Datenanalyse mit R

Beschreiben, Explorieren, Schätzen und Testen

Peter Sedlmeier
Markus Burkhardt

Pearson

EXTRAS ONLINE

Datenanalyse mit R

Beschreiben, Explorieren, Schätzen und Testen

Peter Sedlmeier
Markus Burkhardt

Pearson

det, wenn es theoretische Vorarbeiten gibt, die *Kausalbezüge* zwischen zwei Variablen nahelegen. Mittels der Regressionsrechnung kann dann ein entsprechendes (Kausal-) Modell überprüft werden. Oft wird die Regressionsrechnung aber „nur" zur Vorhersage benutzt, ohne genau zu wissen, warum die entsprechende Vorhersage funktioniert: Dann spricht man nicht von kausalen, sondern von *funktionalen* Zusammenhängen. Man kann also mithilfe der Regressionsrechnung sowohl Modelle überprüfen als auch Vorhersagen machen.

Wir werden zunächst kurz erklären, wie die Regressionsrechnung funktioniert, und das Verfahren anhand einiger Beispiele illustrieren. Danach erläutern wir, wann es nicht (oder nicht unmittelbar) sinnvoll ist, eine Regressionsanalyse durchzuführen. Und schließlich befassen wir uns damit, wie man auch nichtlineare Zusammenhänge sichtbar machen und sie unter Umständen „geradebiegen" kann, um so mit den transformierten Werten doch noch eine sinnvolle Regressionsanalyse durchführen zu können.

4.1 Regression: Wie funktioniert sie?

Bei einer Vorhersagevariablen *x*, dem sogenannten *Prädiktor*, und einer vorhergesagten Variablen *y*, dem sogenannten *Kriterium*, ist das die Gleichung für die *Regressionsgerade*:

$$\hat{y}_i = a + b \times x_i$$

Das „Dach" über dem *y* zeigt an, dass es sich um eine Vorhersage, also einen probabilistischen (keinen deterministischen) Zusammenhang handelt. Die Konstante *a* und der Regressionskoeffizient *b* werden dabei so bestimmt, dass die Summe der quadrierten Abweichungen der Werte von der Regressionsgerade minimal ist. Diese sogenannte *Kleinste-Quadrate-Lösung* ergibt (für eine Ableitung siehe z. B. Bortz & Schuster, 2010):

$$b = \frac{Kovarianz(X,Y)}{s_X^2}$$

und

$$a = \bar{y} - b \times \bar{x}$$

Wie gut diese Lösung für die Vorhersage von *y* aufgrund der Kenntnis von *x* ist,[1] kann mittels dreier Arten von Gütemaßen erfasst werden, die wir uns weiter unten noch etwas genauer ansehen werden: der *Residualwerte*, des *Determinationskoeffizienten* und des *Standardschätzfehlers* (manchmal auch als *Standardfehler der Regression* bezeichnet). Die Residualwerte, die Abweichungen der einzelnen Werte (parallel zur *Y*-Achse) von der Regressionsgeraden, werden oft hauptsächlich visuell überprüft –

1 Der Name *Regressions*analyse kommt aus dem Lateinischen: *regredi* heißt zurückgehen. Es geht also darum, herauszufinden, wie gut die *y*-Werte auf die *x*-Werte zurückführbar sind.

anhand von entsprechenden Streuungsdiagrammen. Der Determinationskoeffizient ist nichts anderes als das Quadrat der Korrelation zwischen Prädiktor und Kriterium (r^2), und das ist wiederum nichts anderes als die durch den Prädiktor im Kriterium vorhergesagte oder aufgeklärte Varianz, also der Anteil der Varianz der vorhergesagten Werte (auf der Regressionsgeraden) an der Varianz der tatsächlichen Werte:

$$r^2 = \frac{Aufgeklärte\,Varianz}{Gesamtvarianz}$$

Und der Standardschätzfehler (s_e) ist die Streuung der Residualwerte, also ein Maß dafür, wie stark die vorhergesagten Werte im Durchschnitt von den tatsächlichen Werten abweichen:

$$s_e = s_y\sqrt{1-r^2}$$

In der psychologischen Fachliteratur wird der Determinationskoeffizient nahezu immer berichtet, Grafiken für die Residualwerte manchmal und der Standardschätzfehler selten.

4.1.1 Ein einfaches Beispiel

Greifen wir das IQ-Beispiel aus dem letzten Kapitel wieder auf und nehmen wir an, wir möchten aufgrund der Kenntnis der Werte in einem verbalen IQ-Test die Werte in einem numerischen IQ-Test vorhersagen. Die dazu nötige Vorhersagegleichung bekommen wir mit der Funktion `lm` (linear model). Diese Funktion erzeugt eine Fülle von Ergebnissen, auf die man dann mit weiteren Funktionen zugreifen kann. Wir weisen also zunächst das Ergebnis der Regressionsrechnung dem Objekt *IQ_reg* zu. *IQ_reg* kann dann später als „Quelle" für detailliertere Ausgaben benutzt werden:

```
attach(IQ_Daten)
IQ_reg <- lm(Num.Test ~ Verb.Test)
```

Dabei gibt die Tilde (~) an, was der Prädiktor ist (rechts von ~) und was das Kriterium (links von ~). Für die Geradengleichung brauchen wir nun die Konstante (englisch: *Intercept*) und den Regressionskoeffizienten. Diese erhalten wir mit der Funktion `coef` (wir runden die Ausgabe der Übersichtlichkeit halber auf zwei Nachkommastellen):

```
round (coef(IQ_reg), 2)
```

```
  (Intercept)    Verb.Test
         4.28         0.36
```

Die Regressionsgerade schneidet also die *Y*-Achse (bei $x = 0$) bei einem numerischen Testwert von 4.28; und wenn sich der Wert für den verbalen Test um einen Punkt erhöht, wird der Wert für den numerischen Test um 0.36 Punkte größer – das ist die Aussage, die im Regressionskoeffizienten steckt. Wenn man diese Werte in die Formel für die Regressionsgleichung einträgt, ergibt sich:

$$\hat{y}_i = 4.28 + 0.36 \times x_i$$

Das illustrieren wir nun auch grafisch. Wir erstellen eine Kombination aus drei Zeichnungen. Zunächst zeichnen wir Regressionsgerade und Konstante, danach die vorhergesagten Werte und schließlich die Residualwerte oder Residuen. Da wir für die erste Abbildung die *X*-Achse etwas verlängern (um den Schnittpunkt der Geraden auf der *Y*-Achse bei $x = 0$ zu zeigen), sollte diese Abbildung etwas breiter sein als die anderen beiden. Wie das mithilfe der Funktion `layout` geht, wird im *Kasten Komplexere Aufteilung des Zeichenfelds* beschrieben. Hier ist die Aufteilung, die wir benutzt haben:

```
layout (matrix(c(0, 0, 2, 2, 1, 1, 2, 2, 1, 1, 3, 3, 0, 0, 3, 3), 4, 4),
        width = c(1, 2, 2, 1))
```

Und nun die erste der drei Abbildungen (wir verlängern mit `xlim` und `ylim` die *X*- und *Y*-Achsen so, dass man die Konstante in der Grafik erkennen kann):

```
plot(Verb.Test, Num.Test, xlim = c(0, 15), ylim = c(4, 11),
     main = "Regressionsgerade und Konstante")
```

Dann zeichnen wir mit der Funktion `abline` die Regressionsgerade ein

```
abline(IQ_reg)
```

zeichnen eine vertikale (`v`) Linie bei $x = 0$ (mit `lty` wird die Art der Linie spezifiziert)

```
abline(v = 0,lty = 3)
```

und fügen schließlich mit der Funktion `text` das Wort „Konstante“ bei $x = 0$ und $y = 4.2$ hinzu, wobei `pos` die Position des Texts in Bezug auf die Koordinaten angibt (1 = unterhalb, 2 = links, 3 = oberhalb, 4 = rechts):

```
text(0, 4.2, "Konstante", pos = 4)
```

Nun die zweite Abbildung: Bei dieser Abbildung fügen wir die vorhergesagten Werte mittels der Funktion `points` hinzu (die darin eingebettete Funktion `fitted` ermittelt die vorhergesagten Werte). Wenn man die Punkte in ▶*Abbildung 4.1* links unten miteinander verbindet erhält man die Regressionsgerade:

```
plot(Verb.Test, Num.Test, main = "Vorhergesagte Werte")
points(Verb.Test, fitted(IQ_reg), pch = 19)
```

Das Attribut `pch` legt die Symbole fest (19 steht für schwarze Punkte).

Und schließlich zeichnen wir noch ein Streuungsdiagramm für die Residuen, die Abweichungen der Werte von den vorhergesagten Werten:

```
Residuen <- resid(IQ_reg)
plot(Verb.Test, Residuen, ylab = "Abweichung von der Vorhersage", main = "Residuen")
```

Wenn wir nun wieder eine Regressionsgerade für die Residuen erstellen, bekommen wir (immer) eine Gerade, die die *Y*-Achse bei 0 schneidet und parallel zur *X*-Achse verläuft:

```
abline(lm(Residuen ~ Verb.Test))
```

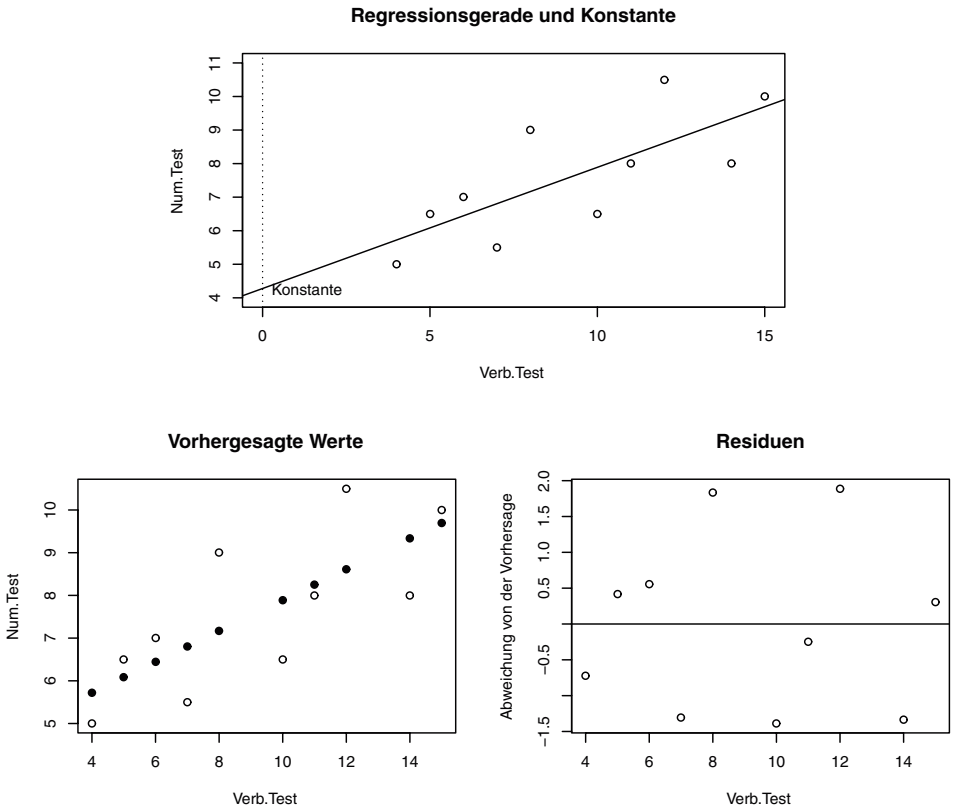

Abbildung 4.1: Illustration der Ergebnisse einer Regressionsanalyse am Beispiel des IQ-Datensatzes (siehe Text für Einzelheiten).

Gütemaße

Die Teilabbildung rechts unten in ▶*Abbildung 4.1* zeigt die Residualwerte, also die Abweichungen aller Werte von ihrem jeweils vorhergesagten Wert. Die entsprechende Punktewolke sollte nicht weit von der Nulllinie entfernt sein und einigermaßen symmetrisch dazu in Form einer Ellipse streuen. Beides trifft in unserem Beispiel zu.

Die Ergebnisse für den Determinationskoeffizienten und den Standardschätzfehler bekommt man mit der Funktion `summary`. Diese Funktion zeigt viele Werte an, auf die wir in späteren Kapiteln noch zurückkommen werden. Hier zeigen wir nur die zwei relevanten Ergebnisse:

```
summary(IQ_reg)
```

```
   Residual standard error: 1.299
   Multiple R-squared:  0.5559
```

```
detach(IQ_Daten)
```

Der Determinationskoeffizient (*Multiple R-squared*) ist also $r^2 = 0.56$ und der Standardschätzfehler (*Residual standard error*) beträgt $s_e = 1.299$. Wenn man also die Werte im verbalen Test kennt, kann man 56% der Varianz der Werte des numerischen Tests vorhersagen; und die Standardabweichung der vorhergesagten Werte von den tatsächlichen Werten ist mit 1.3 Punkten im numerischen Test (dessen Werte ja offensichtlich zwischen 5 und 10.5 variieren) relativ gering.

Komplexere Aufteilung des Zeichenfelds

Wir haben schon in ▶*Kapitel 2* (Lage und Streuungsmaße) die Funktion `par` benutzt, um ein Zeichenfeld für vier Abbildungen zu erstellen. Die Funktion `layout`, die wir Ihnen hier vorstellen, kann alles, was die Funktion `par` kann, ist aber noch weitaus flexibler. Wir demonstrieren Ihnen die Möglichkeiten anhand von zwei Beispielen. Zunächst erstellen wir ein Zeichenfeld für fünf Abbildungen, die in Form eines Kreuzes angeordnet sind (siehe ▶*Abbildung 4.2*):

```
Kreuz <- layout (matrix (c(0, 1, 0,
                           2, 3, 4,
                           0, 5, 0), 3, 3, byrow = T))
```

Das erste Argument für die Funktion `matrix` ist dabei die Liste der Fenster. Wir haben diese Liste so angebracht (auf drei Zeilen aufgeteilt), dass man die einzelnen Fenster schon „sehen“ kann. Eine 0 bedeutet dabei „leeres Fenster“. Nach der „Fensterliste“ muss die Dimensionalität der Matrix angegeben werden – hier also eine 3 × 3 Matrix, und danach haben wir mit dem Argument `byrow = T` gewährleistet, dass die Darstellung der Fenster auch der späteren Aufteilung des Zeichenfelds entspricht. [Nähme man `byrow = F`, die Standardeinstellung, die man auch weglassen kann, werden die Fensterzuordnungen spaltenweise (und nicht zeilenweise wie hier) vergeben.] Man kann sich die Aufteilung auch schon mal ansehen, ohne etwas in die Fenster zu zeichnen. Das geht mit der Funktion `layout.show`:

```
layout.show(Kreuz)
```

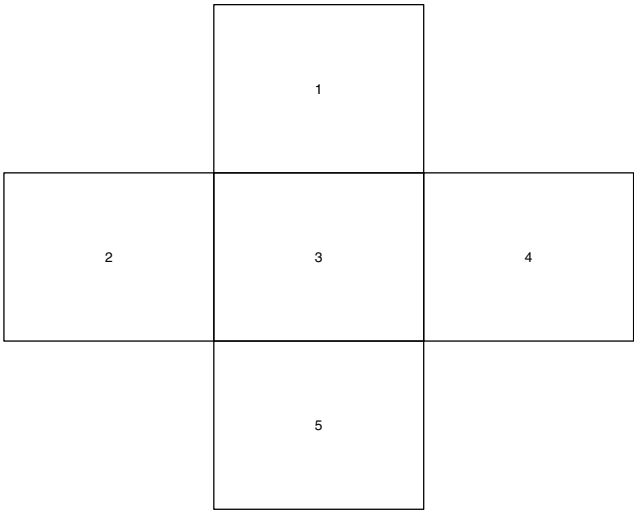

Abbildung 4.2: Aufteilung des Zeichenfelds für fünf Abbildungen, die in Form eines Kreuzes angebracht sind.

Das zweite Beispiel soll illustrieren, wie man unterschiedlich große Zeichenfenster bereitstellen kann. Hierfür gibt es zwei Möglichkeiten. Zunächst können gleiche Zahlen für mehrere aneinandergrenzende Fenster vergeben werden. Die Fenster mit den identischen Zahlen werden dann zu einem Fenster zusammengefasst. Anschließend kann man auch die relative Breite der Spalten (mit `widths`) und die Höhe der Zeilen (mit `heights`) modifizieren. Nehmen wir an, wir wollen drei Abbildungen in eine Grafik zeichnen: oben eine etwas breitere Abbildung zentriert und darunter zwei gleich große. Eine Möglichkeit wäre:

```
Tripel <- layout (matrix(c(0, 1, 1, 0,
                           0, 1, 1, 0,
                           2, 2, 3, 3,
                           2, 2, 3, 3), 4, 4, byrow = T), widths = c(1, 2, 2, 1))
```

Eigentlich haben wir 16 Zeichenfenster (4 × 4 Matrix), die wir jedoch in drei größere Fenster zusammenfassen. Alle Fenster, in denen eine 1 steht, werden zum ersten Fenster zusammengefasst usw. Zusätzlich geben wir noch an, dass die zwei mittleren Spalten doppelt so breit sein sollen wie die äußeren zwei Spalten (`widths = c(1, 2 , 2, 1)`). Die entsprechende Aufteilung des Zeichenfelds ist in ▶*Abbildung 4.3* zu sehen.[2]

2 Im Text verwenden wir dieselbe Aufteilung, zur Abwechslung jedoch mit der Voreinstellung `byrow = F`.

```
layout.show(Tripel)
```

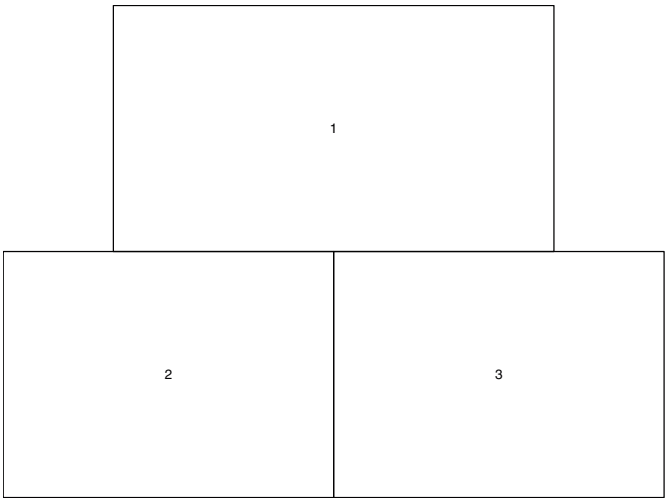

Abbildung 4.3: Aufteilung des Zeichenfelds für drei Abbildungen, wobei die obere Abbildung etwas breiter und zentriert ist.

Nicht vergessen: Nach Beendigung sollten Sie immer die Standardeinstellung für das Zeichenfenster einstellen, z. B. mit `dev.off()`.

4.1.2 Ein komplexeres Beispiel

Nun führen wir eine Regressionsanalyse mit einem komplexeren Datensatz durch, den Werten aus der Vorlesungsbefragung, die wir auch schon im Kapitel zur Korrelation verwendet haben. Wir betrachten die Punktwerte im Deutsch- und im Matheabitur, wieder ohne die Ausreißerwerte (in den Variablen *Deutsch_o_A* und *Mathe_o_A* aus dem letzten Kapitel):

```
attach(ML_I)
DM_reg <- lm(Deutsch_o_A ~ Mathe_o_A)
round (coef(DM_reg), 2)
```

```
(Intercept)   Mathe_o_A
       8.60        0.24
```

Wir können also die Punktewerte im Deutschabitur ($\hat{y}_i$) unter der Kenntnis der Punktewerte im Matheabitur (x_i) mit dieser Regressionsgleichung vorhersagen:

$$\hat{y}_i = 8{,}60 + 0{,}24 \times x_i$$

So wäre beispielsweise die beste Vorhersage für einen Schüler, der 0 Punkte im Matheabitur hat, ein Punktwert von 8.6 im Deutschabitur (die Konstante). Und pro Mathepunkt mehr würde unser Modell 0.24 Punkte mehr im Deutschabitur vorhersagen.

Nun noch Determinationskoeffizient und Standardschätzfehler[3] (mit ausgewählten Ergebnissen der Funktion `summary`):

```
summary(DM_reg)
```

```
   Residual standard error: 1.646
   Multiple R-squared:  0.1089
```

Der Determinationskoeffizient (*Multiple R-squared*) ist also $r^2 = 0.11$ und der Standardschätzfehler (*Residual standard error*) beträgt $s_e = 1.646$.

Nun zeichnen wir noch (nebeneinander) das Streuungsdiagramm mit Regressionsgerade und das dazugehörige Diagramm für die Residualwerte. Hierfür benutzen wir die Funktion `sunflowerplot`, die Sie schon aus dem letzten Kapitel kennen:

```
par(mfrow = c(1, 2))
sunflowerplot(Mathe_o_A, Deutsch_o_A, xlab = "Wert im Matheabitur",
              ylab = "Wert im Deutschabitur", main = "Regressionsgerade")
abline(DM_reg)
Residuen <- resid(DM_reg)
sunflowerplot(Mathe_o_A, Residuen, xlab = "Wert im Matheabitur",
              ylab = "Abweichung von der Vorhersage", main = "Residualwerte")
abline(lm(Residuen ~ Mathe_o_A))
detach(ML_I)
```

Die (einigermaßen) symmetrische Streuung der Residualwerte auf beiden Seiten der Regressionsgerade deutet darauf hin, dass die Beziehung zwischen den beiden Variablen linear ist. In ▶*Kapitel 9* werden wir noch weitere diagnostische Plots für das Ergebnis von Regressionsrechnungen kennenlernen.

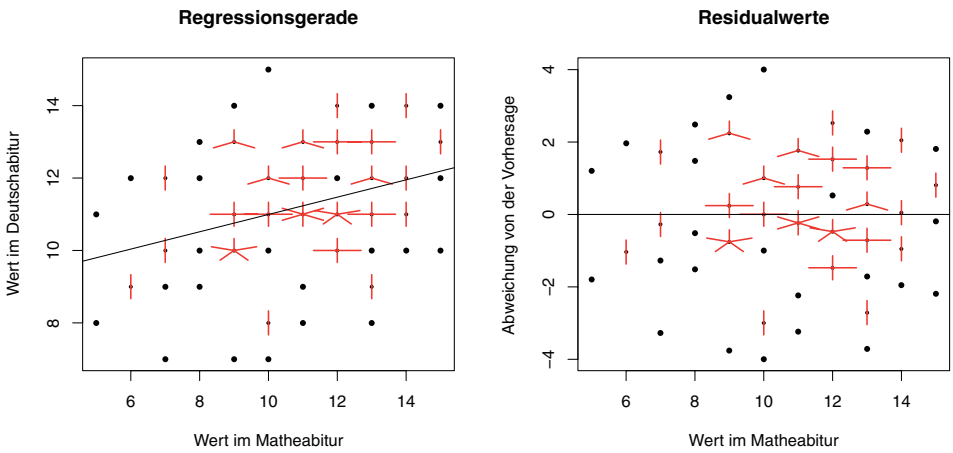

Abbildung 4.4: Streuungsdiagramm mit Regressionsgerade (links) und Residualwerten (rechts) für die Vorhersage der Punkte im Deutschabitur aufgrund der Kenntnis der Mathepunktwerte.

3 R benutzt standardmäßig zur Berechnung der Standardabweichung für y (s_y) die geschätzte Populations-Standardabweichung ($\hat{\sigma}$) und nicht die Stichprobenstandardabweichung (s). Das führt zu tendenziell größeren Werten für den Standardschätzfehler. Bei Bedarf kann der Wert per Hand mithilfe der Funktion `s` aus ▶*Kapitel 2* berechnet werden.

4.1.3 Standardisierte Regressionskoeffizienten

Die Regressionskoeffizienten, die wir bislang berechnet haben, sind in den Originaleinheiten ausgedrückt (z. B. 0.36 Punkte in einem numerischen Intelligenztest). Möchte man die Ergebnisse aus verschiedenen Studien vergleichen, wäre es günstiger, wenn die Regressionskoeffizienten standardisiert wären (wie z. B. die Korrelationskoeffizienten). Das kann einfach dadurch geschehen, dass man die Variablenwerte standardisiert, also *z*-transformiert (siehe ▶*Kapitel 2*). Der standardisierte Regressionskoeffizient wird üblicherweise mit β bezeichnet. Wie Sie sich sicher noch erinnern, haben *z*-transformierte Variablen immer einen Mittelwert von 0. Deswegen ist in diesem Fall auch die Konstante 0:

$$z_a = z_{\bar{y}} - b \times z_{\bar{x}} = 0 - b \times 0 = 0$$

Es bleibt also nur noch:

$$z_{\hat{y}_i} = \beta \times z_{x_i}$$

Illustrieren wir das zunächst für das IQ-Test Beispiel:

```
attach(IQ_Daten)
IQ_reg_stand <- lm(scale(Num.Test) ~ scale(Verb.Test))
coef(IQ_reg_stand)
```

```
    (Intercept) scale(Verb.Test)
   3.572608e-17     7.456011e-01
```

Der Wert für die Konstante (Intercept) ist so klein, dass er praktisch ununterscheidbar von 0 ist (die ersten 16 Nachkommastellen haben den Wert 0). Das ist auch das Ergebnis, das man für eine standardisierte Variable erwarten würde: Bei standardisierten Werten schneidet die Regressionsgerade die *Y*-Achse bei $x = 0$ immer beim Wert 0. Die Regressionsgerade für die standardisierten Koeffizienten wäre also für dieses Beispiel:

$$z_{\hat{y}_i} = 0.75 \times z_{x_i}$$

Der standardisierte Regressionskoeffizient kommt Ihnen vielleicht bekannt vor, und das mit gutem Grund: Er ist identisch mit dem Korrelationskoeffizienten, den wir für dieses Beispiel im vorigen Kapitel berechnet haben. Wie sieht es mit Standardschätzfehler und Determinationskoeffizient aus (ein Teil der Ausgabe ist weggelassen)?

```
summary(IQ_reg_stand)
```

```
   Residual standard error: 0.7068
   Multiple R-squared:  0.5559
```

Der Standardschätzfehler ist natürlich deutlich kleiner, aber der Determinationskoeffizient ist identisch zu dem, den wir für die Originalwerte bekommen haben. Wenn wir ein Streuungsdiagramm mit Regressionsgerade und ein dazugehöriges für die Residualwerte zeichnen, sehen wir, dass sich im Vergleich zu den Ergebnissen in ▶*Abbildung 4.1* außer den Achsenwerten nichts ändert (▶*Abbildung 4.5*):

```
plot(scale(Verb.Test), scale(Num.Test))
abline(IQ_reg_stand)
Residuen_stand <- resid(IQ_reg_stand)
plot(scale(Verb.Test), Residuen_stand)
```

Die Regressionsgerade für die Residualwerte ist immer eine Parallele zur X-Achse mit $y = 0$:

```
abline(lm(Residuen_stand ~ scale(Verb.Test)))
par(mfrow = c(1, 1))
```

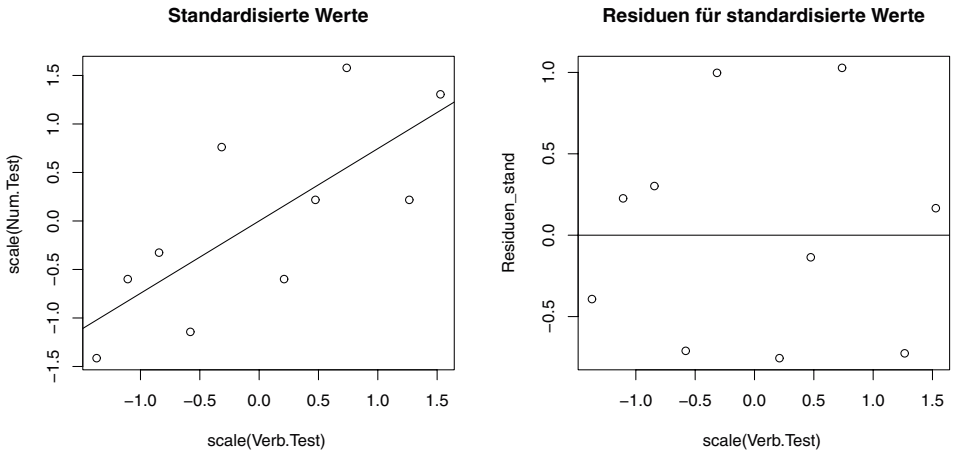

Abbildung 4.5: Regressionsgerade und Residualwerte (Residuen) für standardisierte Werte aus dem IQ-Beispiel.

Der Vollständigkeit halber sehen wir uns auch noch die Ergebnisse für die standardisierten Variablen aus der Vorlesungsbefragung an, diesmal zur Abwechslung mit der Funktion `lm.beta` aus dem gleichnamigen Paket (Behrendt, 2014), die uns direkt die standardisierten Koeffizienten liefert. Im *Abschnitt 4.1.2* haben wir mit der Funktion `lm` die Punktewerte im Deutschabitur aus den Punktewerten im Matheabitur vorhergesagt (ohne Ausreißerwerte) und das Ergebnis in dem Objekt `DM_reg` gespeichert. Das ist nun das Argument für unser `lm.beta`:

```
library(lm.beta)
lm.beta(DM_reg)
```

```
Standardized Coefficients::
(Intercept)    Mathe_o_A
  0.0000000    0.3300671
```

Wieder ist der Wert für den standardisierten Regressionskoeffizienten identisch mit der Korrelation, die wir im vorigen Kapitel berechnet haben. Diese Äquivalenz zwischen standardisiertem Regressionskoeffizient und Korrelationskoeffizient gilt allerdings nur für *einen* Prädiktor (siehe hierzu auch ▶*Kapitel 9*).

Copyright