

# Formant-Analysen von Vokalen (2)

Jonathan Harrington

- 1, 2 Struktur und Abbildung einer Trackdatei
- 3, 4. Anwendung von vorhandenen und eigenen Funktionen auf Trackdateien mit `by()`
5. Automatische Identifizierung des Vokal-Targets und Export nach EMU/Praat.

# Vorhandene EMU-R-Objekte

```
data(package="emu")
```

vowlax	Segmentliste
vowlax.fdat	Trackdatei of F1-F4
vowlax.l	Etikettierungen "E" "a" "I" "O"
...	
vowlax.spkr	Sprecher-Etikettierungen "67" "68"

# 1. Struktur einer Trackdatei

Eine Trackdatei funktioniert in vielen Hinsichten wie eine Matrix.

Wieviele Segmente?

`dim(vowlax.fdat)`

Trackdatei, Segmente 2-5

`vowlax.fdat[2:5,]`

Wieviele Segmente?

`nrow(vowlax.fdat)`

Segmente 2-5, nur F2 und F4

`vowlax.fdat[2:5,c(2,4)]`

Wieviele Parameter  
(= wieviele Spalten)?

`ncol(vowlax.fdat)`

## 2. Abbildung einer Trackdatei

Ein Segment: `plot()` oder `dplot()`

Mehrere Segmente aufeinander überlagern: `dplot()`

Abbildung, F2 Segment 11

`plot(vowlax.fdat[11,2])`            oder            `dplot(vowlax.fdat[11,2])`

Abbildung, F1 und F2 Segment 11

`plot(vowlax.fdat[11,1:2])`            oder            `dplot(vowlax.fdat[11,1:2])`

Überlagerte Abbildung F1,  
Segmente 1-20, nach Labels markiert

`dplot(vowlax.fdat[1:20,1], vowlax.l[1:20])`

## 2. Abbildung einer Trackdatei (fortgesetzt)

Abbildung F2 aller Segmente Sprecher 67

# Logischer Vektor auf vowlax.spkr

temp = vowlax.spkr == "67"

# dplot() Abbildung

dplot(vowlax.fdat[temp,2], vowlax.l[temp])

Das gleiche aber **synchronisiert zum zeitlichen Mittelpunkt**

dplot(vowlax.fdat[temp,2], vowlax.l[temp], **offset=0.5**)

Das gleiche aber **pro Kategorie gemittelt**

dplot(vowlax.fdat[temp,2], vowlax.l[temp], **offset=0.5, average=T**)

## 2. Abbildung, Trackdatei, dcut()

Werte zu einem Zeitpunkt

```
dcut(vowlax.fdat, 0.1, prop=T)
```

Alle Werte zwischen zwei Zeitpunkten – zB den mittleren 25% des Vokals

```
vowlax.fdatm = dcut(vowlax.fdat, 0.25, 0.75, prop=T)
```

Abbildung F1 alle Werte  
des 10en Segmentes

```
plot(vowlax.fdat[10,1])
```

Dasselbe aber nur mit  
Werten zwischen den 25%  
und 75% Zeitpunkten.

```
plot(vowlax.fdatm[10,1])
```

Eine Abbildung F2 synchronisiert zum zeitlichen Mittelpunkt aller "E" und "a" Segmente, Sprecher 67, zwischen dem 30% und 70% Zeitpunkt.

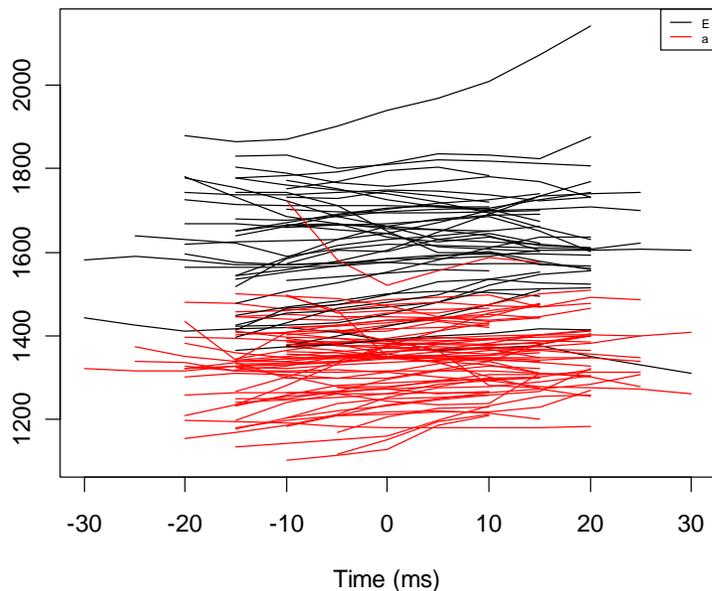
Logischer Vektor um "E" und "a" und Sprecher 67 zu identifizieren  
`temp = vowlax.l %in% c("E", "a") & vowlax.spkr=="67"`

Trackdatei zwischen diesen Zeitpunkten

`neu = dcut(vowlax.fdat[,2], .3, .7, prop=T)`

dplot() Abbildung

`dplot(neu[temp,], vowlax.l[temp], offset=.5)`



vowlax  
vowlax.fdat  
vowlax.l  
vowlax.spkr

Segmentliste  
Trackdatei of F1-F4  
Etikettierungen "E" "a" "I" "O"  
Sprecher-Etikettierungen "67" "68"

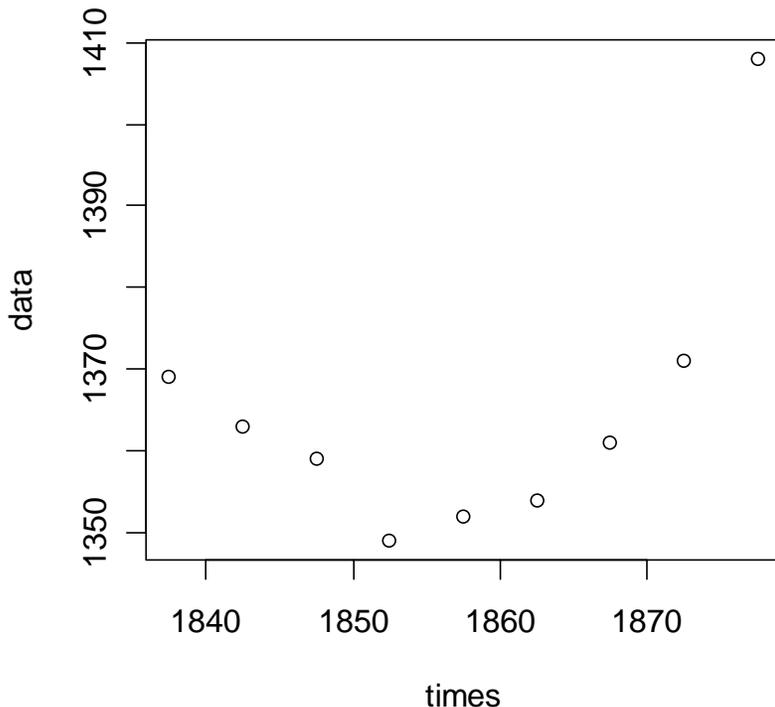
# 3. Trackdateien und die by() Funktion

by() ermöglicht Funktionen auf Trackdateien-Segmente getrennt angewandt zu werden.

**by(trackdatei, Funktion, simplify=T)**

zB Abbildung F2 vom 10en Segment

plot(vowlax.fdat[10,2])



Mittelwert aller dieser F2-Werte

`by(vowlax.fdat[10,2], mean, simplify=T)`

1365.111

Höchstwert

`by(vowlax.fdat[10,2], max, simplify=T)`

Varianz

`by(vowlax.fdat[10,2], var, simplify=T)`

### 3. Trackdateien und die by() Funktion (Fortsetzung)

Trackdatei `vowlax.fdat`

Label-Vektor `vowlax.l`

F1-Maximum Segmente 1-10 (10 Werte)?

`by(vowlax.fdat[1:10,1], max, simplify=T)`

F2-Durchschnitt aller Segmente (410 Werte, ein Wert pro Segment)?

`by(vowlax.fdat[,2], mean, simplify=T)`

F2-Varianz aller "E" Segmente?

Logischer Vektor

`temp = vowlax.l=="E"`

Anwendung von `by()`

`by(vowlax.fdat[temp,2], var, simplify=T)`

### 3. Trackdateien und die by() Funktion (Fortsetzung)

F2-Zentralwert aller "E" Segmente zwischen den 30% und 70% Zeitpunkten ?

# logischer Vektor

temp = vowel.l=="E"

# Trackdatei zwischen diesen Zeitpunkten

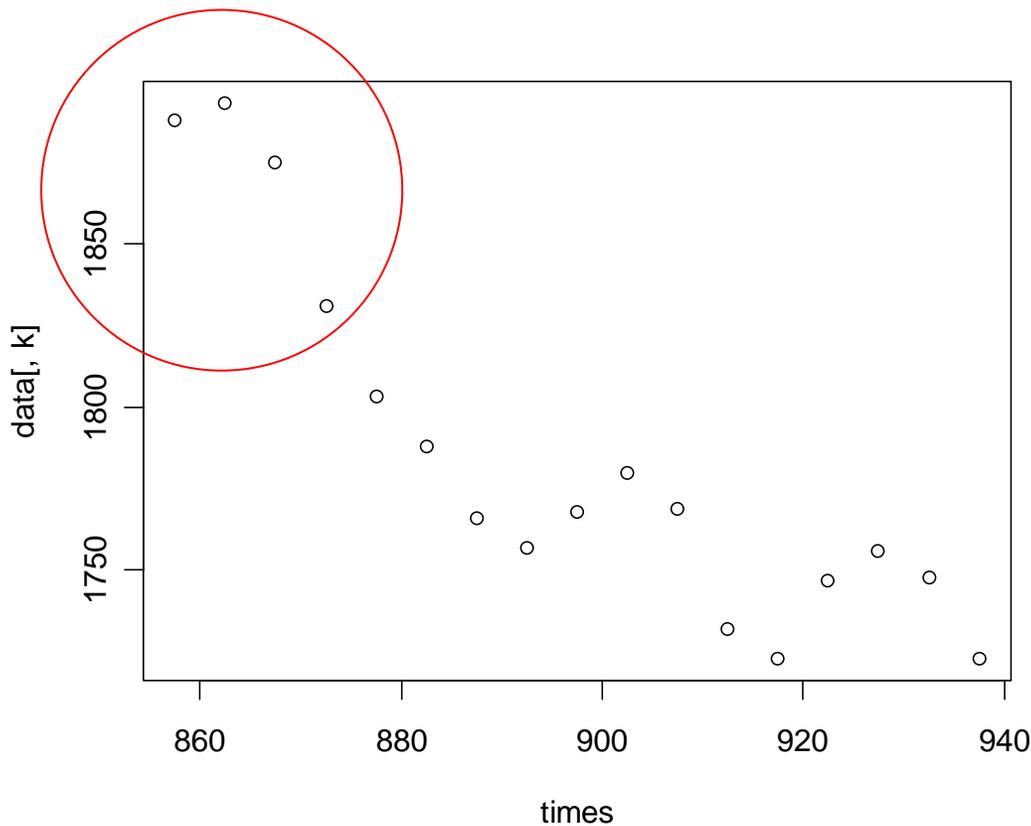
neu = dcut(vowel.fdat[temp,2], .3, .7, prop=T)

# Anwendung von by()

by(neu, median, simplify=T)

## 4. by() und die Anwendung eigener Funktionen

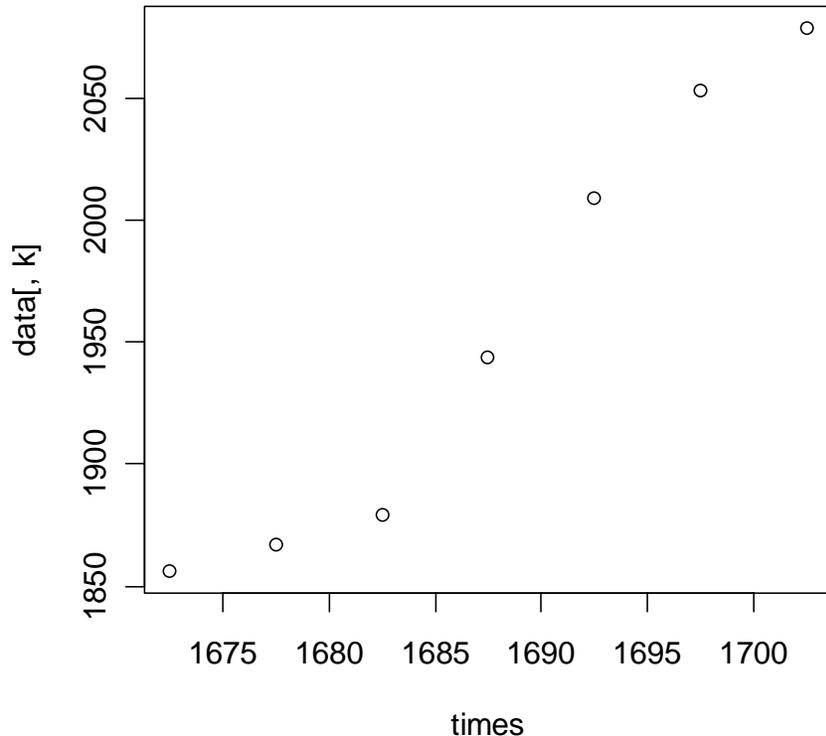
Das Problem: wir wollen die ersten vier F2-Werte pro Segment mitteln



Durchschnitt  
dieser 4 Werte:  
1871.75 Hz

Die tatsächlichen Werte in einer Trackdatei sind in \$data

Abbildung F2 des ersten Segmentes: deren Werte:  
`plot(vowlax.fdat[4,2])` `vowlax.fdat[4,2]$data`



**T2**

<b>1672.5</b>	<b>1856</b>
<b>1677.5</b>	<b>1867</b>
<b>1682.5</b>	<b>1879</b>
<b>1687.5</b>	<b>1944</b>
<b>1692.5</b>	<b>2009</b>
<b>1697.5</b>	<b>2053</b>
<b>1702.5</b>	<b>2079</b>

## 4. by() und Anwendung eigener Funktionen

	T2
1672.5	1856
1677.5	1867
1682.5	1879
1687.5	1944
1692.5	2009
1697.5	2053
1702.5	2079

```
f2 = vowlax.fdat[4,2]$data
```

```
# Durchschnitt der ersten 4 Werte davon?
```

```
mean(f2[1:4])
```

## 4. by() und Anwendung eigener Funktionen

Funktion schreiben, `mfun()`, um den Durchschnitt der ersten 4 Werte irgendeines Vektors zu berechnen.

```
a = 1:10
```

```
mfun(a)
```

```
[1] 2.5
```

```
mfun <- function(vec)
{
mean(vec[1:4])
}
```

Durchschnitt der ersten 4 F1-Werte vom 12en Segment?

```
mfun( vowlax.fdat[12,1]$data )
```

## 4. by() und Anwendung eigener Funktionen

Durchschnitt der ersten 4 F2-Werte aller Segmente?

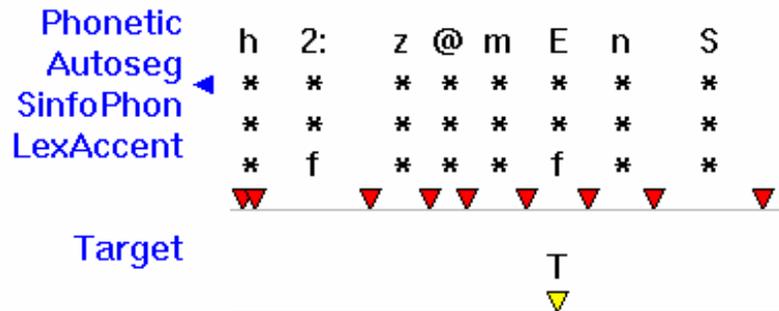
[by\(vowlax.fdat\[,2\], mfun, simplify=T\)](#)

Durchschnitt der ersten 4 F1-Werte Segmente 11 bis 30?

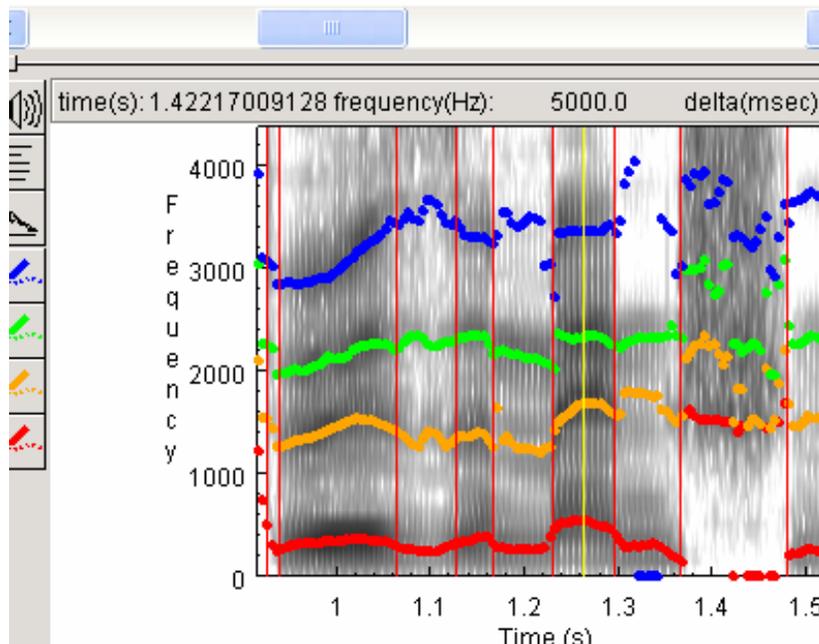
[by\(vowlax.fdat\[11:30,1\], mfun, simplify=T\)](#)

## 5. Vokalzielposition automatisch markieren

T in eine neue Target-Ebene zum Zeitpunkt des F1-Maximums setzen.



**targ()** Eine neue Funktion schreiben, um diesen Zeitpunkt in einem beliebigen Segment zu identifizieren



**by()** Die neue Funktion auf alle Segmente anwenden.

**makelab()** daraus EMU-Labeldateien erzeugen ggf. nach Praat exportieren.

# F1-Werte, Segment 10?

`f1 = vowlax.fdat\[10,1\]\$data`

`f1`

`T1`

```
1837.5 697
1842.5 698
1847.5 686
1852.5 650
1857.5 649
1862.5 641
1867.5 653
1872.5 645
1877.5 640
```

`tracktimes()` : Die Zeitpunkte, zu denen Trackdateien-Werte vorkommen.

`zeiten = tracktimes(f1)`

`zeiten`

```
[1] 1837.5 1842.5 1847.5 1852.5
1857.5 1862.5 1867.5 1872.5
1877.5
```

1. F1-Maximum Segment 10 in `m` speichern

`m = max(f1)`

`m`

```
[1] 698
```

2. Logischer Vektor um den F1-Maximum zu identifizieren

`temp = f1 == m`

3. Die Zeit, zu der `m` vorkommt

`zeiten[temp]`

4. Für den Fall, dass es mehr als ein Maximum gibt, nehmen wir **den ersten** davon

`zeiten[temp][1]`

Die 4 Schritte in eine Funktion `targ()` packen, um auf eine beliebige Matrix anzuwenden.

werte

	T1	T2	T3	T4
1672.5	347	1856	2316	3406
1677.5	379	1867	2354	3357
1682.5	383	1879	2336	3296
1687.5	368	1944	2362	3275
1692.5	361	2009	2364	3270
1697.5	343	2053	2400	3291
1702.5	235	2079	2471	3373

`targ(werte[,1])`

[1] 1682.5

`targ(werte[,3])`

[1] 1702.5

```
targ <- function(werte)
{
zeiten = tracktimes(werte)
temp = werte == max(werte)
zeiten[temp][1]
}
```

	T1
1837.5	697
<b>1842.5</b>	<b>698</b>
1847.5	686
1852.5	650
1857.5	649
1862.5	641
1867.5	653
1872.5	645
1877.5	640

Diese Funktion auf die F1 Werte des  
10en Segmentes anwenden

```
targ( vowlax.fdat[10,1]$data )
```

```
[1] 1842.5
```

Wie können wir **für jedes Segment** den Zeitpunkt des F1-Maximums berechnen?

```
f1zeit = by(vowlax.fdat[,1], targ, simplify=T)
```

Das gleiche aber zwischen dem 30% und 70% Zeitpunkt.

```
# Trackdatei schneiden
```

```
neu = dcut(vowlax.fdat[,1], .3, .7, prop=T)
```

```
# Zeiten des F1-Maximums (eine Zeit pro Segment)?
```

```
f1zeit = by(neu, targ, simplify=T)
```

Die ersten 10 Segmente überprüfen:

```
for(j in 1:10){  
  plot(vowlax.fdat[j,1], type="b")  
  abline(v=f1zeit[j], col=2)  
  locator(1)  
}
```

**Mit der linken Maustaste klicken, um zur nächsten  
Abbildung zu gelangen!**

# Etikettierungen + Zeiten nach Emulabel/Praat exportieren.

**makelab(zeiten, Äußerungs-namen, verzeichnis)**

- die Zeiten (eine Zeit pro Segment)

f1zeit

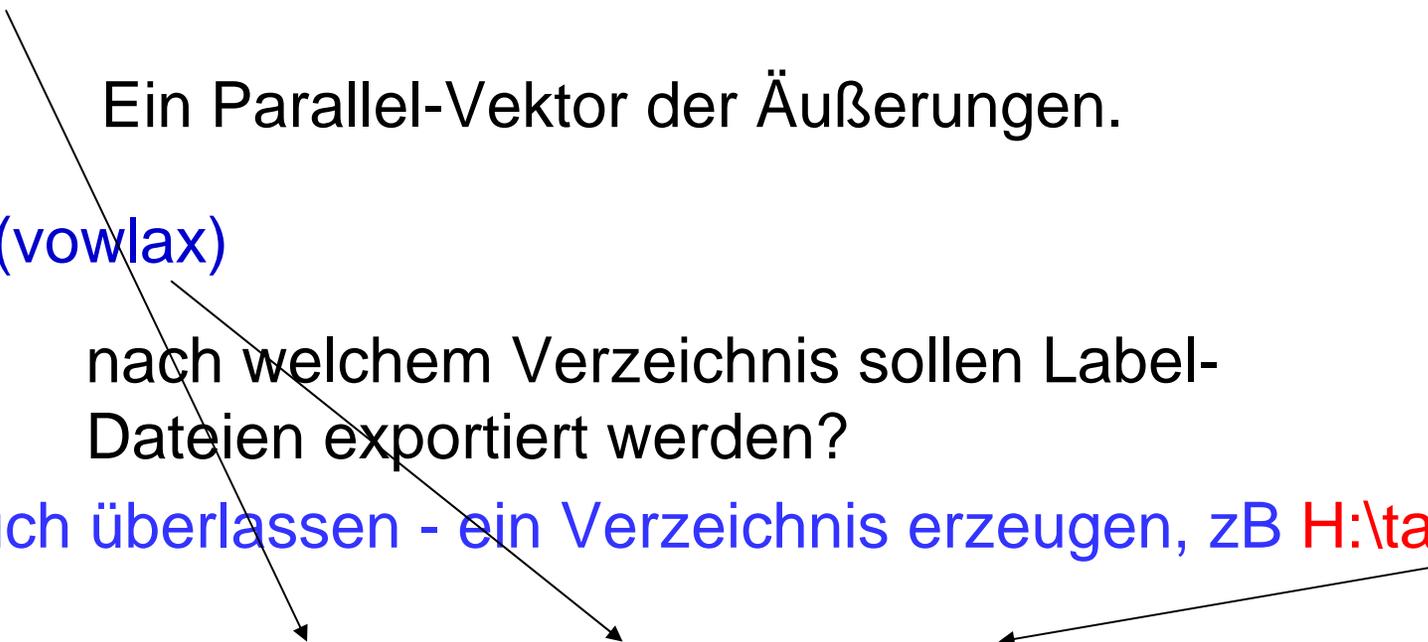
- Ein Parallel-Vektor der Äußerungen.

utt(vowlax)

- nach welchem Verzeichnis sollen Label-Dateien exportiert werden?

(Euch überlassen - ein Verzeichnis erzeugen, zB **H:\targlabs**)

zB: makelab(f1zeit, utt(vowlax), "**H:/targlabs**")



Bitte dann die Template-Datei umbenennen,  
und dann ändern, damit die exportierten  
Labels in einer Emulabel Event-Ebene  
sichtbar sind.