UAT SingleTable class: Jeffery prior, OR, Samanov

Jiajie Chen, Xiao Su

2022-12-29

## Enviroment setup

rm(list = ls())  
knitr::opts\_chunk$set(echo = TRUE)  
source\_all\_files <- function(path){  
 files <- list.files(path)  
 for(i in 1:length(files)){  
 full\_path <- paste(path, files[i],sep="")  
 source(full\_path)  
 }  
}  
### Load the new version  
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.2.2

new\_verrsion\_path <- 'G:/My Drive/ShareFolder/mmeta\_package/working/new\_mmeta\_dev/func\_dev/'  
source\_all\_files(new\_verrsion\_path)

## Test sampling method

### Create object

* specify data (y1, n1, y2, n2)
* specify parameters (a1, b1, a2, b2, rho)
* specify model (Sarmanov/Independent)
* Specify measure(OR/RR/RD)

single\_table\_obj\_samling <- SingleTable.create(a1=0.5, b1=0.5,   
 a2=0.5, b2=0.5,  
 rho=0.5, y1=40, n1=96, y2=49, n2=109,  
 model="Sarmanov",measure="OR")

### model fit

default

single\_table\_obj\_samling <- SingleTable.modelFit(single\_table\_obj\_samling, method = 'sampling')

control list can be specified to control the fitting process: \* n\_samples: number of posterior samples; Defualt is 5000.

* mcmc\_initial: initial values for (p1, p2) in MCMC; Default is c(0.5, 0.5)
* upper\_bound: upper bound for the measure. Default is 100.
* lower\_bound: lower bound for the measure. For RD, default is -1. For RR/OR, defualt is 0.
* num\_grids: number of grids to calculate density; The defualt is 20498.

set number of posterior samples as 3000 (default is 5000)

single\_table\_obj\_samling <- SingleTable.modelFit(single\_table\_obj\_samling, method = 'sampling', control = list(n\_sample = 3000))

set initial values for MCMC is c(0.2, 0,4) (default is c(0.5,0.5))

single\_table\_obj\_samling <- SingleTable.modelFit(single\_table\_obj\_samling, method = 'sampling', control = list(mcmc\_initial = c(0.2,0.4)))

set upper bound for the measure is 20( default is 100)

single\_table\_obj\_samling <- SingleTable.modelFit(single\_table\_obj\_samling, method = 'sampling', control = list(upper\_bound = 20))

### Summary

single\_table\_obj\_samling <- SingleTable.summary(single\_table\_obj\_samling, alpha = 0.05)

## Measure: Odds ratio   
## Model: Sarmanov Beta-Binomial Model   
## Posterior distribution:   
## Mean: 1.193   
## Median: 1.144   
## 95% ET CI: [0.652,1.988]   
## 95% HDR CI: [0.577,1.852]

change alpha level to 0.1

single\_table\_obj\_samling <- SingleTable.summary(single\_table\_obj\_samling, alpha = 0.1)

## Measure: Odds ratio   
## Model: Sarmanov Beta-Binomial Model   
## Posterior distribution:   
## Mean: 1.193   
## Median: 1.144   
## 90% ET CI: [0.720,1.812]   
## 90% HDR CI: [0.656,1.725]

change digit to 2

single\_table\_obj\_samling <- SingleTable.summary(single\_table\_obj\_samling, digit = 2)

## Measure: Odds ratio   
## Model: Sarmanov Beta-Binomial Model   
## Posterior distribution:   
## Mean: 1.19   
## Median: 1.14   
## 95% ET CI: [0.65,1.99]   
## 95% HDR CI: [0.58,1.85]

not print output

single\_table\_obj\_samling <- SingleTable.summary(single\_table\_obj\_samling, verbose = FALSE)

## Structure of SingleTable object

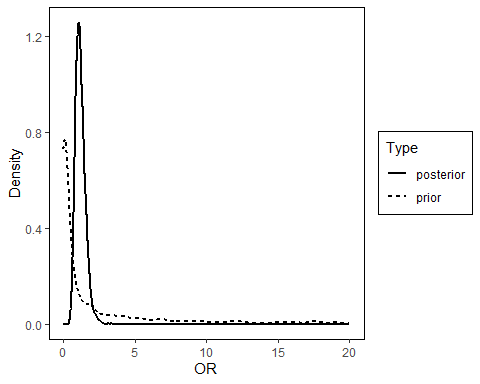
str(single\_table\_obj\_samling)

## List of 10  
## $ parms\_prior:List of 5  
## ..$ a1 : num 0.5  
## ..$ b1 : num 0.5  
## ..$ a2 : num 0.5  
## ..$ b2 : num 0.5  
## ..$ rho: num 0.5  
## $ data :List of 4  
## ..$ y1: num 40  
## ..$ n1: num 96  
## ..$ y2: num 49  
## ..$ n2: num 109  
## $ model : chr "Sarmanov"  
## $ measure : chr "OR"  
## $ method : chr "sampling"  
## $ density :List of 2  
## ..$ posterior:List of 2  
## .. ..$ x: num [1:2048] 0 0.00977 0.01954 0.02931 0.03908 ...  
## .. ..$ y: num [1:2048] 1.05e-15 4.10e-15 1.50e-14 5.39e-14 1.88e-13 ...  
## ..$ prior :List of 2  
## .. ..$ x: num [1:2048] 0 0.00977 0.01954 0.02931 0.03908 ...  
## .. ..$ y: num [1:2048] 0.728 0.735 0.742 0.748 0.754 ...  
## $ samples :List of 2  
## ..$ posterior: Named num [1:5000] 0.831 1.189 1.126 1.546 2.416 ...  
## .. ..- attr(\*, "names")= chr [1:5000] "" "" "" "" ...  
## ..$ prior : Named num [1:4169] 0.152 4.927 0.128 11.63 8.401 ...  
## .. ..- attr(\*, "names")= chr [1:4169] "" "" "" "" ...  
## $ alpha : num 0.05  
## $ summary :List of 2  
## ..$ posterior:List of 4  
## .. ..$ mean : num 1.19  
## .. ..$ median: num 1.14  
## .. ..$ ET\_CI : Named num [1:2] 0.652 1.988  
## .. .. ..- attr(\*, "names")= chr [1:2] "2.5%" "97.5%"  
## .. ..$ HDR\_CI: Named num [1:2] 0.577 1.852  
## .. .. ..- attr(\*, "names")= chr [1:2] "0.78%" "95.78%"  
## ..$ prior :List of 4  
## .. ..$ mean : num 2.2  
## .. ..$ median: num 0.323  
## .. ..$ ET\_CI : Named num [1:2] 5.54e-05 1.62e+01  
## .. .. ..- attr(\*, "names")= chr [1:2] "2.5%" "97.5%"  
## .. ..$ HDR\_CI: Named num [1:2] 1.15e-07 1.23e+01  
## .. .. ..- attr(\*, "names")= chr [1:2] "0.01201923%" "95.01202%"  
## $ digit : num 3  
## - attr(\*, "class")= chr "SingleTable"

## overlay density:

default

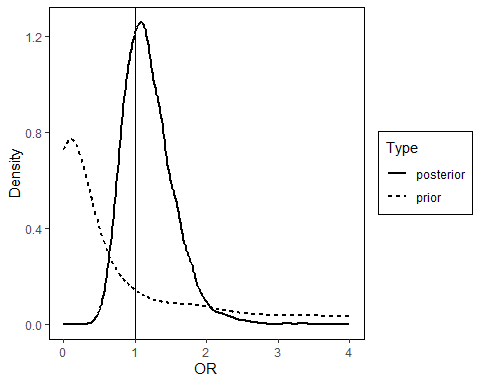
SingleTable.plot(single\_table\_obj\_samling, type = 'overlay')



set xlim between 0 to 4 and add vertical line at x = 1

plot\_obj <- SingleTable.plot(single\_table\_obj\_samling, type = 'overlay',xlim = c(0,4), add\_vertical = 1)  
plot\_obj

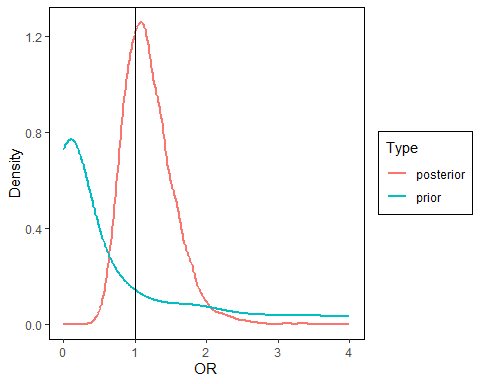
## Warning: Removed 3276 rows containing missing values (`geom\_line()`).



By color instead of line type

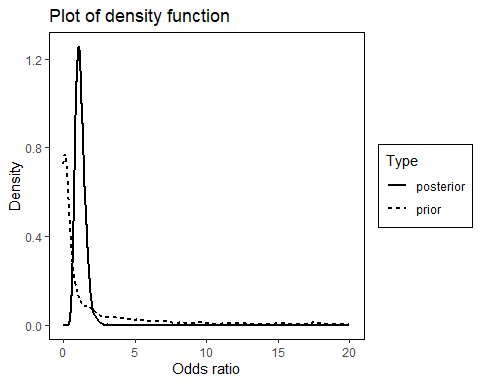
plot\_obj <- SingleTable.plot(single\_table\_obj\_samling, type = 'overlay',  
 xlim = c(0,4),   
 add\_vertical = 1,   
 by = 'color')  
plot\_obj

## Warning: Removed 3276 rows containing missing values (`geom\_line()`).



Override the xlab and add title via ggplot2

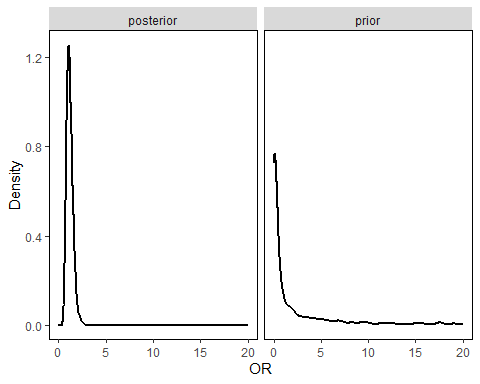
plot\_obj <- SingleTable.plot(single\_table\_obj\_samling, type = 'overlay')  
plot\_obj + xlab('Odds ratio') + ggtitle("Plot of density function")



### Plot side-by-side

default

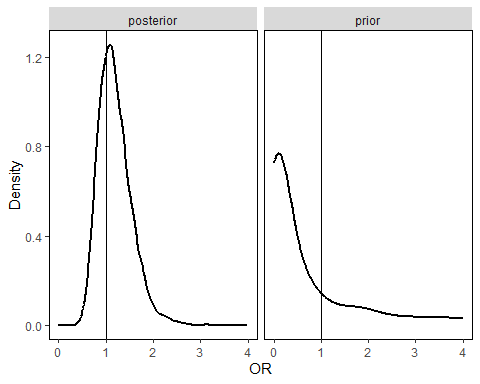
SingleTable.plot(single\_table\_obj\_samling, type = 'side\_by\_side')



set xlim between 0 to 4 and add vertical line at x = 1

SingleTable.plot(single\_table\_obj\_samling, type = 'side\_by\_side',  
 xlim = c(0,4), add\_vertical = 1)

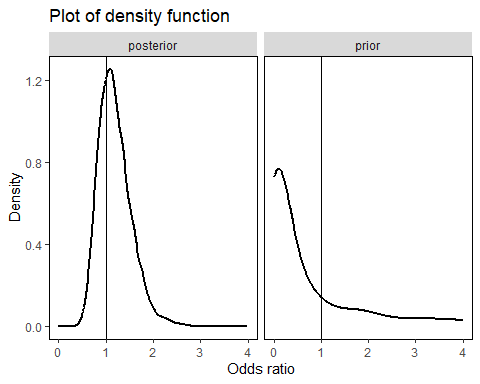
## Warning: Removed 3276 rows containing missing values (`geom\_line()`).



override xlab and add title via ggplot2

plot\_obj <- SingleTable.plot(single\_table\_obj\_samling, type = 'side\_by\_side',  
 xlim = c(0,4), add\_vertical = 1)  
  
plot\_obj + xlab('Odds ratio') + ggtitle("Plot of density function")

## Warning: Removed 3276 rows containing missing values (`geom\_line()`).



## Test eaxct method

### Create object

* specify data (y1, n1, y2, n2)
* specify parameters (a1, b1, a2, b2, rho)
* specify model (Sarmanov/Independent)
* Specify measure(OR/RR/RD)

single\_table\_obj\_exact <- SingleTable.create(a1=0.5, b1=0.5,   
 a2=0.5, b2=0.5,  
 rho=0.5, y1=40, n1=96, y2=49, n2=109,  
 model="Sarmanov",measure="OR")

### model fit

default

single\_table\_obj\_exact <- SingleTable.modelFit(single\_table\_obj\_exact, method = 'exact')

control list can be specified to control the fitting process: \* n\_samples: number of posterior samples; Defualt is 5000.

* mcmc\_initial: initial values for (p1, p2) in MCMC; Default is c(0.5, 0.5)
* upper\_bound: upper bound for the measure. Default is 100.
* lower\_bound: lower bound for the measure. For RD, default is -1. For RR/OR, defualt is 0.
* num\_grids: number of grids to calculate density; The defualt is 20498.

set number of posterior samples as 3000 (default is 5000)

single\_table\_obj\_exact <- SingleTable.modelFit(single\_table\_obj\_exact, method = 'exact',   
 control = list(n\_sample = 3000))

set initial values for MCMC is c(0.2, 0,4) (default is c(0.5,0.5))

single\_table\_obj\_exact <- SingleTable.modelFit(single\_table\_obj\_exact, method = 'exact',  
 control = list(mcmc\_initial = c(0.2,0.4)))

set upper bound for the measure is 20( default is 100)

single\_table\_obj\_exact <- SingleTable.modelFit(single\_table\_obj\_exact, method = 'exact',   
 control = list(upper\_bound = 20))

### Summary

single\_table\_obj\_exact <- SingleTable.summary(single\_table\_obj\_exact, alpha = 0.05)

## Measure: Odds ratio   
## Model: Sarmanov Beta-Binomial Model   
## Posterior distribution:   
## Mean: 1.188   
## Median: 1.149   
## 95% ET CI: [0.658,2.028]   
## 95% HDR CI: [0.580,1.885]

change alpha level to 0.1

single\_table\_obj\_exact <- SingleTable.summary(single\_table\_obj\_exact, alpha = 0.1)

## Measure: Odds ratio   
## Model: Sarmanov Beta-Binomial Model   
## Posterior distribution:   
## Mean: 1.188   
## Median: 1.149   
## 90% ET CI: [0.708,1.853]   
## 90% HDR CI: [0.625,1.717]

change digit to 2

single\_table\_obj\_exact <- SingleTable.summary(single\_table\_obj\_exact, digit = 2)

## Measure: Odds ratio   
## Model: Sarmanov Beta-Binomial Model   
## Posterior distribution:   
## Mean: 1.19   
## Median: 1.15   
## 95% ET CI: [0.66,2.03]   
## 95% HDR CI: [0.58,1.89]

not print output

single\_table\_obj\_exact <- SingleTable.summary(single\_table\_obj\_exact, verbose = FALSE)

## Structure of SingleTable object

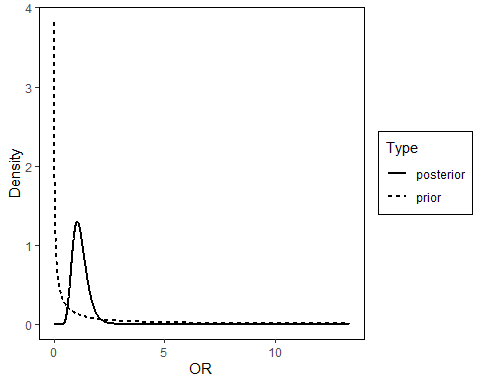
str(single\_table\_obj\_exact)

## List of 10  
## $ parms\_prior:List of 5  
## ..$ a1 : num 0.5  
## ..$ b1 : num 0.5  
## ..$ a2 : num 0.5  
## ..$ b2 : num 0.5  
## ..$ rho: num 0.5  
## $ data :List of 4  
## ..$ y1: num 40  
## ..$ n1: num 96  
## ..$ y2: num 49  
## ..$ n2: num 109  
## $ model : chr "Sarmanov"  
## $ measure : chr "OR"  
## $ method : chr "exact"  
## $ density :List of 2  
## ..$ posterior:List of 2  
## .. ..$ y: num [1:2048] 0.00 1.51e-63 1.31e-49 9.87e-42 2.45e-36 ...  
## .. ..$ x: num [1:2048] 0 0.00651 0.01302 0.01953 0.02604 ...  
## ..$ prior :List of 2  
## .. ..$ y: num [1:2048] 3.82 3.14 2.58 2.2 1.92 ...  
## .. ..$ x: num [1:2048] 0 0.00651 0.01302 0.01953 0.02604 ...  
## $ samples :List of 2  
## ..$ posterior: Named num [1:5000] 1.06 0.983 1.574 1.485 2.169 ...  
## .. ..- attr(\*, "names")= chr [1:5000] "" "" "" "" ...  
## ..$ prior : Named num [1:4170] 6.53e-05 6.53e-05 7.43e-05 2.67e-05 2.67e-05 ...  
## .. ..- attr(\*, "names")= chr [1:4170] "" "" "" "" ...  
## $ alpha : num 0.05  
## $ summary :List of 2  
## ..$ posterior:List of 4  
## .. ..$ mean : num 1.19  
## .. ..$ median: num 1.15  
## .. ..$ ET\_CI : Named num [1:2] 0.658 2.028  
## .. .. ..- attr(\*, "names")= chr [1:2] "2.5%" "97.5%"  
## .. ..$ HDR\_CI: Named num [1:2] 0.58 1.89  
## .. .. ..- attr(\*, "names")= chr [1:2] "0.66%" "95.66%"  
## ..$ prior :List of 4  
## .. ..$ mean : num 1  
## .. ..$ median: num 0.347  
## .. ..$ ET\_CI : Named num [1:2] 1.25e-04 1.60e+01  
## .. .. ..- attr(\*, "names")= chr [1:2] "2.5%" "97.5%"  
## .. ..$ HDR\_CI: Named num [1:2] 5.34e-07 1.25e+01  
## .. .. ..- attr(\*, "names")= chr [1:2] "0.02403846%" "95.02404%"  
## $ digit : num 3  
## - attr(\*, "class")= chr "SingleTable"

## overlay density:

default

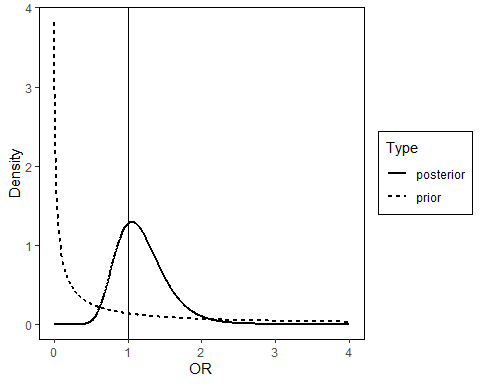
SingleTable.plot(single\_table\_obj\_exact, type = 'overlay')



set xlim between 0 to 4 and add vertical line at x = 1

plot\_obj <- SingleTable.plot(single\_table\_obj\_exact, type = 'overlay',xlim = c(0,4), add\_vertical = 1)  
plot\_obj

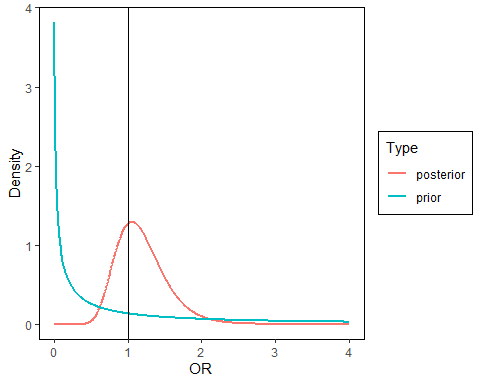
## Warning: Removed 2866 rows containing missing values (`geom\_line()`).



By color instead of line type

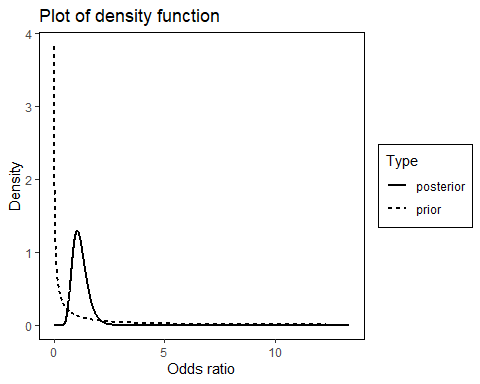
plot\_obj <- SingleTable.plot(single\_table\_obj\_exact, type = 'overlay',  
 xlim = c(0,4),   
 add\_vertical = 1,   
 by = 'color')  
plot\_obj

## Warning: Removed 2866 rows containing missing values (`geom\_line()`).



Override the xlab and add title via ggplot2

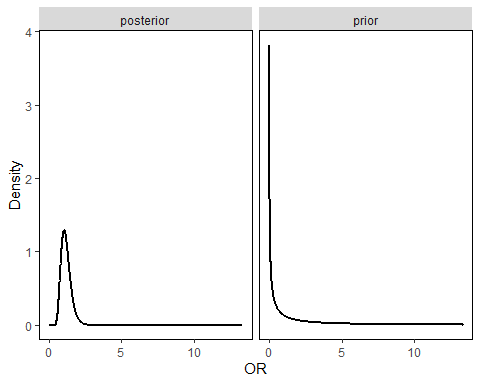
plot\_obj <- SingleTable.plot(single\_table\_obj\_exact, type = 'overlay')  
plot\_obj + xlab('Odds ratio') + ggtitle("Plot of density function")



### Plot side-by-side

default

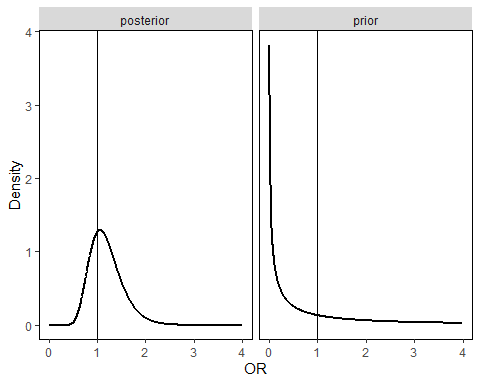
SingleTable.plot(single\_table\_obj\_exact, type = 'side\_by\_side')



set xlim between 0 to 4 and add vertical line at x = 1

SingleTable.plot(single\_table\_obj\_exact, type = 'side\_by\_side',  
 xlim = c(0,4), add\_vertical = 1)

## Warning: Removed 2866 rows containing missing values (`geom\_line()`).



override xlab and add title via ggplot2

plot\_obj <- SingleTable.plot(single\_table\_obj\_exact, type = 'side\_by\_side',  
 xlim = c(0,4), add\_vertical = 1)  
  
plot\_obj + xlab('Odds ratio') + ggtitle("Plot of density function")

## Warning: Removed 2866 rows containing missing values (`geom\_line()`).

