Robuste Statistik

Musterlösung zu Blatt 8

Aufgabe 8.1:

Zu zeigen ist für den MAD

$$\epsilon^*(d_{MAD}, y) = \frac{1}{N}\frac{N-1}{2},$$

für $y \in \mathbb{R}^N$ bei ungeradem N.

Nach Satz 4.2.4 gilt für den Bruchpunkt des MADs

$$\epsilon^*(d_{MAD}, y) \leq \frac{1}{N}\frac{N-1}{2} \tag{1}$$

da der MAD lokations-invariant und skalen-äquivariant ist (1 Punkt).

Es genügt zu zeigen, dass der Explosionspunkt und der Implosionspunkt nicht kleiner als die angegebene obere Schranke in Formel (1) werden kann.

Für den Explosionspunkt kann man sogar zeigen

$$\epsilon^+(d_{MAD}, y) \geq \frac{1}{N}\frac{N+1}{2}.$$

Sei dazu $M \leq \frac{N+1}{2} - 1$ eine beliebige Anzahl an Verfälschungen von y dargestellt durch den Vektor $\tilde{y}$. Wir schreiben $d_n := |\tilde{y}_n - med(\tilde{y})|$ und $d = (d_1, \ldots, d_N)^\top$. Angenommen, $d_{MAD}(\tilde{y}) \to \infty$ durch M Verfälschungen. Dann muss einer der beiden Fälle eingetreten sein:

$$\begin{aligned} (I) \quad & med(\tilde{y}) \to \infty, \\ (II) \quad & med(d) \to \infty. \end{aligned}$$

(I) ist ein Widerspruch zum Explosionspunkt des Medians mit $\epsilon^+(med, y) = \frac{1}{N}\frac{N+1}{2}$ (vgl. Übung 5). Angenommen (II) gilt, d.h. $d_{(\frac{N+1}{2})} \to \infty$. Dann folgt

$$\begin{aligned} \Rightarrow \quad & |\tilde{y}_{(\frac{N+1}{2})} - med(\tilde{y})| \to \infty \\ \Rightarrow \quad & \exists k : \tilde{y}_k \to \pm\infty \\ \Rightarrow \quad & \exists k_1, \ldots, k_{\frac{N+1}{2}} : \tilde{y}_{k_1}, \ldots, \tilde{y}_{k_{\frac{N+1}{2}}} \to -\infty \text{ oder } \tilde{y}_{k_1}, \ldots, \tilde{y}_{k_{\frac{N+1}{2}}} \to \infty. \end{aligned}$$

Beide Aussagen implizieren aber, dass es mindestens $\frac{N+1}{2}$ Verfälschungen geben muss, was ein Widerspruch zur Annahme an M ist (2 Punkte).

Für den Implosionspunkt zeigen wir

$$\epsilon^-(d_{MAD}, y) \geq \frac{1}{N}\frac{N-1}{2}.$$

Sei $M \leq \frac{N-1}{2} - 1$ eine beliebige Anzahl von Verfälschungen von Y dargestellt durch den Vektor $\tilde{y}$. Angenommen,

$d_{MAD}(\tilde{y}) = 0$ durch M Verfälschungen. Dann gilt

$$d_{\left(\frac{N+1}{2}\right)} = 0, \text{ also } y_{\left(\frac{N+1}{2}\right)} = med(\tilde{y})$$
$$\Rightarrow \; d_{(1)} = \ldots = d_{\left(\frac{N-1}{2}\right)} = 0.$$

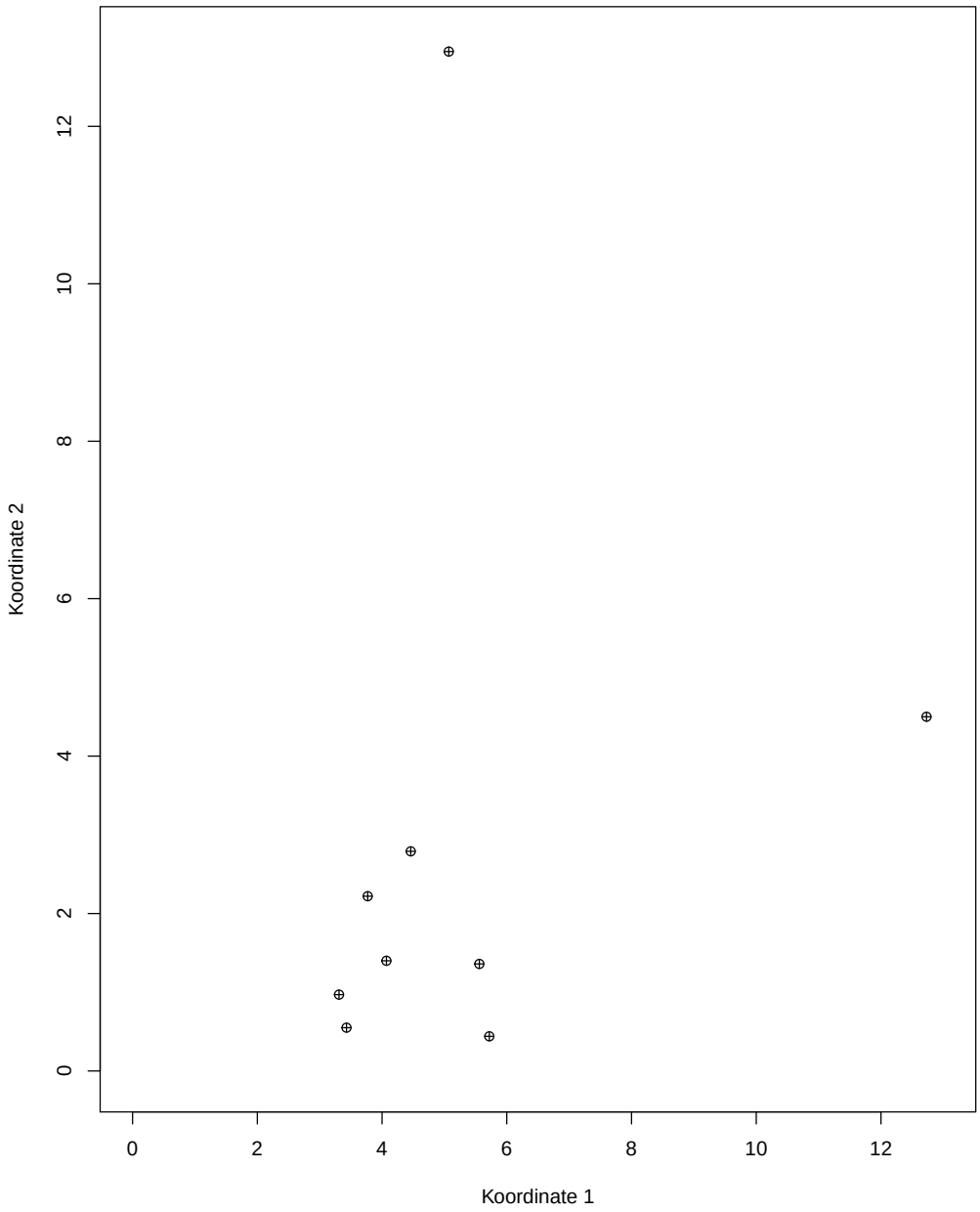
Beachte, dass es immer ein $n \in \{1, \ldots, N\}$ gibt, sodass $d_n = 0$, da es bei einem Datensatz mit ungerader Länge immer ein n gibt mit $\tilde{y}_n = med(\tilde{y})$. Da y paarweise unterschiedliche Werte aufwies, gibt es $\frac{N-1}{2}$ Verfälschungen, was ein Widerspruch dazu ist, dass es maximal $\frac{N-1}{2} - 1$ Verfälschungen gibt (2 Punkte).

Hinweis: Statt Satz 4.2.4 zu nutzen, kann man auch zeigen, dass der Implosionspunkt ebenso durch $\frac{1}{N}\frac{N-1}{2}$ nach oben beschränkt ist. Eine obere Schranke für den Explosionspunkt zu zeigen ist in diesem Fall nicht zielführend (aber trotzdem interessant beim Durchrechnen).

Aufgabe 8.2:

a) Lese zunächst die Daten ein und erstelle Funktionen zur Berechnung der Kenngrößen:

```
# Liste mit den neun Ortspunkten
data <- list(
  y1 = c(3.43, 0.55),
  y2 = c(3.77, 2.22),
  y3 = c(5.56, 1.36),
  y4 = c(4.07, 1.4),
  y5 = c(5.07, 12.95),
  y6 = c(5.72, 0.44),
  y7 = c(4.46, 2.79),
  y8 = c(12.73, 4.5),
  y9 = c(3.31, 0.97)
)

X <- matrix(unlist(data), ncol = 9)


plot(X[1,], X[2,], xlab = "Koordinate 1", ylab = "Koordinate 2",
     pch = 20, xlim = c(0,13), ylim = c(0,13))
```



Es fallen zwei Datenpunkte (y_5 und y_8) besonders auf.

b) Berechnung der Maßzahlen für den Datensatz, die sich weit weg vom zentralen Cluster befinden.

```
# kompMean: Funktion zur Berechnung des komponentenweisen Mittelwerts
# Eingabe: data (list): Liste mit gleichlangen numerischen Vektoren, die die
#                       Datenpunkte beschreiben
# Ausgabe: (numeric): Vektor, der den komponentenweisen Mittelwert beschreibt
kompMean <- function(data) rowMeans(as.data.frame(data))

# kompMedian: Funktion zur Berechnung des komponentenweisen Medians
# Eingabe: data (list): Liste mit gleichlangen numerischen Vektoren, die die
#                       Datenpunkte beschreiben
# Ausgabe: (numeric): Vektor, der den komponentenweisen Median beschreibt
kompMedian <- function(data) apply(as.data.frame(data), 1, median)

# l1Median: Funktion zur Berechnung eines l1-Medians
```

```
# Eingabe: data (list): Liste mit gleichlangen numerischen Vektoren, die die
#                       Datenpunkte beschreiben
#          par (numeric): Startpunkt fuer den Optimierungsalgorithmus.
#                         Standardmaessig auf den komponentenweisen Mittelwert
#                         gesetzt
# Ausgabe: (numeric): Vektor, der einen l1-Median (approximativ) beschreibt
l1Median <- function(data, par = rowMeans(as.data.frame(data))){
  optFun <- function(mu)
    sum(sapply(data, function(x) sqrt(sum((x - mu)^2))))
  return(optim(par = par, fn = optFun)$par)
}
```

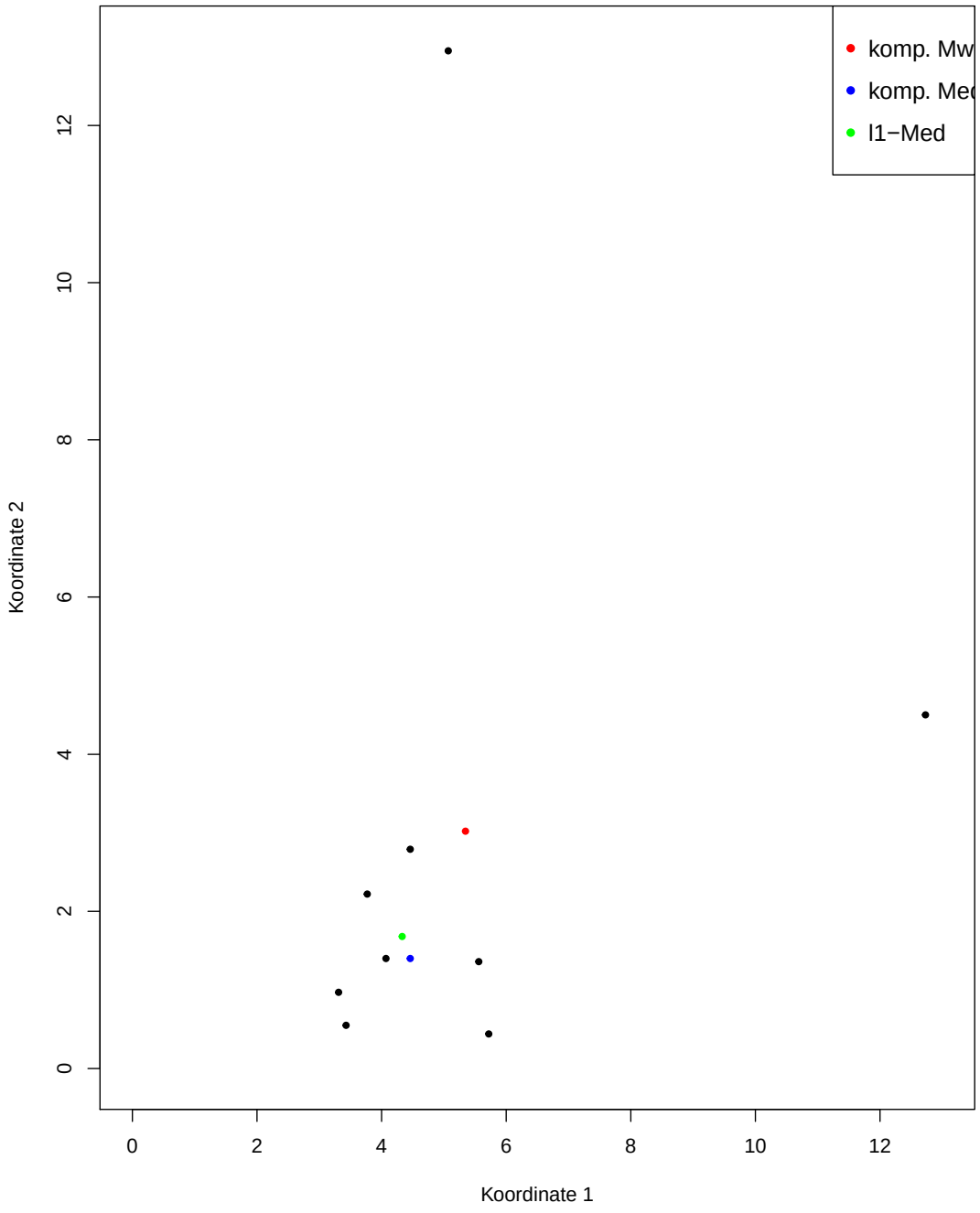
Berechne nun die geforderten Kennzahlen:

```
# komponentenweiser Mittelwert
kompMean(data)
# [1] 5.346667 3.020000

# komponentenweiser Median
kompMedian(data)
# [1] 4.46 1.40

# l1-Median
l1Median(data)
# [1] 4.330181 1.680238
```

Es fällt auf, dass sich bei diesem Datensatz der komponentenweise Median und der l_1-Median sich ähneln. Der komponentenweise Mittelwert ist jedoch von den anderen beiden Werten verschieden. Er liegt hier in beiden Komponenten über den Werten der Mediane, da auch jeweils in beiden Koordianten ein großer Wert bei 12 liegt, der die Mittelwerte koordinatenweise nach oben drückt.

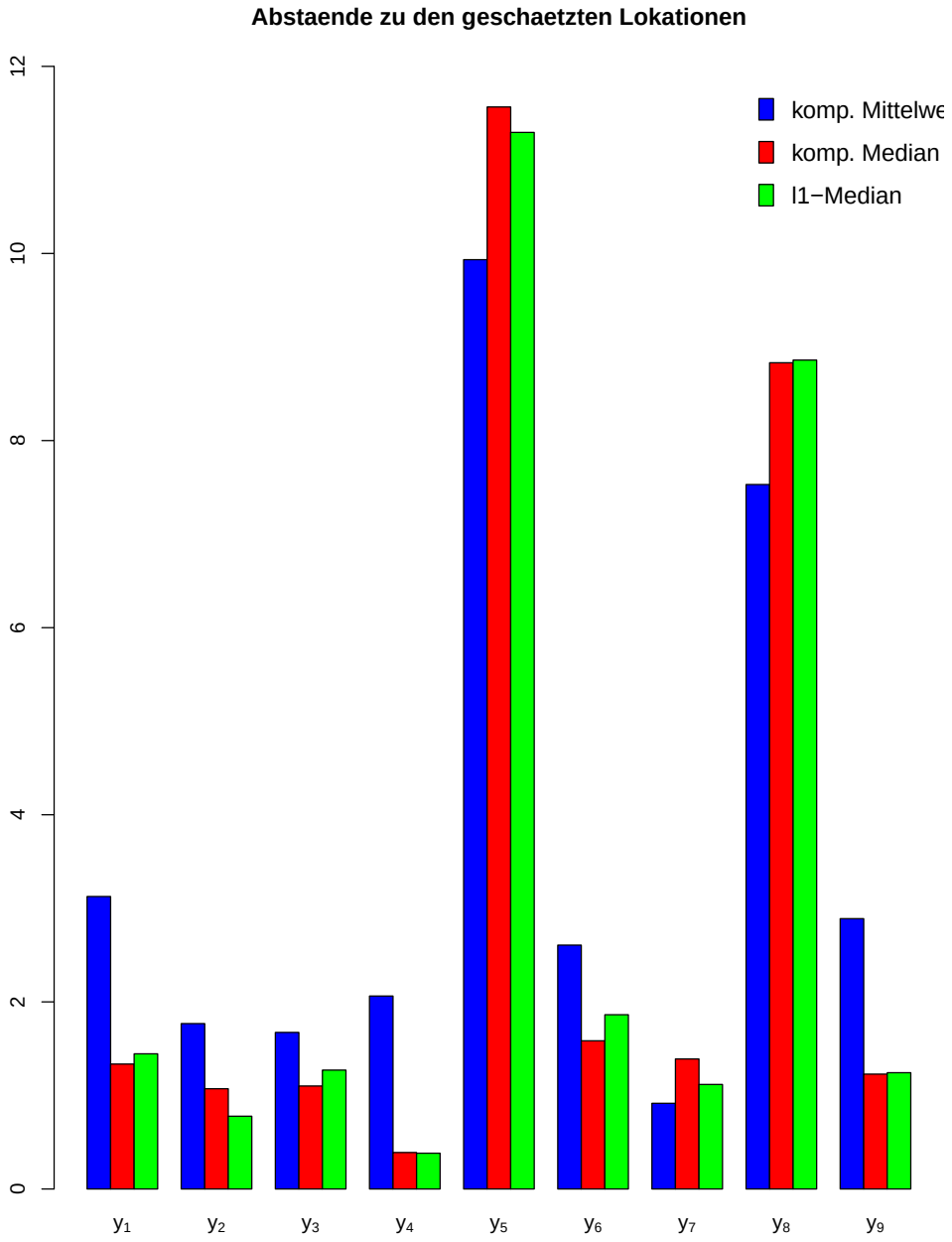
Man kann die geschätzten Lokationen in den Datensatz hinzufügen:

```
plot(X[1,], X[2,], xlab = "Koordinate 1", ylab = "Koordinate 2",
     pch = 10, xlim = c(0,13), ylim = c(0,13))
points(loc_mean[1],loc_mean[2], col = "red")
points(loc_med[1],loc_med[2], col = "blue")
points(loc_l1med[1],loc_l1med[2], col = "green")
legend("topright", legend = c("komp. Mw", "komp. Med", "l1-Med"), col = c("red", "blue", "green"),
       lty = c(1,1,1))
```



c) Berechnung der Abstände

```
a1 <- apply(X-loc_mean, 2, norm, type = "2")
a2 <- apply(X-loc_med, 2, norm, type = "2")
a3 <- apply(X-loc_l1med, 2, norm, type = "2")

barplot(t(matrix(c(a1,a2,a3),ncol = 3)), beside = T, ylim = c(0,12),
        col = c('blue','red','green'),
        names.arg = c(expression(y[1]),expression(y[2]),expression(y[3]),
                      expression(y[4]),expression(y[5]),expression(y[6]),
                      expression(y[7]),expression(y[8]),expression(y[9])),
        main = "Abstaende zu den geschaetzten Lokationen")
legend('topright', c('komp. Mittelwert','komp. Median', 'l1-Median'),
       fill = c('blue','red','green'), bty='n', cex=1.2)
```



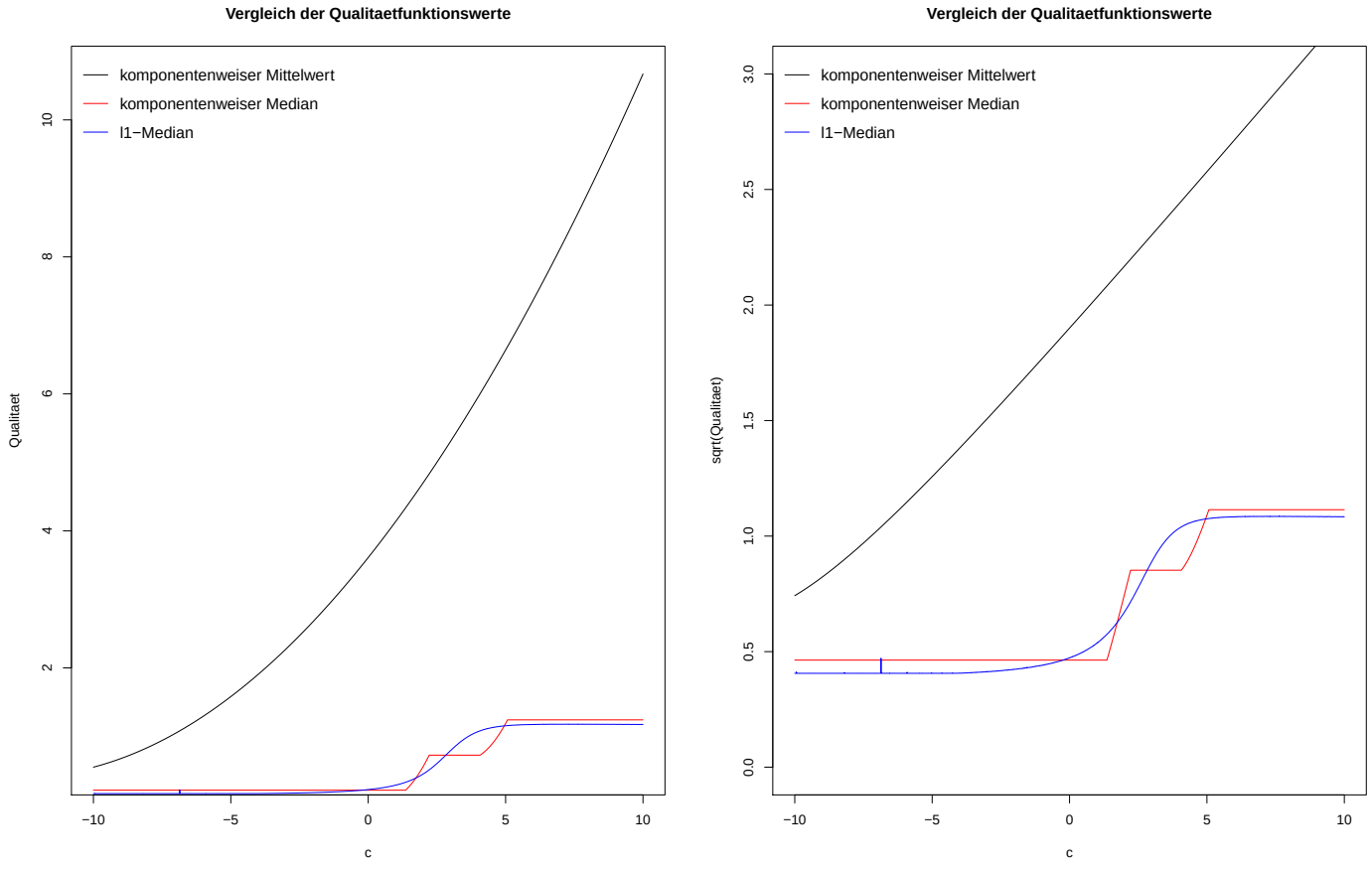
Auffällig sind erneut die Werte y_5 und y_8. Bei den Medianen fallen diese beiden Werte stärker auf als beim komponentenweisen Mittelwert. Dieses Verfahren kann genutzt werden, um Ausreißer zu klassifizieren. Insbesondere ist es **unabhängig von der Dimension**, weswegen sie einen deutlichen Vorteil gegenüber der Grafik aus a) hat! Es werden gerade im Hochdimensionalen solche Verfahren verwendet, da dort das Ausreißerverhalten erst durch simultane Koordinatenuntersuchung erst bemerkbar werden kann und man durch einfache Projektionen dieses Verhalten noch nicht erkennt. Ein Nachteil der Grafik in c) ist, dass sie nicht die Ausrichtung der Ausreißer angibt.

d) Erstelle zunächst die Qualitätsfunktion in `R`:

```
# guete: Funktion zur Berechnung der Qualitaetsfunktion Q
# Eingabe: estim (numeric): Vektor der geschaetzten Parameter
#          true (numeric): Vektor der wahren Parameter
# Ausgabe: (numeric(1)): Wert der Qualitaetsfunktion Q an der Stelle estim
guete <- function(estim, true) sum((true - estim)^2)
```

```
# guete_values: Funktion zur Berechnung von mehreren Werten der Qualitaetsfunktion
# Eingabe: c (numeric): Vektor mit sequentiell hinzuzufuegenden Werten
#          fun (function): Funktion, mit der die geschaetzten Werte berechnet
#                          werden sollen
#          dat (list): Liste mit gleichlangen numerischen Vektoren, die die
#                      Datenpunkte beschreiben
#          true (numeric): Vektor mit den wahren Parametern
# Ausgabe: eine benannte Liste mit den Eintraegen:
#          par (numeric): Die Werte von c
#          value (numeric): Die zugehoerigen Werte der Qualitaetsfunktion
guete_values <- function(c, fun, dat = data, true = c(4, 1)){
  res <- sapply(c, function(x){
    dat[[length(dat) + 1]] <- rep(x, length(dat[[1]]))
    estim <- fun(dat)
    return(guete(estim, true))
  })
  return(list(par = c, value = res))
}
```

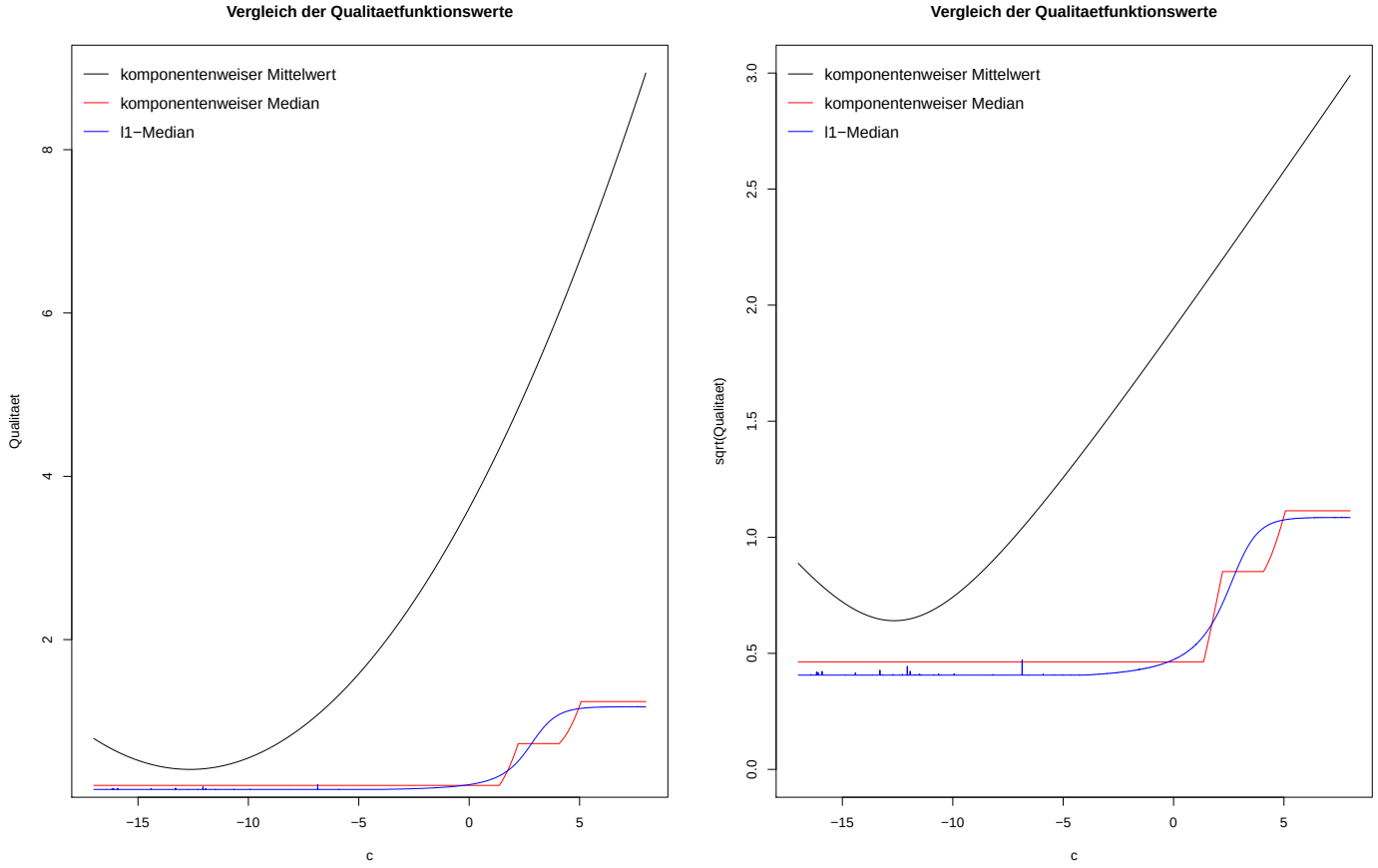
Erstelle nun Grafiken zur Veranschaulichung:



Auf der linken Grafik sind die Werte der Qualitätsfunktion für die drei Kenngrößen in Abhängigkeit des $c \in [-10, 10]$ abgetragen. Die rechte Grafik zeigt die gleichen Werte, allerdings unter einer Wurzel-Transformation, um die unterschiedlichen Größenordnungen der Werte besser darstellen zu können. Die abgetragene Qualitätsfunktion misst den quadratischen Abstand zwischen den geschätzten Parametern und den wahren Parametern. Ein kleiner Wert der Qualitätsfunktion bedeutet also eine geringe „Verzerrung“ des Schätzers, während große Werte für große Abweichungen der Schätzer vom wahren Parametervektor stehen. Es ist in beiden Grafiken zu sehen, dass der komponentenweise Mittelwert bei betragsmäßig großem c größer wird und insgesamt

unbeschränkt ist. Die Werte der Qualitätsfunktion bilden dabei ungefähr eine Parabel, was logisch erscheint, da der eindimensionale Mittelwert eine lineare Verzerrung aufweist, wodurch der quadratische Abstand parabelförmig verläuft. Die beiden Mediane (komponentenweise und l_1) sind hingegen bei der Größe ihrer Werte der Qualitätsfunktion beschränkt und unterscheiden sich kaum.

Sinnvoll ist es den linken Bereich des Definitionsbereichs für c zu erweitern, hier z.B. bis $c \geq -17$.



So erkennt man die quadratische Struktur der Qualitätsfunktion des komponentenweisen Mittelwerts. Die Qualitätsfunktionen bezüglich der Mediane bleiben von einem hinzugefügten Wert mit sehr kleinem c kaum berührt.