In [1]:

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

Task-1

In [6]:
```
import pandas as pd

data=pd.read_csv("D:\\rohit.csv")
df = pd.DataFrame(data)
print(df)
```

```
                   Id         groupId         matchId  assists  boosts  \
0      2f262dd9795e60  78437bcd91d40e  d5db3a49eb2955        0       0   
1      a32847cf5bf34b  85b7ce5a12e10b  65223f05c7fdb4        0       0   
2      1b1900a9990396  edf80d6523380a  1cadec4534f30a        0       3   
3      f589dd03b60bf2  804ab5e5585558  c4a5676dc91604        0       0   
4      c23c4cc5b78b35  b3e2cd169ed920  cd595700a01bfa        0       0   
...               ...             ...             ...      ...     ...   
9995   ef4f474acd8e85  2eca2a8391f75d  492ecdfae90b46        0       3   
9996   cf0bf82fb4d80e  2eaf2765f93adb  14bffd71e96320        0       0   
9997   a0a31a0b1dcbe1  8d50c64ccc5071  147e4bbb62e3bb        0       0   
9998   f6874657399d69  d31843d7e62ccb  662567dcf280f5        0       0   
9999   90359b0b8f8b0d  61d5b1bb8da43f  258bfa48d88014        0       0   

      damageDealt  DBNOs  headshotKills  heals  killPlace  ...  revives  \
0             0.0      0              0      0         92  ...        0   
1           163.2      1              1      0         42  ...        0   
2           278.7      2              1      8         16  ...        3   
3           191.9      1              0      0         31  ...        0   
4           100.0      1              0      0         87  ...        0   
...           ...    ...            ...    ...        ...  ...      ...   
9995        204.5      1              0      0         17  ...        3   
9996          0.0      0              0      0         49  ...        0   
9997          0.0      0              0      0         83  ...        0   
9998          0.0      0              0      0         82  ...        0   
9999          0.0      0              0      0         78  ...        0   

      rideDistance  roadKills  swimDistance  teamKills vehicleDestroys  \
0              0.0          0           0.0          0               0   
1              0.0          0           0.0          0               0   
2              0.0          0           0.0          0               0   
3              0.0          0           0.0          0               0   
4              0.0          0           0.0          0               0   
...            ...        ...           ...        ...             ...   
9995           0.0          0           0.0          0               0   
9996           0.0          0           0.0          0               0   
9997           0.0          0           0.0          0               0   
9998           0.0          0           0.0          0               0   
9999           0.0          0           0.0          0               0   

      walkDistance  weaponsAcquired  winPoints  winPlacePerc  
0             0.00                0       1470        0.0000  
1           132.70                2       1531        0.2222  
2          3591.00               10          0        0.8571  
3           332.70                3          0        0.3462  
4           252.70                3       1557        0.0690  
...            ...              ...        ...           ...  
9995       1648.00                4       1471        0.8333  
9996        897.10                6       1500        0.7174  
9997        188.20                2       1434        0.2083  
9998        108.10                5       1534        0.2449  
```

```
9999         53.36                 1          0        0.1875

[10000 rows x 29 columns]
```

Task-2

In [7]:

```
#Check the datatype of all the columns
data_type = df.dtypes
data_type
```

Out[7]:

```
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

In [8]:
```
#Find the summary of all the numerical columns and write your findings about it.
num_summary = df.describe()
num_summary
```

Out[8]:

	assists	boosts	damageDealt	DBNOs	headshotKills	heals	ki
count	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.000000	10000.0
mean	0.234600	1.088500	129.211264	0.64400	0.221700	1.354000	47.0
std	0.575149	1.703279	167.193945	1.09562	0.577046	2.629102	27.4
min	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	1.0
25%	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	24.0
50%	0.000000	0.000000	83.805000	0.00000	0.000000	0.000000	48.0
75%	0.000000	2.000000	185.325000	1.00000	0.000000	2.000000	71.0
max	7.000000	18.000000	3469.000000	11.00000	14.000000	31.000000	100.0

8 rows × 25 columns

Task-4

In [9]:
```
#The average person kills how many players
avg = df['kills'].mean()
print("\nThe average person kills :", avg,"player")
```

```

The average person kills : 0.9134 player
```

Task-5

In [10]:
```
#99% of people have how many kills
nn_per = df["kills"].quantile(0.99)
print("\n99% of people have",nn_per,"kills")
```

```

99% of people have 7.0 kills
```

Task-6

```
In [11]: most_kill = df["kills"].max()
         print("\nThe most kill ever recorded are :",most_kill)
```

```

The most kill ever recorded are : 35
```

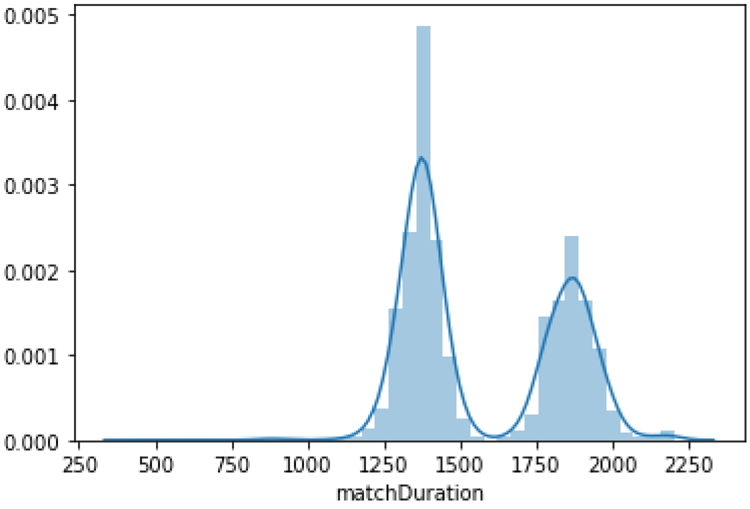
Task-7

```
In [12]: # The most kills ever recorded are how much
         df.columns
```

```
Out[12]: 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

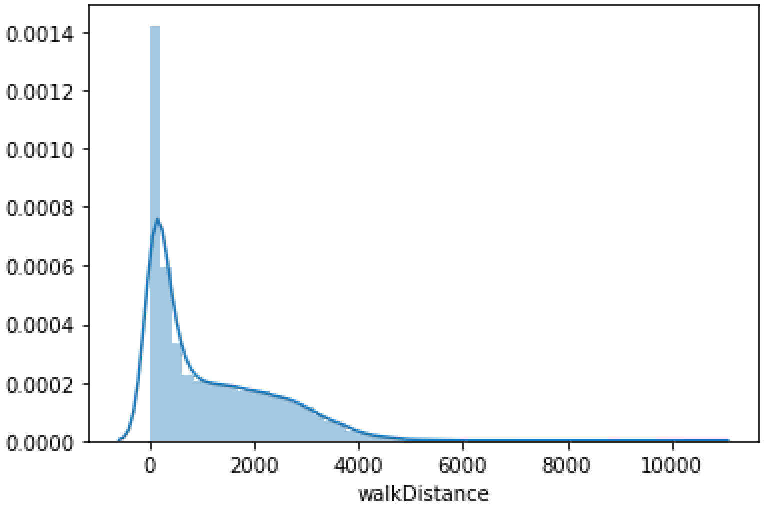
```
In [13]: #Comment on distribution of the match's duration. Use seaborn.

         import seaborn as sns
         sns.distplot( df['matchDuration'] );
```



Task-9

In [14]:

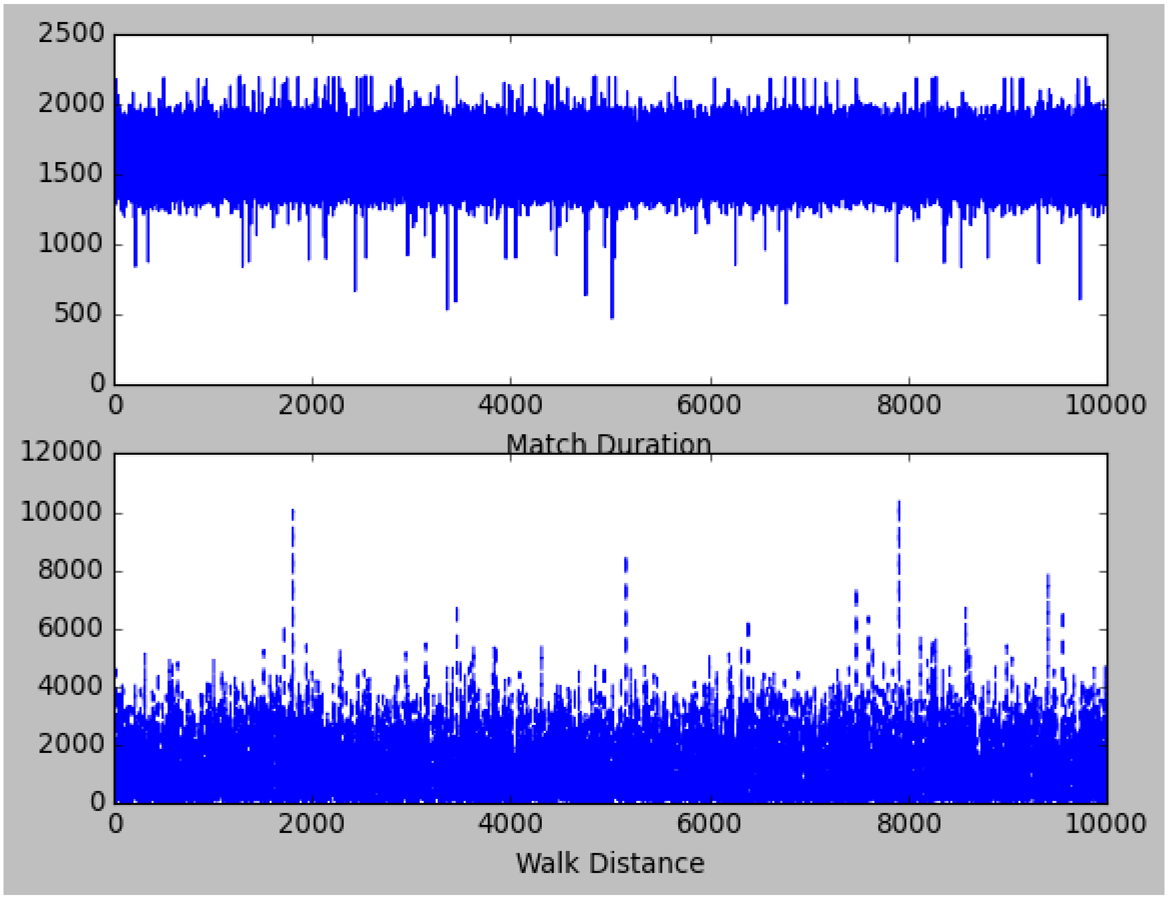
```
#Comment on distribution of the walk distance. Use seaborn.
sns.distplot( df['walkDistance'] );
```



Task-10

```
In [15]: #Plot distribution of the match's duration vs walk distance one below the other.

         %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")
```

```
Out[15]: Text(0.5, 0, 'Walk Distance')
```



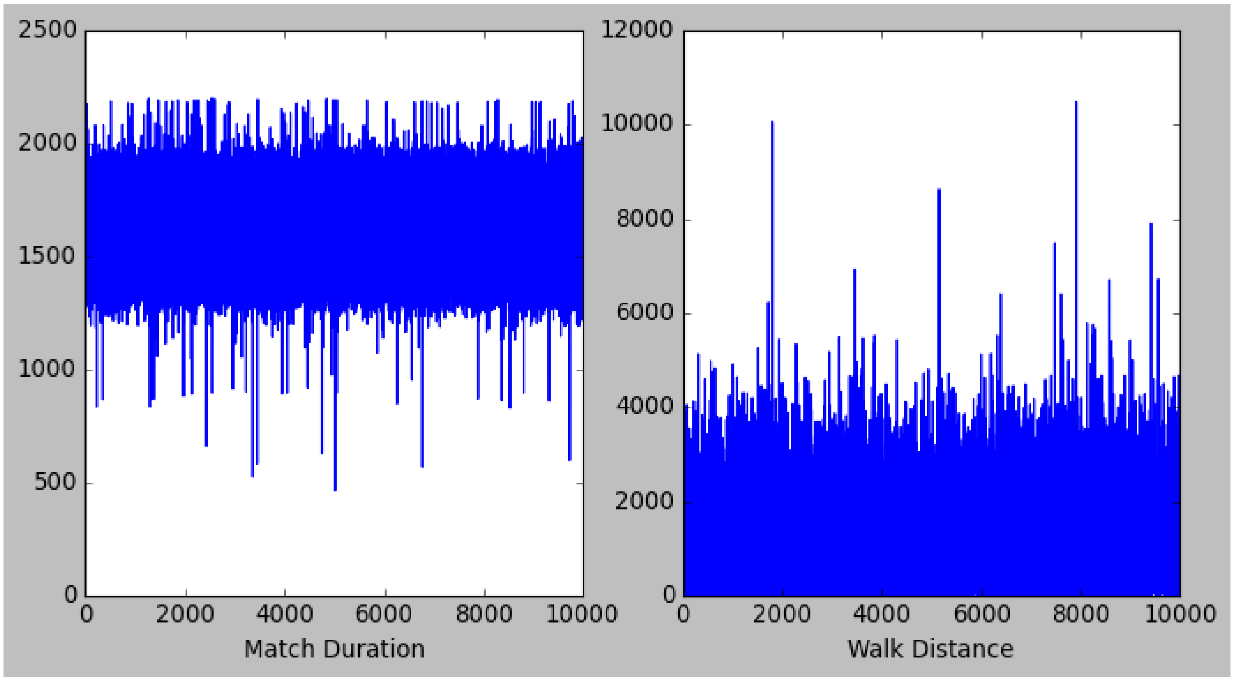
Task-11

In [16]:

```
#Plot distribution of the match's duration vs walk distance side by side.
%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")
```

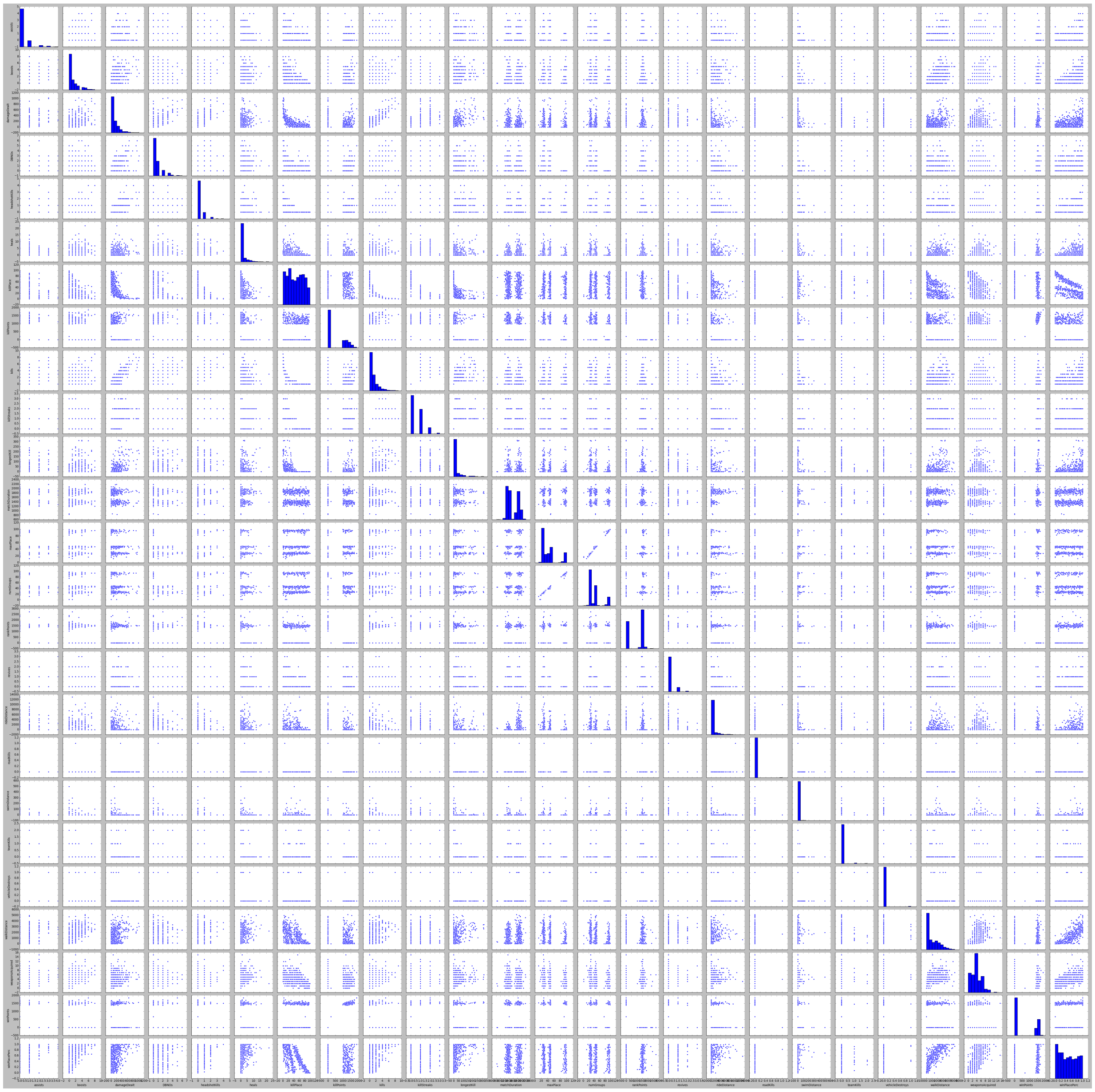
Out[16]:

```
Text(0.5, 0, 'Walk Distance')
```



Task-12

```
In [17]: #Pairplot the dataframe. Comment on kills vs damage dealt, Comment on maxPlace vs

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



Task-13

In [18]:

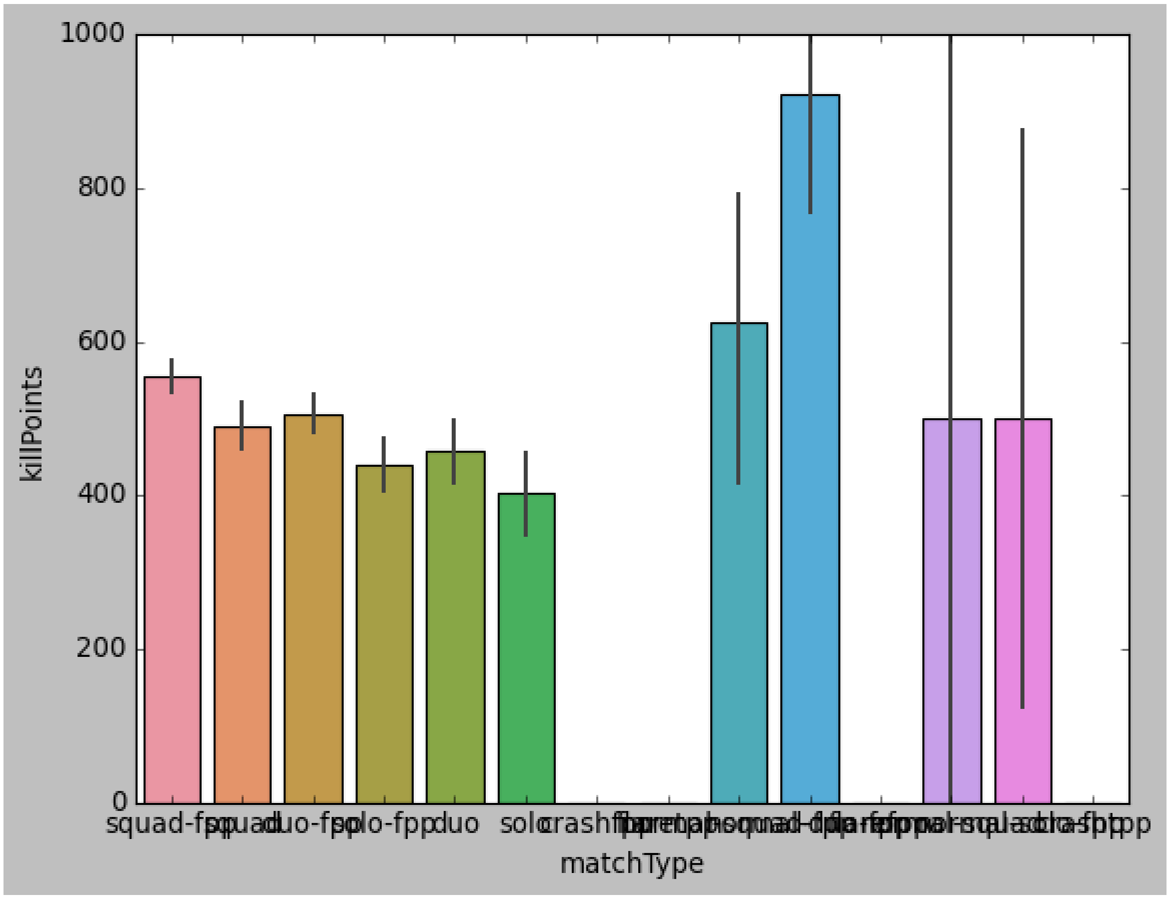
```
#How many unique values are there in 'matchType' and what are their counts

uni = pd.unique(df['matchType'])
print("\nUnique value in matchType is :",uni)
n_uni = len(uni)
print("\nCount of unique value in matchType is :",n_uni)
```

```
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

In [19]:

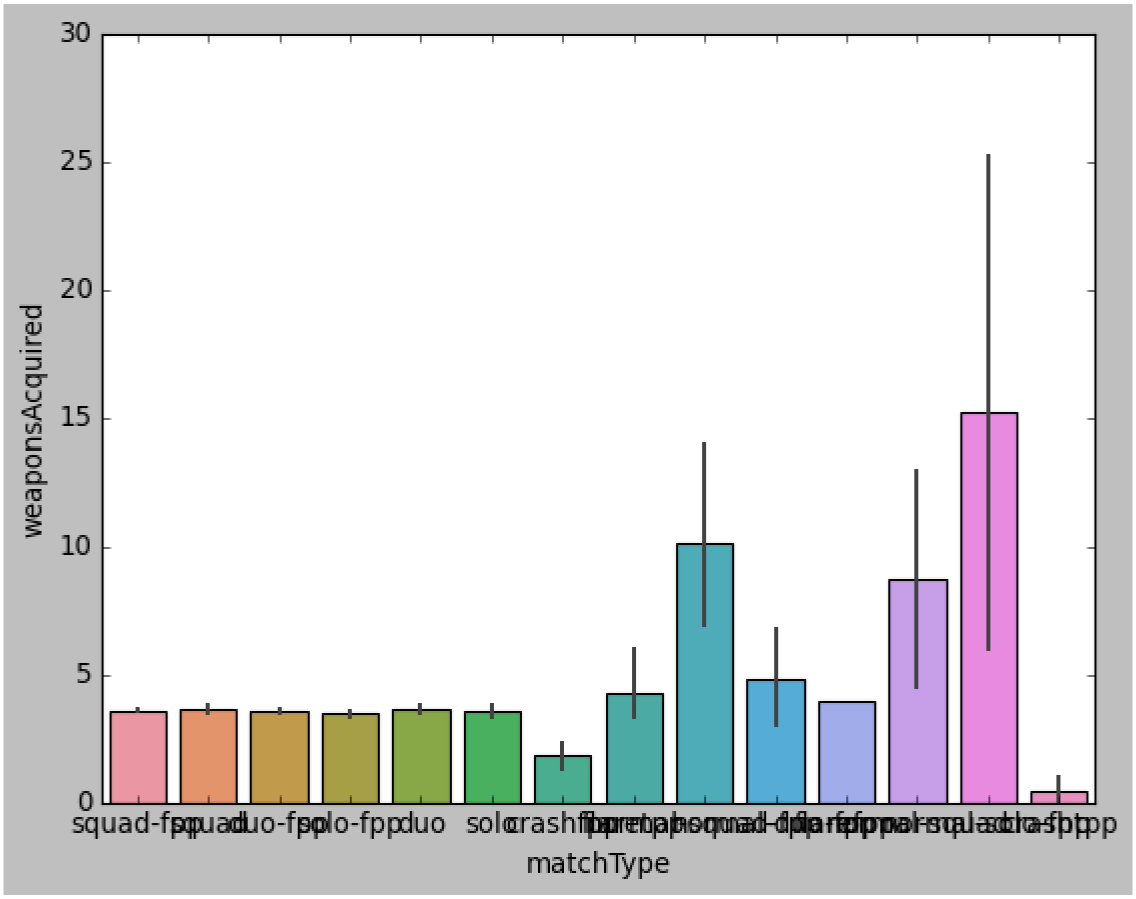
```
#Plot a barplot of ‘matchType’ vs 'killPoints'. Write your inferences.
sns.barplot(df['matchType'],df['killPoints']);
```



Task-15

In [20]:

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

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



Task-16

In [21]:

```
#Find the Categorical columns.
cat_col = df.select_dtypes('category').columns
cat_col
```

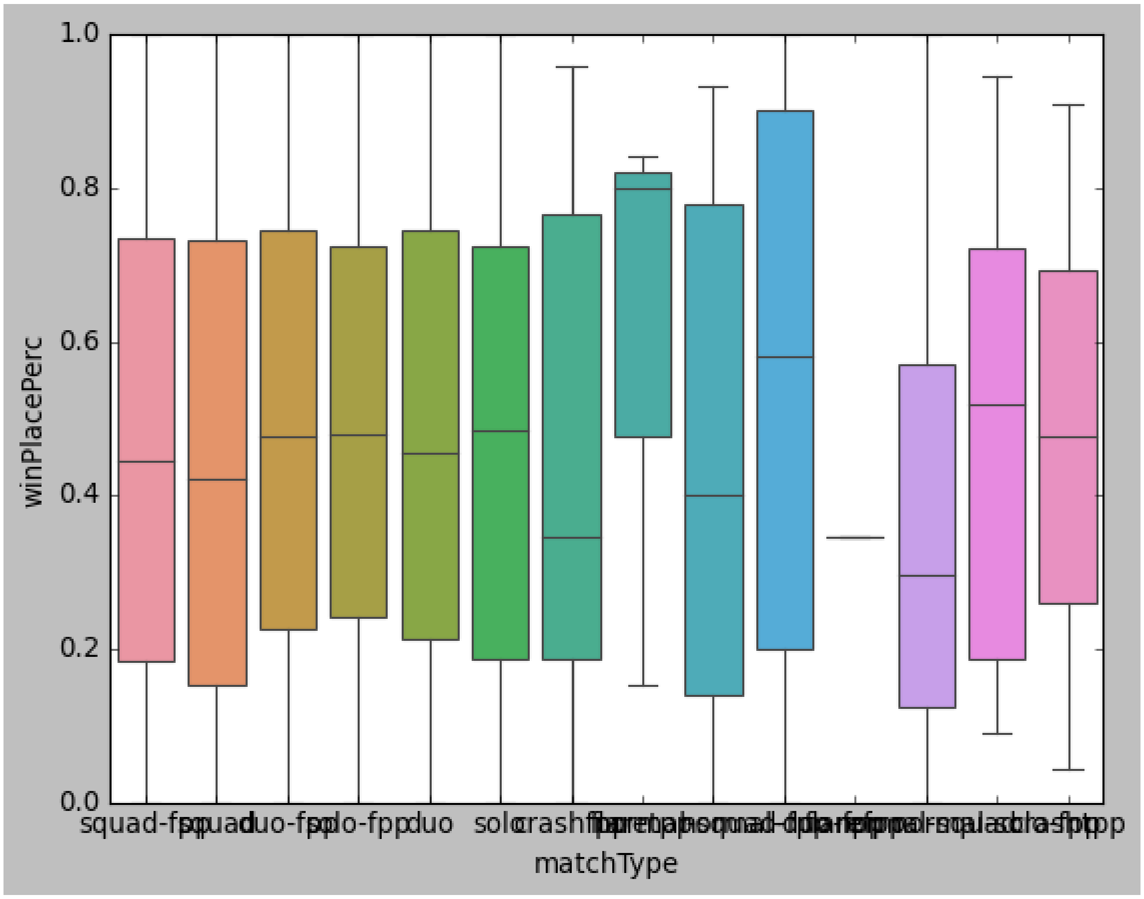
Out[21]:

```
Index([], dtype='object')
```

Task-17

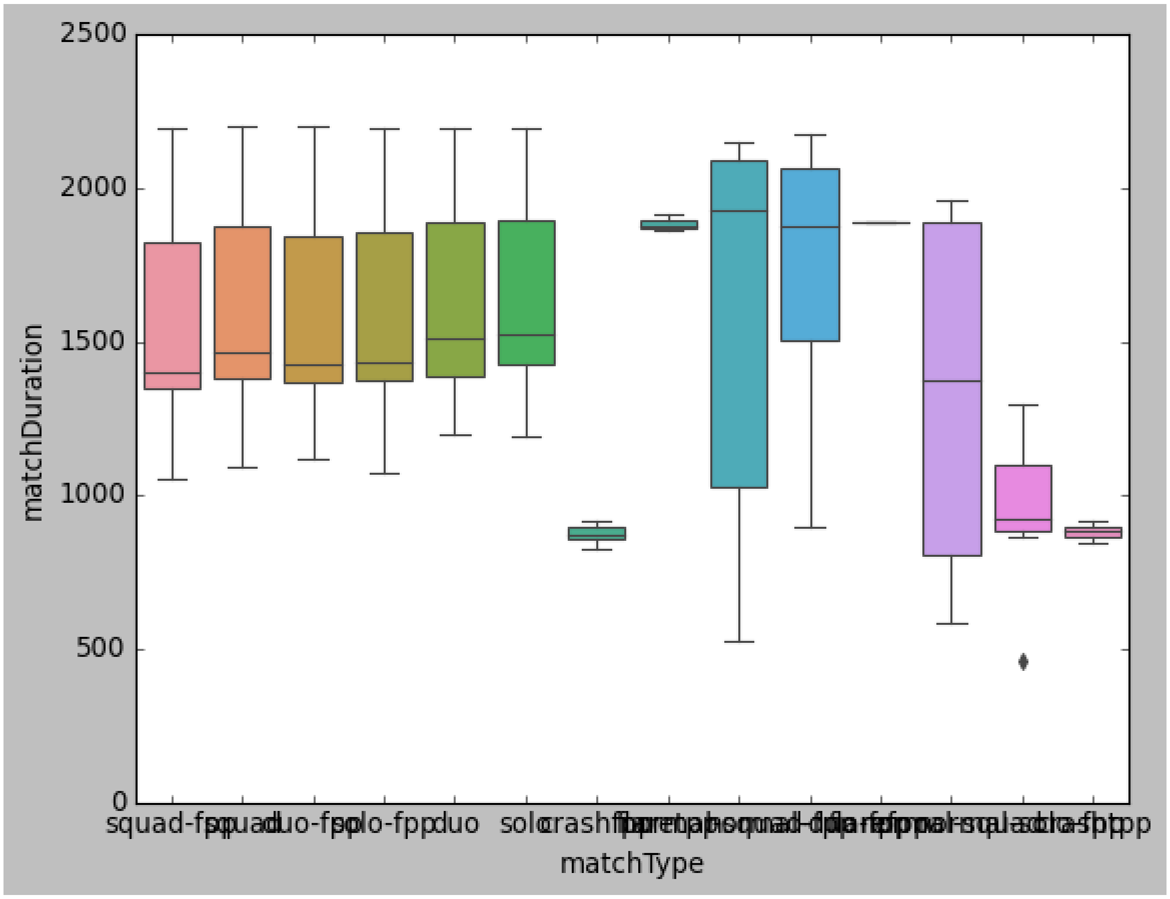
In [22]:

```
# Plot a boxplot of ‘matchType’ vs ‘winPlacePerc’. Write your inferences.
sns.boxplot(x='matchType', y='winPlacePerc', data=df);
```



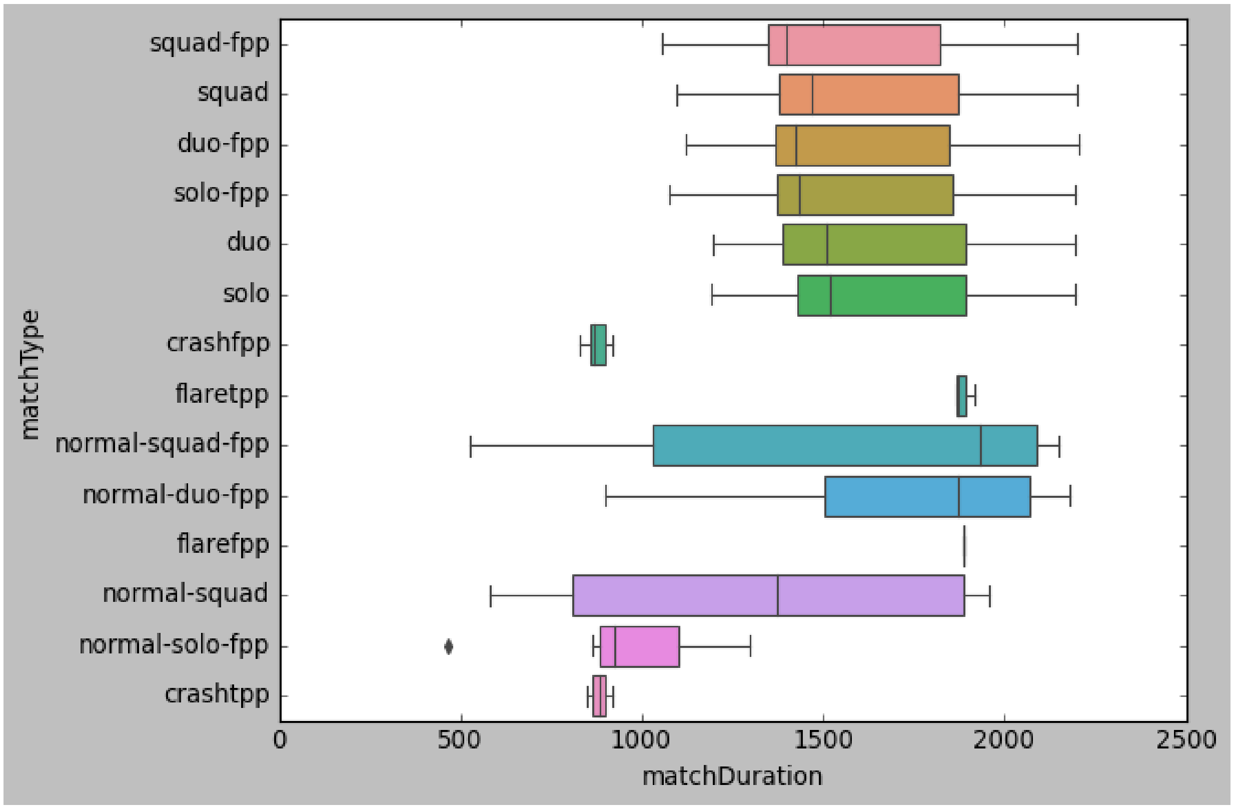
Task-18

```
In [31]: #Plot a boxplot of ‘matchType’ vs ‘matchDuration’. Write your inferences.
         sns.boxplot(x='matchType', y='matchDuration', data=df);
```



Task-19

```
In [23]: # Change the orientation of the above plot to horizontal.
         sns.boxplot( x='matchDuration', y='matchType',data=df);
```



Task-20

In [24]:
```
# Add a new column called ‘KILL’ which contains the sum of following columns viz.
df['KILL'] = df['headshotKills'] + df['teamKills'] + df['roadKills']
df['KILL']
```

Out[24]:
```
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

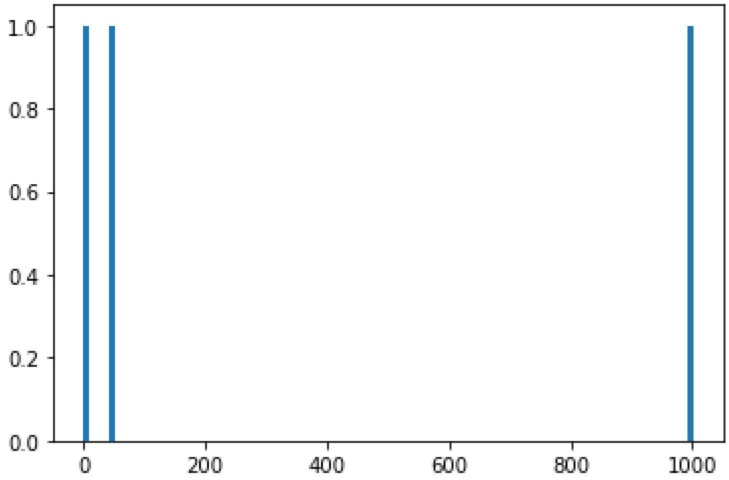
In [25]:
```
# Round off column ‘winPlacePerc’ to 2 decimals.
df['winPlacePerc'].round(decimals=2)
```

Out[25]:
```
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

In [24]:

```
#Take a sample of size 50 from the column damageDealt for 100 times and calculate
import matplotlib.pyplot as plt
m
x = [0,1001,50]
plt.hist(x, bins=100)
plt.show()
```



In [ ]: