

Einfache Verfahren zur Überprüfung von Referenzintervallen

Georg Hoffmann ¹, Frank Klawonn ²

2022-07-14

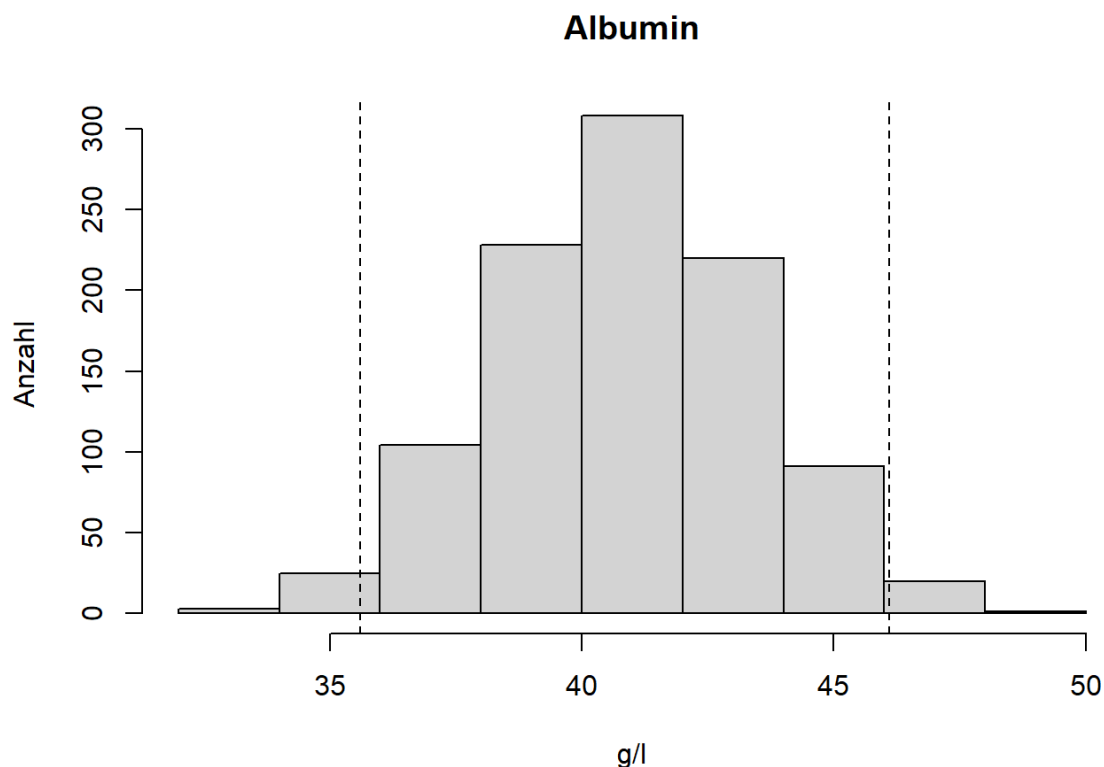
1. [Trillium GmbH Med. Fachverlag](#), Grafrath
2. [Helmholtzzentrum für Infektionsforschung \(HZI\)](#), Braunschweig

Inhalt

1. [Hintergrund und Aufgabenstellung](#)
2. [Material und Methodik](#)
3. [Datenaufbereitung und Bewertung nach Geschlecht und Alter](#)
4. [Prüfung der Referenzintervalle anhand von trunkierten zlog-Werten](#)
5. [Grafische Ausgabe des Ergebnisses](#)
6. [Ampelfarben als Interpretationshilfe](#)
7. [Schätzung absoluter Zielwerte](#)
8. [Diskussion und Ausblick](#)

Dieser Beitrag basiert auf einer [Publikation von Georg Hoffmann und Frank Klawonn](#) mit einem [Gastkommentar von Jakob Adler](#), erschienen in *Trillium Diagnostik 2022, Band 20, 102–105*. Wir liefern hier die Daten und Funktionen, die es Interessierten ermöglichen sollen, die dort beschriebenen Berechnungen praktisch nachzuvollziehen. Alle grau hinterlegten Programmbeispiele können direkt nach *R* bzw. *RStudio* kopiert und ausgeführt werden. So erzeugt man beispielsweise mit dem folgenden Testcode 1000 normalverteilte Albuminwerte mit einem Referenzintervall (abgekürzt RI) von etwa 35.6 bis 46.1 g/l.

```
hist(rnorm(n = 1000, mean = 40.85, sd = 2.68), main = "Albumin", xlab = "g/l", ylab = "Anzahl")  
  
abline(v = c(35.6, 46.1), lty = 2) #Referenzgrenzen einzeichnen
```



Für eine Einführung in die öffentlich verfügbare Programmiersprache R sei auf ein Skript der Trillium Akademie verwiesen, das ebenso wie der vorliegende Beitrag nach [kostenloser Registrierung](#) unter www.trillium.de/services/software.html zur Verfügung steht.

Kapitel 1

Hintergrund und Aufgabenstellung

Nach nationalen und internationalen Vorschriften sind medizinische Laboratorien verpflichtet, alle Referenzintervalle, die sie aus Produktblättern oder anderen externen Quellen übernommen haben, anhand eigener Messungen zu überprüfen ([Sack U & Özçürümez M 2019](#), [Hoffmann 2020](#)). Definitionsgemäß umfassen RI die zentralen 95 % von Laborwerten, die an gesunden Referenzpersonen gemessen wurden. Die leitliniengerechte Identifizierung und Rekrutierung solcher Personen ist allerdings in der Praxis aus Zeit- und Kostengründen sowie ethischen Bedenken so gut wie nicht realisierbar ([Ozarda Y et al. 2021](#)).

Deshalb werden heute vorwiegend indirekte Verfahren eingesetzt, die es erlauben, aus Routinewerten die „vermutlich normalen“ Werte zu identifizieren und deren nicht direkt ablesbare Referenzgrenzen über mathematische Parameter zu schätzen ([Jones G et al. 2018](#)). Dies ist organisatorisch deutlich einfacher als das direkte Verfahren, erfordert dafür aber statistischen Mehraufwand, der oft mit langen Rechenzeiten und hohen Anforderungen an die Fallzahlen verbunden ist.

In diesem Beitrag geht es um eine Routineaufgabe der Laboratoriumsdiagnostik, die hundert- bis tausendfach pro Labor anfällt, nämlich die Prüfung, ob vorgegebene Referenzgrenzen zu den lokalen Gegebenheiten hinsichtlich Analytik und Präanalytik passen oder nicht. Hierfür sollte nach Möglichkeit kein so großer Aufwand betrieben werden wie für die De-novo-Bestimmung von Referenzintervallen, um zeitliche und personelle Ressourcen zu schonen.

Wir stellen im Folgenden einfache Verfahren vor, die es erlauben, mit geringem mathematischem Aufwand und moderaten Fallzahlen zu überprüfen, welche Referenzintervalle im

Laborinformationssystem akzeptiert werden können und welche einer Korrektur bedürfen. Zur einfachen Beurteilung der Ergebnisse haben wir ein Ampelfarbenschema entwickelt, das auf einen Blick anzeigt, bei welchen Analyten der höchste Handlungsbedarf besteht.

Kapitel 2

Material und Methodik

Daten

Die verwendeten Echtdaten stammen aus einem öffentlich zugänglichen [Data Repository](#) für Projekte zum maschinellen Lernen ([Hoffmann G et al. 2018](#), [Oladimeji O 2021](#)). Sie können mit R unmittelbar eingelesen und analysiert werden. Mit folgendem Code werden die ersten Zeilen des Datensatzes dargestellt.

```
x <- read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/00571/hcvdat0.csv")

head(x) #zeigt die ersten Zeilen des Datensatzes an
```

##	X	Category	Age	Sex	ALB	ALP	ALT	AST	BIL	CHE	CHOL	CREA	GGT	PROT
## 1	1	0=Blood Donor	32	m	38.5	52.5	7.7	22.1	7.5	6.93	3.23	106	12.1	69.0
## 2	2	0=Blood Donor	32	m	38.5	70.3	18.0	24.7	3.9	11.17	4.80	74	15.6	76.5
## 3	3	0=Blood Donor	32	m	46.9	74.7	36.2	52.6	6.1	8.84	5.20	86	33.2	79.3
## 4	4	0=Blood Donor	32	m	43.2	52.0	30.6	22.6	18.9	7.33	4.74	80	33.8	75.7
## 5	5	0=Blood Donor	32	m	39.2	74.1	32.6	24.8	9.6	9.15	4.32	76	29.9	68.7
## 6	6	0=Blood Donor	32	m	41.6	43.3	18.5	19.7	12.3	9.92	6.05	111	91.0	74.0

Der Datensatz besteht aus 615 Zeilen mit quantitativen Laborwerten von 238 Frauen und 377 Männern im Alter von 19 bis 77 Jahren. Davon sind insgesamt 540 Personen Blutspender, die übrigen 75 weisen unterschiedliche Schweregrade einer Hepatitis C auf. Bei allen Personen wurden zehn Biomarker für die Diagnostik einer Leberschädigung bestimmt (ALB, ALP, ALT, AST, BIL, CHE, CHOL, CREA, GGT, PROT). Die Angaben für Geschlecht und Krankheitskategorie wandelt man mit folgendem Code in sog. "Faktoren" (nominale Variablen) um.

```
x$Sex <- as.factor(x$Sex)
x$Category <- as.factor(x$Category)

summary(x[, c(2, 4)])
```

##	Category	Sex
## 0=Blood Donor	:533	f:238
## 0s=suspect Blood Donor:	7	m:377
## 1=Hepatitis	: 24	
## 2=Fibrosis	: 21	
## 3=Cirrhosis	: 30	

Als zweites Input-File benötigen wir die Grenzwerte des Herstellers, die überprüft werden sollen. Beispielhaft erzeugen wir eine Matrix mit den Angaben für Albumin (ALB) in g/l und Alanin-Aminotransferase (ALT) in U/l.

```

ref <- matrix(c(35.6, 46.1, 35.6, 46.1, 10, 35, 10, 50), nrow = 4)
rownames(ref) <- c("Frauen UG", "Frauen OG", "Männer UG", "Männer OG")
colnames(ref) <- c("ALB", "ALT")
ref
##           ALB ALT
## Frauen UG 35.6  10
## Frauen OG 46.1  35
## Männer UG 35.6  10
## Männer OG 46.1  50

```

Die Abkürzungen UG und OG stehen hier für Untergrenze und Obergrenze.

Zusatzpakete

Für die Umsetzung der nachfolgend beschriebenen Algorithmen werden zwei zusätzliche R-Pakete benötigt:

- *truncnorm* von [O Mersmann et al. 2018](#) für trunkierte Normalverteilungen
- *zlog* von [S Gibb 2021](#) für die Berechnung von zlog-Werten.

Die Installation erfolgt mit den Funktionen *install.packages(truncnorm)* und *install.packages(zlog)*. Diese Packages muss man bei jedem Programmstart aktivieren.

```

library(truncnorm)
library(zlog)

```

Kapitel 3

Datenaufbereitung und Bewertung nach Geschlecht und Alter

Im ersten Schritt erzeugen wir den gewünschten Ausschnitt aus den Daten (ALB und ALT) und entfernen alle Zeilen mit fehlenden Werten (NA).

```

xx <- na.omit(x[, c(2 : 5, 7)]) #extrahiert die Spalten 2 bis 4 plus die Werte für
ALB und ALT

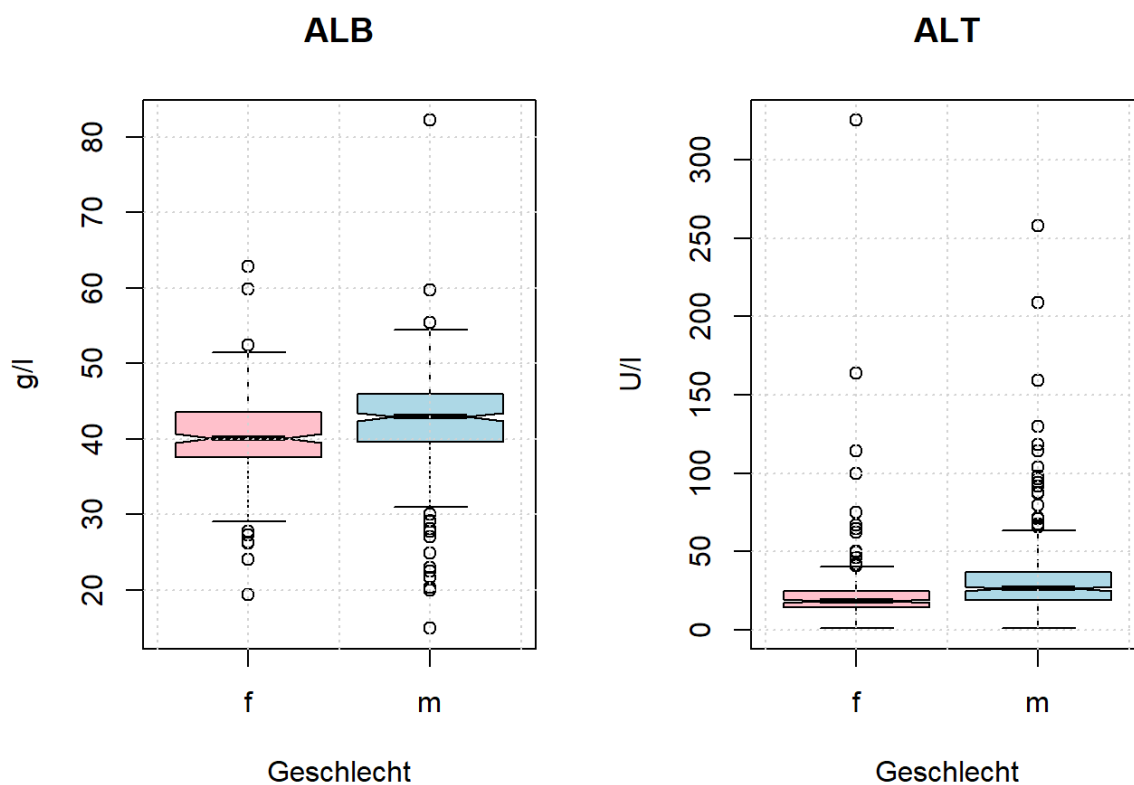
```

Zur groben Prüfung auf Geschlechtsunterschiede werden die Messwerte für Frauen und Männer getrennt als Boxplots angezeigt. Um die Bilder für ALB und ALT nebeneinander zu platzieren, verwenden wir die Funktion *par* (Parametereinstellungen für Grafiken) mit dem Argument *mfrow* (multiple figures arranged by rows).

```

par(mfrow = c(1, 2)) #Bildanordnung in 1 Zeile und 2 Spalten
boxplot(xx$ALB ~ xx$Sex,
        main = "ALB", xlab = "Geschlecht", ylab = "g/l",
        col = c("pink", "lightblue"), notch = TRUE)
grid()
boxplot(xx$ALT ~ xx$Sex,
        main = "ALT", xlab = "Geschlecht", ylab = "U/l",
        col = c("pink", "lightblue"), notch = TRUE)
grid()

```



Man kann aufgrund der Lage der Kästen (boxes) und Kerben (notches) auch ohne aufwendige Statistik bereits vorhersagen, dass sich die Grenzwerte für Frauen und Männer unterscheiden werden, obwohl der Hersteller beim Albumin nicht nach dem Geschlecht differenziert. Für die weitere Auswertung werden wir die Datensätze folglich nach dem Geschlecht trennen.

```

x.f <- subset(xx, xx$Sex == "f")
x.m <- subset(xx, xx$Sex == "m")

```

Zur Datenaufbereitung gehört auch stets eine Überprüfung der Altersabhängigkeit. Für ausgefüllte Punkte verwenden wir den Grafikparameter *pch* (plotting character) mit dem Wert 20.

```

par(mfrow = c(2, 2)) #Bildanordnung in 2 Zeilen und 2 Spalten)
plot(x.f$Age, x.f$ALB, main = "ALB (f)", xlab = "Alter (Jahre)", ylab = "g/l",
      ylim = c(min(xx$ALB), max(xx$ALB)), col = "red", pch = 20)
grid()

abline(h = ref[1 : 2, 1], lty = 2) #Grenzwerte ALB für Frauen als waagrechte gestrichelte Linien

plot(x.m$Age, x.m$ALB, main = "ALB (m)", xlab = "Alter (Jahre)", ylab = "g/l",
      ylim = c(min(xx$ALB), max(xx$ALB)), col = "blue", pch = 20)
grid()

abline(h = ref[3 : 4, 1], lty = 2) #Grenzwerte ALB für Männer

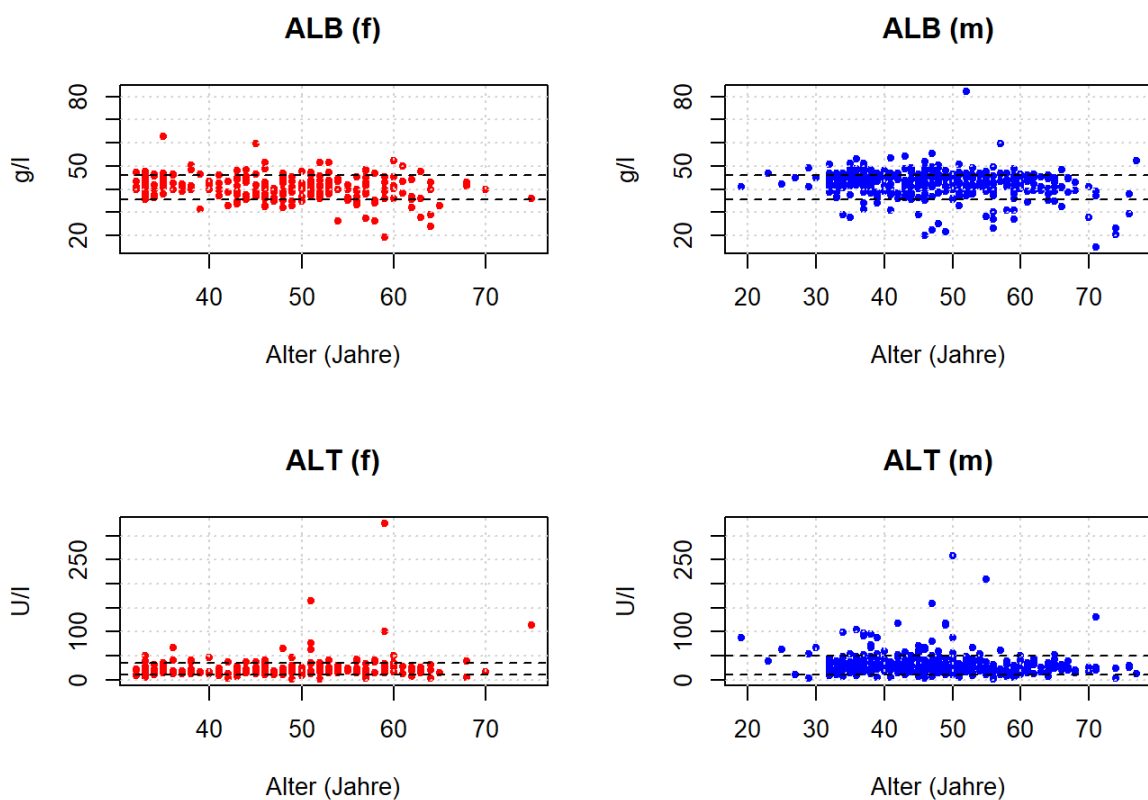
plot(x.f$Age, x.f$ALT, main = "ALT (f)", xlab = "Alter (Jahre)", ylab = "U/l",
      ylim = c(min(xx$ALT), max(xx$ALT)), col = "red", pch = 20)
grid()

abline(h = ref[1 : 2, 2], lty = 2) #Grenzwerte ALT für Frauen

plot(x.m$Age, x.m$ALT, main = "ALT (m)", xlab = "Alter (Jahre)", ylab = "U/l",
      ylim = c(min(xx$ALT), max(xx$ALT)), col = "blue", pch = 20)
grid()

abline(h = ref[3 : 4, 2], lty = 2) #Grenzwerte ALT für Männer

```



Auf den Bildern sind keine auffälligen Trends zu erkennen. Deshalb werden wir die Werte für die Referenzintervallprüfung ohne Alterspartitionierung verwenden. Ganz überschlägig erkennen wir an dieser grafischen Darstellung, dass die Hauptmenge der Punkte innerhalb der jeweiligen

Grenzwerte des Herstellers liegt. Für eine genaue Überprüfung der Referenzintervalle reicht diese grobe Darstellung allerdings nicht aus.

Kapitel 4

Prüfung der Referenzintervalle anhand von trunkierten zlog-Werten

Die Idee hinter dem nun ausführlich vorgestellten zweistufigen Algorithmus ist einfach:

- Durch die zlog-Normalisierung werden die Echtdaten so transformiert, dass ihr Referenzintervall unabhängig von Methode, Einheit etc. einheitlich von -1.96 bis +1.96 reicht (siehe [Hoffmann G et al. J Lab Med 2017](#)).
- Der klassische Boxplot wird so modifiziert, dass die Enden der Whiskers möglichst exakt die zentralen 95 % der (nicht-pathologischen) Hauptfraktion aller Werte markieren (siehe Klawonn F & Hoffmann G in [Advances in Intelligent Systems and Computing, Springer Nature 2022](#)). Diese entsprechen definitionsgemäß den zlog-Werten innerhalb des geschätzten Referenzintervalls.

Man muss dann nur noch prüfen, wie weit die so erhaltenen Referenzgrenzen von den erwarteten Grenzwerten -1.96 und +1.96 abweichen, um eine Aussage machen zu können, ob die vorgegebenen Grenzen zu den Echtdaten passen oder nicht.

Berechnung von zlog-Werten

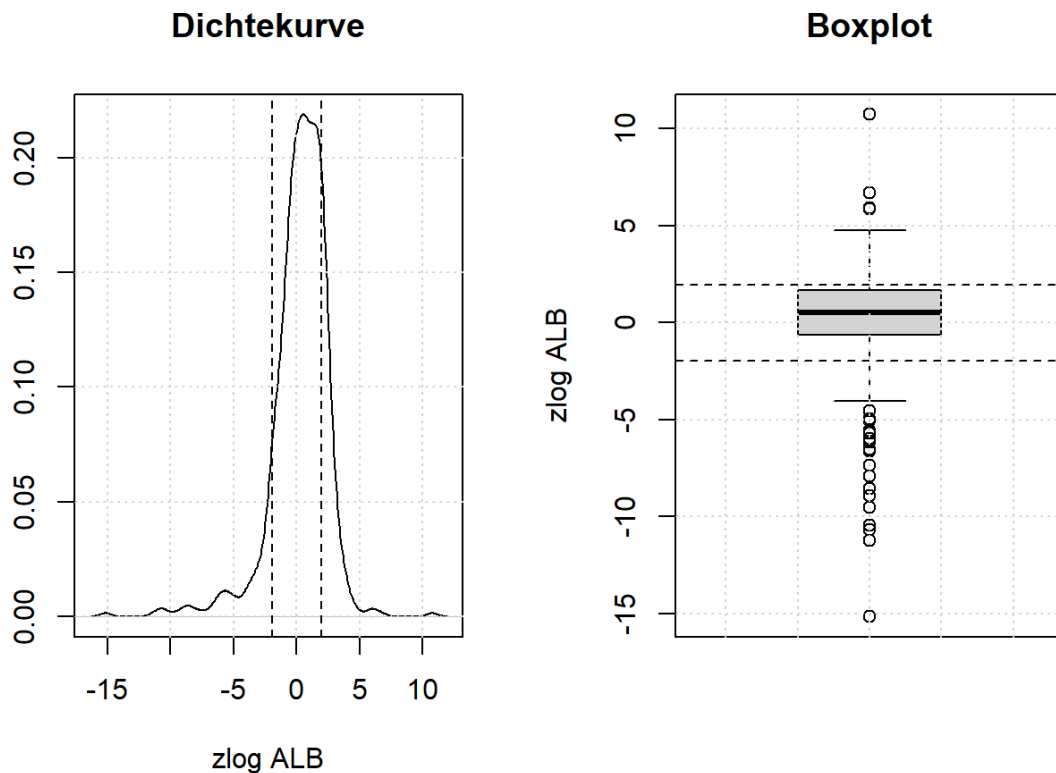
Zur Berechnung von zlog-Werten verwenden wir die zlog-Funktion aus dem zlog-Package (eine Beschreibung erhält man wie üblich mit `?zlog`, wenn das Package korrekt installiert und aktiviert wurde). Die Funktion erwartet als Parameter einen Absolutwert (z. B. 40) und die zugehörigen Referenzgrenzen (z. B. 35.6 und 46.1), und gibt den transformierten Wert zurück.

```
zlog(40, c(35.6, 46.1))  
## [1] -0.1926064
```

Das Ergebnis von etwa -0.2 liegt nahe bei 0 und besagt, dass der Wert mitten im standardisierten Referenzintervall von -1.96 bis +1.96 liegt.

Hier ist ein Codebeispiel für die Berechnung und grafische Darstellung der zlog-Werte von Albumin als Dichtekurve und Boxplot:

```
par(mfrow = c(1, 2))  
zlog.alb <- zlog(xx$ALB, ref[1 : 2, 1])  
plot(density(zlog.alb), main = "Dichtekurve", xlab = "zlog ALB", ylab = "")  
grid()  
abline(v = c(-1.96, 1.96), lty = 2)  
boxplot(zlog.alb, main = "Boxplot", ylab = "zlog ALB", xlab = "")  
grid()  
abline(h = c(-1.96, 1.96), lty = 2)
```



Die Hauptmenge der zlog-Werte streut erwartungsgemäß um 0 und befindet sich zwischen den standardisierten Referenzgrenzen von -1.96 und +1.96. Man sieht der dezentralen Lage der Dichtekurve bzw. der Box aber bereits mit freiem Auge an, dass das vorgegebene Referenzintervall (gestrichelte Linien) nicht gut zu den Messwerten passt.

Hinweis: Dichtekurve und Boxplot wurden für ALB geschlechtsunabhängig erstellt, da die vorgegebenen Grenzwerte für Männer und Frauen gleich sind. Bei unterschiedlichen Grenzwerten erfolgt die Berechnung und grafische Darstellung geschlechtsspezifisch. Beispielsweise würde die zlog-Normalisierung für die ALT-Grenzwerte bei Männern wie folgt berechnet:

```
zlog.alt.m <- zlog(x.m$ALT, ref[3 : 4, 2])
```

Trunkierung der Werte mit dem iBoxplot95-Verfahren

Beim klassischen Boxplot reichen die Enden der Whiskers im obigen Beispiel (rechte Abbildung) weit über die gestrichelten Referenzgrenzen hinaus. Das liegt vor allem an den pathologischen Werten im Datensatz. Das Ziel der Modifikation *iBoxplot95* ist es nun, die Whiskers so weit zu kürzen, dass sie im Idealfall mit den Referenzgrenzen der unauffälligen Werte übereinstimmen. Ausgehend von der Überlegung, dass die zentralen 50 % der Daten (also die Werte innerhalb der Box des Boxplots) so gut wie keine pathologischen Werte enthalten sollten, versuchen wir also, mit einem statistischen Verfahren von den zentralen 50 % auf die zentralen 95 % zu schließen.

Wir wählen als Startpunkt der beiden Whiskers den Median (Q2) und berechnen einen Faktor *qf* ("Quantilfaktor") aus der jeweils kleineren Hälfte der Box (unter der Annahme, dass diese Seite der Verteilung durch die pathologischen Werte weniger beeinflusst wird). Bei einer idealen Normalverteilung sind die beiden Hälften gleich breit und der Quantilfaktor lässt sich leicht aus den Quantilen 0.25 und 0.025 bzw. 0.75 und 0.975 der Standardnormalverteilung berechnen.

```
qnorm(p = 0.025) / qnorm(p = 0.25)
## [1] 2.905847
qnorm(p = 0.975) / qnorm(p = 0.75)
## [1] 2.905847
```

Für den Trunkierungs-Algorithmus gibt es bislang noch kein Package. Die Funktion *iBoxplot95()* ist jedoch relativ leicht selbst zu programmieren. Zur besseren Verständlichkeit gliedern wir den Algorithmus in zwei Funktionen. Die erste namens *truncate.x* führt die eigentliche Trunkierung durch, indem sie von den Quartilen des Datensatzes mithilfe des Quantilfaktors *qf* auf die theoretischen Referenzgrenzen einer Standardnormalverteilung schließt.

```
truncate.x <- function(x, qf = 2.906){
  #qf = quantile factor to derive qnorm(0.025) from qnorm(0.25)
  Q <- quantile(x, c(0.25, 0.5, 0.75))
  var1 <- Q[2] - Q[1]
  var2 <- Q[3] - Q[2]
  var <- min(var1, var2)
  lim <- c(Q[2] - qf * var, Q[2] + qf * var)
  return(subset(x, x >= lim[1] & x <= lim[2]))
}
```

Als Beispiel ergeben sich für Albumin folgende statistischen Kennzahlen:

```
trunc.alb <- truncate.x(zlog.alb)
summary(trunc.alb)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -2.7918 -0.4604  0.6194  0.5836  1.6945  3.8446
```

Diese Zahlen bestätigen den visuellen Eindruck der obigen Abbildungen quantitativ: Der Median liegt nicht bei 0, sondern ist mit etwa 0.6 nach rechts verschoben, und die Extremwerte liegen jenseits der erwarteten Referenzgrenzen von -1.96 bzw. +1.96. Das letztere Phänomen kann man allerdings noch nicht sicher einem zu breiten Referenzintervall anlasten, denn es ist durchaus möglich, dass dieser erste Trunkierungsschritt noch nicht alle pathologischen Werte entfernt hat.

Deshalb ruft die zweite Funktion namens *iBoxplot95* diesen Algorithmus iterativ so oft auf, bis außerhalb der so berechneten Whiskers keine Werte mehr gefunden werden.

```
iBoxplot95 <- function(x, lognorm = FALSE){
  if(lognorm){xx <- log(xx)} else {xx <- x}
  #setzt die Startparameter
  n0 <- 1
  n1 <- 0
  qf <- 2.906
  i <- 1
  #trunkiert xx solange, bis keine Ausreißer mehr gefunden werden
  while (n0 > n1){
    n0 <- length(xx)
    xx <- truncate.x(xx, qf = qf)
    n1 <- length(xx)
    qf <- 3.083
  }
  if (lognorm){xx <- exp(xx)}
  return(xx)
}
```

Eine Besonderheit dieser iterativen Funktion besteht darin, dass der Quantilfaktor *qf* nach dem ersten Durchgang von 2.906 auf 3.083 erhöht wird. Diese geringfügige Verlängerung der Whiskers trägt dem Umstand Rechnung, dass die Trunkierung ab dem zweiten Durchgang aus einer bereits trunkierten Wertemenge erfolgt. Bei der Berechnung von *qf* verwenden wir deshalb für das 0.25-Quantil die Funktion *qtruncnorm* anstelle von *qnorm*, um von der trunkierten auf die untrunkierte Verteilung zu schließen (aus der wir dann das 0.025- und das 0.975-Quantil schätzen).

```
qnorm(p = 0.025) / qtruncnorm(p = 0.25, a = -1.96, b = 1.96)
## [1] 3.083352
```

Mit der so angepassten Funktion *iBoxplot95* erhält man für Albumin folgende Kennzahlen:

```
trunc.alb <- iBoxplot95(zlog.alb)
summary(trunc.alb)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -2.3921 -0.3834   0.6194   0.6012  1.6945   3.6399
```

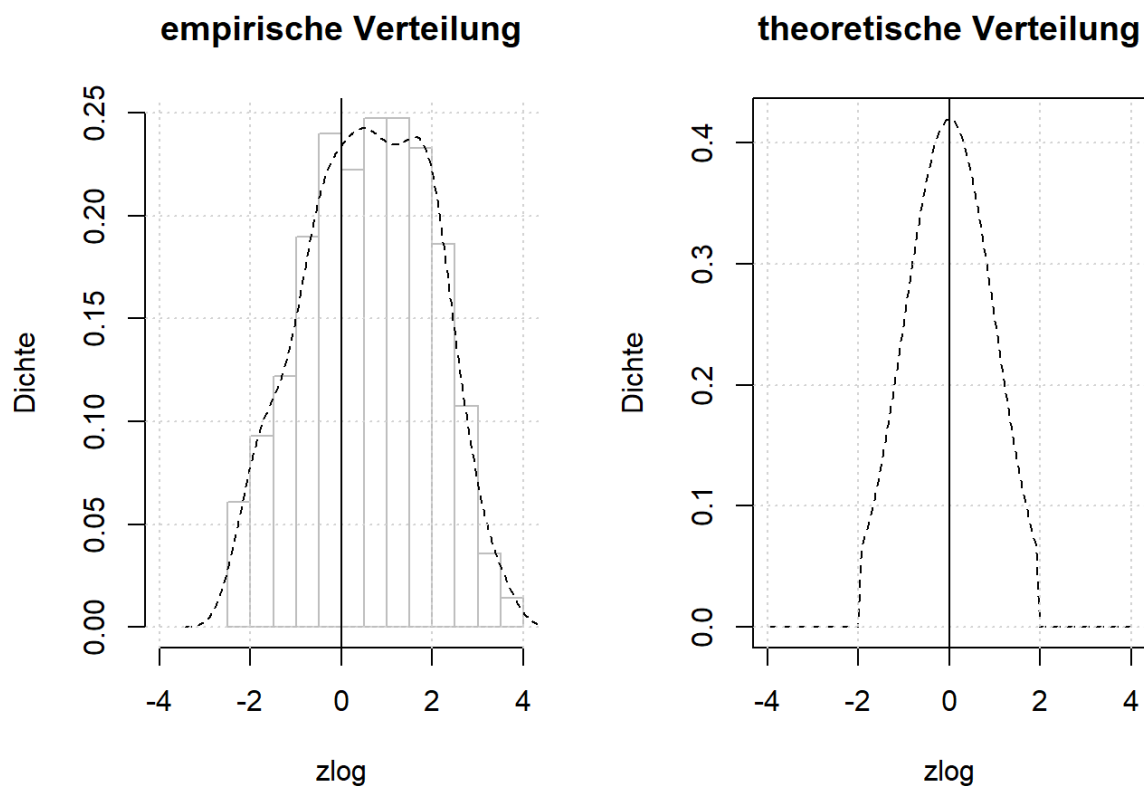
Wie man sieht, sind die Grenzen durch den wiederholten Aufruf von *truncate.x* etwas enger geworden, sprechen jedoch weiterhin dafür, dass das vom Hersteller vorgegebene Referenzintervall zu weit ist.

Kapitel 5

Grafische Ausgabe des Ergebnisses

Dieses Ergebnis wollen wir nun so visualisieren, dass man das Referenzintervall auch ohne tiefergehendes Statistikwissen intuitiv beurteilen kann. Dazu erstellen wir zwei Grafiken, die die tatsächliche Verteilung der trunkierten zlog-Werte mit der theoretisch zu erwartenden Verteilung vergleicht.

```
par(mfrow = c(1, 2))  
hist(trunc.alb, freq = FALSE, xlim = c(-4, 4),  
     col = "white", border = "gray",  
     main = "empirische Verteilung", xlab = "zlog", ylab = "Dichte")  
lines(density(trunc.alb), lty = 2)  
grid()  
abline(v = 0)  
curve(dtruncnorm(x, a = -1.96, b = 1.96), from = -4, to = 4, lty = 2,  
      main = "theoretische Verteilung", xlab = "zlog", ylab = "Dichte")  
grid()  
abline(v = 0)
```



Man sieht links das Histogramm der realen zlog-Werte nach Entfernung der Ausreißer; die gestrichelte Linie zeigt die "empirische Dichtekurve", die sich sehr einfach mit `density(trunc.alb)` berechnen lässt. Rechts sehen wir zum Vergleich, wie die Dichtekurve aussehen müsste, wenn

die zlog-Werte in etwa standardnormalverteilt wären (Mittelwert 0, Standardabweichung 1), und wenn das vorgegebene Referenzintervall exakt den Quantilen 0.025 und 0.975 entspräche.

Der Unterschied ist augenfällig: Die empirische Dichtekurve erscheint andeutungsweise zweigipflig (bimodal), wobei der zweite Peak nach rechts verschoben ist. Wenn wir uns noch einmal die Boxplots für Albumin im [Kapitel 3](#) ansehen, so bestätigt dieses Bild unseren Verdacht, dass ein gemeinsames Referenzintervall für beide Geschlechter die Realität nicht gut widerspiegelt, weil die Werte der Männer etwas höher als diejenigen der Frauen sind.

Kapitel 6

Ampelfarben als Interpretationshilfe

Nun wollen wir die Abweichungen zwischen Soll und Ist klinisch bewerten und farbig kennzeichnen. Angelehnt an eine einfache Daumenregel von IFCC und EFLM ([Henny J et al. 2016](#)) zur RI-Überprüfung wird eine Abweichung von $\pm 10\%$ des normierten Referenzbereichs als tolerierbar angesehen. Ausgedrückt in zlog-Werten entspricht dies ± 0.392 , weil das standardisierte Referenzintervall ja unabhängig vom Analyten immer $2 * 1.96 = 3.92$ beträgt.

Die nachfolgende Funktion *zlog.density* kombiniert die obigen beiden Dichtekurven und kennzeichnet Abweichungen zwischen empirischen und theoretischen Referenzgrenzen mit Ampelfarben: Grün werden Abweichungen von weniger als $\pm 10\%$ des Referenzbereichs gekennzeichnet, rot markieren wir mehr als doppelt so große Abweichungen und gelb den Bereich dazwischen.

```

zlog.density <- function(x, limits, main = ""){
  # x = Vektor positiver Zahlen, limits = Vektor der Länge 2 mit den vorgegebenen R
  eferenzgrenzen

  zlog.x <- zlog(x, limits) # berechnet zlog-Werte

  trunc.x <- iBoxplot95(zlog.x) # filtert die zentralen 95 % der "wahrscheinlich no
  rmalen" Werte

  # zeichnet eine kombinierte Grafik aus empirischer und theoretischer Verteilung
  d <- density(trunc.x)
  y.max <- max(max(d$y), 0.45) # obere Begrenzung der y-Achse
  hist(trunc.x, freq = FALSE,
       xlim = c(-4, 4), ylim = c(0, y.max),
       col = "white", border = "gray",
       main = main, xlab = "", ylab = "")
  lines(d, lty = 2)
  curve(dtruncnorm(x, a = -1.96, b = 1.96), from = -4, to = 4, add = TRUE)
  grid()
  abline(v = 0)

  # zeichnet farbige Toleranzbereiche mit Ampelfarben (tf.col = traffic light color
  s) ein
  tf.col <- c(rgb(0.1, 1, 0.1, 0.5), rgb(1, 1, 0.1, 0.5), rgb(1, 0.1, 0.1, 0.5))
  verif.col <- c(tf.col[1], tf.col[1]) # Voreinstellung zweimal Grün
  col.lim <- c(-2.744, -2.352, -1.568, -1.176, 1.176, 1.568, 2.352, 2.744)

  if(min(trunc.x) < col.lim[2] | min(trunc.x) > col.lim[3]){verif.col[1] <- tf.col[
  2]}
  if(min(trunc.x) < col.lim[1] | min(trunc.x) > col.lim[4]){verif.col[1] <- tf.col[
  3]}
  if(max(trunc.x) < col.lim[6] | max(trunc.x) > col.lim[7]){verif.col[2] <- tf.col[
  2]}
  if(max(trunc.x) < col.lim[5] | max(trunc.x) > col.lim[8]){verif.col[2] <- tf.col[
  3]}

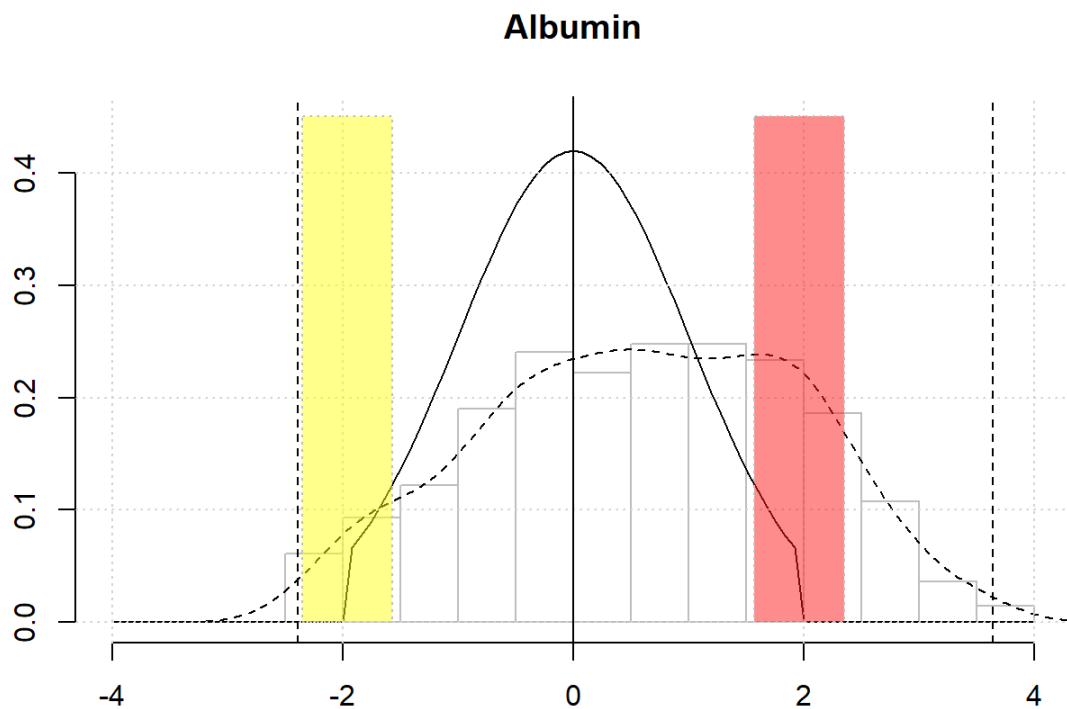
  rect(col.lim[2], 0, col.lim[3], y.max, col = verif.col[1], border = "gray", lty =
  3)
  rect(col.lim[6], 0, col.lim[7], y.max, col = verif.col[2], border = "gray", lty =
  3)
  abline(v = c(min(trunc.x), max(trunc.x)), lty = 2)

  return(c(low.lim = round(min(trunc.x), 2), upp.lim = round(max(trunc.x), 2)))
}

```

Mithilfe dieser Funktion erstellen wir nun das farbig markierte Dichtediagramm für Albumin.

```
zlog.density(xx$ALB, ref[1 : 2, 1], main = "Albumin")
```



```
## low.lim upp.lim  
## -2.39 3.64
```

Aus dem Bild und den darunter abgedruckten zlog-Werten ergeben sich folgende zwei Schlussfolgerungen:

- Die empirische Untergrenze von -2.39 liegt knapp unterhalb der Toleranzgrenze von -2.35. Der linke Toleranzbereich ist deshalb gelb gefärbt.
- Die empirische Obergrenze von 3.64 liegt weit oberhalb der Toleranzgrenze von +2.35. Der rechte Toleranzbereich ist deshalb rot gefärbt.

Mit wenigen Zeilen Code lässt sich diese Grafik für alle zu prüfenden Referenzintervalle erstellen. Die Funktionsergebnisse werden in eine Tabelle geschrieben und ausgegeben.

```

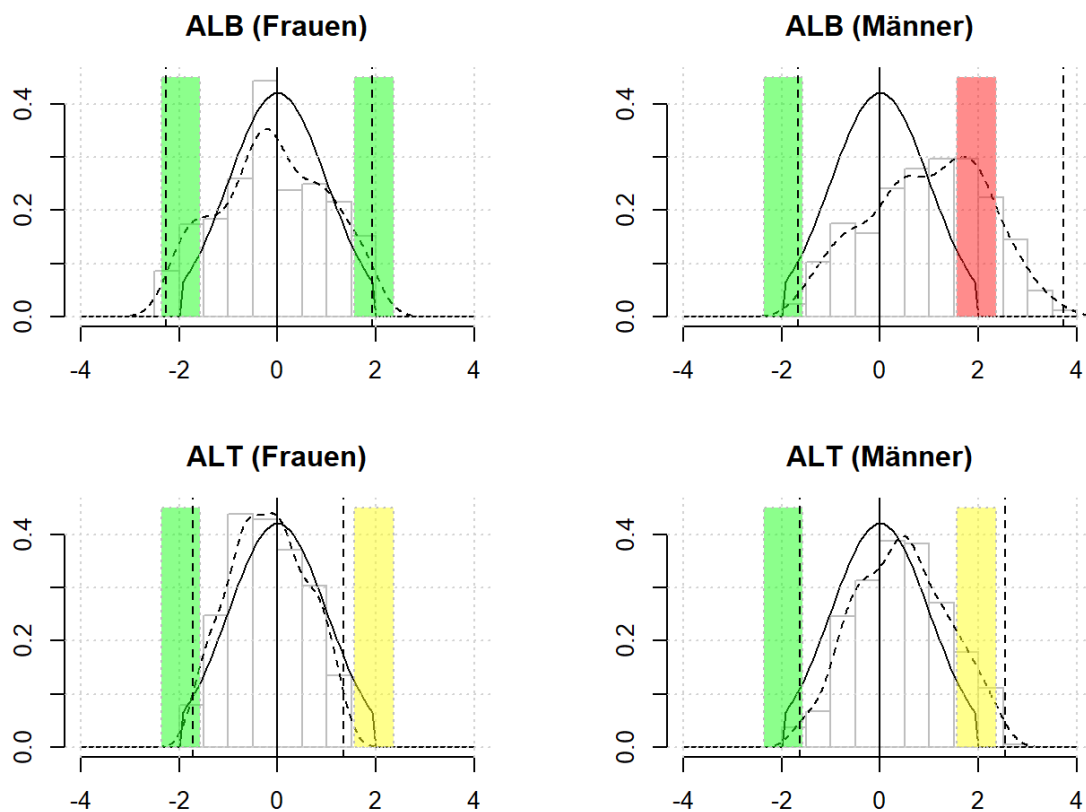
zlog.results <- matrix(NA, nrow = 4, ncol = 2)
rownames(zlog.results) <- rownames(ref)
colnames(zlog.results) <- c("ALB Schätzung", "ALT Schätzung")

par(mfrow = c(2, 2), mar=c(3, 4.1, 3, 2.1)) # vier Grafiken mit verkleinerten Rändern (margins)

zlog.results[1 : 2, 1] <- zlog.density(x.f$ALB, ref[1 : 2, 1], main = "ALB (Frauen)")
zlog.results[3 : 4, 1] <- zlog.density(x.m$ALB, ref[3 : 4, 1], main = "ALB (Männer)")

zlog.results[1 : 2, 2] <- zlog.density(x.f$ALT, ref[1 : 2, 2], main = "ALT (Frauen)")
zlog.results[3 : 4, 2] <- zlog.density(x.m$ALT, ref[3 : 4, 2], main = "ALT (Männer)")

```



```

zlog.results
##           ALB Schätzung ALT Schätzung
## Frauen UG          -2.26          -1.72
## Frauen OG           1.93           1.34
## Männer UG          -1.66          -1.62
## Männer OG           3.73           2.53

```

Auch diese Abbildungen sind sehr leicht interpretierbar:

- Von den insgesamt acht überprüften Referenzgrenzen liegen fünf im “grünen Bereich”, sprich innerhalb des Toleranzbereichs von 1.96 ± 0.392 (Obergrenze) bzw. -1.96 ± 0.392 (Untergrenze).
- Die zwei Obergrenzen der ALT werden gelb markiert, sollten also einer Inspektion (zum Beispiel durch Vergleich mit der Literatur) unterzogen werden.
- Die geschlechtsübergreifend angegebene Obergrenze für Albumin ist nur für Frauen geeignet. Für Männer wurde sie rot markiert; hier sollte vermutlich ein höherer Grenzwert eingesetzt werden.

Ein Blick in die Literatur zeigt, dass die Grenzwerte für Albumin in der Tat in Abhängigkeit von Alter und Geschlecht sowie dem eingesetzten Analyseverfahren variieren können und deshalb individuell angepasst werden müssen ([Weaving G et al. Ann Clin Biochem 2016](#), [Jassam N et al. Ann Clin Biochem 2020](#))

Kapitel 7

Schätzung absoluter Zielwerte

Das hier vorgestellte Screeningverfahren ist kein Ersatz für eine sorgfältige Bestimmung von Referenzintervallen nach den Leitlinien und Empfehlungen internationaler und nationaler Fachgesellschaften. Dennoch ist aus den Abbildungen natürlich die Richtung ablesbar, in der der korrekte Grenzwert zu suchen ist.

Mithilfe der Funktion *izlog()* (*inverse zlog*) lassen sich zlog-Werte bei Kenntnis des zugrunde liegenden Referenzintervalls in Absolutwerte umrechnen. Das nachfolgende Codebeispiel ermittelt absolute Zielwerte aus der Tabelle *zlog.results*.

```
absolute.results <- zlog.results
absolute.results[1 : 4, 1] <- izlog(zlog.results[1 : 4, 1], ref[1 : 2, 1])
absolute.results[1 : 2, 2] <- izlog(zlog.results[1 : 2, 2], ref[1 : 2, 2])
absolute.results[3 : 4, 2] <- izlog(zlog.results[3 : 4, 2], ref[3 : 4, 2])
cbind(ref, round(absolute.results, 1))

##           ALB ALT ALB Schätzung ALT Schätzung
## Frauen UG 35.6  10           34.9          10.8
## Frauen OG 46.1  35           46.0          28.7
## Männer UG 35.6  10           36.3          11.5
## Männer OG 46.1  50           51.8          63.2
```

Die Spalten 1 und 2 dieser Tabelle enthaltenen die vorgegebenen Referenzgrenzen, die Spalten 3 und 4 die aus den zlog-Werten geschätzten Zielwerte. Auch hier sieht man, dass vor allem die Obergrenze für das Albumin der Männer erheblich von der Vorgabe des Herstellers abweicht (51.8 vs. 46.1 g/l); der Unterschied von 5.7 g/l entspricht immerhin mehr als 50 % des Referenzbereichs ($46.1 - 35.6 = 10.5$ g/l). Ob die Abweichungen der ALT-Obergrenzen um etwa 6 bzw. 11 U/l (rund 25 % des Referenzbereichs) klinisch relevant sind, sollte das jeweilige Labor individuell entscheiden.

Mit der *izlog*-Funktion lassen sich auch die Toleranzgrenzen aus der “IFCC-Daumenregel” in Absolutwerten angeben.

```

tol.abs <- matrix(0, nrow = 3, ncol = 6)
rownames(tol.abs) <- c("ALB", "ALT(f)", "ALT(m)")
colnames(tol.abs) <- c("UG", "von", "bis", "OG", "von", "bis")
tol.abs[, 1] <- c(ref[1, 1 : 2], ref[3, 2]) #Übertragung der Untergrenzen aus ref
tol.abs[, 4] <- c(ref[2, 1 : 2], ref[4, 2]) #Übertragung der Obergrenzen aus ref
tol.zlog <- c(-2.352, -1.568, 1.568, 2.352) #-1.96 ± 0.392 und 1.96 ± 0.392
for (i in 1 : 3){
  lim <- tol.abs[i, c(1, 4)] # Referenzgrenzen für die Funktion izlog
  tol.abs[i, 2] <- izlog(tol.zlog[1], limits = lim)
  tol.abs[i, 3] <- izlog(tol.zlog[2], limits = lim)
  tol.abs[i, 5] <- izlog(tol.zlog[3], limits = lim)
  tol.abs[i, 6] <- izlog(tol.zlog[4], limits = lim)
}
round(tol.abs, 1)
##           UG  von  bis   OG  von  bis
## ALB      35.6 34.7 36.5 46.1 44.9 47.3
## ALT(f)  10.0  8.8 11.3 35.0 30.9 39.7
## ALT(m)  10.0  8.5 11.7 50.0 42.6 58.7

```

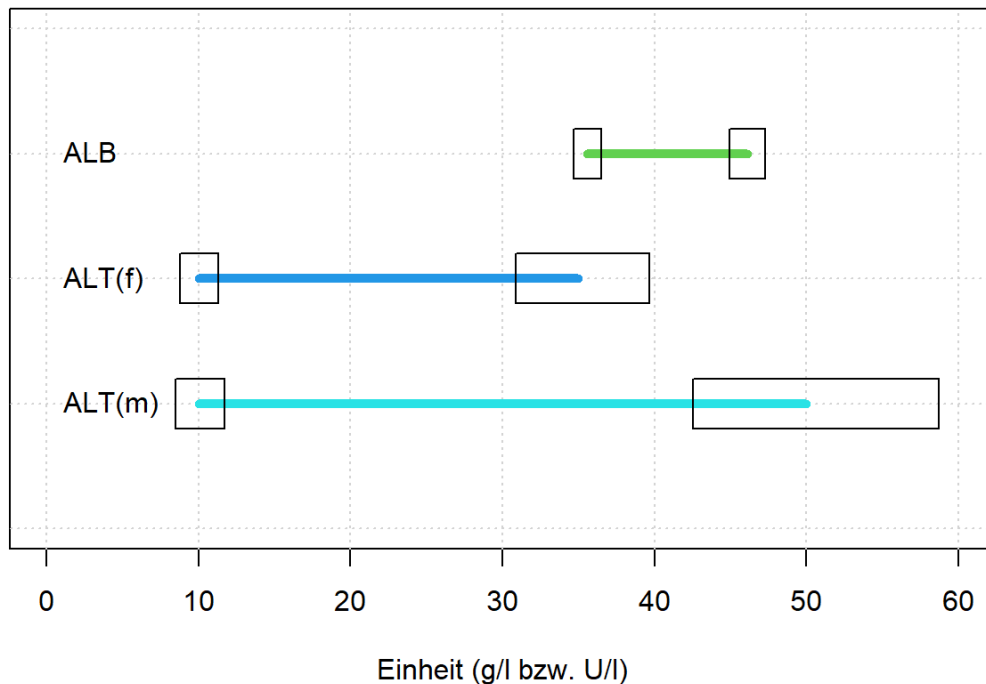
Wie man sieht, bewirkt die in der zlog-Normalisierung enthaltene Logarithmierung, dass der Toleranzbereich für die untere Referenzgrenze enger als derjenige für die Obergrenze ist. Dies entspricht einer Forderung der einschlägigen DGKL-Arbeitsgruppe ([Haeckel R et al. CCLM 2017](#)). Grafisch lassen sich die vorgegebenen Referenzintervalle und Toleranzbereiche zum Beispiel mit Linien und Kästchen folgendermaßen darstellen:

```

plot(0, type = "n", xlim = c(0, 60), ylim = c(0, 4), yaxt = "n",
     main = "Referenzintervalle und Toleranzbereiche", xlab = "Einheit (g/l bzw. U/
l)", ylab = "")
grid()
for (i in 1 : 3){
  text(x = 0, y = 4 - i, rownames(tol.abs)[i], pos = 4)
  lines(x = tol.abs[i, c(1, 4)], y = rep(4 - i, 2), lwd = 5, col = i + 2)
  rect(tol.abs[i, 2], 3.8 - i, tol.abs[i, 3], 4.2 - i)
  rect(tol.abs[i, 5], 3.8 - i, tol.abs[i, 6], 4.2 - i)
}

```

Referenzintervalle und Toleranzbereiche



Das hier vorgeschlagene empirische Prüfverfahren legt beim Albumin recht strenge Maßstäbe an die Referenzintervall-Überprüfung an, während er bei der Obergrenze der ALT "toleranter" ist. Dies ist klinisch plausibel, weil der Organismus den Albuminspiegel deutlich enger reguliert als denjenigen der Transaminasen, sodass sich Abweichungen im ersten Fall auch klinisch stärker auswirken als im zweiten Fall.

Kapitel 8

Diskussion und Ausblick

Unsere Studie schließt eine Lücke im Repertoire der Methoden für die RI-Prüfung. Bislang gibt es nur ein einziges leitliniengerechtes Verfahren, das ohne großen organisatorischen und rechnerischen Aufwand feststellt, ob ein vorgegebenes Referenzintervall zu den eigenen Messwerten passt oder nicht: Man führt hierfür Messungen an zwanzig "offensichtlich gesunden" Versuchspersonen durch und akzeptiert die Grenzen, wenn nicht mehr als zwei Messwerte außerhalb liegen ([Horowitz G et al. CLSI/IFCC Document EP28-A3c, 2010](#)).

Dieses Verfahren wurde vielfach als unzureichend kritisiert, insbesondere weil

- ein n von 20 kaum repräsentativ für das Einsenderkollektiv sein kann.
- die erlaubte Zahl von zwei Ausreißern mehr als die erwarteten 5 % falsch positiver Resultate (also Zurückweisungen von theoretisch korrekten Referenzgrenzen) bewirkt.
- zu weite Referenzintervalle mit diesem Verfahren grundsätzlich nicht erkannt werden können, weil in diesem Fall zu wenige Werte außerhalb liegen.

Dazu kommt, dass der Rekrutierungsaufwand weiterhin relativ hoch ist, da die 20 gesunden Versuchspersonen ja pro Subkollektiv (Frauen, Männer, Kinder usw.) anfallen. Deshalb wurden in jüngerer Zeit viele indirekte Verfahren publiziert, die diese Probleme beheben sollen. Sie sind allerdings in der Regel zeit- und rechenaufwendig und benötigen sehr hohe Fallzahlen, die in kleineren Laboratorien nicht zur Verfügung stehen.

Unser Ziel war es, ein indirektes Verfahren vorzuschlagen, das in Millisekunden mit überschaubarer Fallzahl zu einer Aussage führt, ob die vorgegebenen Referenzgrenzen akzeptiert werden können oder nicht. Im Gegensatz zur derzeit üblichen Vorgehensweise bestimmen wir kein Referenzintervall *de novo*, sondern verwenden – ähnlich wie die o.g. CLSI/IFCC-Leitlinie – die vorgegebenen Grenzwerte. Aber anstelle der pauschalen Aussage, dass nicht mehr als 10 % der gemessenen Werte außerhalb liegen dürfen, analysieren wir die Verteilung der Werte zwischen den Grenzen und können auf diese Weise wesentlich detailliertere Aussagen machen als das Leitlinienverfahren. Durch den Einsatz von zlog-Werten anstelle von Absolutwerten erreichen wir zudem eine standardisierte Auswertung, die für alle Analyten unabhängig von Verfahren, Einheit usw. einheitlich beurteilt werden kann.

Wir demonstrieren in dieser Arbeit, dass man solch detaillierte Aussagen mit einer ganzen Reihe visueller Verfahren untermauern kann. Neben Boxplots und Scatterplots eignen sich insbesondere Histogramme und Dichtekurven dazu, recht genau anzugeben, ob beispielsweise die Lage oder die Streuung der gemessenen Werte von der Erwartung abweicht. Weitere Darstellungsformen wie z. B. Quantil-Quantil-Plots von zlog-Werten sind aktuell in der Erprobung, um die Aussagen noch präziser und intuitiver zu machen.

Wir arbeiten daran, die vorgeschlagenen Kennzahlen und Grafiken an einer Vielzahl von Analyten und Kollektiven zu testen und laden unsere Kolleginnen und Kollegen aus Labormedizin und Statistik dazu ein, sich an diesen Studien mit ihren eigenen Daten zu beteiligen.

Danksagung

Die Autoren bedanken sich bei Jonas Heisterkamp, Magdeburg, für das Korrekturlesen dieses Manuskripts.