steamAnalysis

July 12, 2023

1 STEAM DATA ANALYSIS

[2]:
```
import numpy as np
import pandas as pd
import sqlite3
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
```

1.1 Connect to Database and pull data into the dataframe

[3]:
```
conn = sqlite3.connect("SteamDB.sqlite")
cur = conn.cursor()
```

[4]:
```
sqlQuery = """SELECT name, popularTags, price, features, lanInterface,␣
  ↪LanAudio, lanSubtitle, lanAllSupported, genre, developer, publisher,␣
  ↪releaseDate, minSysReq, recSysReq, reviewTotal, reviewPositive,␣
  ↪reviewNegative, reviewPercentage
from GameDetails
JOIN Games ON GameDetails.gameId=Games.id
order by GameDetails.gameId"""
```

[5]:
```
df = pd.read_sql_query(sqlQuery, conn, index_col="name",␣
  ↪parse_dates="releaseDate")
conn.close()
```

1.2 Log Count

[6]:
```
df.shape
```

[6]:
```
(26457, 17)
```

[7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 26457 entries, Counter-Strike: Global Offensive to DEKONSTRUKT
Data columns (total 17 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   popularTags       26457 non-null  object
 1   price             26435 non-null  object
 2   features          26457 non-null  object
 3   lanInterface      26457 non-null  object
 4   LanAudio          26457 non-null  object
 5   lanSubtitle       26457 non-null  object
 6   lanAllSupported   26457 non-null  int64
 7   genre             26457 non-null  object
 8   developer         26457 non-null  object
 9   publisher         26457 non-null  object
 10  releaseDate       26428 non-null  datetime64[ns]
 11  minSysReq         26457 non-null  object
 12  recSysReq         26457 non-null  object
 13  reviewTotal       26457 non-null  int64
 14  reviewPositive    26457 non-null  int64
 15  reviewNegative    26457 non-null  int64
 16  reviewPercentage  21751 non-null  float64
dtypes: datetime64[ns](1), float64(1), int64(4), object(11)
memory usage: 3.6+ MB
```

1.3 Data Summary

[8]:

```
df.head()
```

[8]:

```
popularTags  \
name
Counter-Strike: Global Offensive
FPS,Shooter,Multiplayer,Competitive,Action,Tea…
ELDEN RING                          Souls-like,Dark Fantasy,RPG,Open
World,Difficu…
Red Dead Redemption 2               Open World,Story
Rich,Western,Adventure,Action…
Forza Horizon 5                     Racing,Open
World,Driving,Multiplayer,Automobi…
Rust                                Survival,Crafting,Multiplayer,Open World,Open
…

                                             price  \
name
Counter-Strike: Global Offensive  Free to Play
ELDEN RING                           699,00 TL
Red Dead Redemption 2              1.150,00 TL
Forza Horizon 5                      599,00 TL
Rust                                 308,00 TL
```

```
features  \
name
Counter-Strike: Global Offensive  Steam Achievements,Full controller
support,Ste…
ELDEN RING                        Single-player,Online PvP,Online Co-op,Steam
Ac…
Red Dead Redemption 2             Single-player,Online PvP,Online Co-op,Steam
Ac…
Forza Horizon 5                   Single-player,Online PvP,Online Co-op,Cross-
Pl…
Rust                              MMO,Online PvP,Online Co-op,Cross-Platform
Mul…

lanInterface  \
name
Counter-Strike: Global Offensive
English,Czech,Danish,Dutch,Finnish,French,Germ…
ELDEN RING                        English,French,Italian,German,Spanish -
Spain,…
Red Dead Redemption 2             English,French,Italian,German,Spanish -
Spain,…
Forza Horizon 5                   English,French,Italian,German,Spanish -
Spain,…
Rust                              English,French,Italian,German,Spanish -
Spain,…

LanAudio  \
name
Counter-Strike: Global Offensive
English
ELDEN RING
English
Red Dead Redemption 2
English
Forza Horizon 5                   English,French,German,Portuguese -
Brazil,Span…
Rust                              English,French,Italian,German,Spanish -
Spain,…

lanSubtitle  \
name
Counter-Strike: Global Offensive
ELDEN RING                        English,French,Italian,German,Spanish -
Spain,…
Red Dead Redemption 2             English,French,Italian,German,Spanish -
Spain,…
Forza Horizon 5                   English,Italian,Spanish -
```

```
Spain,Czech,Hungaria…
Rust                              English,French,Italian,German,Spanish -
Spain,…

                                  lanAllSupported  \
name
Counter-Strike: Global Offensive               28
ELDEN RING                                     14
Red Dead Redemption 2                          13
Forza Horizon 5                                16
Rust                                           25

genre  \
name
Counter-Strike: Global Offensive                               Action,Free to
Play
ELDEN RING
Action,RPG
Red Dead Redemption 2
Action,Adventure
Forza Horizon 5
Action,Adventure,Racing,Simulation,Sports
Rust                              Action,Adventure,Indie,Massively
Multiplayer,RPG

                                                        developer  \
name
Counter-Strike: Global Offensive  Valve,Hidden Path Entertainment
ELDEN RING                                      FromSoftware Inc.
Red Dead Redemption 2                              Rockstar Games
Forza Horizon 5                                  Playground Games
Rust                                            Facepunch Studios

                                                                     publisher
\
name
Counter-Strike: Global Offensive                                         Valve
ELDEN RING                        FromSoftware Inc.,Bandai Namco Entertainment
Red Dead Redemption 2                                           Rockstar Games
Forza Horizon 5                                              Xbox Game Studios
Rust                                                         Facepunch Studios

                                 releaseDate  \
name
Counter-Strike: Global Offensive  2012-08-21
ELDEN RING                        2022-02-24
Red Dead Redemption 2             2019-12-05
```

```
Forza Horizon 5                   2021-11-08
Rust                              2018-02-08

minSysReq  \
name
Counter-Strike: Global Offensive
ELDEN RING                        Requires a 64-bit processor and operating
syst…
Red Dead Redemption 2             Requires a 64-bit processor and operating
syst…
Forza Horizon 5                   Requires a 64-bit processor and operating
syst…
Rust                              Requires a 64-bit processor and operating
syst…

recSysReq  \
name
Counter-Strike: Global Offensive
ELDEN RING                        Requires a 64-bit processor and operating
syst…
Red Dead Redemption 2             Requires a 64-bit processor and operating
syst…
Forza Horizon 5                   Requires a 64-bit processor and operating
syst…
Rust                              Requires a 64-bit processor and operating
syst…

                                  reviewTotal  reviewPositive  reviewNegative  \
name
Counter-Strike: Global Offensive      7327687         6502966          824721
ELDEN RING                             683586          628828           54758
Red Dead Redemption 2                  420421          379969           40452
Forza Horizon 5                        124996          110002           14994
Rust                                   938300          816121          122179

                                  reviewPercentage
name
Counter-Strike: Global Offensive              88.0
ELDEN RING                                    91.0
Red Dead Redemption 2                         90.0
Forza Horizon 5                               88.0
Rust                                          86.0
```

1.4 Data Cleaning

[9]:

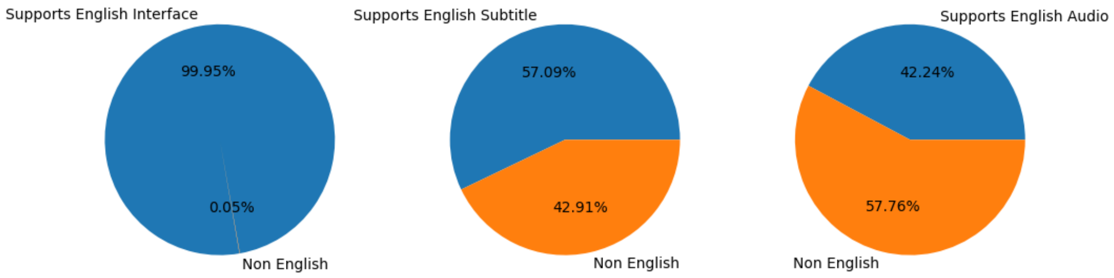
```
df.dropna(subset=["price"], inplace=True)
#df = df[df["reviewTotal"] > 100]
```

1.5 Analysis of Language support

1.5.1 English Supporting Games

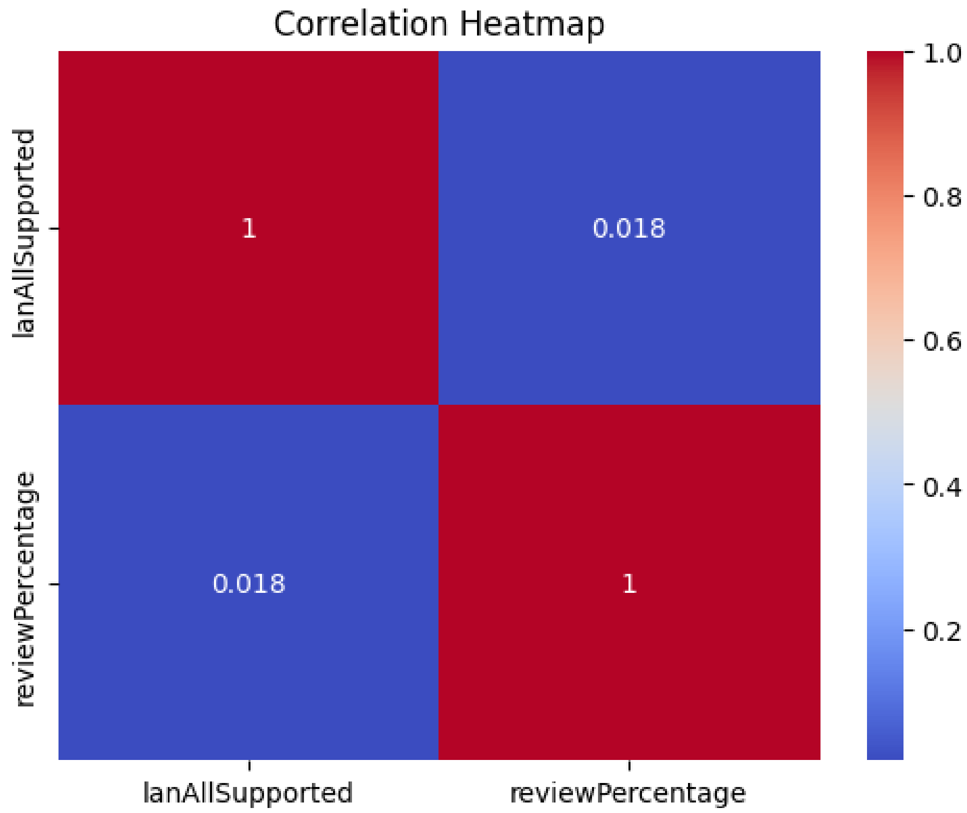
[10]:

```
enInSupNum = df.loc[df["lanInterface"].str.contains("English", case=False,␣
 ↪na=False), "lanInterface"].count()
enSubSupNum = df.loc[df["lanSubtitle"].str.contains("English", case=False,␣
 ↪na=False), "lanSubtitle"].count()
enAudSupNum = df.loc[df["LanAudio"].str.contains("English", case=False,␣
 ↪na=False), "LanAudio"].count()
totalNum = len(df)

# Create 3 column figure
fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(12, 5))

# First plot --------------------

enLabels = ["Supports English Interface", "Non English"]
xy = np.array([enInSupNum, totalNum-enInSupNum])
axes[0].pie(xy, labels = enLabels, startangle=-80, autopct='%1.2f%%')

# Second plot ---------------------
enLabels = ["Supports English Subtitle", "Non English"]
xy = np.array([enSubSupNum, totalNum-enSubSupNum])
axes[1].pie(xy, labels = enLabels, autopct='%1.2f%%')

# Third plot --------------------------
enLabels = ["Supports English Audio", "Non English"]
xy = np.array([enAudSupNum, totalNum-enAudSupNum])
axes[2].pie(xy, labels = enLabels, autopct='%1.2f%%')

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



1.5.2 Supported Language Number and positive review percentage correlation

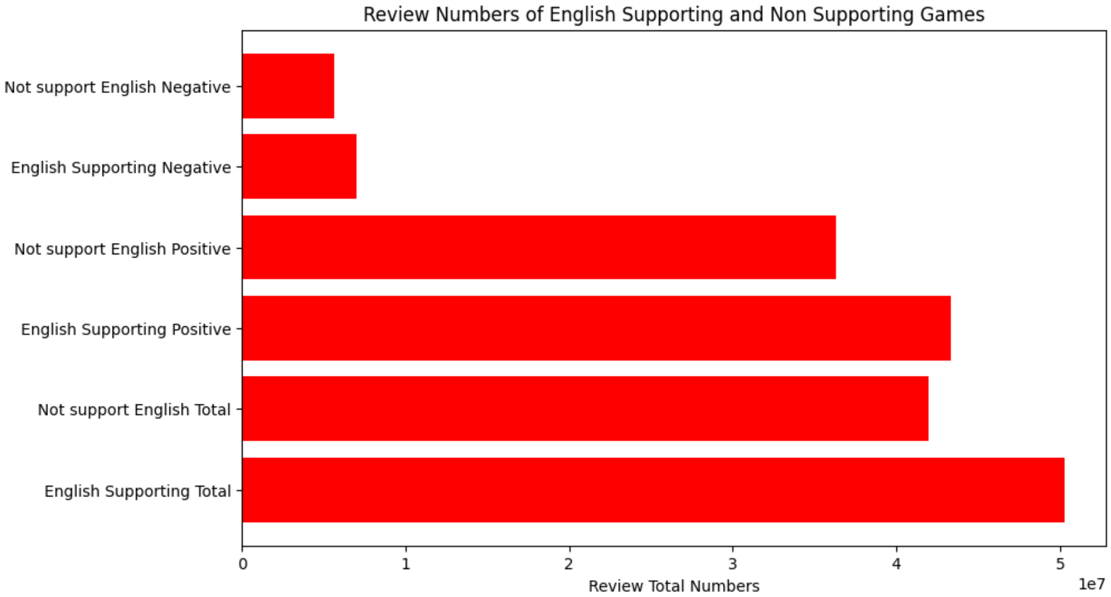
```
[11]: corrMatrix =  df[["lanAllSupported", "reviewPercentage"]].corr()
      sns.heatmap(corrMatrix, annot=True, cmap="coolwarm")
      plt.title("Correlation Heatmap")
      plt.show()
```



1.6 Total review of English supported games and not supported

```
[12]: df_eng = df[df["reviewTotal"] > 100]
      enSupRevTotal = df_eng.loc[df_eng["lanSubtitle"].str.contains("English"),␣
       ↪"reviewTotal"].sum()
      enSupPosTotal = df_eng.loc[df_eng["lanSubtitle"].str.contains("English"),␣
       ↪"reviewPositive"].sum()
      enSupNegTotal = df_eng.loc[df_eng["lanSubtitle"].str.contains("English"),␣
       ↪"reviewNegative"].sum()
```

```
nenSupRevTotal = df_eng.loc[~df_eng["lanSubtitle"].str.contains("English"),␣
 ↪"reviewTotal"].sum()
nenSupPosTotal = df_eng.loc[~df_eng["lanSubtitle"].str.contains("English"),␣
 ↪"reviewPositive"].sum()
nenSupNegTotal = df_eng.loc[~df_eng["lanSubtitle"].str.contains("English"),␣
 ↪"reviewNegative"].sum()

enSupport = ["English Supporting Total", "Not support English Total","English␣
 ↪Supporting Positive", "Not support English Positive", "English Supporting␣
 ↪Negative", "Not support English Negative"]
enSupportValues = [enSupRevTotal, nenSupRevTotal, enSupPosTotal,␣
 ↪nenSupPosTotal, enSupNegTotal, nenSupNegTotal]

fig = plt.figure(figsize=(10,6))
ax = fig.add_subplot()
ax.barh(enSupport, enSupportValues, color='Red')
ax.set_xlabel("Review Total Numbers")
ax.set_title("Review Numbers of English Supporting and Non Supporting Games")
plt.show()
```



```
[ ]:
```

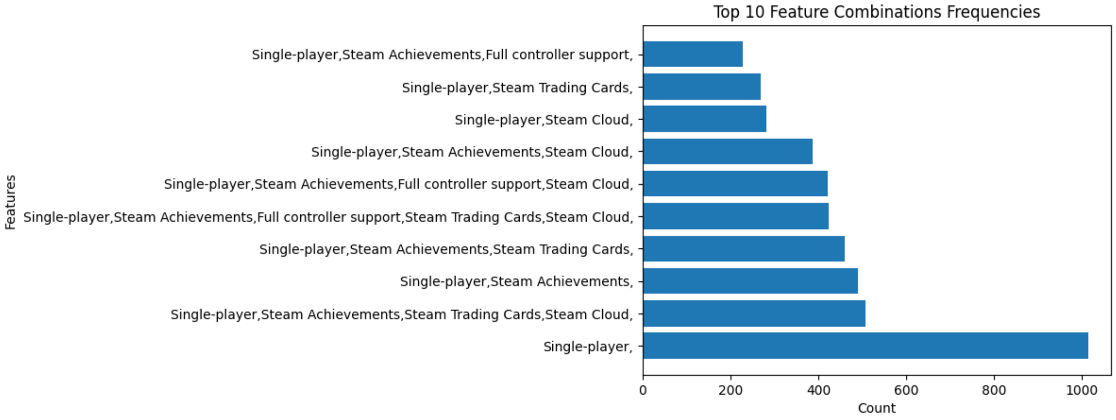
1.7 Features

[13]:
```
feature_counts = df.loc[df["reviewTotal"] > 100, "features"].value_counts()
feature_counts_df = pd.DataFrame({'Features': feature_counts.index, 'Count':␣
 ↪feature_counts.values})
feature_counts_df = feature_counts_df.sort_values(by='Count', ascending=False)
feature_counts_df['Count'] = feature_counts_df['Count'].astype(int)
feature_counts_df.loc[feature_counts_df["Count"] > 10].head(10)
```

[13]:
```
                                         Features  Count
0                                  Single-player,   1015
1  Single-player,Steam Achievements,Steam Trading…   507
2                Single-player,Steam Achievements,    490
3  Single-player,Steam Achievements,Steam Trading…   460
4  Single-player,Steam Achievements,Full controll…   424
5  Single-player,Steam Achievements,Full controll…   421
6      Single-player,Steam Achievements,Steam Cloud,    387
7                     Single-player,Steam Cloud,    281
8              Single-player,Steam Trading Cards,    268
9  Single-player,Steam Achievements,Full controll…   229
```

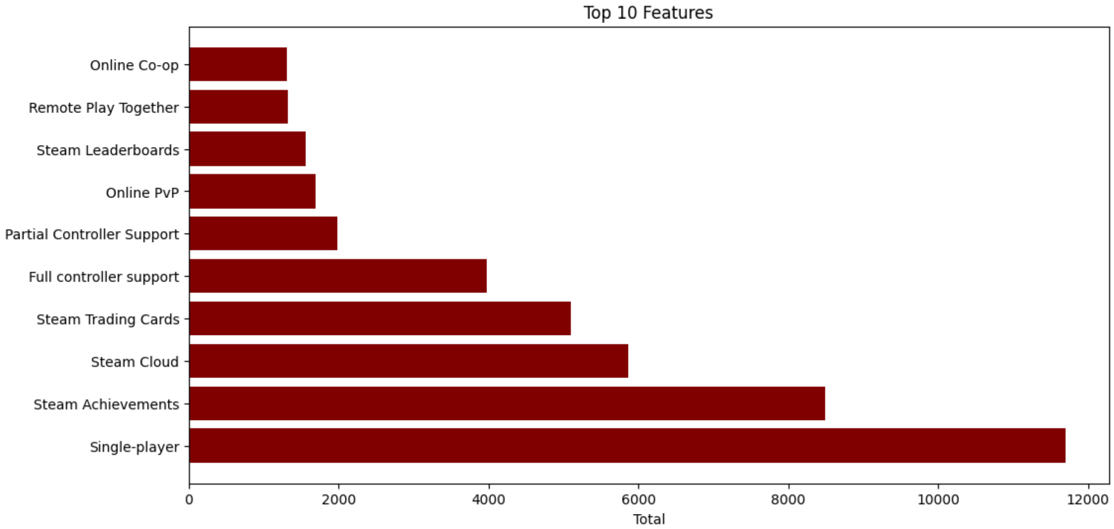
1.7.1 Top 10 Features Combinations

[14]:
```
plt.barh(feature_counts_df['Features'].head(10), feature_counts_df['Count'].
 ↪head(10))

plt.ylabel('Features')
plt.xlabel('Count')
plt.title('Top 10 Feature Combinations Frequencies')

plt.xticks(rotation=0)

plt.show()
```



1.7.2 Top 10 Features

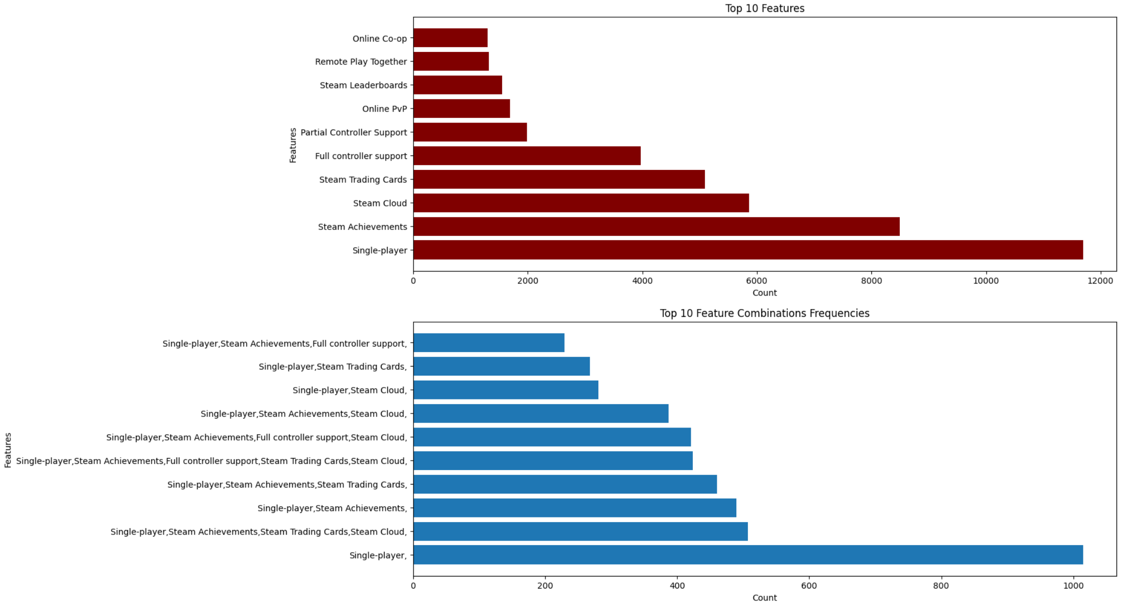
[15]:
```
features = []
def adFe(x):
    for i in x:
        features.append(i)
dfFeatures = df.loc[df["reviewTotal"] > 100, "features"].str.split(",")
dfFeatures.apply(adFe)
dfFeatures = pd.DataFrame(data=pd.Series(features), columns=["features"])
dfFeatures = dfFeatures[dfFeatures["features"] != ""]
dfFeatures.reset_index(drop=True, inplace=True)
dfFeatures = dfFeatures["features"].value_counts().head(10)

fig = plt.figure(figsize=(12, 6))
ax = fig.add_subplot()
ax.barh(dfFeatures.index, dfFeatures.values, color='maroon')
ax.set_xlabel("Total")
ax.set_title("Top 10 Features")

plt.show()
```



1.7.3 Compairing Features Combination Count and Single Feature Count

[16]:
```
#Create Figure
fig, ax = plt.subplots(nrows=2, ncols=1 ,figsize=(15, 12))

#bar 1
```

```
ax[0].barh(dfFeatures.index, dfFeatures.values, color='maroon')
ax[0].set_ylabel('Features')
ax[0].set_xlabel('Count')
ax[0].set_title('Top 10 Features')

#bar 2
ax[1].barh(feature_counts_df['Features'].head(10), feature_counts_df['Count'].
 ↪head(10))
ax[1].set_ylabel('Features')
ax[1].set_xlabel('Count')
ax[1].set_title('Top 10 Feature Combinations Frequencies')

plt.show()
```



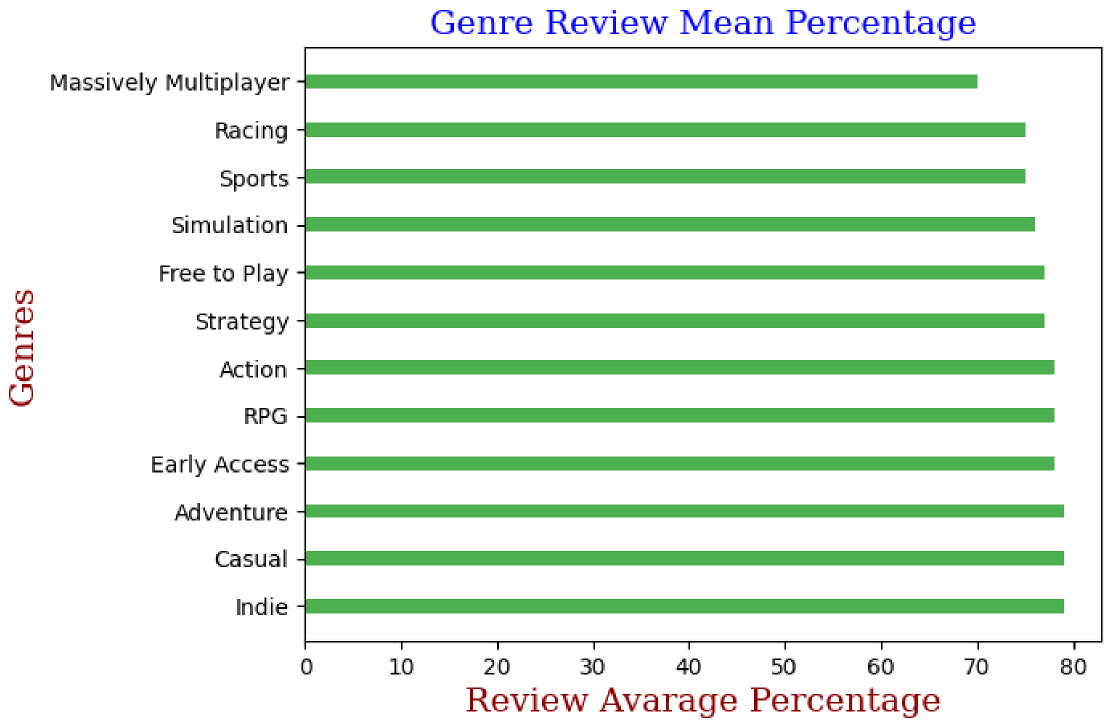
1.8 Genre Analysis

1.9 Splitting genre column to examine the number and positive review percentage by genre

[17]:
```
genreSeries = pd.Series(df["genre"].str.split(","))
genreSeries.reset_index(drop=True, inplace=True)

genres = []

for i in range(len(genreSeries)):
    for j in range(len(genreSeries[i])):
        genre = genreSeries[i][j]
        if genre not in genres and genre != None and genre != "":
            genres.append(genre)


d = {"count":0,"reviewMean":0,"reviewTotal":0}
genreDF = pd.DataFrame(data=d, index=genres)

for index, row in genreDF.iterrows():
    row["count"] = df.loc[df["genre"].str.contains(index), "genre"].count()
    row["reviewMean"] = int(df.loc[df["genre"].str.contains(index),␣
 ↪"reviewPercentage"].mean())
    row["reviewTotal"] = df.loc[df["genre"].str.contains(index), "reviewTotal"].
 ↪sum()
```

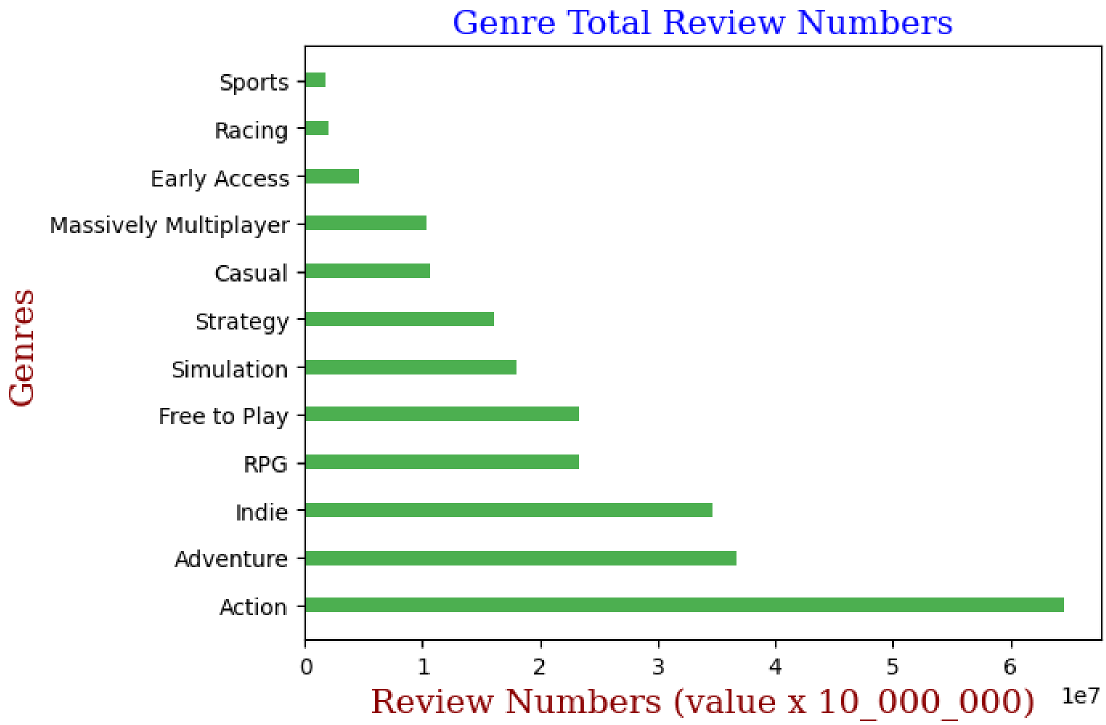
[18]:

```
genreDF = genreDF.sort_values(["reviewMean"], ascending=False)
plt.barh(genreDF[genreDF["count"] > 10].index, genreDF.loc[genreDF["count"] >␣
 ↪10, "reviewMean"], color = "#4CAF50", height = 0.3)

font1 = {'family':'serif','color':'blue','size':15}
font2 = {'family':'serif','color':'darkred','size':15}
#font3 = {'family':'serif','color':'green','size':5}

plt.title("Genre Review Mean Percentage", fontdict = font1)
plt.xlabel("Review Avarage Percentage", fontdict = font2)
plt.ylabel("Genres", fontdict = font2)

plt.show()
```



1.9.1 Most Reviewed Genres

[19]:
```
genreDF = genreDF.sort_values(["reviewTotal"], ascending=False)
plt.barh(genreDF[genreDF["reviewTotal"] > 1000].index, genreDF.
  ↪loc[genreDF["reviewTotal"] > 1000, "reviewTotal"], color = "#4CAF50", height␣
  ↪= 0.3)

font1 = {'family':'serif','color':'blue','size':15}
font2 = {'family':'serif','color':'darkred','size':15}
#font3 = {'family':'serif','color':'green','size':5}

plt.title("Genre Total Review Numbers", fontdict = font1)
plt.xlabel("Review Numbers (value x 10_000_000)", fontdict = font2)
plt.ylabel("Genres", fontdict = font2)

plt.show()
```

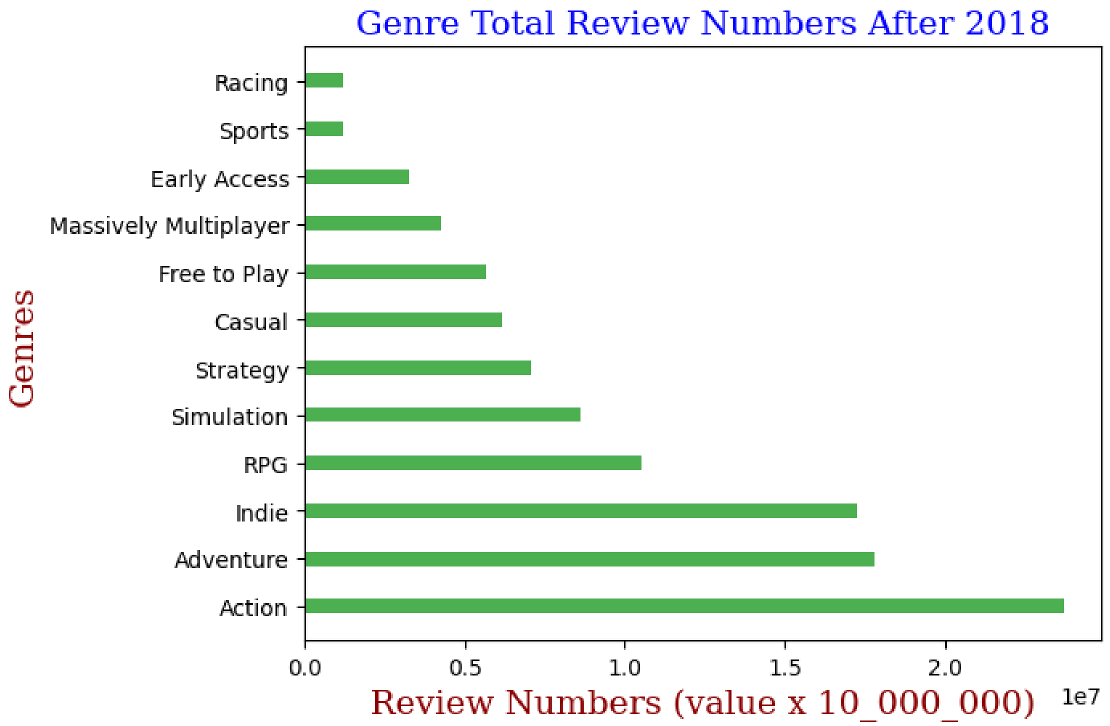


1.9.2 Most Reviewed Genres after 2018

[20]:

```
date_before = datetime.datetime(2018, 1, 1)
genDF2K18 = df[df["releaseDate"] >= date_before]

genreSeries = pd.Series(genDF2K18["genre"].str.split(","))
genreSeries.reset_index(drop=True, inplace=True)

genres = []

for i in range(len(genreSeries)):
    for j in range(len(genreSeries[i])):
        genre = genreSeries[i][j]
        if genre not in genres and genre != None and genre != "":
            genres.append(genre)


d = {"count":0,"reviewMean":0,"reviewTotal":0}
genreDF = pd.DataFrame(data=d, index=genres)

for index, row in genreDF.iterrows():
    row["count"] = genDF2K18.loc[genDF2K18["genre"].str.contains(index),␣
  ↪"genre"].count()
```

```
    row["reviewMean"] = int(genDF2K18.loc[genDF2K18["genre"].str.
 ↪contains(index), "reviewPercentage"].mean())
    row["reviewTotal"] = genDF2K18.loc[genDF2K18["genre"].str.contains(index),␣
 ↪"reviewTotal"].sum()

genreDF = genreDF.sort_values(["reviewTotal"], ascending=False)
plt.barh(genreDF[genreDF["reviewTotal"] > 1000].index, genreDF.
 ↪loc[genreDF["reviewTotal"] > 1000, "reviewTotal"], color = "#4CAF50", height␣
 ↪= 0.3)

font1 = {'family':'serif','color':'blue','size':15}
font2 = {'family':'serif','color':'darkred','size':15}
#font3 = {'family':'serif','color':'green','size':5}

plt.title("Genre Total Review Numbers After 2018", fontdict = font1)
plt.xlabel("Review Numbers (value x 10_000_000)", fontdict = font2)
plt.ylabel("Genres", fontdict = font2)

plt.show()
```



1.10 Analysis of paid and free games

```
[23]: dfFree = df.loc[df["price"].str.contains("Free", case=False) & ~df["price"].str.
       ↪contains("Demo", case=False) & ~df["price"].str.contains("Trial",␣
       ↪case=False)].describe()
      dfFree
```

```
[23]:        lanAllSupported                    releaseDate   reviewTotal  \
      count      2312.000000                           2307  2.312000e+03
      mean          4.208478  2020-02-18 02:27:18.491547392  1.022097e+04
      min           1.000000            1996-09-06 00:00:00  1.000000e+00
      25%           1.000000            2018-07-16 00:00:00  2.400000e+01
      50%           1.000000            2020-09-25 00:00:00  1.020000e+02
      75%           4.000000            2022-04-20 00:00:00  8.245000e+02
      max         103.000000            2023-07-06 00:00:00  7.327687e+06
      std           8.509152                            NaN  1.684356e+05

             reviewPositive  reviewNegative  reviewPercentage
      count    2.312000e+03     2312.000000       1732.000000
      mean     8.337888e+03     1883.080882         78.335450
      min      0.000000e+00        0.000000         18.000000
      25%      1.700000e+01        5.000000         70.000000
      50%      7.800000e+01       19.000000         81.000000
      75%      6.580000e+02      137.250000         90.000000
      max      6.502966e+06   936657.000000        100.000000
      std      1.453246e+05    27878.134806         15.199275
```

```
[27]: dfPaid = df.loc[df["price"].str.contains("TL", case=False)].describe()
      dfPaid
```

```
[27]:        lanAllSupported                    releaseDate   reviewTotal  \
      count     23844.000000                          23821  2.384400e+04
      mean          5.921238  2019-02-14 08:30:34.079173632  2.854561e+03
      min           1.000000            1983-01-17 00:00:00  1.000000e+00
      25%           1.000000            2017-06-18 00:00:00  1.900000e+01
      50%           2.000000            2020-02-21 00:00:00  7.900000e+01
      75%           7.000000            2022-01-27 00:00:00  4.600000e+02
      max         103.000000            2023-07-06 00:00:00  1.589692e+06
      std          12.311398                            NaN  2.638294e+04

             reviewPositive  reviewNegative  reviewPercentage
      count    2.384400e+04    23844.000000      19771.000000
      mean     2.509320e+03      345.240522         79.178696
      min      0.000000e+00        0.000000          0.000000
      25%      1.400000e+01        3.000000         72.000000
      50%      6.000000e+01       14.000000         84.000000
      75%      3.670000e+02       77.000000         92.000000
```

```
max      1.365490e+06   224202.000000        100.000000
std      2.379044e+04     3430.814461         17.855900
```