uses
2021-05-05 14:00	group3	1
2021-05-05 15:00	group1	43
2021-05-05 15:00	group2	4
2021-05-05 15:00	group3	2
2021-05-05 16:00	group1	44
2021-05-05 16:00	group2	3

```
...
| xyseries _time, groups, uses
| fillnull value=0
```

B

_time	group1	group2	group3
2021-05-05 14:00	0	0	1
2021-05-05 15:00	43	4	2
2021-05-05 16:00	44	3	2
2021-05-05 17:00	25	32	1
2021-05-05 18:00	23	19	7





xyseries versus chart

- Generally, **chart** is used to display a chartable output

```
...| chart a over b by c
```

is equivalent to:

```
...
| stats a by b,c
|...
| xyseries b c a
```

- If the aggregated field needs to be manipulated, then use **stats** followed by the varying commands and then **xyseries**

```
index=sales sourcetype=vendor_sales
| stats sum(price) as sales by Vendor product_name
| where sales > 100
| xyseries Vendor product_name sales
| fillnull
```

Vendor	Dream Crusher	Manganiello Bros.	Orvil the Wolverine
New Jack Games	119.97	0	0
Playtime Game & Hobby Shop	119.97	119.97	0
Spa und Spiele	0	119.97	119.97

Note

The **fillnull** command replaces null values with 0 by default.

chart, untable, xyseries versus stats

Using **chart**, **untable**, and **xyseries** is more efficient than a **stats**-based search in some cases

simple

```
index=web sourcetype=access_combined
categoryId!=TEE AND categoryId!=ACCESSORIES
| chart sum(price) as sales limit=5 useother=f
  by product_name, clientip
| untable product_name, clientip, sales
| xyseries clientip, product_name, sales
```

complex

```
index=web sourcetype=access_combined categoryId!=TEE AND
categoryId!=ACCESSORIES
| stats sum(price) as sales by product_name clientip
| eventstats sum(sales) as total_per_clientip by clientip
| sort -total_per_clientip clientip
| streamstats dc(clientip) as top_clientip
| search top_clientip < 6
| xyseries clientip product_name sales
```

clientip	Benign Space Debris	Curling 2014	Dream Crusher	Final Sequel	Manganiello Bros.	Mediocre Kingdoms	Orvil the Wolverine	Puppies vs. Zombies	SIM Cubicle	World of Cheese
107.3.146.207		159.92	319.92	174.93	439.89	199.92	159.96	14.97	159.92	49.98
194.215.205.19	199.92	119.94	279.93	199.92	439.89	124.95	239.94	54.89	19.99	149.94
201.3.120.132	99.96	39.98	479.88	124.95	319.92	224.91	159.96	24.95	39.98	99.96
211.166.11.101	49.98	119.94	239.94	149.94		149.94	239.94	44.91		399.84
87.194.216.51	174.93	79.96	679.83	274.89	359.91	324.87	279.93	19.96	399.80	499.80

Manipulating Output Lab Exercise

Time: 30 minutes

Tasks:

- Use the **`xyseries`** command to complete a search and create output that can be visualized
- Use the **`untable`** command to convert the results of a search to **`stats`**-like output
- Use the **`xyseries`** and **`untable`** commands to find the 3 most active customers of the 5 worst performing products

Modifying Result Sets

Topic Objectives

- Append data to search results with **`appendpipe`**
- Calculate event statistics with **`eventstats`**
- Calculate "streaming" statistics with **`streamstats`**

appendpipe Command

```
...| appendpipe [<subpipeline>]
```

- Transforms results and appends output to end of results set
- The **subpipeline** is executed when Splunk reaches the **appendpipe** command
  - Contains one or more transforming commands
  - Does not overwrite original results; instead, appends output as new lines to the bottom of the original results set
  - Multiple **appendpipes** can exist in a search

Note

A transforming command orders results into a data table that Splunk can use for statistical purposes.

`appendpipe` Command Example

Scenario

The CTO wants to find the number of nonbusiness-related connections to the internet for the last 24 hours, by user and usage, and the total attempts by usage.

- A Exclude Business **`usage`** connections in the basic search
- B Count events by each unique **`usage`** and **`user`** combination
- C Use **`appendpipe`** to calculate total event counts for each usage type

```
index=network sourcetype=cisco_wsa_squid usage!=Business   A
| stats count by usage, cs_username                         B
| appendpipe [stats sum(count) as count by usage]           C
```

usage	cs_username	count
Borderline	acurry@buttercupgames.com	3
Borderline	adombrowski@buttercupgames.com	1
Borderline	apreusig@buttercupgames.com	3
Borderline	apucci@buttercupgames.com	1
Borderline	basselin@buttercupgames.com	4

…at the end of the results…

Borderline	30
Personal	78
Unknown	122
Violation	1

`appendpipe` Command Example (cont.)

- The results table is complete but not easy to read

D Manipulate the subpipeline to create usage totals under the **`cs_username`** column

E Sort by **`usage`** to group categories with their totals and create a meaningful table

```
index=network sourcetype=cisco_wsa_squid usage!=Business
| stats count by usage, cs_username
| appendpipe
  [stats sum(count) as count by usage
  | eval cs_username = "Total for usage of ".usage]
| sort usage
```

D

usage	cs_username	count
Unknown	pbridgland@buttercupgames.com	2
Violation	gfacello@buttercupgames.com	1
Borderline	Total for usage of Borderline	30
Personal	Total for usage of Personal	78
Unknown	Total for usage	
Violation	Total for usage	

E

usage	cs_username	count
Borderline	cberztiss@buttercupgames.com	28
Borderline	ewarwick@buttercupgames.com	1
Borderline	yschonegge@buttercupgames.com	1
Borderline	Total for usage of Borderline	30
Personal	adombrowski@buttercupgames.com	1

`appendpipe` Command Example (cont.)

Scenario

Before you send off the report, you decide to add a grand total to the end of the report.

1st appendpipe

2nd appendpipe

```
index=network sourcetype=cisco_wsa_squid usage!=Business
| stats count by usage, cs_username
| appendpipe
  [stats sum(count) as count by usage
  | eval cs_username = "Total for usage of ".usage]
| sort usage
| appendpipe
  [search cs_username = "Total for*"
  | stats sum(count) as count
  | eval cs_username = "GRAND TOTAL"]
```

- Multiple **`appendpipe`** commands can be used in a search
- Second **`appendpipe`** adds up the usage totals and appends a grand total to results

usage	cs_username	count
Unknown	rroberts@buttercupgames.com	2
Unknown	svoronoff@buttercupgames.com	2
Unknown	syoungin@buttercupgames.co	2
Unknown	Total for usage of Unknown	117
Violation	gfacello@buttercupgames.com	1
Violation	Total for usage of Violation	1
	GRAND TOTAL	254

1st appendpipe

2nd appendpipe

`eventstats` Command

```
...| eventstats statsfunction(<field>) [as <field>]
```

- Generates statistics of all existing fields in your search results and saves them as values in new fields
- **`statsfunction`** is applied to **`<field>`** and the resulting value is appended to each of the results
- Supports multiple functions

eventstats Command Example

Scenario

For a new campaign, the online sales manager wants to see which products are losing more sales than the average during the last 24 hours, visualized in a bar chart.

A ...
B | eventstats avg(lostSales) as averageLoss

product_name	lostSales
Benign Space Debris	174.93
Curling 2014	119.94
Dream Crusher	559.86
	99.96
	27.93

A Results before the **eventstats** command

product_name	lostSales	averageLoss
Benign Space Debris	174.93	228.53500000000003
Curling 2014	19.94	228.53500000000003
Dream Crusher	9.86	228.53500000000003
Final Sequel	9.96	228.53500000000003
Fire Resistance Suit of Provolone	31.92	228.53500000000003

B The **eventstats** command calculates a value for **averageLoss** and appends this value to all events

eventstats Command Example (cont.)

Scenario

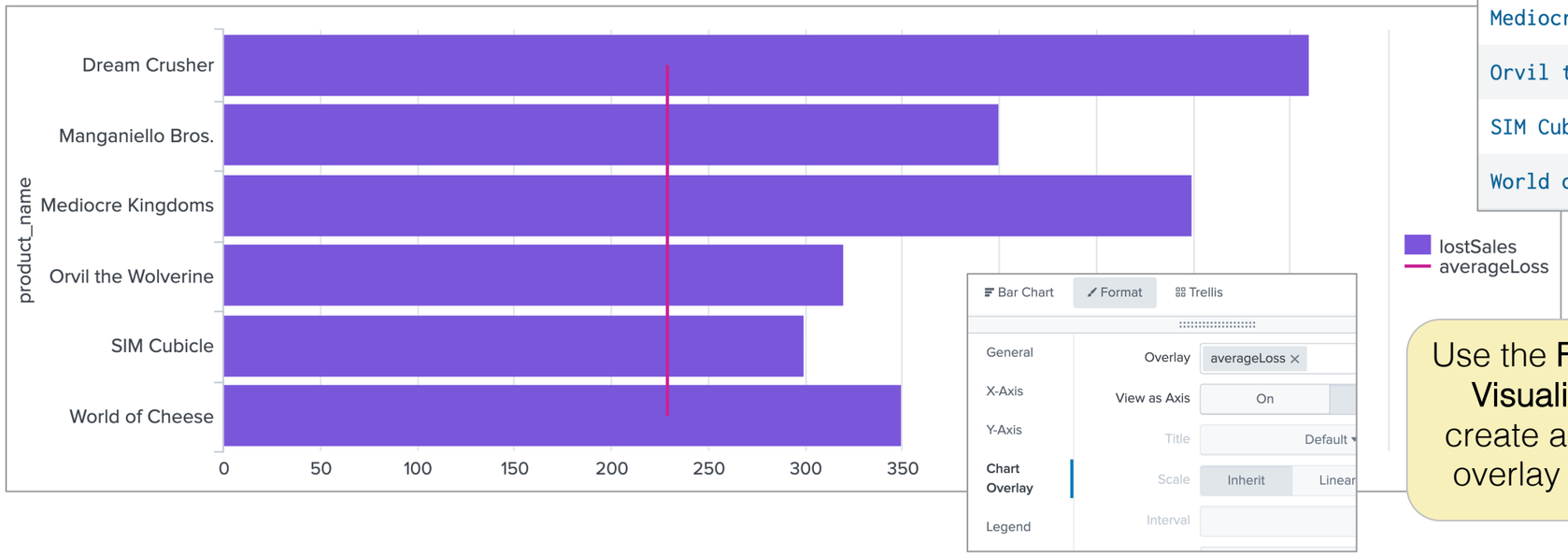
For a new campaign, the online sales manager wants to see which products are losing more sales than the average during the last 24 hours, visualized in a bar chart.

```
...
| eventstats avg(lostSales) as averageLoss
| where lostSales > averageLoss
```

The **where** command limits results where the values of **lostSales** exceeds the **averageLoss** value

This search has completed and has returned **6** results

product_name	lostSales	averageLoss
Dream Crusher	559.86	228.53500000000003
Manganiello Bros.	399.90	228.53500000000003
Mediocre Kingdoms	499.80	228.53500000000003
Orvil the Wolverine	319.92	228.53500000000003
SIM Cubicle	299.85	228.53500000000003
World of Cheese	349.86	228.53500000000003



Use the **Format** tab in the **Visualizations** tab to create an **averageLoss** overlay on a **Bar Chart**

`streamstats` Command

```
...| streamstats statsfunction(<field>) [as <field>] [by <field-list>]
     [window=<int>] [current=<Bool>]
```

- Calculates statistics for each result row at the time the command encounters it and adds these values to the results
- Supports multiple functions
- The following options are available:
  - **`window`** specifies the number of events to use; default=**`0`** (all events)
  - **`current`** includes the current event in summary calculations
    - Default behavior; **`current=true`**
    - If set to **`current=false`**, then the search uses the field value from previous result

Note

`streamstats` does not require the entire results set to produce statistics. It works in a "streaming" manner by processing each event as it arrives.

`streamstats` Command Example 1

```
index=network sourcetype=cisco_wsa_squid action!="TCP_REFRESH_HIT"
| streamstats count as recentAttempts by bcg_ip
```

A

bcg_ip	recentAttempts
10.1.10.98	1
10.3.10.180	1
10.1.10.172	1
10.1.10.211	1
10.1.10.98	2
10.1.10.35	1
10.1.10.211	2
10.1.10.231	1
10.1.10.231	2
10.1.10.231	3

A The first time a **`bcg_ip`** is encountered, it is assigned a 1. The second time the same **`bcg_ip`** is encountered, it is assigned a 2, etc.

`streamstats` Command Example 2

Scenario

Sales wants to monitor a moving average of the price of a purchase on the Buttercup Games website over the previous 100 purchases during the last 4 hours.

```
index=web sourcetype=access_combined action=purchase status=200 productId=*
| table _time, price
| sort _time
| streamstats avg(price) as averageOrder window=100 current=f
```

A B

This search has completed and has returned **51** results

_time	price	averageOrder
2022-05-25 17:20:07	4.99	
2022-05-25 17:25:19	39.99	4.99
2022-05-25 17:26:05	24.99	22.490000000000002
2022-05-25 17:29:42	5.99	23.323333333333334
2022-05-25 17:35:20	24.99	18.99

B The current event is not included in summary calculations

A The **`averageOrder`** value will change for each event as more **`price`** values are included in the **`streamstats`** command calculation – up to 100 events

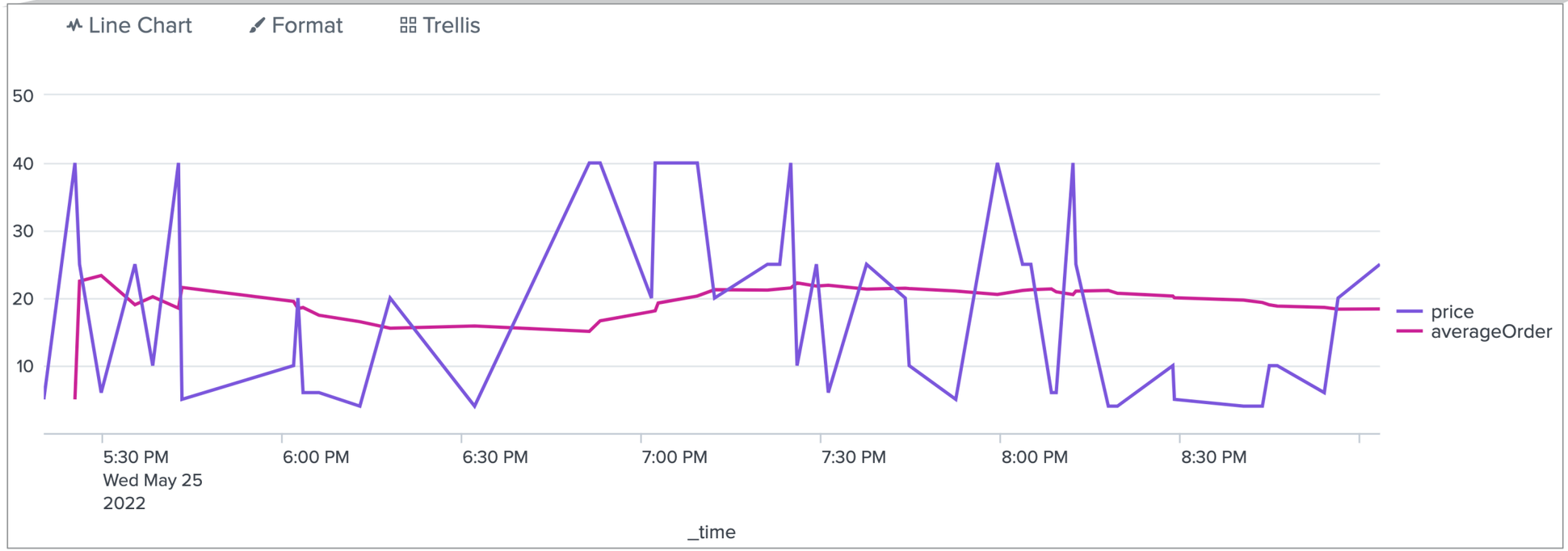
`streamstats` Command Example 2 (cont.)

Scenario

Sales wants to monitor a moving average of the price of a purchase on the Buttercup Games website over the previous 100 purchases during the last 4 hours.

```
index=web sourcetype=access_combined action=purchase status=200 productId=*
| table _time, price
| sort _time
| streamstats avg(price) as averageOrder window=100 current=f
```

A B



Modifying Result Sets Lab Exercise

Time: 20 minutes

Tasks:

- Modify a search with the **`appendpipe`** command to generate category subtotals for a sales report
- Complete a search that will identify retail products with lower-than-average sales with the **`stats`**, **`eventstats`**, and **`where`** commands
- Use the **`streamstats`** command to rank employees by their non-business network activity and find the 3 most active users

Modifying Field Values

Topic Objectives

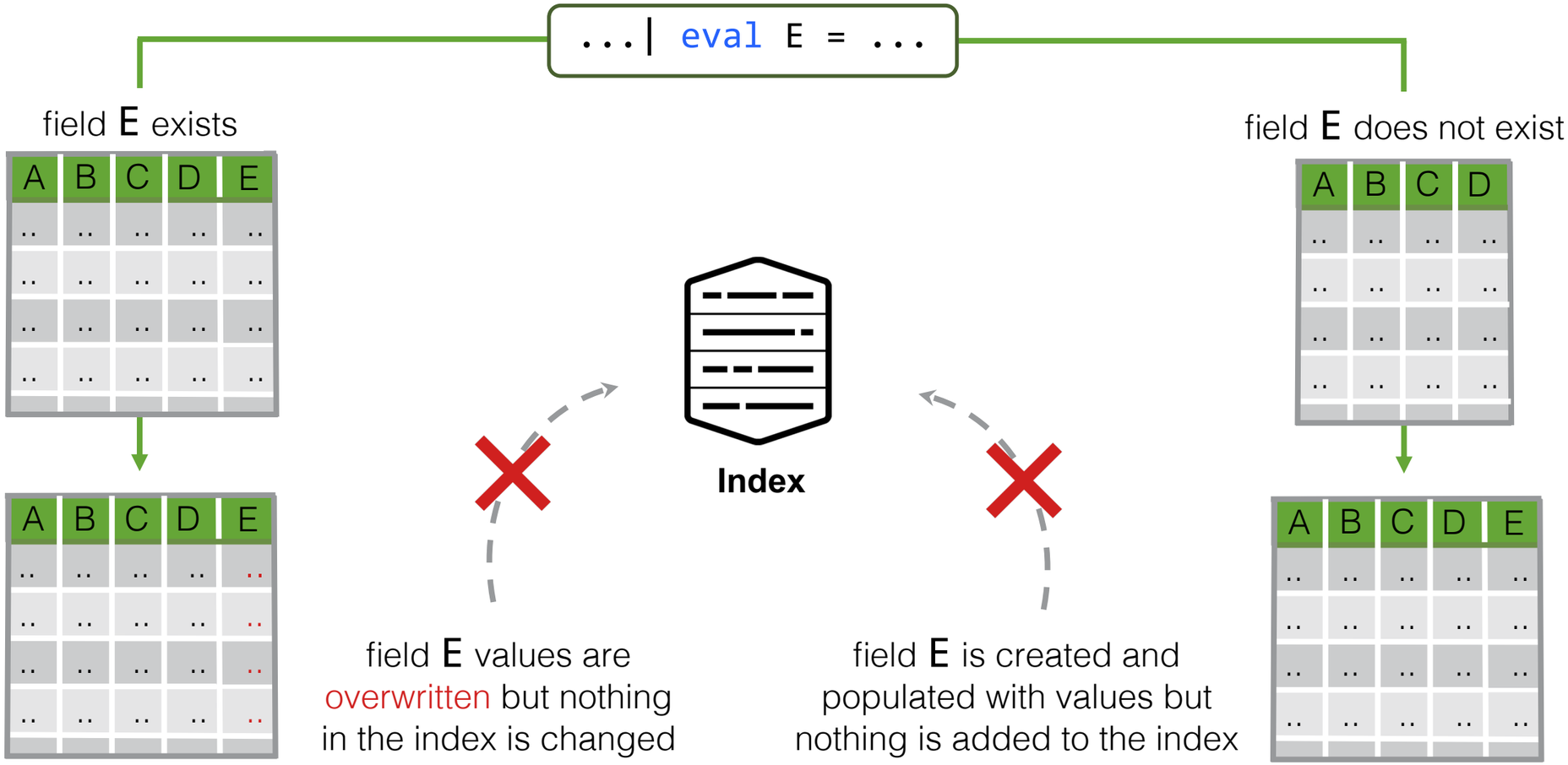
- Explore the **`eval`** command
- Identify evaluation functions for modifying field values:
  - `tostring`
  - `tonumber`
  - `lower`
  - `upper`
- Reformat fields with **`foreach`**

`eval` Command

```
...| eval <field1>=<expression1>[, <field2>=<expression2>]
```

- Calculates an expression and puts the resulting value into a new or existing field which can be reused in the search pipeline
- Extremely powerful and supports a vast assortment of functions for performing specific tasks
- Can exist as an expression

`eval` Command (cont.)



`eval` Command (cont.)

The `eval` command supports various operators

Type	Operators
arithmetic	`+ - * / %`
concatenation	`+ .`
Boolean	`AND OR NOT XOR`
comparison	`< > <= >= != = LIKE`

Note

This topic focuses on using concatenation operators with certain `eval` functions.

eval Command Syntax

```
index=sales sourcetype=vendor_sales VendorCountry="United States"
| stats sum(price) as Sales by VendorStateProvince
| eval Performance = case(Sales<=500,"Needs immediate evaluation",
                         Sales<1000,"Underperformer",Sales>=1000,"Overperformer")
| eval Verdict = if(Performance IN("Underperformer","Needs immediate evaluation"), "Send to marketing",null())
| eval Sales = "$".tostring(Sales,"commas")
```

- Field values are treated in a case-sensitive manner
- String values must be "double-quoted"
- Field names must be unquoted or single quoted when they include a special character like a space
- Use a period (.) instead of (+) when concatenating strings and numbers to avoid conflicts

Ways to Write Multiple `eval`s

Expressions can be separate, nested, or linked with a comma

Separate **eval** pipeline segments

```
index=network sourcetype=cisco_wsa_squid
| stats sum(sc_bytes) as bytes by usage
| eval bandwidth = bytes/(1024*1024)
| eval bandwidth = round(bandwidth, 2)
```

Nested **eval** commands targeting the same field

```
index=network sourcetype=cisco_wsa_squid
| stats sum(sc_bytes) as bytes by usage
| eval bandwidth = round(bytes/(1024*1024), 2)
```

Combining **eval** commands with commas

```
index=network sourcetype=cisco_wsa_squid
| stats sum(sc_bytes) as bytes by usage
| eval bandwidth = bytes/(1024*1024),
  bandwidth = round(bandwidth, 2)
```

usage	bytes	bandwidth
Borderline	1298542	1.24
Business	2909449	2.77
Personal	9771346	9.32
Unknown	997092	0.95
Violation	495606	0.47

Note

round is a mathematical function that rounds field values up to a specified decimal place.

Referencing `eval` Fields

Temporary fields created using the **`eval`** command can be referenced in the search pipeline by succeeding commands

```
index=network sourcetype=cisco_wsa_squid
| stats sum(sc_bytes) as Bytes by usage
| eval bandwidth = round(Bytes/pow(1024,2), 2)
| sort -bandwidth
| rename bandwidth AS "Bandwidth (MB)"
```

usage	Bytes	Bandwidth (MB)
Personal	299584913	285.71
Unknown	77187989	73.61
Business	66844576	63.75
Borderline	54011022	51.51
Violation	3203231	3.05

Note

`pow(X,Y)` is a mathematical function that returns the `X` to the power of `Y`.

Evaluation Functions

- Evaluates an expression based on your events and returns a result
- There are 11 categories of evaluation functions:
  - Conversion
  - Comparison and Conditional
  - Cryptographic
  - Informational
  - Statistical
  - Text
  - etc.

Evaluation Functions (cont.)

Category	Function Syntax	Description
Conversion	`tostring(X[,Y])`	Converts the input (numeric or Boolean) to a string
	`tonumber(NUMSTR[,BASE])`	Converts the input string `NUMSTR` to a number; `NUMSTR` can be a field name or a value
Text	`lower(X)`	Converts the string `X` to lowercase
	`upper(X)`	Converts the string `X` to uppercase
	`substr(X,Y,Z)`	Returns a substring of `X`, starting at the index specified by `Y` with the number of characters specified by `Z`
Comparison & Conditional	`coalesce(X,…)`	Takes an arbitrary number of arguments (`X`) and returns the first value that is not NULL

Conversion Functions: `tostring`

```
...| eval <field> = tostring(X,Y)
```

- Specify a numeric field for **X**
- Determine formatting of **X** by defining **Y**
  - **"commas"** formats **X** with commas
  - **"hex"** converts **X** to hexadecimal
  - **"duration"** converts second **X** to HH:MM:SS
- String values cannot be mathematically manipulated

```
index=web sourcetype=access_combined action=purchase status=503
| stats sum(price) as lostRevenue
| eval string_lostRevenue = "$".tostring(lostRevenue)
| table lostRevenue, string_lostRevenue
```

lostRevenue	string_lostRevenue
591.67	$591.67

Conversion Functions: `tostring` Example

Scenario

Identify the five longest client sessions over the last 4 hours in HH:MM:SS format.

```
...
| eval duration = tostring(sessionTime,"duration")
```

JSESSIONID	sessionTime
SD3SL4FF4ADFF4956	122
SD8SL9FF3ADFF4959	113
SD6SL7FF7ADFF198674	80
SD0SL5FF7ADFF198393	78
SD6SL9FF9ADFF198321	63

JSESSIONID	sessionTime	duration
SD3SL4FF4ADFF4956	122	00:02:02
SD8SL9FF3ADFF4959	113	00:01:53
SD6SL7FF7ADFF198674	80	00:01:20
SD0SL5FF7ADFF198393	78	00:01:18
SD6SL9FF9ADFF198321	63	00:01:03

Conversion Functions: `tonumber`

```
...| eval <field> = tonumber(NUMSTR[,BASE])
```

- Specify a field name or literal string value with **NUMSTR**
- Define the base of **NUMSTR** with **BASE** (optional)
  - Can be 2 – 36
  - Defaults to 10

Conversion Functions: `tonumber` Examples

Task

Convert string values for the field `store_sales` to numeric.

```
...| eval myValue="1.4848974e+12"
   | eval myValueAsInteger = tonumber(myValue)
```

myValue	myValueAsInteger
1.4848974e+12	1484897400000.0000000

Task

Convert octal number (base-8) 244 and the hexadecimal number (base-16) A4 to decimal

```
...| eval n1 = tonumber("244",8), n2 = tonumber("A4",16)
```

n1	n2
164	164

Text Functions Example

Convert values of a field (**X**) to uppercase or lowercase with the **upper(X)** and **lower(X)** functions

```
...
| eval uppercase_product = upper(product_name),lowercase_categoryId = lower(categoryId)
| table product_name, categoryId, itemId, uppercase_product, lowercase_categoryId
```

A

product_name	categoryId	itemId	uppercase_product	lowercase_categoryId
Holy Blade of Gouda	ACCESSORIES	EST-11	HOLY BLADE OF GOUDA	accessories
Holy Blade of Gouda	ACCESSORIES	EST-16	HOLY BLADE OF GOUDA	accessories
World of Cheese Tee	TEE	EST-19	WORLD OF CHEESE TEE	tee
Dream Crusher	STRATEGY	EST-18	DREAM CRUSHER	strategy
Orvil the Wolverine	ARCADE	EST-6	ORVIL THE WOLVERINE	arcade
Final Sequel	STRATEGY	EST-7	FINAL SEQUEL	strategy
SIM Cubicle	SIMULATION	EST-16	SIM CUBICLE	simulation

`foreach` Command

```
...| foreach <wc-field-list>
     [template-subsearch]
```

- Applies a template to multiple fields
- The **`template-subsearch`** is a subsearch that contains formatting instructions that will be applied to **`<wc-field-list>`**
- Use the **`<<FIELD>>`** token as a field placeholder in the subsearch

Note

The **`wc`** in **`<wc-field-list>`** means that **`foreach`** supports wildcarded field names.

foreach Command Example 1

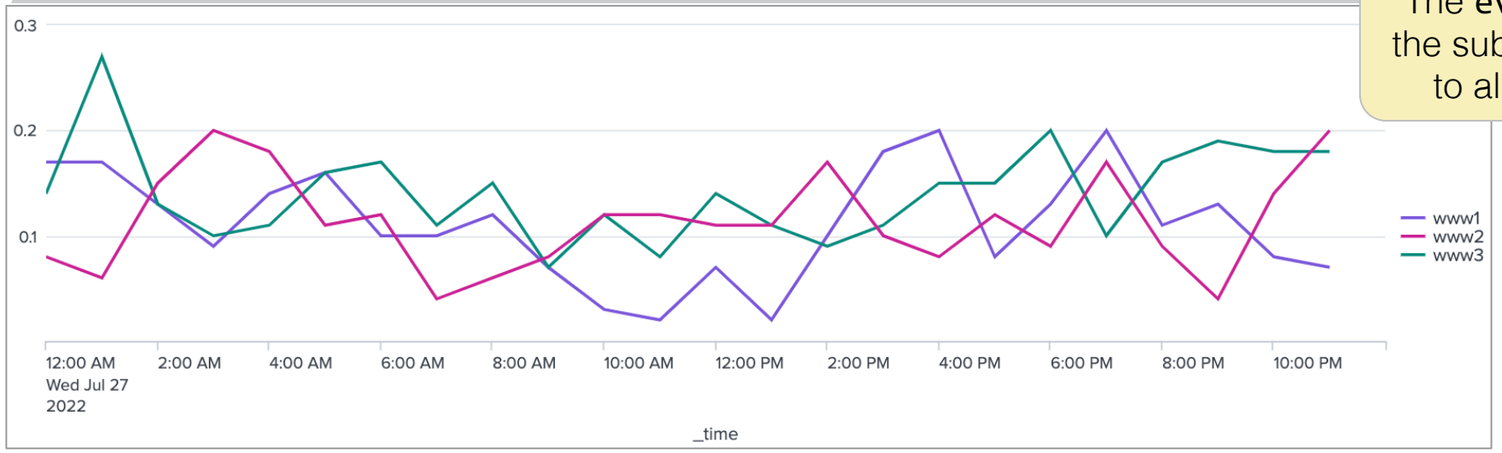
A
```
...
```
B
```
| foreach www*
  [eval <<FIELD>> = round(<<FIELD>>/(1024*1024),2)]
```

A

_time	www1	www2	www3
2022-07-27 00:00	179827	83806	146440
2022-07-27 01:00	176430	59010	282238
2022-07-27 02:00	135457	157254	131994

B

_time	www1	www2	www3
2022-07-27 00:00	0.17	0.08	0.14
2022-07-27 01:00	0.17	0.06	0.27
2022-07-27 02:00	0.13	0.15	0.13



The **eval** expression in the subsearch is applied to all fields with **www**

foreach Command Example 2

foreach with a wildcard * will apply a template to all field values

```
...
| foreach *
    [eval <<FIELD>>=upper(<<FIELD>>)]
```

domain	revenue	products
bing	$567.75	Benign Space Debris Curling 2014 Dream Crusher Final Sequel Fire Resistance Suit of Provolone Holy Blade of Gouda Manganiello Bros. Manganiello Bros. Tee Orvil the Wolverine SIM Cubicle World of Cheese World of Cheese Tee
buttercupgames	$70,338.62	Benign Space Debris Curling 2014 Dream Crusher Final Sequel Fire Resistance Suit of Provolone

domain	revenue	products
BING	$567.75	BENIGN SPACE DEBRIS CURLING 2014 DREAM CRUSHER FINAL SEQUEL FIRE RESISTANCE SUIT OF PROVOLONE HOLY BLADE OF GOUDA MANGANIELLO BROS. MANGANIELLO BROS. TEE ORVIL THE WOLVERINE SIM CUBICLE WORLD OF CHEESE WORLD OF CHEESE TEE
BUTTERCUPGAMES	$70,338.62	BENIGN SPACE DEBRIS CURLING 2014 DREAM CRUSHER FINAL SEQUEL FIRE RESISTANCE SUIT OF PROVOLONE

Modifying Field Values Lab Exercise

Time: 5 minutes

Tasks:

- Use `foreach` to convert multiple numeric fields to currency format

Normalizing with `eval`

Topic Objectives

- Normalize data using evaluation functions:
  - `substr`
  - `coalesce`

Why Normalize?

- Data representing the same information can be named differently
- Normalization is the process of organizing data to appear similar across all records, making the information easier to search
- Two types of normalization are discussed in this topic:
  - Data Normalization
  - Field Normalization
- This topic discusses the various **`eval`** commands that assist with normalizing data

substr Function

```
...| eval <field> = substr(X,Y[,Z])
```

- Returns a substring of **X** starting at **Y** with the number of characters specified by **Z**
- **X** is a literal string (e.g. “abcd” or “1234”) or an existing field
- **Y** is numeric and specifies where the substring begins
  - If positive, substring starts at **Y** characters from the beginning
  - If negative, Splunk starts at **Y** characters from the end
- **Z** (optional) is numeric and specifies the number of characters to return if **Y** is positive; if not specified, returns the rest of the string

substr Function Example 1

Scenario

Return the employee username from workstations where the names are BG##-username.

```
...| eval employee = substr(bcg_workstation,6)
```

bcg_workstation	employee
BG01-ptoscani	ptoscani
BG01-rroberts	rroberts
BG03-gfacello	gfacello

substr Function Example 2

Create information-dense field values by using **substr** with other text functions and two or more fields

```
...
| eval ItemCode = tostring(substr(categoryId, 1, 3)."-".upper(product_name)."-".substr(itemId,5))
| table product_name, categoryId, itemId, ItemCode
```

A B C

product_name	categoryId	itemId	ItemCode
Curling 2014	SPORTS	EST-6	SPO-CURLING 2014-6
Curling 2014	SPORTS	EST-15	SPO-CURLING 2014-15
Manganiello Bros.	ARCADE	EST-17	ARC-MANGANIELLO BROS.-17
Manganiello Bros. Tee	TEE	EST-17	TEE-MANGANIELLO BROS. TEE-17
Benign Space Debris	ARCADE	EST-26	ARC-BENIGN SPACE DEBRIS-26

`coalesce` Function

```
...| eval field1 = coalesce(X1,X2,...)
```

- Returns the first non-null values from the provided fields `X1, X2,…`
- Great for normalizing fields from the results set where two or more field names represent the same data

Field Normalization with `coalesce`

Normalize fields with the same values but different names

Scenario

Normalize IPs from the online sales and web security appliance data servers over the last 5 minutes.

```
(index=network sourcetype=cisco_wsa_squid) OR
(index=web sourcetype=access_combined)
| eval oneIP = coalesce(clientip,c_ip)
| table c_ip, clientip, oneIP, sourcetype
```

c_ip	clientip	oneIP	sourcetype
202.91.242.117		202.91.242.117	cisco_wsa_squid
	178.19.3.199	178.19.3.199	access_combined
194.215.205.19		194.215.205.19	cisco_wsa_squid
27.101.11.11		27.101.11.11	cisco_wsa_squid
207.36.232.245		207.36.232.245	cisco_wsa_squid
	203.45.206.135	203.45.206.135	access_combined

`eval` assigns the first value from `clientip` or `c_ip` to `oneIP`

178.19.3.199	178.19.3.199	access_combined
194.215.205.19	194.215.205.19	cisco_wsa_squid

Normalizing with `eval` Lab Exercise

Time: 15 minutes

Tasks:

- Use the `eval` command to normalize username data from two different sourcetypes and find the 5 most-active employees on the network
- Use various `eval` functions to find an individual who gained access to a Buttercup Games office without legitimate credentials
- Challenge: Troubleshoot and modify a search for a sales report

Wrap-up Slides

Wrap-up

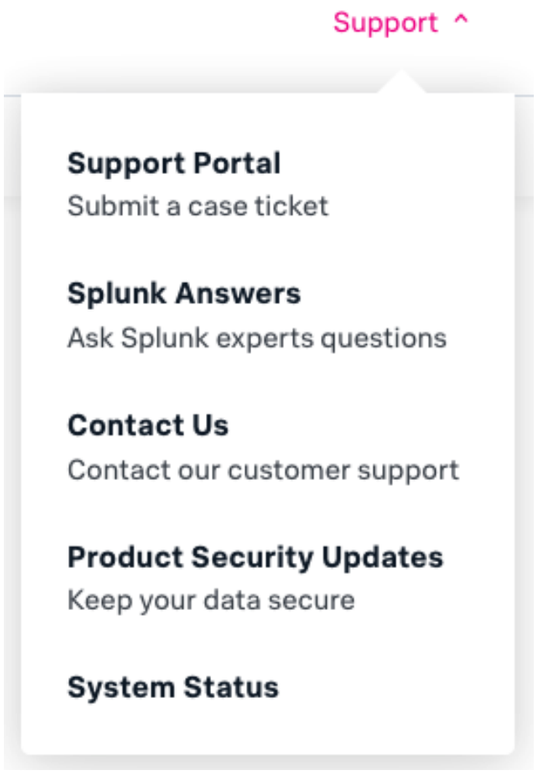
- You are now able to:
  - Convert stats-like output into chartable data and vice versa
  - Use a **`foreach`** subsearch to quickly reformat fields and field values
  - Modify field values and normalize data with **`eval`**
  - Generate summary statistics that can be used to filter data
  - Use subpipelines to append data to results

Community

- Splunk Community Portal
  community.splunk.com
  - Answers
  - Discussions
  - Splunk Trust
  - User Groups
  - Ideas
- Splunk Blogs
  splunk.com/blog/
- Splunk Apps
  splunkbase.com
- Splunk Dev Google Group
  groups.google.com/forum/#!forum/splunkdev
- Splunk Docs on Twitter
  twitter.com/splunkdocs
- Splunk Dev on Twitter
  twitter.com/splunkdev
- Splunk Live!
  splunklive.splunk.com
- .conf
  conf.splunk.com

Support Programs

- **Web**
  - Documentation: dev.splunk.com and docs.splunk.com
- **Splunk Lantern**
  Guidance from Splunk experts
  - lantern.splunk.com
- **Global Support**
  Support for critical issues, a dedicated resource to manage your account – 24 x 7 x 365
  - Web: splunk.com/index.php/submit_issue
- **Enterprise, Cloud, ITSI, Security Support**
  - Web: splunk.com/en_us/about-splunk/contact-us.html#tabs/customersupport
  - Phone: (855) SPLUNK-S or (855) 775-8657



Learning Paths

Search Expert - Recommended Courses

Free eLearning courses are in blue and courses with an * are present in both learning paths.

- What is Splunk *
- Introduction to Splunk *
- Using Fields *
- Scheduling Reports and Alerts
- Visualizations
- Statistical Processing
- Working with Time
- Comparing Values
- Result Modification
- Leveraging Lookups and Subsearches
- Correlation Analysis
- Search Under the Hood
- Multivalue Fields
- Search Optimization *

Learning Paths (cont.)

Knowledge Manager - Recommended Courses

Free eLearning courses are in blue and courses with an * are present in both learning paths.

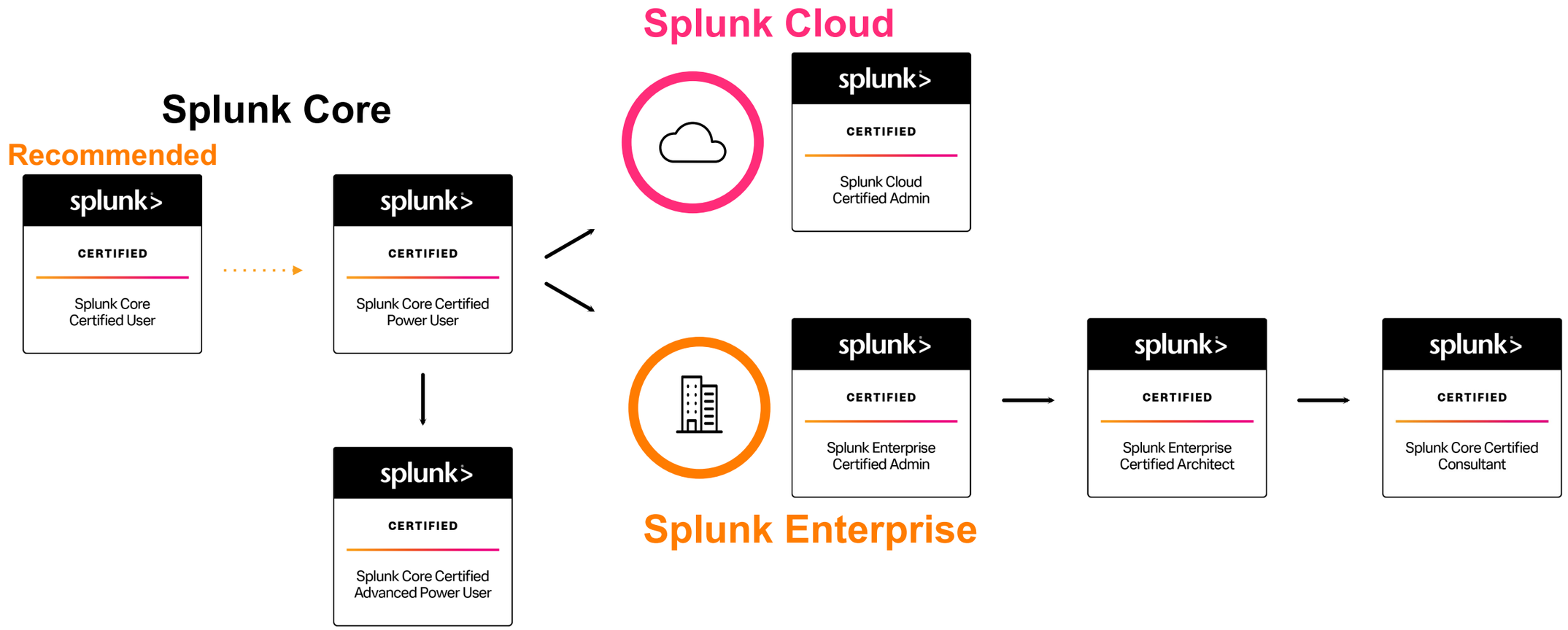
- What is Splunk *
- Introduction to Splunk *
- Using Fields *
- Introduction to Knowledge Objects
- Creating Knowledge Objects
- Creating Field Extractions
- Enriching Data with Lookups
- Data Models
- Introduction to Dashboards
- Dynamic Dashboards
- Using Choropleth
- Search Optimization *

Splunk Certification

Offerings & Requirements

Splunk Core and Beyond

Regardless of which Splunk product you use, it all starts with Splunk Core



App-Specific Offerings

For Splunk Add-Ons

App Developer

ES Administration

ITSI Administration

SOAR Automation Developer

Splunk Core Certified User

This entry-level certification demonstrates an individual's basic ability to navigate and use Splunk software

Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk Core Certified User Exam

Time to study! We suggest candidates looking to prepare for this exam complete Fundamentals 1 **or** the following courses:

- What is Splunk?
- Intro to Splunk
- Using Fields
- Scheduling Reports and Alerts
- Visualizations
- Statistical Processing
- Working with Time
- Leveraging Lookups and Subsearches
- Search Optimization
- Enriching Data with Lookups
- Data Models

See here for registration assistance.

Congratulations! You are a...

Recommended Next Step

- Splunk Core Certified Power User

Splunk Core Certified Power User

This entry-level certification demonstrates an individual's foundational competence of Splunk's core software

Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk Core Certified Power User Exam

Time to study! We suggest candidates looking to prepare for this exam complete Fundamentals 2 **or** the following courses:

- Visualizations
- Statistical Processing
- Working with Time
- Comparing Values
- Result Modification
- Correlation Analysis
- Search Under the Hood
- Introduction to Knowledge Objects
- Creating Knowledge Objects
- Creating Field Extractions
- Data Models
- Using Choropleth

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Splunk Core Certified Advanced Power User
- Splunk Enterprise Certified Admin
- Splunk Cloud Certified Admin

 | 22 September 2022

Splunk Core Certified Advanced Power User

This certification demonstrates an individual's ability to generate complex searches, reports, and dashboards with Splunk's core software to get the most out of their data

Prerequisite Certification(s):

- Splunk Core Certified Power User

Prerequisite Course(s):

- None

Splunk Core Certified Advanced Power User Exam

Time to study! We suggest candidates looking to prepare for this exam complete Fundamentals 3, Creating Dashboards, and Advanced Searching & Reporting **or** the following courses:

- Using Fields
- Working with Time
- Comparing Values
- Result Modification
- Leveraging Lookups and Subsearches
- Correlation Analysis
- Search Under the Hood
- Multivalue Fields
- Search Optimization
- Creating Field Extractions
- Enriching Data with Lookups
- Data Models
- Using Choropleth
- Introduction to Dashboards
- Dynamic Dashboards

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Splunk Enterprise Certified Admin
- Splunk Cloud Certified Admin

Splunk Cloud Certified Admin

This certification demonstrates an individual's ability to support the day-to-day administration and health of a Splunk Cloud environment

Prerequisite Certification(s):

- Splunk Core Certified Power User

Prerequisite Course(s):

- None

Splunk Cloud Certified Admin Exam

Time to study! We suggest candidates looking to prepare for this exam complete **either** the Splunk Cloud Administration **or** the Transitioning to Splunk Cloud course.

Both courses will equally prepare candidates for the exam, but are tailored to meet the needs of the individual based on prior Splunk experience.

Splunk Cloud Administration is designed for net-new administrators working in a Splunk Cloud environment. **Transitioning to Splunk Cloud** is for experienced Enterprise administrators looking to maximize their success in migrating to a Cloud environment.

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Splunk Certified Developer

Splunk Enterprise Certified Admin

This certification demonstrates an individual's ability to support the day-to-day administration and health of a Splunk Enterprise environment

Prerequisite Certification(s):

- Splunk Core Certified Power User

Prerequisite Course(s):

- None

Splunk Enterprise Certified Admin Exam

Time to study! We suggest candidates looking to prepare for this exam complete the following courses:

- Splunk System Administration
- Splunk Data Administration

See here for registration assistance.

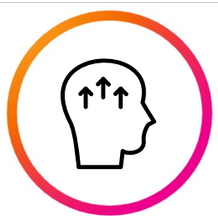
Congratulations! You are a...

Recommended Next Steps

- Splunk Enterprise Certified Architect
- Splunk Certified Developer

Splunk Certified Architect

This certification demonstrates an individual's ability to deploy, manage, and troubleshoot complex Splunk Enterprise environments

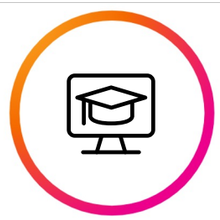


Prerequisite Certification(s):

- Splunk Core Certified Power User
- Splunk Enterprise Certified Admin

Prerequisite Course(s):

- Architecting Splunk Enterprise Deployments
- Troubleshooting Splunk Enterprise
- Splunk Cluster Administration
- Splunk Deployment Practical Lab



Splunk Enterprise Certified Architect Exam

Time to study! We **require** candidates looking to register for this exam to complete the following prerequisite courses:

- Architecting Splunk Enterprise Deployments
- Troubleshooting Splunk Enterprise
- Splunk Cluster Administration
- Splunk Deployment Practical Lab

Candidates who are **Splunk Enterprise Certified Admin** and have completed all of the above courses will automatically receive an exam authorization for the Splunk Enterprise Certified Architect exam within 5-7 business days of receiving their passing lab results.

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Splunk Core Certified Consultant

Splunk Core Certified Consultant

This certification demonstrates an individual's ability to properly size, install, and implement Splunk environments and to advise others on how to utilize the product and maximize its value for their needs

Prerequisite Certification(s):

- Splunk Core Certified Power User
- Splunk Enterprise Certified Admin
- Splunk Enterprise Certified Architect

Prerequisite Course(s):

- *Advanced Power User courses **or** digital badge**
- Core Consultant Labs
  - Indexer Cluster Implementation
  - Distributed Search Migration
  - Implementation Fundamentals
  - Architect Implementation 1-3
- Services Core Implementation

Splunk Core Certified Consultant Exam

Time to study! We **require** candidates looking to register for this exam to complete the following prerequisite courses:

- *Fundamentals 3*, *Creating Dashboards*, *Advanced Searching & Reporting**
- Core Consultant Labs
- Services Core Implementation

Candidates who are **Splunk Enterprise Certified Architects** and have completed all of the above courses must contact certification@splunk.com to request their Core Consultant exam authorization.

See here for registration assistance.

These Advanced Power User courses can be replaced with a Splunk Certified Advanced Power User badge **or completion of the following courses:*

- Using Fields
- Creating Field Extractions
- Enriching Data with Lookups
- Data Models
- Search Optimization
- Working with Time
- Leveraging Lookups and Subsearches
- Comparing Values
- Correlation Analysis
- Result Modification
- Multivalue Fields
- Search Under the Hood
- Introduction to Dashboards
- Dynamic Dashboards
- Using Choropleth

Congratulations! You are a...

Recommended Next Steps

- None

Splunk Certified Developer

This certification demonstrates an individual's expertise in drilldowns, advanced behaviors and visualizations, planning, creating, and packaging apps, and REST endpoints

Prerequisite Certification(s):

- Splunk Core Certified Power User

AND

- Splunk Enterprise Certified Admin

OR

- Splunk Cloud Certified Admin

Prerequisite Course(s):

- None

Splunk Certified Developer Exam

Time to study! We suggest candidates looking to prepare for this exam complete the following courses:

- Creating Dashboards with Splunk*
- Advanced Dashboards & Visualizations
- Building Splunk Apps
- Developing with Splunk's REST API

This course may also be substituted with the following newly-launched courses:

- Introduction to Dashboards
- Dynamic Dashboards
- Using Choropleth

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- None

Splunk Enterprise Security Certified Admin

This certification demonstrates an individual's ability to install, configure, and manage a Splunk Enterprise Security deployment

Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk Enterprise Security Certified Admin Exam

Time to study! We suggest candidates looking to prepare for this exam complete the following course:

- Administering Splunk Enterprise Security

Please note: all candidates are expected to have working knowledge and experience as either Splunk Cloud or Splunk Enterprise Administrators.

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Splunk Phantom Certified Admin

Splunk IT Service Intelligence Certified Admin

This certification demonstrates an individual's ability to deploy, manage, and utilize Splunk ITSI to monitor mission-critical services

Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk IT Service Intelligence Certified Admin Exam

Time to study! We suggest candidates looking to prepare for this exam complete the following course:

- Implementing Splunk IT Service Intelligence

Please note: all candidates are expected to have working knowledge and experience as either Splunk Cloud or Splunk Enterprise Administrators.

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Courses on Observability

Splunk SOAR Certified Automation Developer

This certification demonstrates an individual's ability to install and configure a SOAR server, integrate it with Splunk, and plan, design, create, and debug playbooks

Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk SOAR Certified Automation Developer Exam

Time to study! We suggest candidates looking to prepare for this exam complete the following courses:

- Administering SOAR (Phantom)
- Developing SOAR (Phantom) Playbooks
- Advanced SOAR (Phantom) Implementation

Please note: all candidates are expected to have working knowledge and experience as either Splunk Cloud or Splunk Enterprise Administrators.

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- None

Thank You