

Rauschreduktion

Christoph Steinbeck, et al.

10/05/2018

Rauschreduktion mit gleitendem Mittelwert

Erzeugen der Ausgangsdaten

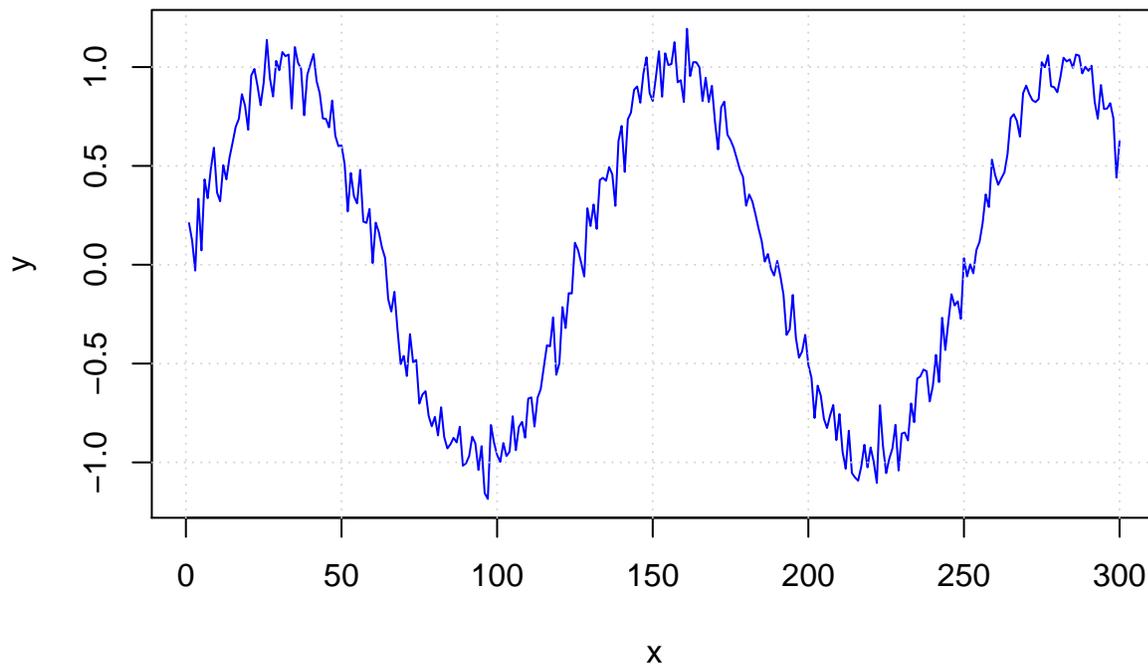
Um zum besseren Verständnis der Vorgänge größere Kontrolle relevant Parameter zu haben, arbeiten wir mit künstlich erzeugten Daten.

Zu diesem Zweck erzeugen wir zunächst eine Sinuskurve, der wir mit der Funktion `rnorm` ein Rauschen überlagern.

```
set.seed(993)
x <- 1:300
y <- sin(x/20) + rnorm(300,sd=.1)
# y[251:255] <- NA
```

Jetzt lassen Sie uns die erzeugten Daten plotten:

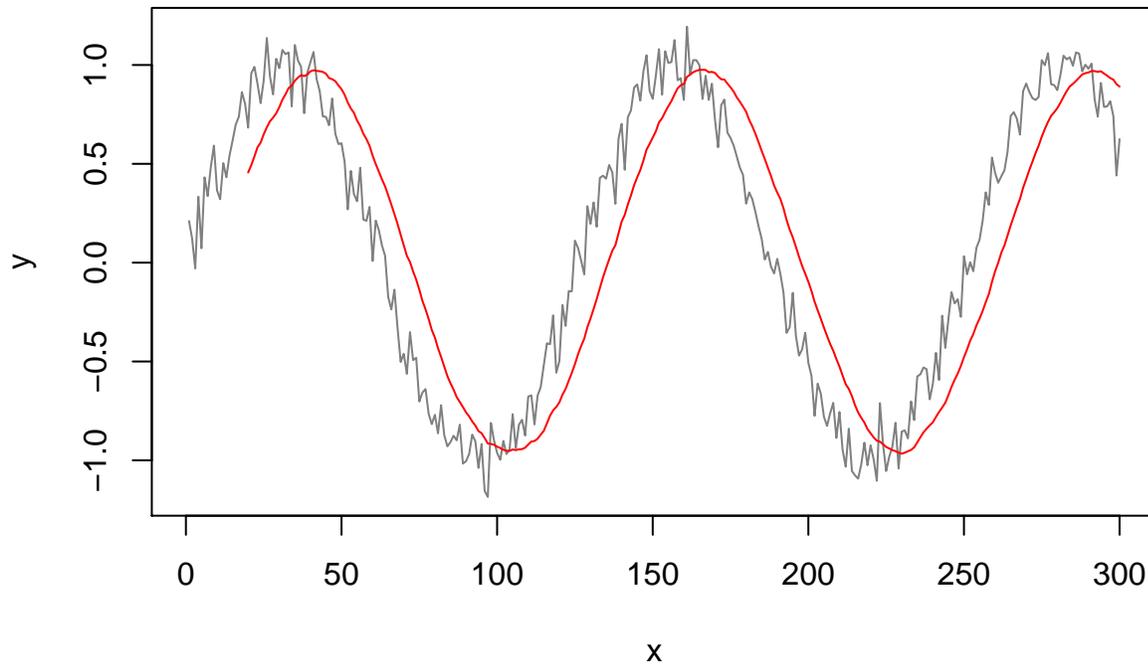
```
plot(x, y, type="l", col="blue")
grid()
```



AUFGABE: Lesen Sie die Hilfe zu `rnorm()` und verstehen Sie die hier verwendeten Parameter. Verändern Sie die Parameter und beobachten Sie den Effekt durch nochmaliges Plotten der neuen Daten.

Glätten der Daten

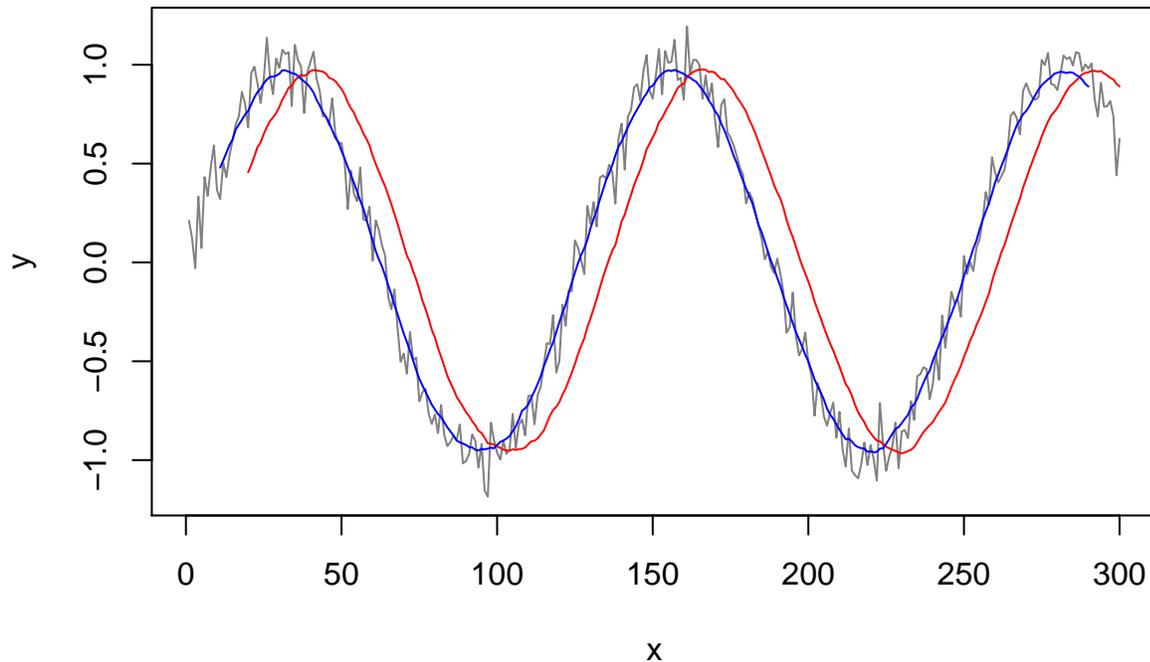
```
f20 <- rep(1/20, 20)
y_lag <- filter(y, f20, sides=1)
plot(x, y, type="l", col=grey(.5))
lines(x, y_lag, col="red")
```



AUFGABE: Betrachten Sie den Inhalt von f20. AUFGABE: Lesen Sie die Hilfe zur filter() Funktion und verstehen Sie die Funktionsweise und die Funktion von F20.

Das Hinterherhinken der geglätteten Daten ist nicht in jedem Fall wünschenswert. Deshalb erzeugen wir eine Rauschreduktion, die für den aktuellen Mittelwert auch prospektive Daten verwendet:

```
f21 <- rep(1/21, 21)
y_zent <- filter(y, f21, sides=2)
plot(x, y, type="l", col=grey(.5))
lines(x, y_lag, col="red")
lines(x, y_zent, col="blue")
```



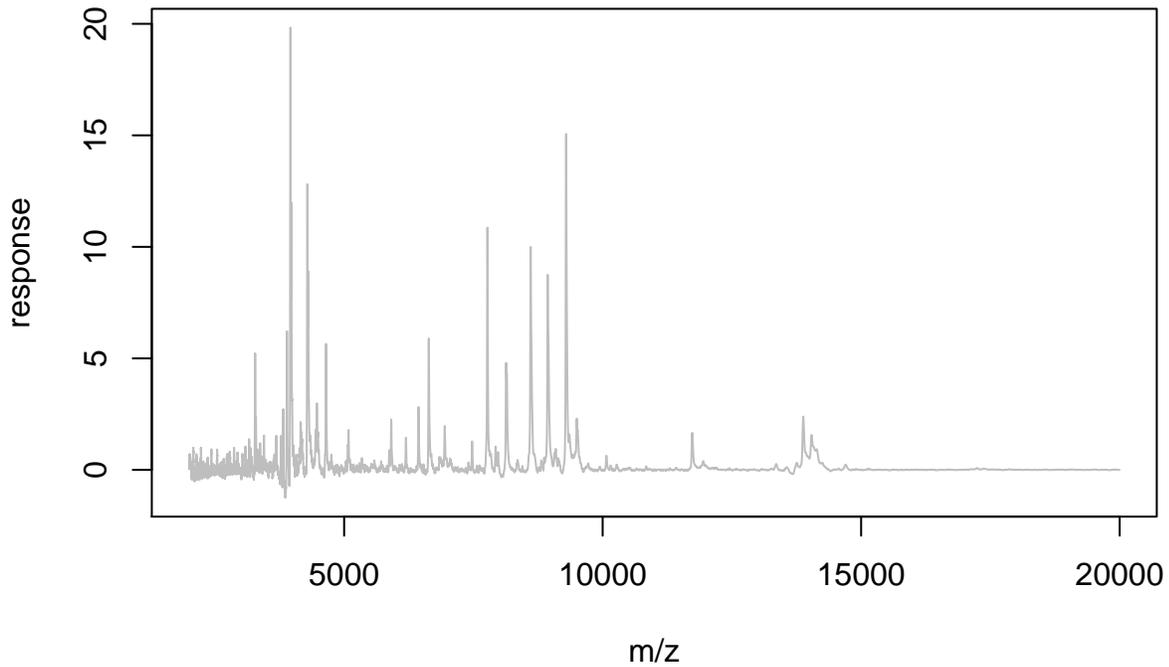
AUFGABE: Im Block Sinusdaten ist eine Zeile auskommentiert. Entfernen Sie das Kommentarzeichen und beobachten Sie den Effekt.

Jetzt versuchen wir das mit Daten aus einem realen Experiment, und zwar mit den Prostatakrebs-Proteomdaten aus Adam, B.-L. et al. Cancer Res. 62, 3609–3614 (2002), die wir in der Vorlesung bereits gesehen haben. Zunächst laden wir die Rohdaten von <https://github.com/rwehrens/CWR/tree/master/ChemometricsWithRData/data>

Zur Ansicht plotten wir zunächst das erste Spektrum

```
setwd("/Users/steinbeck/Downloads/")
load("Prostate2000Raw.rda")
plot(Prostate2000Raw$mz, Prostate2000Raw$intensity[,1],type = "l", xlab = "m/z", ylab = "response",main
```

Prostatalkrebs-Daten



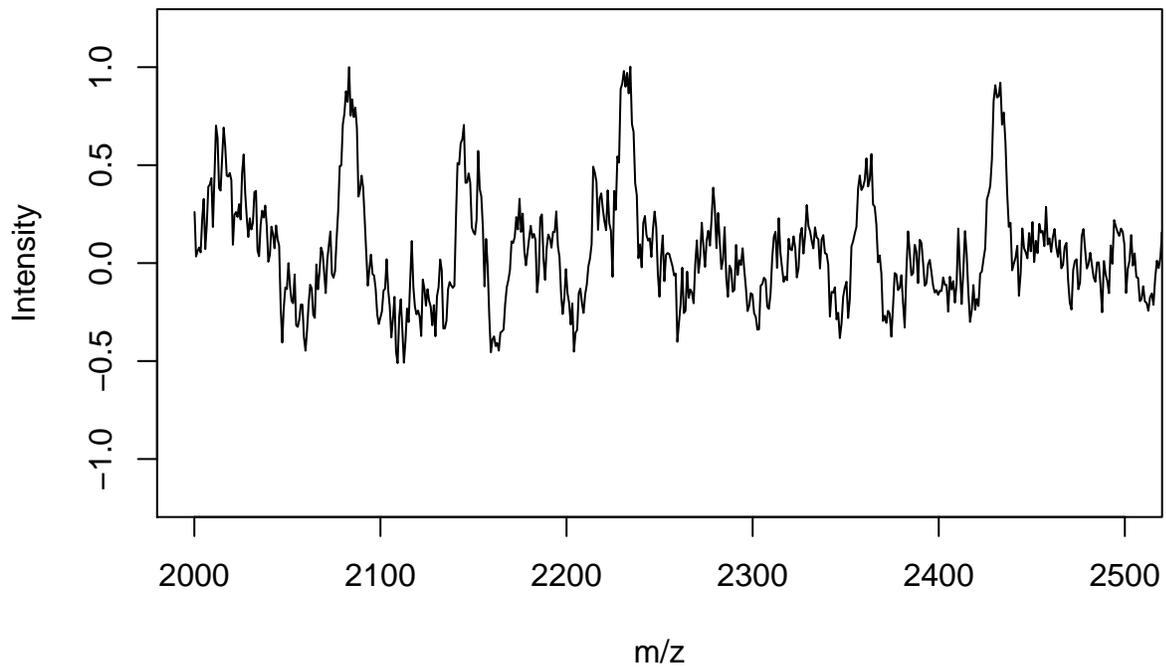
AUF-

GABE: Plotten Sie ein paar weitere Spektren der Serie und vergleichen Sie sie visuell

Lassen Sie uns nun in den verrauschten frühen Teil der Daten zoomen:

```
plot(Prostate2000Raw$zmz, Prostate2000Raw$intensity[,1],type = "l", xlab = "m/z", ylab = "Intensity",mai
```

Prostate data

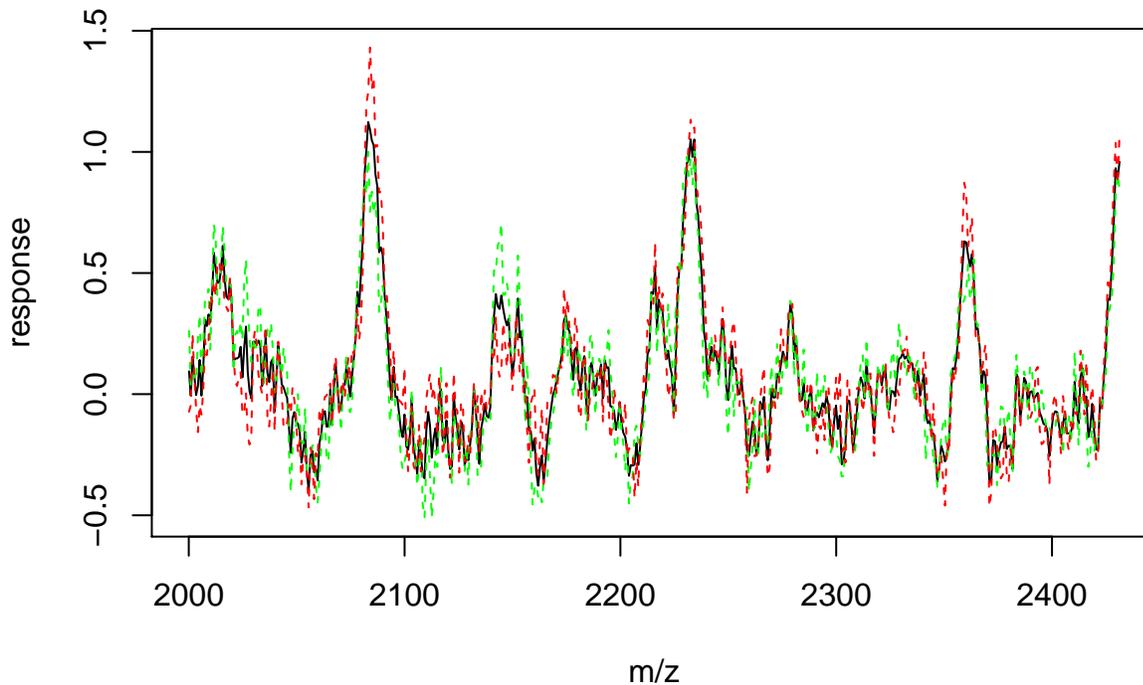


Zunächst wenden wir die Methode der Mittelung über mehrere Messungen unter identischen Bedingungen an,

um das Rauschen zu verringern:

Die Funktion `rowsum()` summiert alle Zeilen, für die die Gruppierungsvariable (das zweite Argument) gleich ist. Da es für jede Probe zwei Replikate gibt, wird das Ergebnis durch zwei geteilt, um die Durchschnittswerte zu erhalten, und in der variablen `Prostate` gespeichert. Für diese neue Variable sollten wir auch die entsprechenden Klassenbezeichnungen im Auge behalten (`prostate.type` Zuweisung)

```
prostate <- rowsum(t(Prostate2000Raw$intensity), group = rep(1:327, each = 2), reorder = FALSE) / 2 # B
prostate.type <- Prostate2000Raw$type[seq(1, 654, by = 2)]
x <- cbind(prostate[1,1:500],Prostate2000Raw$intensity[1:500, 1:2])
matplot(Prostate2000Raw$zmz[1:500], x, type = "l", col = c(1, "green", "red"), lty = c(1,2,2), xlab = "m/z"
```

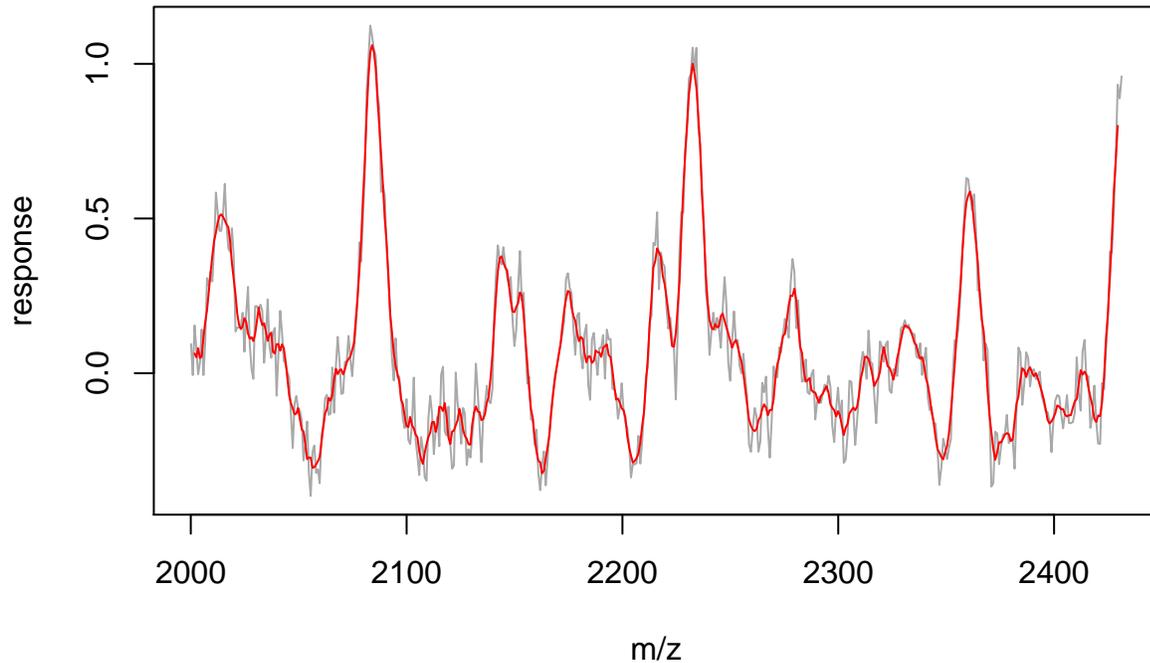


AUFGABE: A) Inspizieren Sie die `rowsum()` Methode und sowie die Rolle der `rep()` Funktion innerhalb dieses Kommandos. B) Was genau macht die `cbind()` Funktion. c) Färben Sie den Mittelwert blau.

Nun lernen wir eine weitere Methode kennen, durch die Kombination der Funktionen `rowMeans()` und `embed()` gleitende Mittelwerte zu erzeugen:

```
rmeans <- rowMeans(embed(prostate[1,1:500], 5))
plot(Prostate2000Raw$zmz[1:500], prostate[1,1:500], type = "l", xlab = "m/z", ylab = "response", main = "r
lines(Prostate2000Raw$zmz[3:498], rmeans, type = "l", col="red")
```

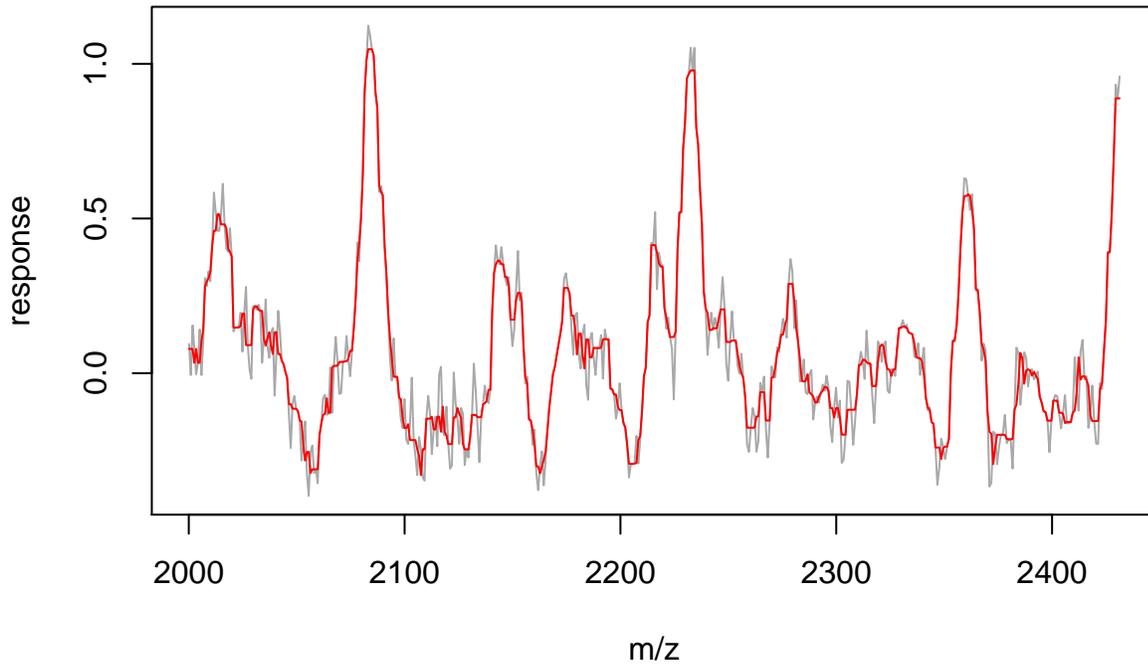
running mean



Wie im obigen Diagramm zu sehen ist, reduziert die Glättung den Rauschpegel effektiv. Beachten Sie, dass die Punkte an den Extremen in dieser Implementierung gesondert behandelt werden müssen. Der zu zahlende Preis ist, dass die Peakhöhen verringert werden, und vor allem bei größeren Spannen wird es zu einer spürbaren Peakverbreiterung kommen. Diesen Effekten kann manchmal durch die Verwendung von gleitenden Medianen anstelle von gleitenden Mitteln entgegengewirkt werden. Die Funktion `runmed()`, Teil des Statistikpakets, ist dafür verfügbar.

```
plot(Prostate2000Raw$mz[1:500], prostate[1,1:500], type = "l", xlab = "m/z", ylab = "response", main = "r  
lines(Prostate2000Raw$mz[1:500], runmed(prostate[1,1:500], k = 5), type = "l", col="red")
```

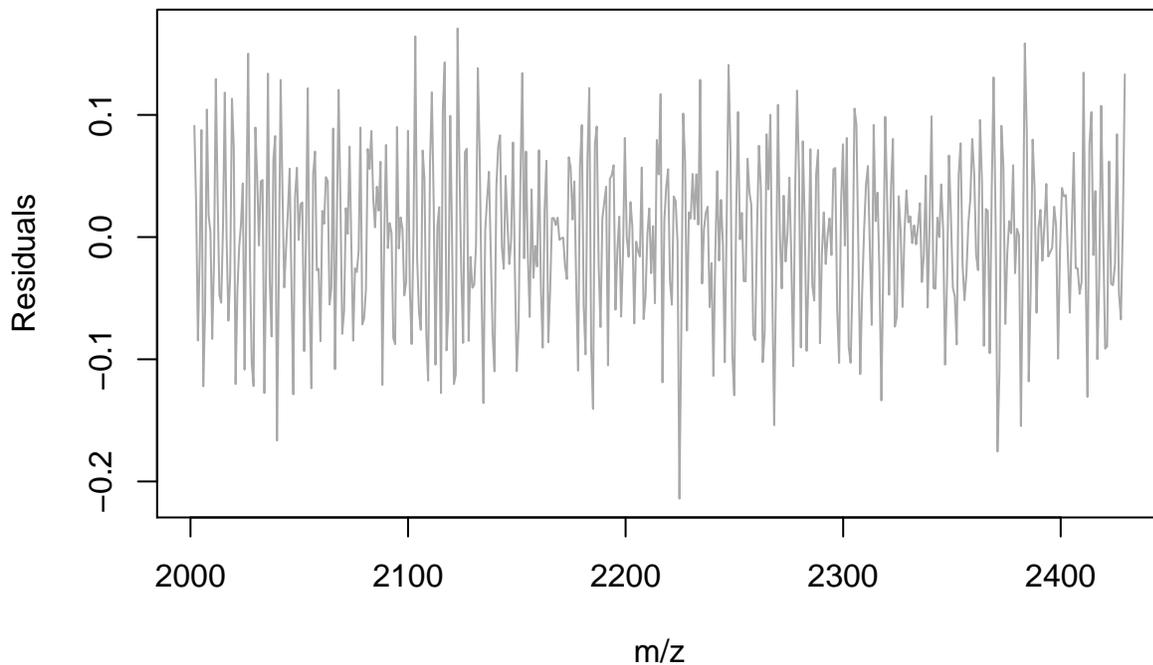
running median



Um nun herauszufinden, wie gut die Glättung funktioniert, berechnen wir die s. g. Residuen, d.h. die Differenz zwischen unserem geglätteten Graphen und den tatsächlichen Werten.

```
rmeans <- rowMeans(embed(prostate[1,1:500], 5))  
residuen <- prostate[1,3:498] - rmeans  
plot(Prostate2000Raw$smz[3:498], residuen ,type = "l", xlab = "m/z", ylab = "Residuals",main = "Running Mean")
```

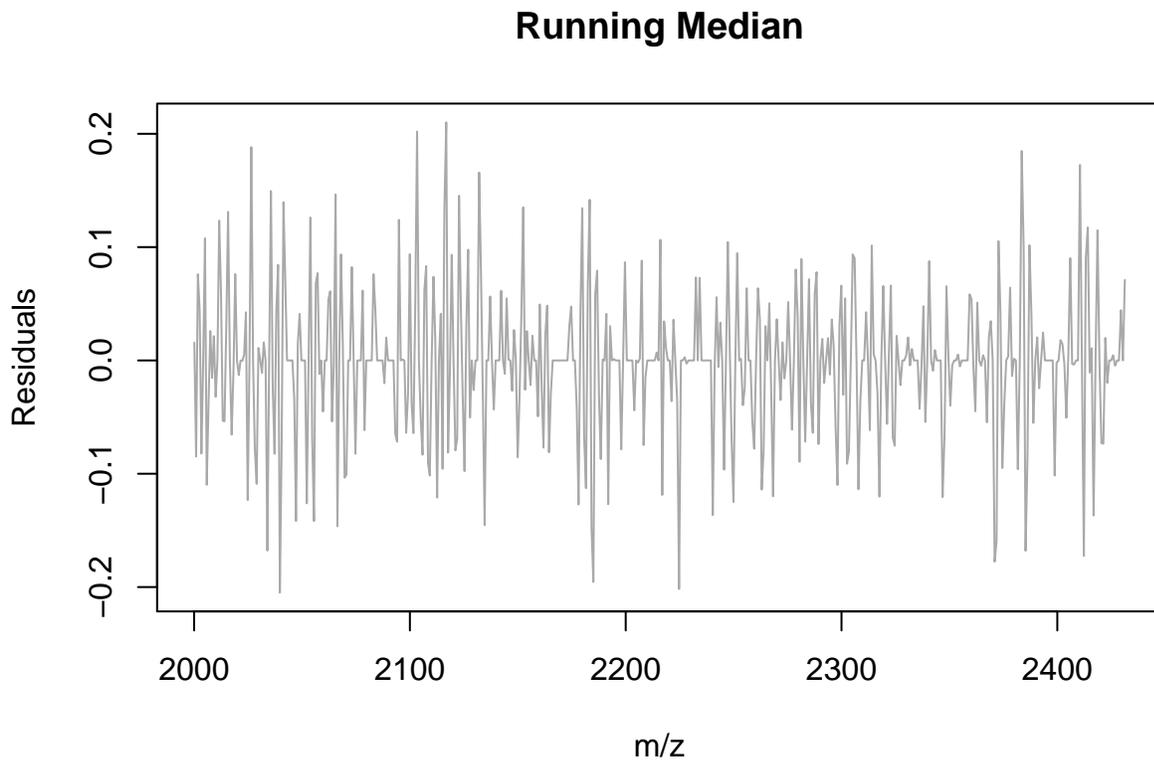
Running Mean



AUFGABE: Fällt Ihnen am Code etwas auf und was ist der Grund für die Abnormalität?

Und das gleich nochmal für den Running Median

```
rmedian <- runmed(prostate[1,1:500], k = 5)
residuen <- prostate[1,1:500] - rmedian
plot(Prostate2000Raw$mz[1:500], residuen, type = "l", xlab = "m/z", ylab = "Residuals", main = "Running Median")
```

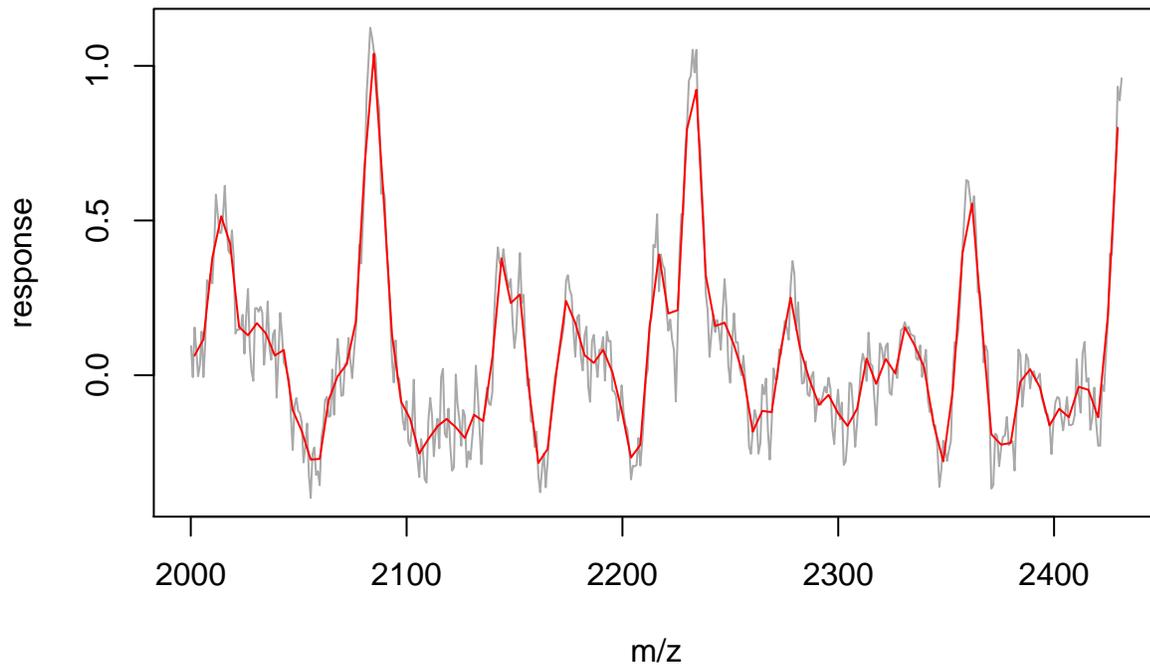


Rauschreduktion mit Binning

Eine besondere Art der Glättung bildet das Binning, auch Bucketing genannt, das nicht nur aufeinanderfolgende Werte mittelt, sondern auch die Anzahl der Variablen verringert.

```
mznew <- colMeans(matrix(Prostate2000Raw$mz[1:500], nrow = 5))
xnew <- colMeans(matrix(prostate[1, 1:500], nrow = 5))
plot(Prostate2000Raw$mz[1:500], prostate[1, 1:500], type = "l", xlab = "m/z", ylab = "response", main = "Binning")
lines(mznew, xnew, col="red")
```

binning



AUFGABE: Experimentieren Sie mit den Bin-Größen und beobachten Sie den Effekt

Rauschreduktion mit Savitzky-Golay Filter

```
library(signal)
```

```
##
```

```
## Attaching package: 'signal'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

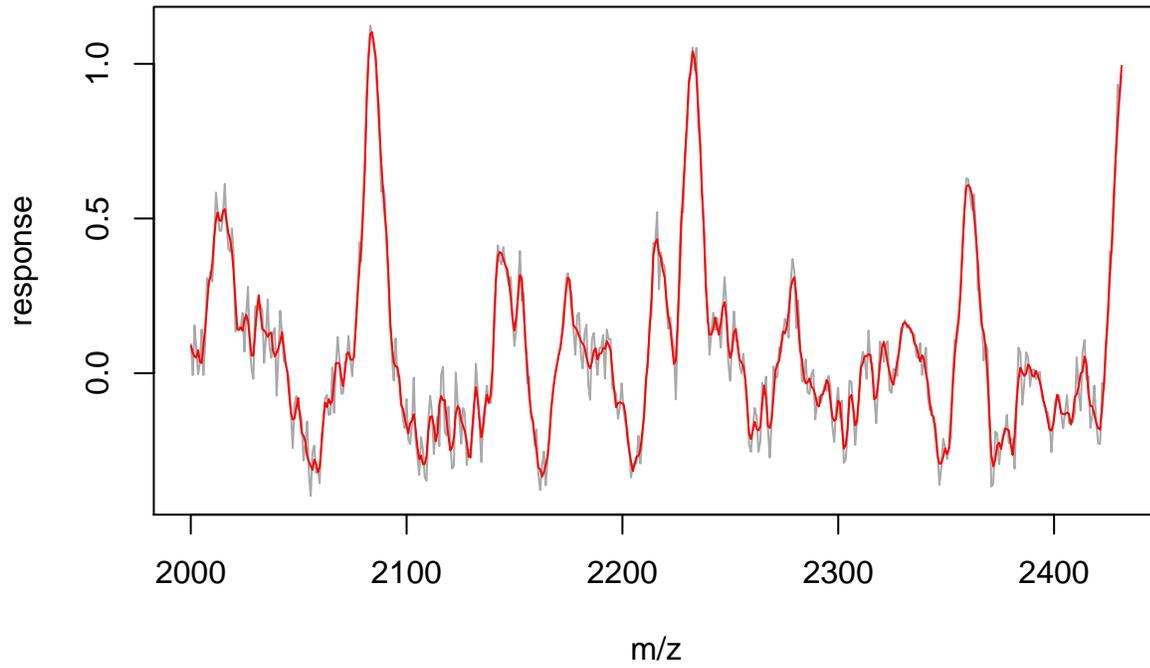
```
## filter, poly
```

```
rsvgo <- sgolayfilt(prostate[1,1:500], p=2, n=7)
```

```
plot(Prostate2000Raw$mz[1:500], prostate[1,1:500], type = "l", xlab = "m/z", ylab = "response", main = "S
```

```
lines(Prostate2000Raw$mz[1:500], rsvgo[1:500], type = "l", col="red")
```

Savitzky-Golay-Filter



AUFGABE: Lesen Sie die Hilfe zum Savitzky-Golay-Filter. Experimentieren Sie mit der Ordnung des Filters (Was sagt die noch gleich?) und der Filter-Länge (Worum geht es da?)