

Quantile difference estimation bias: Harrell-Davis and quantile(type=8)

Guillaume A. Rousselet

2019-01-16

Contents

g & h distributions	2
Simulation with same number of trials in each group	4
Compute bias	4
Illustrate bias results	5
Illustrate sampling distributions: HD	7
Simulation with different numbers of trials in each condition	14
Compute bias	14
Illustrate bias results	15
Illustrate bias results: median bias	16
Illustrate sampling distributions: HD	18
Illustrate sampling distributions: QT8	24
Simulation with varying numbers of trials in both groups	31
Compute bias	32
Illustrate bias results: HD	32
Illustrate bias results: QT8	33
Simulation with same number of trials in each group using exG dist	34
Define ex-Gaussian parameters	35
Simulation	35
Compute bias	36
Illustrate bias results	36
Illustrate bias results: median bias	38
Illustrate sampling distributions	40
Simulation with different numbers of trials in each group using exG dist	46
Simulation	46
Compute bias	47
Illustrate bias results	47
Illustrate bias results: median bias	49
Illustrate sampling distributions	51

```
# dependencies
library(rogme)
library(tibble)
library(tidyr)
library(cowplot)
library(retimes)
source("../functions/gh.txt")
source("../functions/akerd.txt")
library(beepR)
```

```
sessionInfo()
```

```
## R version 3.4.0 (2017-04-21)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS 10.14.2
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.4/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_0.9.1  tidyr_0.7.2    tibble_1.4.2
## [6] rogme_0.2.0     ggplot2_3.0.0
##
## loaded via a namespace (and not attached):
## [1] Rcpp_0.12.19    pillar_1.3.0    compiler_3.4.0  plyr_1.8.4
## [5] bindr_0.1.1     tools_3.4.0     digest_0.6.12   evaluate_0.10.1
## [9] gtable_0.2.0    pkgconfig_2.0.2  rlang_0.2.2     rstudioapi_0.8
## [13] yaml_2.1.15     bindrcpp_0.2.2   withr_2.1.0     dplyr_0.7.6
## [17] stringr_1.2.0   knitr_1.17       rprojroot_1.2   grid_3.4.0
## [21] tidyselect_0.2.4 glue_1.3.0       R6_2.3.0        rmarkdown_1.8
## [25] purrr_0.2.5     magrittr_1.5     backports_1.1.1  scales_0.5.0
## [29] htmltools_0.3.6 assertthat_0.2.0 colorspace_1.3-2 stringi_1.1.6
## [33] lazyeval_0.2.1  munsell_0.4.3    crayon_1.3.4     audio_0.1-5.1
```

Goal: to illustrate bias of shift functions using g & h distributions and Miller's 12 ex-Gaussian distributions. We limit our investigation to deciles, estimated using `hd()` and `quantile(type = 8)`.

The `quantile` function in base R offers 9 options. Option 8 was recommended by Hyndman and Fan (1996). It gives quantile estimates that are approximately median-unbiased regardless of the distribution of x . Preliminary tests suggest weaker bias for the extreme deciles compared to the Harrell-Davis quantile estimator. If this is the case, should be tested in `sim_gp_fp` in the hierarchical context.

g & h distributions

The `ghdist()` function is used to generate random numbers from g & h distributions. All such distributions have a median of zero. The parameter g controls the asymmetry of the distribution, while the parameter h controls the thickness of the tails. The g & h distributions are described in this 1985 book: <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-047004005X.html> There is also a description in Rand Wilcox's book Introduction to Robust Estimation. See also: <https://www.jstor.org/stable/25471119>

Examples in which we vary g from 0 to 1.

```
ng <- seq(0,1,0.1)
x <- seq(-4, 15, 0.05)
res.g <- array(0, dim = c(length(x), length(ng)))
for(G in 1:length(ng)){
  set.seed(7)
```

```

  res.g[,G] <- akerd(ghdist(10000, g = ng[G], h = 0), pts = x, pyhat = TRUE, plotit = FALSE)
}

```

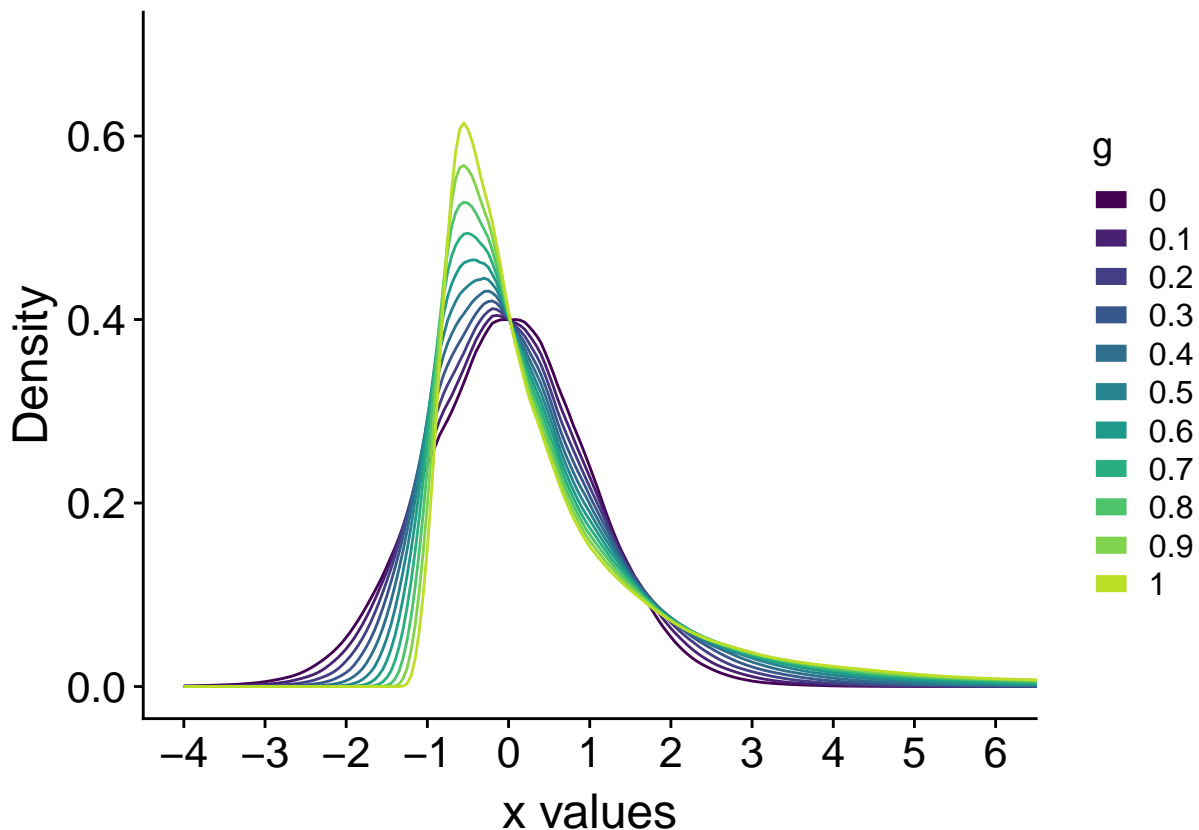
Combine all kernel density functions into one data frame and make summary figure.

```

# make data frame
fm <- array(0, dim = c(length(x), length(ng)+1)) # make full matrix
fm[,1] <- x
fm[,2:(length(ng)+1)] <- res.g
colnames(fm) <- c("x", paste(seq(0, 1, 0.1)))
df <- as_tibble(fm[,])
df <- tidyr::gather(df, g, Density, 2:(length(ng)+1))
df[[2]] <- as.factor(df[[2]])

# make plot
p <- ggplot(df, aes(x, Density, group = g)) +
  geom_line(aes(colour = g), size=0.5) +
  scale_colour_viridis_d(end = 0.9) +
  theme(axis.title.x = element_text(size = 18),
        axis.text = element_text(size = 16),
        axis.title.y = element_text(size = 18)) +
  scale_y_continuous(limits = c(0, 0.7),
                    breaks = seq(0, 1, 0.2)) +
  scale_x_continuous(breaks = seq(-4, 15, 1)) +
  coord_cartesian(xlim = c(-4, 6)) +
  labs(x = "x values", y = "Density") +
  guides(colour = guide_legend(override.aes = list(size=3)))
p

```



```
# save figure
# ggsave(filename='figure_g_distributions.pdf',width=7,height=5)
```

Simulation with same number of trials in each group

```
gvec <- seq(0, 1, 0.1) # g values
nvec <- c(seq(30,100,10), 150, 200, 300) # vector of sample sizes to test
nsim <- 10000 # simulation samples
qseq <- seq(0.1,0.9,0.1)

# declare matrices of results - save all iterations
sim.sfhd.bias <- array(NA, dim = c(nsim, length(gvec), length(nvec), 9))
sim.sfqt8.bias <- array(NA, dim = c(nsim, length(gvec), length(nvec), 9))

set.seed(21)
beep(3)

for(S in 1:nsim){ # simulation iterations

  if(S %% 500 == 0){
    print(paste0("SF bias: simulation ",S," out of ",nsim,"..."))
    beep(2)
  }

  for(G in 1:length(gvec)){

    large.sample1 <- ghdist(max(nvec), g = gvec[G], h = 0)
    large.sample2 <- ghdist(max(nvec), g = gvec[G], h = 0)

    for(N in 1:length(nvec)){ # sample sizes

      sim.sfhd.bias[S,G,N,] <- hdseq(large.sample1[1:nvec[N]]) - hdseq(large.sample2[1:nvec[N]])
      sim.sfqt8.bias[S,G,N,] <- quantile(large.sample1[1:nvec[N]], probs = qseq, type = 8) -
    }
  }
}
save(
  sim.sfhd.bias,
  sim.sfqt8.bias,
  gvec,
  nvec,
  nsim,
  file=paste0('./data/sim_sf_bias1.RData'))
beep(8)
```

Compute bias

Bias = average of sample estimates minus population value. Because we subtract estimates (quantiles) from 2 samples from the same population, the expected population values are zero for all quantile differences.

```
load('./data/sim_sf_bias1.RData')
bias.hd <- apply(sim.sfhd.bias, c(2,3,4), mean, na.rm = TRUE)
bias.qt8 <- apply(sim.sfqt8.bias, c(2,3,4), mean, na.rm = TRUE)
xlabels <- paste(c(30,"",50,"",70,"",90,"",150, 200, 300))
```

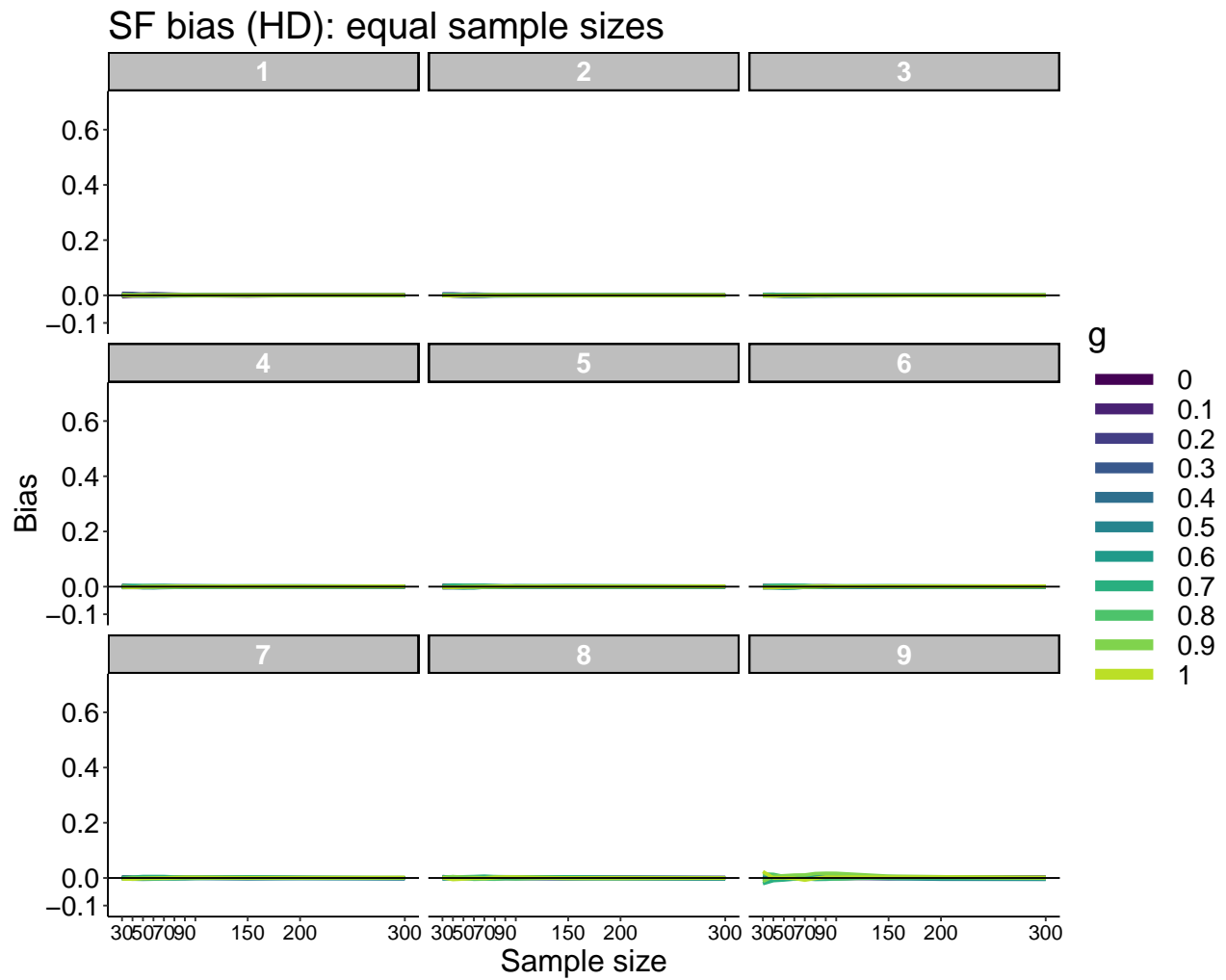
Illustrate bias results

Make function

```
make_bias_figure <- function(data, gvec, nvec){
  df <- tibble(Bias = as.vector(data),
               g = factor(rep(gvec, length(nvec)*9)),
               Size = rep(rep(nvec, each=length(gvec)), 9),
               q = factor(rep(seq(1,9), each=length(gvec)*length(nvec))))

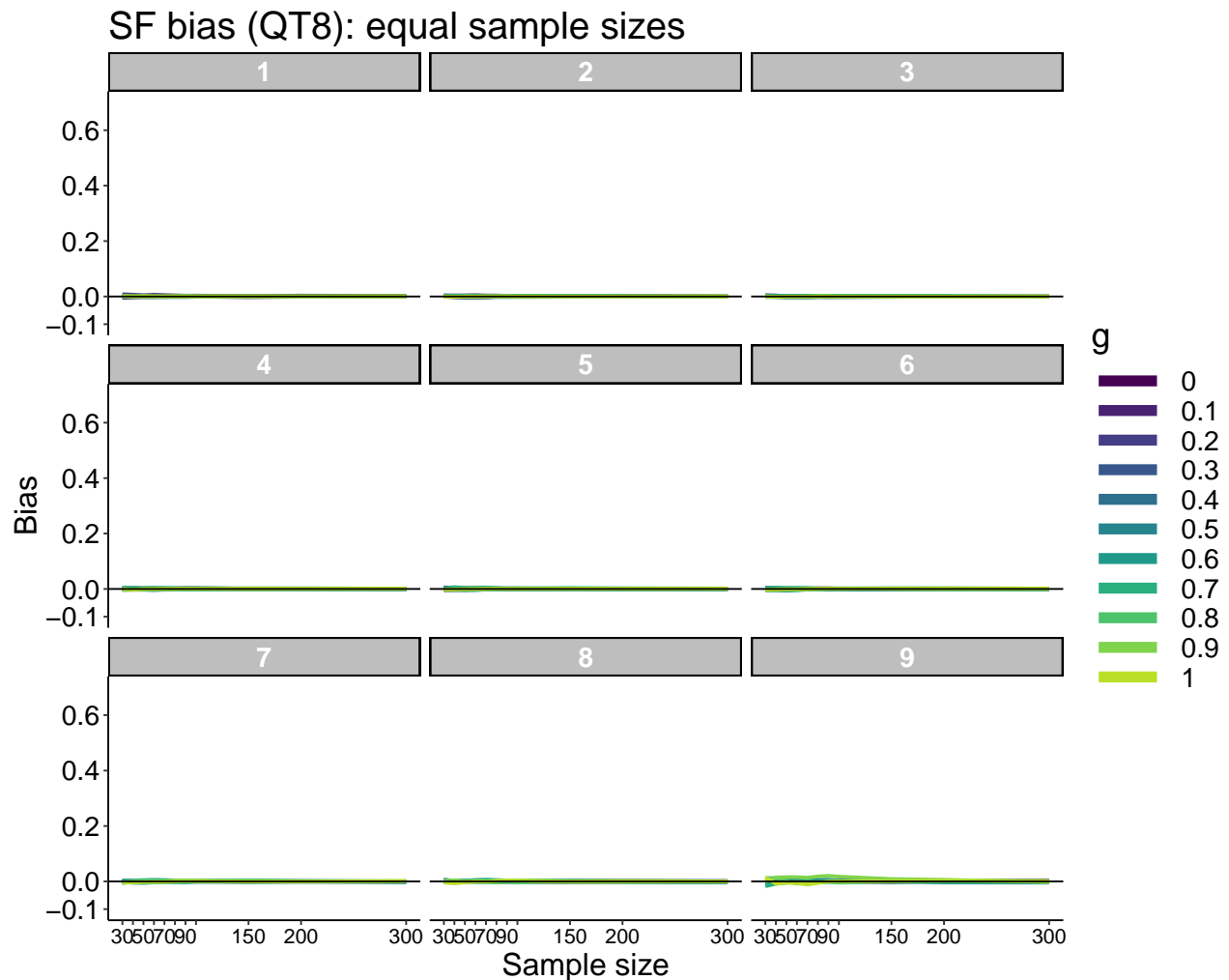
  # make plot
  p <- ggplot(df) + theme_classic() +
    geom_line(aes(x=Size, y=Bias, colour = g), size = 1) +
    geom_abline(intercept=0, slope=0, colour="black") +
    scale_colour_viridis_d(end=0.9) +
    scale_x_continuous(breaks=nvec, labels = xlabels) +
    scale_y_continuous(breaks=c(-0.1,seq(0,0.7,0.2))) +
    coord_cartesian(ylim=c(-0.1,0.7)) +
    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.position = c(0.55,0.85),
          legend.direction = "vertical",
          legend.text=element_text(size=16),
          legend.title=element_text(size=20),
          strip.background=element_rect(fill="grey", colour="black"),
          strip.text=element_text(size=16, colour="white", face="bold")) +
    labs(x = "Sample size", y = "Bias") +
    # make thicker legend lines
    guides(colour = guide_legend(override.aes = list(size=3))) +
    facet_wrap( ~ q, ncol = 3)
  p
}

p <- make_bias_figure(bias.hd, gvec, nvec)
p <- p + ggtitle("SF bias (HD): equal sample sizes")
p
```



Bias is very close to zero in all conditions. Even with 10,000 iterations, variability is noticeably larger for the upper quantiles.

```
p <- make_bias_figure(bias.qt8, gvec, nvec)
p <- p + ggtitle("SF bias (QT8): equal sample sizes")
p
```



Illustrate sampling distributions: HD

Sample size = 30, $q = 1$

```
# fig.width = 10, fig.height = 6
df <- tibble(Difference = as.vector(sim.sfhd.bias[,1,1]),
             g = factor(rep(gvec, each = nsim))
            )

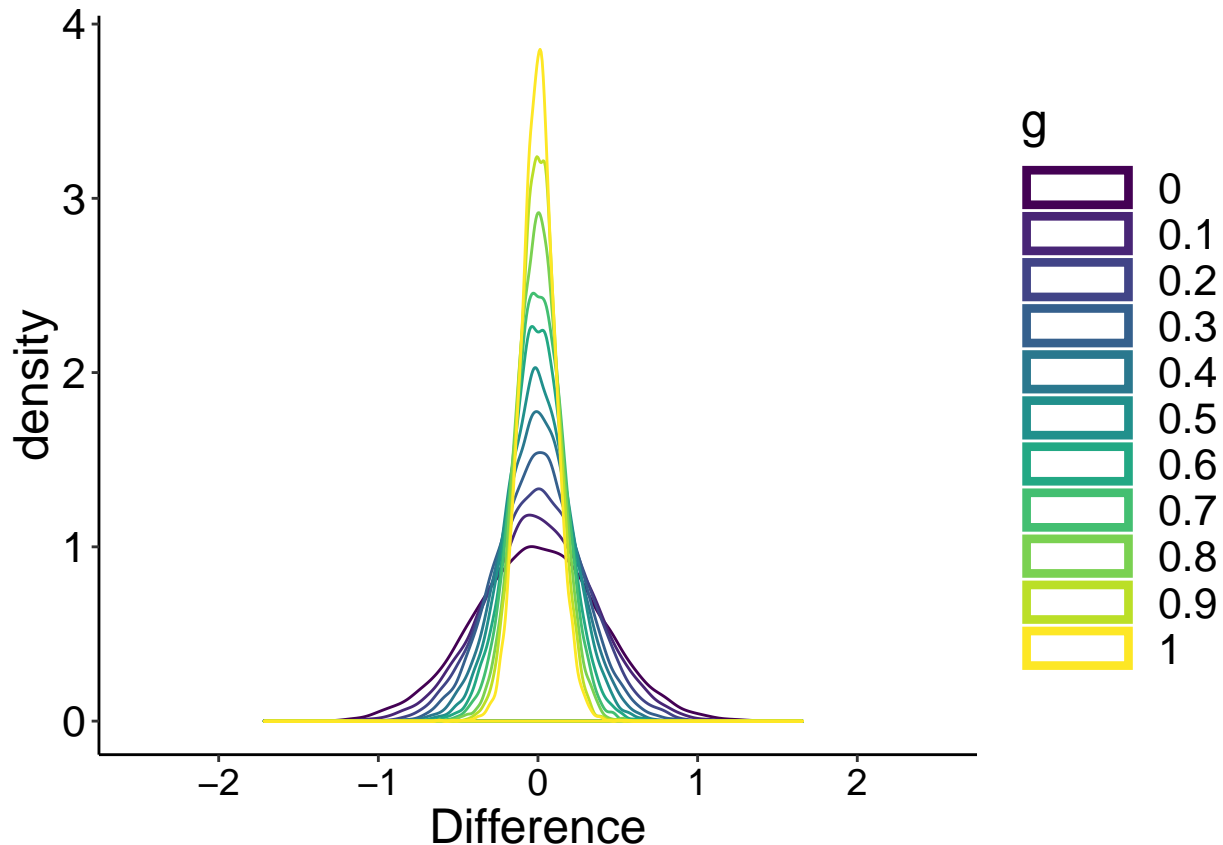
# make plot
p <- ggplot(df, aes(x = Difference, colour = g)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d() +
  # scale_x_continuous(breaks=nvec) +
  # scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
  coord_cartesian(xlim=c(-2.5,2.5)) +
  theme(plot.title = element_text(size=22),
        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 = "right", #c(0.85,0.65),
legend.text=element_text(size=16),
legend.title=element_text(size=18)) +
# labs(x = "Sample size", y = "Bias in ms") +
# make thicker legend lines
guides(colour = guide_legend(override.aes = list(size=3)))
# + ggtitle("Median RT")

```

p



```

# save figure
# ggsave(filename=paste0('figure_miller_bias_md.pdf'),width=10,height=6)

```

Narrower sampling distributions with increasing skewness for the first decile: as variability on the right of the distribution increases, variability on the left decreases.

Sample size = 30, q = 5

```

# fig.width = 10, fig.height = 6
df <- tibble(Difference = as.vector(sim.sfhd.bias[,1,5]),
             g = factor(rep(gvec, each = nsim))
             )

# make plot
p <- ggplot(df, aes(x = Difference, colour = g)) + theme_classic() +

```

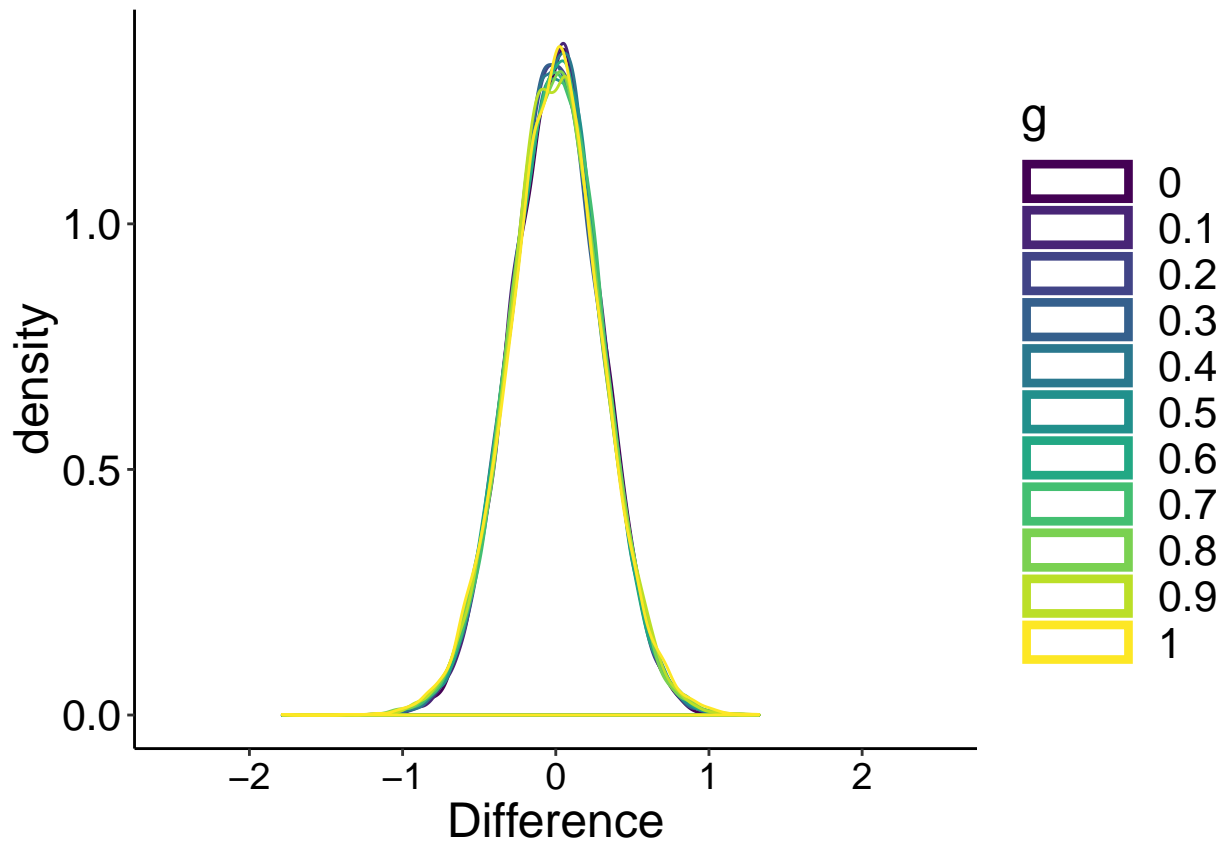


```

geom_density() +
scale_colour_viridis_d() +
# scale_x_continuous(breaks=nvec) +
# scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
coord_cartesian(xlim=c(-2.5,2.5)) +
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 = "right", #c(0.85,0.65),
      legend.text=element_text(size=16),
      legend.title=element_text(size=18)) +
# labs(x = "Sample size", y = "Bias in ms") +
# make thicker legend lines
guides(colour = guide_legend(override.aes = list(size=3)))
# + ggtitle("Median RT")

```

p



```

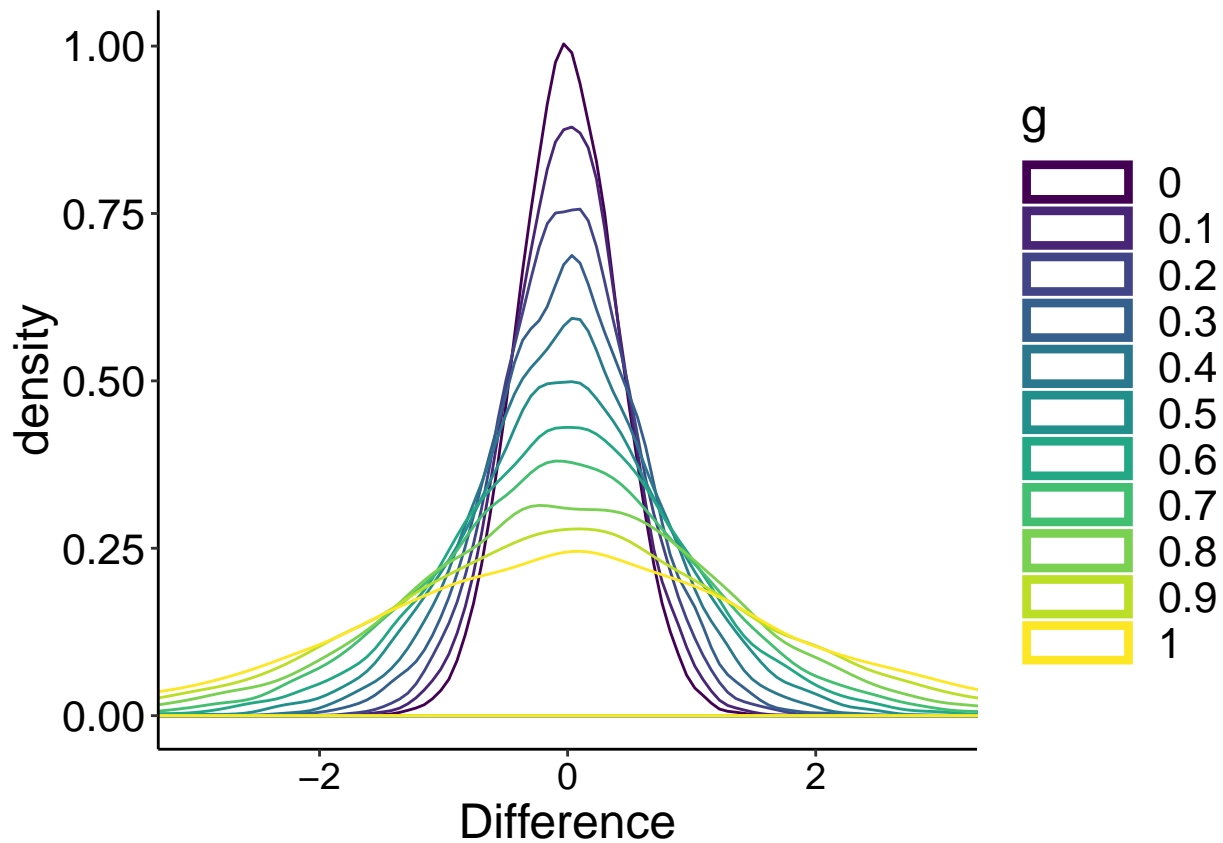
# save figure
# ggsave(filename=paste0('figure_miller_bias_md.pdf'),width=10,height=6)

```

Sample size = 30, q = 9

```
# fig.width = 10, fig.height = 6
df <- tibble(Difference = as.vector(sim.sfhd.bias[, , 1, 9]),
             g = factor(rep(gvec, each = nsim))
             )

# make plot
p <- ggplot(df, aes(x = Difference, colour = g)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d() +
  # scale_x_continuous(breaks=nvec) +
  # scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
  coord_cartesian(xlim=c(-3,3)) +
  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 = "right", #c(0.85,0.65),
        legend.text=element_text(size=16),
        legend.title=element_text(size=18)) +
  # labs(x = "Sample size", y = "Bias in ms") +
  # make thicker legend lines
  guides(colour = guide_legend(override.aes = list(size=3)))
# + ggtitle("Median RT")
p
```



```
# save figure
# ggsave(filename=paste0('figure_miller_bias_md.pdf'),width=10,height=6)
```

Sample size = 30

```
df <- tibble(Difference = as.vector(sim.sfhd.bias[,1,]),
             g = factor(rep(rep(gvec, each = nsim),9)),
             q = factor(rep(seq(1,9), each = nsim*length(gvec)))
             )

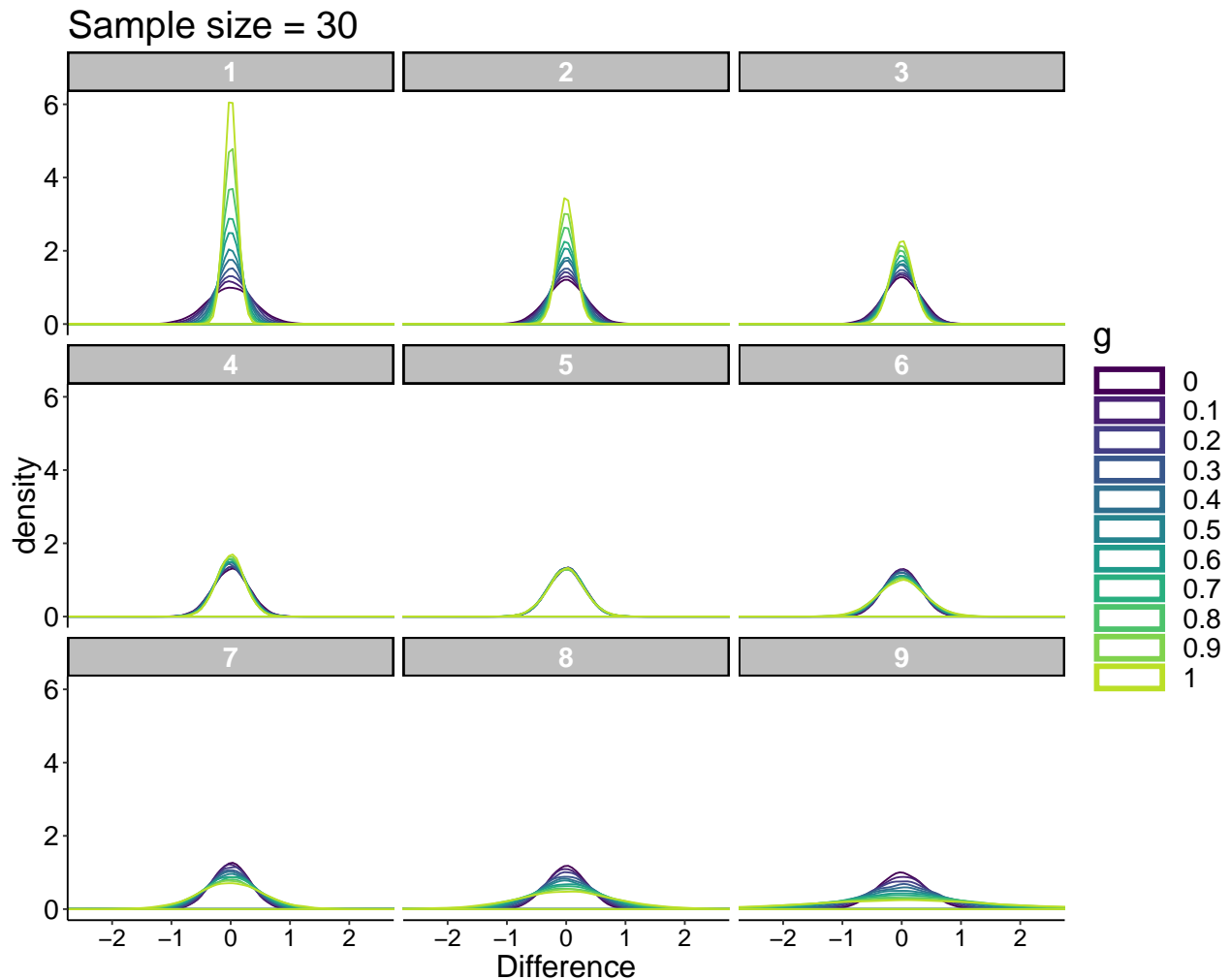
# make plot
p <- ggplot(df, aes(x = Difference, colour = g)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d(end = 0.9) +
  # scale_x_continuous(breaks=nvec) +
  # scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
  coord_cartesian(xlim=c(-2.5,2.5)) +
  theme(plot.title = element_text(size=22),
        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 = "right", #c(0.85,0.65),
        legend.text=element_text(size=16),
        legend.title=element_text(size=20),
```

```

strip.background=element_rect(fill="grey", colour="black"),
strip.text=element_text(size=16, colour="white", face="bold")) +
# labs(x = "Sample size", y = "Bias in ms") +
# make thicker legend lines
guides(colour = guide_legend(override.aes = list(size=3))) +
facet_wrap( ~ q, ncol = 3) +
ggtitle("Sample size = 30")

```

p



Sample size = 300

```

df <- tibble(Difference = as.vector(sim.sfhd.bias[, , 11, ]),
             g = factor(rep(rep(gvec, each = nsim), 9)),
             q = factor(rep(seq(1, 9), each = nsim * length(gvec))))

# make plot
p <- ggplot(df, aes(x = Difference, colour = g)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d(end = 0.9) +

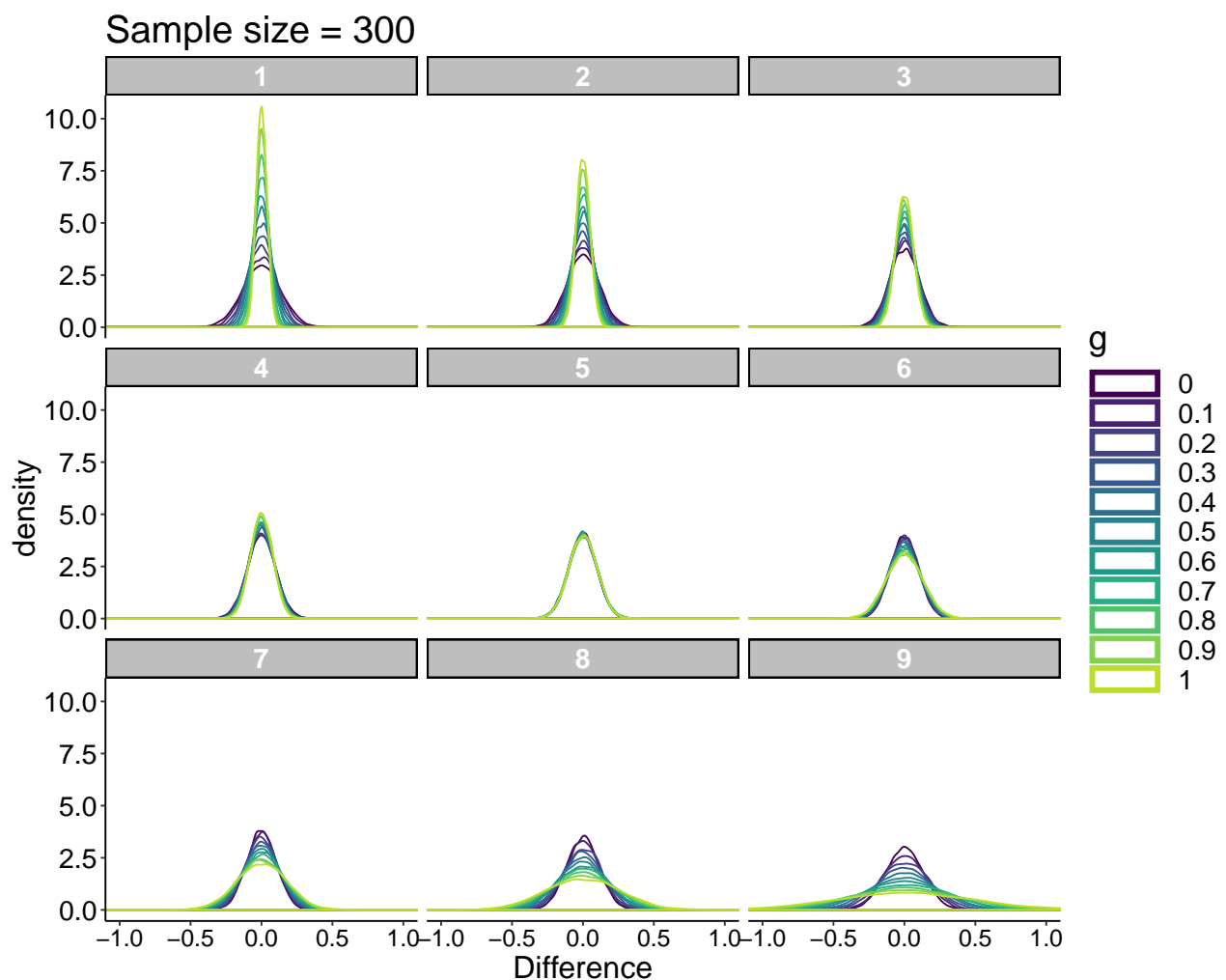
```

```

# scale_x_continuous(breaks=nvec) +
# scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
coord_cartesian(xlim=c(-1,1)) +
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 = "right", #c(0.85,0.65),
      legend.text=element_text(size=16),
      legend.title=element_text(size=20),
      strip.background=element_rect(fill="grey", colour="black"),
      strip.text=element_text(size=16, colour="white", face="bold")) +
# labs(x = "Sample size", y = "Bias in ms") +
# make thicker legend lines
guides(colour = guide_legend(override.aes = list(size=3))) +
facet_wrap( ~ q, ncol = 3) +
ggtitle("Sample size = 300")

```

p



Simulation with different numbers of trials in each condition

10,000 random samples: - condition 1: n trials = c(seq(30, 100, 10), 150, 200) - condition 2: 200 trials

```
gvec <- seq(0, 1, 0.1) # g values
nvec1 <- c(seq(30, 100, 10), 150, 200) # vector of sample sizes to test
nvec2 <- rep(200, length(nvec1)) # number of trials in group 2
nsim <- 10000 # simulation samples
qseq <- seq(0.1, 0.9, 0.1)

# declare matrices of results - save all iterations
sim.sfhd.bias <- array(NA, dim = c(nsim, length(gvec), length(nvec1), 9))
sim.sfqt8.bias <- array(NA, dim = c(nsim, length(gvec), length(nvec1), 9))

set.seed(21)
beep(3)

for(S in 1:nsim){ # simulation iterations

  if(S %% 500 == 0){
    print(paste0("SF bias: simulation ", S, " out of ", nsim, "..."))
    beep(2)
  }

  for(G in 1:length(gvec)){

    large.sample1 <- ghdist(max(nvec1), g = gvec[G], h = 0)
    large.sample2 <- ghdist(max(nvec2), g = gvec[G], h = 0)

    for(N in 1:length(nvec1)){ # sample sizes

      sim.sfhd.bias[S,G,N,] <- hdseq(large.sample1[1:nvec1[N]]) - hdseq(large.sample2[1:nvec2[N]])
      sim.sfqt8.bias[S,G,N,] <- quantile(large.sample1[1:nvec1[N]], probs = qseq, type = 8) -
    }
  }
}

save(
  sim.sfhd.bias,
  sim.sfqt8.bias,
  gvec,
  nvec1,
  nvec2,
  nsim,
  file=paste0('./data/sim_sf_bias2.RData'))
beep(8)
```

Compute bias

Bias = average of sample estimates minus population value.

```
load('./data/sim_sf_bias2.RData')
bias.hd <- apply(sim.sfhd.bias[, , 1:length(nvec1), ], c(2,3,4), mean, na.rm = TRUE)
bias.hd.md <- apply(sim.sfhd.bias[, , 1:length(nvec1), ], c(2,3,4), median, na.rm = TRUE)
```

```

bias.qt8 <- apply(sim.sfqt8.bias[,1:length(nvec1)], c(2,3,4), mean, na.rm = TRUE)
bias.qt8.md <- apply(sim.sfqt8.bias[,1:length(nvec1)], c(2,3,4), median, na.rm = TRUE)
xlabels <- paste(c(30,"",50,"",70,"",90,"",150, 200))

```

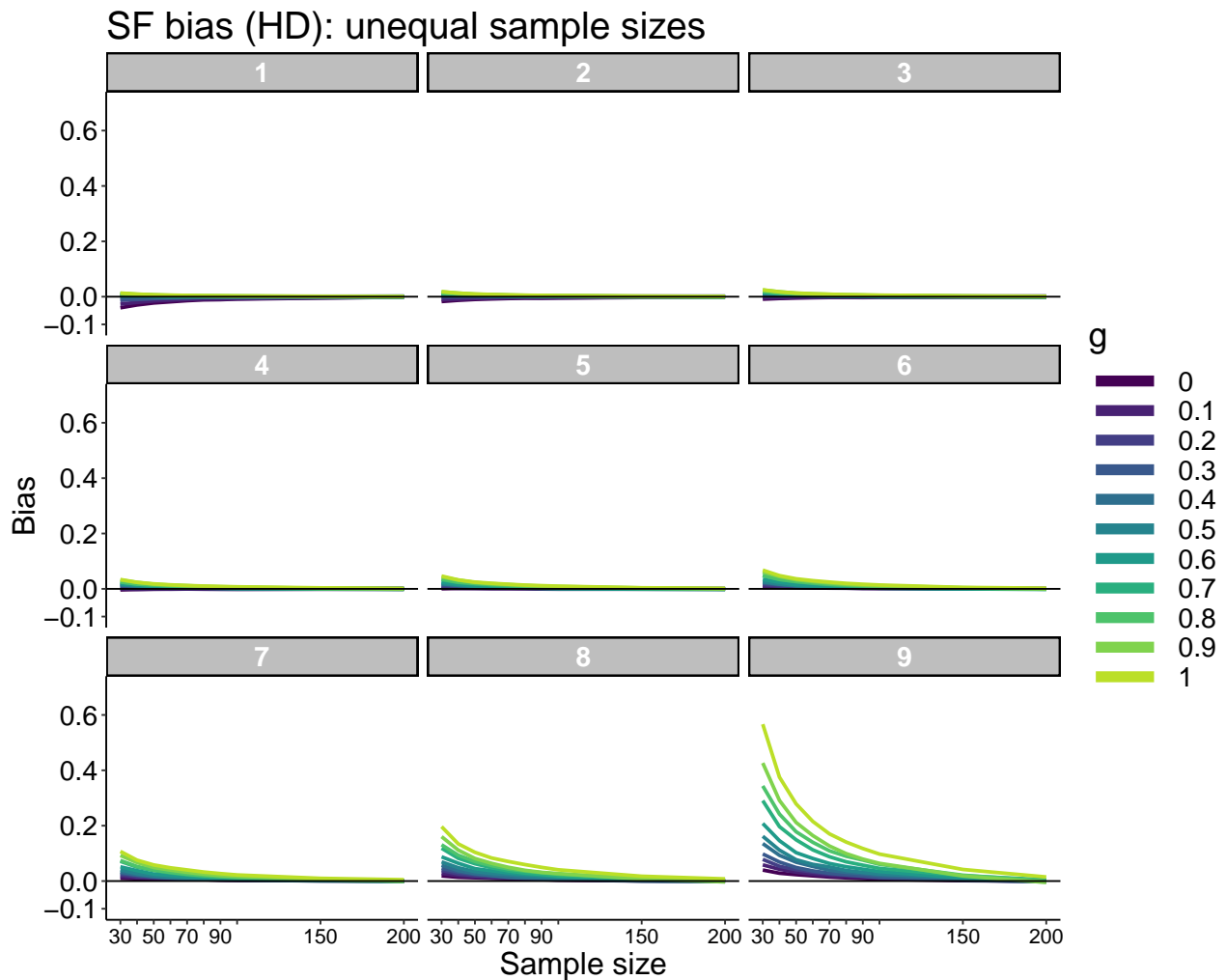
Illustrate bias results

HD results

```

p <- make_bias_figure(bias.hd, gvec, nvec1)
p <- p + ggtitle("SF bias (HD): unequal sample sizes")
p

```

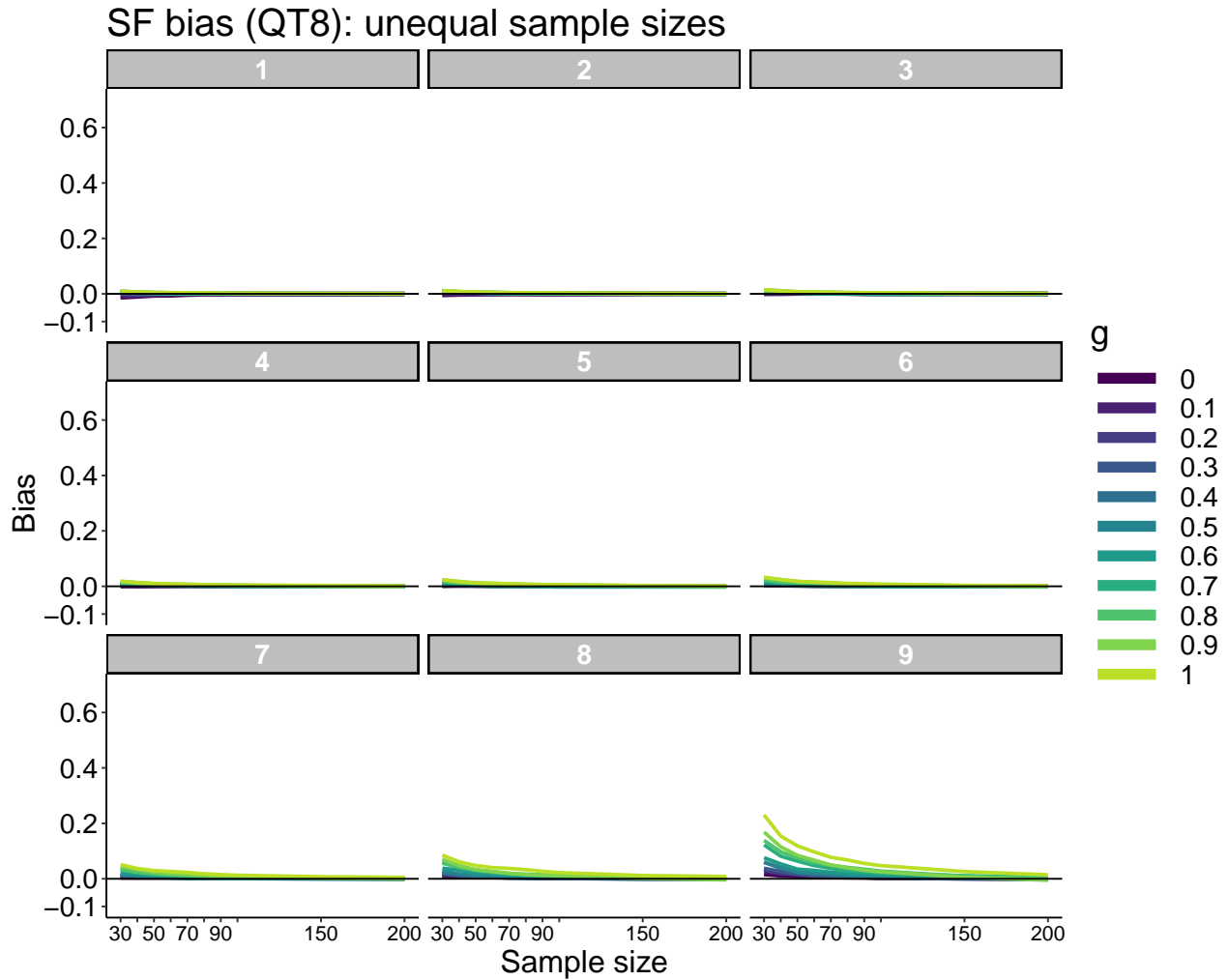


For skewed distributions the bias is positive, as expected: with increased skewness, the smaller number of trials in group 1 leads to over-estimation of deciles, such that the difference between group 1 and group 2 is positive.

$n = 100$ appears sufficient to minimise bias for all g , except for deciles 8 and 9.

QT8 results

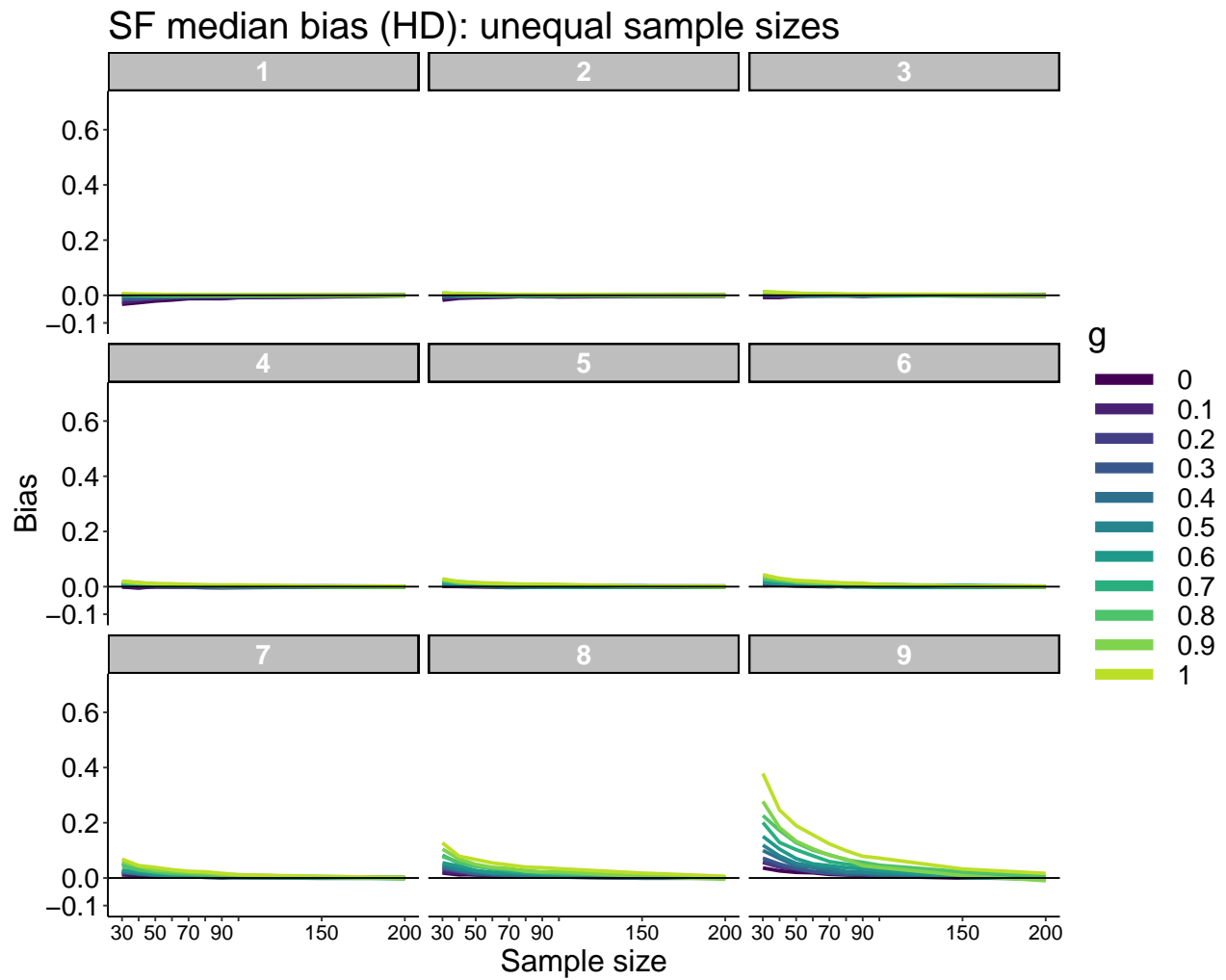
```
p <- make_bias_figure(bias.qt8, gvec, nvec1)
p <- p + ggtitle("SF bias (QT8): unequal sample sizes")
p
```



Illustrate bias results: median bias

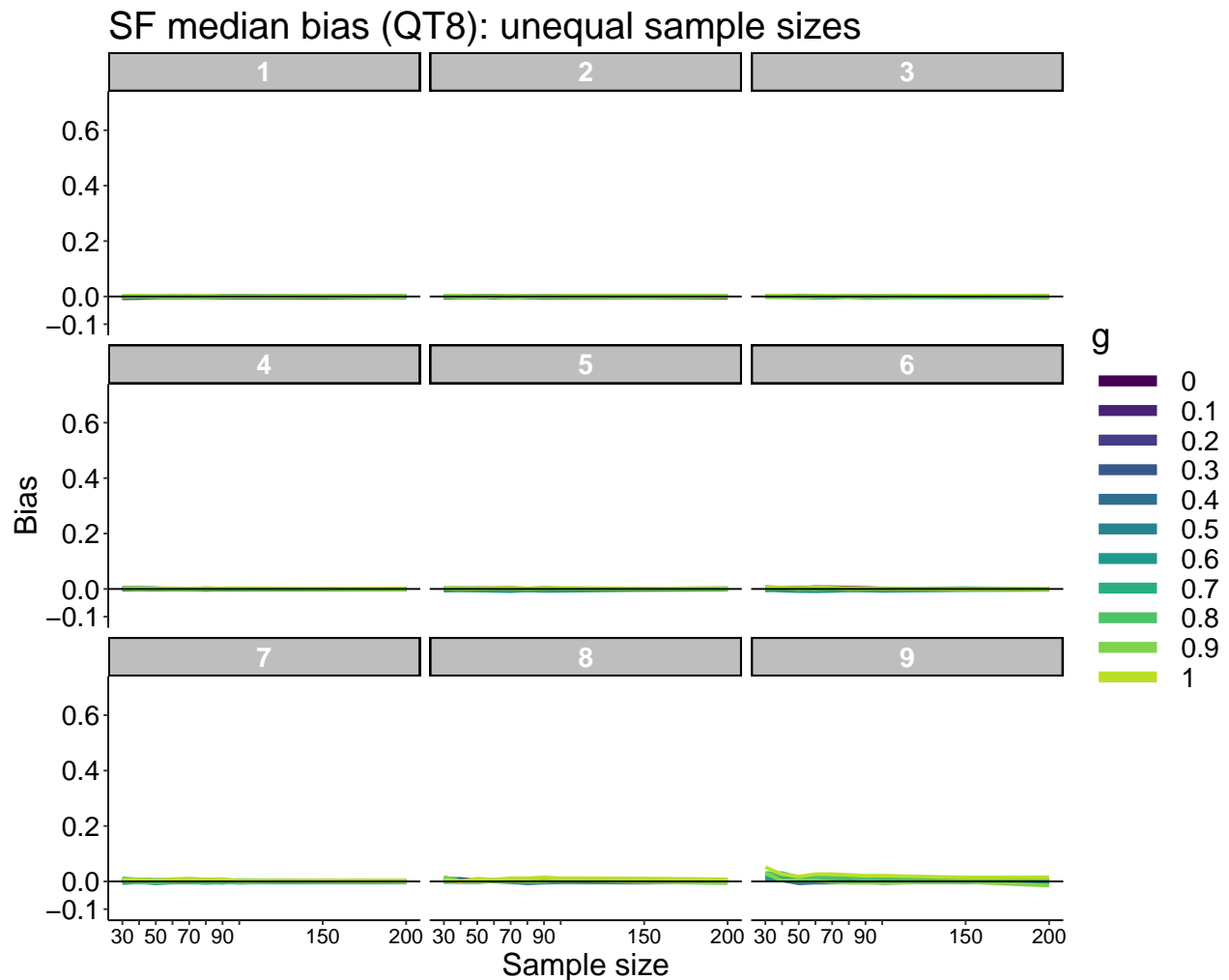
HD results

```
p <- make_bias_figure(bias.hd.md, gvec, nvec1)
p <- p + ggtitle("SF median bias (HD): unequal sample sizes")
p
```

QT8 results

```
p <- make_bias_figure(bias.qt8.md, gvec, nvec1)
p <- p + ggtitle("SF median bias (QT8): unequal sample sizes")
p
```



Illustrate sampling distributions: HD

Sample size = 30, $q = 1$

```
# fig.width = 10, fig.height = 6
df <- tibble(Difference = as.vector(sim.sfhd.bias[,1,1]),
             g = factor(rep(gvec, each = nsim))
            )

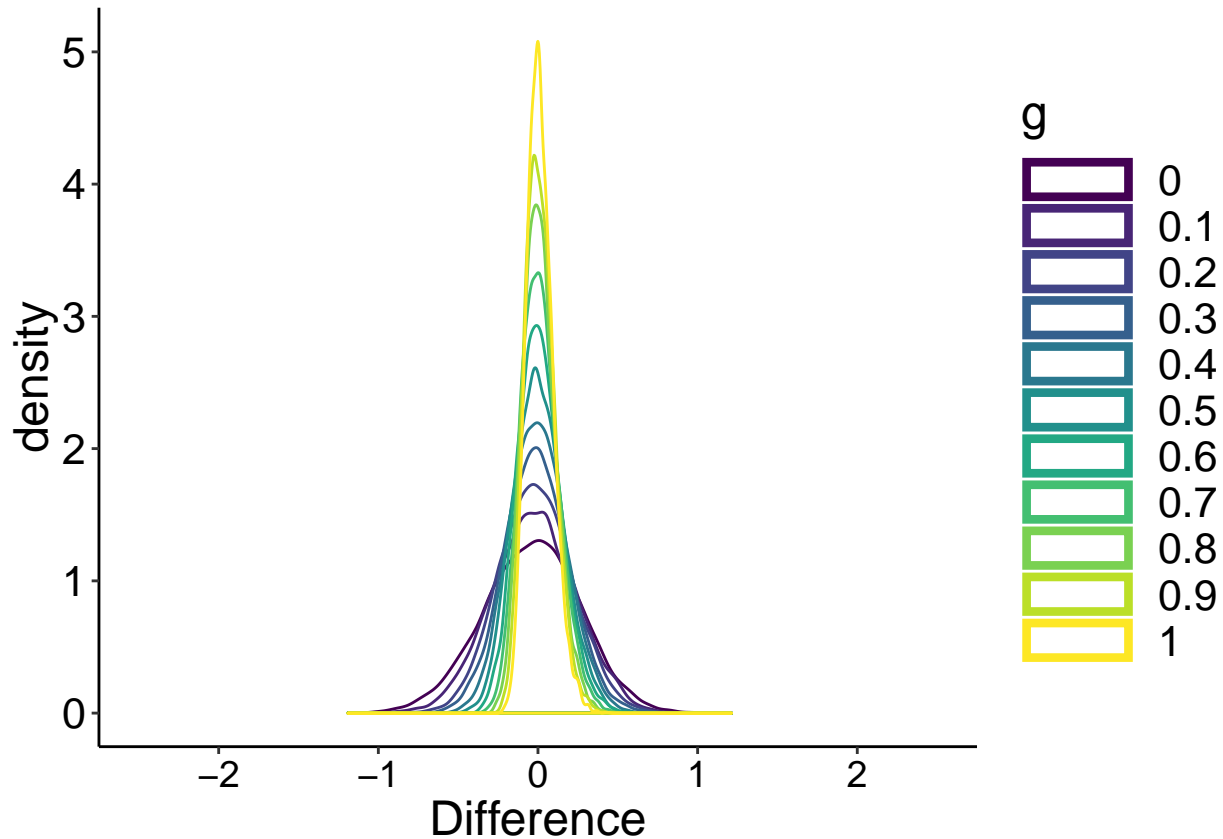
# make plot
p <- ggplot(df, aes(x = Difference, colour = g)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d() +
  # scale_x_continuous(breaks=nvec) +
  # scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
  coord_cartesian(xlim=c(-2.5,2.5)) +
  theme(plot.title = element_text(size=22),
        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 = "right", #c(0.85,0.65),
legend.text=element_text(size=16),
legend.title=element_text(size=18)) +
# labs(x = "Sample size", y = "Bias in ms") +
# make thicker legend lines
guides(colour = guide_legend(override.aes = list(size=3)))
# + ggtitle("Median RT")

```

p



```

# save figure
# ggsave(filename=paste0('figure_miller_bias_md.pdf'),width=10,height=6)

```

Sample size = 30, q = 5

```

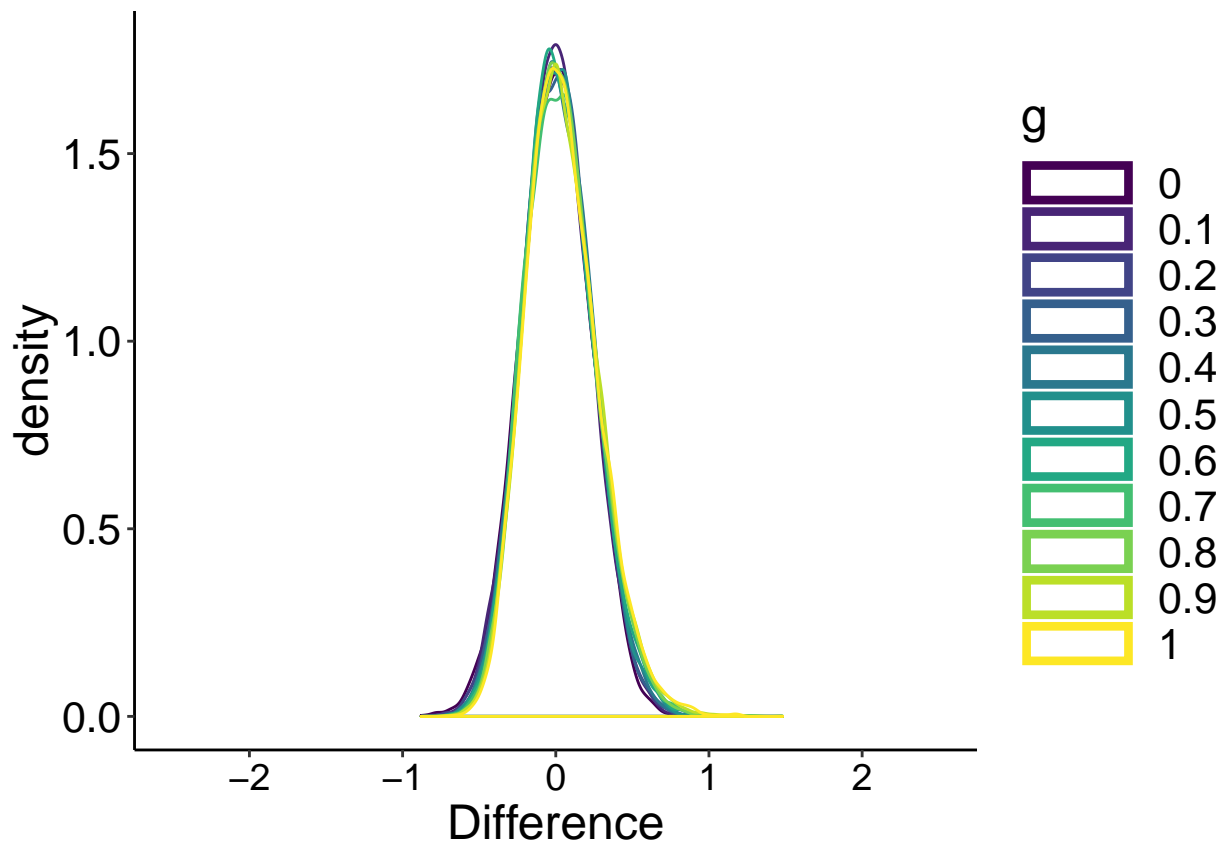
# fig.width = 10, fig.height = 6
df <- tibble(Difference = as.vector(sim.sfhd.bias[,1,5]),
             g = factor(rep(gvec, each = nsim))
             )

# make plot
p <- ggplot(df, aes(x = Difference, colour = g)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d() +
  # scale_x_continuous(breaks=nvec) +

```

```
# scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
coord_cartesian(xlim=c(-2.5,2.5)) +
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 = "right", #c(0.85,0.65),
      legend.text=element_text(size=16),
      legend.title=element_text(size=18)) +
# labs(x = "Sample size", y = "Bias in ms") +
# make thicker legend lines
guides(colour = guide_legend(override.aes = list(size=3)))
# + ggtitle("Median RT")
```

p



```
# save figure
# ggsave(filename=paste0('figure_miller_bias_md.pdf'),width=10,height=6)
```

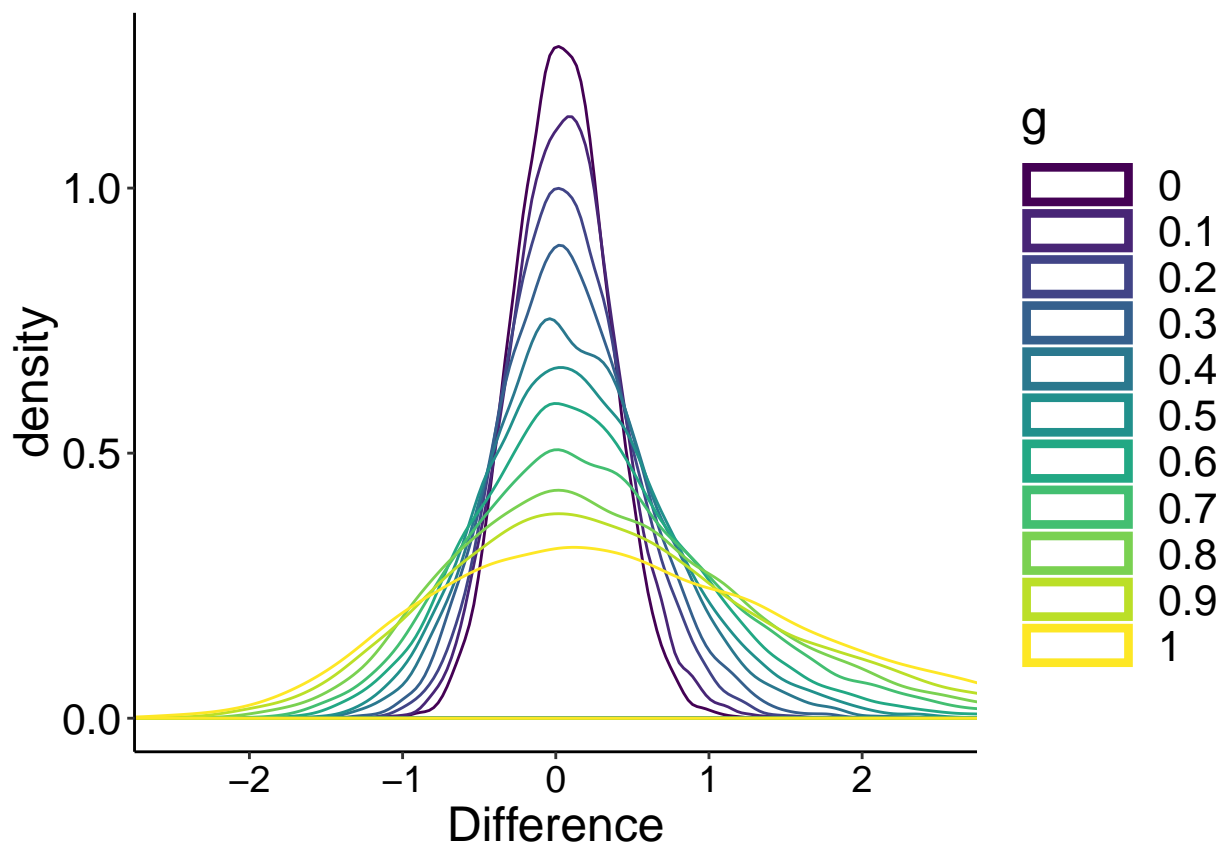
Sample size = 30, q = 9

```
# fig.width = 10, fig.height = 6
df <- tibble(Difference = as.vector(sim.sfhd.bias[,1,9]),
             g = factor(rep(gvec, each = nsim))
            )
```

```

# make plot
p <- ggplot(df, aes(x = Difference, colour = g)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d() +
  # scale_x_continuous(breaks=nvec) +
  # scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
  coord_cartesian(xlim=c(-2.5,2.5)) +
  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 = "right", #c(0.85,0.65),
        legend.text=element_text(size=16),
        legend.title=element_text(size=18)) +
  # labs(x = "Sample size", y = "Bias in ms") +
  # make thicker legend lines
  guides(colour = guide_legend(override.aes = list(size=3)))
# + ggtitle("Median RT")
p

```



```

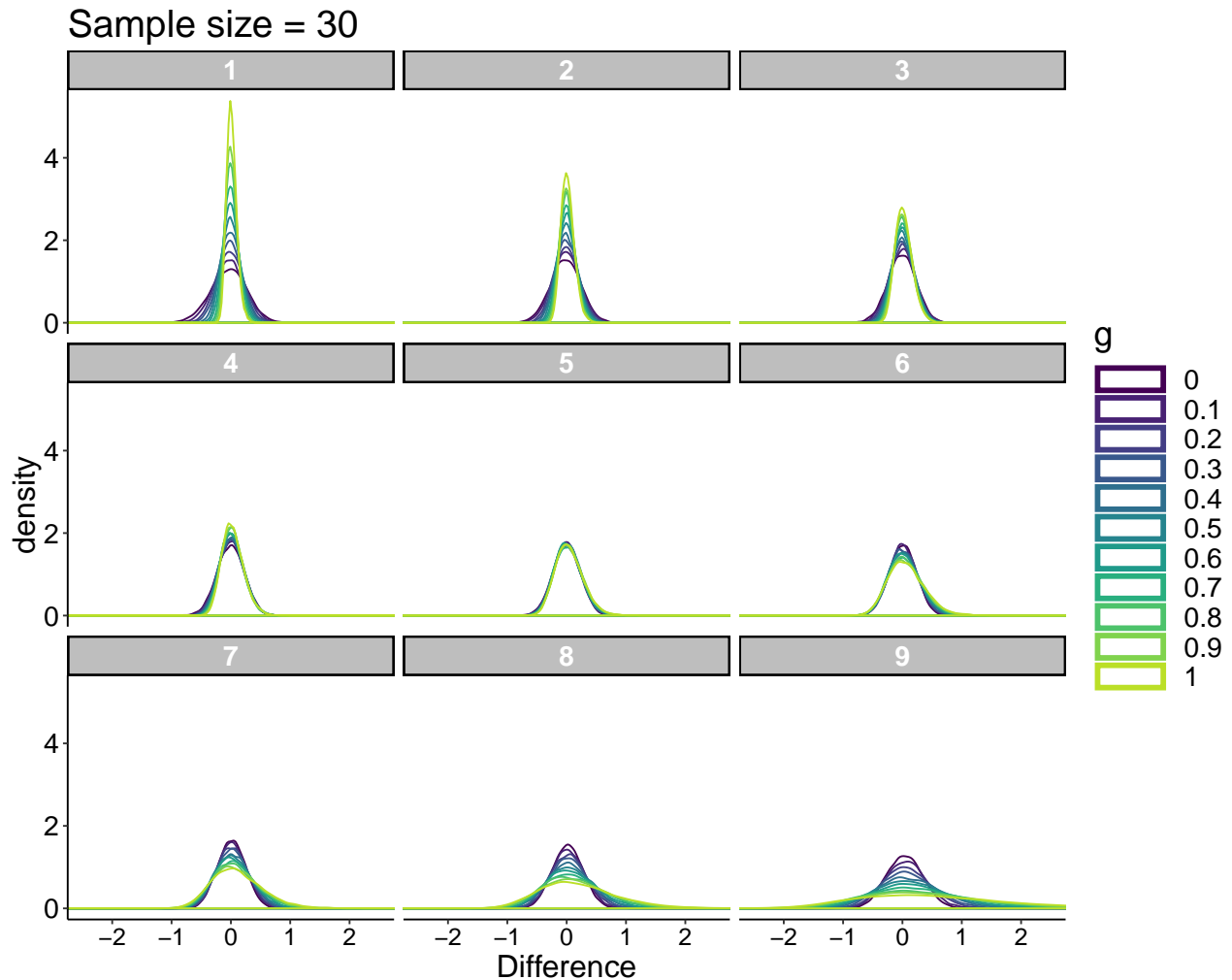
# save figure
# ggsave(filename=paste0('figure_miller_bias_md.pdf'),width=10,height=6)

```

Sample size = 30

```
df <- tibble(Difference = as.vector(sim.sfhd.bias[,1]),
             g = factor(rep(rep(gvec, each = nsim),9)),
             q = factor(rep(seq(1,9), each = nsim*length(gvec)))
             )

# make plot
p <- ggplot(df, aes(x = Difference, colour = g)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d(end=0.9) +
  coord_cartesian(xlim=c(-2.5,2.5)) +
  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 = "right",
        legend.text=element_text(size=16),
        legend.title=element_text(size=20),
        strip.background=element_rect(fill="grey", colour="black"),
        strip.text=element_text(size=16, colour="white", face="bold")) +
  guides(colour = guide_legend(override.aes = list(size=3))) +
  facet_wrap( ~ q, ncol = 3) +
  ggtitle("Sample size = 30")
p
```



Sample size = 100

```
df <- tibble(Difference = as.vector(sim.sfhd.bias[,10,]),
             g = factor(rep(rep(gvec, each = nsim),9)),
             q = factor(rep(seq(1,9), each = nsim*length(gvec))))

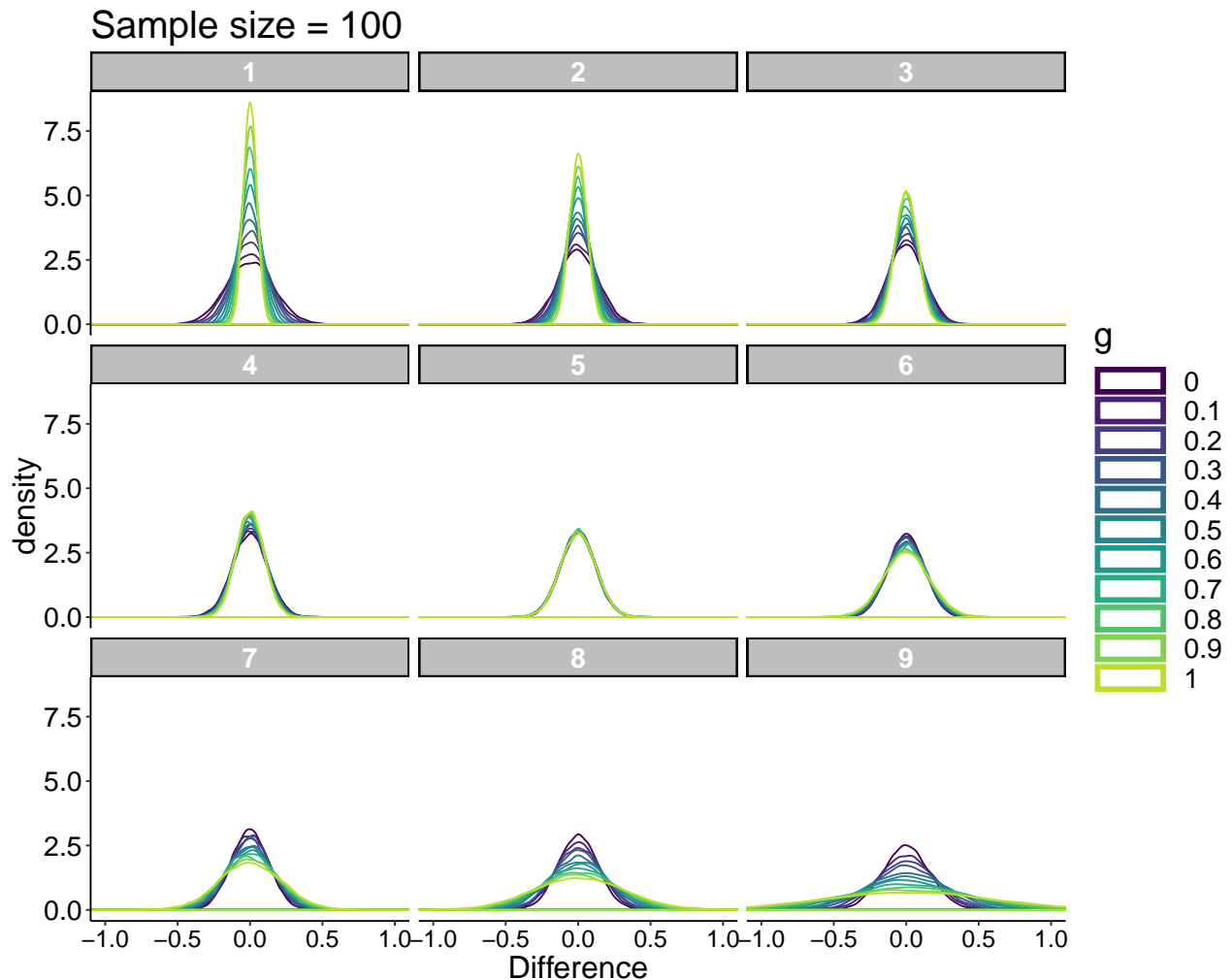
# make plot
p <- ggplot(df, aes(x = Difference, colour = g)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d(end=0.9) +
  coord_cartesian(xlim=c(-1,1)) +
  theme(plot.title = element_text(size=22),
        axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 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 = "right",
        legend.text=element_text(size=16),
```

```

legend.title=element_text(size=20),
strip.background=element_rect(fill="grey", colour="black"),
strip.text=element_text(size=16, colour="white", face="bold")) +
# labs(x = "Sample size", y = "Bias in ms") +
# make thicker legend lines
guides(colour = guide_legend(override.aes = list(size=3))) +
facet_wrap( ~ q, ncol = 3) +
ggtitle("Sample size = 100")

```

p



Illustrate sampling distributions: QT8

Sample size = 30, q = 1

```

# fig.width = 10, fig.height = 6
df <- tibble(Difference = as.vector(sim.sfqt8.bias[,1,1]),
             g = factor(rep(gvec, each = nsim))
             )

# make plot

```

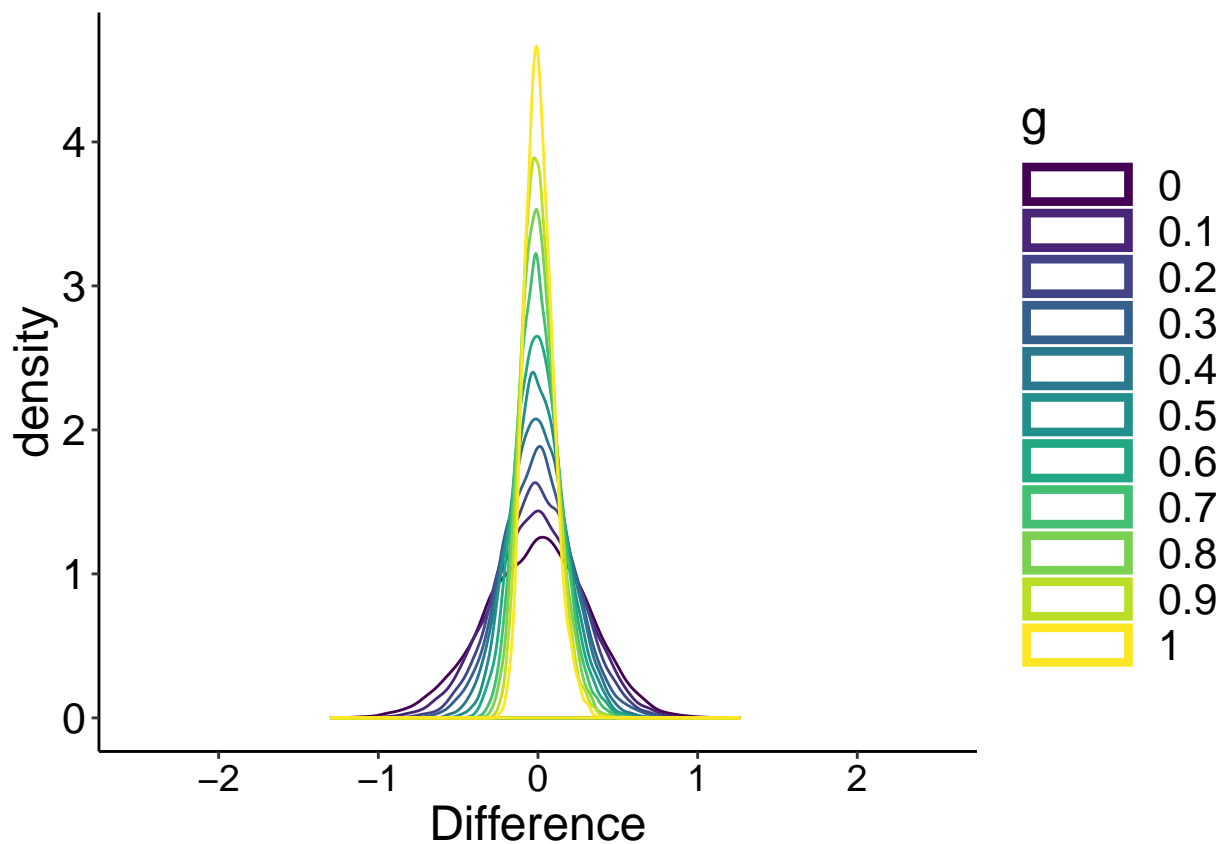


```

p <- ggplot(df, aes(x = Difference, colour = g)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d() +
  # scale_x_continuous(breaks=nvec) +
  # scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
  coord_cartesian(xlim=c(-2.5,2.5)) +
  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 = "right", #c(0.85,0.65),
        legend.text=element_text(size=16),
        legend.title=element_text(size=18)) +
  # labs(x = "Sample size", y = "Bias in ms") +
  # make thicker legend lines
  guides(colour = guide_legend(override.aes = list(size=3)))
# + ggtitle("Median RT")

```

p



```

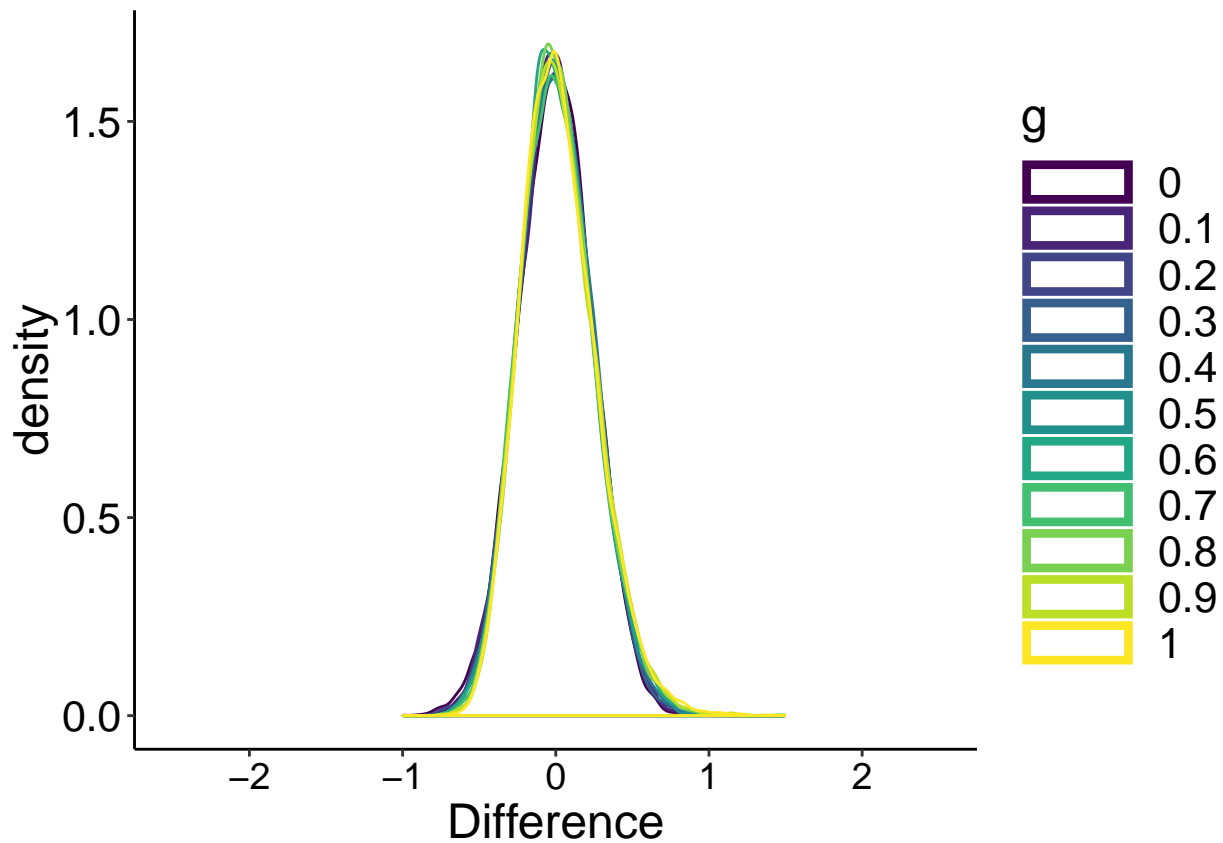
# save figure
# ggsave(filename=paste0('figure_miller_bias_md.pdf'),width=10,height=6)

```

Sample size = 30, q = 5

```
# fig.width = 10, fig.height = 6
df <- tibble(Difference = as.vector(sim.sfqt8.bias[,1,5]),
             g = factor(rep(gvec, each = nsim))
             )

# make plot
p <- ggplot(df, aes(x = Difference, colour = g)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d() +
  # scale_x_continuous(breaks=nvec) +
  # scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
  coord_cartesian(xlim=c(-2.5,2.5)) +
  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 = "right", #c(0.85,0.65),
        legend.text=element_text(size=16),
        legend.title=element_text(size=18)) +
  # labs(x = "Sample size", y = "Bias in ms") +
  # make thicker legend lines
  guides(colour = guide_legend(override.aes = list(size=3)))
# + ggtitle("Median RT")
p
```



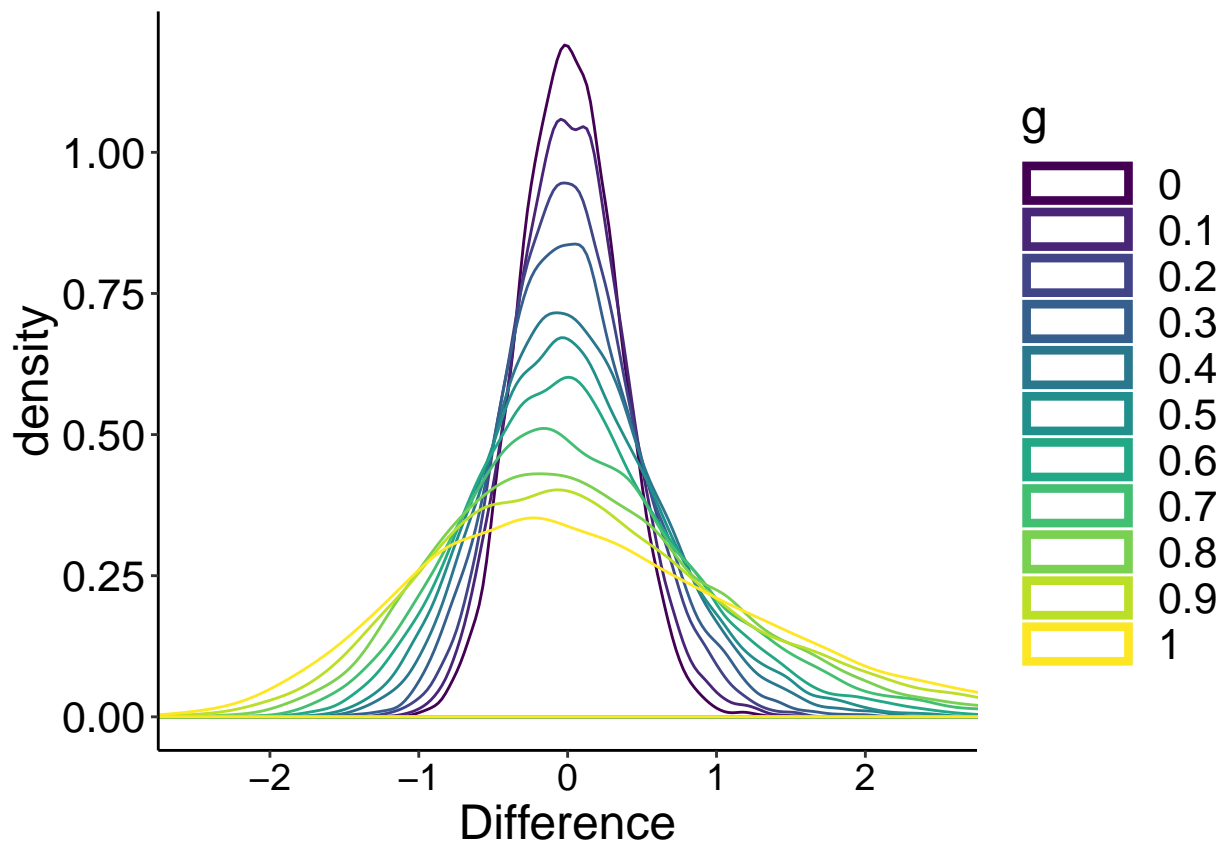
```
# save figure
# ggsave(filename=paste0('figure_miller_bias_md.pdf'),width=10,height=6)
```

Sample size = 30, q = 9

```
# fig.width = 10, fig.height = 6
df <- tibble(Difference = as.vector(sim.sfqt8.bias[,1,9]),
             g = factor(rep(gvec, each = nsim))
)

# make plot
p <- ggplot(df, aes(x = Difference, colour = g)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d() +
  # scale_x_continuous(breaks=nvec) +
  # scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
  coord_cartesian(xlim=c(-2.5,2.5)) +
  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 = "right", #c(0.85,0.65),
        legend.text=element_text(size=16),
        legend.title=element_text(size=18)) +
```

```
# labs(x = "Sample size", y = "Bias in ms") +
# make thicker legend lines
guides(colour = guide_legend(override.aes = list(size=3)))
p
```



Sample size = 30

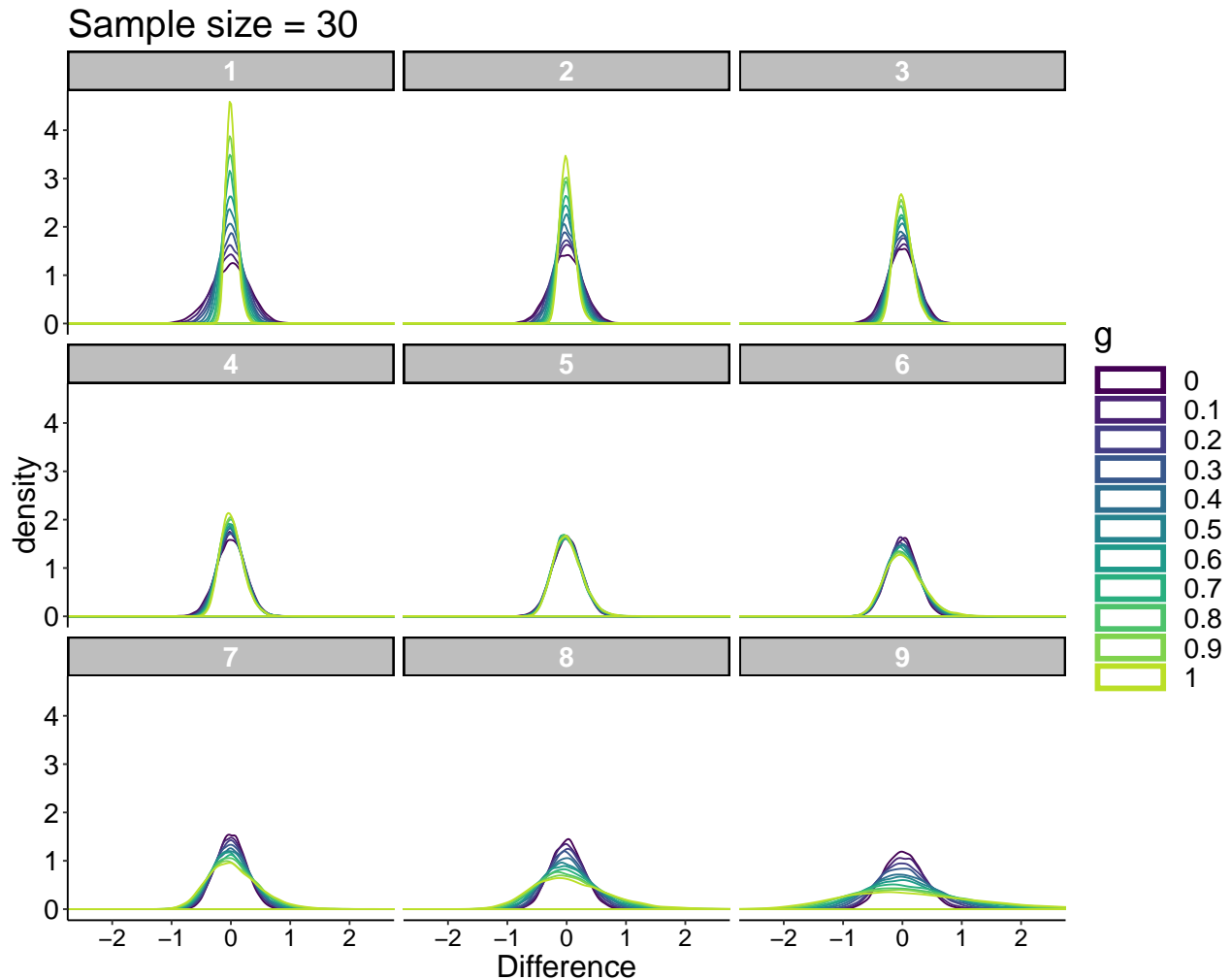
```
df <- tibble(Difference = as.vector(sim.sfqt8.bias[,1,]),
             g = factor(rep(rep(gvec, each = nsim), 9)),
             q = factor(rep(seq(1,9), each = nsim*length(gvec)))
             )

# make plot
p <- ggplot(df, aes(x = Difference, colour = g)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d(end=0.9) +
  coord_cartesian(xlim=c(-2.5,2.5)) +
  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 = "right",
        legend.text=element_text(size=16),
```

```

    legend.title=element_text(size=20),
    strip.background=element_rect(fill="grey", colour="black"),
    strip.text=element_text(size=16, colour="white", face="bold")) +
  guides(colour = guide_legend(override.aes = list(size=3))) +
  facet_wrap( ~ q, ncol = 3) +
  ggtitle("Sample size = 30")
p

```



Sample size = 100

```

df <- tibble(Difference = as.vector(sim.sfqt8.bias[, ,10,]),
             g = factor(rep(rep(gvec, each = nsim),9)),
             q = factor(rep(seq(1,9), each = nsim*length(gvec))))
)

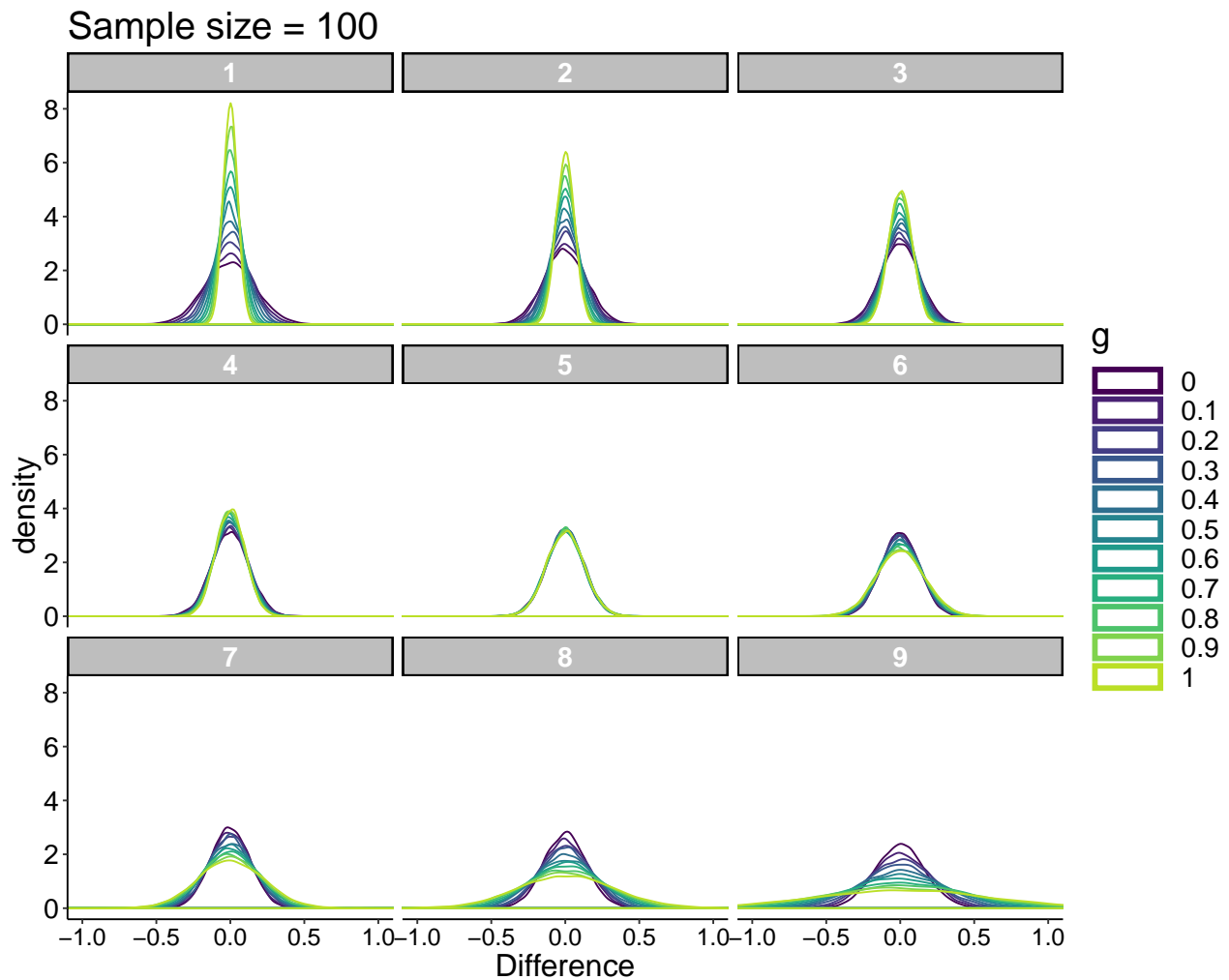
# make plot
p <- ggplot(df, aes(x = Difference, colour = g)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d(end=0.9) +
  coord_cartesian(xlim=c(-1,1)) +

```

```

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 = "right",
      legend.text=element_text(size=16),
      legend.title=element_text(size=20),
      strip.background=element_rect(fill="grey", colour="black"),
      strip.text=element_text(size=16, colour="white", face="bold")) +
# labs(x = "Sample size", y = "Bias in ms") +
# make thicker legend lines
guides(colour = guide_legend(override.aes = list(size=3))) +
facet_wrap( ~ q, ncol = 3) +
ggtitle("Sample size = 100")
p

```



Simulation with varying numbers of trials in both groups

Goal: to evaluate bias for decile 9 only, as a function of the absolute number of trials in group 1 and group 2.

10,000 random samples: - condition 1: n trials = c(seq(50, 100, 10), 150, 200, 300) - condition 2: n trials = cond1 + seq(0, 100, 10)

```
gvec <- seq(0, 1, 0.1) # g values
nvec1 <- c(seq(50, 100, 10), 150, 200, 300) # vector of sample sizes to test
nvec2 <- seq(0, 100, 10) # number of trials in group 2 + nvec1
nsim <- 10000 # simulation samples

# declare matrices of results - save all iterations
sim.sfhd.bias <- array(NA, dim = c(nsim, length(gvec), length(nvec2), length(nvec1)))
sim.sfqt8.bias <- array(NA, dim = c(nsim, length(gvec), length(nvec2), length(nvec1)))

set.seed(21)

for(S in 1:nsim){ # simulation iterations

  if(S %% 500 == 0){
    print(paste0("Decile 9 bias: simulation ",S," out of ",nsim,"..."))
  }

  for(G in 1:length(gvec)){

    large.sample1 <- ghdist(max(nvec1), g = gvec[G], h = 0)
    large.sample2 <- ghdist(max(nvec1)+max(nvec2), g = gvec[G], h = 0)

    for(N1 in 1:length(nvec1)){ # sample sizes

      HD1 <- hd(large.sample1[1:nvec1[N1]], q = 0.9)
      QT1 <- quantile(large.sample1[1:nvec1[N1]], probs = 0.9, type = 8)

      for(N2 in 1:length(nvec2)){ # sample sizes

        sim.sfhd.bias[S,G,N2,N1] <- HD1 - hd(large.sample2[1:(nvec1[N1]+nvec2[N2])], q = 0.9)
        sim.sfqt8.bias[S,G,N2,N1] <- QT1 - quantile(large.sample2[1:(nvec1[N1]+nvec2[N2])], probs = 0.9)
      }
    }
  }
}

save(
  sim.sfhd.bias,
  sim.sfqt8.bias,
  gvec,
  nvec1,
  nvec2,
  nsim,
  file=paste0('./data/sim_sf_bias3.RData'))
```

Compute bias

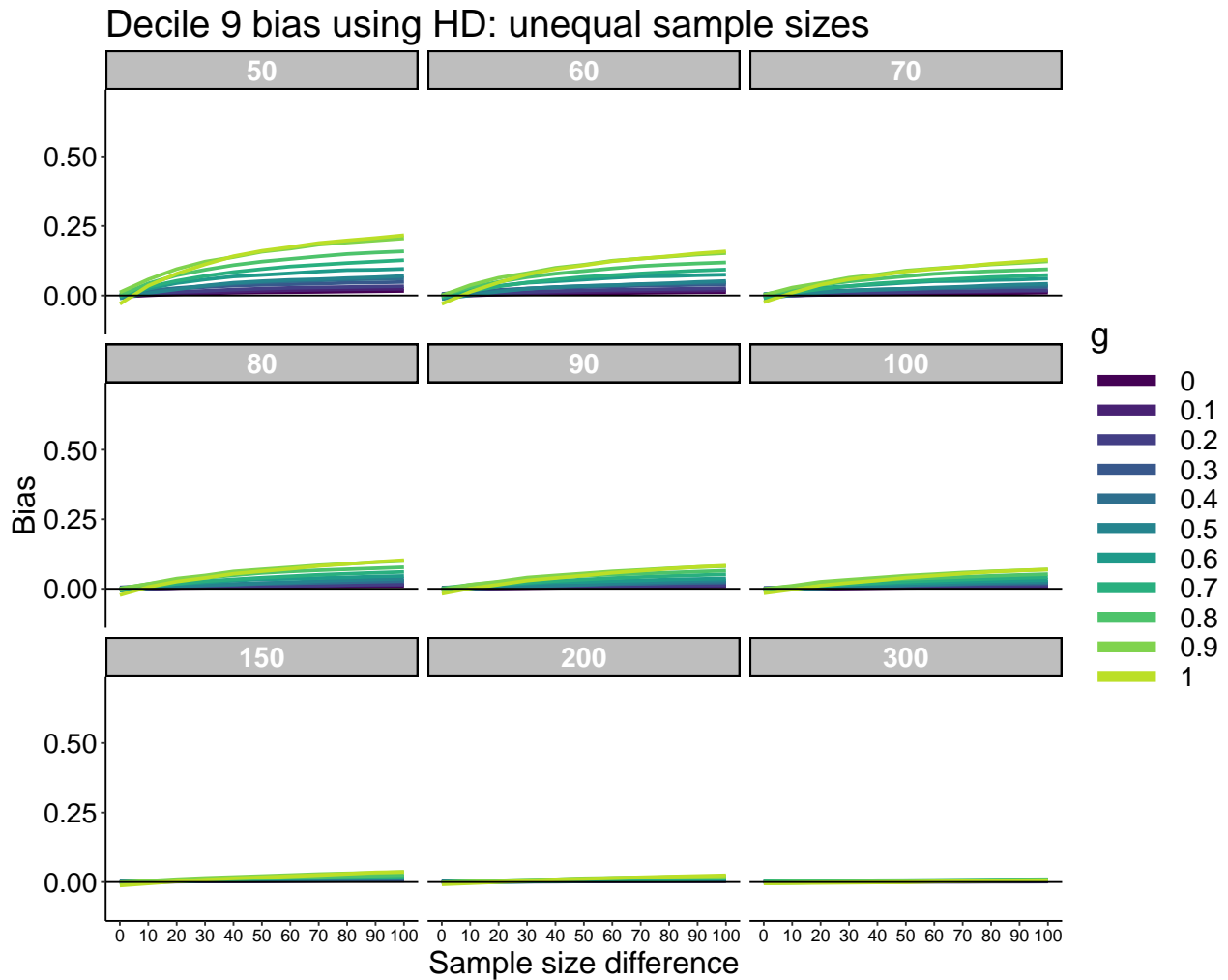
Bias = average of sample estimates minus population value.

```
load('./data/sim_sf_bias3.RData')
bias.hd <- apply(sim.sfhd.bias, c(2,3,4), mean, na.rm = TRUE)
bias.qt8 <- apply(sim.sfqt8.bias, c(2,3,4), mean, na.rm = TRUE)
```

Illustrate bias results: HD

```
df <- tibble(Bias = as.vector(bias.hd),
             g = factor(rep(gvec, length(nvec2)*length(nvec1))),
             Size = rep(rep(nvec2, each=length(gvec)), length(nvec1)),
             q = factor(rep(nvec1, each=length(gvec)*length(nvec2))))

# make plot
p <- ggplot(df) + theme_classic() +
  geom_line(aes(x=Size, y=Bias, colour = g), size = 1) +
  geom_abline(intercept=0, slope=0, colour="black") +
  scale_colour_viridis_d(end=0.9) +
  scale_x_continuous(breaks=nvec2) +
  # scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
  coord_cartesian(ylim=c(-0.1,0.7)) +
  theme(plot.title = element_text(size=22),
        axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 10, 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.position = c(0.55,0.85),
        legend.direction = "vertical",
        legend.text=element_text(size=16),
        legend.title=element_text(size=20),
        strip.background=element_rect(fill="grey", colour="black"),
        strip.text=element_text(size=16, colour="white", face="bold")) +
  labs(x = "Sample size difference", y = "Bias") +
  # make thicker legend lines
  guides(colour = guide_legend(override.aes = list(size=3))) +
  facet_wrap( ~ q, ncol = 3) +
  ggtitle("Decile 9 bias using HD: unequal sample sizes")
p
```

Illustrate bias results: QT8

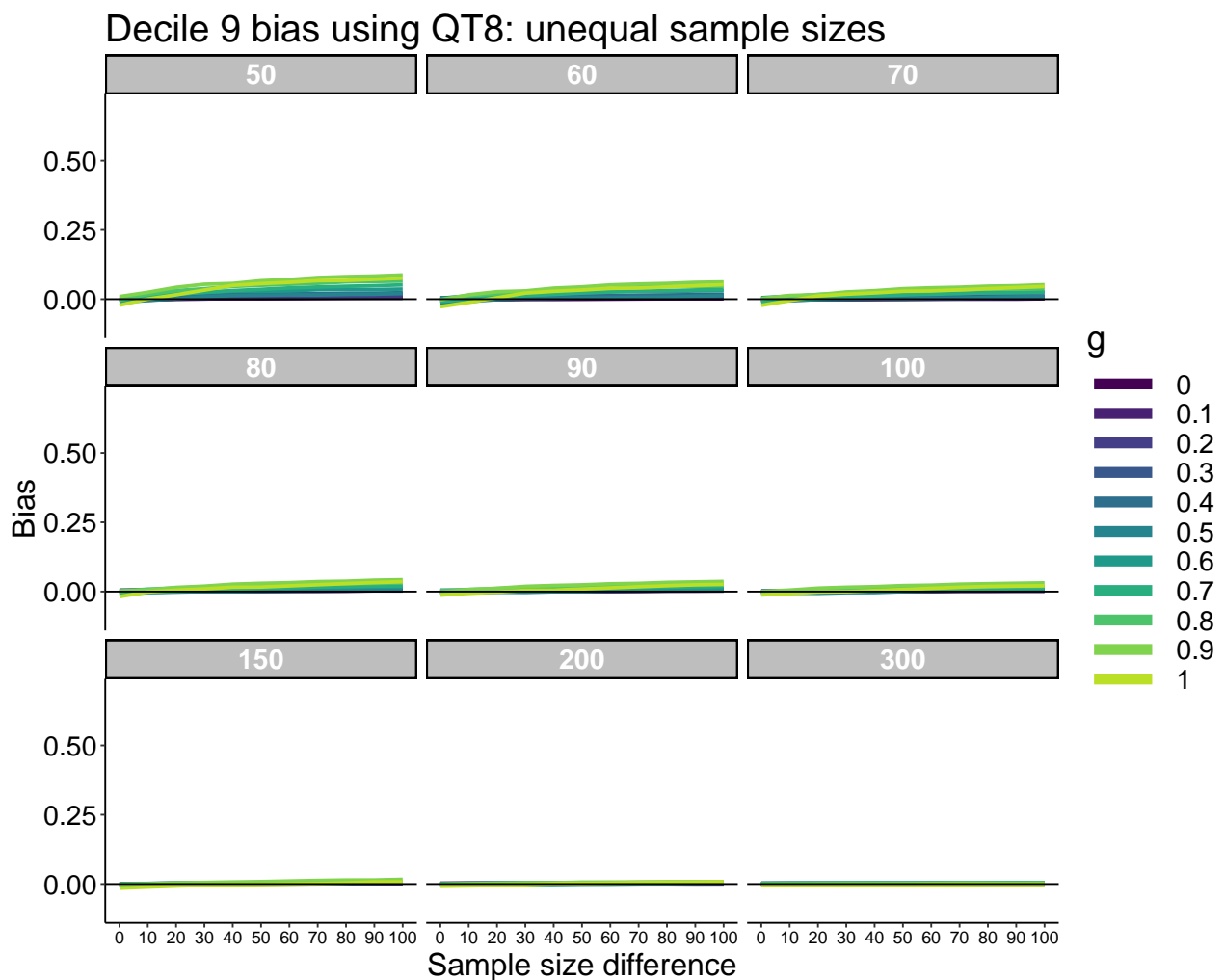
```
df <- tibble(Bias = as.vector(bias.qt8),
             g = factor(rep(gvec, length(nvec2)*length(nvec1))),
             Size = rep(rep(nvec2, each=length(gvec)), length(nvec1)),
             q = factor(rep(nvec1, each=length(gvec)*length(nvec2))))

# make plot
p <- ggplot(df) + theme_classic() +
  geom_line(aes(x=Size, y=Bias, colour = g), size = 1) +
  geom_abline(intercept=0, slope=0, colour="black") +
  scale_colour_viridis_d(end=0.9) +
  scale_x_continuous(breaks=nvec2) +
  # scale_y_continuous(breaks=c(-5, seq(0, 50, 10))) +
  coord_cartesian(ylim=c(-0.1, 0.7)) +
  theme(plot.title = element_text(size=22),
        axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 10, 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.position = c(0.55,0.85),
legend.direction = "vertical",
legend.text=element_text(size=16),
legend.title=element_text(size=20),
strip.background=element_rect(fill="grey", colour="black"),
strip.text=element_text(size=16, colour="white", face="bold")) +
labs(x = "Sample size difference", y = "Bias") +
# make thicker legend lines
guides(colour = guide_legend(override.aes = list(size=3))) +
facet_wrap( ~ q, ncol = 3) +
ggtitle("Decile 9 bias using QT8: unequal sample sizes")

```



Simulation with same number of trials in each group using exG dist

10,000 random samples for each of the 12 distributions.

Define ex-Gaussian parameters

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

Simulation

```
nvec <- c(seq(30,100,10), 150, 200, 300) # vector of sample sizes to test
maxnt <- max(nvec)
nsim <- 10000 # simulation samples
qseq <- seq(0.1,0.9,0.1)

# declare matrices of results - save all iterations
sim.sfhd.bias <- array(NA, dim = c(nsim, nP, length(nvec), 9))
sim.sfqt8.bias <- array(NA, dim = c(nsim, nP, length(nvec), 9))

set.seed(21)

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

  beep(2)
  # generate all data at once
  print(paste0("SF bias - exG: 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*nsim, mu = mu, sigma = sigma, tau = tau),
    dim = c(maxnt, nsim))
  mc.data2 <- array(rexgauss(maxnt*nsim, mu = mu, sigma = sigma, tau = tau),
    dim = c(maxnt, nsim))

  for(N in 1:length(nvec)){ # sample sizes

    for(S in 1:nsim){

      sim.sfhd.bias[S,P,N,] <- hdseq(mc.data1[1:nvec[N],S]) - hdseq(mc.data2[1:nvec[N],S])
      sim.sfqt8.bias[S,P,N,] <- quantile(mc.data1[1:nvec[N],S], probs = qseq, type = 8) - quantile(mc.d

    }
  }
}
save(
  sim.sfhd.bias,
  sim.sfqt8.bias,
  nP,
  nvec,
  nsim,
  file=paste0('./data/sim_sf_bias1_exg.RData'))
beep(8)
```

Compute bias

```
load('./data/sim_sf_bias1_exg.RData')
bias.hd <- apply(sim.sfhd.bias[, , 1:length(nvec)], , c(2,3,4), mean, na.rm = TRUE)
bias.hd.md <- apply(sim.sfhd.bias[, , 1:length(nvec)], , c(2,3,4), median, na.rm = TRUE)
bias.qt8 <- apply(sim.sfqt8.bias[, , 1:length(nvec)], , c(2,3,4), mean, na.rm = TRUE)
bias.qt8.md <- apply(sim.sfqt8.bias[, , 1:length(nvec)], , c(2,3,4), median, na.rm = TRUE)
```

Make function

```
make_bias_figure_exg <- function(data, pop.m, pop.md, nvec, nP){
  df <- tibble(Bias = as.vector(data),
               Skewness = factor(rep(round(pop.m - pop.md), length(nvec)*9)),
               Size = rep(rep(nvec, each=nP), 9),
               q = factor(rep(seq(1,9), each=nP*length(nvec))))

  # make plot
  p <- ggplot(df) + theme_classic() +
    geom_line(aes(x=Size, y=Bias, colour = Skewness), size = 1) +
    geom_abline(intercept=0, slope=0, colour="black") +
    scale_colour_viridis_d() +
    scale_x_continuous(breaks=nvec) +
    scale_y_continuous(breaks=c(-5, seq(0,50,10))) +
    coord_cartesian(ylim=c(-5,50)) +
    theme(plot.title = element_text(size=22),
          axis.title.x = element_text(size = 18),
          axis.text.x = element_text(size = 10, 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.position = c(0.55, 0.85),
          legend.direction = "vertical",
          legend.text=element_text(size=16),
          legend.title=element_text(size=18),
          strip.background=element_rect(fill="grey", colour="black"),
          strip.text=element_text(size=16, colour="white", face="bold")) +
    labs(x = "Sample size", y = "Bias") +
    # make thicker legend lines
    guides(colour = guide_legend(override.aes = list(size=3))) +
    facet_wrap(~ q, ncol = 3)

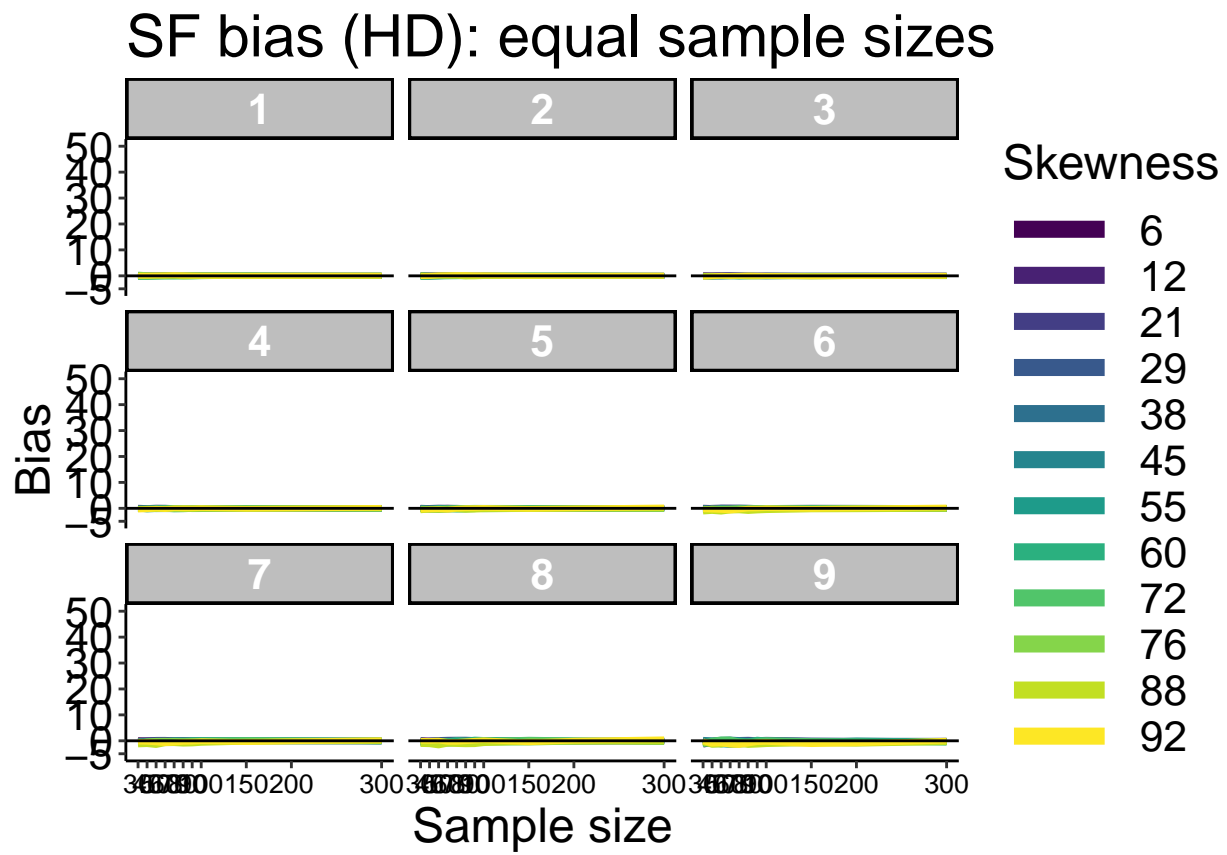
  p
}
```

Illustrate bias results

HD results

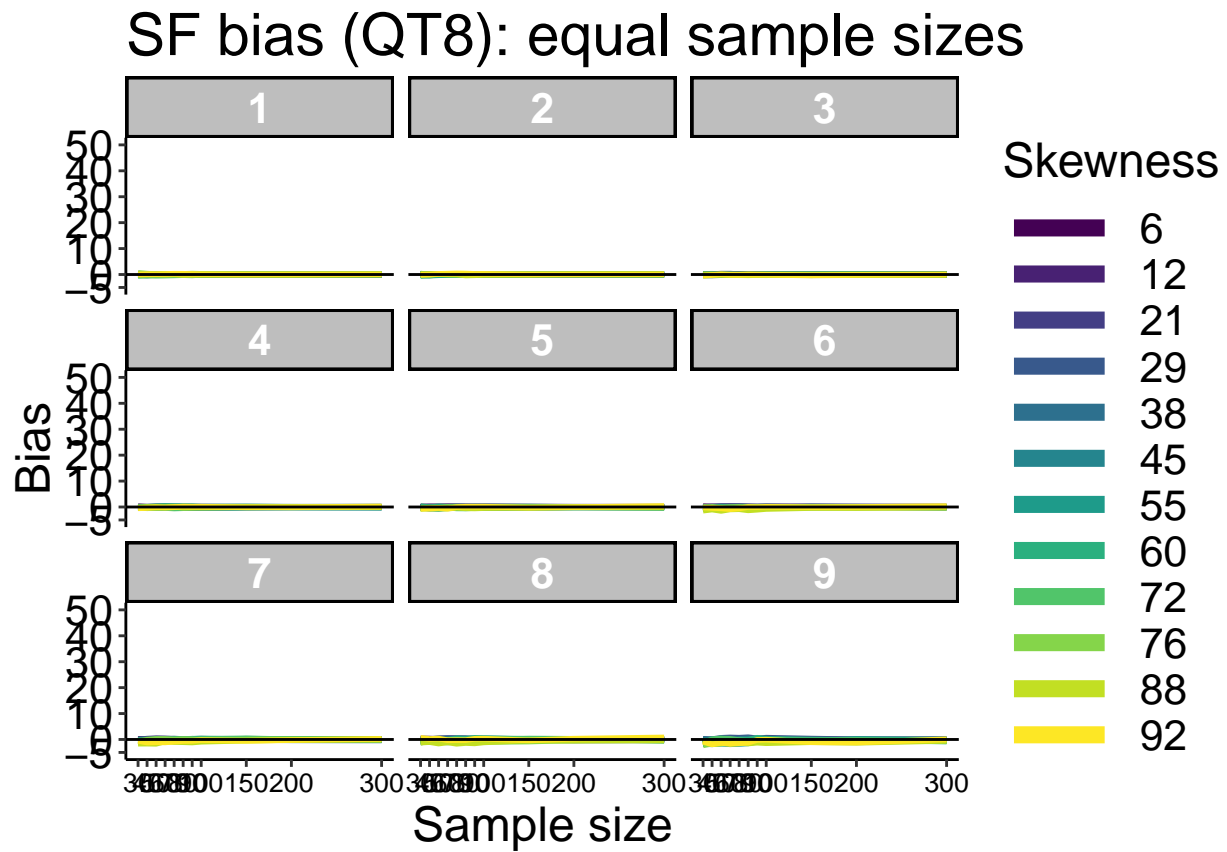
```
p <- make_bias_figure_exg(bias.hd, pop.m, pop.md, nvec, nP)
p <- p + ggtitle("SF bias (HD): equal sample sizes")
```

p



QT8 results

```
p <- make_bias_figure_exg(bias.qt8, pop.m, pop.md, nvec, nP)
p <- p + ggtitle("SF bias (QT8): equal sample sizes")
p
```

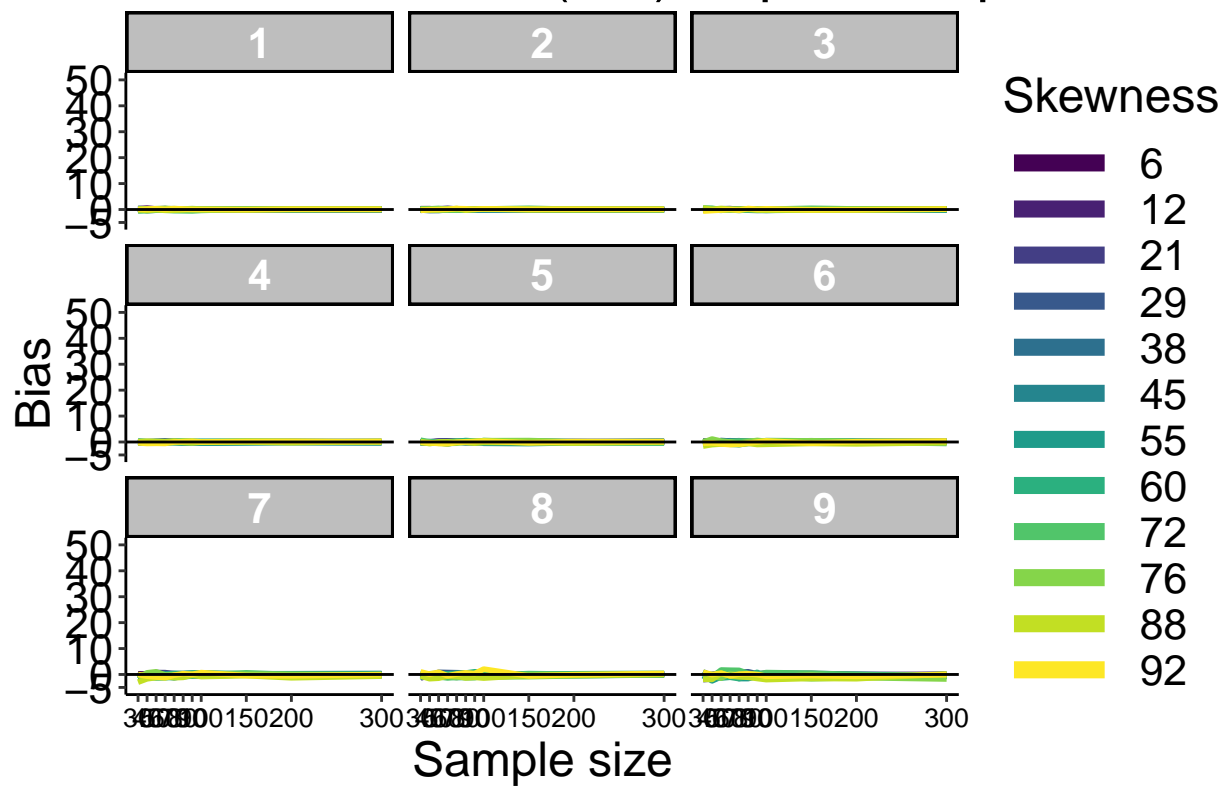


Illustrate bias results: median bias

HD results

```
p <- make_bias_figure_exg(bias.hd.md, pop.m, pop.md, nvec, nP)
p <- p + ggtitle("SF median bias (HD): equal sample sizes")
p
```

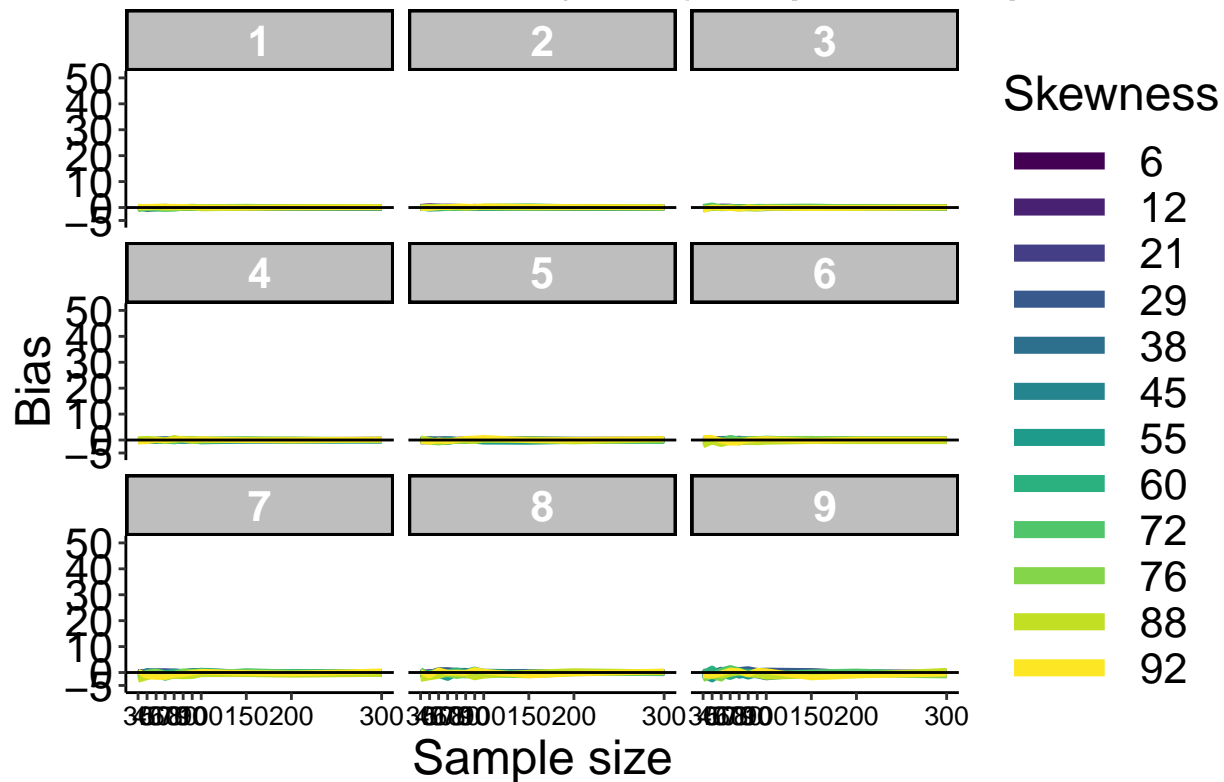
SF median bias (HD): equal sample sizes



QT8 results

```
p <- make_bias_figure_exg(bias.qt8.md, pop.m, pop.md, nvec, nP)
p <- p + ggtitle("SF median bias (QT8): equal sample sizes")
p
```

SF median bias (QT8): equal sample sizes



Illustrate sampling distributions

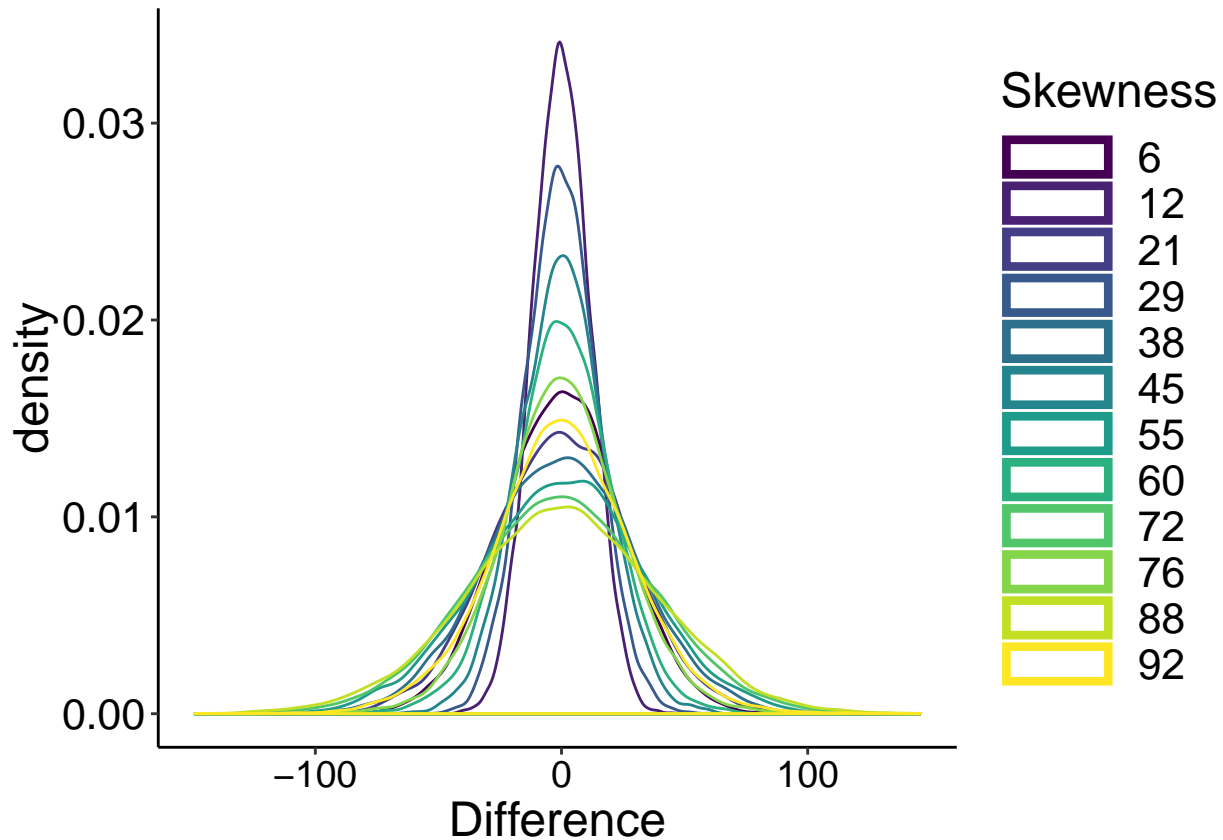
Sample size = 30, $q = 1$

```
# fig.width = 10, fig.height = 6
df <- tibble(Difference = as.vector(sim.sfqt8.bias[,1,1]),
             Skewness = factor(rep(round(pop.m-pop.md), each = nsim))
)

# make plot
p <- ggplot(df, aes(x = Difference, colour = Skewness)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d() +
  # scale_x_continuous(breaks=nvec) +
  # scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
  # coord_cartesian(xlim=c(-2.5,2.5)) +
  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 = "right", #c(0.85,0.65),
        legend.text=element_text(size=16),
        legend.title=element_text(size=18)) +
```



```
# labs(x = "Sample size", y = "Bias in ms") +
# make thicker legend lines
guides(colour = guide_legend(override.aes = list(size=3)))
# + ggtitle("Median RT")
p
```



```
# save figure
# ggsave(filename=paste0('figure_miller_bias_md.pdf'),width=10,height=6)
```

Sample size = 30, $q = 5$

```
# fig.width = 10, fig.height = 6
df <- tibble(Difference = as.vector(sim.sfqt8.bias[,1,5]),
             Skewness = factor(rep(round(pop.m-pop.md), each = nsim))
)

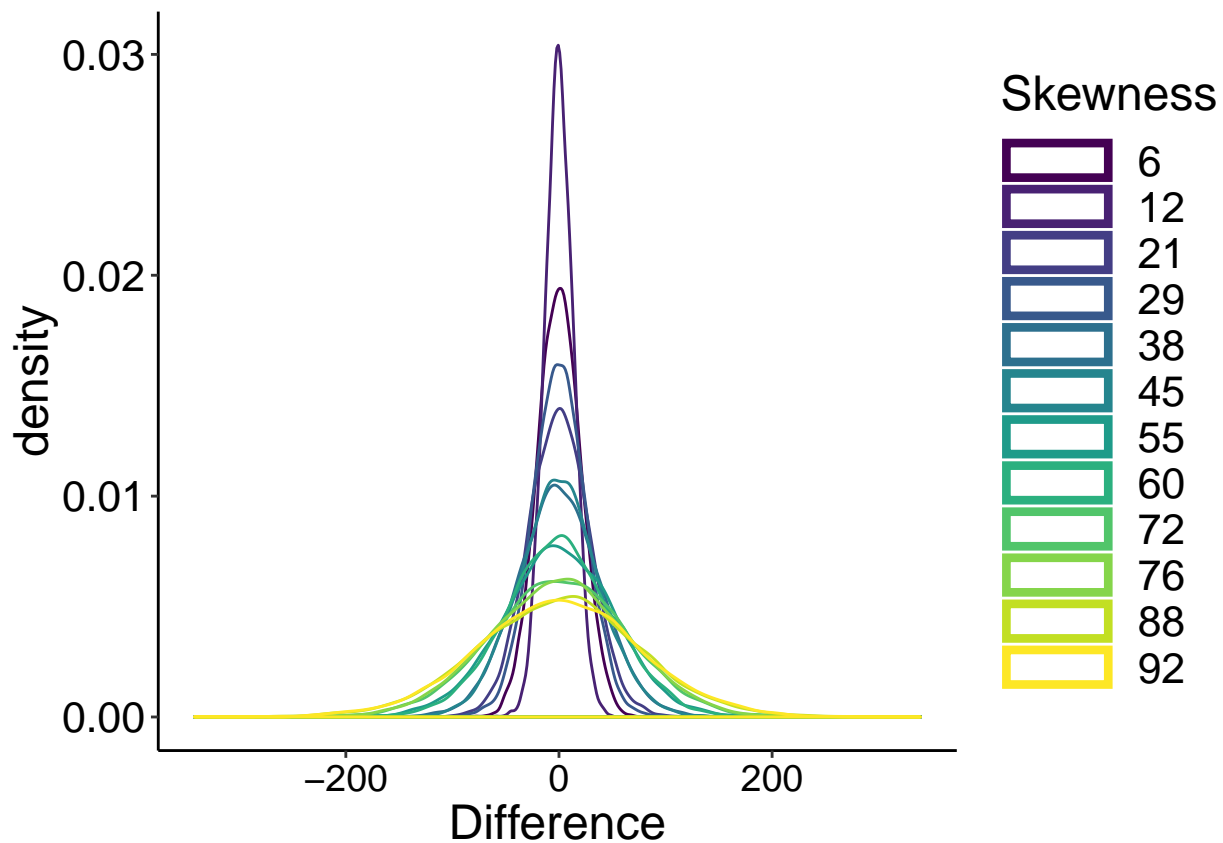
# make plot
p <- ggplot(df, aes(x = Difference, colour = Skewness)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d() +
  # scale_x_continuous(breaks=nvec) +
  # scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
  # coord_cartesian(xlim=c(-2.5,2.5)) +
  theme(plot.title = element_text(size=22),
        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 = "right", #c(0.85,0.65),
legend.text=element_text(size=16),
legend.title=element_text(size=18)) +
# labs(x = "Sample size", y = "Bias in ms") +
# make thicker legend lines
guides(colour = guide_legend(override.aes = list(size=3)))
# + ggtitle("Median RT")

```

p



```

# save figure
# ggsave(filename=paste0('figure_miller_bias_md.pdf'),width=10,height=6)

```

Sample size = 30, q = 9

```

# fig.width = 10, fig.height = 6
df <- tibble(Difference = as.vector(sim.sfqt8.bias[,1,9]),
             Skewness = factor(rep(round(pop.m-pop.md), each = nsim))
             )

# make plot
p <- ggplot(df, aes(x = Difference, colour = Skewness)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d() +

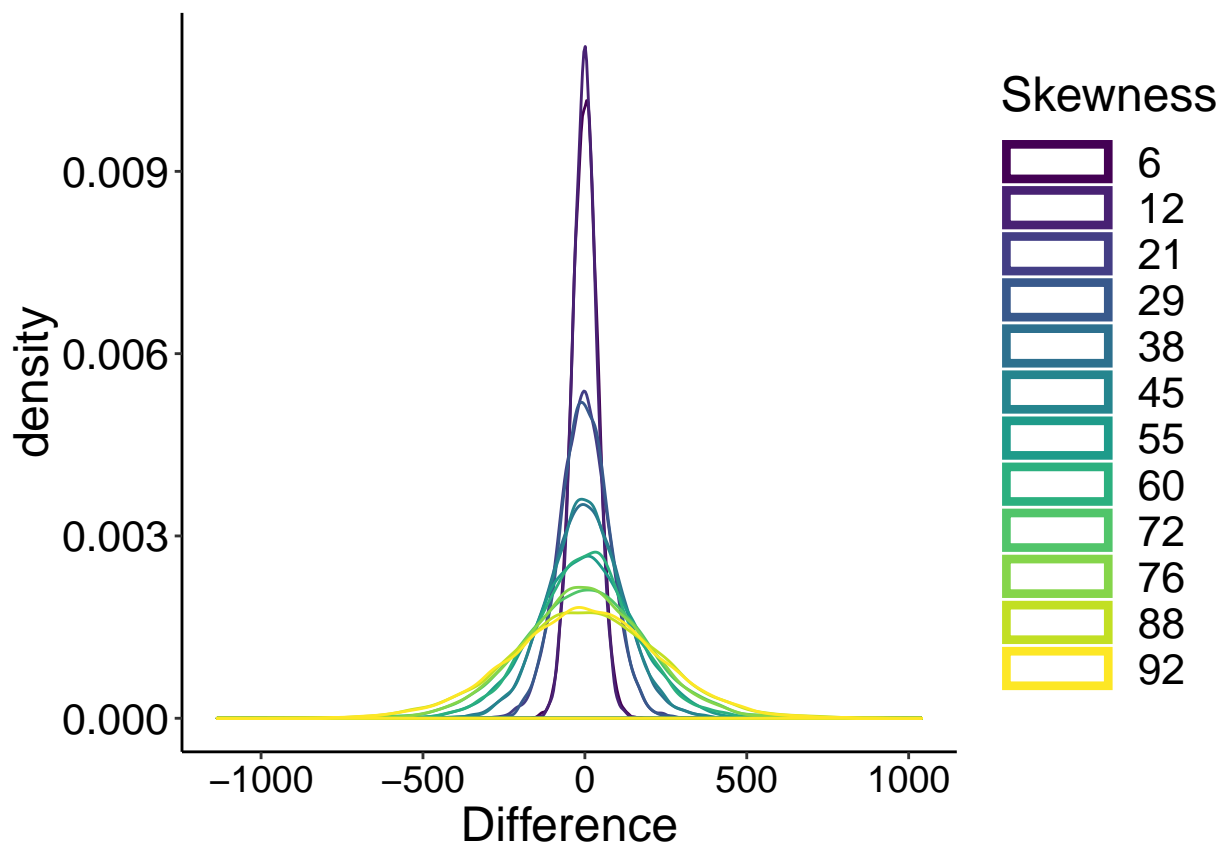
```

```

# scale_x_continuous(breaks=nvec) +
# scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
# coord_cartesian(xlim=c(-2.5,2.5)) +
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 = "right", #c(0.85,0.65),
      legend.text=element_text(size=16),
      legend.title=element_text(size=18)) +
# labs(x = "Sample size", y = "Bias in ms") +
# make thicker legend lines
guides(colour = guide_legend(override.aes = list(size=3)))
# + ggtitle("Median RT")

```

p



```

# save figure
# ggsave(filename=paste0('figure_miller_bias_md.pdf'),width=10,height=6)

```

Sample size = 30

```

# fig.width = 10, fig.height = 6
df <- tibble(Difference = as.vector(sim.sfqt8.bias[,1,]),
             Skewness = factor(rep(rep(round(pop.m-pop.md), each = nsim),9)),

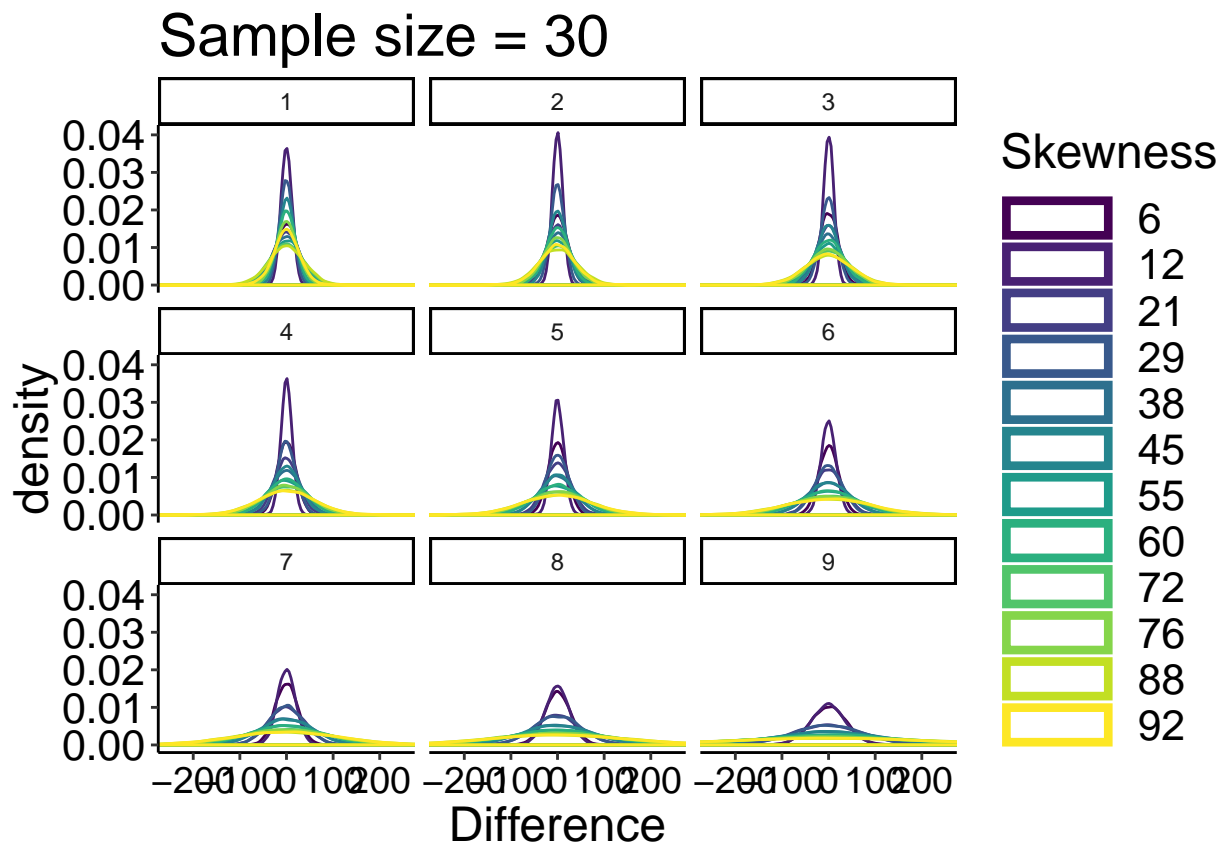
```

```

    q = factor(rep(seq(1,9), each = nsim*nP))
  )

# make plot
p <- ggplot(df, aes(x = Difference, colour = Skewness)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d() +
  # scale_x_continuous(breaks=nvec) +
  # scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
  coord_cartesian(xlim=c(-250,250)) +
  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 = "right", #c(0.85,0.65),
        legend.text=element_text(size=16),
        legend.title=element_text(size=18)) +
  # labs(x = "Sample size", y = "Bias in ms") +
  # make thicker legend lines
  guides(colour = guide_legend(override.aes = list(size=3))) +
  facet_wrap( ~ q, ncol = 3) +
  ggtitle("Sample size = 30")
p

```

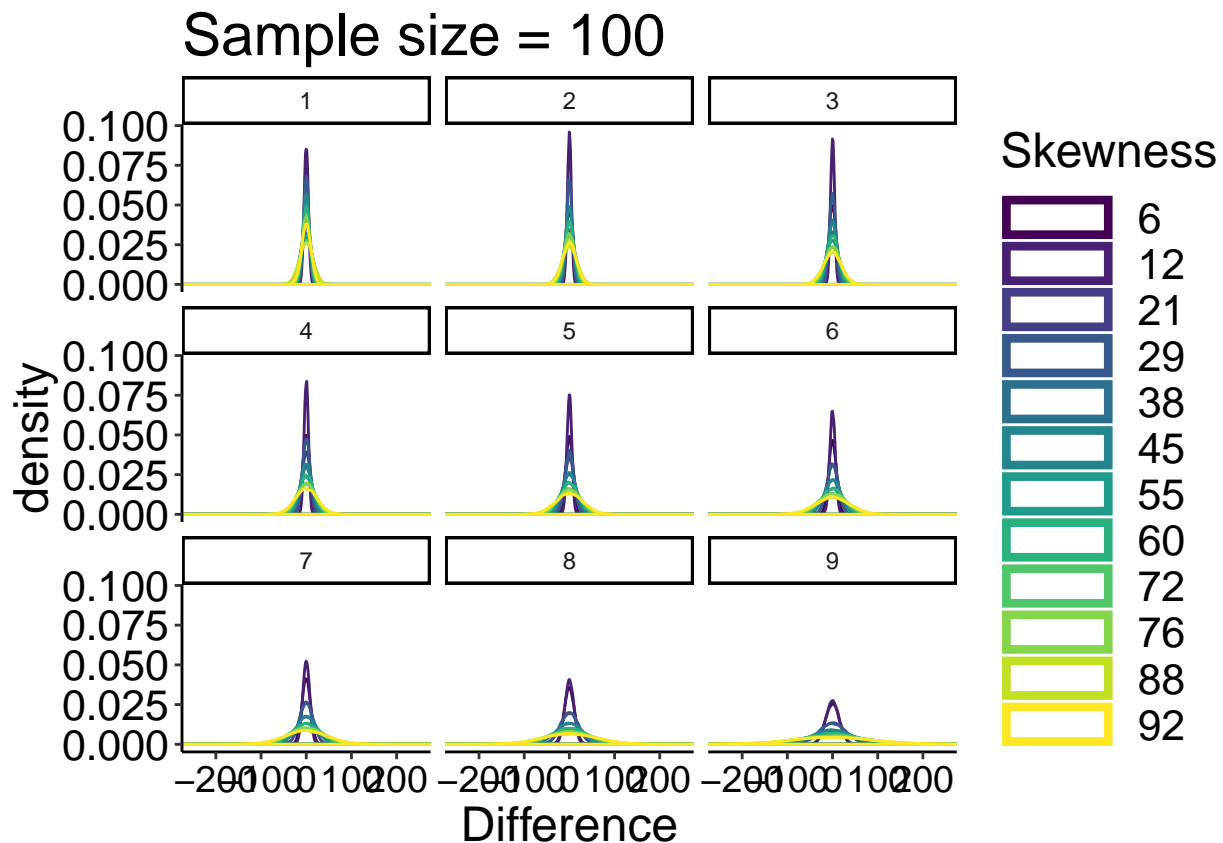


```
# save figure
# ggsave(filename=paste0('figure_miller_bias_md.pdf'),width=10,height=6)
```

Sample size = 100

```
df <- tibble(Difference = as.vector(sim.sfqt8.bias[, ,10,]),
             Skewness = factor(rep(rep(round(pop.m-pop.md), each = nsim),9)),
             q = factor(rep(seq(1,9), each = nsim*nP))
             )

# make plot
p <- ggplot(df, aes(x = Difference, colour = Skewness)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d() +
  # scale_x_continuous(breaks=nvec) +
  # scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
  coord_cartesian(xlim=c(-250,250)) +
  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 = "right", #c(0.85,0.65),
        legend.text=element_text(size=16),
        legend.title=element_text(size=18)) +
  # labs(x = "Sample size", y = "Bias in ms") +
  # make thicker legend lines
  guides(colour = guide_legend(override.aes = list(size=3))) +
  facet_wrap( ~ q, ncol = 3) +
  ggtitle("Sample size = 100")
p
```



```
# save figure
# ggsave(filename=paste0('figure_miller_bias_md.pdf'),width=10,height=6)
```

Simulation with different numbers of trials in each group using exG dist

10,000 random samples for each of the 12 distributions: - group 1: n trials = c(seq(30, 100, 10), 150, 200) - group 2: 200 trials

Simulation

```
nvec1 <- c(seq(30, 100, 10), 150, 200) # vector of sample sizes to test
nvec2 <- rep(200, length(nvec1)) # number of trials in group 2
maxnt1 <- max(nvec1)
maxnt2 <- max(nvec2)
nsim <- 10000 # simulation samples
qseq <- seq(0.1, 0.9, 0.1)

# declare matrices of results - save all iterations
sim.sfhd.bias <- array(NA, dim = c(nsim, nP, length(nvec1), 9))
sim.sfqt8.bias <- array(NA, dim = c(nsim, nP, length(nvec1), 9))

set.seed(21)
```

```

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

  beep(2)
  # generate all data at once
  print(paste0("SF bias - exG: 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*nsim, mu = mu, sigma = sigma, tau = tau),
    dim = c(maxnt1, nsim))
  mc.data2 <- array(rexgauss(maxnt2*nsim, mu = mu, sigma = sigma, tau = tau),
    dim = c(maxnt2, nsim))

  for(N in 1:length(nvec1)){ # sample sizes

    for(S in 1:nsim){

      sim.sfhd.bias[S,P,N,] <- hdseq(mc.data1[1:nvec1[N],S]) - hdseq(mc.data2[1:nvec2[N],S])
      sim.sfqt8.bias[S,P,N,] <- quantile(mc.data1[1:nvec1[N],S], probs = qseq, type = 8) -
    }
  }
}
save(
  sim.sfhd.bias,
  sim.sfqt8.bias,
  nP,
  nvec1,
  nvec2,
  nsim,
  file=paste0('./data/sim_sf_bias2_exg.RData'))
beep(8)

```

Compute bias

```

load('./data/sim_sf_bias2_exg.RData')
bias.hd <- apply(sim.sfhd.bias[, , 1:length(nvec1), ], c(2,3,4), mean, na.rm = TRUE)
bias.hd.md <- apply(sim.sfhd.bias[, , 1:length(nvec1), ], c(2,3,4), median, na.rm = TRUE)
bias.qt8 <- apply(sim.sfqt8.bias[, , 1:length(nvec1), ], c(2,3,4), mean, na.rm = TRUE)
bias.qt8.md <- apply(sim.sfqt8.bias[, , 1:length(nvec1), ], c(2,3,4), median, na.rm = TRUE)

```

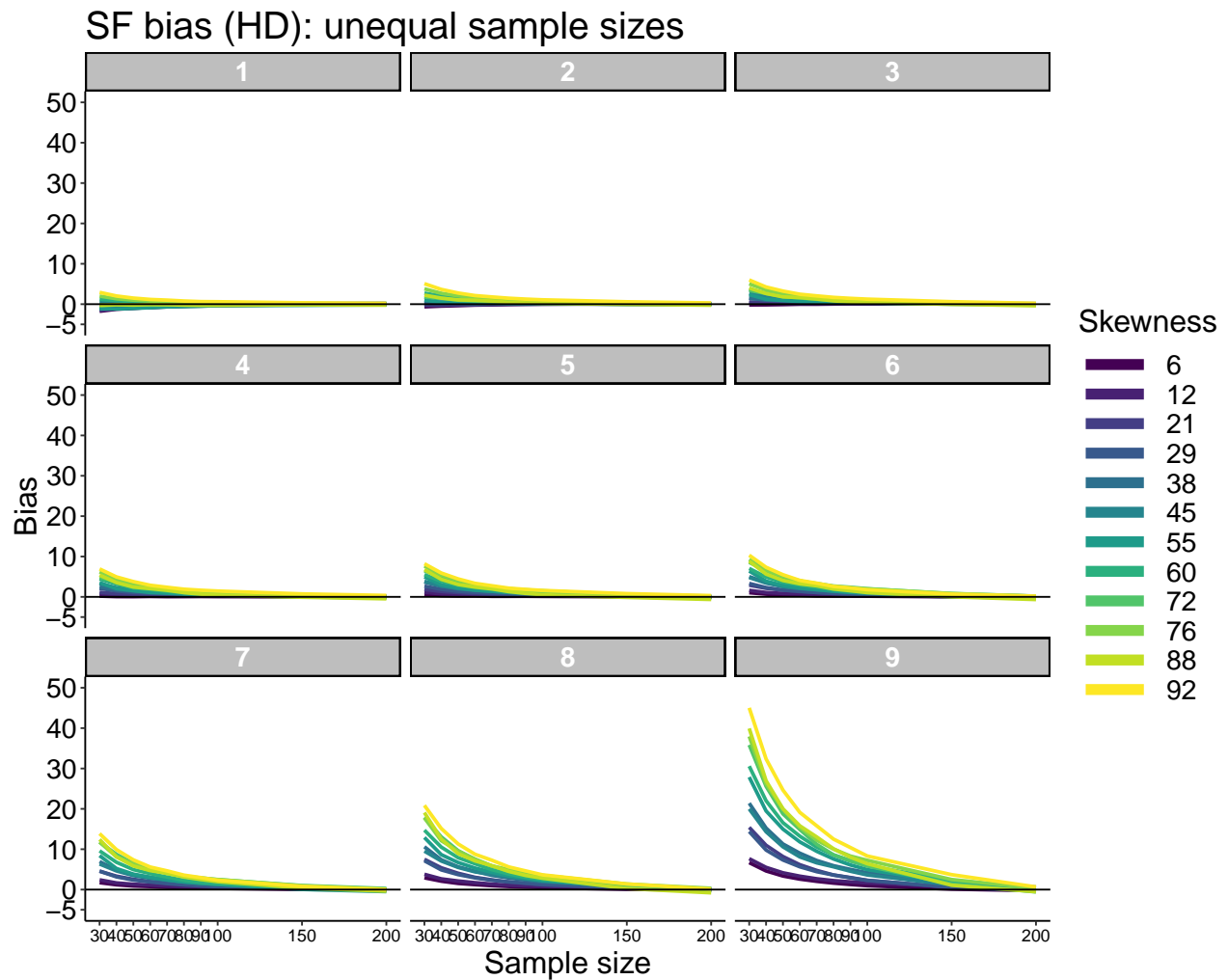
Illustrate bias results

HD results

```

p <- make_bias_figure_exg(bias.hd, pop.m, pop.md, nvec1, nP)
p <- p + ggtitle("SF bias (HD): unequal sample sizes")
p

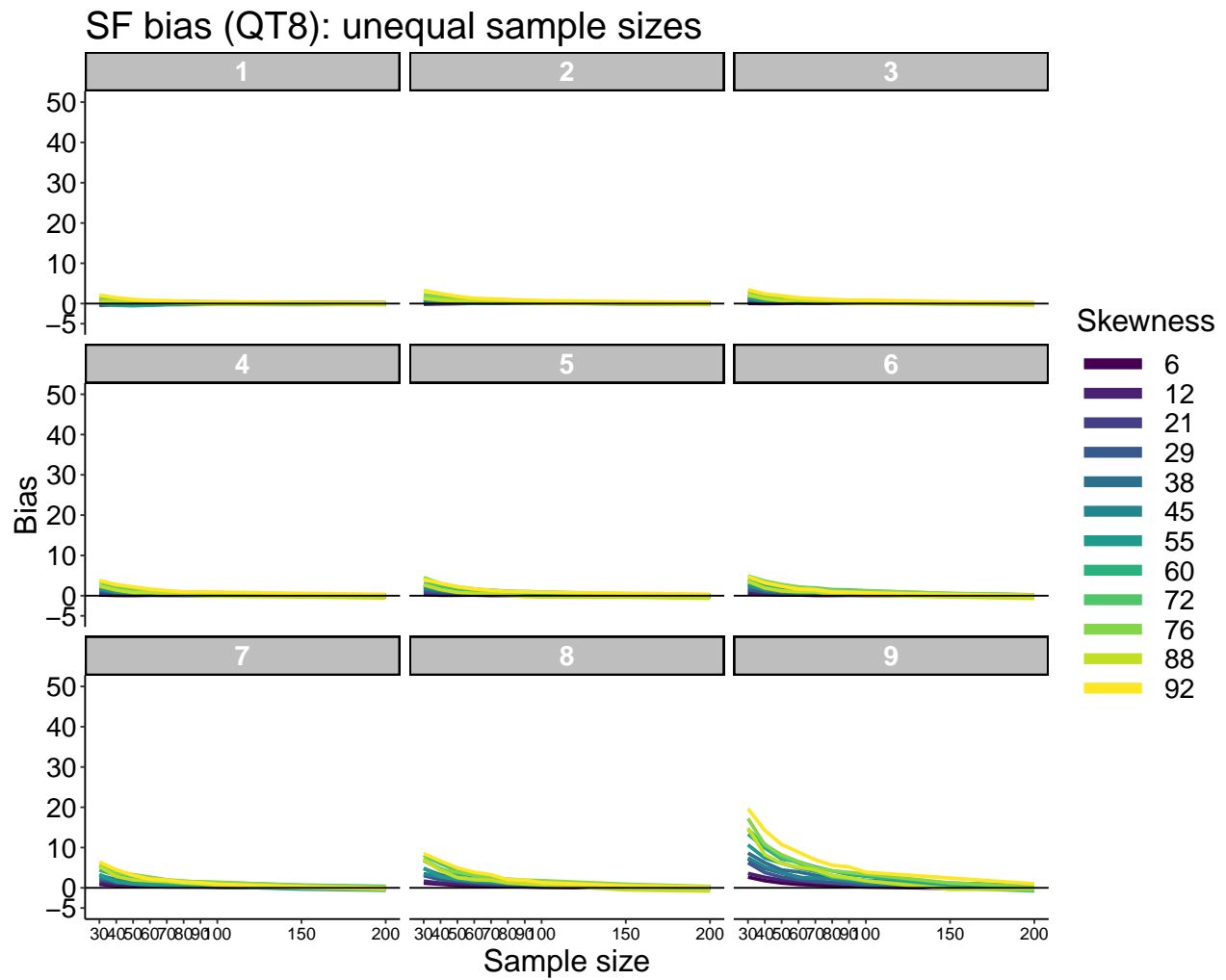
```



Positive bias, as expected: with increased skewness, the smaller number of trials in group 1 leads to over-estimation of deciles, such that the difference between group 1 and group 2, is positive.

QT8 results

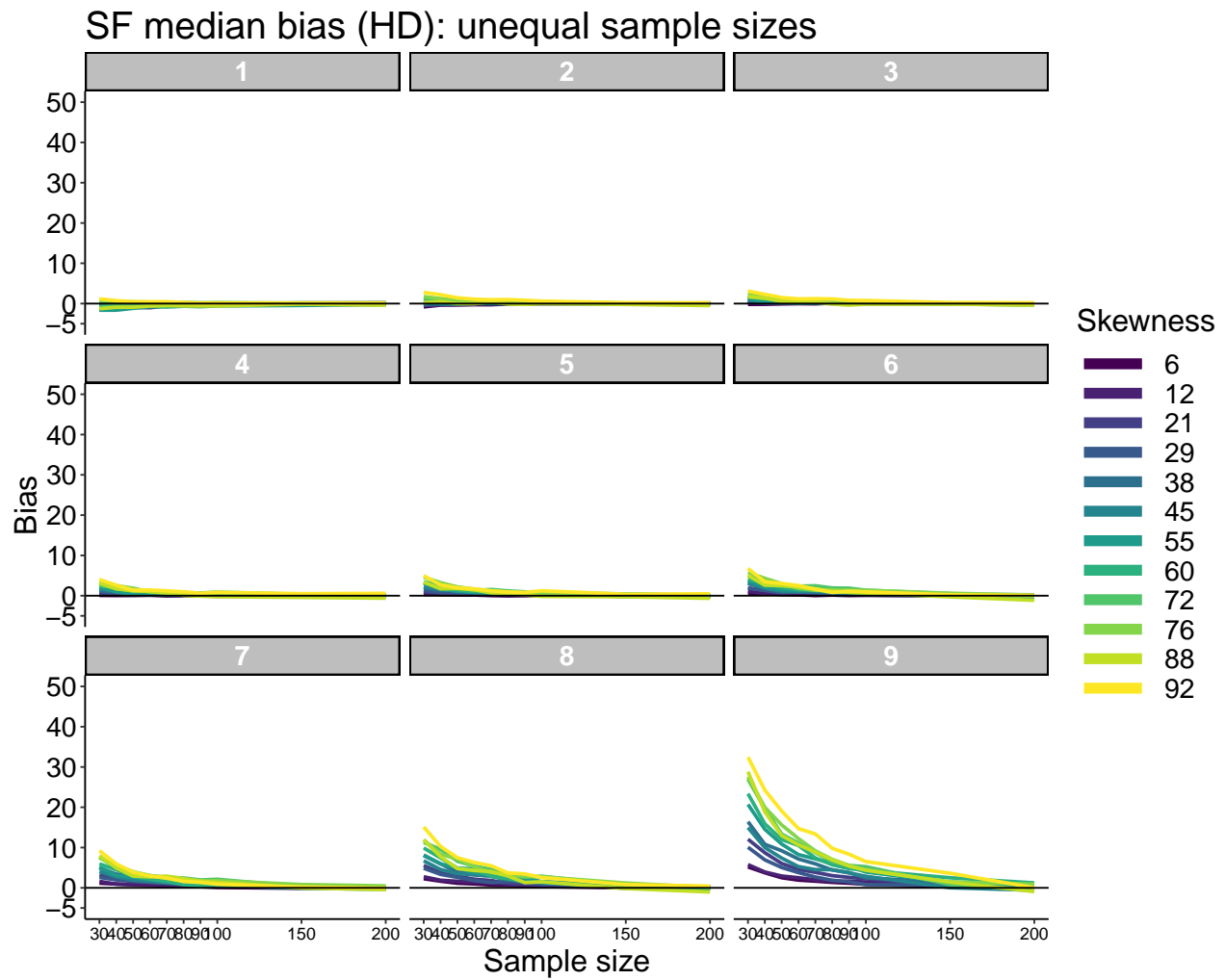
```
p <- make_bias_figure_exg(bias.qt8, pop.m, pop.md, nvec1, nP)
p <- p + ggtitle("SF bias (QT8): unequal sample sizes")
p
```

Illustrate bias results: median bias

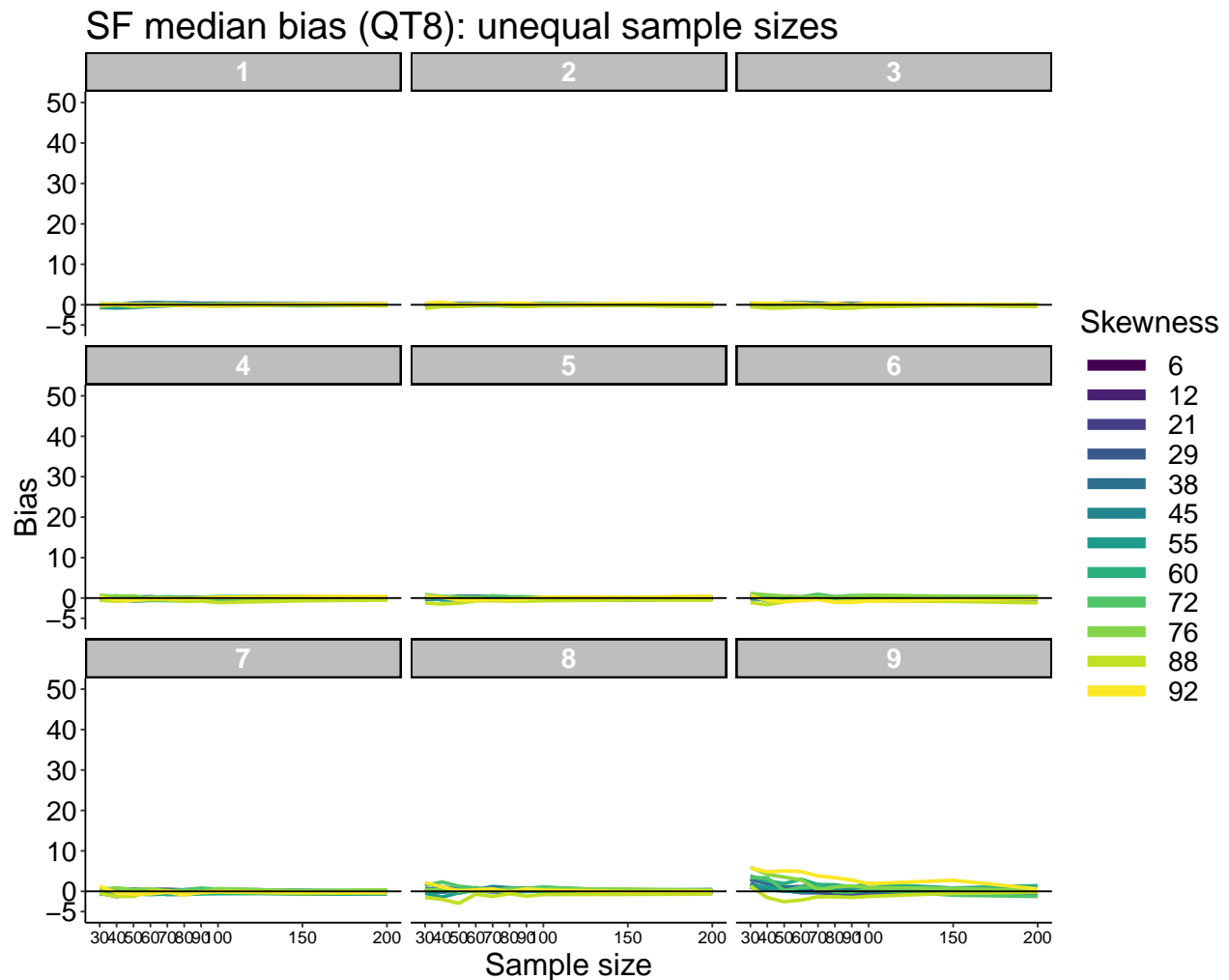
HD results

```
p <- make_bias_figure_exg(bias.hd.md, pop.m, pop.md, nvec1, nP)
p <- p + ggtitle("SF median bias (HD): unequal sample sizes")
p
```



QT8 results

```
p <- make_bias_figure_exg(bias.qt8.md, pop.m, pop.md, nvec1, nP)
p <- p + ggtitle("SF median bias (QT8): unequal sample sizes")
p
```



Illustrate sampling distributions

Sample size = 30, $q = 1$

```
# fig.width = 10, fig.height = 6
df <- tibble(Difference = as.vector(sim.sfqt8.bias[, , 1, 1]),
             Skewness = factor(rep(round(pop.m-pop.md), each = nsim))
            )

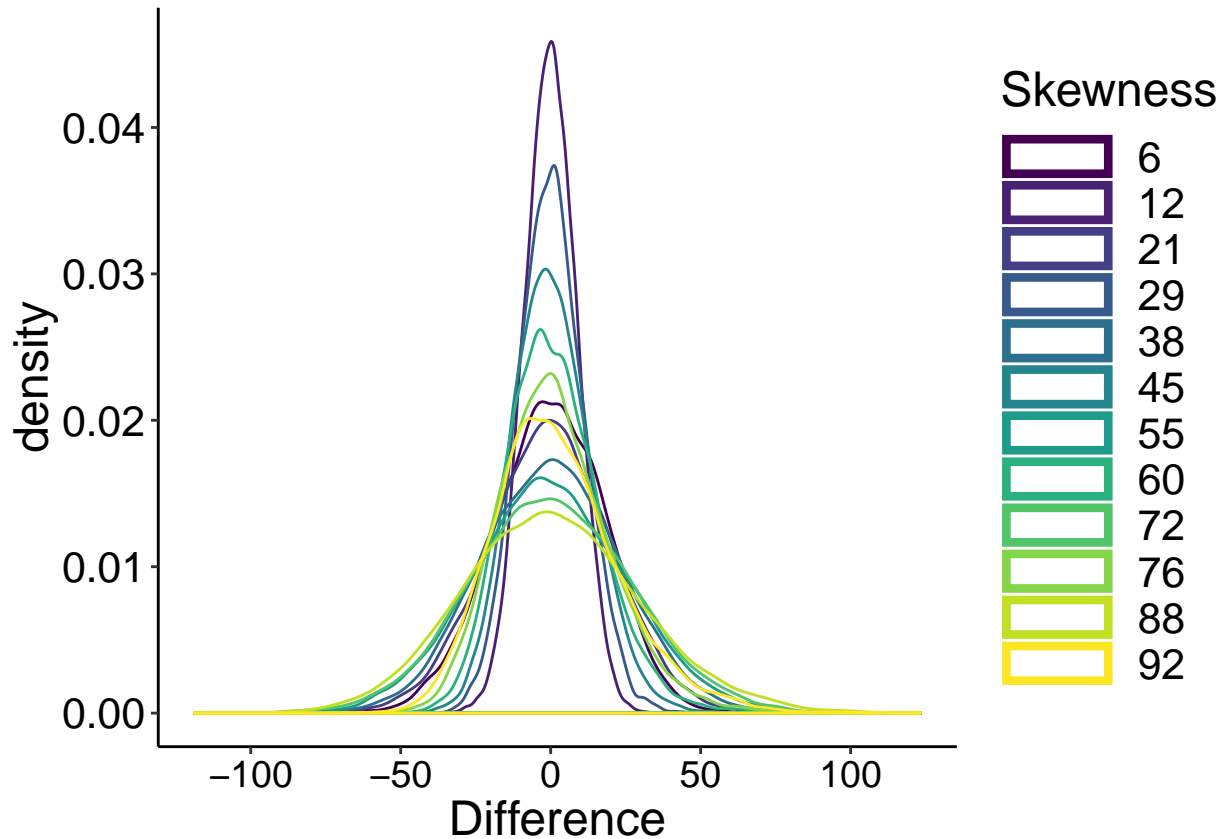
# make plot
p <- ggplot(df, aes(x = Difference, colour = Skewness)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d() +
  # scale_x_continuous(breaks=nvec) +
  # scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
  # coord_cartesian(xlim=c(-2.5,2.5)) +
  theme(plot.title = element_text(size=22),
        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 = "right", #c(0.85,0.65),
legend.text=element_text(size=16),
legend.title=element_text(size=18)) +
# labs(x = "Sample size", y = "Bias in ms") +
# make thicker legend lines
guides(colour = guide_legend(override.aes = list(size=3)))
# + ggtitle("Median RT")

```

p



```

# save figure
# ggsave(filename=paste0('figure_miller_bias_md.pdf'),width=10,height=6)

```

Sample size = 30, q = 5

```

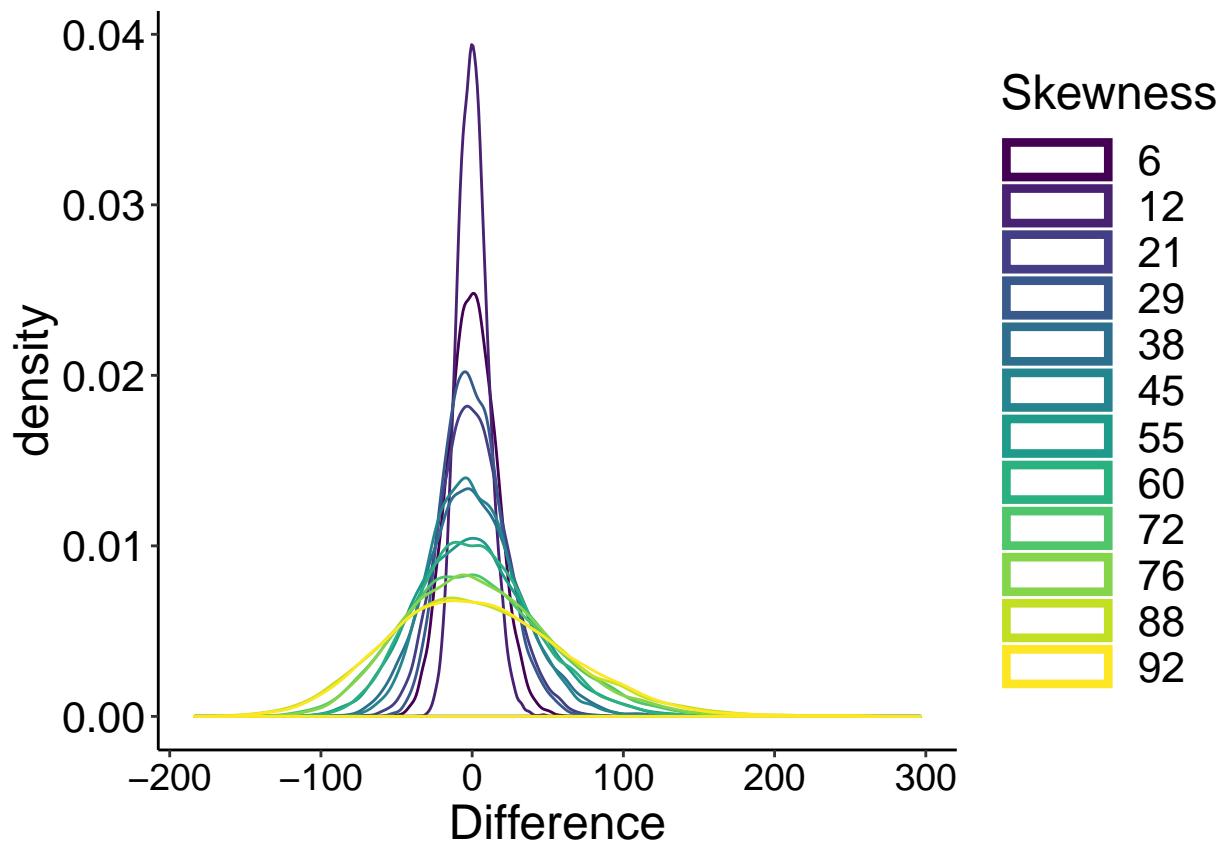
# fig.width = 10, fig.height = 6
df <- tibble(Difference = as.vector(sim.sfqt8.bias[,1,5]),
             Skewness = factor(rep(round(pop.m-pop.md), each = nsim))
             )

# make plot
p <- ggplot(df, aes(x = Difference, colour = Skewness)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d() +
  # scale_x_continuous(breaks=nvec) +

```

```
# scale_y_continuous(breaks=c(-5,seq(0,50,10))) +
# coord_cartesian(xlim=c(-2.5,2.5)) +
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 = "right", #c(0.85,0.65),
      legend.text=element_text(size=16),
      legend.title=element_text(size=18)) +
# labs(x = "Sample size", y = "Bias in ms") +
# make thicker legend lines
guides(colour = guide_legend(override.aes = list(size=3)))
# + ggtitle("Median RT")
```

p

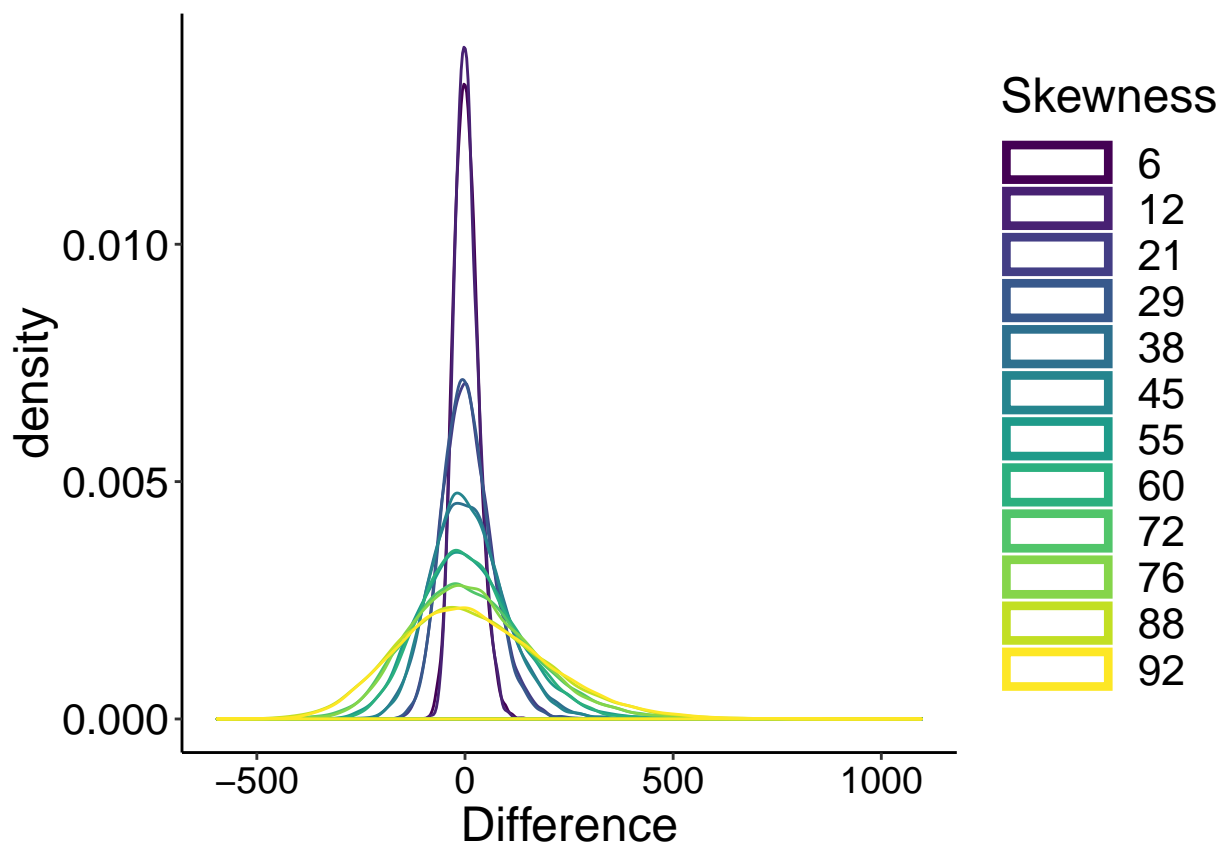


```
# save figure
# ggsave(filename=paste0('figure_miller_bias_md.pdf'),width=10,height=6)
```

Sample size = 30, q = 9

```
df <- tibble(Difference = as.vector(sim.sfqt8.bias[, ,1,9]),
             Skewness = factor(rep(round(pop.m-pop.md), each = nsim))
            )
```

```
# make plot
p <- ggplot(df, aes(x = Difference, colour = Skewness)) + theme_classic() +
  geom_density() +
  scale_colour_viridis_d() +
  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 = "right", #c(0.85,0.65),
        legend.text=element_text(size=16),
        legend.title=element_text(size=18)) +
  guides(colour = guide_legend(override.aes = list(size=3)))
p
```



Sample size = 30

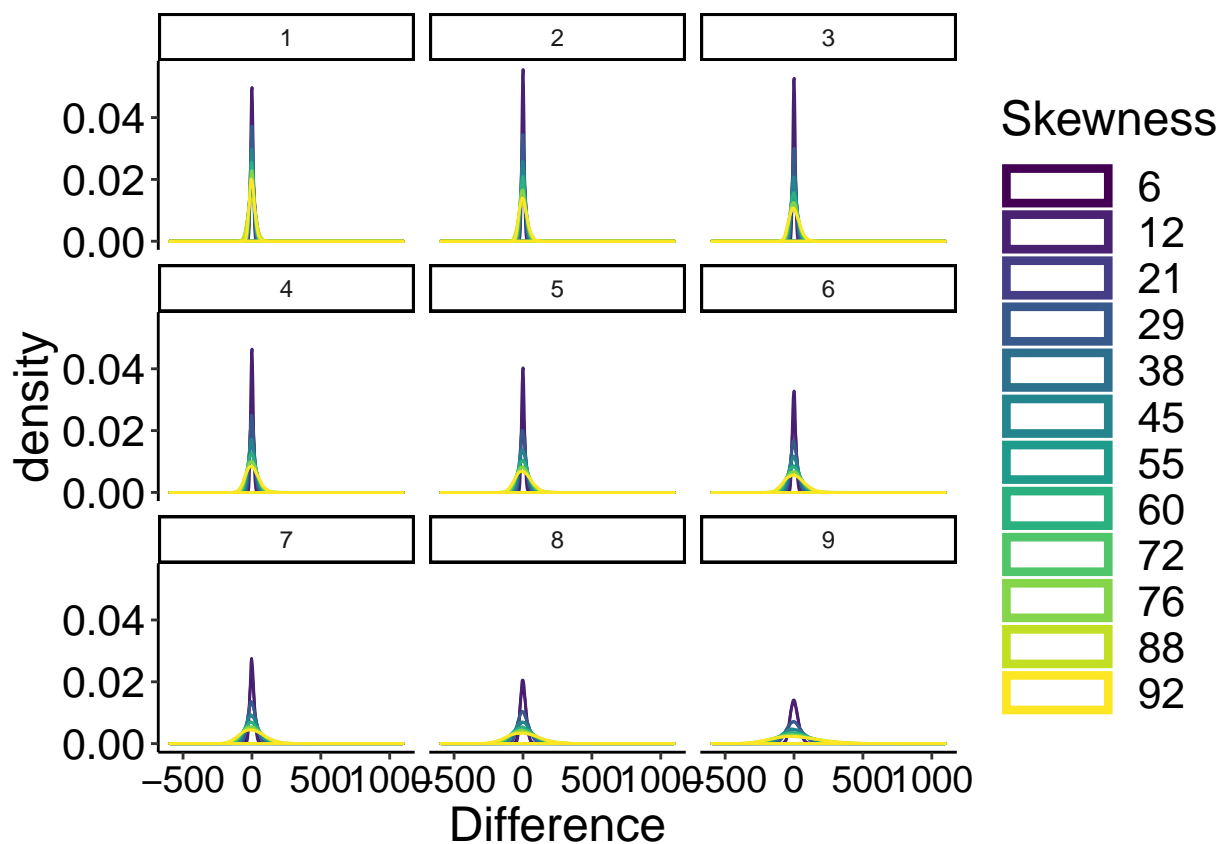
```
df <- tibble(Difference = as.vector(sim.sfqt8.bias[,1,]),
             Skewness = factor(rep(rep(round(pop.m-pop.md), each = nsim),9)),
             q = factor(rep(seq(1,9), each = nsim*nP))
)

# make plot
p <- ggplot(df, aes(x = Difference, colour = Skewness)) + theme_classic() +
```

```

geom_density() +
scale_colour_viridis_d() +
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 = "right", #c(0.85,0.65),
      legend.text=element_text(size=16),
      legend.title=element_text(size=18)) +
guides(colour = guide_legend(override.aes = list(size=3))) +
facet_wrap( ~ q, ncol = 3)
p

```



Sample size = 100

```

df <- tibble(Difference = as.vector(sim.sfqt8.bias[,10,]),
             Skewness = factor(rep(rep(round(pop.m-pop.md), each = nsim),9)),
             q = factor(rep(seq(1,9), each = nsim*nP))
)

# make plot
p <- ggplot(df, aes(x = Difference, colour = Skewness)) + theme_classic() +
  geom_density() +

```

```

scale_colour_viridis_d() +
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 = "right", #c(0.85,0.65),
      legend.text=element_text(size=16),
      legend.title=element_text(size=18)) +
guides(colour = guide_legend(override.aes = list(size=3))) +
facet_wrap( ~ q, ncol = 3)

```

p

