README: Replicating 'The Limits (and Strengths) of Single-Topic Experiments'

Reproduction tl;dr

The code below (or `make-figures.R`) creates both figures in the paper. The code runs almost instantly.

The file `install-packages.R` (written dynamically below) will install all the packages I use for this project in the appropriate version. Please run if you want to use the same package versions as I did.

```
# remove all generated files
file.remove("figure1.png", "figure2.png")

# run the code that makes the two figures in the paper
source("make-figures.R")
```

Output

The code produces two figures:

- `figure1.png`, which corresponds to Figure 1 in the paper.
- `figure2.png`, which corresponds to Figure 1 in the paper.

Session Information

```
sessioninfo::session_info(info = "platform")
```

```
- Session info -------------------------------------------------------------
 setting  value
 version  R version 4.3.2 (2023-10-31)
 os       macOS Ventura 13.4
 system   aarch64, darwin20
 ui       X11
 language (EN)
 collate  en_US.UTF-8
 ctype    en_US.UTF-8
 tz       America/New_York
 date     2024-08-26
 pandoc   3.1.6.1 @ /usr/local/bin/ (via rmarkdown)

---------------------------------------------------------------------------------
```

System Information

```
  # os info
  Sys.info() |>
    tibble::as_tibble(rownames = "par") |>
    dplyr::filter(par != "nodename")
```

```
# A tibble: 7 x 2
  par            value
  <chr>          <chr>
1 sysname        Darwin
2 release        22.5.0
3 version        Darwin Kernel Version 22.5.0: Mon Apr 24 20:53:19 PDT 2023; ro~
4 machine        arm64
5 login          root
6 user           carlislerainey
7 effective_user carlislerainey
```

```
  # cpu info
  benchmarkme::get_cpu()
```

```
$vendor_id
character(0)
```

```
$model_name
[1] "Apple M2 Max"

$no_of_cores
[1] 12
```

```
# ram info
ram_info <- system("sysctl hw.memsize", intern = TRUE)
print(ram_info)
```

```
[1] "hw.memsize: 34359738368"
```

R Packages Used

```
library(tidyverse)

# Function to get package version
get_version <- function(package_names){
  sapply(package_names, function(pkg) as.character(packageVersion(pkg)))
}

dep <- renv::dependencies("make-figures.R") %>%
  bind_rows(renv::dependencies("README.qmd")) %>%
  select(Package) %>%
  distinct() %>%
  arrange(Package) %>%
  mutate(Version = get_version(Package))
```

```
Finding R package dependencies ... Done!
Finding R package dependencies ... Done!
```

```
dep
```

```
       Package Version
1  benchmarkme   1.0.8
2        dplyr   1.1.3
3      ggplot2   3.4.3
```

```
4    patchwork   1.2.0
5         renv   1.0.7
6    rmarkdown    2.28
7       scales   1.3.0
8  sessioninfo   1.2.2
9       tibble   3.2.1
10   tidyverse   2.0.0
```

Write Script to Install Packages and Versions

```r
# create a new script file
script_file <- file("install-packages.R", "w")

# write install.packages() commands for each package to the file
for (i in 1:nrow(dep)) {
  install_cmd <- sprintf("remotes::install_version('%s', version = '%s', repos = 'http://c
  cat(install_cmd, file = script_file, append = TRUE)
}

# close the file
close(script_file)
```