

Supplementary materials for ‘General model-free weighted envelope estimation’

Daniel J. Eck

Department of Statistics, University of Illinois

This supplement serves as a README for anyone that wishes to fully reproduce our simulation results or alter our code for their own analyses. The Matlab based simulations adopted from Zhang and Mai [2018] are not fully reproduced here, however the code to produce those simulations is available upon request. This document sources in output from twenty nine scripts:

1. source.R - functions used in all of our simulations.
2. logistic_simulation_A.R - first half of our first batch of logistic regression simulations.
3. logistic_simulation_A-2.R - second half of our first batch of logistic regression simulations.
4. logistic_simulation_B.R - first half of our second batch of logistic regression simulations.
5. logistic_simulation_B-2.R - second half of our second batch of logistic regression simulations.
6. poisson_simulation.R - first part of our poisson regression simulations.
7. poisson_simulation-2.R - second part of our poisson regression simulations.
8. poisson_simulation-3.R - third part of our poisson regression simulations.
9. poisson_simulation-4.R - fourth part of our poisson regression simulations.
10. logistic_simulation_p20.R - first part of our third batch of logistic regression simulations.
- 11-29. logistic_simulation_p20-j.R - j th part of our third batch of logistic regression simulations, $j = 2, \dots, 20$.

The twenty eight regression simulation scripts first load in source.R. These scripts are each then executed on a 20 core node on the UIUC campus cluster and they are run in parallel. The following software packages and scripts are required for this document to run:

```
library(tidyverse)
library(ggplot2)
library(faraway)
library(xtable)
library(parallel)
source("source.R")
```

Our numerical simulations begin with envelope estimation in logistic regression and end with envelope estimation in Poisson regression.

Numerical examples

Logistic regression simulations

Setting A: We provide a second set of logistic regression simulations for illustration. We load in the output from the second batch of logistic regression simulations.

```
load("lsimsA.RData")
load("lsimsA-2.RData")
```

The setup for this simulation is similar to the previous logistic regression simulations. The simulation configurations are again included for completeness. We first create the basis matrix Γ for the true envelope space and the basis matrix for its orthogonal complement Γ_o .

```
p <- 6; u <- 3
init <- rep(1, 6)
v1 <- matrix(init, nrow = p)
O <- qr.Q(qr(v1), complete = TRUE)
Gamma <- O[, 1:u]
Gamma0 <- O[, (u+1):p]
```

We next create the core of the material and immaterial variation, denoted as Ω and Ω_o respectively.

```
Omega <- diag(exp(c(0.25, 0.5, 0.75)))
Omega0 <- diag(exp(c(-4, -2, 1)))
```

We now build the variance matrix of the predictor variables and construct the true canonical parameter vector (regression coefficient vector) as an element contained in $\text{span}(\Gamma)$.

```
SigmaX <- Gamma%*%Omega%*%t(Gamma) + Gamma0%*%Omega0%*%t(Gamma0)
eig <- eigen(SigmaX)
SigmaX.half <- eig$vec %*% diag(sqrt(eig$val)) %*% t(eig$vec)
beta <- (-0.5*Gamma[, 1] - 0.5*Gamma[, 2] - 1.5*Gamma[, 3]) / 2
matrix(beta, ncol = 1)
```

```
##           [,1]
## [1,]  0.51031036
## [2,] -0.02958759
## [3,] -0.52958759
## [4,]  0.22041241
## [5,]  0.22041241
## [6,]  0.22041241
```

We now describe the nonparametric bootstrap procedure for all envelope estimators of the canonical parameter vector mentioned in the main text and computed in the accompanying R scripts. Our bootstrap simulation will consider two model selection regimes for determining the envelope

Quantity	Description
$\hat{\theta}_w^{\text{FG}*}$ or $\hat{\theta}_w^{\text{ID}*}$	The weighted envelope estimator $\hat{\theta}_w^{\text{FG}}$ or $\hat{\theta}_w^{\text{ID}}$ computed with respect to resampled data.
$\hat{\theta}_k^{\text{FG}*}$ or $\hat{\theta}_k^{\text{ID}*}$	The envelope estimator $\hat{\theta}_k^{\text{FG}}$ or $\hat{\theta}_k^{\text{ID}}$ fit at dimension k computed with respect to resampled data.
$\hat{\theta}_{\hat{u}_{\text{FG}}}^{\text{FG}*}$ or $\hat{\theta}_{\hat{u}_{\text{ID}}}^{\text{ID}*}$	The envelope estimator $\hat{\theta}_{\hat{u}_{\text{FG}}}^{\text{FG}}$ or $\hat{\theta}_{\hat{u}_{\text{ID}}}^{\text{ID}}$ fit at the dimension estimated from the original sample \hat{u}_{FG} or \hat{u}_{ID} and is then computed with respect to resampled data. We call this the fixed u regime.
$\hat{\theta}_{\hat{u}_{\text{FG}}^*}^{\text{FG}*}$ or $\hat{\theta}_{\hat{u}_{\text{ID}}^*}^{\text{ID}*}$	The envelope estimator $\hat{\theta}_{\hat{u}_{\text{FG}}^*}^{\text{FG}}$ or $\hat{\theta}_{\hat{u}_{\text{ID}}^*}^{\text{ID}}$ fit at the dimension estimated from resampled data \hat{u}_{FG}^* or \hat{u}_{ID}^* and is then also computed with respect to resampled data. We call this the variable u regime. This estimator is similar to the consistent model free estimators developed by Zhang and Mai [2018].
$r(\hat{\theta}_1, \hat{\theta}_2)$	A column vector containing ratios of bootstrapped standard errors $\text{se}^*(\hat{\theta}_{1,j})/\text{se}^*(\hat{\theta}_{2,j})$, $j = 1, \dots, p$.

Table 1: Description of bootstrap estimators.

dimension \hat{u}_{1D} at every iteration. In one scheme, we estimate the envelope dimension at every iteration of the bootstrap (variable u). In the other scheme, we estimate the dimension of the envelope space in the original data set and then treat this estimated dimension as the true dimension when we resample our data and calculate these envelope estimators (fixed u), thus ignoring the variability associated with model selection. Theorem 3 in the main text provides some guidance for the performance of the nonparametric bootstrap for estimating the variability of $\hat{\theta}_w^{\text{ID}}$, an analog does not exist for the other envelope estimators.

The `sim_function` function (included in the source.R file) generates a logistic regression model that incorporates the above envelope structure that is stored in your global environment. The function performs a nonparametric bootstrap with respect to all considered estimators that can be viewed in Table 1. This function also returns the estimated envelope dimension at every iteration of the nonparametric bootstrap.

We now combine the two halves from our Monte Carlo simulations to estimate ratios, coverage probabilities, and distribution of the estimated envelope dimension.

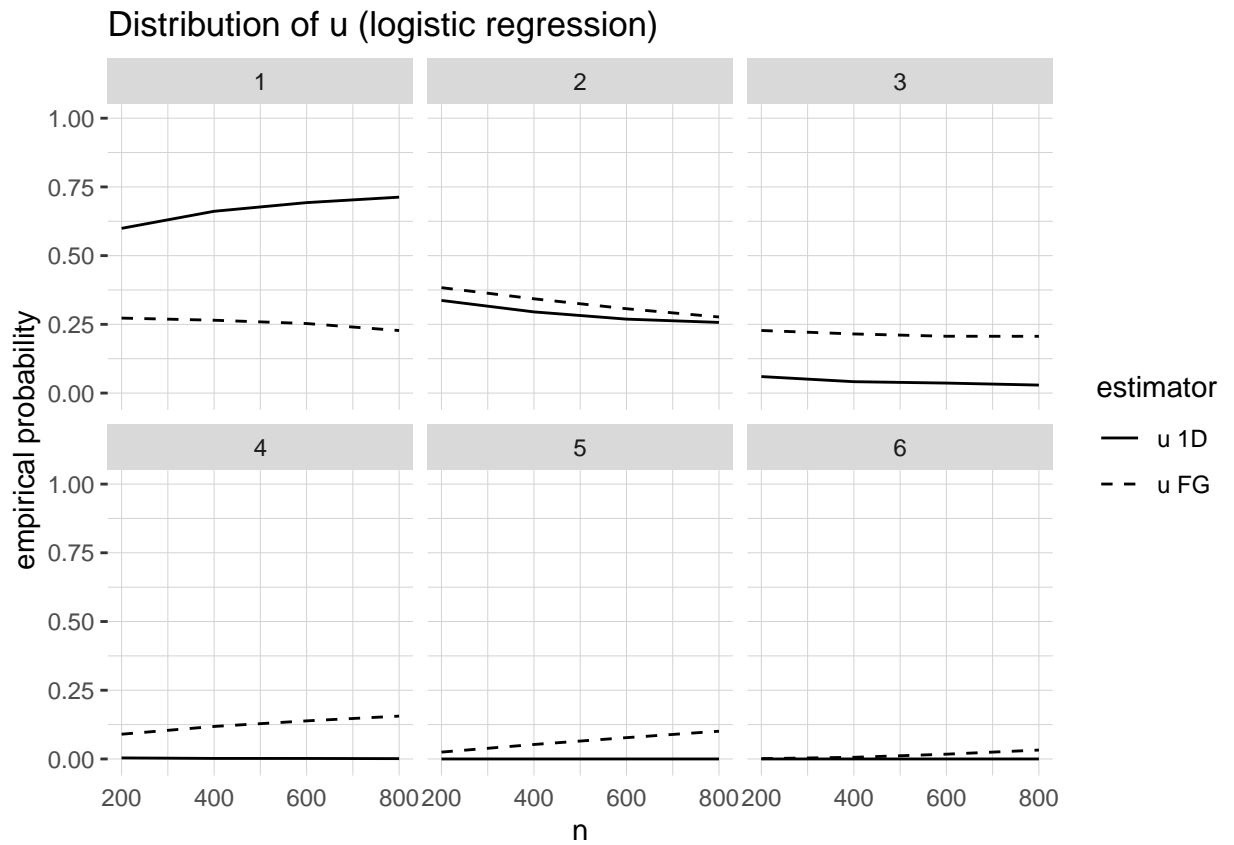
```
for(j in 1:4) lsims_MC_A[[j]] <-
  rbind(lsims_MC_A[[j]], lsims_MC_A_2[[j]])
```

Important sample sizes are given below.

```
nMC <- 400 # Monte Carlo sample size
ns <- c(200, 400, 600, 800) # sample sizes
nboot <- 1e3 # bootstrap sample size
```

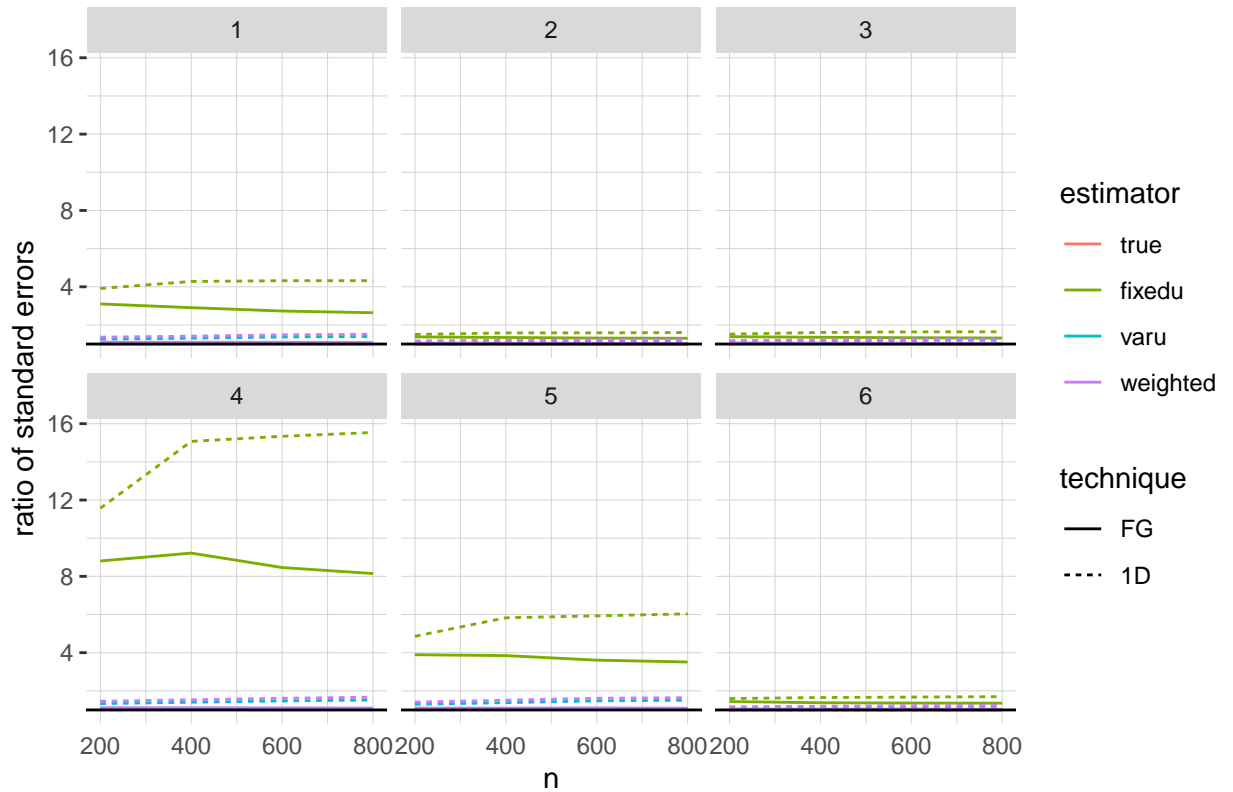
The distribution of u across bootstrap and Monte Carlo iterates is presented below. Code is

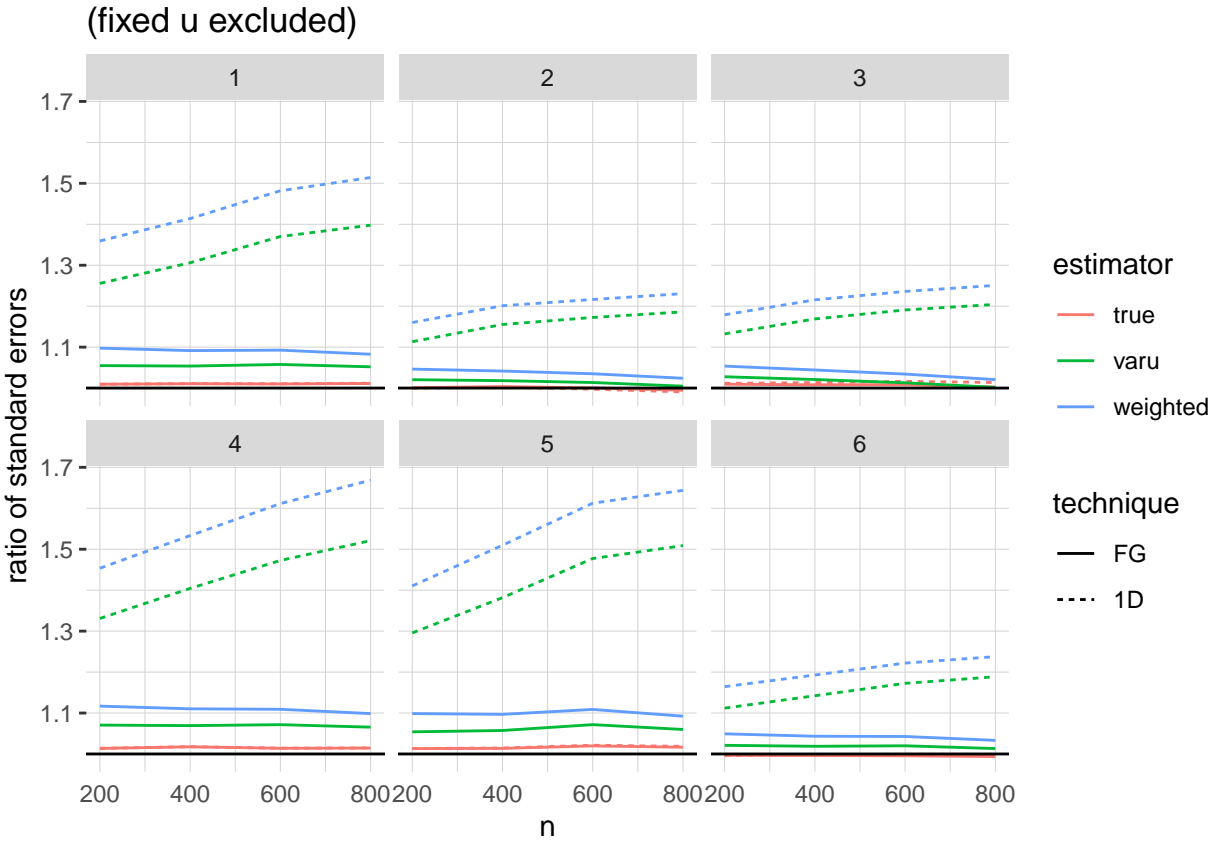
included in the accompanying .Rmd file.



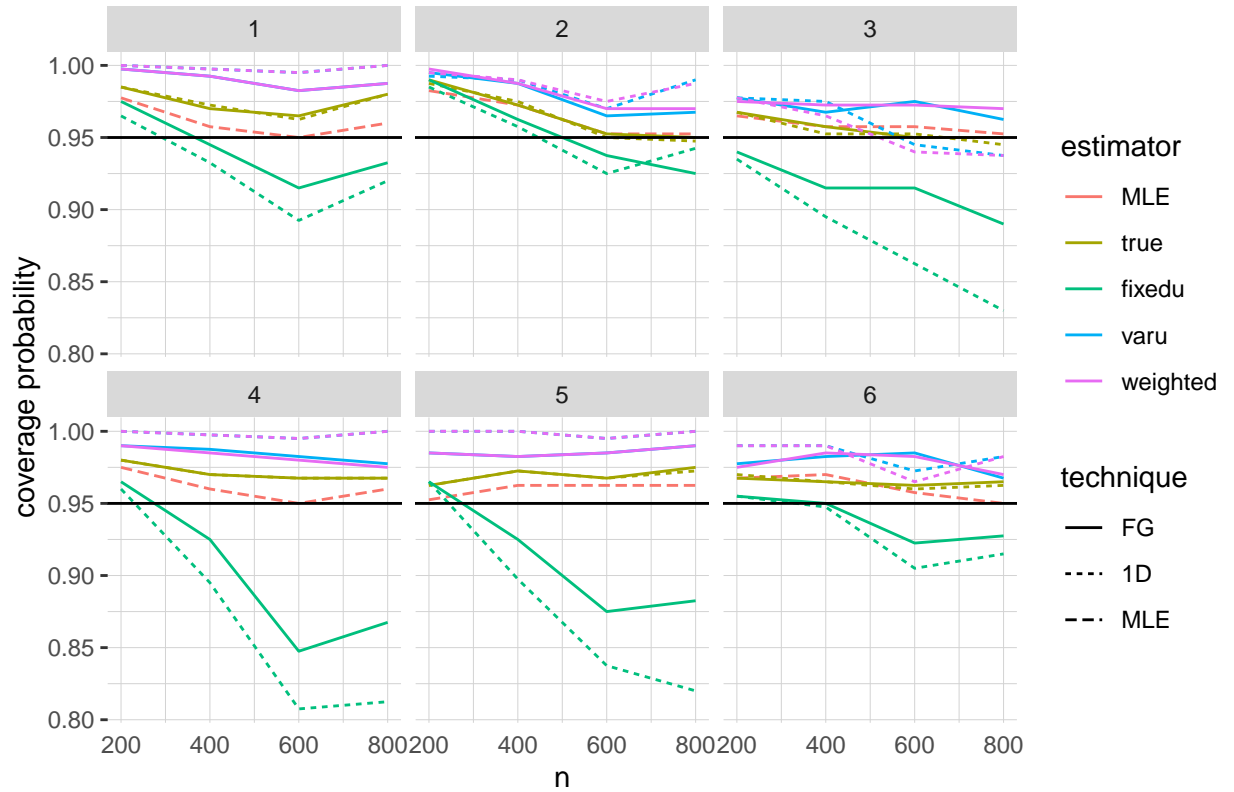
Results for ratios of bootstrap standard errors relative to the MLE are depicted here and on the next page.

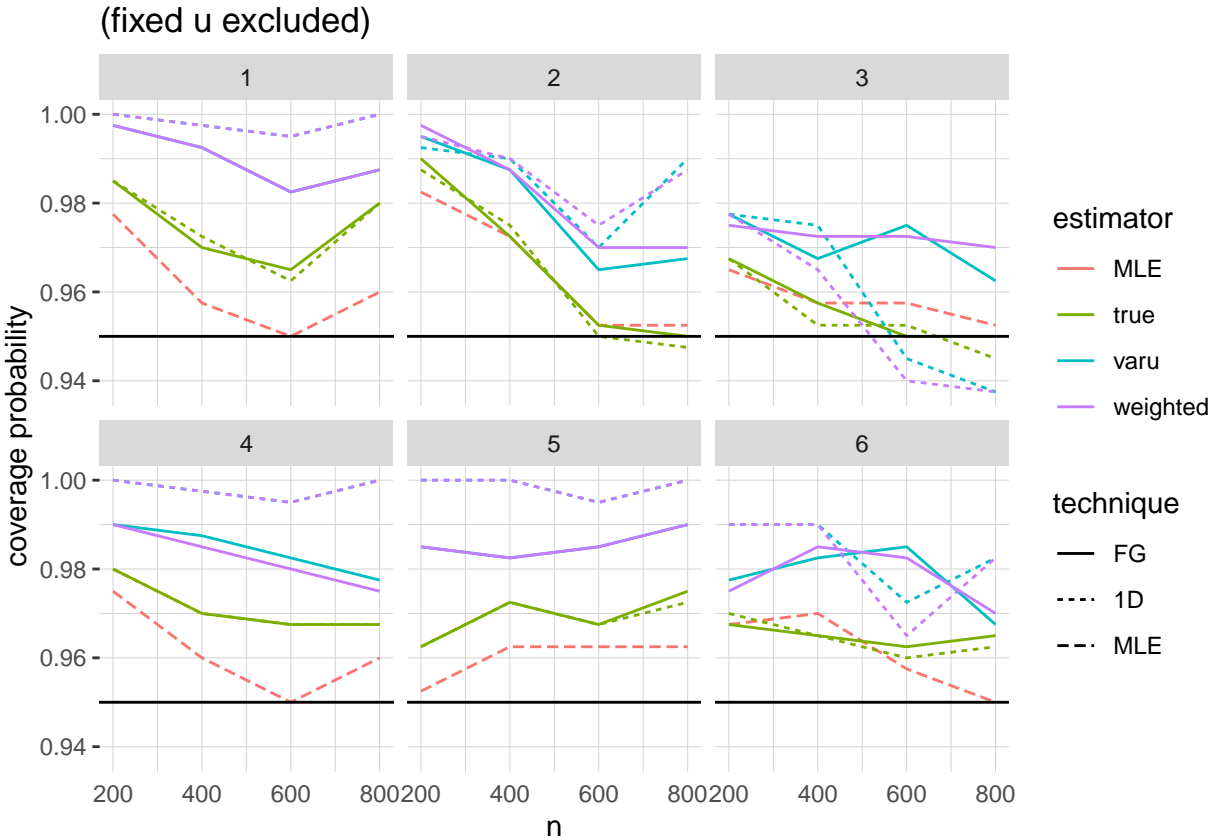
Monte Carlo simulation of ratios (logistic regression)





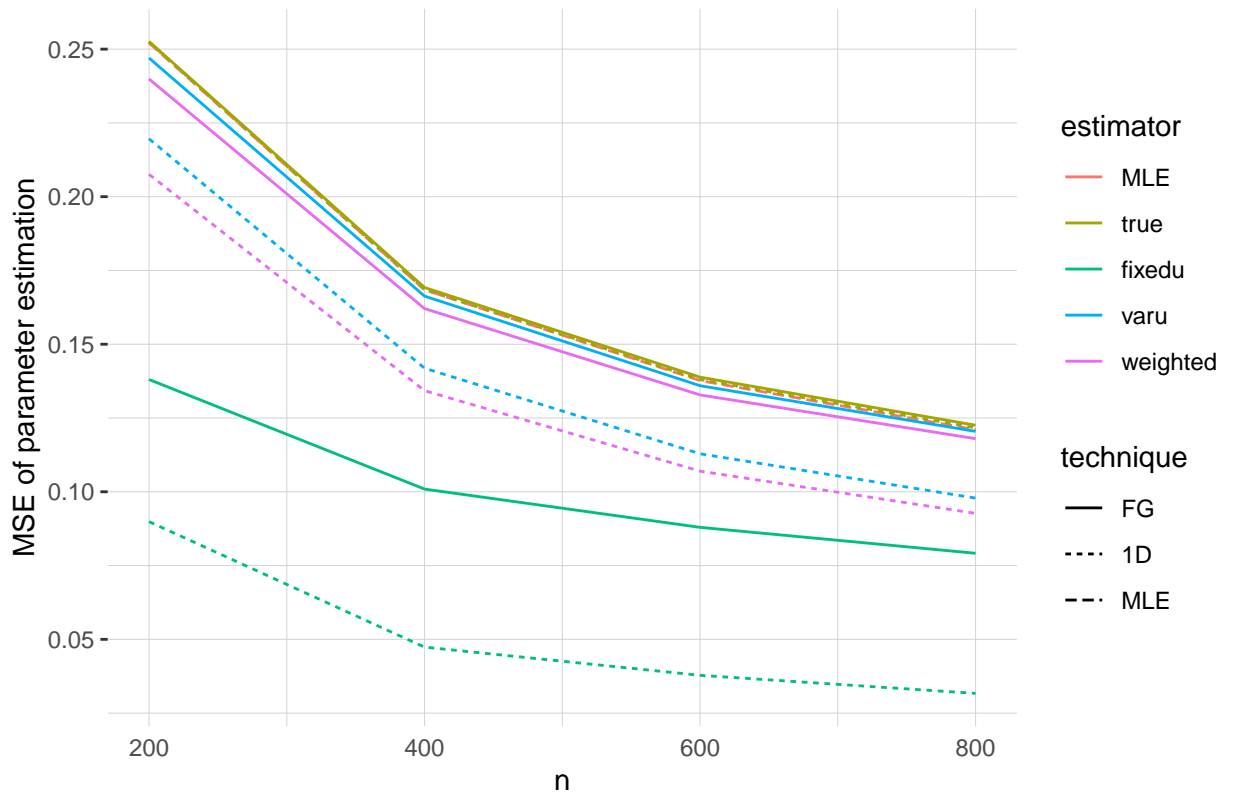
Results for univariate coverage probabilities are depicted here and on the next page.
Monte Carlo simulation of coverage probabilities (logistic regression)

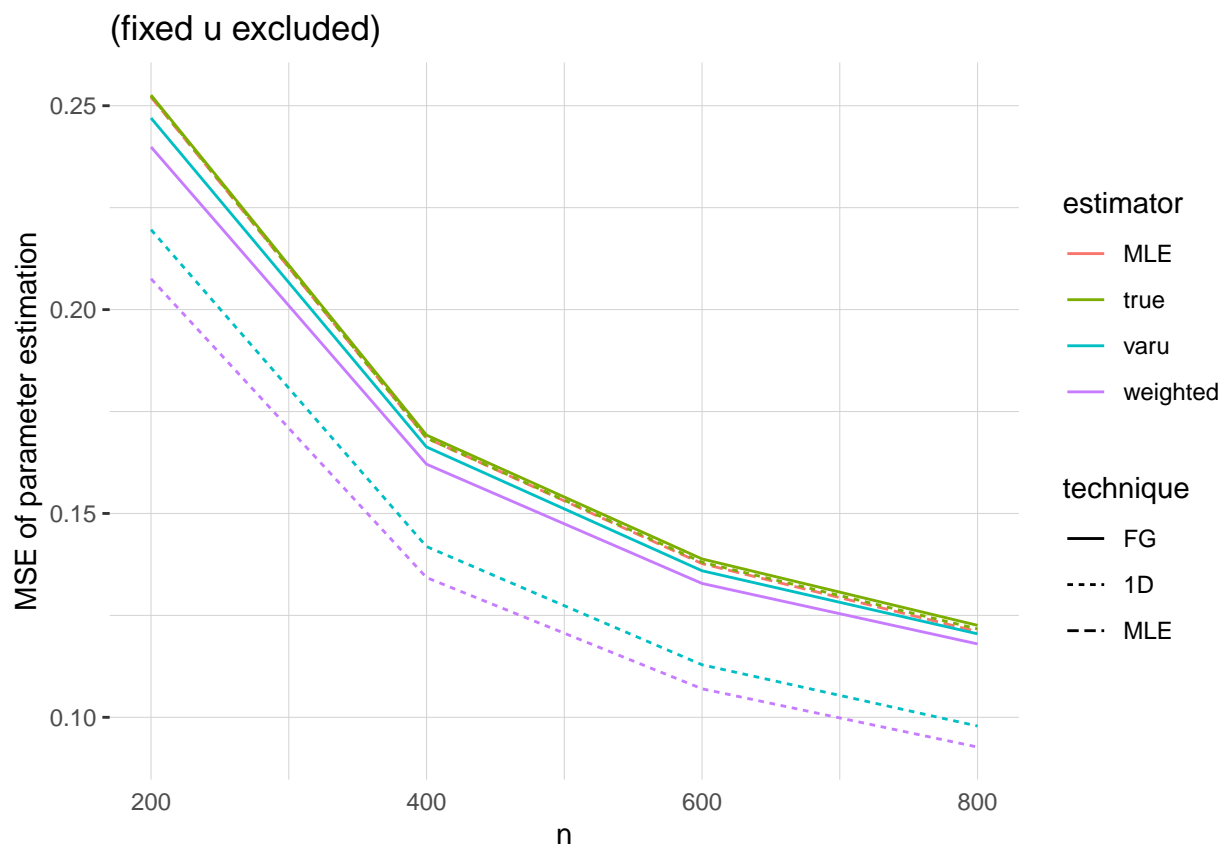




Results for the MSE of the averaged over Monte Carlo samples of are depicted here and on the next page.

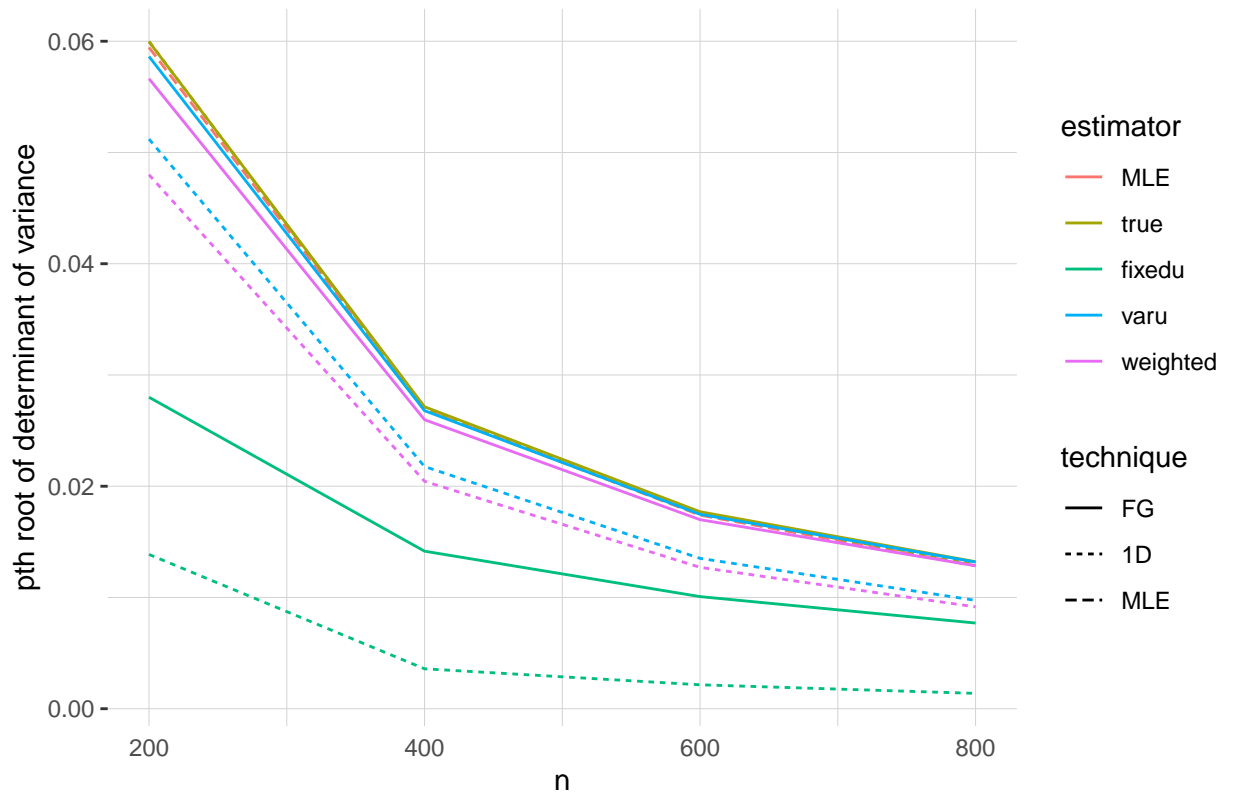
Monte Carlo simulation of MSE

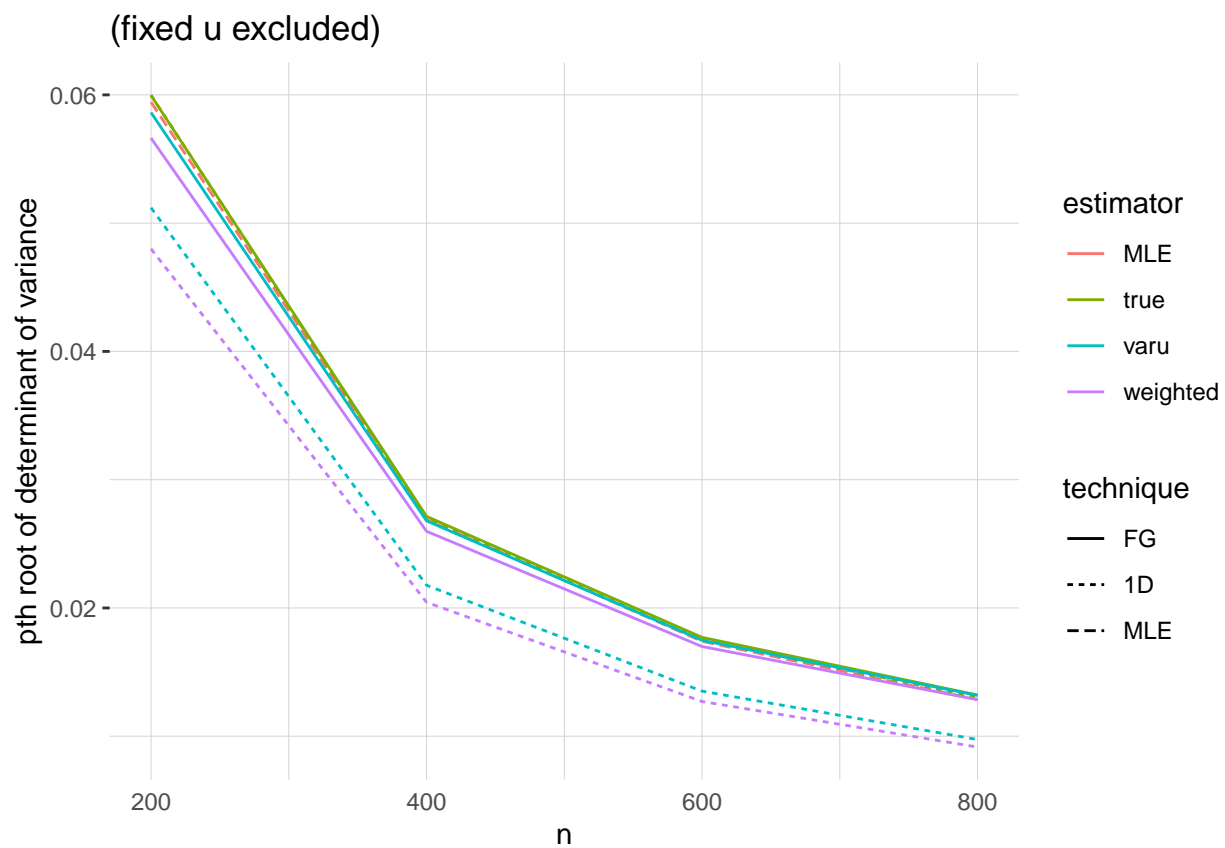




Results for the p th root determinants of the bootstrapped variances averaged over Monte Carlo samples of are depicted here and on the next page.

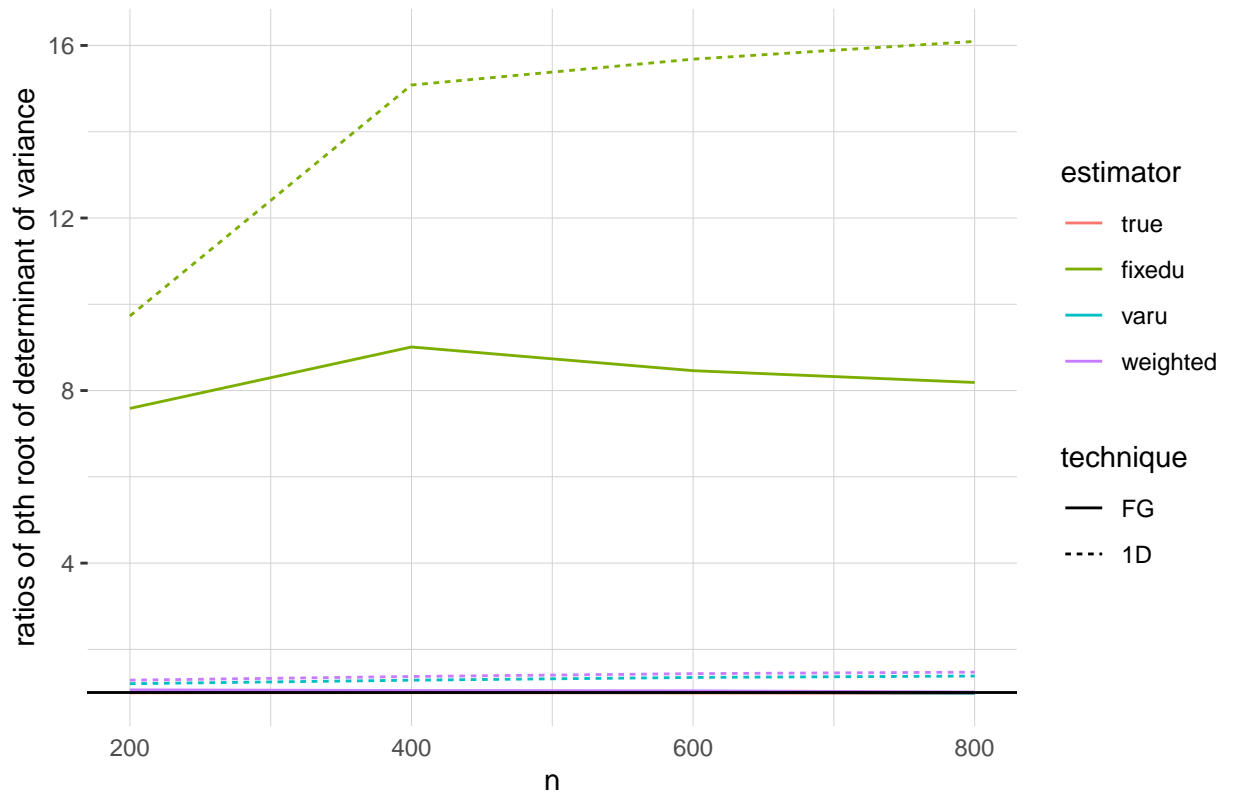
Monte Carlo simulation of variances (logistic regression)

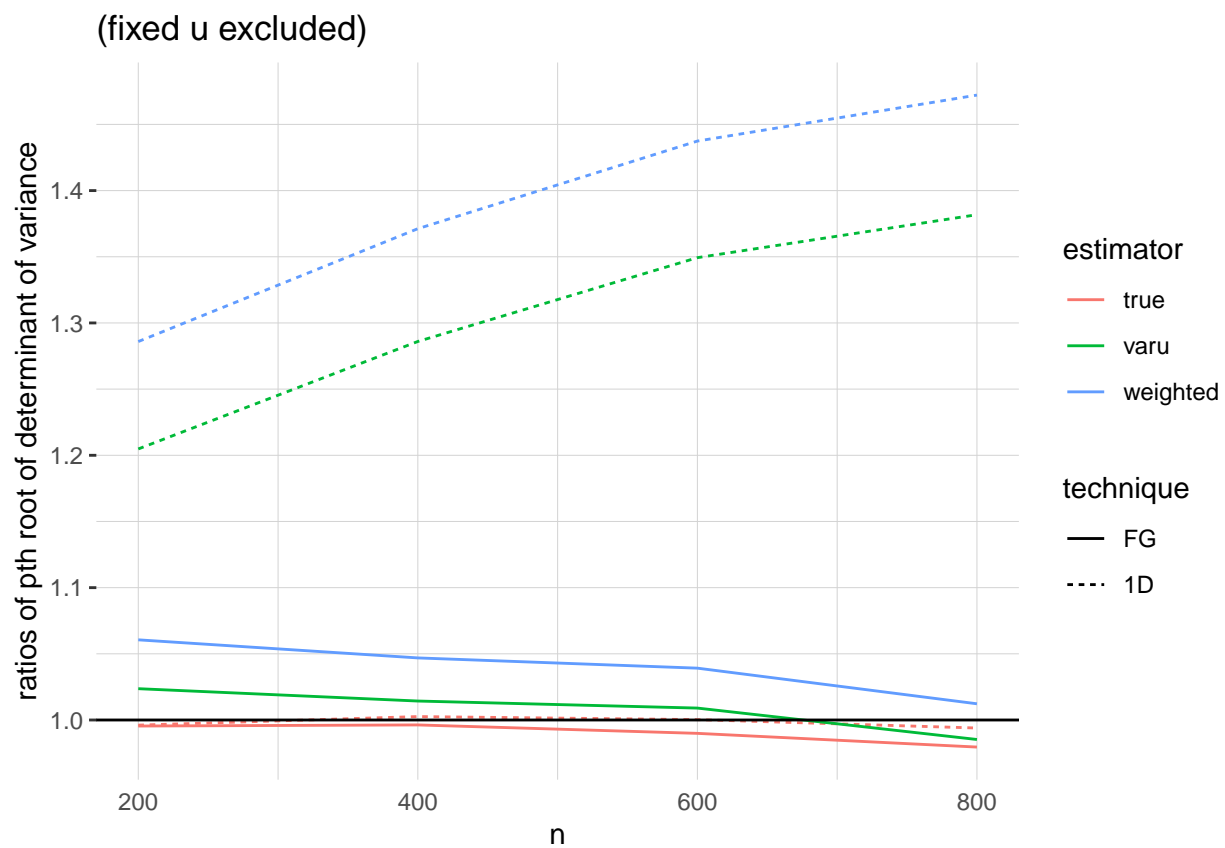




Results for the ratios of the p th root determinants of the bootstrapped variances relative to the MLE averaged over Monte Carlo samples of are depicted here and on the next page.

Monte Carlo simulation of variances (logistic regression)





Setting B We load in the output from the first batch of logistic regression simulations.

```
load("lsimsB.RData")
load("lsimsB-2.RData")
```

The simulation configurations are included here for completeness. First we construction the basis matrix for the envelope space.

```
p <- 6; u <- 3
init <- rep(1, 6)
v1 <- matrix(init, nrow = p)
O <- qr.Q(qr(v1), complete = TRUE)
Gamma <- O[, 1:u]
Gamma0 <- O[, (u+1):p]
```

We next create the core of the material and immaterial variation, denoted as Ω and Ω_o respectively.

```
Omega <- diag(c(0.5, 1, 1.5))
Omega0 <- diag(exp(c(-3, -2, 1)))
```

We now build the variance matrix of the predictor variables and construct the true canonical parameter vector (regression coefficient vector) as an element contained in $\text{span}(\Gamma)$.

```
SigmaX <- Gamma%*%Omega%*%t(Gamma) + Gamma0%*%Omega0%*%t(Gamma0)
eig <- eigen(SigmaX)
SigmaX.half <- eig$vec %*% diag(sqrt(eig$val)) %*% t(eig$vec)
beta <- -(0.5*Gamma[, 1] - 2*Gamma[, 2] - 2*Gamma[, 3]) / 3
as.matrix(beta, ncol = 1)

##           [,1]
## [1,] -0.47628967
## [2,]  0.57690759
## [3,]  0.57690759
## [4,] -0.08975907
## [5,] -0.08975907
## [6,] -0.08975907
```

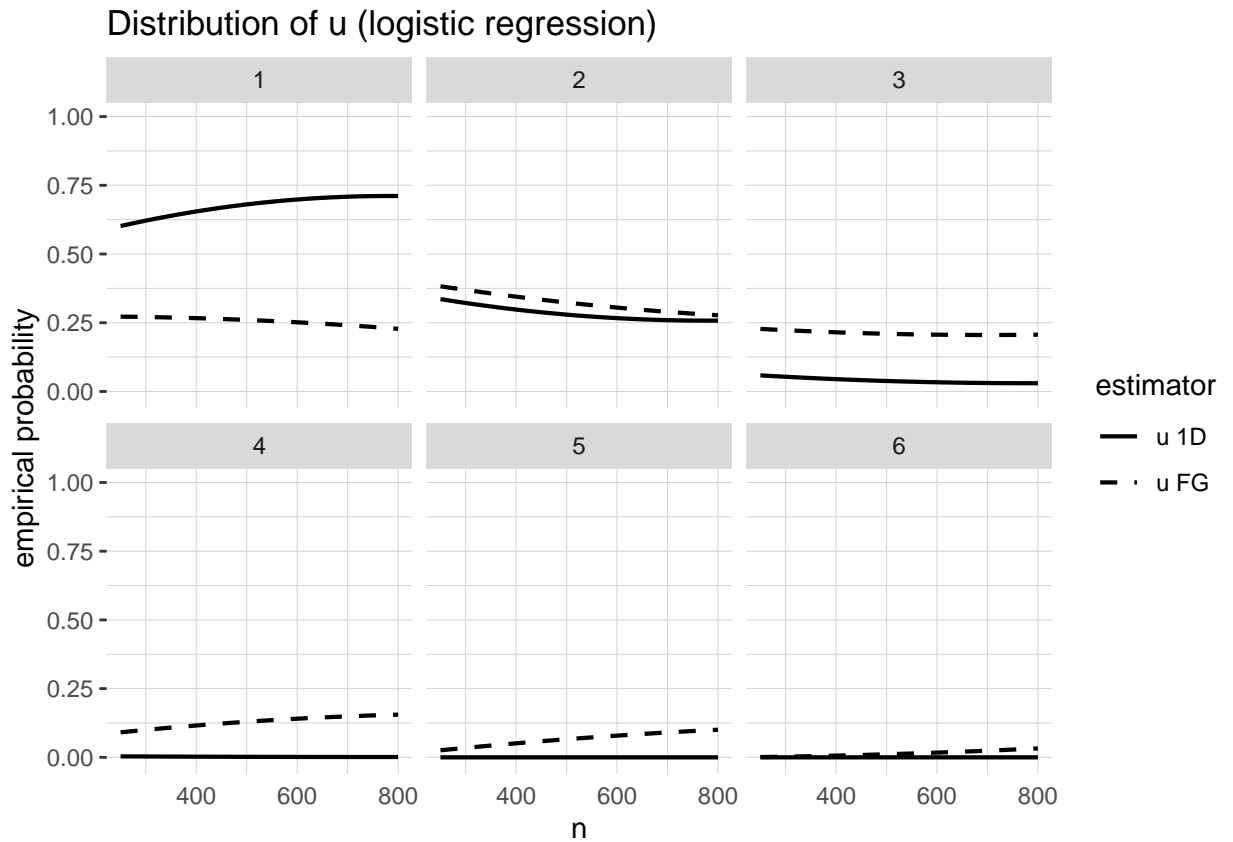
We now combine the two halves from our Monte Carlo simulations to estimate ratios, coverage probabilities, and distribution of the estimated envelope dimension.

```
for(j in 1:4) lsims_MC_B[[j]] <-
  rbind(lsims_MC_B[[j]], lsims_MC_B_2[[j]])
```

Important sample sizes are given below.

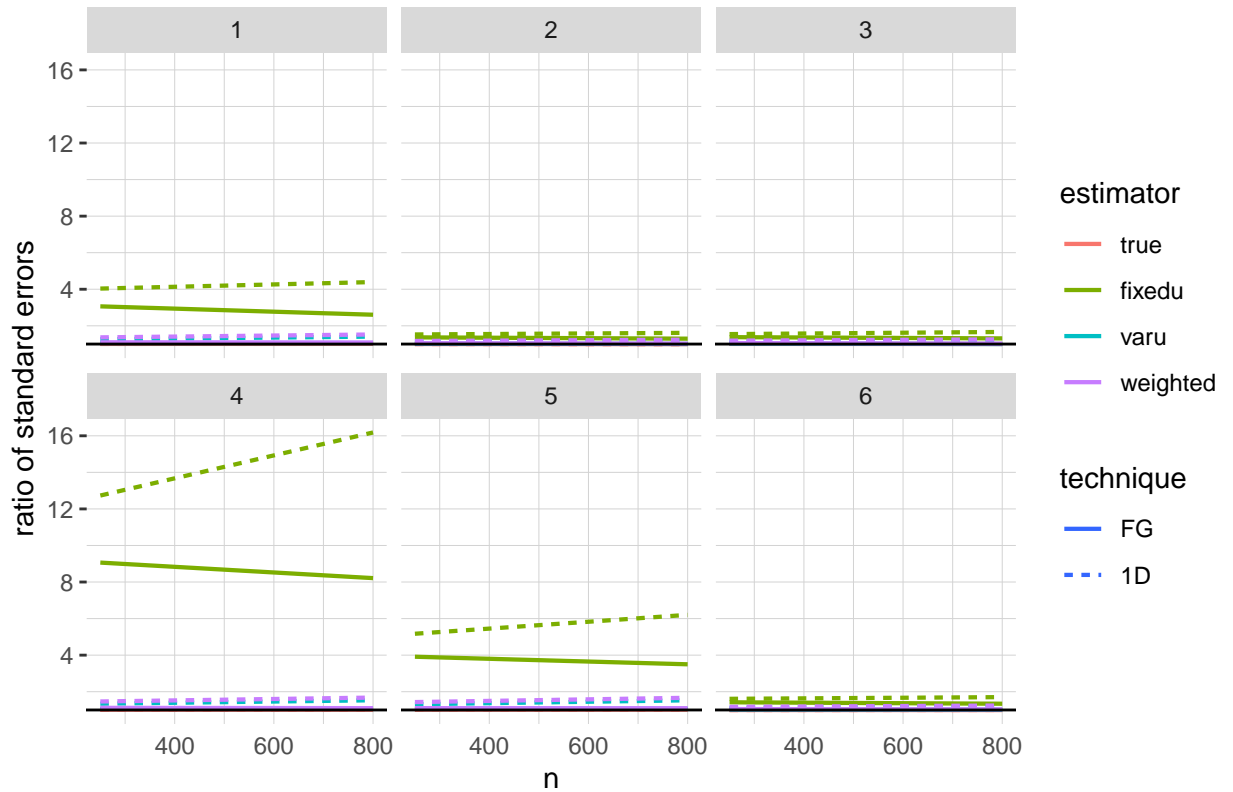
```
nMC <- 400 # Monte Carlo sample size
ns <- c(250, 400, 600, 800) # sample sizes
nboot <- 1e3 # bootstrap sample size
```

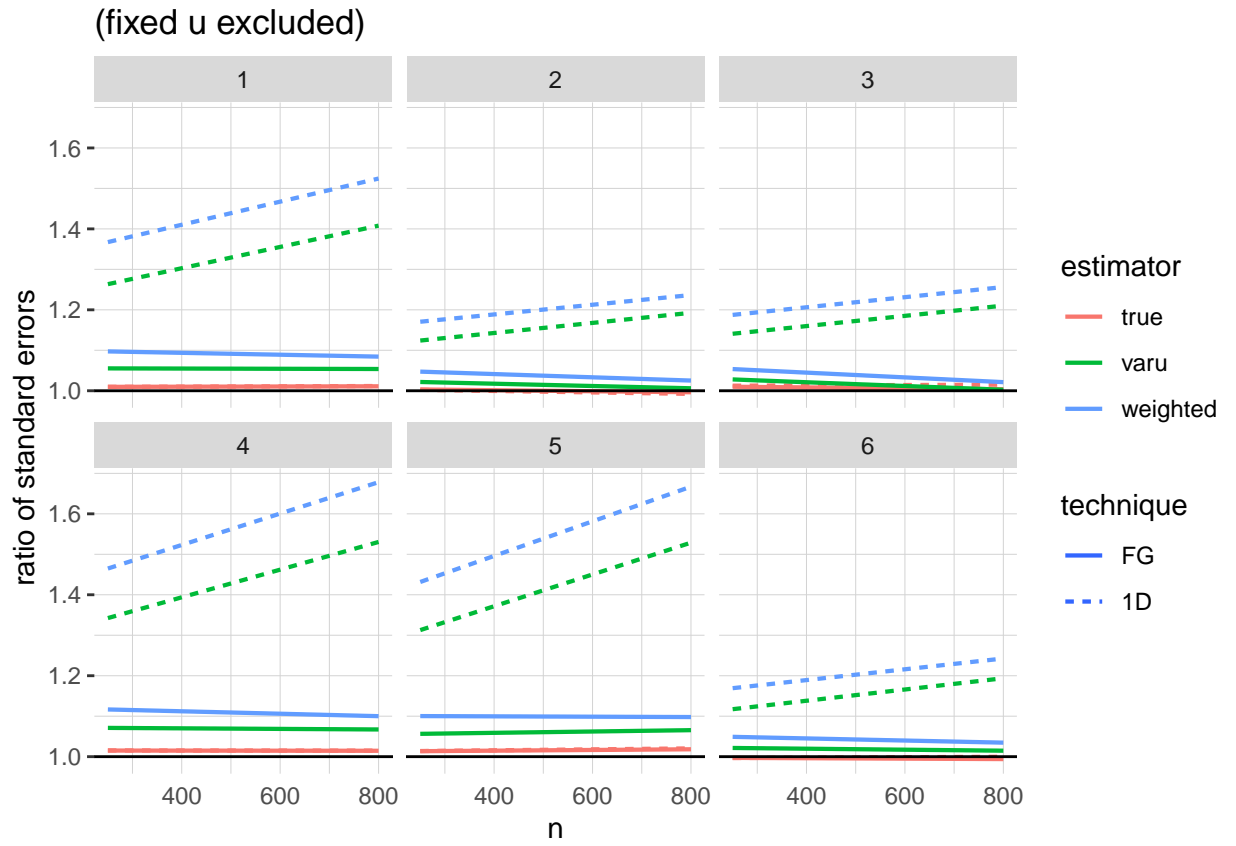
The distribution of u across bootstrap and Monte Carlo iterates is presented below. Code is included in the accompanying .Rmd file.



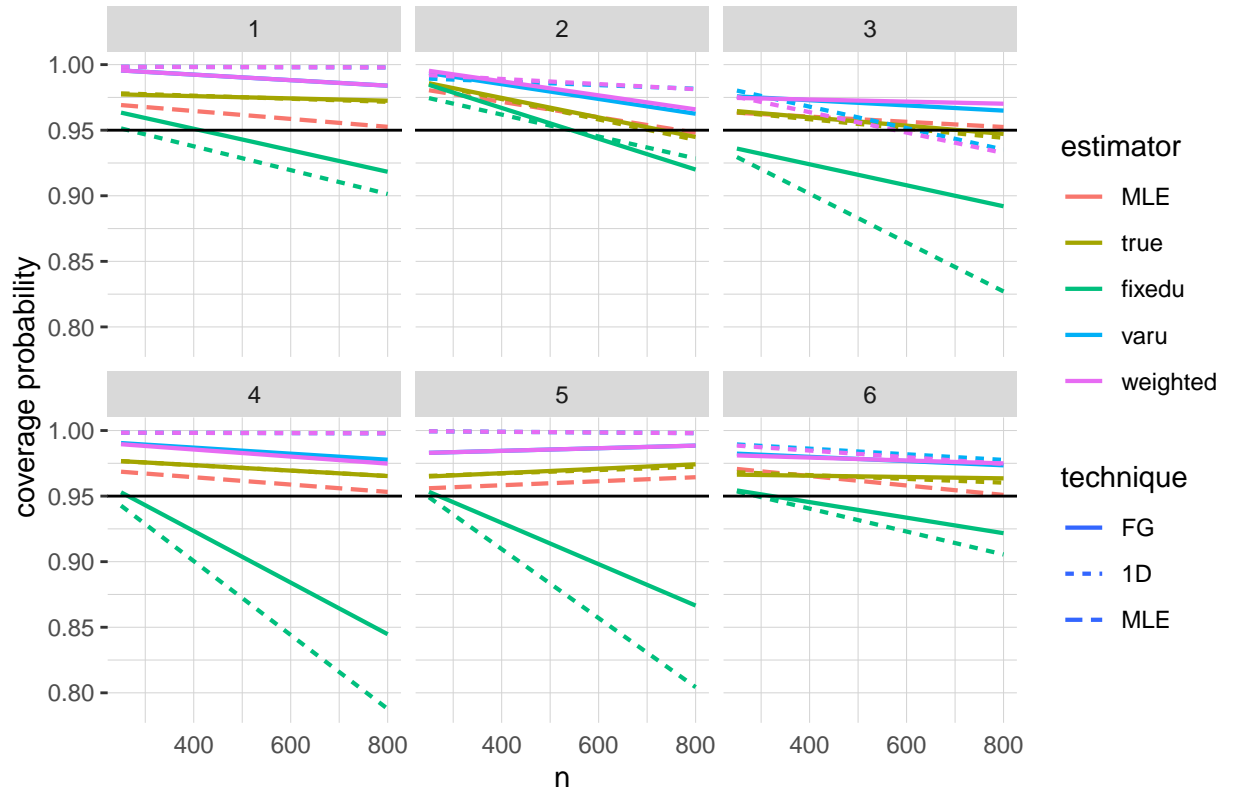
Results for ratios of bootstrap standard errors relative to the MLE are depicted here and on the next page.

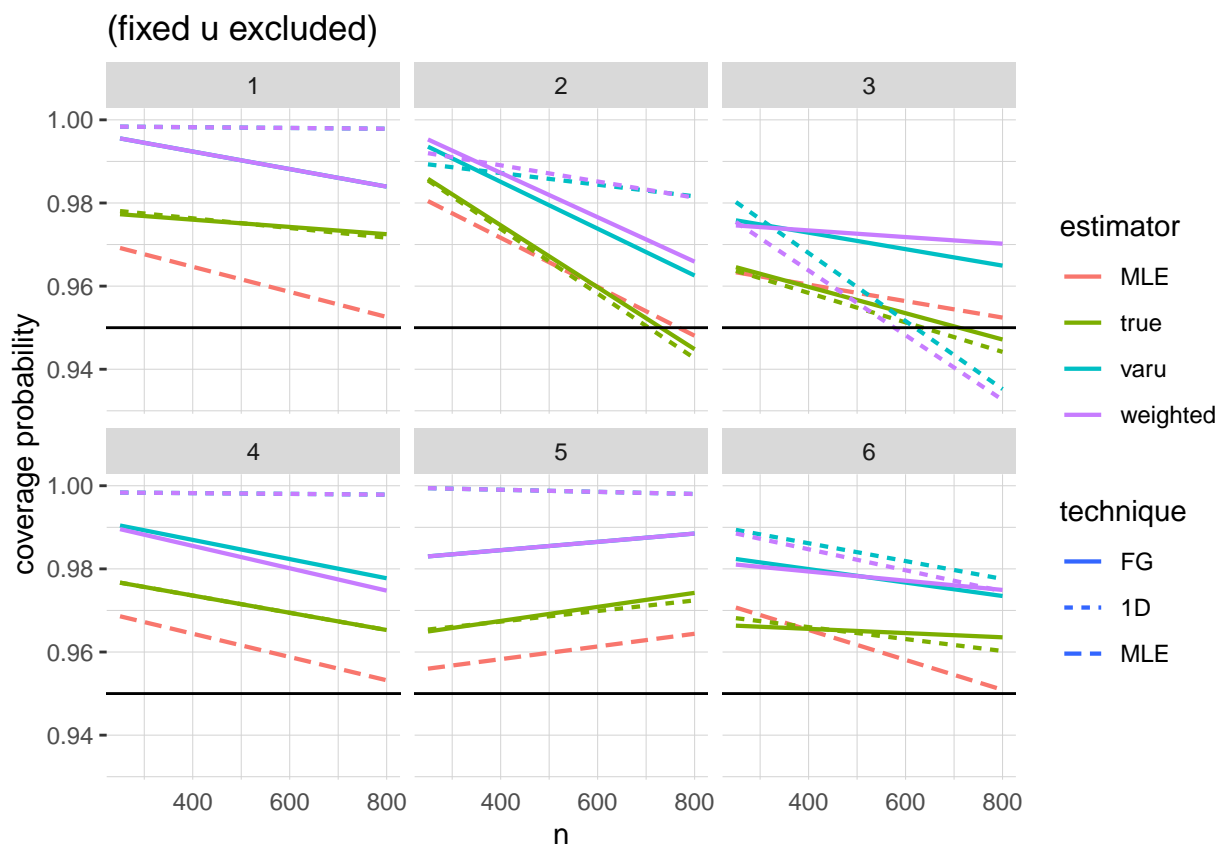
Monte Carlo simulation of ratios (logistic regression)





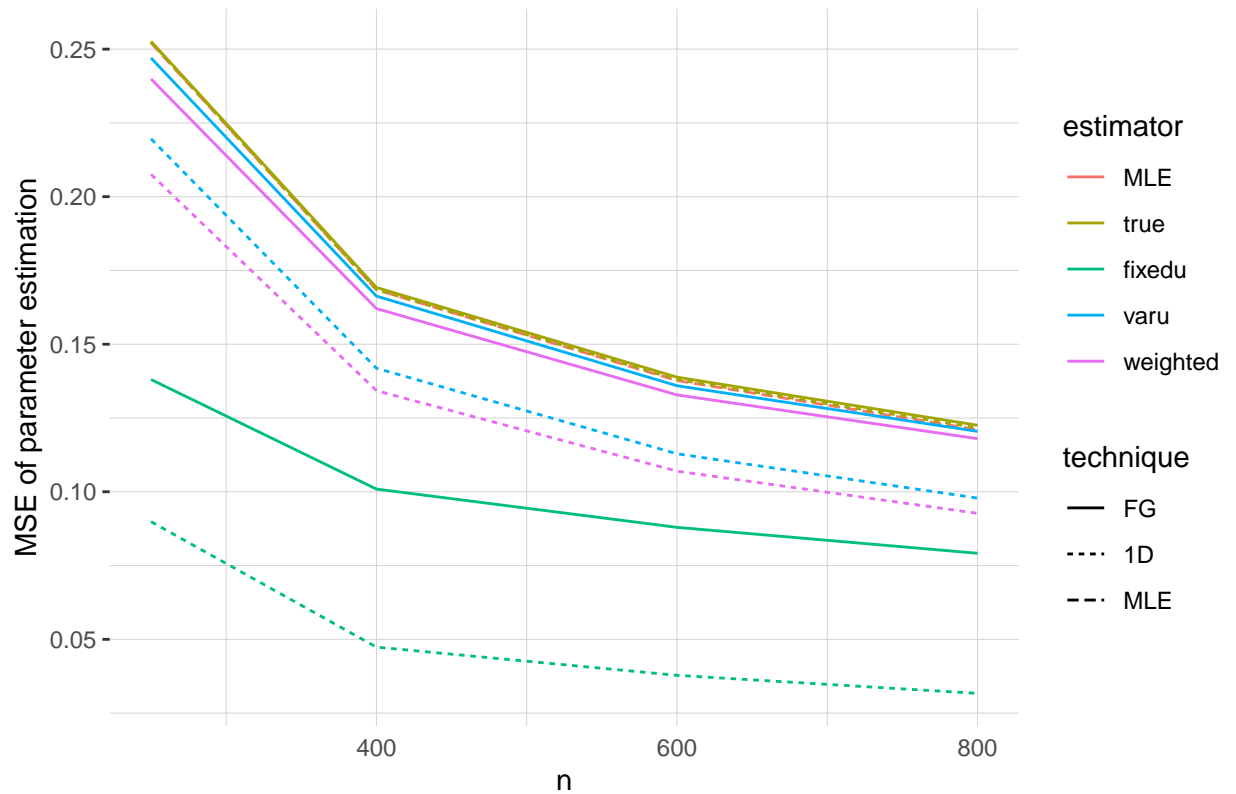
Results for univariate coverage probabilities are depicted here and on the next page.
Monte Carlo simulation of coverage probabilities (logistic regression)

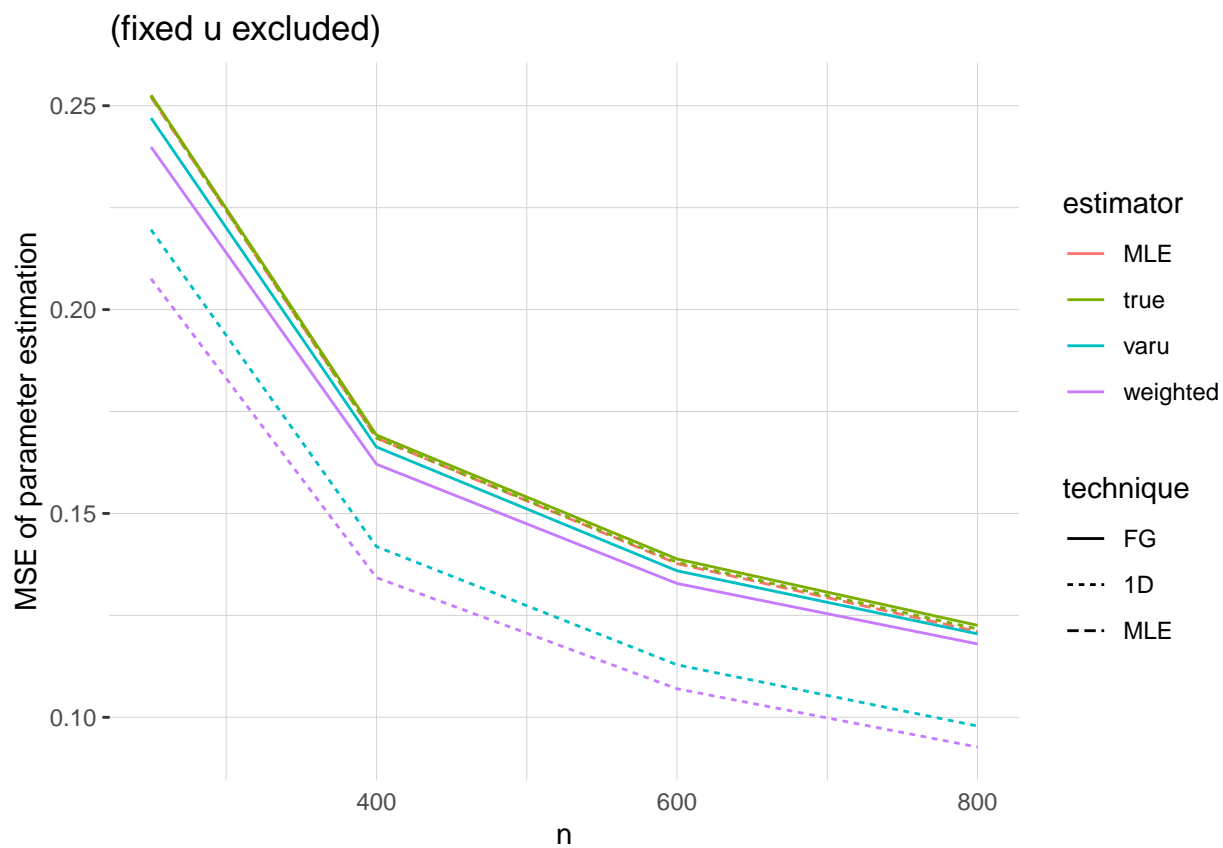




Results for the MSE of the averaged over Monte Carlo samples of are depicted here and on the next page.

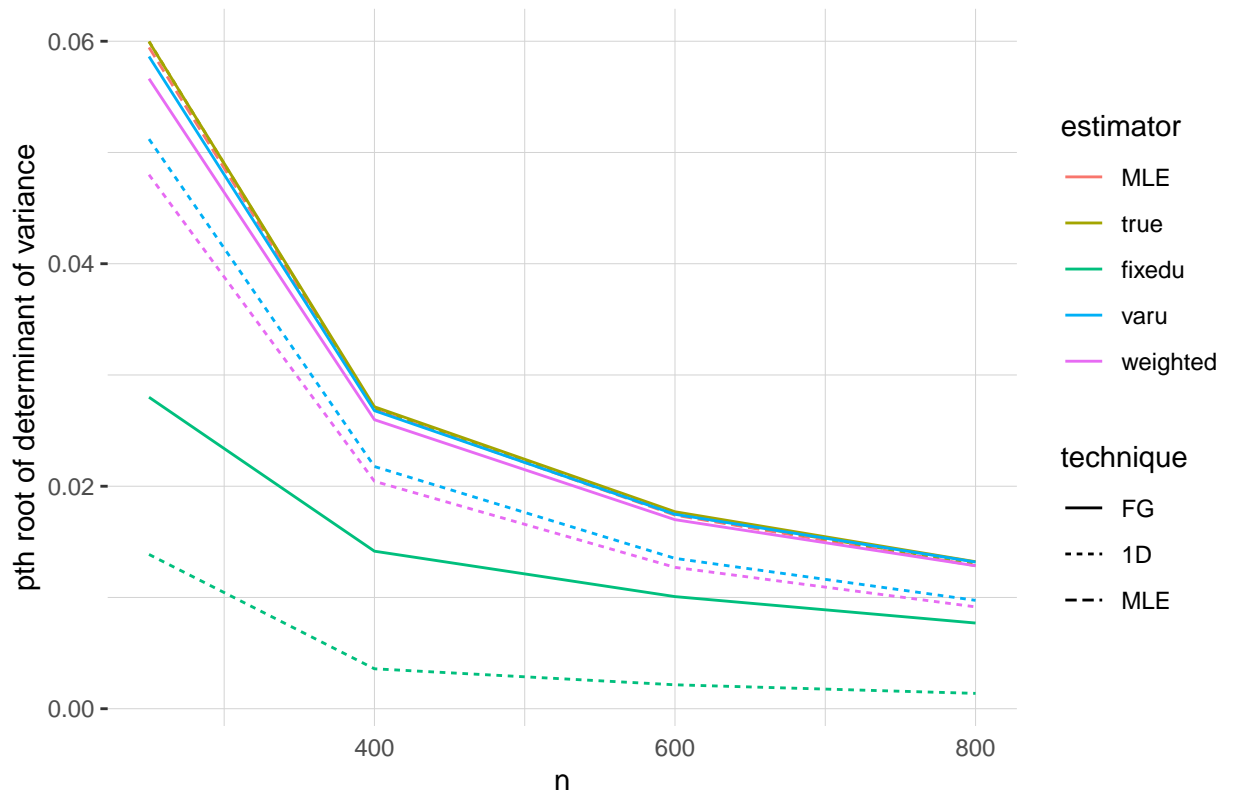
Monte Carlo simulation of MSE

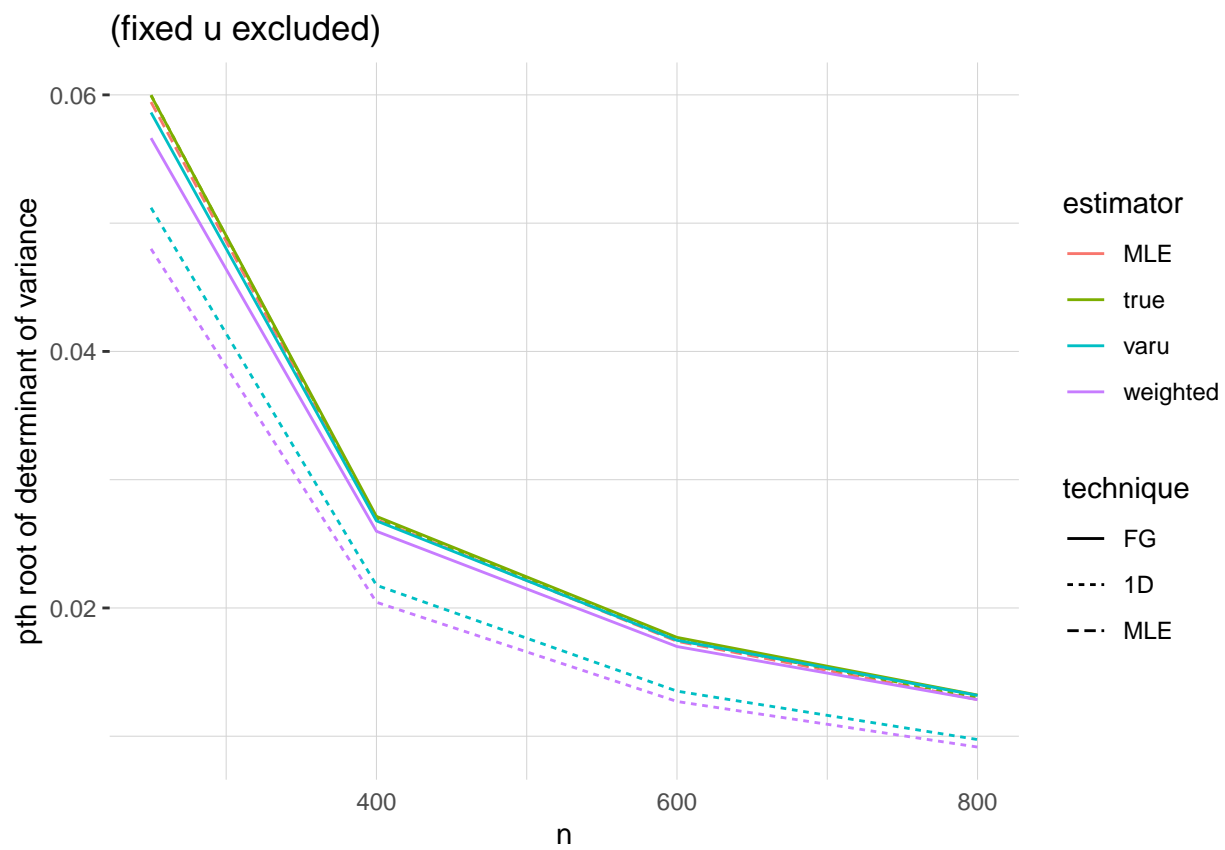




Results for the p th root determinants of the bootstrapped variances averaged over Monte Carlo samples of are depicted here and on the next page.

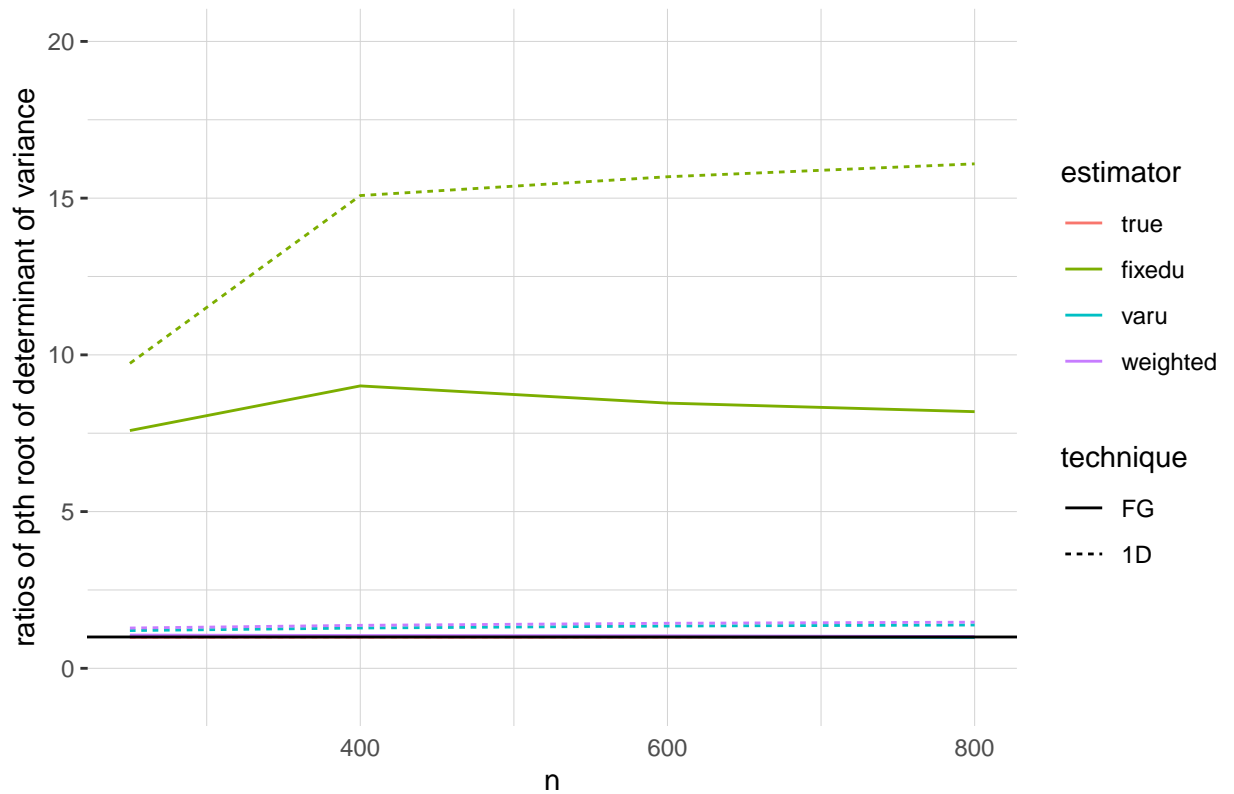
Monte Carlo simulation of variances (logistic regression)

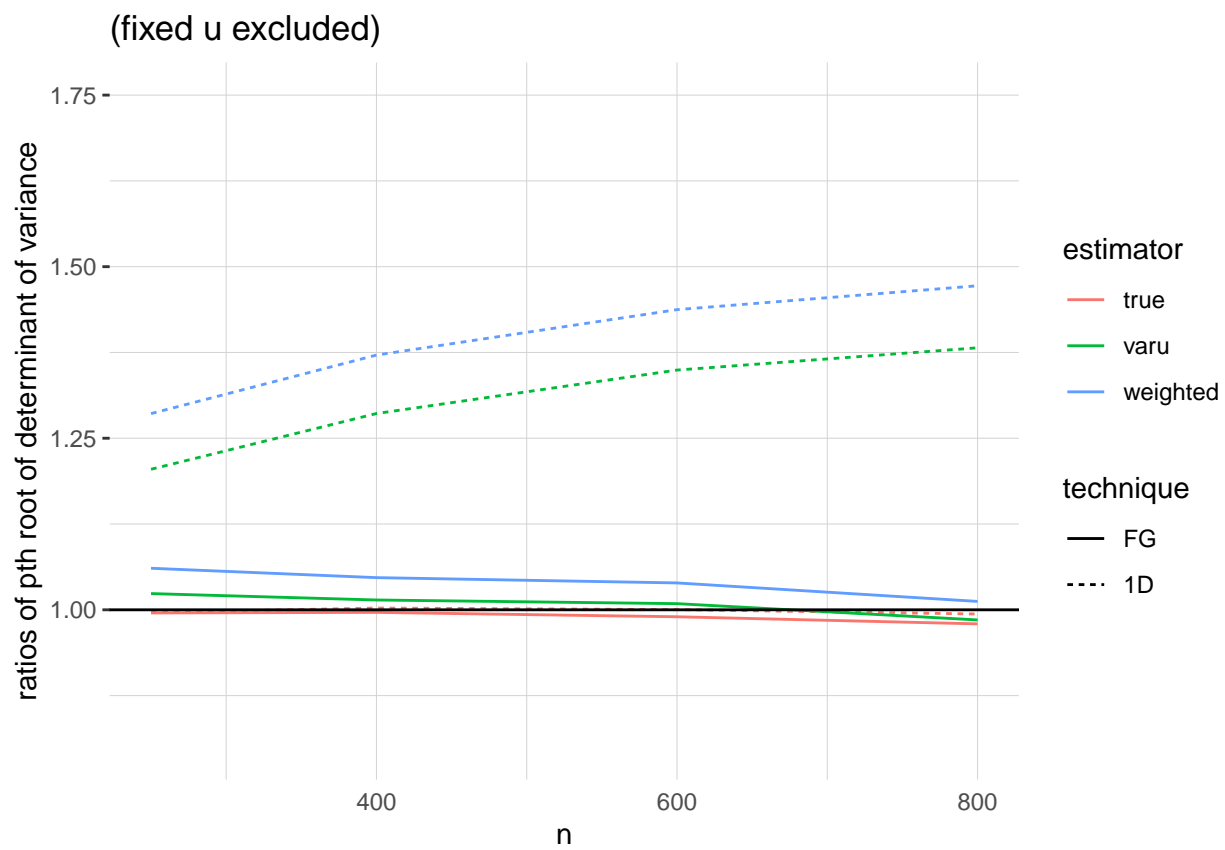




Results for the ratios of the p th root determinants of the bootstrapped variances relative to the MLE averaged over Monte Carlo samples of are depicted here and on the next page.

Monte Carlo simulation of variances (logistic regression)





Setting C ($p = 15$): We provide a second set of logistic regression simulations for illustration. We load in the output from the second batch of logistic regression simulations.

```
load("lsims20.RData")
load("lsims20-2.RData")
load("lsims20-3.RData")
load("lsims20-4.RData")
load("lsims20-5.RData")
load("lsims20-6.RData")
load("lsims20-7.RData")
load("lsims20-8.RData")
load("lsims20-9.RData")
load("lsims20-10.RData")
load("lsims20-11.RData")
load("lsims20-12.RData")
load("lsims20-13.RData")
load("lsims20-14.RData")
load("lsims20-15.RData")
load("lsims20-16.RData")
load("lsims20-17.RData")
load("lsims20-18.RData")
load("lsims20-19.RData")
load("lsims20-20.RData")
```

The setup for this simulation is similar to the previous logistic regression simulations. One of the twenty simulation configurations is included for completeness (the others just vary in which seed was set). We first create the basis matrix Γ for the true envelope space, the basis matrix for its orthogonal complement Γ_o , and β

```
p <- 15; u <- 5
foo <- matrix(0, nrow = p, ncol = u)
for(j in 1:u){
  foo[(2*(j-1)+1):(2*j), j] <- 1
}
O <- qr.Q(qr(foo), complete = TRUE)
Gamma <- O[, 1:u]
Gamma0 <- O[, (u+1):p]
beta <- rowSums(Gamma) / 7
as.matrix(beta, ncol = 1)

##           [,1]
## [1,] -0.1010153
## [2,] -0.1010153
```

```
## [3,] -0.1010153
## [4,] -0.1010153
## [5,] -0.1010153
## [6,] -0.1010153
## [7,] -0.1010153
## [8,] -0.1010153
## [9,] -0.1010153
## [10,] -0.1010153
## [11,] 0.0000000
## [12,] 0.0000000
## [13,] 0.0000000
## [14,] 0.0000000
## [15,] 0.0000000
```

We next create the core of the material and immaterial variation, denoted as Ω and Ω_o respectively, and we construct the square root of the variance matrix for the predictors.

```
Omega <- diag( 2 * 1:5 )
Omega0 <- diag(exp( seq(from = -7, to = -1, length = 10) ))
SigmaX <- Gamma%*%Omega%*%t(Gamma) + Gamma0%*%Omega0%*%t(Gamma0)
eig <- eigen(SigmaX)
SigmaX.half <- eig$vec %*% diag(sqrt(eig$val)) %*% t(eig$vec)
```

We now combine the 20 pieces from our Monte Carlo simulations to estimate ratios, coverage probabilities, and distribution of the estimated envelope dimension.

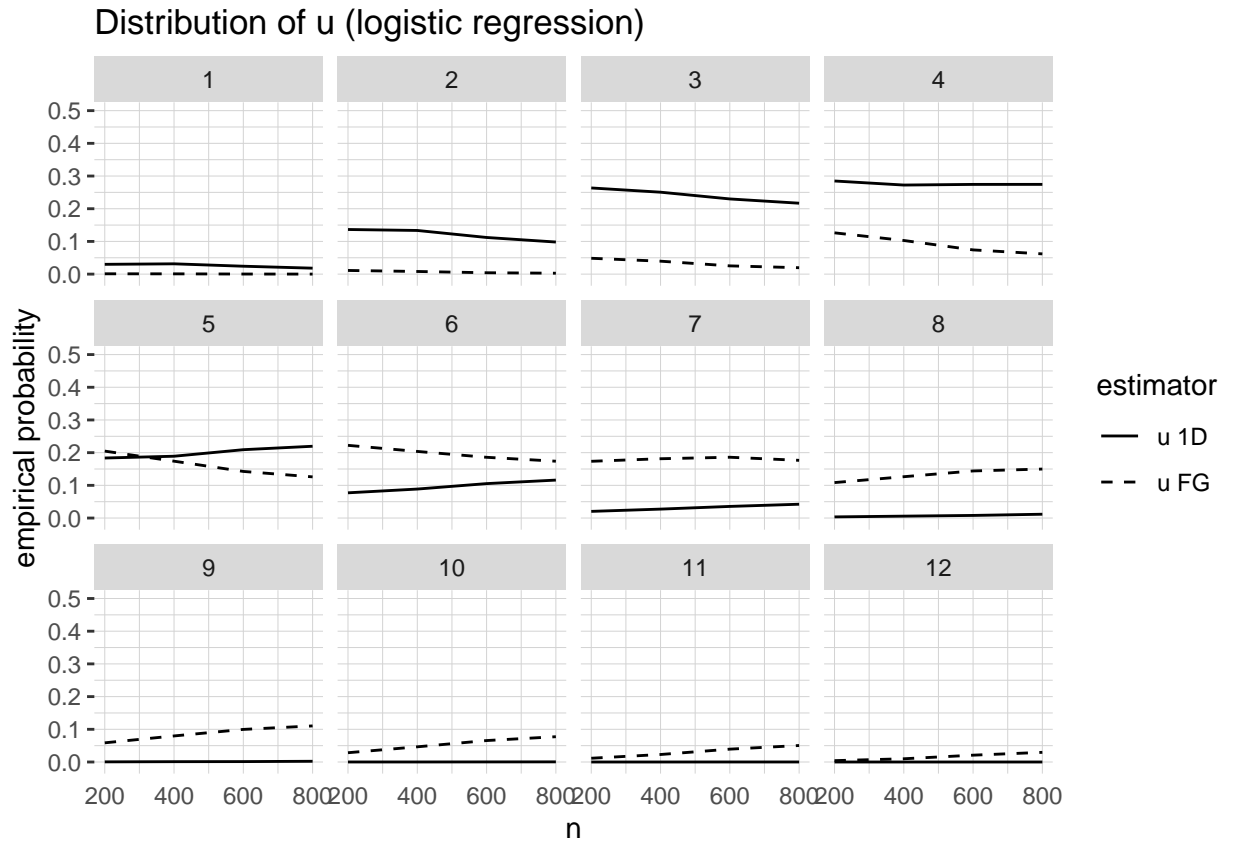
```
lsims_MC_C <- NULL
for(j in 1:4) lsims_MC_C[[j]] <-
  rbind(lsims_MC[[j]], lsims_MC_2[[j]], lsims_MC_3[[j]],
        lsims_MC_4[[j]], lsims_MC_5[[j]], lsims_MC_6[[j]],
        lsims_MC_7[[j]], lsims_MC_8[[j]], lsims_MC_9[[j]],
        lsims_MC_10[[j]], lsims_MC_11[[j]], lsims_MC_12[[j]],
        lsims_MC_13[[j]], lsims_MC_14[[j]], lsims_MC_15[[j]],
        lsims_MC_16[[j]], lsims_MC_17[[j]], lsims_MC_18[[j]],
        lsims_MC_19[[j]], lsims_MC_20[[j]])
```

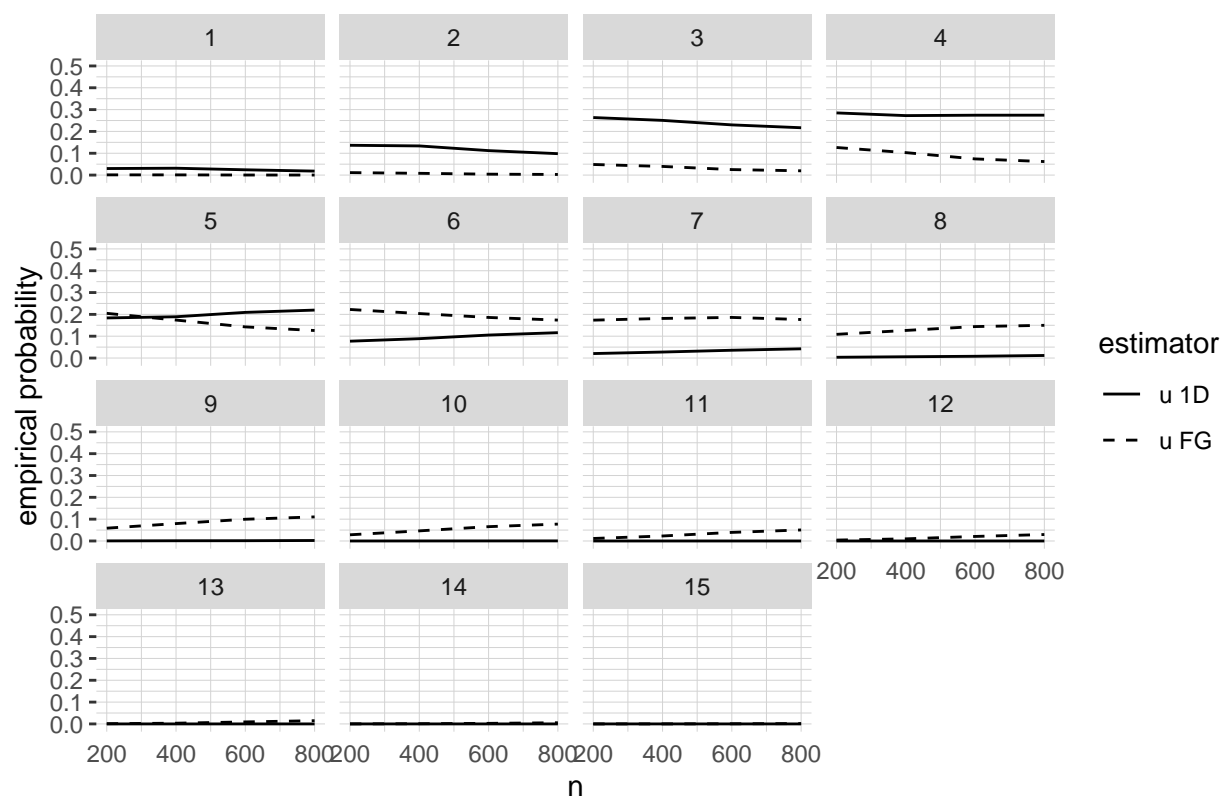
Important sample sizes are given below.

```
nMC <- 200 # Monte Carlo sample size (divided among 20 scripts)
ns <- c(200, 400, 600, 800) # sample sizes
nboot <- 500 # bootstrap sample size
```

The distribution of u across bootstrap and Monte Carlo iterates is presented below. Code is included in the accompanying .Rmd file. In the paper we only consider dimensions 1-12 for visual

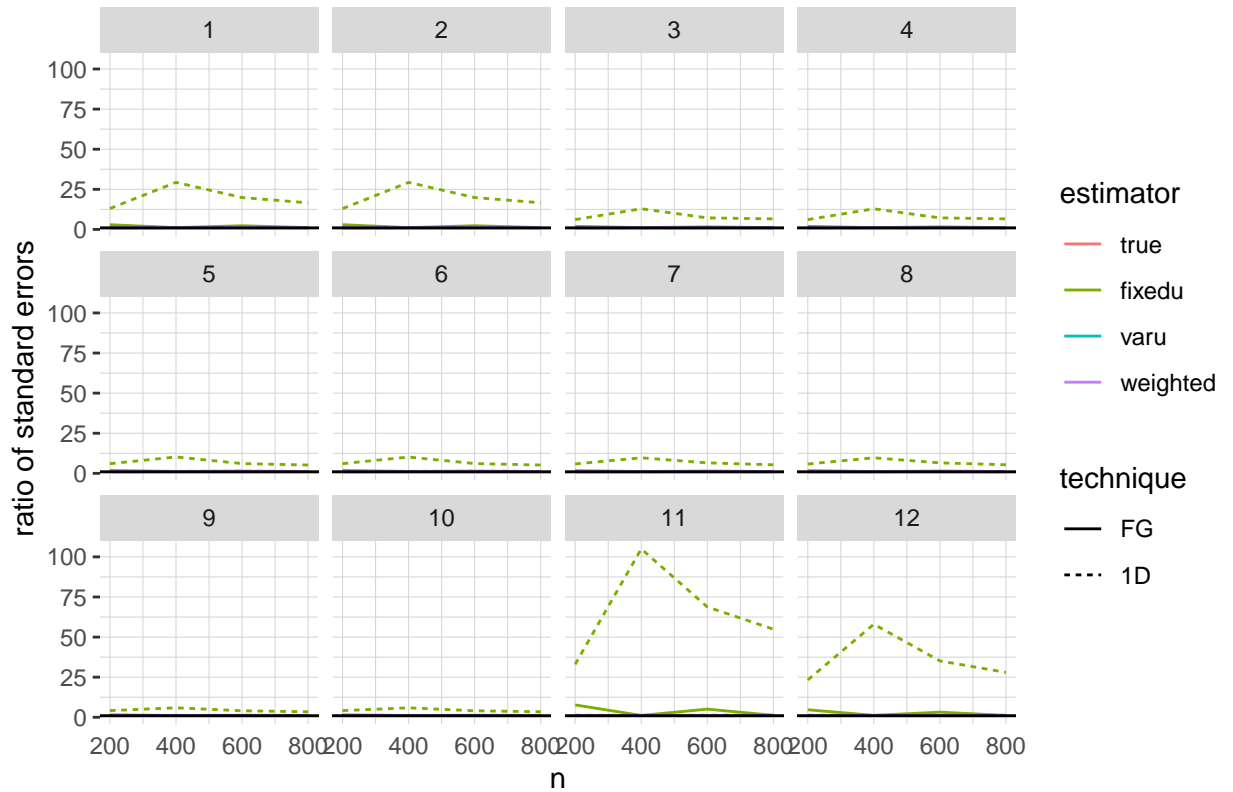
reasons. Here we also include the other 3 components for completeness.



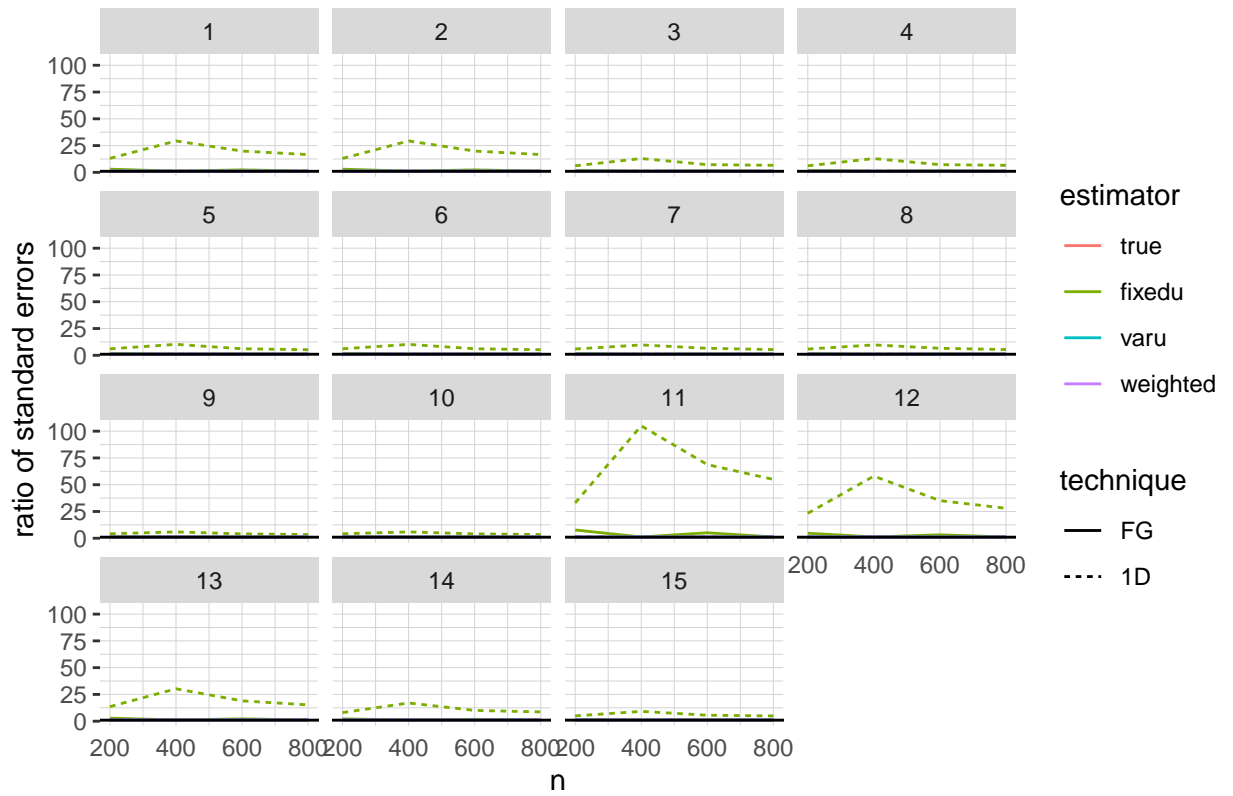
Distribution of u (logistic regression)

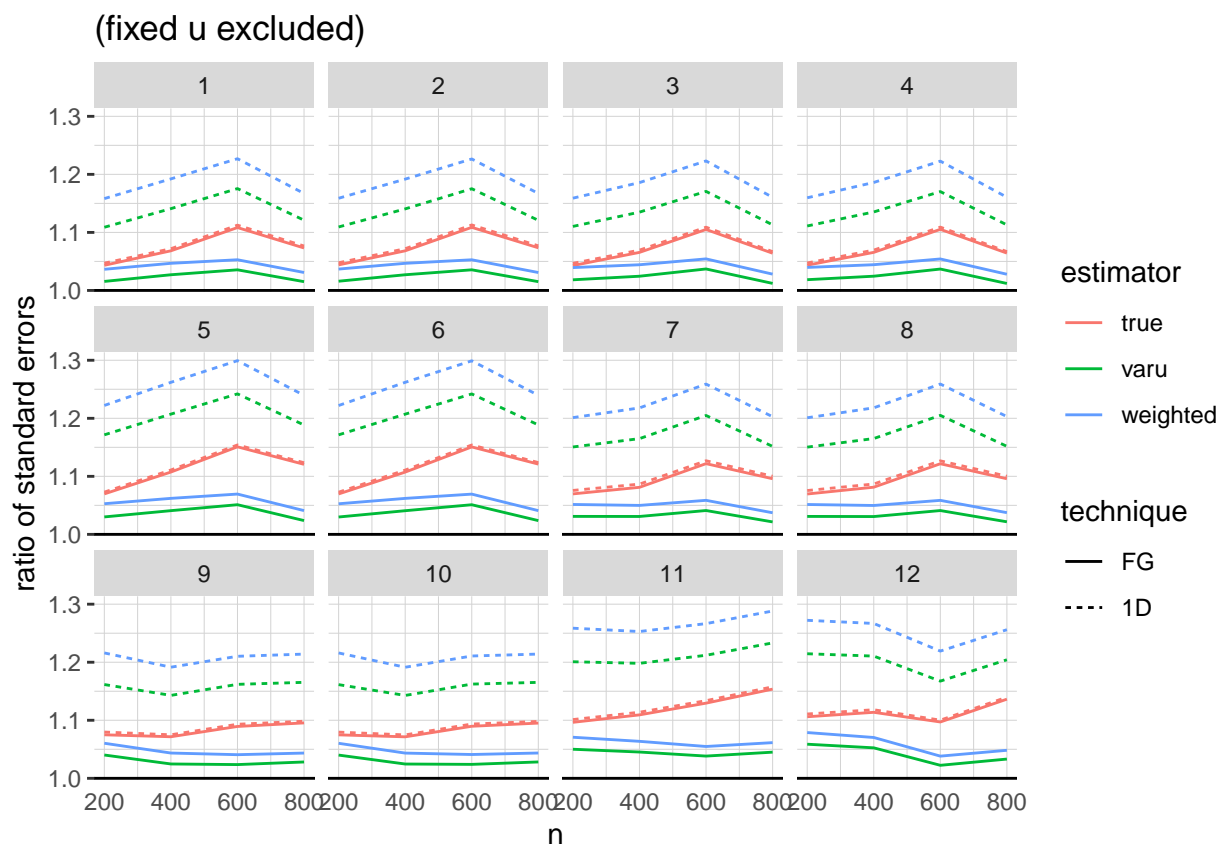
Results for ratios of bootstrap standard errors relative to the MLE are depicted here and on the next page.

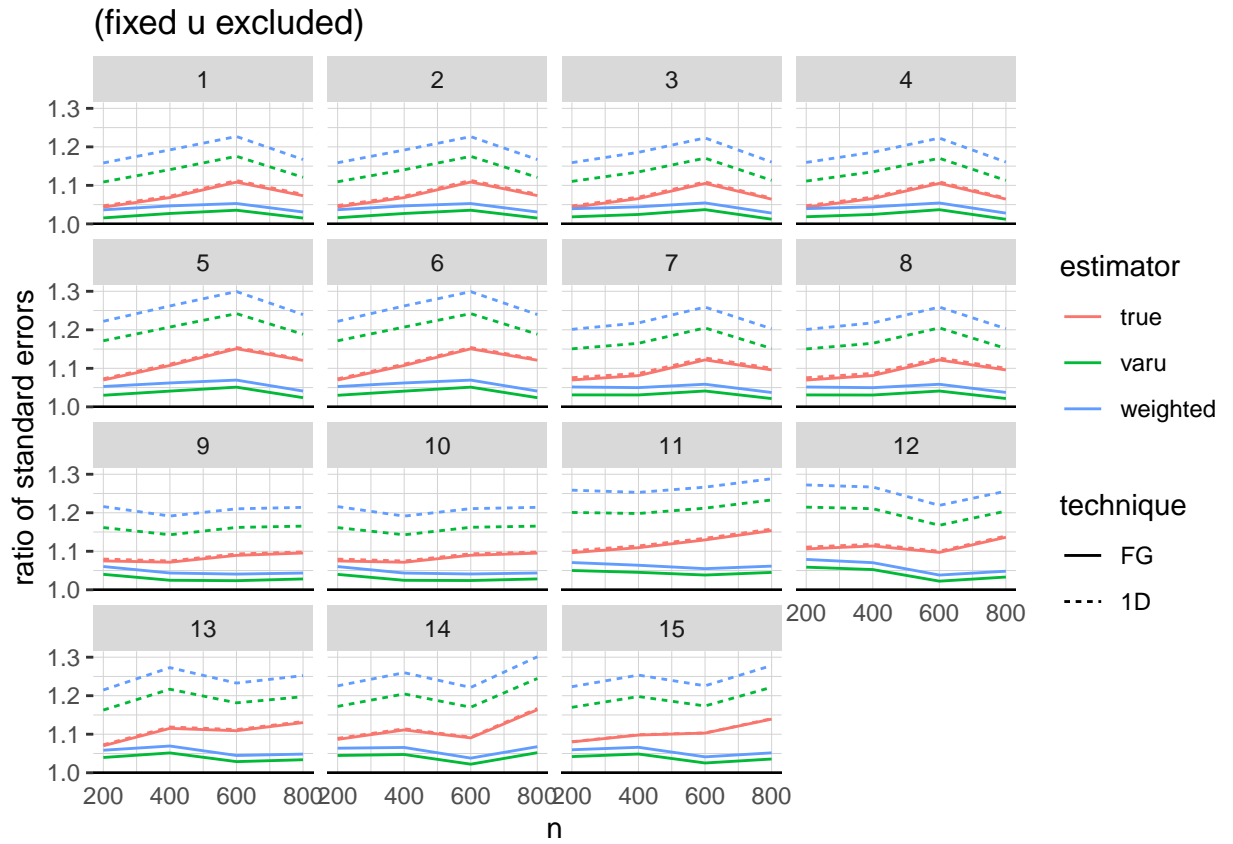
Monte Carlo simulation of ratios (logistic regression)

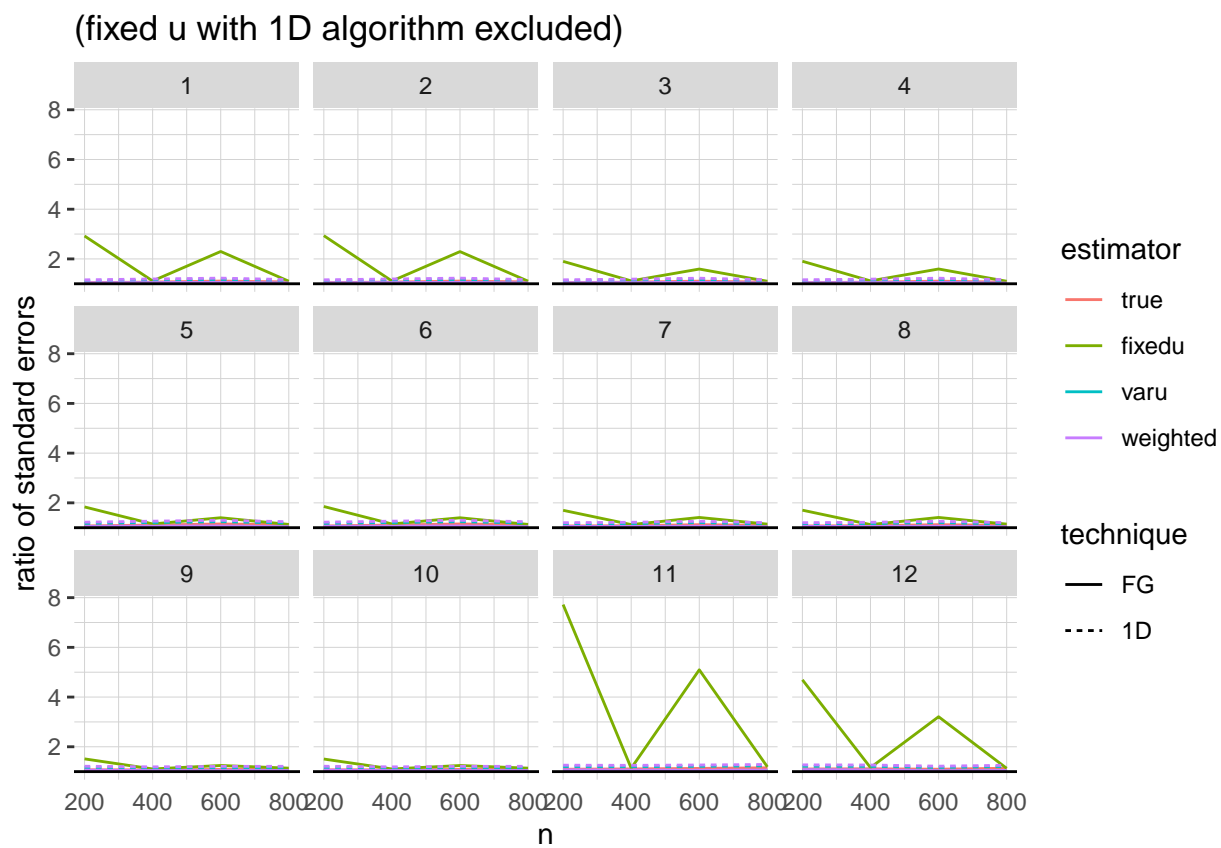


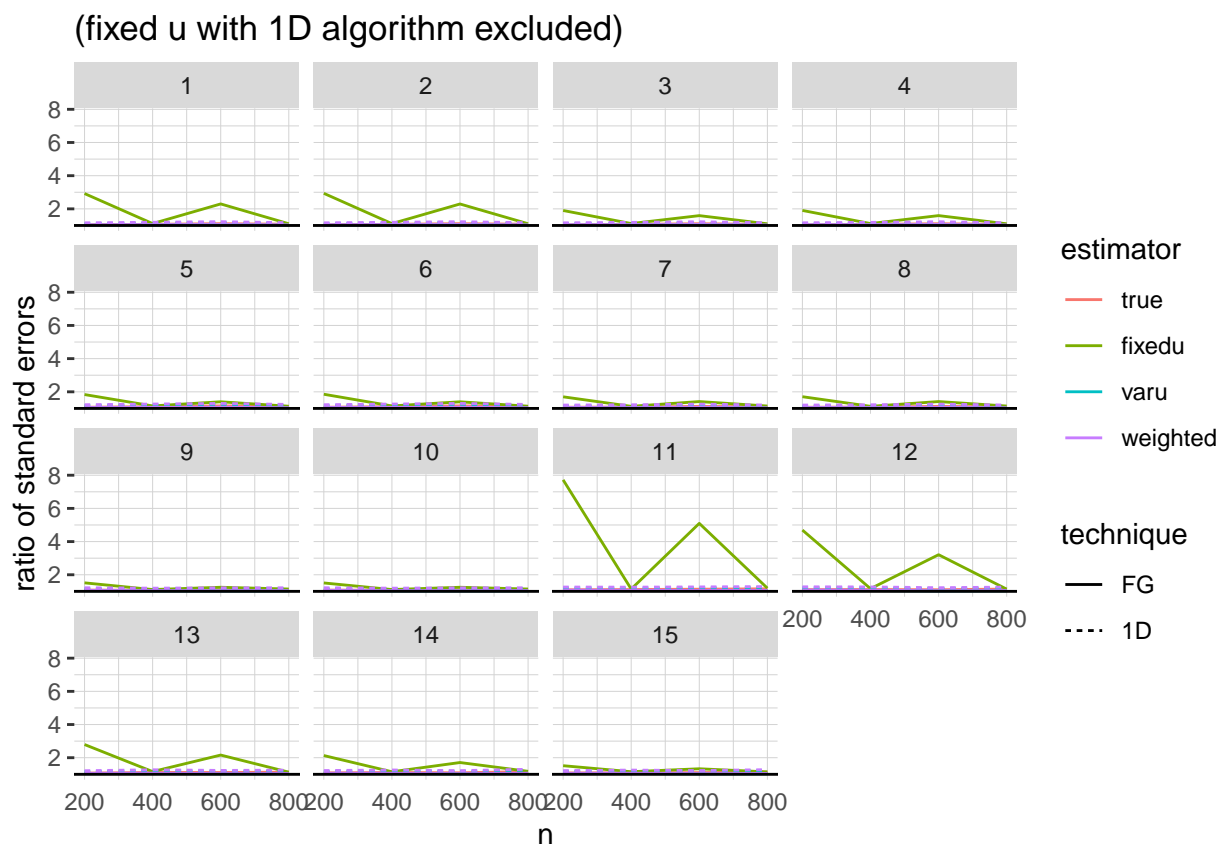
Monte Carlo simulation of ratios (logistic regression)



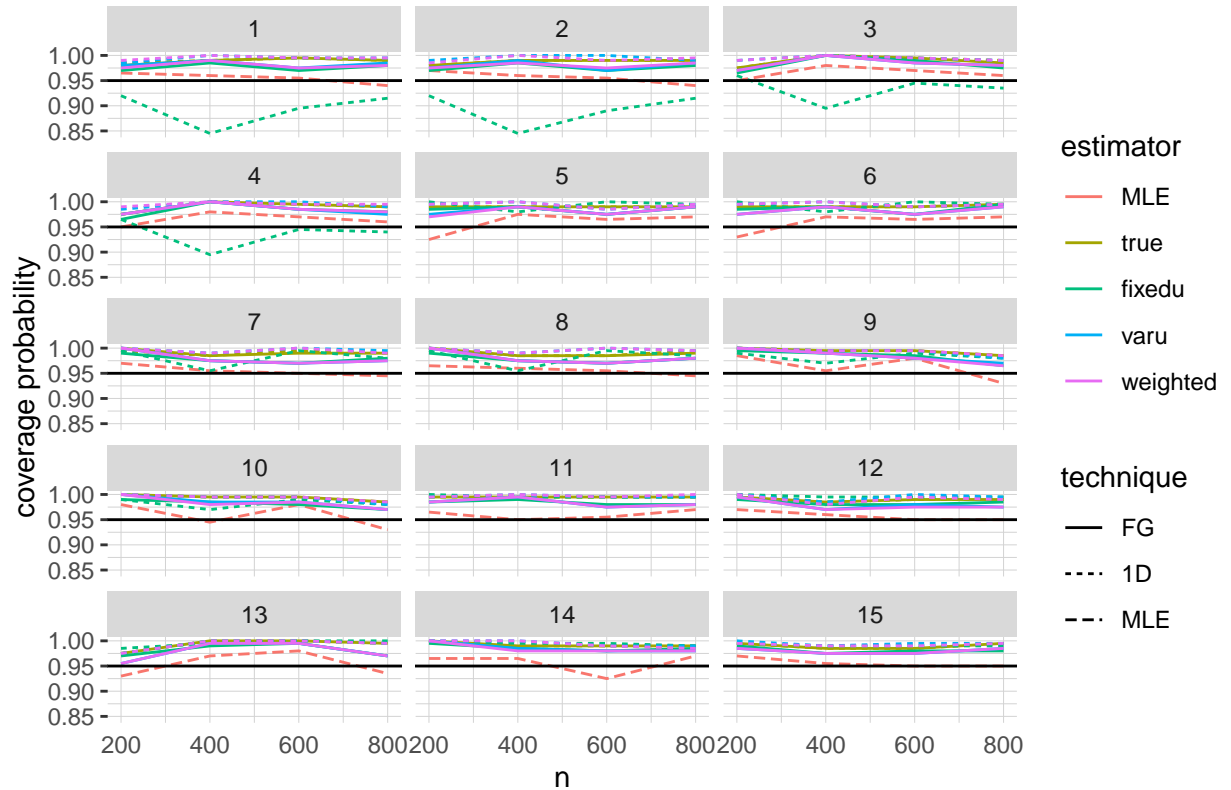


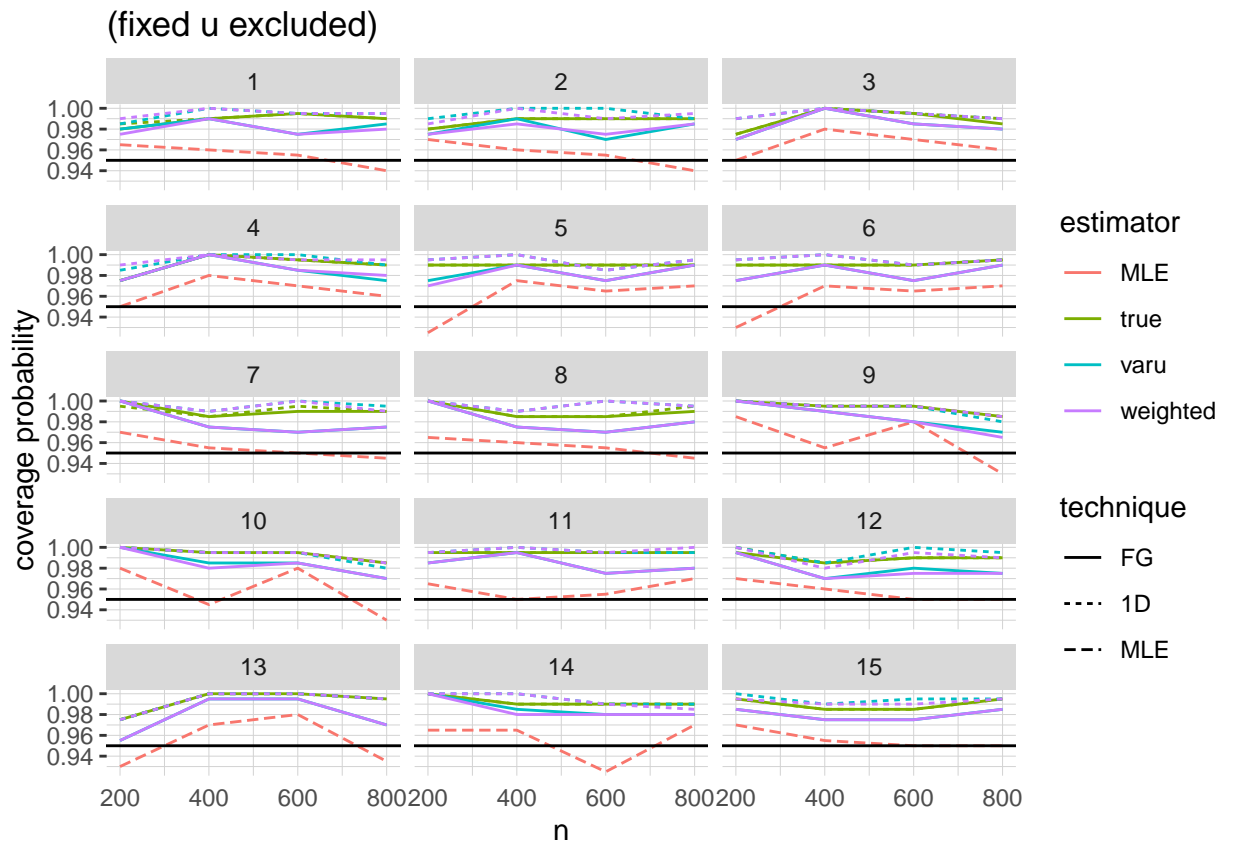






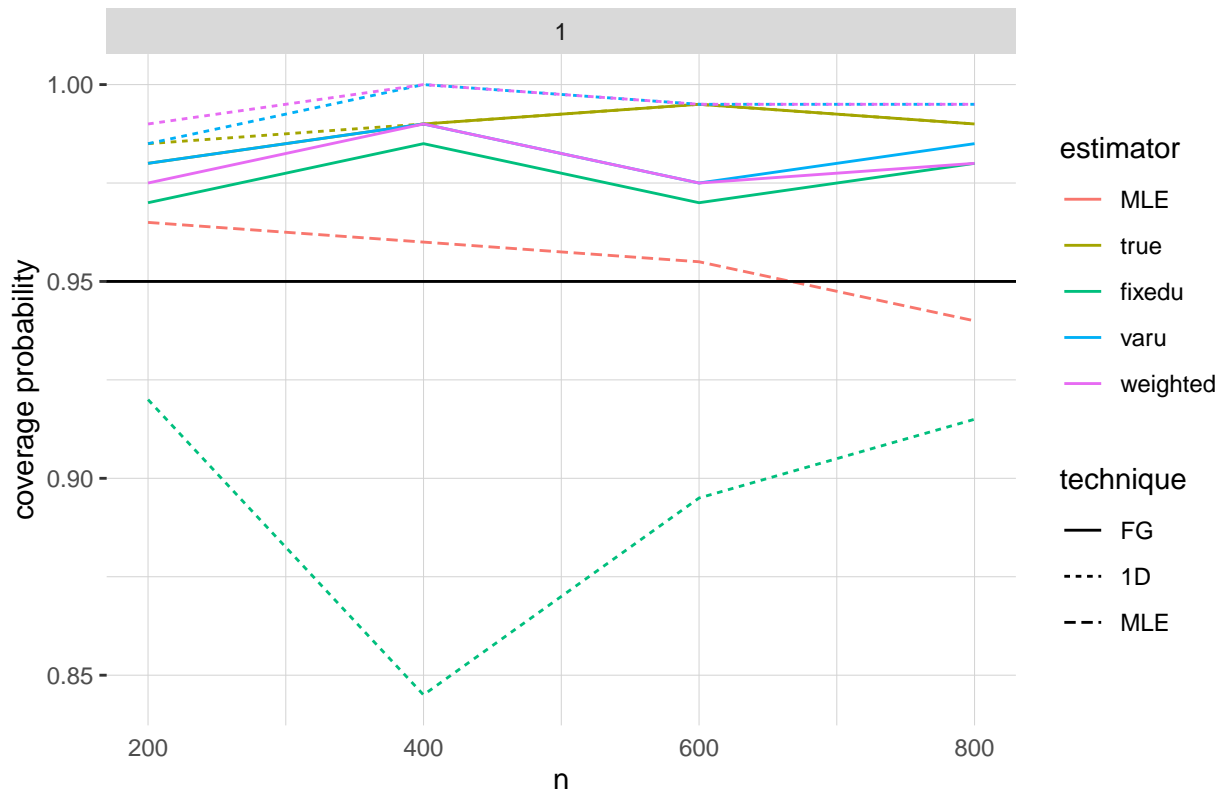
Results for univariate coverage probabilities are depicted here and on the next page.
 Monte Carlo simulation of coverage probabilities (logistic regression)





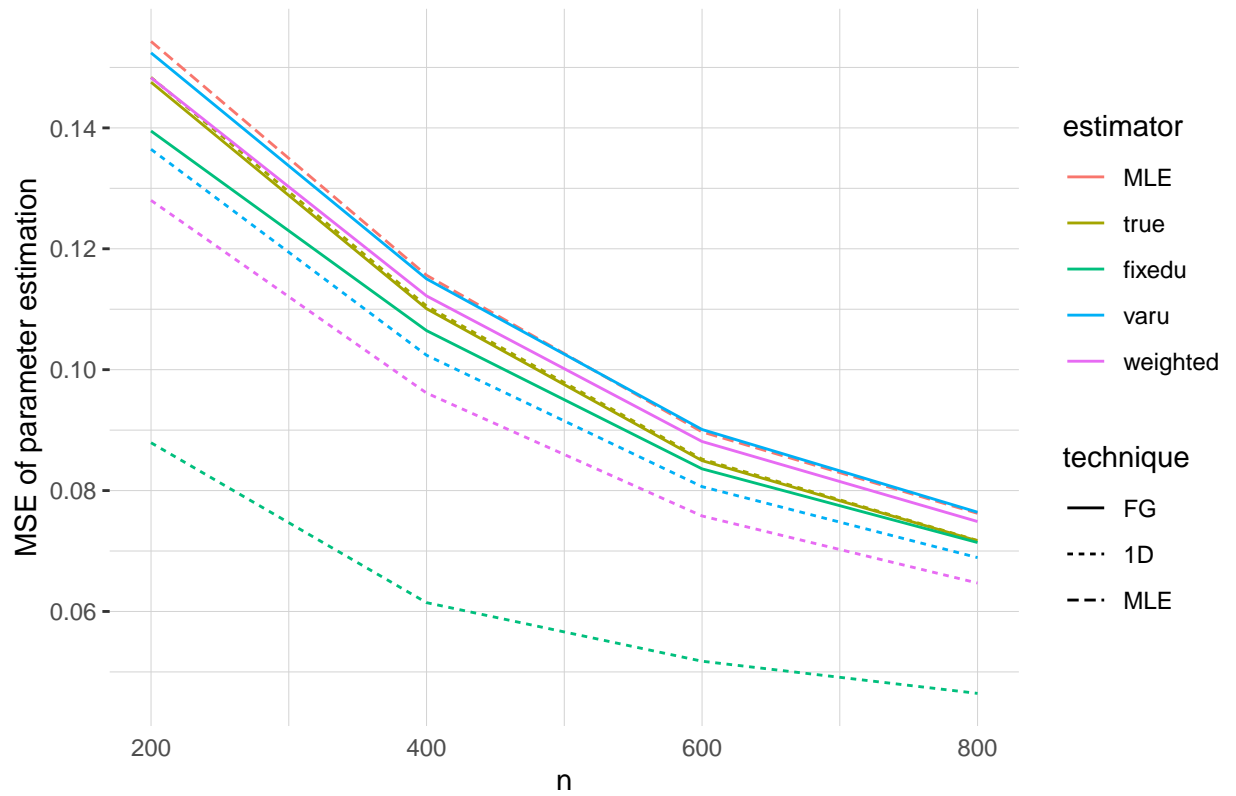
We now take a closer look at the first component. We see that the fixed u estimation regime demonstrates worrisome under coverage when using either the FG or 1D algorithms. Other regimes also demonstrate undercoverage for this component, but the problem is not as dramatic as it is for the fixed u regime.

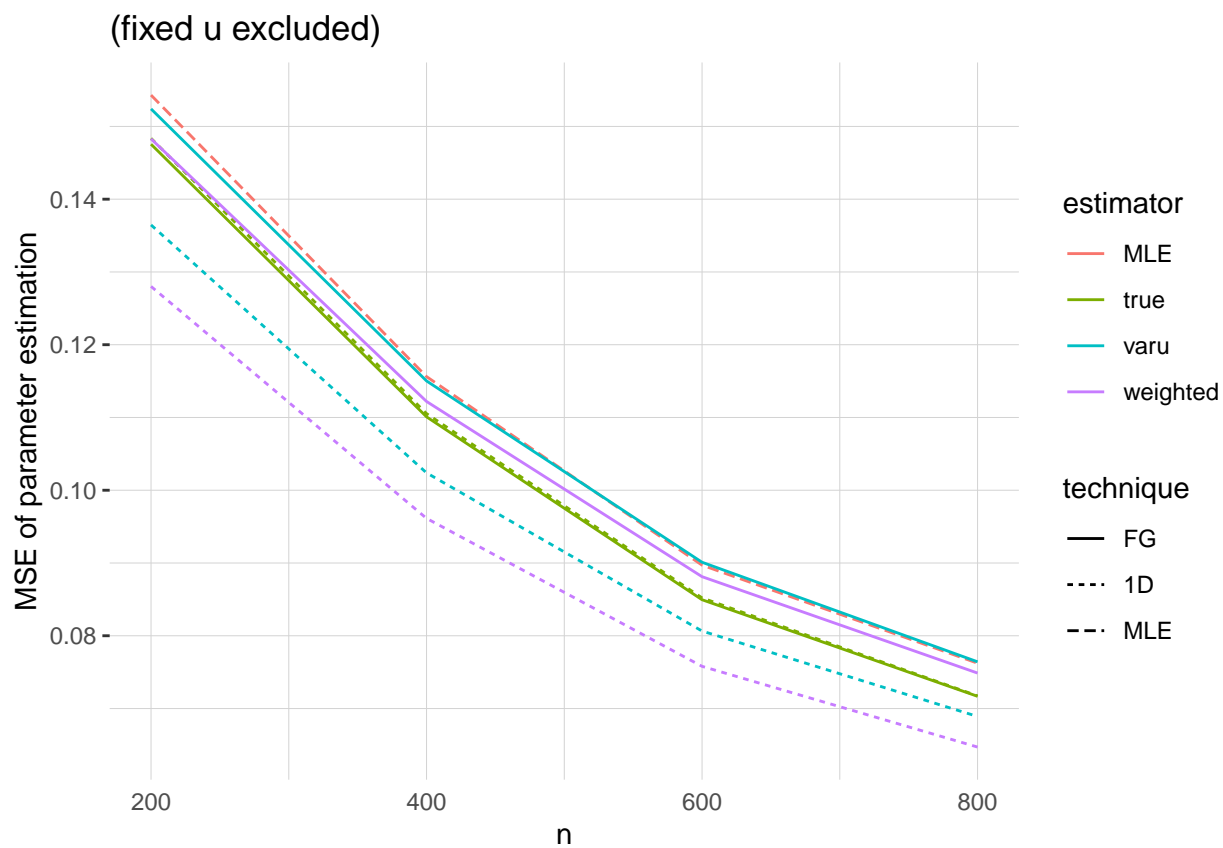
Monte Carlo simulation of coverage probabilities (logistic regression)



Results for the MSE of the averaged over Monte Carlo samples of are depicted here and on the next page.

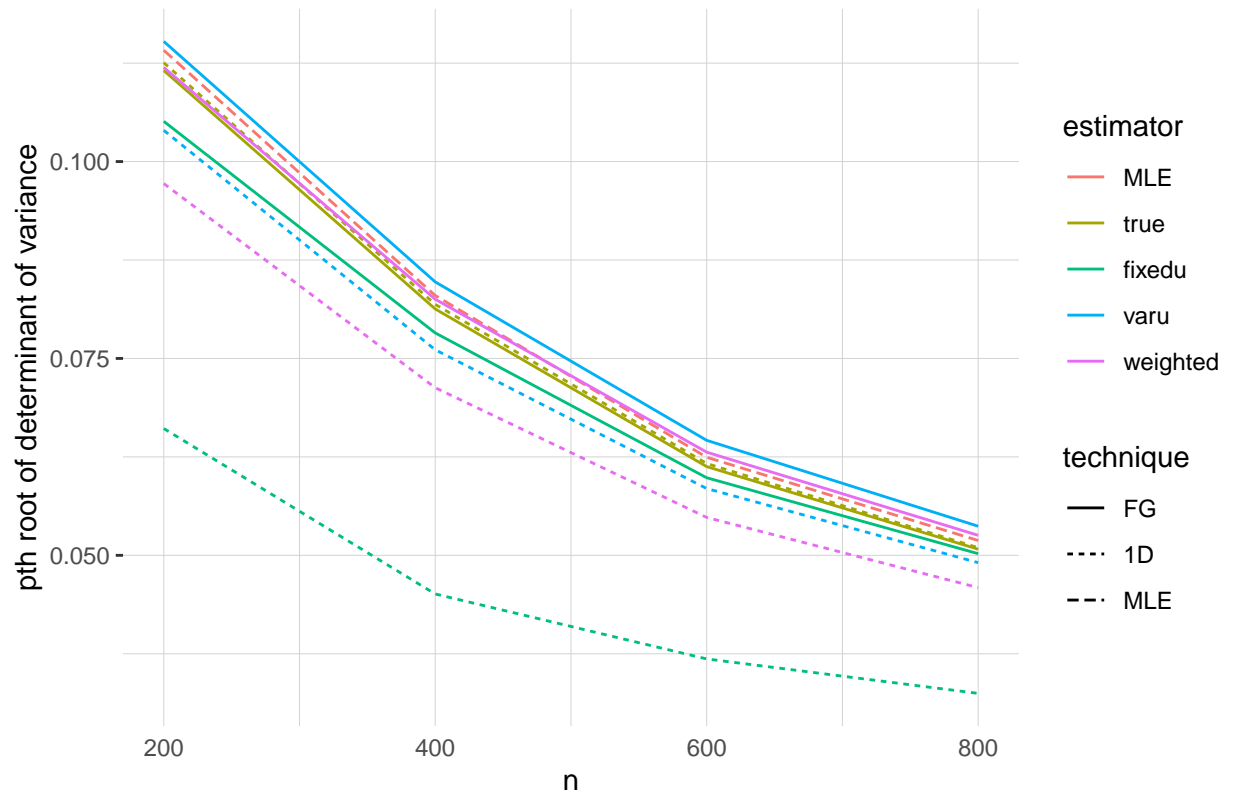
Monte Carlo simulation of MSE

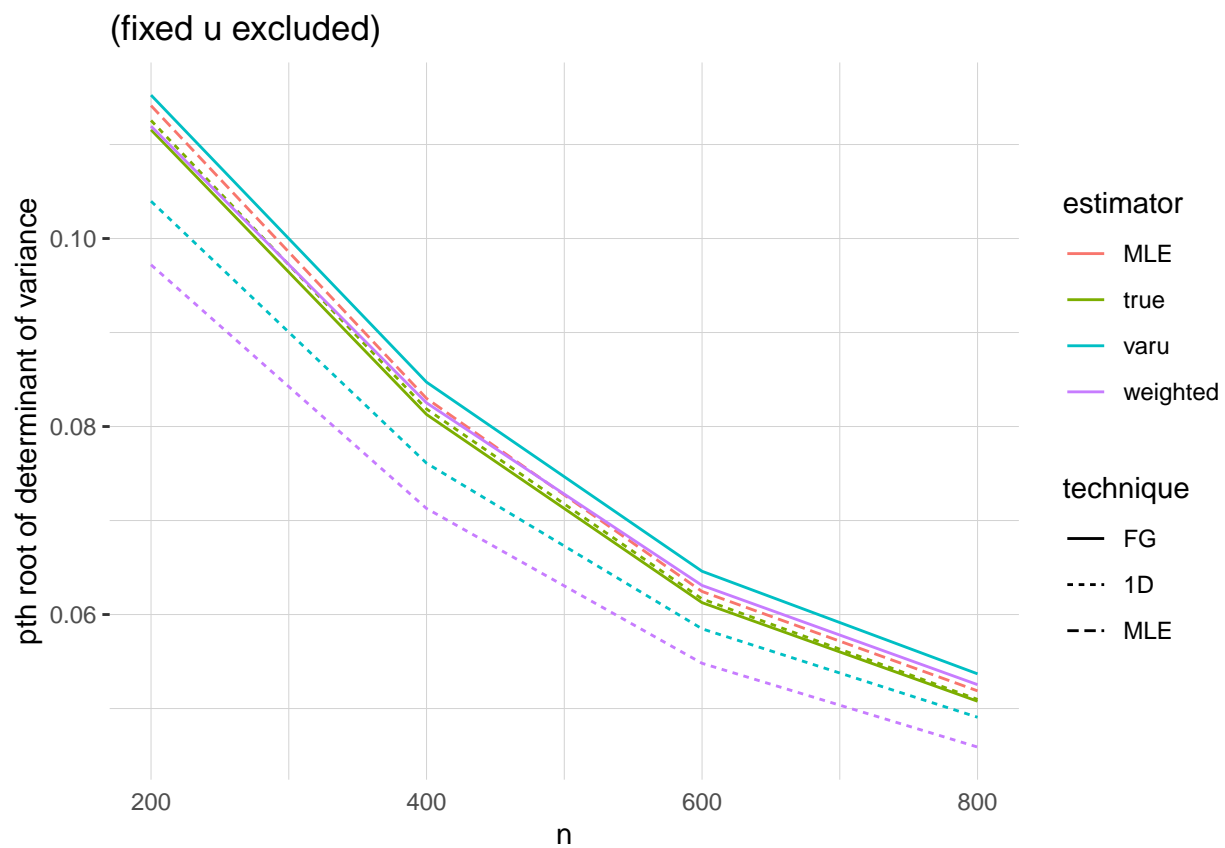




Results for the p th root determinants of the bootstrapped variances averaged over Monte Carlo samples of are depicted here and on the next page.

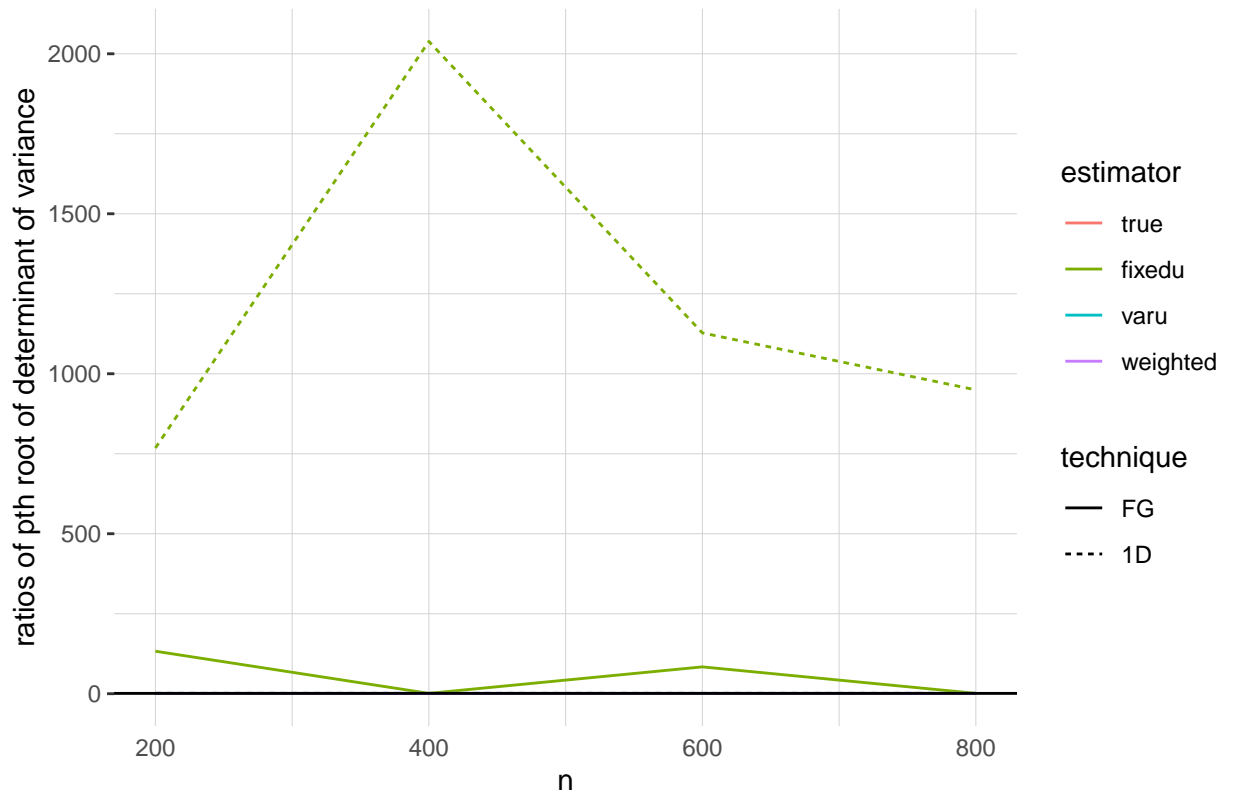
Monte Carlo simulation of variances (logistic regression)

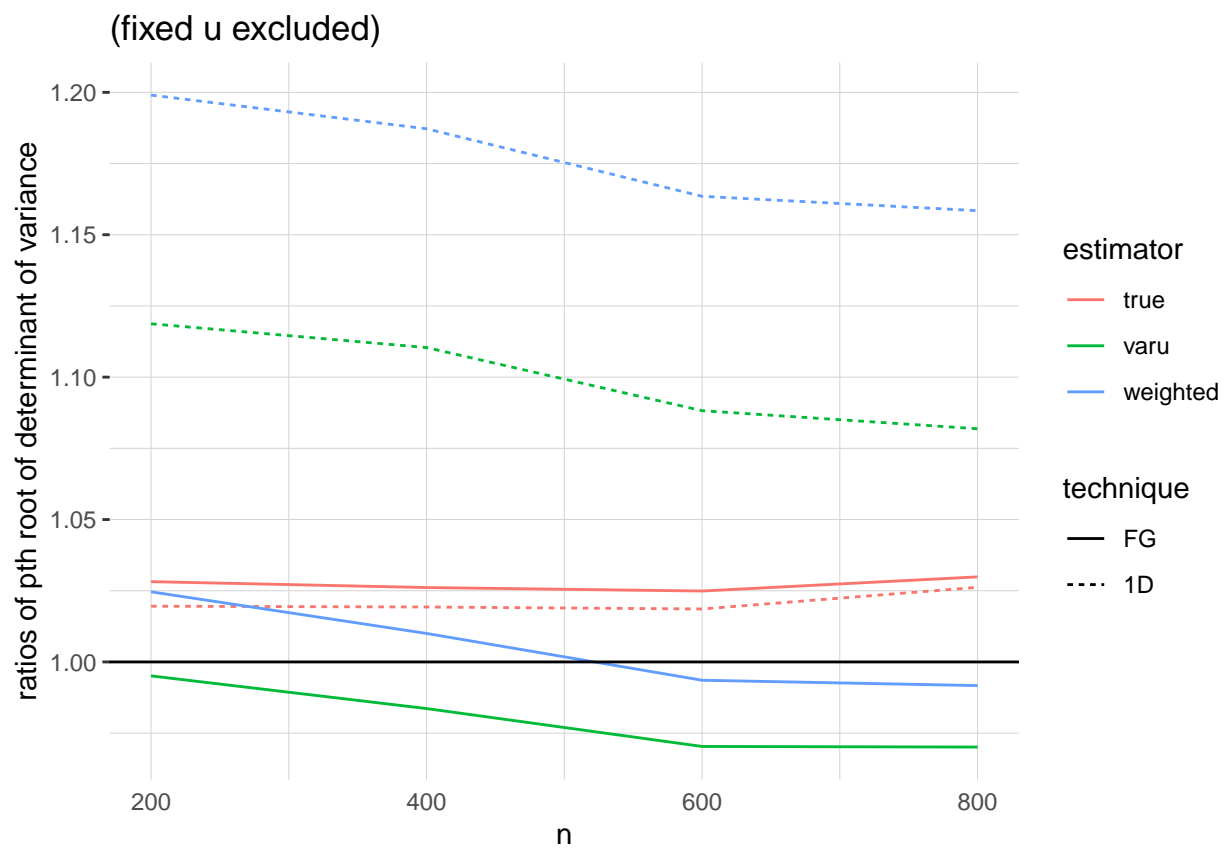




Results for the ratios of the pth root determinants of the bootstrapped variances relative to the MLE averaged over Monte Carlo samples of are depicted here and on the next page.

Monte Carlo simulation of variances (logistic regression)





Poisson regression simulations

We now provide the Poisson regression simulation in the main text. The simulation output is loaded.

```
load("psimsB.RData")
load("psimsB-2.RData")
load("psimsB-3.RData")
load("psimsB-4.RData")
```

The setup for these simulations are similar to the logistic regression simulations in the previous sections. The specific configurations are again included for completeness. We first create the basis matrix Γ for the true envelope space and the basis matrix for its orthogonal complement Γ_o .

```
p <- 6; u <- 3
init <- rep(1, 6)
v1 <- matrix(init, nrow = p)
O <- qr.Q(qr(v1), complete = TRUE)
Gamma <- O[, 1:u]
Gamma0 <- O[, (u+1):p]
```

We next create the core of the material and immaterial variation, denoted as Ω and Ω_o respectively.

```
Omega <- diag(exp(c(0, 1, 2)))
Omega0 <- diag(exp(c(-4, -3, -1)))
```

We now build the variance matrix of the predictor variables and construct the true canonical parameter vector (regression coefficient vector) as an element contained in $\text{span}(\Gamma)$.

```
SigmaX <- Gamma%*%Omega%*%t(Gamma) + Gamma0%*%Omega0%*%t(Gamma0)
eig <- eigen(SigmaX)
SigmaX.half <- eig$vec %*% diag(sqrt(eig$val)) %*% t(eig$vec)
beta <- -(5*Gamma[, 1] + 5*Gamma[, 2] + Gamma[, 3]) / 10
as.matrix(beta, ncol = 1)
```

```
##           [,1]
## [1,]  0.4490731
## [2,] -0.2248656
## [3,]  0.1751344
## [4,]  0.2751344
## [5,]  0.2751344
## [6,]  0.2751344
```

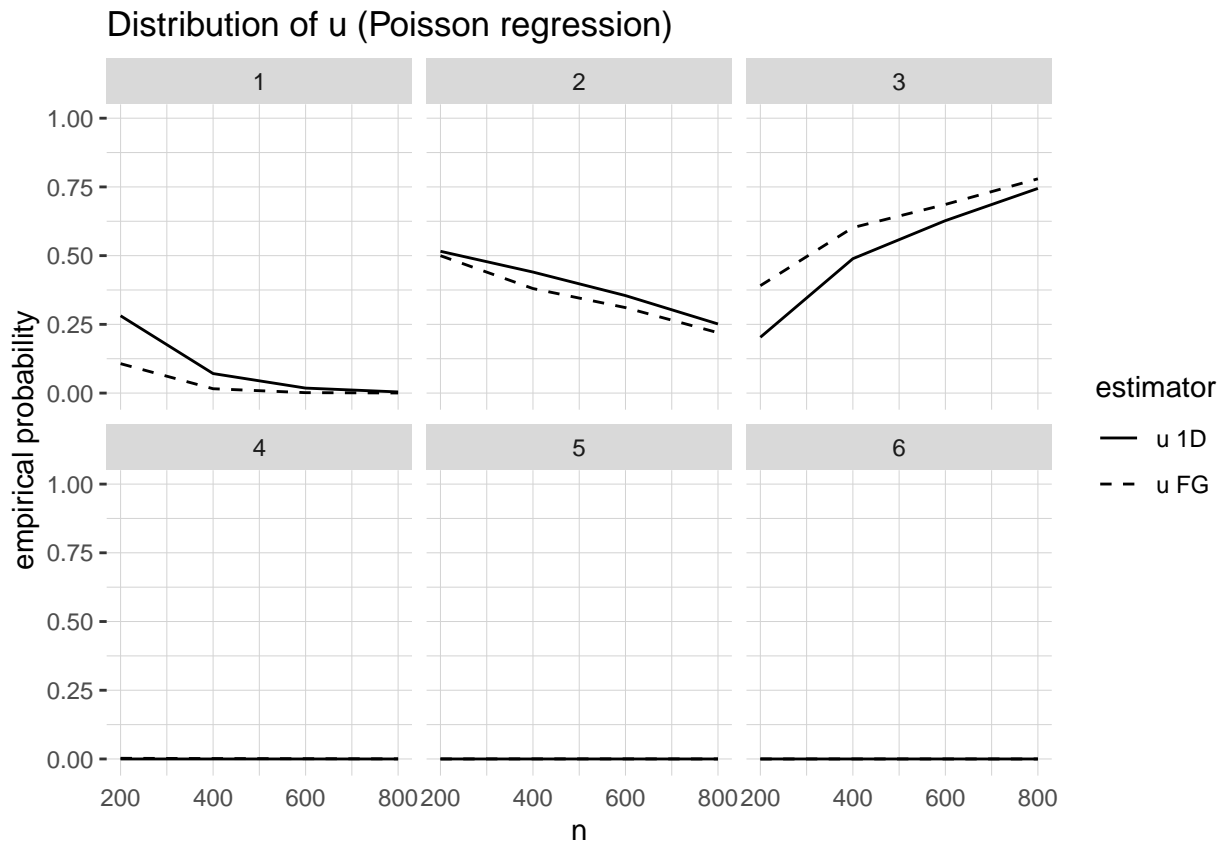
We now combine the four separate Monte Carlo simulations to estimate ratios, coverage probabilities, and distribution of the estimated envelope dimension.

```
for(j in 1:4) psims_MC_B[[j]] <-
  rbind(psim_MC_B[[j]], psims_MC_B_2[[j]],
        psims_MC_B_3[[j]], psims_MC_B_4[[j]])
```

Important sample sizes are given below.

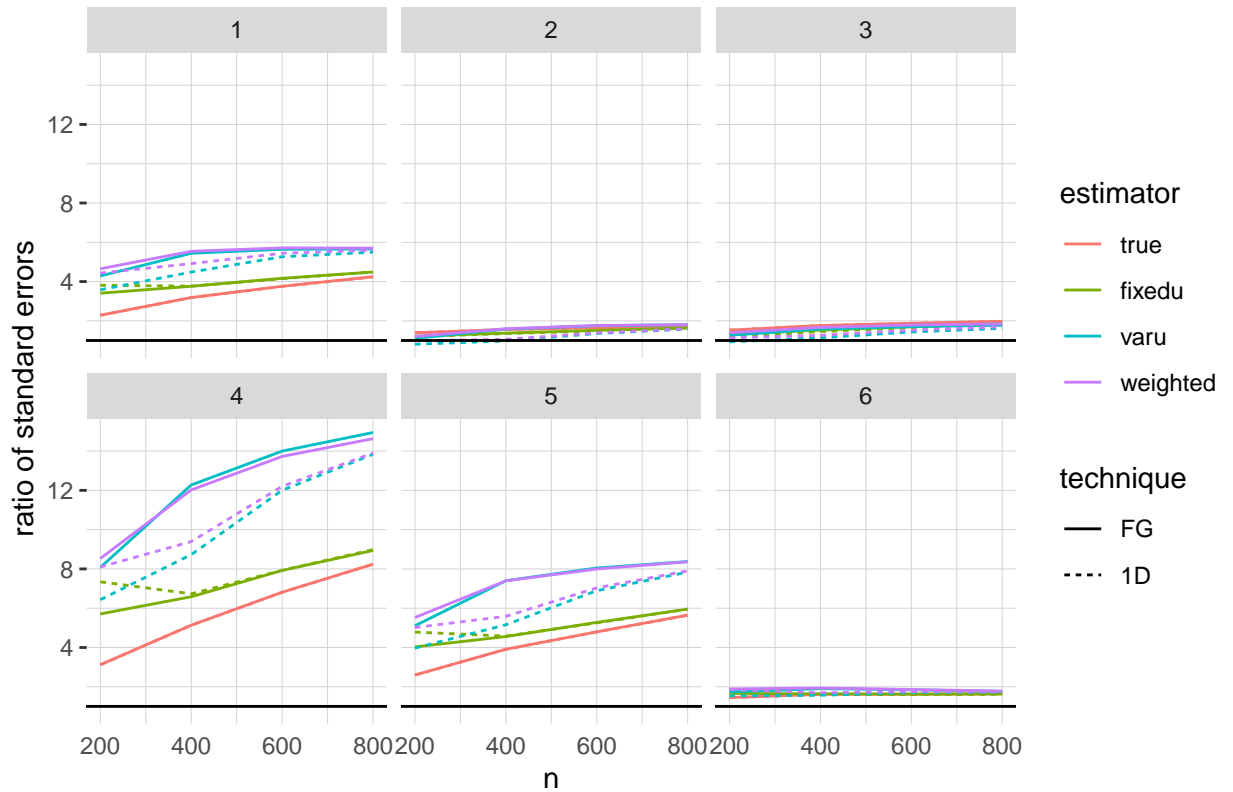
```
nMC <- 400 # Monte Carlo sample size
ns <- c(200, 400, 600, 800) # sample sizes
nboot <- 1e3 # bootstrap sample size
```

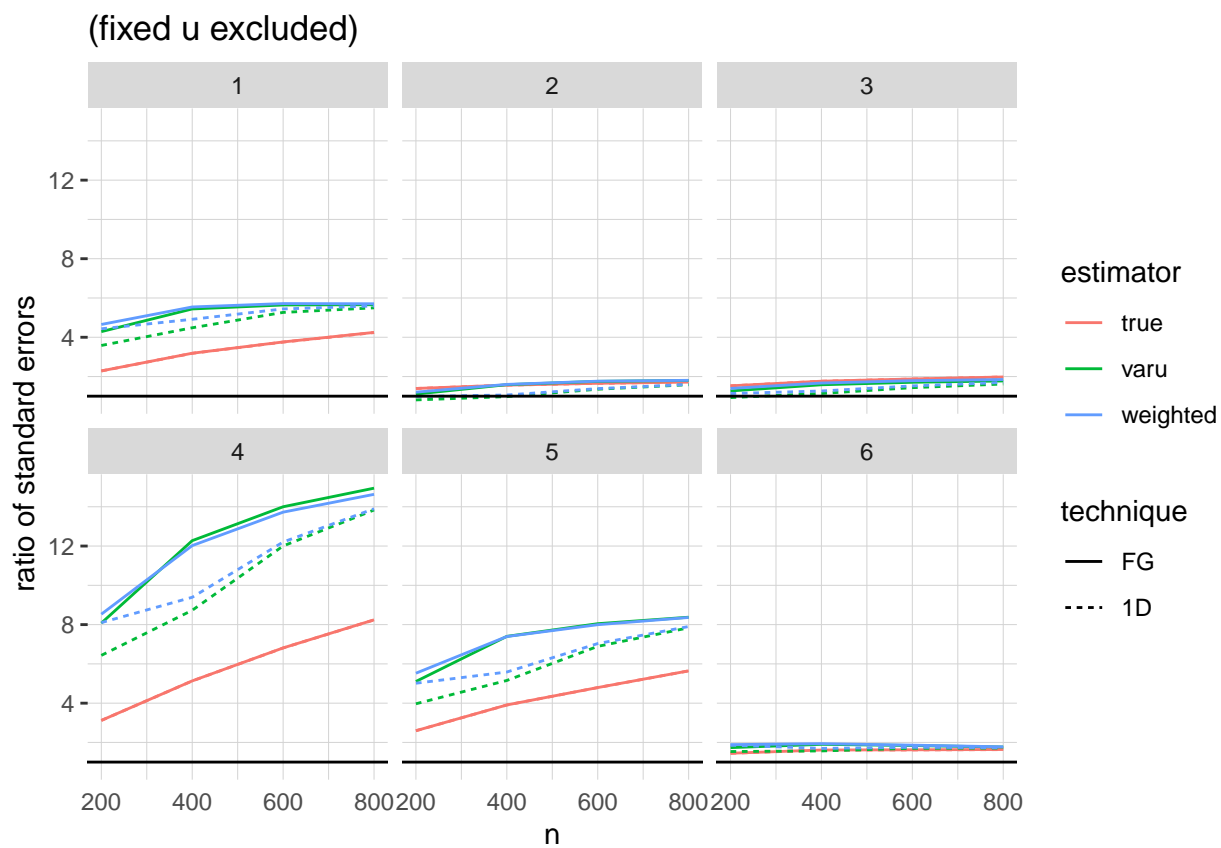
The distribution of u across bootstrap and Monte Carlo iterates is presented below. Code is included in the accompanying .Rmd file.



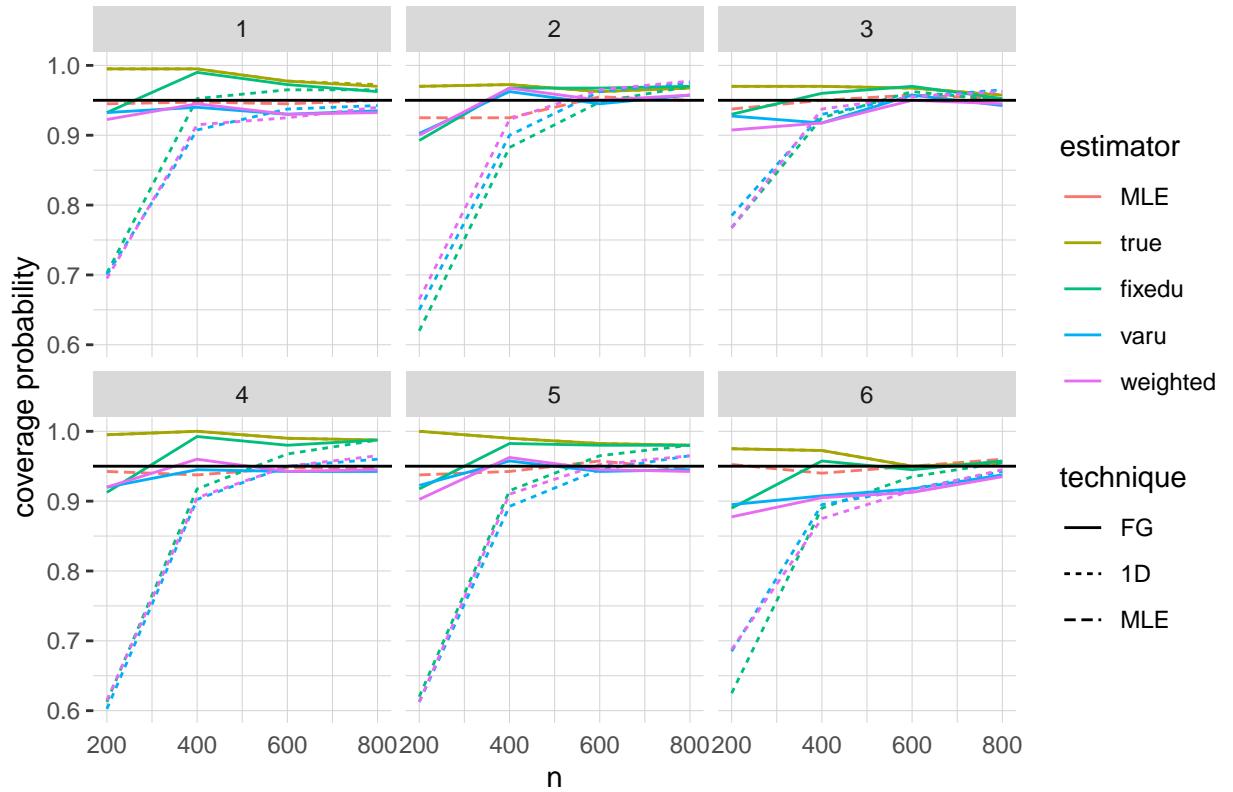
Results for ratios of bootstrap standard errors relative to the MLE are depicted here and on the next page.

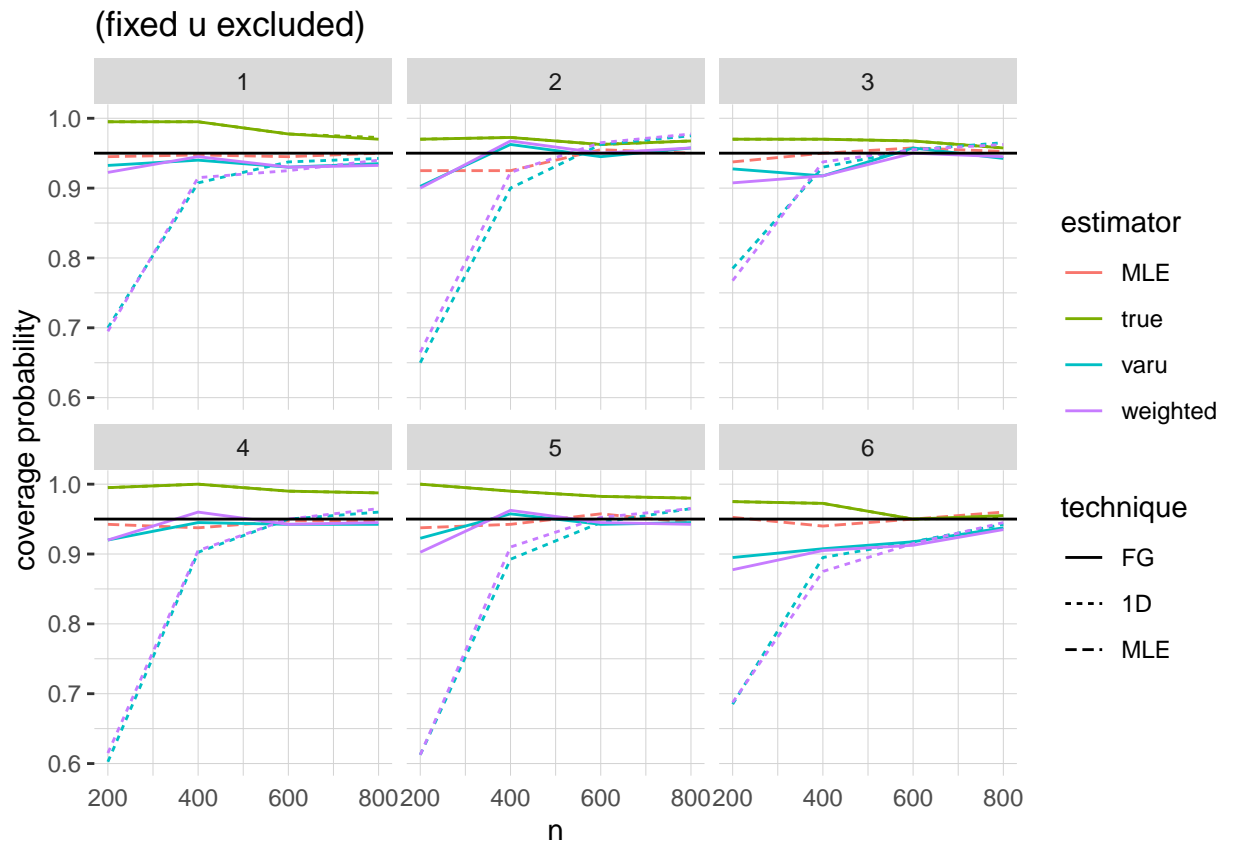
Monte Carlo simulation of ratios (Poisson regression)





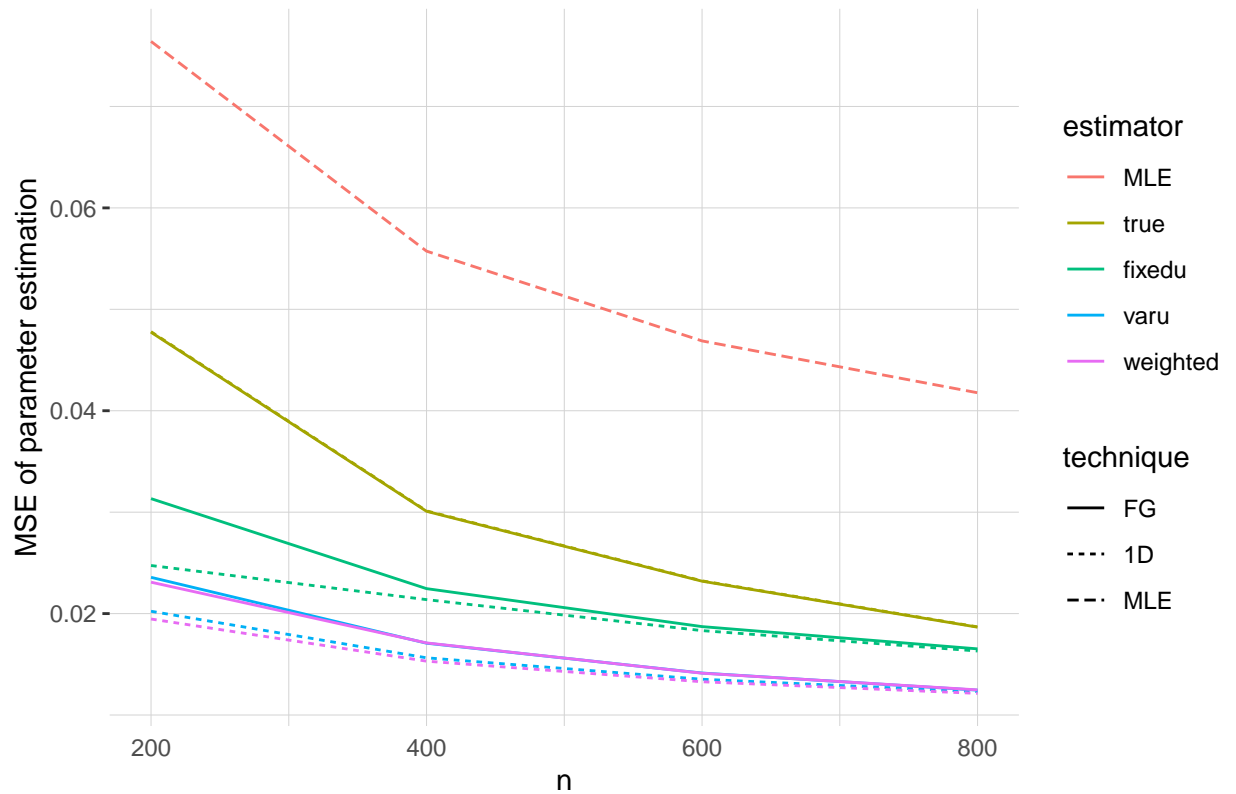
Results for univariate coverage probabilities are depicted here and on the next page.
Monte Carlo simulation of coverage probabilities (Poisson regression)





Results for the MSE of the averaged over Monte Carlo samples of are depicted here and on the next page.

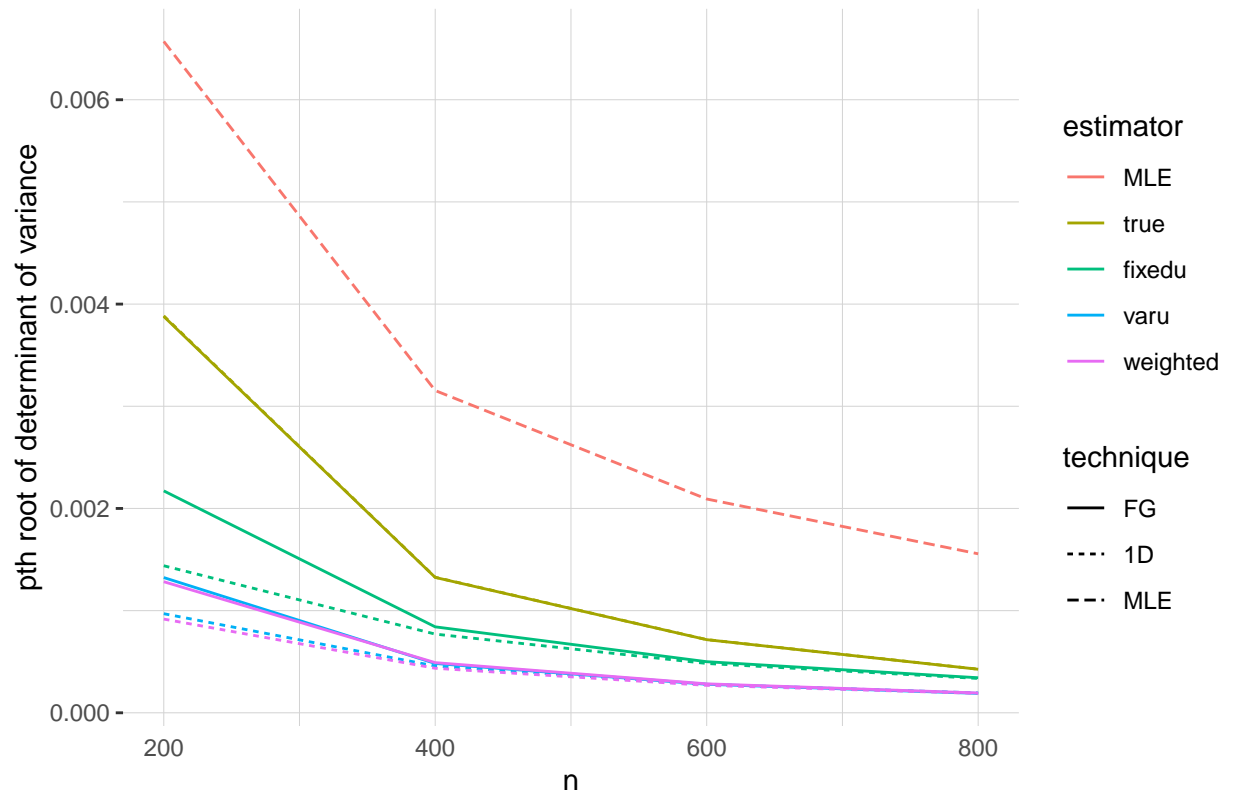
Monte Carlo simulation of MSE

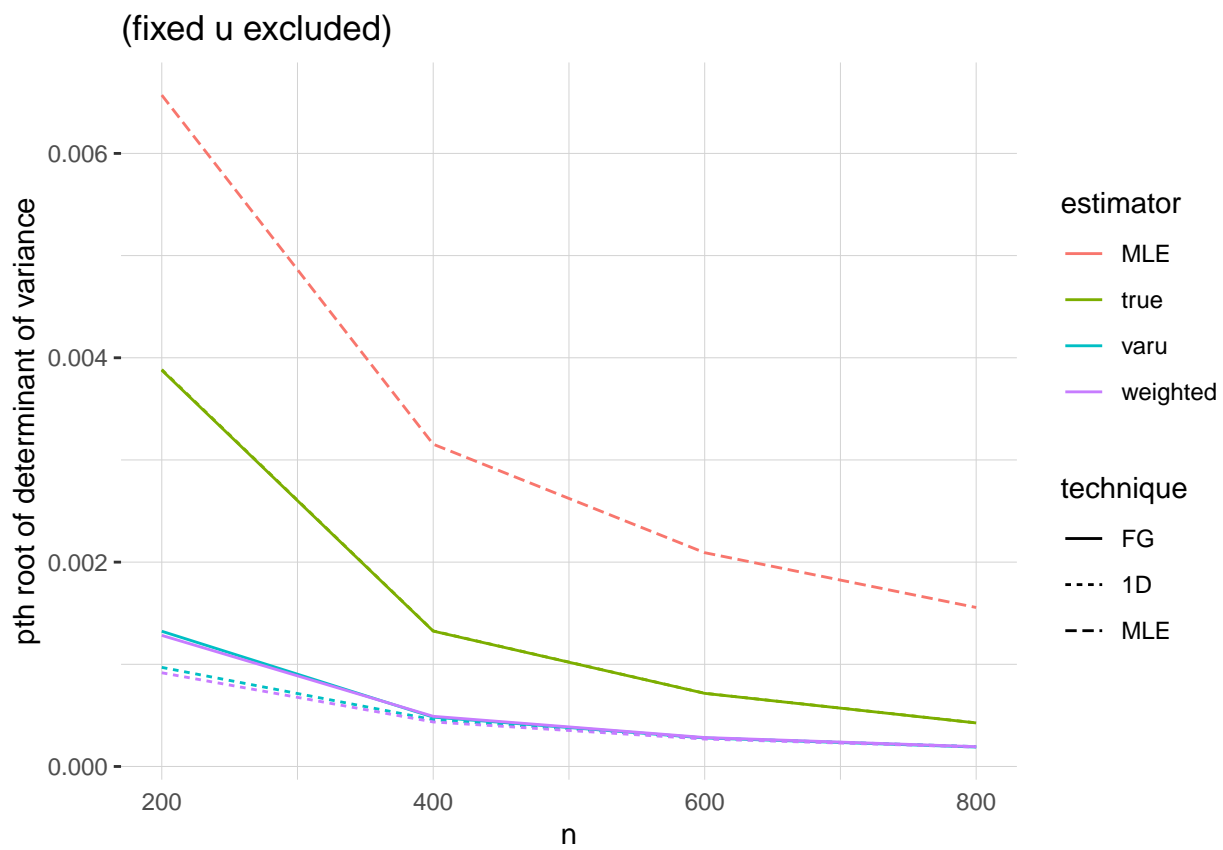




Results for the p th root determinants of the bootstrapped variances averaged over Monte Carlo samples of are depicted here and on the next page.

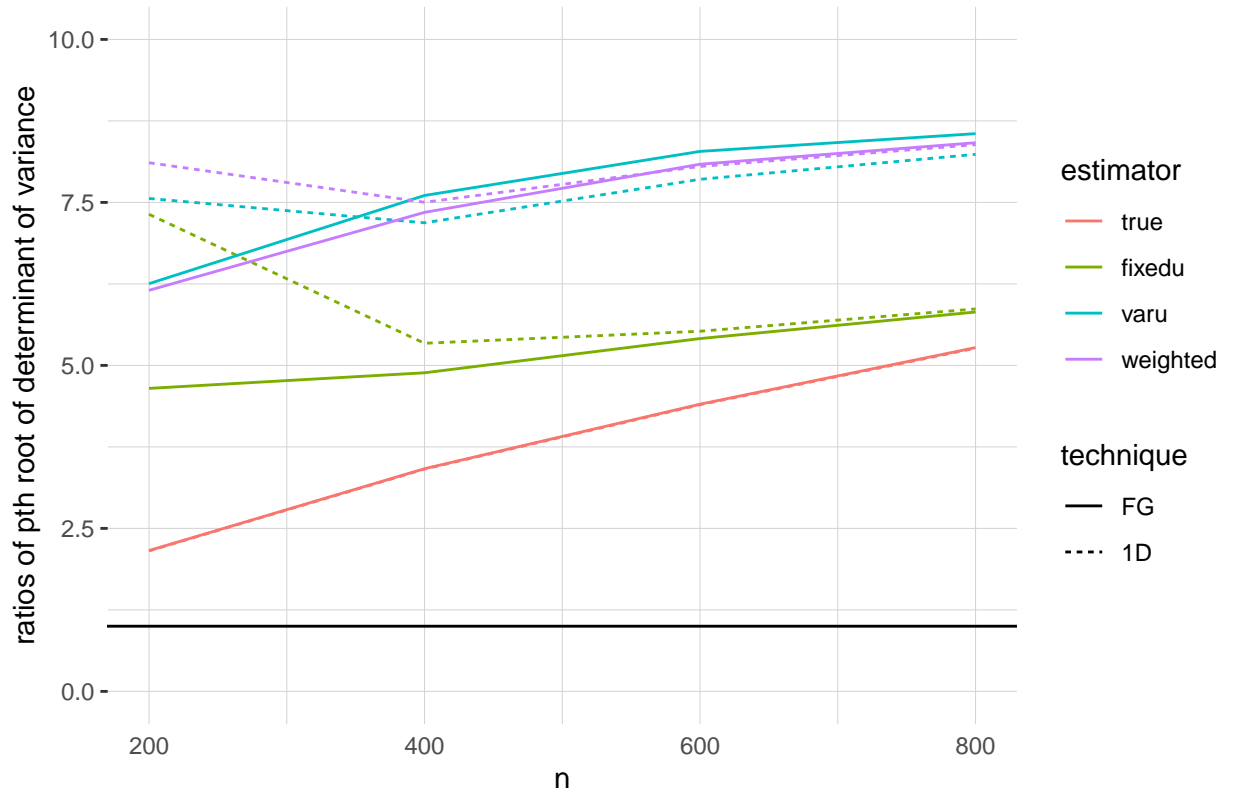
Monte Carlo simulation of variances (Poisson regression)





Results for the ratios of the pth root determinants of the bootstrapped variances relative to the MLE averaged over Monte Carlo samples of are depicted here and on the next page.

Monte Carlo simulation of variances (Poisson regression)



Diabetes example

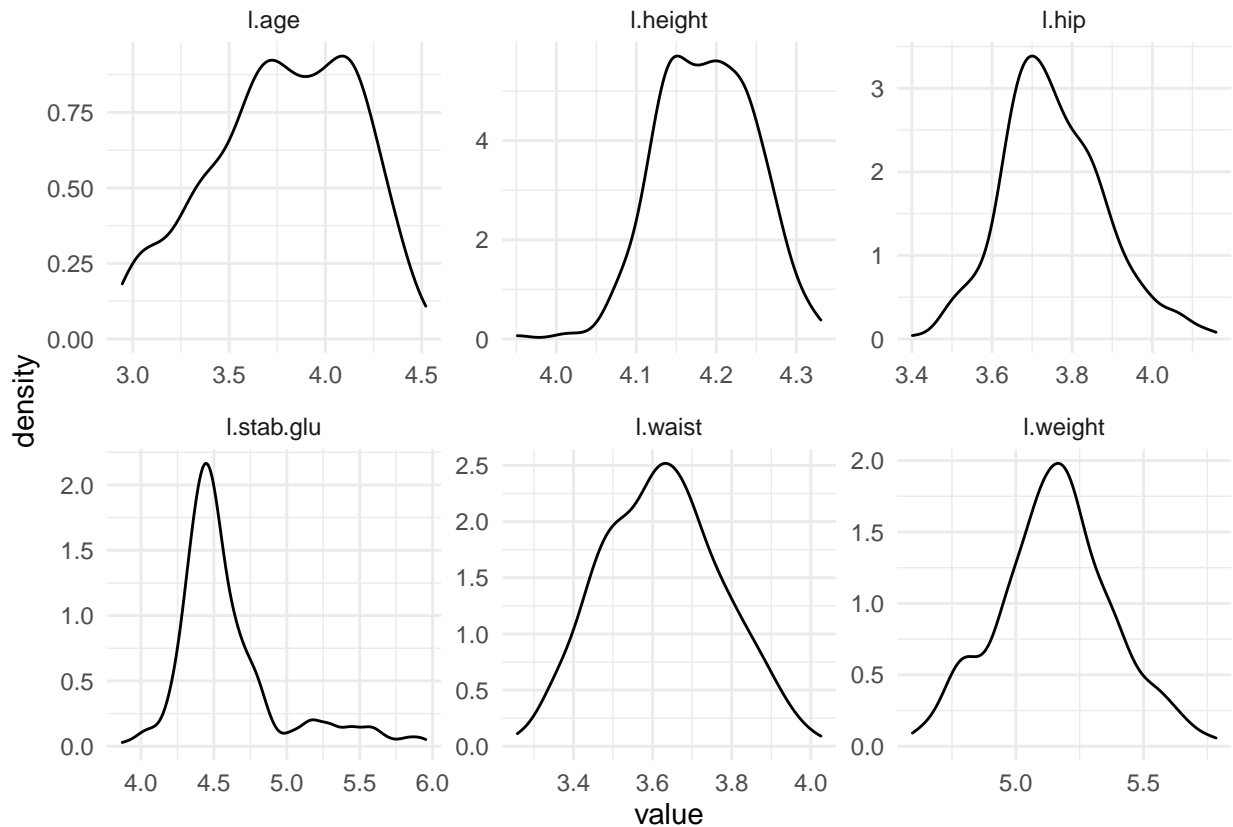
Model free envelope estimation techniques and maximum likelihood estimation are then used to estimate the canonical parameter vector corresponding to a logistic regression model (the regression coefficient vector with inverse logit link function) for diabetes diagnoses.

We load in the `diabetes` data from the `faraway` package. Log transformations are used for several predictor variables as a means to transform the variable to approximate normality while maintaining a scale that is interpretable. The response variable is a diagnosis of diabetes based in an individual's hemoglobin percentage (HbA1c). A positive diagnosis is when $\text{HbA1c} > 6.5\%$. Only complete records are kept for this analysis.

```
data(diabetes)
## add diagnosis and roughly transform predictors to univariate
## normality
dat <- diabetes %>%
  mutate(diagnose = ifelse(glyhb > 6.5, 1, 0)) %>%
  mutate(l.stab.glu = log(stab.glu), l.weight = log(weight),
         l.age = log(age), l.hip = log(hip), l.waist = log(waist),
         l.height = log(height)) %>%
  ## taking out gender
  dplyr::select(diagnose, l.age, l.weight, l.height, l.waist,
               l.hip, l.stab.glu, gender)
## exclude missing observations
dat <- na.omit(dat)
## turn gender to factor variable
dat$gender <- as.factor(dat$gender)
```

Here are density plots of the log transformed predictor variables.

```
dat %>% dplyr::select(-diagnose) %>% keep(is.numeric) %>%
  gather() %>% ggplot(aes(value)) +
  facet_wrap(~ key, scales = "free") +
  theme_minimal() +
  geom_density()
```



We now fit the model with diagnosis as a response variable and log transformed versions of age, weight, height, waist, hip, and stabilized glucose as predictors.

```
m1 <- glm(diagnose ~ ., family = "binomial", data = dat,
          x = TRUE, y = TRUE)
betahat <- m1$coefficients
p <- length(betahat)
n <- nrow(dat)
xtable(summary(m1))
```

% latex table generated in R 4.2.0 by xtable 1.8-4 package % Tue May 3 13:26:32 2022

We now perform the nonparametric bootstrap procedure for all envelope estimators mentioned in the main text and the MLE. This nonparametric bootstrap has a bootstrap sample size of 1000.

```
set.seed(13)
nboot <- 5e3
RNGkind("L'Ecuyer-CMRG")
system.time(boot_sample_diabetes <- model_boot(model = m1,
      nboot = nboot, numCores = numCores, intercept = TRUE))
```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-21.3575	23.0558	-0.93	0.3543
l.age	2.0285	0.7583	2.67	0.0075
l.weight	1.2546	2.4216	0.52	0.6044
l.height	-4.3927	5.2798	-0.83	0.4054
l.waist	2.6525	3.0997	0.86	0.3922
l.hip	-2.6424	3.6213	-0.73	0.4656
l.stab.glu	5.0333	0.6602	7.62	0.0000
genderfemale	0.1744	0.6025	0.29	0.7722

```
##      user      system    elapsed
## 5454.488    69.922    653.840
```

The distribution of the estimated dimension across bootstrap iterations is depicted below. A non-trivial amount of model selection volatility exists across iterations of our nonparametric bootstrap.

1D algorithm

```
round(table(boot_sample_diabetes[, 4*(p-1)+1]) / nboot, 3)

##
##      1      2      3      4      5
## 0.578 0.354 0.063 0.005 0.001
```

FG optimization

```
round(table(boot_sample_diabetes[, 7*(p-1)+2]) / nboot, 3)

##
##      1      2      3      4      5      6      7
## 0.239 0.355 0.213 0.106 0.057 0.022 0.008
```

We now replicate Table 2 in the main text. This table displays the ratios of bootstrap standard errors. These ratios compare the MLE to the envelope estimators under study. The bootstrap standard error for the MLE is displayed in the numerator, so that a value greater than 1 indicate variance reduction via envelope methodology.

variances

```
var_MLE <- var(boot_sample_diabetes[, 1:(p-1)])
var_fixedu_1D <- var(boot_sample_diabetes[, (p-1+1):(2*(p-1))])
var_varu_1D <- var(boot_sample_diabetes[, (2*(p-1)+1):(3*(p-1))])
var_wt_1D <- var(boot_sample_diabetes[, (3*(p-1)+1):(4*(p-1))])
var_fixedu_FG <- var(boot_sample_diabetes[, (4*(p-1)+2):(5*(p-1)+1)])
var_varu_FG <- var(boot_sample_diabetes[, (5*(p-1)+2):(6*(p-1)+1)])
var_wt_FG <- var(boot_sample_diabetes[, (6*(p-1)+2):(7*(p-1)+1)])
```

standard errors

```

SE_MLE <- sqrt(diag(var_MLE))
SE_fixedu_1D <- sqrt(diag(var_fixedu_1D))
SE_varu_1D <- sqrt(diag(var_varu_1D))
SE_wt_1D <- sqrt(diag(var_wt_1D))
SE_fixedu_FG <- sqrt(diag(var_fixedu_FG))
SE_varu_FG <- sqrt(diag(var_varu_FG))
SE_wt_FG <- sqrt(diag(var_wt_FG))

# ratios
ratio_fixedu_1D <- SE_MLE / SE_fixedu_1D
ratio_varu_1D <- SE_MLE / SE_varu_1D
ratio_wt_1D <- SE_MLE / SE_wt_1D
ratio_fixedu_FG <- SE_MLE / SE_fixedu_FG
ratio_varu_FG <- SE_MLE / SE_varu_FG
ratio_wt_FG <- SE_MLE / SE_wt_FG

# table of ratios
ratios_diabetes <- cbind(ratio_fixedu_1D,
  ratio_varu_1D, ratio_wt_1D, ratio_fixedu_FG,
  ratio_varu_FG, ratio_wt_FG)
xtable(ratios_diabetes, digits = 3)

```

% latex table generated in R 4.2.0 by xtable 1.8-4 package % Wed Oct 5 16:45:58 2022

	ratio_fixedu_1D	ratio_varu_1D	ratio_wt_1D	ratio_fixedu_FG	ratio_varu_FG	ratio_wt_FG
l.age	1.213	1.230	1.264	1.213	1.083	1.102
l.weight	7.002	1.509	1.612	7.002	1.145	1.185
l.height	54.439	1.157	1.256	54.439	0.983	1.026
l.waist	11.299	1.532	1.646	11.299	1.137	1.184
l.hip	17.146	1.287	1.384	17.146	1.061	1.094
l.stab.glu	1.244	1.129	1.131	1.244	1.050	1.053
genderfemale	1.186	1.179	1.197	1.186	1.061	1.076

The estimated envelope dimension is displayed in the code below. We can see that the estimated envelope dimension is 1 in the original sample, and the vector of weights is close to a point mass at 1. That being said, the bootstrap distribution for the estimated envelope dimension u is far from a point mass at 1 and the bootstrap standard errors for the weighted envelope estimator are lower than those obtained by using the variable u approach which selects u at every dimension.

```

X <- m1$x; Y <- m1$y
log_fit <- Logistic_cov(Y=Y, X=X[, -1], a=coef(m1)[1], b=coef(m1)[-1])
M <- log_fit$M; U <- log_fit$U

```

```
u1D_fit <- oneD_bic(M = M, U = U, n = n, maxdim = p-1)$u
uFG_fit <- FG_bic(M = M, U = U, n = n)
u1D_fit; uFG_fit
## [1] 1
## [1] 1
```

References

X. Zhang and Q. Mai. Model-free envelope dimension selection. *Electronic Journal of Statistics*, 12:2193–2216, 2018.