

6.6 Regressionsdiagnostik

Einer konventionellen Regressionsanalyse liegen verschiedene Annahmen zugrunde, deren Gültigkeit vorauszusetzen ist, damit die berechneten Standardfehler der Parameterschätzungen und die p -Werte korrekt sind (vgl. Abschn. 11.9.4). Dazu gehören Normalverteiltheit und gemeinsame Unabhängigkeit der Messfehler des Kriteriums, die zudem unabhängig von den Prädiktoren sein müssen. Hinzu kommt die sog. *Homoskedastizität*, also die Gleichheit aller bedingten Fehlervarianzen (Abb. 11.5).

Mit Hilfe der Regressionsdiagnostik soll zum einen geprüft werden, ob die Daten mit den gemachten Annahmen konsistent sind. Zum anderen kann die Parameterschätzung der konventionellen Regression durch wenige Ausreißer überproportional beeinflusst werden. Daher ist es von Interesse, diese zu identifizieren und den Einfluss einzelner Beobachtungen auf das Ergebnis zu bestimmen. Schließlich ist in der multiplen Regression das Ausmaß der Multikollinearität bedeutsam, also die wechselseitige lineare Abhängigkeit der Prädiktoren untereinander. Für eine ausführliche Darstellung vgl. Fox und Weisberg (2011) sowie das zugehörige Paket `car` für Funktionen, mit denen sich eine Vielzahl diagnostischer Diagramme erstellen lassen.

6.6.1 Extremwerte, Ausreißer und Einfluss

Unter einem Ausreißer sind hier Beobachtungen zu verstehen, deren Kriteriumswert stark vom Kriteriumswert anderer Beobachtungen abweicht, die ähnliche Prädiktorwerte besitzen. Extremwerte einer gegebenen Variable zeichnen sich dadurch aus, dass sie weit außerhalb der Verteilung der übrigen Beobachtungen liegen. Die graphische Beurteilung, ob Extremwerte vorliegen, lässt sich getrennt für jede Variable etwa durch Histogramme oder boxplots durchführen (Abb. 6.4; vgl. Abschn. 12.6.1, 12.6.3). Als numerische Indikatoren eignen sich die z -transformierten Werte, an denen abzulesen ist, wie viele Streuungseinheiten ein Wert vom Mittelwert entfernt ist. Als numerisches Maß der multivariaten Extremwertanalyse bietet sich die Mahalanobisdistanz als Verallgemeinerung der z -Transformation an (vgl. Abschn. 11.1.4). Sie repräsentiert den Abstand eines Datenpunkts zum Zentroid der Verteilung, wobei der Gesamtform der Verteilung i. S. der Kovarianzmatrix Rechnung getragen wird. Für die graphische Darstellung der gemeinsamen Verteilung von zwei oder drei Variablen vgl. Abschn. 12.6.8, 12.8.¹⁸

```
> Xpred <- cbind(height, age, sport)      # Prädiktoren als Datenmatrix
> boxplot(Xpred, main="Verteilung der Prädiktoren") # boxplots
> Xz <- scale(Xpred)                     # z-Transformierte
> summary(Xz)
```

	height	age	sport
Min.	:-2.534e+00	Min. :-1.952e+00	Min. :-1.796e+00
1st Qu.:	-6.040e-01	1st Qu.:-7.110e-01	1st Qu.:-6.884e-01
Median :	1.075e-02	Median :-1.069e-01	Median :-9.467e-02
Mean :	1.781e-15	Mean :-8.978e-17	Mean : 1.087e-16
3rd Qu.:	6.338e-01	3rd Qu.: 7.411e-01	3rd Qu.: 6.519e-01

¹⁸Da Extremwerte die Streuungen mit beeinflussen, sollten evtl. robuste Schätzer für Varianzen und Kovarianzmatrizen in Betracht gezogen werden, die etwa an das Argument `cov` von `mahalanobis()` übergeben werden können (vgl. Abschn. 2.7.8). Für fortgeschrittene Tests, ob Ausreißer in multivariaten Daten vorliegen, vgl. das Paket `mvoutlier` (Filzmoser & Gschwandtner, 2012).

```

Max.      : 2.200e+00  Max.      : 2.839e+00  Max.      : 2.727e+00

# Mahalanobis-Distanz der Beobachtungen zum Zentroid der Prädiktoren
> ctrX    <- colMeans(Xpred)                # Zentroid
> sX      <- cov(Xpred)                    # Kovarianzmatrix
> mahaSq  <- mahalanobis(Xpred, ctrX, sX)   # quadrierte M-Distanzen
> summary(sqrt(mahaSq))
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.2855 1.1640 1.5940 1.5970 1.9750 3.3090

```

Durch Extremwerte oder Ausreißer wird die ermittelte Vorhersagegleichung womöglich in dem Sinne verzerrt, dass sie die Mehrzahl der Daten nicht mehr gut repräsentiert. Um den Einfluss der einzelnen Beobachtungen auf die Parameterschätzungen der Regression direkt zu quantifizieren, existieren verschiedene Kennwerte, darunter der Hebelwert h (engl. leverage). h wird durch die Funktion `hatvalues(<lm-Modell>)` berechnet.¹⁹ Für Modelle, die einen absoluten Term b_0 einschließen, kann h Werte im Intervall $[1/n, 1]$ annehmen, wobei n die Anzahl an Beobachtungen ist. Der Mittelwert ist dann gleich $(p+1)/n$ mit $p+1$ als Anzahl zu schätzender Parameter der Regression (p Prädiktoren sowie absoluter Term). Um besonders große Hebelwerte zu identifizieren, kann ihre Verteilung etwa über ein Histogramm oder einen spike-plot veranschaulicht werden (Abb. 6.4, vgl. Abschn. 12.2). Als numerisches Kriterium für auffällig große Hebelwerte dient bisweilen das Zwei- bis Dreifache seines Mittelwerts.

```

> fitHAS <- lm(weight ~ height + age + sport) # Regression
> h      <- hatvalues(fitHAS)                # Hebelwerte
> hist(h, main="Histogramm der Hebelwerte")  # Histogramm
> summary(h)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.01082 0.02368 0.03568 0.04000 0.04942 0.12060

```

Zur Kontrolle lässt sich die Beziehung nutzen, dass die quadrierte Mahalanobisdistanz einer Beobachtung i zum Zentroid der Prädiktoren gleich $(n-1)(h_i - 1/n)$ ist.

```

> all.equal(mahaSq, (N-1) * (h - (1/N)))
[1] TRUE

```

Die Indizes DfFITS bzw. DfBETAS liefern für jede Beobachtung ein standardisiertes Maß, wie stark sich die Vorhersagewerte (DfFITS) bzw. jeder geschätzte Parameter (DfBETAS) tatsächlich ändern, wenn die Beobachtung aus den Daten ausgeschlossen wird. In welchem Ausmaß sich dabei der Standardschätzfehler ändert, wird über das Verhältnis beider resultierenden Werte (mit bzw. ohne ausgeschlossene Beobachtung) ausgedrückt. Cooks Distanz ist ein weiteres Einflussmaß, das sich als $E^2/(\hat{\sigma}^2(1-h)^2) \cdot h/(p+1)$ berechnet, wobei E für die Residuen $Y - \hat{Y}$ und $\hat{\sigma}$ für den Standardschätzfehler der Regression mit $p+1$ Parametern steht.

Mit `influence.measures(<lm-Modell>)` lassen sich die genannten Kennwerte gleichzeitig berechnen. Auffällige Beobachtungen können aus der zurückgegebenen Liste mit `summary()` extrahiert werden, wobei der Wert der abweichenden diagnostischen Größe durch einen Stern * markiert ist.

¹⁹Zudem ist h_i gleich dem i -ten Eintrag H_{ii} in der Diagonale der Hat-Matrix H (vgl. Abschn. 6.3.1).

```

> inflRes <- influence.measures(fitHAS)      # Diagnosegrößen
> summary(inflRes)                          # auffällige Beobachtungen
Potentially influential observations of
lm(formula = weight ~ height + age + sport) :
      dfb.1 dfb.hght dfb.age dfb.sprt dffit  cov.r  cook.d  hat
6   -0.05    0.07   -0.23    0.25  0.40  0.86_*  0.04  0.03
13  -0.21    0.21   -0.10    0.36  0.41  1.13_*  0.04  0.12_*
30   0.19   -0.23    0.17   -0.01 -0.42  0.74_*  0.04  0.02
31  -0.08    0.11   -0.08   -0.05  0.27  0.86_*  0.02  0.01
67   0.02   -0.02   -0.03    0.03 -0.05  1.17_*  0.00  0.11
72  -0.03    0.03    0.04    0.01  0.05  1.17_*  0.00  0.11
86   0.10   -0.16    0.60   -0.25  0.68_*  0.90    0.11  0.08
97  -0.01    0.01   -0.01    0.01  0.02  1.16_*  0.00  0.10

```

In der Ausgabe beziehen sich die Spalten `dfb.` (Prädiktor) auf das DfBETA jedes Prädiktors (inkl. des absoluten Terms 1 in der ersten Spalte), `dffit` auf DfFITS, `cov.r` auf das Verhältnis der Standardschätzfehler, `cook.d` auf Cooks Distanz und `hat` auf den Hebelwert. Für Funktionen zur separaten Berechnung der Maße vgl. `?influence.measures`.

```

> cooksDst <- cooks.distance(fitHAS)        # Cooks Distanz
> plot(cooksDst, main="Cooks Distanz", type="h") # spike-plot

# manuelle Berechnung
> P <- 3                                     # Anzahl Prädiktoren
> E <- residuals(fitHAS)                    # Residuen
> MSE <- sum(E^2) / (N - (P+1))             # quadr. Standardschätzfehler
> CD <- (E^2 / (MSE * (1-h)^2)) * (h / (P+1)) # Cooks Distanz
> all.equal(cooksDst, CD)                  # Kontrolle
[1] TRUE

```

Graphisch aufbereitete Informationen über den Einfluss einzelner Beobachtungen sowie über die Verteilung der Residuen (s. u.) liefert auch eine mit `plot(<lm-Modell>, which=1:6)` aufrufende Serie von Diagrammen. Über das Argument `which` können dabei einzelne Graphiken der Serie selektiv gezeigt werden. Vergleiche dazu auch die Funktionen `influencePlot()` und `influenceIndexPlot()` aus dem `car` Paket.

6.6.2 Verteilungseigenschaften der Residuen

Anhand verschiedener graphischer Darstellungen der Residuen einer Regression lässt sich heuristisch beurteilen, ob die vorliegenden Daten mit den Voraussetzungen der Normalverteilung, Unabhängigkeit und Homoskedastizität der Messfehler vereinbar sind. Als Grundlage können die Residuen $E = Y - \hat{Y}$ selbst, oder aber zwei Transformationen von ihnen dienen: Die *standardisierten* und *studentisierten* Residuen besitzen eine theoretische Streuung von 1 und ergeben sich als $E/(\hat{\sigma}\sqrt{1-h})$, wobei $\hat{\sigma}$ eine Schätzung der theoretischen Fehlerstreuung ist (vgl. Abschn. 11.9.4).

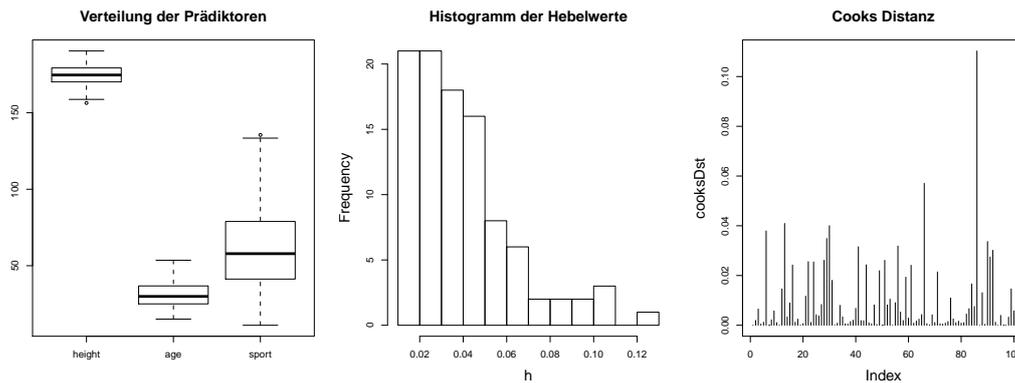


Abbildung 6.4: Beurteilung von Extremwerten und Einflussgrößen in der Regression

Für die standardisierten Residuen wird der Standardschätzfehler als globale Schätzung in der Rolle von $\hat{\sigma}$ verwendet, bei den studentisierten Residuen dagegen eine beobachtungswise Schätzung $\hat{\sigma}_i$. Dabei wird im Fall sog. *extern* studentisierter Residuen (engl. leave-one-out) $\hat{\sigma}_{(i)}$ für jede Beobachtung i auf Basis des Regressionsmodells berechnet, in das alle Daten bis auf die der i -ten Beobachtung einfließen. Für die im folgenden aufgeführten Prüfmöglichkeiten werden oft standardisierte oder studentisierte Residuen E gegenüber vorgezogen.

Die `lm.influence()` Funktion speichert $\hat{\sigma}_{(i)}$ in der Komponente `sigma` der ausgegebenen Liste. Die Residuen selbst lassen sich mit den Funktionen `residuals()` für E , `rstandard()` für die standardisierten und `rstudent()` für die extern studentisierten Residuen ermitteln. Bei allen Funktionen ist als Argument ein von `lm()` erstelltes Modell zu übergeben.

```
> Estnd <- rstandard(fitHAS) # standardisierte Residuen
> all.equal(Estnd, E / sqrt(MSE * (1-h))) # manuelle Kontrolle
[1] TRUE
```

```
# studentisierte Residuen und manuelle Kontrolle
> Estud <- rstudent(fitHAS)
> all.equal(Estud, E / (lm.influence(fitHAS)$sigma * sqrt(1-h)))
[1] TRUE
```

Für eine visuell-exploratorische Beurteilung der Normalverteiltetheit wird die Verteilung der bevorzugten Residuen-Variante mit einem Histogramm der relativen Klassenhäufigkeiten dargestellt, dem die Dichtefunktion der Standardnormalverteilung hinzugefügt wurde (Abb. 6.5, vgl. Abschn. 12.6.1). Das Histogramm sollte in seiner Form nicht stark von der Dichtefunktion abweichen. Zudem lässt sich ein Q-Q-plot nutzen, um die empirischen Quantile der Residuen mit jenen der Standardnormalverteilung zu vergleichen. Die Datenpunkte sollten hier auf einer Geraden liegen (Abb. 6.5, vgl. Abschn. 12.6.5). Für einen inferenzstatistischen Test mit der H_0 , dass Normalverteilung vorliegt, bietet sich jener nach Shapiro-Wilk an.

```
> hist(Estud, main="Histogramm stud. Residuen", freq=FALSE ) # Histogramm

# Dichtefunktion der Standardnormalverteilung hinzufügen
> curve(dnorm(x, mean=0, sd=1), col="red", lwd=2, add=TRUE)
> qqnorm(Estud, main="Q-Q-Plot stud. Residuen") # Q-Q-plot
```

```
> qqline(Estud, col="red", lwd=2)                # Referenzgerade für NV

# Shapiro-Wilk-Test auf Normalverteilung der studentisierten Residuen
> shapiro.test(Estud)
Shapiro-Wilk normality test
data:  Estud
W = 0.9879, p-value = 0.4978
```

Soll eingeschätzt werden, ob die Annahme von Homoskedastizität plausibel ist, kann die bevorzugte Residuen-Variante auf der Ordinate gegen die Vorhersage auf der Abszisse abgetragen werden (sog. spread-level-plot, Abb. 6.5).²⁰ Die Datenpunkte sollten überall gleichmäßig um die 0-Linie streuen. Anhand desselben Diagramms kann auch die Unabhängigkeit der Messfehler heuristisch geprüft werden: Die Residuen sollten eine Verteilung aufweisen, die nicht systematisch mit der Vorhersage zusammenhängt.²¹

```
# studentisierte Residuen gegen Vorhersage darstellen
> plot(fitted(fitHAS), Estud, pch=20, xlab="Vorhersage",
+      ylab="studentisierte Residuen", main="Spread-Level-Plot")

> abline(h=0, col="red", lwd=2)                # Referenz für Modellgültigkeit
```

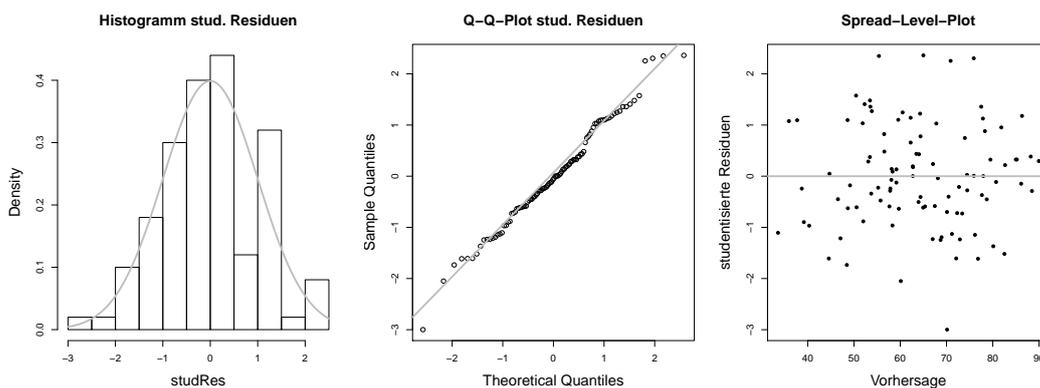


Abbildung 6.5: Graphische Prüfung der Verteilungsvoraussetzungen für eine Regressionsanalyse

Für manche Fälle, in denen die Daten darauf hindeuten, dass die Verteilung einer Variable Y nicht den Voraussetzungen genügt, können streng monotone Transformationen die Verteilung günstig beeinflussen, bei echt positiven Daten etwa Potenzfunktionen der Form Y^λ : Dazu zählen der Kehrwert Y^{-1} , der Logarithmus $\ln Y$ (per definitionem für $\lambda = 0$), die Quadratwurzel $Y^{1/2}$ (z. B. bei absoluten Häufigkeiten), oder der Arkussinus der Quadratwurzel $\arcsin Y^{1/2}$ (z. B. bei Anteilen). In der Regression finden häufig Box-Cox-Transformationen $(Y^\lambda - 1)/\lambda$ für $\lambda \neq 0$ bzw.

²⁰Mitunter werden hierfür auch die Beträge der Residuen bzw. deren Wurzel gewählt (sog. scale-location plot). Vergleiche weiterhin die Funktion `residualPlots()` aus dem Paket `car`. Der Breusch-Pagan-Test auf Heteroskedastizität kann mit der Funktion `bptest()` aus dem Paket `lmtest` (Zeileis & Hothorn, 2002) durchgeführt werden, jener nach White mit `white.test()` aus dem Paket `tseries` (Trapletti & Hornik, 2012).

²¹Der Durbin-Watson-Test auf Autokorrelation der Messfehler lässt sich mit der Funktion `durbinWatsonTest()` aus dem Paket `car` durchführen.

$\ln Y$ für $\lambda = 0$ Verwendung, die ebenfalls echt positive Daten voraussetzen. Für sie stellt das Paket `car` die in Kombination miteinander zu verwendenden Funktionen `powerTransform()` und `bcPower()` bereit.

```
> powerTransform(<lm-Modell>, family="bcPower")
> bcPower(<Vektor>, <lambda>)
```

Als erstes Argument von `powerTransform()` kann ein mit `lm()` erstelltes Modell angegeben werden. Für die Maximum-Likelihood-Schätzung des Parameters λ ist das Argument `family` auf "bcPower" zu setzen. λ erhält man aus dem zurückgegebenen Objekt durch die Funktion `coef()`. Die Box-Cox-Transformation selbst führt `bcPower()` durch und benötigt dafür als Argument zum einen den Vektor der zu transformierenden Werte, zum anderen den Parameter λ der Transformation.

```
> library(car) # für boxCox(), powerTransform()
> lamObj <- powerTransform(fitHAS, family="bcPower")
> (lambda <- coef(lamObj)) # max log-likelihood lambda
      Y1
1.057033

# transformiertes Kriterium und manuelle Kontrolle
> wTrans <- bcPower(weight, coef(lambda)) # Transformation
> all.equal(wTrans, ((weight^lambda) - 1) / lambda) # manuell
[1] TRUE
```

6.6.3 Multikollinearität

Multikollinearität liegt in einer multiplen Regression dann vor, wenn sich die Werte eines Prädiktors gut aus einer Linearkombination der übrigen Prädiktoren vorhersagen lassen. Dies ist insbesondere dann der Fall, wenn Prädiktoren paarweise miteinander korrelieren. Für die multiple Regression hat dies als unerwünschte Konsequenz einerseits weniger stabile Schätzungen der Koeffizienten zur Folge, die mit hohen Schätzfehlern versehen sind. Ebenso kann sich die Parameterschätzung bzgl. desselben Prädiktors stark in Abhängigkeit davon ändern, welche anderen Prädiktoren noch berücksichtigt werden. Andererseits ergeben sich Schwierigkeiten bei der Interpretation der b_j - bzw. der standardisierten b_j^z -Gewichte: Verglichen mit der Korrelation der zugehörigen Variable mit dem Kriterium können letztere unerwartet große oder kleine Werte annehmen und auch im Vorzeichen von der Korrelation abweichen.²²

Ob paarweise lineare Abhängigkeiten vorliegen, lässt sich anhand der Korrelationsmatrix \mathbf{R}_x der Prädiktoren prüfen. Gegebenenfalls sollte auf Prädiktoren verzichtet werden, bei denen eine zu hohe Korrelation vorhanden ist.

```
> (Rx <- cor(cbind(height, age, sport))) # Korrelationsmatrix
           height      age      sport
height 1.00000000 0.06782798 -0.20937559
```

²²Auf numerischer Seite bringt starke Multikollinearität das Problem mit sich, dass die interne Berechnung der Parameterschätzungen anfälliger für Fehler werden kann, die aus der notwendigen Ungenauigkeit der Repräsentation von Gleitkommazahlen in Computern herrühren (vgl. Abschn. 1.3.6).

```
age      0.06782798  1.00000000  0.09496699
sport   -0.20937559  0.09496699  1.00000000
```

Die Diagonalelemente der Inversen \mathbf{R}_x^{-1} liefern den *Varianzinflationsfaktor* VIF_j jedes Prädiktors j als weitere Möglichkeit zur Kollinearitätsdiagnostik: $\sqrt{VIF_j}$ ist der Faktor, um den das Konfidenzintervall für das wahre β_j -Gewicht breiter als im analogen Fall unabhängiger Prädiktoren ist. VIF_j berechnet sich alternativ als $1/(1-R_j^2)$, also als Kehrwert der Toleranz $1 - R_j^2$, wobei R_j^2 der Determinationskoeffizient bei der Regression des Prädiktors j auf alle übrigen Prädiktoren ist. Aus dem Paket `car` stammt die Funktion `vif(<lm-Modell>)`, die als Argument ein durch `lm()` erstelltes lineares Modell erwartet.

```
> library(car)                                # für vif()
> vif(fitHAS)
  height      age      sport
1.054409  1.017361  1.059110
```

In der Ausgabe findet sich unter dem Namen jedes Prädiktors der zugehörige VIF_j -Wert. Da geringe Werte für die Toleranz auf lineare Abhängigkeit zwischen den Prädiktoren hindeuten, gilt dasselbe für große VIF -Werte. Konventionell werden VIF -Werte von bis zu ca. 4 als unkritisch, jene über 10 als starke Indikatoren für Multikollinearität gewertet. Es folgt die manuelle Kontrolle anhand der Regressionen jeweils eines Prädiktors auf alle übrigen.

```
# Regression jeweils eines Prädiktors auf alle übrigen
> fitHeight <- lm(height ~ age + sport)
> fitAge    <- lm(age ~ height + sport)
> fitSport  <- lm(sport ~ height + age)

# VIF_j aus zugehörigem Determinationskoeffizienten R^2
> 1 / (1 - summary(fitHeight)$r.squared)      # VIF height
[1] 1.054409

> 1 / (1 - summary(fitAge)$r.squared)        # VIF age
[1] 1.017361

> 1 / (1 - summary(fitSport)$r.squared)      # VIF sport
[1] 1.05911

# alternativ: Diagonalelemente der Inversen der Korrelationsmatrix
> diag(solve(Rx))
  height      age      sport
1.054409  1.017361  1.059110
```

Ein weiterer Kennwert zur Beurteilung von Multikollinearität ist die Kondition κ der Designmatrix \mathbf{X} des meist mit standardisierten Variablen gebildeten linearen Modells (vgl. Abschn. 11.1.5, 11.9.1). Werte von $\kappa > 20$ sprechen einer Faustregel folgend für Multikollinearität. Zur Berechnung dient die `kappa(<lm-Modell>)` Funktion. Neben κ eignen sich zur differenzierteren Diagnose auch die Eigenwerte von $\mathbf{X}^t \mathbf{X}$ selbst sowie ihr jeweiliger Konditionsindex.²³

²³Fortgeschrittene Methoden zur Diagnostik von Multikollinearität enthält das Paket `perturb` (Hendrickx, 2012).

```

# Regressionsmodell mit standardisierten Prädiktoren
> fitScl <- lm(scale(weight) ~ scale(height) + scale(age) + scale(sport))
> kappa(fitScl, exact=TRUE)
[1] 1.279833

> X      <- model.matrix(fitScl)          # Designmatrix
> (eigVals <- eigen(t(X) %*% X)$values)   # Eigenwerte von  $X^t * X$ 
[1] 119.93081 103.85018 100.00000 73.21902

# Konditionsindizes: jeweils Wurzel aus Eigenwert / (Minimum != 0)
> sqrt(eigVals / min(eigVals[eigVals >= .Machine$double.eps]))
[1] 1.279833 1.190945 1.168660 1.000000

```

Wenn von den ursprünglichen Variablen zu zentrierten oder standardisierten Variablen übergegangen wird, ändern sich die VIF_j Werte nur dann, wenn multiplikative Terme, also Interaktionen in der Regression einbezogen sind (vgl. Abschn. 6.3.4). Dagegen ändert sich κ bei solchen Variablentransformationen praktisch immer.²⁴

```

# VIF: Regression mit standardisierten Variablen -> kein Unterschied
> vif(fitScl)
scale(height) scale(age) scale(sport)
      1.054409   1.017361   1.059110

# kappa: Regression mit ursprünglichen Variablen -> Unterschied
> kappa(lm(weight ~ height + age + sport), exact=TRUE)
[1] 4804.947

```

²⁴Ursache dafür ist die Änderung der Eigenwerte bei Datentransformationen: Ist \mathbf{X} die Designmatrix des ursprünglichen Modells und \mathbf{X}' die Designmatrix des Modells der transformierten Daten, so gehen die Eigenwerte von $\mathbf{X}'^t \mathbf{X}'$ nicht auf einfache Weise aus denen von $\mathbf{X}^t \mathbf{X}$ hervor. Insbesondere verändern sich der größte und kleinste Eigenwert jeweils unterschiedlich, so dass deren Quotient nicht konstant ist.