```
> cv.lars
function (x, y, K = 10, index, trace = FALSE, plot.it = TRUE,
    se = TRUE, type = c("lasso", "lar", "forward.stagewise",
        "stepwise"), mode = c("fraction", "step"), ...)
{
    type = match.arg(type)
    if (missing(mode)) {
        mode = switch(type, lasso = "fraction", lar = "step",
            forward.stagewise = "fraction", stepwise = "step")
    }
    else mode = match.arg(mode)
    all.folds <- cv.folds(length(y), K)    ?
    if (missing(index)) {
        index = seq(from = 0, to = 1, length = 100)
        ~~if (mode == "step") {~~  { ~~don't use~~ "step)
            fit = lars(x, y, type = type, ...)    invoke lars
            nsteps = nrow(fit$beta)
            maxfold = max(sapply(all.folds, length))
            nsteps = min(nsteps, length(y) - maxfold)
            index = seq(nsteps)
        ~~}~~                                   100 value  K. 10次
    }                                          index
    residual matrix
    residmat <- matrix(0, length(index), K)
    for (i in seq(K)) {
        omit <- all.folds[[i]] ~~去掉一个fold~~. 第i个 fold, to be removed
        fit <- lars(x[-omit, , drop = FALSE], y[-omit], trace = trace,
            type = type, ...)
 X̂      fit <- predict(fit, x[omit, , drop = FALSE], mode = mode,
            s = index)$fit
        if (length(omit) == 1)
            fit <- matrix(fit, nrow = 1)  if length=1, 也变成 matrix.
        residmat[, i] <- apply((y[omit] - fit)^2, 2, mean)
        if (trace)              y  -  X̂(y).
            cat("\n CV Fold", i, "\n\n")
    }
    cv <- apply(residmat, 1, mean)  Averge over the folders.
    cv.error <- sqrt(apply(residmat, 1, var)/K) ✓
    object <- list(index = index, cv = cv, cv.error = cv.error,
        mode = mode)
    if (plot.it)
        plotCVLars(object, se = se)
    invisible(object)
}
<environment: namespace:lars>
>
```