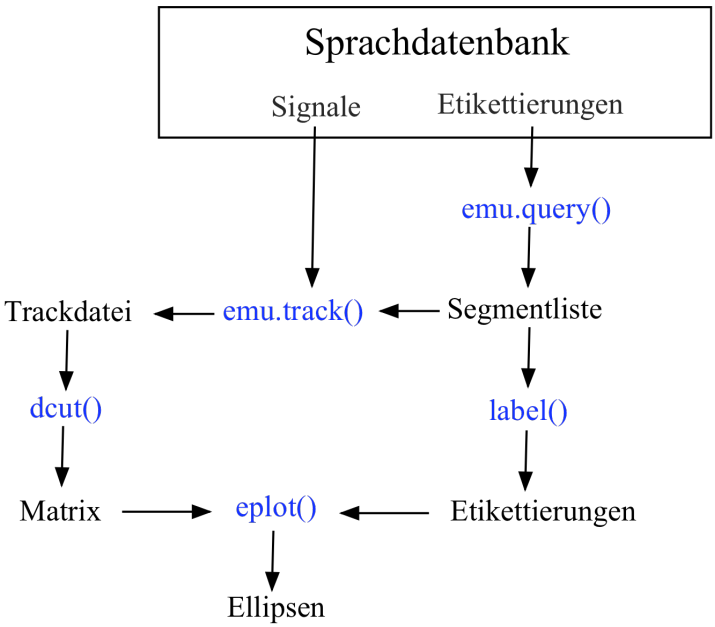
Ellipse-Abbildungen von Formanten in R



`second`: Name der Sprachdatenbank
`gam*`: Nur in den Äußerungen suchen, die mit *gam* beginnen
`Phonetic`: Ebene, aus der die Segmente entnommen werden

Prüfen welche Signale vorhanden sind ("fm" (Formantsignale) muss für diese Übung vorhanden sein). Formanten mit Emu berechnen...

```
library(emu)
trackinfo("second")
[1] "samples" "fm"
```

`i:|e:|a:|o:|u:` : "i:", oder "e:", oder "a:", oder "o:", oder "u:" Vokale dieser Ebene entnehmen

```
vok.s = emu.query("second", "gam*", "Phonetic=i:|e:|a:|o:|u:")
```

Ein Vektor nur mit den Vokaletikettierungen

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

Eine sogenannte Trackdatei. vok.fm enthält alle Formantwerte, F1-F4, zwischen den Start- und Endzeiten aller Segmente in vok.s

```
vok.fm = emu.track(vok.s,  "fm")
```

Prüfen, dass wir eine Trackdatei haben

```
class(vok.fm)
is.trackdata(vok.fm)
```

Trackdateien lassen sich mehr oder weniger wie eine Matrix behandeln:

```
dim(vok.fm)
nrow(vok.fm)
# Formanten vom zehnten Segment
vok.fm[10,]
```
usw.

Formant-Abbildungen z.B. F2 vom zehnten Segment
```
plot(vok.fm[10,2], type="l")
```
Die tatsächlichen Formantwerte bekommt man mit der frames() Funktion; die Zeiten, zu denen die Werte vorkommen mit tracktimes()

```
frames(vok.fm[10,])
```

sind die Formantwerte von diesem Segment
```
vok.s[10,]
```

Die Zeiten, du denen diese Formantwerte vorkommen:
```
tracktimes(vok.fm[10,])
```

F2 von allen /e:/ Vokalen
```
temp = vok.l == "e:"
dplot(vok.fm[temp,2])
```
F2 von allen /e:/ und /o:/ Vokalen
```
temp = vok.l %in% c("i:", "u:")
dplot(vok.fm[temp,2], vok.l[temp])
```

Mit dem Befehl dcut() entnehmen wir der Trackdatei F1 und F2 zum zeitlichen Mittelpunkt von jedem Vokal (prop bedeutet: die Zeiten werden *proportional* in der Zeit entnommen)

```
vok.fm5 = dcut(vok.fm[,1:2], .5, prop=T)
```

Eine Ellipse-Abbildung im Raum F1 und F2. form=T dreht die Achsen, damit F2 auf der *x*-Achse, F1 auf der *y*-Achse erscheinen, und sodass die Werte von links nach rechts (F2) und von unten nach oben (F1) fallen.

```
eplot(vok.fm5, vok.l, form=T, dopoints=T)
```

Nehmen wir an, wir möchten die Äußerung(en) mit /a:/ finden, in denen F1 höher als 650 Hz ist. (Diese Methode werden wir ggf. später benötigen, um Äußerungen zu finden, in denen Formantfehler auftreten)

Ein logischer Vektor. T wenn F1 > 650 Hz und die Etikettierung ist "a:"
```
temp = vok.fm5[,1] > 650 & vok.l == "a:"
vok.s[temp,]
```
```
segment  list from database:  second
query was:  Phonetic=i:|e:|a:|o:|u:
   labels   start     end   utts
14     a: 473.912 647.072 gam023
```

```
20      a: 500.450 677.354 gam031
```

Aufgabe: Bitte jetzt das gleiche für die Sprecherin **agr** wiederholen. In tkassp Nominal F1 auf 600 Hz setzen und ggf. Formanten korrigieren.