06_figure3_small_simulation

randy

2023-10-24

The simulation study with 500 samples and 400 in training set and 100 in testing set

```
## remember to add the / at the end of the location /

## there are two setup for simulation
##  L simulation with 900 individuals ------------------
##  S simulation with 500 individuals ------------------
## with 400 for training set; 100 for testing set
folder1 <- "result_ss500_simulation"

## for each simulation sample size (large or small)
## each has three setup with different anchor time set
## 3 anchor time setting for (6, 9, 12)
## 4 anchor time setting for (3, 6, 9, 12)
## 6 anchor time setting for (3, 6, 9, 12, 13, 15)
# folder2 <- "sim1000_ss500_anchor3time"
# folder2 <- "sim1000_ss500_anchor4time"
folder2 <- "sim1000_ss500_anchor6time"

# file_location <- paste0("results/", folder1, "/", folder2, "/")
# files <- list.files(path = file_location, pattern = ".Rdata")
# ## pull out the simulation results from the folder each as a list of
# ## bias, rmse, coverage50, coverage80, and coverage90
# sim_ss <- map_dfr(files, ~pull_simulation(file_location, .),
#                       .progress = list(type = "iterator",
#                                        format = "Calculating {cli::pb_bar} {cli::pb_percent}",
#                                        clear = TRUE))
# sim_ss <- sim_ss %>%
#   mutate(term = rep(c("bias", "mse", "coverage50", "coverage80", "coverage90"), length(files)))
# save(sim_ss, file = paste0("data/S06_", folder2, "_0summary_", Sys.Date(), ".Rdata"))
```

```
## There will be three Rdata files saved in the data folder
#
# load(paste0("data/S06_", folder2, "_0summary_", Sys.Date(), ".Rdata"))
load("~/Desktop/paper2023/data/S06_sim1000_ss500_anchor3time_0summary_2023-08-31.Rdata")
data <- sim_ss500 %>%
  pivot_longer(cols = -term,
               names_to = "method",
               values_to = "values") %>%
  mutate(method = factor(method, levels = c("sgl_n", "eld_n", "mhl_n",
                                            "mhl_p75", "mhl_p80", "mhl_p85",
```

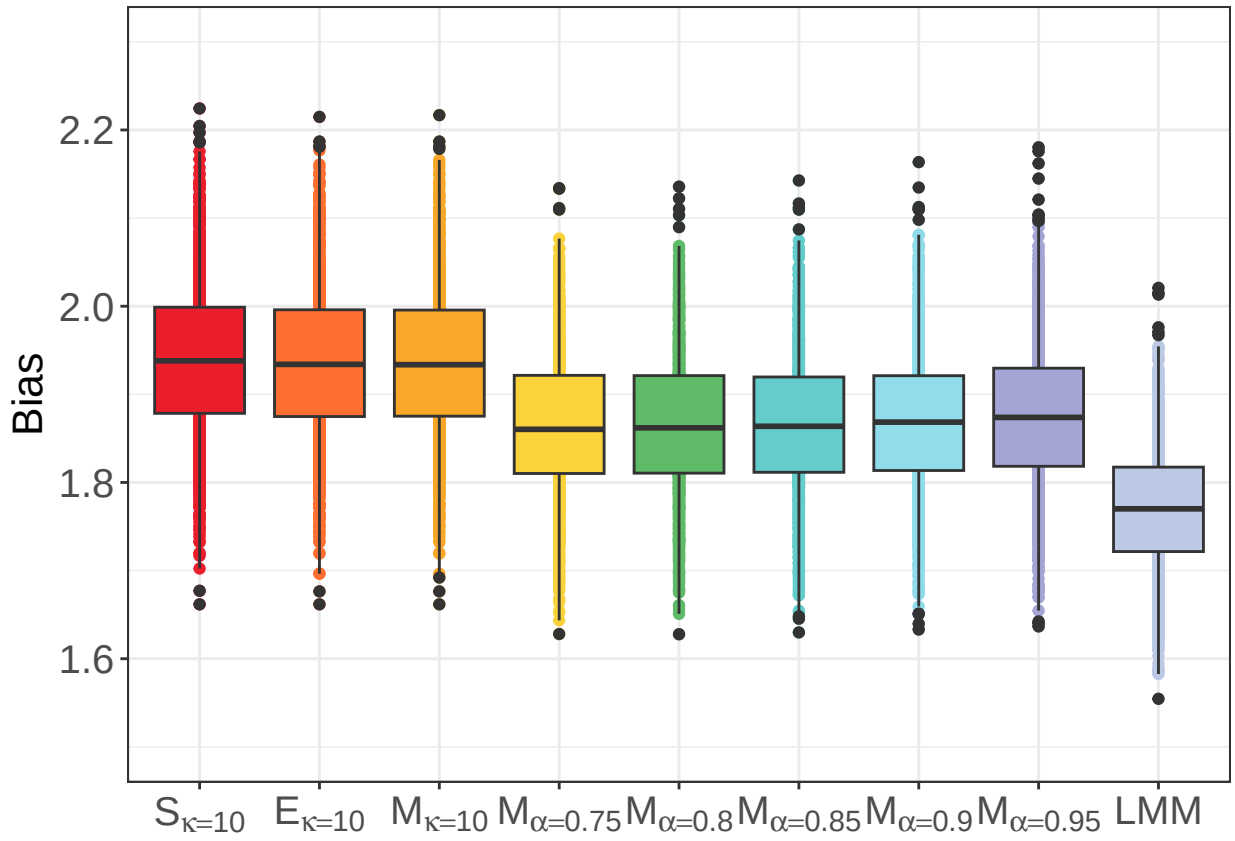
```
                                          "mhl_p90", "mhl_p95", "lmm")))

library(latex2exp)
bias_plot <- data %>%
  filter(term == "bias") %>%
  ggplot(aes(x = method, y = as.numeric(values), group = factor(method))) +
  scale_x_discrete(labels = c("sgl_n" = parse(text = latex2exp::TeX("$S_{\\kappa = 10}$")),
                              "eld_n" = parse(text = latex2exp::TeX("$E_{\\kappa = 10}$")),
                              "mhl_n" = parse(text = latex2exp::TeX("$M_{\\kappa = 10}$")),
                              "mhl_p75" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.75}$")),
                              "mhl_p80" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.80}$")),
                              "mhl_p85" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.85}$")),
                              "mhl_p90" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.90}$")),
                              "mhl_p95" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.95}$")),
                              "lmm" = "LMM")) +
  geom_point(aes(color = method)) +
  geom_boxplot(aes(fill = method)) +
  theme_bw() +
  labs(x = NULL) +
  labs(y = "Bias") +
  ggthemes::scale_fill_tableau("Jewel Bright") +
  ## so far the best color composition
  ggthemes::scale_colour_tableau("Jewel Bright") +
  ## facet_wrap("group") +
  theme(legend.position = "none") +

  theme(axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        axis.title.y = element_text(size = 16)) +
  ylim(c(1.5, 2.3))
bias_plot
```

```
## Warning: Removed 5 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: Removed 5 rows containing missing values (`geom_point()`).
```

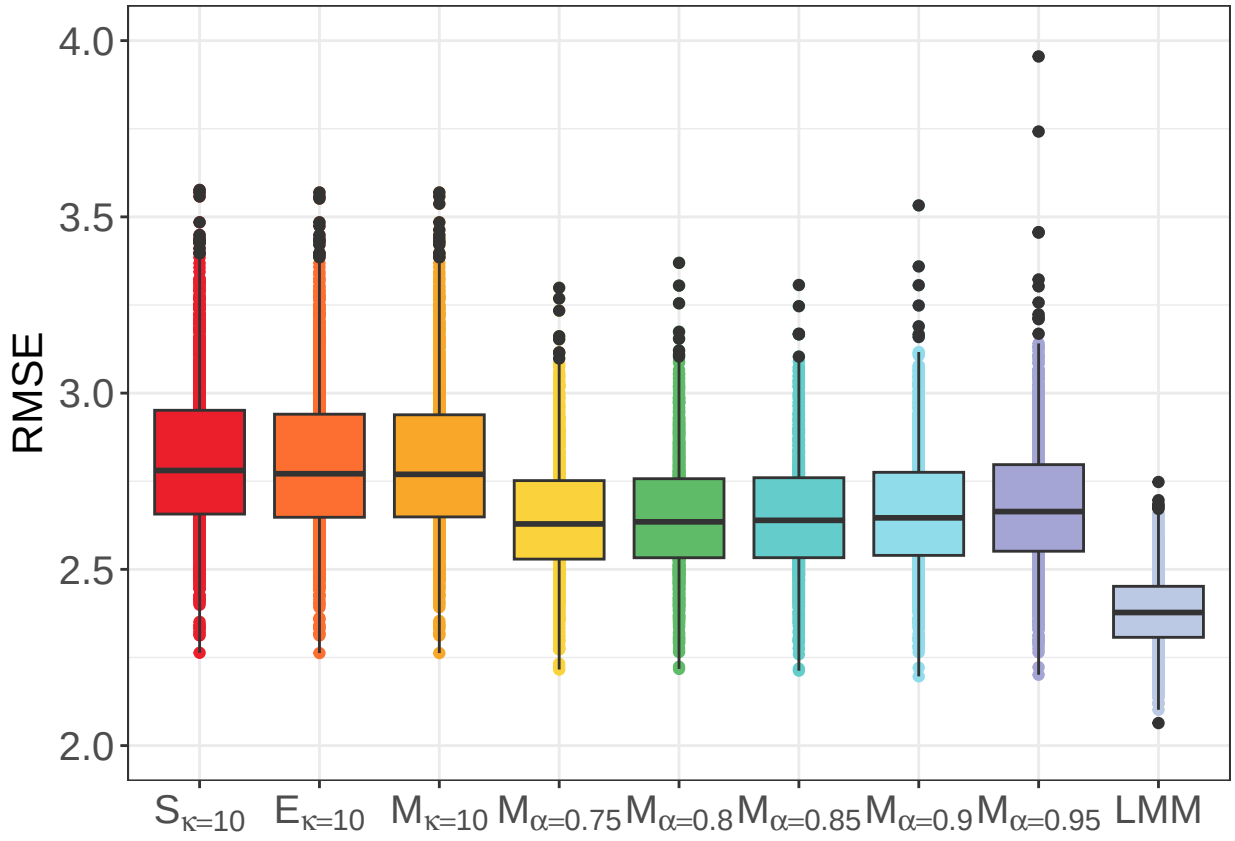



```
rmse_plot <- data %>%
  filter(term == "mse") %>%
  mutate(values = sqrt(as.numeric(values))) %>%
  ggplot(aes(x = method, y = as.numeric(values), group = method)) +
  geom_point(aes(color = method)) +
  geom_boxplot(aes(fill = method)) +
  theme_bw() +
  labs(x = NULL) +
  labs(y = "RMSE") +
  ggthemes::scale_fill_tableau("Jewel Bright") +
  ## so far the best color composition
  ggthemes::scale_colour_tableau("Jewel Bright") +
  ## facet_wrap("group") +
  theme(legend.position = "none") +
  scale_x_discrete(labels = c("sgl_n" = parse(text = latex2exp::TeX("$S_{\\kappa = 10}$")),
                              "eld_n" = parse(text = latex2exp::TeX("$E_{\\kappa = 10}$")),
                              "mhl_n" = parse(text = latex2exp::TeX("$M_{\\kappa = 10}$")),
                              "mhl_p75" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.75}$")),
                              "mhl_p80" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.80}$")),
                              "mhl_p85" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.85}$")),
                              "mhl_p90" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.90}$")),
                              "mhl_p95" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.95}$")),
                              "lmm" = "LMM")) +
  theme(axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        axis.title.y = element_text(size = 16)) +
```

```
  ylim(c(2, 4))
rmse_plot
```

```
## Warning: Removed 10 rows containing non-finite values (`stat_boxplot()`).
```

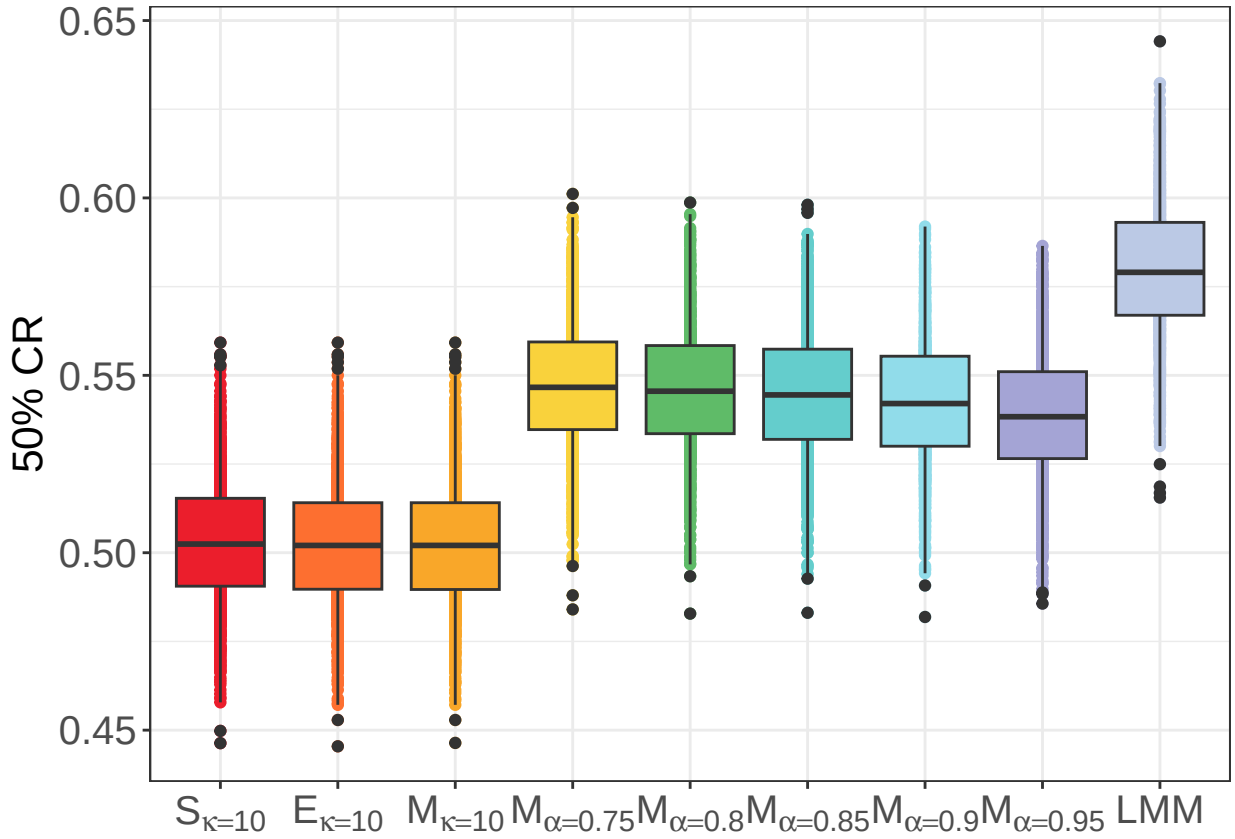
```
## Warning: Removed 10 rows containing missing values (`geom_point()`).
```




```
cov50_plot <- data %>%
  filter(term == "coverage50") %>%
  ggplot(aes(x = method, y = as.numeric(values)),
         group = method) +
  geom_point(aes(color = method)) +
  geom_boxplot(aes(fill = method)) +
  theme_bw() +
  labs(x = NULL) +
  labs(y = "50% CR") +
  ggthemes::scale_fill_tableau("Jewel Bright") +
  ## so far the best color composition
  ggthemes::scale_colour_tableau("Jewel Bright") +
  ## facet_wrap("group") +
  theme(legend.position = "none") +
  scale_x_discrete(labels = c("sgl_n" = parse(text = latex2exp::TeX("$S_{\\kappa = 10}$")),
                              "eld_n" = parse(text = latex2exp::TeX("$E_{\\kappa = 10}$")),
                              "mhl_n" = parse(text = latex2exp::TeX("$M_{\\kappa = 10}$")),
```

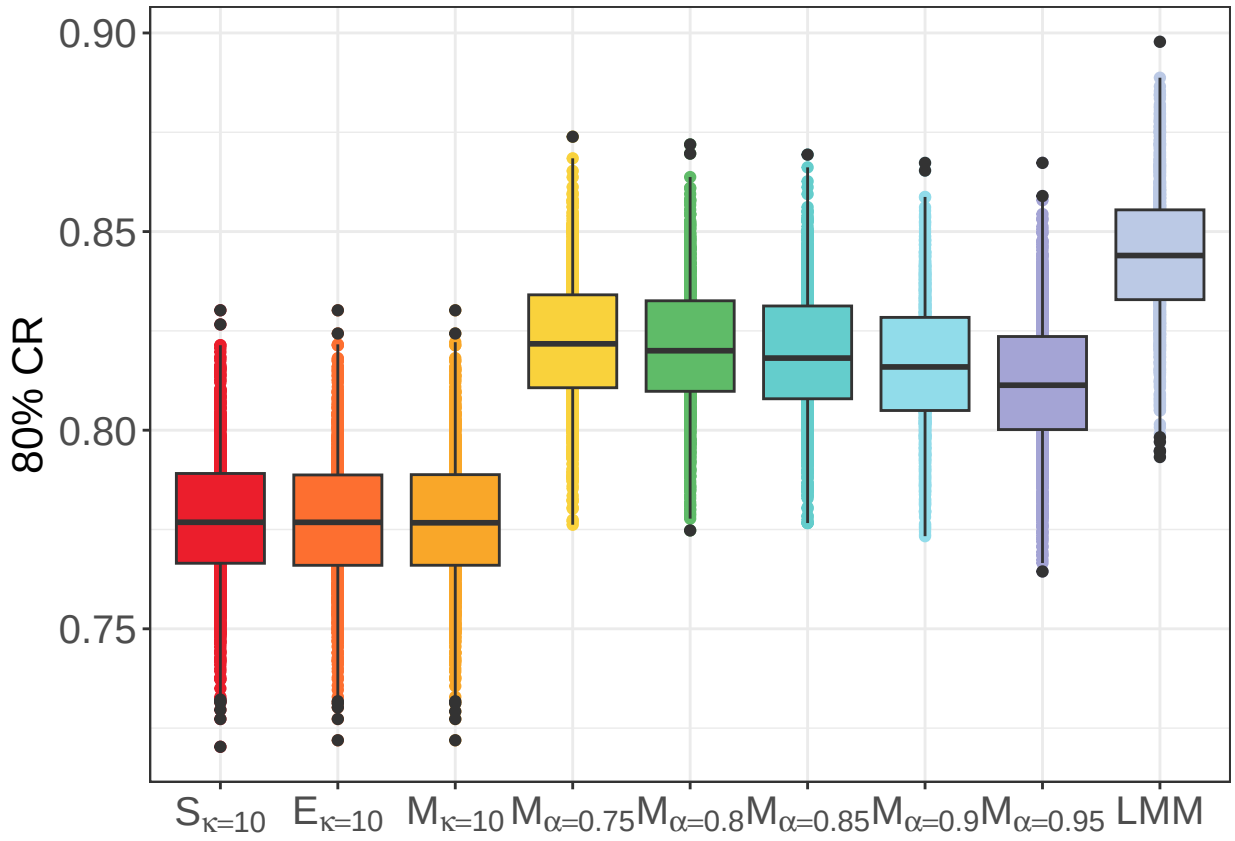
```
                              "mhl_p75" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.75}$")),
                              "mhl_p80" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.80}$")),
                              "mhl_p85" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.85}$")),
                              "mhl_p90" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.90}$")),
                              "mhl_p95" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.95}$")),
                              "lmm" = "LMM")) +
  theme(axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        axis.title.y = element_text(size = 16))

## the tableau is in the ggthemes
cov50_plot
```



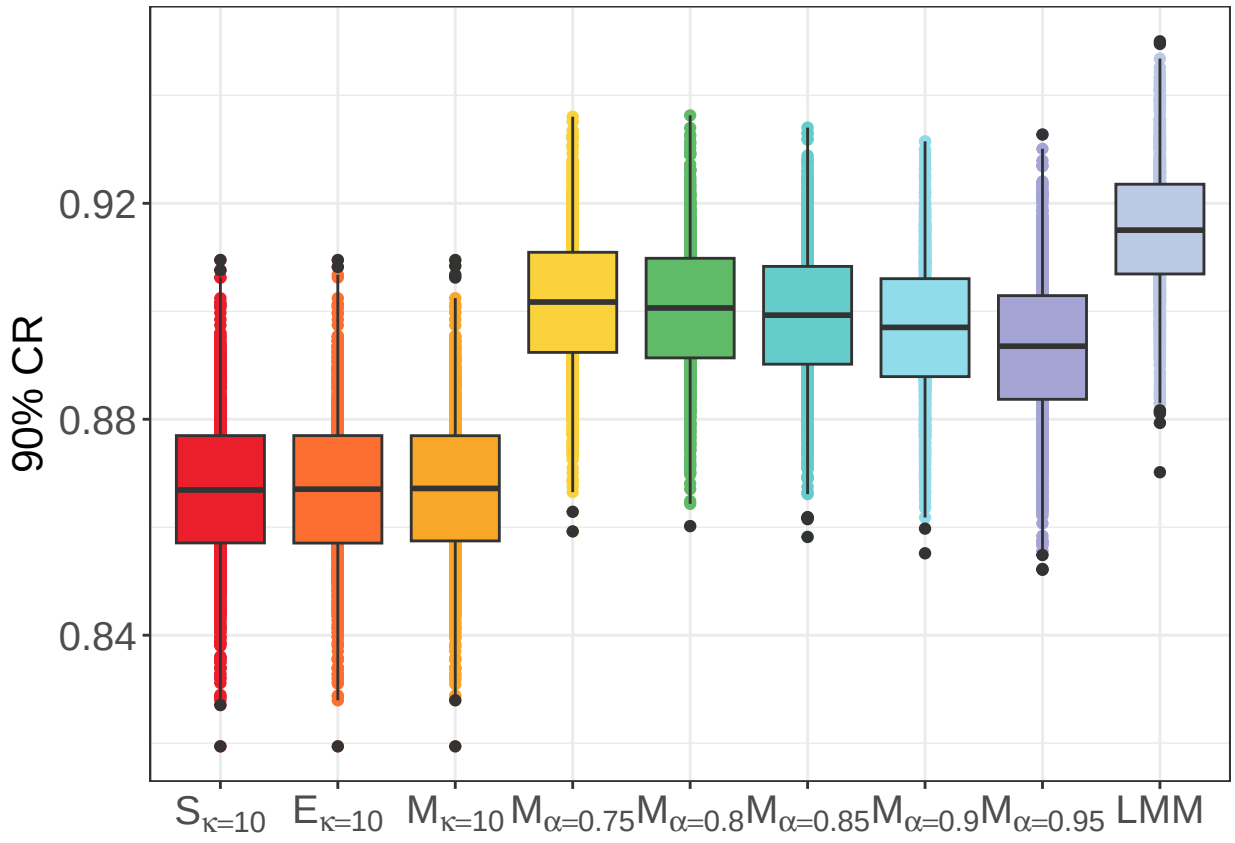

```
cov80_plot <- data %>%
          filter(term == "coverage80") %>%
          ggplot(aes(x = method, y = as.numeric(values)),
                 group = method) +
          geom_point(aes(color = method)) +
          geom_boxplot(aes(fill = method)) +
          theme_bw() +
         labs(x = NULL) +
          labs(y = "80% CR") +
          ggthemes::scale_fill_tableau("Jewel Bright") +
          ## so far the best color composition
          ggthemes::scale_colour_tableau("Jewel Bright") +
```

```
          ## facet_wrap("group") +
          theme(legend.position = "none") +
  scale_x_discrete(labels = c("sgl_n" = parse(text = latex2exp::TeX("$S_{\\kappa = 10}$")),
                              "eld_n" = parse(text = latex2exp::TeX("$E_{\\kappa = 10}$")),
                              "mhl_n" = parse(text = latex2exp::TeX("$M_{\\kappa = 10}$")),
                              "mhl_p75" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.75}$")),
                              "mhl_p80" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.80}$")),
                              "mhl_p85" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.85}$")),
                              "mhl_p90" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.90}$")),
                              "mhl_p95" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.95}$")),
                              "lmm" = "LMM")) +
  theme(axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        axis.title.y = element_text(size = 16))
          ## the tableau is in the ggthemes
cov80_plot
```

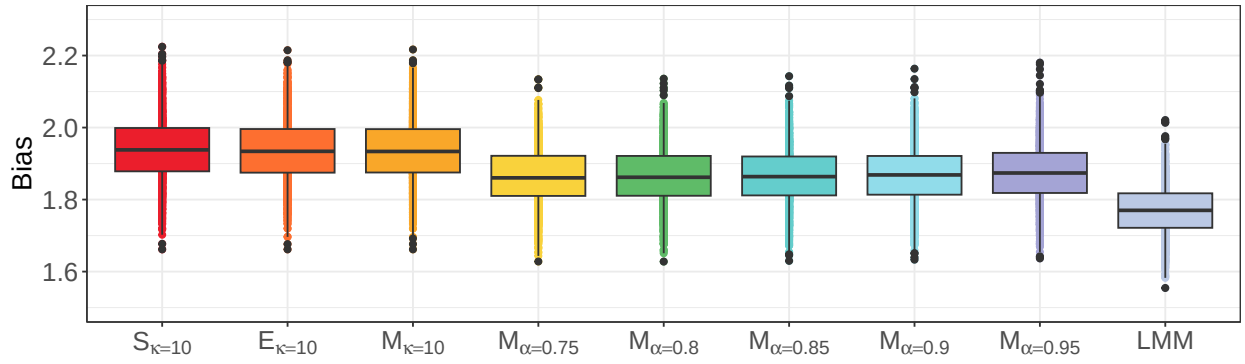



```
cov90_plot <- data %>%
          filter(term == "coverage90") %>%
          ggplot(aes(x = method, y = as.numeric(values)),
                 group = method) +
          geom_point(aes(color = method)) +
          geom_boxplot(aes(fill = method)) +
          theme_bw() +
          labs(x = NULL) +
```

```
          labs(y = "90% CR") +
          ggthemes::scale_fill_tableau("Jewel Bright") +
          ## so far the best color composition
          ggthemes::scale_colour_tableau("Jewel Bright") +
          ## facet_wrap("group") +
          theme(legend.position = "none") +
  scale_x_discrete(labels = c("sgl_n" = parse(text = latex2exp::TeX("$S_{\\kappa = 10}$")),
                              "eld_n" = parse(text = latex2exp::TeX("$E_{\\kappa = 10}$")),
                              "mhl_n" = parse(text = latex2exp::TeX("$M_{\\kappa = 10}$")),
                              "mhl_p75" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.75}$")),
                              "mhl_p80" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.80}$")),
                              "mhl_p85" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.85}$")),
                              "mhl_p90" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.90}$")),
                              "mhl_p95" = parse(text = latex2exp::TeX("$M_{\\alpha = 0.95}$")),
                              "lmm" = "LMM")) +
  theme(axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        axis.title.y = element_text(size = 16))
          ## the tableau is in the ggthemes
cov90_plot
```



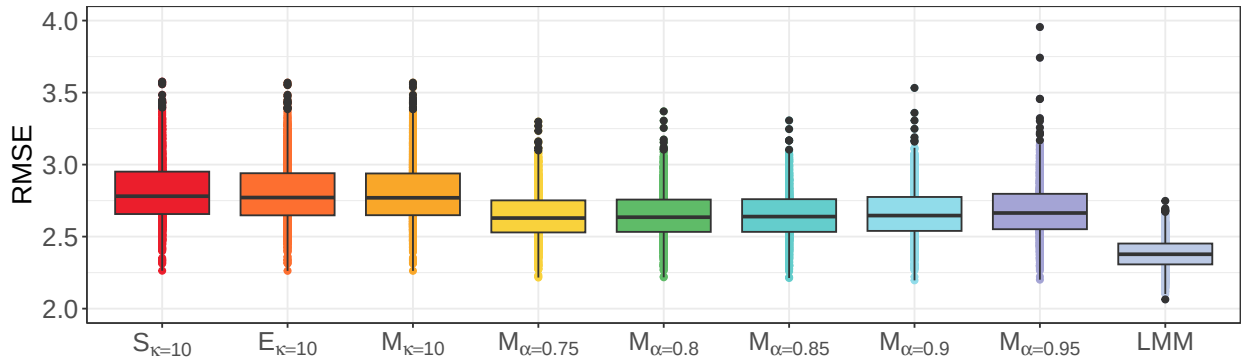
```
bias_plot
```




```
ggsave(paste0("figure/S06_", folder2, "_1bias_", Sys.Date(), ".png"))
```

```
## Saving 10 x 3 in image
```

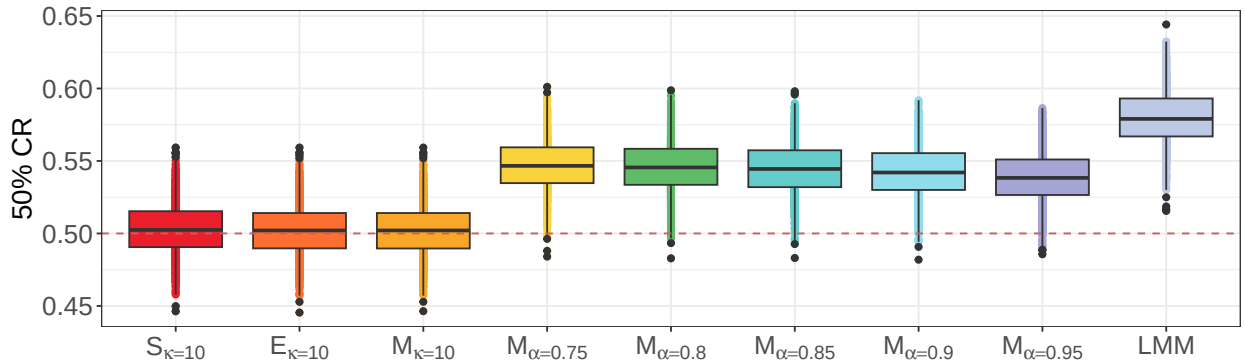
```
rmse_plot
```




```
ggsave(paste0("figure/S06_", folder2, "_2rmse_", Sys.Date(), ".png"))
```

```
## Saving 10 x 3 in image
```

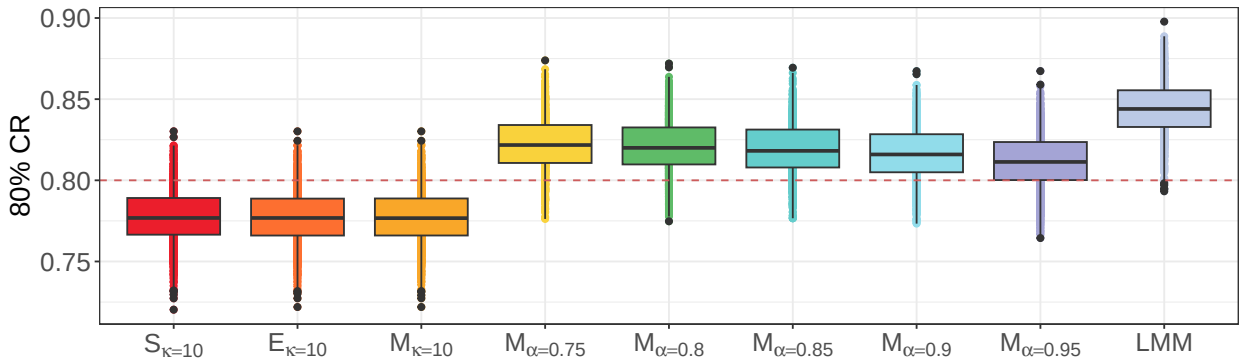
```
cov50_plot +
  geom_hline(yintercept = 0.5, linetype="dashed", color = "indianred")
```



```
ggsave(paste0("figure/S06_", folder2, "_3cov50_", Sys.Date(), ".png"))
```

```
## Saving 10 x 3 in image
```

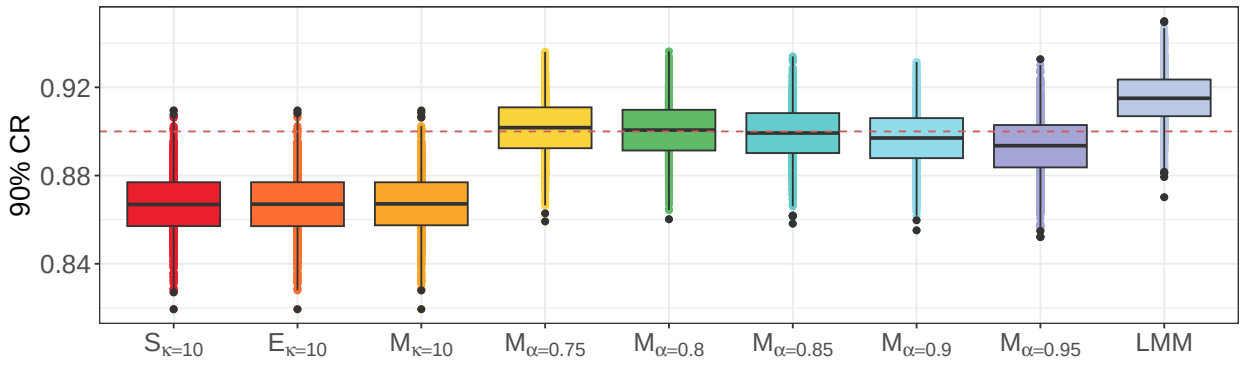
```
cov80_plot +
  geom_hline(yintercept = 0.8, linetype="dashed", color = "indianred")
```




```
ggsave(paste0("figure/S06_", folder2, "_4cov80_", Sys.Date(), ".png"))
```

```
## Saving 10 x 3 in image
```

```
cov90_plot +
  geom_hline(yintercept = 0.9, linetype="dashed", color = "indianred")
```




```
ggsave(paste0("figure/S06_", folder2, "_5cov90_", Sys.Date(), ".png"))
```

```
## Saving 10 x 3 in image
```

```
library(ggpubr)
```

```
##
## Attaching package: 'ggpubr'

## The following objects are masked from 'package:flextable':
##
##     border, font, rotate
```

```r
figure1 <- ggarrange(bias_plot,
                     # + theme(axis.text.x=element_blank()),
                     rmse_plot,
                     # + theme(axis.text.x=element_blank()),
                     # cov50_plot,
                     # + theme(axis.text.x=element_blank()),
                     cov90_plot +
                       geom_hline(yintercept = 0.9,
                                  linetype = "dashed",
                                  color = "indianred"),
                     # + theme(axis.text.x=element_blank()),
                     ncol = 1, nrow = 3)
```

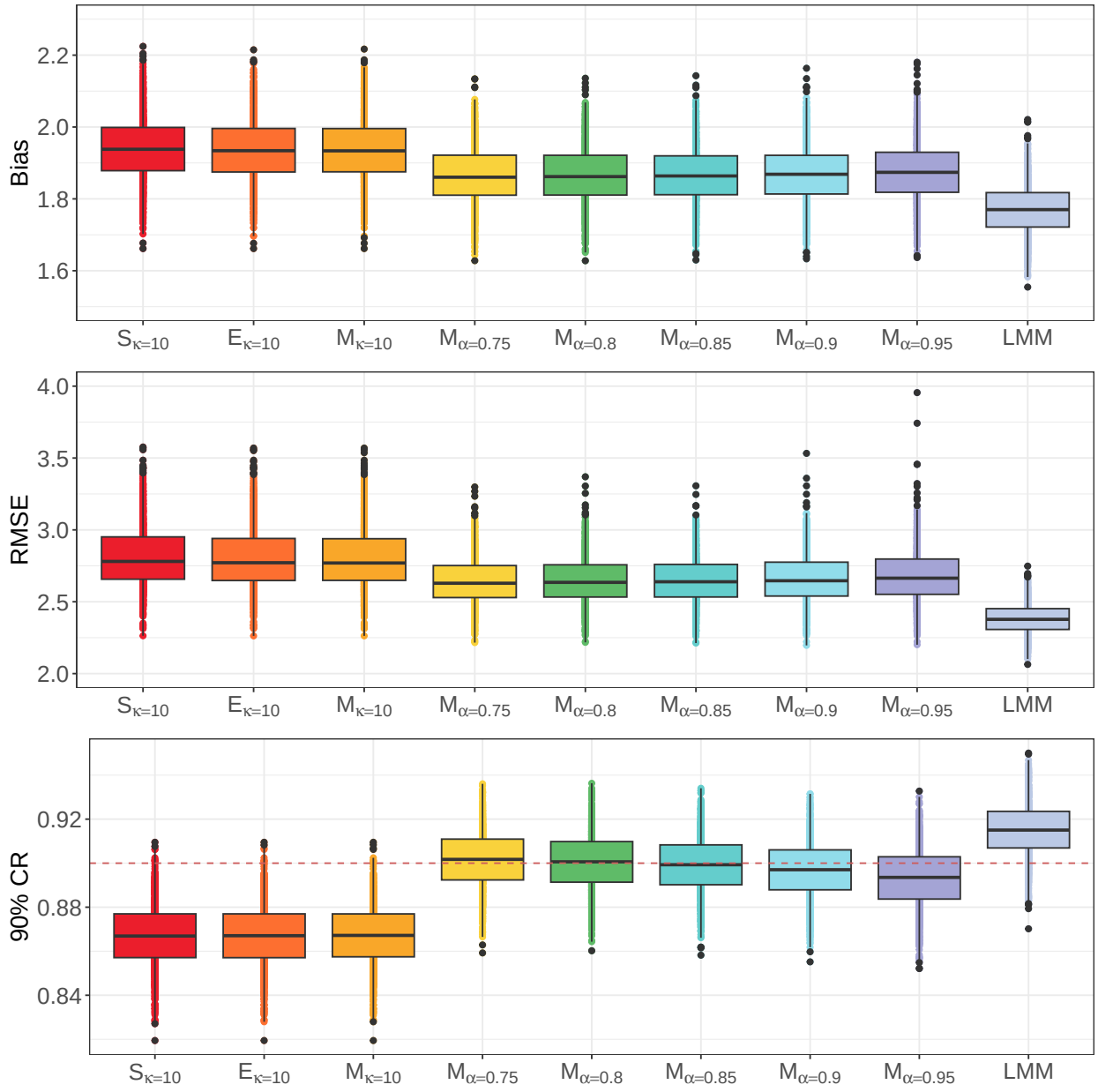
```
## Warning: Removed 5 rows containing non-finite values (`stat_boxplot()`).

## Warning: Removed 5 rows containing missing values (`geom_point()`).

## Warning: Removed 10 rows containing non-finite values (`stat_boxplot()`).

## Warning: Removed 10 rows containing missing values (`geom_point()`).
```

```r
figure1
```




```
ggsave(paste0("figure/S06_", folder2, "_6combine_", Sys.Date(), ".png"))
```

```
## Saving 10 x 10 in image
```

```
figure2 <- ggarrange(bias_plot,
                     # + theme(axis.text.x=element_blank()),
                     rmse_plot,
                     # + theme(axis.text.x=element_blank()),
                     cov50_plot +
                       geom_hline(yintercept = 0.5,
                                  linetype = "dashed",
                                  color = "indianred"),
                     cov80_plot +
```

```
                        geom_hline(yintercept = 0.8,
                                   linetype = "dashed",
                                   color = "indianred"),
                      # + theme(axis.text.x=element_blank()),
                      cov90_plot +
                        geom_hline(yintercept = 0.9,
                                   linetype = "dashed",
                                   color = "indianred"),
                      # + theme(axis.text.x=element_blank()),
                      ncol = 1, nrow = 5)
```

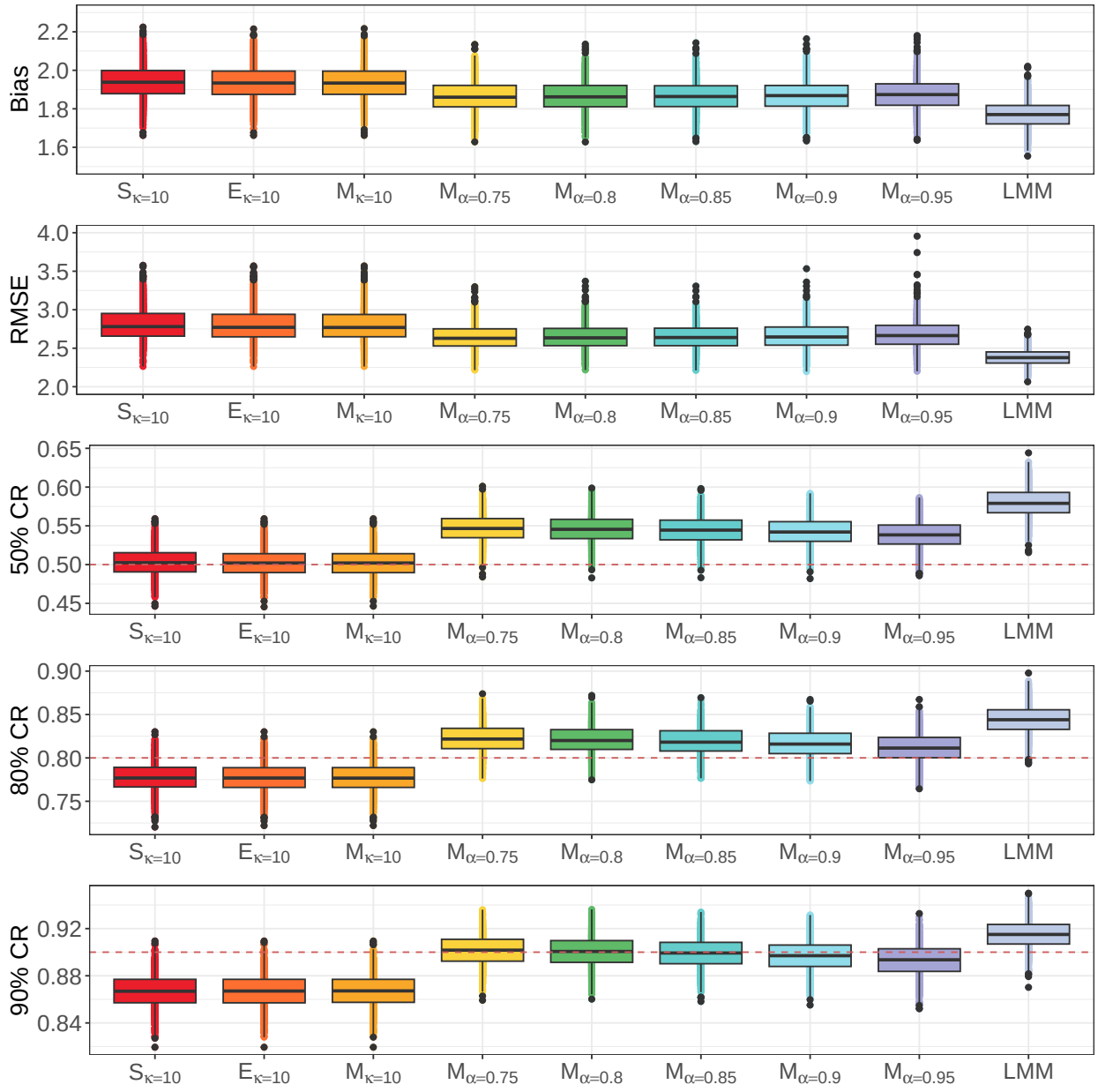
```
## Warning: Removed 5 rows containing non-finite values (`stat_boxplot()`).

## Warning: Removed 5 rows containing missing values (`geom_point()`).

## Warning: Removed 10 rows containing non-finite values (`stat_boxplot()`).

## Warning: Removed 10 rows containing missing values (`geom_point()`).
```

```
figure2
```



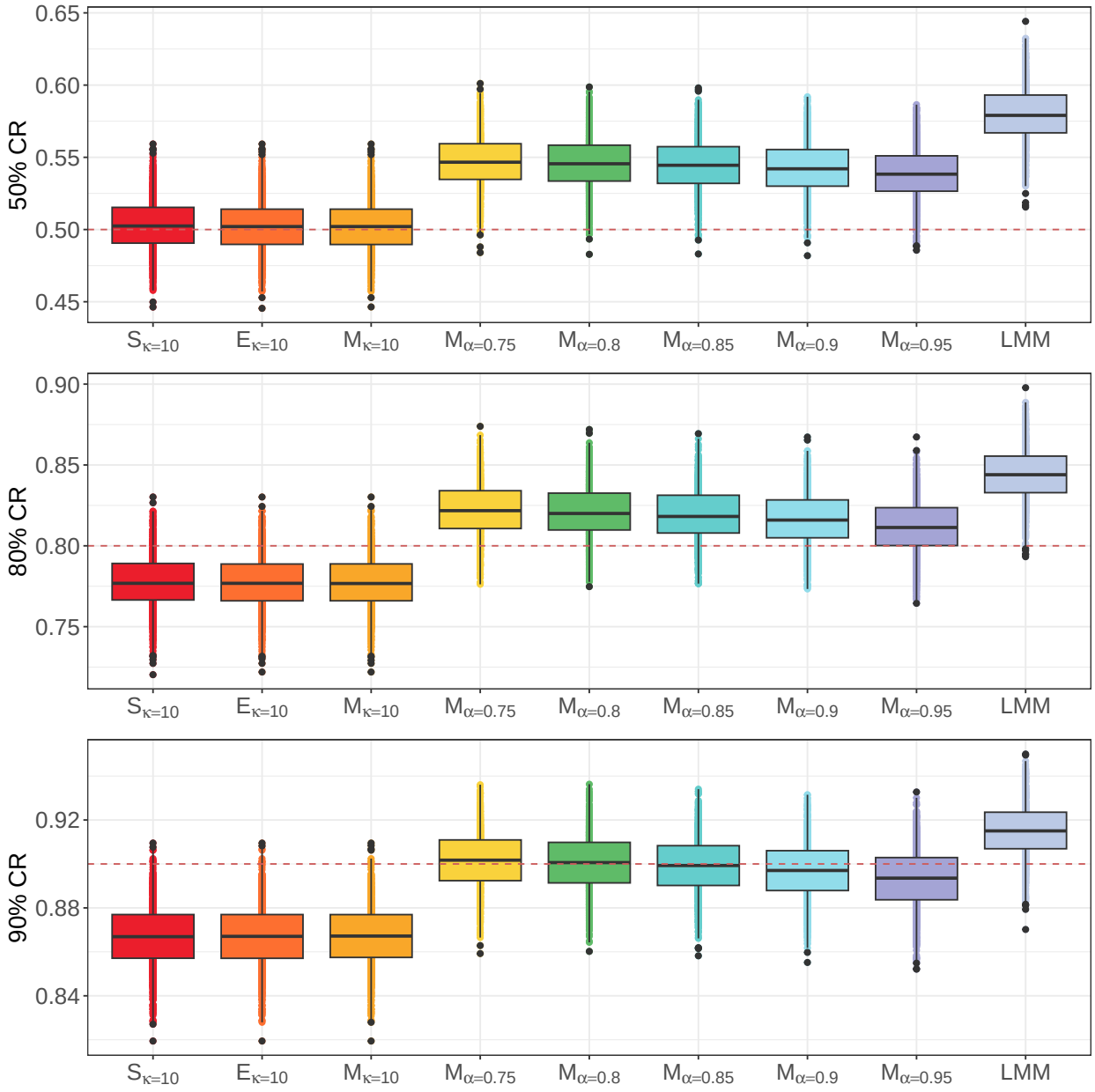

```
ggsave(paste0("figure/S06_", folder2, "_7all_", Sys.Date(), ".png"))
```

```
## Saving 10 x 10 in image
```

```
figure3 <- ggarrange(#bias_plot,
                     # + theme(axis.text.x=element_blank()),
                     #rmse_plot,
                     # + theme(axis.text.x=element_blank()),
                     cov50_plot +
                       geom_hline(yintercept = 0.5,
                                  linetype = "dashed",
                                  color = "indianred"),
                     cov80_plot +
```

```
                    geom_hline(yintercept = 0.8,
                               linetype = "dashed",
                               color = "indianred"),
                  # + theme(axis.text.x=element_blank()),
                  cov90_plot +
                    geom_hline(yintercept = 0.9,
                               linetype = "dashed",
                               color = "indianred"),
                  # + theme(axis.text.x=element_blank()),
                  ncol = 1, nrow = 3)
```

figure3



```r
ggsave(paste0("figure/S06_", folder2, "_8cr_", Sys.Date(), ".png"))
```

```
## Saving 10 x 10 in image
```

```r
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS 14.0
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ggpubr_0.6.0    latex2exp_0.9.6 flextable_0.9.2 gtsummary_1.7.1
##  [5] lubridate_1.9.2 forcats_1.0.0   stringr_1.5.0   dplyr_1.1.2
##  [9] purrr_1.0.1     readr_2.1.4     tidyr_1.3.0     tibble_3.2.1
## [13] ggplot2_3.4.3   tidyverse_2.0.0 here_1.0.1
##
## loaded via a namespace (and not attached):
##  [1] fontquiver_0.2.1        rprojroot_2.0.3         tools_4.2.2
##  [4] backports_1.4.1         utf8_1.2.3              R6_2.5.1
##  [7] colorspace_2.1-0        withr_2.5.0             tidyselect_1.2.0
## [10] curl_5.0.1              compiler_4.2.2          textshaping_0.3.6
## [13] cli_3.6.1               gt_0.9.0                xml2_1.3.5
## [16] officer_0.6.2           fontBitstreamVera_0.1.1 labeling_0.4.2
## [19] scales_1.2.1            askpass_1.1             systemfonts_1.0.4
## [22] digest_0.6.33           rmarkdown_2.23          gfonts_0.2.0
## [25] pkgconfig_2.0.3         htmltools_0.5.5         fastmap_1.1.1
## [28] highr_0.10              rlang_1.1.1             ggthemes_4.2.4
## [31] rstudioapi_0.15.0       httpcode_0.3.0          shiny_1.7.4.1
## [34] farver_2.1.1            generics_0.1.3          jsonlite_1.8.7
## [37] zip_2.3.0               car_3.1-2               magrittr_2.0.3
## [40] Rcpp_1.0.11             munsell_0.5.0           fansi_1.0.4
## [43] abind_1.4-5             gdtools_0.3.3           lifecycle_1.0.3
## [46] stringi_1.7.12          yaml_2.3.7              carData_3.0-5
## [49] grid_4.2.2              promises_1.2.0.1        crayon_1.5.2
## [52] cowplot_1.1.1           hms_1.1.3               knitr_1.43
## [55] pillar_1.9.0            uuid_1.1-0              ggsignif_0.6.4
## [58] crul_1.4.0              glue_1.6.2              freshr_1.0.2
## [61] evaluate_0.21           fontLiberation_0.1.0    data.table_1.14.8
## [64] broom.helpers_1.13.0    vctrs_0.6.3             tzdb_0.4.0
## [67] httpuv_1.6.11           gtable_0.3.3            openssl_2.1.0
## [70] xfun_0.39               mime_0.12               xtable_1.8-4
## [73] broom_1.0.5             rstatix_0.7.2           later_1.3.1
## [76] ragg_1.2.5              timechange_0.2.0        ellipsis_0.3.2
```