Lineare Modelle in R:
Klassische lineare Regression

Achim Zeileis

2009-02-20

1 Das Modell

Das klassische lineare Regressionsmodell versucht den Zusammenhang zwischen einer abhängigen Variablen (oder Responsevariablen) Y und einer oder mehreren erklärenden Variablen (oder Regressoren oder Prädiktorvariablen) $X_1, \ldots, X_k$ zu modellieren. Dabei ist der Einfluß jeder Variablen linear und der erste Regressor ist normalerweise einfach eine Konstante $X_1 = 1$.

Eine Stichprobe vom Umfang n, an die ein solches Modell angepaßt werden soll, wird üblicherweise so notiert: die Beobachtungen der abhängigen Variablen y_i $(i = 1, \ldots, n)$ und eines Regressorvektors $x_i = (1, x_{i2}, \ldots, x_{ik})^\top$ $(i = 1, \ldots, n)$. Damit läßt sich das Modell für jede Beobachtung i $(i = 1, \ldots, n)$ also folgendermaßen schreiben:

$$\begin{aligned} y_i &= \beta_1 + \beta_2 \cdot x_{i2} + \ldots + \beta_k \cdot x_{ik} + \varepsilon_i \\ &= x_i^\top \beta + \varepsilon_i \end{aligned}$$

Völlig äquivalent kann man es auch in Vektorschreibweise als

$$y = X\beta + \varepsilon$$

schreiben. Ziel ist es nun auf Basis von Daten

- die unbekannten Regressionskoeffizienten β zu schätzen,
- zu beurteilen, ob man überhaupt alle Variablen X_j braucht oder ob nicht vielleicht einige der wahren $\beta_j = 0$ sind (und somit also die zugehörige Variable keinen Einfluß auf Y hat),
- ob das resultierende Modell die Daten gut erklärt,
- mit dem geschätzen Modell, Prognosen durchzuführen.

2 Schätzung der Regressionskoeffizienten

Gesucht ist nun also eine Gerade der Form $\hat{y}_i = x_i^\top \hat{\beta}$, so daß die resultierenden Prognosen $\hat{y}_i$ möglichst gut zu den wahren Beobachtungen y_i passen. Möglichst gut heißt dabei, daß die Summe der quadratischen Fehler minimal wird. Formal: die Summe der quadrierten Residuen (oder Prognosefehler) $\hat{\varepsilon}_i = y_i - \hat{y}_i$ soll minimiert werden. Der zugehörige Kleinste-Quadrate-Schätzer $\hat{\beta}$ muß deshalb die Normalgleichungen

$$(X^\top X)\,\hat{\beta} = X^\top y$$

erfüllen und läßt sich unter geeigneten Voraussetzungen als

$$\hat{\beta} \quad = \quad (X^\top X)^{-1} X^\top y$$

schreiben und ist der beste lineare unverzerrte Schätzer.

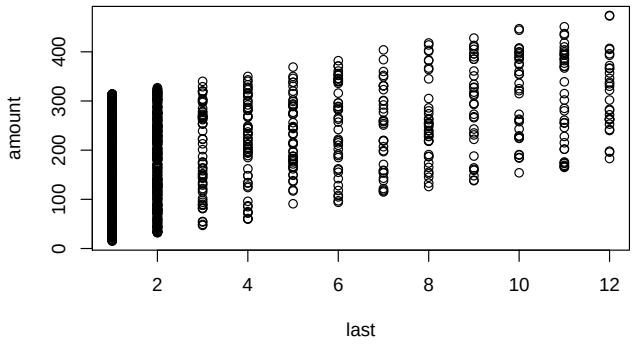
In R kann dieser Schätzer sehr leicht mit dem Befehl `lm` (für 'linear model') berechnet werden. Als Argument übergibt man diesem Datensatz eine Formel, wie wir sie auch schon zur Visualisierung verwendet haben, bspw. `y ~ x + z`. Diese hat auf der linken Seite die abhängige Variable `y` stehen, die erklärt wird (`~`) durch die Regressoren `x` und (`+`) `z`. Falls die Daten in einem Datensatz enthalten sind, kann man `lm` auch noch ein zweites Argument `data` mit dem zughörigen `data.frame` übergeben.

Zur Illustration wird wieder der `BBBClub` Datensatz geladen:

```
R> load("BBBClub.rda")
```

Im Tutorium ‚EDA2.pdf' hatten wir bereits gesehen, daß ein gewisser Zusammenhang zwischen den Gesamtausgaben der Kunden `amount` und der Anzahl der Monate seit ihrem letzten Einkauf `last` vorliegt:

```
R> plot(amount ~ last, data = BBBClub)
```



Daß wir hier das **`data`** *Argument spezifizieren müssen, liegt daran, daß wir den Datensatz nicht attached haben.*

Wenn wir nun das Modell

$$\text{Ausgaben} \quad = \quad \beta_1 \; + \; \beta_2 \cdot \text{Monate} \; + \; \text{Fehler}$$

schätzen wollen, sagen wir in R

```
R> fm <- lm(amount ~ last, data = BBBClub)
R> fm

Call:
lm(formula = amount ~ last, data = BBBClub)

Coefficients:
(Intercept)         last
     156.28        14.09
```

Die Syntax ist also genau dieselbe beim Streudiagramm und beim Anpassen des linearen Modells. Bei der Anzeige des resultierenden angepaßten Modells `fm` zeigt uns R die geschätzten Koeffizienten an. Das geschätzte Modell lautet also

$$\text{Ausgaben} \quad = \quad 156.28 \; + \; 14.09 \cdot \text{Monate} \; + \; \text{Fehler}$$

Damit sind also die Gesamtausgaben um durchschnittlich USD 14.09 höher für jeden Monat, den der letzte Einkauf länger zurückliegt. Oder anders formuliert: Kunden, deren letzter Einkauf vor x Monaten war, geben durchschnittlich USD 14.09 mehr aus als Kunden, deren letzter Einkauf vor $x - 1$ Monaten war.

3 Prognose

Um nun mit dem angepaßten Modell Prognosen durchzuführen muß man lediglich die interessierenden Werte der Erklärungsvariablen in obiges Modell einsetzen. Für den unbekannten Fehler setzt man einfach den zugehörigen Erwartungswert, nämlich 0, ein. Wenn wir also die Prognose ausrechnen wollen für Kunden, deren letzter Einkauf 9 Monate zurückliegt, berechnen wir:

$$\text{Erwartete Ausgaben} \quad = \quad 156.28 \; + \; 14.09 \cdot 9.$$

In R kann dies leicht mit Hilfe der Funktion `predict` durchgeführt werden, der man ein angepaßtes Modell übergibt, sowie einen `data.frame` mit den Beobachtungen der Erklärungsvariablen. Um also die Prognose für Kunden auszurechnen, deren letzter Einkauf 9 oder 10 Monate zurückliegt:

```
R> predict(fm, newdata = data.frame(last = c(9, 10)))

       1        2
283.0654 297.1528
```

4 Inferenz

Um zu überprüfen, ob die verwendeten Regressoren X_j überhaupt einen Einfluß auf Y haben, stehen vor allem zwei Tests zur Verfügung: der sogenannte t-Test und der F-Test. Der t-Test dient dazu die Hypothese zu überprüfen, daß ein bestimmter Koeffizient β_j in Wahrheit 0 ist. Der F-Test dagegen dient dazu, die Hypothese zu testen, ob $q \geq 1$ Koeffizienten gleichzeitig 0 sind, also alle zugehörigen Variablen keinen Einfluß auf Y haben: $\beta_{k-q+1} = \ldots = \beta_k = 0$. Der t-Test ist äquivalent zum F-Test wenn $q = 1$.

4.1 t-Test

Der t-Test für jeden Koeffizienten β_j verwendet als Teststatistik

$$t \quad = \quad \frac{\hat{\beta}_j}{\widehat{SD(\hat{\beta}_j)}}$$

also den standardisierten Quotienten des Schätzers $\hat{\beta}_j$ und seiner geschätzten Standardabweichung $\widehat{SD(\hat{\beta}_j)}$. Wenn dieser weit von 0 abweicht (im Vergleich zu einer t-Verteilung), dann ist der entsprechende Koeffizient signifikant von 0 verschieden und die zugehörige Variable X_j hat einen signifikanten Einfluß auf Y.

In R werden die Schätzer $\hat{\beta}_j$ zusammen mit ihren geschätzten Standardfehlern, den t-Statistiken und p-Werten von

```
R> summary(fm)

Call:
lm(formula = amount ~ last, data = BBBClub)

Residuals:
     Min       1Q   Median       3Q      Max
-155.366  -68.563    5.328   70.644  149.847

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 156.2787     3.4031   45.92   <2e-16 ***
last         14.0874     0.7714   18.26   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 84.44 on 1298 degrees of freedom
Multiple R-squared: 0.2044,        Adjusted R-squared: 0.2038
F-statistic: 333.5 on 1 and 1298 DF,  p-value: < 2.2e-16
```

unter `Coefficients` angezeigt. Daraus können wir nochmal ablesen, daß $\hat{\beta}_2 = 14.0874$ ist bei einer Standardabweichung von $\widehat{SD(\hat{\beta}_j)} = 0.7714$. Damit ist die t-Statistik $t = 18.2615$ und damit hochsignifikant, da sie ihren kritischen Wert 2 (Faustregel) klar überschreitet und der zugehörige p-Wert kleiner als 10^{-16} und damit praktisch 0 ist (und insbesonders kleiner als 0.05). Kurz gesprochen: `last` hat einen hochsignifikanten Einfluß auf `amount`.

Aus dieser Teststatistik läßt sich auch leicht ein Konfidenzintervall für β_j berechnen, nämlich $\hat{\beta}_j \pm 2 \cdot \widehat{SD(\hat{\beta}_j)}$ (Faustregel). Hier ist das also $14.0874 \pm 2 \cdot 0.7714 = [12.5446, 15.6303]$.

5 F-Test

Die Idee des F-Tests ist folgende: man schaut sich an, ob die Fehlerquadratsumme (also die Summe der quadrierten Residuen) signifikant sinkt, wenn man zusätzliche Regressoren in das Modell aufnimmt. Durch Hinzunahme von zusätzlichen Regressoren kann die Fehlerquadratsumme nur sinken, weil ja genau die Fehlerquadratsumme bei KQ-Schätzung minimiert wird. Durch zusätzliche Regressoren kann das Modell also nie schlechter werden; die Frage ist bloß, ob es signifikant besser wird.

Wenn wir also das volle Modell bereits geschätzt haben, schätzen wir auch noch das vereinfachte Modell $\tilde{y}_i = \tilde{\beta}_1 + \tilde{\beta}_2 \cdot x_{i2} + \ldots + \tilde{\beta}_{k-q} \cdot x_{i,k-q}$. Um dann zu überprüfen, ob die zusätzlichen Variablen $X_{k-q+1}, \ldots, X_k$ einen signifikanten Einfluß haben, tun wir folgendes:

- Wir berechnen die Fehlerquadratsumme RSS_1 für das volle Modell $\hat{y}_i$ mit k Parametern.
- Wir berechnen die Fehlerquadratsumme RSS_2 für das vereinfachte Modell $\tilde{y}_i$ mit $k - q$ Parametern, also mit q Parametern weniger.
- Um zu beurteilen, ob RSS_1 signifikant kleiner ist als RSS_2 berechnen wir die F-Statistik

$$F \quad = \quad \frac{(RSS_2 - RSS_1)/q}{RSS_1/(n-k)}$$

  und schauen uns an, ob diese signifikant größer ist als 0.

Was sind jetzt also gute Vergleichsmodelle für unser Modell `fm`? In R wird immer die F-Statistik für den Vergleich des einfachst denkbaren Modells – also einem Modell, das nur eine Konstante enthält – und dem angepaßten Modell berechnet. Dies überprüft also, ob überhaupt einer der Regressoren einen Einfluß auf Y hat. In der letzten der Zeile des Resultats von `summary(fm)` (s.o.) wird genau dieser F-Test angezeigt. Die Statistik $F = 333.5$ ist hochsignifikant, der zughörige p-Wert ist wieder praktisch 0.

Das heißt also, daß alle Regressoren zusammen eine signifikante Verbesserung gegenüber dem trivialen konstanten Modell erzielen. Das ist in diesem Fall ja nur ein einziger Regressor, nämlich `last` und deshalb äquivalent zu dem zugehörigen t-Test, denn $t^2 = F$. Äquivalent heißt hier, daß die p-Werte genau gleich sind.

Eine andere Möglichkeit genau denselben F-Test durchzuführen ist die Funktion `anova`, die generell eine beliebige Anzahl von angepaßten linearen Modellen miteinander vergleichen kann. Hier wollten wir also das Modell `fm` mit dem trivialen Modell

```
R> fm1 <- lm(amount ~ 1, data = BBBClub)
R> fm1

Call:
lm(formula = amount ~ 1, data = BBBClub)

Coefficients:
(Intercept)
      201.4
```

vergleichen. Das tun wir mit

```
R> anova(fm1, fm)

Analysis of Variance Table

Model 1: amount ~ 1
Model 2: amount ~ last
  Res.Df      RSS   Df Sum of Sq      F    Pr(>F)
1   1299 11631929
2   1298  9254317    1   2377612 333.48 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

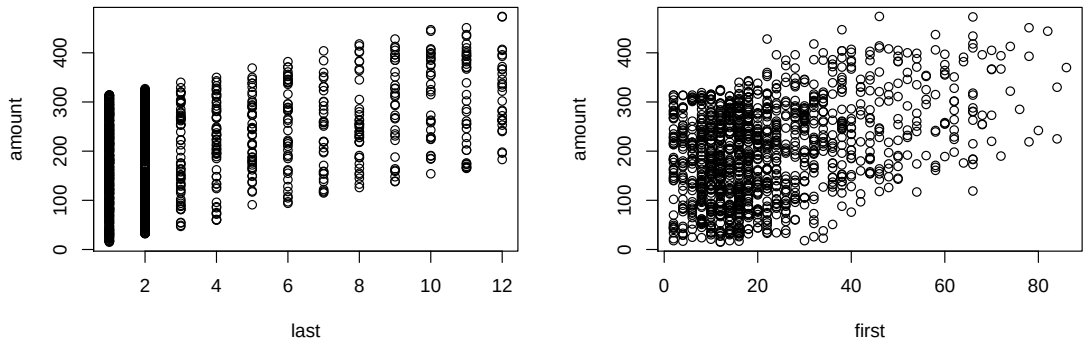
was wieder dieselbe F-Statistik berechnet und denselben hochsignifikanten p-Wert liefert. Die Funktion `anova` kann aber auch weitere Modelle vergleichen wie wir im folgenden Abschnitt sehen werden.

6 Modellwahl

Im Allgemeinen hat man im linearen Regressionsmodell nicht nur einen Regressor als Kandidaten für ein Modell (wie im vorangegangenen Abschnitt) sondern mehrere. Um ein gutes Modell zu finden, probiert man deshalb mehrere oder sogar alle möglichen Modelle aus und versucht daraus das Beste herauszufinden.

Als Illustration verwenden wir wieder als abhängige Variable `amount` und als mögliche Regressoren `last` (wie im letzten Abschnitt) und `first`. Um uns einen groben Eindruck von den Daten zu verschaffen, schauen wir uns Streudiagramme für beide Erklärungsvariablen mit der abhängigen Variablen `amount` an:

```
R> plot(amount ~ last, data = BBBClub)
R> plot(amount ~ first, data = BBBClub)
```



Beide deuten auf einen gewissen Zusammenhang mit der abhängigen Variablen hin. Dann passen wir einmal alle möglichen Kandidaten für ein Modell an: kein Regressor (nur Konstante), nur `last` als Regressor, nur `first` als Regressor und beide Variablen `last` und `first` als Regressoren.

```
R> fm1 <- lm(amount ~ 1, data = BBBClub)
R> fm1

Call:
lm(formula = amount ~ 1, data = BBBClub)

Coefficients:
(Intercept)
      201.4

R> fm2 <- lm(amount ~ last, data = BBBClub)
R> fm2

Call:
lm(formula = amount ~ last, data = BBBClub)

Coefficients:
(Intercept)         last
     156.28        14.09

R> fm3 <- lm(amount ~ first, data = BBBClub)
R> fm3

Call:
lm(formula = amount ~ first, data = BBBClub)

Coefficients:
(Intercept)        first
    151.421        2.241

R> fm23 <- lm(amount ~ last + first, data = BBBClub)
R> fm23
```

```
Call:
lm(formula = amount ~ last + first, data = BBBClub)

Coefficients:
(Intercept)         last        first
   154.3886      13.1389       0.2210
```

Dann überprüfen wir, ob überhaupt ein Regressor einen Einfluß hat

```
R> anova(fm1, fm23)

Analysis of Variance Table

Model 1: amount ~ 1
Model 2: amount ~ last + first
  Res.Df      RSS   Df Sum of Sq      F    Pr(>F)
1   1299 11631929
2   1297  9248760    2   2383169 167.10 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

und sehen daß dies der Fall ist, weil der zugörige p-Wert hochsignifikant ist. Jetzt müssen wir uns also überlegen, ob nur einer oder beide Regressoren für diese Signifikanz verantwortlich sind. Also schauen wir einmal den Verlauf an, wie sich die Güte des Modells entwickelt, wenn erst `last` und dann `first` in das Modell aufgenommen wird bzw. umgekehrt wenn erst `first` und dann `last` in das Modell aufgenommen wird.

```
R> anova(fm1, fm3, fm23)

Analysis of Variance Table

Model 1: amount ~ 1
Model 2: amount ~ first
Model 3: amount ~ last + first
  Res.Df      RSS   Df Sum of Sq       F    Pr(>F)
1   1299 11631929
2   1298  9952353    1   1679576 235.535 < 2.2e-16 ***
3   1297  9248760    1    703593  98.668 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hier erzielen wir durch Hinzunahme von `first` zum trivialen Modell eine hochsignifikante Verbesserung und die weitere Hinzunahme von `last` ist nochmal hochsignifikant. Wenn wir aber nun

```
R> anova(fm1, fm2, fm23)

Analysis of Variance Table

Model 1: amount ~ 1
Model 2: amount ~ last
Model 3: amount ~ last + first
  Res.Df      RSS   Df Sum of Sq        F Pr(>F)
1   1299 11631929
```

```
2   1298  9254317    1   2377612 333.4244 <2e-16 ***
3   1297  9248760    1      5557   0.7793 0.3775
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

anschauen, sehen wir, daß die Hinzunahme von `last` zum trivialen Modell eine hochsignifikante Verbesserung erzielt, aber die weitere Hinzunahme von `first` das Modell nicht weiter verbessert, da der zugehörige p-Wert mit $p = 0.3775$ größer als 0.05 ist. Das heißt also, daß `last` einen signifikanten Erklärungswert hat, nicht aber `first`. Man sollte also das Modell `fm2` (mit `amount ~ last`) verwenden.

In diesem speziellen Fall hätte man all diese Information auch aus dem einzigen Befehl

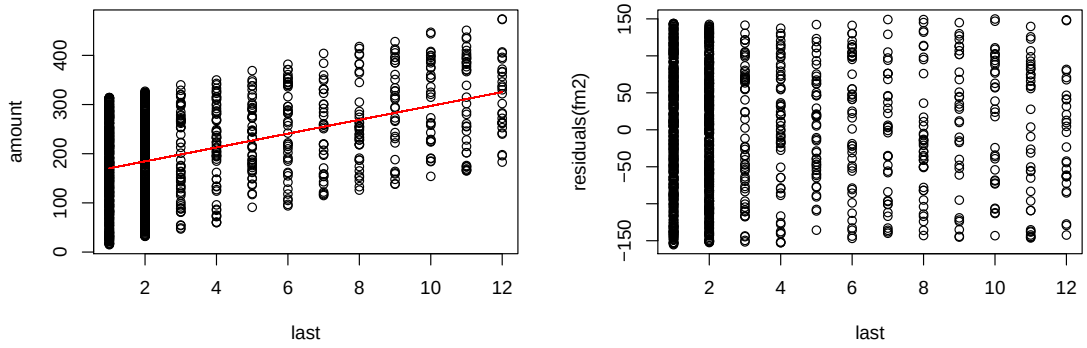
```
R> summary(fm23)

Call:
lm(formula = amount ~ last + first, data = BBBClub)

Residuals:
     Min       1Q   Median       3Q      Max
-156.159  -69.401    5.728   70.721  152.823

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 154.3886     4.0208  38.397   <2e-16 ***
last         13.1389     1.3227   9.933   <2e-16 ***
first         0.2210     0.2504   0.883    0.378
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 84.44 on 1297 degrees of freedom
Multiple R-squared: 0.2049,        Adjusted R-squared: 0.2037
F-statistic: 167.1 on 2 and 1297 DF,  p-value: < 2.2e-16
```

ablesen können. Zunächst zeigt uns dessen F-Statistik an, daß das Modell signifikant besser ist als das triviale Modell. Dies ist derselbe Test wie in `anova(fm1, fm23)` (s.o.). Dann sehen wir, daß `last` einen hochsignifikanten Einfluß hat (der Test ist äquivalent zu `anova(fm3, fm23)`) aber daß der Koeffizient von `first` nicht signifikant von 0 verschieden ist, also keinen Einfluß hat (äquivlant zu `anova(fm2, fm23)`). Das Resultat ist also wieder, daß `last` der einzige Regressor mit einem signifikanten Einfluß auf `amount` ist.

7 Diagnostische Plots

Zum Abschluß wollen wir noch zwei Plots anschauen, die uns helfen, den Zusammenhang zwischen den Variablen zu verstehen und die Anpassung des Modells zu beurteilen. Wir schauen uns dafür ein Streudiagramm an, das die Beobachtungen y_i gegen einen Regressor x_i abträgt und die angepaßte Regressionsgerade hinzufügt. Weiters betrachten wir einen sogenannten Residualsplot, der die Residuen $\hat{\varepsilon}_i$ gegen einen Regressor x_i abträgt.

```
R> plot(amount ~ last, data = BBBClub)
R> lines(fitted(fm2) ~ last, data = BBBClub, col = 2)
R> plot(residuals(fm2) ~ last, data = BBBClub)
```



Die erste Zeile erzeugt also ein Streudiagramm `amount ~ last`. Die zweite Zeile fügt mit `lines` eine Linie für die Regressionsgerade hinzu. Dabei werden also die Prognosen $\hat{y}_i$, die mit `fitted(fm2)` ausgerechnet werden können, gegen `last` abgetragen. Die dritte Zeile trägt dann also die Residuen $\hat{\varepsilon}_i$, die mit `residuals(fm2)` ausgerechnet werden können, gegen `last` ab.

Im linken Plot sieht man, daß die Gerade den mittleren Verlauf des Zusammenhangs recht gut widerspiegelt. Auch die Residuen zeigen keine systematischen Fehler, so daß Modell recht brauchbar ist. Einziger Nachteil ist, daß die Streuung um die angepaßte Gerade recht groß ist: die Residuen liegen etwa zwischen ± USD 150. Man kann also nur einen geringen Teil der Streuung von `amount` durch `last` erklären. Genauer gesagt ist das Bestimmtheitsmaß $R^2 = 0.2044$, es kann also nur ein Anteil von etwa 20% der Streuung von `amount` durch `last` erklärt werden. Das Bestimmtheitsmaß R^2 kann man immer in der `summary(fm2)` ablesen (s.o.).

Zum Abschluß sollen auch wieder die angelegten Objekte

```
R> objects()

[1] "BBBClub" "fm"      "fm1"     "fm2"     "fm23"    "fm3"
```

aufgeräumt werden:

```
R> remove(BBBClub, fm, fm1, fm2, fm3, fm23)
```