Regressionsanalyse in R

Session 6

1 Einfache Regression

Lineare Regression ist eines der nützlichsten Werkzeuge in der Statistik. Regressionsanalyse erlaubt es Zusammenhänge zwischen Parametern zu schätzen und somit ein "erklärendes" Model für das Auftreten gewisser Phenomäne zu geben. Wirkliche Kausalität wird durch statistische Analysen dieser Art zwar nicht aufgedeckt, die Ergebnisse aus solchen Analysen können aber Hinweise in diese Richtung geben.

Ein lineares Regressionsmodell ist durch

$$y = X\beta + \epsilon \tag{1}$$

gegeben, wobei X eine Matrix ist, die die Realisierungen der erklärenden Variablen enthält. X hingegen besteht aus n Beobachtungen von k (potentiell) erklärenden Variablen welche in Spalten angeordnet sind – eine Beobachten (aller Variablen) ist eine Spalte in der Matrix, die Matrix hat also Dimension $(n \times k)$. β ist ein $(n \times 1)$ Vektor und stellt die zu schätzenden Koeffizienten der Regression dar. Die Koeffizienten wirken linear (Matrixmultiplikation) auf die Daten X und sollen hierdurch die Variable y (bis auf einen Fehler ϵ) erklären (siehe Gleichung (1)). Der Vektor β wird üblicherweise mit der *Kleinste Quadrate Methode* geschätzt. Hierbei wird die Summe der quadrierten Abweichungen der Schätzwerte aus dem Modell von den wahren Werten also

$$\sum_{i=1}^{n} (y_i - x_i \hat{\beta}_{OLS})^2$$

minimiert ($\hat{\beta}_{OLS}$ ist hierbei der Schätzer für β und x_i ist die i-te Zeile der Matrix X). Für eine grafische Illustration des Verfahrens siehe Abbildung 1.

Diese Schätzung ist in R durch die Funktion *lm(formula, data, subset, weights, na.action, method = "qr")* implementiert. Es kann hier ähnlich wie in dem Befehl *boxplot()* mit der Option *subset* nur eine Teilmenge der Daten verwendet werden. Die Option *weights* kann dazu verwendet werden eine sogenannte *weighted regression* zu spezifizieren. Hierbei werden einzelnen Beobachtungen (oder Gruppen von Beobachtungen) meistens Aufgrund ihrer Varianz Gewichte zugemessen, die sie in der Optimierung der Residuenquadrate haben sollen (die Standardeinstellung ist gleiche Gewichte).

```
> attach(faithful)
> model <- lm(eruptions ~ waiting, faithful)
Call:
lm(formula = eruptions ~ waiting, data = faithful)
Coefficients:
(Intercept)     waiting
-1.87402        0.07563
```

In obigen Beispiel wurde das bereits bekannte *faithful* Datenset geladen und das einfache Model

$$eruptions = \alpha + waiting * \beta + \epsilon$$

geschätzt. Die Funktion *lm(...)* gibt als Rückgabewert ein Objekt der Klasse *lm*, welches neben den Koeffizienten auch noch andere Kennzahlen der Regression enthält (siehe unten). Die Regression wird von

lm() automatisch mit einem *intercept* geschätzt. Will man das verhindern, kann man die Modellgleichung als *eruptions* $\sim$ *0+waiting* spezifizieren.

Bevor man eine Regressionsanalyse durchführt sieht man sich die Daten üblicherweise an und überprüft, ob es sinnvoll ist, einen linearen Zusammenhang zwischen den Variablen zu unterstellen. Dies kann zB mittels eines Scatterplotes geschehen. Die beiden Variablen aus dem Data-Frame *faithful* weisen einen klar linearen Zusammenhang auf (siehe Abbildung 1). Es macht daher Sinn eine lineare Regression durch-

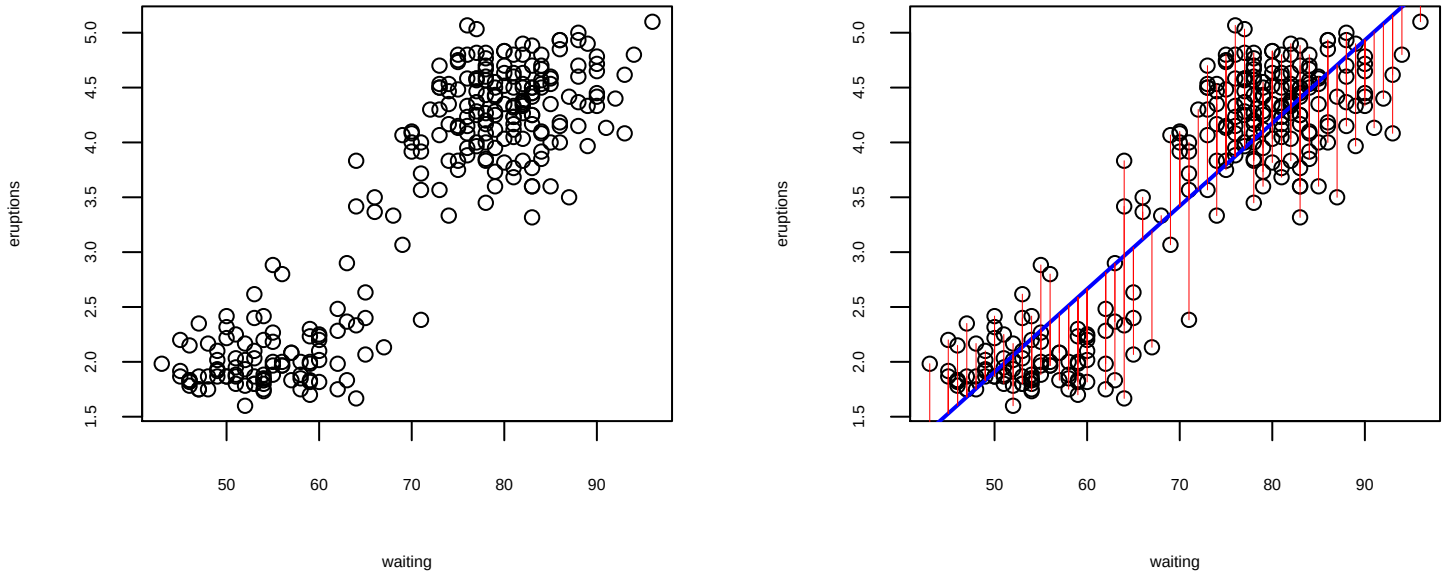

Abbildung 1: Scatterplot *eruptions* gegen *waiting* (links) und Veranschaulichung des kleinste Quadrate Prinzips (rechts).

zuführen. Unterstellt man den Fehlern ϵ eine Normalverteilung, dann kann man testen, ob die geschätzten Parameter signifikant von 0 verschieden sind. Dies hilft abzuklären, ob der geschätzte Einfluß von X auf y nur ein numerisches Artefakt, oder tatsächlich ein statistisch valider Zusammenhang ist. Um festzustellen wieviel der Variation der zu erklärenden Variable, durch das angegebene Model erklärt wird betrachtet man das sogenannte R^2, welches angibt wieviel Prozent der Varianz durch den geschätzten linearen Zusammenhang der Variablen erklärbar ist. Diese und andere Informationen erhält man durch den Befehl *summary()*.

```
> summary(model)
Call:
lm(formula = eruptions ~ waiting)

Residuals:
Min       1Q       Median   3Q       Max
-1.29917 -0.37689 0.03508 0.34909 1.19329

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.874016 0.160143 -11.70 <2e-16 ***
waiting      0.075628 0.002219 34.09 <2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4965 on 270 degrees of freedom
Multiple R-Squared: 0.8115, Adjusted R-squared: 0.8108
F-statistic: 1162 on 1 and 270 DF, p-value: < 2.2e-16
```

In diesem Fall ist klar ersichtlich, dass sowohl der Intercept als auch der Anstiegt der Geraden signifikant von Null verschieden sind. Das R^2 beträgt 80% – man kann also 80% der Varianz der Variable *eruptions* durch das einfache beschriebene Model erklären. Um sich die Konfidenzintervalle der Parameter (unter den entsprechenden Verteilungsannahmen) anzeigen zu lassen, verwendet man die Funktion *confint(object, parm, level = 0.95)*.

```
> confint(model)
                2.5 %       97.5 %
(Intercept) -2.18930436 -1.55872761
waiting      0.07126011   0.07999579
```

Der Befehl

```
> plot(model)
```

zeigt eine Sequenz von Plots die helfen sollen, die Güte der Regression zu beurteilen. Die Plots für die obige Regression sind in Abbildung 2 zu sehen. Die Grafiken zeigen:

1. Einen Plot der Residuen gegen die vom Model vorhergesagten Werte. Optimalerweise sollte hier keine Struktur zu erkennen sein. Die Inhomogenität der analysierten Daten bewirkt das gezeigte "Abwärtsmuster" in dieser Graphik. Dies ist ein Hinweis auf einen Strukturbruch in den Daten (siehe letzte Einheit).

2. Einen QQ-Plot der Quantile der standardisierten Residuen gegen die Quantile der Normalverteilung. Diese Graphik dient der Überprüfung der Normalverteilungshypothese der Residuen, die für die Validität der Tests verantwortlich ist. Die Normalverteilungshypothese scheint in unserem Fall zumindest nicht grob verletzt zu sein.

3. Einen Scale Location Plot, der es ermöglicht zu analysieren, ob die Standardabweichung der Residuen über den Bereich der erklärten Variable gleich bleibt. Dies scheint in unserem Model nicht der

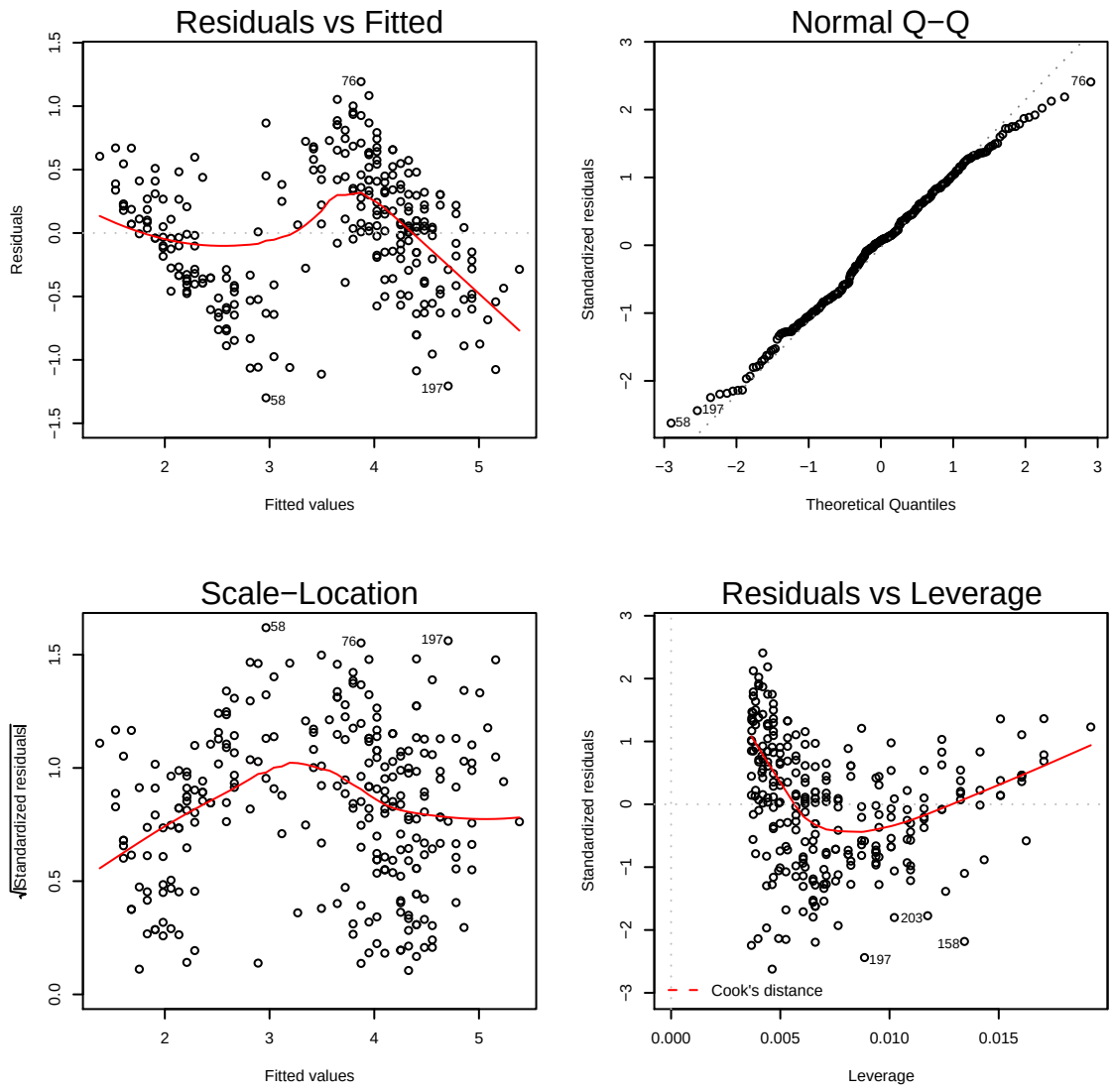

Abbildung 2: Die von *plot(model)* erzeugte Grafiken.

Fall zu sein. Die Annahme der Varianzhomogenität (*Homoskedastizität*), die bei der hier angewandten *OrdinarylLeast-Squares* (OLS) Analyse vorausgesetzt wird, ist in unseren Daten also scheinbar verletzt.

4. Ein Cook's Distance Plot der anzeigt, ob es "einflussreiche" Datenpunkte gibt (also Datenpunkte ohne die das Ergebnis signifikant anders wäre). Werte von über 1 werden in der Regel als besorgniseregend bezeichnet. In unserem Fall ist der einflussreichste Datenpunkte (gemessen in Cook's Distance) mit einem Wert von ca 0.03 nicht beunruhigend.

Ist ein Modell geschätzt, so kann es verwendet werden, um – gegeben Werten für die Variablen in X – eine Vorhersage für den Wert y zu geben. In obigen Beispiel kann also die erwartete Stärke einer Eruption durch Angabe der Wartezeit ermittelt werden. Die Vorhersage in linearen Modellen erfolgt über die generische Funktion *predict(model, newData)* (die auch für andere Modellklassen spezifiziert ist).

```
> nd <- data.frame( waiting = c(25, 60, 100) )
> (pred <- predict(mod, newdata = nd) )
     1          2          3
0.01668271 2.66366089 5.68877881
```

Wird der Funktion zusätzlich der Parameter *interval = "predict"* mitgeliefert, dann besteht die Rückgabe nicht nur aus den vorhergesagten Werten, sondern auch aus den Grenzen des Konfidenzintervalles, indem die wahren Werte (mit einer bestimmten Wahrscheinlichkeit) liegen sollten. Folgendes Beispiel illustriert diese Funktionalität.

```
> x <- seq(from = min(waiting), to = max(waiting), by = 0.01 )
> nd <- data.frame(waiting = x)
> pred <- predict(model, newdata = nd, interval = ''predict'')
> plot(eruptions ~ waiting)
> abline(model, lwd = 2, col = ''blue'')
> lines(x, pred[, 2], lty = 2, col = ''green'')
> lines(x, pred[, 3], lty = 2, col = ''green'')
> pred <- predict(model, newdata = nd, interval = ''predict'', level=0.99)
> lines(x, pred[, 2], lty = 2, col = ''red'')
> lines(x, pred[, 3], lty = 2, col = ''red'')
```

Man beachte die Verwendung des *plot(...)* Befehles mit der Formel als Argument bzw des *ablines(...)* Befehles mit einem Modell als Argument. Das Ergebnis der Plot-Befehle ist in Abbildung 3 zu sehen.

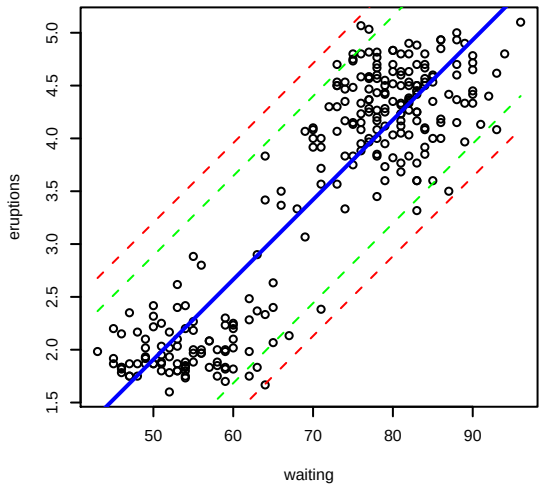

Abbildung 3: Grafische Darstellung der 95% und 99% Konfidenzintervalle für die vorhergesagten Werte aus dem linearen Modell für die Eruptionsstärken.

2 Modellselektion

Oft ist man der Situation, dass man einige Variablen x_i zur Erklärung einer Variable y zur Verfügung hat aber nicht genau weiß, welche der Variablen man für ein Modell verwenden soll. Da die Variablen auch untereinander Korreliert sein können, gibt es auf diese Frage im Allgemeinen keine Eindeutige Antwort. Bei der Selektion eines Modelles stehen sich typischerweise zwei entgegengesetze Ziele gegenüber

1. Das Modell soll minimal sein, also so wenige Terme wie möglich enthalten
2. Das Modell soll die Variationen der Variable y so gut wie möglich erklären

Es ist klar, dass man keines der beiden Ziele zur Gänze erfüllen kann, da

1. ein Hinzufügen eines weiteren Regressors die erklärende Kraft des Modells – ausgedrückt in R^2 – immer erhöht (außer der Regressor ist eine Linearkombination aus anderen Regressoren) und
2. ein Modell mit gar keinem Regressor auch nicht sinnvoll ist

Um einen Mittelweg zu finden bieten sich sogenannte Modellselektionskriterien – wie zum Beispiel *Akaike Information Criterion (AIC)*, *Bayesian Information Criterion* (BIC) oder *adjusted* R^2 – an. Diese Kriterien messen auf die eine oder andere Weise bis zu welchem Grad die beiden obigen Ziele erfüllt sind und drücken dies in einer einzelnen Maßzahl aus, die als Leitfaden für die Auswahl der Regressoren verwendet werden kann.

Im folgenden wird der R interne Datensatz *mtcars* untersucht, der diverse Spezifikationen für einige gängige Autotypen enthält. Ziel der Analyse ist es, den Verbrauch *mpg* (*miles per gallon*) zu erklären. Hierzu können andere Variablen aus dem Datensatz verwendet werden (Gewicht, PS, Anzahl der Gänge, ...). Es ist einerseits zwar klar, dass nicht alle Variablen benötigt werden, andererseits kann a priori nicht ohne weiteres festgestellt werden welche Variablen in welcher Kombination signifikant zur Erklärung beitragen und welche nicht. Im Folgenden wird ein einfaches Modell schrittweise erweitert indem Regressoren aufgenommen werden, wenn Sie zu einer Verbesserung (also Verringerung) im AIC führen. Die Reihenfolge, in der die Variablen getestet werden, ist hierbei vollkommen willkürlich.

```
> attach(mtcars)
> model1 <- lm(mpg ~ hp)
> AIC(model1)
[1] 181.2386
> model2 <- lm(mpg ~ hp + wt)
> AIC(model2)
[1] 156.6523
> model3 <- lm(mpg ~ hp + wt + gear)
> AIC(model3)
[1] 157.0528
> model4 <- lm(mpg ~ hp + wt + cyl)
> AIC(model4)
[1] 155.4766
> model5 <- lm(mpg ~ hp + wt + cyl + qsec)
> AIC(model5)
[1] 157.2218
> model6 <- lm(mpg ~ hp + wt + cyl + hp)
> AIC(model6)
[1] 155.4766
> model7 <- lm(mpg ~ hp + wt + cyl + vs)
> AIC(model7)
[1] 157.4658
```

Es ist zu bemerken, dass bei solchen und ähnlichen Prozeduren nie ein Modell zweifelsfrei als das "richtige" identifiziert werden kann, da unterschiedliche Modellselektionskriterien aber auch unterschiedliche Algorithmen für die Aufnahme bzw den Ausschluss von Variablen zu unterschiedlichen Modellen führen. Auf keinen Fall sollte blind einem der Kriterien vertraut werden.

3 Beispiele

1. Versuche die Variable *mpg* aus dem Data Frame *mtcars* mittels einer linearen Regression durch andere Variablen aus diesem Data Frame zu erklären. Führe hierzu die in Sektion 1 beschriebenen Analysen durch. Illustriere Deine Schlüsse mit Grafiken und Kennzahlen.

2. Versuche die Variable y aus dem Data Frame *regression* (aus *Session6BSP2.R*) mittels einer linearen Regression durch die anderen Variablen in diesem Data-Frame zu erklären. Illustriere Deine Schlüsse mit Grafiken und Kennzahlen.

3. Schreibe eine Modellselektionsfunktion, die als Parameter die zu erklärende Variable, die potentiellen erklärenden Variablen (als Data Frame) und die Modellselektionsmethode (AIC, BIC, adjusted R^2) übergeben bekommt. Die Modellselektion soll folgendermaßen von statten gehen

   (a) schätze das volle Modell $\mathcal{M}$ (also das Modell, dass alle Variablen enthält) und berechne den Wert des Modellselektionskriteriums

   (b) schätze alle Modelle, die eine Variable weniger enthalten als $\mathcal{M}$ und berechne das Modellselektionskriterium für alle diese Modelle.

   (c) i. falls es unter den reduzierten Modellen ein Modell $\mathcal{M}'$ gibt, welches besser (im Sinne des gewählten Kriteriums) ist, dann setze $\mathcal{M} = \mathcal{M}'$ und gehe zu Schritt (b)

   ii. falls dies nicht der Fall ist, dann gib das Modell $\mathcal{M}$ als Ergebnis zurück.

Verwende die Prozedur, um ein optimales Modell für den Treibstoffverbrauch zu finden (Datenset *mtcars*, siehe oben). Vergleiche die Ergebnisse, die Du mit unterschiedlichen Modellselektionsprozeduren erhälst.

4. Wiederhole Aufgabe 3 mit dem Unterschied, dass der Modellselektionsmechanismus nicht schrittweise Regressoren aus dem vollen Modell ausschließt, sondern mit einem einfachen Modell (ein Regressor) anfängt und schrittweise jene Regressoren hinzunimmt, welche die größte Verbesserung in dem gewählten Kriterium bringen. Vergleiche Deine Ergebnisse für unterschiedliche Anfangsmodelle (also unterschiedliche erste Regressoren).

5. Wiederhole Aufgabe 3 und schreibe eine Prozedur, die alle möglichen Modelle (also Modelle mit allen möglichen Subsets aus Regressoren) bezüglich des gewählten Kriteriums miteinander vergleicht. Probiere die Funktion an dem Treibstoffproblem aus. Welche Vor- und Nachteile hat dieser Zugang gegenüber den Varianten in Beispiel 3 und 4 ?