Getting Started with sssvcqr

Houjian Hou

June 15, 2026

This vignette shows the basic workflow for sparse-smooth spatially varying coefficient quantile regression.

```
library("sssvcqr")

dat <- simulate_sssvcqr_data(n = 90, q = 1, p = 3, seed = 1)
dat$active

#>    x1    x2    x3
#>  TRUE FALSE  TRUE


fit <- ss_svcqr(
  y = dat$y,
  Z = dat$Z,
  X = dat$X,
  u = dat$u,
  tau = 0.5,
  lambda1 = 5,
  lambda2 = 0.1,
  k_nn = 8,
  control = list(max_iter = 80, warn_nonconvergence = FALSE)
)

fit

#> Sparse-smooth SVC quantile regression fit
#>   n = 90  q = 1  p = 3  tau = 0.5
#>   lambda1 = 5  lambda2 = 0.1
#>   iterations = 80  converged = FALSE

summary(fit)

#> Sparse-smooth SVC quantile regression summary
#>   n = 90  q = 1  p = 3  tau = 0.5
#>   lambda1 = 5  lambda2 = 0.1
#>   iterations = 80  converged = FALSE
#>
#> alpha:
#> [1] 0.7470532
#> beta_G:
#> [1]  1.7075697  1.3053485 -0.4766898
#> delta L2 norms:
#> [1] 0 0 0

selection_recovery_table(fit, dat)
```

```
#>   covariate_index covariate true_active true_deviation_norm
#> 1               1        x1        TRUE            9.959249
#> 2               2        x2       FALSE            0.000000
#> 3               3        x3        TRUE           18.424478
#>   estimated_deviation_norm selected_active
#> 1                        0           FALSE
#> 2                        0           FALSE
#> 3                        0           FALSE
```

```
head(predict(fit))
```

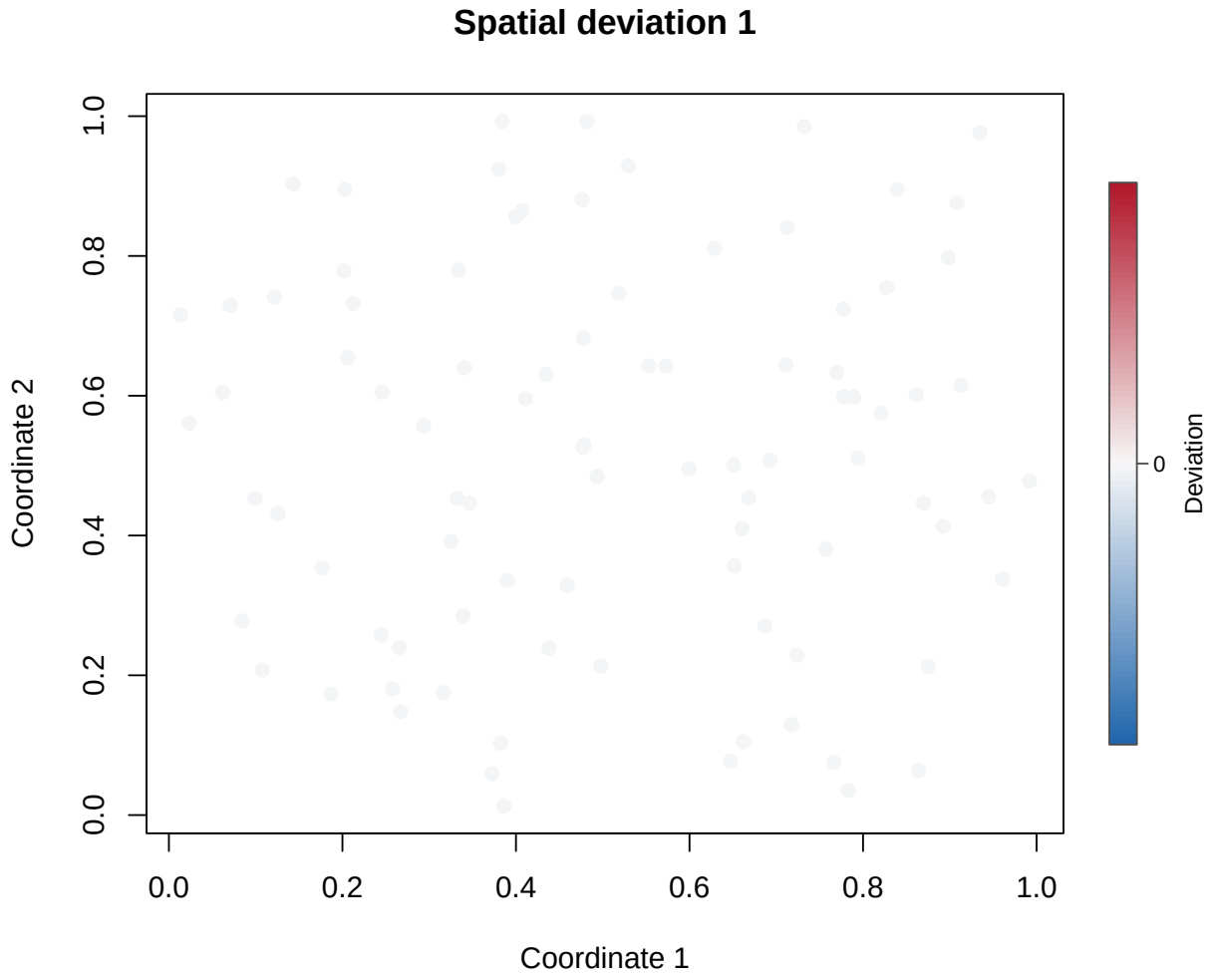
```
#> [1] -1.6101228  2.4708425  0.4334059  1.6807177  2.3594935  1.3296107
```

```
head(predict(fit, type = "coefficients"))
```

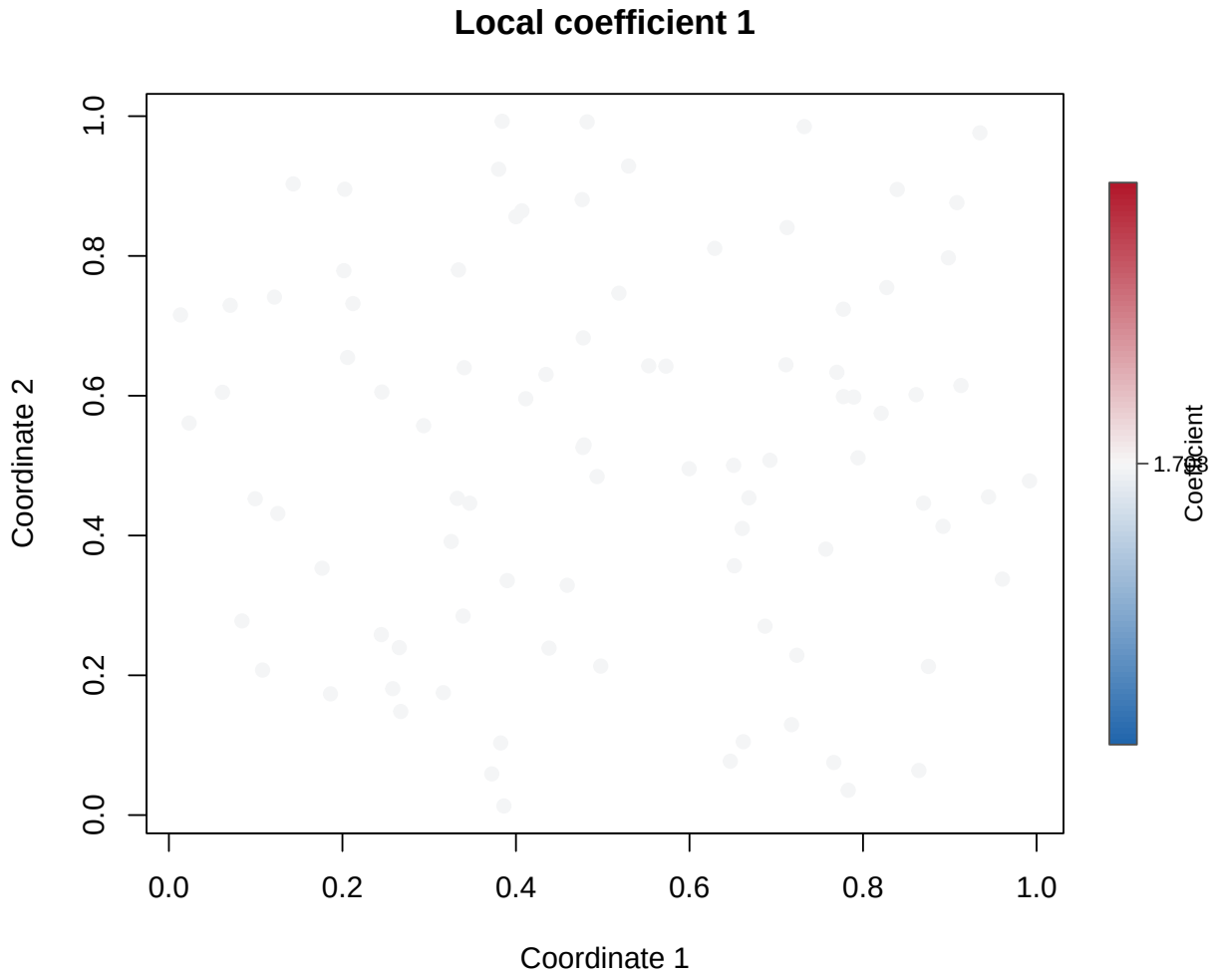
```
#>         [,1]     [,2]       [,3]
#> [1,] 1.70757 1.305348 -0.4766898
#> [2,] 1.70757 1.305348 -0.4766898
#> [3,] 1.70757 1.305348 -0.4766898
#> [4,] 1.70757 1.305348 -0.4766898
#> [5,] 1.70757 1.305348 -0.4766898
#> [6,] 1.70757 1.305348 -0.4766898
```

The spatial plot types use the first two columns of the coordinate matrix saved in the fitted object. Their colorbars show the numeric scale for deviations, local total coefficients, or residuals.

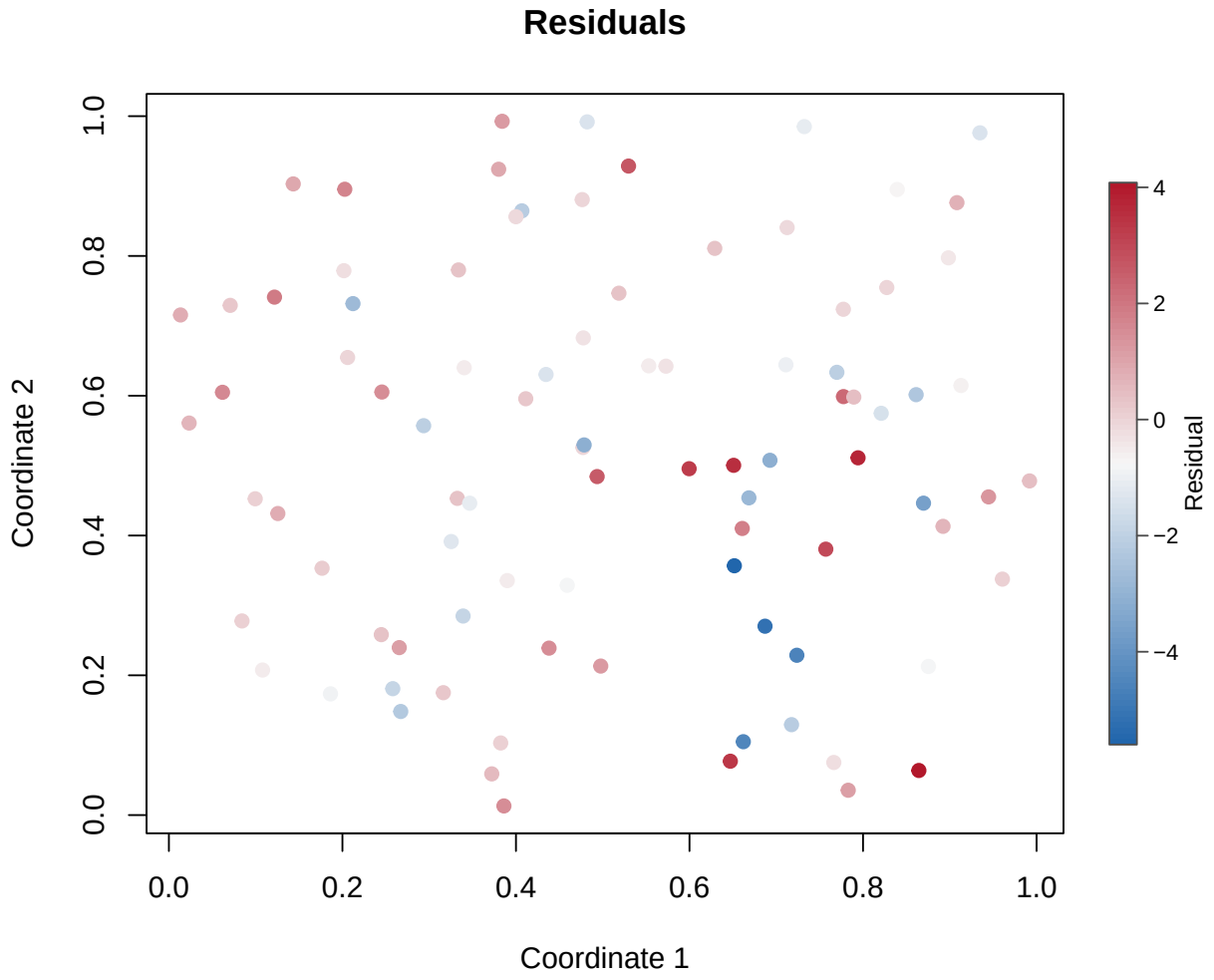
```
plot(fit, type = "deviation", index = 1)
```




```
plot(fit, type = "coefficient", index = 1)
```




```
plot(fit, type = "residual")
```



Penalty parameters can be selected by spatially blocked cross-validation. The small grid below is for demonstration; empirical applications should use a broader grid and more iterations.

```
cv <- cv_ss_svcqr(
  y = dat$y,
  Z = dat$Z,
  X = dat$X,
  u = dat$u,
  tau = 0.5,
  lambda1_seq = c(1, 2),
  lambda2_seq = c(0.5, 1),
  K_folds = 3,
  adaptive_weights = FALSE,
  control = list(max_iter = 25, warn_nonconvergence = FALSE)
)

cv

#> Spatially blocked CV for SS-SVCQR
#>   tau = 0.5
```

```
#>   best lambda1 = 1
#>   best lambda2 = 0.5
#>   best mean check loss = 0.7350447

cv$best

#> $lambda1
#> [1] 1
#>
#> $lambda2
#> [1] 0.5
#>
#> $cv_mean
#> [1] 0.7350447
#>
#> $cv_sd
#> [1] 0.1356838
```