Züchtungslehre - Lösung 3

Peter von Rohr

October 20, 2015

Aufgabe 1 (8)

Der in dieser Aufgabe verwendete Datensatz unter
`http://charlotte-ngs.github.io/LivestockBreedingAndGenomics/w5/simgenphen.csv`
umfasst nur 30 Records und wurde mit folgenden Parametern simuliert.

Locus	additiver genotypischer Wert (a)
1	10
2	5
3	2

Als Intercept wurde ein Wert von -10.76 angenommen. Die Standardabweichung der Fehler betrug 2.5.

Analyse des ursprünglichen Datensatzes

In der Aufgabe ging es darum die multiple lineare Regression mit den simplen Regressionen für die drei Loci miteinander zu vergleichen. Bei der multiplen Regression sollten die in der Simulation verwendeten Parameter wieder als Effekte erscheinen.

```
> lmMultRegOrg <- lm(y ~ locus1 + locus2 + locus3, data = dfSimGenPhenDat)
> summary(lmMultRegOrg)

Call:
lm(formula = y ~ locus1 + locus2 + locus3, data = dfSimGenPhenDat)

Residuals:
    Min      1Q  Median      3Q     Max
-4.3756 -1.5085 -0.4046  1.5652  4.2774

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -10.710      1.010 -10.605 6.16e-11 ***
locus1         8.670      1.085   7.991 1.81e-08 ***
locus2         8.048      2.745   2.933  0.00693 **
```

```
locus3         6.489      2.258   2.874  0.00798 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 2.206 on 26 degrees of freedom
Multiple R-squared:  0.8133,        Adjusted R-squared:  0.7918
F-statistic: 37.75 on 3 and 26 DF,  p-value: 1.274e-09
```

Analyse eines neuen Datensatzes

Damit der in der Aufgabe beschriebene Effekt trotzdem sichtbar gemacht werden kann, wird ein neuer Datensatz simuliert. Die Parameterwerte wurden etwas anders gewählt, so dass die Effekte eindeutig schätzbar sind.

Locus	additiver genotypischer Wert (a)
1	29
2	11
3	5

Als Intercept wurde ein Wert von -10.76 angenommen. Die Standardabweichung der Fehler betrug 1.

Der korrigierte Datensatz ist verfügbar unter
`http://charlotte-ngs.github.io/LivestockBreedingAndGenomics/w5/simgenphencorr.csv`.

Wird die Analyse mit diesem zweiten Datensatz gemacht, und werden die Resultate der multiplen Regression mit den einzelnen einfachen Regressionen verglichen zeigt sich sehr schön der gewünschte Effekt, dass eben nur die multiple Regression nicht aber die einfachen Regressionen die Parameter aus der Simulation als Effekte aufweisen.

Das Einlesen der Daten erfolgt wie gewohnt mit:

```
> dfSimGenPhenDat <- read.csv2(file =
+   "http://charlotte-ngs.github.io/LivestockBreedingAndGenomics/w5/simgenphencorr.csv",
+   as.is = TRUE)
```

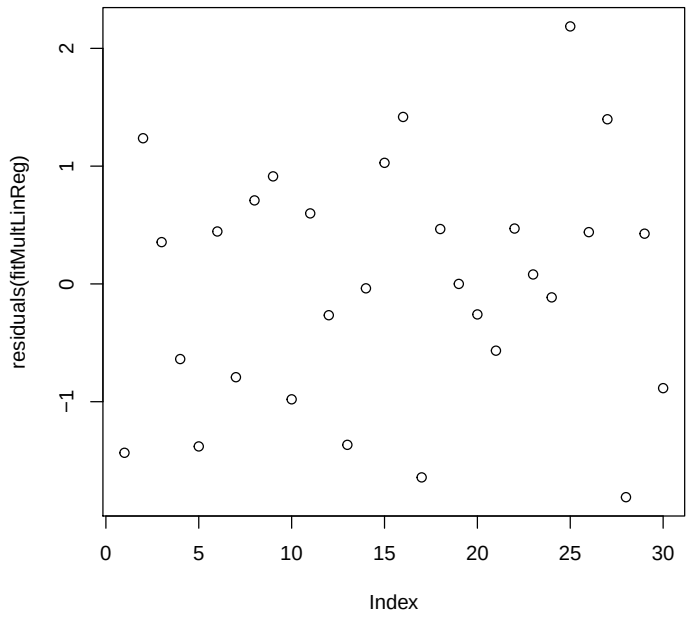
Die multiple lineare Regression

```
> fitMultLinReg <- lm(y ~ locus1 + locus2 + locus3, data = dfSimGenPhenDat)
> summary(fitMultLinReg)

Call:
lm(formula = y ~ locus1 + locus2 + locus3, data = dfSimGenPhenDat)

Residuals:
     Min       1Q   Median       3Q      Max
-1.81058 -0.75347  0.04015  0.56657  2.18623

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -11.3989     0.4855 -23.481  < 2e-16 ***
locus1       28.7382     0.5215  55.111  < 2e-16 ***
locus2       10.3505     0.9255  11.184 1.96e-11 ***
locus3        4.8390     0.5428   8.916 2.19e-09 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 1.06 on 26 degrees of freedom
Multiple R-squared:  0.995,        Adjusted R-squared:  0.9944
F-statistic:  1730 on 3 and 26 DF,  p-value: < 2.2e-16
```

Der Plot der Residuen zeigt kein erkennbares Muster.

```
> plot(residuals(fitMultLinReg))
```



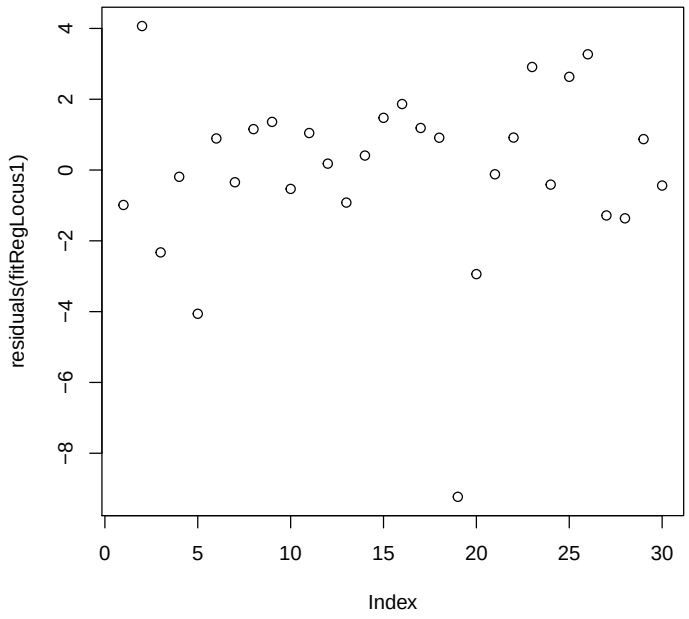
Vergleich mit einfachen Regressionen

Wird immer nur ein Locus in einer einfachen Regression gefittet sieht das Ergebnis ganz anders aus.

```
> fitRegLocus1 <- lm(y ~ locus1, data = dfSimGenPhenDat)
> summary(fitRegLocus1)
```

```
Call:
lm(formula = y ~ locus1, data = dfSimGenPhenDat)

Residuals:
    Min      1Q  Median      3Q     Max
-9.2314 -0.8229  0.2951  1.1797  4.0671

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -14.2291     0.7361  -19.33   <2e-16 ***
locus1       25.6102     0.8596   29.79   <2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 2.531 on 28 degrees of freedom
Multiple R-squared:  0.9694,        Adjusted R-squared:  0.9683
F-statistic: 887.6 on 1 and 28 DF,  p-value: < 2.2e-16
```

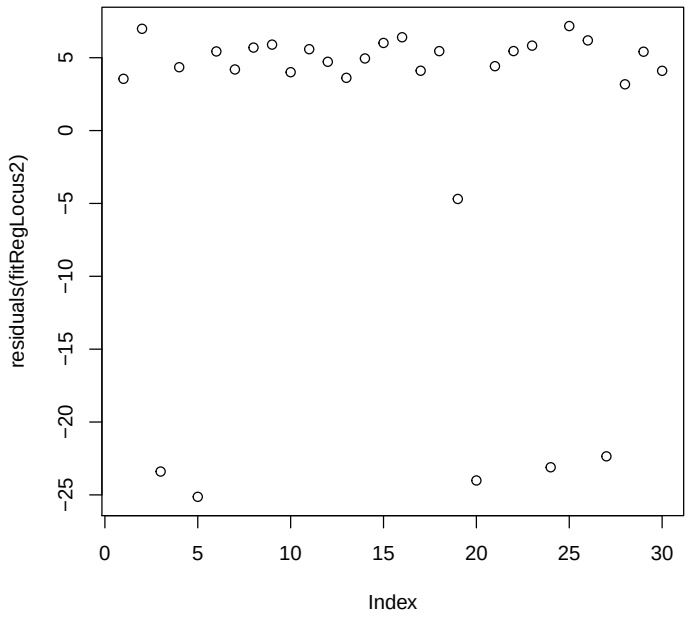
```
> plot(residuals(fitRegLocus1))
```




```
> fitRegLocus2 <- lm(y ~ locus2, data = dfSimGenPhenDat)
> summary(fitRegLocus2)
```

```
Call:
lm(formula = y ~ locus2, data = dfSimGenPhenDat)

Residuals:
    Min      1Q  Median      3Q     Max
-25.133   3.568   4.569   5.667   7.171

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -17.147      4.976  -3.446 0.001813 **
locus2       -23.989      5.450  -4.401 0.000142 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 11.13 on 28 degrees of freedom
Multiple R-squared:  0.4089,        Adjusted R-squared:  0.3878
F-statistic: 19.37 on 1 and 28 DF,  p-value: 0.0001421
```

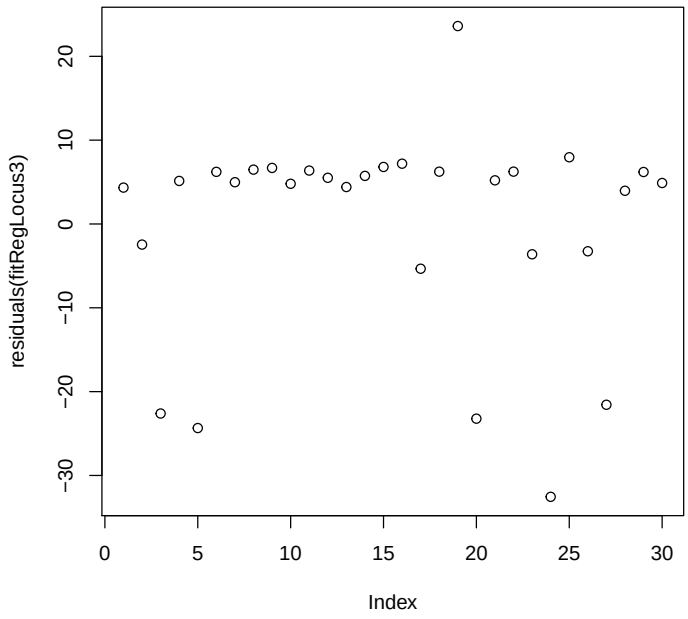
```
> plot(residuals(fitRegLocus2))
```




```
> fitRegLocus3 <- lm(y ~ locus3, data = dfSimGenPhenDat)
> summary(fitRegLocus3)
```

```
Call:
lm(formula = y ~ locus3, data = dfSimGenPhenDat)

Residuals:
    Min      1Q  Median      3Q     Max
-32.549  -3.057   5.058   6.243  23.610

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -7.703      4.249  -1.813  0.08062 .
locus3        13.757      4.655   2.955  0.00627 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 12.63 on 28 degrees of freedom
Multiple R-squared:  0.2378,        Adjusted R-squared:  0.2105
F-statistic: 8.734 on 1 and 28 DF,  p-value: 0.006271
```

```
> plot(residuals(fitRegLocus3))
```



Vor allem die Resultate der letzten beiden einfachen Regressionen zeigen, dass die Zerlegung einer multiplen linearen Regression in mehrere einfache Regressionen nicht funktioniert. Die geschätzten Effekte weisen eine grosse Dif-

ferenz auf zu den Parametern, welche in der Simulation verwendet wurden. Auch die Plots der Resiuduen zeigen offensichtliche Strukturen mit Ausreissern, welche auf eine ungünstige Modellwahl hindeuten. Die Schätzwerte der Restvarianzen sind bei den einfachen Regressionen um einiges höher im Vergleich zur multiplen linearen Regression, welche eine geschätzte Restvarianz im Bereich des in der Simulation verwendeten Wertes hat.

Alle Resultate in einer Tabelle zusammengefasst sehen wie folgt aus

Parameter	Simulation	Simple Regression	Multiple Regression
Locus 1	29	25.6	28.7
Locus 2	11	-24	10.4
Locus 3	5	13.8	4.8
Intercept	-10.76	-14.2 -17.1 -7.7	-11.4
Reststandardabweichung	1	2.53 11.13 12.63	1.06

Aufgabe 2 (5)

Analog zur in der Funktion `quadratzahl()` definieren wir die Funktion `convertCelsiusToFahrenheit()`. Als Input nimmt die Funktion die Temparatur in Grad Celsius als einziges Funktionsargument. Folgende Definition der Funktion `convertCelsiusToFahrenheit()` ist denkbar.

```
> convertCelsiusToFahrenheit <- function(pnTempC){
+   nResultTempF <-32 + (9/5) * pnTempC
+   return(nResultTempF)
+ }
```

Umrechnung einzelner Werte

Einzelne Temperaturwerte können mit einzelnen Funktionsaufrufen umgerechnet werden.

```
> convertCelsiusToFahrenheit(7)

[1] 44.6

> convertCelsiusToFahrenheit(13)

[1] 55.4

> convertCelsiusToFahrenheit(35)

[1] 95
```

Umrechnungstabelle

Angenommen, wir möchten die Temparaturen zwischen -50 und $+50$ Grad Celsius in Fahrenheit umrechnen und die Werte dann in einer Tabelle darstellen, dann können wir das in einem Loop machen.

```
> Celsius <- seq(from = -50,to = 50,by = 10)
> nNrValues <- length(Celsius)
> Fahrenheit <- vector(mode = "numeric", length = nNrValues)
> for (i in 1:nNrValues){
+   Fahrenheit[i] <- convertCelsiusToFahrenheit(Celsius[i])
+ }
> print(Celsius)

 [1] -50 -40 -30 -20 -10   0  10  20  30  40  50

> print(Fahrenheit)

 [1] -58 -40 -22  -4  14  32  50  68  86 104 122
```

Die Tabelle sieht dann wie folgt aus.

	Celsius	Fahrenheit
1	-50.00	-58.00
2	-40.00	-40.00
3	-30.00	-22.00
4	-20.00	-4.00
5	-10.00	14.00
6	0.00	32.00
7	10.00	50.00
8	20.00	68.00
9	30.00	86.00
10	40.00	104.00
11	50.00	122.00