

# HTML Arbeitsblatt 3 - Notengebung zu Punkten

Bert Niehaus

2025-05-23

## Einleitung

In dem folgende R-Code werden mit Hilfe von 2 Funktionen zu einem Vektor mit Punkten  $\mathbf{punkte} \in \mathbb{R}^n$  und die Note automatisiert für alle Punkte berechnet. \*  $\mathbf{punkte} \in \mathbb{R}^n$  enthält die Punkte (z.B. aus einer Klausur), für die die Noten ermitteln sollen. \*  $\mathbf{bestgrenze} \in \mathbb{R}^+$  ist die Bestehensgrenze, \*  $\mathbf{maxpunkte} \in \mathbb{R}^+$  ist die maximale Anzahl der Punkte, die in einer Klausur erreicht werden kann. \* Mit der Funktion `notengrenzen_calc(pPunkteMax, pBestGrenze, pBestIndex, pNotenSkala)` kann die Notengrenzen berechnen, um die Noten zuordnen danach mit der Funktion `note_zuordnen()` dem Punktevektor zuordnen zu können. \*  $\mathbf{notenskala}$  enthält die Noten, die entsprechend der Notengrenzen den Punkten zugewiesen werden. Diese können Zahlenwert 1-6, Text “sehr gut”, “sehr gut-” ..., oder auch Buchstaben A,B,C,D, und F für “failed” (amerikanisches Notensystem) sein, um entsprechend der Notengrenzen die Noten zuordnen zu können.

## Notenberechnung

Die Notenberechnung erfolgt in zwei Schritten: \* **Notengrenzen:** Berechnung der Notengrenzen bei gegebener Maximalpunktzahl, Bestehensgrenze und den verwendeten Notenskala, \* **Notenzuordnung:** Dabei wird den Punkten die entsprechende Note bei Erreichen der Notengrenze zugeordnet

## Funktion für Berechnung der Notengrenzen

```
notengrenzen_calc <- function(pPunkteMax, pBestGrenze, pBestIndex, pNotenSkala) {
  grenze = rep(0, length(pNotenSkala))
  imax <- length(pNotenSkala)
  schritt <- pBestGrenze/(imax-pBestIndex)
  ## Index für 6 bis zur 4 berechnen
  gi <- imax
  ## Grenze unterhalb der Bestehensgrenze
  g <- 0
  while (gi > pBestIndex) {
    # Füge die Grenze in den Vektor ein
    g <- g + schritt
    grenze[gi] <- g

    # Vermindere den Index
    gi <- gi - 1
  }
  ## Grenze oberhalb der Bestehensgrenze
  schritt <- (pPunkteMax - pBestGrenze)/(pBestIndex)
  ## Index für 4 und besser
  while (gi > 0) {
    g <- g + schritt
    grenze[gi] <- g
  }
}
```

```

    # Vermindere den Index
    gi <- gi - 1
  }
  ## Rückgabewert Vektor "grenze"
  grenze
}

```

Die oben definierte Funktion kann nun verwendet werden, um Notengrenzen zu berechnen.

## Notengrenzen in Abhängigkeit von Bestehensgrenze

In dem folgenden Beispiel werden die Notengrenzen in Abhängigkeit von der Bestehensgrenze bestimmt.

In dieser Beispielrechnung für eine Klausur konnten maximal 200 erreicht werden. Dabei wurde als Bestehensgrenze die Punktezahl `r` festgelegt. Das sind 40 Prozent der Punkte. Nach der Berechnung der Notengrenzen ergibt sich.

Nun wird die Tabelle mit den Notengrenzen erstellt:

Table 1: Geladene Daten aus der Datei data.csv.

Notenskala	Notengrenze
sehr gut	200 - 170
gut	170 - 140
befriedigend	140 - 110
ausreichend	110 - 80
mangelhaft	80 - 40
ungenügend	40 - 0

## Funktion zu Berechnung der Noten

Die Anwendung der folgenden Funktion `note_zuordnen()` setzt voraus, dass man vorher bei einer gegebenen \* maximalen Anzahl von Punkten, \* Bestehensgrenze und \* einem Bestehensindex für eine Notenskala die Notengrenze berechnet hat.

```

note_zuordnen <- function(pPunkte,pGrenzen,pNotenskala) {
  ## Vektor mit der Länge der Punkte erstellen
  ## in dem Vektor wird die Note eingetragen
  note4punkte <- rep(0,length(pPunkte))
  for (i in 1:length(pPunkte)) {
    ## Notenindex auf schlechteste Note setzen
    gi <- length(pNotenskala)
    ## Noten auf schlechtesten Index setzen
    note4punkte[i] <- pNotenskala[gi]
    ## so lange die Note verbessern bis Note jeweilige Notengrenze kleiner als Punkte
    while ((gi > 0) & (pPunkte[i] >= pGrenzen[gi])) {
      ## Index für die Grenze auf bessere Note setzen
      gi <- gi - 1
      ## Notenbezeichnung setzen
      note4punkte[i] <- pNotenskala[gi]
    }
  }
  ### Rückgabewert der Noten fuer alle Punkte
  note4punkte
}

```

## Beispielberechnung für die Noten 1-6

```
maxpunkte <- 100
bestgrenze <- 40
notenskala <- c("sehr gut","gut","befriedigend","ausreichend","mangelhaft","ungenügend")
bestindex <- 4 ## Index für Note, mit der man gerade noch bestanden hat 4 "ausreichend"

grenzen <- notengrenzen_calc(maxpunkte, bestgrenze,bestindex,notenskala)
## grenzen <- c(100,85,70,55,40,20)
## liefert die Grenzen für die Noten punkte<20 bedeutet "ungenügend"
## liefert die Grenzen für die Noten 40 > punkte >= 20 bedeutet "mangelhaft"
### Noten berechnen

### Punkte für 5 Klausuren im Vektor "punkte"
punkte <- c(23,92,33,89,60)

noten <- note_zuordnen(punkte, grenzen, notenskala)

### Notenberechnung ist nun abgeschlossen - Ausgabe
print(noten)

## [1] "mangelhaft"    "sehr gut"       "mangelhaft"     "sehr gut"       "befriedigend"
```

## Beispielberechnung für Noten mit Tendenz

In dem folgenden Beispiel werden die Gesamtpunkte und die Notengrenze beibehalten. Es nun mit den gleichen Punkte eine feinere Unterteilung der Notenskala vorgenommen, dabei werden Noten zwischen 1+ und 6 mit Tendenzen + und - vergeben, wobei die Note 6 ohne Tendenz vergeben wird. Damit ändert sich die Notenskala und der Index für die schwächste Note (hier 4), mit der man gerade nicht bestanden hat.

```
maxpunkte <- 100
bestgrenze <- 40
notenskala <- c("1", "1-", "2+", "2", "2-", "3+", "3", "3-", "4+", "4", "5+", "5", "5-", "6")
bestindex <- 10 ## Index für Note, mit der man gerade noch bestanden hat 4 "ausreichend"

grenzen <- notengrenzen_calc(maxpunkte, bestgrenze,bestindex,notenskala)
## grenzen <- c(100,85,70,55,40,20)
## liefert die Grenzen für die Noten punkte<20 bedeutet "ungenügend"
## liefert die Grenzen für die Noten 40 > punkte >= 20 bedeutet "mangelhaft"
### Noten berechnen

### Punkte für 5 Klausuren im Vektor "punkte"
punkte <- c(23,92,33,89,60)

noten <- note_zuordnen(punkte, grenzen, notenskala)

### Notenberechnung ist nun abgeschlossen - Ausgabe
print(noten)

## [1] "5"  "1-" "5+" "1-" "3"
```

## Klausurergebnisse laden

Nun kann man das auf die Klausurergebnisse der Demodatei anwenden-

```
studentdata <- read.csv("data/klausur.csv", header=TRUE, stringsAsFactors=FALSE)

a1 <- studentdata$aufgabe1
a2 <- studentdata$aufgabe2
a3 <- studentdata$aufgabe3
```

Die erste Datenzeile (2. Zeile) enthält die maximale Anzahl der Punkte für die jeweilige Aufgabe.

### Ausgangstabelle anzeigen

Die folgende Tabelle enthält die geladenen Daten.

Table 2: Geladene Daten aus der Datei data.csv.

name	aufgabe1	aufgabe2	aufgabe3
MAX	16.0	20	16.0
Anna	10.5	15	13.0
Bert	3.5	13	0.0
Camilla	12.0	8	15.0
Dieter	5.0	6	7.0
Emilia	15.0	15	14.5

### Punktesummen berechnen

In folgenden Beispiel werden die Punktesummen in der Variable `summe` berechnet und dann dem Dataframe `studentdata` als Spalte zugewiesen.

```
summe <- a1 + a2 + a3
### in studentdata die Spalte "summe" ergänzen
studentdata$summe <- summe
```

### Ausgangstabelle mit berechneter Summe

Die folgende Tabelle enthält die geladenen Daten.

```
knitr::kable(
  studentdata, booktabs = TRUE,
  caption = 'Geladene Daten mit Summespalte für data.csv.'
)
```

Table 3: Geladene Daten mit Summespalte für data.csv.

name	aufgabe1	aufgabe2	aufgabe3	summe
MAX	16.0	20	16.0	52.0
Anna	10.5	15	13.0	38.5
Bert	3.5	13	0.0	16.5
Camilla	12.0	8	15.0	35.0
Dieter	5.0	6	7.0	18.0
Emilia	15.0	15	14.5	44.5

### Notenskala und Noten für die Punkte berechnen

```

maxpunkte <- summe[1] ## erste Zeile MAX enthält die maximal erreichbaren Punkte
bestgrenze <- summe[1] * 0.5 ## Bestehensgrenze ist 50% der Gesamtpunktzahl
### Als Notenskala werden die deutschen Noten mit Tendenz verwendet
notenskala <- c("1", "1-", "2+", "2", "2-", "3+", "3", "3-", "4+", "4", "4-", "5+", "5", "5-", "6")
bestindex <- 10 ## Index für Note, mit der man gerade noch bestanden hat.
### In Index 10 steht "4" als "ausreichend" und mit "4-" hat man nicht mehr bestanden.

# Der Vektor "grenzen" enthält die Punktegrenzen, ab der man die betreffende Note
# aus der Notenskala erhält. Diese werden in Abhängigkeit von der Bestehensgrenze gewählt.
grenzen <- notengrenzen_calc(maxpunkte, bestgrenze, bestindex, notenskala)

### Noten berechnen
# Gesamtpunktzahl der Klausur findet man im Vektor "summe"
# die Summe ist ein Vektor z.B. summe <- c(23, 92, 33, 89, 60)

noten <- note_zuordnen(summe, grenzen, notenskala)
studentdata$noten <- noten

```

## Ausgangstabelle mit berechneter Note

Die folgende Tabelle enthält Noten auf Basis der berechneten Punktesummen für die geladenen Daten. Als Bestehensgrenze wurde 26 Punkte festgelegt bei einer möglichen Gesamtpunktzahl von 52 Punkten.

```

knitr::kable(
  studentdata, booktabs = TRUE,
  align=c('l', 'r', 'r', 'r', 'r', 'c'),
  caption = 'Geladene Daten mit Summespalte für data.csv.'
)

```

Table 4: Geladene Daten mit Summespalte für data.csv.

name	aufgabe1	aufgabe2	aufgabe3	summe	noten
MAX	16.0	20	16.0	52.0	1
Anna	10.5	15	13.0	38.5	3+
Bert	3.5	13	0.0	16.5	5+
Camilla	12.0	8	15.0	35.0	3
Dieter	5.0	6	7.0	18.0	5+
Emilia	15.0	15	14.5	44.5	2+

## Bestandenvektor berechnen

Die Funktion bekommt eine Punkteliste der Klausur und eine Bestehensgrenze als Eingabe.

```

bestanden_vektor <- function(pPunkte, pBestGrenze) {
  # pPointVec <- c(23, 42, 12, 44, 30)
  ### Der bestandenvektor soll nachher als Textausgabe für die Übungsblätter verwendet werden.
  bestvektor <- rep("-", length(pPunkte))
  for (i in 1:length(pPunkte)) {
    ## Wenn die Punkte der i-ten Klausur die Bestehensgrenze erreicht,
    ## dann soll in dem Bestehensvektor die "bestanden" stehen und
    ## unterhalb der Bestehensgrenze "nicht bestanden" in den Bestehensvektor beschrieben werden.
    if (pPunkte[i] >= pBestGrenze) {
      bestvektor[i] <- "bestanden"
    }
  }
}

```

```

    } else {
      bestvektor[i] <- "nicht bestanden"
    }
  }
  ### bestvektor <- c("bestanden","bestanden","nicht bestanden", "nicht bestanden", "bestanden")
  ### bestandenvektor wird zurückgegeben
  bestvektor
}

### Im Vektor "summe" stehen die Punktesumme und "bestgrenze" ist die Bestehensgrenze
best <- bestanden_vektor(summe,bestgrenze)
### Nun wird der Bestehensvektor in dem Dataframe "studentdata" ergänzt.
studentdata$bestanden <- best

```

## Spalte für das Bestanden

In letzten Spalte wurde nun noch eine weitere Textspalte ergänzt, die das Bestehen als Textvariable mit “bestanden” bzw. “nicht bestanden” enthält.

```

knitr::kable(
  studentdata, booktabs = TRUE,
  align=c('l','r','r','r','r','c','l'),
  caption = 'Geladene Daten mit Summespalte für data.csv.'
)

```

Table 5: Geladene Daten mit Summespalte für data.csv.

name	aufgabe1	aufgabe2	aufgabe3	summe	noten	bestanden
MAX	16.0	20	16.0	52.0	1	bestanden
Anna	10.5	15	13.0	38.5	3+	bestanden
Bert	3.5	13	0.0	16.5	5+	nicht bestanden
Camilla	12.0	8	15.0	35.0	3	bestanden
Dieter	5.0	6	7.0	18.0	5+	nicht bestanden
Emilia	15.0	15	14.5	44.5	2+	bestanden

## Speichern der Daten

In der obigen Datenverarbeitung wurde im Dataframe `studentdata` die Spalte `summe`, `bestanden` und `note` ergänzt und diese erweiterte Datentabelle wird nun in einer neuen Datei mit dem Namen `klausurnote.csv` abgespeichert, damit die Ausgangsdatei in diesem Demo nicht überschrieben wird. Wahl eines neuen Dateinamens ist prinzipiell empfehlenswert, damit bei einer fehlerhaften Berechnung nicht die Rohdaten verändert werden.

```

write.csv(studentdata, "data/klausurnote.csv", row.names = FALSE)

```