Amazon\_best\_selling\_books\_with\_R

Malay Raj

2023-05-30

## AMAZON BEST SELLING BOOKS FROM 2009 TO 2019 ANALYSIS:

Amazon Top 50 Best Selling Books (made available by Souter Saalu under a CC0 license): Data about Amazon’s bestselling books from 2009 to 2019, categorized into fiction and nonfiction.

## Loading Required Packages.

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.2.3

## Warning: package 'ggplot2' was built under R version 4.2.3

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.0 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.1.8  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become errors

library(lubridate)  
library(ggplot2)  
library(plyr)

## ------------------------------------------------------------------------------  
## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)  
## ------------------------------------------------------------------------------  
##   
## Attaching package: 'plyr'  
##   
## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize  
##   
## The following object is masked from 'package:purrr':  
##   
## compact

library(skimr)

## Warning: package 'skimr' was built under R version 4.2.3

## Displaying my current working directory.

getwd()

## [1] "C:/Amazon top 50 best selling books analysis"

## Collecting the data.

Books = read.csv("C:/Amazon top 50 best selling books analysis/bestsellers with categories.csv")

## Displaying the first six rows of the dataset.

head(Books)

## Name  
## 1 10-Day Green Smoothie Cleanse  
## 2 11/22/63: A Novel  
## 3 12 Rules for Life: An Antidote to Chaos  
## 4 1984 (Signet Classics)  
## 5 5,000 Awesome Facts (About Everything!) (National Geographic Kids)  
## 6 A Dance with Dragons (A Song of Ice and Fire)  
## Author User.Rating Reviews Price Year Genre  
## 1 JJ Smith 4.7 17350 8 2016 Non Fiction  
## 2 Stephen King 4.6 2052 22 2011 Fiction  
## 3 Jordan B. Peterson 4.7 18979 15 2018 Non Fiction  
## 4 George Orwell 4.7 21424 6 2017 Fiction  
## 5 National Geographic Kids 4.8 7665 12 2019 Non Fiction  
## 6 George R. R. Martin 4.4 12643 11 2011 Fiction

## Checking the number of rows and dimension of the dataset.

nrow(Books)

## [1] 550

dim(Books)

## [1] 550 7

## Renaming the column User.Rating to Ratings.

Books=plyr::rename(Books, replace=c( "User.Rating" = "Ratings"))

## Viewing the new column names.

colnames(Books)

## [1] "Name" "Author" "Ratings" "Reviews" "Price" "Year" "Genre"

## Inspecting the dataframe and look for inconguencies.

str(Books)

## 'data.frame': 550 obs. of 7 variables:  
## $ Name : chr "10-Day Green Smoothie Cleanse" "11/22/63: A Novel" "12 Rules for Life: An Antidote to Chaos" "1984 (Signet Classics)" ...  
## $ Author : chr "JJ Smith" "Stephen King" "Jordan B. Peterson" "George Orwell" ...  
## $ Ratings: num 4.7 4.6 4.7 4.7 4.8 4.4 4.7 4.7 4.7 4.6 ...  
## $ Reviews: int 17350 2052 18979 21424 7665 12643 19735 19699 5983 23848 ...  
## $ Price : int 8 22 15 6 12 11 30 15 3 8 ...  
## $ Year : int 2016 2011 2018 2017 2019 2011 2014 2017 2018 2016 ...  
## $ Genre : chr "Non Fiction" "Fiction" "Non Fiction" "Fiction" ...

## Checking for null values in our dataset.

sum(is.null(Books))

## [1] 0

## Checking for any duplicate values in our dataset.

anyDuplicated(Books)

## [1] 0

## Seeing how many observations fall under each of these columns.

table(Books$Genre)

##   
## Fiction Non Fiction   
## 240 310

table(Books$Year)

##   
## 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019   
## 50 50 50 50 50 50 50 50 50 50 50

table(Books$Ratings)

##   
## 3.3 3.6 3.8 3.9 4 4.1 4.2 4.3 4.4 4.5 4.6 4.7 4.8 4.9   
## 1 1 2 3 14 6 8 25 38 60 105 108 127 52

## CONDUCTNG DESCRIPTIVE ANALYSIS:

Descriptive analysis on few columns.

summary(Books$Ratings)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 3.300 4.500 4.700 4.618 4.800 4.900

summary(Books$Reviews)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 37 4058 8580 11953 17253 87841

summary(Books$Price)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 7.0 11.0 13.1 16.0 105.0

Comparing Ratings, Reviews, Price with different Genre.

aggregate(Books$Ratings ~ Books$Genre, FUN = mean)

## Books$Genre Books$Ratings  
## 1 Fiction 4.648333  
## 2 Non Fiction 4.595161

aggregate(Books$Ratings ~ Books$Genre, FUN = median)

## Books$Genre Books$Ratings  
## 1 Fiction 4.7  
## 2 Non Fiction 4.6

aggregate(Books$Ratings ~ Books$Genre, FUN = max)

## Books$Genre Books$Ratings  
## 1 Fiction 4.9  
## 2 Non Fiction 4.9

aggregate(Books$Ratings ~ Books$Genre, FUN = min)

## Books$Genre Books$Ratings  
## 1 Fiction 3.3  
## 2 Non Fiction 4.0

aggregate(Books$Price ~ Books$Genre, FUN = mean)

## Books$Genre Books$Price  
## 1 Fiction 10.85000  
## 2 Non Fiction 14.84194

aggregate(Books$Price ~ Books$Genre, FUN = median)

## Books$Genre Books$Price  
## 1 Fiction 9  
## 2 Non Fiction 12

aggregate(Books$Price ~ Books$Genre, FUN = max)

## Books$Genre Books$Price  
## 1 Fiction 82  
## 2 Non Fiction 105

aggregate(Books$Price ~ Books$Genre, FUN = min)

## Books$Genre Books$Price  
## 1 Fiction 0  
## 2 Non Fiction 0

aggregate(Books$Reviews ~ Books$Genre, FUN = mean)

## Books$Genre Books$Reviews  
## 1 Fiction 15683.792  
## 2 Non Fiction 9065.145

aggregate(Books$Reviews ~ Books$Genre, FUN = median)

## Books$Genre Books$Reviews  
## 1 Fiction 10922  
## 2 Non Fiction 6346

aggregate(Books$Reviews ~ Books$Genre, FUN = max)

## Books$Genre Books$Reviews  
## 1 Fiction 87841  
## 2 Non Fiction 61133

aggregate(Books$Reviews ~ Books$Genre, FUN = min)

## Books$Genre Books$Reviews  
## 1 Fiction 548  
## 2 Non Fiction 37

## SUMMARY STATISTICS.

skimr::skim\_without\_charts(Books)

Data summary

|  |  |
| --- | --- |
| Name | Books |
| Number of rows | 550 |
| Number of columns | 7 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 3 |
| numeric | 4 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | 0 | 1 | 4 | 121 | 0 | 351 | 0 |
| Author | 0 | 1 | 2 | 34 | 0 | 248 | 0 |
| Genre | 0 | 1 | 7 | 11 | 0 | 2 | 0 |

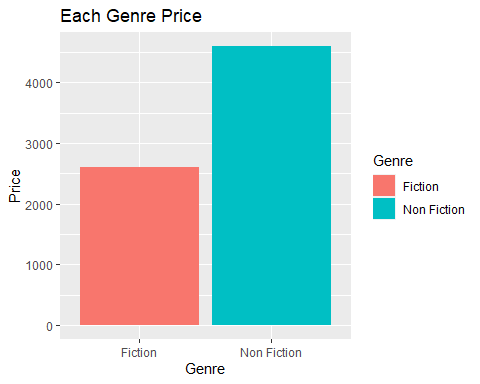
**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ratings | 0 | 1 | 4.62 | 0.23 | 3.3 | 4.5 | 4.7 | 4.80 | 4.9 |
| Reviews | 0 | 1 | 11953.28 | 11731.13 | 37.0 | 4058.0 | 8580.0 | 17253.25 | 87841.0 |
| Price | 0 | 1 | 13.10 | 10.84 | 0.0 | 7.0 | 11.0 | 16.00 | 105.0 |
| Year | 0 | 1 | 2014.00 | 3.17 | 2009.0 | 2011.0 | 2014.0 | 2017.00 | 2019.0 |

## DATA VISUALIZATION:

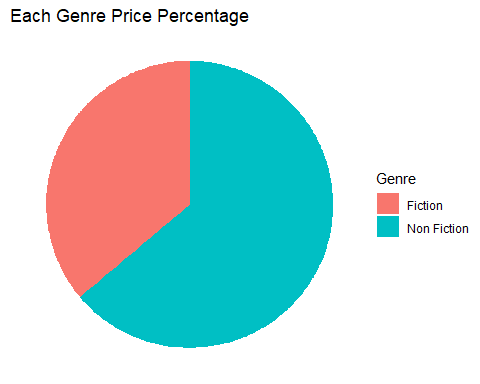
Comparing each genre price values.

ggplot(Books) +   
 geom\_col(mapping = aes(x=Genre, y=Price, fill=Genre)) +   
 labs(title="Each Genre Price")



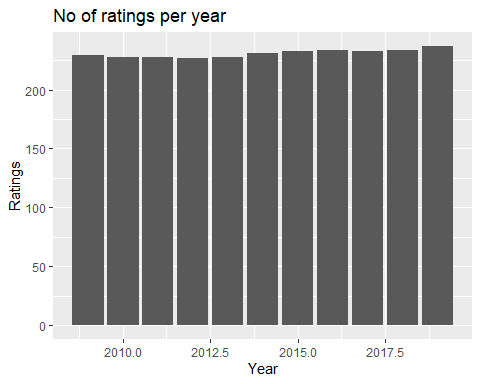
Comparing each genre price distribution.

ggplot(Books, aes(x="", y=Price, fill=Genre)) +  
 ggtitle("Each Genre Price Percentage") +  
 geom\_bar(width = 1, stat = "identity") +  
 coord\_polar("y", start=0) +  
 theme\_void()



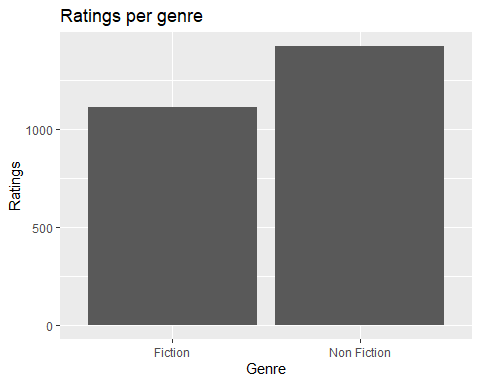
Checking the number of ratings per year.

ggplot(Books) +   
 geom\_col(mapping = aes(x=Year, y=Ratings)) +  
 labs(title="No of ratings per year")



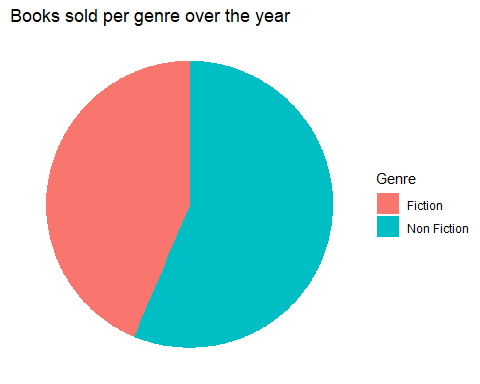
Comparing the rating per genre.

ggplot(Books) +   
 geom\_col(mapping = aes(x=Genre, y=Ratings)) +  
 labs(title="Ratings per genre")



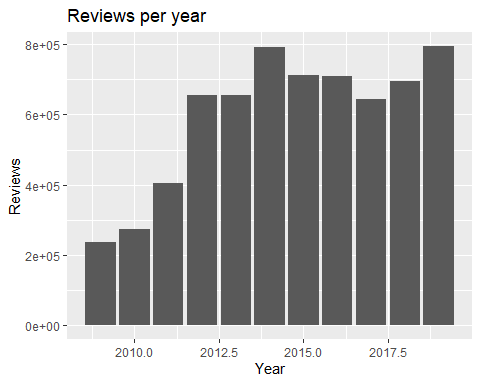
Checking number of books sold per genre over 10 years.

ggplot(Books, aes(x="", y=Year, fill=Genre)) +  
 ggtitle("Books sold per genre over the year") +  
 geom\_bar(width = 1, stat = "identity") +  
 coord\_polar("y", start=0) +  
 theme\_void()



Checking the number of reviews got per year.

ggplot(Books) +   
 geom\_col(mapping = aes(x=Year, y=Reviews)) +  
 labs(title="Reviews per year")



Checking the top 5 books over the 10 years period reviews vise.

Books %>%  
 arrange(-Reviews) %>%  
 slice\_head(n=10) %>%  
 ggplot(aes(reorder(Name, Reviews), Reviews)) +  
 geom\_point(colour = "blue", size = 3) +  
 theme(text=element\_text(size=10)) +   
 coord\_flip() +  
 labs(title = "Top 5 best selling books Review vise", x = "")

