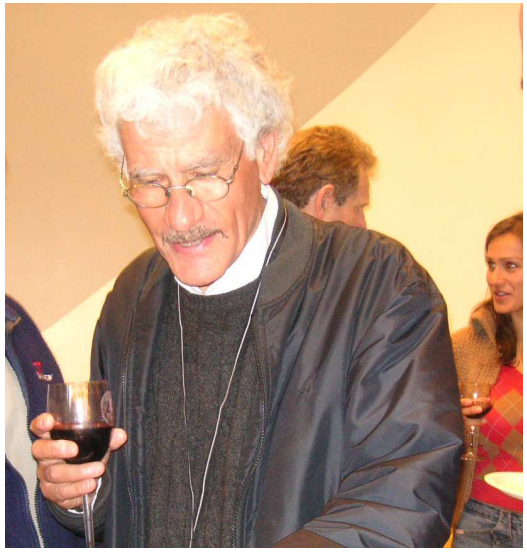
Sparse Linear Models

Trevor Hastie

Stanford University

joint work with Jerome Friedman, Rob Tibshirani and many students



Linear Models for Wide Data

As datasets grow *wide*—i.e. many more features than samples—the linear model has regained favor as the tool of choice.

Document classification: bag-of-words easily leads to $p = 20K$ features and $N = 5K$ document samples. Much more if bigrams, trigrams etc, or documents from Facebook, Google, Yahoo!

Genomics, microarray studies: $p = 40K$ genes are measured for each of $N = 300$ subjects.

Genome-wide association studies: $p =$1–2M SNPs measured for $N = 2000$ case-control subjects.

In examples like these we tend to use linear models — e.g. linear regression, logistic regression, Cox model. Since $p \gg N$, we cannot fit these models using standard approaches.

Forms of Regularization

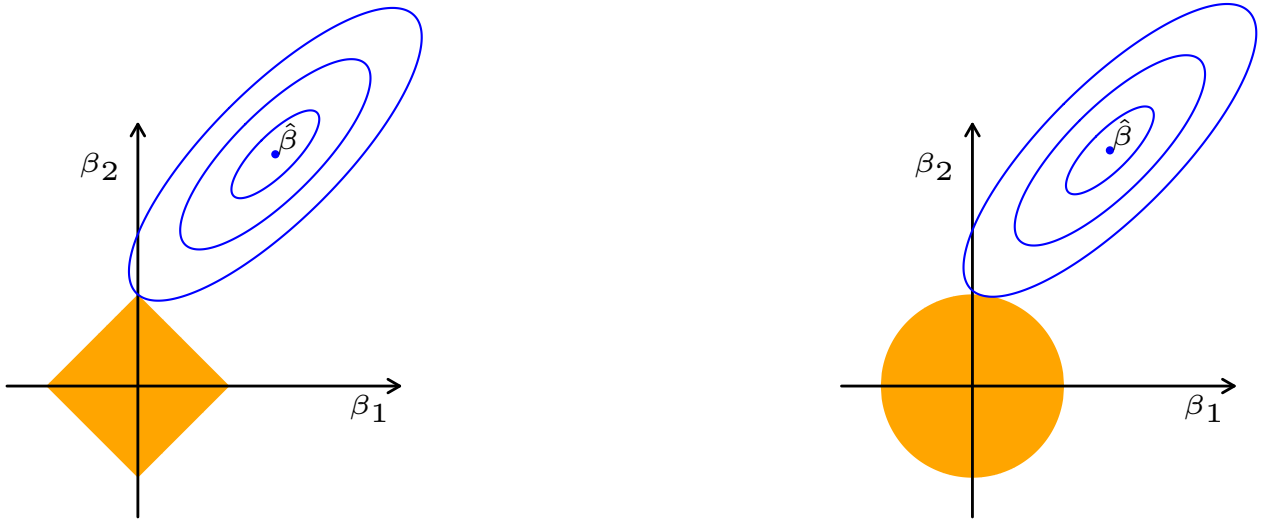
We cannot fit linear models with $p > N$ without some constraints. Common approaches are

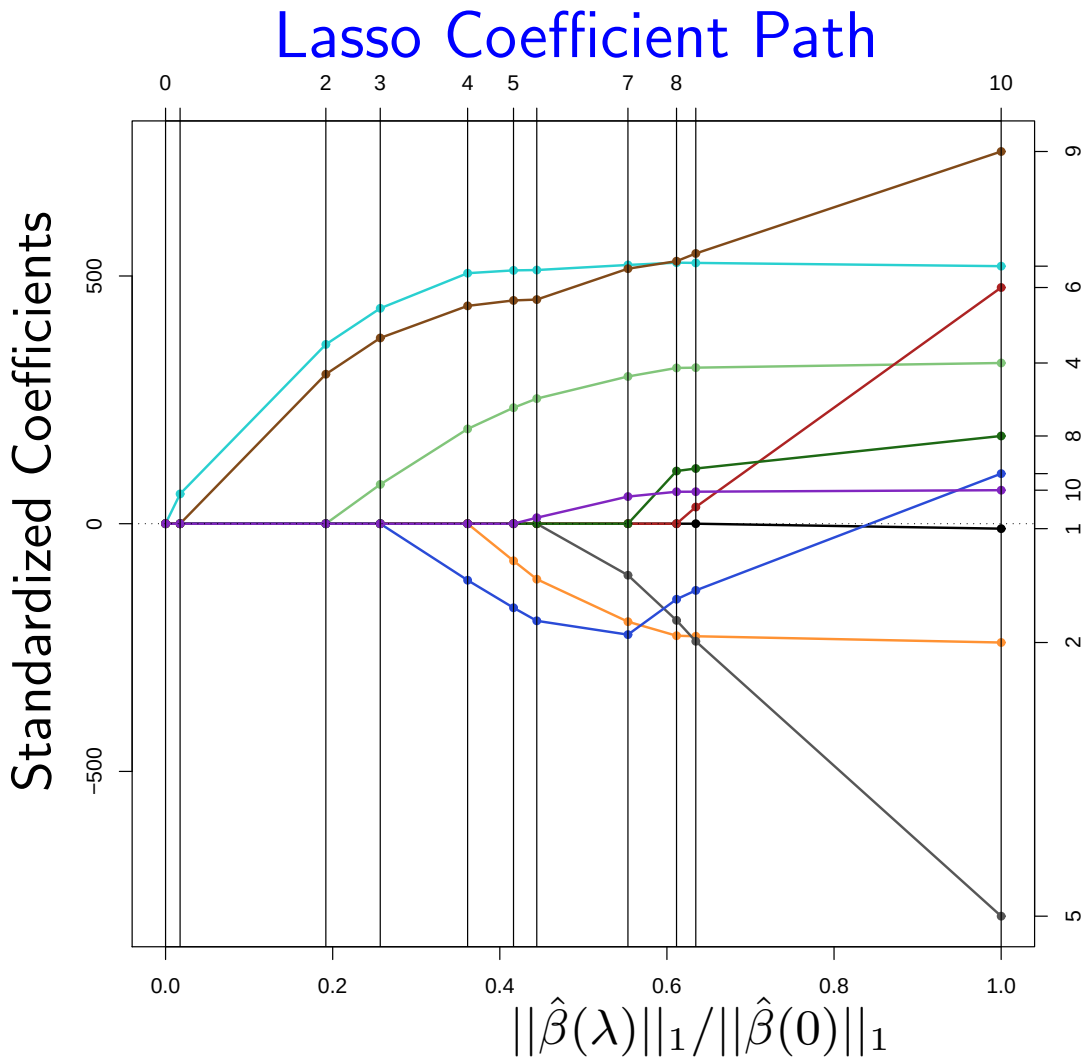
Forward stepwise adds variables one at a time and stops when overfitting is detected. Regained popularity for $p \gg N$, since it is the only feasible method among it's subset cousins (backward stepwise, best-subsets).

Ridge regression fits the model subject to constraint $\sum_{j=1}^{p} \beta_j^2 \leq t$. Shrinks coefficients toward zero, and hence controls variance. Allows linear models with arbitrary size p to be fit, although coefficients always in row-space of X.

Lasso regression (Tibshirani, 1995) fits the model subject to constraint $\sum_{j=1}^{p} |\beta_j| \leq t$.

Lasso does variable selection and shrinkage, while ridge only shrinks.

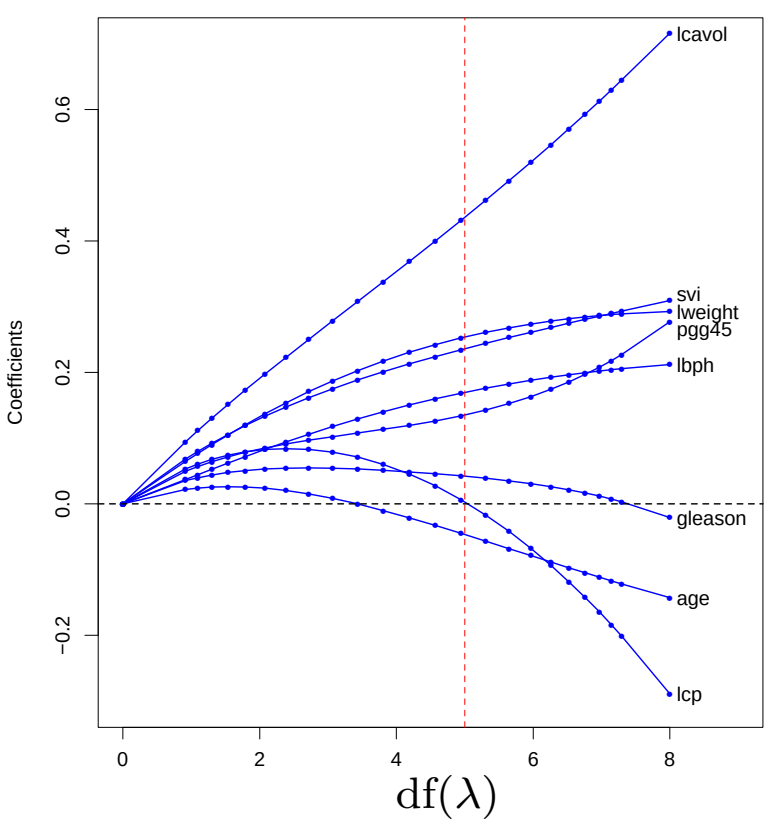


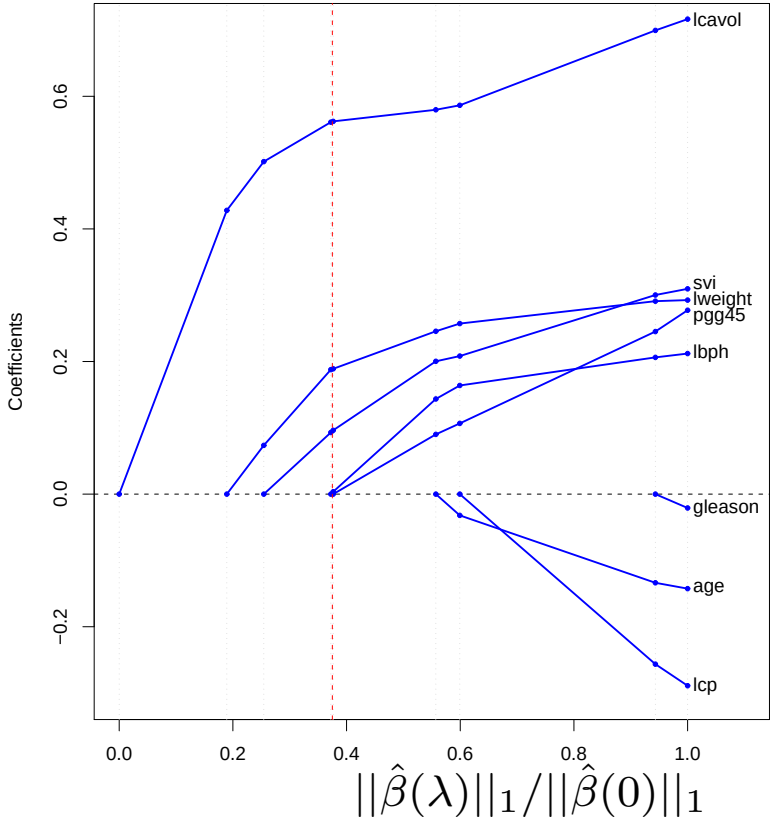


$$\text{Lasso: } \hat{\beta}(\lambda) = \text{argmin}_\beta \frac{1}{N} \sum_{i=1}^{N} (y_i - \beta_0 - x_i^T \beta)^2 + \lambda ||\beta||_1$$

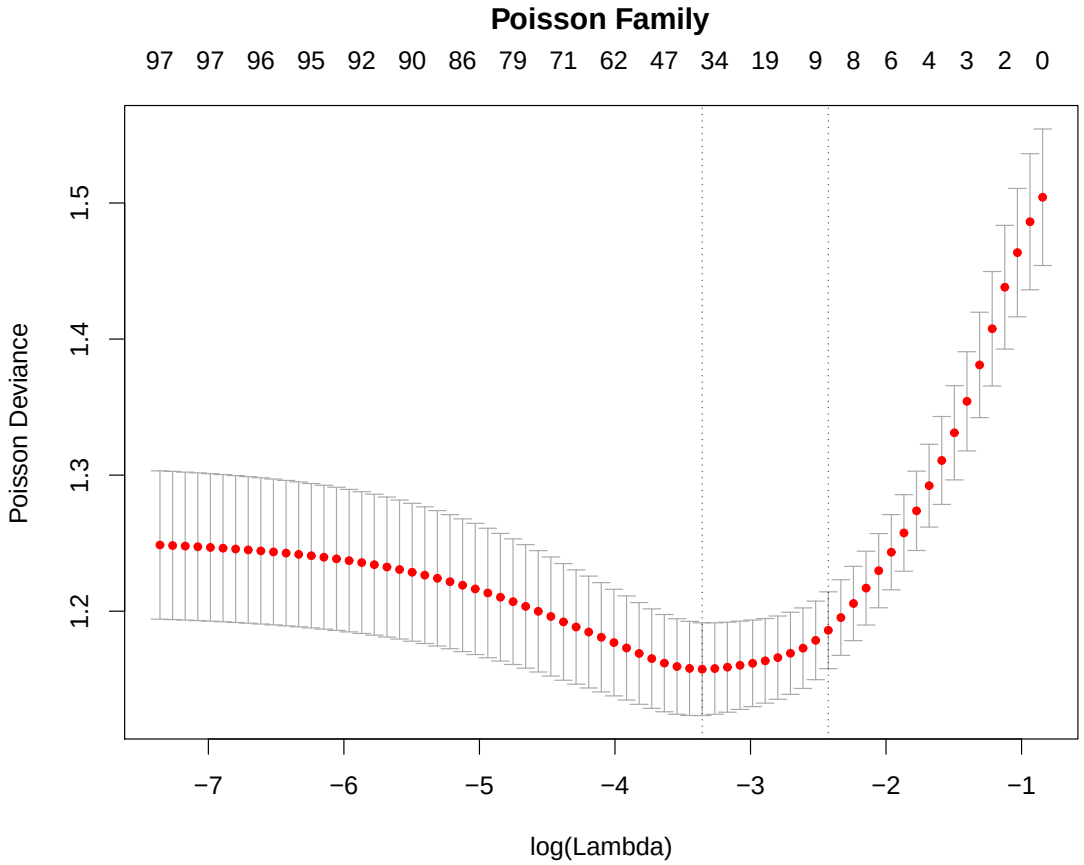
fit using LARS package in R (Efron, Hastie, Johnstone, Tibshirani 2002)

Ridge versus Lasso





Cross Validation to select λ



K-fold cross-validation is easy and fast. Here K=10, and the true model had 10 out of 100 nonzero coefficients.

History of Path Algorithms

Efficient path algorithms for $\hat{\beta}(\lambda)$ allow for easy and exact cross-validation and model selection.

- In 2001 the LARS algorithm (Efron et al) provides a way to compute the entire lasso coefficient path efficiently at the cost of a full least-squares fit.
- 2001 – 2008: path algorithms pop up for a wide variety of related problems: Group lasso (Yuan & Lin 2006), support-vector machine (Hastie, Rosset, Tibshirani & Zhu 2004), elastic net (Zou & Hastie 2004), quantile regression (Li & Zhu, 2007), logistic regression and glms (Park & Hastie, 2007), Dantzig selector (James & Radchenko 2008), ...
- Many of these do not enjoy the piecewise-linearity of LARS, and seize up on very large problems.

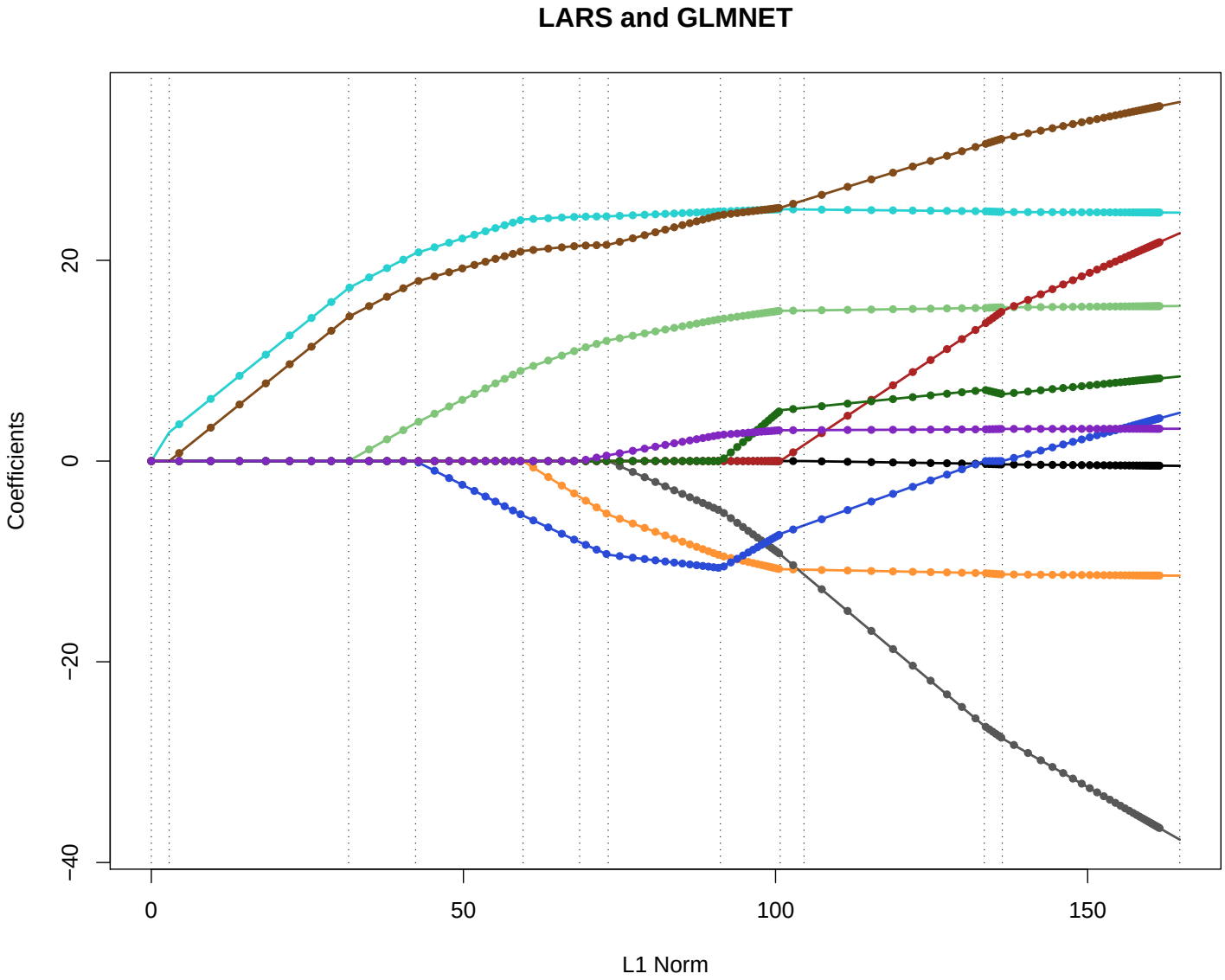
GLMNET and coordinate descent

- Solve the lasso problem by coordinate descent: optimize each parameter separately, holding all the others fixed. Updates are trivial. Cycle around till coefficients stabilize.
- Do this on a grid of λ values, from λ_{max} down to λ_{min} (uniform on log scale), using warms starts.
- Can do this with a variety of loss functions and additive penalties.

Coordinate descent achieves dramatic speedups over all competitors, by factors of 10, 100 and more.

Example: Newsgroup data: 11K obs, 778K features (sparse), 100 values λ across entire range, lasso logistic regression; time 29s on Macbook Pro.

References: Friedman, Hastie and Tibshirani 2010 + long list of other who have also worked with coordinate descent.


LARS and GLMNET
Coefficients
20
0
−20
−40
0
50
100
150
L1 Norm

A brief history of coordinate descent for the lasso

1997 Tibshirani's student Wenjiang Fu at U. Toronto develops the "shooting algorithm" for the lasso. Tibshirani doesn't fully appreciate it.

A brief history of coordinate descent for the lasso

1997 Tibshirani's student Wenjiang Fu at U. Toronto develops the "shooting algorithm" for the lasso. Tibshirani doesn't fully appreciate it.

2002 Ingrid Daubechies gives a talk at Stanford, describes a one-at-a-time algorithm for the lasso. Hastie implements it, makes an error, and Hastie +Tibshirani conclude that the method doesn't work.

A brief history of coordinate descent for the lasso

1997 Tibshirani's student Wenjiang Fu at U. Toronto develops the "shooting algorithm" for the lasso. Tibshirani doesn't fully appreciate it.

2002 Ingrid Daubechies gives a talk at Stanford, describes a one-at-a-time algorithm for the lasso. Hastie implements it, makes an error, and Hastie +Tibshirani conclude that the method doesn't work.

2006 Friedman is external examiner at PhD oral of Anita van der Kooij (Leiden) who uses coordinate descent for elastic net. Friedman, Hastie + Tibshirani revisit this problem. Others have too — Shevade and Keerthi (2003), Krishnapuram and Hartemink (2005), Genkin, Lewis and Madigan (2007), Wu and Lange (2008), Meier, van de Geer and Buehlmann (2008).

GLMNET package in R

Fits coefficient paths for a variety of different GLMs and the *elastic net* family of penalties.

Some features of `glmnet`:

- Models: linear, logistic, multinomial (grouped or not), Poisson, Cox model, and multiple-response grouped linear.
- Elastic net penalty includes *ridge* and *lasso*, and hybrids in between (more to come)
- *Speed!*
- Can handle large number of variables p. Along with screening rules we can fit GLMs on GWAS scale (more to come)
- Cross-validation functions for all models.
- Can allow for sparse matrix formats for $\mathbf{X}$, and hence massive

problems (eg $N = 11K$, $p = 750K$ logistic regression).

- Can provide lower and upper bounds for each coefficient; eg: positive lasso
- Useful bells and whistles:
  - Offsets — as in GLM, can have part of the linear predictor that is given and not fit. Often used in Poisson models (sampling frame).
  - Penalty strengths — can alter relative strength of penalty on different variables. Zero penalty means a variable is *always in* the model. Useful for adjusting for demographic variables.
  - Observation weights allowed.
  - Can fit no-intercept models
  - Session-wise parameters can be set with new `glmnet.options` command.

Coordinate descent for the lasso

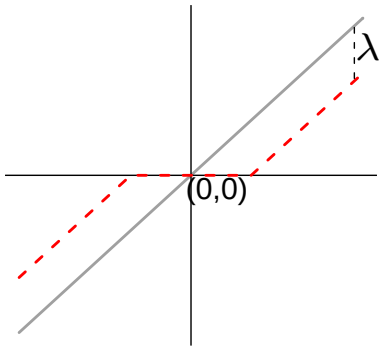
$$\min_\beta \frac{1}{2N}\sum_{i=1}^N (y_i - \sum_{j=1}^p x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^p |\beta_j|$$

Suppose the p predictors and response are standardized to have mean zero and variance 1. Initialize all the $\beta_j = 0$.

Cycle over $j = 1, 2, \ldots, p, 1, 2, \ldots$ till convergence:

- Compute the partial residuals $r_{ij} = y_i - \sum_{k\neq j} x_{ik}\beta_k$.
- Compute the simple least squares coefficient of these residuals on jth predictor: $\beta_j^* = \frac{1}{N}\sum_{i=1}^N x_{ij}r_{ij}$
- Update β_j by *soft-thresholding*:

$$\begin{aligned}\beta_j &\leftarrow S(\beta_j^*, \lambda)\\ &= \text{sign}(\beta_j^*)(|\beta_j^*| - \lambda)_+\end{aligned}$$



Elastic-net penalty family

Family of convex penalties proposed in Zou and Hastie (2005) for $p \gg N$ situations, where predictors are correlated in groups.

$$\min_\beta \frac{1}{2N}\sum_{i=1}^N (y_i - \sum_{j=1}^p x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^p P_\alpha(\beta_j)$$

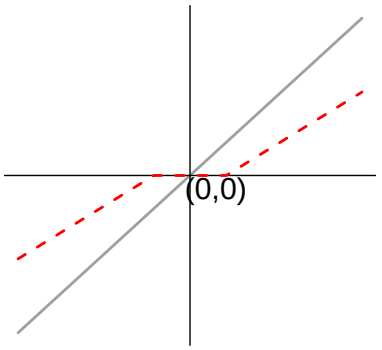
$$\text{with } P_\alpha(\beta_j) = \frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j|.$$

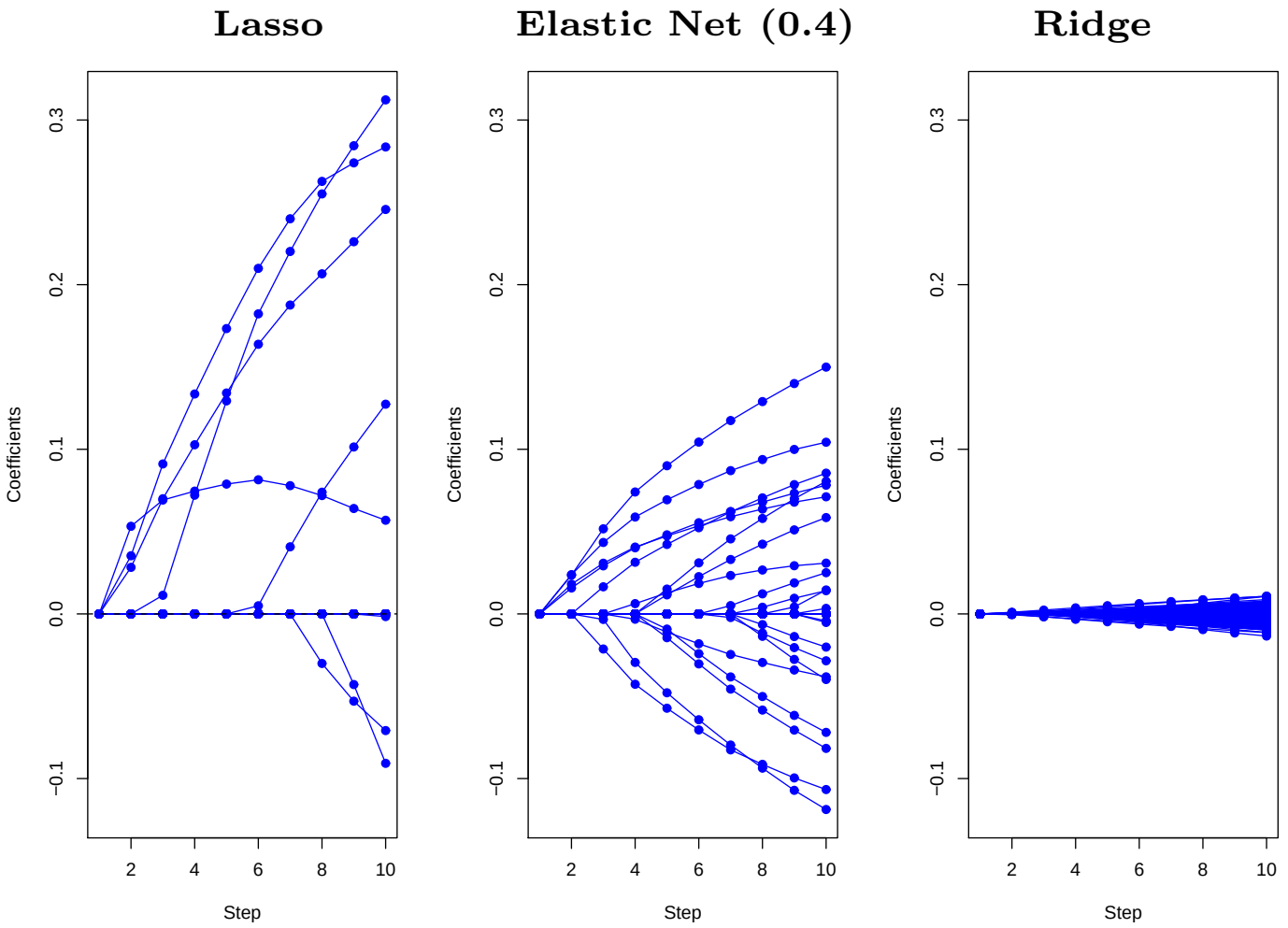
α creates a compromise between the *lasso* and *ridge*.

Coordinate update is now

$$\beta_j \leftarrow \frac{S(\beta_j^*,\ \lambda\alpha)}{1+\lambda(1-\alpha)}$$

where $\beta_j^* = \frac{1}{N}\sum_{i=1}^N x_{ij}r_{ij}$ as before.





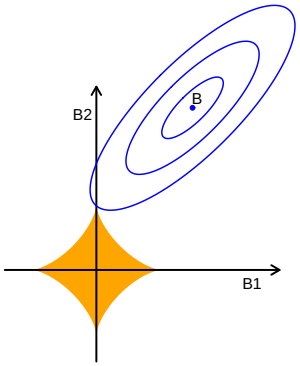
Leukemia Data, Logistic, N=72, p=3571, first 10 steps shown

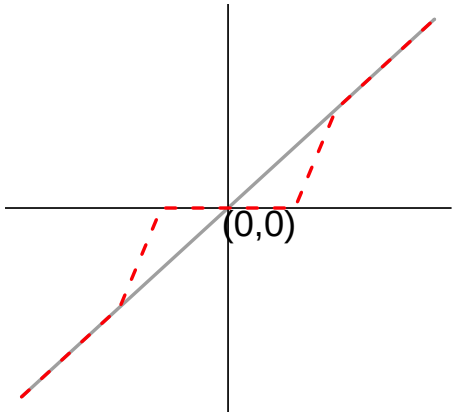
Sparser than Lasso — Concave Penalties

Work with past PhD student Rahul Mazumder and Jerry Friedman (2010).

Extends elastic net family into concave domain

Many approaches. We propose family that bridges ℓ_1 and ℓ_0 based on MC+ penalty (Zhang 2010), and a coordinate-descent scheme for fitting model paths, implemented in SPARSENET





Screening Rules

Logistic regression for GWAS: $p \sim$ million, $N = 2000$
(Wu et al, 2009)

- Compute $|\langle x_j, y - \bar{y}\rangle|$ for each Snp $j = 1, 2, \ldots, 10^6$, where $\bar{y}$ is the mean of (binary) y.
  Note: the largest of these is λ_{max} — smallest value of λ for which all coefficients are zero.
- Fit lasso logistic regression path using only largest 1000 (typically fit models of size around 20 or 30 in GWAS)
- Simple confirmations check that omitted Snps would not have entered the model.

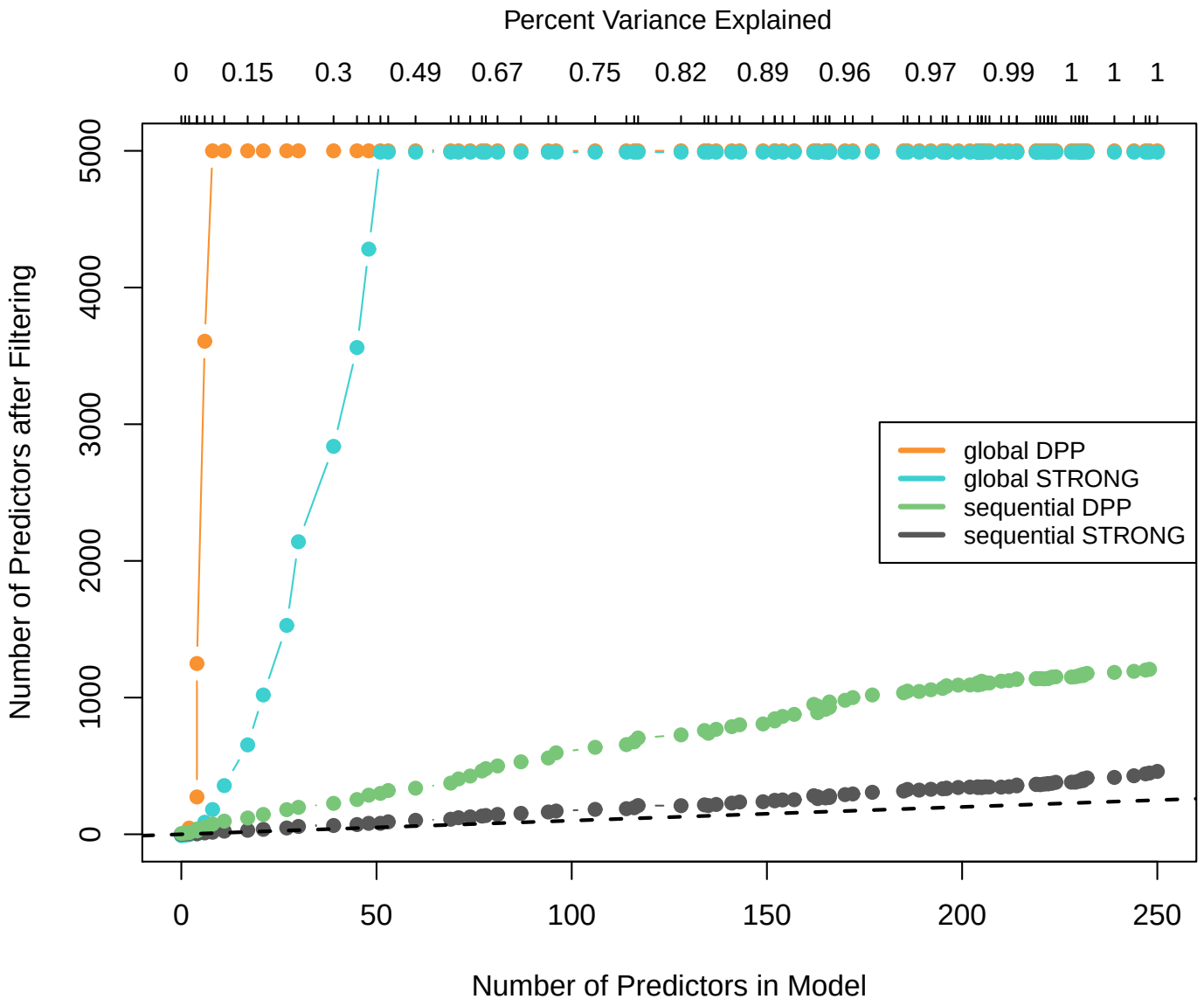
Safe and Strong Rules

- El Ghaoui et al (2010), improved by Wang et al (2012) propose SAFE rules for Lasso for screening predictors — can be quite conservative.
- Tibshirani et al (2012) improve these using STRONG screening rules.

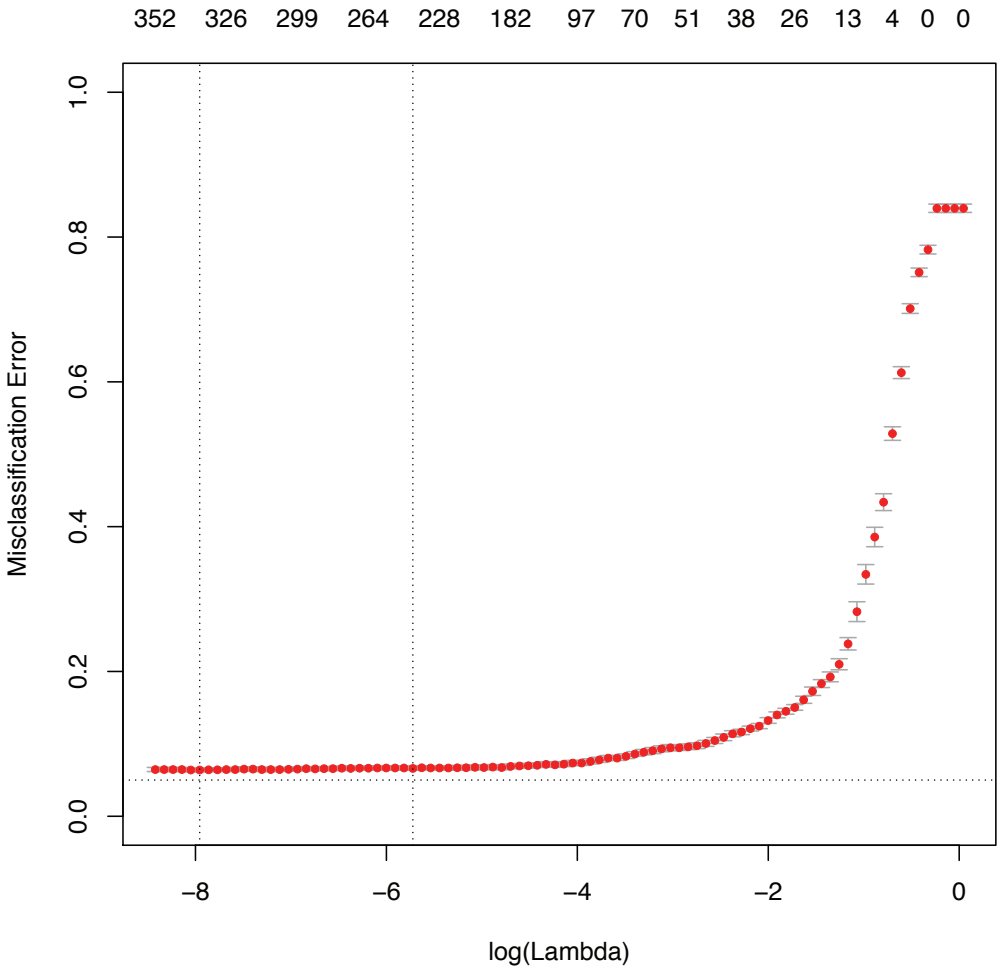
  Suppose fit at λ_ℓ is $\mathbf{X}\hat{\beta}(\lambda_\ell)$, and we want to compute the fit at $\lambda_{\ell+1} < \lambda_\ell$. Note: $|\langle \mathbf{x}_j, \mathbf{y} - \mathbf{X}\hat{\beta}(\lambda_\ell)\rangle| = \lambda_\ell \ \forall j \in \mathcal{A}$. ,
  $\leq \lambda_\ell \ \forall j \notin \mathcal{A}$.
  Strong rules only consider set

$$\left\{ j : |\langle \mathbf{x}_j, \mathbf{y} - \mathbf{X}\hat{\beta}(\lambda_\ell)\rangle| > \lambda_{\ell+1} - (\lambda_\ell - \lambda_{\ell+1}) \right\}$$

  GLMNET screens at every λ step, and after convergence, checks if any violations.


Percent Variance Explained
0
0.15
0.3
0.49
0.67
0.75
0.82
0.89
0.96
0.97
0.99
1
1
1
Number of Predictors after Filtering
0
1000
2000
3000
4000
5000
0
50
100
150
200
250
Number of Predictors in Model
global DPP
global STRONG
sequential DPP
sequential STRONG

Example: multiclass classification



Pathwork® Diagnostics

Microarray classification: tissue of origin

3220 samples

22K genes

17 classes (tissue type)

Multinomial regression model with

$17 \times 22\text{K} = 374\text{K}$ parameters

Elastic-net ($\alpha = 0.25$)

Coordinate Descent in General

Many problems have the form

$$\min_{\{\beta_j\}_1^p} \left[R(y, \beta) + \lambda \sum_{j=1}^{p} P_j(\beta_j) \right].$$

- If R and P_j are convex, and R is differentiable, then coordinate descent converges to the solution (Tseng, 1988).
- Often each coordinate step is trivial. E.g. for lasso, it amounts to soft-thresholding, with many steps leaving $\hat{\beta}_j = 0$.
- Decreasing λ slowly means not much cycling is needed.
- Coordinate moves can exploit sparsity.

Other Applications

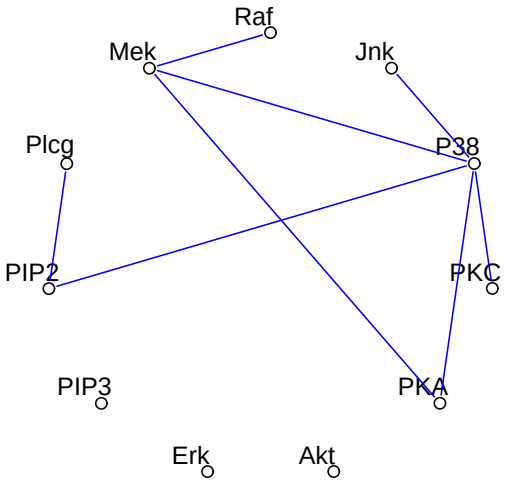
Undirected Graphical Models — learning dependence structure via the lasso. Model the inverse covariance $\mathbf{\Theta}$ in the Gaussian family with L_1 penalties applied to elements.

$$\max_{\mathbf{\Theta}} \log\det \mathbf{\Theta} - \mathrm{Tr}(\mathbf{S\Theta}) - \lambda||\mathbf{\Theta}||_1$$

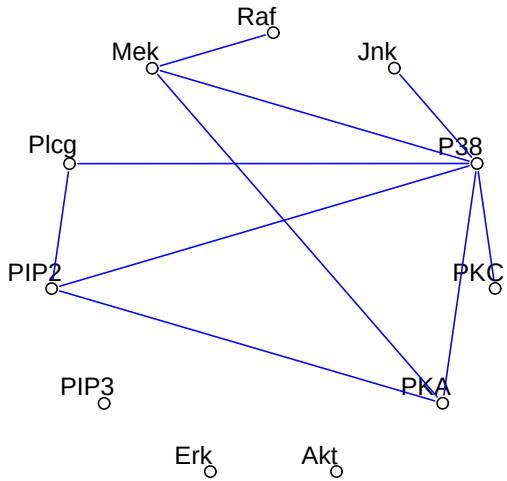
GLASSO: modified block-wise lasso algorithm, which we solve by coordinate descent (FHT 2007). Algorithm is very fast, and solve moderately sparse graphs with 1000 nodes in under a minute.

Example: flow cytometry - $p = 11$ proteins measured in $N = 7466$ cells (Sachs et al 2003) (next page)

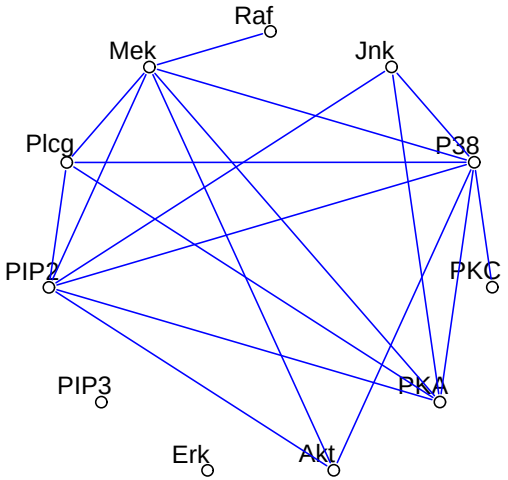
$\lambda = 36$



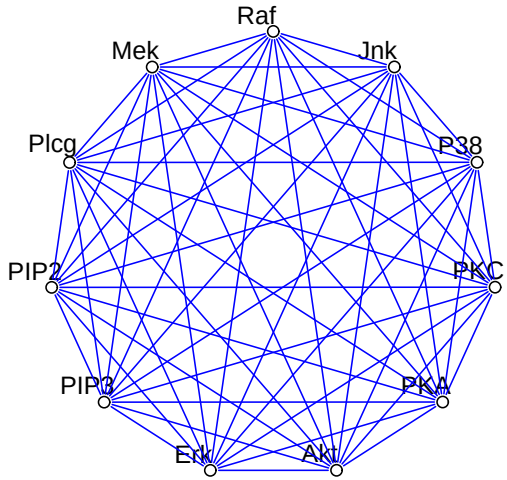
$\lambda = 27$



$\lambda = 7$



$\lambda = 0$



Group Lasso (Yuan and Lin, 2007, Meier, Van de Geer, Buehlmann, 2008) — each term $P_j(\beta_j)$ applies to *sets* of parameters:

$$R(y, \sum_{j=1}^{J} \mathbf{X}_j \beta_j) + \lambda \sum_{j=1}^{J} \gamma_j ||\beta_j||_2.$$

Example: each block represents the levels for a categorical predictor.

- entire groups are zero, or all elements are nonzero.
- γ_j is penalty modifier for group j; $\gamma_j = \|\mathbf{X}_j\|_F$ is good choice.
- Leads to a block-updating form of coordinate descent.
- Strong rules apply here: $\|\mathbf{X}_j^T \mathbf{r}\|_2 > \gamma_j[\lambda_{\ell+1} - (\lambda_\ell - \lambda_{\ell+1})]$

Mixed Graphical Models

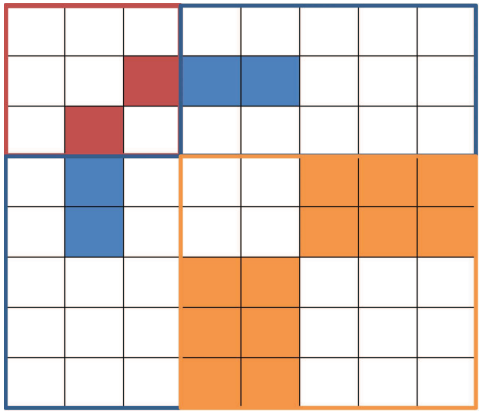
Project with PhD student Jason Lee (JCGS 2014).
General Markov random field representation, with edge and node potentials.

$$p(x,y;\Theta) \propto \exp\left(\sum_{s=1}^{p}\sum_{t=1}^{p} -\frac{1}{2}\beta_{st}x_s x_t + \sum_{s=1}^{p}\alpha_s x_s + \sum_{s=1}^{p}\sum_{j=1}^{q}\rho_{sj}(y_j)x_s + \sum_{j=1}^{q}\sum_{r=1}^{q}\phi_{rj}(y_r,y_j)\right)$$

- Pseudo likelihood allows simple inference with mixed variables. Conditionals for continuous are Gaussian linear regression models, for categorical are binomial or multinomial logistic regressions.
- Parameters come in symmetric blocks, and the inference should respect this symmetry (next slide)

Mixed Graphical Model: group-lasso penalties

Parameters in blocks. Here we have an interaction between a pair of quantitative variables (red), a 2-level qualitative with a quantitative (blue), and an interaction between the 2 level and a 3 level qualitative.



Maximize a pseudo-likelihood with lasso and group-lasso penalties on parameter blocks.

$$\max_{\Theta} \ell(\Theta) - \lambda \left(\sum_{s=1}^{p} \sum_{t=1}^{s-1} |\beta_{st}| + \sum_{s=1}^{p} \sum_{j=1}^{q} \|\rho_{sj}\|_2 + \sum_{j=1}^{q} \sum_{r=1}^{j-1} \|\phi_{rj}\|_F \right)$$

Solved using proximal Newton algorithm for a decreasing sequence of values for λ [Lee and Hastie, JCGS 2013].

Overlap Group Lasso (Jacob et al, 2009) Example: consider the model

$$\eta(X) = X_1\beta_1 + X_1\theta_1 + X_2\theta_2$$

with penalty

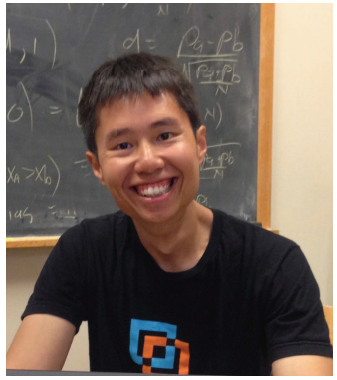
$$|\beta_1| + \sqrt{\theta_1^2 + \theta_2^2}$$

The coefficient of X_1 is nonzero if either group is nonzero; allows one to enforce hierarchy.

We look at two applications:

- Modeling interactions with strong hierarchy — interactions present only when main-effects are present. Project with just-graduated Ph.D student Michael Lim.
- Sparse additive models (SPAM, Ravikumar et al 2009). We use overlap group lasso in a different approach to SPAM models. Work near completion with Ph.D student Alexandra Chouldechova.

Glinternet



Project with PhD student Michael Lim (JCGS 2014)
Linear + first-order interaction models using group lasso

Example: GWAS with $p = 27K$ Snps , each a 3-level factor, and a binary response, $N = 3500$.

- Let X_j be $N \times 3$ indicator matrix for each Snp, and $X_{j:k} = X_j \star X_k$ be the $N \times 9$ *interaction* matrix.
- We fit model

$$\log \frac{\Pr(Y = 1|X)}{\Pr(Y = 0|X)} = \alpha + \sum_{j=1}^{p} X_j \beta_j + \sum_{j<k} X_{j:k} \theta_{j:k}$$

- note: $X_{j:k}$ encodes main effects and interactions.

- Maximize group-lasso penalized likelihood:

$$\ell(\mathbf{y}, \mathbf{p}) - \lambda \left[\sum_{j=1}^{p} \|\beta_j\|_2 + \sum_{j<k} \|\theta_{j:k}\|_2 \right]$$

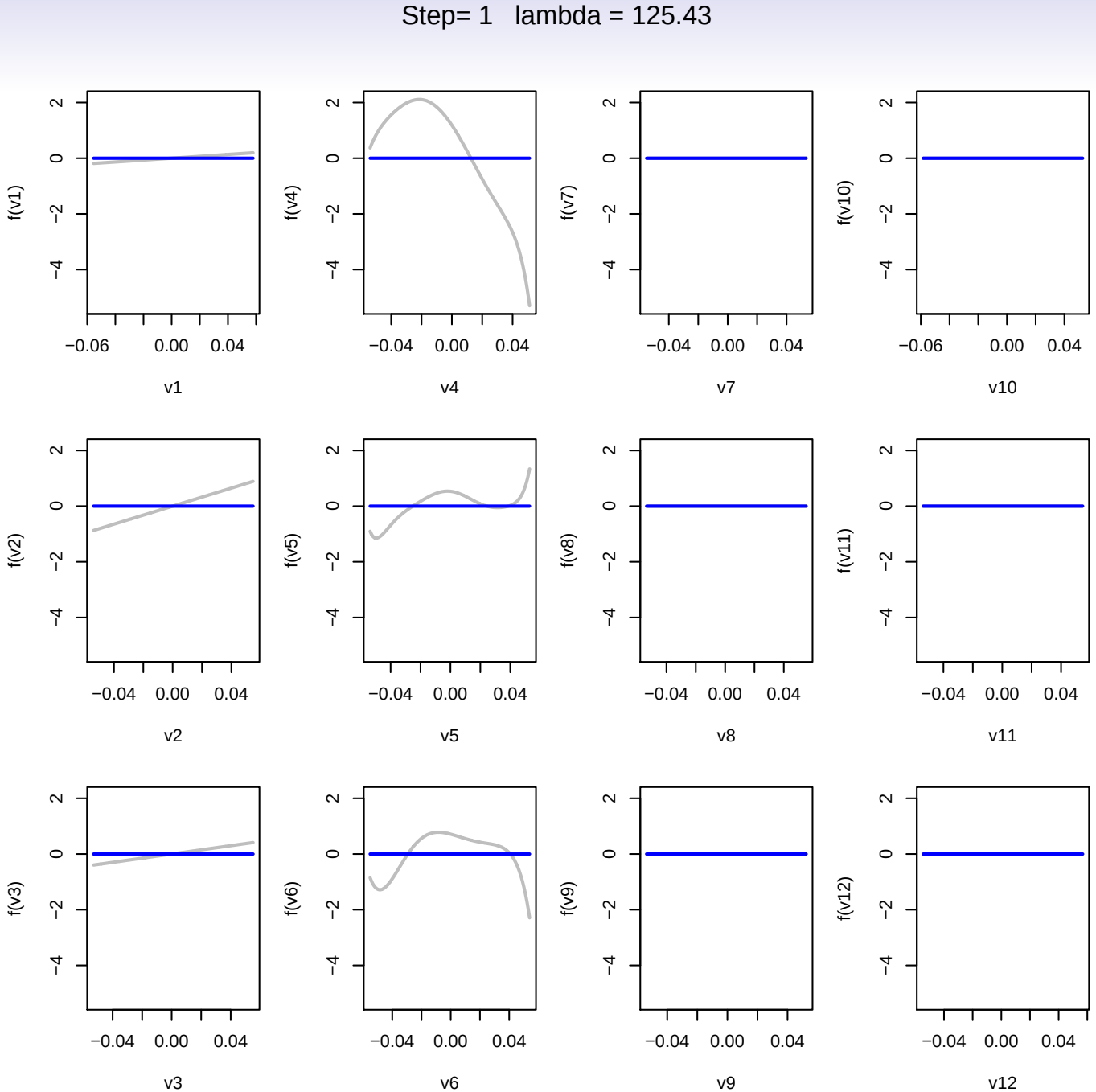
- Solutions map to traditional hierarchical main-effects/interactions model (with effects summing to zero).
- Strong rules for feature filtering essential here — parallel and distributed computing useful too. GWAS search space of 729M interactions!
- Formulated for all types of interactions, not just catgeorical variables.
- GLINTERNET very fast — two-orders of magnitude faster than competition, with similar performance.

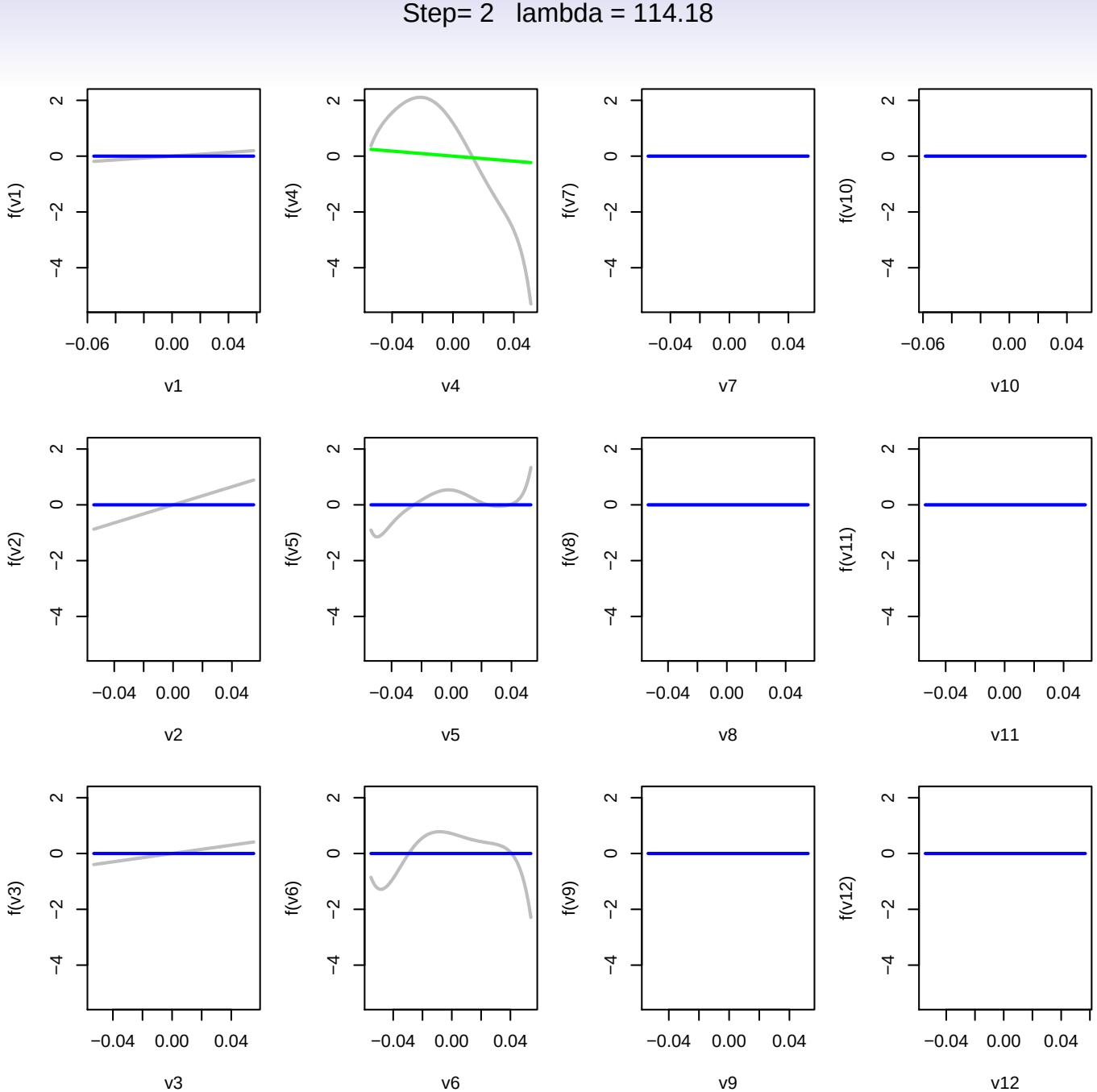
Sparse Generalized Additive Models

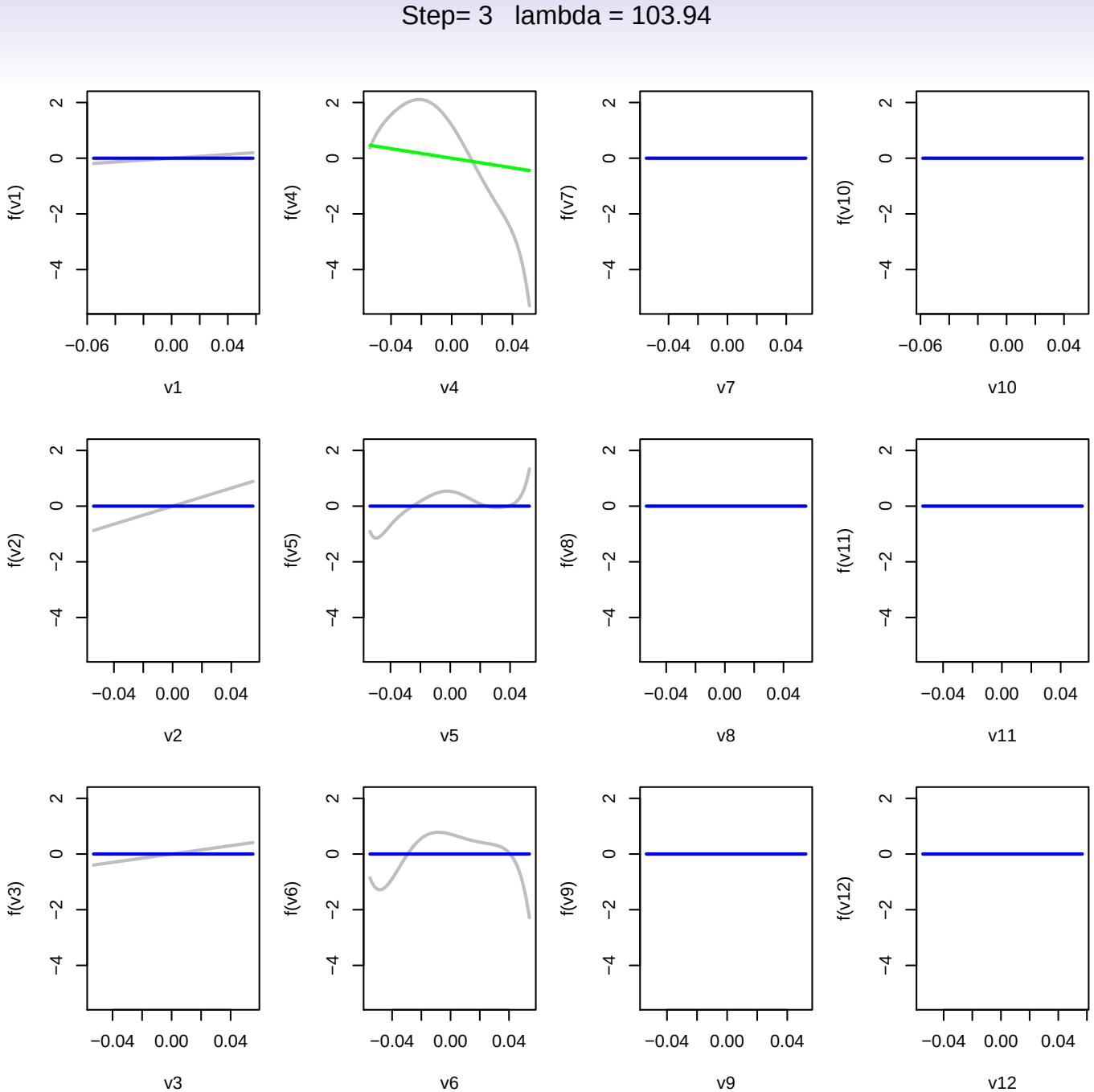
Work with Alexandra Chouldechova.
Automatic, *sticky* selection between zero, linear or nonlinear terms in GAMs. E.g. $y = \sum_{j=1}^{p} f_j(x_j) + \epsilon$.

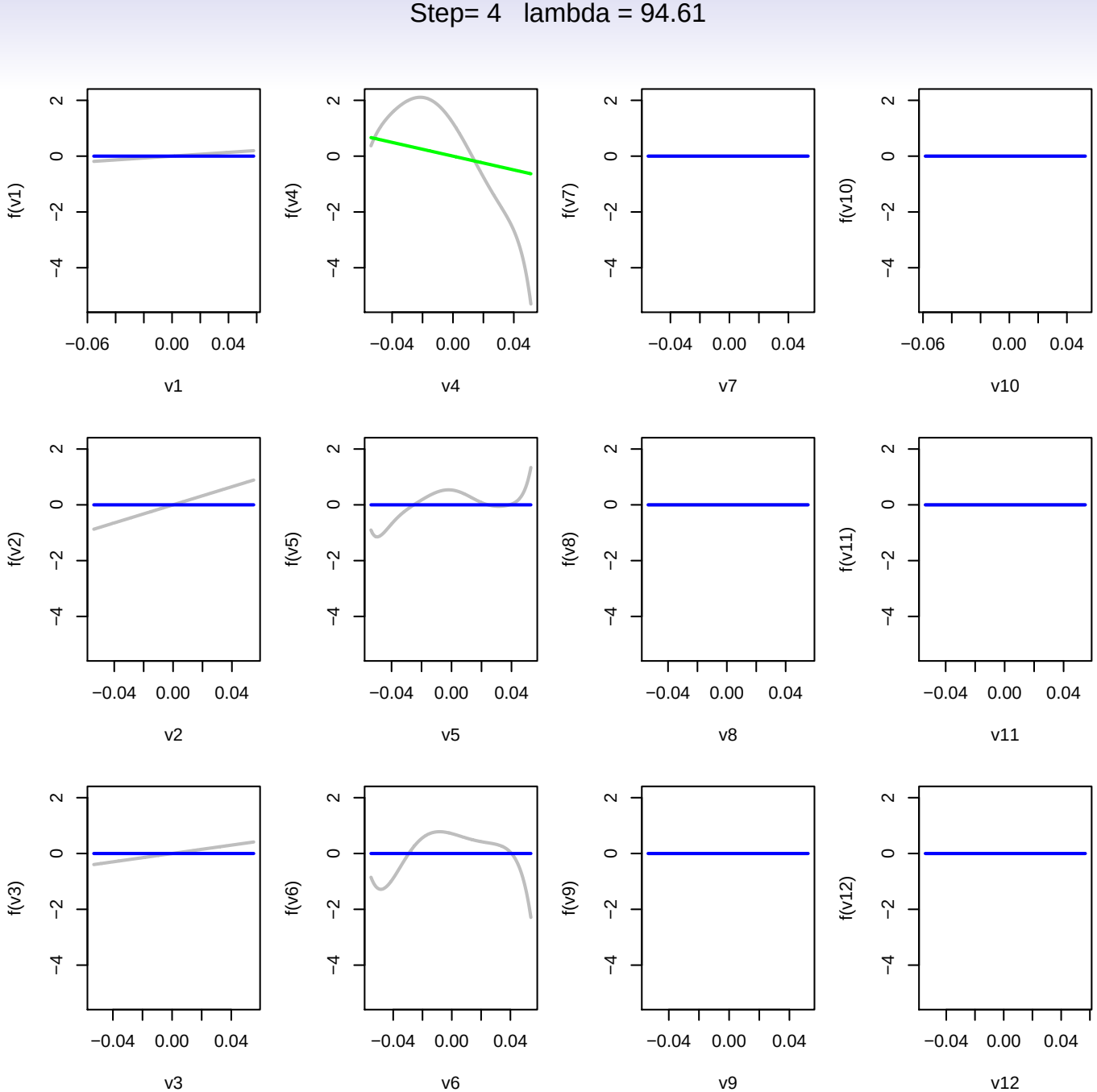
$$\frac{1}{2}\left\| y - \sum_{j=1}^{p} \alpha_j x_j - \sum_{j=1}^{p} U_j \beta_j \right\|^2 + \lambda \left\{ \sum_{j=1}^{p} |\alpha_j| + \gamma\lambda \sum_{j=1}^{p} \|\beta_j\|_D \right\} + \frac{1}{2} \sum_{j=1}^{p} \psi_j \beta_j^T D_{(-1)} \beta_j$$

- $U_j = [x_j \; p_1(x_j) \; \cdots \; p_k(x_j)]$ where the p_i are orthogonal Demmler-Reinsch spline basis functions of increasing degree.
- $D = \text{diag}(d_0, d_1, \ldots, d_k)$ diagonal penalty matrix with $1 = d_0 < d_1 \leq d_2 \leq \cdots \leq d_k$, and $D_{(-1)}$ sets $d_0 = 0$.

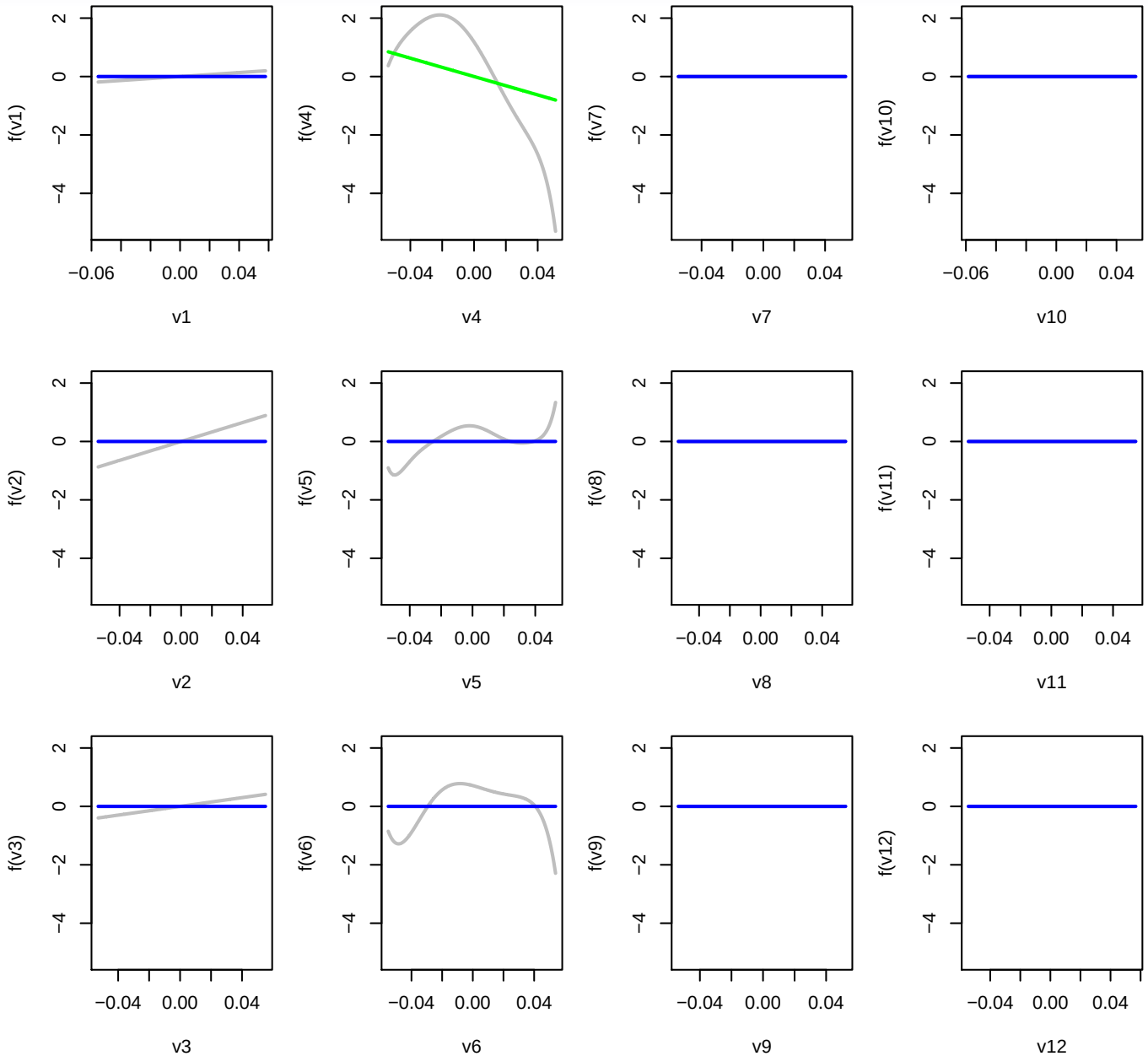

Step= 1 lambda = 125.43
f(v1)
f(v4)
f(v7)
f(v10)
f(v2)
f(v5)
f(v8)
f(v11)
f(v3)
f(v6)
f(v9)
f(v12)
v1
v4
v7
v10
v2
v5
v8
v11
v3
v6
v9
v12

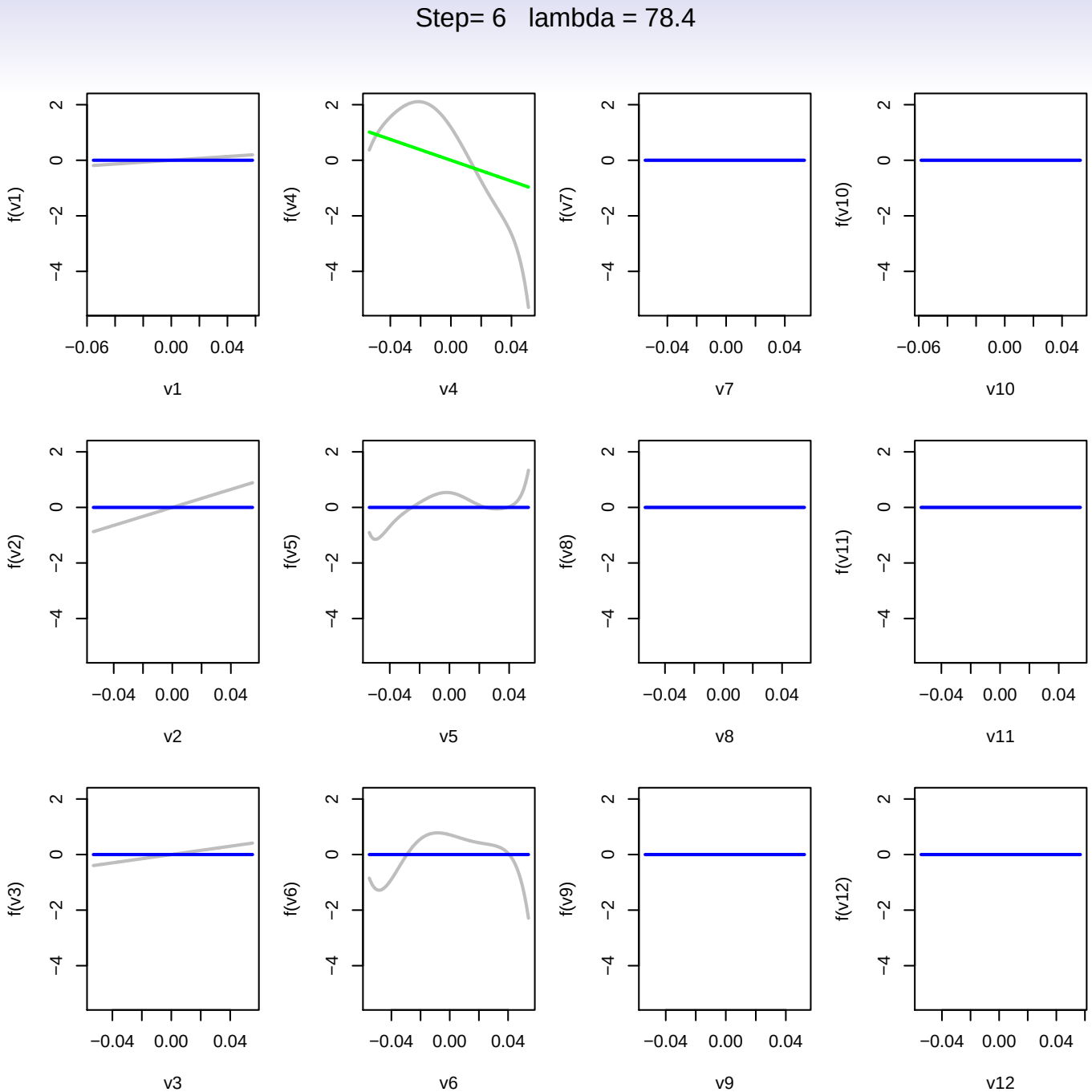

Step= 2 lambda = 114.18
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12


Step= 3 lambda = 103.94
f(v1)
f(v4)
f(v7)
f(v10)
f(v2)
f(v5)
f(v8)
f(v11)
f(v3)
f(v6)
f(v9)
f(v12)
v1
v4
v7
v10
v2
v5
v8
v11
v3
v6
v9
v12

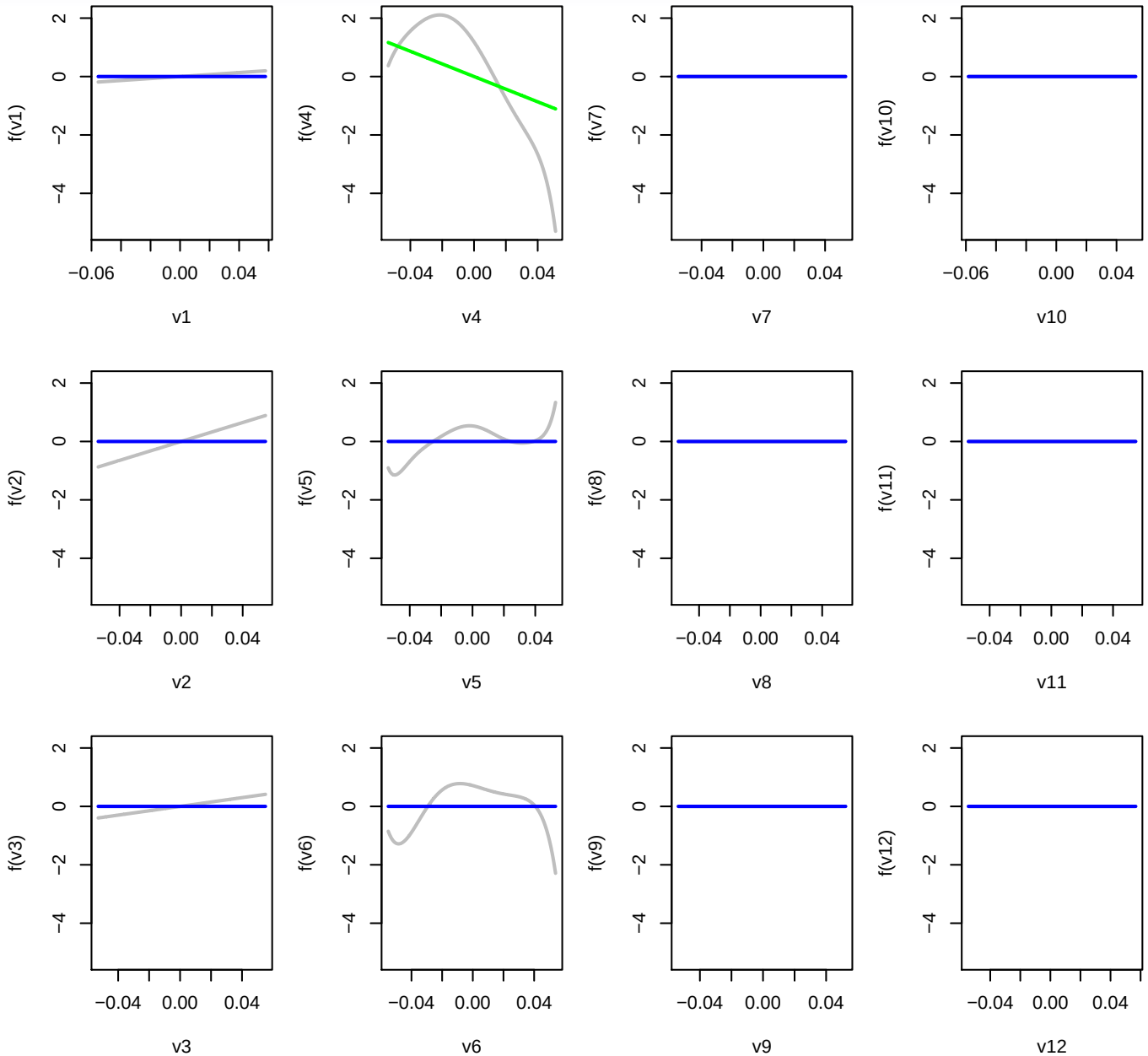

Step= 4 lambda = 94.61
f(v1)
f(v4)
f(v7)
f(v10)
f(v2)
f(v5)
f(v8)
f(v11)
f(v3)
f(v6)
f(v9)
f(v12)
v1
v4
v7
v10
v2
v5
v8
v11
v3
v6
v9
v12

Step= 5 lambda = 86.13

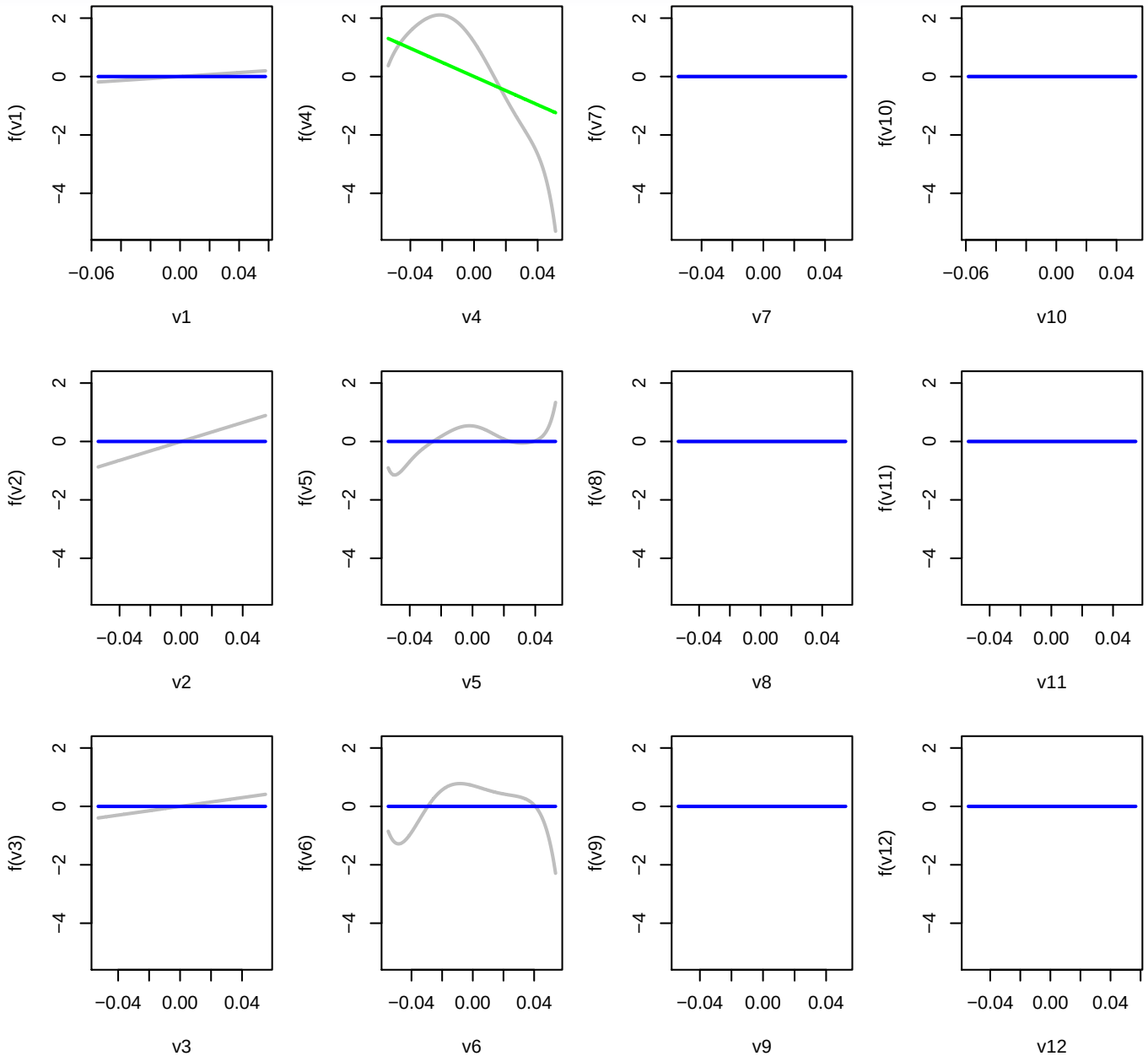


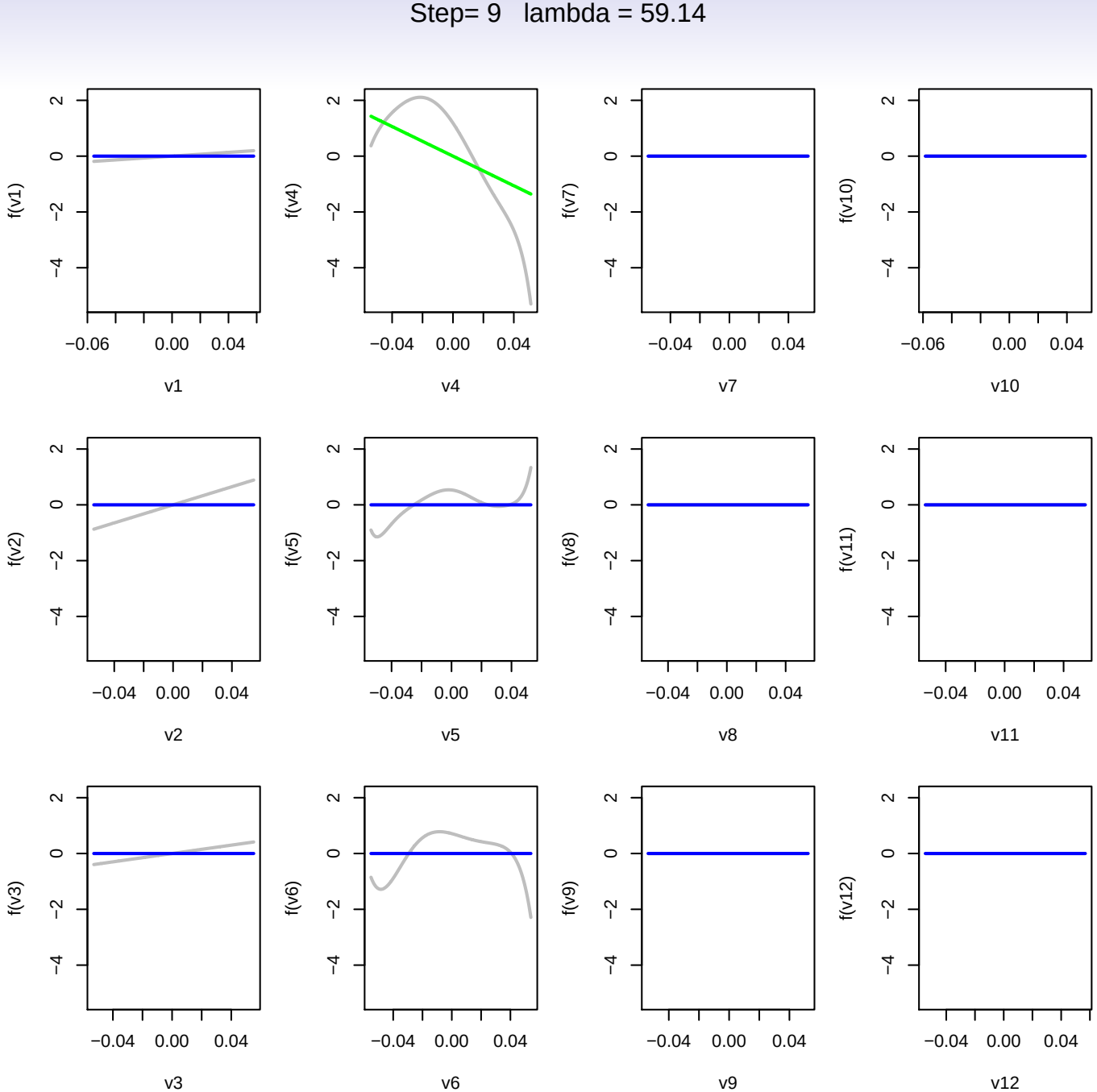

Step= 6 lambda = 78.4
f(v1)
f(v4)
f(v7)
f(v10)
f(v2)
f(v5)
f(v8)
f(v11)
f(v3)
f(v6)
f(v9)
f(v12)
v1
v4
v7
v10
v2
v5
v8
v11
v3
v6
v9
v12

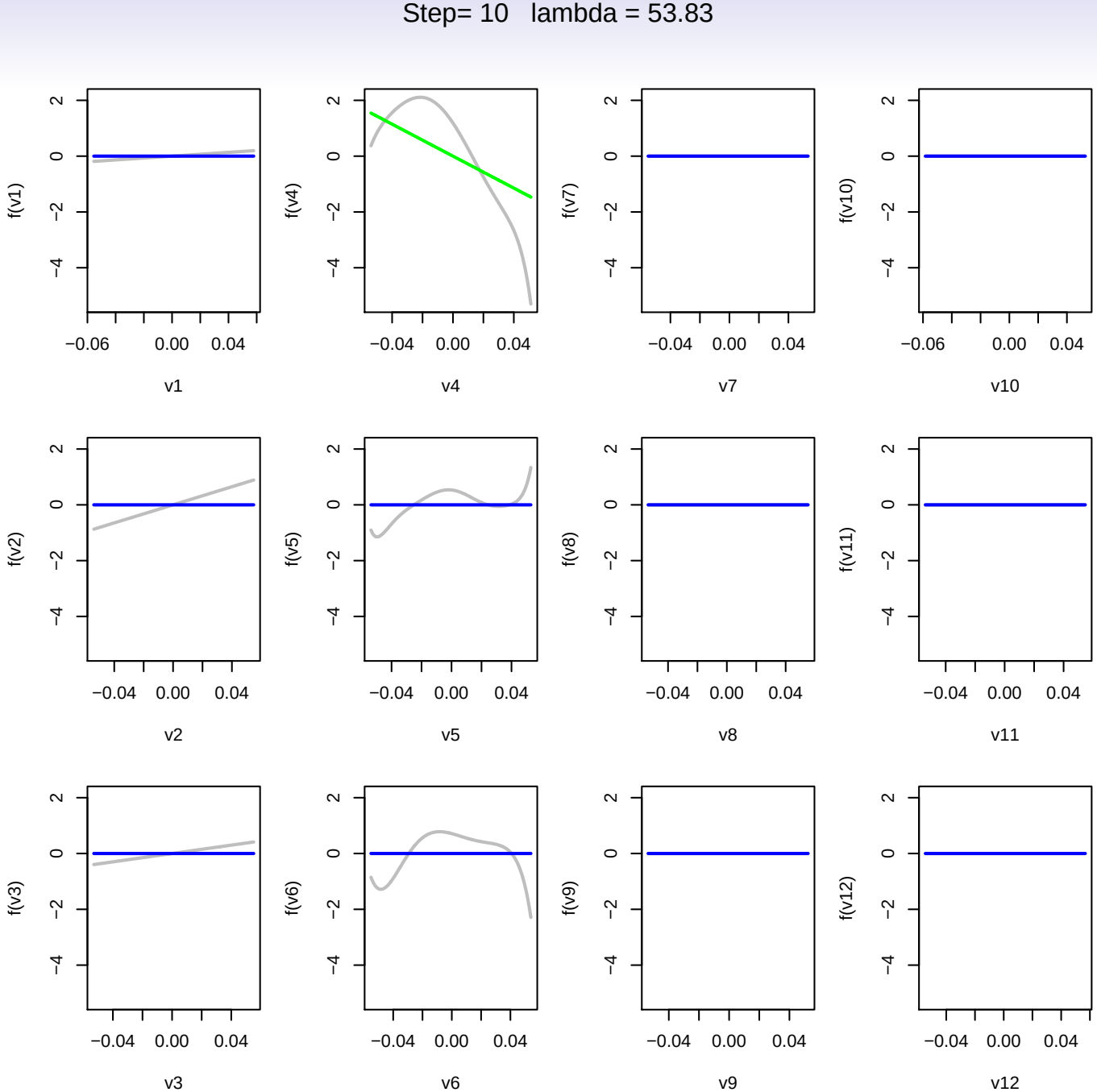
Step= 7 lambda = 71.37

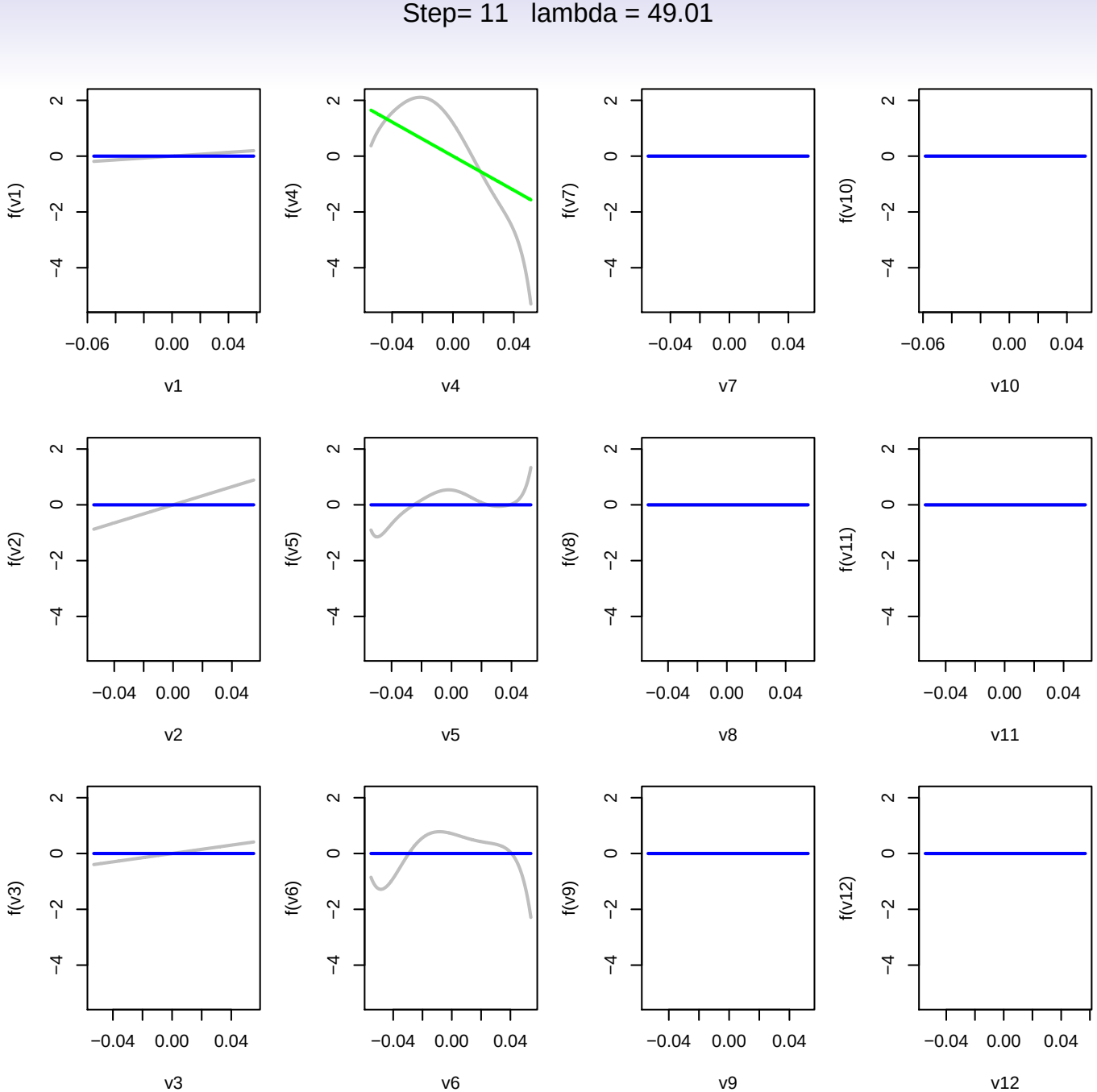


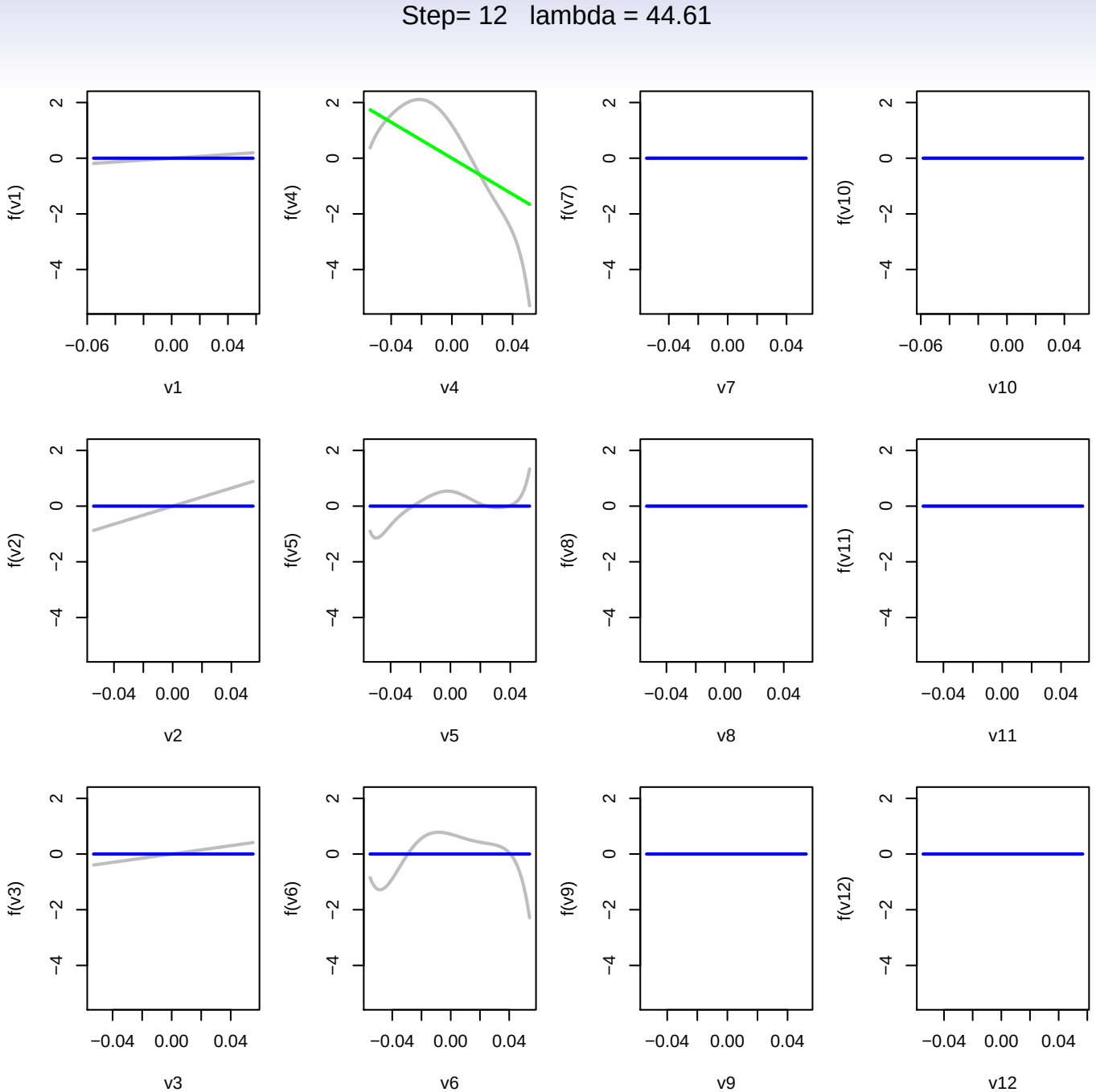
Step= 8 lambda = 64.97

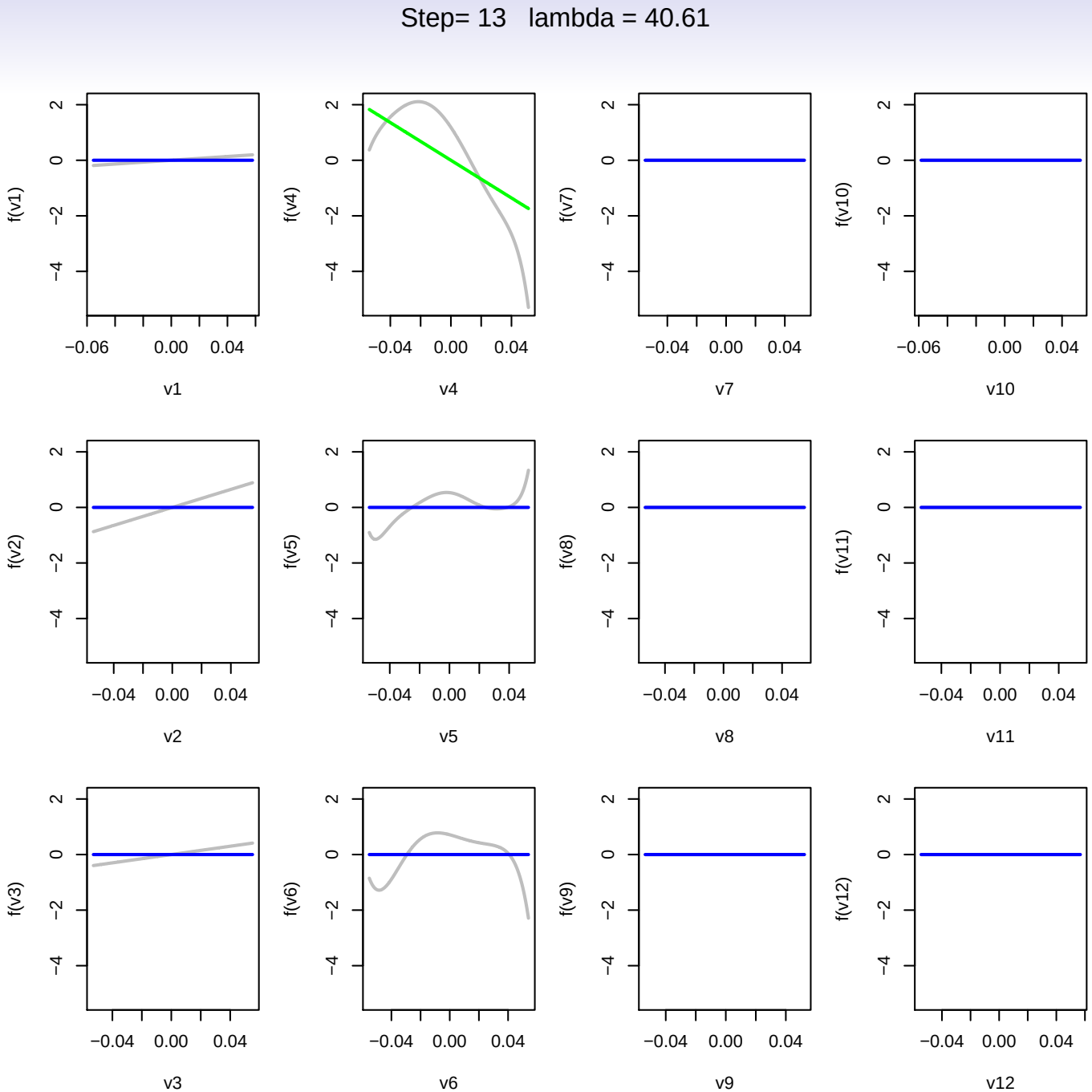


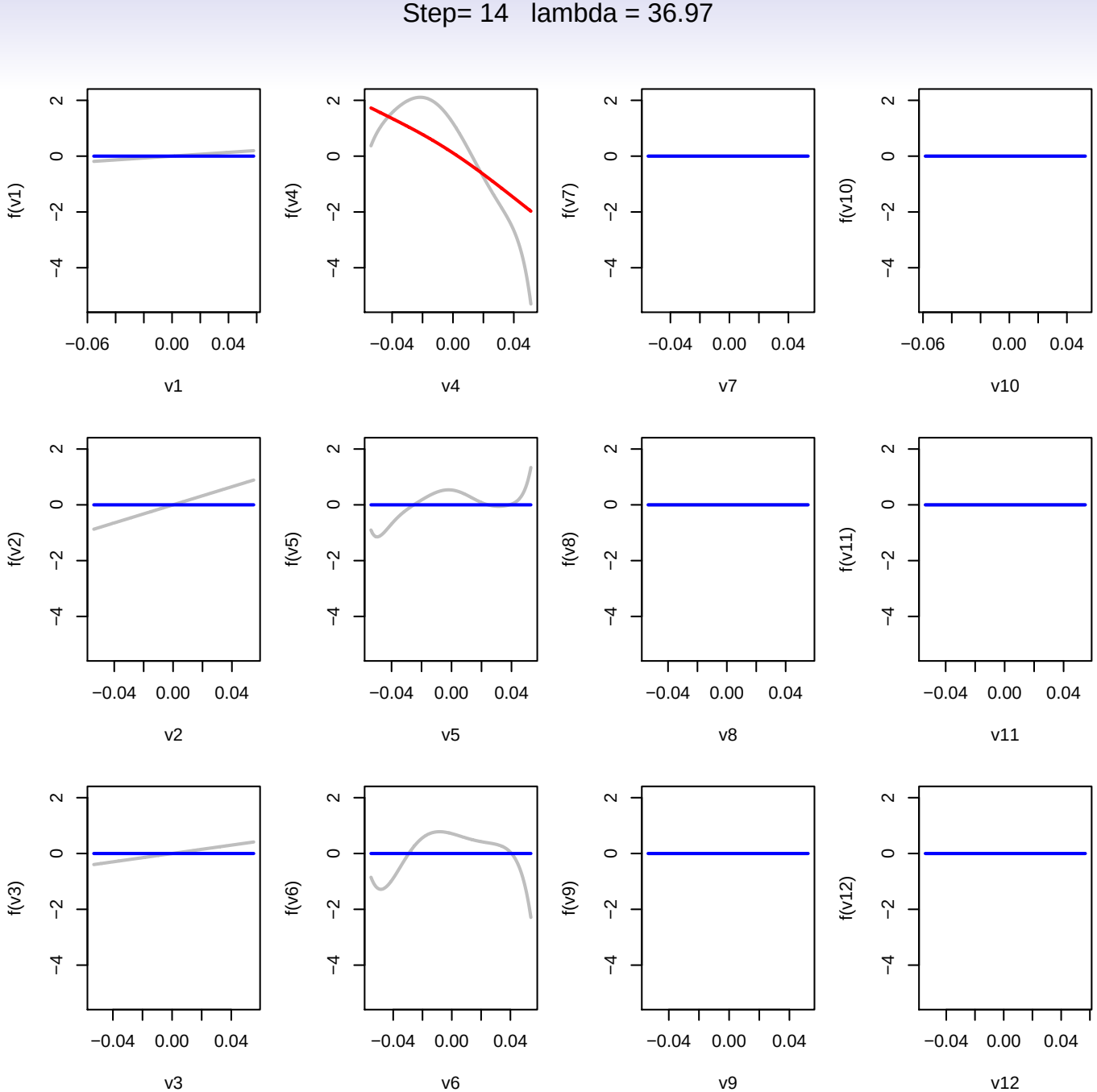

Step= 9 lambda = 59.14
f(v1)
f(v4)
f(v7)
f(v10)
f(v2)
f(v5)
f(v8)
f(v11)
f(v3)
f(v6)
f(v9)
f(v12)
v1
v4
v7
v10
v2
v5
v8
v11
v3
v6
v9
v12

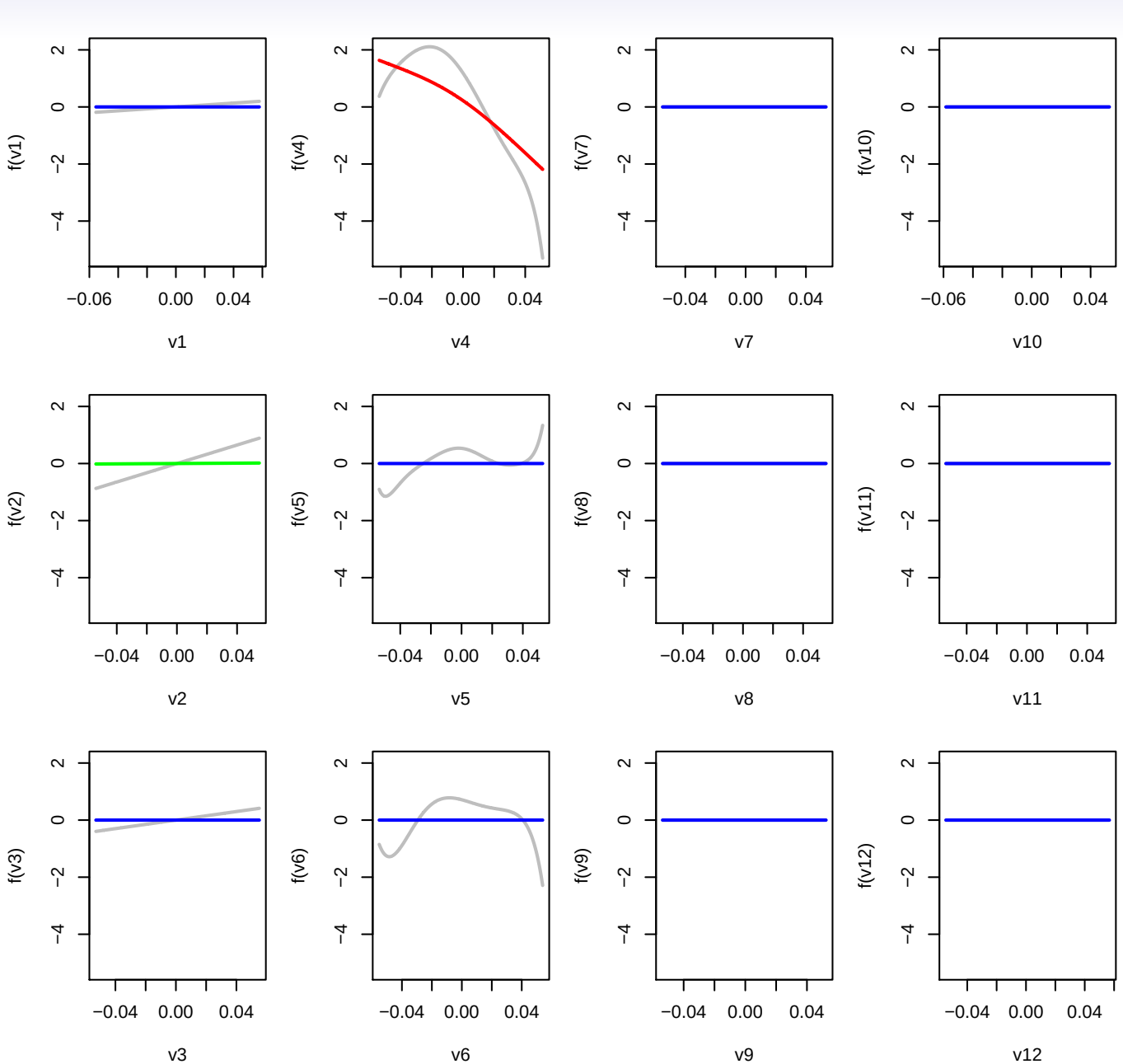

Step= 10 lambda = 53.83
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

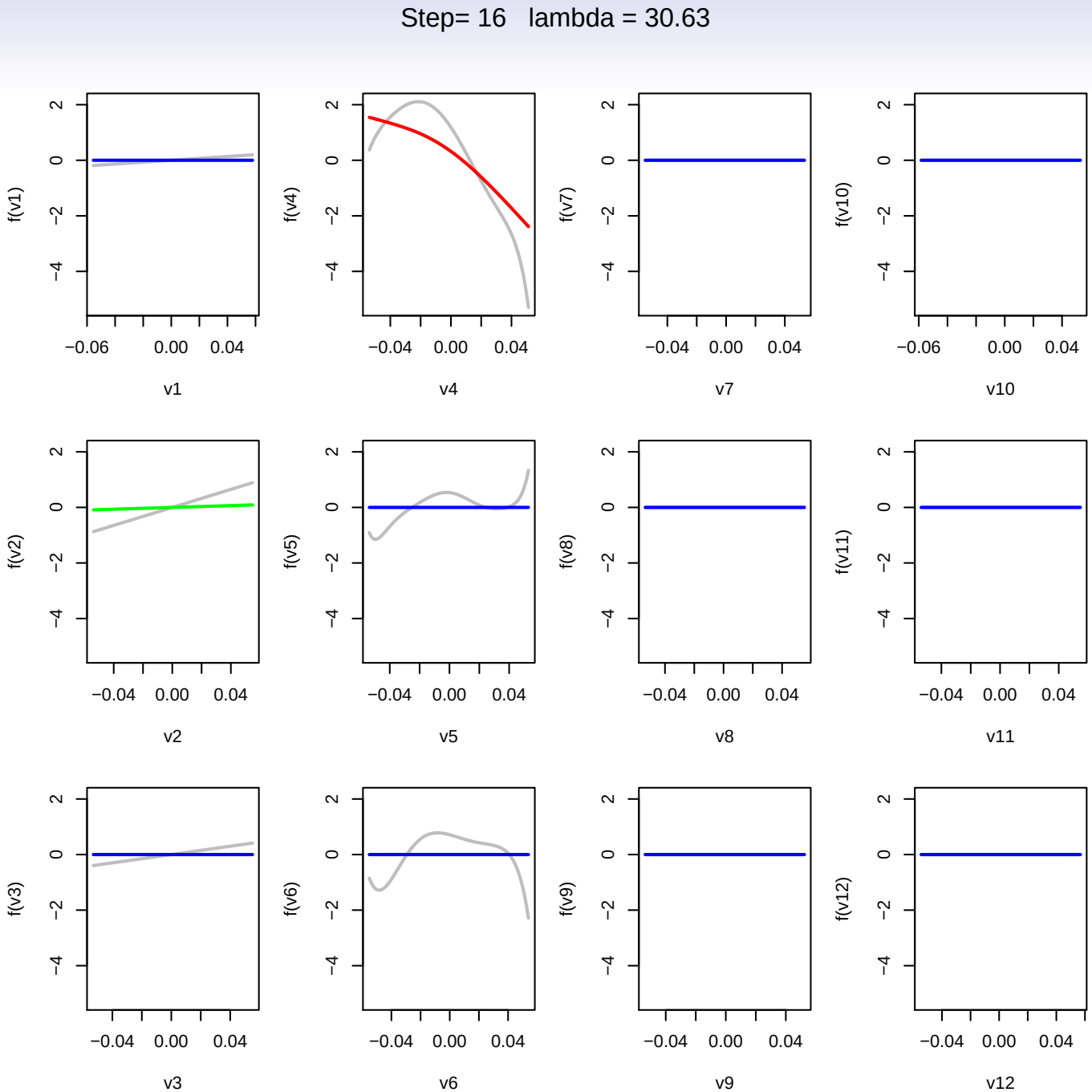

Step= 11 lambda = 49.01
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

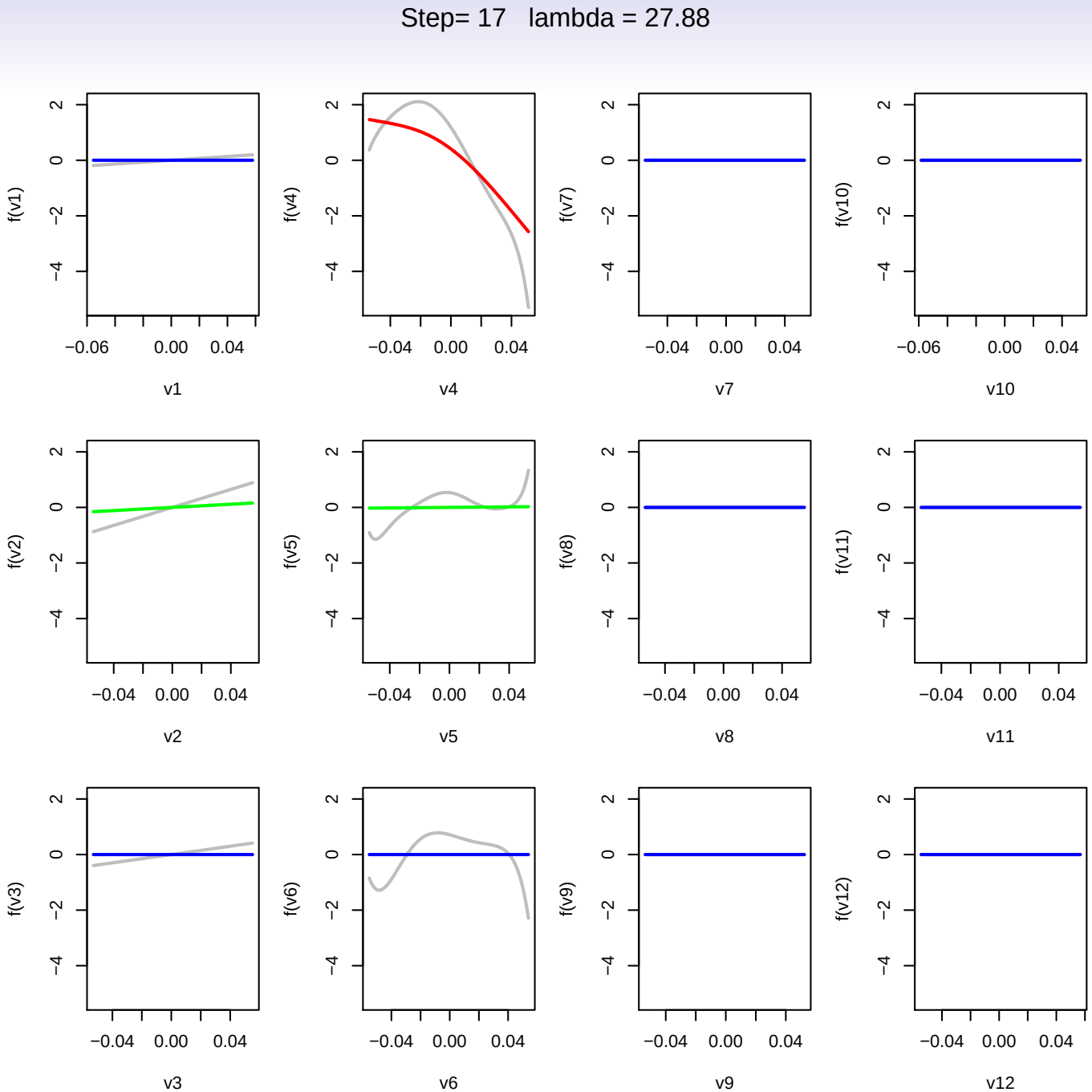

Step= 12 lambda = 44.61
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

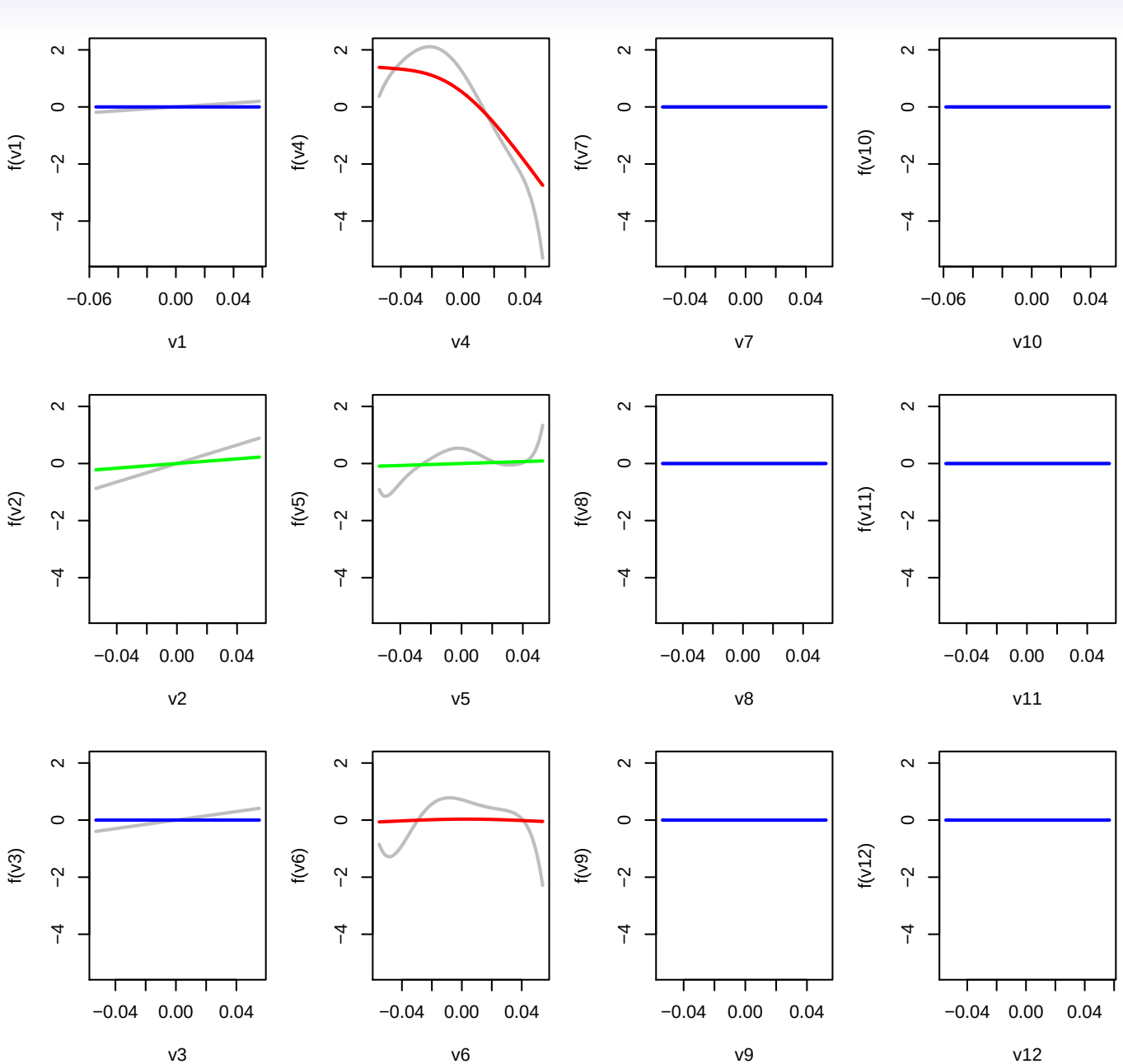

Step= 13 lambda = 40.61
f(v1)
f(v4)
f(v7)
f(v10)
f(v2)
f(v5)
f(v8)
f(v11)
f(v3)
f(v6)
f(v9)
f(v12)
v1
v4
v7
v10
v2
v5
v8
v11
v3
v6
v9
v12


Step= 14 lambda = 36.97
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

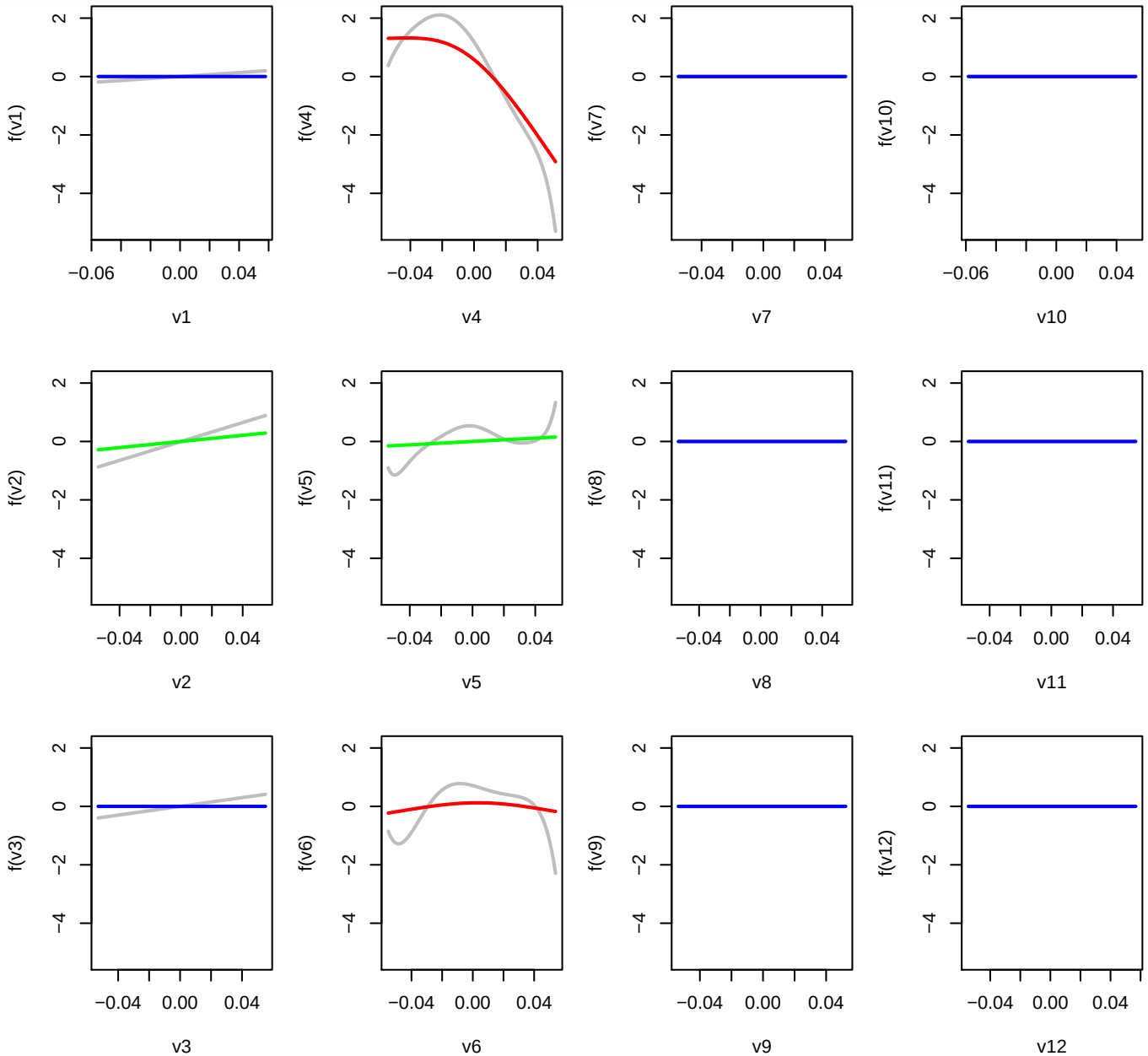

Step= 15 lambda = 33.65
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

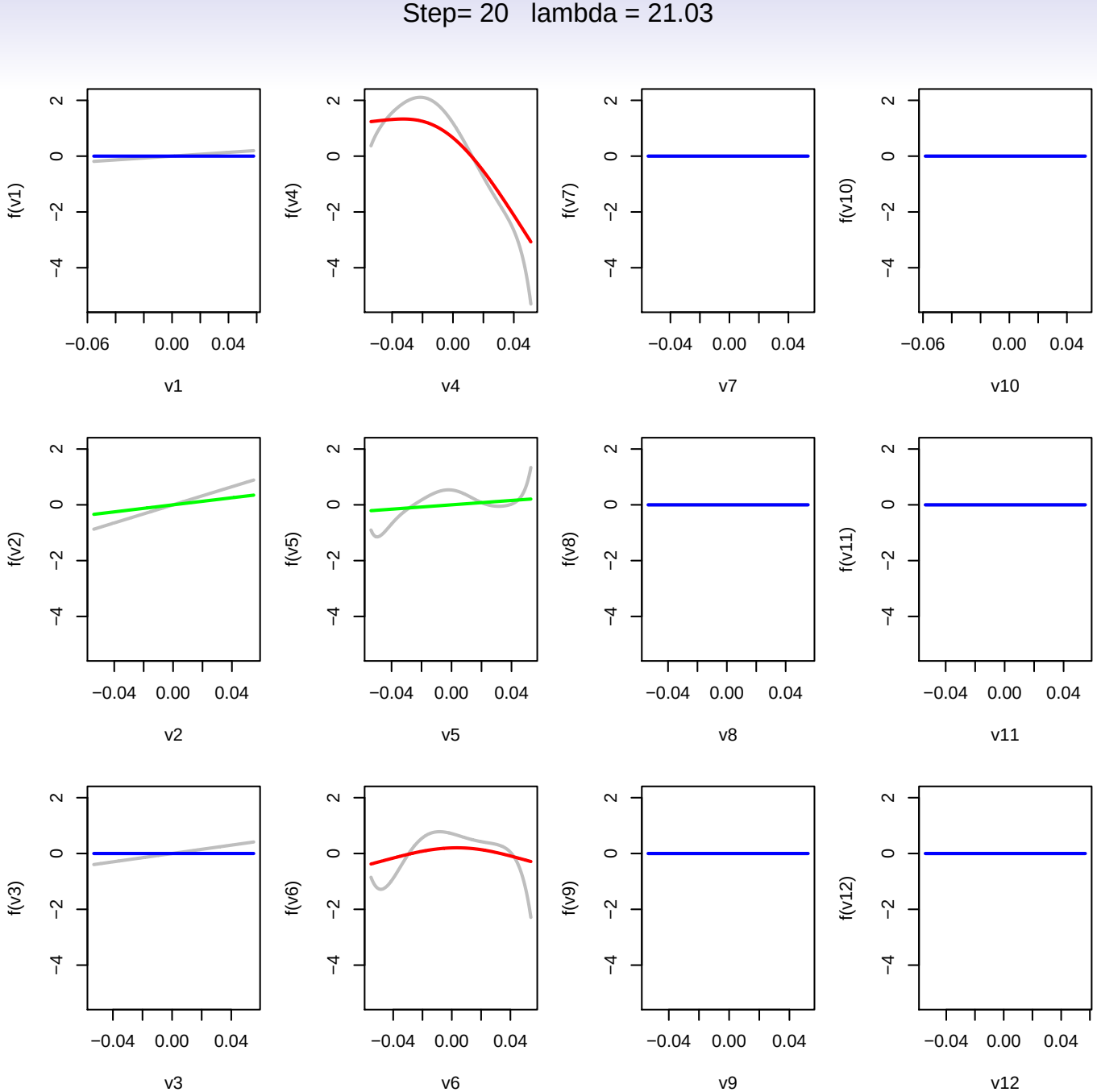

Step= 16 lambda = 30.63
f(v1)
f(v4)
f(v7)
f(v10)
f(v2)
f(v5)
f(v8)
f(v11)
f(v3)
f(v6)
f(v9)
f(v12)
v1
v4
v7
v10
v2
v5
v8
v11
v3
v6
v9
v12

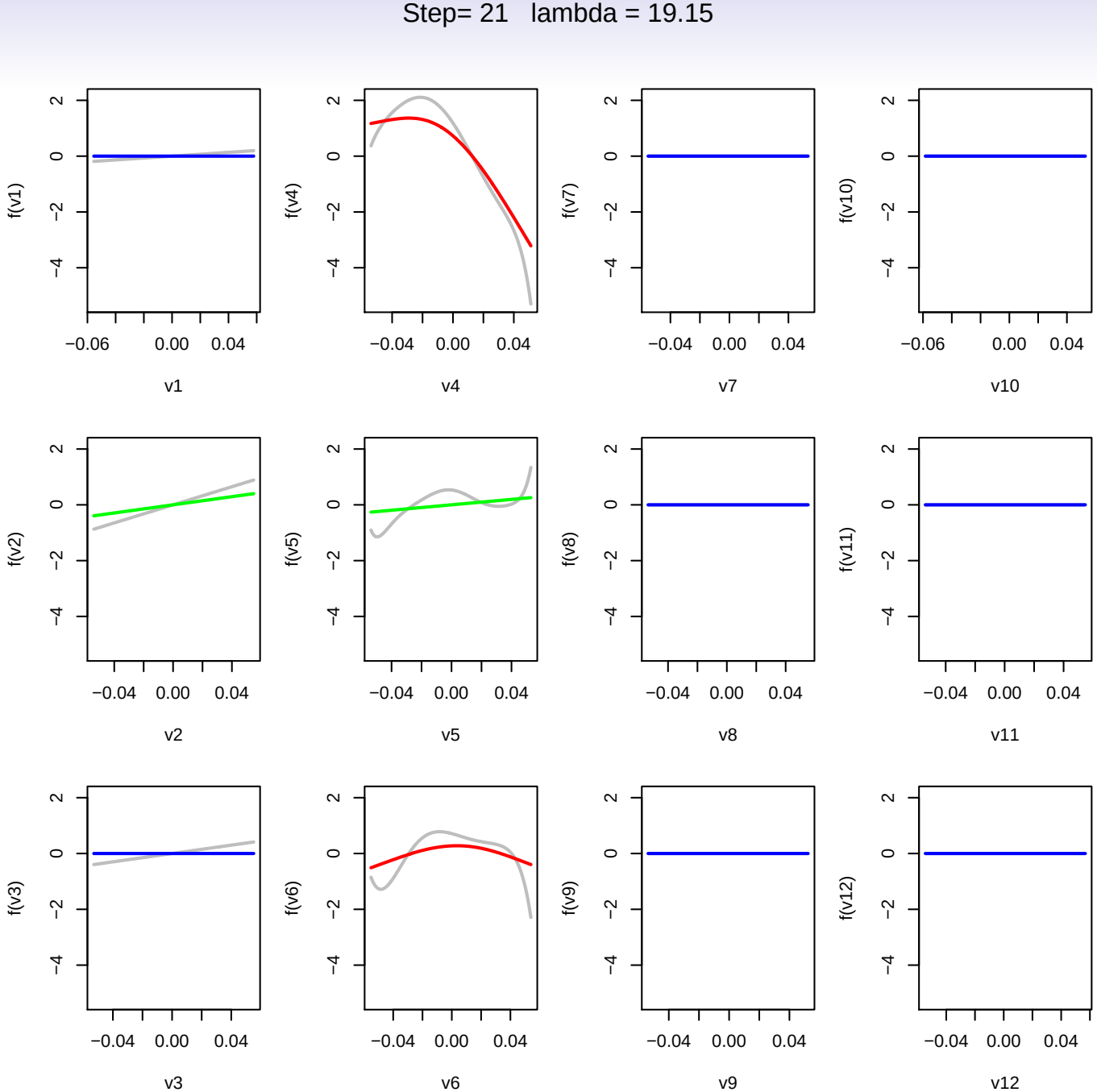

Step= 17 lambda = 27.88
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12


Step= 18 lambda = 25.38
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

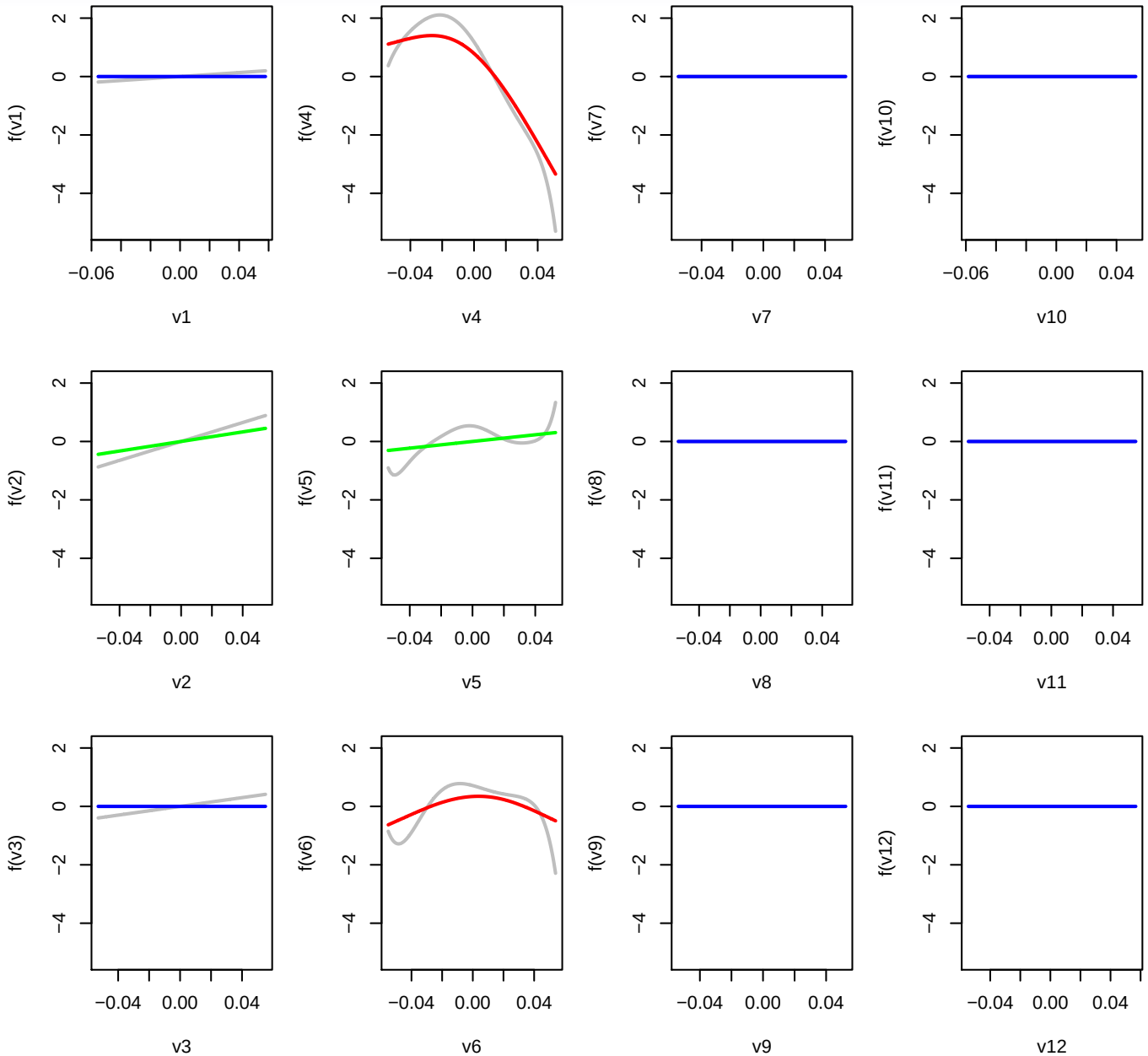
Step= 19 lambda = 23.11

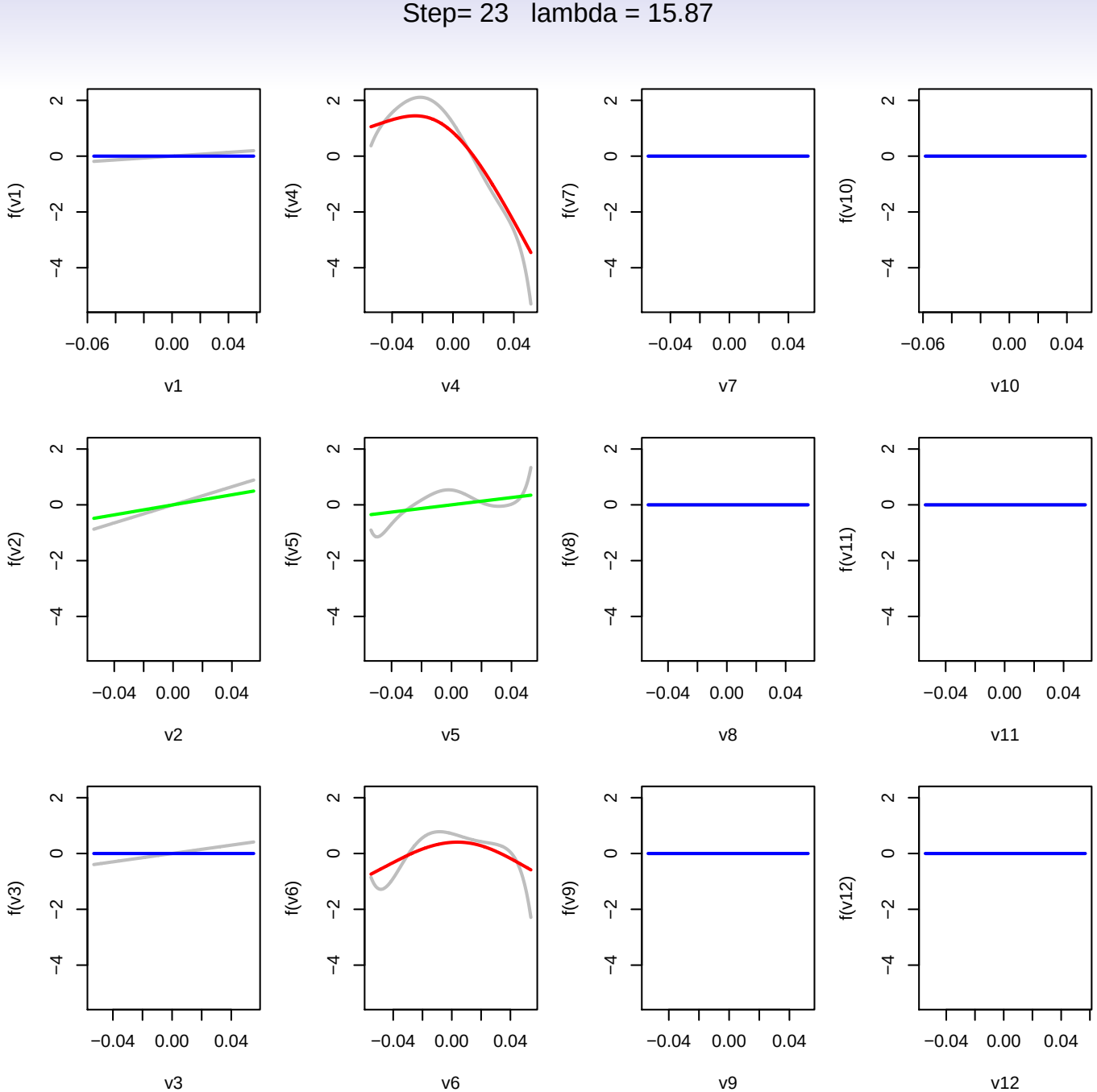


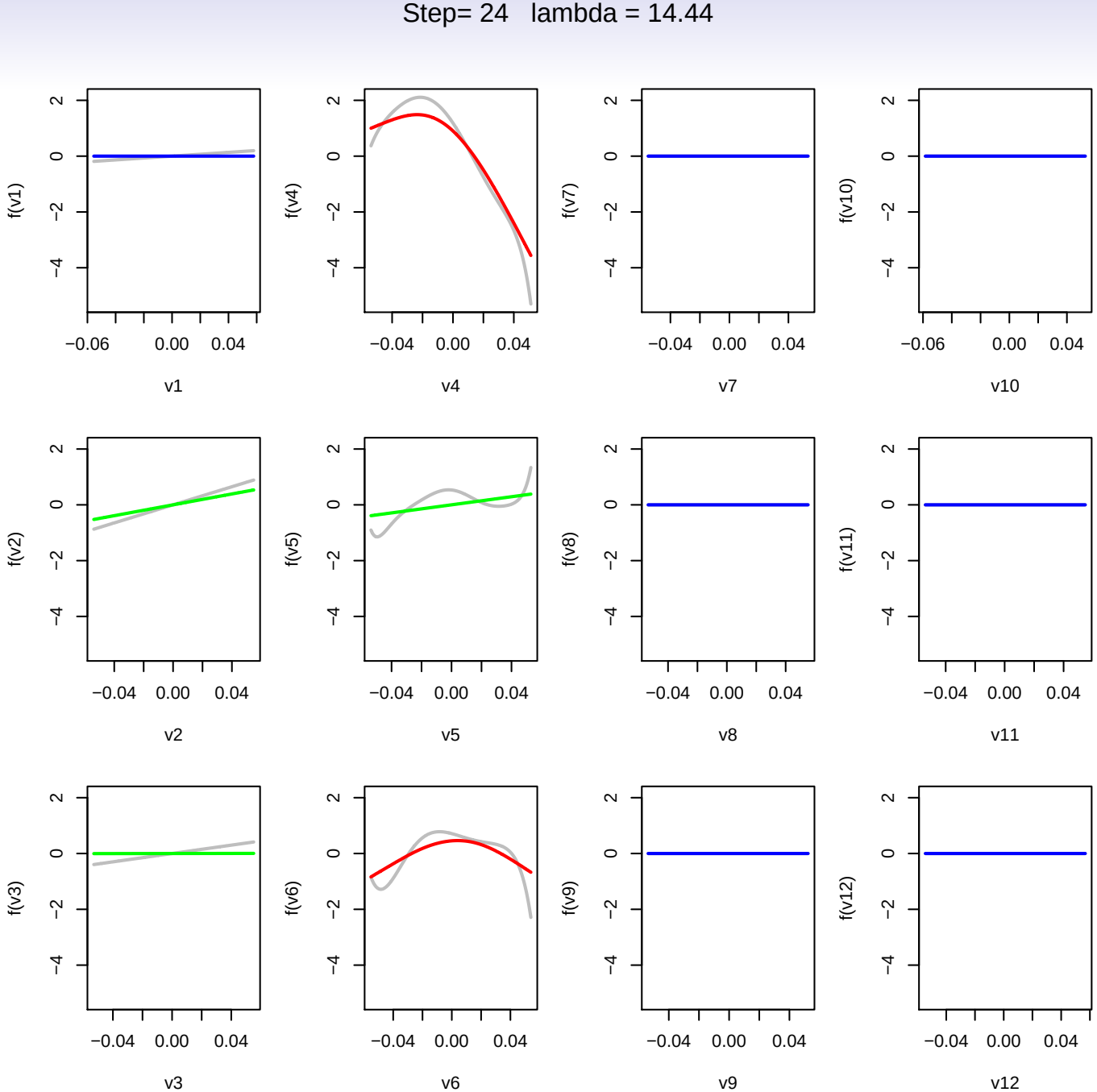

Step= 20 lambda = 21.03
f(v1)
f(v4)
f(v7)
f(v10)
f(v2)
f(v5)
f(v8)
f(v11)
f(v3)
f(v6)
f(v9)
f(v12)
v1
v4
v7
v10
v2
v5
v8
v11
v3
v6
v9
v12

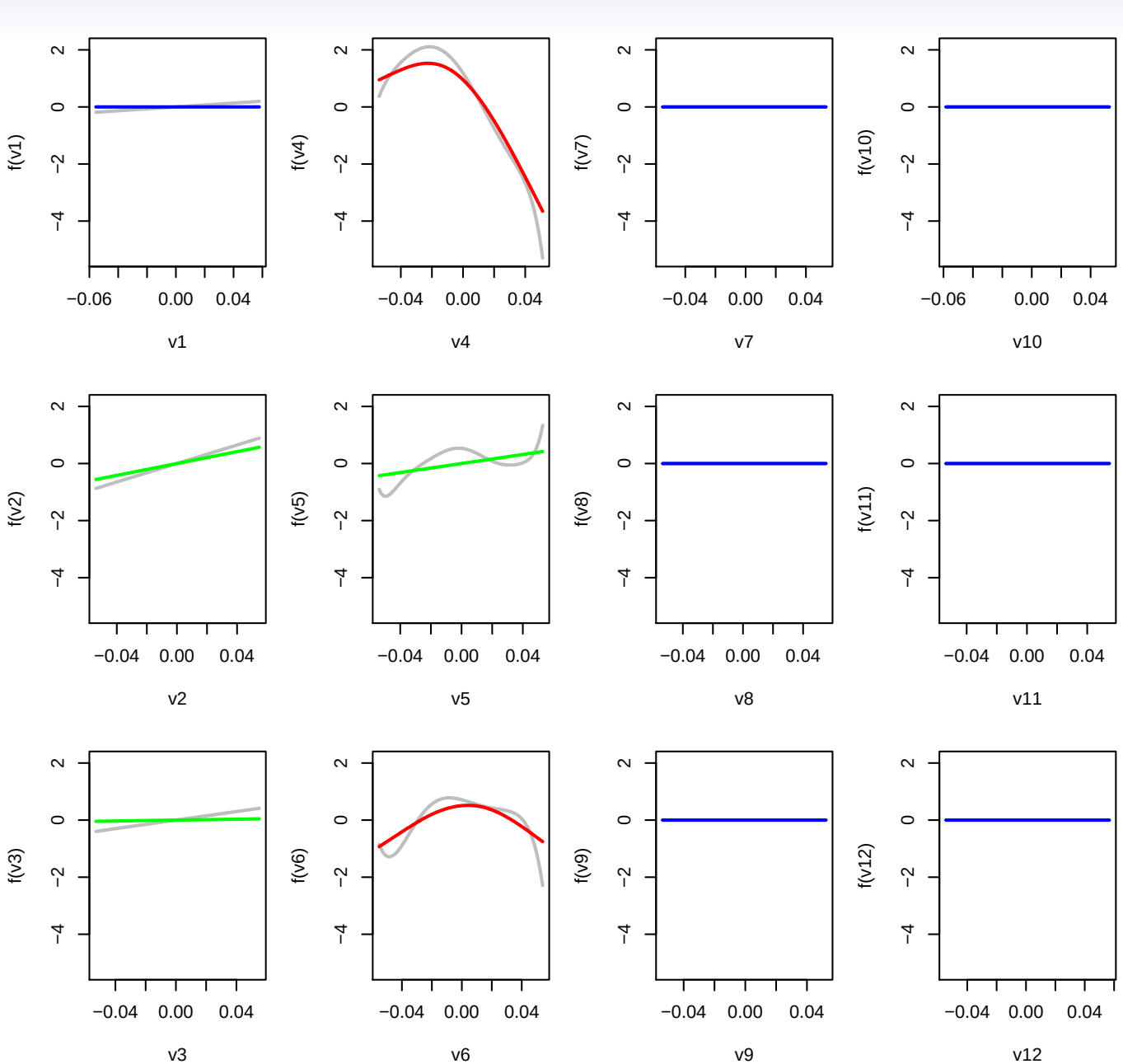

Step= 21 lambda = 19.15
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

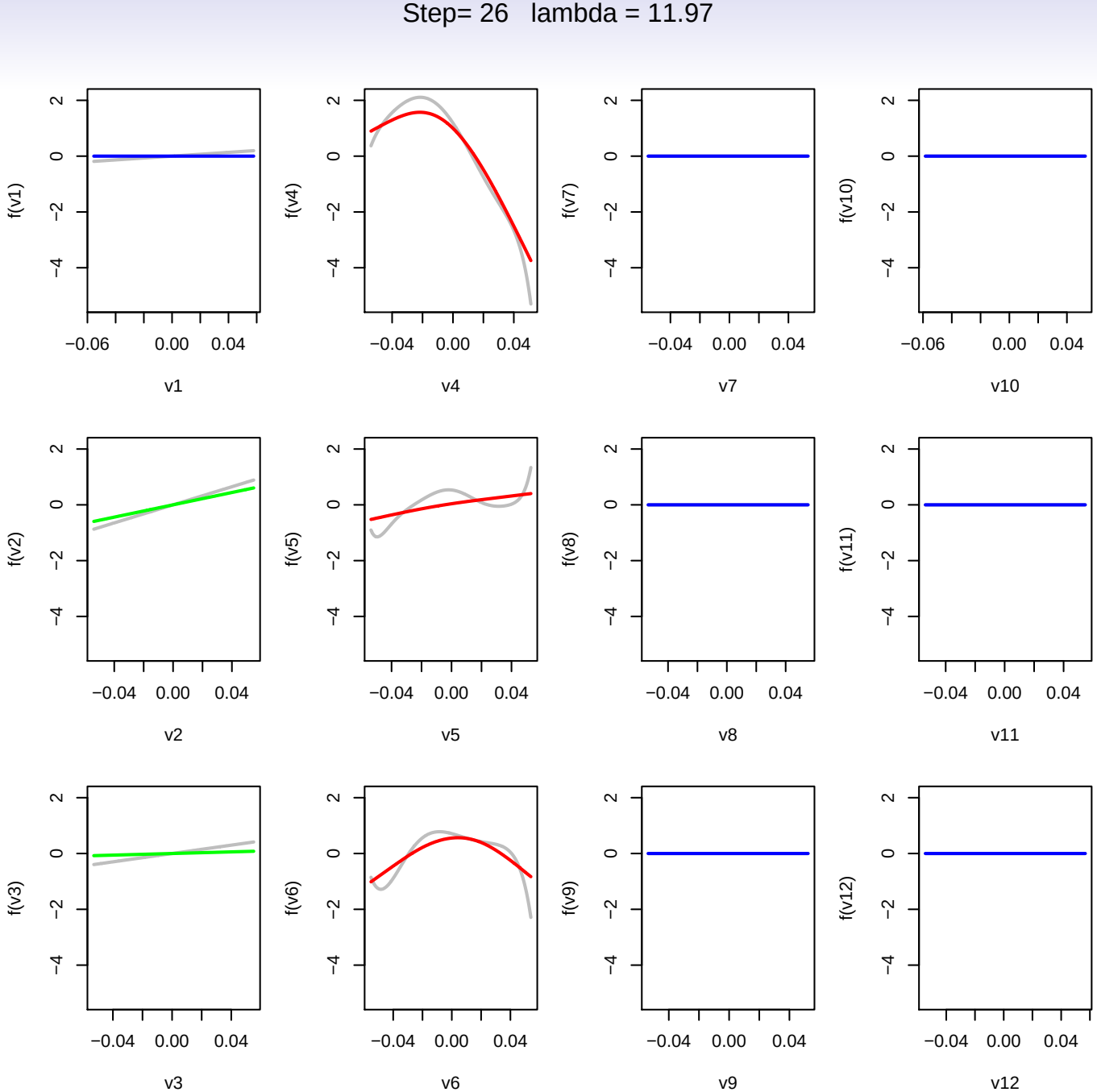
Step= 22 lambda = 17.43

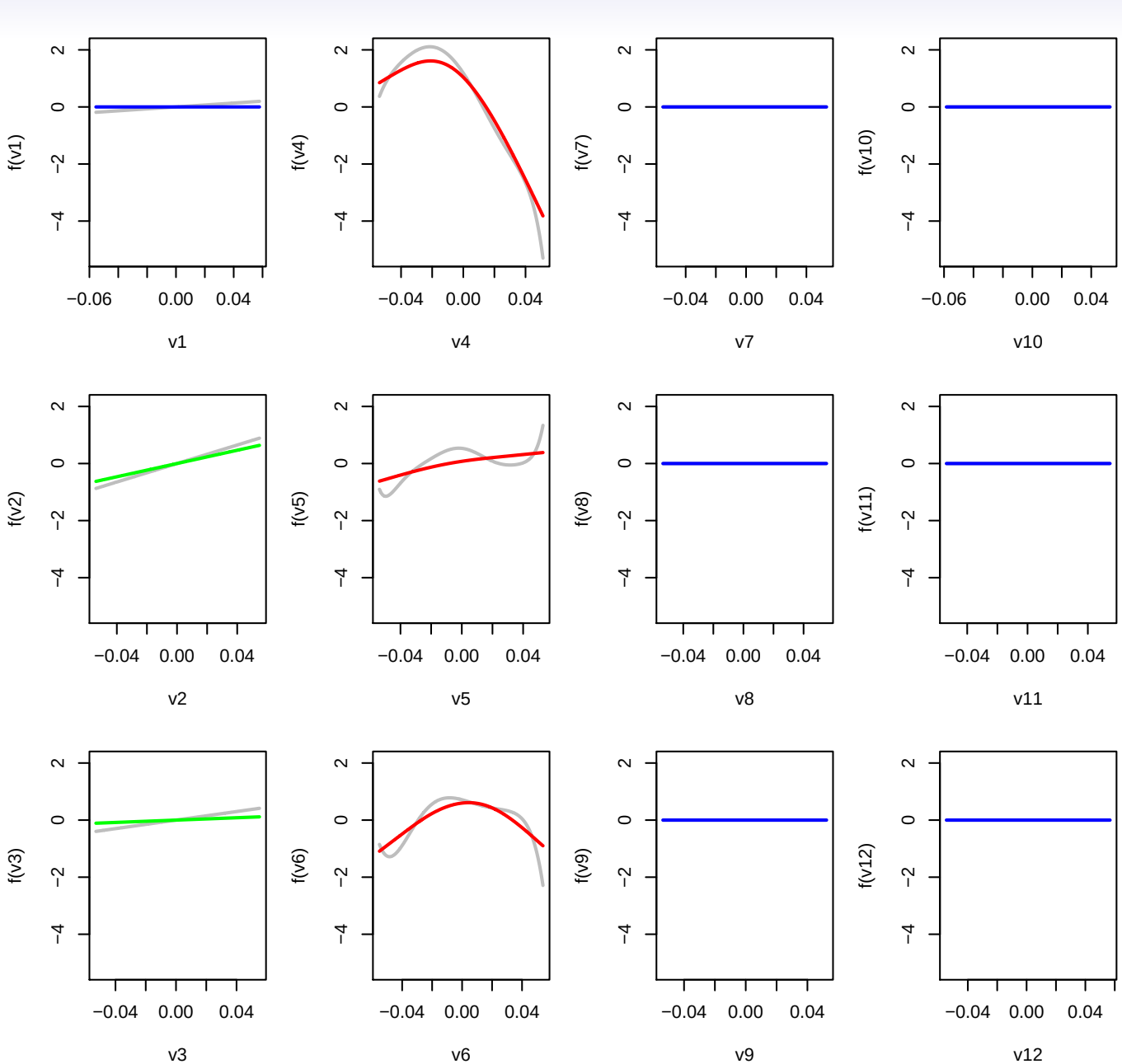


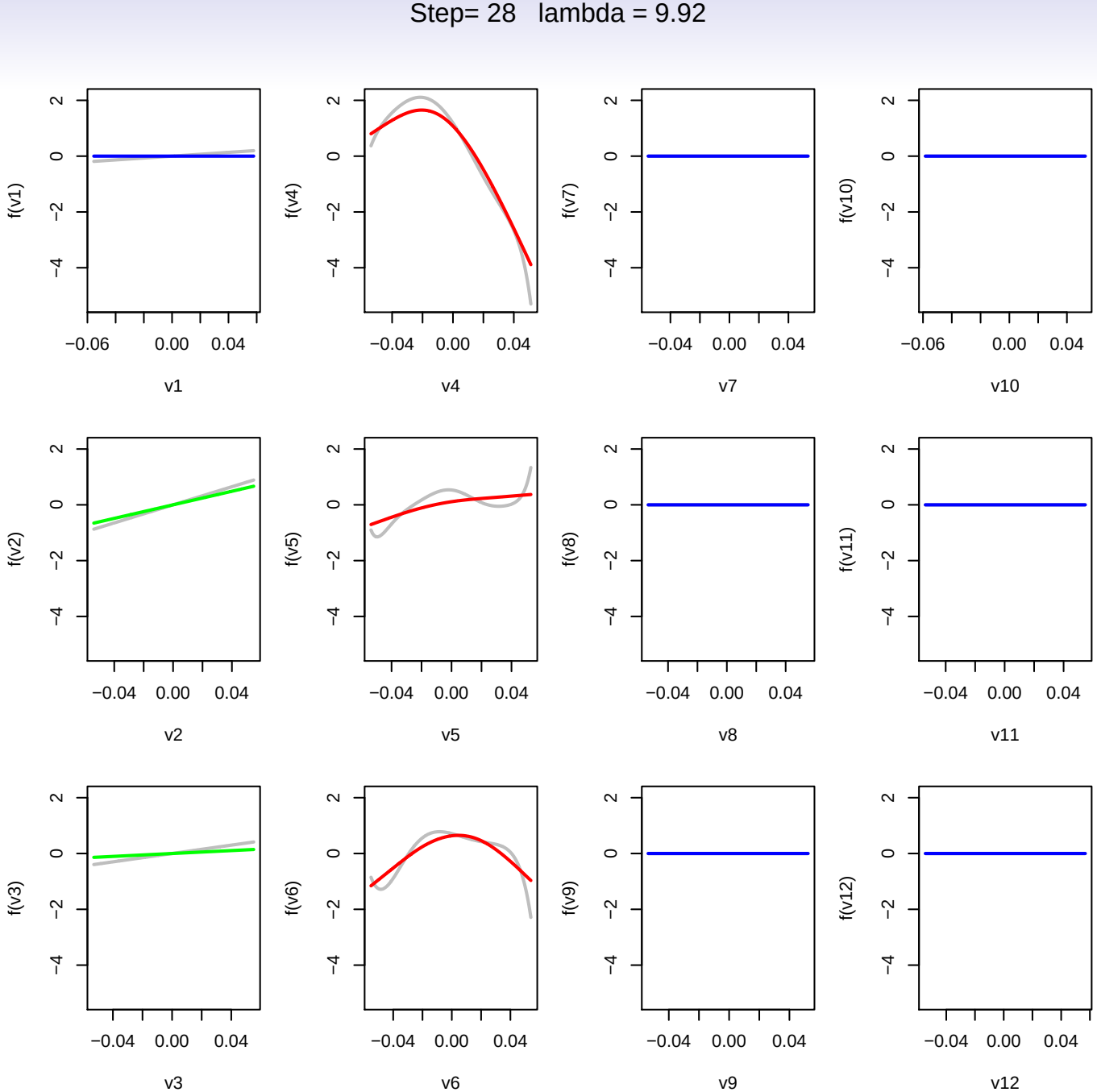

Step= 23 lambda = 15.87
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

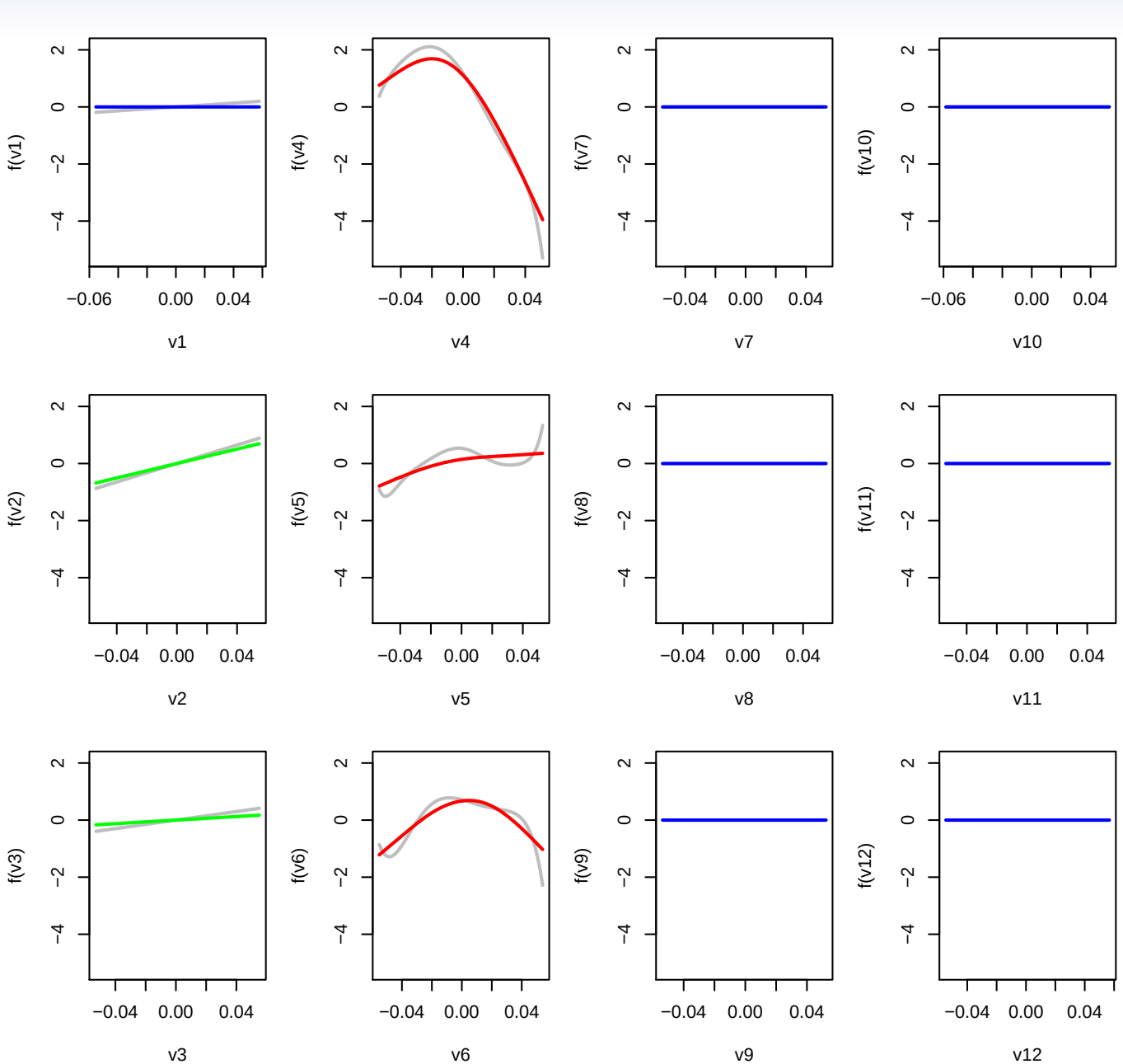

Step= 24 lambda = 14.44
f(v1)
f(v4)
f(v7)
f(v10)
f(v2)
f(v5)
f(v8)
f(v11)
f(v3)
f(v6)
f(v9)
f(v12)
v1
v4
v7
v10
v2
v5
v8
v11
v3
v6
v9
v12

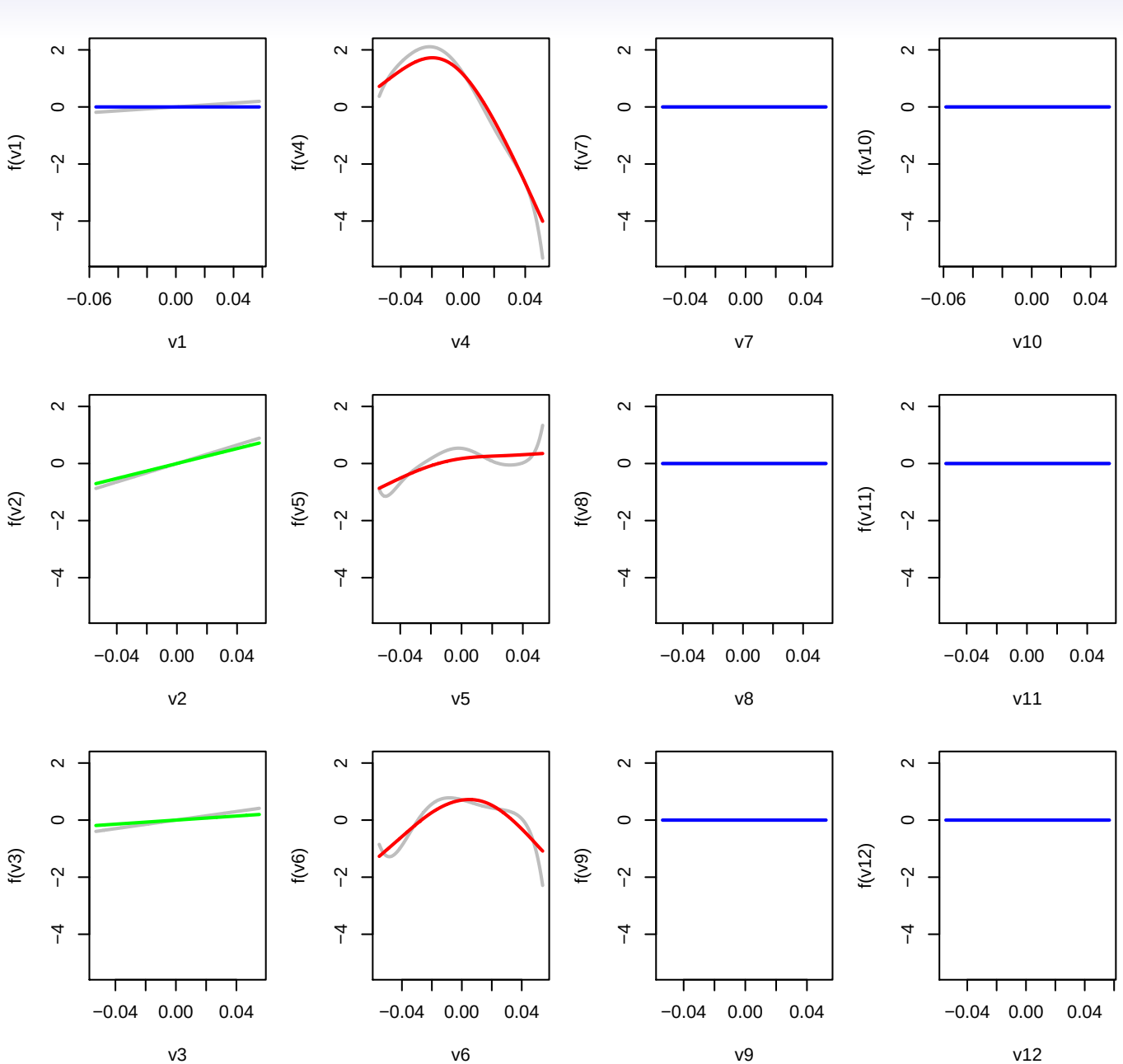

Step= 25 lambda = 13.15
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

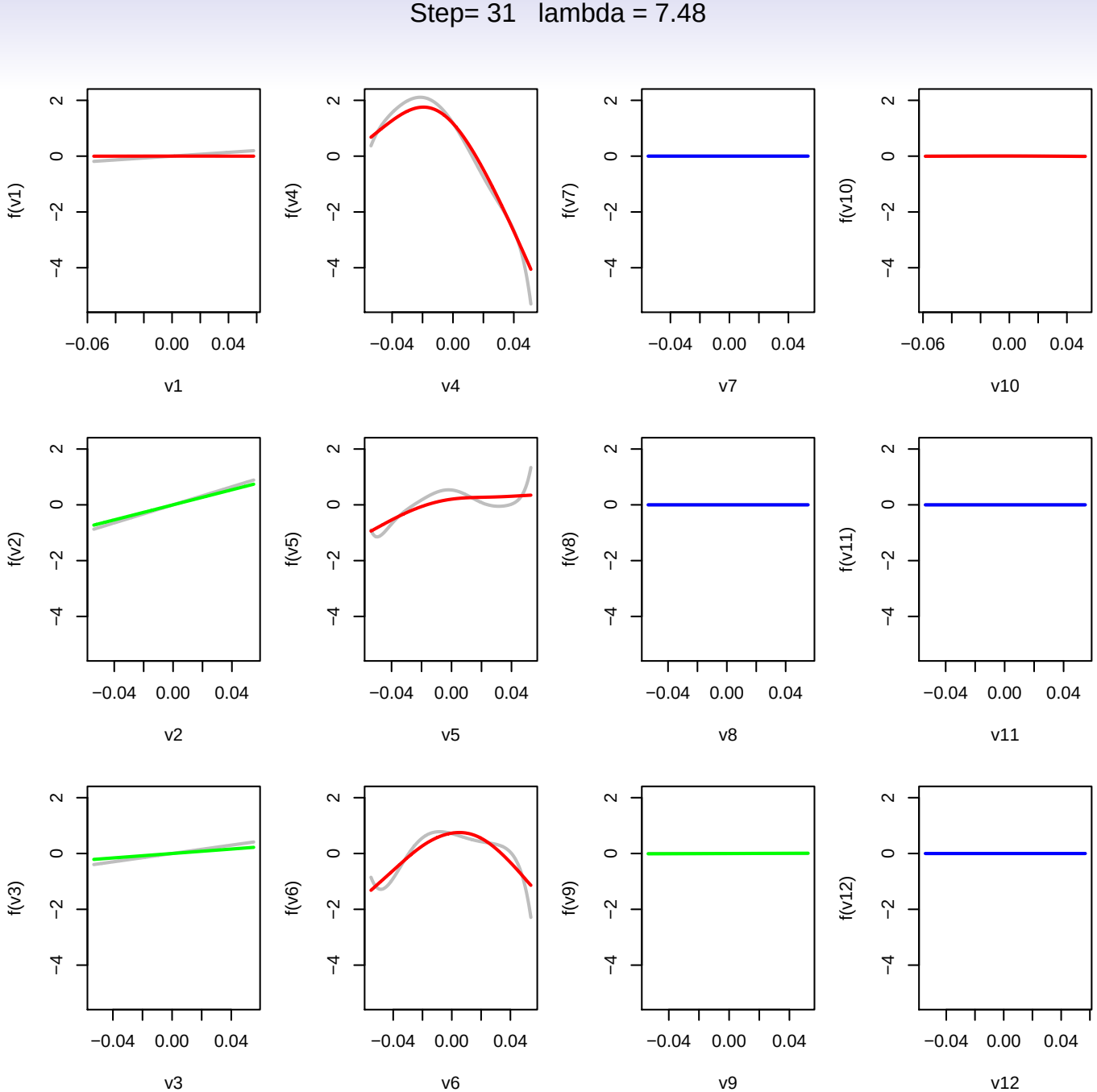

Step= 26 lambda = 11.97
f(v1)
f(v4)
f(v7)
f(v10)
f(v2)
f(v5)
f(v8)
f(v11)
f(v3)
f(v6)
f(v9)
f(v12)
v1
v4
v7
v10
v2
v5
v8
v11
v3
v6
v9
v12


Step= 27 lambda = 10.89
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

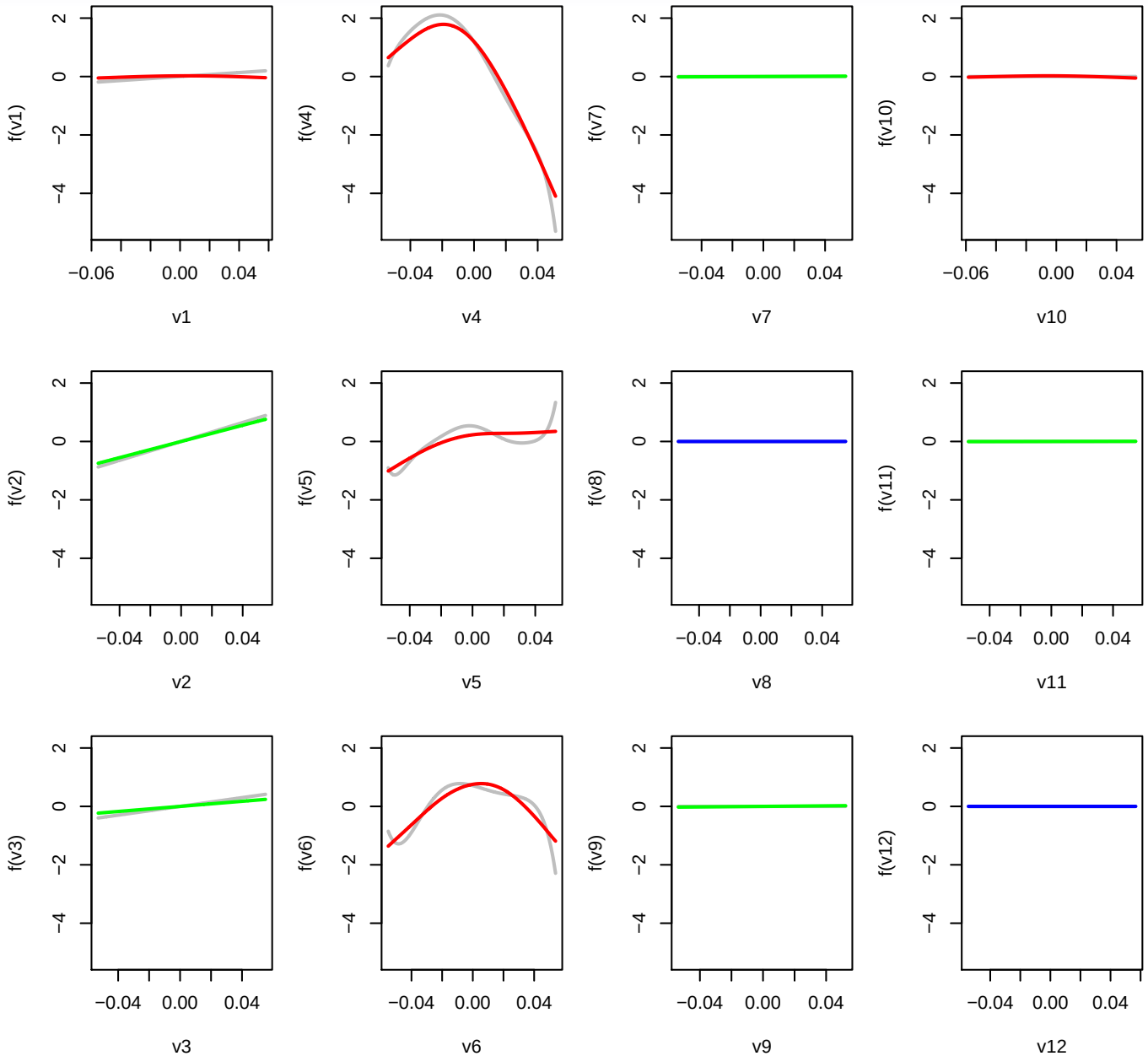

Step= 28 lambda = 9.92
f(v1)
f(v4)
f(v7)
f(v10)
f(v2)
f(v5)
f(v8)
f(v11)
f(v3)
f(v6)
f(v9)
f(v12)
v1
v4
v7
v10
v2
v5
v8
v11
v3
v6
v9
v12

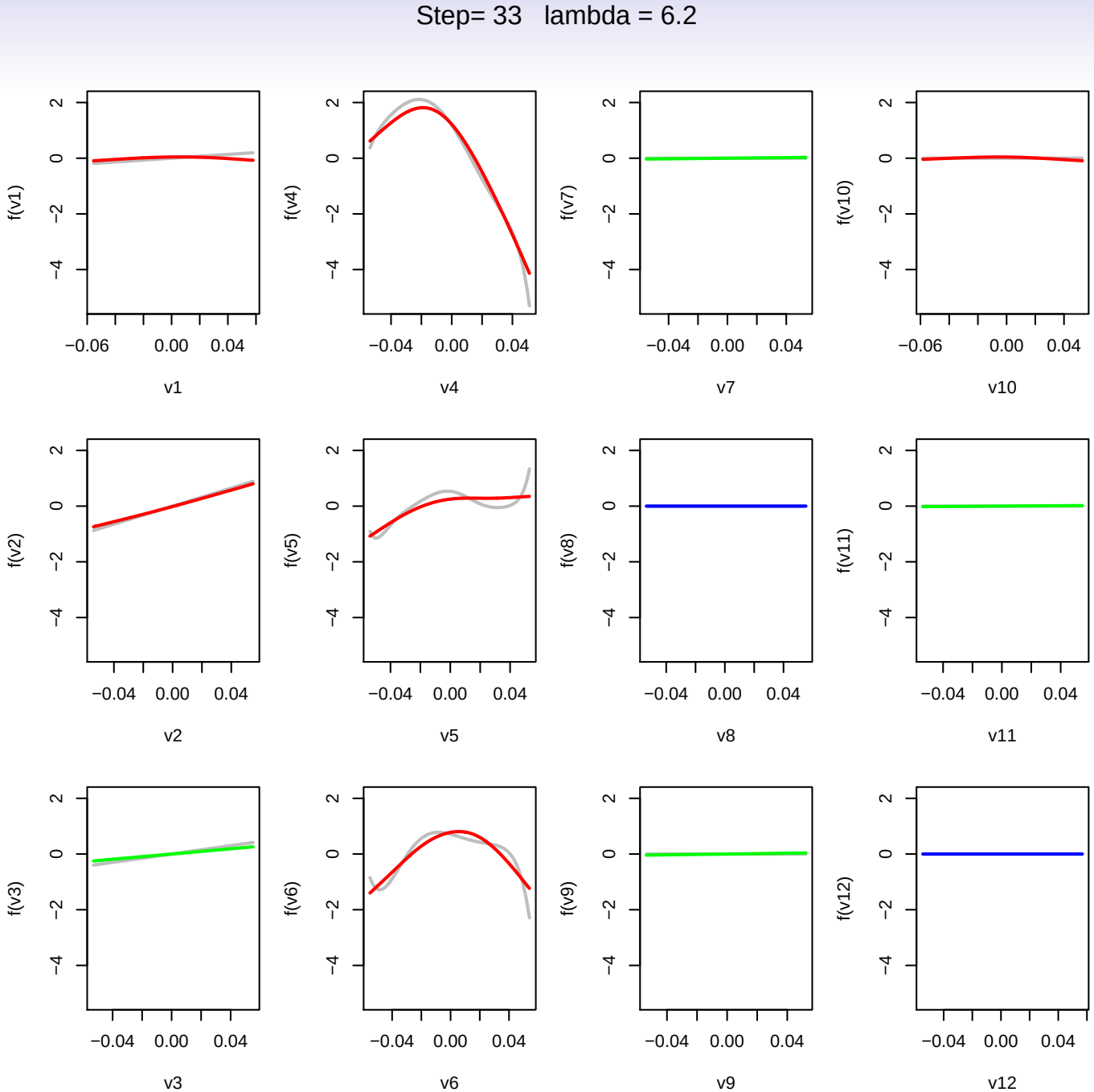

Step= 29 lambda = 9.03
f(v1)
f(v4)
f(v7)
f(v10)
f(v2)
f(v5)
f(v8)
f(v11)
f(v3)
f(v6)
f(v9)
f(v12)
v1
v4
v7
v10
v2
v5
v8
v11
v3
v6
v9
v12

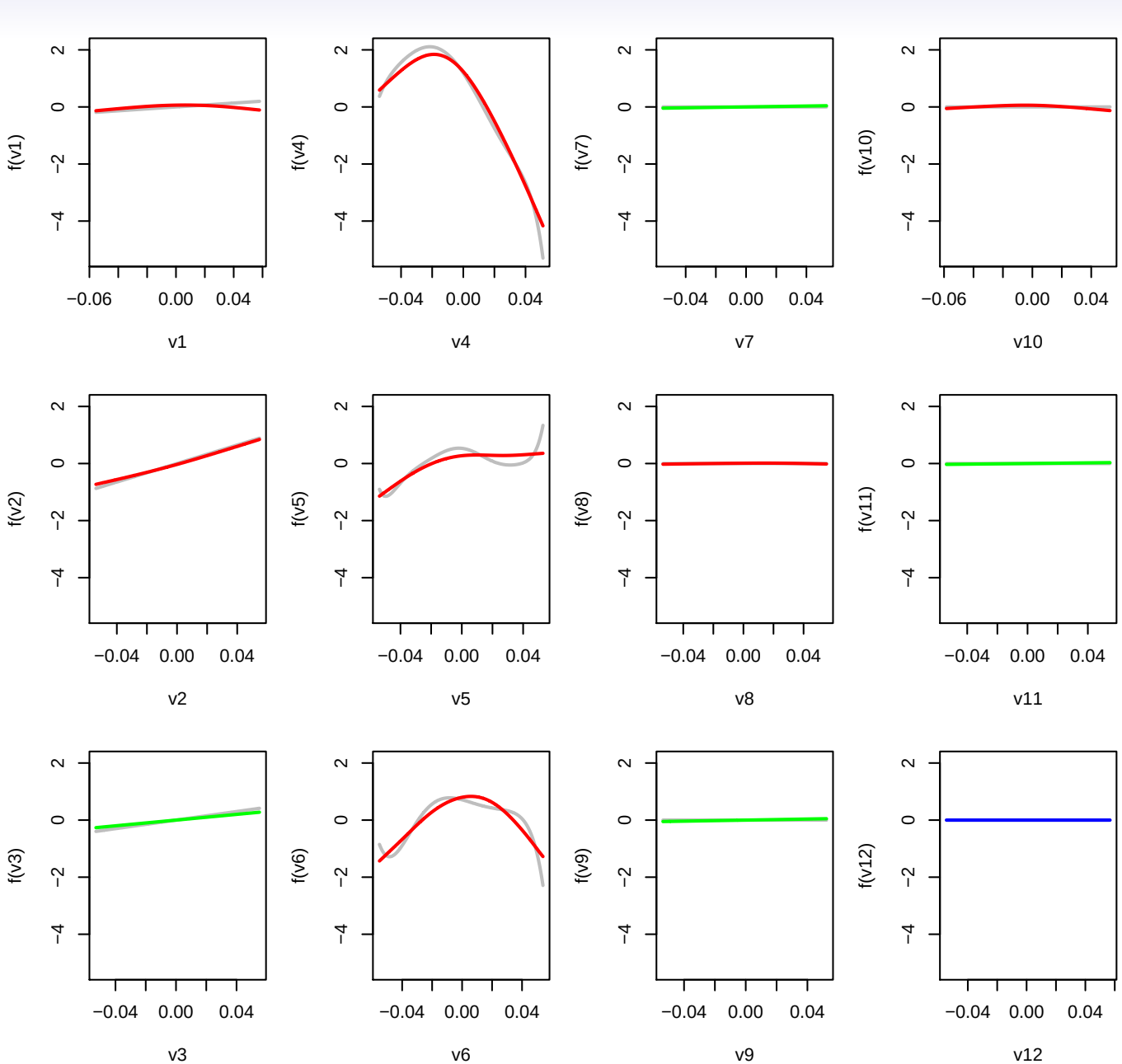

Step= 30 lambda = 8.22
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

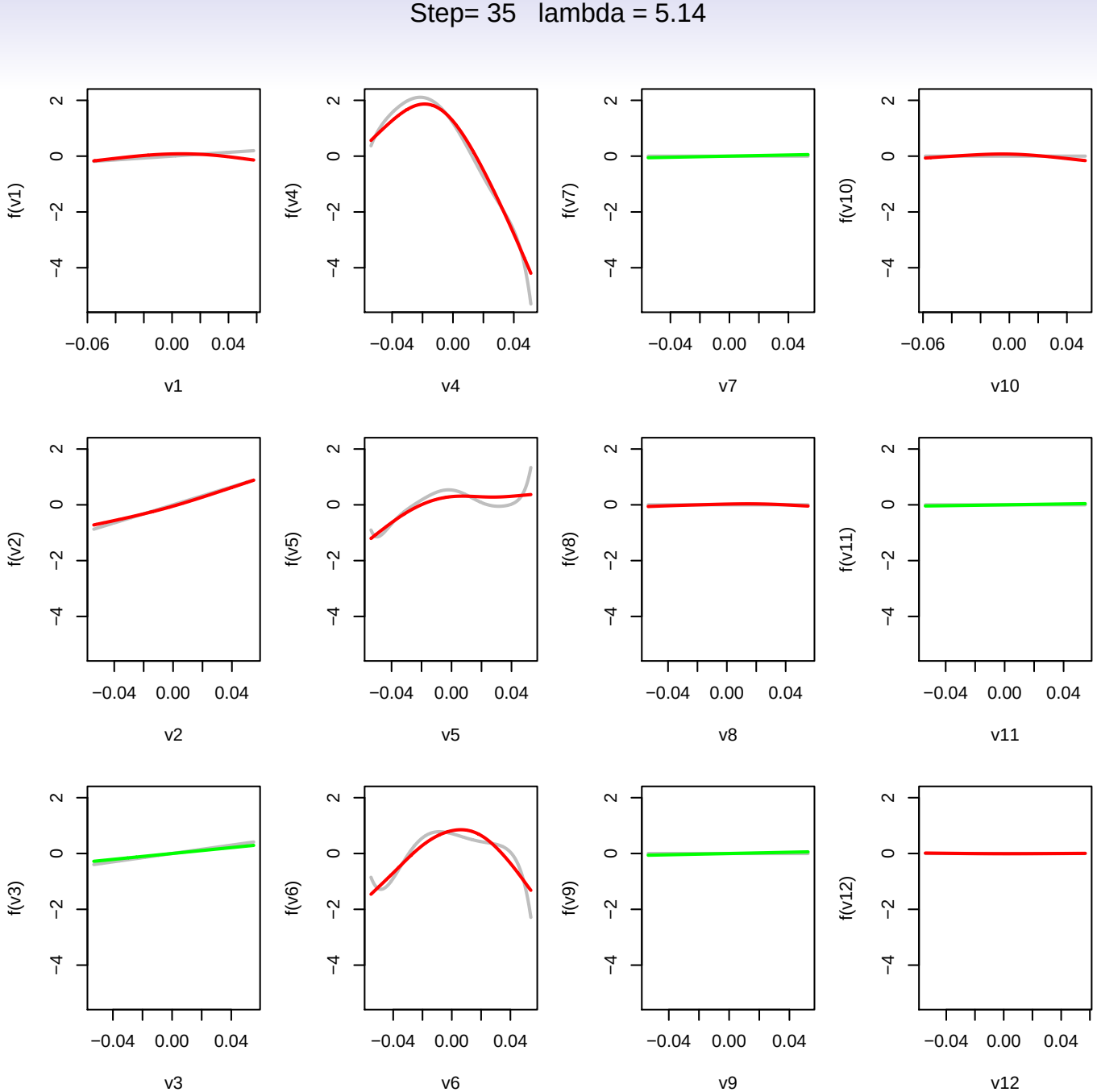

Step= 31 lambda = 7.48
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

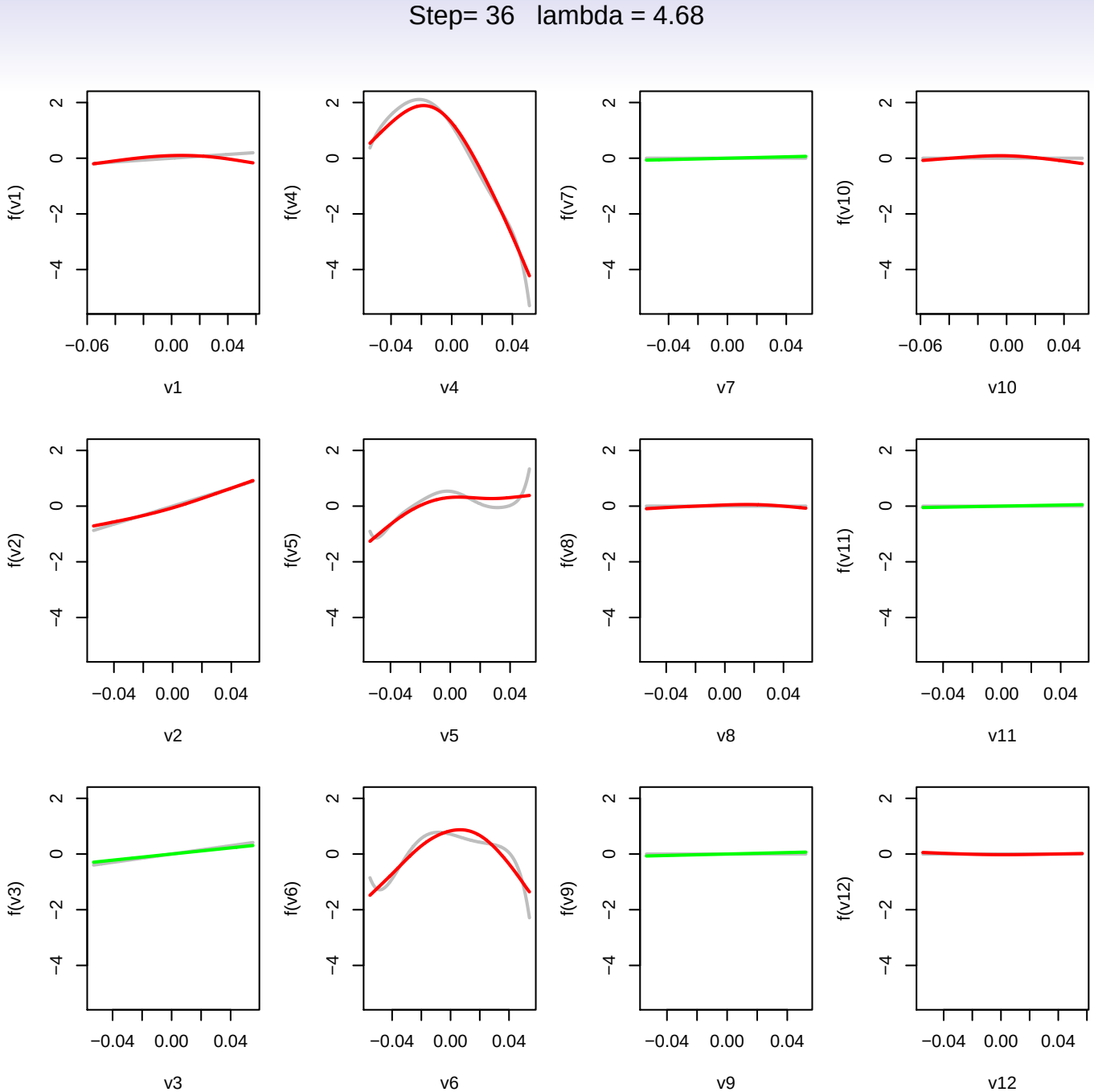
Step= 32 lambda = 6.81

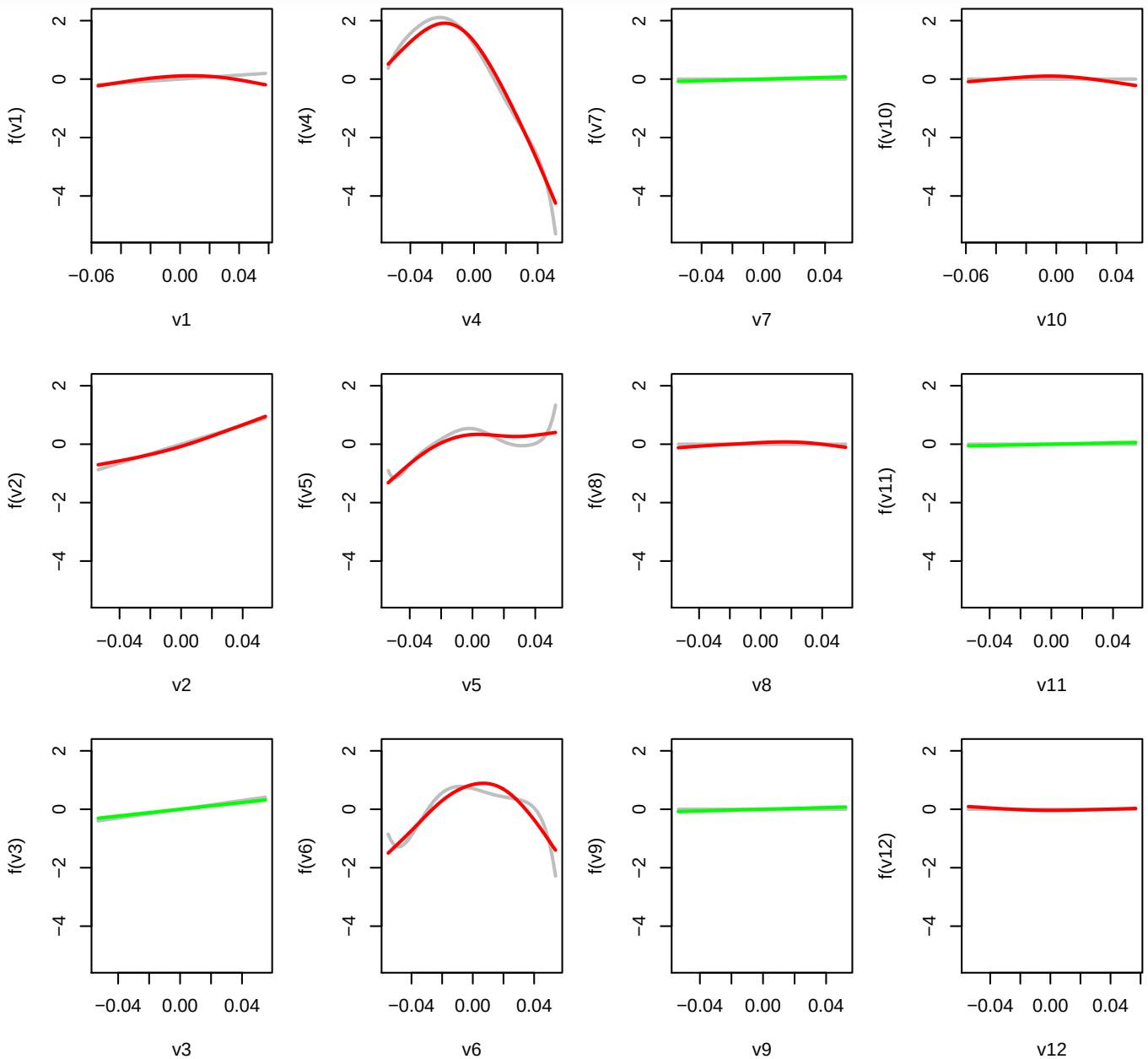


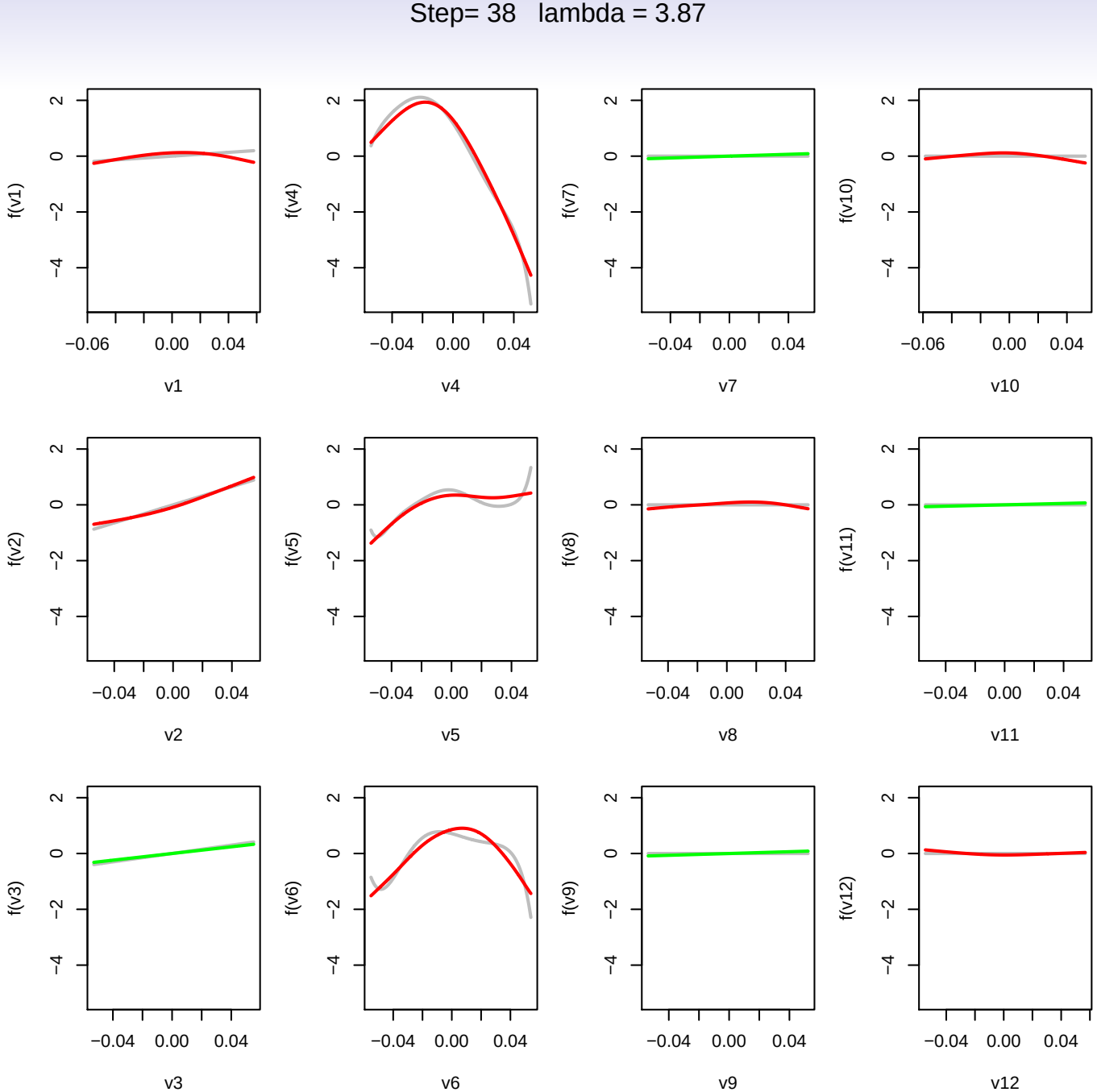

Step= 33 lambda = 6.2
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

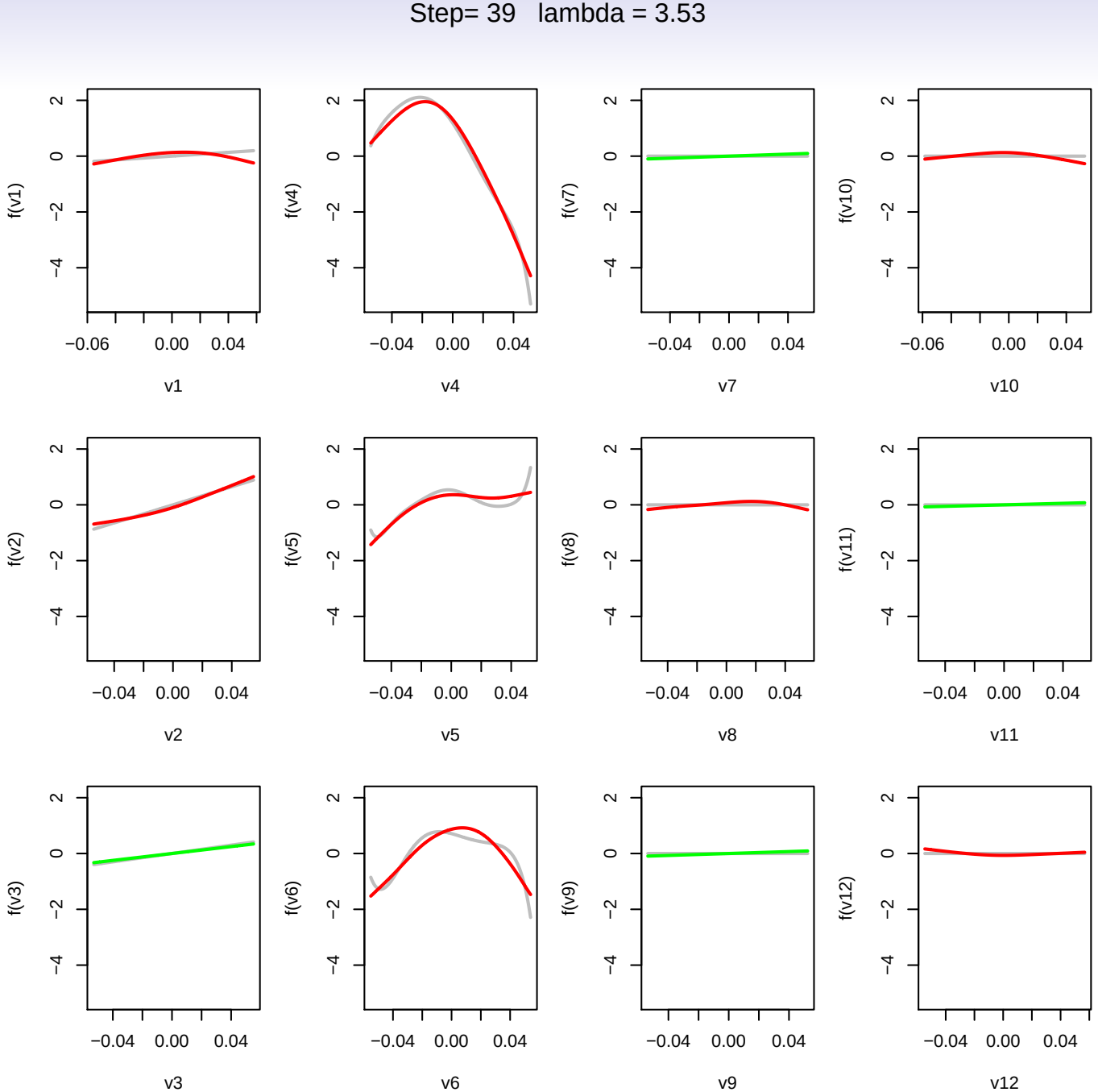

Step= 34 lambda = 5.64
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

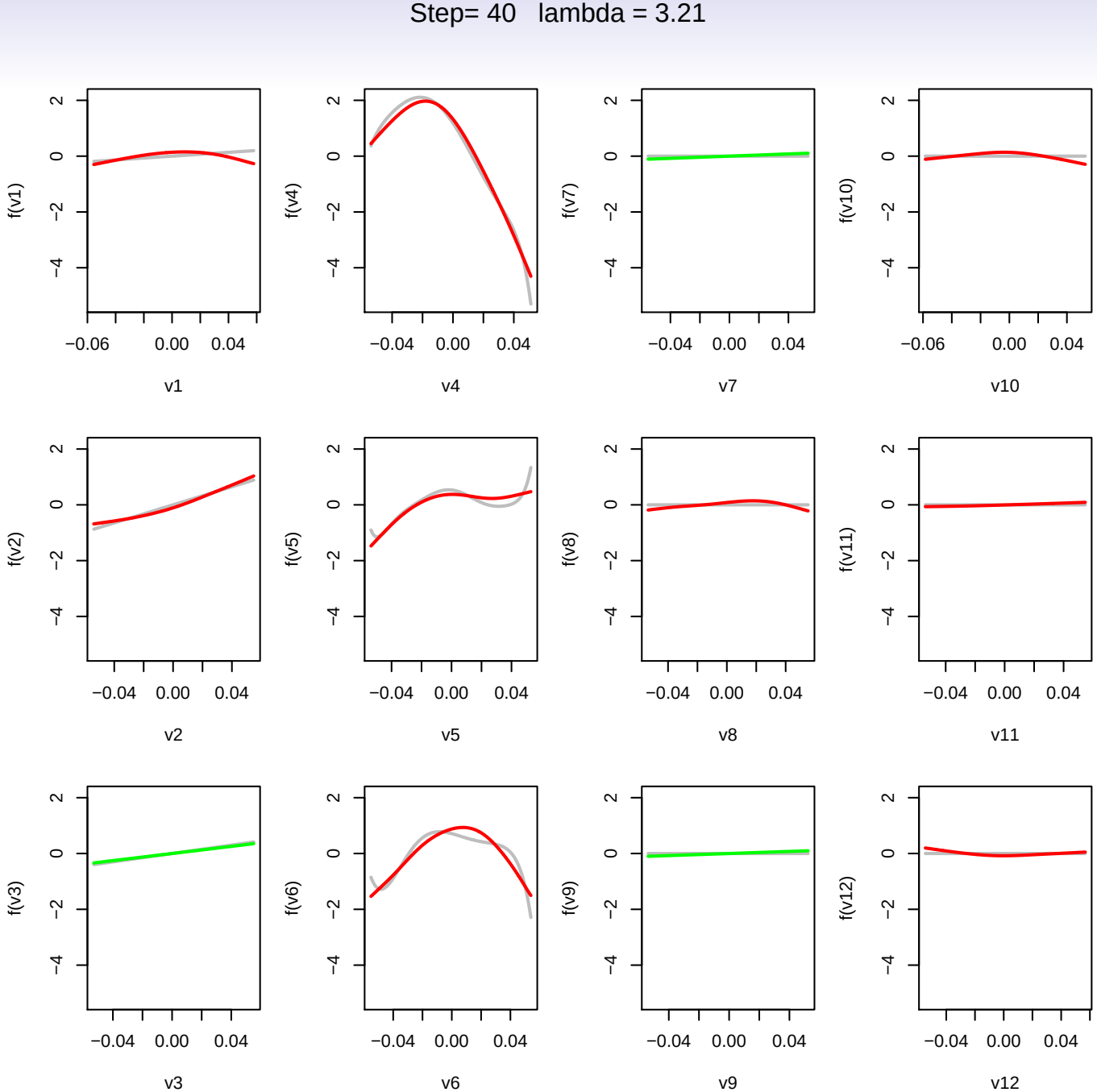

Step= 35 lambda = 5.14
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

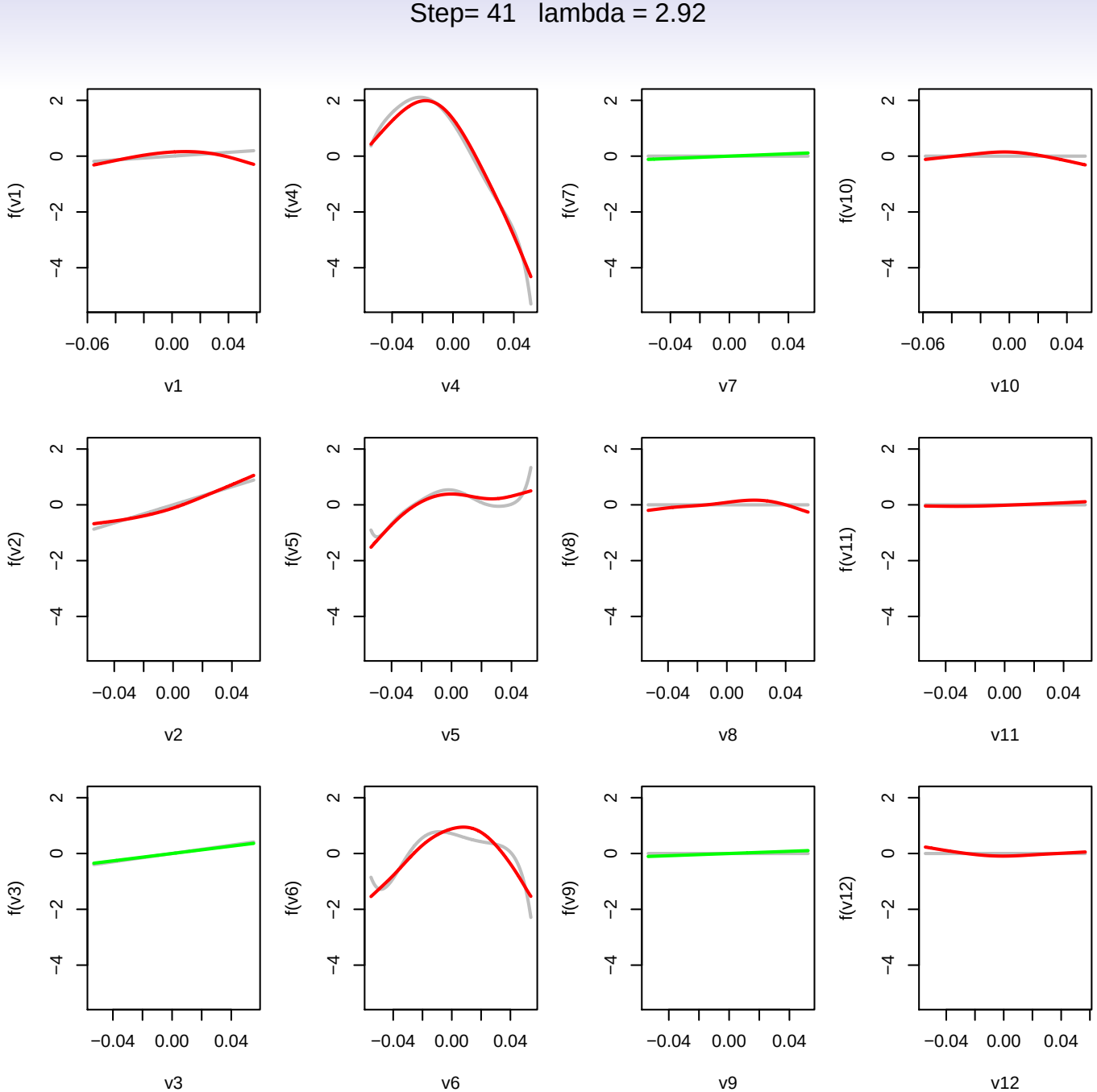

Step= 36 lambda = 4.68
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

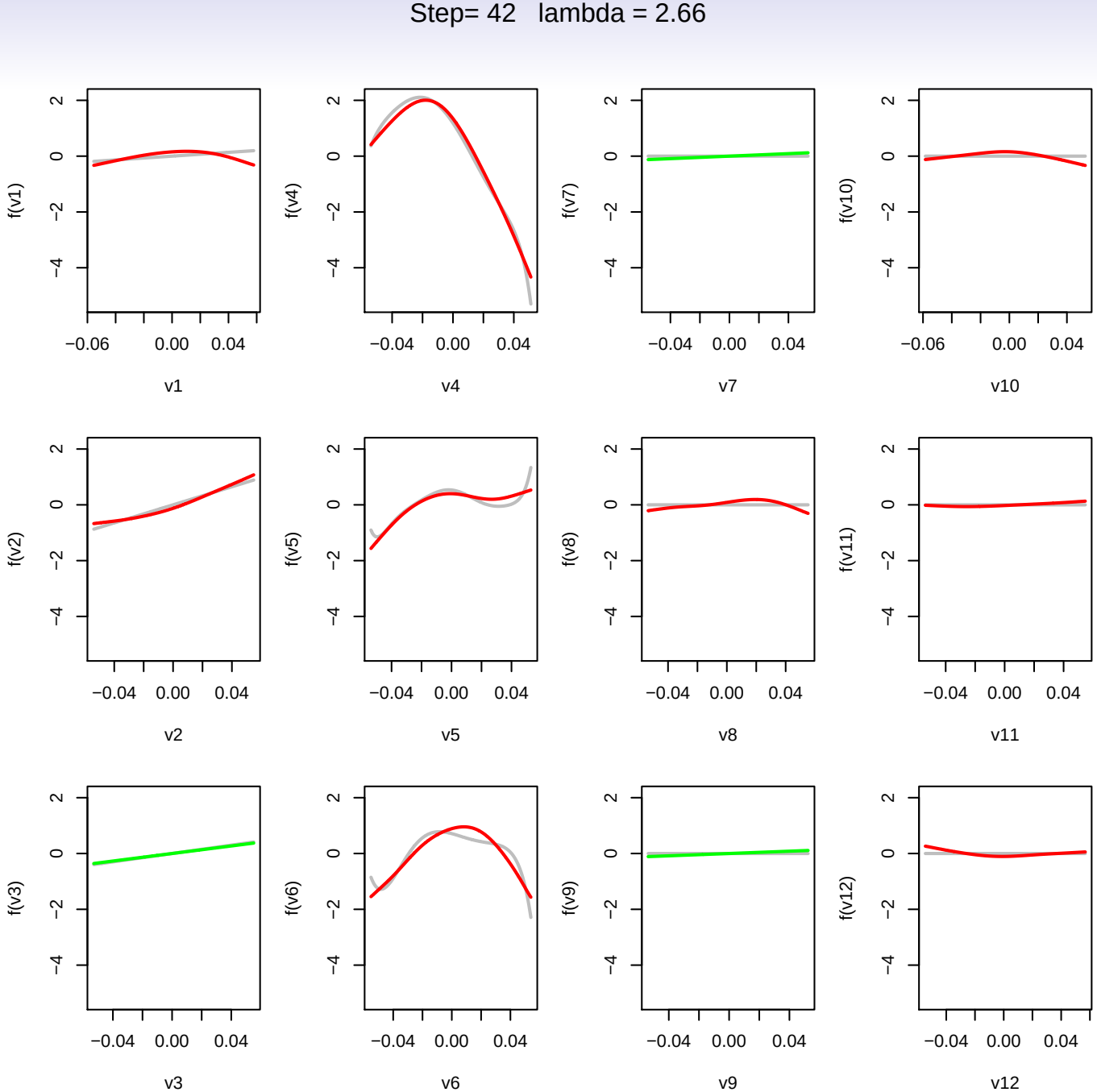

Step= 37 lambda = 4.26
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

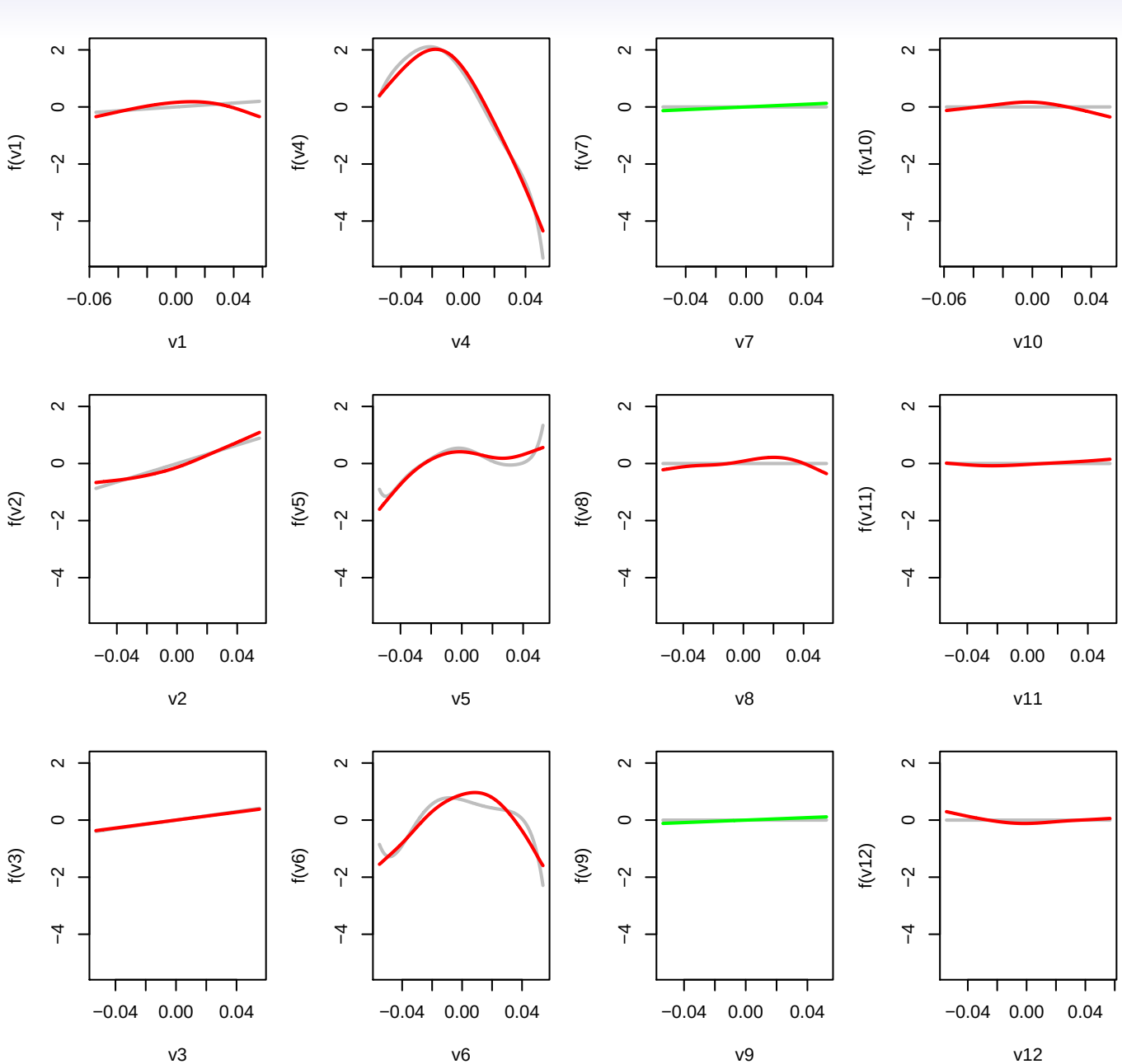

Step= 38 lambda = 3.87
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

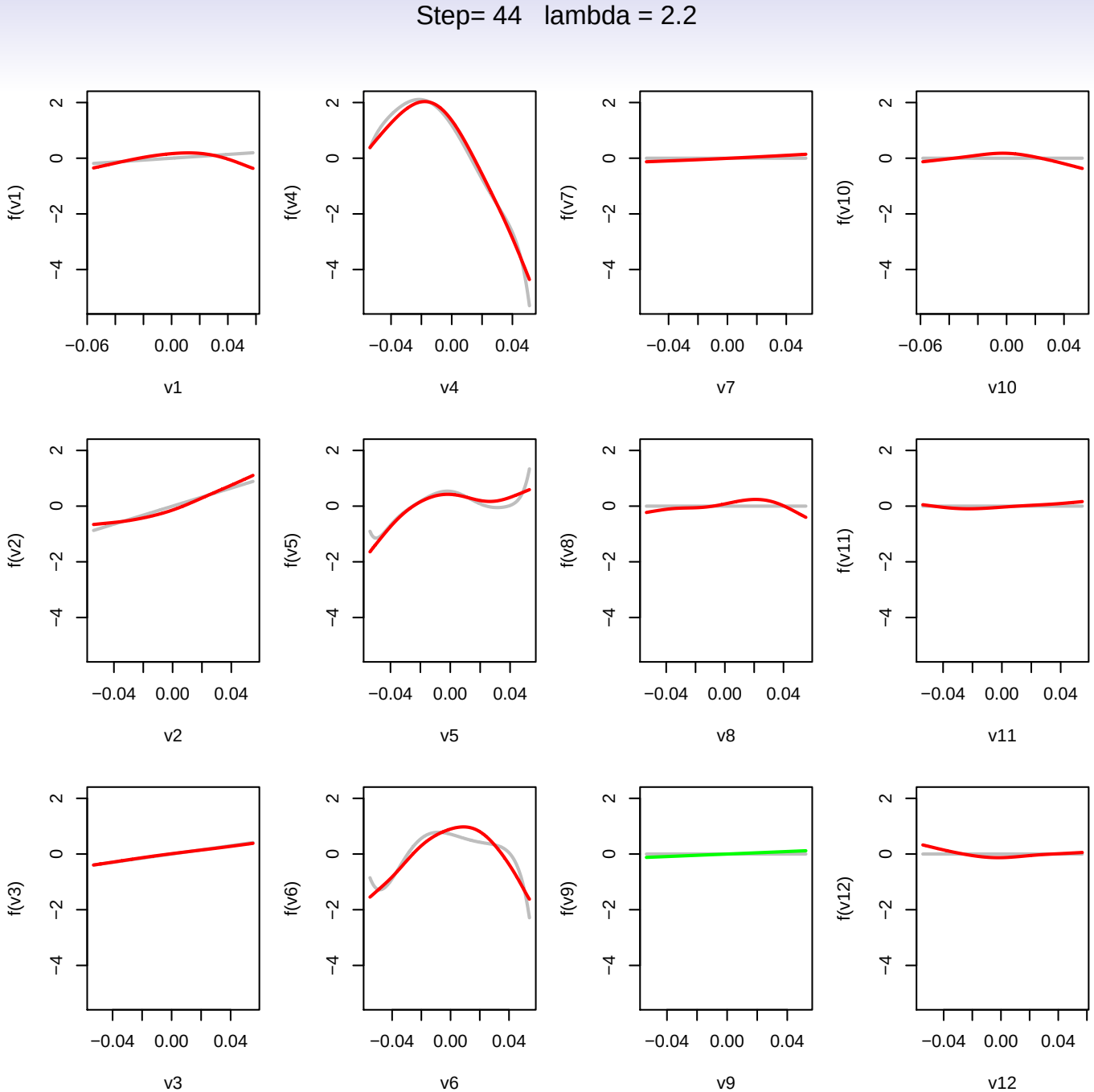

Step= 39 lambda = 3.53
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

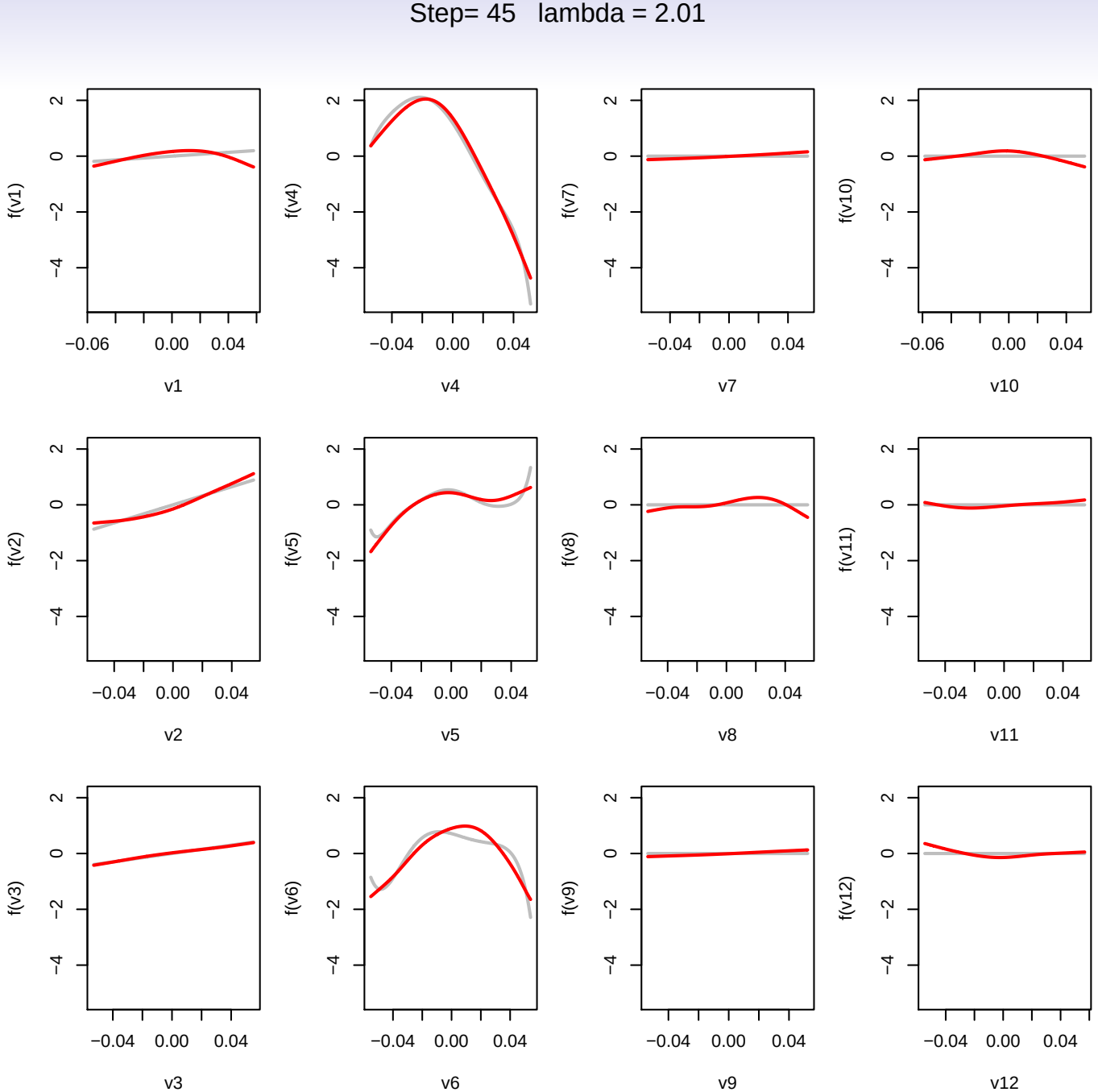

Step= 40 lambda = 3.21
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

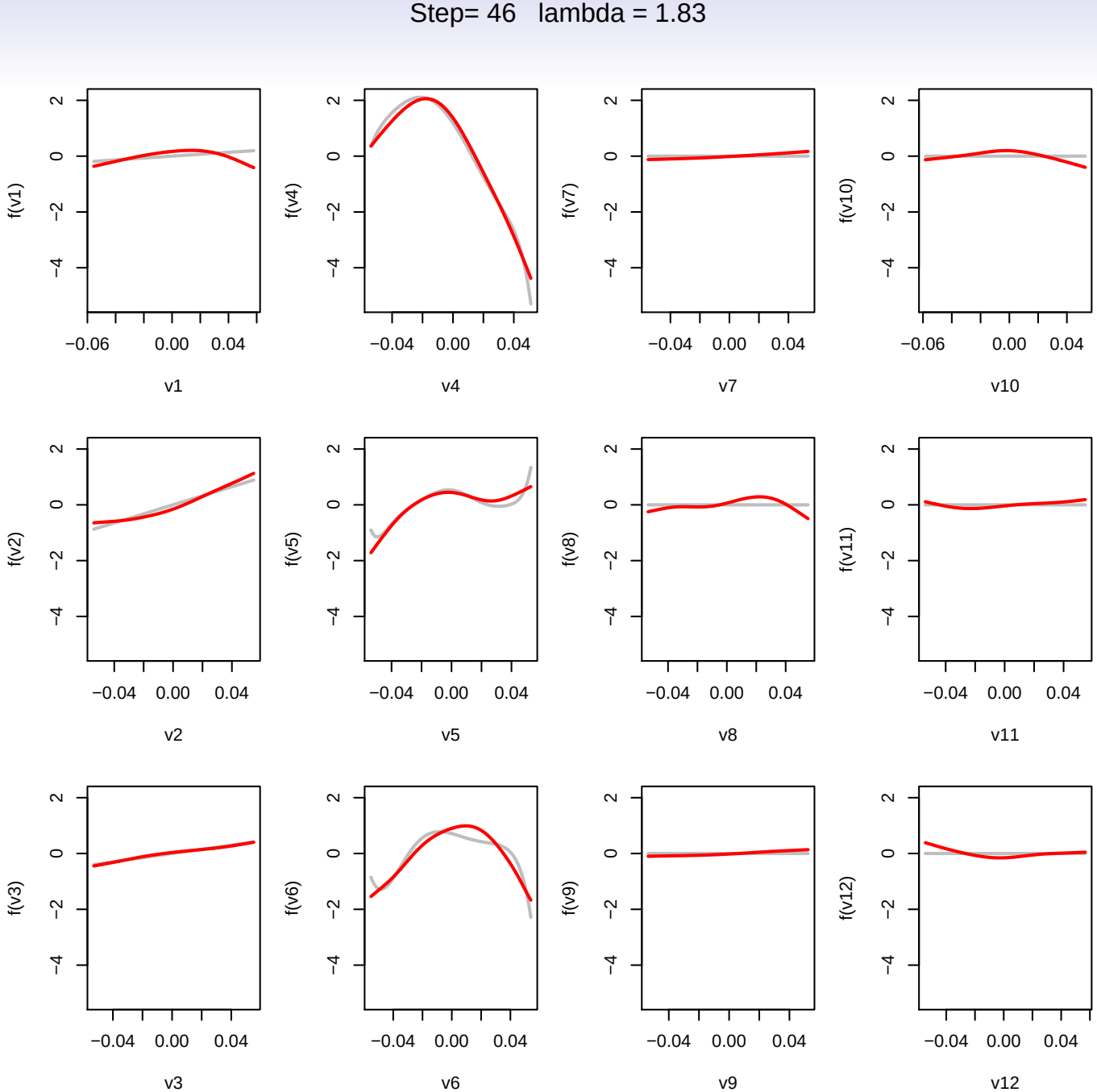

Step= 41 lambda = 2.92
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

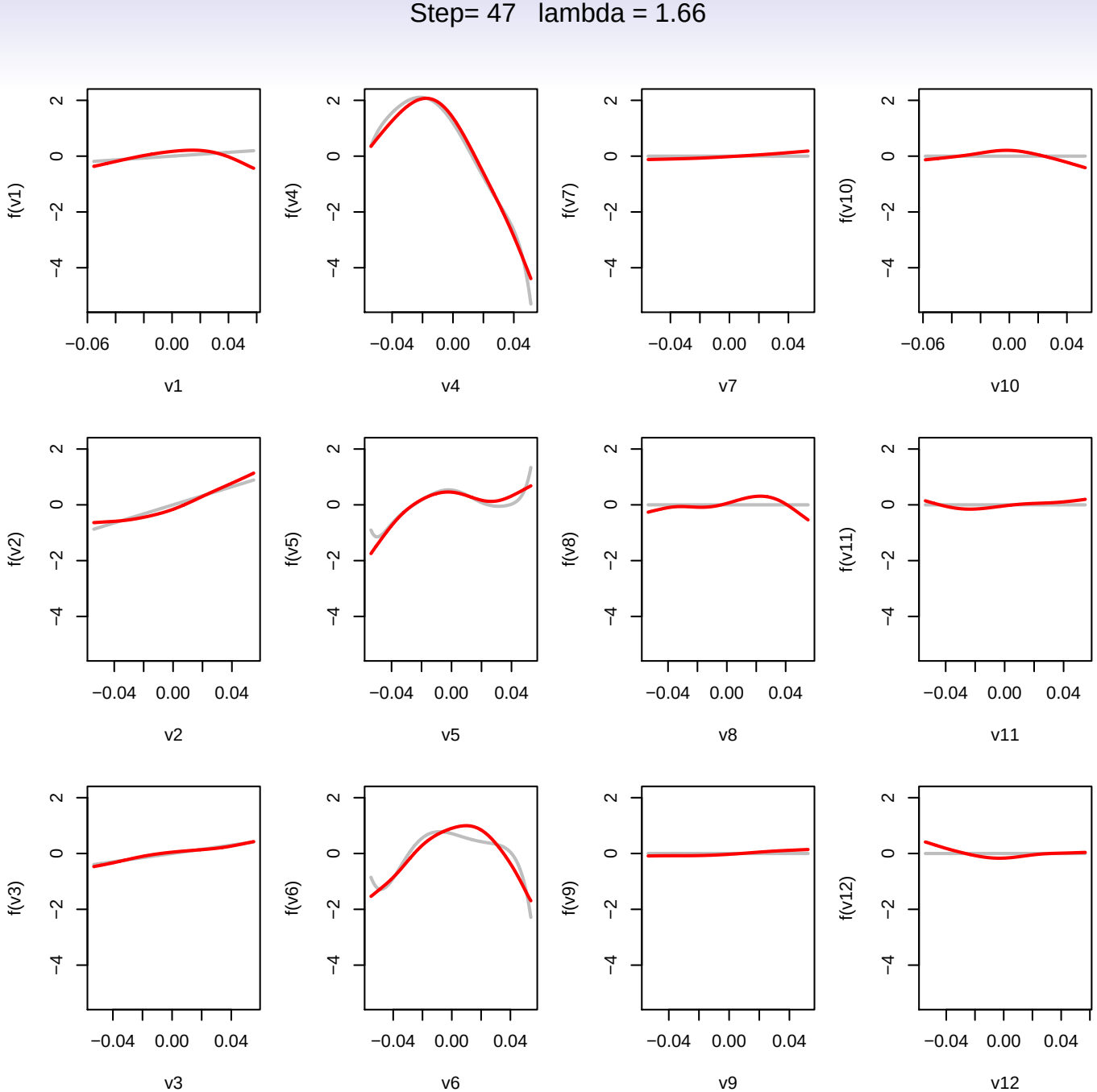

Step= 42 lambda = 2.66
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

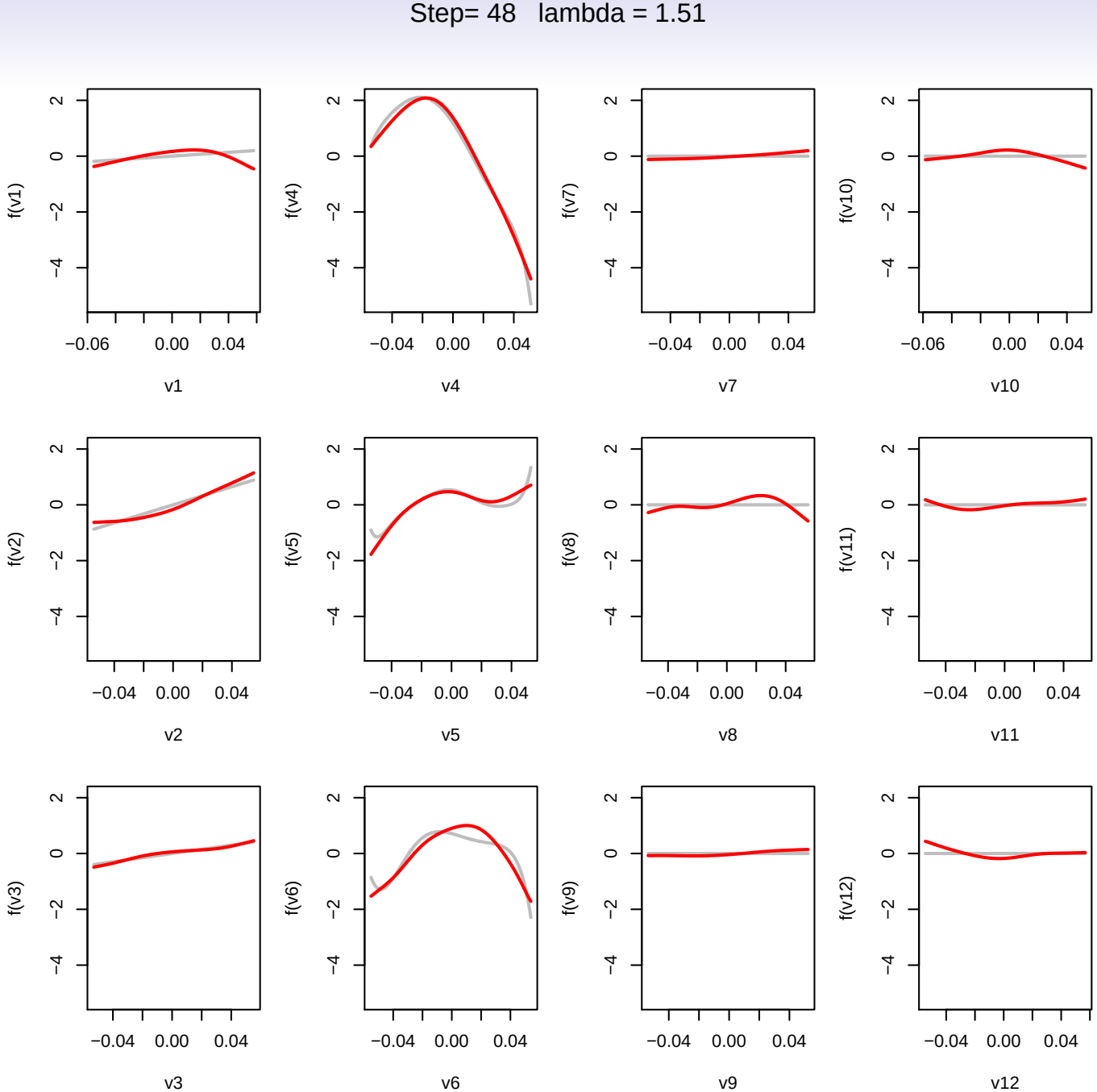

Step= 43 lambda = 2.42
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

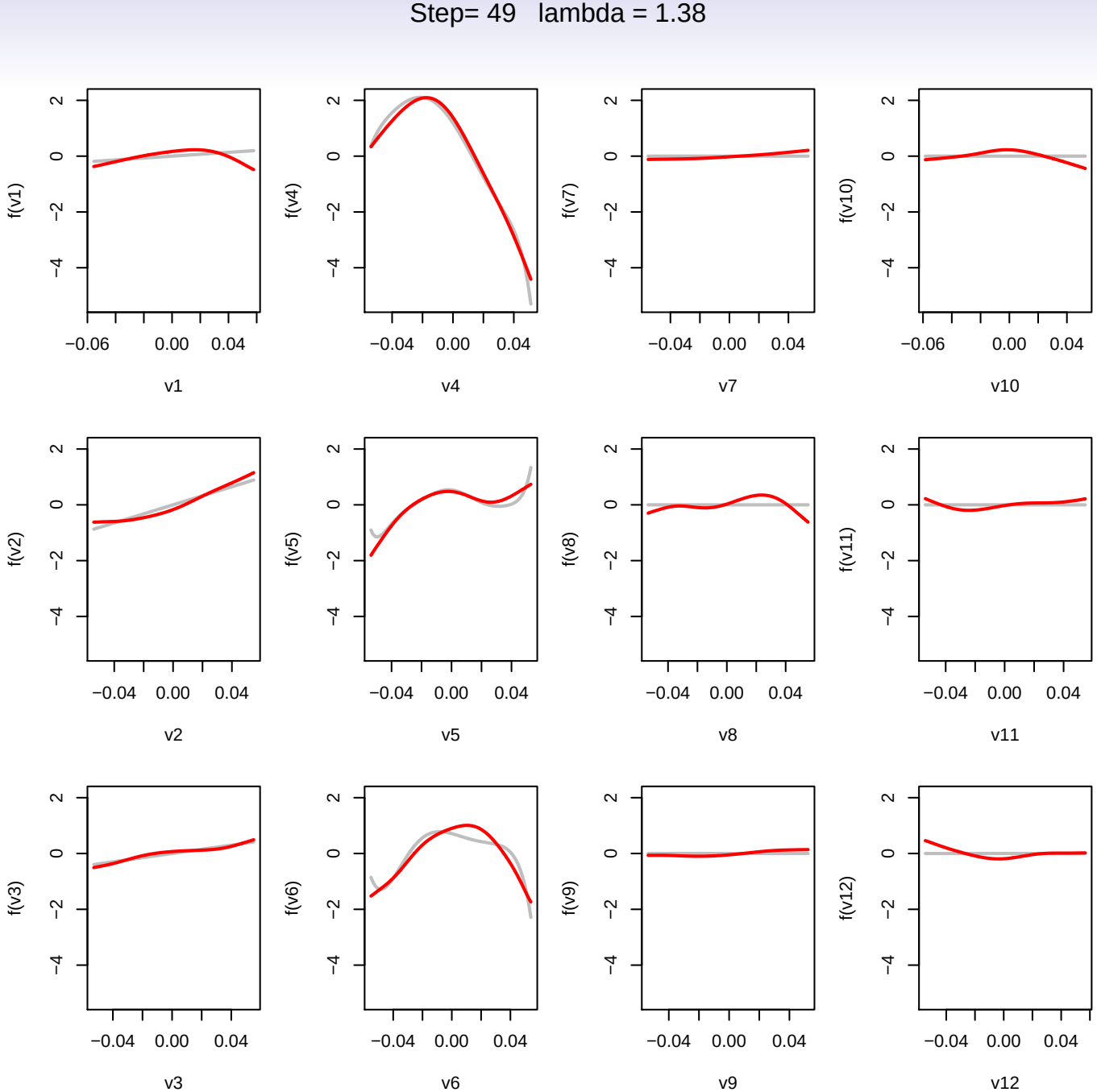

Step= 44 lambda = 2.2
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

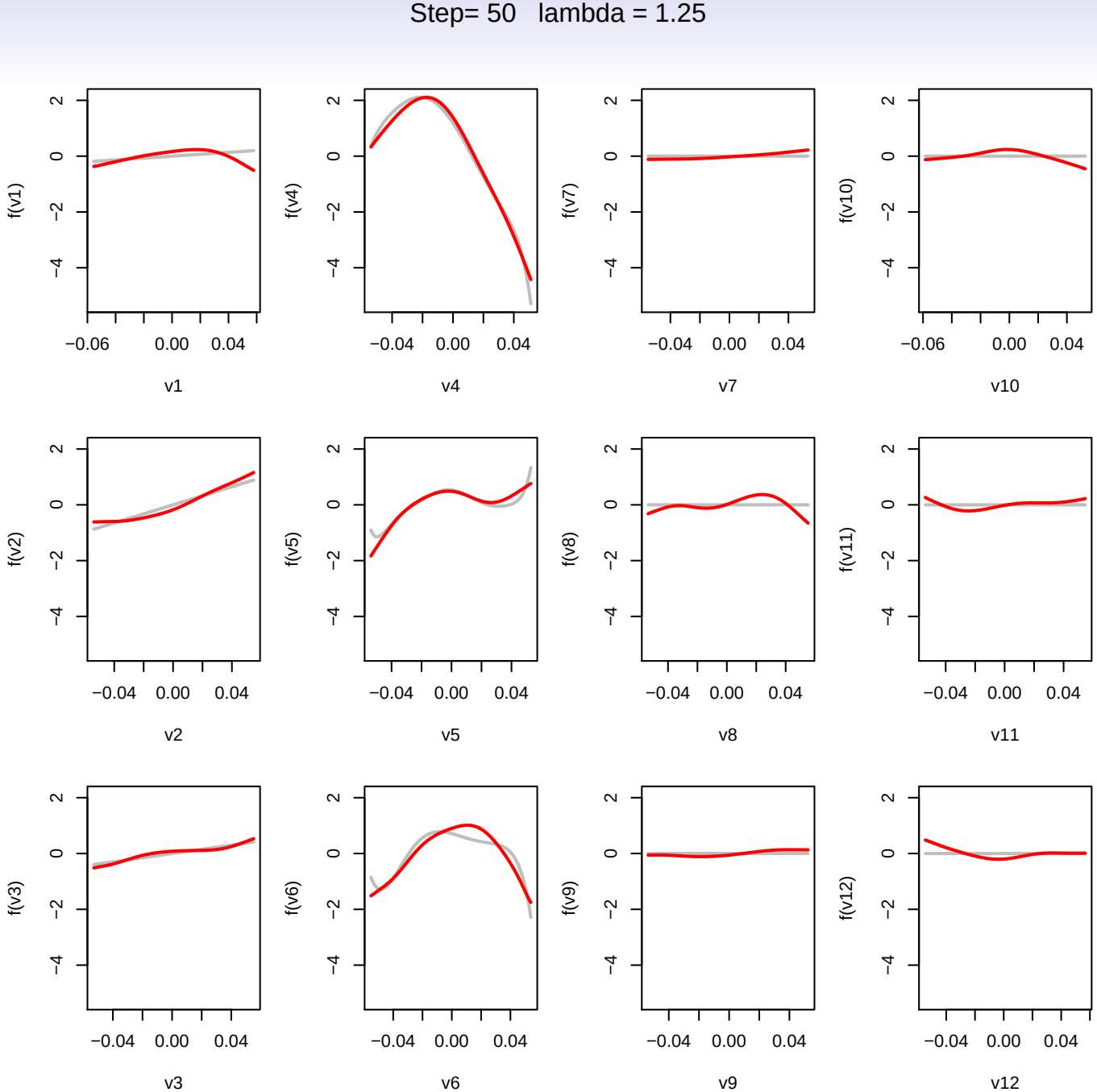

Step= 45 lambda = 2.01
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12


Step= 46 lambda = 1.83
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12


Step= 47 lambda = 1.66
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12


Step= 48 lambda = 1.51
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12


Step= 49 lambda = 1.38
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12


Step= 50 lambda = 1.25
f(v1)
v1
f(v4)
v4
f(v7)
v7
f(v10)
v10
f(v2)
v2
f(v5)
v5
f(v8)
v8
f(v11)
v11
f(v3)
v3
f(v6)
v6
f(v9)
v9
f(v12)
v12

Inference for Lasso?

- Can become Bayesian! Lasso penalty corresponds to Laplacian prior. However, need priors for everything, including λ (variance ratio).
- Bootstrap. Easier to do, with similar results.
- Covariance test for LARS sequence.
  "A Significance Test for the Lasso" — Richard Lockhart, Jonathan Taylor, Ryan Tibshirani and Rob Tibshirani (AoS, 2014)
- Conditional inference with Lasso.
  "Exact Post-Selection Inference with the Lasso" — Jason Lee, Dennis Sun, Yuekai Sun, Jonathan Taylor (2014, arXiv

Conditional Inference with Lasso

"Exact Post-Selection Inference with the Lasso" — Jason Lee, Dennis Sun, Yuekai Sun, Jonathan Taylor (2013) arXiv

$$y = \mu + \epsilon; \quad \epsilon \sim N(0, \sigma^2 I)$$

- Characterize a lasso solution set $\hat{E}$ via a set of linear innequalities $\{Ay \leq b\}$.
- Make inference on $\eta^T \mu$, where η can depend on $\hat{E}$; eg η extracts jth coefficient in projection of μ on $X_{\hat{E}}$.
- Characterize conditional distribution of $\eta^T y$ as one-dimensional truncated Gaussian (exact, non-asymptotic).

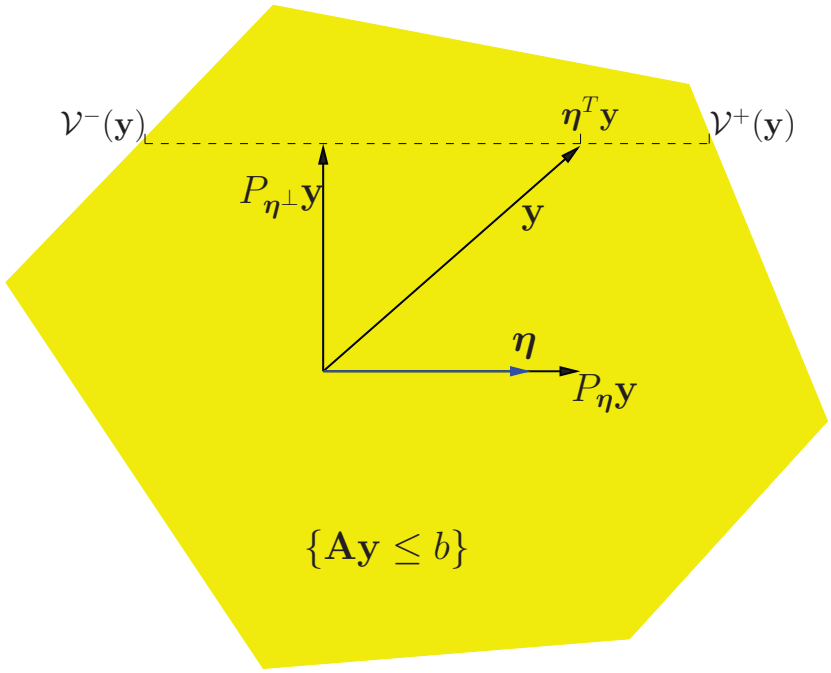


Figure 6.9. Schematic illustrating result (6.13), for the case $N = 2$ and $||\boldsymbol{\eta}||_2 = 1$ (due to Will Fithian). The yellow region is the selection event $\{\mathbf{A}\mathbf{y} \leq b\}$. We decompose $\mathbf{y}$ as the sum of two terms: its projection $P_{\boldsymbol{\eta}}\mathbf{y}$ onto $\boldsymbol{\eta}$ (with coordinate $\boldsymbol{\eta}^T\mathbf{y}$) and its projection onto the $(N-1)$-dimensional subspace orthogonal to $\boldsymbol{\eta}$: $\mathbf{y} = P_{\boldsymbol{\eta}}\mathbf{y} + P_{\boldsymbol{\eta}^\perp}\mathbf{y}$. Conditioning on $P_{\boldsymbol{\eta}^\perp}\mathbf{y}$, we see that the event $\{\mathbf{A}\mathbf{y} \leq b\}$ is equivalent to the event $\{\mathcal{V}^-(\mathbf{y}) \leq \boldsymbol{\eta}^T\mathbf{y} \leq \mathcal{V}^+(\mathbf{y})\}$. Furthermore $\mathcal{V}^+(\mathbf{y})$ and $\mathcal{V}^-(\mathbf{y})$ are independent of $\boldsymbol{\eta}^T\mathbf{y}$ since they are functions of $P_{\boldsymbol{\eta}^\perp}\mathbf{y}$ only, which is independent of $\mathbf{y}$.

Summary

- Coordinate descent effective, especially when models are sparse.
- Screening rules allow for massive computational savings.
- Group lasso and overlap group lasso allow for interesting specializations: Gam selection, mixed graphical model selection, interaction selection.
- Very exciting recent work of Jonathan Taylor group on post-selection inference.