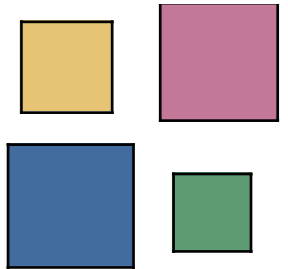


iPlots Anwendungen

- Hauptanwendung von **iPlots** ist der Einsatz von Linked-Highlighting (und anderen interaktiven Features) in der Datenanalyse
Dort sind andere Programme wie **DataDesk** oder **Mondrian** jedoch leistungsfähiger
- Die besonderen Vorteile von **iPlots** ergeben sich erst im Zusammenspiel mit **R**
- Hauptpunkte:
 - Ergänzung der interaktiven Plots um Linien und Text
 - Setzen und Abfragen der Selektion
 - Dynamisches Erstellen von Modellen



iPlots Anwendungen: Ergänzen von Objekten

- Dichteschätzung in `ihist(...)`
- Problem
 - `ihist` unterstützt die Option `freq=[F|T]` nicht

- Lösung:

```
# create histogram
```

```
ihist(Tip.in...of.Bill)
```

```
# fix bin width to a known (sensible) value
```

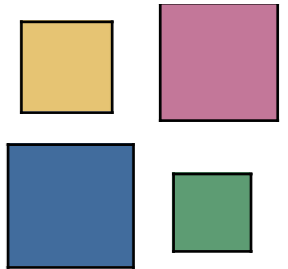
```
iplot.opt(breaks=(0:10)/20)
```

```
# create density estimate
```

```
d <- density(Tip.in...of.Bill)
```

```
# plot scaled lines
```

```
ilines(d$x, d$y*length(Tip.in...of.Bill)*0.05)
```



iPlots Anwendungen: Arbeiten mit der Selektion

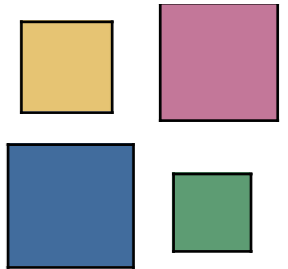
- Erstellen eines Lowess Glätters für eine best. Untermenge
- Lösung

```
# setup plots
iplot(Total.Bill.in.USD, Tip.in.USD)
ihist(Tip.in.perc.of.Bill)
ibar(Sex); ibar(Smoker); ibar(Size.of.Party)

# select scatterplot
iplot.set(1)

# set lowess smoother
l  <- lowess(Total.Bill.in.USD, Tip.in.USD)
ilines(l$x, l$y)

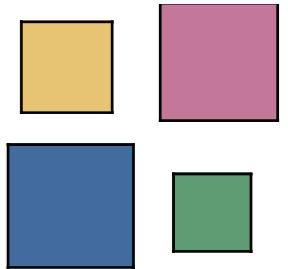
# get selected IDs and plot
ids <- iset.selected()
ls  <- lowess(Total.Bill.in.USD[ids], Tip.in.USD[ids])
ilines(ls$x, ls$y, col="marked")
```



iPlots Anwendungen: Dynamische Modelle

- Da **iPlots** NICHT Signale an **R** senden kann, muss aus R eine Event-Loop simuliert werden.
- Zum Herausspringen aus dieser Loop können **iPlots** ein Flag setzen:
`ievent.wait()` wartet auf einen solchen **iPlots** Event
- Ein “Event Loop” für **iPlots** in **R** sieht dann folgendermaßen aus:

```
while (!is.null(ievent.wait())) {  
  #  
  # check whether iplots have changed and update  
  #  
}
```
- Der Break-Event wird in **iPlots** mittels `<meta> <shift> B` ausgelöst



Dynamische Modelle: Beispiel

- **Film Daten**

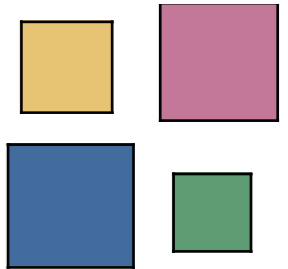
Informationen über 335 Filme die in den Jahren 1998/99 in den USA released wurden (nicht nur US-amerikanische Filme)

- **Wichtigste Variablen**

- Movie
- Genre
- ...
- $\log(\text{First.weekend})$
- $\log(\text{Opening.screens})$
- $\log(\text{Total.domestic.gross})$

- **Fragestellung**

“Wie groß ist der Anteil des Umsatzes eines Films am Start-Wochenende gemessen am gesamten Umsatz in Abhängigkeit von Genre und/oder Anzahl der Startkinos?”



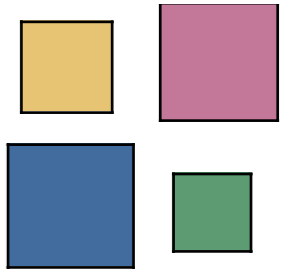
Dynamische Modelle: Code (1)

- Dynamisches lineares Modell mit Konfidenz Bändern

```
# set up iplots
iplot(Log.1st.weekend, Log.domestic)
ibar(Genre)
ihist((Total.domestic.gross - First.weekend) /
Total.domestic.gross)
ihist(Log.opening.screens, breaks=seq(0,4,.25))

# select scatterplot
iplot.set(1)

# create iobj for later modification
iabline(0, 0, col="marked", visible=FALSE)
ilines(c(0,0), c(0,0), col="marked", visible=FALSE)
ilines(c(0,0), c(0,0), col="marked", visible=FALSE)
```



Dynamische Modelle: Code (2)

```
while (!is.null(ievent.wait())) {
  if (iset.sel.changed()) {
    s <- iset.selected()
    if (length(s)>0) {
      y <- Log.domestic[s]
      x <- Log.1st.weekend[s]
      iobj.rm()
      iobj.rm()
      iobj.opt(reg=lm(y ~ x), visible=TRUE)
      ci<-predict(lm(y ~ x),
                  data.frame(x=sort(x)), level=.9, interval="confidence")
      ilines(sort(x), ci[,2])
      ilines(sort(x), ci[,3])
    } else {
      iobj.opt(visible=FALSE)
    }
  }
}
```