Analyse der „hills“-Daten aus dem Paket „MASS“

Knowledgedump.org — E. Keller

1 Laden & allgemeine Betrachtung der Daten

Wir wollen uns nun zunächst die Daten einfach ausgeben lassen, um einen Ersten Eindruck über Größe/Beschaffenheit des Datensatzes zu erhalten. Man sieht, dass die Daten vollständig sind und es zunächst keine offensichtlichen Ausreißer gibt. Das Aufrufen der Hilfsfunktion liefert uns die notwendigen Informationen zu den Attributen.

```
> library(MASS)
> hills                          #Fuer ersten Blick auf die Daten..
```

	dist	climb	time
Greenmantle	2.5	650	16.083
Carnethy	6.0	2500	48.350
Craig Dunain	6.0	900	33.650
Ben Rha	7.5	800	45.600
Ben Lomond	8.0	3070	62.267
Goatfell	8.0	2866	73.217
Bens of Jura	16.0	7500	204.617
Cairnpapple	6.0	800	36.367
Scolty	5.0	800	29.750
Traprain	6.0	650	39.750
Lairig Ghru	28.0	2100	192.667
Dollar	5.0	2000	43.050
Lomonds	9.5	2200	65.000
Cairn Table	6.0	500	44.133
Eildon Two	4.5	1500	26.933
Cairngorm	10.0	3000	72.250
Seven Hills	14.0	2200	98.417
Knock Hill	3.0	350	78.650
Black Hill	4.5	1000	17.417
Creag Beag	5.5	600	32.567
Kildcon Hill	3.0	300	15.950
Meall Ant-Suidhe	3.5	1500	27.900
Half Ben Nevis	6.0	2200	47.633
Cow Hill	2.0	900	17.933
N Berwick Law	3.0	600	18.683
Creag Dubh	4.0	2000	26.217
Burnswark	6.0	800	34.433
Largo Law	5.0	950	28.567
Criffel	6.5	1750	50.500

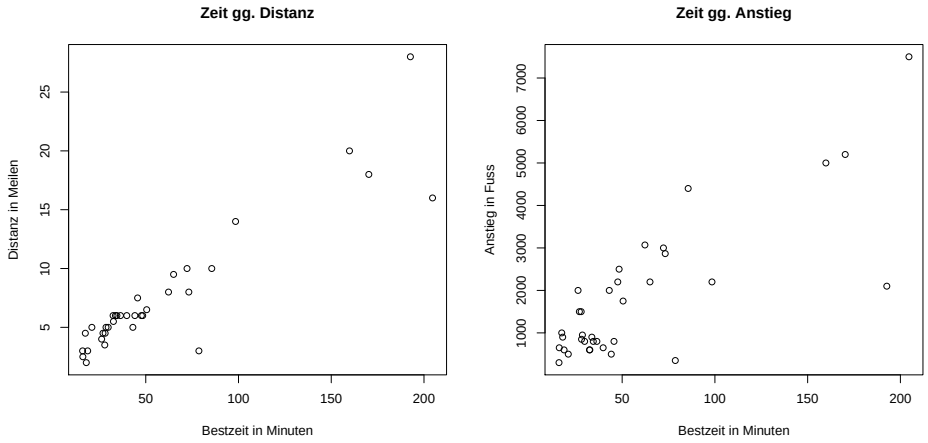
```
Acmony            5.0   500  20.950
Ben Nevis        10.0  4400  85.583
Knockfarrel       6.0   600  32.383
Two Breweries    18.0  5200 170.250
Cockleroi         4.5   850  28.100
Moffat Chase     20.0  5000 159.833

> help(hills)                #Informationen zum Datensatz
>
```

Durch das Aufrufen des Datensatzes konnten wir die Attribute der verschiedenen Rennen - die Distanz, den Anstieg und die Zeiten - erkennen. Über die Hilfsfunktion erfahren wir, dass die Distanz in Meilen gemessen wurde, der Anstieg in Fuss und die Zeit in Minuten den jeweiligen Bestzeiten des Jahres 1984 entspricht. Außerdem wissen wir nun, dass der Datensatz 35 Rennen umfasst. Wäre diese Information nicht gegeben und die Daten zu umfangreich zum Zählen, könnte man für einen groben Überblick auch die „str"-Funktion verwenden, welche die Struktur des Datensatzes analysiert und ausgibt.
Zusätzlich lassen wir uns mit dem „summary"-Befehl die Mittelwerte, Quartile, Minima und Maxima der Attributwerte ausgeben.

```
> str(hills)                 #Liefert kurzen Ueberblick

'data.frame':        35 obs. of  3 variables:
 $ dist : num  2.5 6 6 7.5 8 8 16 6 5 6 ...
 $ climb: int  650 2500 900 800 3070 2866 7500 800 800 650 ...
 $ time : num  16.1 48.4 33.6 45.6 62.3 ...

> summary(hills)        #Liefert grobe Zusammenfassung der Attributwerte

      dist            climb          time
 Min.   : 2.000   Min.   : 300   Min.   : 15.95
 1st Qu.: 4.500   1st Qu.: 725   1st Qu.: 28.00
 Median : 6.000   Median :1000   Median : 39.75
 Mean   : 7.529   Mean   :1815   Mean   : 57.88
 3rd Qu.: 8.000   3rd Qu.:2200   3rd Qu.: 68.62
 Max.   :28.000   Max.   :7500   Max.   :204.62

>
```

Nun stellt sich die Frage, ob und wie der Anstieg und die Distanz mit den Bestzeiten eines Hügel-Rennens zusammenhängen. Dass Distanz/Anstieg mit der Zeit korrelieren ist aus logischen Gruenden offensichtlich - doch wie stark ist diese Korrelation?
Für einen ersten Eindruck plotten wir die Zeiten in Abhängigkeit von Distanz oder Anstieg.

```
> oldpar<-par(mfrow=c(1,2))
> plot(hills$time,hills$dist,xlab="Bestzeit in Minuten",
  ylab="Distanz in Meilen",main="Zeit gg. Distanz")
> plot(hills$time,hills$climb,xlab="Bestzeit in Minuten",
  ylab="Anstieg in Fuss",main="Zeit gg. Anstieg")
>
```



Wir sehen, dass die Zeit im Verhältnis zu Anstieg oder Distanz zu einem linearen Modell passen könnte, da die Werte in etwa um eine Gerade liegen.
Es gibt aber auch diverse Ausreisser, welche nicht direkt zu den anderen Werten zu passen scheinen, falls die anderen tatsächlich linear angeordnet sind. Für eine weitere Untersuchung führen wir somit eine lineare Regression durch.

2 Lineare Regression

Wie im vorigen Teil bemerkt, wollen wir den Zusammenhang zwischen Anstieg/Distanz und Zeit untersuchen. Somit erhalten wir zunächst das allgemeine Modell

$$y_i = m(x_{i_1}, x_{i_2}) + \epsilon_i$$

wobei

- i dem Index des Rennens,
- y_i der Bestzeit in Minuten, (Response)
- x_{i_1} der Distanz in Meilen, (1. Regressor)
- x_{i_2} dem Anstieg in Fuß, (2. Regressor)
- ϵ_i der Abweichung vom Modell entspricht.

Wir wollen nun eine geeignete Schätzung der Modellfunktion m finden. Wir stellen zunächst fest, dass die Modellfunktion stetig sein muss und insbesondere keine Singularitäten enthalten darf. Somit sollte auch unsere geschätzte Modellfunktion stetig sein.
Ausserdem gebietet die Logik, dass die Modellfunktion (und damit auch unsere Schätzung) streng monoton steigend in beiden Variablen sein muss.
Ebenso liefert die Modellfunktion für $m(0,0) = 0$. Diese Vorraussetzung ist für den Schätzer der Modellfunktion nicht bindend, da wir in dem Datensatz nur Rennen mit einer Länge über 2 Meilen oder 300 Fuss Anstieg betrachten. Damit wäre eine Anwendung der Schätzung auf Kurzstreckenläufe vermutlich ohnehin relativ ungenau.
Wollte man jedoch auch Bestzeiten von Kurzstreckenläufen testen, müsste man $m(0,0) = 0$ zusätzlich verlangen, um halbwegs plausible Zeiten zu erhalten.

Eine einfache und (wie zuvor erwähnt) potentiell effektive Schätzung stellt die lineare Regression dar, welche insbesondere monotone und stetige Schätzer für die Modellfunktion liefert.
In diesem Fall ist $\hat{m}$ durch $X \cdot \beta = \beta_1 * x_{i_1} + \beta_2 * x_{i_2} + c$ gegeben, wobei β ein 3-dimensionaler Vektor und c eine reelle Zahl ist. (für den Spezialfall $c = 0$ wäre insb. $\hat{m}(0,0) = 0$)
Das Modell hat dann die Form

$$y_i = \beta_1 * x_{i_1} + \beta_2 * x_{i_2} + c + \epsilon_i$$

und wir wollen β_1, β_2, sowie c schätzen.
Wir nehmen nicht $c = 0$ an, d.h. wir erhalten insbesondere ein Ergebnis, welches höchstwahrscheinlich nicht durch den Nullpunkt geht - dies kann potentiell unsere Approximation für Mittel- und Langstreckenläufe verbessern, macht den Schätzer aber ungeeignet für Kurzstreckenläufe.

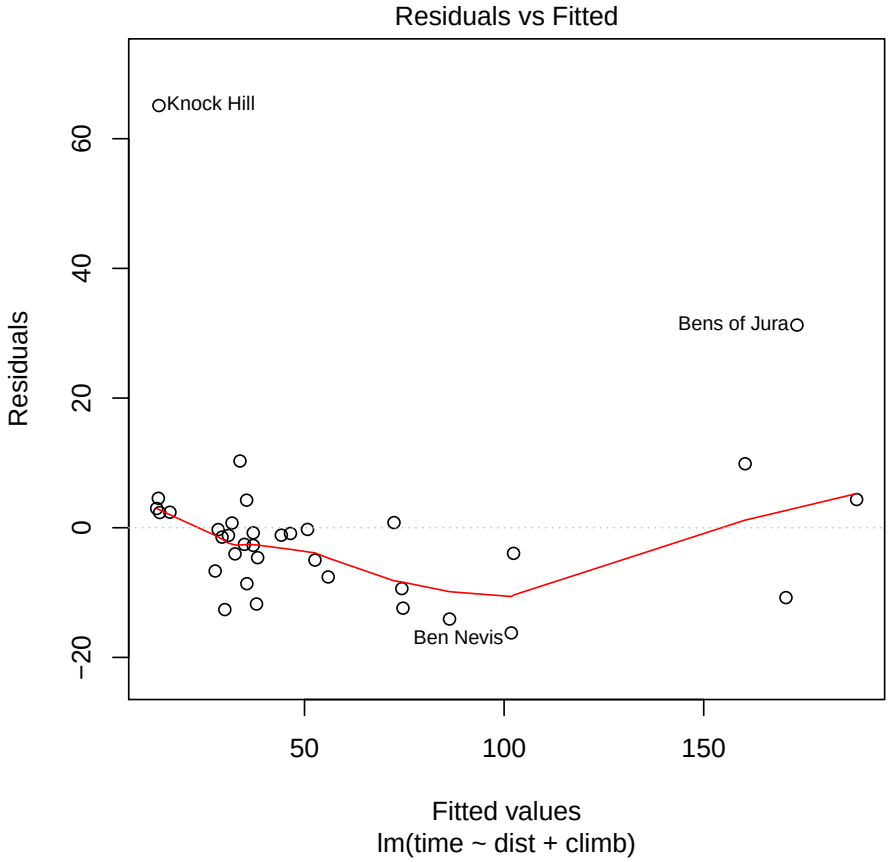
Die Wilkinson Rogers Notation lautet: time $\sim$ dist + climb.
Mit folgendem Code kann man nun die lineare Regression durchführen und sich die Koeffizienten, sowie einige weitere Daten ausgeben:

```
> lmhills <- lm( time ~ dist + climb, data = hills)
> summary(lmhills)

Call:
lm(formula = time ~ dist + climb, data = hills)

Residuals:
    Min      1Q  Median      3Q     Max
-16.215  -7.129  -1.186   2.371  65.121

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -8.992039   4.302734  -2.090   0.0447 *
```

```
dist         6.217956   0.601148  10.343 9.86e-12 ***
climb        0.011048   0.002051   5.387 6.45e-06 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 14.68 on 32 degrees of freedom
Multiple R-squared:  0.9191,        Adjusted R-squared:  0.914
F-statistic: 181.7 on 2 and 32 DF,  p-value: < 2.2e-16
```

Wir stellen uns die Frage nach der Güte dieser Approximation. Hinweise darauf können die Residuen liefern. Diese sind schon in unserem Objekt der Klasse „lm“ enthalten und über lmhills$residuals einsehbar.
Die gefitteten Werte erhalten wir mit lmhills$fitted.values (oder kurz: lmhills$fitted).
Da wir zwei Regressoren haben, liefert der „Residuen gg. Fit“-plot einen geeigneten Ersatz für den „Residuen gg. Regressoren“-plot, der nicht in einem 2-dimensionalen plot darstellbar wäre. Man kann den plot entweder über die lmhills-Werte aufrufen und diese gegeneinander plotten, oder den plot direkt mit lmhills (Klasse „lm“) aufrufen - letztere Variante liefert zusätzlich eine „Trend-Kurve“ und benennt Werte, die betragsmäßig große Residuen aufweisen.

```
> par(mfrow=c(1,1))
> plot(lmhills, which =1)
```



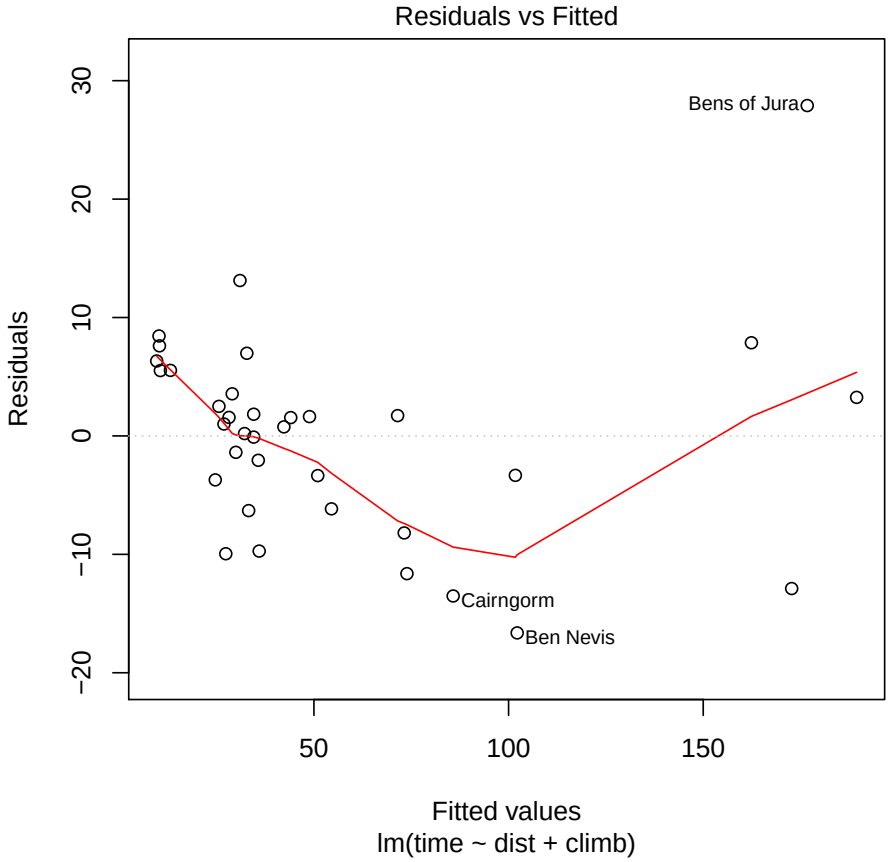
Man erkennt damit sofort ein von den anderen Werten stark abweichendes Rennen: Das „Knock Hill“ Rennen. Wir wollen zunächst den Index des Rennens bestimmen, um die genauen Werte auszulesen - dieser lässt sich leicht mit dem rownames Befehl finden. (Alternativ: Aufrufen von „identify(hills)“ nach plot)
Ein Aufruf der Daten zu dem Rennen macht schließlich deutlich, dass es sich um einen Fehler im Datensatz handeln muss:

```
> rownames(hills)

 [1] "Greenmantle"      "Carnethy"         "Craig Dunain"
 [4] "Ben Rha"          "Ben Lomond"       "Goatfell"
 [7] "Bens of Jura"     "Cairnpapple"      "Scolty"
[10] "Traprain"         "Lairig Ghru"      "Dollar"
[13] "Lomonds"          "Cairn Table"      "Eildon Two"
[16] "Cairngorm"        "Seven Hills"      "Knock Hill"
[19] "Black Hill"       "Creag Beag"       "Kildcon Hill"
[22] "Meall Ant-Suidhe" "Half Ben Nevis"   "Cow Hill"
[25] "N Berwick Law"    "Creag Dubh"       "Burnswark"
[28] "Largo Law"        "Criffel"          "Acmony"
```

```
[31] "Ben Nevis"        "Knockfarrel"      "Two Breweries"
[34] "Cockleroi"        "Moffat Chase"

> hills[18,]

           dist climb  time
Knock Hill    3   350 78.65

>
```

Das Rennen geht über eine vergleichsweise geringe Distanz von 3 Meilen und ist mit einem Anstieg von 350 Fuß auch nicht sehr hügelig. Damit ist die Bestzeit von ca. 79 Minuten offensichtlich ein falsch eingetragener Wert. Der richtige Wert könnte ca. 19 Minuten gewesen sein, wurde aber vermutlich falsch abgelesen. Deswegen verbessern wir den Wert der Bestzeit und speichern dies in einem neuen Datensatz ab.

```
> hills2<-hills
> hills2[18,3] <- 18.65
> lmhills2 <- lm( time ~ dist + climb, data = hills2)
> lmhills2

Call:
lm(formula = time ~ dist + climb, data = hills2)

Coefficients:
(Intercept)         dist        climb
  -12.94198      6.34556      0.01175

> plot(lmhills2, which =1)
>
```



Ein weiteres Rennen, das stark hervorsticht, ist "Bens of Jura".

```
> hills[7,]

             dist climb    time
Bens of Jura   16  7500 204.617

>
```

Mit einem Anstieg von 7.500 Fuß ist es jedoch auch das hügeligste Rennen auf einer relativ großen Distanz von 16 Meilen, weswegen die Bestzeit plausibel bleibt. Zusätzlich kann man auf der Internetseite des Rennens (http://www.jurafellrace.org) nachlesen, dass die Zeit stimmt.

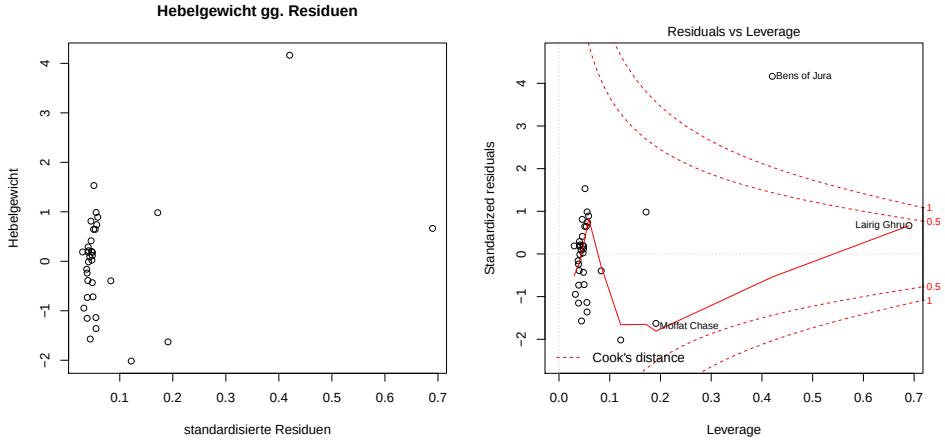
Auch in der korrigierten Version sieht man, dass die Residuen der kurzen und langen Rennen tendenziell positiv sind, d.h. $\hat{y}_i < y_i$ und die Rennen dazwischen eher negative Residuen aufweisen, mit $\hat{y}_i > y_i$. Dies könnte nahelegen, dass unsere Modellschätzung nicht erwartungstreu ist. Es kann aber auch daran liegen, dass die Datendichte im Mittelstrecken-Bereich viel größer ist als im Langstreckenbereich, was unsere Schätzung für diese Rennen potentiell ungenau macht.

Wir untersuchen nun zusätzlich den Einfluss der verschiedenen Werte auf das lineare Modell, um zu sehen welche dieses stark prägen und ggf. korrigierend nachzubessern für eine genauere Schätzung. Dazu betrachten wir die Diagonalwerte der Hutmatrix, welche ein Maß für das Hebelgewicht der Werte darstellen - mit dem Befehl hatvalues lassen wir uns diese ausgeben.

```
> leverage<-hatvalues(lmhills2)
> leverage

     Greenmantle         Carnethy     Craig Dunain
       0.05375572       0.04946414       0.03840444
          Ben Rha       Ben Lomond          Goatfell
       0.04848872       0.05527121       0.04680469
     Bens of Jura      Cairnpapple           Scolty
       0.42043463       0.04103328       0.04025783
         Traprain      Lairig Ghru           Dollar
       0.04570891       0.68981613       0.04345357
          Lomonds      Cairn Table       Eildon Two
       0.03231875       0.05126338       0.03877135
         Cairngorm      Seven Hills       Knock Hill
       0.04436257       0.08313942       0.05535523
        Black Hill       Creag Beag     Kildcon Hill
       0.03850209       0.04590867       0.05657466
Meall Ant-Suidhe   Half Ben Nevis          Cow Hill
       0.04825780       0.03977381       0.05842537
    N Berwick Law       Creag Dubh         Burnswark
       0.05072281       0.05499644       0.04103328
         Largo Law          Criffel           Acmony
       0.03758135       0.02992818       0.04824732
         Ben Nevis      Knockfarrel    Two Breweries
       0.12158212       0.04746275       0.17158482
         Cockleroi     Moffat Chase
       0.04032547       0.19098908

>
```

Um die verschiedenen Werte ein wenig anschaulicher darzustellen, plotten wir die Werte im Vergleich zu den Residuen. Um die Darstellung anschaulicher zu machen reskalieren wir die Werte zunächst, indem wir sie standardisieren. Dies geschieht über das teilen durch ihre geschätzte Standardabweichung (auslesbar mit summary(lmhills2)$sigma) und durch $\sqrt{1 - \text{hatvalues(lmhills2)}}$. Der Schritt ist bereits in der Funktion „stdres“ implementiert - genauso wie der „Residuen gg. Hebelgewicht“-plot, der noch einige zusätzliche Informationen, wie z.B. die Cooks-Distanz und eine Beschriftung der drei Werte mit dem größten Hebelgewicht enthält.

```
> stdresiduals<- stdres(lmhills2)
> par(mfrow=c(1,2))
> plot(leverage,stdresiduals,xlab="standardisierte Residuen",
  ylab="Hebelgewicht",main="Hebelgewicht gg. Residuen")
> plot(lmhills2, which=5)
>
```

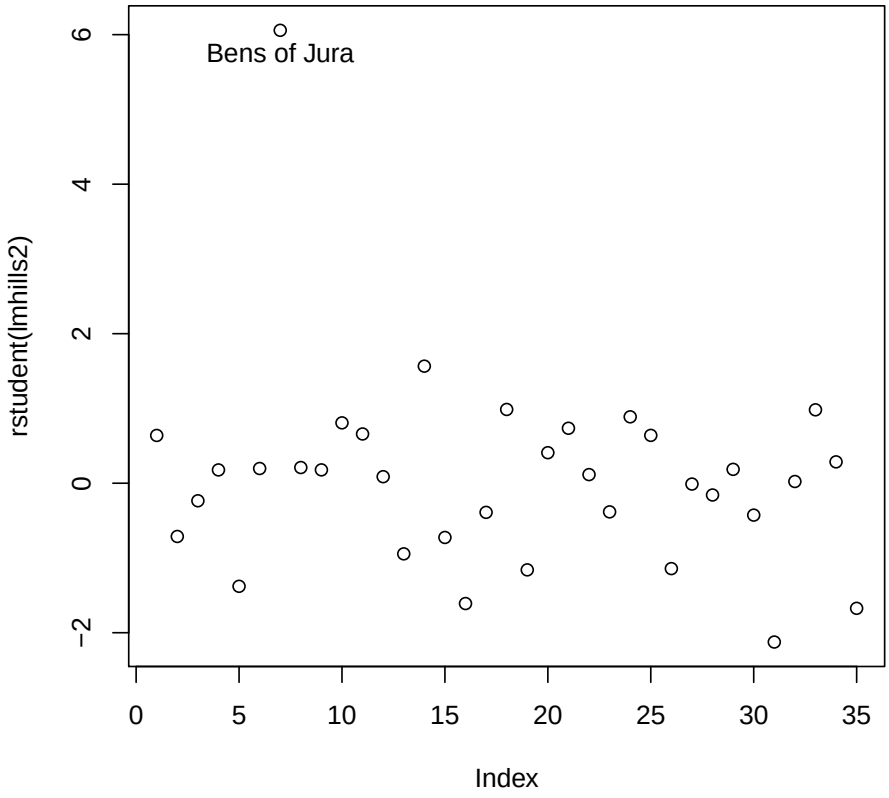


Man sieht, dass das schon zuvor betrachtete Rennen „bens of jura“ ein großes Hebelgewicht und Residuum aufweist - also insbesondere stark von den anderen Werten abweicht und einen großen Einfluss auf unsere Schätzung hat. Das 11. Rennen „Lairig Ghru“ weist ein noch größeres Hebelgewicht auf. Da das zugehörige Residuum jedoch nicht so groß ist, ist der potentielle Einfluss auf unsere Schätzung nicht so „schlimm“. Problematisch ist jedoch, wenn das Residuum nur so klein ist, weil der Einfluss des Wertes so groß ist. Selbiges gilt andersherum für die anderen Werte, welche ein geringes Hebelgewicht aber relativ großes Residuum besitzen. Um den Einfluss einzelner Werte auf die Residuen, bzw. deren Varianz zu untersuchen, gibt es die Funktion „rstudent“ (standardisiert Werte und lässt bei der Berechnung der geschätzten Standardabweichung den entsprechenden Wert weg - „Studentisierung“), welche wir im Folgenden gegen den Index plotten.

```
> par(mfrow=c(1,1))
> rstud<-rstudent(lmhills2)
> rstud
```

```
     Greenmantle          Carnethy      Craig Dunain
       0.63926079       -0.71209727       -0.23478365
          Ben Rha         Ben Lomond           Goatfell
       0.17782834       -1.37857425        0.19696789
     Bens of Jura        Cairnpapple            Scolty
       6.05833087        0.20978378        0.17865144
        Traprain       Lairig Ghru            Dollar
       0.80765428        0.65882442        0.08734242
.....
```

```
> plot(rstudent(lmhills2))
> text(7,rstud[7],rownames(hills)[7],pos=1)
>
```

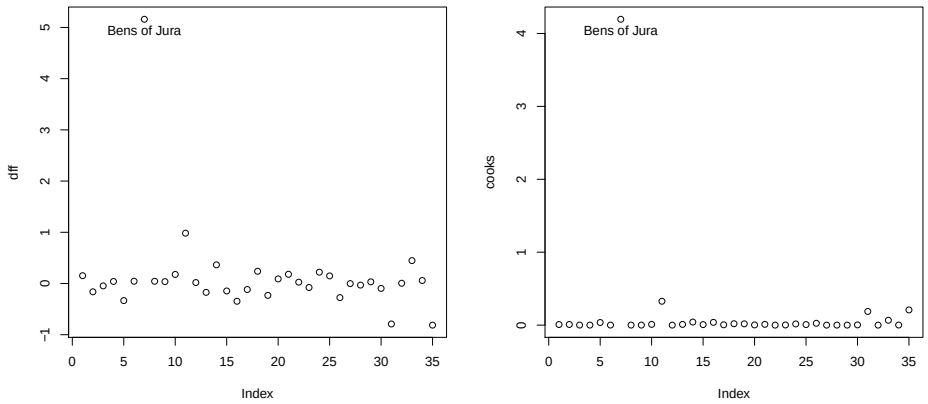


Außerdem besteht die Möglichkeit den Einfluss auf die gefitteten Werte und das Modell zu testen, indem man die „dffits“ und „cooks-Distanz“ betrachtet.

```
> par(mfrow=c(1,2))
> dff<-dffits(lmhills2)
> dff
```

```
     Greenmantle          Carnethy      Craig Dunain
      0.152366251     -0.162442730     -0.046920469
          Ben Rha        Ben Lomond          Goatfell
      0.040143421     -0.333446435      0.043646475
     Bens of Jura       Cairnpapple           Scolty
      5.160015566      0.043394858      0.036589328
        Traprain       Lairig Ghru           Dollar
      0.176760616      0.982486870      0.018615937
.....
```

```
> plot(dff)
> text(7,dff[7],rownames(hills)[7],pos=1)
> cooks<-cooks.distance(lmhills2)
> cooks

     Greenmantle         Carnethy     Craig Dunain
    7.884188e-03     8.933489e-03     7.561713e-04
         Ben Rha       Ben Lomond         Goatfell
    5.539276e-04     3.604780e-02     6.546696e-04
    Bens of Jura      Cairnpapple           Scolty
    4.194889e+00     6.470345e-04     4.601813e-04
        Traprain      Lairig Ghru           Dollar
    1.052918e-02     3.275532e-01     1.192147e-04
.....

> plot(cooks)
> text(7,cooks[7],rownames(hills)[7],pos=1)
>
```

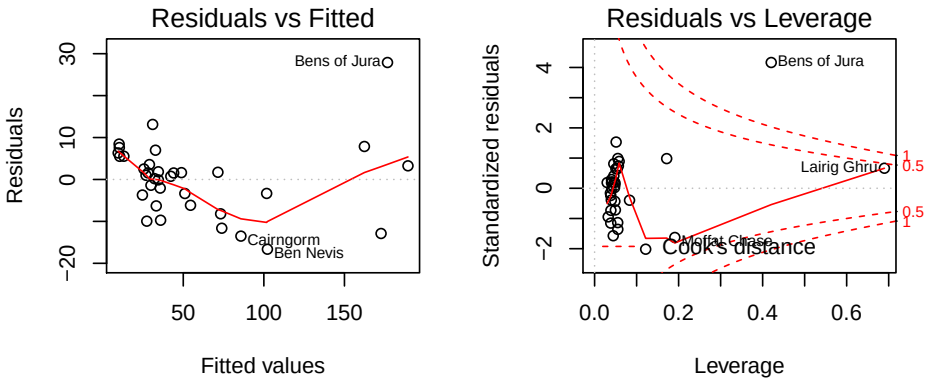


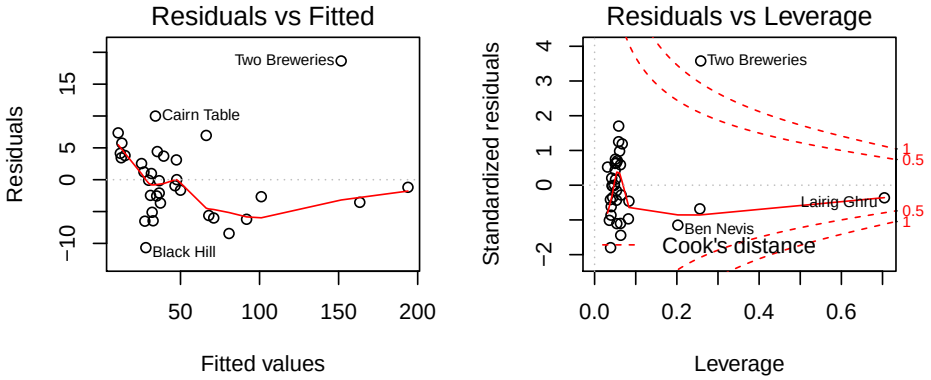
Da der Einfluss von „bens of jura“ auf das Modell wieder hervorsticht, entfernen wir diesen aus unserer Schätzung und führen einige der vorigen Schritte erneut durch. Man sieht, dass die geschätzte Standardabweichung der Residuen sinkt und wir erhalten ein Modell, welches wohl insbesondere für Mittelstreckenläufe besser geeignet ist.

```
> par(mfrow=c(2,2))
> hills3<-hills2[-7,]
> lmhills3<-lm(time ~ dist + climb, data = hills3)
> summary(lmhills3)

Call:
lm(formula = time ~ dist + climb, data = hills3)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-10.6641  -3.6353  -0.5493   3.6351  18.6463

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -9.921740   1.842499  -5.385 7.12e-06 ***
dist         6.683247   0.254011  26.311  < 2e-16 ***
climb        0.007928   0.001055   7.516 1.81e-08 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 6.05 on 31 degrees of freedom
Multiple R-squared:  0.9821,        Adjusted R-squared:  0.981
F-statistic: 851.4 on 2 and 31 DF,  p-value: < 2.2e-16
```

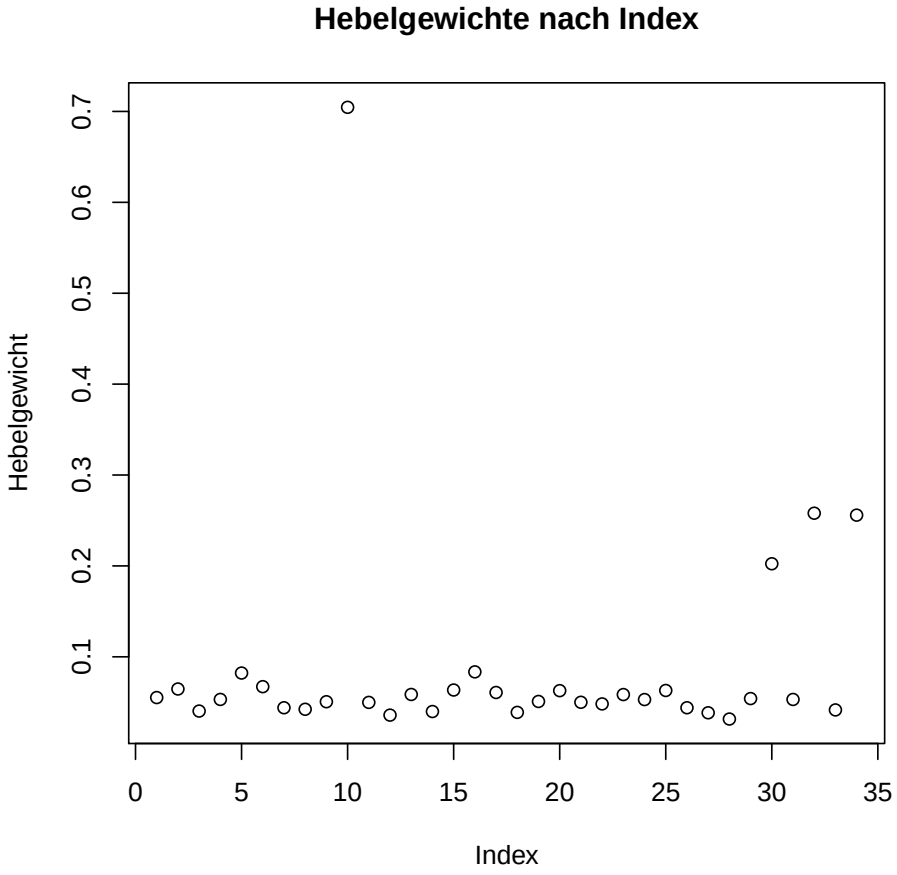
```
> plot(lmhills2, which=c(1,5))
> plot(lmhills3, which=c(1,5))
>
```



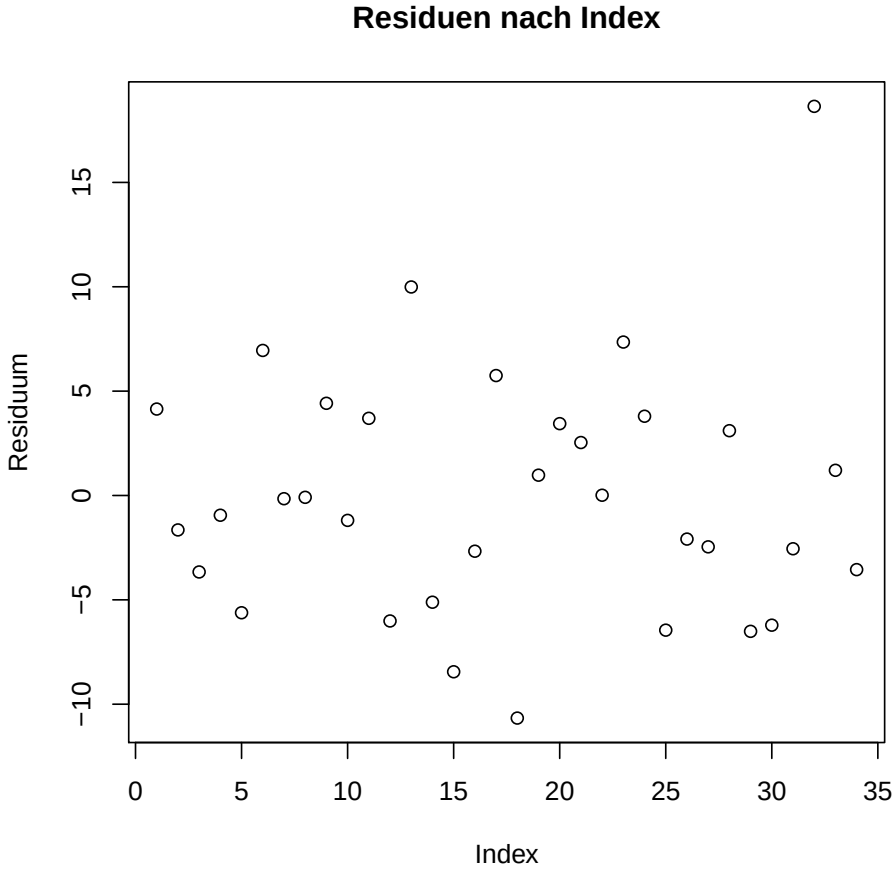


Man sieht, dass nun im neuen Modell das Rennen „Two Breweries“ mit Index 32 hervorsticht. Betrachtung der Hutwerte liefert auch ein relativ großes Residuum bei großem „Hutwert“.

```
> par(mfrow=c(1,1))
> plot(hatvalues(lmhills3), main="Hebelgewichte nach Index",
  ylab="Hebelgewicht", xlab="Index")
> identify(hatvalues(lmhills3))
```




```
> plot(residuals(lmhills3), main="Residuen nach Index",
  ylab="Residuum", xlab="Index")
> identify(residuals(lmhills3))
> par(oldpar)
```



Streicht man auch dieses Rennen aus den Daten und sieht es als Ausreißer an, lässt sich die Schätzung möglicherweise weiter verbessern.

3 Weiterführende Untersuchung des Datensatzes

Eine weitere Analyse des Datensatzes könnte unter anderem das Verwenden anderer Modelle einbeziehen - insbesondere, da wir bis jetzt nur die relativ simple lineare Regression verwendet haben. Hier stellt sich die Frage nach Kriterien für die Modellwahl und damit auch nach Möglichkeiten, die Güte dieser Modelle zu schätzen und verschiedene Modelle untereinander zu vergleichen.
Zusätzlich könnte man die verschiedenen Werte gewichten, um das Modell weiter anzupassen. Hierbei könnte man beispielsweise entscheiden, ob man mit dem Modell beliebig lange Rennen oder nur Rennen mit einer bestimmten Länge/Anstieg schätzen will. Im Allgemeinen stellt sich die Frage, mit welchem Ziel man die Daten untersucht, da danach die Untersuchung und Anpassung des Modells und der Schätzung ausgerichtet wird.
Außerdem ist es fraglich, ob das Herausnehmen bestimmter Werte die beste Art ist mit Ausreißern umzugehen, da dies insbesondere bei kleiner Anzahl von Stichproben nicht praktikabel ist.