Examples of Effect Displays with Partial Residuals Using Contrived Regression Data

John Fox and Sanford Weisberg

2017-11-22

The examples developed in this vignette are meant to supplement Fox and Weisberg [2018].

1 Basic Setup

We will analyze contrived data generated according to the following setup:

- We sample $n = 5000$ observations from a trivariate distribution for predictors x_1, x_2, and x_3, with uniform margins on the interval $[-2, 2]$, and with a prespecified bivariate correlation ρ between each pair of predictors. The method employed, described by Schumann [2009] and traceable to results reported by Pearson [1907], produces predictors that are nearly linearly related. Using 5000 observations allows us to focus on essentially asymptotic behavior of partial residuals in effect plots while still being able to discern individual points in the resulting graphs.
- We then generate the response y according to the model

$$y = \beta_0 + h\left(\boldsymbol{\beta}, \{x_1, x_2, x_3\}\right) + \varepsilon \tag{1}$$

  where $\varepsilon \sim \mathrm{N}(0, 1.5^2)$. The regression function $h(\cdot)$ varies from example to example.

The following functions make it convenient to generate data according to this setup. These functions are more general than is strictly necessary so as to encourage further experimentation.

```
mvrunif <- function(n, R, min = 0, max = 1){
    # method (but not code) from E. Schumann,
    # "Generating Correlated Uniform Variates"
    # URL:
    # <http://comisef.wikidot.com/tutorial:correlateduniformvariates>
    # downloaded 2015-05-21
    if (!is.matrix(R) || nrow(R) != ncol(R) ||
    max(abs(R - t(R))) > sqrt(.Machine$double.eps))
    stop("R must be a square symmetric matrix")
    if (any(eigen(R, only.values = TRUE)$values <= 0))
    stop("R must be positive-definite")
    if (any(abs(R) - 1 > sqrt(.Machine$double.eps)))
    stop("R must be a correlation matrix")
    m <- nrow(R)
    R <- 2 * sin(pi * R / 6)
    X <- matrix(rnorm(n * m), n, m)
    X <- X %*% chol(R)
    X <- pnorm(X)
    min + X * (max - min)
}
```

```
gendata <- function(n = 5000, R, min = -2, max = 2, s = 1.5,
    model = expression(x1 + x2 + x3)){
    data <- mvrunif(n = n, min = min, max = max, R = R)
    colnames(data) <- c("x1", "x2", "x3")
    data <- as.data.frame(data)
    data$error <- s * rnorm(n)
    data$y <- with(data, eval(model) + error)
    data
}

R <- function(offdiag = 0, m = 3){
    R <- diag(1, m)
    R[lower.tri(R)] <- R[upper.tri(R)] <- offdiag
    R
}
```

2 Unmodelled Interaction

We begin with uncorrelated predictors and the true regression mean function $\mathrm{E}(y|\mathbf{x}) = x_1 + x_2x_3$, but fit the incorrect additive working model $y \sim x_1 + x_2 + x_3$ to the data.

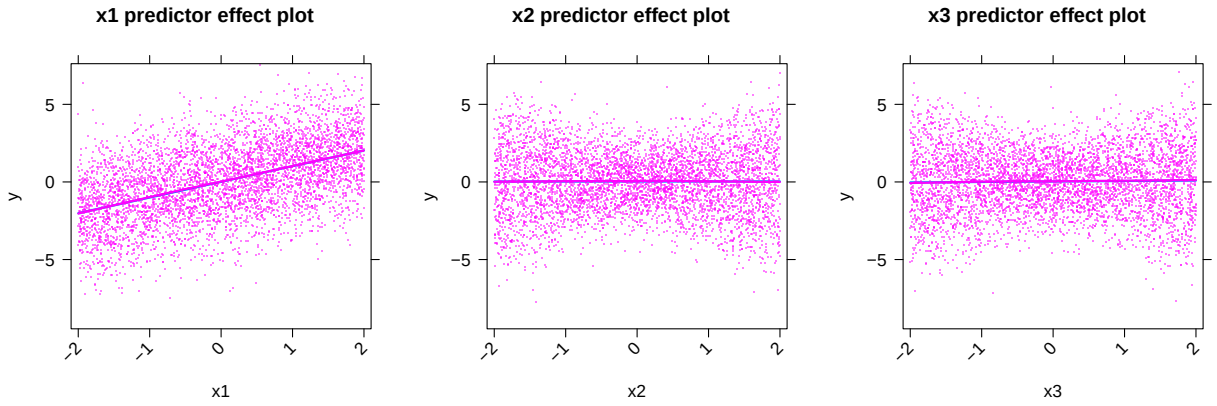
```
set.seed(682626)
Data.1 <- gendata(R = R(0), model = expression(x1 + x2 * x3))
round(cor(Data.1), 2)
```

```
##          x1    x2    x3 error     y
## x1     1.00 -0.03  0.03  0.01  0.49
## x2    -0.03  1.00 -0.01  0.00 -0.01
## x3     0.03 -0.01  1.00  0.02  0.03
## error  0.01  0.00  0.02  1.00  0.66
## y      0.49 -0.01  0.03  0.66  1.00
```

```
summary(mod.1 <- lm(y ~ x1 + x2 + x3, data = Data.1))
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3, data = Data.1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7493 -1.3702  0.0438  1.3873  8.3059
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.005536   0.028923   0.191    0.848
## x1          1.010175   0.025299  39.929   <2e-16
## x2          0.001697   0.024929   0.068    0.946
## x3          0.031178   0.024717   1.261    0.207
##
## Residual standard error: 2.045 on 4996 degrees of freedom
## Multiple R-squared:  0.2426,Adjusted R-squared:  0.2422
## F-statistic: 533.5 on 3 and 4996 DF,  p-value: < 2.2e-16
```

Figure 1: Effect displays with partial residuals for the individual predictors x_1, x_2, and x_3 in the incorrect model $y \sim x_1 + x_2 + x_3$ fit to data generated with the mean function $\mathrm{E}(y|\mathbf{x}) = x_1 + x_2x_3$, with uncorrelated predictors.



For reproducibility, we set a known seed for the pseudo-random number generator; this seed was itself generated pseudo-randomly, and we reuse it in the examples reported below. As well, in this first example, but not for those below, we show the correlation matrix of the randomly generated data along with the fit of the working model to the data.

Effect plots with partial residuals corresponding to the terms in the working model are shown in Figure 1:

```
library(effects)
plot(predictorEffects(mod.1, partial.residuals=TRUE),
     partial.residual=list(pch=".", col="#FF00FF80"),
     axes=list(x=list(rotate=45)),
     rows=1, cols=3)
```

In these graphs and, unless noted to the contrary, elsewhere in this vignette, the loess smooths are drawn with span 2/3. Because of the large number of points in the graphs, optional arguments to `plot` are specified to de-emphasize the partial residuals. To this end, the residuals are plotted as small points (`pch="."`) and in a translucent magenta color (`col="#FF00FF80"`).

The failure of the model is not apparent in these traditional partial residual plots, but it is clear in the term effect plot for $\{x_2, x_3\}$, corresponding to the unmodelled interaction x_2:x_3, and shown in the top panel of Figure 2, generated using

```
plot(Effect(c("x2", "x3"), mod.1, partial.residuals = TRUE),
     partial.residual=list(pch=".", col="#FF00FF80"),
     axes=list(x=list(rotate=45)),
     lattice=list(layout=c(4, 1)))
```

Moreover, the effect plot in the bottom panel of the figure for $\{x_1, x_2\}$, corresponding to a term *not* in the true mean function, correctly indicates lack of interaction between these two predictors:

```
plot(Effect(c("x1", "x2"), mod.1, partial.residuals = TRUE),
    partial.residual=list(pch=".", col="#FF00FF80"),
    axes=list(x=list(rotate=45)),
    lattice=list(layout=c(4, 1)))
```

As a partly contrasting example, we turn to a similar data set, generated with the same regression mean function but with moderately correlated predictors, where the pairwise predictor correlations are $\rho = 0.5$:

Figure 2: Term effect displays with partial residuals for $\{x_2, x_3\}$, corresponding to the missing interaction x_2:x_3, and for $\{x_1, x_2\}$, corresponding to an interaction not present in the model that generated the data.

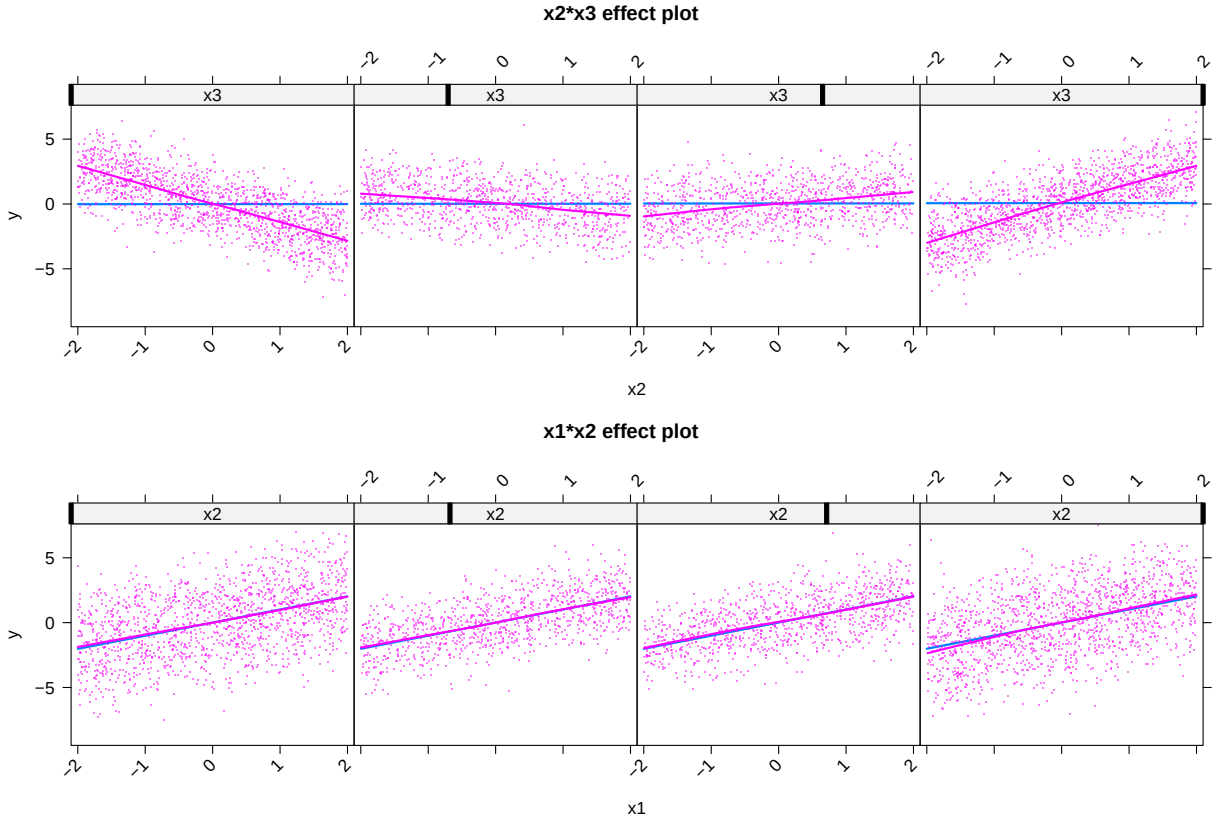
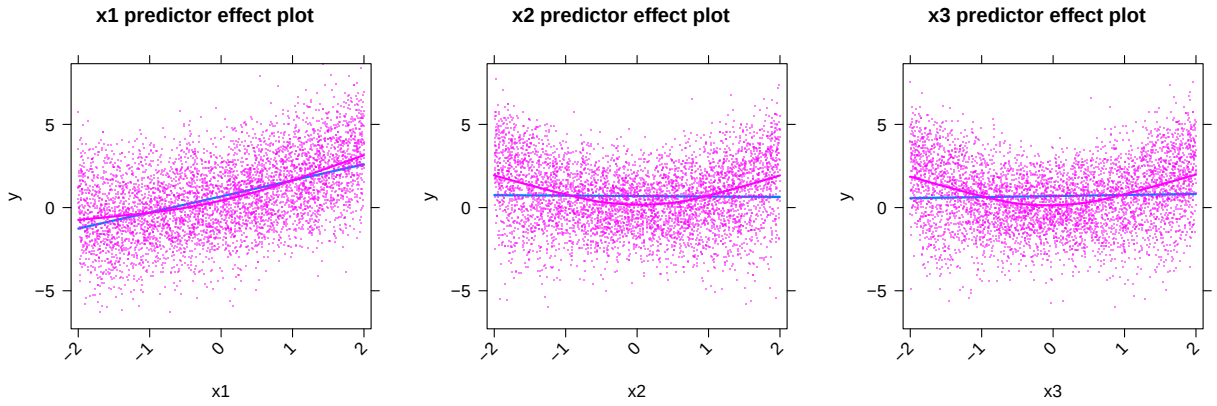


Figure 3: Predictor effect displays with partial residuals for the individual predictors x_1, x_2, and x_3 in the incorrect model $y \sim x_1 + x_2 + x_3$ fit to data generated with the mean function $\mathrm{E}(y|\mathbf{x}) = x_1 + x_2x_3$, with moderately correlated predictors.




```
set.seed(682626)
Data.2 <- gendata(R = R(0.5), model = expression(x1 + x2 * x3))
mod.2 <- lm(y ~ x1 + x2 + x3, data = Data.2)
```

Graphs analogous to those from the preceding example appear in Figures 3 and 4:

```
plot(predictorEffects(mod.2, partial.residuals=TRUE),
     partial.residual=list(pch=".", col="#FF00FF80",fig.show='hide'),
     axes=list(x=list(rotate=45)),
     rows=1, cols=3)
```

```
plot(Effect(c("x2", "x3"), mod.2, partial.residuals = TRUE),
     partial.residual=list(pch=".", col="#FF00FF80"),
     axes=list(x=list(rotate=45)),
     lattice=list(layout=c(4, 1)))
```

```
plot(Effect(c("x1", "x2"), mod.2, partial.residuals = TRUE),
    partial.residual=list(pch=".", col="#FF00FF80",fig.show='hide'),
    axes=list(x=list(rotate=45)),
    lattice=list(layout=c(4, 1)))
```

The predictor effect plots for x_2 and x_3, and to a much lesser extent, for x_1, in the incorrect model in Figure 3 show apparent nonlinearity as a consequence of the unmodelled interaction and the correlations among the predictors. A similar phenomenon was noted in our analysis of the Canadian occupational prestige data in Fox and Weisberg [2018, Section 4.2], where the unmodelled interaction between `type` and `income` induced nonlinearity in the partial relationship of `prestige` to `income`. The omitted interaction is clear in the effect plot for $\{x_2, x_3\}$, but also, to a lesser extent, contaminates the effect plot for $\{x_1, x_2\}$, which corresponds to an interaction that does not enter the model generating the data. These artifacts become more prominent if we increase the predictor correlations, say to $\rho = 0.9$ (as we invite the reader to do).

3 Unmodelled Nonlinearity

We generate data as before, but from the true model $\mathrm{E}(y|\mathbf{x}) = x_1^2 + x_2 + x_3$, where the predictors are moderately correlated, with pairwise correlations $\rho = 0.5$, but fit the incorrect additive working model $y \sim x_1 + x_2 + x_3$ to the data:

Figure 4: Term effect displays with partial residuals for $\{x_2, x_3\}$, corresponding to the missing interaction $x_2{:}x_3$, and for $\{x_1, x_2\}$, corresponding to an interaction not present in the model that generated the data.

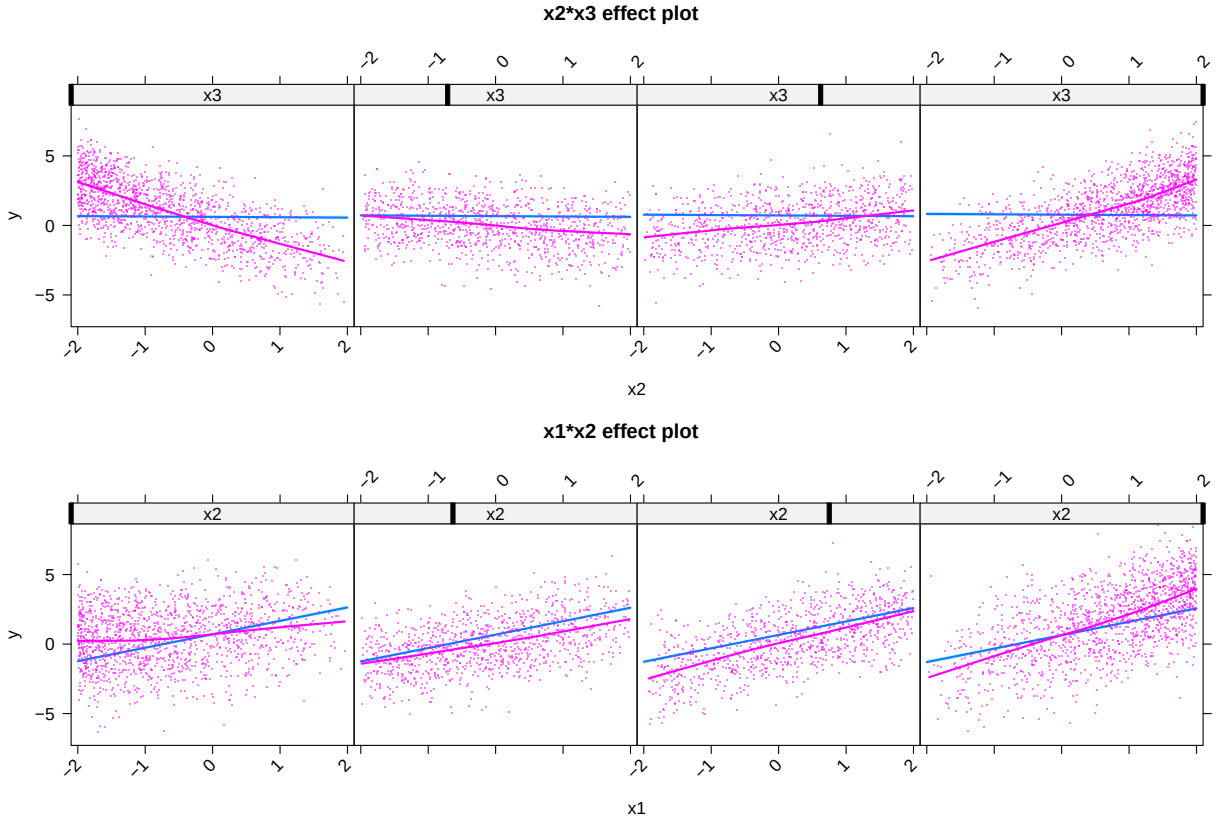
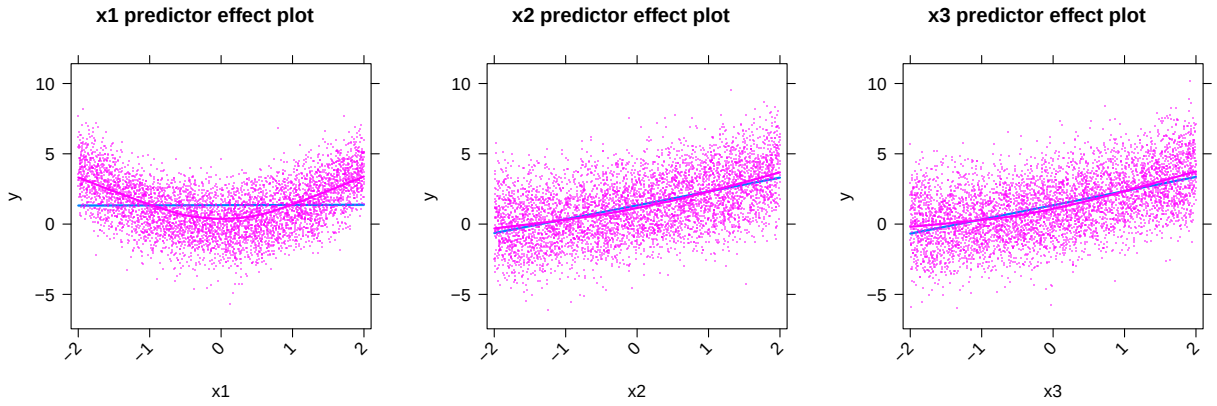


Figure 5: Predictor effect displays with partial residuals for the individual predictors x_1, x_2, and x_3 in the incorrect model $y \sim x_1 + x_2 + x_3$ fit to data generated with the mean function $\mathrm{E}(y|\mathbf{x}) = x_1^2 + x_2 + x_3$, with moderately correlated predictors.




```
set.seed(682626)
Data.3 <- gendata(R = R(0.5), model = expression(x1^2 + x2 + x3))
mod.3 <- lm(y ~ x1 + x2 + x3, data = Data.3)
```

Effect plots with residuals for the predictors in the working model appear in Figure 5. The unmodelled nonlinearity in the partial relationship of y to x_1 is clear, but there is some contamination of the plots for x_2 and x_3. The contamination is much more dramatic if the correlations among the predictors are increased to, say, $\rho = 0.9$ (as the reader may verify).

```
plot(predictorEffects(mod.3, partial.residuals=TRUE),
     partial.residual=list(pch=".", col="#FF00FF80"),
     axes=list(x=list(rotate=45)),
     rows=1, cols=3)
```

Effect plots for $\{x_1, x_2\}$ and $\{x_2, x_3\}$ are shown in Figure 6:

```
plot(Effect(c("x2", "x3"), mod.3, partial.residuals = TRUE),
     partial.residual=list(pch=".", col="#FF00FF80"),
     axes=list(x=list(rotate=45)),
     lattice=list(layout=c(4, 1)))
```

```
plot(Effect(c("x1", "x2"), mod.3, partial.residuals = TRUE),
    partial.residual=list(pch=".", col="#FF00FF80"),
    axes=list(x=list(rotate=45)),
    lattice=list(layout=c(4, 1)))
```

Neither of these graphs corresponds to a term in the model generating the data nor in the working model, and the effect plots largely confirm the absence of x_1:x_2 and x_2:x_3 interactions, along with the nonlinearity of the partial effect of x_1, apparent in the top panel.

4 Simultaneous Unmodelled Nonlinearity and Interaction

This last example also appears in Fox and Weisberg [2018, Section 4.3]. We consider a true model that combines nonlinearity and interaction, $\mathrm{E}(y|\mathbf{x}) = x_1^2 + x_2 x_3$; the predictors are moderately correlated, with

Figure 6: Term effect displays with partial residuals for $\{x_1, x_2\}$ and for $\{x_2, x_3\}$, neither of which corresponds to an interaction in the model generating the data.

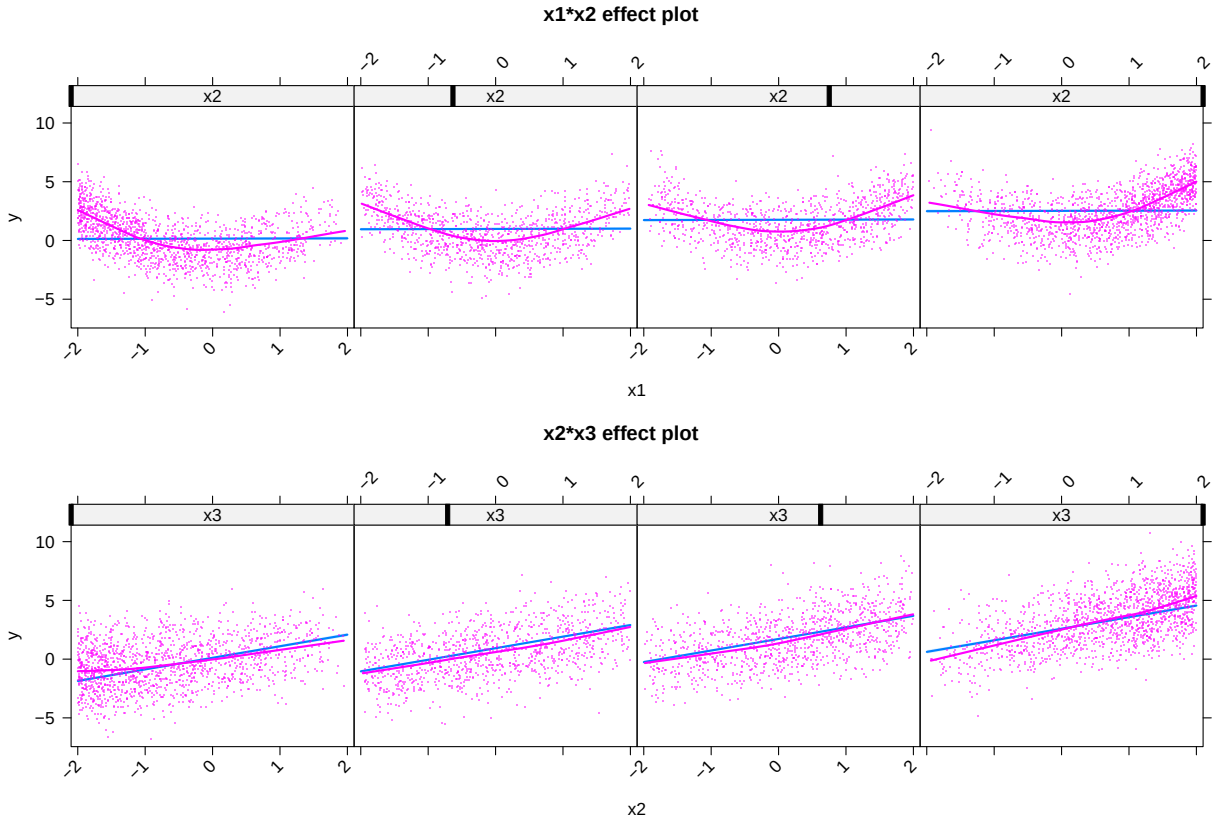
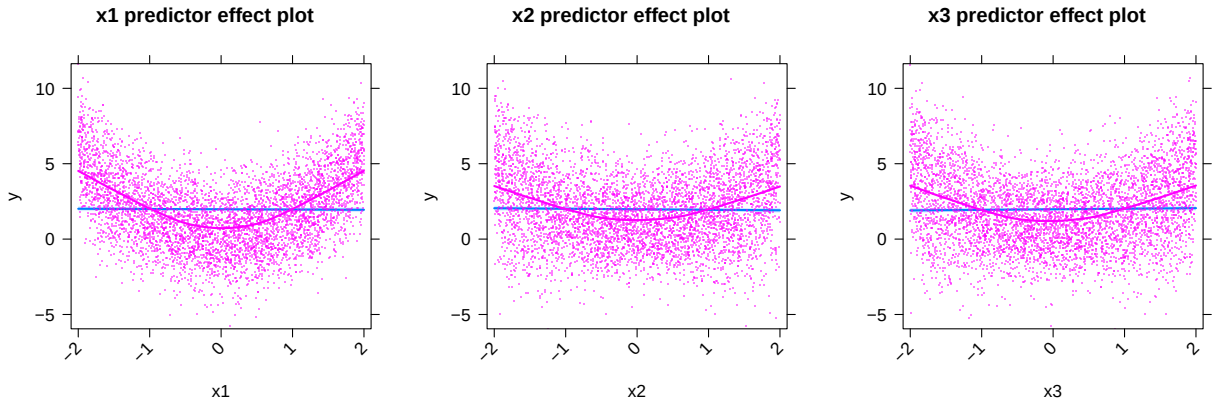


Figure 7: Effect displays with partial residuals for the predictors x_1, x_2, and x_3 in the incorrect model $y \sim x_1 + x_2 + x_3$ fit to data generated with the mean function $\mathrm{E}(y|\mathbf{x}) = x_1^2 + x_2 x_3$, with moderately correlated predictors.



$\rho = 0.5$. We then fit the incorrect working model $y \sim x_1 + x_2 + x_3$ to the data, producing the predictor effect displays with partial residuals in Figure 7, for the predictors x_1, x_2, and x_3, which appear additively in the working model, and the term effect displays in Figure 8 for $\{x_2, x_3\}$ and $\{x_1, x_2\}$, corresponding respectively to the incorrectly excluded x_2:x_3 term and the correctly excluded x_1:x_2 interaction.

```
set.seed(682626)
Data.4 <- gendata(R = R(0.5), model = expression(x1^2 + x2 * x3))
mod.4 <- lm(y ~ x1 + x2 + x3, data = Data.4)
```

```
plot(predictorEffects(mod.4, partial.residuals=TRUE),
     partial.residual=list(pch=".", col="#FF00FF80"),
     axes=list(x=list(rotate=45)),
     rows=1, cols=3)
```

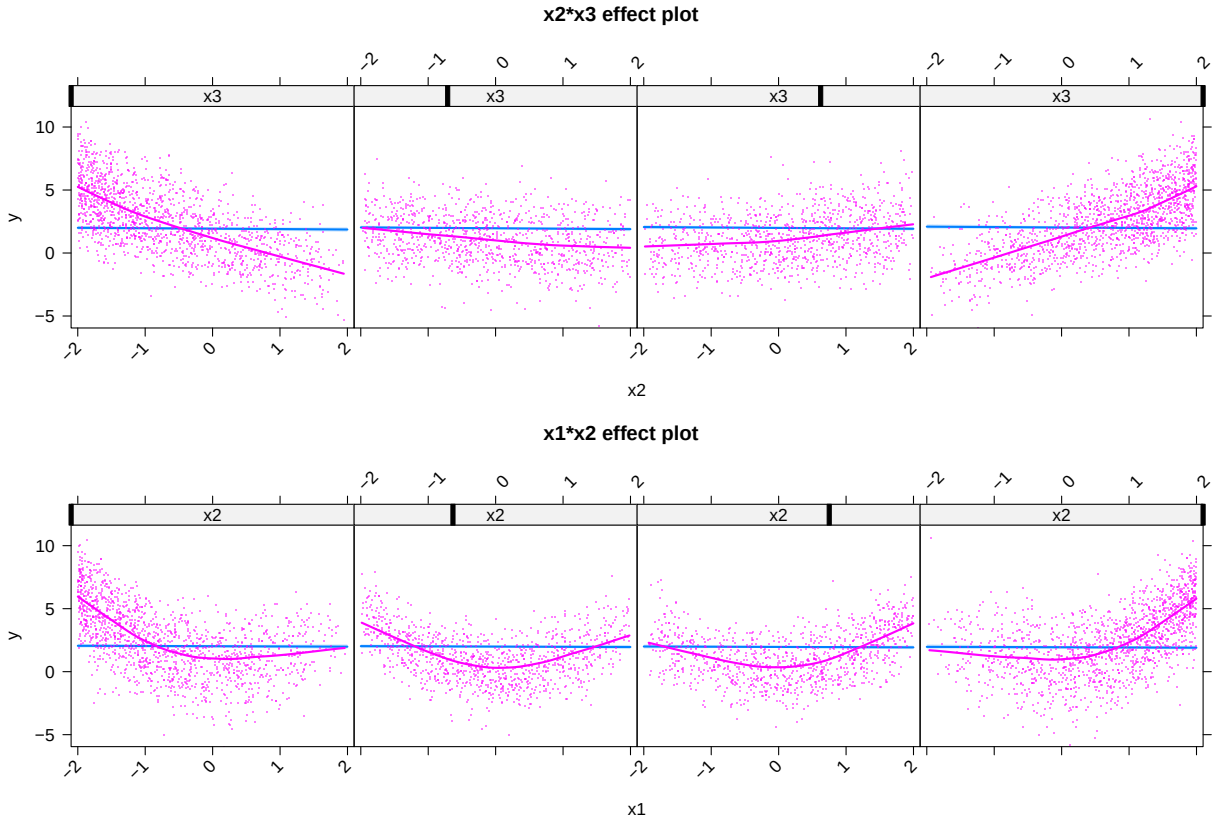
```
plot(Effect(c("x2", "x3"), mod.4, partial.residuals = TRUE),
     partial.residual=list(pch=".", col="#FF00FF80"),
      axes=list(x=list(rotate=45)),
     lattice=list(layout=c(4, 1)))
```

```
plot(Effect(c("x1", "x2"), mod.4, partial.residuals = TRUE),
    partial.residual=list(pch=".", col="#FF00FF80"),
    axes=list(x=list(rotate=45)),
    lattice=list(layout=c(4, 1)))
```

The nonlinearity in the partial relationship of y to x_1 shows up clearly. The nonlinearity apparent in the plots for x_2 and x_3 is partly due to contamination with x_1, but largely to the unmodelled interaction between x_2 and x_3, coupled with the correlation between these predictors. The plot corresponding to the missing x_2:x_3 term (in the top panel of Figure 8) does a good job of detecting the unmodelled interaction, and curvature in this plot is slight. The plot for the x_1:x_2 term (in the bottom panel of Figure 8), a term neither in the true model nor in the working model, primarily reveals the unmodelled nonlinearity in the partial relationship of y to x_1.

If we fit the correct model, $y \sim x_1^2 + x_2 * x_3$, to the data, we obtain the plots shown in Figure 9. As theory suggests, the partial residuals in these effect displays validate the model, supporting the exclusion of the

Figure 8: Term effect displays with partial residuals for $\{x_2, x_3\}$ (top) and for $\{x_1, x_2\}$ (bottom), the first of which corresponds to the missing x_2:x_3 interaction in the model generating the data.



x_1:x_2 interaction, the linear-by-linear interaction between x_1 and x_2, and the quadratic partial relationship of y to x_1.

```
mod.5 <- lm(y ~ poly(x1, 2) + x2*x3, data=Data.4)
plot(Effect("x1", mod.5, partial.residuals=TRUE),
     partial.residual=list(pch=".", col="#FF00FF80", span=0.2))
```

```
plot(Effect(c("x2", "x3"), mod.5, partial.residuals = TRUE),
     partial.residual=list(pch=".", col="#FF00FF80"),
     axes=list(x=list(rotate=45)),
     lattice=list(layout=c(4, 1)), span=0.5)
```

```
plot(Effect(c("x1", "x2"), mod.5, partial.residuals = TRUE),
    partial.residual=list(pch=".", col="#FF00FF80", span=0.35),
    axes=list(x=list(rotate=45)),
    lattice=list(layout=c(4, 1)))
```

In these graphs, we adjust the span of the loess smoother to the approximately smallest value that produces a smooth fit to the partial residuals in each case.

References

J. Fox and S. Weisberg. Visualizing fit and lack of fit in complex regression models with predictor effect plots and partial residuals. *Journal of Statistical Software*, 87(9):1–27, 2018. doi: 10.18637/jss.v087.i09.

K. Pearson. *Mathematical Contributions to the Theory of Evolution.—XVI. On Further Methods of Determining Correlation.* Drapers' Company Research Memoirs. Biometric Series. IV. Cambridge University Press, London, 1907.

E. Schumann. *Generating Correlated Uniform Variates*, 2009. `http://comisef.wikidot.com/ tutorial:correlateduniformvariates` [Accessed: 2015-05-21].

Figure 9: Effect displays with partial residuals for x_1 and $\{x_2, x_3\}$, which correspond to terms in the model generating *and* fitted to the data, $y \sim x_1^2 + x_2 * x_3$, and for $\{x_1, x_2\}$, which corresponds to an interaction that is not in the model.