

Mehrfache und polynomiale Regression

Jonathan Harrington

Einfache Regression

$$\hat{y} = bx + k$$

Mehrfache Regression

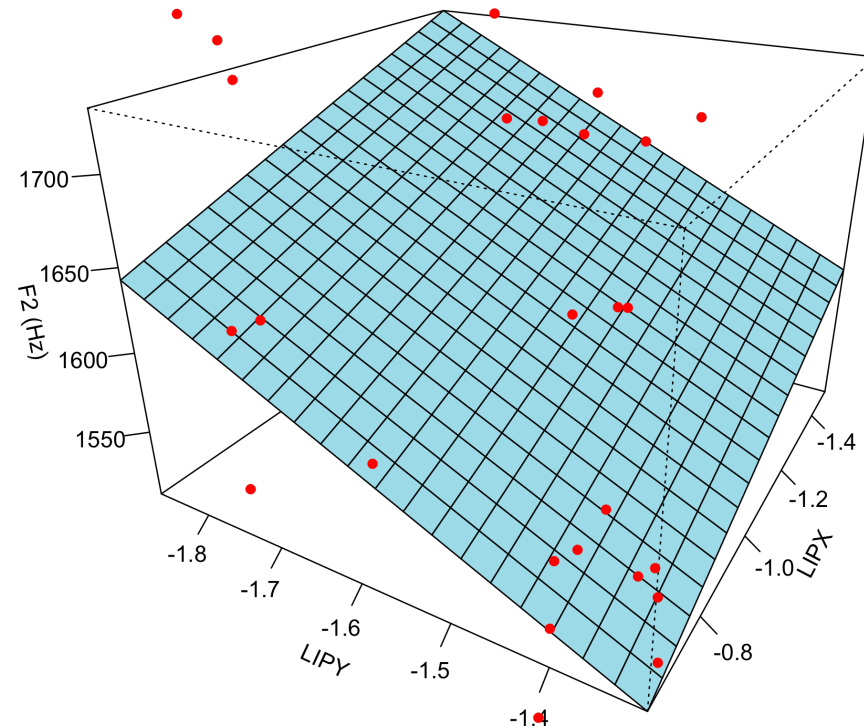
$$\hat{y} = b_1x_1 + b_2x_2 + k$$

In diesem Fall: 2

Regressoren (x_1, x_2), 2

Neigungen (b_1, b_2), ein
Intercept, k

und eine Ebene
im 3D-Raum



Mehrfache Regression

Es können auch mehrere Regressoren sein...

$$\hat{y} = b_1x_1 + b_2x_2 + b_3x_3 + \dots b_nx_n + k$$

n verschiedene Neigungen, ein Intercept

Eine Hyper-Ebene in einem n -dimensionalen Raum

Einige Daten

```
ydata = read.table(file.path(pfadu, "ydata.txt"))
```

```
names(ydata)
```

Zungendorsum Untere Lippe

```
[1] "F2" "DORSY" "DORSX" "LIPY" "LIPX"
```

[y] Vokale, alle Werte zum zeitlichen Mittelpunkt

DORSX, DORSY (horizontale und vertikale Position des Zungendorsums)

LIPX, LIPY (horizontale Verlagerung und vertikale Position der Unterlippe)

Ein mehrfaches Regressionsmodell

$$\hat{F2} = b_1 \text{DORSX} + b_2 \text{DORSY} + b_3 \text{LIPX} + b_4 \text{LIPY} + k$$

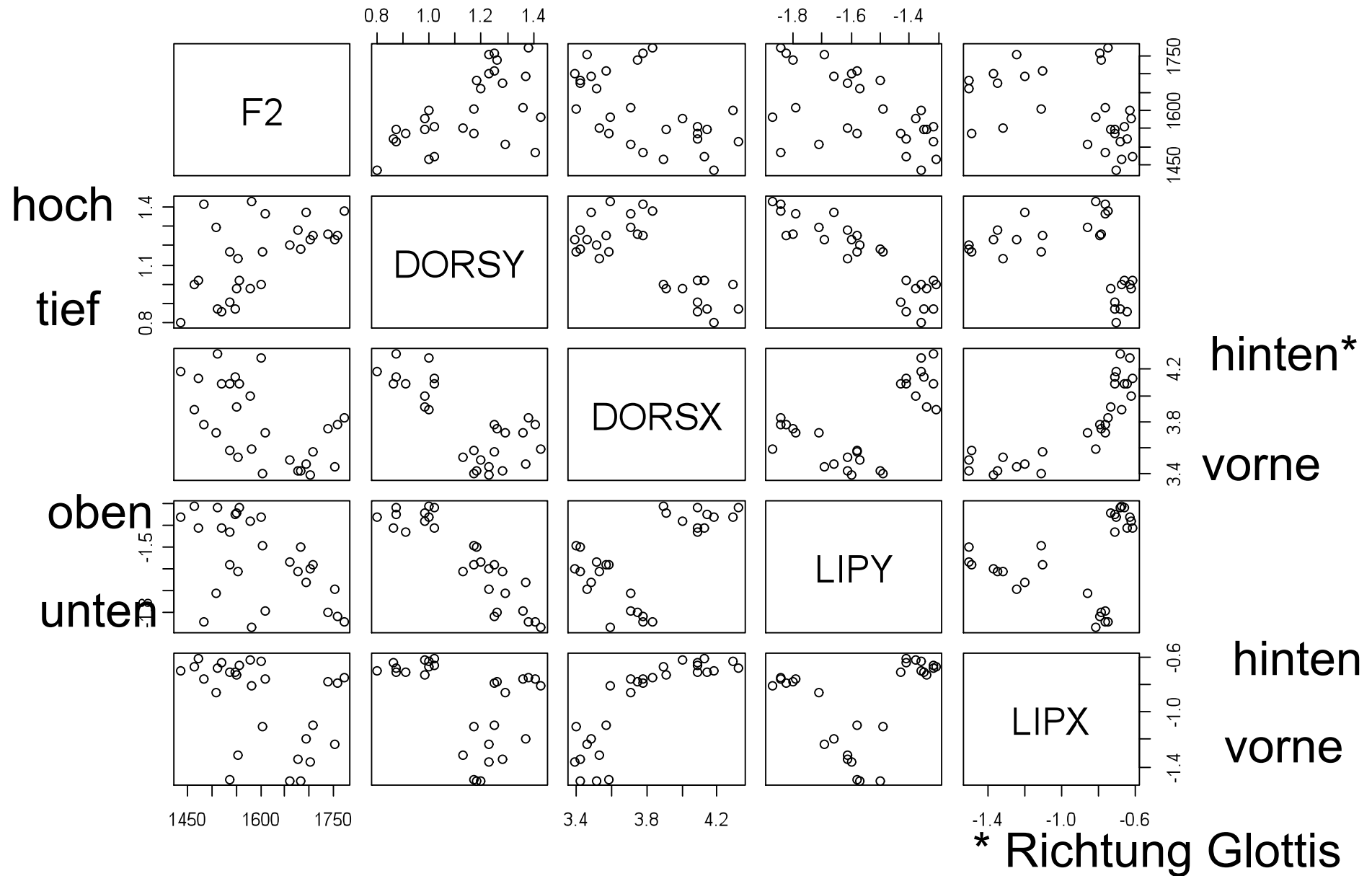
hat die Bedeutung

$\hat{F2}$ = ein Gewicht Mal die horizontale Position der Zunge +
ein anderes Gewicht Mal die vertikale Position der Zunge +
.... + ein Intercept

Mehrfache Regression in R

- Festlegung von b_1, b_2, b_3, b_4, k
- Sind alle diese Parameter notwendig? Wenn nicht, welches Parameter oder Parameterkombination hat die deutlichste lineare Beziehung zu $F2$?

pairs(ydata)



$$\hat{F2} = b_1 \text{DORSX} + b_2 \text{DORSY} + b_3 \text{LIPX} + b_4 \text{LIPY} + k$$

```
regm = lm(F2 ~ DORSX+DORSY+LIPX+LIPY, data=ydata)
```

Koeffiziente

```
coef(regm)
```

summary(regm)

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1355.05	822.63	1.647	0.113
DORSX	-38.33	157.21	-0.244	0.810
DORSY	63.08	254.27	0.248	0.806
LIPX	-67.36	110.90	-0.607	0.550
LIPY	-164.08	205.04	-0.800	0.432

Residual standard error: 83 on 23 degrees of freedom

Multiple R-Squared: 0.3939, Adjusted R-squared: 0.2884

F-statistic: 3.736 on 4 and 23 DF, p-value: 0.01747

F2 kann mit einer multidimensionalen Regression aus diesen artikulatorischen Parametern modelliert werden:

Adjusted $R^2 = 0.29$, $F[4, 23] = 3.7$, $p < 0.05$.

(In mehrfacher Regression Adjusted R^2 nehmen, weil adjusted R^2 für die Anzahl der Regressoren kompensiert)

Modell-Prüfung durch AIC (Akaike's Information Criterion)

Mit der `stepAIC()` Funktion in `library(MASS)` wird geprüft, ob für die Regression wirklich alle (in diesem Fall 4) Regressoren benötigt werden.

Je kleiner AIC, umso nützlicher die Kombination für die Regression (umso höher adjusted R^2)

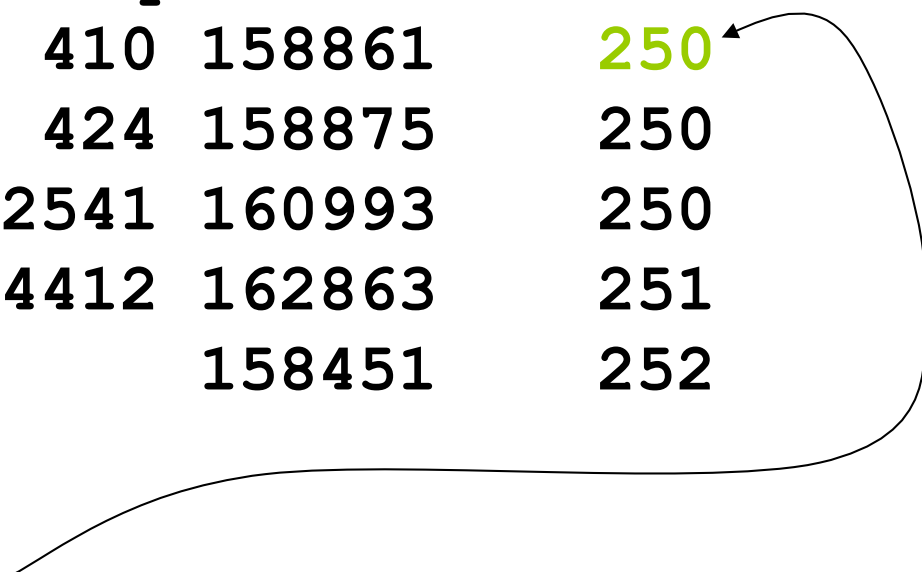
`library(MASS)`

`stepAIC(regm)`

Start: **AIC= 251.95**

F2 ~ DORSX + DORSY + LIPX + LIPY

	Df	Sum of Sq	RSS	AIC
- DORSX	1	410	158861	250
- DORSY	1	424	158875	250
- LIPX	1	2541	160993	250
- LIPY	1	4412	162863	251
<none>			158451	252



sortiert nach AIC. **Dies ist der AIC-Wert**, wenn **aus diesem Modell** DORSX weggelassen wäre.

- Vor allem ist dieser AIC Wert weniger als **AIC mit allen Parametern (= 251.95)**.
- Daher wird im nächsten Modell DORSX weggelassen.

Step: **AIC= 250.02**

F2 ~ LIPX + LIPY + DORSY

	Df	Sum of Sq	RSS	AIC
- DORSY	1	1311	160172	248
- LIPY	1	4241	163102	249
<none>			158861	250
- LIPX	1	16377	175238	251

AIC ist am kleinsten, wenn aus diesem Modell DORSY weggelassen wird. Und dieser Wert ohne DORSY ist kleiner als derjenige mit LIPX+LIPY+DORSY zusammen.

Daher wird DORSY weggelassen...

Step: **AIC= 248.25**

F2 ~ LIPX + LIPY

	Df	Sum of Sq	RSS	AIC
<none>			160172	248
- LIPX	1	25225	185397	250
- LIPY	1	50955	211127	254

Wenn wir entweder LIPX oder LIPY weggelassen, dann wird AIC **höher** im Vergleich zu **AIC mit beiden Parametern zusammen**.

Daher bleiben wir bei **F2 ~ LIPX + LIPY**

Dieses Modell $F2 \sim LIPX + LIPY$ müsste auch den höchsten adjusted R^2 haben. Prüfen, zB:

```
lip.lm = lm(F2 ~ LIPX+ LIPY, data= ydata)
```

```
summary(regm)
```

Adjusted R-squared: 0.2884

```
summary(lip.lm)
```

Adjusted R-squared: 0.3383

Also wird die Variation in F2 in [y] am meisten durch die horizontale und vertikale Position der Unterlippe erklärt.

Polynomiale Regression

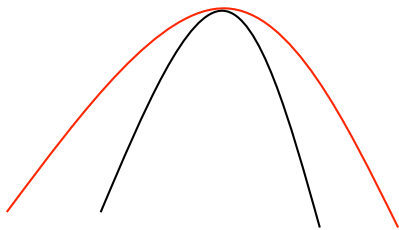
ein Regressor ein Koeffizient

$$\hat{y} = bx + k$$

ein Regressor, 2 Koeffiziente

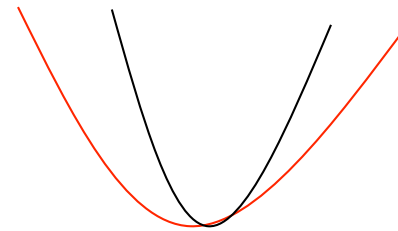
$$\hat{y} = b_1x + b_2x^2 + k$$

bestimmt die Krümmung;



b_2 ist negativ

b_2 ist näher an 0



b_2 ist positiv

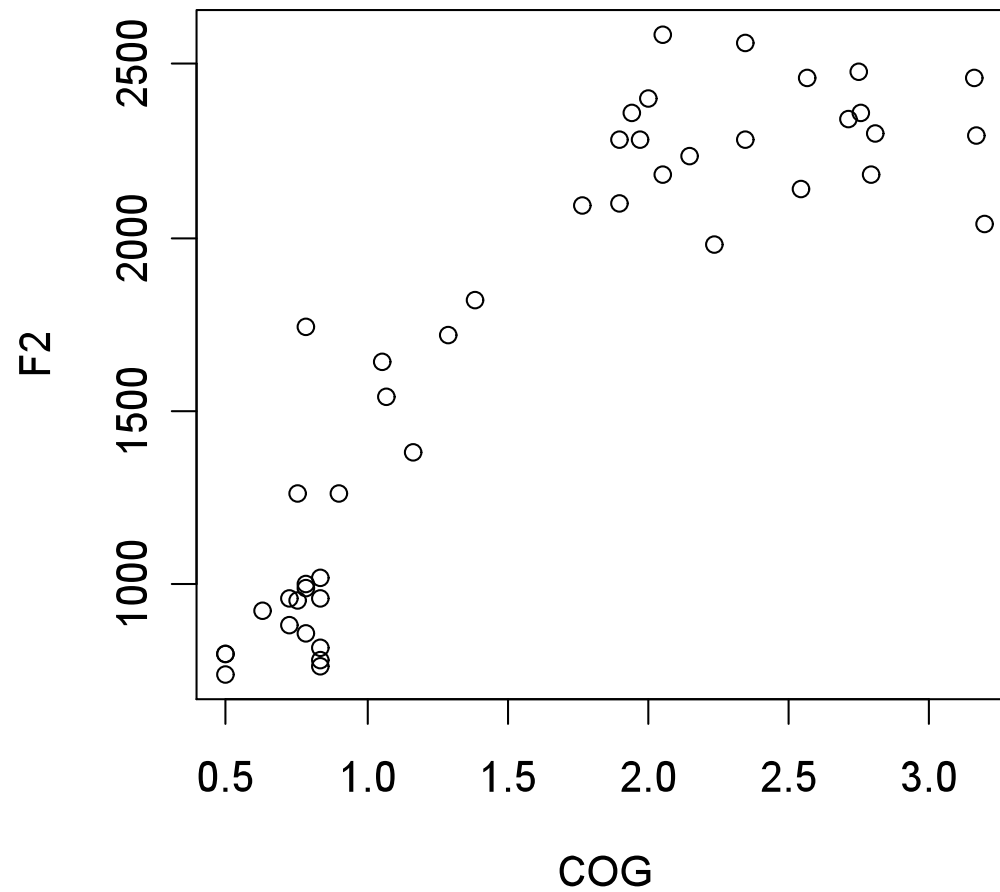
ein Regressor, n Koeffiziente

$$\hat{y} = b_1x + b_2x^2 + b_3x^3 + b_4x^4 + \dots b_{n-1}x^n + k$$

In allen Fällen handelt es sich um Abbildung/Beziehungen im **2D**-Raum (da wir mit **einem** Regressor zu tun haben).

```
epg = read.table(paste(pfadu, "epg.txt", sep="/"))
```

```
plot(F2 ~ COG, data = epg)
```



$$\hat{F2} = b_1 COG + b_2 COG^2 + k$$

```
regp = lm(F2 ~ COG + I(COG^2), data = epg)
```

```
k = coef(regp)
```

```
(Intercept)    COG  I(COG^2)
-294.3732  2047.8403 -393.5154
```

$$\hat{F2} = 2047.8 COG - 393.5 COG^2 - 294.3732$$

```
plot(F2 ~ COG, data = epg)
```

Die Parabel überlagern

```
curve(k[1] + k[2]*x + k[3]*x^2, add=T)
```

summary(regp)

Beide Komponente, COG und COG² der Parabel scheinen notwendig zu sein, um die F2-COG Beziehungen zu modellieren.

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-294.37	139.95	-2.103	0.0415	*
COG	2047.84	192.83	10.620	1.81e-13	***
I(COG^2)	-393.52	54.17	-7.264	6.10e-09	***

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

Residual standard error: 202.1 on 42 degrees of freedom
Multiple R-Squared: 0.9092, Adjusted R-squared: 0.9049
F-statistic: 210.4 on 2 and 42 DF, p-value: < 2.2e-16

mit `stepAIC()` kann wieder festgestellt werden, ob wir den COG^2 Parameter wirklich benötigen:

`stepAIC(regp)`

Start: AIC= 480.69

F2 ~ COG + I(COG²)

	Df	Sum of Sq	RSS	AIC
<none>			1715550	481
- I(COG ²)	1	2155469	3871019	515
- COG	1	4606873	6322423	537

Call:

`lm(formula = F2 ~ COG + I(COG2))`

Coefficients:

(Intercept)	COG	I(COG ²)
-294.4	2047.8	-393.5

Scheinbar ja (da AIC höher wird, wenn entweder COG oder COG^2 aus dem Modell weggelassen werden).