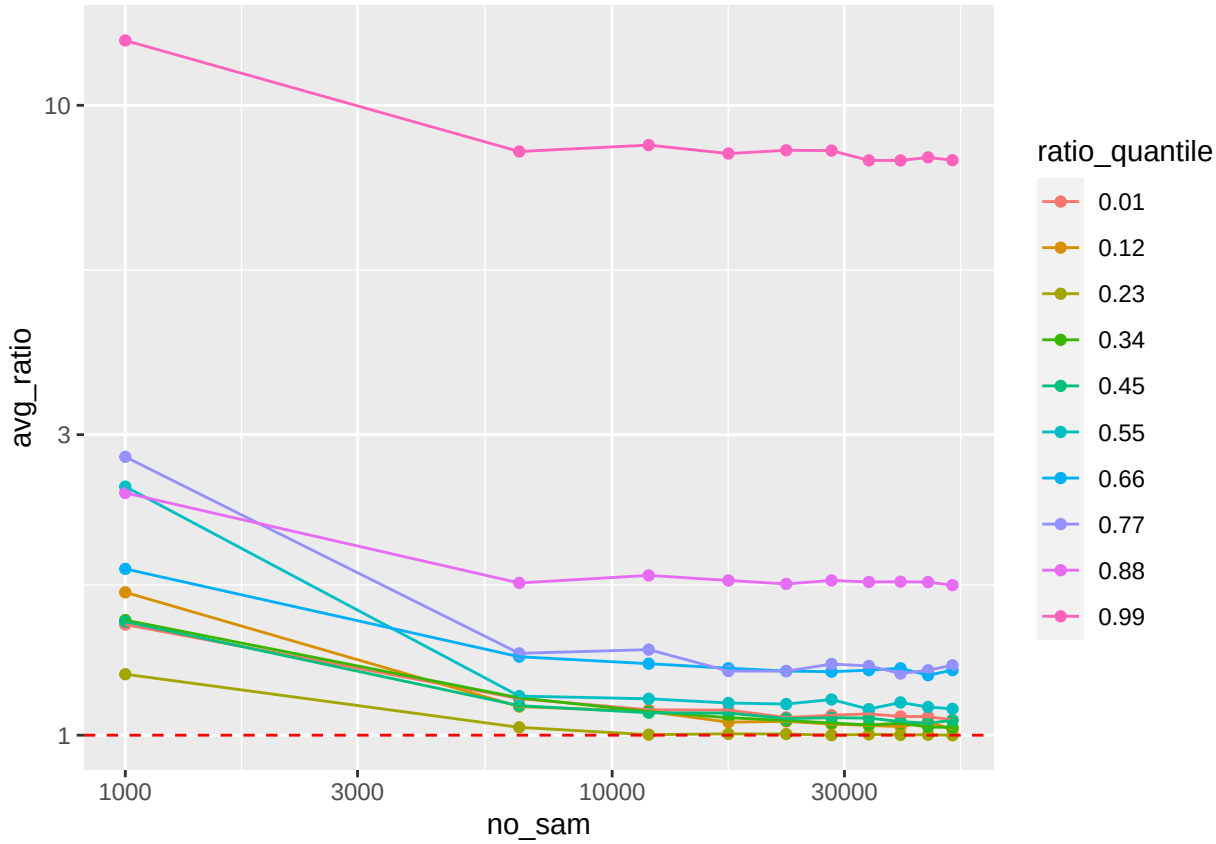
Downsample result analysis

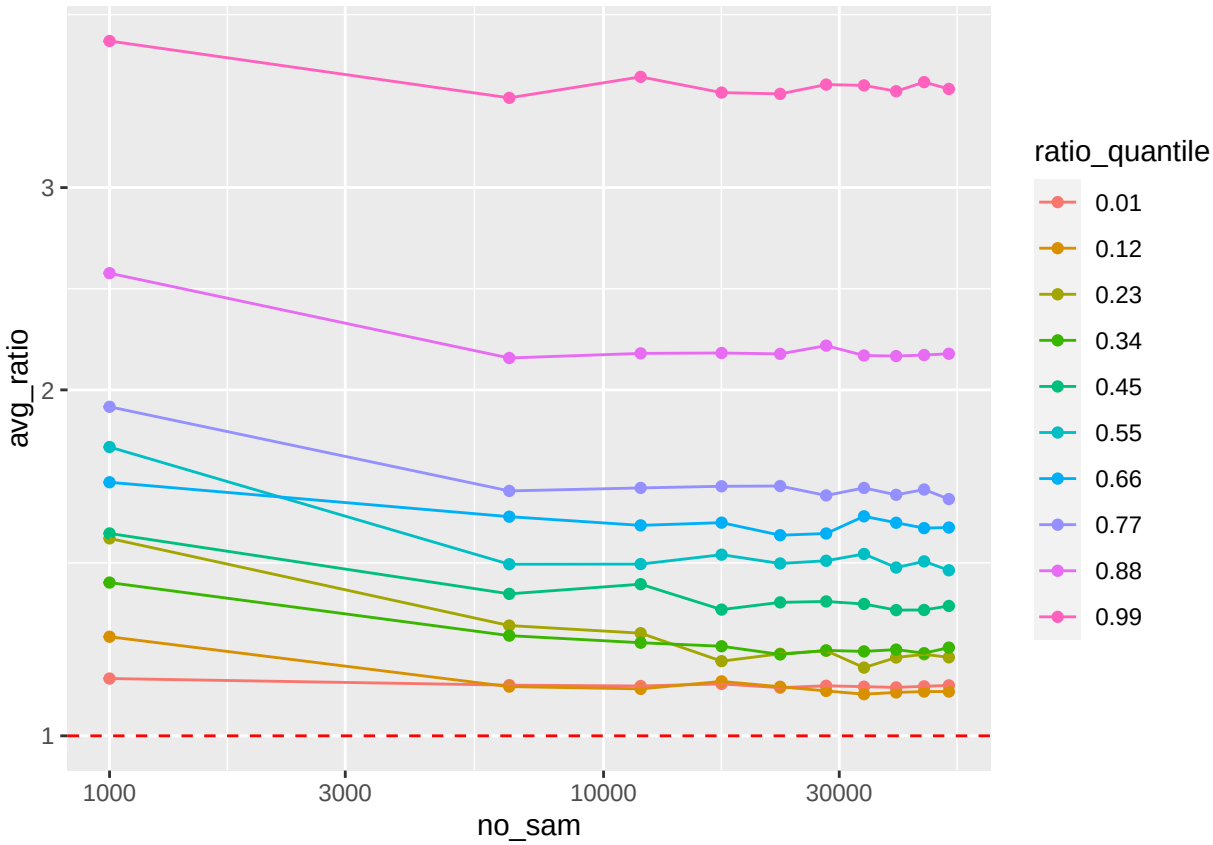
1.Load results

```
library(dplyr)
library(ggplot2)
library(tidyverse)
undershoot <- read_csv("undershoot.csv")[,-1]
overshoot <- read_csv("overshoot.csv")[,-1]
quantile_list <- seq(0.01, 0.99, length.out = 10)
no_sam <- round(seq(1e3, 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()
```

```
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")
```




```
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

```
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      6444     11889     17333     22778     28222     33667
## 0.01  1.5465305 1.1795988 1.1323199 1.1310297 1.1011056 1.1100676 1.1156847
## 0.119 1.7040993 1.1221893 1.1024325 1.0602501 1.0632279 1.0527543 1.0476594
## 0.228 1.2560126 1.0343570 1.0069400 1.0099298 1.0094221 1.0050047 1.0081512
## 0.337 1.5126511 1.1385615 1.0827705 1.0588101 1.0481115 1.0384518 1.0310475
## 0.446 1.4570489 1.0720351 1.0445407 1.0432772 1.0225848 1.0257426 1.0244638
## 0.554 2.3001031 1.0693271 1.0589524 1.0427868 1.0387225 1.0561964 1.0196093
## 0.663 1.5970991 1.1584774 1.1294012 1.1106510 1.0993655 1.0968166 1.1029458
## 0.772 2.2078241 1.0765315 1.0912999 1.0091065 1.0085541 1.0352858 1.0282095
## 0.881 1.5275035 1.0991726 1.1298855 1.1091303 1.0953080 1.1092465 1.1031029
## 0.99  0.4413496 0.2940941 0.3010372 0.2918923 0.2954166 0.2950637 0.2846757
##           39111     44556     50000
## 0.01  1.1052068 1.1042940 1.0914588
## 0.119 1.0431206 1.0601023 1.0341369
## 0.228 1.0061607 1.0065629 1.0047855
## 0.337 1.0376033 1.0237003 1.0214583
## 0.446 1.0111627 1.0054499 1.0160377
## 0.554 1.0448494 1.0275569 1.0206840
## 0.663 1.1104633 1.0827233 1.1030643
## 0.772 1.0000902 1.0120168 1.0303488
## 0.881 1.1040848 1.1024535 1.0902697
## 0.99  0.2846458 0.2878003 0.2849203
```

```
# 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      6444     11889     17333     22778     28222     33667
## 0.01  1.1711130 1.1557135 1.1537702 1.1585431 1.1499282 1.1542931 1.1521705
## 0.119 1.0880128 0.9845718 0.9800489 0.9949159 0.9842294 0.9760974 0.9697382
## 0.228 1.2499173 1.0496203 1.0337018 0.9773649 0.9919001 0.9980693 0.9647877
## 0.337 1.0736902 0.9653932 0.9517578 0.9450618 0.9296693 0.9371788 0.9353515
## 0.446 1.1021360 0.9764753 0.9955806 0.9461586 0.9599375 0.9617454 0.9568762
## 0.554 1.2251337 0.9686648 0.9689706 0.9873614 0.9701593 0.9754466 0.9887871
## 0.663 1.0572414 0.9868336 0.9697912 0.9750857 0.9508886 0.9540896 0.9875902
## 0.772 1.1314665 0.9559408 0.9616882 0.9648478 0.9653205 0.9472507 0.9617242
## 0.881 1.1901252 1.0041668 1.0134750 1.0142705 1.0124605 1.0293514 1.0092123
## 0.99  0.8520581 0.7603266 0.7930170 0.7684031 0.7664030 0.7808087 0.7795389
##           39111     44556     50000
## 0.01  1.1505156 1.1529826 1.1551572
## 0.119 0.9730613 0.9748372 0.9748805
## 0.228 0.9845126 0.9909201 0.9848929
## 0.337 0.9387200 0.9320047 0.9424949
## 0.446 0.9451545 0.9454638 0.9531057
## 0.554 0.9622121 0.9742107 0.9570058
## 0.663 0.9749441 0.9644694 0.9656707
## 0.772 0.9484668 0.9586320 0.9404603
## 0.881 1.0080382 1.0101321 1.0127099
## 0.99  0.7704995 0.7846748 0.7740258
```

We consider 10 cases where we vary quantile of $p^{emp}/p^{fit}(p^{fit}/p^{emp})$. The overshoot_set and undershoot_set repspectively store the index of resampling distribution and the corresponding ratio. Note that for both undershoot and overshoot cases, when we increase the number of resamples, all the quantiles except for

0.99, are stable in terms of the ratio of estimate of p-value ratio to the true p-value ratio immediately after 6444 resamples. For the extreme quantile case 0.99, we should notice the p-value ratio of 327-th resampling distribution reaches the highest (lowest) in the extreme right quantile position. Thus it is hard to capture the extreme ratio unless the number of resamples are close to 5e5. Other than this, I suggest we should use no more than 1e4 resamples.

On the other hand, we should be careful on interpreting the power result. For example, for the extreme quantile case 0.99, it is hard to reach the extreme ratio when resample is less than 1e5 but this does not mean we cannot have a decent estimate for true tail probability. Notice that for both overshoot and undershoot cases, it is even closer towards the true tail probability when using fewer resamples, ranging from 6444 to $5e4$.