**Amazon Bestselling Books Analysis with Python**

* [Aman KharwalAman Kharwal](https://thecleverprogrammer.com/author/amankharwal/)
* November 30, 2020
* [Machine Learning](https://thecleverprogrammer.com/category/by-aman-kharwal/)

In this article, I’m going to introduce you to a data science project on Amazon bestselling books analysis with the Python programming language. The data I’ll be using in this data science project is a dataset of Amazon’s 50 Best Books between 2009 and 2019.

The dataset contains 550 books and has been categorized into fiction and non-fiction using Goodreads.

**Data Science Project on Amazon Bestselling Books Analysis with Python**

I will start the task of amazon bestselling books analysis with Python by importing the necessary Python libraries and the dataset:

|  |
| --- |
| import pandas as pd # dataframe manipulation |
|  | import numpy as np # linear algebra |
|  |  |
|  | # data visualization |
|  | import matplotlib.pyplot as plt |
|  | %matplotlib inline |
|  | import seaborn as sns |
|  | print('Seaborn verion', sns.\_\_version\_\_) |
|  | sns.set\_style('whitegrid') |
|  |  |
|  | # text data |
|  | import string |
|  | import re |
|  | df = pd.read\_csv('bestsellers with categories.csv') |

**Data Preparation:**

Now the next step is to prepare the data, here I will rename User Rating as user\_rating, and then we will fix some spellings in the data:

|  |  |
| --- | --- |
|  | df.rename(columns={"User Rating": "User\_Rating"}, inplace=True) |
|  | df[df.Author == 'J. K. Rowling'] |
|  | df[df.Author == 'J.K. Rowling'] |
|  | df.loc[df.Author == 'J. K. Rowling', 'Author'] = 'J.K. Rowling' |
|  | df['name\_len'] = df['Name'].apply(lambda x: len(x) - x.count(" ")) # subtract whitespaces |
|  | punctuations = string.punctuation |
|  | print('list of punctuations : ', punctuations) |
|  |  |
|  | # percentage of punctuations |
|  | def count\_punc(text): |
|  | """This function counts the number of punctuations in a text""" |
|  | count = sum(1 for char in text if char in punctuations) |
|  | return round(count/(len(text) - text.count(" "))\*100, 3) |
|  |  |
|  | # apply function |
|  | df['punc%'] = df['Name'].apply(lambda x: count\_punc(x)) |

[view raw](https://gist.github.com/amankharwal/b8be6517a9be30ccbe17f1a2003ed3b5/raw/e41c95cbd2a841ea516e83317bd1fd89e2046c3a/amazon.py)[amazon.py](https://gist.github.com/amankharwal/b8be6517a9be30ccbe17f1a2003ed3b5#file-amazon-py)hosted with ❤ by [GitHub](https://github.com/)

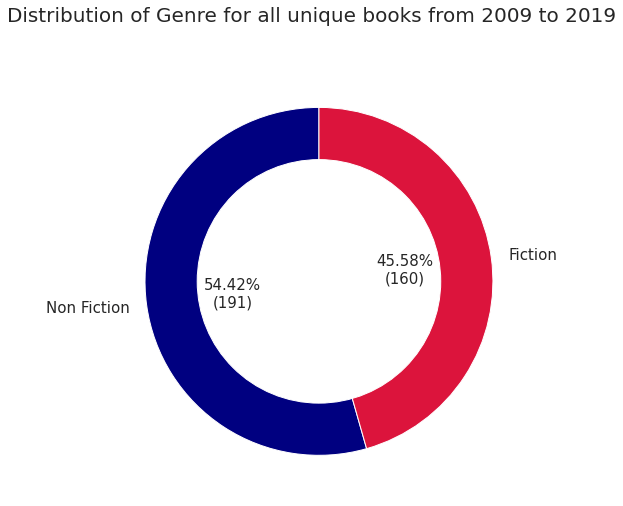
**Amazon Bestselling Books Analysis with Python**

In the data set, Genre is a categorical dummy variable; Fiction and non-fiction. Non-fiction was a more popular category than fiction, each year from 2009 to 2019. Of the 351 unique books, 54.4% were non-fiction and 45.6% were fiction.

The highest fraction (66%) of non-fiction books were sold in 2015 and the lowest for fiction books. For fiction books, the highest fraction (48%) of books were sold in 2009, 2013 and 2017, and the lowest for non-fiction books. Let’s visualize the data according to the genre:

|  |  |
| --- | --- |
|  | no\_dup = df.drop\_duplicates('Name') |
|  | g\_count = no\_dup['Genre'].value\_counts() |
|  |  |
|  | fig, ax = plt.subplots(figsize=(8, 8)) |
|  |  |
|  | def make\_autopct(values): |
|  | def my\_autopct(pct): |
|  | total = sum(values) |
|  | val = int(round(pct\*total/100.0)) |
|  | return '{p:.2f}%\n({v:d})'.format(p=pct,v=val) |
|  | return my\_autopct |
|  |  |
|  | genre\_col = ['navy','crimson'] |
|  | #genre\_col = ['khaki','plum'] |
|  |  |
|  | center\_circle = plt.Circle((0, 0), 0.7, color='white') |
|  | plt.pie(x=g\_count.values, labels=g\_count.index, autopct=make\_autopct(g\_count.values), |
|  | startangle=90, textprops={'size': 15}, pctdistance=0.5, colors=genre\_col) |
|  | ax.add\_artist(center\_circle) |
|  |  |
|  | fig.suptitle('Distribution of Genre for all unique books from 2009 to 2019', fontsize=20) |
|  | fig.show() |

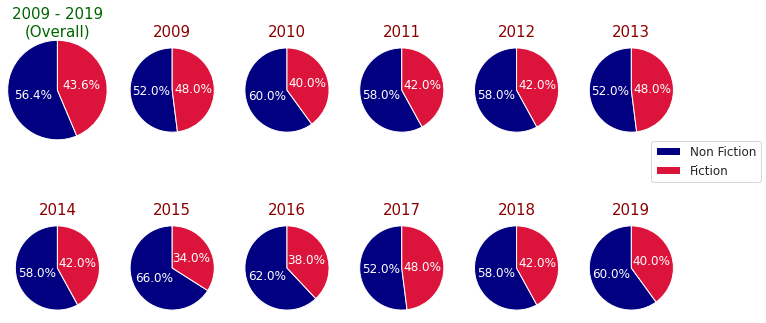
[view raw](https://gist.github.com/amankharwal/39e8303cfccc239a672dbae6a318cd56/raw/a368f51c378ef60d8c967c5ab326bbbb4b3520af/amazon.py)[amazon.py](https://gist.github.com/amankharwal/39e8303cfccc239a672dbae6a318cd56#file-amazon-py)hosted with ❤ by [GitHub](https://github.com/)



Now let’s visualize the above insights according to each year:

|  |
| --- |
| y1 = np.arange(2009, 2014) |
|  | y2 = np.arange(2014, 2020) |
|  | g\_count = df['Genre'].value\_counts() |
|  |  |
|  | fig, ax = plt.subplots(2, 6, figsize=(12,6)) |
|  |  |
|  | ax[0,0].pie(x=g\_count.values, labels=None, autopct='%1.1f%%', |
|  | startangle=90, textprops={'size': 12, 'color': 'white'}, |
|  | pctdistance=0.5, radius=1.3, colors=genre\_col) |
|  | ax[0,0].set\_title('2009 - 2019\n(Overall)', color='darkgreen', fontdict={'fontsize': 15}) |
|  |  |
|  | for i, year in enumerate(y1): |
|  | counts = df[df['Year'] == year]['Genre'].value\_counts() |
|  | ax[0,i+1].set\_title(year, color='darkred', fontdict={'fontsize': 15}) |
|  | ax[0,i+1].pie(x=counts.values, labels=None, autopct='%1.1f%%', |
|  | startangle=90, textprops={'size': 12,'color': 'white'}, |
|  | pctdistance=0.5, colors=genre\_col, radius=1.1) |
|  |  |
|  | for i, year in enumerate(y2): |
|  | counts = df[df['Year'] == year]['Genre'].value\_counts() |
|  | ax[1,i].pie(x=counts.values, labels=None, autopct='%1.1f%%', |
|  | startangle=90, textprops={'size': 12,'color': 'white'}, |
|  | pctdistance=0.5, colors=genre\_col, radius=1.1) |
|  | ax[1,i].set\_title(year, color='darkred', fontdict={'fontsize': 15}) |
|  |  |
|  | #plt.suptitle('Distribution of Fiction and Non-Fiction books for every year from 2009 to 2019', |
|  | #fontsize=25) |
|  | fig.legend(g\_count.index, loc='center right', fontsize=12) |
|  | fig.show() |

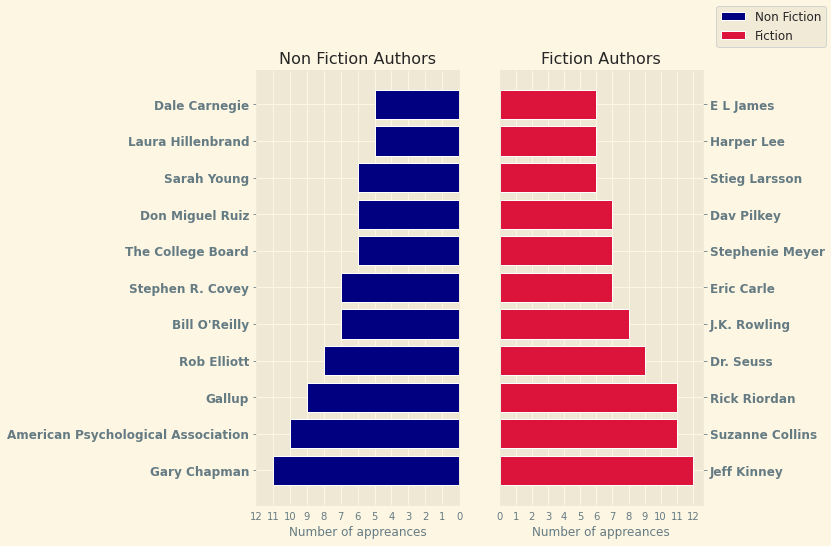
[view raw](https://gist.github.com/amankharwal/0dd2773d3c7c0df3939ff928606fbe7c/raw/4203d5906774de07691ac012a769f43319620035/amazon.py)[amazon.py](https://gist.github.com/amankharwal/0dd2773d3c7c0df3939ff928606fbe7c#file-amazon-py)hosted with ❤ by [GitHub](https://github.com/)



The bestselling authors are selected based on their appearances in the top 50 bestselling books each year, from 2009 to 2019. Now let’s look at the top 10 bestselling authors of both fiction and non-fiction categories:

|  |  |
| --- | --- |
|  | best\_nf\_authors = df.groupby(['Author', 'Genre']).agg({'Name': 'count'}).unstack()['Name', 'Non Fiction'].sort\_values(ascending=False)[:11] |
|  | best\_f\_authors = df.groupby(['Author', 'Genre']).agg({'Name': 'count'}).unstack()['Name', 'Fiction'].sort\_values(ascending=False)[:11] |
|  |  |
|  | with plt.style.context('Solarize\_Light2'): |
|  | fig, ax = plt.subplots(1, 2, figsize=(8,8)) |
|  |  |
|  | ax[0].barh(y=best\_nf\_authors.index, width=best\_nf\_authors.values, |
|  | color=genre\_col[0]) |
|  | ax[0].invert\_xaxis() |
|  | ax[0].yaxis.tick\_left() |
|  | ax[0].set\_xticks(np.arange(max(best\_f\_authors.values)+1)) |
|  | ax[0].set\_yticklabels(best\_nf\_authors.index, fontsize=12, fontweight='semibold') |
|  | ax[0].set\_xlabel('Number of appreances') |
|  | ax[0].set\_title('Non Fiction Authors') |
|  |  |
|  | ax[1].barh(y=best\_f\_authors.index, width=best\_f\_authors.values, |
|  | color=genre\_col[1]) |
|  | ax[1].yaxis.tick\_right() |
|  | ax[1].set\_xticks(np.arange(max(best\_f\_authors.values)+1)) |
|  | ax[1].set\_yticklabels(best\_f\_authors.index, fontsize=12, fontweight='semibold') |
|  | ax[1].set\_title('Fiction Authors') |
|  | ax[1].set\_xlabel('Number of appreances') |
|  |  |
|  | fig.legend(['Non Fiction', 'Fiction'], fontsize=12) |
|  |  |
|  | plt.show() |

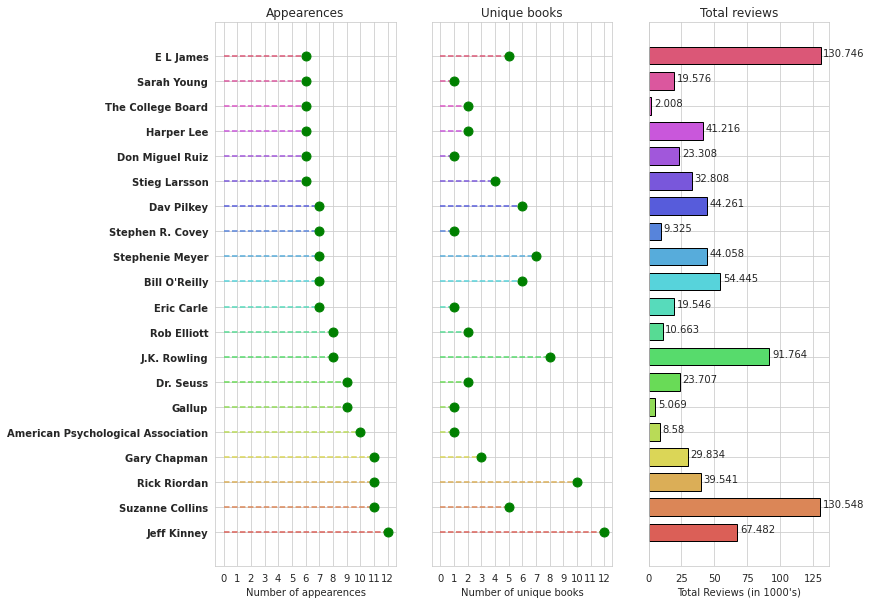
[view raw](https://gist.github.com/amankharwal/ac398aa519cc581f08242439099482d7/raw/ed53c93a76c719b82ee308998a7e98fa91dfb2d7/amazon.py)[amazon.py](https://gist.github.com/amankharwal/ac398aa519cc581f08242439099482d7#file-amazon-py)hosted with ❤ by [GitHub](https://github.com/)



Top-selling authors are selected based on their appearances in the top 50 best-selling books each year. The number of appearances includes duplicate book names. Their unique posts and overall reviews are featured below:

|  |  |
| --- | --- |
|  | n\_best = 20 |
|  |  |
|  | top\_authors = df.Author.value\_counts().nlargest(n\_best) |
|  | no\_dup = df.drop\_duplicates('Name') # removes all rows with duplicate book names |
|  |  |
|  | fig, ax = plt.subplots(1, 3, figsize=(11,10), sharey=True) |
|  |  |
|  | color = sns.color\_palette("hls", n\_best) |
|  |  |
|  | ax[0].hlines(y=top\_authors.index , xmin=0, xmax=top\_authors.values, color=color, linestyles='dashed') |
|  | ax[0].plot(top\_authors.values, top\_authors.index, 'go', markersize=9) |
|  | ax[0].set\_xlabel('Number of appearences') |
|  | ax[0].set\_xticks(np.arange(top\_authors.values.max()+1)) |
|  | ax[0].set\_yticklabels(top\_authors.index, fontweight='semibold') |
|  | ax[0].set\_title('Appearences') |
|  |  |
|  | book\_count = [] |
|  | total\_reviews = [] |
|  | for name, col in zip(top\_authors.index, color): |
|  | book\_count.append(len(no\_dup[no\_dup.Author == name]['Name'])) |
|  | total\_reviews.append(no\_dup[no\_dup.Author == name]['Reviews'].sum()/1000) |
|  | ax[1].hlines(y=top\_authors.index , xmin=0, xmax=book\_count, color=color, linestyles='dashed') |
|  | ax[1].plot(book\_count, top\_authors.index, 'go', markersize=9) |
|  | ax[1].set\_xlabel('Number of unique books') |
|  | ax[1].set\_xticks(np.arange(max(book\_count)+1)) |
|  | ax[1].set\_title('Unique books') |
|  |  |
|  | ax[2].barh(y=top\_authors.index, width=total\_reviews, color=color, edgecolor='black', height=0.7) |
|  | for name, val in zip(top\_authors.index, total\_reviews): |
|  | ax[2].text(val+2, name, val) |
|  | ax[2].set\_xlabel("Total Reviews (in 1000's)") |
|  | ax[2].set\_title('Total reviews') |
|  |  |
|  | #plt.suptitle('Top 20 best selling Authors (from 2009 to 2019) details', fontsize=15) |
|  | plt.show() |

[view raw](https://gist.github.com/amankharwal/65a5c286d4eb652f93a076e0ea89c1c3/raw/d0605837dab6ee297e8790141b150b9f8620c74a/amazon.py)[amazon.py](https://gist.github.com/amankharwal/65a5c286d4eb652f93a076e0ea89c1c3#file-amazon-py)hosted with ❤ by [GitHub](https://github.com/)



Author Jeff Kinney is the best-selling author with 12 appearances in best-selling books from 2009 to 2019. This is how we can analyze any data related to sales consisting of different categories.

I hope you liked this article on the Data Science project on Amazon Bestselling books analysis with Python programming language. Feel free to ask your valuable questions in the comments section below.