

Data preprocessing for analysis

Signal detection of spontaneous medical device reports over time

Ty Stanford and Curtis Murray

Table of contents

1	Set up	2
1.1	Packages	2
1.2	Constants	2
2	Data wrangling	4
2.1	Read data	4
2.2	Clean/remove duplicates	4
2.3	Inspect and summarise data	5
3	Create data for analysis from raw data	10
3.1	Creation of analysis data	10
3.2	Check analysis data	12
3.3	Export analysis data	15
4	Session information	16

1 Set up

1.1 Packages

```
suppressPackageStartupMessages({  
  library("readr")  
  library("dplyr")  
  library("tidyr")  
  library("lubridate") # way to handle dates better than default R way  
  library("ggplot2")  
  library("ggrepel")  
  library("knitr")  
  library("foreach")  
  library("arrow") # read/write parquet files  
})  
  
# here are the functions written for this project  
source("../r/_funcs.R")
```

1.2 Constants

```
# arbitrarily, let's go with minimum cell count of 1  
arbitrary_cell_min <- 1  
  
# these are the thresholds for pain_topic to be pain == TRUE  
# thresholds <- c(0.010, 0.025, 0.05, 0.075, 0.100)  
thresholds <- seq(0.010, 0.100, by = 0.005) #, 0.150)  
  
col_pal <- c("cyan4", "darkorange", "purple", "dodgerblue")  
  
target_lst <-  
  list(  
    "pelvic_mesh",  
    "pelvic_mesh",  
    "pelvic_mesh",
```

```
    "hernia_mesh",  
    c("hernia_mesh", "other_mesh")  
  )  
  
compar_lst <-  
  list(  
    "hernia_mesh",  
    c("hernia_mesh", "other_mesh"),  
    c("hernia_mesh", "other_mesh", "other_device"),  
    "other_mesh",  
    "other_device"  
  )
```

2 Data wrangling

2.1 Read data

```
clean_data_cols <-  
  cols(  
    Report_ID = col_double(),  
    Date = col_date(format = ""),  
    pain_word = col_logical(),  
    pain_topic = col_double(),  
    type = col_character()  
  )  
  
clean_data <- read_csv("../dat/clean_data.csv", col_types = clean_data_cols)
```

2.2 Clean/remove duplicates

```
### all look like duplicates  
inner_join(  
  clean_data,  
  clean_data %>%  
    group_by(Report_ID) %>%  
    summarise(n = n(), .groups = "drop") %>%  
    dplyr::filter(n > 1),  
  "Report_ID"  
) %>%  
  arrange(Report_ID) %>%  
  print(., n = nrow(.))
```

A tibble: 26 x 6

	Report_ID	Date	pain_word	pain_topic	type	n
	<dbl>	<date>	<lgl>	<dbl>	<chr>	<int>
1	29914	2014-07-03	TRUE	0.0270	other_device	2
2	29914	2014-07-03	TRUE	0.0270	other_device	2
3	31508	2014-07-03	TRUE	0.0882	other_device	2
4	31508	2014-07-03	TRUE	0.0882	other_device	2
5	32629	2014-07-03	FALSE	0	other_device	2
6	32629	2014-07-03	FALSE	0	other_device	2
7	36586	2015-03-25	FALSE	0	other_device	2

8	36586	2015-03-25	FALSE	0	other_device	2
9	36677	2015-06-26	FALSE	0	other_device	2
10	36677	2015-06-26	FALSE	0	other_device	2
11	36953	2015-06-06	FALSE	0	other_device	2
12	36953	2015-06-06	FALSE	0	other_device	2
13	41788	2016-12-08	FALSE	0	other_device	2
14	41788	2016-12-08	FALSE	0	other_device	2
15	43614	2016-12-13	FALSE	0	other_device	2
16	43614	2016-12-13	FALSE	0	other_device	2
17	45287	2017-06-04	FALSE	0	other_device	2
18	45287	2017-06-04	FALSE	0	other_device	2
19	45369	2017-05-20	FALSE	0	other_device	2
20	45369	2017-05-20	FALSE	0	other_device	2
21	45749	2017-10-06	FALSE	0	other_device	2
22	45749	2017-10-06	FALSE	0	other_device	2
23	46029	2017-10-06	FALSE	0	other_device	2
24	46029	2017-10-06	FALSE	0	other_device	2
25	46030	2017-09-06	FALSE	0	other_device	2
26	46030	2017-09-06	FALSE	0	other_device	2

```
# make dup free
clean_data <-
  clean_data %>%
  arrange(Report_ID, Date, desc(pain_word)) %>% # pain first in dups
  group_by(Report_ID) %>%
  dplyr::filter(row_number() == 1) %>%
  ungroup(.) %>%
  arrange(Date, Report_ID, desc(pain_word), desc(pain_topic))

clean_data %>%
  dplyr::filter(type == "other_mesh") %>%
  # select(Report_ID) %>%
  write_csv("../out/other_mesh_ids.csv")
```

2.3 Inspect and summarise data

```
cat("First 10 rows of raw data\n")
```

First 10 rows of raw data

```
clean_data %>%
  arrange(Date) %>%
  dplyr::filter(row_number() < 11) %>%
  kable(.)
```

Report_ID	Date	pain_word	pain_topic	type
26696	2012-01-08	FALSE	0.0555556	other_device
27722	2012-01-08	FALSE	0.0000000	other_device
28827	2012-01-10	FALSE	0.0000000	other_device
28828	2012-01-10	TRUE	0.0500000	other_device
28452	2012-01-11	FALSE	0.0000000	other_device
28758	2012-01-11	FALSE	0.0000000	other_device
28826	2012-01-11	FALSE	0.0400000	other_device
29097	2012-01-11	FALSE	0.0000000	other_device
29100	2012-01-11	FALSE	0.0000000	other_device
29101	2012-01-11	FALSE	0.0000000	other_device

```
# clean_data <-
# clean_data %>%
# dplyr::filter(
#   type %in% c("pelvic_mesh", "hernia_mesh")
# )
```

```
clean_data %>%
  with(., table(type, pain_word)) %>%
  knitr::kable(.)
```

	FALSE	TRUE
hernia_mesh	42	4
other_device	12741	1184
other_mesh	52	32
pelvic_mesh	32	70

```
clean_data %>%
  with(., table(type, pain_topic >= 0.05)) %>%
```

```
knitr::kable(.)
```

	FALSE	TRUE
hernia_mesh	38	8
other_device	12386	1539
other_mesh	47	37
pelvic_mesh	25	77

```
# These are the device groups and subgroups.
```

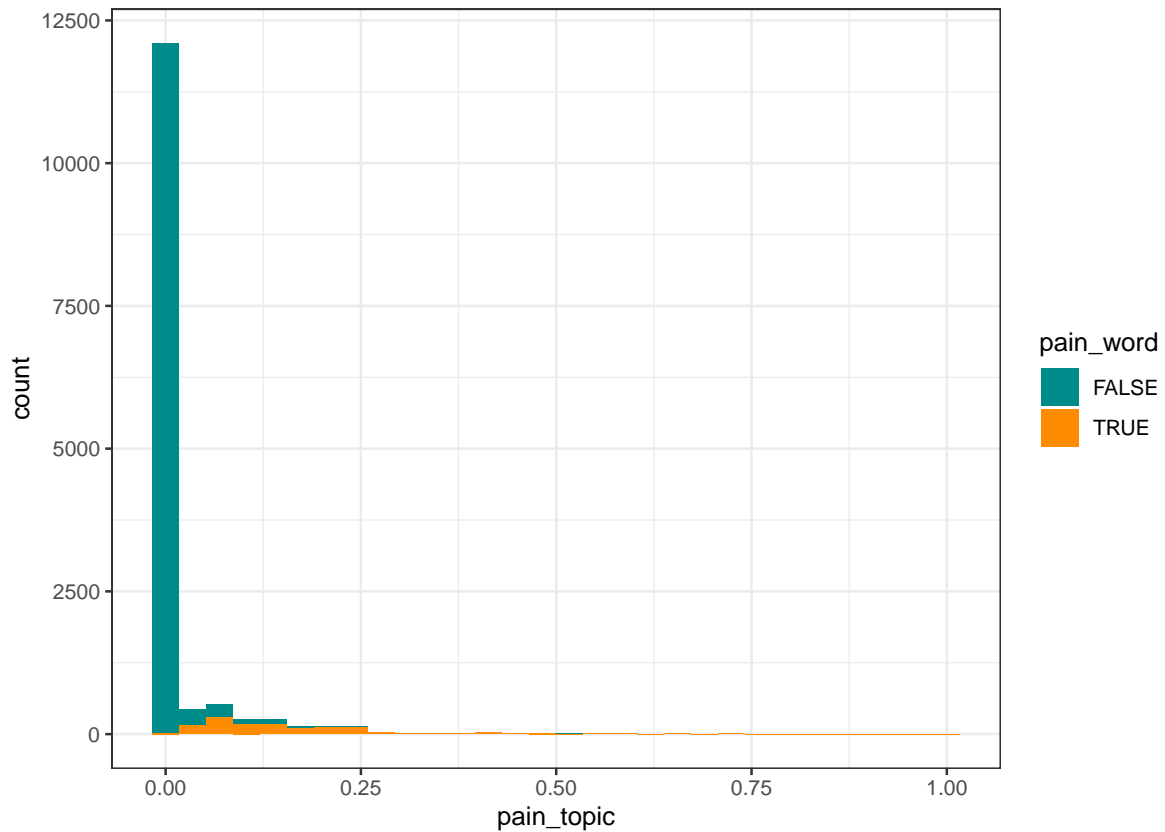
```
clean_data %>%
  group_by(type) %>%
  summarise(count = n()) %>%
  kable(.)
```

type	count
hernia_mesh	46
other_device	13925
other_mesh	84
pelvic_mesh	102

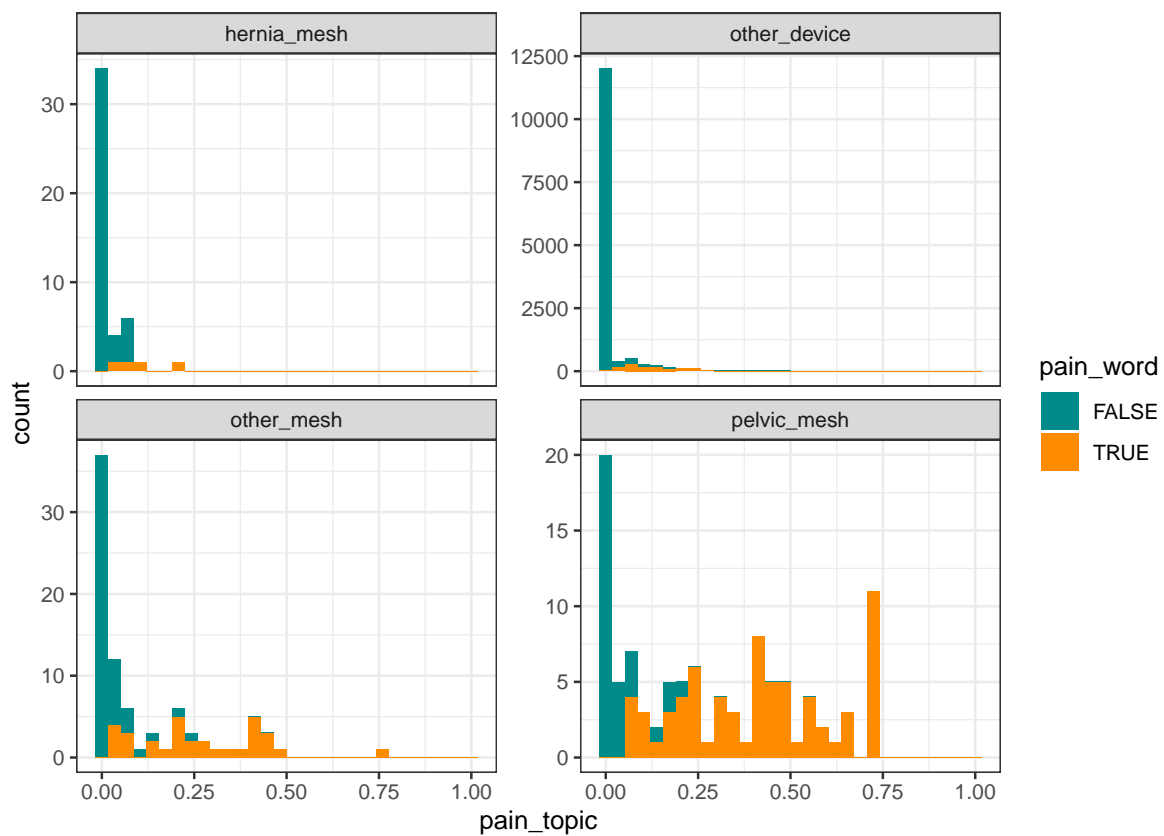
```
cat("\n\n## Histogram of `pain_word` (boolean) v `pain_topic` (score)")
```

```
## Histogram of `pain_word` (boolean) v `pain_topic` (score)
```

```
clean_data %>%
  ggplot(., aes(pain_topic, fill = pain_word)) +
  geom_histogram(bins = 30) +
  scale_fill_manual(values = col_pal[1:2]) +
  theme_bw()
```



```
clean_data %>%
  dplyr::filter(
    type %in% c("pelvic_mesh", "hernia_mesh", "other_mesh", "other_device")
  ) %>%
  ggplot(., aes(pain_topic, fill = pain_word)) +
  geom_histogram(bins = 30) +
  scale_fill_manual(values = col_pal[1:2]) +
  facet_wrap(~ type, scales = "free_y") +
  theme_bw()
```

3 Create data for analysis from raw data

3.1 Creation of analysis data

```
### testing: example 1
# Use pelvic mesh as group 1 and hernia_mesh mesh devices as group 2.
# The value of interest is the pain topic, being above the threshold of 0.05.
# (i.e. 5% of the document contains words from the pain topic)
# You can adjust the topic threshold if you want to balance the groups more.
# A higher topic_threshold will look for documents that discuss "pain" more, and
# hence find less pain documents.

# get_signal_dat(
#   g1 = "pelvic_mesh",
#   g2 = "hernia_mesh",
#   pain_type = "pain_topic",
#   thresh = 0.05,
#   cell_min = 1,
#   cumul = TRUE,
#   verbose = FALSE
# ) %>%
#   bind_cols(., thresh = 0.05)

# takes ~ 20 sec
cumul_dat <-
  foreach(i = 1:length(target_lst), .combine = bind_rows, .packages = "dplyr") %do% {
    foreach(th_j = thresholds, .combine = bind_rows, .packages = "dplyr") %do% {

      get_signal_dat(
        g1 = target_lst[[i]],
        g2 = compar_lst[[i]],
        pain_type = "pain_topic",
        thresh = th_j,
        cell_min = 1,
        cumul = TRUE,
        verbose = FALSE
      ) %>%
      mutate(
        grps =
```

```

    paste0(
      paste(target_lst[[i]], collapse = "/"),
      " v ",
      paste(compar_lst[[i]], collapse = "/")
    ),
    dat_type = "cumulative",
    thresh = th_j
  ) %>%
  select(grps, dat_type, thresh, everything())
}
}
cumul_dat

```

A tibble: 4,523 x 8

	grps		dat_type	thresh	mnth	nA	nB	nC	nD
	<chr>		<chr>	<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	pelvic_mesh v hernia_mesh	cumulative	0.01	2013-01	3	7	1	2	
2	pelvic_mesh v hernia_mesh	cumulative	0.01	2013-02	3	7	1	4	
3	pelvic_mesh v hernia_mesh	cumulative	0.01	2013-04	3	7	1	5	
4	pelvic_mesh v hernia_mesh	cumulative	0.01	2013-05	4	10	1	5	
5	pelvic_mesh v hernia_mesh	cumulative	0.01	2013-07	4	11	1	7	
6	pelvic_mesh v hernia_mesh	cumulative	0.01	2013-08	5	11	1	7	
7	pelvic_mesh v hernia_mesh	cumulative	0.01	2013-09	5	11	2	9	
8	pelvic_mesh v hernia_mesh	cumulative	0.01	2013-11	8	11	2	9	
9	pelvic_mesh v hernia_mesh	cumulative	0.01	2013-12	9	11	2	9	
10	pelvic_mesh v hernia_mesh	cumulative	0.01	2014-03	9	11	2	10	

... with 4,513 more rows

takes ~ 20 sec

```

snpsht_dat <-
  foreach(i = 1:length(target_lst), .combine = bind_rows, .packages = "dplyr") %do% {
    foreach(th_j = thresholds, .combine = bind_rows, .packages = "dplyr") %do% {

      get_signal_dat(
        g1 = target_lst[[i]],
        g2 = compar_lst[[i]],
        pain_type = "pain_topic",
        thresh = th_j,
        cell_min = 1,

```

```

        cumul = FALSE,
        verbose = FALSE
      ) %>%
      mutate(
        grps =
          paste0(
            paste(target_lst[[i]], collapse = "/"),
            " v ",
            paste(compar_lst[[i]], collapse = "/")
          ),
        dat_type = "snapshot",
        thresh = th_j
      ) %>%
      select(grps, dat_type, thresh, everything())
    }
  }
  snpsh_dat

```

A tibble: 4,523 x 8

	grps		dat_type	thresh	mnth	nA	nB	nC	nD
	<chr>		<chr>	<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	pelvic_mesh v hernia_mesh	snapshot	0.01	2013-01	3	7	1	2	
2	pelvic_mesh v hernia_mesh	snapshot	0.01	2013-02	0	0	0	2	
3	pelvic_mesh v hernia_mesh	snapshot	0.01	2013-04	0	0	0	1	
4	pelvic_mesh v hernia_mesh	snapshot	0.01	2013-05	1	3	0	0	
5	pelvic_mesh v hernia_mesh	snapshot	0.01	2013-07	0	1	0	2	
6	pelvic_mesh v hernia_mesh	snapshot	0.01	2013-08	1	0	0	0	
7	pelvic_mesh v hernia_mesh	snapshot	0.01	2013-09	0	0	1	2	
8	pelvic_mesh v hernia_mesh	snapshot	0.01	2013-11	3	0	0	0	
9	pelvic_mesh v hernia_mesh	snapshot	0.01	2013-12	1	0	0	0	
10	pelvic_mesh v hernia_mesh	snapshot	0.01	2014-03	0	0	0	1	

... with 4,513 more rows

3.2 Check analysis data

```
nrow(cumul_dat)
```

```
[1] 4523
```

```

if (nrow(cumul_dat) != nrow(snpsh_dat)) {
  stop("logic of creating analysis data producing different # rows in data")
}

chk_start_vals <-
  inner_join(
    cumul_dat %>%
      group_by(grps, dat_type, thresh) %>%
      dplyr::filter(row_number() == 1) %>%
      ungroup(.),
    snpsh_dat %>%
      group_by(grps, dat_type, thresh) %>%
      dplyr::filter(row_number() == 1) %>%
      ungroup(.),
    c("grps", "thresh")
  ) %>%
  mutate(
    mnth_same = (mnth.x == mnth.y),
    counts_same = (nA.x == nA.y) & (nB.x == nB.y) & (nC.x == nC.y) & (nD.x == nD.y)
  )

chk_start_vals %>%
  select(grps, thresh, dat_type.x, dat_type.y, mnth_same, counts_same)

```

A tibble: 95 x 6

	grps		thresh	dat_type.x	dat_type.y	mnth_same	counts_same
	<chr>		<dbl>	<chr>	<chr>	<lgl>	<lgl>
1	pelvic_mesh	v hernia_mesh	0.01	cumulative	snapshot	TRUE	TRUE
2	pelvic_mesh	v hernia_mesh	0.015	cumulative	snapshot	TRUE	TRUE
3	pelvic_mesh	v hernia_mesh	0.02	cumulative	snapshot	TRUE	TRUE
4	pelvic_mesh	v hernia_mesh	0.025	cumulative	snapshot	TRUE	TRUE
5	pelvic_mesh	v hernia_mesh	0.03	cumulative	snapshot	TRUE	TRUE
6	pelvic_mesh	v hernia_mesh	0.035	cumulative	snapshot	TRUE	TRUE
7	pelvic_mesh	v hernia_mesh	0.04	cumulative	snapshot	TRUE	TRUE
8	pelvic_mesh	v hernia_mesh	0.045	cumulative	snapshot	TRUE	TRUE
9	pelvic_mesh	v hernia_mesh	0.05	cumulative	snapshot	TRUE	TRUE
10	pelvic_mesh	v hernia_mesh	0.055	cumulative	snapshot	TRUE	TRUE

... with 85 more rows

```
with(chk_start_vals, table(mnth_same, counts_same, useNA = "ifany"))
```

```
      counts_same
mntn_same TRUE
      TRUE      95
```

```
# check the first + second row in snapshot == second row in cumulative data
inner_join(
  cumul_dat %>%
    group_by(grps, thresh) %>%
    dplyr::filter(row_number() %in% 1:2) %>%
    ungroup(.),
  snpsh_dat %>%
    group_by(grps, thresh) %>%
    dplyr::filter(row_number() %in% 1:2) %>%
    ungroup(.),
  c("grps", "thresh", "mntn")
)
```

```
# A tibble: 190 x 13
```

	grps	dat_t~1	thresh	mntn	nA.x	nB.x	nC.x	nD.x	dat_t~2	nA.y	nB.y	nC.y
	<chr>	<chr>	<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<dbl>	<dbl>	<dbl>
1	pelvi~	cumula~	0.01	2013~	3	7	1	2	snapsh~	3	7	1
2	pelvi~	cumula~	0.01	2013~	3	7	1	4	snapsh~	0	0	0
3	pelvi~	cumula~	0.015	2013~	3	7	1	2	snapsh~	3	7	1
4	pelvi~	cumula~	0.015	2013~	3	7	1	4	snapsh~	0	0	0
5	pelvi~	cumula~	0.02	2013~	5	11	1	10	snapsh~	5	11	1
6	pelvi~	cumula~	0.02	2013~	8	11	1	10	snapsh~	3	0	0
7	pelvi~	cumula~	0.025	2013~	5	11	1	10	snapsh~	5	11	1
8	pelvi~	cumula~	0.025	2013~	8	11	1	10	snapsh~	3	0	0
9	pelvi~	cumula~	0.03	2013~	5	11	1	10	snapsh~	5	11	1
10	pelvi~	cumula~	0.03	2013~	8	11	1	10	snapsh~	3	0	0

```
# ... with 180 more rows, 1 more variable: nD.y <dbl>, and abbreviated variable
#   names 1: dat_type.x, 2: dat_type.y
```

3.3 Export analysis data

```
# all spontaneous report analysis data
sra_dat <-
  bind_rows(
    cumul_dat,
    snpsh_dat
  )

sra_dat %>%
  write_parquet(., sink = "../dat/sra_dat.parquet")
```

4 Session information

```
format(Sys.time(), '%d %b %Y')
```

```
[1] "15 Jun 2023"
```

```
Sys.info() %>% as.data.frame(.)
```

```
      .  
sysname      Windows  
release      10 x64  
version      build 19044  
nodename     DESKTOP-R5P5N23  
machine      x86-64  
login        ty  
user         ty  
effective_user ty
```

```
sessionInfo()
```

```
R version 4.2.2 (2022-10-31 ucrt)  
Platform: x86_64-w64-mingw32/x64 (64-bit)  
Running under: Windows 10 x64 (build 19044)
```

```
Matrix products: default
```

```
locale:  
[1] LC_COLLATE=English_Australia.utf8  LC_CTYPE=English_Australia.utf8  
[3] LC_MONETARY=English_Australia.utf8 LC_NUMERIC=C  
[5] LC_TIME=English_Australia.utf8
```

```
attached base packages:  
[1] stats      graphics  grDevices  utils      datasets  methods   base
```

```
other attached packages:  
[1] arrow_11.0.0.2  foreach_1.5.2  knitr_1.42      ggrepel_0.9.3  
[5] ggplot2_3.4.1   lubridate_1.9.2 tidyr_1.3.0     dplyr_1.1.0
```


[9] readr_2.1.4

loaded via a namespace (and not attached):

[1] Rcpp_1.0.10	pillar_1.8.1	compiler_4.2.2	iterators_1.0.14
[5] tools_4.2.2	bit_4.0.5	digest_0.6.31	jsonlite_1.8.4
[9] evaluate_0.20	lifecycle_1.0.3	tibble_3.1.8	gtable_0.3.1
[13] timechange_0.2.0	pkgconfig_2.0.3	rlang_1.0.6	cli_3.6.0
[17] rstudioapi_0.14	parallel_4.2.2	yaml_2.3.7	xfun_0.37
[21] fastmap_1.1.0	withr_2.5.0	generics_0.1.3	vctrs_0.5.2
[25] hms_1.1.2	bit64_4.0.5	grid_4.2.2	tidyselect_1.2.0
[29] glue_1.6.2	R6_2.5.1	fansi_1.0.4	vroom_1.6.1
[33] rmarkdown_2.20	farver_2.1.1	tzdb_0.3.0	purrr_1.0.1
[37] magrittr_2.0.3	codetools_0.2-18	scales_1.2.1	ellipsis_0.3.2
[41] htmltools_0.5.4	assertthat_0.2.1	colorspace_2.1-0	labeling_0.4.2
[45] utf8_1.2.3	munsell_0.5.0	crayon_1.5.2	