CODE:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import string

import re

from google.colab import files

uploaded = files.upload()

df = pd.read\_csv(r'bestsellers with categories.csv')

df.rename(columns={"User Rating": "User\_Rating"}, inplace=True)

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

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))

no\_dup = df.drop\_duplicates('Name')

g\_count = no\_dup['Genre'].value\_counts()

fig, ax = plt.subplots(figsize=(8, 8))

genre\_col = ['navy', 'crimson']

explode = [0.1, 0]  # Explode the first slice (optional)

wedges, texts, autotexts = ax.pie(g\_count.values, explode=explode, labels=g\_count.index, autopct='%.2f%%',

                                  startangle=90, textprops={'size': 14}, colors=genre\_col, pctdistance=0.85)

# Customize wedges and texts

for wedge in wedges:

    wedge.set\_edgecolor('white')

    wedge.set\_linewidth(1)

for text, autotext in zip(texts, autotexts):

    text.set\_fontsize(14)

    text.set\_fontweight('bold')

    autotext.set\_fontsize(12)

    autotext.set\_color('white')

# Add a central circle

center\_circle = plt.Circle((0, 0), 0.7, color='white')

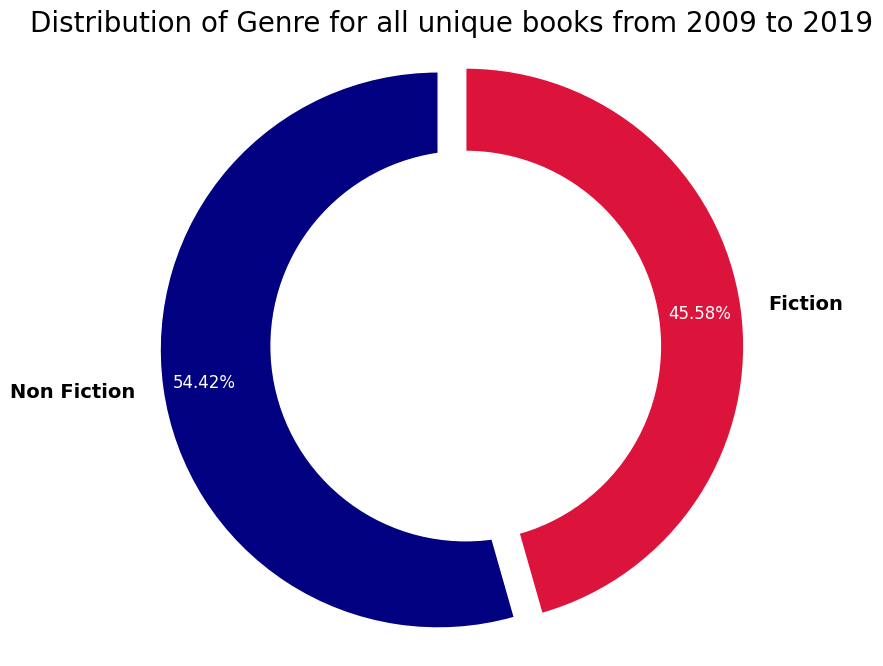
ax.add\_artist(center\_circle)

# Title and aspect ratio

ax.set\_title('Distribution of Genre for all unique books from 2009 to 2019', fontsize=20)

ax.axis('equal')

plt.show()



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))

# Adjust the spacing between subplots

fig.subplots\_adjust(hspace=0.4, wspace=0.3)

# Create a custom color palette for the pie charts

colors = ['#ff9999', '#66b3ff']

# Set the font size for the titles

title\_fontsize = 14

# Iterate over the subplots and create the pie charts

for i, year in enumerate(y1):

    counts = df[df['Year'] == year]['Genre'].value\_counts()

    ax[0, i+1].pie(x=counts.values, labels=None, autopct='%.1f%%',

                   startangle=90, textprops={'size': 12, 'color': 'white'},

                   pctdistance=0.7, colors=colors, radius=1.1)

    ax[0, i+1].set\_title(year, color='darkred', fontsize=title\_fontsize)

for i, year in enumerate(y2):

    counts = df[df['Year'] == year]['Genre'].value\_counts()

    ax[1, i].pie(x=counts.values, labels=None, autopct='%.1f%%',

                 startangle=90, textprops={'size': 12, 'color': 'white'},

                 pctdistance=0.7, colors=colors, radius=1.1)

    ax[1, i].set\_title(year, color='darkred', fontsize=title\_fontsize)

# Set the title and font size for the overall chart

overall\_title = ax[0, 0].set\_title('2009 - 2019\n(Overall)', color='darkgreen', fontsize=title\_fontsize)

# Create a legend for the genres

fig.legend(g\_count.index, loc='center right', fontsize=12)

# Remove unnecessary spines and labels

for row in ax:

    for col in row:

        col.axis('equal')

        col.spines['top'].set\_visible(False)

        col.spines['right'].set\_visible(False)

        col.spines['bottom'].set\_visible(False)

        col.spines['left'].set\_visible(False)

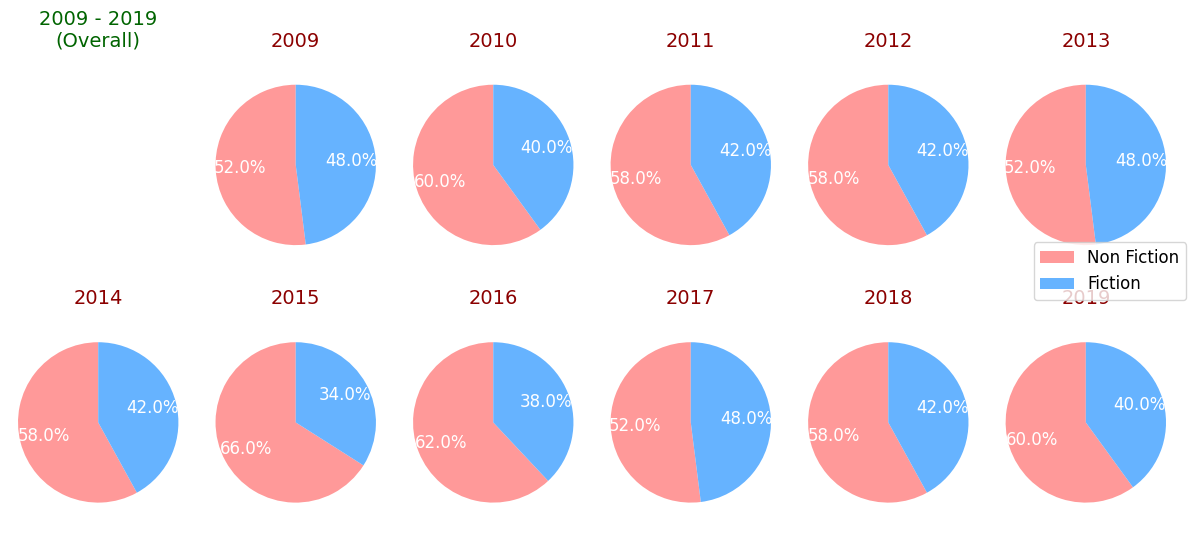
        col.set\_xticks([])

        col.set\_yticks([])

# Adjust the position of the subplots and title

fig.tight\_layout(rect=[0, 0.03, 1, 0.95])

plt.show()



st\_nf\_authors = df.groupby(['Author', 'Genre']).agg({'Name': 'count'}).unstack()['Name', 'Non Fiction'].sort\_values(ascending=False)[:11]

best\_nf\_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=(10, 8))

    # Create a custom color palette

    colors = ['#ff9999', '#66b3ff']

    # Plot the horizontal bar chart for Non Fiction Authors

    ax[0].barh(y=best\_nf\_authors.index, width=best\_nf\_authors.values, color=colors[0])

    ax[0].invert\_xaxis()

    ax[0].yaxis.tick\_left()

    ax[0].set\_xticks(np.arange(max(best\_nf\_authors.values) + 1))

    ax[0].set\_yticklabels(best\_nf\_authors.index, fontsize=12, fontweight='semibold')

    ax[0].set\_xlabel('Number of Appearances', fontsize=12)

    ax[0].set\_title('Top Non Fiction Authors', fontsize=14)

    # Plot the horizontal bar chart for Fiction Authors

    ax[1].barh(y=best\_nf\_authors.index, width=best\_nf\_authors.values, color=colors[1])

    ax[1].yaxis.tick\_right()

    ax[1].set\_xticks(np.arange(max(best\_nf\_authors.values) + 1))

    ax[1].set\_yticklabels(best\_nf\_authors.index, fontsize=12, fontweight='semibold')

    ax[1].set\_title('Top Fiction Authors', fontsize=14)

    ax[1].set\_xlabel('Number of Appearances', fontsize=12)

    # Set the legend

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

    # Remove spines

    ax[0].spines['right'].set\_visible(False)

    ax[0].spines['top'].set\_visible(False)

    ax[1].spines['left'].set\_visible(False)

    ax[1].spines['top'].set\_visible(False)

    # Adjust space between subplots

    plt.subplots\_adjust(wspace=0.4)

    # Add a horizontal line at the bottom

    for a in ax:

        a.spines['bottom'].set\_linewidth(0.5)

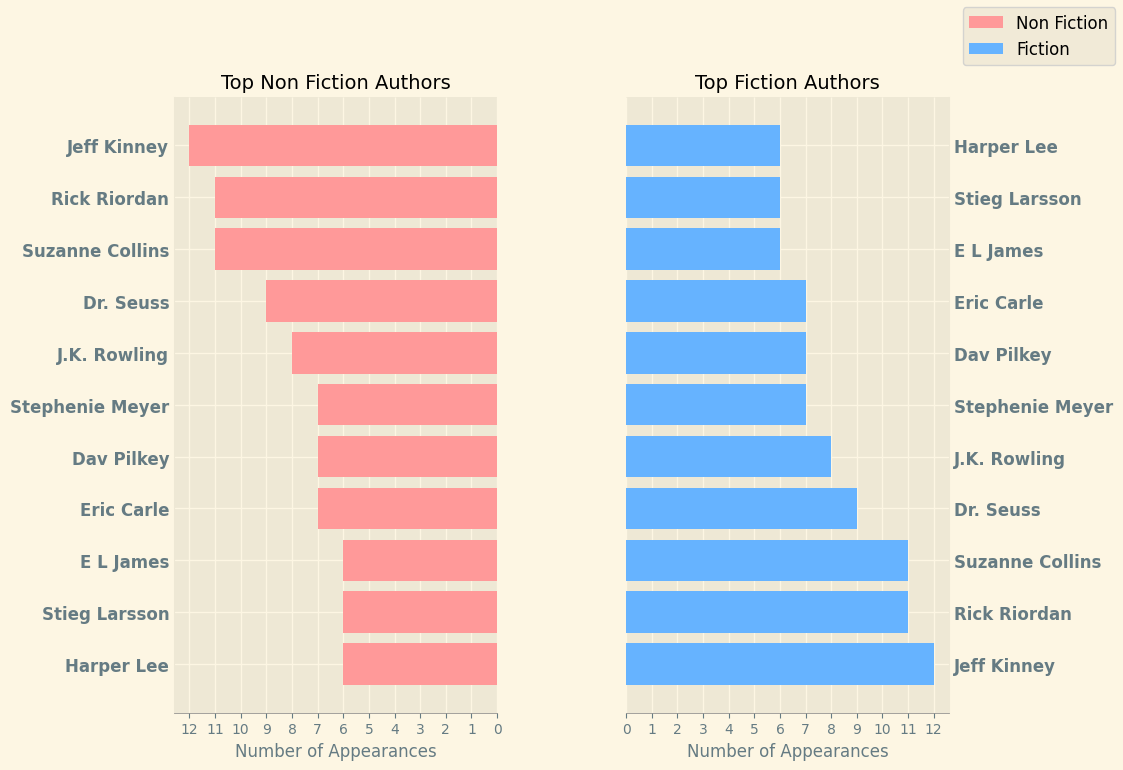
        a.spines['bottom'].set\_color('gray')

        a.tick\_params(axis='y', which='both', length=0)

    # Invert y-axis to show the highest count at the top

    ax[0].invert\_yaxis()

    plt.show()



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=color, linestyles='dashed')

ax[0].plot(top\_authors.values, top\_authors.index, 'go', markersize=9)

ax[0].set\_xlabel('Number of appearances')

ax[0].set\_xticks(np.arange(top\_authors.values.max() + 1))

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)")

plt.show()

