MATH 1311 A: Introduction to Data Science

Mount Allison University

Instructor: Dr. Matthew Betti, PhD

Name: Jean Michel Mugabe

Date: 15th April 2023.

Final Project

Report

Title: The Impact of Genres and Types of Publishers on Book Sales, Ranking, Rating, and Reviews. Who Benefits More: Authors of books or Publishers and Sellers of the books?

Introduction:

The dataset contains information about books sold by publishers, including their genres, sales figures, and review ratings. The aim of the report is to provide insights into the sales of books patterns and answer some important questions about the data.

Background Information:

The dataset contains information about book sales, including the genres, sellers, publishers, types of publishers, daily publishers' revenue, daily gross sales, daily Amazon revenue, daily average revenue, units sold daily, publisher name, and statistics such as average rating, sales prices, sales rank, and total reviews. The data was collected daily in the book industry across different genres, taking into consideration the publishers. However, the dataset does not include the names of the books or authors; instead, it emphasizes the publishers and genres of the books. The data can be used to determine how the sale price of the books according to the genres affects the ratings of the books. Overall, how do genres and publisher types affect genres' average ratings, daily average gross sales, and reviews of the books. Additionally, the data can be used to determine which

publishers sell the most units in a day and whether the publisher who sold the books or the type of publisher that published the books benefits more than the authors.

I would like to focus on the genres, sale price, and publisher type and types as to whether these categories play a big role in affecting the daily average gross sales, average rating, and reviews of the books and whether the seller who sold the books or the type of publisher that published the books benefits more than the authors by answering important questions such as which genre has the highest average daily Amazon revenue, the highest average statistics average rating, the highest average total reviews, the highest average daily average gross sales, the highest average sale price and the highest average daily average units sold. The report also explores whether there is a strong relationship between publisher type and units sold, which is the most popular genre among the publishers included in the dataset, does the seller who sold the books or the type of publisher that published the books benefit more than the authors in terms of revenue, how does the publisher type affect the sales, ranking, rating, and reviews of the books, how does the average sale price vary by publisher type? and which publisher has the highest daily average revenue and units sold? Additionally, the report examines the relationship between average rating and sales rank and determines the most popular publisher type. Finally, the report investigates the relationships between the price and rating of books.

The data contains a significant number of details. It contains 13 columns that represent different aspects of the books that were published and sold. Below is an explanation of what each column represents, based on my interpretation of the data:

genre: This column represents the genre or category of a book in the dataset. It may include genres such as fiction, non-fiction, romance, thriller, etc.

sold by: This column represents the name of the seller who sold the book.

“daily average.amazon revenue”: This column represents the daily average amount of revenue generated by Amazon from daily sales of the book. This report uses daily average amazon revenue instead.

“daily average.author revenue”: This column represents the daily average revenue earned by the author of the book. The term daily average author revenue would be used in this report instead.

“daily average.gross sales”: This column represents the total amount of revenue generated from daily sales of the book. In the report I would it called daily average gross sales.

“daily average.publisher revenue”: This column represents the daily average revenue earned by the publisher of the book. In the report I would it called the daily average publisher.

“daily average.units sold”: This column represents the number of units of the book sold daily. In the report I would it called daily average units sold.

publisher.name: This column likely represents the name of the publisher who published the book. In the report I would it called the publisher name.

“publisher.type”: This column may represent the type of publisher who published the book, such as a major publishing house or a smaller independent press. In the report I would it called publisher type.

“statistics.average rating”: This column represents the average rating of the book, as rated by customers. In the report I would it called statistics average rating.

“statistics.sale price”: This column represents the sale price of the book. In the report I would it called statistics sale price.

“statistics.sales rank”: This column represents the rank of the book based on sales performance. In the report I would it called statistics sales rank.

“statistics.total reviews”: This column represents the total number of reviews that the book has received. In the report I would it called statistics total reviews.

Methods

To answer the previous questions above, I used various statistical methods and tools such as descriptive statistics and Inferential statistics. Descriptive statistics are used to describe many characteristics of data. So, one can measure the central tendency of data by calculating the mean, mode, and median as well as measure the variability of data using mean deviation, standard deviation, and range. In addition, data can be presented in a form of plots such as histograms, bar plots, scatterplots, and others to visualize the collected data and we can also measure the distribution of data such as percentages. Whereas Inferential statistics involves drawing conclusions based on collected data. For instance, a hypothesis test is used to determine if the results of the data being studied are likely to occur by chance or if they represent a true effect or relationship between variables. Inferential statistics are important for predicting trends in data and helping identify likely outcomes (Bradley University Online, 2023). You can make predictions with inferential statistics using probability. You can also make inferences about large data based on small, collected data. For example, in this data set, we can determine the genre of books with the highest revenue daily and based on that we can conclude that it is the most profitable genre in the book industry.

Tools

```
import math
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from lmfit import minimize, Parameters, fit_report
import statistics as stats
publisher_data=pd.read_csv("publishers-corgis.csv")
```

I used Python programming language, specifically Python libraries such as Pandas, which is used to analyze, filter, and clean data if necessary, and NumPy, to perform statistical analysis on the data set such as calculating summary statics such as mean, median, and other necessary calculation such as standard deviation. Then I also used other Python libraries such as statistics which I used to determine the mode, math library which I used to perform certain mathematical operations, I also used SciPy to carry out the hypothesis test. We also used data visualization libraries such as Matplotlib and Seaborn to create charts and graphs to visualize the findings. To conduct the analysis, the data was first filtered and pre-processed to remove any missing values and ensure consistency in formatting. The cleaned data was then explored using descriptive statistics and visualizations such as histograms, and scatter plots to gain an understanding of the distribution and relationships between the variables. Fitting trend models were also used to investigate the relationships between variables by using Python libraries such as parameters, minimize, and fit_report which are in a package called "lmfit".

Data cleaning

There were no null values in the data but zero in the data set, which can be interpreted as missing values or no data record.

To check if there was a null value I used " isnull()" and "print()" functions and the result confirmed that there was indeed no null value.

The code to verify that there were no null values in the data set.

```
null_values = publisher_data.isnull().sum()
print(null_values)
```

The result :

```
genre                              0
sold by                            0
daily average.amazon revenue       0
daily average.author revenue       0
daily average.gross sales          0
daily average.publisher revenue    0
daily average.units sold           0
publisher.name                     0
publisher.type                     0
statistics.average rating          0
statistics.sale price              0
statistics.sales rank              0
statistics.total reviews           0
```

The code checks the number of zeros in the data set.

```
num_zeros = (publisher_data==0).sum().sum()
print(num_zeros)
```

```
number of zeros in data set:  8733
```

There were no null values. However, when I checked the number of zeros in the data sets there were a total of 8733. Now, the zeros were filtered out when working with the columns that have them such as statistic total reviews, statistics average rating, and daily average publisher revenue since it affects calculations such as median, mode, median, and other statistical operations and this leads to inaccurate results, but I did not filter the

So, I used the original data and the data without zeros depending on the columns that I worked on as there was no missing value in the data set.

Analysis section

Task1.Using Python code to determine which genre has the highest average "daily average.amazon revenue", the highest average "statistics.average rating", the highest average "statistics.total reviews", the highest average "daily average.gross sales", the highest average "statistics.sale price" and the highest average "daily average.units sold".

In determining which genre has the highest average daily Amazon revenue, the highest average statistics average rating, the highest average statistics total reviews, the highest average daily average gross sales, the highest average statistics sale price, and the highest average daily average units sold, I first filtered the data to have no zeros in the case of calculating summary statistics of columns of statistic total reviews and statistics average rating since they have zero values and it can skew the calculations and give inaccurate results. But I use the original data which I did not filter for the rest of the columns.

```
publisher_data1 = publisher_data[publisher_data != 0].dropna()
```

Then I used Python's " groupy()" function to group all the data such as sale price, reviews, and others based on genres, I used "agg" which stands for aggregate to group the data set by columns and NumPy Python to find the genre that on average has the highest daily average Amazon revenue, the statistics average rating, the highest statistics total reviews, the highest daily average gross sales, statistics sale price, and the highest daily average units sold by finding the mean of the

daily amazon revenue, median of the average rating, the mean of the total reviews, mean of the daily average gross sales, mean of the statistics sale price and mean of the daily average units sold.

The code below is used to determine the mean of the daily average Amazon revenue, the average rating, the total reviews, the daily average gross sales, the sale price, and the daily sold units by each genre.

```
# Calculate the daily average revenue for each genre
daily_avg_revenue = publisher_data.groupby('genre').agg({'daily average.amazon revenue': [np.mean, np.std, np.min, np.max]})
print("daily average.amazon revenue : ", daily_avg_revenue)
# Calculate the average rating for each genre
avg_rating = publisher_data1.groupby('genre').agg({'statistics.average rating': [np.median, np.std, np.min, np.max]})
print("avg_rating  : ", avg_rating)
# Calculate the total reviews for each genre
total_reviews = publisher_data1.groupby('genre').agg({'statistics.total reviews': [np.mean, np.std, np.min, np.max]})
print("total_reviews : ", total_reviews)
# Calculate the daily average gross sales for each genre
daily_avg_gross_sales = publisher_data.groupby('genre').agg({'daily average.gross sales': [np.mean, np.std, np.min, np.max]})
print("daily_avg_gross_sales  : ", daily_avg_gross_sales)
# Calculate the most sold units daily for each genre
most_sold_units_daily = publisher_data.groupby('genre').agg({'daily average.units sold': [np.mean, np.std, np.min, np.max]})
print("most_sold_units_daily: ", most_sold_units_daily)
# Calculate the average sale price for each genre
avg_statistics_sale_price = publisher_data.groupby('genre').agg({'statistics.sale price': [np.mean, np.std, np.min, np.max]})
print("avg_statistics_sale_price: ", avg_statistics_sale_price)
```

The result gives us the mean of the daily Amazon revenue, the average rating, the total reviews, the daily average gross sales, the sales price, and the daily units sold

```
daily average.amazon revenue :                  daily average.amazon revenue
                                        mean          std   amin       amax
genre
children                                9.894998   21.433274  0.198    375.564
comics                                  7.574145    7.950374  0.198     65.436
fiction                                63.208282  356.631216  0.570   8250.000
foreign language                        3.918045    4.906090  0.396     23.140
genre fiction                          76.986073  414.176940  0.198  12974.000
nonfiction                             18.699150   64.705730  0.004   5236.400
avg_rating  :                  statistics.average rating
                                    median       std  amin amax
genre
children                              4.610  0.353390  1.00  5.0
comics                                4.425  0.579644  1.00  5.0
fiction                               4.360  0.428930  2.33  5.0
foreign language                      4.555  0.485082  3.00  5.0
genre fiction                         4.350  0.436843  1.00  5.0
nonfiction                            4.440  0.424973  1.00  5.0
total_reviews :                   statistics.total reviews
                                       mean         std amin     amax
genre
children                           110.302927  284.755663  1.0   4698.0
comics                              48.053061   94.811627  1.0   1247.0
fiction                            332.735075  929.351401  1.0  13290.0
foreign language                    20.362069   32.620464  1.0    132.0
genre fiction                      232.500088  819.155646  1.0  23362.0
nonfiction                         125.235004  305.126281  1.0   6936.0
daily_avg_gross_sales  :                  daily average.gross sales
                                     mean          std  amin      amax
genre
children                           45.600240   105.788267  0.99   1877.82
comics                             37.684982    39.499220  0.99    327.18
fiction                           285.289468  1761.390288  2.85  41250.00
foreign language                   18.110165    24.836978  0.99    115.70
genre fiction                     279.896745  1351.963315  0.99  47795.00
nonfiction                         86.789653   317.052919  0.02  26182.00
```

```
most_sold_units_daily:                daily average.units sold
                                 mean          std amin  amax
genre
children                     9.115309    16.395832    1   263
comics                       6.040493     5.958124    1    82
fiction                     54.399727   255.108414    2  5500
foreign language             3.669421     3.573953    1    13
genre fiction               62.140065   251.078948    1  7000
nonfiction                  12.727491    57.949354    1  3800
avg_statistics_sale_price:                statistics.sale price
                                mean       std  amin    amax
genre
children                    4.687454  2.378540  0.90   15.12
comics                      6.416479  3.826307  0.99   19.98
fiction                     4.635880  3.630134  0.95   24.99
foreign language            4.991074  4.344664  0.99   29.79
genre fiction               4.628930  2.979824  0.81   50.31
nonfiction                  8.643909  7.870476  0.01  141.52
```

Result

The result indicates that the genre with the highest statistics sales price is nonfiction with a mean of 8.6 and the genre with the highest average daily Amazon revenue with a mean of 76.99. The genre with the highest statistics total reviews with a mean of 332.7 and the highest daily average gross sales with a mean of 285.29 is fiction. The highest daily units sold with a mean of 62.14 is genre fiction, and the genre with the highest statistics average rating with a mean of 4.5 is children.

Interpretation of the result

This means that on average in the day the genre of book that has the highest statistics total reviews, and highest average gross sales is fiction, and the genre with the highest Amazon revenue daily and is sold in a great number of units is genre fiction and the genre that tends to have high ratings is children. Therefore, the most profitable genre of the book with a large range of prices since it has a high standard deviation considering the average revenue made on Amazon, average gross sales, average reviews, and the number of units sold is fiction and genre fiction since genre fiction

has the highest revenue and fiction has the highest gross sale. the most highly rated genre of books is children, and the most priced genre of book is nonfiction. From the results, we can see that the most popular genre is fiction is among the genres with the least mean sale price, we can conclude that the sale price being low was one of the factors that made the genre fiction and fiction to it becoming a popular genre.

Task2.Using Python code to determine whether there is a strong relationship between publisher type and daily units sold.

To determine whether there is a strong relationship between the publisher type and daily units sold, I first created dummy variables for the publisher type column and then concatenate the dummy variables with the original data. After that, then I calculated the correlation between the dummy variables and the daily average units sold column, printed the results, and plotted a scatterplot between the dummy variables and the daily average units sold column.

The following is the code that determines whether there is a strong relationship between publisher type and daily units sold.

```
# create dummy variables for publisher type
publisher_dummies = pd.get_dummies(publisher_data['publisher.type'])
# concatenate the dummy variables with the original data
publisher_data_dummies = pd.concat([publisher_data, publisher_dummies], axis=1)
# calculate the correlation between the dummy variables and daily average unit sold
corr = publisher_data_dummies.iloc[:,7:].corrwith(publisher_data_dummies['daily average.units sold'])
print(corr)
```

The result gives a correlation between the dummy variables which includes the big five, single author, small/medium, amazon, indie and the daily average units sold column but also a correlation with other columns and daily average units sold.

```
amazon                    0.084727
big five                  0.051459
indie                     0.024488
single author            -0.025998
small/medium             -0.072222
```

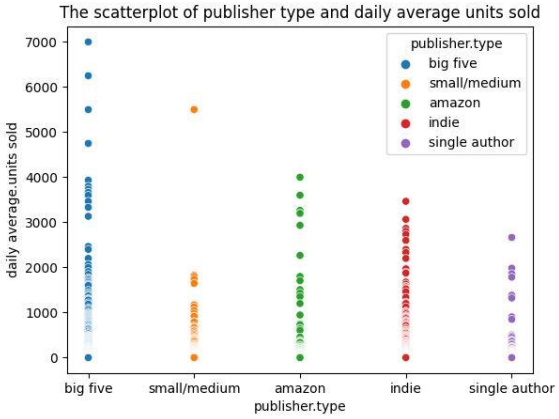
Interpretation of the result

from the result, there is a weak correlation between the dummy variables such as the big five, single author, small/medium, amazon, indie, and the daily average units sold with the correlation coefficient of 0.051,-0.025,-0.072,0.084 and 0.024 respectively. Therefore, there is a weak relationship between the publisher type and daily units sold since there is a weak correlation as 1 represents a positive strong correlation and -1 represents a negative strong correlation. In this case, the correlation is approximately 0 therefore there is a weak correlation. But this does not mean the publisher type does not influence the units sold daily but it means the degree of influence of publisher type towards the units sold in a day is small. But one of the objectives is to find whether the publisher affects the ratings, daily average gross sales, and reviews of the books depending on genres. It is plausible to believe that since there is a weak relationship between the publisher type and units sold in a day, there would also be a weak relationship between the publisher type and the daily average gross sales, but one must consider other factors may be influencing the relationship between these variables. Further analysis would be needed to confirm the strength and direction of the relationship between publisher type and daily average gross sales.

The code gives a scatterplot of the relationship between the publisher type and daily average units sold.

```
sns.scatterplot(data=publisher_data_dummies, x='publisher.type', y='daily average.units sold')
plt.show()
```

The scatterplot of dummy variables and daily average units sold.



Observation: From the scatter plot, the dots represent the daily average units sold depending on publisher types such as big five, small/medium, single author, amazon, or indie. There is a clearly weak relationship between the publisher type and daily average units sold as most of the dots do not follow certain trends such as increasing or decreasing as the daily average units sold goes up.

Task3.Using Python code to determine which is the most popular genre among the publishers included in the dataset.

to determine which is the most popular genre among the publishers included in the dataset. I use the “value_counts()” method on the 'genre' column of the dataset then print out the most popular genre which is the index of the first element of the resulting series.

The following is the code that determines which is the most popular genre among the publishers included in the dataset.

```
# count the number of books in each genre
most_popular_genre = publisher_data["genre"].value_counts().index[0]
# print the most popular genre
print('The most popular genre is:',most_popular_genre)
```

The result prints the most popular genre.

```
The most popular genre is: nonfiction
```

Interpretation of the result: The most popular genre among the publishers included in the dataset was Fiction.

Task4.Using Python code to determine whether the seller who sold the books or the type of publisher that published the books benefit more than the authors in terms of revenue.

To determine whether the seller who sold the books or the type of publisher that published the books benefit more than the authors in terms of revenue, I first calculated the total daily seller revenue and then I calculated the means of the seller revenue, publisher revenue, and author revenue to compare which group benefits more in terms of revenue using NumPy in python. Created a bar graph to visualize the comparison.

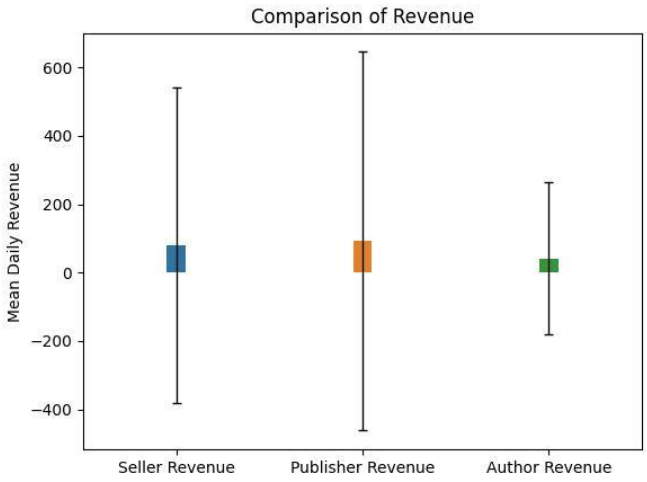
The code gives the means of the seller revenue, publisher revenue, and author revenue to compare which group benefits more in terms of revenue and creates a bar graph to visualize the comparison.

```
publisher_data['total_daily_seller_revenue'] =publisher_data['daily average.amazon revenue'] + publisher_data['daily average.author revenue']
mean_seller_revenue=np.mean(publisher_data['total_daily_seller_revenue'])
mean_publisher_revenue=np.mean(publisher_data1['daily average.publisher revenue'])
mean_author_revenue=np.mean(publisher_data['daily average.author revenue'] )
# print the results
print('Mean Seller Revenue:', mean_seller_revenue)
print('Mean Publisher Revenue:', mean_publisher_revenue)
print('Mean Author Revenue:', mean_author_revenue)
labels = ['Seller Revenue', 'Publisher Revenue', 'Author Revenue']
means = [mean_seller_revenue, mean_publisher_revenue, mean_author_revenue]
stds = [np.std(publisher_data['total_daily_seller_revenue']),
        np.std(publisher_data1['daily average.publisher revenue']),
        np.std(publisher_data['daily average.author revenue'])]

sns.barplot(x=labels, y=means, yerr=stds, width=0.1,capsize=2, error_kw={'elinewidth':1, 'capsize':3, 'capthick':1})
plt.title('Comparison of Revenue')
plt.ylabel('Mean Daily Revenue')
plt.show()
```

The results

```
Mean Seller Revenue: 79.44459052059051
Mean Publisher Revenue: 91.71916200607903
Mean Author Revenue: 41.465653772153765
```

The barplot of Sellers , publishers and authors revenue



From the code results and bar plot, we can see that both the means of the seller revenue daily and publisher revenue daily 79.4 and 91.72 respectively are greater than the mean of the daily author revenue. The confidence intervals are also big when it comes to sekker revenue and publisher revenue than the single author.

Interpretation of the result

Therefore based on the given dataset, this confirms that the seller who sold the books or the type of publisher that published the books benefit more than the authors in terms of revenue and have a wide of range if revenue. However, it is important to note that this analysis is based on a limited amount of data and only takes into account the variables provided in the data sets. It is possible that a more in-depth analysis could yield different results or uncover additional factors that affect revenue distribution.

Task5.Determining how the publisher type affects the sales, ranking, rating, and reviews of the books, how does the average sale price vary by publisher type?

To answer these questions, I used descriptive statistics and data visualization techniques on the dataset you have.

First, I calculated summary statistics such as the mean, median, and standard deviation for each variable such as sales, ranking, rating, reviews, and sale price for each publisher type. This will give you an idea of how the different publisher types are performing in terms of these variables.

Then I created visualizations of data through plots such as box plots to see the distribution and relationship between the variables and publisher types. For example, we can create a box plot to compare the distribution of sales for the big five publishers versus the small/medium publisher.

The code determines how the publisher type affects the gross sales, ranking, rating, and reviews of the books, and how the average sale price varies by publisher type.

```
# group the data by publisher type
publisher_groups = publisher_data.groupby("publisher.type")
publisher_groups1 = publisher_data1.groupby("publisher.type")
# calculate mean and std for each variable by publisher type
sales_mean = publisher_groups["daily average.gross sales"].aggregate([np.mean, np.std])
rating_med = publisher_groups1["statistics.average rating"].aggregate(np.median)
price_mean = publisher_groups["statistics.sale price"].aggregate([np.mean, np.std])
rank_med = publisher_groups["statistics.sales rank"].aggregate(np.median)
review_mean = publisher_groups1["statistics.total reviews"].aggregate([np.mean, np.std])
# print mean and std for each variable by publisher type
print("Sales Mean by Publisher Type:\n", sales_mean)
print()
print("\nRating Median by Publisher Type:\n", rating_med)
print()
print("\nPrice Mean by Publisher Type:\n", price_mean)
print()
print("\nRank Median by Publisher Type:\n", rank_med)
print()
print("\nReview Mean by Publisher Type:\n", review_mean)
print()
```

result :

```
Gross Sales Mean by Publisher Type:
                      mean          std
publisher.type
amazon          594.031584  2180.602227
big five        297.250296  1389.974599
indie           103.310521   383.096353
single author    60.944138   169.294401
small/medium     83.306048   538.416683


Rating Median by Publisher Type:
 publisher.type
big five         4.39
single author    4.48
small/medium     4.45
Name: statistics.average rating, dtype: float64


Sale Price Mean by Publisher Type:
                     mean       std
publisher.type
amazon           4.010804  1.305711
big five         8.388245  3.664164
indie            3.428123  2.626634
single author    4.768780  4.416391
small/medium     8.477521  8.800135


Rank Median by Publisher Type:
 publisher.type
amazon           6274.0
big five        25838.0
indie           23312.5
single author   32889.5
small/medium    44570.0
Name: statistics.sales rank, dtype: float64
```

```
Review Mean by Publisher Type:
                        mean         std
publisher.type
big five          295.458733  735.915682
single author      65.498985  155.171449
small/medium       88.571167  393.626032
```

The result: from the results of the code, amazon has the highest mean and standard deviation across the gross sales of the books, and the big five has the highest mean and standard deviation for total reviews and highest median for the average rating while small/median has highest median for rank and highest mean for sale price.

Interpretation of the result

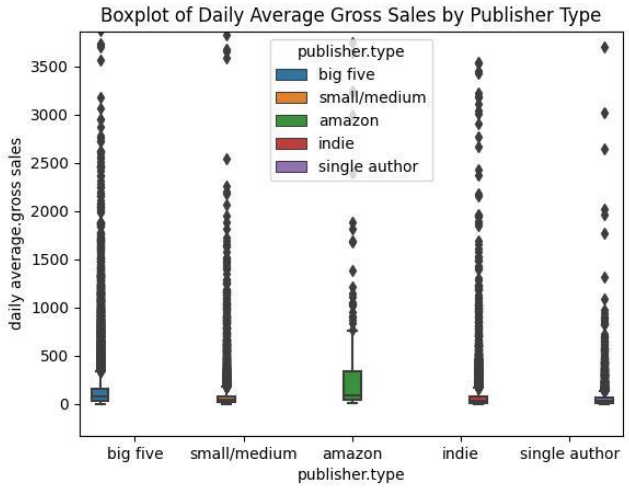
From the result, we can confirm that the publisher type does affect the affects gross sales, ranking, rating, and reviews of the books, and the average sale price increases from indie, amazon, single author, big five to small/medium. From the result, we can confirm that big publishing types such as Amazon and the big five have the biggest gross sales due to the ability to be able to sell large numbers of units in a day.

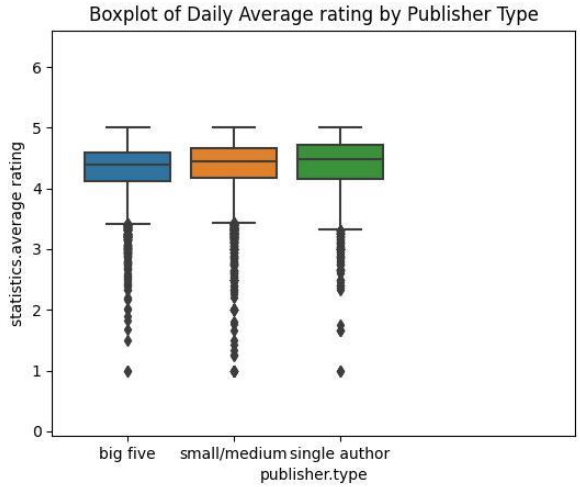
To visualize the effect of publisher type on sales, ranking, rating, and reviews of the books, and the variation of the average sale price by publisher type, we can use data visualization tools such as box plots.

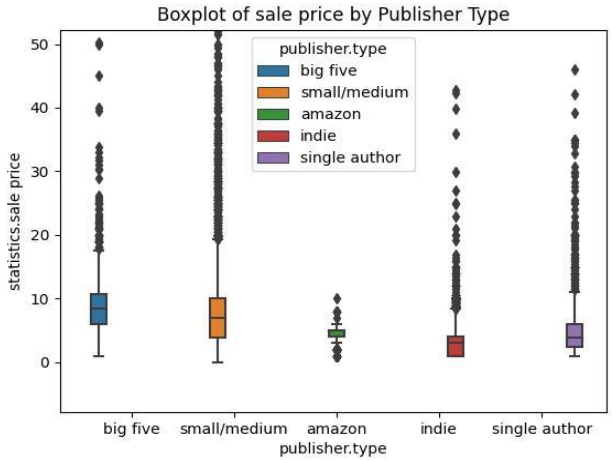
The code creates box plots for the variables of interest grouped by publisher type and compares the big five publishers versus the small/medium publisher.

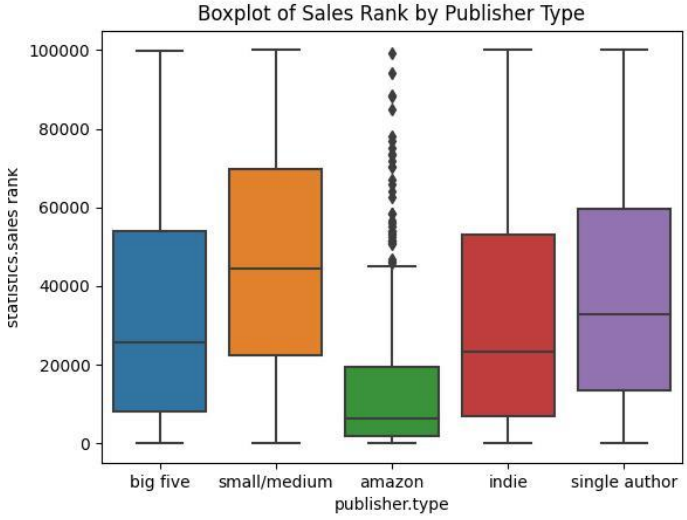
```
sns.boxplot(data=publisher_data, x="publisher.type", y="daily average.gross sales", hue="publisher.type")
plt.ylim(-200,4000)
plt.title("Boxplot of Daily Average Gross Sales by Publisher Type")
plt.show()
sns.boxplot(data=publisher_data1, x="publisher.type", y="statistics.average rating")
plt.ylim(0, 20)
plt.title("Boxplot of Daily Average rating by Publisher Type")
plt.show()
sns.boxplot(data=publisher_data, x="publisher.type", y="statistics.sale price", hue="publisher.type")
plt.ylim(-1, 99)
plt.title("Boxplot of sale price by Publisher Type")
plt.show()
sns.boxplot(data=publisher_data, x="publisher.type", y="statistics.sales rank")
plt.title("Boxplot of Sales Rank by Publisher Type")
plt.show()
sns.boxplot(data=publisher_data1, x="publisher.type", y="statistics.total reviews", hue="publisher.type")
plt.ylim(-1, 2000)
plt.title("Boxplot of total reviews by Publisher Type")
plt.show()
```

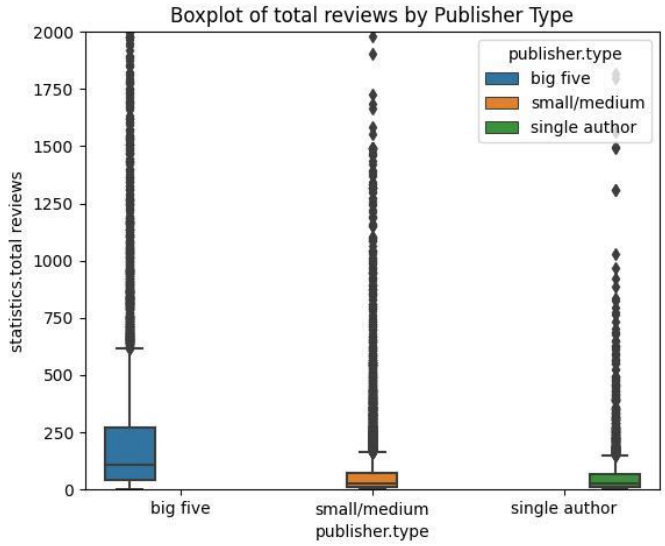
The result:




Boxplot of Daily Average rating by Publisher Type
statistics.average rating
0
1
2
3
4
5
6
big five
small/medium
single author
publisher.type


Boxplot of sale price by Publisher Type
statistics.sale price
0
10
20
30
40
50
publisher.type
big five
small/medium
amazon
indie
single author
big five
small/medium
amazon
indie
single author
publisher.type


Boxplot of Sales Rank by Publisher Type
statistics.sales rank
0
20000
40000
60000
80000
100000
big five
small/medium
amazon
indie
single author
publisher.type



Observation: we can compare the distribution of gross Sales, Rating, Sale price, Rank, and review for the big five publishers versus the small/medium publisher to verify if it indeed publisher type does affect the variables such as gross sales and others.

Gross Sales: The box plots show that books published by the "big five" publishers tend to have higher sales than those published by small or medium publishers, as indicated by the higher median values and wider interquartile ranges. However, there is also a wider spread of data for the big five publishers and indie, suggesting greater variability in sales for these books.

Rating: The box plots suggest that books published by small or medium publishers tend to have slightly higher average ratings than those published by the big five. The interquartile ranges are relatively similar for both groups, indicating similar levels of variability in ratings.

Sale price: The box plots indicate that books published by small or medium publishers tend to have lower average sale prices than those published by the big five, as indicated by the lower median values and narrower interquartile ranges. However, there is also greater variability in the price for small or medium publishers, suggesting a wider range of prices for their books.

Rank: The box plots suggest that books published by small or medium publishers tend to have slightly higher sales ranks than those published by the big five. The interquartile ranges are relatively similar for both groups, indicating similar levels of variability in sales ranks.

Reviews: The box plots suggest that books published by small or medium publishers tend to have slightly lower total reviews than those published by the big five. The interquartile ranges are relatively different for publisher types, indicating different levels of variability in total reviews.

Overall, it appears that publisher type does affect sales, rating, price, rank, and reviews, but its magnitude varies.

Task6.Determining which publisher has the highest daily average revenue and units sold.

To find the publisher with the highest daily average revenue and units sold, I grouped the data by publisher name using groupy() function, calculate the mean of the “daily average.publisher revenue" and "daily average. units sold" variables for each publisher using aggregate function in python , and then use idmax()to identify the publisher with the highest values. After that, I plotted a bar graph to visualize the results.

The code Determines which publisher has the highest daily average revenue and units sold.

```
revenue_mean = publisher_data.groupby(['publisher.name', 'genre'])['daily average.publisher revenue'].mean().reset_index()
units_mean = publisher_data.groupby(['publisher.name', 'genre'])['daily average.units sold'].mean().reset_index()
highest_revenue_publisher = revenue_mean.loc[revenue_mean['daily average.publisher revenue'].idxmax()]
highest_units_sold_publisher = units_mean.loc[units_mean['daily average.units sold'].idxmax()]
print("Publisher with the highest daily average revenue and units sold:")
print(highest_revenue_publisher['publisher.name'], "(", highest_revenue_publisher['genre'], ")")
print("Daily average revenue for this publisher:", highest_revenue_publisher['daily average.publisher revenue'])
print("Daily average units sold for this publisher:", highest_units_sold_publisher['daily average.units sold'])
```

The printed result

```
Publisher with the highest daily average revenue and units sold:
Dutton Children's ( genre fiction )
Daily average revenue for this publisher: 22771.5
Daily average units sold for this publisher: 4750.0
```

The result:

The publisher with the highest daily average revenue and daily average unit sold is Dutton Children's with a value of 4704 and 979.6 respectively and the genre publisher Dutton Children's published mostly genre fiction.

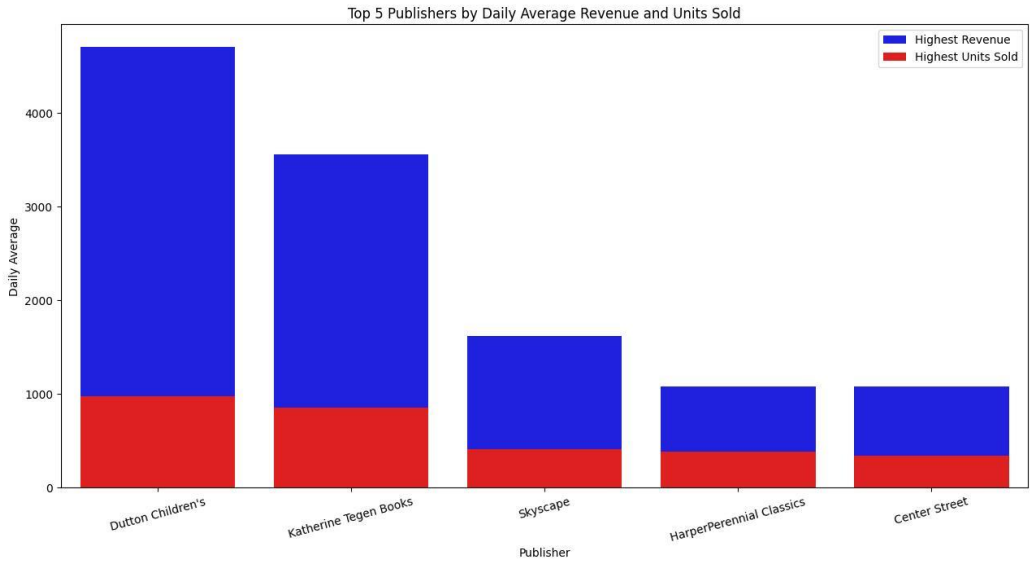
Interpretation of the result

The publisher's revenue is affected by the genre since the genre affects the daily average revenue and average units of books sold.

The code generates a bar plot that gives the visualization of the top 5 publishers in the book industry according to the data set.

```
N=5
top_revenue_publishers = revenue_mean.sort_values(ascending=False)[:N]
top_units_sold_publishers = units_mean.sort_values(ascending=False)[:N]
sns.barplot(x=top_revenue_publishers.index, y=top_revenue_publishers.values, color='blue', label='Highest Revenue')
sns.barplot(x=top_units_sold_publishers.index, y=top_units_sold_publishers.values,  color='red',label='Highest Units Sold')
plt.xlabel('Publisher')
plt.ylabel('Daily Average')
plt.title(f'Top {N} Publishers by Daily Average Revenue and Units Sold')
plt.xticks(rotation=15)
plt.legend()
plt.show()
```

The result



Observation

The graph represents the daily average of revenue and units sold by the top 5 publishers which most of which appears to publish either fiction, genre fiction, or children.

Task 7: Examine the relationship between average rating and sales rank and use linear fitting trends to model the relationship to give visualization.

I first define a function to model the relationship between sales rank and an average rating and define a function to calculate the error between the observed and predicted values. extract the data for sales rank and average rating. Set the initial values for the parameters to be estimated and create the Parameter object. minimize the error function and obtain the best-fit parameters. Calculate the predicted values using the best-fit parameters.

The code

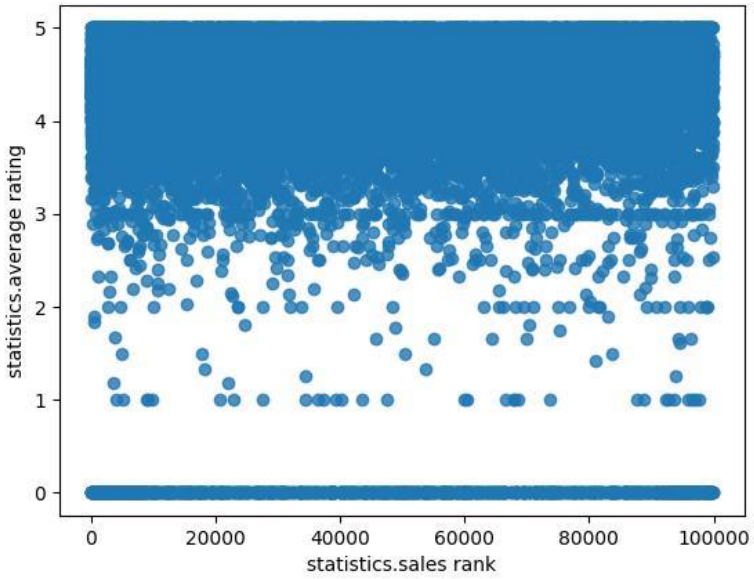
```
def f(x, params):
    a1 = params['a1'].value
    a0 = params['a0'].value
    R = a1*x + a0
    return R

# Define the error function to be minimized
def E(params, x, R):
    f_vals = f(x, params)
    error = R - f_vals
    return error
# Set up the data
D = publisher_data["statistics.sales rank"]
R = publisher_data["statistics.average rating"]
params = Parameters()
params.add('a1', value=1)
params.add('a0', value=5, vary=False)
# Minimize the error function and report the fit results
result = minimize(E, params, args=(D, R))
# Calculate the fitted values and plot the data with the regression line
R_fit=f(D,result.params)
sns.regplot(x=D,y=R)
plt.show()
# Calculate and print the correlation coefficient
correlation_coefficient = np.corrcoef(D, R)[0][1]
print("Correlation coefficient:", correlation_coefficient)
```

results

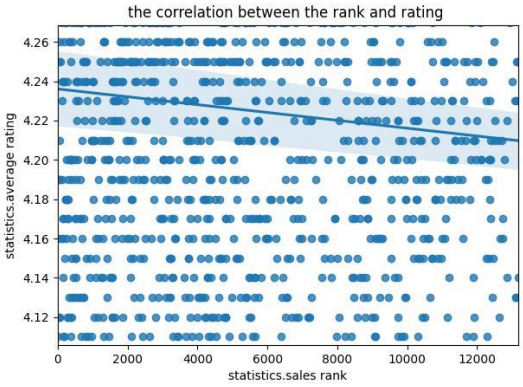
```
Correlation coefficient: -0.05827402412505519
```

Interpretation of the result : There is a weak relationship between the rank and the ratings.

The graph



Observation: the graph shows that the correlation between the rating and rank is so low that the line of best fit is so small to be viewed. But there is zoomed version of the graph.



This indicated there is a negative weak relationship between the rank and the rating.

Task 8: determining the most popular publisher type

To check the most popular publisher type, I decided to calculate the mode of the publisher type column using the stats library in Python.

```
publishertype_mode=stats.mode(publisher_data["publisher.type"])
print(publishertype_mode)
```

Result: most popular publisher type is small/medium.

Interpretation: this makes sense since among the population of the publishers the majority are small or medium publishers.

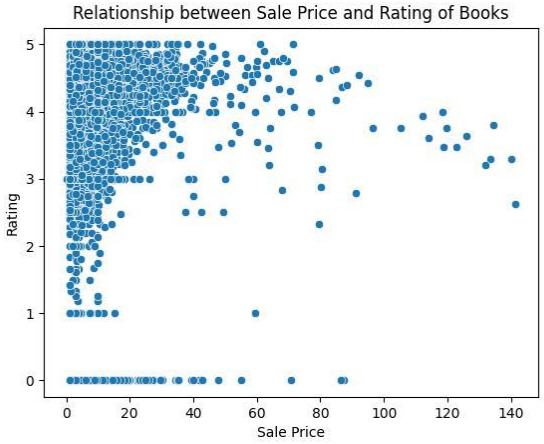
Task 9: investigates the relationships between the Sale price and rating of books.

To investigate the relationships between the price and rating of books and check for outliers, we can use scatter plots and box plots.

Here's the code to create a scatter plot of price vs. rating.

```
sns.scatterplot(x="statistics.sale price", y="statistics.average rating", data=publisher_data)
plt.title("Relationship between Sale Price and Rating of Books")
plt.xlabel("Sale Price")
plt.ylabel("Rating")
plt.show()
```

The results



Interpretation: From the scatterplot, there seems to be a weak positive relationship between the sale price and Ratings since most of the data points are concentrated at the sale price of 0 and 40 there seems to be a positive trend which is weak between the sale price and ratings of books. This confirms that the sale price does not affect the ratings of the books in a great magnitude.

I confirmed that there is indeed a weak correlation between the sale price and ratings by using hypothesis testing.

To verify the relationship between the rating and price, I used a correlation test, such as the Pearson correlation test which will calculate the correlation coefficient and the associated p-value.

The code

```
from scipy.stats import pearsonr, ttest_ind
correlation, p_value = pearsonr(publisher_data["statistics.sale price"], publisher_data['statistics.average rating'])
print("Pearson correlation coefficient:", correlation)
print("p-value:", p_value)
# Perform a hypothesis test
alpha = 0.05
if p_value < alpha:
    print("Reject null hypothesis: There is a strong correlation between price and rating")
else:
    print("Fail to reject null hypothesis: There is not enough evidence to support a strong correlation between price and rating")
```

The result

```
Pearson correlation coefficient: 0.0006070820214221244
p-value: 0.9205038917670825
Fail to reject null hypothesis: There is not enough evidence to support a strong correlation between price and rating
```

Interpretation of the result: This confirms there is a weak correlation between the sale price and average rating.

Predictions:

Based on this analysis, we predict that fiction and genre fiction books will continue to be the most profitable genres since they have greater average revenue than the rest of the genres. Therefore, the publishers of those books would also be more popular. Books published by small or medium publishers receive better rankings than the big five or Amazon. Due to their high sale prices, which consumers are unlikely to be able to afford, books with high ratings such as children's will generate less revenue for publishers in the future. For instance, nonfiction is the most popular genre in the publishing industry. If it continues to have high sales prices its popularity would decline.

Conclusion

In conclusion, the sale price of the books does affect the ratings of the books and neither does rank affect it also. However, the sale price does affect the variables such as the daily average revenue and reviews. Publishing several books depends on the type of publisher. The popularity of the publisher depends on the genre of books published. We can see that book authors receive less revenue than publishers and sellers. However, that does not mean it is the only money they receive from writing them. Overall, sale prices, genres, and publisher type play a significant role in the publishing industry. One must consider these factors when engaging in the publishing industry.

Reference:

Descriptive statistics vs. inferential statistics. Bradley University Online. (2022, March 9). Retrieved January 28, 2023, from https://onlinedegrees.bradley.edu/blog/whats-the-difference-between-descriptive-and-inferential-statistics/