Microbiome Data Simulation

Dahun Seo

2024-12-11

Contents

1 Data Generation

1.1 The logistic normal (LN) distribution

- generate random binary tree with p variables
- calculate cophenetic distanace between variables
- define variance-covariance matrix using distance matrix

$$\Sigma_{ij} = \exp(-d_{ij})/2$$

- generate data from multivariate normal distribution

$$M_i \sim \mathcal{N}_p(\alpha_0, \Sigma)$$

- transformation

$$Z_{ij} = \log\left(\frac{\exp(M_{ij})}{\sum_{j=1}^{p}\exp(M_{ij})}\right)$$

- generate outcome variable

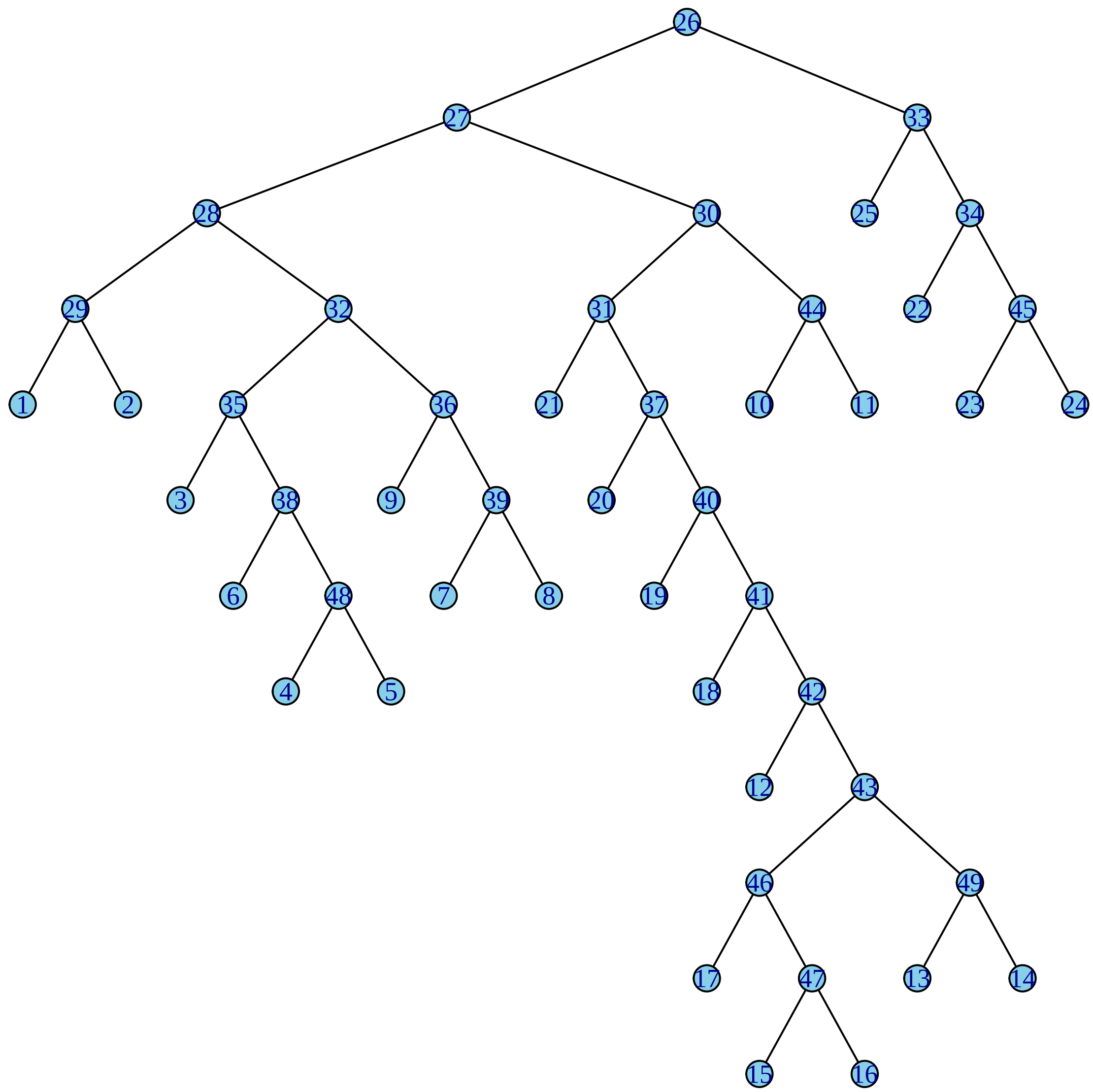
$$Y_i = \beta_0 + Z_i^\top \beta + \varepsilon$$

  where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.

```r
set.seed(1234)

# The logistic normal (LN) distribution
n <- 50  # n: sample size
p <- 25  # p: number of features
noise_sigma <- 1  # noise_sigma: noise level for response

# parameters for normal distribution base_mu <-
# rep(0, p)

# parameters for normal distribution Some taxa
# are often significantly more abundant than
# others, as commonly seen in real microbiome
```

```r
# compositional data.
base_mu <- c(rep(p/2, 5), rep(1, p - 5))

# Create a random binary tree for the p features
random_tree <- ape::rcoal(p)
```

```r
tree_info <- create_tree_structure(tips = 1:p, edges = random_tree$edge)
tree_info$levels
#> [[1]]
#>  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
#>
#> [[2]]
#> [1] 29 48 39 44 49 47 45
#>
#> [[3]]
#> [1] 38 36 46 34
#>
#> [[4]]
#> [1] 35 43 33
#>
#> [[5]]
#> [1] 32 42
#>
#> [[6]]
#> [1] 28 41
#>
#> [[7]]
#> [1] 40
#>
#> [[8]]
#> [1] 37
#>
#> [[9]]
#> [1] 31
#>
#> [[10]]
#> [1] 30
#>
#> [[11]]
#> [1] 27
#>
#> [[12]]
#> [1] 26

g <- igraph::graph_from_edgelist(random_tree$edge,
    directed = FALSE)

layout <- igraph::layout_as_tree(g, root = tree_info$levels[[length(tree_info$levels)]],
    mode = "out")

plot(g, layout = layout, vertex.label = igraph::V(g)$name,
    vertex.size = 5, vertex.label.cex = 0.8, vertex.color = "skyblue",
    edge.color = "black")
```



```
# Compute the distance matrix using cophenetic
# distances
dist_matrix <- stats::cophenetic(random_tree)

# variance-covariance setting
sigma <- exp(-dist_matrix)/2

# calculate mu for each i
mu_matrix <- matrix(NA, n, p)
for (i in 1:n) {
    mu_matrix[i, ] <- base_mu
}

# generation of logistic normal samples
z <- matrix(NA, n, p)
for (i in 1:n) {
    normal_sample <- MASS::mvrnorm(1, mu_matrix[i,
        ], sigma)
    exp_sample <- exp(normal_sample)
    z[i, ] <- exp_sample/sum(exp_sample)
}
```

```r
# label
colnames(z) <- paste0("x", 1:p)

# add the psudeo-count 0.5
z <- ifelse(z == 0, 0.5, z)
```

```r
# check
head(z)
#>             x1         x2         x3         x4         x5           x6
#> [1,] 0.4166178 0.07886284 0.12241441 0.18773823 0.19433201 1.721043e-06
#> [2,] 0.4822764 0.08763486 0.12712105 0.16419308 0.13875815 1.068401e-06
#> [3,] 0.6140859 0.14167069 0.04572877 0.08991144 0.10858594 1.011331e-06
#> [4,] 0.1873623 0.14168675 0.31800328 0.15651067 0.19639526 1.707286e-06
#> [5,] 0.3015840 0.09697176 0.45863457 0.08499018 0.05775715 8.713791e-07
#> [6,] 0.0855901 0.45506211 0.17909561 0.14151325 0.13863832 2.789405e-06
#>                x7           x8           x9          x10          x11
#> [1,] 1.155662e-06 1.804202e-06 1.229595e-06 2.238900e-06 2.070884e-06
#> [2,] 9.827634e-07 9.305502e-07 7.370918e-07 1.282054e-06 1.034070e-06
#> [3,] 6.210888e-07 9.085818e-07 7.507895e-07 1.038482e-06 7.120450e-07
#> [4,] 1.079628e-06 1.519743e-06 1.730878e-06 1.031367e-06 9.518118e-07
#> [5,] 2.104838e-06 1.533299e-06 2.222478e-06 3.748508e-06 3.396739e-06
#> [6,] 1.862062e-06 1.297728e-06 1.886145e-06 3.747733e-06 5.005015e-06
#>               x12          x13          x14          x15          x16
#> [1,] 1.342950e-06 1.513783e-06 1.215567e-06 1.146037e-06 1.154319e-06
#> [2,] 5.003072e-07 5.648774e-07 5.527305e-07 5.801050e-07 5.183376e-07
#> [3,] 6.615553e-07 6.268391e-07 5.775577e-07 7.885924e-07 8.250934e-07
#> [4,] 2.965191e-06 2.823846e-06 2.848904e-06 3.190506e-06 2.931662e-06
#> [5,] 4.667765e-06 4.363367e-06 4.829771e-06 3.555014e-06 3.823796e-06
#> [6,] 6.659235e-06 6.292166e-06 6.802304e-06 8.874532e-06 8.381462e-06
#>               x17          x18          x19          x20          x21
#> [1,] 1.147252e-06 1.115385e-06 1.346882e-06 1.435060e-06 1.630955e-06
#> [2,] 6.232795e-07 5.764124e-07 5.248963e-07 6.193002e-07 1.200945e-06
#> [3,] 8.904197e-07 7.095153e-07 6.232497e-07 5.318029e-07 9.840006e-07
#> [4,] 2.956579e-06 3.619698e-06 3.143121e-06 1.529270e-06 1.364783e-06
#> [5,] 4.263189e-06 2.845719e-06 4.595156e-06 1.001302e-05 1.952893e-06
#> [6,] 6.350940e-06 6.641948e-06 7.326805e-06 5.024688e-06 6.345650e-06
#>               x22          x23          x24          x25
#> [1,] 3.675162e-06 2.179832e-06 1.734461e-06 3.838480e-06
#> [2,] 9.428217e-07 9.132947e-07 9.172107e-07 1.337000e-06
#> [3,] 1.107713e-06 1.160148e-06 1.029808e-06 1.714572e-06
#> [4,] 1.182711e-06 1.584775e-06 1.713135e-06 1.908347e-06
#> [5,] 1.147831e-06 6.282569e-07 6.048532e-07 1.201952e-06
#> [6,] 4.257302e-06 4.372694e-06 4.038336e-06 2.650898e-06
apply(z, 1, sum)
#>  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
#> [39] 1 1 1 1 1 1 1 1 1 1 1 1
```

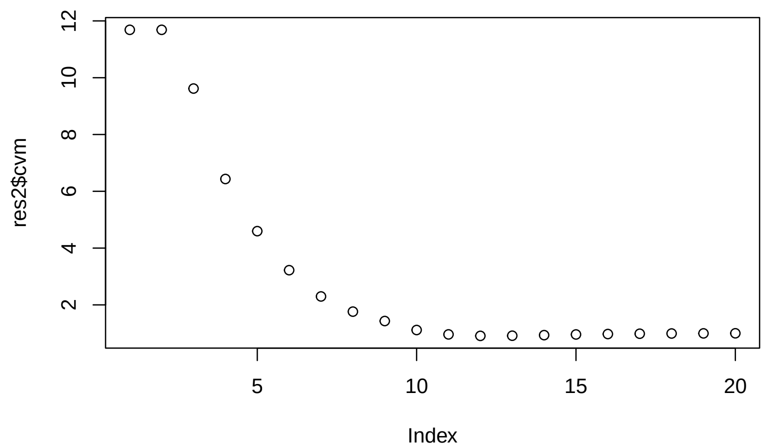
```r
# log transformation
log_z <- log(z)

# coefficients
beta_non_zero <- c(-3, 3, 2.5, -1, -1.5, 3, 3, -2,
    -2, -2, 1, -1, 3, -2, -1)

if (length(beta_non_zero) >= p) {
    beta <- beta_non_zero[1:p]
```

```
} else {
    beta <- c(beta_non_zero, rep(0, p - length(beta_non_zero)))
}

base_y <- rep(100, n)

y <- base_y + as.vector(log_z %*% beta) + stats::rnorm(n,
    0, sd = noise_sigma)
```

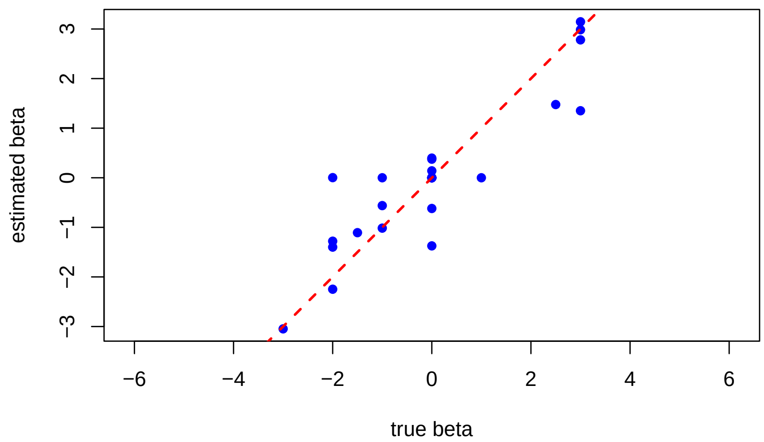
2 Simulation Study Results

2.1 Original

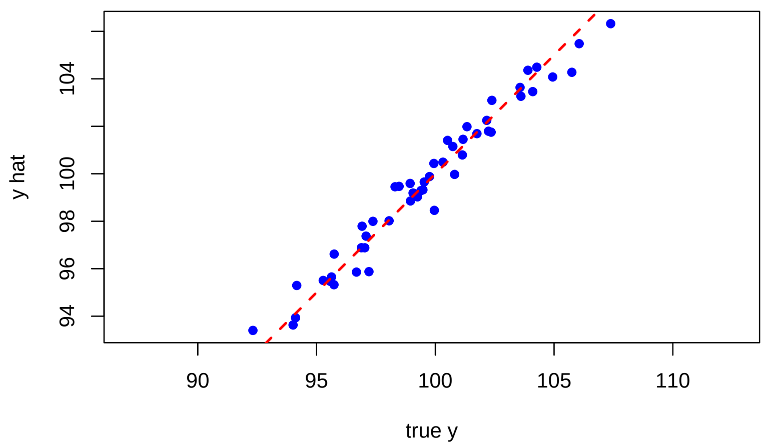
```
expanded_z <- construct_features(z, tips = 1:p, edges = random_tree$edge,
    eta = 0, tau = 1)

C <- matrix(1, p, ncol = 1)

res2 <- ConstrLassoCrossVal(y = y, x = log_z, C = C,
    nfolds = 10)

plot(res2$cvm)
```




```
res2$bet.sel
#>            x1           x2           x3           x4           x5           x6
#> -3.047419897  2.780903783  1.476457399 -1.015879206 -1.107517321  2.985398290
#>            x7           x8           x9          x10          x11          x12
#>  3.145699208 -1.279194233 -2.248240772 -1.398938666  0.000000000 -0.561615382
#>           x13          x14          x15          x16          x17          x18
#>  1.351593830  0.002219062  0.000000000 -1.373929289  0.000000000  0.000000000
#>           x19          x20          x21          x22          x23          x24
#>  0.139114416  0.000000000  0.396244166 -0.619560752  0.374665368  0.000000000
#>           x25
#>  0.000000000
t(C) %*% res2$bet.sel
#>              [,1]
#> [1,] 2.228595e-09

res2$cvm[res2$sel]
#> [1] 1.11507
res2$Rsq.sel
#> [1] 0.96367

{
    plot(beta, res2$bet.sel, xlab = "true beta", ylab = "estimated beta",
        pch = 16, col = "blue", asp = 1)
    abline(a = 0, b = 1, col = "red", lty = 2, lwd = 2)
}
```

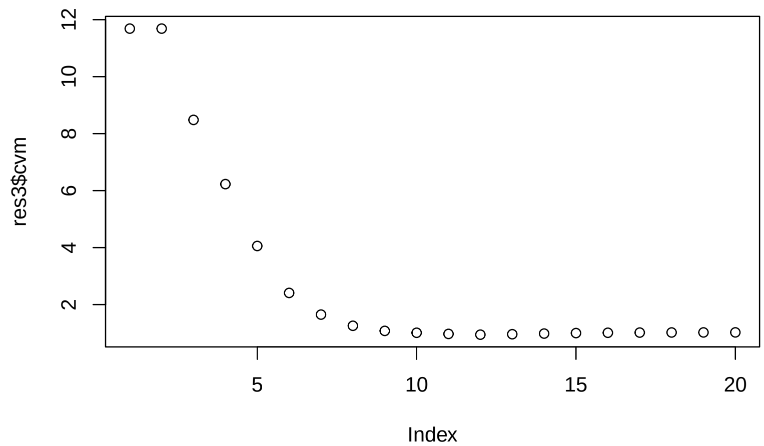



```
{
    plot(y, log_z %*% res2$bet.sel + res2$int.sel,
        xlab = "true y", ylab = "y hat", pch = 16,
        col = "blue", asp = 1)
    abline(a = 0, b = 1, col = "red", lty = 2, lwd = 2)
}
```

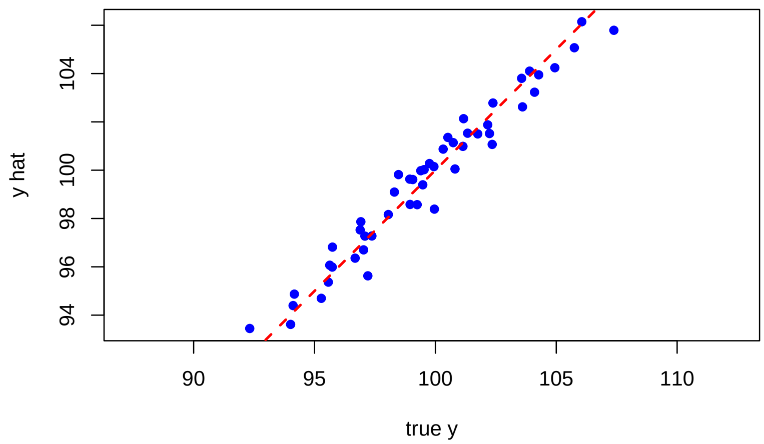


2.2 Expanded

```
head(expanded_z$data)
#>           h0_x1      h0_x2      h0_x3      h0_x4      h0_x5     h0_x6
#> [1,] -0.1733590 -1.8378182 -2.1003432 -0.7105559 -0.6760364 -12.31043
#> [2,] -0.1669633 -1.8723019 -2.0626155 -0.6125288 -0.7808393 -12.55517
#> [3,] -0.2075845 -1.6742140 -3.0850276 -0.7919508 -0.6032339 -12.18727
#> [4,] -0.5631628 -0.8425881 -1.1456936 -0.8130774 -0.5860723 -12.23906
#> [5,] -0.2787989 -1.4134275 -0.7795015 -0.5185404 -0.9048290 -12.00652
#> [6,] -1.8432066 -0.1723423 -1.7198355 -0.6829374 -0.7034623 -11.51727
#>           h0_x7      h0_x8      h0_x9     h0_x10     h0_x11    h0_x12
#> [1,] -0.9404700 -0.4950249 -1.2258867 -0.6549031 -0.7329122 -13.52064
#> [2,] -0.6662234 -0.7208159 -1.2797555 -0.5914314 -0.8063929 -14.50804
#> [3,] -0.9013337 -0.5209228 -1.1110072 -0.5221567 -0.8995313 -14.22867
#> [4,] -0.8786531 -0.5367282 -0.9169961 -0.6538160 -0.7340889 -12.72857
#> [5,] -0.5472332 -0.8640502 -0.9696318 -0.6450894 -0.7436316 -12.27483
#> [6,] -0.5288210 -0.8898906 -0.9840481 -0.8482165 -0.5589274 -11.91951
#>          h0_x13     h0_x14     h0_x15     h0_x16    h0_x17    h0_x18    h0_x19
#> [1,] -0.5894517 -0.8088528 -0.6967540 -0.6895533 -1.100311 -13.70631 -13.51772
#> [2,] -0.6823372 -0.7040753 -0.6384395 -0.7510219 -1.016085 -14.36644 -14.46007
#> [3,] -0.6530442 -0.7349258 -0.7160266 -0.6707796 -1.033994 -14.15868 -14.28832
#> [4,] -0.6975742 -0.6887397 -0.6517369 -0.7363467 -1.121903 -12.52912 -12.67029
#> [5,] -0.7452133 -0.6436583 -0.7302533 -0.6573688 -1.004602 -12.76969 -12.29051
#> [6,] -0.7328846 -0.6549286 -0.6649740 -0.7221372 -1.312938 -11.92211 -11.82397
#>         h0_x20    h0_x21     h0_x22     h0_x23     h0_x24    h0_x25     h1_x29
#> [1,] -13.45430 -13.32634 -0.7251626 -0.5853870 -0.8139380 -12.47043 -0.7022269
#> [2,] -14.29468 -13.63240 -1.0789258 -0.6952888 -0.6910101 -13.52508 -0.5622745
#> [3,] -14.44699 -13.83164 -1.0909184 -0.6353339 -0.7545090 -13.27635 -0.2800359
#> [4,] -13.39072 -13.50452 -1.3319523 -0.7328466 -0.6549638 -13.16927 -1.1115486
#> [5,] -11.51162 -13.14620 -0.7296219 -0.6743457 -0.7123090 -13.63156 -0.9199079
#> [6,] -12.20115 -11.96774 -1.0904697 -0.6541647 -0.7337112 -12.84061 -0.6149791
#>             h1_x48     h1_x39    h1_x44    h1_x49     h1_x47     h1_x45
#> [1,] -4.504509e-06 -0.3474283 -12.35462 -12.81145 -0.4046166 -0.6621251
#> [2,] -3.526636e-06 -0.3258760 -12.97562 -13.70432 -0.4494317 -0.4154556
#> [3,] -5.094921e-06 -0.3993248 -13.25559 -13.62953 -0.4394107 -0.4093344
#> [4,] -4.837782e-06 -0.5103557 -13.13081 -12.07984 -0.3940199 -0.3064728
#> [5,] -6.104328e-06 -0.4767828 -11.84906 -11.59705 -0.4560068 -0.6579562
#> [6,] -9.956723e-06 -0.4680773 -11.64614 -11.24332 -0.3133809 -0.4095614

res3 <- ConstrLassoCrossVal(y = y, x = expanded_z$data,
    C = expanded_z$C, nfolds = 10)

plot(res3$cvm)
```




```
res3$bet[, which.min(res3$cvm)]
#>          h0_x1          h0_x2          h0_x3          h0_x4          h0_x5
#> -2.8834057389  2.8834057460  1.6259982611 -0.6297848585  0.6297848584
#>          h0_x6          h0_x7          h0_x8          h0_x9         h0_x10
#>  2.9088122110  2.4340338650 -2.4340338630 -1.6127478053 -1.6313757653
#>         h0_x11         h0_x12         h0_x13         h0_x14         h0_x15
#>  1.6313757649 -1.6672803276 -0.0567503639  0.0567503617  3.5324045751
#>         h0_x16         h0_x17         h0_x18         h0_x19         h0_x20
#> -3.5324045739  0.0000000000 -0.6752564624  0.0000000000 -0.1539252043
#>         h0_x21         h0_x22         h0_x23         h0_x24         h0_x25
#>  0.6609644159 -0.4558691330  1.8007568516 -1.8007568565  0.0005281182
#>         h1_x29         h1_x48         h1_x39         h1_x44         h1_x49
#> -0.7836493332 -2.9088122118  1.6127478060 -1.3847698032  1.5728165206
#>         h1_x47         h1_x45
#>  0.0000000000  0.4558691336

t(expanded_z$C) %*% res3$bet[, which.min(res3$cvm)]
#>                [,1]
#>  [1,]  7.135238e-09
#>  [2,] -9.520773e-11
#>  [3,]  2.059915e-09
#>  [4,] -4.027703e-10
#>  [5,] -2.181549e-09
#>  [6,]  1.188188e-09
#>  [7,] -4.927729e-09
#>  [8,] -8.617551e-10
#>  [9,]  6.963723e-10
#> [10,]  0.000000e+00
#> [11,]  5.120674e-10

res3$cvm[res3$sel]
#> [1] 1.076272
res3$Rsq.sel
#> [1] 0.9560572

{
    plot(y, expanded_z$data %*% res3$bet.sel + res3$int.sel,
        xlab = "true y", ylab = "y hat", pch = 16,
        col = "blue", asp = 1)
    abline(a = 0, b = 1, col = "red", lty = 2, lwd = 2)
}
```


y hat
94
96
98
100
104
90
95
100
105
110
true y