Vergleich verschiedener linearer Regressionsmodelle für die „hills“-Daten

Knowledgedump.org — E.Keller

Modellkandidaten:
time ~ 1
time $\sim 1 + \text{dist}$
time $\sim 1 + \text{dist} + \text{climb}$

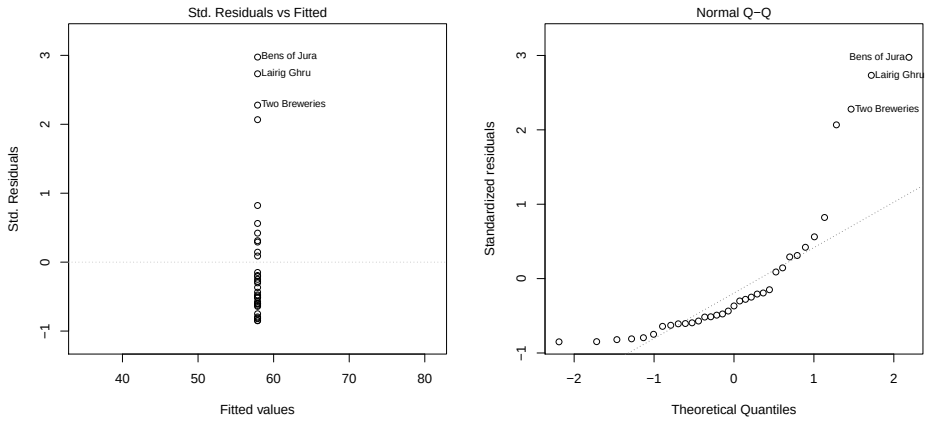
1 Visualisierung der Modelleigenschaften

Zunächst laden wir die benötigten Daten und insbesondere die abgeändete Variante der plot.lm-Funktion. Im Anschluss definieren wir die von den zuvor genannten Formeln bestimmten linearen Modelle und lassen uns allgemeine Informationen ausgeben.

```
> library(MASS)
> #source("C:\\.............\\plot.lm2.R")
> ##plot.lm2 laden!
>
> lm1<-lm(time~1, data=hills, y=TRUE)
> lm1d<-lm(time~1+dist, data=hills, y=TRUE)
> lm1dc<-lm(time~1+dist+climb, data=hills,y=TRUE)
> summary(lm1)

Call:
lm(formula = time ~ 1, data = hills, y = TRUE)

Residuals:
   Min     1Q Median     3Q    Max
-41.93 -29.88 -18.13  10.75 146.74

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   57.876      8.458   6.842 7.09e-08 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 50.04 on 34 degrees of freedom

> summary(lm1d)

Call:
lm(formula = time ~ 1 + dist, data = hills, y = TRUE)
```

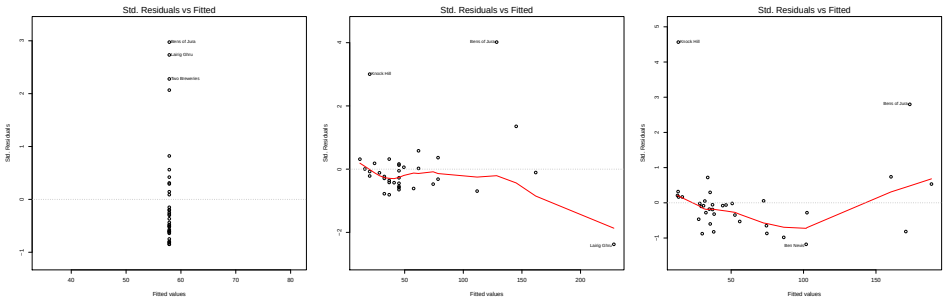
```
Residuals:
    Min      1Q  Median      3Q     Max
-35.745  -9.037  -4.201   2.849  76.170

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -4.8407     5.7562  -0.841    0.406
dist          8.3305     0.6196  13.446 6.08e-15 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 19.96 on 33 degrees of freedom
Multiple R-squared:  0.8456,        Adjusted R-squared:  0.841
F-statistic: 180.8 on 1 and 33 DF,  p-value: 6.084e-15
```

```
> summary(lm1dc)

Call:
lm(formula = time ~ 1 + dist + climb, data = hills, y = TRUE)

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

> plot.lm2(lm1, which=1:2)
```

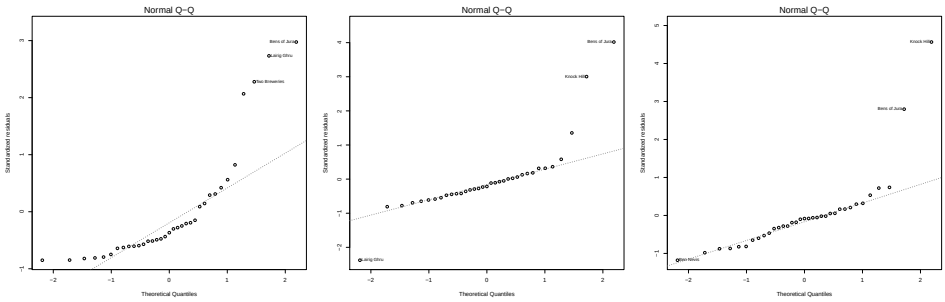


Man sieht schon bei dieser recht oberflächlichen Betrachtung, dass das Modell lm1 unbrauchbar als Schätzer für die hills-Daten ist. Dies folgt bereits aus der Logik, dass die Bestzeit nicht unabhängig von Anstieg und Distanz sein kann. Mit dem summary-Befehl sieht man zusätzlich, dass die residuelle Standardabweichung übermäßig groß ist und der QQ-Plot nicht annähernd zu einer Normalverteilung passt. lm1 kann damit prinzipiell schon ausgeschlossen werden - zur Verdeutlichung der schrittweisen Zunahme an Modellinformationen behalten wir es jeoch in unserer weiteren Untersuchung. Wir lassen uns zunächst diverse plots Ausgeben, um ein Gefühl für die verschiedenen Modelle zu bekommen.

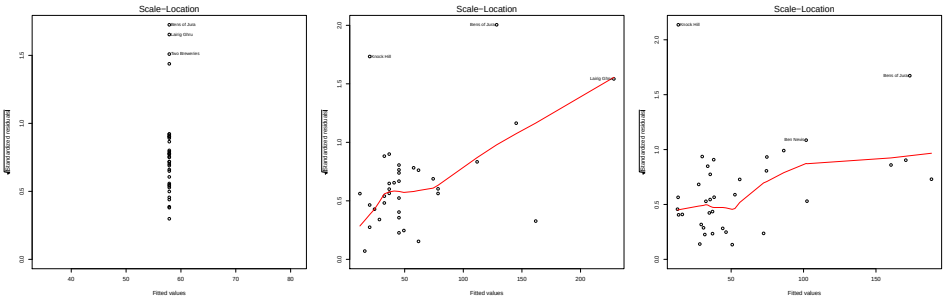
```
> par(mfrow=c(1,3))
> plot.lm2(lm1,which=1)
> plot.lm2(lm1d,which=1)
> plot.lm2(lm1dc,which=1)
```



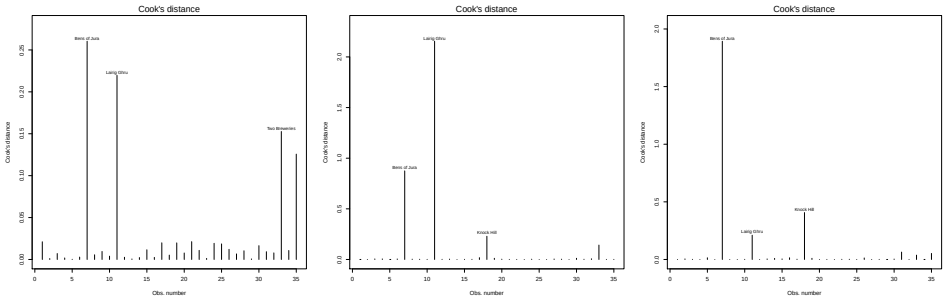

```
> plot.lm2(lm1,which=2)
> plot.lm2(lm1d,which=2)
> plot.lm2(lm1dc,which=2)
```



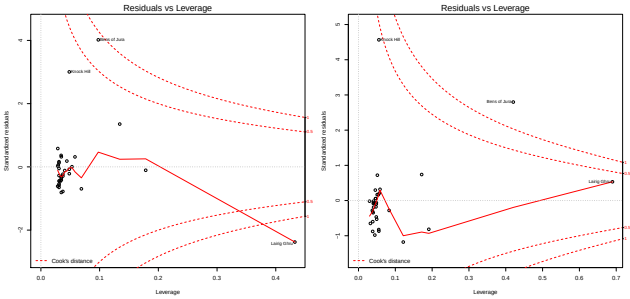

```
> plot.lm2(lm1,which=3)
> plot.lm2(lm1d,which=3)
> plot.lm2(lm1dc,which=3)
```



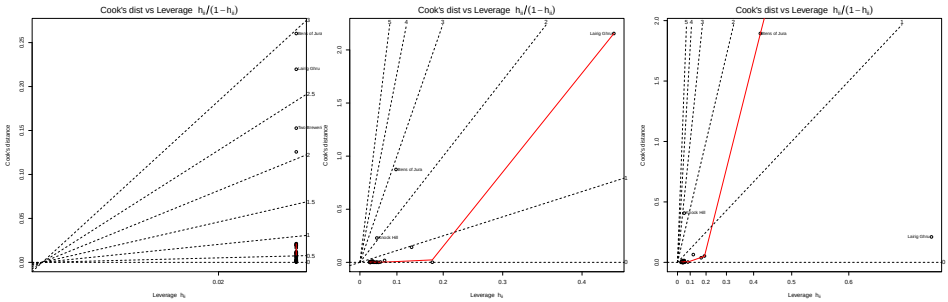

```
> plot.lm2(lm1,which=4)
> plot.lm2(lm1d,which=4)
> plot.lm2(lm1dc,which=4)
```



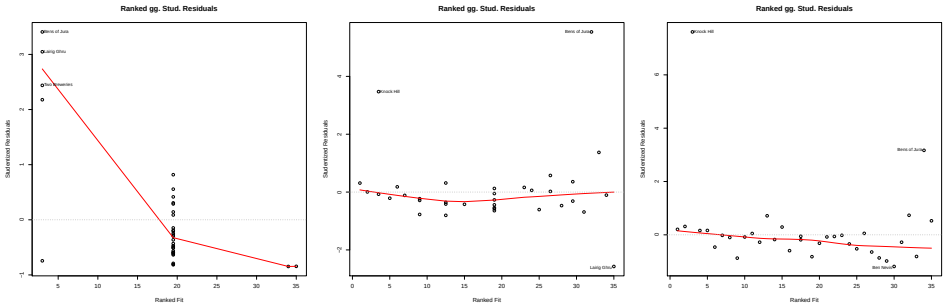

```
> plot.lm2(lm1,which=5)          ##Alle Hutwerte identisch -> kein plot
> plot.lm2(lm1d,which=5)
> plot.lm2(lm1dc,which=5)
```



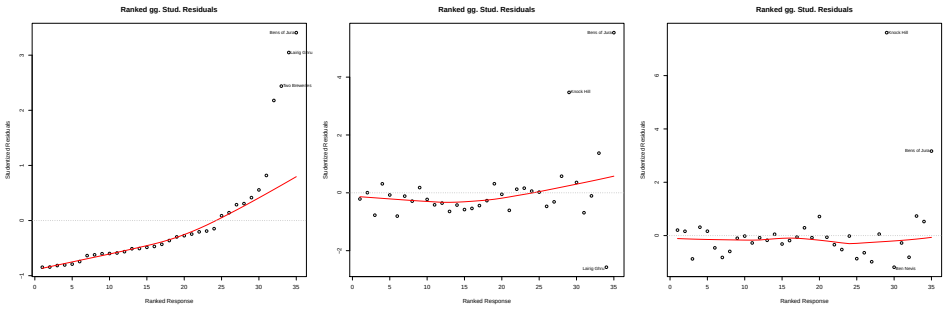

```
> plot.lm2(lm1,which=6)
> plot.lm2(lm1d,which=6)
> plot.lm2(lm1dc,which=6)
```



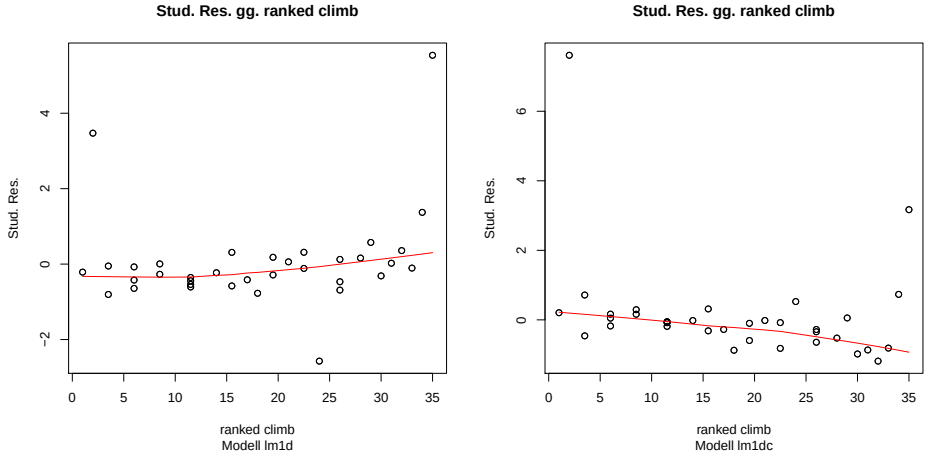

```
> plot.lm2(lm1,which=7)          ##Ranked Fit - Vorsicht Rundungsfehler!
> plot.lm2(lm1d,which=7)
> plot.lm2(lm1dc,which=7)
```




```
> plot.lm2(lm1,which=7, rank.type="response")
> plot.lm2(lm1d,which=7, rank.type="response")
> plot.lm2(lm1dc,which=7, rank.type="response")
```




```
> par(mfrow=c(1,2))
> plot(rank(hills[,2]),studres(lm1d), xlab="ranked climb", ylab="Stud. Res.",
  main="Stud. Res. gg. ranked climb", sub="Modell lm1d")
> panel.smooth(rank(hills[,2]),studres(lm1d))
> plot(rank(hills[,2]),studres(lm1dc), xlab="ranked climb", ylab="Stud. Res.",
  main="Stud. Res. gg. ranked climb", sub="Modell lm1dc")
> panel.smooth(rank(hills[,2]),studres(lm1dc))
```



Durch die Hinzunahme der climb-Variable wird die Schätzung besser für Rennen mit einem signifikanten Anstieg - ob die Qualität der Schätzung beliebiger Hügelrennen dadurch verbessert wird ist jedoch noch offen (overfitting?). Außerdem ist der Informationszuwachs vom Modell lm1d zu lm1dc offensichtlich nicht so groß wie der von lm1 zu lm1d oder lm1dc.
In den „Ranked gg. Stud Residuals“-plots sieht man, wie sich die hinzugefügten Variablen auf die Modelle auswirken: Während lm1 nur eine Art Mittelwert der Rennen zu sein scheint, sind die anderen Modelle viel genauer und approximieren die Bestzeit für unterschiedliche Längen-/Anstiegswerte „ähnlich gut“.
Die beiden letzten plots verdeutlichen dabei, wie sich die Hinzunahme der Anstiegsvariable auf die Modellschätzung auswirkt: Sortiert man die Rennen dem Anstieg nach weist das Modell lm1d wachsende und das Modell lm1dc fallende studentisierte Residuen auf, da in ersterem eine hohe Bestzeit nur von der Distanz abhing und in letzterem Modell der Anstieg in die Schätzung einfließt.

Um entscheiden zu können, ob wir lm1d oder lm1dc favorisieren sollen, führen wir nun eine Varianzanalyse, sowie eine Schrittweise Orthogonalzerlegung durch, um die Modelle zu untersuchen.

2 Modellwahl mit Hilfe von Varianzanalyse (ANOVA)

Um eine Varianzanalyse durchzuführen müssen wir zunächst prüfen, ob die Vorraussetzungen gegeben sind:

- unabhängige Beobachtungen,
- normalverteilte Fehler (siehe QQ-Plot) mit Mittelwert 0 und
- gleicher Varianz (Homoskedastizität).

Dies scheint für lm1d und lm1dc gegeben zu sein oder zumindest nicht grob verletzt. Ein Problem könnte ggf. die Abhängigkeit des Anstiegs von der Distanz darstellen.
Mit Hilfe der Funktion „anova()“ erhalten wir als Ausgabe ein Objekt der Klasse ANOVA:

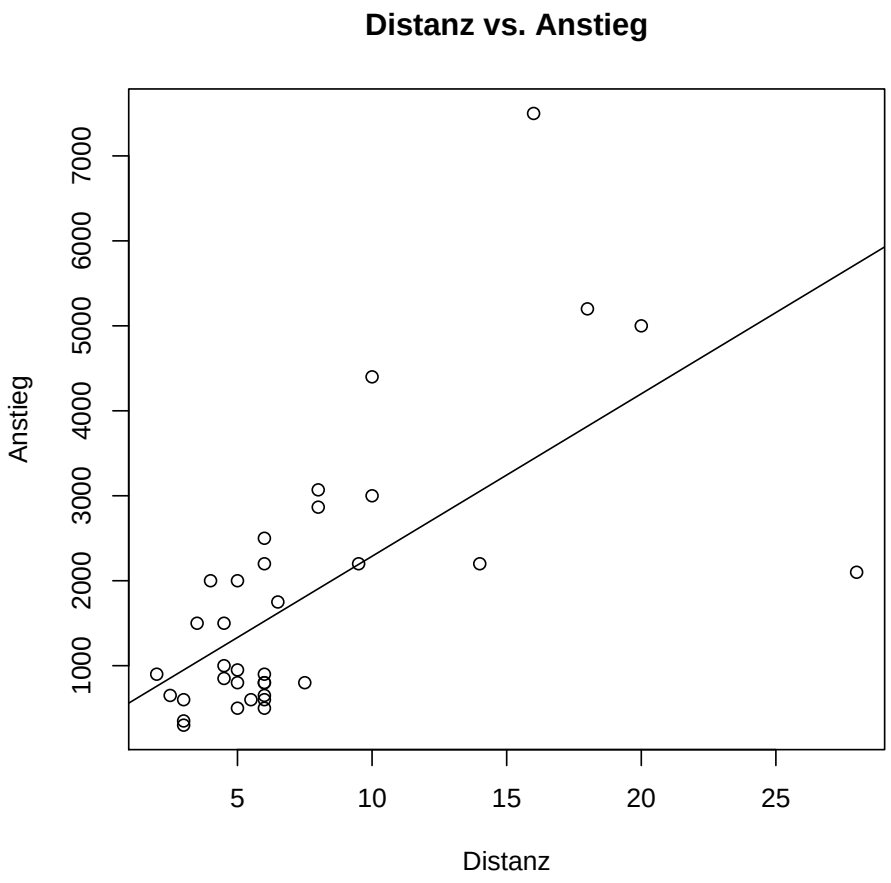
```
> anova(lm1dc)

Analysis of Variance Table

Response: time
          Df Sum Sq Mean Sq F value    Pr(>F)
dist       1  71997   71997 334.293 < 2.2e-16 ***
climb      1   6250    6250  29.018 6.445e-06 ***
Residuals 32   6892     215
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Wir erfahren, wie groß die Sum of Sq.- und F-Werte zu den entsprechenden Variablen, sowie zum eingegebenen Modell (lm1dc) sind. Die Sum of Sq. bezeichnen dabei die Differenz der RSS zwischen den verschiedenen Modellen. Anschaulich ergibt das, um wie viel die Summe der quadrierten Residuen ansteigen würde, wenn diese Variable weggelassen wird. Dies lässt sich beispielsweise leicht mit „sum(lm1$residuals^2)-sum(lm1d$residuals^2)“ berechnen. „Mean Sq“ ergibt sich, indem man den vorigen Wert durch die Freiheitsgrade teilt. Man sieht, dass lm1dc die RSS minimiert.
Für uns von großem Interesse ist der Wert des F-Tests, welcher bestimmt ob die Hinzunahme der Variablen eine signifikante Verbesserung des Modells ergibt oder ob diese Variablen tatsächlich keinen Einfluss auf die Bestzeit haben (in diesem Fall müsste der entsprechende geschätzte Koeffizent 0 sein). In unserem Fall ergibt sich für beide Variablen ein sehr großes Signifikanzniveau, was nahelegt das Modell lm1dc zu wählen, also insbesondere alle Variablen in die Schätzung einzubeziehen.

3 Modellwahl mit Schrittweiser Orthogonalzerlegung

Eine Alternative Herangehensweise liefert die schrittweise Ortogonalzerlegung der Modelle. Wir haben bereits festgestellt, dass lm1 den anderen Modell unterlegen ist. Um zu entscheiden, ob lm1d oder lm1dc besser, ist nutzen wir nun die Orthogonalzerlegung.

```
> par(mfrow=c(1,1))
> lmc.d<-lm(climb~dist, data=hills)
> plot(hills$dist, hills$climb, xlab="Distanz",ylab="Anstieg",
  main="Distanz vs. Anstieg")
> abline(lmc.d)
```



Wir bemerken zunächst, dass höchstwahrscheinlich eine Korrelation zwischen Distanz und Anstieg besteht - was auch logisch erscheint - und definieren uns dann über die entsprechenden Residuen das Orthokomplement zu lm1d.

```
> lm1d.orth<-lm(lm1d$residuals ~ 0 + lmc.d$residuals, data=hills)
> summary(lm1d.orth)

Call:
lm(formula = lm1d$residuals ~ 0 + lmc.d$residuals, 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|)
lmc.d$residuals  0.01105    0.00199   5.553 3.29e-06 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 14.24 on 34 degrees of freedom
Multiple R-squared:  0.4756,        Adjusted R-squared:  0.4601
F-statistic: 30.83 on 1 and 34 DF,  p-value: 3.286e-06
```

Wir erhalten einen sehr kleinen p-Wert zum t-Test, was bedeutet, dass die Residuen des lmc.d-Modells einen signifikanten Einfluss auf die Residuen des lm1d-Modells haben. Dies legt nahe, dass unser Modell lm1d durch die Hinzunahme der Anstiegs-Variable verbessert wird.

Aus beiden Herangehensweisen lässt sich folgern, dass lm1dc das bessere Modell darstellt. Dies hängt jedoch natürlich auch von den zu schätzenden Werten ab - so könnte man mit lm1d bei sehr flachen Rennen eine bessere Schätzung erhalten.