

Intro to ggplot

LBE

09/02/2022

Welcome to ggplot!

- Based on tidyverse principles
- Iterative principles, layering data and graphical themes
- Uses a set Grammar of Graphics:
- **Data** are **mapped** to aesthetics
 - **Layers** are **geometric** elements, and **statistical** transformations
 - **scales** map values to geometric space
 - **coordinate** systems describe how data coordinates are mapped to a graphical plane
 - **Facets** break up and display subsets of data
 - **Themes** control details about the display, like font size etc.

Libraries

```
packages = c('tidyverse',  
             'patchwork',  
             'ggpubr')  
  
load.pack = lapply(packages, require, char=T)  
load.pack
```

```
## [[1]]  
## [1] TRUE  
##  
## [[2]]  
## [1] TRUE  
##  
## [[3]]  
## [1] TRUE
```

Data

Today's data: a collection of dispersal and self-recruitment data for anemonefish

```
load("../PublicationFigures/NemoDatabase.Rdata")
```

Let's start with some data wrangling

```
nemo_dat = nemo %>%
  mutate(Species = paste("A.", Species),
         Range_disp = as.character(Range_disp),
         PLD = as.character(PLD),
         Mkr_n = as.character(Mkr_n)) %>%
  mutate(SR_mean = ifelse(is.na(SR_mean), (SR_min + SR_max) / 2, SR_mean)) %>%
  mutate(Method = str_replace(Method, "IBD", "Genetic structure"),
         Method = str_replace(Method, "Assignment", "Assignment tests"),
         Method = factor(Method, levels = c("Parentage", "Assignment tests", "Genetic structure", "Otolith")) %>%
  mutate(Range_disp.1 = as.character(Range_disp)) %>%
  separate(Range_disp.1, c("min_Range", "max_Range"), sep = "-") %>%
  mutate(min_Range = as.numeric(min_Range),
         max_Range = as.numeric(max_Range)) %>%
  mutate(Adults = replace_na(Adults, 0),
         Juveniles = replace_na(Juveniles, 0)) %>%
  mutate(Species = as.factor(Species))
nemo_dat
```

```
## # A tibble: 33 x 39
##   Reference First_author Publication_year Genus Species Host PLD Country
##   <chr>      <chr>                <dbl> <chr> <chr> <chr> <chr> <chr>
## 1 Jones, G., ~ Jones                2005 Amph~ A. pol~ Stich~ 9-12 Papua ~
## 2 Jones, G., ~ Jones                2005 Amph~ A. pol~ Stich~ 9-12 Papua ~
## 3 Almany, G. ~ Almany                2007 Amph~ A. per~ Stich~ 10-12 Papua ~
## 4 Saenz-Agude~ Saenz-Agude~          2009 Amph~ A. pol~ Stich~ 1639~ Papua ~
## 5 Saenz-Agude~ Saenz-Agude~          2009 Amph~ A. pol~ Stich~ 1639~ Papua ~
## 6 Planes, S., ~ Planes              2009 Amph~ A. per~ Stich~ 11 Papua ~
## 7 Planes, S., ~ Planes              2009 Amph~ A. per~ Stich~ 11 Papua ~
## 8 Pinsky, M. ~ Pinsky               2010 Amph~ A. cla~ <NA> NULL Philip~
## 9 Saenz-Agude~ Saenz-Agude~          2011 Amph~ A. pol~ Stich~ 1639~ Papua ~
## 10 Saenz-Agude~ Saenz-Agude~          2011 Amph~ A. pol~ Stich~ 1639~ Papua ~
## # ... with 23 more rows, and 31 more variables: Location <chr>, Sea <chr>,
## # Coordinates <chr>, MPA <chr>, Method <fct>, Mkr_type <chr>, Mkr_n <chr>,
## # Software <chr>, Anemone <dbl>, Adults <dbl>, P_adults <list>,
## # Juveniles <dbl>, Ass_n <dbl>, Ass_rate <dbl>, Time_ya <dbl>, Period <list>,
## # Scale_km <list>, Area <dbl>, Sites <dbl>, Kernel <dbl>, Range_disp <chr>,
## # Mean_disp <list>, Median <dtm>, dist_50 <dbl>, dist_90 <list>,
## # SR_min <dbl>, SR_mean <dbl>, SR_max <dbl>, min_Range <dbl>, ...
```

Fig 1.

Our first figure!

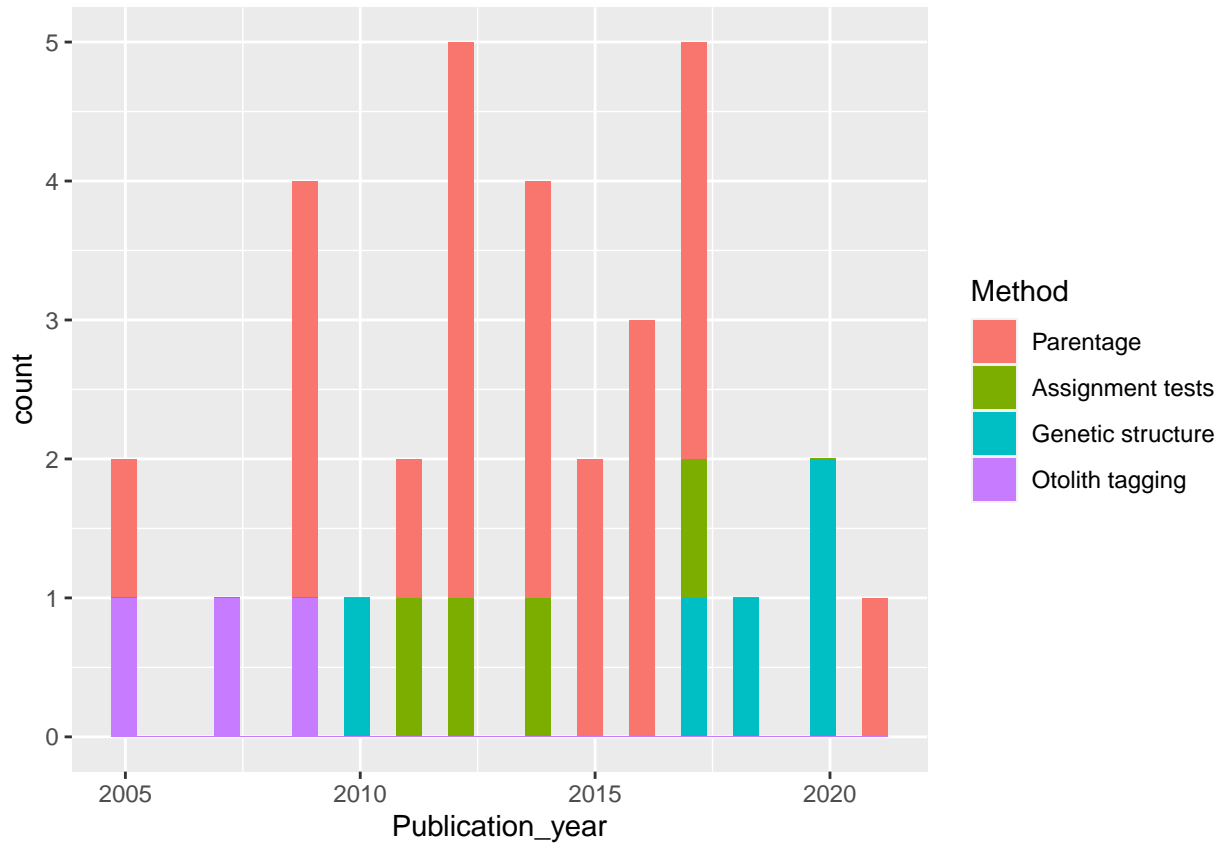
Every ggplot pot has three main components:

- **data**
- **aesthetic mappings** between data & visual properties

- At least one **layer** that describes how each observation is rendered, usually a **geom** functions

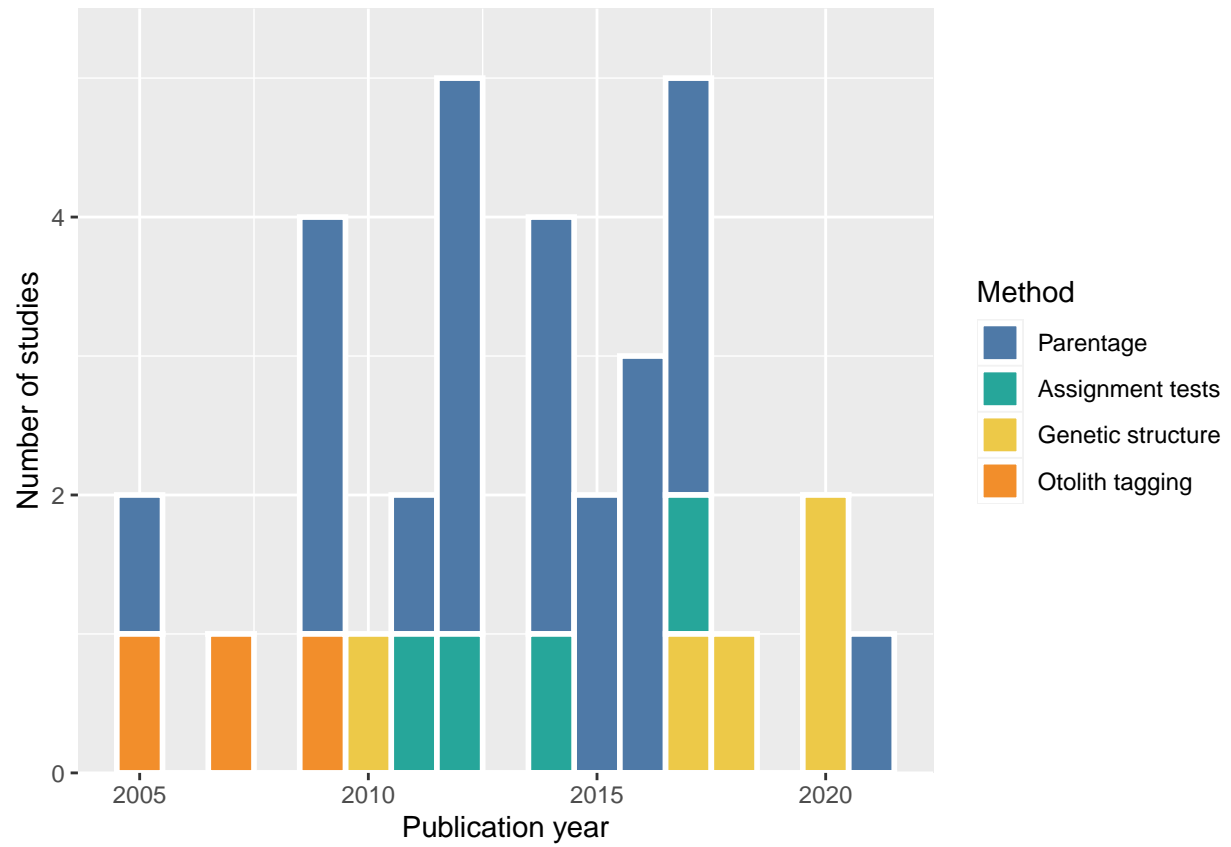
```
ggplot(data = nemo_dat, aes(x = Publication_year, fill = Method)) +  
  geom_histogram()
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```

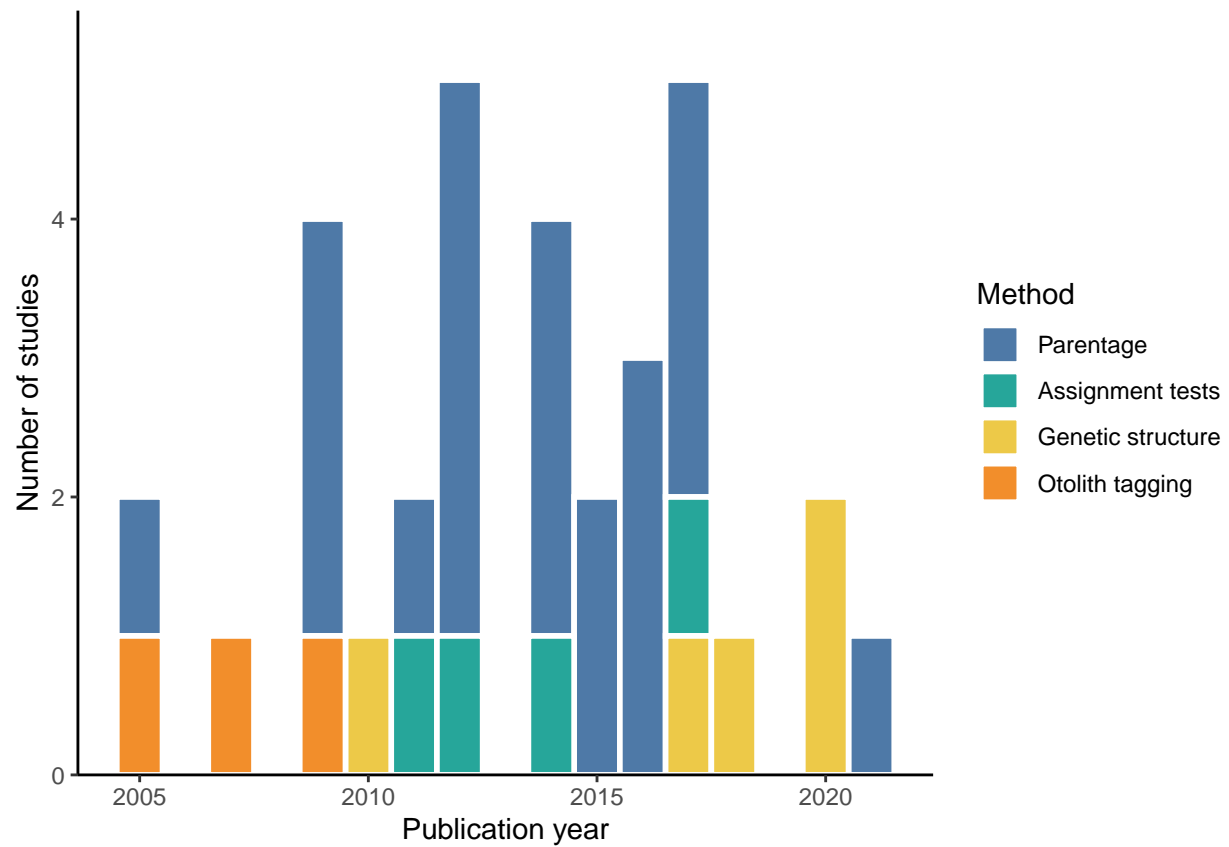


p1

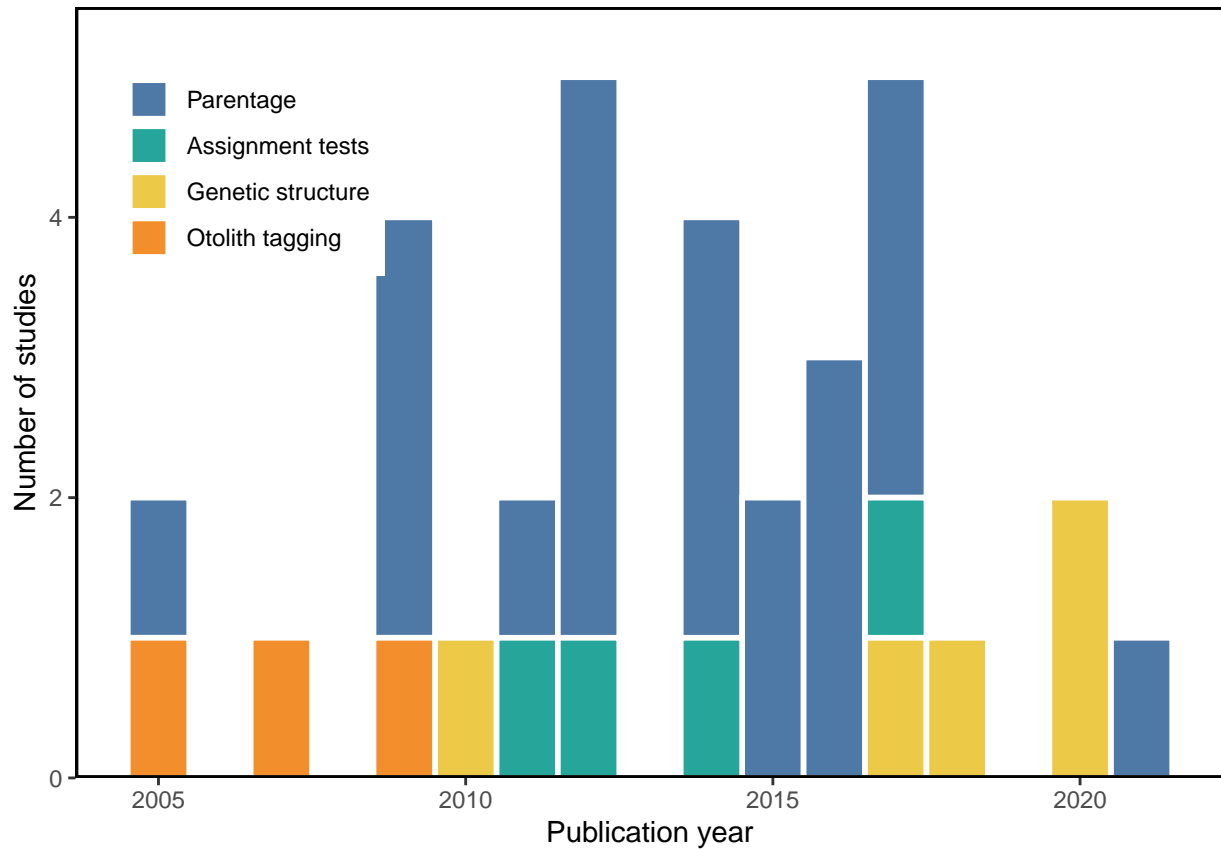
```
p1 <- ggplot(nemo_dat, aes(x = Publication_year, fill = Method)) +  
  geom_histogram(binwidth = 1, col = "white", size = 1) +  
  scale_fill_manual(values = c("#4e79a7", "#26A69A", "#edc948", "#f28e2b")) +  
  labs(x = "Publication year", y = "Number of studies") +  
  scale_y_continuous(expand = c(0,0), limits = c(0,5.5), breaks = c(0,2,4,6)) #  
p1
```



```
p1 = p1 +
  theme_classic()
p1
```

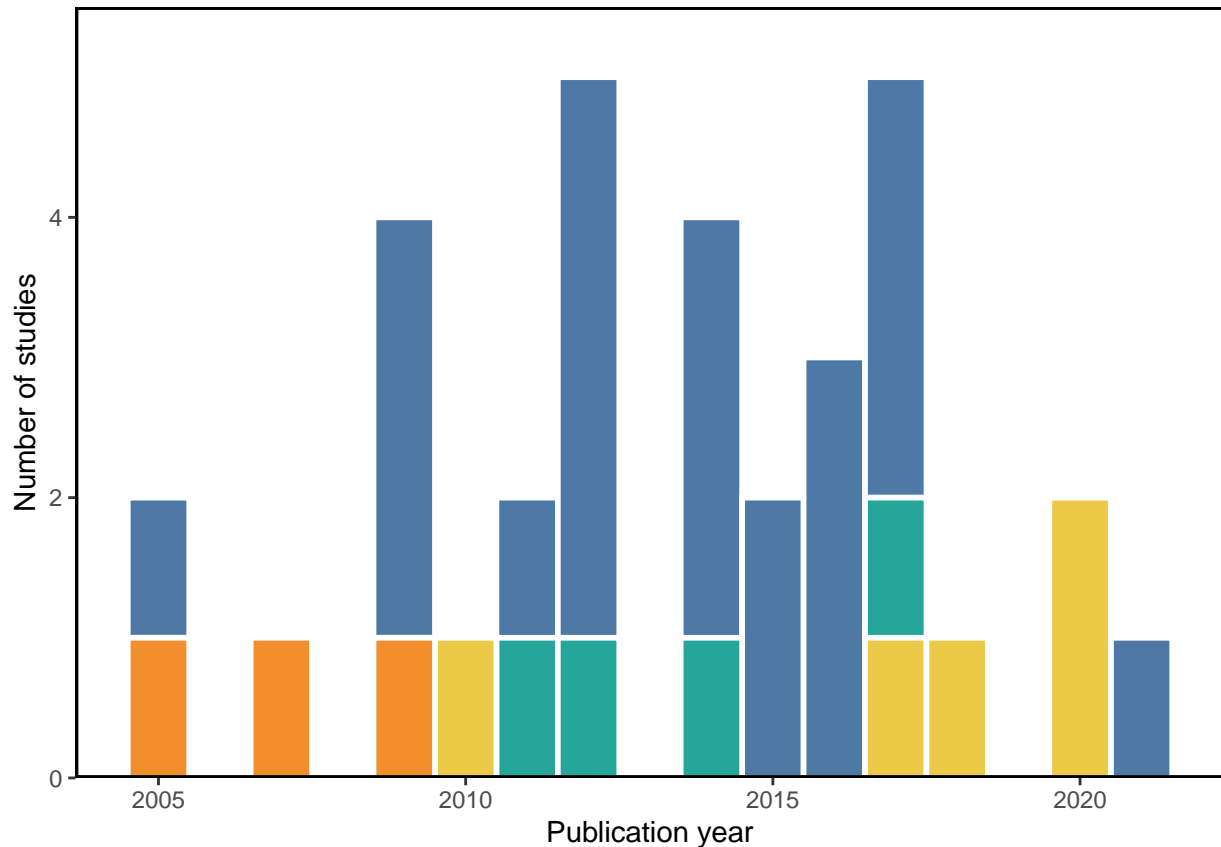


```
p1 +
  theme_classic() +
  theme(axis.line = element_blank(),
        panel.border = element_rect(size = 1, fill = "transparent"),
        legend.title = element_blank(),
        legend.position = c(.15,.8),
        plot.margin = margin(.2,.1,.1,.4, unit = "cm"))
```



```
Fig2_theme = theme_classic() +
  theme(axis.line = element_blank(),
        panel.border = element_rect(size = 1, fill = "transparent"),
        legend.title = element_blank(),
        legend.position = "none",
        legend.background = element_rect(fill = "transparent"),
        plot.margin = margin(.2,.1,.1,.4, unit = "cm"))
### add elements

p1 = p1 + Fig2_theme
p1
```



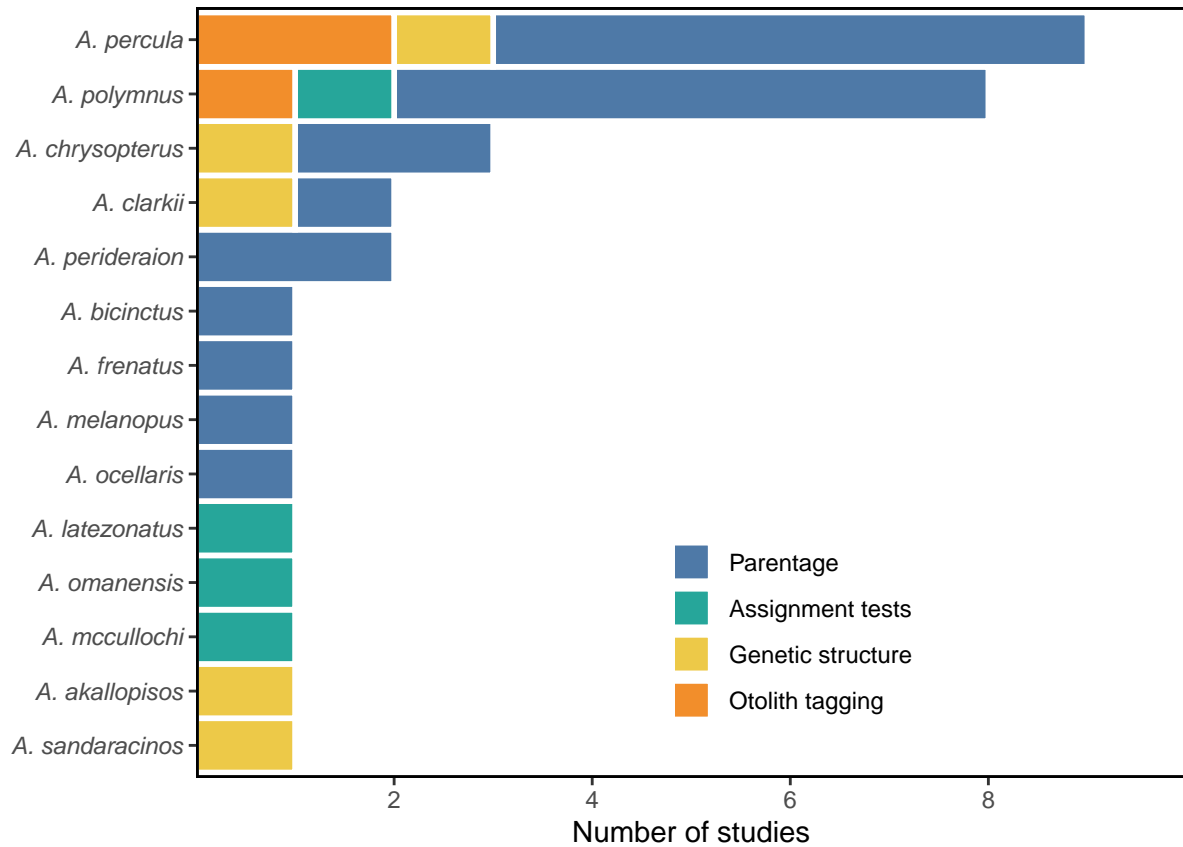
p2

```
my_species = c("A. percula", "A. polymnus", "A. chrysopterus", "A. clarkii",
               "A. perideraion", "A. bicinctus", "A. frenatus", "A. melanopus", "A. ocellaris", "A. latezonatus",
               "A. omanensis", "A. mccullochi", "A. akallopisos", "A. sandaracinos")

p2 = ggplot(nemo_dat %>% mutate(Species = factor(Species, levels = rev(my_species))),
            aes(x = Species, fill = Method)) +
  geom_histogram(stat = "count", width = 1, col = "white", size = 1) +
  scale_fill_manual(values = c("#4e79a7", "#26A69A", "#edc948", "#f28e2b")) +
  labs(x = "", y = "Number of studies") +
  scale_y_continuous(expand = c(0,0), breaks = c(2,4,6,8), limits = c(0, 10)) +
  coord_flip() + ## flip it!
  Fig2_theme +
  theme(axis.text.y = element_text(face = "italic"), #edit axis
        legend.position = c(.6,.2))
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

p2

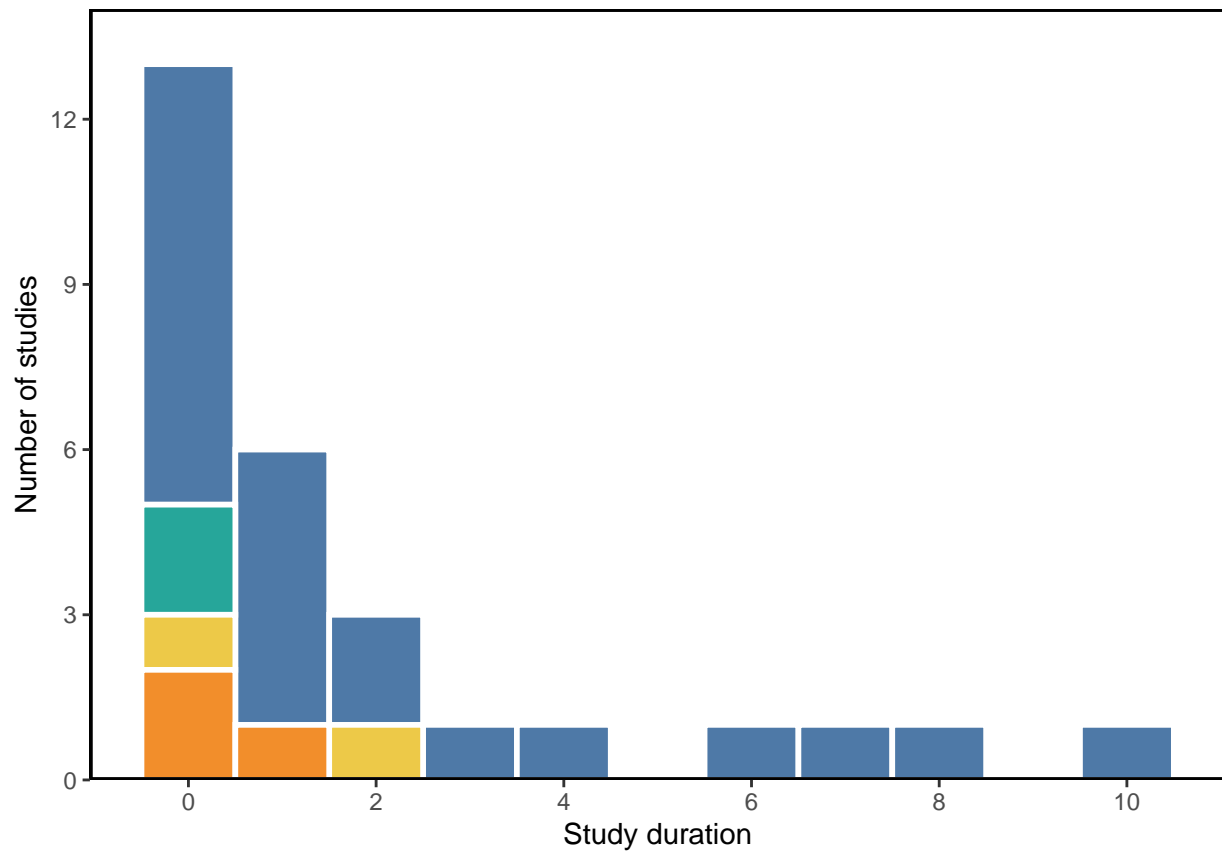


p3

```
p3 = ggplot(nemo_dat, aes(x = Time_ya, fill = Method)) +
  geom_histogram(binwidth = 1, col = "white", size = 1) +
  labs(x = "Study duration", y = "Number of studies") +
  scale_fill_manual(values = c("#4e79a7", "#26A69A", "#edc948", "#f28e2b")) +
  scale_y_continuous(expand = c(0,0), limits = c(0,14), breaks = c(0,3,6,9,12)) +
  scale_x_continuous(breaks = c(0,2,4,6,8,10)) +
  Fig2_theme
```

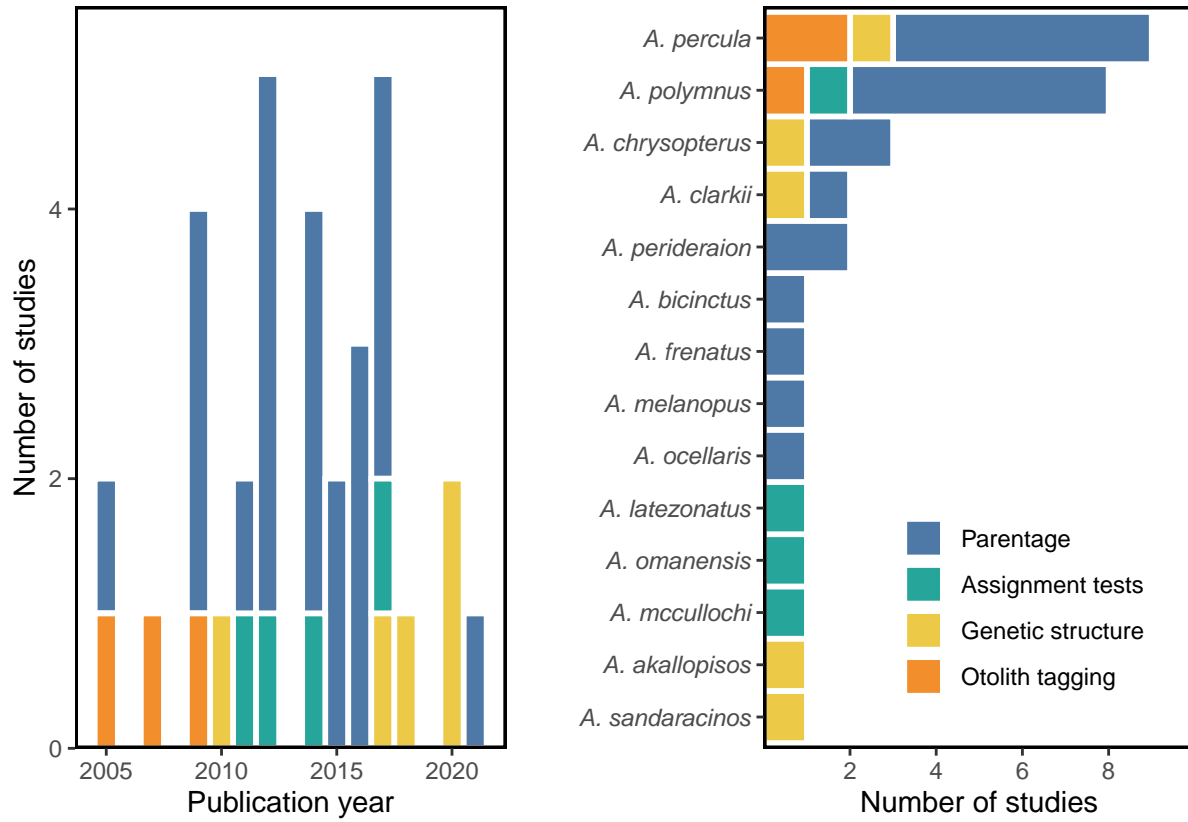
p3

```
## Warning: Removed 5 rows containing non-finite values (stat_bin).
```

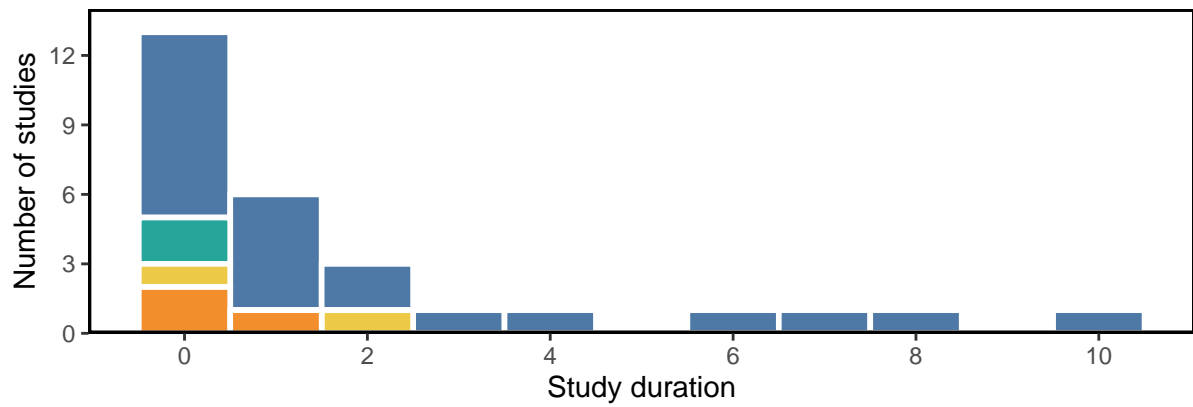
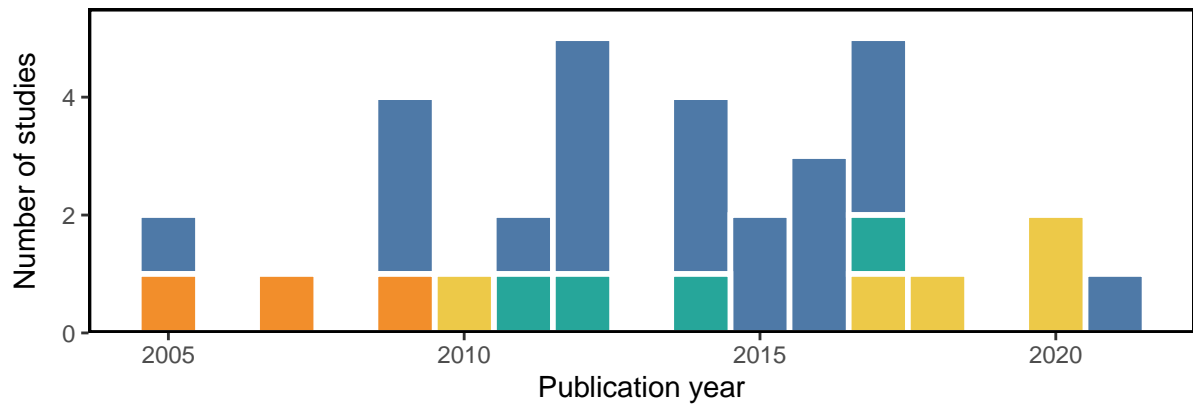
patchwork

p1 + p2



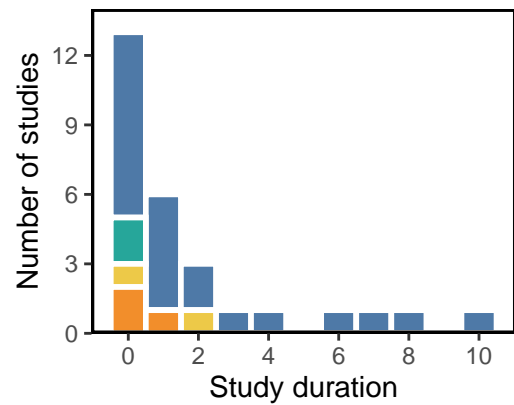
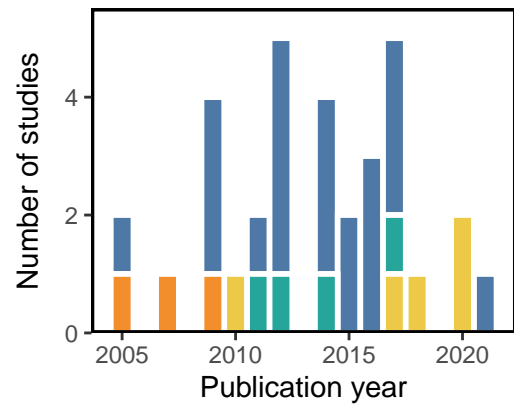
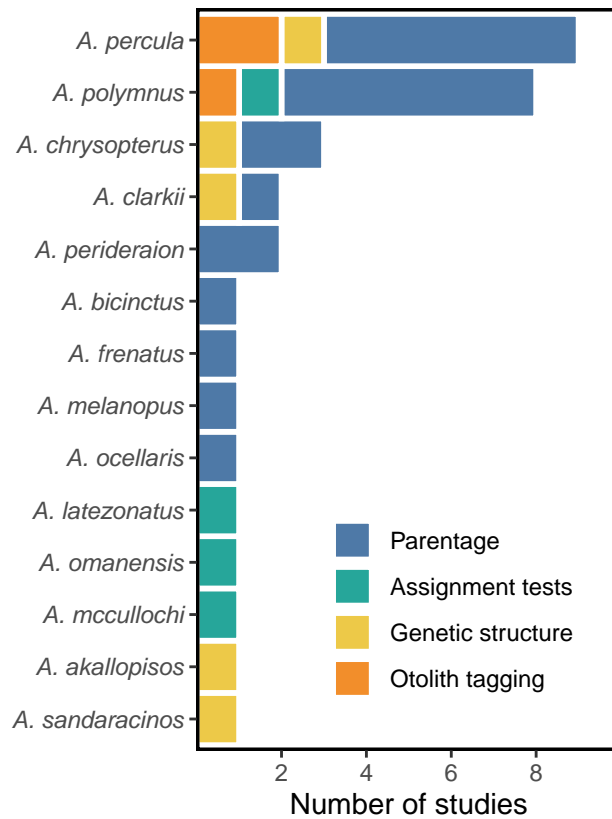
p1/p3

Warning: Removed 5 rows containing non-finite values (stat_bin).



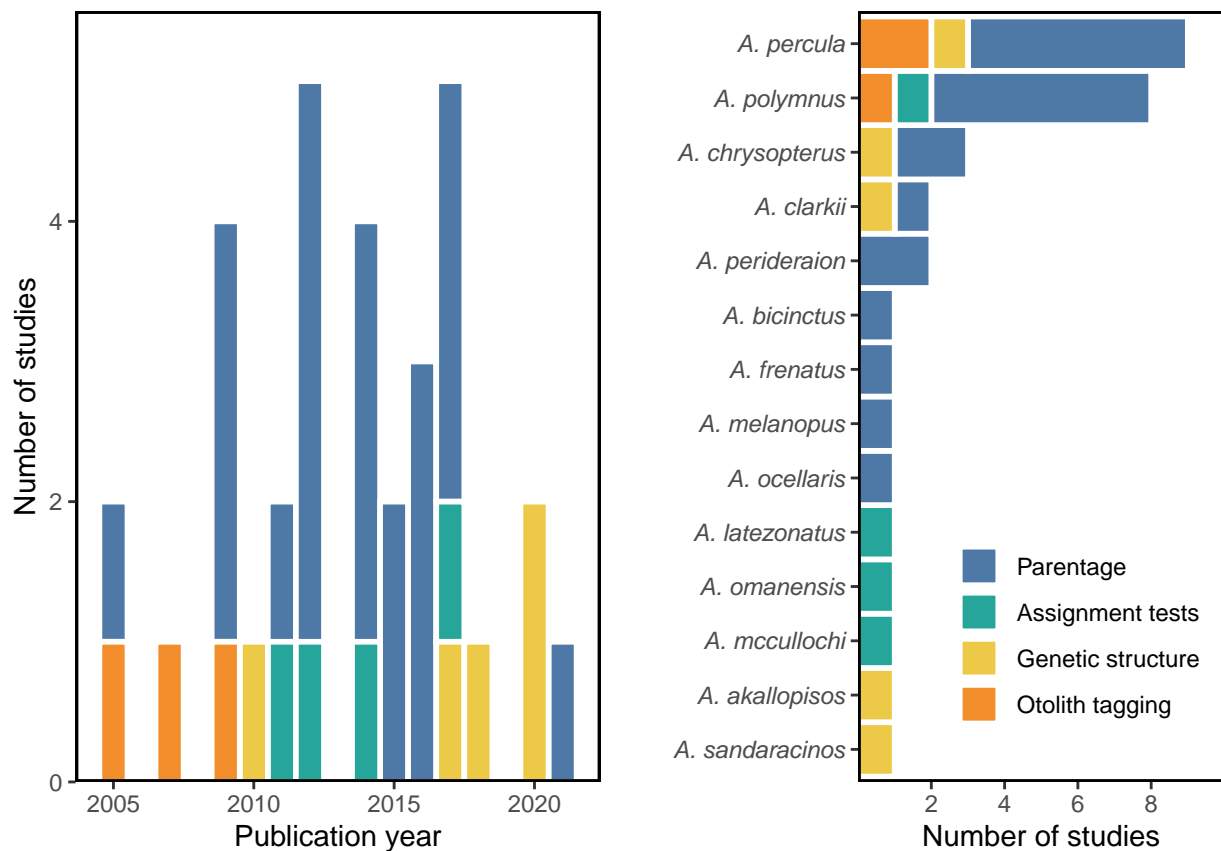
p2 + p1/p3

Warning: Removed 5 rows containing non-finite values (stat_bin).



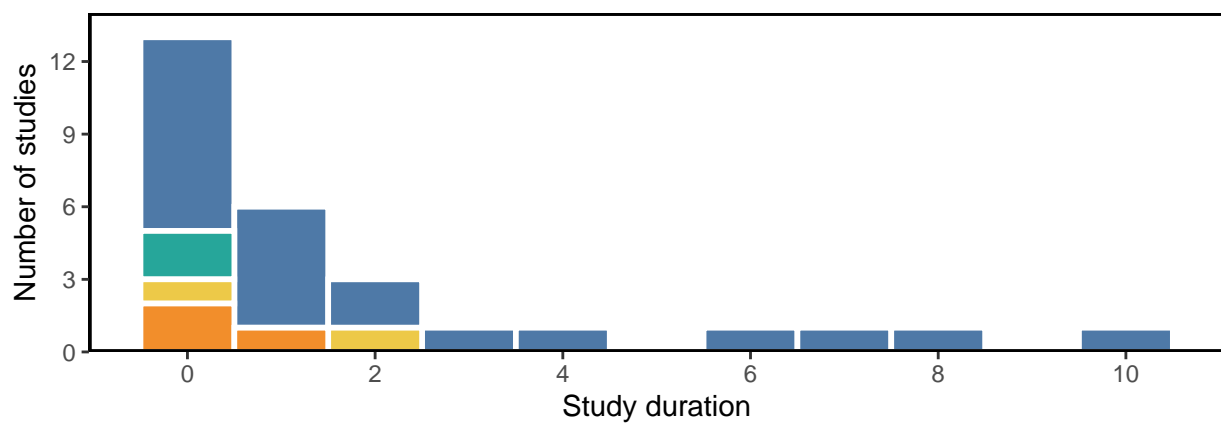
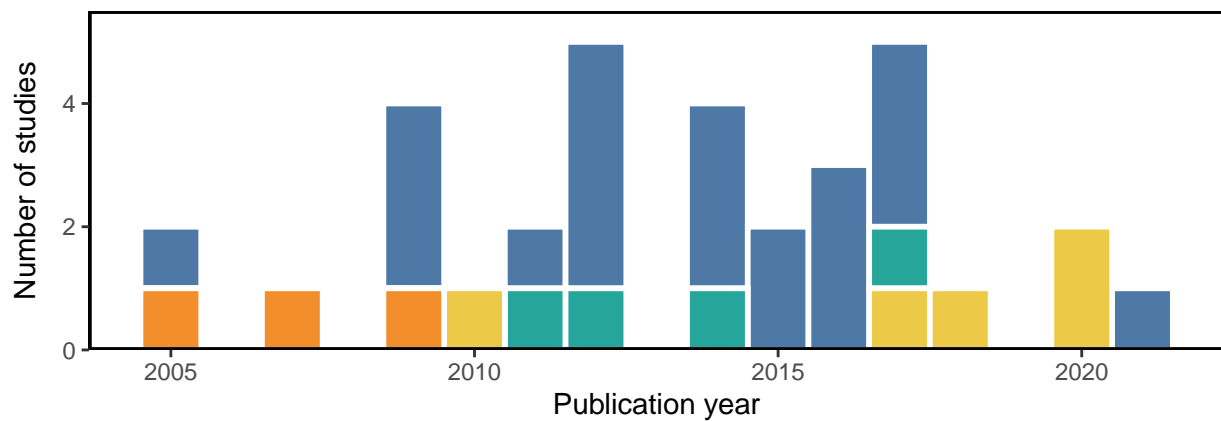
ggpubr

```
ggarrange(p1,p2, align = "h")
```



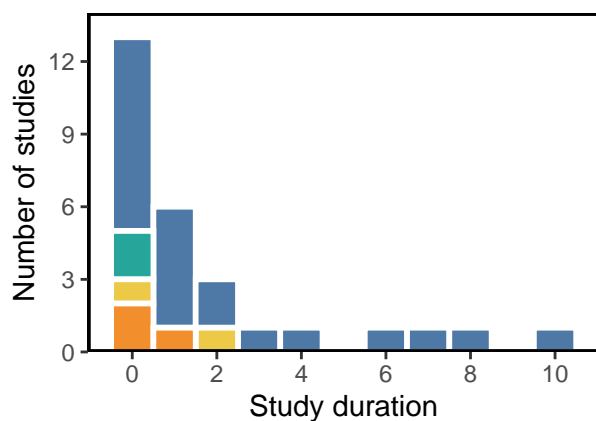
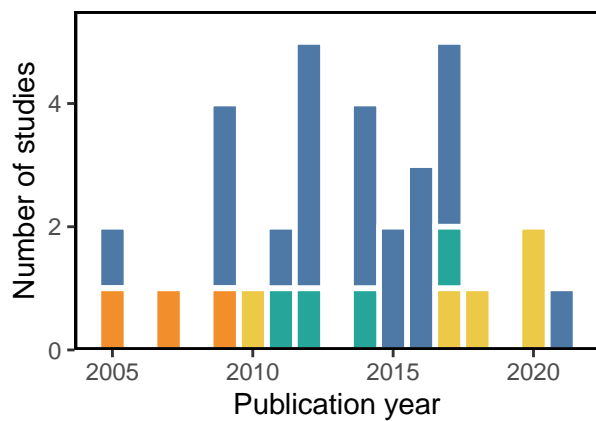
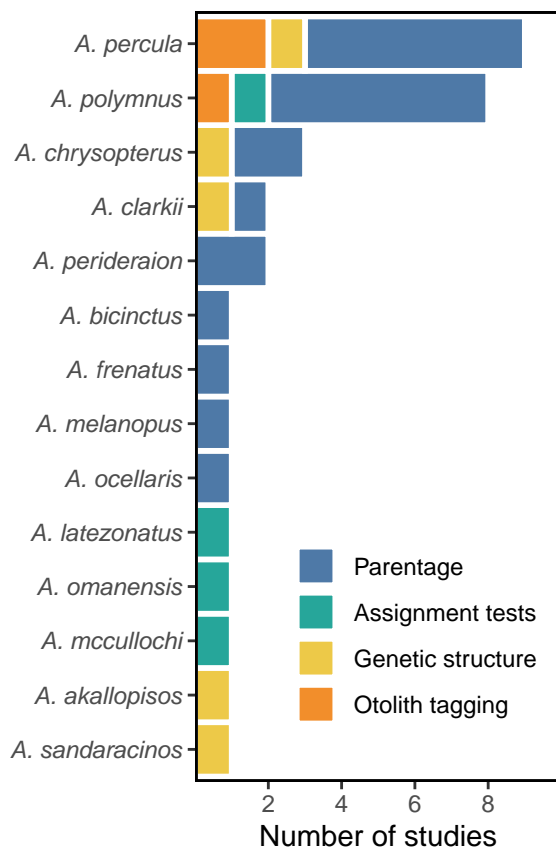
```
ggarrange(p1,p3, nrow = 2, align = "v")
```

```
## Warning: Removed 5 rows containing non-finite values (stat_bin).
```



```
ggarrange(p2,
  ggarrange(p1,p3, nrow = 2), ncol = 2)
```

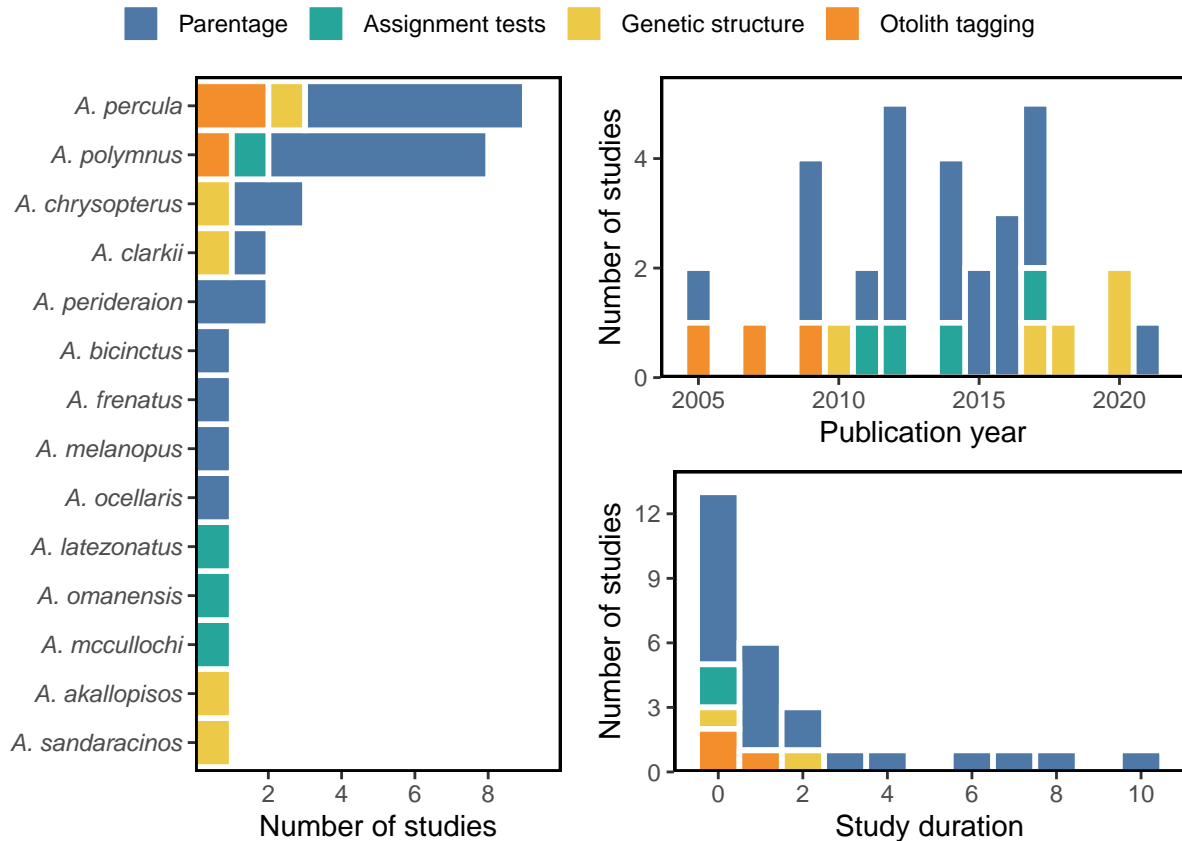
```
## Warning: Removed 5 rows containing non-finite values (stat_bin).
```



Shared legends

```
ggarrange(p2,
  ggarrange(p1,p3, nrow = 2, legend = "none"), ncol = 2, common.legend = TRUE, legend = "top" )
```

```
## Warning: Removed 5 rows containing non-finite values (stat_bin).
```

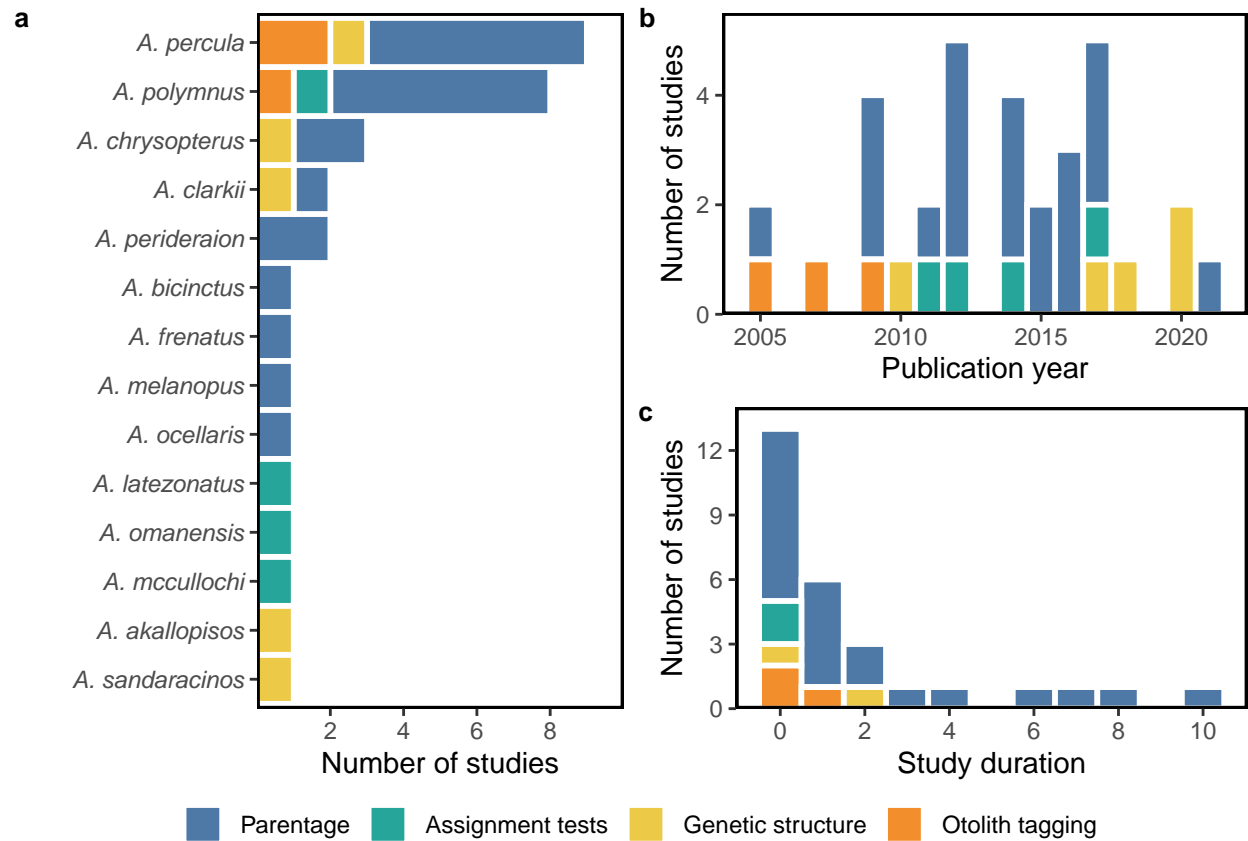


Panel labels

```
right.panel = ggarrange(p1,p3, nrow = 2, legend = "none",
                        labels = c("b", "c"),
                        font.label = list(size = 10, color = "black", face = "bold", family = "Helvetica"))
```

```
## Warning: Removed 5 rows containing non-finite values (stat_bin).
```

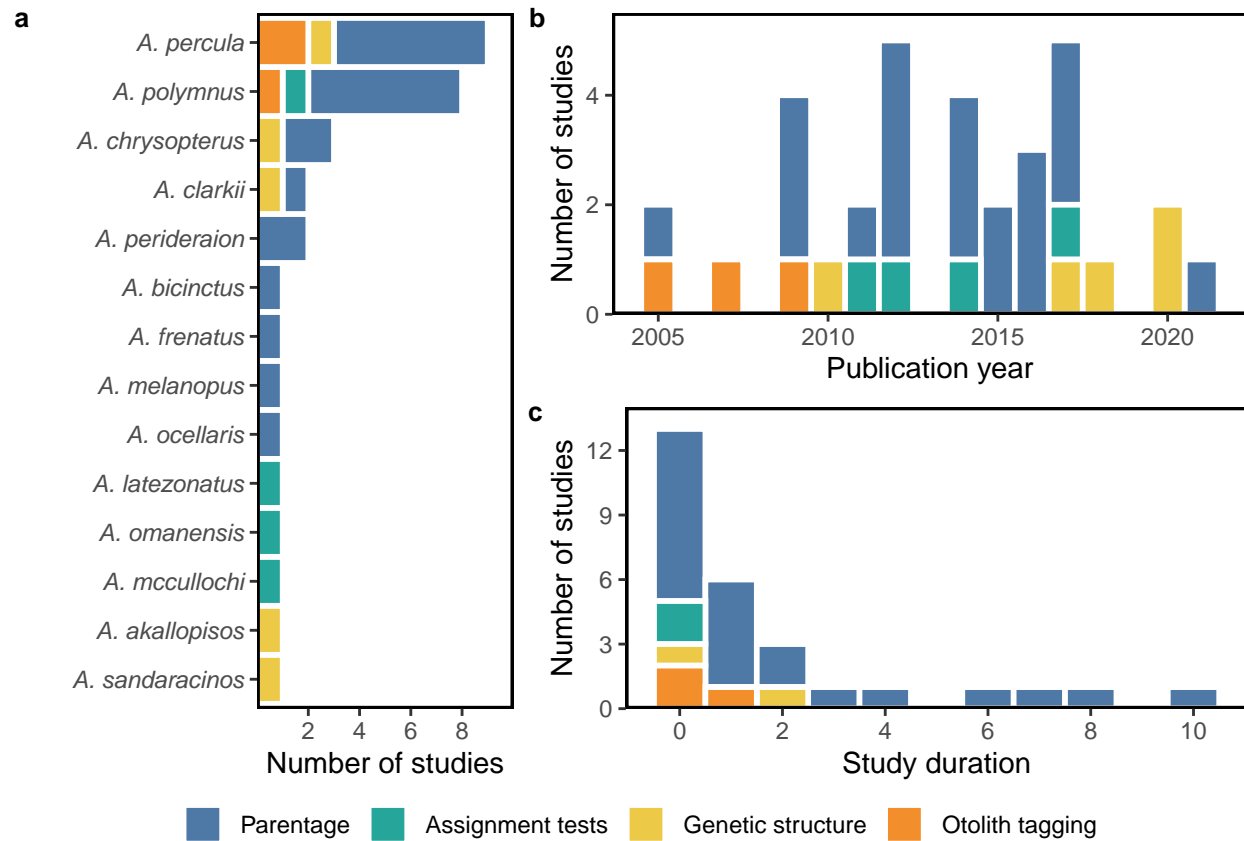
```
ggarrange(p2, right.panel, ncol = 2,
          common.legend = TRUE, legend = "bottom",
          labels = c("a"),
          font.label = list(size = 10, color = "black", face = "bold", family = "Helvetica", position = "bottom"))
```

Widths and heights

```
Fig2 = ggarrange(p2, right.panel, ncol = 2, widths = c(.7,1),
  common.legend = TRUE, legend = "bottom",
  labels = c("a"),
  font.label = list(size = 10, color = "black", face = "bold", family = "Helvetica", pos
```

Fig2



I love it!

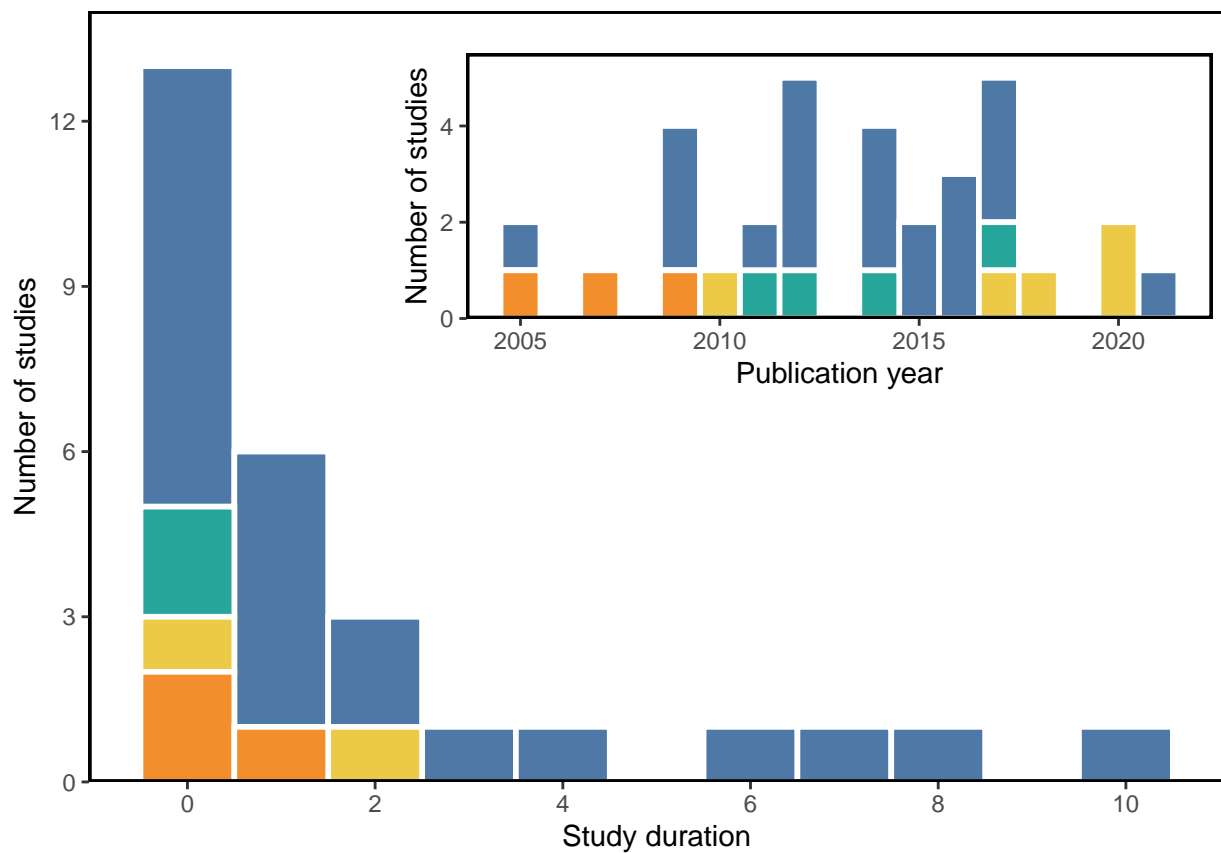
```
ggsave("./PublicationFigures/figures/Fig2.1.pdf", plot = Fig2, width = 174, height = 100, units = "mm",
```

Fig 3. Complex figures

a. Insets

```
p3 + annotation_custom(
  ggplotGrob(p1),
  xmin = 2, xmax = 11, ymin = 7, ymax = 13.5)
```

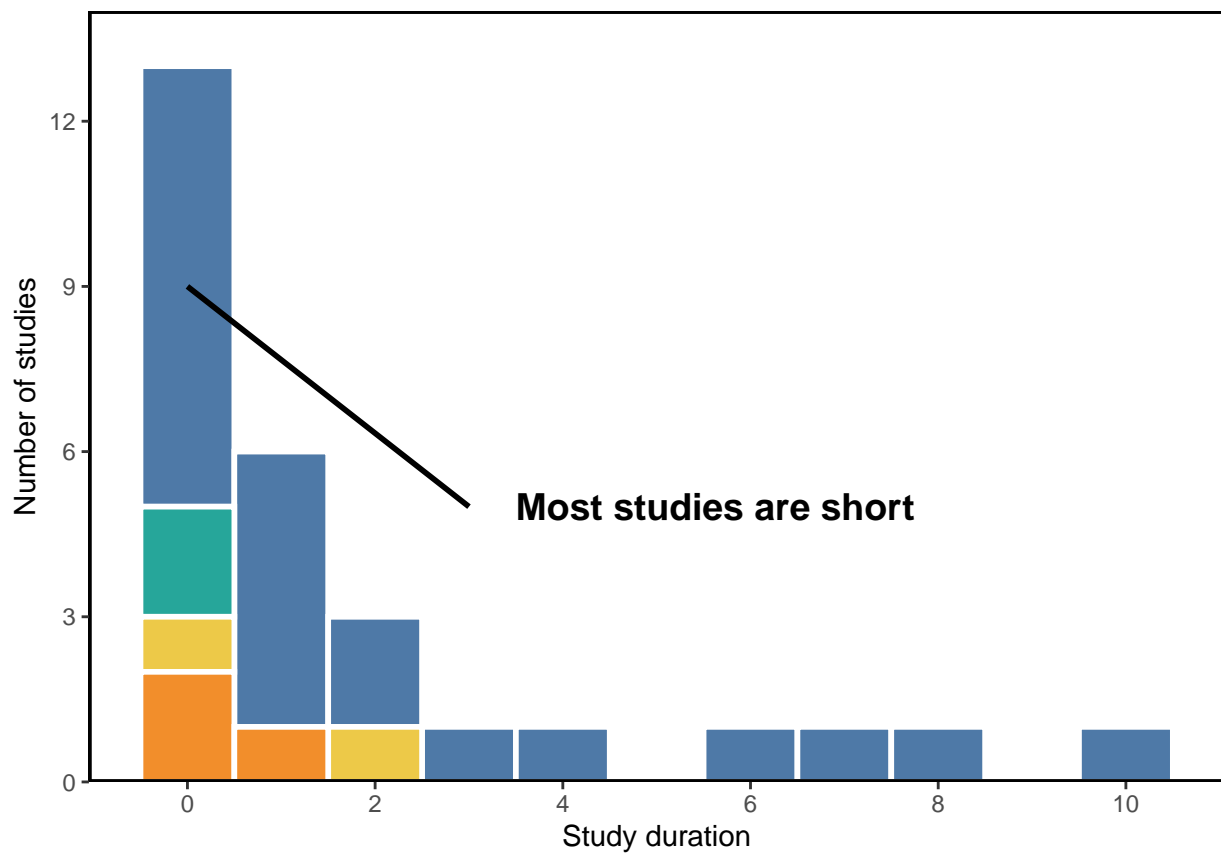
```
## Warning: Removed 5 rows containing non-finite values (stat_bin).
```



b. annotations

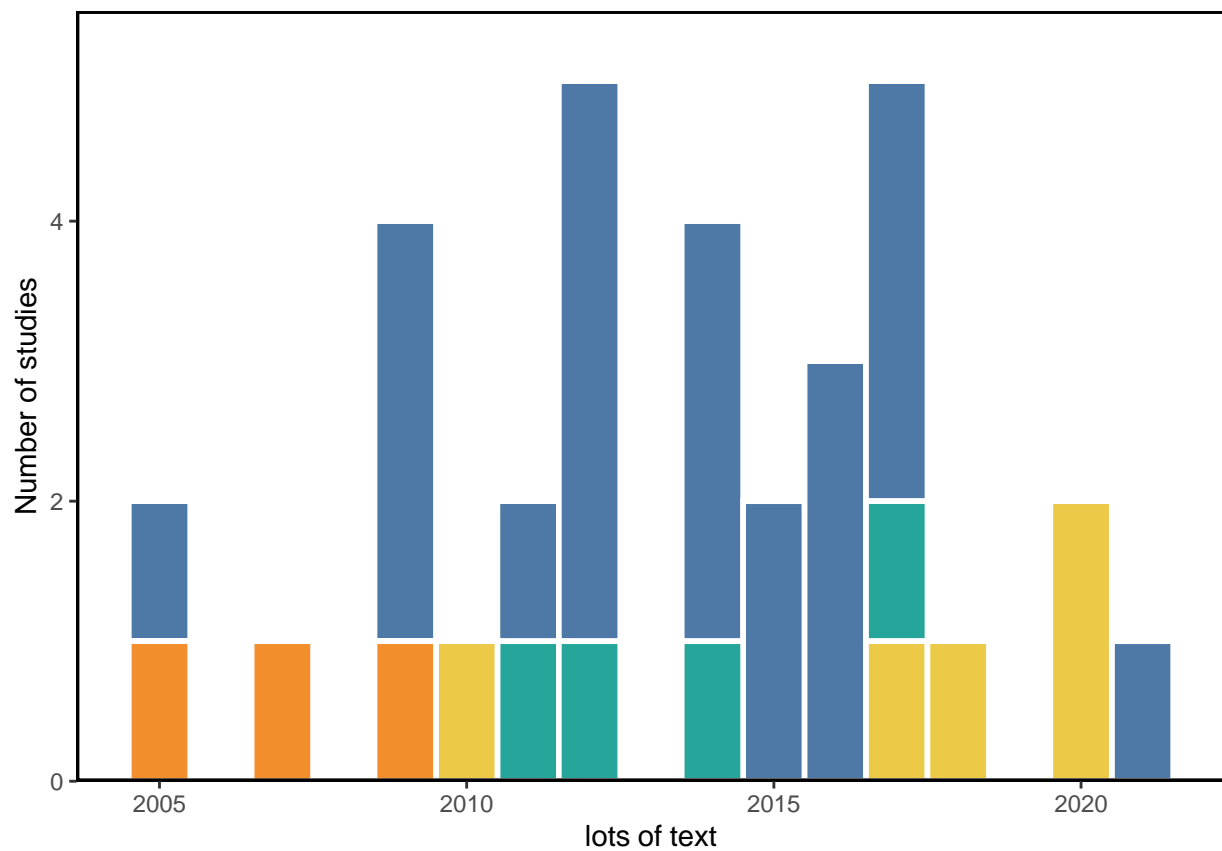
```
p3 + annotate("segment", x = 0, xend = 3, y = 9, yend = 5, colour = "black", size = 1) +
  annotate("text", x = 3.5, y = 5, label = "Most studies are short",
    colour = "black", size = 5, fontface = "bold", family = "Helvetica", hjust = 0)
```

```
## Warning: Removed 5 rows containing non-finite values (stat_bin).
```



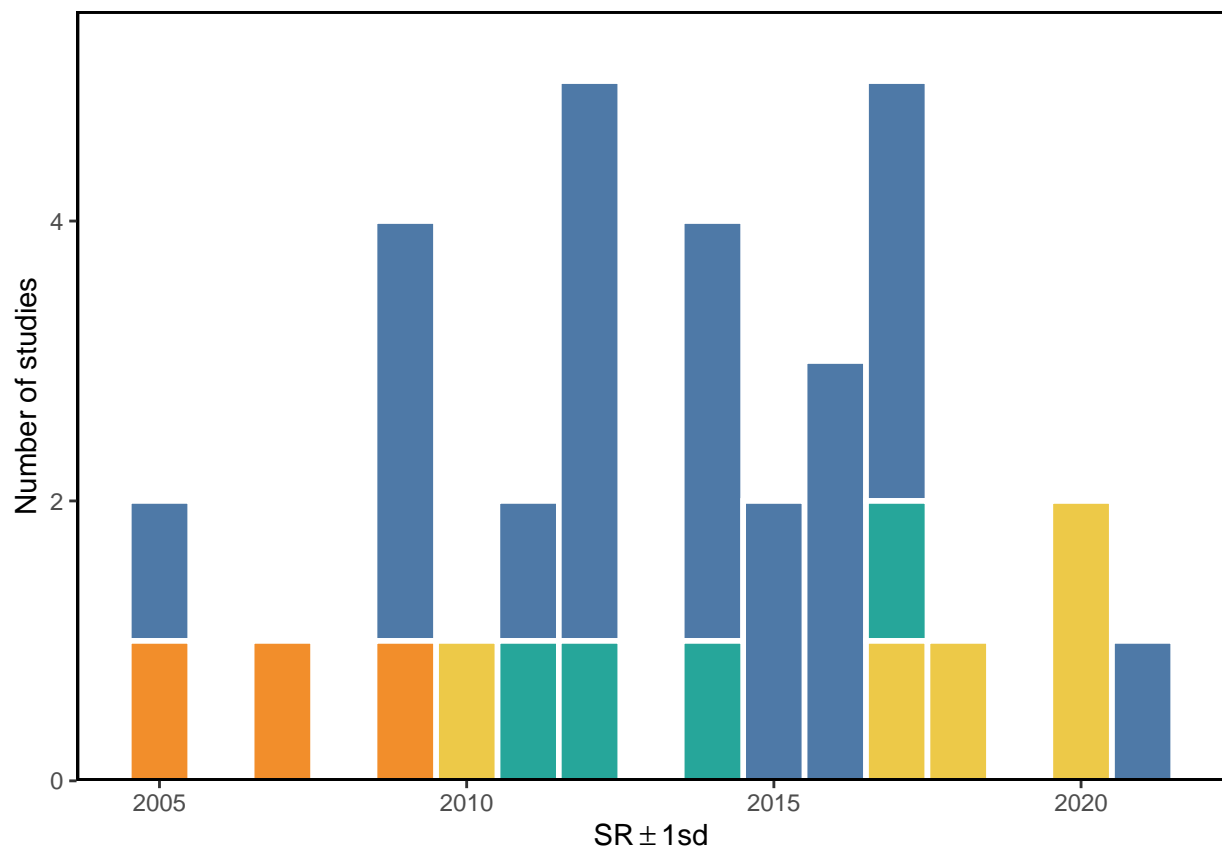
c. difficult text

```
#text  
p1 + labs(x = expression(lots~of~text))
```

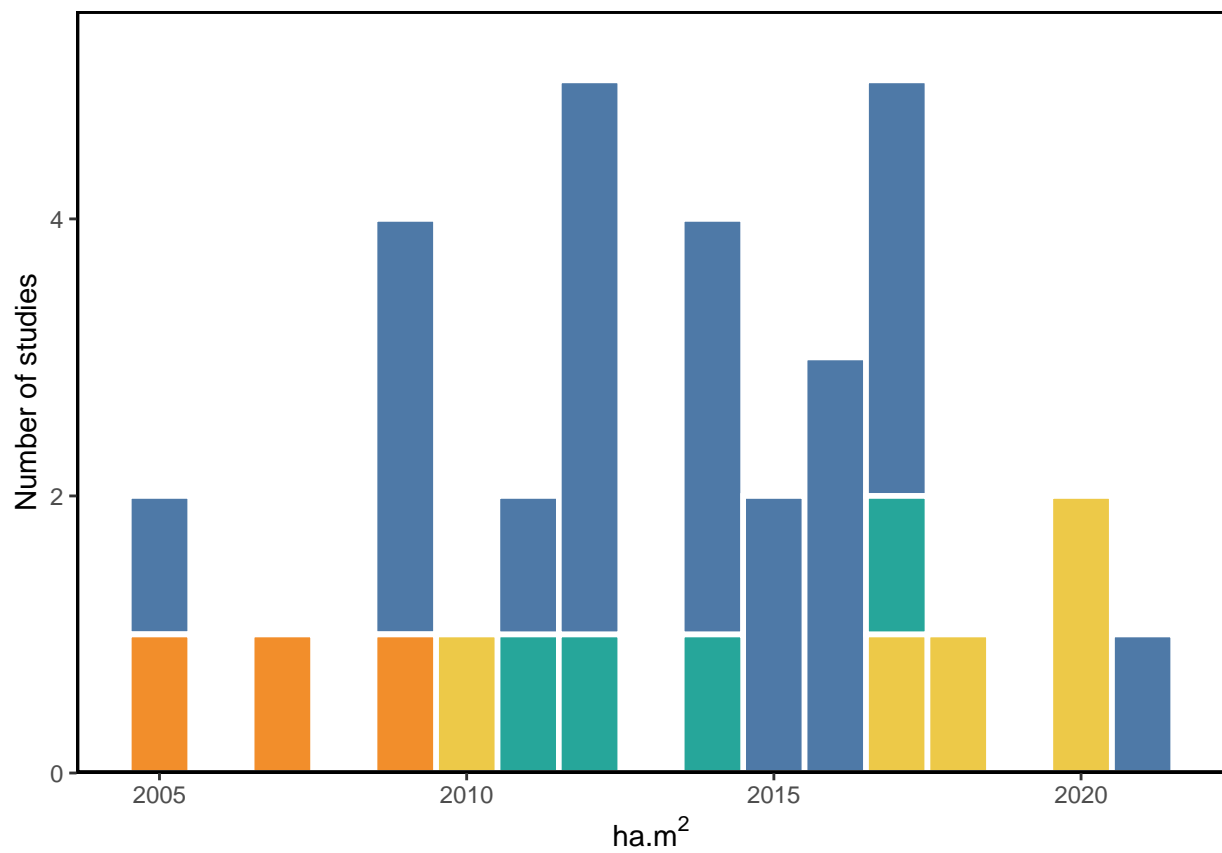


#Symbols

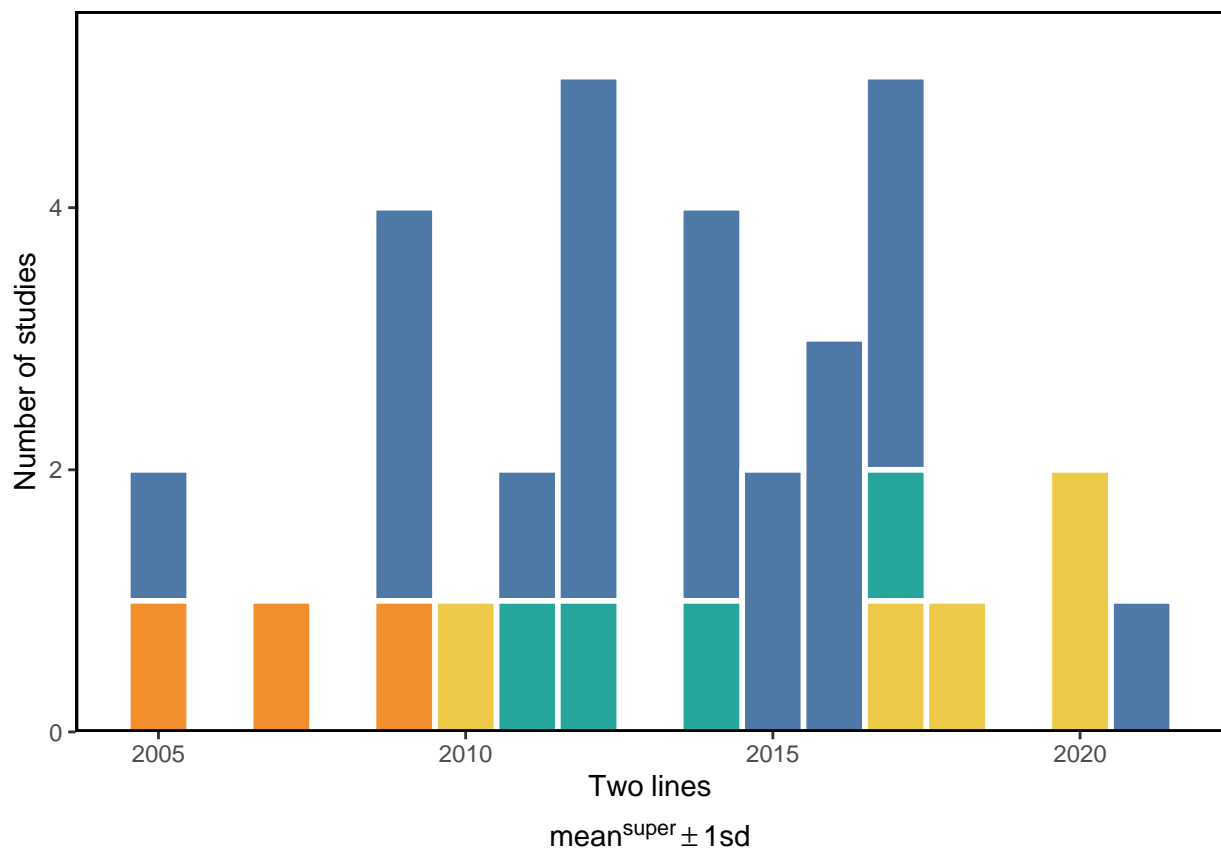
```
p1 + labs(x = expression("SR" %+-% "1sd" ))
```



```
#superscript
p1 + labs(x = expression(ha.m2))
```



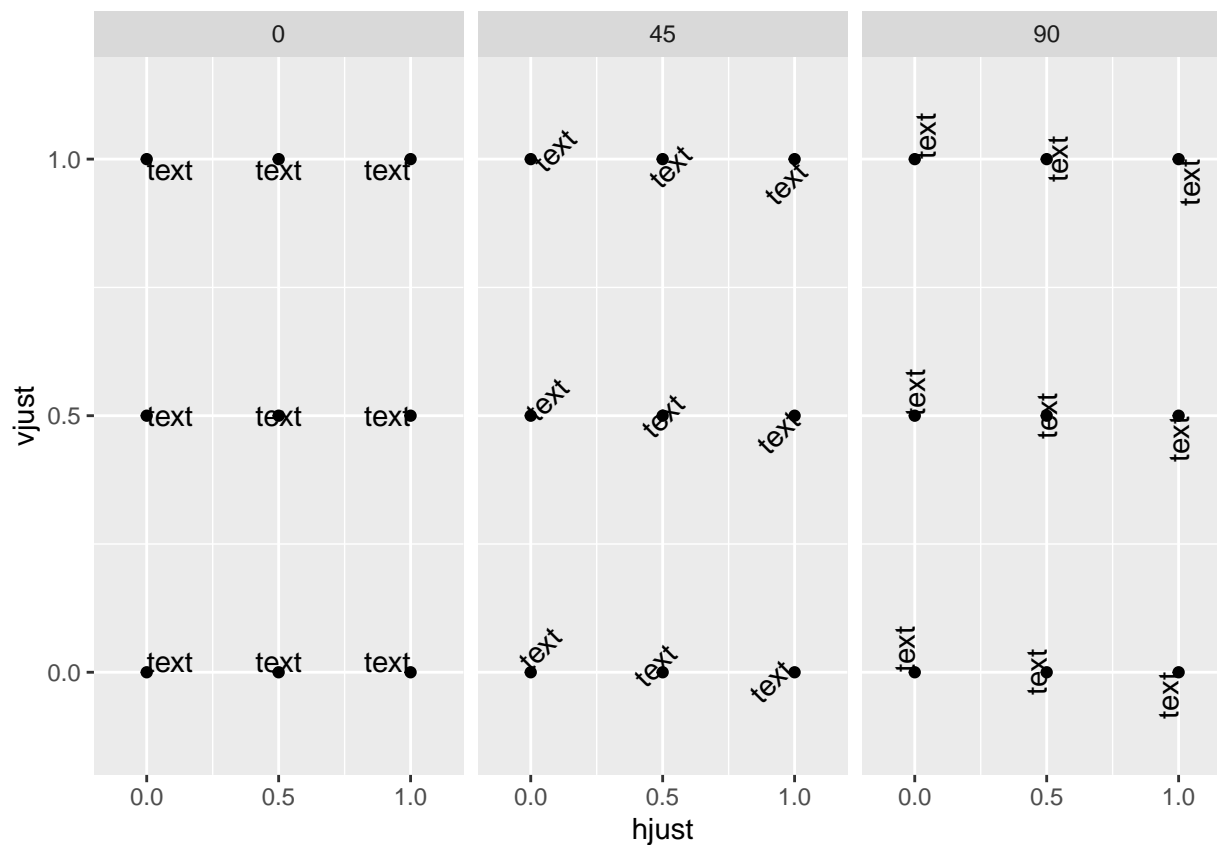
```
#multiple lines
p1 + labs(x = expression(atop(Two~lines, mean^super %+-% "1sd")))
```



d. text justification

```
td <- expand.grid(
  hjust=c(0, 0.5, 1),
  vjust=c(0, 0.5, 1),
  angle=c(0, 45, 90),
  text="text"
)

ggplot(td, aes(x=hjust, y=vjust)) +
  geom_point() +
  geom_text(aes(label=text, angle=angle, hjust=hjust, vjust=vjust)) +
  facet_grid(~angle) +
  scale_x_continuous(breaks=c(0, 0.5, 1), expand=c(0, 0.2)) +
  scale_y_continuous(breaks=c(0, 0.5, 1), expand=c(0, 0.2))
```

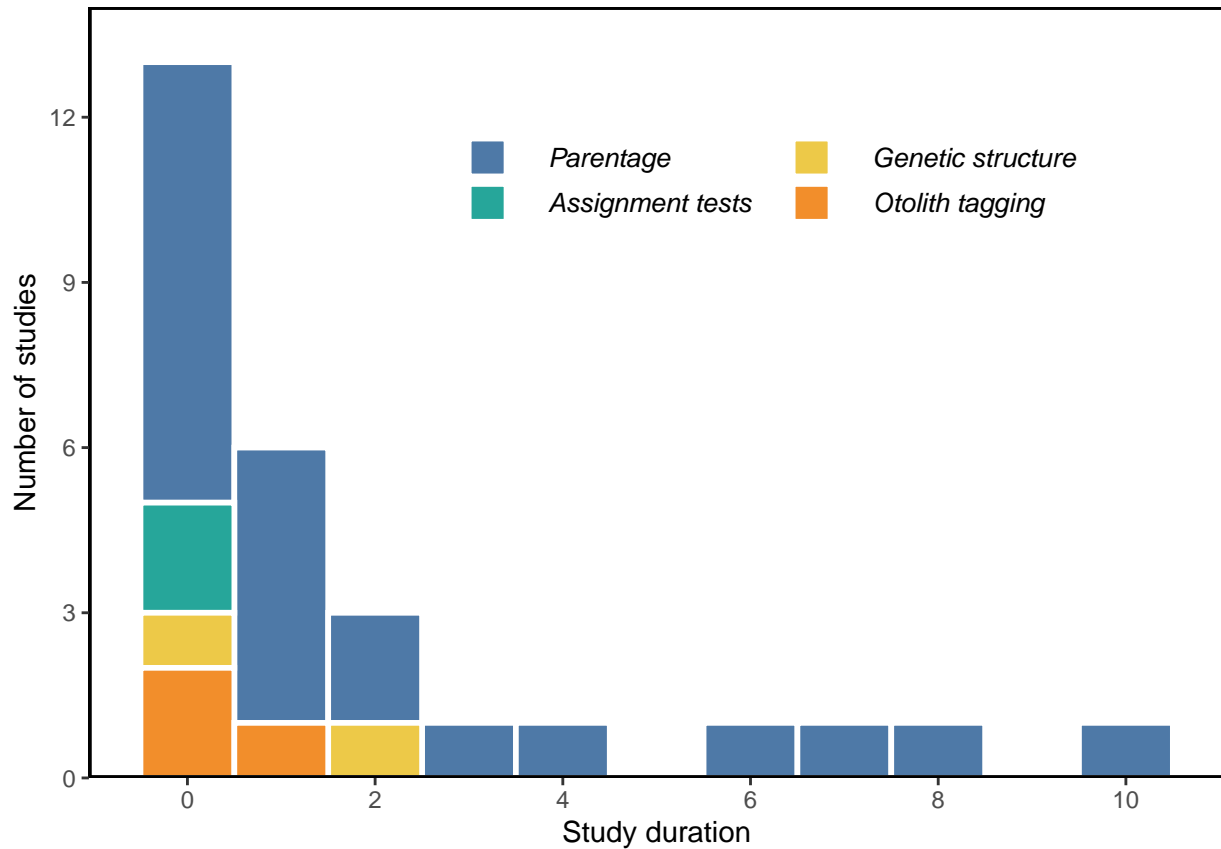



e. difficult legends

```
#Edit the legend
p3.legend = p3 +
  theme(legend.background = element_rect(fill = "transparent"),
        legend.text = element_text(face = "italic", size = 10),
        legend.title = element_blank(),
        legend.position = c(.6,.8),
        legend.spacing.x = unit(.5, 'cm'),
        legend.spacing.y = unit(.5, 'cm'),
        legend.key = element_rect(colour = "white",size = 2),
        legend.key.size = unit(.6, "cm")) +
  guides(fill=guide_legend(nrow=2, byrow=FALSE))

p3.legend
```

```
## Warning: Removed 5 rows containing non-finite values (stat_bin).
```

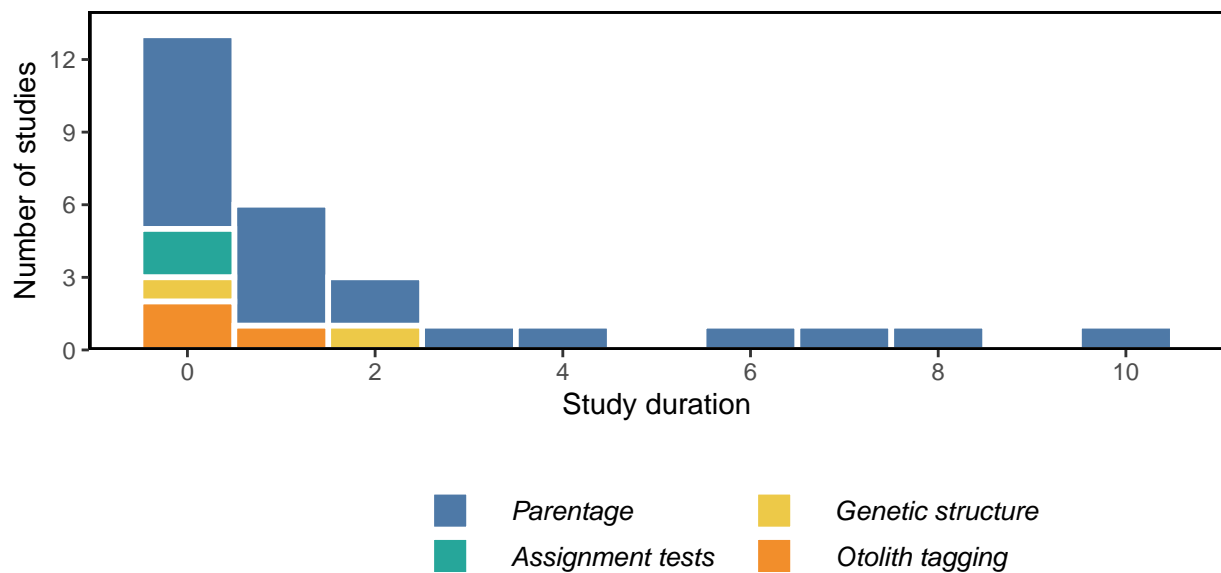


```
#Extract the legend
legend = as_ggplot(get_legend(p3.legend))
```

```
## Warning: Removed 5 rows containing non-finite values (stat_bin).
```

```
#plot the legend like a plot
ggarrange(p3, legend, nrow = 2)
```

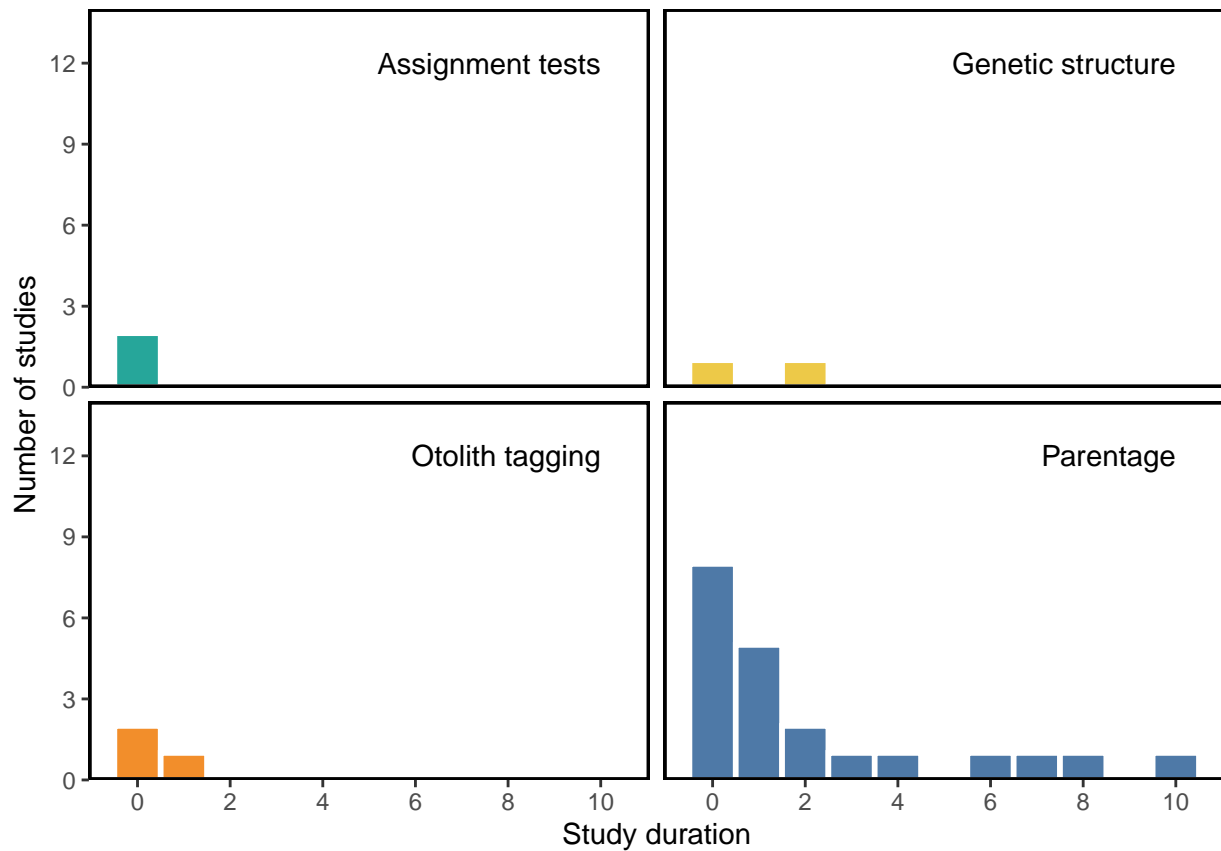
```
## Warning: Removed 5 rows containing non-finite values (stat_bin).
```



f. facetting

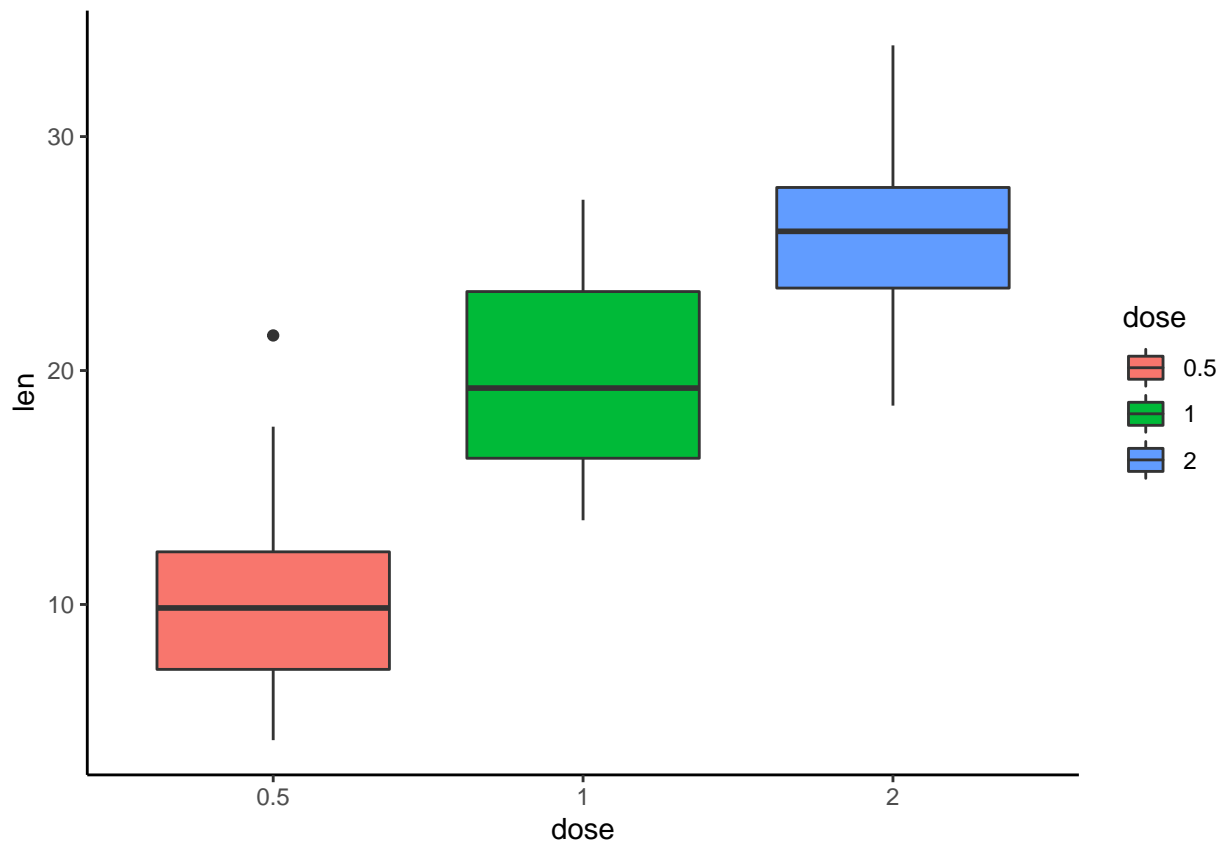
```
p3 + facet_wrap(~Method) +
  theme(strip.background = element_blank(),
        strip.text = element_blank()) +
  geom_text(data = data.frame(Method = c("Parentage", "Assignment tests", "Genetic structure", "Otolith tagging"),
    aes(x = 10, y = 12, label = Method), hjust = 1)
```

```
## Warning: Removed 5 rows containing non-finite values (stat_bin).
```



```
data("ToothGrowth")

ToothGrowth = ToothGrowth %>%
  mutate(dose = as.factor(dose))
p <- ggplot(ToothGrowth, aes(x = dose, y = len, fill = dose)) +
  geom_boxplot() +
  theme_classic()
p
```

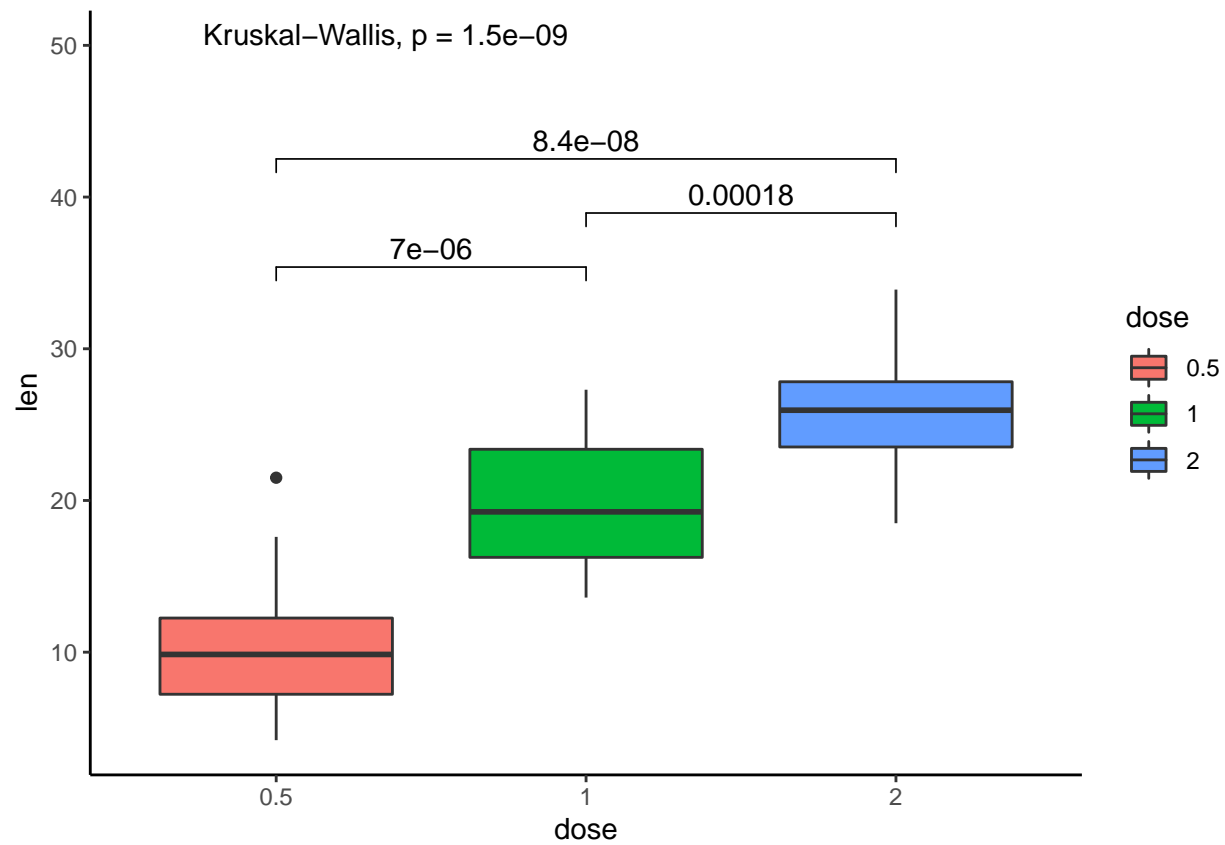


```
# Add p-values comparing groups
# Specify the comparisons you want
my_comparisons <- list( c("0.5", "1"), c("1", "2"), c("0.5", "2") )
p + stat_compare_means(comparisons = my_comparisons)+ # Add pairwise comparisons p-value
  stat_compare_means(label.y = 50)                  # Add global p-value
```

```
## Warning in wilcox.test.default(c(4.2, 11.5, 7.3, 5.8, 6.4, 10, 11.2, 11.2, :
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(c(4.2, 11.5, 7.3, 5.8, 6.4, 10, 11.2, 11.2, :
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(c(16.5, 16.5, 15.2, 17.3, 22.5, 17.3, 13.6, :
## cannot compute exact p-value with ties
```



Playing with colour

Personalised colour palettes library(wesanderson) <https://colors.co/> <https://medialab.github.io/iwanthue/>