Predicting Network Fault Severity

Milestone Report

### Introduction

In the telephony network, telephone exchanges and nodes are connected by wire, microwaves or via satellite to provide communication service for people to communicate across long distances. However service disruption can arise from physical damage and network congestion from increased call traffic. When these disruptions occur, the matter could be life and death in emergency situations or delays and losses for businesses. So to ensure network availability to service customers, Telstra is constantly monitoring the network to ensure the network is available to service customers for their communication needs.

Since these disruptions are done by nature or human activity, their forces can at times can be hard to predict but is acknowledged and managed as part of the system. When there a disruption is present, a log of the error is recorded and alternative paths is applied to cater for repairs. Some faults will only disservice customers from a decrease service capability whilst other faults can be a complete shutdown that will need immediate attention.Thus in a business point of view the dilemma is how to allocate resources effectively to reduce repair costs and proactively fix faults before they occur and avoid unnecessary costs.

### DataSet

Data has been sourced with Telstra's kaggle competition at https://www.kaggle.com/c/telstra-recruiting-network. There are 6 data tables about the following:

train.csv - the training set for fault severity with *id , location &* *faulty\_severity\** data

test.csv - the test set for fault severity with  *id* & *location* data

event\_type.csv - event type related to the main dataset with *id* & *event\_type\** data

log\_feature.csv - features extracted from log files with *id , log\_feature\* &* *volume* data

resource\_type.csv - type of resource related to the main dataset with *id* & *resource\_type\** data

severity\_type.csv - warning message from the log with *id* & *severity\_type\** data

(\* is a variable of discrete nature)

#### *Limitations and Assumptions*

The data retrieved is a log of fault errors occurred and cannot describe the specific equipment stored that raises these fault records.

Log feature data have not been described to what matter it refers to in the system and will be assumed to describe the description of the network service’s contributing output since each entry is paired with volume data.

Also, at each location, the fault\_severity index is categorical and hence describes unknown combinations of variables to meet threshold for that index to be assigned. Since all variables are coded we will not be able to deduce specific reasons or use much business expertise to help solve.

#### *Data Cleaning and Wrangling*

In this data, most data did not require tidying because all tables presented columns of categorised variables and not data. Also the data itself is relatively clean, where most values are discrete values. However in loading from the log\_feature dataset, all values were joined as one variable, so these values have been extracted into 3 columns (id, log\_feature & volume ) using separate() .

During the loading of data, all data was extracted as plain strings so as for manual factorising conversion when required for simple analysis and control of data types. All discrete variables are trimmed to become numbers to assist analysis.

The data is ordered by **id** to join tables together as one set to uncover and remove missing data ready for analysis. In this step it is found that the train dataset has less entries than the other tables of log\_feature, resource\_type, event\_type and severity\_type.

*Joined Table contents:* ***network***

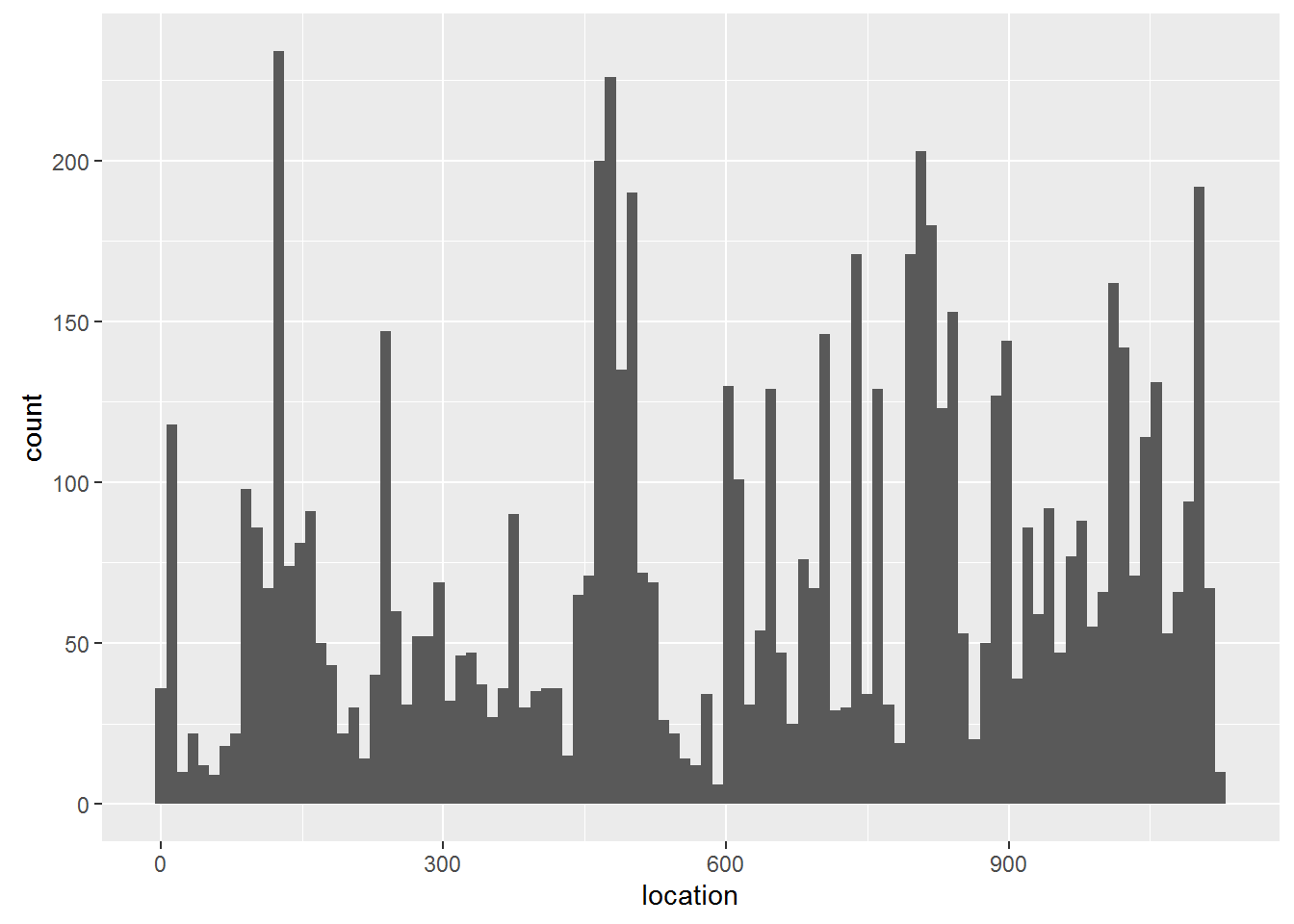
|  |  |  |
| --- | --- | --- |
| **Data name** | **Data type** | **Brief description** |
| id | integer | Record’s identifier , time point |
| location | Integer (factor) | Fault location |
| faulty\_severity | Integer (factor) | 3 levels, 0= no fault ; 1 = few ; 2 = many - actual data from reported fault from users |
| log\_feature | Integer (factor) | Assumption: network service’s faulting feature |
| volume | integer | Assumption: unit measure of faulting feature |
| resource\_type | Integer (factor) | 10 distinct types |
| event\_type | Integer (factor) | Assumption: network service feature’s fault behaviour |
| severity\_type | Integer (factor) | 5 unordered types - warning message received from monitoring machines |

### Preliminary Exploration

#### Histograms

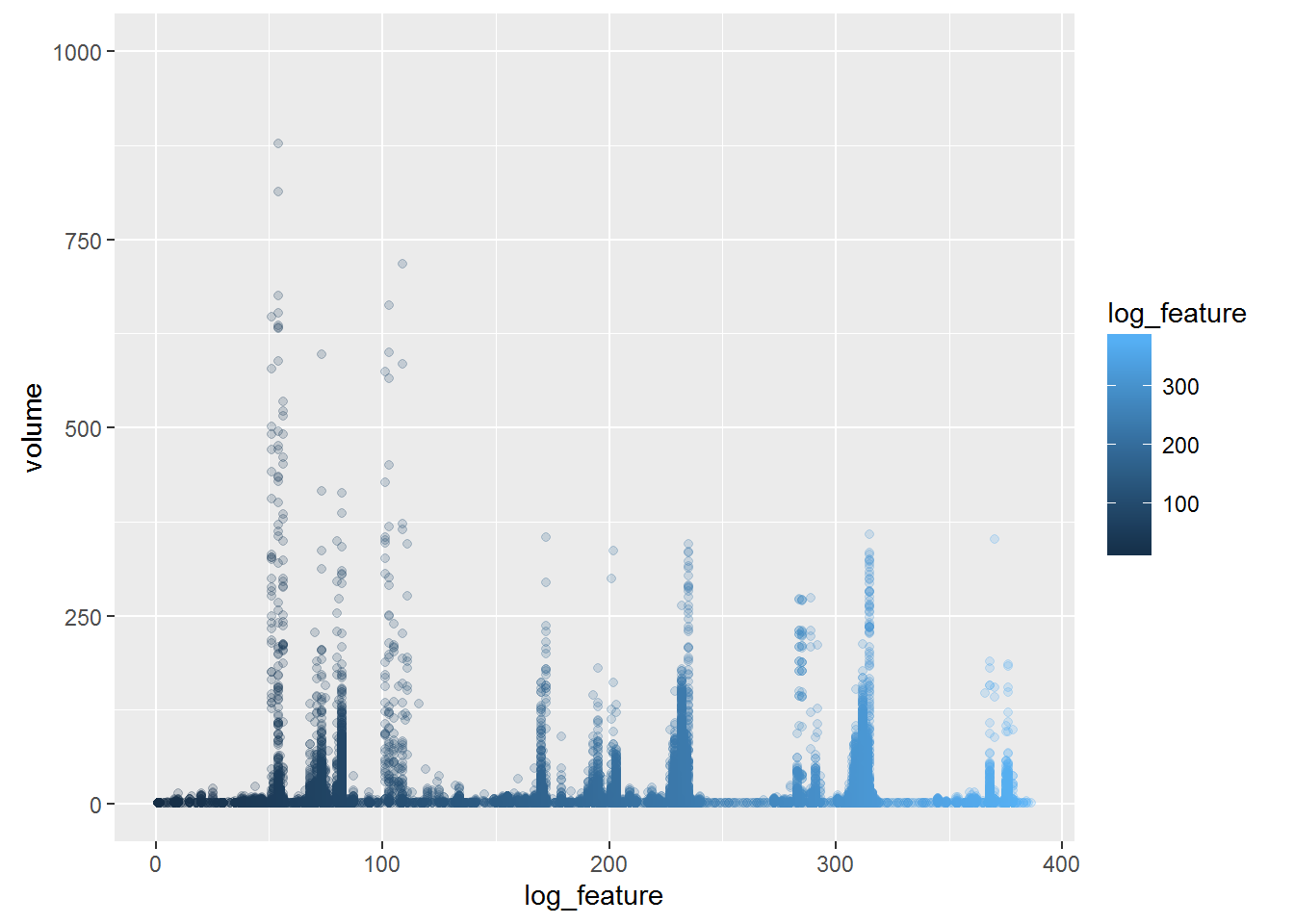
Histograms of each dataset to understand shape and distribution, bins width set to 100 to reveal if there is distinct categorical values

**Train**



ggplot(train, aes(x = location)) + geom\_histogram(bins = 100)

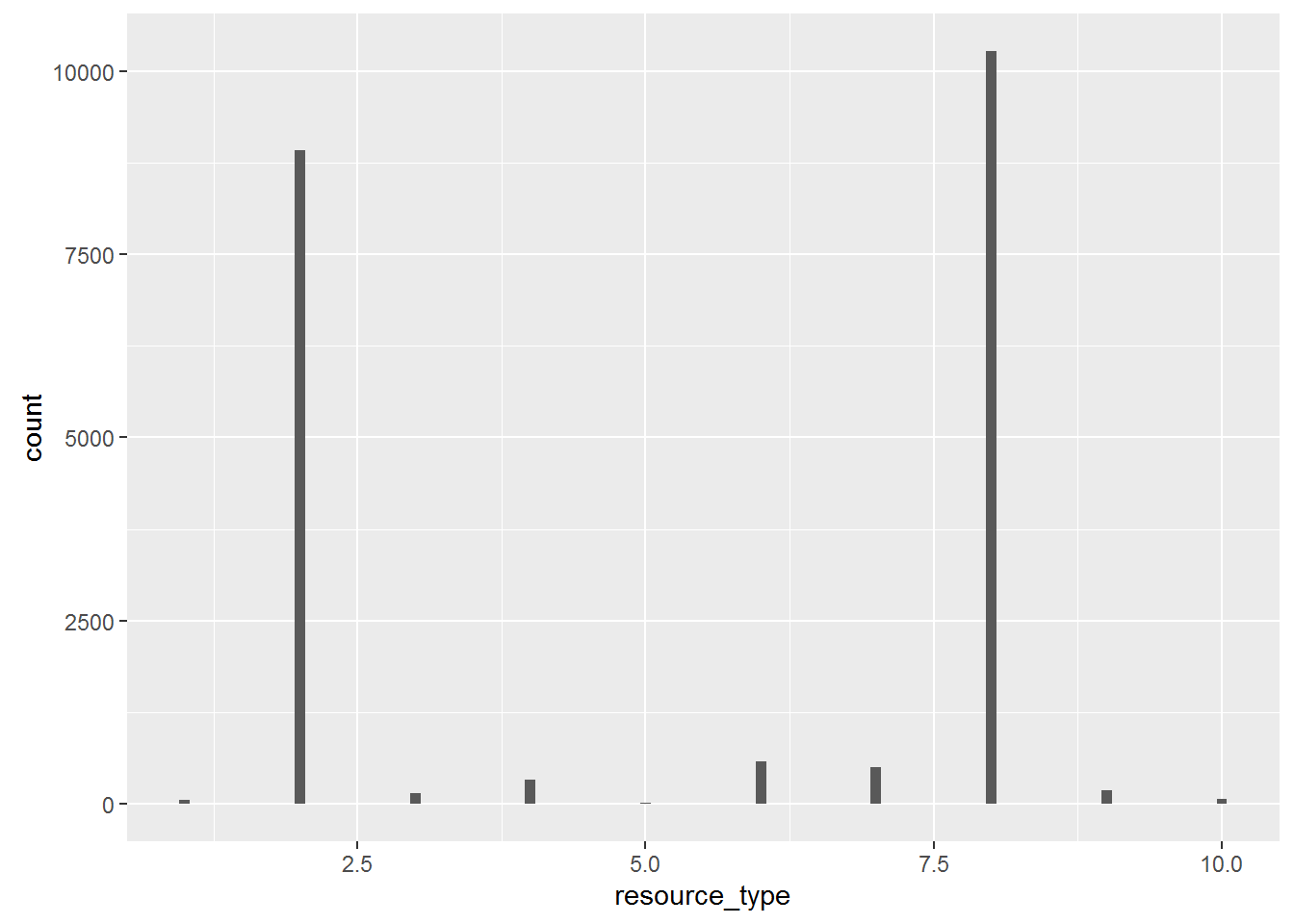
**Log\_feature**



ggplot(log\_feature, aes(x = log\_feature, y = volume, col = log\_feature)) + geom\_point(alpha = 0.2) + coord\_cartesian(ylim = c(0, 1000)) # removed outlier

**#Comment:** The Volume variable here is a continuous measure with some log\_feature more frequent than others

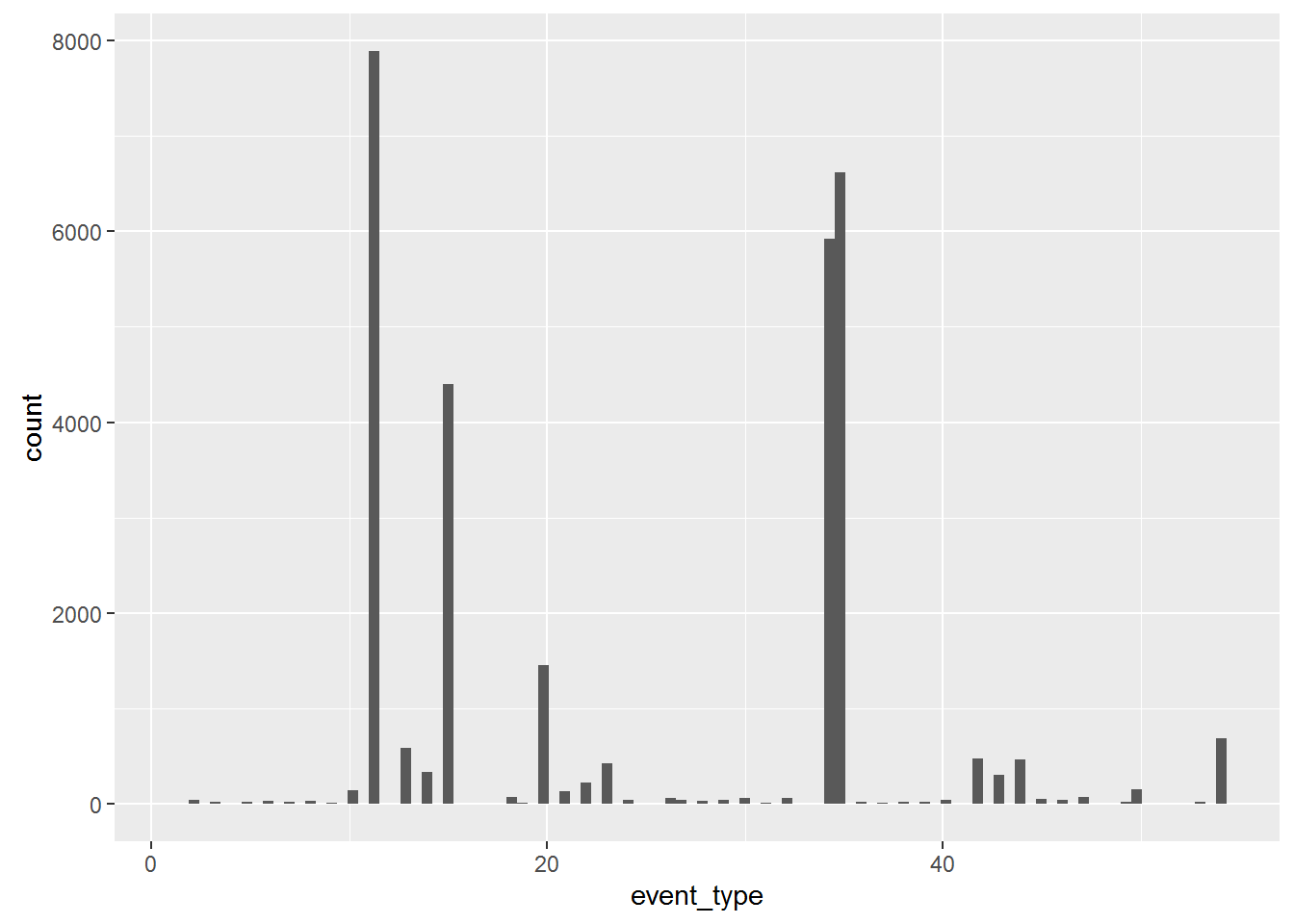
**Resource\_type**



ggplot(resource\_type, aes(x= resource\_type)) + geom\_histogram(bins = 100)

**#Comment:** There are 10 distinct resource types

**Event\_type**



ggplot(event\_type, aes(event\_type)) + geom\_histogram(bins = 100)

**#Comment:** There appears to be distinct but also popular events.

**Severity\_type**



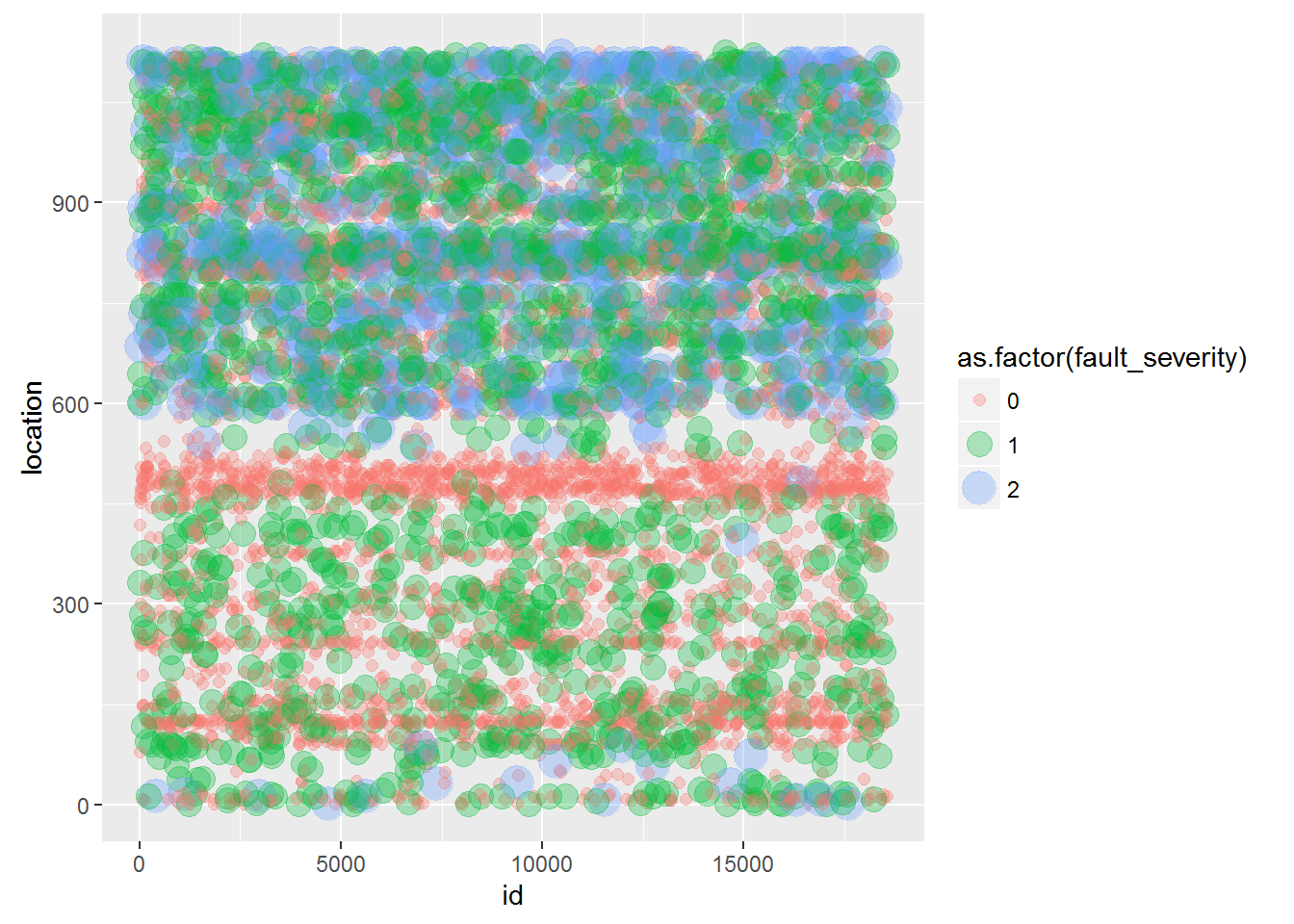
ggplot(sev\_type, aes(severity\_type)) + geom\_histogram(bins = 100)

**#Comment:** There appears to be 5 distinct types.

#### Scatter plot : id(time) scale

Scatter plot of each dataset compared to id (time) to explore time related patterns

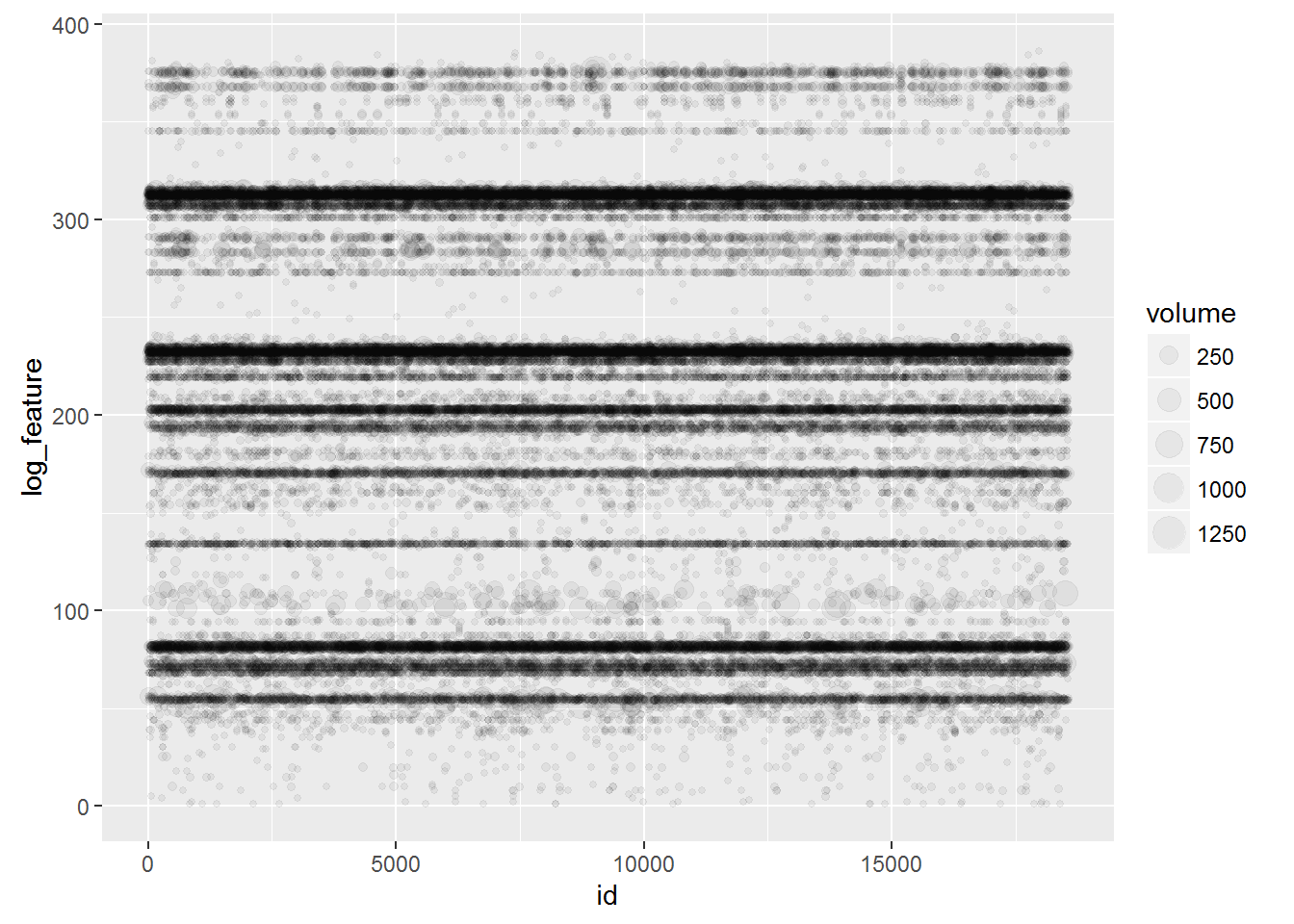
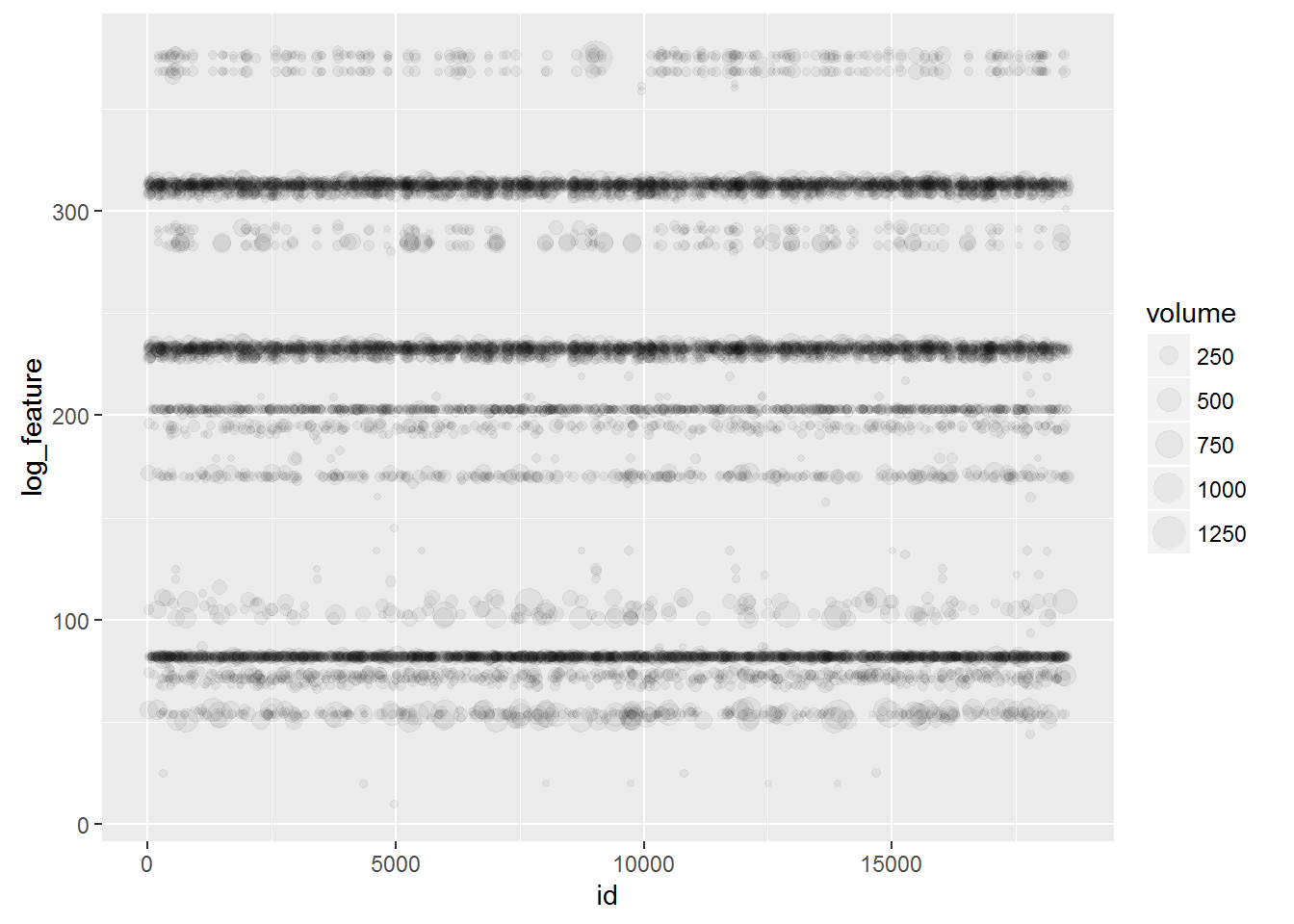
**Train**



ggplot(train, aes(x = id, y = location, col = as.factor(fault\_severity), size = as.factor(fault\_severity) )) + geom\_point(alpha = 0.3)

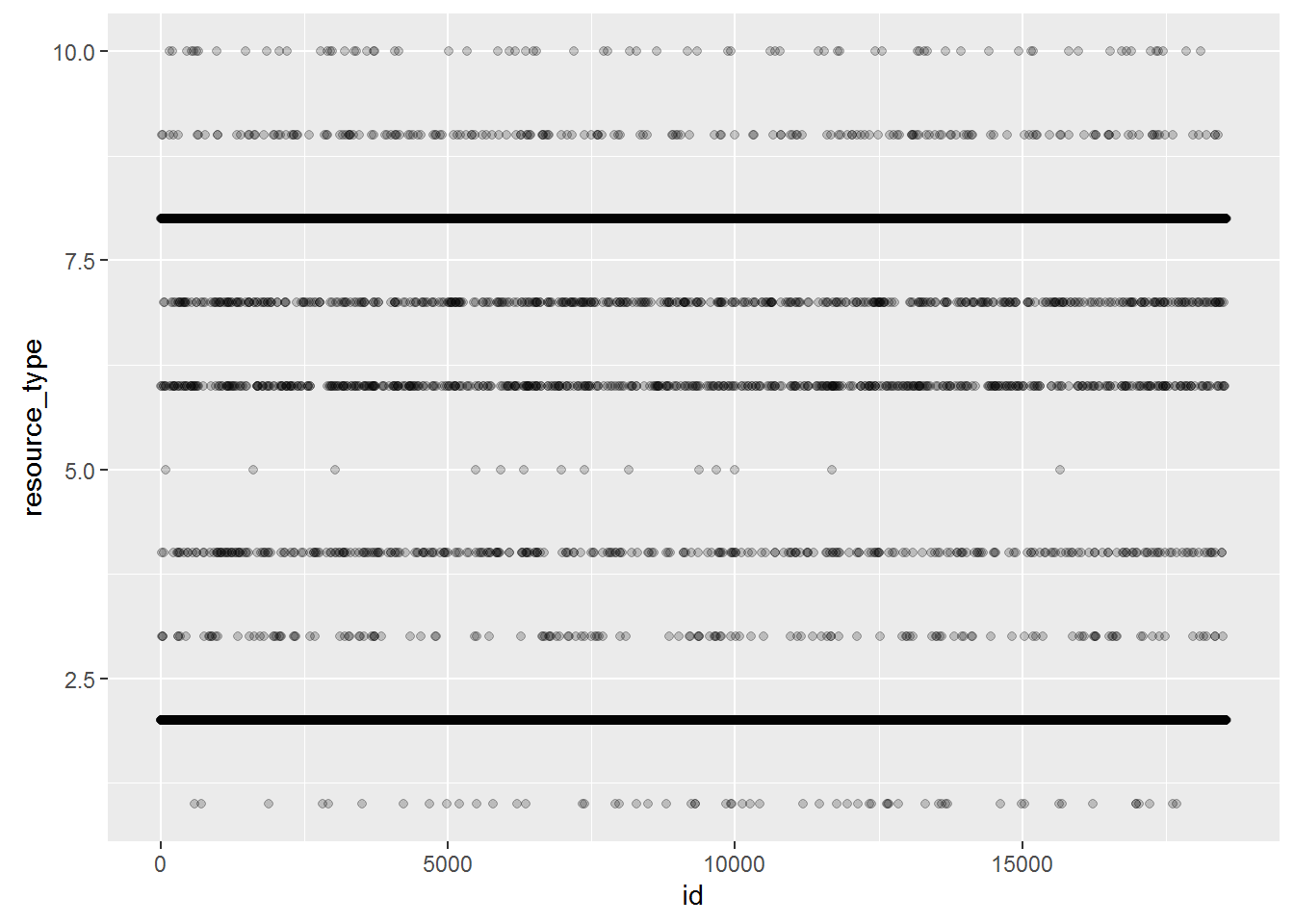
**#Comment:** There are some locations that have continuous faults reported

**Log\_feature**

ggplot(log\_feature, aes(x = id, y = log\_feature, size = volume)) + geom\_point(alpha = 0.05)

**#Comment:** This confirms the train dataset that some log features have continous feature logs, the right is where volume is above 10. This pattern may correlate to machine type found at particular locations.

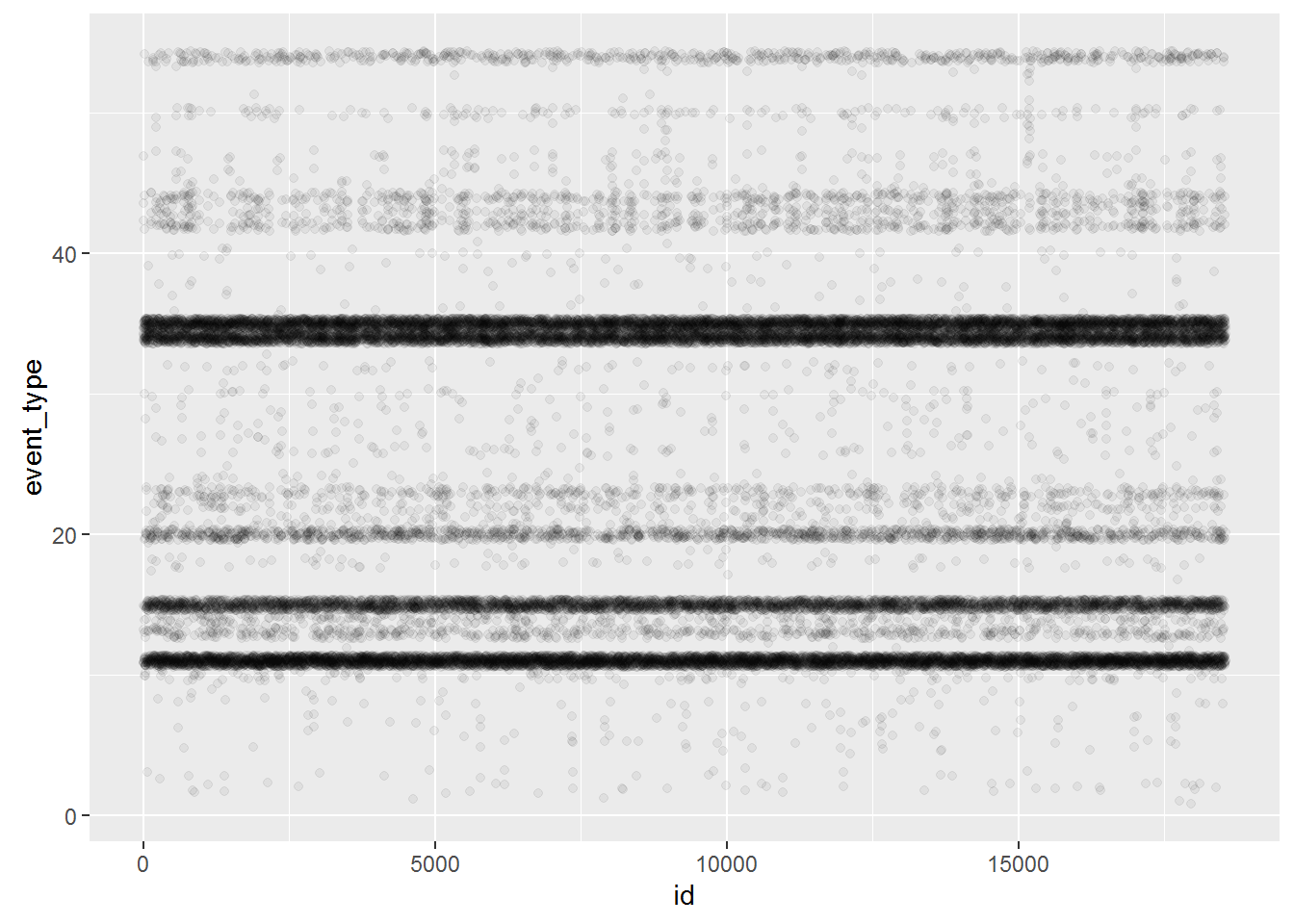
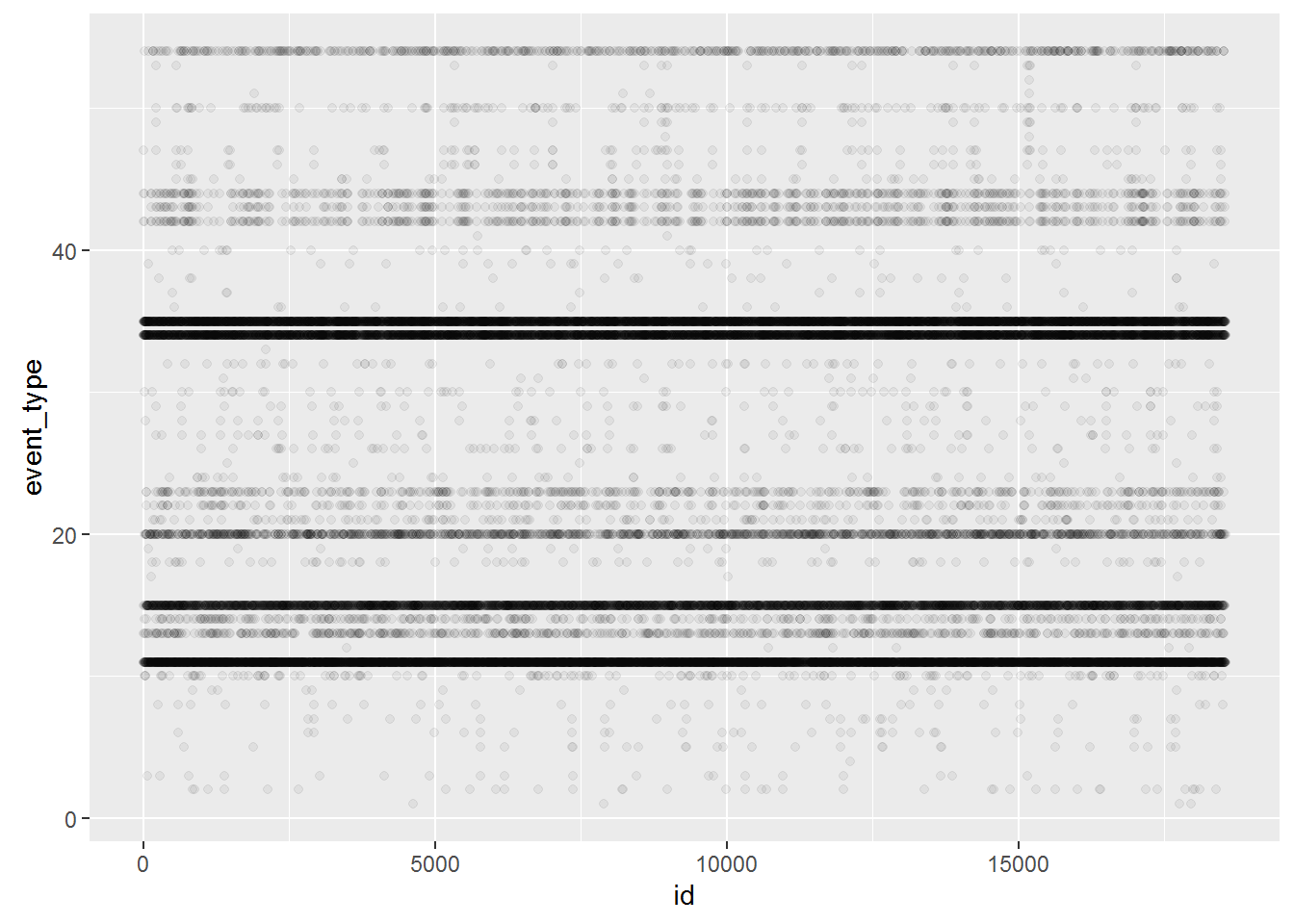
**Resource\_type**



ggplot(resource\_type, aes(x = id, y = resource\_type)) + geom\_point(alpha = 0.2)

**#Comment:** This is a cleaner plot with some resource being continuous like Log\_feature and Train datasets.

**Event\_type**

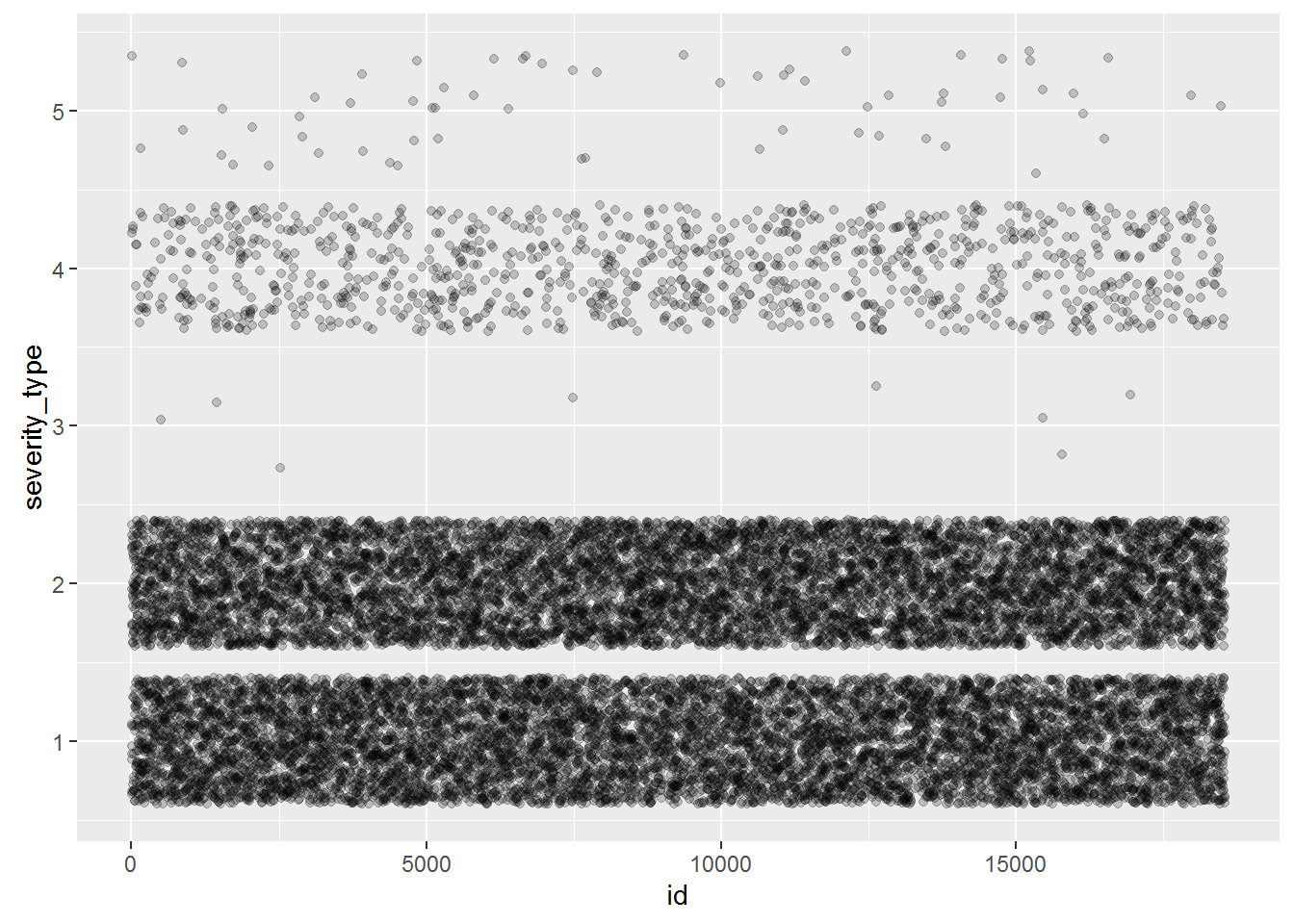


ggplot(event\_type, aes(x = id, y = event\_type)) + geom\_point(alpha = 0.05)

ggplot(event\_type, aes(x = id, y = event\_type)) + geom\_jitter(alpha = 0.05)

**#Comment:** There are certain event types that are frequently occuring or being recorded. Reviewing with jitter does not show much gaps in frequency

**Severity\_type**



ggplot(sev\_type, aes(x = id, y = severity\_type)) + geom\_jitter(alpha = 0.2)

ggplot(network, aes(x = id, y = severity\_type, col = as.factor(fault\_severity), size = as.factor(fault\_severity))) + geom\_jitter(alpha = 0.1)

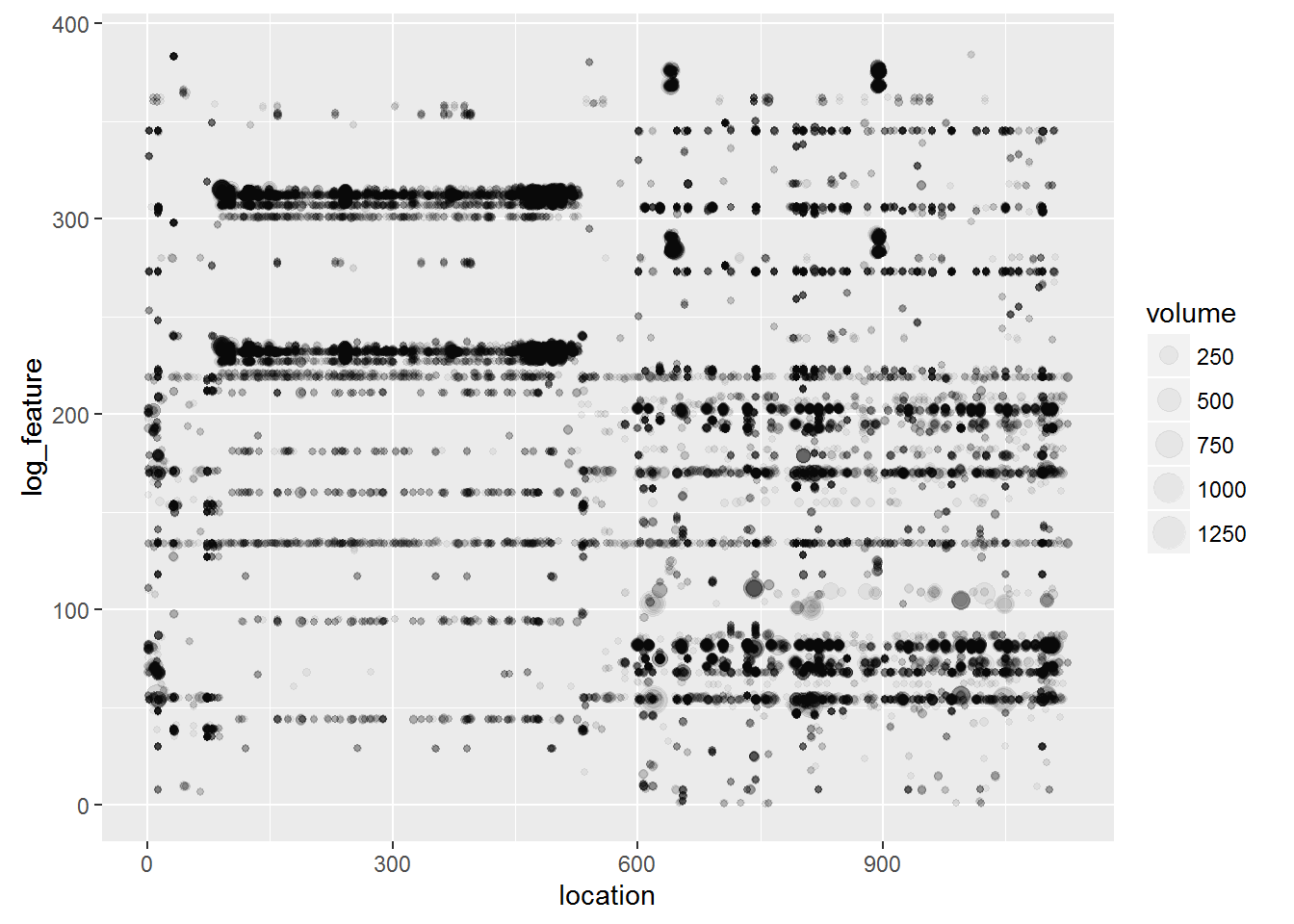
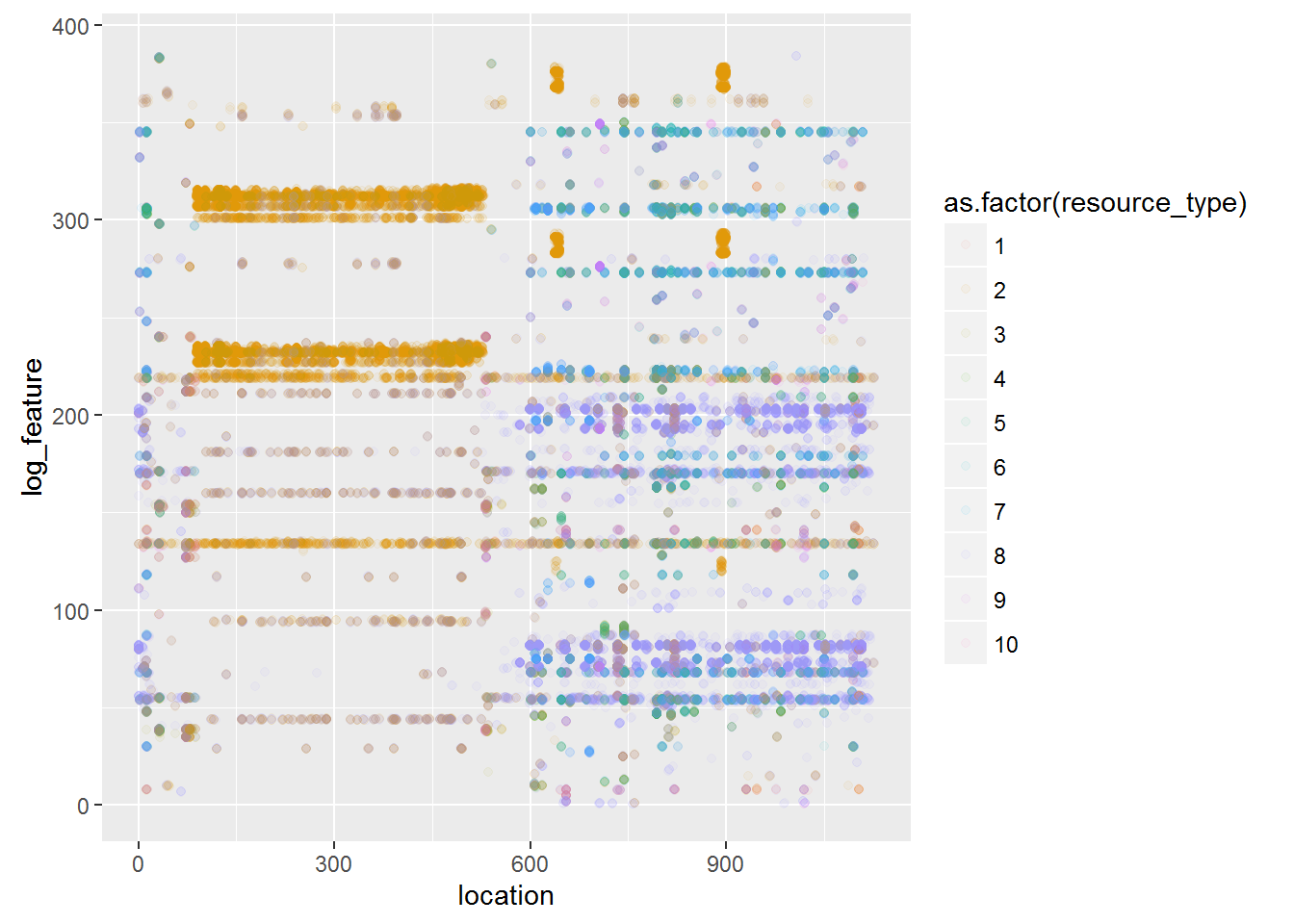
**#Comment:** The distinct categories have clear proportion clear distributed in the logs. Severity\_type no.1 has the most severity impact on users. Severity\_type no.3 only gives a low level of severity on rare occasions.

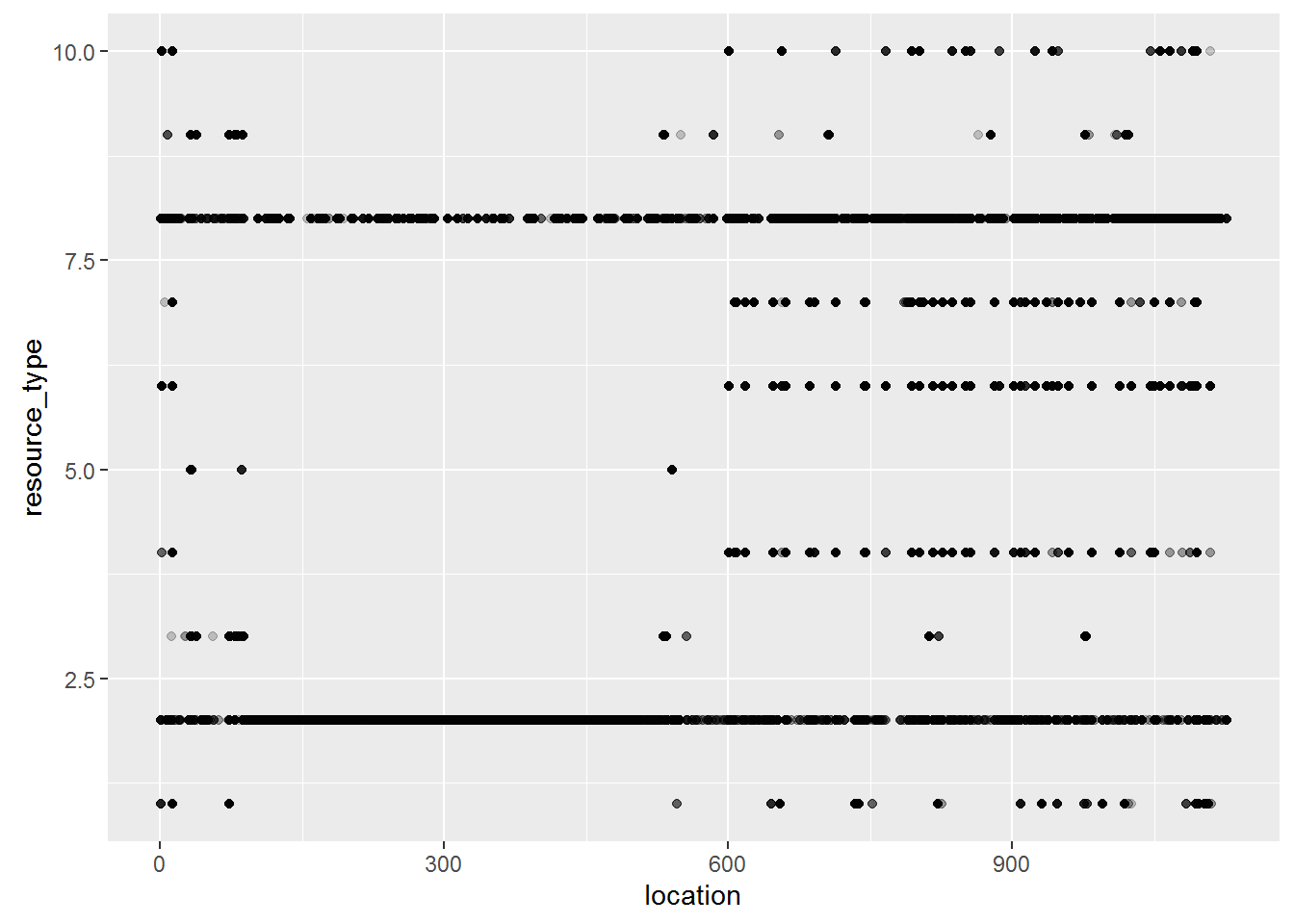
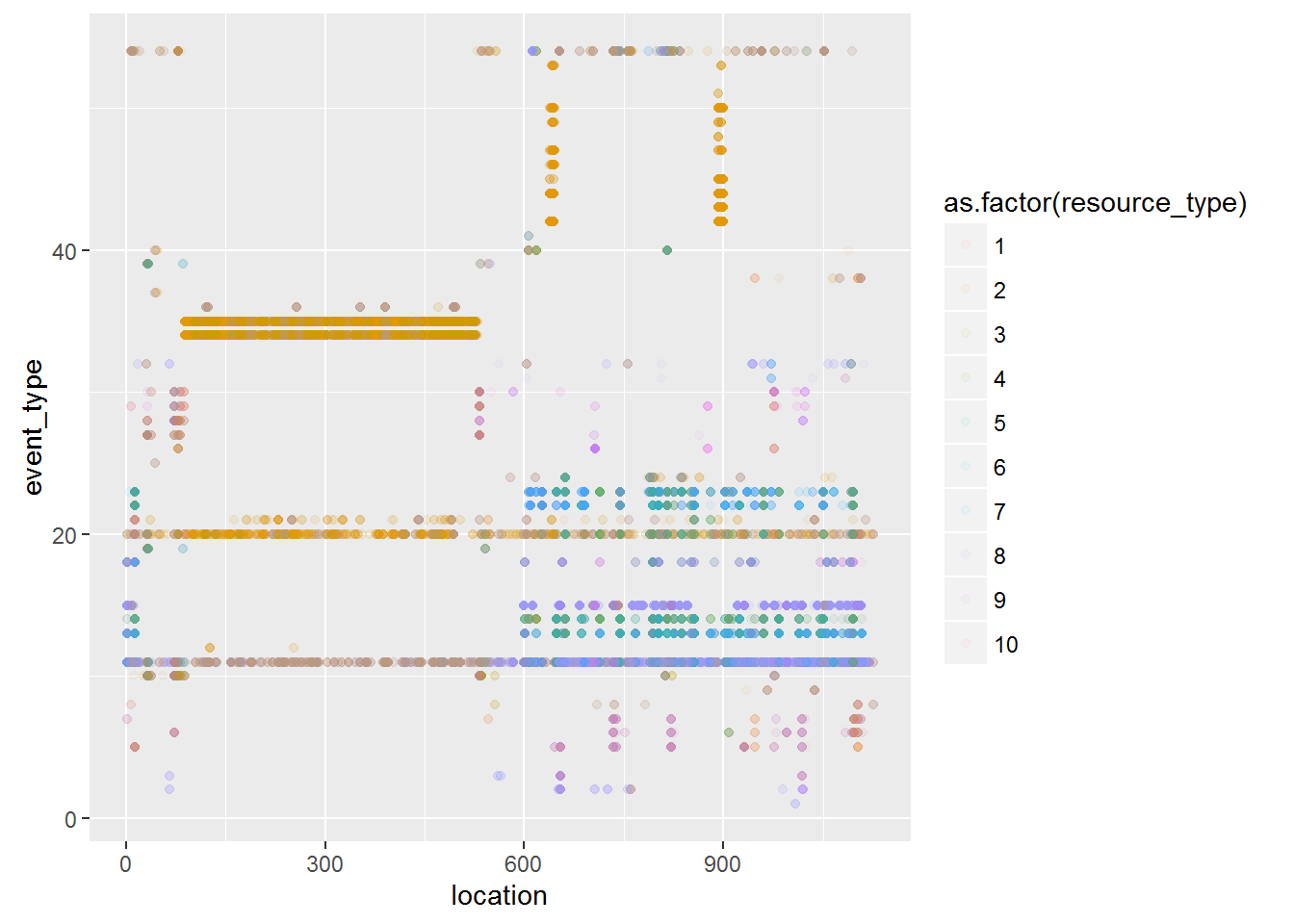
#### 

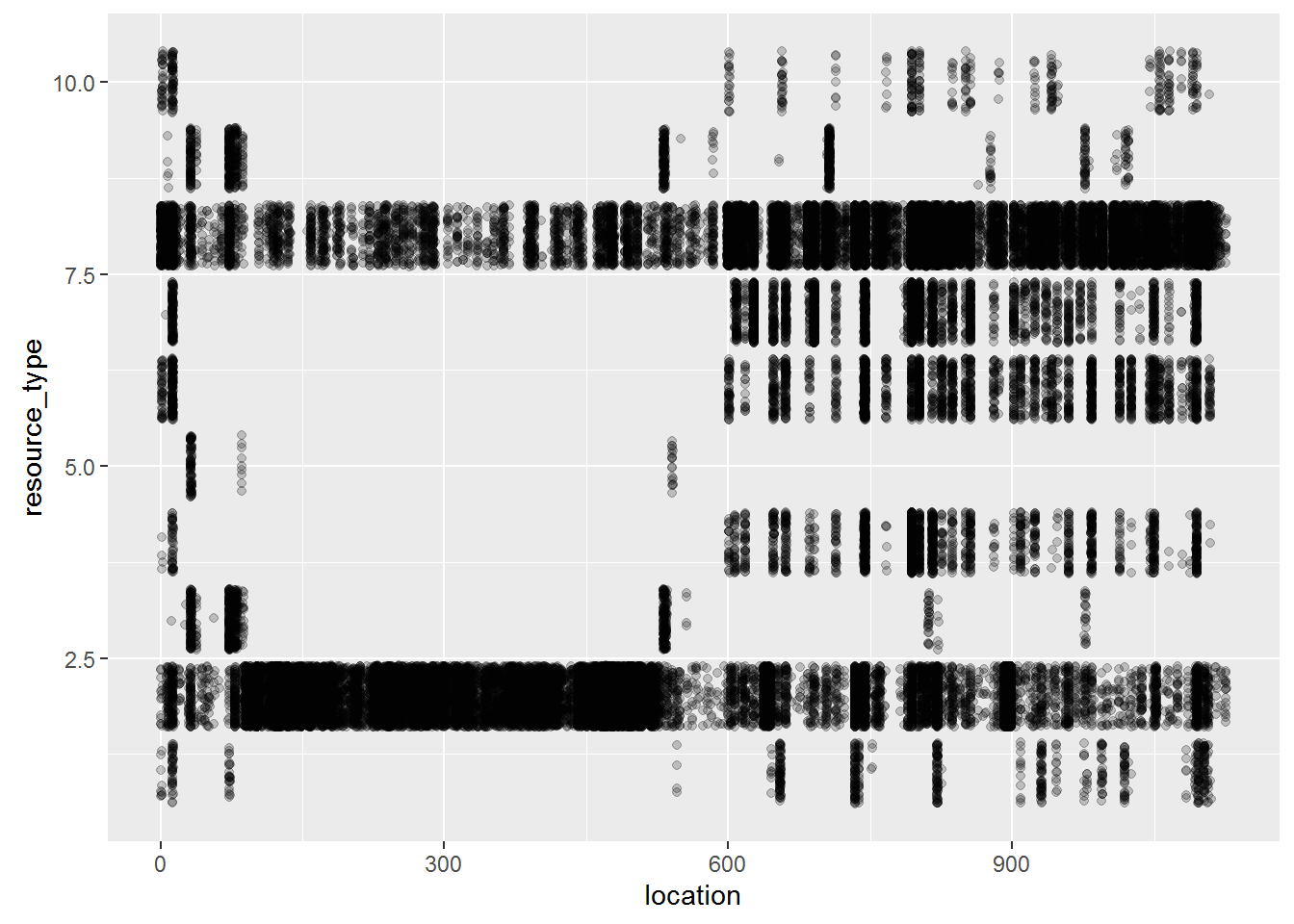
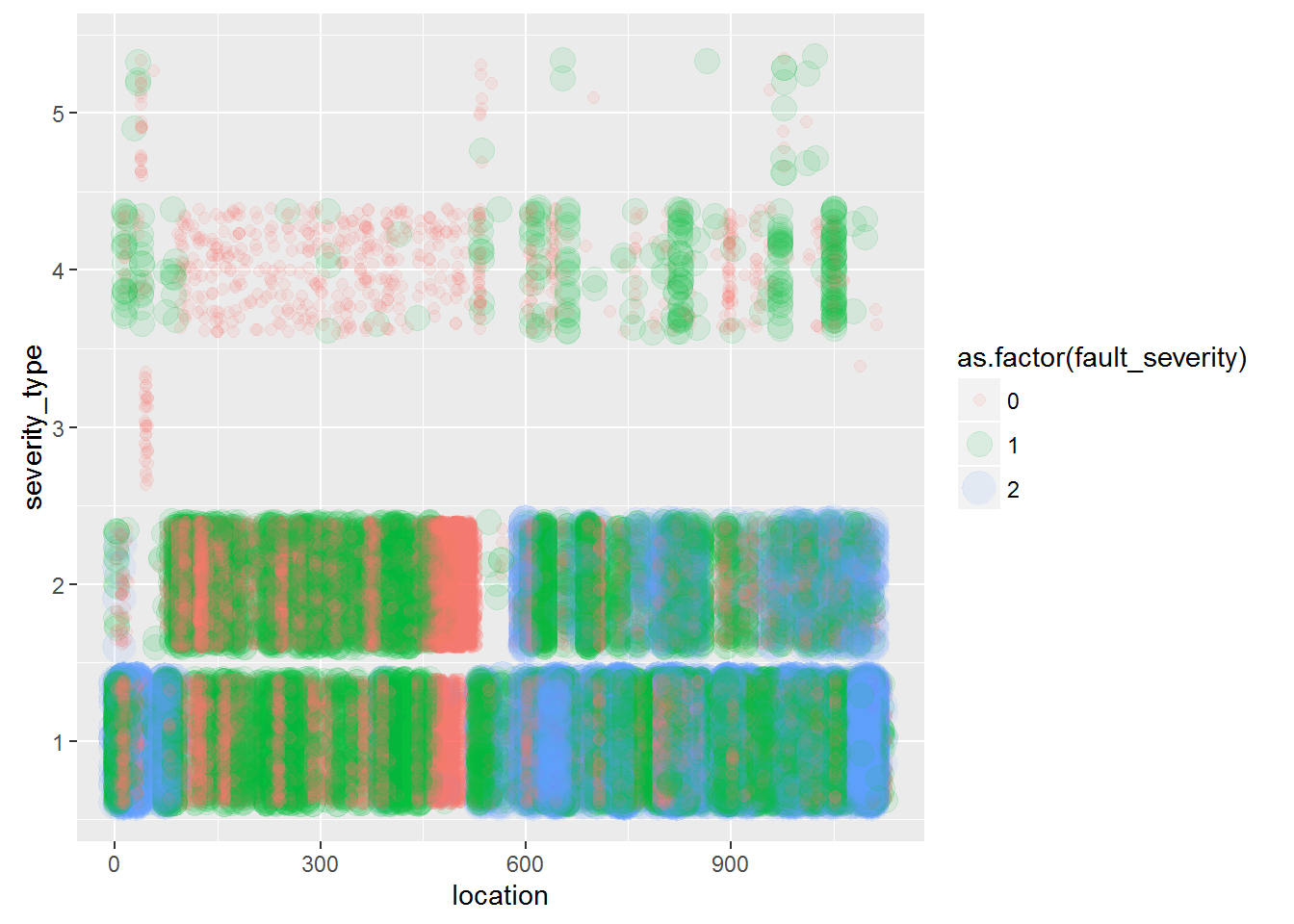
#### Scatter plot : other significant comparisons

Scatter plot of other datasets to explore relationship between log\_feature, resource\_type, event\_type, severity\_type events.

The following scatter plots are all elements viewing with location as base:







**Comment:**

The question is if there is a relation of these log\_features with location.

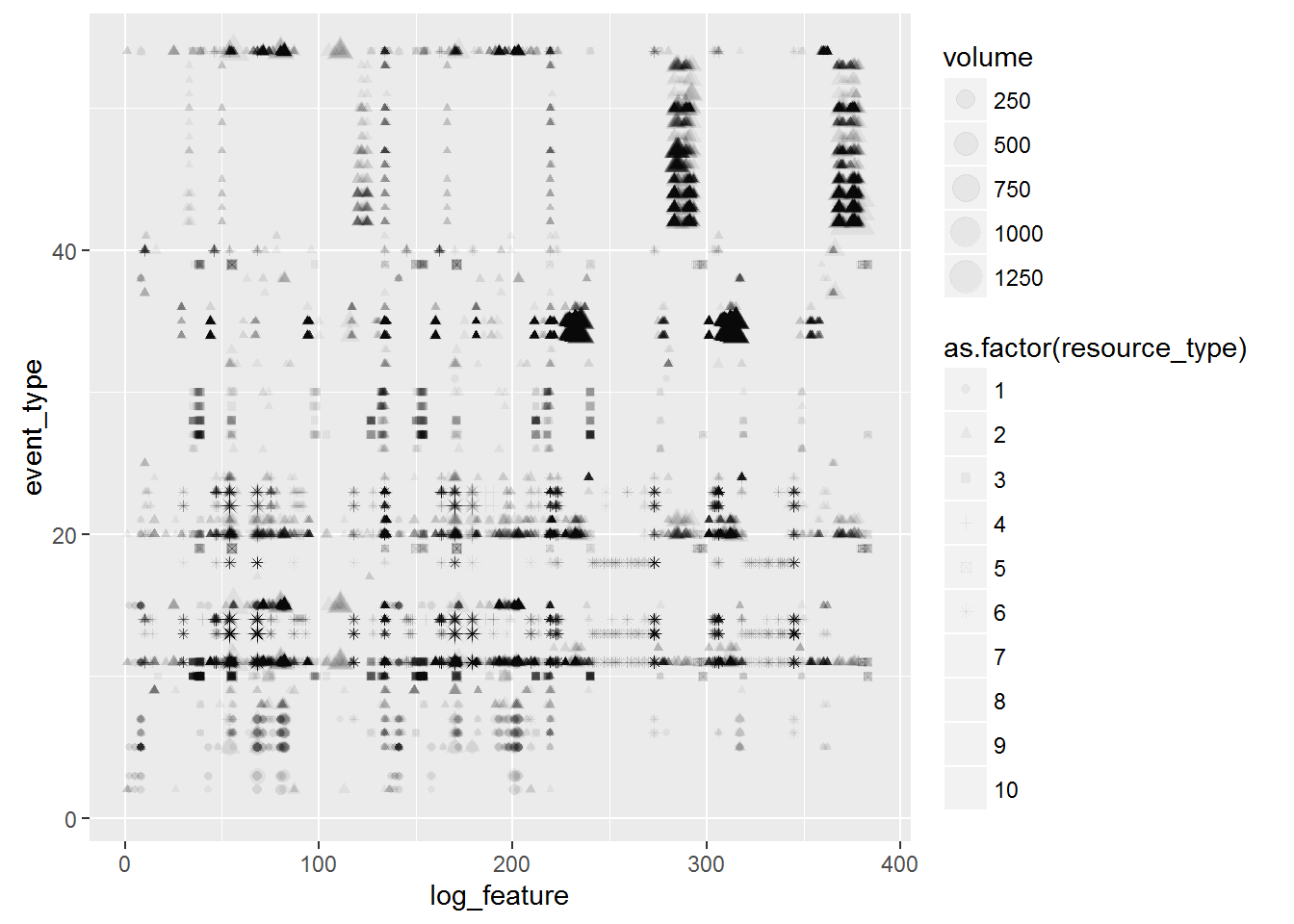
In the 1st graph, Log features appear distinctly at grouped location numbers, showing where the features regularly appear like a map.

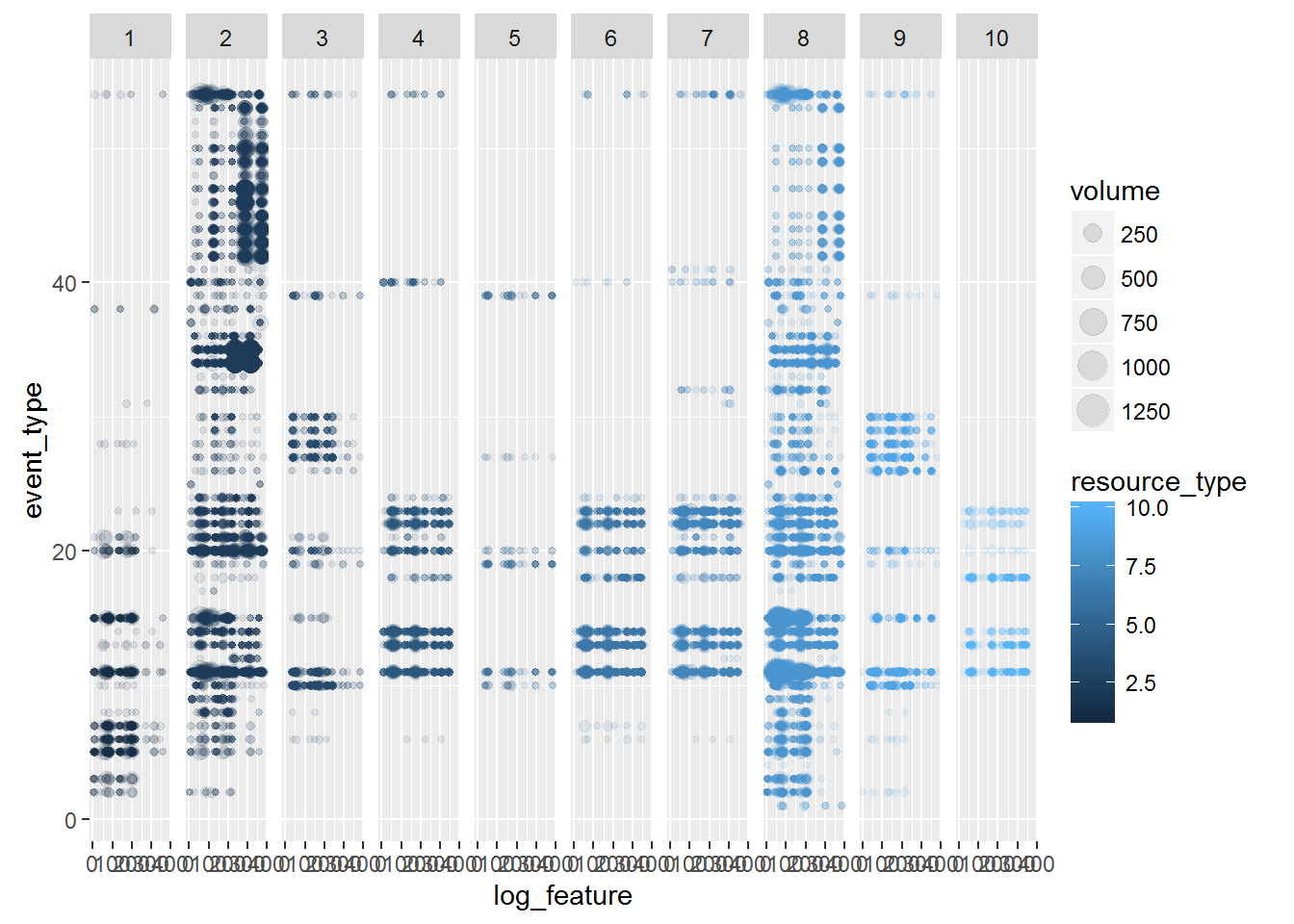
With resources, there is clustering on certain resource type by location number

In the 2nd graph down, noting the dataset records faults, it is unclear whether all locations have all resource types stationed, but some location's equipment may be more prone to fault then others, this may be due to the location's position in a high traffic hub or that there particular resource is due for replacement.

In the 3rd graph down, we can tell that by location error features logged correspond to certain resource type.

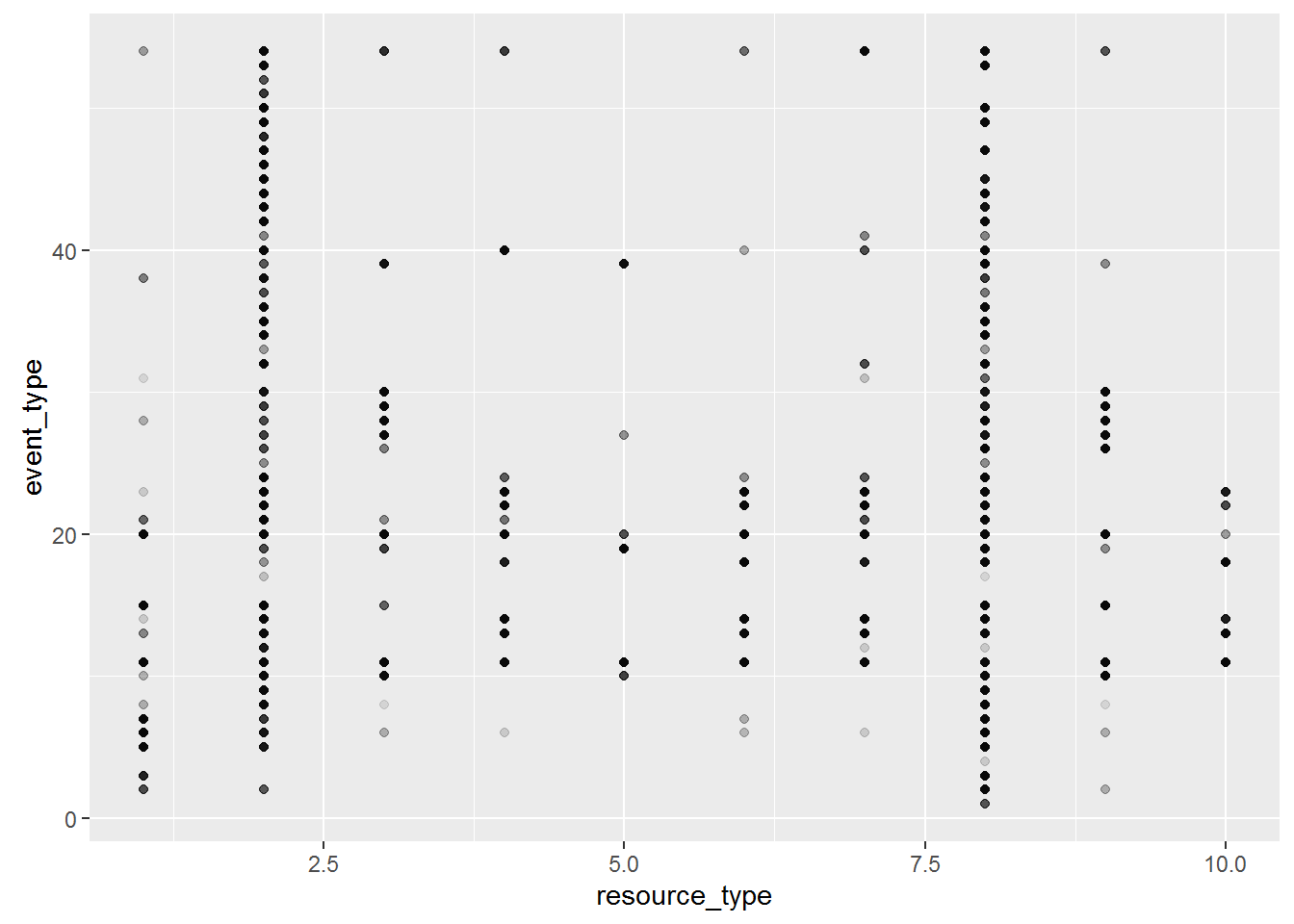
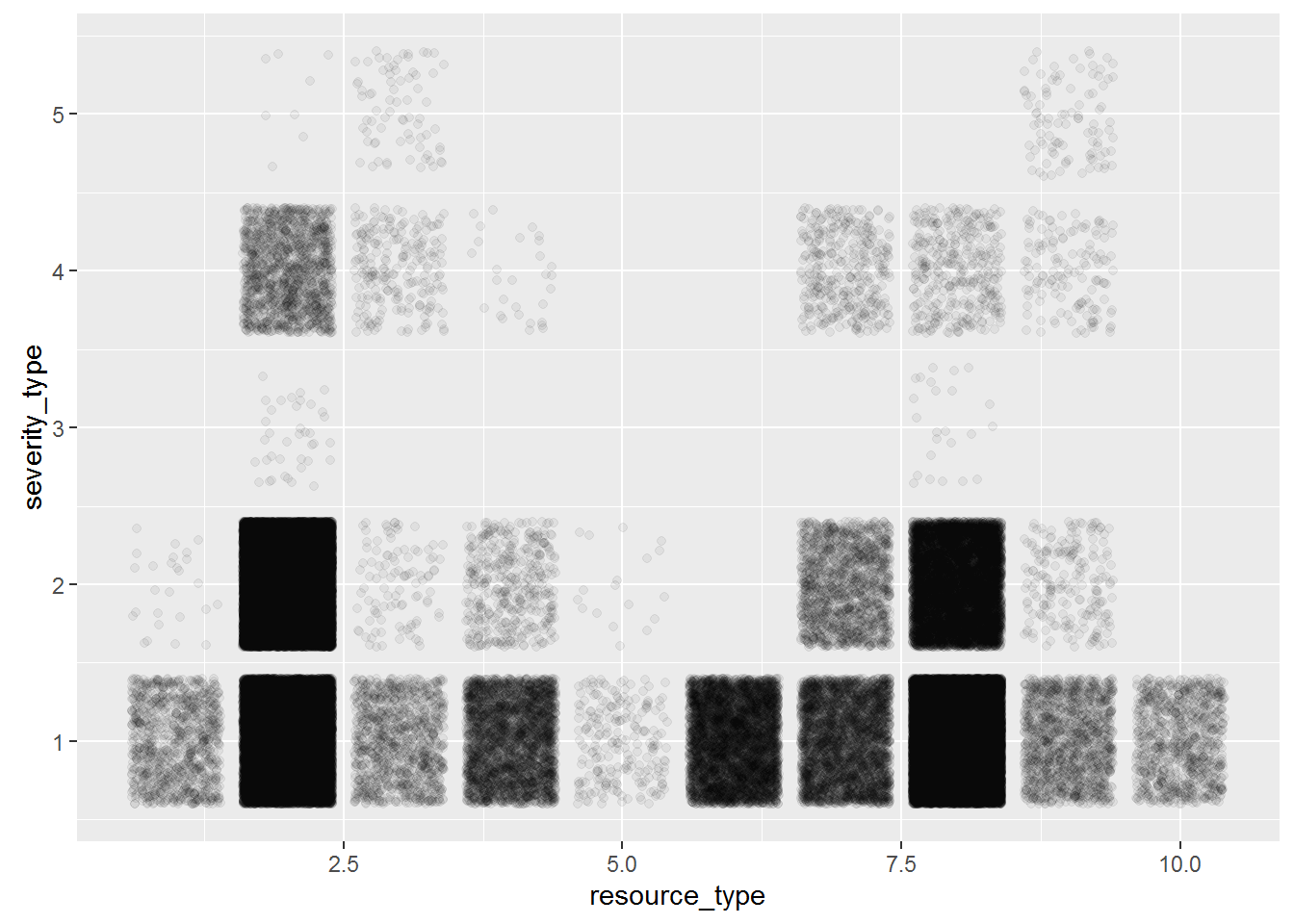
The following 2 colour graphs using colour legend to reference to resource type which has similar patterns on event and log\_feature drawing that they have high correlation.

The following scatter plots looks at the relationship between event\_type and log\_feature with sized by volume and differentiated by resource type:

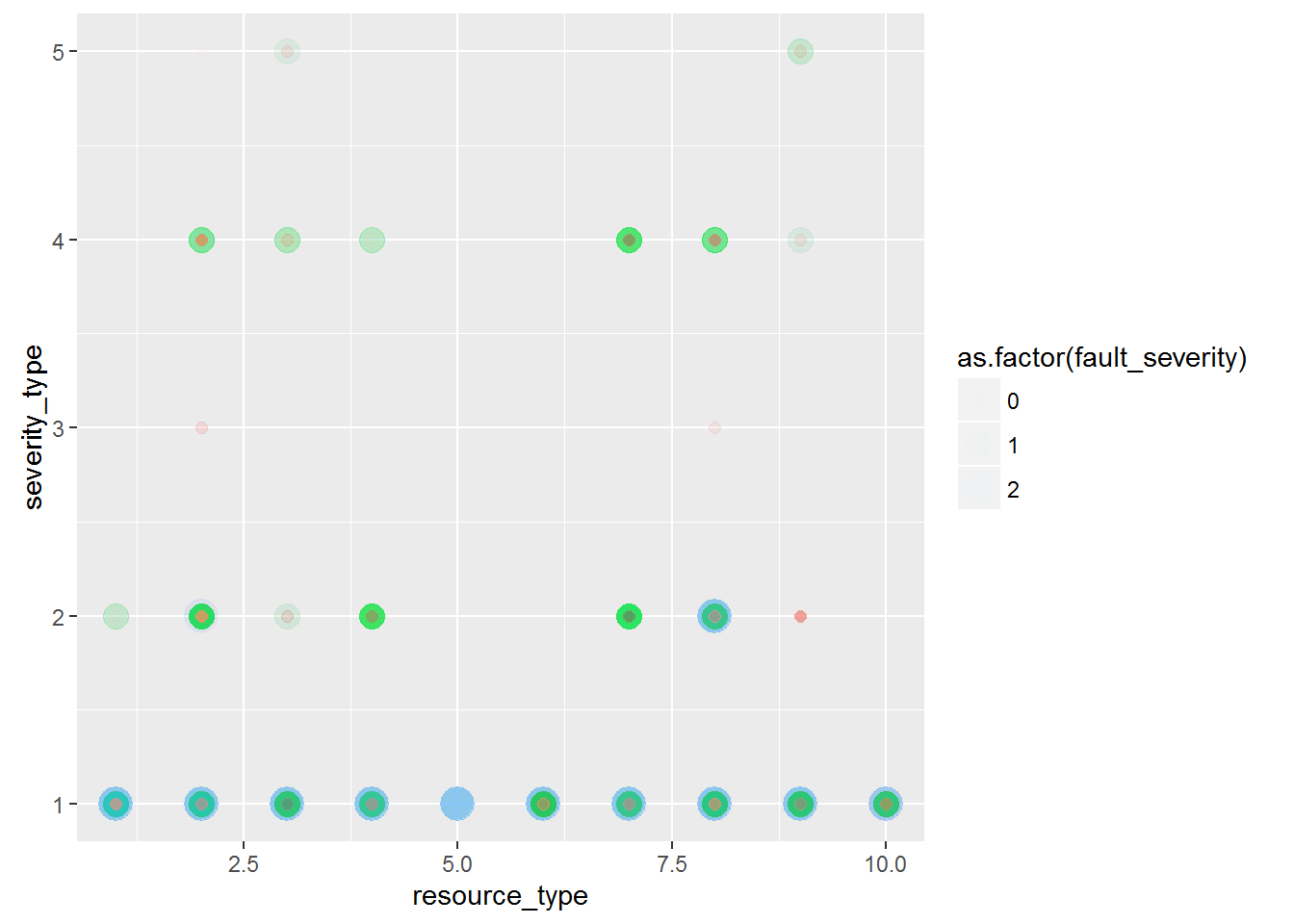
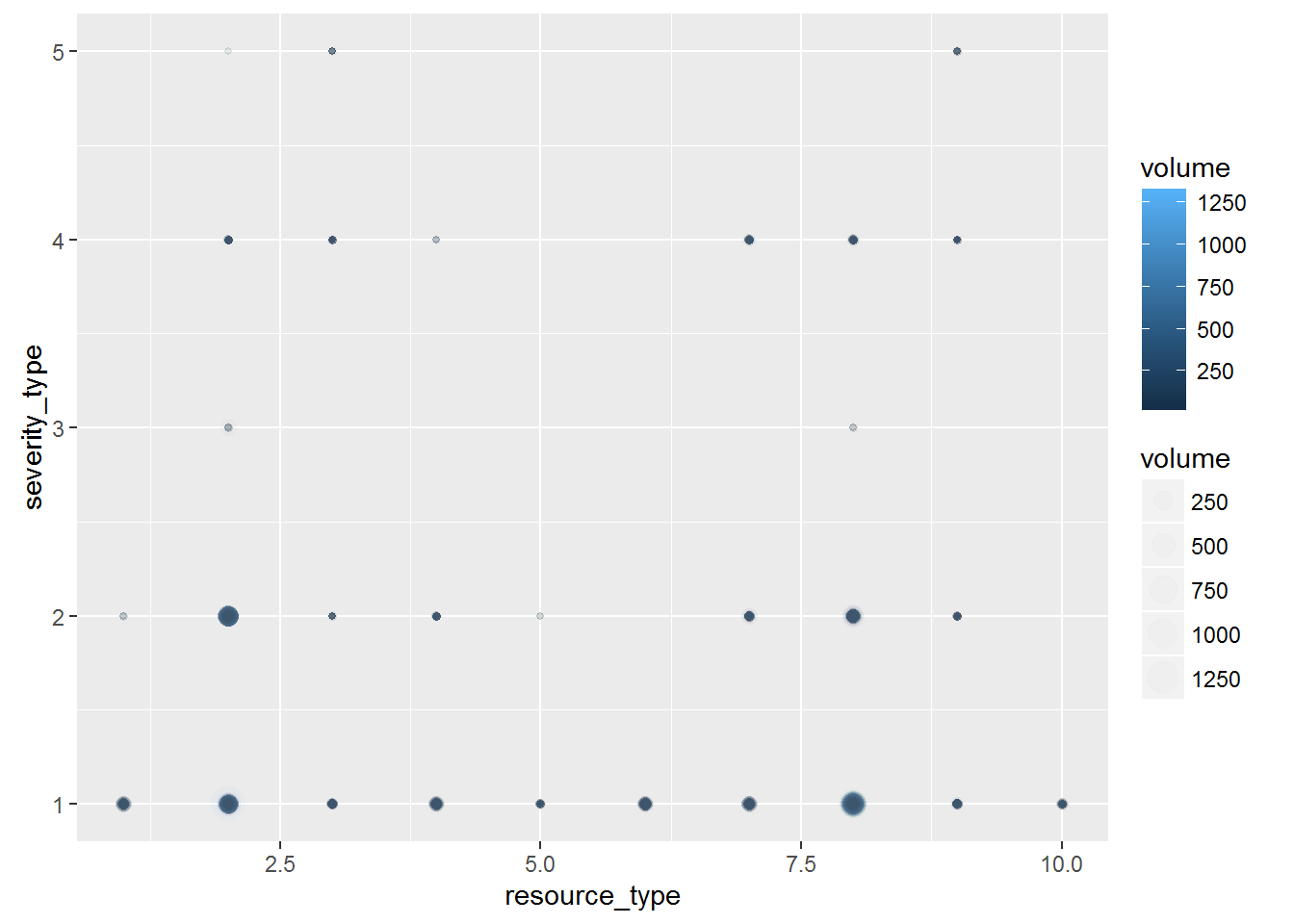


There are some event types carried over to multiple log\_feature, with various event types have varying degrees of log\_feature volumes, hence there may not be any correction there, certain fault events are more highly probable with particular resources as evidence of clusters. Resources 4 & 6 have very similar patterns.

The following scatter plots seeks relationship between event\_type, severity\_type and resource type to see if there are common behaviour patterns.



**Comment:** Some resources share the same event\_type. The plot further shows that resource\_type is a discrete variable of range 10. Each resource type also has a distinct pattern regarding severity type.

Lastly, as found previous that each resource\_type displays similar behaviours then from this a study of the correlation between fault\_severity and volume to understand if volume has an impact on faulty\_severity levels.

**Comment:** By the sizes exhibited in the graphs, the conditions they appear at is the same but the intensity has a slight relationship.

### Initial Findings

The fault\_severity and volume reported seems to be highly correlated and looks to be descriptors from resource’s fault event.

This is because each resource type seems to adopt with particular fault events and these faulty events are also sprung from particular log\_features. Thus the data tells us which log is has certain circumstances and issues develop fault.

Since each resource\_type has particular locations it often logs issues with, it is under these certain events that a prediction can be made regarding fault\_severity. There are 4 distinct characters arising from the log features where each has different volume demands at different events. Some resource types require regular attention yet the has little impact on the network.

Meanwhile severity\_types are selective depending on the type of resource, with types 1 & 2 yielding higher probability to highly network disruption.

The trend found is that towards fault\_severity, volume and severity is deterministic to it. Meanwhile with location, particular log\_features describe resource\_type

To determine fault\_severity = 2 the following events gives a high chance of outcome:

1. Location > resource\_type >log\_feature / volume > severity\_type (1 or 2)

or

2. Resource\_type = 5

### Approach to the problem

From this analysis, different combinations will present different fault\_severity predictions. The data is highly structural with many discrete variables.

The approach to this problem is to determine the different event combinations and work out the probability to predict the fault\_severity. By category and walking through the tree to obtain an estimate.

***Steps***

1. Define the resource types logged at each location.
2. Make another table using all datasets but the train table to understand how feature or particular event type can derive which resource is at fault thus if able to reveal severity level. This then uses all values with no location log attached.
3. Determine the probability levels of fault\_severity to resource/location to conclude a prediction.
4. Use Random Tree or XGBoost to create the model.
5. Test the model using the test data.

### Appendix: Rmd Coding

```{r }

#loading csv files into dataframes named - train, logfeat, resourcetype, eventtype, sevtype, test

train <-

read.csv("~/Foundations/DataSources/Telstra/train.csv", stringsAsFactors = FALSE ) %>% tbl\_df

log\_feature <-

read.csv2("~/Foundations/DataSources/Telstra/log\_feature.csv", stringsAsFactors = FALSE) %>% tbl\_df

resource\_type <-

read.csv("~/Foundations/DataSources/Telstra/resource\_type.csv", stringsAsFactors = FALSE) %>% tbl\_df

event\_type <-

read.csv("~/Foundations/DataSources/Telstra/event\_type.csv", stringsAsFactors = FALSE) %>% tbl\_df

sev\_type <-

read.csv("~/Foundations/DataSources/Telstra/severity\_type.csv", stringsAsFactors = FALSE ) %>% tbl\_df

test <-

read.csv("~/Foundations/DataSources/Telstra/test.csv", stringsAsFactors = FALSE ) %>% tbl\_df

```

#####Tidy log\_feature

```{r }

str(log\_feature)

log\_feature <-

log\_feature %>%

separate(id.log\_feature.volume , c("id", "log\_feature", "volume"), sep = ",")

```

#####Clean types - log\_feature

```{r }

str(log\_feature) # view Structure

summary(log\_feature) # Check for missing values

log\_feature$log\_feature <- sub("feature ", "", log\_feature$log\_feature)

log\_feature$id <- as.integer(log\_feature$id)

log\_feature$volume <- as.integer(log\_feature$volume)

log\_feature$log\_feature <- as.integer(log\_feature$log\_feature)

head(log\_feature)

```

#####Clean types - event\_type

```{r }

str(event\_type)

summary(log\_feature)

event\_type$event\_type <- sub("event\_type ", "", event\_type$event\_type)

event\_type$event\_type <- as.integer(event\_type$event\_type)

head(event\_type)

```

#####Clean types - resource\_type

```{r }

str(resource\_type)

resource\_type$resource\_type <- sub("resource\_type ","", resource\_type$resource\_type)

resource\_type$resource\_type <- as.integer(resource\_type$resource\_type)

head(resource\_type)

```

#####Clean types - sev\_type

```{r }

str(sev\_type)

sev\_type$severity\_type <- sub("severity\_type ", "", sev\_type$severity\_type)

sev\_type$severity\_type <- as.integer(sev\_type$severity\_type)

head(sev\_type)

```

#####Clean types - train

```{r }

str(train)

train$location <- sub("location ","",train$location)

train$location <- as.integer(train$location)

train$fault\_severity <- as.integer(train$fault\_severity)

head(train)

```

#####Clean types - test

```{r }

str(test)

test$location <- sub("location ","",test$location)

test$location <- as.integer(test$location)

head(test)

```

Order tables to prepare Joining

```{r }

arrange(train, id)

arrange(log\_feature, id)

arrange(resource\_type, id)

arrange(event\_type, id)

arrange(sev\_type, id)

arrange(test, id)

```

Full join

``` {r }

network <- full\_join(train, log\_feature, by = c("id"="id"))

network <- full\_join(network, resource\_type, by = c("id"="id"))

network <- full\_join(network, event\_type, by = c("id"="id"))

network <- full\_join(network, sev\_type, by = c("id"="id"))

head(network)

summary(network) # There are some logs with no location details

str(network)

#view na entries

network %>% filter(is.na(fault\_severity)) %>% head(10)

```

###Preliminary exploration

Firstly, let's examine each dataset

#####train

```{r error= FALSE, warning= FALSE}

## frequency at each location

ggplot(train, aes(x = location)) + geom\_histogram(bins = 100)

ggplot(network, aes(x = location)) + geom\_histogram(bins = 100)

## Removed 84584 rows containing non-finite values (stat\_bin).

##-Conclusion: some ids from joining other tables together have missing logs.

##-Some locations have more frequent fault reporting (any correlation with resource type?)

## relationship of location with id

ggplot(train, aes(x = id, y = location, col = as.factor(fault\_severity), size = as.factor(fault\_severity) )) + geom\_point(alpha = 0.3)

##-There is a pattern of faults to have the same severity type hence the linear repetition across the id scale

##ggplot(train, aes(x = id, y = location)) + geom\_point() + facet\_grid( . ~ faulty\_severity)

##Error in layout\_base(data, cols, drop = drop) :

## At least one layer must contain all variables used for facetting

##relationship of feature with fault\_severity

ggplot(train, aes(x = location, y = fault\_severity, col = as.factor(fault\_severity), size = as.factor(fault\_severity) )) + geom\_point(alpha = 0.3)

```

#####log\_feature

``` {r error= FALSE, warning= FALSE}

## frequency at each location

ggplot(log\_feature, aes(x = log\_feature, y = volume, col = log\_feature)) + geom\_point(alpha = 0.2) + coord\_cartesian(ylim = c(0, 1000)) # removed outlier

##-The Volume variable here is a continuous measure with some log\_feature more frequent than others

## relationship of log\_feature with id

ggplot(log\_feature, aes(x = id, y = log\_feature, size = volume)) + geom\_point(alpha = 0.05)

#-There are no missing values in this dataset, but reveals some common error log\_features logged regularly

## Volume of the faults

volume\_above10 <- log\_feature %>% filter(volume > 10)

ggplot(volume\_above10, aes(x = id, y = log\_feature, size = volume)) + geom\_point(alpha = 0.05)

##-The Volume depicted is fairly uniform to the type of feature the fault being recorded with, showing a uniform match with the fault feature recorded

##-Question is of there is a relation of these log\_features with location.

ggplot(network, aes(x = location, y = log\_feature, size = volume)) + geom\_jitter(alpha = 0.05)

##-Log features appear distinctly at grouped location numbers, showing where the features regularly appear like a map

##relationship of log\_feature with fault\_severity

ggplot(network, aes(x = log\_feature, y = fault\_severity, col = as.factor(fault\_severity), size = as.factor(fault\_severity) )) + geom\_point(alpha = 0.3)

##relationship of log\_feature X location with fault\_severity

ggplot(network, aes(x = location, y = log\_feature, col = as.factor(fault\_severity))) + geom\_point(alpha = 0.1) + facet\_grid( . ~ resource\_type)

```

#####resource\_type

``` {r error= FALSE, warning= FALSE}

## frequency at each location

ggplot(resource\_type, aes(x= resource\_type)) + geom\_histogram(bins = 100)

##There are 10 distinct resource types

## relationship of resource\_type with id

ggplot(resource\_type, aes(x = id, y = resource\_type)) + geom\_point(alpha = 0.2)

##-Shows some resource types are more likely to be logged more frequently

## relationship of resource\_type with location

ggplot(network, aes(x = location, y = resource\_type)) + geom\_point(alpha = 0.2)

ggplot(network, aes(x = location, y = resource\_type)) + geom\_jitter(alpha = 0.2)

##-In comparing the locations, there is clustering on certain resource type by location number

##-As the dataset records errors, it is unclear whether all locations have all resource types

##-but some location's equipment may be more prone to fault then others, this may be due to the

##-location's position in a high traffic hub or that there particular resource is due for replacement

## relationship of resource\_type with log\_feature, as the resource type is a discrete variable

ggplot(network, aes(x = location, y = log\_feature, col= as.factor(resource\_type)) ) + geom\_jitter(alpha = 0.05)

##-Here we can tell that by location, error features logged correspond to certain resource type

##-at that location, with each resource having distinct log features.

##-Question here will be is there any correlation of the resource type, log feature to

##-severity\_type to produce the level of fault\_severity.

##-In plain english this maybe the pinpoint of a particular machine that location that

##-works as a main artery of the network to publish repair urgency.

##- "location" > "resource" > "log\_feature" > "Volume" ~ fault occurance

## compare resource\_type with fault\_severity

ggplot(network, aes(x = resource\_type, y = fault\_severity)) + geom\_jitter(alpha = 0.05)

```

#####event\_type

``` {r error= FALSE, warning= FALSE}

## frequency at each location

ggplot(event\_type, aes(event\_type)) + geom\_histogram(bins = 100)

ggplot(network, aes(x = event\_type)) + geom\_histogram(bins = 100) # There are no missing values

##-There appears to be distinct but also popular events.

##-Question is, do these events correlation with particular resource (machine) that can

##-interpret level of fault severity?

## relationship of event\_type with id

ggplot(event\_type, aes(x = id, y = event\_type)) + geom\_point(alpha = 0.05)

ggplot(event\_type, aes(x = id, y = event\_type)) + geom\_jitter(alpha = 0.05)

##-There are certain event types that are frequently occuring or being recorded. Reviewing with jitter does not show much gaps in frequency

## relationship of event\_type with resource\_type

ggplot(network, aes(x = resource\_type, y = event\_type)) + geom\_point(alpha = 0.05)

## Some resource share the same event\_type but the plot further shows that resource\_type is a discrete variable of range 10.

## relationship of event\_type with log\_feature where Volume is displayed differentiated by resource\_type

ggplot(network, aes(x = log\_feature, y = event\_type, size = volume, shape = as.factor(resource\_type))) + geom\_point(alpha = 0.05)

## The shape palette can deal with a maximum of 6 discrete values because more than 6 becomes

## difficult to discriminate; you have 10. Consider specifying shapes manually if you must have

## them.

##-There are some event types carried over to multiple log\_feature, with various event types

##-have varying degrees of log\_feature volumes, hence there may not be any correction there,

##-certain fault events are more highly probable with particular resources as evidence of clusters

##- At this point, volume in reference with log\_feature may be telling of the quantity of equipment

##- and not severity

## further look at relationship of event\_type with log\_feature, sized in Volume, faceted by resource\_type

ggplot(network, aes(x = log\_feature, y = event\_type, size = volume, col = resource\_type)) + geom\_point(alpha = 0.1) + facet\_grid( . ~ resource\_type )

##-Resources 4 & 6 have very similar patterns

## relationship of event\_type with location

ggplot(network, aes(x = location, y = event\_type, col = as.factor(resource\_type))) + geom\_point(alpha = 0.05)

##-There is a similarity of pattern with "locationXresource\_type" and "locationXlog\_feature".

##relationship of event\_type with fault\_severity

ggplot(network, aes(x = event\_type, y = fault\_severity, col = as.factor(fault\_severity), size = as.factor(fault\_severity) )) + geom\_point(alpha = 0.3)

```

######severity\_type

``` {r error= FALSE, warning= FALSE}

## frequency at each location

ggplot(sev\_type, aes(severity\_type)) + geom\_histogram(bins = 100)

ggplot(network, aes(x = severity\_type)) + geom\_histogram(bins = 100) # There are no missing values

##-There appears to be 5 distinct types.

## relationship of feature with id

ggplot(sev\_type, aes(x = id, y = severity\_type)) + geom\_jitter(alpha = 0.2)

## The distinct categories have clear proportion clear distributed in the logs

## relationship of feature with id highlighted by fault\_severity

ggplot(network, aes(x = id, y = severity\_type, col = as.factor(fault\_severity), size = as.factor(fault\_severity))) + geom\_jitter(alpha = 0.1)

##-Most severe faults are categorized by severity\_type no.1 .

##-Severity\_type no.3 only attracts fault level of 1 .

##relationship of severity\_type with fault\_severity

ggplot(network, aes(x = severity\_type, y = fault\_severity, col = as.factor(fault\_severity), size = as.factor(fault\_severity) )) + geom\_point(alpha = 0.3)

##-Severity\_type no.3 only attracts fault level of 1 confirmed.

##relationship of feature with location highlighted with fault\_severity

ggplot(network, aes(x = location, y = severity\_type, col = as.factor(fault\_severity), size = as.factor(fault\_severity))) + geom\_jitter(alpha = 0.1)

##-High levels of faults appear at low number locations or locations numbered greater than 500

#relationship of feature with resource\_type

ggplot(network, aes(x = resource\_type, y = severity\_type)) + geom\_jitter(alpha = 0.05)

##relationship of feature with resource\_type with fault\_severity highlights

ggplot(network, aes(x = resource\_type, y = severity\_type, col = as.factor(fault\_severity), size = as.factor(fault\_severity))) + geom\_point(alpha = 0.01)

##-all resources can display the first severity type with high fault level

## compare faulty severity highlights with volume

ggplot(network, aes(x = resource\_type, y = severity\_type, col = volume, size = volume)) + geom\_point(alpha = 0.01)

```

#####Other exploration with fault\_severity

```{r error= FALSE, warning = FALSE}

##relationship of event\_type X location with fault\_severity

ggplot(network, aes(x = location, y = event\_type, col = as.factor(fault\_severity))) + geom\_point(alpha = 0.1) + facet\_grid( . ~ resource\_type)

##relationship of event\_type X log\_feature with fault\_severity

ggplot(network, aes(x = log\_feature, y = event\_type, col = as.factor(fault\_severity))) + geom\_point(alpha = 0.1) + facet\_grid( . ~ resource\_type)

## compare volume with fault\_severity

ggplot(network, aes(x = volume, y = fault\_severity, col = as.factor(fault\_severity))) + geom\_point(alpha = 0.2)

## compare event\_type with fault\_severity

ggplot(network, aes(x = event\_type, y = fault\_severity,col = as.factor(fault\_severity))) + geom\_jitter(alpha = 0.2)

## compare resource\_type with fault\_severity

ggplot(network, aes(x = resource\_type, y = fault\_severity, col = as.factor(fault\_severity), size = as.factor(fault\_severity))) + geom\_jitter(alpha = 0.3)

##-Useful to interpret end prediction outcomes

```