IST-719

Week-3 Homework

Topic: Visually describe a Dataset

Taught by Prof. Gary Krudys
Student: Sharat Sripada
(vssripad@syr.edu)

TABLE OF CONTENTS

1.0 Introduction

This homework and all following deliverables for this course will consume a dataset extracted from a network log aggregator, that delivers insights about problems that may exist in on-premises datacenters to network and application administrators. Several nodes and applications generate logs that help developers and administrators root-cause issues and sometimes it becomes important to aggregate them in one place so insights can be drawn across stacks.

In this specific case, logs from the syslog daemon on several nodes like network/security-based control-plane, data-plane (Hypervisors), edges and virtual-machines hosting services like DHCP, DNS etc. have been re-directed to a network log-insight tool. The idea is to use the logs in their raw form and extract interesting insights like the following:

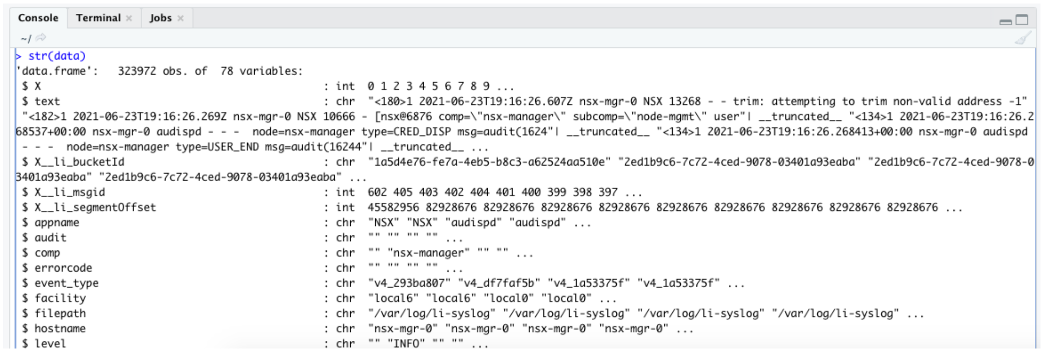
- Alert categories and their distribution by counts *(see Fig1. Plot: Alerts by severity−type in network)*
- Alerts grouped by network nodes that tend to provide an insight into what nodes are generating higher volumes *(see Fig2. Plot: Alerts by node−type in network)*
- Alerts plotted by type across time to understand if there is correlation of higher volumes at certain times of the day *(see Fig3. Plot: Alerts across time-of-day)*
- Grouping network nodes by function and visualizing a distribution across alert types *(see Fig4. Plot: Dissecting alerts generated by network functions)*
- Gathering an overall health of the network at-a-glance *(see Fig5. Plot: Health of the network)*
- A spatial map of the nodes in the network based on hostnames and components (*TBD*)
- Heat map of alerts to indicate hot zones that require immediate attention (*TBD*)

2.0 About the data

The data was downloaded in a csv format (separated by ‘,’) and loaded into a data-frame using the read.csv function. A sample line is shown below:

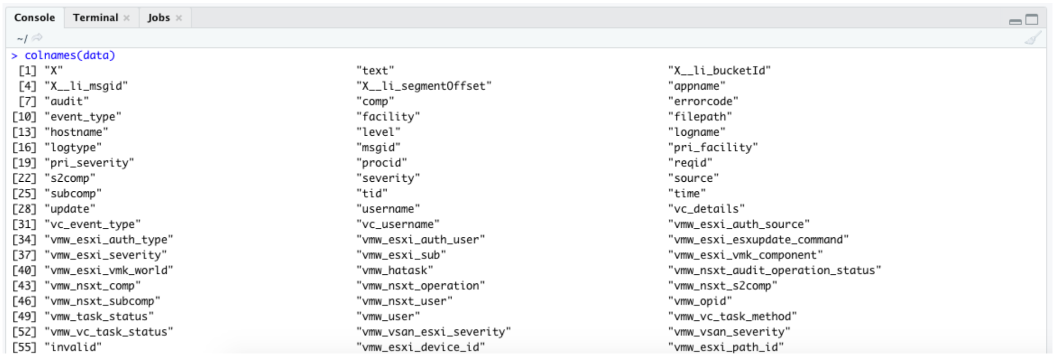
```
240,"<179> 2021-06-23T19:16:02.291784+00:00 nsx-edge-1-c0 NSX 32192 LB [nsx@6876 comp=""nsx-edge""
subcomp=""nsx-edge-lb.lb"" level=""ERROR""] ""no virtual server defined in lbs ff70393e-4490-471b-
8007-8e22b647fa8f.""",2ed1b9c6-7c72-4ced-9078-03401a93eaba,545,82883403,NSX,,nsx-
edge,,v4_dcb8d3e0,local6,/var/log/li-syslog,nsx-edge-1-
c0,ERROR,,,LB,22.0,3.0,32192,,,err,20.20.216.157,nsx-edge-lb.lb,,2021-06-23
19:16:02.291,,,,,,,,,,,,,,,nsx-edge,,,nsx-edge-lb.lb,,,,,,,,,,,,,,,,,,,,,,,,,,,,edg  lb  comp
edg subcomp edg lb
```

Following are some insights of the data:



```
> str(data)
'data.frame':   323972 obs. of  78 variables:
 $ X                                  : int  0 1 2 3 4 5 6 7 8 9 ...
 $ text                               : chr  "<180>1 2021-06-23T19:16:26.607Z nsx-mgr-0 NSX 13268 - - trim: attempting to trim non-valid address -1"
 "<182>1 2021-06-23T19:16:26.269Z nsx-mgr-0 NSX 10666 - [nsx@6876 comp=\"nsx-manager\" subcomp=\"node-mgmt\" user"| __truncated__ "<134>1 2021-06-23T19:16:26.2
68537+00:00 nsx-mgr-0 audispd - - -  node=nsx-manager type=CRED_DISP msg=audit(1624"| __truncated__ "<134>1 2021-06-23T19:16:26.268413+00:00 nsx-mgr-0 audispd
 - - -  node=nsx-manager type=USER_END msg=audit(16244"| __truncated__ ...
 $ X__li_bucketId                     : chr  "1a5d4e76-fe7a-4eb5-b8c3-a62524aa510e" "2ed1b9c6-7c72-4ced-9078-03401a93eaba" "2ed1b9c6-7c72-4ced-9078-0
3401a93eaba" "2ed1b9c6-7c72-4ced-9078-03401a93eaba" ...
 $ X__li_msgid                        : int  602 405 403 402 404 401 400 399 398 397 ...
 $ X__li_segmentOffset                : int  45582956 82928676 82928676 82928676 82928676 82928676 82928676 82928676 82928676 82928676 ...
 $ appname                            : chr  "NSX" "NSX" "audispd" "audispd" ...
 $ audit                              : chr  "" "" "" "" ...
 $ comp                               : chr  "" "nsx-manager" "" "" ...
 $ errorcode                          : chr  "" "" "" "" ...
 $ event_type                         : chr  "v4_293ba807" "v4_df7faf5b" "v4_1a53375f" "v4_1a53375f" ...
 $ facility                           : chr  "local6" "local6" "local0" "local0" ...
 $ filepath                           : chr  "/var/log/li-syslog" "/var/log/li-syslog" "/var/log/li-syslog" "/var/log/li-syslog" ...
 $ hostname                           : chr  "nsx-mgr-0" "nsx-mgr-0" "nsx-mgr-0" "nsx-mgr-0" ...
 $ level                              : chr  "" "INFO" "" "" ...
```

Fig. Code: R code showing structure of data



```
> colnames(data)
 [1] "X"                             "text"                          "X__li_bucketId"
 [4] "X__li_msgid"                   "X__li_segmentOffset"           "appname"
 [7] "audit"                         "comp"                          "errorcode"
[10] "event_type"                    "facility"                      "filepath"
[13] "hostname"                      "level"                         "logname"
[16] "logtype"                       "msgid"                         "pri_facility"
[19] "pri_severity"                  "procid"                        "reqid"
[22] "s2comp"                        "severity"                      "source"
[25] "subcomp"                       "tid"                           "time"
[28] "update"                        "username"                      "vc_details"
[31] "vc_event_type"                 "vc_username"                   "vmw_esxi_auth_source"
[34] "vmw_esxi_auth_type"            "vmw_esxi_auth_user"            "vmw_esxi_esxupdate_command"
[37] "vmw_esxi_severity"             "vmw_esxi_sub"                  "vmw_esxi_vmk_component"
[40] "vmw_esxi_vmk_world"            "vmw_hatask"                    "vmw_nsxt_audit_operation_status"
[43] "vmw_nsxt_comp"                 "vmw_nsxt_operation"            "vmw_nsxt_s2comp"
[46] "vmw_nsxt_subcomp"              "vmw_nsxt_user"                 "vmw_opid"
[49] "vmw_task_status"               "vmw_user"                      "vmw_vc_task_method"
[52] "vmw_vc_task_status"            "vmw_vsan_esxi_severity"        "vmw_vsan_severity"
[55] "invalid"                       "vmw_esxi_device_id"            "vmw_esxi_path_id"
```

Fig. Code: R code showing column-names of data

2.1 Calculating the score of the data

Using the dim function, the row and column count was obtained as below:

```
> dim(data)
[1] 323972     78
```

Now, using the calculator `(NumberOfColumns * 4) * (NumberOfRows/100)` we obtain the following score:

`(78 * 4) * (323972/100) >> 100`

2.2 Visualizing the data

2.2.1 Alerts grouped by severity in network

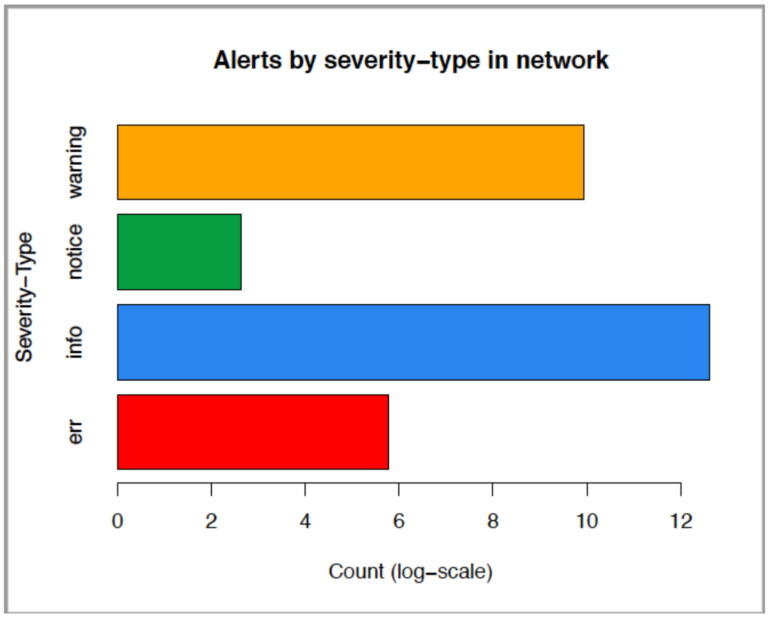


Fig1. Plot: Alerts by severity–type in network

2.2.2 Alerts grouped by node–type in network

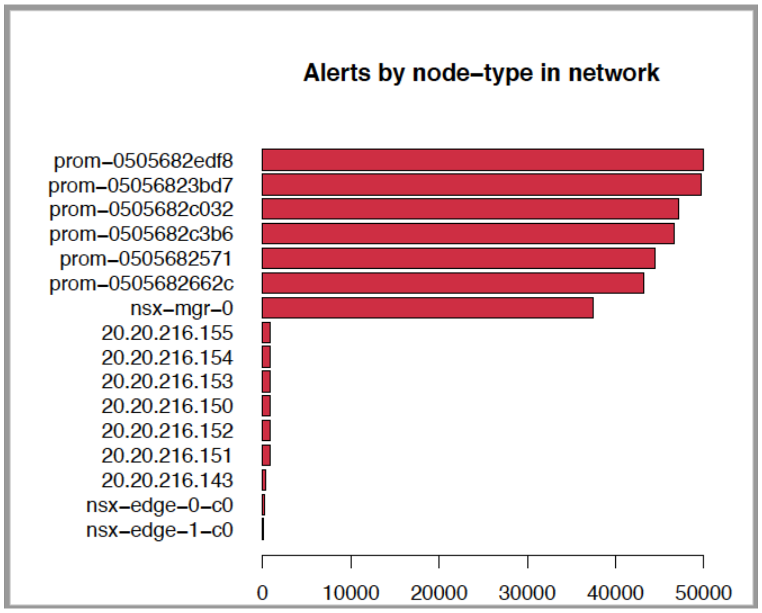


Fig2. Plot: Alerts by node–type in network

2.2.3 Alert spread across time-of-day

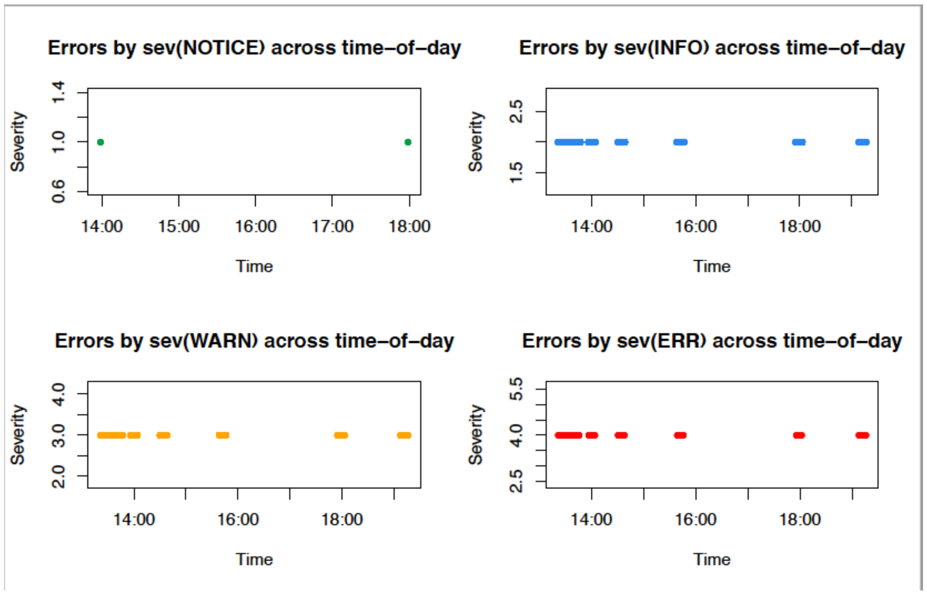


Fig3. Plot: Alerts across time-of-day

2.2.4 Dissecting alerts generated by network functions

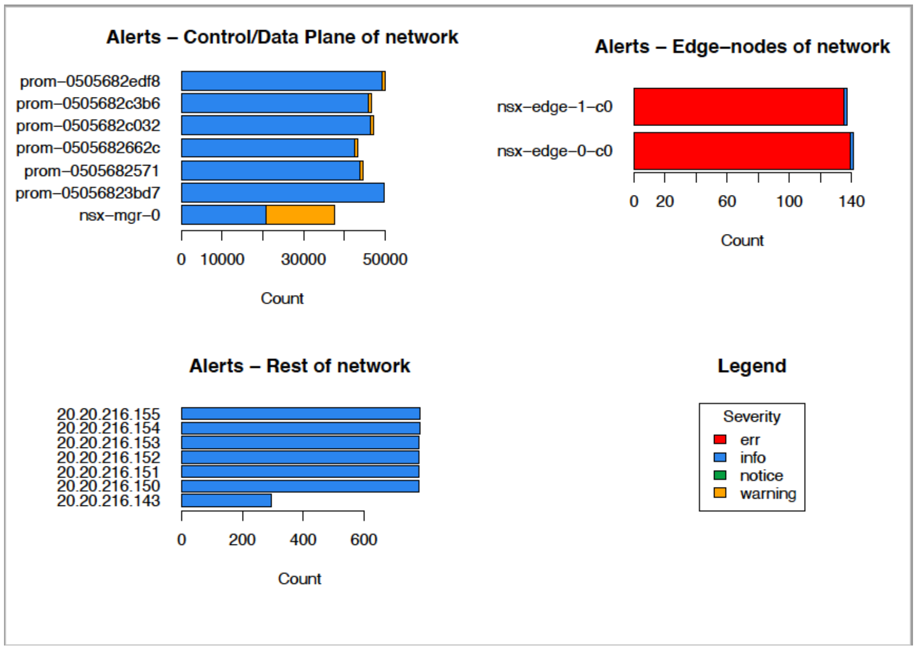


Fig4. Plot: Dissecting alerts generated by network functions

2.2.5 Visualizing health of the network

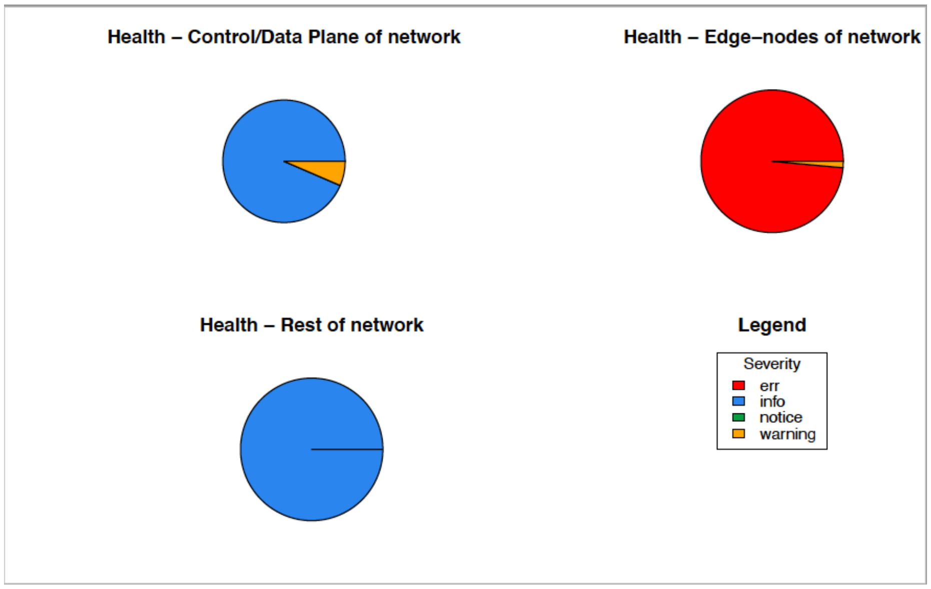


Fig5. Plot: Health of the network

NOTE:
A separate PDF comprising all merged plots will be separately attached to the submission.

3.0 Conclusion

Using the visualization techniques taught in the first 3-weeks of class IST-719, an unstructured dataset of logs was analyzed to recognize various alert severities and indicate where the network was running hot. Extracting these visual insights from raw data can quickly help dive into the root of the problem and improve critical indices like Mean Time to Recover (MTTR). This is especially useful as we drill from chart to another using APIs or URLs at the back end.

Via visualization plots (especially Fig4 and Fig5), it seems apparent that the Edge-nodes are generating alerts of severity 'Error'. This would have likely had an impact on network productivity.

In the weeks to come and lining up to the *Final Poster* the scope will expand to identify components within the nodes that caused 'Error' alerts. Also, spatial graphs interconnecting network nodes and heat-maps should further alleviate the data story.