HW_Lab1

September 13, 2021

1 Problem 1

[ ]:

```
from scipy.stats import binom

a = binom(5, 0.5)
a_p = 1 - a.cdf(2)
print(f"a) p-value: {a_p}")

b = binom(10, 0.5)
b_p = 1 - b.cdf(5)
print(f"b) p-value: {b_p}")

c = binom(100, 0.5)
c_p = 1 - c.cdf(59)
print(f"c) p-value: {c_p}")

d = binom(100, 0.5)
d_p = 1 - d.cdf(39)
print(f"d) p-value: {d_p}")
```

```
a) p-value: 0.5
b) p-value: 0.3769531250000001
c) p-value: 0.028443966820490144
d) p-value: 0.9823998998911476
```

2 Problem 2

[ ]:

```
from scipy.special import comb


def p_Y_given_X(y, x):
    return comb(y, x)*2**(-y)/sum(comb(y_prime, x)*2**(-y_prime) for y_prime in␣
 ↪range(x, 6+1))
```

[ ]:

```
import numpy as np

for x in range(7):
    print(f"{x=}")
    y_given_x = [p_Y_given_X(y, x) for y in range(1, 7)]
    y_hat = np.argmax(y_given_x) + 1
    print(y_given_x)
    print(f"{y_hat=}\n")
```

```
x=0
[0.25196850393700787, 0.12598425196850394, 0.06299212598425197,
0.031496062992125984, 0.015748031496062992, 0.007874015748031496]
y_hat=1

x=1
[0.26666666666666666, 0.26666666666666666, 0.2, 0.13333333333333333,
0.08333333333333333, 0.05]
y_hat=1

x=2
[0.0, 0.16161616161616163, 0.24242424242424243, 0.24242424242424243,
0.20202020202020202, 0.15151515151515152]
y_hat=3

x=3
[0.0, 0.0, 0.125, 0.25, 0.3125, 0.3125]
y_hat=5

x=4
[0.0, 0.0, 0.0, 0.13793103448275862, 0.3448275862068966, 0.5172413793103449]
y_hat=6

x=5
[0.0, 0.0, 0.0, 0.0, 0.25, 0.75]
y_hat=6

x=6
[0.0, 0.0, 0.0, 0.0, 0.0, 1.0]
y_hat=6
```

3 Problem 4

```
[ ]: def p(x):
         return (1/6)*sum(comb(y, x)*2**(-y) for y in range(1, 7))


     E_x = sum(x*p(x) for x in range(7))
     print(E_x)
```

```
1.75
```

[ ]:

```python
E_x2 = sum(x**2 * p(x) for x in range(7))
sigma_x2 = E_x2 - E_x**2
print(Fraction(E_x2).limit_denominator())
print(Fraction(sigma_x2).limit_denominator())
```

```
14/3
77/48
```

[ ]:

```python
from fractions import Fraction

E_xy = (1/6)*sum(sum(comb(y, x)*2**(-y)*x*y for y in range(1, 7))
                 for x in range(0, 7))
Cov_xy = E_xy - E_x * (7/2)
print(Fraction(E_xy).limit_denominator())
print(Fraction(Cov_xy).limit_denominator())
```

```
91/12
35/24
```

[ ]:

```python
corr_coef = Cov_xy/(np.sqrt(sigma_x2)*np.sqrt(35/12))
print(corr_coef)
```

```
0.674199862463242
```

[ ]:

```python
a = corr_coef*np.sqrt(35/12)/np.sqrt(sigma_x2)
b = 7/2-a*E_x
print(Fraction(a).limit_denominator())
print(Fraction(b).limit_denominator())
```

```
10/11
21/11
```

[ ]:

```python
def f_opt(x):
    return a*x + b
```

[ ]:

```python
for x in range(7):
    print(f_opt(x))
    print(Fraction(f_opt(x)).limit_denominator())
```

```
1.909090909090909
21/11
2.8181818181818183
31/11
3.7272727272727275
41/11
4.636363636363637
51/11
5.545454545454546
61/11
6.454545454545455
```

```
71/11
7.363636363636364
81/11
```

4 Problem 5

[ ]:

```python
p_horses_winning = (1/2, 1/4, 1/8, 1/16, 1/64, 1/64, 1/64, 1/64)

def entropy(p_list):
    return sum(p*np.log2(1/p) for p in p_list)

entropy(p_horses_winning)
```

[ ]:

```
2.0
```

[ ]: