

Importing the Library

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import datetime
import calendar
```

Loading the dataset as dataframe

In [2]:

```
df= pd.read_csv('S:\eda and sql\Google-Playstore.csv')
```

In [3]:

```
rows=df.shape[0]
coloumns=df.shape[1]
```

Printing the number of rows and the column in the dataset

In [4]:

```
print('there are {} rows and {} columns in the dataset'.format(rows,coloumns))
print(df.shape)
```

```
there are 2312944 rows and 24 columns in the dataset
(2312944, 24)
```

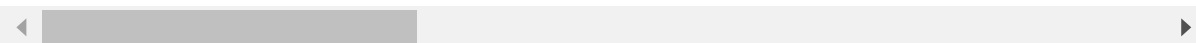
In [5]:

```
df.head(5)
```

Out[5]:

	App Name	App Id	Category	Rating	Rating Count	Installs	Minimum Installs	Maximum Installs
0	Gakondo	com.ishakwe.gakondo	Adventure	0.0	0.0	10+	10.0	
1	Ampere Battery Info	com.webserveis.batteryinfo	Tools	4.4	64.0	5,000+	5000.0	
2	Vibook	com.doantiepvien.crm	Productivity	0.0	0.0	50+	50.0	
3	Smart City Trichy Public Service Vehicles 17UC...	cst.stJoseph.ug17ucs548	Communication	5.0	5.0	10+	10.0	
4	GROW.me	com.horodyski.grower	Tools	0.0	0.0	100+	100.0	

5 rows × 9 columns



In [6]:

```
df.columns = [c.replace(' ', '_') for c in df.columns]
```

Finding the null values in the dataset

In [7]:

```
df.isnull().sum()
```

Out[7]:

App_Name	2
App_Id	0
Category	0
Rating	22883
Rating_Count	22883
Installs	107
Minimum_Installs	107
Maximum_Installs	0
Free	0
Price	0
Currency	135
Size	196
Minimum_Android	6530
Developer_Id	33
Developer_Website	760835
Developer_Email	31
Released	71053
Last_Updated	0
Content_Rating	0
Privacy_Policy	420953
Ad_Supported	0
In_App_Purchases	0
Editors_Choice	0
Scraped_Time	0

dtype: int64

In [8]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2312944 entries, 0 to 2312943
Data columns (total 24 columns):
#   Column                Dtype
---  -
0   App_Name              object
1   App_Id                object
2   Category              object
3   Rating                float64
4   Rating_Count          float64
5   Installs              object
6   Minimum_Installs      float64
7   Maximum_Installs      int64
8   Free                  bool
9   Price                 float64
10  Currency              object
11  Size                  object
12  Minimum_Android        object
13  Developer_Id           object
14  Developer_Website      object
15  Developer_Email        object
16  Released               object
17  Last_Updated           object
18  Content_Rating         object
19  Privacy_Policy         object
20  Ad_Supported           bool
21  In_App_Purchases       bool
22  Editors_Choice         bool
23  Scraped_Time           object
dtypes: bool(4), float64(4), int64(1), object(15)
memory usage: 361.8+ MB
```

In [9]:

```
df.describe()
```

Out[9]:

	Rating	Rating_Count	Minimum_Installs	Maximum_Installs	Price
count	2.290061e+06	2.290061e+06	2.312837e+06	2.312944e+06	2.312944e+06
mean	2.203152e+00	2.864839e+03	1.834452e+05	3.202017e+05	1.034992e-01
std	2.106223e+00	2.121626e+05	1.513144e+07	2.355495e+07	2.633127e+00
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	0.000000e+00	0.000000e+00	5.000000e+01	8.400000e+01	0.000000e+00
50%	2.900000e+00	6.000000e+00	5.000000e+02	6.950000e+02	0.000000e+00
75%	4.300000e+00	4.200000e+01	5.000000e+03	7.354000e+03	0.000000e+00
max	5.000000e+00	1.385576e+08	1.000000e+10	1.205763e+10	4.000000e+02

In [10]:

```
df.tail(20)
```

Out[10]:

	App_Name	App_Id	Category	Rating	Rating_C
2312924	How to Make Money From Website	com.rojas.htmkmmfst	Books & Reference	0.0	
2312925	GOKarli Carrera Rennbahn und Dart Onlineshop	de.gokarli.shop	Shopping	0.0	
2312926	Asset Data Collection	com.linq.assetdatacollection	Business	0.0	
2312927	Gear Ratio Calculator	jp.abt.lopnur.gearratioalc	Sports	0.0	
2312928	Nutrishop Boise Meridian	com.tapmango.nutrishopbm	Health & Fitness	0.0	
2312929	Murottal Muh Tha al Junayd Offline	com.andromo.dev471643.app436193	Entertainment	4.4	
2312930	PAX	com.apreciasoft.mobile.pax	Travel & Local	3.3	
2312931	Driving Day	com.day.drivingday	Entertainment	2.8	10
2312932	Hopeless 2: Cave Escape	com.upopa.hopeless2	Action	4.3	1034
2312933	Caustic Editor for VolcaSample	com.singlecellsoftware.kvsampler	Music & Audio	4.0	;
2312934	Vietnamese - English Translator	com.eliminatesapps.vietnamesetranslator	Education	0.0	
2312935	Floral Wallpaper	com.arfdev.floralwallpaper	Personalization	0.0	
2312936	Engineers Careers	com.eventapps.eventapps	Business	0.0	
2312937	STMIK Mercusuar - Aditya Rachman	com.aplikasi.datapribadiadit	Education	0.0	
2312938	Lero TOEFL Recorder + Timer	com.toefltimer	Education	3.4	
2312939	大俠客—熱血歸來	com.rxsj.ssjj	Role Playing	4.3	16
2312940	ORU Online	com.threedream.oruonline	Education	0.0	

	App_Name	App_Id	Category	Rating	Rating_C
2312941	Data Structure	datastructure.appoworld.datastructure	Education	0.0	
2312942	Devi Suktam	ishan.devi.suktam	Music & Audio	3.5	
2312943	Biliyor Musun - Sonsuz Yariş	com.yyazilim.biliyormusun	Trivia	5.0	

20 rows × 24 columns

In [11]:

```
df.columns
```

Out[11]:

```
Index(['App_Name', 'App_Id', 'Category', 'Rating', 'Rating_Count', 'Installs',
      'Minimum_Installs', 'Maximum_Installs', 'Free', 'Price', 'Currency',
      'Size', 'Minimum_Android', 'Developer_Id', 'Developer_Website',
      'Developer_Email', 'Released', 'Last_Updated', 'Content_Rating',
      'Privacy_Policy', 'Ad_Supported', 'In_App_Purchases', 'Editors_Choice',
      'Scraped_Time'],
      dtype='object')
```

In [12]:

```
pd.set_option('display.max_columns', None)
df.sample(5)
```

Out[12]:

	App_Name	App_Id	Category	Rating	Rating_Count	Inst
2307225	Prontmed	br.com.prontmed	Health & Fitness	2.8	54.0	1,0
172260	BabyNite	com.sleepace.hrbrid.babynite	Health & Fitness	0.0	0.0	
280681	Línea 130 Tu Voz Si Cuenta	com.shani123.linea	Social	0.0	0.0	
489710	Baby Names PT	com.classicosdeleitura.Baby_Names_PT	Parenting	0.0	0.0	
1995182	Cofonder	com.cofonder	Social	0.0	0.0	1

Looking for unique values in category column

In [13]:

```
pd.unique(df['Category'])
```

Out[13]:

```
array(['Adventure', 'Tools', 'Productivity', 'Communication', 'Social',  
      'Libraries & Demo', 'Lifestyle', 'Personalization', 'Racing',  
      'Maps & Navigation', 'Travel & Local', 'Food & Drink',  
      'Books & Reference', 'Medical', 'Puzzle', 'Entertainment',  
      'Arcade', 'Auto & Vehicles', 'Photography', 'Health & Fitness',  
      'Education', 'Shopping', 'Board', 'Music & Audio', 'Sports',  
      'Beauty', 'Business', 'Educational', 'Finance', 'News & Magazines',  
      'Casual', 'Art & Design', 'House & Home', 'Card', 'Events',  
      'Trivia', 'Weather', 'Strategy', 'Word', 'Video Players & Editors',  
      'Action', 'Simulation', 'Music', 'Dating', 'Role Playing',  
      'Casino', 'Comics', 'Parenting'], dtype=object)
```

Defining the function to view the insights of the dataset

In [14]:

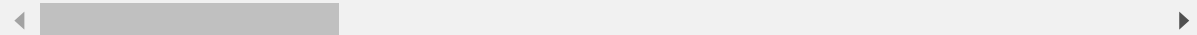
```
def printinfo():  
    temp = pd.DataFrame(index=df.columns)  
    temp['dataType'] = df.dtypes  
    temp['nullCount']=df.isnull().sum()  
    temp['uniqueCount']=df.nunique()  
    return temp
```

In [15]:

```
df.head(1)
```

Out[15]:

	App_Name	App_Id	Category	Rating	Rating_Count	Installs	Minimum_Installs
0	Gakondo	com.ishakwe.gakondo	Adventure	0.0	0.0	10+	10.0



In [16]:

```
printinfo()
```

Out[16]:

	dataType	nullCount	uniqueCount
App_Name	object	2	2177944
App_Id	object	0	2312944
Category	object	0	48
Rating	float64	22883	42
Rating_Count	float64	22883	38482
Installs	object	107	22
Minimum_Installs	float64	107	22
Maximum_Installs	int64	0	251563
Free	bool	0	2
Price	float64	0	1063
Currency	object	135	15
Size	object	196	1657
Minimum_Android	object	6530	154
Developer_Id	object	33	758371
Developer_Website	object	760835	810440
Developer_Email	object	31	950456
Released	object	71053	4158
Last_Updated	object	0	3918
Content_Rating	object	0	6
Privacy_Policy	object	420953	977743
Ad_Supported	bool	0	2
In_App_Purchases	bool	0	2
Editors_Choice	bool	0	2
Scraped_Time	object	0	67374

In [17]:

```
df.isnull().sum()
```

Out[17]:

App_Name	2
App_Id	0
Category	0
Rating	22883
Rating_Count	22883
Installs	107
Minimum_Installs	107
Maximum_Installs	0
Free	0
Price	0
Currency	135
Size	196
Minimum_Android	6530
Developer_Id	33
Developer_Website	760835
Developer_Email	31
Released	71053
Last_Updated	0
Content_Rating	0
Privacy_Policy	420953
Ad_Supported	0
In_App_Purchases	0
Editors_Choice	0
Scraped_Time	0
dtype:	int64

Finding the rows and columns in Rating having the Null values

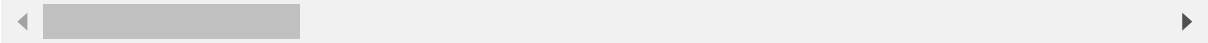
In [18]:

```
df[df['Rating'].isnull()]
```

Out[18]:

	App_Name	App_Id	Category	Rating	Rat
115	VM-Delay	com.irv.vm_delay	Tools	NaN	
210	Promotor	com.touchesbegan.promotor	Productivity	NaN	
284	xrsta xatr	com.xrsta.xatr	Entertainment	NaN	
501	GridChats	com.thegridnet.conference	Communication	NaN	
662	Restaurant POS(Admin)	org.wisdomfish.posadmin	Tools	NaN	
...	
2312553	Merlins Idle Apelsin	com.elitegamesltd.merlinsidle	Simulation	NaN	
2312712	Joule Mobile App	com.companyname.Joule.Xamarin	Productivity	NaN	
2312751	Iris Profissional	com.sys4web.irisprofissional	Health & Fitness	NaN	
2312764	Wool Sort Puzzle	com.cla.wool.ballsort.puzzle	Puzzle	NaN	
2312842	Living Goods (Innovation Network)	org.medicmobile.webapp.mobile.livinggoods_inno...	Medical	NaN	

22883 rows × 24 columns



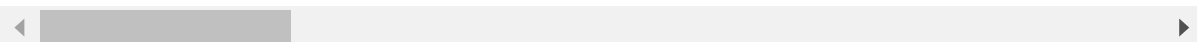
In [19]:

```
df.loc[34453:54345,:]
```

Out[19]:

	App_Name	App_Id	Category	Rating	Rating_Count
34453	IMMIGRANTS: FRIENDS OR FOES?	com.mbspringer.imig	News & Magazines	0.0	0.0
34454	Together UCOB	br.org.ucob.together	Dating	4.3	35.0
34455	EnergyVUE	com.rainforestautomation.energyvuern	House & Home	2.3	23.0
34456	Fondos de Paisajes Bonitos	com.andromo.dev604137.app583342	Personalization	4.9	60.0
34457	Regalprüfer	com.secumundi.regalpruefer	Libraries & Demo	0.0	0.0
...
54341	Bug Climber	com.aa.bugclimber	Casual	0.0	0.0
54342	世界各国の首 都 問題集	jp.co.solcreo.quiz0012	Education	0.0	0.0
54343	Japanese Hmong Dictionary	nerdcats.japanesehmong	Books & Reference	0.0	0.0
54344	Smart Retailer	in.socialtitli.retailerapp	Business	4.0	9.0
54345	Staff Ready	net.staffconnect.mobile.staffready	Business	0.0	0.0

19893 rows × 24 columns



Finding and filling the null values in Free columns

In [20]:

```
df[df.Free.isnull()]
```

Out[20]:

App_Name	App_Id	Category	Rating	Rating_Count	Installs	Minimum_Installs	Maximum_Ins
----------	--------	----------	--------	--------------	----------	------------------	-------------



In [21]:

```
df['Free'].fillna("Free",inplace=True)
```

In [22]:

```
df.isnull().sum()
```

Out[22]:

```
App_Name          2
App_Id            0
Category          0
Rating           22883
Rating_Count      22883
Installs          107
Minimum_Installs  107
Maximum_Installs  0
Free              0
Price             0
Currency          135
Size             196
Minimum_Android   6530
Developer_Id       33
Developer_Website 760835
Developer_Email    31
Released          71053
Last_Updated       0
Content_Rating     0
Privacy_Policy     420953
Ad_Supported       0
In_App_Purchases  0
Editors_Choice     0
Scraped_Time       0
dtype: int64
```

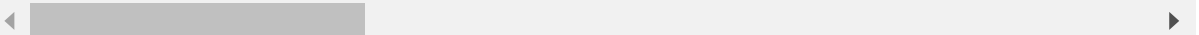
Content Rating (cleaning)

finding the null values in content Rating

In [23]:

```
df[df['Content_Rating'].isnull()]
```

Out[23]:

App_Name	App_Id	Category	Rating	Rating_Count	Installs	Minimum_Installs	Maximum_Ins
							

looking for the missing value in the content rating

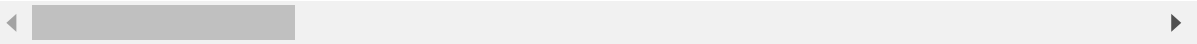
In [24]:

```
df.loc[34234:54653,:]
```

Out[24]:

	App_Name	App_Id	Category	Rating	Rating_Count
34234	Cara Menjadi Kaya -How to become rich person M...	com.howto.getrichmalay	Books & Reference	0.0	0.0
34235	HAJJA SARA AMMAL MATRIC HR SEC SCHOOL-Parent's...	com.hajjasara	Education	0.0	0.0
34236	Quizzes: Game Show Games	top.freegames.generalknowledge.quiz.trivia	Trivia	NaN	NaN
34237	Fiedora	com.heimavista.pandoraSelfie	Photography	4.1	64.0
34238	New Zealand Newspapers	com.newzealand.bydrcmob.newzealandnews	News & Magazines	0.0	0.0
...
54649	Dan Powers Advantage	com.mobileappsprn.danpowers	Business	0.0	0.0
54650	Inflight Reader	com.magmakeredition.inflight	News & Magazines	3.0	47.0
54651	ResolvedX Stage	com.sfl.is.resolvedx.stage	Business	0.0	0.0
54652	在宅ケア支援システム (bmic-ZR)	jp.bmic.zr	Medical	NaN	NaN
54653	BEAT TRAINING	com.uphydn.beat_training	Tools	0.0	0.0

20420 rows × 24 columns



Dropping the row from the column

In [25]:

```
df.dropna(subset =['Content_Rating'], inplace= True)
```

Replacing the missing values of the Rating Column

with the Mode value of that entire column

In [83]:

```
modeRating=df['Rating'].mode()
```

In [84]:

```
df['Rating'].fillna(value=modeRating[0],inplace=True)
```

In [28]:

```
printinfo()
```

Out[28]:

	dataType	nullCount	uniqueCount
App_Name	object	2	2177944
App_Id	object	0	2312944
Category	object	0	48
Rating	float64	0	42
Rating_Count	float64	22883	38482
Installs	object	107	22
Minimum_Installs	float64	107	22
Maximum_Installs	int64	0	251563
Free	bool	0	2
Price	float64	0	1063
Currency	object	135	15
Size	object	196	1657
Minimum_Android	object	6530	154
Developer_Id	object	33	758371
Developer_Website	object	760835	810440
Developer_Email	object	31	950456
Released	object	71053	4158
Last_Updated	object	0	3918
Content_Rating	object	0	6
Privacy_Policy	object	420953	977743
Ad_Supported	bool	0	2
In_App_Purchases	bool	0	2
Editors_Choice	bool	0	2
Scraped_Time	object	0	67374

Column: Size

Converting the Size column to string

In [29]:

```
df['Size'] = df['Size'].astype(str)
```

Removing the + Symbol

In [30]:

```
df['Size'] = df.Size.apply(lambda x: x.strip('+'))
```

Removing the , Symbol

In [31]:

```
df['Size'] = df.Size.apply(lambda x: x.replace(',', ''))
```

For converting the M to 1000000

In [32]:

```
df['Size'] = df.Size.apply(lambda x: x.replace('M', 'e+6'))
```

Replacing the k by multiplying the value with 1000.

In [33]:

```
df['Size'] = df.Size.apply(lambda x: x.replace('k', 'e+3'))
```

In [34]:

```
df = df[df["Size"].str.contains("G") == False]
```

Replacing the Varies with device value with Nan.

In [35]:

```
df['Size'] = df.Size.replace('Varies with device', np.NaN)
```

dropping the subset of size

In [36]:

```
df.dropna(subset = ['Size'], inplace=True)
```

Filling the null values

In [37]:

```
df['Size'] = df['Size'].fillna(0)
```

Dropping the null values

In [38]:

```
df.dropna(inplace=True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1250858 entries, 0 to 2312942
Data columns (total 24 columns):
#   Column                Non-Null Count  Dtype
---  -
0   App_Name              1250858 non-null object
1   App_Id                1250858 non-null object
2   Category              1250858 non-null object
3   Rating                1250858 non-null float64
4   Rating_Count          1250858 non-null float64
5   Installs              1250858 non-null object
6   Minimum_Installs      1250858 non-null float64
7   Maximum_Installs      1250858 non-null int64
8   Free                  1250858 non-null bool
9   Price                 1250858 non-null float64
10  Currency              1250858 non-null object
11  Size                  1250858 non-null object
12  Minimum_Android       1250858 non-null object
13  Developer_Id          1250858 non-null object
14  Developer_Website     1250858 non-null object
15  Developer_Email       1250858 non-null object
16  Released              1250858 non-null object
17  Last_Updated          1250858 non-null object
18  Content_Rating        1250858 non-null object
19  Privacy_Policy        1250858 non-null object
20  Ad_Supported          1250858 non-null bool
21  In_App_Purchases      1250858 non-null bool
22  Editors_Choice        1250858 non-null bool
23  Scraped_Time          1250858 non-null object
dtypes: bool(4), float64(4), int64(1), object(15)
memory usage: 205.2+ MB
```

Finally converting to Numeric type.

In [39]:

```
df['Size'] = pd.to_numeric(df['Size'])
```


In [40]:

```
printinfo()
```

Out[40]:

	dataType	nullCount	uniqueCount
App_Name	object	0	1210642
App_Id	object	0	1250858
Category	object	0	48
Rating	float64	0	42
Rating_Count	float64	0	31925
Installs	object	0	20
Minimum_Installs	float64	0	20
Maximum_Installs	int64	0	188439
Free	bool	0	2
Price	float64	0	607
Currency	object	0	8
Size	float64	0	1557
Minimum_Android	object	0	106
Developer_Id	object	0	441882
Developer_Website	object	0	641735
Developer_Email	object	0	546838
Released	object	0	4154
Last_Updated	object	0	3596
Content_Rating	object	0	6
Privacy_Policy	object	0	683800
Ad_Supported	bool	0	2
In_App_Purchases	bool	0	2
Editors_Choice	bool	0	2
Scraped_Time	object	0	67349

Column: Installs

Converting the Installs column from object to str

In [41]:

```
df['Installs']=df.Installs.astype(str)
```

In [42]:

```
df.Installs
```

Out[42]:

```
0          10+
1       5,000+
4        100+
5         50+
9      50,000+
...
2312933  500,000+
2312934         5+
2312938   1,000+
2312940    100+
2312942   1,000+
Name: Installs, Length: 1250858, dtype: object
```

Removing the + symbol from the values

In [43]:

```
df['Installs'] =df.Installs.apply(lambda x: x.strip('+'))
```

Removing the , from the numbers.

In [44]:

```
df['Installs'] =df.Installs.apply(lambda x: x.replace(',',''))
```

Converting it from string type to numeric type

In [45]:

```
df['Installs'] = pd.to_numeric(df['Installs'])
```

In [46]:

```
df.Installs
```

Out[46]:

```
0          10
1       5000
4        100
5         50
9      50000
...
2312933  500000
2312934         5
2312938   1000
2312940    100
2312942   1000
Name: Installs, Length: 1250858, dtype: int64
```

Converting the Installs columnn from str to float

In [47]:

```
df['Installs']=df['Installs'].astype(float)
```

In [48]:

```
printinfo()
```

Out[48]:

	dataType	nullCount	uniqueCount
App_Name	object	0	1210642
App_Id	object	0	1250858
Category	object	0	48
Rating	float64	0	42
Rating_Count	float64	0	31925
Installs	float64	0	20
Minimum_Installs	float64	0	20
Maximum_Installs	int64	0	188439
Free	bool	0	2
Price	float64	0	607
Currency	object	0	8
Size	float64	0	1557
Minimum_Android	object	0	106
Developer_Id	object	0	441882
Developer_Website	object	0	641735
Developer_Email	object	0	546838
Released	object	0	4154
Last_Updated	object	0	3596
Content_Rating	object	0	6
Privacy_Policy	object	0	683800
Ad_Supported	bool	0	2
In_App_Purchases	bool	0	2
Editors_Choice	bool	0	2
Scraped_Time	object	0	67349

Column: Price

In [49]:

```
df['Price'].value_counts()
```

Out[49]:

```
0.000000    1228757
0.990000     4586
1.990000     2908
2.990000     2261
4.990000     1650
...
2.148679         1
10.930000         1
48.800000         1
5.630000         1
3.041816         1
```

Name: Price, Length: 607, dtype: int64

In [50]:

```
df["Price"].sum()
```

Out[50]:

133811.911669

Exploratory Analysis and Visualization

In [51]:

```
sns.set_style('darkgrid')
matplotlib.rcParams['font.size']=18
matplotlib.rcParams['figure.figsize']=(15,10)
matplotlib.rcParams['figure.facecolor']='cyan'
```

exploring the column of object type.

In [52]:

```
printinfo()
```

Out[52]:

	dataType	nullCount	uniqueCount
App_Name	object	0	1210642
App_Id	object	0	1250858
Category	object	0	48
Rating	float64	0	42
Rating_Count	float64	0	31925
Installs	float64	0	20
Minimum_Installs	float64	0	20
Maximum_Installs	int64	0	188439
Free	bool	0	2
Price	float64	0	607
Currency	object	0	8
Size	float64	0	1557
Minimum_Android	object	0	106
Developer_Id	object	0	441882
Developer_Website	object	0	641735
Developer_Email	object	0	546838
Released	object	0	4154
Last_Updated	object	0	3596
Content_Rating	object	0	6
Privacy_Policy	object	0	683800
Ad_Supported	bool	0	2
In_App_Purchases	bool	0	2
Editors_Choice	bool	0	2
Scraped_Time	object	0	67349

Value count of the different section in category column

In [53]:

```
df['Category'].value_counts()
```

Out[53]:

Education	123772
Business	97744
Music & Audio	85739
Lifestyle	73591
Tools	65156
Entertainment	61992
Books & Reference	55605
Health & Fitness	50682
Shopping	48265
Productivity	45417
Travel & Local	45411
Food & Drink	45045
Finance	42811
Personalization	37212
Communication	29907
Sports	28076
News & Magazines	27961
Social	26543
Puzzle	23446
Casual	20717
Medical	19721
Arcade	19051
Photography	16493
Maps & Navigation	15262
Educational	12373
Simulation	11919
Action	11637
Auto & Vehicles	10436
Adventure	9796
House & Home	8499
Events	8309
Art & Design	7880
Video Players & Editors	6558
Beauty	6116
Trivia	5715
Role Playing	5371
Racing	5056
Board	4888
Word	4757
Card	4355
Strategy	3890
Weather	3785
Dating	3273
Casino	2609
Parenting	2322
Libraries & Demo	2297
Music	2127
Comics	1271

Name: Category, dtype: int64

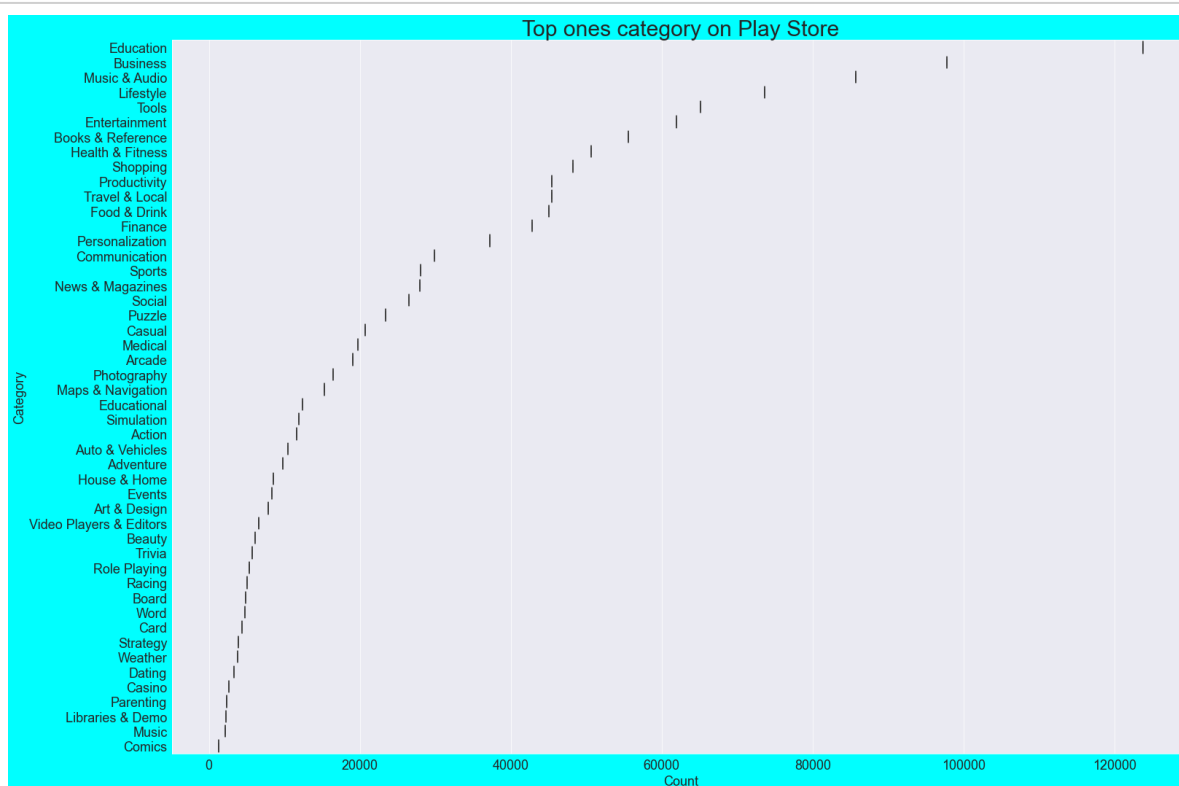
Top category on playstore

In [54]:

```
y=df['Category'].value_counts().index
x=df['Category'].value_counts()
xaxis=[]
yaxis=[]
for i in range(len(x)) :
    xaxis.append(x[i])
    yaxis.append(y[i])
```

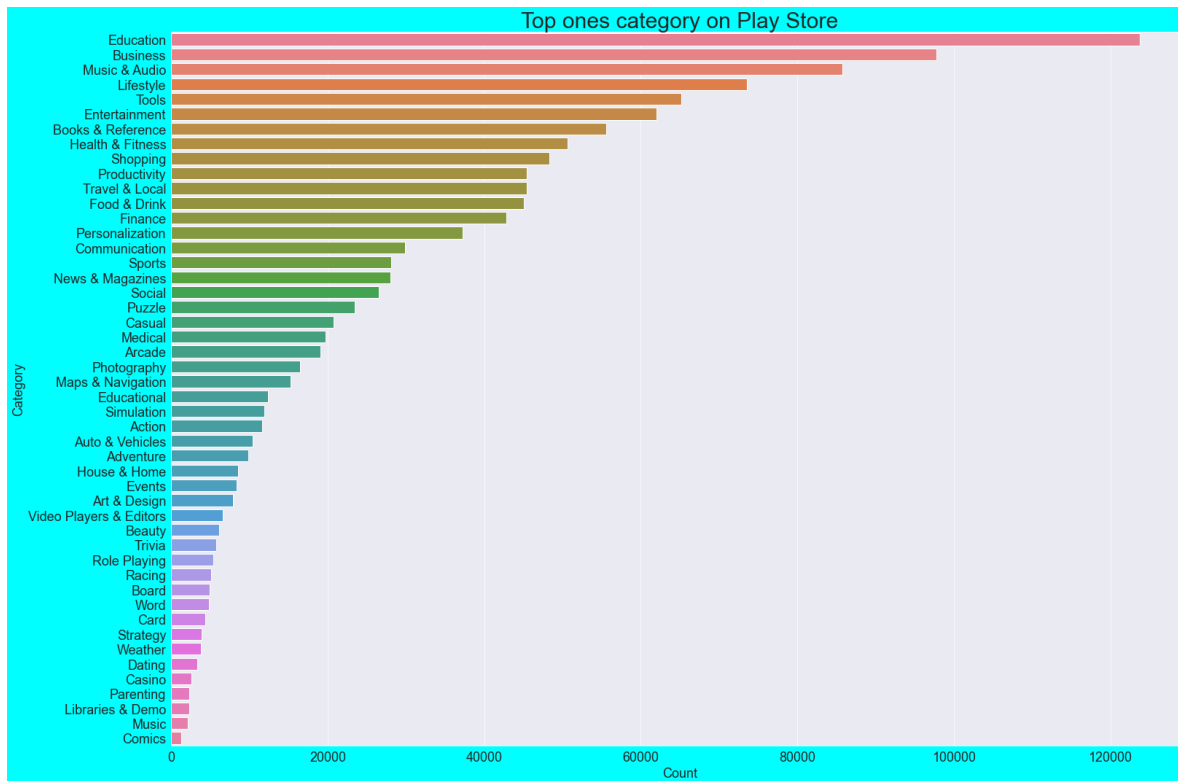
In [55]:

```
plt.figure(figsize=(25,18))
plt.xlabel('Count')
plt.ylabel('Category')
graph = sns.boxplot(x= xaxis, y = yaxis, palette = "husl")
graph.set_title("Top ones category on Play Store", fontsize = 30);
```



In [56]:

```
plt.figure(figsize=(25,18))
plt.xlabel('Count')
plt.ylabel('Category')
graph = sns.barplot(x= xaxis, y = yaxis, palette = "husl")
graph.set_title("Top ones category on Play Store", fontsize = 30);
```



Note:- From the above graphs we come to know that there are total 50 categories in this dataset and the most apps on the playstore falls under the Education and Business and the least falls under the comics

Which category of Apps from the Content Rating column are found more on playstore

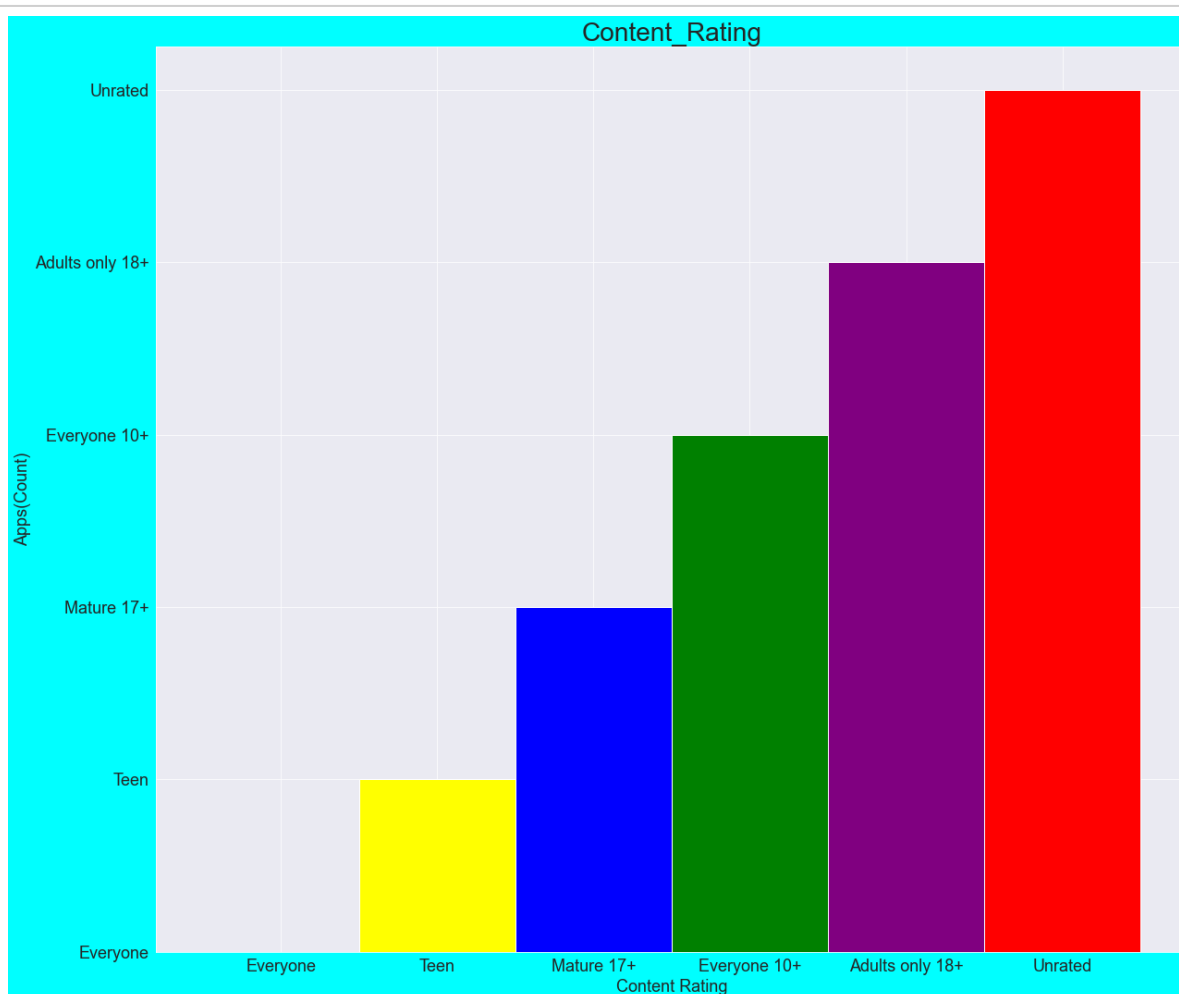
In [57]:

```
x2= df['Content_Rating'].value_counts().index
y2 =df['Content_Rating'].value_counts()

x2axis=[]
y2axis=[]
for i in range(len(x2)):
    x2axis.append(x2[i])
    y2axis.append(x2[i])
```

In [58]:

```
plt.figure(figsize=(20,18))
plt.bar(x2axis,y2axis,width=1,color=['grey','yellow','blue','green','purple','red'],alpha=1)
plt.title('Content_Rating',size = 28);
plt.ylabel('Apps(Count)');
plt.xlabel('Content Rating');
```



Note: From the above plot we can see that Unrated category has the highest number of apps.

Distribution of the ratings the dataframe.

In [59]:

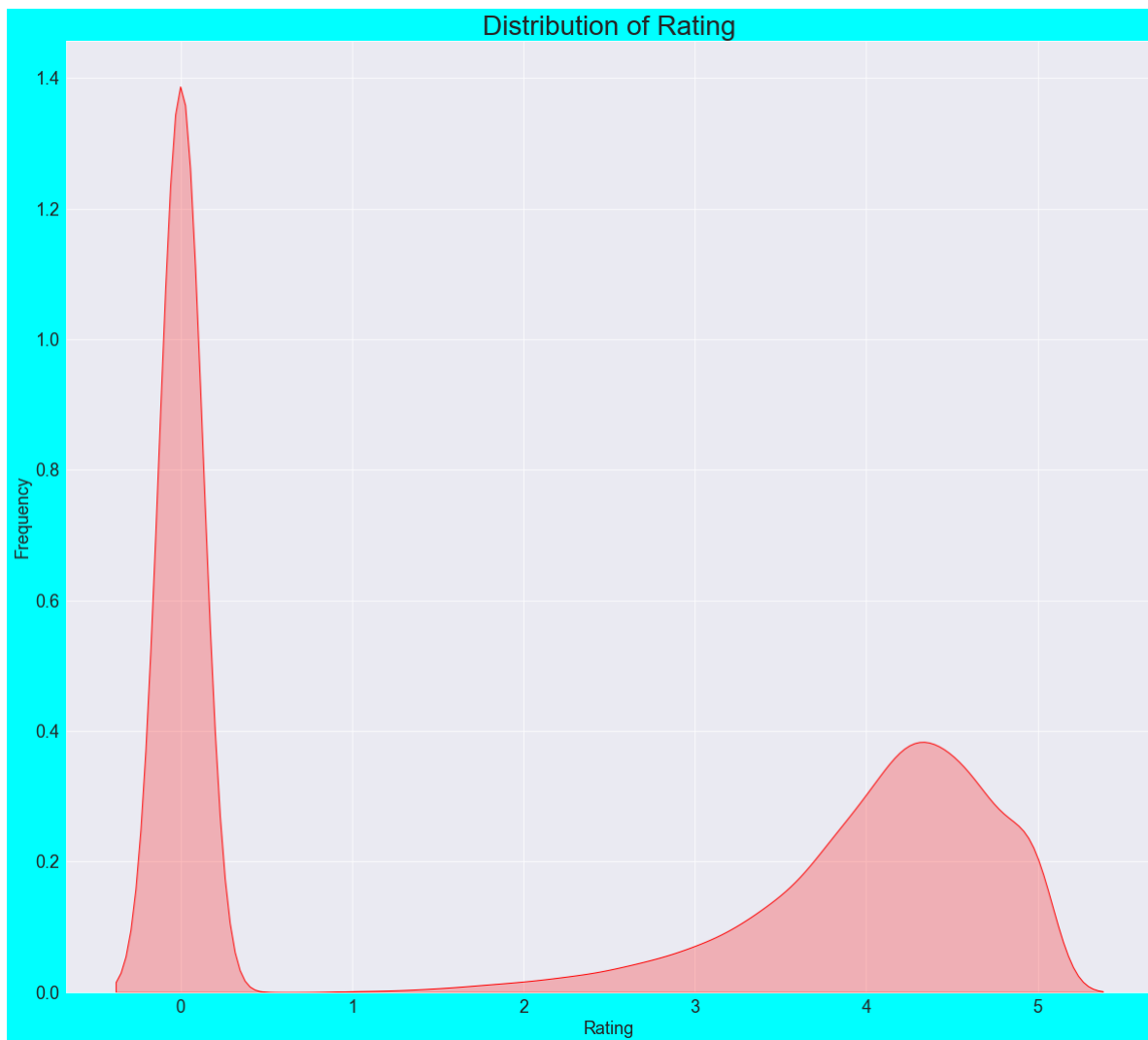
```
df['Rating'].describe()
```

Out[59]:

```
count    1.250858e+06
mean     2.292466e+00
std      2.094202e+00
min      0.000000e+00
25%      0.000000e+00
50%      3.200000e+00
75%      4.300000e+00
max      5.000000e+00
Name: Rating, dtype: float64
```

In [60]:

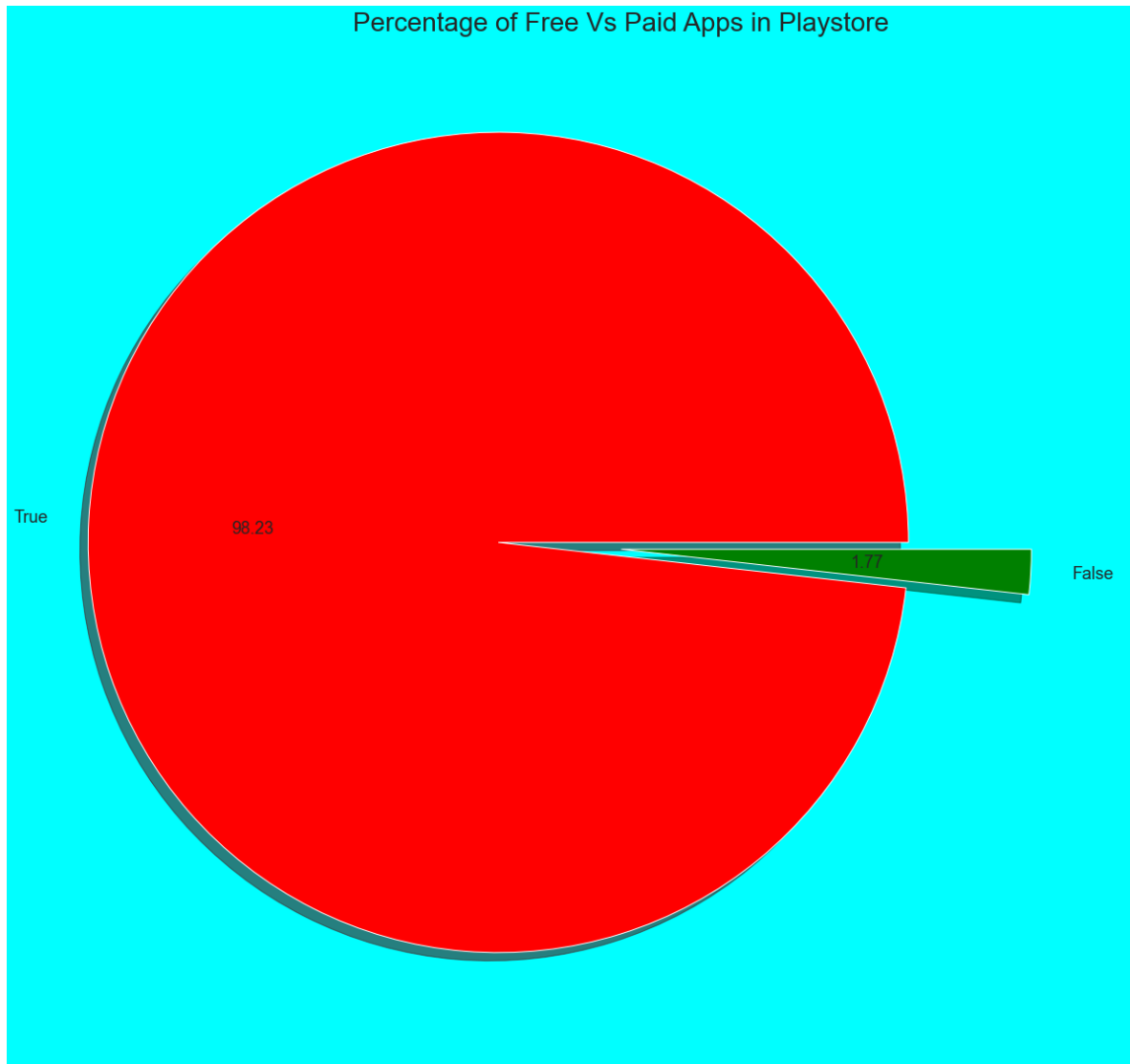
```
plt.figure(figsize=(20,18))
graph = sns.kdeplot(df.Rating, color = 'red',shade = True)
plt.title('Distribution of Rating',size = 28);
plt.ylabel('Frequency');
plt.xlabel('Rating');
```



The apps in playstore are paid or free

In [61]:

```
plt.figure(figsize=(20,20))
labels=df['Free'].value_counts(sort =True).index
sizes =df['Free'].value_counts(sort=(True))
colors = ["red","green"]
explode =(0.3,0)
plt.pie(sizes,explode=explode,labels=labels,colors=colors ,autopct='%2.2f',shadow = True, s
plt.title('Percentage of Free Vs Paid Apps in Playstore',size=28)
plt.show()
```



Note: From the above graph we can see that 98.23%(Approx.) of apps in google play store are free and 1.77% (Approx.) are paid.

Which category App's have most number of installs?

In [62]:

```
highest__Installs_df=df.groupby('Category')[['Installs']].sum().sort_values(by='Installs',a
```

In [63]:

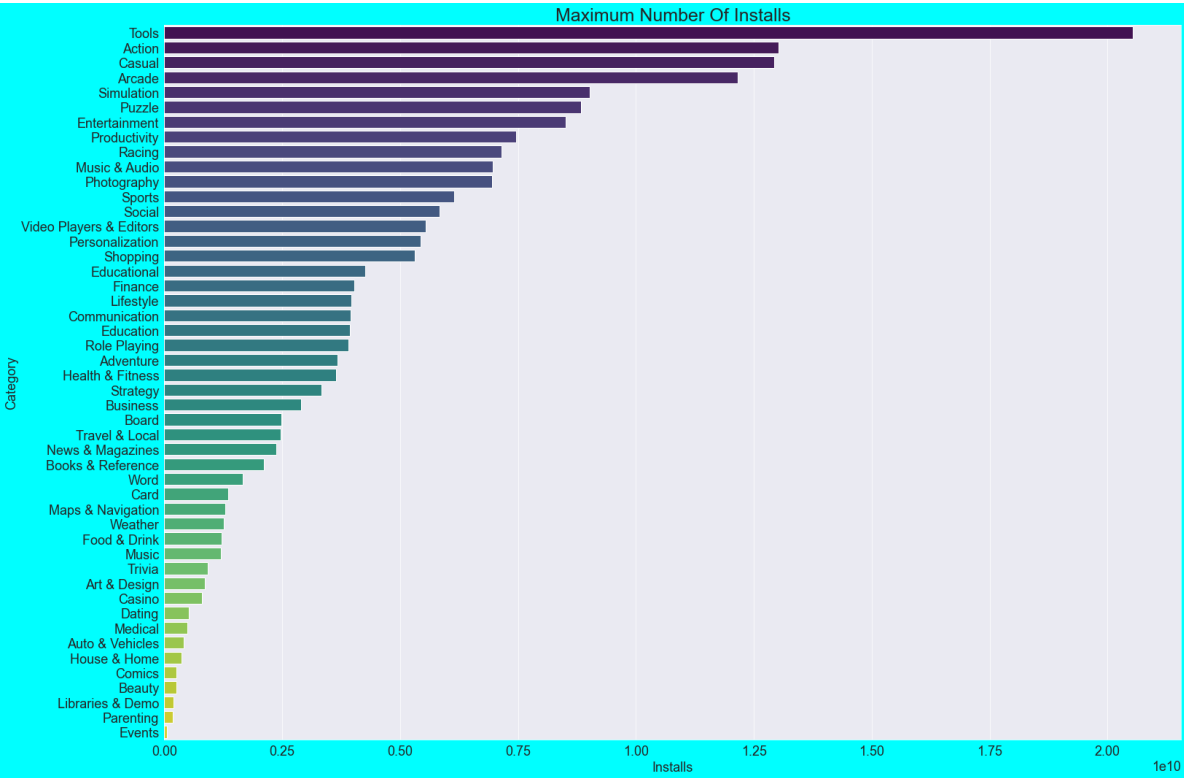
```
highest__Installs_df.head()
```

Out[63]:

Installs	
Category	
Tools	2.053725e+10
Action	1.302693e+10
Casual	1.293569e+10
Arcade	1.216118e+10
Simulation	9.022042e+09

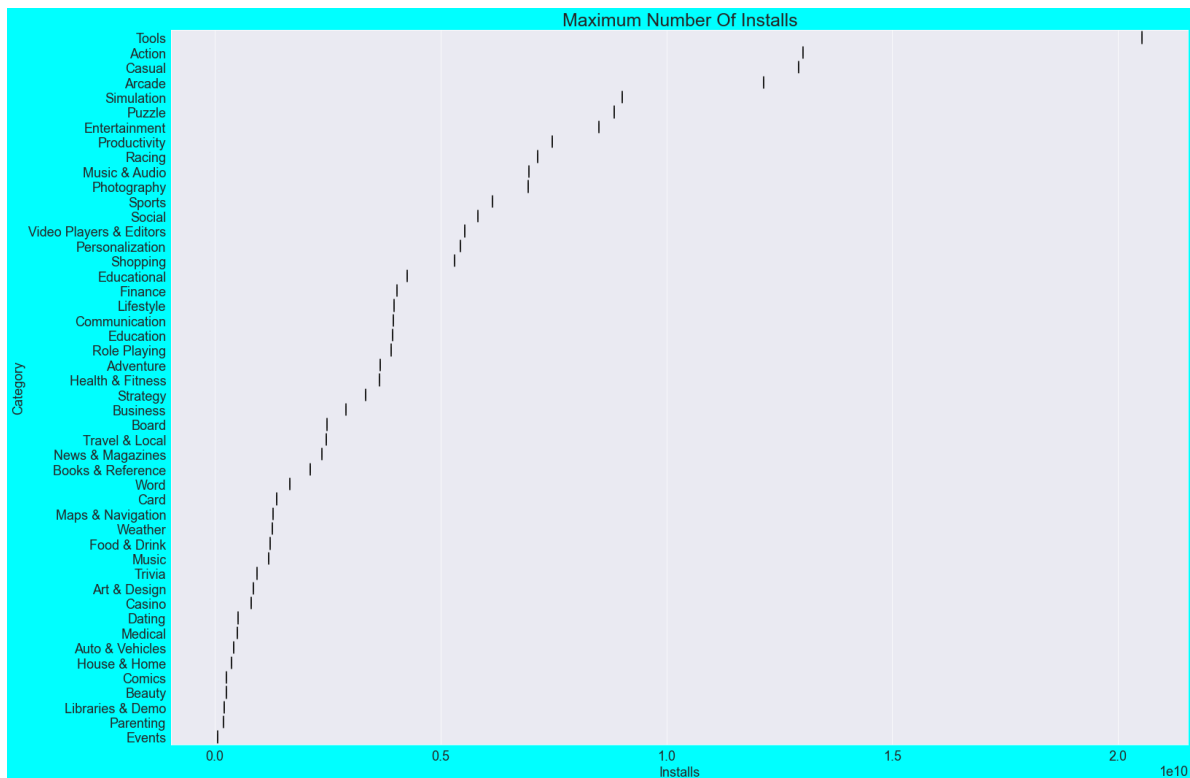
In [64]:

```
x2axis=[]
y2axis=[]
for i in range (len(highest__Installs_df)):
    x2axis.append(highest__Installs_df.Installs[i])
    y2axis.append(highest__Installs_df.index[i])
plt.figure(figsize=(25,18))
plt.xlabel("Installs")
plt.ylabel("Category")
graph=sns.barplot(x=x2axis,y=y2axis, alpha =1,palette = "viridis")
graph.set_title('Maximum Number Of Installs', fontsize = 25);
```



In [65]:

```
x2axis=[]
y2axis=[]
for i in range (len(highest__Installs_df)):
    x2axis.append(highest__Installs_df.Installs[i])
    y2axis.append(highest__Installs_df.index[i])
plt.figure(figsize=(25,18))
plt.xlabel("Installs")
plt.ylabel("Category")
graph=sns.boxplot(x=x2axis,y=y2axis,palette = "viridis")
graph.set_title('Maximum Number Of Installs', fontsize = 25);
```



Note: From the above visualization, it can be interpreted that the top categories with highest installs are Tools, Action, Arcade, Casual.

What are the Top 10 installed apps in any category?

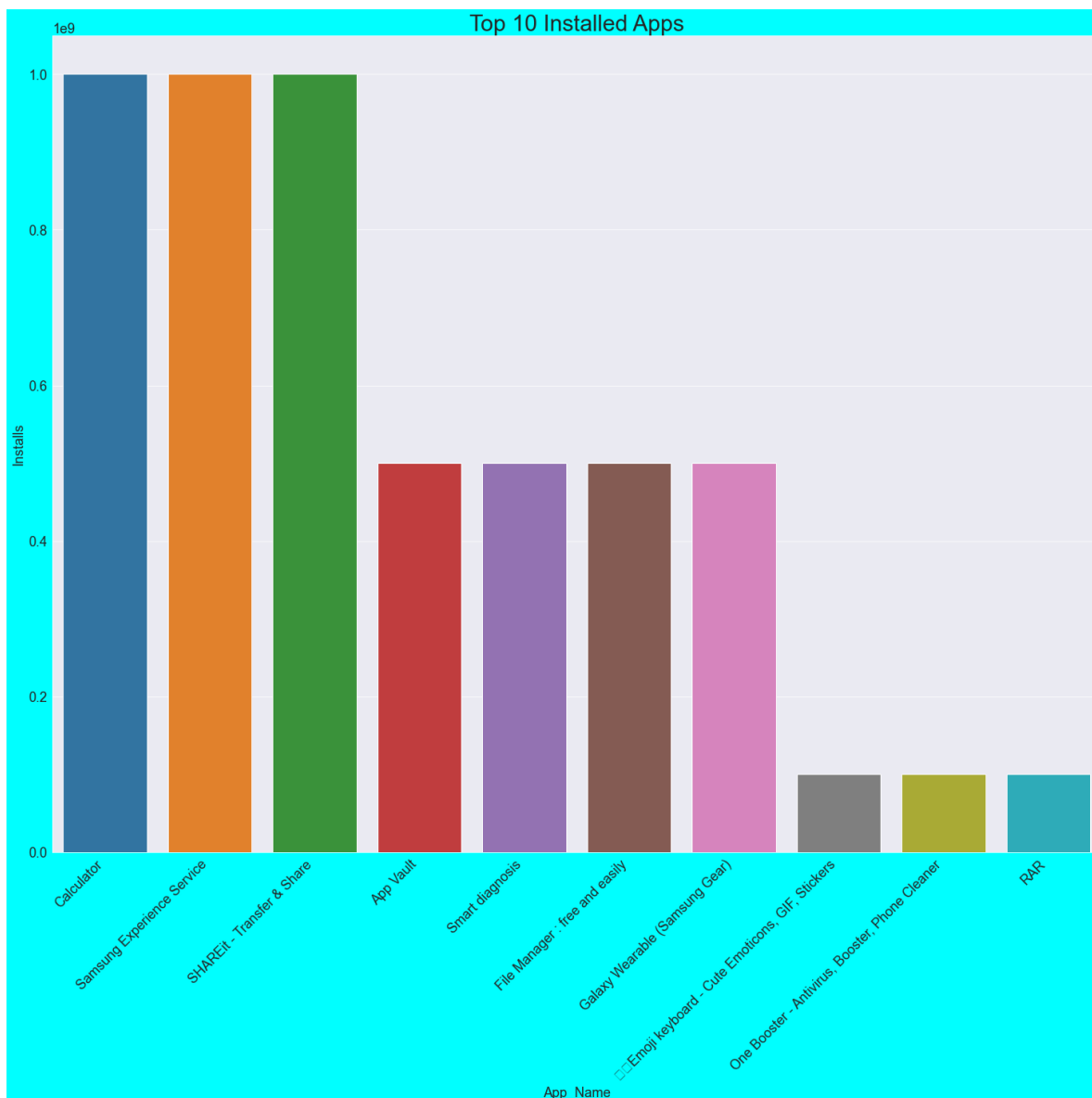
In [66]:

```
def find_Top_10_In_Category(str):
    top10=df[df['Category']==str]
    top10apps=top10.sort_values(by='Installs',ascending=False).head(10)
    plt.figure(figsize=(25,20))
    plt.title('Top 10 Installed Apps',size=30);
    graph=sns.barplot(x=top10apps.App_Name, y =top10apps.Installs)
    graph.set_xticklabels(graph.get_xticklabels(), rotation= 45, horizontalalignment='right')
```

In [67]:

```
find_Top_10_In_Category('Tools')
```

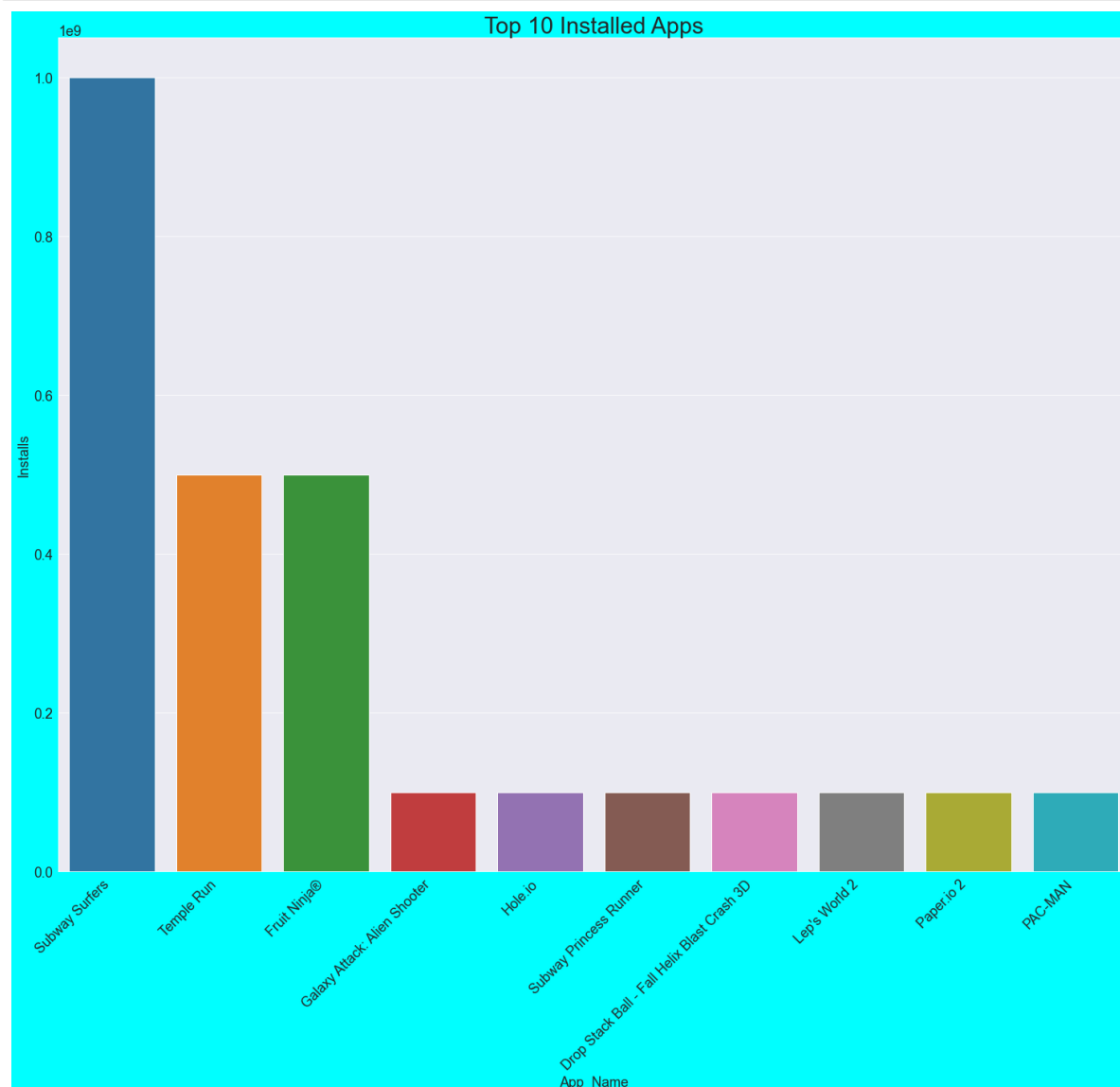
```
C:\Users\nager\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.p
y:240: RuntimeWarning: Glyph 10084 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\nager\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.p
y:240: RuntimeWarning: Glyph 65039 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\nager\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.p
y:203: RuntimeWarning: Glyph 10084 missing from current font.
  font.set_text(s, 0, flags=flags)
C:\Users\nager\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.p
y:203: RuntimeWarning: Glyph 65039 missing from current font.
  font.set_text(s, 0, flags=flags)
```



C In the same way we by passing different category names to the function, we can get the top 10 installed apps.

In [68]:

```
find_Top_10_In_Category('Arcade')
```



Note: From the above graph we can see that in the Sports category Subway Surfers has the highest installs.

Which are the top 10 expensive Apps in playstore?

In [69]:

```
top10 = df[df['Free']==False]
print(top10)
top10_Paid_Apps = top10.sort_values(by=['Price'],ascending = False).head(5)
print(top10_Paid_Apps.head(5))
(https://sites.google.com/view/wastickerapps-wa...)
```

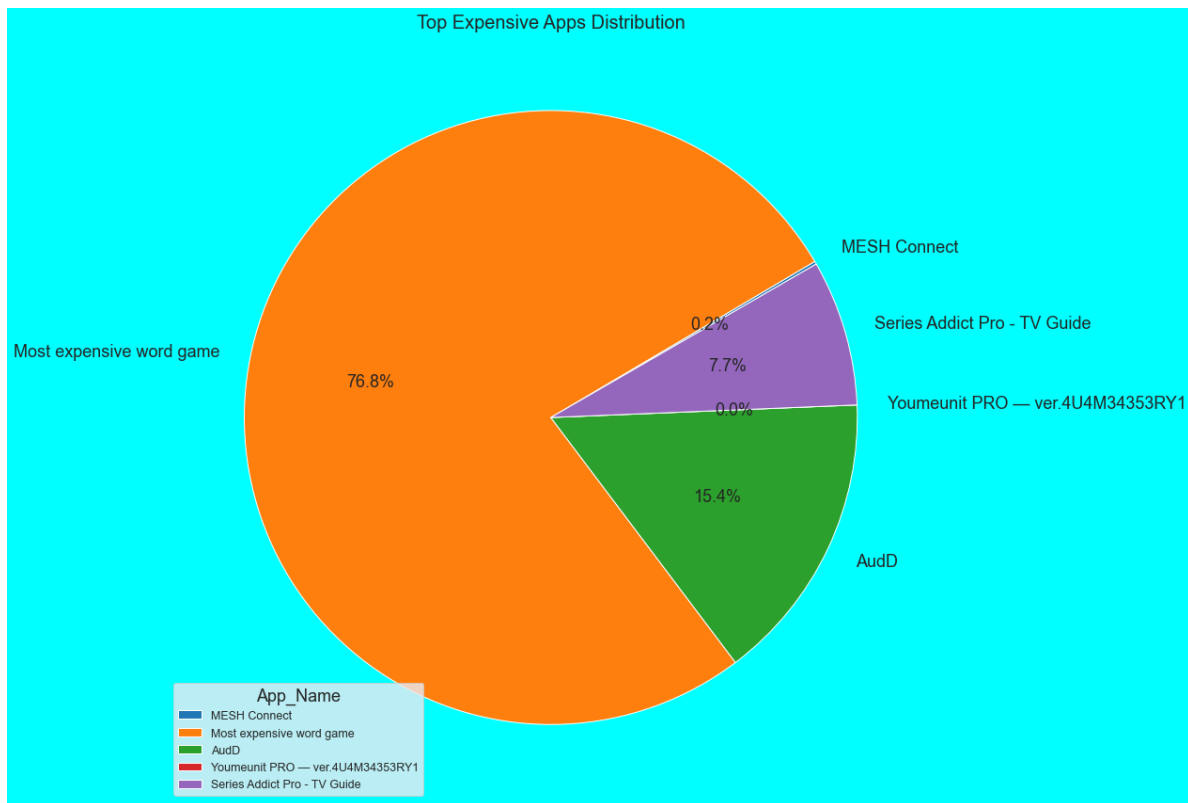
	Ad_Supported	In_App_Purchases	Editors_Choice	Scraped_Ti
me				
296	False	False	False	2021-06-15 20:19:
53				
554	False	False	False	2021-06-15 20:20:
07				
618	True	True	False	2021-06-15 20:20:
11				
637	False	False	False	2021-06-15 20:20:
13				
821	False	False	False	2021-06-15 20:20:
23				
...	
...				
2312483	False	False	False	2021-06-16 12:58:
54				
2312628	False	False	False	2021-06-16 12:59:
02				

In [70]:

```
top10_Paid_Apps_df = top10_Paid_Apps[['App_Name', 'Installs']]
```


In [71]:

```
plt.figure(figsize=(20,15));
plt.pie(top10_Paid_Apps_df.Installs, explode=None, labels=top10_Paid_Apps_df.App_Name, auto
plt.title('Top Expensive Apps Distribution',size = 20);
plt.legend(top10_Paid_Apps_df.App_Name,
          loc="lower left",
          title="App_Name",
          fontsize = "x-small"
);
```



Note: From the above graph we can interpret that the App Most Expensive word game is the most expensive app in the google playstore followed by AudD.

Which are the Apps with highest number of rating?

In [72]:

```
AppwithHighestrev=df.sort_values(by='Rating',ascending=False).head(10)
AppwithHighestrev
```

Out[72]:

	App_Name	App_Id	Category	Rating	Rating_Count	Install
534320	Delico Store	com.qiotic.delicostore	Shopping	5.0	7.0	5000.
1711859	Electronics Reuse Conference	events.socio.app307	Business	5.0	6.0	100.
302552	Mudakhir	com.mudakhir	Finance	5.0	5.0	10.
2105991	GlassGet	com.bsg.glassget	House & Home	5.0	6.0	500.
981367	عالم رقمي	com.sis.digitalworld	News & Magazines	5.0	9.0	100.
1571166	SPERA HOT YOGA	com.fitnessmobileapps.sperahotyoga	Health & Fitness	5.0	5.0	1000.
302583	Dümen	com.dumen.app	Events	5.0	14.0	10.
1975054	Wiener tarife 2018 Crna Gora	me.wiener.android.tarife.x2018	Finance	5.0	9.0	100.
1092875	Digimealz	store.digimealz.menu	Food & Drink	5.0	10.0	10.
470026	Dicas para o Carnaval	com.wgpapps.dicascarnaval	Events	5.0	22.0	50.

What are the count of Apps in different genres?

In [73]:

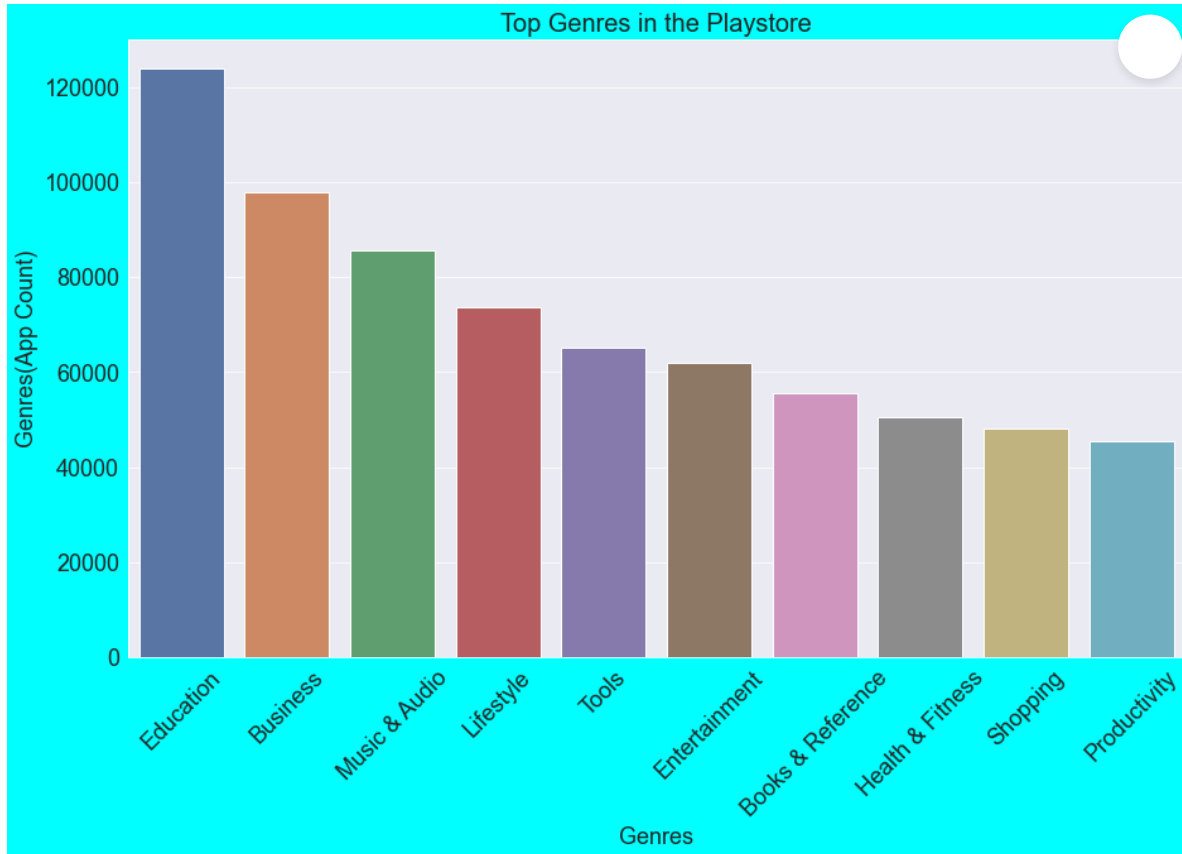
```
top_apps_in_genres=df['Category'].value_counts().head(10)
```

In [74]:

```
x3axis = []
y3axis = []
for i in range(len(top_apps_in_genres)):
    x3axis.append(top_apps_in_genres.index[i])
    y3axis.append(top_apps_in_genres[i])
```

In [75]:

```
plt.figure(figsize=(15,9))
plt.ylabel('Genres(App Count)')
plt.xlabel('Genres')
graph = sns.barplot(x=x3axis,y=y3axis,palette="deep")
graph.set_xticklabels(graph.get_xticklabels(), rotation=45, fontsize=18)
graph.set_title("Top Genres in the Playstore", fontsize = 20);
```



Note: From the above visualization we can see that the Highest Number of Apps found in the Tools and Business genres followed by Education, Medical and many more.

Which are the apps that have made the highest earning?

In [86]:

```
paid_apps_df=df[df['Free']==False]
```

In [87]:

```
earnings_df=paid_apps_df[['App_Name','Installs','Price']]
```

In [88]:

```
earnings_df['Earnings']=paid_apps_df['Installs']*paid_apps_df['Price']
```

C:\Users\nager\AppData\Local\Temp\ipykernel_9968\4203358835.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
earnings_df['Earnings']=paid_apps_df['Installs']*paid_apps_df['Price']
```

In [79]:

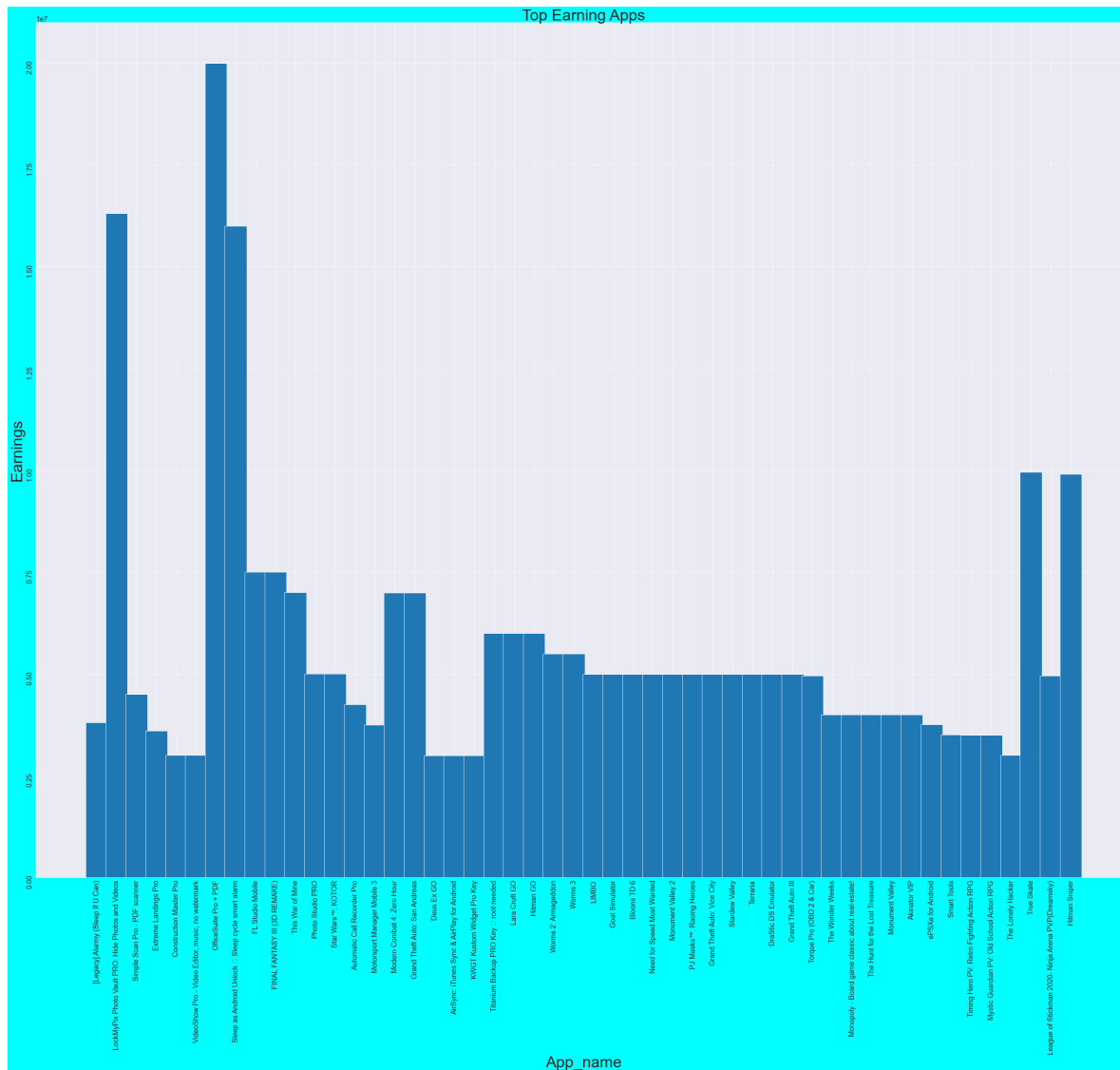
```
earnings_df_sorted_by_Earnings = earnings_df.sort_values(by='Earnings',ascending=False).head()
```

In [80]:

```
earnings_df_sorted_by_Price = earnings_df_sorted_by_Earnings.sort_values(by='Price', ascending=False)
```

In [91]:

```
plt.figure(figsize=(50,40))
plt.xlabel("App_name",size=40)
plt.ylabel("Earnings",size=40)
plt.bar(earning_df_sorted_by_Price.App_Name, earning_df_sorted_by_Price.Earnings, width=1.1)
plt.tick_params(rotation=90)
plt.title("Top Earning Apps",size=40);
```



Inferences and Conclusion

After Analyzing the dataset I have got answers to some of the serious & interesting question which any of the android users would love to know.

Top categories on Google Playstore? Which category of Content are found more? Distribution of the ratings of the apps? What percentage of apps are Free and Paid? Which category of App's have most number of installs? What are the Top 10 installed apps in different category? Which are the top expensive Apps? Which are the Apps with highest number of reviews? Count of Apps found in different genres? Which are the apps that have made the highest earning? After the completion of my project and the course I have learned and got exposure to different tools and techniques in data analysis. I was able to complete the project successfully with the help of tools like Python, Pandas, Matplotlib, Numpy, Seaborn. Also I learned different techniques like Data Cleaning, Data Preparation, Data Exploration and visualization, and Data Interpretation.

In []: