Aufgabe 2.5: Univariate Darstellung kategorialer Variablen mit R (keine Punkte)

(i) Lesen Sie die Datei „AbsolventenDat.csv“ in den data frame `dat.absolventen` ein. Beachten Sie, dass fehlende Werte teils mit „-“ und teils mit „k.A.“ gekennzeichnet sind. Die Datei finden Sie in unserem Moodle-Kursraum.

(ii) Lesen Sie die Komponente „Geschlecht“ des data frames in den Vektor `geschlecht` ein und erstellen Sie eine Häufigkeitstabelle dieses Vektors.

(iii) Erzeugen Sie mit dem Befehl `barplot` oder mit `geom_bar` ein einfaches Säulendiagramm der Häufigkeitstabelle des Merkmals `geschlecht` ohne weitere Formatierung.

(iv) Definieren Sie einen Vektor `plot.erg`, in den Sie das Ergebnis des `barplot`-Kommandos schreiben. Lesen Sie die Hilfe zum Kommando `barplot`, um zu verstehen, welche Werte der Vektor `plot.erg` enthält.

(v) Verwenden Sie den Vektor `plot.erg` und die Häufigkeitstabelle aus Teilaufgabe (ii), um die Säulen mit dem `text`-Kommando mit dem String

```
paste(round(prop.table(table(geschlecht))*100,1),"%") )
```

der die relativen Häufigkeiten der beiden Geschlechter in Prozent enthält, zu beschriften.

(vi) Schreiben Sie eine Funktion `pretty.barplot( dat.kat )`, die für einen Vektor `dat.kat` mit kategorialen Daten ein Säulendiagramm zeichnet, das mit den relativen Häufigkeiten beschriftet ist. Testen Sie Ihre Funktion mit dem Vektor `geschlecht`.

(vii) Verwenden Sie nun die Option `main` des `barplot`-Befehls, um die `pretty.barplot`-Funktion um das Argument `titel` zu erweitern, so dass der String `titel` als Überschrift des Säulendiagramms ausgegeben wird. Verwenden Sie ferner die Option `axes`, damit keine y-Achse gezeichnet wird. Testen Sie Ihre Funktion mit dem Vektor `geschlecht` und dem Titel „Geschlecht“.

(viii) Im Titel soll nun neben dem übergebenen String `Titel` noch die Anzahl der gültigen Werte ausgegeben werden. Testen Sie Ihre Funktion wieder mit dem Vektor `geschlecht` und dem Titel „Geschlecht“.

Hinweise:

i.) Verwenden Sie die Befehle `length` und `is.na`, um herauszufinden, wie viele gültige Werte der Vektor `geschlecht` enthält.

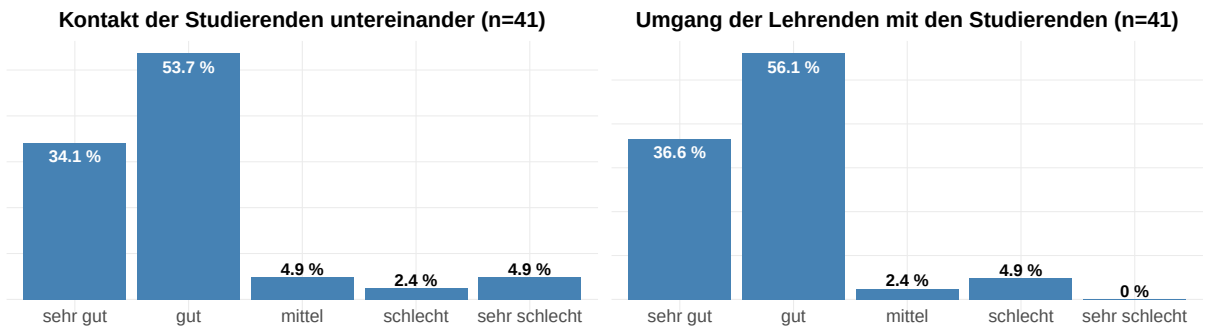
ii.) Mit dem Befehl `paste` können mehrere Strings zu einem zusammen gebaut werden.

(ix) Verwenden Sie die Funktion `pretty.barplot`, um Säulendiagramme für alle kategorialen Komponenten von `dat.absolventen` zu zeichnen.

(x) Verschönern Sie die Säulendiagramme aus Teilaufgabe (ix) noch weiter. Achten Sie dabei auf folgende Details:

- Prinzipiell soll die Prozentzahl oben in weißer Farbe in die dunkelblauen Säulen geschrieben werden. Wenn eine Säule zu klein für die Beschriftung ist, soll die relative Anzahl in schwarzer Farbe über die Säule geschrieben werden.
- Achten Sie auf die korrekte Reihenfolge der Kategorien in dem Säulendiagramm. Beispielsweise sollte die Reihenfolge bei den Beurteilungsfragen „sehr gut“, „gut“, „mittel“, „schlecht“, „sehr schlecht“ sein. Außerdem sollen bei den Beurteilungsfragen auch Ausprägungen auf der x-Achse aufgeführt werden, zu denen es keine Beobachtungen gibt, um die verschiedenen Fragen besser vergleichen zu können.

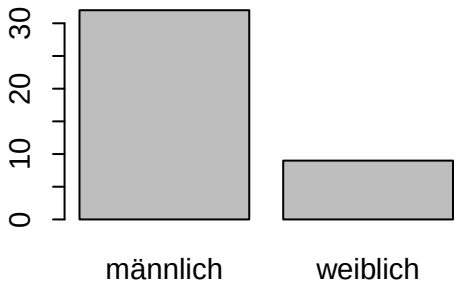
Beispielsweise könnte die Grafik so aussehen:



Hinweis: Wenn Sie wollen, dürfen Sie gerne die Pakete `tidyverse` *und* `ggplot2` *verwenden.*

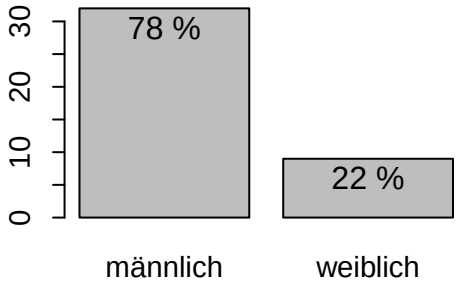
Lösung:

```
#
# (i) (über Oberfläche "Import Dataset" und dann anpassen )
dat.absolventen <- read.csv2("AbsolventenDat.csv",na.strings = c("-","k.A."))
# (ii)
geschlecht <- dat.absolventen$Geschlecht
table(geschlecht)

geschlecht
männlich weiblich
      32        9

# (iii)
barplot(table(geschlecht))
```

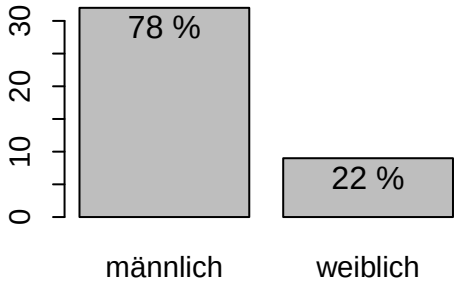



```
# (iv)
plot.erg <- barplot(table(geschlecht))
```

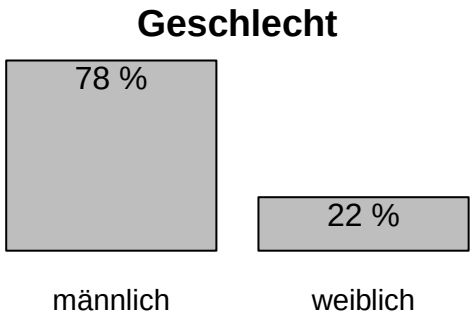
```
# (v)
text( plot.erg[,1], table(geschlecht)-3,
      paste(round(prop.table(table(geschlecht))*100,1),"%") )
```



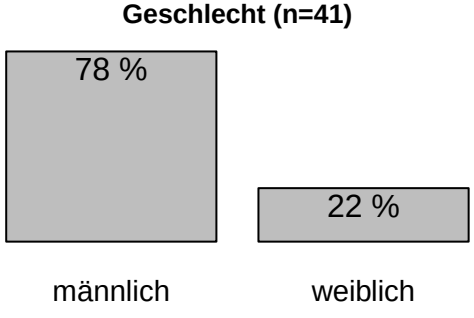

```
# (vi)
pretty.barplot <- function(dat.kat)
{
  plot.erg <- barplot(table(dat.kat))
  text( plot.erg[,1], table(dat.kat)-3,
        paste(round(prop.table(table(dat.kat))*100,1),"%") )
}
pretty.barplot(geschlecht)
```




```
# (vii)
pretty.barplot <- function(dat.kat, Titel)
{
  par( mar = c(5, 4, 4, 2) - 2 ) # nur wg. Rmarkdown
  plot.erg <- barplot(table(dat.kat),
                  main=Titel,
                  axes=FALSE)
  text( plot.erg[,1], table(dat.kat)-3,
        paste(round(prop.table(table(dat.kat))*100,1),"%") )
}
pretty.barplot(geschlecht, "Geschlecht")
```

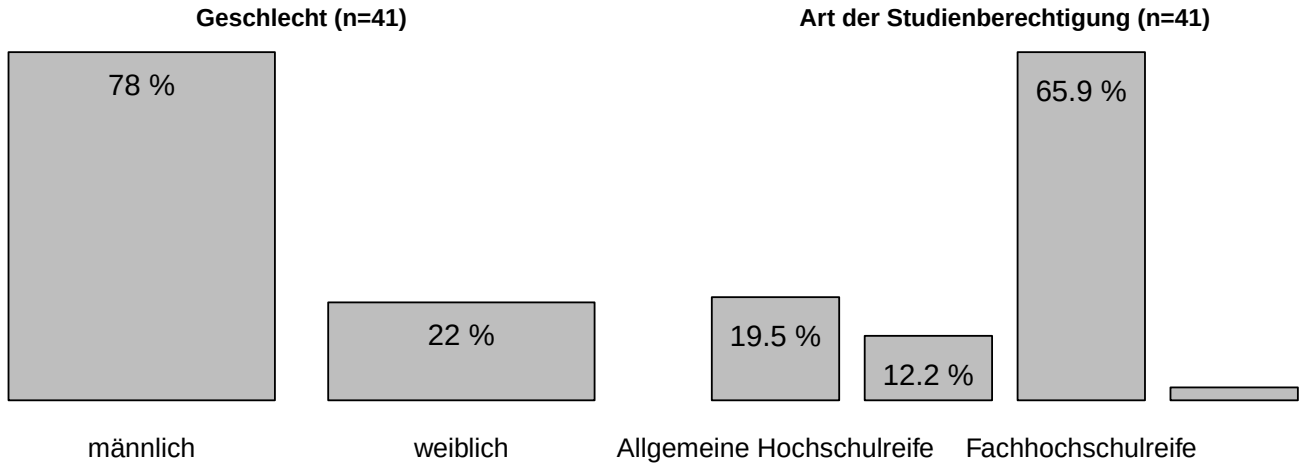



```
# (viii)
pretty.barplot <- function(dat.kat, Titel)
{
  par( mar = c(5, 4, 4, 2) -2 )
  plot.erg <- barplot(table(dat.kat),
                  main=paste(Titel," (n=",length(dat.kat[!is.na(dat.kat)]),")",
                             sep=""),
                  axes=FALSE, cex.main=0.8)
  text( plot.erg[,1], table(dat.kat)-3,
        paste(round(prop.table(table(dat.kat))*100,1),"%") )
}
pretty.barplot(geschlecht, "Geschlecht")
```



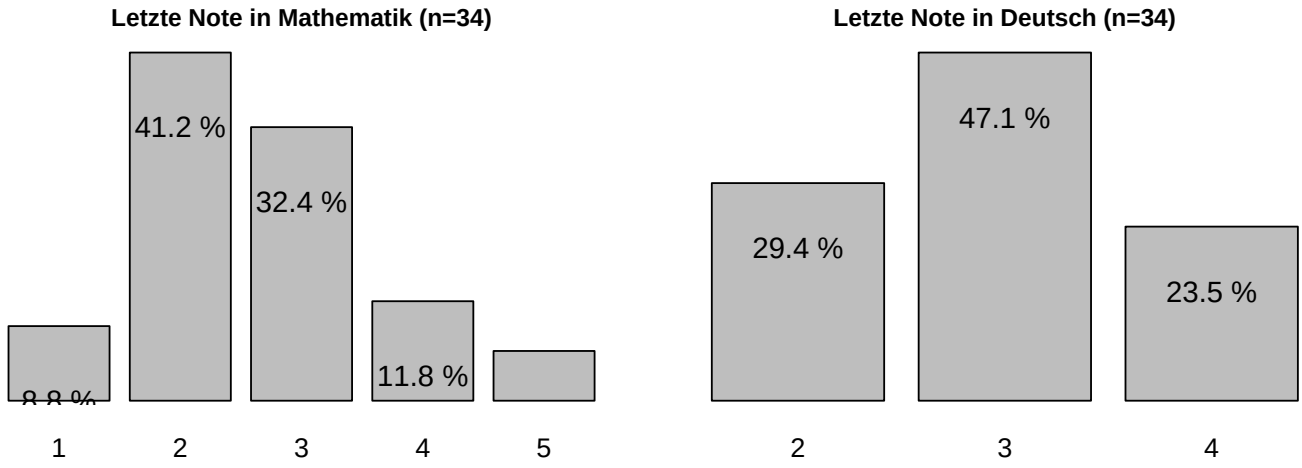
```r
# (ix)
par(mfrow=c(1,2))
pretty.barplot(dat.absolventen$Geschlecht,"Geschlecht")
```

```r
pretty.barplot(dat.absolventen$Art.der.Studienberechtigung,
               "Art der Studienberechtigung")
```



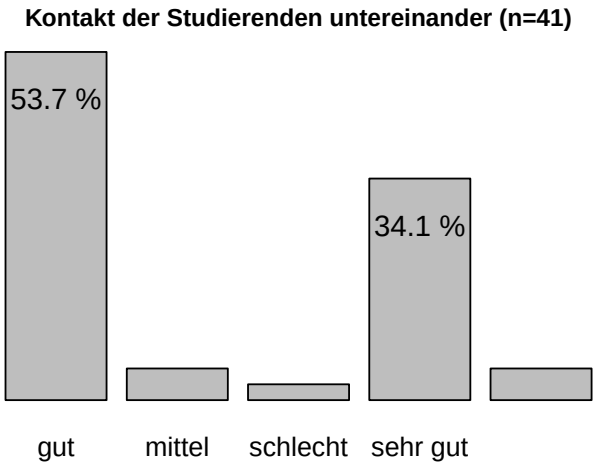

```r
pretty.barplot(dat.absolventen$Letzte.Note.in.Mathematik,"Letzte Note in Mathematik")
```

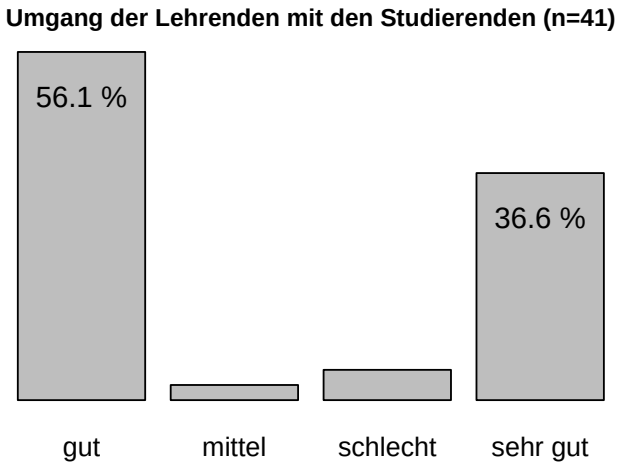
```r
pretty.barplot(dat.absolventen$Letzte.Note.in.Deutsch,"Letzte Note in Deutsch")
```




```r
pretty.barplot(dat.absolventen$Kontakt.der.Studierenden.untereinander,
               "Kontakt der Studierenden untereinander")
```

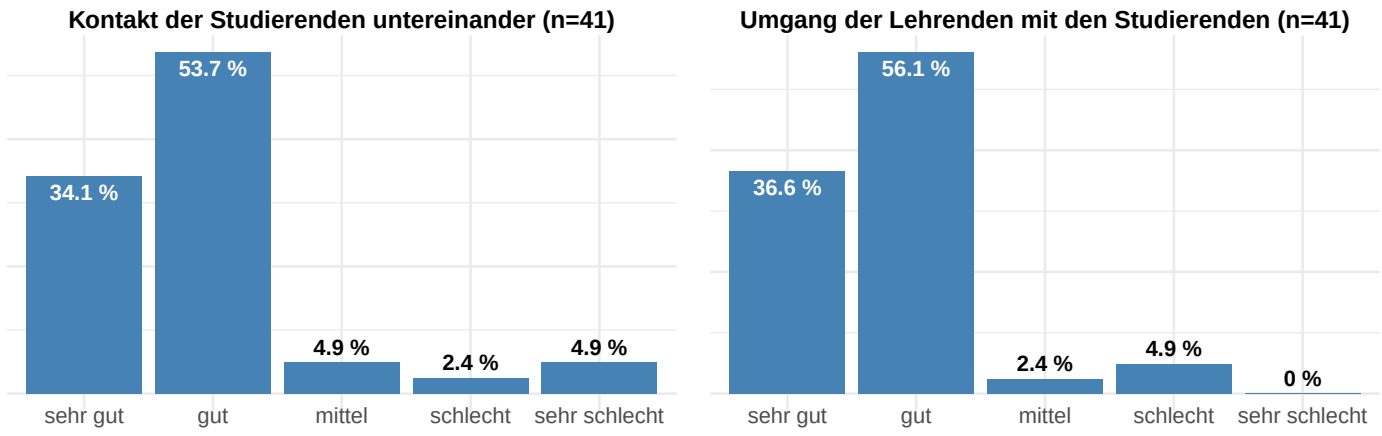
```r
pretty.barplot(dat.absolventen$Umgang.der.Lehrenden.mit.den.Studierenden,
               "Umgang der Lehrenden mit den Studierenden")
```






```r
# (x)
# noch schöneres Säulendiagramm (ich hab's mit ggplot gemacht)

# List of packages required for this analysis
pkg <- c("ggplot2","grid","gridExtra")

# Check if packages are not installed and assign the
# names of the packages not installed to the variable new.pkg
new.pkg <- pkg[!(pkg %in% installed.packages())]

# If there are any packages in the list that aren't installed,
# install them
if (length(new.pkg)) {
  install.packages(new.pkg, repos = "http://cran.rstudio.com")
}
# benötigte Pakete laden
library(ggplot2)
library(grid)
library(gridExtra)
#
even.prettier.barplot <- function(dat.kat, Titel)
{
  # Prozentuale Häufigkeiten des kategoriellen Vektors in data frame umwandeln,
  # weil der ggplot-Befehl eine data frame erwartet
  df <- as.data.frame(table(dat.kat)/length(dat.kat[!is.na(dat.kat)])*100)
  names(df)[1]="Cat"    # Kategorie
  lbl.vjust <- df$Freq  # Häufigkeit
  lbl.col <- rep("white",length(df$Freq)) # weiße Beschriftung bzw.
  lbl.col[lbl.vjust<8] = "black"          # schwarze Beschriftung, falls weniger
  # als acht Prozent
  lbl.vjust[lbl.vjust<8] = -0.5             # bis 8% über die Säule schreiben
  lbl.vjust[lbl.vjust>=8] = 1.5             # ab 8% in die Säule schreiben
  plot <- ggplot(data=df, aes(x=Cat,y=Freq) ) + ggtitle(paste(Titel," (n=",length(dat.kat[!is.na(dat.kat)])
    geom_bar(stat="identity", fill="steelblue")+
    geom_text(aes(label=paste(round(Freq,1),"%"), fontface = "bold"), vjust = lbl.vjust, color= lbl.col, siz
    theme_minimal()+
    theme(plot.title = element_text(size = 10,hjust = 0.5,margin = margin(t = 15, b = -10), face="bold"))+
    theme(axis.title.x = element_blank(), axis.text.x  = element_text(size=9,margin = margin(t = -5, b = 20)
    theme(axis.title.y = element_blank(), axis.text.y  = element_blank())

  return (plot)
}
attach(dat.absolventen)
# Umsortierung der Levels
Kontakt.der.Studierenden.untereinander <- factor(Kontakt.der.Studierenden.untereinander,
                                                 levels = c("sehr gut", "gut", "mittel", "schlecht", "sehr
Umgang.der.Lehrenden.mit.den.Studierenden <- factor(Umgang.der.Lehrenden.mit.den.Studierenden,
                                                    levels = c("sehr gut", "gut", "mittel", "schlecht", "sel
```

```
plot1 <- even.prettier.barplot(Kontakt.der.Studierenden.untereinander,
                               "Kontakt der Studierenden untereinander")
plot2 <- even.prettier.barplot(Umgang.der.Lehrenden.mit.den.Studierenden,
                               "Umgang der Lehrenden mit den Studierenden")
grid.arrange(plot1, plot2, ncol=2)
```




```
detach(dat.absolventen)
```

Aufgabe 3.4: Histogramme (4 Punkte)

(STO-deskrSta07 | 1)

Die Jahresumsätze von 1000 Unternehmen sind in der folgenden Tabelle klassifiziert gegeben:

Umsatz [in Mio. Euro]	[1,5)	[5,10)	[10,20)	[20,40]
Anzahl Unternehmen	490	410	50	50

(i) Spezifizieren Sie für die gegebenen Daten die Begriffe Stichprobe, Merkmal, Merkmalsträger, Skalenniveau und Stichprobenumfang.

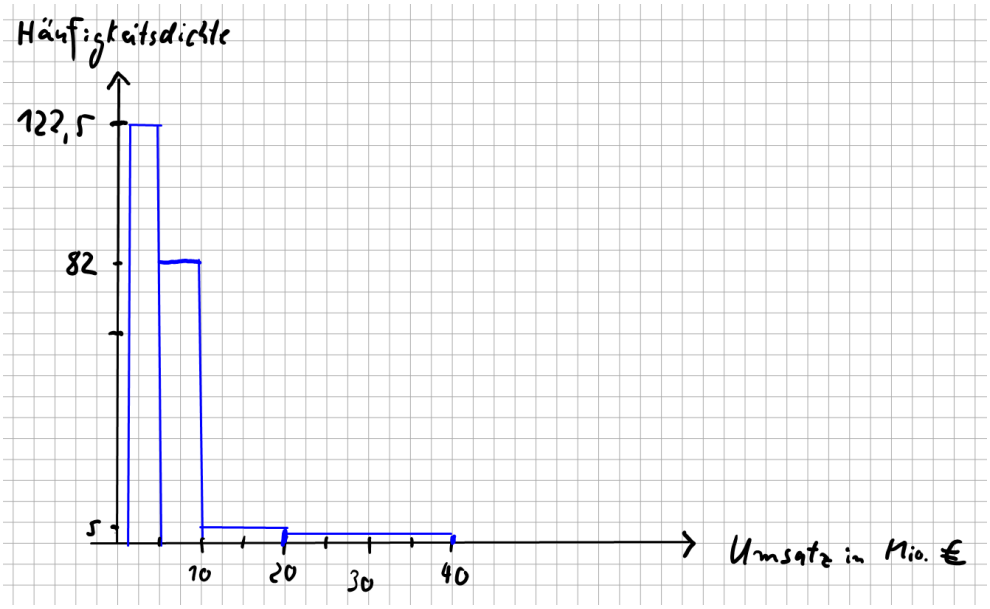
(ii) Skizzieren Sie mit der gegebenen Intervalleinteilung das zugehörige Histogramm der absoluten Klassenhäufigkeiten, wobei die Höhe H_1 des Histogrammrechtecks der ersten Klasse [1,5) [Mio. Euro] $H_1 = 20$[cm] sein soll.

Lösung:

(i) Die Stichprobe besteht aus den 1000 Unternehmen, für die das Merkmal „Umsatz“ erfasst wurde. Merkmalsträger sind die einzelnen Unternehmen, das Skalenniveau ist metrisch und der Stichprobenumfang beträgt $n = 1000$. (2 Punkte)

(ii) Die Höhe der vier Rechtecke des Histogramms stellt die Häufigkeitsdichte, das ist die absolute Häufigkeit geteilt durch die Intervallbreite dar. Es ergeben sich folgende Werte:

Umsatz [in Mio. Euro]	[1,5)	[5,10)	[10,20)	[20,40]
Anzahl Unternehmen	490	410	50	50
Breite des Intervalls	4	5	10	20
Häufigkeitsdichte	122.5	82	5	2.5
Höhe in cm	20	13,4	0,8	0,4



(2 Punkte)

Aufgabe 3.5: Histogramme (keine Punkte)

(i) Lesen Sie die Datei `AbsolventenDat.csv`, die Sie in unserem Moodle-Kursraum finden, ein. Achten Sie dabei auf das korrekte Einlesen von fehlenden Werten.

(ii) Lassen Sie sich mit `summary` die wichtigsten Kenngrößen des Merkmals „Eingangsnote“ des data frames `dat.absolventen` ausgeben.

(iii) Erzeugen Sie ein einfaches Histogramm des Merkmals „Eingangsnote“ ohne weitere Formatierung. Dabei sollen alle Intervalle die Länge 0,5 haben und das erste bei 1 starten.

(iv) Verwenden Sie das `stat_bin`-Kommando, um ein Histogramm zu zeichnen, bei dem die Rechtecke mit den entsprechenden relativen Häufgkeiten (in Prozent) beschriftet sind. Über 5% soll die Häufigkeit im Rechteck angegeben werden. Darunter über dem Rechteck. Die y-Achse soll mit „Anzahl“ statt „count“ beschriftet werden.

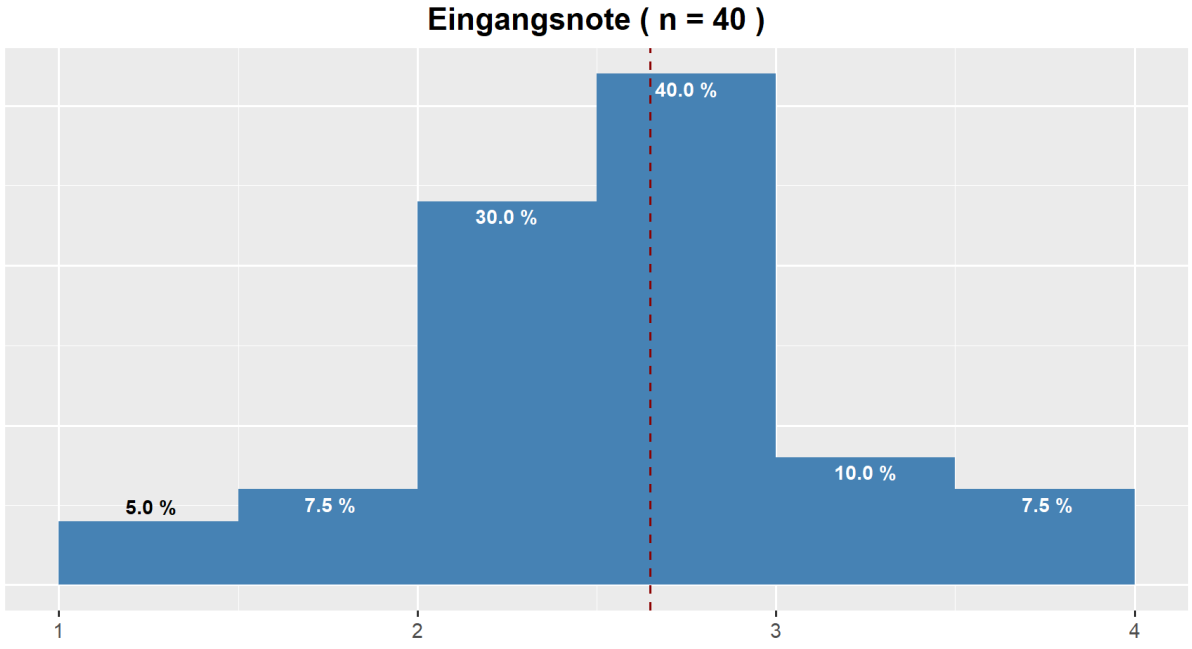
(v) Schreiben Sie eine Funktion `pretty.hist( dat.num )`, die für einen numerischen Vektor `dat.num` ein Histogramm zeichnet, das mit den relativen Häufigkeiten beschriftet ist.

Testen Sie Ihre Funktion mit dem Vektor `dat.absolventen$Abschlussnote`.

(vi) Erweitern Sie nun die `pretty.hist`-Funktion um das Argument `titel`. Im Titel des Histogramms soll neben dem übergebenen String `titel` noch die Anzahl der gültigen Werte ausgegeben werden. Außerdem sollen die x-Achse keinen Titel haben und die y-Achse ganz ausgeblendet wird. Testen Sie Ihre Funktion wieder mit dem Merkmal `Eingangsnote` und dem Titel „Eingangsnote“.

(vii) Zeichnen Sie in das Histogramm beim arithmetische Mittel (vulgo: Durchschnitt) eine gestrichelte senkrechte Linie ein und beschriften Sie diese Linie mit dem Durchschnittswert.

Beispielsweise könnte die fertige Grafik so aussehen:

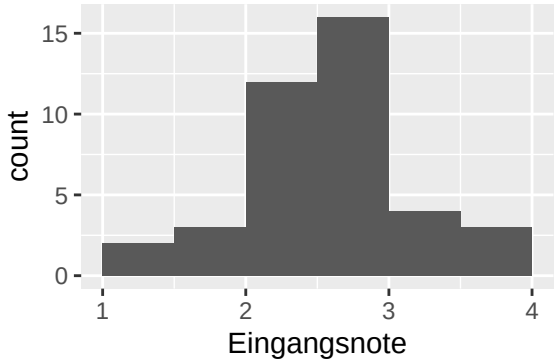


Lösung:

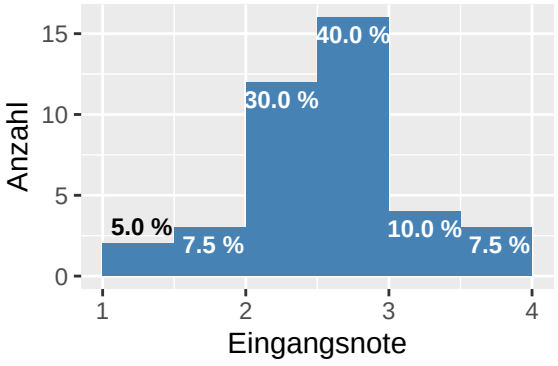
```
#
# (i)
dat.Absolventen <- read.csv2("AbsolventenDat.csv", na.strings=c("NA","-","k.A."))
#
# (ii)
summary(dat.Absolventen$Eingangsnote)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   1.33    2.40    2.65    2.65    2.90    3.80       1
```

```
#
# (iii)
library(tidyverse)
dat.Absolventen |> ggplot() + geom_histogram(aes(x=Eingangsnote), binwidth = 0.5,
                                             boundary=0.5)
```

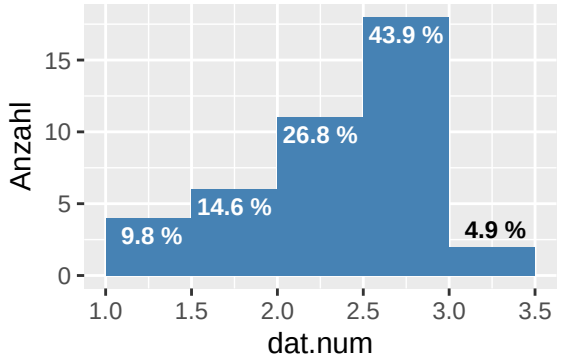



```
#
# (iv)
anzahl <- sum( !is.na(dat.Absolventen$Eingangsnote) )

dat.Absolventen |> ggplot(aes(x=Eingangsnote)) + geom_histogram(
                                             binwidth = 0.5,
                                             boundary=0.5, fill = "steelblue") +
                  labs( y = "Anzahl" ) +
   stat_bin(binwidth=0.5, boundary=0.5, geom='text',
            aes(label=ifelse(after_stat(count)/anzahl>=0.06,
                       paste(
                         format(round(after_stat(count)/anzahl*100,1),nsmall=1),
                         "%"),"")),
            vjust = 1.5, fontface = "bold", size=3, color = "white") +
   stat_bin(binwidth=0.5, boundary=0.5, geom='text',
            aes(label=ifelse(after_stat(count)/anzahl<0.06,
                       paste(
                         format(round(after_stat(count)/anzahl*100,1),nsmall=1),
                         "%"),"")),
            vjust = -0.5, fontface = "bold", size=3, color = "black")
```




```
#
# (v)
pretty_hist <- function( dat.num )
{
  anzahl <- sum( !is.na(dat.num) )

  aux <- tibble( dat.num = dat.num )

  aux |> ggplot(aes(x=dat.num)) + geom_histogram(
                                             binwidth = 0.5,
                                             boundary=0.5, fill = "steelblue") +
                  labs( y = "Anzahl" ) +
```

```
   stat_bin(binwidth=0.5, boundary=0.5, geom='text',
            aes(label=ifelse(after_stat(count)/anzahl>=0.06,
                       paste(
                         format(round(after_stat(count)/anzahl*100,1),nsmall=1),
                         "%"),"")),
            vjust = 1.5, fontface = "bold", size=3, color = "white") +
   stat_bin(binwidth=0.5, boundary=0.5, geom='text',
            aes(label=ifelse(after_stat(count)/anzahl<0.06,
                       paste(
                         format(round(after_stat(count)/anzahl*100,1),nsmall=1),
                         "%"),"")),
            vjust = -0.5, fontface = "bold", size=3, color = "black")
}
pretty_hist( dat.Absolventen$Abschlussnote)
```




```
#
# (vi) & (vii)
pretty_hist <- function( dat.num, titel="" )
{
  anzahl <- sum( !is.na(dat.num) )
  arithMittel <- mean( dat.num, na.rm = TRUE )

  aux <- tibble( dat.num = dat.num )

  aux |> ggplot(aes(x=dat.num)) + geom_histogram(
                                          binwidth = 0.5,
                                          boundary=0.5, fill = "steelblue") +
                labs( title = paste(titel,"( n =", anzahl,")") ) +
   stat_bin(binwidth=0.5, boundary=0.5, geom='text',
            aes(label=ifelse(after_stat(count)/anzahl>=0.06,
                       paste(
                         format(round(after_stat(count)/anzahl*100,1),nsmall=1),
                         "%"),"")),
            vjust = 1.5, fontface = "bold", size=3, color = "white") +
   stat_bin(binwidth=0.5, boundary=0.5, geom='text',
            aes(label=ifelse(after_stat(count)/anzahl<0.06,
                       paste(
                         format(round(after_stat(count)/anzahl*100,1),nsmall=1),
                         "%"),"")),
            vjust = -0.5, fontface = "bold", size=3, color = "black") +
    theme(axis.text.y = element_blank(), axis.title = element_blank(),
          axis.ticks.y.left = element_blank(),
          plot.title = element_text(hjust = 0.5, face = "bold")) +
   geom_vline(xintercept=arithMittel, linetype='dashed', color = "darkred")
}
pretty_hist( dat.Absolventen$Eingangsnote, "Eingangsnote")
```

Eingangsnote (n = 40)