Amazon Bestselling Books Analysis

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
import string
import re
```

```
C:\Users\rutik\AppData\Roaming\Python\Python310\site-packages\pandas\core
\arrays\masked.py:60: UserWarning: Pandas requires version '1.3.6' or newe
r of 'bottleneck' (version '1.3.5' currently installed).
  from pandas.core import (
```

In [2]:
```
df = pd.read_csv(r"D:\projects\bestsellers with categories.csv")
```

In [3]:
```
df
```

Out[3]:

	Name	Author	User Rating	Reviews	Price	Year	Genre
0	10-Day Green Smoothie Cleanse	JJ Smith	4.7	17350	8	2016	Non Fiction
1	11/22/63: A Novel	Stephen King	4.6	2052	22	2011	Fiction
2	12 Rules for Life: An Antidote to Chaos	Jordan B. Peterson	4.7	18979	15	2018	Non Fiction
3	1984 (Signet Classics)	George Orwell	4.7	21424	6	2017	Fiction
4	5,000 Awesome Facts (About Everything!) (Natio...	National Geographic Kids	4.8	7665	12	2019	Non Fiction
...	...	...	...	...	...	...	...
545	Wrecking Ball (Diary of a Wimpy Kid Book 14)	Jeff Kinney	4.9	9413	8	2019	Fiction
546	You Are a Badass: How to Stop Doubting Your Gr	Jen Sincero	4.7	14331	8	2016	Non Fiction

Data Preprocessing

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:

In [4]:

```
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(" "))
punctuations = string.punctuation
print('list of punctuations: ',punctuations)

def count_punc(text):
    count = sum(1 for char in text if char in punctuations)
    return round(count/len(text) - text.count(" ")*100, 3)

df['punc%'] = df['Name'].apply(lambda x:count_punc(x))
```

```
list of punctuations:  !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
```

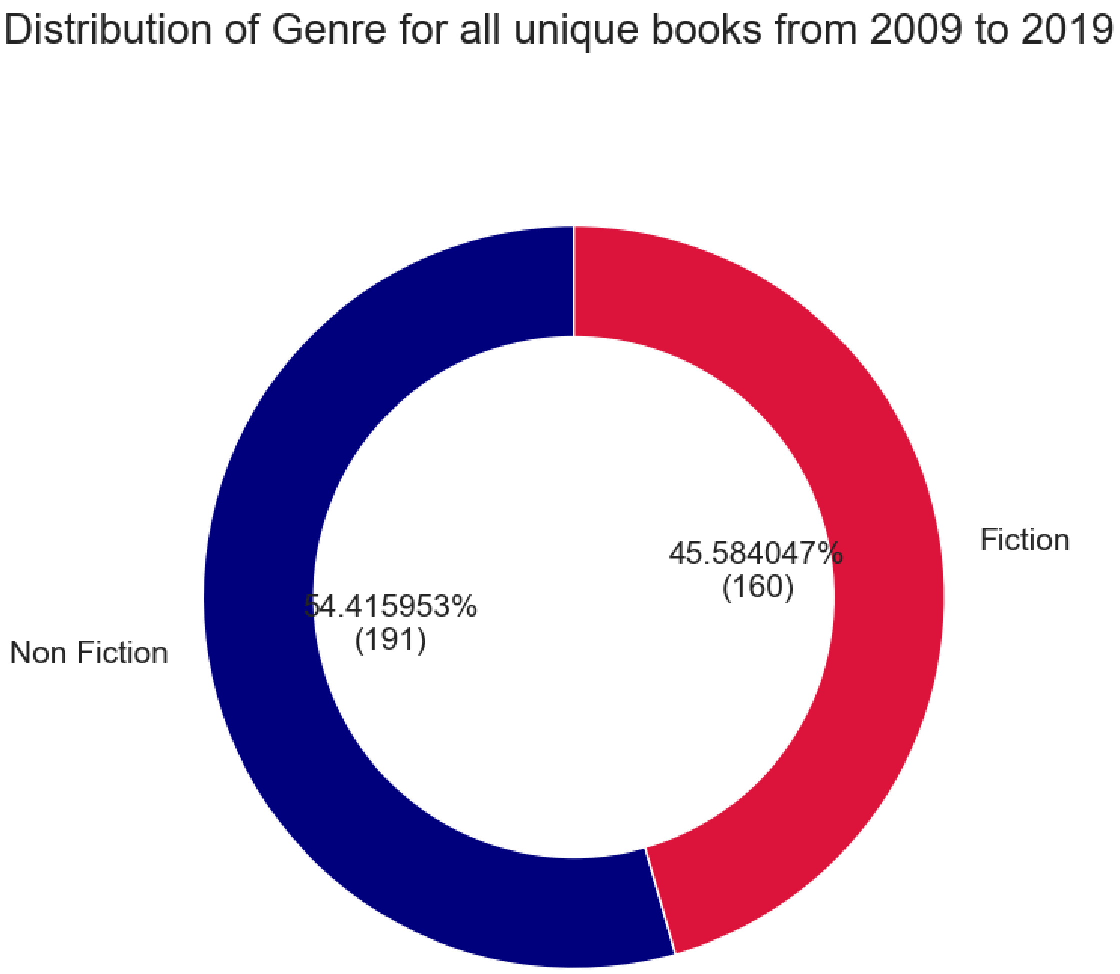
In the data set, Genre is a categorical dummy variable; Fiction and non-fiction. Non-fiction was a more popular category than fiction, each year from 2009 to 2019. Of the 351 unique books, 54.4% were non-fiction and 45.6% were fiction.

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:

In [5]:

```
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']

center_circle = plt.Circle((0,0),0.7,color='white')
plt.pie(x=g_count.values, labels=g_count.index, autopct=make_autopct(g_coun
          startangle=90, textprops={'size': 15}, pctdistance=0.5, colors=ge
ax.add_artist(center_circle)

fig.suptitle('Distribution of Genre for all unique books from 2009 to 2019'
fig.show()
```

```
C:\Users\rutik\AppData\Local\Temp\ipykernel_12100\2527593086.py:21: UserWa
rning: FigureCanvasAgg is non-interactive, and thus cannot be shown
  fig.show()
```

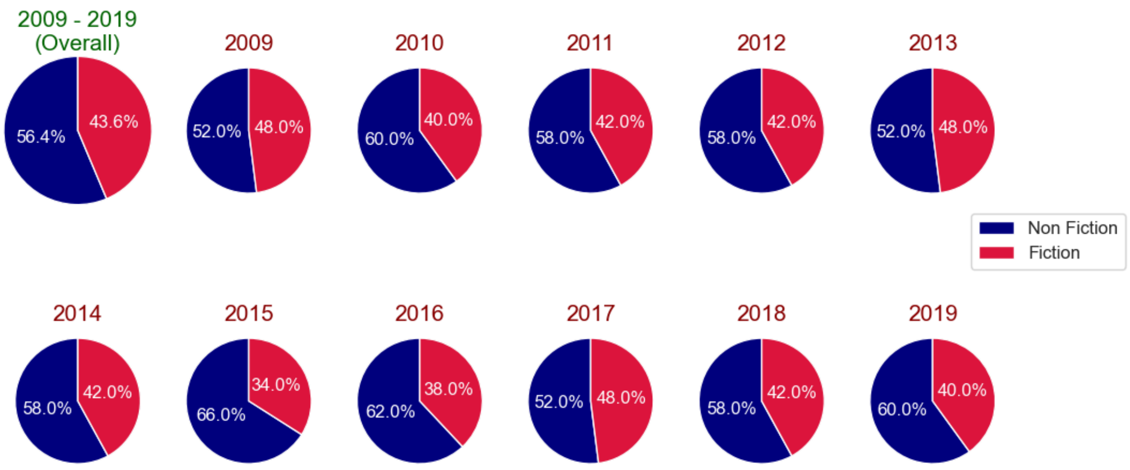


Now let’s visualize the above insights according to each year:

In [6]:

```
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={'f

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
             #fontsize=25)
fig.legend(g_count.index, loc='center right', fontsize=12)
fig.show()
```

```
C:\Users\rutik\AppData\Local\Temp\ipykernel_12100\3063757028.py:29: UserWa
rning: FigureCanvasAgg is non-interactive, and thus cannot be shown
  fig.show()
```

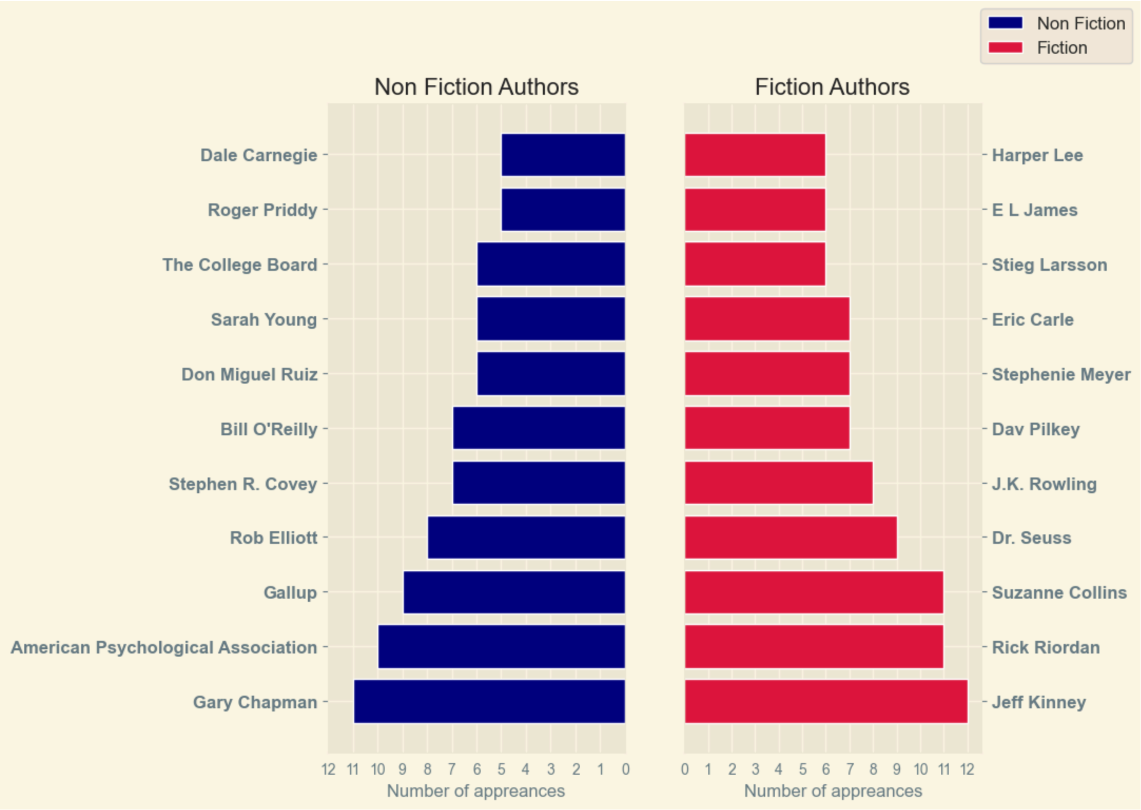


The bestselling authors are selected based on their appearances in the top 50 bestselling books each year, from 2009 to 2019. Now let’s look at the top 10 bestselling authors of both fiction and non-fiction categories:

In [7]:

```
best_nf_authors = df.groupby(['Author', 'Genre']).agg({'Name': 'count'}).un
best_f_authors = df.groupby(['Author', 'Genre']).agg({'Name': 'count'}).uns

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='s
    ax[0].set_xlabel('Number of apprearences')
    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='se
    ax[1].set_title('Fiction Authors')
    ax[1].set_xlabel('Number of apprearences')

    fig.legend(['Non Fiction', 'Fiction'], fontsize=12)

plt.show()
```

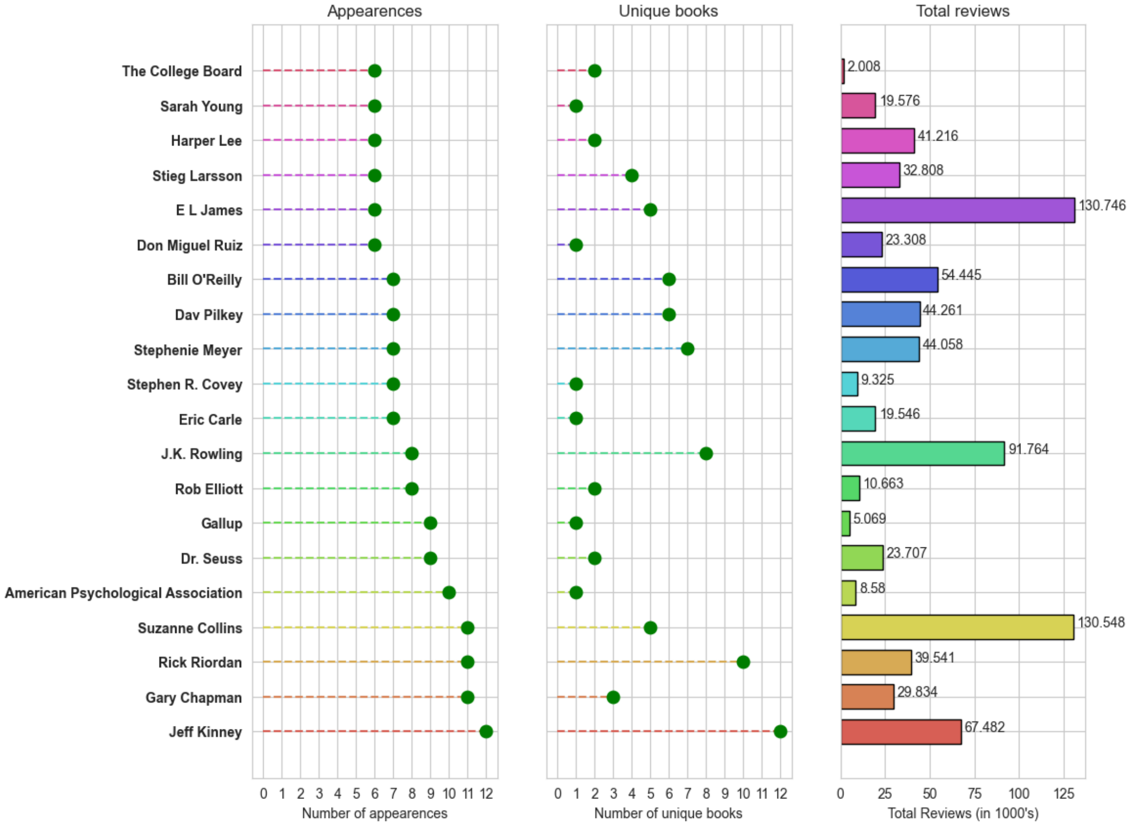
```
C:\Users\rutik\AppData\Local\Temp\ipykernel_12100\2335528297.py:12: UserWa
rning: set_ticklabels() should only be used with a fixed number of ticks,
i.e. after set_ticks() or using a FixedLocator.
  ax[0].set_yticklabels(best_nf_authors.index, fontsize=12, fontweight='se
mibold')
C:\Users\rutik\AppData\Local\Temp\ipykernel_12100\2335528297.py:20: UserWa
rning: set_ticklabels() should only be used with a fixed number of ticks,
i.e. after set_ticks() or using a FixedLocator.
  ax[1].set_yticklabels(best_f_authors.index, fontsize=12, fontweight='sem
ibold')
```


Non Fiction
Fiction
Non Fiction Authors
Fiction Authors
Dale Carnegie
Roger Priddy
The College Board
Sarah Young
Don Miguel Ruiz
Bill O'Reilly
Stephen R. Covey
Rob Elliott
Gallup
American Psychological Association
Gary Chapman
Harper Lee
E L James
Stieg Larsson
Eric Carle
Stephenie Meyer
Dav Pilkey
J.K. Rowling
Dr. Seuss
Suzanne Collins
Rick Riordan
Jeff Kinney
12 11 10 9 8 7 6 5 4 3 2 1 0
0 1 2 3 4 5 6 7 8 9 10 11 12
Number of appreances
Number of appreances

In [8]:

```
n_best = 20
top_authors=df.Author.value_counts().nlargest(n_best)
no_dup = df.drop_duplicates('Name')

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=c
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()/100
ax[1].hlines(y=top_authors.index , xmin=0, xmax=book_count, color=color, li
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
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.show()
```

```
C:\Users\rutik\AppData\Local\Temp\ipykernel_12100\3894942863.py:13: UserWa
rning: set_ticklabels() should only be used with a fixed number of ticks,
i.e. after set_ticks() or using a FixedLocator.
  ax[0].set_yticklabels(top_authors.index, fontweight='semibold')
```



Author Jeff Kinney is the best-selling author with 12 appearances in best-selling books from 2009 to 2019.

In [ ]:

```

```

In [ ]:

```
P
```