

# Mathematik 3: Übungsblatt - Wahrscheinlichkeitsverteilungen mit R

---

## Diskrete Wahrscheinlichkeitsverteilungen mit R

### 1. Aufgabe:

Wenn einer Wahrscheinlichkeitsverteilung eine bekannte mathematische Formel zugrunde liegt, kann man diese direkt in R übersetzen und damit Berechnungen rund um die Wahrscheinlichkeitsverteilung durchführen.

Nehmen wir als Beispiel die Wahrscheinlichkeitsfunktion der Binomialverteilung. R hat eingebaute Funktionen `factorial()`, `choose()` und `dbinom()` was die Berechnung von Binomialkoeffizienten erleichtert.

Bestimmen Sie damit die Wahrscheinlichkeit von 3 mal "Kopf", bei 10 Würfeln einer fairen Münze - also mit  $P(\{\text{Kopf}\}) = P(\{\text{Zahl}\}) = 0.5$ , d.h. es handelt sich um eine binomialverteilte Zufallsvariable  $X$ :

#### Lösung 1:

```
> x <- 3
> n <- 10
> p <- 0.5
> factorial(n)/(factorial(x) * factorial(n-x)) * p ^ x * (1-p) ^ (n-x)
[1] 0.1171875
```

Die Wahrscheinlichkeit liegt also bei knapp 12%.

#### Lösung 2:

Es geht aber einfacher, denn R hat auch eine eingebaute Funktion für den Binomialkoeffizienten, `choose()`:

```
> choose(n, x) * p ^ x * (1-p) ^ (n-x)
[1] 0.1171875
```

#### Lösung 3:

Und es geht sogar noch einfacher, denn R hat eine Funktion speziell für die Berechnung von binomialverteilten Wahrscheinlichkeiten, `dbinom()`:

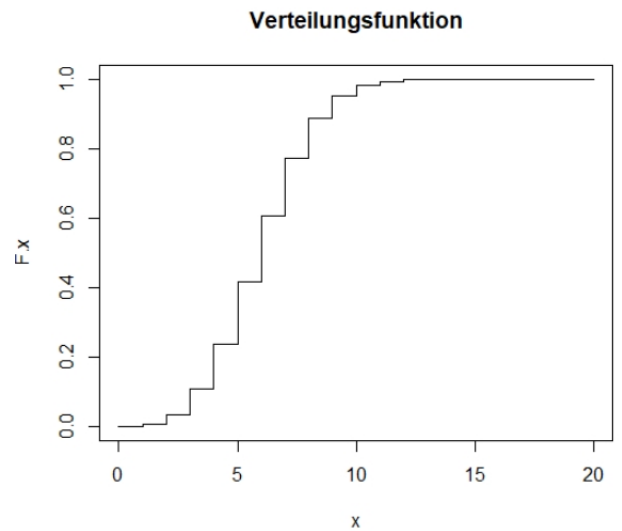
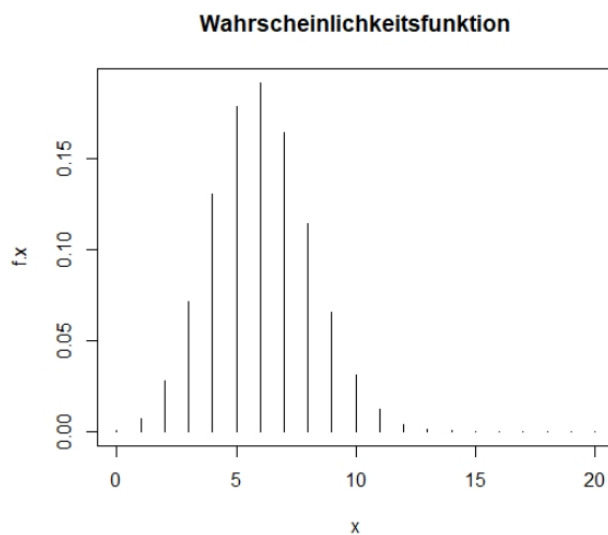
```
> dbinom(x, n, p)
[1] 0.1171875
```

## 2. Aufgabe:

Plotten Sie die Wahrscheinlichkeitsfunktion (Dichtefunktion) **und** Verteilungsfunktion (Stufendiagramm) der **Binomialverteilung** für  $n = 20$ ,  $p = 0.3$ )

### Lösung:

```
> #Wahrscheinlichkeitsfunktion der Binomialverteilung
> n=20
> p=0.3
> x=0:n
> x
[1] 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
> f.x=dbinom(x,n,p)
> f.x
[1] 7.979227e-04 6.839337e-03 2.784587e-02 7.160367e-02 1.304210e-01 1.788631e-01
[7] 1.916390e-01 1.642620e-01 1.143967e-01 6.536957e-02 3.081708e-02 1.200665e-02
[13] 3.859282e-03 1.017833e-03 2.181070e-04 3.738977e-05 5.007558e-06 5.049639e-07
[19] 3.606885e-08 1.627166e-09 3.486784e-11
> sum(f.x)
[1] 1
> plot(x,f.x)
> plot(x,f.x,type="h")
> plot(x,f.x,type="h",main="Wahrscheinlichkeitsfunktion")
> F.x=pbinom(x,n,p)
> F.x
[1] 0.0007979227 0.0076372598 0.0354831323 0.1070868045 0.2375077789 0.4163708294
[7] 0.6080098122 0.7722717974 0.8866685371 0.9520381027 0.9828551836 0.9948618385
[13] 0.9987211204 0.9997389530 0.9999570600 0.9999944497 0.9999994573 0.9999999623
[19] 0.9999999983 1.0000000000 1.0000000000
> plot(x,F.x)
> plot(x,F.x,type="s")
> plot(x,F.x,type="s",main="Verteilungsfunktion")
```



---

## Stetige Wahrscheinlichkeitsverteilungen und Umgang mit Zufallszahlen.

### 3. Aufgabe:

- a) Erzeugen Sie 100 standardnormalverteilte Zufallszahlen als Vektor  $x$ .

```
x=rnorm(100) oder  
x=rnorm(n=100, mean=0, sd=1)
```

- b) Bestimmen Sie die **Dichte** der Standardnormalverteilung in den Werten  $x_1 = -1, x_2 = 0, x_3 = 1$ :

```
dnorm(x, mean=0, sd=1)
```

```
dnorm(c(-1,0,1))
```

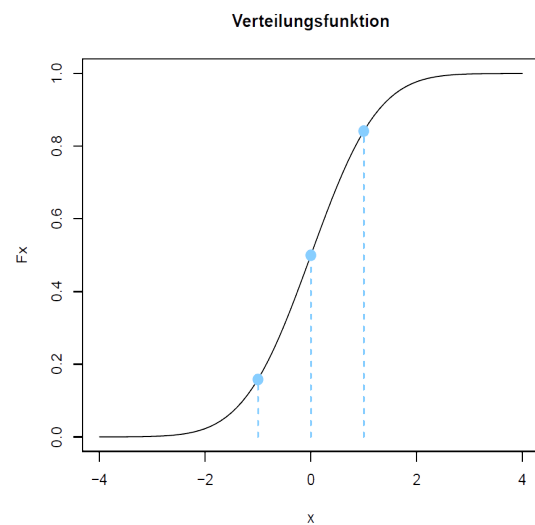
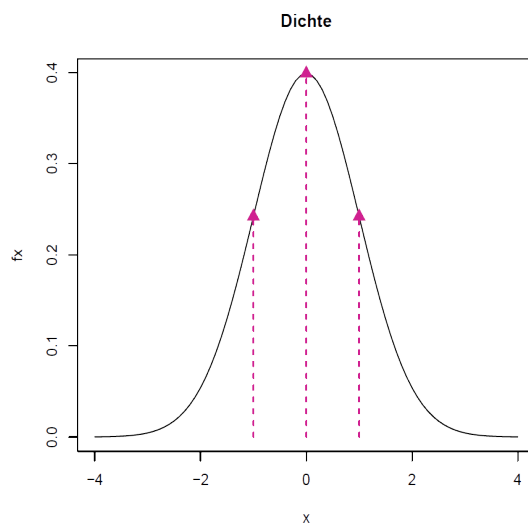
0.24197 0.39894 0.24197

- c) Bestimmen Sie die Werte der **Verteilungsfunktion** der Standardnormalverteilung für  $x_1 = -1, x_2 = 0, x_3 = 1$ :

```
pnorm(x, mean=0, sd=1)
```

```
pnorm(c(-1,0,1))
```

0.15866 0.50000 0.84134



- d) Bestimmen Sie die **Quartile** der Standardnormalverteilung:

$p_1 = 0.25, p_2 = 0.5, p_3 = 0.75$

```
qnorm(c(0.25,0.5,0.75))
```

-0.67449 0.00000 0.67449

#### 4. Aufgabe:

Plotten Sie für zufällig 100 normalverteilte Zufallszahlen das Histogramm, die Dichte und Verteilungsfunktion mit R.

#### Lösung 1:

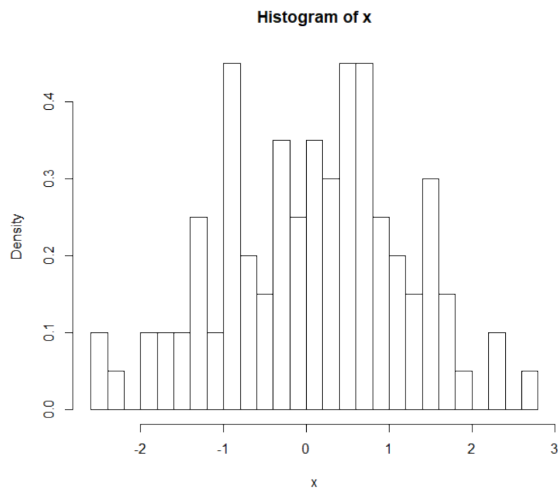
```
x <- rnorm(100)
```

Erstellung eines einfachen Histogrammes:

```
hist(x, breaks = "AnzahlBins", freq = NULL )
```

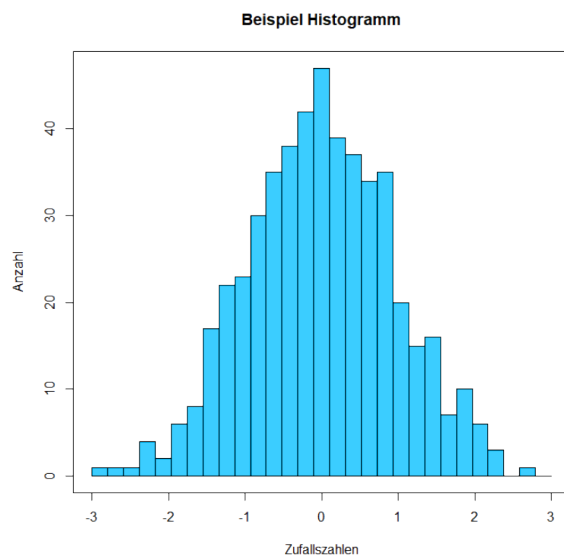
- `x`: Daten
- `breaks = "AnzahlBins"`: Steuerung der Teilintervalle
- `freq=TRUE`: absolute Häufigkeiten
- `freq=FALSE`: relative Häufigkeiten ("empirische Dichte")

```
hist(x,breaks=20,freq=FALSE)
```



Farbe und Anzahl der Intervalle für Histogramme in R

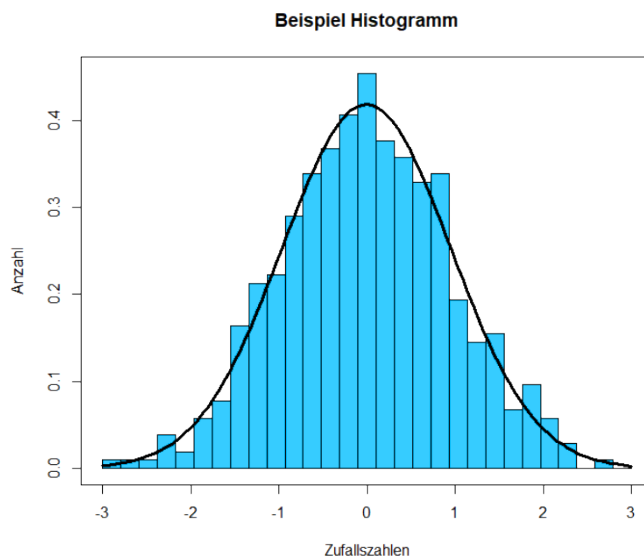
```
hist(x,main="Beispiel Histogramm",xlab="Zufallszahlen",ylab="Anzahl",  
col="deepskyblue",breaks=seq(-3,3,length=30))  
box()
```



Beachten Sie zunächst, dass bei Einzeichnung der Dichtefunktion die Option `freq=FALSE` in der Funktion `hist()` gewählt werden muss. Dadurch werden auf der y-Achse des Histogrammes nicht mehr die Häufigkeiten, sondern die Dichte abgebildet.

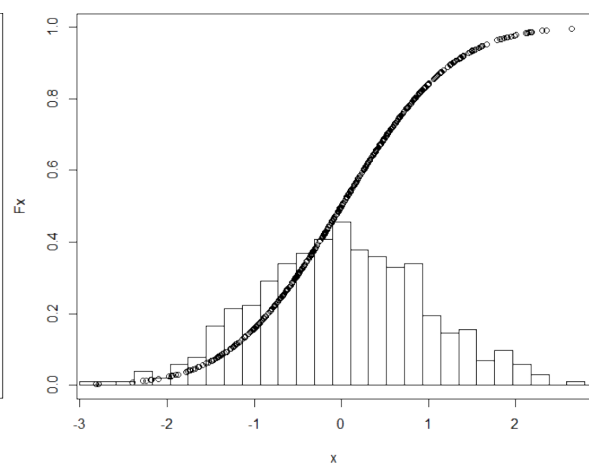
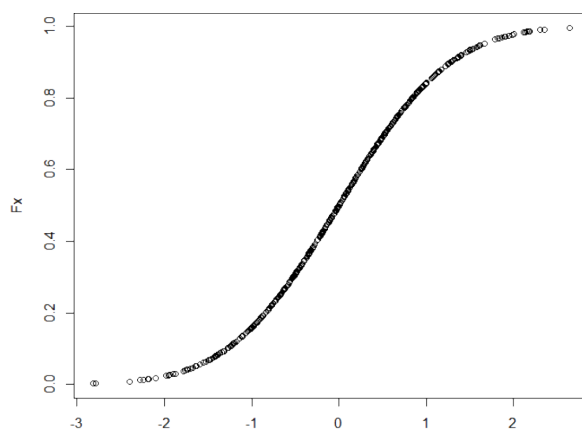
```
hist(x,main="Beispiel Histogramm",xlab="Zufallszahlen",ylab="Anzahl",
col="deepskyblue",breaks=seq(-3,3,length=30),freq=FALSE)
box()

m <- mean(x)
s <- sd(x)
curve(dnorm(x,m,s),add=TRUE,lwd=3)
box()
```



Dichtefunktion:

```
curve(pnorm(x, 0, 1), -3, 3, type="b")
hist(x,breaks=seq(-3,3,length=30),add=TRUE,freq=FALSE)
box()
```



---

### Lösung 2:

```
x=seq(from=-4, to=4, by=0.1)
```

```
fx=dnorm(x)
```

```
plot(x,fx,type="l")
```

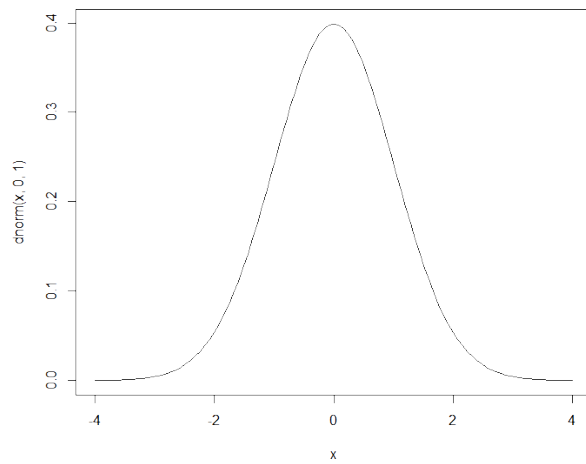
```
Fx=pnorm(x)
```

```
plot(x,Fx,type="l")
```

### Lösung 3:

Am einfachsten:

```
curve(dnorm(x, 0, 1), -4, 4)
```



```
curve(pnorm(x, 0, 1), -4, 4)
```

