Supplementary Materials for "Tree-based methods for individualized treatment rules"

Eric B. Laber[1] and Ying-Qi Zhao[2]

[1]Department of Statistics, North Carolina State University

[2]Department of Biostatistics and Medical Informatics, University of Wisconsin-Madison

April 20, 2015

Supplement A: Potential predictors for Nafazodone study

Variable	Brief description
HAMA	Total Hamilton anxiety score (HAM-A)
HAMD	Total Hamilton depression score (HAM-D)
MOOD	General mood cognition score (IDS-SR)
GENDER	Subject sex
AGE	Subject age
ANXSOM	Anxiety/somatic symptoms (HAM-D)
BMI	Body mass index
PHYFUN	Physical functioning score (MOS-36)
BODPAI	Bodily pain factor score (MOS-36)
SLEEPD	Sleep disturbance factor score (HAM-D)
SLEEPD1	Sleep score 1 (IDS-SR)
SLEEPD2	Sleep score 2 (IDS-SR)
RETARDS	Retardation score (HAM-D)
VITALI	Vitality score (MOS-36)
SOCFUN	Social functioning score (MOS-36)
IDSSR	IDS-SR depression total score (IDS-SR)
ROLFUN	Role functioning score (MOS-36)
ROLSCD	Role area score (social and leisure, SAS-SR)
ROLSCE (Role area score (extended family, SAS-SR)	
DSMGAFCD	Global assessment of function
TRAN	General health (MOS-36)
MENTAL	Mental health score (MOS-36)

Table 1: Predictors considered for tree-based decision rule for Nafazodone study. Variables are taken from the Hamilton depression inventory (HAM-D), the Hamilton anxiety inventory (HAM-A), the Inventory of Depression Symptomatology (IDS-SR), the Medical Outcomes Study (MOS-36), and the Social Adjustment Scale (SAS-SR).

Supplement B: Proofs

Proof of Lemma 1. The proof follows from application of the iterated-variance formula. In particular, note that

$$\begin{aligned}\mathrm{var}\left[\frac{\{Y-g(X)\}1_{\pi(X)=A}}{p\{\pi(X)|X\}}\right] = E\left[\frac{\mathrm{var}\,(Y|X,A)\,1_{\pi(X)=A}}{p\{\pi(X)|X\}}\right] \\ +\,\mathrm{var}\left[\frac{\{E(Y|X,\pi(X))-g(X)\}1_{\pi(X)=A}}{p\{\pi(X)|X\}}\right],\end{aligned}$$

which is minimized by taking $g(X)=E(Y|X,\pi(X))$. □

Proof of Lemma 2. Let f_0 be the density of X and f_1 be the density of Y conditional on (X,A). It can be seen that

$$\begin{aligned}E(\widehat{C}_g(\nu_{\pi,h})) &= E\left(\frac{\{Y-g(X)\}}{p(A|X)}\frac{1}{h}\kappa\left(\frac{f(A)-f\{\pi(X)\}}{h}\right)f'(A)\right)\\ &= E\left[E\left(\{Y-g(X)\}\frac{1}{h}\kappa\left(\frac{f(A)-f\{\pi(X)\}}{h}\right)f'(A)|A,X\right)\right]\\ &= \int_x\int_0^1 E\left(\{Y-g(x)\}\frac{1}{h}\kappa\Big(\frac{f(a)-f\{\pi(x)\}}{h}\Big)f'(a)\Big|A=a,X=x\right)da f_0(x)dx\\ &= \int_x\int_0^1 E\left(\{Y-g(x)\}\frac{1}{h}\kappa\Big(\frac{f(a)-f\{\pi(x)\}}{h}\Big)f'(a)\Big|f(A)=f(a),X=x\right)da f_0(x)dx.\end{aligned}$$

Using the change of variables $u=(f(a)-f\{\pi(x)\})/h$, the above quantity equals

$$\begin{aligned}&\int_x\int_{-\infty}^{\infty}\kappa(u)[E\left\{Y-g(x)|X=x,f(A)=f\{\pi(x)\}+hu\right\}]du f_0(x)dx\\ =\;&\int_0^{\infty}\int_x\int_{-\infty}^{\infty}\kappa(u)\Big(E[\{Y-g(x)\}|X=x,f(A)=f\{\pi(x)\}]\\ &+\frac{\partial}{\partial a}E[\{Y-g(x)\}|X=x,f(A)=f(a)]\Big|_{a=\pi(x)}hu\\ &+\;\frac{1}{2}\frac{\partial^2}{\partial a^2}E[\{Y-g(x)\}|X=x,f(A)=f(a)]\Big|_{a=\pi(x)}h^2u^2+o(h^2)\Big)du f_0(x)dx,\end{aligned}$$

which can be seen to equal

$$
\begin{aligned}
&= \int_x E\Big(\{Y-g(x)\}\int_{-\infty}^{\infty}\kappa(u)du|X=x,A=\pi(x)\Big)f_0(x)dx \\
&\quad +\int_{-\infty}^{\infty}u^2\kappa(u)du\cdot\frac{h^2}{2}E\Big\{E\Big(\frac{\partial^2}{\partial a^2}E[Y-g(X)|X,A=a]\Big|_{a=\pi(X)}\Big)\Big\}+o(h^2) \\
&= E[E\{Y-g(X)|X,A=\pi(X)\}] \\
&\quad +\int_{-\infty}^{\infty}u^2\kappa(u)du\cdot\frac{h^2}{2}E\Big\{E\Big(\frac{\partial^2}{\partial a^2}E[Y-g(X)|X,A=a]\Big|_{a=\pi(X)}\Big)\Big\}+O(h^4) \\
&= \int_{-\infty}^{\infty}u^2\kappa(u)du\cdot\frac{h^2}{2}E\Big\{E\Big(\frac{\partial^2}{\partial a^2}E[Y-g(X)|X,A=a]\Big|_{a=\pi(X)}\Big)\Big\}+O(h^4).
\end{aligned}
$$

The second last equality follows since $\kappa(u)$ is symmetric around 0, and $\int_{-\infty}^{\infty}u^m\kappa(u)du=0$ when m is odd. The last equality follows since $g(X)=E\{Y|X,A=\pi(X)\}$.

The variance of $\widehat{C}_g(\nu_{\pi,h})$ equals

$$
\begin{aligned}
&\frac{1}{n}E\left(\frac{\{Y-g(X)\}}{p(A|X)}\frac{1}{h}\kappa\left(\frac{f(A)-f\{\pi(X)\}}{h}\right)f'(A)\right)^2 \\
&-\frac{1}{n}\left[E\left(\frac{\{Y-g(X)\}}{p(A|X)}\frac{1}{h}\kappa\left(\frac{f(A)-f\{\pi(X)\}}{h}\right)f'(A)\right)\right]^2.
\end{aligned}
$$

According to the previous derivation on bias term, we can see that the second term is of the order $O(n^{-1})$. Using the change of variables, we obtain for the first term above

$$
\begin{aligned}
& E\left(\frac{\{Y-g(X)\}}{p(A|X)}\frac{1}{h}\kappa\left(\frac{A-\pi(X)}{h}\right)f'(A)\right)^2 \\
=\;& \frac{1}{h^2}\int_x\int_0^1 E\left(\frac{\{Y-g(x)\}^2}{p(a|x)}\kappa^2\Big(\frac{a-\pi(x)}{h}\Big)f'(a)^2|X=x,f(A)=f(a)\right)daf_0(x)dx \\
=\;& \frac{1}{h}\int_x\int_{-\infty}^{\infty}E\Big(\frac{(Y-g(x))^2}{p(f^{-1}(f\{\pi(x)\}+hu)|x)}\kappa^2(u)f'(f^{-1}(f\{\pi(x)\}+hu))\Big|X=x, \\
& \qquad f(A)=f\{\pi(x)\}+hu\Big)duf_0(x)dx
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{h}\int_x \int_{-\infty}^{\infty} E\Big(\frac{(Y-g(x))^2}{p(\pi(x)|x)}\kappa^2(u)f'\{\pi(x)\}\Big|X=x, f(A)=f\{\pi(x)\}\Big)du f_0(x)dx + O(1) \\
&= \frac{1}{h}E\left(E\Big[\frac{\{Y-g(X)\}^2}{p(\pi(X)|X)}f'\{\pi(X)\}|X, A=\pi(X)\Big]\right)\int_{-\infty}^{\infty}\kappa^2(u)du + O(1).
\end{aligned}
$$

Therefore, $var(\widehat{C}_g(\nu_{\pi,h})) = O(1/nh)$. □

Supplement C: Details for the tree-growing algorithm

In this section, we describe our algorithm for estimating optimal tree-based decision rules. We term our algorithm Minimum impurity Decision Assignments (MIDAs). Like most tree estimation algorithms, MIDAs is greedy, however, our framework allows for global search algorithms, e.g., simulated annealing, as well (see Remark 1). It will be useful to represent a decision tree with as a matrix. For a tree with K nodes we number the root node as 1 and arbitrarily label the remaining $K-1$ nodes as $2,\ldots,K$. For any node, say $\mathcal{N}$, we associate a four-tuple (r, i_ℓ, i_r, a) where: r is the rectangle describing the split at node $\mathcal{N}$; $i_\ell \in \{2,\ldots,K-1\}$ indexes the left-node of $\mathcal{N}$; $i_r \in \{2,\ldots,K-1\}$ indexes the right-node of $\mathcal{N}$; and $a \in \mathcal{A}$ is the recommended treatment at node $\mathcal{N}$. Non-terminal nodes will have values for first three components, namely r, i_ℓ, i_r, and be missing a value for a; conversely, terminal nodes will be missing the first three components and only have a value for a. We code missing values as NA. A matrix coding of a decision tree with K nodes is $K \times 4$ with the kth row storing the four-tuple for the node k. The tree in Figure 1 is represented in matrix coding as

$$\begin{pmatrix} (1,1,1) & 3 & 2 & \texttt{NA} \\ \texttt{NA} & \texttt{NA} & \texttt{NA} & 1 \\ (2,-3,-1) & 4 & 5 & \texttt{NA} \\ \texttt{NA} & \texttt{NA} & \texttt{NA} & 0 \\ \texttt{NA} & \texttt{NA} & \texttt{NA} & 1 \end{pmatrix} \Leftrightarrow \begin{array}{ccccc} \text{node} & \text{rectangle} & i_\ell & i_r & a \\ 1 & (1,1,1) & 3 & 2 & \texttt{NA} \\ 2 & \texttt{NA} & \texttt{NA} & \texttt{NA} & 1 \\ 3 & (2,-3,-1) & 4 & 5 & \texttt{NA} \\ 4 & \texttt{NA} & \texttt{NA} & \texttt{NA} & 0 \\ 5 & \texttt{NA} & \texttt{NA} & \texttt{NA} & 1. \end{array}$$

Matrix notation offers a compact, platform/programming-language independent, representation of a decision tree; it is also useful for implementing global search methods (see Remark 1).

Algorithm 1 gives the MIDAs algorithm for the discrete treatment case, though the extension to the continuous case is obvious. The description of the algorithm assumes a basic understanding of stacks, for an introduction see [Drozdek, 2005]. This algorithm produces a stack of 5-tuples, the first entry of each tuple is the row in the matrix representation of the tree and the last four entries in each tuple correspond to the four columns in the tree representation. An implementation of the MIDAs algorithm, for both continuous and discrete treatments is provided in the supplemental material. The algorithm requires both a value of $\lambda > 0$ and an estimator $\hat{m}$ of m as input. An estimator of $m(x) = E(Y|X = x, A = \pi^{\text{opt}}(x))$ is constructed as follows. First, construct estimate $\widehat{Q}(x,a)$ of $Q(x,a) = E(Y|X = x, A = a)$ using a flexible regression model; in our simulations we use random forests [Breiman, 2001]. Then, using this estimator compute $\hat{m}(x) = \sup_{a\in\mathcal{A}} \widehat{Q}(x,a)$. Given an estimator $\hat{m}$ of m, the tuning parameter λ is chosen using cross-validation with objective $\widehat{C}(\pi)$. Specifically, if $\hat{\pi}^\lambda$ denotes the decision rule learned using tuning parameter λ then let $\widehat{C}^{\text{cv}}(\hat{\pi}^\lambda)$ denote the cross-validated estimate of $C(\hat{\pi}^\lambda)$; the chosen tuning parameter is thereby $\hat{\lambda} = \arg\max_\lambda \widehat{C}^{\text{cv}}(\hat{\pi}^\lambda)$.

Remark 1. The MIDAs algorithm, like most tree-based regression/classification algorithms, is greedy in the sense that it chooses splits myopically to produce the greatest increase in purity. However, this procedure need not produce a global maximum. An alternative approach would be to improve the solution produced by MIDAs using simulated annealing or another stochastic search algorithm. One approach for generating proposal trees for use with stochastic search algorithms is to randomly: (i) change the rectangle in a non-terminal node; (ii) prune two terminal nodes; or (iii) split a non-terminal node. Note that the matrix representation makes performing these operations particularly easy.

Algorithm 1: MIDAs

```
Input $\lambda > 0$, $\hat{m}_{r,a,a'} : \mathbb{R}^p \to \mathbb{R}$ for any rectangle $r$ and $a$, $a' \in \mathcal{A}$.
initialize regionStack, treeStack as empty stacks, rowNumber $= 1$
push $\mathbb{R}^p$ into regionStack
while regionStack is not empty do
    Set $\mathcal{R} =$ pop regionStack
    if $\sum_{i=1}^{n} 1_{X_i \in \mathcal{R}} < 2\mu$ then
        push (rowNumber, NA, $\arg\max_a \frac{\mathbb{P}_n 1_{X \in \mathcal{R}} 1_{A=a} Y}{\mathbb{P}_n 1_{X \in \mathcal{R}} 1_{A=a}}$, NA, NA) into treeStack
        rowNumber $=$ rowNumber $+1$
    end
    else
        $\hat{r} = \arg\max_r \{\mathcal{P}(\mathcal{R}, r) \,:\, \min(n\mathbb{P}_n 1_{X \in \mathcal{R} \cap r},\, n\mathbb{P}_n 1_{X \in \mathcal{R} \cap r^c}) \geq \mu\}$
        if $\mathcal{P}(\mathcal{R}, \hat{r}) \geq \mathcal{P}(\mathcal{R}) + \lambda$ then
            push (rowNumber, $\hat{r}$, NA, rowNumber $+ 1$, rowNumber $+ 2$) into treeStack
            push $\mathcal{R} \cap r$ into regionStack
            push $\mathcal{R} \cap r^c$ into regionStack
            rowNumber $=$ rowNumber $+2$
        end
        else
            push (rowNumber, NA, $\arg\max_a \frac{\mathbb{P}_n 1_{X \in \mathcal{R}} 1_{A=a} Y}{\mathbb{P}_n 1_{X \in \mathcal{R}} 1_{A=a}}$, NA, NA) into treeStack
            rowNumber $=$ rowNumber $+ 1$
        end
    end
end
```

Residual diagnostic plots

In this section we present residual diagnostic plots for the ridge regression model used to build a linear decision rule for the binary treatment setting when $p = 25$, $n = 250$. The plug-in estimator of the R^2 in each case was between 0.70 and 0.80 and the residual diagnostic plots do not indicate significant deviations from the underlying assumptions.

Example one

Example two

Example three

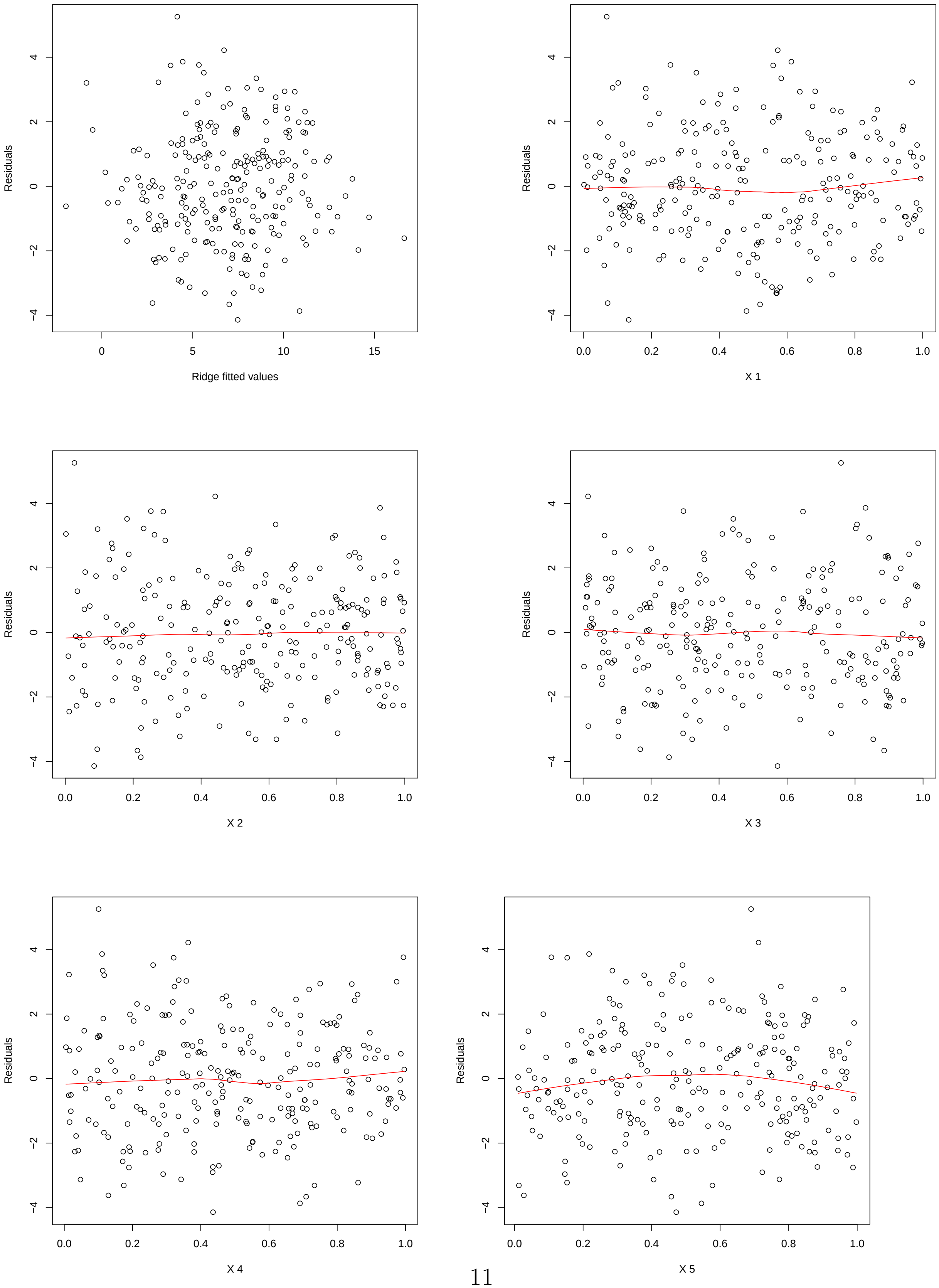

Figure 1: Residual vs. fitted and residual vs. predictor plots for $X_1 - X_5$ in (DM1).

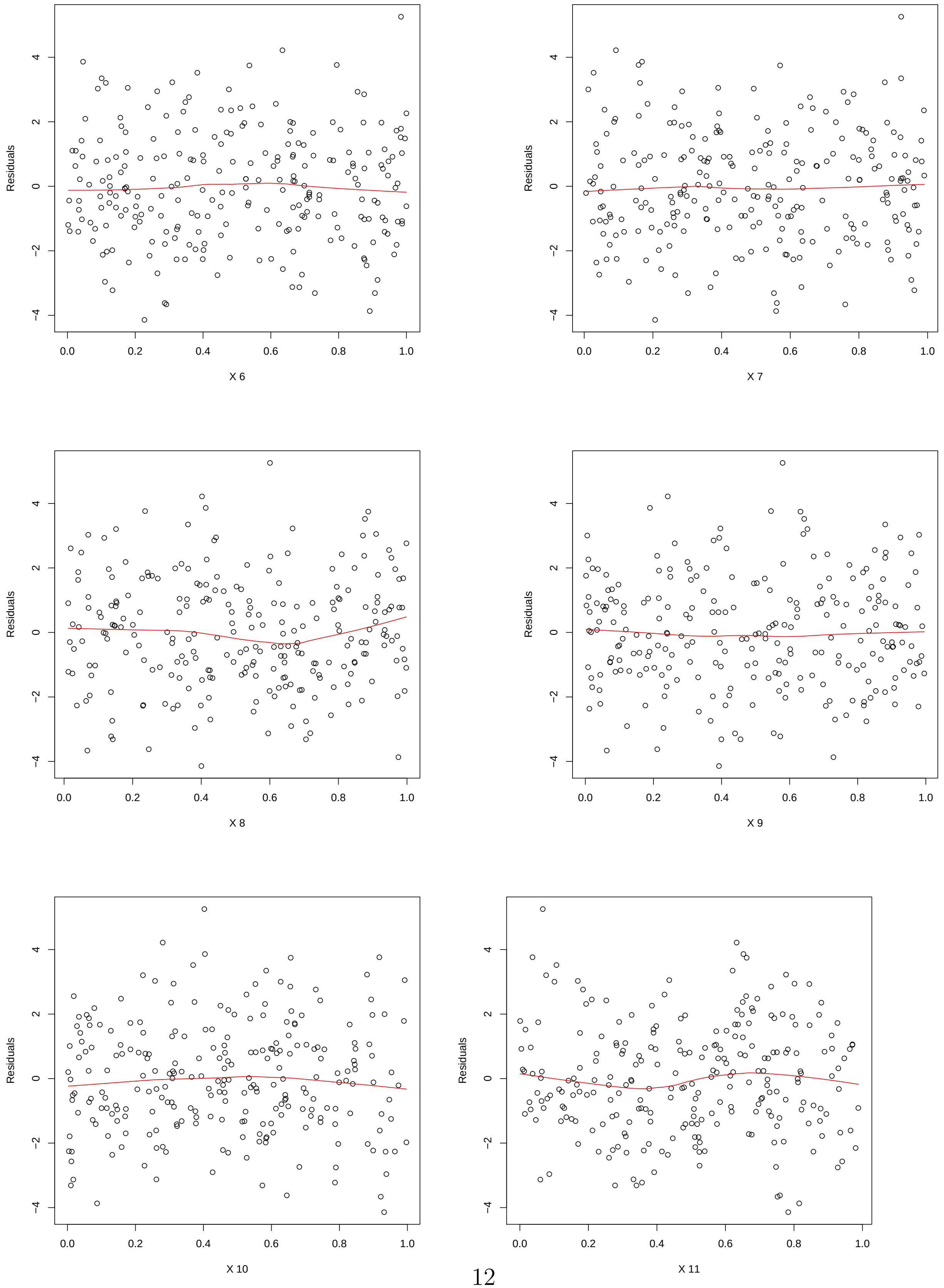

Figure 2: Residual vs. predictor plots for $X_6 - X_{11}$ in (DM1).

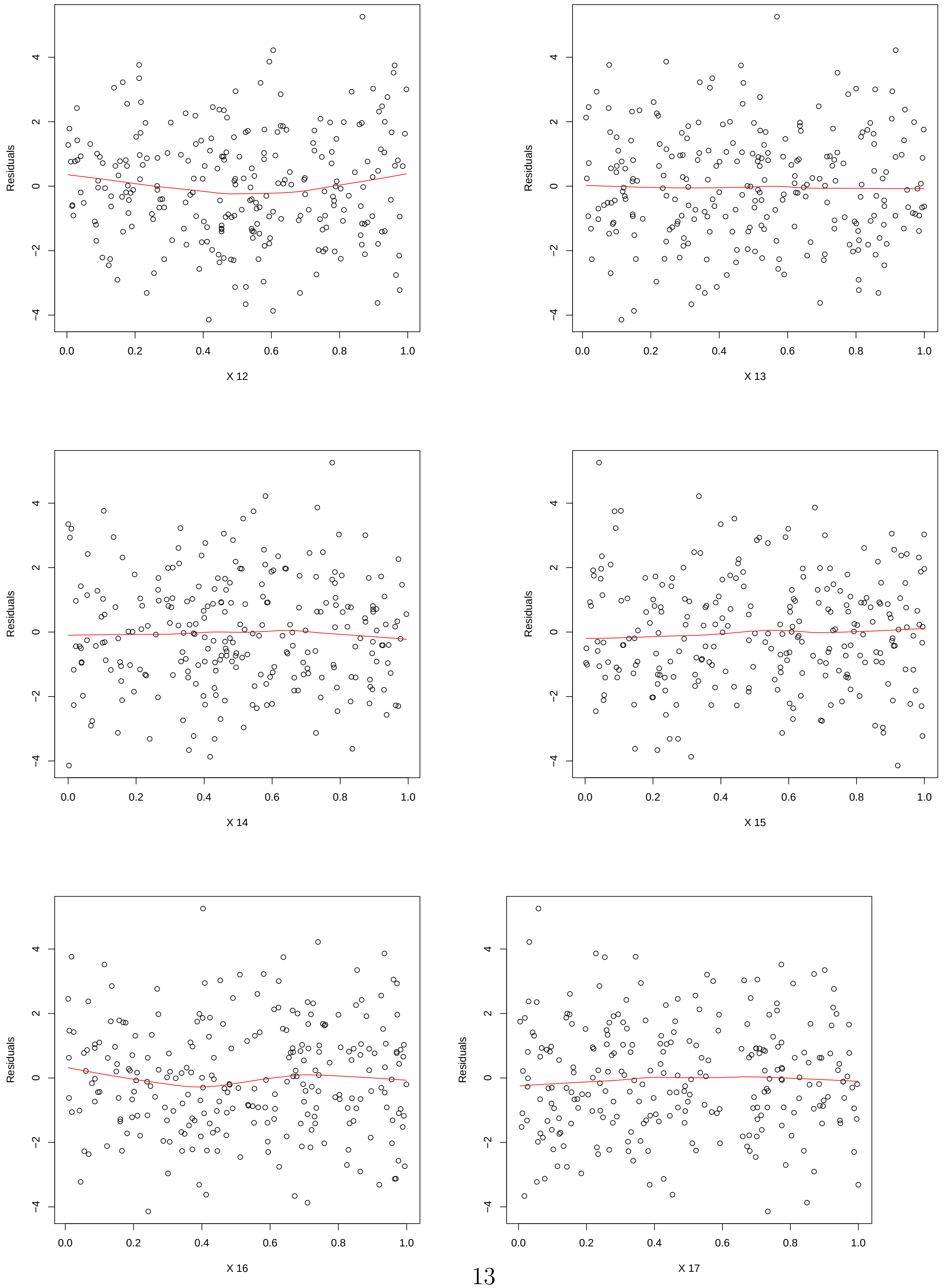

Figure 3: Residual vs. predictor plots for $X_{12} - X_{17}$ n (DM1).

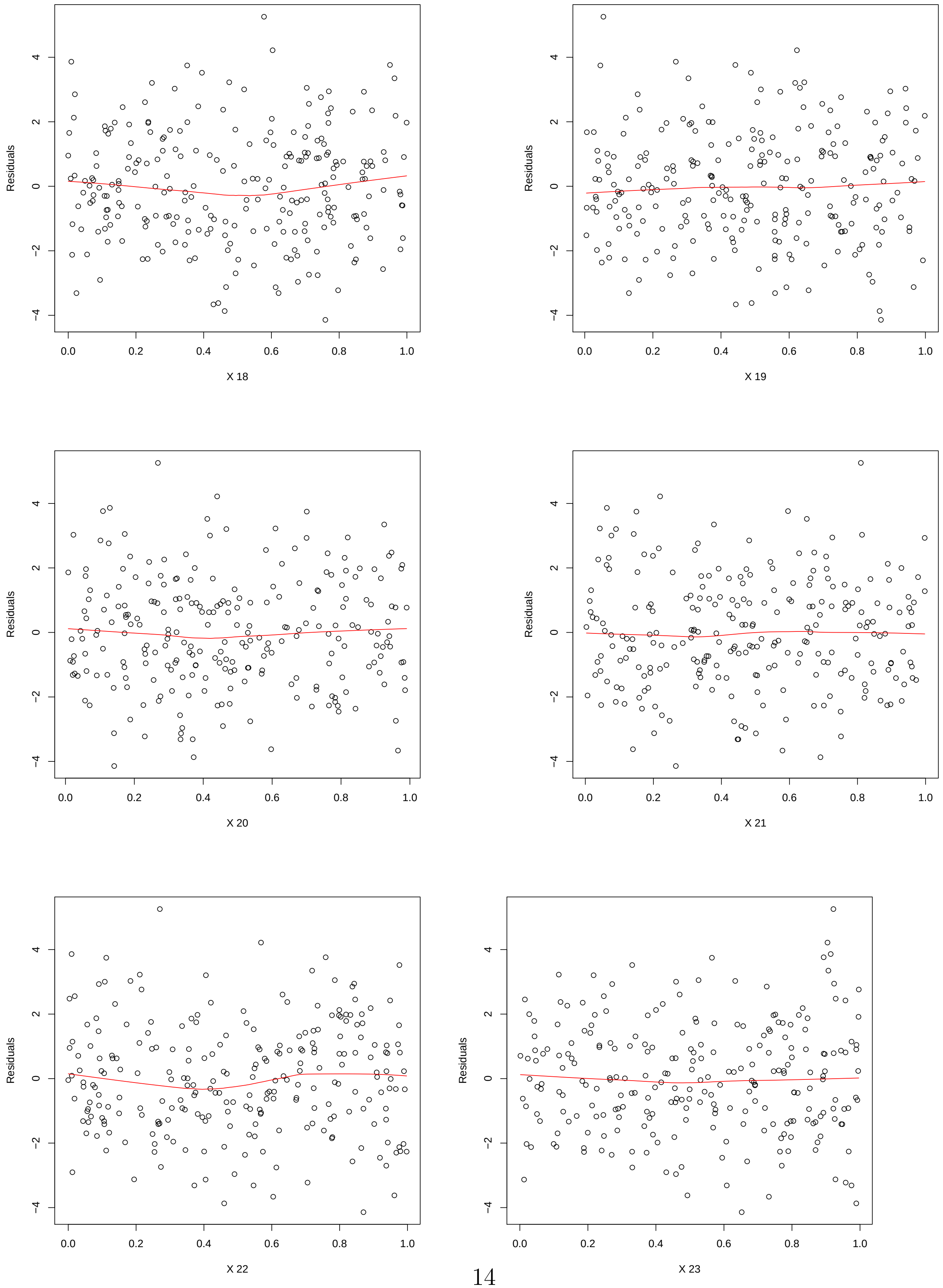

Figure 4: Residual vs. predictor plots for $X_{18} - X_{23}$ n (DM1).

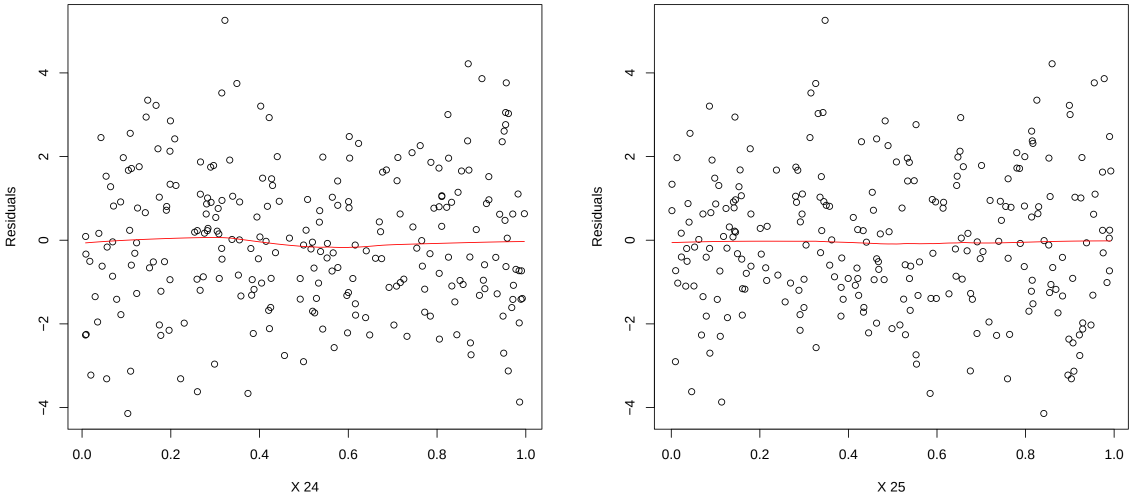

Figure 5: Residual vs. predictor plots for $X_{24} - X_{25}$ n (DM1).

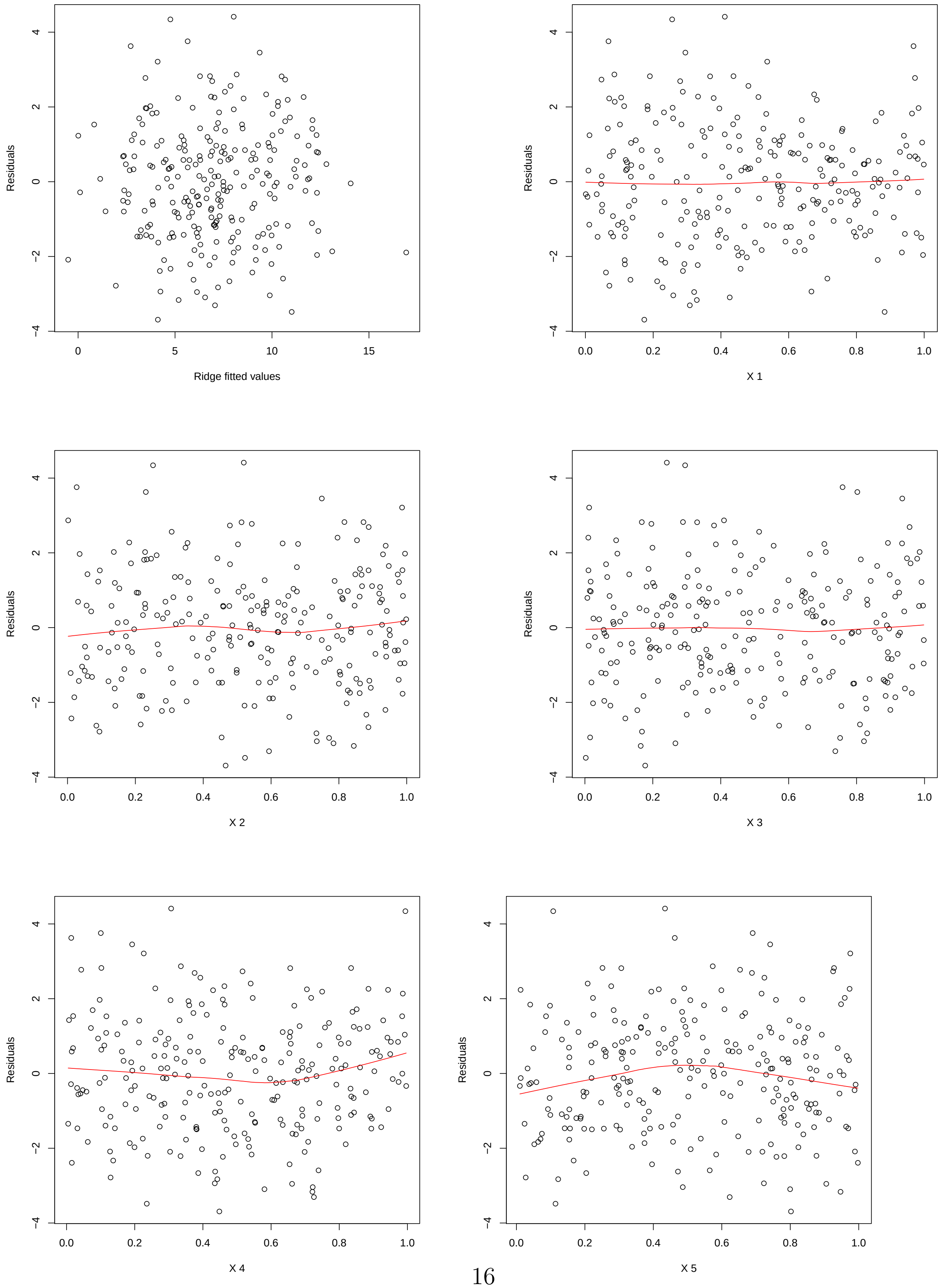

Figure 6: Residual vs. fitted and residual vs. predictor plots for $X_1 - X_5$ in (DM2).

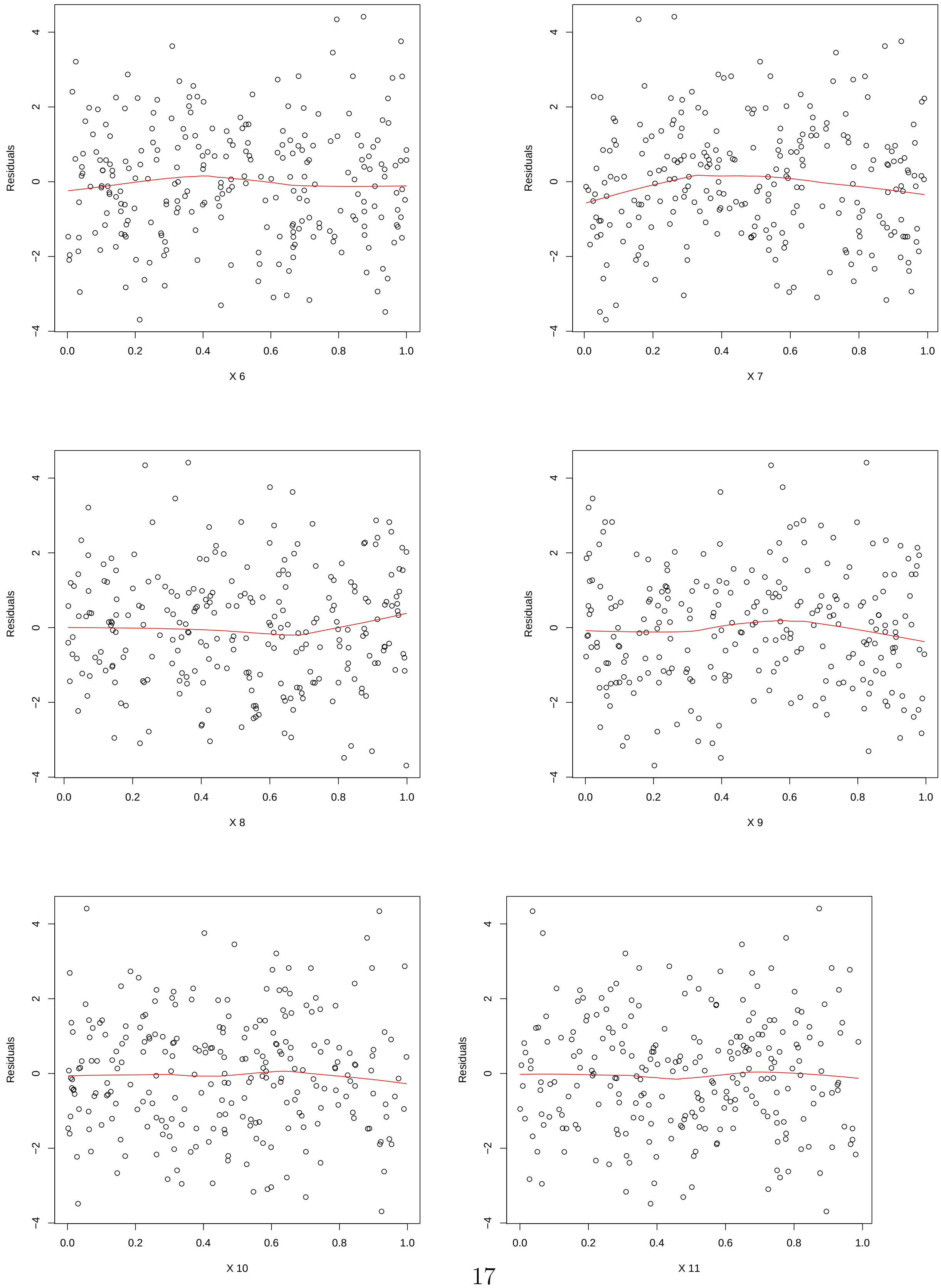

Figure 7: Residual vs. predictor plots for $X_6 - X_{11}$ n (DM2).

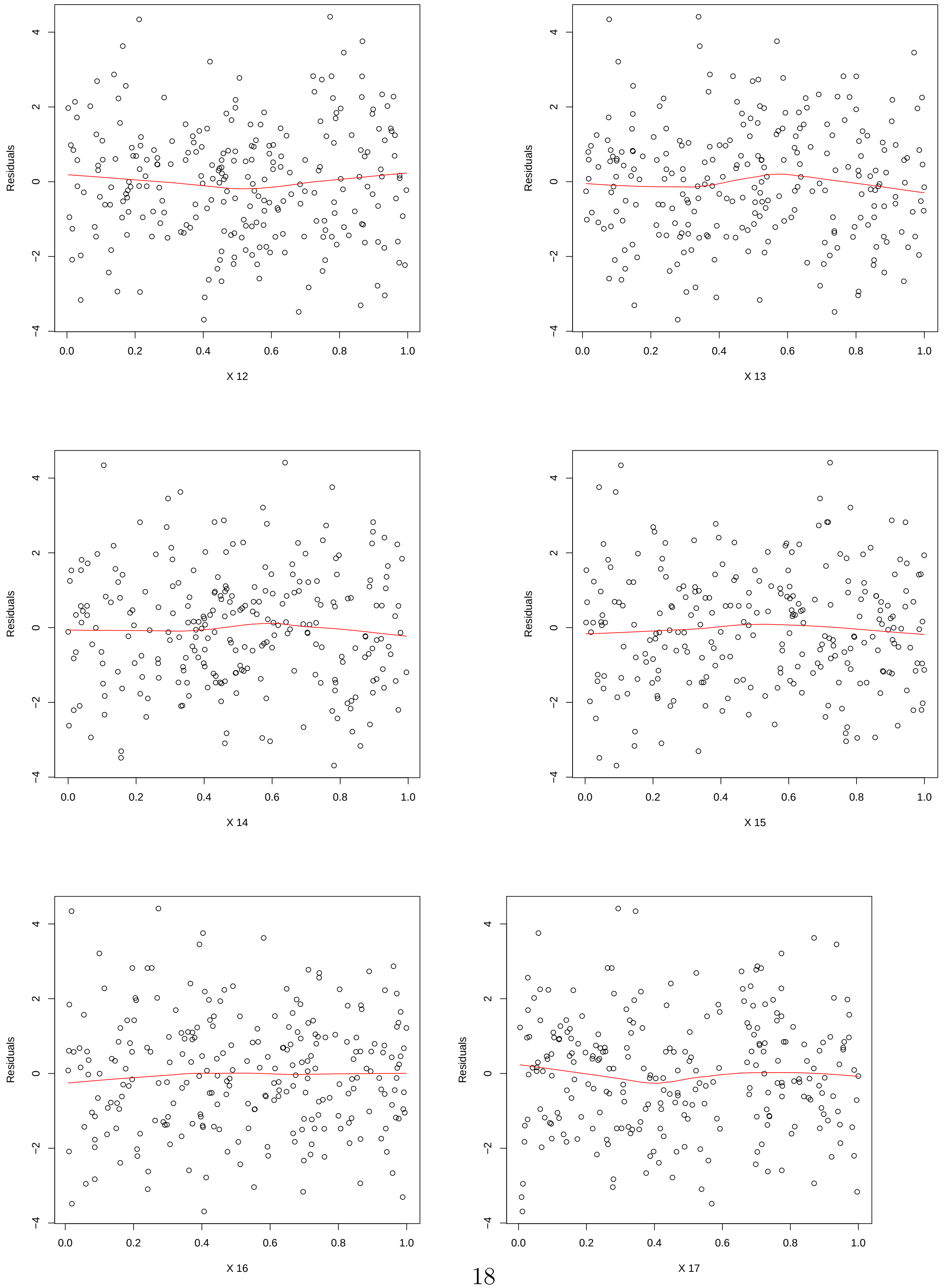

Figure 8: Residual vs. predictor plots for $X_{12} - X_{17}$ n (DM2).

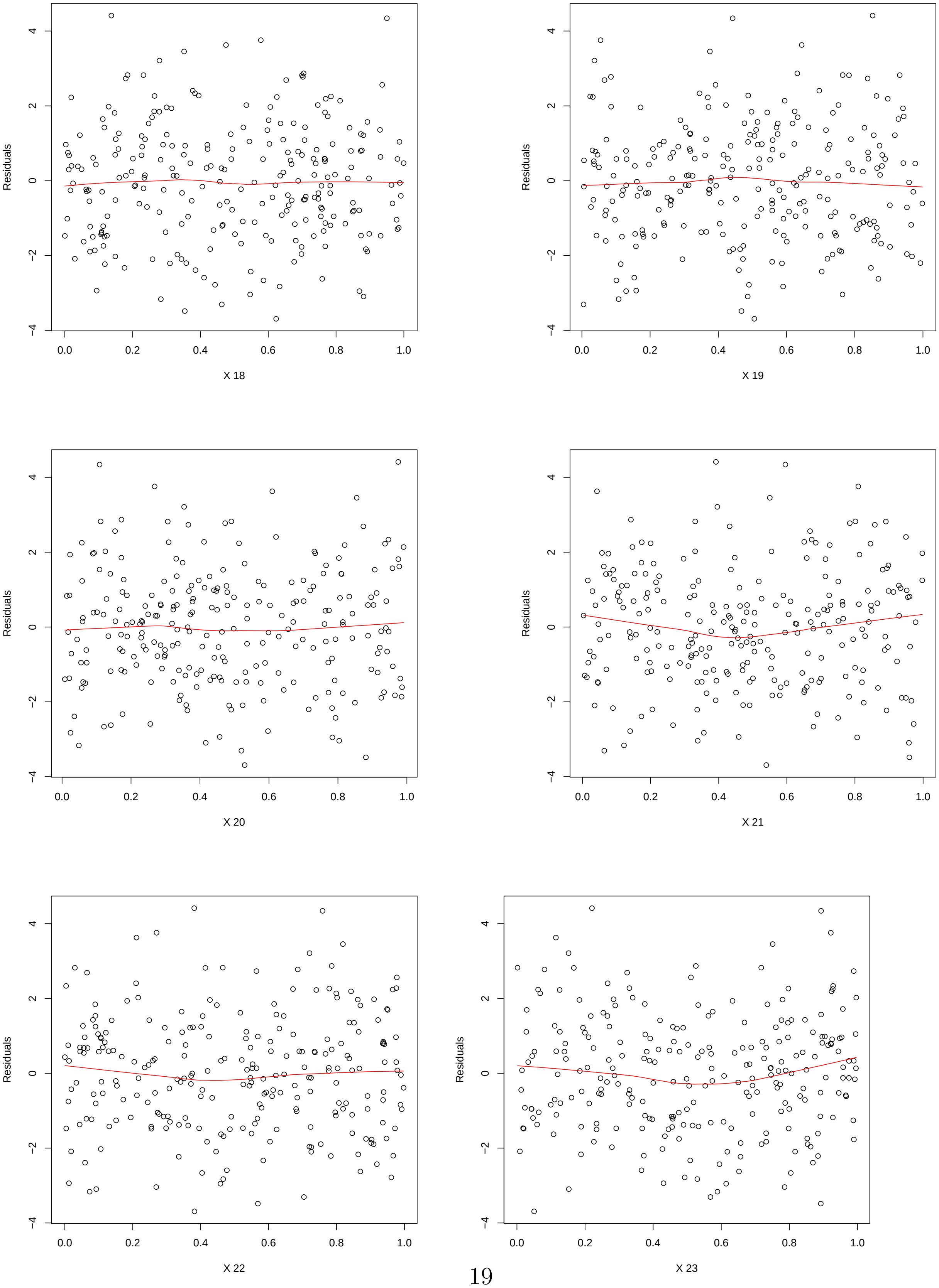

Figure 9: Residual vs. predictor plots for $X_{18} - X_{23}$ n (DM2).

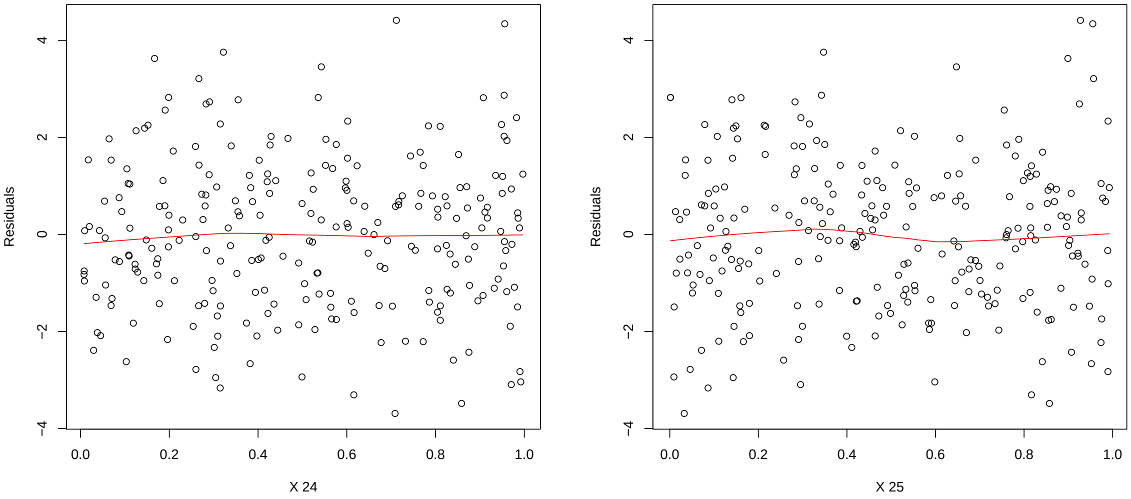

Figure 10: Residual vs. predictor plots for $X_{24} - X_{25}$ n (DM2).

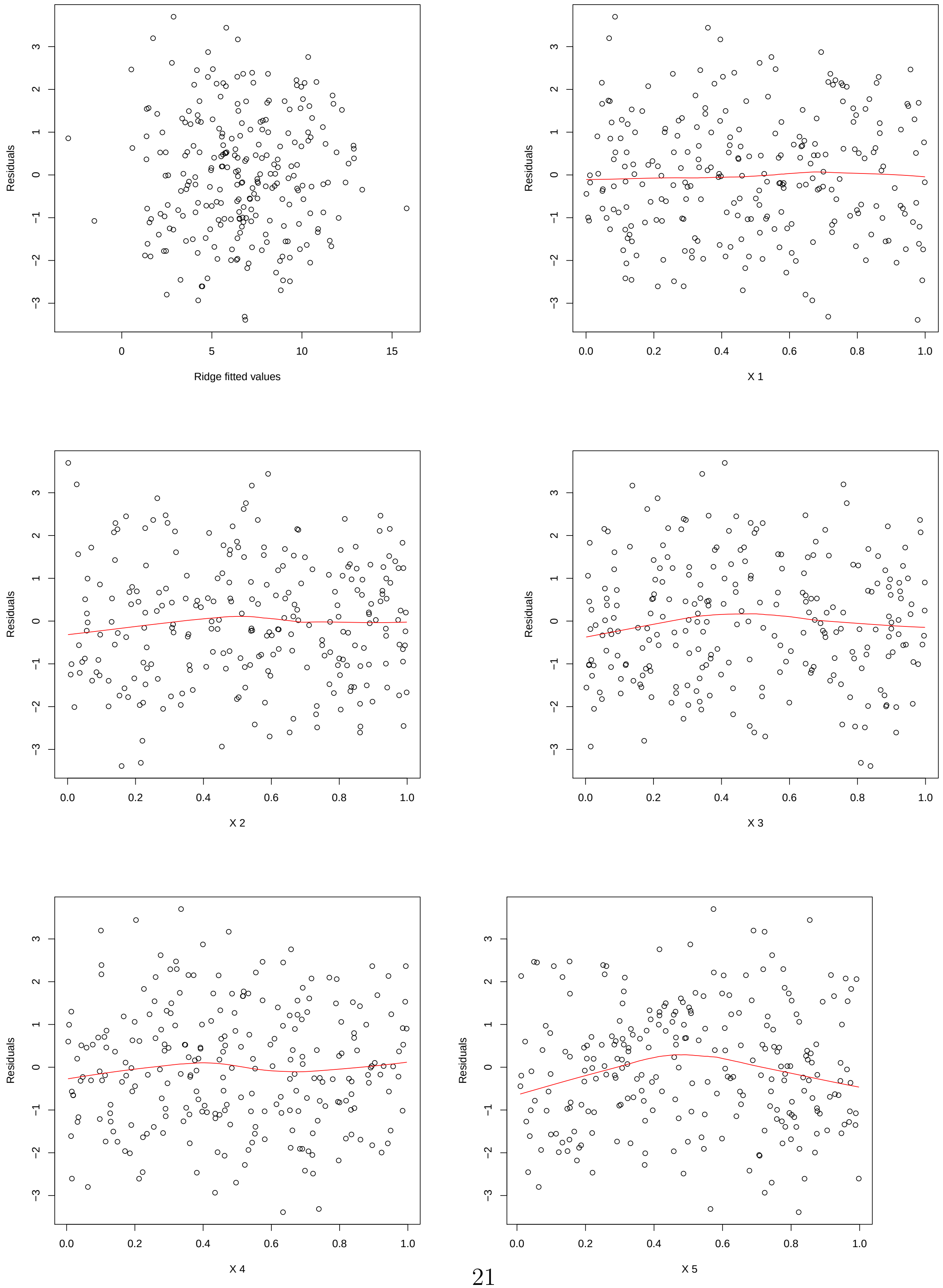

Figure 11: Residual vs. fitted and residual vs. predictor plots for $X_1 - X_5$ in (DM3).

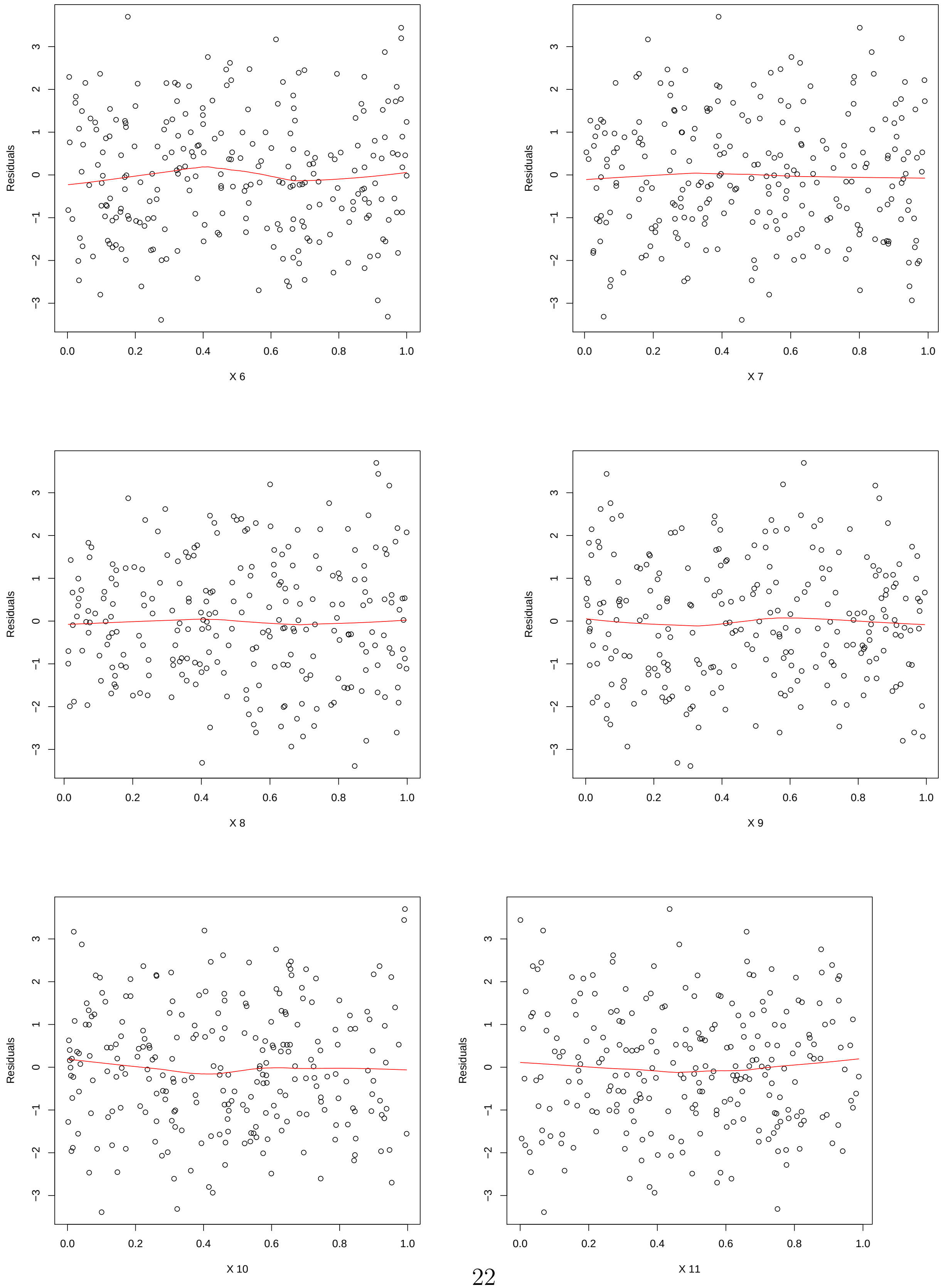

Figure 12: Residual vs. predictor plots for $X_6 - X_{11}$ n (DM3).

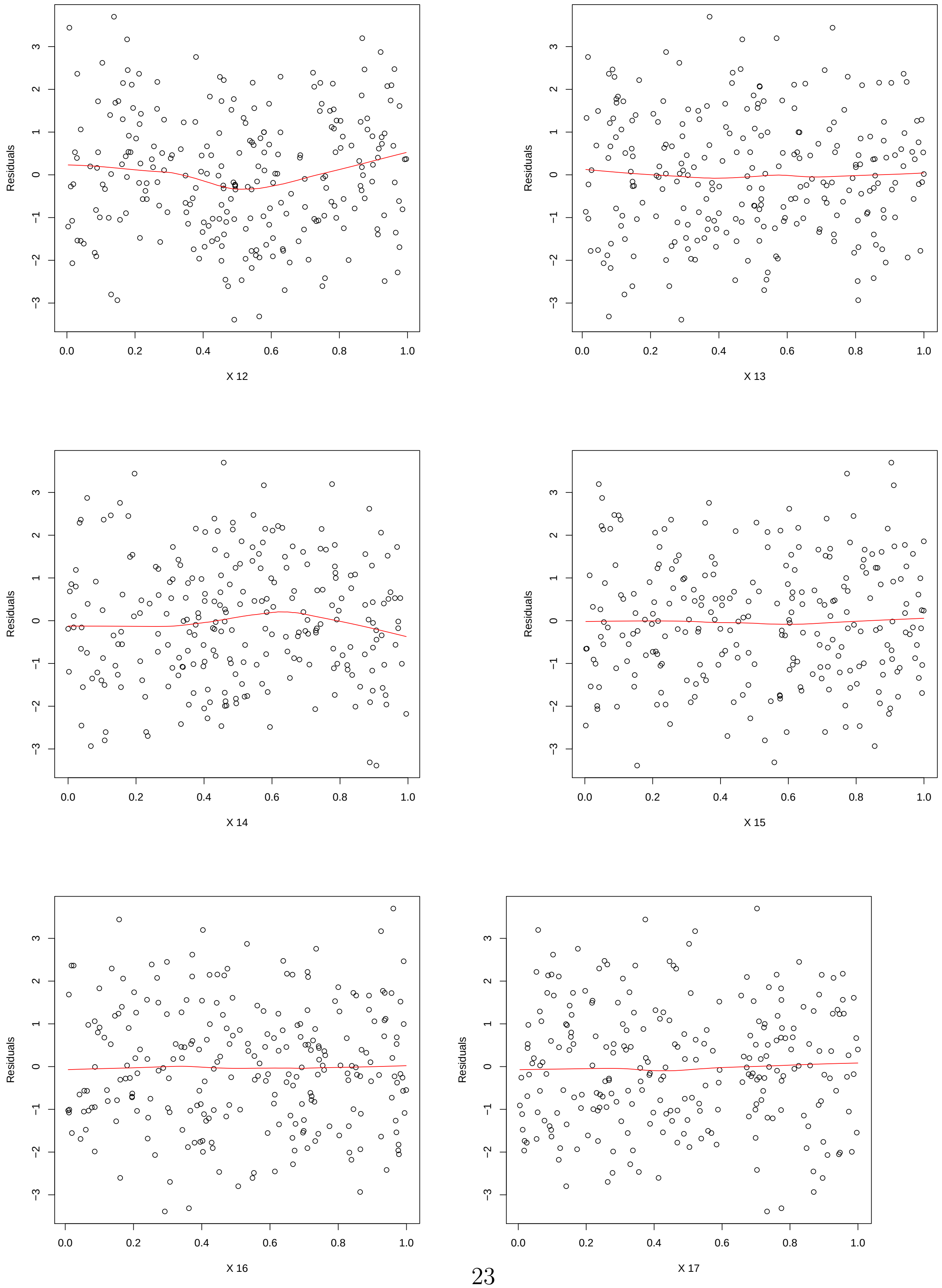

Figure 13: Residual vs. predictor plots for $X_{12} - X_{17}$ n (DM3).

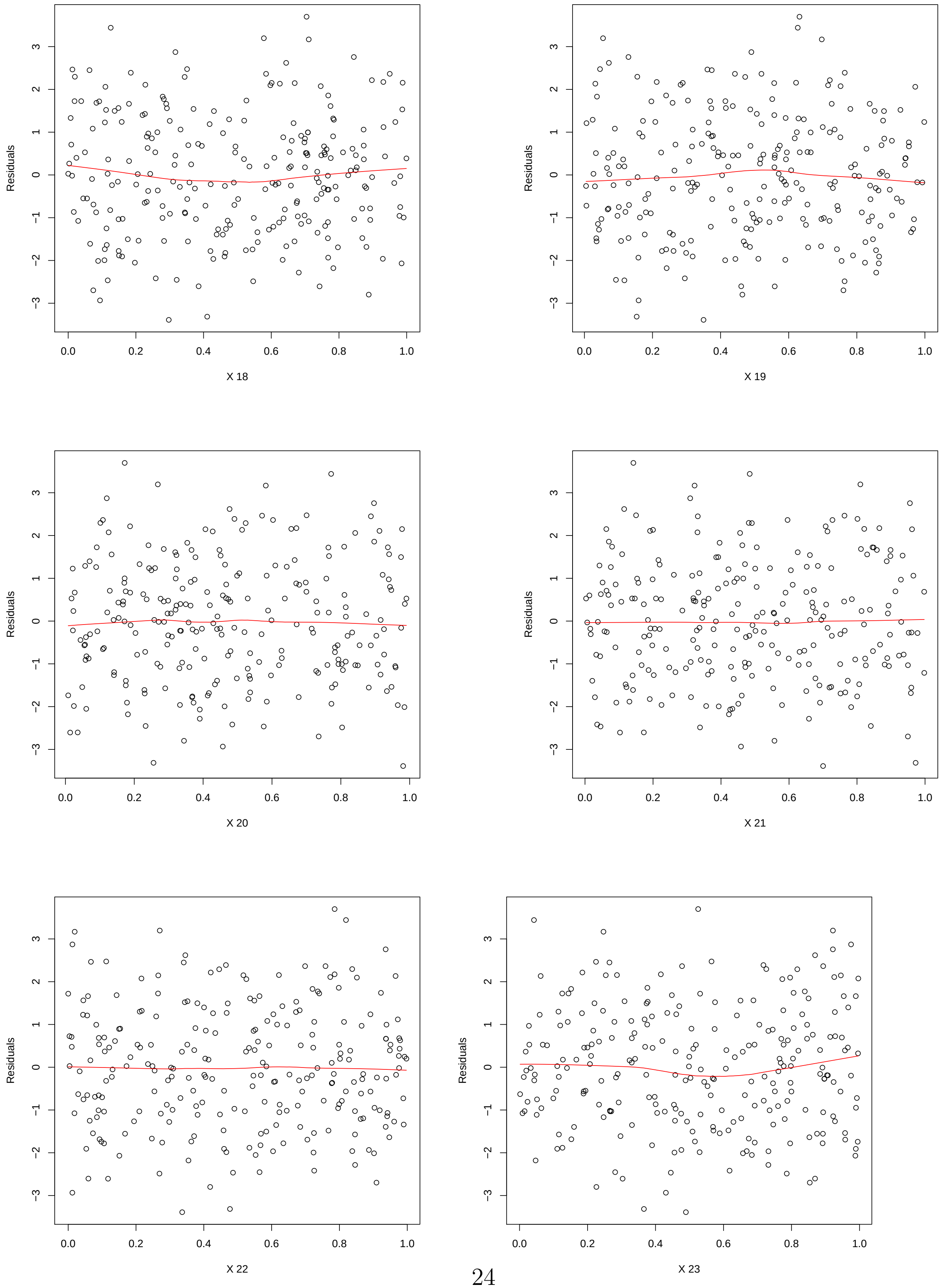

Figure 14: Residual vs. predictor plots for $X_{18} - X_{23}$ n (DM3).

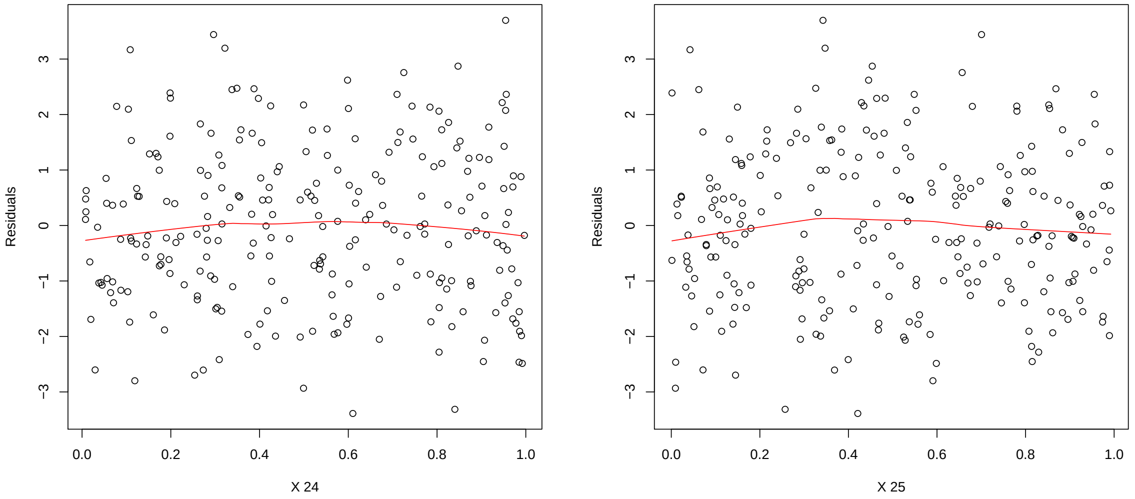

Figure 15: Residual vs. predictor plots for $X_{24} - X_{25}$ n (DM3).

Estimated tree plots

Here we show the treatment rule estimated by MIDAs averaged over 150 training sets in the $p = 25$ case. The case (CM1) was shown in the main body and is therefore omitted.

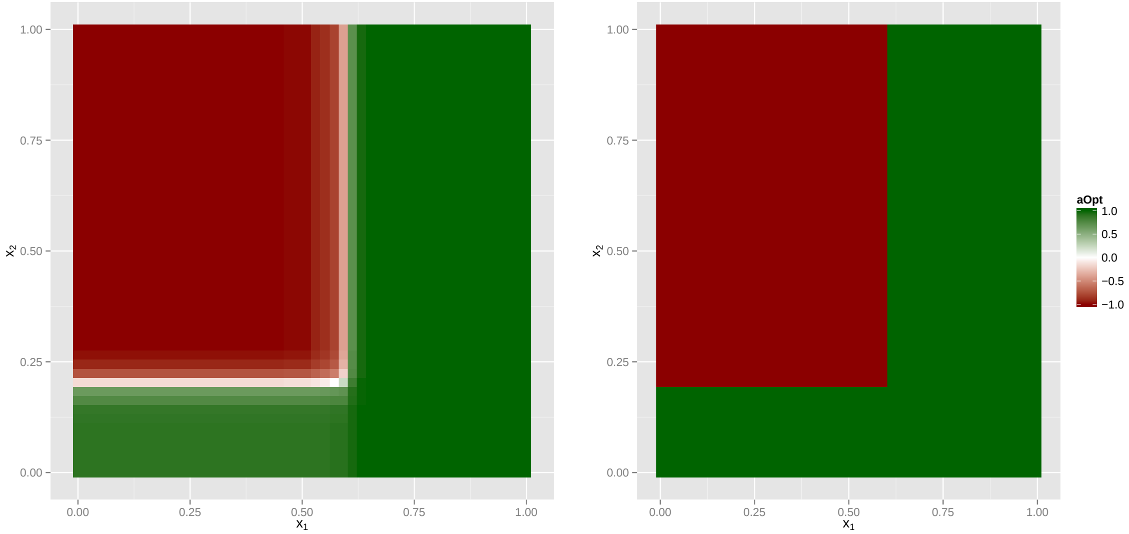

Figure 16: **Left:** average of MIDAs estimated optimal rule for (DM1). **Right:** true optimal rule for (DM1).

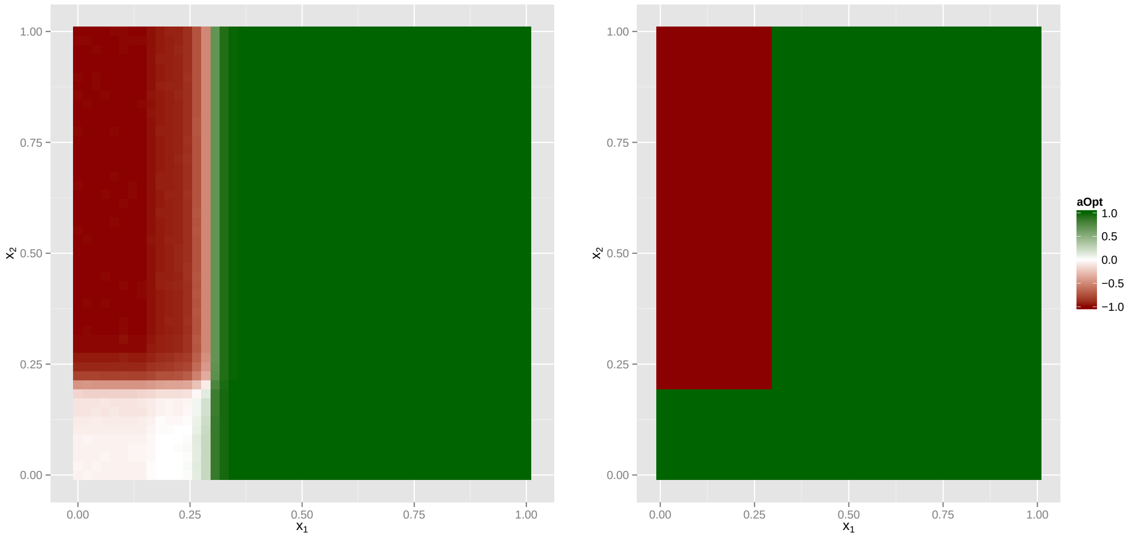

Figure 17: **Left:** average of MIDAs estimated optimal rule for (DM2). **Right:** true optimal rule for (DM2).

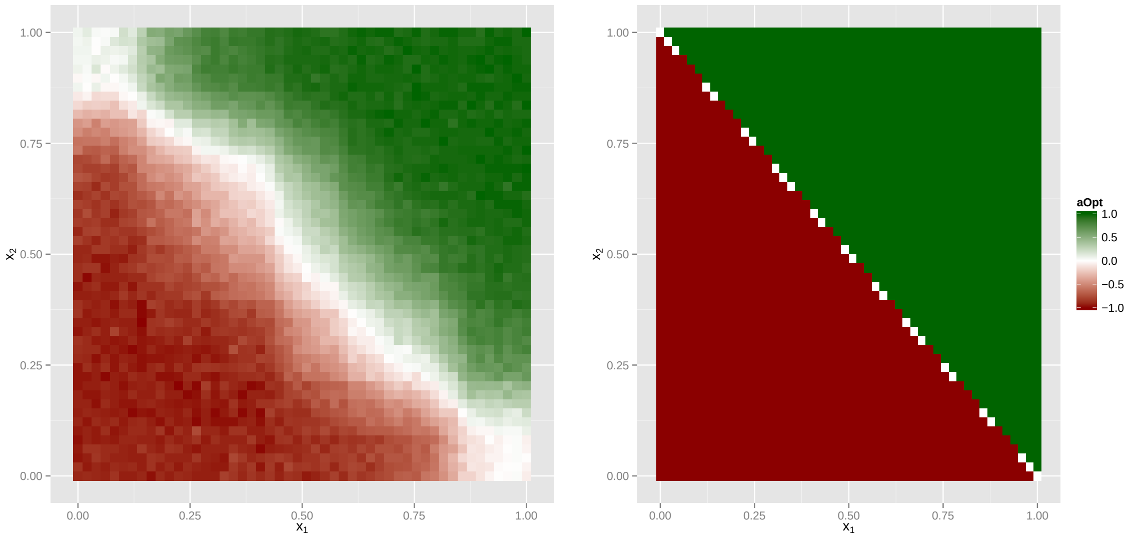

Figure 18: **Left:** average of MIDAs estimated optimal rule for (DM3). **Right:** true optimal rule for (DM3).

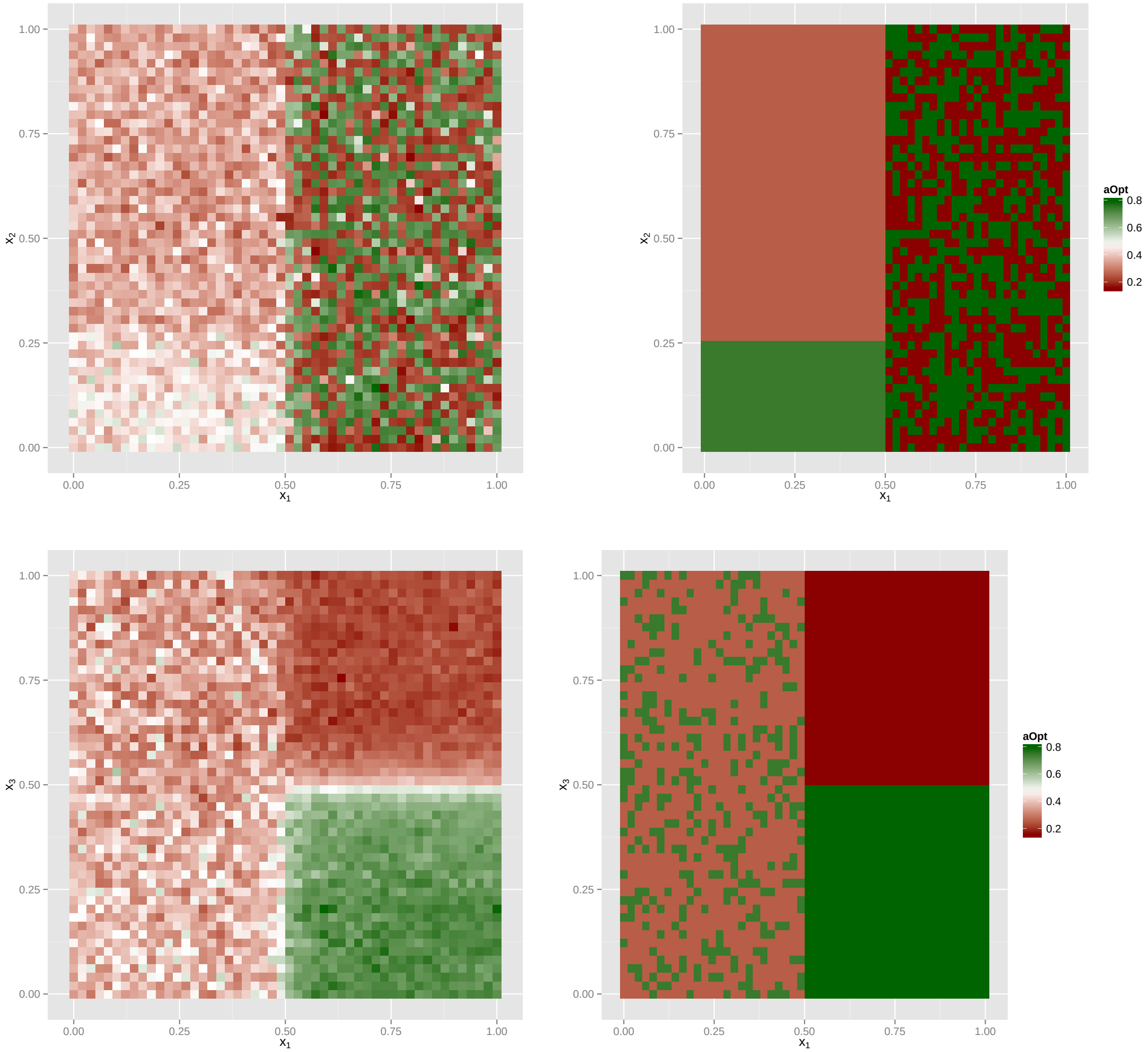

Figure 19: **Top left:** average of MIDAs estimated optimal rule for (CM2) as a function of x_1 and x_2. **Top right:** true optimal rule for (CM2) as a function of x_1 and x_2. **Bottom left:** average of MIDAs estimated optimal rule for (CM2) as a function of x_1 and x_3. **Bottom right:** true optimal rule for (CM2) as a function of x_1 and x_3.

References

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Adam Drozdek. *Data structures and algorithms in C++*. CengageBrain. com, 2005.