1 Grundbegriffe

1.1 Artihmetisches Mittel

Das arithmetische Mittel beschreibt den Mittelwert der Summe aller Elemente.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

1.2 R-Tipps

1.2.1 Zahlenfolgen definieren

Vektor für eine lineare Zahlenfolge definieren mit Intervall = 1

```
> x <- 1:10
> x
```

```
 [1]  1  2  3  4  5  6  7  8  9 10
```

Vektor für eine Zahlenfolge mit beliebigem Intervall (z.B. 3)

```
> x <- seq(1,20,3)
> x
```

```
[1]  1  4  7 10 13 16 19
```

Eine spezielle Zahlenfolge kann auch manuell definiert werden

```
> x <- c(3,2,5,8,9,10,55,1,12)
> x
```

```
[1]  3  2  5  8  9 10 55  1 12
```

1.2.2 Artihmetisches Mittel mit R berechnen

Mit R kann das arithmetische Mittel mit der Funktion `mean()` ermittelt werden

```
> x <- c(2,5,1,7,8,9)
> mean(x)
```

```
[1] 5.333333
```

1.3 Standardabweichung

Die Standardabweichung beschreibt wie gross die mittlere Abweichung der Beobachtungen vom arithmetischen Mittel derselben Beobachtungen ist.

$$s_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Bsp.: Wir nehmen eine zufällige Zahlenfolge innerhalb $(1, 10)$ und rechnen das arithmetische Mittel als auch die Standardabweichung (`sd()`).

```
> x <- round(x=runif(n=10, min=1, max=10), digits=0)
> x
```

```
 [1] 8 3 6 3 7 6 9 5 8 3
```

```
> mean(x)

[1] 5.8

> sd(x)

[1] 2.250926
```

1.4 Quantile

Quantile beschreiben folgenden Zusammenhang: Hat man z.B. 20 Messungen gemacht und sortiert diese, dann beschreibt ein x%-iges Qantil eine Punkt oder Grenze in der Messreihe, wo x% der Werte darunter liegen.

$\alpha :$ Prozentwert $\quad \alpha \in [0,1]$

$x_1 - x_n$ sortiert nach grösse

$\alpha \cdot n$

hier müssen 2 Fälle unterschieden werden: ganze Zahlen und gebrochene

ganze Zahlen: $\frac{1}{2} \cdot (x_{\alpha \cdot n} + x_{\alpha \cdot (n+1)})$

gebrochene Zahlen: $k = \alpha \cdot n + \frac{1}{2}$
k runden
$\Rightarrow x_{(k)}$

```
> x<-round(x=runif(n=20, min=1, max=20), digits=0)
> x<-sort(x)
> x

 [1]  2  3  3  5  5  6  7  7  9 10 11 12 13 14 15 17 18 18 18 20

> quantile(x, prob=0.2)

20%
  5

> quantile(x, prob=0.2, type=1)

20%
  5

> quantile(x, prob=0.2, type=2)

20%
  5
```

Im obigen Beispiel wird mit R das 20%-Quantil bestimmt. Hier ist aber Vorsicht geboten, denn R hat 9 verschiedene `type` für die Funktion `quantile()` (default-Wert ist 7). Für uns aus dem Stochastik-Modul ist der `type=2` der einzig richtige Wert!

1.5 Median

Der Median ist ein Spezialfall der Quantile, nämlich ist dies jenes Quantil, welches die 50%-Marke beschreibt.

Bsp.: Wir haben 5 Personen, und messen deren Höhe. Danach sortieren wir die Ergebnisse. Der Median ist nun jene Person in der Mitte (unabhängig von seiner genauen Höhe!). Interssant oder eben speziell am Median ist, dass es immer die selbe Person bleibt auch wenn die kleineren und grösseren noch grösser und noch kleiner werden. Dies bedeutet, dass der Median unempfindlich gegenüber sog. Ausreissern ist (denke an Durchschnittsvermögen in einem Land mit vielen Armen und wenigen aber extrem Reichen).

```
> x <- c(1.6, 1.7, 1.75, 1.87, 1.94)
> median(x)

[1] 1.75

> x <- c(1.2, 1.4, 1.75, 1.99, 2.14)
> median(x)

[1] 1.75
```

1.6 Varianz

Die Varianz beschreibt die quadratische Abweichung von Daten von ihrem arithmetischen Mittelwert. Sie ist das Quadrat der Standardabweichung.

$$s_x{}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

```
> x=runif(n=10)
> y=runif(n=10)
> var(x)

[1] 0.1040337
```

1.7 Kovarianz

Die Kovarianz beschreibt, wie stark die Abweichungen von zwei Vektoren von ihren jeweiligen arithmetischen Mittelwerten korrelieren. Die Kovarianz eines Vektors mit sich selbst entspricht der Varianz des Vektors.

$$s_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

```
> cov(x,y)

[1] 0.03391453
```

1.8 Korrelationskoeffizient

Der Korrelationskoeffizient beschreibt die Linearität von zwei Vektoren zueinander. Der Wertebereich des Korrelationskoeffizienten ist $[-1, 1]$.

$$r = \frac{s_{xy}}{s_x \cdot s_y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \cdot \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

```
> cor(x,y)

[1] 0.4262696
```

2 Diskrete Verteilungen

2.1 Hypergeometrische Verteilung

Die Hypergeometrische Verteilung beschreibt die Wahrscheinlichkeit beim Ziehen ohne zurücklegen. Im folgenden Beispiel mit farbigen Kugeln.

$$X \sim Hyp(n, r, s)$$

n: Anzahl Ziehungen
r: Anzahl rote Kugeln in der Urne (positive Ergebnisse)
s: Anzahl schwarze Kugeln in der Urne (negative Ergebnisse)
N: Anzahl Kugeln in der Urne ($N = r + s$)

$$P(X = k) = \frac{\binom{r}{k} \cdot \binom{s}{n-k}}{\binom{r+s}{n}} = \frac{\binom{r}{k} \cdot \binom{N-r}{n-k}}{\binom{N}{n}}$$

```
> k=6;
> n=6;
> r=6;
> s=36;
> dhyper(x=k,m=r,n=s,k=n)
```

```
[1] 1.906292e-07
```

2.1.1 Kumulative Verteilungsfunktion

```
> q=1;
> phyper(q=q,m=r,n=s,k=n)
```

```
[1] 0.8025001
```

2.2 Binomialverteilung

Die Binomialverteilung kann bei Ereignissen eingesetzt werden, die zwei mögliche Ergebnisse zeigen können. Sie ist ein Grenzfall der Hypergeometrischen Verteilung. [1] Die Wahrscheinlichkeit ist dann wie folgt gegeben:

$$P(X = x) = \binom{n}{x} \cdot p^x \cdot (1-p)^{n-x} = \frac{n!}{x!(n-x)!} \cdot p^x \cdot (1-p)^{n-x}$$

n: Anzahl Versuche
x: Anzahl Versuche mit positivem Ergebnis
p: Wahrscheinlichkeit für ein positives Ergebnis jedem einzelnen Versuch

```
> n=5;
> x=3;
> p=0.2;
> dbinom(x=x,size=n,prob=p)
```

```
[1] 0.0512
```

[1] Beispiele für Ereignisse mit zwei möglichen Ergebnissen:

- Münzwurf $\Rightarrow$ Kopf $\leftrightarrow$ Zahl
- Würfeln $\Rightarrow$ Sechser $\leftrightarrow$ kein Sechser

2.2.1 Kumulative Verteilungsfunktion

```
> q=3;
> pbinom(q=q,size=n,prob=p)

[1] 0.99328
```

2.3 Poissonverteilung

Die Poissonverteilung wird bei Ereignissen verwendet, deren maximale Anzahl nicht begrenzt ist. Sie ist ein Grenzfall der Binomialverteilung.

$$P(X = x) = \exp(-\lambda) \cdot \frac{\lambda^x}{x!}$$

x: Anzahl Versuche mit positivem Ergebnis
λ: Erwartungswert

```
> lambda=2;
> x=3;
> dpois(x=x,lambda=lambda)

[1] 0.180447
```

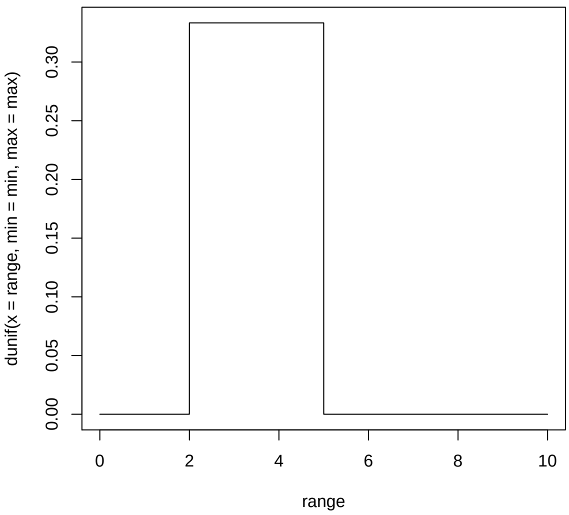
2.3.1 Kumulative Verteilungsfunktion

```
> q=3;
> ppois(q=q,lambda=lambda)

[1] 0.8571235
```

3 Stetige Verteilungen

3.1 Uniform

```
> x=3;
> min=2;
> max=5;
> dunif(x=x,min=min,max=max)

[1] 0.3333333

> range=seq(from=0,to=10,by=0.001);
> plot(range,dunif(x=range,min=min,max=max),type='l')
```



3.1.1 Kumulative Verteilungsfunktion

```
> q=3;
> punif(q=q,min=min,max=max)
```

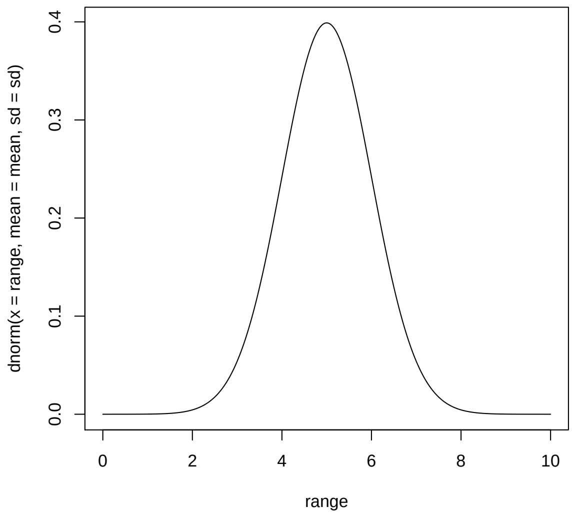
```
[1] 0.3333333
```

3.2 Normalverteilung

```
> x=7;
> mean=5;
> sd=1;
> dnorm(x=x,mean=mean,sd=sd)
```

```
[1] 0.05399097
```

```
> range=seq(from=0,to=10,by=0.001);
> plot(range,dnorm(x=range,mean=mean,sd=sd),type='l')
```

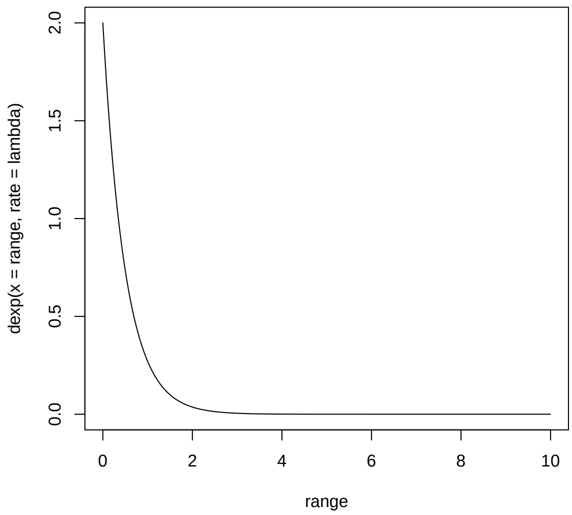


3.2.1 Kumulative Verteilungsfunktion

```
> q=7;
> pnorm(q=q,mean=mean,sd=sd)

[1] 0.9772499
```

3.3 Exponentialverteilung

```
> x=3;
> lambda=2;
> dexp(x=x,rate=lambda)

[1] 0.004957504

> range=seq(from=0,to=10,by=0.001);
> plot(range,dexp(x=range,rate=lambda),type='l')
```



3.3.1 Kumulative Verteilungsfunktion

```
> q=3;
> pexp(q=q,rate=lambda)

[1] 0.9975212
```

4 Statistischer Test

Beim statistischen Test wird überprüft, ob eine statistische Verteilung zu ermittelten Daten passt. Dieser Test besteht aus 6 Schritten.

4.1 allgemeiner Ablauf

1. Modell
   Verteilung bestimmen

2. Nullhypothese
   Nullhypothese und Alternativhypotese aufstellen

3. Teststatistik
   Teststatistik aus Modell und Nullhypothese erstellen

4. Signifikanzniveau
   Signifikanzniveau festlegen

5. Verwerfungsbereich
   Aus Teststatistik und Signifikanzniveau Verwerfungsbereich berechnen

6. Testentscheid
   Messwert mit Verwerfungsbereich vergleichen

4.2 Binomial-Test

Der Binomial-Test ist immer dann anzuwenden, wenn eine Binomialverteilung vorliegt. Der Binomial-Test kann ein- oder zweiseitig erfolgen, d.h. das Signifikanzniveau wird entweder von unten oder von oben angewendet (einseitig) oder es wird geteilt auf den unteren und oberen Bereich (zweiseitig). Hierbei wird nicht zwingend symmetrisch geteilt, sondern nur dann wenn die Verteilung symmetrisch ist.

4.3 z-Test

Der z-Test ist ein Test für eine Normalverteilung $\mathcal{N}(\mu, \sigma^2)$

1. Modell
   $X_1, ..., X_n$ i.i.d. $\sim \mathcal{N}(\mu, \sigma_x^2)$ wobei σ_x bekannt ist

2. Nullhypothese
   H_0: $\mu = \mu_0$
   Alternativhypothese
   H_A: $\mu \neq \mu_0 (\mu < \mu_0, \mu > \mu_0)$

3. Teststatistik
   $Z = \frac{\overline{X}_n - \mu_0}{\sigma_{\overline{X}_n}} = \frac{\sqrt{n} \cdot (\overline{X}_n - \mu_0)}{\sigma_x}$ $\overline{X}_n$ ist der Mittelwert der Beobachtungen `mean()`
   Verteilung der Teststatistik unter H_0
   $Z \sim \mathcal{N}(0,1)$ (sog. standardisierte Normalverteilung)

4. Signifikanzniveau
   $\alpha = \ldots$

5. Verwerfungsbereich
   Hier ist wichtig zu beachten, dass es drei mögliche Formulierungen gibt:

   - Die Alternative kann beidseitig liegen
     H_A: $\mu \neq \mu_0 \Rightarrow K = (-\infty, -\Phi^{-1}(1 - \frac{\alpha}{2})) \cup [\Phi^{-1}(1 - \frac{\alpha}{2}), \infty)$

   - Die Alternative wird linksseitig vermutet
     H_A: $\mu < \mu_0 \Rightarrow K = (-\infty, -\Phi^{-1}(1 - \alpha)]$

   - Die Alternative wird rechtsseitig vermutet
     H_A: $\mu > \mu_0 \Rightarrow K = [\underbrace{\Phi^{-1}(1 - \alpha)}_{\texttt{qnorm(1-}\alpha\texttt{,0,1)}}, \infty)$

6. Testentscheid
   Wir prüfen ob Z im Verwerfungsbereich K liegt. Falls ja, dann wird H_0 verworfen.

4.3.1 Beispiel mit R

In R ist der z-Test nicht im Standardumfang dabei, da es ein Spezialfall des t-Test ist und nur zu Schulungszwecken verwendet werden sollte. Man kann es aber manuell nachrüsten mit folgenden zwei Zeilen.

```
install.packages('TeachingDemos')
library('TeachingDemos')
```

Wir nehmen das Beispiel mit dem Weingeniesser, der prüfen möchte ob die Füllmengen richtig sind. Angeschrieben ist 70cl, desshalb ist unser $\mu = 70$. Die Standardabweichung ist 2. Beim z-Test kann man aber auf drei mögliche Alternativen prüfen (beidseitig und je Seite einseitig), diese sind in R: `less, greater, two.sided`. Im Fall des Weingeniessers müssen wir natürlich auf `less` prüfen. Wir möchten das auf einem Signifikanzniveau von $\alpha = 0.05$ testen.

```
> weinmenge <- c(71, 69, 67, 68, 73, 72, 71, 71, 68, 72, 69, 72)
> z.test(x=weinmenge, mu=70, sd=2, alternative='less')

        One Sample z-test

data:  weinmenge
z = 0.433, n = 12.000, Std. Dev. = 2.000, Std. Dev. of the sample mean
= 0.577, p-value = 0.6675
alternative hypothesis: true mean is less than 70
95 percent confidence interval:
     -Inf 71.19966
sample estimates:
mean of weinmenge
            70.25
```

Der P-Wert ist hier 0.6675, unser α ist aber 0.05, d.h. die Beobachtungen liegen nicht im Verwerfungsbereich!

4.4 t-Test

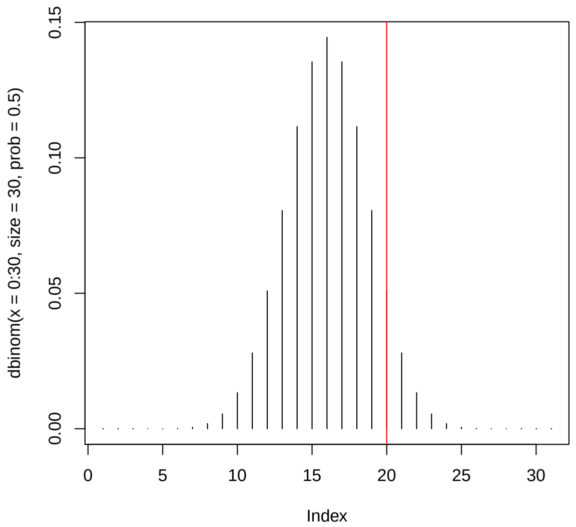
4.5 Wilcoxon Rangsummen Test

4.6 Vorzeichen-Test

5 P-Wert

Der P-Wert (engl. oder R *P-Value*) ist die Summe der Wahrscheinlichkeiten, welche die Beobachtung und alle extremeren Beobachtungen vereint. Im folgenden Beispiel wärde dies die Summe aller Wahrscheinlichkeiten rechts von der Beobachtung (80).

```
> plot(dbinom(x=0:30, size=30, prob=0.5), type='h')
> abline(v=20, col='red')
```



Der P-Value wird also genau gleich wie das α-Quantil berechnet. Für statistische Tests können wir mit Hilfe des P-Wertes das Berechnen des Verwerfungsbereiches auslassen indem wir den P-Wert und unser Signifikanzniveau vergleichen. Ist der P-Wert kleiner als das Signifikanzniveau, so liegt die Beobachtung im Verwerfungsbereich und umgekehrt.

6 Abkürzungsverzeichnis

i.i.d. Identically Identipednent Distributed

7 R-Glossar

Aufruf	**Name**	**Beschreibung**
`help(func)`	Hilfe	Hilfe zu Funktionen
`sum(x)`	Summe	Summe der Daten
`mean(x)`	Mittelwert	arithmetischer Mittelwert der Daten
`sd(x)`	Standardabweichung	Standardabweichung der Daten (σ)
`var(x)`	Varianz	Varianz der Daten (`var(x)` = `sd(x)^2` = `cov(x,x)`)
`median(x)`	Median	50 % Quantil
`quantile(x,prob,type=2)`	y % Quantil	Quantil der Daten **Achtung! type=2!**
`length(x)`	Länge	Länge des Datenvektors
`rep(x,times)`	Replicate	repliziert Elemente von Vektoren. Jedes `x` wird `times` mal aufgelistet
`cumsum(x)`	Kumulative Summe	Liefert einen Vektor, dessen Elemente die Summe aller vorhergehenden Summe des Eingangsvektors sind.
`scale(x)`	Standardisierung	Standardisiert Daten (mean=0, sd=1)
`unique(x)`	“Einzigartig“	entfernt doppelt vorhandene Werte
`apply(x)`	Anwenden	wendet eine Funktion auf jedes Element eines Vektors an.