Final Project

The purpose of this assignment is to find a topic that is interesting to you. Then present an interesting aspect of the data using descriptive text and data visualizations.

Author: Tyler Laudenslager

CSC 223 - Advanced Scientific Programming

Fall 2020

Introduction:

I am always trying to find new books to read and I tend to search for lists of the best books of all time on the internet for a guide on what to read next. Naturally the first dataset I chose for this assignment is amazon's top 50 bestselling books from 2009-2019. This dataset includes the name, author, average user rating, number of reviews, price, year and genre of each bestselling book. The second dataset includes all the books the goodreads community voted on to be the best books ever. This dataset has alot of information about each book notably name, authors, average rating, number of pages, and number of ratings. Goodreads is a website that helps book readers discover new books that they might be interested in based on the previous books they enjoyed.

Amazon Best Seller List Dataset - https://www.kaggle.com/sootersaalu/amazon-top-50-bestselling-books-2009-2019

Goodreads Best Books Dataset - https://www.kaggle.com/meetnaren/goodreads-best-books

Goodreads Best Books Ever - https://www.goodreads.com/list/show/1.Best_Books_Ever

Kaggle - https://www.kaggle.com - is a online respository of public datasets. I found out about kaggle from searching for datasets using Google's dataset search engine.

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:

```
amzn_books_df = pd.read_csv('amazon_best_sellers.csv')
good_reads_books_df = pd.read_csv('good_reads_book_data.csv')
```

In [3]:

```
good_reads_books_df.rename(columns={'book_title':'Name','book_rating':'good_reads_ratin
good_reads_books_df['good_reads_rating'] = [x.round(1) for x in good_reads_books_df['go
amzn_books_df.rename(columns={'User Rating':'amazon_rating','Author':'author'},inplace=
```

The normal distribution function below was written by Dr.Schwesinger

In [4]:

```
def normal_distribution(x, mu, sigma):
    return 1/(sigma * np.sqrt(2*np.pi)) * np.exp(-(x-mu)**2/(2*sigma**2))
```

In [5]:

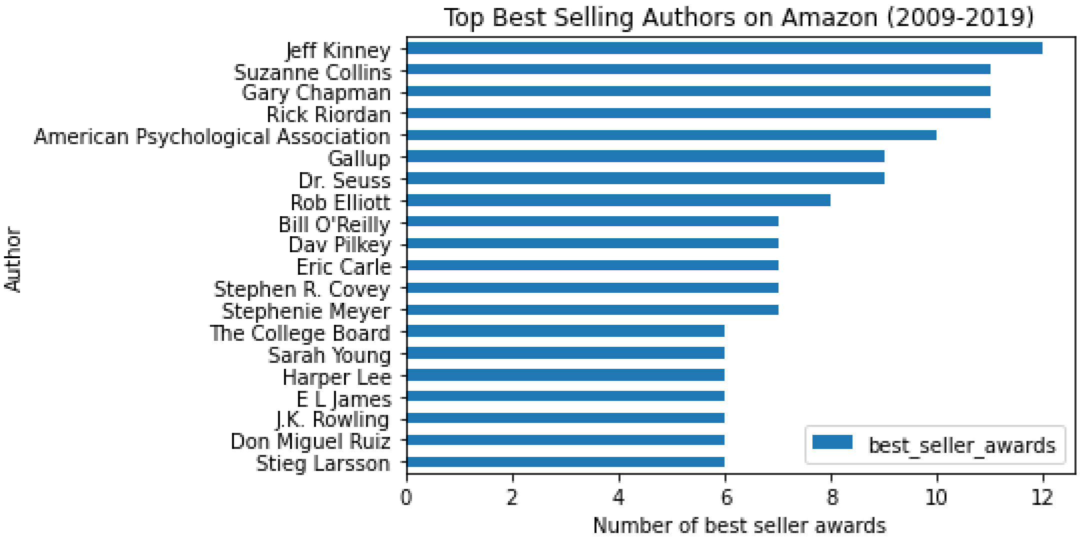
```
#counts all the times an author got bestseller for all years
#returns dictionary of author names as keys and frequency as values.
def count_best_sellers(data_frame):
    author_bestseller_count = dict()
    for author_name in data_frame['author']:
        if author_name in author_bestseller_count.keys():
            author_bestseller_count[author_name] += 1
        else:
            author_bestseller_count[author_name] = 1
    return author_bestseller_count


author_dict = count_best_sellers(amzn_books_df)
```

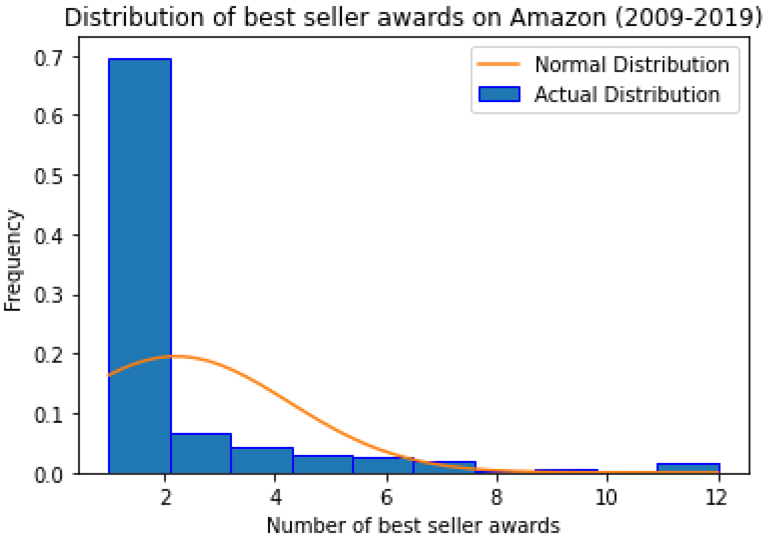
In [6]:

```
author_dict_df = pd.DataFrame({'Author':author_dict.keys(),'best_seller_awards':author_
top_author = author_dict_df.loc[author_dict_df['best_seller_awards'] >= 6].sort_values(
top_author.set_title('Top Best Selling Authors on Amazon (2009-2019)')
top_author.set_xlabel('Number of best seller awards')
author_best_sellers = author_dict_df.plot(kind='hist',density=True,edgecolor='b')
x_values = np.linspace(author_dict_df['best_seller_awards'].min(), author_dict_df['best
best_seller_count_mean = author_dict_df['best_seller_awards'].mean()
best_seller_count_std = author_dict_df['best_seller_awards'].std(ddof=0)
author_best_sellers.plot(x_values, normal_distribution(x_values,best_seller_count_mean,
author_best_sellers.legend(["Normal Distribution", "Actual Distribution"])
author_best_sellers.set_title("Distribution of best seller awards on Amazon (2009-2019)
author_best_sellers.set_xlabel("Number of best seller awards")
rating_distribution_df = author_dict_df['best_seller_awards'].value_counts().to_frame()
rating_distribution_df.index.name = 'Number of Awards'
rating_distribution_df.columns=['Number of Authors']
rating_distribution_df.sort_index().T
```

Out[6]:

Number of Awards	1	2	3	4	5	6	7	8	9	10	11	12
Number of Authors	130	60	18	12	8	7	5	1	2	1	3	1





Looking at the histogram graph above we can clearly see that most authors that are included in the dataset received a best seller book award between the years 2009-2019 and then in subsequent years did not receive a best selling book award. However in the horizontal bar chart above we can see the top authors that were awarded multiple best seller awards in the same time frame. The top best selling authors on amazon graph shows that between 2009-2019 alot of the best selling authors wrote books for the young adult (YA) / Childern categories. Most notably the authors Rick Riordan, Dr. Seuss, Dav Pilkey and J.K. Rowling. The author Jeff Kinney is not a surprising top author in this dataset, because he wrote the Diary of a Wimpy Kid series.

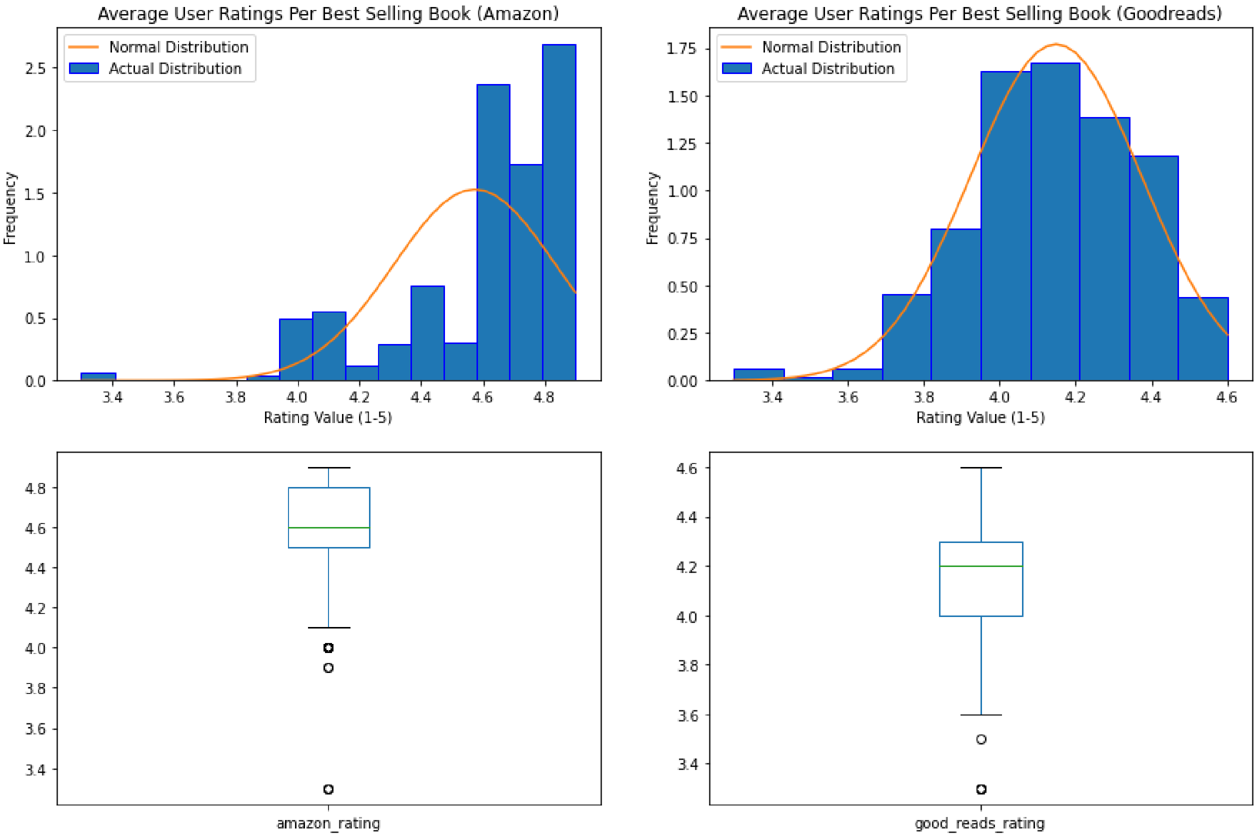
In [7]:

```
fig, axs = plt.subplots(2,2)
amzn_good_reads_df = amzn_books_df.merge(good_reads_books_df,on='Name')
amzn_books_user_rating_df = amzn_good_reads_df['amazon_rating'].to_frame()
good_reads_books_user_rating_df = amzn_good_reads_df['good_reads_rating'].to_frame()
good_reads_hist = good_reads_books_user_rating_df.plot(kind='hist', density=True, bins=
                                                         edgecolor='b', ax=axs[0,1], figs
#create a normalized histogram with 10 bins
book_ratings_hist = amzn_books_user_rating_df.plot(kind='hist', density=True, bins=15,
                                                     edgecolor='b', ax=axs[0,0], figsize=

good_reads_hist.set_title('Average User Ratings Per Best Selling Book (Goodreads)')
good_reads_hist.set_xlabel('Rating Value (1-5)')

book_ratings_hist.set_title('Average User Ratings Per Best Selling Book (Amazon)')
book_ratings_hist.set_xlabel('Rating Value (1-5)')

#create the normal distribution curve to be displayed along with the histogram
#the larger the value at the end the smoother the line becomes
x_values_good_reads = np.linspace(good_reads_books_user_rating_df.values.min(), good_re
x_values_amzn = np.linspace(amzn_books_df['amazon_rating'].min(), amzn_books_df['amazon

#get the average grade
good_ratings_mean = amzn_good_reads_df['good_reads_rating'].mean()
amzn_ratings_mean = amzn_good_reads_df['amazon_rating'].mean()

#get the standard deviation
good_ratings_std = amzn_good_reads_df['good_reads_rating'].std(ddof=0)
amzn_ratings_std = amzn_good_reads_df['amazon_rating'].std(ddof=0)

#plot the normal distribution curve
```

```
good_reads_hist.plot(x_values_good_reads, normal_distribution(x_values_good_reads, good
                                                               good_ratings_std),label='
good_reads_hist.legend(["Normal Distribution", "Actual Distribution"])
good_reads_books_user_rating_df.plot(kind='box',ax=axs[1,1])

book_ratings_hist.plot(x_values_amzn, normal_distribution(x_values_amzn,amzn_ratings_me
                       label='Normal Distribution')
book_ratings_hist.legend(["Normal Distribution", "Actual Distribution"])
amzn_books_user_rating_df.plot(kind='box',ax=axs[1,0])
df_rating_amzn = amzn_books_user_rating_df.describe().T
df_rating_goodreads = good_reads_books_user_rating_df.describe().T
goodreads_amzn_concat = pd.concat([df_rating_amzn,df_rating_goodreads])
goodreads_amzn_concat
```

Out[7]:

	count	mean	std	min	25%	50%	75%	max
amazon_rating	493.0	4.573834	0.262076	3.3	4.5	4.6	4.8	4.9
good_reads_rating	493.0	4.148682	0.225830	3.3	4.0	4.2	4.3	4.6



The graphs above show the average user ratings per best selling book on amazon. The left histogram shows the average user rating that amazon users gave for the books. The right histogram shows the average user rating goodreads users gave for the same books. As we can see it seems that the amazon users tend to give higher ratings than the goodreads users do for the same book. Also the average user ratings on goodreads for the best selling books on amazon seem to be normally distributed. Accordingly the amazon user rating histogram shows the ratings are not as normally distributed as the goodreads user ratings historgram.

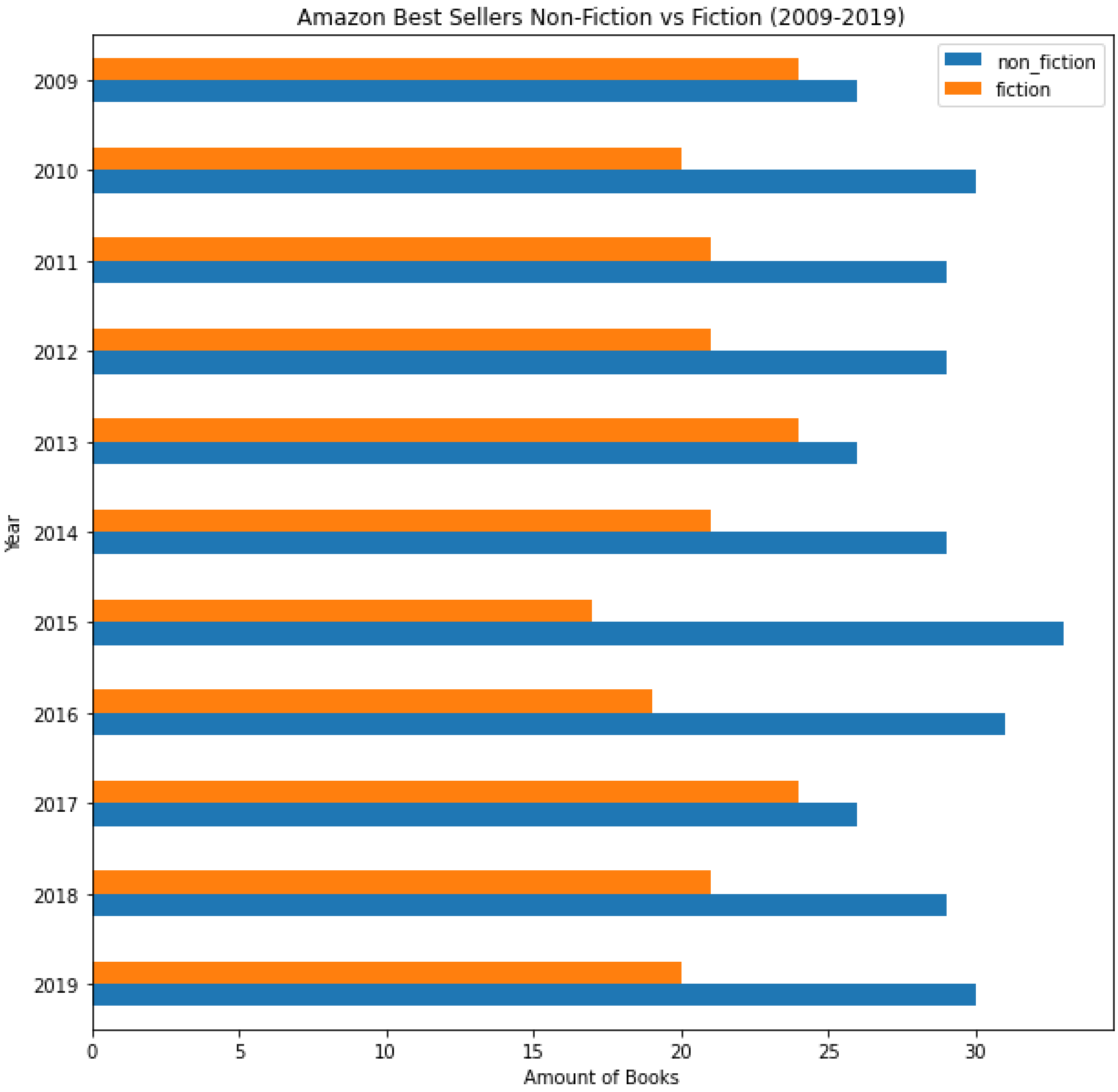
In [8]:

```
amzn_books_year = amzn_books_df.groupby('Year')
sorted_years = sorted(list(set(amzn_books_df['Year'].values)))
fiction_non_fiction_dict = dict()
```

```
for year in sorted_years:
    fiction_non_fiction_df = amzn_books_year.get_group(year)['Genre'].value_counts()
    fiction_non_fiction_values = fiction_non_fiction_df.values
    fiction_non_fiction_dict[str(year)] = {'non_fiction':fiction_non_fiction_values[0],


fiction_non_fiction_values_df = pd.DataFrame(fiction_non_fiction_dict)
non_fiction_vs_fiction_bar = fiction_non_fiction_values_df.T.sort_index(ascending=False
non_fiction_vs_fiction_bar.set_title('Amazon Best Sellers Non-Fiction vs Fiction (2009-
non_fiction_vs_fiction_bar.set_xlabel("Amount of Books")
non_fiction_vs_fiction_bar.set_ylabel("Year")
fiction_non_fiction_values_df
```

Out[8]:

	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
non_fiction	26	30	29	29	26	29	33	31	26	29	30
fiction	24	20	21	21	24	21	17	19	24	21	20



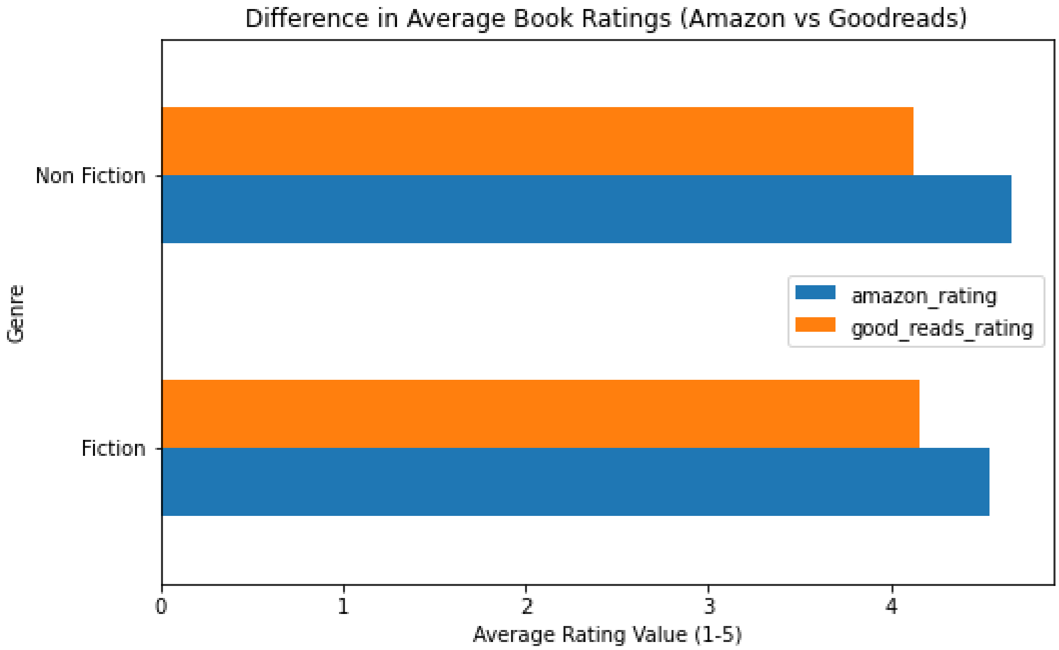
The bar chart above destinguishes the amount of non-fiction bestsellers vs fiction bestsellers during the years 2009 to 2019. From this graph we notice that more people chose to buy non-fiction books over fiction books during 2009-2019. Notice that in 2015 there is a huge difference between fiction and non-fiction best sellers. I believe this is because alot of people who purschase books on amazon were very interested in adult coloring books that year.

In [9]:

```
amzn_book_name_df = amzn_books_df.set_index('Name')

amzn_good_reads_df = amzn_book_name_df.merge(good_reads_books_df,on='Name')

good_reads_vs_amzn_ratings_df = amzn_good_reads_df[['Name','amazon_rating','good_reads_
good_reads_amzn_genre_groups = good_reads_vs_amzn_ratings_df.groupby('Genre')
good_reads_amzn_mean_ratings = good_reads_amzn_genre_groups.mean()
ratings_bar_graph = good_reads_amzn_mean_ratings.plot(kind='barh',figsize=(8,5))
ratings_bar_graph.set_title("Difference in Average Book Ratings (Amazon vs Goodreads)")
ratings_bar_graph.set_xlabel("Average Rating Value (1-5)");
good_reads_amzn_mean_ratings['difference'] = good_reads_amzn_mean_ratings['amazon_ratin
good_reads_amzn_mean_ratings
```

Out[9]:

	amazon_rating	good_reads_rating	difference
Genre			
Fiction	4.536257	4.161988	0.374269
Non Fiction	4.658940	4.118543	0.540397



Looking at the graph above there tends to be more of a difference in ratings in the Non-Fiction category of the best selling books on amazon between goodreads and amazon ratings than the Fiction category of the best selling books. From this graph I would believe Fiction books on amazon have a more accurate general rating than Non-fiction books on amazon best seller lists.

Conclusion:

The best selling authors from 2009-2019 tend to be young adult / children authors. However the bestselling books from 2009-2019 are of the Non-fiction category. The goodreads site tends to have more accurate ratings for best selling books that are listed on amazon. I believe this is because there are more people who review books on goodreads than amazon. Goodread users tend to give conservative ratings for best selling books while amazon users have a tendency to give 5 star ratings more frequently for books they enjoyed.

In [ ]: