

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings("ignore")
```

#Amazon Sales Data

```
# Uploading data set
df = pd.read_csv("amazon_modify.csv")
df.head()

{"summary":{"\n  \"name\": \"df\", \n  \"rows\": 1465, \n  \"fields\": [\n    {\n      \"column\": \"product_id\", \n      \"properties\": {\n        \"dtype\": \"string\", \n        \"num_unique_values\": 1351, \n        \"samples\": [\n          \"B09GFLXVH9\", \n          \"B0BC9BW512\", \n          \"B097JVLW3L\", \n          ], \n          \"semantic_type\": \"\", \n          \"description\": \"\", \n          \"column\": \"product_name\", \n          \"properties\": {\n            \"dtype\": \"string\", \n            \"num_unique_values\": 1337, \n            \"samples\": [\n              \"Gluon Multipurpose Portable Electronic Digital Weighing Scale Weight Machine (10 Kg - with Back Light)\", \n              \"Akiara - Makes life easy Mini Sewing Machine with Table Set | Tailoring Machine | Hand Sewing Machine with extension table, foot pedal, adapter\", \n              \"TTK Prestige Limited Orion Mixer Grinder 500 Watts, 3 Jars (1200ml, 1000ml, 500ml) (Red)\", \n              ], \n              \"semantic_type\": \"\", \n              \"description\": \"\", \n            }, \n            {\n              \"column\": \"category\", \n              \"properties\": {\n                \"dtype\": \"category\", \n                \"num_unique_values\": 211, \n                \"samples\": [\n                  \"Electronics|Mobiles&Accessories|MobileAccessories|Photo&VideoAccessories|SelfieSticks\", \n                  \"Home&Kitchen|Kitchen&HomeAppliances|Coffee,Tea&Espresso|CoffeeGrinders|ElectricGrinders\", \n                  \"Computers&Accessories|Accessories&Peripherals|HardDriveAccessories|Caddies\", \n                  ], \n                  \"semantic_type\": \"\", \n                  \"description\": \"\", \n                }, \n                {\n                  \"column\": \"rating\", \n                  \"properties\": {\n                    \"dtype\": \"number\", \n                    \"std\": 0.29215304353491645, \n                    \"min\": 2.0, \n                    \"max\": 5.0, \n                    \"num_unique_values\": 25, \n                    \"samples\": [\n                      3.3, \n                      4.7, \n                      4.2, \n                      ], \n                      \"semantic_type\": \"\", \n                      \"description\": \"\", \n                    }, \n                    {\n                      \"column\": \"rating_count\", \n                      \"properties\": {\n                        \"dtype\": \"number\", \n                        \"std\": 42753.864952249205, \n                        \"min\": 2.0, \n                        \"max\": 426973.0, \n                        \"num_unique_values\": 1143, \n                        \"samples\": [\n                          197.0, \n                          7945.0, \n                          140036.0, \n                          ], \n                          \"semantic_type\": \"\", \n                        }, \n                      }, \n                    ], \n                  }, \n                }, \n              }, \n            }, \n          ], \n        }, \n      }, \n    }, \n  ], \n}
```

```

\"description\": \"\n      }\n    },\n    {\n      \"column\":  

\"about_product\", \n      \"properties\": {\n      \"dtype\":  

\"string\", \n      \"num_unique_values\": 1293, \n  

\"samples\": [\n      \"Advanced Bluetooth calling: Upgrade to an  

effortless calling experience - attend/reject calls and dial numbers,  

from your wrist.;Digital crown: Navigate through the watch, adjust  

volume and change the watch face via the fully-functional crown.|  

1.72\\u2019\\u2019display: ColorFit Pro 4 features 1.72\\u2019\\u2019  

TFT LCD with 25% more screen area than ColorFit Pro 3.;Vivid clarity:  

View information under the brightest sun, thanks to 311 PPI and 500  

nits of brightness.|60Hz refresh rate: Get smoother scrolling &  

navigation experience.;100 sports modes: Take your pick from 100  

sports modes and ace your game.|Noise Health Suite: Know how your body  

is doing with the battery of fitness features.;Productivity suite: Get  

more work done with quick reply options, stock market updates, alarm  

and disconnect with smart DND when you want to.; Water Resistance  

Level: Water Resistant|Item Type Name: Smartwatch; Connectivity  

Technology: Usb; Included Components: \\u200eSmartwatch, Magnetic  

Charger, User Manual, Warranty Card\", \n      \"Fire-Boltt is  

India' No 1 Wearable Watch Brand Q122 by IDC Worldwide quarterly  

wearable device tracker Q122.\\u30101.69\\u201d HD Large Touch  

Screen\\u3011- Fire-Boltt Ninja 3 comes with a 1.69\\u201d HD Full  

Touch Display for smooth swipes and clear vision;\\u3010SP02/ Oxygen,  

Heart Rate\\u3011 - Fire-Boltt Ninja 3 Smartwatch comes with real time  

24*7 SP02 / Blood Oxygen tracking, Dynamic Heart Rate Monitoring (If a  

patient is suffering from Covid 19 please use a medical device  

prescribed by the Doctor)|\\u301060 workout modes\\u3011- This  

smartwatch consists of 60 sports mode to track. Keep a track of all  

your activities and compare history to analyse your performance. Count  

steps, distance, and calories burned.;\\u3010IP68 Water Resistant\\u3011-  

This smartwatch can withstand dust, spills, raindrops and is  

sweatproof too|\\u3010POWERFUL BATTERY\\u3011 - About 7 days battery  

life and a Standby Time of 25 Days \\u3010Multiple Watch Faces\\u3011-  

Unlimited Customized Built in Watch Faces and also multiple watch  

faces through the app;\\u3010Stay Social Stay Updated\\u3011 \\u2013  

Inbuilt Social Media Notifications.|\\u3010All In One Smart Coach\\u3011  

- Track your Daily Steps, Sleep, Fitness, Sports, Heart Rate and  

SP02 \\u3010Enjoy Music And Camera Control\\u3011 \\u3010IP68 Water  

Resistant\\u3011- This smartwatch can withstand dust, spills,  

raindrops and is sweatproof too; Water Resistance Level:  

water_resistant|Connectivity Technology: Bluetooth; Clasp Type: Tang  

Buckle; Compatible Devices: Smartphonetablet; Human Interface Input:  

Touch Screenbuttons; Item Type Name: Smart Watch; Included Components:  

1 Smartwatch, 1 Manual, 1 Magnetic Charger, 1 Warranty Card; Band  

Color: Green; Band Material Type: Silicone; Case Material Type:  

Plastic; Color Name: Green\", \n      \"Keyboard : Standard  

keyboard|Rupee key, Comfortable|Silent Durable keys|Mouse : Ergonomic  

design, Accurate optical sensor|High resolution enabling faster  

navigation\\n      ], \n      \"semantic_type\": \"\", \n

```

```

\"description\": \"\\n      }\n    },\n    {\n      \"column\":
\"user_id\", \n      \"properties\": {\n        \"dtype\": \"string\", \n
        \"num_unique_values\": 1194, \n        \"samples\": [\n
\"AEWW4LJ0VXD65UKE7QCBCHQZMG7A,AGVWB5YTQZC7GAIV4PCCOFF2U27A,AFUR2THG6B
YV6IRA5JV6LBQNG2AA,AFSG5TXKKCLHBK3FABKJABBBUHEQ,AF2D0UVTY5LHTVWGGVE6YH
W5KEGA,AFEX50M5UOST6P0IWTBW6TCEZ2YA,AFKZZ0Q7J2S0XP30HFBEDXNFINCQ,AFD6P
5IRXY6KWXUW4H7X6ECRMSLA\", \n
\"AG65C34LATM4J3ZFKJJPDNISZKUQ,AG76GICZHJGA7YVN4TORX36ONVYA,AHHIHCEKEY
DIRPJ5W7WXGLB3E66Q,AFYSF663502EAPR4GMVBH74FSIFQ,AHAVRPA7Z3PKTTWVBVUISC
KI7RYQ,AEDH674UH53A5FKLUZCCM5LVKUQQ,AEUK344UA4FNU4PR4AWSPKW5PPQ,AGPAK
6ELVZPVKQ7GEZ7IUHNK2C3Q\", \n
\"AE3S2ZAEMH765KUJ57DR6HBZBB3Q,AHSIVUNTJMI5S5AJGFDE5EDQ355Q,AGQUDHVCMB
W7DYS2HT5HA3QCZIEA,AFNXZNINQLTHKVRFI37VQAFFGOA,AGVKQNHNS7PQK63FIB6EVC
5GUAMQ,AEHZ4N0Z5SIDQLGODWS4UZ6RVQJA,AGBVAEUPMWYCDQIKNAD2DXEYWXZA,AE0JW
70WUZROZ6Z66ZQU33Y2ZYLQ\"\\n      ], \n      \"semantic_type\":
\"\", \n      \"description\": \"\\n      }\n    },\n    {\n
\"column\": \"user_name\", \n      \"properties\": {\n
\"dtype\": \"string\", \n      \"num_unique_values\": 1194, \n
\"samples\": [\n        \"Kindle Customer,Aryan,pooja reddy,Amazon
Customer,Meenakshi jasrotia,imamthulla,Anan,Sanjay Chavan\", \n
\"zain,Deepak,VIMAL,Shiv Sagar,Tamil selvan,Rakesh yadav,PAGOLA
SURESH,Olivia\", \n        \"Fardeen mujawar,Pavan,Danny,Siddhartha
Pratap,Rabindra Kumar Das,Amazon Customer,Rakesh Ranga Yadav,Nivedita
Chatterjee\"\\n      ], \n      \"semantic_type\": \"\", \n
\"description\": \"\\n      }\n    },\n    {\n      \"column\":
\"review_id\", \n      \"properties\": {\n        \"dtype\":
\"string\", \n        \"num_unique_values\": 1194, \n
\"samples\": [\n
\"R1T4TKPYU5EJCB,R1D38AX8G0RVNS,R1KHCRDEEREQG7,R396UL830TSD8F,R3CY781P
K5CB8A,RBCCWRI4IUHH5,R2K7JYQMGQ31YJ,R3P0GJ4V5HPF2M\", \n
\"R2JCUKBR0BQ8ES,RNVX0V6SJF3CP,RW5MJG9LTX6QD,R37PSG13H70Z1F,R17RIHK0XX
QDH5,R2P187SB04SEMH,R1V49G7PD8Y93G,RU78E5A4MW0PK\", \n
\"RD60IJUG0R241,R3EUJ7A6LG8X7V,R1DWGT4USEVGYK,R187KH5XJBPS86,R2XYH31E9
NK0GU,RDYNZZPHU7SZK,R2MR0DYZVFN3HA,R3PV91U8ZYN5DU\"\\n      ], \n
\"semantic_type\": \"\", \n      \"description\": \"\\n      }\n
    },\n    {\n      \"column\": \"review_title\", \n
\"properties\": {\n        \"dtype\": \"string\", \n
\"num_unique_values\": 1194, \n        \"samples\": [\n
\"Clearly makes a difference,Good,Value for money,Good material,The
ink of parker is very lite,Good,Good,Very good\", \n        \"Good
product with less money,At this price ok ok.,Good product,Good mouse
at this price range,Good,Good for daily use ke liye,Good,Good\", \n
\"Ok,Like all other ball pens,Regular pen over priced,Nice,It is
fine.,Awful blue ink,Nice and my Favorite Pen,Reasonable price\"\\n
      ], \n      \"semantic_type\": \"\", \n      \"description\": \"\\n
    }\n    },\n    {\n      \"column\": \"review_content\", \n
\"properties\": {\n        \"dtype\": \"string\", \n
\"num_unique_values\": 1212, \n        \"samples\": [\n
\"Reviewing just after a day of using this product. We made French

```

[illegible]

recently purchased a new Qualcomm 3 Qbix car charger & was looking for a cable with USB A to Micro B connector. I listed my priorities under various heads to come up with a conclusion and let us compare the actual product based upon my initial requirements:1.) DATA

EXCHANGE:\*\*\*\*\*Since it was to be primarily used for fast mobile devices charging in car, Data exchange capability was not much of my concern. Preference though would surely have been a USB 3 but it didn't bother me if I could only get a USB 2.0 too.

ACTUAL PRODUCT: I did try to copy a movie file just for the sake of testing data exchange and I found it to be pretty well. I did not capture any speed data but then we all know speed of data transfer also varies with the type of data being transferred. The more variety of data being transferred simultaneously the lower will be the speed.2.) LENGTH OF THE

CABLE :\*\*\*\*\*Again since I could not afford to have a lengthy loop of cable bunched around my gear knob, I preferred to keep it short & simple hence my only lookout was upto around a meter or below. ACTUAL PRODUCT: The cable came nicely packed in a paper packet and was precisely 0.9 meters or approximately 3 feet long. The length was sufficient for me to plug-in any of the mobile devices to my car charger at the drivers or the side passenger's seat.3.) TANGLE FREE/ FLEXIBILITY/ STRENGTH

STANDARDS:\*\*\*\*\*

\*\*\*\*\*I am not particularly a fan of those stubborn braided wires which are so hard that they retain the shape in which they are bent. I wanted something that was thick yet flexible enough to acquire a circular shape when bunched. ACTUAL PRODUCT: The cable received looked exactly as shown over the site with good flexibility , reasonable thickness & a sturdy intermediate cable. The whole construction of the cable due to its cable size & flexibility is almost tangle-free. The associated cable was not exactly thick but can't be termed as thin or delicate too. It's not the thickest I've seen but then thickest doesn't always means most durable. Given my application it's more than just suitable. The overall built & quality of the cable & insulation looks promising enough to last few years. Even if used for other than car charging it looks durable enough to last long. I had further shortlisted mansaa & an amkette cable for the same purpose but they were too long for my requirement.4.) COMPACT MOULDED

CONNECTORS:\*\*\*\*\*Had an inclination towards moulded connectors to avoid any issues where the connectors break open exposing the terminal PCBs. ACTUAL PRODUCT : There are no complaints regarding the connectors of the actual cable. The connectors are perfectly moulded without any joints or risks of splitting open. The connector casings are further quite compact at terminals to fit in comfortably at scarce spaces. The connector ports are sturdy enough both at USB A & micro B ports. The micro B port pins lock securely onto the charging mobile devices which is quite good. No signs of loose construction. Being Gold-plated is more of a misleading



& fancy term(in this case) as most of the metal ports designed today already have a corrosion resistance & nobody is going to use them in saline sea water anyways.5.) AVAILABILITY OF TIES/VELCRO

STRAPS :\*\*\*\*\*As per my intended use in a car where compactness was of paramount importance, I expected an included cable tie or a Velcro strap would be a nice add-on to properly adjust & arrange the cable as per requirement.ACTUAL PRODUCT: This I miss the most in the provided actual cable, there is no provision of an included strap or cable tie through which I could adjust my required cable length easily.6.)

DECENT CURRENT HANDLING

CAPABILITY:\*\*\*\*\*

\*\*\*\*\*On the newest QC 3.0 certified chargers the current transmission can go up to 3.4 amps in certain cases hence the cable needed to have a decent current handling ability.ACTUAL PRODUCT : The actual cable has a nameplate rating of handling up to 2.1 amps against demands at new QC 3.0 chargers that could go up to 3.4 amps. Here, let me clear out that the latest QC chargers vary voltage to current ratios to achieve desired fast charging & thus it is not like a continuous flow of 3.4 amps is there, it keeps on reducing hence cable ratings designed for a continuous current handling of anything above 2.0 amps would sufficiently work with QC 3.0 chargers.Have tried it safely multiple times charging my Samsung S7 from 10 % without the cable getting hot.7.) WARRANTY:\*\*\*\*\*Comes with a 1 year warranty as expected. Not the best in the industry but reasonable.OTHER

OBSERVATIONS:\*\*\*\*\*Apart from the above listed features, the other details of the actual product received worth mentioning are:8.) The cable is manufactured in China & imported by amazon warehouse dealers under the brand name & philosophy of amazonbasics.9.) The cable has a manufacturing date of Oct 2017 & it was imported to India in Dec 2017.10.) The cable has a MRP tag of 495 bucks however I purchased it online for 269 bucks.being a prime member shipping was free.11.) A customer care toll free number & email is also printed over the label for registration of any consumer grievances.FINAL VERDICT:xxxxxxxxxxxxxxxxIt has been few days since I have received & been using this product. So far everything from construction to performance seems to be convincing enough to recommend it and for a price of around 260 I suppose, its worth it.Will surely update if any malfunction is observed.,Worth for buy!,The quality that amazon basics at times give at cheap prices is beyond imagination. simply superb, goverment shouldnt hinder amazon products , amazon products rather gives competition to local qualityless products which consumers are forced to buy beacause they have no quality competition. Make in india is good, but if the make in india products are simple cheap copies of branded products without any investment in R and D, without R and D make in india would never be successfull and ousting companies like amazon will only lead to loss for consumers , govt should infact encourage such competition.,Amazon basics provides one

of the best cables available for charging your phone or connecting devices. As a past customer of many cables from Amazon this cable doesn't also disappoint, Supports fast charging for all my Samsung phones. I use Samsung a9pro 2016, Samsung s8plus 2017, which this cable is compatible with. Very sturdy, thick and very long. 6 ft Very affordable pricing. Thanks Amazon I also use a USB c cable for my Samsung s20fe., Super, Product charging is ok.. however it's mere 1 foot in length.. the vendor could have mentioned correct product description.. there is no need to mislead.. too early to say performance as I have received it today., Good, I have bought many cheap chinese micro usb cable in Rs 50 and Rs 100 of ubon and of many other chinese local companies, and none of them worked properly. Finally i decided to go for this. And it is charging as well transferring data, without any issue and i am very happy with my purchase. My advice : Dont buy, cheap chinese local cables of. You will have to throw them in dustbin after some time. Better buy this one.

```

    ],
    "semantic_type": "",
    "description": "",
    "column": "img_link",
    "properties": {
      "dtype": "string",
      "num_unique_values": 1412,
      "samples": [
        "https://m.media-amazon.com/images/W/WEBP_402378-T2/images/I/51esjc0y79L._SY300_SX300_QL70_FMwebp_.jpg",
        "https://m.media-amazon.com/images/I/41nRBNNDnNL._SX300_SX300_QL70_FMwebp_.jpg",
        "https://m.media-amazon.com/images/I/31-hWNXDxiL._SX300_SX300_QL70_ML2_.jpg"
      ]
    },
    "column": "product_link",
    "properties": {
      "dtype": "string",
      "num_unique_values": 1465,
      "samples": [
        "https://www.amazon.in/Snapdragon-Resolution-Refresh-27-81Cm-Display/dp/B09XXZXQC1/ref=sr_1_437?qid=1672903017&s=computers&sr=1-437",
        "https://www.amazon.in/Skadiio-Accessories-Receiver-Compatible-dongle/dp/B09LHXNZLR/ref=sr_1_195?qid=1672909134&s=electronics&sr=1-195",
        "https://www.amazon.in/LOHAYA-Assistant-Compatible-Xstream-Function/dp/B09LV13JFB/ref=sr_1_408?qid=1672909144&s=electronics&sr=1-408"
      ]
    },
    "column": "discounted_price(\\u20b9)",
    "properties": {
      "dtype": "number",
      "std": 6944.304394400705,
      "min": 39.0,
      "max": 77990.0,
      "num_unique_values": 550,
      "samples": [
        20999.0,
        1699.0,
        419.0
      ]
    },
    "column": "actual_price(\\u20b9)",
    "properties": {
      "dtype": "number",
      "std": 10874.826864024224,
      "min": 39.0,
      "max": 139900.0,
      "num_unique_values": 449,
      "samples": [
        3210.0,
        1129.0,
        4500.0
      ]
    },
    "column": "discount_percentage%",
    "properties": {
      "dtype": "number",
      "std": 10874.826864024224,
      "min": 39.0,
      "max": 139900.0,
      "num_unique_values": 449,
      "samples": [
        3210.0,
        1129.0,
        4500.0
      ]
    }
  ],
  "semantic_type": "",
  "description": "",
  "column": "discount_percentage%",
  "properties": {
    "dtype": "number",
    "std": 10874.826864024224,
    "min": 39.0,
    "max": 139900.0,
    "num_unique_values": 449,
    "samples": [
      3210.0,
      1129.0,
      4500.0
    ]
  }
}
```

```
\n\"properties\": {\n      \"dtype\": \"number\", \n      \"std\": 21.635905180856668, \n      \"min\": 0.0, \n      \"max\": 94.0, \n      \"num_unique_values\": 92, \n      \"samples\": [\n      86.0, \n      72.0, \n      26.0\n      ], \n      \"semantic_type\": \"\", \n      \"description\": \"\" \n    } \n  ] \n}\", \"type\": \"dataframe\", \"variable_name\": \"df\"}
```

```
df.shape
```

```
(1465, 16)
```

```
df['rating'].sort_values(ascending= False).reset_index()
```

```
{\"summary\": \"{\\n  \\\"name\\\": \\\"df['rating']\\\", \\n  \\\"rows\\\": 1465, \\n  \\\"fields\\\": [\\n    {\\n      \\\"column\\\": \\\"index\\\", \\n      \\\"properties\\\": {\\n        \\\"dtype\\\": \\\"number\\\", \\n        \\\"std\\\": 423, \\n        \\\"min\\\": 0, \\n        \\\"max\\\": 1464, \\n        \\\"num_unique_values\\\": 1465, \\n        \\\"samples\\\": [\\n          1188, \\n          895, \\n          676 \\n        ], \\n        \\\"semantic_type\\\": \\\"\\\", \\n        \\\"description\\\": \\\"\\\" \\n      } \\n    }, \\n    {\\n      \\\"column\\\": \\\"rating\\\", \\n      \\\"properties\\\": {\\n        \\\"dtype\\\": \\\"number\\\", \\n        \\\"std\\\": 0.2921530435349165, \\n        \\\"min\\\": 2.0, \\n        \\\"max\\\": 5.0, \\n        \\\"num_unique_values\\\": 25, \\n        \\\"samples\\\": [\\n          4.1, \\n          3.3, \\n          5.0 \\n        ], \\n        \\\"semantic_type\\\": \\\"\\\", \\n        \\\"description\\\": \\\"\\\" \\n      } \\n    } \\n  ] \\n}\", \"type\": \"dataframe\"}
```

```
df['discounted_price(₹)'] =
df['discounted_price'].str.replace('₹','').str.replace(","," ")
df['discounted_price(₹)'] = df['discounted_price(₹)'].astype(float)
df['actual_price(₹)'] =
df['actual_price'].str.replace('₹','').str.replace(","," ")
df['actual_price(₹)'] = df['actual_price(₹)'].astype(float)
df['discount_percentage(%)'] =
df['discount_percentage'].str.replace('%','')
df['discount_percentage(%)'] =
df['discount_percentage(%)'].astype(float)
df['rating'] = df['rating'].str.replace('|','4.8')
df['rating'] = df['rating'].astype(float)
df['rating_count'] = df['rating_count'].str.replace(',','')
df['rating_count'] = df['rating_count'].astype(float)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 1465 entries, 0 to 1464
```

```
Data columns (total 16 columns):
```

#	Column	Non-Null Count	Dtype
0	product_id	1465 non-null	object
1	product_name	1465 non-null	object



```

2   category          1465 non-null object
3   rating            1465 non-null float64
4   rating_count      1463 non-null float64
5   about_product     1465 non-null object
6   user_id           1465 non-null object
7   user_name         1465 non-null object
8   review_id         1465 non-null object
9   review_title      1465 non-null object
10  review_content    1465 non-null object
11  img_link          1465 non-null object
12  product_link      1465 non-null object
13  discounted_price(₹) 1465 non-null float64
14  actual_price(₹)    1465 non-null float64
15  discount_percentage(%) 1465 non-null float64
dtypes: float64(5), object(11)
memory usage: 183.2+ KB

```

## 1. What is the average rating for each product category ?

```

# Average rating by product and Category
avg_rating_by_procat = df.groupby(['category', 'product_name'])
['rating'].mean().reset_index()

# Top ten average rating product and category
top_ten_avg_rating =
avg_rating_by_procat.sort_values(by='rating', ascending=
False).head(10)
print(top_ten_avg_rating)

# Plotting into the bar graph
plt.figure(figsize= (8,6))
top_ten_avg_rating.plot(kind='bar')
plt.xlabel("Category and Product")
plt.ylabel("Rating")
plt.title("Top Ten Category and Product of Average Rating")
plt.xticks(rotation= 45, ha= 'right')
plt.show()

# Observation or Insights
print("OBSERVATION OR INSIGHTS")
print("You can match the x-axis number with the category and product
number both are same so you can see the rating by matching its
number")
print("You can see that the 3 index (120,109,217) have 5 rating")

```

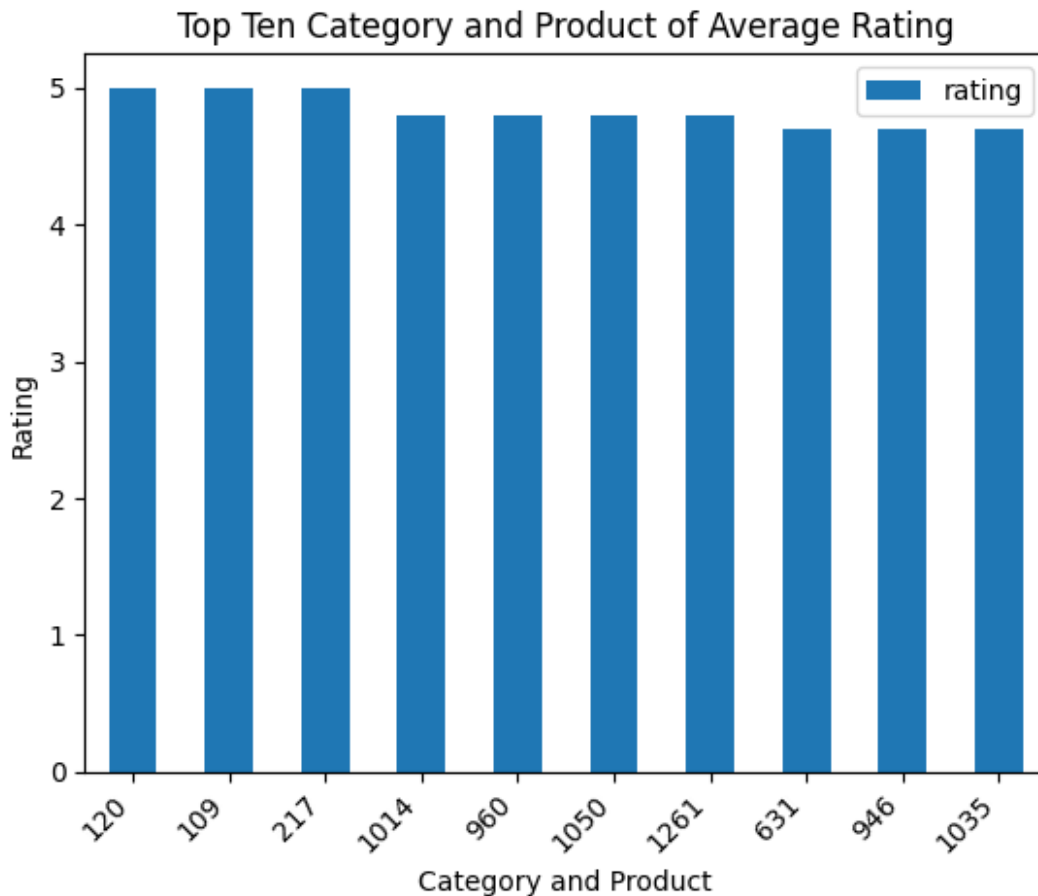
```
print("so you can say that three product are better than others in  
rating")
```

	category \	
120	Computers&Accessories Accessories&Peripherals ...	
109	Computers&Accessories Accessories&Peripherals ...	
217	Computers&Accessories Accessories&Peripherals ...	
1014	Home&Kitchen Kitchen&HomeAppliances SmallKitch...	
960	Home&Kitchen Heating,Cooling&AirQuality WaterH...	
1050	Home&Kitchen Kitchen&HomeAppliances SmallKitch...	
1261	Home&Kitchen Kitchen&HomeAppliances Vacuum,Cle...	
631	Electronics HomeTheater,TV&Video Televisions S...	
946	Home&Kitchen Heating,Cooling&AirQuality WaterH...	
1035	Home&Kitchen Kitchen&HomeAppliances SmallKitch...	

	product_name	rating
120	Syncwire LTG to USB Cable for Fast Charging Co...	5.0
109	REDTECH USB-C to Lightning Cable 3.3FT, [Apple...	5.0
217	Amazon Basics Wireless Mouse   2.4 GHz Connect...	5.0
1014	Instant Pot Air Fryer, Vortex 2QT, Touch Contr...	4.8
960	Swiffer Instant Electric Water Heater Faucet T...	4.8
1050	Oratech Coffee Frother electric, milk frother ...	4.8
1261	Eureka Forbes car Vac 100 Watts Powerful Sucti...	4.8
631	Sony Bravia 164 cm (65 inches) 4K Ultra HD Sma...	4.7
946	Campfire Spring Chef Prolix Instant Portable W...	4.7
1035	Multifunctional 2 in 1 Electric Egg Boiling St...	4.7

<Figure size 800x600 with 0 Axes>



#### OBSERVATION OR INSIGHTS

You can match the x-axis number with the category and product number both are same so you can see the rating by matching its number  
 You can see that the 3 index (120,109,217) have 5 rating  
 so you can say that three product are better than others in rating

## 2. What are the top rating\_count products by category ?

```
# Categories product and category by rating count
avg_rating_count_by_procat = df.groupby(['category', 'product_name'])
['rating_count'].value_counts().reset_index()
top_ten_rating_count = avg_rating_count_by_procat.sort_values(by=
'reating_count', ascending= False).head(10)
print(top_ten_rating_count)

# Plotting into the bar graph
plt.figure(figsize= (8,6))
top_ten_rating_count.plot(kind='bar')
```

```
plt.xlabel("Product and Category index number")
plt.ylabel("Rating Count")
plt.title("Rating Count of Product by Category")
plt.xticks(rotation= 45, ha= 'right')
plt.show()
```

#### # Observation and Insights

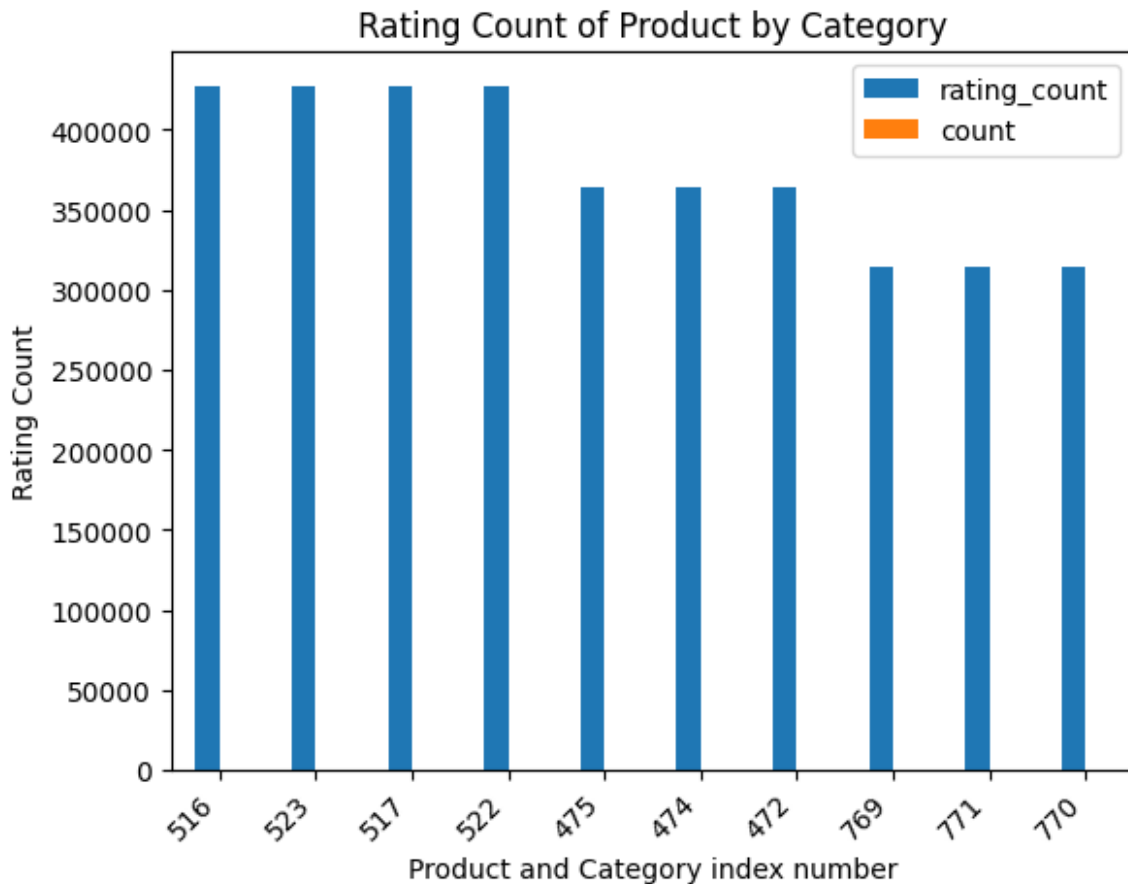
```
print("Observation and Insights")
print("You can match the x-axis with the index number of product and  
category both are same so can categories by index")
print("You can see that the rating count of index(523,517,516,522) are  
almost same so yo can say that 4 product have highest index")
```

	category \
516	Electronics HomeTheater,TV&Video Accessories C...
523	Electronics HomeTheater,TV&Video Accessories C...
517	Electronics HomeTheater,TV&Video Accessories C...
522	Electronics HomeTheater,TV&Video Accessories C...
475	Electronics Headphones,Earbuds&Accessories Hea...
474	Electronics Headphones,Earbuds&Accessories Hea...
472	Electronics Headphones,Earbuds&Accessories Hea...
769	Electronics Mobiles&Accessories Smartphones&Ba...
771	Electronics Mobiles&Accessories Smartphones&Ba...
770	Electronics Mobiles&Accessories Smartphones&Ba...

	product_name	rating_count
count		
516	Amazon Basics High-Speed HDMI Cable, 6 Feet (2...	426973.0
1		
523	AmazonBasics Flexible Premium HDMI Cable (Blac...	426973.0
1		
517	Amazon Basics High-Speed HDMI Cable, 6 Feet - ...	426973.0
1		
522	AmazonBasics Flexible Premium HDMI Cable (Blac...	426972.0
1		
475	boAt Bassheads 100 in Ear Wired Earphones with...	363713.0
1		
474	boAt Bassheads 100 in Ear Wired Earphones with...	363713.0
1		
472	boAt BassHeads 100 in-Ear Wired Headphones wit...	363711.0
1		
769	Redmi 9 Activ (Carbon Black, 4GB RAM, 64GB Sto...	313836.0
1		
771	Redmi 9A Sport (Coral Green, 2GB RAM, 32GB Sto...	313836.0
1		
770	Redmi 9A Sport (Carbon Black, 2GB RAM, 32GB St...	313832.0
1		

<Figure size 800x600 with 0 Axes>



#### Observation and Insights

You can match the x-axis with the index number of product and category both are same so can categories by index

You can see that the rating count of index(523,517,516,522) are almost same so you can say that 4 product have highest index

### 3. What is the distribution of discounted prices vs. actual prices?

```
import plotly.express as px

# Create a scatter plot with Plotly
fig = px.scatter(df, x='actual_price(₹)', y='discounted_price(₹)',
title='Distribution of Discounted Prices vs. Actual Prices',
labels={'actual_price': 'Actual Price',
'discounted_price': 'Discounted Price'})

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



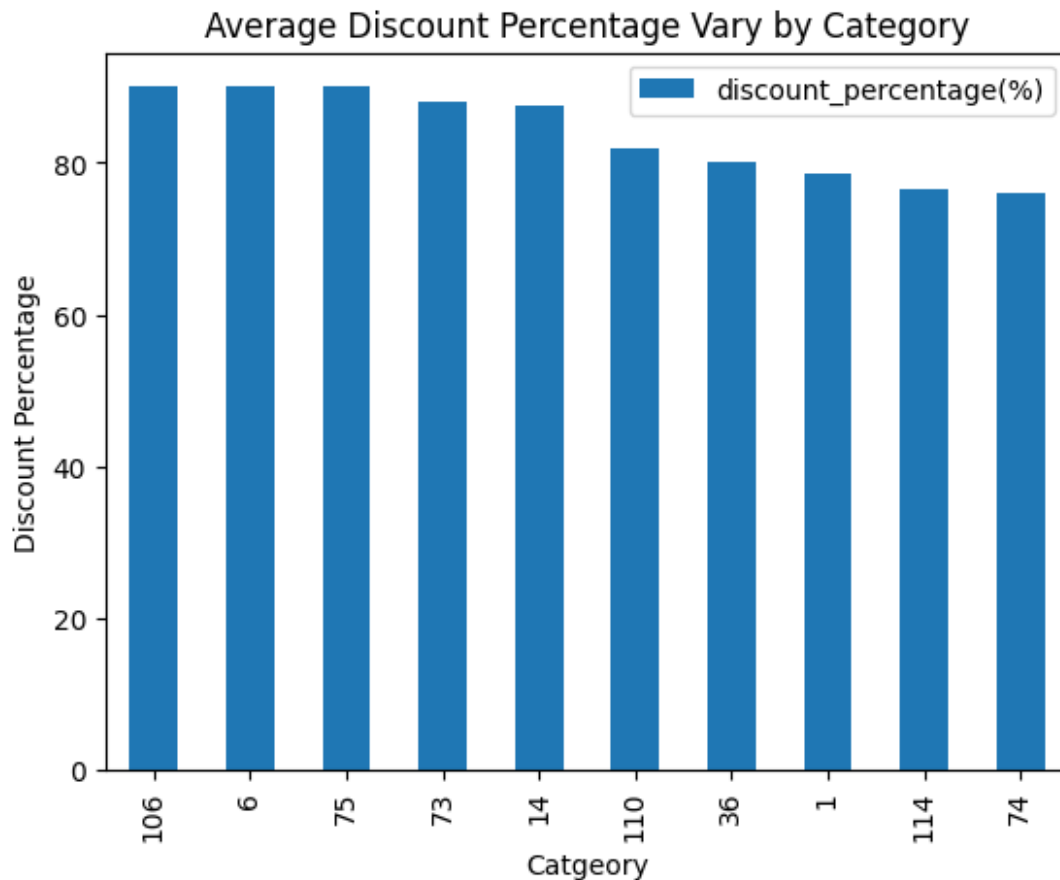
## 4. How does the average discount percentage vary across categories?

```
# Top ten category by discount percentage
avg_disc_per_by_cat = df.groupby(['category'])
['discount_percentage(%)'].mean().reset_index().sort_values(by=
'discount_percentage(%)', ascending= False).head(10)
print(avg_disc_per_by_cat)

# Plot into the graph
avg_disc_per_by_cat.plot(kind= 'bar')
plt.xlabel("Category")
plt.ylabel("Discount Percentage")
plt.title("Average Discount Percentage Vary by Category")
plt.show()

# Observation and Insights
print("Observation OR Insights")
print("You can match the x-axis number with index number of the
category so can distinguished it as category")
print("You can see that the first three category have highest discount
than other product")
```

	category
discount_percentage(%)	
106	Electronics Mobiles&Accessories MobileAccessor...
90.0	
6	Computers&Accessories Accessories&Peripherals ...
90.0	
75	Electronics Headphones, Earbuds&Accessories Ear...
90.0	
73	Electronics Headphones, Earbuds&Accessories Ada...
88.0	
14	Computers&Accessories Accessories&Peripherals ...
87.5	
110	Electronics Mobiles&Accessories MobileAccessor...
82.0	
36	Computers&Accessories Components InternalHardD...
80.0	
1	Computers&Accessories Accessories&Peripherals ...
78.5	
114	Electronics Mobiles&Accessories MobileAccessor...
76.4	
74	Electronics Headphones, Earbuds&Accessories Cases
76.0	



#### Observation OR Insights

You can match the x-axis number with index number of the category so can distinguished it as category

You can see that the first three category have highest discount than other product

## 5. What are the most popular product names?

```
# Count the occurrences of each product name
popular_product_names =
df['product_name'].value_counts().reset_index().head(10)

# Rename the columns for clarity
popular_product_names.columns = ['Product Name', 'Frequency']

# Print or display the top 10 most popular product names
print("Top 10 Most Popular Product Names:")
print(popular_product_names.head(10))

plt.figure(figsize= (8,6))
popular_product_names.plot(kind='bar')
```

```
plt.ylabel("Frequency")
plt.xlabel("Product Name")
plt.title("Top Ten Product Name ")
plt.show()
```

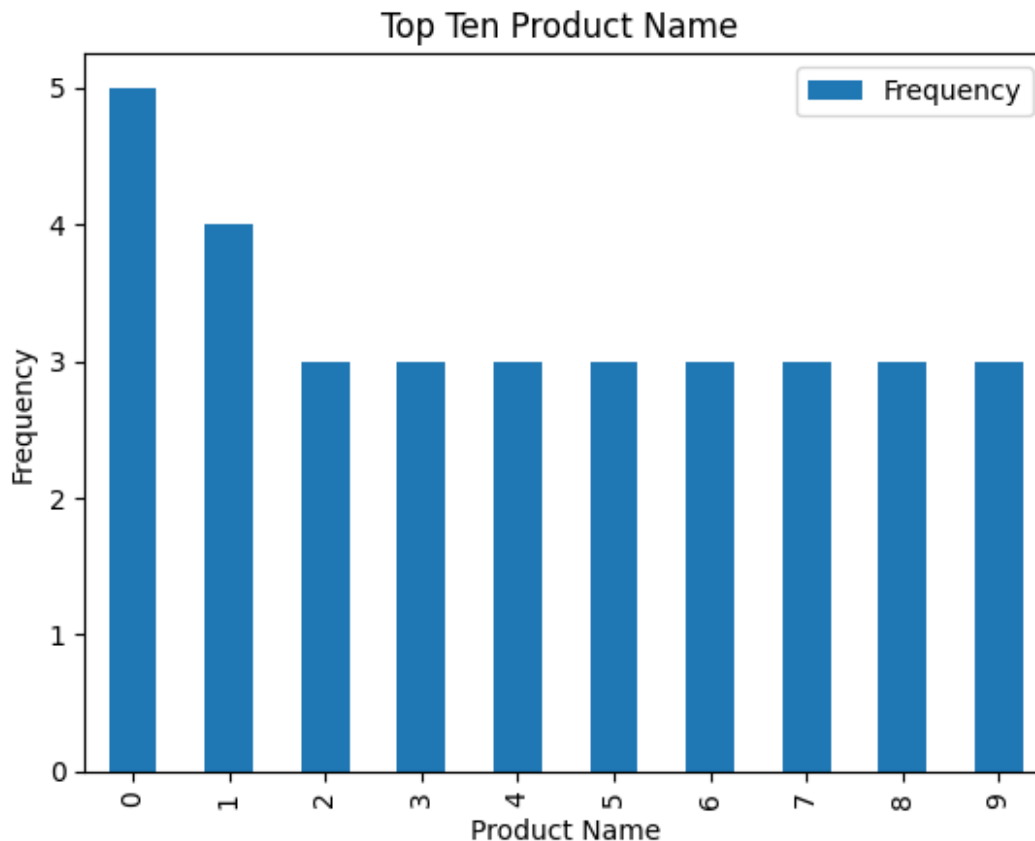
*# OBSERVATION OR INSIGHTS*

```
print("Top ten popular product")
print("The 1st product is more popular than other product so that its frequency is higher than others frequency")
```

Top 10 Most Popular Product Names:

	Product Name	Frequency
0	Fire-Boltt Ninja Call Pro Plus 1.83" Smart Wat...	5
1	Fire-Boltt Phoenix Smart Watch with Bluetooth ...	4
2	Wayona Nylon Braided USB to Lightning Fast Cha...	3
3	MI Braided USB Type-C Cable for Charging Adapt...	3
4	Amazonbasics Nylon Braided Usb-C To Lightning ...	3
5	Samsung Galaxy M13 5G (Aqua Green, 6GB, 128GB ...	3
6	Amazon Basics USB Type-C to USB-A 2.0 Male Fas...	3
7	boAt A400 USB Type-C to USB-A 2.0 Male Data Ca...	3
8	Duracell USB C To Lightning Apple Certified (M...	3
9	AmazonBasics New Release Nylon USB-A to Lightn...	3

<Figure size 800x600 with 0 Axes>



Top ten popular product  
 The 1st product is more popular than other products so that its frequency is higher than others' frequency

## 6. What are the most popular product keywords?

```
from collections import Counter

# Extract keywords from product names
keywords = ' '.join(df['product_name']).split()

# Count the occurrences of each keyword
keyword_counts = Counter(keywords)

# Convert the counter to a DataFrame
keyword_df = pd.DataFrame(keyword_counts.items(), columns=['Keyword', 'Frequency'])

# Sort the DataFrame by frequency in descending order
keyword_df = keyword_df.sort_values(by='Frequency', ascending=False)
```

```
# Print or display the top 10 most popular keywords
print("Top 10 Most Popular Product Keywords:")
print(keyword_df.head(10))
```

Top 10 Most Popular Product Keywords:

	Keyword	Frequency
71	with	718
13	for	614
211		457
43	&	432
3	USB	342
8	and	310
11	Cable	306
113	-	220
7	Charging	212
6	Fast	208

## 7. What are the most popular product reviews?

```
popular_reviews =
df['review_content'].value_counts().reset_index().head(10)
```

```
# Rename the columns for clarity
popular_reviews.columns = ['Review', 'Frequency']
```

```
# Print or display the top 10 most popular product reviews
print("Top 10 Most Popular Product Reviews:")
print(popular_reviews)
```

Top 10 Most Popular Product Reviews:

	Review	Frequency
0	I am not big on camera usage, personally. I wa...	8
1	Worked on iPhone 7 and didn't work on iPhone X...	7
2	I ordered this cable to connect my phone to An...	7
3	Good product,long wire,Charges good,Nice,I bou...	7
4	128GB SD Card is showing 134GBDon't buy this p...	6
5	I purchased the 6/128gb variant.To sum it up ...	6
6	Everything is fine but it is bulky and hard, i...	5
7	Good budget mfi certified lightly cable for th...	5
8	About the TV - Wonderful-----	5
9	[Update: Sept 29] boAt seems to have heard the...	5



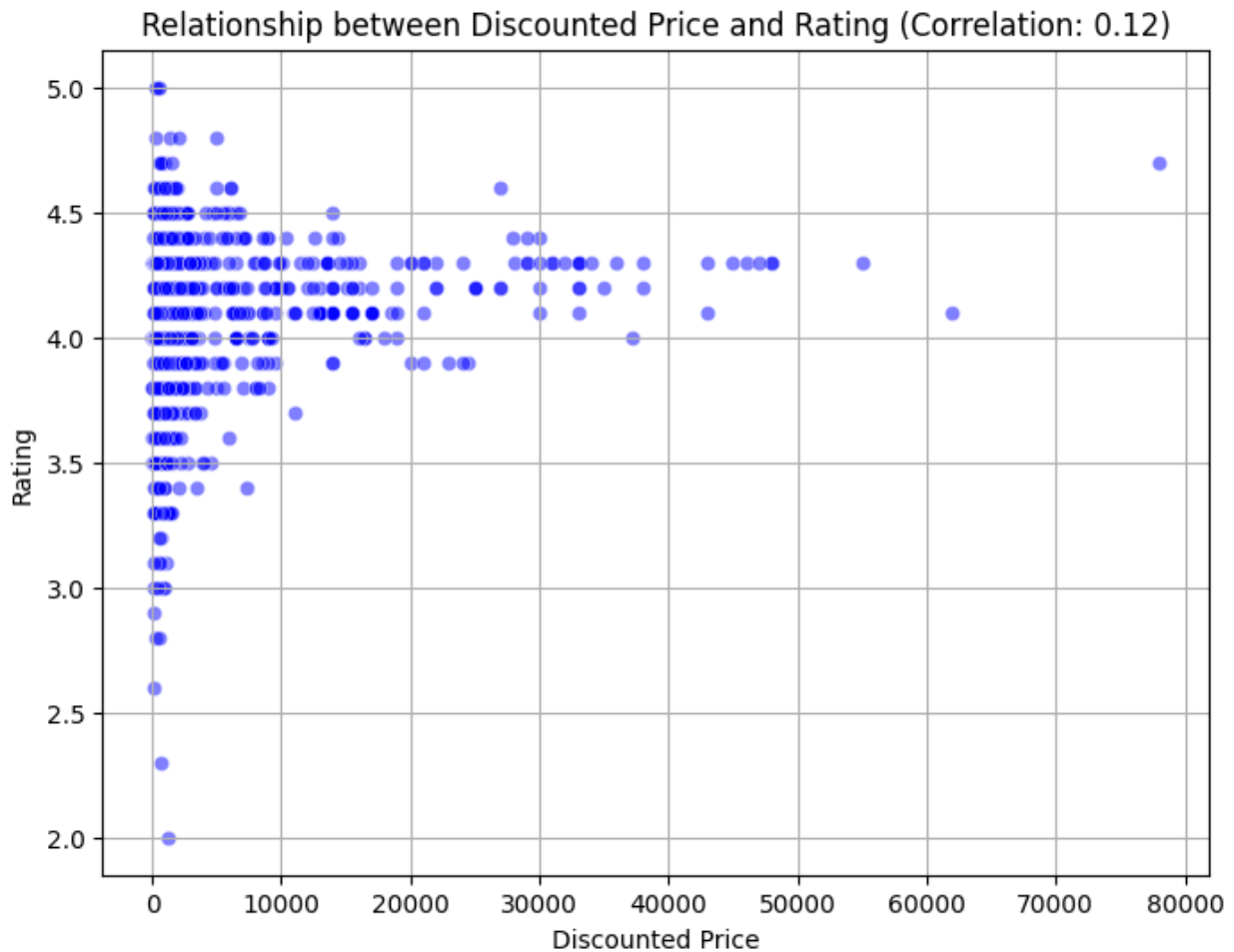
## 8. What is the correlation between discounted\_price and rating?

```
correlation_of_dis_rat = df['discounted_price(₹)'].corr(df['rating'])

# Plotting the relationship between 'discounted_price' and 'rating'
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='discounted_price(₹)', y='rating',
color='blue', alpha=0.5)
plt.title('Relationship between Discounted Price and Rating
(Correlation: {:.2f})'.format(correlation_of_dis_rat))
plt.xlabel('Discounted Price')
plt.ylabel('Rating')
plt.grid(True)
plt.show()

print("Correlation between Discounted Price and Rating:",
correlation_of_dis_rat)
print("So this is positive correlation")

# Observation OR Insights
print("1 indicates a perfect positive correlation\n-1 indicates a
perfect negative correlation, and\n0 indicates no correlation.")
```



Correlation between Discounted Price and Rating: 0.11985513917356677  
 So this is positive correlation  
 1 indicates a perfect positive correlation  
 -1 indicates a perfect negative correlation, and  
 0 indicates no correlation.

## 9. What are the Top 5 categories based on the highest ratings?

```
top_5_category = df.groupby(['category'])
['rating'].sum().reset_index()
top_five_catgeory = top_5_category.sort_values(by= 'rating',
ascending= False).head(5)
print("Top Five Catgeory by rating\n", top_five_catgeory)

# Plotting into the Graph
```

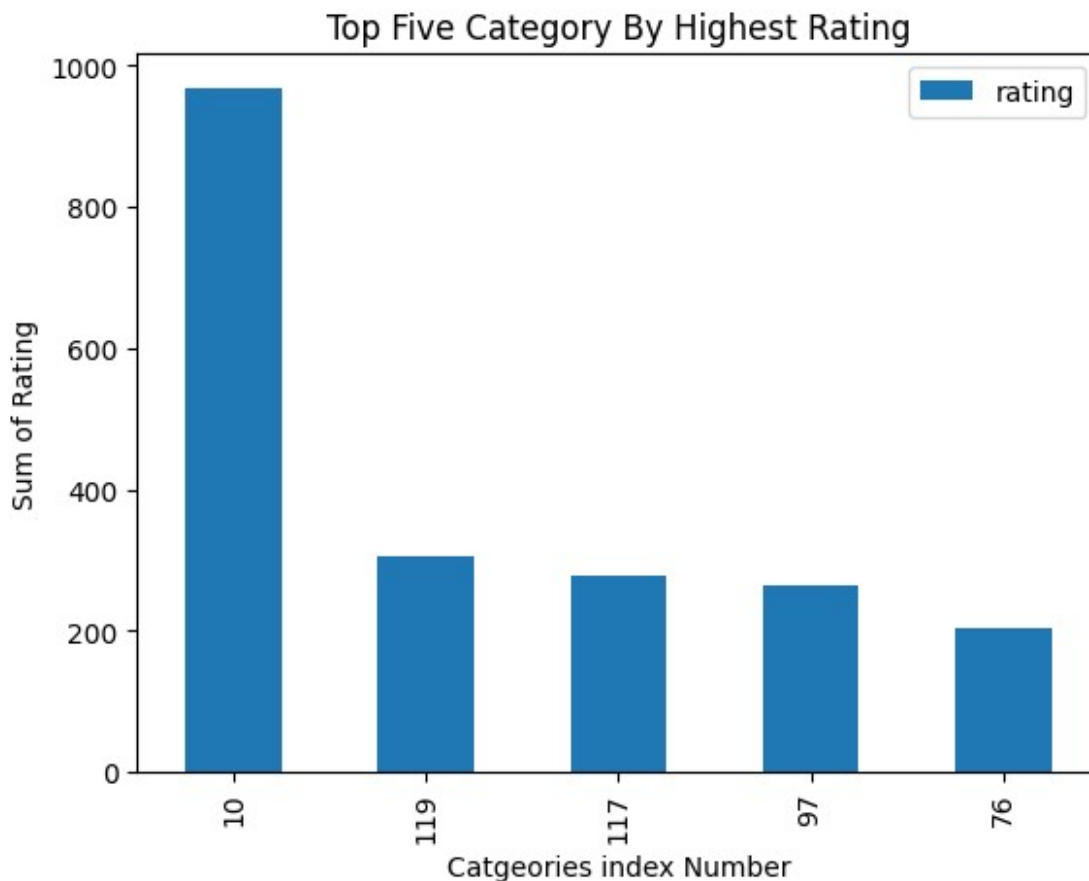
```
plt.figure(figsize= (8,6))
top_five_catgeory.plot(kind= 'bar')
plt.ylabel("Sum of Rating")
plt.xlabel("Catgeories index Number")
plt.title("Top Five Category By Highest Rating")
plt.show()

# Observation Or Insights
print("\nObservation or Insights")
print("You can see the category name by matching index number")
print("You can see that 10 number index have a highest rating")
```

Top Five Catgeory by rating

	category	rating
10	Computers&Accessories Accessories&Peripherals ...	967.4
119	Electronics WearableTechnology SmartWatches	305.9
117	Electronics Mobiles&Accessories Smartphones&Ba...	278.8
97	Electronics HomeTheater,TV&Video Televisions S...	265.2
76	Electronics Headphones,Earbuds&Accessories Hea...	202.7

<Figure size 800x600 with 0 Axes>



### Observation or Insights

You can see the category name by matching index number  
You can see that 10 number index have a highest rating

## 10. Identify any potential areas for improvement or optimization based on the data analysis.

--> : There are two missing values in the 'rating\_count' column. Depending on the significance of this information, you may need to handle these missing values and can be hadle with median, mean.Ensure that the data types are appropriate for each column.

# "Spotify Data: Popular Hip-hop Artists and Tracks" Dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

## 1. Load the dataframe and ensure data quality by checking for missing values and duplicate rows. Handle missing values and remove duplicate rows if necessary.

```
df = pd.read_csv("spotify.csv")
df.head()

{"summary":{"\n  \"name\": \"df\", \n  \"rows\": 440, \n  \"fields\": [\n    {\n      \"column\": \"Artist\", \n      \"properties\": {\n        \"dtype\": \"category\", \n        \"num_unique_values\": 115, \n        \"samples\": [\n          \"Playboi Carti\", \n          \"Nicki Minaj\", \n          \"NEIKED\", \n          ], \n        \"semantic_type\": \"\", \n        \"description\": \"\", \n        }, \n      }, \n      {\n        \"column\": \"Track Name\", \n        \"properties\": {\n          \"dtype\": \"string\", \n          \"num_unique_values\": 412, \n          \"samples\": [\n            \"Shoota (feat. Lil Uzi Vert)\", \n            \"PUFFIN ON ZOOTIEZ\", \n            \"ROCKSTAR (feat. Roddy Ricch)\", \n            ], \n          \"semantic_type\": \"\", \n          \"description\": \"\", \n          }, \n        }, \n        {\n          \"column\": \"Popularity\", \n          \"properties\": {\n            \"dtype\": \"number\", \n            \"std\": 9, \n            \"min\": 29, \n            \"max\": 97, \n            \"num_unique_values\": 51, \n            \"samples\": [\n              35, \n            ]\n          }\n        }\n      ]\n    }\n  ]\n}
```

```
54,\n                52\n                ],\n                \"semantic_type\": \"\",\n                \"description\": \"\"\n            },\n            {\n                \"column\":\n                \"Duration (ms)\",\n                \"properties\": {\n                    \"dtype\":\n                    \"number\",\n                    \"std\": 53576,\n                    \"min\": 81666,\n                    \"max\": 501648,\n                    \"num_unique_values\": 410,\n                    \"samples\": [\n                        203894,\n                        225905,\n                        213593\n                    ],\n                    \"semantic_type\": \"\",\n                    \"description\": \"\"\n                },\n                {\n                    \"column\":\n                    \"Track ID\",\n                    \"properties\": {\n                        \"dtype\":\n                        \"string\",\n                        \"num_unique_values\": 413,\n                        \"samples\":\n                        [\n                            \"50ceCGZ3oD3U5caQV5bP6f\",\n                            \"4Lw0rnuxJwR7C5Sw4liY4Z\",\n                            \"1c7MITQmNJTrvfbDSzWT6x\"\n                        ],\n                        \"semantic_type\": \"\",\n                        \"description\": \"\"\n                    }\n                }\n            ],\n            \"type\": \"dataframe\", \"variable name\": \"df\"}
```

```
print(df['Artist'].isna().value_counts(), "\n", df['Track  
Name'].isnull().value_counts())  
print(df['Popularity'].isnull().value_counts(), "\n", df['Duration  
(ms)'].isna().value_counts(), "\n", df['Track  
ID'].isnull().value_counts())
```

```
# Check for duplicate rows
duplicate_rows = df.duplicated()
print("Duplicate Rows:")
print(duplicate_rows.sum())
```

```
# Remove duplicate rows
df.drop_duplicates(inplace=True)
```

```
# Observation and Insights
print("As you can see that there is no any missing values and it
contain 27 duplicate ")
print("So we drop the duplicate row and now there is 413 rows")
```

```
Artist
False      440
Name: count, dtype: int64
Track Name
False      440
Name: count, dtype: int64
Popularity
False      440
Name: count, dtype: int64
Duration (ms)
False      440
Name: count, dtype: int64
Track ID
False      440
Name: count, dtype: int64
```



Duplicate Rows:

27

As you can see that there is no any missing values and it contain 27 duplicate

So we drop the duplicate row and now there is 413 rows

## 2. What is the distribution of popularity among the tracks in the dataset? Visualize it using a histogram.

```
# Extract Popularity Data
```

```
popularity_data = df['Popularity']
```

```
# Plot Histogram
```

```
plt.figure(figsize=(10, 6))
```

```
sns.histplot(popularity_data, bins=20, color='skyblue',  
edgecolor='black')
```

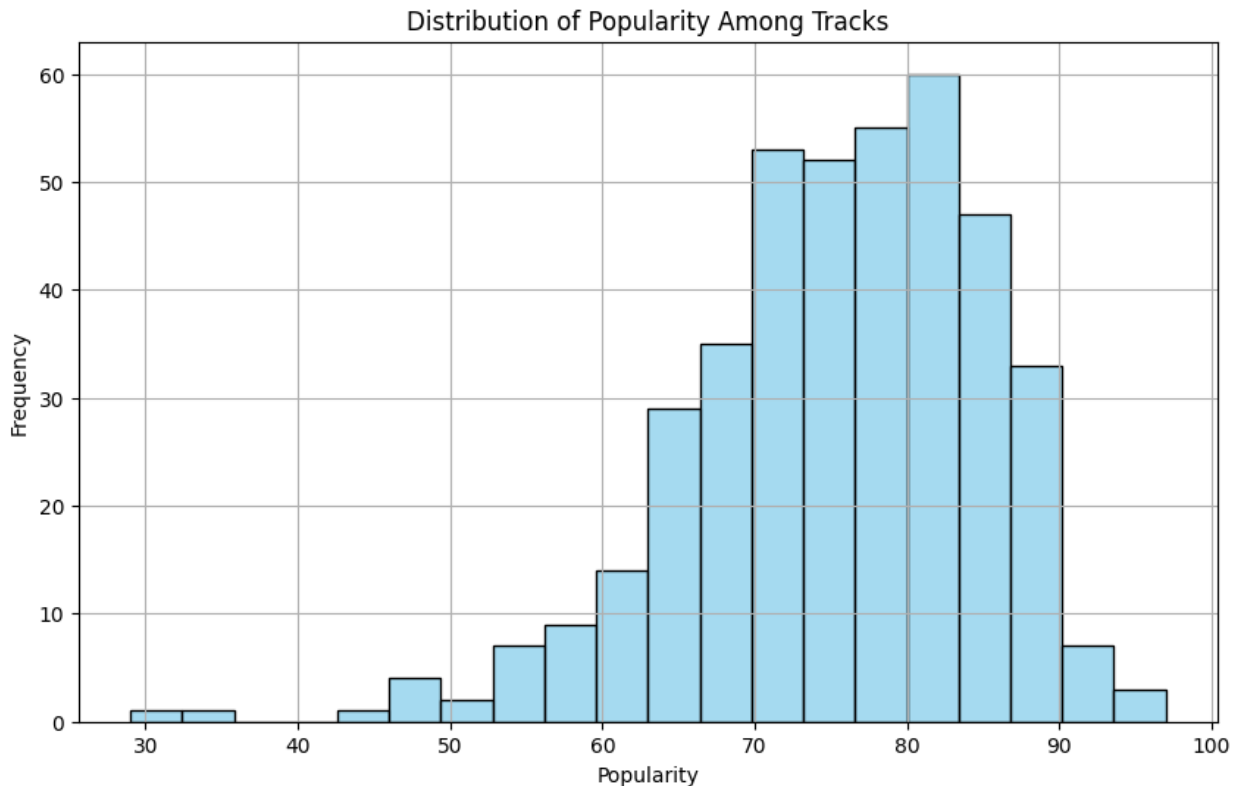
```
plt.title('Distribution of Popularity Among Tracks')
```

```
plt.xlabel('Popularity')
```

```
plt.ylabel('Frequency')
```

```
plt.grid(True)
```

```
plt.show()
```



3. Is there any relationship between the popularity and the duration of tracks? Explore this using a scatter plot.

```
import plotly.express as px
fig = px.scatter(df, x='Popularity', y='Duration (ms)', title =
"Relation Between Popularity and Duration of
Tracks", labels={"Popularity": "Popularity", "Duration (ms)": "Duration
of Track"})
```

```
fig.show()
```

*# observation Or Insights*

```
print("Observation Or Insights")
```

```
print("We can see that popularity is between(65 - 90) when duration of
track lies (100k to 300k)ms ")
```

Observation Or Insights

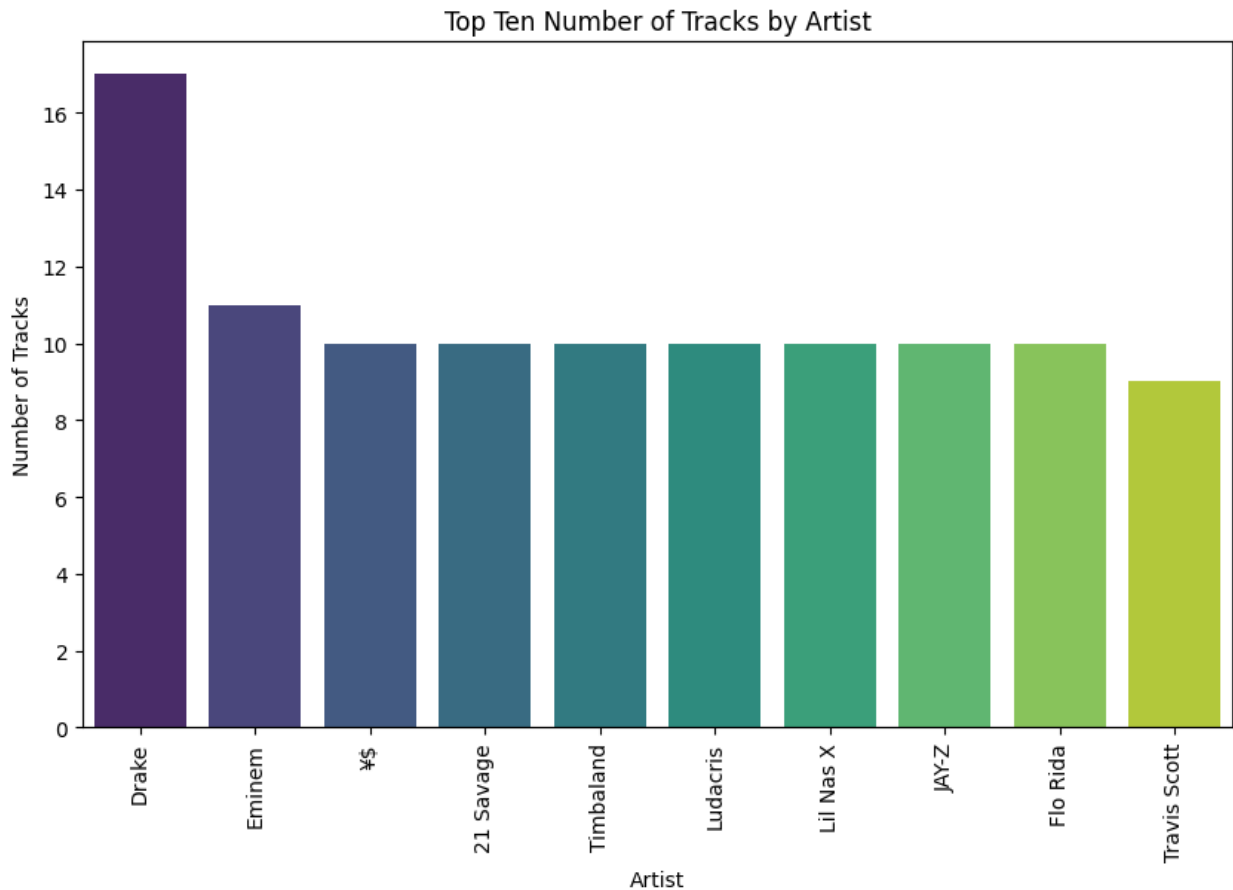
We can see that popularity is between(65 - 90) when duration of track lies (100k to 300k)ms

4. Which artist has the highest number of tracks in the dataset? Display the count of tracks for each artist using a countplot.

```
# Group the DataFrame by 'Artist' and count the number of tracks for each artist
artist_track_count =
df.groupby('Artist').size().reset_index(name='Track Count')

# Sort the artists based on the count of tracks
artist_track_count_sorted = artist_track_count.sort_values(by='Track Count', ascending=False).head(10)

# Plot the count of tracks for each artist using a countplot
plt.figure(figsize=(10, 6))
sns.barplot(data=artist_track_count_sorted, x='Artist', y='Track Count', palette='viridis')
plt.title('Top Ten Number of Tracks by Artist')
plt.xlabel('Artist')
plt.ylabel('Number of Tracks')
plt.xticks(rotation=90)
plt.show()
```



5. What are the top 5 least popular tracks in the dataset? Provide the artist name and track name for each.

```
# Sort the DataFrame by 'Popularity' column in ascending order
least_popular_tracks = df.sort_values(by='Popularity', ascending=True)

# Select the top 5 least popular tracks
top_5_least_popular_tracks = least_popular_tracks.head(5)

# Display the artist name and track name for each of these tracks
for index, row in top_5_least_popular_tracks.iterrows():
    print("Artist:", row['Artist'])
    print("Track Name:", row['Track Name'])
    print()

# Plotting the top 5 least popular tracks
plt.figure(figsize=(10, 6))
```

```
plt.barh(top_5_least_popular_tracks['Track Name'],
top_5_least_popular_tracks['Popularity'], color='skyblue')
plt.xlabel('Popularity')
plt.ylabel('Track Name')
plt.title('Top 5 Least Popular Tracks')
plt.gca().invert_yaxis() # Invert y-axis to display the track with
the highest popularity at the top
plt.show()
```

Artist: Pressa

Track Name: Attachments (feat. Coi Leray)

Artist: Justin Bieber

Track Name: Intentions

Artist: French Montana

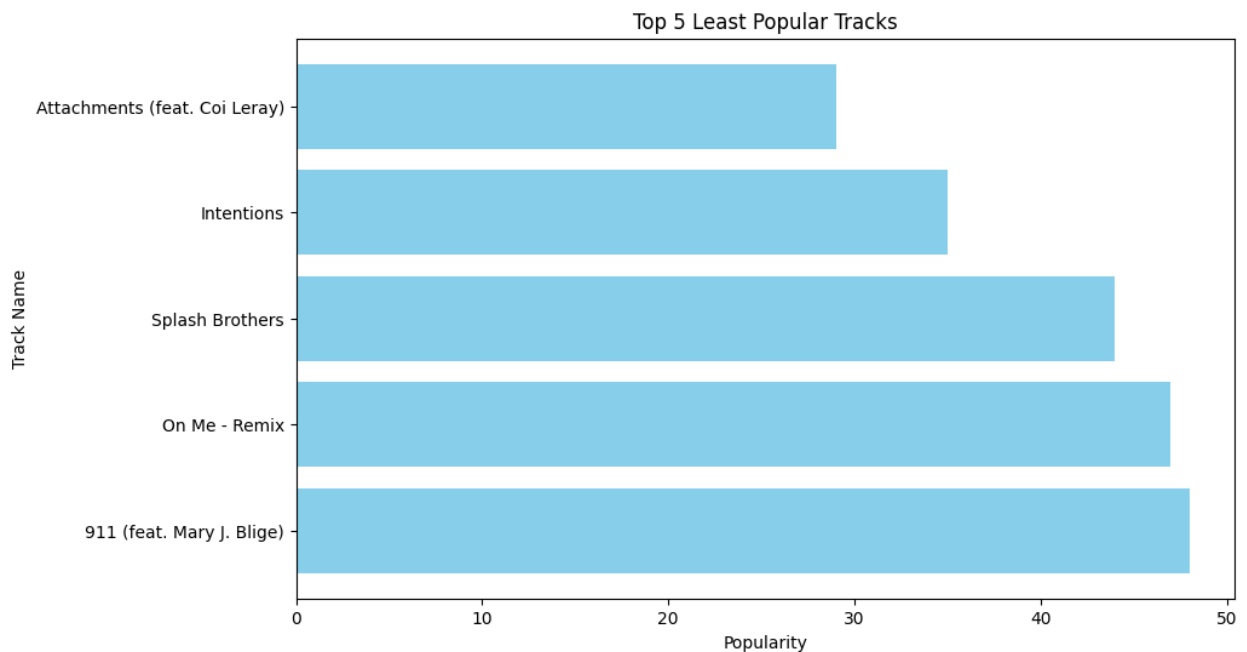
Track Name: Splash Brothers

Artist: Lil Baby

Track Name: On Me - Remix

Artist: Wyclef Jean

Track Name: 911 (feat. Mary J. Blige)





6. Among the top 5 most popular artists, which artist has the highest popularity on average? Calculate and display the average popularity for each artist.

```
# Get the top 5 most popular artists
top_5_most_popular_artists = df.groupby('Artist')
['Popularity'].mean().nlargest(5)

# Display the average popularity for each artist
print("Average Popularity for Each Artist:")
print(top_5_most_popular_artists)

# Find the artist with the highest average popularity
artist_highest_avg_popularity = top_5_most_popular_artists.idxmax()

# Display the artist with the highest average popularity
print("\nArtist with the Highest Average Popularity:",
      artist_highest_avg_popularity)

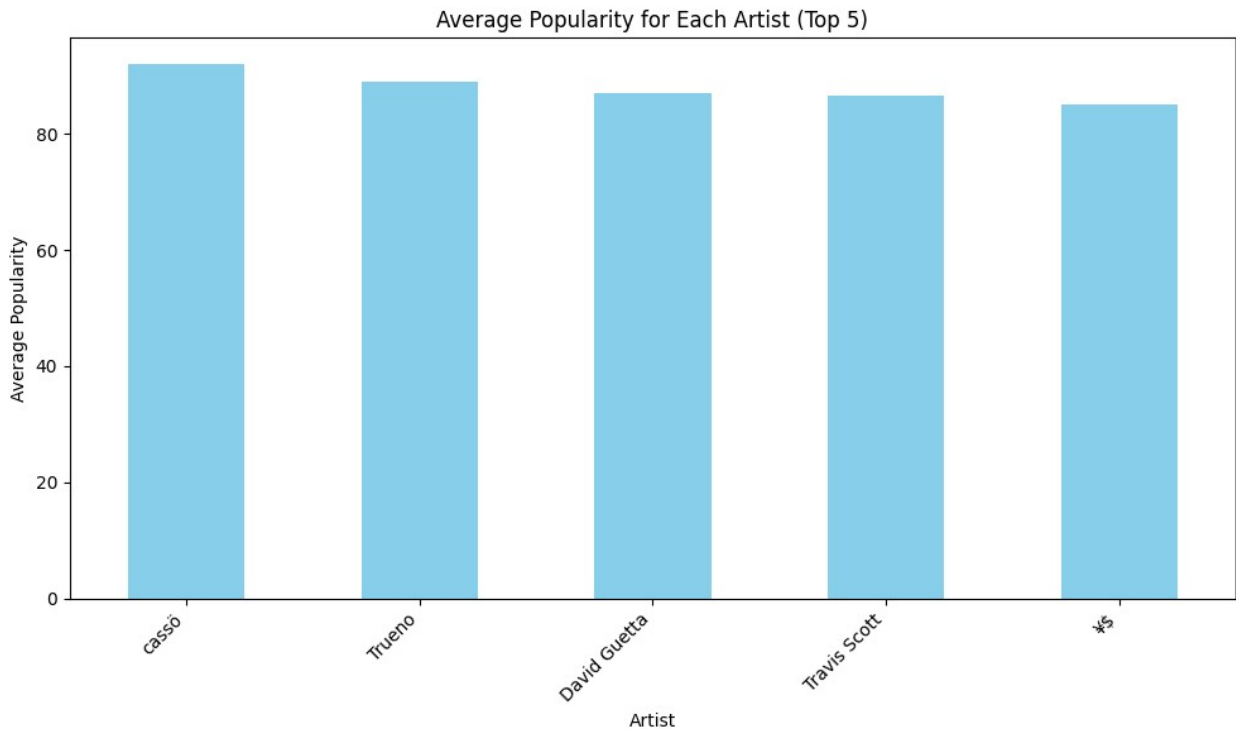
# Plotting into gaph
# Plotting the average popularity for each artist
plt.figure(figsize=(10, 6))
top_5_most_popular_artists.plot(kind='bar', color='skyblue')
plt.title('Average Popularity for Each Artist (Top 5)')
plt.xlabel('Artist')
plt.ylabel('Average Popularity')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

Average Popularity for Each Artist:

Artist	
cassö	92.000000
Trueno	89.000000
David Guetta	87.000000
Travis Scott	86.555556
¥\$	85.100000

Name: Popularity, dtype: float64

Artist with the Highest Average Popularity: cassö



7. For the top 5 most popular artists, what are their most popular tracks? List the track name for each artist

```
# Get the top 5 most popular artists
top_5_most_popular_artists = df.groupby('Artist')
['Popularity'].mean().nlargest(5).index

# Filter the DataFrame to include only the tracks of the top 5 most
popular artists
top_5_tracks = df[df['Artist'].isin(top_5_most_popular_artists)]

# Group the filtered DataFrame by 'Artist' and find the most popular
track for each artist
most_popular_tracks = top_5_tracks.groupby('Artist').apply(lambda x:
x.loc[x['Popularity'].idxmax()])

# List the track name for each artist
for artist, track_data in most_popular_tracks[['Artist', 'Track
Name']].iterrows():
    print("Artist:", track_data['Artist'])
    print("Most Popular Track:", track_data['Track Name'])
    print()
```

```

# Plotting the most popular tracks for each artist
plt.figure(figsize=(10, 6))
for artist, track_data in most_popular_tracks[['Artist', 'Track
Name']].iterrows():
    plt.barh(track_data['Artist'], track_data['Track Name'],
color='skyblue')

plt.xlabel('Track Name')
plt.ylabel('Artist')
plt.title('Most Popular Track for Each Artist (Top 5)')
plt.gca().invert_yaxis() # Invert y-axis to display the artist with
the highest popularity at the top
plt.show()

```

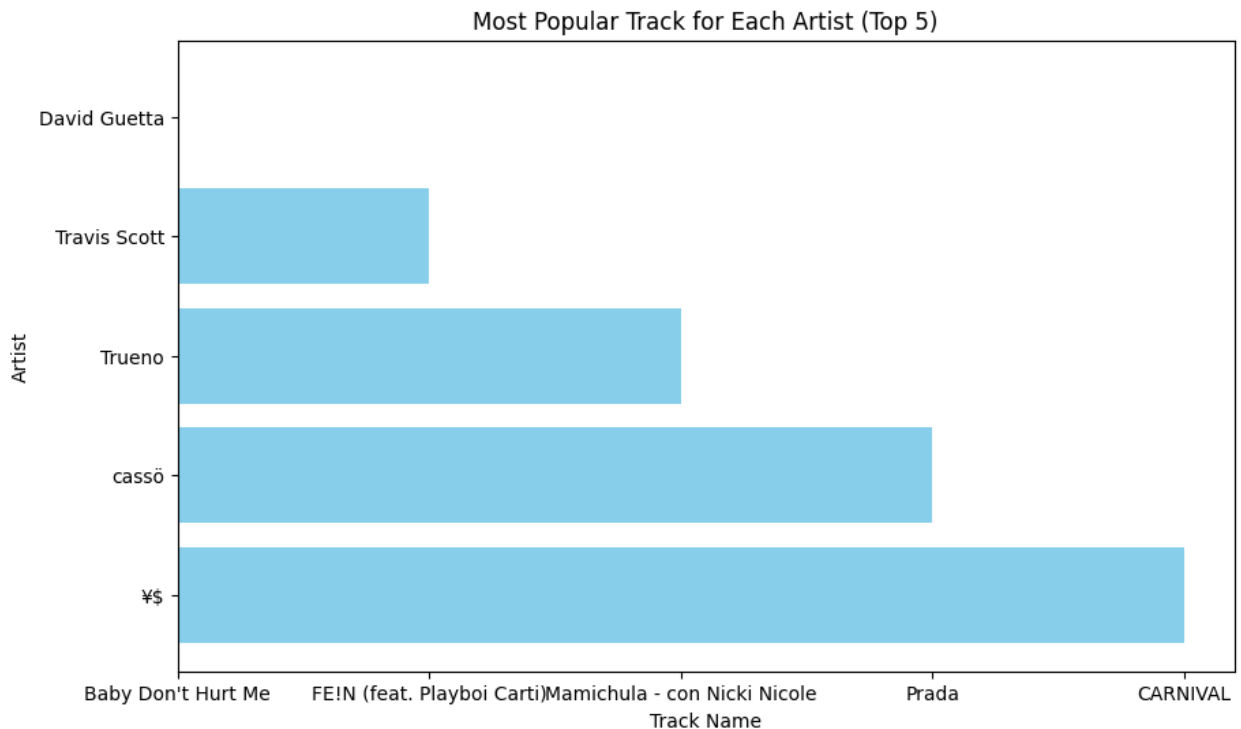
Artist: David Guetta  
Most Popular Track: Baby Don't Hurt Me

Artist: Travis Scott  
Most Popular Track: FE!N (feat. Playboi Carti)

Artist: Trueno  
Most Popular Track: Mamichula - con Nicki Nicole

Artist: cassö  
Most Popular Track: Prada

Artist: ¥\$  
Most Popular Track: CARNIVAL

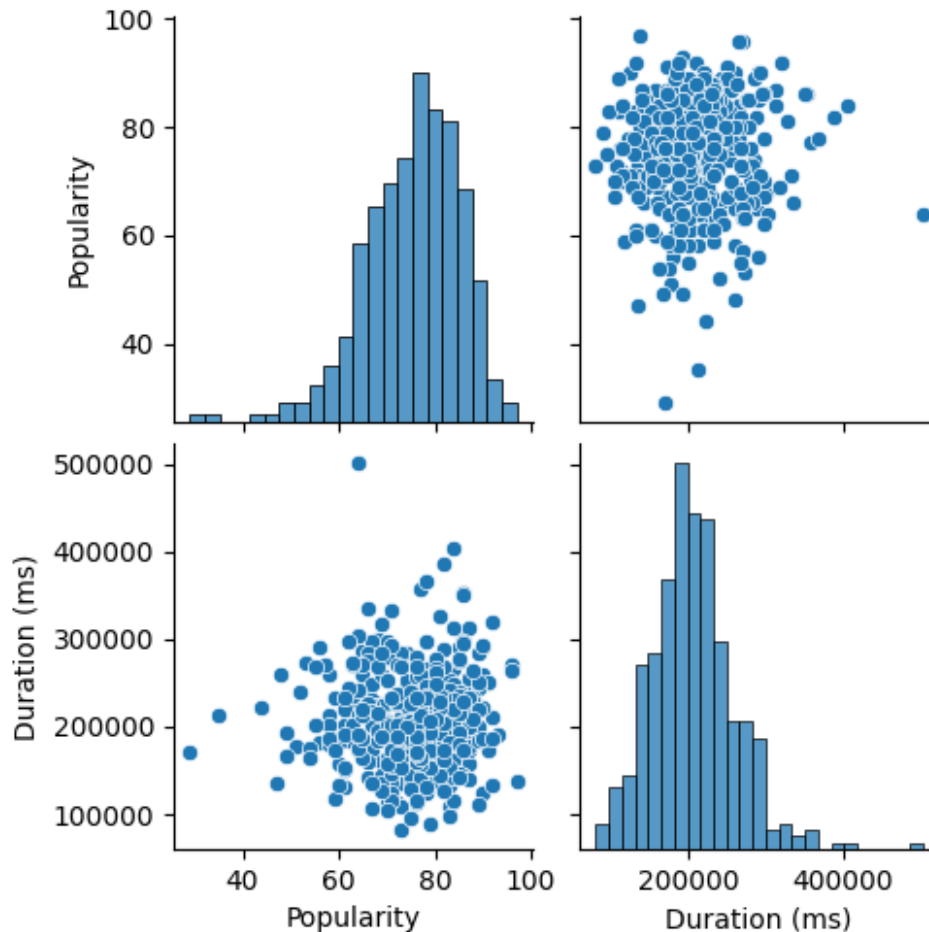


8. Visualize relationships between multiple numerical variables simultaneously using a pair plot.

```
import seaborn as sns
import matplotlib.pyplot as plt

# Selecting numerical variables for pair plotting
numerical_vars = df.select_dtypes(include=['float64', 'int64'])

# Creating a pair plot
sns.pairplot(numerical_vars)
plt.show()
```

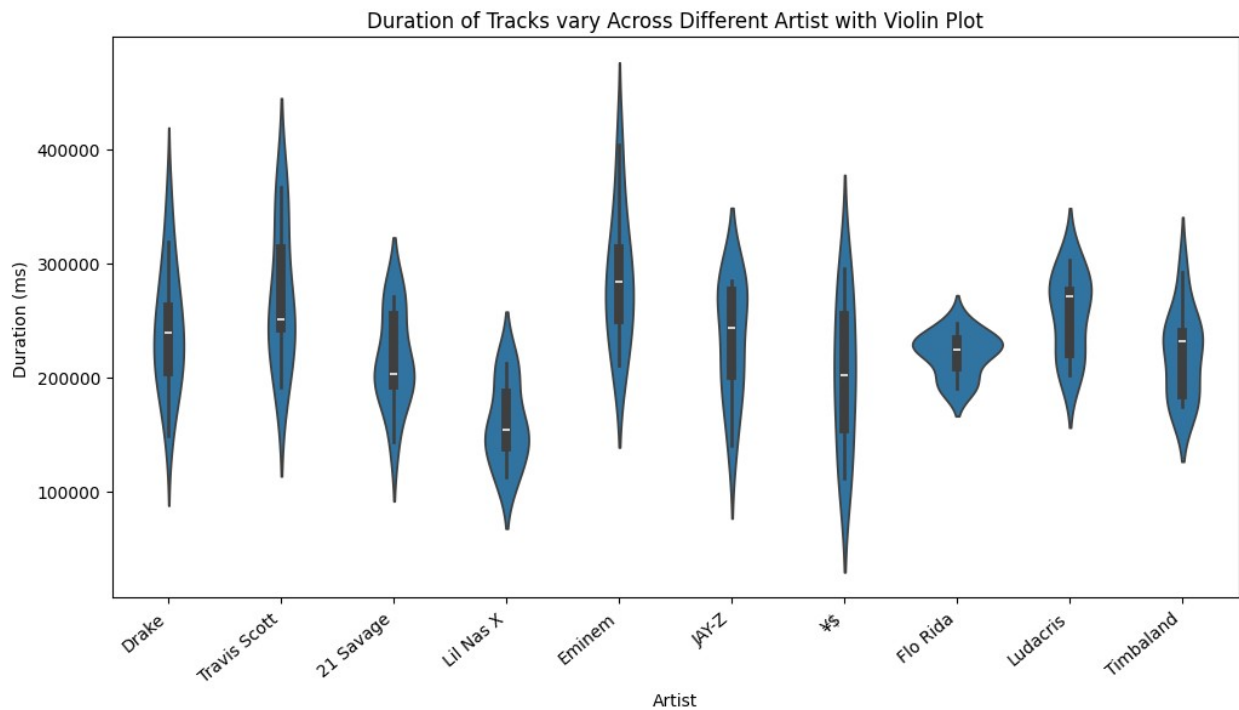


9. Does the duration of tracks vary significantly across different artists? Explore this visually using a box plot or violin plot.

```
top_artist = df['Artist'].value_counts().head(10).index
print(top_artist)
top_artist1 = df[df['Artist'].isin(top_artist)]

plt.figure(figsize= (12,6))
sns.violinplot(data= top_artist1, x= 'Artist', y= 'Duration (ms)')
plt.title("Duration of Tracks vary Across Different Artist with Violin Plot")
plt.xlabel("Artist")
plt.ylabel("Duration (ms)")
plt.xticks(rotation= 40, ha= 'right')
plt.show()
```

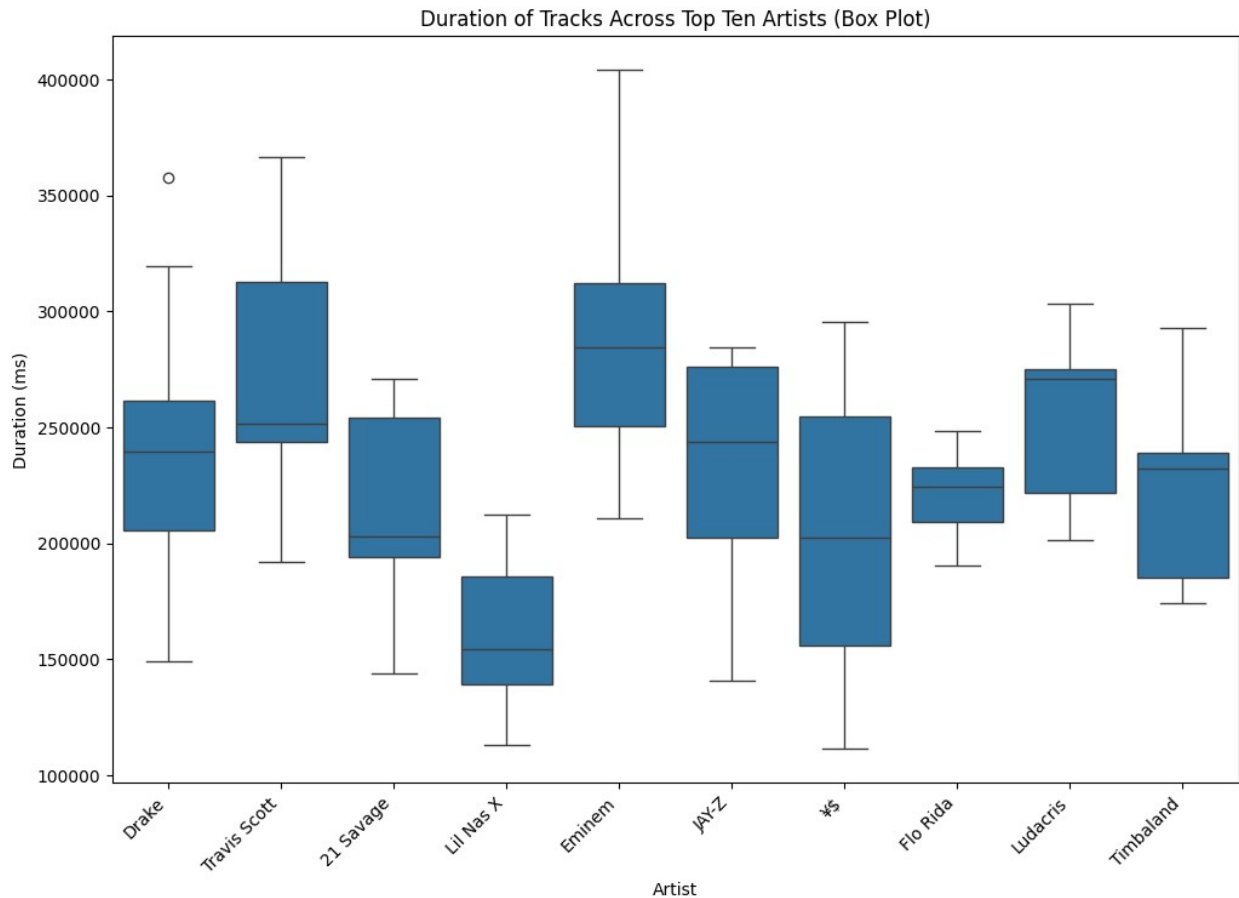
```
Index(['Drake', 'Eminem', 'Flo Rida', 'Ludacris', 'Timbaland', '¥$',
      '21 Savage', 'JAY-Z', 'Lil Nas X', 'Travis Scott'],
      dtype='object', name='Artist')
```



```
# Get the top ten artists
top_ten_artists = df['Artist'].value_counts().nlargest(10).index

# Filter the DataFrame to include only tracks by the top ten artists
top_ten_tracks = df[df['Artist'].isin(top_ten_artists)]

# Create a box plot or violin plot
plt.figure(figsize=(12, 8))
sns.boxplot(data=top_ten_tracks, x='Artist', y='Duration (ms)')
plt.title('Duration of Tracks Across Top Ten Artists (Box Plot)')
plt.xticks(rotation=45, ha='right')
plt.xlabel('Artist')
plt.ylabel('Duration (ms)')
plt.show()
```



10. How does the distribution of track popularity vary for different artists? Visualize this using a swarm plot or a violin plot.

```
top_track = df.groupby('Artist')
['Popularity'].sum().nlargest(10).index

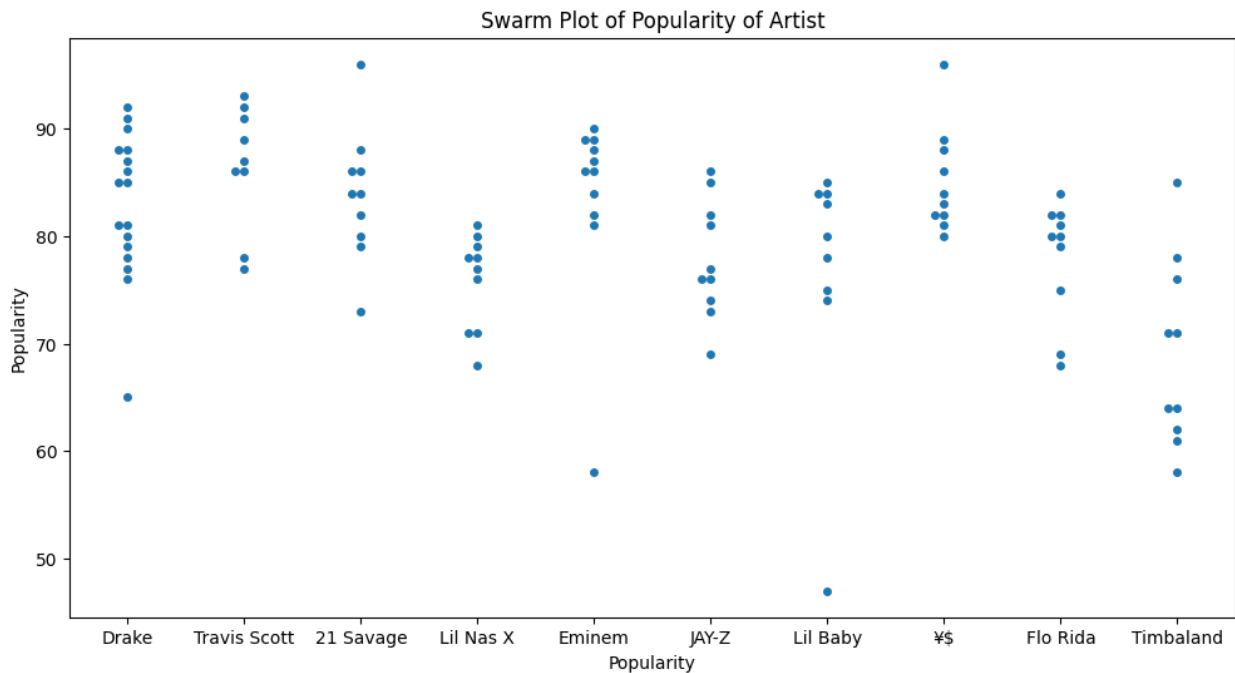
top_ten_track = df[df['Artist'].isin(top_track)]

plt.figure(figsize= (12,6))
sns.swarmplot(data= top_ten_track, x= 'Artist', y= 'Popularity')
plt.title("Swarm Plot of Popularity of Artist")
plt.xlabel("Artist")
plt.ylabel("Popularity")
plt.show()

# Observation Or Insights
print("\nObservation Or Insights")
print("In the swarm plot, each point represents a track's popularity")
```

```
for a specific artist, providing a detailed view of the
distribution.")
```

```
print("Artists with more scattered swarm plots may have a broader
range of track popularity, while artists with narrower violins or
tightly clustered swarm plots may have more consistent track
popularity.")
```



### Observation Or Insights

In the swarm plot, each point represents a track's popularity for a specific artist, providing a detailed view of the distribution. Artists with more scattered swarm plots may have a broader range of track popularity, while artists with narrower violins or tightly clustered swarm plots may have more consistent track popularity.