



HAL
open science

Rajouter des intervalles de confiance avec ggplot

Laurent Cauquil

► **To cite this version:**

| Laurent Cauquil. Rajouter des intervalles de confiance avec ggplot. 2024. hal-04603540

HAL Id: hal-04603540

<https://hal.inrae.fr/hal-04603540>

Submitted on 10 Jun 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



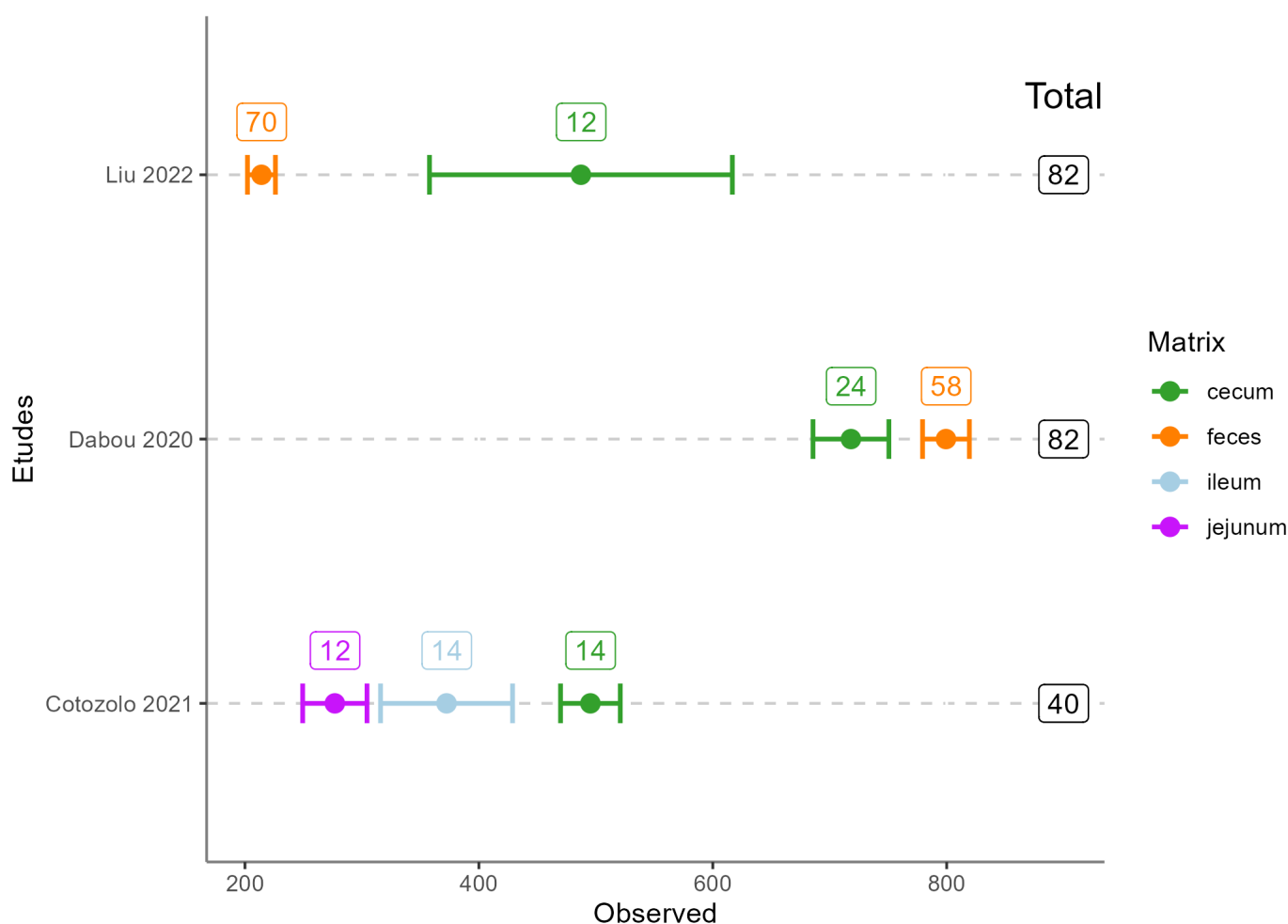
Open licence - etalab

Rajouter des intervalles de confiance avec ggplot

Objectif

On se propose de produire un graphique qui représente les moyennes et intervalles de confiance des mesures de richesse bactérienne d'échantillons issus de différents sites du tractus digestif de lapin à travers différentes études.

Graphique attendu:



Nous allons procéder par étapes:

- Affichage des moyennes sous forme de points
- Affichage des intervalles de confiance avec des barres
- Affichage du nombre total d'échantillons par matrice et par étude
- Affichage du nombre total d'échantillons par étude
- Changement des couleurs
- Customisation du graph

Packages

```
library(dplyr)

library(ggplot2)
library(ggrepel)
library(pals)
```

Jeu de données

Importation et aperçu des données

```
alpha_div <- readRDS("data/alpha_div.rds")
alpha_div
```

```
# A tibble: 204 × 3
  matrix study      Measure
  <chr> <chr>      <dbl>
1 feces Liu 2022      142
2 feces Liu 2022      231
3 feces Liu 2022      183
4 feces Liu 2022      197
5 feces Liu 2022      201
6 feces Liu 2022      235
7 feces Liu 2022      173
8 feces Liu 2022      273
9 feces Liu 2022      243
10 feces Liu 2022      232
# i 194 more rows
```

Il s'agit d'un data.frame avec 3 colonnes:

- matrix: compartiment de l'appareil digestif
- study: auteur et année de sortie de l'article
- Measure: richesse mesurée dans chaque échantillon

Table de contingence

La fonction `table()` nous permet d'afficher la table de contingence des échantillons répartis entre les matrices et les articles

```
table(alpha_div[, 1:2])
```

```
      study
matrix Cotozolo 2021 Dabou 2020 Liu 2022
```

cecum	14	24	12
feces	0	58	70
ileum	14	0	0
jejunum	12	0	0

Affichage des moyennes sous forme de points

Calcul des moyennes

On associe la fonction `group_by()` et `summarise()` pour calculer les moyennes par matrice et par étude

La variable `moy` est créée et la variable `Measure` disparaît de la table.

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE), .groups = NULL)
```

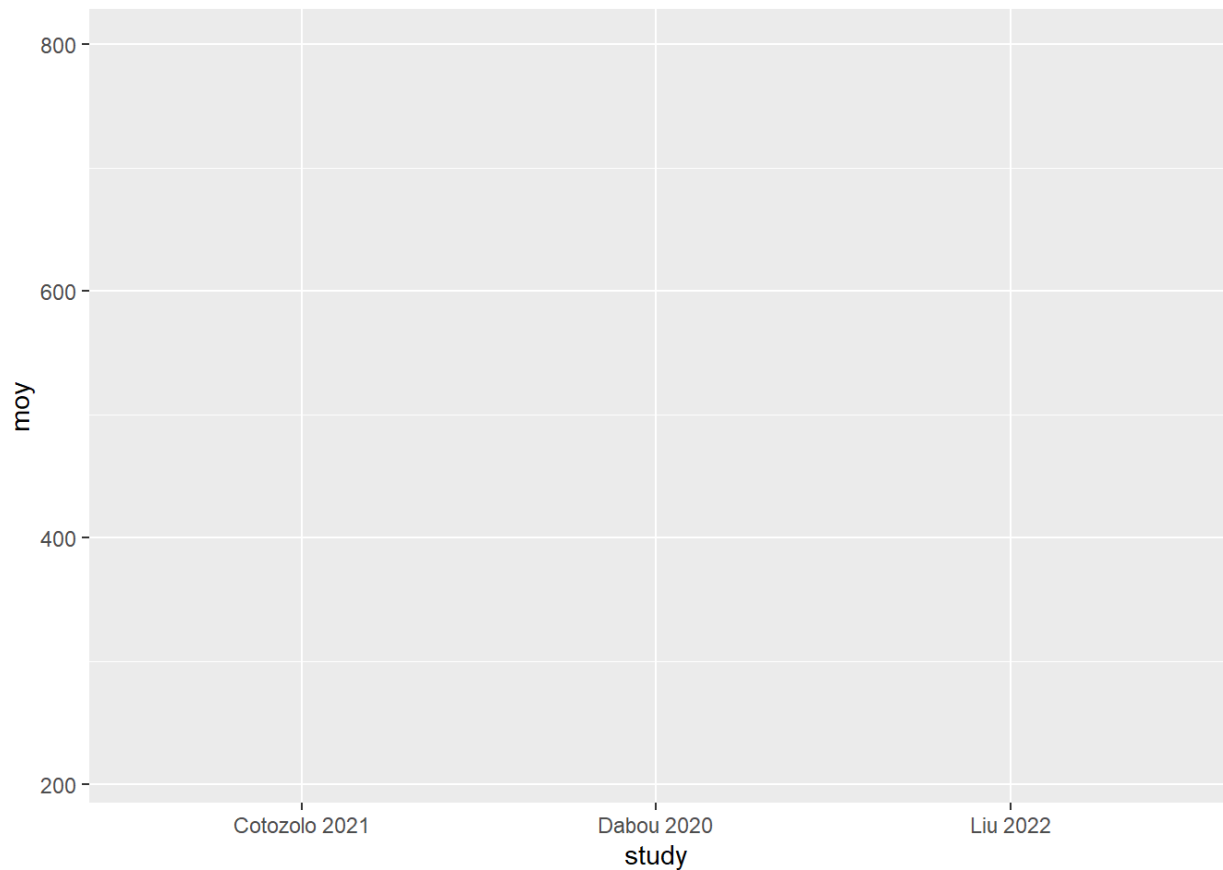
``summarise()`` has grouped output by 'study'. You can override using the ``.groups`` argument.

```
# A tibble: 7 × 3
# Groups:   study [3]
  study      matrix  moy
<chr>      <chr> <dbl>
1 Cotozolo 2021 cecum  495.
2 Cotozolo 2021 ileum  372.
3 Cotozolo 2021 jejunum 277.
4 Dabou 2020 cecum  718.
5 Dabou 2020 feces  799.
6 Liu 2022 cecum  487.
7 Liu 2022 feces  214.
```

Initialisation du graphique

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE)) |>
  ggplot(aes(x = study,
             y = moy))
```

``summarise()`` has grouped output by 'study'. You can override using the ``.groups`` argument.

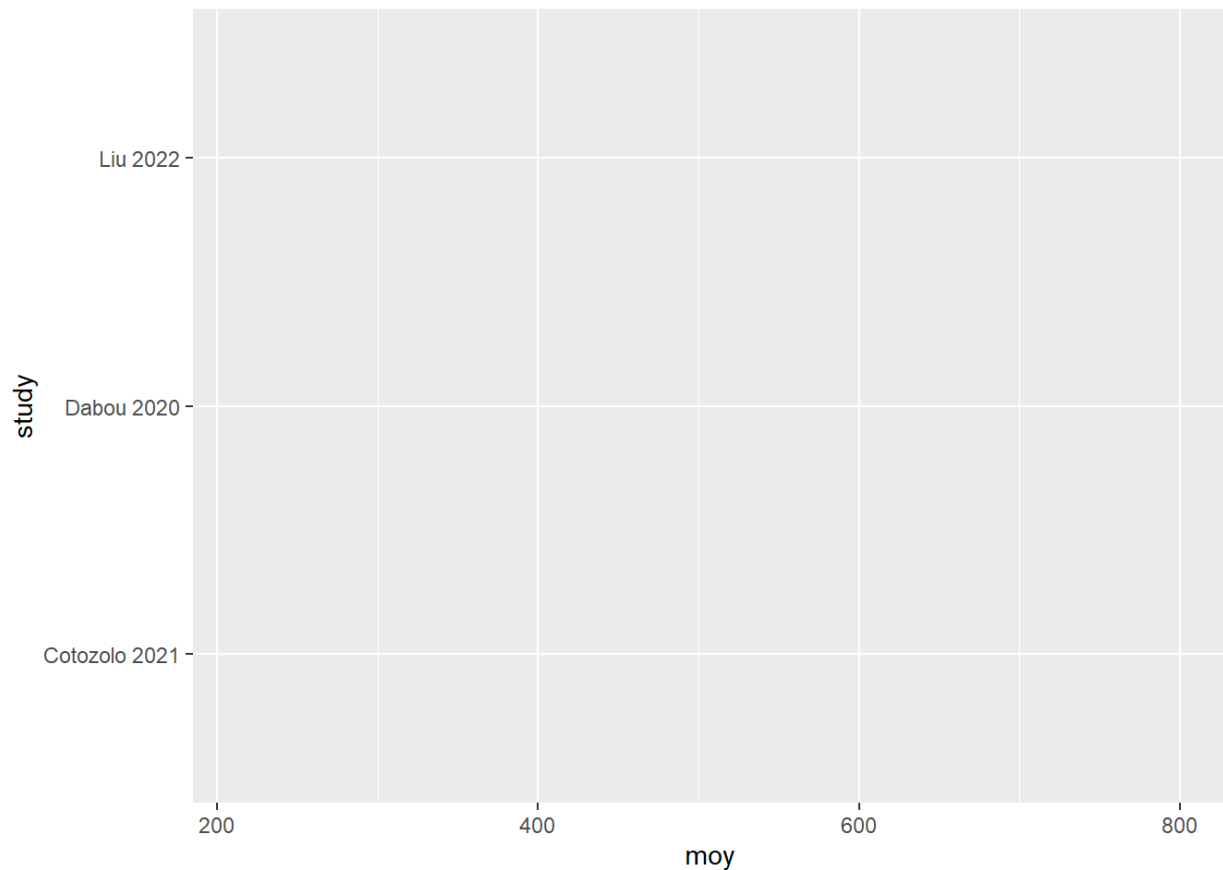


Rotation du graphique

La fonction `coord_flip()` “transpose” le graphique

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE)) |>
  ggplot(aes(x = study,
             y = moy)) +
  coord_flip()
```

``summarise()`` has grouped output by 'study'. You can override using the ``.groups`` argument.

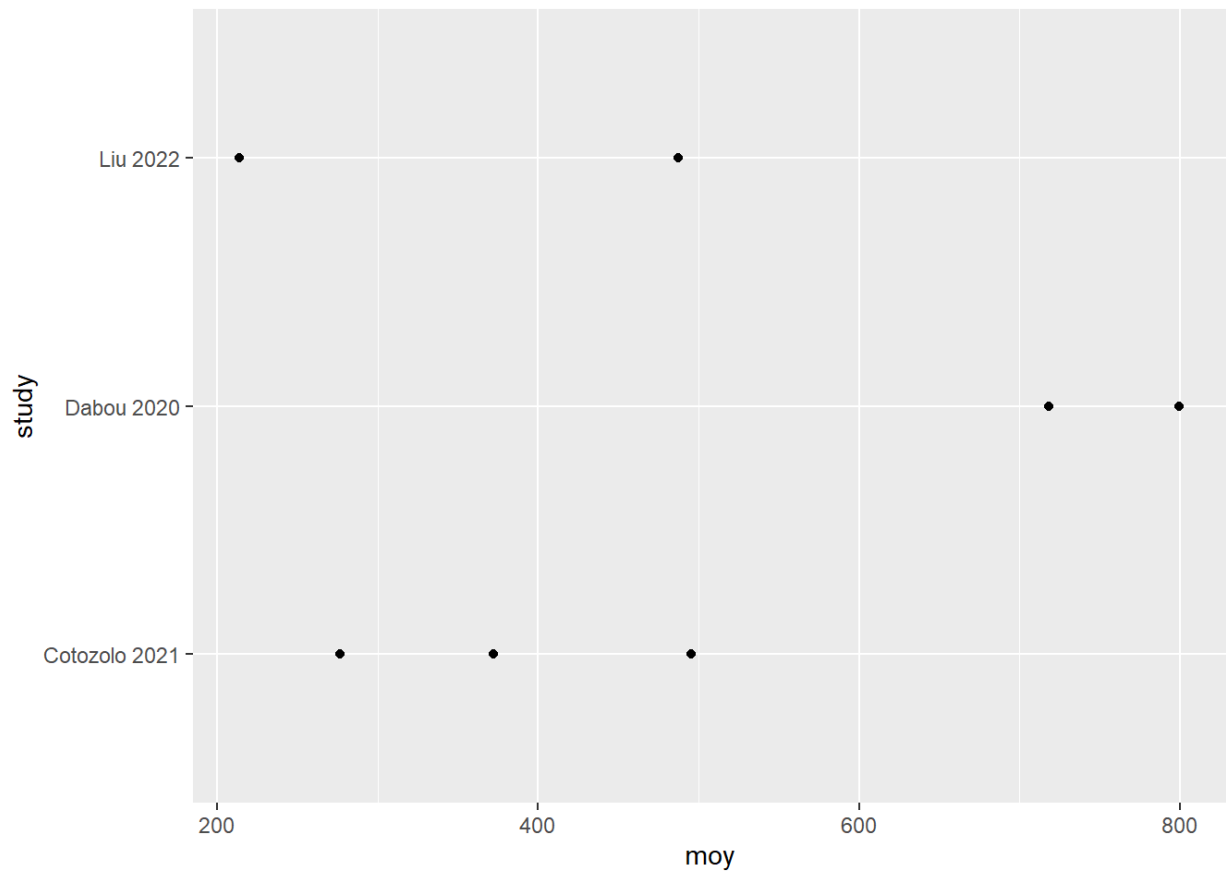


Ajout des moyennes au graph

On utilise la fonction `geom_point()` qui récupère les coordonnées définies par la fonction `aes()` dans la fonction `ggplot()`

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE)) |>
  ggplot(aes(x = study,
            y = moy)) +
  coord_flip() +
  geom_point()
```

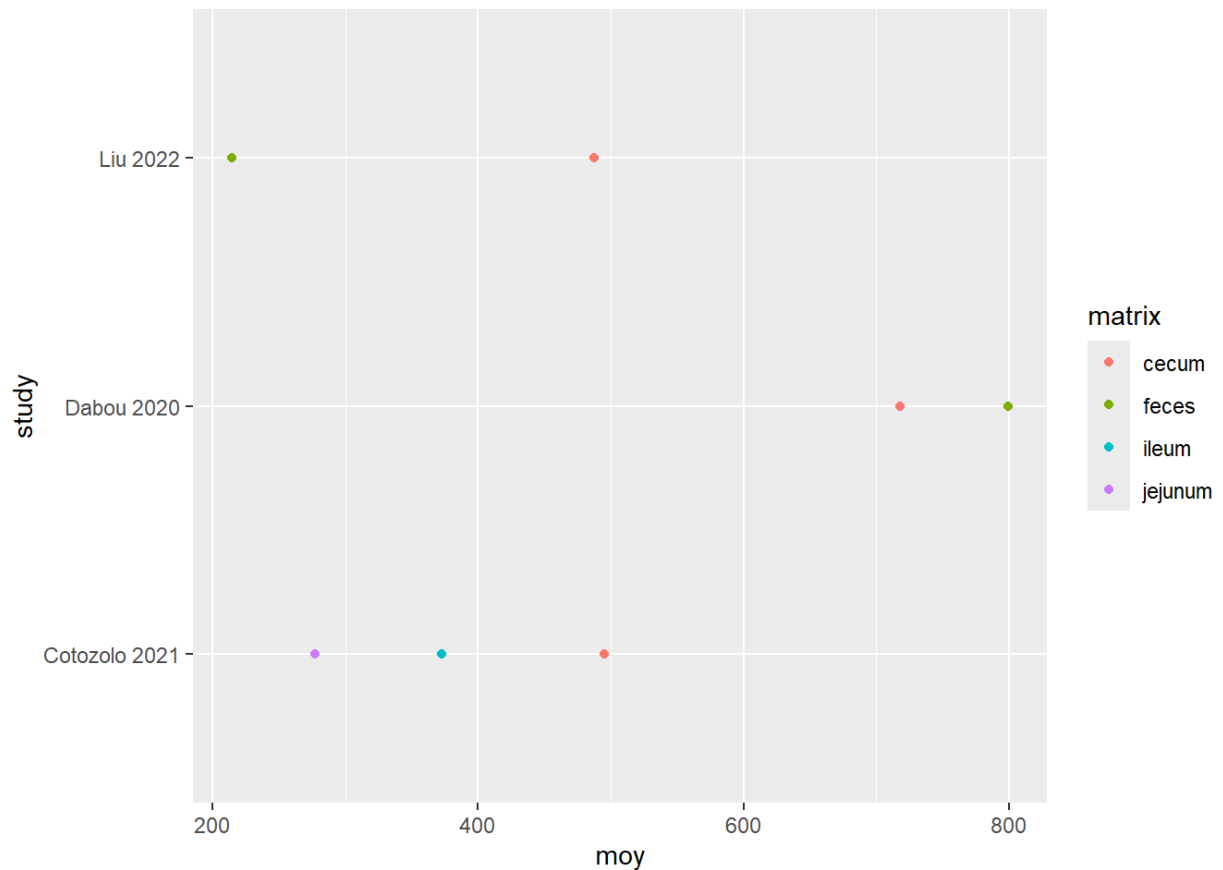
``summarise()`` has grouped output by 'study'. You can override using the ``.groups`` argument.



Ajout des couleurs en fonction de la matrice

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE)) |>
  ggplot(aes(x = study,
             y = moy,
             color = matrix)) +
  coord_flip() +
  geom_point()
```

`summarise()` has grouped output by 'study'. You can override using the `.groups` argument.

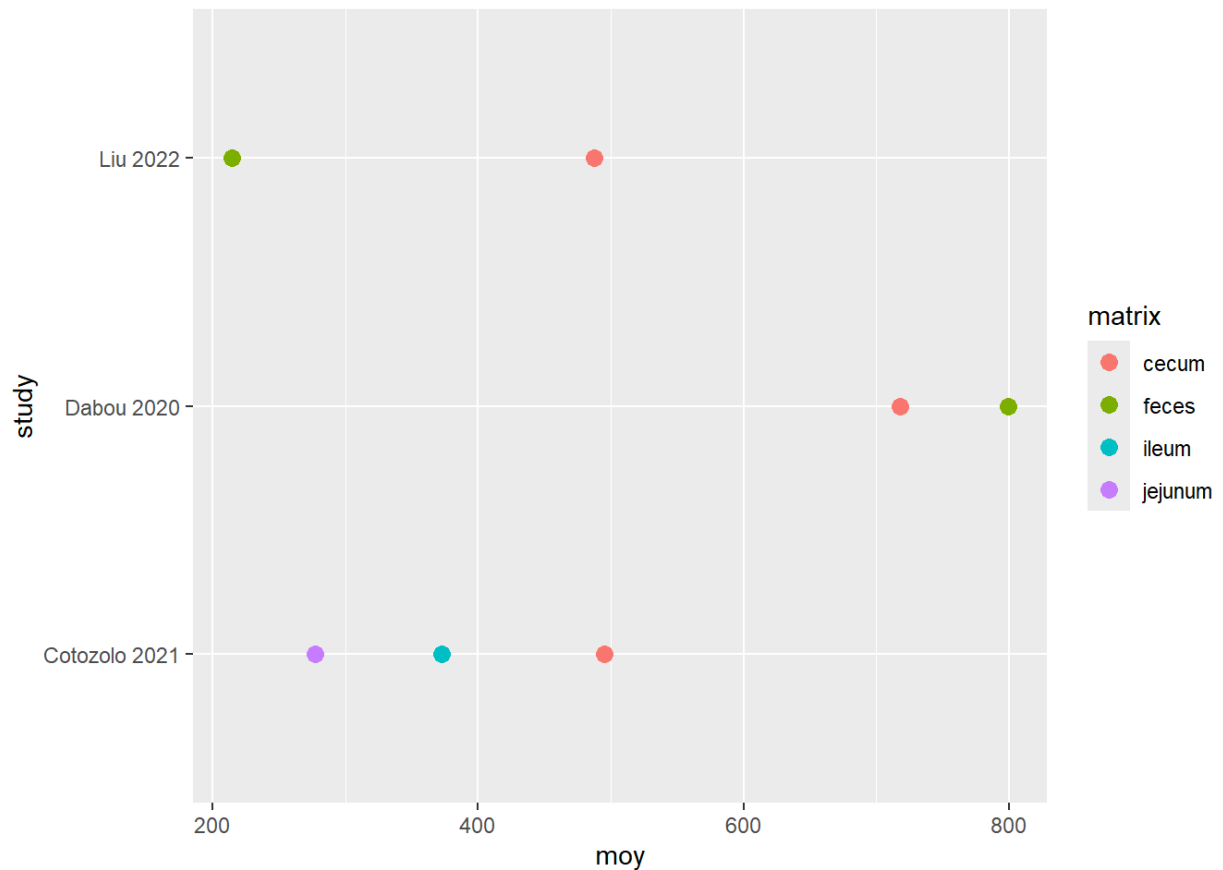


Augmente la taille des points

Comme on définit la taille pour tous les points sans rapports avec les données, le paramètre `size` n'est pas positionné dans l'`aes()`

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE)) |>
  ggplot(aes(x = study,
            y = moy,
            color = matrix)) +
  coord_flip() +
  geom_point(size = 3)
```

``summarise()`` has grouped output by 'study'. You can override using the ``.groups`` argument.



Affichage des intervalles de confiance avec des barres

On intègre les intervalles de confiance à notre jeu de données

Calcul des intervalles de confiance

On utilise la fonction `t.test()` dans le `summarise()`

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE),
            CI_min = t.test(Measure)$conf.int[1],
            CI_max = t.test(Measure)$conf.int[2])
```

``summarise()`` has grouped output by 'study'. You can override using the ``.groups`` argument.

```
# A tibble: 7 × 5
# Groups:   study [3]
  study      matrix    moy CI_min CI_max
<chr>      <chr>    <dbl> <dbl> <dbl>
1 Cotozolo 2021 cecum    495.  470.  521.
2 Cotozolo 2021 ileum    372.  316.  429.
3 Cotozolo 2021 jejunum  277.  249.  304.
```

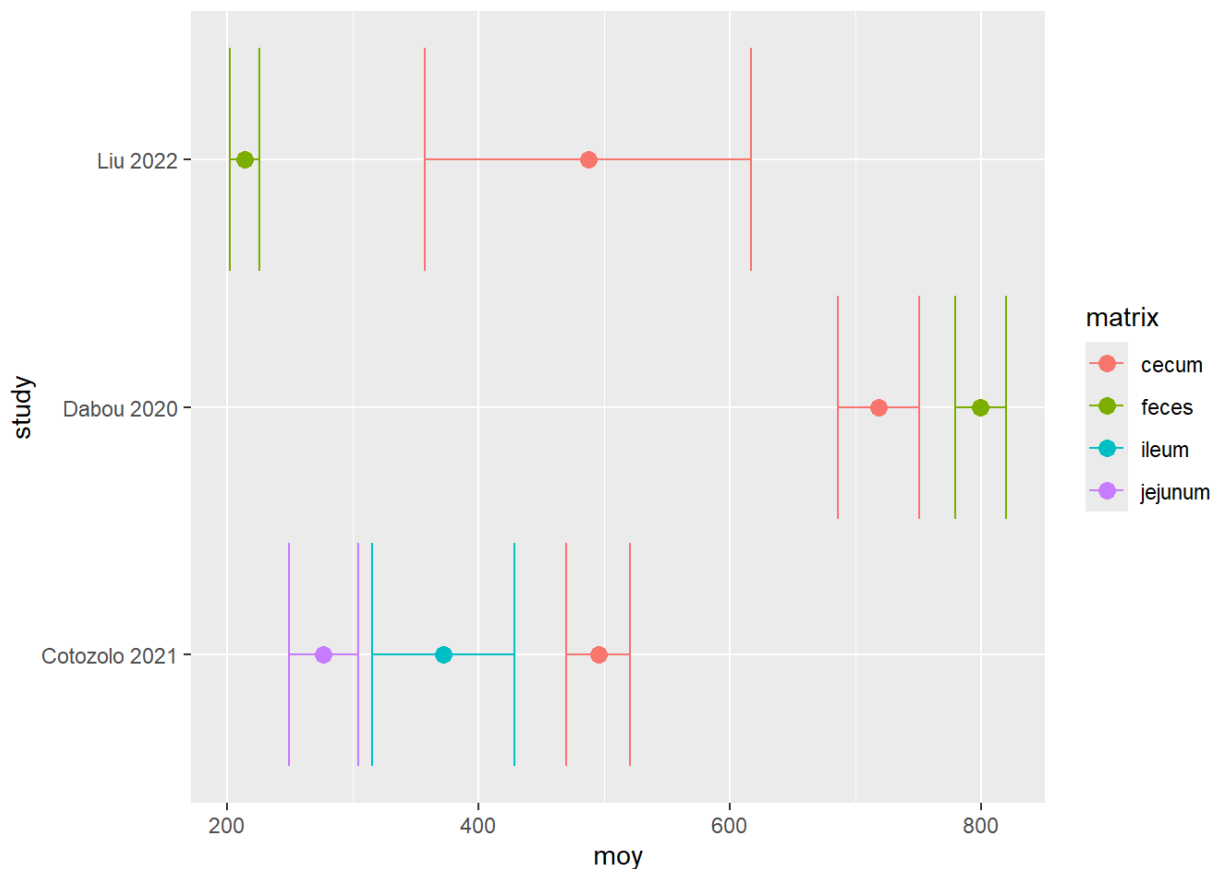
4	Dabou 2020	cecum	718.	686.	751.
5	Dabou 2020	feces	799.	779.	819.
6	Liu 2022	cecum	487.	358.	617.
7	Liu 2022	feces	214.	202.	226.

Ajout des intervalles de confiance au graph

On utilise la fonction `geom_errorbar()`

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE),
            CI_min = t.test(Measure)$conf.int[1],
            CI_max = t.test(Measure)$conf.int[2]) |>
  ggplot(aes(x = study,
            y = moy,
            color = matrix)) +
  coord_flip() +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = CI_min,
                    ymax = CI_max))
```

`summarise()` has grouped output by 'study'. You can override using the `.groups` argument.



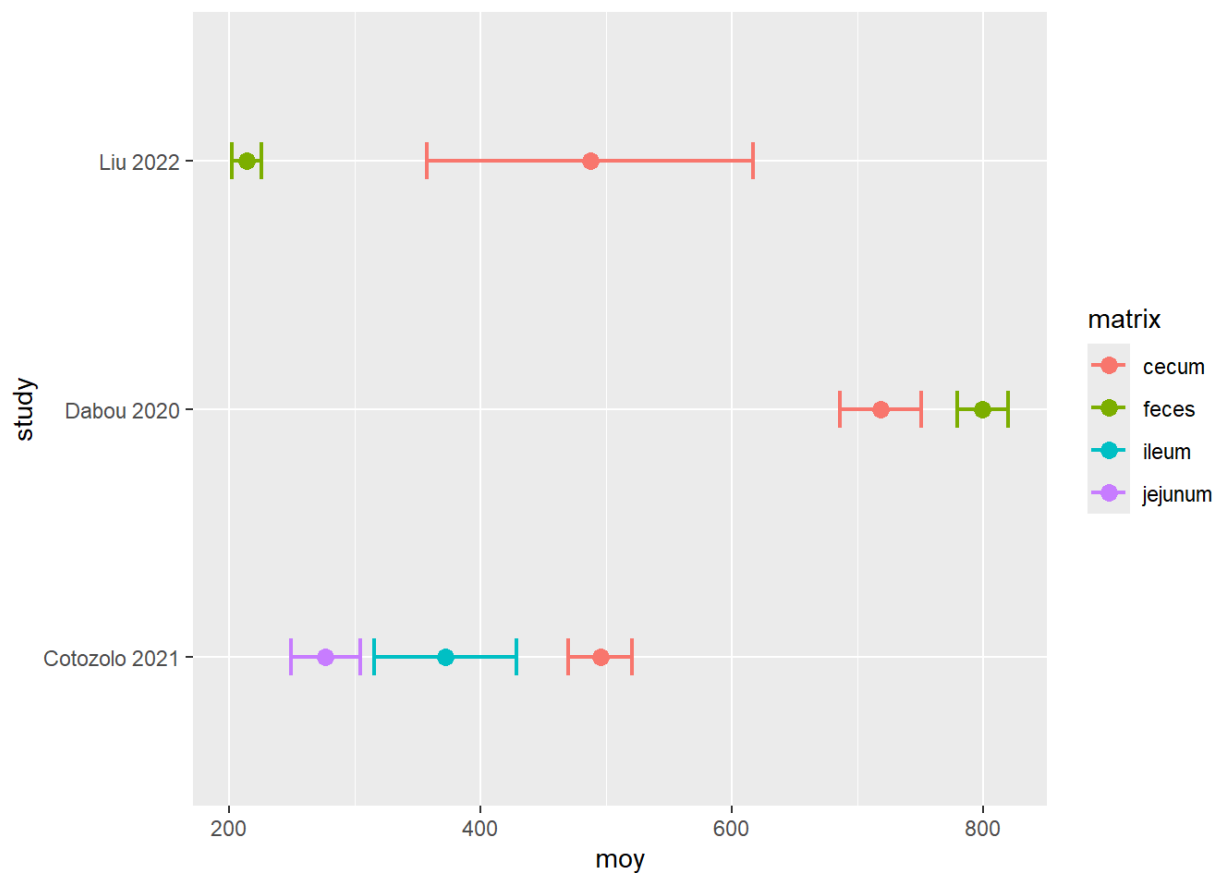
Customisation des barres d'erreurs

Paramètres de customisation:

- `linewidth`: épaisseur des traits ()
- `width`: longueur des barres aux extrémités

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE),
            CI_min = t.test(Measure)$conf.int[1],
            CI_max = t.test(Measure)$conf.int[2]) |>
  ggplot(aes(x = study,
            y = moy,
            color = matrix)) +
  coord_flip() +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = CI_min,
                  ymax = CI_max),
              width = 0.15,
              linewidth = .85)
```

``summarise()`` has grouped output by 'study'. You can override using the ``.groups`` argument.



Affichage du nombre total d'échantillon par matrice et par étude

Récupération du nombre d'échantillons par matrice et par étude

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE),
            CI_min = t.test(Measure)$conf.int[1],
            CI_max = t.test(Measure)$conf.int[2],
            nb = n())
```

`summarise()` has grouped output by 'study'. You can override using the `.groups` argument.

```
# A tibble: 7 × 6
# Groups:   study [3]
  study      matrix    moy CI_min CI_max  nb
<chr>      <chr>  <dbl> <dbl> <dbl> <int>
1 Cotozolo 2021 cecum   495.  470.  521.  14
2 Cotozolo 2021 ileum   372.  316.  429.  14
3 Cotozolo 2021 jejunum 277.  249.  304.  12
4 Dabou 2020 cecum   718.  686.  751.  24
5 Dabou 2020 feces   799.  779.  819.  58
6 Liu 2022 cecum   487.  358.  617.  12
7 Liu 2022 feces   214.  202.  226.  70
```

Affichage du nombre d'échantillons par matrice et par étude

Plusieurs fonctions disponibles:

- `geom_text()`
- `geom_label()`
- `geom_text_repel()`
- `geom_label_repel()`

Caractéristiques de ces fonctions:

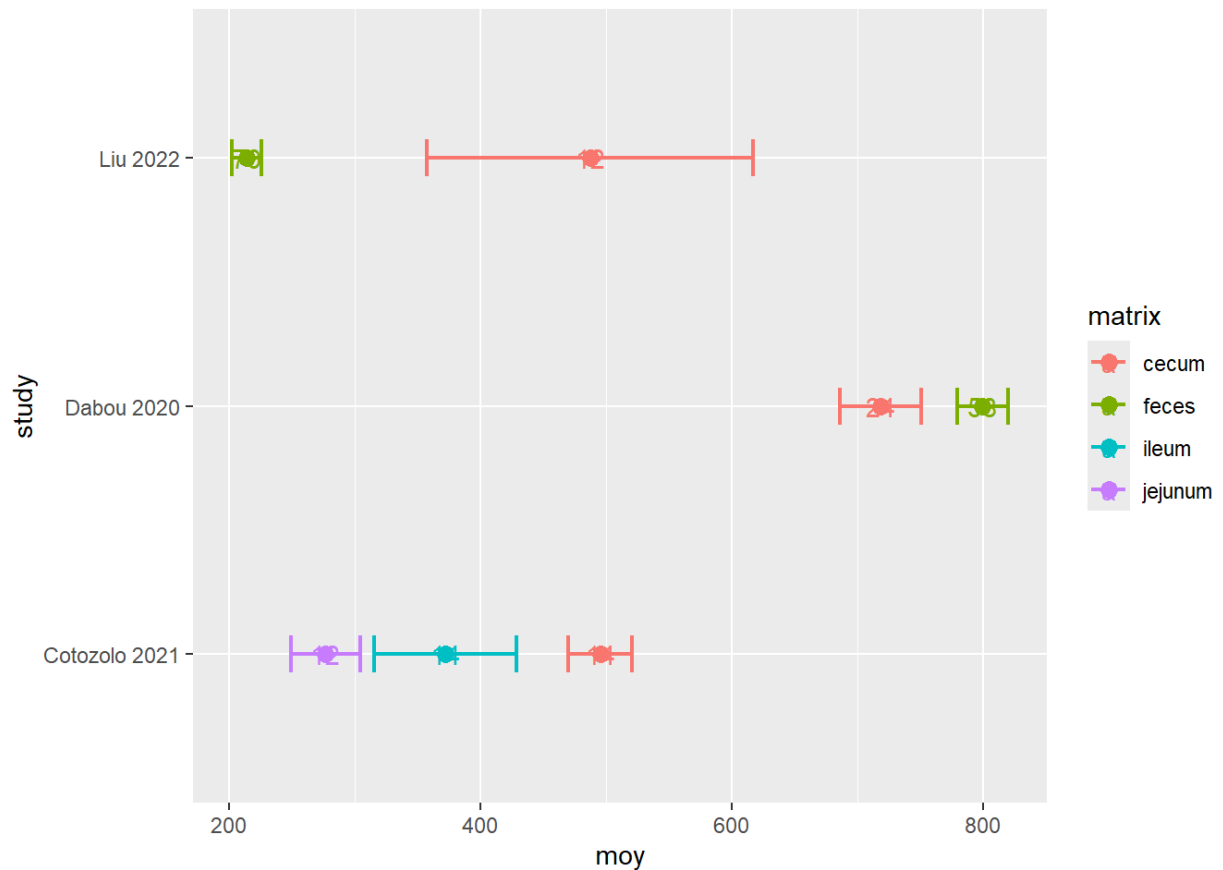
- Les fonctions `geom_text()` et `geom_label()` sont incluses dans le package `ggplot2`, les fonctions `geom_text_repel()` et `geom_label_repel()` proviennent du package `ggrepel`
- Les 2 fonctions `geom_text()` et `geom_text_repel()` affiche juste un texte alors que les fonctions `geom_label()` et `geom_label_repel()` affiche un texte à l'intérieur d'un boîte configurable
- Les 2 fonctions `geom_text_repel()` et `geom_label_repel()` issues du package `ggrepel` positionnent automatiquement les textes et boîtes de façon à ce qu'elles ne se superposent pas.

Ces fonctions reprennent les paramètres de l'`aes()` déjà définis au-dessus, il faut juste rajouter la liaison avec la variable `nb: label = nb`

Exemple avec `geom_text()`

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE),
            CI_min = t.test(Measure)$conf.int[1],
            CI_max = t.test(Measure)$conf.int[2],
            nb = n()) |>
  ggplot(aes(x = study,
            y = moy,
            color = matrix,
            label = nb)) +
  coord_flip() +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = CI_min,
                  ymax = CI_max),
              width = 0.15,
              linewidth = .85) +
  geom_text()
```

``summarise()`` has grouped output by 'study'. You can override using the ``.groups`` argument.



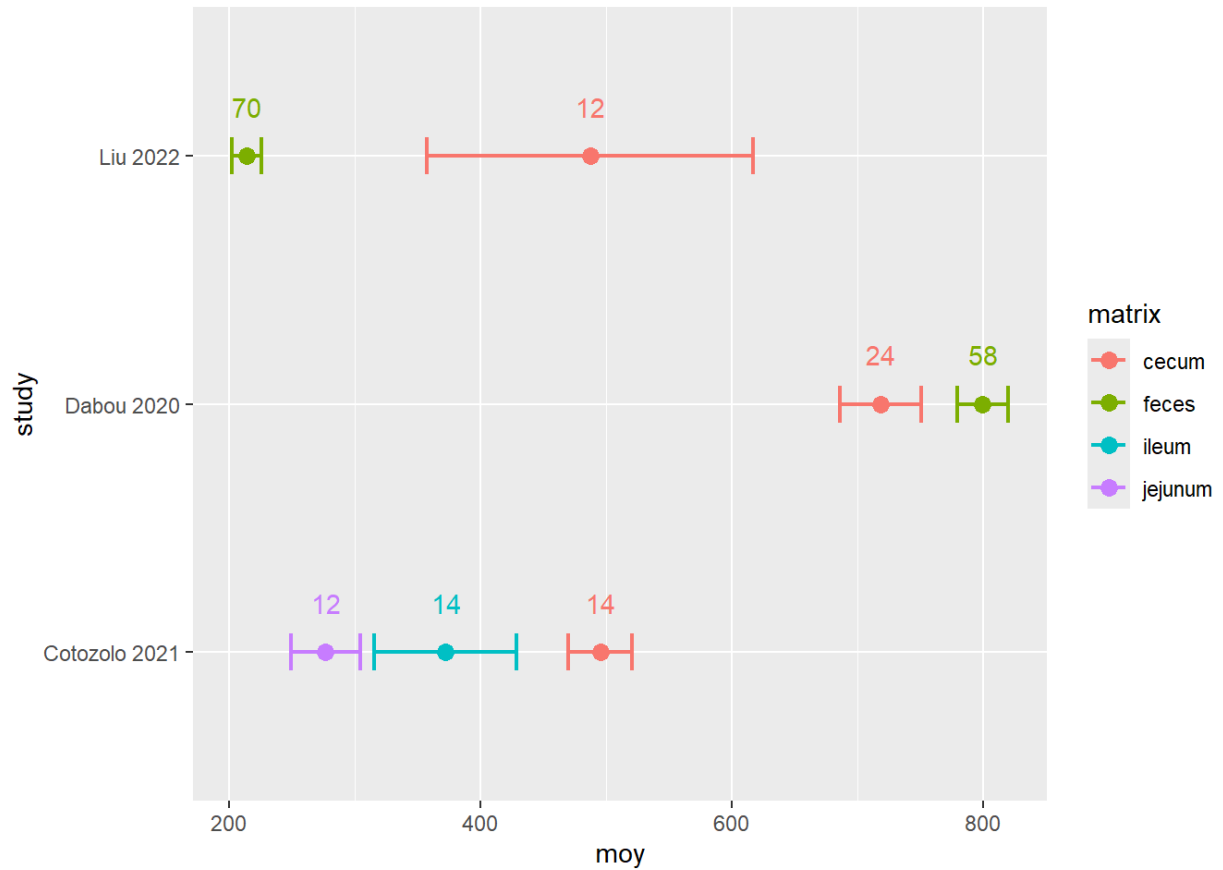
Il est possible de positionner les labels avec les paramètres `nudge_x` et `nudge_y` de la fonction

On rajoute le paramètre `show.legend = FALSE` pour enlever le caractère “a” qui se rajoute dans la légende

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE),
            CI_min = t.test(Measure)$conf.int[1],
            CI_max = t.test(Measure)$conf.int[2],
            nb = n()) |>
  group_by(study) |>
  mutate(tot = sum(nb)) |>
  ggplot(aes(x = study,
            y = moy,
            color = matrix,
            label = nb)) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = CI_min,
                  ymax = CI_max),
              width = 0.15,
              linewidth = .85) +
  coord_flip() +
  geom_text(nudge_x = .2,
```

```
nudge_y = .2,
show.legend = FALSE)
```

``summarise()`` has grouped output by 'study'. You can override using the ``.groups`` argument.

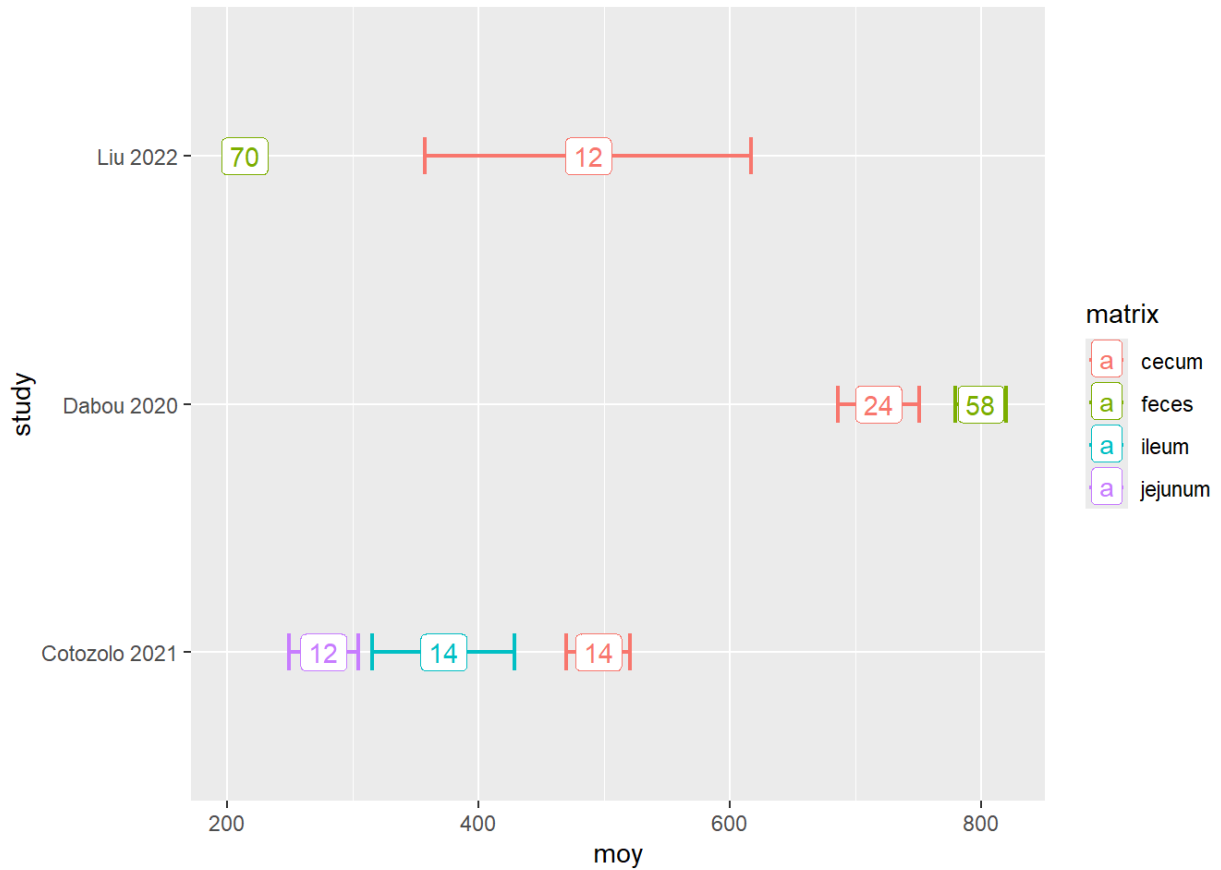


Exemple avec `geom_label()`

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE),
            CI_min = t.test(Measure)$conf.int[1],
            CI_max = t.test(Measure)$conf.int[2],
            nb = n()) |>
  ggplot(aes(x = study,
            y = moy,
            color = matrix,
            label = nb)) +
  coord_flip() +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = CI_min,
                  ymax = CI_max),
              width = 0.15,
```

```
linewidth = .85) +  
geom_label()
```

``summarise()`` has grouped output by 'study'. You can override using the ``.groups`` argument.



`geom_label()` utilise les mêmes paramètres pour repositionner les textes et boîtes en même temps
Il faut aussi rajouter `show.legend = FALSE`

```
alpha_div |>  
  group_by(study, matrix) |>  
  summarise(moy = mean(Measure, na.rm = TRUE),  
            CI_min = t.test(Measure)$conf.int[1],  
            CI_max = t.test(Measure)$conf.int[2],  
            nb = n()) |>  
  group_by(study) |>  
  mutate(tot = sum(nb)) |>  
  ggplot(aes(x = study,  
            y = moy,  
            color = matrix,  
            label = nb)) +  
  coord_flip() +  
  geom_point(size = 3) +  
  geom_errorbar(aes(ymin = CI_min,
```

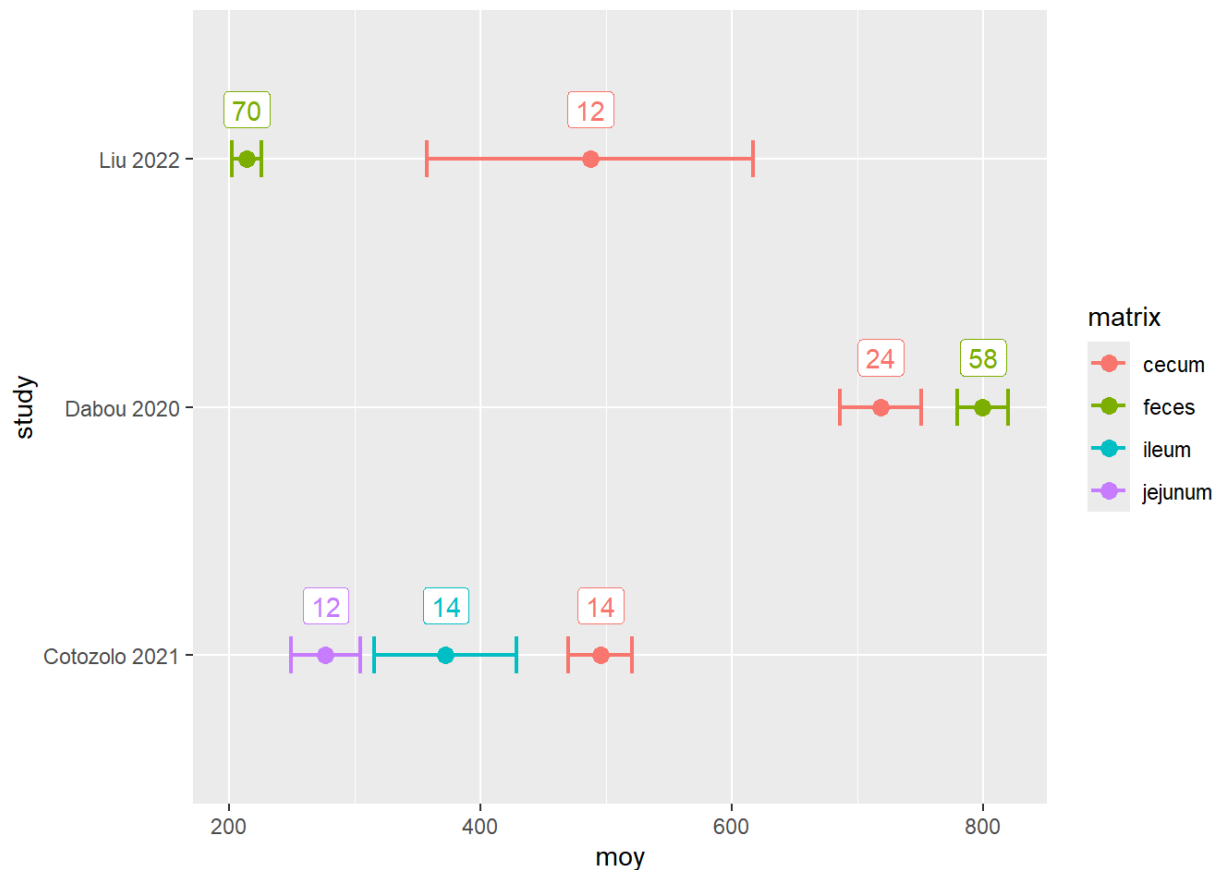


```

      ymax = CI_max),
      width = 0.15,
      linewidth = .85) +
geom_label(nudge_x = .2,
          nudge_y = .2,
          show.legend = FALSE)

```

`summarise()` has grouped output by 'study'. You can override using the `.groups` argument.



On peut également modifier la taille et l'apparence des boîtes avec les paramètres suivants:

- `label.padding = unit(0.25, "lines")`: taille de la boîte
- `label.r = unit(0.15, "lines")`: courbure des coins
- `label.size = 0.25`: épaisseur du trait

```

alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE),
            CI_min = t.test(Measure)$conf.int[1],
            CI_max = t.test(Measure)$conf.int[2],
            nb = n()) |>
  group_by(study) |>
  mutate(tot = sum(nb)) |>

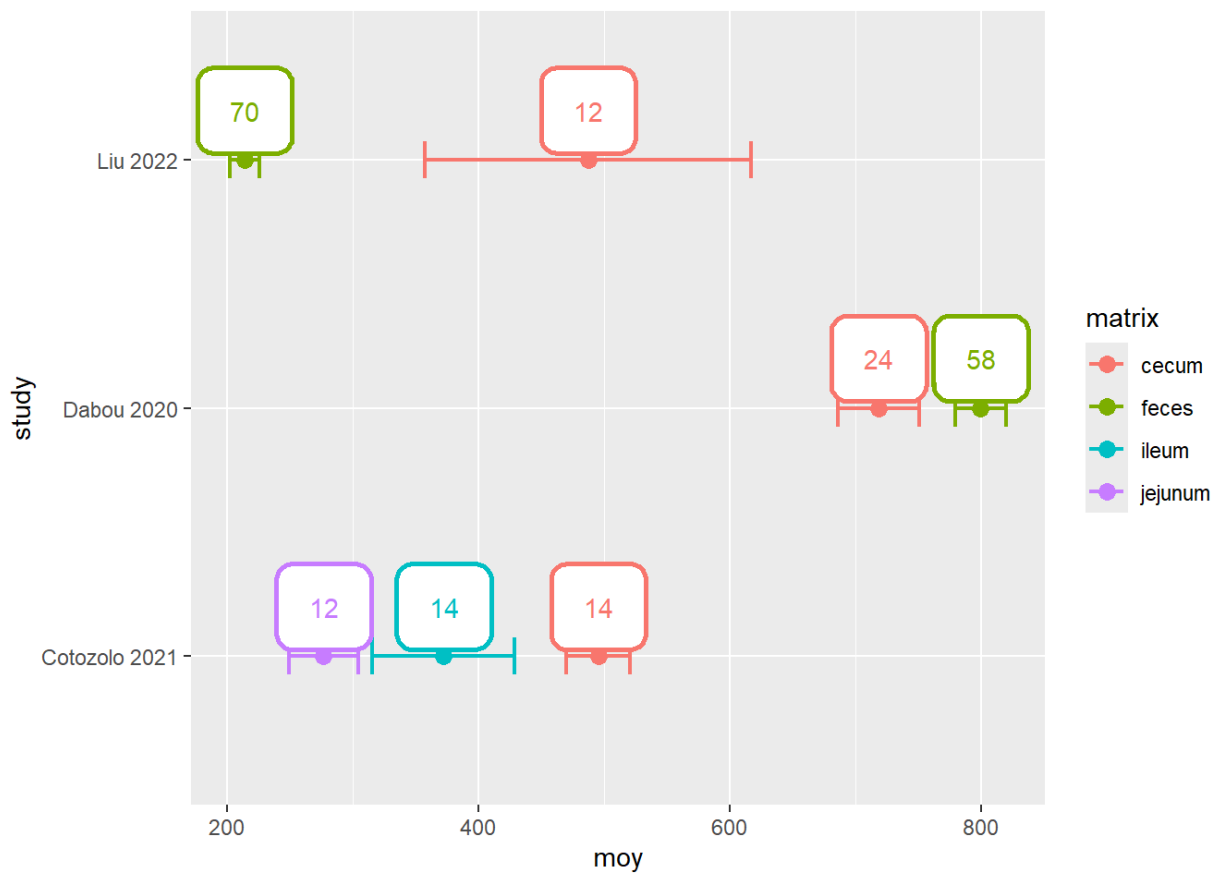
```

```

ggplot(aes(x = study,
           y = moy,
           color = matrix,
           label = nb)) +
  coord_flip() +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = CI_min,
                  ymax = CI_max),
              width = 0.15,
              linewidth = .85) +
  geom_label(nudge_x = .2,
            nudge_y = .2,
            label.padding = unit(1, "lines"),
            label.r = unit(0.5, "lines"),
            label.size = 1,
            show.legend = FALSE)

```

`summarise()` has grouped output by 'study'. You can override using the `.groups` argument.



Package ggrepel

Le package [ggrepel](#) est une extension du package `ggplot2` qui permet d'éviter la superposition des textes et des étiquettes dans les graphiques.

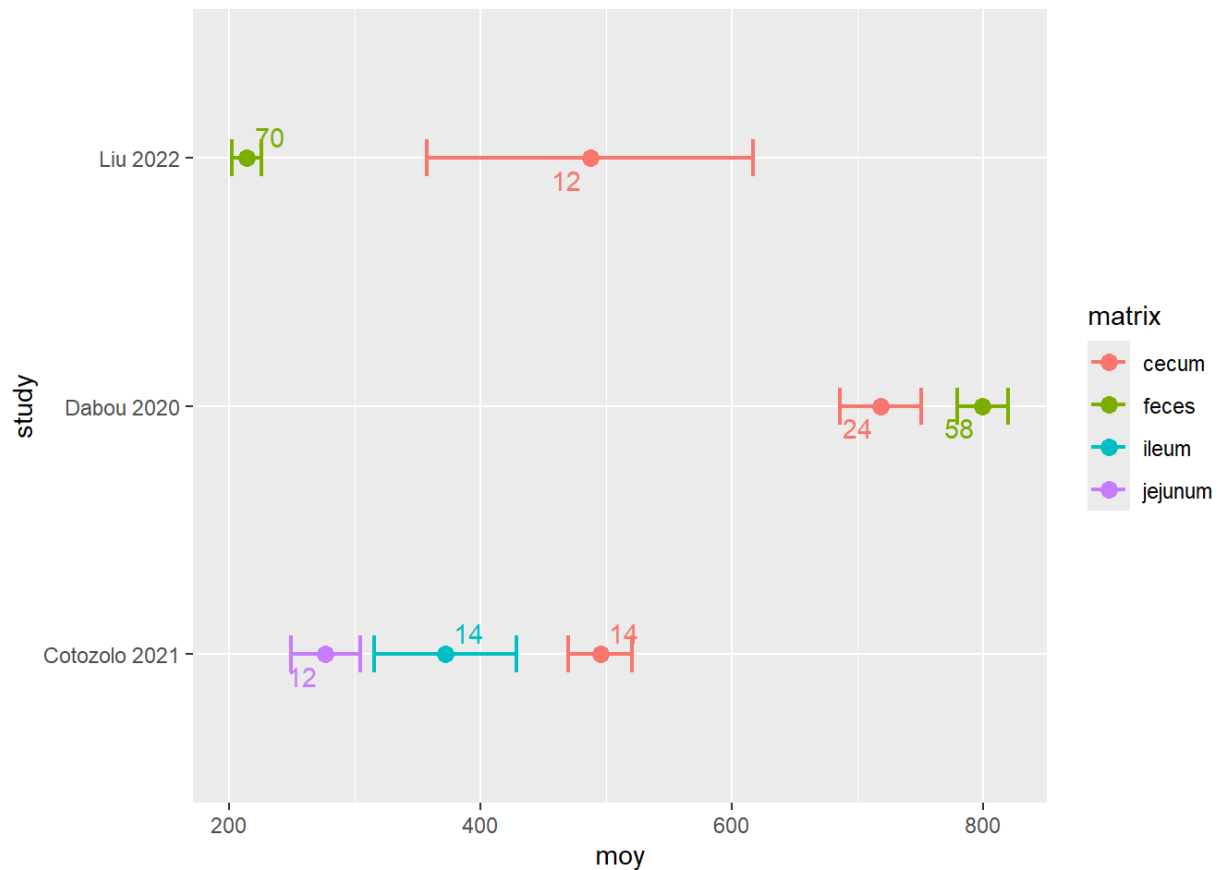
Il contient les fonctions `geom_text_repel()` et `geom_label_repel()` qui sont les équivalents des fonctions `geom_text()` et `geom_label()` de `ggplot2`

ATTENTION: On ne plus pas utiliser la fonction `coord_flip()` avec ces fonctions du package
Le bug est apparu lors de la dernière mise à jour

Etant donné que l'on ne peut pas utiliser la fonction `coord_flip()`, il suffit d'inverser le coordonnées des axes dans la fonction `aes()` et de changer les paramètres `aes` dans la fonction `geom_errorbar()` !

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE),
            CI_min = t.test(Measure)$conf.int[1],
            CI_max = t.test(Measure)$conf.int[2],
            nb = n()) |>
  group_by(study) |>
  mutate(tot = sum(nb)) |>
  ggplot(aes(x = moy,
            y = study,
            color = matrix,
            label = nb)) +
  geom_point(size = 3) +
  geom_errorbar(aes(xmin = CI_min,
                  xmax = CI_max),
               width = 0.15,
               linewidth = .85) +
  geom_text_repel(show.legend = FALSE)
```

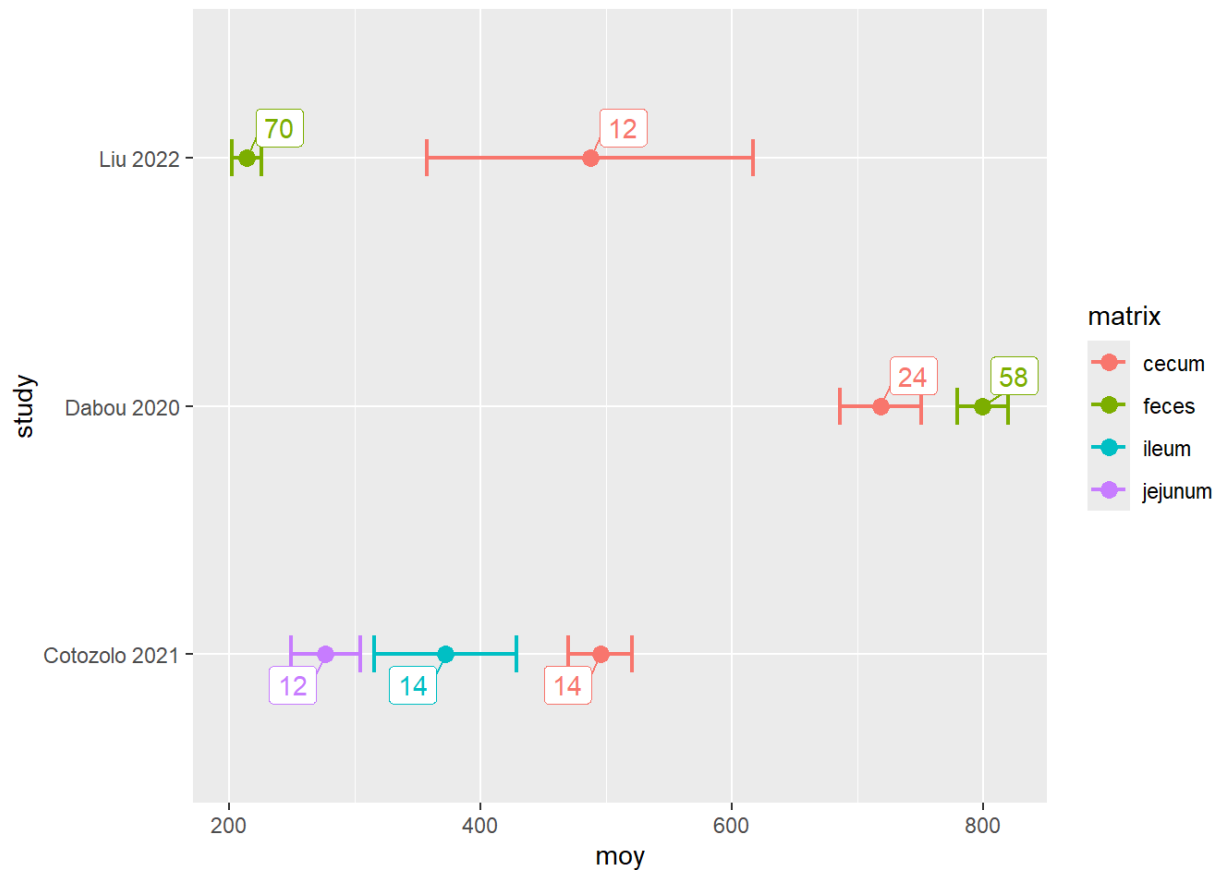
```
`summarise()` has grouped output by 'study'. You can override using the
`.groups` argument.
```



Même chose pour `geom_label_repel()`

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE),
            CI_min = t.test(Measure)$conf.int[1],
            CI_max = t.test(Measure)$conf.int[2],
            nb = n()) |>
  group_by(study) |>
  mutate(tot = sum(nb)) |>
  ggplot(aes(x = moy,
            y = study,
            color = matrix,
            label = nb)) +
    geom_point(size = 3) +
    geom_errorbar(aes(xmin = CI_min,
                    xmax = CI_max),
                width = 0.15,
                linewidth = .85) +
    geom_label_repel(show.legend = FALSE)
```

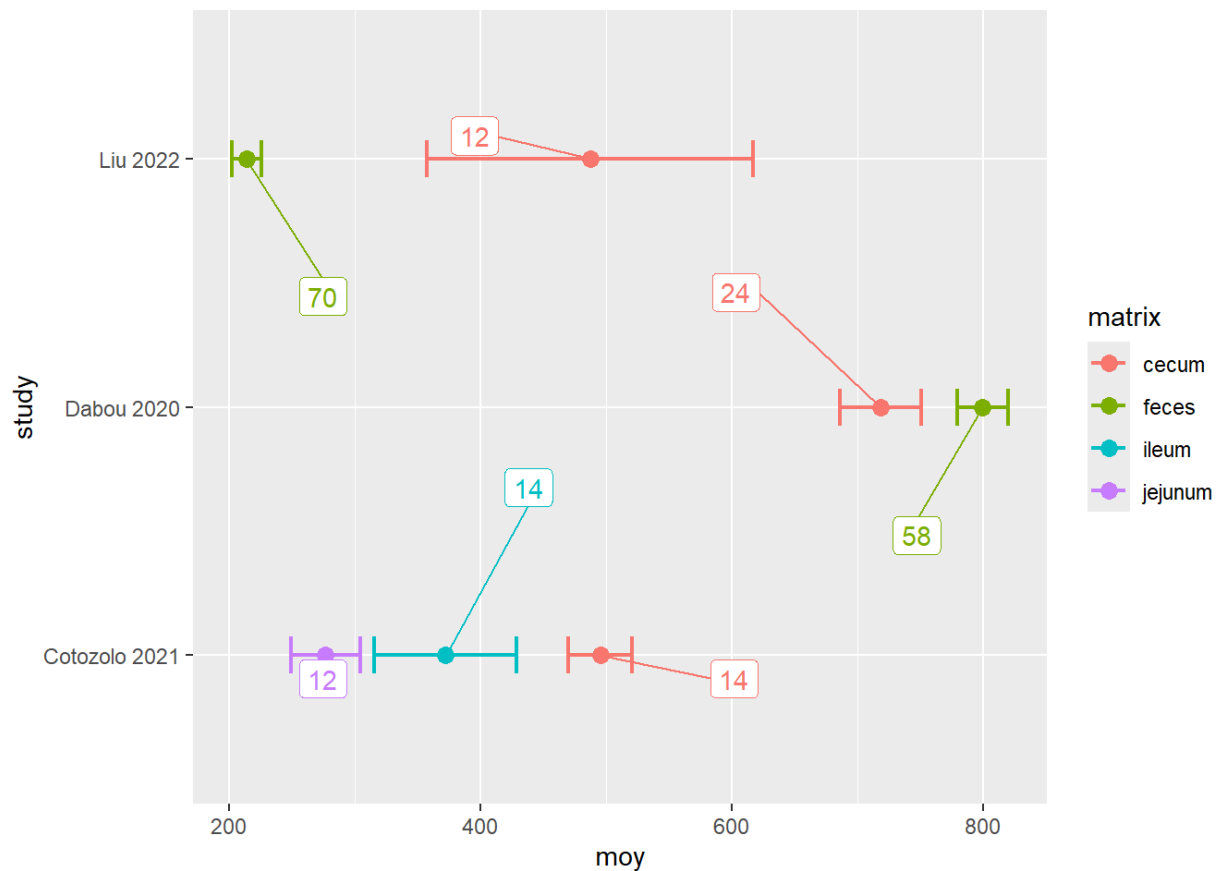
``summarise()`` has grouped output by 'study'. You can override using the ``.groups`` argument.



Des paramètres supplémentaires permettent de contrôler les textes et les boîtes ainsi que les espacements

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE),
            CI_min = t.test(Measure)$conf.int[1],
            CI_max = t.test(Measure)$conf.int[2],
            nb = n()) |>
  group_by(study) |>
  mutate(tot = sum(nb)) |>
  ggplot(aes(x = moy,
            y = study,
            color = matrix,
            label = nb)) +
  geom_point(size = 3) +
  geom_errorbar(aes(xmin = CI_min,
                  xmax = CI_max),
               width = 0.15,
               linewidth = .85) +
  geom_label_repel(box.padding = 3,
                  show.legend = FALSE)
```

``summarise()`` has grouped output by 'study'. You can override using the ``.groups`` argument.



Affichage du nombre total d'échantillon par étude

Récupération du nombre d'échantillons par étude

Cette fois-ci on groupe par étude et on utilise la fonction `mutate()` pour garder notre tableau de départ.

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE),
            CI_min = t.test(Measure)$conf.int[1],
            CI_max = t.test(Measure)$conf.int[2],
            nb = n()) |>
  group_by(study) |>
  mutate(som = sum(nb))
```

``summarise()`` has grouped output by 'study'. You can override using the ``.groups`` argument.

```
# A tibble: 7 × 7
# Groups:   study [3]
  study      matrix      moy CI_min CI_max  nb  som
  <fct> <fct> <dbl> <dbl> <dbl> <dbl> <dbl>
```

	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<int>	<int>
1	Cotozolo	2021	cecum	495.	470.	521.	14 40
2	Cotozolo	2021	ileum	372.	316.	429.	14 40
3	Cotozolo	2021	jejunum	277.	249.	304.	12 40
4	Dabou	2020	cecum	718.	686.	751.	24 82
5	Dabou	2020	feces	799.	779.	819.	58 82
6	Liu	2022	cecum	487.	358.	617.	12 82
7	Liu	2022	feces	214.	202.	226.	70 82

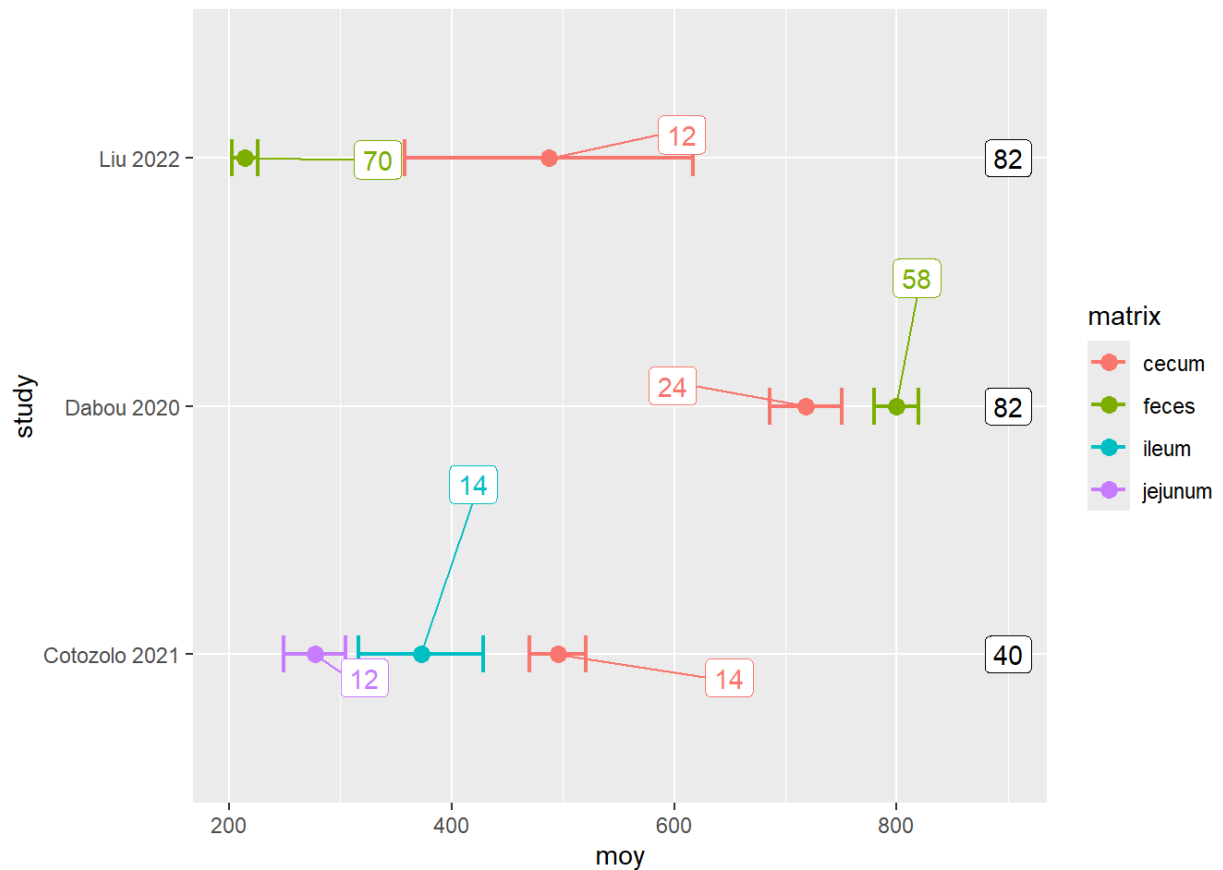
Affichage du nombre d'échantillons par étude

On utilise la fonction `geom_label()` pour afficher le nombre total d'échantillons par étude

On fixe manuellement les coordonnées en x à 900

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE),
            CI_min = t.test(Measure)$conf.int[1],
            CI_max = t.test(Measure)$conf.int[2],
            nb = n()) |>
  group_by(study) |>
  mutate(tot = sum(nb)) |>
  ggplot(aes(x = moy,
            y = study,
            color = matrix,
            label = nb)) +
  geom_point(size = 3) +
  geom_errorbar(aes(xmin = CI_min,
                  xmax = CI_max),
               width = 0.15,
               linewidth = .85) +
  geom_label_repel(box.padding = 3,
                  show.legend = FALSE) +
  geom_label(aes(x = 900,
                y = study,
                label = tot),
            color = "black",
            show.legend = FALSE)
```

`summarise()` has grouped output by 'study'. You can override using the `.groups` argument.



Affichage du text “Total” sur le graphique

Le texte “Total” est positionné manuellement, on utilise la fonction `annotate()` qui est une couche de type `geom` mais qui n'utilise pas `d'aes()`

On déclare le type d'annotation (`text`, `segment`, `line`...) puis les coordonnées et les paramètres classiques que l'on retrouve dans les autres `geom` (`color`, `size`...)

```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE),
            CI_min = t.test(Measure)$conf.int[1],
            CI_max = t.test(Measure)$conf.int[2],
            nb = n()) |>
  group_by(study) |>
  mutate(tot = sum(nb)) |>
  ggplot(aes(x = moy,
            y = study,
            color = matrix,
            label = nb)) +
    geom_point(size = 3) +
    geom_errorbar(aes(xmin = CI_min,
                    xmax = CI_max),
                width = 0.15,
                linewidth = .85) +
```

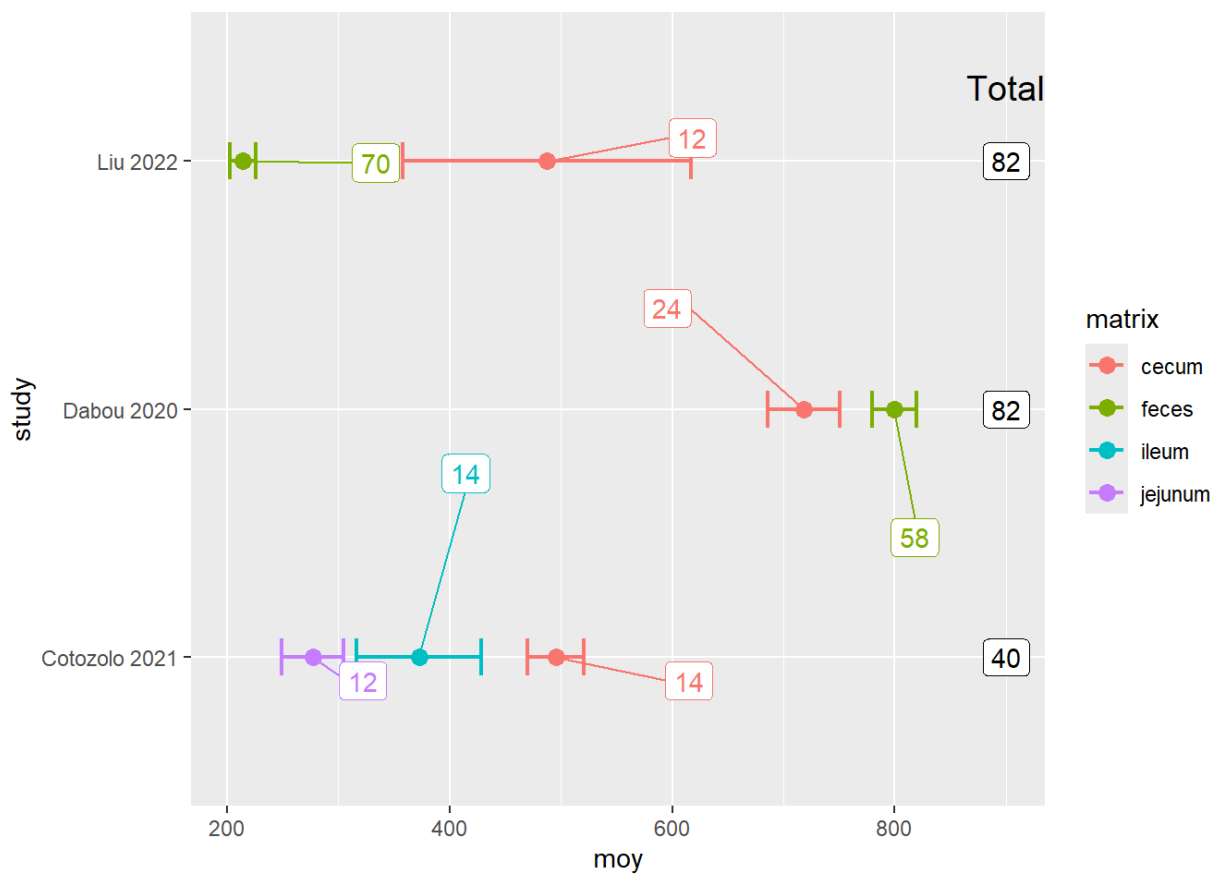


```

geom_label_repel(box.padding = 3,
                 show.legend = FALSE) +
geom_label(aes(x = 900,
               y = study,
               label = tot),
           color = "black",
           show.legend = FALSE) +
annotate(geom = "text",
         x = 900,
         y = 3.3,
         label = "Total",
         size = 5)

```

`summarise()` has grouped output by 'study'. You can override using the `.groups` argument.



Changement des couleurs

Palette

On utilise la palette `cols25` du package `pals` qui fournit une palette de 25 couleurs contrastées

```
pal.bands(cols25)
```

Only 25 colors are available with 'cols25'.



On choisit les 4 couleurs que l'on souhaite dans la fonction `scale_color_manual()` et on en profite pour modifier le titre de la légende

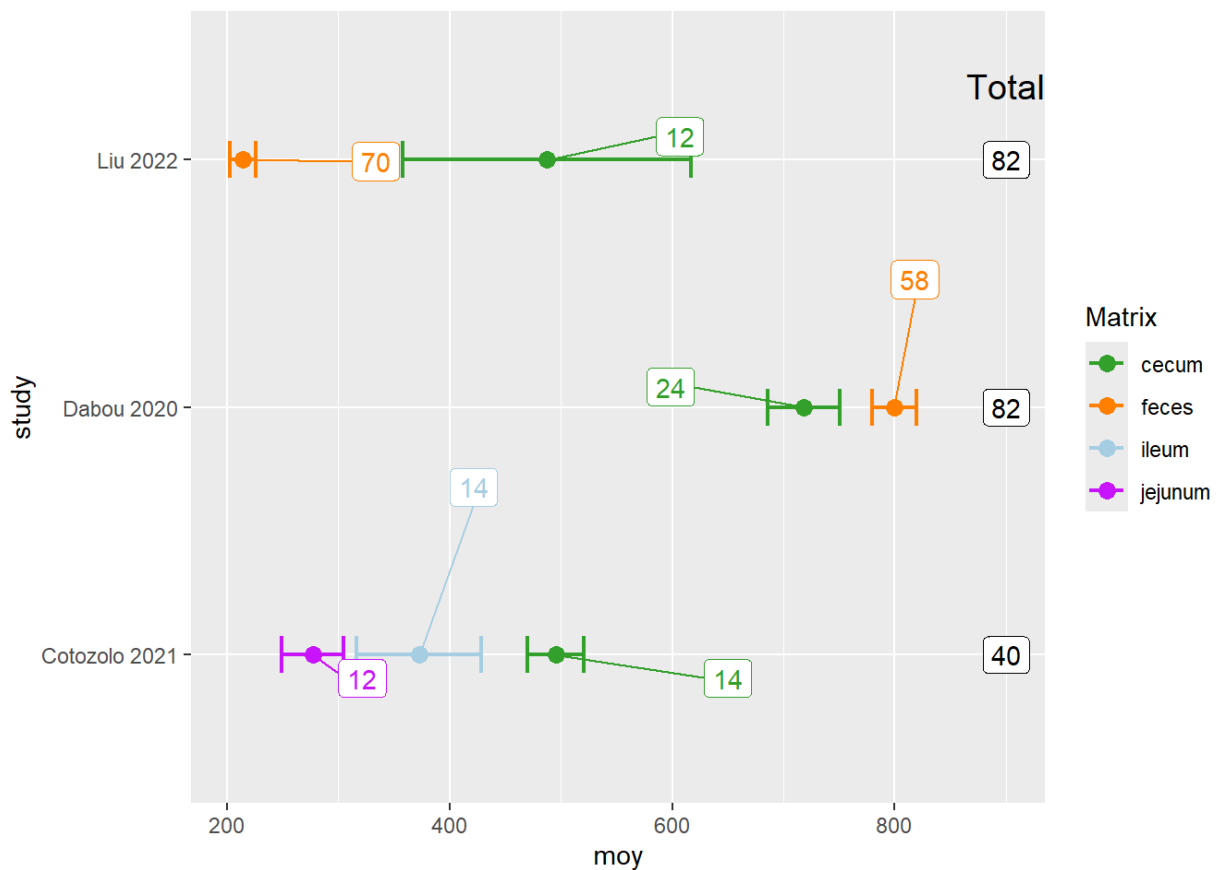
```
alpha_div |>
  group_by(study, matrix) |>
  summarise(moy = mean(Measure, na.rm = TRUE),
            CI_min = t.test(Measure)$conf.int[1],
            CI_max = t.test(Measure)$conf.int[2],
            nb = n()) |>
  group_by(study) |>
  mutate(tot = sum(nb)) |>
  ggplot(aes(x = moy,
            y = study,
            color = matrix,
            label = nb)) +
  geom_point(size = 3) +
  geom_errorbar(aes(xmin = CI_min,
                  xmax = CI_max),
              width = 0.15,
              linewidth = .85) +
  geom_label_repel(box.padding = 3,
                  show.legend = FALSE) +
```

```

geom_label(aes(x = 900,
              y = study,
              label = tot),
          color = "black",
          show.legend = FALSE) +
annotate(geom = "text",
        x = 900,
        y = 3.3,
        label = "Total",
        size = 5) +
scale_color_manual("Matrix", values = cols25()[c(3,5,8, 17)])

```

`summarise()` has grouped output by 'study'. You can override using the `.groups` argument.



Customisation du graphique

Enfin, customisation du graphique avec les fonctions:

- `labs()` : pour les titres des axes
- `theme()` :
 - pour les couleurs des axes
 - l'aspect de la grille
 - couleur de l'arrière-plan

```

## couleur des axes
axis.line = element_line(color = "grey50")
## enlève les petites lignes de la grille
panel.grid.minor = element_blank()
## change la couleur et l'aspect (pointillé) des grandes lignes de la
grille
panel.grid.major.y = element_line(color = "grey80",
                                   linetype = "dashed")
## Supprime la couleur de l'arrière-plan
panel.background = element_blank()

```

Graphique final

```

alpha_div |>
  group_by(study, matrix) |>
  reframe(moy = mean(Measure, na.rm = TRUE),
          CI_min = t.test(Measure)$conf.int[1],
          CI_max = t.test(Measure)$conf.int[2],
          nb = n()) |>
  group_by(study) |>
  mutate(tot = sum(nb)) |>
  ggplot(aes(x = study,
            y = moy,
            color = matrix)) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = CI_min,
                  ymax = CI_max),
              width = 0.15,
              linewidth = .85) +
  coord_flip() +
  geom_label(aes(label = nb), nudge_x = .2, nudge_y = .2,
            show.legend = FALSE) +
  geom_label(aes(x = study,
                y = 900,
                label = tot),
            color = "black",
            show.legend = FALSE) +
  annotate(geom = "text",
         x = 3.3,
         y = 900,
         label = "Total", size = 5) +
  scale_color_manual("Matrix", values = cols25()[c(3,5,8, 17)]) +
  labs(x = "Etudes",
       y = "Observed") +

```

```
theme(axis.line = element_line(color = "grey50"),
      panel.grid.minor = element_blank(),
      panel.grid.major.y = element_line(color = "grey80",
                                         linetype = "dashed"),
      panel.background = element_blank())
```

