

# Abbildungen in EMU-R

dplot eplot hplot

Jonathan Harrington & Tina John

# dplot()

Eine Abbildung eines Parameters als Funktion der Zeit

Segmentliste aller [e:] und [a:] Vokale (Sprecher "gam" der seminar03 Sprachdatenbank)

```
s.ea = emu.query("seminar03", "gam*", "[Phonetic=e: | a:]")
```

Etikettierungen

```
l.ea = label(s.ea)
```

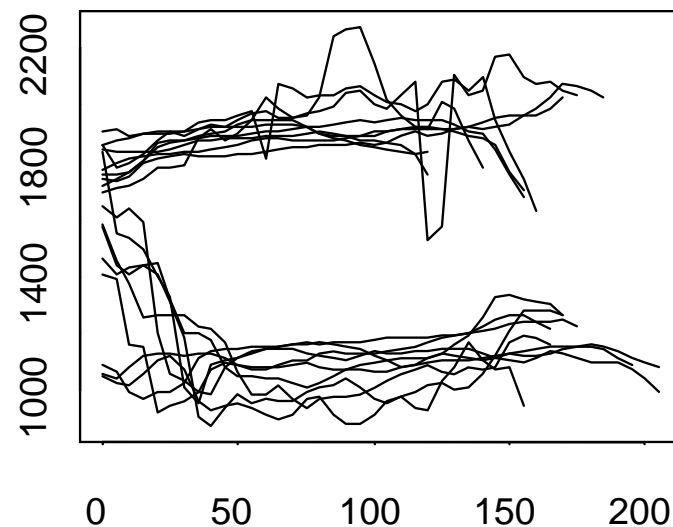
Formanten

```
fvals = emu.track(s.ea, "fm")
```

Abbildung von F2 aller Segmente

```
dplot(fvals[,2])
```

↑  
Werte in einer Trackdatei



# dplot() mit Trennung der Laute

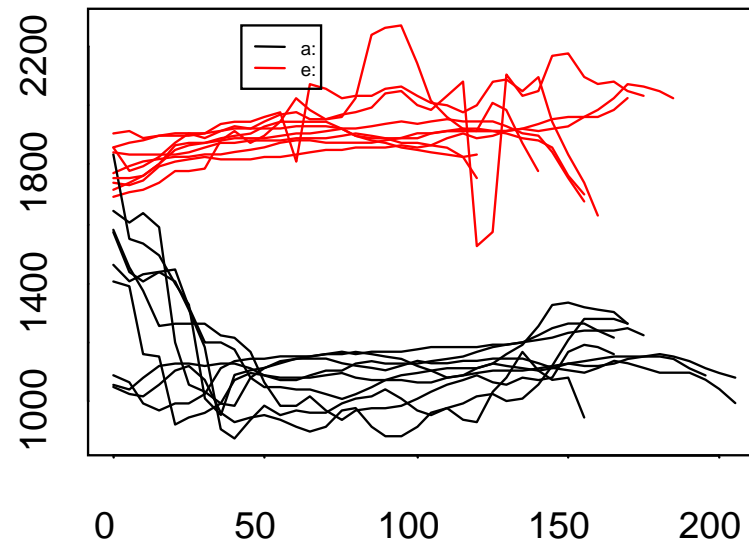
dplot(TRACKWERTE, **labelvektor**)

```
s.ea = emu.query("seminar03", "gam*", "[Phonetic=e: | a:]")  
l.ea = label(s.ea)  
fvals = emu.track(s.ea, "fm")
```

Abbildung von F2 aller Segmente **nach Lauten getrennt**

Werte  
↓  
dplot(fvals[,2], **l.ea**)

Labelvektor, der die Labels  
der Segmentliste enthält für  
welche auch die Trackdatei  
erzeugt wurde



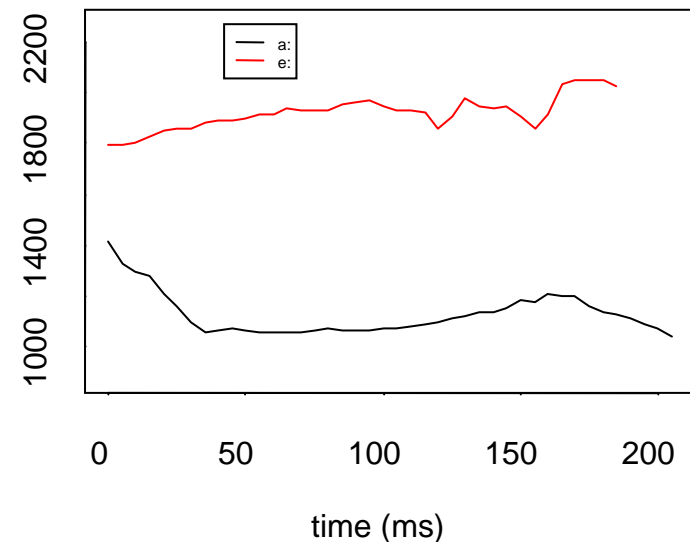
# dplot() Durchschnitt der Signale

dplot(TRACKWERTE , labelvektor, **average=T**)

```
s.ea = emu.query("seminar03", "gam*", "[Phonetic=e: | a:]")  
l.ea = label(s.ea)  
fvals = emu.track(s.ea, "fm")
```

Abbildung vom **Durchschnitt** der F2 Werte aller Segmente nach Lauten  
getrennt

Werte    Labelvektor  
    ↙     ↘  
dplot(fvals[,2], l.ea, **average=T**)  
          ↖  
    **Durchschnitt der Trackwerte  
    an den verschiedenen  
    Zeitpunkten**



Ohne Labelvektor würde der Durchschnitt über alle Segmente berechnet werden, was auch funktioniert.

# dplot() für eine Auswahl an Segmenten

Segmentliste aller [i: u: a:] Vokale (Sprecher "agr" der seminar04 Sprachdatenbank)

```
segs = emu.query("seminar04", "agr*", "[Phonetic= i: | u: | a:]")
```

Etikettierungen

```
l.segs = label(segs)
```

Formanten

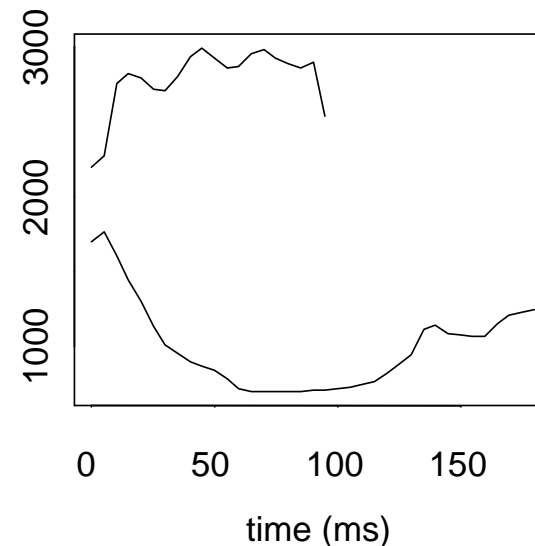
```
fvals = emu.track(segs, "fm")
```

F2 (alle Werte) vom zweiten und vierten Segment?

```
fvals[c(2,4),2]
```

Eine Abbildung davon

```
dplot(fvals[c(2,4),2])
```



# dplot() Synchronisierung der Signale

dplot(TRACKWERTE , **offset = 0-1**)

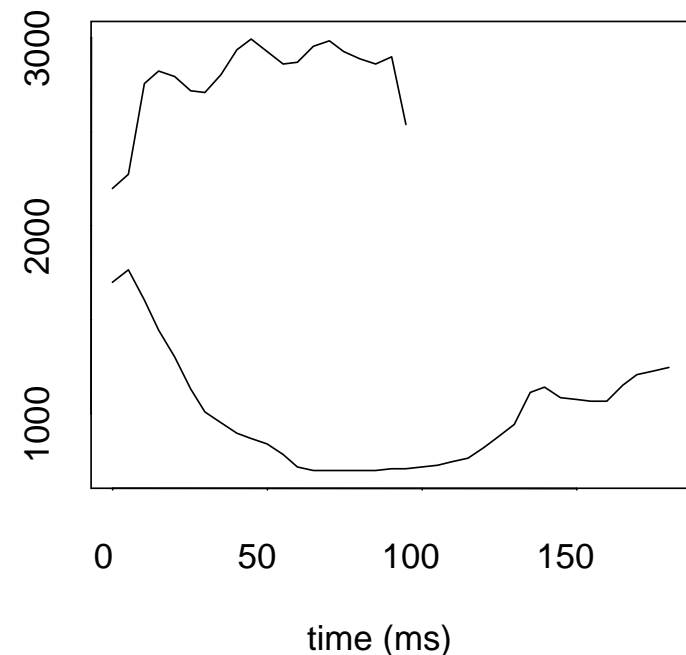
```
segs = emu.query("seminar04", "agr*", "[Phonetic= i: | u: | a:]")  
l.segs = label(segs)  
fvals = emu.track(segs, "fm")
```

Abbildung der F2 Werte der Segmente 2 und 4 **synchronisiert zum Segmentbeginn**

Werte

dplot(fvals[c(2,4),2], **offset = 0**)

**Synchronisierung der Signale der verschiedenen Segmente zu einem Zeitpunkt (hier der Segmentonset) – default Einstellung**



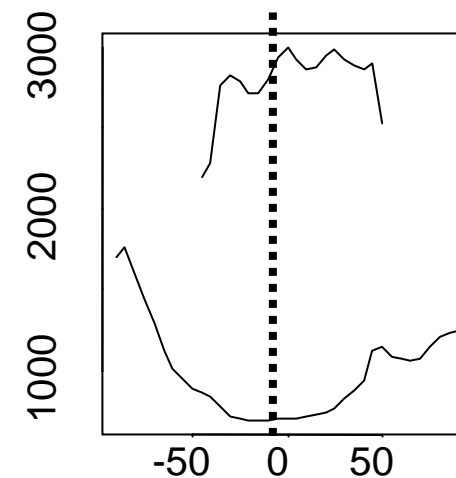
# dplot() Synchronisierung der Signale

dplot(TRACKWERTE, **offset = 0-1**)

```
segs = emu.query("seminar04", "agr*", "[Phonetic= i: | u: | a:]")  
l.segs = label(segs)  
form = emu.track(segs, "fm")
```

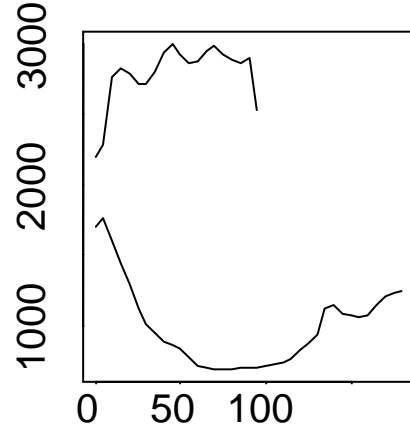
Abbildung der F2 Werte der Segmente 2 und 4 **synchronisiert zum zeitlichen Mittelpunkt**

Werte  
↓  
dplot(form[c(2,4),2], **offset = 0.5**)  
↑  
**Zeitlicher Mittelpunkt**



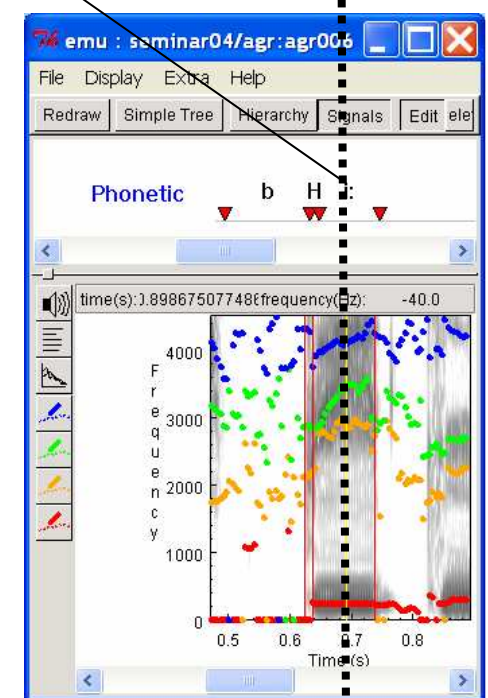
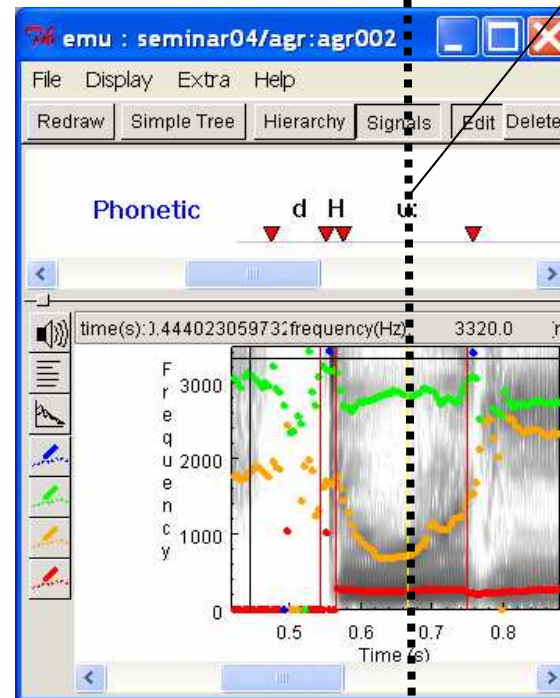
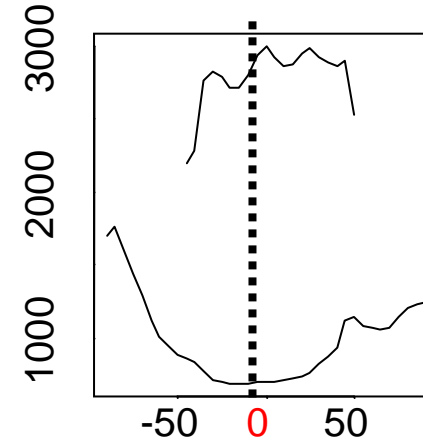
# Synchronisierungszeitpunkt Segmentbeginn (default)

```
dplot(form[c(2,4),2])
```



# Synchronisierungszeitpunkt Mittelpunkt

```
dplot(form[c(2,4),2], offset=0.5)
```





# eplot()

Eine Abbildung eines Parameters über einen anderen Parameter

```
segs = emu.query("seminar04", "agr*", "[Phonetic= i: | u: | a:]")
l.segs = label(segs)
fvals = emu.track(segs, "fm")
```

F1 x F2 Ellipsen von [i: u: a:] zum zeitlichen Mittelpunkt.

F1-F4 zum zeitlichen Mittelpunkt

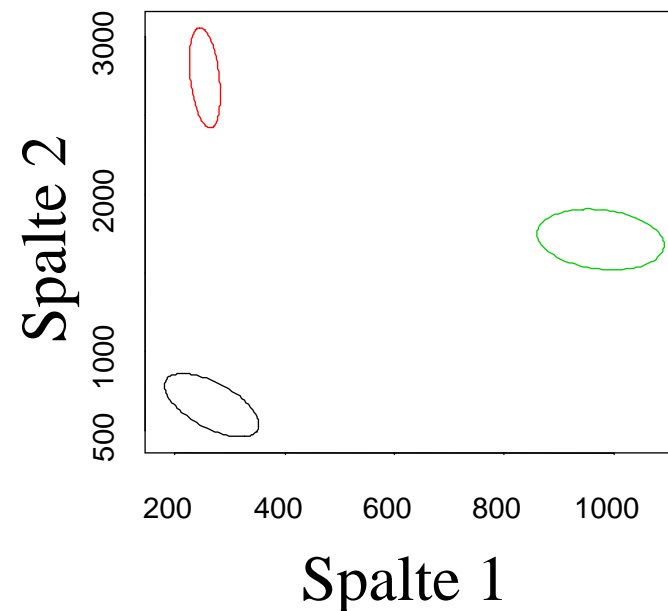
fvalsmitte = dcut(fvals, .5, prop=T)

eplot(fvalsmitte[,1:2], l.segs)

Eine Matrix von 2 Spalten

dazu passende Etikettierungen

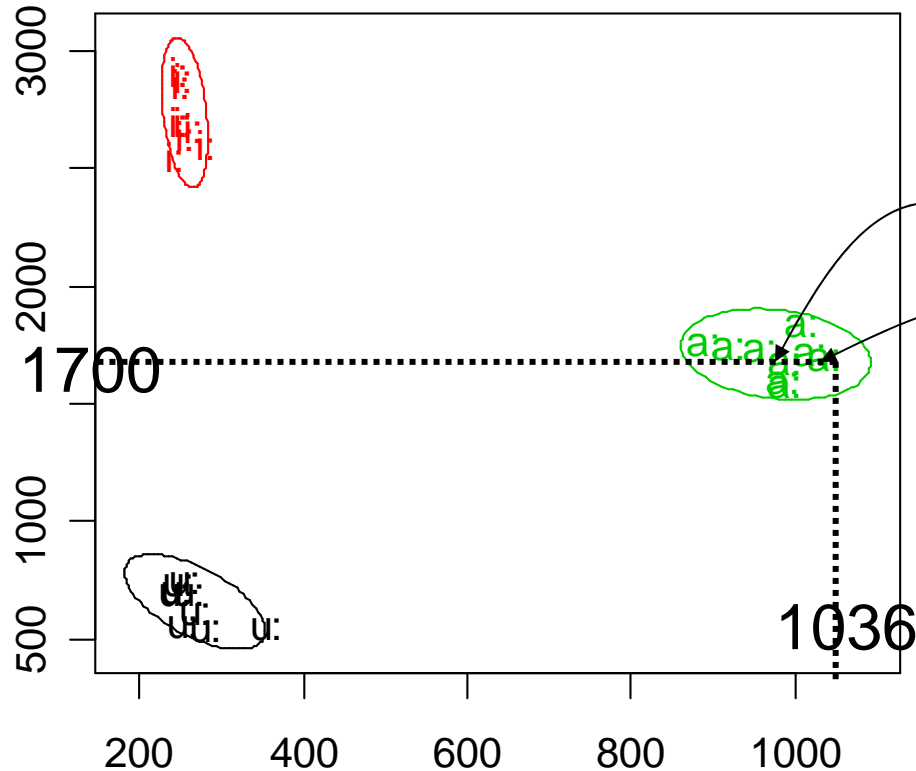
Jede Ellipse schließt 95% der Werte ein



# eplot() mit Abbildung der tatsächlichen Werte

```
eplot(formmitte[,1:2],l.segs, dopoints=T)
```

die tatsächlichen Werte abbilden



Die F1 und F2 Werte von [a:]

```
> temp = l.segs=="a:"
```

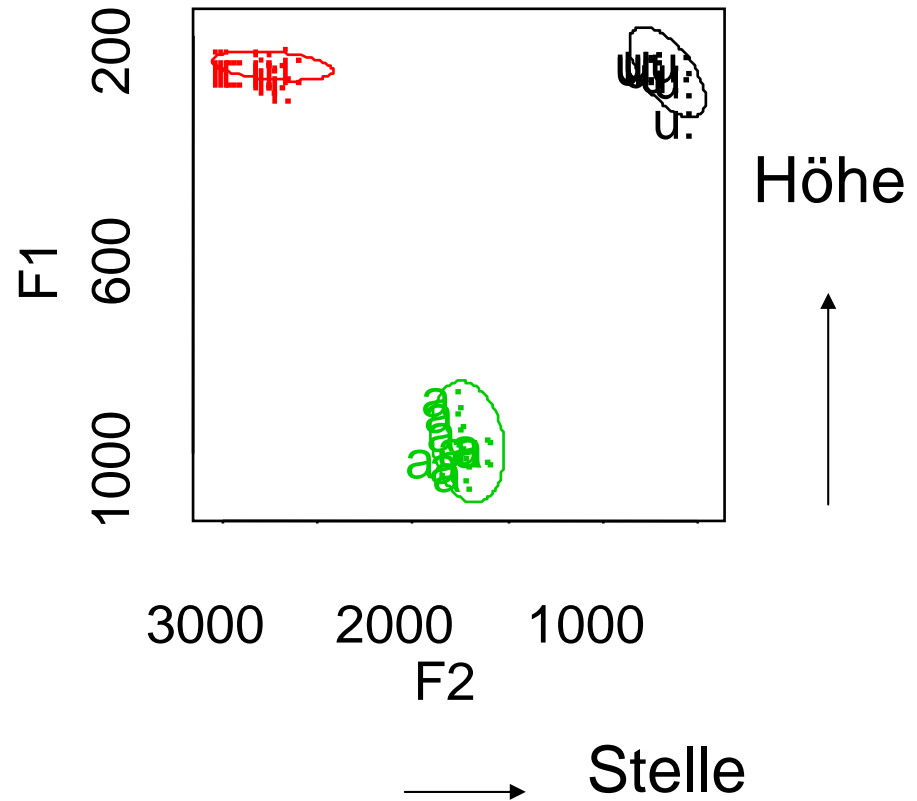
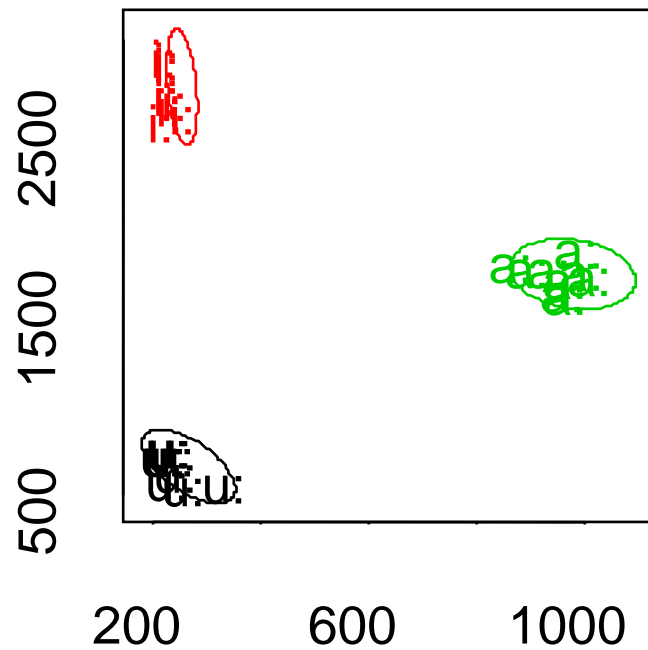
```
> form5[temp,1:2]
```

	V3	V4
181	989	1684
244	985	1607
307	1036	1700
413	1020	1721
454	1009	1845
495	920	1744
620	890	1765
727	957	1741
822	988	1588

# eplot() transponieren (drehen) der Achsen

```
eplot(formmitte[,1:2],1.segs, dopoints=T, form=T)
```

Dreht die Achsen



# eplot() Beispiel

Ellipse-Abbildung: Dauer-Werte (x-Achse) und F1 (y-Achse)

## Segmentliste von [i: u: a:] Vokalen

```
> segs = emu.query("seminar04", "agr*", "Phonetic=i: | u: | a:")
```

## Etikettierungen davon

```
> l.segs = label(segs)
```

## Trackdatei F1-F4

```
form = emu.track(segs, "fm")
```

## F1-F4 zum zeitlichen Mittelpunkt

```
formmitte = dcut(form, .5, prop=T)
```

## Dauer

```
d = mudur(segs)
```

## Verbinden der Parameter zu einer Matrix

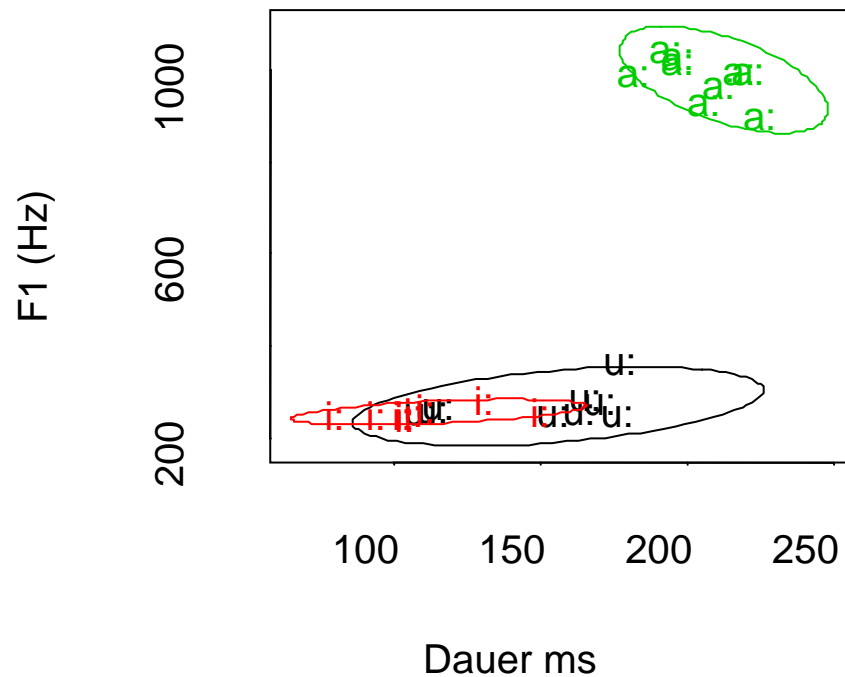
```
mat = cbind(d, formmitte[,1])
```

mat ist nun eine zweispaltige Matrix der Dauerwerte (Spalte 1)  
und F1 zum zeitlichen Mittelpunkt (Spalte 2)

# eplot() Beispiel

Plotten der mat Matrix mit dazugehörigen Labelvektor

```
eplot(mat, l.segs, dopoints=T, xlab="Dauer ms", ylab="F1 (Hz)")
```



# Zusammenfassung dplot() eplot()

`dplot(TRACKDATEI , optionen )`

Eine Abbildung eines Parameters als Funktion der Zeit

`optionen`

Labelvektor – trennt Lautkategorien

`offset=ZAHL` – zeitliche Synchronisierung

`average=T|F` – wenn T - Durchschnitt der einzelnen Kategorien

`eplot(2DIMMATRIX , labels , optionen )`

Abbildung eines Parameters über einem anderen als Ellipsen

`optionen`

`dopoints = T` – zeigt die labels an den absoluten Werten

`form = T` – transponiert die Achsen des plots

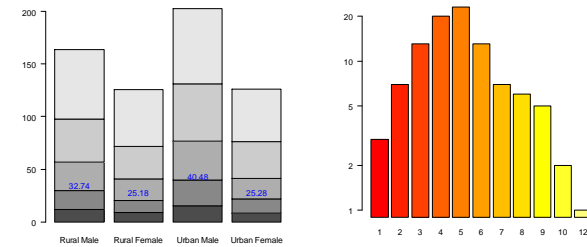
Weitere Optionen für die verschiedenen Plots

tippe `help(PLOT)` in R, PLOT jeweils der gesuchte Plottyp

# weitere plot-Funktionen

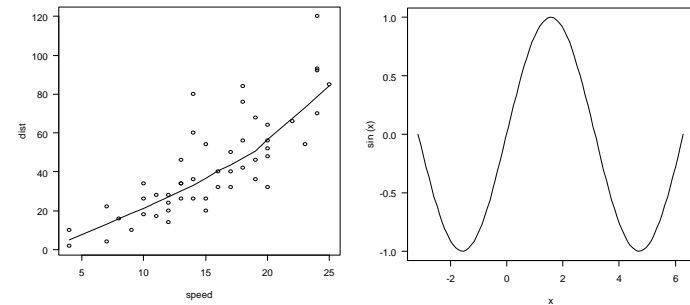
`barplot(WERTEVEKTOR, names.arg = LABELVEKTOR,...)`

Säulendiagramm



`plot(WERTEx,WERTEy, ...)`

Funktionsplot



`boxplot(WERTEVEKTOR~LABELVEKTOR,...)`  
box-and-whisker plot

