```
theme_set(theme_bw())

cols <- wes_palette(name = "IsleofDogs1")
```

Each project closes with a table summarising the R tools used. By visualising my most frequently used packages and functions I get a sense of where I may most benefit from going deeper and keeping abreast of the latest breaking changes.

I may also spot superseded functions e.g. `spread` and `gather` may now be replaced by `pivot_wider` and `pivot_longer`. Or an opportunity to switch a non-tidyverse package for a newer tidyverse (or ecosystem) alternative, e.g. for UpSetR I can now use ggupset which plays well with ggplot.

I'll start by listing the paths to the html files in the project directory.

```
files <- list.files(
  path = "/Users/carl/R Projects/blogdown6/content/project/",
  pattern = "\\.html$", recursive = TRUE
) %>%
  str_c("/Users/carl/R Projects/blogdown6/content/project/", .) %>%
  as_tibble() %>%
  filter(!str_detect(value, "world|dt1|appfiles")) %>%
  pull()
```

This enables me to extract the usage table for each project.

```
table_df <- map_dfr(files, function(x) {
  x %>%
    read_html() %>%
    html_nodes("#r-toolbox table") %>%
    html_table() %>%
    bind_rows()
}) %>% clean_names(replace = c("io" = ""))
```

A little "spring cleaning" is needed, and separation of tidyverse and non-tidyverse packages.

- tidyverse
- tidymodels
- tidyverts

```
tidyv <- tidyverse_packages()
tidyf <- fpp3_packages()
tidym <- tidymodels_packages()

tidy <- c(tidyv, tidyf, tidym) %>% unique()

tidy_df <- table_df %>%
  separate_rows(functn, sep = ";") %>%
  separate(functn, c("functn", "count"), literal("[")) %>%
  mutate(
    count = str_remove(count, "]") %>% as.integer(),
    functn = str_squish(functn)
  ) %>%
  group_by(package, functn) %>%
  summarise(count = sum(count)) %>%
  mutate(multiverse = case_when(
    package %in% tidy ~ "tidy",
    package %in% c("base", "graphics") ~ "base",
    TRUE ~ "special"
  ))
```

Then I can summarise usage and prepare for a faceted plot.

```
pack_df <- tidy_df %>%
  group_by(package, multiverse) %>%
  summarise(count = sum(count)) %>%
  ungroup() %>%
  mutate(name = "package")

fun_df <- tidy_df %>%
  group_by(functn, multiverse) %>%
  summarise(count = sum(count)) %>%
  ungroup() %>%
  mutate(name = "function")

n_url <- files %>% n_distinct()

packfun_df <- pack_df %>%
  bind_rows(fun_df) %>%
  group_by(name) %>%
  arrange(desc(count)) %>%
  mutate(
    packfun = coalesce(package, functn),
    name = fct_rev(name)
  )
```

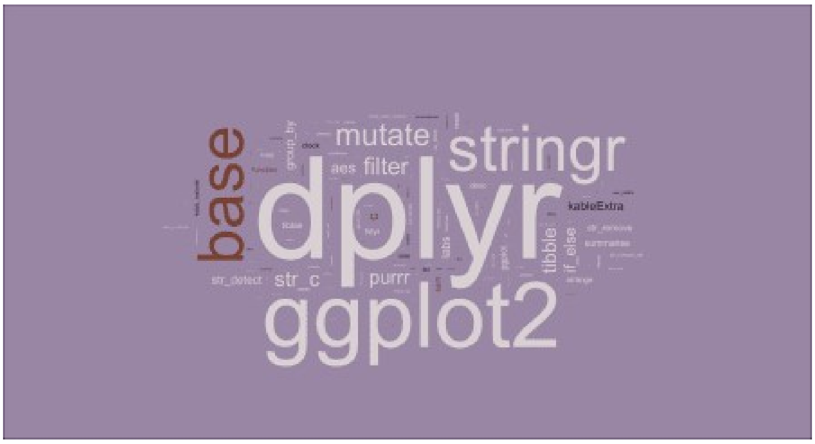
Clearly “dplyr rules”! And `mutate` is slugging it out with `library`.

```
packfun_df %>%
  slice(1:20) %>%
  ggplot(aes(reorder_within(packfun, count, name), count, fill = multiverse)) +
  geom_col() +
  geom_label(aes(label = count), size = 3, fill = "white") +
  facet_wrap(~name, ncol = 1, scales = "free", strip.position = "left") +
  scale_x_reordered() +
  scale_y_continuous(expand = expansion(mult = c(0, .15))) +
  scale_fill_manual(values = cols[c(2, 3, 1)]) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "bottom",
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank(),
    strip.background = element_rect(fill = cols[6]),
    strip.text = element_text(colour= "white")
  ) +
  labs(
    title = "Favourite Things",
    subtitle = glue("Most Frequent Usage Across {n_url} Projects"),
    x = NULL, y = NULL
  )
```

I’d also like a wordcloud. And thanks to blogdown, the updated visualisation is picked up as the new featured image for this project.

```
set.seed = 123
```

```
packfun_df %>%
  mutate(angle = 90 * sample(c(0, 1), n(), replace = TRUE, prob = c(60, 40)))
%>%
  ggplot(aes(label = packfun, size = count, colour = multiverse, angle = angle))
+
  geom_text_wordcloud(eccentricity = .9, seed = 456) +
  scale_radius(range = c(0, 40), limits = c(0, NA)) +
  scale_colour_manual(values = cols[c(2:4)]) +
  theme_void() +
  theme(plot.background = element_rect(fill = cols[1]))
```



R Toolbox

A little bit circular I know, but I might as well include this code too in my “favourite things”.

Package	Function
base	library[11]; c[9]; sum[4]; function[2]; as.integer[1]; conflicts[1]; cumsum[1]; list.files[1]; sample[1]; search[1]; unique[1]
dplyr	mutate[10]; count[5]; filter[5]; group_by[5]; summarise[4]; if_else[3]; arrange[2]; as_tibble[2]; bind_rows[2]; desc[2]; tibble[2]; ungroup[2]; case_when[1]; coalesce[1]; n[1]; n_distinct[1]; pull[1]; select[1]; slice[1]
forcats	fct_rev[1]
fpp3	fpp3_packages[1]
ggplot2	aes[3]; element_blank[2]; element_rect[2]; element_text[2]; ggplot[2]; theme[2]; expansion[1]; facet_wrap[1]; geom_col[1]; geom_label[1]; labs[1]; scale_colour_manual[1]; scale_fill_manual[1]; scale_radius[1]; scale_y_continuous[1]; theme_bw[1]; theme_set[1]; theme_void[1]
ggwordcloud	geom_text_wordcloud[1]; ggwordcloud[1]
glue	glue[2]
janitor	clean_names[1]
kableExtra	kable[1]
purrr	map[1]; map_dfr[1]; map2_dfr[1]; possibly[1]; set_names[1]
readr	read_lines[1]
rebus	literal[5]; lookahead[3]; whole_word[2]; ALPHA[1]; lookbehind[1]; one_or_more[1]; or[1]
rvest	html_nodes[1]; html_table[1]
stringr	str_detect[4]; str_c[3]; str_remove[3]; str_count[1]; str_remove_all[1]; str_squish[1]
tibble	enframe[1]
tidymodels	tidymodels_packages[1]
tidyr	as_tibble[2]; tibble[2]; separate[1]; separate_rows[1]; unnest[1]
tidytext	reorder_within[1]; scale_x_reordered[1]
tidyverse	tidyverse_packages[1]

Package	Function
tsibble	as_tibble[2]; tibble[2]
wesanderson	wes_palette[1]
xml2	read_html[1]