

One-sample / paired t-test: false positive & power analysis

Guillaume A. Rousselet & Rand R. Wilcox

2019-01-16

Contents

g & h distributions	1
Examples in which g varies from 0 to 1.	2
Examples in which we vary h from 0 to 0.5, with g = 0.	4
Examples in which we vary h from 0 to 0.5, with g = 0.3.	6
Simulation	8
Results	12
G distributions: Type I error rate	12
G distributions: Power	15
H distributions with g=0: Type I error rate	18
H distributions with g=0: Power	21
H distributions with g=0.3: Type I error rate	24
H distributions with g=0.3: Power	28
Make summary figures	31

```
# dependencies
library(ggplot2)
library(cowplot)
library(tibble)
library(tidyr)
library(beep)
library(knitr)
source("../functions/gh.txt")
source("../functions/skew.txt")
source("../functions/akerd.txt")
source("../functions/tests.txt")
```

To run a power analysis, you need to consider these choices: - the shape of the distribution to sample from; - the sample size; - the effect size; - the statistical tests.

Here we run simulations with a selection of parameters, to illustrate that when sampling from a skewed distribution with a relatively large sample size, the choice of statistical test can have large effects on power. In particular, a t-test on the mean can have very low power.

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 g varies from 0 to 1.

A $g \& h$ distribution with $g=1$ and $h=0$ corresponds to a lognormal distribution.

```
ng <- seq(0,1,0.1)
x <- seq(-4, 15, 0.05)
res.g <- array(0, dim = c(length(x), length(ng)))
mean.g <- vector(mode = "numeric", length = length(ng))
md.g <- vector(mode = "numeric", length = length(ng))
skew.g <- vector(mode = "numeric", length = length(ng))
kurt.g <- vector(mode = "numeric", length = length(ng))
for(G in 1:length(ng)){
  set.seed(7)
  samp <- ghdist(10000, g = ng[G], h = 0)
  res.g[,G] <- akerd(samp, pts = x, pyhat = TRUE, plotit = FALSE)
  mean.g[G] <- mean(samp)
  md.g[G] <- median(samp)
  skew.g[G] <- skew(samp)
  kurt.g[G] <- kurt(samp)
}
save(res.g,
      mean.g,
      md.g,
      skew.g,
      kurt.g,
      ng,
      x,
      file = "./data/ex_g1_h0.RData")
```

Table of summary statistics

```
load("./data/ex_g1_h0.RData")
res <- tibble(mean = mean.g,
              median = md.g,
              skewness = skew.g,
              kurtosis = kurt.g)
kable(res, digits = c(2, 2, 2, 1))
```

mean	median	skewness	kurtosis
0.00	0	0.01	3.0
0.05	0	0.31	3.2
0.10	0	0.63	3.7
0.16	0	0.96	4.6
0.21	0	1.32	6.0
0.27	0	1.72	8.1
0.34	0	2.18	11.3
0.41	0	2.71	16.0
0.48	0	3.32	23.0
0.57	0	4.05	33.1
0.66	0	4.91	47.7

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

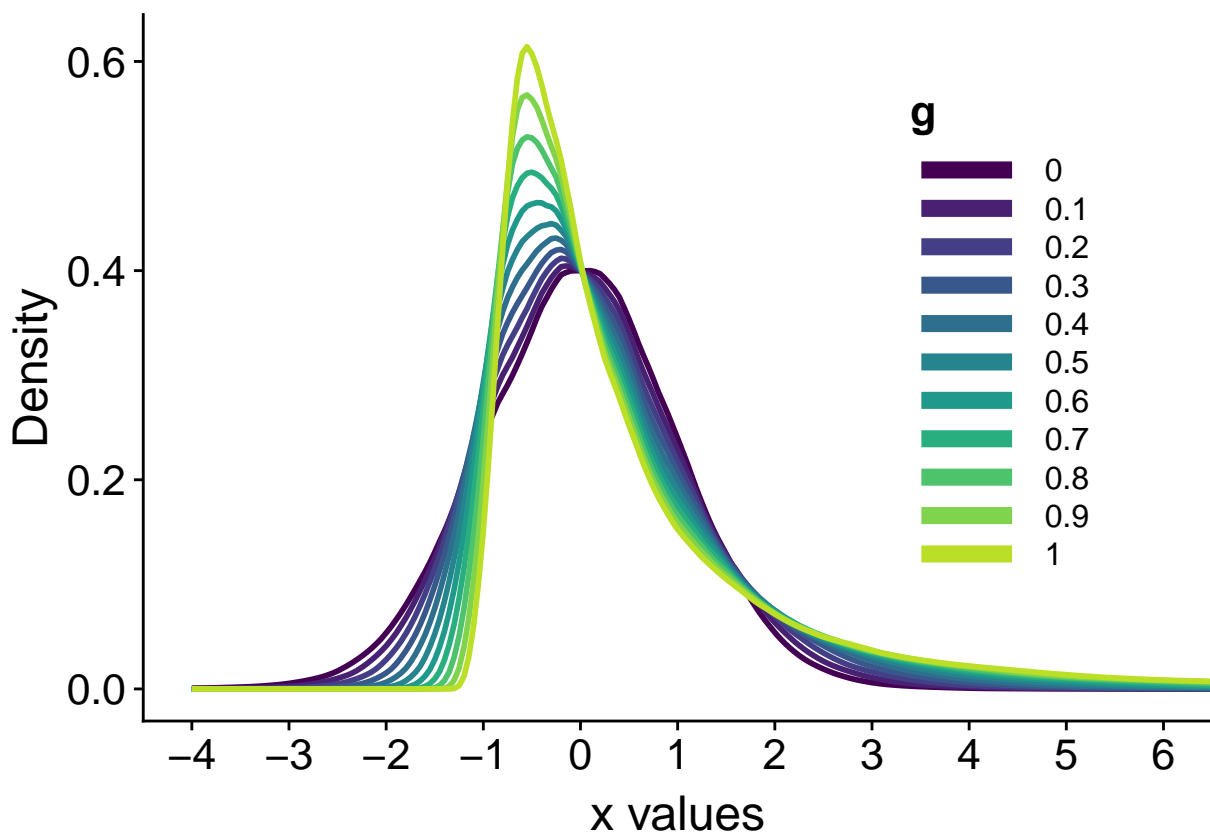
```

# 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=1) +
  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),
        legend.key.width = unit(1.5, "cm"),
        legend.title = element_text(size=16, face="bold"),
        legend.position = c(.7, .55)) +
  coord_cartesian(xlim = c(-4, 6)) +
  scale_y_continuous(breaks = seq(0, 1, 0.2)) +
  scale_x_continuous(breaks = seq(-4, 15, 1)) +
  labs(x = "x values", y = "Density") +
  guides(colour = guide_legend(override.aes = list(size=3)))
p

```



```
p.g <- p
```

Examples in which we vary h from 0 to 0.5, with $g = 0$.

```
nh <- seq(0,0.5,0.1)
x <- seq(-6, 6, 0.05)
res.h0 <- array(0, dim = c(length(x), length(nh)))
mean.h0 <- vector(mode = "numeric", length = length(nh))
md.h0 <- vector(mode = "numeric", length = length(nh))
skew.h0 <- vector(mode = "numeric", length = length(nh))
kurt.h0 <- vector(mode = "numeric", length = length(nh))
for(H in 1:length(nh)){
  set.seed(7)
  samp <- ghdist(10000, g = 0, h = nh[H])
  res.h0[,H] <- akerd(samp, pts = x, pyhat = TRUE, plotit = FALSE)
  mean.h0[H] <- mean(samp)
  md.h0[H] <- median(samp)
  skew.h0[H] <- skew(samp)
  kurt.h0[H] <- kurt(samp)
}
save(res.h0,
      mean.h0,
      md.h0,
      skew.h0,
      kurt.h0,
      nh,
      x,
      file = "./data/ex_g0_h05.RData")
```

Table of summary statistics

```
load("./data/ex_g0_h05.RData")
res <- tibble(h = nh,
              mean = mean.h0,
              median = md.h0,
              skewness = skew.h0,
              kurtosis = kurt.h0)
kable(res, digits = c(2, 2, 2, 1))
```

h	mean	median	skewness	kurtosis
0.0	0	0	0.0	3.01
0.1	0	0	0.0	5.22
0.2	0	0	-0.1	13.68
0.3	0	0	-0.7	48.99
0.4	0	0	-2.5	173.78
0.5	0	0	-6.1	489.48

Estimators for skewness and especially kurtosis, have large standard errors, especially as h increases. Even with $n =$ one million, when $h=.5$ the estimate of kurtosis can change a lot between samples.

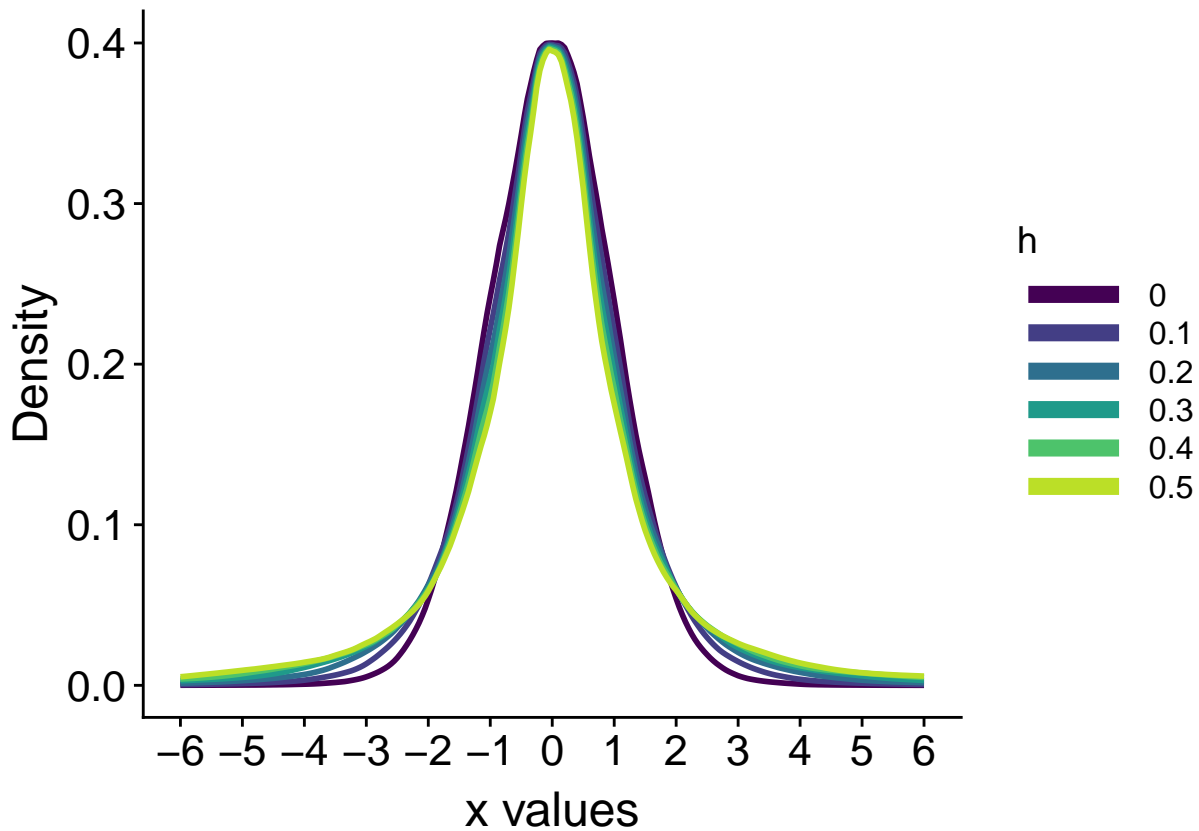
Combine all kernel density functions into one data frame and make summary figure.

```

# make data frame
fm <- array(0, dim = c(length(x), length(nh)+1)) # make full matrix
fm[,1] <- x
fm[,2:(length(nh)+1)] <- res.h0
colnames(fm) <- c("x", paste(seq(0, 0.5, 0.1)))
df <- as_tibble(fm)
df <- tidyr::gather(df, h, Density, 2:(length(nh)+1))
df[[2]] <- as.factor(df[[2]])

# make plot
p <- ggplot(df, aes(x, Density, group = h)) +
  geom_line(aes(colour = h), size=1) +
  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),
        legend.key.width = unit(1.5, "cm")) +
  scale_y_continuous(breaks = seq(0, 0.5, 0.1)) +
  scale_x_continuous(breaks = seq(-6, 6, 1)) +
  coord_cartesian(xlim = c(-6, 6)) +
  labs(x = "x values", y = "Density") +
  guides(colour = guide_legend(override.aes = list(size=3)))
p

```



```

# save figure
# ggsave(filename='figure_h_distributions.pdf',width=7,height=5) #path=pathname

```

Examples in which we vary h from 0 to 0.5, with $g = 0.3$.

The mean is not define for $g=0.3$ and $h=1$ using `ghmean()`. Also, for $g=0.3$, the mean is stable for up to $h=0.6$: repeated calls to `ghdist()` give very similar values which are also very close to the population mean given by `ghmean()`. From 0.7 it fluctuates a lot even for samples with $n = 10,000,000$. For instance:

```
# theoretical value:
ghmean(g=0.3, h=0.7)$mean
```

```
## [1] 0.9848918
```

```
# sample estimates:
mean(ghdist(10000000, g=0.3, h=0.7))
```

```
## [1] 0.9260802
```

```
mean(ghdist(10000000, g=0.3, h=0.7))
```

```
## [1] 0.9790965
```

```
mean(ghdist(10000000, g=0.3, h=0.7))
```

```
## [1] 0.9593666
```

It gets worse with larger h

```
ghmean(g=0.3, h=0.8)$mean
```

```
## [1] 1.880702
```

```
mean(ghdist(10000000, g=0.3, h=0.8))
```

```
## [1] 1.904353
```

```
mean(ghdist(10000000, g=0.3, h=0.8))
```

```
## [1] 1.773731
```

```
mean(ghdist(10000000, g=0.3, h=0.8))
```

```
## [1] 2.243056
```

So we only consider h up to 0.5, which is already an extreme case.

Generate data

```
nh <- seq(0,0.5,0.1)
x <- seq(-6, 6, 0.05)
res.h03 <- array(0, dim = c(length(x), length(nh)))
mean.h03 <- vector(mode = "numeric", length = length(nh))
md.h03 <- vector(mode = "numeric", length = length(nh))
skew.h03 <- vector(mode = "numeric", length = length(nh))
kurt.h03 <- vector(mode = "numeric", length = length(nh))
for(H in 1:length(nh)){
  set.seed(7)
  samp <- ghdist(10000, g = 0.3, h = nh[H])
  res.h03[,H] <- akerd(samp, pts = x, pyhat = TRUE, plotit = FALSE)
  mean.h03[H] <- mean(samp)
  md.h03[H] <- median(samp)
  skew.h03[H] <- skew(samp)
```

```

kurt.h03[H] <- kurt(samp)
}
save(res.h03,
     mean.h03,
     md.h03,
     skew.h03,
     kurt.h03,
     nh,
     x,
     file = "./data/ex_g03_h05.RData")

```

Table of summary statistics

```

load("./data/ex_g03_h05.RData")
res <- tibble(h = nh,
              mean = mean.h03,
              median = md.h03,
              skewness = skew.h03,
              kurtosis = kurt.h03)
kable(res, digits = c(2, 2, 2, 1))

```

	h	mean	median	skewness	kurtosis
	0.0	0.16	0	1.0	4.56
	0.1	0.19	0	1.6	9.75
	0.2	0.22	0	2.8	28.83
	0.3	0.27	0	5.4	97.61
	0.4	0.34	0	10.0	298.50
	0.5	0.44	0	16.6	723.85

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

```

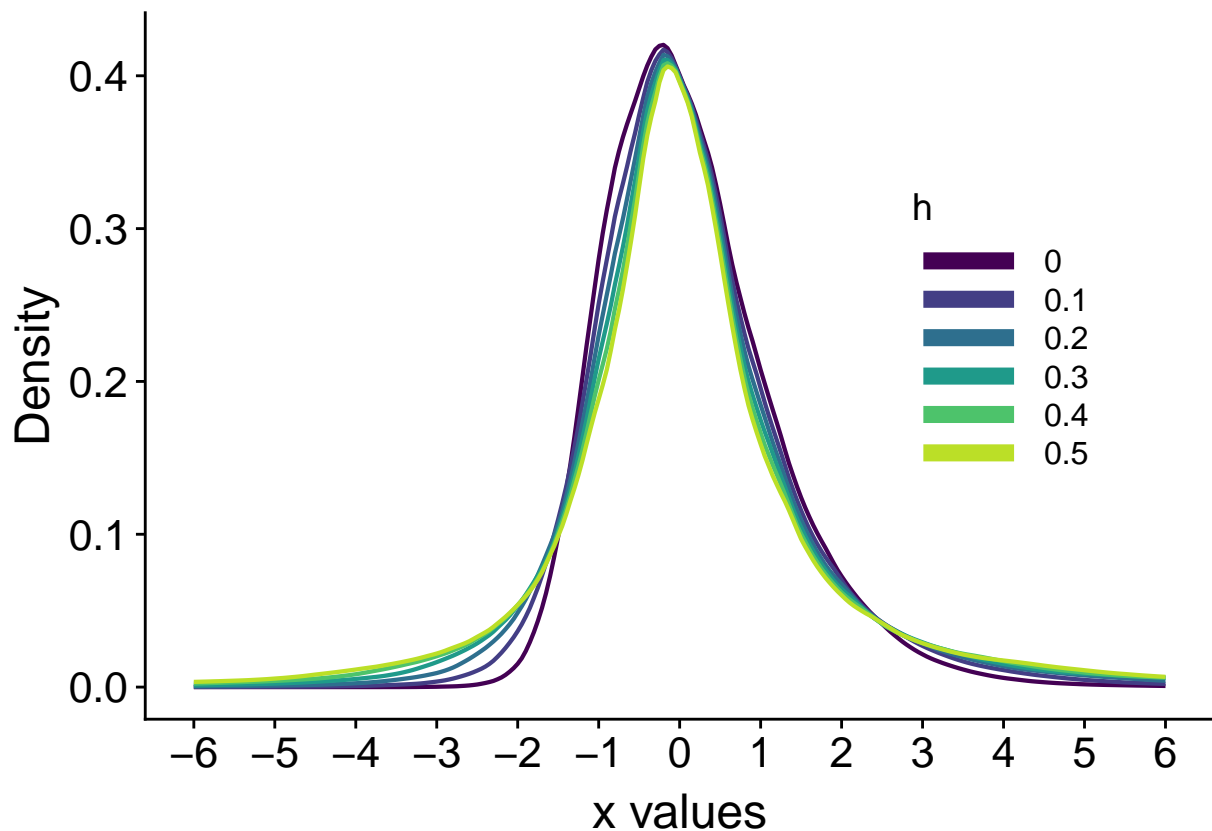
# make data frame
fm <- array(0, dim = c(length(x), length(nh)+1)) # make full matrix
fm[,1] <- x
fm[,2:(length(nh)+1)] <- res.h03
colnames(fm) <- c("x", paste(seq(0, 0.5, 0.1)))
df <- as_tibble(fm)
df <- tidyr::gather(df, h, Density, 2:(length(nh)+1))
df[[2]] <- as.factor(df[[2]])

# make plot
p <- ggplot(df, aes(x, Density, group = h)) +
  geom_line(aes(colour = h), size=0.75) +
  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),
        legend.position = c(.7, .55),
        legend.key.width = unit(1.5, "cm")) +
  scale_y_continuous(breaks = seq(0, 0.5, 0.1)) +
  scale_x_continuous(breaks = seq(-6, 6, 1)) +

```

```
coord_cartesian(xlim = c(-6, 6)) +
labs(x = "x values", y = "Density") +
guides(colour = guide_legend(override.aes = list(size=3)))
```

p



```
p.h <- p
```

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

Simulation

Here we perform a large simulation that includes:

- multiple sample sizes;
- sampling from normal and skewed and kurtotic distributions;
- t-tests using means and 20% trimmed means, tests on medians;
- different effect sizes.

We include an effect size of zero, to assess the type I error rate, or false positives, of the tests.

```
# Define parameters
iter <- 10000 # number of simulations
gvec <- seq(0, 1, 0.1) # g values
hvec <- seq(0, 0.5, 0.1) # h values
nvec <- c(seq(10,100,10),150,200, 300) # vector of sample sizes to test
```



```

# save separately results for 3 estimators (mean, trimmed mean, median), two families of sampling param

ES <- 0.4 # effect size to estimate true positives
gFIX <- 0.3 # constant g for h manipulations

g.m0.res <- array(0, dim = c(length(nvec), length(gvec)))
g.md0.res <- array(0, dim = c(length(nvec), length(gvec)))
g.tm0.res <- array(0, dim = c(length(nvec), length(gvec)))
g.m1.res <- array(0, dim = c(length(nvec), length(gvec)))
g.md1.res <- array(0, dim = c(length(nvec), length(gvec)))
g.tm1.res <- array(0, dim = c(length(nvec), length(gvec)))

h.m0.res <- array(0, dim = c(length(nvec), length(hvec)))
h.md0.res <- array(0, dim = c(length(nvec), length(hvec)))
h.tm0.res <- array(0, dim = c(length(nvec), length(hvec)))
h.m1.res <- array(0, dim = c(length(nvec), length(hvec)))
h.md1.res <- array(0, dim = c(length(nvec), length(hvec)))
h.tm1.res <- array(0, dim = c(length(nvec), length(hvec)))

hg0.m0.res <- array(0, dim = c(length(nvec), length(hvec)))
hg0.md0.res <- array(0, dim = c(length(nvec), length(hvec)))
hg0.tm0.res <- array(0, dim = c(length(nvec), length(hvec)))
hg0.m1.res <- array(0, dim = c(length(nvec), length(hvec)))
hg0.md1.res <- array(0, dim = c(length(nvec), length(hvec)))
hg0.tm1.res <- array(0, dim = c(length(nvec), length(hvec)))

set.seed(45) # set random number generator for reproducibility

# save means and trimmed means of g and h distributions
mu.g <- numeric(length(gvec))
tmu.g <- numeric(length(gvec))
for(G in 2:length(gvec)){
  mu.g[G] <- ghmean(gvec[G], 0)$mean # mean of g and h distribution
  tmu.g[G] <- ghtrim(tr = .2, g = gvec[G], h = 0) # trimmed mean of g and h distribution
}
mu.h <- numeric(length(hvec))
tmu.h <- numeric(length(hvec))
for(H in 1:length(hvec)){
  mu.h[H] <- ghmean(gFIX, hvec[H])$mean # mean of g and h distribution
  tmu.h[H] <- ghtrim(tr = .2, g = gFIX, h = hvec[H]) # trimmed mean of g and h distribution
}
# for g=0.3, the mean is stable for up to h=0.6: repeated calls to ghdist give very similar values which
# mean(ghdist(10000000, g=0.3, h=0.7))

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

  #save(".Random.seed",file="random_state_seed.RData") ## save current state
  #load("random_state_seed.RData")

  if(S == 1){
    print(paste0("Simulation: ",S,"/",iter,"..."))
  }
}

```

```

if(S %% 1000 == 0){
  print(paste0("Simulation: ",S,"/",iter,"..."))
  beep(2)
}

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

  for(G in 1:length(gvec)){
    large.sample <- ghdist(max(nvec), g = gvec[G], h = 0)

    # type I errors =====
    # t-test on mean - subtract mu so the mean is zero on average
    pval <- trimci.pval(large.sample[1:nvec[N]] - mu.g[G], tr=0)
    if(pval<=.05) g.m0.res[N,G] <- g.m0.res[N,G] + 1 # number of type I errors
    # t-test on 20% trimmed mean - subtract tmu so the trimmed mean is zero on average
    pval <- trimci.pval(large.sample[1:nvec[N]] - tmu.g[G], tr=0.2)
    if(pval<=.05) g.tm0.res[N,G] <- g.tm0.res[N,G] + 1 # number of type I errors
    # median test
    # ghdist() generates data from distributions with median = 0, so no need to subtract the median
    pval <- sintv2.pval(large.sample[1:nvec[N]])
    if(pval<=.05) g.md0.res[N,G] <- g.md0.res[N,G] + 1 # number of type I errors

    # true positives =====
    # t-test on mean - subtract mu so the mean is zero on average - add effect size
    pval <- trimci.pval(large.sample[1:nvec[N]] - mu.g[G] + ES, tr=0)
    if(pval<=.05) g.m1.res[N,G] <- g.m1.res[N,G] + 1 # number of type I errors
    # t-test on 20% trimmed mean - subtract tmu so the trimmed mean is zero on average
    pval <- trimci.pval(large.sample[1:nvec[N]] - tmu.g[G] + ES, tr=0.2)
    if(pval<=.05) g.tm1.res[N,G] <- g.tm1.res[N,G] + 1 # number of type I errors
    # median test
    # ghdist() generates data from distributions with median = 0, so no need to subtract the median
    pval <- sintv2.pval(large.sample[1:nvec[N]] + ES)
    if(pval<=.05) g.md1.res[N,G] <- g.md1.res[N,G] + 1 # number of type I errors
  }

  for(H in 1:length(hvec)){
    large.sample <- ghdist(max(nvec), g = gFIX, h = hvec[H])

    # type I errors =====
    # t-test on mean
    pval <- trimci.pval(large.sample[1:nvec[N]] - mu.h[H], tr=0)
    if(pval<=.05) h.m0.res[N,H] <- h.m0.res[N,H] + 1 # number of type I errors
    # t-test on 20% trimmed mean
    pval <- trimci.pval(large.sample[1:nvec[N]] - tmu.h[H], tr=0.2)
    if(pval<=.05) h.tm0.res[N,H] <- h.tm0.res[N,H] + 1 # number of type I errors
    # median test
    pval <- sintv2.pval(large.sample[1:nvec[N]])
    if(pval<=.05) h.md0.res[N,H] <- h.md0.res[N,H] + 1 # number of type I errors

    # true positives =====
    pval <- trimci.pval(large.sample[1:nvec[N]] - mu.h[H] + ES, tr=0)
    if(pval<=.05) h.m1.res[N,H] <- h.m1.res[N,H] + 1 # number of type I errors
    # t-test on 20% trimmed mean

```

```

pval <- trimci.pval(large.sample[1:nvec[N]] - tmu.h[H] + ES, tr=0.2)
if(pval<=.05) h.tm1.res[N,H] <- h.tm1.res[N,H] + 1 # number of type I errors
# median test
pval <- sintv2.pval(large.sample[1:nvec[N]] + ES)
if(pval<=.05) h.md1.res[N,H] <- h.md1.res[N,H] + 1 # number of type I errors

# symmetric distribution to repliace results from 2018 stats guide
# With g=0, mean, trimmed mean and median all = zero
large.sample <- ghdist(max(nvec), g = 0, h = hvec[H])

# type I errors =====
# t-test on mean
pval <- trimci.pval(large.sample[1:nvec[N]], tr=0)
if(pval<=.05) hg0.m0.res[N,H] <- hg0.m0.res[N,H] + 1 # number of type I errors
# t-test on 20% trimmed mean
pval <- trimci.pval(large.sample[1:nvec[N]], tr=0.2)
if(pval<=.05) hg0.tm0.res[N,H] <- hg0.tm0.res[N,H] + 1 # number of type I errors
# median test
pval <- sintv2.pval(large.sample[1:nvec[N]])
if(pval<=.05) hg0.md0.res[N,H] <- hg0.md0.res[N,H] + 1 # number of type I errors

# true positives =====
pval <- trimci.pval(large.sample[1:nvec[N]] + ES, tr=0)
if(pval<=.05) hg0.m1.res[N,H] <- hg0.m1.res[N,H] + 1 # number of type I errors
# t-test on 20% trimmed mean
pval <- trimci.pval(large.sample[1:nvec[N]] + ES, tr=0.2)
if(pval<=.05) hg0.tm1.res[N,H] <- hg0.tm1.res[N,H] + 1 # number of type I errors
# median test
pval <- sintv2.pval(large.sample[1:nvec[N]] + ES)
if(pval<=.05) hg0.md1.res[N,H] <- hg0.md1.res[N,H] + 1 # number of type I errors

}
}
}

save(g.m0.res, g.md0.res, g.tm0.res,
     g.m1.res, g.md1.res, g.tm1.res,
     h.m0.res, h.md0.res, h.tm0.res,
     h.m1.res, h.md1.res, h.tm1.res,
     hg0.m0.res, hg0.md0.res, hg0.tm0.res,
     hg0.m1.res, hg0.md1.res, hg0.tm1.res,
     iter, gvec, hvec, nvec,
     file = "./data/power_gh_onesample_res.RData")

beep(8)

```

Save files and get the type I error rate and power for each combination of tests, sample sizes, effect sizes and distributions.

```

load("./data/power_gh_onesample_res.RData")

# number of positive results -> probabilities
g.m0.res <- g.m0.res / iter
g.md0.res <- g.md0.res / iter

```

```

g.tm0.res <- g.tm0.res / iter
g.m1.res <- g.m1.res / iter
g.md1.res <- g.md1.res / iter
g.tm1.res <- g.tm1.res / iter

h.m0.res <- h.m0.res / iter
h.md0.res <- h.md0.res / iter
h.tm0.res <- h.tm0.res / iter
h.m1.res <- h.m1.res / iter
h.md1.res <- h.md1.res / iter
h.tm1.res <- h.tm1.res / iter

hg0.m0.res <- hg0.m0.res / iter
hg0.md0.res <- hg0.md0.res / iter
hg0.tm0.res <- hg0.tm0.res / iter
hg0.m1.res <- hg0.m1.res / iter
hg0.md1.res <- hg0.md1.res / iter
hg0.tm1.res <- hg0.tm1.res / iter

```

Results

G distributions: Type I error rate

We make two figures, one for g, one for h, each showing 6 panels: a column of 3 panels for false positives, a column of 3 panels for true positives. Panels show results for mean, 20% trimmed mean and median.

```

# g: column 1 = false positives -----

xticklab <- c("10", "", "30", "", "50", "", "70", "", "90", "", "150", "200", "300")

# MEAN
# create data frame
fm <- cbind(nvec, g.m0.res)
colnames(fm) <- c("x", paste(gvec))
df <- as_tibble(fm)
df <- tidyr::gather(df, g, y, 2:(length(gvec)+1))
df$g <- as.factor(df$g)

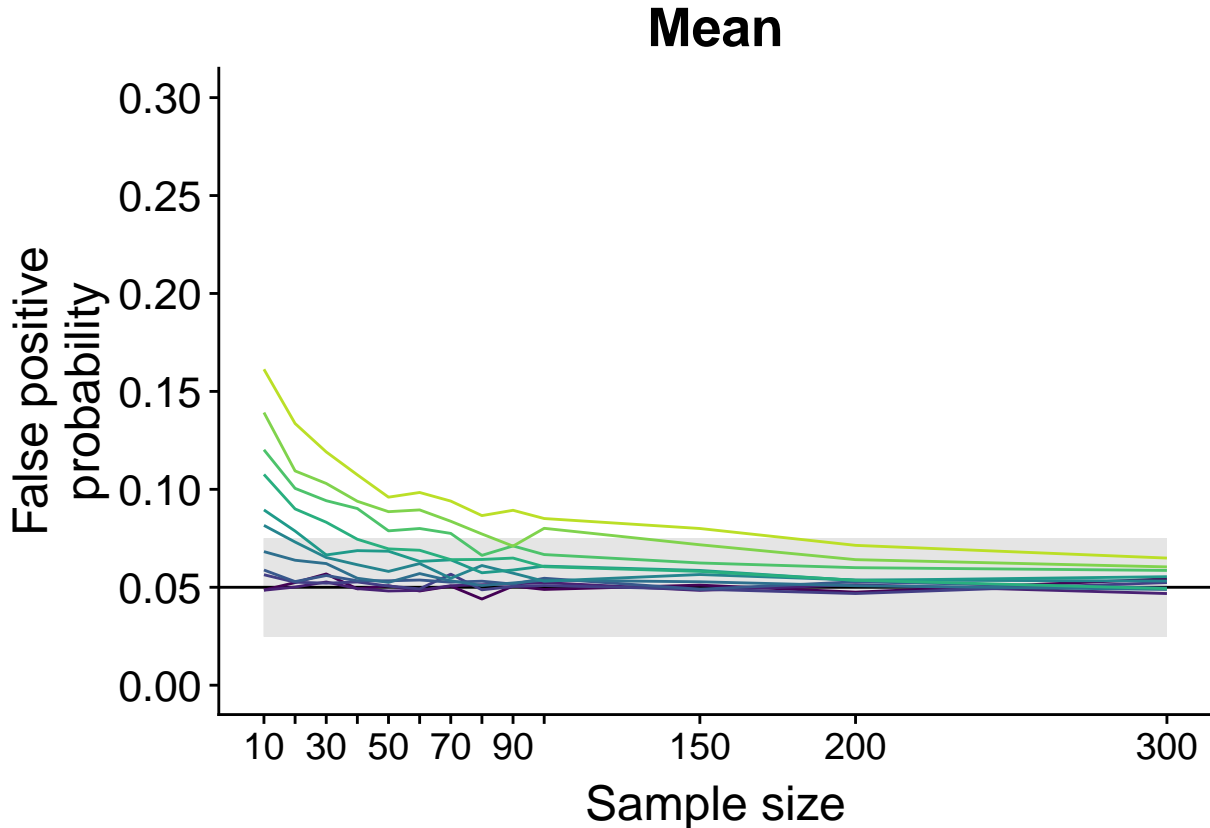
# make plot
pg.m0 <- ggplot(df, aes(x, y, group = g)) +
  geom_ribbon(ymin = 0.025, ymax = 0.075, fill = "grey90") + # Bradley's (1978) satisfactory range
  geom_abline(intercept = 0.05, slope = 0) + # 0.05 reference line
  geom_line(aes(colour=g), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +
  theme(axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 16),
        axis.title.y = element_text(size = 18),
        plot.title = element_text(size = 20),
        legend.position = "none") +
  scale_x_continuous(breaks = nvec,
                    labels = xticklab) +

```

```

    scale_y_continuous(limits = c(0,0.3),
                      breaks = seq(0, 0.4, 0.05)) +
    labs(x = "Sample size", y = "False positive\nprobability", title = "Mean") +
    guides(colour = guide_legend(override.aes = list(size = 3)))
pg.m0

```



```

# 20% TRIMMED MEAN
# create data frame
fm <- cbind(nvec, g.tm0.res)
colnames(fm) <- c("x", paste(gvec))
df <- as_tibble(fm)
df <- tidyr::gather(df, g, y, 2:(length(gvec)+1))
df$g <- as.factor(df$g)

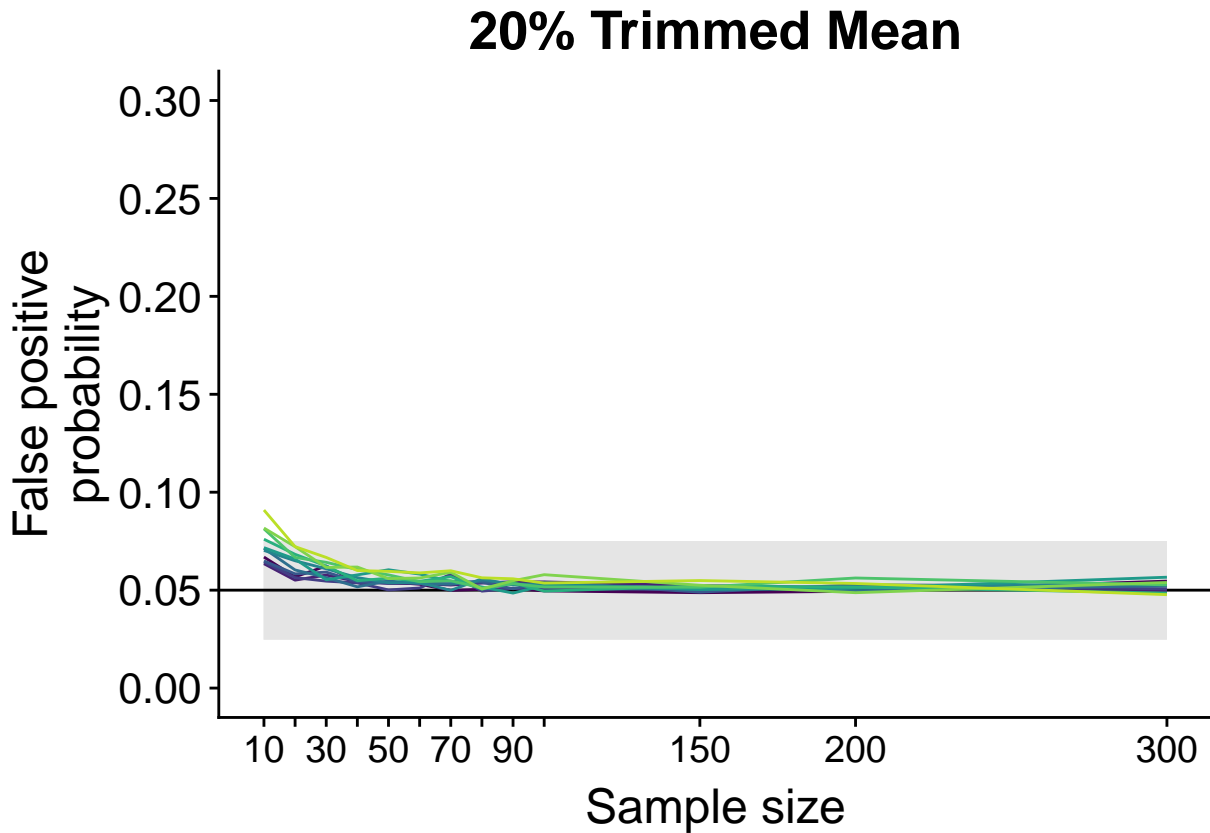
# make plot
pg.tm0 <- ggplot(df, aes(x, y, group = g)) +
  geom_ribbon(ymin = 0.025, ymax = 0.075, fill = "grey90") + # Bradley's (1978) satisfactory range
  geom_abline(intercept = 0.05, slope = 0) + # 0.05 reference line
  geom_line(aes(colour=g), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +
  theme(axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 16),
        axis.title.y = element_text(size = 18),
        plot.title = element_text(size = 20),
        legend.position = "none") +
  scale_x_continuous(breaks = nvec,

```

```

      labels = xticklab) +
      scale_y_continuous(limits = c(0,0.3),
                        breaks = seq(0, 0.4, 0.05)) +
      labs(x = "Sample size", y = "False positive\nprobability", title = "20% Trimmed Mean") +
      guides(colour = guide_legend(override.aes = list(size = 3)))
pg.tm0

```



```

# MEDIAN
# create data frame
fm <- cbind(nvec, g.md0.res)
colnames(fm) <- c("x", paste(gvec))
df <- as_tibble(fm)
df <- tidyr::gather(df, g, y, 2:(length(gvec)+1))
df$g <- as.factor(df$g)

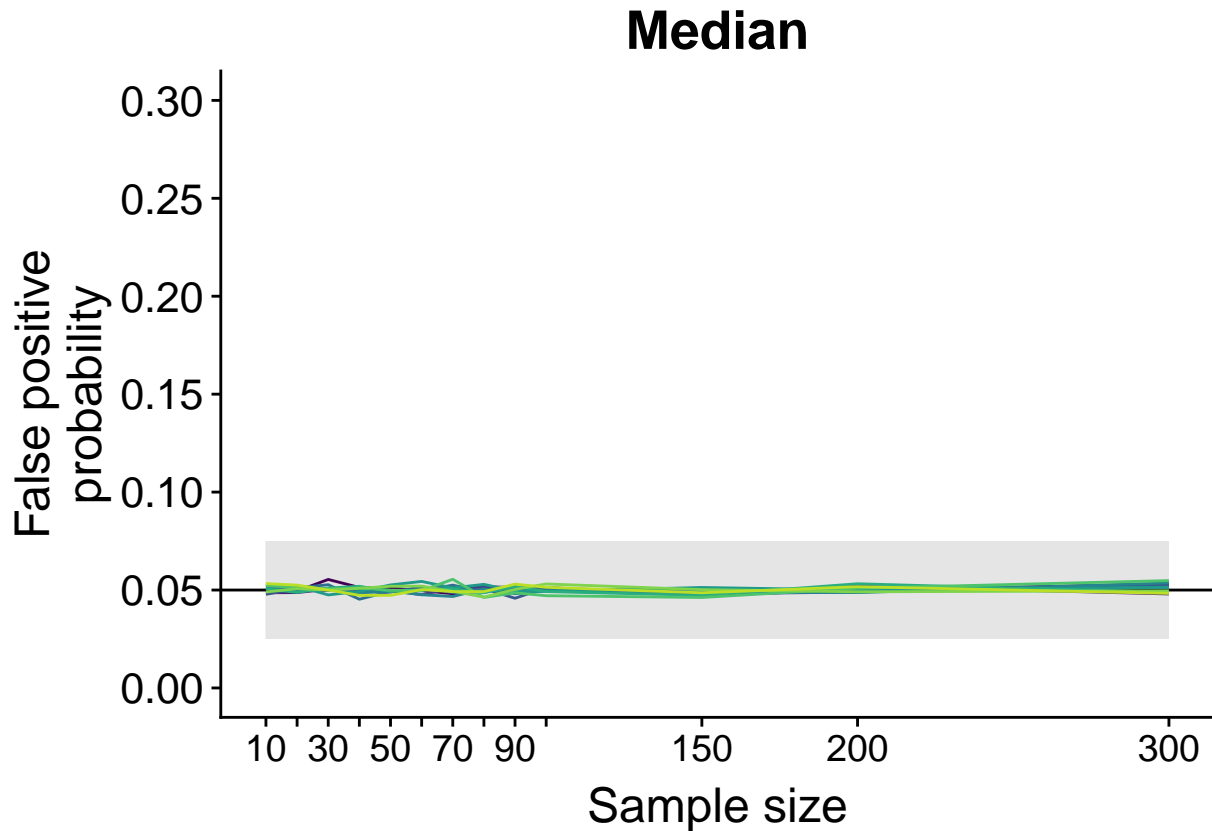
# make plot
pg.md0 <- ggplot(df, aes(x, y, group = g)) +
  geom_ribbon(ymin = 0.025, ymax = 0.075, fill = "grey90") + # Bradley's (1978) satisfactory range
  geom_abline(intercept = 0.05, slope = 0) + # 0.05 reference line
  geom_line(aes(colour=g), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +
  theme(axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 16),
        axis.title.y = element_text(size = 18),
        plot.title = element_text(size = 20),
        legend.position = "none") +

```

```

scale_x_continuous(breaks = nvec,
                   labels = xticklab) +
  scale_y_continuous(limits = c(0,0.3),
                    breaks = seq(0, 0.4, 0.05)) +
  labs(x = "Sample size", y = "False positive\nprobability", title = "Median") +
  guides(colour = guide_legend(override.aes = list(size = 3)))
pg.md0

```



G distributions: Power

```

# g: column 2 = true positives -----

# MEAN
# create data frame
fm <- cbind(nvec, g.m1.res)
colnames(fm) <- c("x", paste(gvec))
df <- as_tibble(fm)
df <- tidyr::gather(df, g, y, 2:(length(gvec)+1))
df$g <- as.factor(df$g)

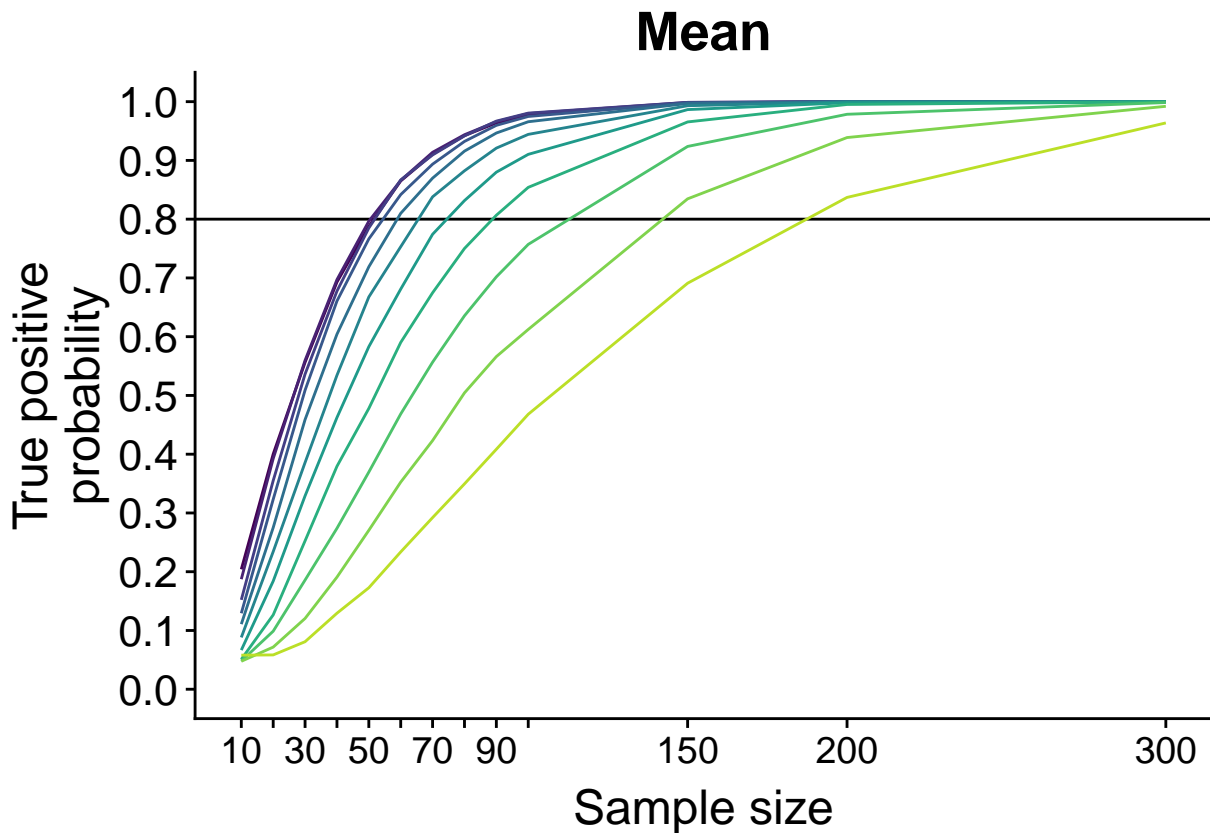
# make plot
pg.m1 <- ggplot(df, aes(x, y, group = g)) +
  geom_abline(intercept = 0.80, slope = 0) + # 0.80 reference line
  geom_line(aes(colour=g), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +

```

```

    theme(axis.title.x = element_text(size = 18),
          axis.text.x = element_text(size = 14),
          axis.text.y = element_text(size = 16),
          axis.title.y = element_text(size = 18),
          plot.title = element_text(size = 20),
          legend.position = "none") +
    scale_x_continuous(breaks = nvec,
                      labels = xticklab) +
    scale_y_continuous(limits = c(0,1),
                      breaks = seq(0, 1, 0.1)) +
    labs(x = "Sample size", y = "True positive\nprobability", title = "Mean") +
    guides(colour = guide_legend(override.aes = list(size = 3)))
pg.m1

```



```

# 20% TRIMMED MEAN
# create data frame
fm <- cbind(nvec, g.tm1.res)
colnames(fm) <- c("x", paste(gvec))
df <- as_tibble(fm)
df <- tidyr::gather(df, g, y, 2:(length(gvec)+1))
df$g <- as.factor(df$g)

# make plot
pg.tm1 <- ggplot(df, aes(x, y, group = g)) +
  geom_abline(intercept = 0.80, slope = 0) + # 0.80 reference line
  geom_line(aes(colour=g), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +

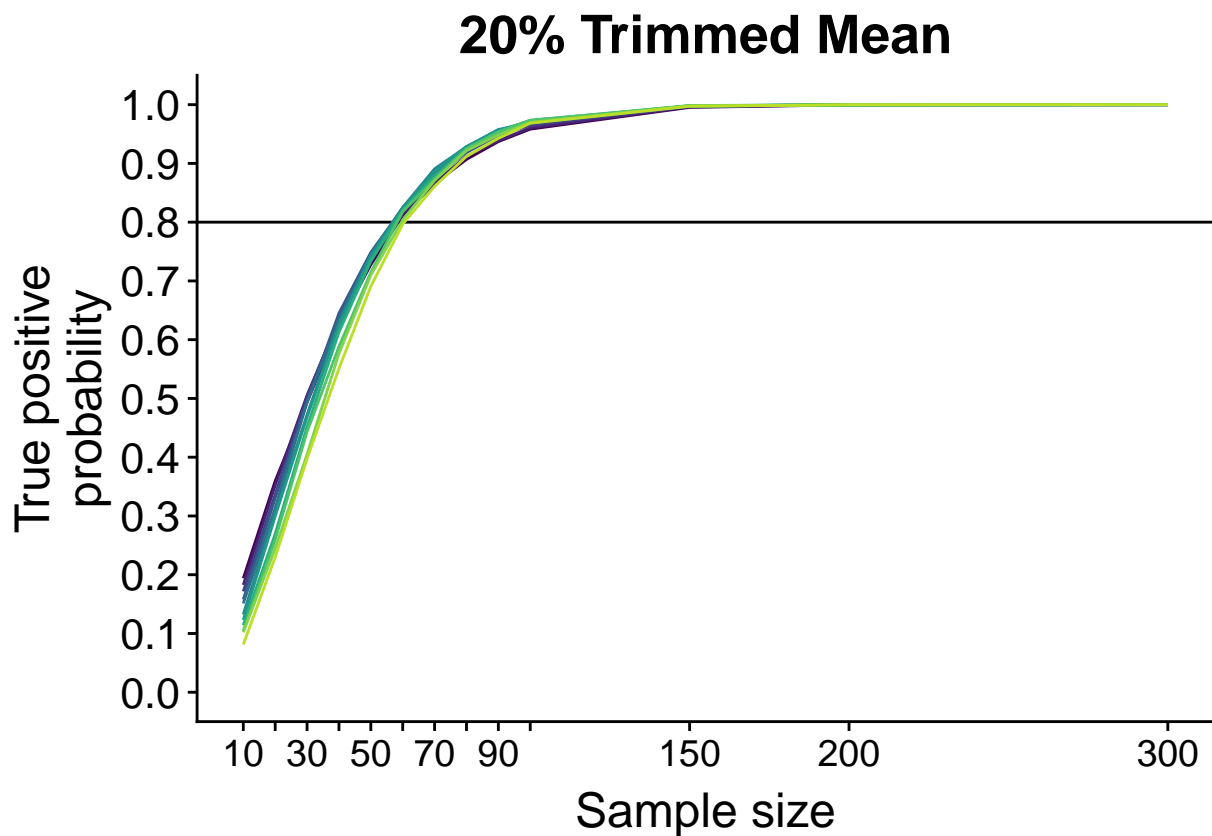
```



```

    theme(axis.title.x = element_text(size = 18),
          axis.text.x = element_text(size = 14),
          axis.text.y = element_text(size = 16),
          axis.title.y = element_text(size = 18),
          plot.title = element_text(size = 20),
          legend.position = "none") +
    scale_x_continuous(breaks = nvec,
                      labels = xticklab) +
    scale_y_continuous(limits = c(0,1),
                      breaks = seq(0, 1, 0.1)) +
    labs(x = "Sample size", y = "True positive\nprobability", title = "20% Trimmed Mean") +
    guides(colour = guide_legend(override.aes = list(size = 3)))
pg.tm1

```



```

# MEDIAN
# create data frame
fm <- cbind(nvec, g.md1.res)
colnames(fm) <- c("x", paste(gvec))
df <- as_tibble(fm)
df <- tidyr::gather(df, g, y, 2:(length(gvec)+1))
df$g <- as.factor(df$g)

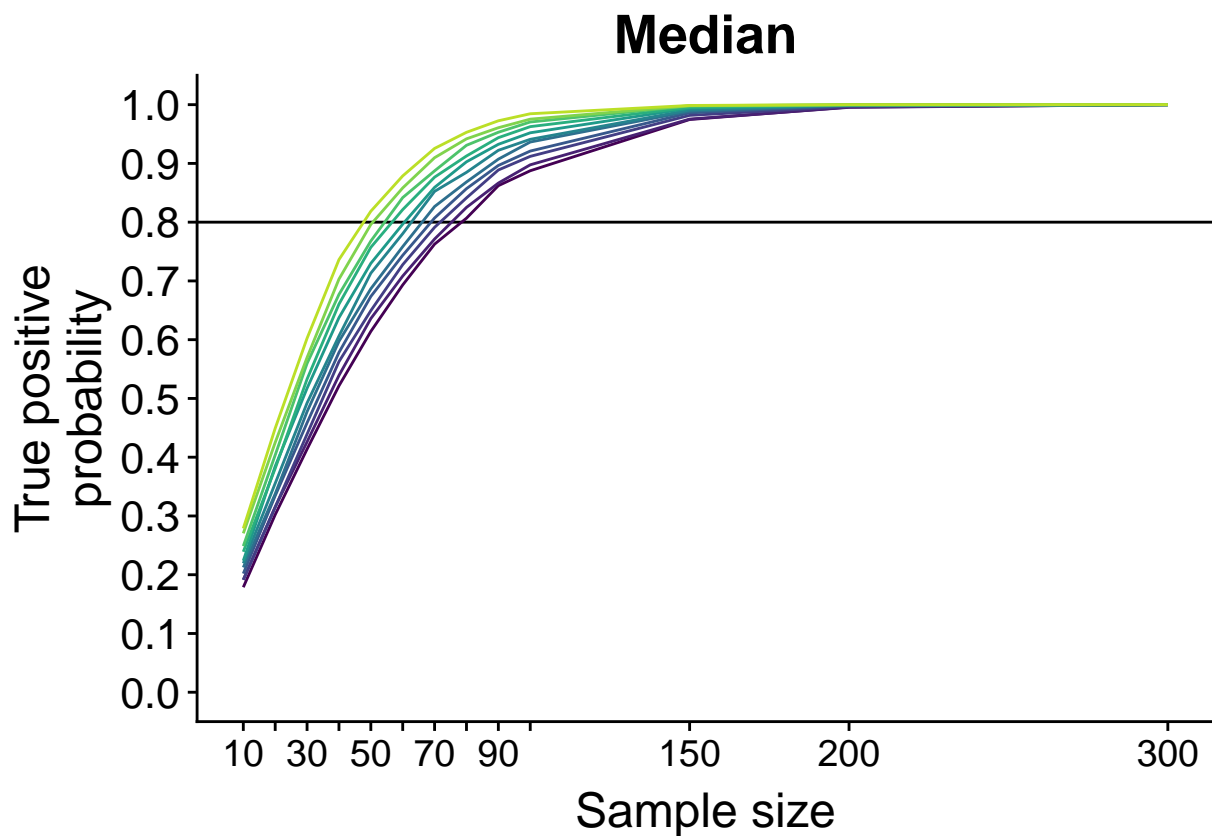
# make plot
pg.md1 <- ggplot(df, aes(x, y, group = g)) +
  geom_abline(intercept = 0.80, slope = 0) + # 0.80 reference line
  geom_line(aes(colour=g), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +

```

```

    theme(axis.title.x = element_text(size = 18),
          axis.text.x = element_text(size = 14),
          axis.text.y = element_text(size = 16),
          axis.title.y = element_text(size = 18),
          plot.title = element_text(size = 20),
          legend.position = "none") +
    scale_x_continuous(breaks = nvec,
                      labels = xticklab) +
    scale_y_continuous(limits = c(0,1),
                      breaks = seq(0, 1, 0.1)) +
    labs(x = "Sample size", y = "True positive\nprobability", title = "Median") +
    guides(colour = guide_legend(override.aes = list(size = 3)))
pg.md1

```



H distributions with $g=0$: Type I error rate

```

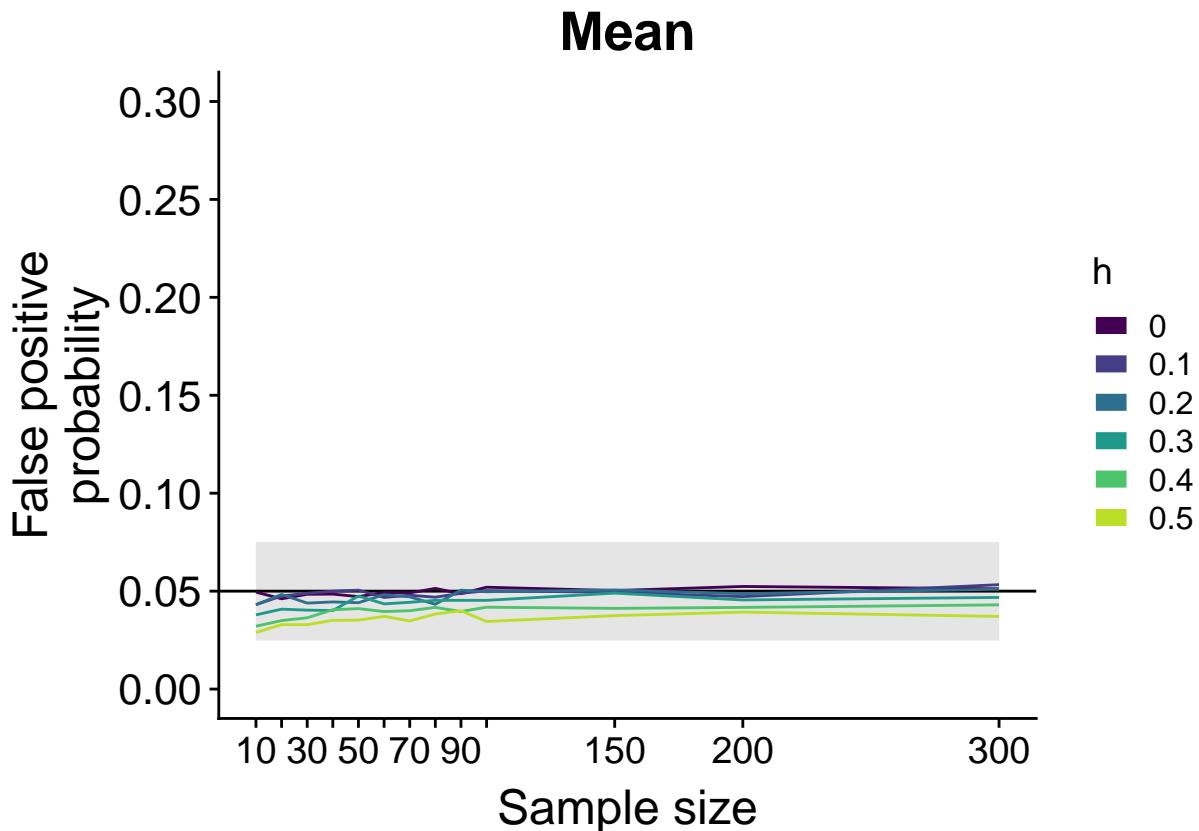
# h: column 1 = false positives -----
# MEAN
# create data frame
fm <- cbind(nvec, hg0.m0.res)
colnames(fm) <- c("x", paste(hvec))
df <- as_tibble(fm)
df <- tidyr::gather(df, h, y, 2:(length(hvec)+1))
df$h <- as.factor(df$h)

```

```

# make plot
ph.m0 <- ggplot(df, aes(x, y, group = h)) +
  geom_ribbon(ymin = 0.025, ymax = 0.075, fill = "grey90") + # Bradley's (1978) satisfactory range
  geom_abline(intercept = 0.05, slope = 0) + # 0.05 reference line
  geom_line(aes(colour=h), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +
  theme(axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 16),
        axis.title.y = element_text(size = 18),
        plot.title = element_text(size = 20)) +
  scale_x_continuous(breaks = nvec,
                    labels = xticklab) +
  scale_y_continuous(limits = c(0,0.3),
                    breaks = seq(0, 0.4, 0.05)) +
  labs(x = "Sample size", y = "False positive\nprobability", title = "Mean") +
  guides(colour = guide_legend(override.aes = list(size = 3)))
ph.m0

```



```

# 20% TRIMMED MEAN
# create data frame
fm <- cbind(nvec, hg0.tm0.res)
colnames(fm) <- c("x", paste(hvec))
df <- as_tibble(fm)
df <- tidyr::gather(df, h, y, 2:(length(hvec)+1))
df$h <- as.factor(df$h)

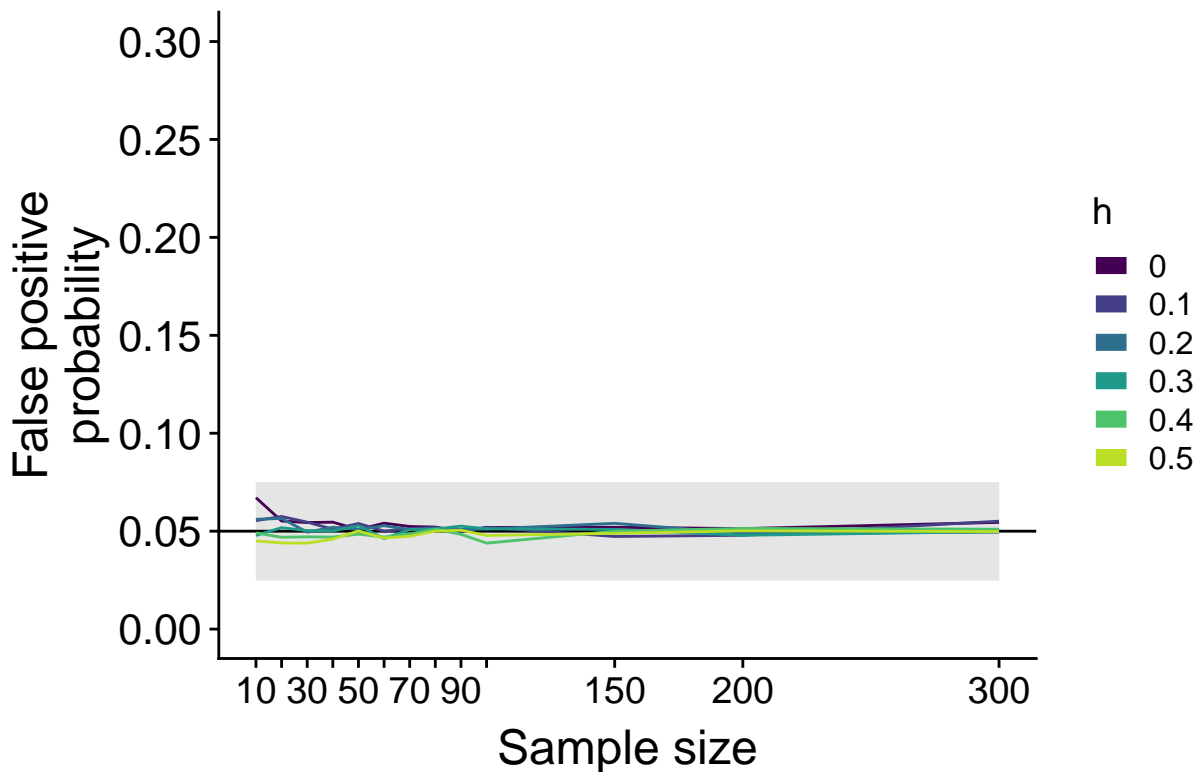
```

```

# make plot
ph.tm0 <- ggplot(df, aes(x, y, group = h)) +
  geom_ribbon(ymin = 0.025, ymax = 0.075, fill = "grey90") + # Bradley's (1978) satisfactory range
  geom_abline(intercept = 0.05, slope = 0) + # 0.05 reference line
  geom_line(aes(colour=h), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +
  theme(axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 16),
        axis.title.y = element_text(size = 18),
        plot.title = element_text(size = 20)) +
  scale_x_continuous(breaks = nvec,
                    labels = xticklab) +
  scale_y_continuous(limits = c(0,0.3),
                    breaks = seq(0, 0.4, 0.05)) +
  labs(x = "Sample size", y = "False positive\nprobability", title = "20% Trimmed Mean") +
  guides(colour = guide_legend(override.aes = list(size = 3)))
ph.tm0

```

20% Trimmed Mean



```

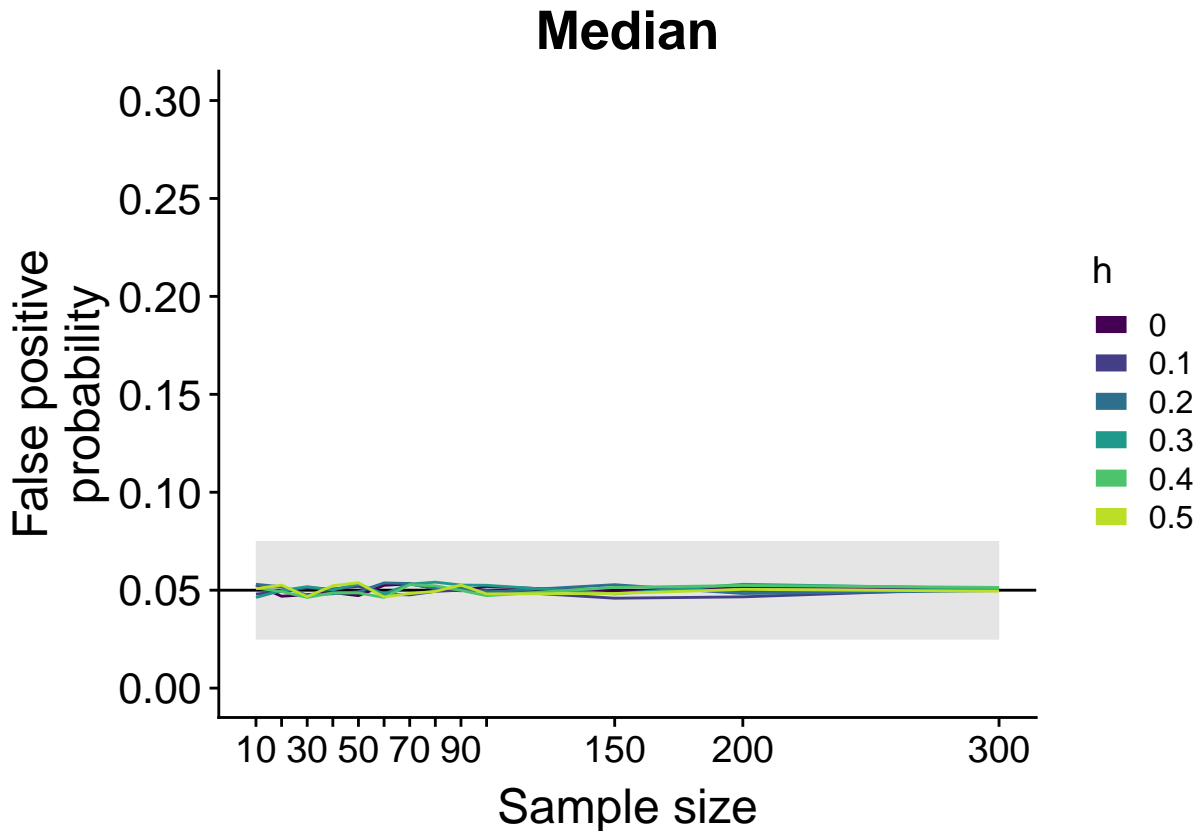
# MEDIAN
# create data frame
fm <- cbind(nvec, hg0.md0.res)
colnames(fm) <- c("x", paste(hvec))
df <- as_tibble(fm)
df <- tidyr::gather(df, h, y, 2:(length(hvec)+1))
df$h <- as.factor(df$h)

```

```

# make plot
ph.md0 <- ggplot(df, aes(x, y, group = h)) +
  geom_ribbon(ymin = 0.025, ymax = 0.075, fill = "grey90") + # Bradley's (1978) satisfactory range
  geom_abline(intercept = 0.05, slope = 0) + # 0.05 reference line
  geom_line(aes(colour=h), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +
  theme(axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 16),
        axis.title.y = element_text(size = 18),
        plot.title = element_text(size = 20)) +
  scale_x_continuous(breaks = nvec,
                    labels = xticklab) +
  scale_y_continuous(limits = c(0,0.3),
                    breaks = seq(0, 0.4, 0.05)) +
  labs(x = "Sample size", y = "False positive\nprobability", title = "Median") +
  guides(colour = guide_legend(override.aes = list(size = 3)))
ph.md0

```



H distributions with $g=0$: Power

```

# h: column 2 = true positives -----
# MEAN
# create data frame

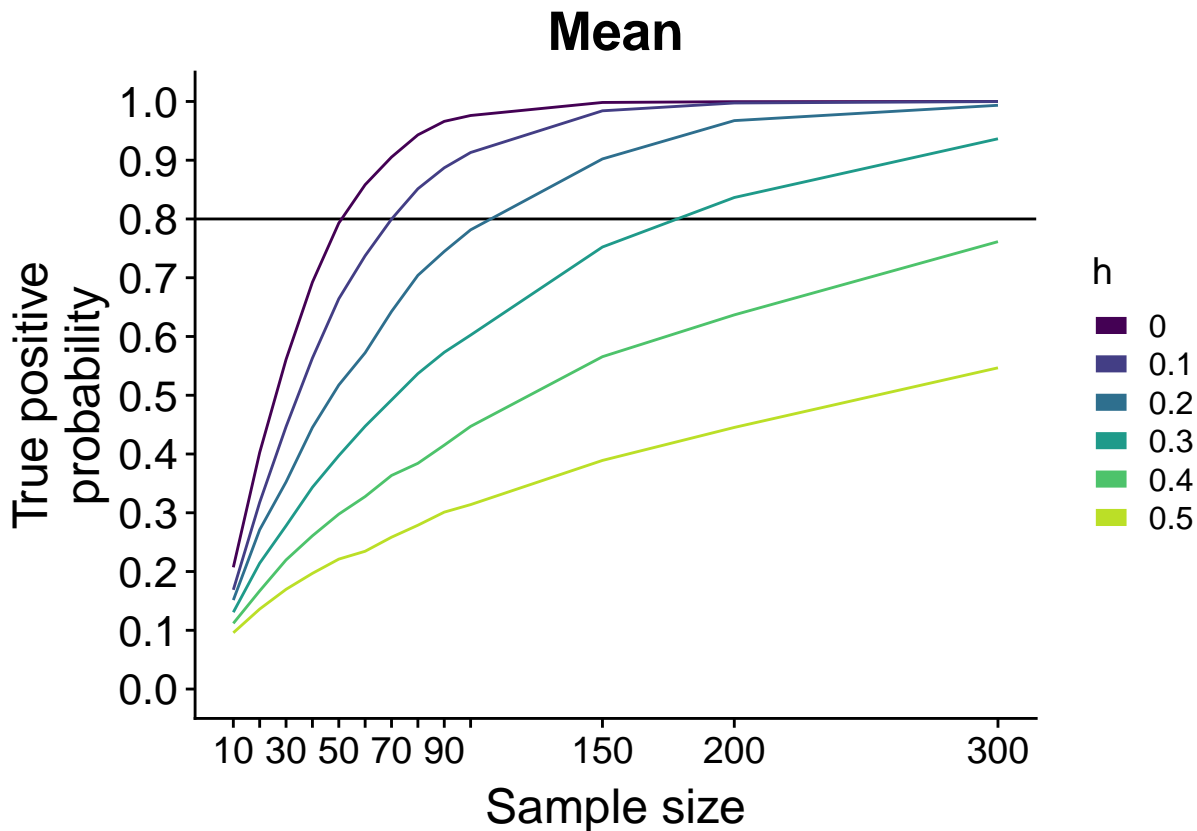
```

```

fm <- cbind(nvec, hg0.m1.res)
colnames(fm) <- c("x", paste(hvec))
df <- as_tibble(fm)
df <- tidyr::gather(df, h, y, 2:(length(hvec)+1))
df$h <- as.factor(df$h)

# make plot
ph.m1 <- ggplot(df, aes(x, y, group = h)) +
  geom_abline(intercept = 0.80, slope = 0) + # 0.80 reference line
  geom_line(aes(colour=h), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +
  theme(axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 16),
        axis.title.y = element_text(size = 18),
        plot.title = element_text(size = 20)) +
  scale_x_continuous(breaks = nvec,
                    labels = xticklab) +
  scale_y_continuous(limits = c(0,1),
                    breaks = seq(0, 1, 0.1)) +
  labs(x = "Sample size", y = "True positive\nprobability", title = "Mean") +
  guides(colour = guide_legend(override.aes = list(size = 3)))
ph.m1

```



```

# 20% TRIMMED MEAN
# create data frame
fm <- cbind(nvec, hg0.tm1.res)

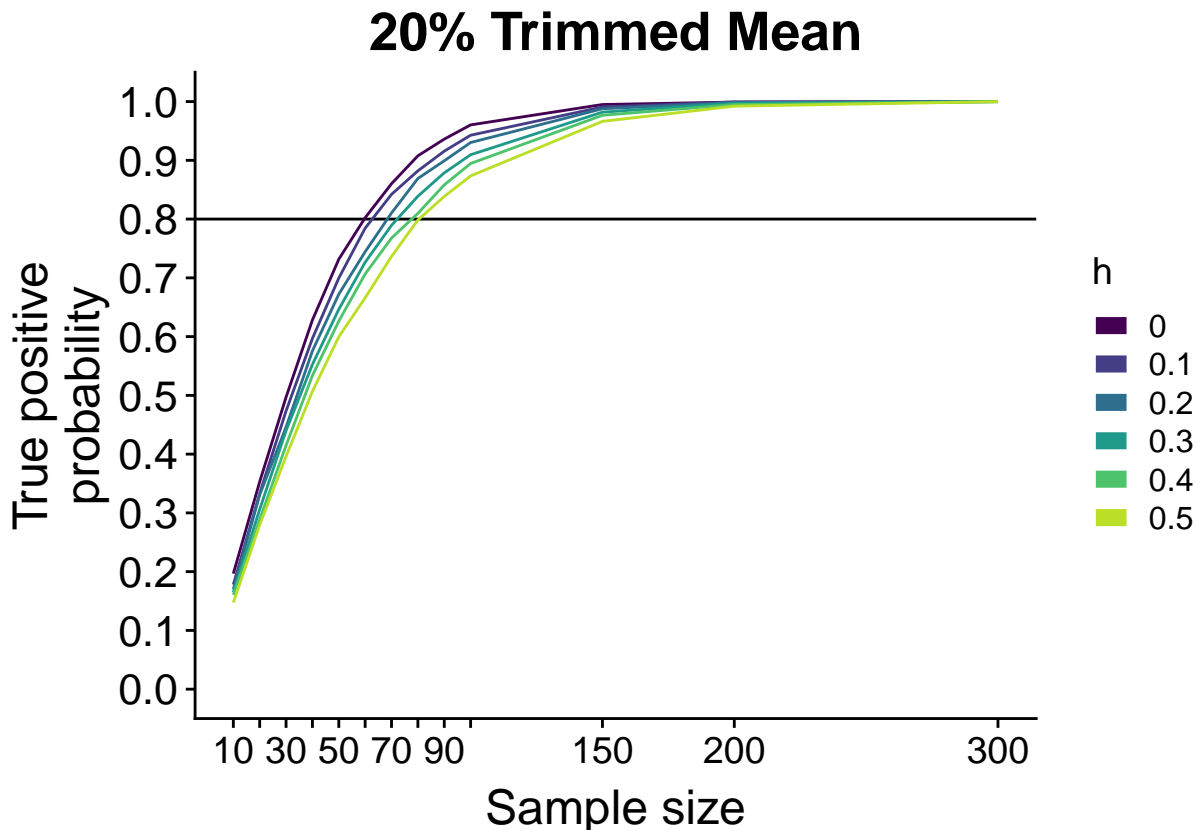
```

```

colnames(fm) <- c("x", paste(hvec))
df <- as_tibble(fm)
df <- tidyr::gather(df, h, y, 2:(length(hvec)+1))
df$h <- as.factor(df$h)

# make plot
ph.tm1 <- ggplot(df, aes(x, y, group = h)) +
  geom_abline(intercept = 0.80, slope = 0) + # 0.80 reference line
  geom_line(aes(colour=h), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +
  theme(axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 16),
        axis.title.y = element_text(size = 18),
        plot.title = element_text(size = 20)) +
  scale_x_continuous(breaks = nvec,
                    labels = xticklab) +
  scale_y_continuous(limits = c(0,1),
                    breaks = seq(0, 1, 0.1)) +
  labs(x = "Sample size", y = "True positive\nprobability", title = "20% Trimmed Mean") +
  guides(colour = guide_legend(override.aes = list(size = 3)))
ph.tm1

```



```

# MEDIAN
# create data frame
fm <- cbind(nvec, hg0.md1.res)
colnames(fm) <- c("x", paste(hvec))

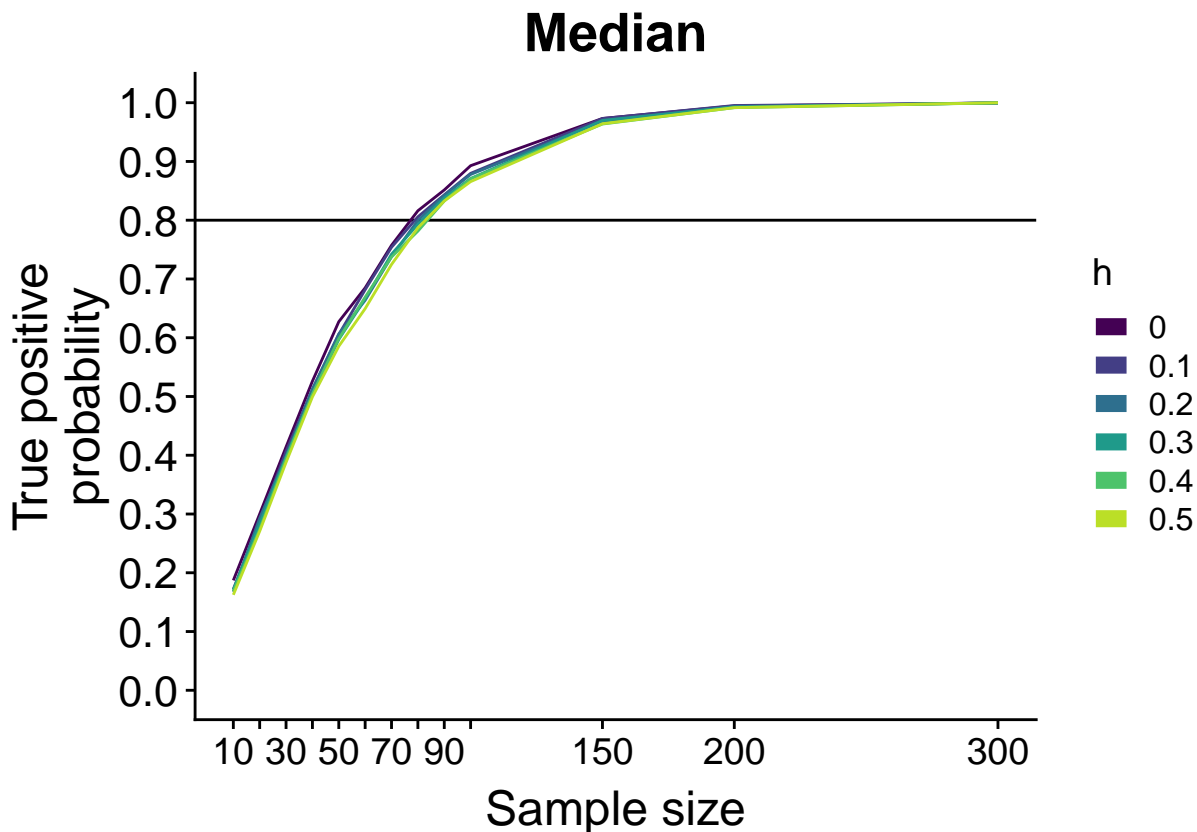
```

```

df <- as_tibble(fm)
df <- tidyr::gather(df, h, y, 2:(length(hvec)+1))
df$h <- as.factor(df$h)

# make plot
ph.md1 <- ggplot(df, aes(x, y, group = h)) +
  geom_abline(intercept = 0.80, slope = 0) + # 0.80 reference line
  geom_line(aes(colour=h), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +
  theme(axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 16),
        axis.title.y = element_text(size = 18),
        plot.title = element_text(size = 20)) +
  scale_x_continuous(breaks = nvec,
                    labels = xticklab) +
  scale_y_continuous(limits = c(0,1),
                    breaks = seq(0, 1, 0.1)) +
  labs(x = "Sample size", y = "True positive\nprobability", title = "Median") +
  guides(colour = guide_legend(override.aes = list(size = 3)))
ph.md1

```



H distributions with $g=0.3$: Type I error rate

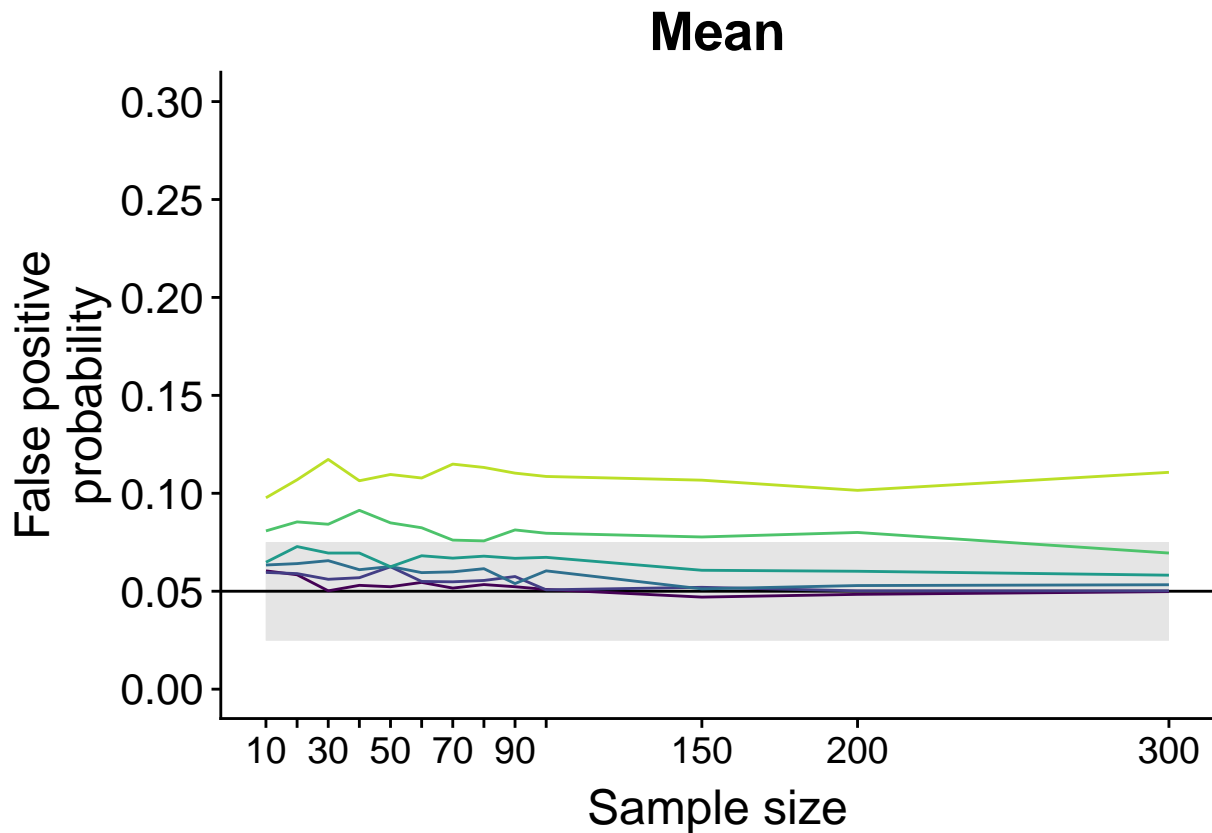

```

# h: column 1 = false positives -----

# MEAN
# create data frame
fm <- cbind(nvec, h.m0.res)
colnames(fm) <- c("x", paste(hvec))
df <- as_tibble(fm)
df <- tidyr::gather(df, h, y, 2:(length(hvec)+1))
df$h <- as.factor(df$h)

# make plot
ph.m0 <- ggplot(df, aes(x, y, group = h)) +
  geom_ribbon(ymin = 0.025, ymax = 0.075, fill = "grey90") + # Bradley's (1978) satisfactory range
  geom_abline(intercept = 0.05, slope = 0) + # 0.05 reference line
  geom_line(aes(colour=h), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +
  theme(axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 16),
        axis.title.y = element_text(size = 18),
        plot.title = element_text(size = 20),
        legend.position = "none") +
  scale_x_continuous(breaks = nvec,
                    labels = xticklab) +
  scale_y_continuous(limits = c(0,0.3),
                    breaks = seq(0, 0.4, 0.05)) +
  labs(x = "Sample size", y = "False positive\nprobability", title = "Mean") +
  guides(colour = guide_legend(override.aes = list(size = 3)))
ph.m0

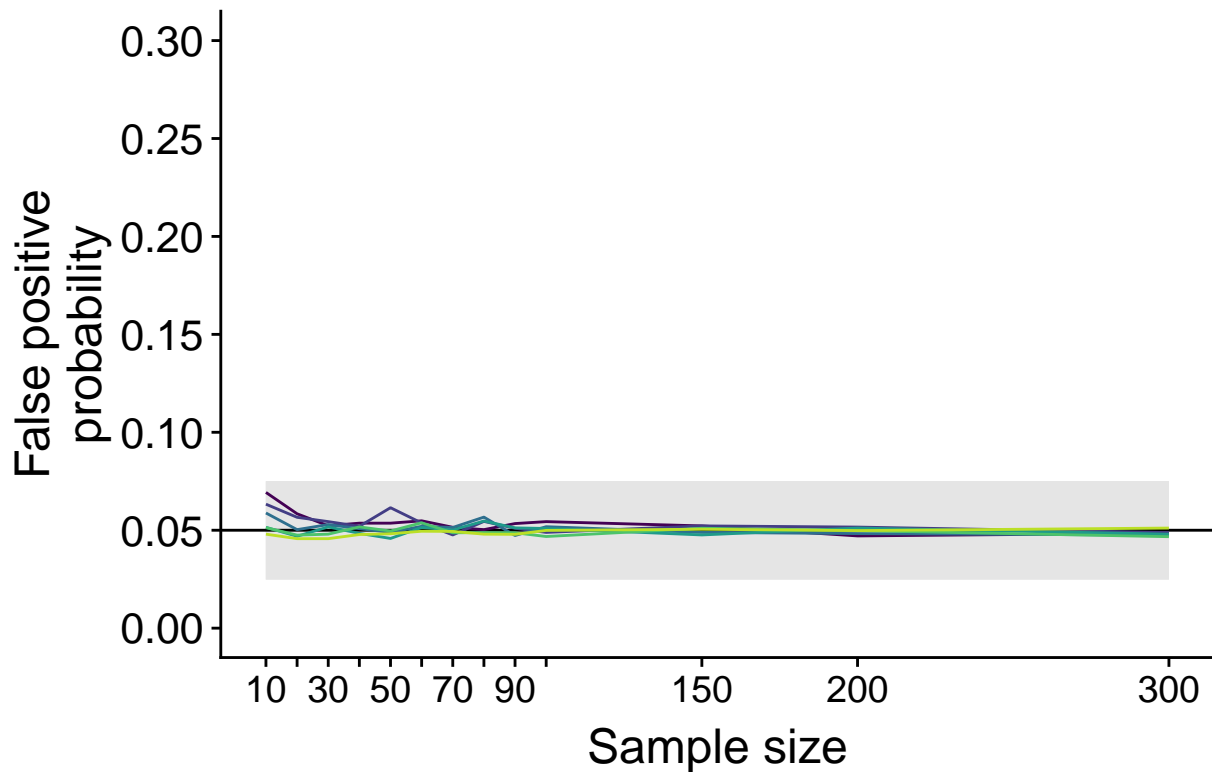
```



```
# 20% TRIMMED MEAN
# create data frame
fm <- cbind(nvec, h.tm0.res)
colnames(fm) <- c("x", paste(hvec))
df <- as_tibble(fm)
df <- tidyr::gather(df, h, y, 2:(length(hvec)+1))
df$h <- as.factor(df$h)

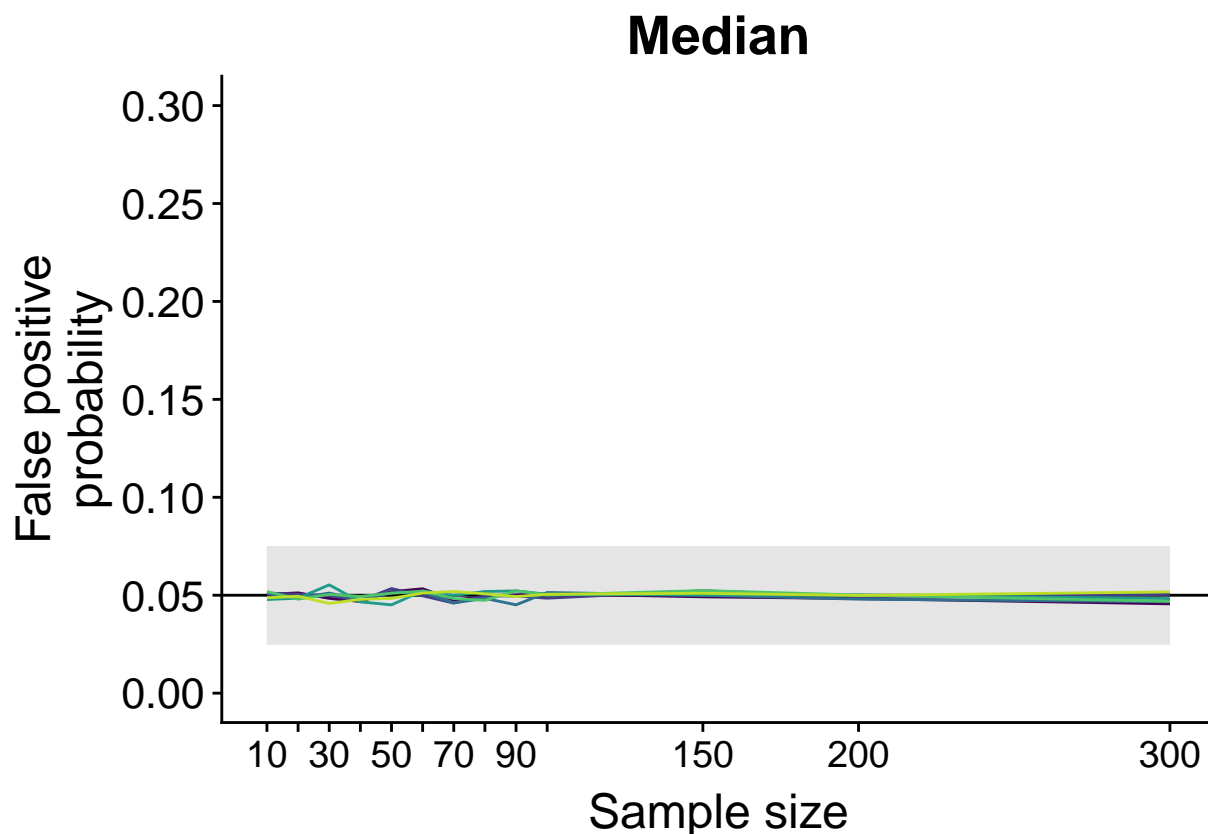
# make plot
ph.tm0 <- ggplot(df, aes(x, y, group = h)) +
  geom_ribbon(ymin = 0.025, ymax = 0.075, fill = "grey90") + # Bradley's (1978) satisfactory range
  geom_abline(intercept = 0.05, slope = 0) + # 0.05 reference line
  geom_line(aes(colour=h), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +
  theme(axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 16),
        axis.title.y = element_text(size = 18),
        plot.title = element_text(size = 20),
        legend.position = "none") +
  scale_x_continuous(breaks = nvec,
                    labels = xticklab) +
  scale_y_continuous(limits = c(0,0.3),
                    breaks = seq(0, 0.4, 0.05)) +
  labs(x = "Sample size", y = "False positive\nprobability", title = "20% Trimmed Mean") +
  guides(colour = guide_legend(override.aes = list(size = 3)))
ph.tm0
```

20% Trimmed Mean



```
# MEDIAN
# create data frame
fm <- cbind(nvec, h.md0.res)
colnames(fm) <- c("x", paste(hvec))
df <- as_tibble(fm)
df <- tidyr::gather(df, h, y, 2:(length(hvec)+1))
df$h <- as.factor(df$h)

# make plot
ph.md0 <- ggplot(df, aes(x, y, group = h)) +
  geom_ribbon(ymin = 0.025, ymax = 0.075, fill = "grey90") + # Bradley's (1978) satisfactory range
  geom_abline(intercept = 0.05, slope = 0) + # 0.05 reference line
  geom_line(aes(colour=h), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +
  theme(axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 16),
        axis.title.y = element_text(size = 18),
        plot.title = element_text(size = 20),
        legend.position = "none") +
  scale_x_continuous(breaks = nvec,
                    labels = xticklab) +
  scale_y_continuous(limits = c(0,0.3),
                    breaks = seq(0, 0.4, 0.05)) +
  labs(x = "Sample size", y = "False positive\nprobability", title = "Median") +
  guides(colour = guide_legend(override.aes = list(size = 3)))
ph.md0
```



H distributions with $g=0.3$: Power

```
# h: column 2 = true positives -----

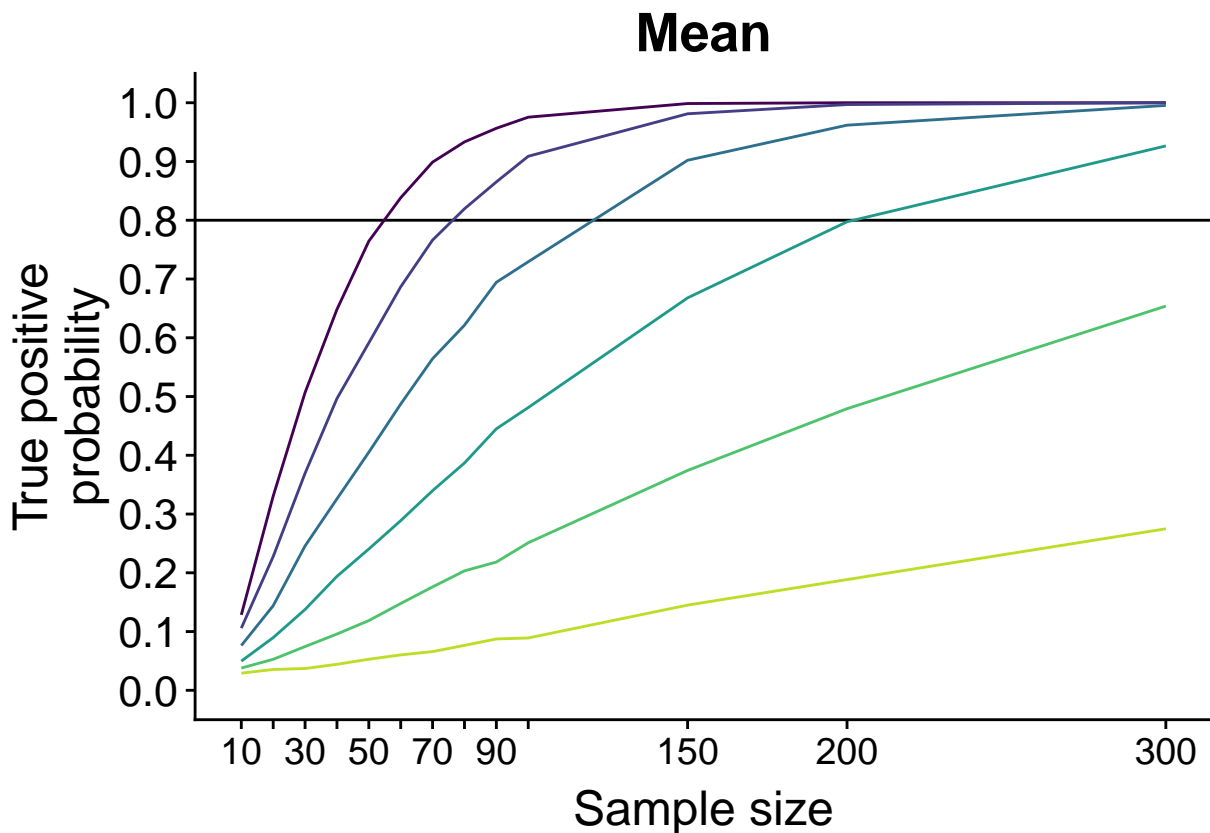
# MEAN
# create data frame
fm <- cbind(nvec, h.m1.res)
colnames(fm) <- c("x", paste(hvec))
df <- as_tibble(fm)
df <- tidyr::gather(df, h, y, 2:(length(hvec)+1))
df$h <- as.factor(df$h)

# make plot
ph.m1 <- ggplot(df, aes(x, y, group = h)) +
  geom_abline(intercept = 0.80, slope = 0) + # 0.80 reference line
  geom_line(aes(colour=h), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +
  theme(axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 16),
        axis.title.y = element_text(size = 18),
        plot.title = element_text(size = 20),
        legend.position = "none") +
  scale_x_continuous(breaks = nvec,
                    labels = xticklab) +
  scale_y_continuous(limits = c(0,1),
```

```

breaks = seq(0, 1, 0.1)) +
labs(x = "Sample size", y = "True positive\nprobability", title = "Mean") +
guides(colour = guide_legend(override.aes = list(size = 3)))
ph.m1

```



```

# 20% TRIMMED MEAN
# create data frame
fm <- cbind(nvec, h.tm1.res)
colnames(fm) <- c("x", paste(hvec))
df <- as_tibble(fm)
df <- tidyr::gather(df, h, y, 2:(length(hvec)+1))
df$h <- as.factor(df$h)

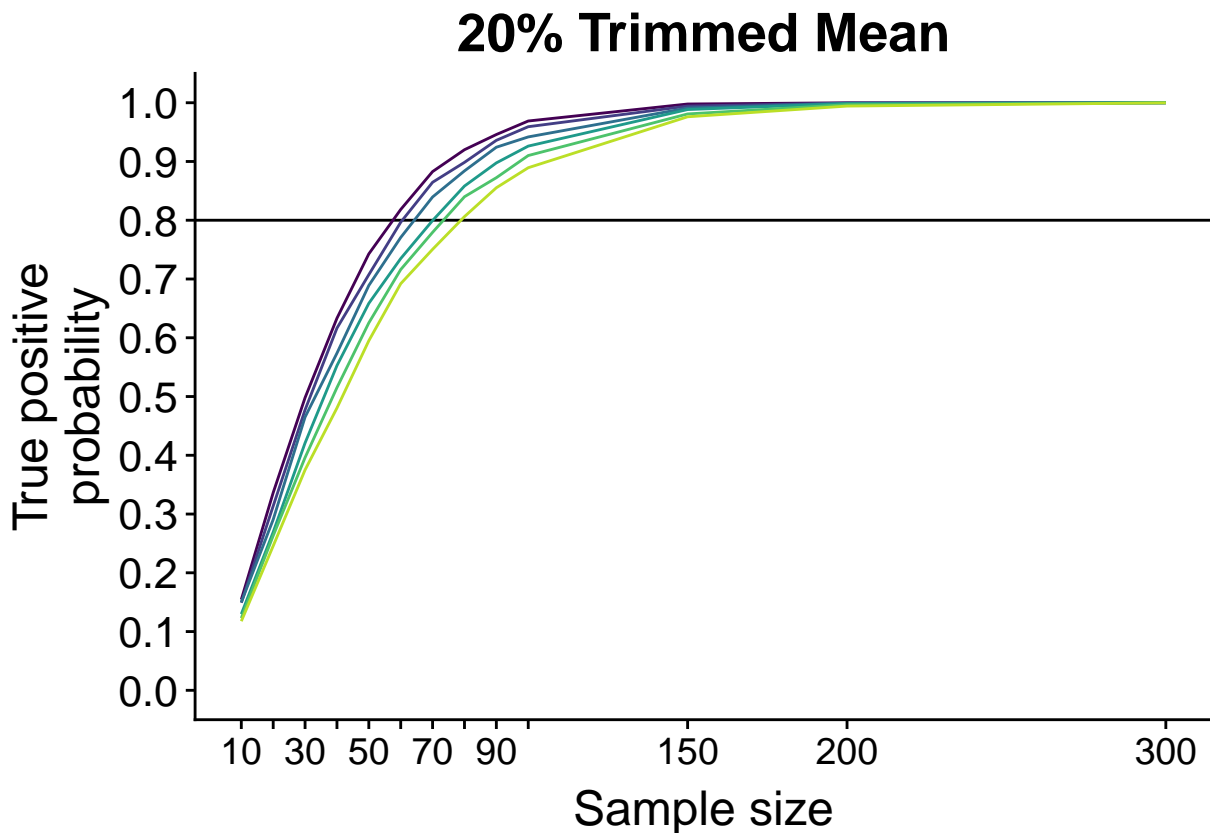
# make plot
ph.tm1 <- ggplot(df, aes(x, y, group = h)) +
  geom_abline(intercept = 0.80, slope = 0) + # 0.80 reference line
  geom_line(aes(colour=h), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +
  theme(axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 16),
        axis.title.y = element_text(size = 18),
        plot.title = element_text(size = 20),
        legend.position = "none") +
  scale_x_continuous(breaks = nvec,
                    labels = xticklab) +
  scale_y_continuous(limits = c(0,1),

```

```

breaks = seq(0, 1, 0.1)) +
labs(x = "Sample size", y = "True positive\nprobability", title = "20% Trimmed Mean") +
guides(colour = guide_legend(override.aes = list(size = 3)))
ph.tm1

```



```

# MEDIAN
# create data frame
fm <- cbind(nvec, h.md1.res)
colnames(fm) <- c("x", paste(hvec))
df <- as_tibble(fm)
df <- tidyr::gather(df, h, y, 2:(length(hvec)+1))
df$h <- as.factor(df$h)

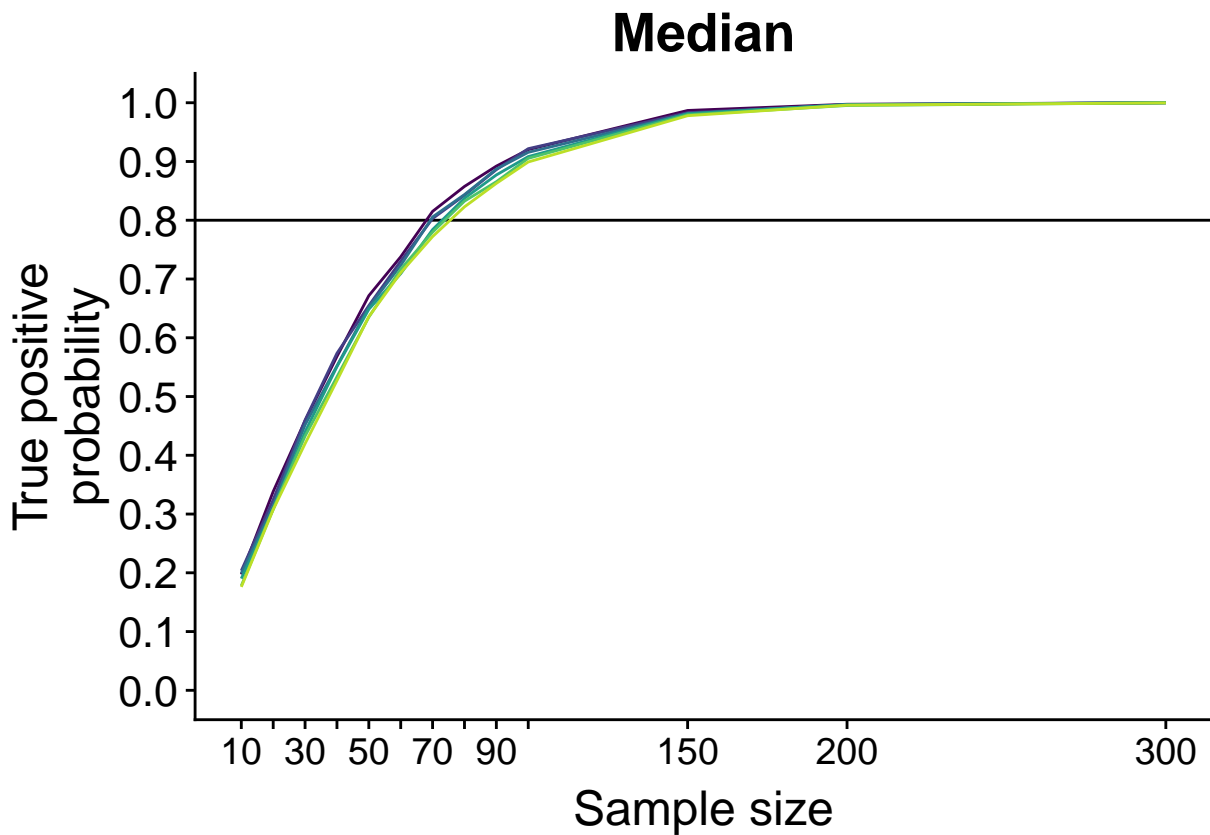
# make plot
ph.md1 <- ggplot(df, aes(x, y, group = h)) +
  geom_abline(intercept = 0.80, slope = 0) + # 0.80 reference line
  geom_line(aes(colour=h), size = 0.5) +
  scale_colour_viridis_d(end = 0.9) +
  theme(axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 16),
        axis.title.y = element_text(size = 18),
        plot.title = element_text(size = 20),
        legend.position = "none") +
  scale_x_continuous(breaks = nvec,
                    labels = xticklab) +
  scale_y_continuous(limits = c(0,1),

```

```

breaks = seq(0, 1, 0.1)) +
labs(x = "Sample size", y = "True positive\nprobability", title = "Median") +
guides(colour = guide_legend(override.aes = list(size = 3)))
ph.md1

```



Make summary figures

Make G figure.

```

sub2 <- cowplot::plot_grid(pg.m0, pg.m1, pg.tm0, pg.tm1, pg.md0, pg.md1,
  labels = c("B", "C", "", "", "", ""),
  ncol = 2,
  nrow = 3,
  label_size = 20,
  hjust = -0.8,
  scale=.95,
  align = "h")

sub1 <- cowplot::plot_grid(NULL, p.g, NULL,
  labels = c("", "A", ""),
  label_size = 20,
  rel_widths = c(1,3,1),
  ncol = 3)

cowplot::plot_grid(sub1, sub2,
  ncol = 1,

```

```

        nrow = 2,
        rel_heights = c(1, 3))

# save figure
ggsave(filename='./figures/figure_gdist_sim.pdf',width=15,height=20)

Make H figure.
sub2 <- cowplot::plot_grid(ph.m0, ph.m1, ph.tm0, ph.tm1, ph.md0, ph.md1,
    labels = c("B", "C", "", "", "", ""),
    ncol = 2,
    nrow = 3,
    label_size = 20,
    hjust = -0.8,
    scale=.95,
    align = "h")

sub1 <- cowplot::plot_grid(NULL, p.h, NULL,
    labels = c("", "A", ""),
    label_size = 20,
    rel_widths = c(1,3,1),
    ncol = 3)

cowplot::plot_grid(sub1, sub2,
    ncol = 1,
    nrow = 2,
    rel_heights = c(1, 3))

# save figure
ggsave(filename='./figures/figure_hdist_sim.pdf',width=15,height=20)

```