

Group level false positives using exGaussian distributions

Guillaume A. Rousselet

2019-04-23

Contents

Define Miller's ex-Gaussian parameters	2
Illustrate level 2 distributions	2
Most skewed	2
Least skewed	4
Summary figure	6
Illustrate bias	7
Bias simulation	7
Illustrate results	9
Group simulation 1: same number of trials in each condition	18
Simulation	18
Simulation: add medians of means (m.md)	20
Compute false positive probability	21
Illustrate results	21
Group simulation 2: one condition = always 200 trials, other condition n trials	28
Illustrate level 2 distributions	28
Simulation	32
Compute false positive probability	34
Illustrate results	34
Medians of means simulation	39
Means of medians simulation, with bias correction	41
Group simulation 3: shift function	45
Results from most skewed distribution	47
Results from least skewed distribution	50

```
# dependencies
# install.packages("devtools")
# devtools::install_github("GRousselet/rogme")
library(rogme)
library(ggplot2)
library(tibble)
library(tidyr)
library(cowplot)
library(retimes)
source("../functions/tests.txt")
source("../functions/skew.txt")
source("../functions/make_figures.txt")
source("../functions/functions.txt")
library(beepr)
```

```
sessionInfo()
```

```
## R version 3.5.2 (2018-12-20)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.4
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_GB.UTF-8/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-8/en_GB.UTF-8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] beeper_1.3      retimes_0.1-2  cowplot_0.9.4  tidyr_0.8.2    rogme_0.2.1
## [6] tibble_2.0.1    ggplot2_3.1.0
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.0      pillar_1.3.1    compiler_3.5.2  plyr_1.8.4
## [5] tools_3.5.2     digest_0.6.18   evaluate_0.13    gtable_0.2.0
## [9] pkgconfig_2.0.2  rlang_0.3.1     yaml_2.2.0      xfun_0.4
## [13] withr_2.1.2     dplyr_0.8.0.1   stringr_1.4.0    knitr_1.21
## [17] grid_3.5.2      tidyselect_0.2.5 glue_1.3.0       R6_2.4.0
## [21] rmarkdown_1.11  purrr_0.3.0     magrittr_1.5     scales_1.0.0
## [25] htmltools_0.3.6 assertthat_0.2.0 colorspace_1.4-0 stringi_1.3.1
## [29] lazyeval_0.2.1  munsell_0.5.0   crayon_1.3.4     audio_0.1-5.1
```

We consider the proportion of false positives in a typical hierarchical situation: for each participant and each condition, first, the mean of the reaction times is computed, second, the distributions of individual means are compared using a t-test on means (one-sample t-test on the pairwise differences). We also consider situations in which we use group means of individual medians, and group medians of individual medians.

Define Miller's ex-Gaussian parameters

```
load('./data/miller_exg_param.RData')
```

Illustrate level 2 distributions

We look at the distribution of 10,000 group means of individual differences between means computed for 50 trials per condition. We do the same for medians of medians. The distributions of differences are fairly symmetric because we sample trials from the same population for the two conditions and there is no between-participant variability.

Most skewed

```

set.seed(1)
P <- 1
mu <- miller.param[P,1]
sigma <- miller.param[P,2]
tau <- miller.param[P,3]

nt1 <- 50
nt2 <- 50
np <- 10000

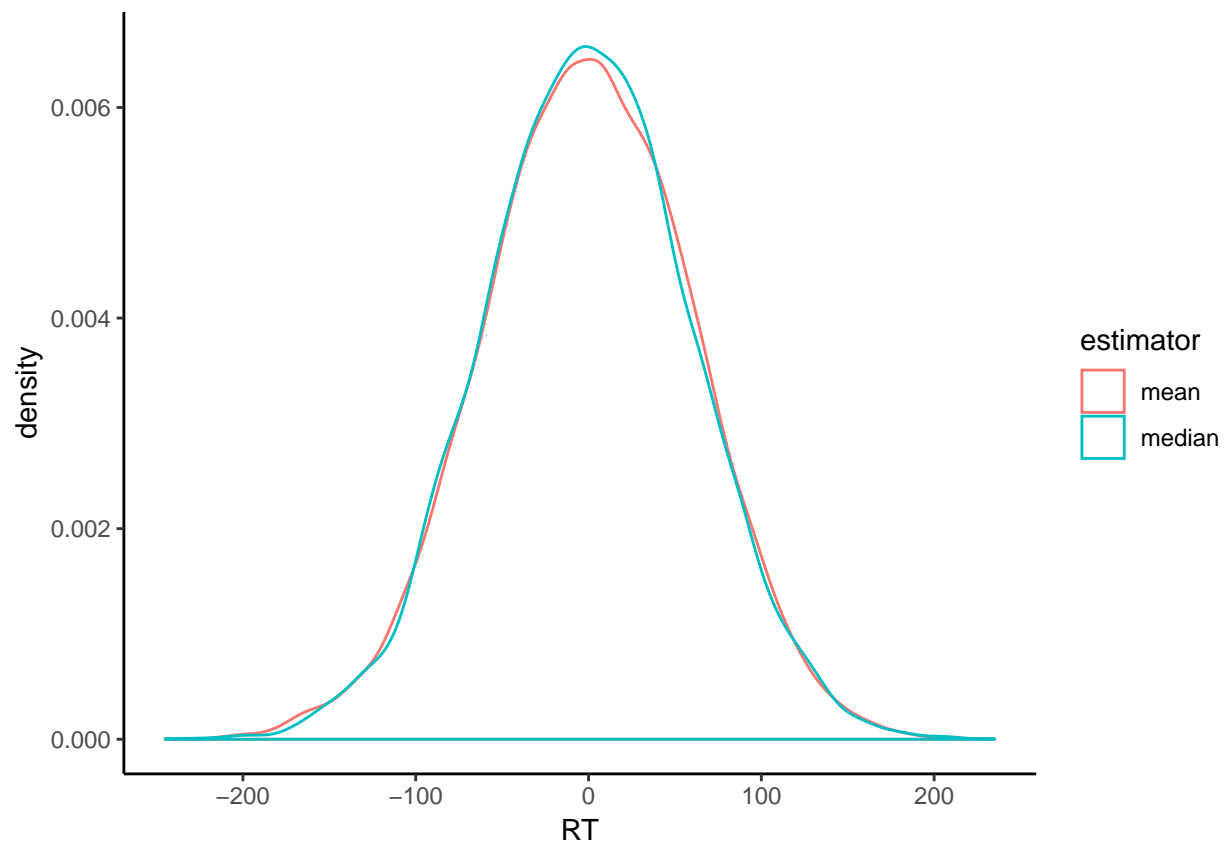
mc.data1 <- array(rexgauss(nt1*np, mu = mu, sigma = sigma, tau = tau),
  dim = c(nt1, np))

mc.data2 <- array(rexgauss(nt2*np, mu = mu, sigma = sigma, tau = tau),
  dim = c(nt2, np))

mc.m <- apply(mc.data1, 2, mean) - apply(mc.data2, 2, mean) # diff in mean RTs
mc.md <- apply(mc.data1, 2, median) - apply(mc.data2, 2, median) # diff in median RTs
df <- tibble(RT = c(mc.m, mc.md),
  estimator = factor(c(rep("mean", np), rep("median", np))))

ggplot(df, aes(x = RT, colour = estimator)) + theme_classic() +
  geom_density()

```



T-tests on individual means and medians.

The confidence intervals are slightly asymmetric.

```
t.test(mc.m)

##
## One Sample t-test
##
## data: mc.m
## t = -0.59046, df = 9999, p-value = 0.5549
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -1.5425916 0.8283943
## sample estimates:
## mean of x
## -0.3570986
```

```
t.test(mc.md)

##
## One Sample t-test
##
## data: mc.md
## t = -1.2034, df = 9999, p-value = 0.2288
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -1.8873011 0.4514656
## sample estimates:
## mean of x
## -0.7179178
```

```
skew(mc.m)

## [1] -0.06538814
```

```
skew(mc.md)

## [1] -0.01143766
```

Median tests on individual means and medians.

```
sint(mc.m)

## [1] -1.430277 1.684391
sint(mc.md)

## [1] -2.1318024 0.8625607
```

Least skewed

Sampling distributions of the mean and median differences are much narrower than for the most skewed ExGaussian distribution considered previously.

```
set.seed(1)
```

```

P <- 12
mu <- miller.param[P,1]
sigma <- miller.param[P,2]
tau <- miller.param[P,3]

nt1 <- 50
nt2 <- 50
np <- 10000

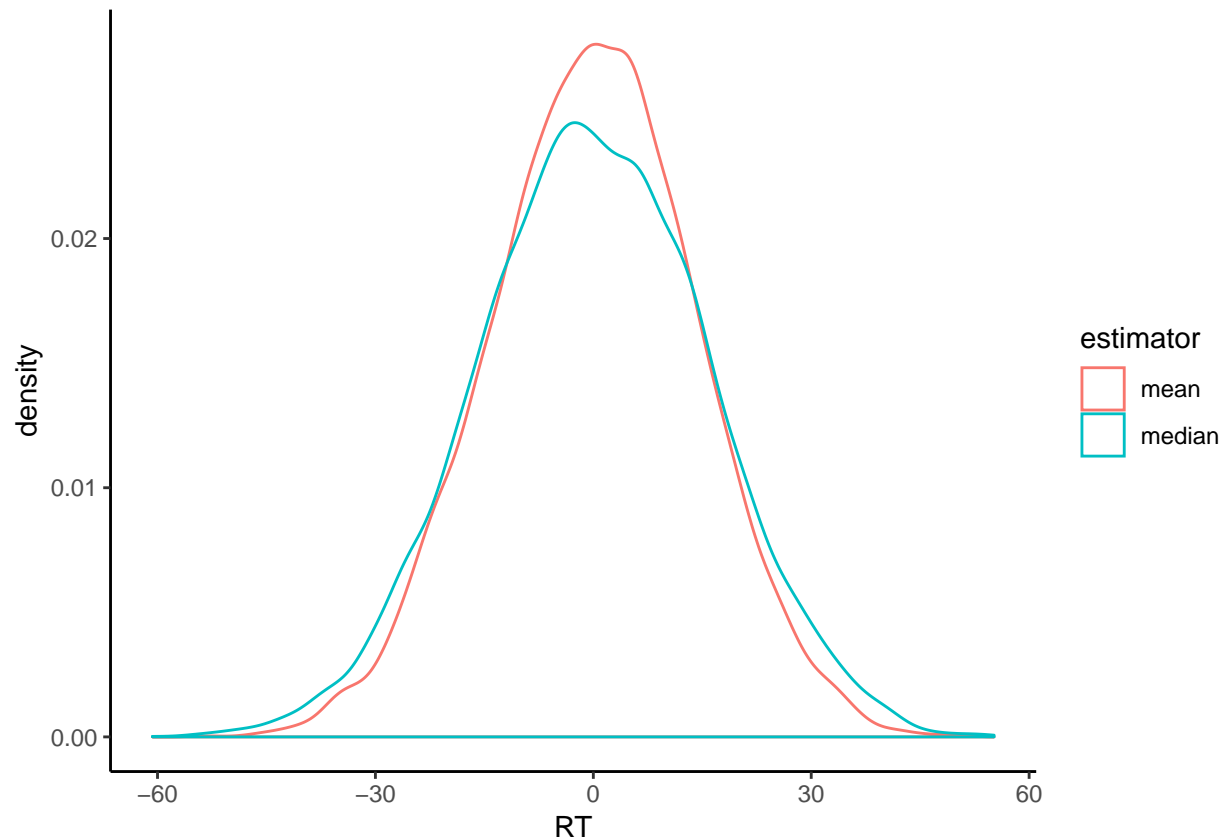
mc.data1 <- array(rexgauss(nt1*np, mu = mu, sigma = sigma, tau = tau),
  dim = c(nt1, np))

mc.data2 <- array(rexgauss(nt2*np, mu = mu, sigma = sigma, tau = tau),
  dim = c(nt2, np))

mc.m <- apply(mc.data1, 2, mean) - apply(mc.data2, 2, mean) # diff in mean RTs
mc.md <- apply(mc.data1, 2, median) - apply(mc.data2, 2, median) # diff in median RTs
df <- tibble(RT = c(mc.m, mc.md),
  estimator = factor(c(rep("mean", np), rep("median", np))))

ggplot(df, aes(x = RT, colour = estimator)) + theme_classic() +
  geom_density()

```



Summary figure

Simulate data

```
set.seed(1)

nt1 <- 50
nt2 <- 50
np <- 10000

res.m <- matrix(NA, nrow = np, ncol = 12)
res.md <- matrix(NA, nrow = np, ncol = 12)
res.skew.m <- vector(mode = "numeric", length = 12)
res.skew.md <- vector(mode = "numeric", length = 12)

for(P in 1:12){
  mu <- miller.param[P,1]
  sigma <- miller.param[P,2]
  tau <- miller.param[P,3]

  mc.data1 <- array(rexgauss(nt1*np, mu = mu, sigma = sigma, tau = tau),
    dim = c(nt1, np))

  mc.data2 <- array(rexgauss(nt2*np, mu = mu, sigma = sigma, tau = tau),
    dim = c(nt2, np))

  res.m[,P] <- apply(mc.data1, 2, mean) - apply(mc.data2, 2, mean) # diff in mean RTs
  res.md[,P] <- apply(mc.data1, 2, median) - apply(mc.data2, 2, median) # diff in mean RTs
  res.skew.m[P] <- skew(res.m[,P])
  res.skew.md[P] <- skew(res.md[,P])
}
```

Skewness of the sampling distribution of the mean:

```
res.skew.m

## [1] -0.065388136 -0.018775380 -0.002198078 -0.028612998 0.009845933
## [6] -0.010909888 0.017146988 0.026971370 -0.007506182 -0.023491977
## [11] 0.018004395 -0.045363639
```

Skewness of the sampling distribution of the median:

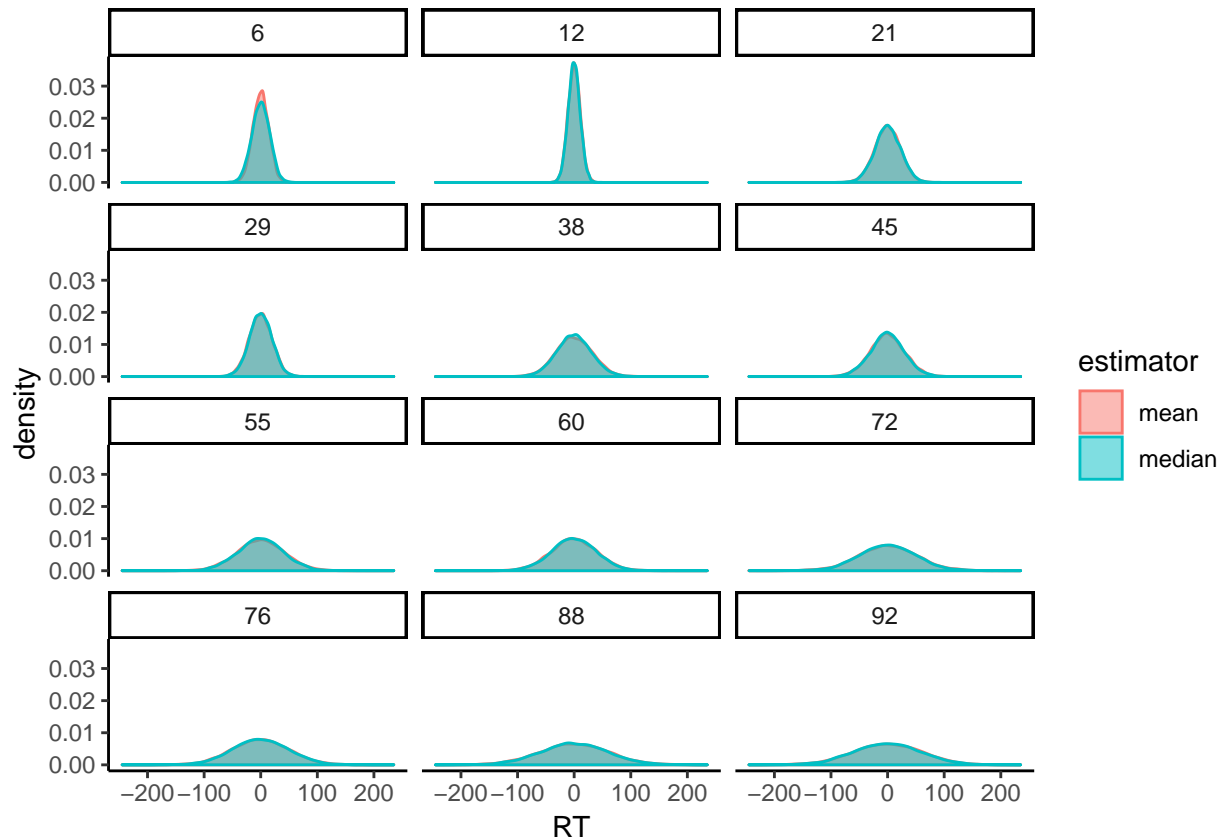
```
res.skew.md

## [1] -0.0114376642 0.0019442886 0.0032896307 -0.0154732866 0.0198097427
## [6] -0.0006038556 0.0208246471 0.0225231510 -0.0178030831 -0.0117130661
## [11] 0.0371126657 -0.0399725534
```

Make figure

```
df <- tibble(RT = c(as.vector(res.m), as.vector(res.md)),
              Skewness = factor(rep(rep(round(pop.m - pop.md), each = np), 2)),
              estimator = factor(c(rep("mean", np*12), rep("median", np*12))))

ggplot(df, aes(x = RT, colour = estimator, fill = estimator)) + theme_classic() +
  geom_density(alpha = 0.5) +
  facet_wrap(vars(Skewness), nrow = 4)
```



Illustrate bias

Before considering false positives, here we look at hierarchical bias: we save the group means and group medians of individual differences in means and medians (4 combinations of differences). The results are very similar to those obtained in the `bias_diff` notebook. This is not surprising because in these simulations we assume that all participants have the same effects. More realistic results, incorporating between-participant variability, are presented in the `flp` notebooks.

Bias simulation

```
npseq <- c(25, 50, 100, 200)
maxnp <- max(npseq) # max number of participants
ntseq1 <- c(seq(10, 100, 10), 150, 200)
maxnt1 <- max(ntseq1) # max number of trials
```

```

ntseq2 <- rep(200, length(ntseq1)) # number of trials in group 2
maxnt2 <- max(ntseq2) # max number of trials
nsim <- 10000 # simulation samples

# declare matrices of results
res.m.m.bias <- array(NA, dim = c(nP, length(ntseq1), length(npseq)))
res.m.md.bias <- array(NA, dim = c(nP, length(ntseq1), length(npseq)))
res.md.m.bias <- array(NA, dim = c(nP, length(ntseq1), length(npseq)))
res.md.md.bias <- array(NA, dim = c(nP, length(ntseq1), length(npseq)))

res.m.m.mdbias <- array(NA, dim = c(nP, length(ntseq1), length(npseq)))
res.m.md.mdbias <- array(NA, dim = c(nP, length(ntseq1), length(npseq)))
res.md.m.mdbias <- array(NA, dim = c(nP, length(ntseq1), length(npseq)))
res.md.md.mdbias <- array(NA, dim = c(nP, length(ntseq1), length(npseq)))

set.seed(21)

for(P in 1:nP){ # ex-Gaussian parameters
  beep(2)
  # generate all data at once
  print(paste0("parameters: ",P," out of ",nP,"..."))
  mu <- miller.param[P,1]
  sigma <- miller.param[P,2]
  tau <- miller.param[P,3]
  mc.data1 <- array(rexgauss(maxnt1*maxnp*nsim, mu = mu, sigma = sigma, tau = tau),
    dim = c(maxnt1, maxnp, nsim))
  mc.data2 <- array(rexgauss(maxnt2*maxnp*nsim, mu = mu, sigma = sigma, tau = tau),
    dim = c(maxnt2, maxnp, nsim))

  for(TR in 1:length(ntseq1)){ # number of trials
    # print(paste0("number of trials: ",ntseq[TR],"..."))

    for(PT in 1:length(npseq)){ # number of participants

      # compute estimates
      todo1 <- mc.data1[1:ntseq1[TR], 1:npseq[PT],]
      todo2 <- mc.data2[1:ntseq2[TR], 1:npseq[PT],]

      # diff in mean RTs
      mc.m <- apply(todo1, c(2,3), mean) - apply(todo2, c(2,3), mean)

      # diff in median RTs
      mc.md <- apply(todo1, c(2,3), median) - apply(todo2, c(2,3), median)

      # MEAN BIAS -----
      # group means
      res.m.m.bias[P, TR, PT] <- mean(apply(mc.m, 2, mean))
      res.md.m.bias[P, TR, PT] <- mean(apply(mc.md, 2, mean))
      # group medians
      res.m.m.bias[P, TR, PT] <- mean(apply(mc.m, 2, median))
      res.md.md.bias[P, TR, PT] <- mean(apply(mc.md, 2, median))

      # MEDIAN BIAS -----

```



```

    # group means
    res.m.m.mdbias[P, TR, PT] <- median(apply(mc.m, 2, mean))
    res.md.m.mdbias[P, TR, PT] <- median(apply(mc.md, 2, mean))
    # group medians
    res.m.md.mdbias[P, TR, PT] <- median(apply(mc.m, 2, median))
    res.md.md.mdbias[P, TR, PT] <- median(apply(mc.md, 2, median))
  }
}
save(
  res.m.m.bias,
  res.m.md.bias,
  res.md.m.bias,
  res.md.md.bias,
  res.m.m.mdbias,
  res.m.md.mdbias,
  res.md.m.mdbias,
  res.md.md.mdbias,
  ntseq1,
  ntseq2,
  npseq,
  nsim,
  file=paste0('./data/sim_gp_fp2_bias_sameskew.RData'))
beep(8)

```

Illustrate results

Get data

No need to compute bias, as it should be zero in the long run for differences. In the simulation we already averaged over 10,000 simulations, so here we simply load these averages.

```

load('./data/sim_gp_fp2_bias.RData')
ntseq <- ntseq1
xlabel <- c("10", "", "30", "", "50", "", "70", "", "90", "", "150", "200")

```

Make function

```

make_bias_figure <- function(data, ntseq, nP, npseq, pop.m, pop.md){
  df <- tibble(`FP`=as.vector(data),
    `Skewness`=rep(round(pop.m - pop.md),length(ntseq)*length(npseq)),
    `Trials`=rep(rep(ntseq,each=nP), length(npseq)),
    `Participants`=rep(npseq,each=nP*length(ntseq))
  )

  df$Skewness <- as.character(df$Skewness)
  df$Skewness <- factor(df$Skewness, levels=unique(df$Skewness))

  df$Participants <- as.character(df$Participants)
  df$Participants <- factor(df$Participants, levels=unique(df$Participants))
}

```

```

labels <- c("25" = "25 participants", "50" = "50 participants",
           "100" = "100 participants", "200" = "200 participants")

# make plot
p <- ggplot(df, aes(x=Trials, y=FP, colour = Skewness)) + theme_classic() +
  geom_line(size = 1) +
  geom_abline(intercept=0, slope=0, colour="black") +
  scale_colour_viridis_d() +
  scale_x_continuous(breaks=ntseq, labels = xlabels) +
  coord_cartesian(ylim=c(-10,15)) +
  theme(plot.title = element_text(size=22),
        axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 12, colour="black"),
        axis.text.y = element_text(size = 16, colour="black"),
        axis.title.y = element_text(size = 18),
        legend.key.width = unit(1.5,"cm"),
        legend.position = c(0.45,0.75),
        legend.direction = "horizontal",
        legend.text=element_text(size=16),
        legend.title=element_text(size=18),
        strip.text = element_text(size=18, face="bold"),
        strip.background = element_rect(colour="black", fill="white")) +
  labs(x = "Number of trials", y = "Mean bias (ms)") +
  guides(colour = guide_legend(override.aes = list(size=3))) + # make thicker legend lines
  facet_grid(cols = vars(Participants), labeller=labeller(Participants = labels))
p
}

```

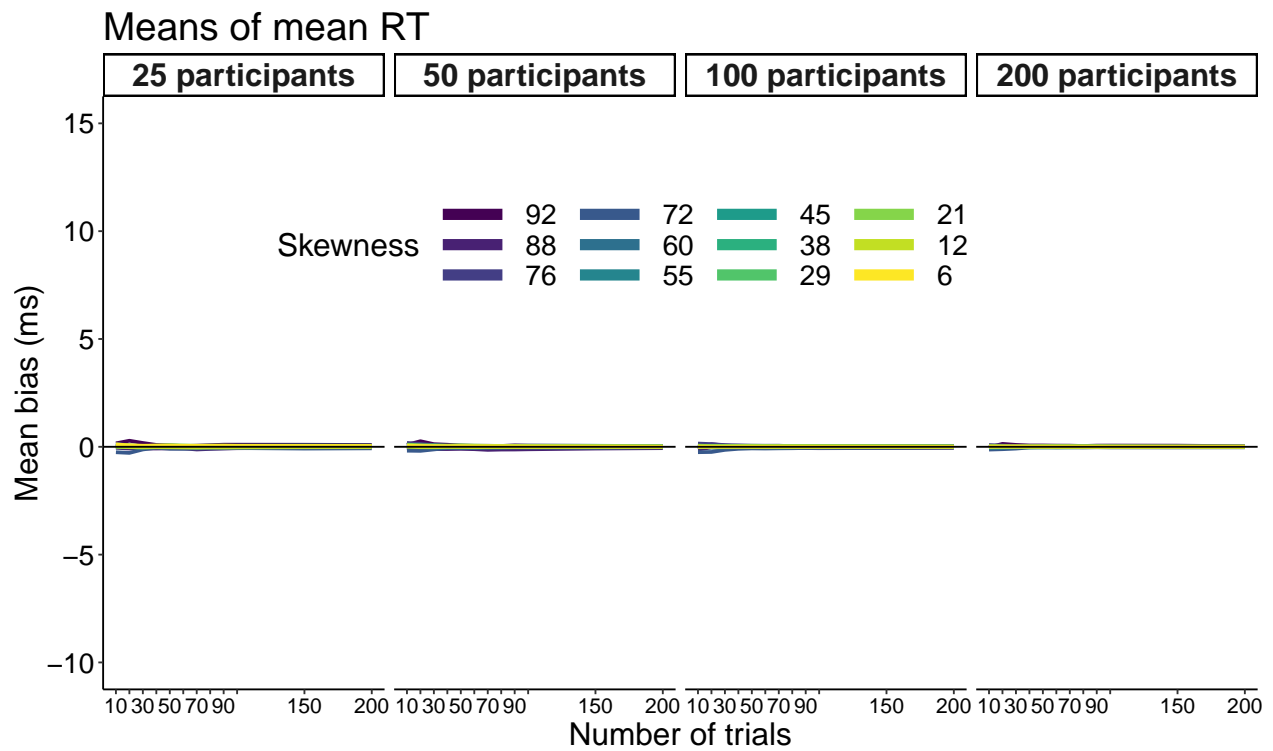
Mean bias

Group means of means

```

p <- make_bias_figure(res.m.m.bias, ntseq, nP, npseq, pop.m, pop.md) +
  ggtitle("Means of mean RT")
p

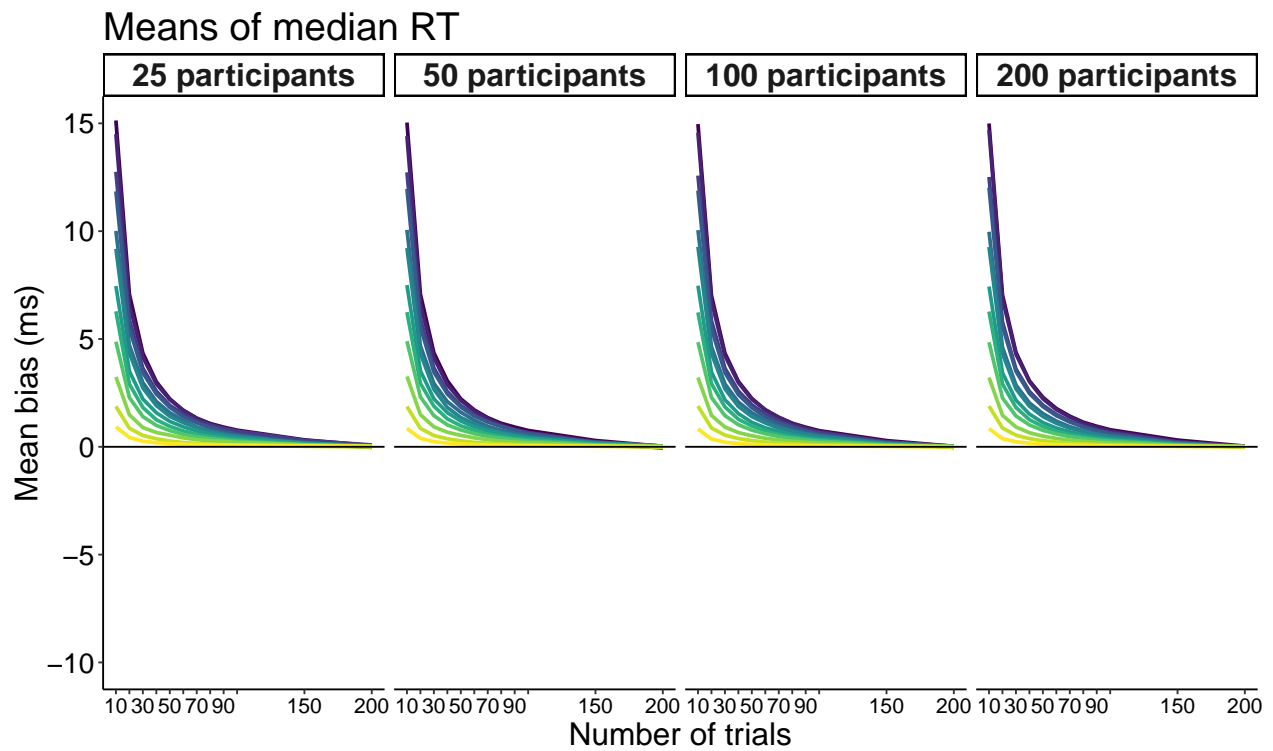
```



```
pm.m <- p
```

Group means of medians

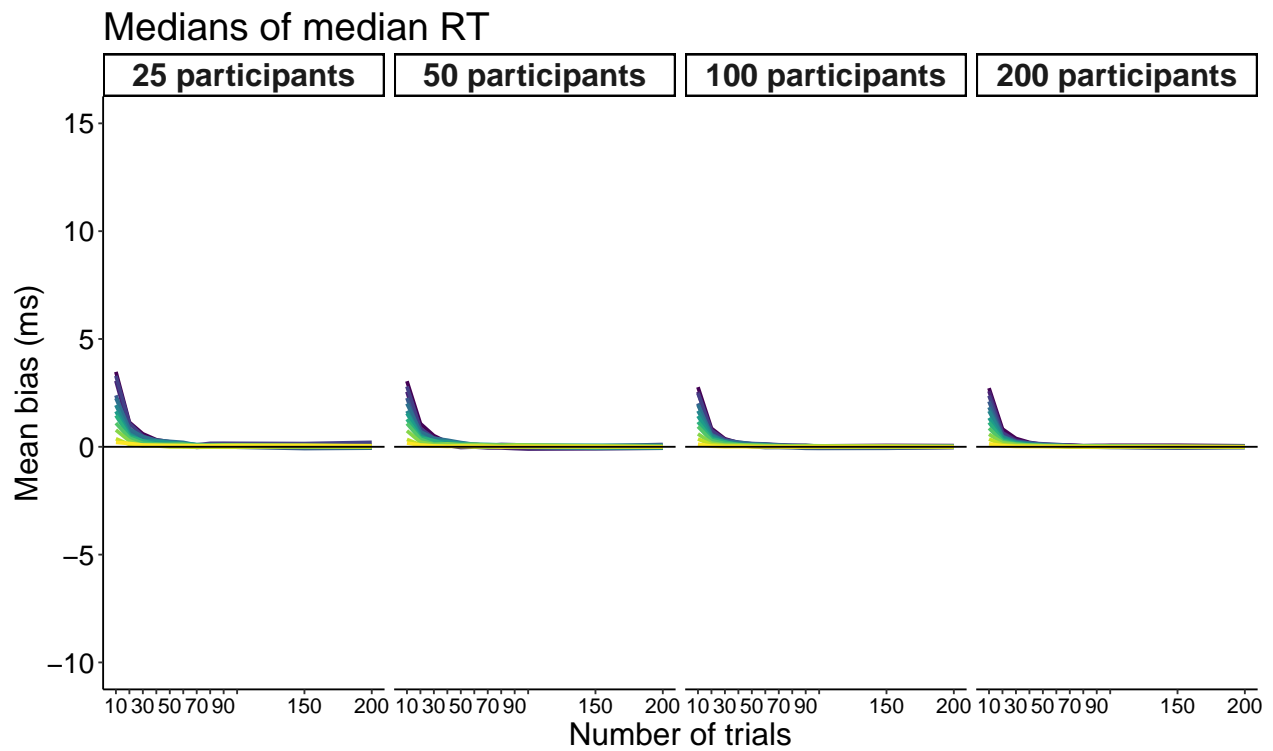
```
p <- make_bias_figure(res.md.m.bias, ntseq, nP, npseq, pop.m, pop.md) +
  theme(legend.position = "none") +
  ggtitle("Means of median RT")
p
```



```
pmd.m <- p
```

Group medians of medians

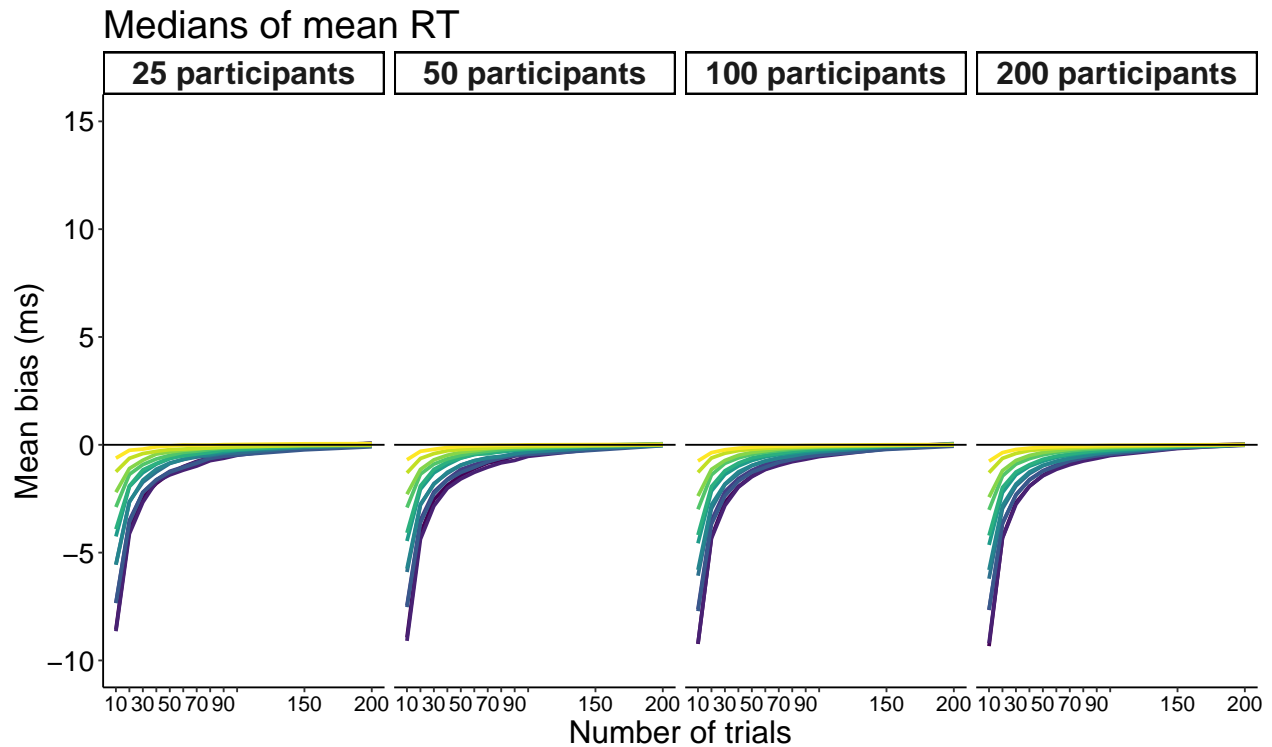
```
p <- make_bias_figure(res.md.md.bias, ntseq, nP, npseq, pop.m, pop.md) +  
  theme(legend.position = "none") +  
  ggtitle("Medians of median RT")  
p
```



```
pmd.md <- p
```

Group medians of means

```
p <- make_bias_figure(res.m.md.bias, ntseq, nP, npseq, pop.m, pop.md) +  
  theme(legend.position = "none") +  
  ggtitle("Medians of mean RT")  
p
```



```
pm.md <- p
```

Summary figure

```
# combine panels into one figure
cowplot::plot_grid(pm.m, pmd.m, pmd.md, pm.md,
  labels = c("A", "B", "C", "D"),
  ncol = 1,
  nrow = 4,
  rel_widths = c(1, 1, 1, 1),
  label_size = 20,
  hjust = -0.5,
  scale=.95,
  align = "h")

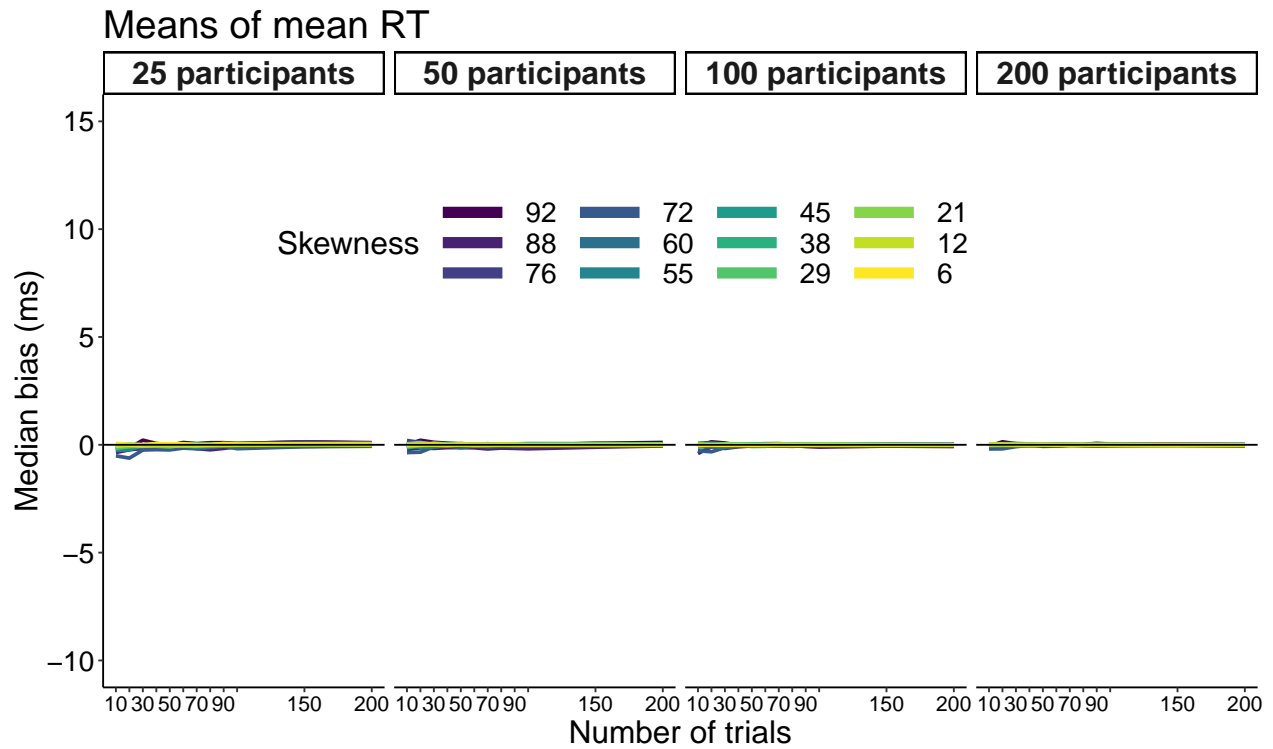
# save figure
ggsave(filename='./figures/figure_sim_gp_fp2_bias.pdf', width=10, height=15)
```

Median bias

Median bias results are very similar to the mean bias results because the sampling distributions are fairly symmetric.

Group means of means

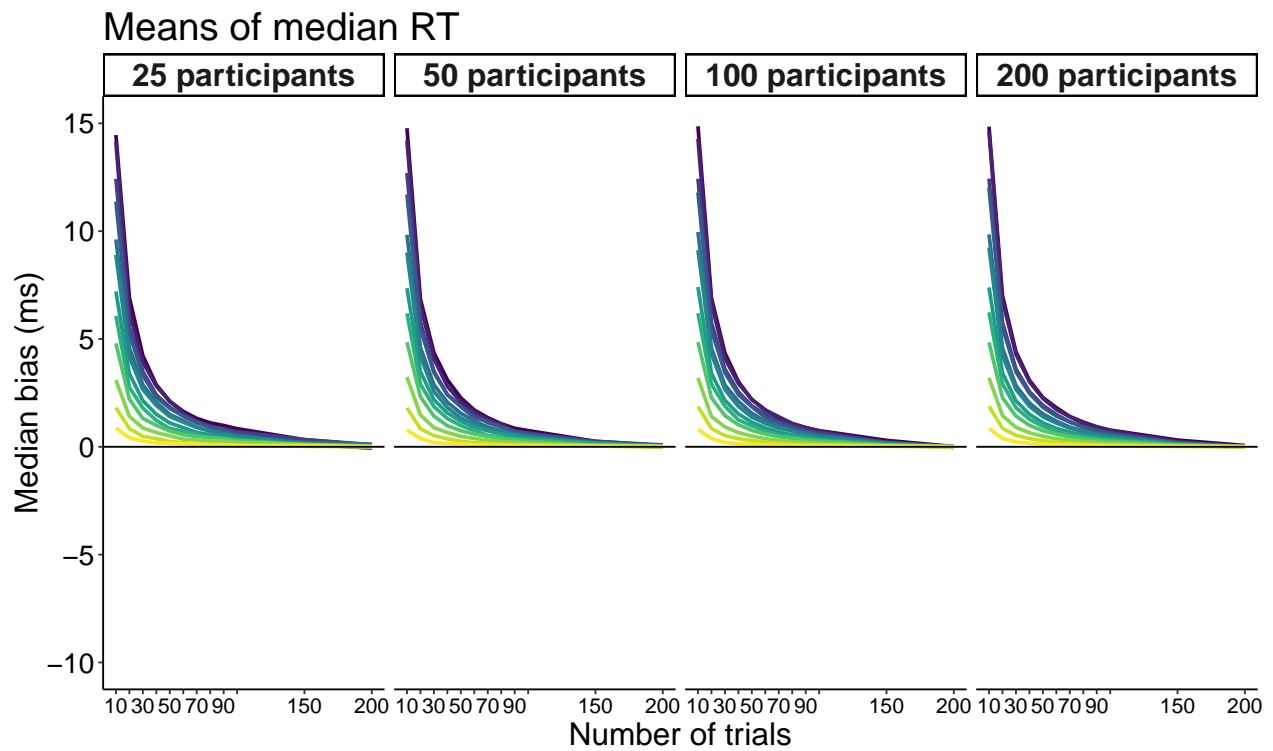
```
p <- make_bias_figure(res.m.m.mdbias, ntseq, nP, npseq, pop.m, pop.md) +
  labs(y = "Median bias (ms)") +
  ggtitle("Means of mean RT")
p
```



```
pm.m.mdb <- p
```

Group means of medians

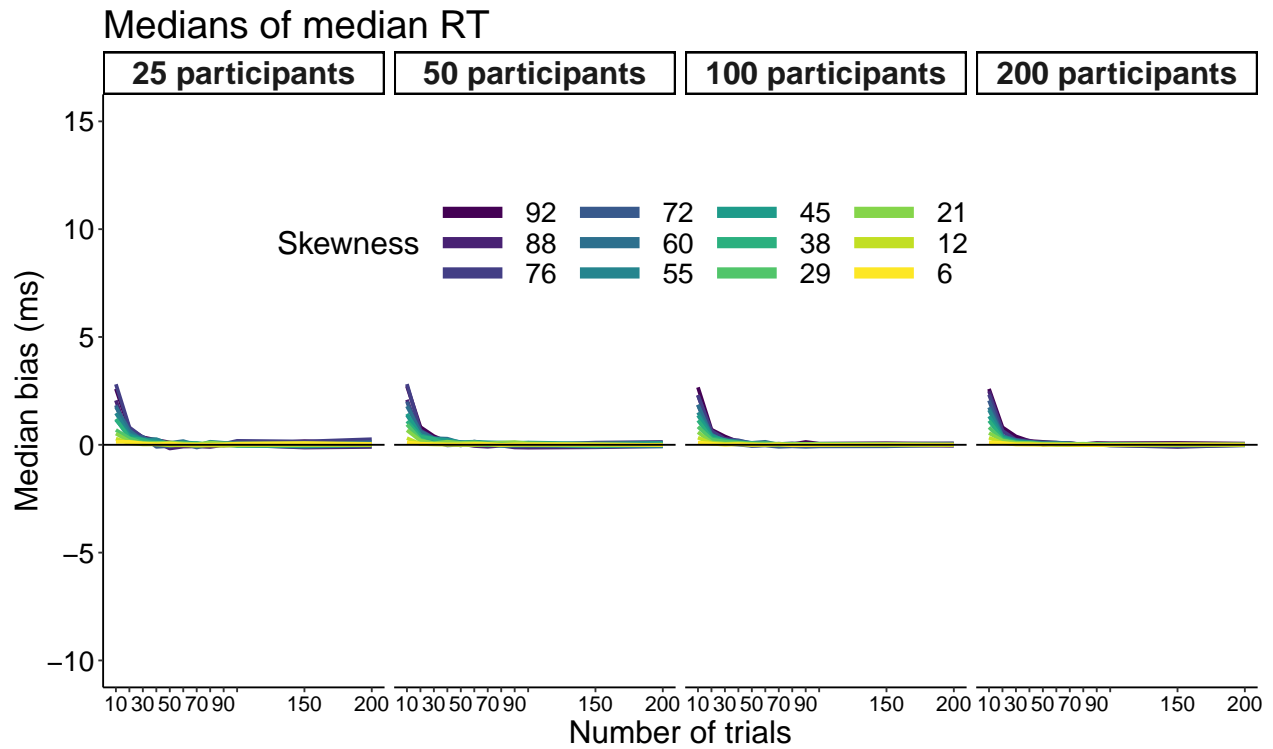
```
p <- make_bias_figure(res.md.m.mdbias, ntseq, nP, npseq, pop.m, pop.md) +
  theme(legend.position = "none") +
  labs(y = "Median bias (ms)") +
  ggtitle("Means of median RT")
p
```



```
pmd.m.mdb <- p
```

Group medians of medians

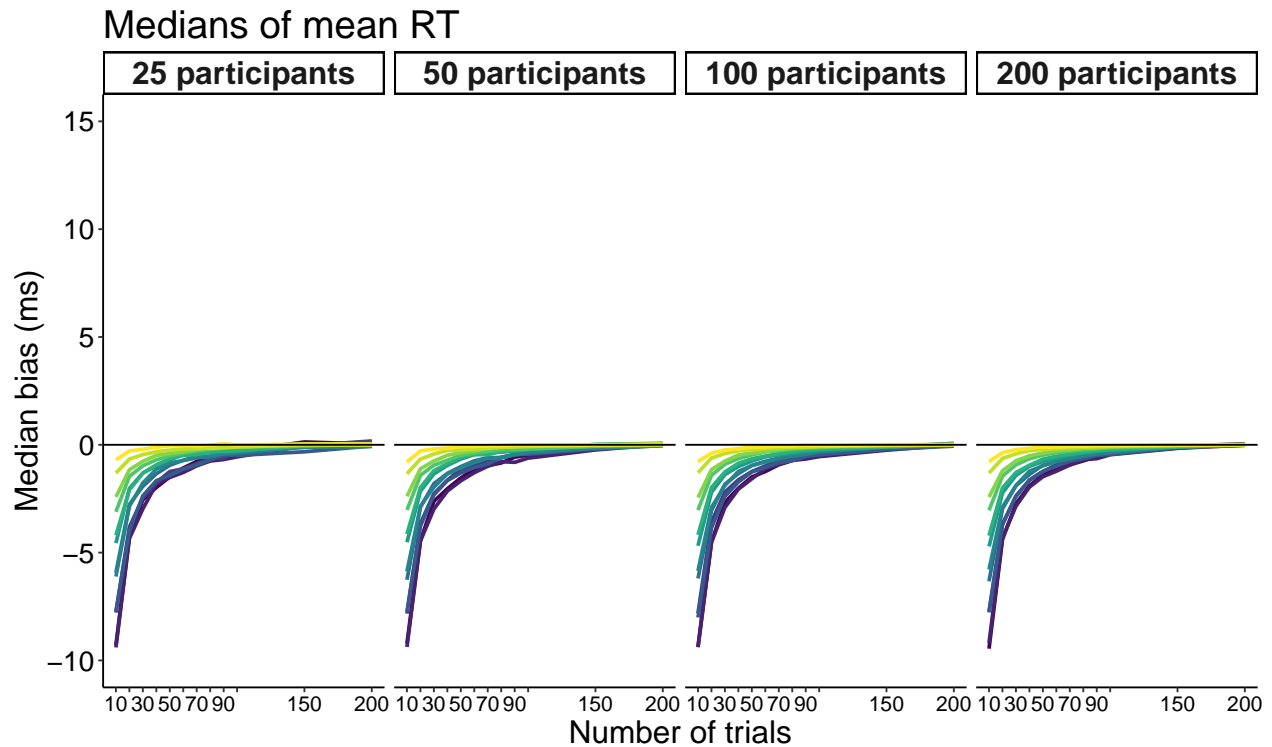
```
p <- make_bias_figure(res.md.md.mdbias, ntseq, nP, npseq, pop.m, pop.md) +  
  labs(y = "Median bias (ms)") +  
  ggtitle("Medians of median RT")  
p
```

```
pmd.md.mdb <- p
```

Group medians of means

```
p <- make_bias_figure(res.m.md.mdbias, ntseq, nP, npseq, pop.m, pop.md) +
  theme(legend.position = "none") +
  labs(y = "Median bias (ms)") +
  ggtitle("Medians of mean RT")
p
```



```
pm.md.mdb <- p
```

Summary figure

```
# combine panels into one figure
cowplot::plot_grid(pm.m.mdb, pmd.m.mdb, pmd.md.mdb, pm.md.mdb,
  labels = c("A", "B", "C", "D"),
  ncol = 1,
  nrow = 4,
  rel_widths = c(1, 1, 1, 1),
  label_size = 20,
  hjust = -0.5,
  scale=.95,
  align = "h")

# save figure
ggsave(filename='./figures/figure_sim_gp_fp2_mdbias.pdf', width=10, height=12)
```

Group simulation 1: same number of trials in each condition

10,000 random samples for each of the 12 distributions: - n trials = c(seq(10, 100, 10), 150, 200) - n participants = c(25, 50, 100, 200)

Contrast: - group t-test on means - group median tests on medians

Simulation

```

npseq <- c(25, 50, 100, 200)
ntseq <- c(seq(10, 100, 10), 150, 200)
maxnp <- max(npseq) # max number of participants
maxnt <- max(ntseq) # max number of trials
nsim <- 10000 # simulation samples

# declare matrices of results - save all iterations
res.m.m.sig <- array(NA, dim = c(nsim, nP, length(ntseq), length(npseq)))
res.md.m.sig <- array(NA, dim = c(nsim, nP, length(ntseq), length(npseq)))
res.md.md.sig <- array(NA, dim = c(nsim, nP, length(ntseq), length(npseq)))
res.q1.md.sig <- array(NA, dim = c(nsim, nP, length(ntseq), length(npseq)))
res.q3.md.sig <- array(NA, dim = c(nsim, nP, length(ntseq), length(npseq)))

set.seed(21)

for(P in 1:nP){ # ex-Gaussian parameters

  # generate all data at once
  print(paste0("parameters: ",P," out of ",nP,"..."))
  mu <- miller.param[P,1]
  sigma <- miller.param[P,2]
  tau <- miller.param[P,3]
  mc.data1 <- array(rexgauss(maxnt*maxnp*nsim, mu = mu, sigma = sigma, tau = tau),
    dim = c(maxnt, maxnp, nsim))
  mc.data2 <- array(rexgauss(maxnt*maxnp*nsim, mu = mu, sigma = sigma, tau = tau),
    dim = c(maxnt, maxnp, nsim))

  for(TR in 1:length(ntseq)){ # number of trials
    # print(paste0("number of trials: ",ntseq[TR],"..."))

    for(PT in 1:length(npseq)){ # number of participants

      # compute estimates
      todo1 <- mc.data1[1:ntseq[TR], 1:npseq[PT],]
      todo2 <- mc.data2[1:ntseq[TR], 1:npseq[PT],]

      # diff in mean RTs
      mc.m <- apply(todo1, c(2,3), mean) - apply(todo2, c(2,3), mean)

      # diff in quartile RTs
      qres <- apply(todo1, c(2,3), quantile, probs = c(0.25, 0.5, 0.75)) -

      # tests for each simulation =====
      # one-sample t-test
      res.m.m.sig[, P, TR, PT] <- ttest.sig(mc.m)
      res.md.m.sig[, P, TR, PT] <- ttest.sig(qres[2,,])
      # parametric median tests of the quartiles
      res.q1.md.sig[, P, TR, PT] <- apply(qres[1,,], 2, sint.sig)
      res.md.md.sig[, P, TR, PT] <- apply(qres[2,,], 2, sint.sig)
      res.q3.md.sig[, P, TR, PT] <- apply(qres[3,,], 2, sint.sig)
    }
  }
}

```

apply

```

save(
  res.m.m.sig,
  res.md.m.sig,
  res.md.md.sig,
  res.q1.md.sig,
  res.q3.md.sig,
  ntseq,
  npseq,
  nsim,
  file=paste0('./data/sim_gp_fp1.RData'))

```

Simulation: add medians of means (m.md)

```

load('./data/sim_gp_fp1.RData')
npseq <- c(25, 50, 100, 200)
ntseq <- c(seq(10, 100, 10), 150, 200)
maxnp <- max(npseq) # max number of participants
maxnt <- max(ntseq) # max number of trials
nsim <- 10000 # simulation samples

# declare matrices of results - save all iterations
res.m.md.sig <- array(NA, dim = c(nsim, nP, length(ntseq), length(npseq)))

set.seed(21)

for(P in 1:nP){ # ex-Gaussian parameters

  beep(2)
  # generate all data at once
  print(paste0("parameters: ",P," out of ",nP,"..."))
  mu <- miller.param[P,1]
  sigma <- miller.param[P,2]
  tau <- miller.param[P,3]
  mc.data1 <- array(rexgauss(maxnt*maxnp*nsim, mu = mu, sigma = sigma, tau = tau),
    dim = c(maxnt, maxnp, nsim))
  mc.data2 <- array(rexgauss(maxnt*maxnp*nsim, mu = mu, sigma = sigma, tau = tau),
    dim = c(maxnt, maxnp, nsim))

  for(TR in 1:length(ntseq)){ # number of trials
    # print(paste0("number of trials: ",ntseq[TR],"..."))

    for(PT in 1:length(npseq)){ # number of participants

      # compute estimates
      todo1 <- mc.data1[1:ntseq[TR], 1:npseq[PT],]
      todo2 <- mc.data2[1:ntseq[TR], 1:npseq[PT],]

      # diff in mean RTs
      mc.m <- apply(todo1, c(2,3), mean) - apply(todo2, c(2,3), mean)

      # tests for each simulation =====
      # median tests of the means

```

```

        res.m.md.sig[, P, TR, PT] <- apply(mc.m, 2, sint.sig)
    }
}
save(
  res.m.m.sig,
  res.md.m.sig,
  res.md.md.sig,
  res.m.md.sig,
  res.q1.md.sig,
  res.q3.md.sig,
  ntseq,
  npseq,
  nsim,
  file='./data/sim_gp_fp1.RData')
beep(8)

```

Compute false positive probability

```

load('./data/sim_gp_fp1.RData')
fp.m.m <- apply(res.m.m.sig, c(2,3,4), mean)
fp.md.m <- apply(res.md.m.sig, c(2,3,4), mean)
fp.m.md <- apply(res.m.md.sig, c(2,3,4), mean)
fp.md.md <- apply(res.md.md.sig, c(2,3,4), mean)
fp.q1.md <- apply(res.q1.md.sig, c(2,3,4), mean)
fp.q3.md <- apply(res.q3.md.sig, c(2,3,4), mean)

xlabels <- c("10", "", "30", "", "50", "", "70", "", "90", "", "150", "200")

```

Illustrate results

Make function

```

make_fp_figure <- function(data, ntseq, nP, npseq, pop.m, pop.md){
  df <- tibble(`FP`=as.vector(data),
    `Skewness`=rep(round(pop.m - pop.md),length(ntseq)*length(npseq)),
    `Trials`=rep(rep(ntseq,each=nP), length(npseq)),
    `Participants`=rep(npseq,each=nP*length(ntseq))
  )

  df$Skewness <- as.character(df$Skewness)
  df$Skewness <- factor(df$Skewness, levels=unique(df$Skewness))

  df$Participants <- as.character(df$Participants)
  df$Participants <- factor(df$Participants, levels=unique(df$Participants))

  labels <- c("25" = "25 participants", "50" = "50 participants",
    "100" = "100 participants", "200" = "200 participants")

  # make plot

```

```

p <- ggplot(df, aes(x=Trials, y=FP, colour = Skewness)) + theme_classic() +
  # Bradley's (1978) satisfactory range
  geom_ribbon(aes(x=Trials), ymin = 0.025, ymax = 0.075,
    fill = "grey85", colour = "grey85", show.legend=FALSE) +
  # 0.05 reference line
  geom_abline(intercept = 0.05, slope = 0, colour="black") +
  geom_line(size = 1) +
  geom_abline(intercept=0, slope=0, colour="black") +
  scale_colour_viridis_d() +
  scale_x_continuous(breaks=ntseq, labels = xlabel) +
  coord_cartesian(ylim=c(0,0.2)) +
  theme(plot.title = element_text(size=22),
    axis.title.x = element_text(size = 18),
    axis.text.x = element_text(size = 12, colour="black"),
    axis.text.y = element_text(size = 16, colour="black"),
    axis.title.y = element_text(size = 18),
    legend.key.width = unit(1.5,"cm"),
    legend.position = c(0.45,0.75),
    legend.direction = "horizontal",
    legend.text=element_text(size=16),
    legend.title=element_text(size=18),
    strip.text = element_text(size=18, face="bold"),
    strip.background = element_rect(colour="black", fill="white")) +
  labs(x = "Number of trials", y = "Prop. of false positives") +
  guides(colour = guide_legend(override.aes = list(size=3))) + # make thicker legend lines
  facet_grid(cols = vars(Participants), labeller=labeller(Participants = labels))
p
}

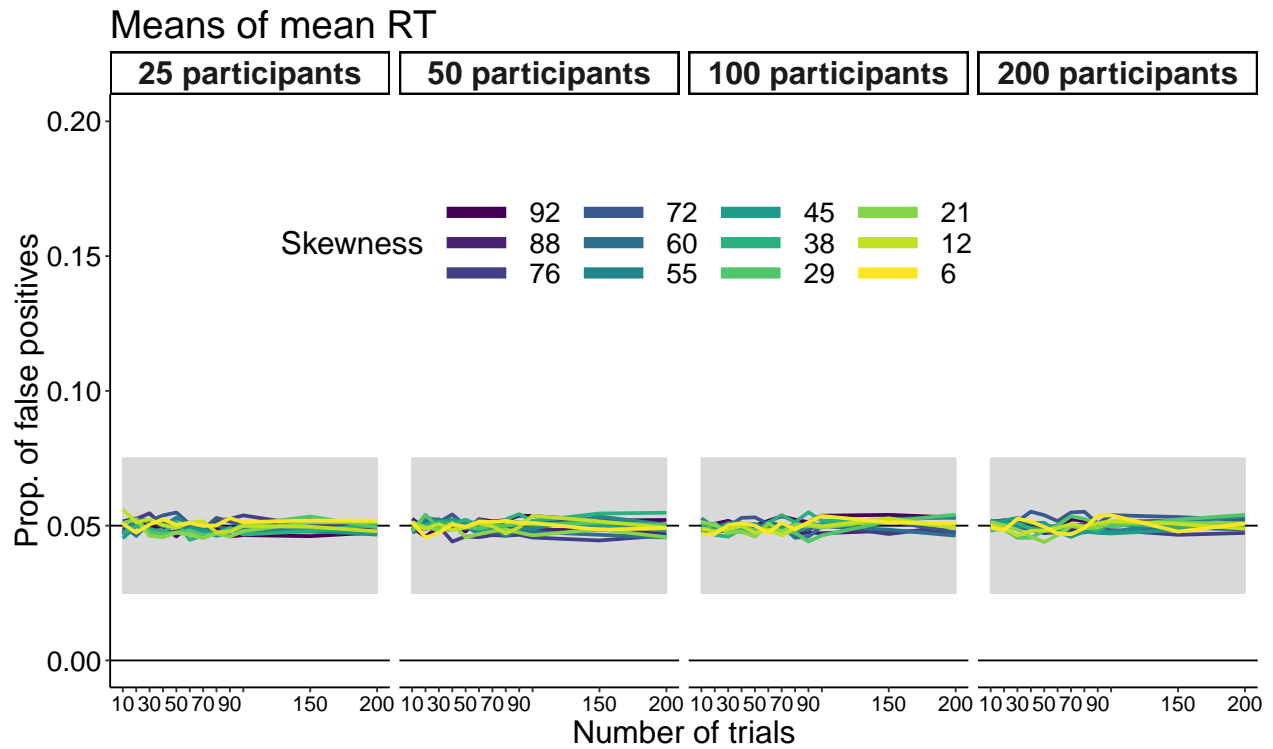
```

Group means of means

```

p <- make_fp_figure(fp.m.m, ntseq, nP, npseq, pop.m, pop.md)
p <- p + ggtitle("Means of mean RT")
p

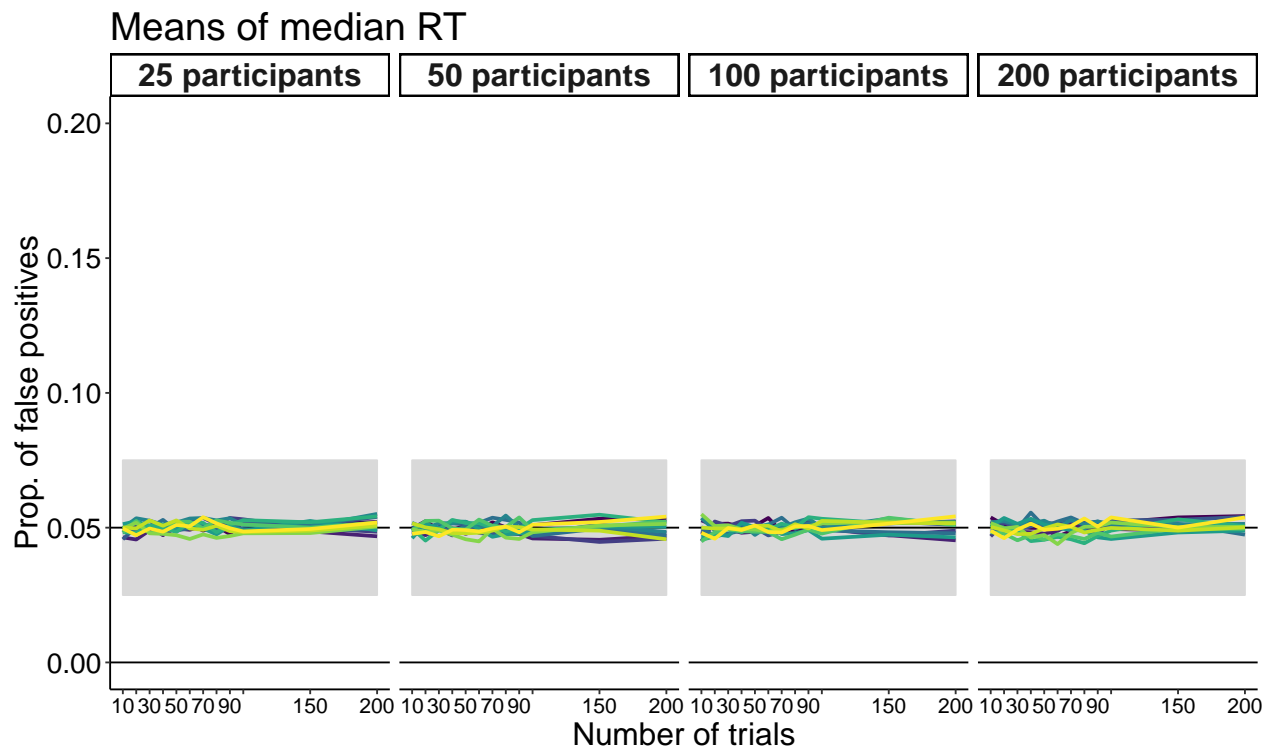
```



```
pm.m <- p
```

Group means of medians

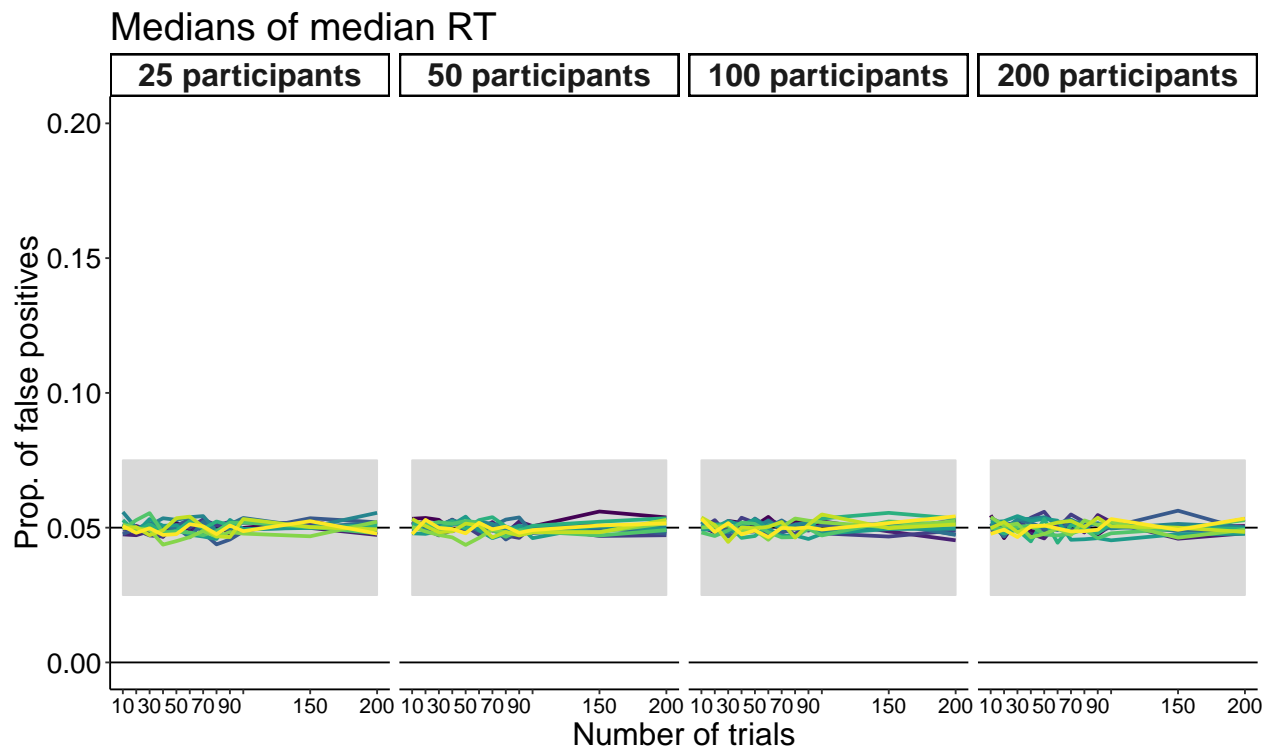
```
p <- make_fp_figure(fp.md.m, ntseq, nP, npseq, pop.m, pop.md)
p <- p + ggtitle("Means of median RT") +
  theme(legend.position = "none")
p
```



```
pmd.m <- p
```

Group medians of medians

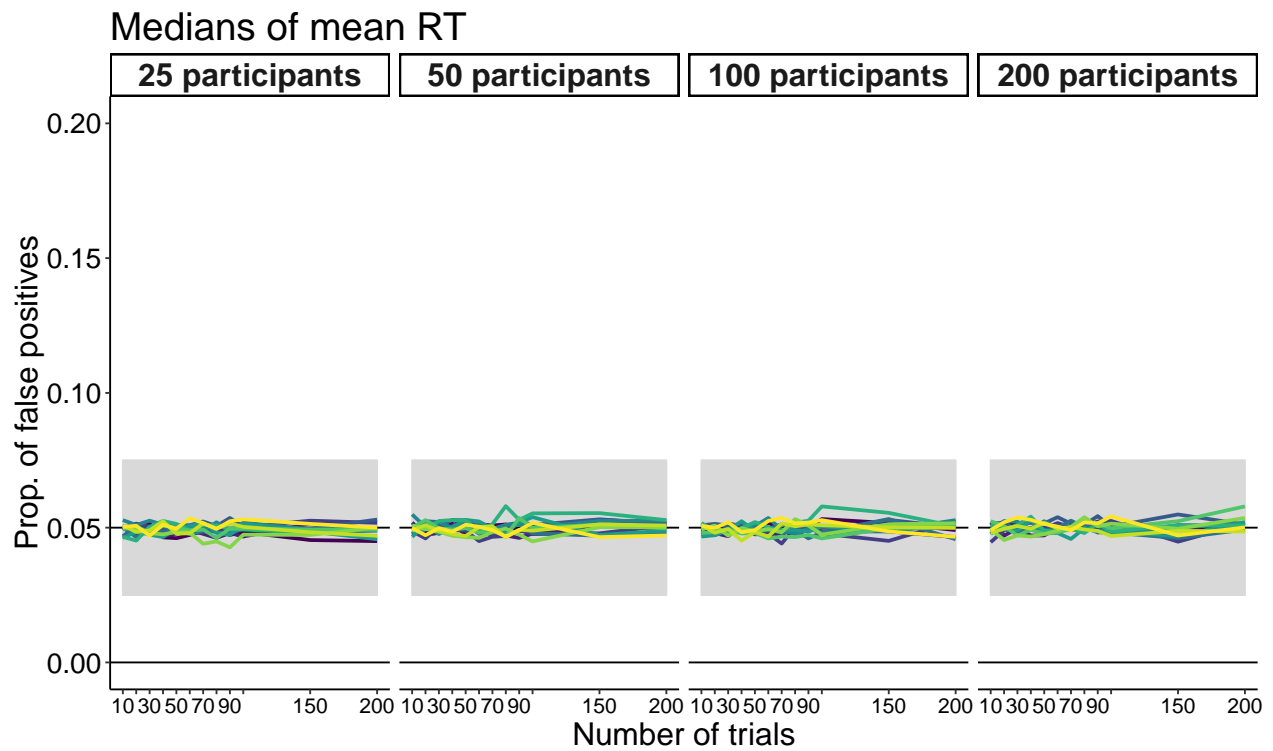
```
p <- make_fp_figure(fp.md.md, ntseq, nP, npseq, pop.m, pop.md)
p <- p + ggtitle("Medians of median RT") +
  theme(legend.position = "none")
p
```

```
pmd.md <- p
```

Group medians of means

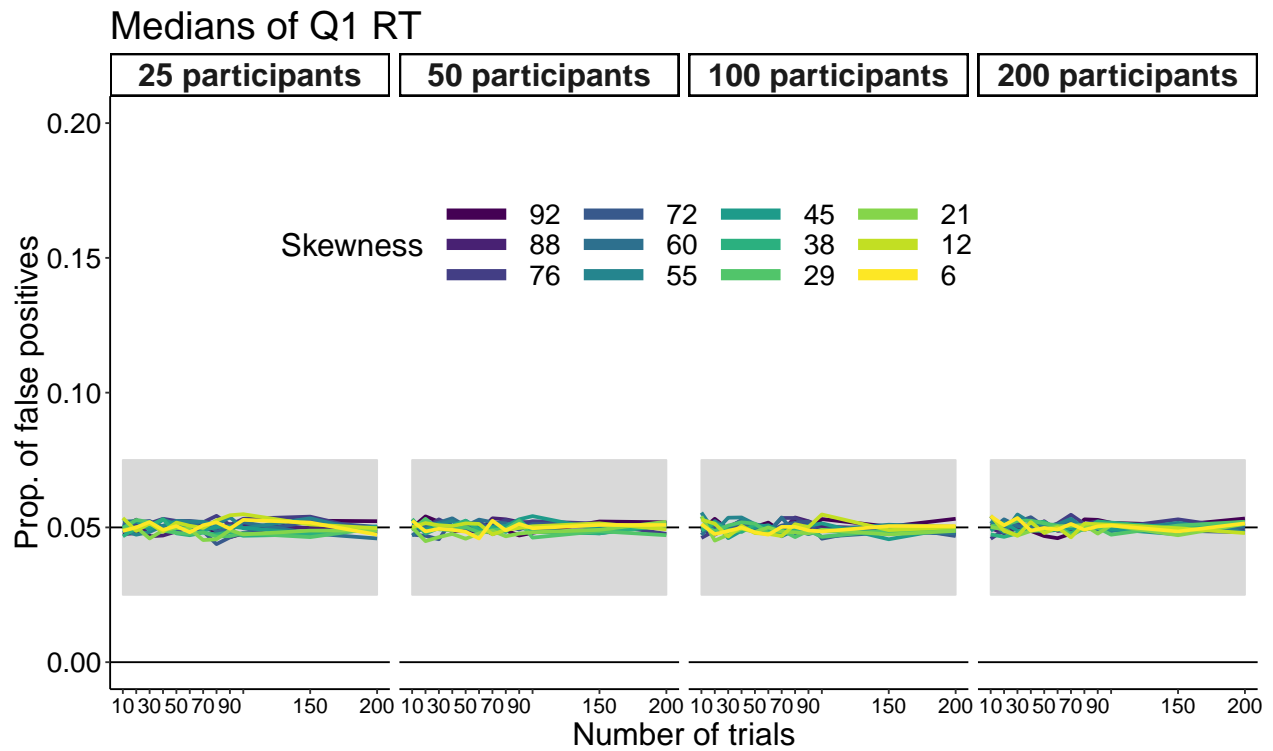
```
p <- make_fp_figure(fp.m.md, ntseq, nP, npseq, pop.m, pop.md)
p <- p + ggtitle("Medians of mean RT") +
  theme(legend.position = "none")
p
```



```
pm.md <- p
```

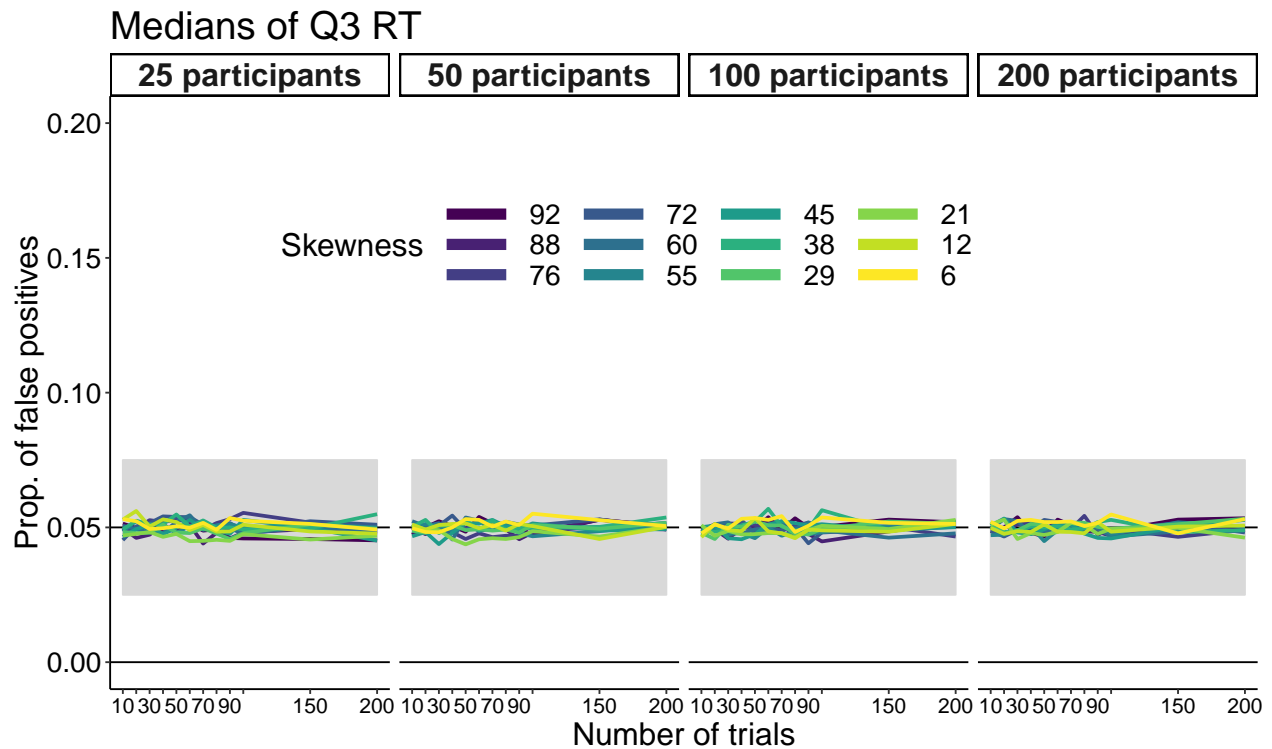
Group medians of Q1

```
p <- make_fp_figure(fp.q1.md, ntseq, nP, npseq, pop.m, pop.md)
p <- p + ggtitle("Medians of Q1 RT")
p
```



Group medians of Q3

```
p <- make_fp_figure(fp.q3.md, ntseq, nP, npseq, pop.m, pop.md)
p <- p + ggtitle("Medians of Q3 RT")
p
```



Summary figure

```
# combine panels into one figure
cowplot::plot_grid(pm.m, pmd.m, pmd.md, pm.md,
  labels = c("A", "B", "C", "D"),
  ncol = 1,
  nrow = 4,
  # rel_widths = c(1, 1, 1),
  label_size = 20,
  hjust = -0.5,
  scale=.95,
  align = "h")

# save figure
ggsave(filename='./figures/figure_sim_gp_fp1_summary.pdf',width=10,height=15)
```

Group simulation 2: one condition = always 200 trials, other condition n trials

10,000 random samples for each of the 12 distributions: - group 1: $c(\text{seq}(10, 100, 10), 150, 200)$ - group 2: 200 trials - n participants = $c(25, 50, 100, 200)$

Contrast: - group t-test on means - group median tests on medians

Illustrate level 2 distributions

The level 2 sampling distributions are positively skewed, and more so when sampling from skewed distributions.

Most skewed

```
set.seed(1)
P <- 1
mu <- miller.param[P,1]
sigma <- miller.param[P,2]
tau <- miller.param[P,3]

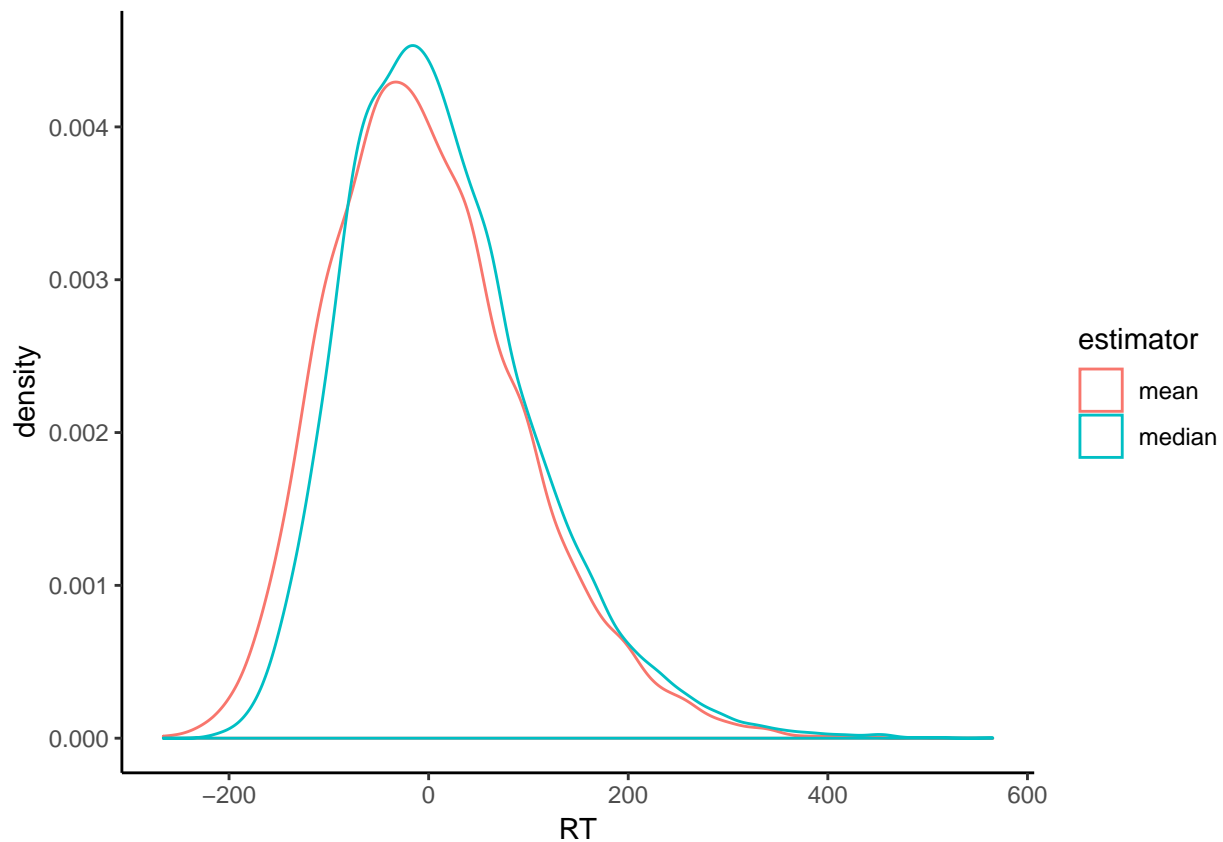
nt1 <- 10
nt2 <- 200
np <- 10000

mc.data1 <- array(rexgauss(nt1*np, mu = mu, sigma = sigma, tau = tau),
  dim = c(nt1, np))

mc.data2 <- array(rexgauss(nt2*np, mu = mu, sigma = sigma, tau = tau),
  dim = c(nt2, np))

mc.m <- apply(mc.data1, 2, mean) - apply(mc.data2, 2, mean) # diff in mean RTs
mc.md <- apply(mc.data1, 2, median) - apply(mc.data2, 2, median) # diff in median RTs
df <- tibble(RT = c(mc.m, mc.md),
  estimator = factor(c(rep("mean", np), rep("median", np))))

ggplot(df, aes(x = RT, colour = estimator)) + theme_classic() +
  geom_density()
```



```
t.test(mc.m)
```

```
##  
## One Sample t-test  
##  
## data: mc.m  
## t = -0.78925, df = 9999, p-value = 0.43  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -2.663275 1.134254  
## sample estimates:  
## mean of x  
## -0.7645106
```

```
t.test(mc.md)
```

```
##  
## One Sample t-test  
##  
## data: mc.md  
## t = 14.654, df = 9999, p-value < 2.2e-16  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 12.04165 15.76055  
## sample estimates:  
## mean of x  
## 13.9011
```

```
skew(mc.m)
```

```
## [1] 0.5841374
```

```
skew(mc.md)
```

```
## [1] 0.8226243
```

```
sint(mc.m)
```

```
## [1] -13.231584 -8.124321
```

```
sint(mc.md)
```

```
## [1] -0.4856476 3.7219334
```

Least skewed

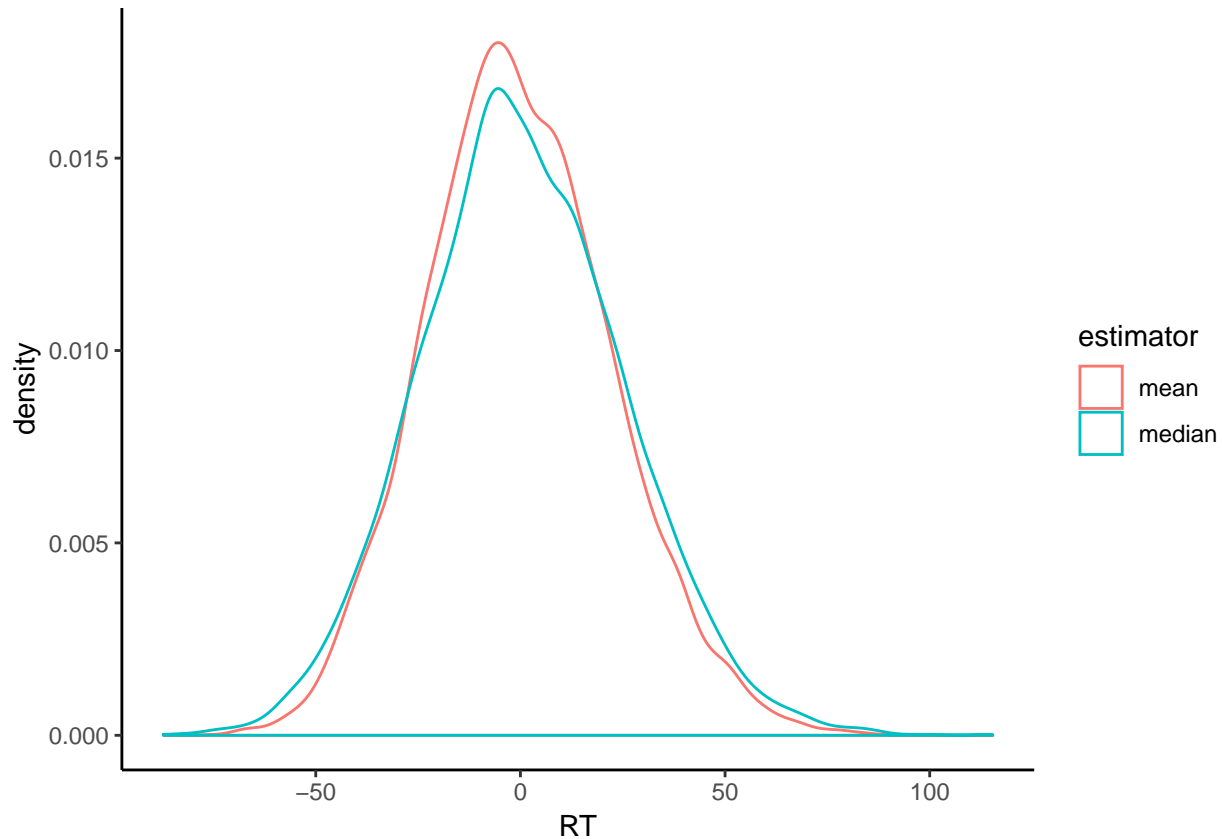
```
set.seed(1)
```

```
P <- 12  
mu <- miller.param[P,1]  
sigma <- miller.param[P,2]  
tau <- miller.param[P,3]  
  
nt1 <- 10  
nt2 <- 200  
np <- 10000  
  
mc.data1 <- array(rexgauss(nt1*np, mu = mu, sigma = sigma, tau = tau),  
dim = c(nt1, np))
```

```
mc.data2 <- array(rexgauss(nt2*np, mu = mu, sigma = sigma, tau = tau),
  dim = c(nt2, np))

mc.m <- apply(mc.data1, 2, mean) - apply(mc.data2, 2, mean) # diff in mean RTs
mc.md <- apply(mc.data1, 2, median) - apply(mc.data2, 2, median) # diff in median RTs
df <- tibble(RT = c(mc.m, mc.md),
  estimator = factor(c(rep("mean", np), rep("median", np))))

ggplot(df, aes(x = RT, colour = estimator)) + theme_classic() +
  geom_density()
```



Summary figure

Simulate data

```
set.seed(1)

nt1 <- 10
nt2 <- 200
np <- 10000

res.m <- matrix(NA, nrow = np, ncol = 12)
res.md <- matrix(NA, nrow = np, ncol = 12)

for(P in 1:12){
  mu <- miller.param[P,1]
```

```

sigma <- miller.param[P,2]
tau <- miller.param[P,3]

mc.data1 <- array(rexgauss(nt1*np, mu = mu, sigma = sigma, tau = tau),
  dim = c(nt1, np))

mc.data2 <- array(rexgauss(nt2*np, mu = mu, sigma = sigma, tau = tau),
  dim = c(nt2, np))

res.m[,P] <- apply(mc.data1, 2, mean) - apply(mc.data2, 2, mean) # diff in mean RTs
res.md[,P] <- apply(mc.data1, 2, median) - apply(mc.data2, 2, median) # diff in median RTs
}

```

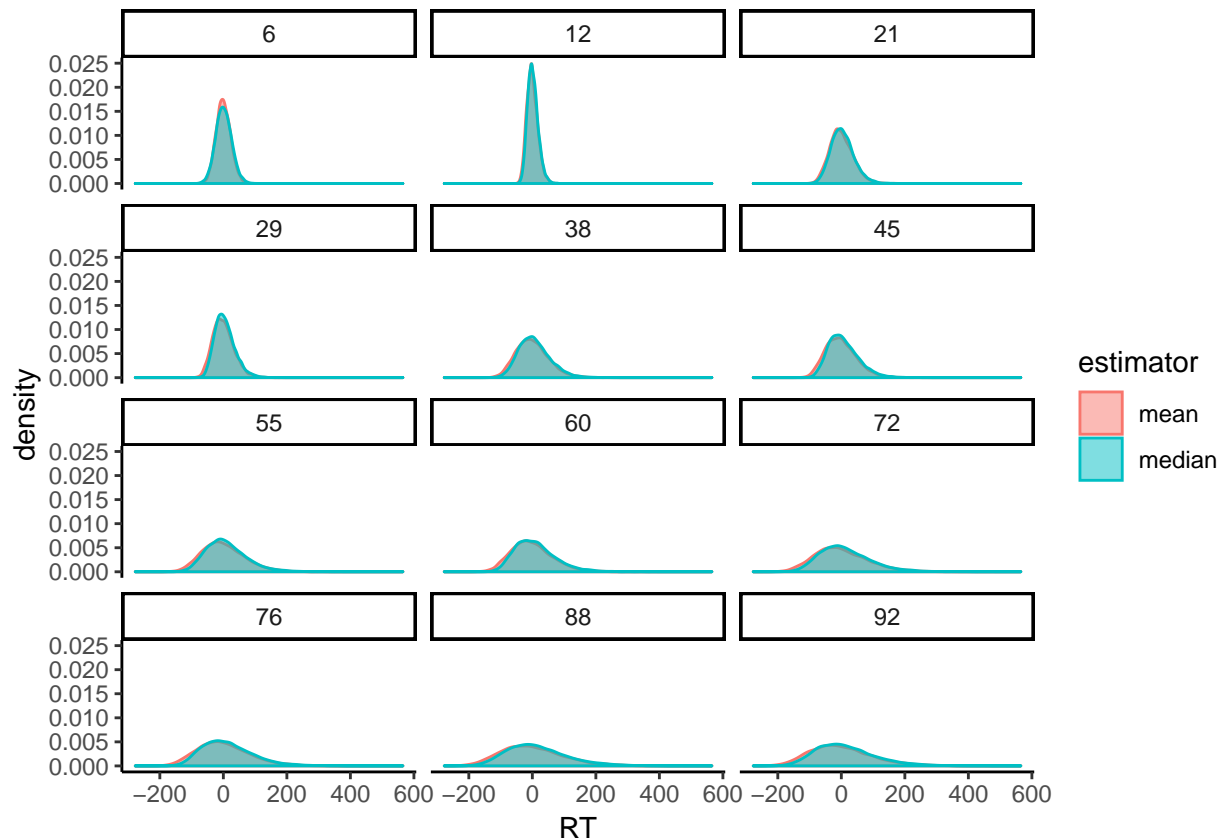
Make figure

```

df <- tibble(RT = c(as.vector(res.m), as.vector(res.md)),
  Skewness = factor(rep(rep(round(pop.m - pop.md), each = np), 2)),
  estimator = factor(c(rep("mean", np*12), rep("median", np*12))))

ggplot(df, aes(x = RT, colour = estimator, fill = estimator)) + theme_classic() +
  geom_density(alpha = 0.5) +
  facet_wrap(vars(Skewness), nrow = 4)

```



Simulation


```

npseq <- c(25, 50, 100, 200)
maxnp <- max(npseq) # max number of participants
ntseq1 <- c(seq(10, 100, 10), 150, 200)
maxnt1 <- max(ntseq1) # max number of trials
ntseq2 <- rep(200, length(ntseq1)) # number of trials in group 2
maxnt2 <- max(ntseq2) # max number of trials
nsim <- 10000 # simulation samples

# declare matrices of results - save all iterations
res.m.m.sig <- array(NA, dim = c(nsim, nP, length(ntseq1), length(npseq)))
res.md.m.sig <- array(NA, dim = c(nsim, nP, length(ntseq1), length(npseq)))
res.md.md.sig <- array(NA, dim = c(nsim, nP, length(ntseq1), length(npseq)))
res.q1.md.sig <- array(NA, dim = c(nsim, nP, length(ntseq1), length(npseq)))
res.q3.md.sig <- array(NA, dim = c(nsim, nP, length(ntseq1), length(npseq)))

set.seed(21)

for(P in 1:nP){ # ex-Gaussian parameters

  # generate all data at once
  print(paste0("parameters: ", P, " out of ", nP, "..."))
  mu <- miller.param[P,1]
  sigma <- miller.param[P,2]
  tau <- miller.param[P,3]
  mc.data1 <- array(rexgauss(maxnt1*maxnp*nsim, mu = mu, sigma = sigma, tau = tau),
    dim = c(maxnt1, maxnp, nsim))
  mc.data2 <- array(rexgauss(maxnt2*maxnp*nsim, mu = mu, sigma = sigma, tau = tau),
    dim = c(maxnt2, maxnp, nsim))

  for(TR in 1:length(ntseq1)){ # number of trials
    # print(paste0("number of trials: ", ntseq[TR], "..."))

    for(PT in 1:length(npseq)){ # number of participants

      # compute estimates
      todo1 <- mc.data1[1:ntseq1[TR], 1:npseq[PT],]
      todo2 <- mc.data2[1:ntseq2[TR], 1:npseq[PT],]

      # diff in mean RTs
      mc.m <- apply(todo1, c(2,3), mean) - apply(todo2, c(2,3), mean)

      # diff in quartile RTs
      qres <- apply(todo1, c(2,3), quantile, probs = c(0.25, 0.5, 0.75)) -

      # tests for each simulation =====
      # one-sample t-test
      res.m.m.sig[, P, TR, PT] <- ttest.sig(mc.m)
      res.md.m.sig[, P, TR, PT] <- ttest.sig(qres[2,,])
      # parametric median tests of the quartiles
      res.q1.md.sig[, P, TR, PT] <- apply(qres[1,,], 2, sint.sig)
      res.md.md.sig[, P, TR, PT] <- apply(qres[2,,], 2, sint.sig)
      res.q3.md.sig[, P, TR, PT] <- apply(qres[3,,], 2, sint.sig)
    }
  }
}

```

apply

```

    }
  }
  save(
    res.m.m.sig,
    res.md.m.sig,
    res.md.md.sig,
    res.q1.md.sig,
    res.q3.md.sig,
    ntseq1,
    ntseq2,
    npseq,
    nsim,
    file=paste0('./data/sim_gp_fp2.RData'))

```

Compute false positive probability

```

load('./data/sim_gp_fp2.RData')
fp.m.m <- apply(res.m.m.sig, c(2,3,4), mean)
fp.md.m <- apply(res.md.m.sig, c(2,3,4), mean)
fp.md.md <- apply(res.md.md.sig, c(2,3,4), mean)
fp.q1.md <- apply(res.q1.md.sig, c(2,3,4), mean)
fp.q3.md <- apply(res.q3.md.sig, c(2,3,4), mean)

ntseq <- ntseq1
xlabels <- c("10", "", "30", "", "50", "", "70", "", "90", "", "150", "200")

```

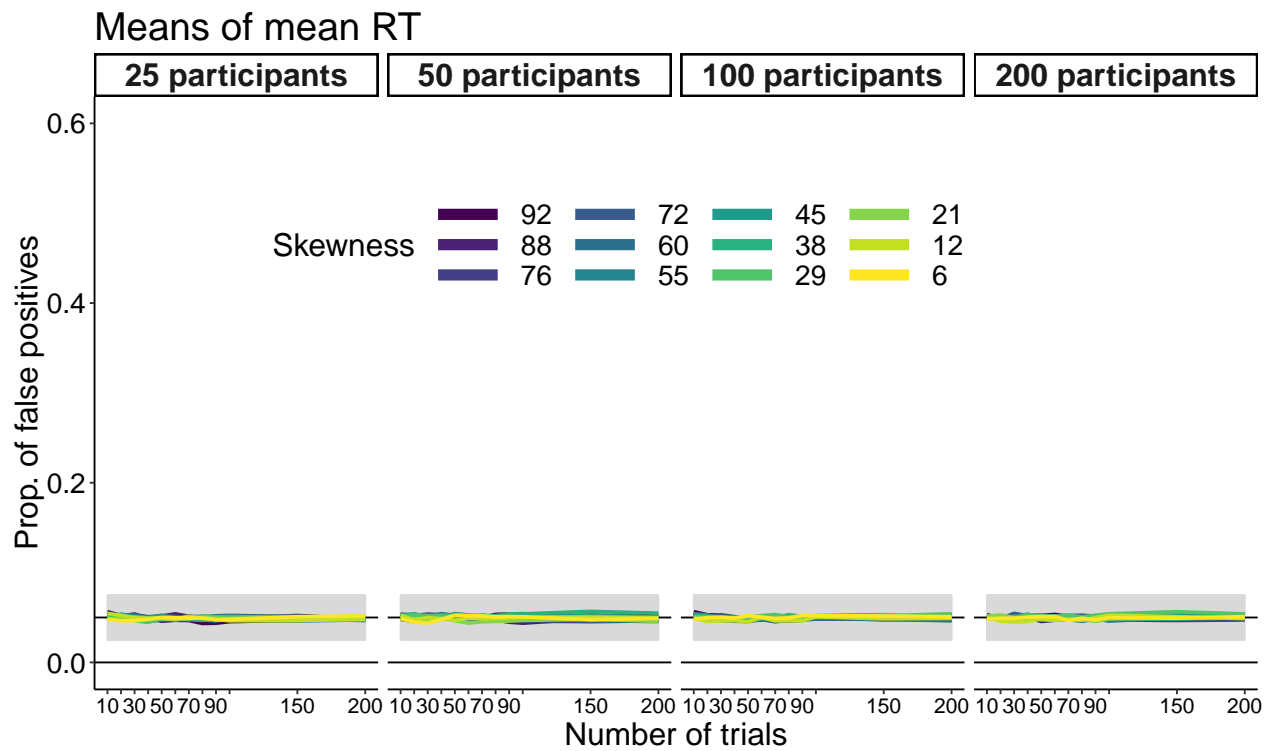
Illustrate results

Group means of means

```

p <- make_fp_figure(fp.m.m, ntseq, nP, npseq, pop.m, pop.md) +
  coord_cartesian(ylim=c(0,0.6)) +
  ggtitle("Means of mean RT")
p

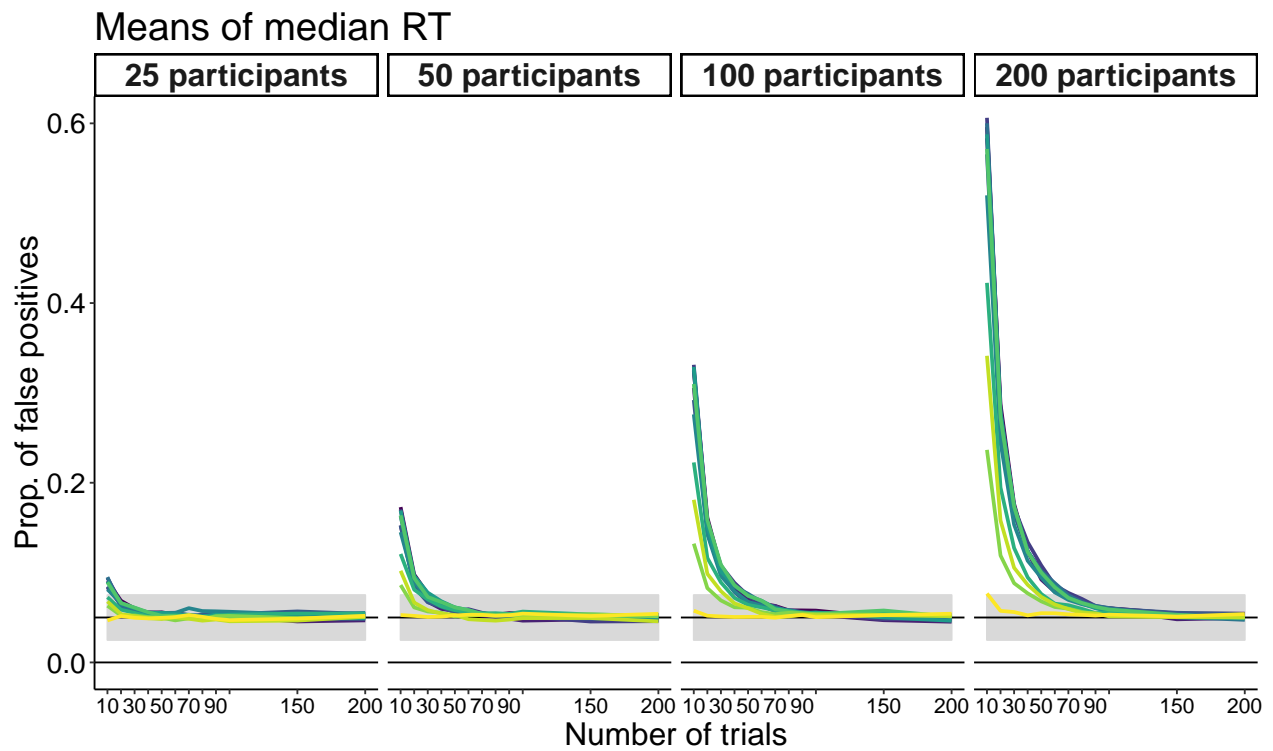
```



```
pm.m <- p
```

Group means of medians

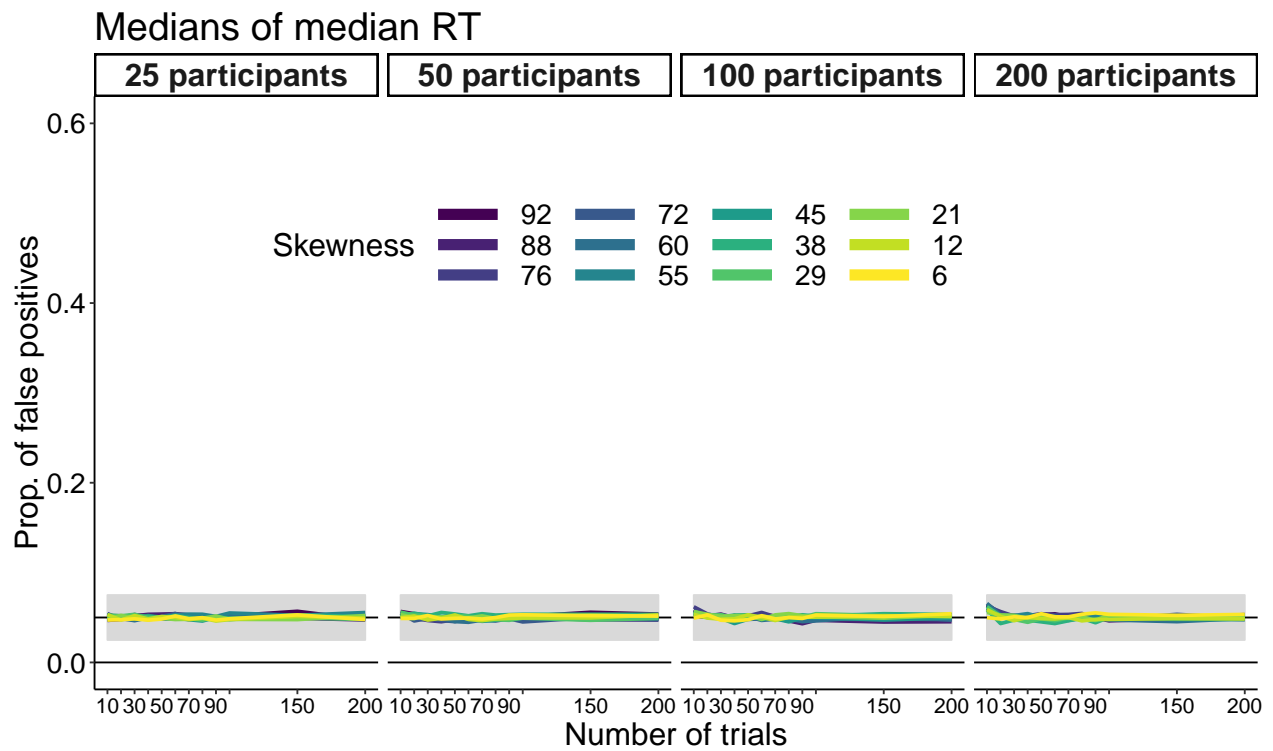
```
p <- make_fp_figure(fp.md.m, ntseq, nP, npseq, pop.m, pop.md) +
  theme(legend.position = "none") +
  coord_cartesian(ylim=c(0,0.6)) +
  ggtitle("Means of median RT")
p
```



```
pmd.m <- p
```

Group medians of medians

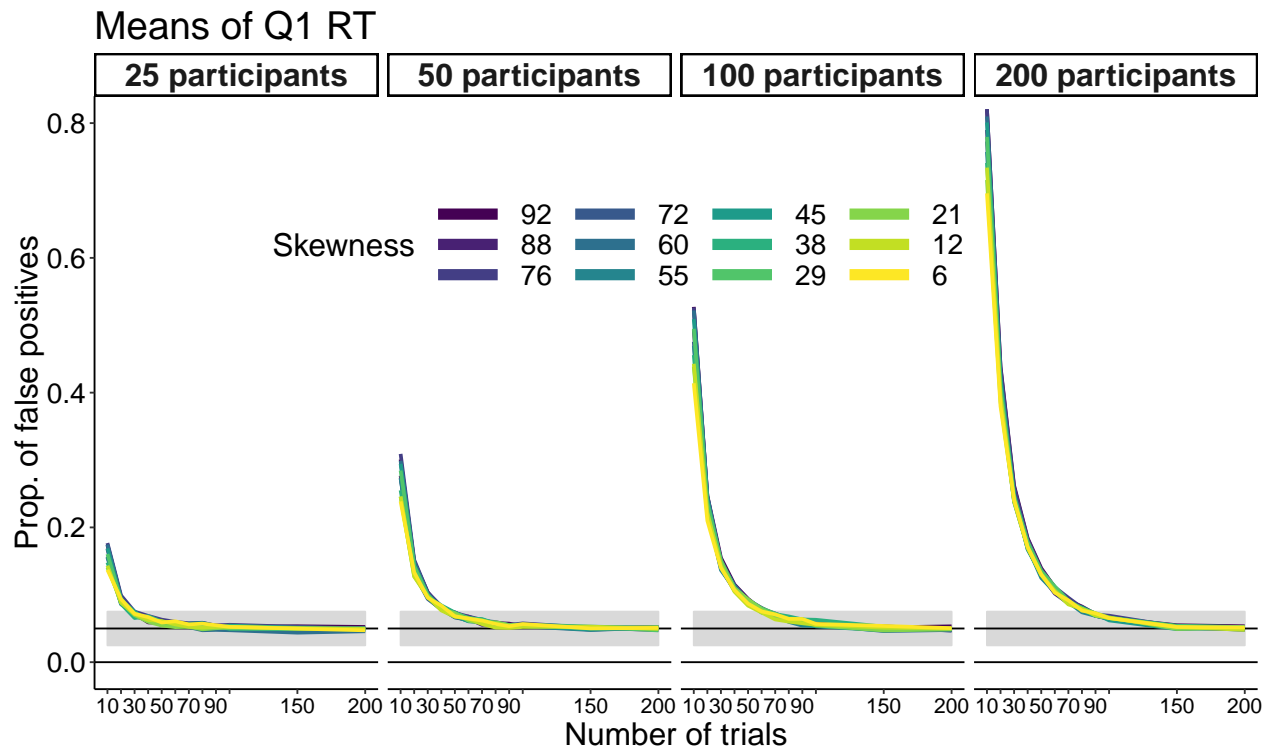
```
p <- make_fp_figure(fp.md.md, ntseq, nP, npseq, pop.m, pop.md) +  
  coord_cartesian(ylim=c(0,0.6)) +  
  ggtitle("Medians of median RT")  
p
```



```
pmd.md <- p
```

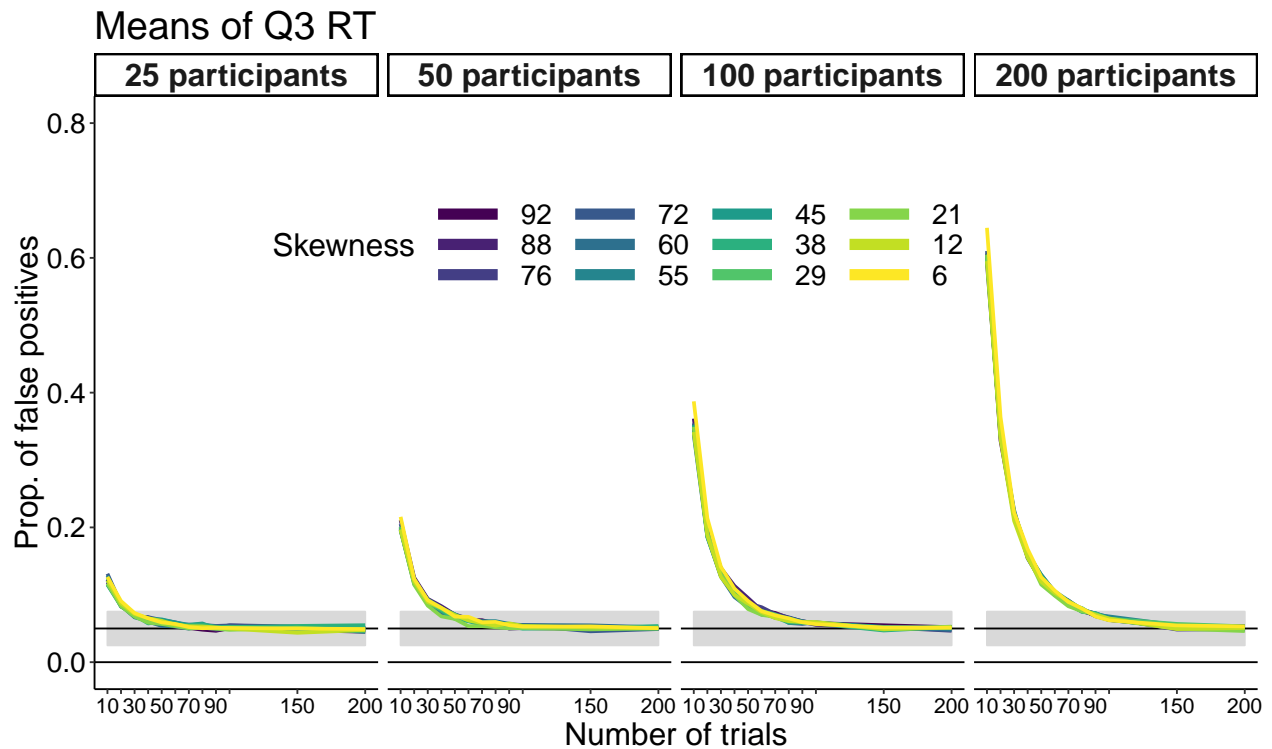
Group medians of Q1

```
p <- make_fp_figure(fp.q1.md, ntseq, nP, npseq, pop.m, pop.md) +
  coord_cartesian(ylim=c(0,0.8)) +
  ggtitle("Means of Q1 RT")
p
```



Group medians of Q3

```
p <- make_fp_figure(fp.q3.md, ntseq, nP, npseq, pop.m, pop.md) +
  coord_cartesian(ylim=c(0,0.8)) +
  ggtitle("Means of Q3 RT")
p
```



Medians of means simulation

```

npseq <- c(25, 50, 100, 200)
maxnp <- max(npseq) # max number of participants
ntseq1 <- c(seq(10, 100, 10), 150, 200)
maxnt1 <- max(ntseq1) # max number of trials
ntseq2 <- rep(200, length(ntseq1)) # number of trials in group 2
maxnt2 <- max(ntseq2) # max number of trials
nsim <- 10000 # simulation samples

# declare matrices of results - save all iterations
res.m.md.sig <- array(NA, dim = c(nsim, nP, length(ntseq1), length(npseq)))

set.seed(21)

for(P in 1:nP){ # ex-Gaussian parameters

  # generate all data at once
  print(paste0("parameters: ", P, " out of ", nP, "..."))
  mu <- miller.param[P,1]
  sigma <- miller.param[P,2]
  tau <- miller.param[P,3]
  mc.data1 <- array(rexgauss(maxnt1*maxnp*nsim, mu = mu, sigma = sigma, tau = tau),
    dim = c(maxnt1, maxnp, nsim))
  mc.data2 <- array(rexgauss(maxnt2*maxnp*nsim, mu = mu, sigma = sigma, tau = tau),
    dim = c(maxnt2, maxnp, nsim))

  for(TR in 1:length(ntseq1)){ # number of trials

```

```

# print(paste0("number of trials: ",ntseq[TR],"..."))

for(PT in 1:length(npseq)){ # number of participants

  # compute estimates
  todo1 <- mc.data1[1:ntseq1[TR], 1:npseq[PT],]
  todo2 <- mc.data2[1:ntseq2[TR], 1:npseq[PT],]

  # diff in mean RTs
  mc.m <- apply(todo1, c(2,3), mean) - apply(todo2, c(2,3), mean)

  # tests for each simulation =====
  # parametric median tests
  res.m.md.sig[, P, TR, PT] <- apply(mc.m, 2, sint.sig)
}
}
}
save(
  res.m.md.sig,
  ntseq1,
  ntseq2,
  npseq,
  nsim,
  file=paste0('./data/sim_gp_fp2_m_md.RData'))

```

Compute false positive probability

```

load('./data/sim_gp_fp2_m_md.RData')
fp.m.md <- apply(res.m.md.sig, c(2,3,4), mean)
ntseq <- ntseq1

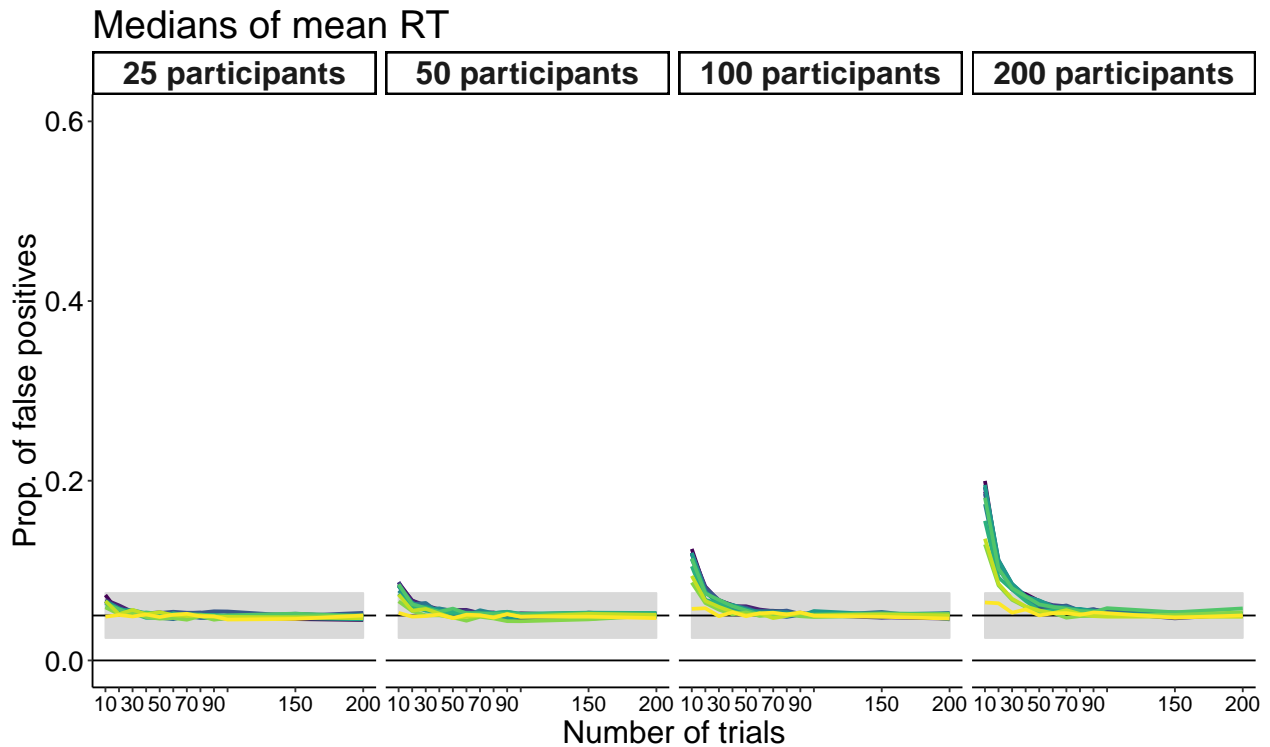
```

Illustrate results

```

p <- make_fp_figure(fp.m.md, ntseq, nP, npseq, pop.m, pop.md) +
  theme(legend.position = "none") +
  coord_cartesian(ylim=c(0,0.6)) +
  ggtitle("Medians of mean RT")
p

```

```
pm.md <- p
```

Means of medians simulation, with bias correction

```
np <- 200 # number of participants
ntseq1 <- c(seq(10, 100, 10))
maxnt1 <- max(ntseq1) # max number of trials
ntseq2 <- rep(200, length(ntseq1)) # number of trials in group 2
maxnt2 <- max(ntseq2) # max number of trials
nsim <- 2000 # simulation samples
nboot <- 200 # bootstrap bias correction

# declare matrices of results - save all iterations
res.md.m.sig <- array(NA, dim = c(nsim, nP, length(ntseq1)))
res.md.bc.sig <- array(NA, dim = c(nsim, nP, length(ntseq1)))

set.seed(21)

for(P in 1:nP){ # ex-Gaussian parameters
  beep(3)
  # generate all data at once
  print(paste0("parameters: ", P, " out of ", nP, "..."))
  mu <- miller.param[P,1]
  sigma <- miller.param[P,2]
  tau <- miller.param[P,3]
  mc.data1 <- array(rexgauss(maxnt1*np*nsim, mu = mu, sigma = sigma, tau = tau),
    dim = c(maxnt1, np, nsim))
  mc.data2 <- array(rexgauss(maxnt2*np*nsim, mu = mu, sigma = sigma, tau = tau),
    dim = c(maxnt2, np, nsim))
}
```

```

for(TR in 1:length(ntseq1)){ # number of trials
  beep(2)
  print(paste0("number of trials: ",ntseq1[TR],"..."))

  # compute estimates
  todo1 <- mc.data1[1:ntseq1[TR],,]
  todo2 <- mc.data2[1:ntseq2[TR],,]

  # diff in median RTs
  mc.md1 <- apply(todo1, c(2,3), median)
  mc.md2 <- apply(todo2, c(2,3), median)
  mc.diff <- mc.md1 - mc.md2

  # compute bias corrected estimates using the same nboot bootstrap samples for all estimators
  bc1 <- matrix(NA, nrow = np, ncol = nsim)
  bc2 <- matrix(NA, nrow = np, ncol = nsim)
  for(iter in 1:nsim){
    for(PT in 1:np){
      boot.md <- apply(matrix(sample(todo1[,PT,iter], ntseq1[TR]*nboot, replace = TRUE), nrow=nboot),
        bc1[PT,iter] <- 2 * mc.md1[PT,iter] - mean(boot.md)
      boot.md <- apply(matrix(sample(todo2[,PT,iter], ntseq2[TR]*nboot, replace = TRUE), nrow=nboot),
        bc2[PT,iter] <- 2 * mc.md2[PT,iter] - mean(boot.md)
    }
  }

  # tests for each simulation =====
  # parametric mean t-tests of the medians
  res.md.m.sig[, P, TR] <- ttest.sig(mc.diff) # apply(mc.diff, 2, sint.sig)
  res.md.bc.sig[, P, TR] <- ttest.sig(bc1 - bc2) # apply(bc1 - bc2, 2, sint.sig)

}
save(
  res.md.m.sig,
  res.md.bc.sig,
  ntseq1,
  ntseq2,
  np,
  nsim,
  file='./data/sim_gp_fp2_bc.RData')
}
beep(8)

```

Compute false positive probability

```

load('./data/sim_gp_fp2_bc.RData')
fp.md.m <- apply(res.md.m.sig, c(2,3), mean)
fp.md.bc <- apply(res.md.bc.sig, c(2,3), mean)
ntseq <- ntseq1

```

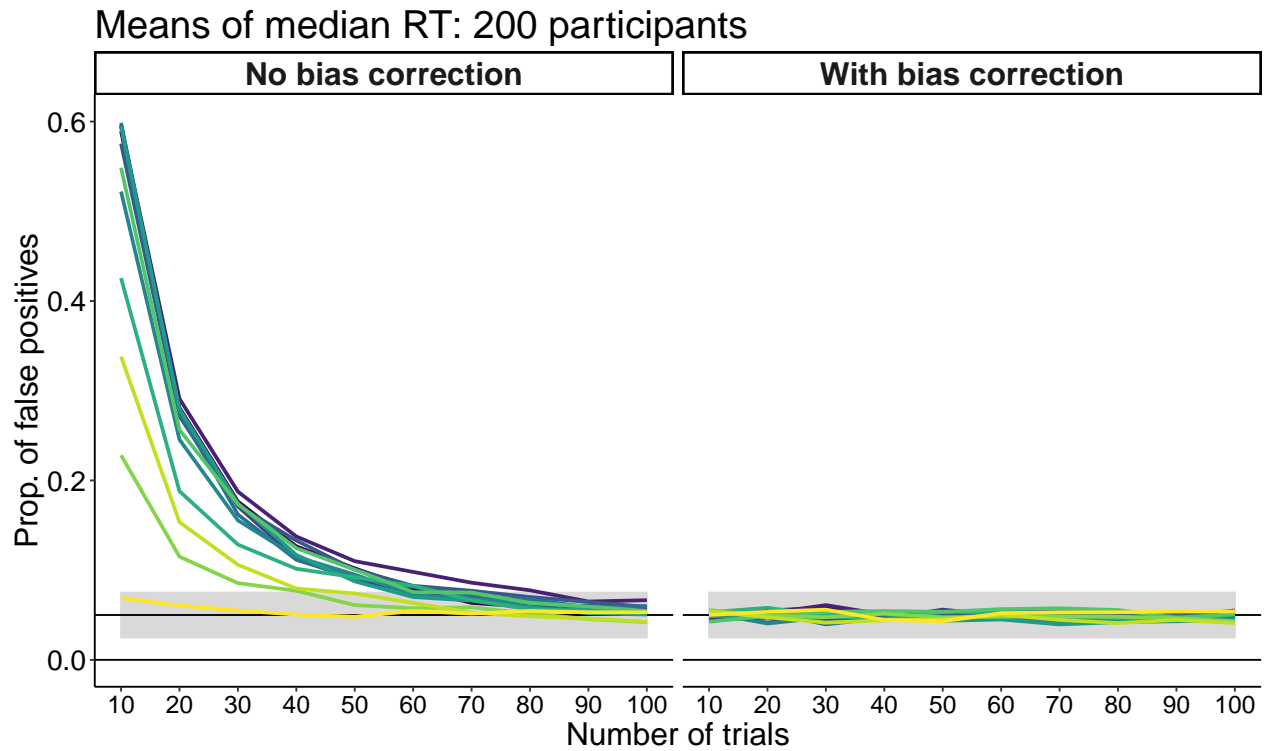
Illustrate results

```
df <- tibble(`FP`=c(as.vector(fp.md.m),as.vector(fp.md.bc)),
             `Skewness`=rep(round(pop.m - pop.md),length(ntseq)*2),
             `BC`=factor(rep(c("No bias correction","With bias correction"),each=nP*length(ntseq)))
             )

df$Skewness <- as.character(df$Skewness)
df$Skewness <- factor(df$Skewness, levels=unique(df$Skewness))

# make plot
p <- ggplot(df, aes(x=Trials, y=FP, colour = Skewness)) + theme_classic() +
  # Bradley's (1978) satisfactory range
  geom_ribbon(aes(x=Trials), ymin = 0.025, ymax = 0.075,
            fill = "grey85", colour = "grey85", show.legend=FALSE) +
  # 0.05 reference line
  geom_abline(intercept = 0.05, slope = 0, colour="black") +
  geom_line(size = 1) +
  geom_abline(intercept=0, slope=0, colour="black") +
  scale_colour_viridis_d() +
  scale_x_continuous(breaks=ntseq) +
  # scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
  coord_cartesian(ylim=c(0,0.6)) +
  theme(plot.title = element_text(size=22),
        axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 14, colour="black"),
        axis.text.y = element_text(size = 16, colour="black"),
        axis.title.y = element_text(size = 18),
        legend.key.width = unit(1.5,"cm"),
        legend.position = "none",
        legend.direction = "horizontal",
        legend.text=element_text(size=16),
        legend.title=element_text(size=18),
        strip.text = element_text(size=18, face="bold"),
        strip.background = element_rect(colour="black", fill="white")) +
  labs(x = "Number of trials", y = "Prop. of false positives") +
  guides(colour = guide_legend(override.aes = list(size=3))) + # make thicker legend lines
  ggtitle("Means of median RT: 200 participants") +
  facet_grid(cols = vars(BC))
```

p



```
pmdbc.m <- p
```

Summary figure 2: group means

```
# combine panels into one figure
cowplot::plot_grid(pm.m, pmd.m, pmdbc.m,
  labels = c("A", "B", "C"),
  ncol = 1,
  nrow = 3,
  rel_widths = c(1, 1, 1),
  label_size = 20,
  hjust = -0.5,
  scale=.95,
  align = "h")

# save figure
ggsave(filename=('./figures/figure_sim_gp_fp2_part1.pdf'),width=10,height=12)
```

Summary figure 3: group medians

```
# combine panels into one figure
cowplot::plot_grid(pmd.md, pm.md,
  labels = c("A", "B"),
  ncol = 1,
  nrow = 2,
  rel_widths = c(1, 1),
  label_size = 20,
  hjust = -0.5,
```

```

        scale=.95,
        align = "h")
# save figure
ggsave(filename='./figures/figure_sim_gp_fp2_part2.pdf',width=10,height=8)

```

Group simulation 3: shift function

Most and least skewed exGaussian distributions with different numbers of trials. Group statistics on quantiles using median, mean and trimmed mean tests.

Vary: - n trials - n participants - skewness

```

npseq <- c(25, 50, 100, 200)
maxnp <- max(npseq) # max number of participants
ntseq1 <- c(seq(50, 100, 10), 150, 200)
maxnt1 <- max(ntseq1) # max number of trials
ntseq2 <- rep(200, length(ntseq1)) # number of trials in group 2
maxnt2 <- max(ntseq2) # max number of trials
nsim <- 10000 # simulation samples
alpha <- 0.05
qseq <- seq(0.1,0.9,0.1) # input to quantile function

# declare matrices of results - save all iterations
# Harrell-Davis + quantile type 8
res.hsf.md.sig <- array(NA, dim = c(nsim, length(ntseq1), length(npseq)))
res.hsf.m.sig <- array(NA, dim = c(nsim, length(ntseq1), length(npseq)))
res.hsf.tm10.sig <- array(NA, dim = c(nsim, length(ntseq1), length(npseq)))
res.hsf.tm20.sig <- array(NA, dim = c(nsim, length(ntseq1), length(npseq)))

res.hqt8.md.sig <- array(NA, dim = c(nsim, length(ntseq1), length(npseq)))
res.hqt8.m.sig <- array(NA, dim = c(nsim, length(ntseq1), length(npseq)))
res.hqt8.tm10.sig <- array(NA, dim = c(nsim, length(ntseq1), length(npseq)))
res.hqt8.tm20.sig <- array(NA, dim = c(nsim, length(ntseq1), length(npseq)))

pvals0 <- vector(mode = "numeric", length = 9)
pvals10 <- vector(mode = "numeric", length = 9)
pvals20 <- vector(mode = "numeric", length = 9)

set.seed(21)

# P <- 1
P <- 12
mu <- miller.param[P,1]
sigma <- miller.param[P,2]
tau <- miller.param[P,3]
mc.data1 <- array(rexgauss(maxnt1*maxnp*nsim, mu = mu, sigma = sigma, tau = tau),
  dim = c(maxnt1, maxnp, nsim))
mc.data2 <- array(rexgauss(maxnt2*maxnp*nsim, mu = mu, sigma = sigma, tau = tau),
  dim = c(maxnt2, maxnp, nsim))

for(TR in 1:length(ntseq1)){ # number of trials
  print(paste0("number of trials: ",ntseq1[TR],"..."))
  beep(2)
}

```

```

for(PT in 1:length(npseq)){ # number of participants

  # Harrell-Davis:
  # array of differences: 9 deciles x participants x simulations
  mc.hd <- apply(mc.data1[1:ntseq1[TR], 1:npseq[PT],], c(2,3), hdseq) -
    apply(mc.data2[1:ntseq1[TR], 1:npseq[PT],], c(2,3), hdseq)
  # median test for each decile distribution
  pval <- apply(mc.hd, c(1,3), sintv2.pval)
  # trimmed mean tests for each decile distribution
  tvals0 <- apply(mc.hd, 1, ctval.yuen, tr=0)
  tvals10 <- apply(mc.hd, 1, ctval.yuen, tr=0.1)
  tvals20 <- apply(mc.hd, 1, ctval.yuen, tr=0.2)
  df0 <- df.yuen(mc.hd[1,,1], tr = 0)
  df10 <- df.yuen(mc.hd[1,,1], tr = 0.1)
  df20 <- df.yuen(mc.hd[1,,1], tr = 0.2)
  for(S in 1:nsim){
    res.hsf.md.sig[S, TR, PT] <- sum(p.adjust(pval[,S], method = "hochberg") <= alpha) > 0
    for(Q in 1:9){
      pvals0[Q] <- cpval(tvals0[S,Q], df0)
      pvals10[Q] <- cpval(tvals10[S,Q], df10)
      pvals20[Q] <- cpval(tvals20[S,Q], df20)
    }
    res.hsf.m.sig[S, TR, PT] <- sum(p.adjust(pvals0, method = "hochberg") <= alpha) > 0
    res.hsf.tm10.sig[S, TR, PT] <- sum(p.adjust(pvals10, method = "hochberg") <= alpha) > 0
    res.hsf.tm20.sig[S, TR, PT] <- sum(p.adjust(pvals20, method = "hochberg") <= alpha) > 0
  }

  # Quantile type 8:
  # array of differences: 9 deciles x participants x simulations
  mc.qt8 <- apply(mc.data1[1:ntseq1[TR], 1:npseq[PT],], c(2,3), quantile, type = 8, probs = qseq) - a
  # median test for each decile distribution
  pval <- apply(mc.qt8, c(1,3), sintv2.pval)
  # 20% trimmed mean test for each decile distribution
  tvals20 <- apply(mc.qt8, 1, ctval.yuen, tr=0.2) # default 0.2
  # 10% trimmed mean test for each decile distribution
  tvals10 <- apply(mc.qt8, 1, ctval.yuen, tr=0.1)
  # mean test for each decile distribution
  tvals0 <- apply(mc.qt8, 1, ctval.yuen, tr=0)
  df0 <- df.yuen(mc.qt8[1,,1], tr=0)
  df10 <- df.yuen(mc.qt8[1,,1], tr=0.1)
  df20 <- df.yuen(mc.qt8[1,,1], tr=0.2)
  for(S in 1:nsim){
    res.hqt8.md.sig[S, TR, PT] <- sum(p.adjust(pval[,S], method = "hochberg") <= alpha) > 0
    for(Q in 1:9){
      pvals0[Q] <- cpval(tvals0[S,Q], df0)
      pvals10[Q] <- cpval(tvals10[S,Q], df10)
      pvals20[Q] <- cpval(tvals20[S,Q], df20)
    }
    res.hqt8.m.sig[S, TR, PT] <- sum(p.adjust(pvals0, method = "hochberg") <= alpha) > 0
    res.hqt8.tm10.sig[S, TR, PT] <- sum(p.adjust(pvals10, method = "hochberg") <= alpha) > 0
    res.hqt8.tm20.sig[S, TR, PT] <- sum(p.adjust(pvals20, method = "hochberg") <= alpha) > 0
  }
}
}

```

```

}
save(
  res.hsf.md.sig,
  res.hsf.m.sig,
  res.hsf.tm10.sig,
  res.hsf.tm20.sig,
  res.hqt8.md.sig,
  res.hqt8.m.sig,
  res.hqt8.tm10.sig,
  res.hqt8.tm20.sig,
  ntseq1,
  ntseq2,
  npseq,
  nsim,
  file=paste0('./data/sim_hsf_fp12.RData'))
# file=paste0('./data/sim_hsf_fp1.RData'))

beep(8)

```

Results from most skewed distribution

```

load('./data/sim_hsf_fp1.RData')
fwer.hsf.md <- apply(res.hsf.md.sig, c(2,3), mean)
fwer.hsf.m <- apply(res.hsf.m.sig, c(2,3), mean)
fwer.hsf.tm10 <- apply(res.hsf.tm10.sig, c(2,3), mean)
fwer.hsf.tm20 <- apply(res.hsf.tm20.sig, c(2,3), mean)

fwer.hqt8.md <- apply(res.hqt8.md.sig, c(2,3), mean)
fwer.hqt8.m <- apply(res.hqt8.m.sig, c(2,3), mean)
fwer.hqt8.tm10 <- apply(res.hqt8.tm10.sig, c(2,3), mean)
fwer.hqt8.tm20 <- apply(res.hqt8.tm20.sig, c(2,3), mean)

xlabels <- c("50", "", "70", "", "90", "", "150", "200")

```

Make plot function

```

plot_sf_fp <- function(data, est, ntseq1, npseq){
  df <- tibble(FP=data,
    Estimator=factor(rep(est, each = length(ntseq1)*length(npseq))),
    Trials=rep(ntseq1, length(npseq)*length(est)),
    Participants=factor(rep(rep(npseq, each=length(ntseq1)), length(est)))
  )
  # make plot
  p <- ggplot(df, aes(x=Trials, y=FP, colour = Estimator)) + theme_classic() +
    # Bradley's (1978) satisfactory range
    geom_ribbon(aes(x=Trials), ymin = 0.025, ymax = 0.075,
      fill = "grey85", colour = "grey85", show.legend=FALSE) +
    # 0.05 reference line
    geom_abline(intercept = 0.05, slope = 0, colour="black") +
    geom_line(size = 1) +

```

```

geom_abline(intercept=0, slope=0, colour="black") +
scale_colour_viridis_d(end = 0.9) +
scale_x_continuous(breaks=ntseq1, labels = xlabel) +
# scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
coord_cartesian(ylim=c(0,0.2)) +
theme(plot.title = element_text(size=22),
      axis.title.x = element_text(size = 18),
      axis.text.x = element_text(size = 12, colour="black"),
      axis.text.y = element_text(size = 16, colour="black"),
      axis.title.y = element_text(size = 18),
      legend.key.width = unit(1.5,"cm"),
      legend.position = "right",
      legend.direction = "vertical",
      legend.text=element_text(size=16),
      legend.title=element_text(size=18),
      strip.text = element_text(size=18, face="bold"),
      strip.background = element_rect(colour="black", fill="white")) +
labs(x = "Number of trials", y = "Familywise error rate") +
guides(colour = guide_legend(override.aes = list(size=3))) + # make thicker legend lines
facet_grid(cols = vars(Participants))
p
}

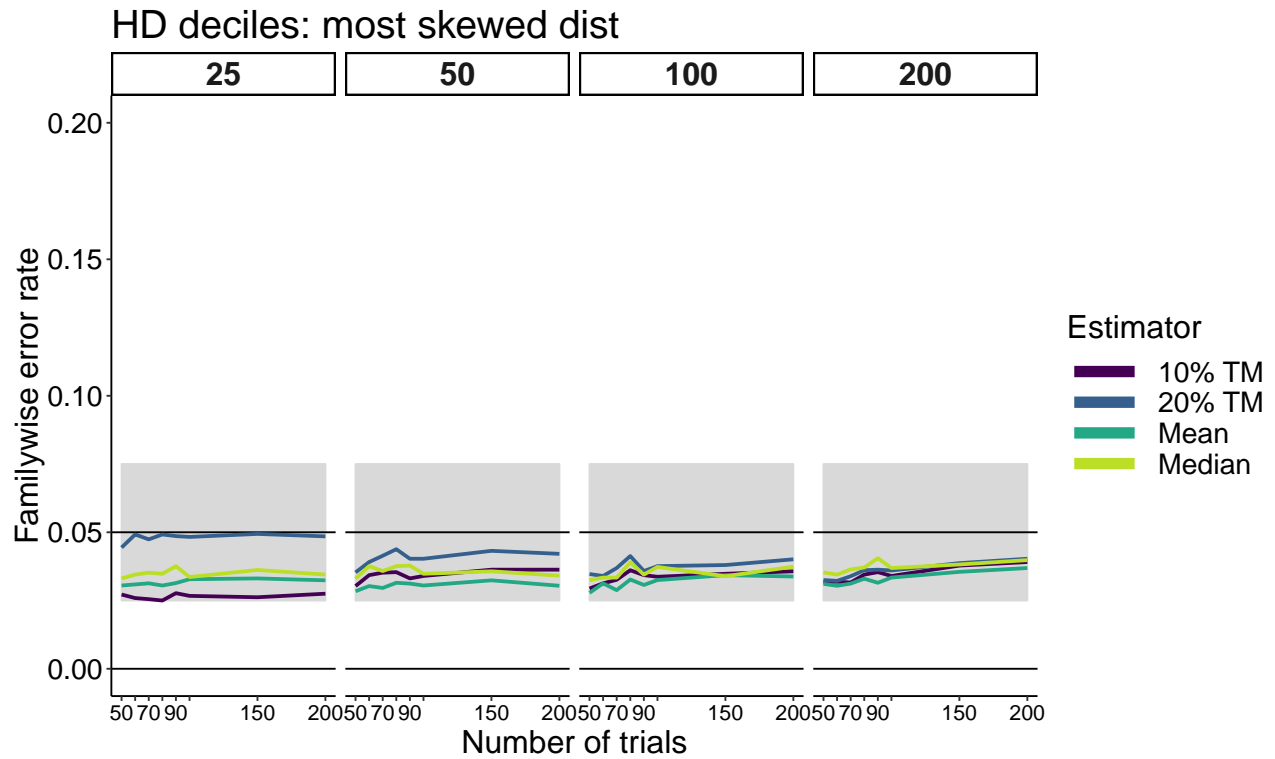
```

Plot HD results:

```

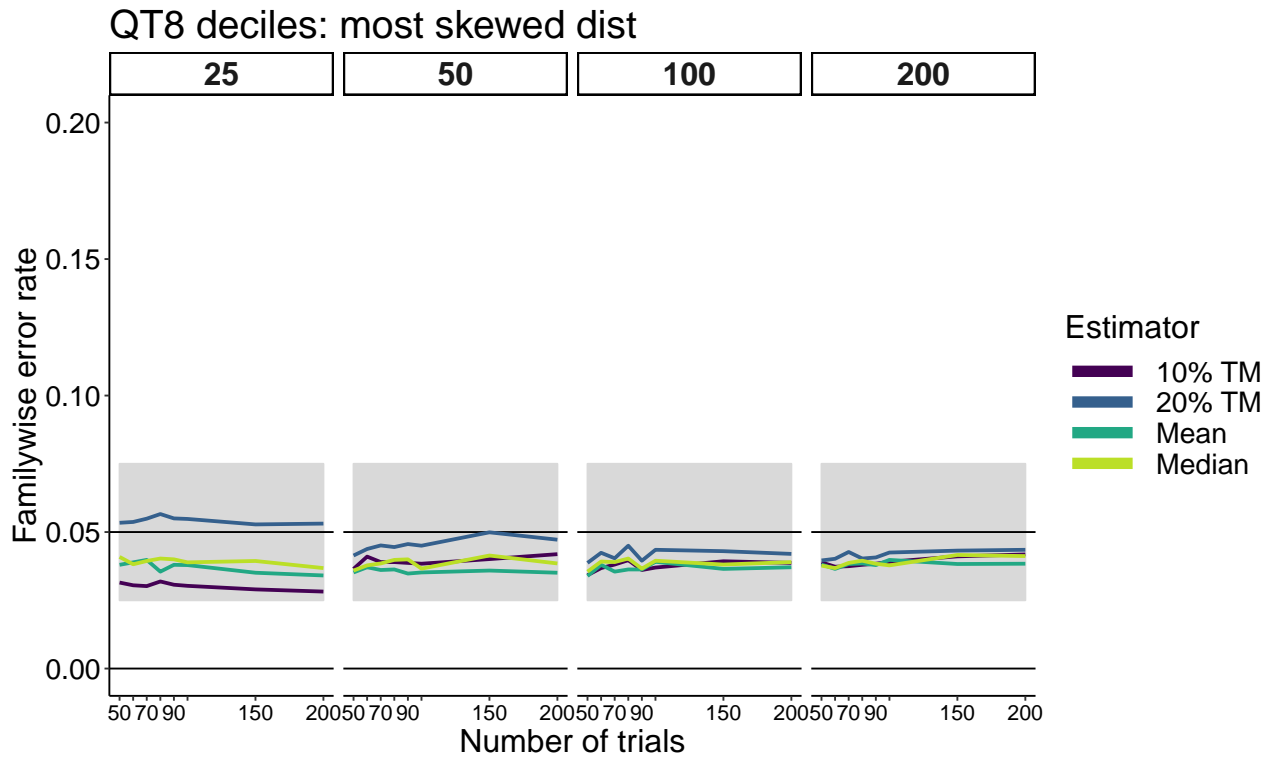
data <- c(as.vector(fwer.hsf.m),
          as.vector(fwer.hsf.md),
          as.vector(fwer.hsf.tm10),
          as.vector(fwer.hsf.tm20))
est <- c("Mean","Median","10% TM","20% TM")
p <- plot_sf_fp(data, est, ntseq1, npseq) + ggtitle("HD deciles: most skewed dist")
p

```

Plot QT8 results:

```
data <- c(as.vector(fwer.hqt8.m),
          as.vector(fwer.hqt8.md),
          as.vector(fwer.hqt8.tm10),
          as.vector(fwer.hqt8.tm20))
est <- c("Mean", "Median", "10% TM", "20% TM")
p <- plot_sf_fp(data, est, ntseq1, npseq) + ggtitle("QT8 deciles: most skewed dist")
p
```



Results from least skewed distribution

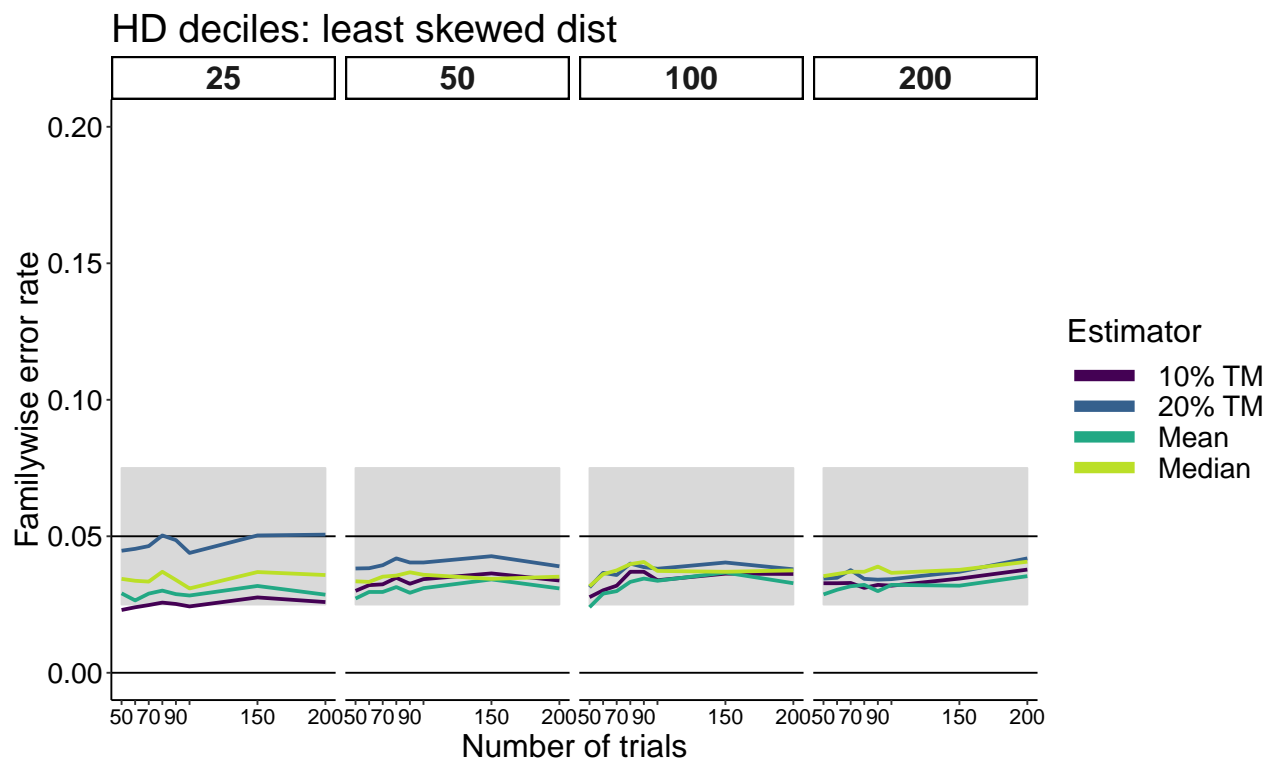
```
load('./data/sim_hsf_fp12.RData')
fwer.hsf.md <- apply(res.hsf.md.sig, c(2,3), mean)
fwer.hsf.m <- apply(res.hsf.m.sig, c(2,3), mean)
fwer.hsf.tm10 <- apply(res.hsf.tm10.sig, c(2,3), mean)
fwer.hsf.tm20 <- apply(res.hsf.tm20.sig, c(2,3), mean)

fwer.hqt8.md <- apply(res.hqt8.md.sig, c(2,3), mean)
fwer.hqt8.m <- apply(res.hqt8.m.sig, c(2,3), mean)
fwer.hqt8.tm10 <- apply(res.hqt8.tm10.sig, c(2,3), mean)
fwer.hqt8.tm20 <- apply(res.hqt8.tm20.sig, c(2,3), mean)

xlabels <- c("50", "", "70", "", "90", "", "150", "200")
```

Plot HD results:

```
data <- c(as.vector(fwer.hsf.m),
          as.vector(fwer.hsf.md),
          as.vector(fwer.hsf.tm10),
          as.vector(fwer.hsf.tm20))
est <- c("Mean", "Median", "10% TM", "20% TM")
p <- plot_sf_fp(data, est, ntseq1, npseq) + ggtitle("HD deciles: least skewed dist")
p
```



Plot QT8 results:

```
data <- c(as.vector(fwer.hqt8.m),
          as.vector(fwer.hqt8.md),
          as.vector(fwer.hqt8.tm10),
          as.vector(fwer.hqt8.tm20))
est <- c("Mean", "Median", "10% TM", "20% TM")
p <- plot_sf_fp(data, est, ntseq1, npseq) + ggtitle("QT8 deciles: least skewed dist")
p
```

