THE POPULARITY OF BOOKS ON GOODREADS

DATA 606 Final Project

Stephanie Chiang

Fall 2024

Abstract

This is a statistical analysis of some of the factors that contribute to the popularity of books on Goodreads. The observational dataset posted on Kaggle was originally sourced from Goodreads' Top 100 lists of the most popular books for each year from 1980 to 2023. In particular, the focus will be to answer the question:

Are serials, or books that are part of multi-volume series, associated with higher ratings and a larger readership than standalone books? In other words, is there a relationship between the independent variables (standalone books vs serials, and first in series vs sequel) and the dependent variables (mean user ratings, number of current readers and potential readership counts)?

Using summary statistics, visualizations and regression modeling, the results indicate that serials do not have significantly different mean user ratings; and sequels also do not show notable difference in average ratings than a first published installment. However, there are more current and potential Goodreads users for standalone book titles than books in a series. This analysis could provide potentially valuable insights for publishers, authors, and marketers for data-driven decision-making.

Data Preparation

The explanatory variables are created from text fields in the raw data. Each of the 4399 rows is given a value in a new categorical column with two levels (TRUE or FALSE) called `serial` based on whether there are non-empty strings under `series_title` and `series_release_number`.

Then, in a second data frame for only those books with `serial` set to TRUE, each observation is marked in a new `first_book` column with a TRUE if it is the first book published in its series or FALSE for sequels and prequels. The determination here is that since prequels are published *after* the initial volume, they should not be considered the first in a series.

The response variables are numerical: `rating_score`, `num_ratings`, `current_readers`, `want_to_read`.

```
library(tidyverse)

raw_books <- read.csv(file = "goodreads_top100.csv")

# select relevant columns
books <- raw_books |>
  select("isbn",
         "title",
         "series_title",
         "series_release_number",
         "rating_score",
```

```
          "num_ratings",
          "current_readers",
          "want_to_read")
```

```
# convert blank strings to NAs in text columns
books <- books |>
  mutate(isbn = na_if(isbn, "")) |>
  mutate(series_title = na_if(series_title, "")) |>
  mutate(series_release_number = na_if(series_release_number, ""))

# remove duplicate ISBN numbers / repeated books
books <- books |>
  distinct(isbn, .keep_all = TRUE)

# add column to indicate if the book in series
books <- books |>
  mutate(serial = !is.na(books$series_title) & !is.na(books$series_release_number))

knitr::kable(head(books[, 2:5]), "pipe")
```

title	series_title	series_release_number	rating_score
Summer Story	Brambly Hedge	2	4.45
The Lake of Darkness	NA	NA	3.76
Beyond the Blue Event Horizon	Heechee Saga	2	3.95
St. Peter's Fair	Chronicles of Brother Cadfael	4	4.12
Twice Shy	NA	NA	3.92
The Door in the Hedge	NA	NA	3.70

```
# create a 2nd table for series-only analysis
series <- filter(books, serial == TRUE)

# add column for if it is the first release of its series
series <- series |>
  mutate(first_book = ifelse(grepl("^1(?!\\d)", series$series_release_number, perl = TRUE),
                             TRUE,
                             FALSE)) |>
  subset(select = -c(serial))

knitr::kable(head(series[, 2:5]), "pipe")
```

title	series_title	series_release_number	rating_score
Summer Story	Brambly Hedge	2	4.45
Beyond the Blue Event Horizon	Heechee Saga	2	3.95
St. Peter's Fair	Chronicles of Brother Cadfael	4	4.12
Pawn of Prophecy	The Belgariad	1	4.16
Pacific Vortex!	Dirk Pitt	1	3.80
Dragons of Autumn Twilight	Dragonlance: Chronicles	1	4.01

Summary Statistics & Data Visualizations

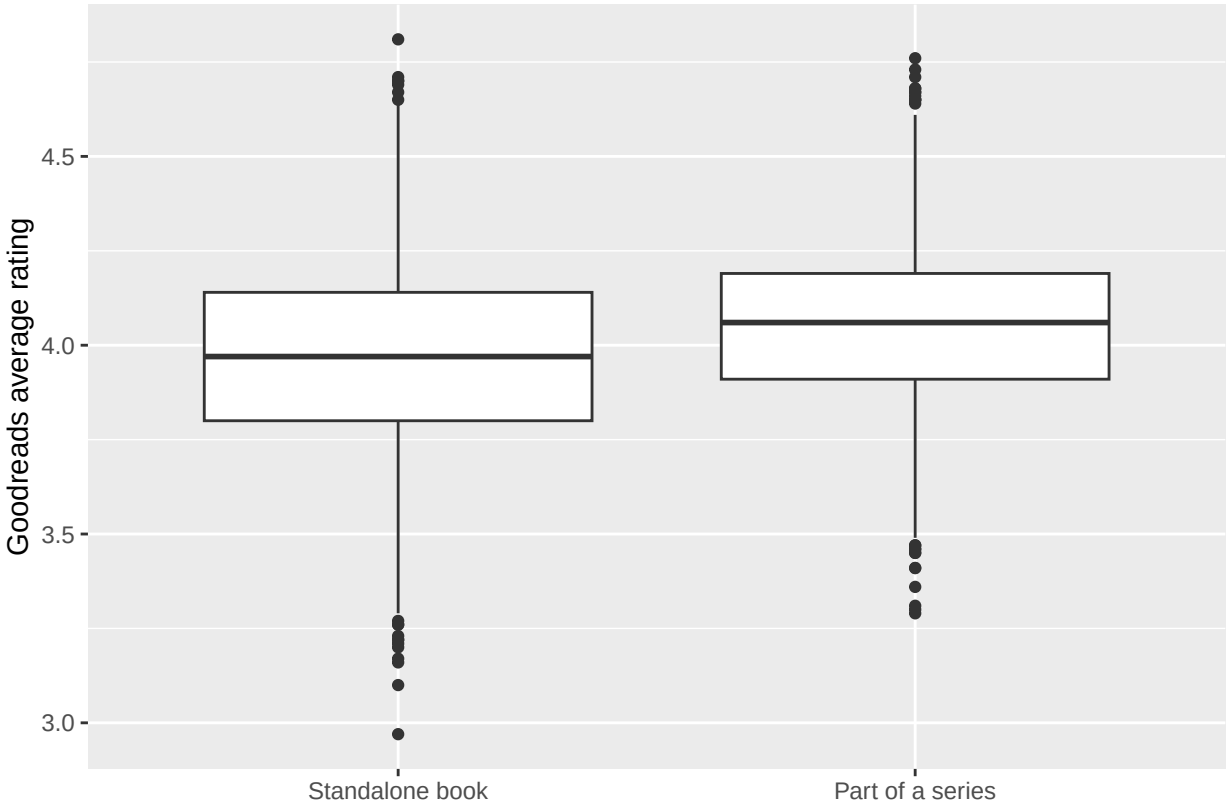
Mean Ratings: Series vs Standalone Books

Comparing the average ratings shows a slight a preference by Goodreads users for serials over standalone books.

```
rating_summary <- books |>
  group_by(serial) |>
  reframe(
    count = n(),
    mean = mean(rating_score),
    sd = sd(rating_score),
    median = median(rating_score),
    min = min(rating_score),
    max = max(rating_score),
  )

rating_summary
```

```
## # A tibble: 2 x 7
##   serial count  mean    sd median   min   max
##   <lgl>  <int> <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1 FALSE   1813  3.96 0.256   3.97  2.97  4.81
## 2 TRUE    1805  4.06 0.216   4.06  3.29  4.76
```

```
ggplot(books, aes(x = serial, y = rating_score)) +
  geom_boxplot() +
  scale_x_discrete(labels = c("FALSE" = "Standalone book", "TRUE" = "Part of a series")) +
  labs(y = "Goodreads average rating", x= "")
```



Since each book is an independent observation and the sample sizes for each group are comfortably large, the conditions for inference are satisfied; a hypothesis test for the difference of the two means can determine any association.

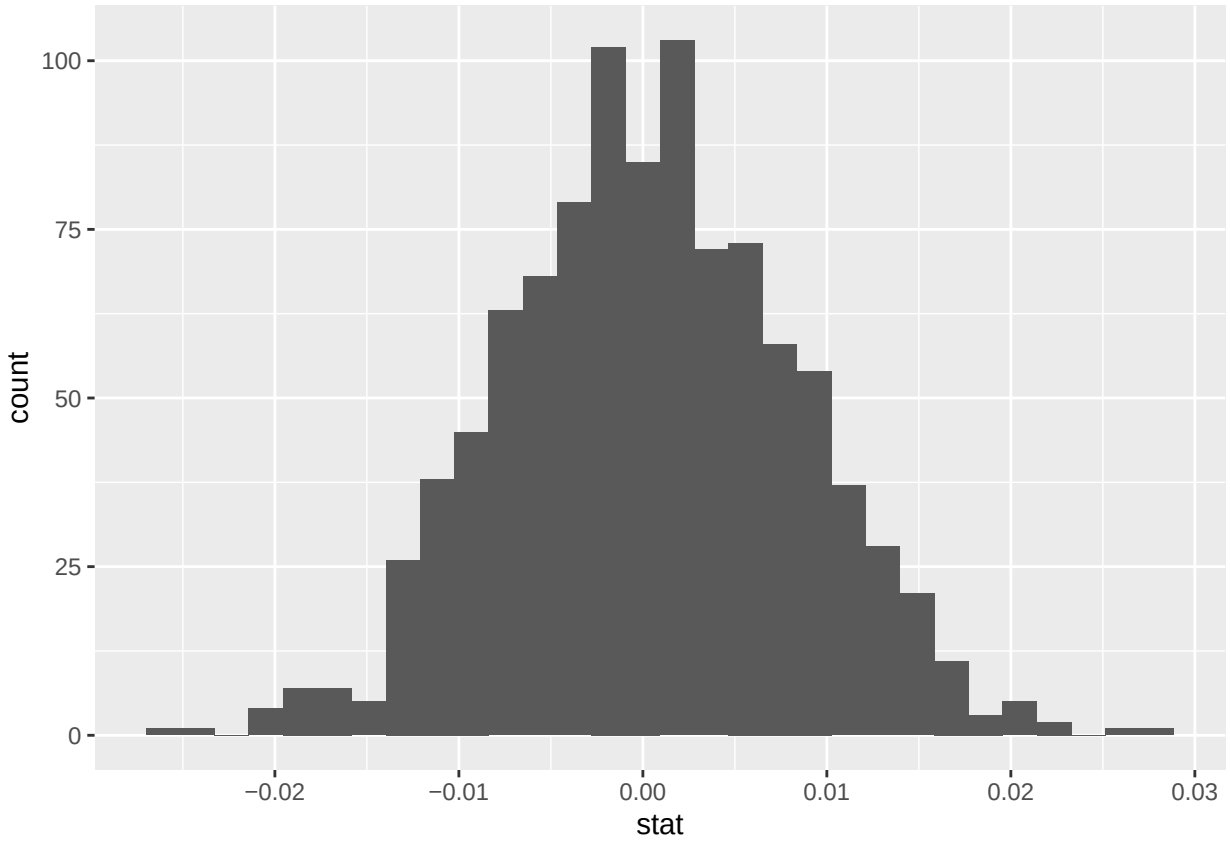
- The null hypothesis H0: There is no relationship between being part of a series and average rating.
- The alternative hypothesis H1: The average ratings are significantly different for serials.

Below, the difference in means is calculated in the order of TRUE - FALSE != 0. The test is then simulated on the null distribution and plotted.

```
library(infer)
set.seed(99)

series_obs_diff <- books |>
  specify(rating_score ~ serial) |>
  calculate(stat = "diff in means", order = c(TRUE, FALSE))

series_obs_diff
```

```
## Response: rating_score (numeric)
## Explanatory: serial (factor)
## # A tibble: 1 x 1
##     stat
##    <dbl>
## 1 0.0932
```

```
series_null_dist <- books |>
  specify(rating_score ~ serial) |>
  hypothesize(null = "independence") |>
  generate(reps = 1000, type = "permute") |>
  calculate(stat = "diff in means", order = c(TRUE, FALSE))

ggplot(data = series_null_dist, aes(x = stat)) +
  geom_histogram()
```




```
series_null_dist |>
  get_p_value(obs_stat = series_obs_diff, direction = "two_sided")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1       0
```

```
series_diff_ci <- series_null_dist |> get_ci(level = 0.95)

series_diff_ci
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1  -0.0139   0.0154
```

At a confidence level of 95%, the difference in mean ratings between series and standalone books should fall between -0.014 to 0.015. Since this contains 0, we can fail to reject the null hypothesis. There is no significant difference in average rating for serials.

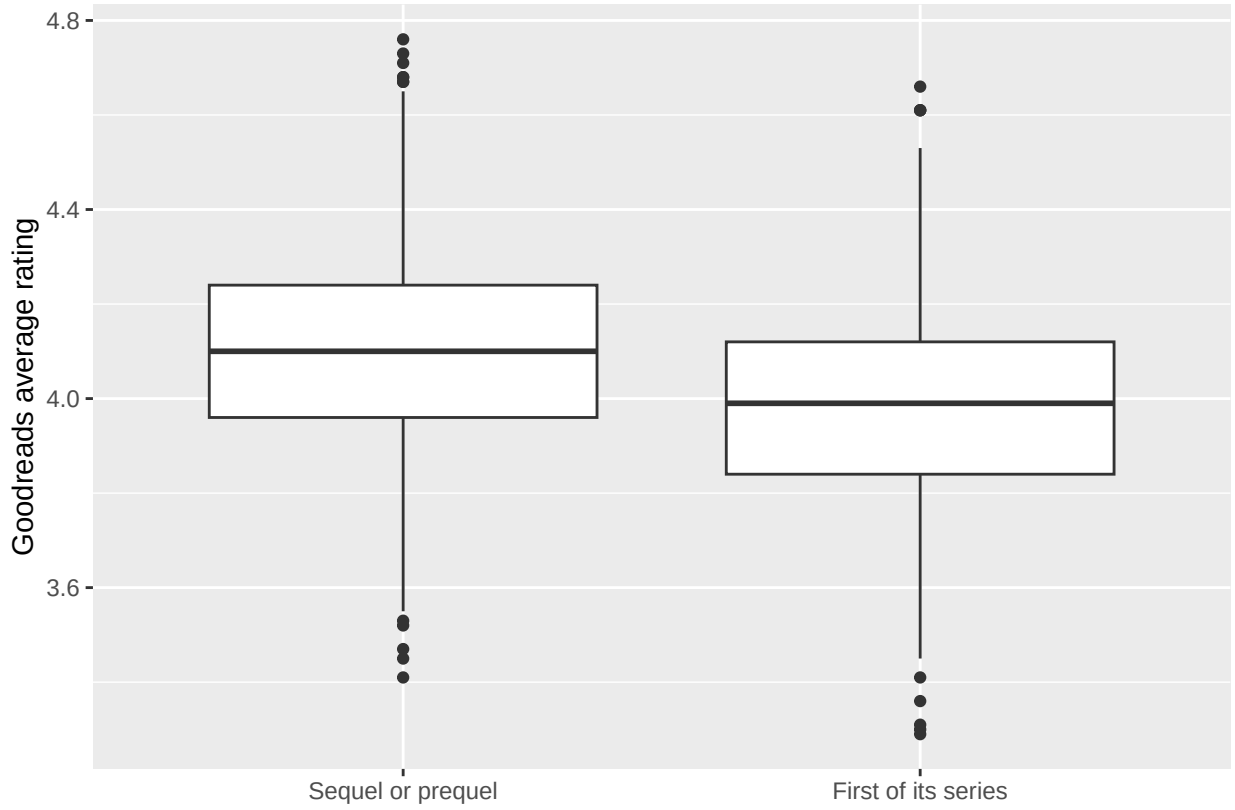
Average Ratings: Firsts vs Sequels

Within series, there is a somewhat more noticeable bump in ratings for sequels over the first book.

```
rating_summary_sequels <- series |>
  group_by(first_book) |>
  reframe(
    count = n(),
    mean = mean(rating_score),
    sd = sd(rating_score),
    median = median(rating_score),
    min = min(rating_score),
    max = max(rating_score),
  )

rating_summary_sequels
```

```
## # A tibble: 2 x 7
##   first_book count  mean    sd median   min   max
##   <lgl>      <int> <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1 FALSE       1141  4.10 0.206   4.1   3.41  4.76
## 2 TRUE         664  3.98 0.209   3.99  3.29  4.66
```

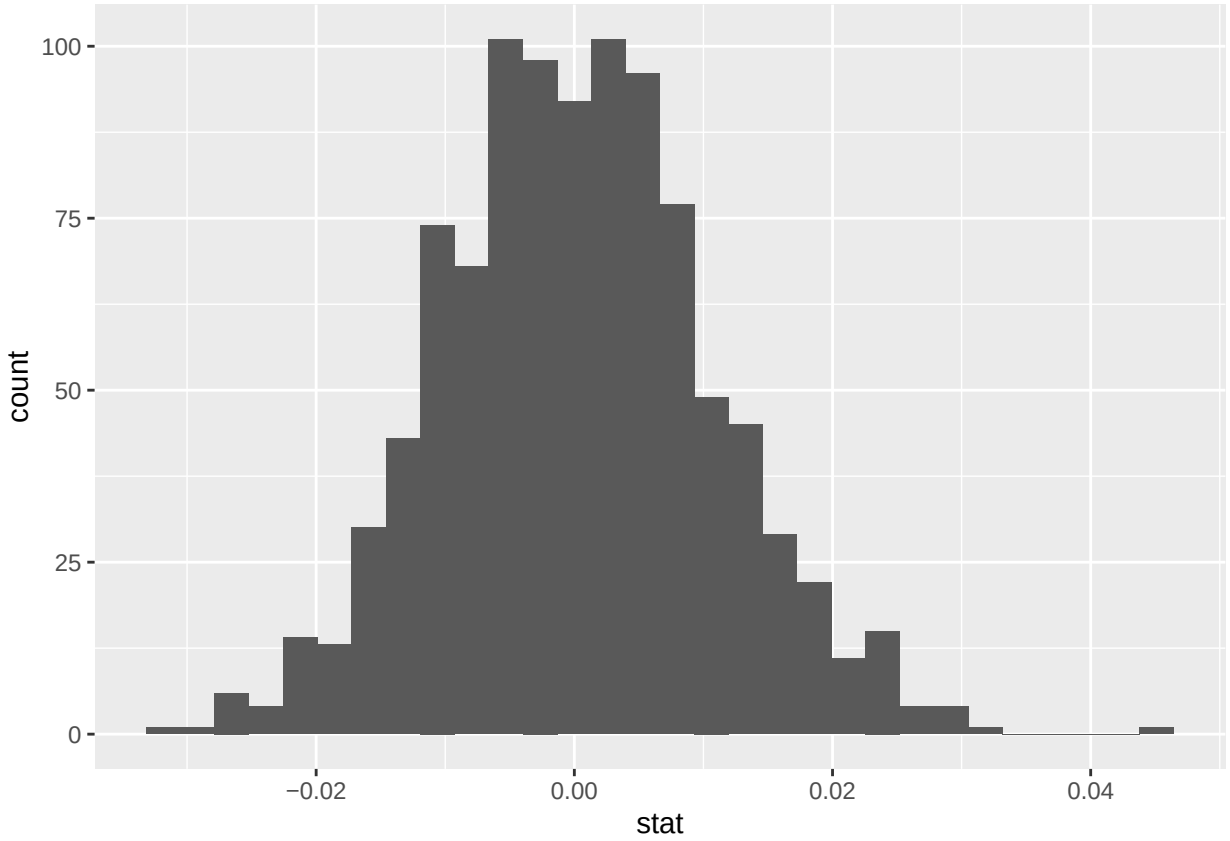
```
ggplot(series, aes(x = first_book, y = rating_score)) +
  geom_boxplot() +
  scale_x_discrete(labels = c("FALSE" = "Sequel or prequel",
                              "TRUE" = "First of its series")) +
  labs(y = "Goodreads average rating", x = "")
```



The hypothesis test for serials alone is as follows:

The null hypothesis H0: There is no relationship between being a sequel and average rating. The alternative hypothesis H1: The average ratings are significantly different for sequels than first books.

```
set.seed(99)

first_obs_diff <- series |>
  specify(rating_score ~ first_book) |>
  calculate(stat = "diff in means", order = c(TRUE, FALSE))

first_null_dist <- series |>
  specify(rating_score ~ first_book) |>
  hypothesize(null = "independence") |>
  generate(reps = 1000, type = "permute") |>
  calculate(stat = "diff in means", order = c(TRUE, FALSE))

ggplot(data = first_null_dist, aes(x = stat)) +
  geom_histogram()
```




```
first_null_dist |>
  get_p_value(obs_stat = first_obs_diff, direction = "two_sided")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1       0
```

```
first_diff_ci <- first_null_dist |> get_ci(level = 0.95)

first_diff_ci
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1  -0.0200   0.0220
```

With the confidence level set to 95%, the difference in mean ratings between first books and sequels/prequels should fall between -0.02 to 0.02. Since this contains 0, we can fail to reject the null hypothesis. There is no significant difference in average rating for sequels.

Readership

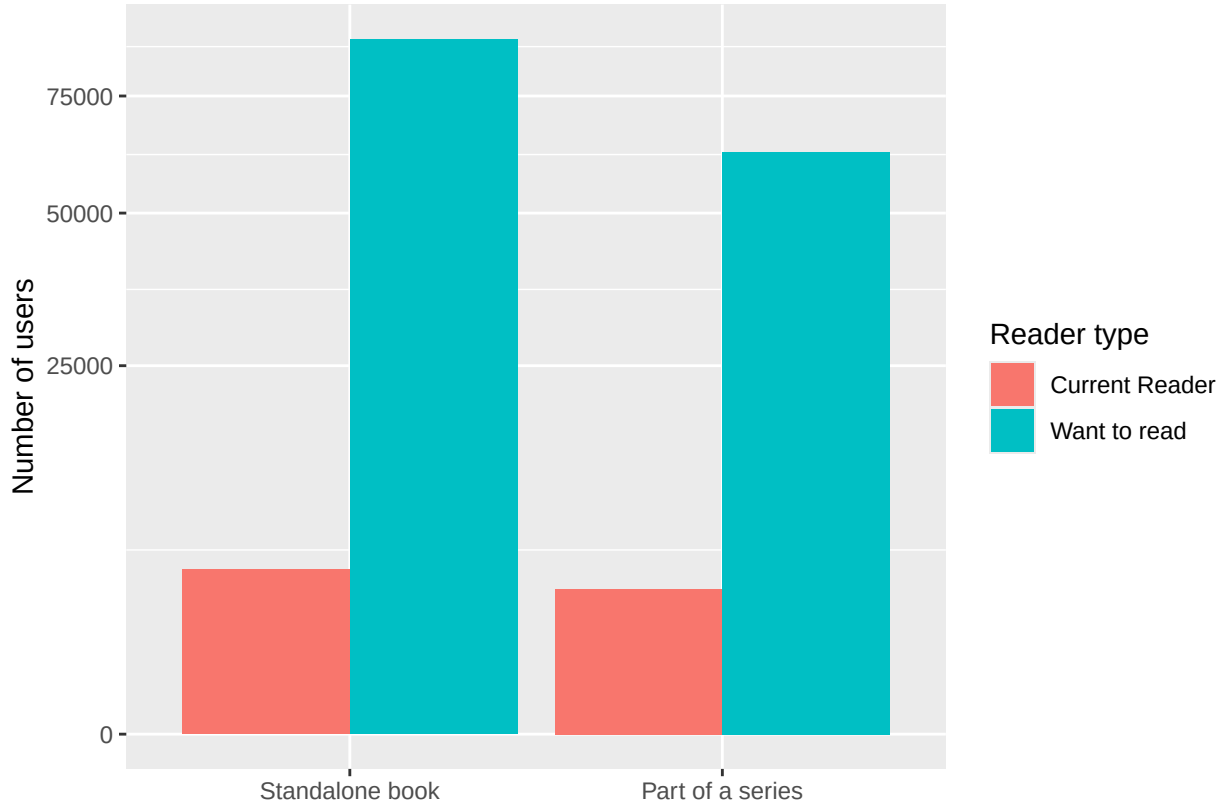
A different angle of examination than rating is readership: users who marked themselves as current readers of a title or interested/potential readers. The average number of users who are either currently reading or

`want_to_read` a standalone book is much higher than for series.

```
books_readership <- books |>
  pivot_longer(cols = c("current_readers", "want_to_read"),
               names_to = "reader_type",
               values_to = "readership")

readership_summary <- books_readership |>
  group_by(serial, reader_type) |>
  summarize(mean_readership = mean(readership, na.rm = TRUE))

readership_summary
```

```
## # A tibble: 4 x 3
## # Groups:   serial [2]
##   serial reader_type     mean_readership
##   <lgl>  <chr>                     <dbl>
## 1 FALSE  current_readers           5013.
## 2 FALSE  want_to_read             88904.
## 3 TRUE   current_readers           3874.
## 4 TRUE   want_to_read             62413.
```

```
ggplot(readership_summary, aes(fill = reader_type, x = serial, y = mean_readership)) +
  geom_bar(position = "dodge", stat = "identity") +
  scale_y_sqrt() +
  labs(y = "Number of users", x= "") +
  scale_x_discrete(labels = c("FALSE" = "Standalone book", "TRUE" = "Part of a series")) +
  scale_fill_discrete(name = "Reader type", labels = c("Current Reader", "Want to read"))
```



To isolate if the series length is a factor for users (is starting a 15-book series daunting to most users?), the data can be transformed to display the readership numbers by length of series.

```
# group by unique series titles, calculate the totals for series length and readership
series_length <- series |>
  group_by(series_title) |>
  summarize(
    series_len = n(),
    total_current = sum(current_readers),
    total_want_read = sum(want_to_read)) |>
  replace_na(list(total_current = 0, total_potential = 0)) |>
  select(-series_title)

series_length
```

```
## # A tibble: 937 x 3
##    series_len total_current total_want_read
##         <int>         <dbl>           <dbl>
##  1          1           121           25500
##  2          1          4711           84900
##  3          2           303             729
##  4          1            65            5248
##  5          5        437100         2007500
##  6          1          7077           81300
##  7          7          2723           23088
##  8          1           242            9212
```

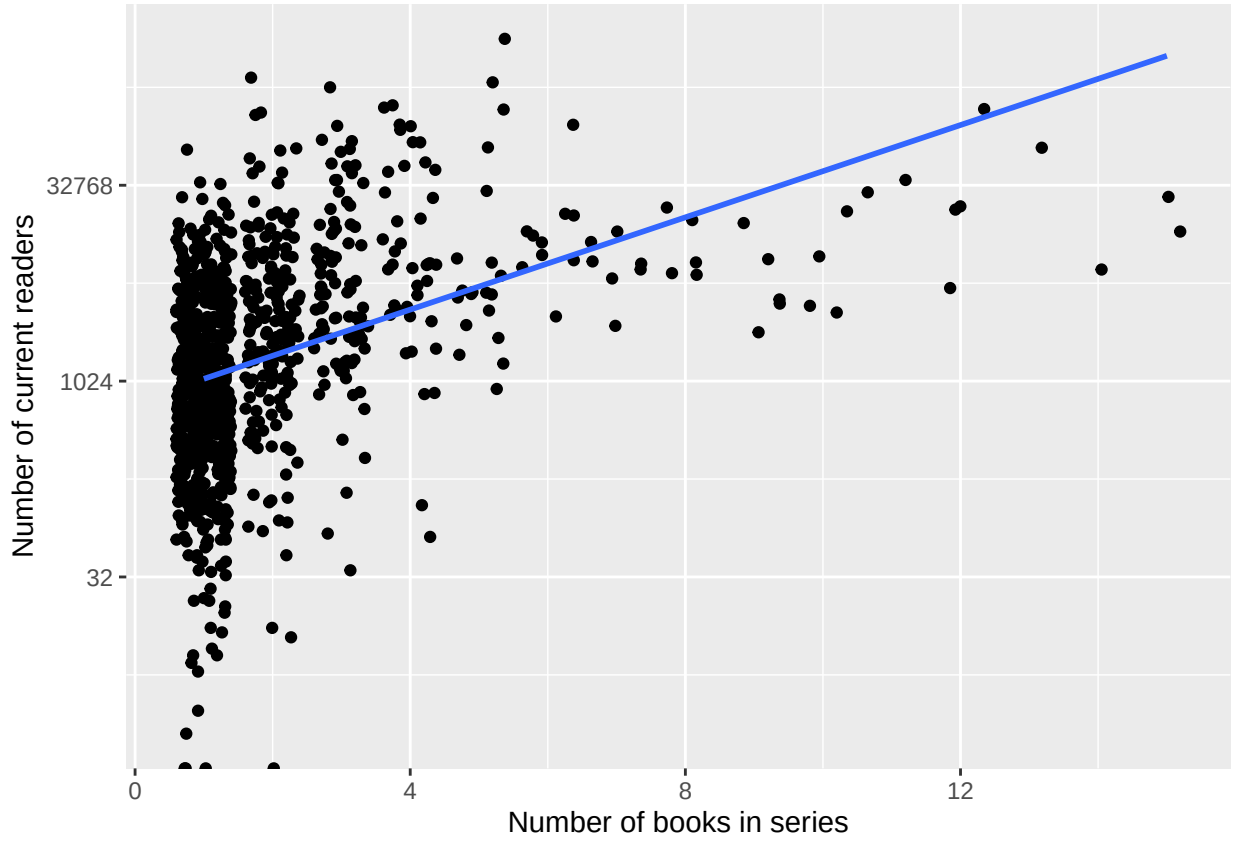
```
##  9          1           543           12400
## 10         12         21291          541300
## # i 927 more rows
```

```
ggplot(series_length, aes(x = series_len, y = total_current)) +
  geom_jitter() +
  scale_y_continuous(transform = "log2") +
  labs(y = "Number of current readers", x = "Number of books in series") +
  stat_smooth(method = "lm", se = FALSE)
```

```
## Warning in scale_y_continuous(transform = "log2"): log-2 transformation introduced infinite values.
## log-2 transformation introduced infinite values.
```

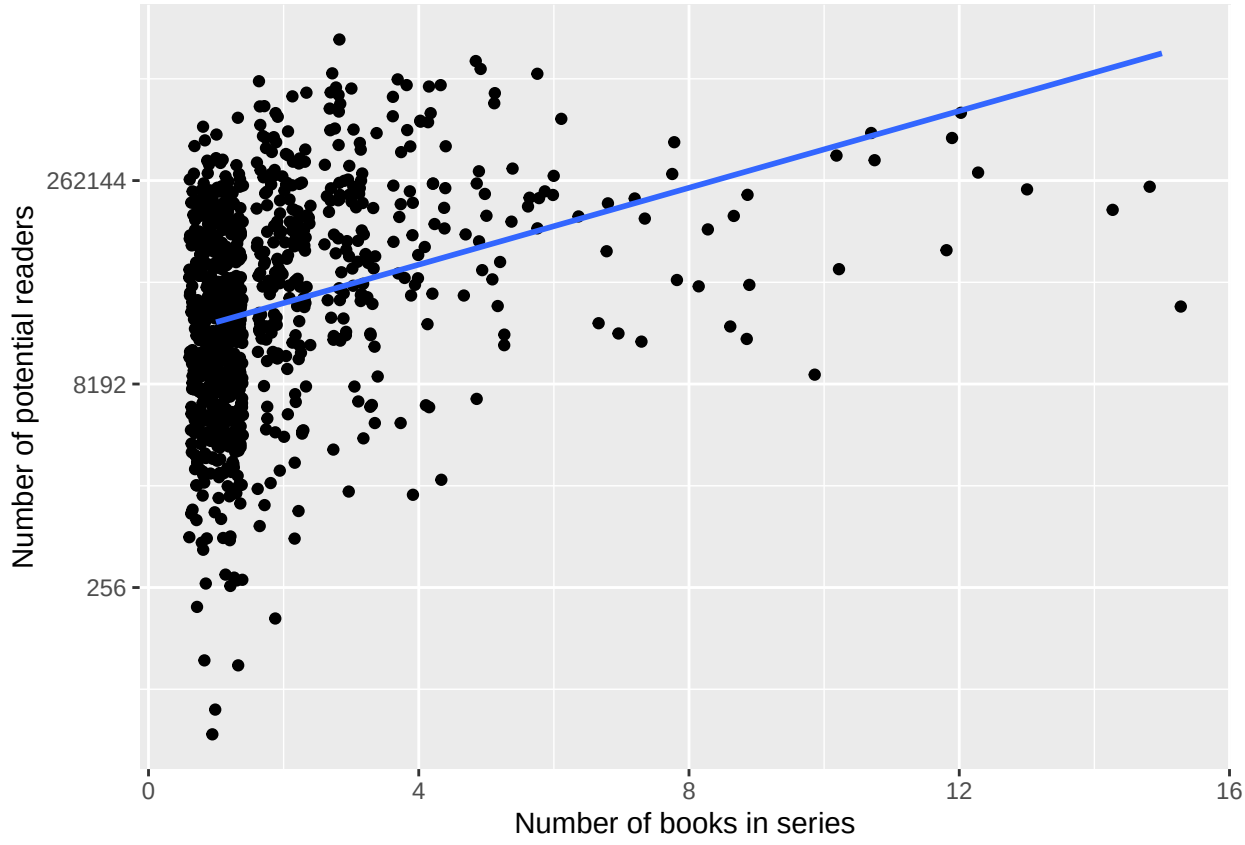
```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 4 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```




```
ggplot(series_length, aes(x = series_len, y = total_want_read)) +
  geom_jitter() +
  scale_y_continuous(transform = "log2") +
  labs(y = "Number of potential readers", x = "Number of books in series") +
  stat_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 4 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



There does not appear to be much of a linear relationship between series length and readership; below the correlation for series length and total interested readers is barely under 0.3, or fairly weak.

```
series_length |>
  summarise(cor(series_len, total_want_read, use = "complete.obs"))
```

```
## # A tibble: 1 x 1
##   `cor(series_len, total_want_read, use = "complete.obs")`
##                                                      <dbl>
## 1                                                    0.288
```

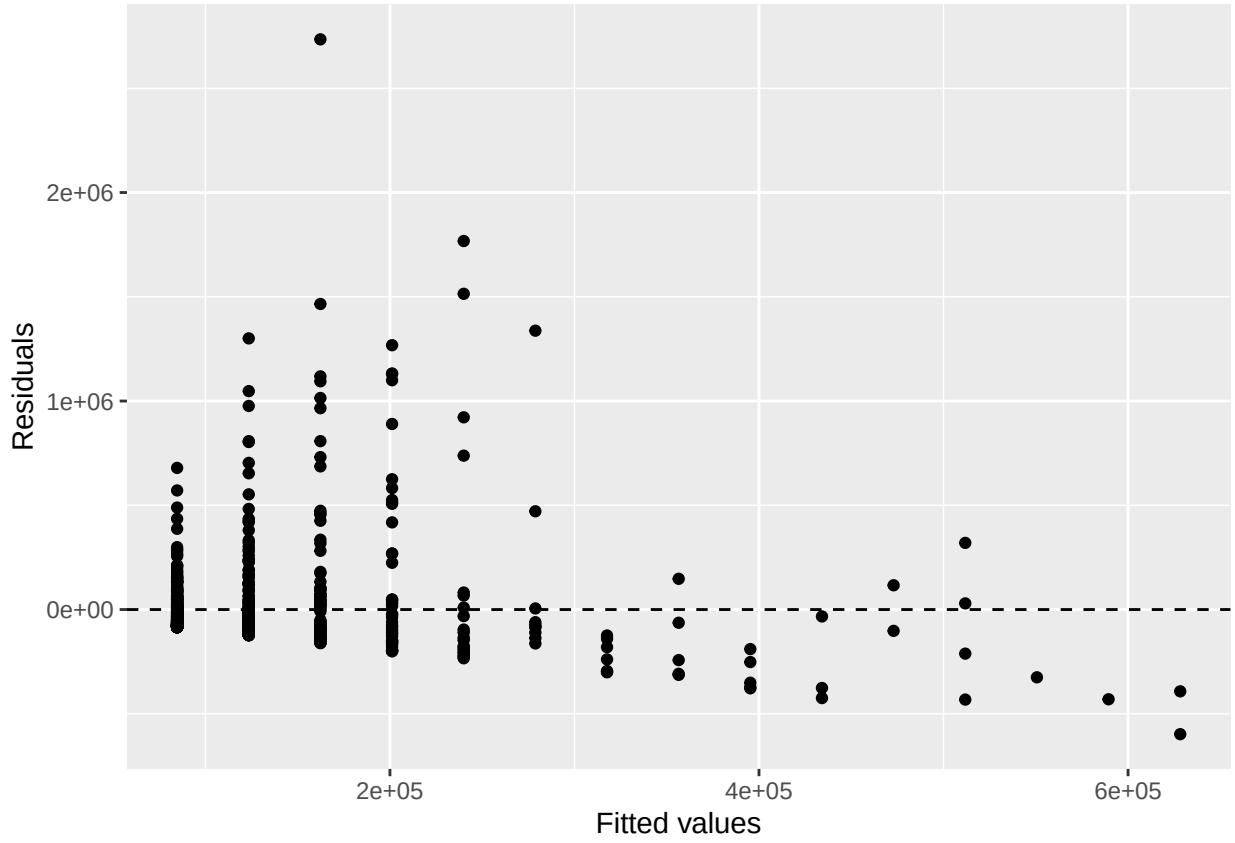
Similarly, the linear model shows a low R-squared of 0.08:

```
m_read <- lm(total_want_read ~ series_len, data = series_length)
summary(m_read)
```

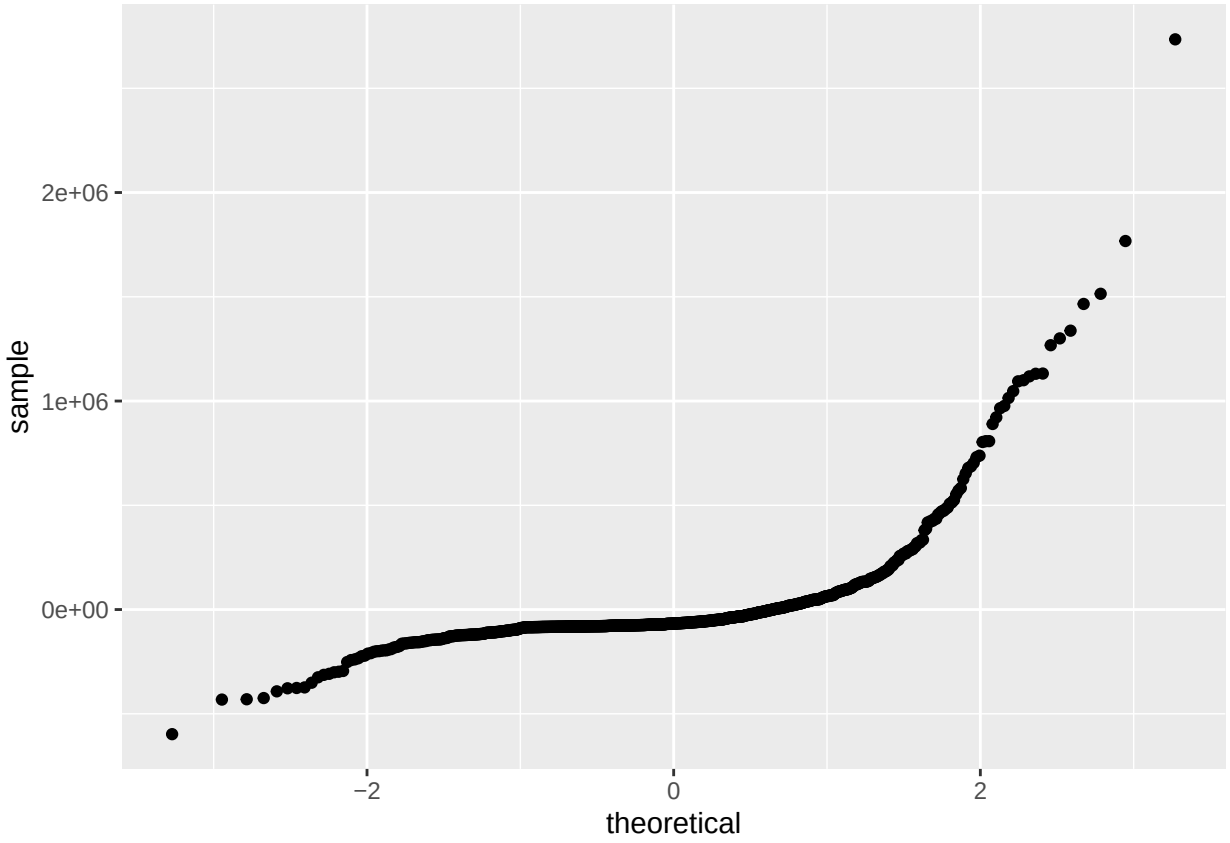
```
##
## Call:
## lm(formula = total_want_read ~ series_len, data = series_length)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -597607  -81733  -67482    4918 2734664
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    45854      11232   4.083 4.84e-05 ***
## series_len     38828       4236   9.166  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 236600 on 931 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.08277,    Adjusted R-squared:  0.08179
## F-statistic: 84.02 on 1 and 931 DF,  p-value: < 2.2e-16
```

The residual plots only further demonstrate the weak relationship. The points do not form much of a cloud around the 0 in the scatter plot and the normal probability plot is flattened in the center.

```
ggplot(data = m_read, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") + xlab("Fitted values") +
  ylab("Residuals")
```




```
ggplot(data = m_read, aes(sample = .resid)) + stat_qq()
```



Conclusion