



UNIVERSITY OF
KARACHI

DATA ANALYSIS

ONE - PIECE

PREPARED BY

M USAMA SADIQ
NABEEL IQBAL
SAMIULLAH

TABLE OF CONTENT

INTRODUCTION -----	Pg3
DATA SCIENCE -----	Pg3
DATA ANALYSIS -----	Pg3
PANDAS -----	Pg3
MATPLOTLIB -----	Pg4
SEABORN -----	Pg4
STEP NO: 1 -----	Pg 4
STEP NO: 2 -----	Pg5
STEP NO: 3 -----	Pg5
STEP NO: 4 -----	Pg5
STEP NO: 5 -----	Pg6
STEP NO: 6 -----	Pg7
STEP NO: 7 -----	Pg8
STEP NO: 8 -----	Pg8
STEP NO: 9 -----	Pg9
STEP NO: 10-----	Pg10
STEP NO: 11-----	Pg11
STEP NO : 12-----	Pg11
STEP NO: 13-----	Pg12
STEP NO: 14-----	Pg13
STEP NO: 15-----	Pg14
STEP NO : 16-----	Pg18
STEP NO: 17-----	Pg19
STEP NO: 18-----Pg20	STEP NO: 19-----Pg21

INTRODUCTION

In this project we use Python (Jupyter Note Book) And Import Pandas , Matplotlib , and seaborn libraries to analyze the data of the show “ONE PIECE”. The data contains thousands of episodes.

In this presentation, we will be learning all the basics and in-depth of;

- Data Science
- Data Analysis

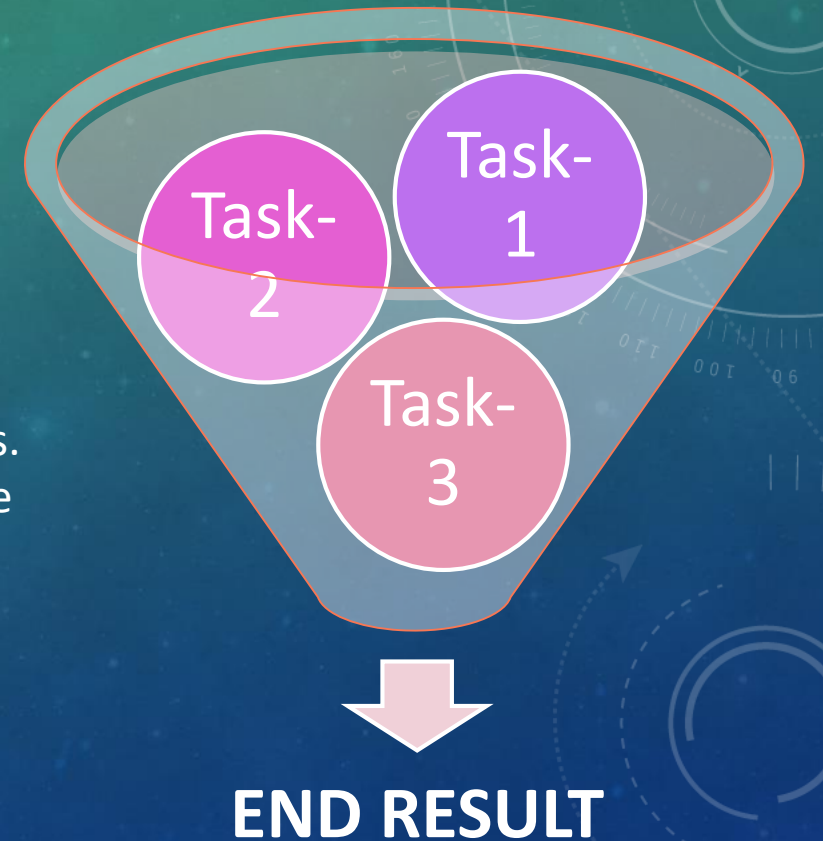
DATA SCIENCE

Data science is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data. Data science practitioners apply machine learning algorithms to numbers, text, images, video, audio, and more to produce artificial intelligence (AI) systems to perform tasks that ordinarily require human intelligence. In turn, these systems generate insights which analysts and business users can translate into tangible business value.



DATA ANALYSIS

Data analysis is a process of inspecting, cleansing, transforming, and modelling data with the goal of discovering useful information, informing conclusions, and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and is used in different business, science, and social science domains. In today's business world, data analysis plays a role in making decisions more scientific and helping businesses operate more effectively



PANDAS

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

MATPLOTLIB:

Matplotlib is a plotting library available for the Python programming language as a component of NumPy, a big data numerical handling resource. Matplotlib uses an object oriented API to embed plots in Python applications

SEABORN:

Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions

...Now moving towards the coding section...

STEP:01

DATA ANALYSIS OF ONE PIECE

Import the Required Modules.

```
In [1]: import pandas as pd  
import numpy as np  
import seaborn as sns  
import matplotlib.pyplot as plt  
%matplotlib inline
```


STEP:02

Read Data

Please ensure that your folder name doesnot have a space or afterward slash..

```
In [2]: anime = pd.read_csv(r"C:\Users\asd\Downloads\ONE PIECE.csv")
```

STEP:03

Inspecting the Rows

check 10 Top Rows of the Dataset.

In [27]: `anime.head(10)`

Out[27]:

	Unnamed: 0	rank	trend	season	episode	name	start	total_votes	average_rating
0	0	24,129	18	1	1	I'm Luffy! The Man Who Will Become the Pirate ...	1999	647	7.6
1	1	29,290	11	1	2	The Great Swordsman Appears! Pirate Hunter, Ro...	1999	473	7.8
2	2	32,043	7	1	3	Morgan vs. Luffy! Who's This Beautiful Young G...	1999	428	7.7
3	3	28,818	8	1	4	Luffy's Past! The Red-haired Shanks Appears!	1999	449	8.1
4	4	37,113	4	1	5	Fear, Mysterious Power! Pirate Clown Captain B...	1999	370	7.5
5	5	36,209	4	1	6	Desperate Situation! Beast Tamer Mohji vs. Luffy!	1999	364	7.7
6	6	37,648	4	1	7	Sozetsu Ketto! Kengo Zoro VS Kyokugei no Kabaji!	1999	344	7.7
7	7	38,371	6	1	8	Shousha wa docchi? Akuma no mi no nouryoku tai...	1999	335	7.7
8	8	42,249	5	1	9	Seigi no usotsuki? Kyaputen Usoppu	2000	327	7.3
9	9	41,829	4	1	10	Chijou saikyou no hen na yatsu! Saiminjutsushi...	2000	314	7.5

STEP:04

check Last 10 Rows of the Dataset.

In [6]: `anime.tail(10)`

Out[6]:

	Unnamed: 0	rank	trend	season	episode	name	start	total_votes	average_rating
948	948	50,714	16	1	949	We're Here to Win! Luffy's Desperate Scream!	2020	189	8.5
949	949	49,737	15	1	950	Warriors' Dream! Luffy's Conquer of Udon!	2020	199	8.4
950	950	59,574	15	1	951	Orochi's Pursuers! Ninja Army Corps vs. Zoro	2020	177	7.9
951	951	54,156	16	1	952	Tension Rises in Onigashima! Two Emperors of t...	2020	191	8.1
952	952	45,193	28	1	953	Hiyori's Confession! Reunion on Oihagi Bridge	2020	249	8.0
953	953	41,448	26	1	954	Its Name is Enma! Oden's Meito!	2020	302	7.7
954	954	35,342	44	1	955	"A New Alliance?! Kaido's Army Gathers"	2020	407	7.4
955	955	33,715	75	1	956	Ticking Down to the Great Battle! The Straw Ha...	2020	353	8.2
956	956	2,940	964	1	957	Big News! The Warlords Attack Incident	2021	2,862	9.1
957	957	14,751	-	1	958	"The Legendary Battle! Garp and Roger"	2021	746	9.4

STEP:05

Find shape of Our Dataset

(Number of Rows and Number of Columns).

Getting Information about our Dataset Like Total Number of Rows and Columns, Datatypes and Memory Required.

```
In [7]: anime.shape
```

```
Out[7]: (958, 9)
```

```
In [8]: print("Number of Rows",anime.shape[0])  
        print("Number of Columns",anime.shape[1])
```

```
Number of Rows 958
```

```
Number of Columns 9
```


STEP:06

Display Title of the Episodes

```
In [14]: anime.columns
```

```
Out[14]: Index(['Unnamed: 0', 'rank', 'trend', 'season', 'episode', 'name', 'start',  
              'total_votes', 'average_rating'],  
              dtype='object')
```

```
In [15]: anime['episode']
```

```
Out[15]: 0      1  
         1      2  
         2      3  
         3      4  
         4      5  
         ...  
        953    954  
        954    955  
        955    956  
        956    957  
        957    958  
        Name: episode, Length: 958, dtype: int64
```

STEP:07

In [9]: anime.info()

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

RangeIndex: 958 entries, 0 to 957

Data columns (total 9 columns):

#	Column	Non-Null Count	Dtype
0	Unnamed: 0	958 non-null	int64
1	rank	958 non-null	object
2	trend	958 non-null	object
3	season	958 non-null	int64
4	episode	958 non-null	int64
5	name	958 non-null	object
6	start	958 non-null	int64
7	total_votes	958 non-null	object
8	average_rating	958 non-null	float64

dtypes: float64(1), int64(4), object(4)

memory usage: 67.5+ KB

STEP:08

Get Overall Statistics About The DataFrame

To Find Mean, Mode and Median.

In [10]: `anime.describe()`

Out[10]:

	Unnamed: 0	season	episode	start	average_rating
count	958.000000	958.0	958.000000	958.000000	958.000000
mean	478.500000	1.0	479.500000	2010.231733	7.796555
std	276.695079	0.0	276.695079	6.049997	0.589670
min	0.000000	1.0	1.000000	1999.000000	5.600000
25%	239.250000	1.0	240.250000	2005.000000	7.500000
50%	478.500000	1.0	479.500000	2010.000000	7.800000
75%	717.750000	1.0	718.750000	2015.000000	8.200000
max	957.000000	1.0	958.000000	2021.000000	9.600000

STEP:09

Statistics for catogrical and numerical.

```
In [11]: anime.describe(include='all')
```

Out[11]:

	Unnamed: 0	rank	trend	season	episode	name	start	total_votes	average_rating
count	958.000000	958	958	958.0	958.000000	958	958.000000	958	958.000000
unique	NaN	958	34	NaN	NaN	957	NaN	210	NaN
top	NaN	24,129	-	NaN	NaN	The Cake Sank?! Sanji and Bege's Getaway Battle!	NaN	118	NaN
freq	NaN	1	374	NaN	NaN	2	NaN	24	NaN
mean	478.500000	NaN	NaN	1.0	479.500000	NaN	2010.231733	NaN	7.796555
std	276.695079	NaN	NaN	0.0	276.695079	NaN	6.049997	NaN	0.589670
min	0.000000	NaN	NaN	1.0	1.000000	NaN	1999.000000	NaN	5.600000
25%	239.250000	NaN	NaN	1.0	240.250000	NaN	2005.000000	NaN	7.500000
50%	478.500000	NaN	NaN	1.0	479.500000	NaN	2010.000000	NaN	7.800000
75%	717.750000	NaN	NaN	1.0	718.750000	NaN	2015.000000	NaN	8.200000
max	957.000000	NaN	NaN	1.0	958.000000	NaN	2021.000000	NaN	9.600000

STEP:10

Check For Duplicate Data

```
In [13]: ▶ dup_data=anime.duplicated().any()  
print("Are there any duplicate Values?",dup_data)
```

Are there any duplicate Values? False

STEP:11

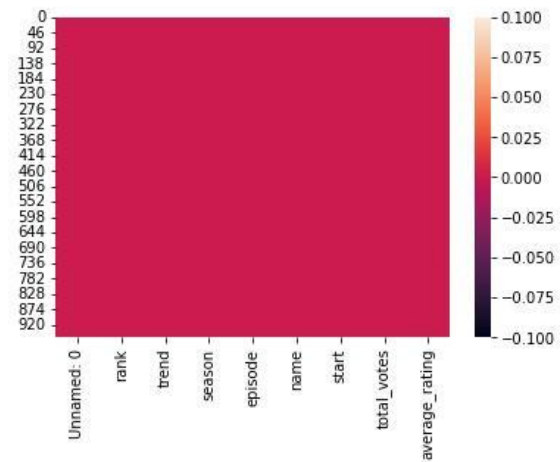
Check Missing Values In the Dataset

we can use the Dataset to create a simple **heatmap**. to see where we are missing data!

```
In [12]: print("Any missing Value?",anime.isnull().values.any())
sns.heatmap(anime.isnull())
```

Any missing Value? False

Out[12]: <AxesSubplot:>

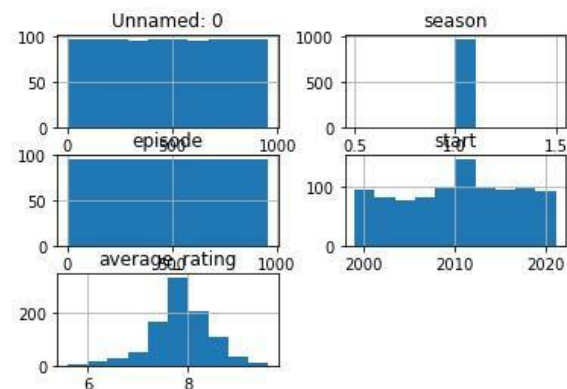


STEP:12

Showing histogrammically

In [16]: `anime.hist()`

Out[16]: array([[<AxesSubplot:title={'center':'Unnamed: 0'}>],
[<AxesSubplot:title={'center':'season'}>],
[<AxesSubplot:title={'center':'episode'}>],
[<AxesSubplot:title={'center':'start'}>],
[<AxesSubplot:title={'center':'average_rating'}>],
dtype=object)

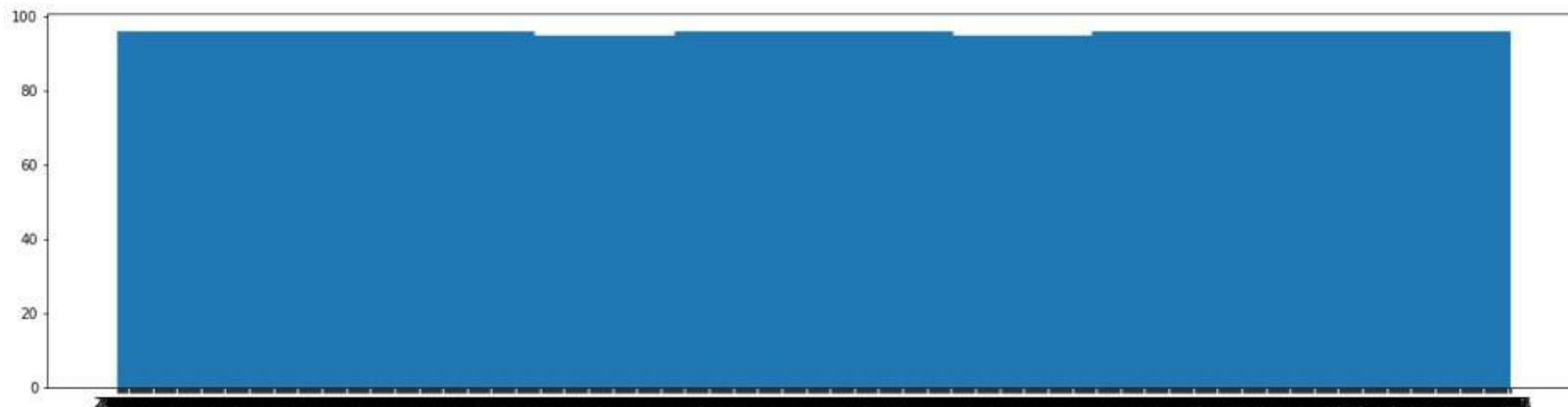


STEP:13

Find the Rank Overall

```
In [17]: plt.figure(figsize = (20,5))  
plt.hist(anime['rank'])
```

```
Out[17]: (array([96., 96., 96., 95., 96., 96., 95., 96., 96., 96.]),  
array([ 0. , 95.7, 191.4, 287.1, 382.8, 478.5, 574.2, 669.9, 765.6,  
      861.3, 957. ]),  
<BarContainer object of 10 artists>)
```

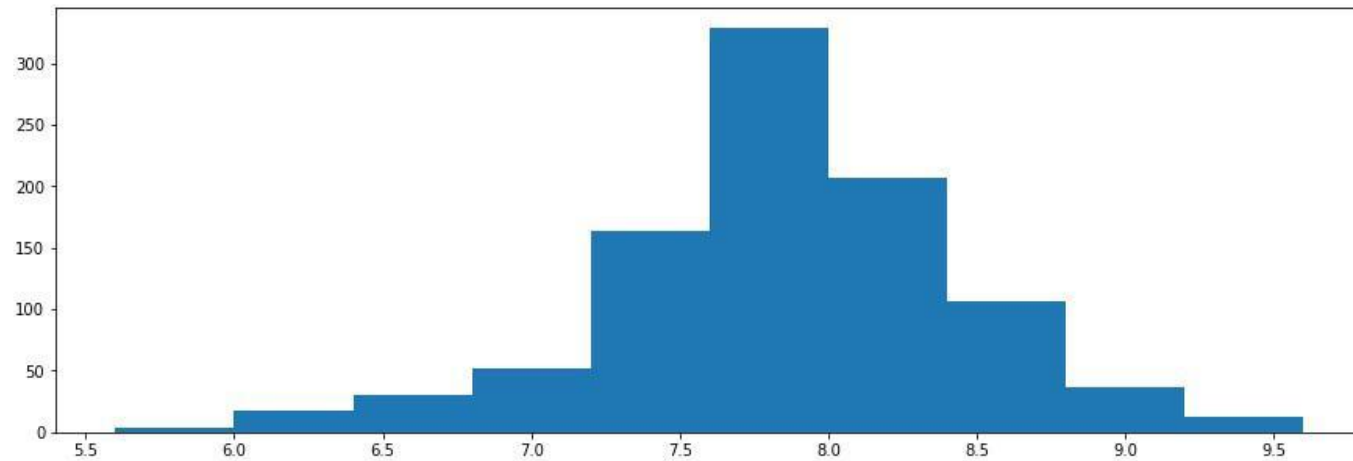


STEP:14

Find The Average Rating Overall

```
In [18]: plt.figure(figsize = (15,5))  
plt.hist(anime['average_rating'])
```

```
Out[18]: (array([ 3., 18., 30., 52., 164., 329., 207., 107., 36., 12.]),  
array([5.6, 6. , 6.4, 6.8, 7.2, 7.6, 8. , 8.4, 8.8, 9.2, 9.6]),  
<BarContainer object of 10 artists>)
```

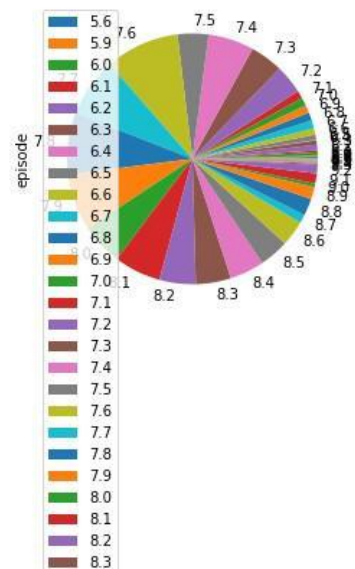


STEP:15

Finding The Average Rating per Episode

```
In [19]: anime.groupby(['average_rating']).sum().plot(kind='pie', y='episode')
```

```
Out[19]: <AxesSubplot:ylabel='episode'>
```

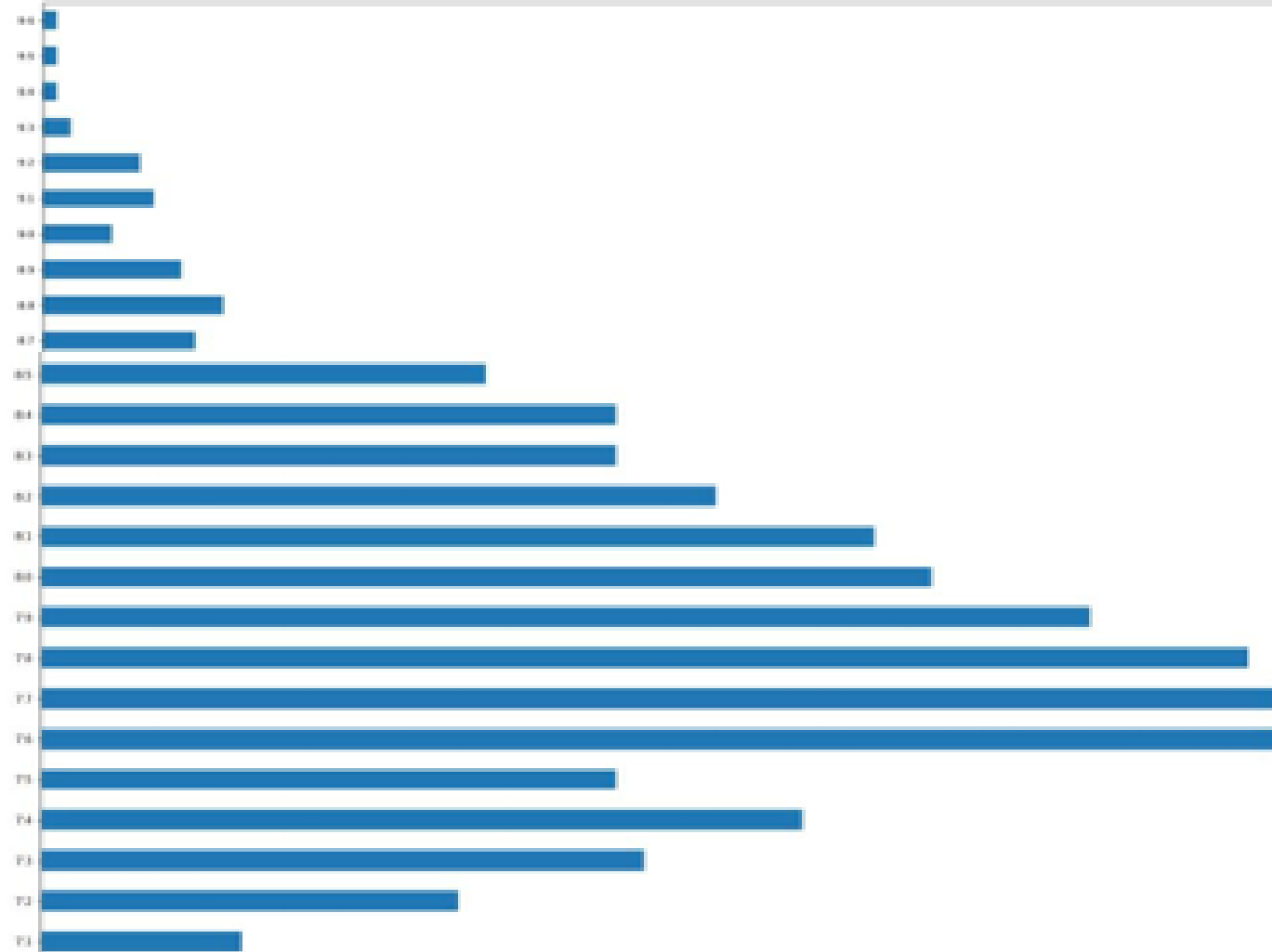


STEP:16

Plotting The Average Rating Graphically

```
In [36]: H plt.figure(figsize=(20, 10))
         avr["average_rating"].value_counts().sort_index().plot.barh()
```

Out[36]:

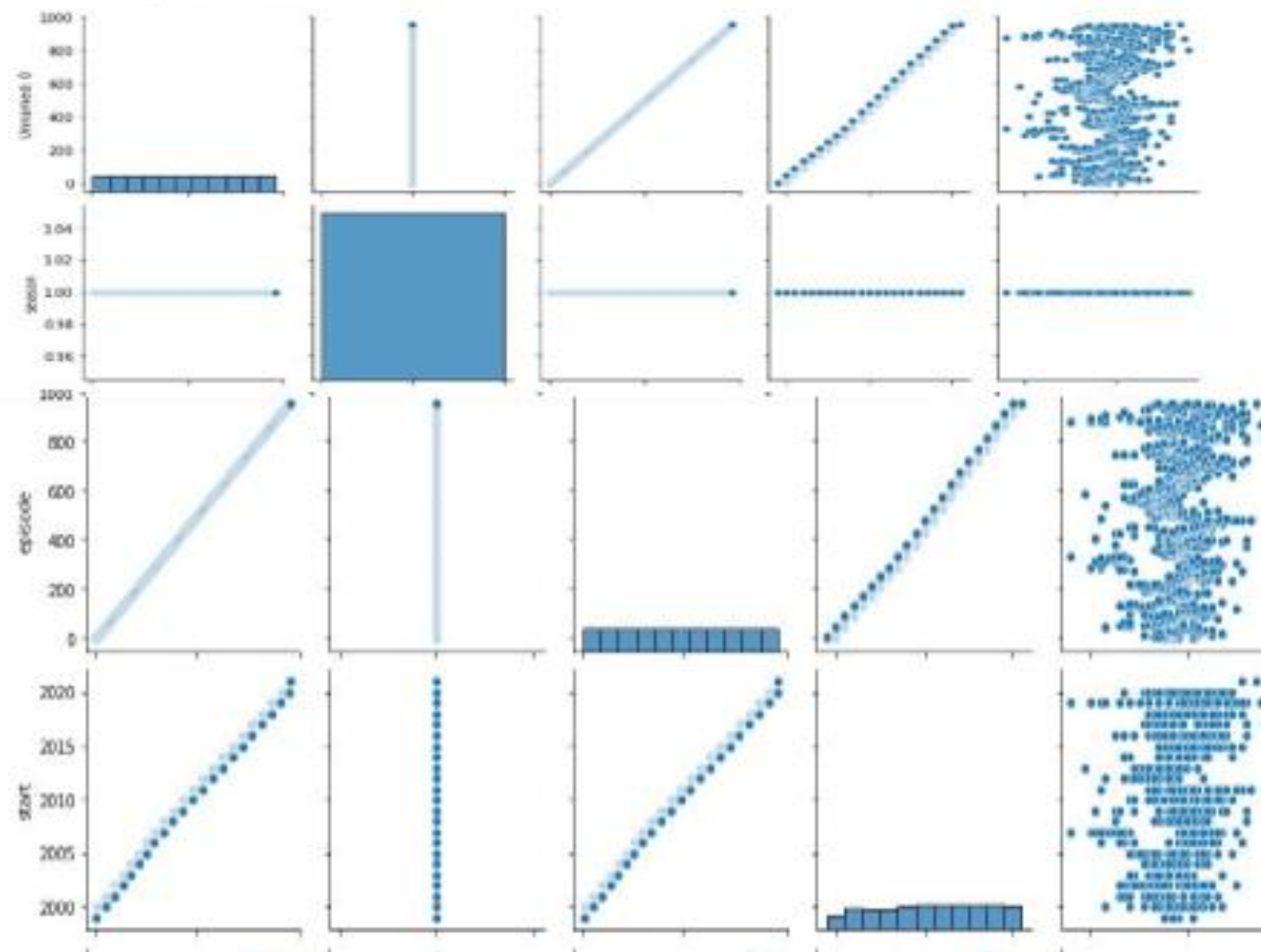


STEP:17

Drawing Attractive Statical Graph

```
[21]: sns.pairplot(anime)
```

```
Out[21]: seaborn.axisgrid.PairGrid at 0xc1f6a1be2e00
```

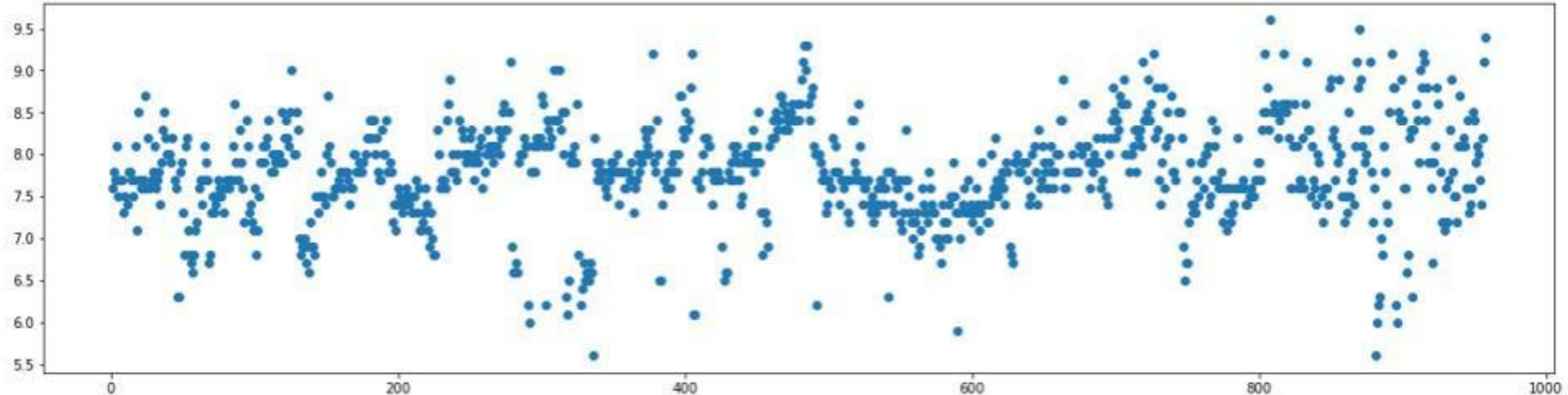


STEP:18

Mathematically using CARTESIAN for Average Rating

```
In [22]: ▶ plt.figure(figsize = (20,5))  
plt.scatter(anime['episode'],anime['average_rating'])
```

```
Out[22]: <matplotlib.collections.PathCollection at 0x1cf6bfb6fd0>
```

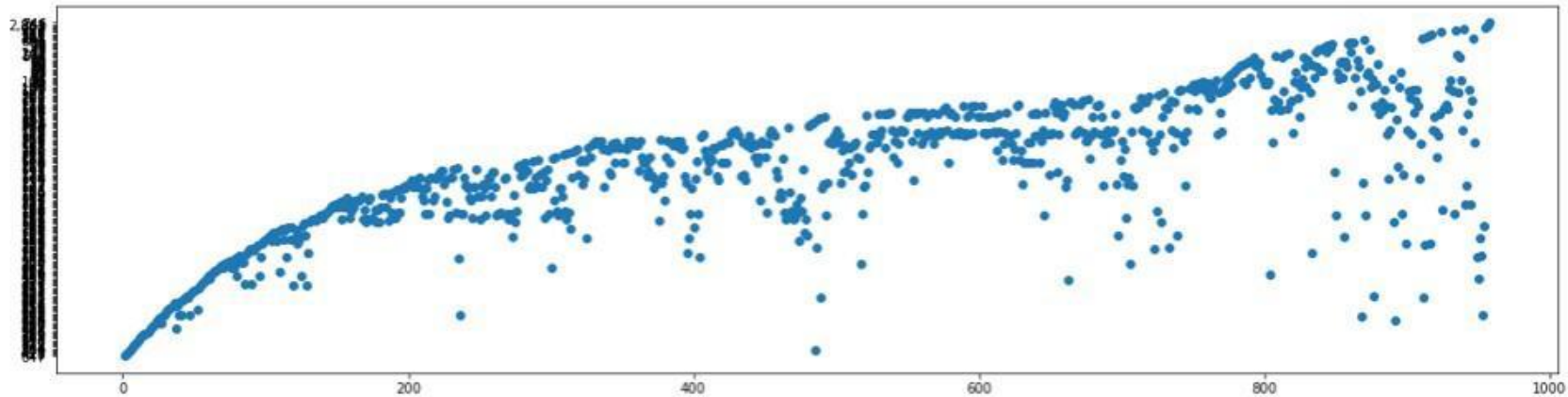


STEP:19

Mathematically using CARTESIAN for Total Votes

```
In [23]: plt.figure(figsize = (20,5))  
plt.scatter(anime['episode'],anime['total_votes'])
```

```
Out[23]: <matplotlib.collections.PathCollection at 0x1cf6c336820>
```

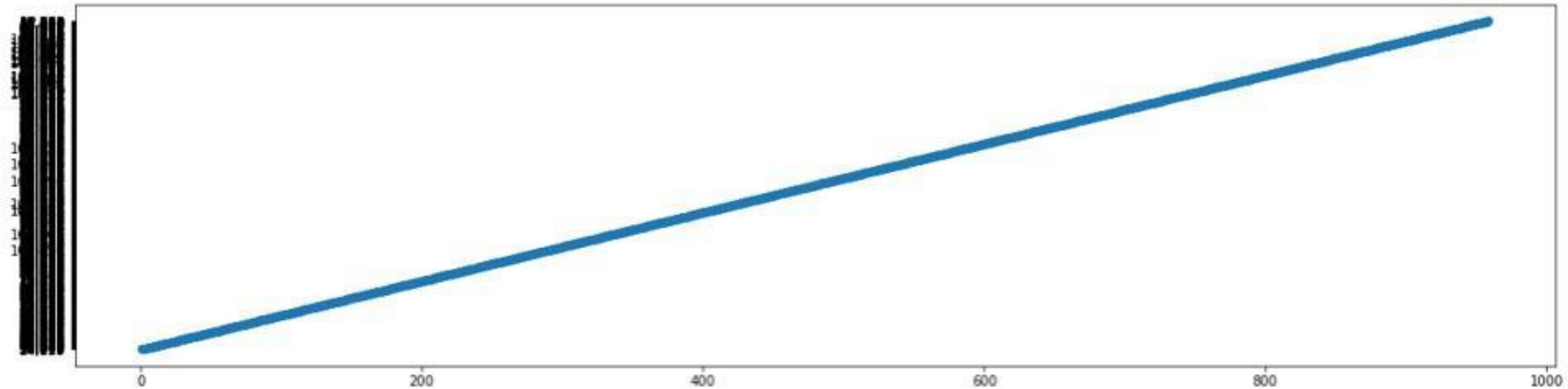


STEP:20

Mathematically using CARTESIAN for Rank

```
In [24]: ▶ plt.figure(figsize = (20,5))  
         plt.scatter(anime['episode'],anime['rank'])
```

```
Out[24]: <matplotlib.collections.PathCollection at 0x1cf6da8b820>
```

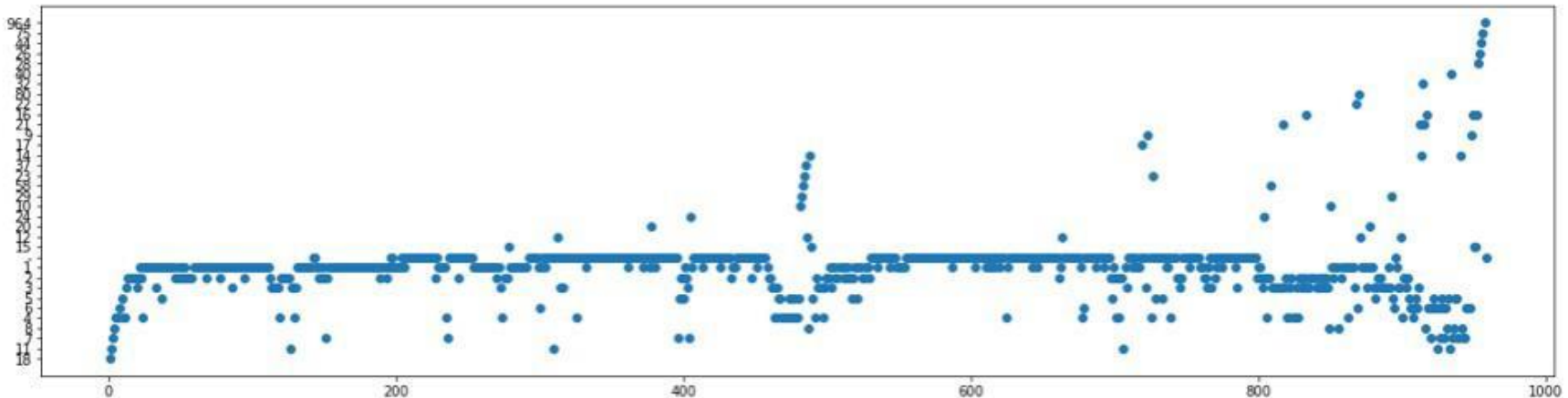


STEP:21

Mathematically using CARTESIAN Trending episodes

```
In [25]: ▶ plt.figure(figsize = (20,5))  
plt.scatter(anime['episode'],anime['trend'])
```

```
Out[25]: <matplotlib.collections.PathCollection at 0x1cf6e2df4c0>
```



STEP:22

Plotting for Votes

```
In [26]: plt.figure(figsize = (50,15))  
anime['total_votes'].value_counts().sort_index().plot.bar()
```

Out[26]: <AxesSubplot:>

