Lattice - Leitfaden

- Lattice ist im Vergleich zu Standardverfahren in der Lage *multiple plots* zu generieren
- Implementation von Trellis Graphiken → Darstellung von gruppierten Graphiken
- Was kann Lattice? → *high-level* Funktionen

Funktion	Anzeige
histogram( )	histogram
densityplot( )	Kernel Density Plot
qqmath( )	Theoretical Quantile Plot
qq( )	Two-sample Quantile Plot
stripplot( )	Stripchart (Comparative 1-D Scatter-Plots)
bwplot( )	Comparative Box-and-Whisker Plots
dotplot( )	Cleveland Dot Plot
barchart( )	Bar Plot
xyplot( )	Scatter Plot
splom( )	Scatter Plot Matrix
contourplot( )	Contour Plot of Surfaces
levelplot( )	False Color Level Plot of Surfaces
wireframe( )	Three-dimensional Perspective Plot of Surfaces
cloud( )	Three-dimensional Scatter Plot
parallel( )	Parallel Coordinates Plots

1 Einführung

- Installieren des Packages „mlmRev“ → Übungsdatensatz

```
install.packages("mlmRev")
library("mlmRev")
data(Chem97, package = "mlmRev")
```

  - es handelt sich hierbei um einen Datensatz von 31.022 britischen Studenten (A-Level examination in Chemie) aus dem Jahr 1997
  - Datensatz in R anzeigen lassen: demographische Variablen, score = Note in Chemie-Abschlussarbeit; gcsescore = Durchschnittsnote aus vorherigen Arbeiten

- Installieren des Packages “lattice” → Package für die Graphiken

```
install.packages("lattice")
library("lattice")
```

- Basisfunktion in Lattice → y ~ x | a * b

- ~ macht Funktion zu einem *formula* Objekt und ist essentiell für Trellis Graphiken
- | benennt die abhängige Variable
- Unabhängige Variablen stehen links des |, abhängige rechts; UV mindestens eine, AV optional; AV werden durch * getrennt

- Kreuztabelle zur Übersicht anzeigen lassen

```
xtabs(~ score, data = Chem97)
```

- Lattice kann gruppierte Panels erstellen!
- Gruppierte Histogramme: Ist die Verteilung der Durchschnittsnote bei allen Examinensnoten gleich?

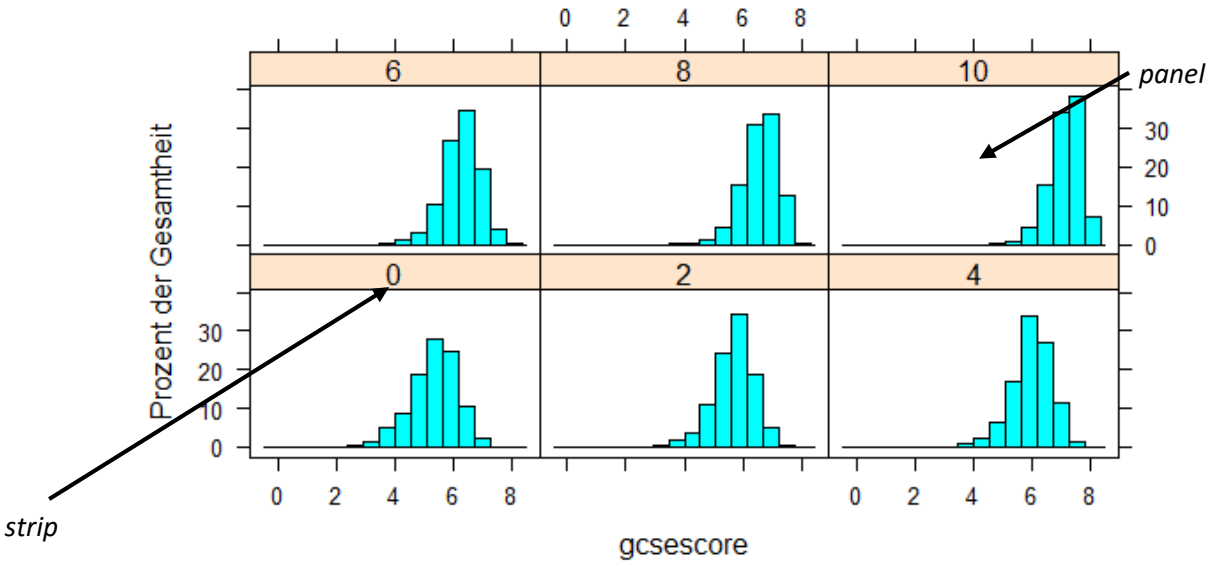
```
histogram(~gcsescore|factor(score), data = Chem97)
```

Welche Variable soll die unabhängige darstellen?

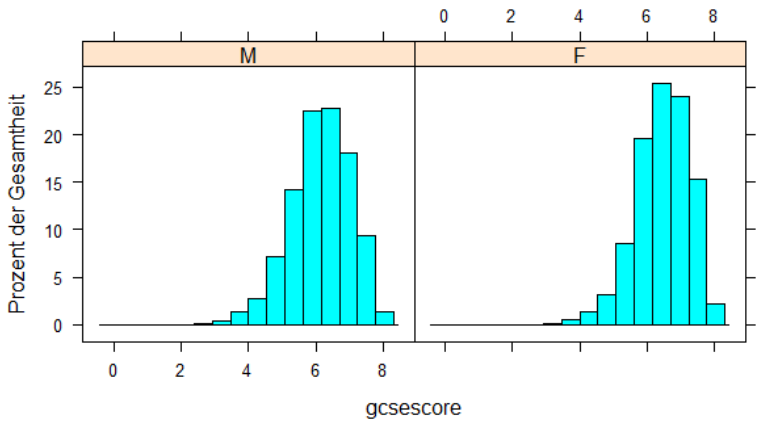
Was ist die abhängige Variable?

Was ist der data frame?

- *multipanel conditioning*: Merkmal der Trellis Graphiken → später weitere Vorteile der *panels*
- Jedes *panel* weist dieselbe Skala auf → bessere Vergleichsmöglichkeiten
- Achsen werden nur außerhalb der *panels* angezeigt (platzsparend)
- *strip* gibt an, welche Ausprägung die abhängige Variable (hier: Examensnote) aufweist



Aufgabe: Erstellt ein Histogramm der Durchschnittsnote getrennt nach dem Geschlecht.



- Ergebnis des ersten Histogramms: Je höher die Examenspunkte, desto weiter rechts ist die Verteilung der Durschnittsnoten → dies wäre besser zu erkennen, wenn die Verteilungen in einem Panel abgetragen sind (Dichtefunktion)

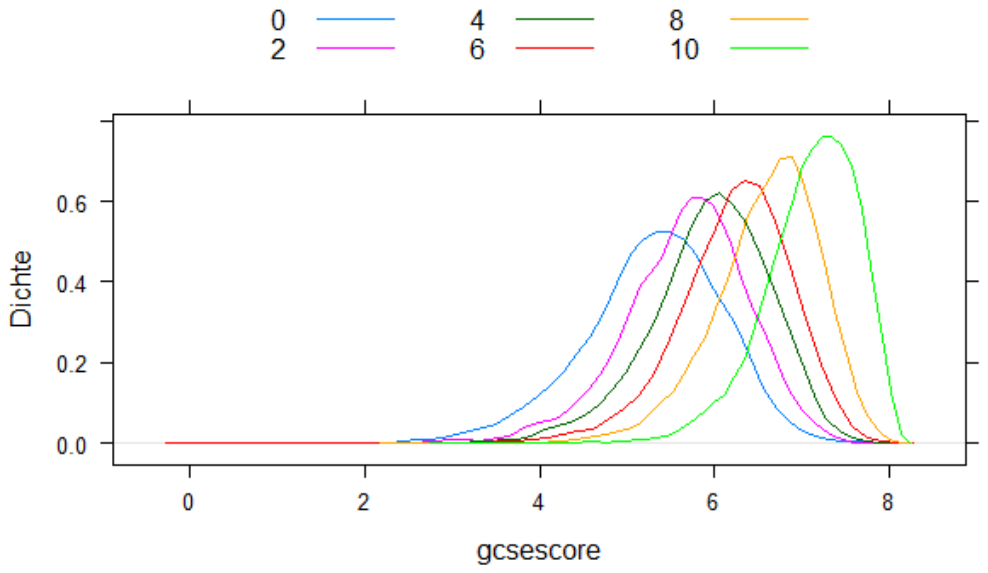
```
densityplot(~ gcsescore, data = Chem97, groups = score, plot.points = FALSE, ref = TRUE, auto.key = list(columns = 3))
```

Einfügen einer Legende

Soll innerhalb eines Panels eine Gruppierung erfolgen?

Sollen die Datenpunkte angezeigt werden?

Soll Referenzlinie bei 0 angezeigt werden?



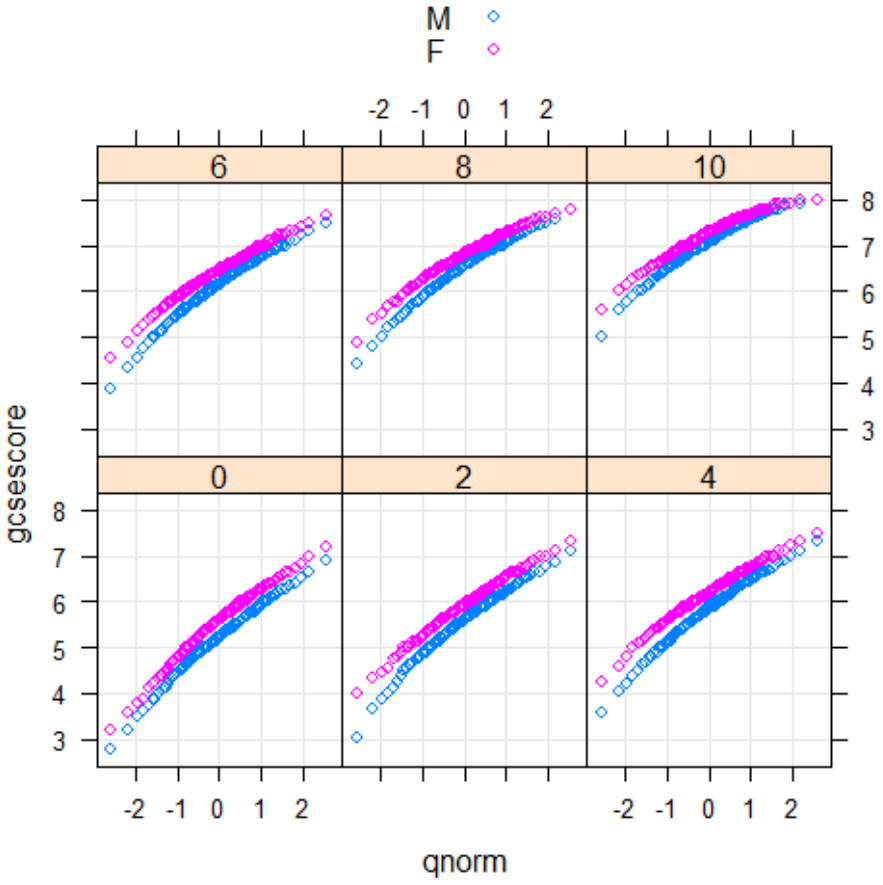
Aufgabe: Erstellt eine Dichtefunktion für die Durchschnittsnote, gruppiert nach Männern und Frauen. Die AV soll jeweils die Examensnote sein. Was ist der Unterschied zwischen `group =` und `|factor( )`?

2 Visualisierung univariater Verteilungen

- Q-Q-Plots sind in der Lage die Normalverteilung graphisch zu überprüfen
- Diese Plots stellen die emprischen Daten der Normalverteilung oder einer anderen theoretischen Verteilung gegenüber → bei guter Passung liegen die Punkte auf einer Geraden

  `qqmath(~ gcsescore | factor(score), Chem97, groups = gender, f.value = ppoints(100), auto.key = TRUE, type = c("p", "g"), aspect = "xy")`

  - f.value: Wie viele Datenpunkte werden angezeigt?
  - type: p = Anzeigen der Punkte; g = Anzeigen einer Gitternetzlinie
  - aspect: In welchem Verhältnis stehen Länge und Breite? → xy: 45° Banking Rule (Durchschnittsslope hat eine Steigung von 45°
  - qnorm: Normalverteilung → Standardeinstellung; andere Verteilung durch `distribution =` (siehe `?qqmath`)



Aufgabe: Vergleicht die Verteilungen der Durchschnittsnote zwischen den Geschlechtern. Verwendet hierzu die Funktion `qq ( )`. Die Gitternetzlinien sollen in einem Verhältnis von 1:1 angezeigt werden. Bei Bedarf ruft die Hilfefunktion für `qq ( )` auf.

- Nachteil des Zwei-Stichproben-qq-Plots: Nur ein Vergleich auf einmal möglich → Alternative zum Vergleich der Verteilungen: Box-Whisker-Plots

  ```
  bwplot(factor(score) ~ gcsescore | gender, Chem97)
  ```

  - Y-Achse ~ X-Achse|Bedingungsvariable
- Für einen besseren Geschlechtsvergleich können die Box-Whisker-Plots auch senkrecht nebeneinander angeordnet werden

Aufgabe: Was sollte X- und Y-Achse sein? Der Befehl muss um `layout = c(Anzahl der Spalten, Anzahl der Zeilen)` erweitert werden.

3 Scatter Plots und Erweiterungen

- Scatter Plots werden oft genutzt bei bivariaten kontinuierlichen Variablen
- Ist geeignet um graphisch zu inspizieren welche Art von Zusammenhang vorliegt (linear, kurvlinear etc)
- Neuer Datensatz:

  ```
  data(Earthquake, package = "nlme")
  ```
- Scatter Plots werden mit dem Befehl `xyplot(Y-Achse ~ X-Achse, data = Name)` durchgeführt → wenn die Achsen nicht mit den Variablennamen beschriftet werden sollen, kann dies durch den Befehl `xlab = „Name“, ylab = „Name“` ergänzt werden

Aufgabe: Erstellt einen Scatterplot. Die Y-Achse soll die Akzeleration angeben, die X-Achse die Distanz. Beschriftet die Achsen sinnvoll.

4 Shingles

- Kontinuierliche Variablen haben i.d.R. so viele Ausprägungen, dass es bei Gruppierungen nach dieser zu unendlich vielen *panels* käme → die Lösung hierbei ist, diese Variable in Intervalle einzuteilen
- Neuer Datensatz: `quakes` → 1000 Beobachtungen bei Erdbeben
  - lat = Latitude
  - long = Longitude

- depth = Tiefe in km
- mag = Richter-Skala
- stations = Anzahl der Stationen, die Erbeben aufzeichnen

- Bildung eines Shingles für die Variable depth:

```
Depth <- equal.count(quakes$depth, number=8, overlap=.1)
summary(Depth)
```

- Number = Anzahl der Intervalle
- Overlap = Wie stark dürfen sich die Intervalle überschneiden?
- Summary = Was sind die Unter- und Obergrenzen der Intervalle? Wie viele Fälle enthalten die Intervalle jeweils? Wie viele Fälle überlappen sich?

Aufgabe: Erstellt einen Scatterplot. Die Y-Achse soll die Latitude, die X-Achse die Longitude darstellen. Die Bedingungsvariable soll der Shingle Depth sein.

5 Trivariate Scatterplots

- Optimal wäre ein dreidimensionaler Scatterplot, in dem Tiefe, Longitude und Latitude abebildet wird
- Dies ist mit `cloud (X ~ Y * Z)` möglich
- Nachteil: Der Graph ist statisch → er dreht sich nicht → Halblösung: Drehen der Achsen

```
cloud (X ~ Y * Z, data = Name, screen = list(z = Zahl, x = Zahl)
```

Aufgabe: Erstellt einen 3D-Scatter-Plot mit den Variablen Tiefe, Longitude und Latitude. Probiert verschiedene Winkel aus, um den Graph aus verschiedenen Richtungen zu betrachten. Beschriftet zudem die Achsen (siehe Scatterplots und Erweiterungen).

6 Visualisierung tabellarischer Daten

- Tabellen sind eine Möglichkeit Daten zu präsentieren; Tabellen können jedoch auch graphisch veranschaulicht werden → z.B. mit Säulendiagrammen oder Punktdiagrammen → dafür wird ein neues package benötigt

```
install.packages("latticeExtra")
library("latticeExtra")
```

- Tabelle: Todesraten 1941 in Virginia von verschiedenen Subgruppen

```
VADeaths
```

- Um Lattice nutzen zu können, müssen die Daten zunächst in einen data frame übertragen werden

```
VADeathsDF <- as.data.frame.table(VADeaths, responseName = "Rate")
```

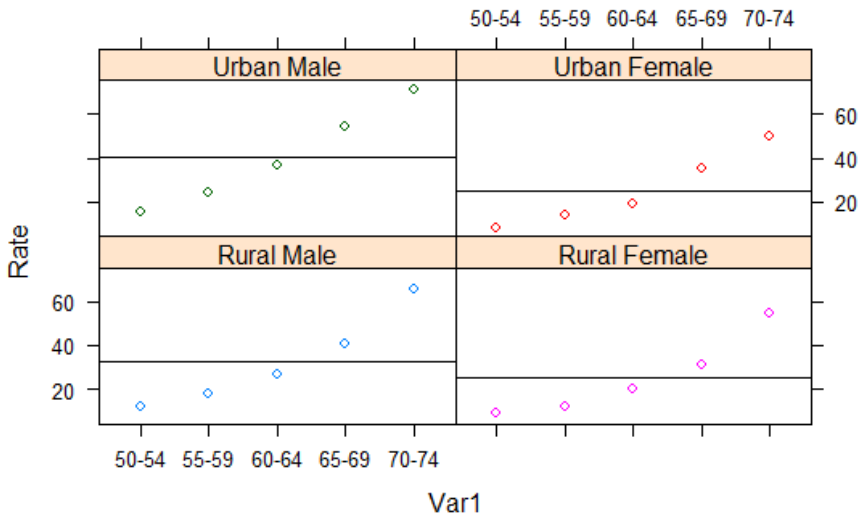
VADeathsDF

- Säulendiagramm: Befehl barchart ( ) → Y-Achse ~ X-Achse|Bedingungsvariable, Datensatz, layout = c(Anzahl der Spalten, Anzahl der Zeilen), zusätzlich kann der Achsenursprung mit origin = 0 ergänzt werden
- Punktediagramm mit Befehl dotplot ( )

Aufgabe: Erstellt ein Säulendiagramm und ein Punktediagramm, gruppiert nach den geschlechtsspezifischen Subgruppen. Der Achsenursprung soll 0 sein. Erstellt im nächsten Schritt ein Liniendiagramm mit den Gruppierungen in einem Panel (beachtet die Anmerkungen bei densityplot ( )). Punkte können mit dem Befehl type = "b" verbunden werden.

- Weiterer Vorteil von *panels*: es können auch eigene Statistiken mit in die panels eingetragen werden → z.B. der Mittelwert der Subgruppen in jedem Panel (siehe Beispiel für VADeathsDF)

```
xyplot(Rate ~ Var1|Var2, data=VADeathsDF,
       group = Var2,
       panel=function(x,y,...) {
         panel.xyplot(x,y,...)
         panel.abline(h=mean(y))
       }
)
```



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