Einfache lineare Regression

Modell: $Y_i = \alpha + \beta x_i + E_i$ mit E_i unabhängig $\sim N\langle 0, \sigma^2 \rangle$, normalverteilt mit Erwartungswert = 0 und konstanter Varianz

Schätzung: Kleinste-Quadrate-Kriterium
- fit <- lm(Y ~ X)
  coef(fit)

Genauigkeit und Vertrauensintervalle für Koeffizienten:
- summary(fit)
  confint(fit, level=0.95) zusätzlich parm = 1 oder 2
  oder : coef(lm1)[2] + qt(0.995,n (-2)) *
  summary(lm1)$coef[2,2] * c(-1,1)

Genauigkeit und Vertrauensintervalle für $\mathbb{E}\langle y_i \rangle$ (deckt nur unsicherheit der Schätzung ab)**:**
- predict(fit, se.fit=TRUE, ...)
  predict(fit, newdata=data.frame(x=c(1, 1/10)),
  interval="confidence", level=0.95) -> Bsp. Windmühlen

Prognoseintervall für $\mathbb{E}\langle y_i \rangle$: (+ σ^2, Varianz E_i -> somit grösseres Intervall als Vertrauensintervall, deckt Variabilität aus Fehler E_i ab)
In welchem Bereich liegt eine zukünftige Beobachtung zum Niveau 95%?
- predict(fit, newdata=xyz, interval="prediction", level=0.95

# Rücktransformation exp(b_0 +b_1*lN + b_2*lC + b_3*P + E_i) -> exp(b_0) * N^{b1} * C^{b2} * exp(b_3*P) + exp(E_i) (kann Bias enthalten)
exp(h) oder mit Korrektur: exp(h + summary(lm1)$sigma^2/2)

Statistische Aussagen sind nur vertrauenswürdig, wenn Modellannahmen erfüllt sind

Hauptziele:
1. Eine Gerade in eine Punktwolke legen, um die Beziehung zwischen einer Zielgrösse Y und einer erklärenden Variablen x zu beschreiben.
2. Bestimmen, ob die Daten mit einem Modell mit vorgegebenen Koeffizienten verträglich sind (statistischer Hypothesentest).
3. Angeben, wie genau der Achsenabschnitt und die Steigung aus den Daten bestimmt werden können (Vertrauensintervall)

Allgemein
Residuen (R_i): «Näherungswerte», Differenzen zwischen den Beobachtungen und den angepassten Werten
Hut: Angepasste Werte (geschätzt)
α = Achsenabschnitt
β = Steigung
E_i = Zufallsfehler

Prüfen der Modelleignung
➔ Linearität vorausgesetzt

Annahmen Fehler E_i:
1. Erwartungswert = 0
2. Alle Fehler haben die gleiche Varianz σ^2
3. Sind normalverteilt

Sind unabhängig

R-Code
Daten einlesen:
MPI <- read.table(paste("Statistisches Modellieren/Arbeitsblätter/MPIZH.dat", sep = ""), header = T)

Gerade (Fitter) konstruieren
uhr.lm <- lm(P ~ A, data=Uhr)
coef(uhr.lm) summary(uhr.lm)

Schätzen der Geraden
paste("y_i =",round(lm1$coef[1],3), "+", round(lm1$coef[2],3), "x_i")
1. J. ältere Uhr: 1* xi -> Preiserhöhung

In Plot einfügen
plot(Uhren$Alter, Uhren$Preis, xlab = "Alter", ylab = "Preis")
abline(uhr.lm, col="red")

Kleinste Quadrate-Methode R^2: (Bestimmung Güte)
$\frac{\text{Quadratsumme Fit}}{\text{Quadratsumme Y}}$.
-> liegt zwischen 0 - 1 (je näher an 1, desto besser der lineare Zusammenhang)

Misst den Anteil der durch die Regression erklärten Streuung der Y-Werte. - Identisch zur Korrelation^2 zwischen Zielvariablen yi und $\hat{y}i$

➔ kein Mass für Eignung des Regressionsmodells
➔ misst die Stärke des linearen Zusammenhangs
➔ Modellannahme muss erfüllt sein (β_0 muss vorhanden sein)

Technik / Naturwissenschaften: Werte > 0.9 üblich
Geistes-/ Sozialwissenschaften: Werte um 0.6 i.O.

R-Code (allgemein)
Mit Matrix x Werte generieren:
x.sim <- c(0,3,2,4,8)
E.sim <- c(matrix(rnorm(10*100, mean=0, sd=2), ncol=100)
y.sim <- 4 + 2*x.sim + E.sim
Koef <- matrix(0, ncol=2, nrow=100)

for (i in 1:100) {
Koef[i,] <- coef(lm(y.sim[,i] ~ x.sim)) }
hist(Koef[, 1]) *# 1 = α, 2 = ß*

Beobachtungen entfernen
Forbes <- lm(y ~ x, Forbes[-12,])
ODER: Forbes <- lm(y ~ x, Forbes, subset = -12)

R^2
lm(…): 2. unterste Zeile «Multiple R-squared» oder summary(lm1)$r.squared

Diagnose Instrumente:

➔ Sicherstellung, dass in den Daten keine für die Theorie gefährlichen Abweichungen zu den Voraussetzungen
➔ Residuen Realisierungen von Zufallsgrösser -> Voraussetzungen können nie exakt erfüllt sein

Tukey-Anscombe-Diagramm
(Residuen gegen angepasste Werte)
➔ Rauschen um die Gerade

Linearitätsannahme E = 0 eingehalten? (konstant innerhalb der stochastischen Fluktuation?)
Robuste Methode (ohne Einfluss Ausreisser)
Residuen sollten gleich streuen (konstante Varianz)
2 Sägezähne: 2 Gruppen Formen beschrieben

Teststatistik
R-Code
tc <- coef(lm1)[2] / summary(lm1)$coef[2,2]

$$T = \frac{\hat{\beta} - \beta^{\circ}}{\mathrm{se}^{(\beta)}}$$

Nullhypothese:
P-Wert: < 0.05 -> Verwerfen / ablehnen, da signifikant

qt(0.975, df=11) *# Grenze auf 5 % Niveau mit 11 Freih.graden*

$$\frac{Estimate - Nullhypothese}{Std.\,Error}$$

Grün < orange ➔ Nullhypothese kann nicht abgelehnt werden

Glätter zeigt eine Abweichung von horizontalen Geraden: (Mitte):
Beurteilung Abweichung anhand Bootstrap-Simulation (mehr Sicherheit, da mehr Simulationen vorhanden, Zufall ausschliessen):
1. Zufällige Beobachtungen erzeugen (Zufallszahlen generieren)
2. Regressionsrechnung durchführen, Glätter in das Tukey-Anscombe-Diagramm einzeichnen
3. Schritte repetieren

(Kann auch auf Streuungsdiagramm und normalen QQ-Plot angewandt werden)
Resultat:
In Beispiel wurden 19 Kurven erzeugt, da in Anlehnung auf das Testen auf dem 5 % Niveau (1/20 -> 19 + 1 = 20)
Nicht kritisch (siehe graue Kurven), Beobachtung oben in Ecke extrem -> Glätter muss innerhalb der stochastischen Fluktuation liegen (Graue Linien / Punkte)

R-Code
SD.lm <- lm(SZ ~ Liefvolumen, data = SD)

par(mfrow=c(2,4))
plot(SD.lm) *# Ohne which: alle 4 Diagr.*
source("Statistisches Modellieren/Arbeitsblaetter/RFn_Plot-lmSim.R") *# Laden Funktion Plot.lmSim*
plot.lmSim(x.fit, SEED=567) *# Bootstrap-Simulation 3 Diagramme*
plot.lmSim(x.fit, SEED=567, rob=TRUE) *# Verdacht auf Langschw. -> SD robust schätzen: rob = TRUE*

plot(Fit, which = 4)
plot(lmFit, which=5, add.smooth=FALSE)

1: Erwartungswert / 2: Normalverteilung der Fehler / 3: Varianz / 4: Distanz gegen Beobachtungsnummer / 5: Residuen gegen Hebel

Streuungs-Diagramm (scale-location-plot)
-> Varianz konstant? Sägezähne vorhanden?
Varianz ist anfälliger auf Ausreisser als Mittelwert
Glätter auf $\sqrt{|R_i|}$ anwenden

Histogramm -> N(0, σ²)
Sind Fehler normalverteilt?
Beurteilung, von Auge schwierig:
- Bei wenig Daten (< einige 100) Form schlecht erkennbar
- Sensitiv auf Balkeneinteilung
- Nicht-lineare Strukturen werden miteinander verglichen

Vergleichbar mit Normalverteilungs-Dichte in R: hist(…., freq=F)

QQ-Plot / Normalverteilungs-Diagramm
Sind Fehler normalverteilt? Langschwänzigkeit / Ausreisser?
Falls Ja: Punkte streuen um eine Gerade
➔ Für die Y_i sinnlos, da verschiedene Erwartungswerte

R-Code:
qqnorm() / qqline() -> Linie in Punkten

Skalierte Residuen – Verteilung der Zufallsfehler überprüfen, wenn Residuen bereits benützt
Fehler normalverteilt: Residuen von einer R^2-Schätzung auch normalverteilt
! Nicht gleiche Varianz: Diese hängt von erklärenden Variabel x_i ab

Skaliert:
$$\frac{R_i}{\sqrt{1 - \left(\frac{1}{n} + \frac{(x_i - \bar{x})^2}{SS_X}\right)}} \sim \mathcal{N}\langle 0, \sigma^2 \rangle$$

Standardisiert:
$$\frac{R_i}{\hat{\sigma}\sqrt{1 - \left(\frac{1}{n} + \frac{(x_i - \bar{x})^2}{SS_X}\right)}}.$$

Vorgehen:
- Grundsätzlich skalierte Residuen verwenden bei Varianz = 1 -> standardisierte Residuen anwenden
- Tukey-Anscombe-Diagramm: Rohresiduen verwenden (Unkorrelation)

Je weiter Beobachtung vom Schwerpunkt entfernt, desto kleiner die Varianz und desto näher geht die Regressionsgerade am Beobachtungspunkt vorbei

Behandlung von Unzulänglichkeiten

First Aid Transformationen
Häufig: Abhängigkeit der Streuung von $\hat{y}$ (nimmt im Diagramm zu, Normal-Plot zeigt eine rechtsschiefe Verteilung) -> Logarithmieren der Zielgrösse hilft

Logarithmus-Transformation
Für Konzentrationen und Beträge / Mengen
summary(data) → Daten anschauen, Differenz Min/ Max < als Faktor 2 → Log lohnt sich nicht

Wurzeltransformation
Für Zähldaten

Arcus-Sinus-Wurzel-Trasformatioin / Logit-Transformation

$\tilde{y} = \arcsin(\sqrt{y}) \quad resp. \tilde{y} = \log\left(\frac{y + 0.005}{1.01 - y}\right)$

Für Anteile / Prozentzahlen

Transformation der Zielvariablen ändert Form der Verteilung der Fehler
→ Potenzgesetz für die ursprünglichen Grössen, somit ist der Fehler proportional
Falls β = 1 ist die Zielvariable proportional zu x bis auf einen multiplikativen zufälligen Fehler.

Rücktransformation ergibt nicht immer dasselbe Ergebnis wie zuvor

Erwartungswert ist nicht konstant 0
Systematische Abweichungen im Erwartungswert können oft durch:
- Transformation der erklärenden Variablen x
- Oder durch Hinzufügen eines zusätzlichen Terms x^2 (quadr. Regression)

zum Verschwinden gebracht werden.

Ausreisser - Fehler sind nicht normalverteilt
Auf Richtigkeit der Daten prüfen!
→ Transformationen können helfen

Ausreisser entfernen:
D.synt.lm2 <- lm(y ~ x1 + x2, data=D.synt, subset=-c(7,17,27))

Langschwänzigkeit - Fehler sind nicht normalverteilt
Extremste Beobachtungen weglassen, bis Langschwänzigkeit verschwindet (! optimistisch)

Kleinste-Quadrate-Methode nicht optimal -> nur robuste Methoden geeignet (geringere Gewichtung von Ausreissern)

Unabhängigkeit zufällige Fehler
Bei Beobachtungen mit zeitlicher Reihenfolge, können Autokorrelationen vorhanden sein.
→ Residuen in dieser Reihenfolge auftragen (keine Strukturen ersichtlich?)
→ oder R_t gegen R_{t-1} in Streudiagramm auftragen – Punkte sollten frei streuen und keine Korrelation zeigen
Wenn Autokorrelationen vorliegen:
- P-Werte der üblichen Tests häufig grob falsch

Vertrauensintervalle üblicherweise zu kurz

Multiple lineare Regression

$Y_i = \beta_0 + (\beta_1 x_i)^{(1)} + (\beta_2 x_i)^{(2)} + \cdots + (\beta_m x_i)^{(m)} + E_i$

Beispiel:
$\log(ersch) = \beta_0 + \beta_1 \log(dist) + \beta_2 \log(lad) + E_i$

Zufällige Fehler Ei: Unabhängig
Koeffizienten β_j(unb. Parameter): normalverteilt
Schätzung der Koeffizienten β_j: Kl. Quadrate Meth.

Erklärende Variablen xi:
- Keinen bestimmten Datentyp
- Keine bestimmte Verteilung (keine Zufallsvar.)
- Keine Voraussetzung über ihre Abhängigkeit untereinander

Bestimmtheitsmass (Multiple R-Squared)
- Analog einf. lin. Regression
- Misst Anteil der durch Regressfunktion erklärten Streuung an der Streuung der Y-Werte $R^2 = \frac{\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$
- Entspricht dem Quadrat der Korrelation zwischen den beobachteten und den angepassten Werten (linearer Zusammenhang)

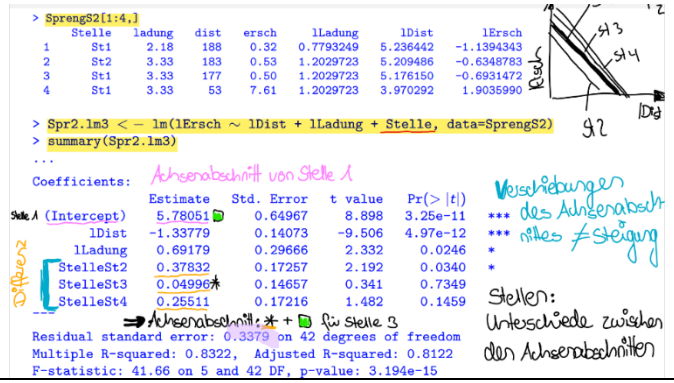
Nach kleinsten Quadrate Methode geschätzte Koeffizienten:
- Minimieren Quadratsumme der Residuen
- Maximieren die Korrelation zwischen den angepassten Werten und den Beobachtungen der Zielgrösse (maximale Wert = multiple Korrelation)

R-Code: Vertrauensintervall:
confint(SP.lm2, level=0.95)
confint(MPIZH.lm1, parm=2, level=0.95)
von Hand: 2.97 + c(-1, 1) * 0.36 * qt(0.99, 43)
aus Summary(): Estimate / Std. Error / F-Statistics (DF)

Level: 0.98 -> 0.02/2 + 0.98 = 0.99

Prognoseintervall:
x.MPI <- data.frame(HZ=4.5, KPI=100.5) -> Reihenfolge egal
predict(MPI.lm, newdata=x.MPI, interval="prediction", level=0.95)

!! Multiple Regression ist nicht gleich der Summe der einfachen Regression



$\log(ersch)_i = \beta_0 + \beta_1 \log(dist)_i + \beta_2 \log(ladung)_i + \beta_{3,1} St1_i + \beta_{3,2} St2_i + \beta_{3,3} St3_i + \beta_{3,4} St4_i + E_i$

$\hat{\sigma}$ -> sollte möglichst klein sein

Vielfalt der Modellierungsmöglichkeiten

Polynomiale Regression (Spezialfall von mult. lin. Regression)
Bsp.: Polynom 2. Ordnung wird zur Beschreibung eines Zusammenhangs verwendet: $Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + E_i$
→ Erklärende Variablen $x_i^{(1)}$ & $x_i^{(2)}$ einfügen
→ Lin. Reg.modell: $Y_i = \beta_0 + \beta_1 x_i^{(1)} + \beta_2 x_i^{(2)} + E_i$

«linear» im Begriff der multiplen linearen Regression bezieht sich darauf, dass die Koeffizienten linear in der Formel vorkommen!

Nichtlineare Funktionen und lineare Regression
Oft müssen Zielvariablen und / oder die erklärenden Variablen transformiert werden. -> «linearisierbaren Gleichungen» (Nach First Aid Transformationen)

$y = \frac{1}{a + b * \exp(-x)} \quad \leftrightarrow \quad \frac{1}{y} = a + b * \exp(-x)$

$y = \frac{a * x}{b + x} \quad \leftrightarrow \quad \frac{1}{y} = \frac{1}{a} + \frac{b}{a} * \frac{1}{x}$

$y = a * x^b \quad \leftrightarrow \quad \ln(y) = \ln(a) + b * \ln(x)$

$y = a * \exp\big(b * g(x)\big) \quad \leftrightarrow \quad \ln(y) = \ln(a) + b * g(x)$

Binär erklärende Variablen
Eine erklärende Variable kann binär sein, also auf die Werte 0 und 1 beschränkt sein.
→ Regressionsmodell beschreibt 2 unabhängige Stichproben (ungepaarter Zweiproben-t-Test)

$Y_i = \beta_0 + E_i \quad für\ x = 0$
$Y_i = \beta_0 + \beta_1 + E_i \quad für\ x = 1$

Beispiel: Allfälliger Unterschied der Lage / sind zwei Geraden gleich?
$Y_i = \beta_0 + \beta_1 x_i^{(1)} + \beta_2 x_i^{(2)} + E_i$
$x_i^{(1)} = log.Distanz$
$x_i^{(2)} = bin.Variabel\ (0\ oder\ 1)\ für\ Messstelle$

Modell umformulieren:
$Y_i = \alpha + \beta x_i + \Delta\alpha g_i + \Delta\beta g_i + E_i$
→ g_i = 0 falls Gruppe A / = 1 falls Gruppe B
Fallweise aufgeschrieben:
$g_i = 0: \ Y_i = \alpha + \beta x_i + E_i$
$g_i = 1: \ Y_i = (\alpha + \Delta\alpha) + (\beta + \Delta\beta) x_i + E_i$

→ Steigung stimmt überein, wenn $\Delta\beta = 0$
→ Geraden stimmen überein, wenn $\Delta\alpha = 0\ und\ \Delta\beta = 0$

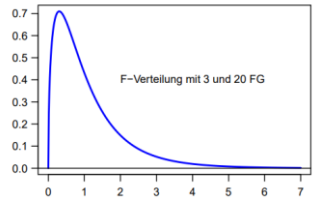
Vergleich von Regressionsmodellen mit F-Test
Einfluss von Koeffizienten: «Kein Einfluss»: alle Koeffizienten (Stellen) =0 (Nullhypothese b_{jq} =0)

$T = \frac{\frac{SS_E^* - SS_E}{q}}{\frac{SS_E}{n - p}}$

$SS_E^* = Quadratsumme\ des\ Fehlers\ im\ red.\ Modell$
$SS_E = Quadratsumme\ des\ Fehlers\ aus\ dem\ alternativen\ Modell$
q, n- p = Freiheitsgrade

R-Code:
anova(Spr2.lm3, Spr.2.lm2) 2 Modelle gegeneinander
drop1(Spr.lm3, test="F") -> ähnlich Summary, jedoch auch für Faktorvariablen, Test Variable wird weggelassen im gleichen Regressionsmodell (eignet sich nicht um signifikante Einflüsse auf Zielvariablen festzustellen, sobald Faktor Variablen vorkommen)

F-Test bei einfacher Regression:
Entspricht derselben Grösse wie t-Test zum Steigungskoeffizienten



Verteilung:
F-Verteilung (rechtsschief, unimodal)
Nur für positive Werte definiert, je schiefer verteilt je kleiner ist q

Globaler F-Test: In letzter Zeile R-Outputs (summary) ->immer die gleiche Antwort wie t-test

```
> anova(Spr2.lm3, Spr2.lm2)
Analysis of Variance Table

Model 1:  lErsch ~ lDist + lLadung + Stelle
Model 2:  lErsch ~ lDist + lLadung
  Res.Df    RSS  Df Sum of Sq      F  Pr(>F)
1     42 4.7964
2     45 5.5785  -3  -0.78205 2.2827 0.09297
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

"RSS"        = Summe der quadrierten Fehler (SS_E^2 oder SS_E)
"Sum of Sq"  = SS_E^2 − SS_E

> drop1(Spr.lm3, test="F")
Single term deletions

Model:
lErsch ~ lDist + lLadung + Stelle
         Df Sum of Sq    RSS     AIC F value    Pr(F)
<none>                4.7964 -98.560
lDist     1    10.321 15.1170 -45.458 90.3722 4.973e-12 ***
lLadung   1     0.621  5.4175 -94.716  5.4379  0.02457 *
Stelle    3     0.782  5.5785 -97.310  2.2827  0.09297 .

Beachten Sie, dass "F_{1,n−p} ∝ (t_{n−p})²" gleiche P-Werte
```

Modell und Schätzung in Matrix-Schreibweise

$Y_i = \beta_0 + (\beta_1 x_i)^{(1)} + (\beta_2 x_i)^{(2)} + \cdots + (\beta_m x_i)^{(m)} + E_i$

Als Vektor schreiben:

$\underline{Y} = \underline{X}\underline{\beta} + \underline{E}$

mit

$$\underline{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}, \quad \underline{E} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}, \quad \underline{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix} \text{ und } \boldsymbol{X} = \begin{bmatrix} 1 & x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(m)} \\ 1 & x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(m)} \\ \vdots & & \vdots & & \vdots \\ 1 & x_n^{(1)} & x_n^{(2)} & \dots & x_n^{(m)} \end{bmatrix}$$

➔ Das Prinzip der Maximalen Likelihood besteht darin, die Parameter so zu wählen, dass die Wahrscheinlichkeit (oder Dichte), die beobachtete Zielvariablenwerte zu erhalten, maximiert wird.
➔ ML & KQ führen zu gleichem Schätzer für β

Residuen: $\underline{R} = \underline{Y} - X\underline{\beta}$
Summe der Quadrate: $\underline{R}^T\underline{R}$
Erwartungswert: $X\underline{\beta}$
Varianz: $\sigma^2 I$
KQ: $\underline{\hat{\beta}} = (\underline{X}^T\underline{X})^{-1}\underline{X}^T\underline{Y}$

➔ Voraussetzung: (X^TX) invertierbar oder nicht singulär3

Zufallsvektoren – Relevant? Skript S. 64 - 70

Erwartungswert

$\mathbb{E}\langle\underline{Y}\rangle := [\mathbb{E}\langle Y^{(1)}\rangle, \mathbb{E}\langle Y^{(2)}\rangle, \dots, \mathbb{E}\langle Y^{(p)}\rangle]^T$

Varianz-Kovarianz-Matrix / Kovarianz-Matrix

$$\text{cov}\langle\underline{Y}\rangle = \text{var}\langle\underline{Y}\rangle := \begin{bmatrix} \text{var}\langle Y^{(1)}\rangle & \text{cov}\langle Y^{(1)}, Y^{(2)}\rangle & \dots & \text{cov}\langle Y^{(1)}, Y^{(p)}\rangle \\ \text{cov}\langle Y^{(2)}, Y^{(1)}\rangle & \text{var}\langle Y^{(2)}\rangle & \dots & \text{cov}\langle Y^{(2)}, Y^{(p)}\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}\langle Y^{(p)}, Y^{(1)}\rangle & \text{cov}\langle Y^{(p)}, Y^{(2)}\rangle & \dots & \text{var}\langle Y^{(p)}\rangle \end{bmatrix}$$

Zusammengefasst:

$$\text{var}\langle \underline{a} + \boldsymbol{B}\underline{Y}\rangle = \boldsymbol{B} \cdot \text{var}\langle\underline{Y}\rangle \cdot \boldsymbol{B}^T = \boldsymbol{B}\,\boldsymbol{\Sigma}\,\boldsymbol{B}^T.$$

Das gilt auch für eindimensionale Z; die Matrix $\boldsymbol{B}$ besteht dann aus einer einzigen Zeile, die wir als transponierten Vektor $\underline{b}^T$ aufschreiben. So wird

$$\text{var}\langle a + \underline{b}^T\underline{Y}\rangle = \underline{b}^T\text{var}\langle\underline{Y}\rangle\,\underline{b} = \underline{b}^T\boldsymbol{\Sigma}\,\underline{b},$$

ein Resultat, das oft nützlich ist.

Für die Zufallsvariable Y gilt
- $\text{var}\langle Y\rangle = \mathbb{E}\langle(Y-\mu)^2\rangle$
- $\text{cov}\langle Y^{(i)}, Y^{(k)}\rangle = \mathbb{E}\langle(Y^{(i)}-\mu^{(i)})\,(Y^{(k)}-\mu^{(k)})\rangle$

Für den Zufallsvektor $\underline{Y}$ gilt analog
- $\text{var}\langle\underline{Y}\rangle = \mathbb{E}\left\langle(\underline{Y}-\underline{\mu})\,(\underline{Y}-\underline{\mu})^T\right\rangle$

Für die Zufallsvariable Y gilt bei **lineare Transformationen**
- $\mathbb{E}\langle a + bY\rangle = a + b\,\mathbb{E}\langle Y\rangle$
- $\text{var}\langle a + bY\rangle = b^2\text{var}\langle Y\rangle$.

Für den Zufallsvektor $\underline{Y}$ gilt analog
- $\mathbb{E}\langle\underline{a} + \boldsymbol{B}\underline{Y}\rangle = \underline{a} + \boldsymbol{B}\,\mathbb{E}\langle\underline{Y}\rangle$
- $\text{var}\langle\underline{a} + \boldsymbol{B}\underline{Y}\rangle = \boldsymbol{B}\cdot\text{var}\langle\underline{Y}\rangle\cdot\boldsymbol{B}^T \overset{kurz}{=} \boldsymbol{B}\boldsymbol{\Sigma}_{\underline{Y}}\boldsymbol{B}^T$
- und falls $\boldsymbol{B}$ nur ein Zeilenvektor ist: $\text{var}\langle a + \underline{b}^T\underline{Y}\rangle = \underline{b}^T\text{var}\langle\underline{Y}\rangle\,\underline{b} = \underline{b}^T\boldsymbol{\Sigma}_{\underline{Y}}\,\underline{b}$

Sensitivität und Robustheit

Theoretische Verteilung der Residuen

Da die Nebendiagonalelemente in $\text{var}\langle\underline{R}\rangle = (\boldsymbol{I} - \boldsymbol{H})\sigma^2$ im Allgemeinen nicht null sind, sind die Residuen im Gegensatz zu den Fehlern E_i *korreliert*,

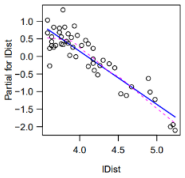
$$\text{cov}\langle R_i, R_k\rangle = -\sigma^2 H_{ik}.$$

Diese Korrelation zwischen den Residuen beeinflusst jedoch die Strukturen in der grafischen Residuen-Analyse kaum. Auch nicht jene, wo wir den zeitlichen oder räumlichen Korrelationsstrukturen (stochastische Abhängigkeit, siehe Abschnitt 3.4) nachgingen.

Residuen-Analyse bei der multiplen Regression

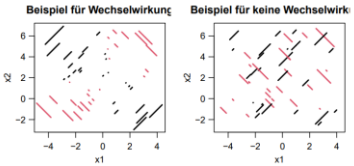
1. Grafiken analysieren. Allenfalls zeitliche Abhängigkeiten klären.
2. Fazit – Welches ist die relevanteste Unstimmigkeit und wie kann sie bereinigt werden?

Überprüfung Abhäng. Streuung:



Beziehung zwischen Residuen und erkl. Variablen

Überprüfung Wechselwirkung:
Voraussetzung: Effekte 2 erkl. Variablen addieren sich



Gewichtete lineare Regression

Annahme, dass Varianzen der Residuen nicht konstant

$\sigma_i^2 = var(E_i) \rightarrow Lösungsansatz: \sigma_i^2 = \frac{\sigma^2}{w_i}$

➔ Mittelwert streut weniger je grösser die Stichprobe -> w_i = Anzahl Stichproben
➔ Generell: Für Beurteilung von Verteilung und Streuung der Fehler skalierte Residuen verwenden!!

Weitere Überprüfungsmethoden Residuenanalyse:

Schauen, ob Streuung um Winkelhalbierende:
```
par(mfrow=c(3,3))
plot(fitted(asp.lm), log(asp$RUT));
abline(a=0,b=1, lty=2)
```

fittes vs resid / Erkl. Variablen vs resid
```
plot(fitted(asp.lm), resid(asp.lm), col=asp$RUN+2)
plot(Ox$TS, resid(Ox.lm1)); abline(h=0, lty=3)
plot(Ox$Day, resid(Ox.lm1), type="h") Zeit Analyse
```

Residuen vs erkl. Var. (um horizontale Null-Linie):
```
plot(log(asp[,'VISC']), resid(asp.lm), col=asp$RUN+2)
```

R-Code:

In multiplen Regression -> Plot pro erkl. Variable erstellen:
```
scatter.smooth(dat$lLadung, resid(fit) , lpars = list(col = 2)); abline(h = 0))
```

 # Partieller-Residuen-Plot:
```
termplot(Spr2.lm2, partial.resid=TRUE,
smooth=panel.smooth, ylim="free",
col.res="black", col.term="blue",
col.smth="magenta")
```

 # Wechselwirkungs-Diagramm
Package sfsmisc
```
p.res.2x(x=SprengS2$lDist, y=SprengS2$lLadung,
z=resid(Spr2.lm2), xlab="log(Distanz)",
ylab="log(Ladung)", scol=c(1,1), size=1.2, slwd=2,
main="")
```

R-Code:

Einzeichnen in Plot (rote Line, skaliert):
```
G <- data.frame(SPEED=GASKETS$SPEED,  RESID=resid(G.lm1))
scatter.smooth(G$SPEED,  sqrt(abs(G$RESID)), span=1)
```

Gewichtung ermitteln:
```
(G.varRes <- aggregate(RESID ~ SPEED, data=G, FUN=var))
# Gegenüberstellung der beiden Variablen

G.varRes[,2]/c(100,150,200) # (100,150, 200) = Variable Speed, Resultat noch nicht konstant, somit weiter ausprobieren:
G.varRes[,2]/c(100,150,200)^2 # +/- konstant, jedoch sehr klein
G.varRes[,2]/(c(100,150,200)/100)^2  # auch +/- konstant, wenn pro 100 Einheiten Produktionsgeschwindigkeit?
```
-> verwende als Gewicht 1/(SPEED/100)^2

Fitter gewichten:
```
GASKETS$w <- 1/((GASKETS$SPEED/100)^2)
```
➢ **G.lm2 <- lm(DEFECTS~SPEED, weights=w, data=GASKETS)**

Residual standard error: 4.219 -> bei Gewicht = 1

Einflussreiche Beobachtungen

Cook's Distance (Ausreisser-Analyse)

$\widetilde{R}_i = \frac{Ri}{\widehat{\sigma}\sqrt{1-H_{ii}}}$

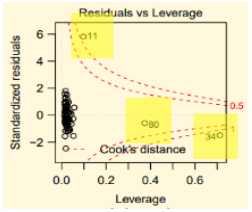
Gibt es eine zu einflussreiche Beobachtung?

Wertebereich für H_{ii} $0 \le H_{ii} \le 1$ -> Funktioniert wie Hebelarm, misst wie untypisch die Beobachtung auf die erklärenden Variablen ist

- Beobachtungen mit einem Hebelarm $> 2\,\frac{p}{n}$ haben zu grosse Hebelwirkung
- Beobachtungen mit einem Hebelarm unter 0.2 sind unbedenklich
- Beobachtungen mit einem Hebelarm über 0.5 sollten vermieden werden

Diagramm

Standardisierte Residuen gegen H_{ii}



11: grosser Ausreisser in y-Richtung

34: zu grosser Hebelarm ($h_i > 0.2$) (siehe Leverage)

80: Grosser Hebel, jedoch kleines standardisiertes Residum daher ungefährlich

«bösartige Beobachtung»

«An Grenze zu einflussreicher Beobachtung»

«Hebelpunkte»

Robuste Anpassungsmethoden

Untersuchung Robustheit:

- Einflussfunktion
- Bruchpunkt

➔ Robuster Schätzer hat beschränkte Sensitivität und Bruchpunkt, der möglichst nahe beim maximal möglichen Wert von 1/2 liegt

Regressions-M-Schätzer

Knick in der Funktion legt fest, ab wo extreme Beobachtungen an Einfluss verlieren

Schätzer für lineares Regressionsmodell:

- Beobachtungen mit grossen Residuen müssen ignoriert werden

R-Code:

Robuste M-Anpassung:

```
Mfit <- rlm(y ~ x, method = "M", data=AQ)
```

Regressions-MM-Schätzer: (Package: robustbase)

```
lmrob(y ~ x1 + x2, data=D.syn, setting="KS2014")
```

besonders effiziente Vertrauensintervalle: setting = "KS2014"

```
par(mfrow= c(2,3))
plot(lmrob)
```

Robuster F-Test

Beruht auf robusten Devianz

Summen der Quadrate im F-Test durch Devianzen ersetzten:

```
lm1 < – lmrob(FoHF ~ ., data=FoHF2, setting="KS2014")
anova(lm1, FoHF ~ LSE + GM + EM + SS, test="Deviance")
```

Variablenselektion und Modellbildung

Welche Variablen sollen wie ins Regressionsmodell?

Kriterien basierte Variablenselektion ist vorzuziehen.

Warum nicht alle Variablen?

- Einfachheit
- Reduktion Schätzvariabilität (unnötige Variablen verschlechtern die Genauigkeit der Schätzung)
- Bessere Interpretierbarkeit (Multikollinearität führt nicht zu eindeutigen Lösungen)
- Vorhersage – weniger erkl. Variablen, weniger Aufwand beim Vor- und Aufbereiten

Modellwahlkriterien

- Modellgenauigkeit / Modellkomplexität

➔ Sind konträr – Zusätzliche erkl. Variablen führen stets zu höherer Modellgenauigkeit

Lineare Regression mit KQ-Anpassung:

- Modellgenauigkeit mit der Summe der Residuenquadrate
- Modellkomplexität mit der Anzahl Koeffizienten

Bestimmtheistmass R^2 korrigieren (adjusted R^2)

R^2 wird grösser je mehr Variablen hinzugefügt werden -> untauglich (Komplexität nicht berücksichtigt)

Korrigiertes Bestimmtheitsmass:

$$R^2_{\text{adj}} := 1 - \frac{SS_E/(n-p)}{SS_Y/(n-1)}$$

Mallow's Cp-Statistik:

Minimiert in gewisser Weise den Vorhersagefehler

$$C_p := \frac{SS_E}{\widehat{\sigma}^2_{p^*}} + 2p - n = (n-p)\left(\frac{\widehat{\sigma}^2_p}{\widehat{\sigma}^2_{p^*}} - 1\right) + p$$

Informations-Kriterium von Akaike (AIC)

Gutes verallgem. Kriterium (auch Zeitreihen)

$$\text{AIC} = -2\,(\text{maximierte Log-Likelihood}) + 2\cdot(\text{Anzahl geschätzter Parameter})$$
$$= n\log\left\langle\frac{1}{n}SS_E\right\rangle + 2p^\circ + \text{Konstante}$$

wobei SS_E die Summe der quadrierten Residuen ist.
(Achtung: p° ist hier die Anzahl der geschätzten Parametern inklusive $\widehat{\sigma}$.)

Variablenselektion mit P-Werten

Ist ein bestimmter Term im Modell nötig / nützlich / überflüssig?

Problem des multiplen Testens:

➔ Bei Faktorvariablen den t-Test durch den F-Test ersetzen (prüft, ob der ganze Block von der entspr. Variabel weggelassen werden kann)

Vorwärts-Selektion (auf P-Wert gestütztes Kriterium)

1. Modell wählen: $Y_i = \beta_0 + E_i$
2. Jeweils Variabel, welche den kleinsten P-Wert besitzt ins Modell aufnehmen.
3. Stopp: Sobald keine Verbesserung mehr möglich

Rückwärts-Selektion

1. Mit dem «vollen» Modell starten
2. Jeweils eine Variabel aus dem Modell entfernen (Beginnen mit Variabel, welche am unwichtigsten)
3. Stopp: Sobald keine Verbesserung mehr möglich

Schrittweise Selektion:

- Kombination aus Vorwärts- und Rückwärtselimination, lokales optimales Modell
- Stopp: Sobald keine Verbesserung mehr möglich

Vollständige Modellselektion (all-subset selection)

Global optimierte Kriterien, berechnet für alle möglichen Modellvarianten

2^m mögliche lineare Modell-Gleichungen, bei grossem m ist Rechenaufwand zu gross

All-Subset-Verfahren mit C_p-Kriterium:

Wenn das Modell alle notwendigen Variablen enthält, Kriteriumwert = $E(C_p) = p$

Fehlen notwendige Variablen: $E(C_p) > p$

Allgemein

➔ Verfahren führen nicht alle zum selben Modell

➔ P-Werte keine Aussagen über Modellgenauigkeit / Komplexität

➔ Schranke von 5 % im Stoppkriterium ist willkürlich

Anmerkungen:

- Empfehlung: All-Subset Selection
- Schrittweise Selektion: Grosses Modell
- Vorwärts Selektion: Datensätze mit vielen erklärenden Variablen und wenig Beobachtungen

Faktorvariablen/Polynomen/Wechselwirkungstermen:

- Faktorvariablen: Faktorstufen dürfen nicht einzeln entfernt werden (nur als ganze Variable)
- Wechselw.terme: immer mit Haupteffekt
- Polynomen: alle Terme bis zur maximalen Ordnung behalten, niedrigen Terme nicht entf.

➔ R-Funktion regsubsets(…) berücksichtigt dies nicht

R-Code:

Vorwärts-Selektion:

```
AKW.lm <- lm( Y ~ 1, data = ....)
add1( AKW.lm, scope = ~ lgG + D +  BW + sqrtN + KG, test="F")
AKW.lm1 <- update(AKW.lm0, . ~ . + KG + lgG)
# Kleinster p-Wert schrittweise hinzufügen
```

Rückwärts-Selektion:

```
AKW.lm0 <- lm(Y ~ ., data = ...)
drop1(AKW.lm0, test="F")
AKW.lm1 <- update(AKW.lm0, . ~ . – WZ - BW) # Grösster P-Wert schrittweise entfernen
```

Selektion basierend auf AIC (step(…):

1. Vorwärtsselektion:

```
M0 <- lm(Y ~ 1, data=DF)
step(M0, scope=list(lower= ~ 1, upper= ~ x1 + xn), direction="forward")   hintere Teil von lm( ~ ....)
```

2. Rückwärts-Elimination:

```
MF <- lm(Y ~ x1 + xn, data=DF) oder lm(Y ~ . , ...)
step(MF, direction="backward")
```

3. Schrittweise Selektion:

```
Mm <- lm(Y ~ x2, data=DF)  kann auch ~ 1 sein
X <- step(Mm, scope=list(lower= ~ 1, upper= ~ x1 + xn), direction="both") # scope: immer kleinstes & grösstes Modell angeben
X$anova # Zusammenfassung über Entfernung / Hinzufügen
```

All-Subset-Verfahren mit C_p-Kriterium:

```
library(leaps); library(car)
AKW.Cp <- regsubsets(lK ~ lG + BW + wN + KG, nbest=6, nvmax=10, data=AKW) # nbest immer angeben
summary(AKW.Cp) # Übersicht
h <- subsets(AKW.Cp, statistic="cp", legend="interactive", min.size=4, main="Mallow Cp", cex.subsets=0.7, las=1) # statistic = "cp", "bic", "adjr2", usw.
abline(a=1, b=1, lty=2) # Intercept wird in Subset size nicht mitgezählt -> + eine Einheit
plot(AKW$Cp, scale="Cp") # Plot mit schwarzen Blöcken
```

Bemerkungen:

- Argument nbest: Anzahl Modell mit p Koeffizienten, die berucksichtigt werden. (default = 1)
- Argument nvmax: Modelle mit maximal p = nvmax werden berucksichtigt; Anzahl Modell mit p Koeffizienten, die berucksichtigt werden. (default = 8)
- Die Funktion subsets() zählt den Achsenabschnitt nicht mit, deshalb ist abline(a=1, b=1) die Winkelhalbierende.

➔ Die in Frage kommenden Modelle, müssen um die Winkelhalbierende $C_p = p$ streuen

➔ Potenzielle Modelle: minimaler C_p-Wert und / oder minimales p (kleinste Anzahl Parameter)

Genügt nicht immer, weil:
- Auswahl der Variablen zufällig (mehrere Modelle in Betracht ziehen)
- Beste Modell muss nicht unbedingt alle Voraussetzungen erfüllen (Residuenanalyse)
- Aufgrund übergrosser Anzahl Möglichkeiten: Nicht alle potentiell nützlichen Transformationen können berücksichtig werden

Welches Modell ist das richtige?
Werte der Modellwahlkriterien sind mit Unsicherheit behaftet – daher kann das beste Modell nur zufälligerweise das Richtige sein.
➔ Mehrere Modelle in Betracht ziehen

Bayes Informationskriterium (BIC, Alternative zu AIC) Bestraft grössere Modelle härter
- BIC: welche Prädiktoren leisten einen Beitrag, schlankes gut interpretierbares Modell
- AIC: Modell wird für die Vorhersage von zukünftigen Werten eingesetzt, Prädiktoren sind weniger zentral

$$BIC = n \log\left\langle \frac{1}{n} SS_R \right\rangle + \log\langle n \rangle \; p^{\circ} + \text{Konstante}$$

```
step(..., k=log(nrow(dataset)))
```

Alternative Ansätze für Variablenselektion:
PRESS
Summe der quadrierten Differenzen zwischen beobachteten und vorhergesagten Zielwerten.
- «leave-one-out» Ansatz (spez. Kreuzvalidierung)
- Mass für Vorhersagegüte
- Grosser Rechenaufwand bei grossen Modellen

$$R^2_{pred} := 1 - \frac{\text{PRESS}}{SS_Y}$$

Kein Overfitting
kann negative Werte annehmen

LASSO
Variablen selektieren (nur einflussreiche)
- Shrinkage Schätzung/ Regularisierungsmethode (Schrumpft Koeffizienten in Richtung 0)
- Stellt sicher, dass kein Overfitting vorliegt (keine unsicheren Vorhersagen, da zu gut ans Modell angepasst)

R-Code:
```
library(lars)
h.x <- model.matrix( IG ~ D + WZ + BZ, data=AKW)
AKW.lasso <- lars(x=h.x, y=AKW$IK)
plot(AKW.lasso) # Bedeut. Koef.: 1. der auf 0 (Plot)
```
Optimal: 5 – 10 fache **Kreuzvalidierung** (1/5 der Daten wird weggelassen), auch für hochdimensionale Daten geeignet:
```
AKW.lasso.cv <- cv.lars(x=h.x, y=AKW$IK, K = 10)
min <- which.min(mf.lassocv$cv) # Plot siehe S. 6
```

Adaptive LASSO: (mit Gewichtung)
```
library(lasso2)
t.r <- l1ce(K ~ .,data=t.d, bound=seq(0.05,1,0.05))
plot(t.r)
summary(t.r[[5]])
```
Mit folgenden Befehlen auf gleichnamigen Paketen ebenfalls möglich: lars, glmnet, sealasso

Kollinearität
Hohe Korrelationen zwischen erkl. Variablen zugelassen, führen zu Problemen bei Interpretation & Modellierung.

Erklärende Variabel lässt sich bei Kollinearität annähernd als Linearkombination der anderen darstellen.
Gilt die Beziehung exakt, gibt es keine eindeutige Lösung bei der Kleinste-Quadrate Schätzung.

Konditionszahl
$K = \frac{\lambda_{max}}{\lambda_{\min}}$ (auch für Entdeckung von Multikollinearitäten geeignet)
Maximaler und minimaler Eigenwert von X^TX
Zahl 100 – 1'000: Moderate bis starke Multikollinearität
Zahl > 1'000: Multikollinearität schwerwiegend

Variance inflation factor (VIF)
Bestimmtheitsmass zeigt:
- Wie stark eine solche Beziehung ist
- Ist also ein sinnvolles Mass für Kollinearität
- Gibt an, welche Variable das Problem verursacht

$VIF_j = \frac{1}{(1-R_j^2)}$ *Nur für num. e rkl. Variablen definiert*

Faustregel: Falls > 5-10, Probleme mit Kollinearität

R-Code:
```
library(car)
round(vif(AKW.lm),2)
```

Auswirkungen der Kollinearität
- Hohe Kollinearität führt zu grossen Standardfehlern bei geschätzten Koeffizienten
- ➔ Mit $\sqrt{VIF_j}$ aufgeblasen (Optimal Fakor 1)
- Viele / alle Variablen gem. t-Test nicht signifikant
- Gewisse Richtungen: sehr kleine oder grosse Prognosefehler -> Progn.intervalle bestimmen
- Interpretation Effekte der einzelnen Variablen auf Zielvariabel nicht möglich

Kollinearität muss bereinigt werden

Was tun gegen Kollinearität?
Wenn immer möglich, soll man Beobachtungen so durchführen, dass das Problem vermieden wird. Ansonsten:
- Variablen linear transformieren; d.h. z.B. stark korrelierte Variablen ersetzt man z.B. durch ihre Summe und ihre Differenz
- Weitere Möglichkeiten:
- 0.5*(Jet$x1 + Jet$x2); Jet$dif <- Jet$x1 - Jet$x2
- Relation: seatpos$rSeated <- seatpos$Seated / seatpos$Ht; seatpos$rArm <- seatpos$Arm / seatpos$Ht
- Variable mit dem höchsten VIF aus dem Modell entfernen (= «Aputation») -> Korrelationen treten fast zwingend auf, wenn Vergleich zur Anzahl Beob. viele erkl. Var. vorhanden sind.
- Setze so genannte Shrinkage-Schätzer ein wie z.B.
- Hauptkomponentenregression (principal component regression), ridge regression oder LASSO, elastic net. Solche Schätzer haben sich vorteilhaft bei Prognosemodellen erwiesen. Allerdings leidet Interpretierbarkeit stark.

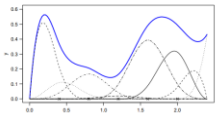
Additive Modelle – nicht lineare Funktion

Was wenn Funktion h nicht linear in den Parametern β ist?
Anpassung durch:
- Nichtlineare Regression, dies muss jedoch von Fachwissen motiviert werden (also nicht relevant)
- Nichtparametrische Schätzung des funktionalen Zusammenhangs (Glätter)

Modelle der Form: $Y_i = h(x_i, \beta) + E_i$

Spline-Interpolation
- Stückweise Polynome der Ordnung k
- Verbindungspunkte der Stücke = Knoten
- Kubische Spline (k=3) ausreichend
- $g\langle x\rangle = \beta_1 * \beta_1^{(2)}\langle x\rangle + \beta_2 * \beta_2^{(2)}\langle x\rangle + \ldots + \beta_q * \beta_q^{(2)}\langle x\rangle$
- Sinnvoller Daten zu glätten als zu interpolieren
- Glattheit der Kurve wird über die Wahl der Anzahl Stützpunkte q bestimmt
  q klein: Kurve sehr glatt, q gross: eher Interpolation

Die B-Spline-Basisfunktionen (dünne Kurven) für eine Spline-Kurve (blaue fettere Kurve) mit 5 inneren Stützstellen (× auf der Nulllinie).



Die fünf B-Spline-Basisfunktionen sind jeweils mit dem Parameterwert β_j multipliziert, sodass ihre Summe gerade die Spline-Kurve ergibt.

Smoothing Splines
Wie soll q (Glattheit der Spline-Kurve) gewählt werden?
- Zielkonflikt: Modellgenauigkeit / Modellglattheit wird durch λ geregelt:
  λ *gross*: Gerade / glatteste Funktion
  $\lambda = 0$: unterglättete Smooting-Spline-Kurve
- Optimale Glattheit: optimale Wahl λ
- Kreuzvalidierungsverfahren zur Wahl λ

Thin plate regression splines
- Knotenfreien Basisfunktionen
- Mehrere erkl. Variablen zugleich glätten
- optimal

Lokale Regression (LOWESS / LOESS)
Jede Funktion h in einer kleinen Umgebung von einem vorgegebenen Punkt linear approximieren. (in Fenster aufteilen)

Umsetzung: Gewichtete KQ-Methode
Zu bestimmende Punkte:
- Wahl der Fensterbreite b
  b klein: Approximationsfehler sehr klein, grosse Varianz der Vorhersage
  b gross: kleine Varianz, lineare Approximation kann ungenügend sein
  -> vorgegebene Anzahl d Punkte im Fenster:
  d = 2/3 * Anzahl Punkte n
- Wahl der Gewichtsfunktion K(u)
  So wählen, dass z.B. Berechnung vereinfacht wird oder Umsetzung mit LOWESS / LOESS: Tukeys Gewichtfunkt.
  - Kontrollieren den Einfluss von schlechten Beobachtungen
  - c klein: desto schneller verlieren extreme Beobachtungen an Einfluss
  -> Verlust von Effizienz, grosse Standardab.
  - Cleveland: Empfehlung c = 4.05
- Behandlung von Ausreissern (in y-Richtung)

R-Code: Splines
Regression-Splines: library(splines)
```
lm(y ~ bs(x, df=...), data=dat)
plot(NOx ~ Equiv, data=exhaust, las=1,
xlab="Äquivalenzverhältnis", ylab="Stickoxidkonz.",
mgp=c(2,0.9,0), cex=0.7)
h.range <- range(exhaust$Equiv)
h.knots4 <- seq(h.range[1], h.range[2], length=6)[-c(1,6)] # 4 Knoten, da 1 und 6 weggelassen werden
bs(x=exhaust$Equiv, knots=h.knots4, degree=3)
# generiert Matrix mit kubischen Spline-Basen
```

Plot kann auch analog Polynomiale Regression erstellt werden!

Polynomiale Regression
```
MU.p3 <- lm(accel ~ poly(times,3), data=mcycle)
hx <- data.frame(times=seq(min(mcycle$times),
max(mcycle$times), length=200))
plot(accel ~ times, data=mcycle, las=1, main="Regression")
lines(hx$times, predict(MU.p3, newdata=hx), col="red")
legend(40, -100, paste("Ordnung =", c(3,6,12)), lty=rep(1,3),
col=c("red", "blue", "green"))
```

Smooting Splines: smooth.spline(x,y, cv=TRUE)
```
smooth.spline(data$x , data$y , df=7) # ohne Angabe df: optimale Anzahl FG
smooth.spline(x=jitter(exhaust$Equiv), y=exhaust$NOx,
cv=TRUE)
```

Kubische Splines: interpSpline(datx, daty)

Thin plate regression splines: gam(y ~ s(x), data=dat)
```
library(mgcv)
gam(Zielvariable ~ s(x1) + s(x2) + ...+ s(xm), ...)
# Identisch mit lowes() bei nur einer erkl. Variabel
plot.gam()
par(mfrow=c(2,2)) # für partielle Residuenplots
plot(P.gam, se=TRUE)
plot(P.gam, se=TRUE, residuals=TRUE)
ODER:
library(gam)
LB.gam2 <- gam(lBurntime ~ lo(lNitrogen) + lo(lChlorine) +
lo(lPotassium), data=LB) # Variablen vorher logarithmieren
```

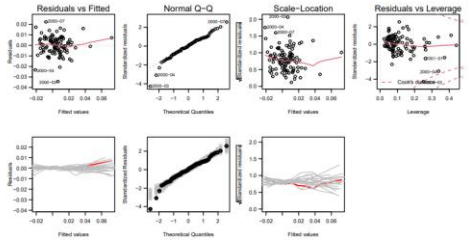
Lokale Regression (LOWESS / LOESS):
```
loess(y ~ x, data=dat, degree=1, span=...)
# mit family="symmetric" für robuste Anpassung,
"gaussian" für Gauss / 1 = lokale linear, 2 = lokal
quadratisch / Fensterbreite 20 % = span 0.2 -> je höher
desto glätter

plot(exhaust$Equiv, exhaust$NOx, mgp=c(2,0.9,0), cex=0.7,
xlab="Aequivalenzverhaeltnis",
ylab="Stickoxidkonzentration")
exhaust.loess <- loess(NOx ~ Equiv, data=exhaust,
span=0.3, degree=2, family="symmetric")
# Beobachtungen d = span
xnew <- seq(0.5, 1.3, length=100)
lines(xnew, predict(exhaust.loess, xnew), col="magenta")

lowess(x1, Zielvariable)
lo(..., span=0.5, degree=1) # LOWESS Glätter
stl( . . . )
```

Zusammenfassung STMO

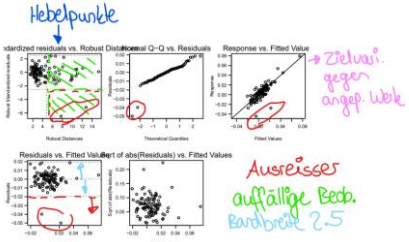
Plots mit klassischen Methoden



Abweichungen nicht mehr normal:

Varianz > als * Faktor 4
Standardabweichung > als * Faktor 2

Plots mit robusten Anpassungsmethoden



Vorgehen Modellentwicklung

(i) Problem (d.h. Auftag, Ziel, Zweck) verstehen; gibt es schon Modellansätze?
(ii) Daten beschaffen, kennenlernen und aufbereiten
- Codieren von fehlenden Werten abklären (Achtung: Manchmal sind diese mit '-99' oder mit '9999' codiert.)
  Umgang mit fehlenden Werten in der Analyse festlegen.
- Bedeutung der Zahl 0 in den verschiedenen Variablen vereinheitlichen.
- Datenqualität beim Zusammenführen mehrerer Datensätze hinterfragen. Die Gesamtqualität ist nie höher als das schwächste Glied.
- Daten gemäss first-aid Transformationen behandeln, ausser es gibt trifftige Gründe dagegen (z.B. bestehendes Modell).

(iii) Erste Anpassung; vorzugsweise mit robusten Methoden
(iv) Residuen-Analyse;
- Tragen die Daten dazu bei, das Problem zu lösen?

ev. zurück nach (ii) oder (i)
(v) Variablenselektion, allenfalls Kollinearitäten behandeln
(vi) Modelleignung klären
- Residuen-Analyse mit selektionierten Modellen
- Modelle mit Fachwissen abgleichen, falls mit dem Modell Zusammenhänge beschrieben und erklärt werden sollen
- 'out-off-sample' Validierung in Betracht ziehen (d.h. Validierung mit noch nicht verwendeten Daten), vor allem wenn das Modell zur Prognose eingesetzt werden soll.

R-Code Plots (Allgmein)

Plot mit abline und eingezeichnetem Vertauensintervall

```
windows(8,4)                    separates Fenster öffnen
identify(x, y)   Punkt in Grafik auswählen (Gibt Nr zurück)

plot(Forbes$x, Forbes$y)
points(Forbes$x[12], Forbes$y[12], col="red", pch=16)     die 12. Beobachtung blau einfärben
abline(ForbesR.lm, col="blue", lty=1)   Kleinste Quadrate Schätzer als Linie einfügen

abline(v=325.81, lty=4, col="red")  Linie bei vorgegebenem Punkt eintragen

x0 <- data.frame(x= seq(min(Forbes$x), max(Forbes$x), length=50))
Data Frame
ForbesR.cia <- predict(ForbesR.lm, newdata=x0,    Vertrauensintervall
                       interval="confidence", level=0.99)
lines(x0$x, ForbesR.cia[,"upr"], col="red")       Vertrauensintervall in Plot einzeichnen
lines(x0$x, ForbesR.cia[,"lwr"], col="red")
```

Plot mit Bestimmtheitsmass (R2) eingezeichnet

```
plot(fitted(MPI.lm), MPI$MPI)
abline(a=0, b=1, lty=2, col="red")
h <- predict(MPI.lm, newdata=x.MPI, interval="prediction", level=0.95)
points(h[,"fit"], h[,"fit"], col="red", pch=16)
```
-> wenn keine Punkte in der Nähe -> Voraussetzungen (gegeben) nicht erfüllt

Streudiagramm für mehrere Variablen (verwirrender Plot aus EXPD, Korrelationen zwischen Variablen)

```
pairs(cat[,3:1])
```

scatter3d-Plot

```
library(car)
scatter3d(y ~ x1 + x2, data = syn, axis.scales=FALSE)
```

Zufallszahlen generieren:

```
for(i in 1:6) {x <-rnorm(10); qqnorm(x);qqline(x, col="grey", lty=2)}
                    normalverteilt
for(i in 1:3) {x <- rt(1000, df=3); qqnorm(x);qqline(x, col="grey", lty=2)}
                    t-verteilt, Freiheitsgrade
x <- rchisq(20, df=7)          chiquadrat-verteilt
qqnorm(x, main="n=20, df=7"); qqline(x, col="grey", lty=2)
```

Differenz zwischen Prognose- / Vertrauensintervall «upr» & «lwr»

```
x <- predict(lm(y ~ x2, catheter), interval="prediction")   Prognose- / Vertrauensintervall bilden
round(cbind(pi.unten=x[,"lwr"], fit=x[,"fit"], pi.oben=x[,"upr"],
      pi.laenge=x[,"upr"]-x[,"lwr"]), 1)   Runden und in Tabelle Differenz berechnen
```
-> übersteigt Differenz das erwartete Intervall: Modell muss verbessert werden (hinzufügen von weiteren erkl. Variablen)

Intervall mit Rücktransformierung

```
predict(FourCm.lm1, newdata=FourCm.new, interval="confidence")
exp(predict(FourCm.lm1, newdata=FourCm.new,
interval="confidence")[,c("lwr", "upr")])
```
PI in der Regel geeigneter -> da stochastische Variabilität des Fehlers Ei berücksichtigt wird und nicht nur die Ungenauigkeit, welche sich aus der Schätzung des Modells ergibt

Sortieren / Filtern etc

```
farm1 <- farm[farm$region==121,]          nur die mit Region = 121
farm1 <- farm1[,-1]          Region (1. Spalte) entfernen
farm1$industry <- as.factor(farm1$industry)   in Faktorvariable umwandeln
is.factor(farm1$industry)    Prüfen ob Faktorvariable
str(farm1)                   summary(farm1)
dummy.coef(farm1.fit1)       Alle geschätzten und festgelegten Koeffizientwerte
table(farm$region)           Sortieren nach Region
```

Plots AB 4

```
plot(MPIZH$KPI, resid(MPIZH.lm1))          KPI gegen Residuen
plot(MPIZH$HZ, resid(MPIZH.lm1))           HZ gegen Residuen
termplot(MPIZH.lm1, partial.resid=T, smooth=panel.smooth)

plot(resid(MPIZH.lm1), type="h")
x.n <- nrow(MPIZH)
scatter.smooth(resid(MPIZH.lm1)[1:(x.n-1)], resid(MPIZH.lm1)[2:x.n])

plot(MPIZH$KPI, resid(MPIZH.lm1)); abline(v=100, col="blue", lty=3)
                    Linie bei 100 - (Residuen / Index-Plot)
plot(MPIZH$KPI, type="l"); abline(h=100, col="blue", lty=3)
                    Linie bei 100 – Index-Plot

load(paste(DPfad,"GASKETS.Rdata",sep=""))          Rdata Dokument einlesen
```

Optimalen Bestimmung des tuning Parameters $\tilde{\beta}$

```
par(mar=c(4,4,2,3), mgp=c(2.5, 0.8, 0))
set.seed(4567)
mfm.lasso.cv <- cv.lars(x=h.x, y=CEDHEC$FoHF, K = 10)
(h.wMin <- which.min(mfm.lasso.cv$cv)) # = 98
h.sel <- which(mfm.lasso.cv$cv <=  (mfm.lasso.cv$cv[h.wMin] +
           mfm.lasso.cv$cv.error[h.wMin]))[1]
h.sel
abline(v=mfm.lasso.cv$index[c(h.wMin, h.sel)],
col=c("magenta","blue"))
abline(h=mfm.lasso.cv$cv[h.wMin] + mfm.lasso.cv$cv.error[h.wMin],
col="blue")

c(mfm.lasso.cv$index[h.sel], mfm.lasso.cv$cv[h.sel])
```

Koeffizientenschätzung an Stelle optimalen b: coef(AKW.lasso, s=AKW.lasso.cv$index[h.sel], mode="fraction")