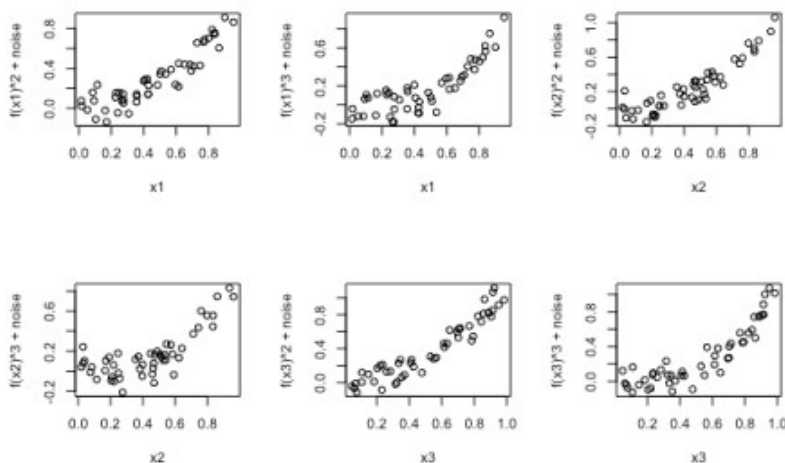


To plot multiple figures on a single canvas in base R, we can change the graphical parameter `mfrow`. For instance, the code below tells R that subsequent figures will be drawn in a 2-by-3 array:

```
par(mfrow = c(2, 3))
```

If we then run this next block of code, we will get the image below:

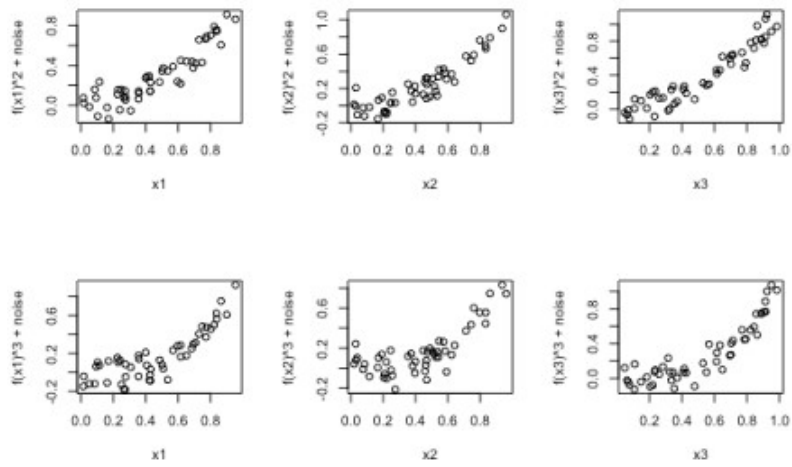
```
set.seed(10)
n <- 50; p <- 3
x <- matrix(runif(n * p), ncol = p)
for (j in 1:p) {
  plot(x[, j], x[, j]^2 + 0.1 * rnorm(n),
       xlab = paste0("x", j), ylab = paste0("f(x", j, ")^2 + noise"))
  plot(x[, j], x[, j]^3 + 0.1 * rnorm(n),
       xlab = paste0("x", j), ylab = paste0("f(x", j, ")^3 + noise"))
}
```



Notice how the plots are filled in by row? That is, the first plot goes in the top-left corner, the next plot goes to its right, and so on. **What if we want the plots to be filled in by column instead?** For instance, in our example above each feature is associated with two panels. It would make more sense for the first column to be filled with plots related to x_1 , and so on.

There is an easy fix for that: **instead of specifying the `mfrow` parameter, specify the `mfcol` parameter.** See the results below:

```
par(mfcol = c(2, 3))
set.seed(10)
n <- 50; p <- 3
x <- matrix(runif(n * p), ncol = p)
for (j in 1:p) {
  plot(x[, j], x[, j]^2 + 0.1 * rnorm(n),
       xlab = paste0("x", j), ylab = paste0("f(x", j, ")^2 + noise"))
  plot(x[, j], x[, j]^3 + 0.1 * rnorm(n),
       xlab = paste0("x", j), ylab = paste0("f(x", j, ")^3 + noise"))
}
```



References:

1. [Graphical layouts.](#)
-