



```

#Alle Daten ohne Werte rausschmeißen
kr <- na.omit(kr)

#Anlegen einer Tabelle in der die Ergebnisse gespeichert werden (pro definiertes
Magerungsmittel nach NoNeK)
erg <-
read.csv(text="Unbestimmt,Nicht_erkennbar,Granit_weiss,Granit_rot,Granit_glimmer
,Granit,Granit_a_weiss,Granit_a_rot,Sand,Organisch,Schamotte,Quarzit,Kalkstein,Q
uarz,HK,Flint")

#Zählvariable anlegen
z=1
#Die erste Zeile wird mit Nullen befüllt, damit in sie gerechnet werden kann
while (z <= length(erg))
{
  erg[1,z] <- 0
  z <- z+1
}

#R interpretiert das Magerungsmittel als Zahl, diese kann nicht mit einer
führenden 0 beginnen, wie sie es in NoNeK kann, daher muss
#in einen Text umgewandelt werden
kr$magmt <- as.character(kr$magmt)

#Zählvariable j
j=1

#Jetzt werden für alle Magerungsmittel, die weniger als sieben Stellen haben,
eine führende 0 angefügt
#!!!!ACHTUNG ist bei der Aufnahme ein Fehler unterlaufen und es wurden zu wenig
Stellen eingegeben, wir es hier blind überschrieben
#und kann zu falschen Ergebnissen führen, also am Besten in Access vorher die
Spalte "48magmt" nach Größe sortieren und schauen, ob alle 8 Stellen haben

#Für jede Zeile in unserer Ausgangstabelle soll geschaut werden
while (j < nrow(kr))
{
  #So lange wie die Anzahl der Stellen kleiner 8 ist, werden 0en angefügt.
  #Es wird eine Schleife ausgeführt, da auch der Eintrag 00 in Nonek möglich
ist. Sind also 8 nullen vorhanden, so wird es als einfache 0 gelesen
  #In diesem Falle müssen sieben angefügt werden
  while (nchar(kr[j,1]) < 8)
  {
    #Null anfügen
    kr[j,1] <- str_replace_all(capture.output(cat("0",kr[j,1])), " ", "")
  }
  #Nächste Zeile
  j=j+1
}

#Zählvariable anlegen

```

```
k=1
```

```
#Jetzt kommt das eigentliche Zerlegen
```

```
#Solange noch Zeilen in der Ausgangstabelle vorhanden sind soll
```

```
while (k <= nrow(kr))
```

```
{
```

```
  #Interne Zählvariable
```

```
  x <- 1
```

```
  #Solange x kleiner 8 ist (bei 8 Stellen, können max. vier Magerungsmittel  
angegeben werden, x wird immer um 2 erhöht)
```

```
  while (x <= 8)
```

```
  {
```

```
    #Wenn der Substring (Textteil) in der Ausgangstabelle zwische x und x+1 (x  
wird immer um 2 erhöht, dadurch werden die Pärchen ausgelesen)
```

```
    #Wenn der Textteil = 11 ist dann wird in der Ergebnistabelle in der Dritten  
Spalte (Granit_weiß, s.o.) die Anzahl addiert
```

```
    #Das wird für jedes definierte Magerungsmittel durchgeführt
```

```
    #ACHTUNG!!! vor jedem Neustart der Schleife, muss die Ergebnistabelle  
zurückgesetzt werden, am Besten erg<- NULL und dann von vorne anfangen
```

```
    if (substr(kr[k,1], x,x+1) == '11')
```

```
    {
```

```
      erg[1,3] <- erg[1,3] + kr[k,2]
```

```
    }
```

```
    #Im Falle von unbekannt und unbestimmt, machen Werte nur Sinn, wenn sie an  
der ersten Posititon stehen, da (zumindest 00) als Füllmenge für den Rest gilt
```

```
    if (substr(kr[k,1], x,x+1) == '99' && x == 1)
```

```
    {
```

```
      erg[1,1] <- erg[1,1] + kr[k,2]
```

```
    }
```

```
    if (substr(kr[k,1], x,x+1) == '00' && x == 1)
```

```
    {
```

```
      erg[1,2] <- erg[1,2] + kr[k,2]
```

```
    }
```

```
    if (substr(kr[k,1], x,x+1) == '21')
```

```
    {
```

```
      erg[1,4] <- erg[1,4]+ kr[k,2]
```

```
    }
```

```
    if (substr(kr[k,1], x,x+1) == '22')
```

```
    {
```

```
      erg[1,5] <- erg[1,5] + kr[k,2]
```

```
    }
```

```
    if (substr(kr[k,1], x,x+1) == '24')
```

```
    {
```

```
      erg[1,6] <- erg[1,6] + kr[k,2]
```

```
    }
```

```

if (substr(kr[k,1], x,x+1) == '25')
{
  erg[1,7] <- erg[1,7] + kr[k,2]
}
if (substr(kr[k,1], x,x+1) == '26')
{
  erg[1,8] <- erg[1,8] + kr[k,2]
}
if (substr(kr[k,1], x,x+1) == '02')
{
  erg[1,9] <- erg[1,9] + kr[k,2]
}
if (substr(kr[k,1], x,x+1) == '03')
{
  erg[1,10] <- erg[1,10] + kr[k,2]
}
if (substr(kr[k,1], x,x+1) == '04')
{
  erg[1,11] <- erg[1,11] + kr[k,2]
}
if (substr(kr[k,1], x,x+1) == '05')
{
  erg[1,12] <- erg[1,12] + kr[k,2]
}
if (substr(kr[k,1], x,x+1) == '06')
{
  erg[1,13] <- erg[1,13] + kr[k,2]
}
if (substr(kr[k,1], x,x+1) == '07')
{
  erg[1,14] <- erg[1,14] + kr[k,2]
}
if (substr(kr[k,1], x,x+1) == '31')
{
  erg[1,15] <- erg[1,15] + kr[k,2]
}
if (substr(kr[k,1], x,x+1) == '32')
{
  erg[1,16] <- erg[1,16] + kr[k,2]
}
}
#Die Zählvariable für die Stellen wird erhöht
x <- x+2
}
#Die Zählvariable für die Zeilen wird erhöht

```

```

k <- k+1

}
#Das Ergebnis als Vector
print("Ergebnis:")
print(erg)
#Kommen wir zur darstellung
#Zunächst drehen wir die Tabelle, für eine bessere Ansicht
zeich <- as.data.frame(t(erg))
#gedrehte Tabelle
print("Gedrehte Tabelle:")
print(zeich)

#Der folgende Block ist durchzuführen, wenn nur die Magerungsmittel dargestellt
werden sollen, die Vorhanden sind
#Sollen alle dargestellt werden, mit ZEICHNEN fortfahren
#Zählvariable anlegen
f=1
#Solange es in der Ergebnistabelle Zeilen gibt
#Jetzt müssen alle rausgeworfen werden, die keine Werte haben, also alle mit 0
durch NA ersetzen
while (f <= nrow(zeich))
{
  if (zeich[f,1] == 0)
  {zeich[f,1] <- NA}
  f=f+1
}
#Danach alle mit NA löschen
zeich <- na.omit(zeich)

#ZEICHNEN
#Setzen des Namens für die Ordinate
colnames(zeich)<-c("Anzahl")
#Die Namen der Spalten in neue Spalte ablegen
zeich$Magerung <- rownames(zeich)
#Das ganze Zeichnen
ggplot(zeich, aes(x= factor(Magerung),y= Anzahl))+geom_bar(stat= "identity",
fill = "white",colour =
"black")+theme(axis.text.x=element_text(angle=-90))+geom_text(aes(label =
Anzahl), hjust = 0.5, vjust = -1) + scale_x_discrete(name="Magerung")
#Beim Export in ein pdf o.ä.
#CM in INCHES für die Angabe beim Export
cm_hoehe <- 15 #Hier die gewünschte cm Zahl für die Höhe angeben
cm_breite <- 5 #Hier die gewünschte cm Zahl für die breite angeben

#Ausführen
print("Höhe:")
print(cm_hoehe * 0.39370)
print("Breite:")
print(cm_breite * 0.39370)

```