

Lab Report — Search Commands Deep Dive: Part 1

Name: Dhruvish

Date: October 24, 2025

Platform: Splunk Cloud Trial

Objective

The objective of this lab was to gain proficiency with the most frequently used **Splunk transforming search commands** — `count`, `values()`, `dc()`, and `avg()`.

You practiced grouping data using `by`, shaping result tables, and creating safe numeric fields for averages with `eval`.

These foundational skills are essential for data summarization, reporting, and dashboard development in any SOC or IT analytics workflow.

Tools Used

- **Splunk Cloud Trial** (or Splunk Free)
 - **Web Browser**
-

Procedure and Observations

Step 1: Pick a Dataset & Set Time

- Navigated to **Apps ▸ Search & Reporting**.
- Set the time range to **Last 24 hours**.
- Selected the preferred dataset:
- `index=_internal sourcetype=splunkd_ui_access`

(Fallback: `index=_internal sourcetype=splunkd` if needed)

- This dataset includes web-access-like fields such as `user`, `status`, `uri_path`, and occasionally `bytes`.

New Search

1 index=_internal sourcetype=splunkd_ui_access

168 events (10/23/25 12:30:00.000 PM to 10/24/25 12:36:48.000 PM) No Event Sampling

Events (168) Patterns Statistics Visualization

Format Timeline Zoom Out Zoom to Selection Deselect

1 hour per column

List Format 20 Per Page

< Hide Fields All Fields

Time	Event
10/24/25 12:36:45.369 PM	127.0.0.1 - admin [24/Oct/2025:12:36:45.369 +0530] "POST /en-US/splunkd/_raw/servicesNS/admin/search/search/v2/jobs/rt_md_1761289599.215/results_preview HTTP/1.1" 200 208 "-" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36 - 012d3d2c4bd358fdda769aa8eb85c4 9ms host = LAPTOP-EU4VF6GR source = C:\Program Files\Splunk\varlog\splunk\splunkd_ui_access.log sourcetype = splunkd_ui_access
10/24/25 12:36:45.306 PM	127.0.0.1 - admin [24/Oct/2025:12:36:45.306 +0530] "GET /en-US/splunkd/_raw/servicesNS/nobody/search/search/v2/jobs/rt_md_1761289599.215/output_mode=json&_t=1761289597599 HTTP/1.1" 200 4885 "-" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36 - 012d3d2c4bd358fdda769aa8eb85c4 58ms host = LAPTOP-EU4VF6GR source = C:\Program Files\Splunk\varlog\splunk\splunkd_ui_access.log sourcetype = splunkd_ui_access
10/24/25 12:36:42.303 PM	127.0.0.1 - admin [24/Oct/2025:12:36:42.303 +0530] "POST /en-US/splunkd/_raw/servicesNS/admin/search/search/v2/jobs/rt_md_1761289599.215/results_preview HTTP/1.1" 200 208 "-" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36 - 012d3d2c4bd358fdda769aa8eb85c4 26ms host = LAPTOP-EU4VF6GR source = C:\Program Files\Splunk\varlog\splunk\splunkd_ui_access.log sourcetype = splunkd_ui_access
10/24/25 12:36:42.230 PM	127.0.0.1 - admin [24/Oct/2025:12:36:42.230 +0530] "GET /en-US/splunkd/_raw/servicesNS/nobody/search/search/v2/jobs/rt_md_1761289599.215/output_mode=json&_t=1761289597598 HTTP/1.1" 200 4816 "-" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36 - 012d3d2c4bd358fdda769aa8eb85c4 58ms host = LAPTOP-EU4VF6GR source = C:\Program Files\Splunk\varlog\splunk\splunkd_ui_access.log sourcetype = splunkd_ui_access
10/24/25 12:36:41.322 PM	127.0.0.1 - admin [24/Oct/2025:12:36:41.322 +0530] "GET /en-US/splunkd/_raw/servicesNS/nobody/search/search/v2/jobs/rt_md_1761289599.215/output_mode=json&_t=1761289597597 HTTP/1.1" 200 3478 "-" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36 - 012d3d2c4bd358fdda769aa8eb85c4 51ms host = LAPTOP-EU4VF6GR source = C:\Program Files\Splunk\varlog\splunk\splunkd_ui_access.log sourcetype = splunkd_ui_access
10/24/25 12:36:41.304 PM	127.0.0.1 - admin [24/Oct/2025:12:36:41.304 +0530] "POST /en-US/splunkd/_raw/servicesNS/nobody/splunk_instrumentation/instrumentation_controller/instrumentation_eligibility?optInVersion=4L->1761289597587 HTTP/1.1" 200 279 "-" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36 - 012d3d2c4bd358fdda769aa8eb85c4 69ms

Step 2: Using count — How Many?

- Found the total number of events by sourcetype:
index=_internal | stats count by sourcetype
- Narrowed the view to UI access logs, grouped by HTTP status:
index=_internal sourcetype=splunkd_ui_access | stats count by status | sort - count
- Observed which status codes appeared most frequently.

New Search

1 index=_internal sourcetype=splunkd_ui_access | stats count by status | sort - count

230 events (10/23/25 12:30:00.000 PM to 10/24/25 12:39:43.000 PM) No Event Sampling

Events Patterns Statistics (6) Visualization

20 Per Page Format Preview

status	count
200	212
201	5
303	5
402	4
404	3
304	1

Step 3: Using values() — Unique Field Values

- Displayed all unique HTTP methods used by each user:
index=_internal sourcetype=splunkd_ui_access | stats values(method) by user
- Listed all distinct URI paths seen per status code:
index=_internal sourcetype=splunkd_ui_access | stats values(uri_path) as paths by status

The screenshot shows the Splunk Enterprise search interface. The search bar contains the query: `index=_internal sourcetype=splunkd_ui_access | stats values(method) by user`. The results are displayed in a table with two columns: `user` and `values(method)`. The `user` column has two entries: `-` and `admin`. The `values(method)` column shows the HTTP methods for each user: `GET` for `-` and `DELETE, GET, POST` for `admin`.

user	values(method)
-	GET
admin	DELETE GET POST

Step 4: Using `dc()` — Distinct Counts

- Counted distinct users per HTTP status:
`index=_internal sourcetype=splunkd_ui_access | stats dc(user) as unique_users by status`
- Counted distinct URI paths used by each method:
`index=_internal sourcetype=splunkd_ui_access | stats dc(uri_path) as unique_paths by method`

The screenshot shows the Splunk Enterprise search interface. The search bar contains the query: `index=_internal sourcetype=splunkd_ui_access | stats dc(uri_path) as unique_paths by method`. The results are displayed in a table with two columns: `method` and `unique_paths`. The `method` column has three entries: `DELETE`, `GET`, and `POST`. The `unique_paths` column shows the count of distinct URI paths for each method: `1` for `DELETE`, `91` for `GET`, and `25` for `POST`.

method	unique_paths
DELETE	1
GET	91
POST	25

Step 5: Using `avg()` — Creating a Safe Numeric Field

Because this dataset lacks a built-in numeric metric like “response time,” created a proxy numeric value to calculate averages:

- Used event raw length as a stand-in metric:
`index=_internal sourcetype=splunkd_ui_access`
`| eval raw_len = len(_raw)`
`| stats avg(raw_len) as avg_raw_len by uri_path`
`| sort - avg_raw_len`
- Observed which URIs had the largest average event size.

New Search

```

1 index=_internal sourcetype=splunkd_ui_access
2 | eval raw_len=len(_raw)
3 | stats avg(raw_len) as avg_raw_len by uri_path
4 | sort - avg_raw_len

```

✓ 320 events (10/23/25 12:30:00.000 PM to 10/24/25 12:44:15.000 PM) No Event Sampling

Events Patterns **Statistics (106)** Visualization

20 Per Page Format Preview

url_path	avg_raw_len
/en-US/splunkd/_raw/servicesNS/admin/~data/ui/views	792.3876923876923
/en-US/splunkd/_raw/servicesNS/nobody/search/search/v2/jobs/1761289608.217/events	586
/en-US/splunkd/_raw/services/search/shelper	552
/en-US/static/8FF8A803369C808B6780D47991A20663B85C0D1C62EAB5A9E3A646FC8B7278/fonts/splunkicons-regular-webfont.woff	453
/en-US/static/8FF8A803369C808B6780D47991A20663B85C0D1C62EAB5A9E3A646FC8B7278/fonts/proxima-regular-webfont.woff	449
/en-US/splunkd/_raw/servicesNS/nobody/search/search/v2/jobs/1761289994.224/results_preview	421
/en-US/splunkd/_raw/servicesNS/nobody/search/search/v2/jobs/1761289918.222/results_preview	419
/en-US/splunkd/_raw/servicesNS/nobody/search/search/v2/jobs/1761289746.219/results_preview	418
/en-US/splunkd/_raw/servicesNS/nobody/search/search/v2/jobs/1761289783.220/results_preview	418
/en-US/splunkd/_raw/servicesNS/nobody/search/search/v2/jobs/1761289859.221/results_preview	418
/en-US/splunkd/_raw/servicesNS/nobody/search/search/v2/jobs/1761289963.223/results_preview	418
/en-US/splunkd/_raw/servicesNS/admin/search/data/ui/visualizations	417
/en-US/splunkd/_raw/servicesNS/nobody/splunk_instrumentation/instrumentation_controller/instrumentation_eligibility	402.25
/en-US/splunkd/_raw/servicesNS/admin/search/data/ui/views	397
/en-US/splunkd/_raw/servicesNS/nobody/search/search/v2/jobs/1761289713.218/results_preview	397

Step 6: Combining Multiple Metrics in One stats

Created a single stats command combining count, distinct, and average metrics for a richer overview.

```
index=_internal sourcetype=splunkd_ui_access
```

```
| eval raw_len=len(_raw)
```

```
| stats count, dc(user) as unique_users, values(method) as methods, avg(raw_len) as avg_raw_len by status
```

```
| table status count unique_users methods avg_raw_len
```

- Produced a well-organized table summarizing activity per status code.
- This combined view is excellent for dashboards or health summaries.

New Search

```

1 index=_internal sourcetype=splunkd_ui_access
2 | eval raw_len=len(_raw)
3 | stats count, dc(user) as unique_users, values(method) as methods, avg(raw_len) as avg_raw_len by status
4 | table status count unique_users methods avg_raw_len

```

✓ 411 events (10/23/25 12:30:00.000 PM to 10/24/25 12:46:20.000 PM) No Event Sampling

Events Patterns **Statistics (6)** Visualization

20 Per Page Format Preview

status	count	unique_users	methods	avg_raw_len
200	382	2	DELETE GET POST	363.2041884816754
201	16	1	POST	291.8625
303	5	2	GET	237
304	1	1	GET	254
402	4	1	GET	387.75
404	3	1	GET	308.3333333333333

Step 7: Grouping Strategy & Filters

- Identified top users:

```
index=_internal sourcetype=splunkd_ui_access | top user limit=5
```

- Performed deep-dive on a specific user (from the top list):

- `index=_internal sourcetype=splunkd_ui_access user=<username> | stats count by uri_path`
- Applied post-aggregation filtering:
`... | stats count by uri_path | where count > 10`

The screenshot shows the Splunk Enterprise interface with a search results table. The search query is `index=_internal sourcetype=splunkd_ui_access (user=admin*) | stats count by uri_path`. The results table has two columns: `uri_path` and `count`. The results are sorted by count in descending order.

uri_path	count
/en-US/	1
/en-US/account/login	1
/en-US/app/launcher	1
/en-US/app/launcher/home	1
/en-US/app/search	1
/en-US/app/search/search	1
/en-US/config	2
/en-US/splunkd/_raw/apps/local	1
/en-US/splunkd/_raw/services/apps/local/launcher	1
/en-US/splunkd/_raw/services/authentication/users/admin	2
/en-US/splunkd/_raw/services/authorization/roles	1
/en-US/splunkd/_raw/services/configs/conf-web/settings	3
/en-US/splunkd/_raw/services/data/ui/prefs/new	2
/en-US/splunkd/_raw/services/data/user-prefs/general	2
/en-US/splunkd/_raw/services/dmc-conf/settings/settings	1
/en-US/splunkd/_raw/services/messages	19
/en-US/splunkd/_raw/services/search/jobs/1761289688_2117/timeline	1

Step 8: Save a Reusable Report

- Created a reusable report summarizing methods and averages:
`index=_internal sourcetype=splunkd_ui_access`
`| eval raw_len=len(_raw)`
`| stats count, dc(user) as unique_users, avg(raw_len) as avg_raw_len by method`
- Saved report as: **Day55: Methods Summary**.
- Added description and verified it under *Reports*.

The screenshot shows the Splunk Enterprise interface with a reusable report titled "Day55: Methods Summary". The report is for the last 24 hours and contains 506 events. The results table has four columns: `method`, `count`, `unique_users`, and `avg_raw_len`.

method	count	unique_users	avg_raw_len
DELETE	1	1	316
GET	365	2	373.2328767123288
POST	140	1	328.7214285714286

Reflection

- **When is `values()` helpful vs. overwhelming?**
`values()` is great for small sets (like listing HTTP methods per user), but can become unreadable when there are too many unique values (e.g., hundreds of URI paths).

- **What business question does `dc()` answer that `count` cannot?**
count measures total volume; `dc()` measures **unique participation** (e.g., number of unique users or endpoints).
 - **What numeric field would you average in a real environment?**
I'd average **response_time**, **bytes**, or **duration** to monitor performance trends. If unavailable, proxy metrics like `_raw` length or calculated fields can still provide relative insights.
-

Summary

In this lab, I explored Splunk's most fundamental transforming commands:

- **count** for totals,
- **values()** for unique listings,
- **dc()** for distinct counts, and
- **avg()** for averages.

By combining these with grouping, field shaping, and report creation, I built the foundation for quantitative analysis in Splunk — a key skill for generating dashboards, alerts, and SOC insights.