Analysis of Riipen's New Feature

STA130 TUT0112-Group 112-1

Shannon Simo Komguem, Kai Wang Pong, Eng Sung and JiaQi Zhao

About the data

- Riipen is an education technology company. On November 21, 2018, Riipen launched a "request expiry" feature which requires users to respond to requests within 14 days, or they will expire.

The Data

```
## Observations: 2,526
## Variables: 8
## $ Id                    <dbl> 2, 3, 4, 5, 6, 7,...
## $ `Recipient Id`        <dbl> 1142, 910, 1108, ...
## $ `Actor Id`            <dbl> 18026, 17140, 118...
## $ `Requestable Model`   <chr> "project", "proje...
## $ `Day of Created At`   <chr> "12 April, 2018",...
## $ `Day of Updated At`   <chr> "24 April, 2018",...
## $ `Day of Expired At`   <chr> NA, NA, NA, NA, N...
## $ State                 <chr> "accepted", "acce...
```

Objective

- How did the November 21 launch of request expiry change user behaviour?

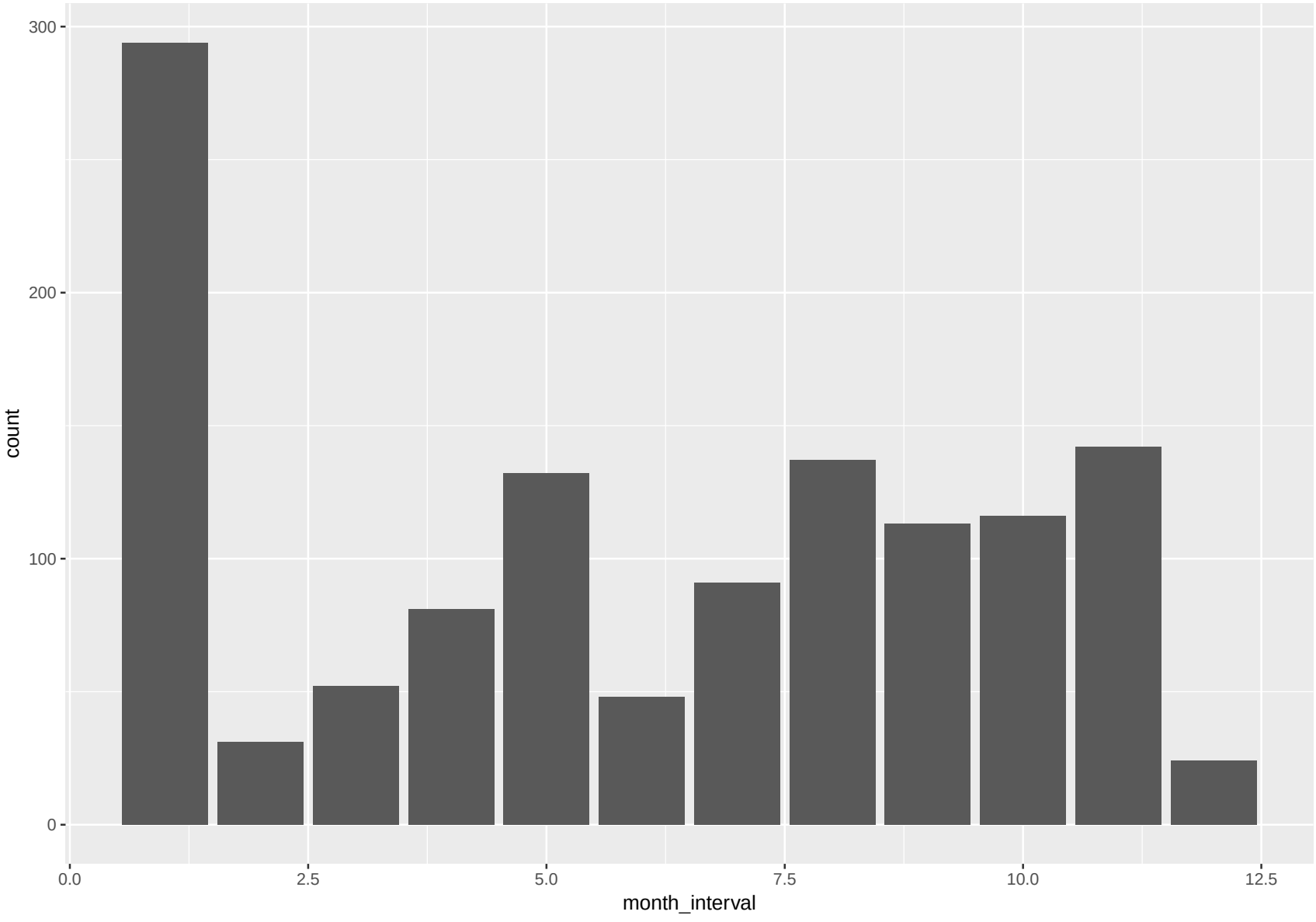
Data Cleaning

- We excluded August 30th, 2018 due to data migration problems.

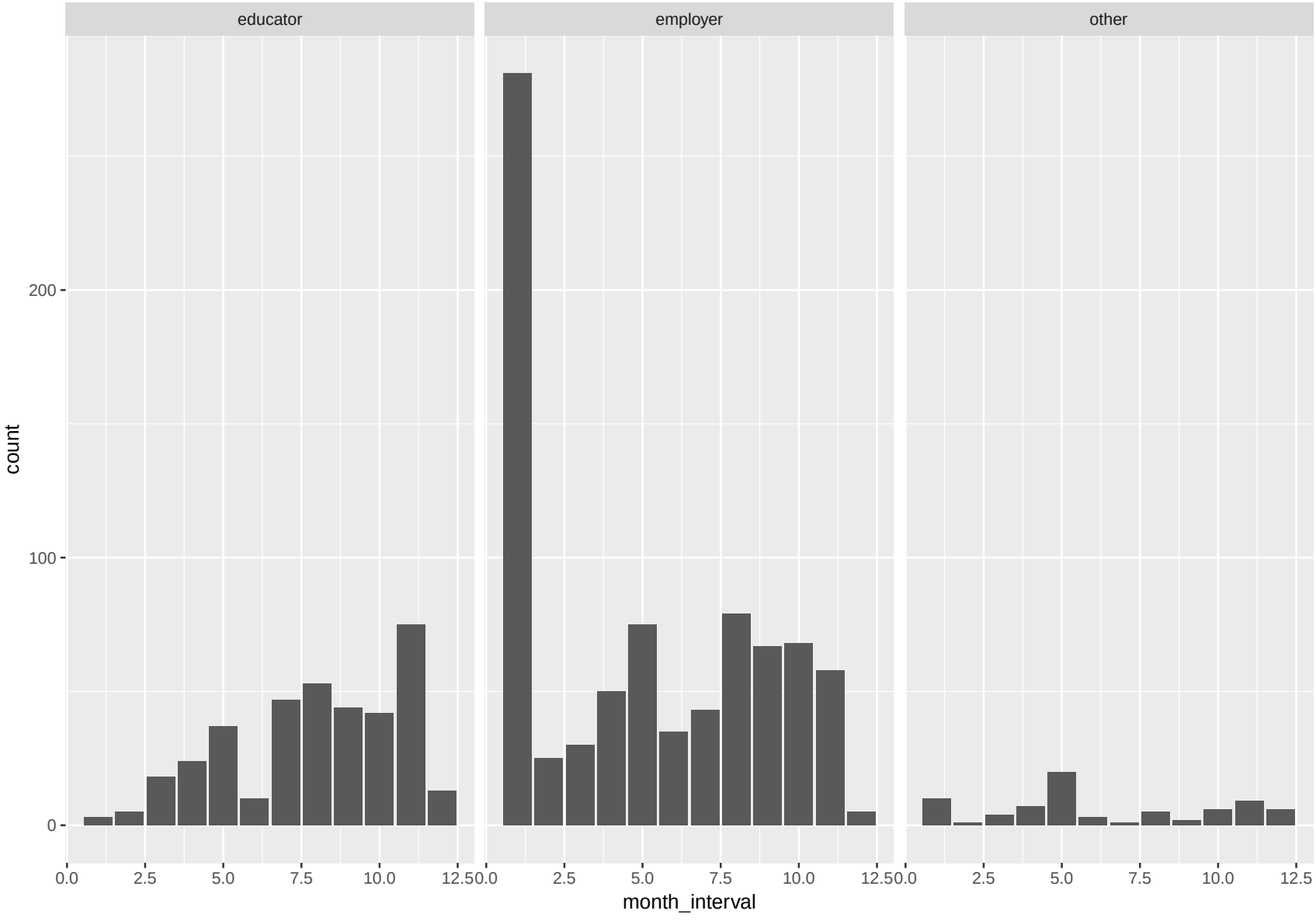
Statistical Assessments

- Volume of Request
- Response Time
- Regression Model: Response Time and Acceptance Rate

Volume of Request



Volume of Request By User Type



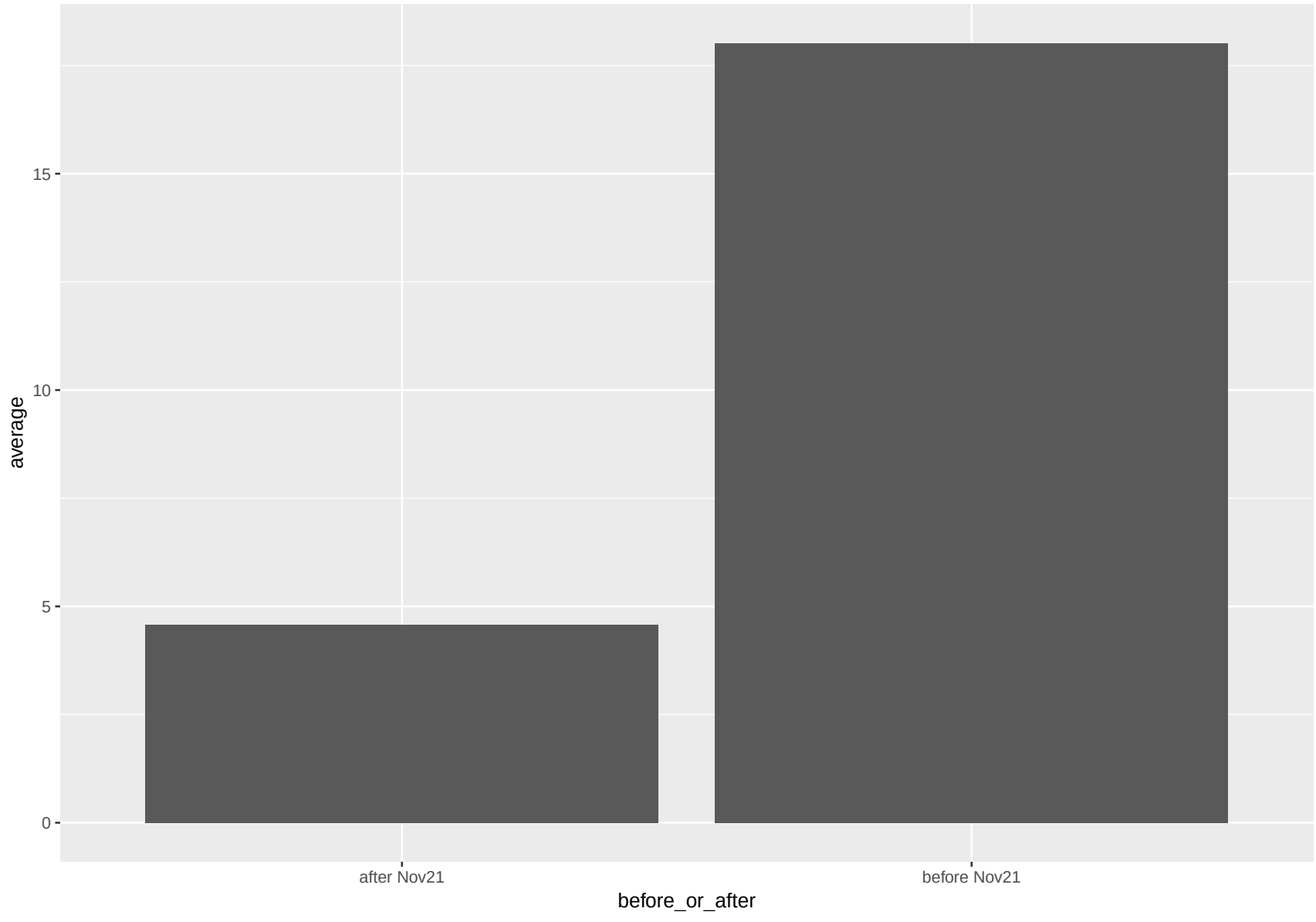
Respone Time - Data Cleaning

- Aug 30
- Pending cases
- Cancelled cases
- Expired cases
- Response Time (in days) = difference of “day of updated” and “day of created”

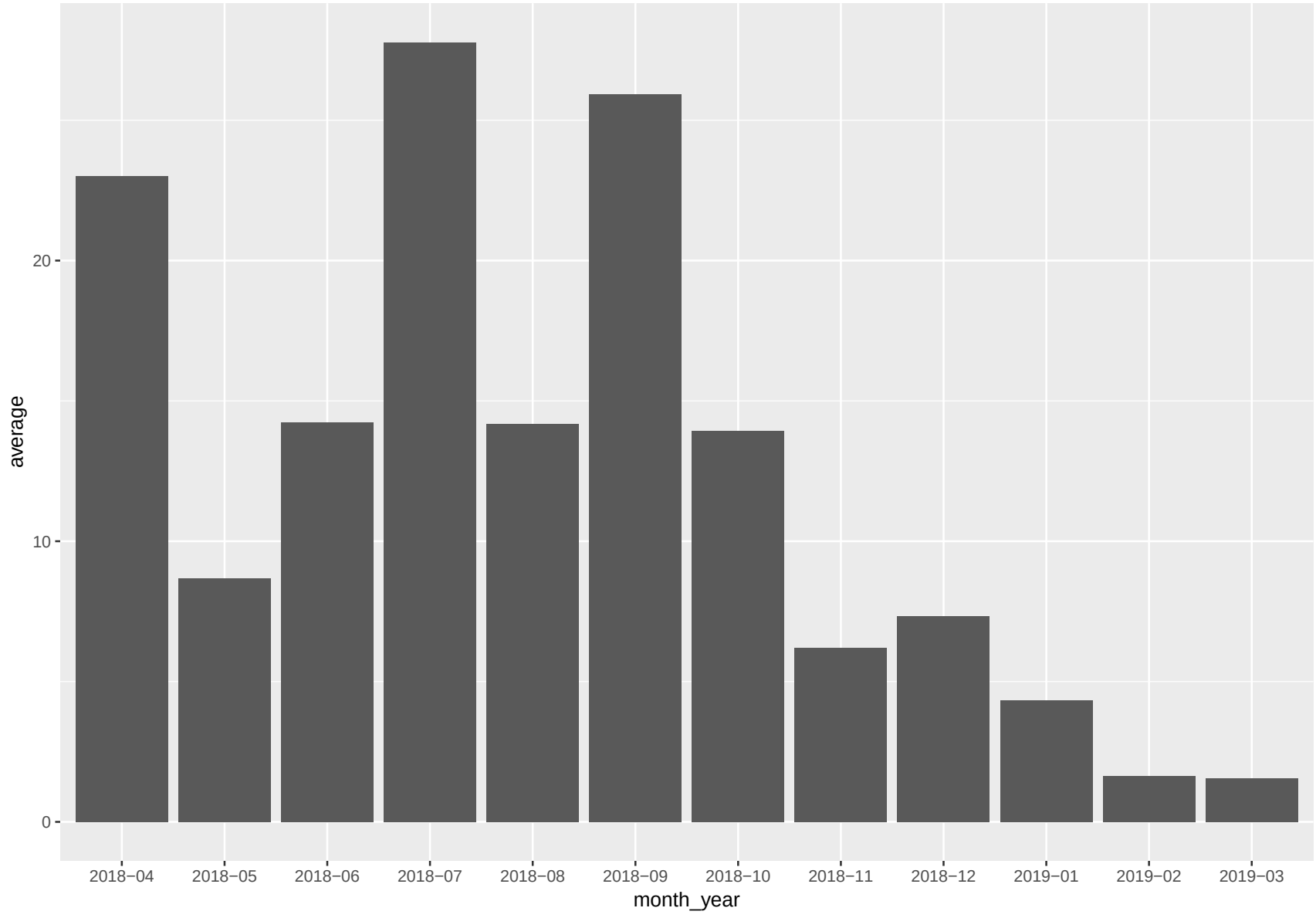
Respone Time

```
## # A tibble: 789 x 19
##       id recipient_id actor_id requestable
##    <dbl>        <dbl>    <dbl> <chr>
##  1     2         1142    18026 project
##  2     3          910    17140 project
##  3     6          910    11947 project
##  4     7         1052    17730 project
##  5     9         1161    18096 project
##  6    13         1052    17750 project
##  7    16         1052    17612 project
##  8    18         1156    15377 project
##  9    21         1142    17798 project
## 10    23         1052    15271 project
## # ... with 779 more rows, and 15 more variables:
## #   day_created <chr>, day_uploaded <chr>,
## #   day_expired <chr>, state <chr>,
## #   day_created2 <date>, month_created <dbl>,
## #   month_year <chr>, month_interval <dbl>,
```

Comparison Mean Response Time Before And After Nov21



Mean Response Time Every 30 Days



Regression Model

Relation between Response time and Acceptance – New Variables

- Acceptance rate = Accepted requests / (Accepted requests + Rejected request) * 100

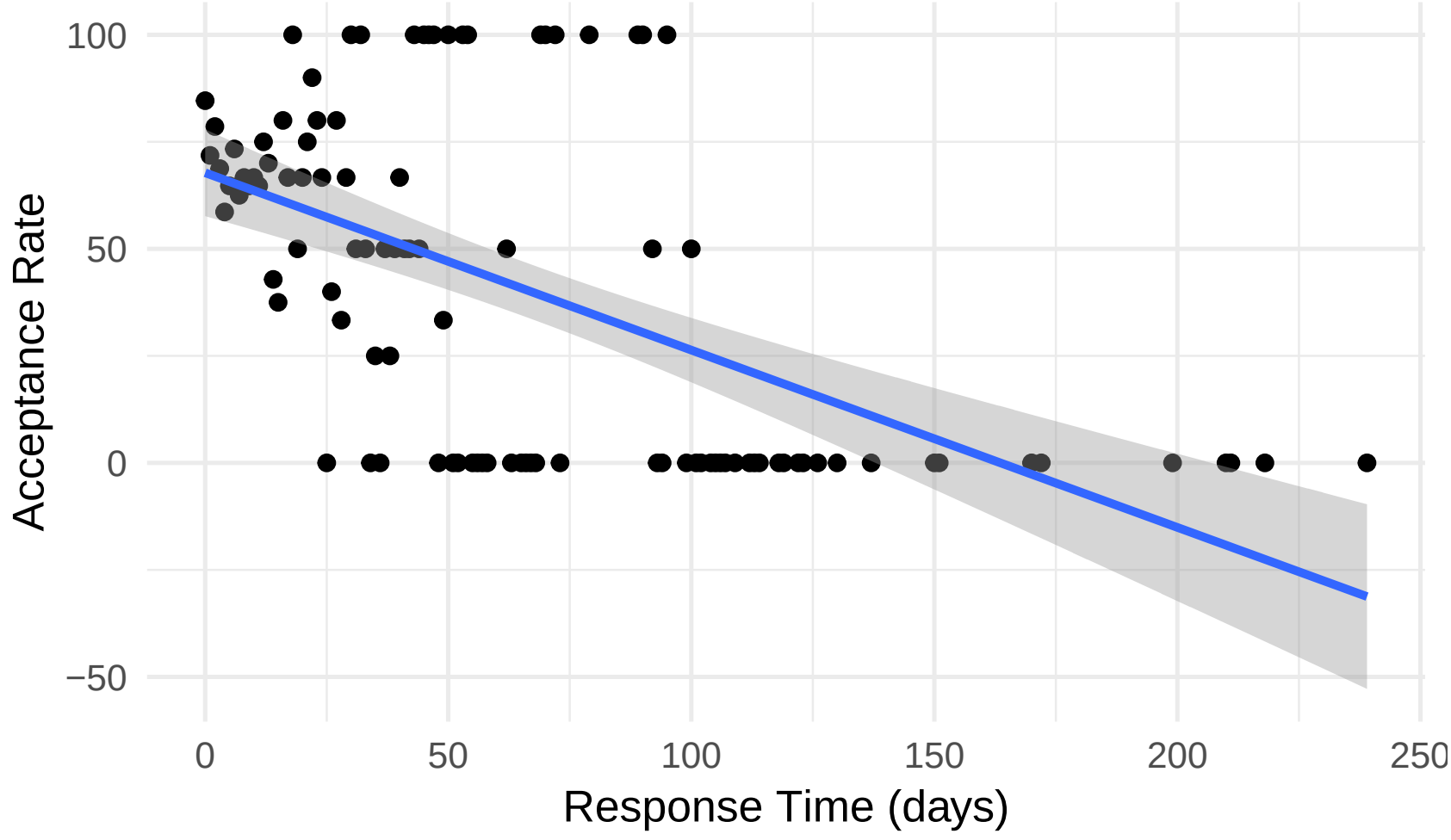
Relation between Response time and Acceptance – New Data

```
## Observations: 104
## Variables: 8
## $ Response_time   <dbl> 0, 1, 2, 3, 4, 5, 6, ...
## $ total           <int> 248, 78, 51, 33, 34, ...
## $ accepted        <dbl> 176, 51, 33, 22, 17, ...
## $ expired         <dbl> 0, 0, 0, 0, 0, 0, 0, ...
## $ rejected        <dbl> 32, 20, 9, 10, 12, 18...
## $ cancelled       <dbl> 16, 5, 9, 1, 4, 8, 0,...
## $ pending         <dbl> 24, 2, 0, 0, 1, 0, 0,...
## $ acceptance_rate <dbl> 84.61538, 71.83099, 7...
```

Relation between Response time and Acceptance – Regression model

$$\hat{y}_i = 67.77 - 0.41 x_i$$

Relation between Response time and Acceptance – Regression Graph



Relation between Response time and Acceptance – Data Summary

- p-value for constant = 6.26^-24
- p-value for response time = 4.74^-10

Relation between Response time and Acceptance – Correlation

- Correlation coefficient = -0.5634846
- Negative relationship
- Moderate relationship

Results

- Volume of request increased over time
- Mean response time decreased
- Regression model: negative relationship between response time and acceptance rate

Conclusion

- Effect? Yes
- Respond quicker
- More requests

Limitations, Challenges and Error

- Data on Aug 30 2018
- Cannot conclude causation
- Challenge: cleaning the data