

## Lattice – Handout

(Voraussetzung zur erfolgreichen Ausführung der R-Codes, ist das Vorliegen der in der Sitzung verwendeten Datensätze)

- Lattice generiert *multiple plots*, implementiert Trellis Graphiken → gruppierte Graphiken
- Lattice repräsentiert *high-level* Funktionen (z.B. histogram(), densityplot(), qqmath(), qq(), stripplot(), dotplot(), barchart(), xyplot(), splom(), contourplot(), levelplot(), wireframe(), cloud(), parallel())

### 1 Einführung

- Installieren des Packages "lattice" → Package für die Graphiken  
`install.packages("lattice") library("lattice")`
- Basisfunktion in Lattice → `y ~ x | a * b`
  - o Links des `|` stehen die unabhängigen Variablen (mindestens eine), rechts die abhängigen Variablen (optional)
  - o Funktion `|`: Was stellt die Bedingungsvariable dar? → für jede Ausprägung wird ein *panel* erstellt
  - o Unterschied zu `group =` → Variable, nach der *innerhalb* eines *panels* gruppiert wird
- Beispiel Histogramm:  
`histogram(~gcsescore | factor(score), data = Chem97)`
- Beispiel Dichtefunktion:  
`densityplot(~ gcsescore, data = Chem97, groups = score, plot.points = FALSE, ref = TRUE, auto.key = list(columns = 3))`

### 2 Visualisierung univariater Verteilungen

- Q-Q-Plots → Normalverteilung graphisch überprüfen  
`qqmath(~ gcsescore | factor(score), Chem97, groups = gender, f.value = ppoints(100), auto.key = TRUE, type = c("p", "g"), aspect = "xy")`
- Zwei-Stichproben-qq-Plots: nur simultaner Vergleich → alternativ Box-Whisker-Plots  
`bwplot(factor(score) ~ gcsescore | gender, Chem97)` (senkrechte Anordnung: Box-Whisker-Plots durch Erweiterung `layout = c(Anzahl der Spalten, Anzahl der Zeilen)`)

### 3 Scatter Plots und Erweiterungen

- Scatter Plots für bivariate kontinuierliche Variablen
- Befehl: `xyplot(Y-Achse ~ X-Achse, data = Name)`, Erweiterung zur Benennung der Achsen: `xlab = „Name“`, `ylab = „Name“`

### 4 Shingles

- Kontinuierliche Variable → Bildung von Intervallen (endliche Anzahl von *panels*)
- Shingles für "Variable": Name der neuen Variable (Shingle) <- `equal.count(Variable$Datensatz, number=Anzahl der Intervalle, overlap=prozentuale Überschneidung)`
- `summary(Name der neuen Variable)` → Wie viel Fälle/Überlappungen enthalten die Intervalle?

### 5 Trivariate Scatterplots

- Dreidimensionaler Scatterplot: `cloud (X ~ Y * Z)`
- Drehen der Achsen: `cloud (X ~ Y * Z, data = Name, screen = list(z = Zahl, x = Zahl))`

### 6 Visualisierung tabellarischer Daten

- Säulendiagramme, Punktdiagramme: `install.packages("latticeExtra") library("latticeExtra")`
- Säulendiagramm: `barchart(Y-Variable ~ X-Variable | Bedingungsvariable, Datensatz, layout = c(Anzahl der Spalten, Anzahl der Zeilen), origin = 0 (X-Achsen-Ursprung))`
- Punktdiagramm : `dotplot ( )` → Ausführung wie bei Säulendiagramm

### 7 Literatur

Sarkar, D. (2008). *Lattice. Multivariate Data Visualization with R*. Springer: New York.

Sarkar, D. (o.J.). *Getting started Lattice Graphics*. URL: <http://lattice.r-forge.r-project.org/Vignettes/src/lattice-intro/lattice-intro.pdf>