

# Šachy - jaký je nejlepší zahajovací tah?

Šimon Kurz

September 24, 2023

Pěšcem na E4 nebo D4 jsou dva zdaleka nejoblíbenější zahajovací tahy. Každý z nich má své příznivce, a na otázku, který z nich je lepší, neexistuje univerzální odpověď (alespoň pro účely her mezi lidmi, nevím, jak na tom jsou nejsilnější algoritmy). Můžeme se však pokusit vykoukat nějaké statistické zajímavosti.

Jako data pro náš výzkum jsem použil dataset z Kaggle: <https://www.kaggle.com/datasets/arevel/chess-games>. Původně obsahuje cca 6 milionů online her ze serveru Lichess.com - rating obou hráčů, výsledek hry, zahájení, atd... Já jsem si ho ořízl na cca 120 000 her. Provedeme na něm dva testy - průměrný rating hráčů obou zahajovacích tahů a úspěšnost obou tahů ve hrách podobně silných hráčů.

```
[66]: import matplotlib.pyplot as plt
import csv
import numpy as np
from scipy import stats
from statsmodels.graphics.gofplots import qqplot

def isE4(openingCode):
    """
    Vrátí True, pokud ECO kód zahájení předaný jako parametr odpovídá prvnímu
    → tahu E4.
    """
    letter = openingCode[0]
    if letter == 'B' or letter == 'C':
        return True
    return False

def isD4(openingCode):
    """
    Vrátí true, pokud ECO kód zahájení předaný jako parametr odpovídá prvnímu
    → tahu D4.
    """
    letter = openingCode[0]
    number = int(openingCode[1:])

    if letter == 'E' or letter == 'D':
        return True
    if letter == 'A' and number >= 40 and number <= 99:
```

```

        return True
    return False

def gameScore(gameResult):
    """
    Převede formát výsledku hry z našeho CSV do číselné hodnoty.
    """
    if gameResult == "1-0": return 1
    elif gameResult == "1/2-1/2": return 0.5
    elif gameResult == "0-1": return 0
    else: return 0.5

with open('chess_games_trimmed.csv','r') as csvfile:
    next(csvfile)

    ratings = []

    E4ratings = [] #rating bíleho ve hrach zahajenych E4
    D4ratings = [] #rating bíleho ve hrach zahajenych D4
    E4opponents = [] #rating černého ve hrach zahajenych E4
    D4opponents = [] #rating černého ve hrach zahajenych D4
    E4resultsList = [] #číslné výsledky her zahajenych E4
    D4resultsList = [] #číslné výsledky her zahajenych D4

    lines = csv.reader(csvfile, delimiter=',')
    for row in lines:

        gameResult = row[3]
        rating = int(row[6])
        opponentRating = int(row[7])
        opening = row[10]
        ratings.append(rating)

        if isE4(opening):
            E4ratings.append(rating)
            E4opponents.append(opponentRating)

            result = gameScore(gameResult)

            E4resultsList.append(result)
            E4score += result

        elif isD4(opening):
            D4ratings.append(rating)
            D4opponents.append(opponentRating)
            D4resultsList.append(gameScore(gameResult))

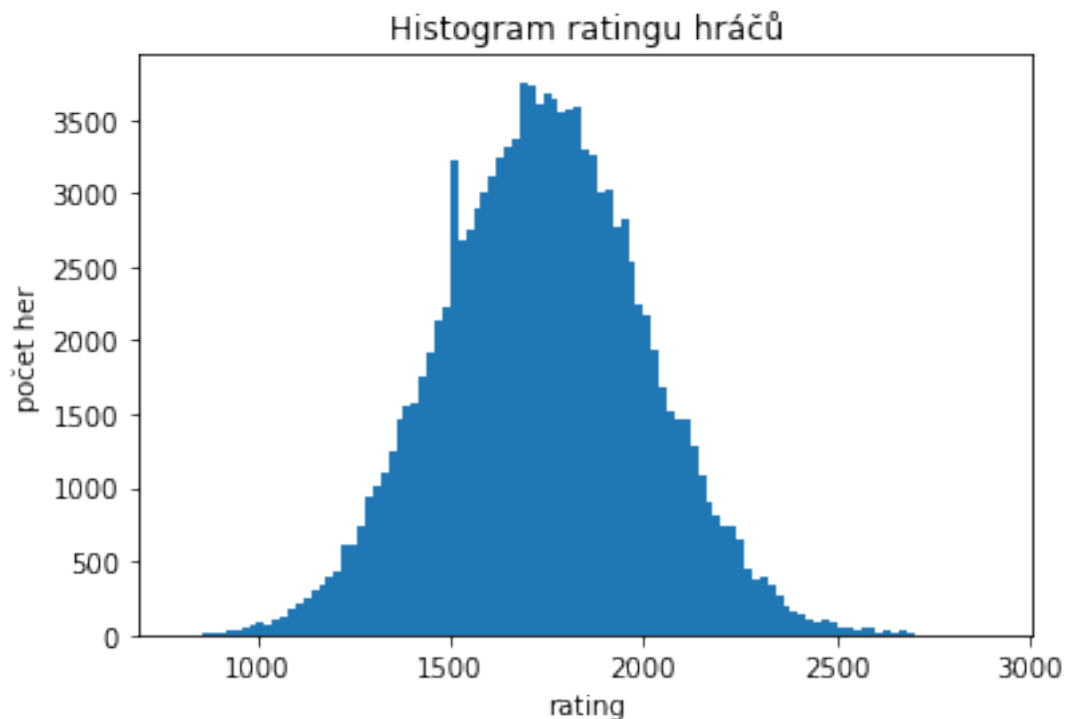
```

```

D4score += gameScore(gameResult)

plt.hist(ratings, bins=range(min(ratings),max(ratings), 20))
plt.title("Histogram ratingu hráčů")
plt.xlabel("rating")
plt.ylabel("počet her")
plt.show()

```



### 0.0.1 Test 1: Který tah hrají hráči s větším ratingem?

H0: Průměrný rating bílého hráče je pro oba zahajovací tahy stejný.

Pokud by však existoval rozdíl, očekávám, že pro D4 bude průměrný rating vyšší. E4 má pověst “divočejšího” tahu, ale zároveň také více přístupného pro začátečníky. Hry plynoucí z D4 jsou často pomalejší a více manévrovací.

Rozdělení ratingu hráčů by mělo být velmi dobře aproximováno normálním rozdělením. Při průměrování naší obrovské velikosti vzorku sice není předpoklad normality nutný, pro znázornění však můžeme vytvořit histogram a QQ-graf:

```

[67]: plt.hist(E4ratings, bins=range(min(ratings),max(ratings), 20))
plt.title("Histogram ratingu hráčů E4")
plt.xlabel("rating")
plt.ylabel("počet her")

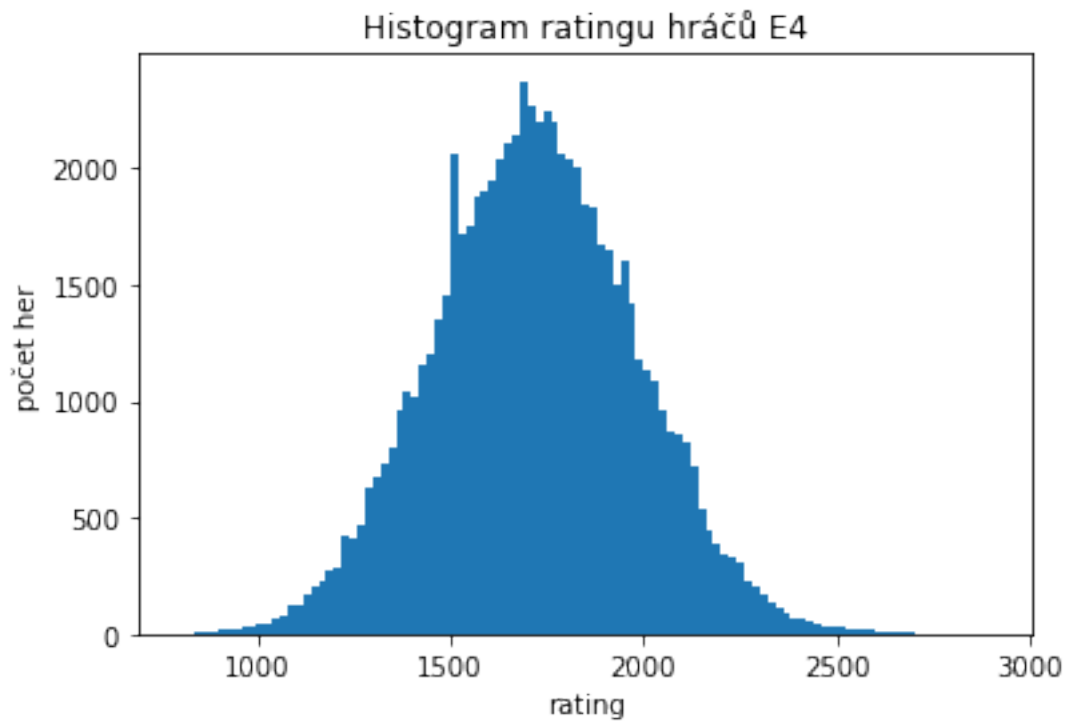
```

```
plt.show()

print(f"Pocet her: {len(E4ratings)}")
print(f"Prumerny rating: {sum(E4ratings)/len(E4ratings)}")

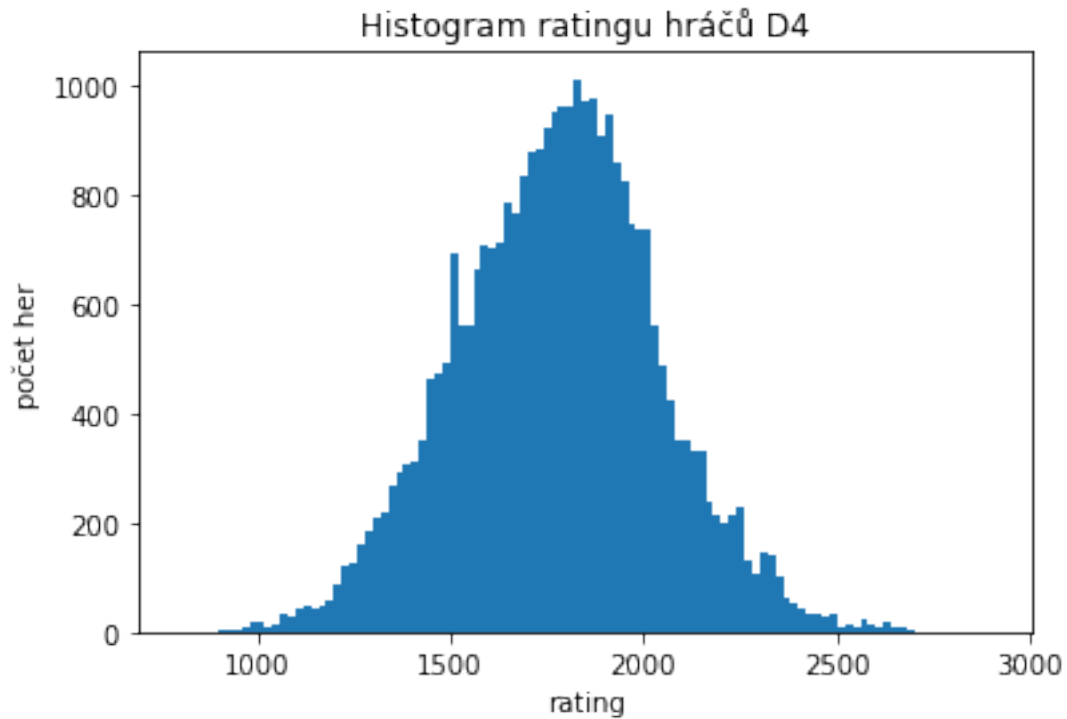
plt.hist(D4ratings, bins=range(min(ratings),max(ratings), 20))
plt.title("Histogram ratingu hracu D4")
plt.xlabel("rating")
plt.ylabel("pocet her")
plt.show()

print(f"Pocet her: {len(D4ratings)}")
print(f"Prumerny rating: {sum(D4ratings)/len(D4ratings)}")
```



Pocet her: 71894

Prumerny rating: 1718.1541853284002



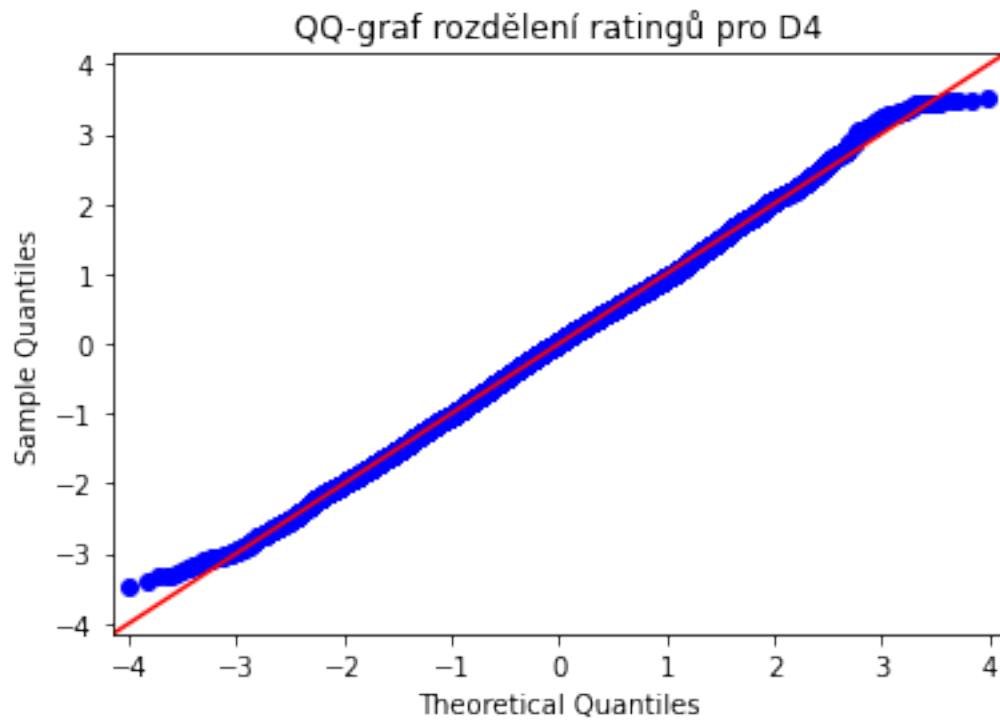
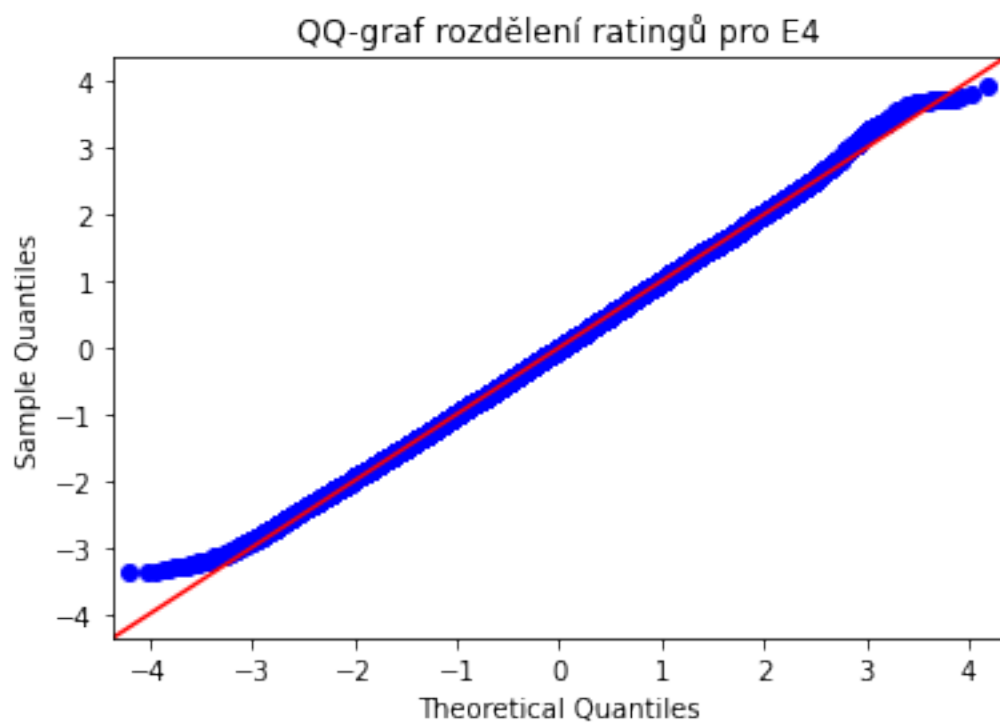
Počet her: 30623

Průměrný rating: 1776.1974333017668

Vidíme, že rozdíl v průměrném ratingu je poměrně velký. Při naší velké velikosti vzorku lze očekávat, že test nulovou hypotézu vyvrátí.

```
[68]: ratings = np.array(E4ratings)

#qq graf, cervena cara znazornuje normalni rozdeleni s parametry nafitovanymi na
→nase data
qqplot(ratings,fit=True, line="45")
plt.title("QQ-graf rozdělení ratingů pro E4")
plt.show()
ratings = np.array(D4ratings)
qqplot(ratings,fit=True, line="45")
plt.title("QQ-graf rozdělení ratingů pro D4")
plt.show()
```



Z QQ-grafu můžeme usoudit, že data odpovídají normálnímu rozdělení, až na okrajové hodnoty (podle očekávání), poměrně dobře.

Provedeme tedy konečně dvouvýběrový test, abychom potvrdili, zda je rozdíl v průměrném ratingu statisticky významný. Použijeme knihovní funkci `scipy.stats.ttest_ind()`.

```
[69]: E4 = np.array(E4ratings)
      D4 = np.array(D4ratings)
      stat, p_value = stats.ttest_ind(E4, D4, equal_var=False)

      print("test statistic:", stat)
      print("p-value:", p_value)
```

```
test statistic: -32.68943826804658
```

```
p-value: 2.9937361435291522e-232
```

Díky velké velikosti vzorku vyšla p-hodnota téměř nulová.

Výsledek je tedy jasný, nulovou hypotézu zamítneme a máme důvod se domnívat, že D4 si opravdu častěji volí silnější hráči.

### 0.0.2 Test 2: Jde pouze o věc vkusu, nebo je některý z tahů opravdu úspěšnější?

D4 je tedy volbou v průměru silnějších hráčů. Znamená to ale, že je lepší? Zkusíme otestovat, zda je některý z tahů lepší i relativně k síle hráče.

Vybereme hry, kde jsou oba hráči podobně silní, protože větší rozdíly v ratingu by mohly zkomplikovat předpokládané pravděpodobnosti výhry, a pokusíme se otestovat, zda je některý z tahů více úspěšný.

(Například ratingový systém ELO je na předpokládaných pravděpodobnostech založen, ale neumím si představit, jak by bylo možné toto uvažovat ve výpočtu. - a navíc - Lichess.com používá ratingový systém Glicko-2, který je ještě výrazně složitější.)

**Samotný test provedeme následovně:**  $H_0$ : Počty výher, remíz a proher jsou pro oba tahy stejně rozdělené.

- 1) Nalezneme počet výher, remíz a proher pro každý z tahů. Budeme uvažovat pouze hry, kde je malý rozdíl v ratingu. (řekněme 10)
- 2) Pro obě naměřené hodnoty provedeme test dobré shody s předpokládaným parametrem. Nejdříve předpokládáme, že obě pocházejí ze stejného rozdělení - nejpravděpodobnější parametr tedy budou hodnoty, kde jsou započítány oba tahy.
- 3) Vypočteme  $\chi^2$  a p-hodnotu pro oba tahy. Protože jsou obě veličiny nezávislé, jako výslednou p-hodnotu našeho testu použijeme jejich součin. Pokud bude nižší než  $\alpha = 0.05$ , nulovou hypotézu zamítneme.

```
[70]: #pocet her
      E4count = 0
      D4count = 0
```

```

#vyhry, remizy, prohry
E4results = [0,0,0]
D4results = [0,0,0]
allResults = [0,0,0]

maxdif = 10

for i in range(len(E4ratings)):
    if E4count == 5000 : break #při příliš velkém vzorku vychází p-hodnota u
    →chi-sq testů velmi malá

    #chceme pouze hry s max rozdílem maxdif
    difference = E4ratings[i] - E4opponents[i]
    if difference >= -maxdif and difference <= maxdif:
        E4count +=1
        result = E4resultsList[i]

        index = int((1-result)*2) #prevedeme ciselny vysledek hry (1, 0.5. 0) na
    →index (0, 1, 2)
        E4results[index] += 1
        allResults[index] += 1

for i in range(len(D4ratings)):
    if D4count == 5000 : break #při příliš velkém vzorku vychází p-hodnota u
    →chi-sq testů velmi malá

    #chceme pouze hry s max rozdílem maxdif
    difference = D4ratings[i] - D4opponents[i]
    if difference >= -maxdif and difference <= maxdif:
        D4count += 1
        result = D4resultsList[i]

        index = int((1-result)*2) #prevedeme ciselny vysledek hry (1, 0.5. 0) na
    →index (0, 1, 2)
        D4results[index] += 1
        allResults[index] += 1

```

Nalezli jsme výsledky her zahájených E4 a D4 s podobně silnými hráči. Použijeme předpokládaný parametr z hodnot pro oba tahy dohromady a provedeme  $\chi^2$  test dobré shody pro oba vzorky zvlášť.

Použijeme vzorec pro  $\chi^2$  statistiku ze skript:

$$\chi^2 = \sum_{i=1}^k \frac{(E_i - O_i)^2}{E_i}$$



$$E_i = n\theta_i^*$$

$$O_i = x_i$$

```
[71]: n1 = E4count
n2 = D4count
count = n1 + n2

p1 = allResults[0]/count
p2 = allResults[1]/count
p3 = allResults[2]/count

#vypocet chi-sq podle vzorce vyse
chi1 = ((n1*p1 - E4results[0])**2)/(n1*p1) + ((n1*p2 - E4results[1])**2)/(n1*p2)
    + ((n1*p3 - E4results[2])**2)/(n1*p3)
chi2 = ((n2*p1 - D4results[0])**2)/(n2*p1) + ((n2*p2 - D4results[1])**2)/(n2*p2)
    + ((n2*p3 - D4results[2])**2)/(n2*p3)

print("Předpokládaný parametr:")
print(f"theta1 = {p1}")
print(f"theta2 = {p2}")
print(f"theta3 = {p3}")

print("")
print("E4:")
print(f"Počet her: {E4count}")
print(f"výhry = {E4results[0]/E4count}")
print(f"remízy = {E4results[1]/E4count}")
print(f"prohry = {E4results[2]/E4count}")

print("")
print("D4:")
print(f"Počet her: {D4count}")
print(f"výhry = {D4results[0]/D4count}")
print(f"remízy = {D4results[1]/D4count}")
print(f"prohry = {D4results[2]/D4count}")

print("")
print(f"chi_square pro E4 = {chi1}")
print(f"chi_square pro D4 = {chi2}")

#p-hodnota pro chi-sq - pravdepodobnost hodnot vyssich nez namerena
pvalue1 = 1-stats.chi2.cdf(chi1,2)
pvalue2 = 1-stats.chi2.cdf(chi2,2)
```

```
print("")
print(f"p-hodnota pro E4 : {pvalue1}")
print(f"p-hodnota pro D4 : {pvalue2}")

print(f"Součin p-hodnot: {pvalue1*pvalue2}")
```

Předpokládaný parametr:

```
theta1 = 0.4940154123626824
theta2 = 0.042138055418921135
theta3 = 0.46384653221839645
```

E4:

```
Počet her: 4247
výhry = 0.49140569813986346
remízy = 0.04285377913821521
prohry = 0.46574052272192135
```

D4:

```
Počet her: 1852
výhry = 0.5
remízy = 0.04049676025917927
prohry = 0.45950323974082075
```

```
chi_square pro E4 = 0.14302425047816608
chi_square pro D4 = 0.3279827169442562
```

```
p-hodnota pro E4 : 0.9309849891108485
p-hodnota pro D4 : 0.8487493563397353
Součin p-hodnot: 0.7901729102697881
```

P-hodnota vyšla poměrně vysoká a je jasné, že nulovou hypotézu nemůžeme zamítnout. Závěr tedy je, že zatímco D4 si jako zahajovací tah volí silnější hráči, z hlediska procentuální šance na výhru není mezi tahy statisticky významný rozdíl.