

5.1

a)

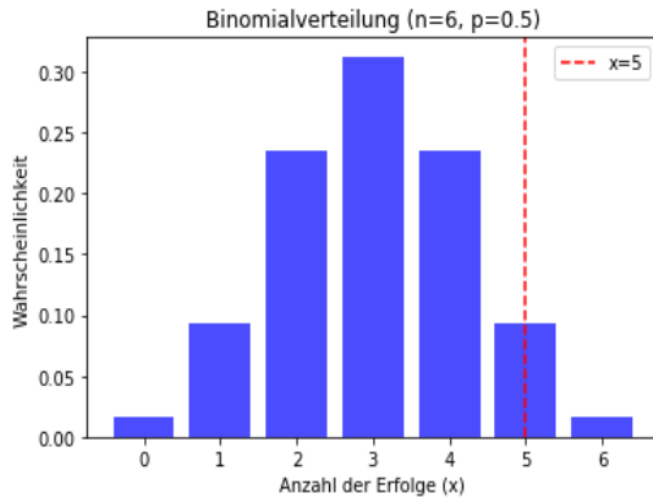
(1)

$n=6 \rightarrow$ Versuche

$p=0,5 \rightarrow$ Wahrscheinlichkeit Erfolg $p(x=5) = \binom{6}{5} * 0,5^5 * (1-0,5)^{6-5} = 0,09375 \rightarrow 9,375\%$

$x=5 \rightarrow$ Anzahl der Erfolge

(2)



(3)

„Schreibe mir ein Python Programm was mir eine Binomialverteilung visualisiert. schreibe das Beispiel mit $n=6$ $p=0,5$ und $x=5$ “

(4)

```
import matplotlib.pyplot as plt
from scipy.stats import binom

def plot_binomial_distribution(n, p, x):
    # Wahrscheinlichkeitsverteilung für die Binomialverteilung berechnen
    dist = binom(n, p)
    x_values = range(n+1)
    y_values = [dist.pmf(i) for i in x_values]

    # Diagramm erstellen
    plt.bar(x_values, y_values, color='blue', alpha=0.7)
    plt.title(f'Binomialverteilung (n={n}, p={p})')
    plt.xlabel('Anzahl der Erfolge (x)')
    plt.ylabel('Wahrscheinlichkeit')
    plt.xticks(x_values)

    # Markiere den spezifizierten Wert von x
    plt.axvline(x=x, color='red', linestyle='--', label=f'x={x}')
    plt.legend()

    # Diagramm anzeigen
    plt.show()

# Variablen
n = 6
p = 0.5
x = 5

plot_binomial_distribution(n, p, x)
```

(5)

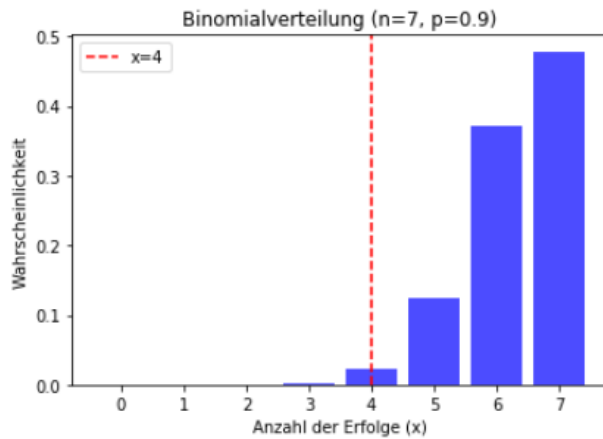
???

b)

(1)

$$p(x=4) = \binom{7}{4} * 0,9^4 * (1-0,9)^{7-4} \stackrel{TR}{=} 0,0229 \rightarrow 2,29\%$$

(2)



(3)

→ Prompt wurde nicht verändert nur Zahlen im Programm getauscht (siehe a.3)

(4)

```
import matplotlib.pyplot as plt
from scipy.stats import binom

def plot_binomial_distribution(n, p, x):
    # Wahrscheinlichkeitsverteilung für die Binomialverteilung berechnen
    dist = binom(n, p)
    x_values = range(n+1)
    y_values = [dist.pmf(i) for i in x_values]

    # Diagramm erstellen
    plt.bar(x_values, y_values, color='blue', alpha=0.7)
    plt.title(f'Binomialverteilung (n={n}, p={p})')
    plt.xlabel('Anzahl der Erfolge (x)')
    plt.ylabel('Wahrscheinlichkeit')
    plt.xticks(x_values)

    # Markiere den spezifizierten Wert von x
    plt.axvline(x=x, color='red', linestyle='--', label=f'x={x}')
    plt.legend()

    # Diagramm anzeigen
    plt.show()

# Variablen
n = 7
p = 0.9
x = 4

plot_binomial_distribution(n, p, x)
```

(5)

???

c)

(1)

$$n=9$$

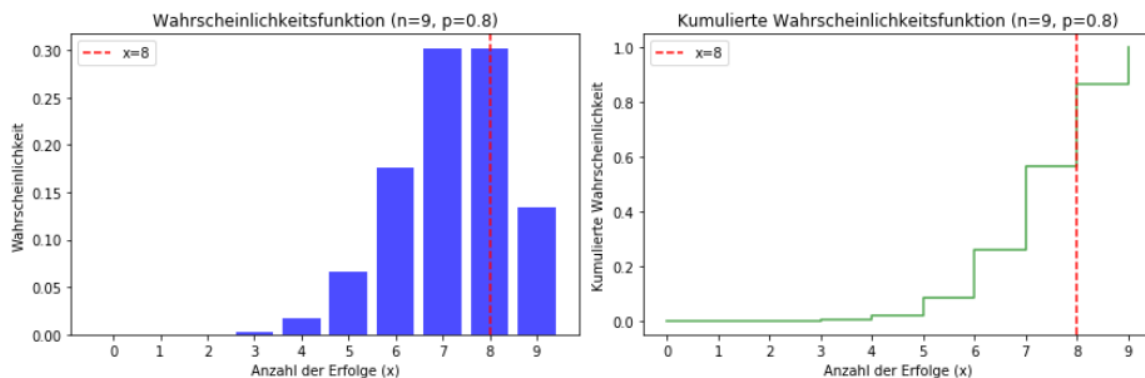
$$p=0,8$$

$$p(x=8) = \binom{9}{8} * 0,8^8 * (1-0,8)^1 \stackrel{TR}{=} 0,3019$$

$$p(x=9) = \binom{9}{9} * 0,8^9 * (1-0,8)^0 \stackrel{TR}{=} 0,0268$$

$$1 - (p(x=8) + p(x=9)) = 1 - (0,3019 + 0,0268) = 0,6713 \rightarrow 67,13\%$$

(2)



(3)

„erweitere das Programm das ich auch kombinierte Wahrscheinlichkeiten visualisieren kann“

(4)

```
import matplotlib.pyplot as plt
from scipy.stats import binom

def plot_binomial_distribution(n, p, x):
    # Wahrscheinlichkeitsverteilung für die Binomialverteilung berechnen
    dist = binom(n, p)
    x_values = range(n+1)
    pmf_values = [dist.pmf(i) for i in x_values]
    cdf_values = [dist.cdf(i) for i in x_values]

    # Diagramm für die Wahrscheinlichkeitsfunktion erstellen
    plt.figure(figsize=(12, 4))
    plt.subplot(1, 2, 1)
    plt.bar(x_values, pmf_values, color='blue', alpha=0.7)
    plt.title(f'Wahrscheinlichkeitsfunktion (n={n}, p={p})')
    plt.xlabel('Anzahl der Erfolge (x)')
    plt.ylabel('Wahrscheinlichkeit')
    plt.xticks(x_values)
    plt.axvline(x=x, color='red', linestyle='--', label=f'x={x}')
    plt.legend()

    # Diagramm für die kumulierte Wahrscheinlichkeitsfunktion erstellen
    plt.subplot(1, 2, 2)
    plt.step(x_values, cdf_values, where='post', color='green', alpha=0.7)
    plt.title(f'Kumulierte Wahrscheinlichkeitsfunktion (n={n}, p={p})')
    plt.xlabel('Anzahl der Erfolge (x)')
    plt.ylabel('Kumulierte Wahrscheinlichkeit')
    plt.xticks(x_values)
    plt.axvline(x=x, color='red', linestyle='--', label=f'x={x}')
    plt.legend()

    # Diagramme anzeigen
    plt.tight_layout()
    plt.show()

n = 9
p = 0.8
x = 8

plot_binomial_distribution(n, p, x)
```

(5)

???

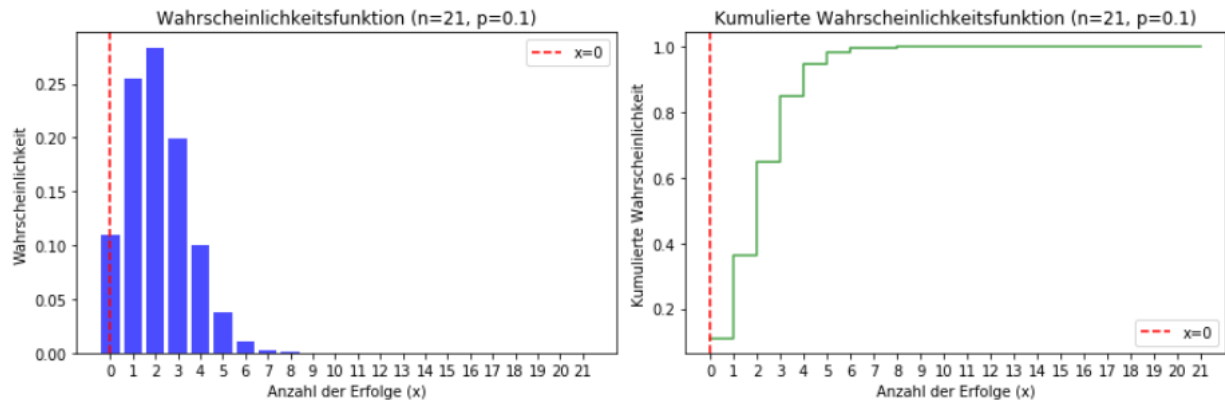
d)

(1)

$$n=21 \quad p=0,1 \quad p(x=0) = \binom{21}{0} * 0,1^0 * (1-0,1)^{21} \stackrel{TR}{=} 0,1094$$

$$1 - p(x=0) = 1 - 0,1094 = 0,8906 \rightarrow 89,06 \%$$

(2)



(3)

→ Prompt wurde nicht verändert nur Zahlen im Programm getauscht (siehe 5.1 c.3)

(4)

```
import matplotlib.pyplot as plt
from scipy.stats import binom

def plot_binomial_distribution(n, p, x):
    # Wahrscheinlichkeitsverteilung für die Binomialverteilung berechnen
    dist = binom(n, p)
    x_values = range(n+1)
    pmf_values = [dist.pmf(i) for i in x_values]
    cdf_values = [dist.cdf(i) for i in x_values]

    # Diagramm für die Wahrscheinlichkeitsfunktion erstellen
    plt.figure(figsize=(12, 4))
    plt.subplot(1, 2, 1)
    plt.bar(x_values, pmf_values, color='blue', alpha=0.7)
    plt.title(f'Wahrscheinlichkeitsfunktion (n={n}, p={p})')
    plt.xlabel('Anzahl der Erfolge (x)')
    plt.ylabel('Wahrscheinlichkeit')
    plt.xticks(x_values)
    plt.axvline(x=x, color='red', linestyle='--', label=f'x={x}')
    plt.legend()

    # Diagramm für die kumulierte Wahrscheinlichkeitsfunktion erstellen
    plt.subplot(1, 2, 2)
    plt.step(x_values, cdf_values, where='post', color='green', alpha=0.7)
    plt.title(f'Kumulierte Wahrscheinlichkeitsfunktion (n={n}, p={p})')
    plt.xlabel('Anzahl der Erfolge (x)')
    plt.ylabel('Kumulierte Wahrscheinlichkeit')
    plt.xticks(x_values)
    plt.axvline(x=x, color='red', linestyle='--', label=f'x={x}')
    plt.legend()

    # Diagramme anzeigen
    plt.tight_layout()
    plt.show()

n = 21
p = 0.1
x = 0

plot_binomial_distribution(n, p, x)
```

(5)

???

5.2

(1)

a)

$$n=16$$

$$p=0,25$$

$$p(x=1) = \binom{16}{1} * 0,25^1 * (1-0,25)^{16-1} \stackrel{TR}{=} 0,0534$$

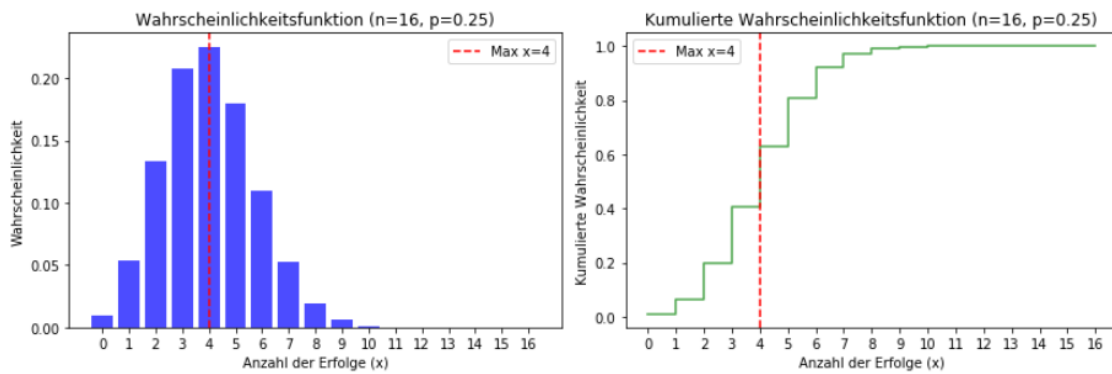
$$p(x=2) = \binom{16}{2} * 0,25^2 * (1-0,25)^{16-2} \stackrel{TR}{=} 0,1336$$

$$p(x=3) = \binom{16}{3} * 0,25^3 * (1-0,25)^{16-3} \stackrel{TR}{=} 0,2078$$

$$p(x=4) = \binom{16}{4} * 0,25^4 * (1-0,25)^{16-4} \stackrel{TR}{=} 0,2251$$

$$p(x=1) + p(x=2) + p(x=3) + p(x=4) = 0,0534 + 0,1336 + 0,2078 + 0,2251 = 0,6199 \rightarrow 61,99\%$$

(2)



(3)

→ Prompt wurde nicht verändert nur Zahlen im Programm getauscht (siehe 5.1 c.3)

(4)

```
import matplotlib.pyplot as plt
from scipy.stats import binom

def plot_binomial_distribution(n, p, x):
    # Wahrscheinlichkeitsverteilung für die Binomialverteilung berechnen
    dist = binom(n, p)
    x_values = range(n+1)
    pmf_values = [dist.pmf(i) for i in x_values]
    cdf_values = [dist.cdf(i) for i in x_values]

    # Diagramm für die Wahrscheinlichkeitsfunktion erstellen
    plt.figure(figsize=(12, 4))
    plt.subplot(1, 2, 1)
    plt.bar(x_values, pmf_values, color='blue', alpha=0.7)
    plt.title(f'Wahrscheinlichkeitsfunktion (n={n}, p={p})')
    plt.xlabel('Anzahl der Erfolge (x)')
    plt.ylabel('Wahrscheinlichkeit')
    plt.xticks(x_values)
    plt.axvline(x=x, color='red', linestyle='--', label=f'x={x}')
    plt.legend()

    # Diagramm für die kumulierte Wahrscheinlichkeitsfunktion erstellen
    plt.subplot(1, 2, 2)
    plt.step(x_values, cdf_values, where='post', color='green', alpha=0.7)
    plt.title(f'Kumulierte Wahrscheinlichkeitsfunktion (n={n}, p={p})')
    plt.xlabel('Anzahl der Erfolge (x)')
    plt.ylabel('Kumulierte Wahrscheinlichkeit')
    plt.xticks(x_values)
    plt.axvline(x=x, color='red', linestyle='--', label=f'x={x}')
    plt.legend()

    # Diagramme anzeigen
    plt.tight_layout()
    plt.show()

n = 16
p = 0.25
x = 4

plot_binomial_distribution(n, p, x)
```

(5)

???

b)

(1)

$$\rightarrow 1 - 0,6199 = 0,3801$$

$$1000 * 0,3801 = 380,1 \rightarrow 380$$

(2)

→ siehe 5.2 a.2

(3)

→ siehe 5.2 a.3

(4)

→ siehe 5.2 a.4

(5)

→ siehe 5.2 a.2

5.3

a)

(1)

$$p(x=12) = \binom{15}{12} * 0,5^{12} * (1-0,5)^{15-12} \stackrel{TR}{=} 0,0138$$

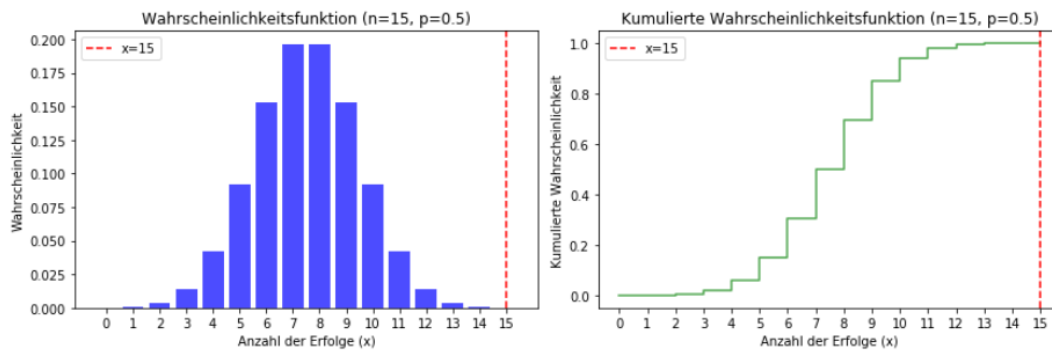
$$p(x=13) = \binom{15}{13} * 0,5^{13} * (1-0,5)^{15-13} \stackrel{TR}{=} 0,0032$$

$$p(x=14) = \binom{15}{14} * 0,5^{14} * (1-0,5)^{15-14} \stackrel{TR}{=} 0,0004$$

$$p(x=15) = \binom{15}{15} * 0,5^{15} * (1-0,5)^{15-15} \stackrel{TR}{=} 0,00003$$

$$p(x=12) + p(x=13) + p(x=14) + p(x=15) = 0,0174 \rightarrow 1,74\%$$

(2)



(3)

→ Prompt wurde nicht verändert nur Zahlen im Programm getauscht (siehe 5.1 c.3)

(4)

```

import matplotlib.pyplot as plt
from scipy.stats import binom

def plot_binomial_distribution(n, p, x):
    # Wahrscheinlichkeitsverteilung für die Binomialverteilung berechnen
    dist = binom(n, p)
    x_values = range(n+1)
    pmf_values = [dist.pmf(i) for i in x_values]
    cdf_values = [dist.cdf(i) for i in x_values]

    # Diagramm für die Wahrscheinlichkeitsfunktion erstellen
    plt.figure(figsize=(12, 4))
    plt.subplot(1, 2, 1)
    plt.bar(x_values, pmf_values, color='blue', alpha=0.7)
    plt.title(f'Wahrscheinlichkeitsfunktion (n={n}, p={p})')
    plt.xlabel('Anzahl der Erfolge (x)')
    plt.ylabel('Wahrscheinlichkeit')
    plt.xticks(x_values)
    plt.axvline(x=x, color='red', linestyle='--', label=f'x={x}')
    plt.legend()

    # Diagramm für die kumulierte Wahrscheinlichkeitsfunktion erstellen
    plt.subplot(1, 2, 2)
    plt.step(x_values, cdf_values, where='post', color='green', alpha=0.7)
    plt.title(f'Kumulierte Wahrscheinlichkeitsfunktion (n={n}, p={p})')
    plt.xlabel('Anzahl der Erfolge (x)')
    plt.ylabel('Kumulierte Wahrscheinlichkeit')
    plt.xticks(x_values)
    plt.axvline(x=x, color='red', linestyle='--', label=f'x={x}')
    plt.legend()

    # Diagramme anzeigen
    plt.tight_layout()
    plt.show()

n = 15
p = 0.5
x = 15

plot_binomial_distribution(n, p, x)

```

(5)

5.4

(1)

$$n=10$$

$$p=0,25$$

$$x \geq 3$$

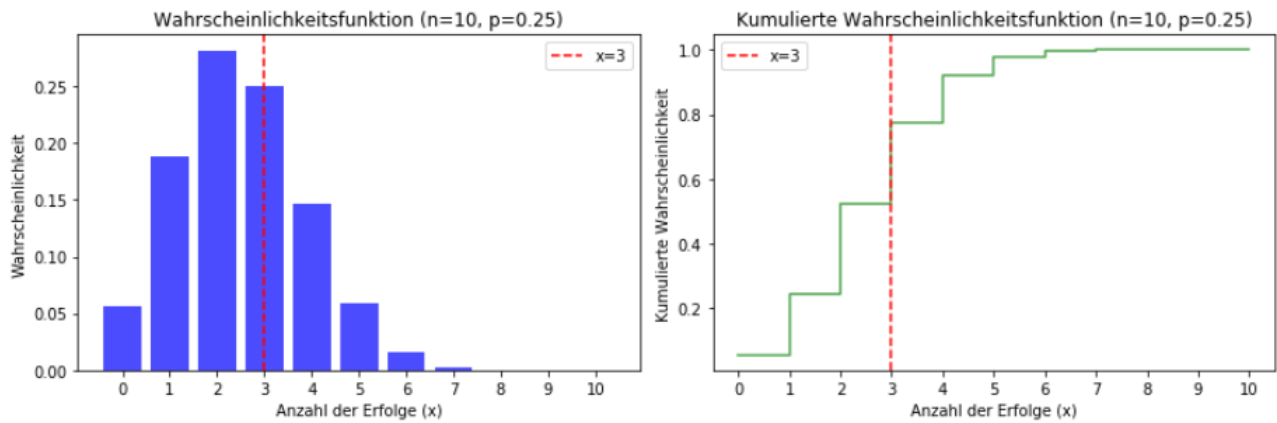
$$p(x=0) = \binom{10}{0} * 0,25^0 * (1-0,25)^{10-0} \stackrel{TR}{=} 0,0563$$

$$p(x=1) = \binom{10}{1} * 0,25^1 * (1-0,25)^{10-1} \stackrel{TR}{=} 0,1877$$

$$p(x=2) = \binom{10}{2} * 0,25^2 * (1-0,25)^{10-2} \stackrel{TR}{=} 0,2815$$

$$(1 - (p(x=0) + p(x=1) + p(x=3))) \stackrel{!}{=} 0,056 + 0,1877 + 0,2815 = 0,5255 \rightarrow 52,55\%$$

(2)



(3)

→ Prompt wurde nicht verändert nur Zahlen im Programm getauscht (siehe 5.1 c.3)

(4)

```
import matplotlib.pyplot as plt
from scipy.stats import binom

def plot_binomial_distribution(n, p, x):
    # Wahrscheinlichkeitsverteilung für die Binomialverteilung berechnen
    dist = binom(n, p)
    x_values = range(n+1)
    pmf_values = [dist.pmf(i) for i in x_values]
    cdf_values = [dist.cdf(i) for i in x_values]

    # Diagramm für die Wahrscheinlichkeitsfunktion erstellen
    plt.figure(figsize=(12, 4))
    plt.subplot(1, 2, 1)
    plt.bar(x_values, pmf_values, color='blue', alpha=0.7)
    plt.title(f'Wahrscheinlichkeitsfunktion (n={n}, p={p})')
    plt.xlabel('Anzahl der Erfolge (x)')
    plt.ylabel('Wahrscheinlichkeit')
    plt.xticks(x_values)
    plt.axvline(x=x, color='red', linestyle='--', label=f'x={x}')
    plt.legend()

    # Diagramm für die kumulierte Wahrscheinlichkeitsfunktion erstellen
    plt.subplot(1, 2, 2)
    plt.step(x_values, cdf_values, where='post', color='green', alpha=0.7)
    plt.title(f'Kumulierte Wahrscheinlichkeitsfunktion (n={n}, p={p})')
    plt.xlabel('Anzahl der Erfolge (x)')
    plt.ylabel('Kumulierte Wahrscheinlichkeit')
    plt.xticks(x_values)
    plt.axvline(x=x, color='red', linestyle='--', label=f'x={x}')
    plt.legend()

    # Diagramme anzeigen
    plt.tight_layout()
    plt.show()

n = 10
p = 0.25
x = 3

plot_binomial_distribution(n, p, x)
```

(5)