# Preliminary analysis

Samuel Pawel

11 August 2023

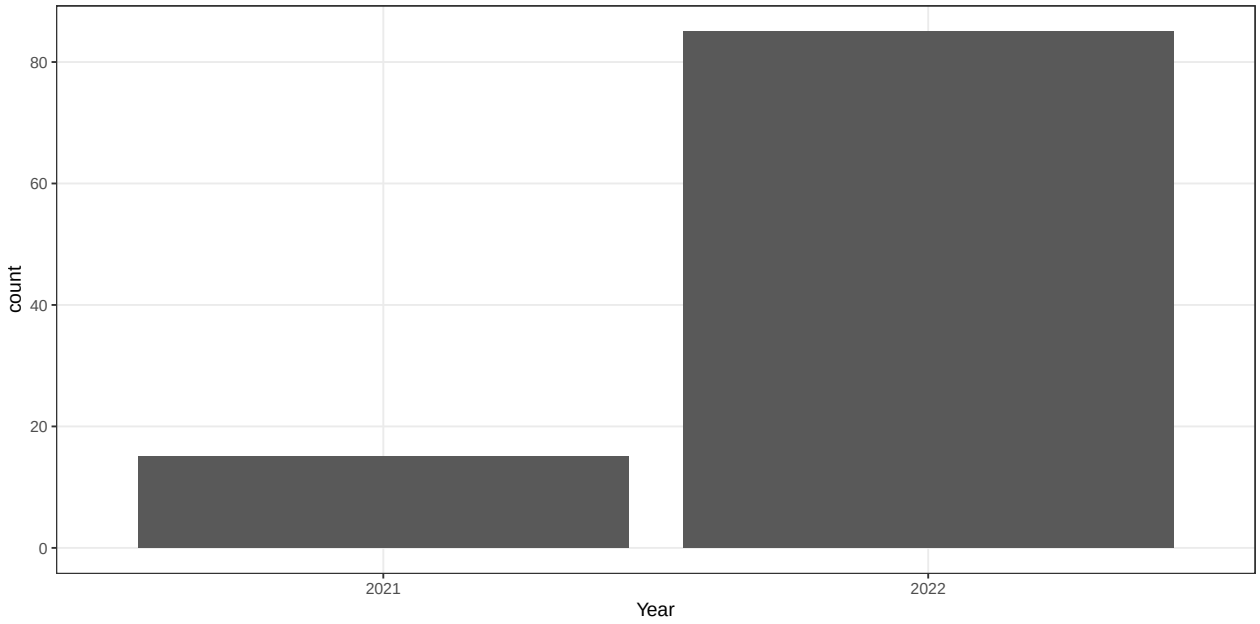
```
## data
sim_res_fac <- readRDS(file = "data/sim_res_fac.RDS")
sim_res_num <- readRDS(file = "data/sim_res_num.RDS")

## libraries
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

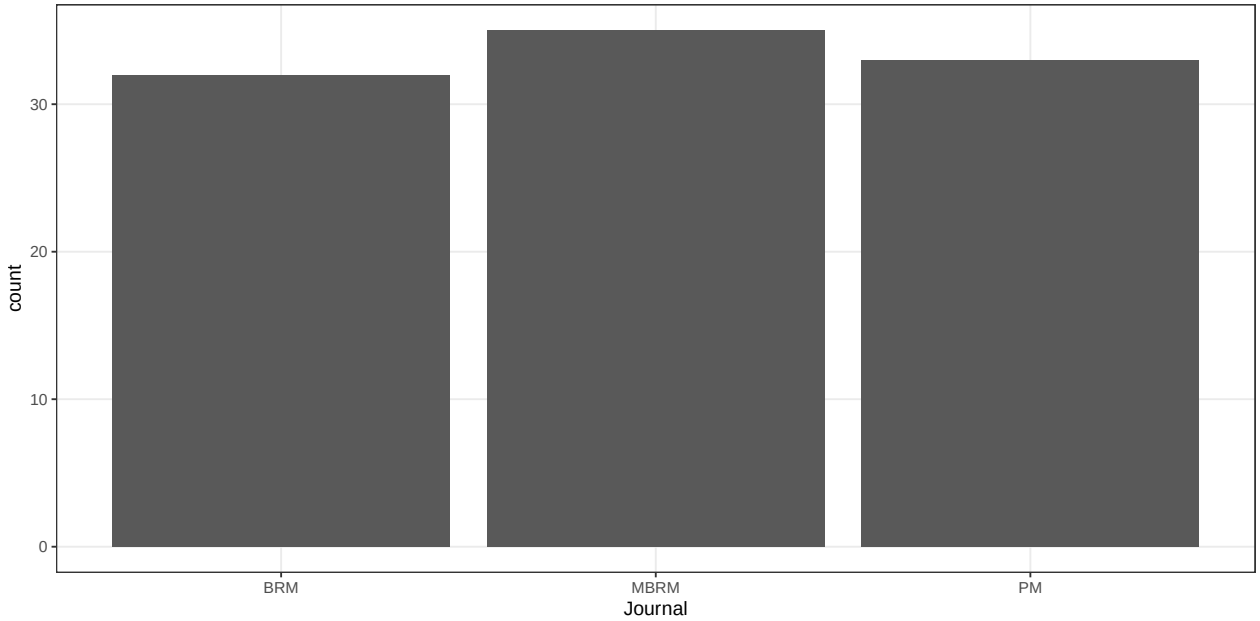
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

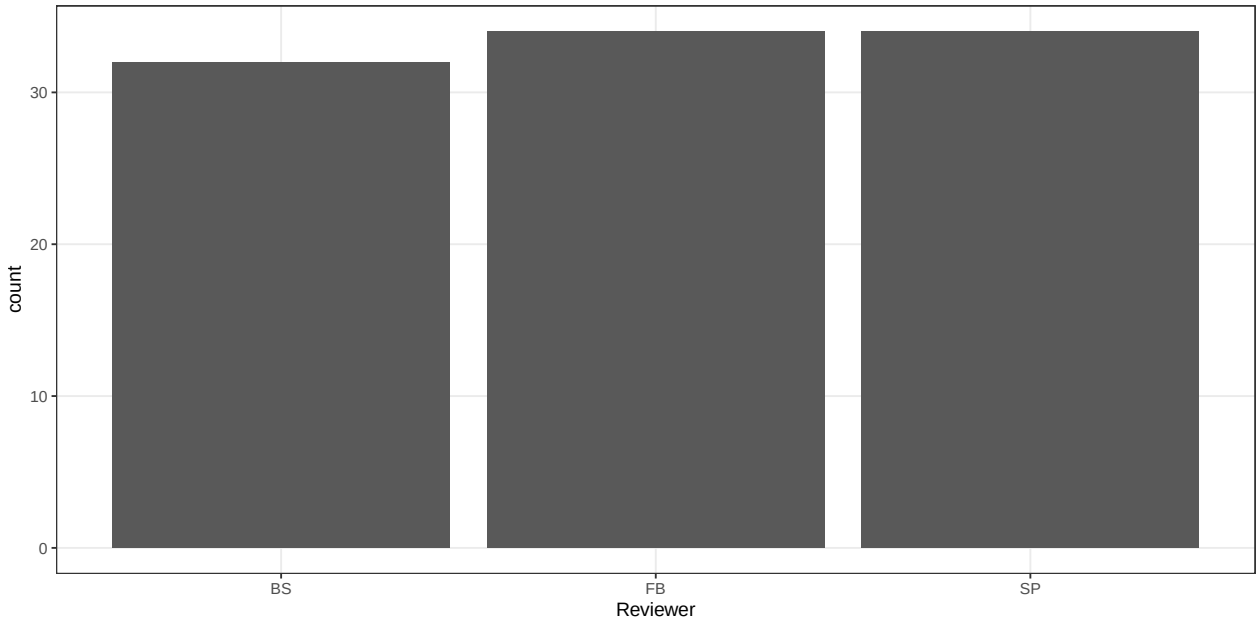
```
library(tidyr)
library(ggplot2)
library(ggpubr)
theme_set(theme_bw() +
          theme(legend.position = "top",
                panel.grid.minor = element_blank()))

## year
ggplot(data = sim_res_fac, aes(x = factor(year))) +
    geom_bar() +
    labs(x = "Year")
```



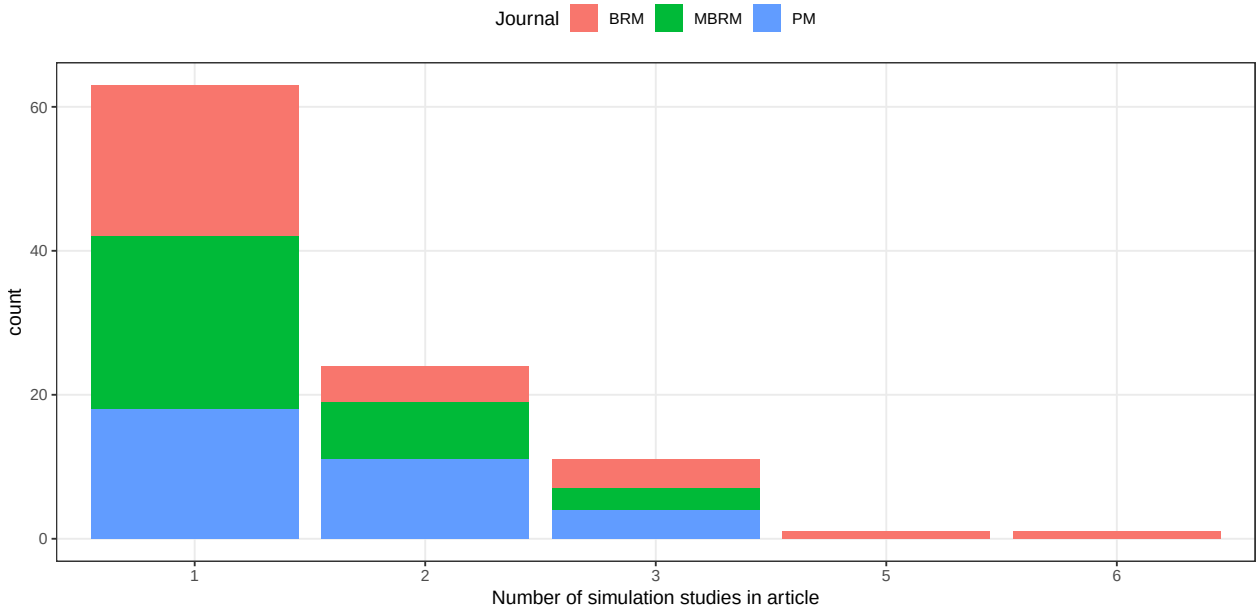

```
## journal
ggplot(data = sim_res_fac, aes(x = journal)) +
    geom_bar() +
    labs(x = "Journal")
```



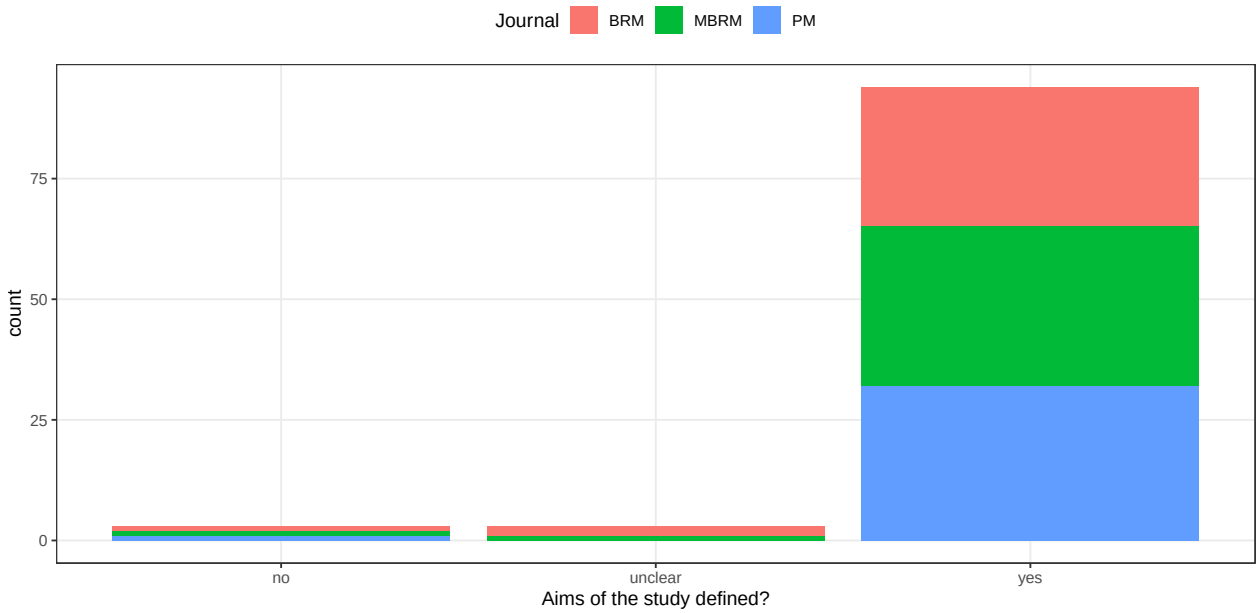

```
## reviewer
ggplot(data = sim_res_fac, aes(x = reviewer)) +
    geom_bar() +
    labs(x = "Reviewer")
```



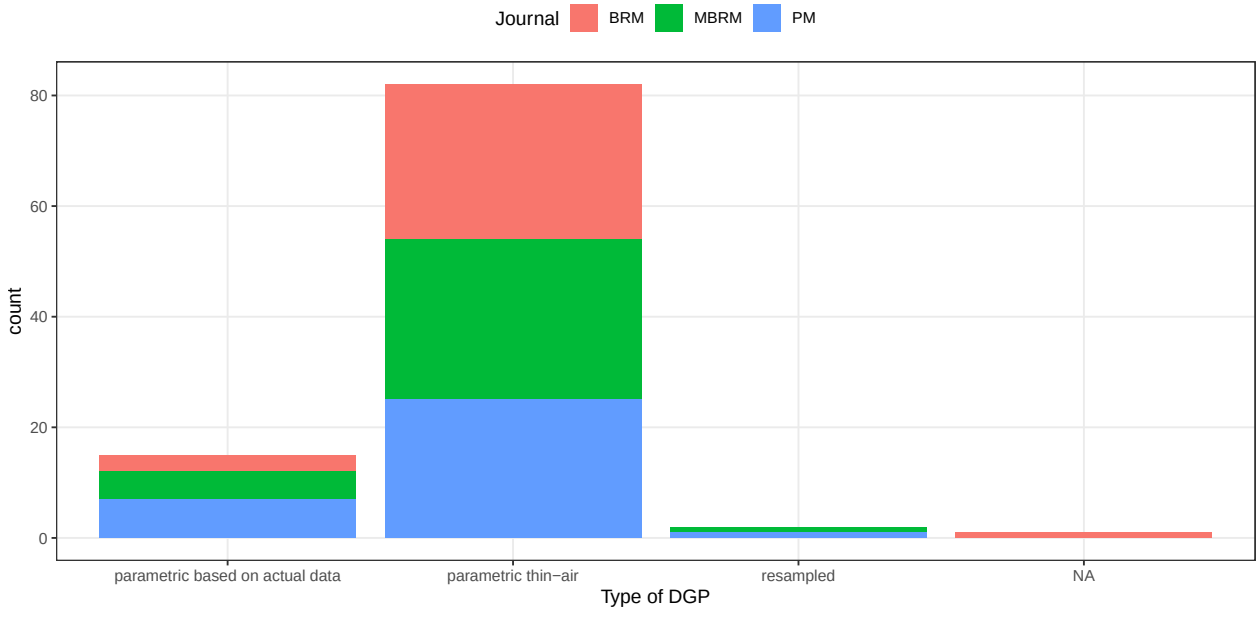

```
## Q2 number of simulation studies
ggplot(data = sim_res_fac, aes(x = nsimstudies_q2, fill = journal)) +
    geom_bar() +
    labs(x = "Number of simulation studies in article", fill = "Journal")
```



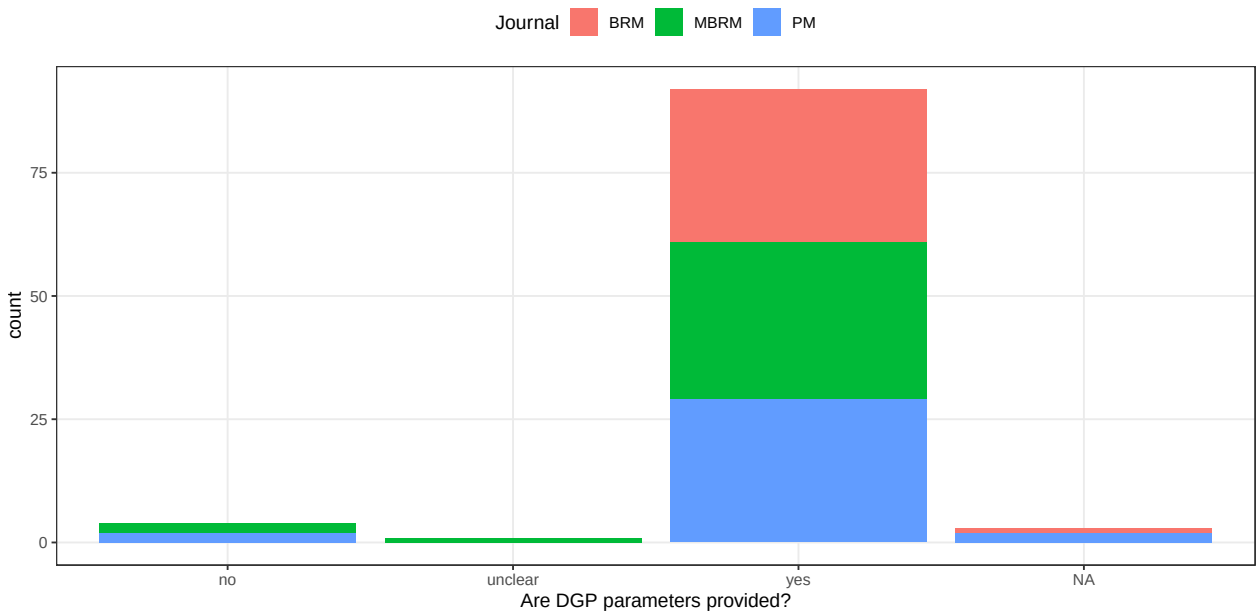

```
## Q3 are the aims of the study defined
ggplot(data = sim_res_fac, aes(x = aimsdefined_q3, fill = journal)) +
    geom_bar() +
    labs(x = "Aims of the study defined?", fill = "Journal")
```




```
## Q4 type of DGP
ggplot(data = sim_res_fac, aes(x = dgptype_q4, fill = journal)) +
    geom_bar() +
    labs(x = "Type of DGP", fill = "Journal")
```




```
## Q5 DGP parameters provided?
ggplot(data = sim_res_fac, aes(x = dgpparameters_q5, fill = journal)) +
    geom_bar() +
    labs(x = "Are DGP parameters provided?", fill = "Journal")
```

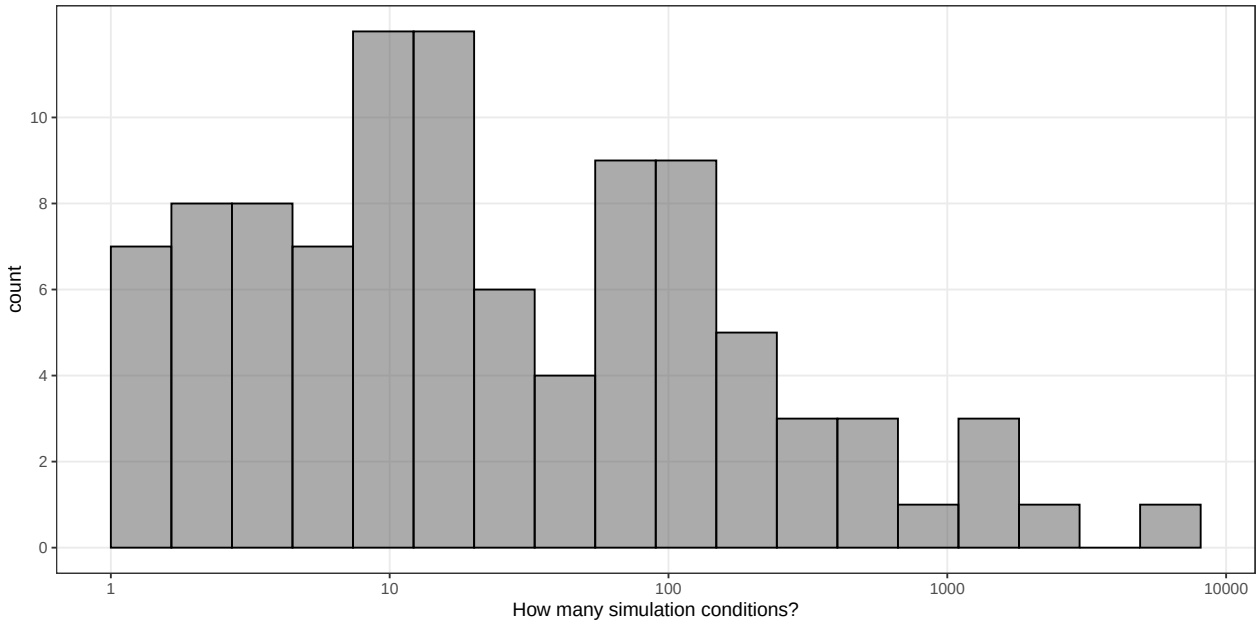



```
## Q6 How many conditions?
summary(sim_res_num$nconds_q6)
```

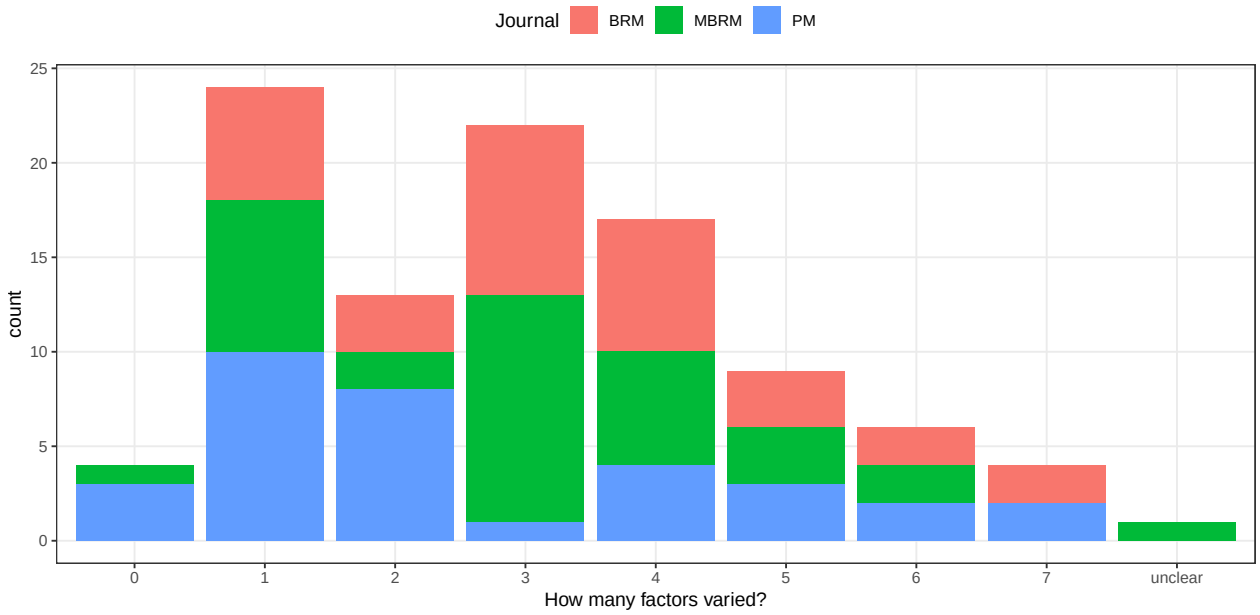
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     1.0     5.0    16.0   185.8    96.0  6000.0       1
```

```
breaks <- c(1, 10, 100, 1000, 10000)
ggplot(data = sim_res_num, aes(x = log(nconds_q6))) +
    geom_histogram(breaks = seq(0, log(10000), 0.5), col = 1, alpha = 0.5) +
    scale_x_continuous(breaks = log(breaks), labels = breaks) +
    scale_y_continuous(breaks = seq(0, 10, 2)) +
    labs(x = "How many simulation conditions?", fill = "Journal")
```

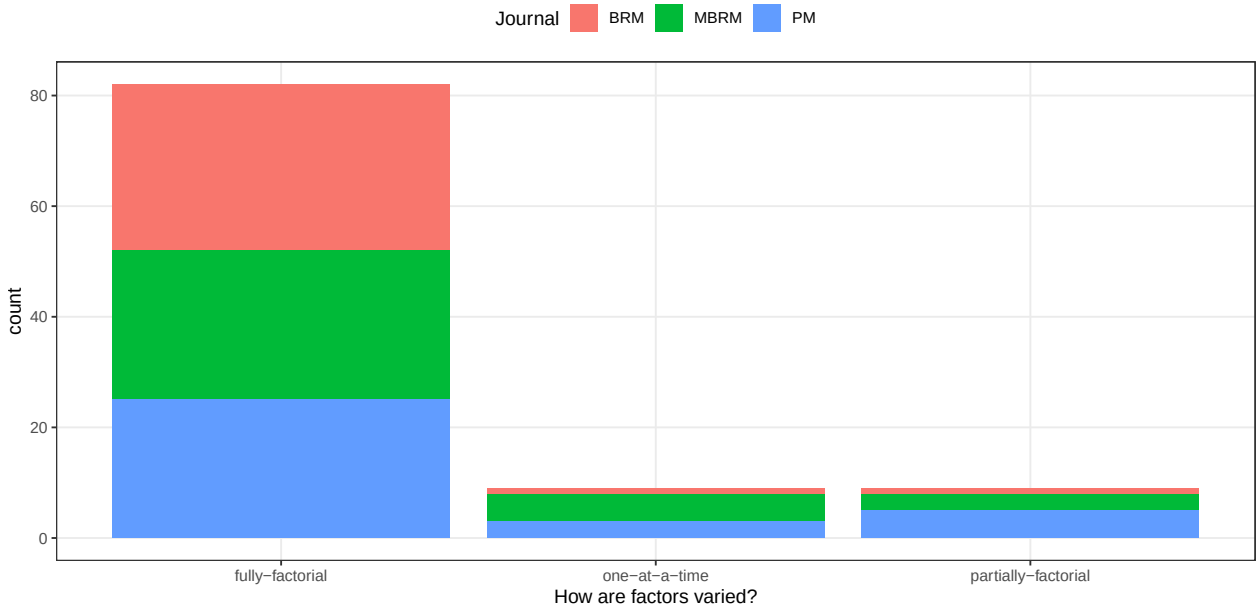
```
## Warning: Removed 1 rows containing non-finite values (`stat_bin()`).
```



```r
## Q7 How many factors?
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() +
    labs(x = "How many factors varied?", fill = "Journal")
```




```r
## Q7 Fully factorial?
ggplot(data = sim_res_fac, aes(x = dgmfactorial_q7, fill = journal)) +
    geom_bar() +
    labs(x = "How are factors varied?", fill = "Journal")
```

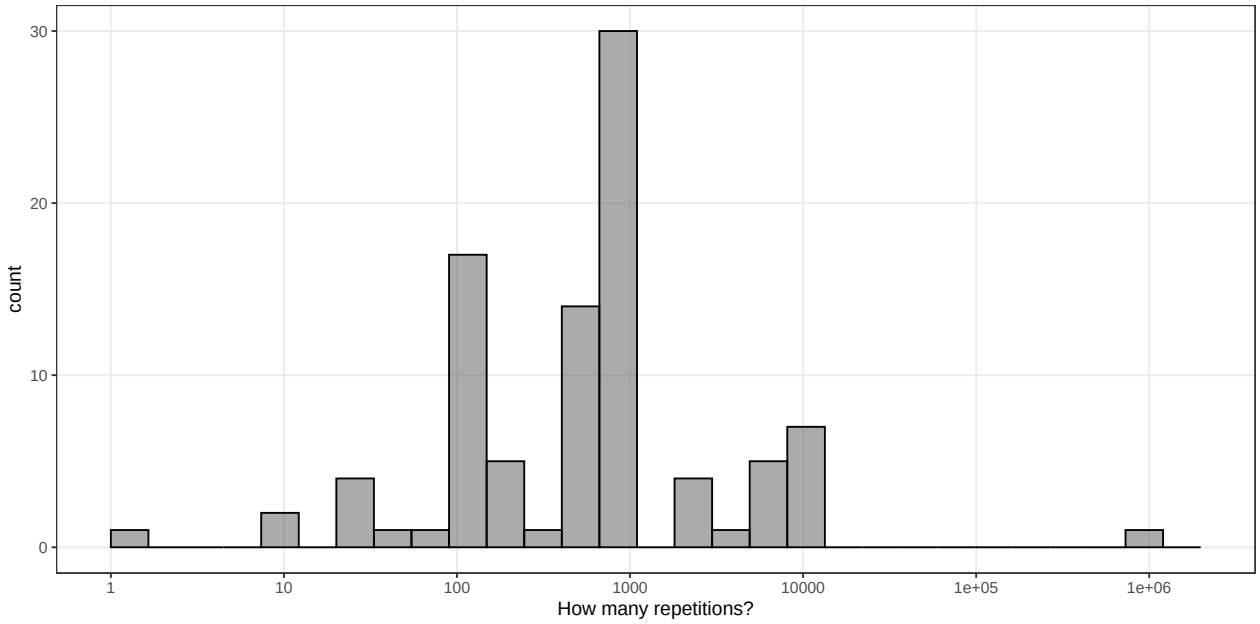



```r
## Q8 How many repetitions?
summary(sim_res_num$nsim_q8)
```

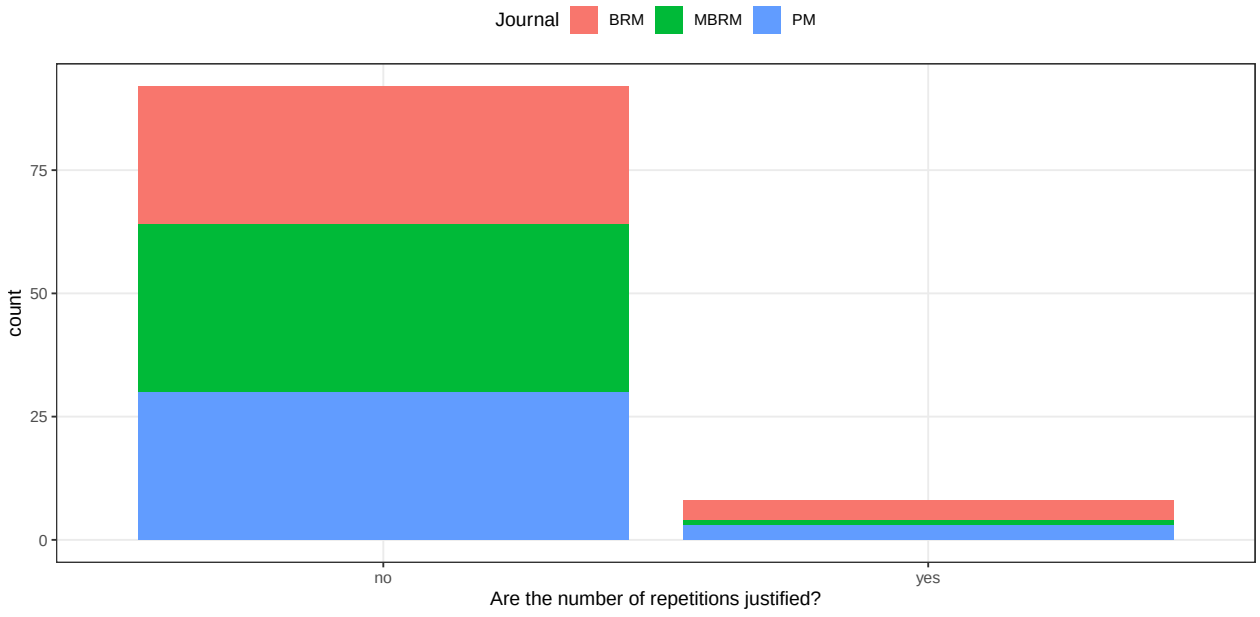
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       1     100     900   12198    1000 1000000       6
```

```
breaks <- c(1, 10, 100, 1000, 10000, 100000, 1000000)
ggplot(data = sim_res_num, aes(x = log(nsim_q8))) +
    geom_histogram(breaks = seq(0, log(2000000), 0.5), col = 1, alpha = 0.5) +
    labs(x = "How many repetitions?", fill = "Journal") +
    scale_x_continuous(breaks = log(breaks), labels = breaks)
```

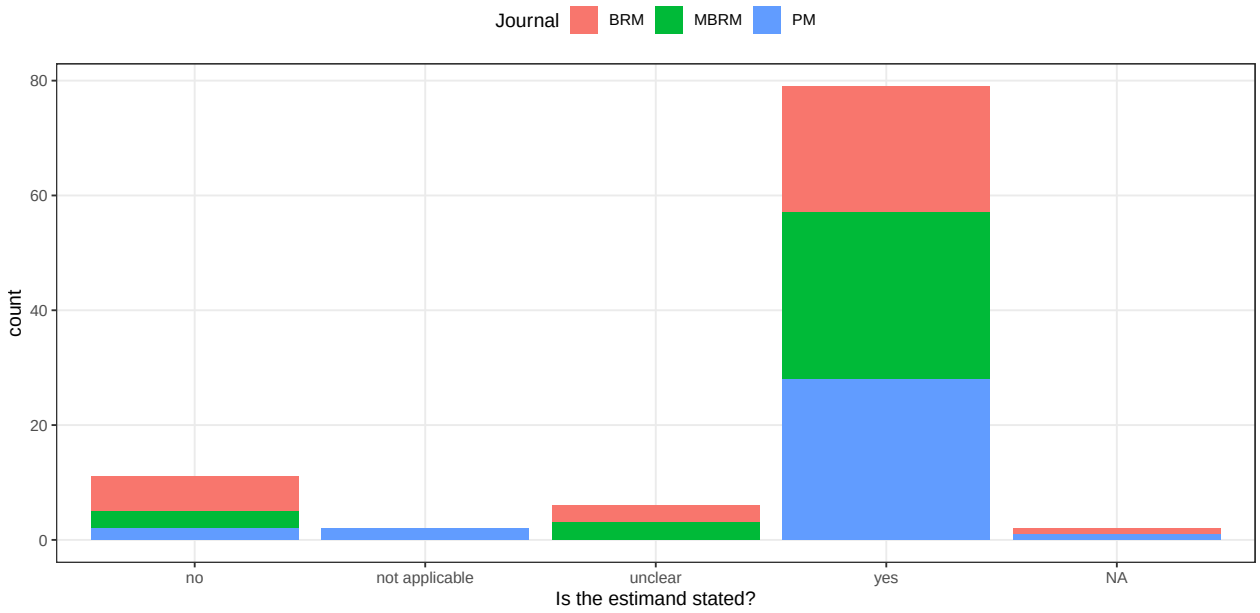
```
## Warning: Removed 6 rows containing non-finite values (`stat_bin()`).
```




```
## Q9 Are the number of repetitions justified?
ggplot(data = sim_res_fac, aes(x = nsimjustified_q9, fill = journal)) +
    geom_bar() +
    labs(x = "Are the number of repetitions justified?", fill = "Journal")
```



```r
## Q10 Is the estimand stated?
ggplot(data = sim_res_fac, aes(x = estimandstated_q10, fill = journal)) +
    geom_bar() +
    labs(x = "Is the estimand stated?", fill = "Journal")
```

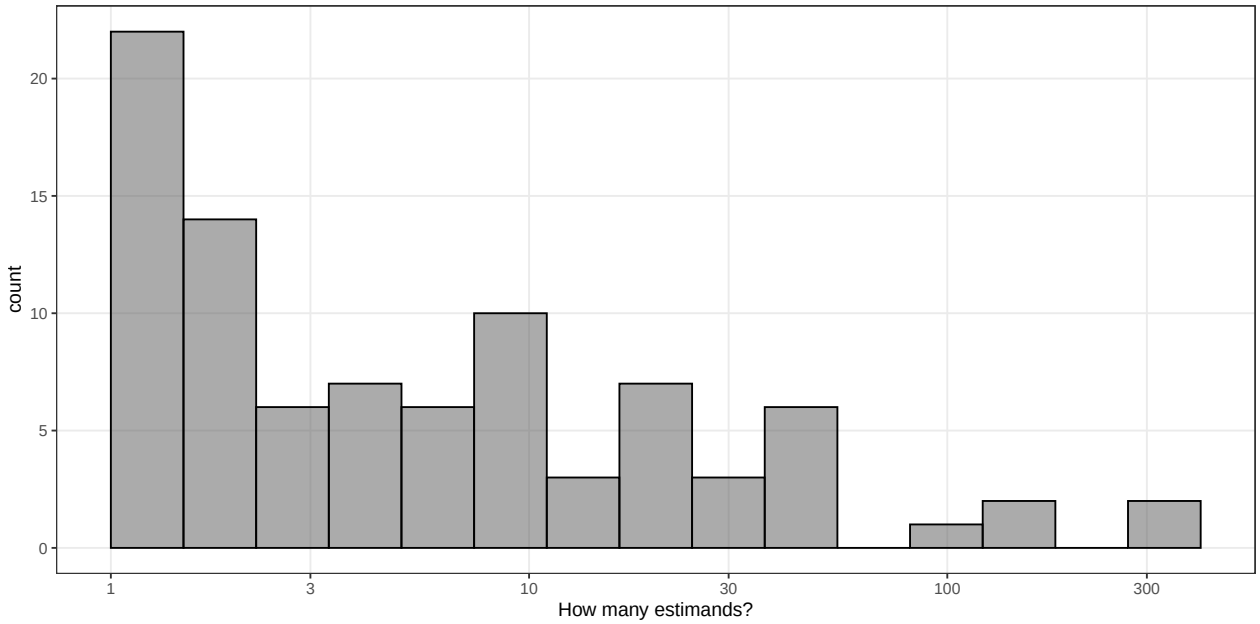



```r
## Q11 How many estimands?
summary(sim_res_num$nestimands_q11)
```

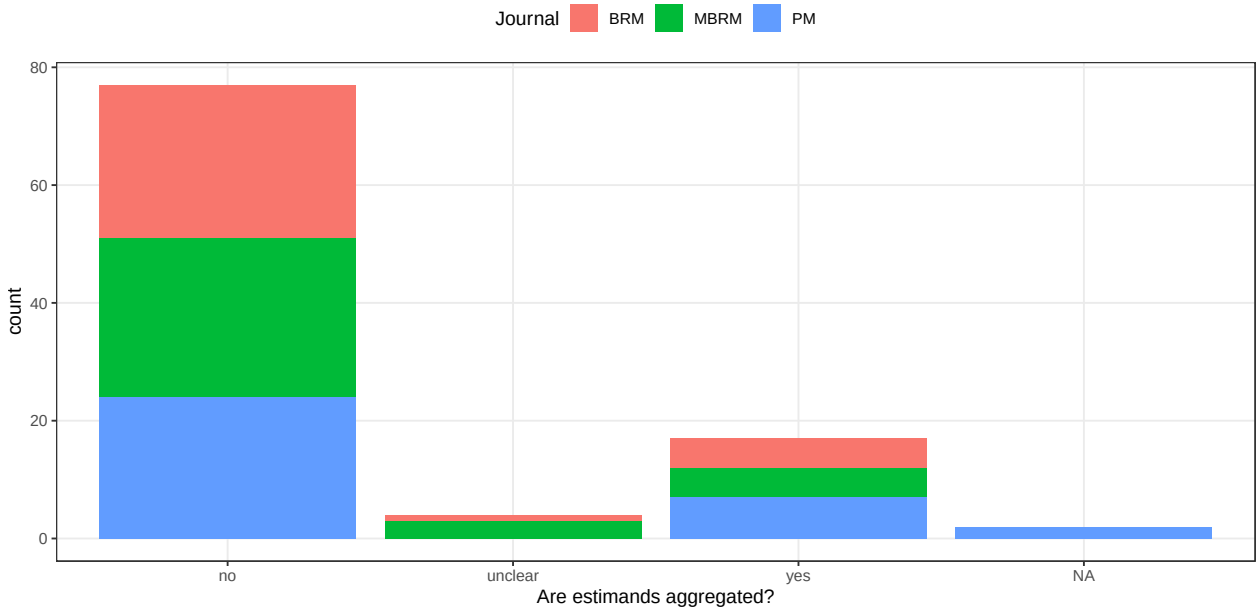
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1.00    2.00    4.00   20.11   15.00  384.00      11
```

```r
breaks <- c(1, 3, 10, 30, 100, 300)
ggplot(data = sim_res_num, aes(x = log(nestimands_q11))) +
    geom_histogram(breaks = seq(0, log(500), 0.4), col = 1, alpha = 0.5) +
    scale_x_continuous(breaks = log(breaks), labels = breaks) +
    labs(x = "How many estimands?", fill = "Journal")
```

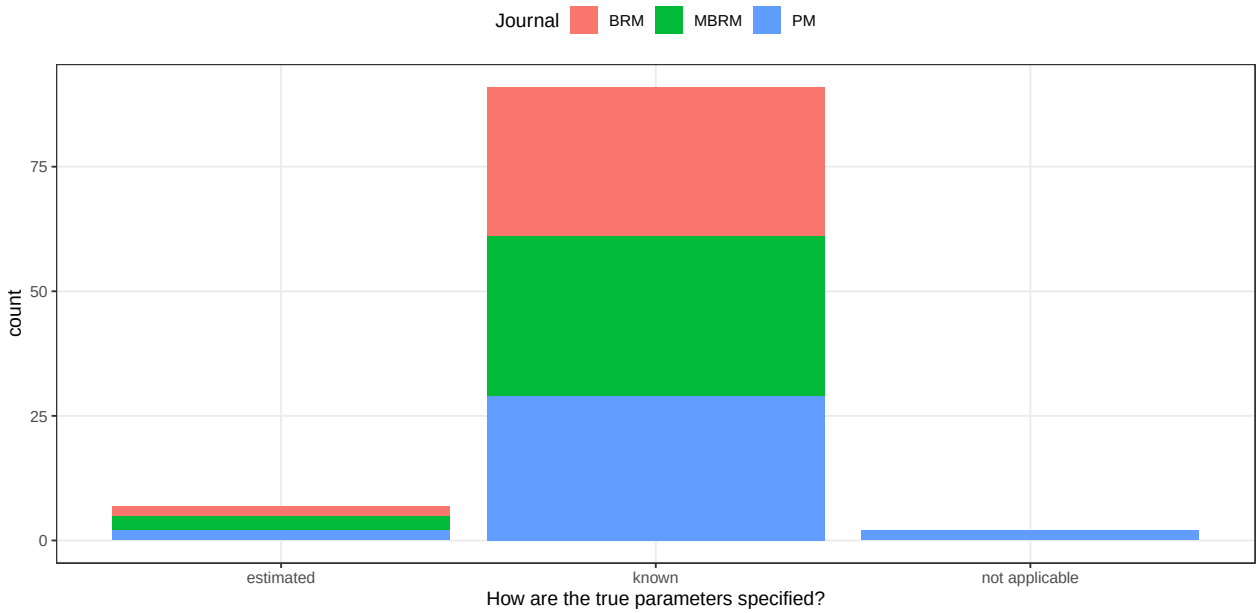
```
## Warning: Removed 11 rows containing non-finite values (`stat_bin()`).
```




```
## Q12 Are estimands aggregated?
ggplot(data = sim_res_fac, aes(x = estimandsagg_q12, fill = journal)) +
    geom_bar() +
    labs(x = "Are estimands aggregated?", fill = "Journal")
```




```
## Q13 How are the true parameters specified?
ggplot(data = sim_res_fac, aes(x = truetheta_q13, fill = journal)) +
    geom_bar() +
    labs(x = "How are the true parameters specified?", fill = "Journal")
```

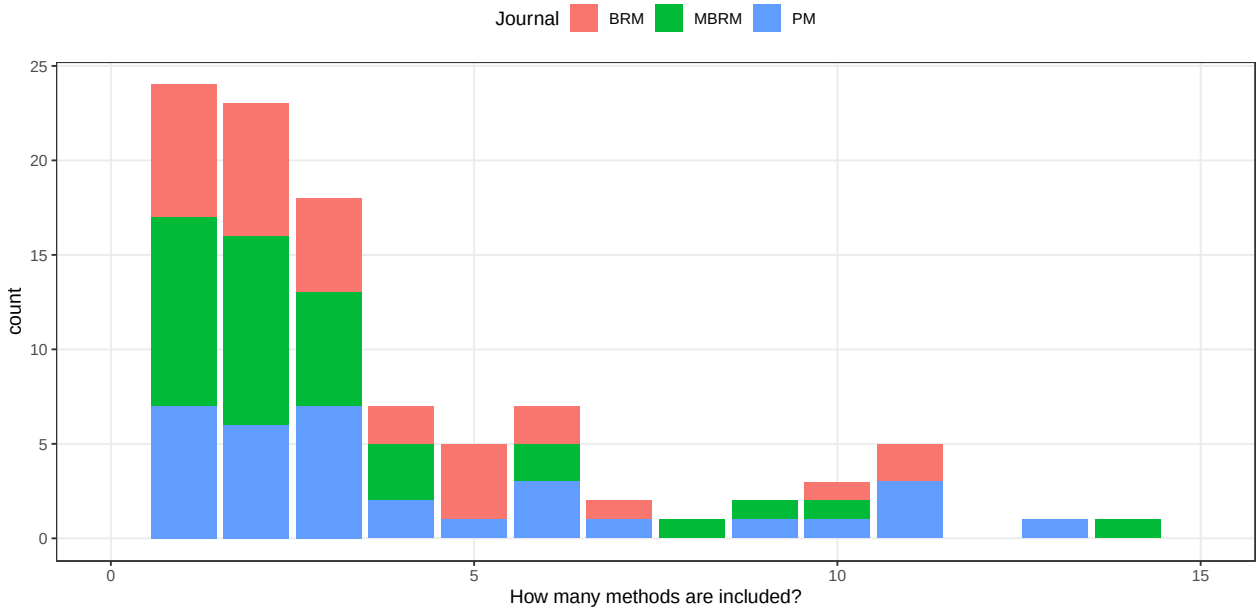



```
## Q14 How many methods are included?
summary(sim_res_num$nmethods_q14)
```

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

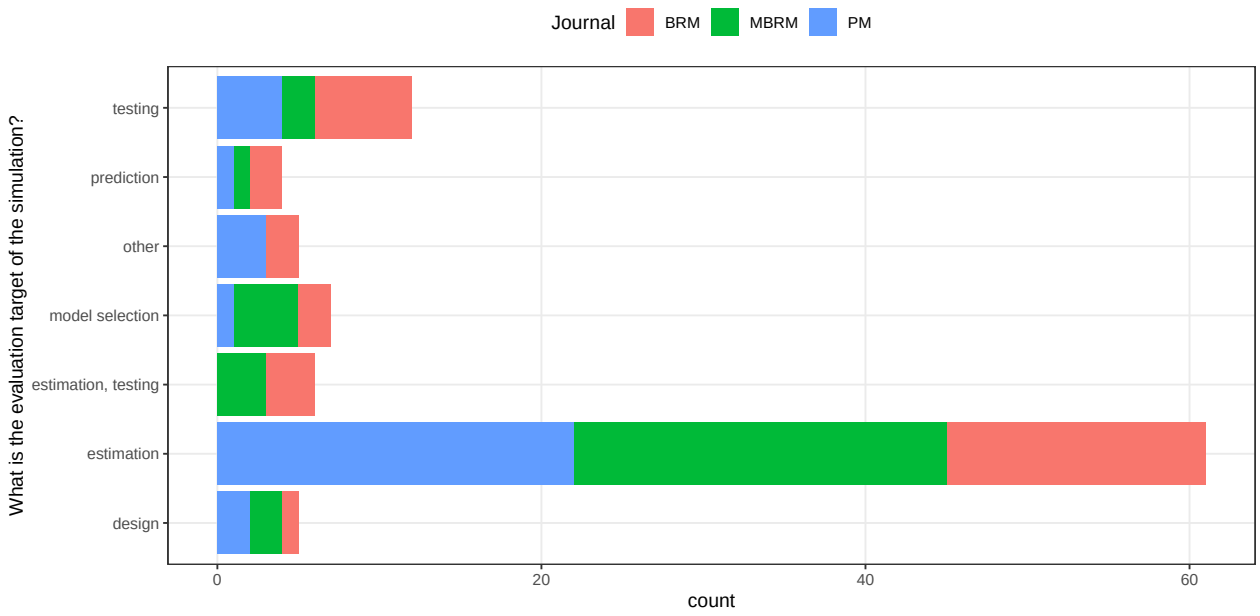
```
# HACK there is one study with 192 methods, let's exclude it for a moment
ggplot(data = sim_res_num, aes(x = nmethods_q14, fill = journal)) +
    geom_bar() +
    lims(x = c(0, 15)) +
    labs(x = "How many methods are included?", fill = "Journal")
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_count()`).
```

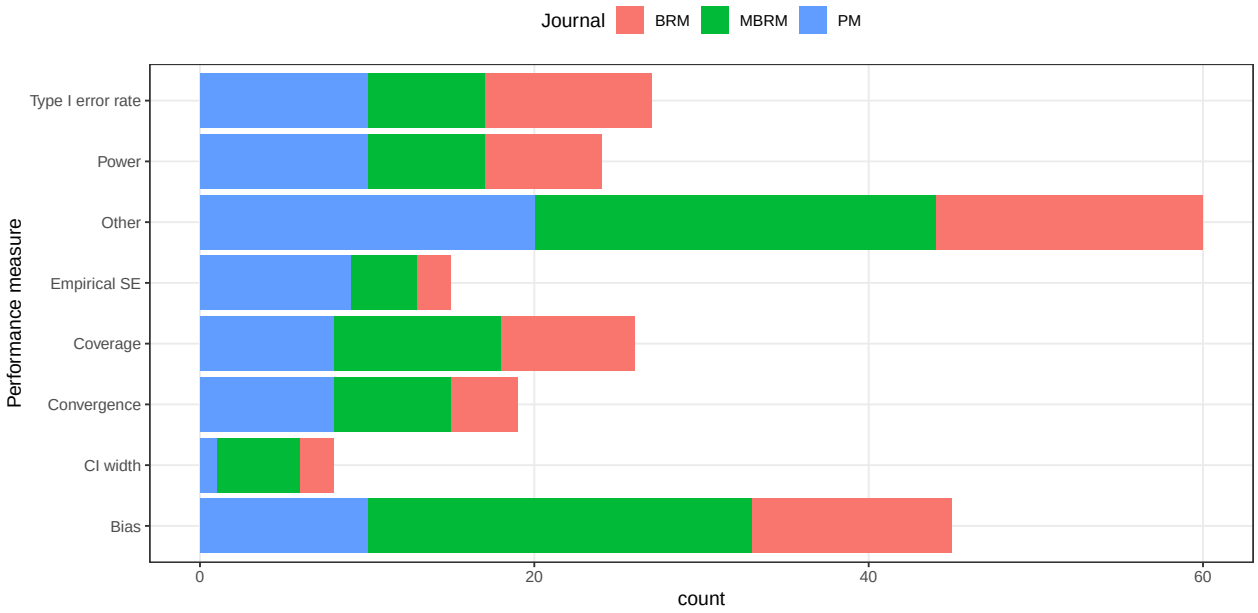



```
## Q15 What is the evaluation target of the simulation?
ggplot(data = sim_res_fac, aes(x = target_q15, fill = journal)) +
```

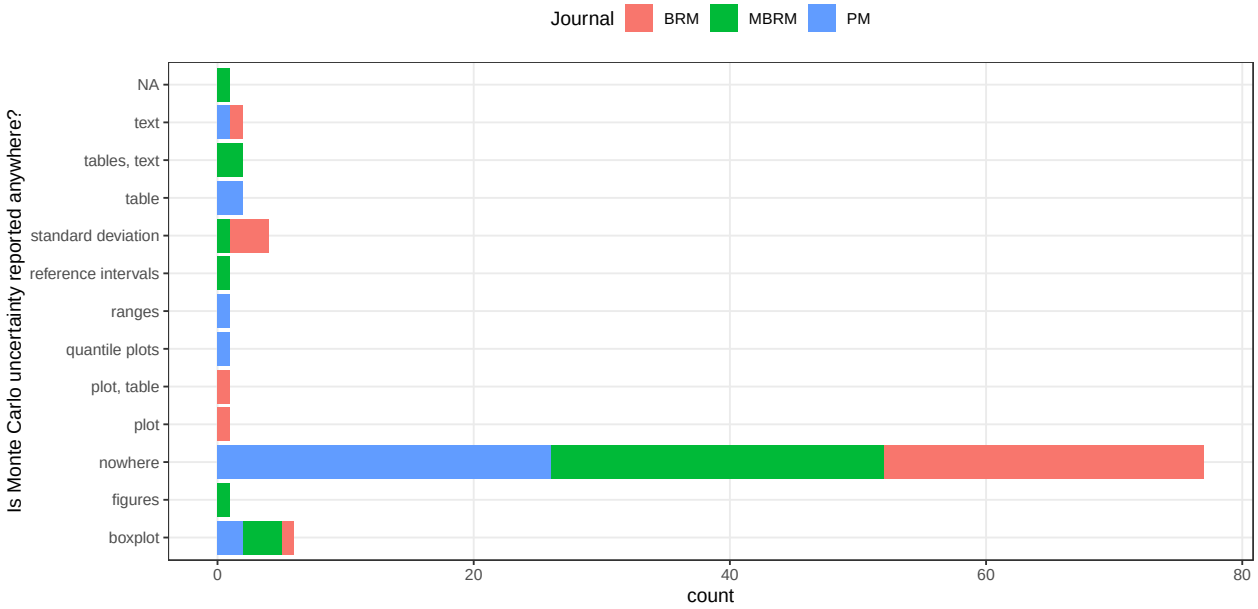
```
    geom_bar() +
    labs(x = "What is the evaluation target of the simulation?", fill = "Journal") +
    coord_flip()
```




```
## Q15 Which performance measures were used?
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", "Empirical SE",
           "Coverage", "Type I error rate", "Power", "CI width", "Other") %>%
    ggplot(aes(x = PM, y = count, fill = journal)) +
    geom_bar(stat = "identity") +
    labs(x = "Performance measure", fill = "Journal") +
    coord_flip()
```

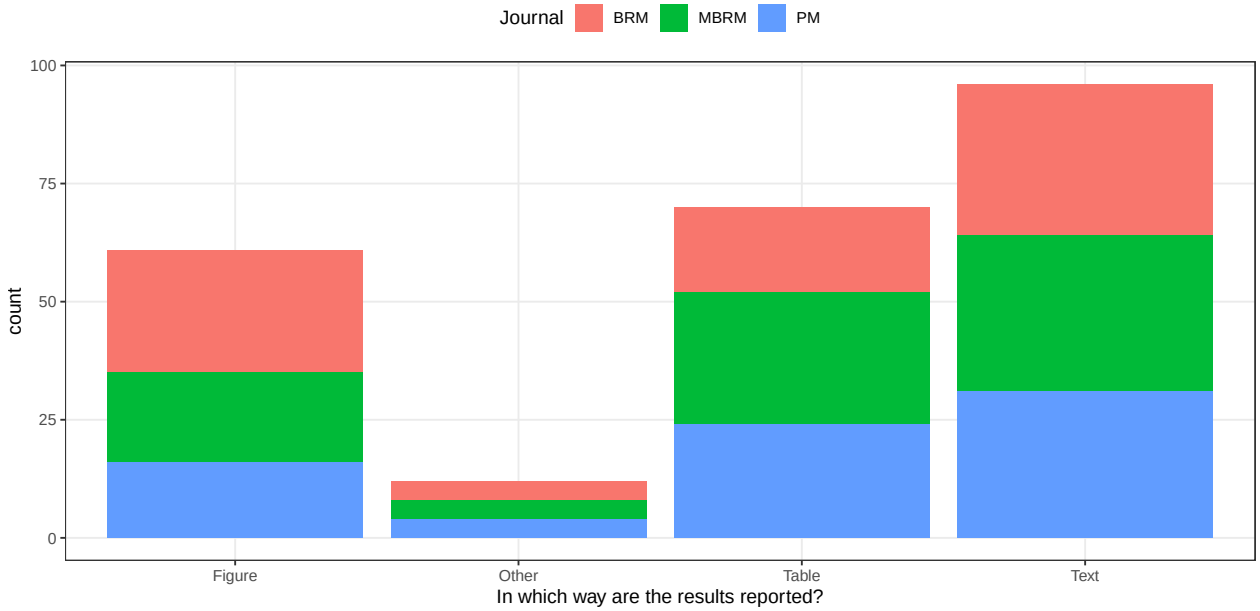



```
## Q16 Is Monte Carlo uncertainty reported anywhere?
ggplot(data = sim_res_fac, aes(x = mcerrors_q16, fill = journal)) +
    geom_bar() +
    labs(x = "Is Monte Carlo uncertainty reported anywhere?", fill = "Journal") +
    coord_flip()
```

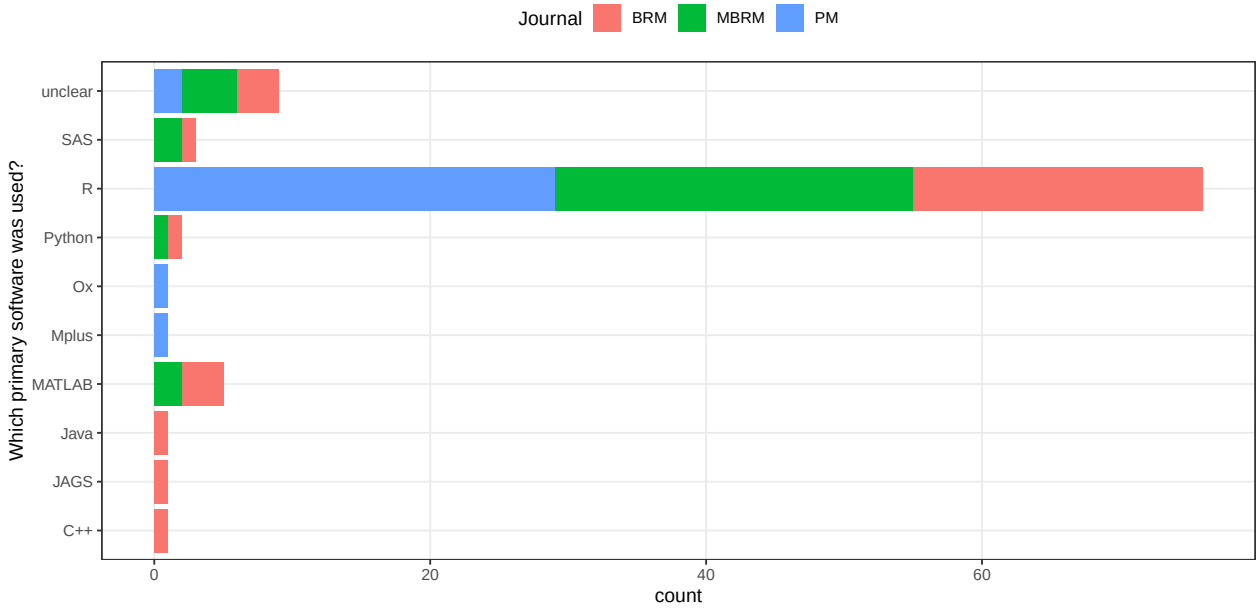



```
## Q17 In which way are the results reported?
sim_res_fac %>%
    group_by(journal) %>%
    summarise("Figure" = sum(resultsfigure_q17 == "yes"),
              "Table" = sum(resultstable_q17 == "yes"),
              "Text" = sum(resultstext_q17 == "yes"),
              "Other" = sum(resultsother_q17 == "yes")) %>%
    gather(key = "Type", value = "count", "Figure", "Table", "Text", "Other") %>%
    ggplot(aes(x = Type, y = count, fill = journal)) +
```

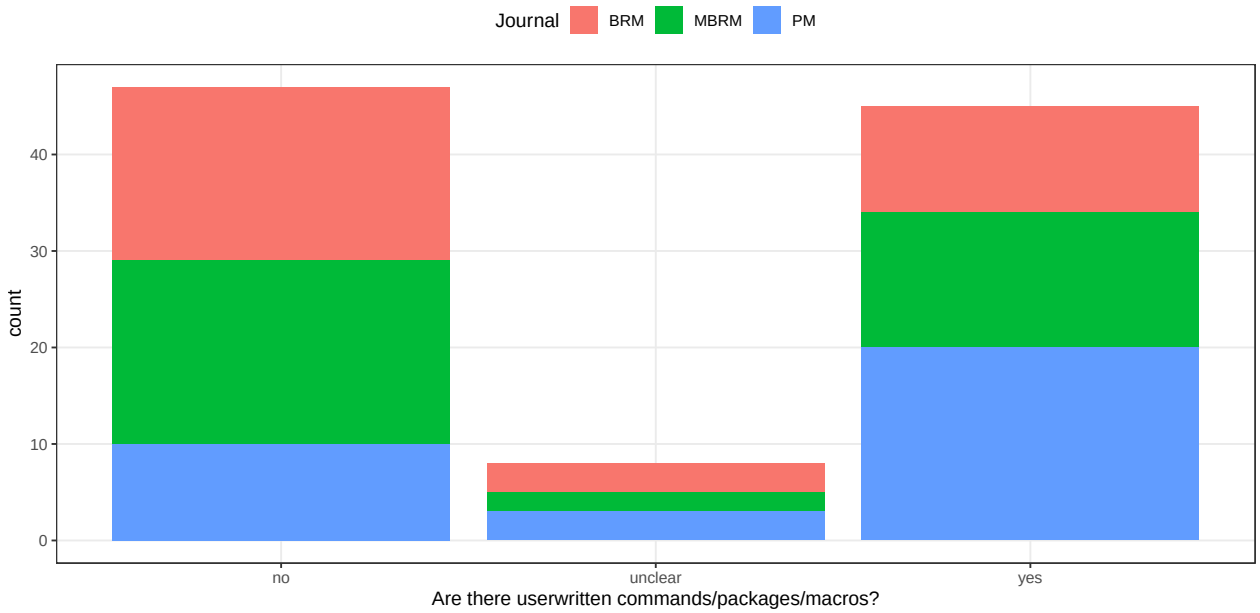
```
    geom_bar(stat = "identity") +
    labs(x = "In which way are the results reported?", fill = "Journal")
```



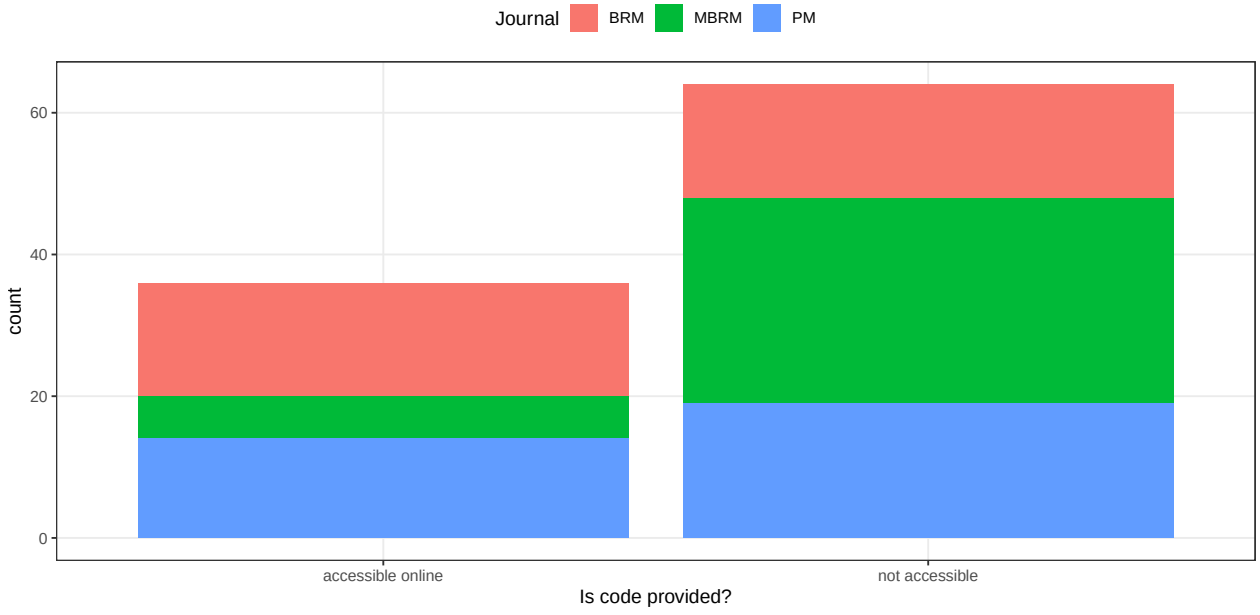

```
## Q18 Which software was used to conduct the simulation?
## TODO add also information from software_2_q18 and software_3_q18
ggplot(data = sim_res_fac, aes(x = software_1_q18, fill = journal)) +
    geom_bar() +
    labs(x = "Which primary software was used?", fill = "Journal") +
    coord_flip()
```



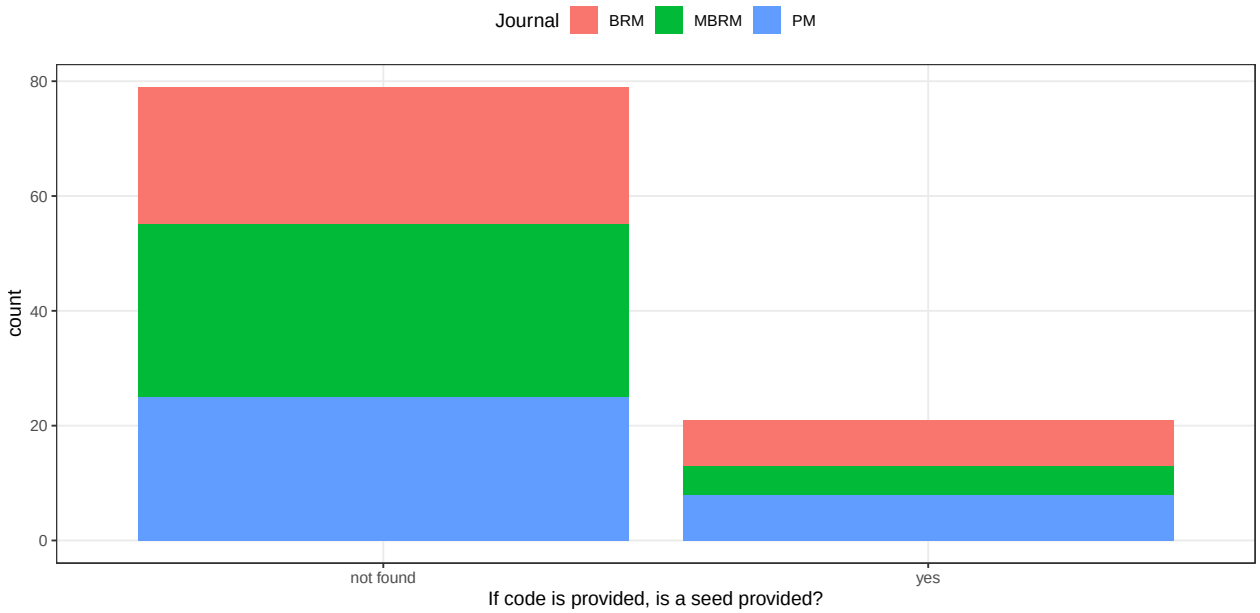

```
## Q19 Are there userwritten commands/packages/macros?
ggplot(data = sim_res_fac, aes(x = userwritten_q19, fill = journal)) +
    geom_bar() +
    labs(x = "Are there userwritten commands/packages/macros?", fill = "Journal")
```



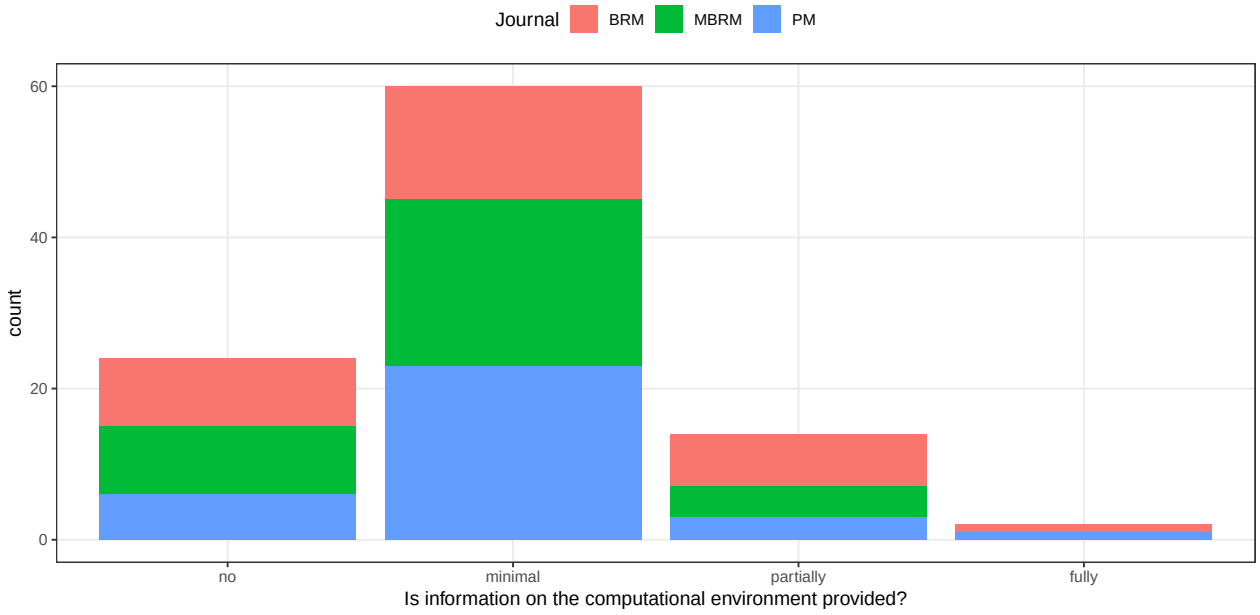

```
## Q20 Is code provided?
ggplot(data = sim_res_fac, aes(x = codeprovided_q20, fill = journal)) +
    geom_bar() +
    labs(x = "Is code provided?", fill = "Journal")
```




```
## Q21 If code is provided, is a seed provided?
ggplot(data = sim_res_fac, aes(x = seedprovided_q21, fill = journal)) +
    geom_bar() +
    labs(x = "If code is provided, is a seed provided?", fill = "Journal")
```

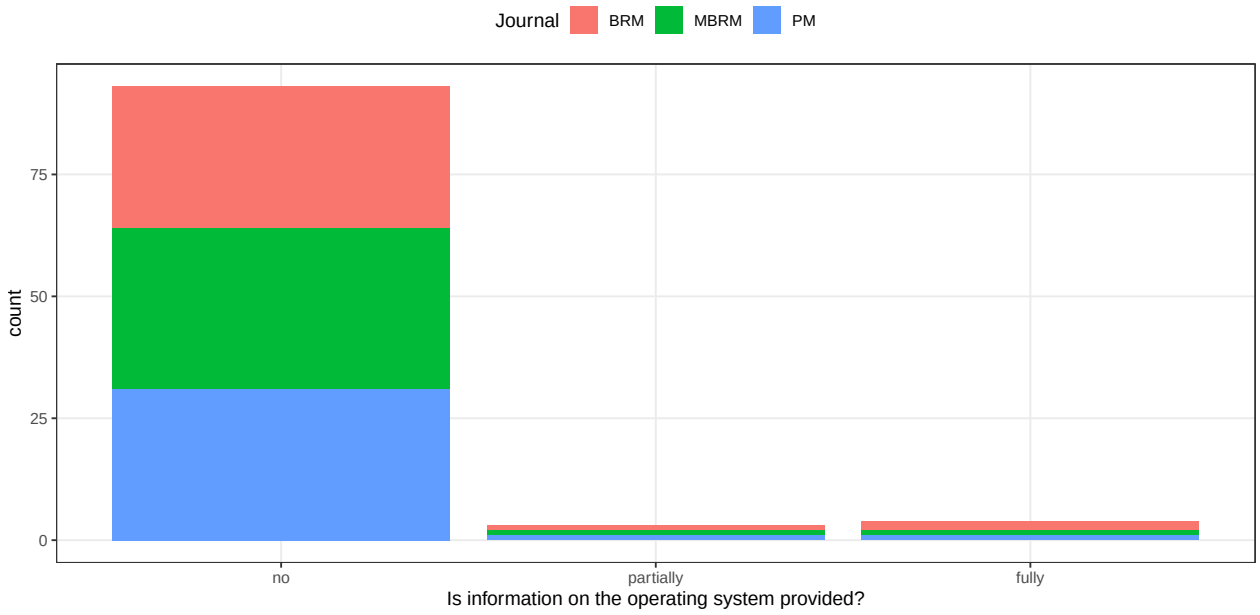



```
## Q22 Is information on the computational environment provided?
sim_res_fac %>%
    mutate(compenvironment_q22 = factor(compenvironment_q22,
                                        levels = c("no", "minimal", "partially", "fully"))) %>%
    ggplot(aes(x = compenvironment_q22, fill = journal)) +
    geom_bar() +
    labs(x = "Is information on the computational environment provided?",
         fill = "Journal")
```

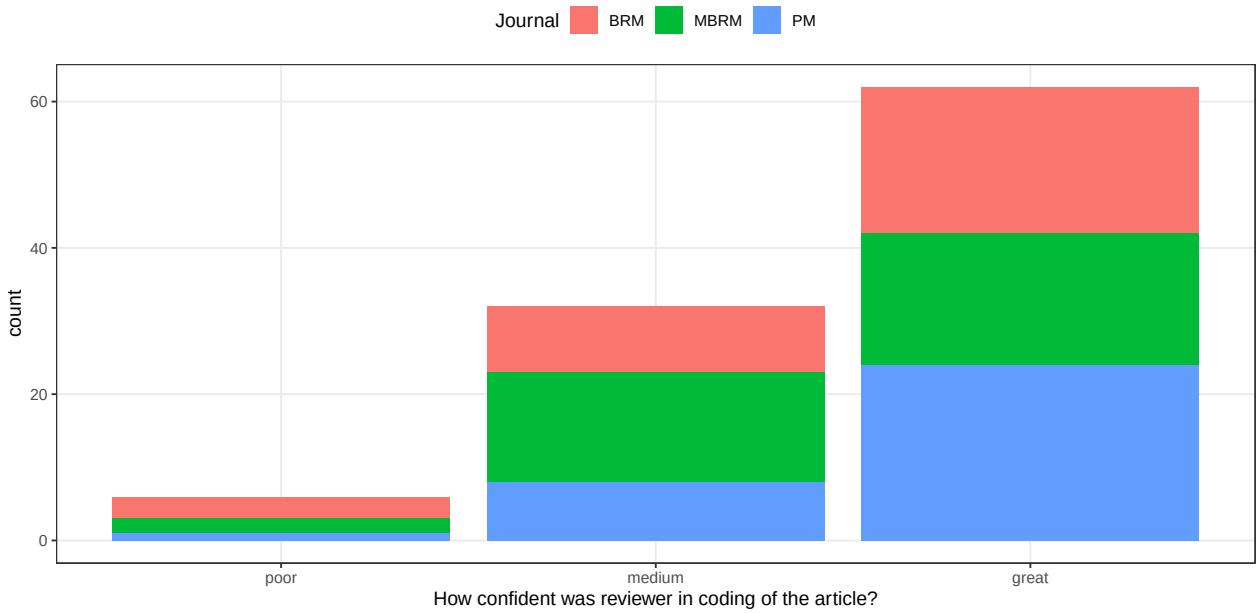



```
## Q23 Is information on the operating system provided?
sim_res_fac %>%
    mutate(compos_q23 = factor(compos_q23,
                               levels = c("no", "partially", "fully"))) %>%
    ggplot(aes(x = compos_q23, fill = journal)) +
    geom_bar() +
```

```
    labs(x = "Is information on the operating system provided?",
         fill = "Journal")
```




```
## Q24 How confident was reviewer in coding of the article?
sim_res_fac %>%
    mutate(coding_confidence = factor(coding_confidence,
                                      levels = c("poor", "medium", "great"))) %>%
    ggplot(aes(x = coding_confidence, fill = journal)) +
    geom_bar() +
    labs(x = "How confident was reviewer in coding of the article?",
         fill = "Journal")
```




```
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16)
```

```
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Debian GNU/Linux 12 (bookworm)
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.21.so;  LAPACK version 3.11.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Europe/Zurich
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] ggpubr_0.6.0  ggplot2_3.4.2 tidyr_1.3.0   dplyr_1.1.2
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.3      compiler_4.3.1    ggsignif_0.6.4    tidyselect_1.2.0
##  [5] scales_1.2.1      yaml_2.3.7        fastmap_1.1.1     R6_2.5.1
##  [9] labeling_0.4.2    generics_0.1.3    knitr_1.43        backports_1.4.1
## [13] tibble_3.2.1      car_3.1-2         munsell_0.5.0     pillar_1.9.0
## [17] rlang_1.1.1       utf8_1.2.3        broom_1.0.5       xfun_0.39
## [21] cli_3.6.1         withr_2.5.0       magrittr_2.0.3    digest_0.6.33
## [25] grid_4.3.1        rstudioapi_0.15.0 lifecycle_1.0.3   vctrs_0.6.3
## [29] rstatix_0.7.2     evaluate_0.21     glue_1.6.2        farver_2.1.1
## [33] abind_1.4-5       carData_3.0-5     fansi_1.0.4       colorspace_2.1-0
## [37] rmarkdown_2.23    purrr_1.0.1       tools_4.3.1       pkgconfig_2.0.3
## [41] htmltools_0.5.5
```