Project 1

Importing Relevant Library

In [15]:
```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
plt.rcParams['figure.figsize'] = (16,8)
```

1. Read the dataset

In [17]:
```
pubg = pd.read_csv("C:\\Users\\Admin\\Downloads\pubg - Dr. Darshan Ingle.csv")
pd.set_option('display.max_columns',100)
pubg.head(10)
```

Out[17]:

	Id	groupId	matchId	assists	boosts	damageDealt	DBNOs	head
0	2f262dd9795e60	78437bcd91d40e	d5db3a49eb2955	0	0	0.00	0	
1	a32847cf5bf34b	85b7ce5a12e10b	65223f05c7fdb4	0	0	163.20	1	
2	1b1900a9990396	edf80d6523380a	1cadec4534f30a	0	3	278.70	2	
3	f589dd03b60bf2	804ab5e5585558	c4a5676dc91604	0	0	191.90	1	
4	c23c4cc5b78b35	b3e2cd169ed920	cd595700a01bfa	0	0	100.00	1	
5	fd034582dd4d2e	9b8930aeee086a	6f6e52b15ddf21	0	1	200.00	2	
6	c60b5633f4dcc8	7c0f817f6627c7	3232c1e0fec04b	0	3	638.20	4	
7	f0ba8246b6980f	7318b5204462cb	112e9711f86001	0	0	27.94	0	
8	79c5d5eda1c72e	a85b81198dfc06	ef5fc25e28ffb1	1	4	275.80	3	
9	94834a28e52abd	bc513cde35fa54	f36a754a9b88f7	1	1	530.40	4	

2. Check the datatype of all the columns.

In [18]:
```
pubg.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 29 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Id               10000 non-null  object 
 1   groupId          10000 non-null  object 
 2   matchId          10000 non-null  object 
 3   assists          10000 non-null  int64  
 4   boosts           10000 non-null  int64  
 5   damageDealt      10000 non-null  float64
 6   DBNOs            10000 non-null  int64  
 7   headshotKills    10000 non-null  int64  
 8   heals            10000 non-null  int64  
 9   killPlace        10000 non-null  int64  
 10  killPoints       10000 non-null  int64  
 11  kills            10000 non-null  int64  
 12  killStreaks      10000 non-null  int64  
 13  longestKill      10000 non-null  float64
 14  matchDuration    10000 non-null  int64  
 15  matchType        10000 non-null  object 
 16  maxPlace         10000 non-null  int64  
 17  numGroups        10000 non-null  int64  
 18  rankPoints       10000 non-null  int64  
 19  revives          10000 non-null  int64  
 20  rideDistance     10000 non-null  float64
 21  roadKills        10000 non-null  int64  
 22  swimDistance     10000 non-null  float64
 23  teamKills        10000 non-null  int64  
 24  vehicleDestroys  10000 non-null  int64  
 25  walkDistance     10000 non-null  float64
 26  weaponsAcquired  10000 non-null  int64  
 27  winPoints        10000 non-null  int64  
 28  winPlacePerc     10000 non-null  float64
dtypes: float64(6), int64(19), object(4)
memory usage: 2.1+ MB
```

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

In [19]:
```
pubg.describe()
```

Out[19]:

	assists	boosts	damageDealt	DBNOs	headshotKills	heals	ki
count	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.000000	10000.(
mean	0.234600	1.088500	129.211264	0.64400	0.221700	1.354000	47.(
std	0.575149	1.703279	167.193945	1.09562	0.577046	2.629102	27.·
min	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	1.(
25%	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	24.(
50%	0.000000	0.000000	83.805000	0.00000	0.000000	0.000000	48.(
75%	0.000000	2.000000	185.325000	1.00000	0.000000	2.000000	71.(
max	7.000000	18.000000	3469.000000	11.00000	14.000000	31.000000	100.(

4. The average person kills how many players?

In [20]:
```
pubg['Id'].nunique()
```

Out[20]: 10000

In [21]:
```
pubg['kills'].mean()
```

Out[21]: 0.9134

5. 99% of people have how many kills?

In [22]:
```
pubg['kills'].quantile(0.99)
```

Out[22]: 7.0

6. The most kills ever recorded are how much?

In [23]:
```
pubg['kills'].max()
```

Out[23]: 35

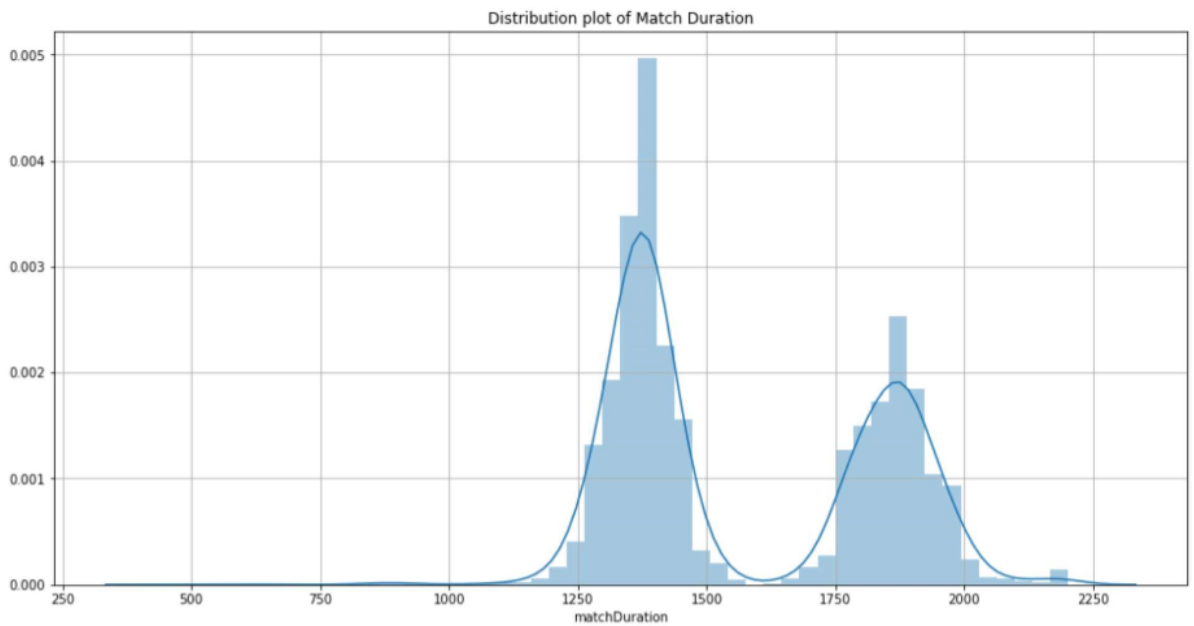
Print all the columns of the dataframe

In [24]:
```
pubg.columns
```

Out[24]:
```
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')
```

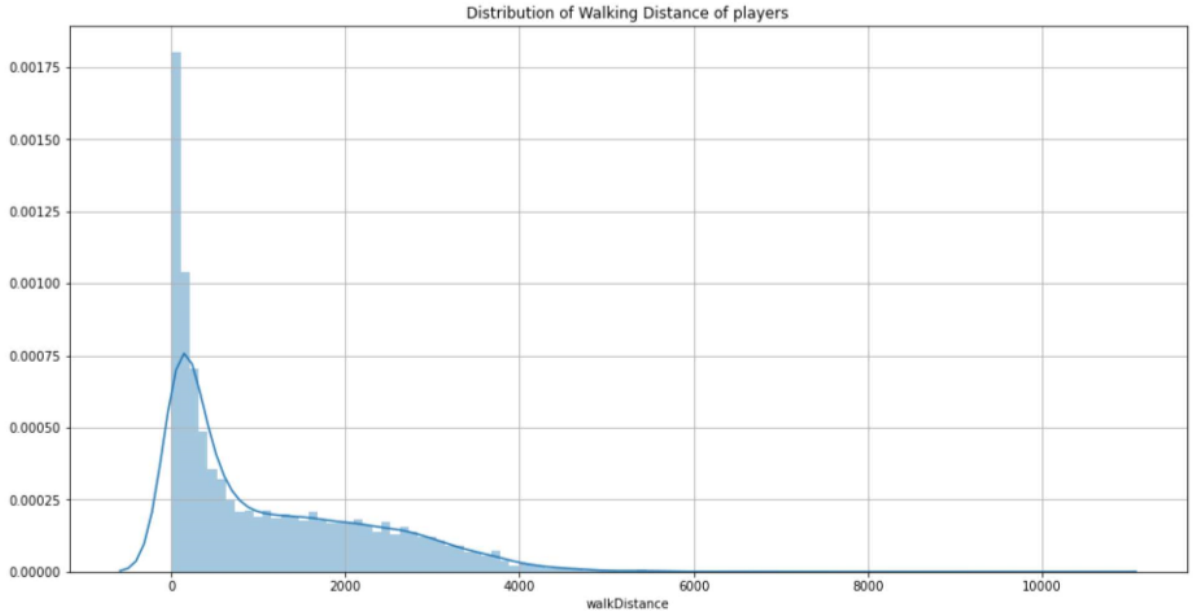
Comment on distribution of the match's duration. Use seaborn

In [27]:
```
sns.distplot(pubg['matchDuration'], bins=50)
plt.grid()
plt.title('Distribution plot of Match Duration');
```



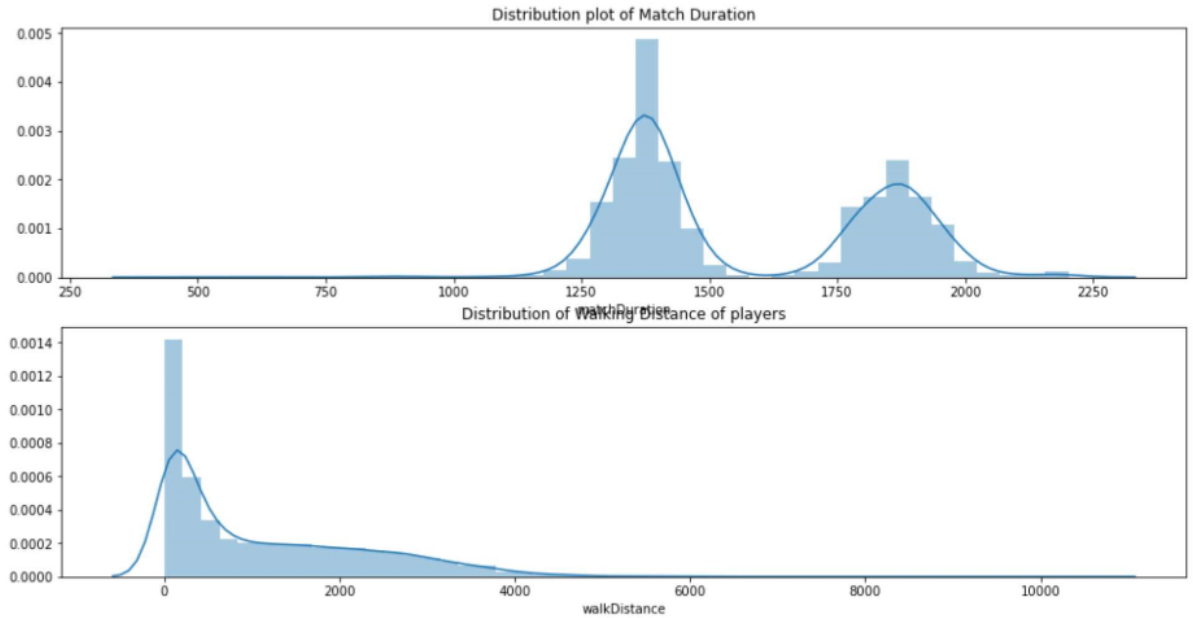
9. Comment on distribution of the walk distance. Use seaborn

In [26]:
```
sns.distplot(pubg['walkDistance'], bins=100)
plt.grid()
plt.title('Distribution of Walking Distance of players');
```



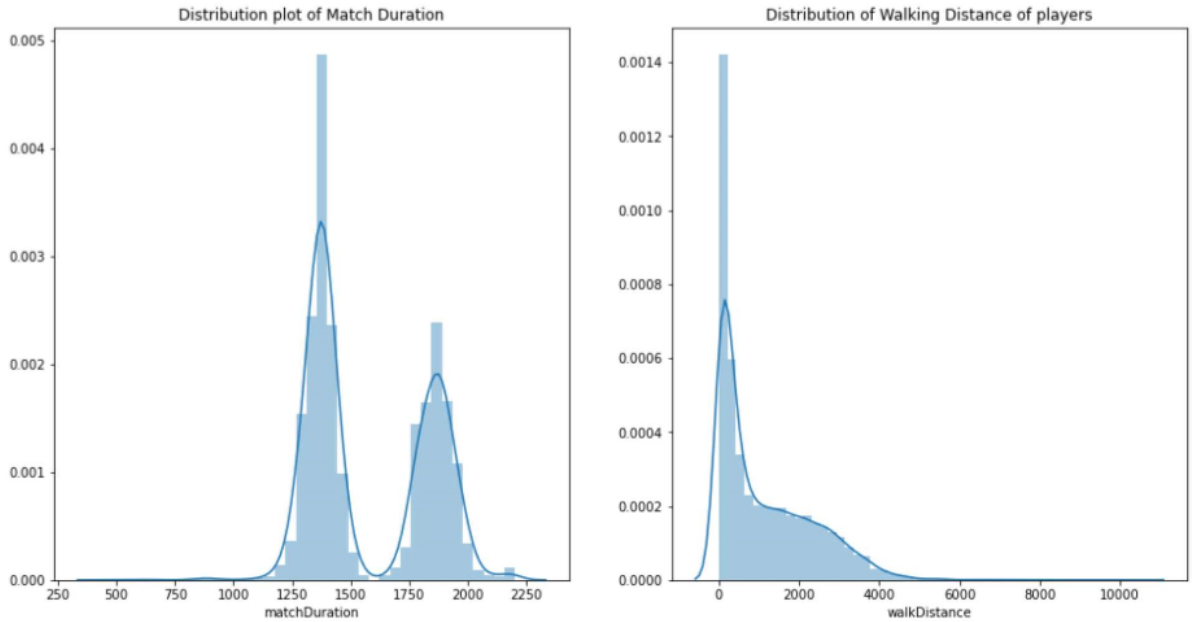
Plot distribution of the match's duration vs walk distance one below the other.

In [28]:
```
fig, ax = plt.subplots(2,1)
sns.distplot(pubg['matchDuration'], ax=ax[0]).set_title('Distribution plot of Mat
sns.distplot(pubg['walkDistance'],ax=ax[1]).set_title('Distribution of Walking Di
plt.show()
```



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

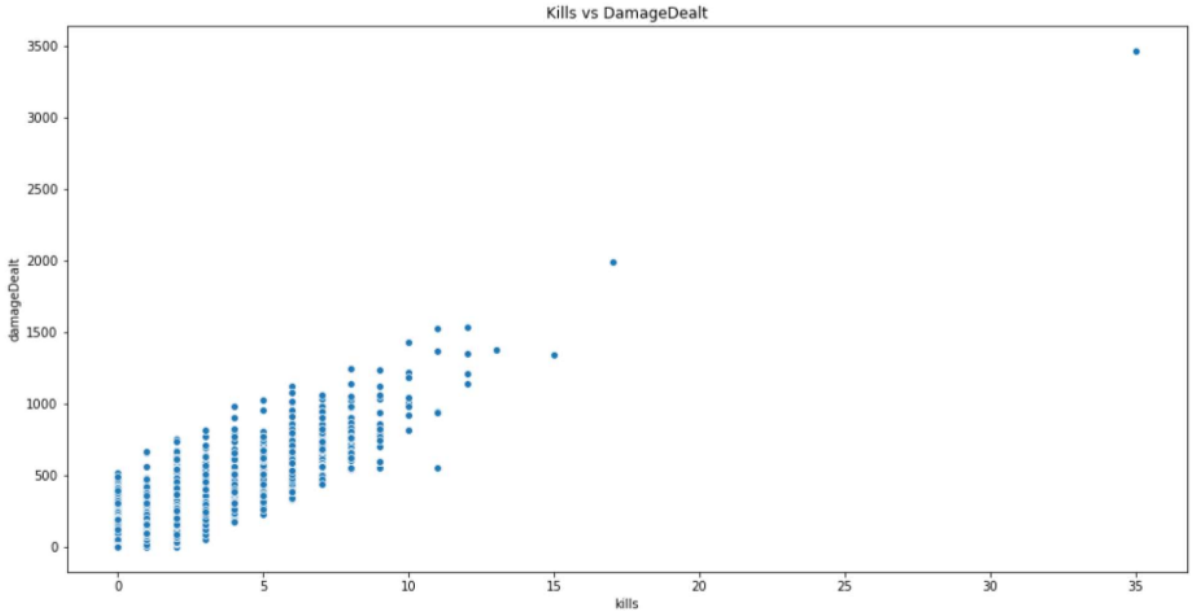
In [29]:
```
fig, ax = plt.subplots(1,2)
sns.distplot(pubg['matchDuration'], ax=ax[0]).set_title('Distribution plot of Mat
sns.distplot(pubg['walkDistance'],ax=ax[1]).set_title('Distribution of Walking Di
plt.show()
```



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

kills vs damage dealt

In [30]:
```
sns.scatterplot(x='kills',y='damageDealt', data=pubg).set_title('Kills vs DamageD
#as two columns are contain numerical value. thats why i use scatterplot.
```

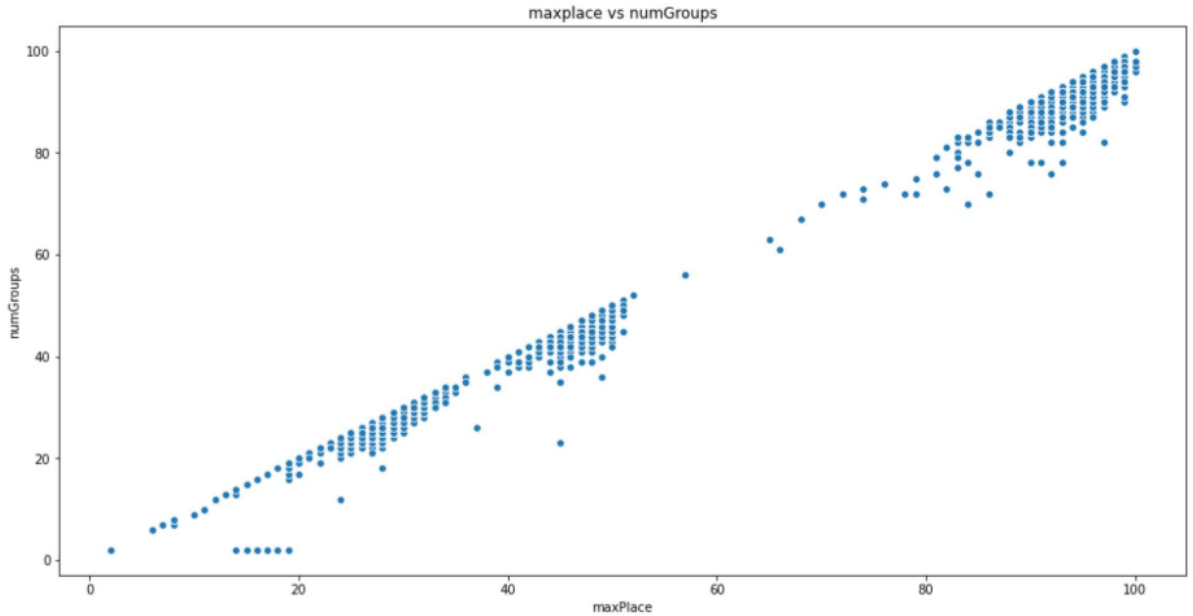


In [31]:
```
pubg['kills'].corr(pubg['damageDealt'])
```

Out[31]:
```
0.883371087672403
```

maxPlace vs numGroups

In [33]:
```
sns.scatterplot(x='maxPlace', y='numGroups', data=pubg).set_title('maxplace vs nu
```



In [34]:
```
pubg['maxPlace'].corr(pubg['numGroups'])
```

Out[34]:
```
0.9980078437439721
```

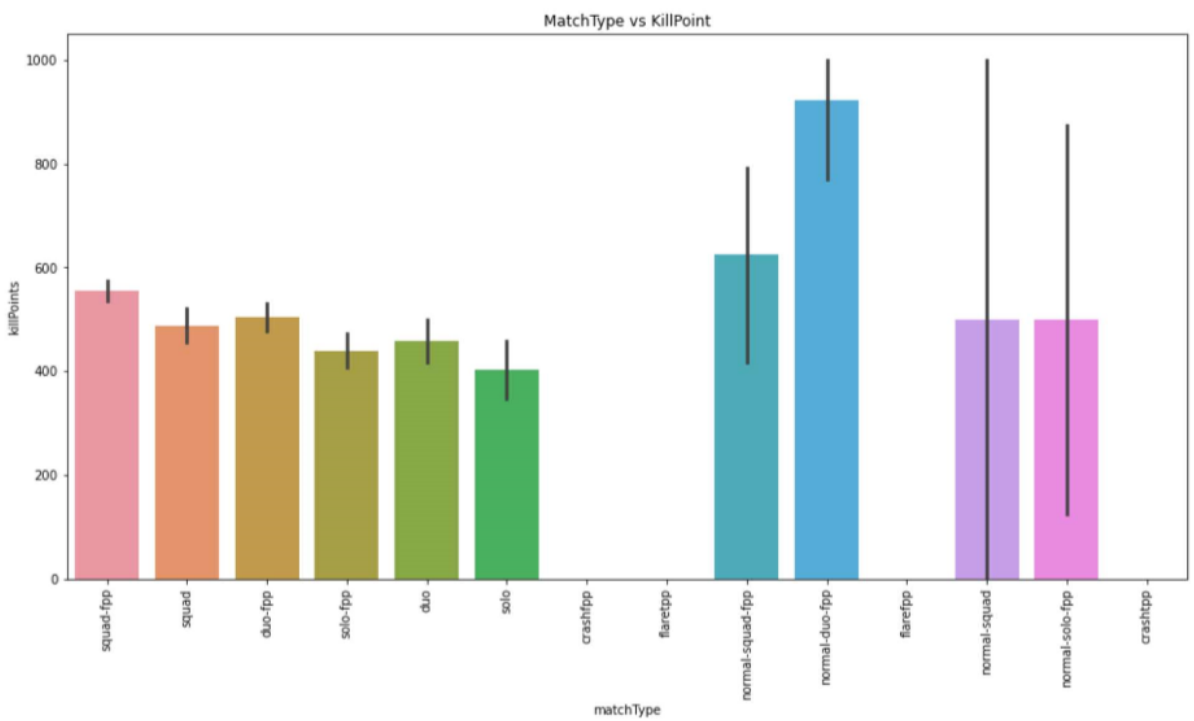
13. How many unique values are there in 'matchType' and what are their counts?

In [35]:
```
pubg['matchType'].value_counts()
```

Out[35]:
```
squad-fpp           3969
duo-fpp             2282
squad               1359
solo-fpp            1234
duo                  702
solo                 386
normal-squad-fpp      24
crashfpp              13
normal-duo-fpp        13
normal-solo-fpp        8
normal-squad           4
flaretpp               3
crashtpp               2
flarefpp               1
Name: matchType, dtype: int64
```

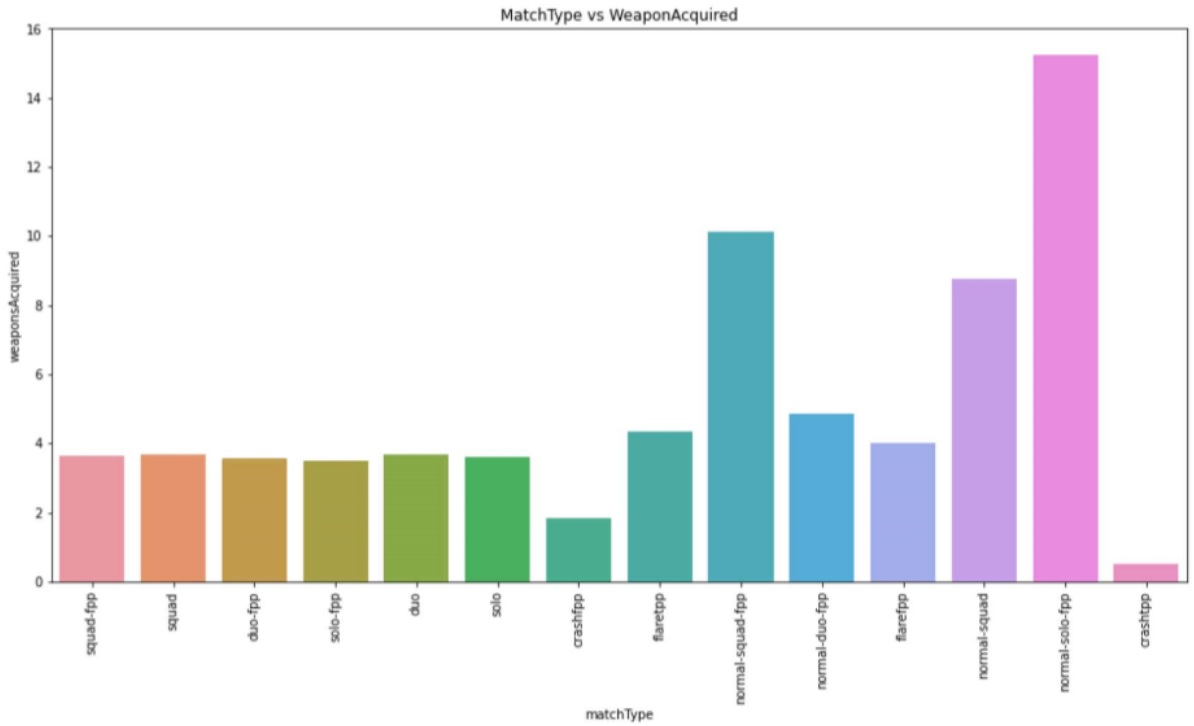
14. Plot a barplot of ‘matchType’ vs 'killPoints'. Write your inferences.

In [37]:
```
sns.barplot(x='matchType',y='killPoints', data=pubg).set_title('MatchType vs Kil
plt.xticks(rotation=90);
```



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

In [38]:
```
sns.barplot(x='matchType',y='weaponsAcquired', data=pubg,ci=None).set_title('Mat
plt.xticks(rotation=90);
```



16. Find the Categorical columns.

In [40]:
```
print('Only Category columns are in the dataset: ',pubg.select_dtypes(['object'])
pubg.select_dtypes(['object'])
```

```
Only Category columns are in the dataset:  Index(['Id', 'groupId', 'matchId',
'matchType'], dtype='object')
```

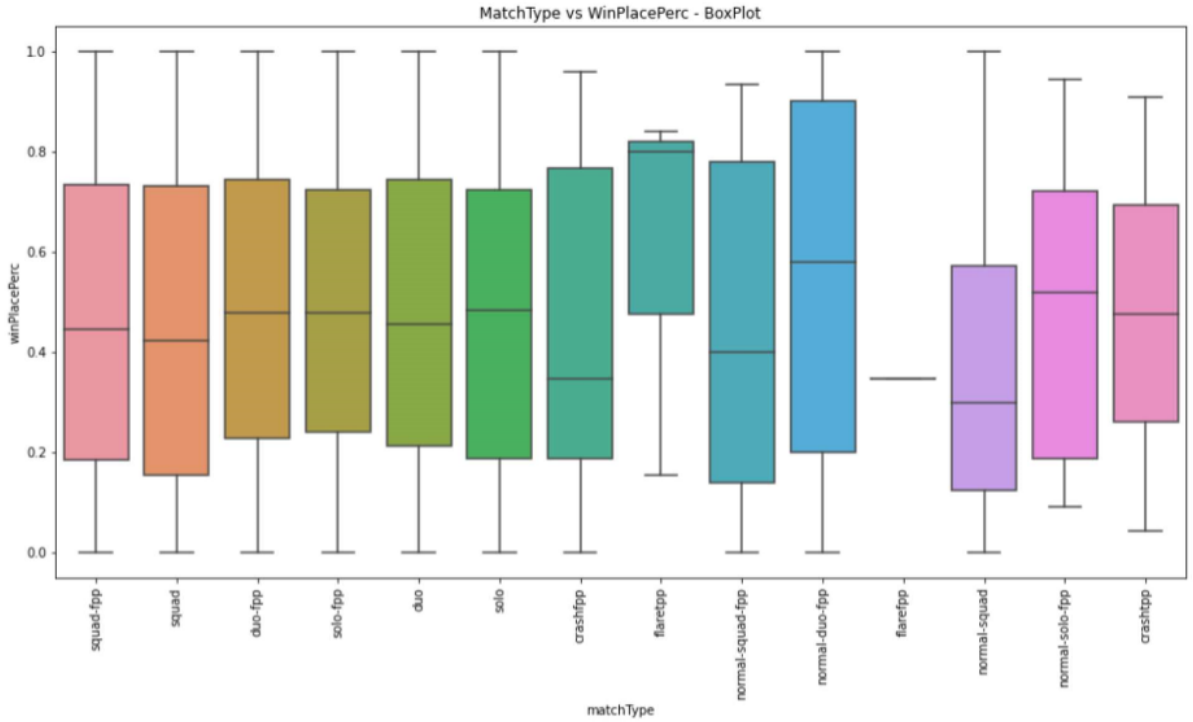
Out[40]:

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

10000 rows × 4 columns

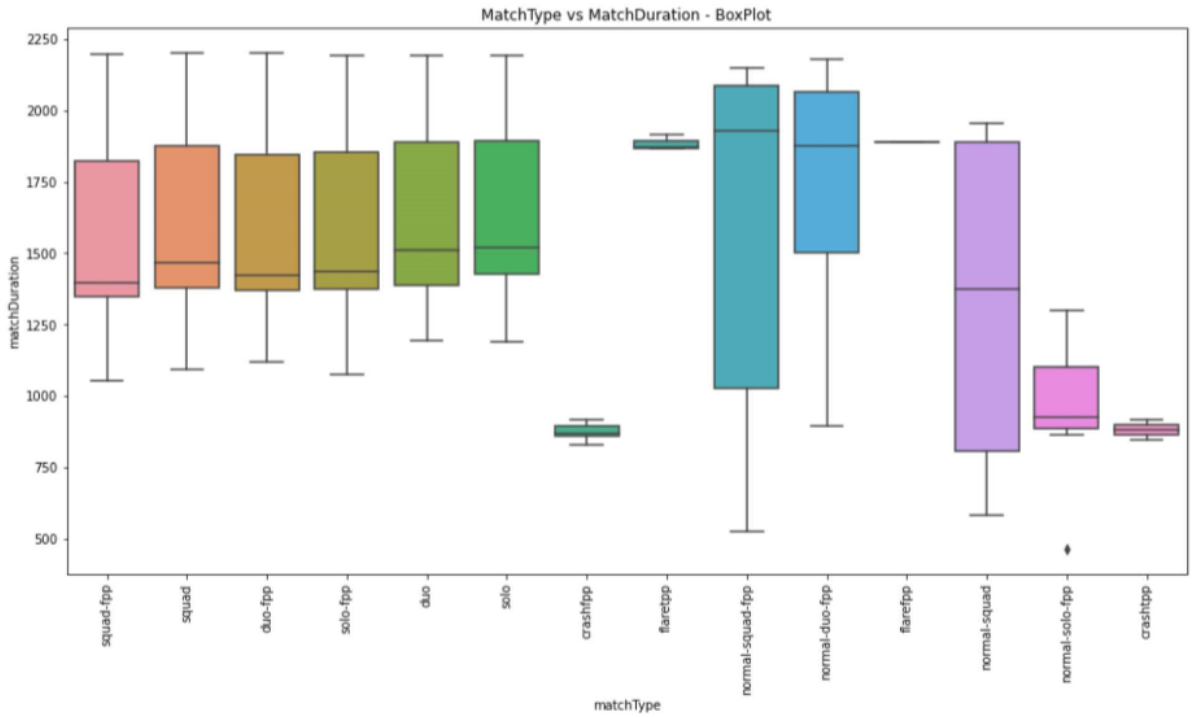
17. Plot a boxplot of ‘matchType’ vs ‘winPlacePerc’. Write your inferences

In [41]:
```
sns.boxplot(x='matchType',y='winPlacePerc',data=pubg).set_title('MatchType vs Wir
plt.xticks(rotation=90);
```



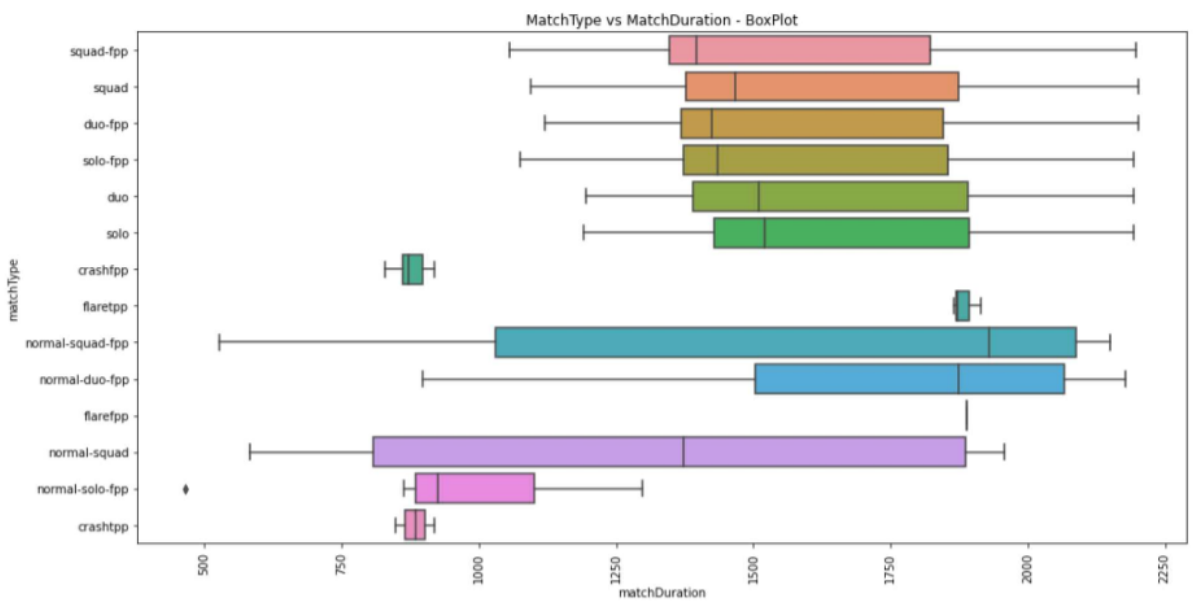
18. Plot a boxplot of ‘matchType’ vs ‘matchDuration’. Write your inferences

In [42]:
```
sns.boxplot(x='matchType',y='matchDuration',data=pubg).set_title('MatchType vs Ma
plt.xticks(rotation=90);
```



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

In [43]:
```
sns.boxplot(y='matchType',x='matchDuration',data=pubg).set_title('MatchType vs Ma
plt.xticks(rotation=90);
```



20. Add a new column called 'KILL' which contains the sum of following columns viz. headshotKills, teamKills,

roadKills.

In [44]:
```
pubg['KILL'] = pubg['headshotKills'] + pubg['teamKills'] + pubg['roadKills']
pubg.head()
```

Out[44]:

	Id	groupId	matchId	assists	boosts	damageDealt	DBNOs	head
0	2f262dd9795e60	78437bcd91d40e	d5db3a49eb2955	0	0	0.0	0	
1	a32847cf5bf34b	85b7ce5a12e10b	65223f05c7fdb4	0	0	163.2	1	
2	1b1900a9990396	edf80d6523380a	1cadec4534f30a	0	3	278.7	2	
3	f589dd03b60bf2	804ab5e5585558	c4a5676dc91604	0	0	191.9	1	
4	c23c4cc5b78b35	b3e2cd169ed920	cd595700a01bfa	0	0	100.0	1	

21. Round off column 'winPlacePerc' to 2 decimals.

In [45]:
```
pubg['winPlacePerc'] = pubg['winPlacePerc'].round(decimals=2)
pubg.head()
```

Out[45]:

	Id	groupId	matchId	assists	boosts	damageDealt	DBNOs	head
0	2f262dd9795e60	78437bcd91d40e	d5db3a49eb2955	0	0	0.0	0	
1	a32847cf5bf34b	85b7ce5a12e10b	65223f05c7fdb4	0	0	163.2	1	
2	1b1900a9990396	edf80d6523380a	1cadec4534f30a	0	3	278.7	2	
3	f589dd03b60bf2	804ab5e5585558	c4a5676dc91604	0	0	191.9	1	
4	c23c4cc5b78b35	b3e2cd169ed920	cd595700a01bfa	0	0	100.0	1	

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.

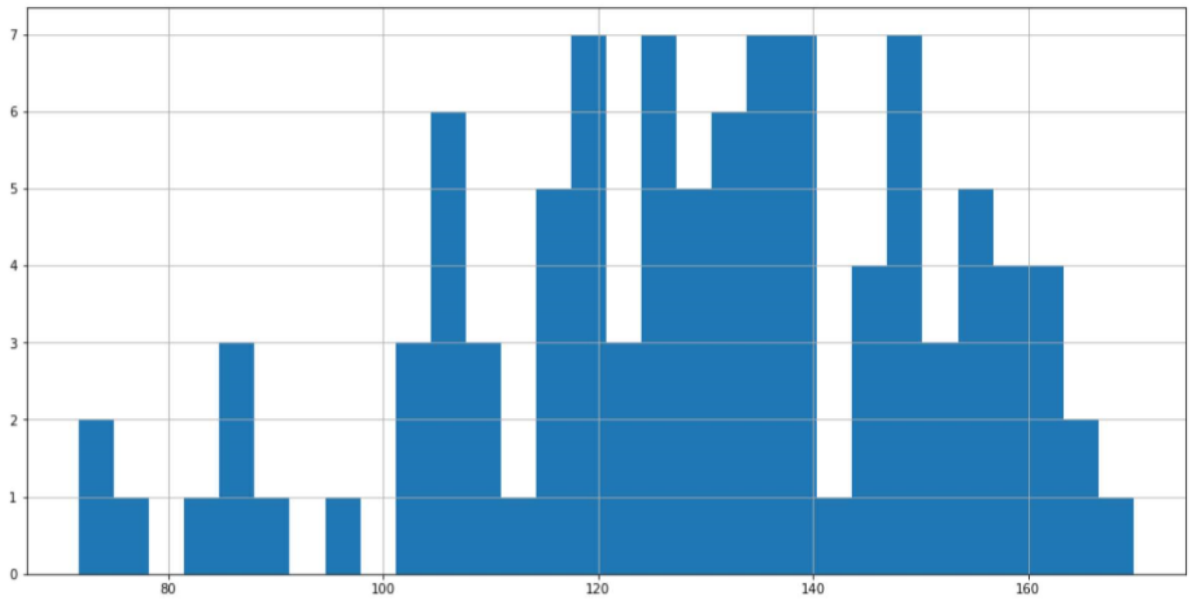
In [50]:
```
sam_100 = []
for i in range(100):
    sample = pubg['damageDealt'].sample(n=50)
    sam_100.append(sample.mean())
bootstrap_avg = sum(sam_100)/100
bootstrap_avg
```

Out[50]: 129.33769583999995

In [51]:
```
pubg['damageDealt'].mean()
```

Out[51]: 129.2112641000002

In [52]:
```
plt.hist(sam_100,bins=30);
plt.grid()
```



In [ ]: