02 Probability Distribution

November 20, 2022

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
```

0.1 Binomial distribution

- Experiment with two possible outcomes

$$P(S_n = k) = \binom{n}{k} * p^k * (1-p)^{n-k}, \quad \text{for } 0 \leq k \leq n$$

- Expected value

$$E(X) = n * p$$

[2]:
```
p = 0.65
n = 3  # Number of repititions
k = 2  # Number of occurences of a certain outcome

model = stats.binom(n=n, p=p)
print(model.pmf(k))  # probability mass function calculates the probability that␣
↪out of n times, k times a certain value occures
print(model.mean())  # Expected Value
```

```
0.443625
1.9500000000000002
```

0.2 Uniform distribution

- Has a constant density in the interval $[a, b]$
- Used when there is a constant probability that a certain event happens
- Density function

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq x \leq b \\ 0 & \text{else} \end{cases}$$

- Distribution

$$F(x) = \begin{cases} 0 & \text{if } x < a \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ 1 & \text{if } x \geq b \end{cases}$$

- Expected value

$$E(X) = \frac{b+a}{2}$$

- Expected variance

$$Var(X) = \frac{(b-a)^2}{12}$$

[3]:
```
min_value = 0
width = 20

model = stats.uniform(min_value, width)

x = 7
# Something more than x
p = 1 - model.cdf(x)
print(p)

# Something less than x
p = model.cdf(x)
print(p)
```

```
0.65
0.35
```

0.3 Exponential distribution

- Model class for survival times

$$X \sim Exp(\lambda), \quad \lambda > 0$$

- Density function:

$$f(x) = \begin{cases} \lambda e^{-\lambda * x} & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases}$$

- Distribution function:

$$F(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 - e^{-\lambda * x} & \text{if } 0 \leq x \end{cases}$$

- Expected value:

$$E(X) = \frac{1}{\lambda}$$

- Expected variance:

$$Var(X) = \frac{1}{\lambda^2}$$

[4]:
```
expected_value = 4  # units
model = stats.expon(scale=expected_value)
model.cdf(1)  # Something happens within 1 unit
```

[4]: 0.22119921692859515

```
[5]: lambda_value = 0.25
expected_value = 1 / lambda_value
expected_value
```

```
[5]: 4.0
```

```
[6]: variance = 1 / (lambda_value ** 2)
variance
```

```
[6]: 16.0
```

```
[7]: # Given p and x
p = 0.63
x = 2
lmda = - np.log(1 - p) / x
expected_value = 1 / lmda
```

0.4 Normal distribution

- Density:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

- Distribution:

$$F(x) = \int -\inf^{\inf}$$

- Expected value:

$$E(X) = \mu$$

- Expected variance:

$$Var(X) = \sigma^2$$

```
[8]: mean = 1337
std = 42

model = stats.norm(mean, std)

p = model.cdf(1300)  # Probability less than 1300 - 0.1892
p = model.pdf(1300)  # Probability for 1300 - 0.006444
p = model.cdf(1300) + 1 - model.cdf(1400)  # Probability less than 1300 and more␣
↪than 1400 - 0.2560
p = 1 - model.cdf(1370)  # Probability more than 1370 - 0.2160
```

```
[ ]:
```