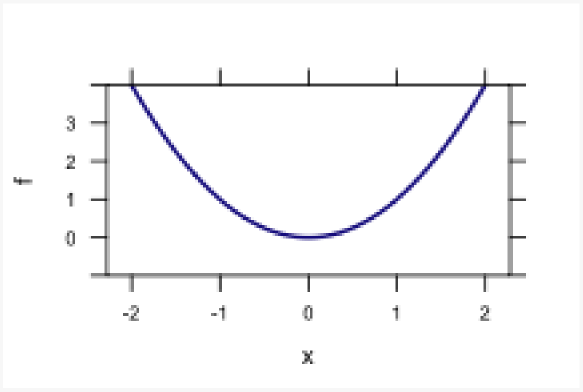
```
f <- function(x) x^2
fplot(f, ylim = c(-1, 4))
```



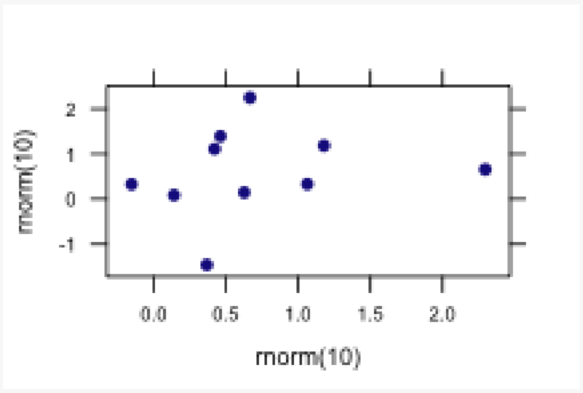
$f(x) = x(1-x)^2$ becomes

```
f <- function(x) {
    x * (1 - x)^2
}
```

```
f(1)

[1] 0

f(0:5)

[1]  0  0  2 12 36 80
```

```
fplot.default

function (x, xlim, ylim, n = 200, args = list(), type = "l",
    xlab, ylab, ...)
{
    if (is.character(x)) {
        temp_ylab = x
    }
    else {
        temp_ylab = deparse(substitute(x))
    }
    makeFunction <- function(f) {
        if (is.function(f)) {
            return(f)
        }
        if (is.numeric(f)) {
            return(function(x) {
                f
            })
        }
        if (is.character(f)) {
            return(function(x) {
                do.call(f, args = list(x))
            })
        }
        stop("Unable convert to a function.")
    }
    if (!is.list(x)) {
        fList <- list(x)
    }
    else {
        fList <- x
    }
    f <- fList[[1]]
    if (missing(ylab)) {
        if (length(fList) > 1) {
            ylab = "function value"
        }
        else {
            ylab = temp_ylab
        }
    }
    fList <- lapply(fList, makeFunction)
    if (!all(unlist(lapply(fList, is.function)))) {
        stop("first argument must be a function or list of functions")
    }
    if (missing(xlab)) {
        xlab = names(formals(fList[[1]]))[1]
    }
    if (missing(xlim)) {
        if (is.finite(do.call(fList[[1]], args = c(list(0), args)))) {
            xlim <- c(-2, 2)
        }
        else {
            xlim <- c(0, 2)
        }
    }
    ddd <- data.frame(x = numeric(0), y = numeric(0), group = character(0))
    id <- 0
    for (f in fList) {
        id <- id + 1
        x <- .adapt_seq(xlim[1], xlim[2], f = f, args = args,
            length.out = n)
        x <- unique(x)
        y <- do.call(f, args = c(list(x), args))
        ddd <- rbind(ddd, data.frame(x = x, y = y, group = rep(as.character(id),
            length(x))))
    }
    xyplot(y ~ x, ddd, groups = group, type = type, ylim = ylim,
        xlab = xlab, ylab = ylab, ...)
}
<environment: namespace:mosaic>
```

```
xyplot(rnorm(10) ~ rnorm(10))
```


rnorm(10)
rnorm(10)