FinalProject

Chengyu Wu

2022-11-12

Introduction

Steam is one of the biggest online video game seller on personal computers. A big advantage of steam is that it shows reviews from players as well as reactions to these reviews. In this report, we will examine a dataset about reviews of several popular games. If you are also interested in this data set, you can download it at: https://www.kaggle.com/datasets/yukawithdata/steam-review-stats-dataset (https://www.kaggle.com/datasets/yukawithdata/steam-review-stats-dataset)

First, let's load some libraries we need.

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.3.6     ✓ purrr   0.3.4
## ✓ tibble  3.1.8     ✓ dplyr   1.0.9
## ✓ tidyr   1.2.1     ✓ stringr 1.4.1
## ✓ readr   2.1.2     ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(infer)
library(moderndive)
```

Check our data set

Let's first load our data set.

```
steam_reviews <- read.csv(file = 'tableau_steam.csv')
#steam_reviews
```

Let take a look at the columns of the dataset:

date_posted: Time that a review is posted to the steam community.

funny: The number of users that marked this review "Funny".

helpful: The number of users that marked this review as "Helpful".

hour_played: The number of hours that the user spent in the game.

is_early_access_review: If the review is posted at the early access period of the game.

recommendation: If the user recommend this game.

title: Title of the game.

ave_sentiment: sentiment value of the review. (Sentiment value is a indicator of the emotion contained in the words)

word_count: The number of words in the review.

new_review: The newest version of the review.

In this report, we will focus on three columns: “helpful”, “hour_played” and “word_count”. The relationship between if a review is helpful and its word count or the number of hours player spent in the game is complex. Intuitively, a review will receive more helpful if it is long. But many player will ignore long reviews in order to save time. Similarly a player who spent lots of time in a game is more likely to write a good review. However, some users do not want to read reviews from fans of a game. Although we do not know the exact relationship, we can make hypothesis that helpful and hour played are related. Helpful and word count are also related. The null hypothesis we have is that helpful is not related to word count or hour played. For hypothesis test, we will set a significant level alpha of 0.05.

Let’s check some descriptive statistic to get an overview of our data . Firstly, we will check hour played column:

```
mean_hour <- mean(steam_reviews$hour_played)
sd_hour <- sd(steam_reviews$hour_played)
median_hour <- median(steam_reviews$hour_played)
min_hour <- min(steam_reviews$hour_played)
max_hour <- max(steam_reviews$hour_played)
p25_hour <- quantile(steam_reviews$hour_played, 0.25)
p75_hour <- quantile(steam_reviews$hour_played, 0.75)

mean_hour
```

```
## [1] 363.2617
```

```
sd_hour
```

```
## [1] 545.4091
```

```
median_hour
```

```
## [1] 189
```

```
min_hour
```

```
## [1] 0
```

```
max_hour
```

```
## [1] 31962
```

```
p25_hour
```

```
## 25%
##  62
```

```
p75_hour
```

```
## 75%
## 448
```

We can see played hour is very different from reviewers to reviews. We have some reviewer who did not spent any time on the game. We also have game experts spending more than 30,000 hours in one game. These experts lead to a big difference between the mean and the median. We have a mean value that is twice as the median value. But despite of the influence of experts, the median value is large. Normally, players spend about 90 hours to experience all events once in a game.

Then we will check word count column:

```
mean_word <- mean(steam_reviews$word_count)
sd_word <- sd(steam_reviews$word_count)
median_word <- median(steam_reviews$word_count)
min_word <- min(steam_reviews$word_count)
max_word <- max(steam_reviews$word_count)
p25_word <- quantile(steam_reviews$word_count, 0.25)
p75_word <- quantile(steam_reviews$word_count, 0.75)

mean_word
```

```
## [1] 38.5058
```

```
sd_word
```

```
## [1] 83.61023
```

```
median_word
```

```
## [1] 12
```

```
min_word
```

```
## [1] 0
```

```
max_word
```

```
## [1] 2285
```

```
p25_word
```

```
## 25%
##   4
```

```
p75_word
```

```
## 75%
##  37
```

Compared to hour played, word count is not as different from player to player, for the standard deviation is not as high as hour played. A median value of 12 and a mean about 39 means that most players do not write long reviews. Another evidence is the 75% percentile value. A 75% percentile value of 37 means that at least 75% of players are writing reviews under 50 words. However, there are some players writing very long reviews, making a big difference between the mean and the median, according to the maximum word count of 2285. Surprisingly, the minimum word count is 0. A possible explanation is that the reviewer deleted the review but it is mistakenly recorded to this data set.

Finally, let's check helpful column:

```
mean_helpful <- mean(steam_reviews$helpful)
sd_helpful <- sd(steam_reviews$helpful)
median_helpful <- median(steam_reviews$helpful)
min_helpful <- min(steam_reviews$helpful)
max_helpful <- max(steam_reviews$helpful)
p25_helpful <- quantile(steam_reviews$helpful, 0.25)
p75_helpful <- quantile(steam_reviews$helpful, 0.75)

mean_helpful
```

```
## [1] 0.9530674
```

```
sd_helpful
```

```
## [1] 41.86904
```

```
median_helpful
```

```
## [1] 0
```

```
min_helpful
```

```
## [1] 0
```

```
max_helpful
```

```
## [1] 12273
```

```
p25_helpful
```

```
## 25%
##   0
```

```
p75_helpful
```

```
## 75%
##   0
```

We can see that it is very hard to find a helpful mark on steam. 75% of reviews do not receive even one helpful mark. But a maximum value of 12,273 shows that some reviews are quite popular in the player community.

Data Wrangling of our data set

If we look back at our data set, we will find that this data set contains some weird data. For example, there is a review which only contains one word “reeeeeeeee”. It is really hard to understand what it means. Another example is a reivew with word count 0. It is not necessary for us to study a deleted review.

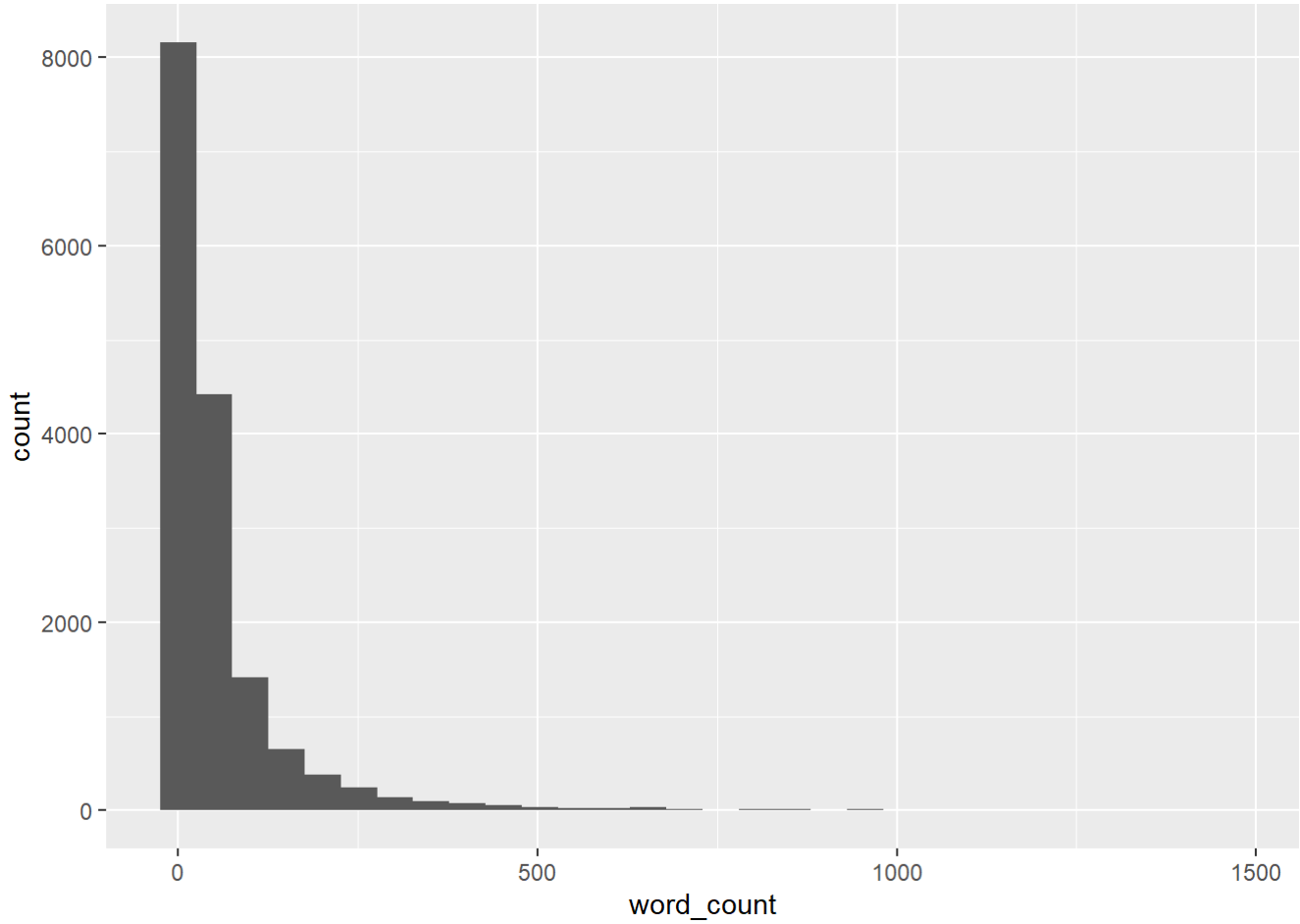
It is necessary for us to filter our data. We will do two things: First, we will discard reviews who did not receive any helpful mark from players. Second, we will discard reviews with word count less than 5.

```
clean_steam_reviews <- steam_reviews %>% filter(helpful > 0) %>% filter(word_count > 5)
#clean_steam_reviews
```

After we filter our data, let’s check the distribution of our three variables. First, lets check word count.

```
ggplot(data = clean_steam_reviews, mapping = aes(x = word_count)) +
  geom_histogram()
```

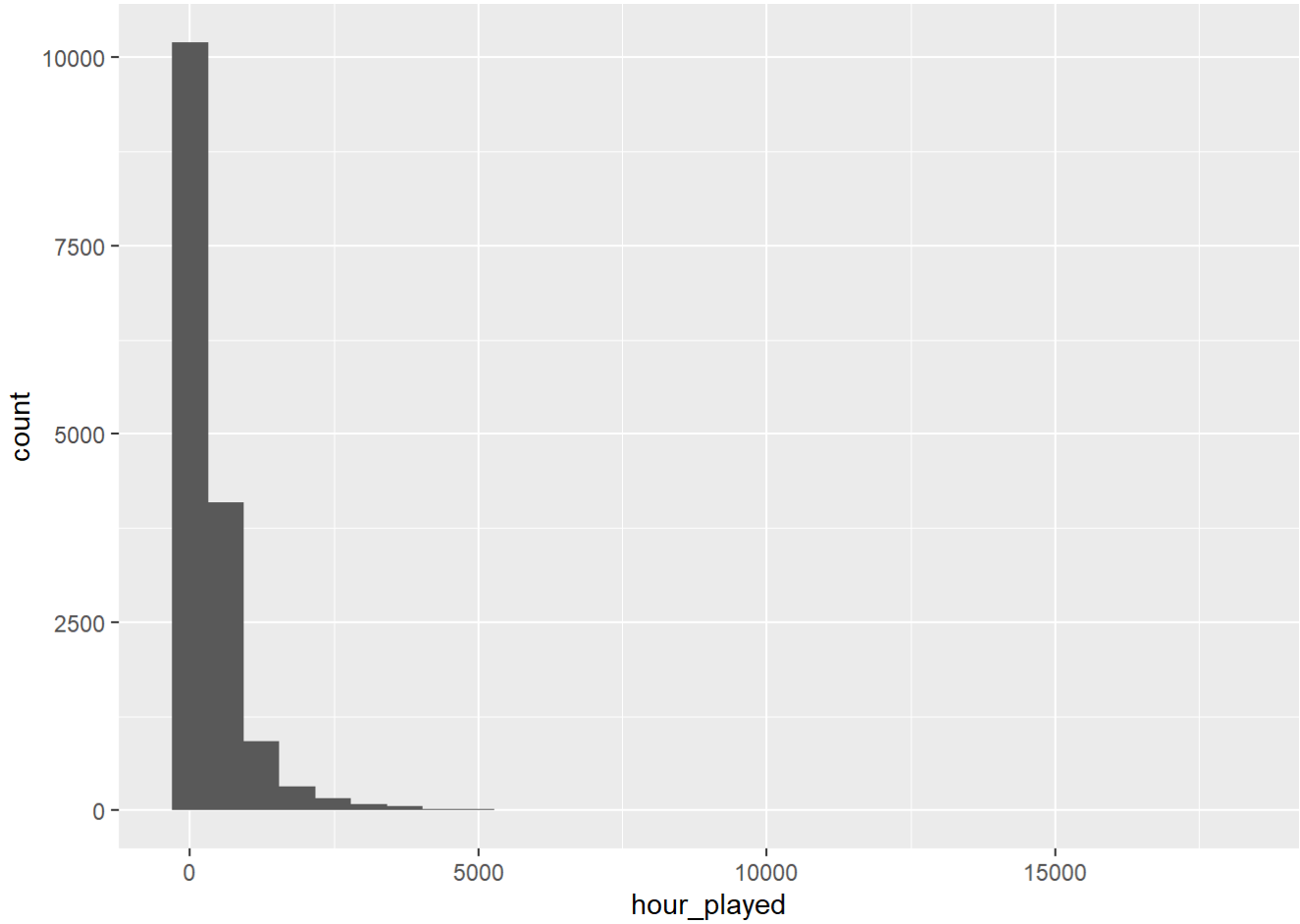
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Then let's check hour played.

```
ggplot(data = clean_steam_reviews, mapping = aes(x = hour_played)) +
  geom_histogram()
```

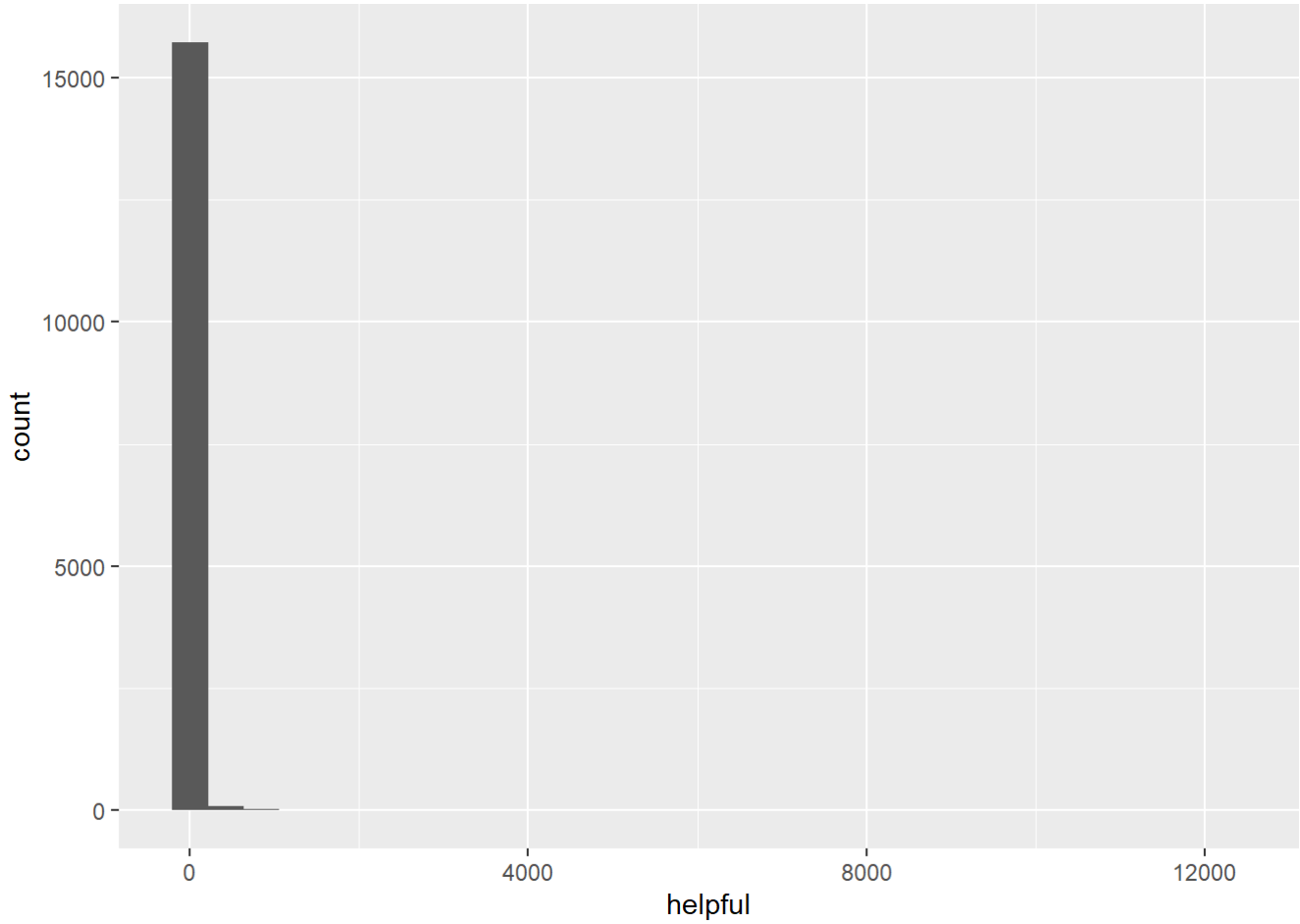
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Finally,let's check helpful.

```
ggplot(data = clean_steam_reviews, mapping = aes(x = helpful)) +
  geom_histogram()
```

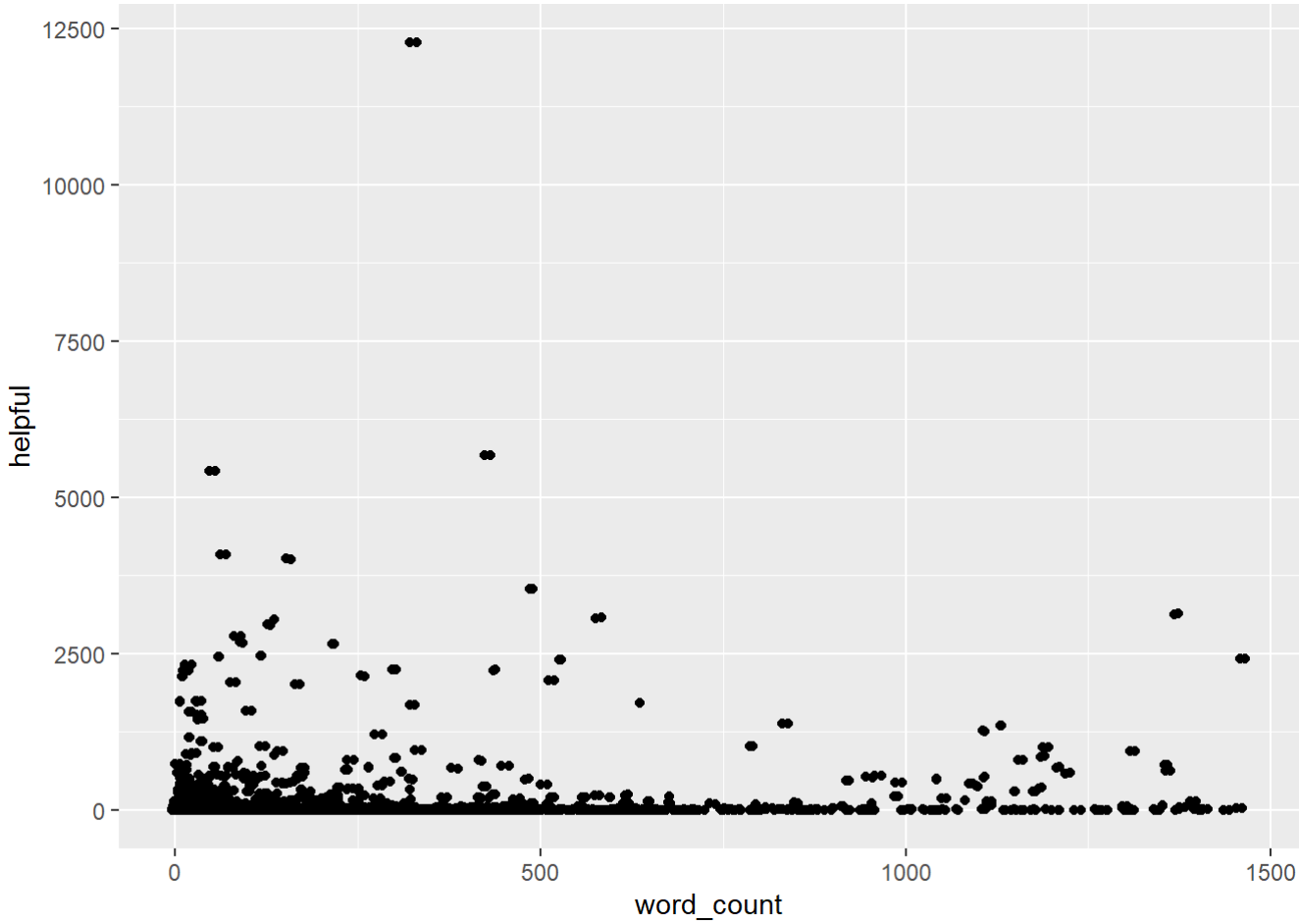
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



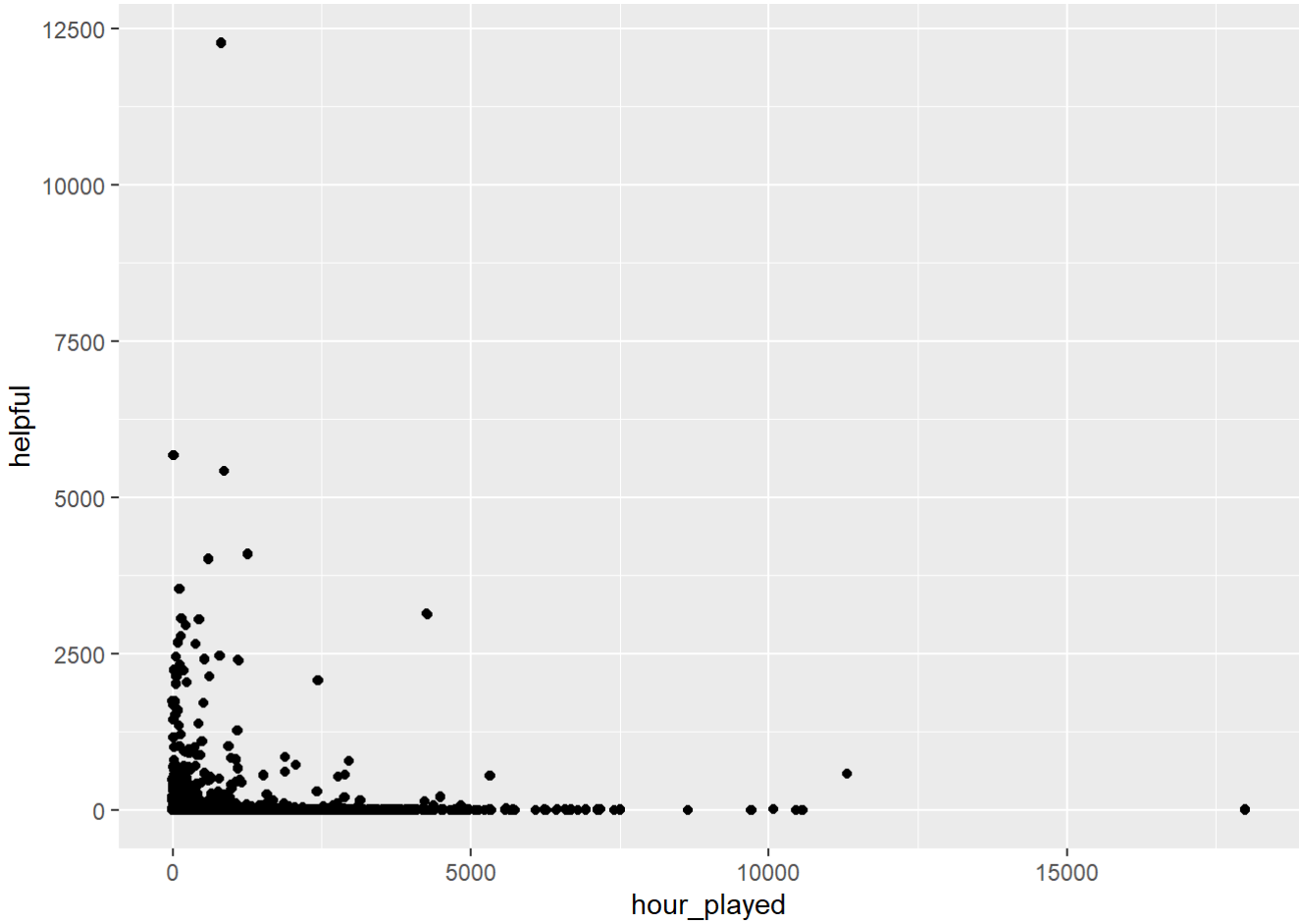
For word count and hour played, we can find that the distributions look like a waterfall. Lots of values are around zero. The larger the value, the lower the count of the value. For most reviews, they only get very few helpful marks.

After we check the distribution of the three variables, it is time for us to plot two variables together using scatter plot. We may find relationship between variables.

```
ggplot(data = clean_steam_reviews, mapping = aes(x = word_count, y = helpful)) +
  geom_point() +
  geom_jitter(width = 10, height = 10)
```




```
ggplot(data = clean_steam_reviews, mapping = aes(x = hour_played, y = helpful)) +
  geom_point() +
  geom_jitter(width = 10, height = 10)
```



Unfortunately, the scatter plots did not show any clear relationship between these data. We can try change the scale of word count and hour played to see if there will be some difference.

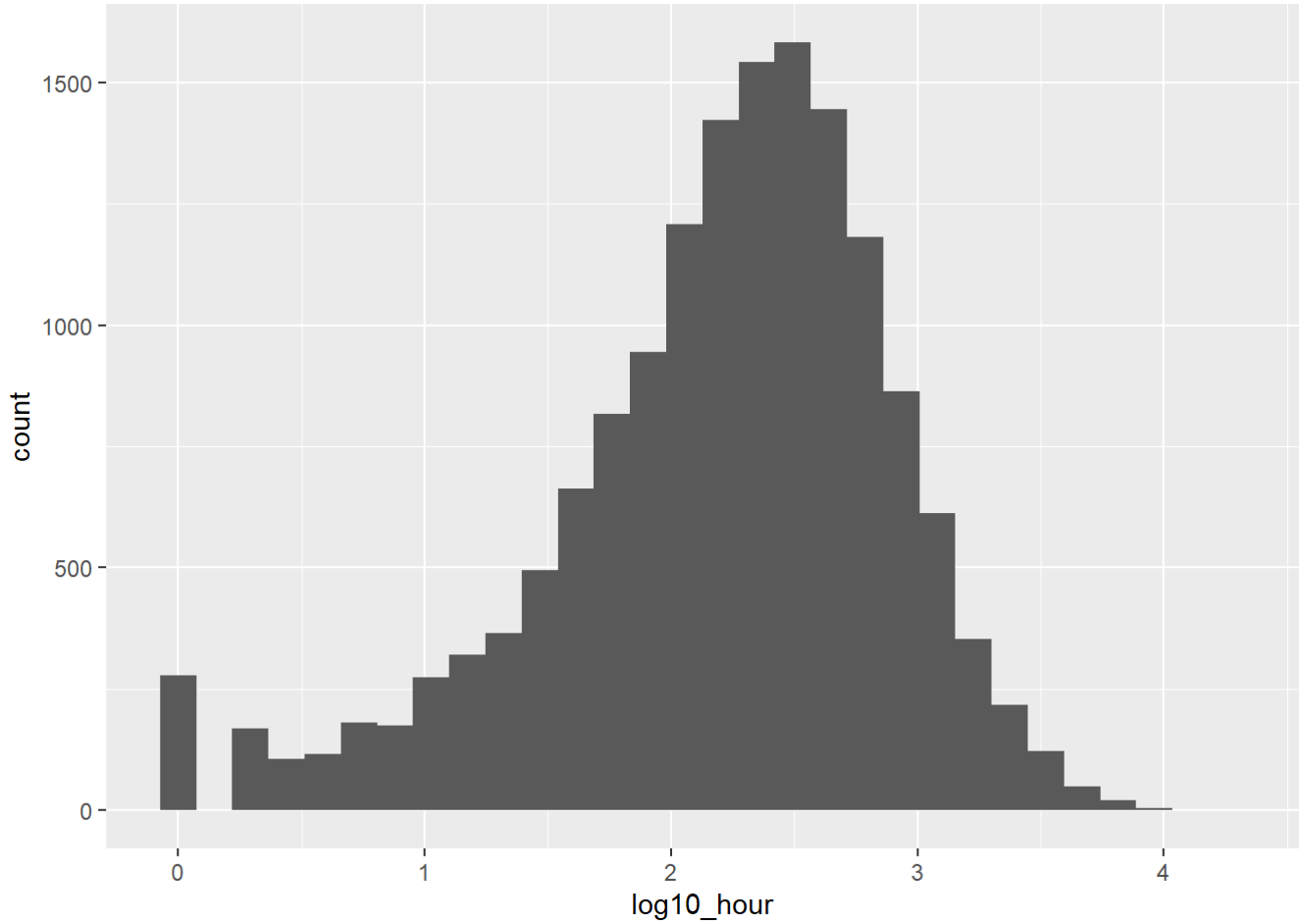
```
clean_steam_reviews <- clean_steam_reviews %>% mutate(log10_hour = log10(hour_played)) %>% muta
te(log10_word = log10(word_count))
#clean_steam_reviews
```

Let's check the histogram of changed data.

```
ggplot(data = clean_steam_reviews, mapping = aes(x = log10_hour)) +
  geom_histogram()
```

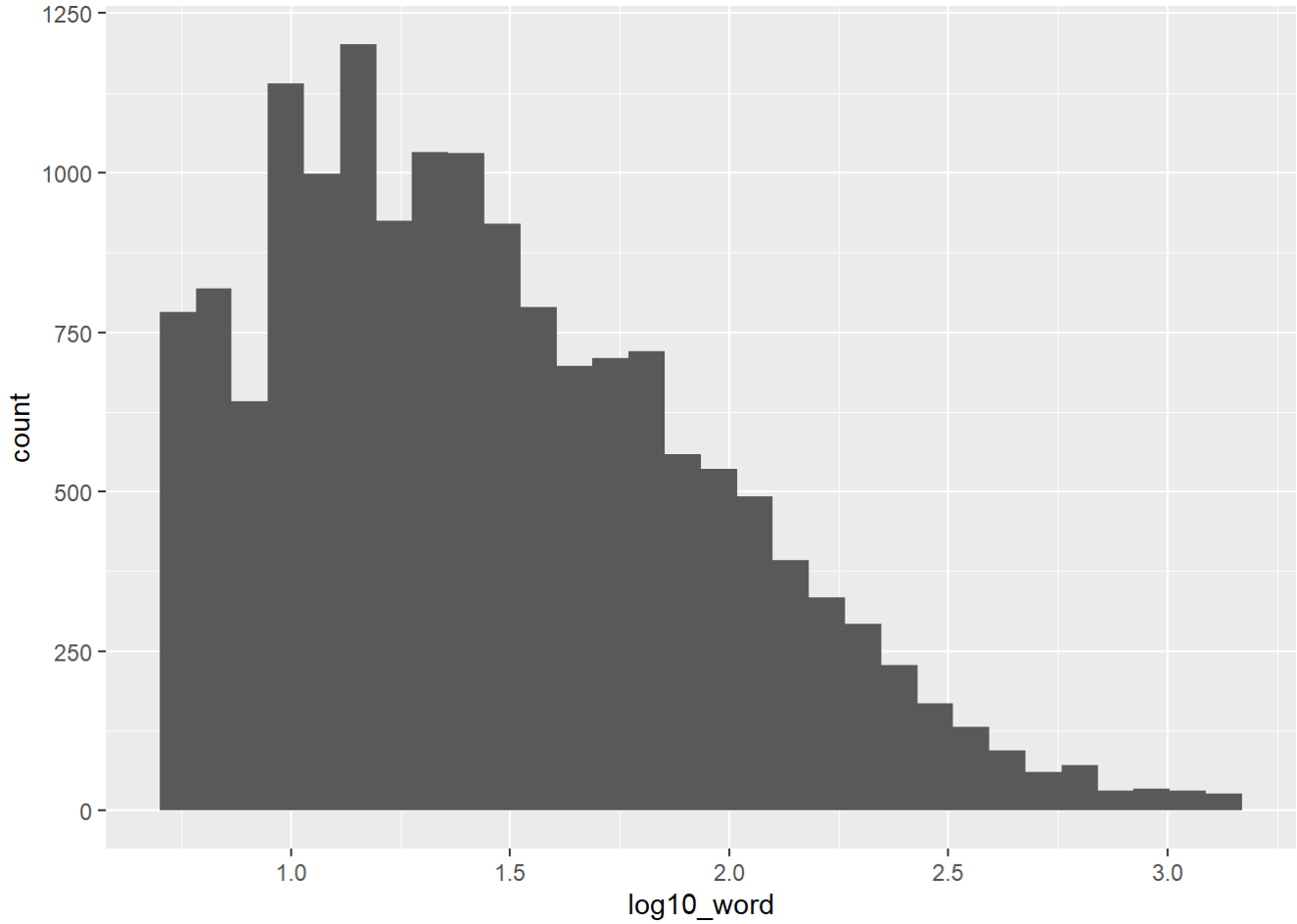
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 353 rows containing non-finite values (stat_bin).
```




```
ggplot(data = clean_steam_reviews, mapping = aes(x = log10_word)) +
  geom_histogram()
```

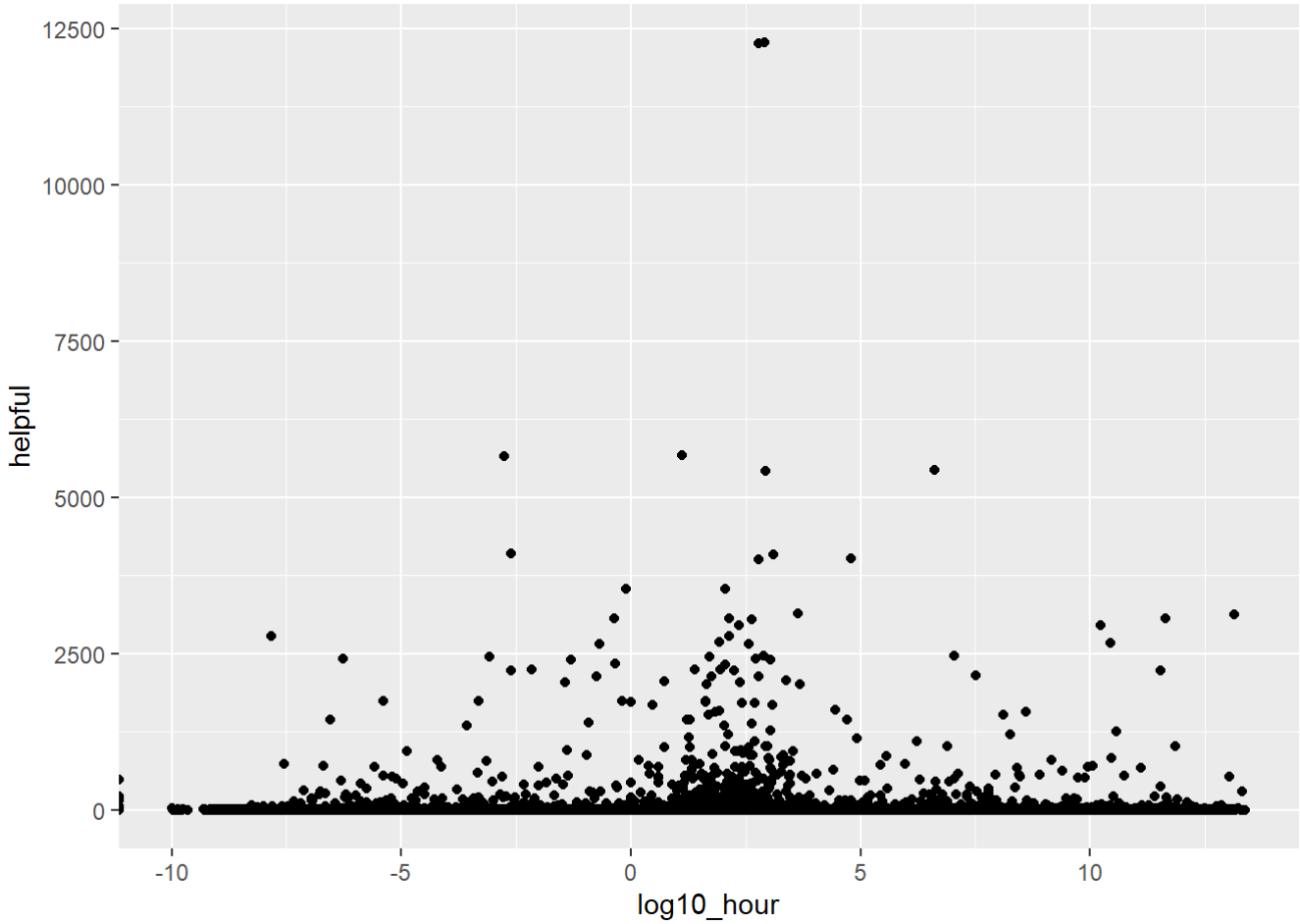
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



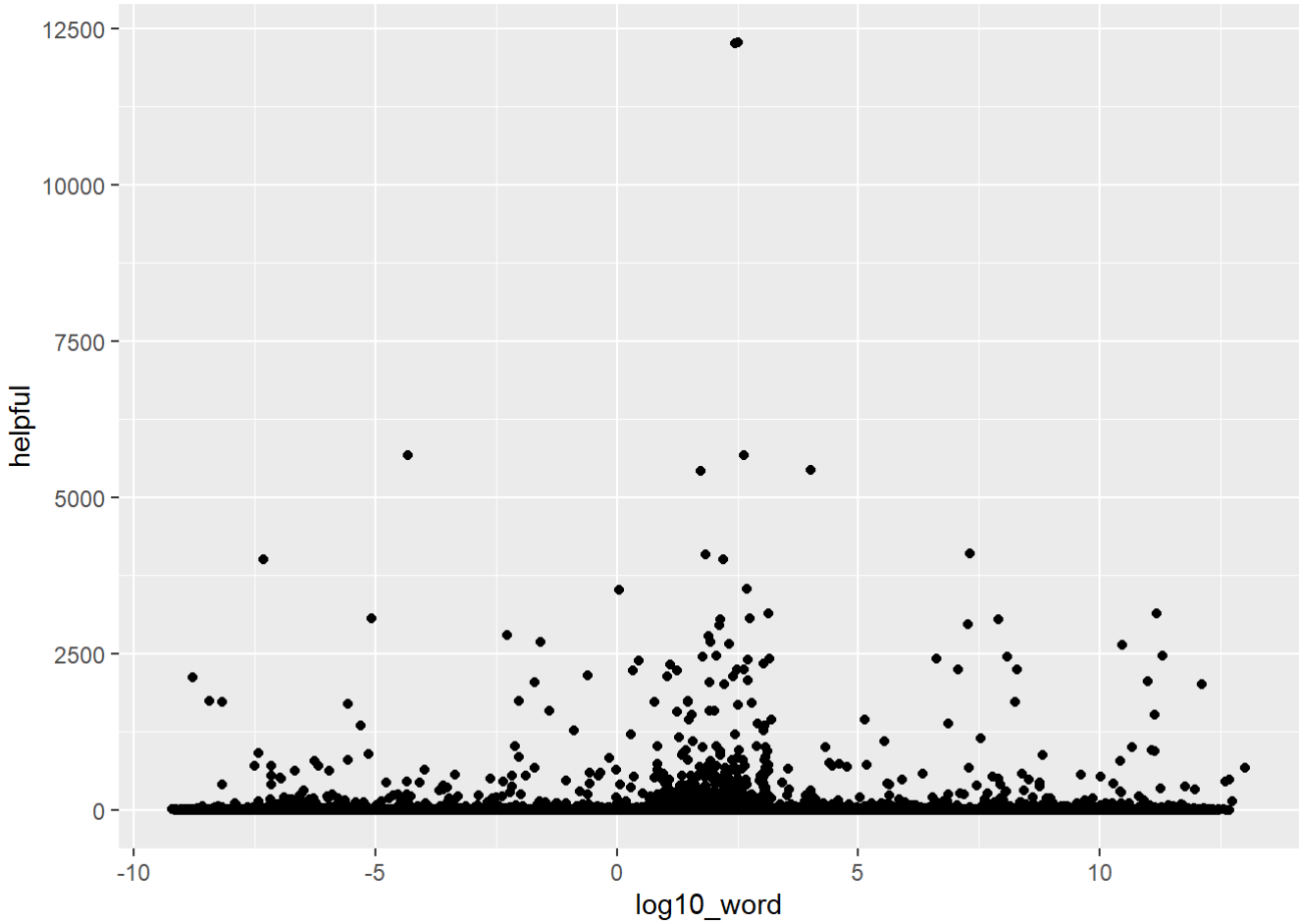
We can see that the log10 hour looks more “normal”. And the log 10 word have a wider distribution. Next, let’s check the scatter plot again.

```
ggplot(data = clean_steam_reviews, mapping = aes(x = log10_hour, y = helpful)) +
  geom_point() +
  geom_jitter(width = 10, height = 10)
```

```
## Warning: Removed 353 rows containing missing values (geom_point).
```




```
ggplot(data = clean_steam_reviews, mapping = aes(x = log10_word, y = helpful)) +
  geom_point() +
  geom_jitter(width = 10, height = 10)
```



Linear regression analysis

Unfortunately, the scatter plots shows that changing the scale of data does not help much. It is clear that the helpful mark does not have any liner relationship with hour played and word count. However, it is possible that helpful mark is decided by the two variables together. Next we will do a linear regression analysis . We want to find a linear model of the three variables.

```
review_model <- lm(helpful ~ word_count + hour_played, data = clean_steam_reviews)
get_regression_table(review_model)
```

```
## # A tibble: 3 × 7
##   term        estimate std_error statistic p_value lower_ci upper_ci
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept      1.05      1.68      0.623   0.533   -2.25     4.35
## 2 word_count     0.197     0.011    18.1     0        0.176    0.219
## 3 hour_played    0.003     0.002     1.29    0.196   -0.001    0.007
```

From this linear model, we can see that the estimate slopes of both word count and hour played are greater than 0. That means the more word count or the more hour the player played the game, the more likely the review from that player receive helpful mark.

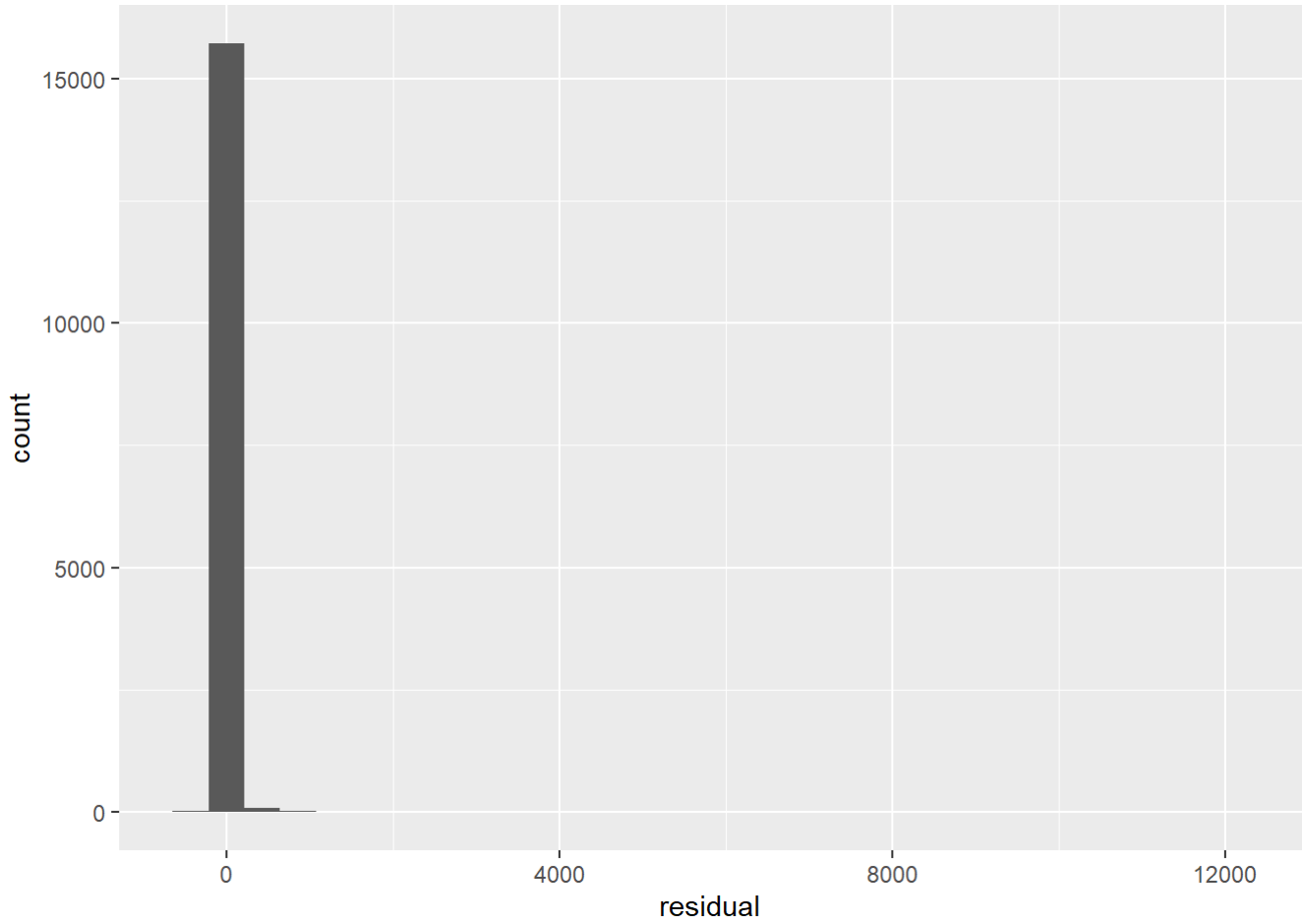
```
points <- get_regression_points(review_model)
points
```

```
## # A tibble: 15,886 × 6
##       ID helpful word_count hour_played helpful_hat residual
##    <int>   <int>      <int>       <int>       <dbl>    <dbl>
##  1     1    2139        260         612        54.0   2085.
##  2     2     219         13          71         3.81   215.
##  3     3     271          9         414         3.93   267.
##  4     4     106        290         900        60.7     45.3
##  5     5     614        310        1878        67.3    547.
##  6     6       1         27          67         6.56    -5.56
##  7     7       1         30         124         7.30    -6.30
##  8     8       1         45         805        12.1     -11.1
##  9     9       1        664         322       133.     -132.
## 10    10       1         24         156         6.20    -5.20
## # … with 15,876 more rows
```

Let's check the distribution of residuals.

```
ggplot(data = points, mapping = aes(x = residual)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The residuals do not follow a normal distribution. As a result, it is not good to make a regression infer. Here I'll still use the P value to analyse this linear model but remember that we are not confident on this result.

We can see a large difference between the P value of two variables in the regression table. Word count has a very small P value. We are very confident to reject the null hypothesis, meaning that there is a relationship between word count and helpful marks. P value of hour played is very big. It is much bigger than the significant

level we set(0.05). As a result, we cannot reject the null hypothesis. We cannot confidently say that there is a relationship between hour played and helpful.

Conclusion

From this data set, we find that how many helpful mark a review get on steam does not have a clear linear relationship with the time reviewers spent in game or the word count of the review. However, we can find that word count has a greater influence than hour players spent in game.