Contents

Introduction

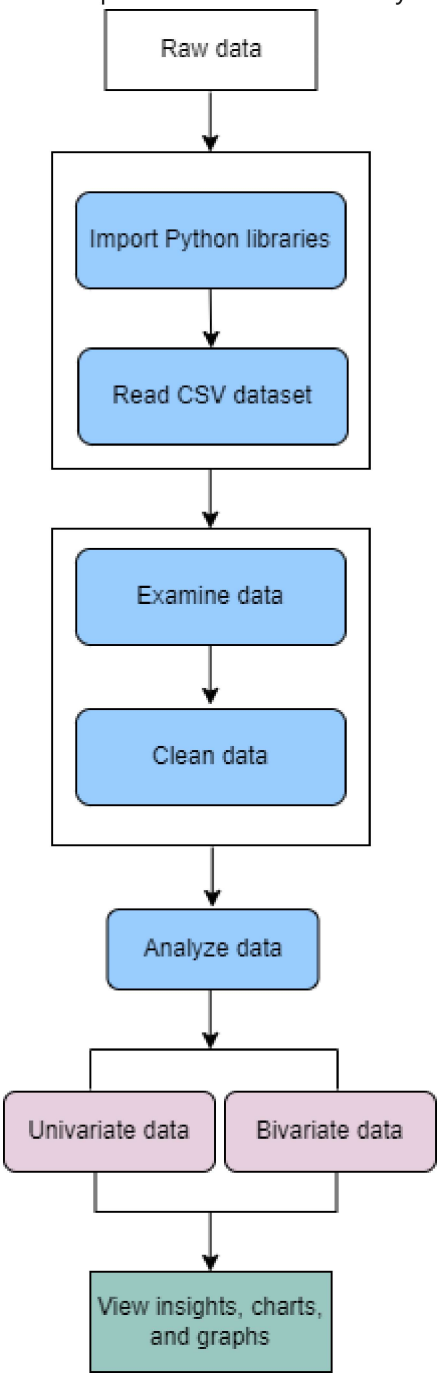
Back to Top

This project analyzes data from the *best_selling_books* CSV file. By analyzing a dataset of best-selling books, you can gain valuable insights into the factors that contribute to a book's success. Use the Exploratory Data Analysis (EDA) technique to identify which factors are most influential in the success of best-selling books. Additionally, you can use data visualization techniques to present your findings effectively.

The *best_selling_books* CSV file contains the following data columns:

- Book
- Authors
- Original language
- First published
- Approximate sales in millions
- Genre

The following diagram depicts the process of the data analysis required for the *best_selling_books* dataset.



For a thorough data analysis, use these data points and examine the outcome.

Getting Started

Prerequisites

Back to Top

Before you perform data analysis,

- Ensure that the *best-selling-books.csv* file is available in the Python working directory. You can find the location of the Python working directory by running the `pwd` command.
- Ensure that you have already installed the following necessary Python libraries:
  - **Pandas and Numpy**: Use for data processing of a CSV file, data manipulation, and numerical calculations
  - **Seabom, Matplotlib, and Plotly**: Use for data visualization. You can view data on attractive graphs, such as histograms, box plots, pie charts, and so on.

Importing Python libraries

Back to Top

In [1]:
```
#importing Required library
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import numpy as np # linear algebra
import seaborn as sns # to get summary statistics
import matplotlib.pyplot as plt # to get summary statistics
%matplotlib inline
```

Importing Plotly libraries

Back to Top

In [2]:
```
#importing plotly library
from plotly.offline import iplot
import plotly as py
import plotly.tools as tls
import cufflinks as cf
py.offline.init_notebook_mode(connected=True) #Turning on notebook mode
cf.go_offline()
```

Reading dataset

Back to Top

Using the `pd.read_csv()` function, convert best-selling-books CSV data to a pandas DataFrame.

In [3]:
```
df=pd.read_csv(r"best-selling-books.csv") #dataset
```

Understanding the data

Back to Top

Before making any inferences, examine all variables in the data. The primary aim of data understanding is to extract key insights about the data, including its size (number of rows and columns), data values, data types, and any missing values in the dataset.

- **shape**: Displays the number of observations(rows) and features(columns) in the dataset
- **info()**: Assists in comprehending data: reveals data types, record counts per column, presence of null values, data types, and dataset memory consumption
- **head()**: Displays the top 5 observations/rows of the dataset
- **tail()**: Displays the bottom 5 observations/rows of the dataset

In [4]:
```
df.shape #displays 174 rows and 6 columns in the dataset
```

Out[4]:
```
(174, 6)
```

Information of dataset

Back to Top

`df.info()` shows the following variables:

- **Categorical variables**: Book, Authors, Original language, and Genre with the object data type. The Genre variable/column has missing values.
- **Numeric variables**: First published with the int64 data type and Approximate sales in millions with the float64 data type.

In [5]:
```
df.info() #get information on dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 174 entries, 0 to 173
Data columns (total 6 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Book                          174 non-null    object
 1   Authors                       174 non-null    object
 2   Original language             174 non-null    object
 3   First published               174 non-null    int64
 4   Approximate sales in millions  174 non-null    float64
 5   Genre                         118 non-null    object
dtypes: float64(1), int64(1), object(4)
memory usage: 8.3+ KB
```

In [6]:
```
df.head() #show top rows
```

Out[6]:

	Book	Authors	Original language	First published	Approximate sales in millions	Genre
0	A Tale of Two Cities	Charles Dickens	English	1859	200.0	Historical fiction
1	The Little Prince (Le Petit Prince)	Antoine de Saint-Exupéry	French	1943	200.0	Novella
2	Harry Potter and the Philosopher's Stone	J. K. Rowling	English	1997	120.0	Fantasy
3	And Then There Were None	Agatha Christie	English	1939	100.0	Mystery
4	Dream of the Red Chamber (紅樓夢)	Cao Xueqin	Chinese	1791	100.0	Family saga

In [7]:
```
df.tail() #show bottom rows
```

Out[7]:

	Book	Authors	Original language	First published	Approximate sales in millions	Genre
169	The Goal	Eliyahu M. Goldratt	English	1984	10.0	NaN
170	Fahrenheit 451	Ray Bradbury	English	1953	10.0	NaN
171	Angela's Ashes	Frank McCourt	English	1996	10.0	NaN
172	The Story of My Experiments with Truth (સત્યનો...	Mohandas Karamchand Gandhi	Gujarati	1929	10.0	NaN
173	Bridget Jones's Diary	Helen Fielding	English	1996	10.0	NaN

Cleaning the data

Back to Top

The following variables/column names are too lengthy and not easy to understand. You can rename them for easy data processing. | Original column name| New column name| | ---------- | ---------- | | First published| Year | | Original language| Language |

Changing lengthy column names

Back to Top

Run the `df.rename(columns={'original column name':'new column name'})` command.

In [8]:
```
#Changing the column names
df = df.rename(columns={'Original language':'Language','First published':'Year',
                        'Approximate sales in millions':'Sales'})
df.columns
```

Out[8]:
```
Index(['Book', 'Authors', 'Language', 'Year', 'Sales', 'Genre'], dtype='object')
```

Cleaning messy book titles

Back to Top

Cleaning column headers is an essential data preparation step. Some of the book titles contain unwanted characters. You must clean these messy column headers to improve data clarity and usability. Remove these characters from the headers to significantly enhance data visualization.

In [9]:
```
print(df.Book.unique())
print(df.Book.nunique())
```

```
['A Tale of Two Cities' 'The Little Prince (Le Petit Prince)'
 "Harry Potter and the Philosopher's Stone" 'And Then There Were None'
 'Dream of the Red Chamber (紅樓夢)' 'The Hobbit'
 'The Lion, the Witch and the Wardrobe' 'She: A History of Adventure'
 'Vardi Wala Gunda (वर्दी वाला गुंडा)' 'The Da Vinci Code'
 'Harry Potter and the Chamber of Secrets'
 'Harry Potter and the Prisoner of Azkaban'
 'Harry Potter and the Goblet of Fire'
 'Harry Potter and the Order of the Phoenix'
 'Harry Potter and the Half-Blood Prince'
 'Harry Potter and the Deathly Hallows' 'The Alchemist (O Alquimista)'
 'The Catcher in the Rye' 'The Bridges of Madison County'
 'Ben-Hur: A Tale of the Christ' 'You Can Heal Your Life'
 'One Hundred Years of Solitude (Cien años de soledad)' 'Lolita' 'Heidi'
 'The Common Sense Book of Baby and Child Care' 'Anne of Green Gables'
 'Black Beauty' 'The Name of the Rose (Il Nome della Rosa)'
 'The Eagle Has Landed' 'Watership Down' 'The Hite Report'
 "Charlotte's Web" 'The Ginger Man' 'The Tale of Peter Rabbit'
 'Jonathan Livingston Seagull' 'The Very Hungry Caterpillar'
 'A Message to Garcia' 'To Kill a Mockingbird' 'Flowers in the Attic'
 'Cosmos' "Sophie's World (Sofies verden)" 'Angels & Demons'
 'Kane and Abel' 'How the Steel Was Tempered (Как закалялась сталь)'
 'War and Peace (Война и мир)'
 'The Adventures of Pinocchio (Le avventure di Pinocchio)'
 'The Diary of Anne Frank (Het Achterhuis)' 'Your Erroneous Zones'
 'The Thorn Birds' 'The Purpose Driven Life' 'The Kite Runner'
 'Valley of the Dolls' 'Alcoholics Anonymous Big Book'
 'How to Win Friends and Influence People' 'The Great Gatsby'
 'Gone with the Wind' 'Rebecca' 'Nineteen Eighty-Four'
 'The Revolt of Mamie Stover'
 'The Girl with the Dragon Tattoo (Män som hatar kvinnor)'
 'The Lost Symbol' 'The Hunger Games' 'James and the Giant Peach'
 'The Young Guard (Молодая гвардия)' 'Who Moved My Cheese?'
 'A Brief History of Time' 'Paul et Virginie' 'Lust for Life'
 'The Wind in the Willows' 'The 7 Habits of Highly Effective People'
 'Virgin Soil Upturned (Поднятая целина)' 'The Celestine Prophecy'
 'The Fault in Our Stars' 'The Girl on the Train' 'The Shack'
 'Uncle Styopa (Дядя Стёпа)' 'The Godfather' 'Love Story' 'Catching Fire'
 'Mockingjay' 'Kitchen (キッチン)' 'Andromeda Nebula (Туманность Андромеды)'
 'Autobiography of a Yogi (योगी कथामृत)' 'Gone Girl'
 'All Quiet on the Western Front (Im Westen nichts Neues)'
 'The Bermuda Triangle' 'Things Fall Apart ' 'Animal Farm '
 'Wolf Totem (狼图腾)' 'The Happy Hooker: My Own Story' 'Jaws'
 'Love You Forever' "The Women's Room"
 "What to Expect When You're Expecting" 'Adventures of Huckleberry Finn'
 'The Secret Diary of Adrian Mole, Aged 13¾' 'Pride and Prejudice'
 'Kon-Tiki: Across the Pacific in a Raft (Kon-Tiki ekspedisjonen)'
 'The Good Soldier Švejk (Osudy dobrého vojáka Švejka za světové války)'
 'Where the Wild Things Are' 'The Power of Positive Thinking' 'The Secret'
 'Fear of Flying' 'Dune' 'Charlie and the Chocolate Factory'
 'The Naked Ape' 'Where the Crawdads Sing'
 'Totto-chan, the Little Girl at the Window (窓ぎわのトットちゃん)' 'Matilda'
 'The Book Thief' 'The Horse Whisperer' 'Goodnight Moon'
 'The Neverending Story (Die unendliche Geschichte)'
 'All the Light We Cannot See' 'Fifty Shades of Grey' 'The Outsiders'
 'Guess How Much I Love You' 'Shōgun' 'The Poky Little Puppy'
 'The Pillars of the Earth' 'Perfume (Das Parfum)' 'The Grapes of Wrath'
 'The Shadow of the Wind (La sombra del viento)' 'Interpreter of Maladies'
 'Becoming' "The Hitchhiker's Guide to the Galaxy" 'Tuesdays with Morrie'
 "God's Little Acre" "Follow Your Heart (Va' dove ti porta il cuore)"
```

```
 'A Wrinkle in Time' 'Long Walk to Freedom' 'The Old Man and the Sea'
 'Life After Life' 'Peyton Place ' 'The Giver' 'Me Before You'
 'Norwegian Wood (ノルウェイの森)' 'The Plague (La Peste)'
 'No Longer Human (人間失格)'
 "Man's Search for Meaning (Ein Psychologe erlebt das Konzentrationslager)"
 'The Divine Comedy (La Divina Commedia)' 'The Prophet'
 'The Boy in the Striped Pyjamas' 'The Exorcist' 'The Gruffalo'
 'Fifty Shades Darker' 'Tobacco Road' "Ronia, the Robber's Daughter"
 'The Cat in the Hat' 'Diana: Her True Story' 'The Help' 'Catch-22'
 "The Stranger (L'Étranger)" 'Eye of the Needle' 'The Lovely Bones'
 'Wild Swans' 'Santa Evita' 'Night (Un di Velt Hot Geshvign)'
 'Confucius from the Heart (于丹《论语》心得)' 'The Total Woman'
 'Knowledge-value Revolution (知価革命)'
 "Problems in China's Socialist Economy (中国社会主义经济问题研究)"
 'What Color Is Your Parachute?' 'The Dukan Diet' 'The Joy of Sex'
 'The Gospel According to Peanuts' 'The Subtle Art of Not Giving a Fuck'
 'Life of Pi' 'The Front Runner' 'The Goal' 'Fahrenheit 451'
 "Angela's Ashes"
 'The Story of My Experiments with Truth (સત્યના પ્રયોગો અથવા આત્મકથા)'
 "Bridget Jones's Diary"]
174
```

To remove multiple characters from a string in Python, you can use regular expressions from the `re` module.

1. Import the regular expression library by using the `import re` function
2. Define the `remove_text_within_parentheses(Book)` custom function that uses regular expressions to remove text enclosed within parentheses from a given string Book.
3. Specify the `re.sub(r'\([^)]*\)'` regular expression that matches any text enclosed within parentheses and replaces it with an empty string `('')`, effectively removing the content within the brackets.
4. Apply the custom function to the 'Book' column using the `apply()` method.

In [10]:
```
import re
# Function to remove text within parentheses from a string
def remove_text_within_parentheses(Book):
    return re.sub(r'\([^)]*\)', '', Book)
df['Book'] = df['Book'].apply(remove_text_within_parentheses)
```

Verifying updated book titles

Back to Top

In [11]:
```
print(df.Book.unique())
print(df.Book.nunique())
```

```
['A Tale of Two Cities' 'The Little Prince '
 "Harry Potter and the Philosopher's Stone" 'And Then There Were None'
 'Dream of the Red Chamber ' 'The Hobbit'
 'The Lion, the Witch and the Wardrobe' 'She: A History of Adventure'
 'Vardi Wala Gunda ' 'The Da Vinci Code'
 'Harry Potter and the Chamber of Secrets'
 'Harry Potter and the Prisoner of Azkaban'
 'Harry Potter and the Goblet of Fire'
 'Harry Potter and the Order of the Phoenix'
 'Harry Potter and the Half-Blood Prince'
 'Harry Potter and the Deathly Hallows' 'The Alchemist '
 'The Catcher in the Rye' 'The Bridges of Madison County'
 'Ben-Hur: A Tale of the Christ' 'You Can Heal Your Life'
 'One Hundred Years of Solitude ' 'Lolita' 'Heidi'
 'The Common Sense Book of Baby and Child Care' 'Anne of Green Gables'
 'Black Beauty' 'The Name of the Rose ' 'The Eagle Has Landed'
 'Watership Down' 'The Hite Report' "Charlotte's Web" 'The Ginger Man'
 'The Tale of Peter Rabbit' 'Jonathan Livingston Seagull'
 'The Very Hungry Caterpillar' 'A Message to Garcia'
 'To Kill a Mockingbird' 'Flowers in the Attic' 'Cosmos' "Sophie's World "
 'Angels & Demons' 'Kane and Abel' 'How the Steel Was Tempered '
 'War and Peace ' 'The Adventures of Pinocchio '
 'The Diary of Anne Frank ' 'Your Erroneous Zones' 'The Thorn Birds'
 'The Purpose Driven Life' 'The Kite Runner' 'Valley of the Dolls'
 'Alcoholics Anonymous Big Book' 'How to Win Friends and Influence People'
 'The Great Gatsby' 'Gone with the Wind' 'Rebecca' 'Nineteen Eighty-Four'
 'The Revolt of Mamie Stover' 'The Girl with the Dragon Tattoo '
 'The Lost Symbol' 'The Hunger Games' 'James and the Giant Peach'
 'The Young Guard ' 'Who Moved My Cheese?' 'A Brief History of Time'
 'Paul et Virginie' 'Lust for Life' 'The Wind in the Willows'
 'The 7 Habits of Highly Effective People' 'Virgin Soil Upturned '
 'The Celestine Prophecy' 'The Fault in Our Stars' 'The Girl on the Train'
 'The Shack' 'Uncle Styopa ' 'The Godfather' 'Love Story' 'Catching Fire'
 'Mockingjay' 'Kitchen ' 'Andromeda Nebula ' 'Autobiography of a Yogi '
 'Gone Girl' 'All Quiet on the Western Front ' 'The Bermuda Triangle'
 'Things Fall Apart ' 'Animal Farm ' 'Wolf Totem '
 'The Happy Hooker: My Own Story' 'Jaws' 'Love You Forever'
 "The Women's Room" "What to Expect When You're Expecting"
 'Adventures of Huckleberry Finn'
 'The Secret Diary of Adrian Mole, Aged 13¾' 'Pride and Prejudice'
 'Kon-Tiki: Across the Pacific in a Raft ' 'The Good Soldier Švejk '
 'Where the Wild Things Are' 'The Power of Positive Thinking' 'The Secret'
 'Fear of Flying' 'Dune' 'Charlie and the Chocolate Factory'
 'The Naked Ape' 'Where the Crawdads Sing'
 'Totto-chan, the Little Girl at the Window ' 'Matilda' 'The Book Thief'
 'The Horse Whisperer' 'Goodnight Moon' 'The Neverending Story '
 'All the Light We Cannot See' 'Fifty Shades of Grey' 'The Outsiders'
 'Guess How Much I Love You' 'Shōgun' 'The Poky Little Puppy'
 'The Pillars of the Earth' 'Perfume ' 'The Grapes of Wrath'
 'The Shadow of the Wind ' 'Interpreter of Maladies' 'Becoming'
 "The Hitchhiker's Guide to the Galaxy" 'Tuesdays with Morrie'
 "God's Little Acre" 'Follow Your Heart ' 'A Wrinkle in Time'
 'Long Walk to Freedom' 'The Old Man and the Sea' 'Life After Life'
 'Peyton Place ' 'The Giver' 'Me Before You' 'Norwegian Wood '
 'The Plague ' 'No Longer Human ' "Man's Search for Meaning "
 'The Divine Comedy ' 'The Prophet' 'The Boy in the Striped Pyjamas'
 'The Exorcist' 'The Gruffalo' 'Fifty Shades Darker' 'Tobacco Road'
 "Ronia, the Robber's Daughter" 'The Cat in the Hat'
 'Diana: Her True Story' 'The Help' 'Catch-22' 'The Stranger '
 'Eye of the Needle' 'The Lovely Bones' 'Wild Swans' 'Santa Evita'
```

```
 'Night ' 'Confucius from the Heart ' 'The Total Woman'
 'Knowledge-value Revolution ' "Problems in China's Socialist Economy "
 'What Color Is Your Parachute?' 'The Dukan Diet' 'The Joy of Sex'
 'The Gospel According to Peanuts' 'The Subtle Art of Not Giving a Fuck'
 'Life of Pi' 'The Front Runner' 'The Goal' 'Fahrenheit 451'
 "Angela's Ashes" 'The Story of My Experiments with Truth '
 "Bridget Jones's Diary"]
174
```

Finding missing values

Back to Top

Use the `isnull()` function to identify null values in the data. Run the `df.isnull().sum()` to retrieve the number of missing records in each column.

In [12]:
```
df.isnull().sum()
```

Out[12]:
```
Book         0
Authors      0
Language     0
Year         0
Sales        0
Genre       56
dtype: int64
```

Deleting empty rows

Back to Top

The Genre column contains 56 empty rows. These blank rows do not add any value to the data analysis. Therefore, you must delete these rows to get accurate data analysis. Run the `df.dropna(how='any',inplace=True)` command to delete the empty rows.

In [13]:
```
#deleting empty/null rows from the Genre column. It has 56 empty rows.
df.dropna(how='any',inplace=True)
```

Verifying updated rows

Back to Top

In [14]:
```
#checking the NaN values
df.isnull().sum()
```

Out[14]:
```
Book        0
Authors     0
Language    0
Year        0
Sales       0
Genre       0
dtype: int64
```

Retrieving statistics summary

Back to Top

Statistics summary provides a high-level overview of the data to identify outliers, data entry errors, and distribution of data such as determining whether the data follows a normal distribution or exhibits deviation to the left or right.

Use the `describe()` function to retrieve all statistics summary of data that belongs to the numerical data type such as int (Year column) and float64 (Sales column).

In [15]:
```
df.describe() #to analyze dataset #describe function is applicable only to numerical c
```

Out[15]:

	Year	Sales
count	118.000000	118.000000
mean	1961.483051	38.765254
std	44.992636	30.295672
min	1788.000000	10.400000
25%	1945.000000	20.000000
50%	1971.000000	30.000000
75%	1992.750000	50.000000
max	2018.000000	200.000000

Analyzing univariate data

Back to Top

Univariate analysis suggests that you can examine or visualize a single variable individually. You can perform univariate analysis for both categorical and numerical variables.

- Categorical variables: Visualize using a Count plot, Bar chart, Pie plot, and so on.
- Numerical variables: Visualize using Histogram, Box plot, Density plot, and so on.

Identify top-selling books

Back to Top

To identify the top selling books, calculate the summary statistics for the `Sales` categorical variable in the DataFrame.

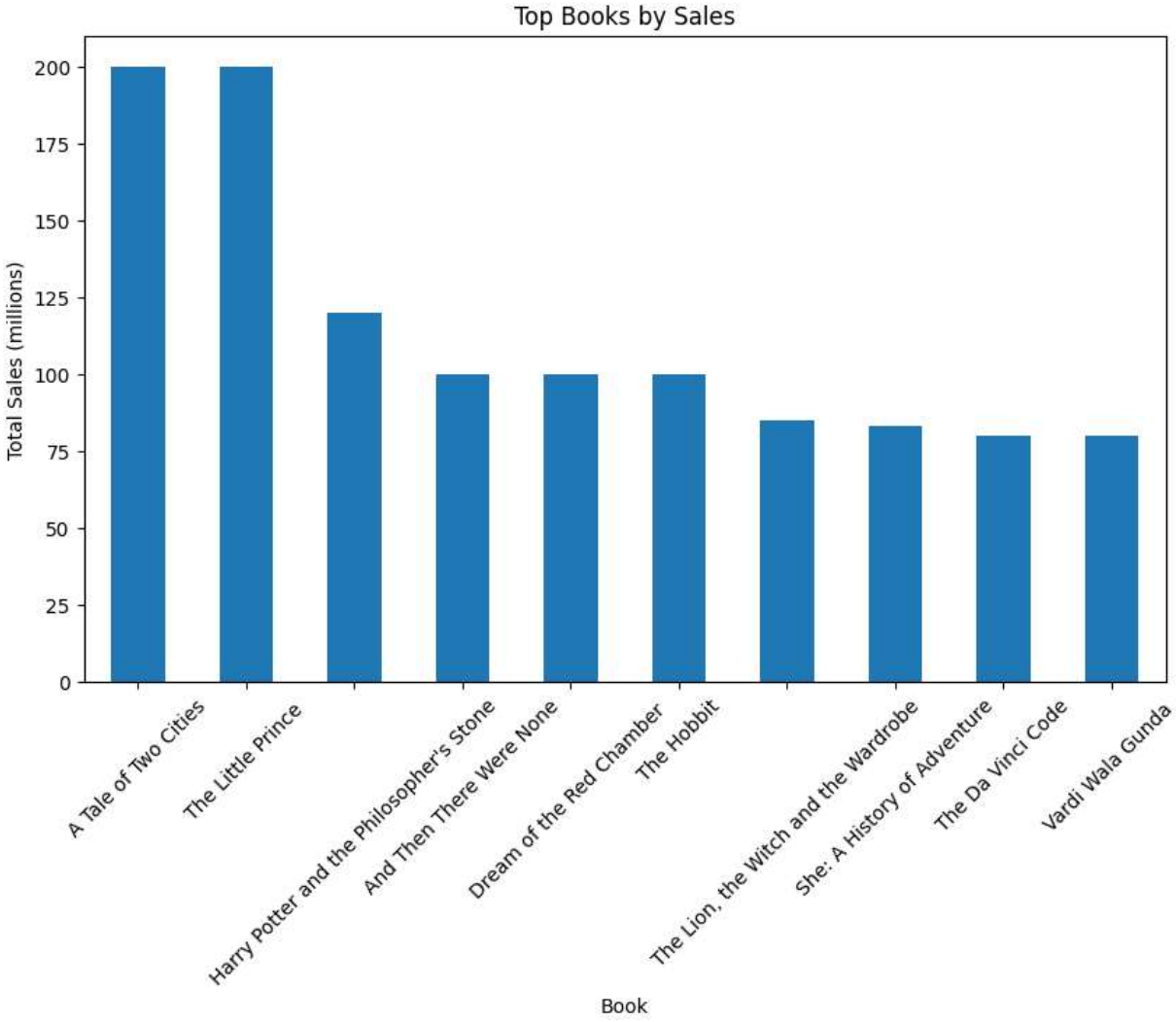
In [16]:
```
top_selling_books = df.sort_values(by="Sales", ascending=False).head(10)
print ("Top selling books:\n", top_selling_books)
```

```
Top selling books:
                                        Book                  Authors  \
0                       A Tale of Two Cities           Charles Dickens
1                         The Little Prince   Antoine de Saint-Exupéry
2  Harry Potter and the Philosopher's Stone             J. K. Rowling
3                  And Then There Were None           Agatha Christie
4                 Dream of the Red Chamber                Cao Xueqin
5                                The Hobbit          J. R. R. Tolkien
6      The Lion, the Witch and the Wardrobe              C. S. Lewis
7               She: A History of Adventure          H. Rider Haggard
8                         Vardi Wala Gunda        Ved Prakash Sharma
9                         The Da Vinci Code                Dan Brown

  Language  Year  Sales                         Genre
0  English  1859  200.0           Historical fiction
1   French  1943  200.0                      Novella
2  English  1997  120.0                      Fantasy
3  English  1939  100.0                      Mystery
4  Chinese  1791  100.0                 Family saga
5  English  1937  100.0                      Fantasy
6  English  1950   85.0  Fantasy, Children's fiction
7  English  1887   83.0                    Adventure
8    Hindi  1992   80.0                    Detective
9  English  2003   80.0            Mystery thriller
```

Displaying bar chart of top-selling books

Back to Top

In [17]:
```
top_selling_books = df.groupby('Book')['Sales'].sum().nlargest(10)
plt.figure(figsize=(10, 6))
top_selling_books.plot(kind='bar')
plt.xlabel('Book')
plt.ylabel('Total Sales (millions)')
plt.title('Top Books by Sales')
plt.xticks(rotation=45)
plt.show()
```

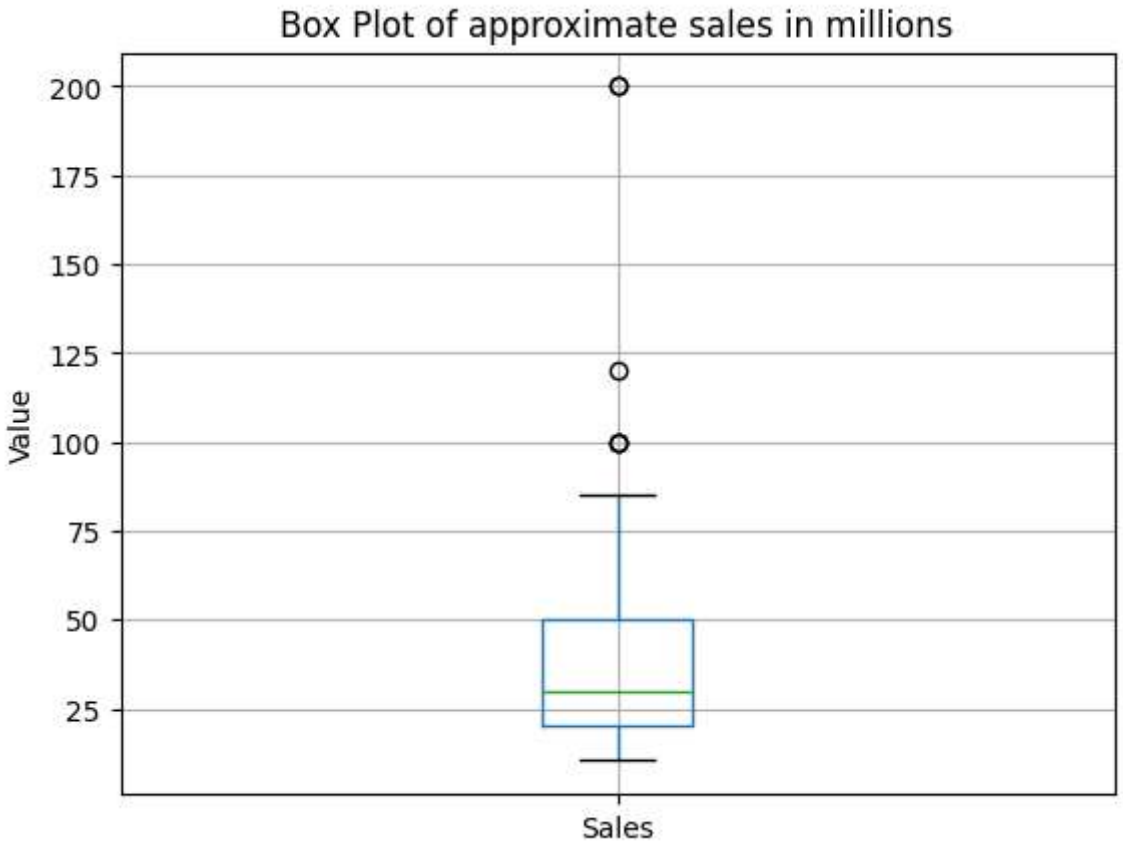


Visualizing distribution of sales

Back to Top

You can visualize data distribution and identify outliers. From the box plot, you can visualize that approximate sale has occurred between 22 to 50 million.

In [18]:
```
df.boxplot(column='Sales')
plt.title('Box Plot of approximate sales in millions')
plt.ylabel('Value')
plt.show()
```



Analyzing bivariate data

Back to Top

To perform bivariate data analysis, you can view the Scatter plot for numerical variables such as Year and Sales.

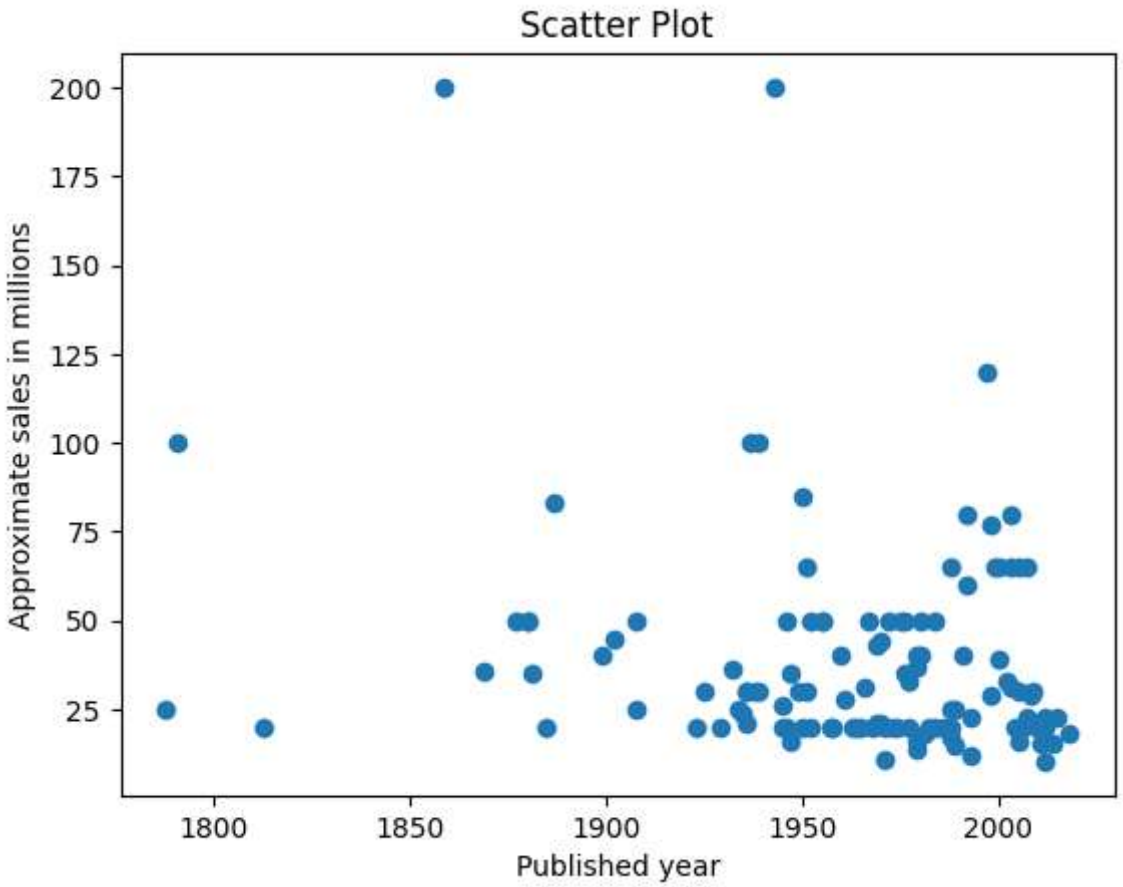
Identifying relationship between sales and year

Back to Top

In the following scatter plot, you can infer the following points:

- The sale is high between the years 1950 to 2000.
- The maximum sale value is between 25 to 75 million.

In [19]:
```
plt.scatter(df['Year'], df['Sales'])
plt.title('Scatter Plot')
plt.xlabel('Published year')
plt.ylabel('Approximate sales in millions')
plt.show()
```

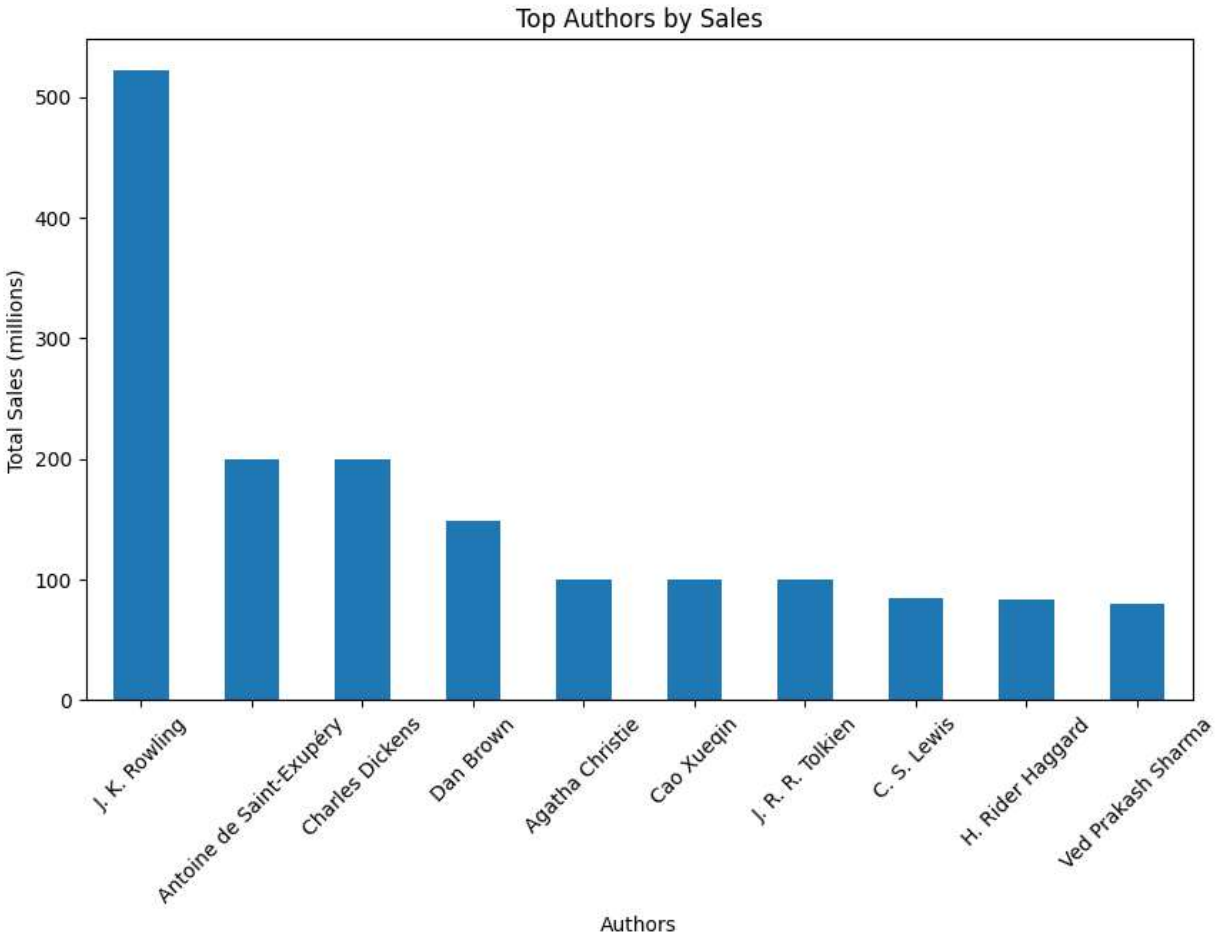


Identifying top authors by sales

Back to Top

Use a bar plot to show the relationship between the categorical variable (Author) and numerical variable (Sales). Identify the top authors by Sales.

In [20]:
```
top_authors = df.groupby('Authors')['Sales'].sum().nlargest(10)
plt.figure(figsize=(10, 6))
top_authors.plot(kind='bar')
plt.xlabel('Authors')
plt.ylabel('Total Sales (millions)')
plt.title('Top Authors by Sales')
plt.xticks(rotation=45)
plt.show()
```

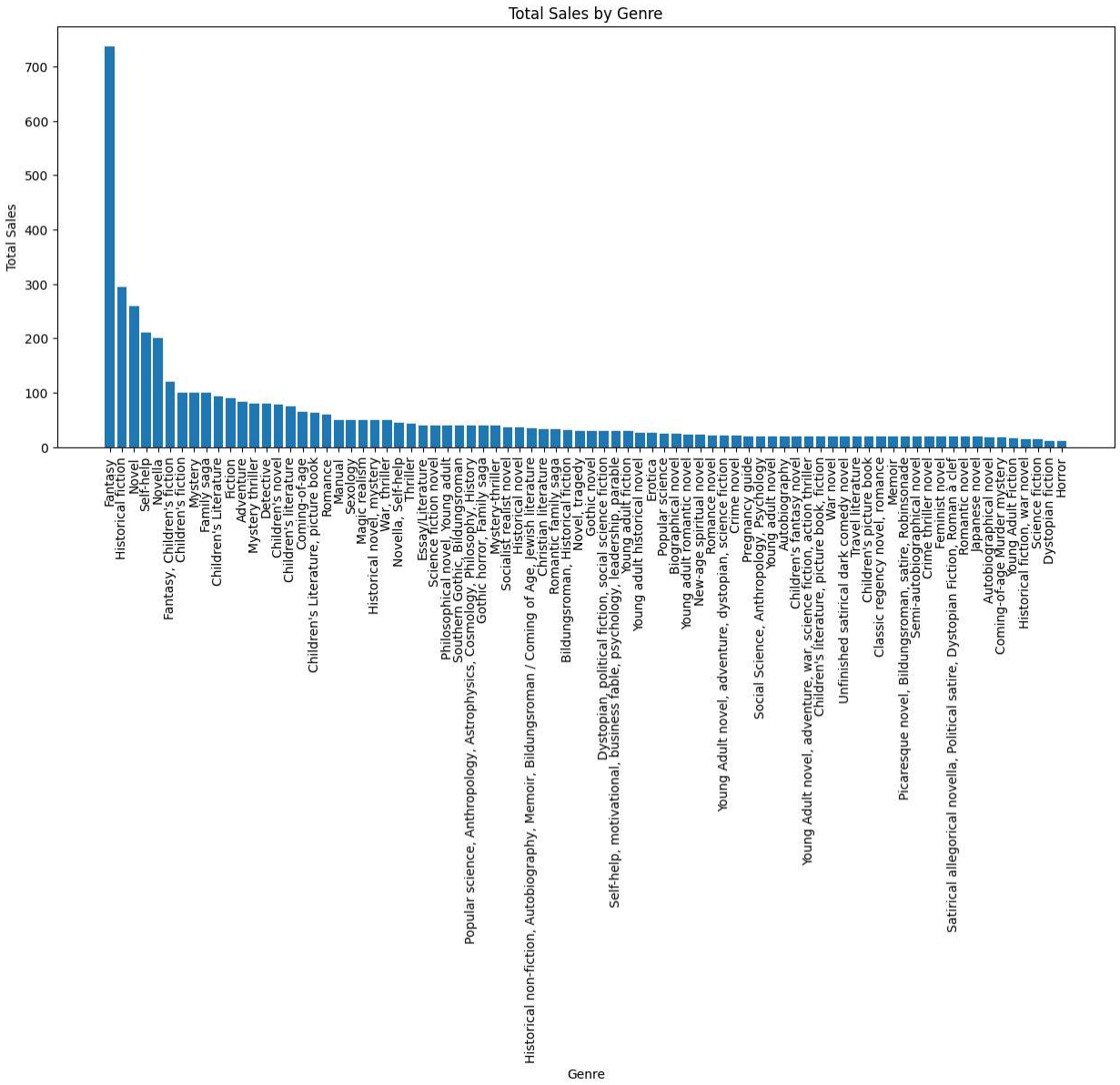


Identifying total sales by genre

Back to Top

Determine which genre performs better in terms of total sales.

In [21]:
```
genre_sales = df.groupby('Genre')['Sales'].sum().reset_index()
# Sort the genres by total sales in descending order
genre_sales = genre_sales.sort_values(by='Sales', ascending=False)

# Set the figure size for the bar chart
plt.figure(figsize=(15, 6))

# Create a bar chart to visualize genre vs. total sales
plt.bar(genre_sales['Genre'], genre_sales['Sales'])
plt.xlabel('Genre')
plt.ylabel('Total Sales')
plt.title('Total Sales by Genre')

# Rotate x-axis labels for better readability (optional)
plt.xticks(rotation=90)

# Show the plot
plt.show()
```



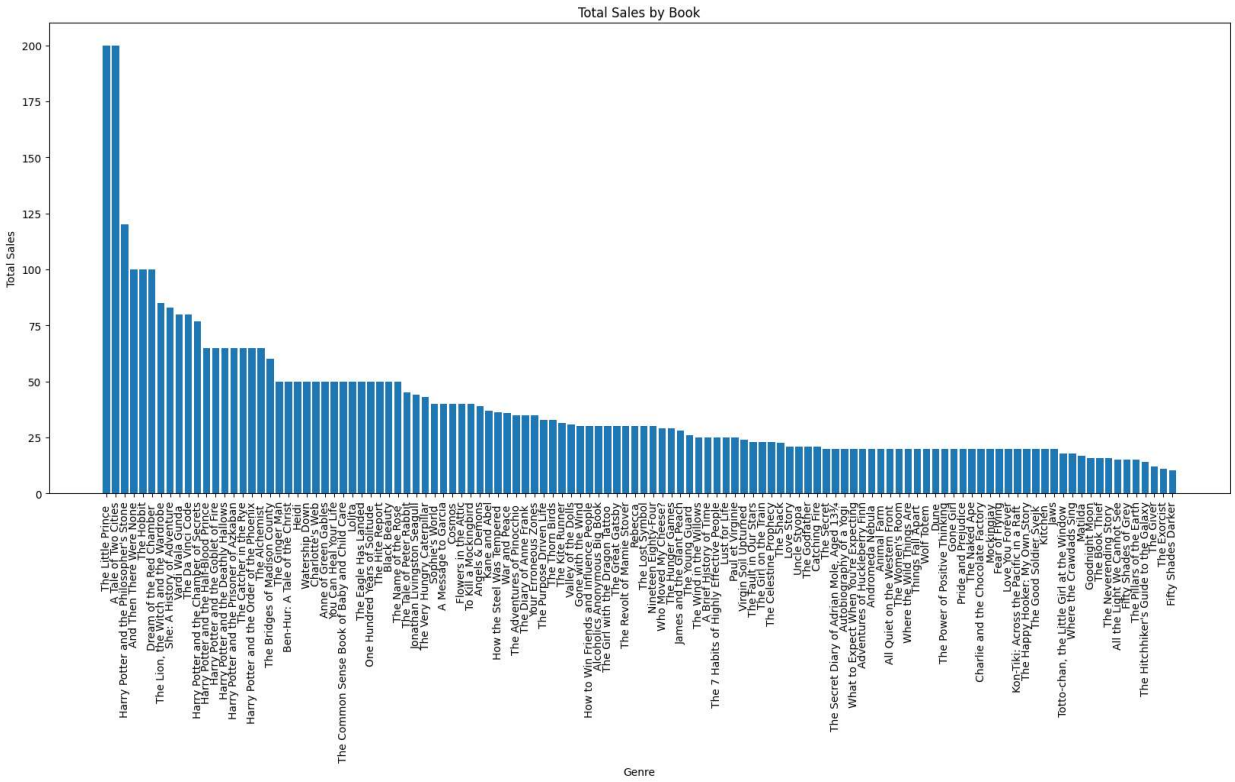
Identifying total sales by book

Back to Top

Identify the books that have the highest sales in proportion to the total sales.

In [22]:
```
genre_sales = df.groupby('Book')['Sales'].sum().reset_index()
# Sort the genres by total sales in descending order
genre_sales = genre_sales.sort_values(by='Sales', ascending=False)

# Set the figure size for the bar chart
plt.figure(figsize=(20, 8))

# Create a bar chart to visualize genre vs. total sales
plt.bar(genre_sales['Book'], genre_sales['Sales'])
plt.xlabel('Genre')
plt.ylabel('Total Sales')
plt.title('Total Sales by Book')
```

```
# Rotate x-axis labels for better readability (optional)
plt.xticks(rotation=90)

# Show the plot
plt.show()
```



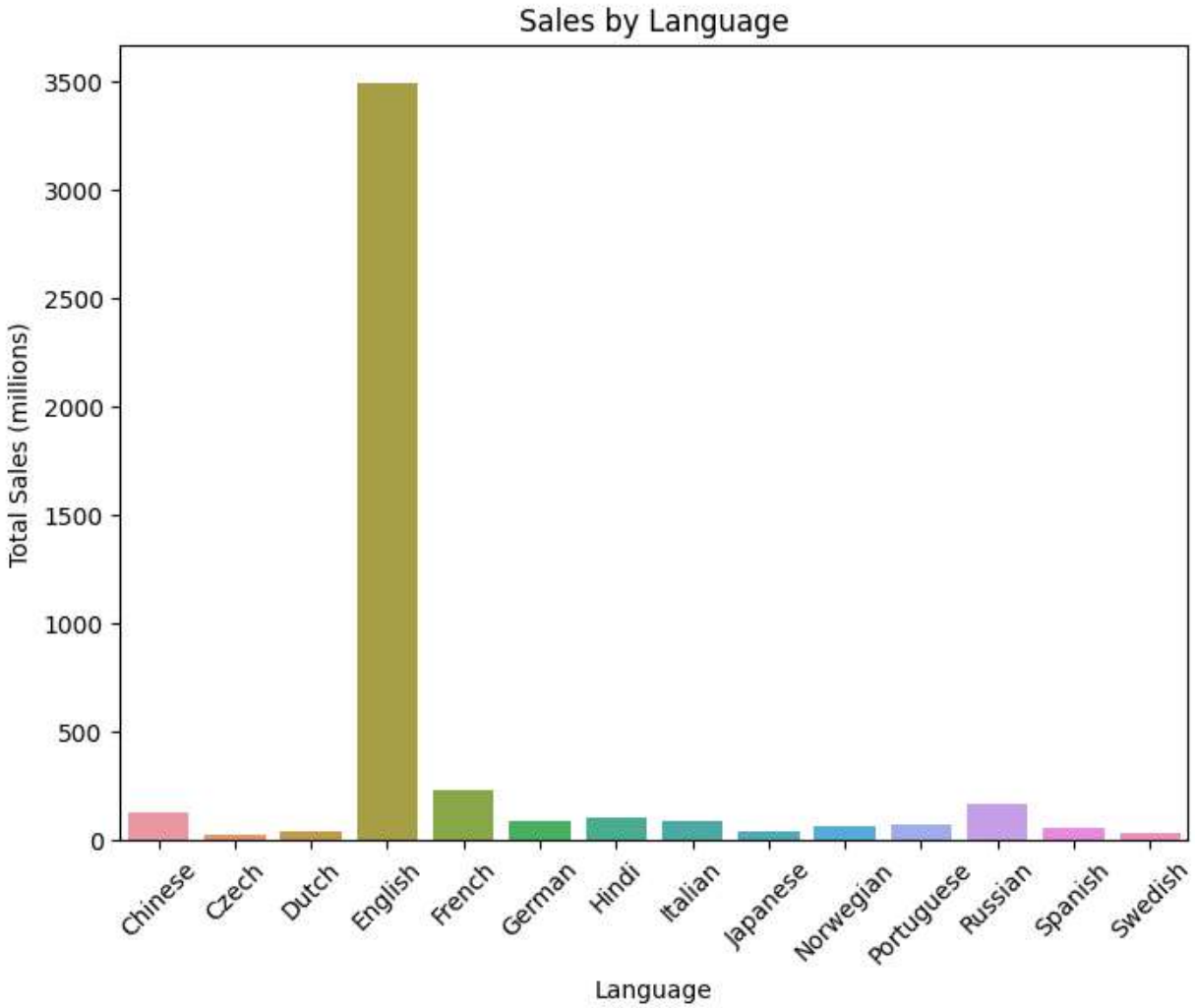
Determining language with the highest sales

Back to Top

Find out which language has the highest sales with the total sales. The bar plot shows that the books in the English language have the highest sales.

In [23]:
```
sales_by_language = df.groupby('Language')['Sales'].sum().reset_index()
plt.figure(figsize=(8, 6))
sns.barplot(data=sales_by_language, x='Language', y='Sales')
plt.xlabel('Language')
plt.ylabel('Total Sales (millions)')
plt.title('Sales by Language')
plt.xticks(rotation=45)
plt.show()
```

```
C:\Users\Kalpana\anaconda3\envs\py3115\Lib\site-packages\seaborn\_oldcore.py:1498: Fu
tureWarning:

is_categorical_dtype is deprecated and will be removed in a future version. Use isins
tance(dtype, CategoricalDtype) instead

C:\Users\Kalpana\anaconda3\envs\py3115\Lib\site-packages\seaborn\_oldcore.py:1498: Fu
tureWarning:

is_categorical_dtype is deprecated and will be removed in a future version. Use isins
tance(dtype, CategoricalDtype) instead

C:\Users\Kalpana\anaconda3\envs\py3115\Lib\site-packages\seaborn\_oldcore.py:1498: Fu
tureWarning:

is_categorical_dtype is deprecated and will be removed in a future version. Use isins
tance(dtype, CategoricalDtype) instead
```

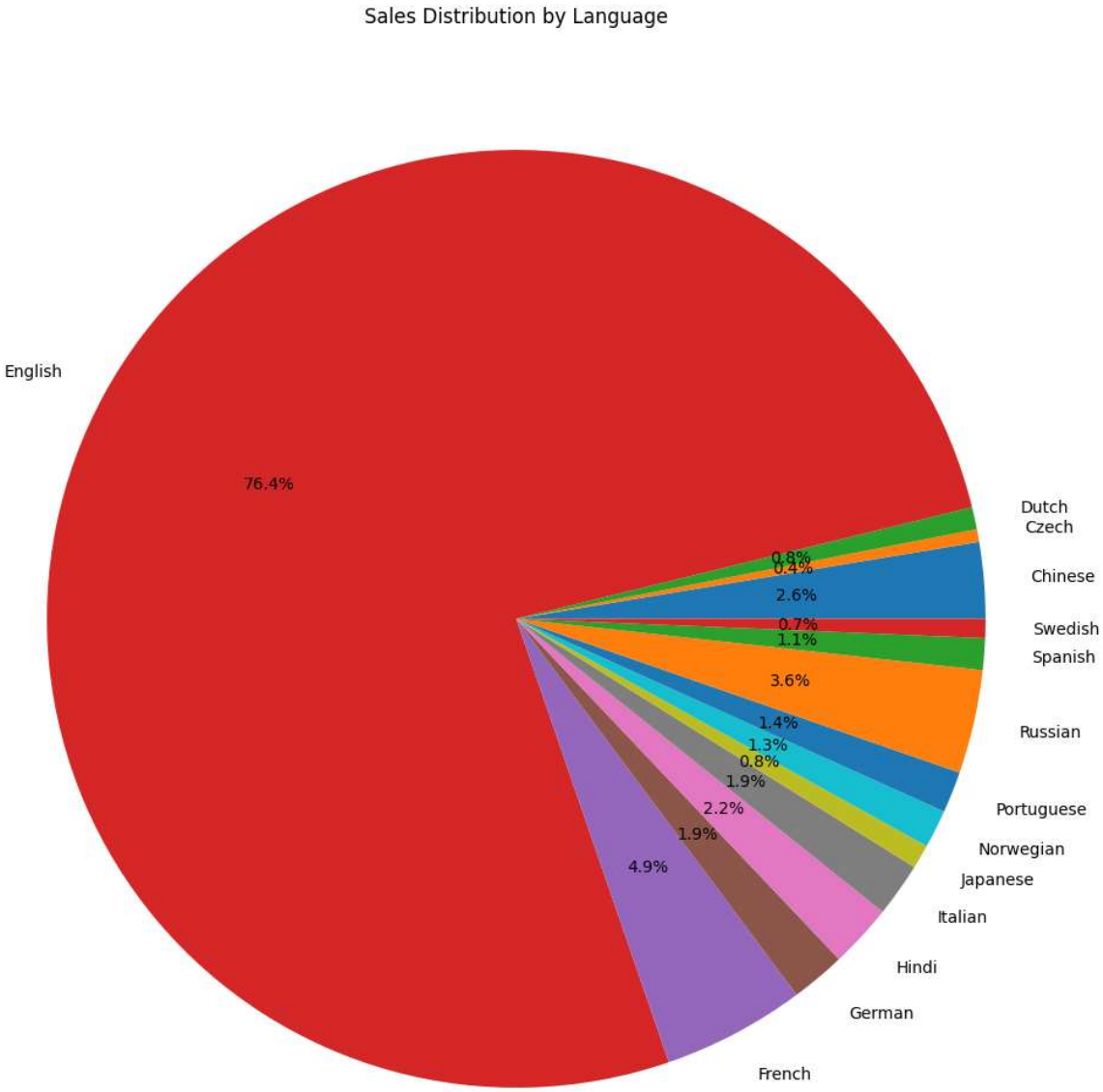


Determining sales distribution by language

Back to Top

The following pie chart shows the distribution of sales in percentage by language.

In [24]:
```
sales_by_language = df.groupby('Language')['Sales'].sum().reset_index()
plt.figure(figsize=(13, 13))
plt.pie(sales_by_language['Sales'], labels=sales_by_language['Language'], autopct='%1.
plt.title('Sales Distribution by Language')
plt.show()
```



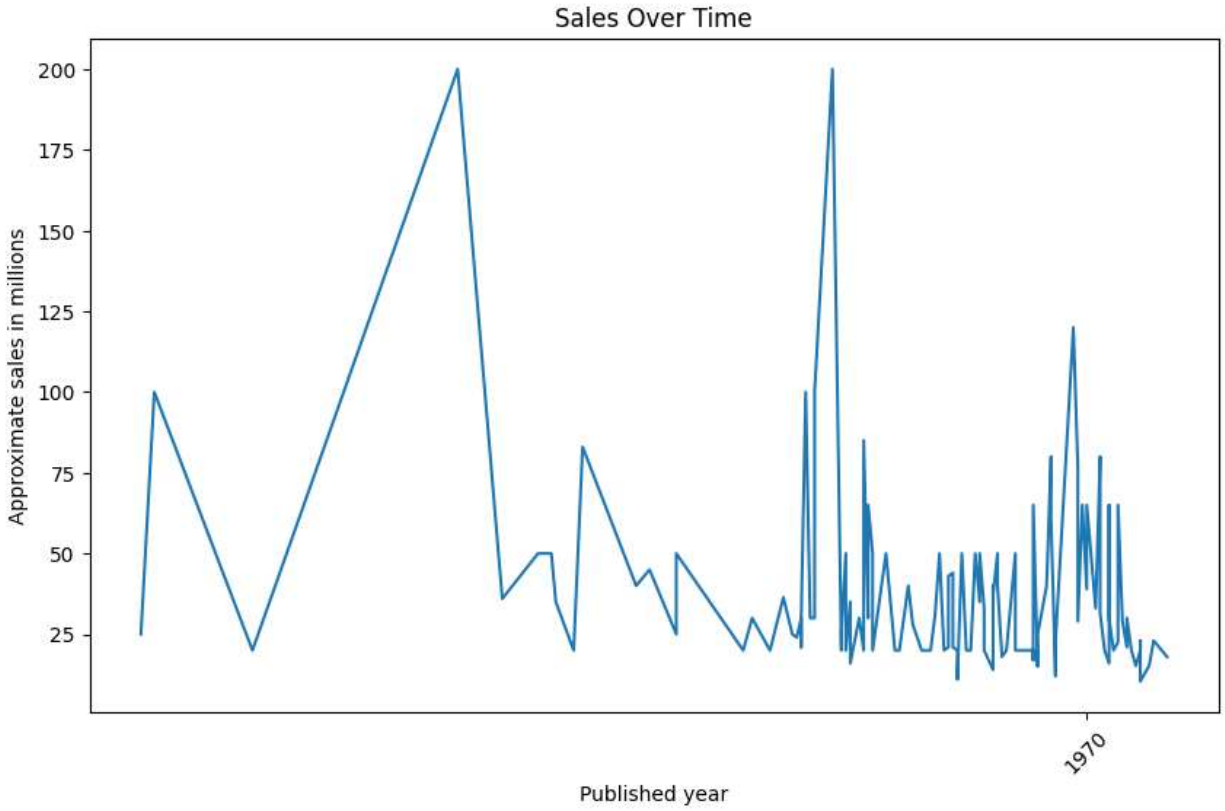
Identifying sales over time

Back to Top

To identify sales over time, use the 'Sales' continuous variable. The following code creates a simple line chart showing sales over time per year.

In [25]:
```
df['Year'] = pd.to_datetime(df['Year'])
df = df.sort_values('Year')
plt.figure(figsize=(10, 6))
plt.plot(df['Year'], df['Sales'])
```

```
plt.xlabel('Published year')
plt.ylabel('Approximate sales in millions')
plt.title('Sales Over Time')
plt.xticks(rotation=45)
plt.show()
```



Conclusion

Back to Top

From the exploratory data analysis conducted on the *Best-selling-books* dataset, the following conclusions emerge:

- The books categorized under the English language consistently achieve the highest sales figures.
- The approximate sales range falls between 22 and 50 million units.
- Sales remained consistently high during the year from 1950 to 2000.
- The peak sales value ranges from 25 to 75 million units.
- The Fantasy genre outperforms other genres in terms of sales.
- Books named **The Little Prince** and **A Tale of Two Cities** consistently attain the highest sales as a proportion of total sales.
- The Sales value fluctuates during the year 1970.

You can use these points to make informed decisions, optimize sales strategies, and achieve financial goals.