Angewandte Statistik

Julian Hinz — Universität Bielefeld

Session 2

Parametrische Regression

Wiederholung: Lineare Regression

Abgaben

1. Abgabe: 24. Mai (Aufgaben am 17. Mai)
2. Abgabe: 5. Juli (Augaben am 28. Juni)

Frist: AoE — “Anywhere on Earth”

an: abgabe-internationale-vwl@uni-bielefeld.de

Tutorien

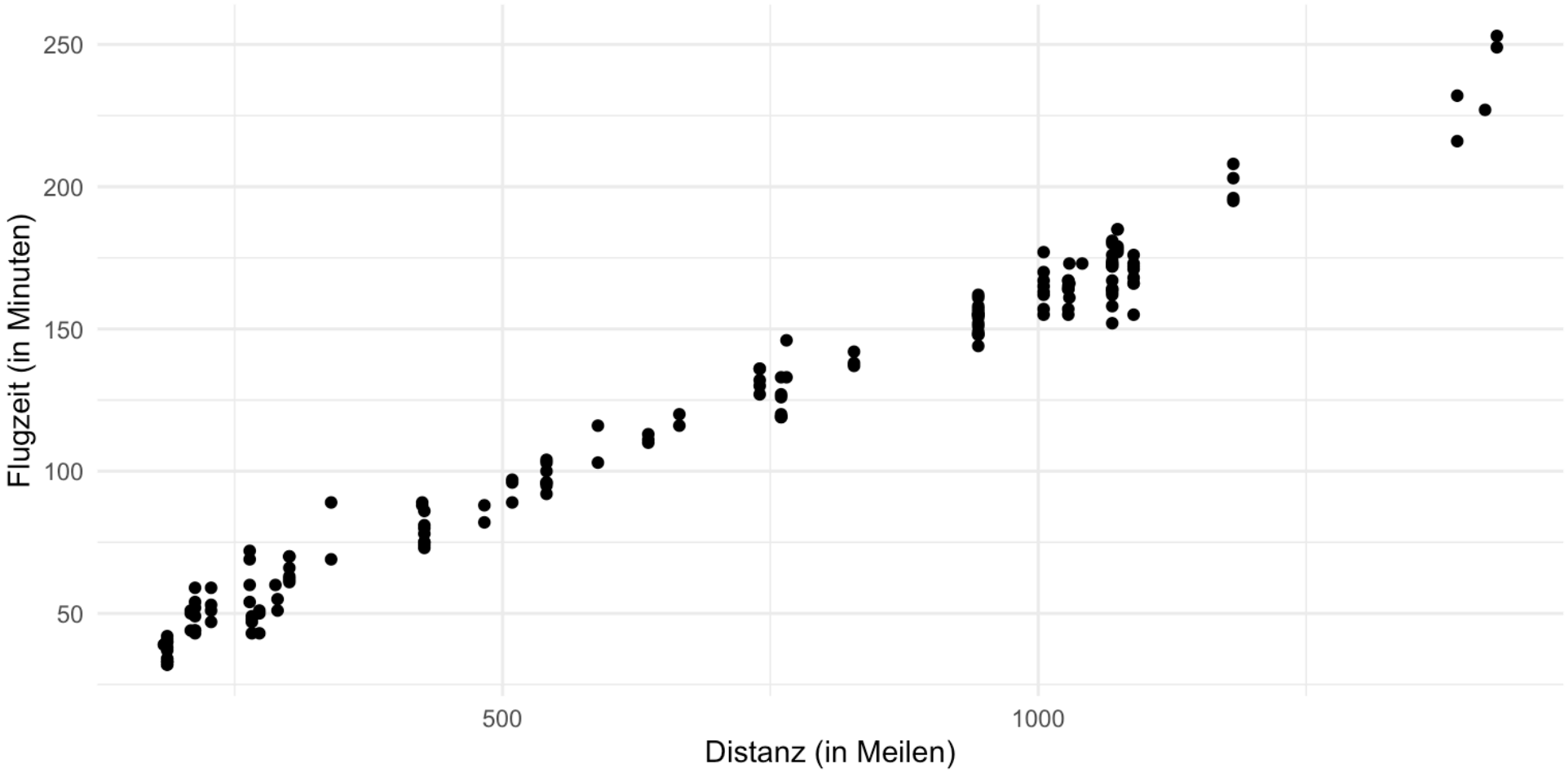
- 197 Anmeldungen für 120 Plätze
- kurzfristige Lösung: Live-Streaming per Zoom
- wir versuchen noch zusätzlichen Termin einzurichten

Parametrische Regressionsanalyse

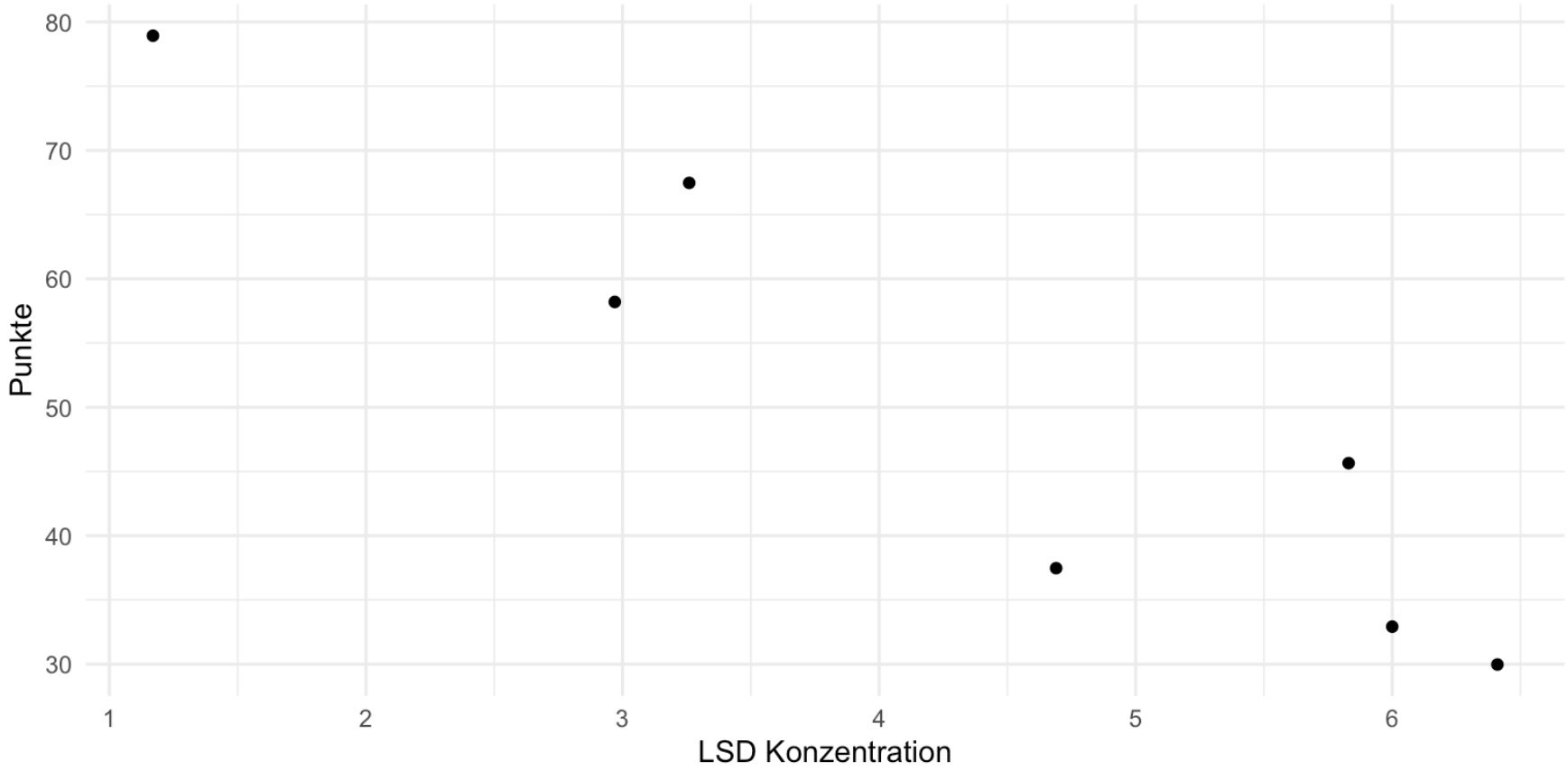
Parametrische Regressionsanalyse

(Wie) hängt bestimmte **Zielgröße** (Y) von **erklärenden Variablen** ($x_1, \ldots, x_p$) ab?

Flugzeit vs. Flugstrecke



Punkte im Mathetest vs. LSD-Konzentration im Blut



Beispiele

- Wirtschaftliche Größe und Exporte
- Mietpreis hängt ab von Wohnfläche, Baujahr, Ausstattung, Lage, etc.
- WiWi-Note hängt ab von Abischnitt, Lernaufwand, etc.
- Gebrauchtwagenpreis hängt ab von Kilometerstand, Marke, Motorleistung, etc.
- …

Analyse von Abhängigkeiten zwischen Variablen

Wir wollen aus Daten ein **Modell** lernen (d.h. schätzen), welches den Einfluss der erklärenden Variablen (x_1, , x_p) auf die Zielgröße (Y) gut beschreibt:

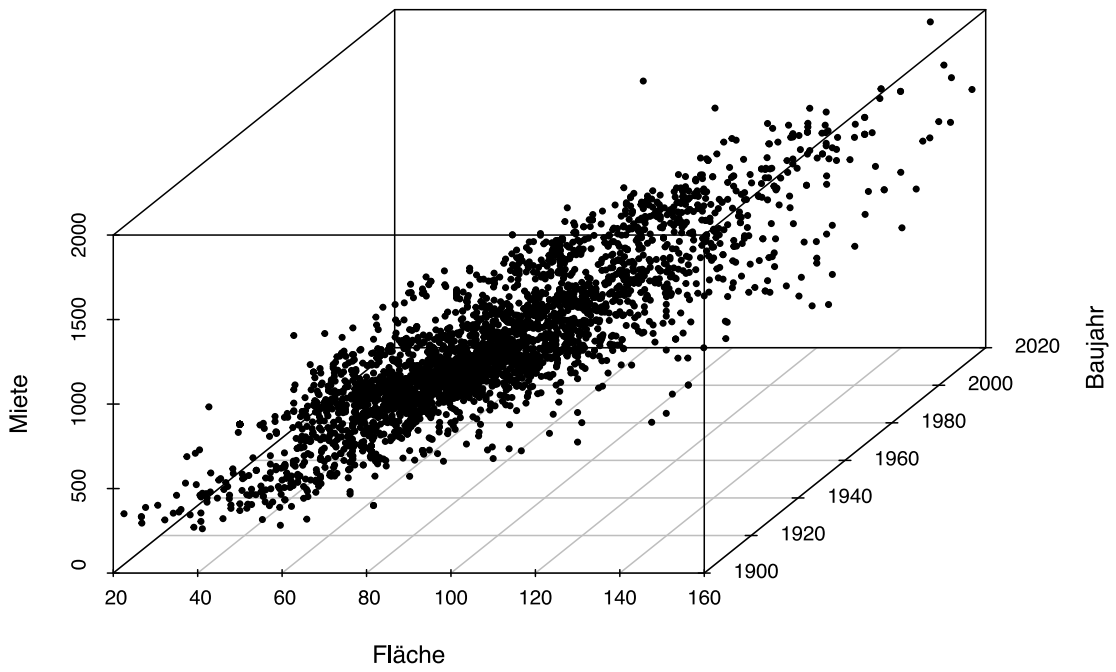
$$Y = f(x_1, \ldots, x_p) + \text{zufälliger Fehler}$$

Wieso brauchen wir den Fehlerterm?

- Ein “perfekter” Zusammenhang liegt bei realen Prozessen so gut wie nie vor.
- Oftmals wird (Y) von weiteren, nicht beobachteten Variablen beeinflusst.

Im Idealfall erklärt (f) einen Großteil der Variabilität von (Y), so dass der Fehlerterm klein ist, was zu ziemlich genauen Vorhersagen führen kann.

Mietpreis (in Euro/Monat) vs. Wohnfläche (in qm) und Baujahr, für Bielefelder Mietwohnungen



Regression — allgemeine Formulierung und Ziele

Statistische Modellierung der Form

$$Y = f(x_1, \dots, x_p) + \epsilon$$

nennt man **Regression** (oder **Regressionsanalyse**)

Konkrete Ziele der Regression könnten u.a. sein

- Vorhersage von Werten (z.B. Flugzeiten)
- Zusammenhänge aufzeigen/testen ob ein Einfluss vorliegt (z.B. Einfluss von Drogen und kognitiver Leistung)
- Identifikation von Ausreißern (z.B. Mietpreise/Wucher)

Modellformulierung

Lineares Regressionsmodell

$$\underbrace{Y_i}_{\text{Zielgröße}} = \underbrace{\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}}_{\text{system. Effekt}} + \underbrace{\epsilon_i}_{\text{Fehler}}, \quad E(\epsilon_i) = 0, \quad i = 1, \ldots, n.$$

Hier: *lineare* Form des systematischen Teils $f(x_{i1}, \ldots, x_{ip})$

Warum ist diese Formulierung so wichtig?

- überraschend oft realistisch
- relativ robust gegenüber einzelnen Ausreißern
- sehr einfach umzusetzen und zu interpretieren
- viel flexibler als es auf den ersten Blick scheint

Einfaches lineare Modell ($p = 1$) in Matrixnotation

Die Modellspezifikation

$$Y_i = \beta_0 + \beta_1 x_{i1} + \epsilon_i, \quad i = 1, \dots, n,$$

umfasst genau genommen n Gleichungen, eine für jeden Datenpunkt:

$$\begin{aligned} Y_1 &= \beta_0 + \beta_1 x_{11} + \epsilon_1 \\ Y_2 &= \beta_0 + \beta_1 x_{21} + \epsilon_2 \\ &\vdots \\ Y_n &= \beta_0 + \beta_1 x_{n1} + \epsilon_n \end{aligned}$$

Überführung in Matrixnotation

$$
\begin{aligned}
Y_1 &= \beta_0 + \beta_1 x_{11} + \epsilon_1 \\
&\vdots \\
Y_n &= \beta_0 + \beta_1 x_{n1} + \epsilon_n
\end{aligned}
\quad \rightsquigarrow \quad
\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \epsilon,
$$

$$
\text{wobei} \quad \mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_{11} \\ 1 & x_{21} \\ \vdots & \vdots \\ 1 & x_{n1} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}, \quad \epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}
$$

Multiples lineares Modell ($p \geq 1$) in Matrixnotation

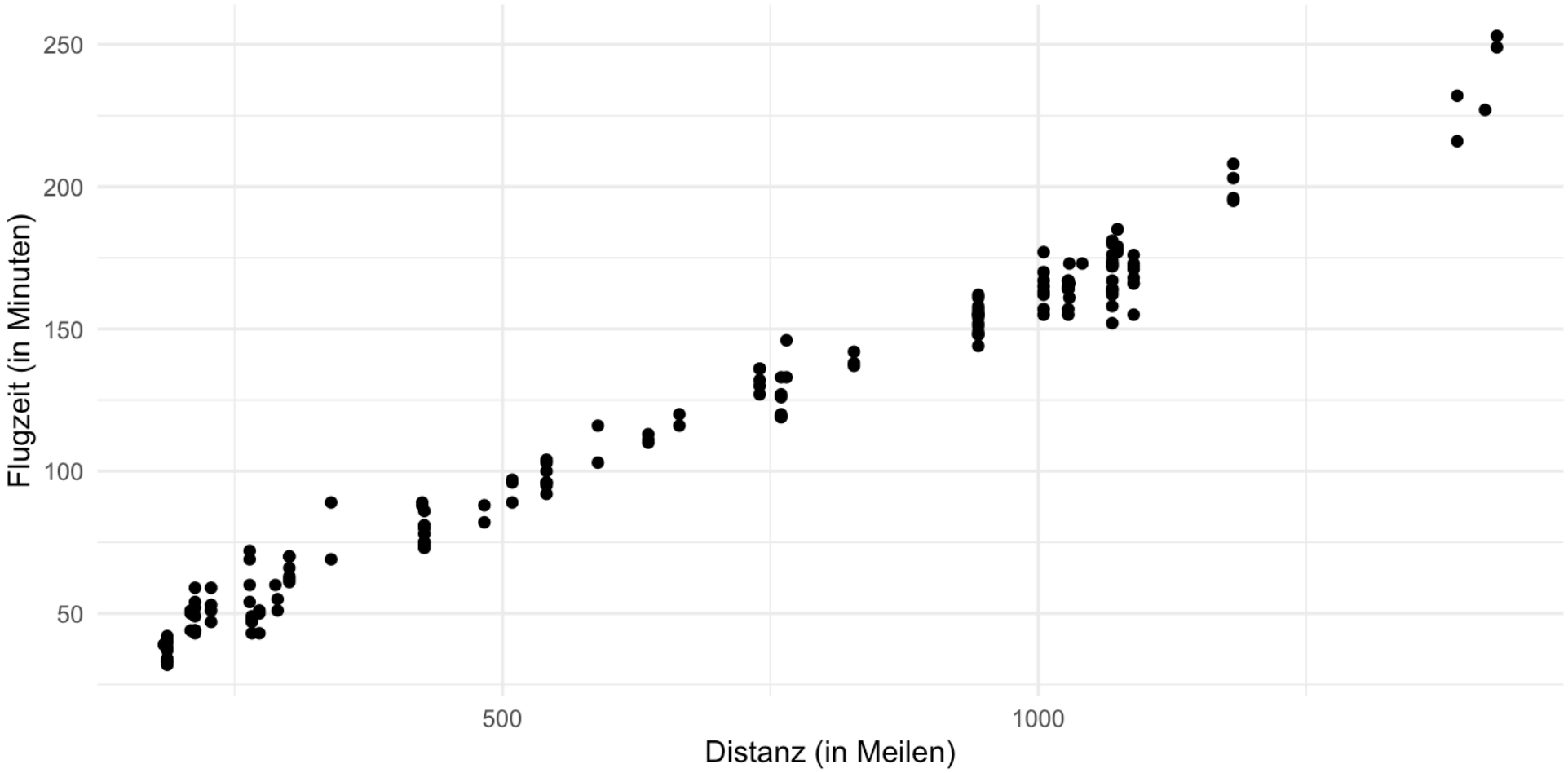
Komplett analog für das multiple lineare Modell

$$
\begin{aligned}
Y_1 &= \beta_0 + \beta_1 x_{11} + \ldots + \beta_p x_{1p} + \epsilon_1 \\
&\vdots \qquad\qquad\qquad\qquad\qquad\qquad \rightsquigarrow \qquad \mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \epsilon, \\
Y_n &= \beta_0 + \beta_1 x_{n1} + \ldots + \beta_p x_{np} + \epsilon_n
\end{aligned}
$$

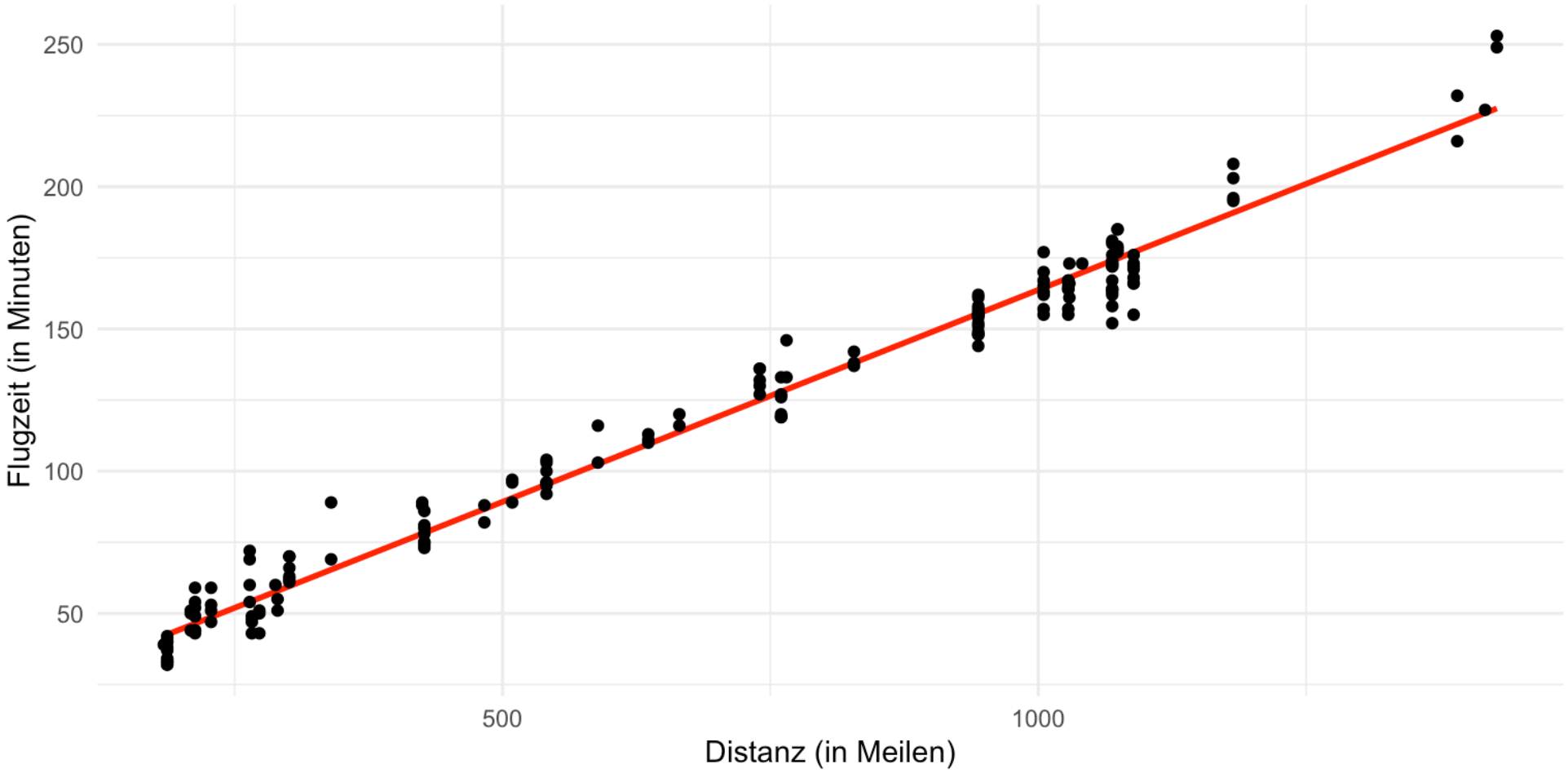
$$
\text{wobei} \quad \mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_{11} & \ldots & x_{1p} \\ 1 & x_{21} & \ldots & x_{2p} \\ \vdots & \vdots & & \\ 1 & x_{n1} & \ldots & x_{np} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}
$$

Parameterschätzung

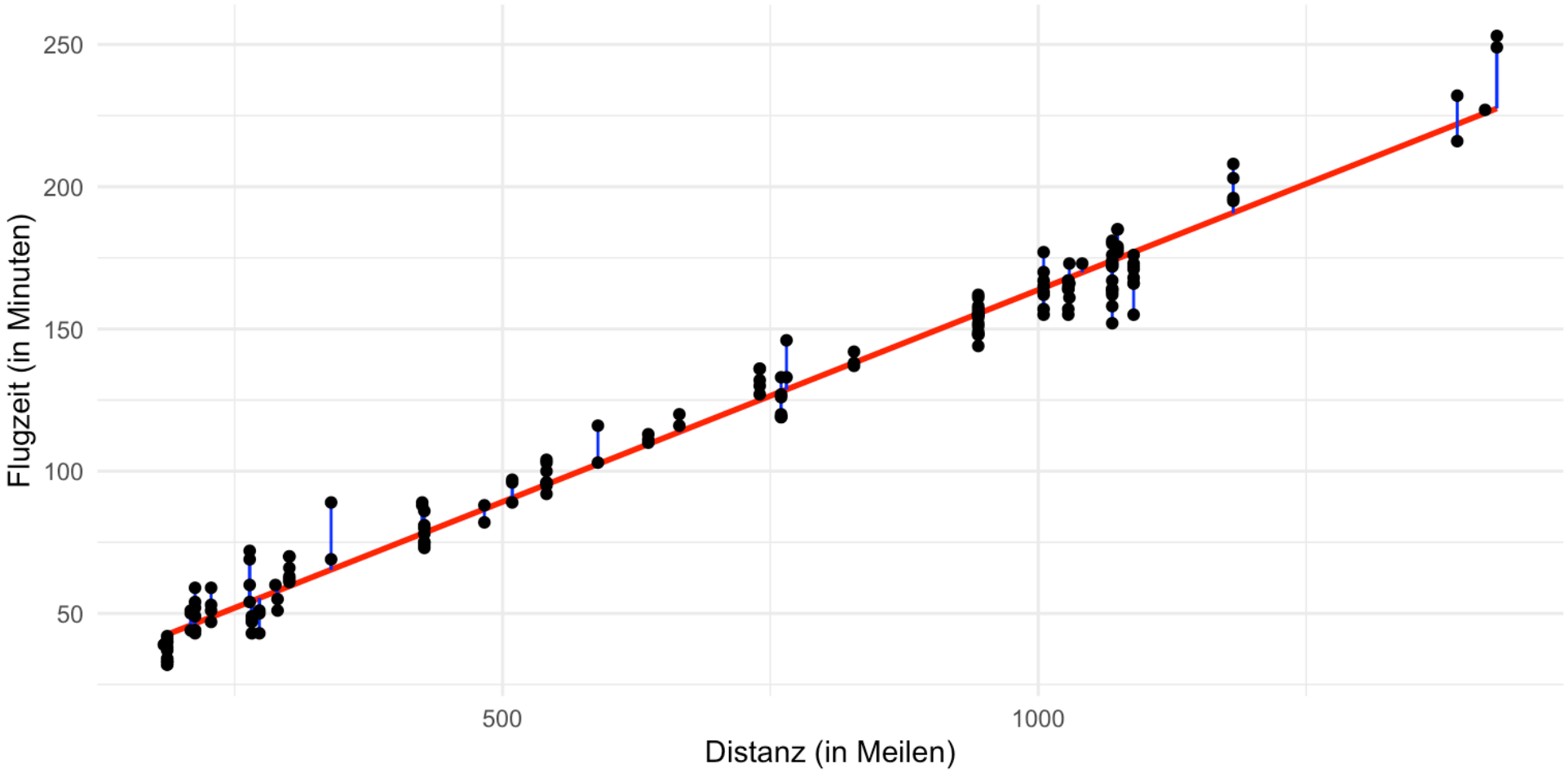
Schätzung der Regressionskoeffizienten



Schätzung der Regressionskoeffizienten



Schätzung der Regressionskoeffizienten



Schätzung der Regressionskoeffizienten

Als “Gesamtabstand” zwischen Modell und Datenpunkten definieren wir die Summe der quadrierten Fehler:

$$S(\beta_0, \beta_1, \ldots, \beta_p) = \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} \Big(y_i - (\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \Big)^2$$

Der **Kleinste-Quadrate-Schätzer** (KQS),

$$\widehat{\boldsymbol{\beta}} = (\widehat{\beta}_0, \widehat{\beta}_1, \ldots, \widehat{\beta}_p) = \underset{\beta_0, \beta_1, \ldots, \beta_p}{\operatorname{argmin}} \; S(\beta_0, \beta_1, \ldots, \beta_p),$$

minimiert diesen Abstand.

KQS für das multiple lineare Regressionsmodell

Für das **multiple lineare Regressionsmodell**

$$Y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + \epsilon_i, \quad E(\epsilon_i) = 0, \quad i = 1, \ldots, n,$$

in Matrixnotation

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

ist der **Kleinste-Quadrate-Schätzer** gegeben durch

$$\hat{\boldsymbol{\beta}} = \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_p \end{pmatrix} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

KQS in R für LSD-Beispiel

Mit Hilfe von `lm`-Funktion:

```r
library(data.table)

daten = fread("../data/data2/LSD.txt")

regression <- lm(Punkte ~ LSD,
                 data = daten)
print(regression)
```

```
Call:
lm(formula = Punkte ~ LSD, data = daten)

Coefficients:
(Intercept)          LSD
     89.124       -9.009
```

KQS in R für LSD-Beispiel

“zu Fuß” mit Matrix Algebra:

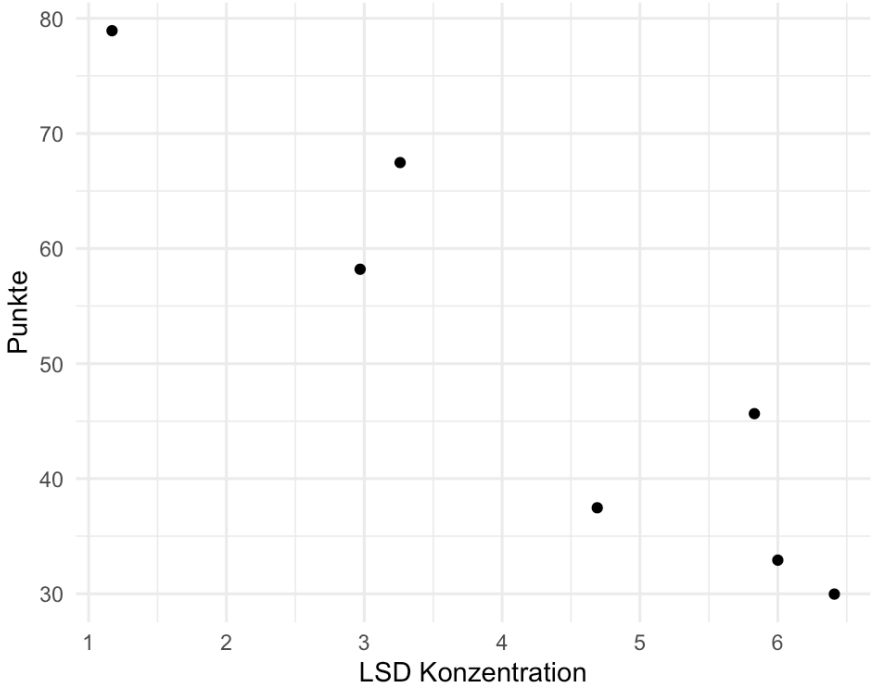
```r
daten = fread("../data/data2/LSD.txt")

X = cbind(rep(1, 7), daten$LSD)
y = daten$Punkte

beta = solve(t(X) %*% X) %*% t(X) %*% y
print(beta)
```

```
          [,1]
[1,] 89.123874
[2,] -9.009466
```

KQS in R für LSD-Beispiel

“zu Fuß” mit Matrix Algebra:

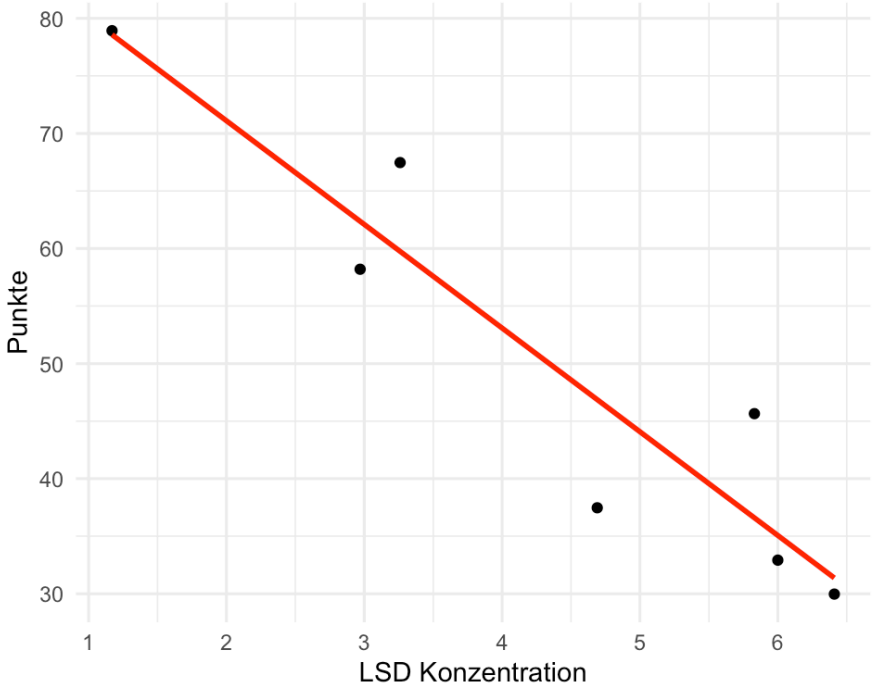
```r
p_load(ggplot2)
p_load(data.table)

# daten laden
data = fread("../data/data2/LSD.txt")

# plotten
ggplot(data, aes(x = LSD, y = Punkte)) +
    theme_minimal() +
    geom_point() +
    scale_x_continuous("LSD Konzentration") +
    scale_y_continuous("Punkte")
```



KQS in R für LSD-Beispiel

“zu Fuß” mit Matrix Algebra:

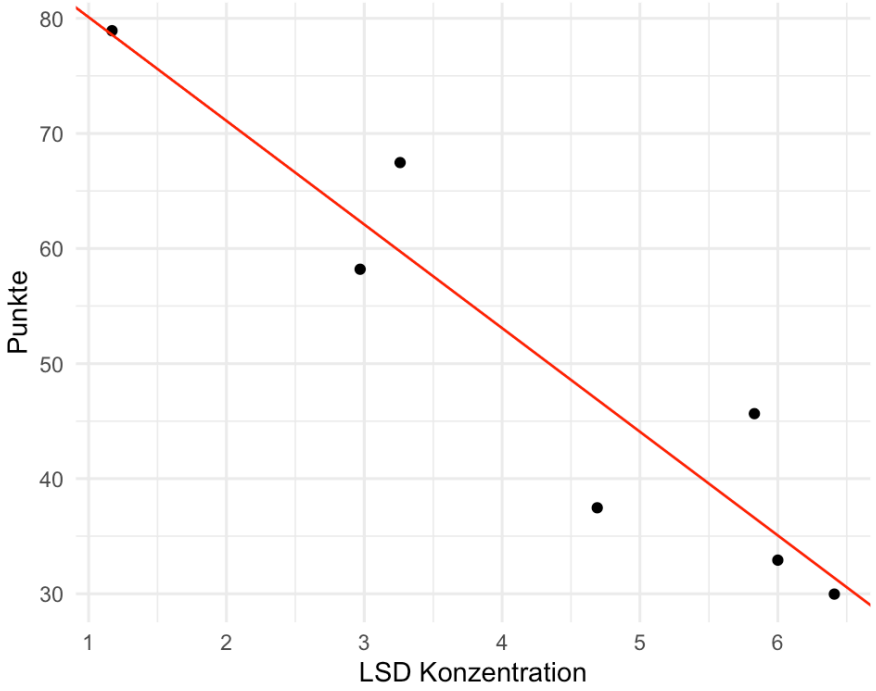
```
p_load(ggplot2)
p_load(data.table)

# daten laden
data = fread("../data/data2/LSD.txt")

# plotten
ggplot(data, aes(x = LSD, y = Punkte)) +
    theme_minimal() +
    geom_point() +
    geom_smooth(method = "lm", color = "red",
    scale_x_continuous("LSD Konzentration") +
    scale_y_continuous("Punkte")
```



KQS in R für LSD-Beispiel

“zu Fuß” mit Matrix Algebra:

```
p_load(ggplot2)
p_load(data.table)

# daten laden
data = fread("../data/data2/LSD.txt")

# regression manuell rechnen
X = cbind(rep(1, 7), daten$LSD)
y = daten$Punkte
beta = solve(t(X) %*% X) %*% t(X) %*% y

# plotten
ggplot(data, aes(x = LSD, y = Punkte)) +
    theme_minimal() +
    geom_point() +
    geom_abline(intercept = beta[1], slope = be
    scale_x_continuous("LSD Konzentration") +
    scale_y_continuous("Punkte")
```



KQS in R für Mietspiegel-Beispiel

Mit Hilfe von - Funktion:

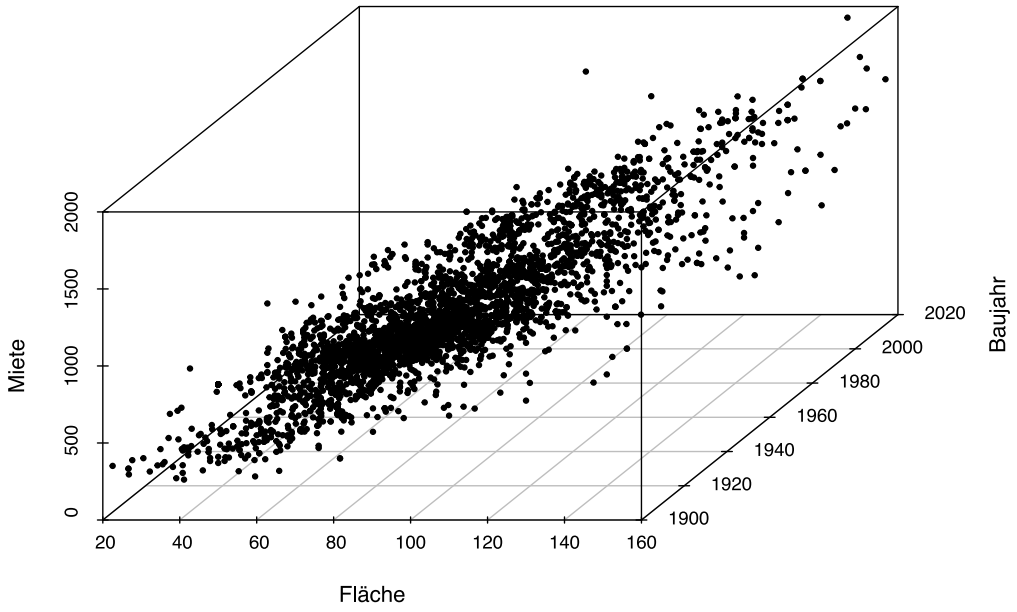
```r
library(data.table)

# daten laden
daten = fread("../data/data2/rents.csv")

# lm funktion
regression = lm(rent ~ area + year,
                data = daten)
summary(regression)
```

```
Call:
lm(formula = rent ~ area + year, data = daten)

Residuals:
    Min      1Q  Median      3Q     Max 
-386.54  -73.32   -3.15   69.16 1062.81 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -2.147e+03  1.459e+02  -14.71   <2e-16 ***
area         7.906e+00  9.265e-02   85.33   <2e-16 ***
year         1.086e+00  7.452e-02   14.58   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
```

```r
library(data.table)

# daten laden
daten = fread("../data/data2/rents.csv")

# zu fuß
X = cbind(rep(1,3854), daten$area, daten$year)
y = daten$rent
beta = solve(t(X) %*% X) %*% t(X) %*% y

print(beta)
```

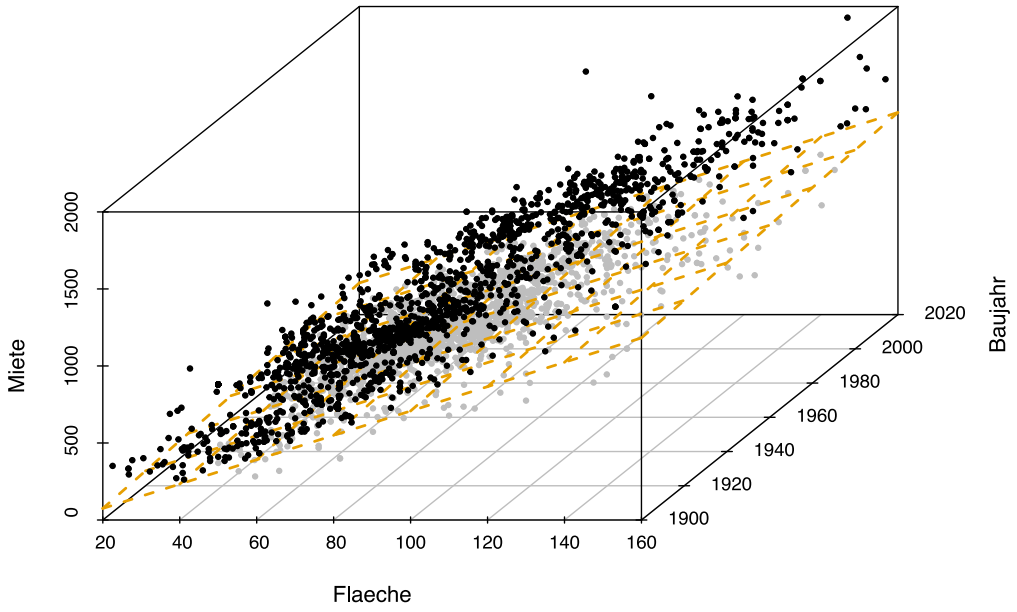
```
             [,1]
[1,] -2146.724893
[2,]     7.906339
[3,]     1.086172
```

Angepasstes Modell im Mietspiegel–Beispiel



$$\text{Miete}_i = -2147 + 7.91 \cdot \text{Fläche}_i + 1.09 \cdot \text{Baujahr}_i + \epsilon_i$$

Angepasstes Modell im Mietspiegel–Beispiel



$$\text{Miete}_i = -2147 + 7.91 \cdot \text{Fläche}_i + 1.09 \cdot \text{Baujahr}_i + \epsilon_i$$