NGAX defect analysis

Silpa Velagapudi

December 27, 2019

Contents

1. **Introduction**:

Analysing 1000 NGAX defects created. Trying to find what kind of defects are more frequent during the course of time.

2. **Data Extraction**: **Pre-requisites: Loading Packages and Libraries**

First, we load all the necessary packages and libraries.

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## Loading required package: caret
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v tibble  2.1.3     v purrr   0.3.2
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
```

Downloaded the defects from the Jira Portal and created new file with the required fields and viewing the first 6 records of the file.

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   Root_cause = col_logical(),
##   Systems_Impacted = col_logical(),
##   Testing_Instructions = col_logical()
## )
```

```
## See spec(...) for full column specifications.
```

```
## # A tibble: 6 x 29
##   Issue_type Status Priority Resolution Affects_version~ Affects_version~
##   <chr>      <chr>  <chr>    <chr>      <chr>            <chr>
## 1 NGAX-6639  Requi~ Medium   <NA>       NGAX-R19.28.0    <NA>
## 2 NGAX-6638  Requi~ Medium   <NA>       NGAX-R19.28.0    <NA>
## 3 NGAX-6637  Requi~ Low      <NA>       AXT-R0.1.6       <NA>
## 4 NGAX-6636  Requi~ Medium   <NA>       NGAX-R19.28.0    <NA>
## 5 NGAX-6635  Requi~ Medium   <NA>       NGAX-R19.28.0    <NA>
## 6 NGAX-6634  Reope~ High     Done       <NA>             <NA>
```

```
## # ... with 23 more variables: Fix_versions1 <chr>, Fix_versions2 <chr>,
## #   Fix_versions3 <chr>, Labels1 <chr>, Labels2 <chr>, Labels3 <chr>,
## #   Labels4 <chr>, Labels5 <chr>, Labels6 <chr>, Labels7 <chr>,
## #   Actual_Result <chr>, Environment <chr>, Epic_Link <chr>,
## #   Failure_Type <chr>, Feature_Channel <chr>, Functional_group <chr>,
## #   Impacts <chr>, Root_cause <lgl>, Systems_Impacted <lgl>, Test_Source <chr>,
## #   Test_Type <chr>, Testing_Instructions <lgl>, Types_of_QA_Needed <chr>
```

3. **Creating key field values**

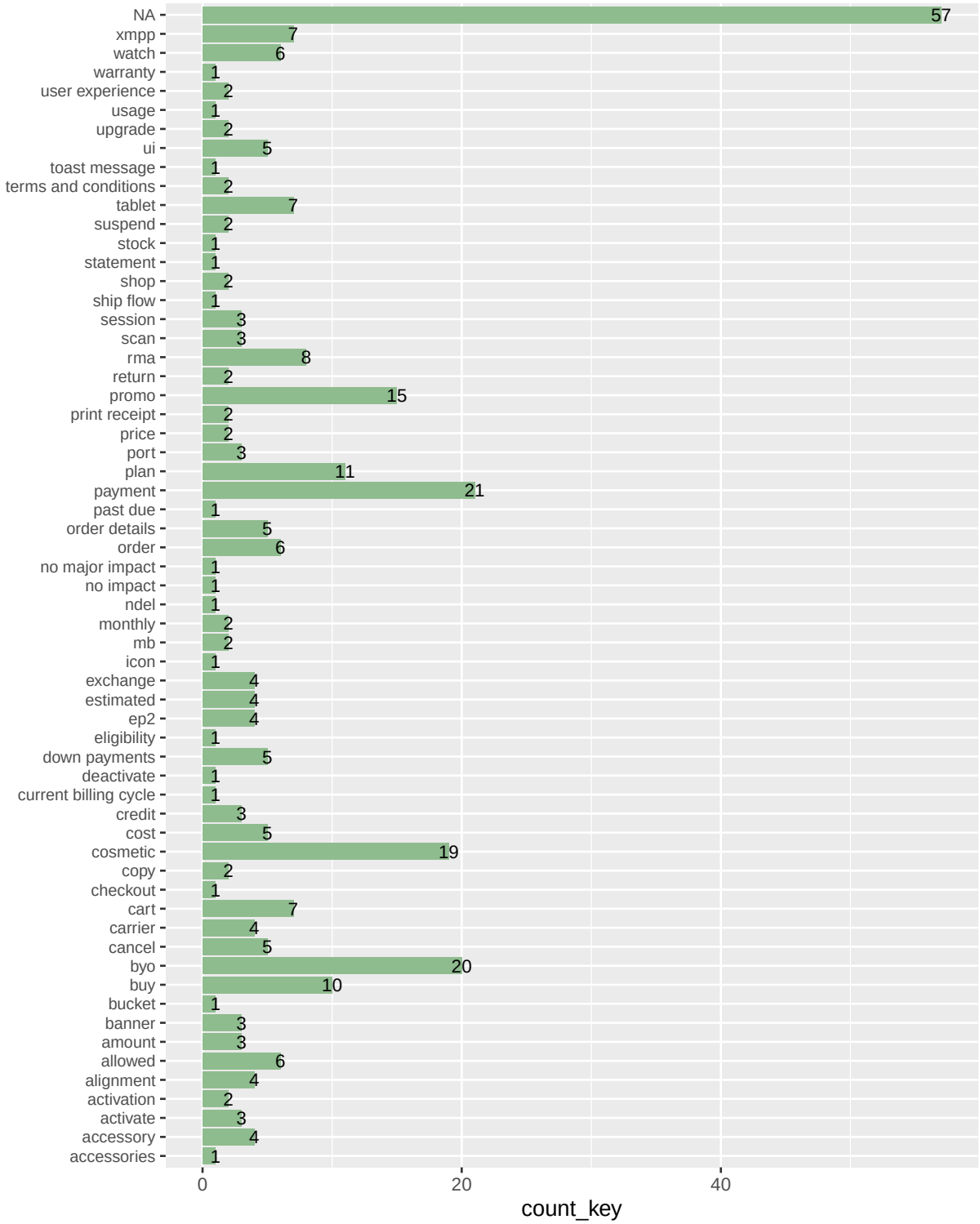
Key_values is created based on the frequently used words from Impact field.

Showing the first 6 records from the file after running the key values and writing the details to the file

```
## # A tibble: 6 x 31
##   Issue_type Status Priority Resolution Affects_version~ Affects_version~
##   <chr>      <chr>  <chr>    <chr>      <chr>            <chr>
## 1 NGAX-6639  Requi~ Medium   <NA>       NGAX-R19.28.0    <NA>
## 2 NGAX-6638  Requi~ Medium   <NA>       NGAX-R19.28.0    <NA>
## 3 NGAX-6637  Requi~ Low      <NA>       AXT-R0.1.6       <NA>
## 4 NGAX-6636  Requi~ Medium   <NA>       NGAX-R19.28.0    <NA>
## 5 NGAX-6635  Requi~ Medium   <NA>       NGAX-R19.28.0    <NA>
## 6 NGAX-6634  Reope~ High     Done       <NA>             <NA>
## # ... with 25 more variables: Fix_versions1 <chr>, Fix_versions2 <chr>,
## #   Fix_versions3 <chr>, Labels1 <chr>, Labels2 <chr>, Labels3 <chr>,
## #   Labels4 <chr>, Labels5 <chr>, Labels6 <chr>, Labels7 <chr>,
## #   Actual_Result <chr>, Environment <chr>, Epic_Link <chr>,
## #   Failure_Type <chr>, Feature_Channel <chr>, Functional_group <chr>,
## #   Impacts <chr>, Root_cause <lgl>, Systems_Impacted <lgl>, Test_Source <chr>,
## #   Test_Type <chr>, Testing_Instructions <lgl>, Types_of_QA_Needed <chr>,
## #   lower_impacts <chr>, key_field[,1] <chr>
```

4. **In-depth Analysis of the Data**:

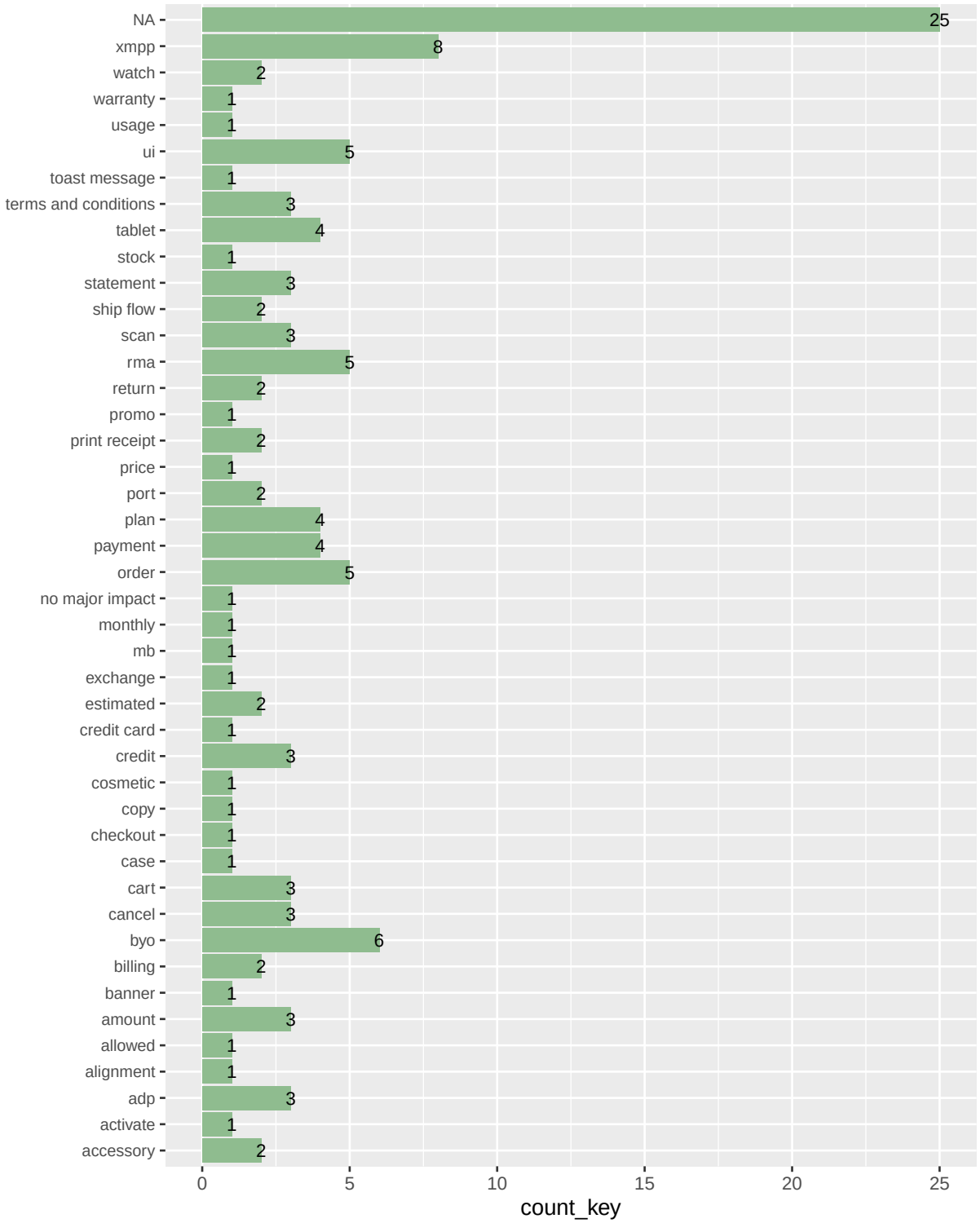
Progression + Integration Below bar graph shows the details of the Progression testing defects in Integration environment.



We can see that we have few key words which are more common during testing like Promo, payment, device issues, cosmetic, BYO. As these are progression test cases we can expect they are related to the stories

created.

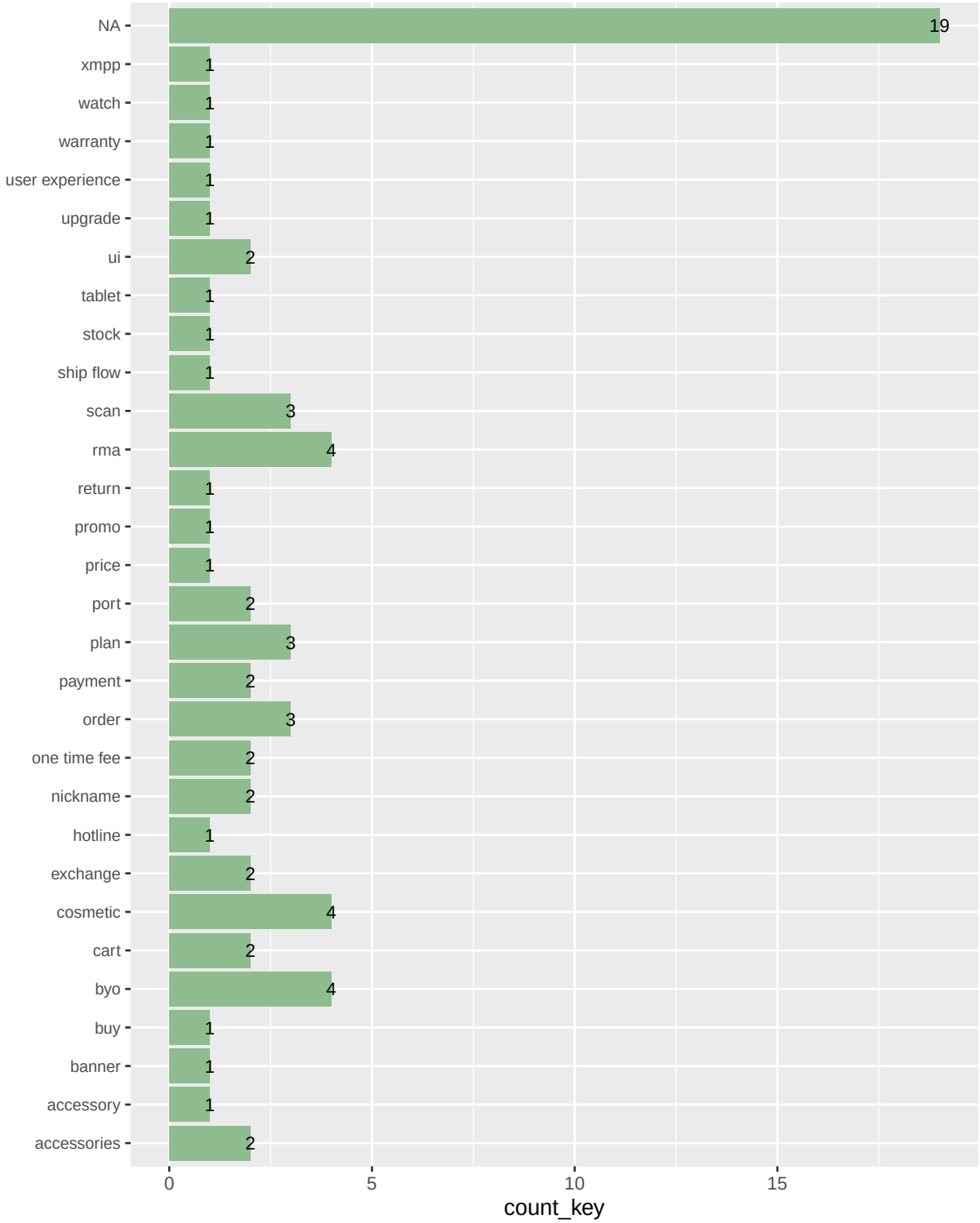
Progression and Staging Below bar graph shows the details of the Progression testing defects in staging environment.



We can see that we have few key words which are more common during testing like Ordering issues, UI issues, device related issues, payment, XMPP and BYOD. As these are progression test cases we can expect

they are related to the stories created.

Progression and Pre-Prod Below bar graph shows the details of the Progression testing defects in staging environment.

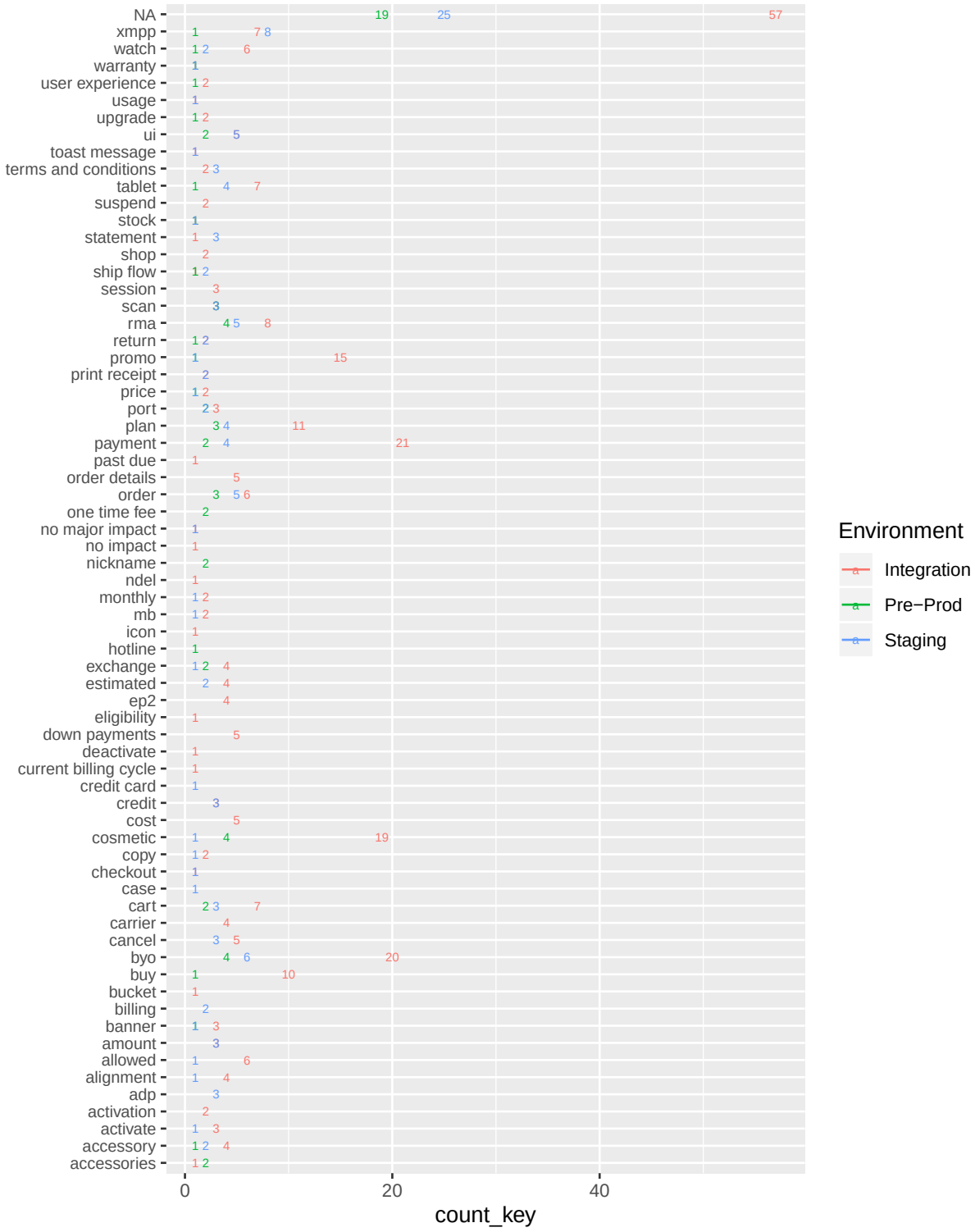


We can see that we have few key words which are more common during testing like RMA, cosmetic, device issue, scan flow and BYOD. We can clearly see that the issues found in pre-prod are same in staging for

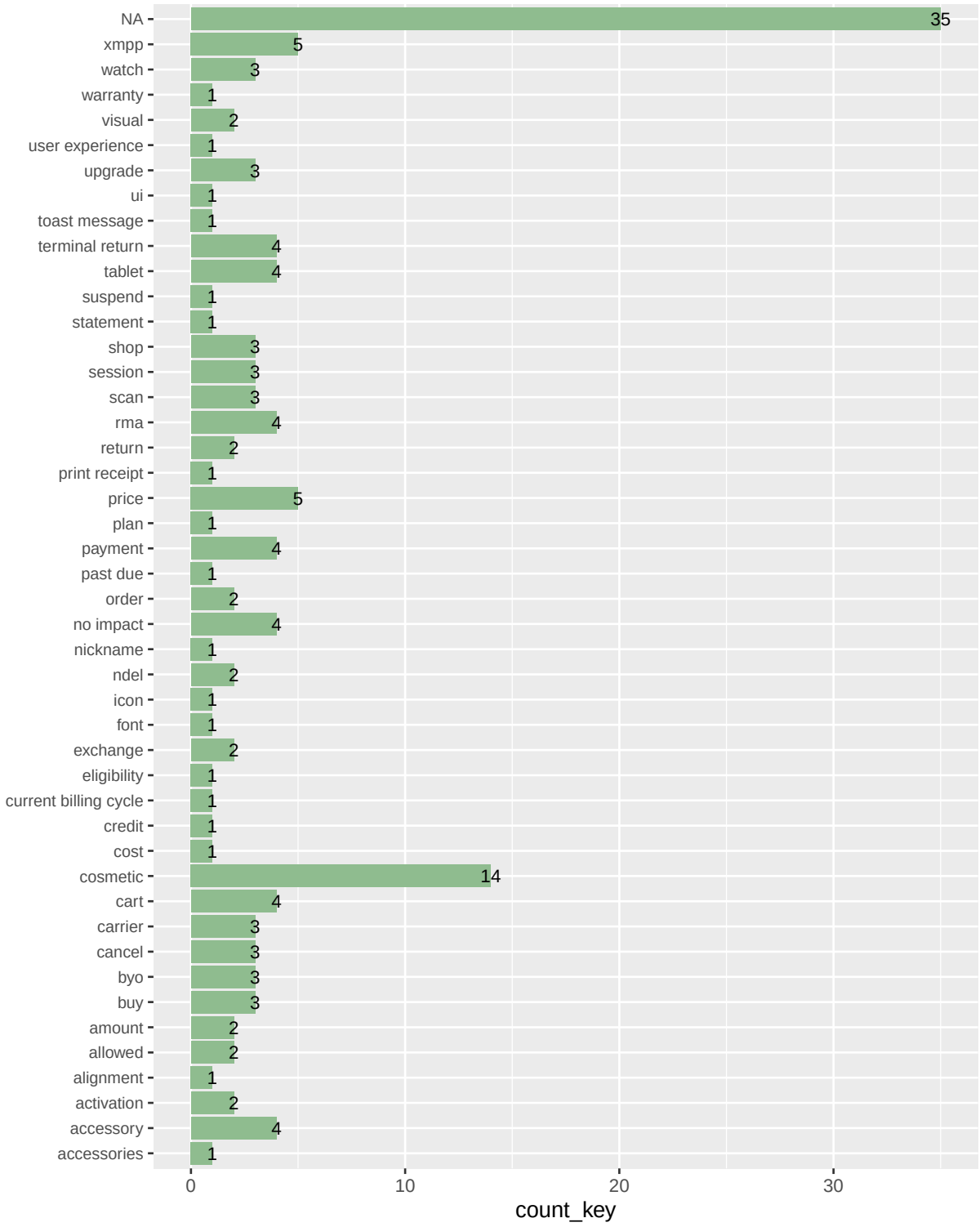
BYOD and device issues. Rest of the defects are different. From this we can say that defects found earlier are not getting repeated in higher environments.

Progression for Integration, Staging and Pre-prod Below graph provide details of the Progression defects in Integration, Staging and Pre-prod environments:

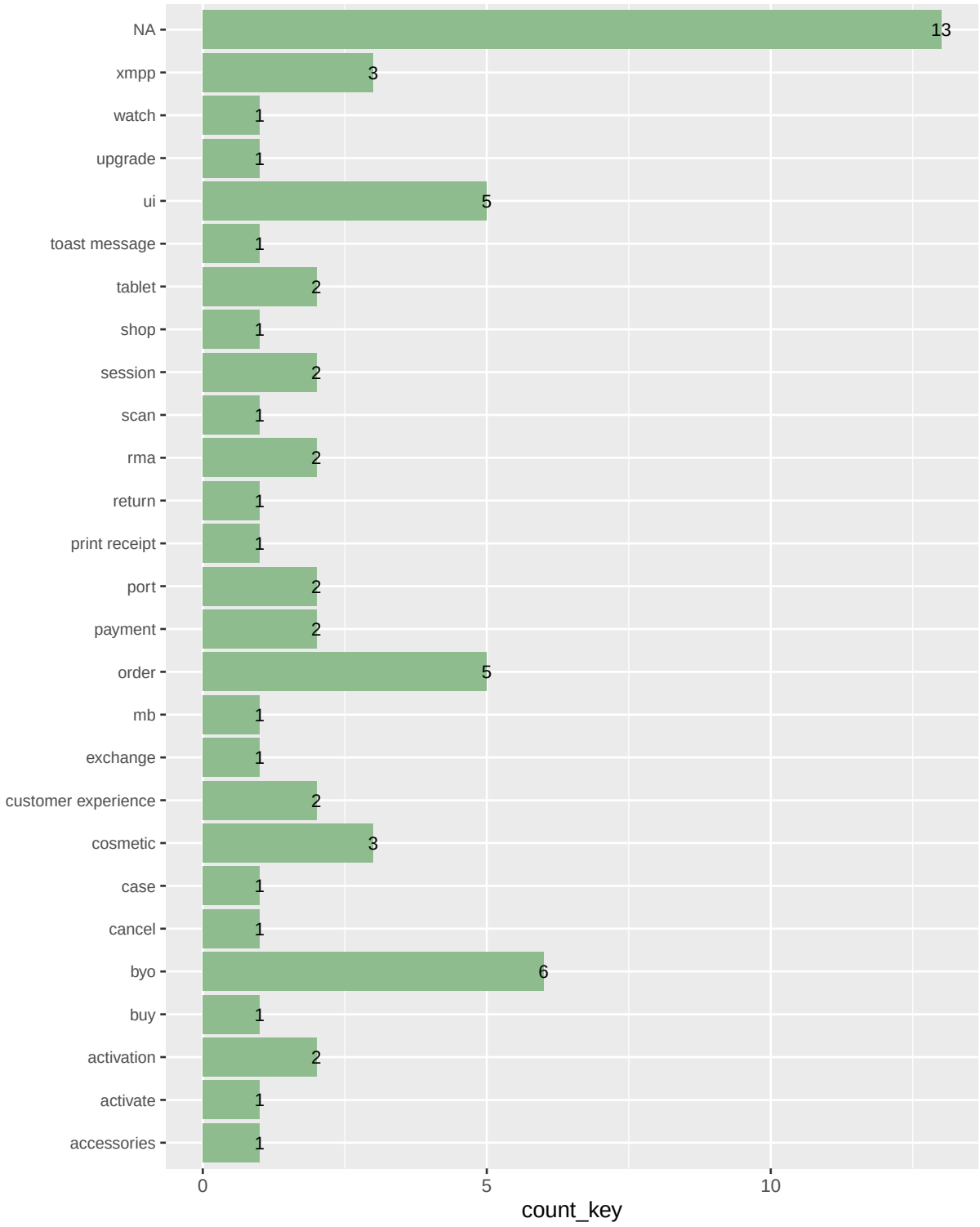
```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```



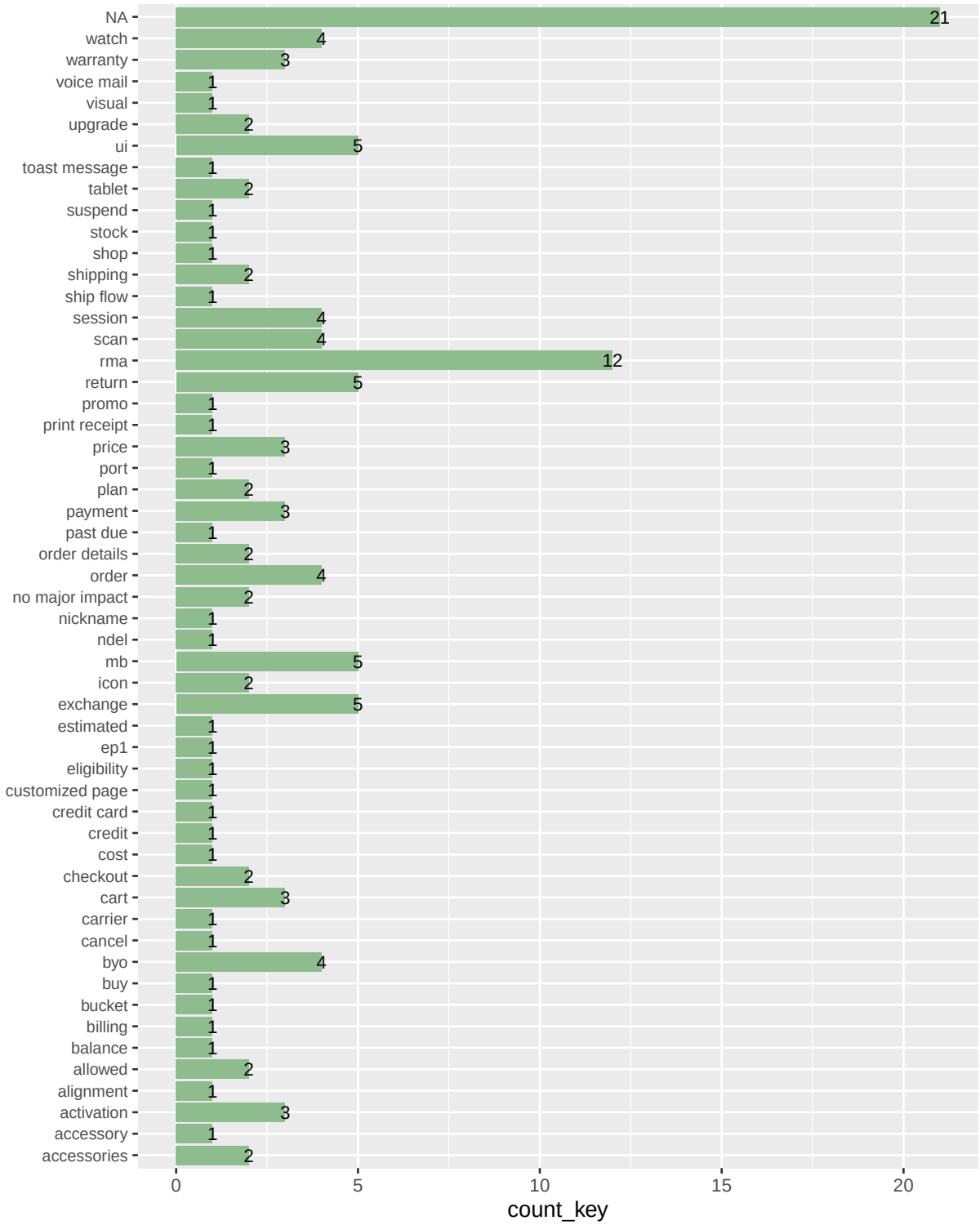
Regression for Integration



Regression for Staging



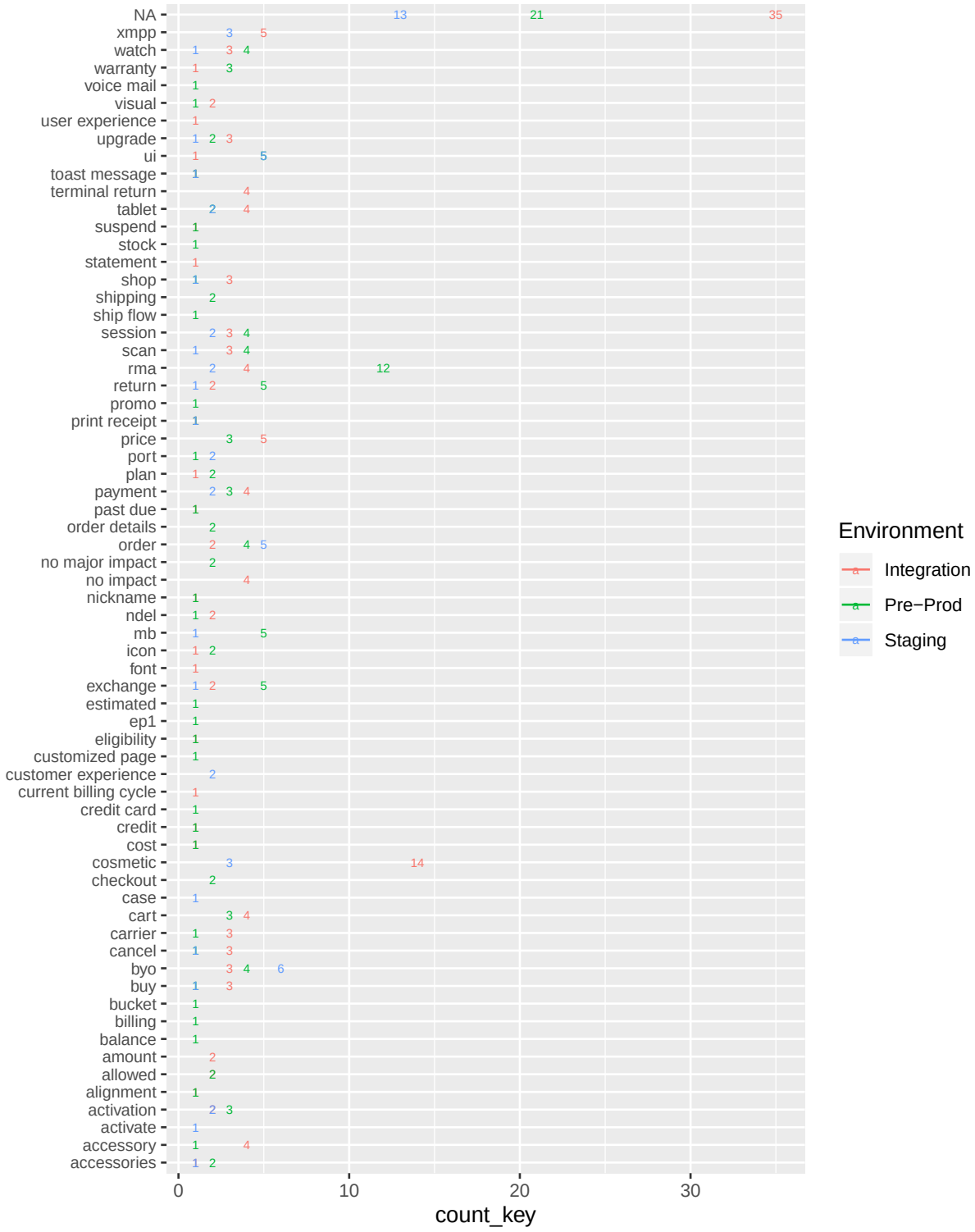
Regression for Pre-Prod



Regression for Integration, Staging and Pre-prod

Below graph provide details of the Progression defects in Integration, Staging and Pre-prod environments:

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```


NA
xmpp
watch
warranty
voice mail
visual
user experience
upgrade
ui
toast message
terminal return
tablet
suspend
stock
statement
shop
shipping
ship flow
session
scan
rma
return
promo
print receipt
price
port
plan
payment
past due
order details
order
no major impact
no impact
nickname
ndel
mb
icon
font
exchange
estimated
ep1
eligibility
customized page
customer experience
current billing cycle
credit card
credit
cost
cosmetic
checkout
case
cart
carrier
cancel
byo
buy
bucket
billing
balance
amount
allowed
alignment
activation
activate
accessory
accessories
0
10
20
30
count_key
Environment
Integration
Pre−Prod
Staging