

S9

March 14, 2023

1 35 Póker

```
[5]: from random import sample

def pókerhönd():
    stokkur = [(i,j) for i in range(1,5) for j in range(1,14)]
    return sample(stokkur, 5)

def með_tvær_tvennur(hönd):
    # Skilar sönnu ef hönd geymir tvær tvennur
    gildi = [spil[1] for spil in hönd]
    M = set(gildi) # búa til mengi
    if len(M) != 3:
        return False # það verða að vera 3 gildi
    for spil in M:
        talning = gildi.count(spil)
        if talning > 2: return False # og ekkert spil má koma 3svar
    return True # það eru tvær tvennur!

def hlutfall_með_2x2(n):
    talning = 0
    for i in range(n):
        hönd = pókerhönd()
        if með_tvær_tvennur(hönd):
            talning += 1
    return talning/n

hf = hlutfall_með_2x2(100000)
print(f'Hlutfall með tvær tvennur: {hf*100:.1f}%')
```

Hlutfall með tvær tvennur: 4.7%

1.1 1.

```
[6]: def ertvenna(hönd):
    # Skilar sönnu ef hönd geymir tvær tvennur
    gildi = [spil[1] for spil in hönd]
    M = set(gildi) # búa til mengi
```

```

    if len(M) == 4:
        for spil in M:
            talning = gildi.count(spil)
            if talning == 2:
                return True # og ekkert spil má koma 3svar
        return False
ertvenna(pókerhönd())

```

[6]: True

1.2 2.

```

[138]: def erþrenna(hönd):
        # Skilar sönnu ef hönd geymir þrennu
        gildi = [spil[1] for spil in hönd]
        M = set(gildi)
        if len(M) == 3:
            for spil in M:
                talning = gildi.count(spil)
                if talning == 3:
                    return True # og ekkert spil má koma 3svar
            return False
        erþrenna(pókerhönd())

```

[138]: False

1.3 3.

```

[49]: def erfullthus(hönd):
        gildi = [spil[1] for spil in hönd]
        M = set(gildi)
        if len(M) == 2:
            for spil in M:
                talning = [gildi.count(i) for i in M]
                if (2 in talning and 3 in talning):
                    return True
            return False
        erfullthus(pókerhönd())

```

[49]: False

1.4 4.

```

[48]: def erlitur(hönd):
        # Skilar sönnu ef hönd geymir þrennu
        gildi = [spil[0] for spil in hönd]
        M = set(gildi)

```

```

    if len(M) == 1:
        return True
    return False
erlitur(pókerhönd())

```

[48]: False

1.5 5.

```

[63]: def hlutfall(n):
    tvennur = 0
    þrennur = 0
    fullthus = 0
    litur = 0

    for i in range(n):
        if ertvenna(pókerhönd()):
            tvennur+=1
        if erþrenna(pókerhönd()):
            þrennur +=1
        if erfullthus(pókerhönd()):
            fullthus += 1
        if erlitur(pókerhönd()):
            litur+=1
    return tvennur, þrennur, fullthus, litur

first, second, third, fourth = hlutfall(1000000)

print(f'Hlutfall með tvennur: {first/10000:.3f}%')
print(f'Hlutfall með þrennur: {second/10000:.3f}%')
print(f'Hlutfall með fullt hús: {third/10000:.3f}%')
print(f'Hlutfall með sama lit: {fourth/10000:.3f}%')

```

Hlutfall með tvennur: 42.172%
Hlutfall með þrennur: 2.093%
Hlutfall með fullt hús: 0.141%
Hlutfall með sama lit: 0.190%

2 36. Stærð Jarðskjálfta

```

[13]: from math import log10, e
import pandas as pd
def earthquake(M0):
    M = 2/3*(log10(M0)-9.1)
    return M

```

```

print(earthquake(1e16), "\n")

stærð = {"USGS": 3.20E17, "GCMT": 3.29e17, "GFZ": 2.9e17}
print("Miðstöð   Stærð")
print("-"*16)

for miðstöð in stærð:
    vægisst = earthquake(stærð[miðstöð])
    print(f"{miðstöð:10} {vægisst:.2f}")

```

4.6

Miðstöð	Stærð
USGS	5.60
GCMT	5.61
GFZ	5.57

3 37. Valkvæðir Stikar

3.1 1.

```

[136]: from math import log, exp

def lnrót(x, a):
    return round(log(x)/log(a))
a = lnrót(100,10)
b = lnrót(243,3)
c = lnrót(128,2)

data = [[100,10,a], [243,3,b],[128,2,c]]
df = pd.DataFrame(data, columns=['x', 'a', "log_a(x)"])
df

```

```

[136]:      x   a  log_a(x)
0  100  10         2
1  243   3         5
2  128   2         7

```

3.2 2.

```

[137]: aba = [exp(x) for x in range(6)]
def nauk(listi):
    print("[", end="")
    for i,k in enumerate(listi):
        if i == len(listi)-1:

```

```
        print(f"{k:.3f}", end="")
    else:
        print(f"{k:.3f}", end=", ")
    print("]")

nauk(aba)
```

[1.000, 2.718, 7.389, 20.086, 54.598, 148.413]