Beispiel 5.14: Multiple Regressionsmodelle für Staubdaten

Der SPSS-File `Staubdaten_GM_0506.sav` enthält Daten der Messstation Graz–Mitte vom 1. Oktober 2005 bis 31. März 2006. Im File sind insgesamt *18 Merkmale* abgespeichert.

- 5 kategorische Merkmale:
  - `nr` (1 bis 182)
  - `datum` (1.10.05 bis 31.03.06)
  - `monat` (1 = Okt bis 6 = März)
  - `wo_tag` (1 = Mo bis 7 = So)
  - `tag` (1 = Mo-Fr, 2=Sa, 3=So/Fe)
- 6 metrische Merkmale für Schadstoffe (24h–Tagesmittelwerte Tag t):
  - Feinstaub `pm10` $[\mu g(=10^{-6}g)/m^3]$
  - `pm_mittag` (24h_MW `pm10` von 12h Tag $t-2$ bis 12h Tag $t-1$)
  - Stickstoffmonoxid `no` $[\mu g/m^3]$
  - Stickstoffdioxid `no2`
  - Stickoxid `nox` $[ppb]$
  - Kohlenmonoxid `co` $[mg/m(=10^{-3}g)^3]$
- 7 metrische Merkmale für Meteorologie (24h–Tagesmittelwerte Tag t):
  - Lufttemperatur Graz–Mitte `lute` $[°C]$
  - `lute_mittag` (analog zu `pm_mittag`),
  - Luftfeuchtigkeit Graz–Mitte `lufe` [%]
  - Niederschlag Graz-Nord `nied` $[l/m^2]$
  - Windgeschwindigkeit Graz–Mitte `wige` $[m/s]$
  - Differenz Lufttemperatur Graz–Mitte minus Kalkleiten `ltusg_k`
  - Differenz Lufttemperatur Graz–Mitte minus Graz–Platte `ltusg_p`

Realisierung in R 2.6.0

```
> setwd("C:/Skript_Statistik_Mathematik_0708/Regression_GrazMitte")
> library(foreign) # Notwendig, um SPSS-Files einlesen zu können
> #************************************************************
> daten <- read.spss("Staubdaten_GM_0506.sav")
Warning messages: 1: In read.spss("Staubdaten_GM_0506.sav") :
  Staubdaten_GM_0506.sav: File-indicated character representation code
  (1252) looks like a Windows codepage
2: In read.spss("Staubdaten_GM_0506.sav") :
  Staubdaten_GM_0506.sav: Unrecognized record type 7,
  subtype 16 encountered in system file
> #************************************************************
> # Zuordnung der Variablen: in GROSSBUCHSTABEN abgespeichert
> nr <- daten$NR; datum <- daten$DATUM; monat <- daten$MONAT
> wotag <- daten$WO_TAG; tag <- daten$TAG; pm10 <- daten$PM10
> pm_mittag <- daten$PM_MITTA; no <- daten$NO; no2 <- daten$NO2
> nox <- daten$NOX; co <- daten$CO; lute <- daten$LUTE
> lute_mittag <- daten$LUTE_MIT; lufe <- daten$LUFE
> nied <- daten$NIED; wige <- daten$WIGE
> ltusg_k <- daten$LTUSG_K; ltusg_p <- daten$LTUSG_P
> #************************************************************
> # Boxplotserien bzgl. Monat, Wochentag und Tag mit
> # 95%-Konfidenzintervallen für den Median (notch=T)
> par(mfrow=c(2,2))

> boxplot(pm10~monat, varwidth=T, notch=T, xlab="Monat",
  ylab="PM10 (mug/m3)", main="Graz-Mitte")
> mtext("Winter 05/06", cex=0.8)

> boxplot(pm10~wotag, varwidth=T, notch=T, xlab="Wochentag",
  ylab="PM10 (mug/m3)", main="Graz-Mitte")
> mtext("Winter 05/06", cex=0.8)
Warning message: In bxp(list(stats = c(19.6323833, 29.81463346,
53.48714581, 68.99777119,  :
  some notches went outside hinges ('box'): maybe set notch=FALSE

> boxplot(pm10~tag, varwidth=T, notch=T, xlab="Tag",
> ylab="PM10 (mug/m3)", main="Graz-Mitte")
> mtext("Winter 05/06", cex=0.8)
```

Interpretation

- *Vergleich der Monate.* Die durchschnittliche Belastung und die Streuung von `pm10` ist im Jänner höher als in den anderen Monaten.
- *Vergleich der Wochentage.* Die Verteilung von `pm10` ist von Dienstag bis Donnerstag ähnlich. Am Montag sind niedrigere Werte und am Freitag eine kleinere Streuung festzustellen. Am Wochenende ist die Belastung tendenziell niedriger. Einige außergewöhnlich hohe Werte gibt es an allen Wochentagen.

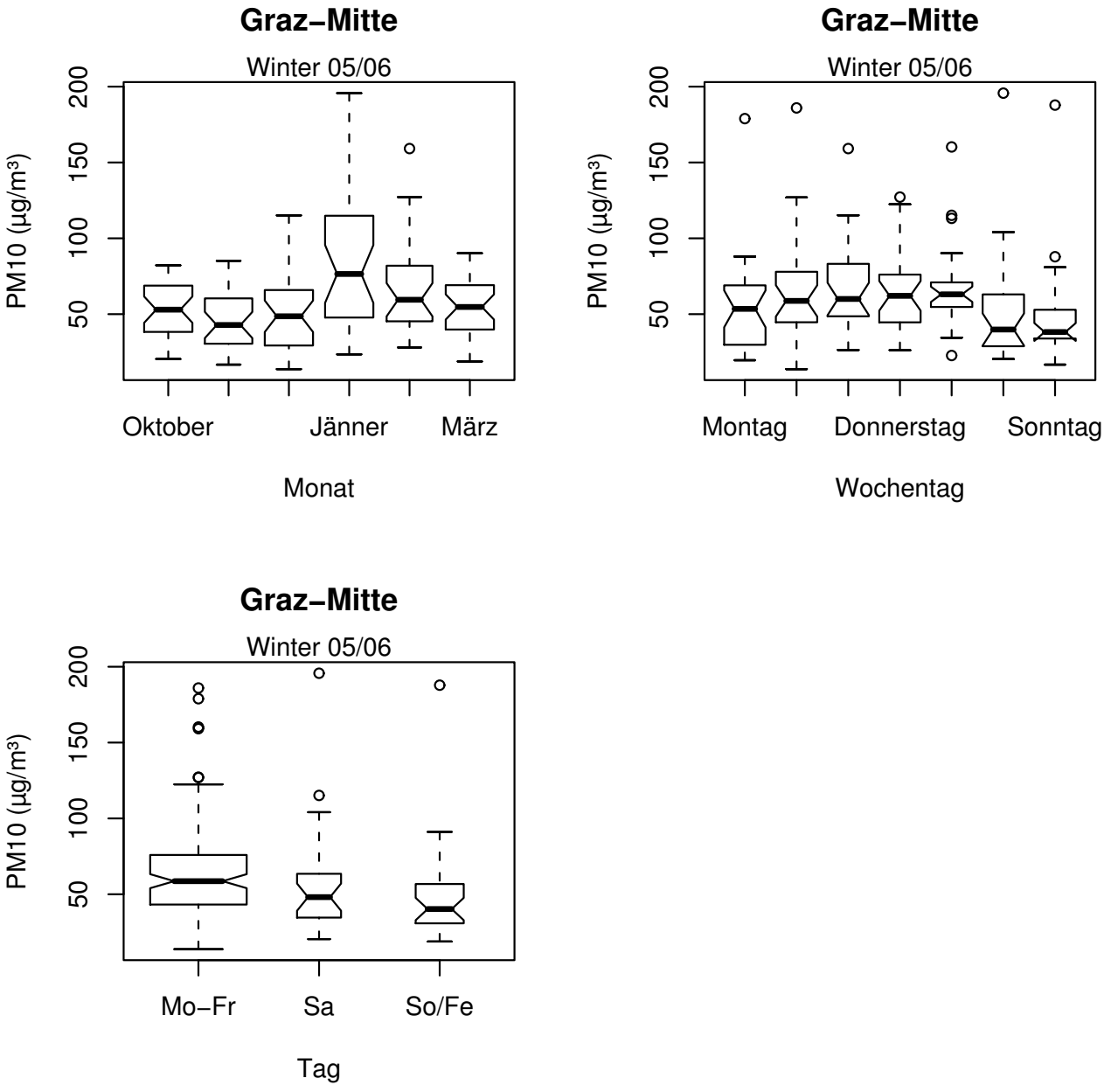

Abbildung 1: Boxplotserien von pm10 bzgl. Monat, Wochentag und Tagestyp

```
> # Monats-Mittelwerte, -Mediane und -Standardabweichungen
> aggregate(pm10, list(monat), mean); aggregate(pm10, list(monat), median)
> aggregate(pm10, list(monat), sd)
   Group.1        x
1  Oktober 52.89092 2 November 46.14350 3 Dezember 51.90545
4   Jänner 87.38002 5  Februar 69.08376 6     März 54.37113
   Group.1        x
1  Oktober 52.96660 2 November 42.84895 3 Dezember 48.60673
4   Jänner 76.49742 5  Februar 59.50230 6     März 54.74842
   Group.1        x
1  Oktober 18.45919 2 November 19.02967 3 Dezember 25.64911
4   Jänner 51.16026 5  Februar 31.52589 6     März 18.27780

> # Wochentags-Mittelwerte, -Mediane und -Standardabweichungen
> aggregate(pm10, list(wotag), mean); aggregate(pm10, list(wotag), median)
> aggregate(pm10, list(wotag), sd)
     Group.1        x
1     Montag 56.67061 2   Dienstag 66.03432 3   Mittwoch 67.60421
4 Donnerstag 64.37945 5    Freitag 67.55763 6    Samstag 51.59192
7    Sonntag 47.76277
     Group.1        x
1     Montag 53.48715 2   Dienstag 58.75675 3   Mittwoch 60.01562
4 Donnerstag 62.09113 5    Freitag 63.08200 6    Samstag 39.88296
7    Sonntag 38.22643
     Group.1        x
1     Montag 32.54696 2   Dienstag 34.71196 3   Mittwoch 32.23759
4 Donnerstag 27.37506 5    Freitag 28.09456 6    Samstag 36.34921
7    Sonntag 32.97808

> # Tagestyp-Mittelwerte, -Mediane und -Standardabweichungen
> aggregate(pm10, list(tag), mean); aggregate(pm10, list(tag), median)
  aggregate(pm10, list(tag), sd)
  Group.1        x
1   Mo-Fr 63.41714 2      Sa 56.23882 3   So/Fe 50.71845
  Group.1        x
1   Mo-Fr 58.62009 2      Sa 48.09824 3   So/Fe 40.25861
  Group.1        x
1   Mo-Fr 31.42244 2      Sa 36.59319 3   So/Fe 32.31122
```

Bemerkung

Explorative Analysen haben gezeigt, dass der Tagesmittelwert `pm10` (Tag t) vom 24h Mittelwert `pm_mittag` (12h Tag $(t-2)$ bis 12h Tag $(t-1)$) abhängt. Zusätzlich spielen meteorologische Einflüsse wie *Niederschlag*, *Windgeschwindigkeit* und die *Inversionswetterlage* (Differenz der Lufttemperatur in Graz-Mitte und ca. 350 m über dem Boden in Kalkleiten) eine Rolle. Unser erstes Regressionsmodell für `pm10` enthält 4 x–Variable.

```
> #*****************************************************************
> # Modell für y = pm10 mit x1=pm_mittag, x2=Niederschlag,
> x3=Windgeschwindigkeit, x4=Temp.differenz Graz-Kalkleiten

> pm.lm <- lm(pm10~pm_mittag+nied+wige+ltusg_k); summary(pm.lm)

Call: lm(formula = pm10 ~ pm_mittag + nied + wige + ltusg_k)

Residuals:
     Min       1Q   Median       3Q      Max
-54.3189 -15.7984  -0.8052  13.4803  74.6371

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  42.93106    6.24241   6.877 1.01e-10 ***
pm_mittag     0.55831    0.05671   9.845  < 2e-16 ***
nied         -2.11084    0.68206  -3.095 0.002290 **
wige        -11.97332    6.81911  -1.756 0.080844 .
ltusg_k      -4.29015    1.11222  -3.857 0.000160 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '  1

Residual standard error: 22.09 on 177 degrees of freedom
Multiple R-Squared: 0.5496,     Adjusted R-squared: 0.5394
F-statistic: 54 on 4 and 177 DF,  p-value: < 2.2e-16

> #*****************************************************************
> par(mfrow=c(2,2))
> # 4 Residuenplots für Beurteilung der Adäquatheit des Modells
> plot(pm.lm)
```

Interpretation des Modells und der Residuenplots

- Die wichtigste x–Variable ist `pm_mittag`, gefolgt von `ltusg_k` und `nied`. `wige` ist auf dem 5%–Niveau *nicht signifikant* und könnte weggelassen werden.
- Durch das Modell werden ca. 54% der Gesamtvarianz erklärt, falls das Modell zulässig ist.
- Die beiden Plots in der ersten Spalte zeigten keine konstante Streuung der Residuen über den Bereich der gefitteten Werte. Dies ist ein Hinweis, dass die Annahme einer konstanten Streuung σ verletzt ist.
- Der Normal-Q-Q-Plot zeigt einige Abweichungen am rechten Rand (standardisierte Residuen $r_i* > 2$), die auf höhere Tails als die der Normalverteilung hindeuten. Daher wird im folgenden ein Modell mit der transformierten Variablen `sqrt_pm` gerechnet.
- Im Leverage-Plot erkennt man einige Beobachtungen mit Hebelwerten > 0.1, aber ohne auffällig großer Cook-Distanz.

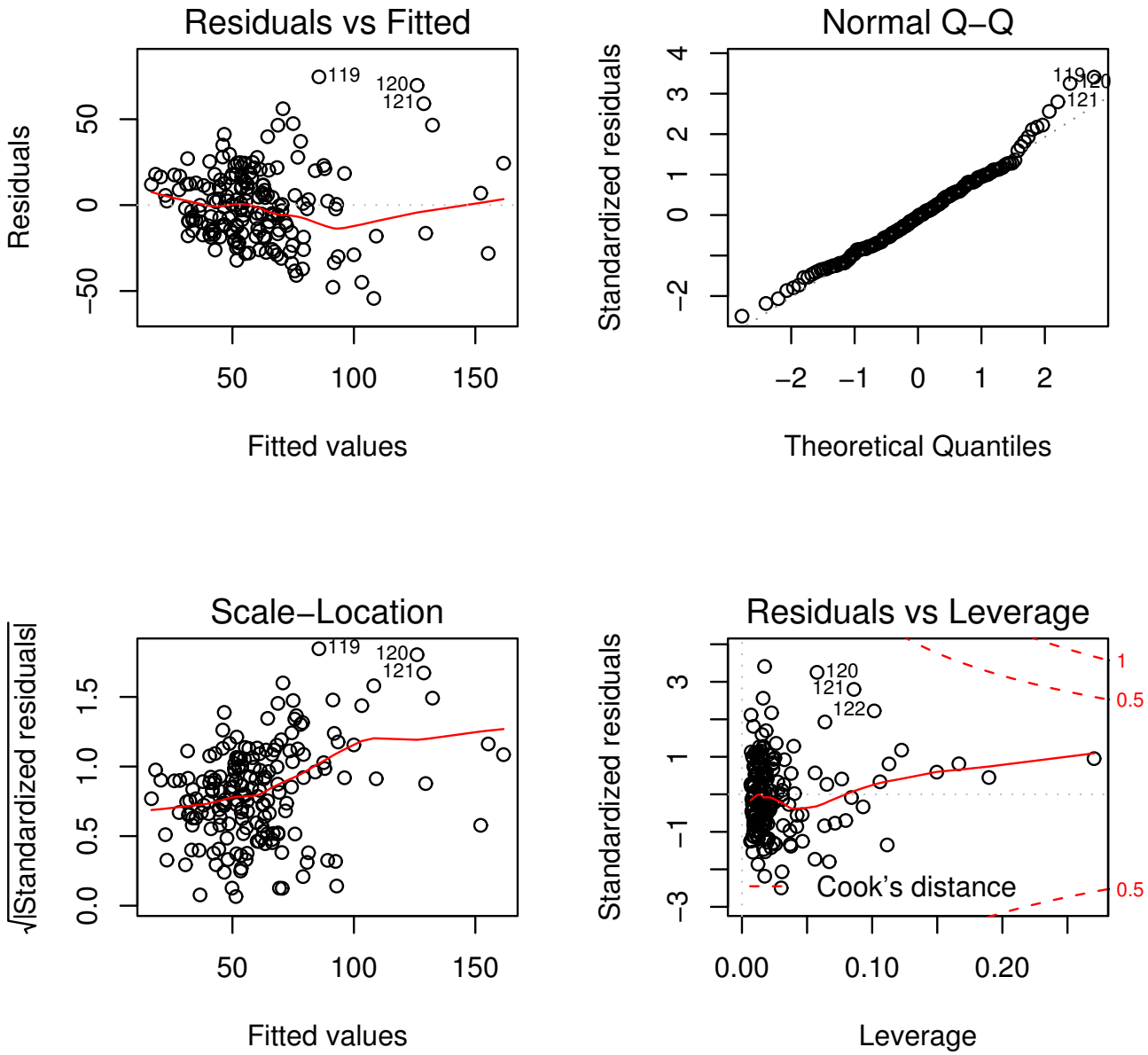

Abbildung 2: Residuenplots für das Modell mit $y = pm10$ und 4 Kovariable

```
> #************************************************************
> # Modell für tranformierte Variable y=sqrt_pm
> sqrt_pm <- sqrt(pm10)
> sqrt_pm.lm <- lm(sqrt_pm~pm_mittag+nied+wige+ltusg_k); summary(sqrt_pm.lm)

Call: lm(formula = sqrt_pm ~ pm_mittag + nied + wige + ltusg_k)

Residuals:
     Min       1Q   Median       3Q      Max
-2.91363 -1.01352  0.03075  0.94300  3.72385

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.798772   0.378447  17.965  < 2e-16 ***
 pm_mittag   0.029591   0.003438   8.607 3.98e-15 ***
 nied       -0.164655   0.041350  -3.982 9.97e-05 ***
 wige       -0.764001   0.413409  -1.848 0.066264 .
 ltusg_k    -0.267786   0.067429  -3.971 0.000104 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.339 on 177 degrees of freedom
Multiple R-Squared: 0.5241,     Adjusted R-squared: 0.5134
F-statistic: 48.74 on 4 and 177 DF,  p-value: < 2.2e-16
< #************************************************************

> # Residuenplots
> par(mfrow=c(2,2))
> plot(sqrt_pm.lm)
```

Interpretation

- Das Modell für die transformierte Variable `sqrt_pm` zeigt einen größeren Einfluss des Niederschlags als im Modell für `pm10`. Ansonsten sind die Charakteristiken ähnlich. So ist z.B. $r^2_{adj} = 0.51$.
- Bei einem hohen 24h-Wert `pm_mittag` ist auch ein hoher `pm10` Wert zu erwarten. Bei *Niederschlag* und *Wind* kann mit einer geringeren Konzentration von `pm10` gerechnet werden. Bei *Temperaturinversion* (`ltusg_k < 0`) ist eine höhere `pm10` Belastung zu erwarten.
- Die Residuenplots in der ersten Spalte zeigen eine gleichmäßigere Varianz über den Bereich der gefitteten Werte.
- Der Normal-Q-Q-Plot weist weniger standardisierte Residuen $r_i^* > 2$ auf.
- Die Transformation hat die Varianz stabilisiert und die Verteilung der Residuen näher in Richtung Normalverteilung gebracht.

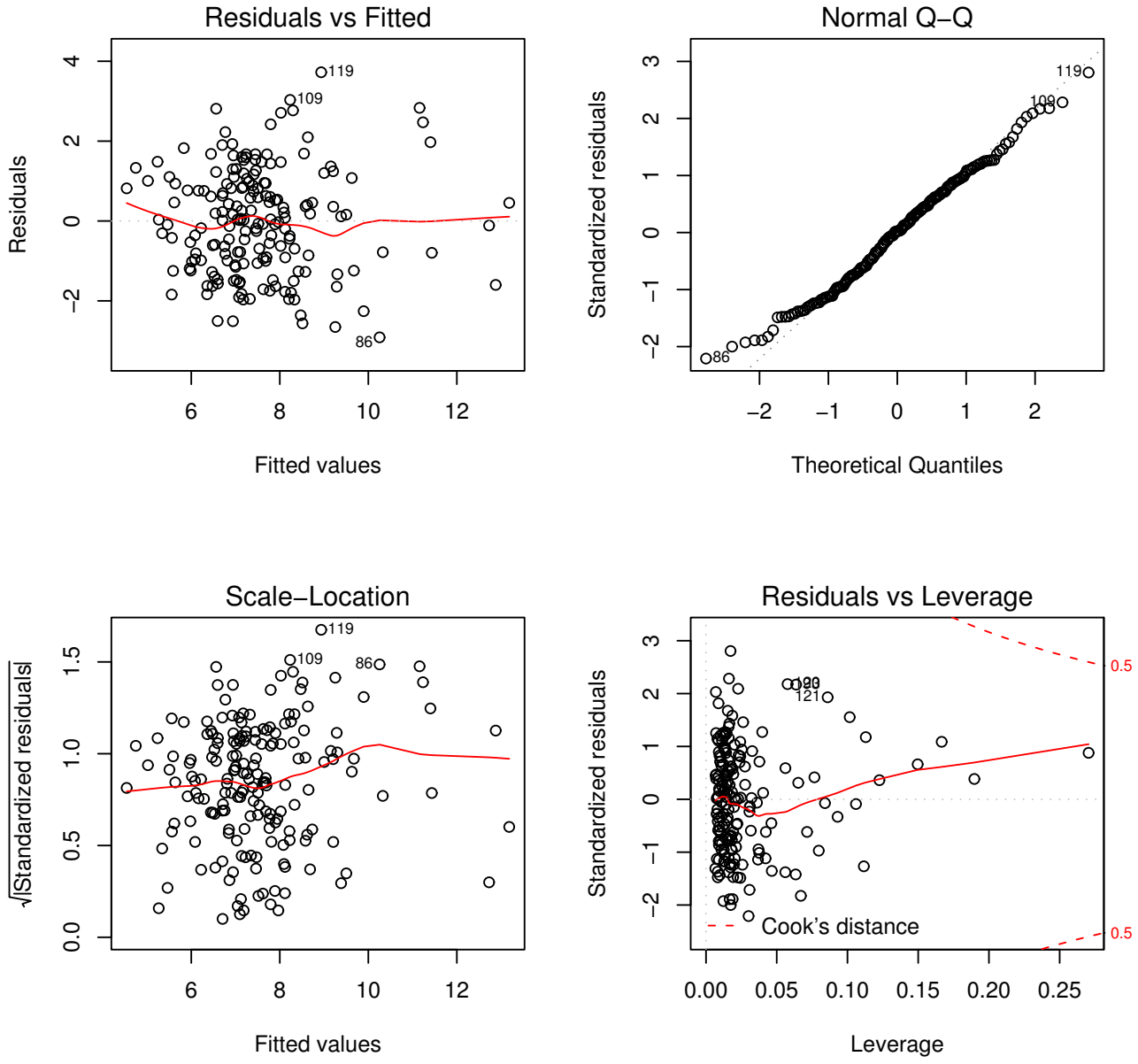

Abbildung 3: Residuenplots für das Modell mit $y = \sqrt{pm10}$ und 4 Kovariable

```
> #*************************************************************
> # Vergleich Modellwerte von (sqrt_pm10)^2 mit Beobachtungen von pm10

> par(mfrow=c(1,1))
> prognose <- (fitted.values(sqrt_pm.lm))^2
> plot(nr, pm10, type = "l", xlab="Datum", ylab="PM10 (mu/m3)",
  main="Graz-Mitte 05/06")
> lines(prognose, col=2, lty=2); abline(h=50, lty=2)
> mtext("PM10 | Prognose", cex=0.8); legend(0, 200, c("- PM10", "-- Prognose"))

> # Scatterplot Prognosewerte gegen Beobachtungswerte
> plot(pm10,prognose,pch=unclass(tag),col=unclass(tag))
> legend(10,170,levels(tag),pch=1:3,col=1:3)
> abline(h=c(50,100), lty=2); abline(v=c(50,100), lty=2)
```

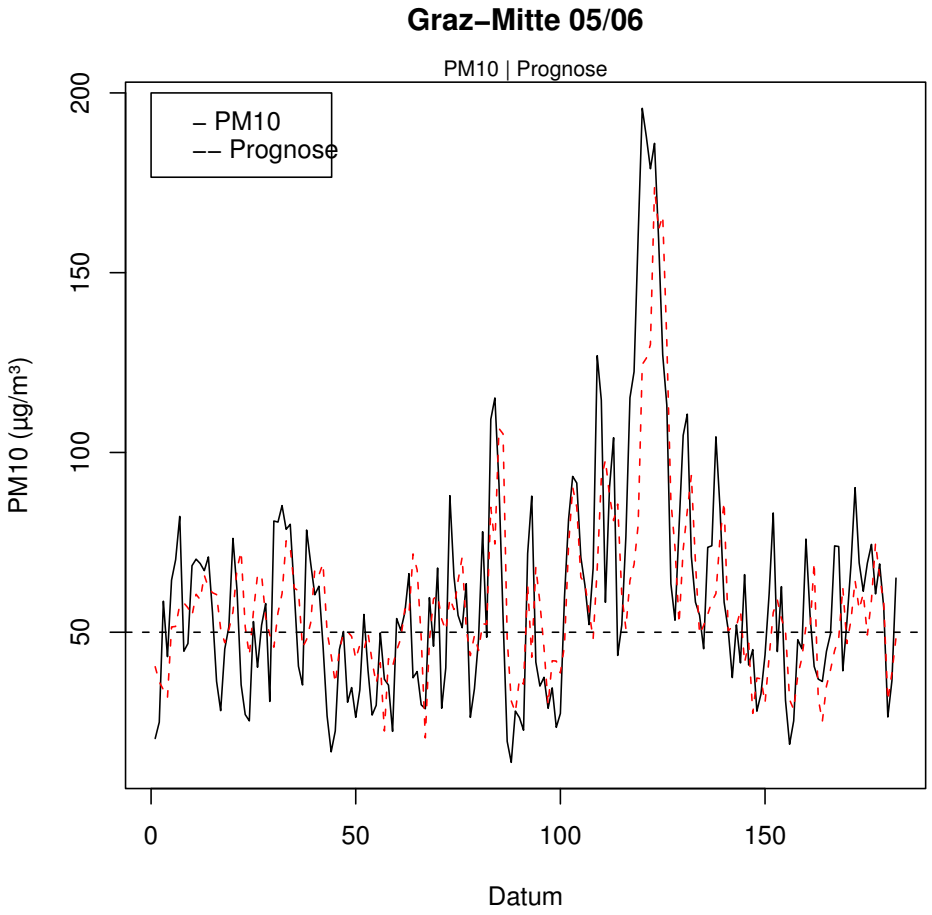

Abbildung 4: Vergleich der Prognosewerte für `pm10` (Modell $y = \sqrt{pm10}$ mit 4 Kovariablen) mit beobachteten Werten von `pm10`

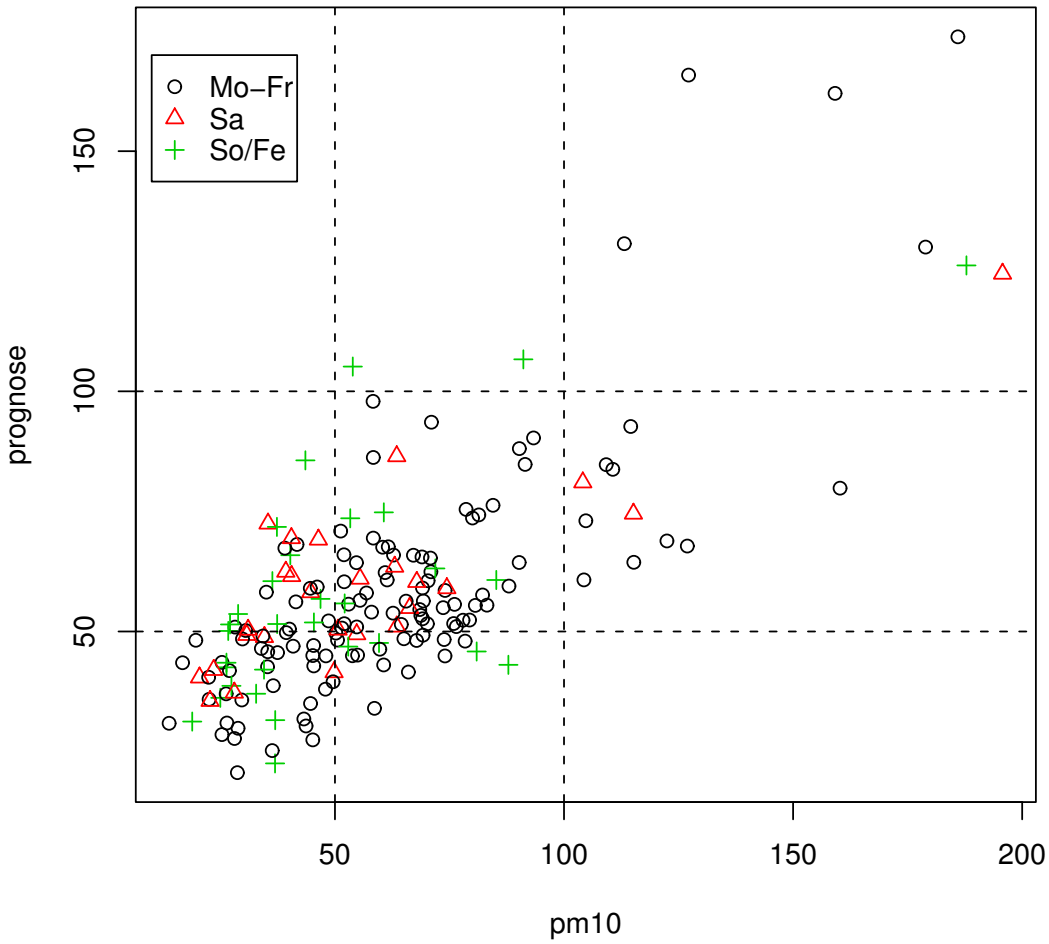

Abbildung 5: Scatterplot der Prognosewerte gegen beobachtete Werte

Interpretation

- Abbildung 4 zeigt den Vergleich der Prognosen mit den beobachteten Werten im Zeitverlauf. Unterschiede lassen sich hier nur schwer erkennen.
- Aus Abbildung 5 erkennt man hingegen Unterschiede zwischen den prognostizierten und den beobachteten Werten. Prognosewerte, die im gleichen Quadranten wie die Beobachtungen liegen, sind prinzipiell als *sehr gut* bis *zufriedenstellend* einzustufen.
- Wertepaare im Quadranten `pm10 > 100` und `50 < prognose < 100` zeigen z.B. zu niedrig prognostizierte Werte an.
- Man kann ca. 85% der Prognosen als *zufriedenstellend* ansehen, wenn man eine entsprechende Qualitätsfunktion einführt, die Abweichungen unterschiedlich stark gewichtet.

```
> #*****************************************************************
> # Zweites Modell für y=sqrt_pm10 unter Berücksichtigung von Wochenende

> # tag wird umgewandelt in 2 dummy variable tagSa(0,1) und tagSo/Fe(0,1),
> # deren Koeffizienten den Einfluss von Samstag bzw. So/Feiertag angeben
> sqrt_pm.lm1 <- lm(sqrt_pm~pm_mittag+nied+wige+ltusg_k+tag)
> summary(sqrt_pm.lm1)

Call:
lm(formula = sqrt_pm ~ pm_mittag + nied + wige + ltusg_k + tag)

Residuals:
     Min       1Q   Median       3Q      Max
-2.74157 -0.97422 -0.05573  0.93986  3.49476

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.989047   0.371994  18.788  < 2e-16 ***
pm_mittag    0.029853   0.003345   8.925 5.92e-16 ***
nied        -0.158962   0.040446  -3.930 0.000122 ***
wige        -0.735495   0.402868  -1.826 0.069608 .
ltusg_k     -0.268914   0.065758  -4.089 6.58e-05 ***
tagSa       -0.659744   0.281508  -2.344 0.020221 *
tagSo/Fe    -0.786870   0.262426  -2.998 0.003109 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.301 on 175 degrees of freedom
Multiple R-Squared: 0.5556,     Adjusted R-squared: 0.5404
F-statistic: 36.47 on 6 and 175 DF,  p-value: < 2.2e-16
> #*****************************************************************
> par(mfrow=c(2,2))
> plot(sqrt_pm.lm1)

> #*****************************************************************
> # Vergleich Prognose mit Beobachtung
> par(mfrow=c(1,1))
> prognose1 <- (fitted.values(sqrt_pm.lm1))^2
> plot(nr, pm10, type = "l", xlab="Datum", ylab="PM10 (mug/m3)",
  main="Graz-Mitte 05/06"); lines(prognose1, col=2,lty=2)
> abline(h=50, lty=2); mtext("PM10 | Prognose1", cex=0.8)
> legend(0, 200, c("- PM10", "-- Prognose1"))

plot(pm10,prognose1,pch=unclass(tag),col=unclass(tag))
legend(10,170,levels(tag),pch=1:3,col=1:3)
abline(h=c(50,100),lty=2); abline(v=c(50,100),lty=2)
```

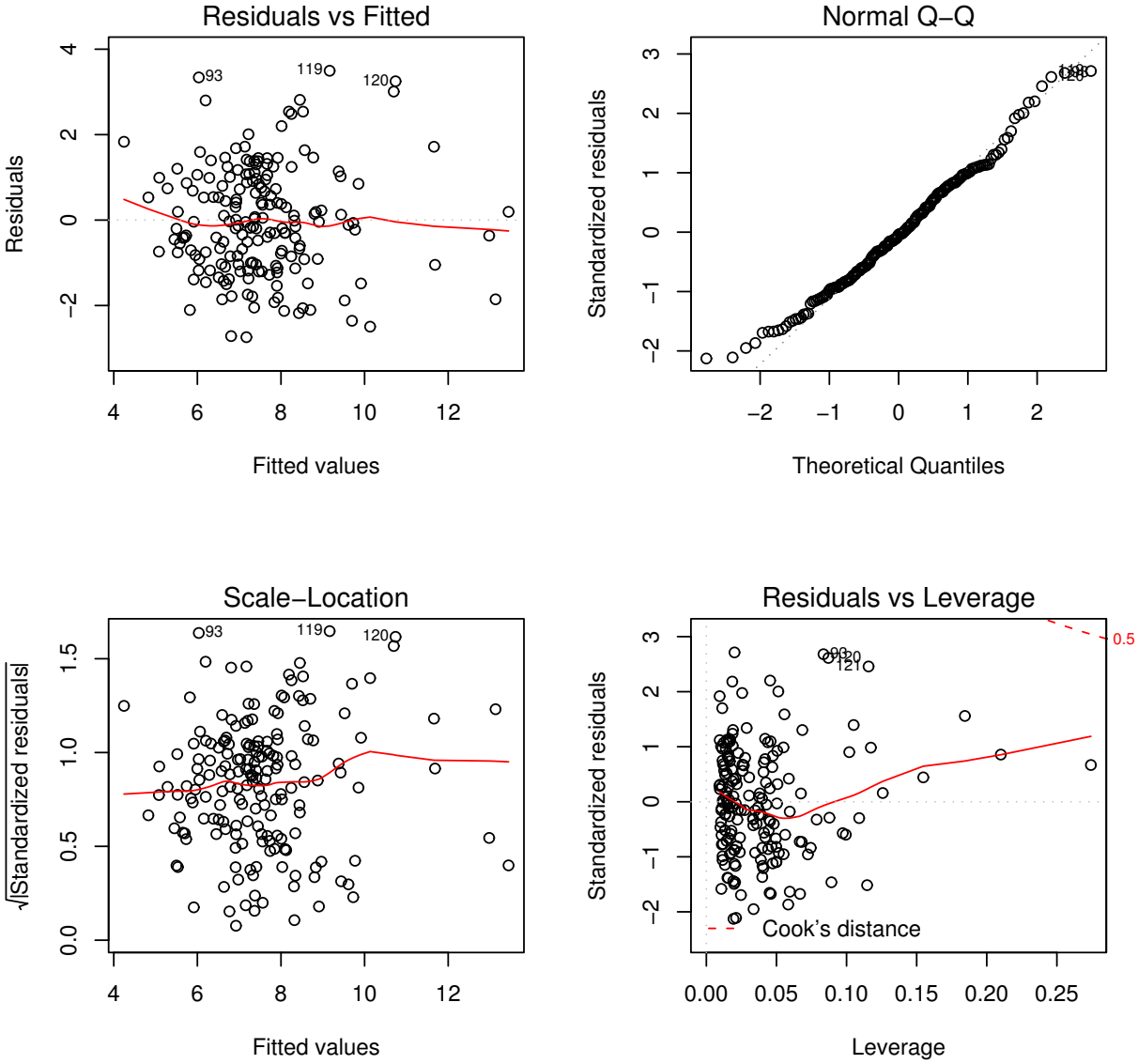

Abbildung 6: Residuenplots für das Modell mit $y = \sqrt{pm10}$ und 5 Kovariable

Interpretation

- An Samstagen sowie Sonn/Feiertagen ist unter gleichen meteorologischen Bedingungen ein niedrigerer `pm10`–Wert als an Werktagen zu erwarten (negative Koeffizienten in der Regressionsgleichung).
- Die Hinzunahme der Variablen `tagSa` und `tagSo/Fe` bewirkt eine Verbesserung des Bestimmtheitsgrades von 51% auf 54% sowie eine Verringerung des Standardfehlers s_e von 1.34 auf 1.30.
- Die Residuenplots weisen eine zufrieden stellende Charakteristik der Residuen aus.

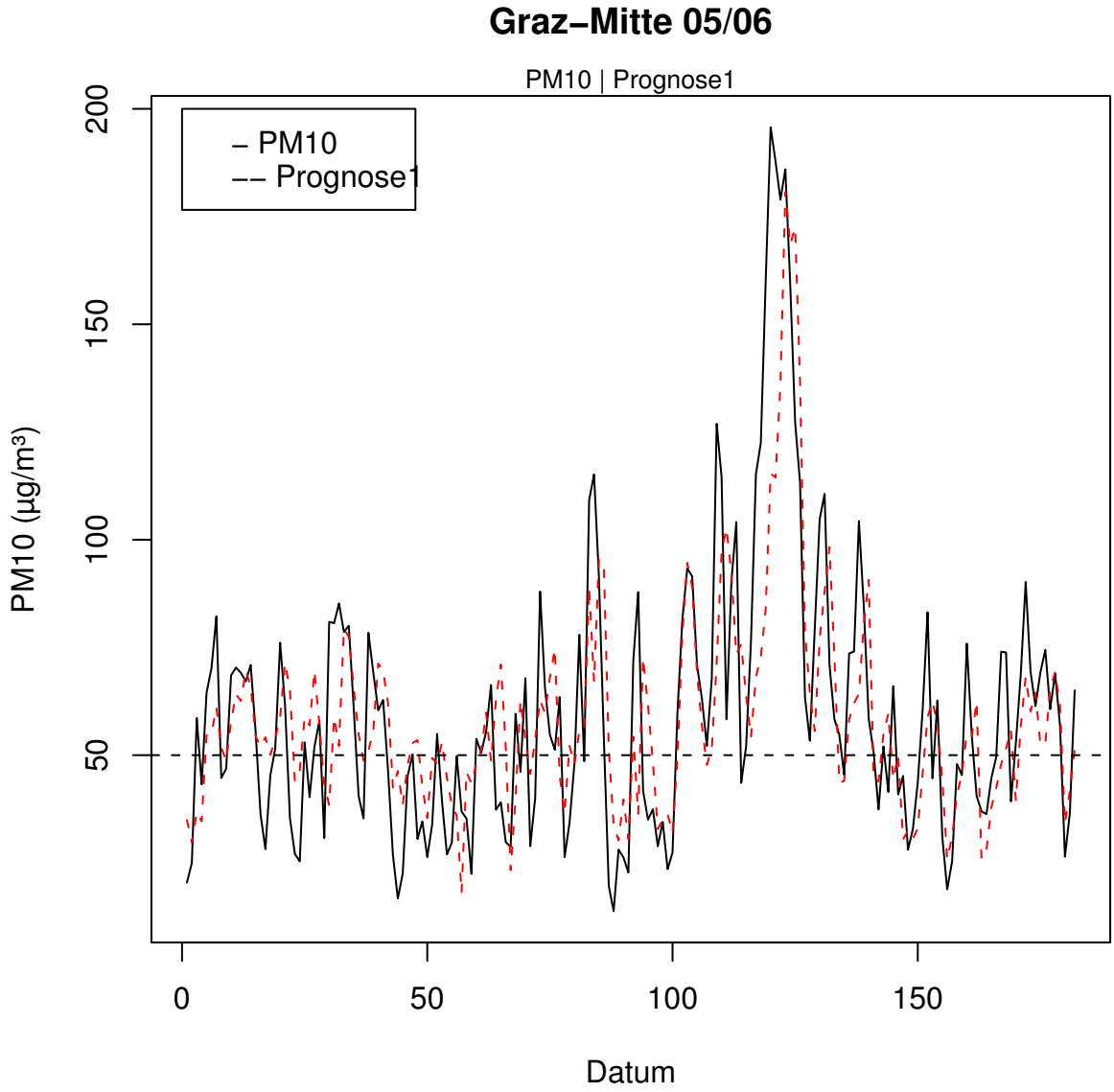

Abbildung 7: Vergleich der Prognosewerte für `pm10` (Modell $y = \sqrt{pm10}$ mit 5 Kovariablen) mit beobachteten Werten von `pm10`

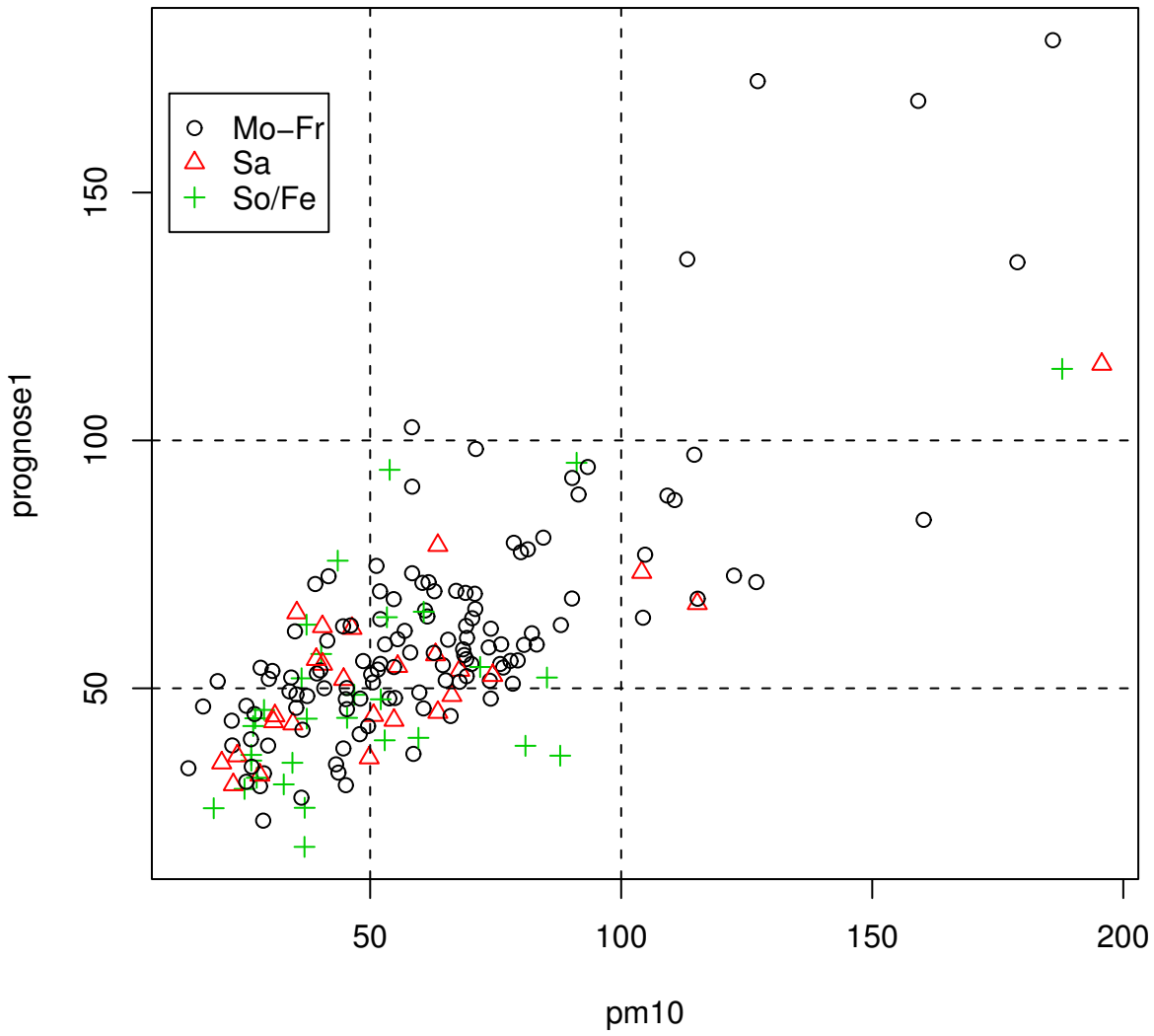

Abbildung 8: Scatterplot der Prognosewerte gegen beobachtete Werte

Interpretation

- Die Qualität einiger Prognosen an Sonn/Feiertagen konnte verbessert werden. (Man erkennt weniger " + " in den " falschen" Quadranten).
- Ein ähnliches, etwas komplexeres Modell mit 9 `x`–Variablen, basierend auf 4 Wintersaisonen, wurde für Graz–Mitte aufgestellt ($r^2_{adj} = 0.63, s_e = 1.09$). Seit 3 Jahren wird damit in der Wintersaison bis spätestens 14 Uhr des Vortages ein täglicher Prognosewert für `pm10` des nächsten Tages erstellt. Anstatt der "beobachteten" Wetterwerte gehen in der Regressionsgleichung die durch Meteorologen gelieferten Wettervorhersagewerte ein (siehe auch www.feinstaub.at).
- Die Analysen der Wintersaisonen 2005/06 und 2006/07 haben ergeben, dass ca. 74% der Vorhersagen als zufriedenstellend eingestuft werden konnten.

Literatur. *Stadlober E., Hörmann S., B. Pfeiler*, Quality and performance of a PM10 daily forecasting model, *Athmospheric Environment* (2007), doi:10.106/j.atmosenv.2007.10.073 (to appear).