theme\_set(theme\_bw())

cols <- wes\_palette(name = "IsleofDogs1")

I may also spot superseded functions e.g. spread and gather may now be replaced by pivot\_wider and pivot\_longer.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)))

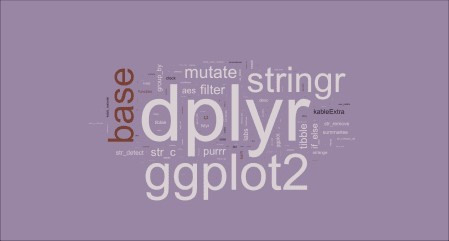
%>%

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

library[11]; c[9]; sum[4]; function[2]; as.integer[1]; conflicts[1]; cumsum[1]; list.files[1]; sample[1];

base

dplyr

search[1]; unique[1]

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]

aes[3]; element\_blank[2]; element\_rect[2]; element\_text[2]; ggplot[2]; theme[2]; expansion[1];

ggplot2

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]