1 Chess

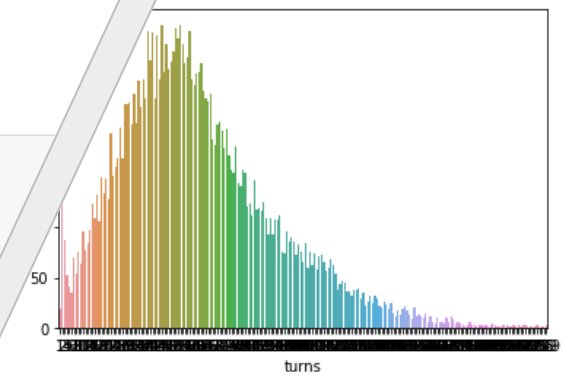
1.0.1 Bendra apžvalga

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

In [125]:
```
chess = pd.read_csv('C:\\Users\\Viktorija\\Desktop\\Duomenų analitika\\Atsiskaitomasis darbas\\chess_games.cs
chess.head(2)
```

Out[125]:

game_id	rated	turns	victory_status	winner	time_increment	white_id	white_rating	black_id	black_rating	moves	opening_code	or
1	False	13	Out of Time	White	15+2	bourgris	1500	a-00	1191	d4 d5 c4 c6 cxd5 e6 dxe6 fxe6 Nf3 Bb4+ Nc3 Ba5...		
2	True	16	Resign	Black	5+10	a-00	1322	skinnerua	1261	d4 Nc6 e4 e5 f4 f6 dxe5 fxe		

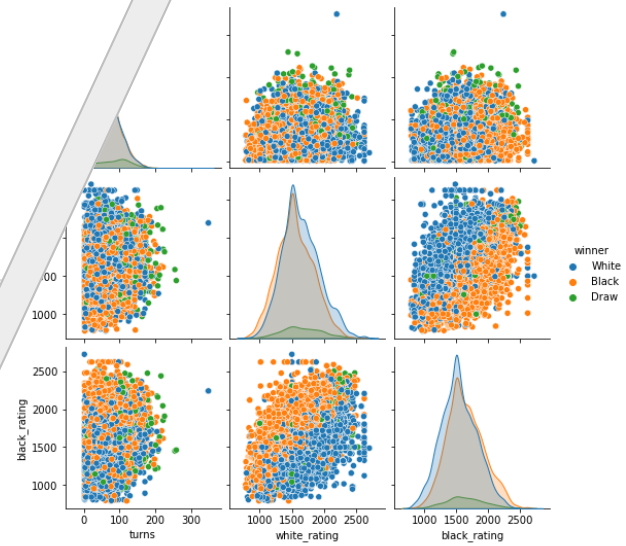



```
# ėjimų skaičius ir kiek kartų buvo tokių partijų
sns.boxplot(x='turns', data=time3, color='red', line
```

[125]: <AxesSubplot:xlabel='turns'>

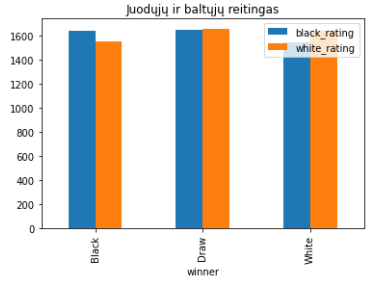
```
...s
..., atsižvelgiant į baltųjų ir juodųjų reitingus tuo metu, ir ėjimų pasiski
hue='winner')
```

Grid at 0x1d97793ebe0>

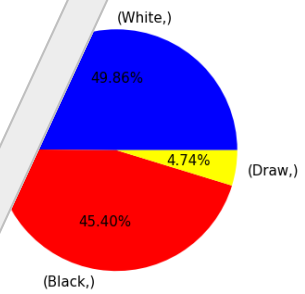


In [122]:
```
reitingas.plot(x='winner', y=['black_rating', 'white_rating'], kind='bar', title='Juodųjų ir baltųjų
```

Out[122]: <AxesSubplot:title={'center':'Juodųjų ir baltųjų reitingas'}, xlabel='winner'>




```
...nner']).value_counts()
...title='Laimėjimų pasiskirstymas', autopct='%.2f%%', colors = ['blue', 'red', 'yellow'], figsize=
...er':'Laimėjimų pasiskirstymas'}, ylabel='None'>
...asiskirstymas
```



1.0.1.1 Baltųjų reitingas, kai jie laimi prieš juoduosius yra žemesnis (1634,18), negu juodųjų r

1.0.1.2 Atitinkamai baltieji susiduria su mažesnį reitingą turinčiu žaidėju (1538,88), palygin reitingą turintį žaidėją (1549,25).

1.0.1.3 Juodieji turi turėti aukštesnį reitingą, tada tikimybė laimėti didesnė.

1.0.1.4 Lygiosiomis baigiasi, kai susiduria panašų reitingą turintys vartotojai.

1.0.2 Žaidėjai yra linkę rinktis ilgesnio formato partijas

In [ ]:
```
# populiariausia žaista forma(naudojamas žaidimui laikas).
```

In [124]:
```
print(chess['time_increment'].value_counts().idxmax(10))
```

```
# viso partijų - 20058, baltieji laimėjo 10001, juodieji - 9107, lygiosios 950
# atitinkamai pasiskirstė procentais: 49,86 %, 45,4% ir 4,74 %
```

```
# Daugiau partijų  laimėjo žaidę baltaisiais - 49,86 proc. palyginus su juodaisiais - 45,4
```

```
### Daugiausiai partijų baigiasi vienai ar kitai pusei paskelbus matą.
```

```
# pasiektų pergaliu pasiskirstymas. Skaičiuojant pergales, eliminuojama partijos, kur
```

In [84]:
```
winner_method = (pd.DataFrame(chess[['winner', 'victory_status']].value_counts())
                 .reset_index())
winner_method = winner_method[~(winner_method['winner'] == 'Draw')]
```

...opening variantai, kai laimėjo juodi ir kai laimėjo balti nesiskiria

```
...opening variantai, kokie buvo naudoti tarp visų
...DataFrame(chess['opening_fullname'].value_counts().head(20))
...me(columns={ 'index': 'opening_fullname', 'opening_fullname': 'naudota_kartu' }, inplace=True)
```

	naudota_kartu
Van't Kruijs Opening	368
Sicilian Defense	358
Sicilian Defense: Bowdler Attack	296
French Defense: Knight Variation	271
Scotch Game	271
...dinavian Defense: Mieses-Kotroc Variation	259
Queen's Pawn Game: Mason Attack	232
Queen's Pawn Game: Chigorin Variation	229
Scandinavian Defense	223
Horwitz Defense	209
Caro-Kann Defense	199
Philidor Defense #3	198
Philidor Defense #2	193
Indian Game	181
Italian Game: Anti-Fried Liver Defense	180
Four Knights Game: Italian Variation	176
Modern Defense	174