The Influence of Book Length on Average Ratings in Goodreads

Pedro Henrique Gonçalves de Paiva

January 27, 2024

Table of Contents

The Influence of Book Length on Average Ratings in Goodreads.

1. Introduction

This report extends the "Page by Page" (Paiva, 2023) to investigate if a book's length affects its average rating, noting that books have increased in size by 25% from 2000 to 2015 (Lea, 2019).

It explores two key questions:

- Does the number of pages in a book significantly correlate with its average rating on Goodreads?
- In addition to the number of pages, does incorporating the count of text reviews enhance the predictive power of a regression model in explaining the average rating of books on Goodreads?

Employing multiple regression analysis, this study aims to deepen understanding of the elements influencing reader ratings on Goodreads, using a sample dataset to make inferences about all books on the platform.

2. Dataset

The dataset is a selection of data from Goodreads from which we are using a sample containing 200 entries (Paiva, 2023).

It's worth noting the dataset has limitations, particularly concerning its temporal relevance. Some records may not reflect the most current data, affecting the average ratings due to changing reader demographics and preferences over time.

The variables:

- **Independent variable:** number of pages. It is a discrete, quantitative variable, measured as a book's total count of pages.
- **Dependent variable:** average rating. This is the primary variable of interest. It serves as an indicator of reader engagement. It is continuous, as between two values we can always find another one.

Hypotheses:

- **Null Hypothesis:** There is no linear relationship between the number of pages and the average rating. $H0: \beta_1 = 0$ (the slope of the regression line is zero, and changes in the independent variable do not predict changes in the dependent variable).
- **Alternative Hypothesis:** There is a linear relationship between the variables. $H1: \beta_1 \neq 0$ (the slope of the regression line is not zero, and changes in the independent variable are associated with changes in the dependent variable).

3. Methods

Let us gather descriptive statistics (Figure 1) (Appendix A).

Figure 1

Descriptive Statistics

	Number of Pages	Average Rating
Count	200.00	200.00
Mean	346.47	3.95
Median	307.50	3.99
Mode	96.00	4.07
Standard Deviation	234.55	0.29
Range	1392.00	2.02

A standard deviation of 234.55 and a high range looks too big for the number of pages. Printing the highest number of pages, using "*print(df['num_pages'].max()"* gives a result of 1392, exactly the range, meaning there is a book with 0 pages. Ideally, all books need to have a number of pages. Now, we remove data with the number of pages being less than 5 and do the descriptive statistics again(Figure 2) (Appendix B).

Figure 2

Corrected Descriptive Statistics

	Number of Pages	Average Rating
Count	198.00	198.00
Mean	349.96	3.95
Median	310.00	3.99
Mode	96.00	4.07
Standard Deviation	233.12	0.29
Range	1382.00	2.02

We now should be working with correct values. [1]

[1] #descriptivestats: By analyzing mean, median, mode, standard deviation, and range, I could interpret the data's distribution and infer the general trends that inform our hypotheses. This was extremely important to notice problems that could surge, such as the outlier that I discovered using the range.

3.1. Conditions for the Model

We need to make sure the relationship of our variables meets the conditions of the regression line: L (linear), I (Independent), N (Normal Distributed), and E (Equal Variances).

To analyze the linearity, we develop a line graph that takes the average rating and number of pages as our variables (Figure 3) (Appendix C). This visualization shows us that we indeed can find a linear relationship.

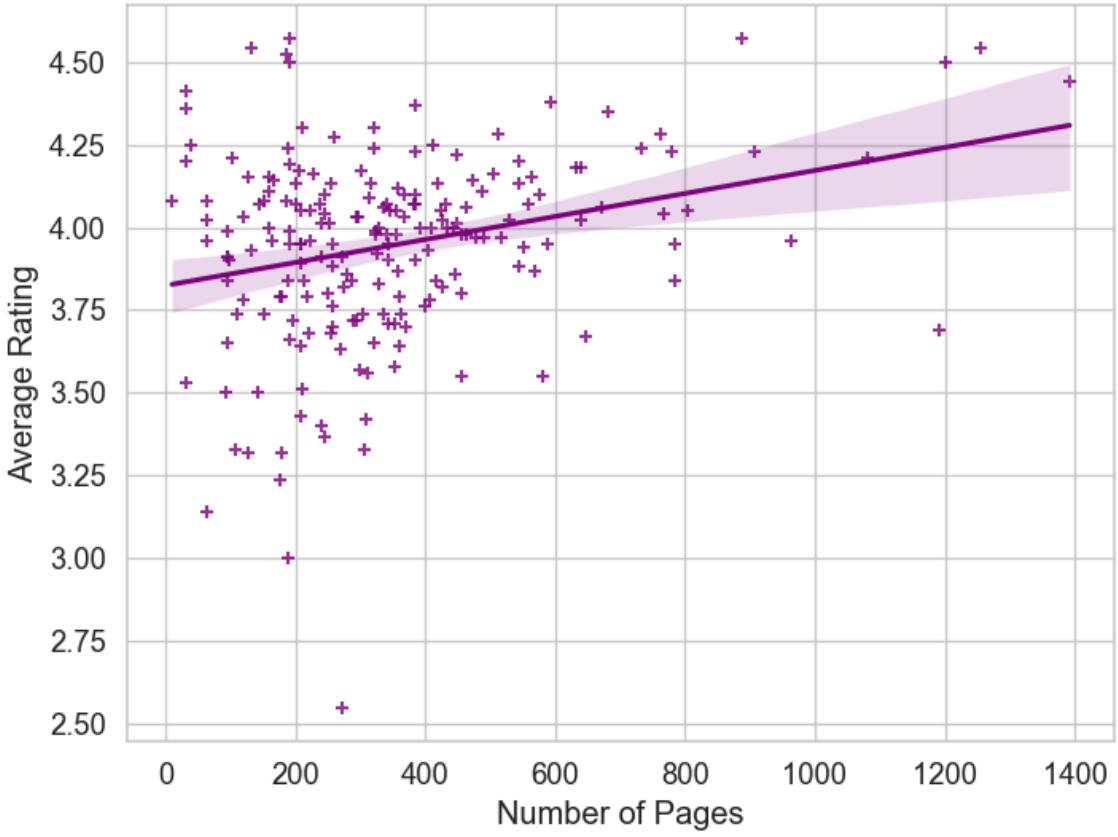

Figure 3. The scatterplot shows the relationship between the number of pages and average ratings of books on Goodreads, indicating a slight positive trend where books with a higher page count may receive higher ratings.

To analyze both independence and equal variance, we plot a graph with the residuals and fitted values (Figure 4) (Appendix D).

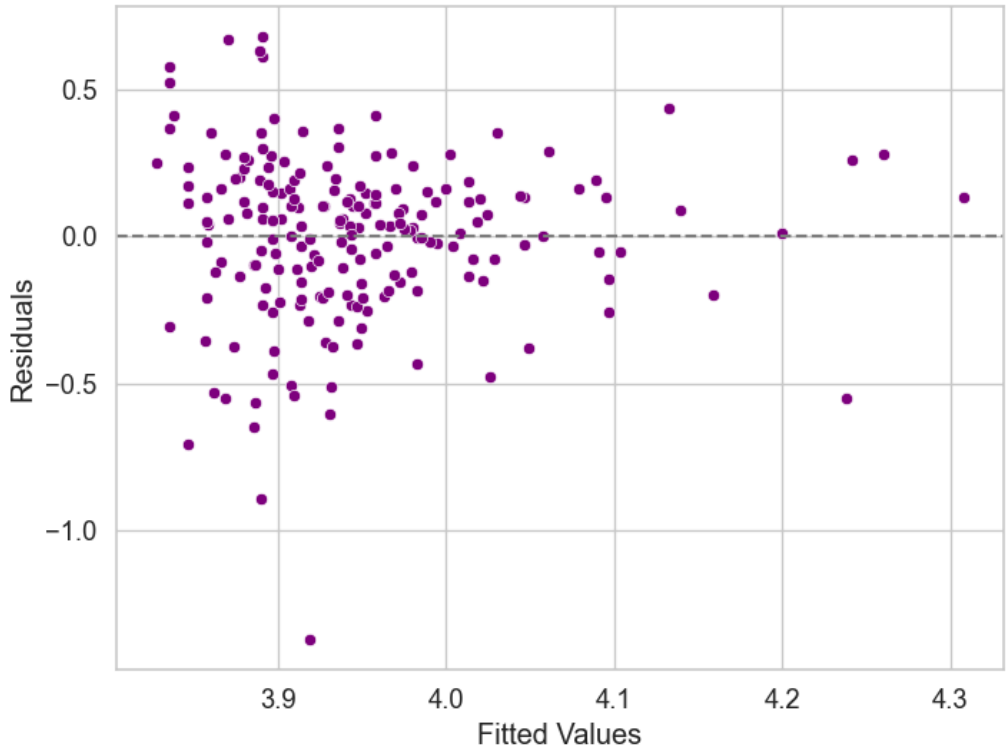

Figure 4. Residuals vs Fitted Values. Scatterplot of residuals versus fitted values for the regression model, showing no apparent patterns, which suggests that the assumptions of linearity and equal variance are reasonably met.

This graph shows we do not have a clear pattern that satisfies the condition for independence for the residuals. Additionally, ensuring the independence of observations, our sample size of 200 books represents only 0.000067% of Goodreads' 300 million books (*Goodreads Catalogs 300 Millionth Book - Goodreads News & Interviews*, n.d.).

Also, the spread or variance of the residuals does not change dramatically across the range of fitted values, satisfying homoscedasticity.

Lastly, we need to consider if our residuals are normally distributed. For this, we create a histogram with the residuals and their frequency (Figure 5) (Appendix E).

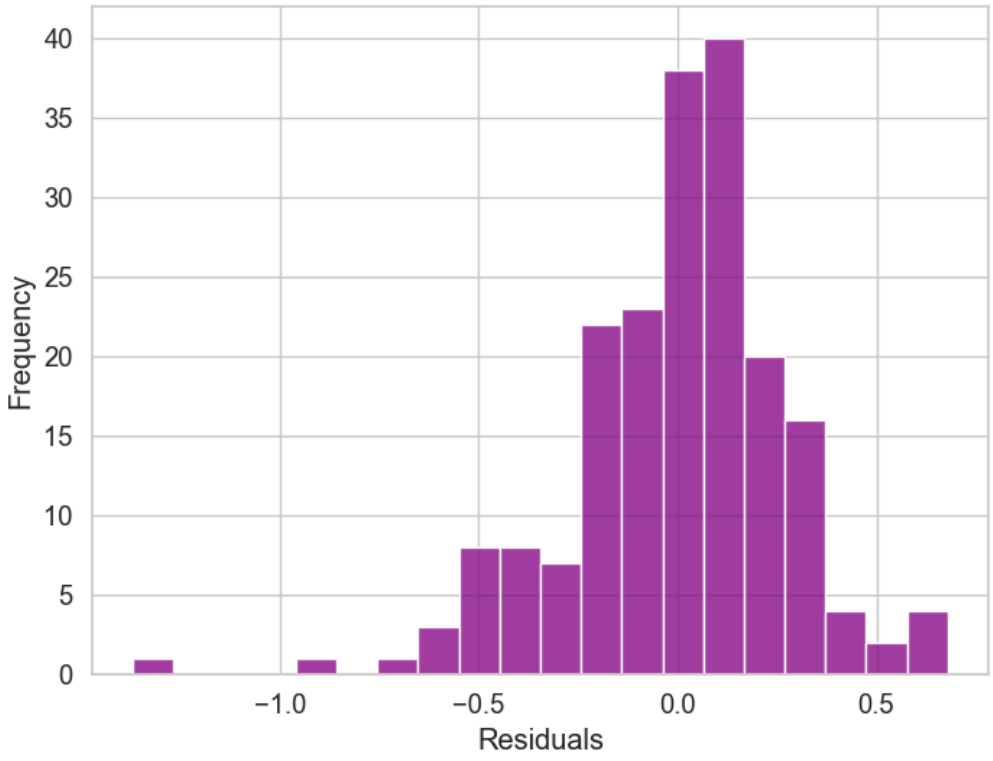

Figure 5. Histogram of Residuals. The histogram illustrates the distribution of residuals derived from the linear regression model. The distribution is bell-shaped, which approximates to a normal curve.

Our histogram looks roughly bell-shaped and symmetric, suggesting that the data is approximately normally distributed, excluding the outlier below -1.

With this, we can conclude all of the conditions of LINE are met, making it possible to proceed.

3.2. Pearson's r

To analyze the relationship between our variables, we use Pearson's r, as our variables fit the LINE conditions, aligning well with Pearson's r requirements.

We are using the formula $r = \frac{1}{n}\Sigma\frac{(x_1 - \bar{x})(y_1 - \bar{y})}{\sigma_x \sigma_y}$, in which:

- n: Data points in the dataset

- x_1 and y_1: Data points for each variable
- $\overline{x}$ and $\overline{y}$: Mean of the variables
- σ_x and σ_y: Standard deviation of the variables

Using Python for accuracy, we obtained a Pearson's r of 0.279 (Appendix F). According to Cohen's d guidelines, this is classified as a weak to moderate positive correlation. It implies a slight trend where books with more pages tend to have higher average ratings. However, this correlation coefficient does not indicate the statistical significance of the relationship nor imply causation.[2]

3.3. Coefficient of Determination

To calculate the R-squared, we use the formula $R^2 = 1 - \frac{SSE}{SSTO}$, where *SSE* (sum of squares due to error) = $\Sigma(y_i - \hat{y}_i)^2$, and *SSTO* (total sum of squares) = $\Sigma(y_i - \overline{y})^2$. However, as we calculated Pearson's r before, we can just do R-squared = r^2, which would be R-squared = 0.279^2 and therefore a result of approximately 0.078.

3.4. Regression Equation

First, we have the formula $y = \beta_0 + \beta_1 x + \epsilon$, where:

- y: Dependent variable
- x: Independent variable
- β_0: Y-intercept of the regression line

[2] #correlation: Utilizing Pearson's r, I quantified the strength of the linear relationship between the number of pages in books and their average ratings on Goodreads. I relied on Python for the calculation, having 0.279. The weak to moderate positive correlation implies a slight tendency for books with more pages to secure higher average ratings.

- β_1: Slope of the regression line
- ϵ: Error term

Using Python, we find the formula $y = 0.0 * numpages + 3.824$ (Appendix G). While initially it appeared as if the slope was 0.0, a more precise calculation revealed it to be 0.000347 (Appendix H). This shows us that we have a positive but very weak change; for every 100 additional pages, the average will increase by only 0.03.

Since the slope is so small, the model suggests that variations in the number of pages have an almost negligible effect on the average rating, meaning that predictions using this model should be taken with caution. Additionally, it is important to notice that having a y-intercept of 3.824 is used for the calculations only, as it is impossible for a book to have 0 pages.[3]

3.5. Statistical Significance

To complement our past analysis and understand if we should reject our null hypothesis, we will calculate the confidence interval for the slope and its p-value.

Setting our confidence level at the standard 95%, we used Python to compute the confidence interval for the slope, obtaining a range of [0.000179, 0.000516] (Appendix I). This interval confidently suggests that the true slope value lies within this range, and since it does not include zero, it indicates that the slope is statistically significant. However, the small magnitude of the interval values underscores that, while statistically significant, the effect of the number of pages on the average rating is relatively minor in practical terms. This reinforces the conclusion

[3] #regression: I applied regression analysis to understand the influence of book length on ratings. I quantified the relationship using the R-squared value, which indicates a low degree of variability in average ratings explained by the number of pages. This understanding underscores regression's role in predictive modeling while acknowledging its limitations in establishing causality or the predictive power in varied contexts. I also delved into the regression formula, looking for the specific slope value and interpreting it.

that page count, though related to average ratings, has a limited practical impact.[4]

For the calculation of the p-value, we will consider the significance level of $\alpha = 0.05$, meaning we are allowing a 5% probability of incorrectly rejecting the null hypothesis. Using Python, we discovered that our p-value is 6.96^{-5}, which is extremely small and definitely less than our significance level ($p < \alpha$) (Appendix J). This value measures the probability of getting a sample with a linear relationship this strong or stronger, given that the null hypothesis is true. Because of this, we can reject the null hypothesis and imply that the relationship observed is very unlikely to be due to chance.

4. Multiple Regression

To explore how multiple factors affect average ratings, we will add another independent variable to our study. We will use a correlation matrix to identify the variable with the highest correlation coefficient (r) for further analysis (Appendix K).

[4] #confidenceintervals: I determined the 95% confidence interval for the regression slope, as it is the standard in statistics. I looked into the importance of not containing zero. Such analysis underscores my commitment to rely on point estimates and consider the precision of my estimates and the statistical significance of my findings.

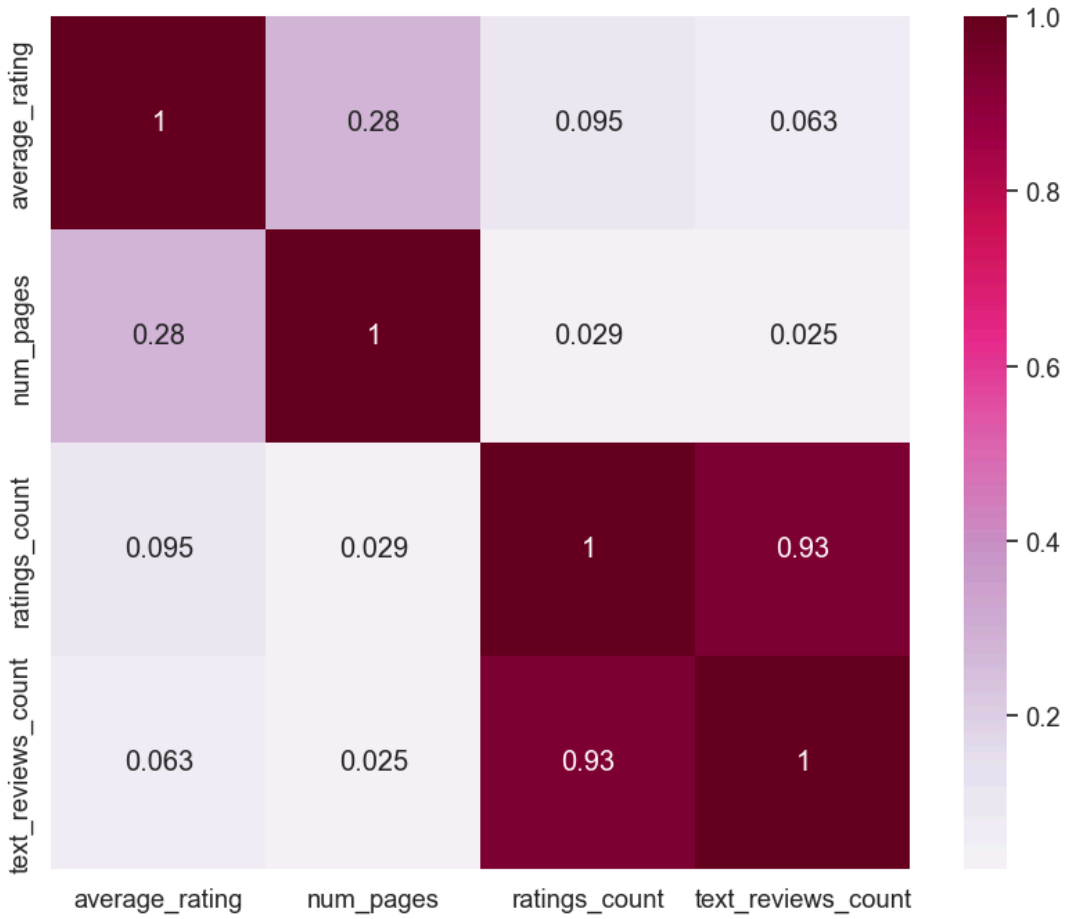

Figure 6. Correlation Matrix Table. This table displays the Pearson's r correlation coefficients between variables, providing insights into the strength and direction of relationships among the variables under the Goodreads dataset.

Analyzing Pearson's r for various variables against the average rating, we found the number of pages to have the highest correlation. Following this, the rating count emerged as the second highest, making it our choice for inclusion in the regression analysis. We can conclude that there is no evidence of multicollinearity between our independent variables, num_pages and ratings_count, because their r coefficient is 0.029, which is very low.

To see if this relationship is also following the LINE condition, we can plot the figures again with the added variable (Figures 7, 8, and 9)(Appendix L, M, and N).

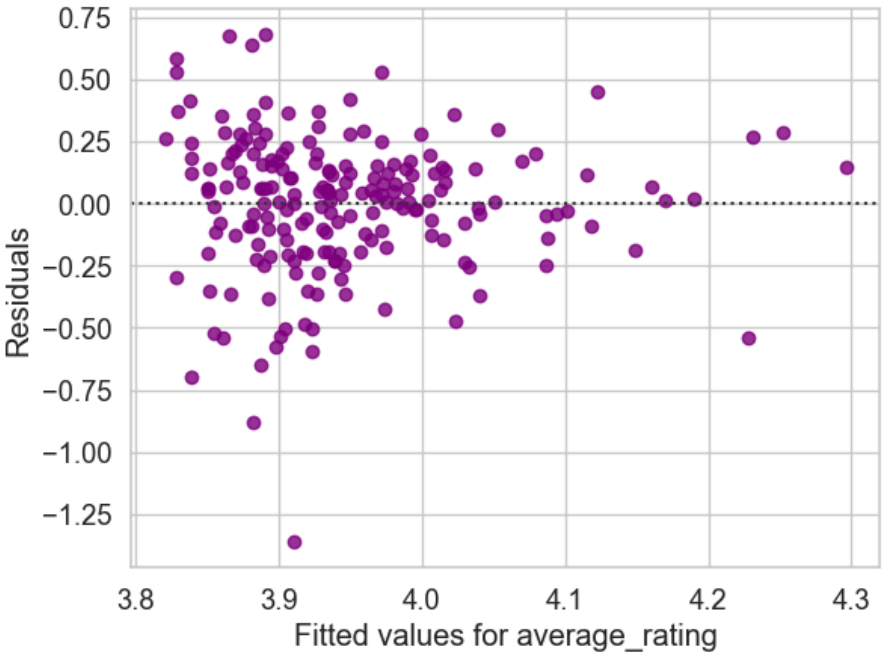

Figure 7. Residuals and Fitted Values. The plot indicates no apparent patterns. The homoscedasticity of residuals across the range of fitted values suggests that the variance of the residuals is constant. There are some outliers visible, such as the point (3.91, -1.27).

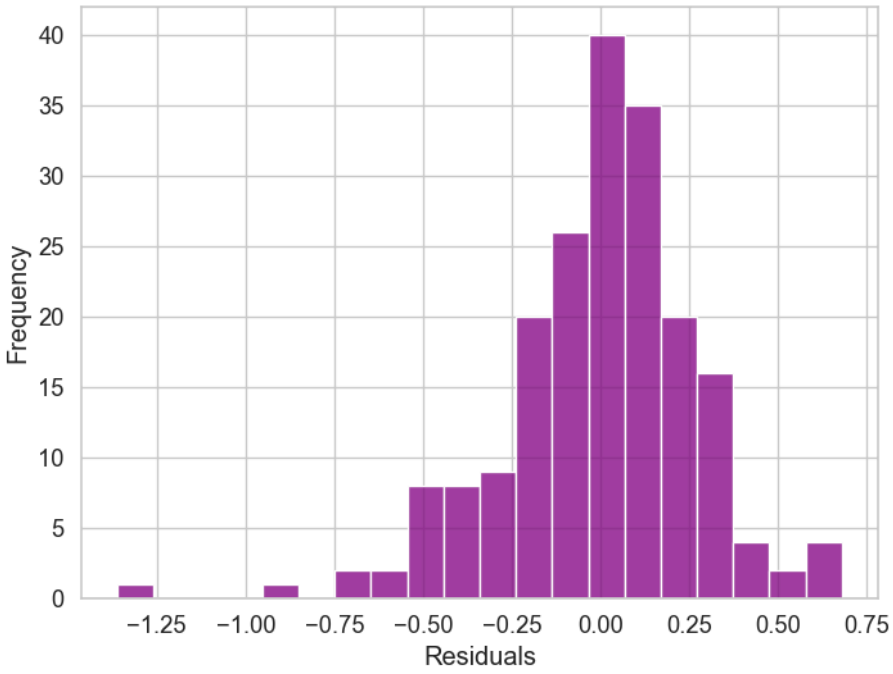

Figure 8. Histogram of Residuals. The near-symmetric, bell-shaped distribution suggests normality of residuals. The presence of a few outliers on the left indicates instances of larger-than-expected prediction errors.

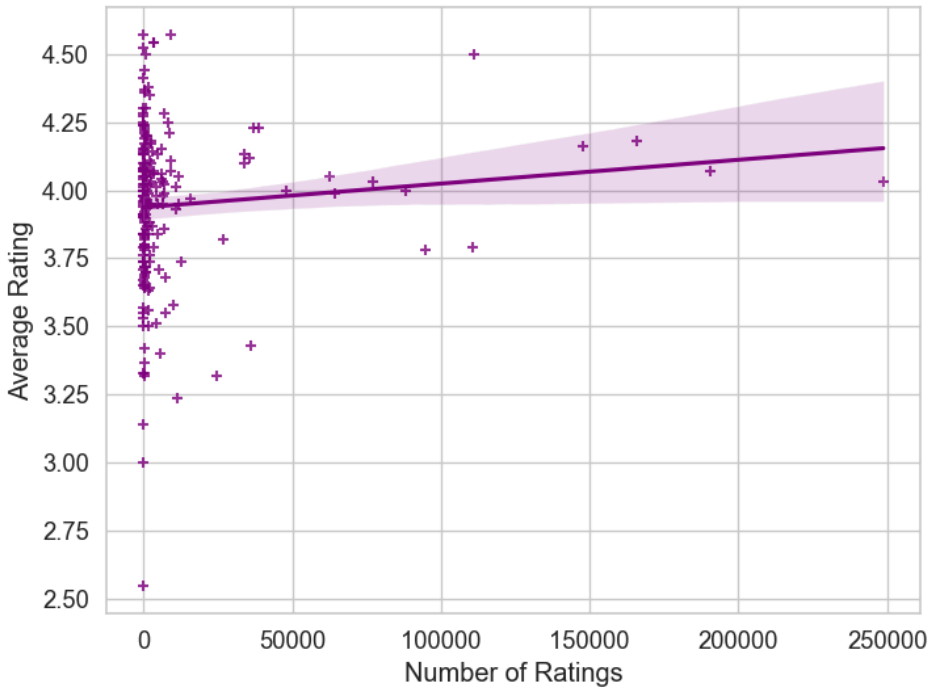

Figure 9. Linearity for Rating Count. There is a regression line indicating a positive trend and a significant clustering of data points near the origin. Despite the wide range in the number of ratings, the average rating tends to hover around 4.00

The residual plots indicate independence and near-homoscedasticity, with a histogram suggesting normal distribution despite a single outlier. However, the rating count plot shows that the values are clustering around the lower ratings, requiring zoomed analysis for clarity (Figure 10)(Appendix O).

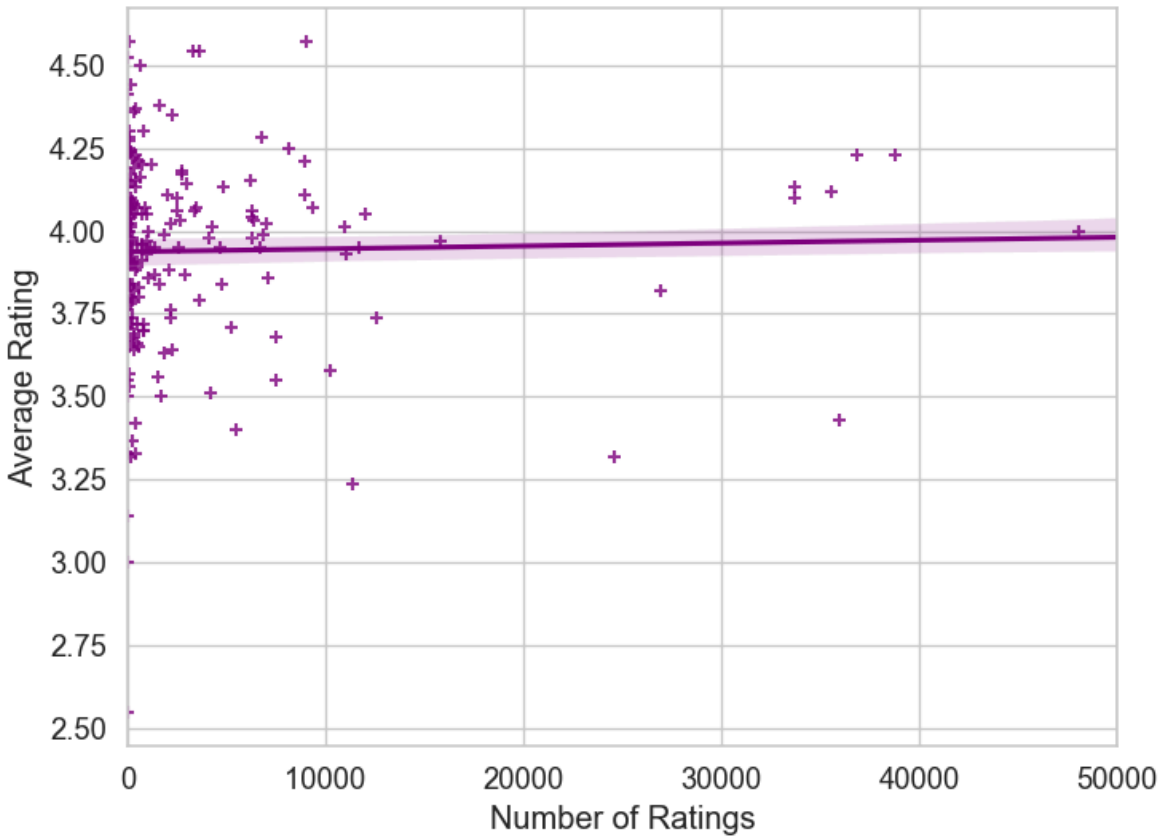

Figure 10. Linearity for Rating Count (zoomed in). There is a dense clustering of data points at lower numbers of ratings. The average ratings predominantly range between 3.5 and 4.5, with a nearly horizontal regression line.

The visualization reveals a positive but nearly flat slope, indicating a minimal linear relationship between our variables. The precise slope calculation, close to 8^{-7}, supports this finding (Appendix P).

To deepen our analysis, we'll calculate the R-squared, derive the regression equation, and determine the confidence interval and p-value to assess the suitability of including multiple variables (Appendix Q).

The model's R-squared value of 0.085 indicates that about 8.5% of the variability in average ratings is explained by the independent variables used. Compared to a previous R-squared of 0.078, the inclusion of additional variables offers a small improvement, indicating

that ratings_count did not substantially increase the explanatory power of the model.

The regression equation is

$$average\ rating = 3.8173 + (0.0000 * ratings\ count) + (0.0003 * num\ pages),$$

meaning when ratings_count and num_pages are zero, the estimated average rating is 3.8173. Both coefficients close to zero suggest these variables have a very small positive effect on the average rating, especially the ratings count.

The confidence interval for the ratings count, [-4.45^{-7}, 0.000002], includes 0, suggesting that the true effect of ratings_count on average_rating might be negligible. With a p-value around 0.2, which is higher than 0.05 ($p > \alpha$), ratings count appears not to significantly affect the average rating.

With all of this, we can say that the model would be better if we kept only the number of pages as the independent variable and did not include the ratings count.

5. Conclusion

Through multiple regression analysis, we found a weak positive correlation (Pearson's r = 0.279) between page count and average ratings, suggesting longer books may have slightly higher ratings. With our regression equation with a single independent variable, writers can make predictions about how many pages they would have to write in their books to have a specific average rating; however, with a marginal slope of 0.0003, book length is not a strong predictor of average ratings.

Additionally, the model's R-squared of 0.085 with multiple variables reflects only a slight improvement in fit over the previous value of 0.078, suggesting other variables, such as genre, may be more influential in determining average ratings. The ratings count's confidence interval

including zero and a p-value of 0.2, implies that its impact on average ratings is not significant. These findings suggest the model would be more precise if focused solely on the number of pages.

Our analysis is subject to limitations, including potential bias from dated information and omitted variables. The derived model suggests trends within the sampled data but does not establish causality.

Lastly, the conclusions drawn from our analysis of the Goodreads dataset are based on inductive reasoning. Our analysis started with specific observations and, from these, we inferred a broader relationship applicable to the wider population of books on Goodreads. However, these inferences are probabilistic rather than certain, meaning the patterns observed in our sample trends might hold true in a larger context. Our model's predictive power is limited, as indicated by the low R-squared value. Because of this, it should be applied with caution, and its predictions are best viewed as tendencies rather than certainties.

Reference

Goodreads Catalogs 300 millionth Book - Goodreads News & Interviews. (n.d.). Goodreads. https://www.goodreads.com/blog/show/359-goodreads-catalogs-300-millionth-book

Lea, R. (2019, August 7). The big question: are books getting longer? *The Guardian*. https://www.theguardian.com/books/2015/dec/10/are-books-getting-longer-survey-marlon-james-hanya-yanagihara

Paiva, P. (2023). Page by Page: The Influence of Book-Length on Reader Ratings on Goodreads. Formal Analysis 50, Minerva University.

Appendix

Appendix A

```
#DESCRIPTIVE STATS CLEAN (APPENDIX A)
# Create an empty dictionary to store descriptive statistics
descriptive_stats = {}

# Calculate descriptive statistics for 'Number of Pages' column
descriptive_stats['Number of Pages'] = {
    'Count': df['num_pages'].count(),
    'Mean': df['num_pages'].mean(),
    'Median': df['num_pages'].median(),
    'Mode': df['num_pages'].mode()[0],
    'Standard Deviation': df['num_pages'].std(),
    'Range': df['num_pages'].max() -  df['num_pages'].min()
}

# Calculate descriptive statistics for 'Average Rating' column
descriptive_stats['Average Rating'] = {
    'Count': df['average_rating'].count(),
    'Mean': df['average_rating'].mean(),
    'Median': df['average_rating'].median(),
    'Mode': df['average_rating'].mode()[0],
    'Standard Deviation': df['average_rating'].std(),
    'Range': df['average_rating'].max() -  df['average_rating'].min()
}

# Create a DataFrame from the dictionary of descriptive statistics
df_stats = pd.DataFrame.from_dict(descriptive_stats, orient='index').transpose().round(2)

# Print the DataFrame containing descriptive statistics
print(df_stats)

# Print the highest number of pages in the dataset
print("The Highest number of pages is", df['num_pages'].max())
```

Appendix B

```
#DESCRIPTIVE STATS CLEAN (APPENDIX B)
df = df[df['num_pages'] > 5]

descriptive_stats = {}
descriptive_stats['Number of Pages'] = {
    'Count': df['num_pages'].count(),
    'Mean': df['num_pages'].mean(),
    'Median': df['num_pages'].median(),
    'Mode': df['num_pages'].mode()[0],
    'Standard Deviation': df['num_pages'].std(),
    'Range': df['num_pages'].max() -  df['num_pages'].min()
}

descriptive_stats ['Average Rating']= {
    'Count': df['average_rating'].count(),
    'Mean': df['average_rating'].mean(),
    'Median': df['average_rating'].median(),
    'Mode': df['average_rating'].mode()[0],
    'Standard Deviation': df['average_rating'].std(),
    'Range': df['average_rating'].max() -  df['average_rating'].min()
}
descriptive_stats

df_stats = pd.DataFrame.from_dict(descriptive_stats, orient='index').transpose().round(2)
print(df_stats)
```

Appendix C

```
#LINEAR REGRESSION LINE (APPENDIX C)
# style settings
sns.set(color_codes=True, font_scale=1.2)
sns.set_style("whitegrid")
data = df
# Define a function for simple linear regression plot
def simple_linear_regression(column_x, column_y, xlabel, ylabel):
    plt.figure(figsize=(8, 6))
    sns.regplot(x=column_x, y=column_y, data=data, marker="+", color = 'purple')
# Set labels for the x-axis and y-axis
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.figtext(0.0, -0.05, 'Figure 3. The scatterplot shows the relationship between the number of pages and averag
                ha='left', fontsize=11)
    plt.show()

simple_linear_regression("num_pages", "average_rating", "Number of Pages", "Average Rating")
```

Appendix D

```
##RESIDUALS AND FITTED VALUES (APPENDIX D)
# style settings
sns.set(color_codes=True, font_scale=1.2)
sns.set_style("whitegrid")
data = df

def plot_residuals_vs_fitted(column_x, column_y):
    # fit the regression line using "statsmodels" library
    X = sm.add_constant(data[column_x])
    Y = data[column_y]
    regression_model = sm.OLS(Y, X).fit()

    # Extract fitted values and residuals
    fitted_values = regression_model.fittedvalues
    residuals = regression_model.resid

    # Create a plot of residuals vs fitted values
    plt.figure(figsize=(8, 6))
    sns.scatterplot(x=fitted_values, y=residuals, color = 'purple')
    plt.axhline(y=0, color='grey', linestyle='--')
    plt.xlabel('Fitted Values')
    plt.ylabel('Residuals')
    plt.figtext(0.0, -0.05, 'Figure 4. Residuals vs Fitted Values. Scatterplot of residuals versus fitted values for
                ha='left', fontsize=11)
    plt.show()

plot_residuals_vs_fitted("num_pages", "average_rating")
```

Appendix E

```
#RESIDUALS HISTOGRAM (APPENDIX E)
sns.set(color_codes=True, font_scale=1.2)
sns.set_style("whitegrid")
data = df

def histogram_of_residuals(column_x, column_y):
    # fit the regression line using "statsmodels" library:
    X = sm.add_constant(data[column_x])
    Y = data[column_y]
    regression_model = sm.OLS(Y, X).fit()  # OLS = "ordinary least squares"

    # create a histogram of the residuals
    plt.figure(figsize=(8, 6))
    sns.histplot(regression_model.resid, kde=False, color='purple')
    plt.xlabel('Residuals')
    plt.ylabel('Frequency')
    plt.figtext(0.05, -0.08, 'Figure 5. Histogram of Residuals. The histogram illustrates the distribution of residu
                ha='left', fontsize=11)

    plt.show()
histogram_of_residuals("num_pages", "average_rating")
```

Appendix F

```
#PEARSON'S R (APPENDIX F)
print("\nThe pearson's r value comparing the number of pages to the average rating is: ",round(df["num_pages"].corr(
```

Appendix G

```
#REGRESSION EQUATION (APPENDIX G)
def regression_model(column_x, column_y):
    # fit the regression line using "statsmodels" library:
    X = statsmodels.add_constant(data[column_x])
    Y = data[column_y]
    regressionmodel = statsmodels.OLS(Y, X).fit()  # OLS = "ordinary least squares"

    # extract regression parameters from model, rounded to 3 decimal places:
    slope = round(regressionmodel.params[1], 3)
    intercept = round(regressionmodel.params[0], 3)

    print("Regression equation: " + column_y + " = ", slope, " * " + column_x + " + ", intercept)

regression_model("num_pages", "average_rating")
```

Appendix H

```
#SLOPE (APPENDIX H)

X = sm.add_constant(data['num_pages'])  # This adds a constant term to our predictor
Y = data['average_rating']

# Fit the OLS regression model
regression_model = sm.OLS(Y, X).fit()

# Get the slope (β1)
slope = regression_model.params['num_pages']

print(f"The slope of num_pages is: {slope}")
```

Appendix I

```
#CONFIDENCE INTERVAL(APPENDIX I)
X = sm.add_constant(data['num_pages'])  # This adds the intercept term
Y = data['average_rating']

# Fit the OLS regression model
model = sm.OLS(Y, X).fit()

# Get the confidence interval for the slope
confidence_interval = model.conf_int(alpha=0.05).loc['num_pages']

print(f"95% confidence interval for the slope: {confidence_interval}")
```

Appendix J

```
#P-VALUE (APPENDIX J)
p_value = model.pvalues['num_pages']
print(f"p-value for the slope: {p_value}")
```

Appendix K

```
#CORRELATION MATRIX (APPENDIX K)
columns_to_include = ['average_rating', 'num_pages', 'ratings_count', 'text_reviews_count']
selected_data = data[columns_to_include]

#use .corr() to create the matrix
correlation_matrix = selected_data.corr()
print(correlation_matrix)

# Plot the heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='PuRd')
plt.figtext(0.1, -0.03, 'Figure 6. Correlation Matrix Table. This table displays the Pearson\'s r correlation coeffi
    ha='left', fontsize=12)
plt.show()
```

```
                   average_rating  num_pages  ratings_count  \
average_rating           1.000000   0.278800       0.094793
num_pages                0.278800   1.000000       0.028917
ratings_count            0.094793   0.028917       1.000000
text_reviews_count       0.063256   0.024841       0.931535

                   text_reviews_count
average_rating                 0.063256
num_pages                      0.024841
ratings_count                  0.931535
text_reviews_count             1.000000
```

Appendix L

```
#LINE CONDITIONS- RESIDUALS VS FITTED VALUES (APPENDIX L)
def plot_residuals_vs_fitted(column_x, column_y, data):
    # Define predictors X and response Y
    X = data[column_x]
    X = sm.add_constant(X)
    Y = data[column_y]

    # Construct the regression model
    regression_model = sm.OLS(Y, X).fit()

    # Create a residual plot
    plt.figure()
    residuals = regression_model.resid
    fitted_values = regression_model.predict()
    sns.residplot(x=fitted_values, y=residuals, color='purple')
    plt.xlabel('Fitted values for ' + column_y)
    plt.ylabel('Residuals')
    plt.figtext(0.0, -0.14, 'Figure 7. Residuals and Fitted Values. The plot indicates no apparent patterns. The homoscedasticity of\nresidual
                ha='left', fontsize=11)

plot_residuals_vs_fitted(['num_pages', 'ratings_count'], 'average_rating', data)
```

Appendix M

```
#LINE CONDITIONS- SLOPE VARIABLE ADDED (APPENDIX M)
sns.set(color_codes=True, font_scale=1.2)
sns.set_style("whitegrid")
data = df

def histogram_of_residuals(column_x, column_y):
    # fit the regression line using "statsmodels" library:
    X = sm.add_constant(data[column_x])
    Y = data[column_y]
    regression_model = sm.OLS(Y, X).fit()  # OLS = "ordinary least squares"

    # create a histogram of the residuals
    plt.figure(figsize=(8, 6))
    sns.histplot(regression_model.resid, kde=False, color='purple')
    plt.xlabel('Residuals')
    plt.ylabel('Frequency')
    plt.figtext(0.0, -0.05, 'Figure 8. Histogram of Residuals. The near-symmetric, bell-shaped distribution suggests normality of residuals.\n
                ha='left', fontsize=11)

    plt.show()
histogram_of_residuals(['num_pages', 'ratings_count'], 'average_rating')
```

Appendix N

```
#LINE CONDITIONS- RESIDUALS HISTOGRAM (APPENDIX N)
# style settings
sns.set(color_codes=True, font_scale=1.2)
sns.set_style("whitegrid")
data = df

def simple_linear_regression(column_x, column_y, xlabel, ylabel):
    # Create a scatter plot with a linear regression line
    plt.figure(figsize=(8, 6))
    sns.regplot(x=column_x, y=column_y, data=data, marker="+", color = 'purple')

    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.figtext(0.0, -0.08, 'Figure 9. Linearity for Rating Count. There is a regression line indicating a positive trend and a significant\nc
                ha='left', fontsize=11)

    plt.show()
simple_linear_regression("ratings_count", "average_rating", "Number of Ratings", "Average Rating")
```

Appendix O

```
#LINE CONDITIONS- RESIDUALS HISTOGRAM ZOOMED IN(APPENDIX O)
# style settings
sns.set(color_codes=True, font_scale=1.2)
sns.set_style("whitegrid")
data = df

def simple_linear_regression(column_x, column_y, xlabel, ylabel):
    # Create a scatter plot with a linear regression line
    plt.figure(figsize=(8, 6))
    sns.regplot(x=column_x, y=column_y, data=data, marker="+", color = 'purple')

    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.xlim(0, 50000)
    plt.figtext(0.0, -0.05, 'Figure 10.  Linearity for Rating Count (zoomed in). There is a dense clustering of data points at lower numbers
                ha='left', fontsize=11)
    plt.show()
simple_linear_regression("ratings_count", "average_rating", "Number of Ratings", "Average Rating")
```

Appendix P

```
##EXACT VALUE OF THE SLOPE (APPENDIX P)
X = sm.add_constant(data[['num_pages', 'ratings_count']])  # Add a constant for the intercept
Y = data['average_rating']
model = sm.OLS(Y, X).fit()

# Get the full value of the slopes for each independent variable
slopes = model.params[1:]

# Print the full value of the slopes
for i, var in enumerate(X.columns[1:]):  # Skip the constant
    print(f"The full slope for {var} is: {slopes[i]}")
```

Appendix Q

```
##R-SQUARED FOR MULTIPLE VARIABLES (APPENDIX Q)
X = sm.add_constant(X)
model = sm.OLS(Y, X).fit()
r_squared = model.rsquared
print(f"The R-squared value of the model is: {r_squared}")

##########REGRESSION EQUATION###########
independent_vars = ['ratings_count', 'num_pages']
dependent_var = 'average_rating'

# Prepare the input data
X = data[independent_vars]
Y = data[dependent_var]
X = sm.add_constant(X)  # Add a constant to the model for the intercept

# Fit the OLS regression model
model = sm.OLS(Y, X).fit()

# Get the model's parameters
intercept, slopes = model.params[0], model.params[1:]

# Construct the regression equation as a string
regression_equation = f"{dependent_var} = {intercept:.4f}"
for slope, var in zip(slopes, independent_vars):
    regression_equation += f" + ({slope:.4f} * {var})"

# Print out the regression equation
print('\nregression equation: ', regression_equation)

######### Calculate the confidence intervals ########
confidence_intervals = model.conf_int()
print('\nconfidence interval: ', confidence_intervals)

#####Calculate the p-values######
p_values = model.pvalues
print('\np-values: ', p_values)
```