**Pubg - DataScience Project**

*# importing modules*

**import** **pandas** **as** **pd**

**import** **numpy** **as** **np**

**import** **seaborn** **as** **sns**

**import** **matplotlib.pyplot** **as** **plt**

### Task - 1:

#### Read the dataset

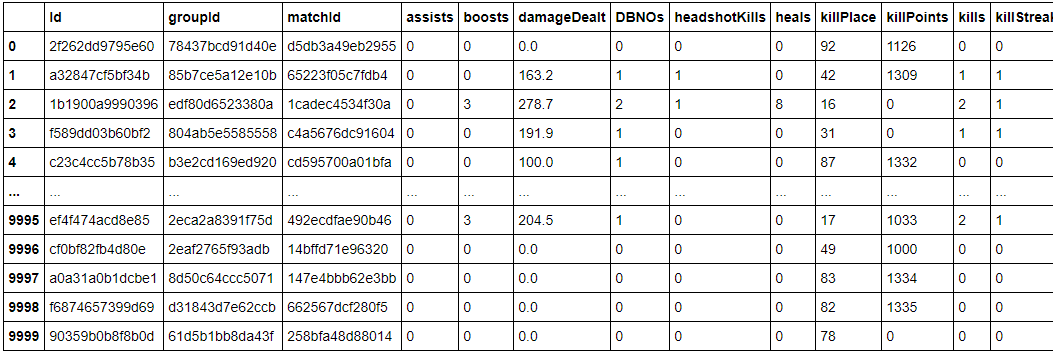
INPUT:-

*# reading csv file using pandas*

df = pd.read\_csv("pubg.csv")

print(df)

OUTPUT:-



**Task - 2:**

**Check the datatype of all the columns.**

**INPUT:-**

*# checking datatype of all columns*

data\_type = df.dtypes

print(data\_type)

**OUTPUT:-**

Id object

groupId object

matchId object

assists int64

boosts int64

damageDealt float64

DBNOs int64

headshotKills int64

heals int64

killPlace int64

killPoints int64

kills int64

killStreaks int64

longestKill float64

matchDuration int64

matchType object

maxPlace int64

numGroups int64

rankPoints int64

revives int64

rideDistance float64

roadKills int64

swimDistance float64

teamKills int64

vehicleDestroys int64

walkDistance float64

weaponsAcquired int64

winPoints int64

winPlacePerc float64

dtype: object

## Task - 3:

#### Find the summary of all the numerical columns and write your findings about it.

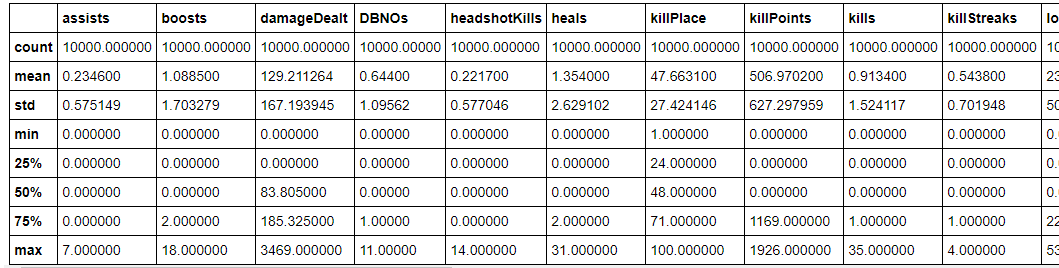
**INPUT:-**

*# DataFrame.describe() is used to give out the summary of all the numerical columns in a dataframe*

num\_summary = df.describe()

print(num\_summary)

**OUTPUT:-**



The mean, min, max, std, count for all numercial column in the dataframe has been displayed.

### **Task - 4:**

#### The average person kills how many players?

**INPUT:-**

avg = df['kills'].mean()

print("**\n**The average person kills :", avg,"player")

**OUTPUT:-**

The average person kills : 0.9134 player

### **Task - 5:**

#### 99% of people have how many kills?

**INPUT:-**

nn\_per = df["kills"].quantile(0.99)

print("**\n**99**% o**f people have",nn\_per,"kills")

OUTPUT:-

99% of people have 7.0 kills

### Task - 6:

#### The most kills ever recorded are how much?

**INPUT**:-

most\_kill = df["kills"].max()

print("**\n**The most kill ever recorded are :",most\_kill)

**OUTPUT**:-

The most kill ever recorded are : 35

### **Task - 7:**

#### Print all the columns of the dataframe.

**INPUT:-**

df.columns

**OUTPUT:-**

Index(['Id', 'groupId', 'matchId', 'assists', 'boosts', 'damageDealt', 'DBNOs',

'headshotKills', 'heals', 'killPlace', 'killPoints', 'kills',

'killStreaks', 'longestKill', 'matchDuration', 'matchType', 'maxPlace',

'numGroups', 'rankPoints', 'revives', 'rideDistance', 'roadKills',

'swimDistance', 'teamKills', 'vehicleDestroys', 'walkDistance',

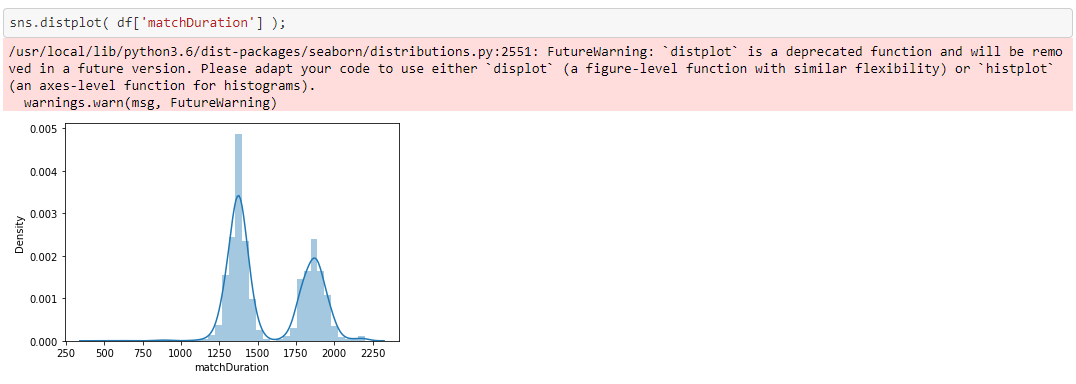
'weaponsAcquired', 'winPoints', 'winPlacePerc'],

dtype='object')

### **Task - 8:**

#### Comment on distribution of the match's duration. Use seaborn.

**INPUT:-**



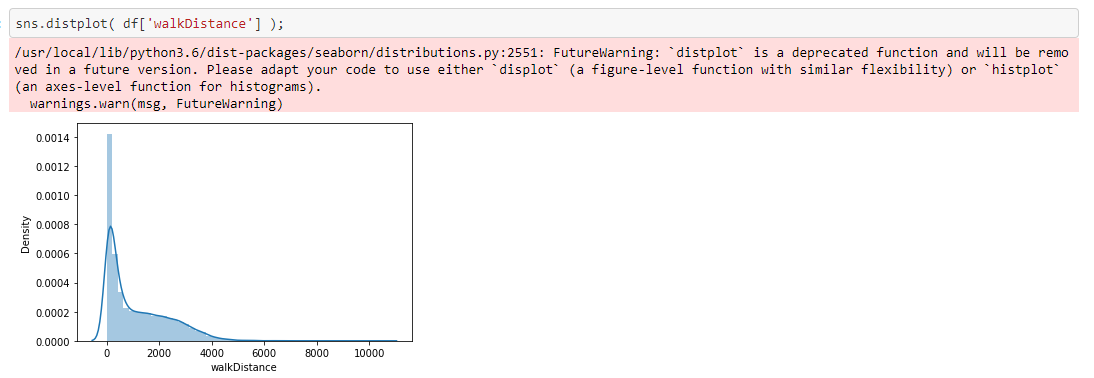
**OUTPUT:-**

Macth duration mostly high during 1250 to 1500

### **Task - 9:**

#### Comment on distribution of the walk distance. Use seaborn.

**INPUT**:-



**OUTPUT**:-

Most of the walking distance lyes between 0 to 2000

### **Task - 10:**

#### Plot distribution of the match's duration vs walk distance one below the other.

**INPUT**:-

%matplotlib inline

plt.style.use('classic')

plt.figure()

*# ploting for matchDuration*

plt.subplot(2,1,1)

plt.plot(df["matchDuration"],"-")

plt.xlabel("Match Duration")

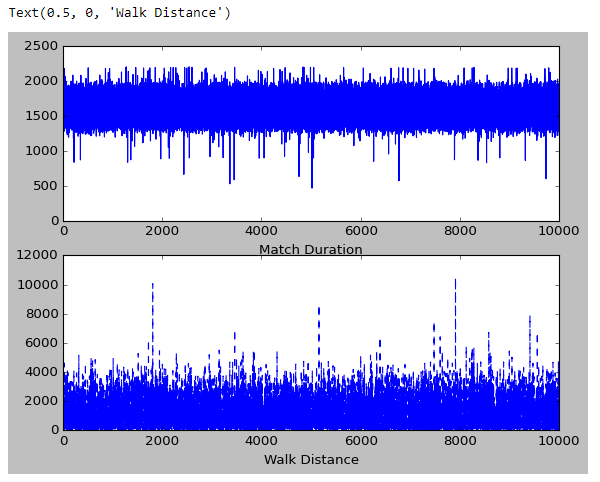
*# ploting for walkDistance*

plt.subplot(2,1,2)

plt.plot(df["walkDistance"],"--")

plt.xlabel("Walk Distance")

**OUTPUT**:-



## Task - 11:

#### Plot distribution of the match's duration vs walk distance side by side.

**INPUT**:-

%matplotlib inline

plt.style.use('classic')

plt.figure(figsize=(10,5))

*# ploting for matchDuration*

plt.subplot(1,2,1)

plt.plot(df["matchDuration"])

plt.xlabel("Match Duration")

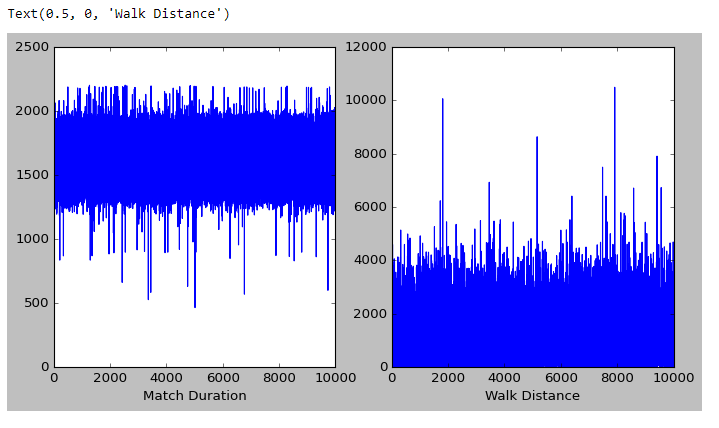
*# ploting for walkDistance*

plt.subplot(1,2,2)

plt.plot(df["walkDistance"])

plt.xlabel("Walk Distance")

**OUTPUT:-**



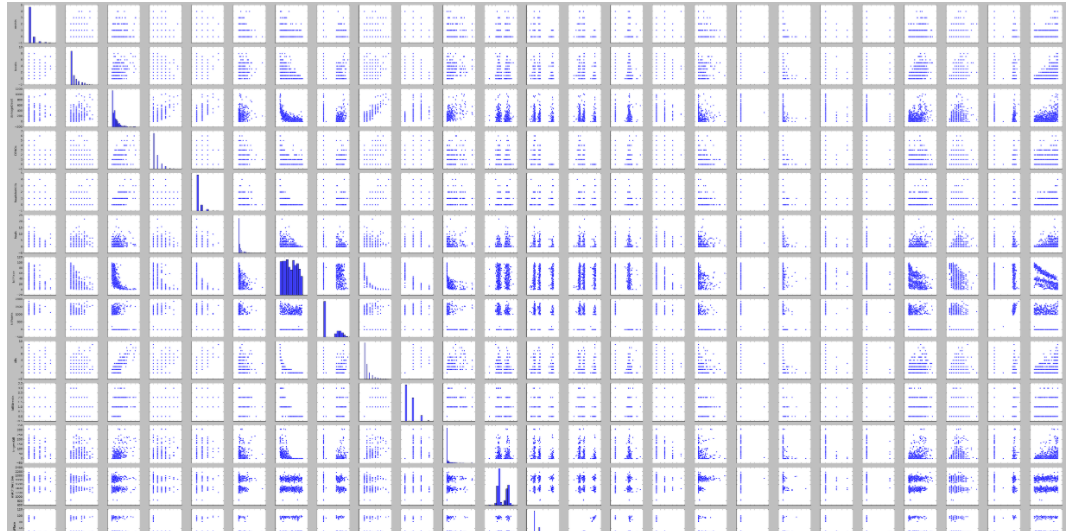
## Task - 12:

#### Pairplot the dataframe. Comment on kills vs damage dealt, Comment on maxPlace vs numGroups.

**INPUT:-**

sns.pairplot(df.head(700));

**OUTPUT:-**



## Task - 13:

#### How many unique values are there in 'matchType' and what are their counts?

**INPUT:-**

uni = pd.unique(df['matchType'])

print("**\n**Unique value in matchType is :",uni)

n\_uni = len(uni)

print("**\n**Count of unique value in matchType is :",n\_uni)

**OUTPUT:-**

Unique value in matchType is : ['squad-fpp' 'squad' 'duo-fpp' 'solo-fpp' 'duo' 'solo' 'crashfpp'

'flaretpp' 'normal-squad-fpp' 'normal-duo-fpp' 'flarefpp' 'normal-squad'

'normal-solo-fpp' 'crashtpp']

Count of unique value in matchType is : 14

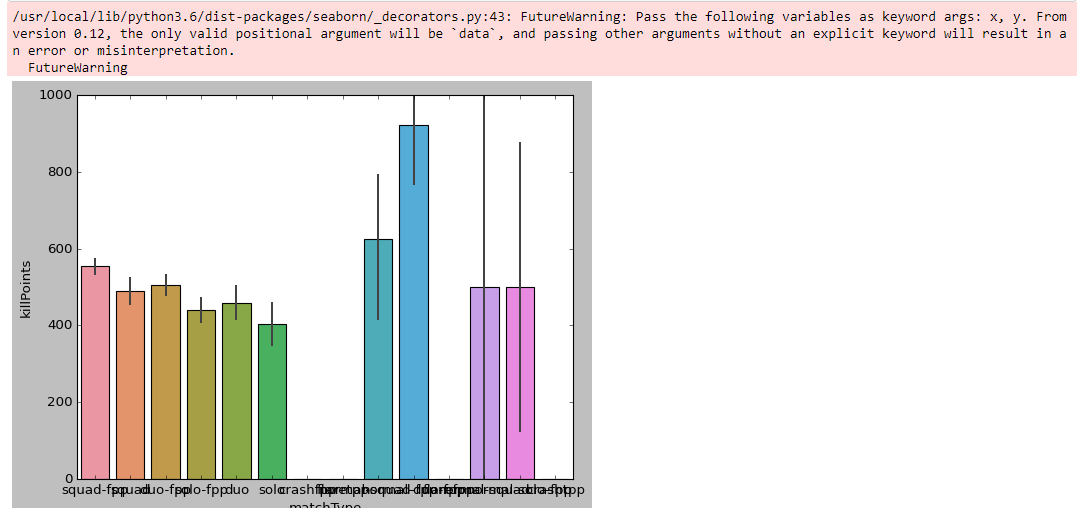
## Task - 14:

#### Plot a barplot of ‘matchType’ vs 'killPoints'. Write your inferences.

**INPUT:-**

sns.barplot(df['matchType'],df['killPoints']);

**OUTPUT:-**



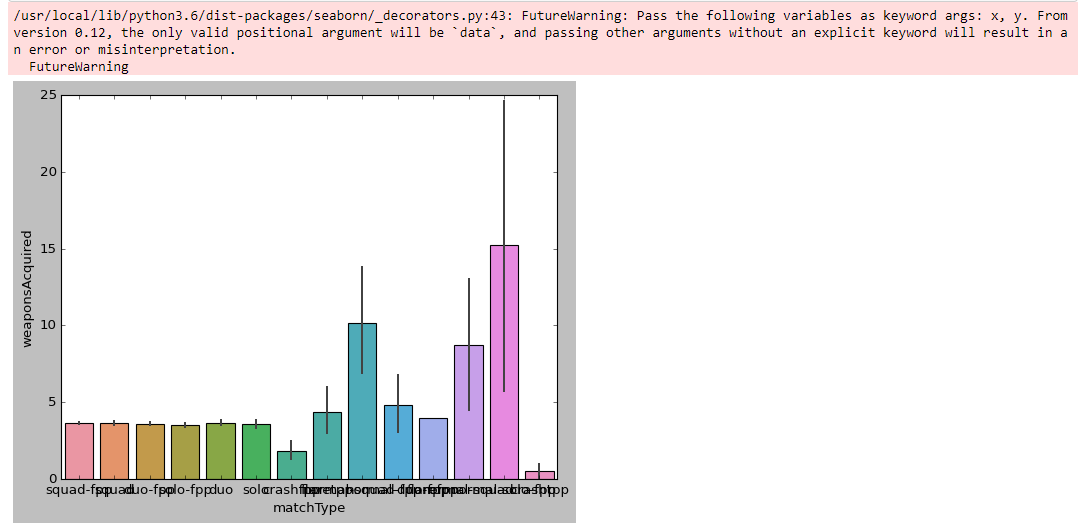
## Task - 15:

#### Plot a barplot of ‘matchType’ vs ‘weaponsAcquired’. Write your inferences.

**INPUT:-**

sns.barplot(df['matchType'],df['weaponsAcquired']);

**OUTPUT:-**



## Task - 16:

#### Find the Categorical columns.

**INPUT:-**

cat\_col = df.select\_dtypes('category').columns

print(cat\_col)

**OUTPUT:-**

Index([], dtype='object')

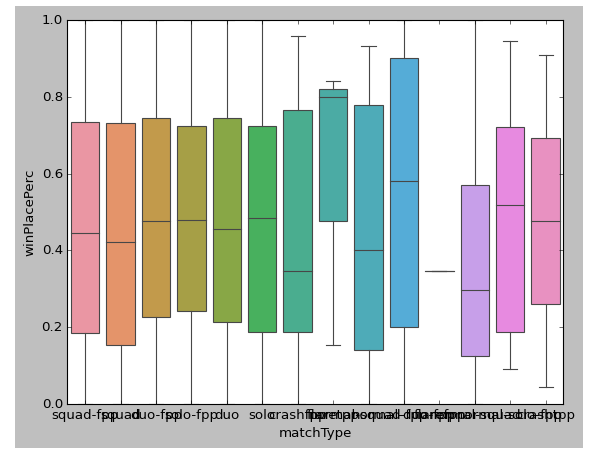
## Task - 17:

#### Plot a boxplot of ‘matchType’ vs ‘winPlacePerc’. Write your inferences.

**INPUT:-**

sns.boxplot(x='matchType', y='winPlacePerc', data=df);

**OUTPUT:-**



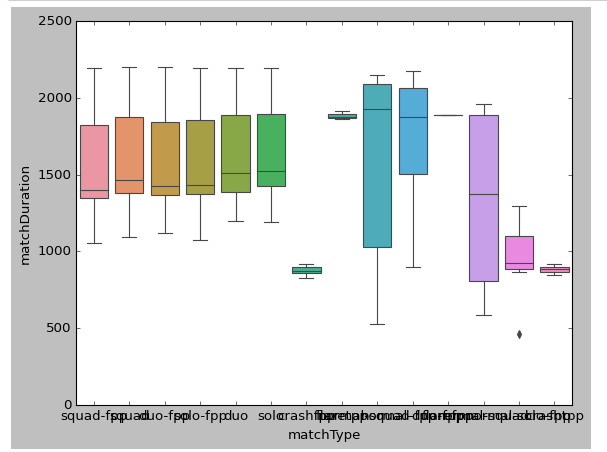
## Task - 18:

#### Plot a boxplot of ‘matchType’ vs ‘matchDuration’. Write your inferences.

**INPUT:-**

sns.boxplot(x='matchType', y='matchDuration', data=df);

**OUTPUT:-**



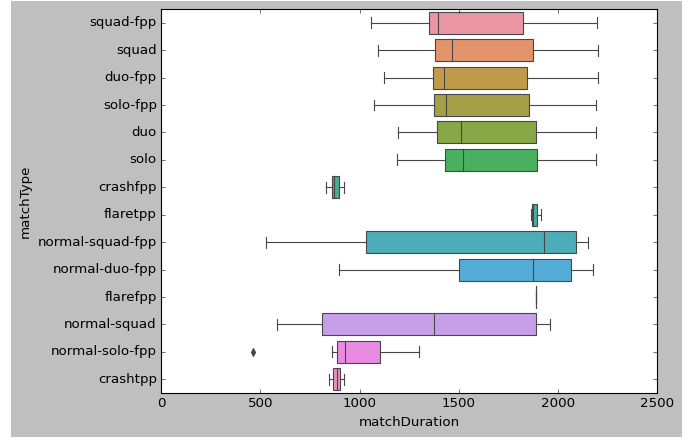
## Task - 19:

#### Change the orientation of the above plot to horizontal.

**INPUT:-**

sns.boxplot( x='matchDuration', y='matchType',data=df);

**OUTPUT:-**



## Task - 20:

#### Add a new column called ‘KILL’ which contains the sum of following columns viz. headshotKills, teamKills, roadKills.

**INPUT:-**

df['KILL'] = df['headshotKills'] + df['teamKills'] + df['roadKills']

df['KILL']

**OUTPUT:-**

0 0

1 1

2 1

3 0

4 0

..

9995 0

9996 0

9997 0

9998 0

9999 0

Name: KILL, Length: 10000, dtype: int64

## Task - 21:

#### Round off column ‘winPlacePerc’ to 2 decimals.

**INPUT:-**

df['winPlacePerc'].round(decimals=2)

**OUTPUT:-**

0 0.00

1 0.22

2 0.86

3 0.35

4 0.07

...

9995 0.83

9996 0.72

9997 0.21

9998 0.24

9999 0.19

Name: winPlacePerc, Length: 10000, dtype: float64

## Task - 22:

#### Take a sample of size 50 from the column damageDealt for 100 times and calculate its mean. Plot it on a histogram and comment on its distribution.

**INPUT:-**

mean = []

**for** i **in** range(100):

**for** x **in** range(0,1001,50):

mean1 = df['damageDealt'].head(x).mean()

mean.append(mean1)

sns.histplot(data=df['damageDealt'], x=mean)

**OUTPUT:-**

