PER SCHOLAS

Lesson 200.5 Using Basic Transforming Commands

PER SCHOLAS

Learning Objectives

At the end of this lesson, learners will be able to:

- Describe Transforming Commands.
- Use the following Transforming Commands:
  - The top command.
  - The rare command.
  - The stats command.

PER SCHOLAS

Introduction

Splunk transforming commands provide powerful tools for manipulating and transforming data during the search process.

These commands allow users to extract, filter, calculate, aggregate, and reshape data in various ways to gain deeper insights and perform advanced analysis.

Users can refine search results, create meaningful visualizations, generate reports, and perform statistical calculations on the data.

Use transforming command to uncover patterns, trends, and anomalies that might otherwise go unnoticed.

In this lesson, we will take a look at three basic transforming commands you can use to tell a story.

image: Freepik.com

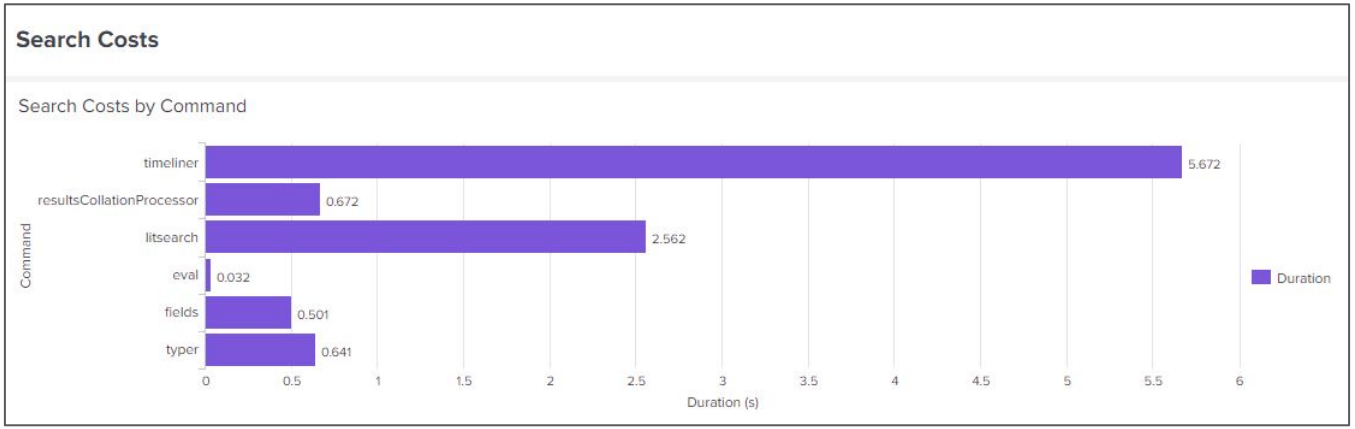
5.0 Transforming Commands

- A **transforming command** is a type of **search** command that **orders** the results into a **data table**.
- Transforming commands "**transform**" the specified cell values for each event into **numerical** values that Splunk Enterprise can use for **statistical** purposes.
- Searches that use transforming commands are called **transforming searches**.
- Transforming commands include **chart**, **timechart**, **stats**, **top**, **rare**, **contingency**, and **highlight**, some of which will be introduced in this lesson.

Source: https://docs.splunk.com/Splexicon:Transformingcommand

5.0 Transforming Commands (continued)

- To create chart **visualizations**, your search must **transform** event data into **statistical** data tables.
- **Transforming commands** are required to transform search result data into the data structures required for **visualizations** such as **column**, **bar**, **line**, **area**, and **pie** charts.



source: https://docs.splunk.com/Documentation/Splunk/9.0.4/Search/Aboutreportingcommands

5.1 The top Command

```
... | top [<int>] [<top-options>...] <field-list> [<by-clause>]
```

- The **top** command in Splunk finds the **most** common values for the fields in the field list.
- It calculates a **count** and a **percentage** of the **frequency** of the **values** that **occur in the events**.
- **field-list** is a **required** argument and is **comma-delimited**. Other arguments are **optional**.
- The optional **<int>** argument sets the number of results to return. The **default** is **10**.
- If the **<by-clause>** is included, the **results** are **grouped** by the **field** you **specify** in the **<by-clause>** option.

PER SCHOLAS

5.1 The top Command (continued)

The top command - example.

- By default, the **top** command will create a **table** under the **statistics** tab**,** containing the **top 10** results (top frequency occurrence in the events list) under the **<field-list>** column.
- When you use the top command, two **fields** are added to the results: **count** and **percent**.

New Search

Save As ▾ Create Table View Close

sourcetype=access_combined | top clientip

Previous business week ▾

✓ **18,673 events** (5/8/23 12:00:00.000 AM to 5/13/23 12:00:00.000 AM) Job ▾ Smart Mode

No Event Sampling ▾

Events Patterns **Statistics (10)** Visualization

20 Per Page ▾ Format Preview ▾

clientip	count	percent
87.194.216.51	395	2.115354
211.166.11.101	363	1.943983
128.241.220.82	283	1.515557
109.169.32.135	247	1.322765
188.138.40.166	217	1.162106
88.12.32.208	193	1.033578
194.215.205.19	186	0.996091
108.65.113.83	179	0.958603
192.188.106.240	173	0.926471
188.143.232.202	166	0.888984

image: screenshot, splunk Search & Reporting app

PER SCHOLAS

5.1 The top Command (continued)

The top command - example.

- The **top countfield=<string>** enables you to change the caption of the **count** column to a more meaningful name.
- **top percentfield=<string>** enables you to change the caption of the **percent** column.

New Search

Save As ▾ Create Table View Close

```
sourcetype=access_combined | top countfield="Hits" percentfield="% Rate" clientip
```

Previous business week ▾

✓ **18,673 events** (5/8/23 12:00:00.000 AM to 5/13/23 12:00:00.000 AM) Job ▾ Smart Mode

No Event Sampling ▾

Events Patterns **Statistics (10)** Visualization

100 Per Page ▾ Format Preview ▾

clientip	Hits	% Rate
87.194.216.51	395	2.115354
211.166.11.101	363	1.943983
128.241.220.82	283	1.515557
109.169.32.135	247	1.322765
188.138.40.166	217	1.162106
88.12.32.208	193	1.033578
194.215.205.19	186	0.996091
108.65.113.83	179	0.958603
192.188.106.240	173	0.926471
188.143.232.202	166	0.888984

image: screenshot, splunk Search & Reporting app

5.1 The top Command (continued)

The top command - example.

- The **top showcount=false** enables you to remove the **count** column from the table.
- **top showperc=false** enables you to remove the the **percent** column from the table.

These two options require a **boolean** (true/false) value.

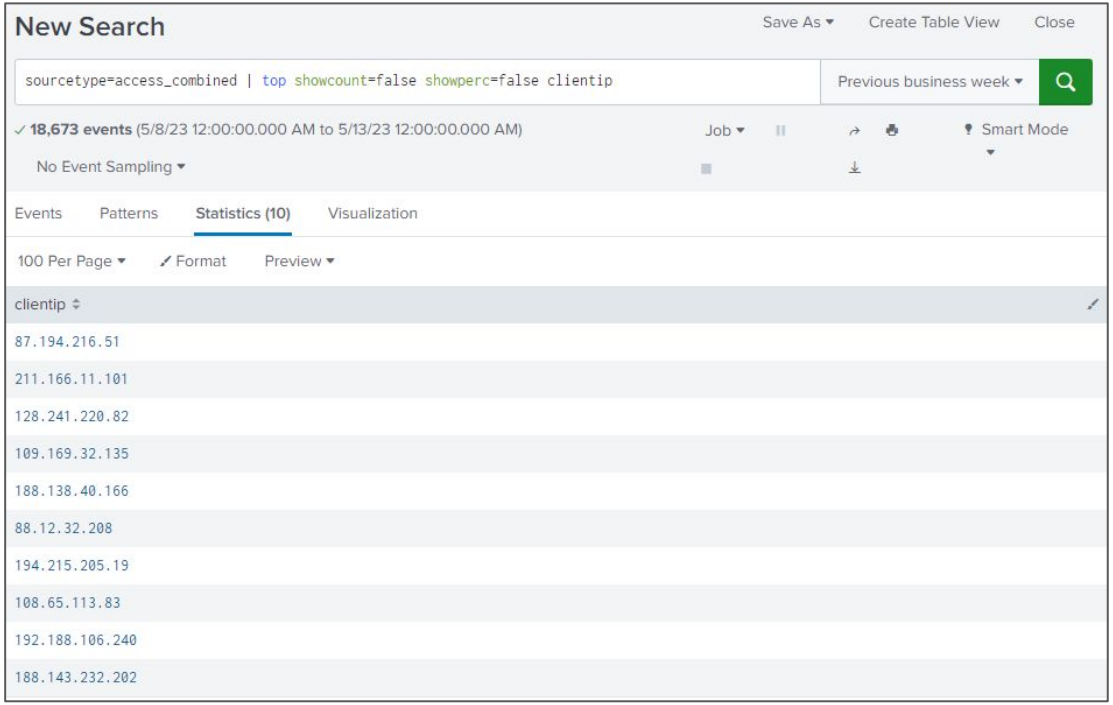


image: screenshot, splunk Search & Reporting app

PER SCHOLAS

5.1 The top Command (continued)

The top command - example.

- Use the **top limit=<int>** option to set the **number** of **values** in the table.
- You can also use **top <int> without** the limit keyword for the same result.
- <int> means an integer.
- **top limit=0** will return all values.

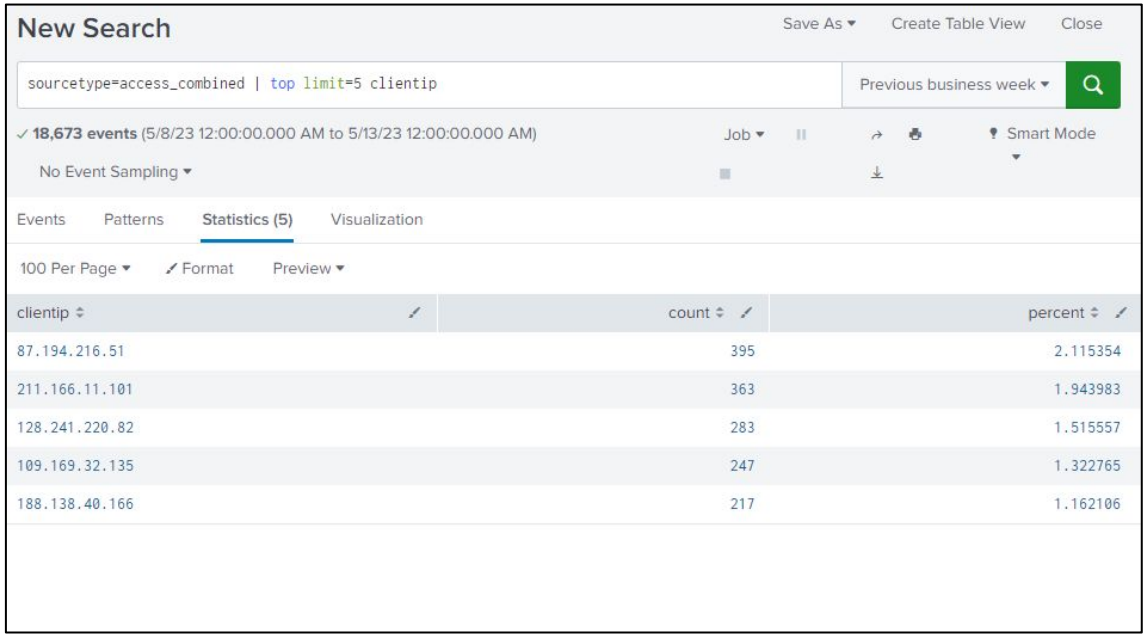


image: screenshot, splunk Search & Reporting app

5.1 The top Command (continued)

The top command - example.

- Use the **top useother=true** option to specify to add a **row** named **OTHER** that represents all **values not included** due to the **limit** cutoff.
- The default value is **false**.

New Search

Save As ▾ Create Table View Close

```
sourcetype=access_combined | top 7 useother=true clientip
```

Previous business week ▾

✓ **18,673 events** (5/8/23 12:00:00.000 AM to 5/13/23 12:00:00.000 AM) Job ▾ Smart Mode

No Event Sampling ▾

Events Patterns **Statistics (8)** Visualization

100 Per Page ▾ Format Preview ▾

clientip	count	percent
87.194.216.51	395	2.115354
211.166.11.101	363	1.943983
128.241.220.82	283	1.515557
109.169.32.135	247	1.322765
188.138.40.166	217	1.162106
88.12.32.208	193	1.033578
194.215.205.19	186	0.996091
OTHER	16789	89.910566

image: screenshot, splunk Search & Reporting app

PER SCHOLAS

5.1 The top Command (continued)

The top command - example.

- **top otherstr=<string>** will allow you to rename the **OTHER** row with a value that **better describes** the meaning of the numbers in the columns.

New Search

Save As ▾ Create Table View Close

```
sourcetype=access_combined | top 7 useother=true otherstr="Other IPs" clientip
```

Previous business week ▾

✓ **18,673 events** (5/8/23 12:00:00.000 AM to 5/13/23 12:00:00.000 AM) Job ▾ Smart Mode

No Event Sampling ▾

Events Patterns **Statistics (8)** Visualization

100 Per Page ▾ Format Preview ▾

clientip	count	percent
87.194.216.51	395	2.115354
211.166.11.101	363	1.943983
128.241.220.82	283	1.515557
109.169.32.135	247	1.322765
188.138.40.166	217	1.162106
88.12.32.208	193	1.033578
194.215.205.19	186	0.996091
Other IPs	16789	89.910566

image: screenshot, splunk Search & Reporting app

PER SCHOLAS

5.1 The top Command (continued)

The top command - example.

- Use the **<by-clause>** to **organize** the results by a **specific** field.
- In this example, the **three most frequent** clientip addresses that are interacting with the web servers are **displayed** for **each host**.

New Search

Save As ▾ Create Table View Close

```
sourcetype=access_combined | top 3 clientip by host
```

Last 7 days ▾

✓ **8,203 events** (5/15/23 7:00:00.000 AM to 5/22/23 7:03:25.000 AM) Job ▾ Smart Mode

No Event Sampling ▾

Events Patterns **Statistics (9)** Visualization

100 Per Page ▾ Format Preview ▾

host	clientip	count	percent
www1	211.166.11.101	66	2.301255
www1	182.236.164.11	66	2.301255
www1	87.194.216.51	59	2.057183
www2	87.194.216.51	116	4.432556
www2	198.35.1.10	71	2.713030
www2	211.166.11.101	58	2.216278
www3	87.194.216.51	70	2.575423
www3	81.18.148.190	49	1.802796
www3	118.142.68.222	48	1.766004

image: screenshot, splunk Search & Reporting app

5.2 The rare Command.

```
.. | rare [<int>] [<rare-options>...] <field-list> [<by-clause>]
```

- The **rare** command in Splunk finds the **least** common values for the fields in the field list.
- This command operates **identically** to the **top** command, except that the **rare** command finds the **least frequent values** instead of the **most frequent values**.
- Like the top command, it calculates a **count** and a **percentage** of the **frequency** the **values occur in the events**.
- **field-list** is a **required** argument, and is **comma-delimited**. Other arguments are **optional**.
- The optional **<int>** argument sets the number of results to return. The **default** is **10**.
- If the **<by-clause>** is included, the **results** are **grouped** by the **field** you **specify** in the **<by-clause>** option.

PER SCHOLAS

5.2 The rare command.

The rare command - example.

- By default the **rare** command will create a **table** under the **statistics** tab**,** containing the **10** the **least frequent** results (least frequent occurrence in the events list) under the **<field-list>** column.
- When you use the rare command, two **fields** are added to the results: **count** and **percent**.

sourcetype=access_combined | rare clientip

Previous business week

✓ **18,673 events** (5/8/23 12:00:00.000 AM to 5/13/23 12:00:00.000 AM) Job Smart Mode

No Event Sampling

Events Patterns **Statistics (10)** Visualization

20 Per Page Format Preview

clientip	count	percent
142.233.200.21	20	0.107107
59.36.99.70	30	0.160660
121.9.245.177	32	0.171370
194.146.236.22	32	0.171370
91.217.178.210	34	0.182081
212.58.253.71	36	0.192792
223.205.219.67	42	0.224924
130.253.37.97	44	0.235634
203.45.206.135	44	0.235634
207.36.232.245	44	0.235634

image: screenshot, splunk Search & Reporting app

PER SCHOLAS

5.2 The rare command.

The rare command - example.

- The **rare countfield=<string>** enables you to change the caption of the **count** column to a more meaningful name.
- **rare percentfield=<string>** enables you to change the caption of the **percent** column.
- **Note**: you can type the options **before** or **after** the <field-list>

New Search

Save As ▾ Create Table View Close

```
sourcetype=access_combined | rare clientip countfield="Hits" percentfield="% Rate"
```

Previous business week ▾

✓ **18,673 events** (5/8/23 12:00:00.000 AM to 5/13/23 12:00:00.000 AM) Job ▾ Smart Mode

No Event Sampling ▾

Events Patterns **Statistics (10)** Visualization

20 Per Page ▾ Format Preview ▾

clientip	Hits	% Rate
142.233.200.21	20	0.107107
59.36.99.70	30	0.160660
121.9.245.177	32	0.171370
194.146.236.22	32	0.171370
91.217.178.210	34	0.182081
212.58.253.71	36	0.192792
223.205.219.67	42	0.224924
130.253.37.97	44	0.235634
203.45.206.135	44	0.235634
207.36.232.245	44	0.235634

image: screenshot, splunk Search & Reporting app

PER SCHOLAS

5.2 The rare command.

The rare command - example.

- The **rare showcount=false** enables you to remove the **count** column from the table.
- **rare showperc=false** enables you to remove the the **percent** column from the table.
- These two options require a **boolean** (true/false) value.

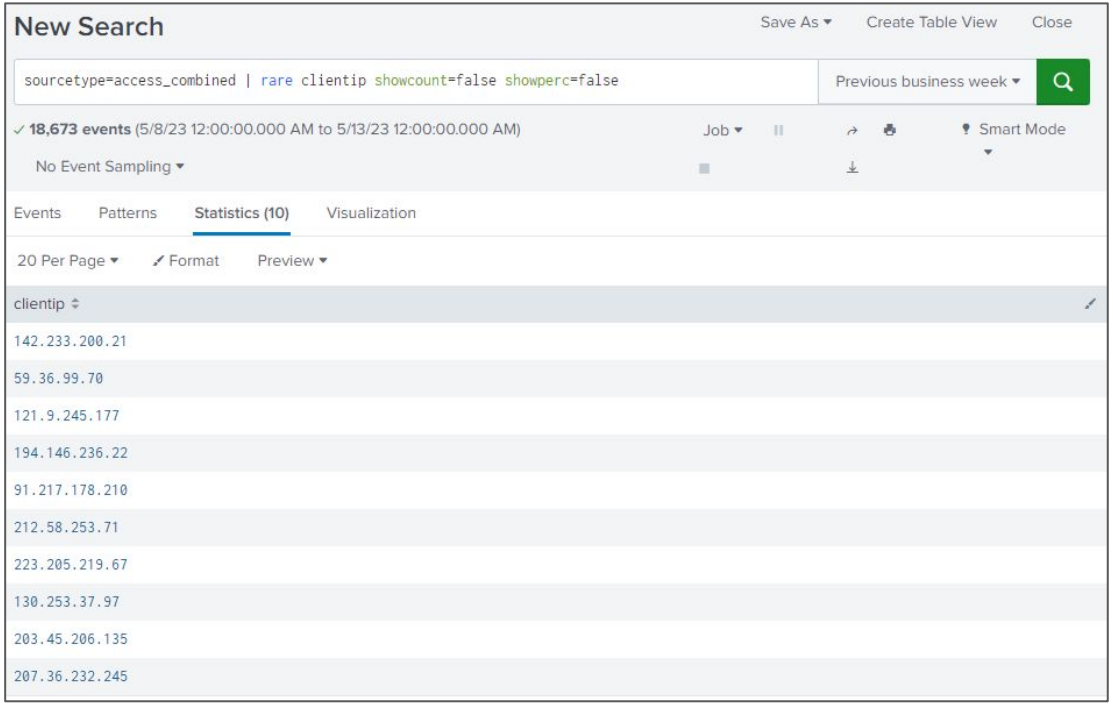


image: screenshot, splunk Search & Reporting app

PER SCHOLAS

5.2 The rare command.

The rare command - example.

- Use the **rare limit=<int>** option to set the **number** of **values** in the table.
- You can also use **rare <int> without** the limit keyword for the same result.
- <int> means an integer.
- **rare limit=0** will return all values.

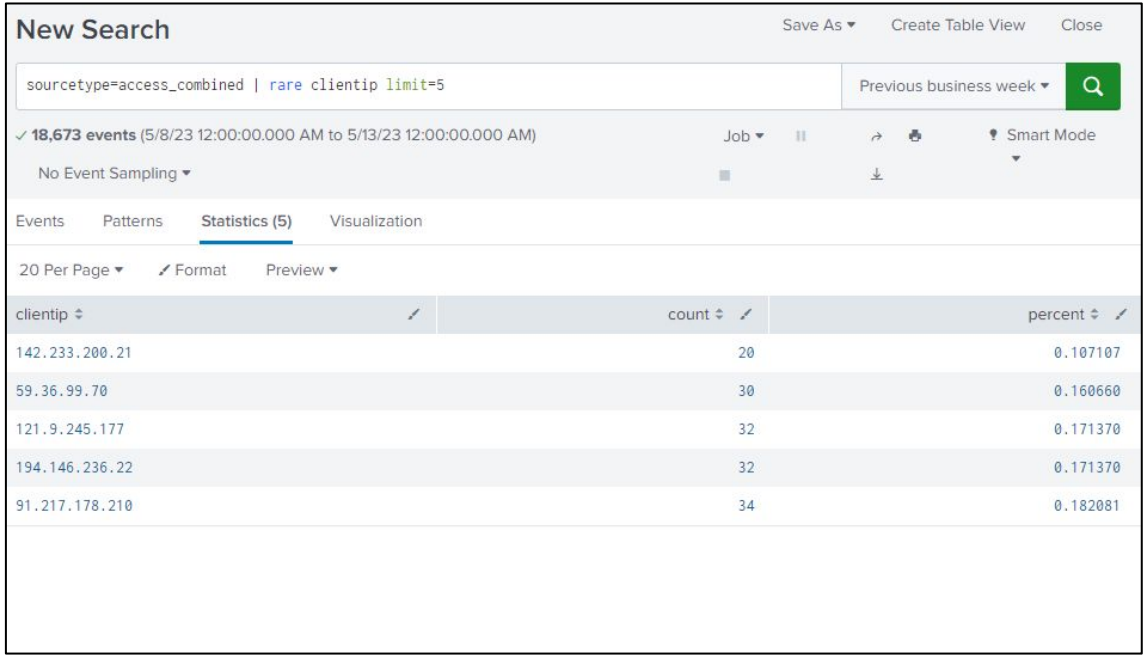


image: screenshot, splunk Search & Reporting app

PER SCHOLAS

5.2 The rare command.

The rare command - example.

- Use the **rare useother=true** option to specify to add a **row** named **OTHER** that represents all **values not included** due to the **limit** cutoff.
- The default value is **false**.

New Search

Save As ▾ Create Table View Close

```
sourcetype=access_combined | rare 7 clientip useother=true
```

Previous business week ▾

✓ **18,673 events** (5/8/23 12:00:00.000 AM to 5/13/23 12:00:00.000 AM) Job ▾ Smart Mode

No Event Sampling ▾

Events Patterns **Statistics (8)** Visualization

20 Per Page ▾ Format Preview ▾

clientip	count	percent
142.233.200.21	20	0.107107
59.36.99.70	30	0.160660
121.9.245.177	32	0.171370
194.146.236.22	32	0.171370
91.217.178.210	34	0.182081
212.58.253.71	36	0.192792
223.205.219.67	42	0.224924
OTHER	18447	98.789696

image: screenshot, splunk Search & Reporting app

PER SCHOLAS

5.2 The rare Command.

The rare command - example.

- **rare otherstr=<string>** will allow you to rename the **OTHER** row with a value that **better describes** the meaning of the numbers in the columns.

New Search

Save As ▾ Create Table View Close

```
sourcetype="access_combined_wcookie" | rare 7 clientip useother=true otherstr="Other IPs"
```

Previous business week ▾

✓ **18,673 events** (5/8/23 12:00:00.000 AM to 5/13/23 12:00:00.000 AM) Job ▾ Smart Mode

No Event Sampling ▾

Events Patterns **Statistics (8)** Visualization

20 Per Page ▾ Format Preview ▾

clientip	count	percent
142.233.200.21	20	0.107107
59.36.99.70	30	0.160660
121.9.245.177	32	0.171370
194.146.236.22	32	0.171370
91.217.178.210	34	0.182081
212.58.253.71	36	0.192792
223.205.219.67	42	0.224924
Other IPs	18447	98.789696

image: screenshot, splunk Search & Reporting app

PER SCHOLAS

5.2 The rare Command (continued)

The rare command - example.

- Use the **<by-clause>** to **organize** the results by a **specific** field.
- In this example, the **three least frequent** clientip addresses that are interacting with the web servers are **displayed** for **each host**.

New Search

Save As ▾ Create Table View Close

```
sourcetype=access_combined | rare 3 clientip by host
```

Last 7 days ▾

✓ **8,203 events** (5/15/23 7:00:00.000 AM to 5/22/23 7:22:20.000 AM) Job ▾ Smart Mode

No Event Sampling ▾

Events Patterns **Statistics (9)** Visualization

20 Per Page ▾ Format Preview ▾

host	clientip	count	percent
www1	121.254.179.199	1	0.034868
www1	64.120.15.156	1	0.034868
www1	123.118.73.155	2	0.069735
www2	12.130.60.5	1	0.038212
www2	178.19.3.199	1	0.038212
www2	84.34.159.23	1	0.038212
www3	91.208.184.24	1	0.036792
www3	58.68.236.98	2	0.073584
www3	74.125.19.106	2	0.073584

image: screenshot, splunk Search & Reporting app

5.3 The stats Command

```
...| stats <stats-function>(<wc-field>) [as <wc-field>] [by <field-list>]
```

- The **stats** command in Splunk is used to generate summary statistics or aggregations from search results.
- It allows you to **calculate metrics** such as **counts**, **averages**, **sums**, **minimum**, **maximum** values, and more (mathematical functions such as sum, and avg can only work with fields that contain **numerical values**, you can't sum host names).
- If the stats command is used **without a BY clause**, only **one row** is returned, which is the aggregation over the entire incoming result set. If a **BY clause is used**, one row is returned for **each distinct value** specified in the BY clause.
- **Note:** The syntax displayed here is a simple version. More options for stats are available.
- **Note 2:** The wc in <wc-field> means the stats command supports wildcard fields.

image: screenshot, splunk Search & Report app

5.3 The stats Command (continued)

Stats function options

- The **stats** command provides various **functions** that perform calculations and generate **summary statistics** on search results.
- **Syntax:** The syntax **depends** on the function. The **()** allow **arguments** to be passed to the functions
- The table lists the **supported** functions by **type** of function.

Type of function	Supported functions and syntax			

Aggregate functions	`avg()`
	`count()`
	`distinct_count()`
	`estdc()`
	`estdc_error()`
		`exactperc<num>()`
		`max()`
		`median()`
		`min()`
		`mode()`
			`perc<num>()`
			`range()`
			`stdev()`
			`stdevp()`
				`sum()`
				`sumsq()`
				`upperperc<num>()`
				`var()`
				`varp()`

Event order functions	`first()`	`last()`		
Multivalue stats and chart functions	`list()`	`values()`		
Time functions	`earliest()` `earliest_time()`	`latest()` `latest_time()`	`rate()`	

image: https://docs.splunk.com/Documentation/Splunk/9.0.4/SearchReference/Stats

PER SCHOLAS

5.3 The stats Command (continued)

The stats command - aggregate and Multivalue functions.

Function category	Function	Description
Aggregate	count	Returns the count of events
Aggregate	count (X)	Returns the number of events with a field value for the field X
Aggregate	dc (X)	Returns a count of unique values for X
Aggregate	distinct_count (X)	Returns a count of unique values for X (same as dc(x))
Aggregate	sum (X)	Returns a sum of numeric values for X
Aggregate	min (X)	Returns the minimum value of X
Aggregate	max (X)	Returns the maximum value of X
Aggregate	median (X)	Returns the middle-most value of X
Aggregate	range (X)	Returns the difference between the min and max values of X
Aggregate	stdev (X)	Returns the standard deviation of X
Aggregate	var (X)	Returns the variance of X
Multivalue	list (X)	Lists all vlaues of X
Multivalue	values (X)	Lists unique values of X

Note: This is not a full list of supported functions.

PER SCHOLAS

5.3 The stats Command (continued)

The stats command - instructor demonstration and examples.

Run the searches and note the different results.

- Return the **total** number of **events** in the search results

```
sourcetype=access_combined | stats count
```

Last 7 days

- Return the **number of events** that contain the **action** field (note the difference between the number of events and the number of table entries).

```
sourcetype=access_combined | stats count(action)
```

Last 7 days

- For more a **efficient** search, filter the event list **BEFORE** executing the stats command.

```
sourcetype=access_combined action=* | stats count
```

Last 7 days

image: screenshot, splunk Search & Reporting app

PER SCHOLAS

5.3 The stats Command (continued)

The stats command - instructor demonstration and examples.

In this example, we want to return the **number of events** where the value in the action field is **purchase**.

- Run the search (or refer to the image) and note the results.
- Explain **why** the returned number of events **match** the number of events under the count column.

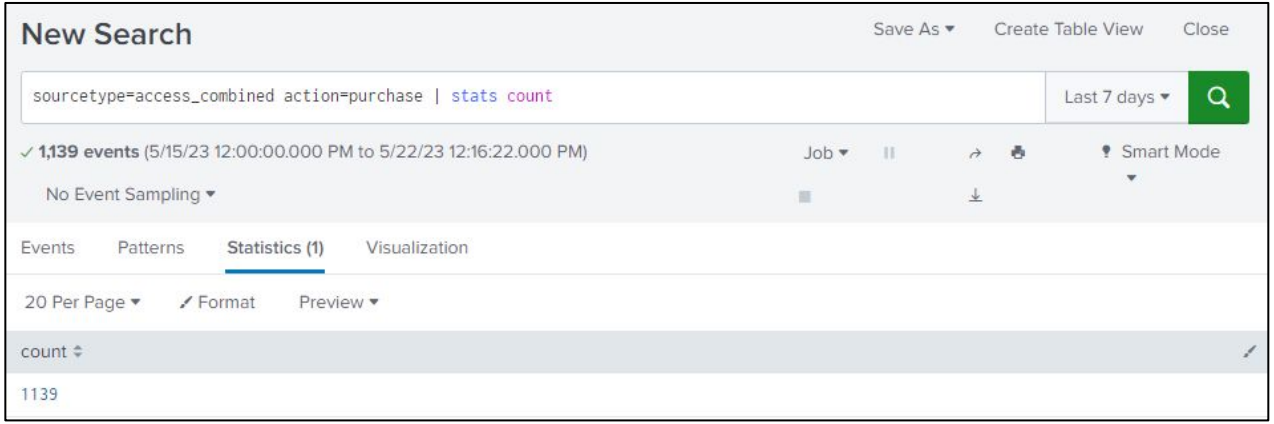


image: screenshot, splunk Search & Reporting app

5.3 The stats Command (continued)

The stats command - instructor demonstration and examples .

In this example, we want to return the **number of events** where the value in the action field is **purchase**, and **rename** the **count** column.

- Use the as clause to **rename** the count field.
- Run the search (or refer to the image) and note the results.

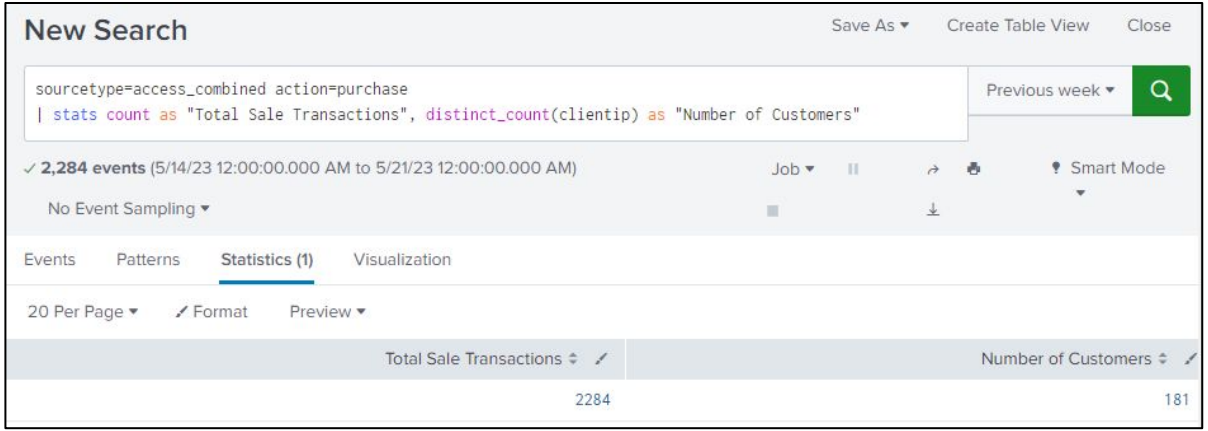
image: screenshot, splunk Search & Reporting app

5.3 The stats Command (continued)

The stats command - instructor demonstration and examples .

Run the searches and note the results.

- You can **call** on **more** stats functions by **delimiting** one function from another using a **comma**.
- In this example, the **distinct_count (clientip)** function will provide insight into the relationship between the number of sale transactions and the IP address of their origin.



distinct_count (or dc) returns only the unique values of clientip field.

While every event that contains the purchase value in the action field will also have a value for the clientip field, only 181 if these IP addresses are unique.

image: screenshot, splunk Search & Reporting app

PER SCHOLAS

5.3 The stats Command (continued)

The stats command - instructor demonstration and examples.

Run the searches and note the results.

- The **values** function with **clientip** as an **argument** will add a **column** displaying the **unique IP addresses** found in the returned **events**.

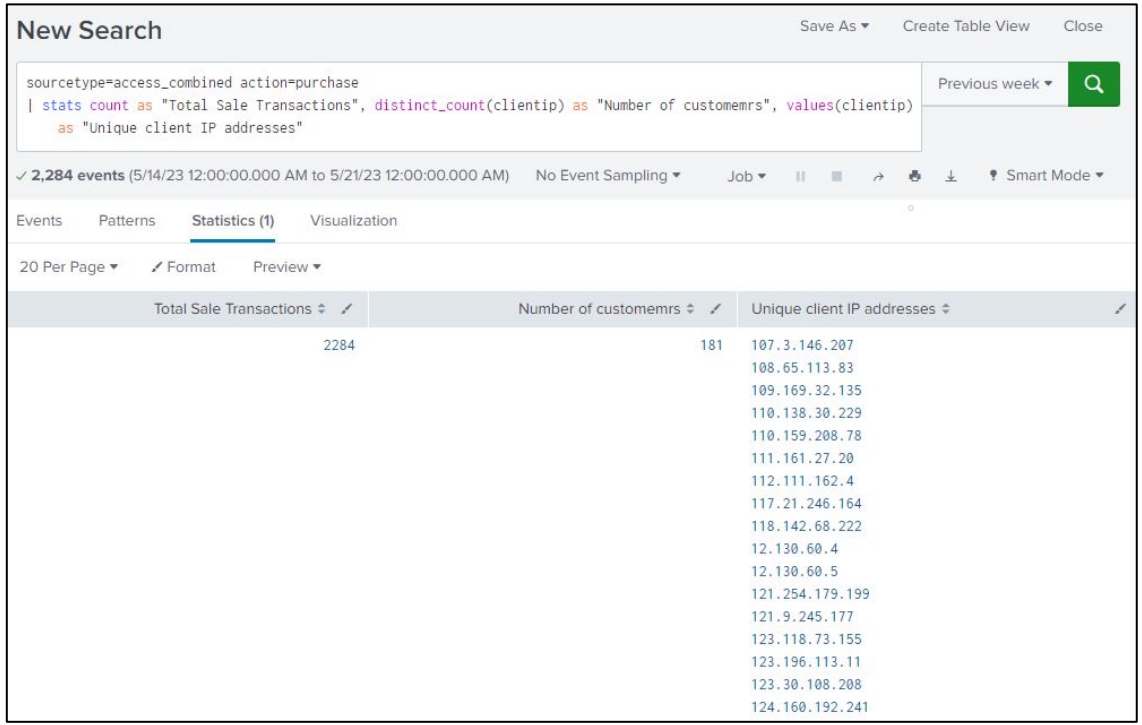


image: screenshot, splunk Search & Reporting app

PER SCHOLAS

5.3 The stats Command (continued)

The stats command - instructor demonstration and examples.

Use the **by** clause to **group** results by a named field or set of fields.

- In this example, **(invalid OR failed)** is searching for **failed login** attempts.
- The by clause groups the **number of failed attempts** by the target **host** and the **source IP** address

New Search

Save As ▾ Create Table View Close

```
index=security sourcetype=linux_secure (invalid OR failed) | stats count as "Potential Issues" by host, src_ip
```

Last 15 minutes ▾

✓ **68 events** (5/30/23 5:43:45.000 PM to 5/30/23 5:58:45.000 PM) No Event Sampling ▾ Job ▾ Verbose Mode ▾

Events (68) Patterns **Statistics (11)** Visualization

20 Per Page ▾ Format Preview ▾

host	src_ip	Potential Issues
mailsv1	10.3.10.46	9
mailsv1	209.160.24.63	6
www1	10.1.10.172	13
www1	107.3.146.207	3
www1	211.25.254.234	17
www1	217.15.20.146	6
www1	92.46.53.223	4
www2	125.89.78.6	2
www3	10.1.10.172	3
www3	10.3.10.46	1
www3	49.212.64.138	4

image: screenshot, splunk Search & Reporting app

PER SCHOLAS

5.3 The stats Command (continued)

The stats command - instructor demonstration and examples.

In this example, we use **stats** with **sum()**, **eval**, **rename**, and **sort** functions to calculate the **bandwidth** used by **employees** in relation to the **websites** they visited in the **last seven days**.

New Search

Save As ▾ Create Table View Close

```
index=network sourcetype=cisco_wsa_squid | eval "Employee name"=fname." ".lname | stats sum(sc_bytes) as Bandwidth
    by s_hostname, "Employee name" | sort - Bandwidth | rename s_hostname as "Websites visited"
```

Last 7 days ▾

✓ **909 events** (5/23/23 6:00:00.000 PM to 5/30/23 6:54:06.000 PM) No Event Sampling ▾ Job ▾ Verbose Mode ▾

Events (909) Patterns **Statistics (119)** Visualization

20 Per Page ▾ Format Preview ▾ < Prev 1 2 3 4 5 6 Next >

Websites visited	Employee name	Bandwidth
www.billboard.com	Gabriel Voronoff	1829816
www.lakeholiday.com	Guoxiang Nooteboom	1718544
www.kare11.com	Ester Williams	1202753
www.designfirstkitchens.com	Mizuhito Kemmerer	1182042
www.educationworld.com	Gabriel Voronoff	807297
www.portfolio.com	Gianpaolo Facello	594494
www.fastcompany.com	Guoxiang Nooteboom	486222
www.aiga.org	Kazuhide Peha	452002
www.macworld.com	Selwyn Howser	407113
www.panduit.com	Bradley Hussain	341759
www.switchfoot.com	Shawn Scallion	330997
www.gaylordhotels.com	Kathleen Percy	316268
www.d11.org	Gioia Bottazzi	287905
www.verizonwireless.com	Daniil Piazza	254545
www.destination360.com	Lien Teng	251423

image: screenshot, splunk Search & Reporting app

PER SCHOLAS

5.3 The stats Command (continued)

The stats command - instructor demonstration and examples.

In this example, we use **stats** with **min()**, **max()**, and **avg()** functions to display the **bandwidth** used by **internet usage type** (policy is configured on the firewall) in the **last seven days**.

Note the **round()** function used to trim the Average Bytes value down to **two decimal places**.

New Search

Save As ▾ Create Table View Close

```
index=* sourcetype=cisco_wsa_squid | stats min(sc_bytes) as "Minimum Bytes" max(sc_bytes) as "Maximum Bytes" avg
    (sc_bytes) as "Average Bytes" by usage | eval Average Bytes = round('Average Bytes',2)
```

Last 7 days ▾

✓ **867 events** (5/24/23 10:00:00.000 AM to 5/31/23 10:33:14.000 AM) No Event Sampling ▾ Job ▾ Smart Mode ▾

Events Patterns **Statistics (5)** Visualization

20 Per Page ▾ Format Preview ▾

usage	Minimum Bytes	Maximum Bytes	Average Bytes
Borderline	388	293067	11618.04
Business	344	1147756	22780.35
Personal	256	266499	13122.20
Unknown	240	1506333	21853.65
Violation	575	2680	2280.43

image: screenshot, splunk Search & Reporting app

5.0 - 5.3: top, rare, and stats - Summary

In this lesson, we looked at three of Splunk's transforming commands (many more exist).

These commands allow extracting, filtering, calculating, aggregating, and reshaping data in various ways to gain deeper insights and perform advanced analysis.

The top command, by default, finds the 10 most common values for the fields in the field list.

The rare command, by default, finds the 10 least common values for the fields in the field list.

The stats command is used to generate summary statistics or aggregations from search results. It supports many functions.

PER SCHOLAS

Knowledge check.

- What are transforming commands in Splunk?
- What is the top command used for?
- How many values will the top command return by default?
- What keyword is used to control the number of results returned by the rare command?
- What keyword is used to remove the percent column from the results table created by the rare command?
- What does the stats dc() function do?
- Why would you use the by clause with the stats command?
- What 2 letter keyword allows you to rename a field calculated by a stats function?