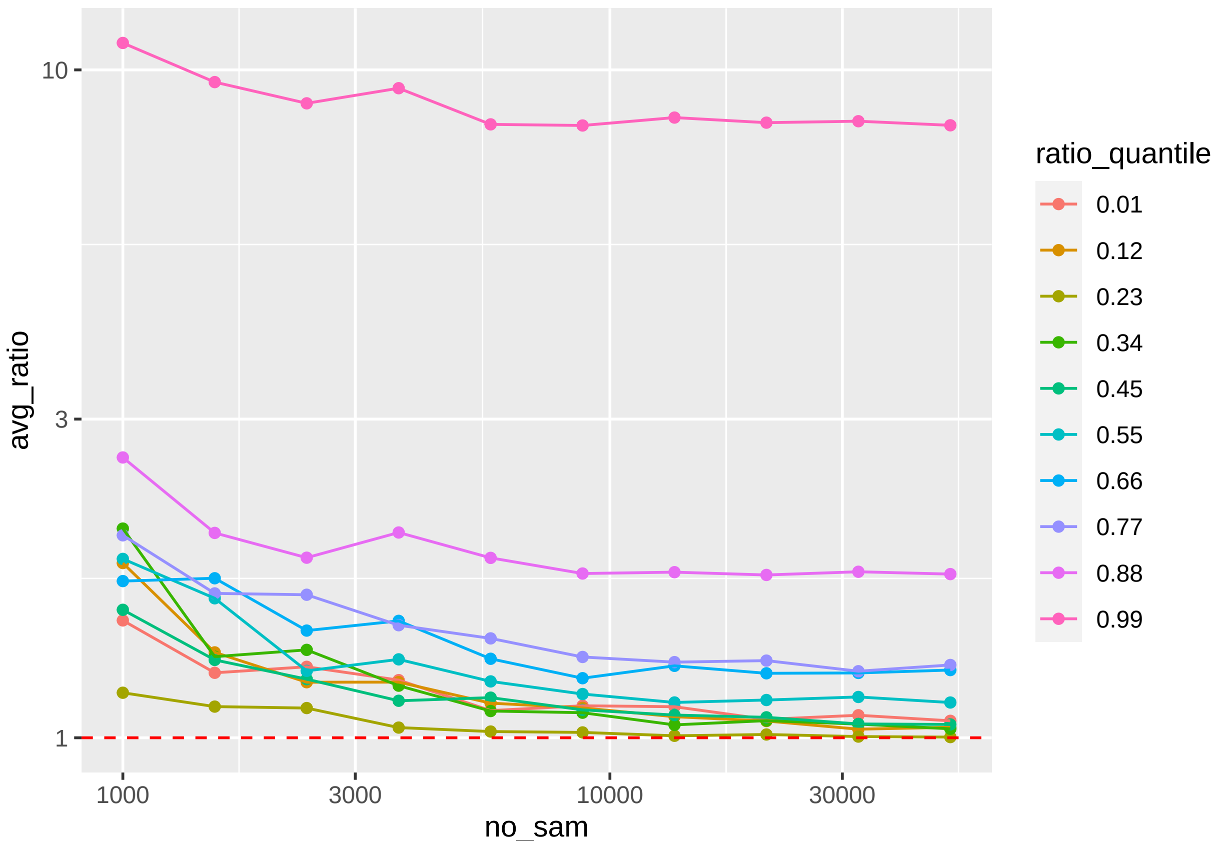
Downsample result analysis

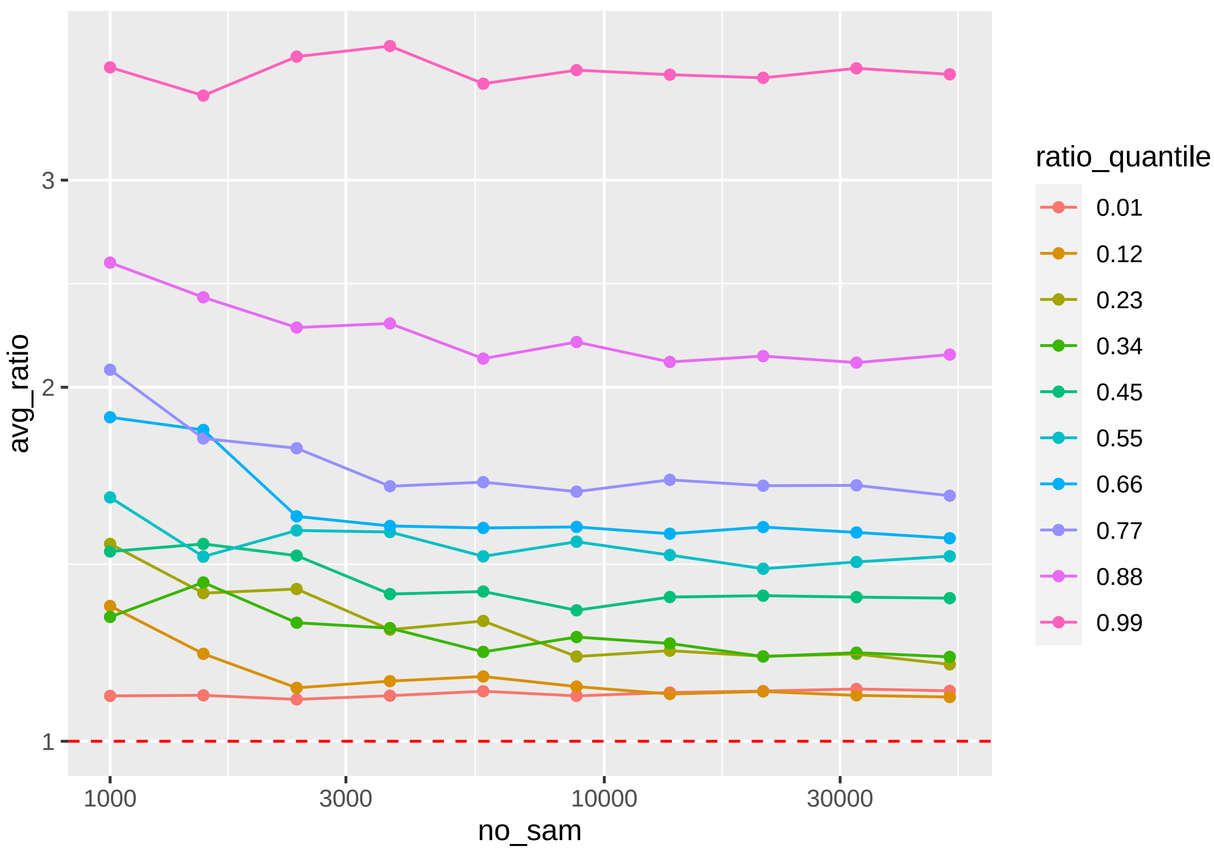
1.Load results

```
library(dplyr)
library(ggplot2)
library(tidyverse)
undershoot <- read_csv("undershoot_refine_fit.csv")[,-1]
overshoot <- read_csv("overshoot_refine_fit.csv")[,-1]
quantile_list <- seq(0.01, 0.99, length.out = 10)
no_sam <- round(exp(seq(log(1e3), log(5e4), length.out = 10)))
# rearrange the data frame
B <- 100
undershoot_df <- data.frame(id = rep(1:B, 10*10),
                            ratio_value = 0,
                            no_sam = 0,
                            ratio_quantile = 0)
overshoot_df <- data.frame(id = rep(1:B, 10*10),
                           ratio_value = 0,
                           no_sam = 0,
                           ratio_quantile = 0)

# i: quantile; j: no of sample
for (i in 1:10) {
  for (j in 1:10) {
    start <- (j - 1 + (i-1)*10)*B +1
    end <- (j + (i-1)*10)*B
    undershoot_df[start:end, 2] <- as.vector(undershoot[(((j-1)*B+1) :(j*B)), (i-1)*3+2])[[1]]
    undershoot_df[start:end, 3] <- rep(no_sam[j], B)
    undershoot_df[start:end, 4] <- rep(quantile_list[i], B)
    overshoot_df[start:end, 2] <- as.vector(overshoot[(((j-1)*B+1) :(j*B)), (i-1)*3+2])[[1]]
    overshoot_df[start:end, 3] <- rep(no_sam[j], B)
    overshoot_df[start:end, 4] <- rep(quantile_list[i], B)
  }
}

# plot for undershoot
under_ratio_avg <- undershoot_df |>
  dplyr::group_by_at(c("no_sam", "ratio_quantile")) |>
  summarise(avg_ratio = mean(ratio_value)) |>
  ungroup()
under_ratio_avg$ratio_quantile <- round(under_ratio_avg$ratio_quantile, 2)
under_ratio_avg$ratio_quantile <- as.character(under_ratio_avg$ratio_quantile)

over_ratio_avg <- overshoot_df |>
  dplyr::group_by_at(c("no_sam", "ratio_quantile")) |>
  summarise(avg_ratio = mean(ratio_value)) |>
  ungroup()
```

```r
over_ratio_avg$ratio_quantile <- round(over_ratio_avg$ratio_quantile, 2)
over_ratio_avg$ratio_quantile <- as.character(over_ratio_avg$ratio_quantile)

under_ratio_avg |>
  ggplot(aes_string(x = "no_sam", y = "avg_ratio", colour = "ratio_quantile")) +
  scale_x_log10() +
  scale_y_log10() +
  geom_point() +
  geom_line() +
  geom_hline(yintercept = 1, linetype = "dashed", colour = "red")
```




```r
over_ratio_avg |>
  ggplot(aes_string(x = "no_sam", y = "avg_ratio", colour = "ratio_quantile")) +
  scale_x_log10() +
  scale_y_log10() +
  geom_point() +
  geom_line() +
  geom_hline(yintercept = 1, linetype = "dashed", colour = "red")
```



2.Quantitative analysis

```r
param_nc <- read_csv("figures/power_exploration/sknorm_tail_prob_500000_resamples_0.96_percentile/param_
param_twosides <- t(param_nc[,-1])
overshoot_ratio <- as.numeric(param_twosides[, 6])
undershoot_ratio <- as.numeric(param_twosides[, 7])
quantile_list <- seq(0.01, 0.99, length.out = 10)
overshoot_set <- data.frame(index = numeric(10), ratio = numeric(10))
undershoot_set <- data.frame(index = numeric(10), ratio = numeric(10))

# find distributions based on right tail
for (r in 1:10){
  dist <- abs(overshoot_ratio[331:660] - quantile(overshoot_ratio[331:660], quantile_list[r]))
  overshoot_set[r, 1] <- which(dist == min(dist))
  overshoot_set[r, 2] <- overshoot_ratio[which(dist == min(dist)) + 330]
  dist <- abs(undershoot_ratio[331:660] - quantile(undershoot_ratio[331:660], quantile_list[r]))
  undershoot_set[r, 1] <- which(dist == min(dist))
  undershoot_set[r, 2] <- undershoot_ratio[which(dist == min(dist)) + 330]
}

# accuracy matrix for undershoot matrix
undershoot_acc <- matrix(under_ratio_avg$avg_ratio / rep(undershoot_set$ratio, 10), 10, 10)
colnames(undershoot_acc) <- as.character(no_sam)
rownames(undershoot_acc) <- as.character(round(quantile_list, 3))
undershoot_acc
```

```
##            1000     1544      2385      3684      5690      8788     13572
## 0.01  1.5465305 1.291071 1.3179851 1.2588489 1.1333088 1.1522920 1.1485451
## 0.119 1.8461840 1.356829 1.2241919 1.2245318 1.1390252 1.1204950 1.0870424
## 0.228 1.1738791 1.119117 1.1134984 1.0410307 1.0269117 1.0238070 1.0119390
## 0.337 2.0429912 1.313591 1.3447114 1.1885807 1.0890790 1.0828511 1.0382896
## 0.446 1.4959361 1.258503 1.1778619 1.0932452 1.1051132 1.0595185 1.0406640
## 0.554 1.7175055 1.499397 1.1674681 1.2147424 1.1258492 1.0778244 1.0470004
## 0.663 1.4925784 1.507163 1.2583553 1.3012183 1.1418587 1.0678190 1.1143479
## 0.772 1.6040176 1.312934 1.3068537 1.1764914 1.1243051 1.0544860 1.0359211
## 0.881 1.6558959 1.276577 1.1720619 1.2784350 1.1713563 1.1098413 1.1150445
## 0.99  0.3820332 0.333807 0.3102554 0.3268537 0.2885369 0.2874422 0.2953761
##           20961     32374     50000
## 0.01  1.0995388 1.1152839 1.0940757
## 0.119 1.0709203 1.0412451 1.0480758
## 0.228 1.0164979 1.0092516 1.0074539
## 0.337 1.0536193 1.0412068 1.0244165
## 0.446 1.0326054 1.0092502 1.0075156
## 0.554 1.0558233 1.0669764 1.0468139
## 0.663 1.0860362 1.0873765 1.0982404
## 0.772 1.0413656 1.0040178 1.0258638
## 0.881 1.1043629 1.1166357 1.1076829
## 0.99  0.2902356 0.2916934 0.2876058
```

```r
# accuracy matrix for overshoot matrix
overshoot_acc <- matrix(over_ratio_avg$avg_ratio / rep(overshoot_set$ratio, 10), 10, 10)
colnames(overshoot_acc) <- as.character(no_sam)
rownames(overshoot_acc) <- as.character(round(quantile_list, 3))
overshoot_acc
```

```
##           1000      1544      2385      3684      5690      8788     13572
## 0.01  1.1410719 1.1424203 1.1333141 1.1415723 1.1515693 1.1411406 1.1486922
## 0.119 1.1626381 1.0587111 0.9904415 1.0035596 1.0127620 0.9928689 0.9785774
## 0.228 1.2385001 1.1246544 1.1338239 1.0471348 1.0649659 0.9933791 1.0048288
## 0.337 1.0074160 1.0777923 0.9960848 0.9856179 0.9407620 0.9685526 0.9564527
## 0.446 1.0653340 1.0811310 1.0566805 0.9801337 0.9850052 0.9494791 0.9743852
## 0.554 1.1071639 0.9859507 1.0377567 1.0346625 0.9865280 1.0151366 0.9889416
## 0.663 1.1996559 1.1700610 0.9880272 0.9696222 0.9658411 0.9678092 0.9549654
## 0.772 1.2111957 1.0587096 1.0386013 0.9641623 0.9718821 0.9539820 0.9763108
## 0.881 1.2022453 1.1233254 1.0587554 1.0672039 0.9962005 1.0291624 0.9897135
## 0.99  0.7921893 0.7496061 0.8090039 0.8259417 0.7672288 0.7878671 0.7808766
##           20961     32374     50000
## 0.01  1.1519172 1.1566490 1.1525240
## 0.119 0.9834655 0.9759259 0.9728011
## 0.228 0.9939430 0.9983364 0.9782855
## 0.337 0.9322542 0.9393728 0.9316795
## 0.446 0.9769932 0.9742804 0.9722826
## 0.554 0.9628465 0.9756772 0.9867646
## 0.663 0.9675221 0.9574431 0.9465985
## 0.772 0.9651509 0.9658167 0.9464360
## 0.881 1.0011947 0.9884499 1.0040925
## 0.99  0.7761137 0.7907123 0.7814143
```