

```

#####
#
# Reinhard Hoepfner, Statistik mit Rechneruebungen, SoSe19
#
# Mittelwert und Varianz eines Datensatzes
# am Beispiel SANDSTEINPOROSITAET
#
# 25.04.19
#
#####

# datensatz
porositaet <- c(
  22.1, 23.5, 25.3, 26.6, 23.9, 26.0, 22.8, 22.3, 23.1, 23.0, 21.0, 21.8,
  22.0, 22.2, 22.3, 22.4, 22.4, 22.4, 22.3, 21.6, 22.1, 22.6, 22.1, 21.9,
  22.3, 23.9, 23.2, 22.5, 23.7, 23.3, 24.4, 22.6, 23.9, 24.2, 27.6, 27.9,
  25.2, 21.7, 20.0, 19.8, 21.5, 25.6, 25.3, 24.1, 28.6, 23.7, 24.0, 21.8,
  24.9, 24.2, 25.0, 23.7, 27.3, 23.0, 23.8, 21.2, 21.1
)
dat <- porositaet ;

#####

# laenge des datensatzes
n <- length(dat) ; n ;
# 57

# mittelwert des datensatzes
m <- mean(dat) ; m ;
# 23.3807
sum(dat)/n ;

# varianz des datensatzes
# beachte: var() ist schon mit  $\frac{1}{n-1}$ -normierung
# in R vorprogrammiert, vergleiche
# help(var) ;
s2 <- var(dat) ; s2 ;
# 3.540871
( (dat-m)%*(dat-m) ) / (n-1) ;
( (dat-m)%*(dat-m) )/n ;

# standardabweichung
s <- sqrt(s2) ; s ;
# 1.88172

#####

# graphische darstellung : mittelwert und varianz
# als minimierung eines L^2-abstandes

# funktionsdefinition
quadratabstand <- function(x){
  ( dat - rep(x,n) )%*( dat - rep(x,n) ) / n
} ; # ende funktionsdefinition

# funktionswerte berechnen
feinheit <- 5 ;
x <- seq( min(dat)-1, max(dat)+1, 10^{-feinheit} ) ;
y <- rep(0,length(x)) ;
# startwert zum ueberschreiben
for( i in 1:length(x) ) y[i] <- quadratabstand(x[i]) ;

# funktion plotten
plot( x, y, type="l" ) ;
rug( dat, col=2 ) ;
title( "die funktion m --> sum_j ( (x_j - m)^2 ) / n fuer den datensatz
sandsteinporositaet" ) ;

```

```

# minimalstelle suchen
s2schaetzer <- min(y) ;
# das minimum der funktionswerte
mschaetzer <- x[ y == s2schaetzer ] ;
# die minimalstelle
points( mschaetzer, s2schaetzer, col=2 ) ;
abline( h = s2schaetzer, col=2 ) ;
abline( v = mschaetzer, col=2 ) ;

# identifiziere die minimalstelle als
# empirischen mittelwert der daten
# und den minimierten quadratischen abstand als
# varianz der daten in 1/n - normierung :

round( mschaetzer, feinheit ) ;
round( sum(dat)/n , feinheit ) ;
round( mschaetzer, feinheit ) == round( sum(dat)/n , feinheit ) ;

round( s2schaetzer, feinheit ) ;
round( ( (dat-m)%*(dat-m) )/n , feinheit ) ;
round( s2schaetzer, feinheit ) == round( ( (dat-m)%*(dat-m) )/n , feinheit ) ;

##### ende 25.04.19 #####

```