Introduction

- Dataset on Amazon's Top 50 bestselling books from 2009 to 2019. Contains 550 books, data has been categorized into fiction and non-fiction using Goodreads

Data source

- https://www.kaggle.com/sootersaalu/amazon-top-50-bestselling-books-2009-2019

Data description

- In this data there are total 550 rows and 7 columns.
- 'Name' - Book title.
- 'Author' - Person who wrote the book.
- 'User Rating' - Book rating out of 5.
- 'Reviews' - Number of people/readers reviewing the book via. rating.
- 'Price' - Cost of each book in US dollars.
- 'Year' - Book launch year.
- 'Genre' - Categories of books
- To analyse tha data using python libraries pandas and numpy and for data visualisation using plotly library.

Tasks performed in this analysis

- Distribution of genre in complete dataset.
- Visualize the distribution of genre per year.
- Top 10 most profitable authors
- Most profitable author per year of each genre
- The top 10 books with maximum number of reviews
- The books with the maximum number of reviews per year.
- Visualize the distribution of genre with respect to reviews.
- Top 10 books with the highest rating.
- Does a higher rating of the books affect its price?
- Is the mean price is changing over the years?
- Mean price per genre.

In [2]:

```
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
import warnings
warnings.filterwarnings('ignore')
```

In [3]:

```
data = pd.read_csv('bestsellers_with_categories.csv')
data.drop_duplicates(inplace=True)
data.head()
```

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

Distribution of genre in complete dataset.

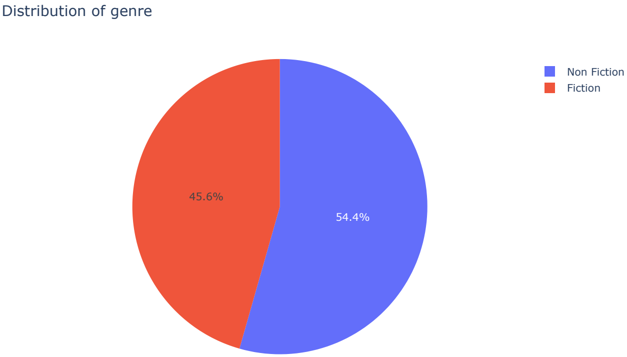
In [4]:

```
print(data.filter(['Name','Genre']).shape)
print(data.filter(['Name','Genre']).drop_duplicates().shape)
```

```
(550, 2)
(351, 2)
```

In [5]:

```python
category = data.filter(['Name','Genre']).drop_duplicates()
category.isnull().sum()
# category.drop_duplicates(inplace=True)
category = category.groupby(['Genre']).agg(count_genre=('Genre','count'))
category.reset_index(level=0,inplace=True)
fig = px.pie(category, values='count_genre', names='Genre',title='Distribution of genre')
fig.show()
```



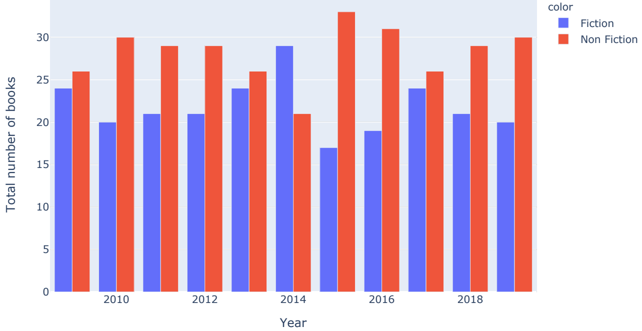
Observation -

- Pie chart shows the distribution between fiction and non-fiction books.
- 160 books genre is fiction and 191 genre is non-fiction which is slightly higher.
- Non-fiction books is 8.8 % higher than fiction books

Visualize the distribution of genre per year.

In [6]:

```python
year_genre = data.filter(['Name','Year','Genre'])
# year_genre.drop_duplicates(inplace=True)
year_genre = year_genre.groupby(['Year','Genre']).agg(count_book = ('Name','count'))
year_genre.reset_index(level=[0,1],inplace=True)
fig = px.bar(x=year_genre.Year, y=year_genre.count_book, color=year_genre['Genre'], barmode='group',
            labels={'x': 'Year', 'y': 'Total number of books'},title='Genre comparison per year')
fig.show()
```



Observation -

- X-axis represents year and Y-axis represents total number of books.
- We have more books from non-fiction genre over all years, except for 2014.
- We can observe a small rise in fiction book from 2010 to 2014.

Top 10 most profitable authors

Limitation

- Sales data is not avialble in a dataset.

Assumption

- If we assume all readers have provided a review then we can assume the number of reviews as number of books sold.

Hence, we can compute profit as the product of reviews and price for a given year.

In [7]:

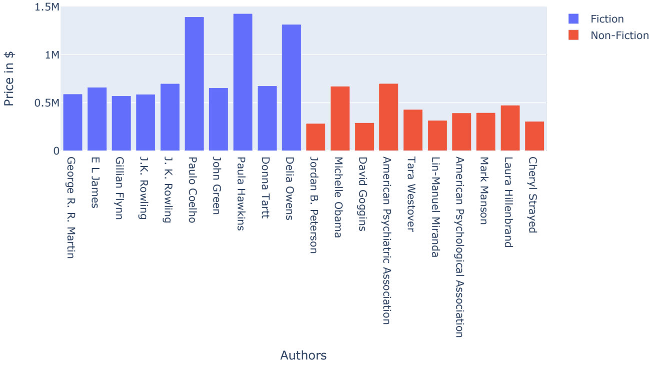
```
bestselling = data.filter(['Name','Author','Reviews','Price','Genre'])
bestselling.drop_duplicates(subset=['Name'],inplace=True)
bestselling['selling_price'] = bestselling['Reviews'] * bestselling['Price']

bestselling_fiction = bestselling[bestselling.Genre == 'Fiction']
bestselling_fiction['rank_fiction_price'] = bestselling_fiction.selling_price.rank(method='first',ascen
ding=False).astype(np.int32)
bestselling_fiction = bestselling_fiction[bestselling_fiction.rank_fiction_price < 11].copy()

bestselling_non_fiction = bestselling[bestselling.Genre == 'Non Fiction']
bestselling_non_fiction['rank_non_fiction_price'] = bestselling_non_fiction.selling_price.rank(method=
'first',ascending=False).astype(np.int32)
bestselling_non_fiction = bestselling_non_fiction[bestselling_non_fiction.rank_non_fiction_price < 11].
copy()

fig = go.Figure(data=[
    go.Bar(name='Fiction', x=bestselling_fiction.Author, y=bestselling_fiction.selling_price),
    go.Bar(name='Non-Fiction', x=bestselling_non_fiction.Author, y=bestselling_non_fiction.selling_pric
e)
])
fig.update_layout(barmode='group',title='Top 10 profitable author per genre')
fig.update_xaxes(title_text="Authors")
fig.update_yaxes(title_text="Price in $")
fig.show()
```

Top 10 profitable author per genre



Observation

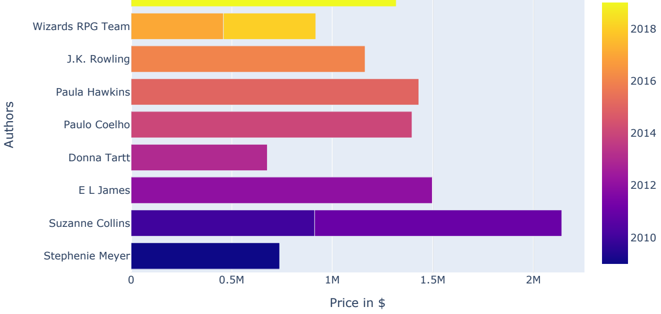
- Bar graph represent the best-seller from 2009 to 2019.
- X-axis represent authors name.
- Y-axis represent selling price.
- Paula hawkins is the best-seller in fiction followed by Paulo Coelho.
- American psychiatric association is the best-seller in non-fiction followed by Michelle Obama.

Most profitable author per year of each genre

In [8]:

```
fiction = data.filter(['Name','Author','Year','Reviews','Price','Genre'])
# df.drop_duplicates(subset=['Name'],inplace=True)
fiction['fiction_selling_price'] = fiction['Reviews']* fiction['Price']
fiction = pd.DataFrame(fiction.groupby(['Year','Author','Genre']).agg({'fiction_selling_price':'sum'}))
fiction.reset_index(level=[0,1,2],inplace=True)
fiction = fiction[fiction.Genre == 'Fiction']
fiction['rank_price'] = fiction.groupby(['Year'])['fiction_selling_price'].rank(method='first',ascendin
g=False).astype(np.int32)
fiction = fiction[fiction.rank_price == 1]

non_fiction = data.filter(['Name','Author','Year','Reviews','Price','Genre'])
non_fiction['non_fiction_selling_price'] = non_fiction['Reviews']* non_fiction['Price']
non_fiction = pd.DataFrame(non_fiction.groupby(['Year','Author','Genre']).agg({'non_fiction_selling_pri
ce':'sum'}))
non_fiction.reset_index(level=[0,1,2],inplace=True)
non_fiction = non_fiction[non_fiction.Genre == 'Non Fiction']
non_fiction['rank_price'] = non_fiction.groupby(['Year'])['non_fiction_selling_price'].rank(method='fir
st',ascending=False).astype(np.int32)
non_fiction = non_fiction[non_fiction.rank_price == 1]
fig = px.bar(y=fiction.Author, x=fiction.fiction_selling_price,color=fiction['Year'],title='Most profit
able author in fiction',
            labels={'x': 'Price in $', 'y': 'Authors','color':'Year'})
fig.show()
```

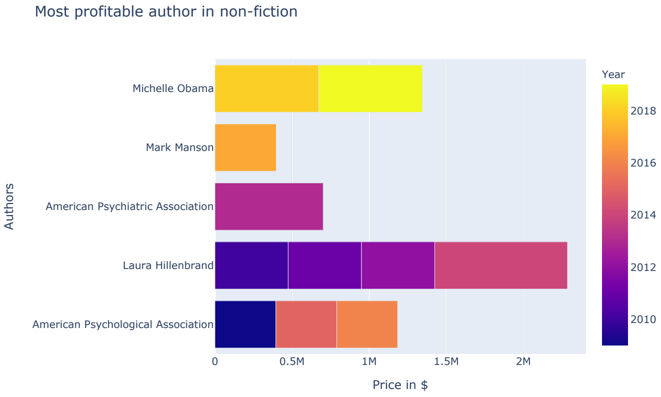


Observation

- X-axis represent profit in dollars.
- Y-axis represent authors.
- Wixards RPG Team profitale in 2017 and 2018.
- Suzanne collins profitable in 2010 and 2011.

In [9]:

```
fig = px.bar(y=non_fiction.Author, x=non_fiction.non_fiction_selling_price,color=non_fiction.Year,title='Most profitable author in non-fiction',
             labels={'x': 'Price in $', 'y': 'Authors','color':'Year'})
fig.show()
```



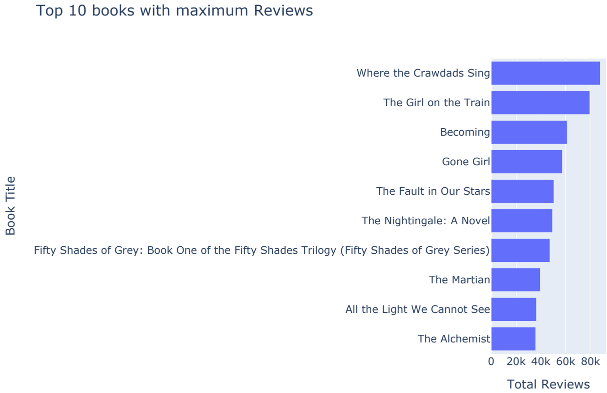
Observation

- X-axis represent profit in dollars.
- Y-axis represet authors.
- American Psychological Association was the profitable author in 2009, 2015 and 2016
- Laura Hillenbrand was the profitable author in 2010, 2011, 2012 and 2014.
- Michelle Obama was the profitable author in 2018 and 2019.

The top 10 books with maximum number of reviews

In [10]:

```
most_rvs = data.filter(['Name','Reviews'])
most_rvs.drop_duplicates(subset=['Name'],inplace=True)
most_rvs.isnull().sum()
most_rvs['rank_reviews'] = most_rvs.Reviews.rank(method='first',ascending=False).astype(np.int32)
most_rvs = most_rvs[most_rvs.rank_reviews < 11]
most_rvs.sort_values(by=['rank_reviews'],ascending=False,inplace=True)
fig = px.bar(y=most_rvs.Name, x=most_rvs.Reviews,title='Top 10 books with maximum Reviews',labels={'x': 'Total Reviews', 'y': 'Book Title'})
fig.show()
```



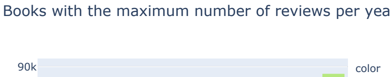
Observation -

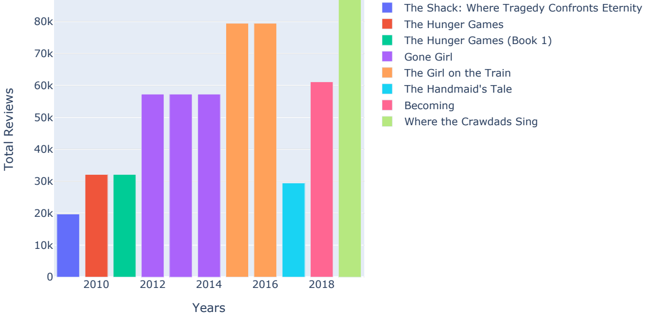
- X-axis represent the total number of reviews.
- Y-axis represent the books name.
- The number of reviews ranges between 37 and 87,841.
- By far the most reviews have been given to 'Where the Crawdads Sing' by Delia Owens with a user rating of 4.8 and 'The Girl on the Train' by Paula Hawkings with a user rating of 4.1.

The books with the maximum number of reviews per year.

In [11]:

```
most_rvs_year = data.filter(['Name','Year','Reviews'])
# most_rvs_year.drop_duplicates(subset=['Name'],inplace=True)
most_rvs_year = pd.DataFrame(most_rvs_year.groupby(['Year','Name']).Reviews.max())
most_rvs_year.reset_index(level=[0,1],inplace=True)
most_rvs_year['rank_Reviews'] = most_rvs_year.groupby(['Year']).Reviews.rank(method='first',ascending=False).astype(np.int32)
most_rvs_year = most_rvs_year[most_rvs_year.rank_Reviews == 1]
most_rvs_year
fig = px.bar(x=most_rvs_year.Year, y=most_rvs_year.Reviews,color=most_rvs_year.Name,title='Books with the maximum number of reviews per year',
             labels={'x': 'Years', 'y': 'Total Reviews'})
fig.show()
```



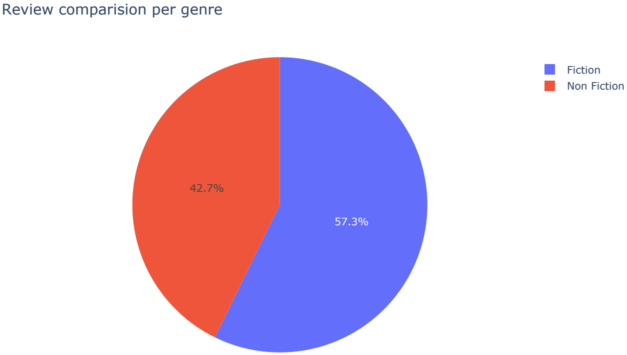


Observation

- X-axis represent years.
- Y-axis represent Reviews.
- "Gone Girl" recorded the maximum number of reviews in three consecutive years(2012-2014).
- "The Girl on the Train" recorded the maximum number of reviews in two consecutive years(2015-2016).
- "Where the Crawdads Sing" recorded highest number of reviews in 2019.

Visualize the distribution of genre with respect to reviews.

In [12]:
```
genre_diff = data.filter(['Genre','Reviews'])
genre_diff = genre_diff.groupby(['Genre']).agg({'Reviews':'sum'})
genre_diff.reset_index(level=0,inplace=True)
fig = px.pie(genre_diff, values='Reviews', names='Genre',title='Review comparision per genre')
fig.show()
```



Observation -

- Total number of reviews 6,574,305.

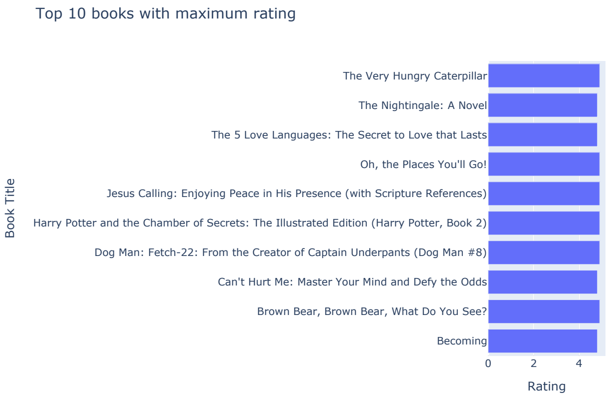
- Fiction reviews 3,764,110.
- Non Fiction reviews 2,810,195.

Top 10 books with the highest rating.

In [13]:

```
max_rating = data.filter(['Name','Author','User Rating','Reviews']).drop_duplicates()
max_rating['rank_rating'] = max_rating['User Rating'].rank(method='first',ascending=False).astype(np.in
t32)
# max_rating = max_rating[max_rating.rank_rating < 11]
max_rating['rank_review'] = max_rating['Reviews'].rank(method='first',ascending=False).astype(np.int32)
# max_rating = max_rating[max_rating.rank_review < 11]
print(max_rating.rank_rating.min(), max_rating.rank_rating.max())
print(max_rating.rank_review.min(), max_rating.rank_review.max())
max_rating['total_rank'] = (max_rating['rank_rating'] + max_rating['rank_review'])/2
max_rating['rank_total'] = max_rating['total_rank'].rank(method='first',ascending=True).astype(np.int32
)
max_rating = max_rating[max_rating.rank_total < 11]
fig = px.bar(y=max_rating.Name, x=max_rating['User Rating'],labels={'x':'Rating','y':'Book Title'},titl
e='Top 10 books with maximum rating')
fig.show()
```

```
1 352
1 352
```



Observation

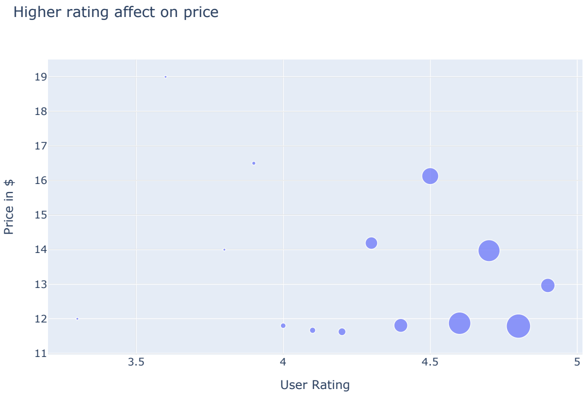
- The bar graph represent top 10 higher rating books.
- X-axis represent rating.
- Y-axis represent book title.
- 6 books have rating of 4.9.
- 4 books have rating of 4.8.

Does a higher rating of the books affect its price?

In [14]:

```python
rating_price = data.filter(['Author','Name','User Rating','Price','Reviews']).drop_duplicates()
print(rating_price.isnull().sum().sum())
rating_price = pd.DataFrame(rating_price.groupby(['User Rating']).agg({'Price': 'mean', 'Name': 'nunique'}))
rating_price.reset_index(level=0,inplace=True)
fig = px.scatter(rating_price, x="User Rating", y="Price", title='Higher rating affect on price', size='Name')
fig.update_xaxes(title_text="User Rating")
fig.update_yaxes(title_text="Price in $")
fig.show()
```

```
0
```



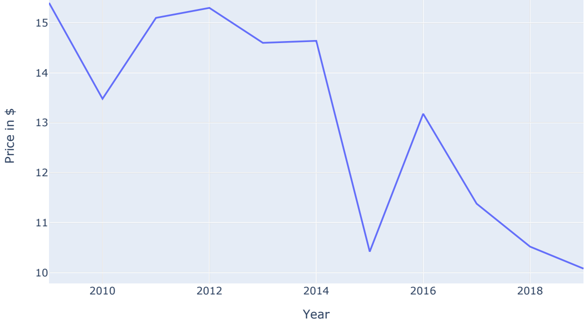
Observation -

- X-axis represent user rating.
- Y-axis represent price in dollars.
- There is no clear relationship between user rating and price.
- We have seen that the number of books are more in higher rating.

Is the mean price is changing over the years?

In [15]:

```python
price_year = data.filter(['Price','Year'])
price_year = pd.DataFrame(price_year.groupby(['Year']).Price.mean())
price_year.reset_index(level=0,inplace=True)
fig = px.line(price_year, x="Year", y="Price", title='Mean price change over the years')
fig.update_xaxes(title_text="Year")
fig.update_yaxes(title_text="Price in $")
fig.show()
```

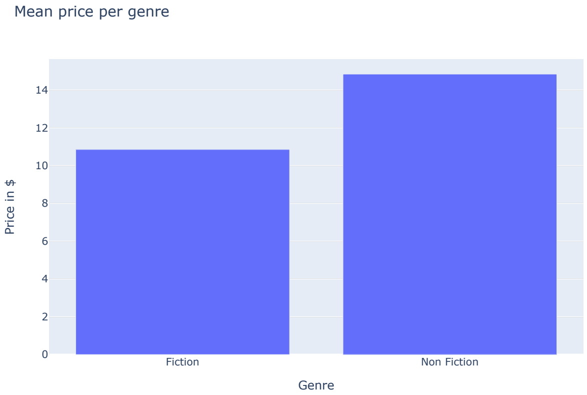


Observation

- X-axis represent years.
- Y-axis represent price in dollars.
- Sudden fall in price between 2014-2015 but we don't know the reason because unavailability of data.

Mean price per genre.

In [16]:
```
genre_price = data.filter(['Price','Genre'])
genre_price = pd.DataFrame(genre_price.groupby(['Genre']).Price.mean())
genre_price.reset_index(level=0,inplace=True)
fig = px.bar(genre_price, x='Genre', y='Price',title='Mean price per genre')
fig.update_xaxes(title_text="Genre")
fig.update_yaxes(title_text="Price in $")
fig.show()
```



Observation

- X-axis represent genre.
- Y-axis represent price in dollars.
- Mean price of non fiction is 14.84.
- Mean price of fiction is 10.85