

# Züchtungslehre - Lösung 2

Peter von Rohr

October 13, 2015

## Aufgabe 1 (6)

Gemäss Aufgabenstellung sind die beiden Unbekannten  $x$  (konsumierte Milchmenge in l/Tag) und  $y$  (konsumierte Menge an Orangensaft in l/Tag). Für den Bedarf an Calcium und Vitamin A kann je eine Gleichung aufgeschrieben werden. Die Koeffizienten der unbekanntes und die rechte Handseite der Gleichungen können folgender Tabelle aus der Aufgabenstellung entnommen werden.

	Milch (mg/l)	Orangensaft (mg/l)	Totaler Konsum (mg/Tag)
Calcium	1285	169	550
Vitamin A	1.89	2.03	1.2

Die Gleichungen in skalarer Schreibweise sehen wie folgt aus.

$$\begin{aligned}1285 * x + 169 * y &= 550 \\1.89 * x + 2.03 * y &= 1.2\end{aligned}\tag{1}$$

In Matrix-Vektor-Schreibweise definieren wir die folgende Matrix

$$\mathbf{A} = \begin{bmatrix} 1285 & 169 \\ 1.89 & 2.03 \end{bmatrix}$$

und die Vektoren

$$\mathbf{b} = \begin{bmatrix} x \\ y \end{bmatrix}$$

und

$$\mathbf{r} = \begin{bmatrix} 550 \\ 1.2 \end{bmatrix}$$

Die Gleichung sieht dann wie folgt aus

$$\mathbf{A} * \mathbf{b} = \mathbf{r}\tag{2}$$

Die Lösung für den Vektor  $\mathbf{b}$  kann berechnet werden durch Multiplizieren beider Seiten der Gleichung (2) von links mit der Inversen von  $\mathbf{A}$ , welche wir als  $\mathbf{A}^{-1}$  bezeichnen. Daraus erhalten wir folgende Gleichung für den Vektor  $\mathbf{b}$

$$\mathbf{b} = \mathbf{A}^{-1} * \mathbf{r} \quad (3)$$

In R erhalten wir die Zahlenwerte für die Komponenten des Vektors  $\mathbf{b}$  mit der Funktion `solve()`. Die Inputdaten aus der Tabelle der Aufgabenstellung werden zuerst einzelnen Objekten zugeordnet, woraus dann die Matrix  $\mathbf{A}$  und der Vektor  $\mathbf{r}$  zusammengestellt werden.

```
> nCaMilk <- 1285
> nCaOJuce <- 169
> nVitAMilk <- 1.89
> nVitAOJuce <- 2.03
> nCaTotReq <- 550
> nVitATotReq <- 1.2
> (matA <- matrix(data = c(nCaMilk, nCaOJuce, nVitAMilk, nVitAOJuce),
+                   ncol = 2,
+                   byrow = TRUE))

      [,1] [,2]
[1,] 1285.00 169.00
[2,]   1.89   2.03

> (vecR <- c(nCaTotReq, nVitATotReq))

[1] 550.0  1.2
```

Den Lösungsvektor erhalten wir dann mit

```
> (vecSolB <- solve(matA, vecR))

[1] 0.3991455 0.2195148
```

Das heisst also, dass der Bedarf an Calcium und Vitamin A durch den Konsum von 0.4 Liter Milch pro Tag und von 0.22 Liter Orangensaft pro Tag gedeckt werden kann.

## Aufgabe 2 (10)

In der folgenden Tabelle ist der Datensatz aus der Aufgabenstellung noch einmal dargestellt.

Brustumfang	Stockmass	Gewicht
176	158	471
177	163	463
178	164	481
179	158	470
179	163	496
180	166	491
181	168	518
182	168	511
183	171	510
184	171	541

Den Datensatz können wir direkt von der Vorlesungswebseite einlesen. Dazu verwenden wir die R-Funktion `read.csv2()`

```
> dfBrStGew <- read.csv2(file =  
+ "http://charlotte-ngs.github.io/LivestockBreedingAndGenomics/w3/br_st_gew.csv")
```

Als Kontrolle empfiehlt es sich immer, die Dimension des eingelesenen Datensatzes zu überprüfen.

```
> dim(dfBrStGew)
```

```
[1] 10 3
```

1. Die Regression der Variablen "Gewicht" auf die Variable "Brustumfang" wird mit der Funktion `lm()` gerechnet. Mit der Funktion `summary()` werden die Resultate angezeigt.

```
> fitBr <- lm(Gewicht ~ Brustumfang, data = dfBrStGew)  
> summary(fitBr)
```

Call:

```
lm(formula = Gewicht ~ Brustumfang, data = dfBrStGew)
```

Residuals:

```
      Min       1Q   Median       3Q      Max  
-17.3941  -6.5525  -0.0673   9.3707  13.2594
```

Coefficients:

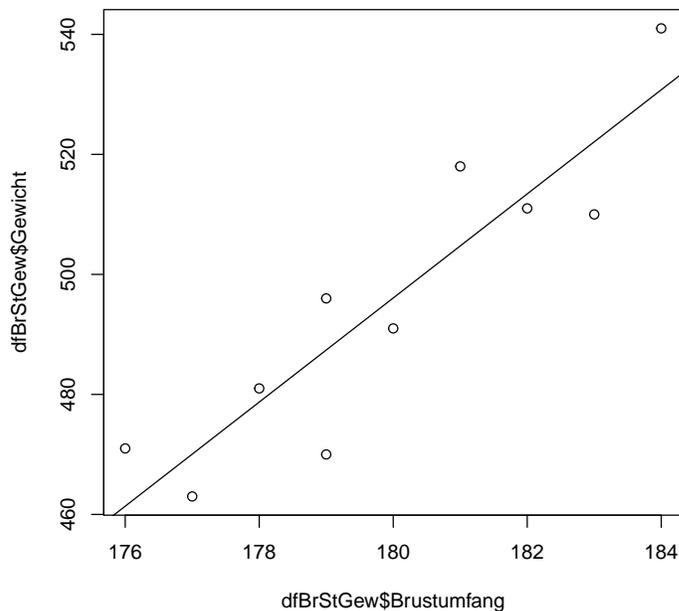
	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-1065.115	255.483	-4.169	0.003126 **
Brustumfang	8.673	1.420	6.108	0.000287 ***

---  
 Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.08 on 8 degrees of freedom  
 Multiple R-squared: 0.8234, Adjusted R-squared: 0.8014  
 F-statistic: 37.31 on 1 and 8 DF, p-value: 0.000287

Der Plot der Punkte und die Regressionsgeraden werden mit den Funktionen `plot()` und `abline()` erzeugt. Damit der Punkteplot des Gewichts auf den Brustumfang erscheint, so wie in der Vorlesung gezeigt, ist es in diesem Beispiel wichtig die  $x$ - und die  $y$ - Achse explizit als Argumente für den Plotbefehl anzugeben. Da in der Vorlesung der Dataframe nur zwei Kolonnen aufwies, konnten wir die `plot()` Funktion mit dem Dataframe als einziges Argument aufrufen.

```
> plot(dfBrStGew$Brustumfang, dfBrStGew$Gewicht)
> abline(coefficients(fitBr))
```



2. Analog mit der Regression der Variablen "Gewicht" auf die Variable "Stockmass".

```
> fitStm <- lm(Gewicht ~ Stockmass, data = dfBrStGew)
> summary(fitStm)
```

Call:

```
lm(formula = Gewicht ~ Stockmass, data = dfBrStGew)
```

Residuals:

Min	1Q	Median	3Q	Max
-23.129	-9.432	4.371	8.782	18.588

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-253.1333	160.1333	-1.581	0.15259
Stockmass	4.5354	0.9702	4.675	0.00159 **

---

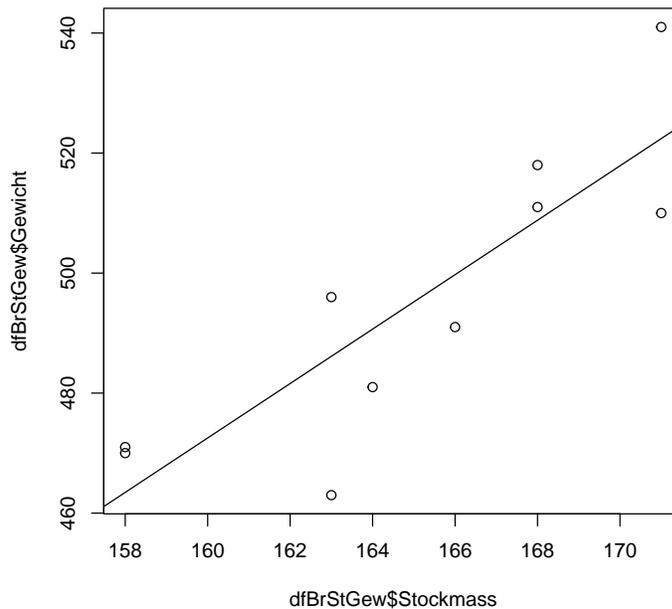
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.65 on 8 degrees of freedom

Multiple R-squared: 0.732, Adjusted R-squared: 0.6985

F-statistic: 21.85 on 1 and 8 DF, p-value: 0.001592

```
> plot(dfBrStGew$Stockmass, dfBrStGew$Gewicht)
> abline(coefficients(fitStm))
```



3. Die multiple linear Regression wird wieder mit der Funktion `lm()` gerechnet. Die beiden x-Variablen "Brustumfang" und "Stockmass" werden über ein Plus verknüpft.

```
> fitMultReg <- lm(Gewicht ~ Brustumfang + Stockmass, data = dfBrStGew)
> summary(fitMultReg)
```

Call:

```
lm(formula = Gewicht ~ Brustumfang + Stockmass, data = dfBrStGew)
```

Residuals:

Min	1Q	Median	3Q	Max
-13.646	-8.744	-1.247	10.470	11.486

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-899.014	325.562	-2.761	0.0280 *
Brustumfang	6.465	2.978	2.171	0.0665 .
Stockmass	1.401	1.651	0.848	0.4244

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

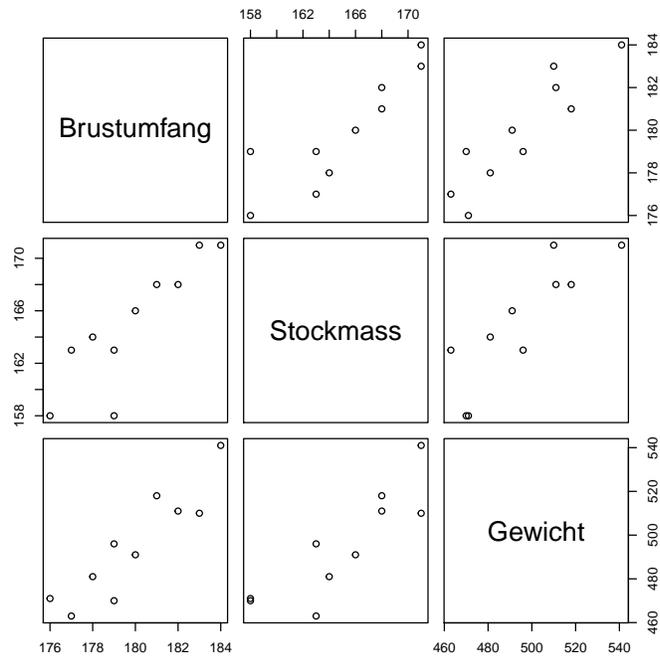
Residual standard error: 11.28 on 7 degrees of freedom

Multiple R-squared: 0.8399, Adjusted R-squared: 0.7941

F-statistic: 18.36 on 2 and 7 DF, p-value: 0.001643

Aufgrund des Resultats der multiplen Regression sehen wir, dass die Variable "Stockmass" einen hohen  $p$ -Wert hat und somit nur sehr wenig zur Erklärung des Gewichts beiträgt. Durch den hohen  $p$ -Wert kann diese Variable "Stockmass" aus dem Modell weggelassen werden. Der Grund dafür ist aus dem sogenannten `pairs()` Plot ersichtlich. Der `pairs()` Plot kann mit der Funktion `pairs()` oder mit der Funktion `plot()` erzeugt werden.

```
> pairs(dfBrStGew)
```



Der `pairs()` Plot zeigt den engen Zusammenhang zwischen den Variablen “Stockmass” und “Brustumfang”. Dieser Zusammenhang kann durch die Korrelation zwischen “Stockmass” und “Brustumfang” quantifiziert werden. Diese Korrelation beträgt

```
> cor(dfBrStGew$Brustumfang, dfBrStGew$Stockmass)
```

```
[1] 0.8742386
```