Preliminary analysis

Samuel Pawel, Björn Siepe, František Bartoš

11 August 2023

Visualizations

```
## libraries
library(dplyr)
```

```
##
## Attache Paket: 'dplyr'
```

```
## Die folgenden Objekte sind maskiert von 'package:stats':
##
##     filter, lag
```

```
## Die folgenden Objekte sind maskiert von 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
library(colorspace)
library(ggpubr)
library(stringr)
library(forcats)
library(knitr)
library(kableExtra)
```

```
##
## Attache Paket: 'kableExtra'
```

```
## Das folgende Objekt ist maskiert 'package:dplyr':
##
##     group_rows
```

```
library(sysfonts)
library(showtext)
```

```
## Lade nötiges Paket: showtextdb
```

```r
library(here)
```

```
## here() starts at C:/Users/Bjoern/nondrive-academia/Projects/SimPsychReview
```

```r
library(Hmisc)
```

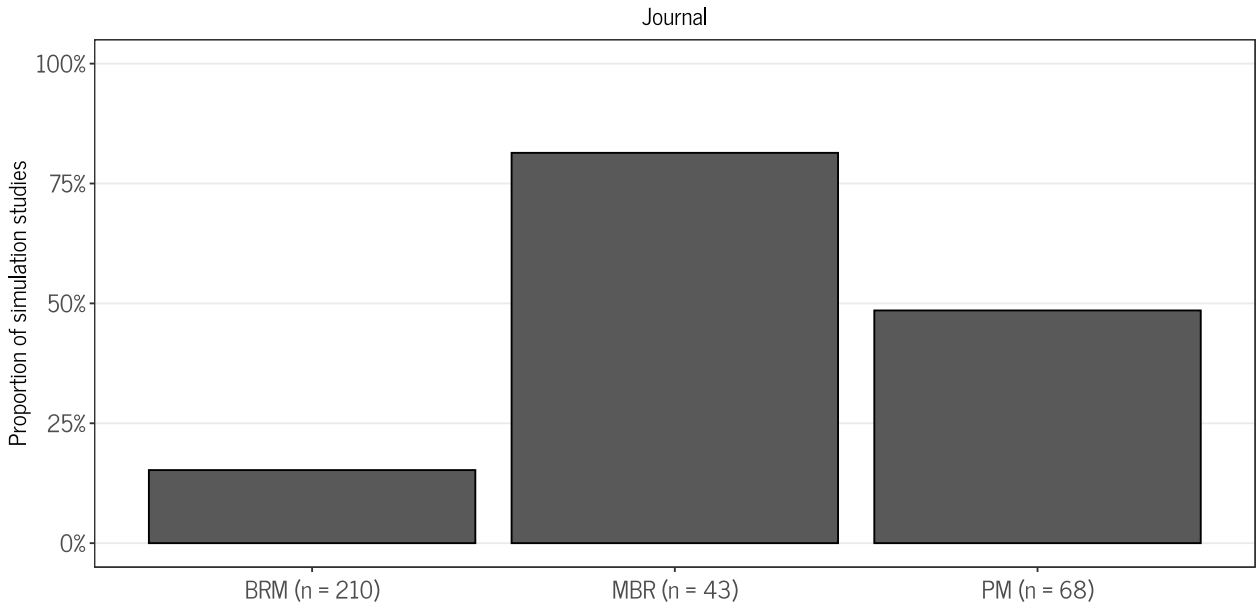
```
##
## Attache Paket: 'Hmisc'
```

```
## Die folgenden Objekte sind maskiert von 'package:dplyr':
##
##     src, summarize
```

```
## Die folgenden Objekte sind maskiert von 'package:base':
##
##     format.pval, units
```

```r
# devtools::install_github("kupietz/kableExtra")

theme_set(theme_bw() +
          theme(legend.position = "top",
                panel.grid.minor = element_blank()))
## pal <- "Harmonic" # change palette here
## ## colorspace::hcl_palettes("qualitative", plot = TRUE)
cols <- c("BRM" = "#E69F00", "MBR" = "#009E73", "PM" = "#0072B2")
# cols <- c("BRM" = "#E69F00", "MBR" = "#56B4E9", "PM" = "#009E73")

# Alternative font
theme_bs <- function(){
  # add google font
  sysfonts::font_add_google("News Cycle", "news")
  # use showtext
  showtext::showtext_auto()
  # theme
  ggplot2::theme_bw(base_family = "news") +
  ggplot2::theme(
    legend.position = "top",
    panel.grid.minor = element_blank(),

    # Title and Axis Texts
    plot.title = ggplot2::element_text(size = ggplot2::rel(1.2), hjust = 0.5),
    plot.subtitle = ggplot2::element_text(size = ggplot2::rel(1.1), hjust = 0.5),
    axis.title = ggplot2::element_text(size = ggplot2::rel(1.2)),
    axis.text = ggplot2::element_text(size = ggplot2::rel(1.25)),
    axis.text.x = ggplot2::element_text(margin = ggplot2::margin(5, b = 10))
  )
}
theme_set(theme_bs())


## data
```

```r
sim_res_fac_full <- readRDS(file = here("data/sim_res_fac.RDS"))
sim_res_num_full <- readRDS(file = here("data/sim_res_num.RDS"))

# subset assessment only
sim_res_fac  <- sim_res_fac_full %>%
    filter(simstudy_q1 == "yes",
           coding_type == "assessment")
sim_res_num  <- sim_res_num_full %>%
    filter(simstudy_q1 == "yes",
           coding_type == "assessment")

## proportion of simulation studies by journal
sim_res_fac_full %>%
    filter(coding_type == "assessment" | is.na(coding_type)) %>%
    group_by(journal) %>%
    dplyr::summarize(propSim = mean(simstudy_q1 == "yes"),
              n = n()) %>%
    mutate(journalLab = paste0(journal, " (n = ", n, ")")) %>%
    ggplot(aes(x = journalLab, y = propSim)) +
    geom_bar(stat = "identity", col = 1) +
    scale_y_continuous(labels = scales::percent, limits = c(0, 1)) +
    labs(x = NULL, title ="Journal", y = "Proportion of simulation studies") +
    theme(panel.grid.major.x = element_blank())
```




```r
sim_res_fac_full %>%
    filter(coding_type == "assessment" | is.na(coding_type),
           year == 2022) %>%
    dplyr::summarize(propSim = mean(simstudy_q1 == "yes"),
              sim = sum(simstudy_q1 == "yes"),
              n = n())
```

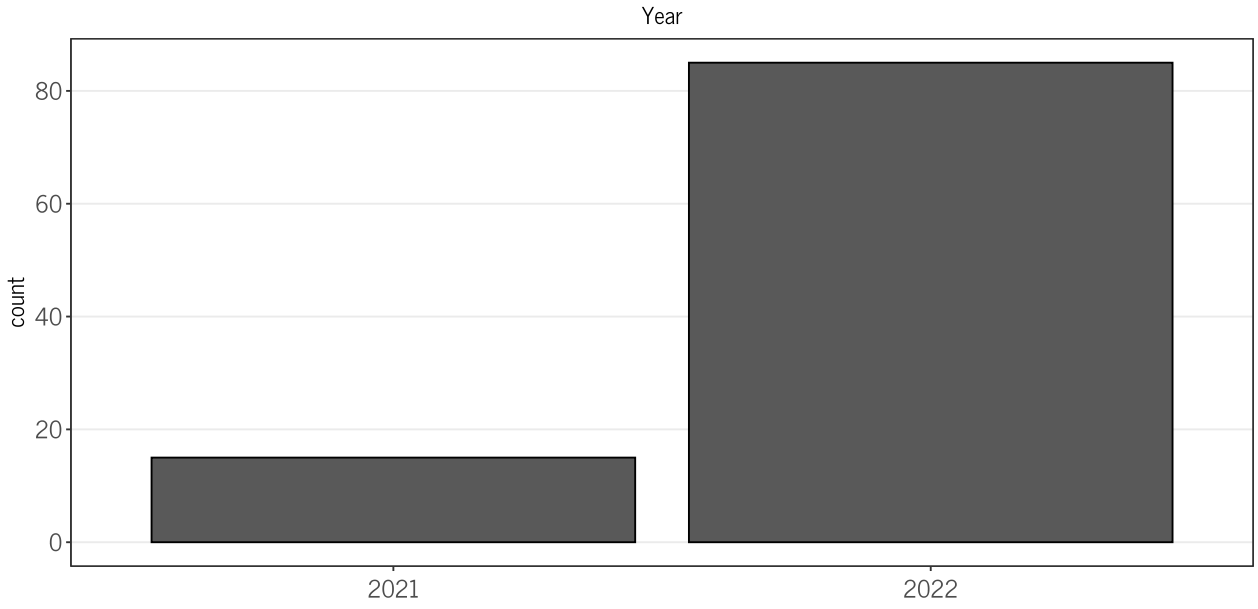
```
## # A tibble: 1 x 3
##   propSim   sim     n
```

```
##     <dbl> <int> <int>
## 1   0.343    85   248
```

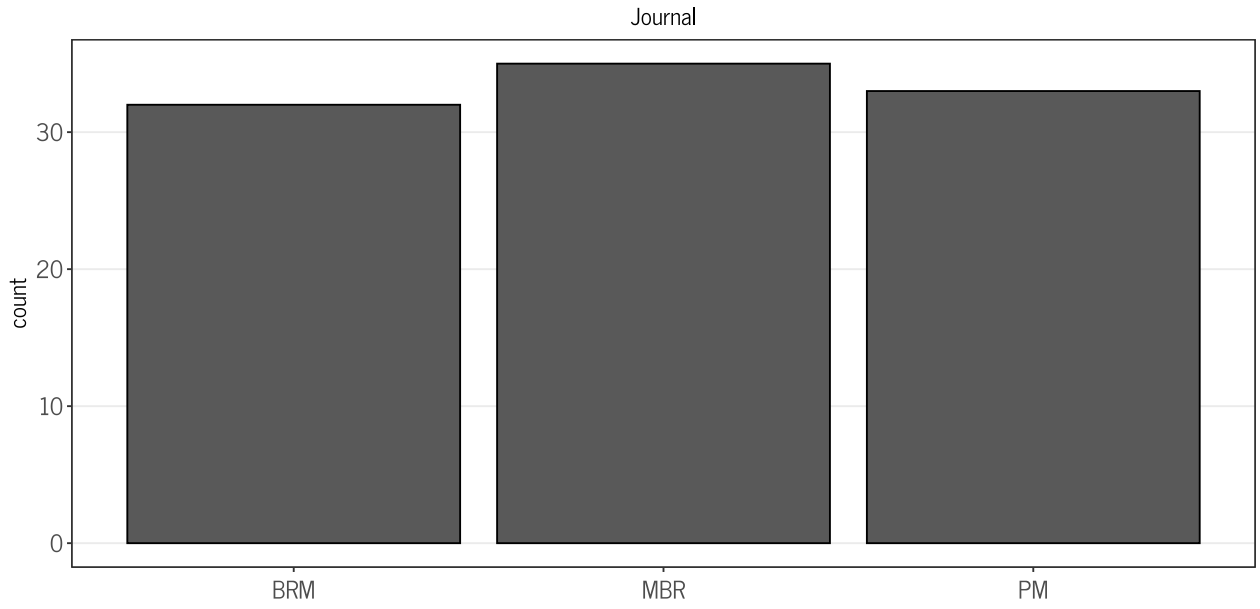
```
sim_res_fac_full %>%
    filter(coding_type == "assessment" | is.na(coding_type),
           year == 2022) %>%
    group_by(journal) %>%
    dplyr::summarize(propSim = mean(simstudy_q1 == "yes"),
              sim = sum(simstudy_q1 == "yes"),
              n = n())
```

```
## # A tibble: 3 x 4
##   journal propSim   sim     n
##   <fct>     <dbl> <int> <int>
## 1 BRM       0.156    24   154
## 2 MBR       0.814    35    43
## 3 PM        0.510    26    51
```

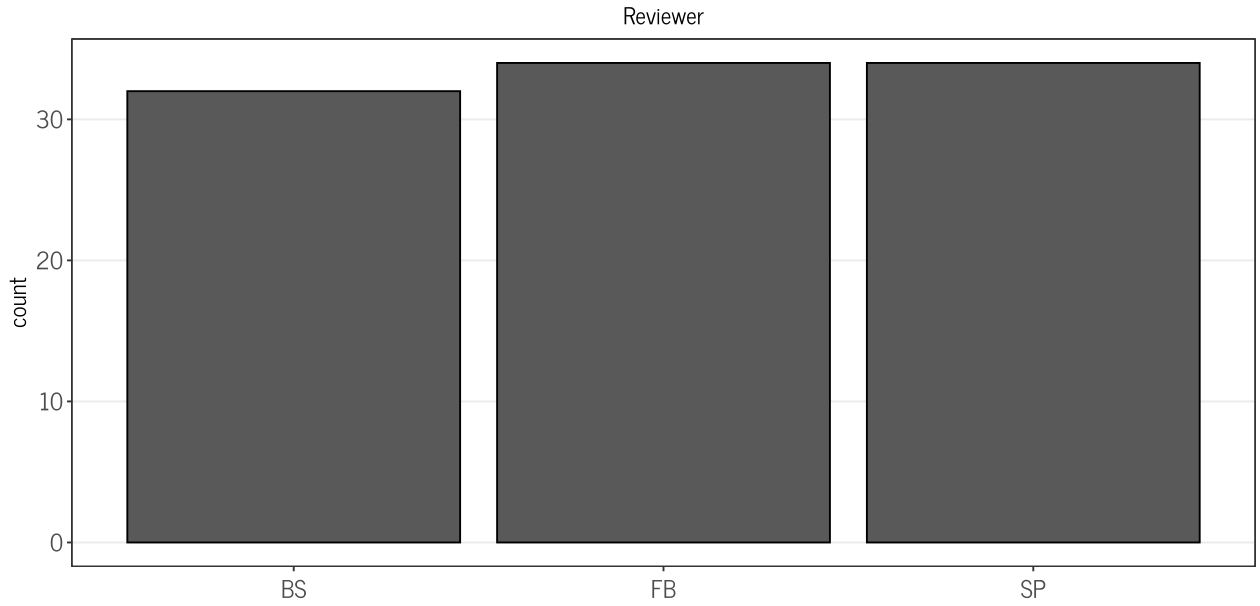
```
## Morris et al. (2019) find
## "264 articles of which 199 (75\%) included at least one simulation study"


## year
ggplot(data = sim_res_fac, aes(x = factor(year))) +
    geom_bar(col = 1) +
    labs(x = NULL, title ="Year") +
    theme(panel.grid.major.x = element_blank())
```




```
## journal
ggplot(data = sim_res_fac, aes(x = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, title ="Journal") +
    theme(panel.grid.major.x = element_blank())
```

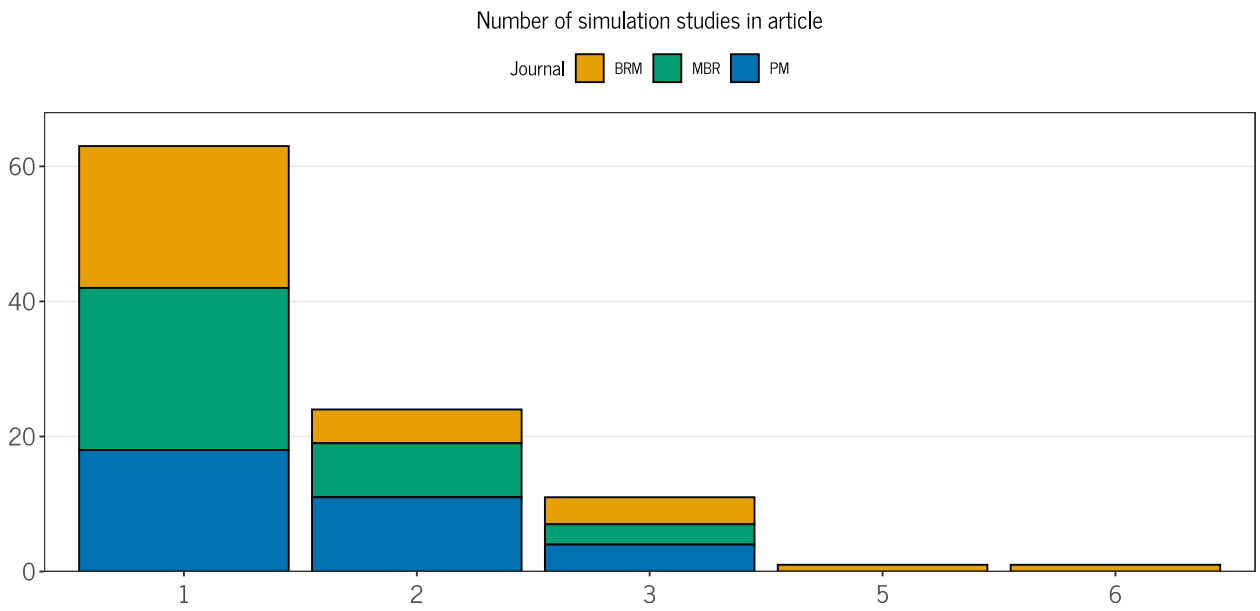



```
## reviewer
ggplot(data = sim_res_fac, aes(x = reviewer)) +
    geom_bar(col = 1) +
    labs(x = NULL, title ="Reviewer") +
    theme(panel.grid.major.x = element_blank())
```

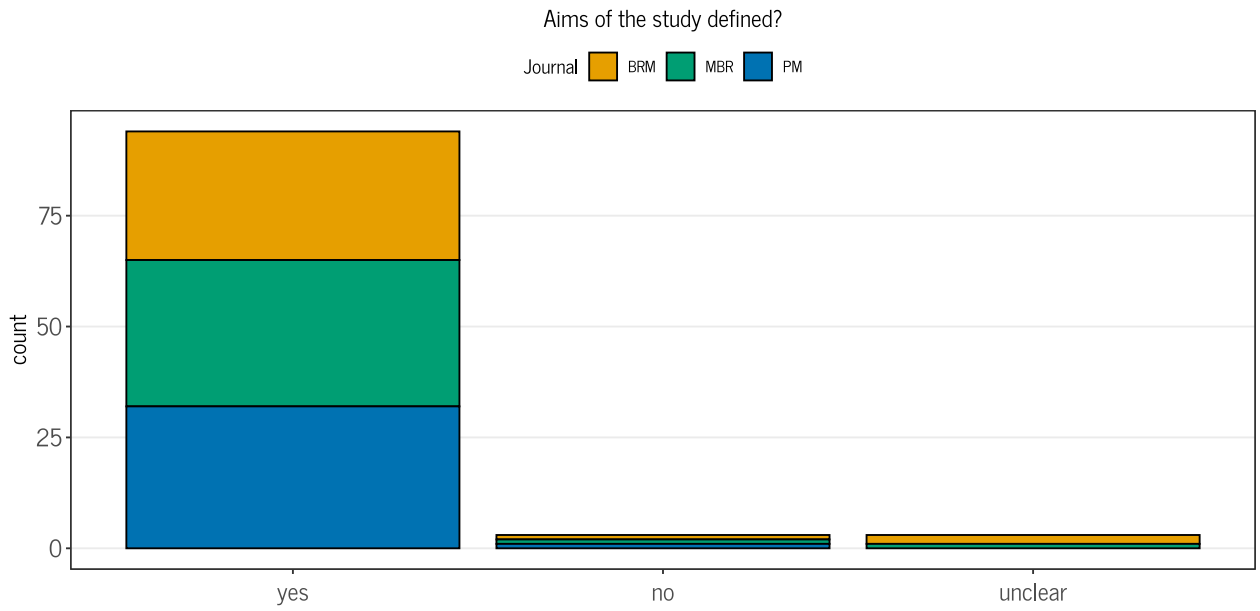



```
## Q2 number of simulation studies
# scale max for plot
q2_plot_max <- max(table(sim_res_fac$nsimstudies_q2)) + 5

q2 <- ggplot(data = sim_res_fac, aes(x = nsimstudies_q2, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, title ="Number of simulation studies in article", fill = "Journal") +
    # scale_fill_discrete_qualitative(palette = pal) +
```

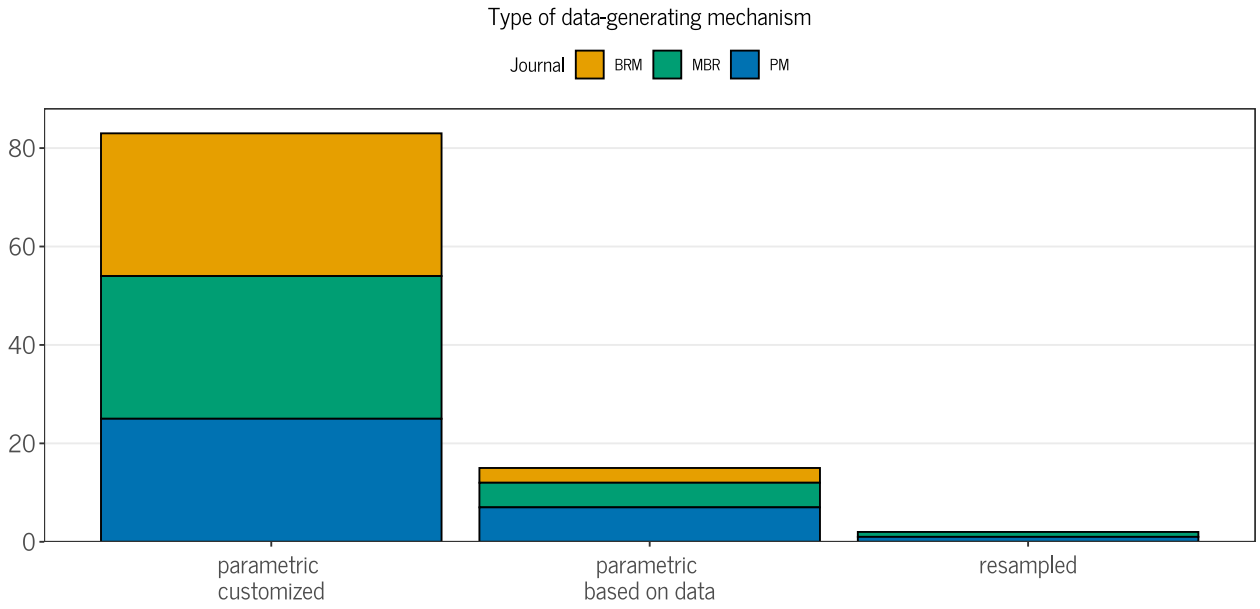
```
    scale_fill_manual(values = cols) +
    scale_y_continuous(limits = c(0, q2_plot_max), expand = c(0,0))+
    theme(panel.grid.major.x = element_blank())+
    labs(y = NULL)
q2
```



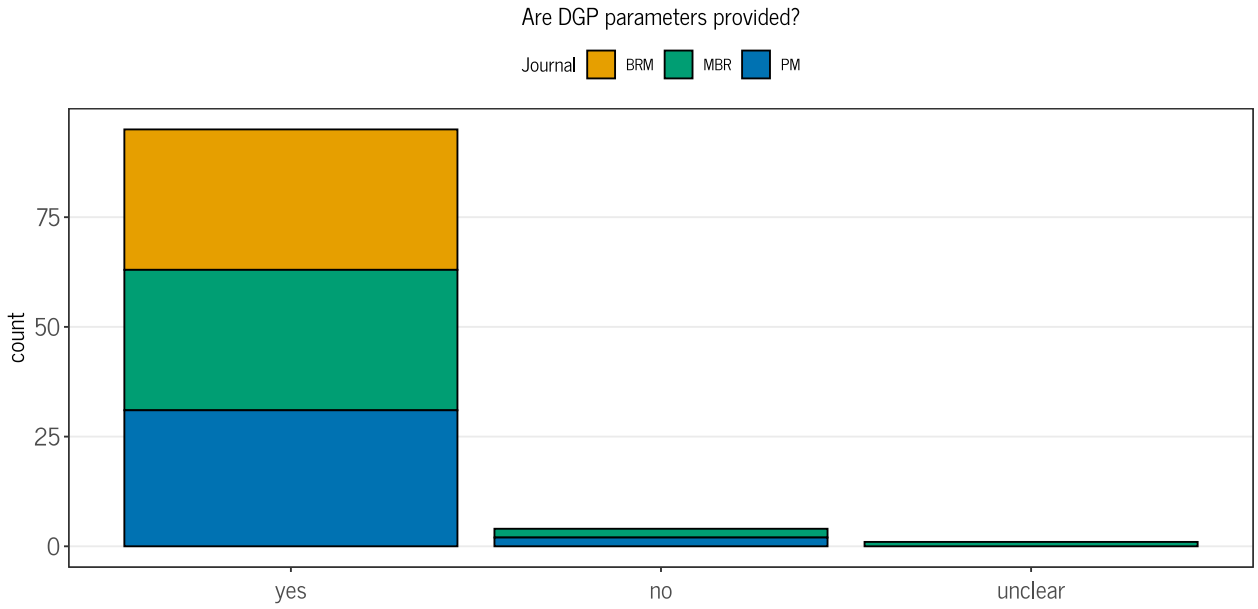

```
## Q3 are the aims of the study defined
q3 <- sim_res_fac %>%
    mutate(aimsdefined_q3 = factor(aimsdefined_q3,
                                   levels = c("yes", "no", "unclear"))) %>%
ggplot(aes(x = aimsdefined_q3, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, title ="Aims of the study defined?", fill = "Journal") +
    scale_fill_manual(values = cols) +
    theme(panel.grid.major.x = element_blank())
q3
```




```
## Q4 type of DGP
q4_plot_max <- max(table(sim_res_fac$dgptype_q4)) + 5

q4 <- sim_res_fac %>%
    mutate(dgptype_q4 = factor(dgptype_q4,
                               levels = c("parametric thin-air",
                                          "parametric based on actual data",
                                          "resampled"),
                               labels = c("parametric \ncustomized",
                                        #"parametric \nthin-air",
                                          "parametric \nbased on data",
                                          "resampled"))) %>%
    mutate(dgptype_q4 = reorder(dgptype_q4, dgptype_q4, length, decreasing = TRUE)) %>%
    ggplot(aes(x = dgptype_q4, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, y = NULL, title ="Type of data-generating mechanism", fill = "Journal") +
    scale_fill_manual(values = cols) +
    scale_y_continuous(limits = c(0, q4_plot_max), expand = c(0,0))+
    theme(panel.grid.major.x = element_blank())
q4
```




```
## from Morris: "97 simulation studies used some form of parametric model to
## generate data while three used resampling methods. Of the 97 that simulated
## from a parametric model, 27 based parameter values on data, one based
## parameter values partly on data, and the remaining 69 on no data. Of these
## 97, 91 (94\%) provided the parameters used." (p. 2079)


## Q5 DGP parameters provided?
q5 <- sim_res_fac %>%
    mutate(dgpparameters_q5 = factor(dgpparameters_q5,
                                     levels = c("yes", "no", "unclear"))) %>%
ggplot(aes(x = dgpparameters_q5, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, title ="Are DGP parameters provided?", fill = "Journal") +
    scale_fill_manual(values = cols) +
    theme(panel.grid.major.x = element_blank())
q5
```

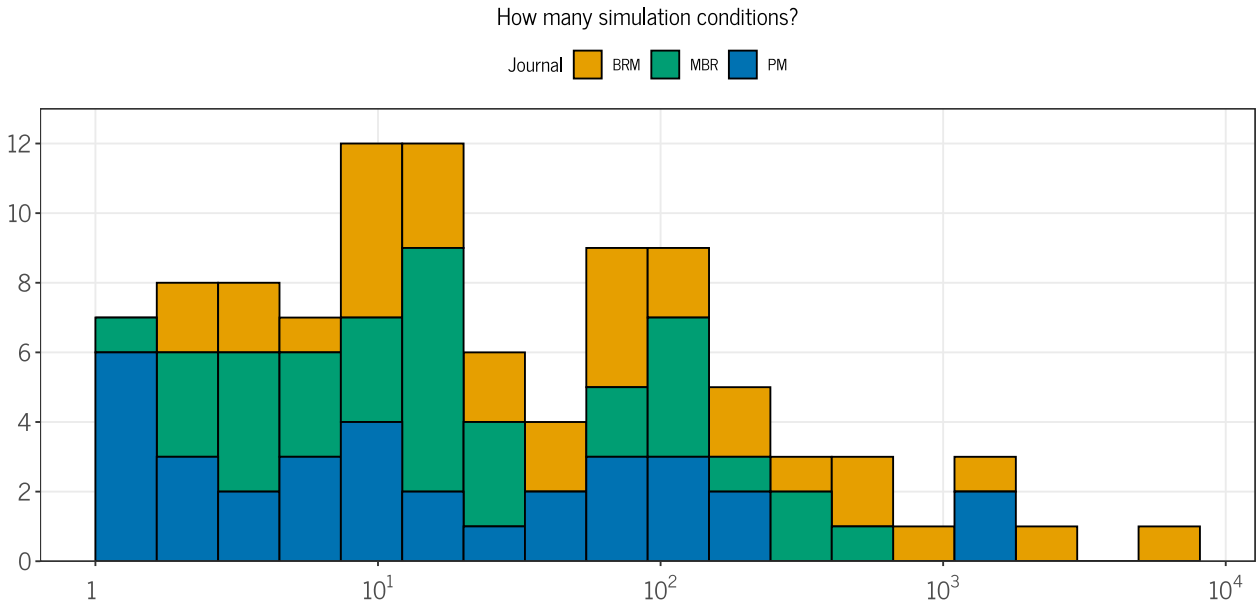



```
## Q6 How many conditions?
summary(sim_res_num$nconds_q6)
```

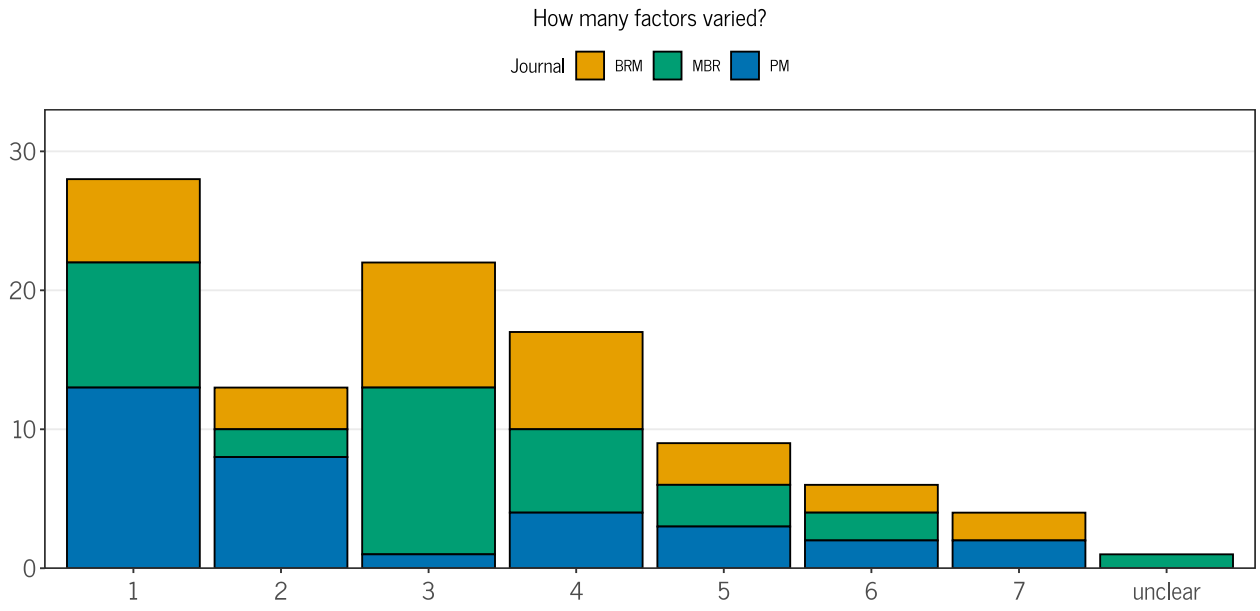
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     1.0     5.0    16.0   185.8    96.0  6000.0       1
```

```
breaks <- c(1, 10, 100, 1000, 10000)
labels <- c("1^phantom(1)", "10^1", "10^2", "10^3", "10^4")
q6 <- ggplot(data = sim_res_num, aes(x = log(nconds_q6), fill = journal)) +
    geom_histogram(breaks = seq(0, log(10000), 0.5), col = 1) +
    scale_x_continuous(breaks = log(breaks), labels = parse(text = labels)) +
    scale_y_continuous(breaks = seq(0, 12, 2), limits = c(0,13), expand = c(0,0)) +
    scale_fill_manual(values = cols) +
    labs(x = NULL, y = NULL,
         title ="How many simulation conditions?", fill = "Journal")
q6
```

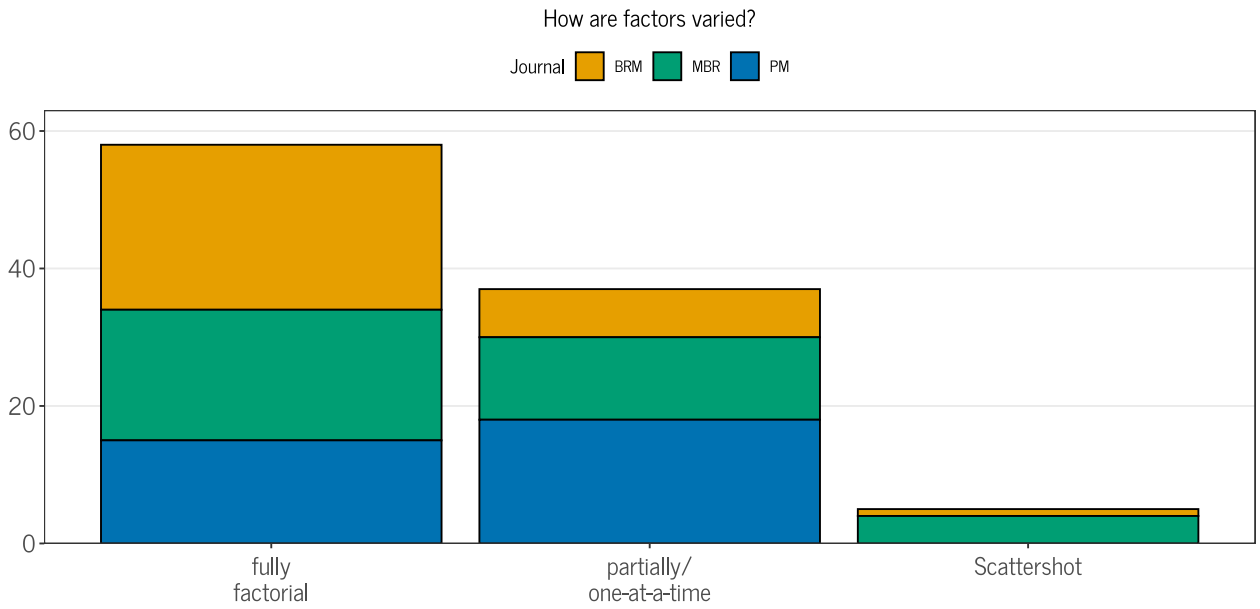
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_bin()`).
```




```
## Q7 How many factors?
q7a_plot_max <- max(table(sim_res_fac$factorsvaried_q7)) + 5

q7a <- sim_res_num %>%
    mutate(factorsvaried_q7_fac = ifelse(is.na(factorsvaried_q7),
                                         "unclear", factorsvaried_q7)) %>%
    ggplot(aes(x = factorsvaried_q7_fac, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, y = NULL,
         title ="How many factors varied?", fill = "Journal") +
    scale_fill_manual(values = cols) +
    scale_y_continuous(limits = c(0, q7a_plot_max), expand = c(0,0))+
    theme(panel.grid.major.x = element_blank())
q7a
```



```
## Q7 Fully factorial?
q7b_plot_max <- max(table(sim_res_fac$dgmfactorial_q7)) + 5

q7b <- sim_res_fac %>%
    mutate(dgmfactorial_q7 = factor(dgmfactorial_q7,
                                    levels = c("fully-factorial",
                                               "one-at-a-time",
                                               "partially-factorial"),
                                    # changed labels here after introduction of terminology "scattersho
                                    labels = c("fully\nfactorial",
                                               "Scattershot",
                                               "partially/\none-at-a-time"))) %>%
    mutate(dgmfactorial_q7 = reorder(dgmfactorial_q7, dgmfactorial_q7,
                                     length, decreasing = TRUE)) %>%
    ggplot(aes(x = dgmfactorial_q7, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, y = NULL,
         title ="How are factors varied?", fill = "Journal") +
    scale_fill_manual(values = cols) +
    scale_y_continuous(limits = c(0, q7b_plot_max), expand = c(0,0))+
    theme(panel.grid.major.x = element_blank())
q7b
```



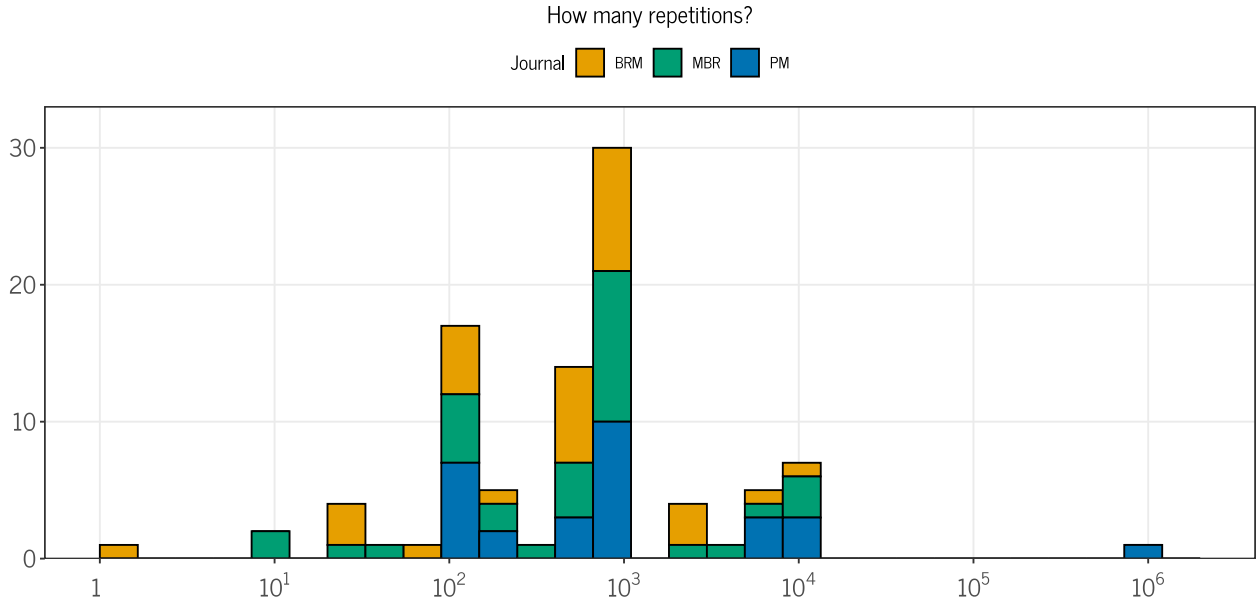

```
## # double check consistency here: how was a single varied factor treated?
## sim_res_fac %>%
##   filter(factorsvaried_q7 == 1) %>%
##   select(reviewer, factorsvaried_q7, dgmfactorial_q7) %>%
##   View()


## Q8 How many repetitions?
summary(sim_res_num$nsim_q8)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
```

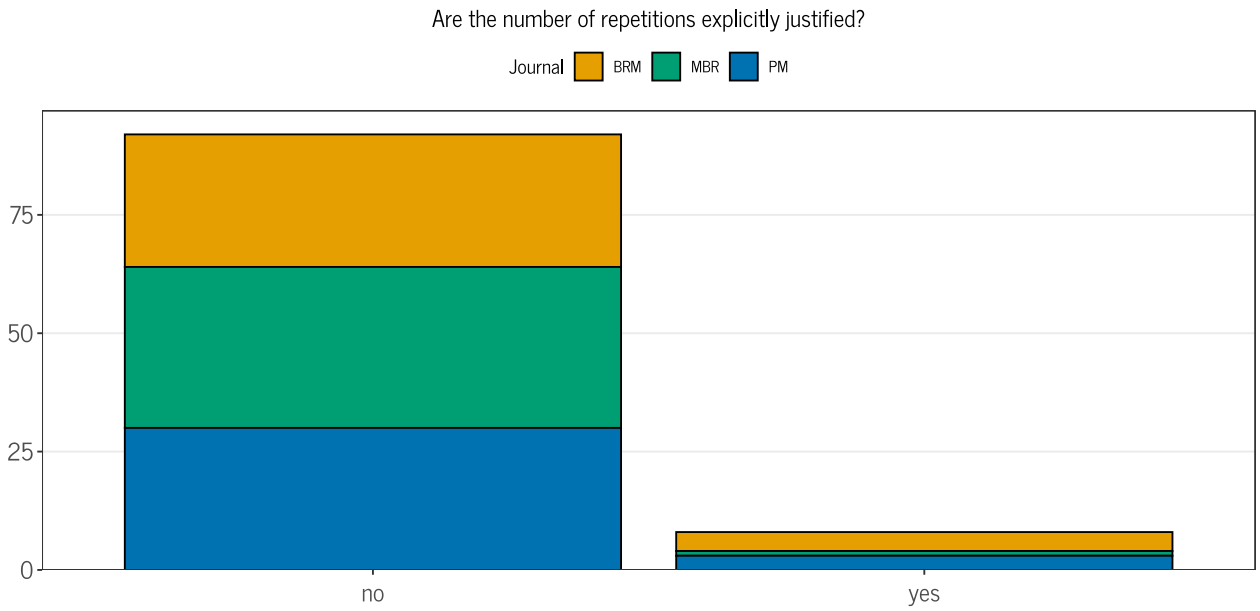
```
##       1     100     900   12198    1000 1000000       6
```

```
breaks <- c(1, 10, 100, 1000, 10000, 100000, 1000000)
labs <- c("1^phantom(1)", "10^1", "10^2", "10^3", "10^4", "10^5", "10^6")
q8 <- ggplot(data = sim_res_num, aes(x = log(nsim_q8), fill = as.factor(journal))) +
    geom_histogram(breaks = seq(0, log(2000000), 0.5), col = 1) +
    labs(x = NULL, y = NULL,
         title ="How many repetitions?", fill = "Journal") +
    scale_x_continuous(breaks = log(breaks), labels = parse(text = labs)) +
    scale_fill_manual(values = cols) +
    scale_y_continuous(limits = c(0,33), expand = c(0,0))
q8
```

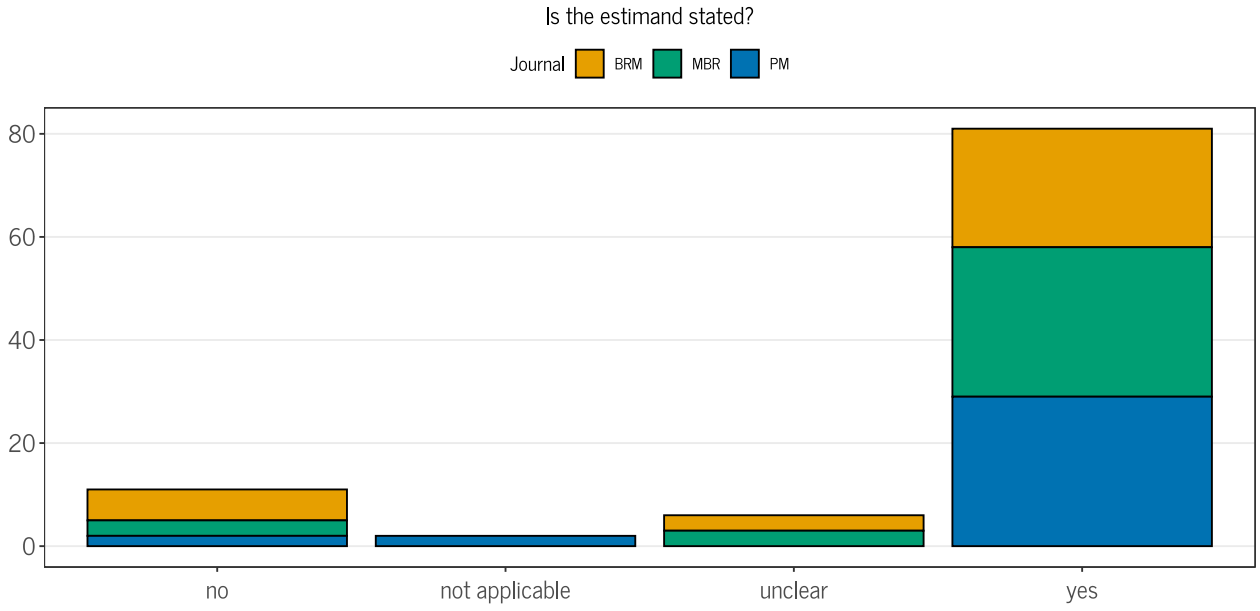
```
## Warning: Removed 6 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```




```
## Q9 Are the number of repetitions justified?
q9_plot_max <- max(table(sim_res_fac$nsimjustified_q9)) + 5

q9 <- ggplot(data = sim_res_fac, aes(x = nsimjustified_q9, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, y = NULL,
         title = "Are the number of repetitions explicitly justified?", fill = "Journal") +
    scale_fill_manual(values = cols) +
    scale_y_continuous(limits = c(0, q9_plot_max), expand = c(0,0))+
    theme(panel.grid.major.x = element_blank())
q9
```




```
## Q10 Is the estimand stated?
q10 <- ggplot(data = sim_res_fac, aes(x = estimandstated_q10, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, y = NULL,
        title ="Is the estimand stated?", fill = "Journal") +
    scale_fill_manual(values = cols) +
    theme(panel.grid.major.x = element_blank())
q10
```

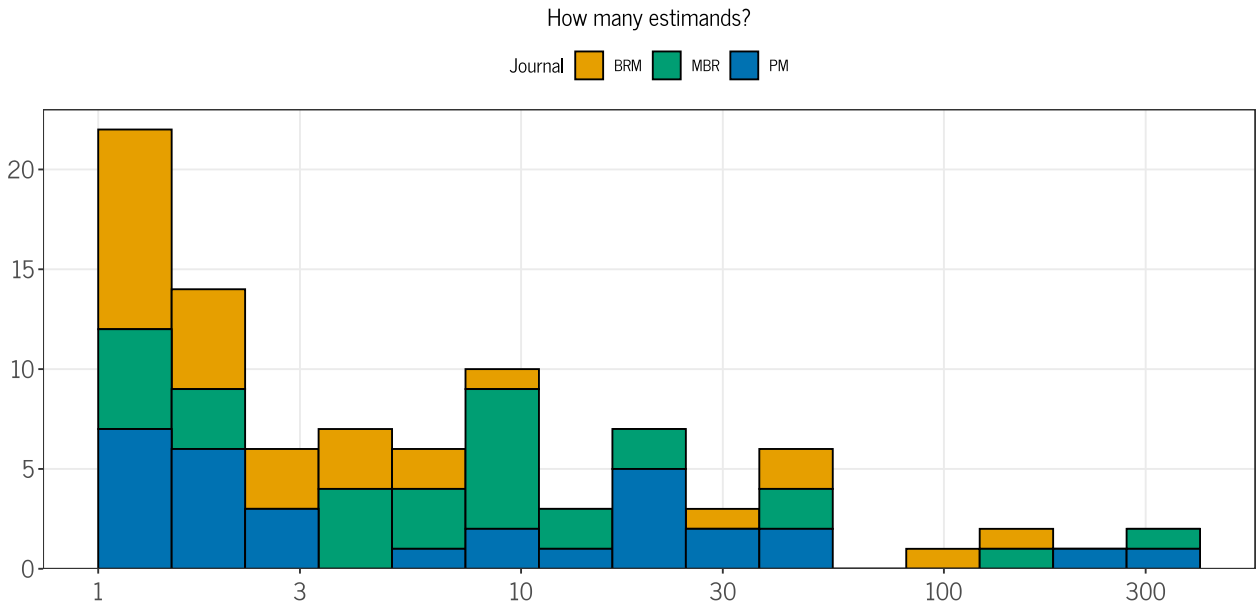



```
## Q11 How many estimands?
summary(sim_res_num$nestimands_q11)
```

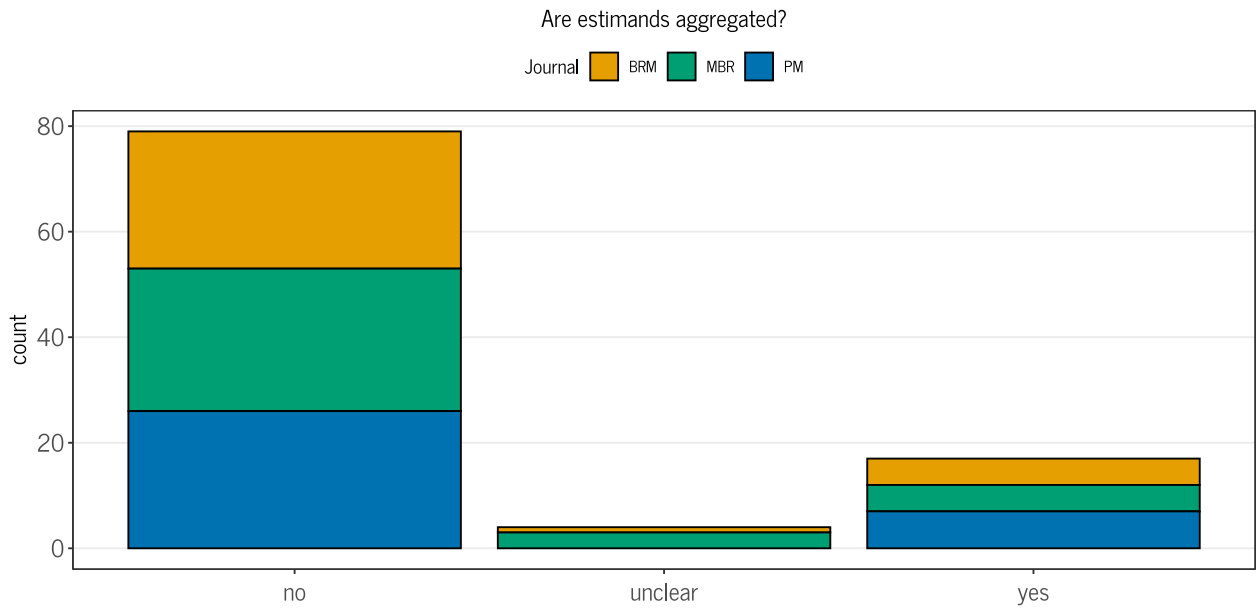
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       1       2       4      22      15     384      10
```

```r
breaks <- c(1, 3, 10, 30, 100, 300)
q11 <- ggplot(data = sim_res_num, aes(x = log(nestimands_q11), fill = journal)) +
    geom_histogram(breaks = seq(0, log(500), 0.4), col = 1) +
    scale_x_continuous(breaks = log(breaks), labels = breaks) +
    scale_y_continuous(limits = c(0, 23), expand = c(0,0))+
    scale_fill_manual(values = cols) +
    labs(x = NULL, y = NULL,
         title ="How many estimands?", fill = "Journal")
q11
```

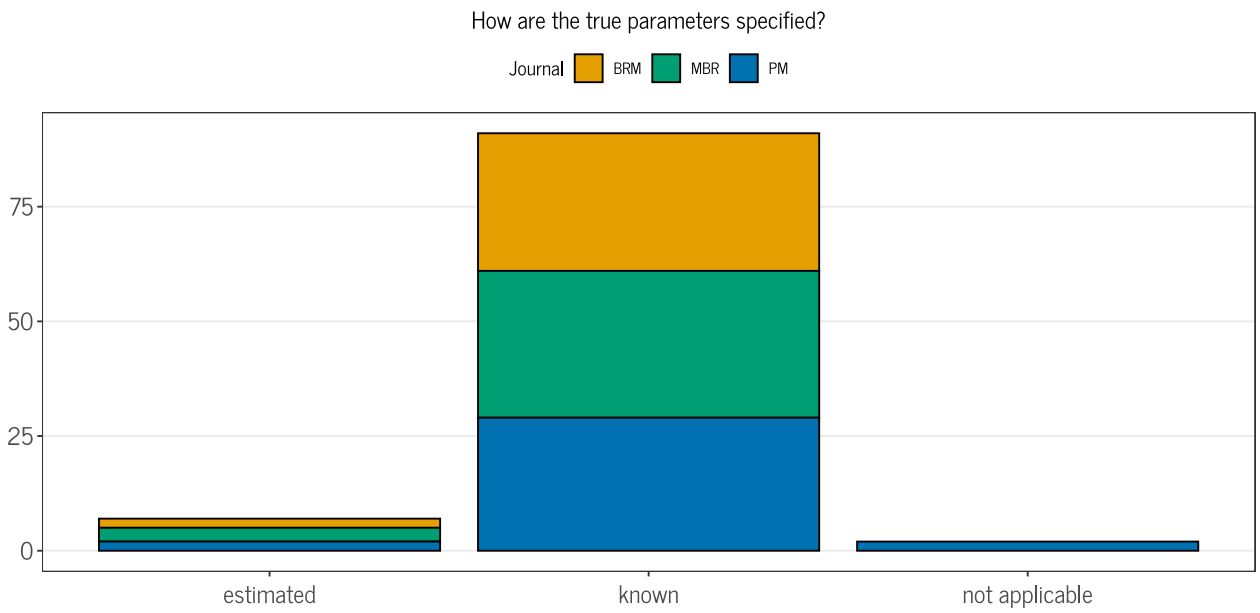
```
## Warning: Removed 10 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```




```r
## Q12 Are estimands aggregated?
q12 <- ggplot(data = sim_res_fac, aes(x = estimandsagg_q12, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, title ="Are estimands aggregated?", fill = "Journal") +
    scale_fill_manual(values = cols) +
    theme(panel.grid.major.x = element_blank())
q12
```



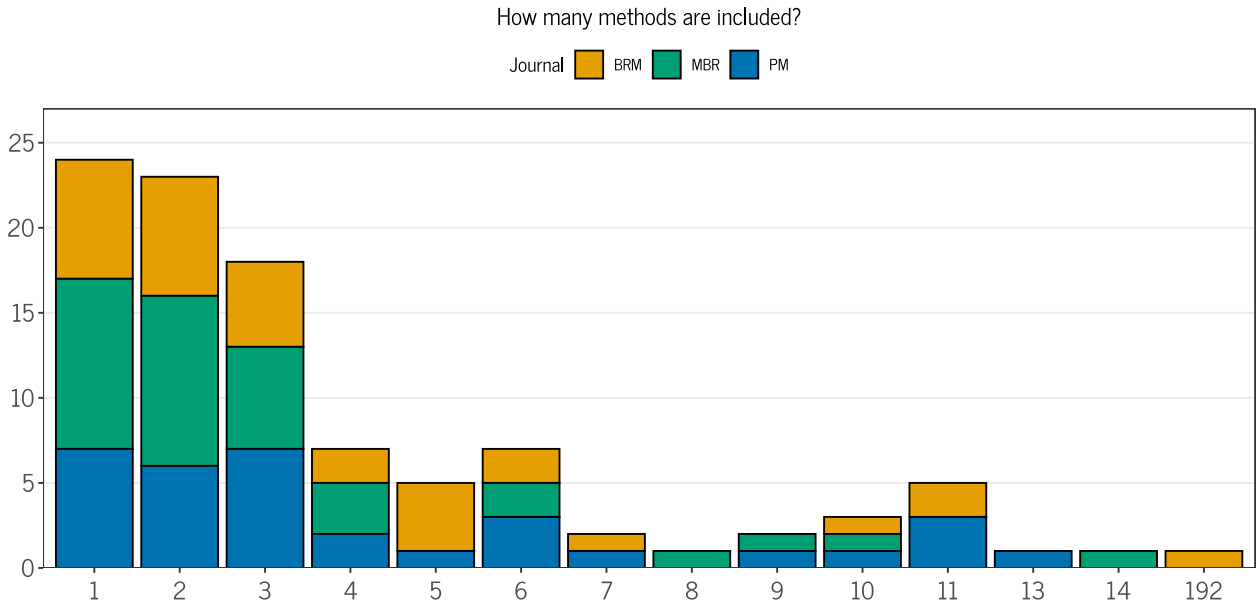

```
## Q13 How are the true parameters specified?
q13 <- ggplot(data = sim_res_fac, aes(x = truetheta_q13, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, y = NULL,
         title ="How are the true parameters specified?", fill = "Journal") +
    scale_fill_manual(values = cols) +
    theme(panel.grid.major.x = element_blank())
q13
```



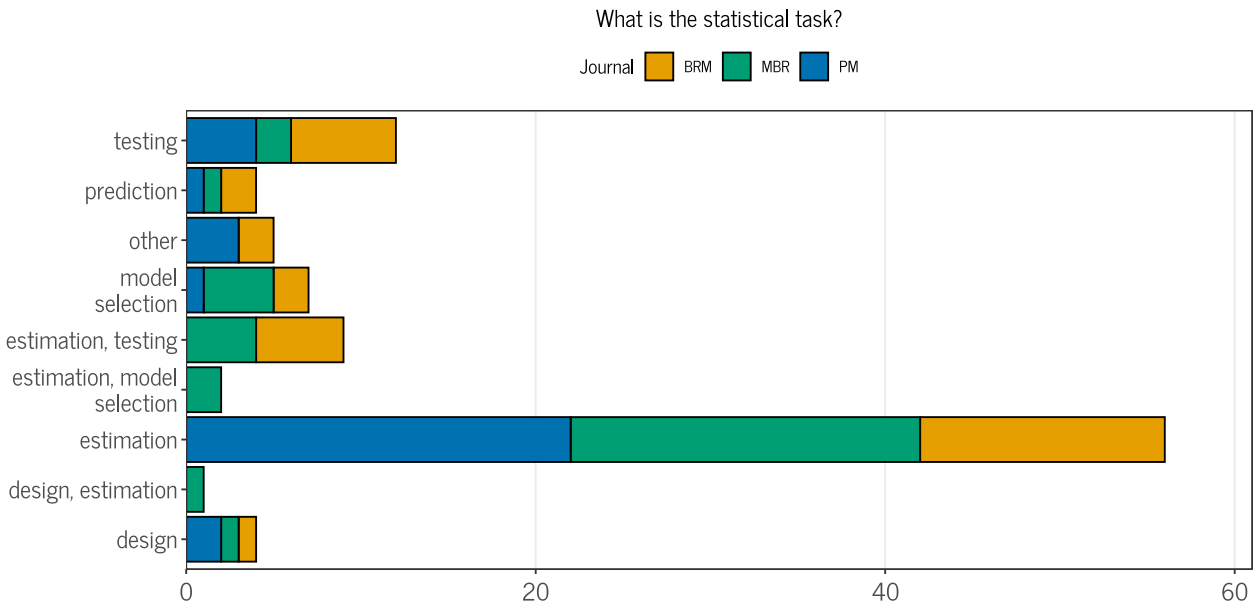

```
## Q14 How many methods are included?
summary(sim_res_num$nmethods_q14)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    2.00    3.00    5.63    5.00  192.00
```

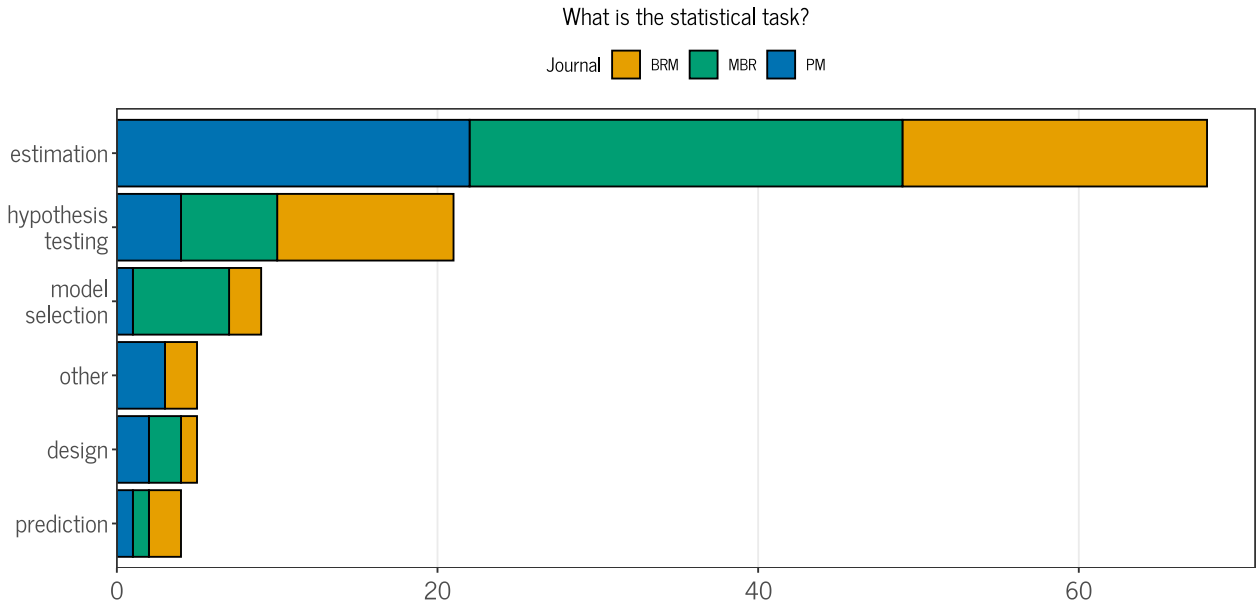
```r
q14_plot_max <- max(table(sim_res_fac$nmethods_q14)) + 3

q14 <- ggplot(data = sim_res_num, aes(x = factor(nmethods_q14), fill = journal)) +
    geom_bar(col = 1) +
    ## scale_x_continuous(breaks = seq(1, 14), limits = c(0.5, 14.5)) +
    labs(x = NULL, y = NULL, title ="How many methods are included?", fill = "Journal") +
    scale_fill_manual(values = cols) +
    scale_y_continuous(limits = c(0, q14_plot_max), expand = c(0,0))+
    theme(panel.grid.major.x = element_blank())
q14
```




```r
## Q15 What is the evaluation target of the simulation?
q15_plot_max <- max(table(sim_res_fac$target_q15)) + 5

q15a <- sim_res_fac %>%
    mutate(target_q15 = as.factor(target_q15)) %>%
    mutate(target_q15 = reorder(target_q15, target_q15, length)) %>%
    mutate(target_q15 = gsub("model selection", "model\nselection", target_q15)) %>%
    ggplot(aes(x = target_q15, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, y = NULL, title ="What is the statistical task?", fill = "Journal") +
    scale_fill_manual(values = cols) +
    scale_y_continuous(limits = c(0, q15_plot_max), expand = c(0,0))+
    theme(panel.grid.major.y = element_blank()) +
    coord_flip()
q15a
```




```
# to keep it consistent with other questions, also spread apart results here
q15a_new <- sim_res_fac %>%
    separate_wider_delim(target_q15,
                         delim = ",",
                         names_sep = "_",
                         too_few = "align_start") %>%
    pivot_longer(cols = contains("target"),
                 names_to = NULL,
                 values_to = "target",
                 values_drop_na = TRUE) %>%
    mutate(target = str_trim(target)) %>%
    mutate(target = gsub("model selection", "model\nselection", target)) %>%
    mutate(target = gsub("testing", "hypothesis\ntesting", target)) %>%
    mutate(target = as.factor(target)) %>%
    mutate(target = reorder(target, target, length)) %>%
    ggplot(aes(x = target, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, y = NULL, title ="What is the statistical task?",
         fill = "Journal") +
    scale_fill_manual(values = cols) +
    scale_y_continuous(limits = c(0, q15_plot_max + 10), expand = c(0,0))+
    theme(panel.grid.major.y = element_blank()) +
    coord_flip()

q15a_new
```

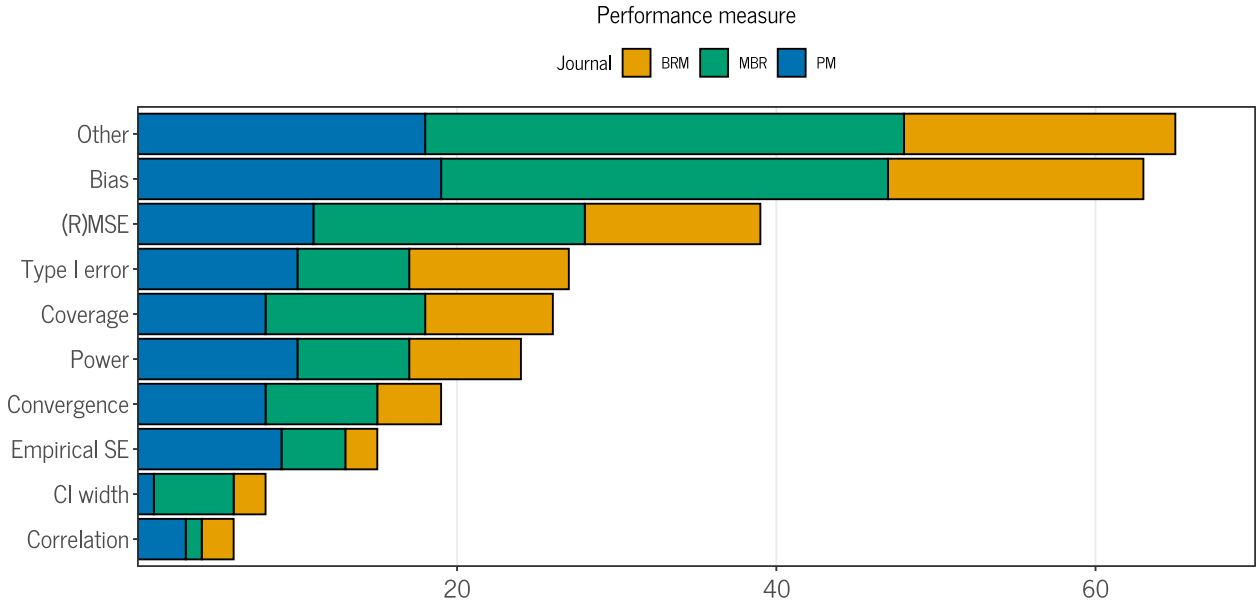



```
## Q15 Which performance measures were used?
# Spread "Other" apart
q15_other <- sim_res_fac %>%
  separate_wider_delim(pmother_q15,
                       delim = ",",
                       names_sep = "_",
                       too_few = "align_start") %>%
  pivot_longer(cols = contains("pmother"),
               names_to = NULL,
               values_to = "pmother",
               values_drop_na = TRUE) %>%
  dplyr::select(pmother, journal) %>%
  # remove whitespace
  mutate(pmother = str_trim(pmother)) %>%
  mutate(pmother = str_replace(pmother, ".*correlation.*", "Correlation")) %>%
  # mutate(pmother = str_replace(pmother, ".*standard deviation.*", "SD")) %>%
  mutate(pmother = str_replace(pmother, ".*bias.*", "Bias")) %>%
  mutate(pmother = as.factor(pmother)) %>%
  mutate(pmother = forcats::fct_lump_n(pmother, 2)) %>%
  group_by(journal) %>%
  count(pmother) %>%
  rename(PM = pmother,
         count = n)

# Visualize
q15b <- sim_res_fac %>%
    group_by(journal) %>%
    summarise("Convergence" = sum(pmconvergence_q15 == "yes"),
              "Bias" = sum(pmbias_q15 == "yes"),
              "Empirical SE" = sum(pmempse_q15 == "yes"),
              "(R)MSE" = sum(pm_r_mse_q15 == "yes"),
              "Coverage" = sum(pmcover_q15 == "yes"),
              "Type I error" = sum(pmtypeierror_q15 == "yes"),
              "Power" = sum(pmpower_q15 == "yes"),
```

```
            "CI width" = sum(pmciwidth_q15 == "yes")) %>%
  gather(key = "PM", value = "count", "Convergence", "Bias", "(R)MSE",
         "Empirical SE", "Coverage", "Type I error",
         "Power", "CI width") %>%
  bind_rows(q15_other) %>%
  mutate(PM = as.factor(PM)) %>%
  mutate(PM = reorder(PM, count, sum)) %>%
  group_by(journal, PM) %>%
  summarise(count = sum(count)) %>%
  ggplot(aes(x = PM, y = count, fill = journal)) +
  geom_bar(stat = "identity", col = 1) +
  labs(x = NULL,
       y = NULL,
       title ="Performance measure",
       fill = "Journal") +
  scale_fill_manual(values = cols) +
  scale_y_continuous(limits = c(0,70), expand = c(0,0),
                     breaks = c(20, 40, 60))+
  theme(panel.grid.major.y = element_blank()) +
  coord_flip()
```

```
## `summarise()` has grouped output by 'journal'. You can override using the
## `.groups` argument.
```

```
q15b
```




```
# Double check if absolute/relative bias ever occur with bias rated as "no"
sim_res_fac %>%
  separate_wider_delim(pmother_q15,
                       delim = ",",
                       names_sep = "_",
                       too_few = "align_start") %>%
```

```
  pivot_longer(cols = contains("pmother"),
               names_to = NULL,
               values_to = "pmother",
               values_drop_na = TRUE) %>%
  mutate(pmother = str_trim(pmother)) %>%
  # mutate(pmother = str_replace(pmother, ".*bias.*", "bias")) %>%
  filter(grepl("bias", pmother)) %>%
  dplyr::select(reviewer, pmbias_q15, pmother)
```

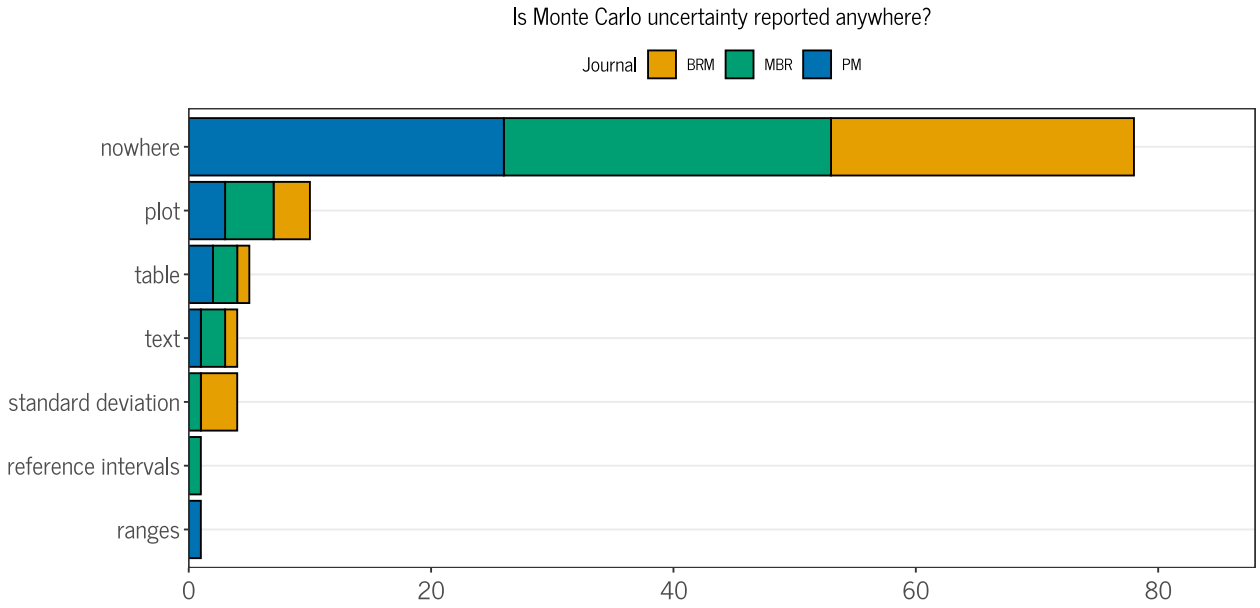
```
## # A tibble: 18 x 3
##    reviewer pmbias_q15 pmother
##    <fct>    <fct>      <chr>
##  1 FB       yes        absolute bias
##  2 FB       yes        relative bias
##  3 FB       yes        relative bias
##  4 FB       no         absolute bias
##  5 FB       yes        bias of standard errors
##  6 FB       no         relative bias
##  7 FB       yes        relative bias
##  8 FB       no         relative bias
##  9 FB       no         relative bias of standard errors
## 10 FB       no         relative bias
## 11 FB       no         relative bias
## 12 FB       no         relative bias
## 13 FB       no         relative bias of se
## 14 FB       no         relative bias
## 15 FB       no         absolute relative bias
## 16 FB       yes        relative bias of se
## 17 FB       no         relative bias
## 18 BS       yes        SD of SE bias (as uncertainty)
```

```
# What is included in "Standard Deviation"?
sim_res_fac %>%
  separate_wider_delim(pmother_q15,
                       delim = ",",
                       names_sep = "_",
                       too_few = "align_start") %>%
  pivot_longer(cols = contains("pmother"),
               names_to = NULL,
               values_to = "pmother",
               values_drop_na = TRUE) %>%
  mutate(pmother = str_trim(pmother)) %>%
  # mutate(pmother = str_replace(pmother, ".*bias.*", "bias")) %>%
  filter(grepl("standard deviation", pmother)) %>%
  dplyr::select(pmother)
```

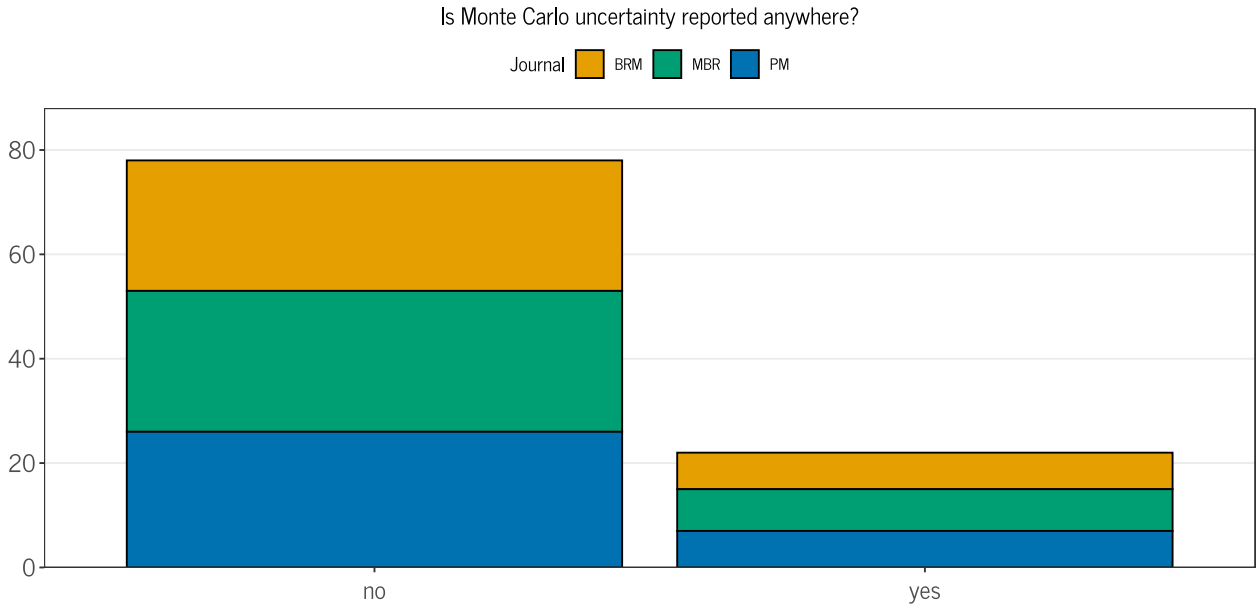
```
## # A tibble: 8 x 1
##   pmother
##   <chr>
## 1 means and standard deviations of the estimates
## 2 efficiency (standard deviation of estimates)
## 3 standard deviations
```

```
## 4 standard deviations
## 5 standard deviation
## 6 efficiency (standard deviation of estimates)
## 7 standard deviation
## 8 standard deviation over aggregated estimates
```

```r
## Q16 Is Monte Carlo uncertainty reported anywhere?
q16_plot_max <- max(table(sim_res_fac$mcerrors_q16)) + 10

q16 <- sim_res_fac %>%
  separate_wider_delim(mcerrors_q16,
                       delim = ",",
                       names_sep = "_",
                       too_few = "align_start") %>%
  pivot_longer(cols = contains("mcerrors"),
               names_to = NULL,
               values_to = "mcerrors",
               values_drop_na = TRUE) %>%
  mutate(mcerrors = str_trim(mcerrors)) %>%
  mutate(mcerrors = gsub("boxplot", "plot", mcerrors)) %>%
  mutate(mcerrors = gsub("quantile plots", "plot", mcerrors)) %>%
  mutate(mcerrors = gsub("figures", "plot", mcerrors)) %>%
  mutate(mcerrors = gsub("tables", "table", mcerrors)) %>%
  mutate(mcerrors = as.factor(mcerrors)) %>%
  mutate(mcerrors = reorder(mcerrors, mcerrors, length)) %>%
  ggplot(aes(x = mcerrors, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, y = NULL,
         title ="Is Monte Carlo uncertainty reported anywhere?", fill = "Journal") +
    scale_fill_manual(values = cols) +
    theme(panel.grid.major.x = element_blank()) +
    scale_y_continuous(limits = c(0, q16_plot_max), expand = c(0,0))+
    coord_flip()
q16
```

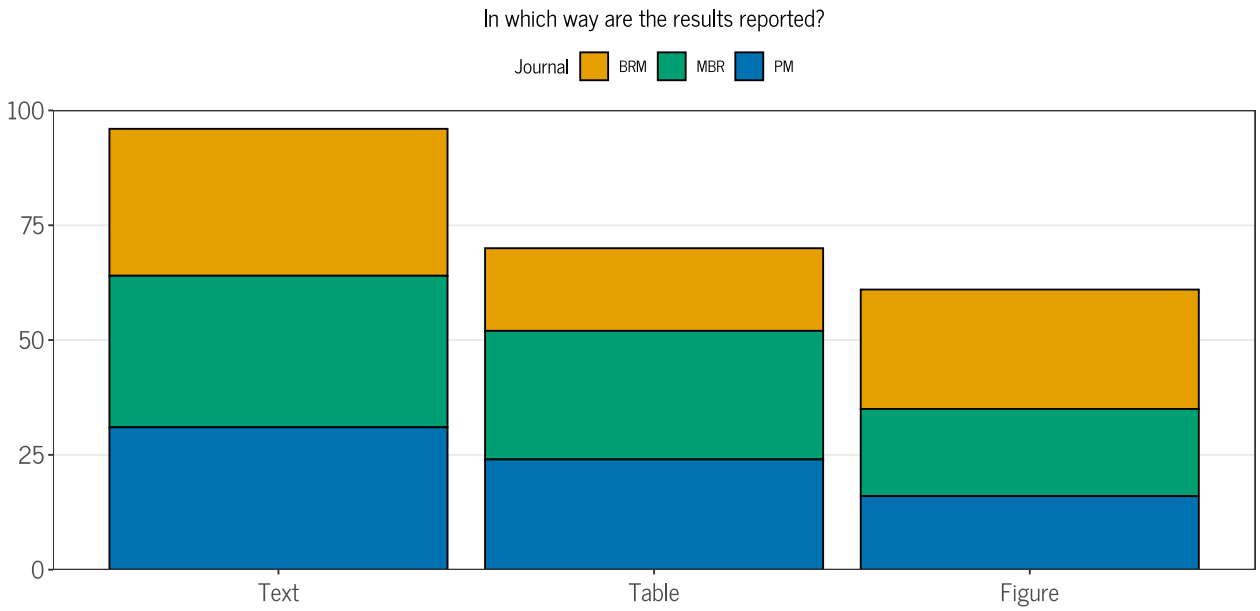


```
# Same plot, only yes/no
q16_yn <- sim_res_fac %>%
  mutate(mcerrors_q16 = ifelse(mcerrors_q16 == "nowhere",
                               "no",
                               "yes")) %>%
  mutate(mcerrors_q16 = as.factor(mcerrors_q16)) %>%
  # mutate(mcerrors_q16 = reorder(mcerrors_q16, mcerrors_q16, length)) %>%
  ggplot(aes(x = mcerrors_q16, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, y = NULL,
         title ="Is Monte Carlo uncertainty reported anywhere?", fill = "Journal") +
    scale_fill_manual(values = cols) +
    theme(panel.grid.major.x = element_blank()) +
    scale_y_continuous(limits = c(0, q16_plot_max), expand = c(0,0))
    # coord_flip()
q16_yn
```

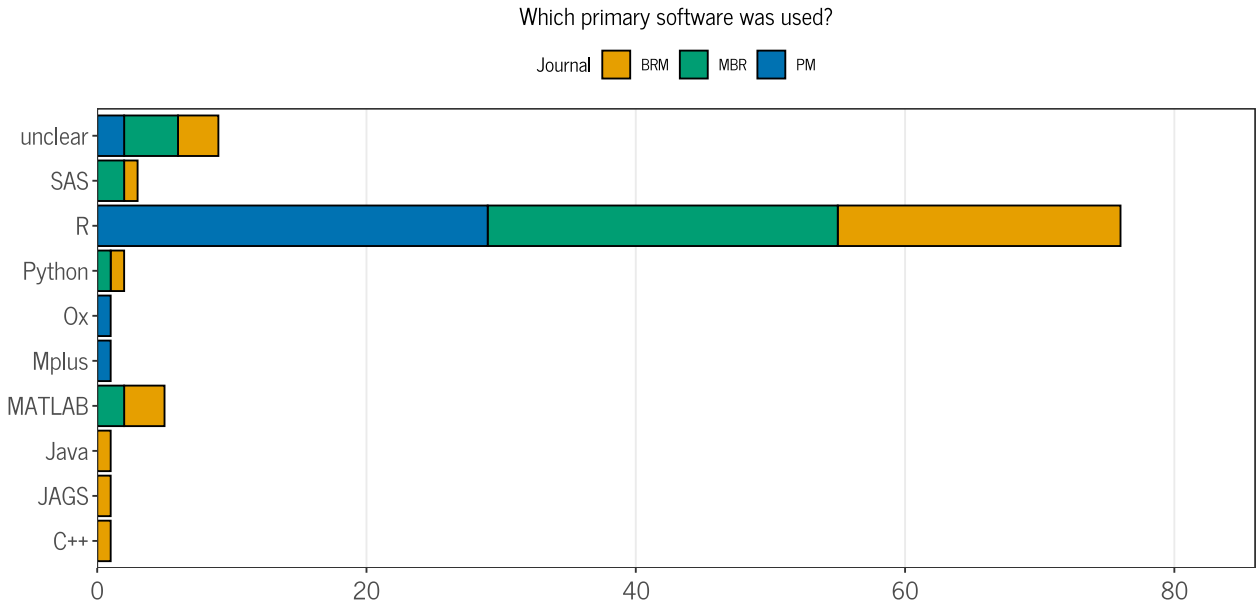



```
## Q17 In which way are the results reported?
q17 <- sim_res_fac %>%
    group_by(journal) %>%
    summarise("Figure" = sum(resultsfigure_q17 == "yes"),
              "Table" = sum(resultstable_q17 == "yes"),
              "Text" = sum(resultstext_q17 == "yes"),
              "Other" = sum(resultsother_q17 == "yes")) %>%
    gather(key = "Type", value = "count", "Figure", "Table", "Text", "Other") %>%
    # omit "Other" category for plot
    filter(Type != "Other") %>%
    mutate(Type = as.factor(Type)) %>%
    mutate(Type = reorder(Type, count, sum, decreasing = TRUE)) %>%
    ggplot(aes(x = Type, y = count, fill = journal)) +
    geom_bar(stat = "identity", col = 1) +
    labs(x = NULL, y = NULL,
         title ="In which way are the results reported?", fill = "Journal") +
```

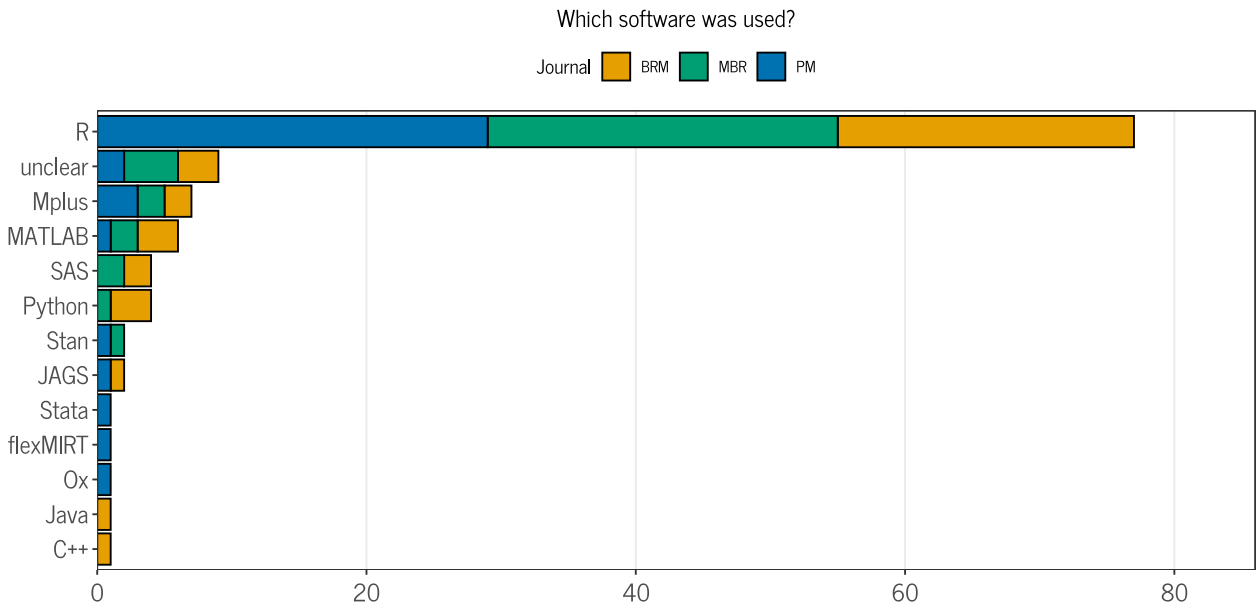
```
    scale_fill_manual(values = cols) +
    scale_y_continuous(limits = c(0,100), expand = c(0,0))+
    theme(panel.grid.major.x = element_blank())
q17
```



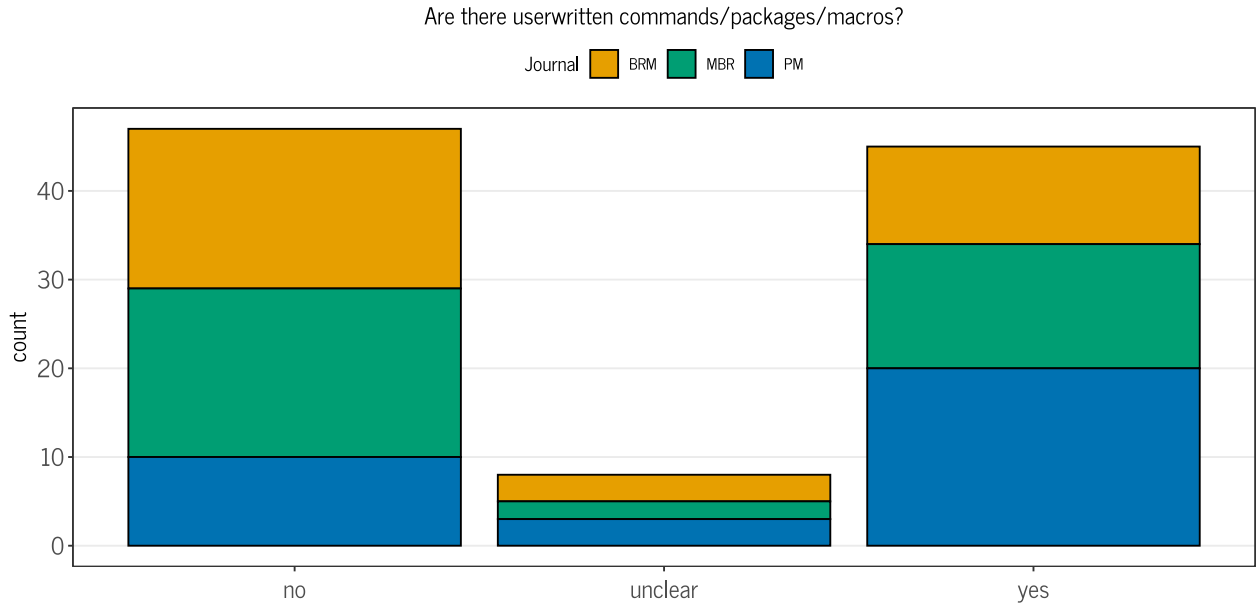

```
## Q18 Which software was used to conduct the simulation?
q18_plot_max <- max(table(sim_res_fac$software_1_q18)) + 10
q18a <- ggplot(data = sim_res_fac, aes(x = software_1_q18, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, y = NULL,
         title ="Which primary software was used?", fill = "Journal") +
    scale_y_continuous(limits = c(0, q18_plot_max), expand = c(0,0))+
    scale_fill_manual(values = cols) +
    theme(panel.grid.major.y = element_blank()) +
    coord_flip()
q18a
```




```
# add information from software_2_q18 and software_3_q18
q18b <- sim_res_fac %>%
  dplyr::select(starts_with("software"), journal) %>%
  pivot_longer(cols = starts_with("software"),
               names_to = NULL,
               values_to = "software",
               values_drop_na = TRUE) %>%
  mutate(software = as.factor(software)) %>%
  mutate(software = reorder(software, software, length)) %>%
  ggplot(aes(x = software, fill = journal)) +
  geom_bar(col = 1) +
  labs(x = NULL, y = NULL, title ="Which software was used?", fill = "Journal") +
  scale_fill_manual(values = cols) +
  scale_y_continuous(limits = c(0, q18_plot_max), expand = c(0,0))+
  theme(panel.grid.major.y = element_blank()) +
  coord_flip()
q18b
```




```
## Q19 Are there userwritten commands/packages/macros?
q19 <- ggplot(data = sim_res_fac, aes(x = userwritten_q19, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, title ="Are there userwritten commands/packages/macros?", fill = "Journal") +
    scale_fill_manual(values = cols) +
    theme(panel.grid.major.x = element_blank())
q19
```

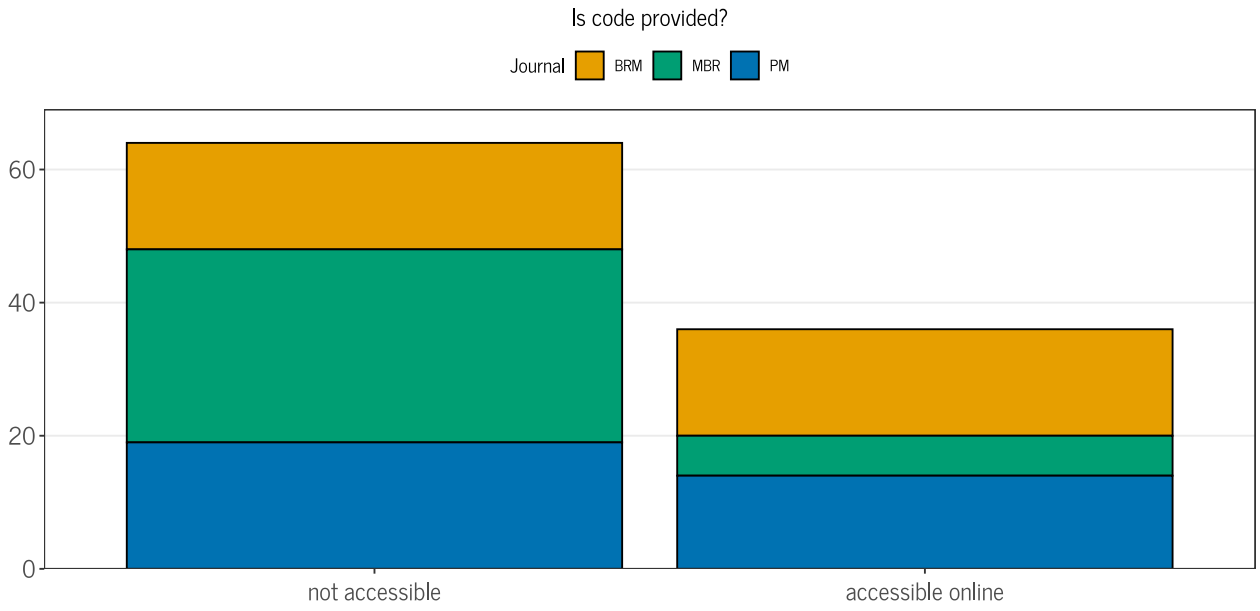



```
## Q20 Is code provided?
q20_plot_max <- max(table(sim_res_fac$codeprovided_q20)) + 5

# Reordered this such that negative is on the left to stay consistent
q20 <- sim_res_fac %>%
    mutate(codeprovided_q20 = as.factor(codeprovided_q20)) %>%
```

```
    mutate(codeprovided_q20 = reorder(codeprovided_q20, codeprovided_q20,
                                      length, decreasing = TRUE)) %>%
    ggplot(aes(x = codeprovided_q20, fill = journal)) +
    geom_bar(col = 1, col = 1) +
    labs(x = NULL, y = NULL, title ="Is code provided?", fill = "Journal") +
    scale_fill_manual(values = cols) +
    scale_y_continuous(limits = c(0, q20_plot_max), expand = c(0,0))+
    theme(panel.grid.major.x = element_blank())
```

```
## Warning in geom_bar(col = 1, col = 1): Ignoring unknown parameters: `col`
```

```
q20
```

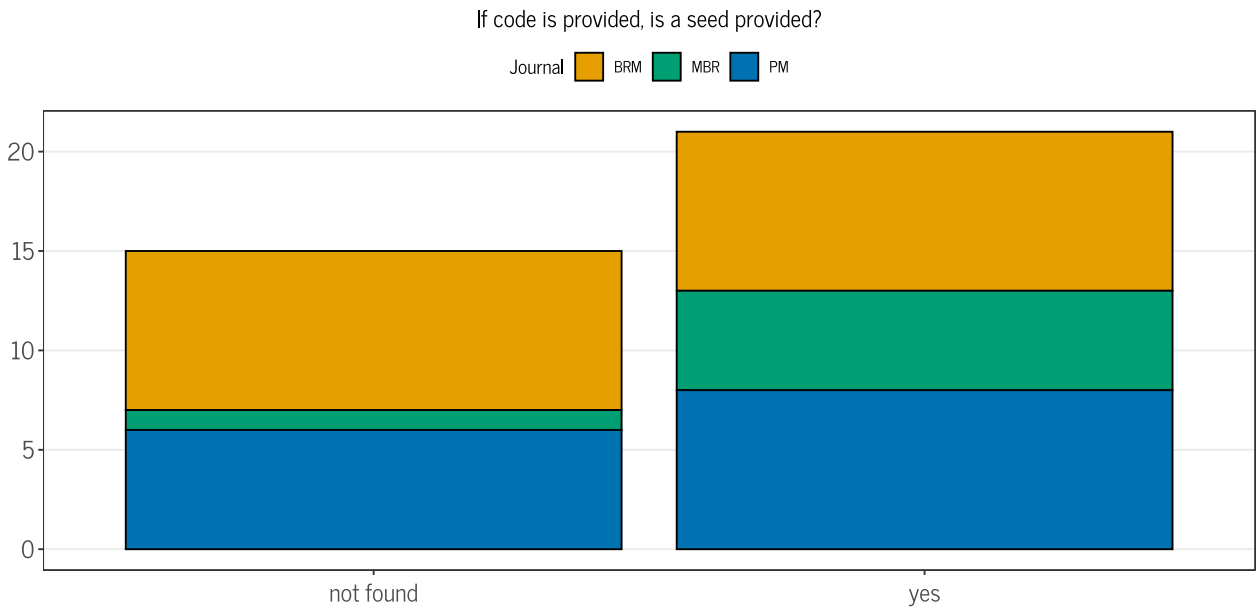



```
## Q21 If code is provided, is a seed provided?
q21 <- ggplot(data = sim_res_fac, aes(x = seedprovided_q21, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, title ="If code is provided, is a seed provided?", fill = "Journal") +
    scale_fill_manual(values = cols) +
    theme(panel.grid.major.x = element_blank())


# Compute as conditional on code provided
q21 <- sim_res_fac %>%
  filter(codeprovided_q20 == "accessible online") %>%
  ggplot(aes(x = seedprovided_q21, fill = journal))+
  geom_bar(col = 1)+
  labs(x = NULL,
       y = NULL,
       title ="If code is provided, is a seed provided?", fill = "Journal") +
    scale_fill_manual(values = cols) +
    theme(panel.grid.major.x = element_blank())
sim_res_fac %>%
```

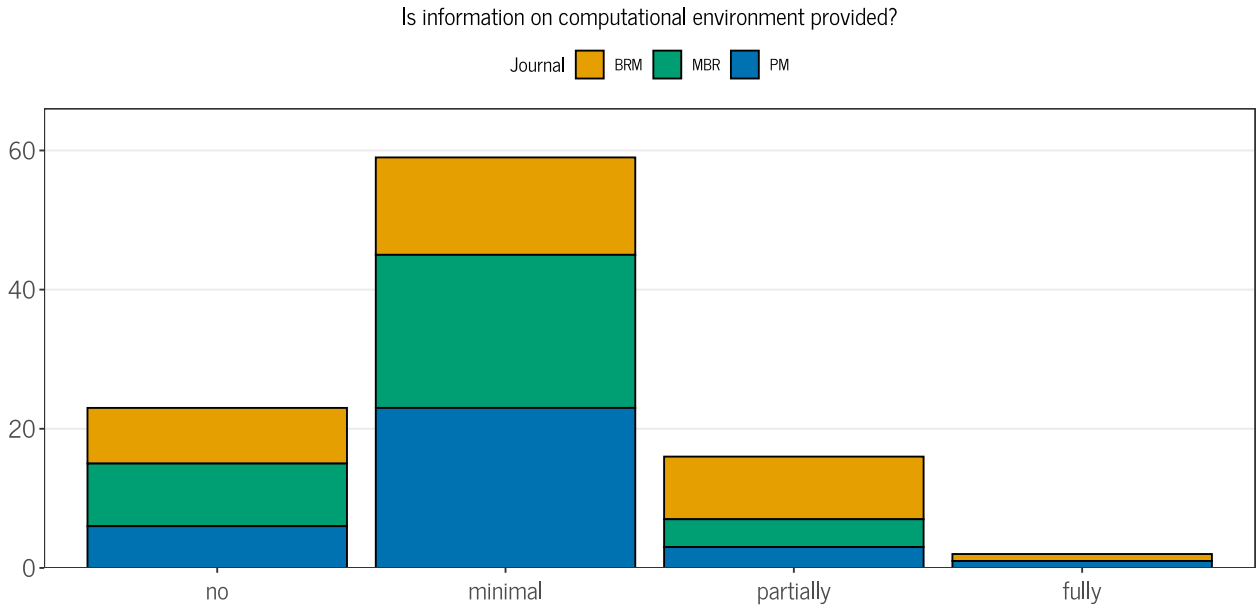
```
    filter(codeprovided_q20 == "accessible online") %>%
    count(seedprovided_q21)
```

```
## # A tibble: 2 x 2
##   seedprovided_q21     n
##   <fct>            <int>
## 1 not found           15
## 2 yes                 21
```

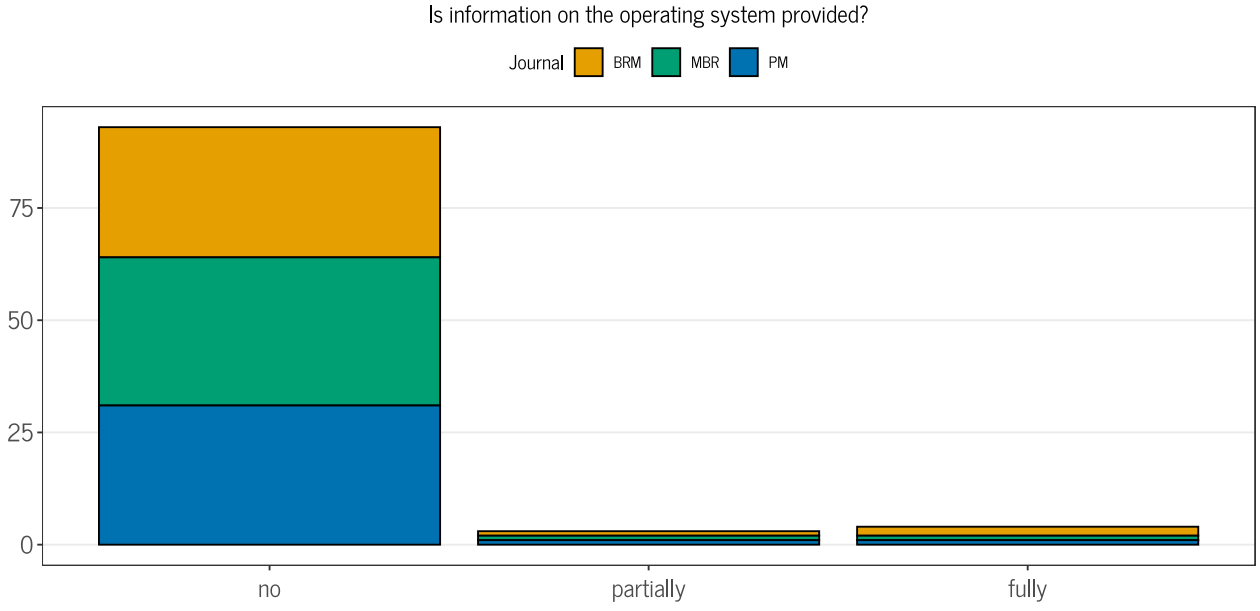
```
q21
```



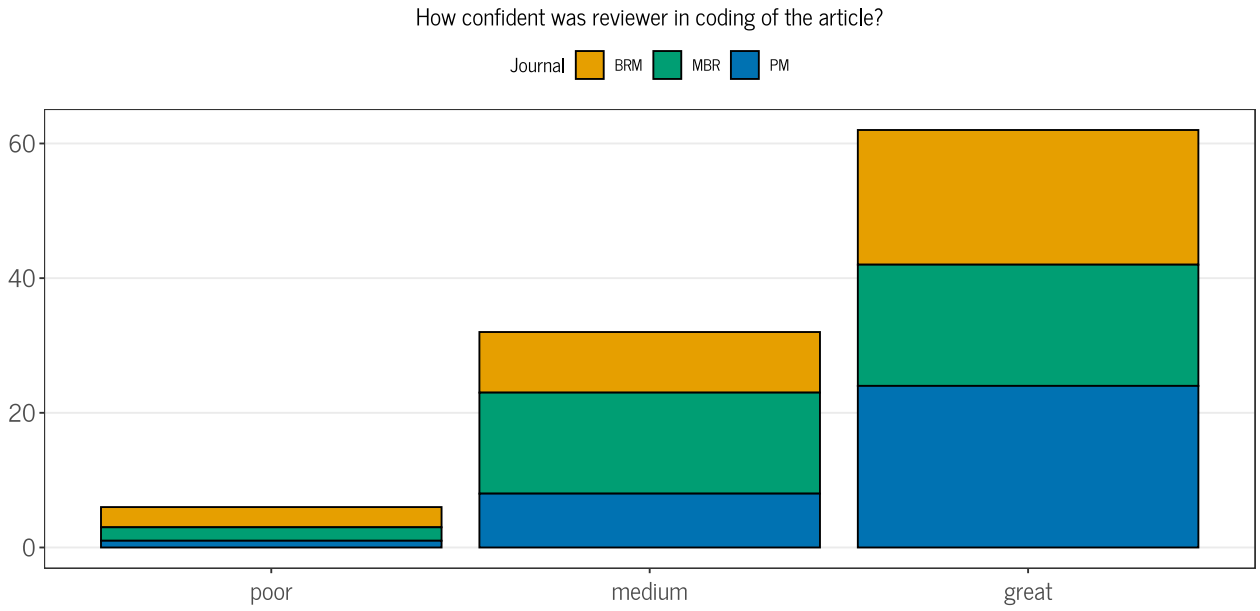

```
## Q22 Is information on the computational environment provided?
q22_plot_max <- max(table(sim_res_fac$compenvironment_q22)) + 7

q22 <- sim_res_fac %>%
    mutate(compenvironment_q22 = factor(compenvironment_q22,
                                        levels = c("no", "minimal", "partially", "fully"))) %>%
    ggplot(aes(x = compenvironment_q22, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, y = NULL,
         title ="Is information on computational environment provided?",
         fill = "Journal")  +
    scale_fill_manual(values = cols) +
    scale_y_continuous(limits = c(0, q22_plot_max), expand = c(0,0)) +
    theme(panel.grid.major.x = element_blank())
q22
```




```
## Q23 Is information on the operating system provided?
q23 <- sim_res_fac %>%
    mutate(compos_q23 = factor(compos_q23,
                               levels = c("no", "partially", "fully"))) %>%
    ggplot(aes(x = compos_q23, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL, y = NULL,
         title ="Is information on the operating system provided?",
         fill = "Journal")  +
    scale_fill_manual(values = cols) +
    theme(panel.grid.major.x = element_blank())
q23
```



```r
## Q24 How confident was reviewer in coding of the article?
q24 <- sim_res_fac %>%
    mutate(coding_confidence = factor(coding_confidence,
                                levels = c("poor", "medium", "great"))) %>%
    ggplot(aes(x = coding_confidence, fill = journal)) +
    geom_bar(col = 1) +
    labs(x = NULL,
         y = NULL,
         title ="How confident was reviewer in coding of the article?",
         fill = "Journal")  +
    scale_fill_manual(values = cols) +
    theme(panel.grid.major.x = element_blank())
q24
```




```r
## composite plot 1 - problematic questions
plotList1 <- lapply(X = list(q9, q16_yn, q20, q22),
                    ## this plot requires a different title size
                    FUN = function(plot) {
                      plot+
                        scale_y_continuous(limits = c(0, 105), expand = c(0,0)) +
                          theme(axis.text.x = element_text(size = rel(1.2)),
                                plot.title = element_text(size = rel(1.35)),
                                legend.title = element_text(size = rel(1.6)),
                                legend.text = element_text(size = rel(1.6)),
                                legend.spacing = unit(1.5, "cm"))
                    })
```

```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```

```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```

```
fig1 <- ggpubr::ggarrange(plotlist = plotList1, labels = LETTERS[1:4], ncol = 2, nrow = 2,
                  common.legend = TRUE, align = "h")
ggsave("fig1.pdf", fig1, path = here("figures/"), width = 10.5, height = 7)
```

```
## composite plot 2 - descriptives
plotList2 <- lapply(X = list(q2, q4, q6, q7a, q7b, q8, q14,
                             q15a_new, q11, q15b, q17, q18b),
                    ## this plot requires a different title size
                    FUN = function(plot) {
                      plot +
                          theme(axis.text.x = element_text(size = rel(1.4)),
                                axis.text.y = element_text(size = rel(1.2)),
                                axis.title = element_text(size = rel(1.4)),
                                plot.title = element_text(size = rel(2)),
                                legend.title = element_text(size = rel(2.2)),
                                legend.text = element_text(size = rel(2.2)),
                                legend.spacing = unit(1.5, "cm"))
                    })
names <- LETTERS[1:length(plotList2)]
fig2 <- ggpubr::ggarrange(plotlist = plotList2, labels = names,
                          ncol = 3, nrow = 4, common.legend = TRUE, align = "h",
                          font.label = list(size = 18, color = "black", face = "bold",
                                            family = NULL))
## fig2
scale <- 0.94
ggsave("fig2.pdf", fig2, path = here("figures/"), width = scale*17, height = scale*22)
```

Descriptives

Show descriptives grouped by journal, then ungrouped.

```
# detailed per-journal descriptives
journal_describe <- sim_res_fac %>%
  split(.$journal) %>%
  purrr::map(~Hmisc::describe(.x))


# In a tidy way for long summary table
summary_vars <- c(
  "reviewer", "simstudy_q1",
  "nsimstudies_q2", "whichsim", "aimsdefined_q3",
  "dgptype_q4",
  # "dgpparameters_q5",
  # "nconds_q6",
  "factorsvaried_q7", "dgmfactorial_q7",
  # "nsim_q8",
  "nsimjustified_q9",
  "estimandstated_q10",
  # "nestimands_q11",
```

```
  "estimandsagg_q12", "truetheta_q13",
  "nmethods_q14",
  "target_q15",
  "pmconvergence_q15", "pmbias_q15", "pmempse_q15", "pm_r_mse_q15",
  "pmcover_q15", "pmtypeierror_q15", "pmpower_q15", "pmciwidth_q15",
  "pmsclear_q15", "mcerrors_q16",
  "resultsfigure_q17", "resultstable_q17", "resultstext_q17",
  "resultsother_q17", "software_1_q18", "software_2_q18",
  "software_3_q18", "software", "userwritten_q19",
  "codeprovided_q20",  "seedprovided_q21",
  "compenvironment_q22", "compos_q23",
  "coding_confidence"
)


sim_res_fac %>%
  as.data.frame() %>%
  group_by(journal) %>%
  pivot_longer(cols = starts_with("software"),
               names_to = NULL,
               values_to = "software",
               values_drop_na = TRUE) %>%
  mutate(software = as.factor(software)) %>%
  pivot_longer(cols = any_of(summary_vars),
               names_to = "col", values_to = "response") %>%
  select(doi, journal, col, response) %>%
  # get distinct responses per paper %>%
  group_by(doi, col) %>%
  distinct(response, .keep_all = TRUE) %>%
  ungroup() %>%
  group_by(journal, col) %>%
  count(response) %>%
  # filter(!is.na(response)) %>%
  # View()
  pivot_wider(id_cols = c(col, response),
              names_from = journal,
              values_from = n,
              values_fill = 0) %>%
  arrange(factor(col, levels = summary_vars)) %>%
  rowwise() %>%
  mutate(Sum = sum(BRM, MBR, PM)) %>%
  knitr::kable("latex", longtable = TRUE, caption = "Grouped by Journal") %>%
  kableExtra::column_spec(1, bold = TRUE) %>%
  kableExtra::collapse_rows(columns = 1:2, valign = "top")
```

Table 1: Grouped by Journal

col	response	BRM	MBR	PM	Sum
reviewer	BS	9	13	10	32
	FB	5	15	14	34
	SP	18	7	9	34
simstudy_q1	yes	32	35	33	100
	1	21	24	18	63

nsimstudies_q2

	2	5	8	11	24
	3	4	3	4	11
	5	1	0	0	1
whichsim	6	1	0	0	1
	1	25	23	25	73
	NA	7	11	7	25
	2	0	1	0	1
aimsdefined_q3	3	0	0	1	1
	no	1	1	1	3
	unclear	2	1	0	3
dgptype_q4	yes	29	33	32	94
factorsvaried_q7	parametric based on actual data	3	5	7	15
	parametric thin-air	29	29	25	83
	resampled	0	1	1	2
	1	6	9	13	28
	2	3	2	8	13
	3	9	12	1	22
	4	7	6	4	17
	5	3	3	3	9
	6	2	2	2	6
	7	2	0	2	4
dgmfactorial_q7	unclear	0	1	0	1
	fully-factorial	24	19	15	58
	one-at-a-time	1	4	0	5
	partially-factorial	7	12	18	37
nsimjustified_q9	no	28	34	30	92
estimandstated_q10	yes	4	1	3	8
	no	6	3	2	11
	unclear	3	3	0	6
	yes	23	29	29	81
estimandsagg_q12	not applicable	0	0	2	2
nmethods_q14	no	26	27	26	79
	unclear	1	3	0	4
	yes	5	5	7	17
truetheta_q13	estimated	2	3	2	7
	known	30	32	29	91
	not applicable	0	0	2	2
	1	7	10	7	24
	10	1	0	1	2
	11	2	0	2	4
	192	1	0	0	1
	2	7	10	6	23
	3	5	6	7	18
	4	2	3	2	7
	5	4	0	1	5
	6	2	2	3	7
	7	1	0	1	2
	10?	0	1	0	1
	14	0	1	0	1
	8	0	1	0	1
	9	0	1	1	2
	11+	0	0	1	1
	13	0	0	1	1
	design	1	1	2	4

target_q15

	estimation	14	20	22	56
	estimation, testing	5	4	0	9
	model selection	2	4	1	7
	other	2	0	3	5
	prediction	2	1	1	4
	testing	6	2	4	12
	design, estimation	0	1	0	1
pmconvergence_q15	estimation, model selection	0	2	0	2
	no	28	28	23	79
	yes	4	7	8	19
	unclear	0	0	2	2
pmbias_q15	no	20	12	23	55
	yes	12	23	10	45
pmempse_q15	no	30	31	24	85
	yes	2	4	9	15
pm_r_mse_q15	no	21	18	22	61
	yes	11	17	11	39
pmcover_q15	no	24	25	25	74
	yes	8	10	8	26
pmtypeierror_q15	no	22	28	23	73
	yes	10	7	10	27
pmpower_q15	no	25	28	23	76
	yes	7	7	10	24
pmciwidth_q15	no	30	30	32	92
mcerrors_q16	yes	2	5	1	8
pmsclear_q15	no	2	1	2	5
	unclear	3	0	2	5
	yes	27	34	29	90
	boxplot	1	3	2	6
	nowhere	25	27	26	78
	plot	1	0	0	1
	plot, table	1	0	0	1
	standard deviation	3	1	0	4
	text	1	0	1	2
	figures	0	1	0	1
	reference intervals	0	1	0	1
	tables, text	0	2	0	2
	quantile plots	0	0	1	1
	ranges	0	0	1	1
	table	0	0	2	2
resultsfigure_q17	no	6	16	17	39
	yes	26	19	16	61
resultstable_q17	no	14	7	9	30
	yes	18	28	24	70
resultstext_q17		32	33	31	96
	no	0	2	2	4
resultsother_q17		28	31	29	88
	yes	4	4	4	12
	C++	1	0	0	1
	JAGS	1	0	1	2
	Java	1	0	0	1
	MATLAB	3	2	1	6
	Mplus	2	2	3	7
	Python	3	1	0	4

	R	22	26	29	77
	SAS	2	2	0	4
	unclear	3	4	2	9
	Stan	0	1	1	2
	Ox	0	0	1	1
	Stata	0	0	1	1
userwritten_q19	flexMIRT	0	0	1	1
	no	18	19	10	47
	unclear	3	2	3	8
	yes	11	14	20	45
codeprovided_q20	accessible online	16	6	14	36
	not accessible	16	29	19	64
seedprovided_q21	not found	24	30	25	79
compenvironment_q22	yes	8	5	8	21
	fully	1	0	1	2
	minimal	14	22	23	59
	no	8	9	6	23
compos_q23	partially	9	4	3	16
	fully	2	1	1	4
	no	29	33	31	93
coding_confidence	partially	1	1	1	3
	great	20	18	24	62
	medium	9	15	8	32
	poor	3	2	1	6

```
# Ungrouped and with proportions
sim_res_fac %>%
  as.data.frame() %>%
  # pivot_longer(cols = starts_with("software"),
  #              names_to = NULL,
  #              values_to = "software",
  #              values_drop_na = TRUE) %>%
  # mutate(software = as.factor(software)) %>%
  pivot_longer(cols = any_of(summary_vars),
               names_to = "col", values_to = "response") %>%
  select(col, response) %>%
  group_by(col) %>%
  count(response) %>%
  arrange(factor(col, levels = summary_vars)) %>%
  knitr::kable("latex", longtable = TRUE, caption = "Ungrouped") %>%
  kableExtra::column_spec(1, bold = TRUE) %>%
  kableExtra::collapse_rows(columns = 1:2, valign = "top")
```

Table 2: Ungrouped

col	response	n
reviewer	BS	32
	FB	34
	SP	34
simstudy_q1	yes	100
	1	63
	2	24
	3	11

nsimstudies_q2

whichsim	5	1
	6	1
aimsdefined_q3	1	73
	2	1
	3	1
	NA	25
dgptype_q4	no	3
	unclear	3
	yes	94
factorsvaried_q7	parametric based on actual data	15
	parametric thin-air	83
	resampled	2
dgmfactorial_q7	1	28
	2	13
	3	22
	4	17
	5	9
	6	6
	7	4
	unclear	1
nsimjustified_q9	fully-factorial	58
	one-at-a-time	5
	partially-factorial	37
estimandstated_q10	no	92
	yes	8
estimandsagg_q12	no	11
	not applicable	2
	unclear	6
	yes	81
nmethods_q14	no	79
	unclear	4
truetheta_q13	yes	17
	estimated	7
	known	91
	not applicable	2
	1	24
	10	2
	10?	1
	11	4
	11+	1
	13	1
	14	1
	192	1
	2	23
	3	18
	4	7
	5	5
	6	7
	7	2
	8	1
	9	2
	design	4
	design, estimation	1

target_q15

	estimation	56
	estimation, model selection	2
	estimation, testing	9
	model selection	7
	other	5
	prediction	4
pmconvergence_q15	testing	12
	no	79
	unclear	2
	yes	19
pmbias_q15	no	55
	yes	45
pmempse_q15	no	85
	yes	15
pm_r_mse_q15	no	61
	yes	39
pmcover_q15	no	74
	yes	26
pmtypeierror_q15	no	73
	yes	27
pmpower_q15	no	76
	yes	24
pmciwidth_q15	no	92
mcerrors_q16	yes	8
pmsclear_q15	no	5
	unclear	5
	yes	90
	boxplot	6
	figures	1
	nowhere	78
	plot	1
	plot, table	1
	quantile plots	1
	ranges	1
	reference intervals	1
	standard deviation	4
	table	2
	tables, text	2
	text	2
resultsfigure_q17	no	39
	yes	61
resultstable_q17	no	30
	yes	70
resultstext_q17	no	4
	yes	96
resultsother_q17	no	88
	yes	12
	C++	1
	JAGS	1
	Java	1
	MATLAB	5
	Mplus	1
	Ox	1
	Python	2

software_2_q18	R	76
	SAS	3
	unclear	9
	JAGS	1
	MATLAB	1
	Mplus	6
	Python	2
	R	1
	SAS	1
	Stan	2
	Stata	1
	flexMIRT	1
	NA	84
software_3_q18		100
userwritten_q19	no	47
	unclear	8
	yes	45
codeprovided_q20	accessible online	36
	not accessible	64
seedprovided_q21	not found	79
compenvironment_q22	yes	21
	fully	2
	minimal	59
	no	23
compos_q23	partially	16
	fully	4
	no	93
coding_confidence	partially	3
	great	62
	medium	32
	poor	6

Analyses of individual questions:

```
# Q8:
sim_res_num$nsim_q8 %>%
  table()
```

```
## .
##    1   10   25   30   50   60  100  200  400  500  800 1000 2000
##    1    2    1    3    1    1   17    5    1   14    1   29    3
## 2500 3000 5000 10000 1e+06
##    1    1    5    7    1
```

```
# Q15a:
sim_res_fac %>%
    separate_wider_delim(target_q15,
                         delim = ",",
                         names_sep = "_",
                         too_few = "align_start") %>%
    pivot_longer(cols = contains("target"),
                 names_to = NULL,
                 values_to = "target",
```

```
              values_drop_na = TRUE) %>%
    mutate(target = str_trim(target)) %>%
    mutate(target = as.factor(target)) %>%
    count(target) %>%
    arrange(desc(n))
```

```
## # A tibble: 6 x 2
##   target              n
##   <fct>           <int>
## 1 estimation         68
## 2 testing            21
## 3 model selection     9
## 4 design              5
## 5 other               5
## 6 prediction          4
```

```
# Q15:
sim_res_fac %>%
    group_by(journal) %>%
    summarise("Convergence" = sum(pmconvergence_q15 == "yes"),
              "Bias" = sum(pmbias_q15 == "yes"),
              "Empirical SE" = sum(pmempse_q15 == "yes"),
              "(R)MSE" = sum(pm_r_mse_q15 == "yes"),
              "Coverage" = sum(pmcover_q15 == "yes"),
              "Type I error rate" = sum(pmtypeierror_q15 == "yes"),
              "Power" = sum(pmpower_q15 == "yes"),
              "CI width" = sum(pmciwidth_q15 == "yes"),
              "Other" = sum(!is.na(pmother_q15))) %>%
    gather(key = "PM", value = "count", "Convergence", "Bias", "(R)MSE",
           "Empirical SE", "Coverage", "Type I error rate",
           "Power", "CI width", "Other") %>%
    bind_rows(q15_other) %>%
    group_by(PM) %>%
    dplyr::summarize(sum = sum(count)) %>%
    arrange(desc(sum))
```

```
## # A tibble: 10 x 2
##    PM                   sum
##    <chr>              <int>
##  1 Other                125
##  2 Bias                  63
##  3 (R)MSE                39
##  4 Type I error rate     27
##  5 Coverage              26
##  6 Power                 24
##  7 Convergence           19
##  8 Empirical SE          15
##  9 CI width               8
## 10 Correlation            6
```

```
sessionInfo()
```

```
## R version 4.4.0 (2024-04-24 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 22631)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=German_Germany.utf8  LC_CTYPE=German_Germany.utf8
## [3] LC_MONETARY=German_Germany.utf8 LC_NUMERIC=C
## [5] LC_TIME=German_Germany.utf8
##
## time zone: Europe/Berlin
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] Hmisc_5.1-2      here_1.0.1       showtext_0.9-7   showtextdb_3.0
##  [5] sysfonts_0.8.9   kableExtra_1.4.0 knitr_1.46       forcats_1.0.0
##  [9] stringr_1.5.1    ggpubr_0.6.0     colorspace_2.1-0 ggplot2_3.5.1
## [13] tidyr_1.3.1      dplyr_1.1.4
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.5      xfun_0.44         htmlwidgets_1.6.4 rstatix_0.7.2
##  [5] vctrs_0.6.5       tools_4.4.0       generics_0.1.3    curl_5.2.1
##  [9] tibble_3.2.1      fansi_1.0.6       cluster_2.1.6     pkgconfig_2.0.3
## [13] data.table_1.15.4 checkmate_2.3.1   lifecycle_1.0.4   farver_2.1.2
## [17] compiler_4.4.0    textshaping_0.3.7 munsell_0.5.1     tinytex_0.51
## [21] carData_3.0-5     htmltools_0.5.8.1 yaml_2.3.8        htmlTable_2.4.2
## [25] Formula_1.2-5     pillar_1.9.0      car_3.1-2         rpart_4.1.23
## [29] abind_1.4-5       tidyselect_1.2.1  digest_0.6.35     stringi_1.8.4
## [33] purrr_1.0.2       labeling_0.4.3    cowplot_1.1.3     rprojroot_2.0.4
## [37] fastmap_1.2.0     grid_4.4.0        cli_3.6.2         magrittr_2.0.3
## [41] base64enc_0.1-3   utf8_1.2.4        broom_1.0.6       foreign_0.8-86
## [45] withr_3.0.0       scales_1.3.0      backports_1.4.1   rmarkdown_2.27
## [49] nnet_7.3-19       gridExtra_2.3     ggsignif_0.6.4    ragg_1.3.2
## [53] evaluate_0.23     viridisLite_0.4.2 rlang_1.1.3       glue_1.7.0
## [57] xml2_1.3.6        jsonlite_1.8.8    svglite_2.1.3     rstudioapi_0.16.0
## [61] R6_2.5.1          systemfonts_1.1.0
```