

Inferenzielle Statistik und ROC-Analyse

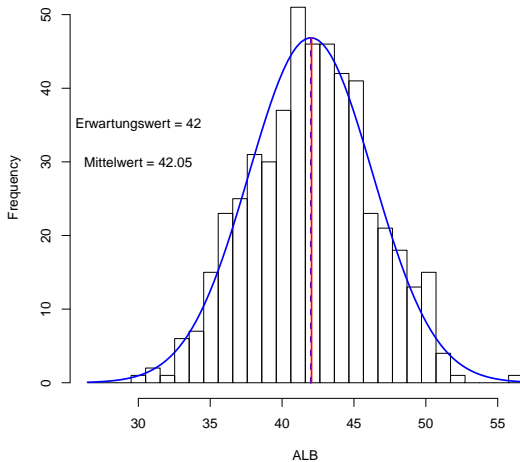
A. Bietenbeck, G. Hoffmann, F. Klawonn, C. Winter

AG Bioinformatik

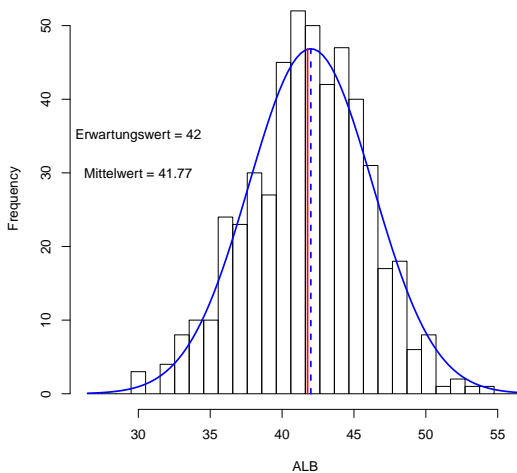
- ▶ t-Test (Mittelwertvergleich)
- ▶ Signifikanz
- ▶ Prinzip der Hypothesentests
- ▶ Wilcoxon-Test
- ▶ F-Test (Varianzvergleich)
- ▶ Fishers exakter Test
- ▶ Test auf Normalverteilung
- ▶ Zeichnen von ROC und Berechnung von AUC-Werten

- ▶ Die **beschreibende (deskriptive) Statistik** visualisiert Daten und berechnet Kennzahlen, ohne Annahmen über die Daten bzw. den Erzeugungsprozess der Daten zu treffen.
- ▶ Die **schließende (inferenzielle) Statistik** trifft Annahmen über den Erzeugungsprozess der Daten – z.B. Normalverteilungsannahme – und kann so Schlussfolgerungen aus den Daten ziehen, etwa Parameterschätzung, Konfidenzintervalle, Hypothesen.

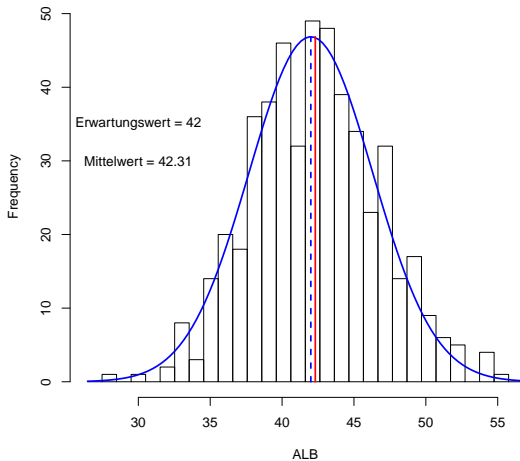
Theoretische Verteilung und Schätzwerte



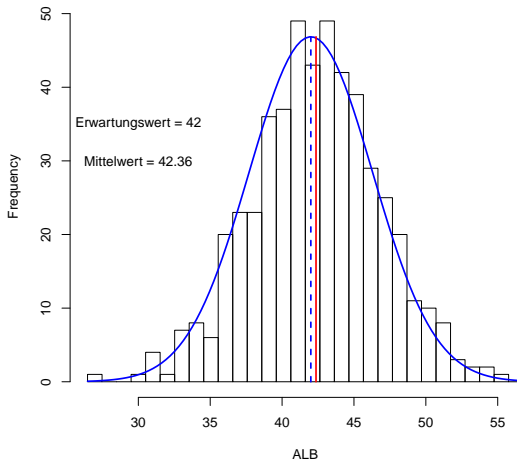
Theoretische Verteilung und Schätzwerte



Theoretische Verteilung und Schätzwerte



Theoretische Verteilung und Schätzwerte



(zweiseitiger zwei-Stichproben) t-Test zum Vergleich der Mittelwerte zweier Stichproben

- ▶ **Voraussetzung:** Die beiden Stichproben stammen aus Normalverteilungen.
- ▶ **Nullhypothese:** Die Erwartungswerte (theoretischen Mittelwerte) der beiden Normalverteilungen sind gleich.
- ▶ **Alternativhypothese:** Die Erwartungswerte (theoretischen Mittelwerte) der beiden Normalverteilungen sind ungleich.
- ▶ **Teststatistik:** Differenz der Mittelwerte (in Relation zu den Standardabweichungen).
- ▶ **p-Value:** Wahrscheinlichkeit, dass sich **unter der Nullhypothese** rein zufällig eine mindestens so große Differenz der Mittelwerte ergibt (in Relation zu den Standardabweichungen).

Vergleich der Mittelwerte von ALP nach Behandlung mit Dosis A bzw. D:

zunächst: Herausfiltern der zu vergleichenden Stichproben mittels der subset-Funktion:

```
> x <- read.csv(file.choose())  
> alp.a <- subset(x$ALP.M, x$dose=="A")  
> alp.d <- subset(x$ALP.M, x$dose=="D")
```

subset erwartet als erstes Argument einen Datenvektor oder eine Tabelle (Data Frame) und als zweites Argument ein Auswahlkriterium. Typische Vergleichsoperatoren sind ==, >, >=, <, <= und !=.

```
> t.test(alp.a,alp.d)
```

```
Welch Two Sample t-test
```

```
data: alp.a and alp.d
```

```
t = -2.2715, df = 234.15, p-value = 0.02402
```

```
alternative hypothesis: true difference in  
means is not equal to 0
```

```
95 percent confidence interval:
```

```
-13.3710739 -0.9500453
```

```
sample estimates:
```

```
mean of x mean of y
```

```
59.60526 66.76582
```

```
> t.test(alp.a,alp.d)
```

Welch Two Sample t-test

```
data:  alp.a and alp.d
```

```
t = -2.2715, df = 234.15, p-value = 0.02402
```

```
alternative hypothesis: true difference in  
means is not equal to 0
```

```
95 percent confidence interval:
```

```
-13.3710739  -0.9500453
```

```
sample estimates:
```

```
mean of x mean of y
```

```
59.60526  66.76582
```

```
> t.test(alp.a,alp.d)
```

Welch Two Sample t-test

data: alp.a and alp.d

t = -2.2715, df = 234.15, p-value = 0.02402

alternative hypothesis: true difference in
means is not equal to 0

95 percent confidence interval:

-13.3710739 -0.9500453

sample estimates:

mean of x mean of y

59.60526 66.76582

```
> t.test(alp.a,alp.d)
```

```
Welch Two Sample t-test
```

```
data:  alp.a and alp.d
```

```
t = -2.2715, df = 234.15, p-value = 0.02402
```

```
alternative hypothesis: true difference in  
means is not equal to 0
```

```
95 percent confidence interval:
```

```
-13.3710739 -0.9500453
```

```
sample estimates:
```

```
mean of x mean of y
```

```
59.60526 66.76582
```

```
> t.test(alp.a,alp.d)
```

Welch Two Sample t-test

data: alp.a and alp.d

t = -2.2715, df = 234.15, p-value = 0.02402

alternative hypothesis: true difference in
means is not equal to 0

95 percent confidence interval:

-13.3710739 -0.9500453

sample estimates:

mean of x mean of y

59.60526 66.76582

```
> t.test(alp.a,alp.d)
```

Welch Two Sample t-test

data: alp.a and alp.d

t = -2.2715, df = 234.15, p-value = 0.02402

alternative hypothesis: true difference in
means is not equal to 0

95 percent confidence interval:

-13.3710739 -0.9500453

sample estimates:

mean of x mean of y

59.60526 66.76582

```
> t.test(alp.a,alp.d)
```

Welch Two Sample t-test

data: alp.a and alp.d

t = -2.2715, df = 234.15, p-value = 0.02402

alternative hypothesis: true difference in
means is not equal to 0

95 percent confidence interval:

-13.3710739 -0.9500453

sample estimates:

mean of x mean of y

59.60526 66.76582


```
> alp.b <- subset(x$ALP.M,x$dose=="B")  
> t.test(alp.a,alp.b)
```

Welch Two Sample t-test

```
data:  alp.a and alp.b  
t = 0.80363, df = 296.32, p-value = 0.4223  
alternative hypothesis: true difference in  
means is not equal to 0  
95 percent confidence interval:  
-2.453124  5.839326  
sample estimates:  
mean of x mean of y  
59.60526  57.91216
```

Einseitiger t-Test in R

Meist geht es darum nachzuweisen, dass ein Medikament oder eine Behandlung zu einer Besserung führt.

Man ist beim t-Test dann nur daran interessiert, ob der Mittelwert der behandelten Gruppe besser ist – je nach Laborwert entweder höher oder niedriger – als in der Kontrollgruppe.

Änderung der Alternativhypothese von „ungleich“ zu „kleiner“ bzw. „größer“.

Man spricht dann von einem **einseitigen** Test.

In R muss dazu beim Test der optionale Parameter `alternative` auf `less` bzw. `greater` gesetzt werden. (s. `?t.test`)

Einseitige Tests liefern kleinere p-Values als zweiseitige Tests.

Einseitiger t-Test in R

```
> t.test(alp.a,alp.d,alternative="less")
```

Welch Two Sample t-test

data: alp.a and alp.d

t = -2.2715, df = 234.15, p-value = 0.01201

alternative hypothesis: true difference in
means is less than 0

95 percent confidence interval:

-Inf -1.954875

sample estimates:

mean of x mean of y

59.60526 66.76582

Gepaarter t-Test in R

Beim Vergleich der Dosen A und D bzw. A und B sind die zu vergleichenden Stichproben jeweils unabhängig.

Es wurden in den Gruppen jeweils andere Patienten behandelt.

Bei **gepaarten Stichproben** gibt es eine eins-zu-eins-Zuordnung zwischen den Werten der beiden Stichproben.

Meist: Die beiden Stichproben stammen von denselben Patienten, jeweils vor und nach der Behandlung.

In R muss dazu beim Test der optionale Parameter `paired` auf `TRUE` gesetzt werden.

Gepaarter t-Test in R für Dosis D (vorher gegen nachher)

```
> alp.d.b <- subset(x$ALP.B,x$dose=="D")  
> t.test(alp.d.b,alp.d,paired=TRUE)
```

Paired t-test

```
data:  alp.d.b and alp.d  
t = -5.7039, df = 157, p-value = 5.666e-08  
alternative hypothesis: true difference in  
means is not equal to 0  
95 percent confidence interval:  
-16.325861 -7.927303  
sample estimates:  
mean of the differences  
-12.12658
```

Ungepaarter t-Test für Dosis D (vorher gegen nachher) X

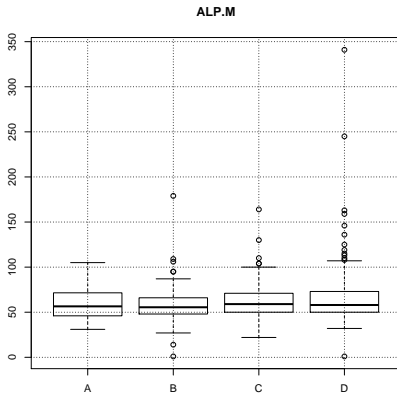
```
> t.test(alp.d.b,alp.d)

Welch Two Sample t-test

data:  alp.d.b and alp.d
t = -3.8895, df = 227.58, p-value =
  0.0001319
alternative hypothesis: true difference in
  means is not equal to 0
95 percent confidence interval:
 -18.269921  -5.983243
sample estimates:
mean of x mean of y
 54.63924  66.76582
```

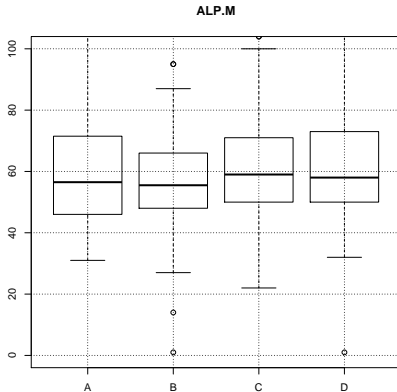
Tukey: There is no excuse for failing to plot and look

```
> boxplot(x$ALP.M ~ x$dose, main="ALP.M")  
> grid(col="black")
```



Tukey: There is no excuse for failing to plot and look **X**

```
> boxplot(x$ALP.M ~ x$dose, main="ALP.B", ylim  
          =c(0,100))  
> grid(col="black")
```



Zoom auf das Intervall [0,100]

Eigenschaften von statistischen Hypothesentests:

- ▶ Bei sehr kleinen Stichproben liefern Hypothesentests selten statistische Signifikanz, d.h. p-Values sind fast immer größer als 0.05.
- ▶ Bei sehr großen Stichproben liefern Hypothesentests fast immer statistische Signifikanz, d.h. man erhält fast immer kleine p-Values, auch wenn die Unterschiede in den Stichproben minimal sind.
- ▶ Statistische und medizinische Signifikanz sind nicht dasselbe. Der p-Value sagt nichts darüber, ob der Unterschied medizinisch bedeutsam ist.

Vergleich von Dosis A und B mit größeren Stichproben

Dazu: Vervierfachung bzw. Verachtfachung der Stichprobengröße (hier: durch Replizierung der Daten).

Mit dem R-Kommando `rbind` (row bind) lassen sich zwei oder mehrere Tabellen (Data Frames) mit der gleichen Spaltenstruktur und den gleichen Spaltennamen aneinanderhängen.

Vergleich von Dosis A und B mit größeren Stichproben X

```
> nrow(x) # Anzahl der Zeilen von x
[1] 606
> xr <- rbind(x,x,x,x)
> nrow(xr)
[1] 2424
> x[1,2:5]
  ALP.B  ALT.B  AST.B  TBL.B
1    80    13    14 12.654
> xr[c(1,607),2:5]
  ALP.B  ALT.B  AST.B  TBL.B
1    80    13    14 12.654
607   80    13    14 12.654
```

Vergleich von Dosis A und B mit größeren Stichproben X

```
> alp.a.r <- subset(xr$ALP.M,xr$dose=="A")  
> alp.b.r <- subset(xr$ALP.M,xr$dose=="B")  
> t.test(alp.a.r,alp.b.r)
```

Welch Two Sample t-test

```
data:  alp.a.r and alp.b.r  
t = 1.6113, df = 1191.3, p-value = 0.1074  
alternative hypothesis: true difference in  
means is not equal to 0  
95 percent confidence interval:  
-0.3684566  3.7546586  
sample estimates:  
mean of x mean of y  
59.60526  57.91216
```

Vergleich von Dosis A und B mit größeren Stichproben

```
> xr <- rbind(x,x,x,x,x,x,x,x)
> alp.a.r <- subset(xr$ALP.M,xr$dose=="A")
> alp.b.r <- subset(xr$ALP.M,xr$dose=="B")
> t.test(alp.a.r,alp.b.r)
```

Welch Two Sample t-test

```
data:  alp.a.r and alp.b.r
t = 2.2797, df = 2384.5, p-value = 0.02271
alternative hypothesis: true difference in
  means is not equal to 0
95 percent confidence interval:
 0.2367095 3.1494925
sample estimates:
mean of x mean of y
59.60526  57.91216
```

Allgemein besteht ein **Hypothesentest** aus:

- ▶ **Voraussetzungen/Annahmen** über die Verteilungen der Daten
- ▶ Einer **Nullhypothese**. Das Ziel besteht in der Medizin meist in der Ablehnung der Nullhypothese.
- ▶ Einer **Alternativhypothese**.
- ▶ Einer **Teststatistik**, die eine „sinnvolle Kennzahl“ für die Nullhypothese ist.
- ▶ Als Ergebnis einen **p-Value**: Wahrscheinlichkeit, dass sich **unter der Nullhypothese** rein zufällig ein so „extremer“ Wert für die Teststatistik gibt.

Wilcoxon/Mann-Whitney-U-Test

(zweiseitiger zwei-Stichproben) Wilcoxon/Mann-Whitney-U-Test
zum Vergleich zweier Stichproben

- ▶ **Voraussetzung:** Keine speziellen Verteilungsannahmen (außer Stetigkeit, d.h. doppelte Werte sollten nicht vorkommen).
- ▶ **Nullhypothese:** Wenn man zufällig einen Wert aus der ersten und einen Wert aus der zweiten Verteilung zieht, beträgt die Wahrscheinlichkeit 50%, dass der erste Wert größer ist als der zweite.
- ▶ **Alternativhypothese:** Die Wahrscheinlichkeit ist ungleich 50%.
- ▶ **Teststatistik:** Rangsumme.
- ▶ **p-Value:** Wahrscheinlichkeit, dass sich **unter der Nullhypothese** rein zufällig eine entsprechende große bzw. kleine Rangsumme ergibt.

Wilcoxon/Mann-Whitney-U-Test

(zweiseitiger zwei-Stichproben) Wilcoxon/Mann-Whitney-U-Test zum Vergleich der **Mediane** zweier Stichproben

- ▶ **Voraussetzung:** Keine speziellen Verteilungsannahmen, außer dass die beiden Verteilungen die gleiche Form bis auf eine Verschiebung haben sollten.
- ▶ **Nullhypothese:** Der Mediane der beiden Verteilungen sind gleich.
- ▶ **Alternativhypothese:** Der Mediane der beiden Verteilungen sind ungleich.
- ▶ **Teststatistik:** Rangsumme.
- ▶ **p-Value:** Wahrscheinlichkeit, dass sich unter der Nullhypothese rein zufällig eine entsprechende große bzw. kleine Rangsumme ergibt.

Wilcoxon/Mann-Whitney-U-Test

(zweiseitiger zwei-Stichproben) Wilcoxon/Mann-Whitney-U-Test
zum Vergleich zweier Stichproben

- ▶ **Voraussetzung:** Keine speziellen Verteilungsannahmen (außer Stetigkeit, d.h. doppelte Werte sollten nicht vorkommen).
- ▶ **Nullhypothese:** Wenn man zufällig einen Wert aus der ersten und einen Wert aus der zweiten Verteilung zieht, beträgt die Wahrscheinlichkeit 50%, dass der erste Wert größer ist als der zweite.
- ▶ **Alternativhypothese:** Die Wahrscheinlichkeit ist ungleich 50%.
- ▶ **Teststatistik:** Rangsumme.
- ▶ **p-Value:** Wahrscheinlichkeit, dass sich **unter der Nullhypothese** rein zufällig eine entsprechende große bzw. kleine Rangsumme ergibt.

Wilcoxon-Test in R: Dosis A gegen D

```
> wilcox.test(alp.a,alp.d)

      Wilcoxon rank sum test with
      continuity correction

data:  alp.a and alp.d
W = 10843, p-value = 0.1398
alternative hypothesis: true location shift
      is not equal to 0
```

Der t-Test lieferte in diesem Fall einen p-Value von $p = 0.02402$.

Wilcoxon-Test in R: Dosis A gegen B im verachtfachen Datensatz



```
> wilcox.test(alp.a.r,alp.b.r)
```

```
Wilcoxon rank sum test with  
continuity correction
```

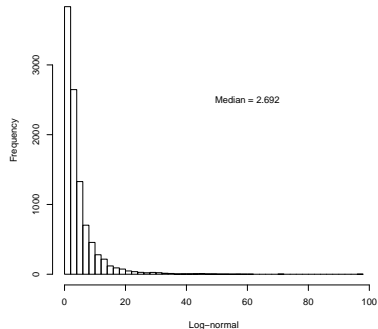
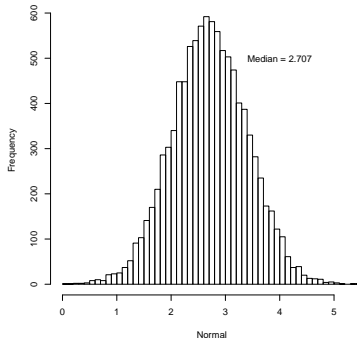
```
data: alp.a.r and alp.b.r
```

```
W = 747330, p-value = 0.1056
```

```
alternative hypothesis: true location shift  
is not equal to 0
```

Der t-Test lieferte in diesem Fall einen p-Value von $p = 0.02271$.

Wilcoxon-Test: Gleiche Mediane, ungleiche Verteilungen



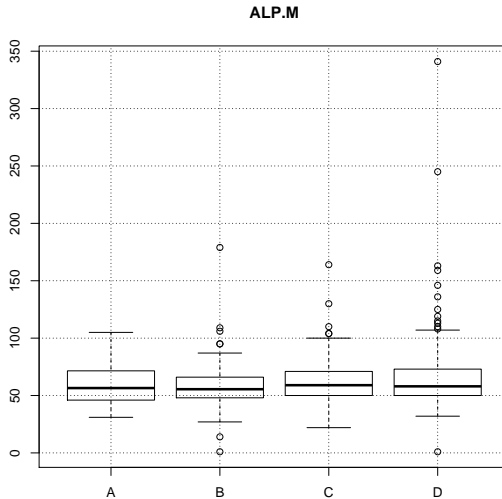
Der Wilcoxon-Test liefert einen p-Value von 0.0063.

t-Test versus Wilcoxon-Test

- ▶ Der t-Test vergleicht Mittelwerte.
- ▶ Bei kleinen Stichprobengrößen können Abweichungen von der Normalverteilungsannahme zu falschen p-Values führen – sowohl zu große als auch zu kleine sind möglich, je nach Art der Verteilungen.
- ▶ Bei größeren Stichprobengrößen spielen Abweichungen von der Normalverteilungsannahme (aufgrund des zentralen Grenzwertsatzes) (fast) keine Rolle.
- ▶ Will man mit dem Wilcoxon-Test Mediane vergleichen, liefert der Test bei kleinen Stichproben annähernd korrekte p-Values, auch wenn die Annahme der bis auf Verschiebung gleichen Verteilung der beiden Stichproben nicht gilt.
- ▶ Bei großen Stichproben kann der Wilcoxon-Test die Nullhypothese auch dann ablehnen, wenn die Mediane gleich sind, sofern die Annahme der bis auf Verschiebung gleichen Verteilung der beiden Stichproben nicht gilt.

F-Test zum Vergleich der Varianzen zweier Stichproben

- ▶ **Voraussetzung:** Die beiden Stichproben stammen aus Normalverteilungen.
- ▶ **Nullhypothese:** Die (theoretischen) Varianzen der beiden Normalverteilungen sind gleich.
- ▶ **Alternativhypothese:** Die Varianzen der beiden Normalverteilungen sind ungleich.
- ▶ **Teststatistik:** Quotient der beiden (Stichproben-)Varianzen.
- ▶ **p-Value:** Wahrscheinlichkeit, dass sich **unter der Nullhypothese** rein zufällig eine mindestens so große Abweichung der Teststatistik zum Wert 1 ergibt.



```
> var.test(alp.a,alp.d)
```

```
      F test to compare two variances
```

```
data:  alp.a and alp.d
```

```
F = 0.25499, num df = 151, denom df = 157,
```

```
p-value = 4.221e-16
```

```
alternative hypothesis: true ratio of  
variances is not equal to 1
```

```
95 percent confidence interval:
```

```
0.1857877 0.3503121
```

```
sample estimates:
```

```
ratio of variances
```

```
0.2549885
```


Fishers exakter Test als (Un-)Abhängigkeitstest zweier binärer Variablen

Daten liegen als 2×2 -Kontingenztafel mit Fallzahlen vor.

Typische Anwendungen von Fishers exaktem Test in der Epidemiologie:

	Kein Risikofaktor R	Risikofaktor R
Hat Krankheit K
Hat Krankheit K nicht

Beispiel: R : Rauchen, K : Lungenkrebs

Fishers exakter Test: Korrelation und Kausalität

Mit Fishers exaktem Test kann nur ein Zusammenhang (Korrelation) zwischen zwei binären Variablen festgestellt werden, nicht aber eine Kausalität.

Fiktives Beispiel:

	Sterbealter ≥ 80	Sterbealter < 80
Anzahl Krankenhaustage ≤ 30	107	59
Anzahl Krankenhaustage > 30	12	42

Fishers exakter Test in R: Korrelation und Kausalität

Eingabe der Kontingenztafel als Matrix

```
> contable <- matrix(c(107,12,59,42),nrow=2,  
  byrow=TRUE)
```

- ▶ Mit `matrix` wird eine Matrix erzeugt.
- ▶ Der Vektor `c(107,12,59,42)` enthält die Einträge der Matrix.
- ▶ `nrow` gibt die Anzahl der Zeilen der Matrix an. (Die Anzahl der Spalten wird dann automatisch berechnet.)
- ▶ `byrow=TRUE` besagt, dass die Matrix zeilenweise aus dem Datenvektor eingelesen werden soll.

```
> contable  
      [,1] [,2]  
[1,]  107  12  
[2,]   59  42
```

```
> fisher.test(contable)
```

```
Fisher's Exact Test for Count Data
```

```
data: contable
```

```
p-value = 9.696e-08
```

```
alternative hypothesis: true odds ratio is  
not equal to 1
```

```
95 percent confidence interval:
```

```
2.974507 14.196655
```

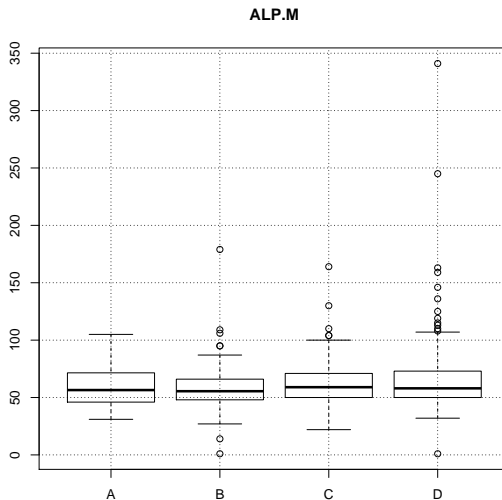
```
sample estimates:
```

```
odds ratio
```

```
6.291177
```

- ▶ Es gibt diverse Tests, die Daten auf Normalverteilung testen.
- ▶ Die Nullhypothese dieser Tests lautet immer: Die Daten stammen aus einer Normalverteilung.
- ▶ Das Ziel bei Tests auf Normalverteilung besteht aber in der Regel darin, die Nullhypothese zu bestätigen und nicht abzulehnen.
- ▶ Bei kleinen Stichprobengrößen kann die Normalverteilungshypothese fast nie abgelehnt werden.
- ▶ Bei großen Stichproben können auch minimale Abweichungen von der Normalverteilungsannahme zur Ablehnung der Normalverteilungsannahme führen.

Test auf Normalverteilung: Shapiro-Wilk-Test in R



Test auf Normalverteilung: Shapiro-Wilk-Test in R

```
> shapiro.test(alp.a)

      Shapiro-Wilk normality test

data:  alp.a
W = 0.95837, p-value = 0.0001562

> shapiro.test(log10(alp.a)) # logarithmisch

      Shapiro-Wilk normality test

data:  log10(alp.a)
W = 0.98256, p-value = 0.05146
```

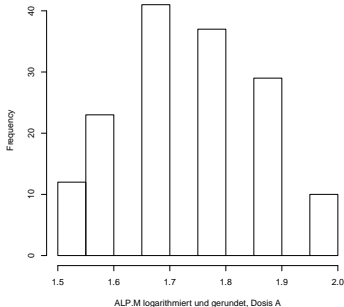
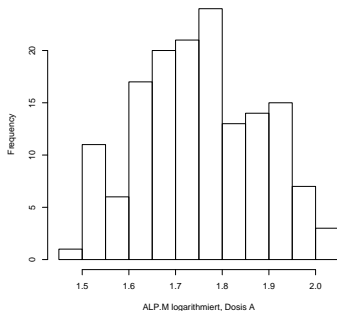
Ablehnung der Normalverteilungshypothese der Originaldaten,
(knappe) Nicht-Ablehnung der Normalverteilungshypothese bei den
logarithmierten Daten.

Auf eine Nachkommstelle gerundete logarithmierte Werte:

```
> shapiro.test(round(log10(alp.a),1))  
  
      Shapiro-Wilk normality test  
  
data:  round(log10(alp.a), 1)  
W = 0.93751, p-value = 2.999e-06
```

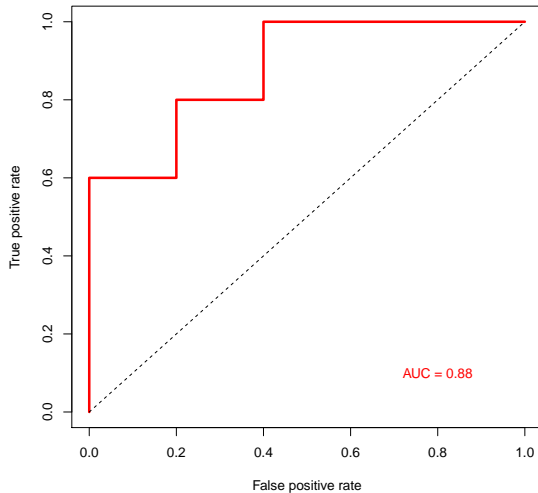
Durch das Runden auf zwei Stellen wird die Normalverteilungshypothese aufgrund des Binning-Effekts klar verworfen.

Test auf Normalverteilung: Shapiro-Wilk-Test in R



ROC-Kurven und AUC

1
2
3
4
5
6
7
8
9
10



Mit dem R-Paket pROC lassen sich ROC-Analysen sehr gut durchführen.

Das Paket beinhaltet auch den Datensatz aSAH (Aneurysmal subarachnoid hemorrhage).

- ▶ Ziel: Frühzeitige Vorhersage langfristiger, irreversibler Hirnschäden während der akuten Phase bei aSAH-Patienten.
- ▶ 113 Patienten, 72 mit gutem, 41 mit schlechtem Outcome.
- ▶ S100 β
- ▶ NDKA-Werte (nucleotide diphosphate kinase A)
- ▶ Geschlecht
- ▶ Alter

Installieren eines R-Paketes

Schritt 1

Schritt 2

Schritt 3

Schritt 4

R version 3.4.3 (2017-11-30) -- "Kite-eating Tree"
Copyright (c) 2017 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/x86_64 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[workspace loaded from ~/.Rdata]

Install Packages

Install from: **Repository (CRAN, RStudio)**

Repository (CRAN, RStudio)

Package (separate multiple with space or comma):

pROC

Documents/R/win-library/3.4 [Default]

Install Cancel

Environment History Connections

Global Environment

Environment is empty

Library

Name	Description	Version
acepack	ACE and AICs for Selecting Multiple Regression Transformations	1.4.1
anim.plots	Simple Animated Plots for R	0.2
animation	A Gallery of Animations in Statistics and Utilities to Create Animations	2.5
arules	Mining Association Rules and Frequent Bernsets	1.6-0
asserthat	Easy Pre and Post Assertions	0.2.0
backports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.2
Barnard	Barnard's Unconditional Test	1.8
base64enc	Tools for base64 encoding	0.1-3
bayesplot	Plotting for Bayesian Models	1.5.0
best	Bayesian Estimation Supersedes the t-Test	0.5.1
BH	Boost C++ Header Files	1.66.0-1
bindr	Parameterized Active Bindings	0.1.1
bindrcpp	An 'Rcpp' Interface to Active Bindings	0.2
Biobase	Biobase: Base functions for Bioconductor	2.38.0
BiocGenerics	S4 generic functions for Bioconductor	0.24.0
BiocInstaller	Install/Update Bioconductor, CRAN, and github Packages	1.28.0
Biocomb	Feature Selection and Classification with the Embedded Validation Procedures for Biomedical Data Analysis	0.3
bitops	Bitwise Operations	1.0-6
bsom	Convert Statistical Analysis Objects into Tidy Data Frames	0.4.5
car	Companion to Applied Regression	2.1-6
carData	Companion to Applied Regression Data Sets	3.0-0
caTools	Tools: moving window statistics, GIF, Base64, ROC AUC, etc.	1.17.1
checkmate	Fast and Versatile Argument Checks	1.8.5
cli	Helpers for Developing Command Line Interfaces	1.0.0
code	Output Analysis and Diagnostics for MCMC	0.19-1

ROC-Analyse in R

```
> library(pROC)
Type 'citation("pROC")' for a citation.
```

```
Attache Paket: "pROC"
```

```
The following objects are masked from "
  package:stats":
```

```
    cov, smooth, var
```

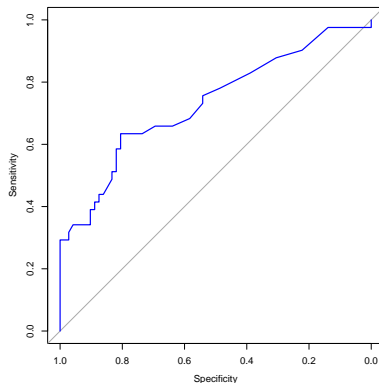
```
> data(aSAH)
```

```
> summary(aSAH)
```

gos6	outcome	gender	age	wfns	s100b	ndka
1:28	Good:72	Male :42	Min. :18.0	1:39	Min. :0.030	Min. : 3.01
2: 0	Poor:41	Female:71	1st Qu.:42.0	2:32	1st Qu.:0.090	1st Qu.: 9.01
3:13			Median :51.0	3: 4	Median :0.140	Median :12.22
4: 6			Mean :51.1	4:16	Mean :0.247	Mean : 19.66
5:66			3rd Qu.:61.0	5:22	3rd Qu.:0.330	3rd Qu.: 17.30
			Max. :81.0		Max. :2.070	Max. :419.19

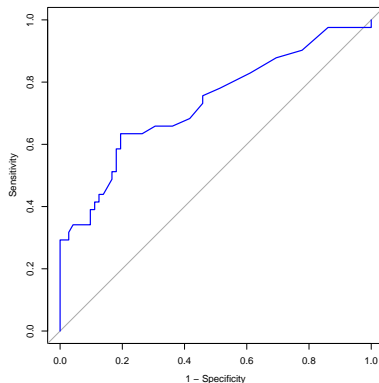
ROC-Analyse in R: ROC-Kurve für S100 β

```
> roc.s100b <- roc(outcome ~ s100b, data=
  aSAH)
> plot(roc.s100b,col="blue")
```

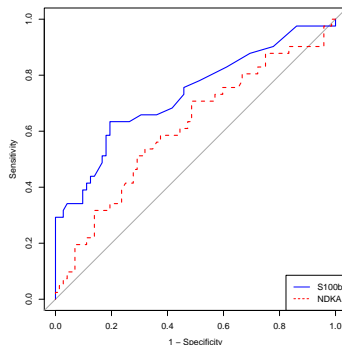


Statt Spezifität die Falsch-Positiv-Rate auf der x-Achse

```
> plot(roc.s100b,col="blue",legacy.axes=TRUE)
```



```
> plot(roc.s100b,col="blue",legacy.axes=T)
> plot(roc.ndka,col="red",add=T,lty=2)
> legend("bottomright",legend=c("S100b","NDKA"),col=c("blue","red"),lty=c(1,2))
```



ROC-Analyse in R: Hinzufügen der NDKA-Kurve

- ▶ TRUE kann auch durch T abgekürzt werden.
- ▶ Entsprechend kann FALSE durch F abgekürzt werden.
- ▶ add=T bedeutet, dass der Plot zu dem bestehenden Plot hinzugefügt werden soll.
- ▶ lty=2 bedeutet, dass die Linie gestrichelt werden soll.
- ▶ legend fügt eine Legende in die Grafik.
- ▶ Standarpositionen topleft, topright, bottomleft, bottomright. Alternativ: x/y-Koordinaten angeben.
- ▶ In legend müssen die Beschriftungen für die Legende angegeben werden.
- ▶ In col können die Farben für die Legendensymbole angegeben werden.
- ▶ In lty (line type) können die Linienarten für die Legendensymbole angegeben werden. (1: durchgezogen, 2: gestrichelt, 3: gepunktet,...)

```
> roc.s100b$auc  
Area under the curve: 0.7314  
> roc.ndka$auc  
Area under the curve: 0.612
```

Schwellenwerte für die verschiedenen Werte auf der ROC-Kurve

```
> roc.s100b$thresholds
[1]  -Inf  0.035  0.045  0.055  0.065  0.075
    0.085  0.095  0.105  0.115  0.125  0.135
    0.145  0.155  0.165  0.175  0.185  0.205
    0.225  0.235
[21]  0.245  0.255  0.265  0.275  0.290  0.310
    0.325  0.335  0.345  0.365  0.395  0.420  0.435
    0.445  0.455  0.465  0.475  0.485  0.495
    0.510
[41]  0.540  0.570  0.640  0.705  0.725  0.755
    0.795  0.840  0.910  1.515    Inf
```

Sensitivitäten für die entsprechenden Schwellenwerte

```
> roc.s100b$sensitivities  
[1] 1.000 0.976 0.976 0.976 0.976 0.902  
    0.878 0.829 0.780 0.756 0.732 0.683  
    0.659 0.659 0.634 0.634 0.634 0.634  
    0.610 0.585  
[21] 0.585 0.561 0.537 0.512 0.512 0.488  
    0.463 0.439 0.439 0.415 0.415 0.390 0.390  
    0.341 0.341 0.341 0.341 0.317 0.293  
    0.293  
[41] 0.268 0.244 0.220 0.195 0.146 0.122  
    0.098 0.073 0.049 0.024 0.000
```

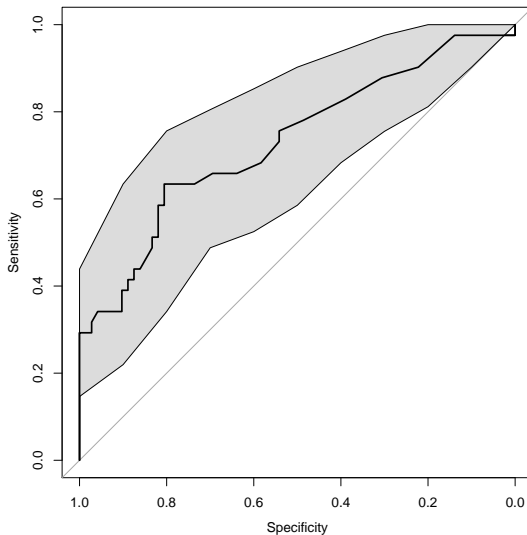
Spezifitäten für die entsprechenden Schwellenwerte

```
> roc.s100b$specificities
[1] 0.000 0.000 0.069 0.111 0.139 0.222
    0.306 0.389 0.486 0.542 0.542 0.583
    0.639 0.694 0.736 0.764 0.778 0.806
    0.806 0.806
[21] 0.819 0.819 0.819 0.819 0.833 0.833
    0.847 0.861 0.875 0.875 0.889 0.889 0.903
    0.903 0.917 0.931 0.958 0.972 0.972
    1.000
[41] 1.000 1.000 1.000 1.000 1.000 1.000
    1.000 1.000 1.000 1.000 1.000
```

Konfidenzband für die Sensitivitäten in der ROC-Kurve

```
> roc.s100b.ci <- roc(outcome ~ s100b,data=
  aSAH,ci=T)
> ci.s100b <- ci.se(roc.s100b.ci)
> plot(roc.s100b.ci)
> plot(ci.s100b,type="shape")
Warning message:
In plot.ci.se(ci.s100, type = "shape") : Low
  definition shape.
```

Konfidenzband für die Sensitivitäten in der ROC-Kurve



Warnmeldung aufgrund geringer Abtastrate entlang der Spezifitäten.

Vermeidbar durch

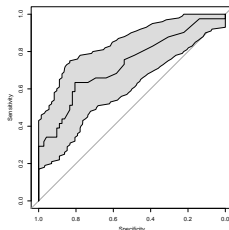
```
ci.s100b <- ci.se(roc.s100b.ci, specificities  
  =seq(0,1,0.01))
```

Abtastung der Spezifitäten von 0 bis 1 in 0,01-Schritten.

Konfidenzband für die Spezifitäten in der ROC-Kurve (analog)



```
> roc.s100b.ci <- roc(outcome ~ s100b,data=
  aSAH,ci=T)
> ci.s100b <- ci.sp(roc.s100b.ci,
  sensitivities=seq(0,1,0.01))
> plot(roc.s100b.ci)
> plot(ci.s100b,type="shape")
```



Datentabellen

`subset`

`nrow`

`rbind`

Hypothesentests

`t.test`

`wilcoxon.test`

`var.test`

`fisher.test`

`shapiro.test`

ROC-Analyse

`roc`

`ci.se`

`ci.sp`

`plot`

Zahlenwerte

`round`

`log10`