2016SS_Statistik_Vorlesung_0622.R

ste

Sun Jun 26 11:08:43 2016

```
# Statistik 22.6.2016
```

```
# Lade Daten
D <- read.csv("http://goo.gl/KCGF16", sep=";", dec=",")
```

```
Geschlecht = D$Geschlecht
# transformiere in 1/0 Variable mit 1 für Mann
X = as.numeric(D$Geschlecht)-1

# Männeranteil, also Mittelwert aus Grundgesamtheit
mu = mean(X)
```

```
# Teste, ob Männeranteil in GG von höchstens 50%
# zu Signifikanzniveau 1% verworfen werden kann

# (1):
alpha = 0.05
#      H0: Männeranteil ist in GG 50%
#      H1: Anteil ist kleiner als 50%
mu.0 = 0.50



# (2) Verwerfungsbereich: Obere Grenze
#                         (unten: -unendlich)
B.oben = qnorm(p=alpha)
B.oben
```

```
## [1] -1.644854
```

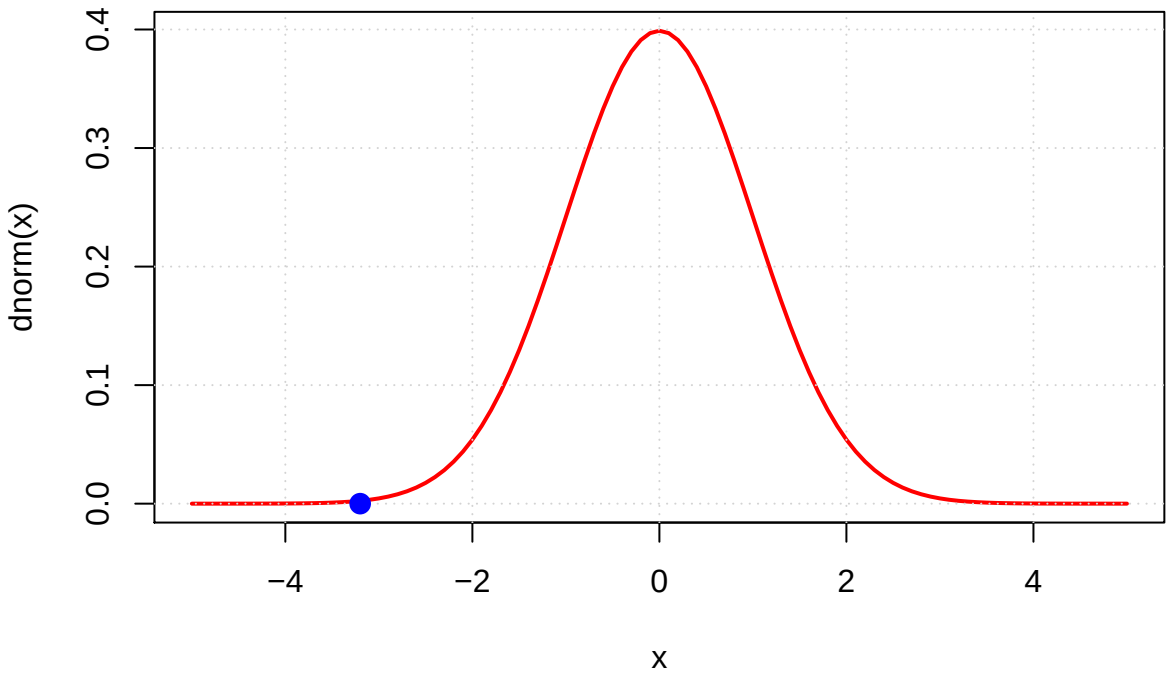
```
# (3) Stichprobe ziehen und Testwert berechnen
set.seed(8)
x.Stichprobe = sample(X, size = 100, replace=TRUE)
n = length(x.Stichprobe)
x.quer = mean(x.Stichprobe)
v =  (x.quer - mu.0)*sqrt(n)/(sqrt(mu.0*(1-mu.0)))
v
```

```
## [1] -3.2
```

```
table(x.Stichprobe)
```

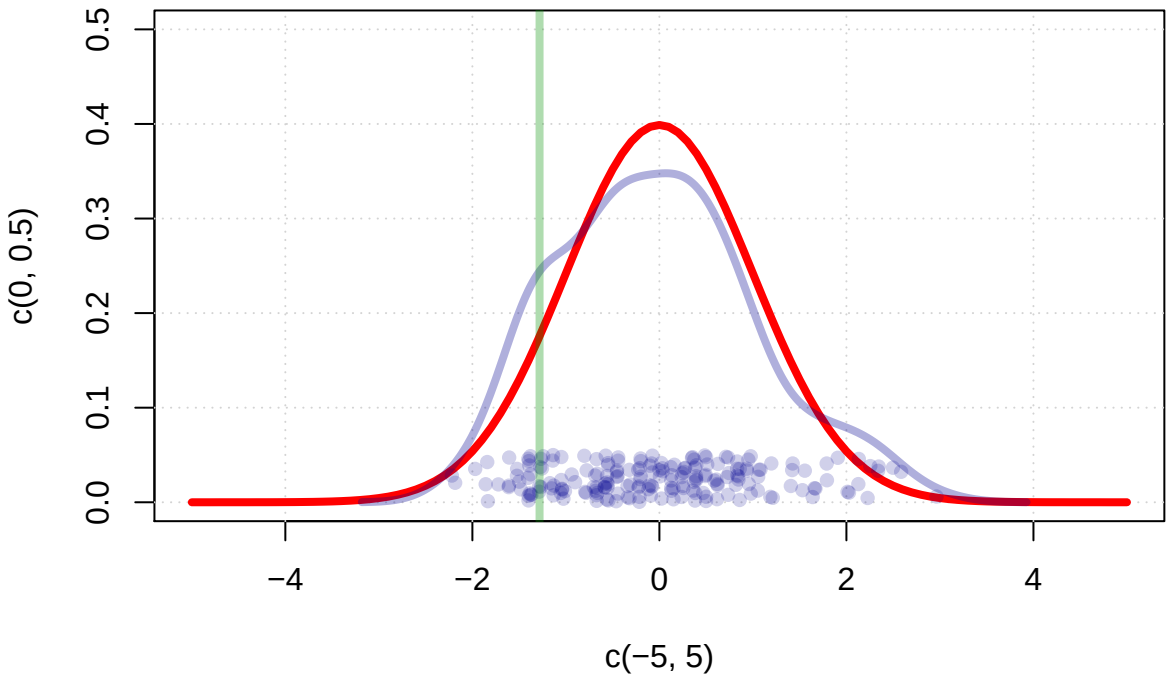
```
## x.Stichprobe
##  0  1
## 66 34
```

```r
curve(dnorm, from=-5, to=5, lwd=2, col="red")
grid()
points(c(v),c(0),col="blue", pch=16, cex=1.5)
```

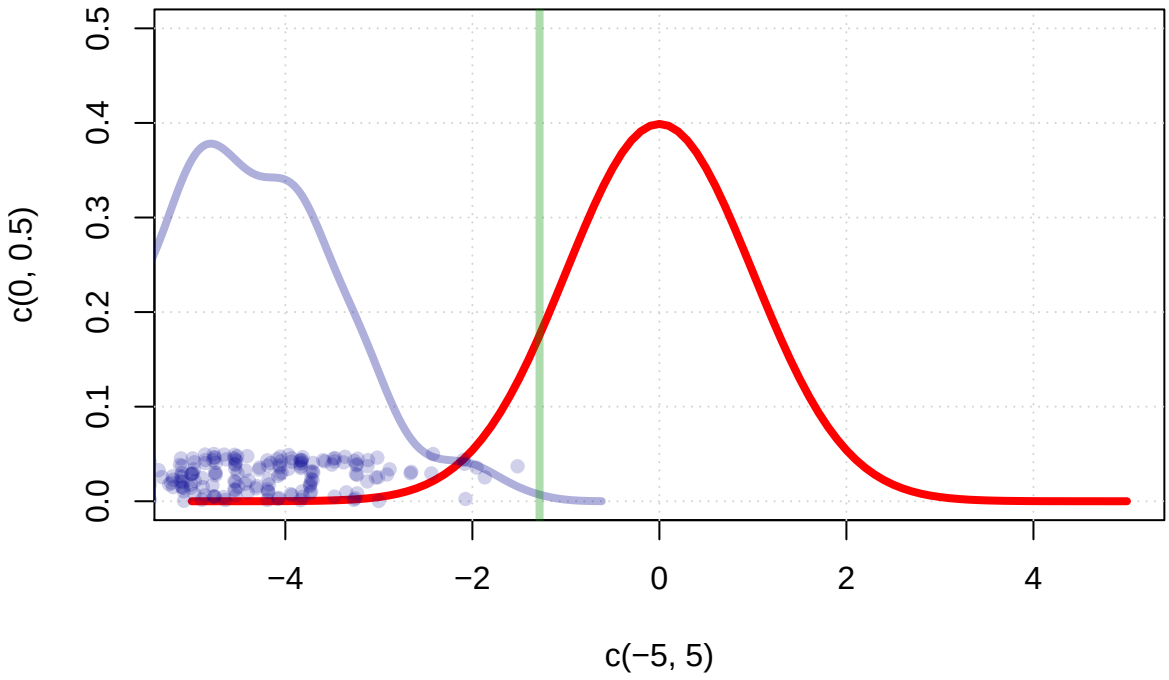



```r
Test.plot = function() {
x.quer.Stichproben = c()
v.Stichproben = c()
# set.seed(2)
for(i in 1:N) {
  x.quer.Stichproben[i] = mean(sample(X, size = n, replace=TRUE))
  v.Stichproben[i] =  (x.quer.Stichproben[i] - mu.0)*sqrt(n)/(sqrt(mu.0*(1-mu.0)))
}

plot(c(-5,5),c(0,0.5), type="n")
grid()
curve(dnorm, from=-5, to=5, lwd=4, col="red", add=TRUE)
points(jitter(v.Stichproben), runif(N, min=0, max = 0.05),
       cex=1, col="#00009030", pch=16)
lines(density(v.Stichproben), lwd=4, col="#00009050")
abline(v=B.oben, lwd=4, col="#00900050")
}

N = 200          # Anzahl Stichproben
n = 300         # Umfang einzelne Stichprobe
alpha = 0.10
```

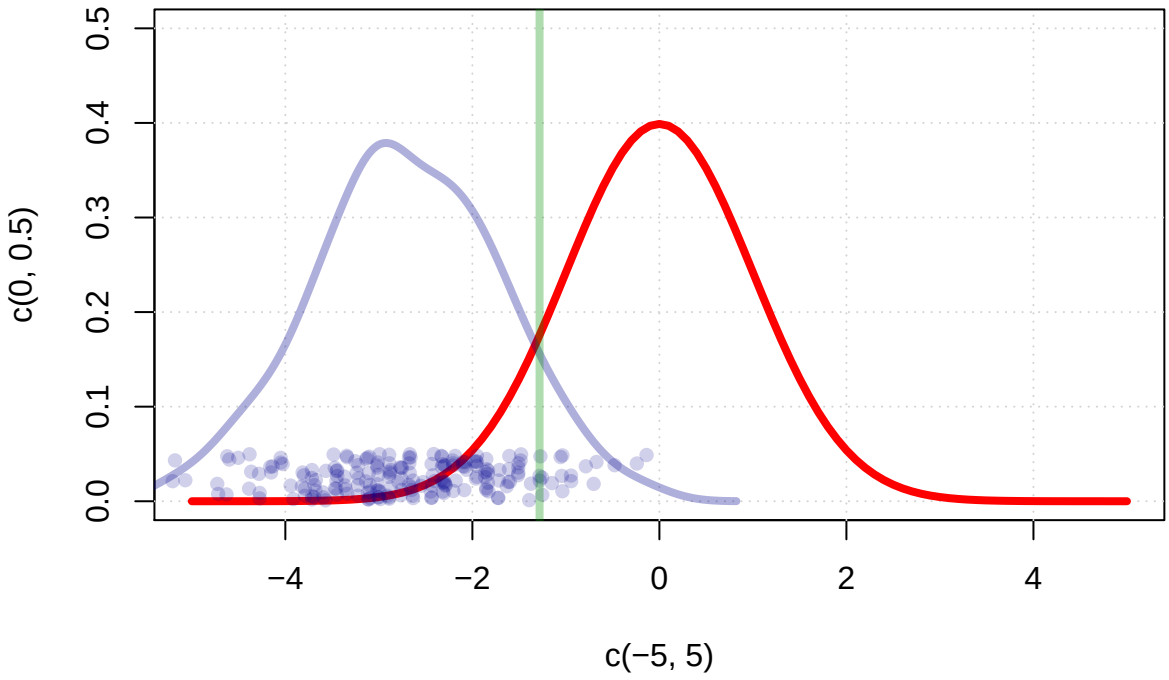
```r
B.oben = qnorm(p=alpha)
mu   = mean(X)

# Experiment, um Fehler erster Art zu zeigen
mu.0 = mu      # Nullhypothese stimmt jetzt
Test.plot()
```



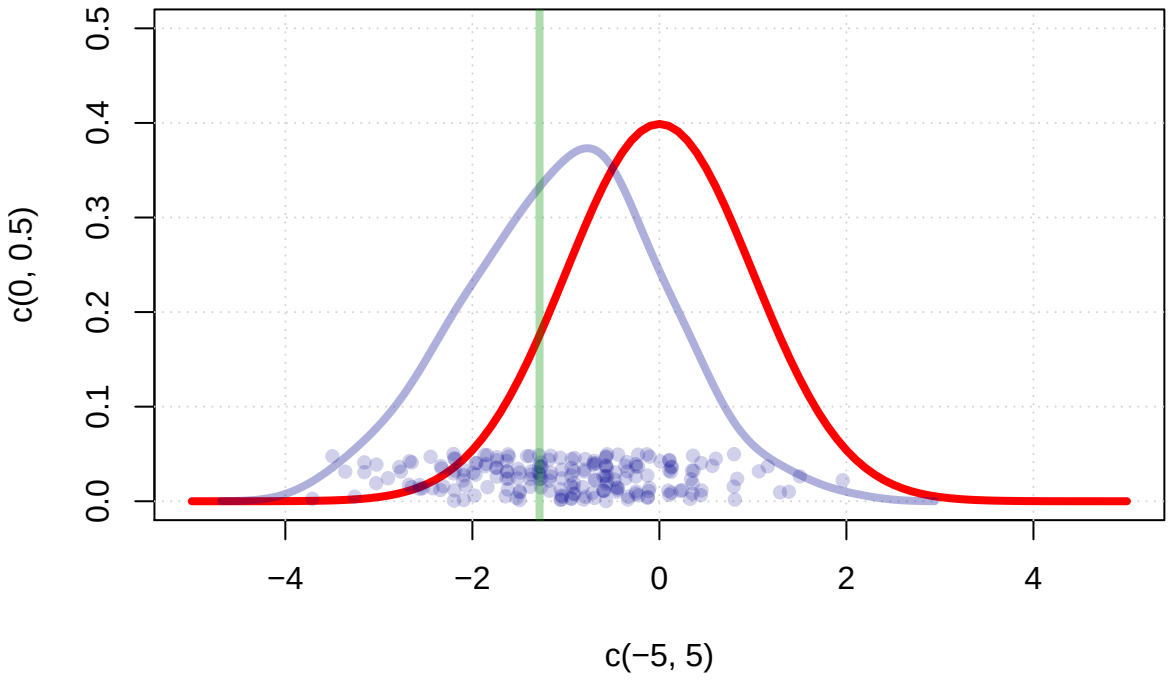

```r
# Experimente, um Fehler zweiter Art zu zeigen
mu.0 = 0.55      # Nullhypothese stimmt jetzt gar nicht
Test.plot()
```



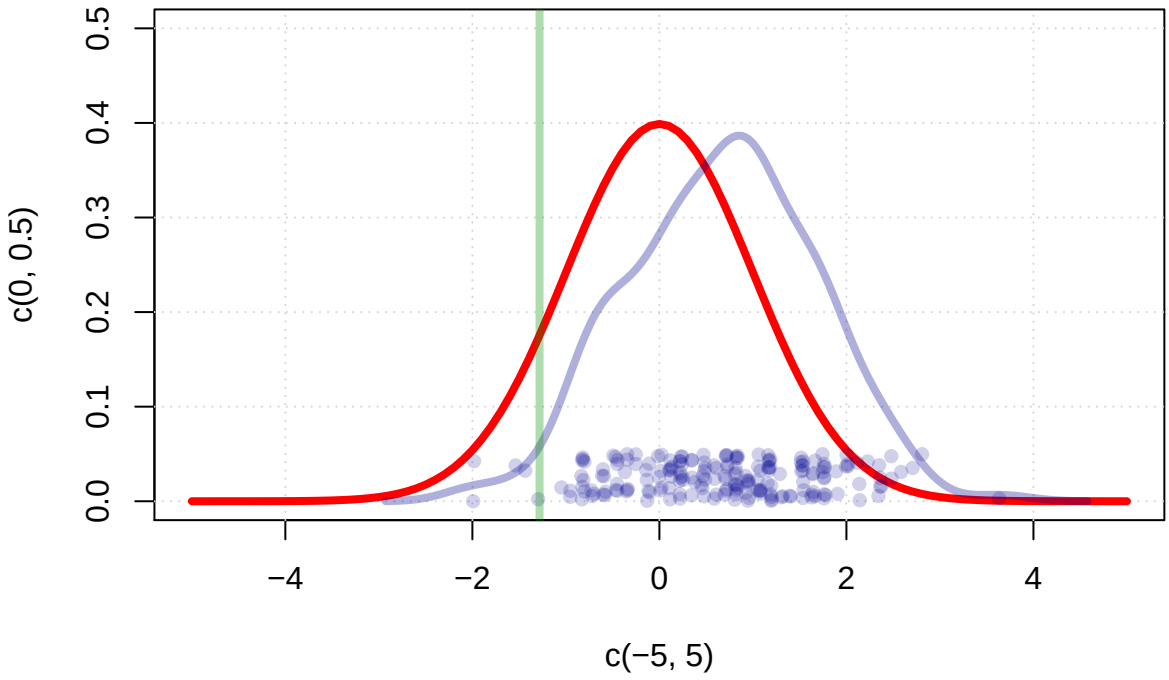

```
mu.0 = 0.50       # Nullhypothese stimmt jetzt nicht
Test.plot()
```




```
mu.0 = 0.45       # Nullhypothese stimmt jetzt (ein bisschen) nicht
Test.plot()
```




```
mu.0 = 0.40       # Nullhypothese stimmt jetzt fast
Test.plot()
```




```
# Test auf Varianz
# H0: Standardabweichung der Väter-Längen ist 10cm
# H1: das ist nicht so
#
Groesse = na.omit(D$GroesseV)
sigma = sd(Groesse)
sigma.0 = 4
n = 20 # Stichprobenumfang


# (1)
alpha = 0.05

# (2)
B.links = qchisq(p=alpha/2, df=n-1)
B.rechts = qchisq(p=1-alpha/2, df=n-1)
cat("Verwerfungsbereich: B=(-unendlich; ", B.links, ") oder (", B.rechts, "; +unendlich)", sep="")
```

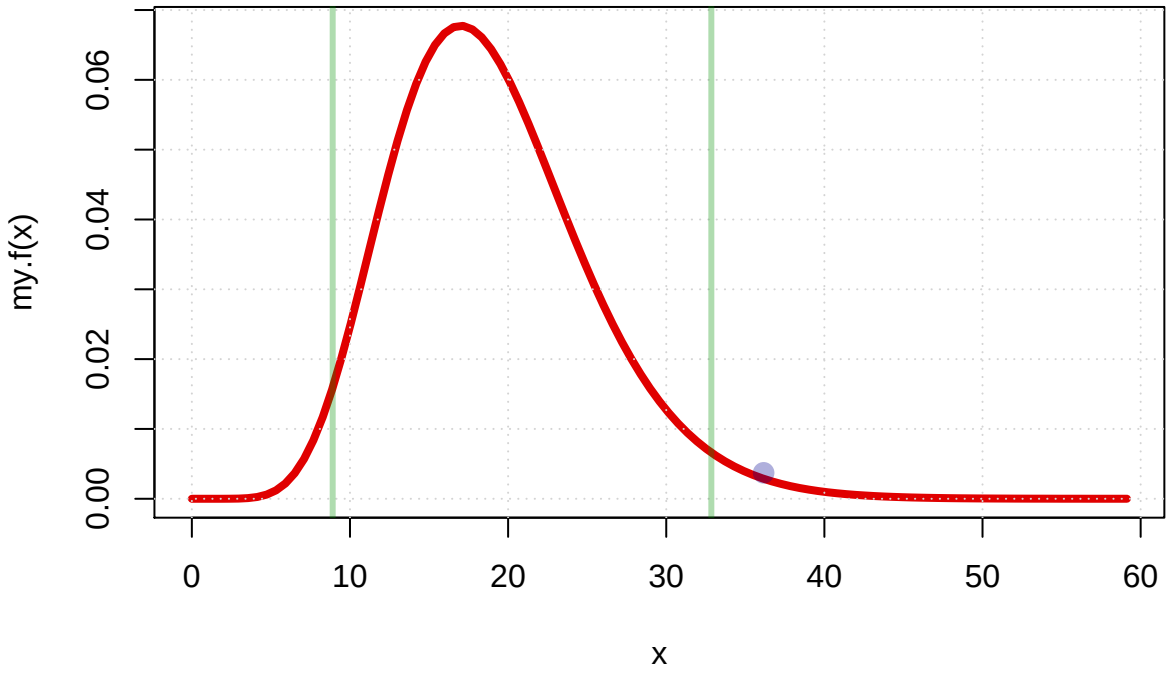
```
## Verwerfungsbereich: B=(-unendlich; 8.906516) oder (32.85233; +unendlich)
```

```
my.f = function(x){dchisq(x, df=n-1)}
curve(my.f, from=0, to=B.rechts*1.8, lwd=4, col="#e00000")
grid()
abline(v = c(B.links, B.rechts), col="#00900050", lwd=3)
```

```r
# (3)
Groesse.Stichprobe = sample(Groesse, size = n, replace = T)
v = var(Groesse.Stichprobe)*(n-1)/sigma.0^2
v
```

```
## [1] 36.15937
```

```r
points(v,runif(1,min = 0, max = 0.005),cex=1.5,col="#00009050", pch=16)
```




```r
# Kontingenztest
```

```r
alpha=0.05
n=50
D = na.omit(D)
set.seed(2)
i = sample.int(length(D[,1]), size = n, replace = T)
# Test, ob Größe der Väter und Note unabhängig (H0)

G = D[i,"GroesseV"]
Groesse = cut(G, breaks=c(min(G), median(G), max(G)), include.lowest = T)

N = D[i,"NoteMathe"]
Note = cut(N, breaks=c(1,2,3,4,5),  include.lowest = T)
```

```r
Tabelle = table(Groesse, Note)

# install.packages("gmodels")
library(gmodels)
Kontingenztabelle = CrossTable(Groesse, Note,
           expected=TRUE,
           prop.t=FALSE, prop.c=FALSE, prop.r=FALSE)
```

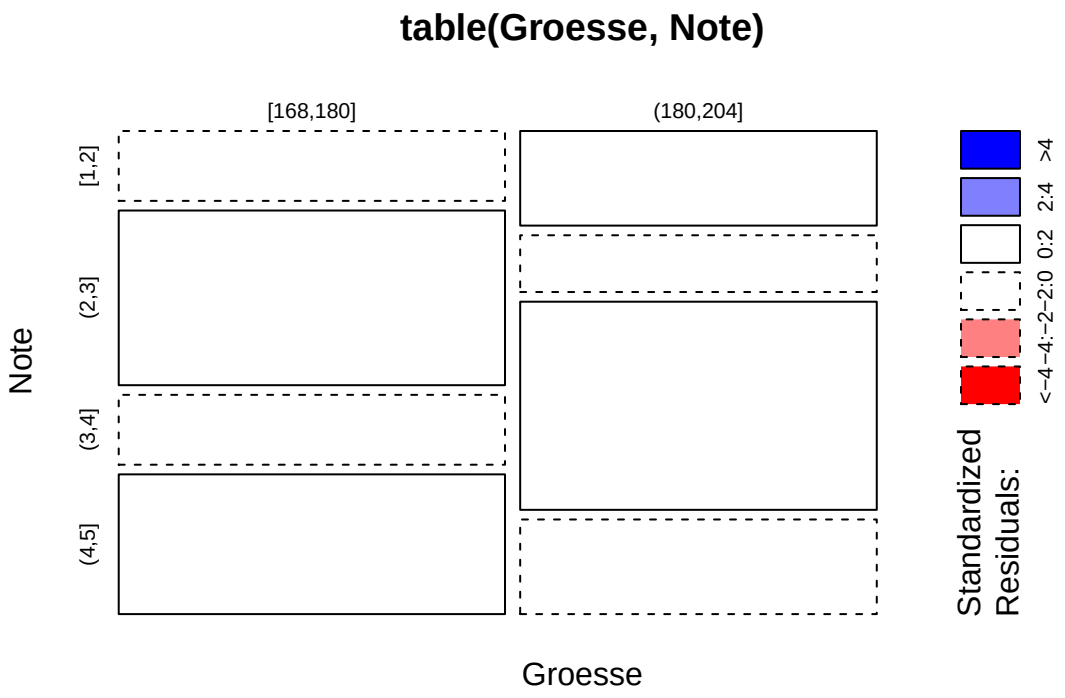
```
## Warning in chisq.test(t, correct = FALSE, ...): Chi-squared approximation
## may be incorrect
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |              Expected N |
## | Chi-square contribution |
## |-------------------------|
##
##
## Total Observations in Table:  50
##
##
##              | Note
##      Groesse |     [1,2] |     (2,3] |     (3,4] |     (4,5] | Row Total |
## -------------|-----------|-----------|-----------|-----------|-----------|
##    [168,180] |         4 |        10 |         4 |         8 |        26 |
##              |     4.680 |     6.760 |     7.800 |     6.760 |           |
##              |     0.099 |     1.553 |     1.851 |     0.227 |           |
## -------------|-----------|-----------|-----------|-----------|-----------|
##    (180,204] |         5 |         3 |        11 |         5 |        24 |
##              |     4.320 |     6.240 |     7.200 |     6.240 |           |
##              |     0.107 |     1.682 |     2.006 |     0.246 |           |
## -------------|-----------|-----------|-----------|-----------|-----------|
## Column Total |         9 |        13 |        15 |        13 |        50 |
## -------------|-----------|-----------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  7.771751     d.f. =  3     p =  0.05097214
##
##
##
```

```r
v = Kontingenztabelle$chisq
B = qchisq(p=1-alpha, df=3)
c(v, B)
```

```
## $statistic
## X-squared
##  7.771751
##
## $parameter
## df
##  3
##
## $p.value
## [1] 0.05097214
##
## $method
## [1] "Pearson's Chi-squared test"
##
## $data.name
## [1] "t"
##
## $observed
##           y
## x          [1,2] (2,3] (3,4] (4,5]
##   [168,180]     4    10     4     8
##   (180,204]     5     3    11     5
##
## $expected
##           y
## x          [1,2] (2,3] (3,4] (4,5]
##   [168,180]  4.68  6.76   7.8  6.76
##   (180,204]  4.32  6.24   7.2  6.24
##
## $residuals
##           y
## x                [1,2]      (2,3]      (3,4]      (4,5]
##   [168,180] -0.3143301  1.2461538 -1.3606183  0.4769231
##   (180,204]  0.3271652 -1.2970380  1.4161764 -0.4963973
##
## $stdres
##           y
## x                [1,2]      (2,3]      (3,4]      (4,5]
##   [168,180] -0.5010239  2.0909092 -2.3472895  0.8002245
##   (180,204]  0.5010239 -2.0909092  2.3472895 -0.8002245
##
## [[10]]
## [1] 7.814728
```

```
mosaicplot(table(Groesse, Note), shade = T)
```


table(Groesse, Note)
[168,180]
(180,204]
[1,2]
(2,3]
(3,4]
(4,5]
Note
Groesse
Standardized Residuals:
<−4
−4:−2
−2:0
0:2
2:4
>4

```
#
```