Aufgabenblatt 2

UK Angewandte Statistik

Cordula Eggerth

Matrikelnummer: 00750881

Kursleiter: Prof. Dr. Wilfried Grossmann

Wintersemester 2018

Aufgabe 1: (t-Test zum Vergleich von Stichproben)

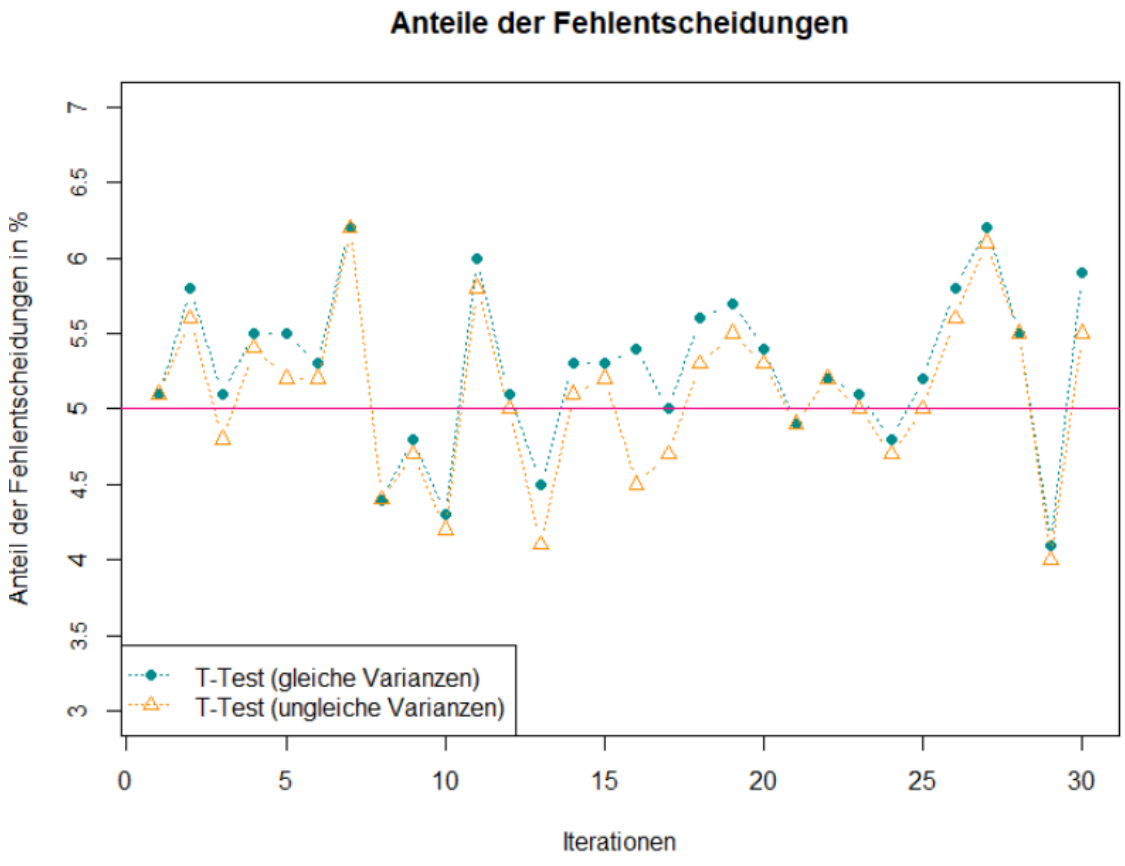
Es werden zunächst 1000 Stichprobenpaare (X1, X2) jeweils vom Umfang 30 erzeugt. Die Zufallsvariable (ZV) X1 sind normalverteilt (NV) mit Mittelwert 0 und Varianz 1, und X2 mit Mittelwert 1 und Varianz 4. Für jedes Paar wird die Hypothese, dass die Mittelwerte der beiden ZV gleich sind mittels t-Test (unter der Annahme von gleichen bzw. ungleichen Varianzen) untersucht auf Signifikanzniveau (alpha) 5%.

Die Anteile der Simulationen, die zur Ablehnung der H0 führen (d.h. die empirische Wahrscheinlichkeit des Fehlers 1. Art), wobei hier anteile_ttest1 sich auf die Fälle der t-Tests mit gleichen Varianzen und anteile_ttest2 sich auf jene mit ungleichen Varianzen bezieht:

```
> anteile_ttest1
 [1] 0.046 0.037 0.053 0.047 0.053 0.047 0.053 0.057 0.046 0.063 0.056 0.052 0.054
 0.041 0.050 0.045
[17] 0.056 0.041 0.043 0.063 0.043 0.053 0.058 0.058 0.046 0.064 0.054 0.041 0.046
 0.052
> anteile_ttest2
 [1] 0.046 0.037 0.052 0.046 0.052 0.044 0.053 0.057 0.045 0.062 0.051 0.051 0.054
 0.039 0.050 0.044
[17] 0.056 0.039 0.043 0.063 0.040 0.052 0.057 0.056 0.045 0.062 0.050 0.040 0.045
 0.051
```

(siehe detaillierte Berechnung im Bereich „Code zu Aufgabe 1“)

Plottet man nun eine Simulation von 30 Durchläufe für die jeweiligen Anteilswerte der verschiedenen t-Tests, ergibt sich folgendes Bild:



Im Plot ist das 5%-Signifikanzniveau als horizontale Linie eingezeichnet. Es ist zu erkennen, dass in einigen Fällen die türkisen Punkte, die für die im Fall der t-Tests mit gleichen Varianzen berechneten Anteile stehen, und in anderen Fällen die gelben Punkte (i.e. t-Tests mit ungleichen Varianzen) näher an der 5%-Marke liegen. Angemerkt sei, dass es sich hier um diskrete Events (bzw. Punkte) handelt, aber die gestrichelte Linie wurde zur besseren Erkennbarkeit der Unterschiede zwischen den untersuchten t-Tests eingezeichnet. Es entspricht also jener Anteil eines der t-Tests, der näher an der 5%-Marke liegt, besser dem Signifikanzniveau von 5%.

R-Code zu Aufgabe 1:

```
anteile_H0ablehnen <- function(){

    # ergebnisvektoren und alpha anlegen:
    ergebnis_Ttest_gleicheVarianzen <- rep(0,1000)
    ergebnis_Ttest_ungleicheVarianzen <- rep(0,1000)
    alpha <- 0.05

    # für 1000 stichprobenpaare den welch-two-sample-t-test machen:
    for(j in 1:1000){
      # stichprobenpaar der j-ten iteration generieren
      stichprobe_x1 <- rnorm(30, sd=1)
      stichprobe_x2 <- rnorm(30, sd=2) # geg. war varianz=4, also sd=2

      # welch-two-sample-t-test durchführen auf stichprobenpaar
      ttest_gleicheVarianzen <- t.test(stichprobe_x1, stichprobe_x2, var.equal=TRUE)
      ttest_ungleicheVarianzen <- t.test(stichprobe_x1, stichprobe_x2, var.equal=FALSE)

      # p-values den ergebnisvektoren zuweisen
      ergebnis_Ttest_gleicheVarianzen[j] <- ttest_gleicheVarianzen$p.value
      ergebnis_Ttest_ungleicheVarianzen[j] <- ttest_ungleicheVarianzen$p.value
    }

    # empirische wahrscheinlichkeit des fehlers 1. art bzw. anteil der
    # simulationen, die zur ablehnung von H0 führen:
    empWahrsch_gleicheVarianzen <- sum(ergebnis_Ttest_gleicheVarianzen < alpha)/
                                   length(ergebnis_Ttest_gleicheVarianzen)

    empWahrsch_ungleicheVarianzen <- sum(ergebnis_Ttest_ungleicheVarianzen < alpha)/
                                     length(ergebnis_Ttest_ungleicheVarianzen)

    # return
    list=list(ttest_gleicheVar=empWahrsch_gleicheVarianzen,
              ttest_ungleicheVar=empWahrsch_ungleicheVarianzen)

}


# berechne anteile an fehlentscheidungen für n durchläufe:
n <- 30
anteile_ttest1 <- rep(0,n)
anteile_ttest2 <- rep(0,n)

for(i in 1:n){
  anteile <- anteile_H0ablehnen()
  anteile_ttest1[i] <- anteile$ttest_gleicheVar
  anteile_ttest2[i] <- anteile$ttest_ungleicheVar
}

# plot anteile an fehlentscheidungen pro test
plot(1:30, anteile_ttest1*100, ylim=c(3,7), xlab="Iterationen",
     ylab="Anteil der Fehlentscheidungen in %", type="b",
     col="darkcyan", pch=16, main="Anteile der Fehlentscheidungen",
     lty="dotted")
axis(side=2, at=c(0.5:8), cex.axis=0.8)
lines(anteile_ttest2*100, col="darkorange", type="b", pch=2,
      lty="dotted")
abline(h=5, col="deeppink2")
legend("bottomleft",legend=c("T-Test (gleiche Varianzen)", "T-Test (ungleiche Varianzen)"),
       col = c("darkcyan", "darkorange"),
       border = "black", lty="dotted", lwd=1, pch=c(16,2))
```

Aufgabe 2 (t-Test vs. Permutationstest):

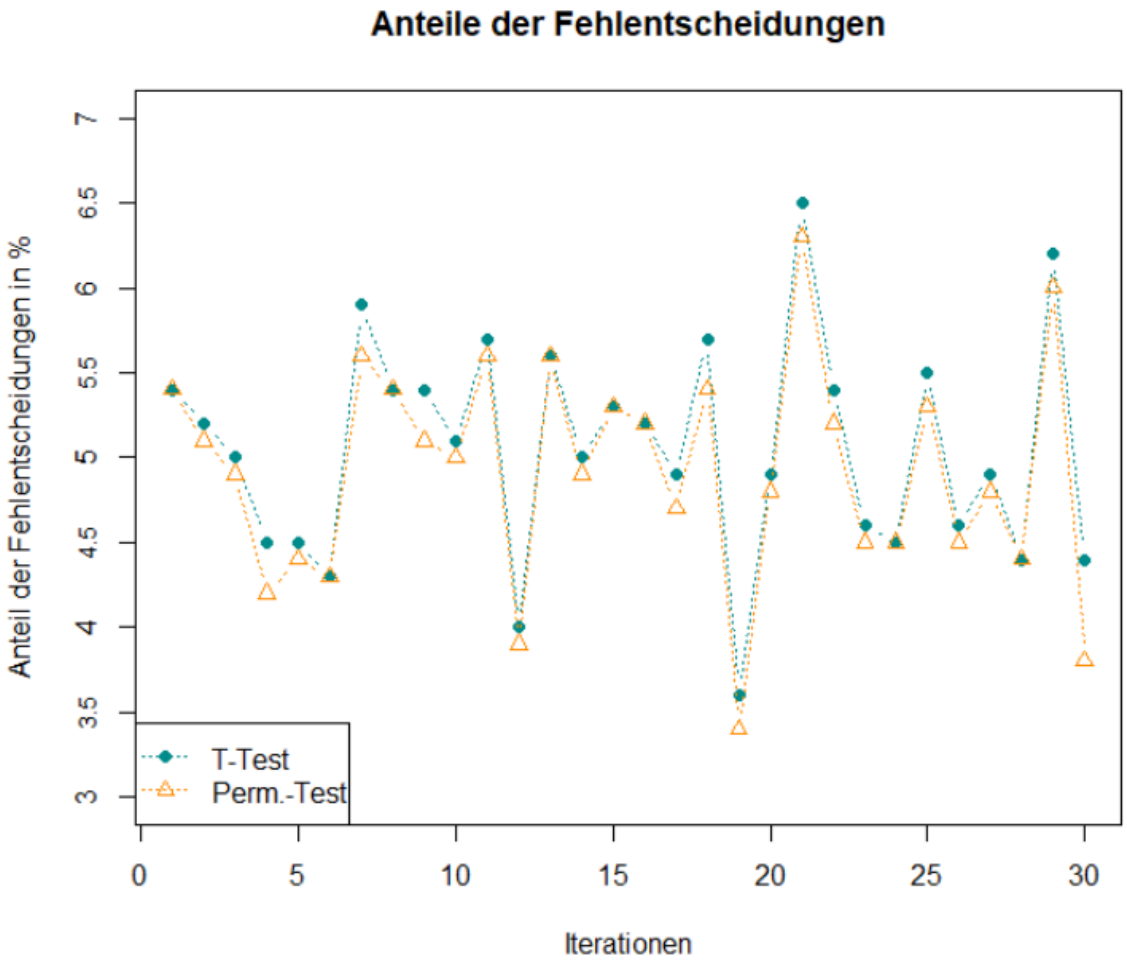
Es werden 1000 Stichprobenpaare (Y1, Y2) vom Umfang 20 aus den natürlichen Zahlen von 1 bis 10 mit der Funktion sample mit Zurücklegen gezogen. Dann testet man die Hypothese auf Mittelwertsgleichheit mit t-Test und Permutationstest.

Die Anteile der Fehlentscheidungen sind für 30 Durchgänge:

```
> anteile_ttest
 [1] 0.054 0.052 0.050 0.045 0.045 0.043 0.059 0.054 0.054 0.051 0.057 0.040
0.056 0.050 0.053 0.052
[17] 0.049 0.057 0.036 0.049 0.065 0.054 0.046 0.045 0.055 0.046 0.049 0.044
0.062 0.044
> anteile_ptest
 [1] 0.054 0.051 0.049 0.042 0.044 0.043 0.056 0.054 0.051 0.050 0.056 0.039
0.056 0.049 0.053 0.052
[17] 0.047 0.054 0.034 0.048 0.063 0.052 0.045 0.045 0.053 0.045 0.048 0.044
0.060 0.038
```

(siehe detaillierte Berechnung im Bereich „Code zu Aufgabe 2“)

Plottet man nun eine Simulation von 30 Durchläufen für die jeweiligen Anteilswerte der verschiedenen t-Tests und Permutationstests, ergibt sich folgendes Bild:



Es ist zu erkennen, dass in einigen Fällen die türkisen Punkte, die für die im Fall der t-Tests berechneten Anteile stehen, und in anderen Fällen die gelben Punkte (i.e. gemäß Permutationstests) näher an der 5%-Marke liegen. Angemerkt sei, dass es sich hier um diskrete Events (bzw. Punkte) handelt, aber die gestrichelte Linie wurde zur besseren Erkennbarkeit der Unterschiede zwischen den untersuchten t-Tests bzw. Permutationstests eingezeichnet. Es entspricht also jener Anteil eines der t-Tests bzw. Permutationstests, der näher an der 5%-Marke liegt, besser dem Signifikanzniveau von 5%.

R-Code zu Aufgabe 2:

```
anteile_fehlentscheidungen <- function(){

  # ergebnisvektoren und alpha anlegen:
  ergebnis_Ttest <- rep(0,1000)
  ergebnis_PERMtest <- rep(0,1000)
  alpha <- 0.05


  # T-TEST & PERM-TEST:
  # (siehe Quelle https://cran.r-project.org/web/packages/perm/perm.pdf
  # für den Permutationstest)
  for(j in 1:1000){
    # stichprobenpaar der j-ten iteration generieren
    stichprobe_y1 <- sample(1:10, 20, replace = TRUE)
    stichprobe_y2 <- sample(1:10, 20, replace = TRUE)

    # t-test durchführen auf stichprobenpaar
    ttest <- t.test(stichprobe_y1, stichprobe_y2, var.equal=TRUE)
    # permutationstest durchführen auf stichprobenpaar
    permtest <- permTS(stichprobe_y1, stichprobe_y2, alternative="two.sided")

    # p-values den ergebnisvektoren zuweisen
    ergebnis_Ttest[j] <- ttest$p.value
    ergebnis_PERMtest[j] <- permtest$p.value
  }

  # empirische wahrscheinlichkeit des fehlers 1. art bzw. anteil der
  # simulationen, die zur ablehnung von H0 führen:
  empWahrsch_Ttest <- sum(ergebnis_Ttest < alpha)/ length(ergebnis_Ttest)
  empWahrsch_PERMtest <- sum(ergebnis_PERMtest < alpha)/ length(ergebnis_PERMtest)

  # return
  list=list(ttest_pval=empWahrsch_Ttest, ptest_pval=empWahrsch_PERMtest)
}


# berechne anteile an fehlentscheidungen für n_iterations durchläufe:
n_iterations <- 30
anteile_ttest <- rep(0,n_iterations)
anteile_ptest <- rep(0,n_iterations)

for(i in 1:n_iterations){
  anteile <- anteile_fehlentscheidungen()
  anteile_ttest[i] <- anteile$'ttest_pval'
  anteile_ptest[i] <- anteile$ptest_pval
}

# plot anteile an fehlentscheidungen pro test
plot(1:30, anteile_ttest*100, ylim=c(3,7), xlab="Iterationen",
     ylab="Anteil der Fehlentscheidungen in %", type="b",
     col="darkcyan", pch=16, main="Anteile der Fehlentscheidungen",
     lty="dotted")
axis(side=2, at=c(0.5:8), cex.axis=0.8)
lines(anteile_ptest*100, col="darkorange", type="b", pch=2,
      lty="dotted")
legend("bottomleft",legend=c("T-Test", "Perm.-Test"),
       col = c("darkcyan", "darkorange"),
       border = "black", lty="dotted", lwd=1, pch=c(16,2))
```

Aufgabe 3 (Maximum-Likelihood-Schätzer Gammaverteilung):

Die Dichte einer Gammaverteilung Gam(v,sigma) ist gegeben (für x>=0):

$$f(x;v,\sigma) = x^{(v-1)} / (\sigma^v * \Gamma(v)) * e^{(-(x/\sigma))} .$$

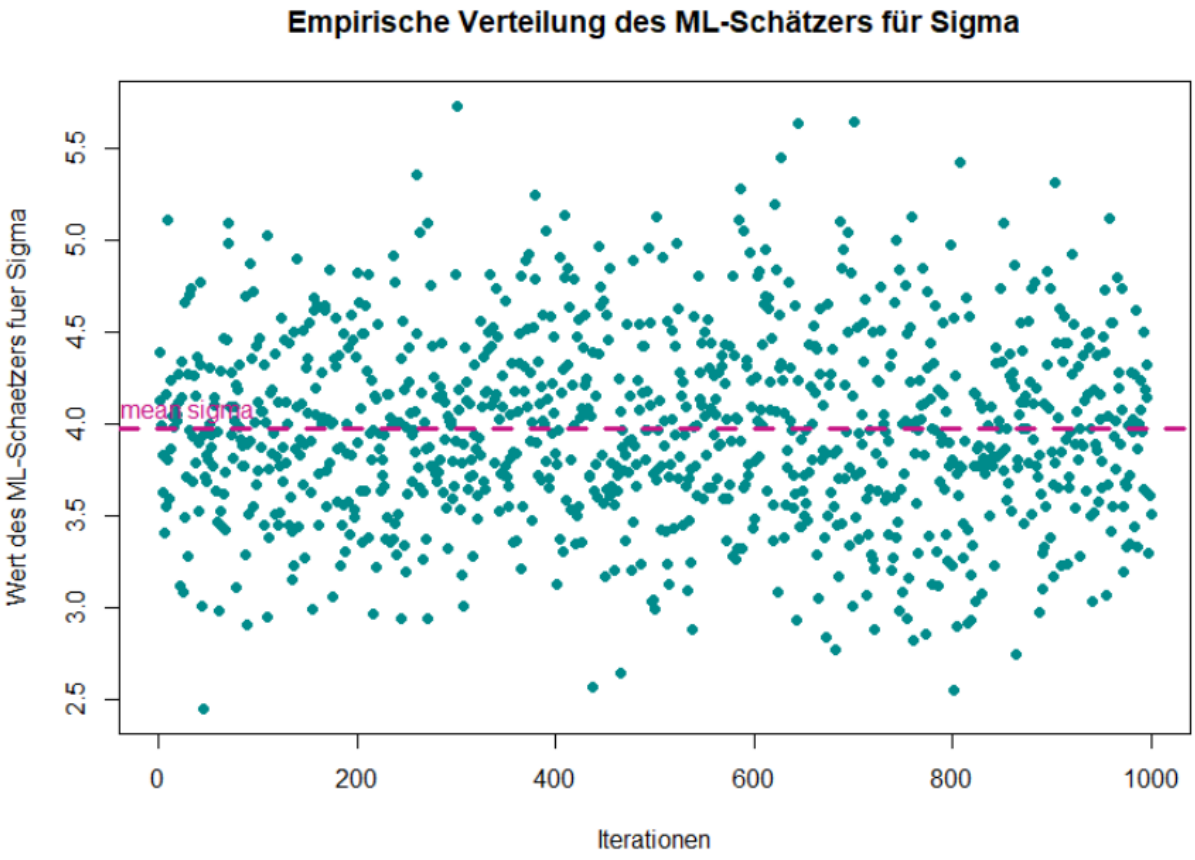
Bei bekanntem Parameter v und n iid Beobachtungen ist der Maximum-Likelihood-Schätzer (ML) für Parameter sigma gegeben als:

$$\hat{\sigma} = \bar{x}/v = 1/n*\sum(x_i) / v .$$

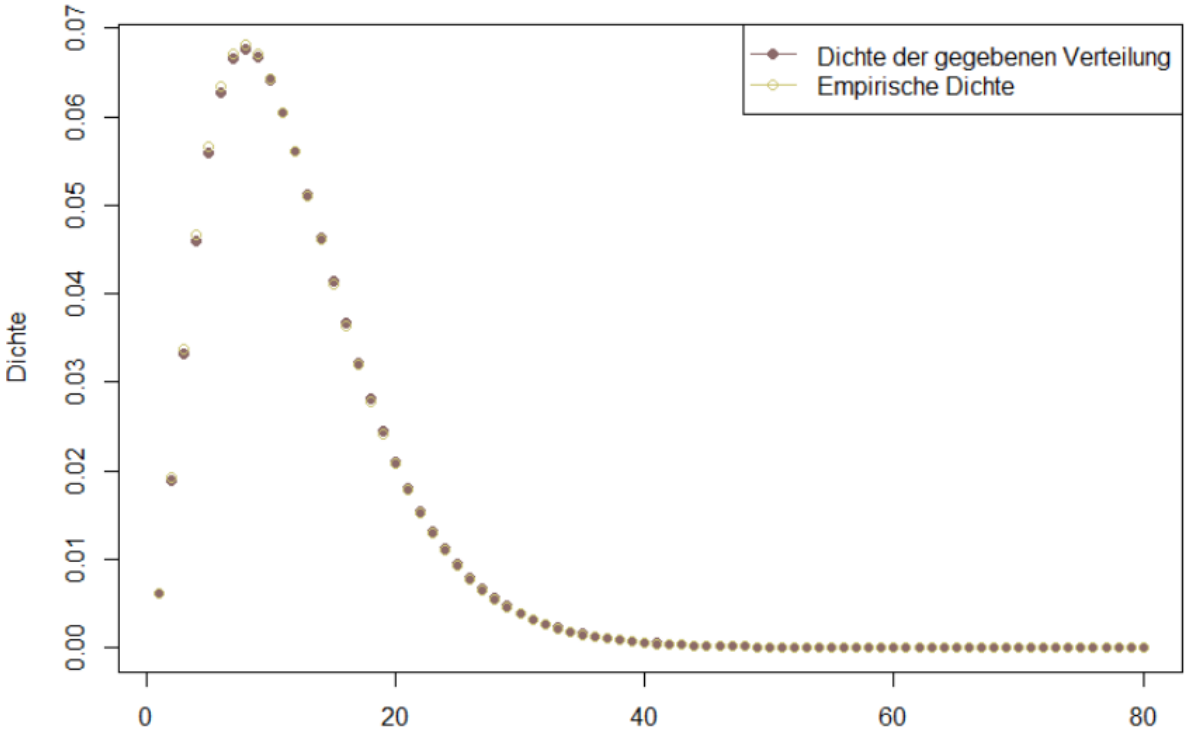
Es werden 1000 Zufallsstichproben vom Umfang 20 nach einer Gam(3,4)-Verteilung generiert.

Der ML-Schätzer für Sigma wird aus den Daten bestimmt und die empirische Verteilung wird geplottet:

```
> ergebnis_sigmahat_ML
   [1] 4.130300 4.386347 3.994356 3.832273 3.625615 3.409414 3.552504 4.156719 5.107963
  [10] 3.806632 3.589206 3.866946 4.235629 4.041478 3.985979 4.052268 4.094387 4.013558
  [19] 4.050771 4.117565 4.268986 3.121151 4.140667 4.340940 3.084435 3.492892 4.659374
  [28] 3.708202 3.282854 4.273151 4.704452 3.967253 4.734327 3.931815 3.690140 4.259268
  [37] 3.790038 4.150880 4.362532 3.526323 3.897123 4.318599 4.771078 3.008998 2.449776
```



Vergleicht man nun die Dichte der gegebenen Gammaverteilung mit der empirischen Dichte (unter Verwendung des ML-Schätzers für den Parameter sigma), ergibt sich folgender Plot, in dem die beiden sich fast entsprechen:



R-Code zu Aufgabe 3:

```
# ergebnisvektor anlegen:
ergebnis_sigmahat_ML <- rep(0,1000)
rate <- 0.25
shape_v <- 3
scale_sigma <- 4
stichprobenumfang <- 20
stichprobenmatrix <- matrix(0, nrow=1000, ncol=20)

# generiere 1000 gammaverteilte zufallsstichproben und berechne ML-schaetzer
# fuer sigma:
for(j in 1:1000){
  # zufallsstichprobe der j-ten iteration generieren
  stichprobenmatrix[j, ] <- rgamma(n=stichprobenumfang, shape=shape_v, scale=scale_sigma)
  # berechne sigma_hat aus den daten
  ergebnis_sigmahat_ML[j] <- (1/stichprobenumfang)*sum(stichprobenmatrix[j, ])/shape_v
}

# plot empirische verteilung des ML-schaetzers fuer sigma (d.h. sigmahat_ML):
plot(1:1000, ergebnis_sigmahat_ML, xlab="Iterationen",
     ylab="Wert des ML-Schaetzers fuer Sigma", type="p",
     col="darkcyan", pch=16, main="Empirische Verteilung des ML-Schätzers für Sigma",
     lty="dotted")
abline(h=mean(ergebnis_sigmahat_ML), col="mediumvioletred", pch=2,
       lty="dashed", lwd=3)
text(30, mean(ergebnis_sigmahat_ML)+0.1, "mean sigma", col="mediumvioletred")

# empirische dichte des ML-schaetzers:
m <- 80
seq <- 1:m
f_empirisch <- rep(0,m)
for(i in 1:m){
  f_empirisch[i] <- seq[i]^(shape_v-1) / (mean(ergebnis_sigmahat_ML)^shape_v *
                    gamma(shape_v)) * exp(-(seq[i]/mean(ergebnis_sigmahat_ML)))
}


# verteilung der daten:
f_verteilung_der_daten <- dgamma(1:m, shape=shape_v, scale=4)

# graphischer vergleich der ergebnisse:
plot(f_verteilung_der_daten, ylab="Dichte", col="rosybrown4", pch=19, xlab="")
points(f_empirisch, col="khaki3")
legend("topright",legend=c("Dichte der gegebenen Verteilung",
       "Empirische Dichte"), col = c("rosybrown4", "khaki3"),
       border = "black", lwd=1, pch=c(19,1))
```

Aufgabe 4 (Bayes-Schätzer für Mittelwert einer Normalverteilung):

Man geht aus von (x1, … , xn) unabhängig N(0,1)-verteilten Beobachtungen. Die Prior-Verteilung des Parameters mu ist durch eine N(x0, 1/n0) definiert. Der Bayes-Schätzer für den Mittelwert ist der Poster-Mittelwert:

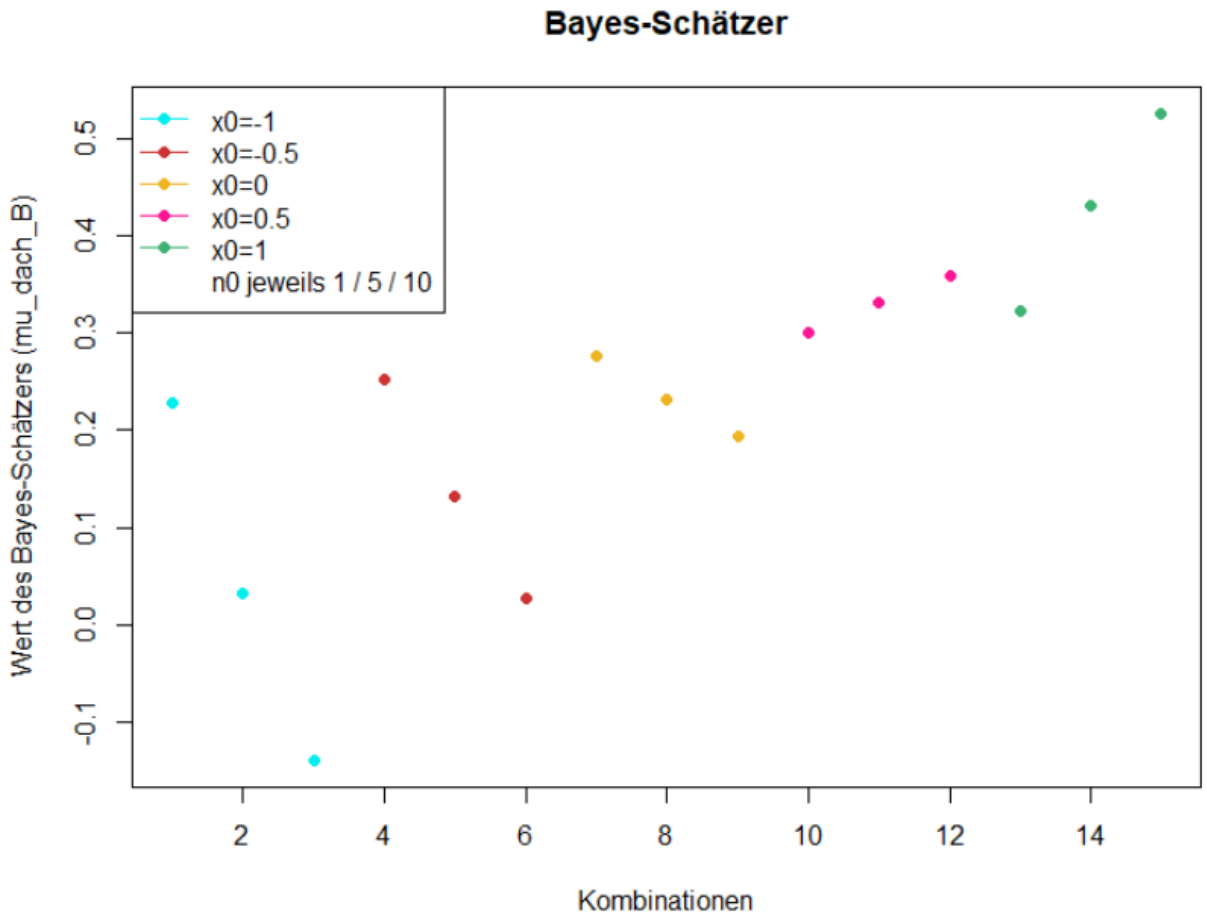
$$mu\_dach\_B = (x0*n0 + sum\_1bisn(x\_i)) / (n+n0) .$$

Die Verteilung des Bayes-Schätzer ist eine N(mu_dach_B,1/(n+n0)).

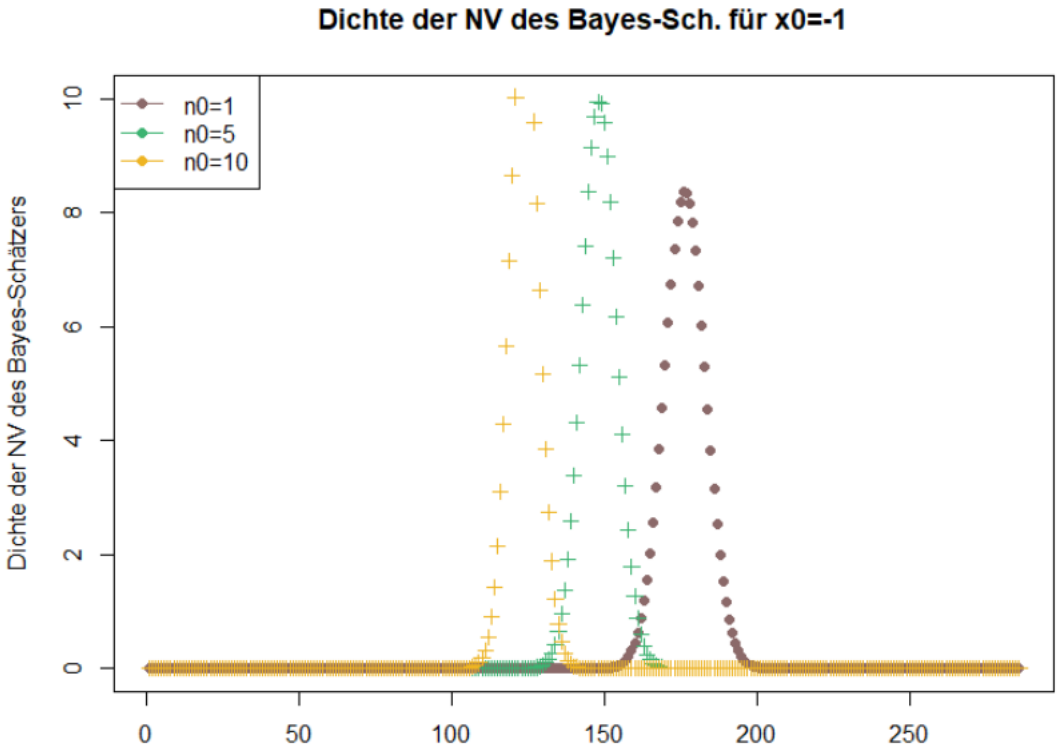
Für verschiedene Kombinationen der Annahmen aus der Angabe werden folgende Ausprägungen des Bayes-Schätzers ermittelt:

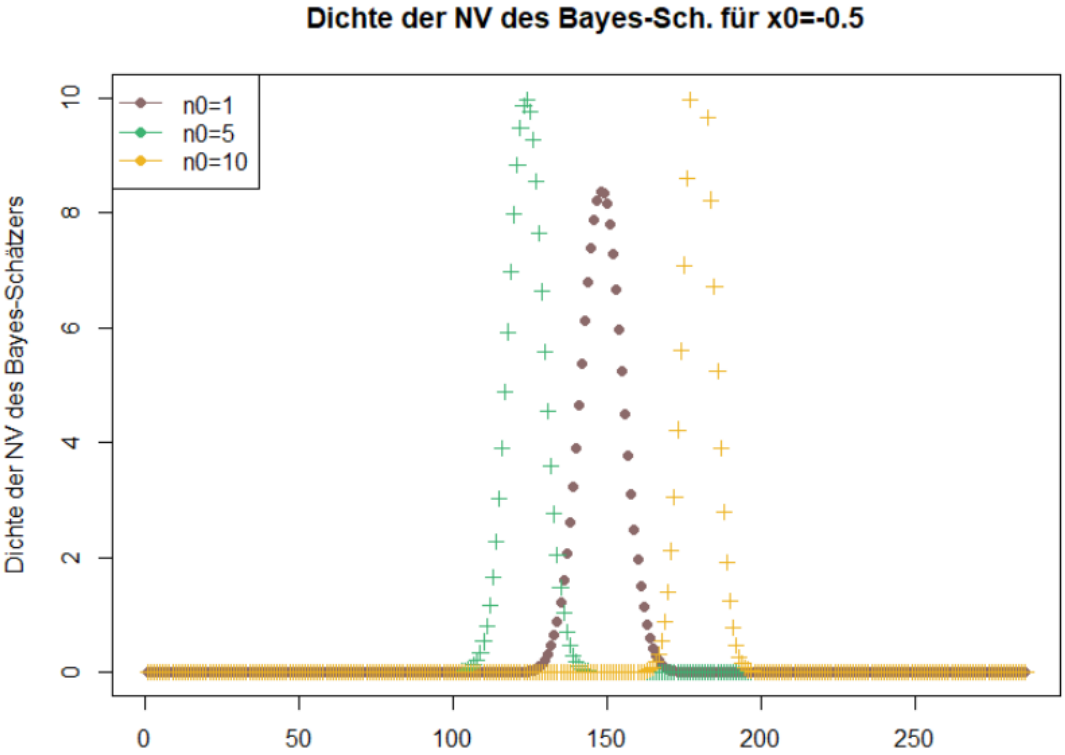
```
> mu_dach_B_vektor
   x0=-1, n0=1    x0=-1, n0=5   x0=-1, n0=10  x0=-0.5, n0=1  x0=-0.5, n0=5 x0=-0.5, n0=10
    0.22830990     0.03178032    -0.14018307     0.25211942     0.13178032     0.02648360
    x0=0, n0=1     x0=0, n0=5    x0=0, n0=10   x0=0.5, n0=1   x0=0.5, n0=5  x0=0.5, n0=10
    0.27592895     0.23178032     0.19315026     0.29973847     0.33178032     0.35981693
    x0=1, n0=1     x0=1, n0=5    x0=1, n0=10
    0.32354799     0.43178032     0.52648360
```

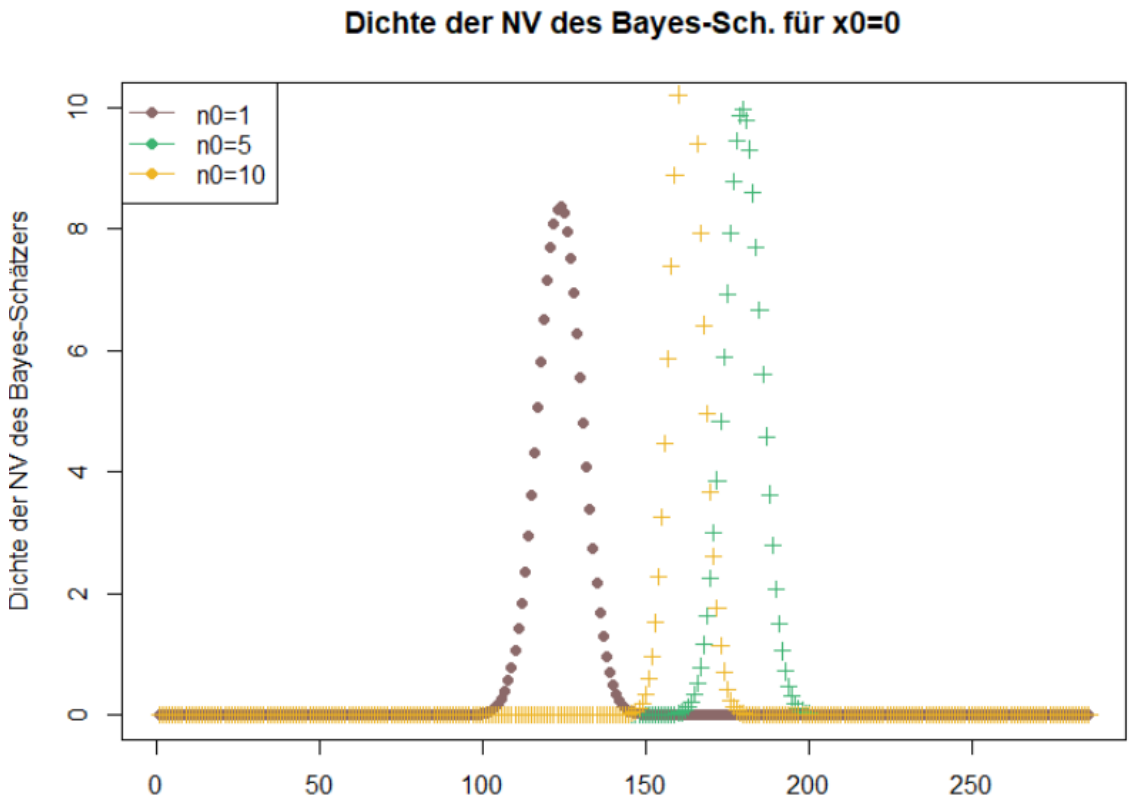
Plottet man nun diese Kombinationen beginnend mit x0=-1 und darin die jeweiligen n0-Werte, ergibt sich die folgende Grafik:

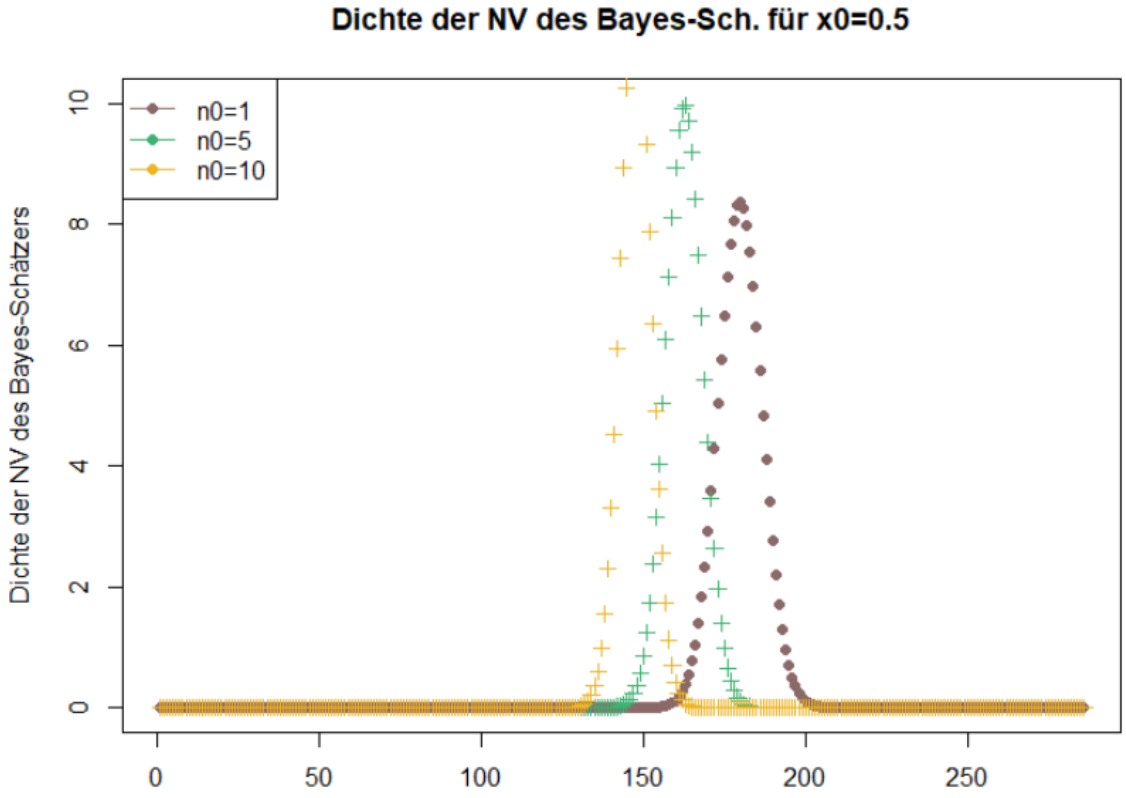


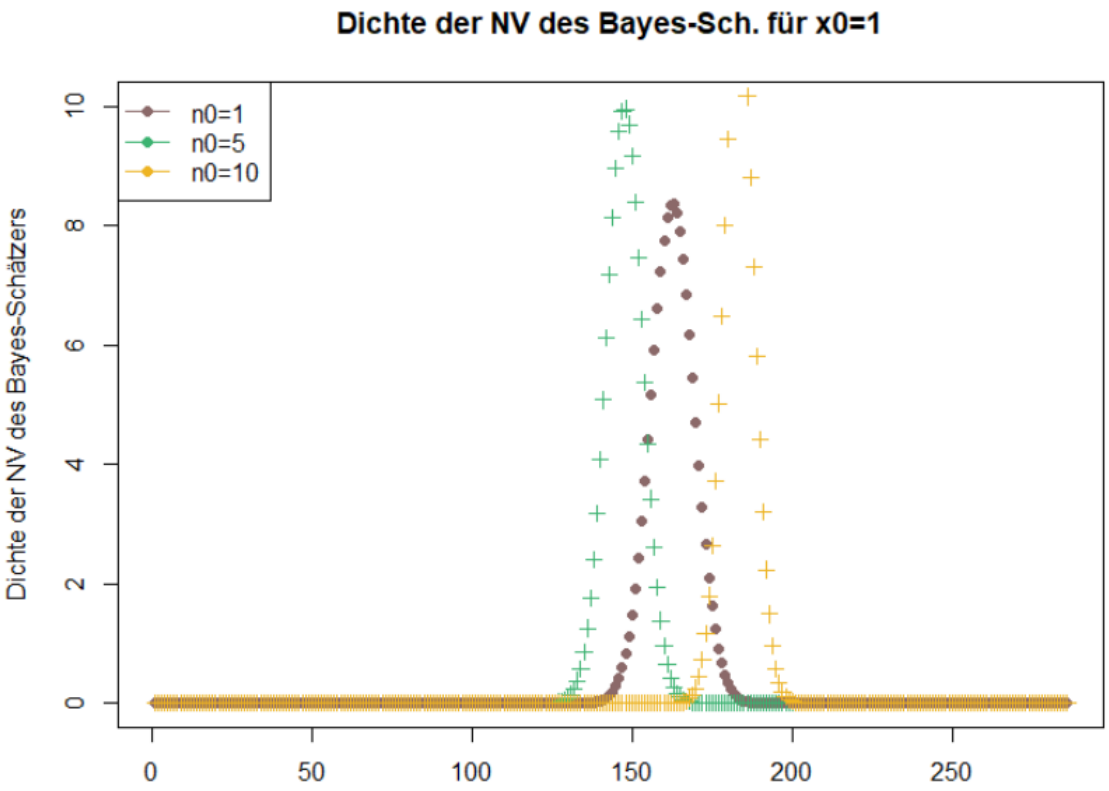
Die Dichte der Normalverteilnug des Bayes-Schätzers kann ebenfalls für die möglichen Kombinationen der Werte aus der Angabe für x0 und n0 grafisch wie untenstehend dargestellt werden:






Dichte der NV des Bayes-Sch. für x0=0
n0=1
n0=5
n0=10
Dichte der NV des Bayes-Schätzers
0
2
4
6
8
10
0
50
100
150
200
250


Dichte der NV des Bayes-Sch. für x0=0.5
n0=1
n0=5
n0=10
Dichte der NV des Bayes-Schätzers
0
2
4
6
8
10
0
50
100
150
200
250


Dichte der NV des Bayes-Sch. für x0=1
n0=1
n0=5
n0=10
Dichte der NV des Bayes-Schätzers
0
2
4
6
8
10
0
50
100
150
200
250

R-Code zu Aufgabe 4:

```
# annahmen für priorverteilungen und stichprobenumfang:
x0 <- c(-1, -0.5, 0, 0.5, 1)
n0 <- c(1, 5, 10)
n <- 20

# berechne poster-mittelwert (mu_dach_B) für versch. kombinationen
# von annahmen:
anzahl_kombis <- length(x0)*length(n0)
mu_dach_B_vektor <- rep(0,anzahl_kombis)
iter <- 1
beobachtungen <- rnorm(n=20, mean=0, sd=1)

for(i in 1:length(x0)){
  for(j in 1:length(n0)){
    mu_dach_B_vektor[iter] <- (x0[i]*n0[j] + sum(beobachtungen)) / (n+n0[j])
    names(mu_dach_B_vektor)[iter] <- paste0("x0=",x0[i],", n0=",n0[j])
    iter <- iter+1
  }
}

# ergebnis der kombis versch. annahmen für bayes-schätzer:
mu_dach_B_vektor

plot(mu_dach_B_vektor, xlab="Kombinationen",
     ylab="Wert des Bayes-Schätzers (mu_dach_B)", type="p",
     col=c("cyan2","cyan2", "cyan2", "brown3", "brown3", "brown3",
           "goldenrod2", "goldenrod2", "goldenrod2",
           "deeppink1", "deeppink1", "deeppink1",
           "mediumseagreen", "mediumseagreen", "mediumseagreen"),
     pch=16, main="Bayes-Schätzer",
     lty="dotted")
legend("topleft",legend=c("x0=-1", "x0=-0.5", "x0=0", "x0=0.5",
                          "x0=1", "n0 jeweils 1 / 5 / 10"),
       col = c("cyan2", "brown3", "goldenrod2", "deeppink1",
               "mediumseagreen", "white"),
       border = "black", lwd=1, pch=16)


# dichte der NV des bayes-schätzers (i.e. N(mu_dach_B,1/(n+n0))):
 # ausgangsdaten:
   mu_dach_B_vektor
   n0_vek <- rep(n0,5)


 # plot NV jeweils für mu_dach_B:
  for(i in 1:length(x0)){
      nv_bayes1 <- dnorm(x=seq(-1,1,0.007), mean = mu_dach_B_vektor[i], sd = 1/(n+n0[1]))
      nv_bayes2 <- dnorm(x=seq(-1,1,0.007), mean = mu_dach_B_vektor[i+1], sd = 1/(n+n0[2]))
      nv_bayes3 <- dnorm(x=seq(-1,1,0.007), mean = mu_dach_B_vektor[i+2], sd = 1/(n+n0[3]))

      plot(nv_bayes1, ylab="Dichte der NV des Bayes-Schätzers", col="rosybrown4",
           pch=19, xlab="", ylim=c(0,10), main=paste0("Dichte der NV des Bayes-Sch. für x0=",
           x0[i]))
      points(nv_bayes2, col="mediumseagreen", pch=3)
      points(nv_bayes3, col="goldenrod2", pch=3)

      legend("topleft",legend=c(paste0("n0=",n0[1]), paste0("n0=",n0[2]),
                                paste0("n0=",n0[3])),
             col = c("rosybrown4", "mediumseagreen", "goldenrod2"),
             border = "black", lwd=1, pch=16)
      i <- i*2
    }
```