Zusammenfassung EXPD / R-Studio

Kategorielle Merkmale

- Keine Zahlenwerte im engeren Sinn
- Endliche viele verschiedene Ausprägungen

Numerische Merkmale

- Nehmen nummerische Werte an

	Nominale Merkmale	Ordinale Merkmale
Eigenschaft	Ausprägungen lassen sich **nicht** in natürlicher Weise anordnen	Ausprägungen lassen sich in natürlicher Weise anordnen
Beispiele	Farbe von Gegenständen, Studiengang, Nationalität	Beförderungsklasse (Economy, Economy Plus, etc), höchster Abschluss, Zustimmungswerte bei einer Kundenumfrage
Bemerkung	Können in einem Datensatz als Zahlen codiert sein (Gruppe 1, 2, 3, ...). Man kann mit diesen Zahlen aber nicht rechnen	Sind häufig als Zahlen codiert (5 = stimme voll zu, ..., 1 = stimme überhaupt nicht zu). Es ist aber nur begrenzt sinnvoll, mit diesen Zahlen zu rechnen.

Diskrete Merkmale	Stetige Merkmale
Wertebereich ist **abzählbar**, aber möglicherweise unendlich gross	Merkmale, die prinzipiell jeden **beliebigen Wert** in einem (evtl. unbeschränkten) Bereich annehmen können. Zwischen zwei beliebigen Werten existiert **immer noch ein Wert dazwischen**
Anzahl Passagiere in einem Flugzeug, Punktzahl in der Klausur	Umsatzzahlen eines KMUs, Durchmesser einer Schraube
	Praktisch können fast alle stetigen Grössen nur diskret gemessen werden, da die Genauigkeit der Messungen limitiert ist.

Histogramm – Klassenzahl

Tipp: Mit verschiedenen Breiten experimentieren und das Histogramm wählen, das die Form der Verteilung am besten wiedergibt. Die Stichprobengrösse insgesamt und pro Balken darf aber für aussagekräftiges Diagramm nicht zu klein sein.

Verschiedene **Faustregeln** zur Wahl der Klassenanzahl k

- $k = \lfloor \sqrt{n} \rfloor$
- $k \approx \sqrt{n}$
- $k = \lfloor 10 \cdot \log_{10}(n) \rfloor$
- $k \approx 1 + \log_2(n)$ (Default in R)

mit n = Anzahl Beobachtungen

- **Lagemasse** beschreiben, um welchen "mittleren" Wert die Daten verteilt sind (nur für unimodal sinnvoll)
- **Streuungsmasse** geben an wie "breit" die Verteilung ist, d.h. wie stark die Werte streuen.

Mittelwerte

Arithmetisches Mittel – Durchschnitt $\bar{x}$

$m = (x_1+x_2+...+x_n)/n$

Geometrisches Mittel - Mittel der Wachstumsraten (Zinssatz/Zinseszins)

$g = (x_1*x_2*...*x_n)^{1/n}$

Harmonisches Mittel - Mittel von Quotienten (Ratios, Verhältnisse)

$h = n/(1/x_1 + 1/x_2 + ... + 1/x_n)$

Median $\tilde{x}$

Quantile

Quantile teilen eine Stichprobe in einem bestimmten Verhältnis. α Quantil teilt die Stichprobe im Verhältnis α : (1 − α). Beispiel: $Q_{90\%}$ teilt die Stichprobe in 90% : 10%

Streumasse

Varianz (s_x^2)

$$s_x^2 = \frac{1}{(n-1)}\sum_{i=1}^{n}(x_i-\bar{x})^2 = \left[\frac{1}{(n-1)}\sum_{i=1}^{n}x_i^2\right] - \frac{n}{(n-1)}(\bar{x})^2$$

Mittlere quadratische Abweichung der Beobachtungen vom arithmetischen Mittel ($\bar{x}$)

In R → var(A) oder sd(A)^2

Standardabweichung (sx)

$$s_x = \sqrt{s_x^2} = \sqrt{\frac{1}{(n-1)}\sum_{i=1}^{n}(x_i-\bar{x})^2}$$

Quadratwurzel der Varianz. Gleiche Einheit wie Beobachtung (besser interpretierbar)

In R → sd(A) oder sqrt(var(A))

MAD (median absolute deviation)

$$\text{MAD}_x = 1.4826 \cdot \text{median}(|\, x_1 - \tilde{x}\,|, |\, x_2 - \tilde{x}\,| \ldots, |\, x_n - \tilde{x}\,|)$$

Robustes Gegenstück zur Standardabweichung. Mittelwertbildung durch Mediane ersetzen. Konstante stellt sicher, dass MAD und s_x, im Fall wenn die Daten aus einer Normalverteilung stammen, das gleiche messen. In R → mad(A)

IQR (inter quartile range)

Abstand zwischen oberem und unterem Quartil: $IQR = Q_3 - Q_1$ In R → IQR(A)

Spannweite

Differenz zwischen Maxima und Minima. Ist jedoch anfällig auf Ausreisser. Nimmt mit Stichprobenumfang zu ⇒ als Streuungsmass ungeeignet In R → max(A) – min(A)

Wann IQR, wann Standardabweichung?

- Bei rechts/linksschiefen Verteilungen liefert die Standardabweichung einen Wert, der für den Grossteil der Daten nicht repräsentativ (zu hoch ist). Die IQR liefert meist eine bessere Vorstellung der Streuung.
- Für symmetrische Verteilungen ohne Ausreisser ist die Standardabweichung ein gut geeignetes Mass.
- Die Standardabweichung ist nicht robust! Eine einzige, falsche Beobachtung kann sie grob zu verfälschen.

Boxplot

Weniger geeignet für bi- /multimodale Verteilungen: Diese wird in einem Boxplot nicht erfasst.

Bivariate Darstellungen

Verschiedene Variablentypen benötigen verschiedene Handhabung. Es gibt 3 Fälle:

1. Kategoriell vs. Kategoriell
2. Metrisch vs. Kategoriell
3. Metrisch vs. Metrisch

Zwei kategorielle Variablen

Kreuztabelle → table(dat$Haarfarbe, dat$Augenfarbe)

Anteil am Gesamten → prop.table(table(dat$Haarfarbe, dat$Augenfarbe))
Anteil pro Zeile → prop.table(table(dat$Haarfarbe, dat$Augenfarbe), **margin = 1**)
Anteil pro Spalte → prop.table(table(dat$Haarfarbe, dat$Augenfarbe), **margin = 2**)

Gruppierte Balkendiagramme

Gestapelten Balkendiagramm → barplot(table(dat$Haarfarbe, dat$Augenfarbe))
Gruppiertes Balkendiagramm → barplot(table(dat$Haarfarbe, dat$Augenfarbe), **beside = TRUE**)

Mosaikplot

mosaicplot(table(dat$Augenfarbe, dat$Haarfarbe)) → variablen können auch vertauscht werden = anderer Plot!

Kategorielle und metrische Variable

Kennzahlentabelle

m <- tapply(kdata$einkauf, kdata$zivilstand, "mean")
s <- tapply(kdata$einkauf, kdata$zivilstand, "sd")
cbind(Mittelwert = m, Standardabweichung = s)

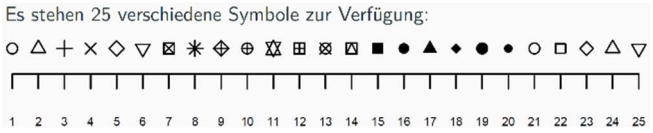


Abbildung 1: Symbole für Plot's, pch = 8

Boxplot (nur für unimodale Verteilung geeignet)

boxplot(einkauf ~ zivilstand, data = kdata)

Stripchart (vor allem bei bimodaler Verteilung sehr geeignet)

stripchart(einkauf ~ zivilstand, data=kdata, vertical=TRUE, method="stack")

Zwei metrische Variablen

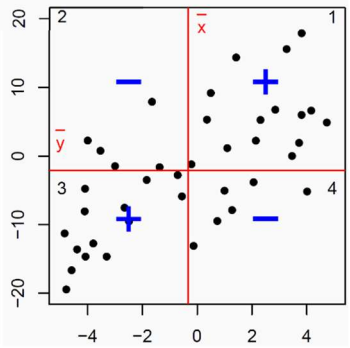
Streudiagramm (plot oder scatter.smooth mit laufenden Mittels →Glätter)

scatter.smooth (x=dat$alter, y=dat$einkauf, pch=16, col = rgb(0,0,0, alpha = 0.1), **lpars = list(col = "red", lwd = 2))**

Aus einem Streudiagramm kann man Form, Richtung und Stärke des Zusammenhangs erkennen.

Form: streuen die Punkte um eine Gerade (linear), um eine Kurve oder existieren diverse Punktewolken?

$$\mathrm{Cov}(x,y) = \frac{1}{n-1}\sum_{i=1}^{n}[(x_i - \bar{x}) \cdot (y_i - \bar{y})]$$



Richtung: Je grösser die Werte einer Variablen,
…desto grösser sind die Werte der anderen Variable (positiver Zusammenhang)
…desto kleiner sind die Werte der anderen Variable (negativer Zusammenhang)

Stärke: Wie breit ist die Punktewolke? Wenn die Punktwolke breit ist, dann ist der Zusammenhang schwach. Wenn die Punktwolke schmal (Extremfall eine Gerade!) ist, dann ist der Zusammenhang stark.

Kovarianz

Der Zusammenhang zweier Variablen wird durch die Kovarianz beschrieben. Wir betrachten die Lage der Punkte bez. des Schwerpunktes ($\bar{x}, \bar{y}$).

- Die Kovarianz informiert nur über die Richtung des Zusammenhangs, nicht dessen Stärke.
- Der absolute Zahlenwert ist schwer zu interpretieren, da die Grösse der Kovarianz vom Massstab x/y abhängt.
- Die Kovarianz ist sehr anfällig für Ausreisser.
- Die Kovarianz von zwei identischen Variablen entspricht der Varianz.

Berechnung in R → cov(dat$umsatz, dat$werbung)

Pearson-Korrelation (einfach oder Produkt-Momenten Korrelation)

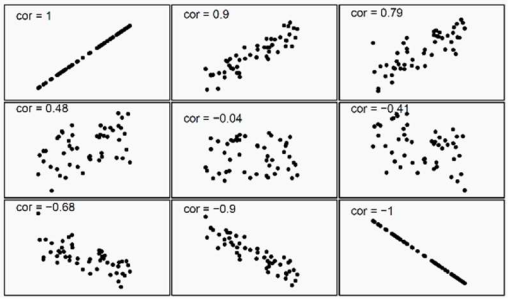
Misst die Stärke des **linearen Zusammenhangs**, d.h. wie eng die Punkte um eine Gerade liegen bzw. wie stark sie streuen, und entspricht der standardisierten Kovarianz.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2\sum_{i=1}^{n}(y_i-\bar{y})^2}} = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{(x_i-\bar{x})(y_i-\bar{y})}{s_x s_y}\right) = \frac{\mathrm{Cov}(x,y)}{s_x s_y}$$

Berechnung in R → cor(variable1, variable2)

Eigenschaften

- $r_{xy} > 0$ → positive Korrelation (Je mehr, desto mehr)
- $r_{xy} < 0$ → negative Korrelation (Je mehr, desto weniger)
- $r_{xy} = 1$ → Punkte liegen exakt auf einer Gerade positiver Steigung
- $r_{xy} = -1$ → Punkte liegen exakt auf einer Gerade mit negativer Steigung
- r_{xy} nahe bei ±1 → Punkte streuen eng um eine Gerade
- $r_{xy} \approx 0$ → kein linearer Zusammenhang, es kann aber ein anderer Zusammenhang bestehen
- r_{xy} ist **nicht die Steigung der Geraden**



Wichtig: Die **Form** des Zusammenhangs wird von der Korrelation **nicht geprüft**, fliesst aber als Annahme in die Berechnung ein! Korrelation nie ohne Blick auf ein Streudiagramm beurteilen!!!

Die Spearman-Korrelation (Rang-Korrelation)

$$r_{x_R,y_R} = \frac{\sum_{i=1}^{n}\left(\mathrm{Rang}(x_i)-\frac{n+1}{2}\right)\left(\mathrm{Rang}(y_i)-\frac{n+1}{2}\right)}{\sqrt{\left(\mathrm{Rang}(x_i)-\frac{n+1}{2}\right)^2}\sqrt{\left(\mathrm{Rang}(y_i)-\frac{n+1}{2}\right)^2}}$$

Misst die Stärke des **monotonen Zusammenhangs**, d.h. wie nahe die Punkte um eine Kurve liegen, die von einer beliebigen, monotonen Funktion definiert ist. Funktionsweise: Man bestimmt separat die Ränge der x- und y-Werte und berechnet auf diesen Rängen die Pearson-Korrelation. Die Rangkorrelation ist der Pearson-Korrelation vorzuziehen, wenn der **Zusammenhang nicht linear**, sondern **monoton** ist, es Ausreisser geben könnte und die Werte (x_i, y_i) nicht glockenförmig verteilt sind.

Berechnung in R → cor(variable1, variable2, method = "spearman")

Eigenschaften:

- rxy = 1 → Punkte liegen exakt auf einer monotonen Kurve mit positiver Steigung
- rxy = −1 ! Punkte liegen exakt auf einer monotonen Kurve mit negativer Steigung
- *r_{xy} nahe bei ±1 ! Punkte streuen eng um eine **monotone Kurve***
- *restliche Eigenschaften sind gleich wie bei der Pearson-Korrelation*

Vorsicht bei **kleiner Stichprobengrösse**! Bei 2 Punkten ist die Korrelation immer ±1.

Logische Operatoren

Reihenfolge der Auswertung: ! kommt vor & kommt vor |, mittels Klammern () kann die Reihenfolge angepasst werden. (Verknüpfung mit & kann als Multiplikation mit | als Addition gesehen werden)

Transformation

Bei einer Transformation werden die Werte einer Variablen mittels einer eindeutigen Zuordnung zu neuen Werten umgerechnet: x |→ f(x)

Gründe für Datentransformation:

- Zusammenfassen von Beobachtungen in Klassen
- Umrechnen von Einheiten
- Informativere Darstellung/Daten standardisieren
- Änderung unvorteilhaften Form einer Verteilung

Nominale sowie ordinale Daten

Es gibt Transformationen bei beiden jeweils mit und ohne Informationsverlust.

Lineare Transformation

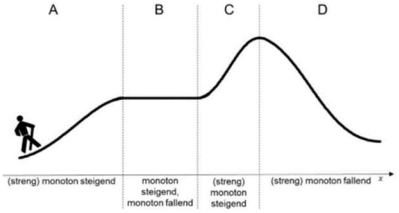
Änderung der Masseinheit (inch in cm): Transformation bei der keine Verschiebung involviert ist: x |→ f(x) = a * x

Korrektur von Messwerten (Waage verstellt): Jeder Wert wird um denselben Betrag korrigiert x |→ f(x) = x + b

Verschiebung der Skala und des Mess-Nullpunktes (°F in °C): x |→ f(x) = a·* x + b

Streng monotone Transformation

Bei Transformationen mit streng monoton **steigenden** Funktionen bleibt die Reihenfolge der Werte bestehen. Umsetzung durch Logarithmus oder Wurzelfunktion!



Auswirkungen von Transformationen

Je nach Transformation können sich die Kennzahlen der Variablen ändern (Mittelwert, Standardabweichung) oder kann sich die Verteilung der Variablen ändern.

Im Fall der linearen Transformation lassen sich die Auswirkungen mathematisch einfach beschreiben. Bei **streng** monoton steigenden Transformationen ist dies etwas schwieriger.

Umrechnung der linearen Transformationen

Die lineare Transformation ändert die Form der Verteilung nicht, nur die Achsenbeschriftung ändert sich.

- Mittelwert: $\bar{y}$ = a * $\bar{x}$ + b.
- Standardabweichung: $sd_y = |a| * sd_x$

Lage- und Streumasse ändern sich bei einer linearen Transformation (x |→ f(x) = a * x + b) wie folgt:

- $Lage_Y = a * Lage_X + b$
- $Streuung_Y = |a| * Streuung_X$

Wobei Lage: arithm. Mittel, Median, Quantile (nur für a ≥ 0), Modus
Streuung: Standardabweichung, MAD, IQR.

Achtung: Varianz ist ein quadratisches Streumass daher $Var_Y = a^2 * Var_X$

Standardisierung (Spezialfall einer linearen Transformation)

Die Standardisierung ist nützlich, um Daten zu vergleichen, wenn man sich für die Verteilungsform interessiert, Lage und Streuung aber nicht berücksichtigen möchte: x |→ $z = \frac{1}{sd_x} * x - \frac{\bar{x}}{sd_x}$

Die standardisierten Variablen haben Mittelwert 0 und Standardabweichung 1

$z = \frac{1}{sd_x} * x - \frac{\bar{x}}{sd_x} = 0$ $sd_z = \frac{1}{sd_x} * sd_x = 1$

Im Gegensatz zur linearen Transformation wird durch eine streng monotone Transformation die Verteilung der Variable geändert.

Was tut man mit mehr als 2 Variablen?

Mehrere kategorielle Variablen

➔ Mosaikplot mosaicplot(~k1 + k2 + k3 + ...)

1 quantitative und mehrere kategorielle Variable

➔ Boxplots boxplot(m1~k1 + k2 + k3 + ...) ➔ Faktor/Design-Plot plot.design

2 quantitative und mehrere kategorielle Variable

Die 2 quantitativen Variablen werden mittels eines Streudiagramms aufgezeichnet und die kategorieller Variable zusätzlich durch Farben, Symbolform und Symbolgrösse visualisiert. plot(m1, m2, col[k1], pch[k2])

Reihenfolge in abnehmender Wirksamkeit ist zu beachten:

1. Grösse
2. Text
3. Orientierung (einer Linie, eines Rechtecks)
4. Form (Stern, Kreis, Rechteck)
5. Intensität
6. Farbton/Farbsättigung

Mehr als 2 quantitative Variablen

3D-Plots Co-plot Streudiagramm-Matrix Korrelationsmatrix

Stichproben - Schlussfolgerungen

- Die Standardabweichung nimmt mit wachsendem n (Anzahl der Stichproben) ab.
- $\sqrt{(n)} * s_x$ ist ungefähr konstant, d.h. Standardabweichung ist umgekehrt proportional zur Wurzel aus n.

Möchte man die Standardabweichung der Mittelwerte halbieren, muss man die Stichprobengrösse vervierfachen! Erkenntnis kann im Boxplot verwendet werden, um bei Stichproben ein Intervall anzugeben, in dem der wahre Median «ziemlich sicher» liegt. **Faustregel**: Wenn sich die Kerben von zwei Boxen nicht überschneiden, so besteht ein signifikanter Unterschied zwischen den Gruppen.

Hauptkomponentenanalyse (PCA)

Idee: Finde diejenigen Achsen, welche die Daten am besten erklären und verwende nur ein paar davon (2-3).
Das Koordinatensystem wird so gedreht, dass die Varianz in Richtung der neuen x-Achse am grössten ist.
→ Drehung der Achsen, so dass nur minimal Information verloren geht.
→ Dies ist mathematisch äquivalent die Richtung zu finden, in der die grösste Streuung (Varianz) vorliegt.
Die PCA wird verwendet, um multidimensionale Daten durch die ersten zwei Hauptkomponenten auf 2 Dimensionen zu reduzieren (Visualisierung), so dass der grösste Teil der Information (Streuung) der Daten erhalten bleibt.

Schritte der PCA:

- Verschieben des Koordinatensystems in den Schwerpunkt der Daten.
- Rotation des ursprünglichen Koordinatensystems zum Koordinatensystem der Hauptkomponenten, sodass die Varianz entlang der ersten Hauptkomponente am grössten ist.
- Der grösste Teil der restlichen Varianz soll entlang der zweiten Hauptkomponenten liegen (Achse steht senkrecht zur ersten Hauptkomponente).

Die Rotationsmatrix/Ladungsmatrix rechnet die ursprünglichen Werte in neuen Werte im "gedrehtem" Koordinatensystem um. Der Absoultbetrag der "Ladungen" gibt an wie wichtig die ursprüngliche Variable für die neue Position ist. Ist die Ladung der Variable 1 mit 0.899 grösser als die von Variable 2 mit 0.44, damit hat die Variable 1 grösseres Gewicht als Variable 2. → res <- prcomp(X, sacle=F) \n res$x (neue Koordinaten)
Res$rotation gibt die neuen Wert im Koordinatensystem nach der Rotationsmatrix an.

Skalierung

```
dat$umsatzS <- (dat$umsatz - mean(dat$umsatz))/sd(dat$umsatz)
dat$werbungS <- (dat$werbung - mean(dat$werbung))/sd(dat$werbung)
Kovarianz der skalierten Daten =  var(dat[,c("umsatzS","werbungS")])
pca <- prcomp(dat[,1:2], scale = TRUE) # Identisch zu: #
prcomp(dat[,c("umsatzS","werbungS")], scale = FALSE)
pca$rotation
```

Auswirkung der Skalierung:

- Wenn Variablen nicht skaliert, hat die Variable mit grösster Streuung grössten Einfluss auf die neue Koordinate.
- Wenn die Variablen skaliert sind, dann haben alle Variablen das gleiche Gewicht.

Faustregel: Skaliere, wenn
die Variablen verschiedene Einheiten haben explizit gewollt ist, dass alle Variablen gleiches Gewicht haben
Nicht skalieren, wenn

- alle Variablen die gleiche Einheit haben und vergleichbar sind
- Wenn man nicht genau weiss, was zu tun ist, ist es meistens besser zu standardisieren.

Dimensionsreduktion

Situation: Wir haben nun die Rotation in ein neues Koordinatensystem. Die Dimension ist hierbei immer noch gleich gross wie im gesamten Datensatz. Ziel: Dimensionsreduktion. Verwenden wir nun nur die ersten paar Hauptkomponenten und hoffen, dass die Daten dadurch möglichst gut beschrieben werden Wie gut ist unsere Näherung? Mass dazu ist die erklärte Varianz.
Die totale Varianz berechnet sich für p Variablen als

$$\text{Var}_{\text{total}} = \sum_{j=1}^{p} \text{Var}(X_j)$$

Durch die Rotation wird die totale Varianz nicht verändert.

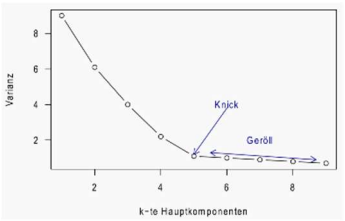
$$p_k = \left(\sum_{j=1}^{k} \text{Var}(Z_j) \right) / \text{Var}_{\text{total}}$$

```
pca <- prcomp(dat[,1:2], scale = FALSE)
sum(diag(cov(dat[,1:2])))   sum(diag(cov(pca$x)))
```

ergeben beide das Gleiche!

Als Gütekriterium der Approximation wählen wir daher den Anteil der Varianz der durch die ersten k-Hauptkomponenten erklärt wird:
1. Kriterium: k sollte so gewählt werden, dass ca. 80% der totalen Varianz durch die berücksichtigten Hauptkomponenten erklärt wird.

2. Kriterium Varianzen in abnehmender Ordnung grafisch betrachten mit screeplot. Die Position des Knicks im Plot ergibt einen weiteren Hinweis, wie viele PCs benötigt werden. Die PCs nach dem Knick tragen nicht viel zur Gesamtinformation bei.




```
pca <- prcomp(...)
screeplot(pca)
```

Interpretation der PCA

Im Score-Plot werden die ersten beiden Hauptkomponenten, die die meiste Varianz erklären visualisiert.

```
load("data/body.Rdata")
pca <- prcomp(body, scale=TRUE)
```

```
library(ggfortify)
autoplot(pca)
```

Der Biplot entspricht dem Score-Plot mit standardisierten Daten und zeigt zusätzlich noch rote Pfeile (Projektionen der ursprünglichen Merkmale in der Ebene der beiden Hauptkomponenten):
Pfeillänge = Varianz der Variable (Bei Skalierung haben alle Pfeile die gleiche Länge)
Bei korrekter Projektion zeigt der Winkel zwischen den Pfeilen die Korrelation. Je kleiner der Winkel, desto grösser die Korrelation zwischen den beiden Variablen.

```
pca <- prcomp(body, scale=TRUE)
biplot(pca)
```

Paket ggfortify gibt es Funktion autoplot, welche sich gut zum Darstellen/Interpretieren einer PCA eignet (ggplot).

```
library(ggfortify)
autoplot(pca, loadings=TRUE, loadings.label=TRUE, label = TRUE, label.hjust = -0.3)
```

Boxplot mit Kerben/Notch (boxplot(dat, notch = T))

Die Kerbe überdeckt mit einer Wahrscheinlichkeit von 95% den Median der Population oder Grundgesamtheit (nicht der Stichprobe) der im Allgemeinen aber nicht bekannt ist. Wenn sich zwei Kerben nicht überschneiden, dann ist die Evidenz gross, dass die Stichproben von unterschiedlichen Populationen stammen.

Mosaicplot – zwei kategorielle Variablen

Merkmale:

- Fläche ist proportional zur Anzahl Beobachtung für die Merkmalskombination in der Stichprobe.
- Breite der Säulen ist proportional zur relativen Häufigkeit der ersten Variable, die Höhe proportional zur zweiten Variable.
- Die Anzahl Beobachtungen ist nicht ablesbar.
- Vertauschen der Variablen führt zu einem anderen Mosaikplot

Stripchart - Kategoriell und metrisch

Bei bimodalen Verteilungen sind Stripcharts eine Alternative zu dem Boxplot.

```
par(mfrow=c(1,2))
stripchart(einkauf ~ zivilstand, data=kdata, vertical=TRUE, method="stack")
stripchart(einkauf ~ kaufkraft, data=kdata, vertical=TRUE, method='stack')
```

Manipulationen von Strings

gsub

Entfernt gewünschte Zeichen aus den Ausprägungen der einzelnen Variablen

```
censUSA$education <- gsub(pattern = " ", replacement = "", x = censUSA$education)
censUSA$occupation <- gsub(pattern = " ", replacement = "", x = censUSA$occupation)
censUSA$education <- factor(censUSA$education)
censUSA$occupation <- factor(censUSA$occupation)
```

substring

Entfernt von den Variablen/Kategorien die ersten hier bspw. fünf Buchstaben des Namens!

```
names(Erhebung) <- substring(text = names(Erhebung), first = 5)
```

nchar(nzz) → Anzahl der Zeichen im String

```
nzz <- gsub(pattern = "ä", replacement = "ae", x = nzz) → suchen und ersetzen durch
nzz <- gsub(pattern = "ö", replacement = "oe", x = nzz)
nzz <- gsub(pattern = "ü", replacement = "ue", x = nzz)
```

strsplit

Sonderzeichen z.B. ? oder . müssen zwischen [ ] stehen, wenn explizit nach ihnen gesucht werden soll.

```
nzzList <- strsplit(x = nzz, split = "[.]") → Text wird in einzelne Sätze umgewandelt
nzzVec <- unlist(nzzList)
```

→ wandelt ihn in Vektor um, dessen Elemente gerade die Sätze des eingelesenen Textes enthalten

Entfernt überflüssige Leerzeichen am Anfang einiger Sätze und fügt dem Satzenden einen Punkt hinzu.

```
vSel <- which(substr(nzzVec,1,1) == " ")
nzzVec[vSel] <- substr(nzzVec[vSel], start = 2, stop = nchar(nzzVec)[vSel])
nzzVec <- paste(nzzVec, ".", sep ="")
nzzVec
```

grepl

In wie vielen Sätzen kommt der String AHV mindestens einmal vor?

```
sum(grepl(pattern = "AHV",x = nzzVec ))
```

plot Argumente

type

"p" Punkte
"l" verbundene Linien
"o" verbundene Linien und Punkte
"b" nicht verbundene Linien mit Punkten
"c" nicht verbundene Linien ohne Punkte
"s" Linien als Treppe
"h" vertikale Linien (ähnlich zum Barplot)
"n" keine Linien und Punkte (leere Grafik)

log

log = "y" log = "x" log = "xy"

legend()

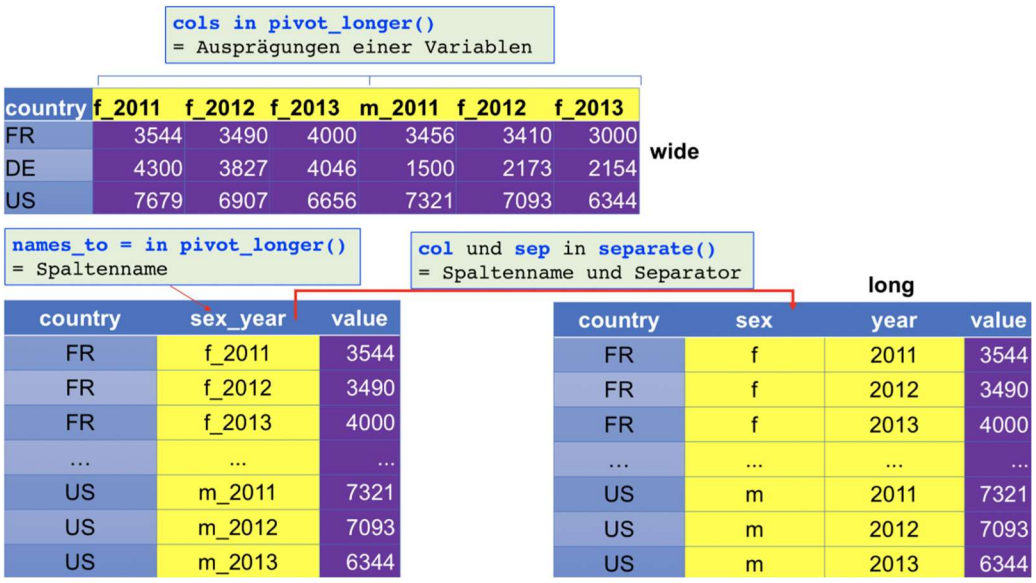
```
legend(x= 4.5, # x-Koordinate, oder "topright"
       y= 35, # y-Koordinate
       title = "Fahrzeugantrieb",
       legend= c("4wd", "Front", "Heck"),  # Text
       col= c("black", "#DF536B", "#61D04F"), # Farbe
       pch= 17, # Symbole lty u. lwd auch möglich
horiz = T # horizontale Auflistung
```

Reshape

Daten vom wide ins long Format transformieren, sodass mit diesen auch «gearbeitet» werden kann.
Dabei werden die Funktionen pivot_longer sowie separate aus dem Paket tidyr verwendet.

```
library(tidyr)
longTBC <- pivot_longer(data = TBCases,
    cols = 2:7, #f_2011 – m_2013
    names_to = "sex_year") #Name der neuen Spalte
```

```
longTBC <- separate(data = longTBC,
        col = "sex_year", #welche Spalte
        sep = "_", #was trennt die Elemente
        into = c("sex", "year")) #neue Spaltennamen
```



country	f_2011	f_2012	f_2013	m_2011	f_2012	f_2013
FR	3544	3490	4000	3456	3410	3000
DE	4300	3827	4046	1500	2173	2154
US	7679	6907	6656	7321	7093	6344

country	sex_year	value
FR	f_2011	3544
FR	f_2012	3490
FR	f_2013	4000
…	…	…
US	m_2011	7321
US	m_2012	7093
US	m_2013	6344

country	sex	year	value
FR	f	2011	3544
FR	f	2012	3490
FR	f	2013	4000
…	…	…	…
US	m	2011	7321
US	m	2012	7093
US	m	2013	6344

Zusammenführen von Datensätzen → merge()

Damit Datensätze zusammengeführt werden können, sollten beide mindestens eine Spalte mit ähnlichem Inhalt (nicht zwingen mit dem gleichen Namen) aufweisen.

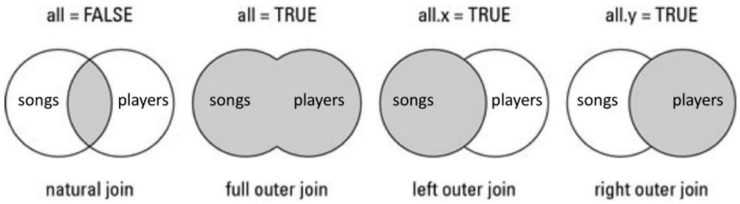
x, y:	x und y Data Frame
by, by.x und by.y:	Name(n) der Spalte(n) des x, y Data Frames, welche für die Vereinigung verwendet werden. Wenn nichts angegeben, wird automatisch anhand gemeinsamer Spaltennamen vereinigt
all:	logical, Art der Zusammenführung (siehe Grafik unten)
all.x:	logical, TRUE = left outer join (siehe Grafik unten)
all.y:	logical, TRUE = right outer join (siehe Grafik unten)
sort:	logical, TRUE = vereinigter Datensatz wird aufsteigend nach den by=("") Spalten sortiert

DF songs

Song	name
Across the Universe	John
Come Together	John
Hello, Goodby	Paul
Peggy Sue	Buddy

DF players

name	plays
George	sitar
John	guitar
Paul	bass
Ringo	drums



Eine der am häufigsten verwendeten Zusammenführungen ist der sog. left outer join, durch den der x DF (im Beispiel songs) mit Beobachtungen aus y DF (players) über gemeinsame Variable(n) (by = c("...")) erweitert wird. Fehlende Beobachtungen in y DF (im Beispiel players) werden zu NA-Werten.

natural join

```
newdat <- merge( x = songs,
                 y = players,
                 by = "name",
                 all = FALSE)
```

```
  name                song  plays
1 John Across the Universe guitar
2 John       Come Together guitar
3 Paul      Hello, Goodbye   bass
```

left outer join

```
newdat <- merge(x = songs,
                y = players,
                by = "name",
                all.x = TRUE)
```

```
   name                song  plays
1 Buddy           Peggy Sue   <NA>
2  John Across the Universe guitar
3  John       Come Together guitar
4  Paul      Hello, Goodbye   bass
```

ggplot2

astehtics:

- color
- shape
- fill
- linetype
- stroke
- size

Pipe-Operator %>%

Insbesondere im Data Science Bereich trifft man in R-Code immer wieder auf den sogenannten Pipe-Operator %>%. Der %>%-Operator wurde 2014 mit dem Paket magrittr eingeführt und erfreute sich rasch grosser Beliebtheit.

Benutz wird die Pipe meisten über das dplyr-Paket bzw. die Paketsammlung tidyverse. Eine Sammlung von R-Paketen, die für Data Science entwickelt wurden. Sie helfen beim managen und «streamlinen» des Daten-Workflows.

Seit neustem (R-Version 4.1.0) wurde der Pipe-Operator auch in Base R integriert. |>

head(mtcars, n = 2)	=	mtcars |> head(n = 2)
head(summary(as.factor(rownames(mtcars))), n = 3)		as.factor() |> summary() |> head(n = 2)
mtcars |> rownames() |>		

Alternative Standardisierung

- Varianzen bzw. Korrelationen sind sensitiv bezüglich Ausreissern → Ergebnis der PCA kann von Ausreissern stark verfälscht werden!
- In der Praxis Standardisierung meistens mit Mittelwert und Standardabweichung.
- Häufig wäre es besser / sicherer die Standardisierung robust durchzuführen (z.B. Median als Lagemass und MAD als Skalenmass) → Ausreisser in den einzelnen Variablen werden besser sichtbar.
  - Die entsprechende Standardisierung muss für prcomp manuell durchgeführt werden und die Hauptkomponenten-Analyse anschliessend mit den manuell standardisierten Daten durchgeführt werden prcomp(dat_robust, scale=FALSE).
  - Ein zweiter Ansatz ist es, die Hauptkomponentenanalyse direkt mit robusten Schätzern durchzuführen. In R gibt es Pakete, die das für uns übernehmen → PcaHubert aus Paket rrcov.

Die robuste PCA wird häufig auch verwendet, um Ausreisser in hochdimensionalen Daten zu detektieren.

Es gibt zwei Arten von Ausreissern.

- Ausreisser im PCA-Raum
  - Grosse Distanz zum Ursprung der PCA
  - Score-Distanz
- Orthogonal Ausreisser (Punkte im Raum der berücksichtigen Hauptkomponenten nicht richtig erfasst werden). Rekonstruktions-Fehler (PCA minimiert die quadrierte Summe dieser Distanzen). Punkte mit grossem Rekonstruktions-Fehler sind Ausreisser!

$$SD_i = \sqrt{\sum_{j=1}^{k} \frac{y_{ij}^2}{\lambda_j}} \text{ (optionale Cutoff-Line } \sqrt{\chi^2_{k.0.975}})$$

$$OD_i = ||x_i - \hat{\mu} - P \cdot y_i^T||$$

Zusammenfassung Hauptkomponentenanalyse

- Drehung (orthogonale Transformation) des p-dimensionalen Raums der zentrierten Originalvariablen in das Hauptkomponentensystem.
- Hauptkomponenten werden so konstruiert, dass sie untereinander unkorreliert sind und in absteigender Weise einen möglichst grossen Teil der Totalvarianz abdecken.
- Die ersten Hauptkomponenten decken ein Grossteil der Varianz ab.
- Visualisierung → Strukturen entdecken → Ausreisser detektieren
- Die wichtigsten Hauptkomponenten können für weiterführende Analysen benutzt werden (Machine Learning).
- Nutzen bzgl. einer Dimensionsreduktion ist besonders hoch, wenn die Originalvariablen stark korreliert sind.

Nachteile:

- Ergebnisse abhängig von der Skalierung.
- Hauptkomponenten schwierig zu interpretieren.
- Anzahl der auszuwählenden bedeutenden Hauptkomponenten ist nicht eindeutig.