FLASH vs Other Confounder Adjustment Methods

David Gerard

April 25, 2016

Abstract

I compare the performance of FLASH using known factors against other confounder adjustment methods. FLASH is conservative in estimating π_0, but does not perform as well as SUCCOTASH or CATE + ASH in terms of AUC. I am getting different results when running CATE + ASH than in previous simulations, and am currently trying to figure out why.

1 Methods

I first normalized the counts by $\log_2(COUNTS + 1)$. The number of hidden confounders was estimated using the methods of Buja and Eyuboglu [1992] implemented in the `num.sv()` function in the `sva` package in `R`.

The confounder adjustment methods I look at in this write-up are:

- Homoscedastic FLASH using the covariates as known factors. I capped the number of hidden confounders by the output of `num.sv()`. On its own, FLASH would estimate approximately 60 unknown factors, which would result in a very long run-time for the backfitting algorithm, making simulations infeasible without some sort of parallelization.
- OLS + qvalue.
- OLS + ASH.
- SUCCOTASH using normal mixtures and heteroscedastic PCA as the factor-analysis method.
- The robust regression version of CATE using PCA as the factor analysis method + qvalue.
- The robust regression version of CATE using PCA as the factor analysis method + ASH.
- SVA + qvalue.
- Negative control version of CATE using PCA as the factor analysis method + qvalue.
- Negative control version of CATE using PCA as the factor analysis method + ASH.
- RUV2 + qvalue.
- RUV4 + qvalue.

2 Simulation Study

I ran through 100 repetitions of generating data from GTEX lung data under the following parameter conditions:

- $n \in \{10, 20, 40\}$,
- $p = 1000$.
- $\pi_0 \in \{0.5, 0.9\}$,

- $\sigma_{log2} = 1$.

I extracted the most expressed p genes (excluding the top 5 expressed genes) from the GTEX lung data and n samples are chosen at random. Half of these samples are randomly given the "treatment" label 1, the other half given the "control" label 0. Of the p genes, $\pi_0 p$ were chosen to be non-null. Signal was added by the Poisson-thinning approach in Mengyin's code with a mean log2-fold change of 0 and a standard deviation log2-fold change of σ_{log2}. That is

$$A_1, \ldots, A_{p/2} \sim N(0, \sigma^2_{log2}) \tag{1}$$

$$B_i = 2^{A_i} \text{ for } i = 1, \ldots, p/2. \tag{2}$$

If $A_i > 0$ then we replace $Y_{[1:(n/2),i]}$ with $Binom(Y_{[j,i]}, 1/B_i)$ for $j = 1, \ldots, n/2$. If $A_i < 0$ then we replace $Y_{[(n/2+1):n,i]}$ with $Binom(Y_{[j,i]}, B_i)$ for $j = n/2 + 1, \ldots, n$.

I now describe the justification for this. Suppose that

$$Y_{ij} \sim Poisson(\lambda_j). \tag{3}$$

I.e., each individual has the Poisson parameter for gene j. Let x_i be the indicator of treatment vs control for individual i. Let Ω be the set of non-null genes. Let Z be the new dataset derived via the steps above. That is

$$Z_{ij} = \begin{cases} 2^{A_j x_i} Y_{ij} & \text{if } A_j < 0 \text{ and } j \in \Omega \\ 2^{-A_j(1-x_i)} Y_{ij} & \text{if } A_j > 0 \text{ and } j \in \Omega \\ Y_{ij} & \text{if } j \notin \Omega. \end{cases} \tag{4}$$

Then

$$Z_{ij} | A_j, A_j < 0, j \in \Omega \sim Poisson(2^{A_j x_i} \lambda_j) \tag{5}$$

$$Z_{ij} | A_j, A_j > 0, j \in \Omega \sim Poisson(2^{-A_j(1-x_i)} \lambda_j), \tag{6}$$

and

$$E[\log_2(Z_{ij}) - \log_2(Z_{kj}) | A_j, A_j < 0, j \in \Omega] \approx A_j x_i - A_j x_k, \text{ and} \tag{7}$$

$$E[\log_2(Z_{ij}) - \log_2(Z_{kj}) | A_j, A_j < 0, j \in \Omega] \approx -A_j(1 - x_i) + A_j(1 - x_k). \tag{8}$$

if individual i is in the treatment group and individual k is in the control group, then this just equals A_j. I treat the A_j's as the true coefficient values when calculating the MSE below.

For each iteration, I calculated three things:

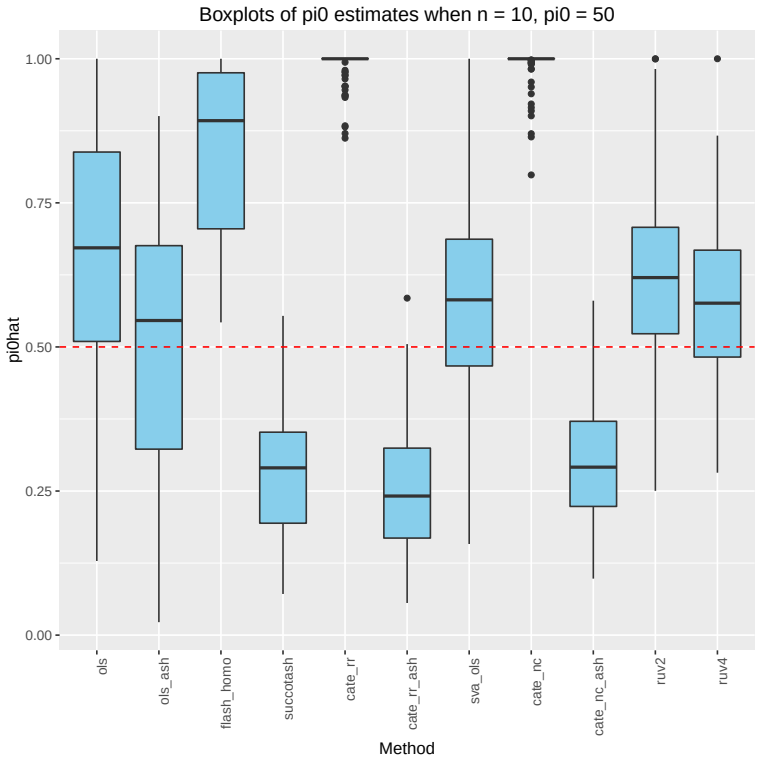
1. The AUC using the lfdr's.
2. The estimates of π_0.
3. The mean squared error from the A_j's.

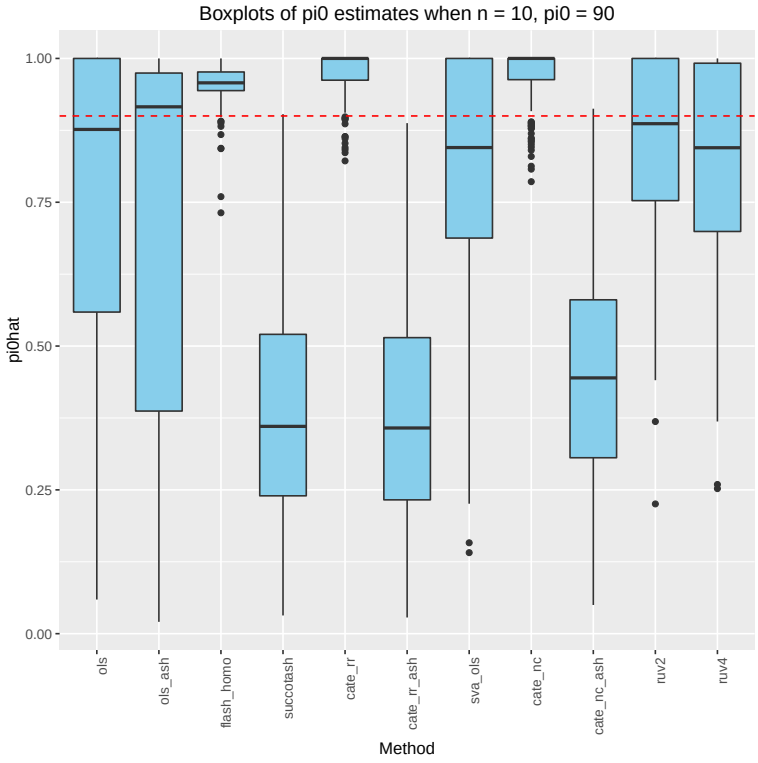
3 Results

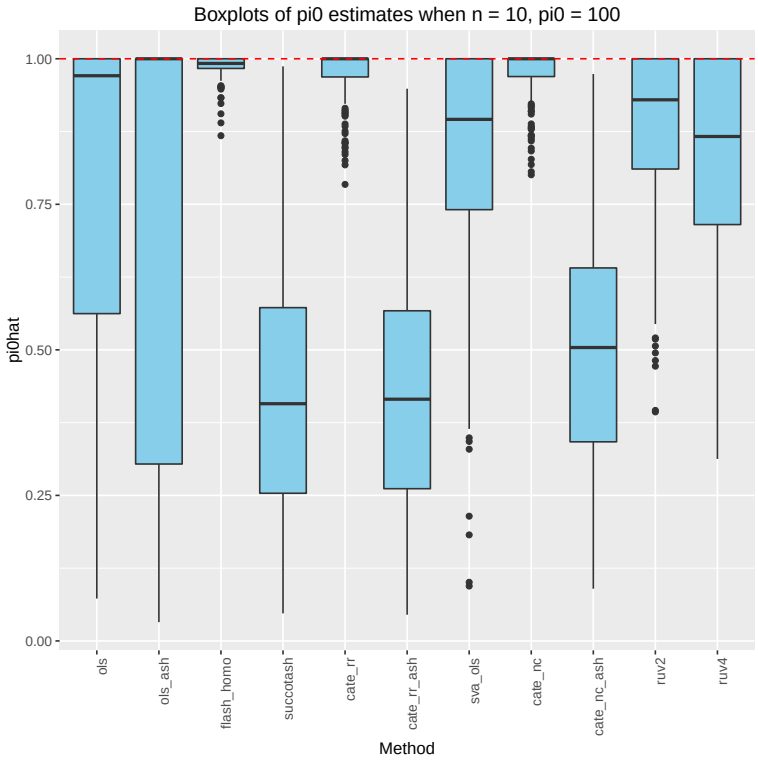
FLASH is in general conservative, but overly so. It does OK in terms of AUC, but not as well as SUCCOTASH or the CATE + ASH methods.

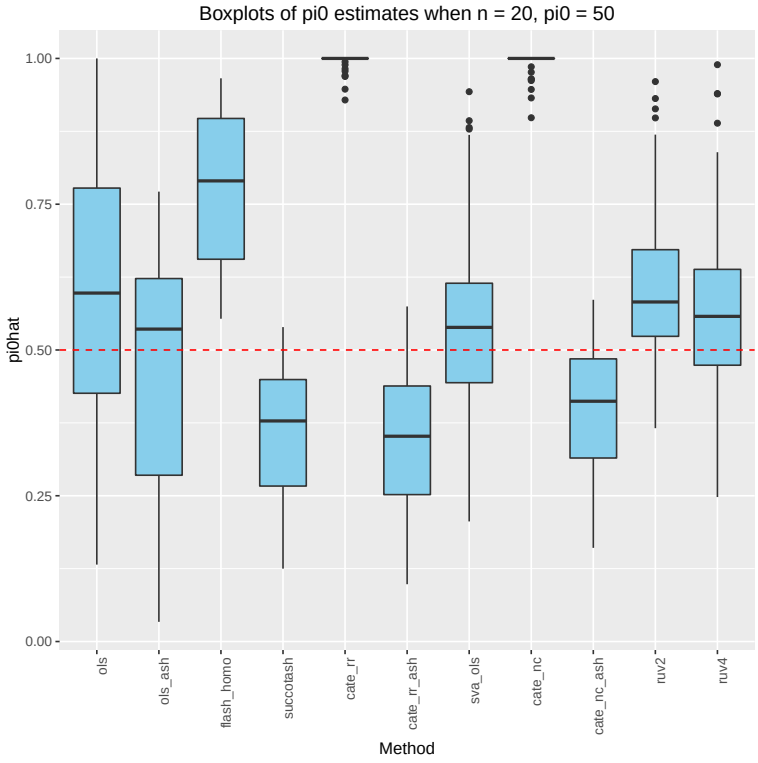
I am seeing something different in the CATE + ASH sims than I saw in previous CATE + ASH simulations. I am seeing ASH behave very similarly to SUCCOTASH — underestimating π_0 and doing really well in terms of AUC and MSE. I've been looking at my code for awhile, and I can't find the difference between my old code and my new code. To rule out that it's because of a newer version of ASH, I am rerunning those old simulations.

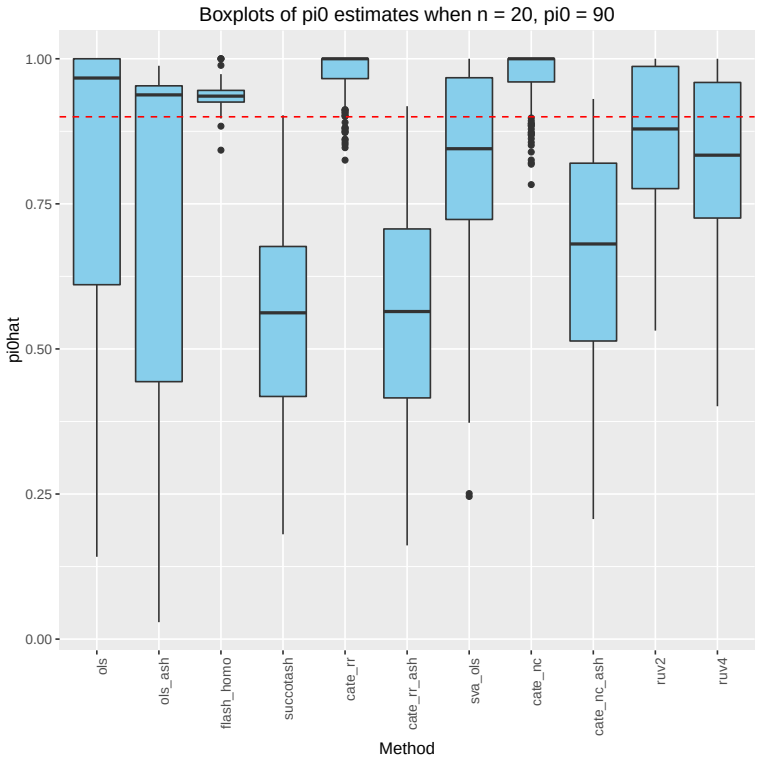
I guess I got some infinite values for MSE's when I did OLS + ASH for three of the scenarios. I don't know how that happened. Maybe my rerunning the simulations will reveal something.

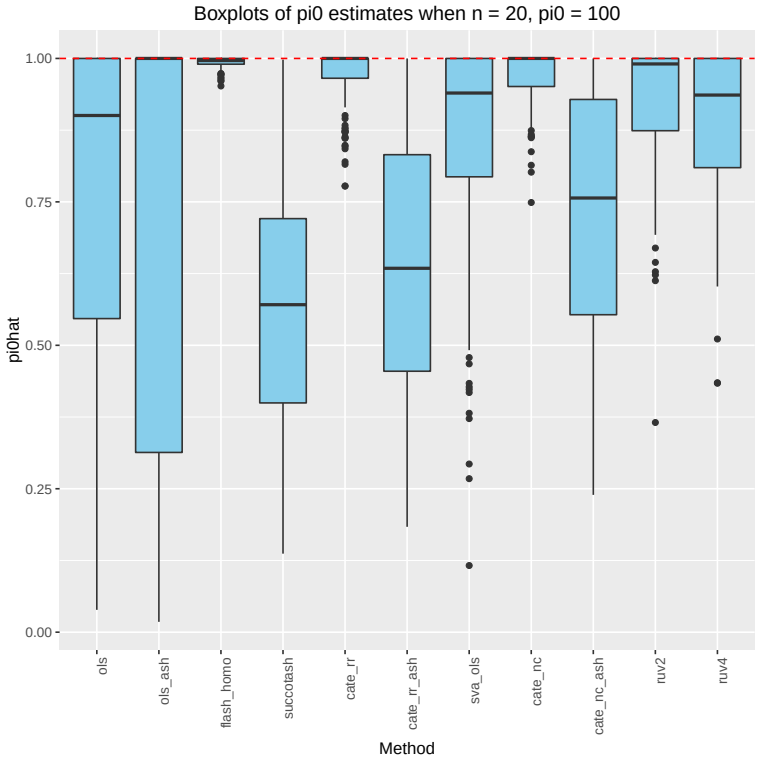

Boxplots of pi0 estimates when n = 10, pi0 = 50
pi0hat
1.00
0.75
0.50
0.25
0.00
ols
ols_ash
flash_homo
succotash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method

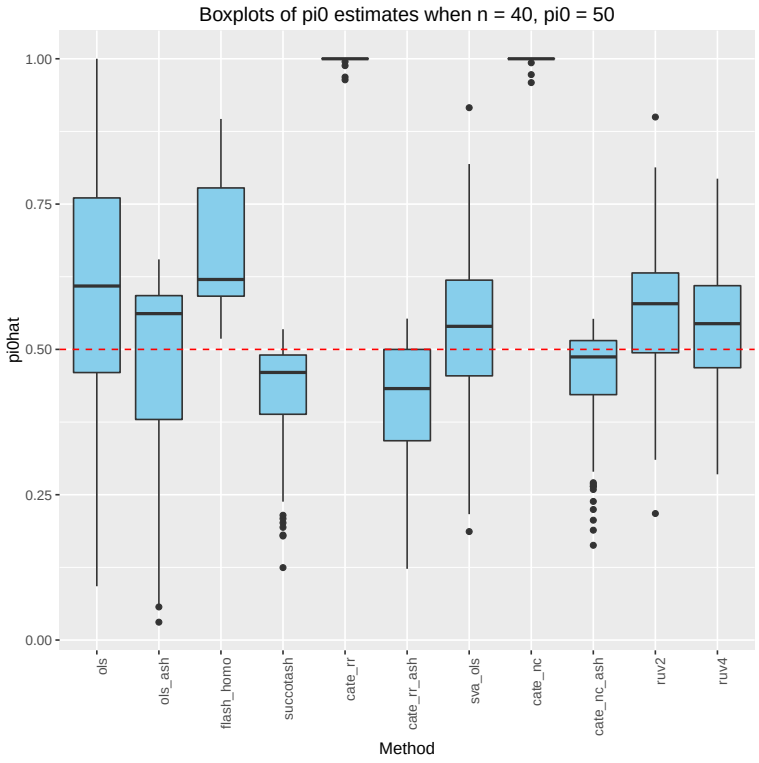

Boxplots of pi0 estimates when n = 10, pi0 = 90
pi0hat
1.00
0.75
0.50
0.25
0.00
ols
ols_ash
flash_homo
succotash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method

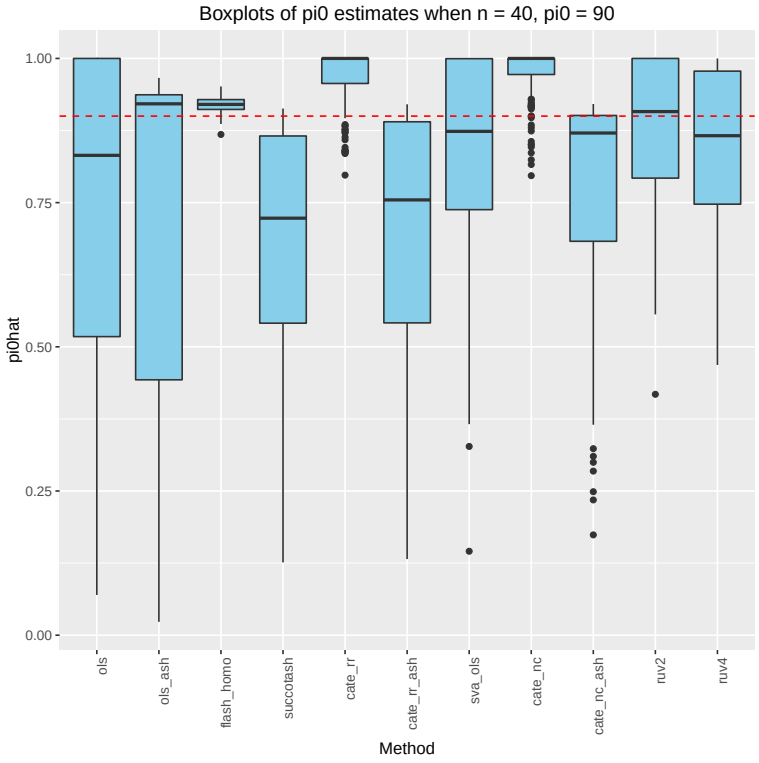

Boxplots of pi0 estimates when n = 10, pi0 = 100
pi0hat
1.00
0.75
0.50
0.25
0.00
ols
ols_ash
flash_homo
smoothash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method

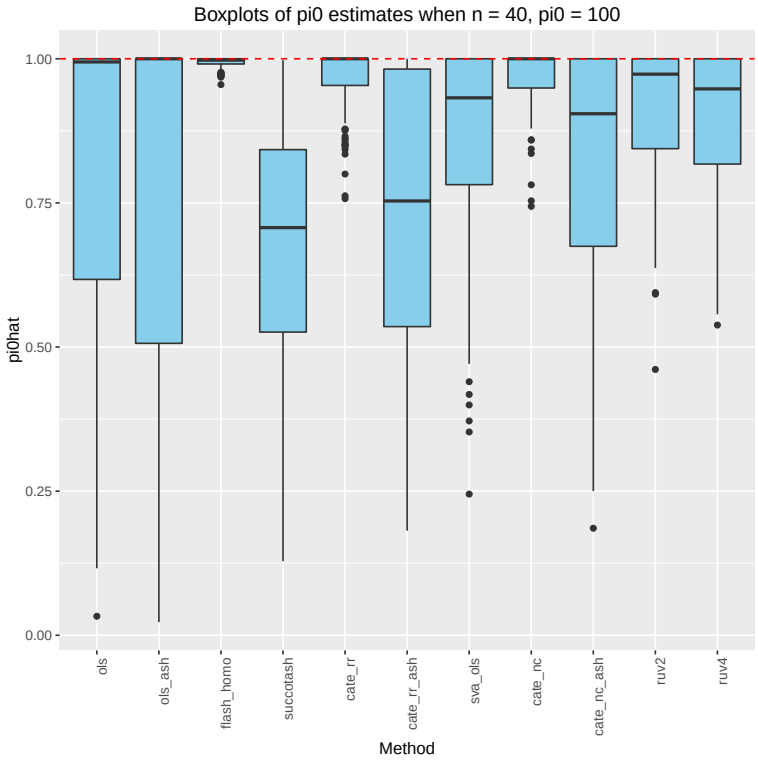

Boxplots of pi0 estimates when n = 20, pi0 = 50
pi0hat
1.00
0.75
0.50
0.25
0.00
ols
ols_ash
flash_homo
smoothash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method

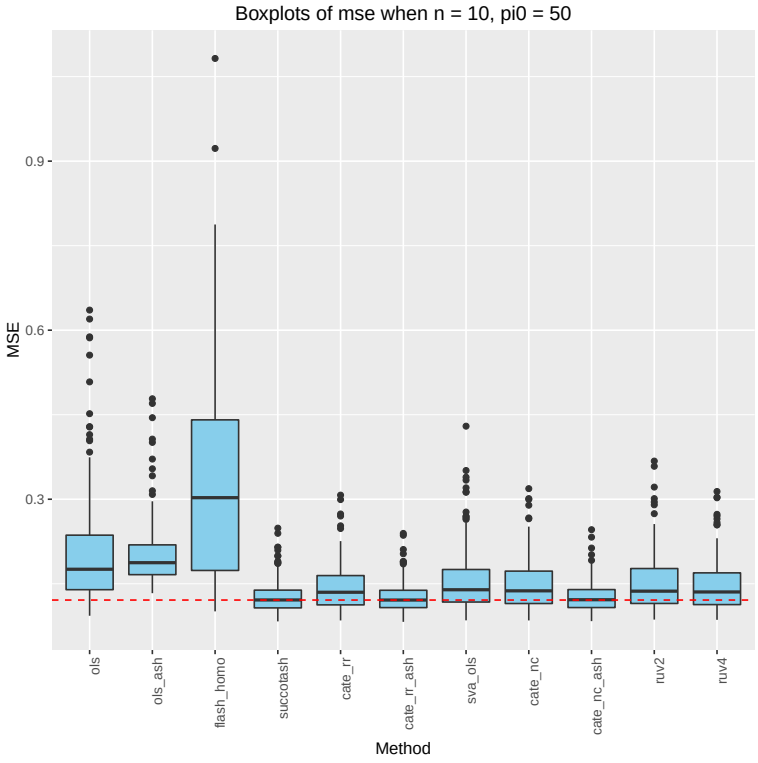

Boxplots of pi0 estimates when n = 20, pi0 = 90
pi0hat
1.00
0.75
0.50
0.25
0.00
ols
ols_ash
flash_homo
succotash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method

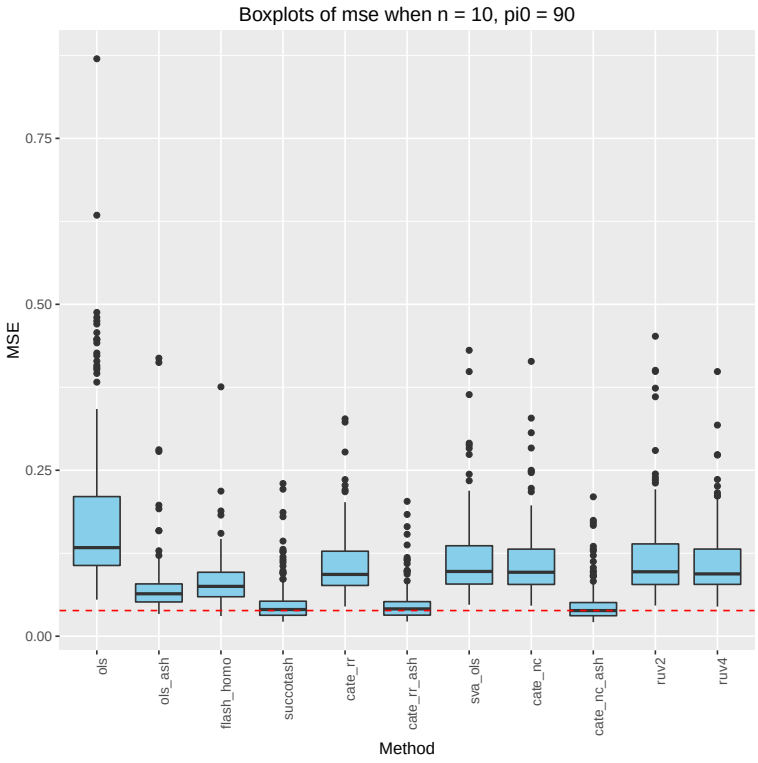

Boxplots of pi0 estimates when n = 20, pi0 = 100
pi0hat
1.00
0.75
0.50
0.25
0.00
ols
ols_ash
flash_homo
succotash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method

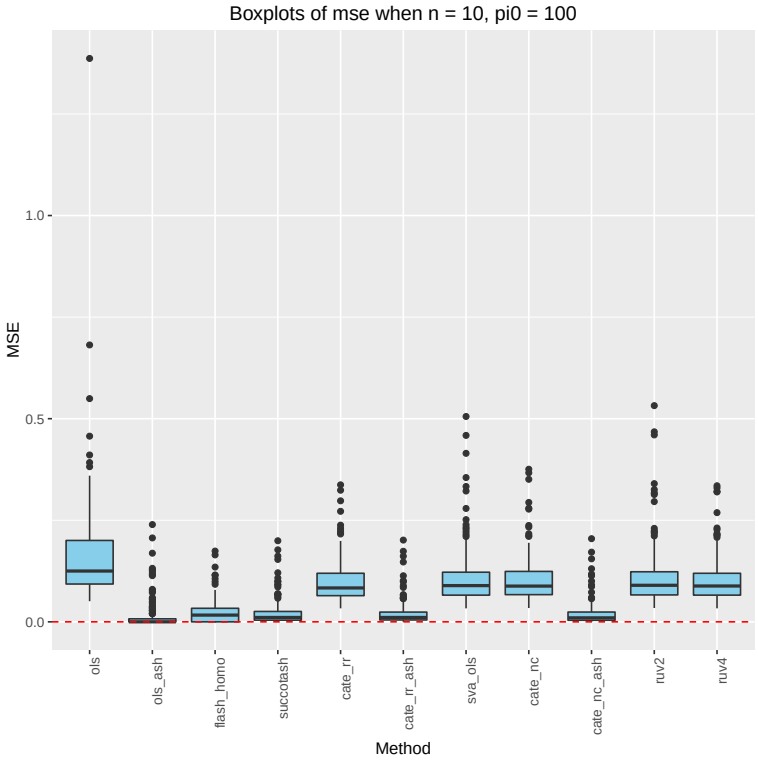

Boxplots of pi0 estimates when n = 40, pi0 = 50
pi0hat
1.00
0.75
0.50
0.25
0.00
ols
ols_ash
flash_homo
smoothash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method

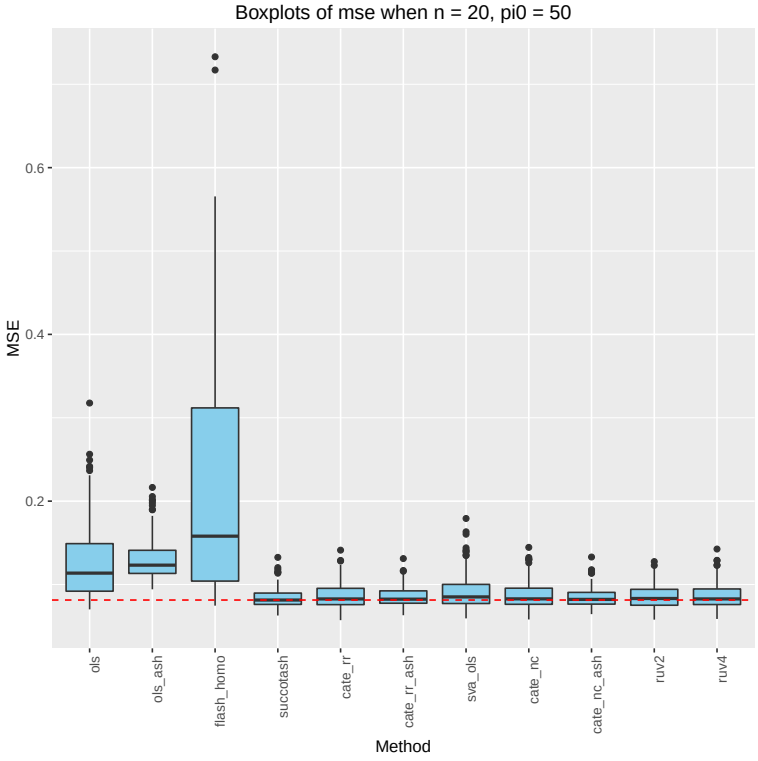

Boxplots of pi0 estimates when n = 40, pi0 = 90
pi0hat
1.00
0.75
0.50
0.25
0.00
ols
ols_ash
flash_homo
smoothash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method

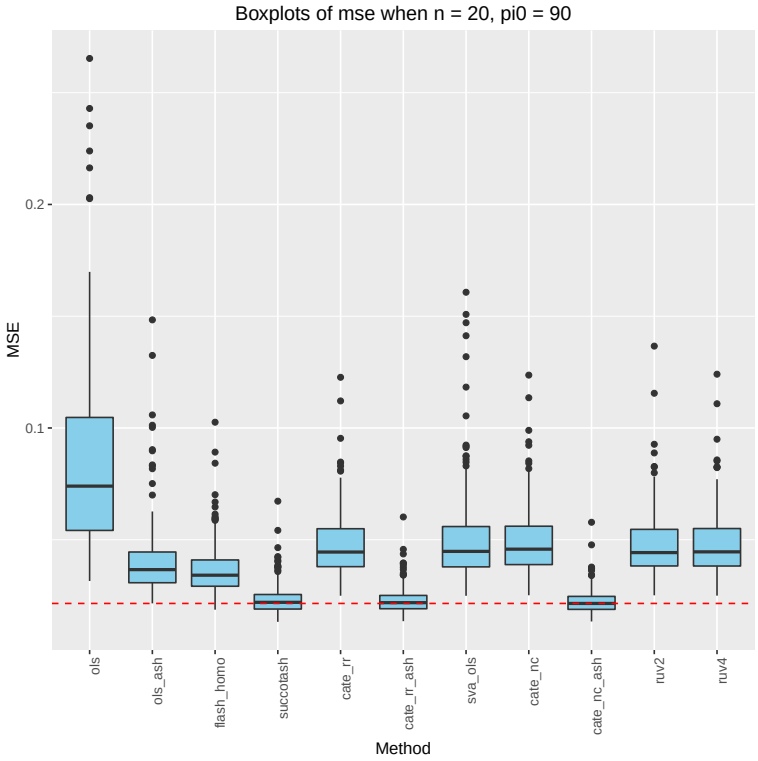

Boxplots of pi0 estimates when n = 40, pi0 = 100
pi0hat
1.00
0.75
0.50
0.25
0.00
ols
ols_ash
flash_homo
succotash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method

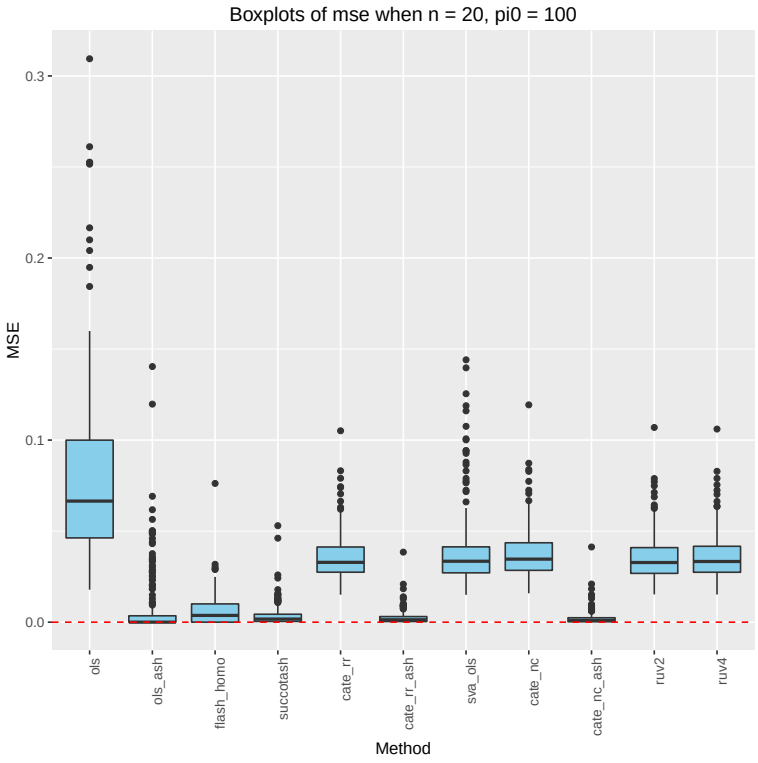

Boxplots of mse when n = 10, pi0 = 50
MSE
0.9
0.6
0.3
ols
ols_ash
flash_homo
succotash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method


Boxplots of mse when n = 10, pi0 = 90
MSE
0.75
0.50
0.25
0.00
ols
ols_ash
flash_homo
succotash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method

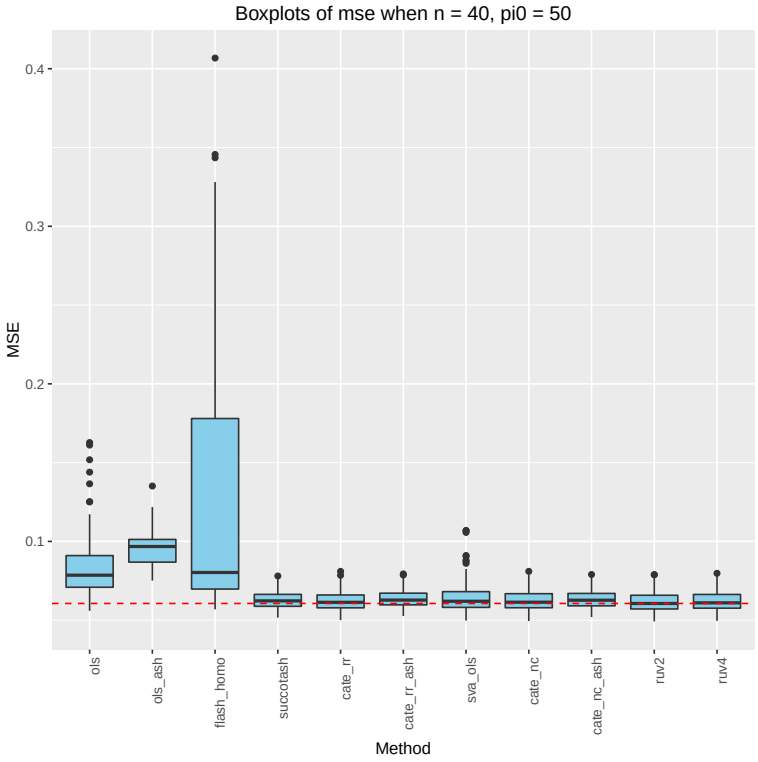

Boxplots of mse when n = 10, pi0 = 100
MSE
1.0
0.5
0.0
ols
ols_ash
flash_homo
succotash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method


Boxplots of mse when n = 20, pi0 = 50
MSE
0.6
0.4
0.2
ols
ols_ash
flash_homo
succotash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method

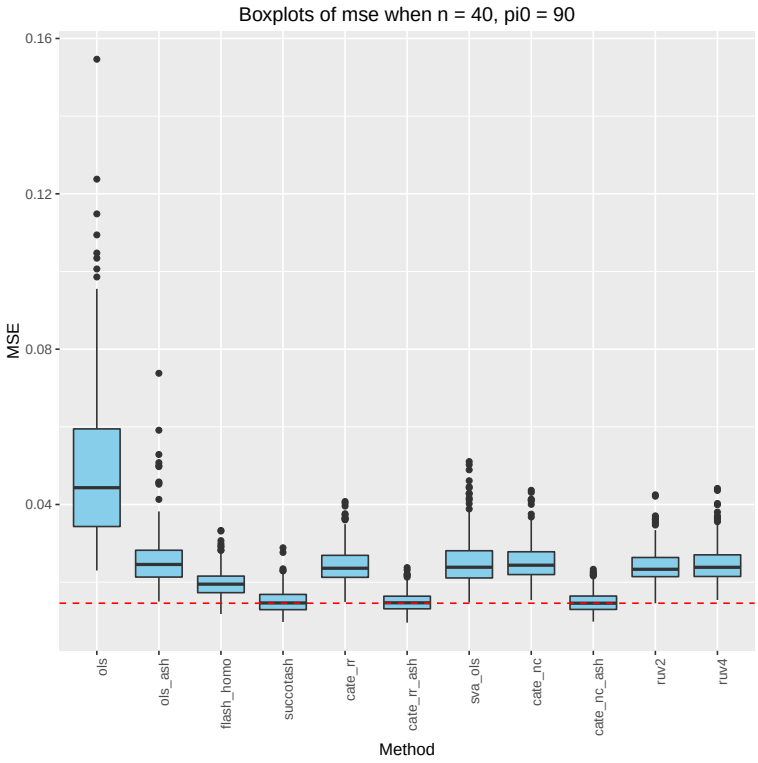

Boxplots of mse when n = 20, pi0 = 90
MSE
0.2
0.1
ols
ols_ash
flash_homo
succotash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method

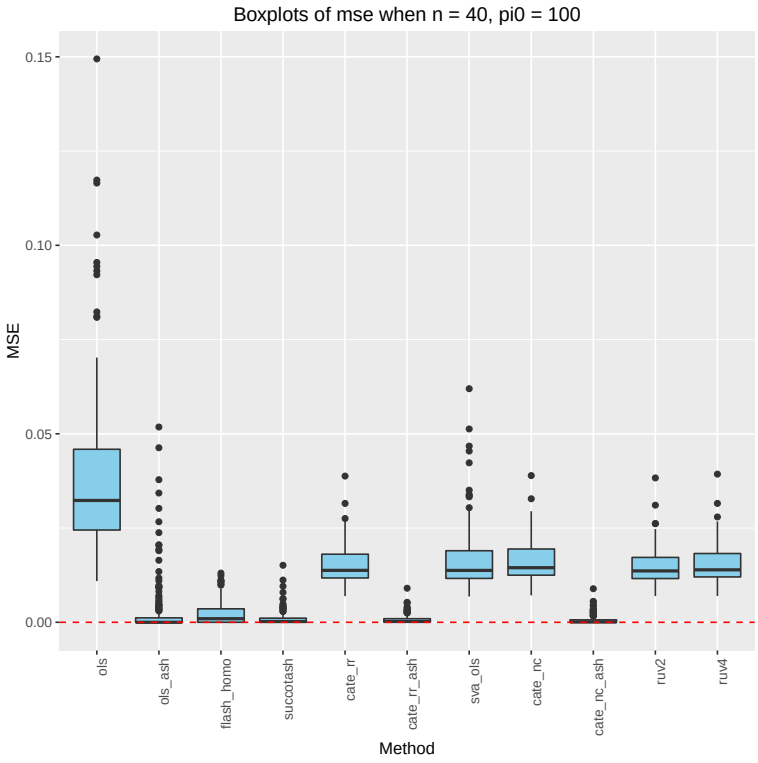


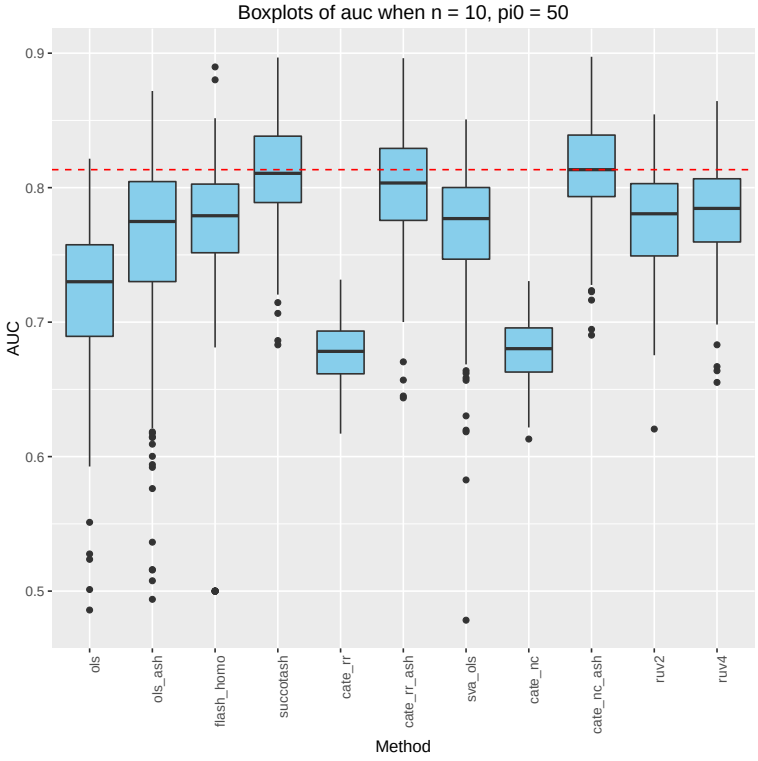

```
## Warning: Removed 171 rows containing non-finite values (stat_boxplot).
```

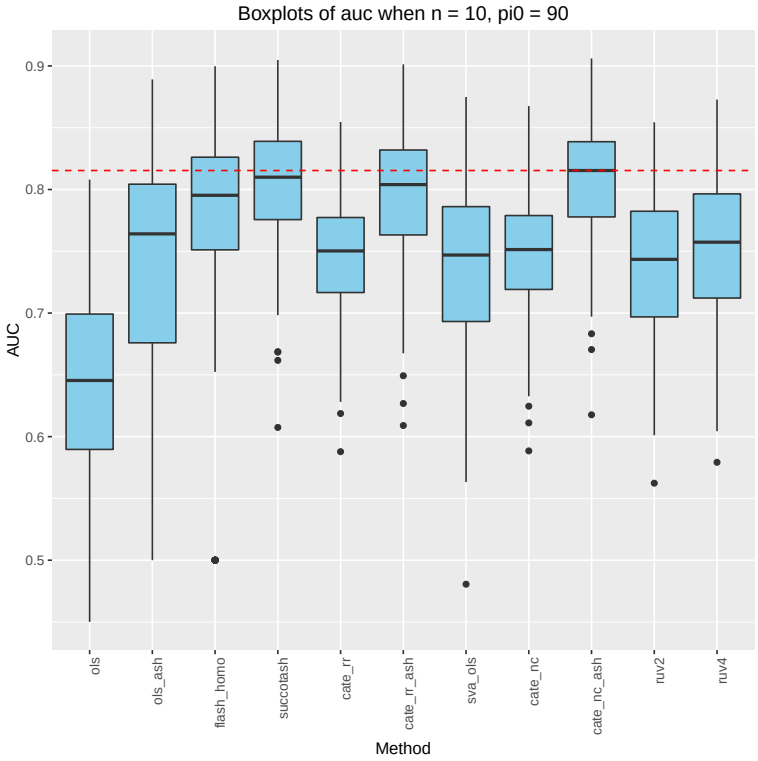


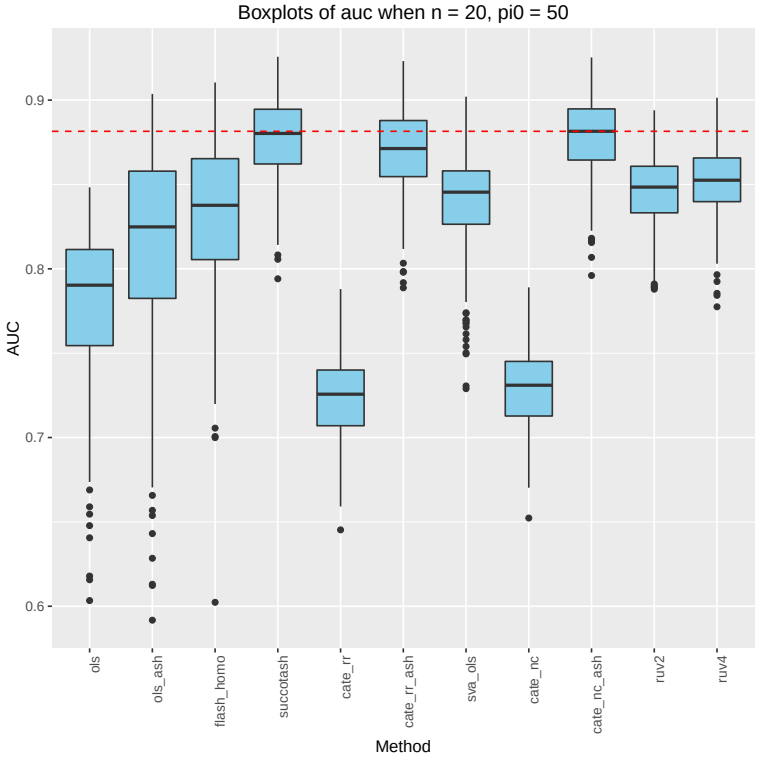
```
## Warning:  Removed 85 rows containing non-finite values (stat_boxplot).
```

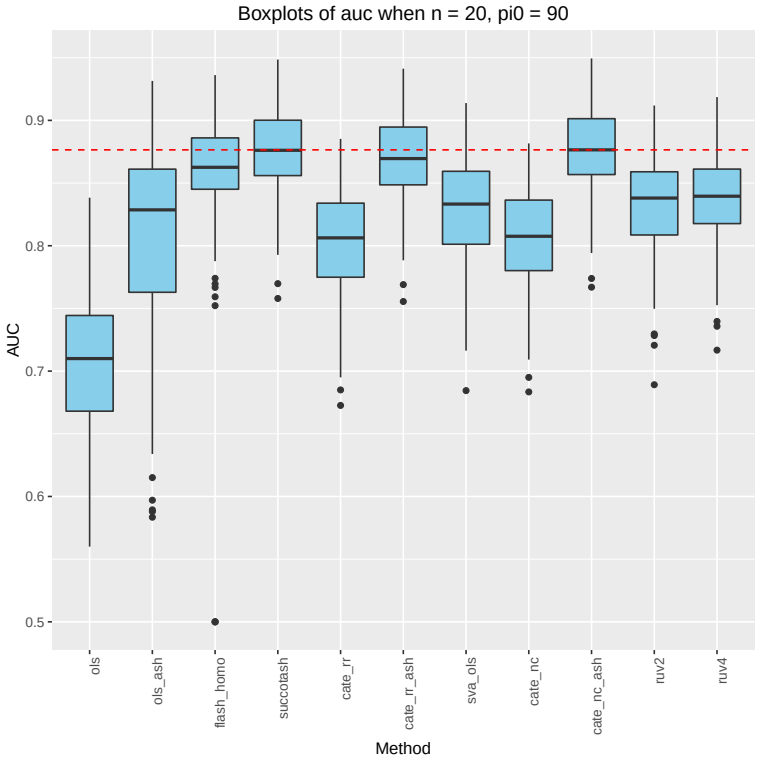


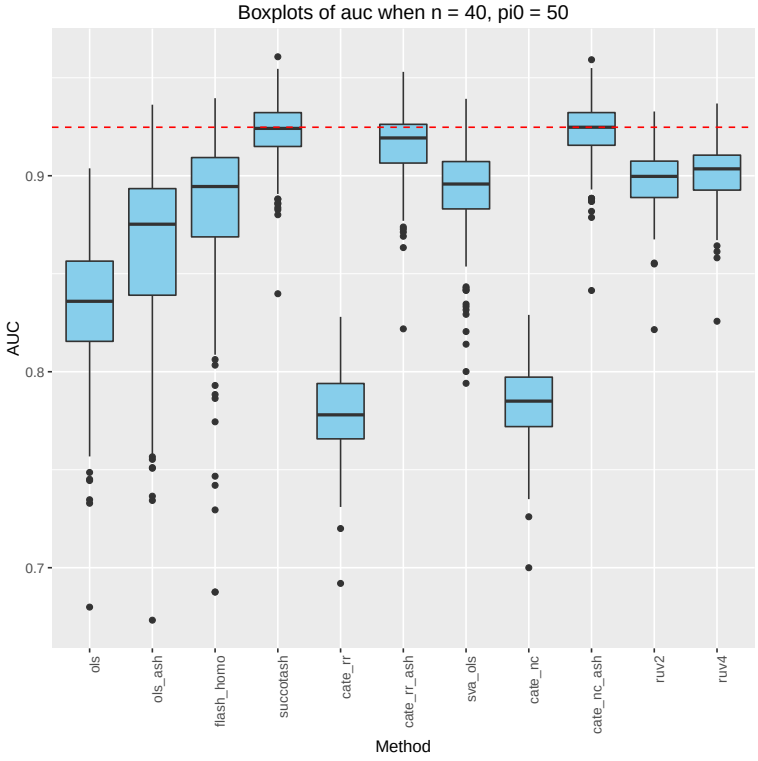


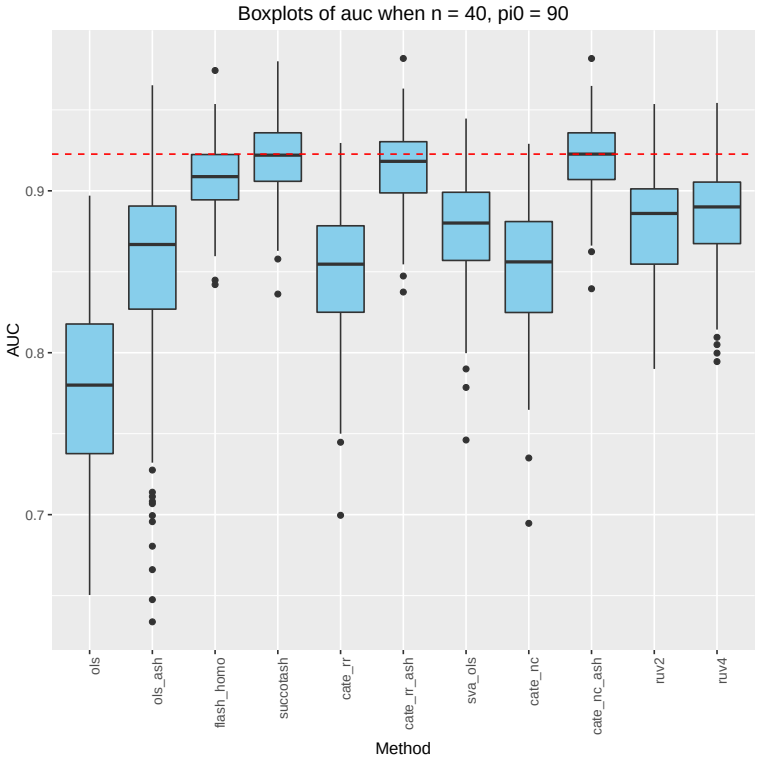

Boxplots of auc when n = 10, pi0 = 50
AUC
0.9
0.8
0.7
0.6
0.5
ols
ols_ash
flash_homo
succotash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method


Boxplots of auc when n = 10, pi0 = 90
AUC
0.9
0.8
0.7
0.6
0.5
ols
ols_ash
flash_homo
succotash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method


Boxplots of auc when n = 20, pi0 = 50
AUC
0.9
0.8
0.7
0.6
ols
ols_ash
flash_homo
succotash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method


Boxplots of auc when n = 20, pi0 = 90
AUC
0.9
0.8
0.7
0.6
0.5
ols
ols_ash
flash_homo
succotash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method


Boxplots of auc when n = 40, pi0 = 50
AUC
0.9
0.8
0.7
ols
ols_ash
flash_homo
succotash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method


Boxplots of auc when n = 40, pi0 = 90
AUC
0.9
0.8
0.7
ols
ols_ash
flash_homo
succotash
cate_rr
cate_rr_ash
sva_ols
cate_nc
cate_nc_ash
ruv2
ruv4
Method

References

Andreas Buja and Nermin Eyuboglu. Remarks on parallel analysis. *Multivariate behavioral research*, 27(4):509–540, 1992.