02 - Visualisation de données

PRO1036 - Analyse de données scientifiques en R

Tim Bollé

September 15, 2025

Qu’est ce qu’un dataset ?

Terminologie

En français, on parle de jeu de données. Concrètement, c'est un tableau:

- Chaque ligne correspond à une `observation`
- Chaque colonne correspond à une `variable`

```
1 starwars
```

```
# A tibble: 87 × 14
   name      height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>      <int> <dbl> <chr>      <chr>      <chr>           <dbl> <chr> <chr>
 1 Luke Sk…     172    77 blond      fair       blue             19   male  mascu…
 2 C-3PO        167    75 <NA>       gold       yellow          112   none  mascu…
 3 R2-D2         96    32 <NA>       white, bl… red              33   none  mascu…
 4 Darth V…     202   136 none       white      yellow           41.9 male  mascu…
 5 Leia Or…     150    49 brown      light      brown            19   fema… femin…
 6 Owen La…     178   120 brown, gr… light      blue             52   male  mascu…
 7 Beru Wh…     165    75 brown      light      blue             47   fema… femin…
 8 R5-D4         97    32 <NA>       white, red red              NA   none  mascu…
 9 Biggs D…     183    84 black      light      brown            24   male  mascu…
10 Obi-Wan…     182    77 auburn, w… fair       blue-gray        57   male  mascu…
# ℹ 77 more rows
# ℹ 5 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>
```

Obiwan Kenobi

- Taille = 182cm
- Poids = 77kg
- Couleur des cheuveux = brun
- Couleur des yeux = gris-bleu
- Année de naissance = 57BBY
- Planète d’origine = Stewjon …

Jetons un coup d'oeil

Nous pouvons avoir une vue d'ensemble avec `glimpse`:

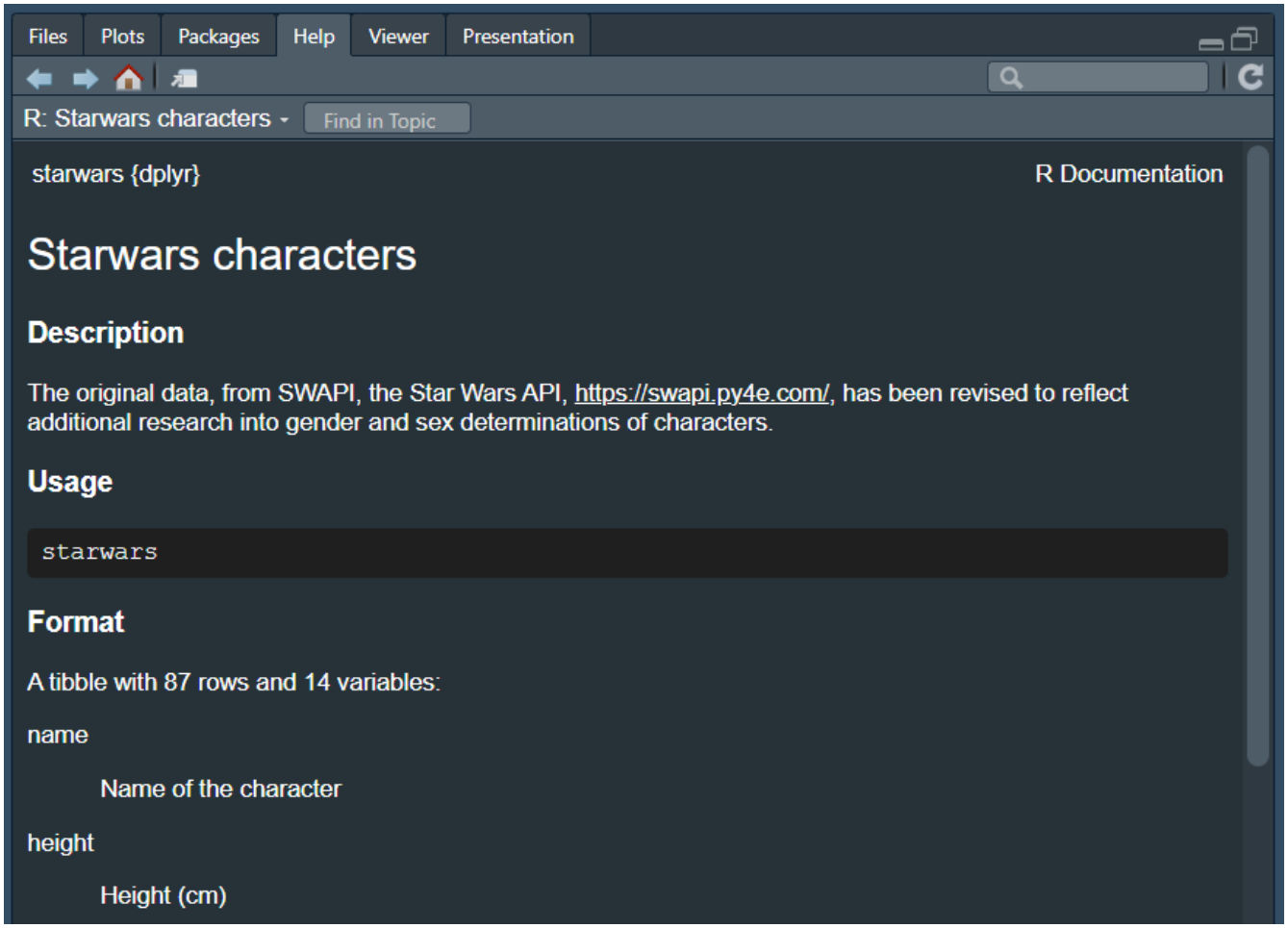
```
1 glimpse(starwars)
```

```
Rows: 87
Columns: 14
$ name       <chr> "Luke Skywalker", "C-3PO", "R2-D2", "Darth Vader", "Leia Or…
$ height     <int> 172, 167, 96, 202, 150, 178, 165, 97, 183, 182, 188, 180, 2…
$ mass       <dbl> 77.0, 75.0, 32.0, 136.0, 49.0, 120.0, 75.0, 32.0, 84.0, 77.…
$ hair_color <chr> "blond", NA, NA, "none", "brown", "brown, grey", "brown", N…
$ skin_color <chr> "fair", "gold", "white, blue", "white", "light", "light", "…
$ eye_color  <chr> "blue", "yellow", "red", "yellow", "brown", "blue", "blue",…
$ birth_year <dbl> 19.0, 112.0, 33.0, 41.9, 19.0, 52.0, 47.0, NA, 24.0, 57.0, …
$ sex        <chr> "male", "none", "none", "male", "female", "male", "female",…
$ gender     <chr> "masculine", "masculine", "masculine", "masculine", "femini…
$ homeworld  <chr> "Tatooine", "Tatooine", "Naboo", "Tatooine", "Alderaan", "T…
$ species    <chr> "Human", "Droid", "Droid", "Human", "Human", "Human", "Huma…
$ films      <list> <"A New Hope", "The Empire Strikes Back", "Return of the J…
$ vehicles   <list> <"Snowspeeder", "Imperial Speeder Bike">, <>, <>, <>, "Imp…
$ starships  <list> <"X-wing", "Imperial shuttle">, <>, <>, "TIE Advanced x1",…
```

Pour en savoir plus…

Pour avoir le plus d'information possible, l'aide reste la meilleure option

```
1  ?starwars
```



Description des données

Une information importante à connaitre est le nombre de ligne et de colonnes dans le jeu de données

```
nrow(starwars) # nombre de lignes
```

```
[1] 87
```

```
ncol(starwars) # nombre de colonnes
```

```
[1] 14
```

```
dim(starwars) # dimensions du tableau: lignes x colonnes
```

```
[1] 87 14
```

Exploration de données

Qu’est ce que c’est ?

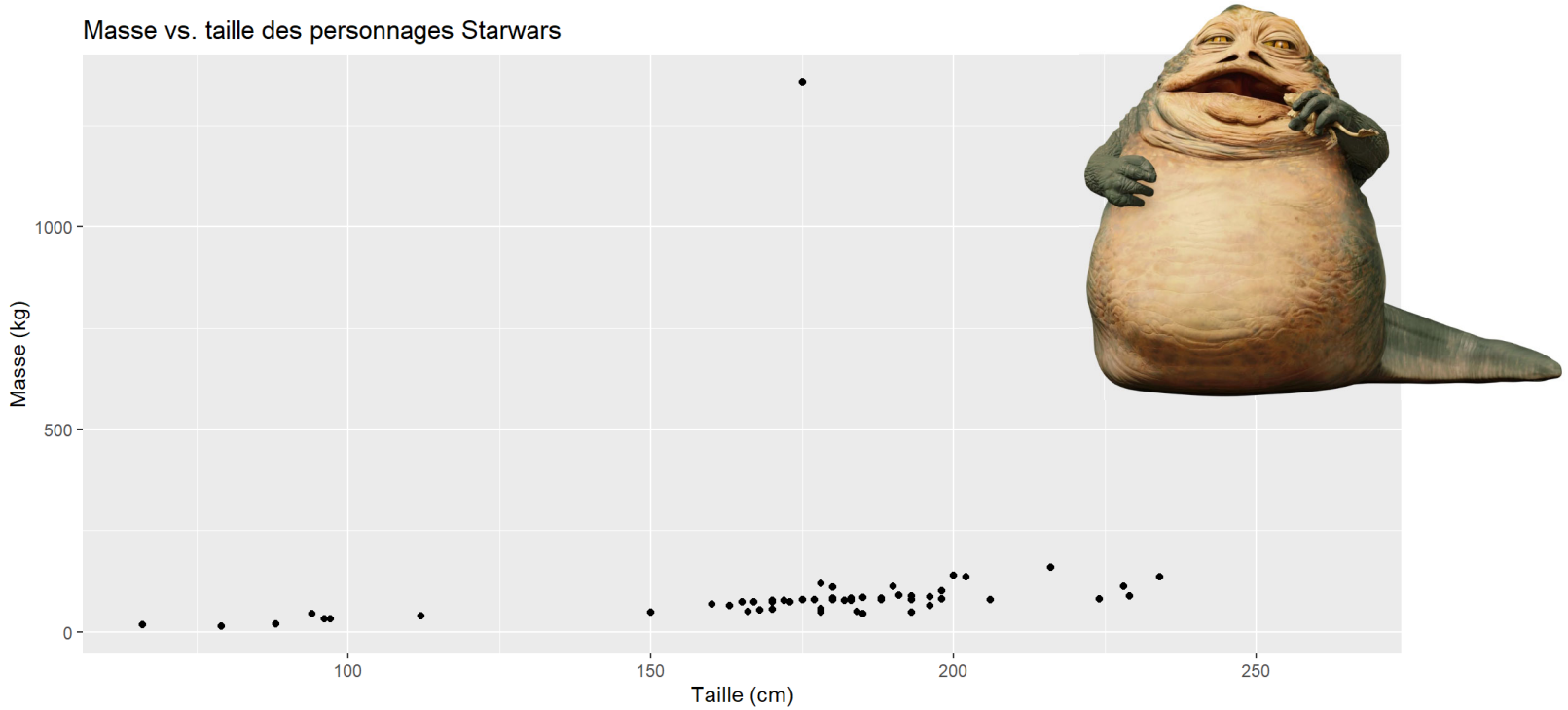
L’idée de l’exploration de données est de chercher à résumer les principales caractérostiques du jeu de données.

Souvent, cette analyse est visuelle: c’est de la visualisation de données !

Elle s’accompagne aussi d’analyses statistiques élémentaires.

Masse vs Taille

> Comment décririez-vous la relation entre la masse et la taille ? Y a-t-il des outliers ?



Data Viz

Data Visualisation

> *“The simple graph has brought more information to the data analyst’s mind than any other device.” — John Tukey*

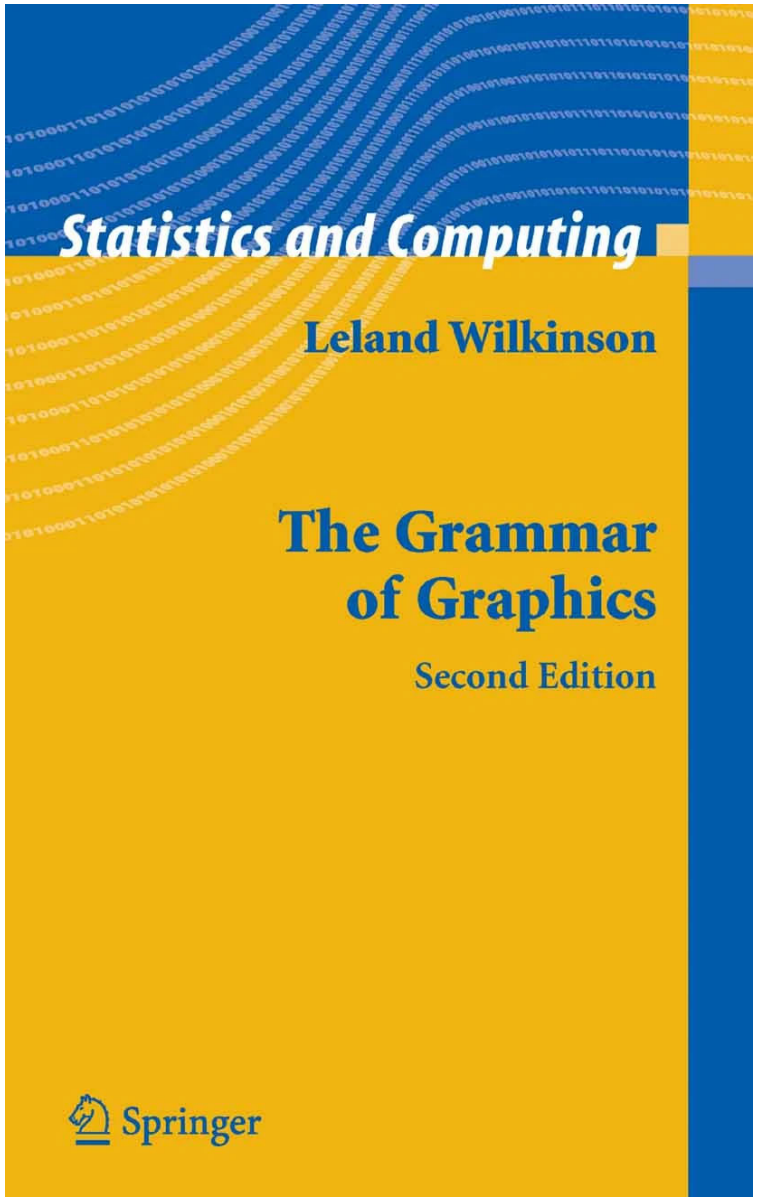
La visualisation de données cherche à étudier comment représenter les données.

Il y a de nombreux outils pour la DataViz

- R est l’un d’eux
- Il y a plusieurs modules pour visualiser les données dans R
  - Nous allons utiliser `ggplot2`

ggplot2

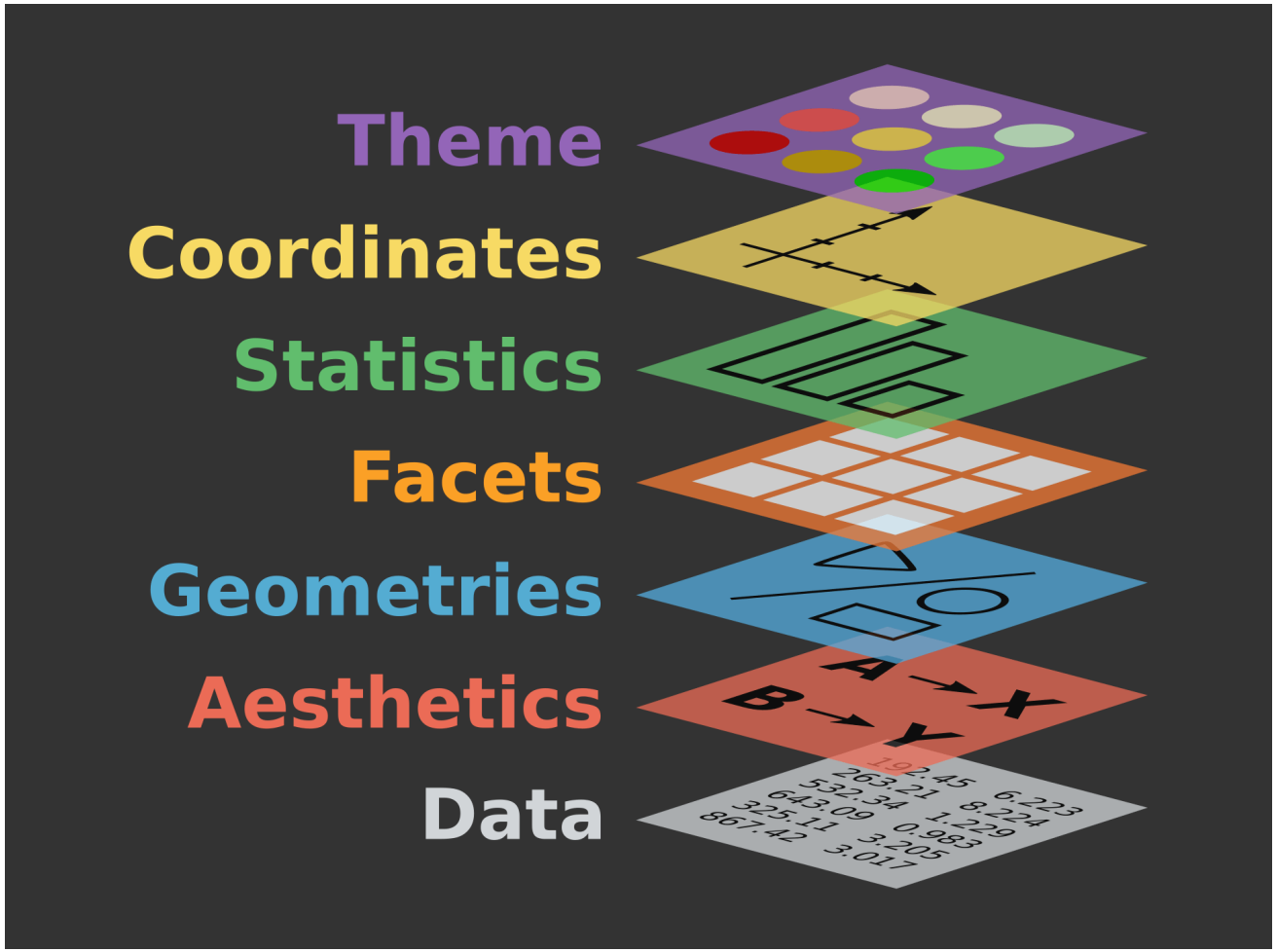
- `ggplot2` fait partie du Tidyverse et sert à la visualisation de données
- gg veut dire *Grammar of Graphics*
- Inspiré du livre *Grammar of Graphics* de Leland Wilkinson



(Wilkinson, 2005)

Grammar of Graphics

Permet de décrire toutes les briques d'un graphe

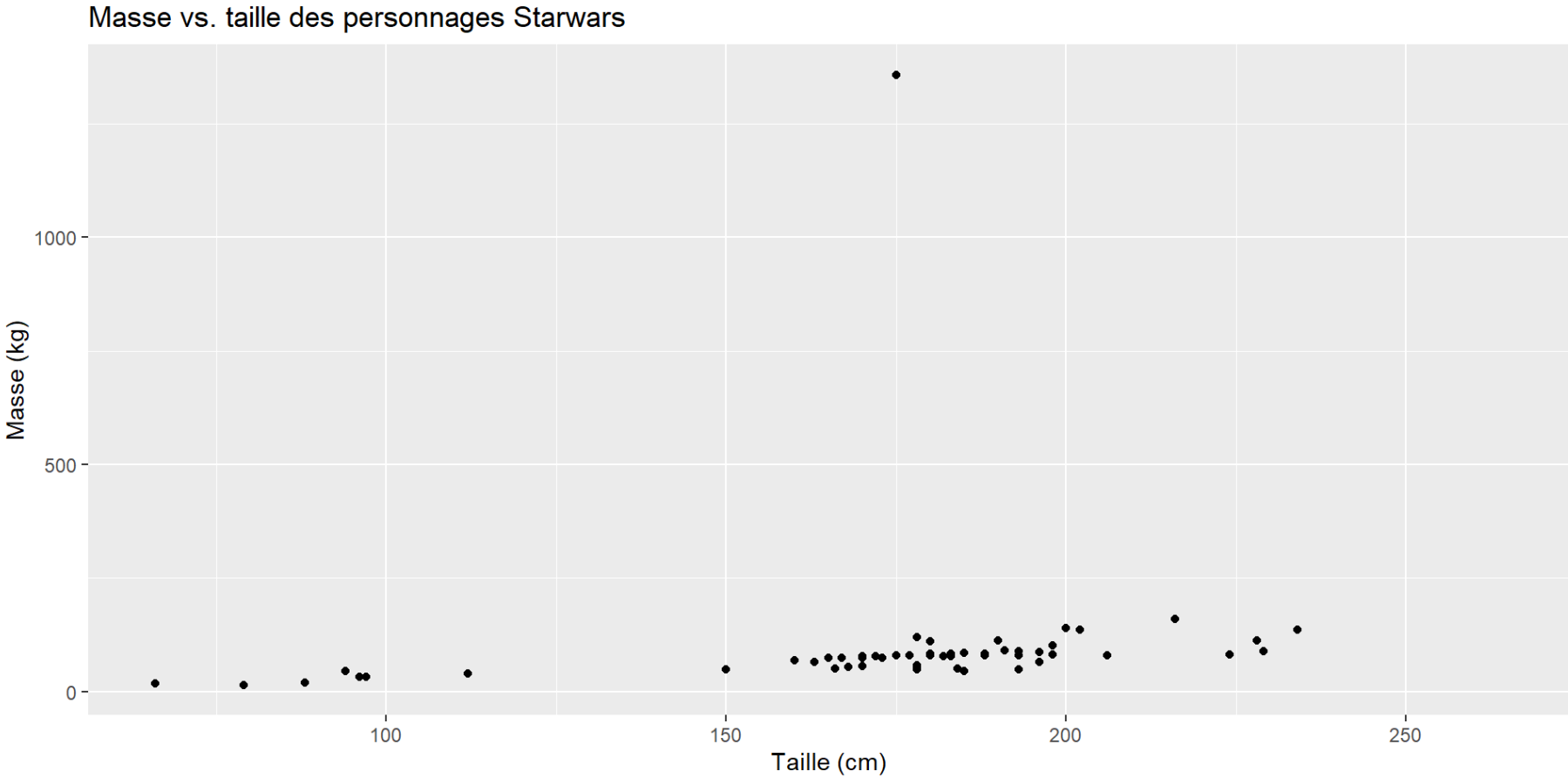


source

Masse vs Taille

```
1 ggplot(data = starwars, mapping = aes(x = height, y = mass)) +
2   geom_point() +
3   labs(title = "Masse vs. taille des personnages Starwars",
4        x = "Taille (cm)", y = "Masse (kg)")
```

```
Warning: Removed 28 rows containing missing values or values outside the scale range
(`geom_point()`).
```



Kesako

- Quelle fonction s'occupe du plotting ?
- Quel jeu de données est visualisé ?
- Quelles variables sont projetées sur quels éléments (*aesthetics*) du graphe ?
- Que siginifie le warning ?

```
ggplot(data = starwars, mapping = aes(x = height, y = mass)) +
  geom_point() +
  labs(title = "Masse vs. taille des personnages Starwars",
       x = "Taille (cm)", y = "Masse (kg)")
```

Hello ggplot2 !

`ggplot()` est la fonction principale dans ggplot2. Elle est toujours présente.

Les graphes sont faits en couches. Nous pouvons les résumer de la manière suivante:

```
ggplot(data = [dataset],
       mapping = aes(x = [x-variable], y = [y-variable])) +
   geom_xxx() +
   other options
```

Pourquoi visualiser ?

Le quartet d’Anscombe

	set	x	y		set	x	y
1	I	10	8.04	12	II	10	9.14
2	I	8	6.95	13	II	8	8.14
3	I	13	7.58	14	II	13	8.74
4	I	9	8.81	15	II	9	8.77
5	I	11	8.33	16	II	11	9.26
6	I	14	9.96	17	II	14	8.10
7	I	6	7.24	18	II	6	6.13
8	I	4	4.26	19	II	4	3.10
9	I	12	10.84	20	II	12	9.13
10	I	7	4.82	21	II	7	7.26
11	I	5	5.68	22	II	5	4.74

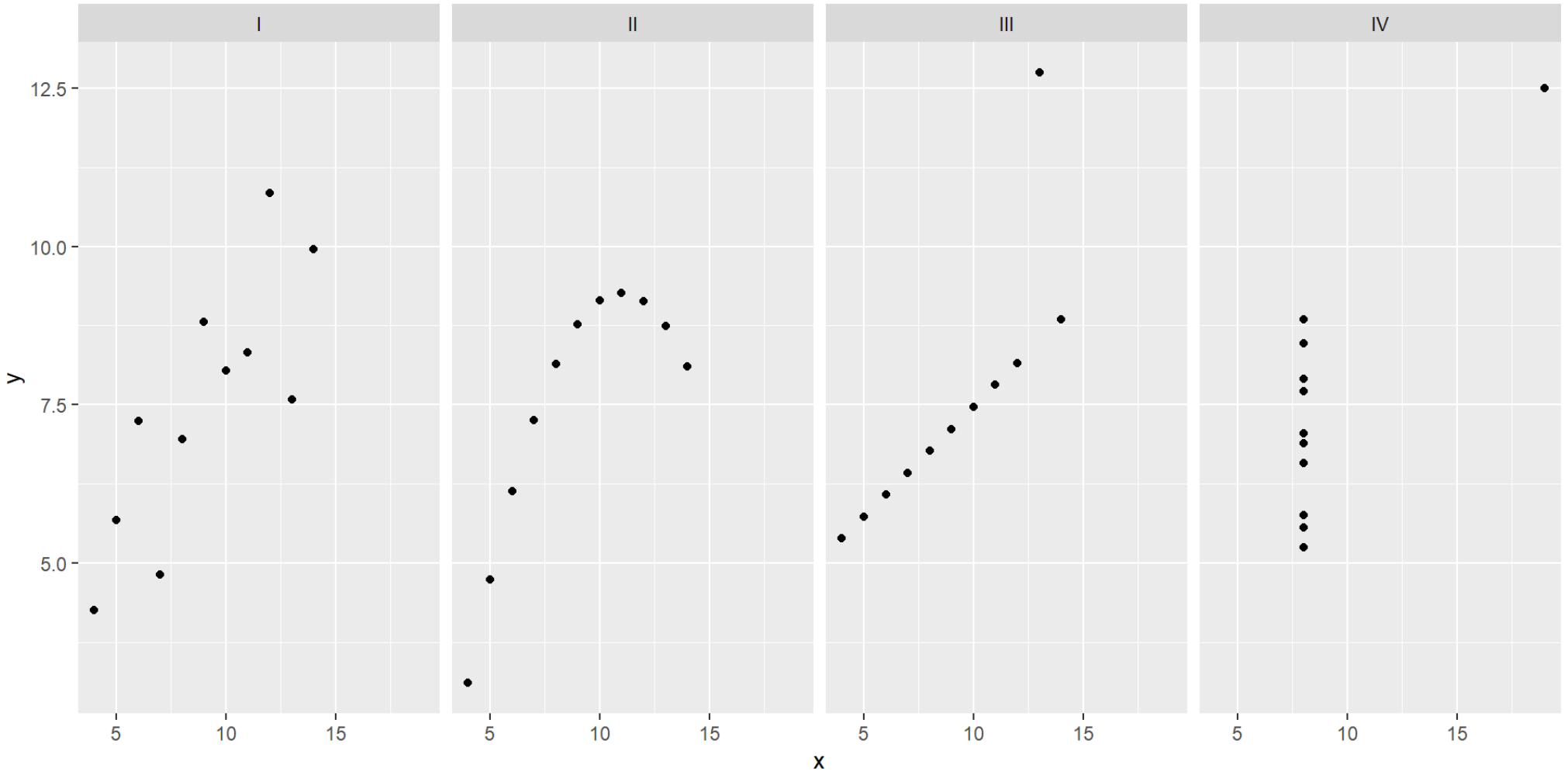
	set	x	y		set	x	y
23	III	10	7.46	34	IV	8	6.58
24	III	8	6.77	35	IV	8	5.76
25	III	13	12.74	36	IV	8	7.71
26	III	9	7.11	37	IV	8	8.84
27	III	11	7.81	38	IV	8	8.47
28	III	14	8.84	39	IV	8	7.04
29	III	6	6.08	40	IV	8	5.25
30	III	4	5.39	41	IV	19	12.50
31	III	12	8.15	42	IV	8	5.56
32	III	7	6.42	43	IV	8	7.91
33	III	5	5.73	44	IV	8	6.89

Descrivons chaque jeu de données

```
quartet %>%
  group_by(set) %>%
  summarise(
    mean_x = mean(x),
    mean_y = mean(y),
    sd_x = sd(x),
    sd_y = sd(y),
    r = cor(x, y)
  )
```

```
# A tibble: 4 × 6
  set   mean_x mean_y  sd_x  sd_y     r
  <fct>  <dbl>  <dbl> <dbl> <dbl> <dbl>
1 I          9   7.50  3.32  2.03 0.816
2 II         9   7.50  3.32  2.03 0.816
3 III        9   7.5   3.32  2.03 0.816
4 IV         9   7.50  3.32  2.03 0.817
```

Maintenant visualisons !

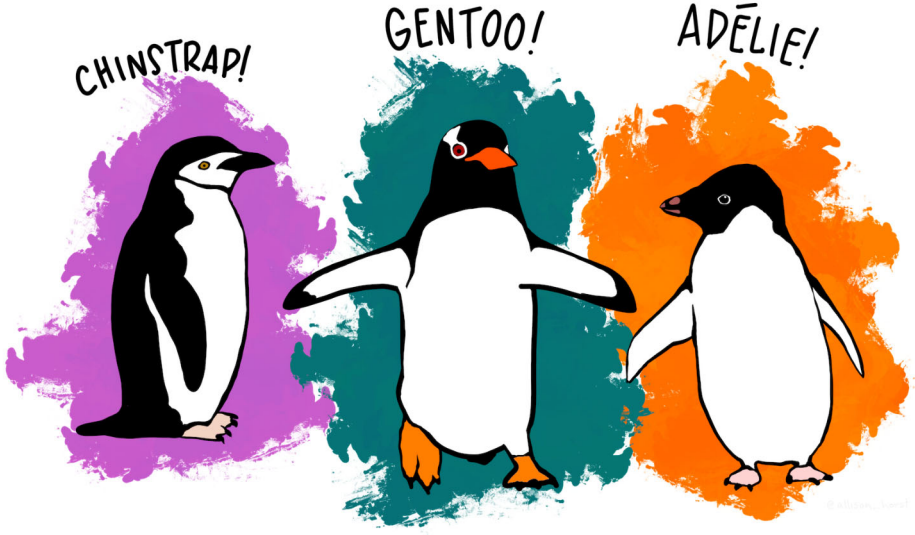


Let’s Dive in !

Dataset: Palmer Penguins

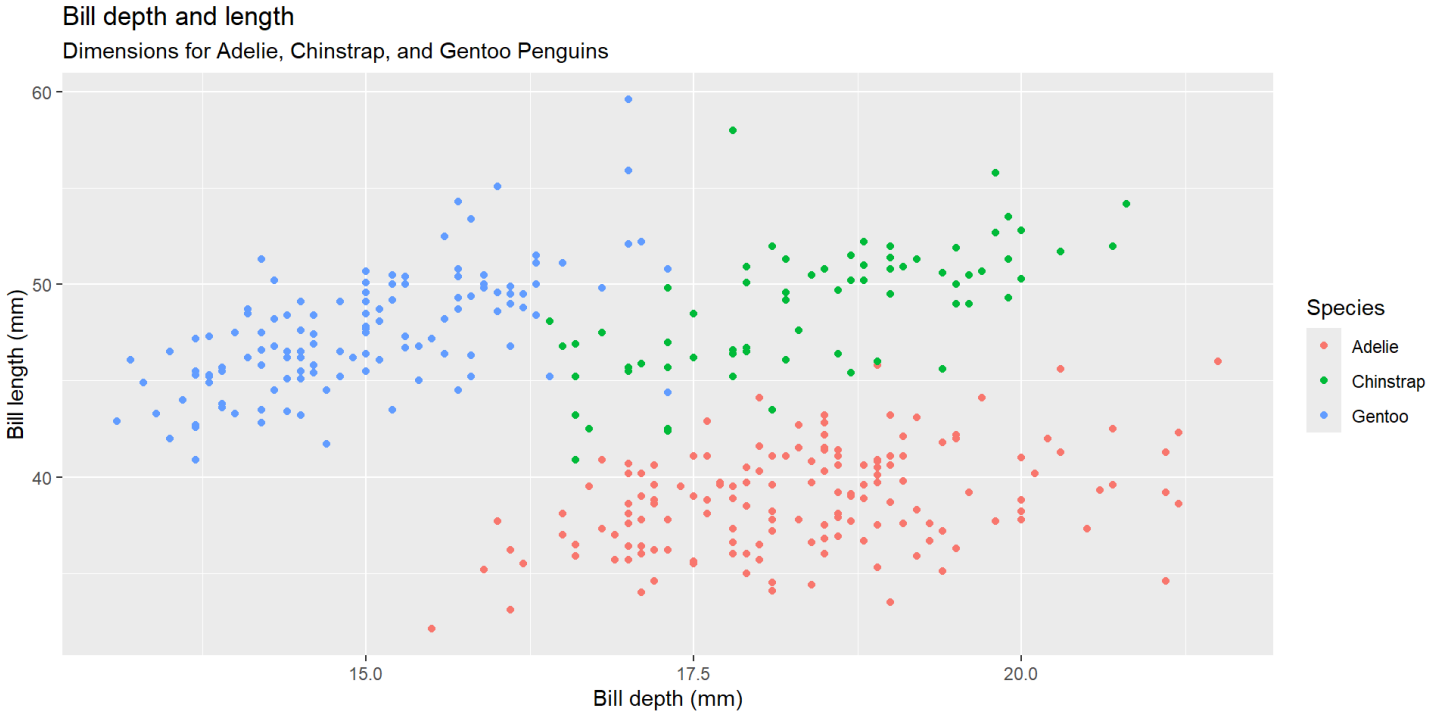
```
1 library(palmerpenguins)
2 glimpse(penguins)
```

```
Rows: 344
Columns: 8
$ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adel…
$ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgerse…
$ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34.1, …
$ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18.1, …
$ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, 186…
$ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 3475, …
$ sex               <fct> male, female, female, NA, female, male, female, male…
$ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007…
```



source

Graphe Code


Bill depth and length
Dimensions for Adelie, Chinstrap, and Gentoo Penguins
Bill length (mm)
60
50
40
15.0
17.5
20.0
Bill depth (mm)
Species
Adelie
Chinstrap
Gentoo

En détails

Détaillons tout ça ! (1)

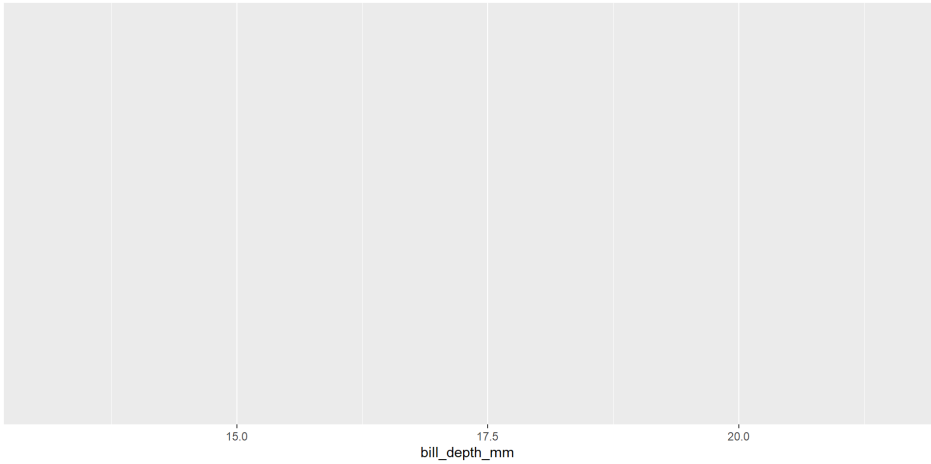
On commence avec le dataset `penguins`

```
1  ggplot(data = penguins)
```

Détaillons tout ça ! (2)

> On commence avec le dataset penguins, on map l'épaisseur du bec sur l'axe x

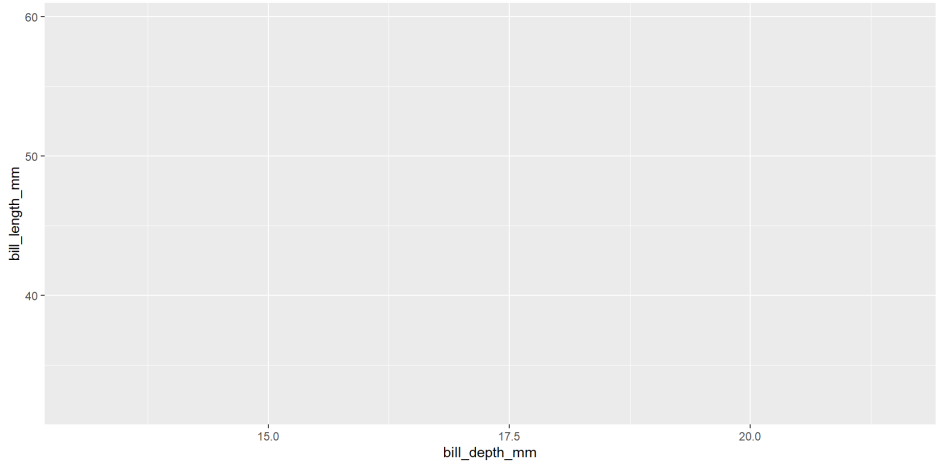
```
1 ggplot(data = penguins,
2        mapping = aes(x = bill_depth_mm))
```



Détaillons tout ça ! (3)

On commence avec le dataset `penguins`, on map l'épaisseur du bec sur l'axe x, `et la longueur du bec sur l'axe y`

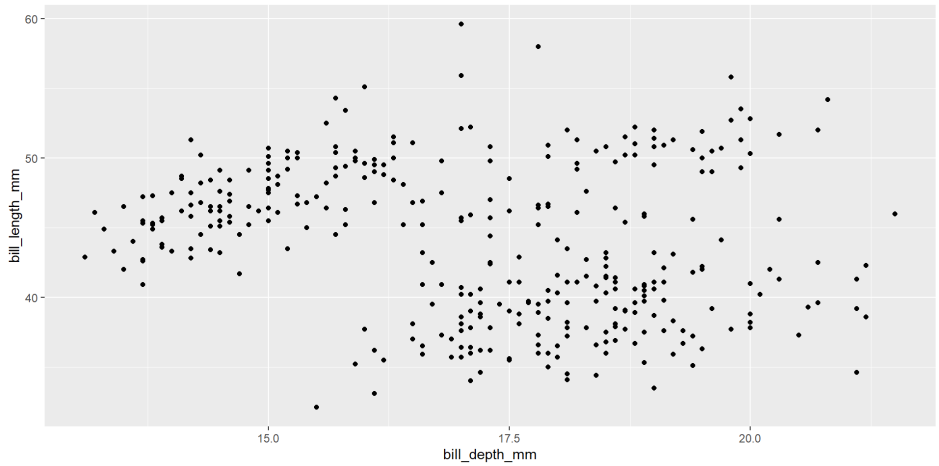
```
1  ggplot(data = penguins,
2         mapping = aes(x = bill_depth_mm,
3                       y = bill_length_mm))
```



Détaillons tout ça ! (4)

On commence avec le dataset `penguins`, on map l'épaisseur du bec sur l'axe x, et la longueur du bec sur l'axe y. `Représente chaque observation par un point`

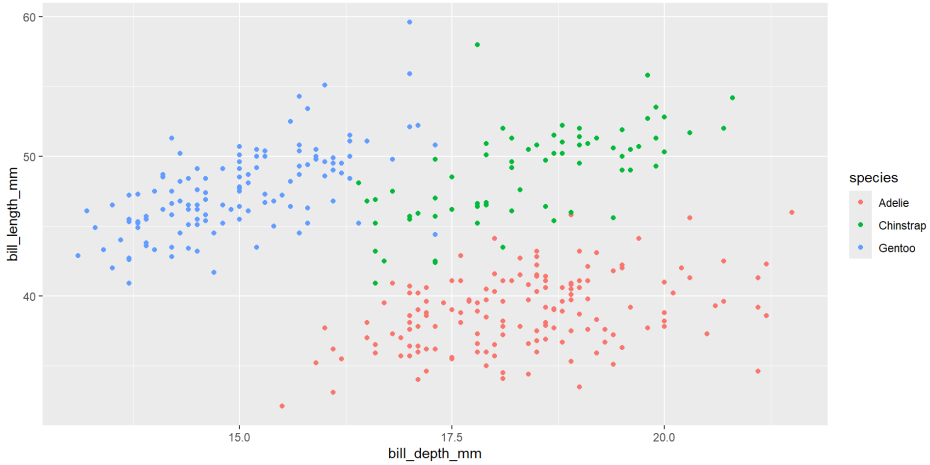
```
1  ggplot(data = penguins,
2         mapping = aes(x = bill_depth_mm,
3                       y = bill_length_mm)) +
4    geom_point()
```



Détaillons tout ça ! (5)

On commence avec le dataset `penguins`, on map l'épaisseur du bec sur l'axe x, et la longueur du bec sur l'axe y. Représente chaque observation par un point et `map la couleur des points en fonction de l'espèce`

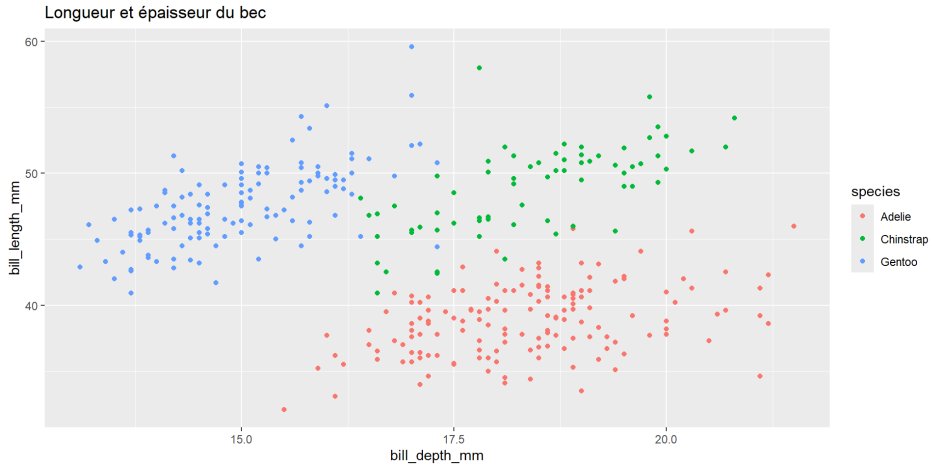
```
1  ggplot(data = penguins,
2         mapping = aes(x = bill_depth_mm,
3                       y = bill_length_mm,
4                       colour = species)) +
5    geom_point()
```



Détaillons tout ça ! (6)

On commence avec le dataset `penguins`, on map l'épaisseur du bec sur l'axe x, et la longueur du bec sur l'axe y. Représente chaque observation par un point et map la couleur des points en fonction de l'espèce. `Ajoute comme titre "Longueur et épaisseur du bec"`

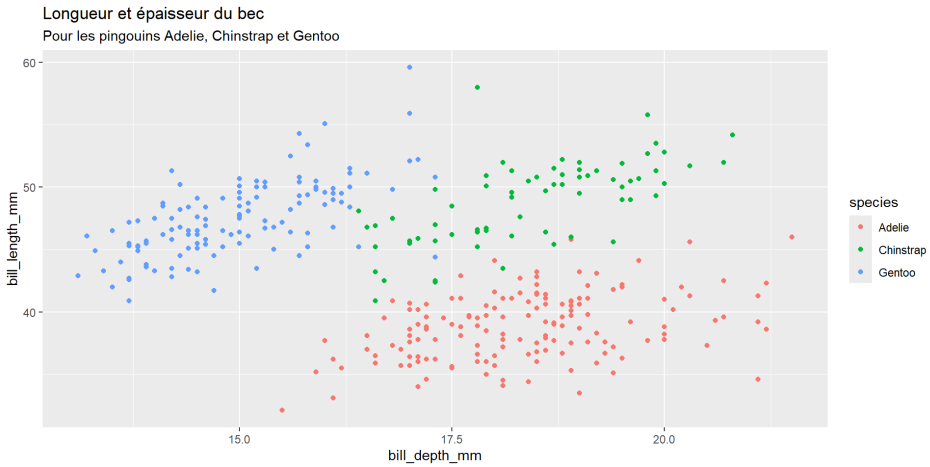
```
ggplot(data = penguins,
       mapping = aes(x = bill_depth_mm,
                     y = bill_length_mm,
                     colour = species)) +
  geom_point() +
  labs(title = "Longueur et épaisseur du bec"
```



Détaillons tout ça ! (7)

On commence avec le dataset `penguins`, on map l'épaisseur du bec sur l'axe x, et la longueur du bec sur l'axe y. Représente chaque observation par un point et map la couleur des points en fonction de l'espèce. Ajoute comme titre "Longueur et épaisseur du bec", `et comme sous titre "Pour les pingouins Adelie, Chinstrap et Gentoo"`.

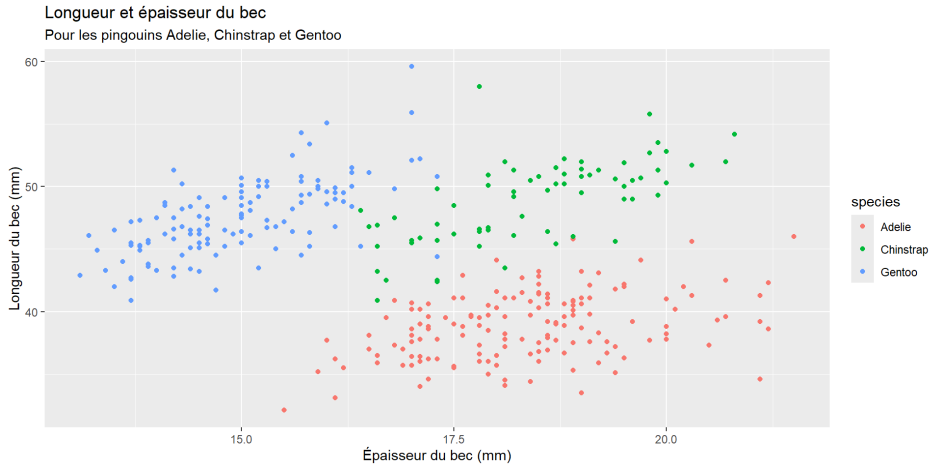
```
ggplot(data = penguins,
       mapping = aes(x = bill_depth_mm,
                     y = bill_length_mm,
                     colour = species)) +
  geom_point() +
  labs(title = "Longueur et épaisseur du bec"
       subtitle = "Pour les pingouins Adelie,
```



Détaillons tout ça ! (8)

On commence avec le dataset `penguins`, on map l'épaisseur du bec sur l'axe x, et la longueur du bec sur l'axe y. Représente chaque observation par un point et map la couleur des points en fonction de l'espèce. Ajoute comme titre "Longueur et épaisseur du bec", et comme sous titre "Pour les pingouins Adelie, Chinstrap et Gentoo". `Nomme les axes x et y, "Épaisseur du bec (mm)" et "Longueur du bec (mm)"`.

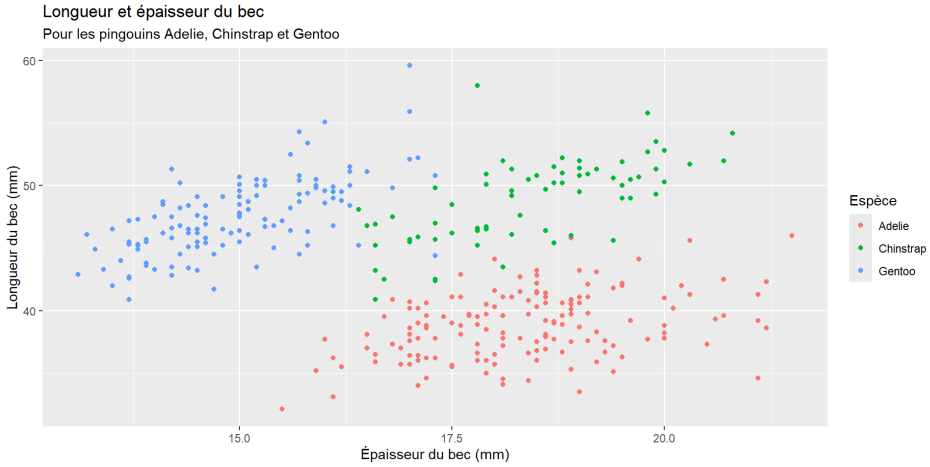
```
ggplot(data = penguins,
       mapping = aes(x = bill_depth_mm,
                     y = bill_length_mm,
                     colour = species)) +
  geom_point() +
  labs(title = "Longueur et épaisseur du bec"
       subtitle = "Pour les pingouins Adelie,
       x = "Épaisseur du bec (mm)", y = "Long
```



Détaillons tout ça ! (9)

On commence avec le dataset `penguins`, on map l'épaisseur du bec sur l'axe x, et la longueur du bec sur l'axe y. Représente chaque observation par un point et map la couleur des points en fonction de l'espèce. Ajoute comme titre "Longueur et épaisseur du bec", et comme sous titre "Pour les pingouins Adelie, Chinstrap et Gentoo". Nomme les axes x et y, "Épaisseur du bec (mm)" et "Longueur du bec (mm)", `et la légende "Espèce"`.

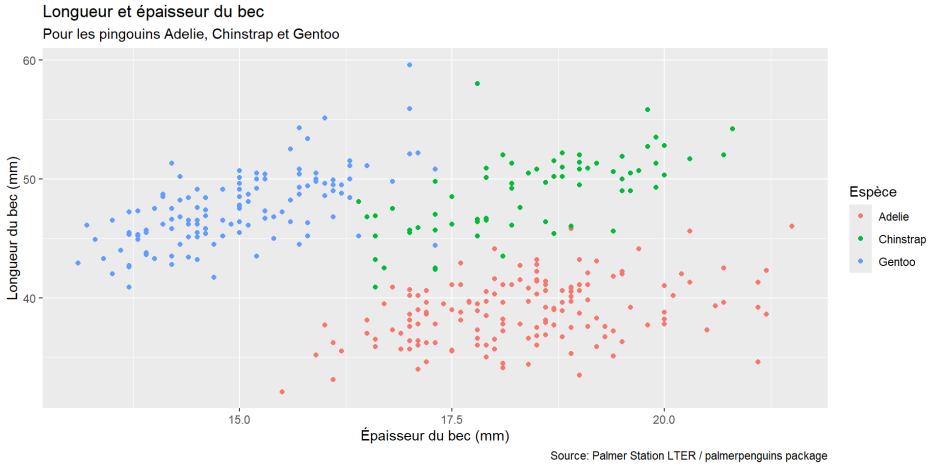
```
1  ggplot(data = penguins,
2         mapping = aes(x = bill_depth_mm,
3                       y = bill_length_mm,
4                       colour = species)) +
5    geom_point() +
6    labs(title = "Longueur et épaisseur du bec"
7         subtitle = "Pour les pingouins Adelie,
8         x = "Épaisseur du bec (mm)", y = "Long
9         colour = "Espèce")
```



Détaillons tout ça ! (10)

On commence avec le dataset `penguins`, on map l'épaisseur du bec sur l'axe x, et la longueur du bec sur l'axe y. Représente chaque observation par un point et map la couleur des points en fonction de l'espèce. Ajoute comme titre "Longueur et épaisseur du bec", et comme sous titre "Pour les pingouins Adelie, Chinstrap et Gentoo". Nomme les axes x et y, "Épaisseur du bec (mm)" et "Longueur du bec (mm)", et la légende "Espèce". `Ajoute une phrase pour indiquer la source des données`.

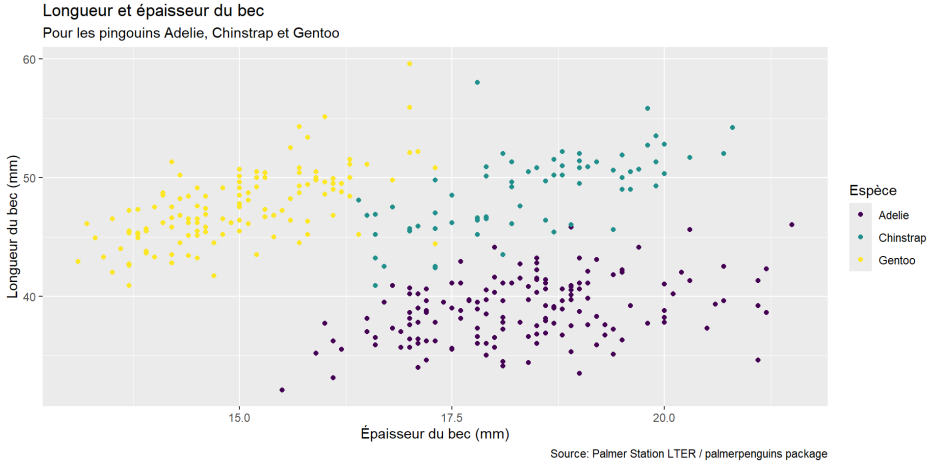
```
ggplot(data = penguins,
       mapping = aes(x = bill_depth_mm,
                     y = bill_length_mm,
                     colour = species)) +
  geom_point() +
  labs(title = "Longueur et épaisseur du bec"
       subtitle = "Pour les pingouins Adelie,
       x = "Épaisseur du bec (mm)", y = "Long
       colour = "Espèce",
       caption = "Source: Palmer Station LTER
```



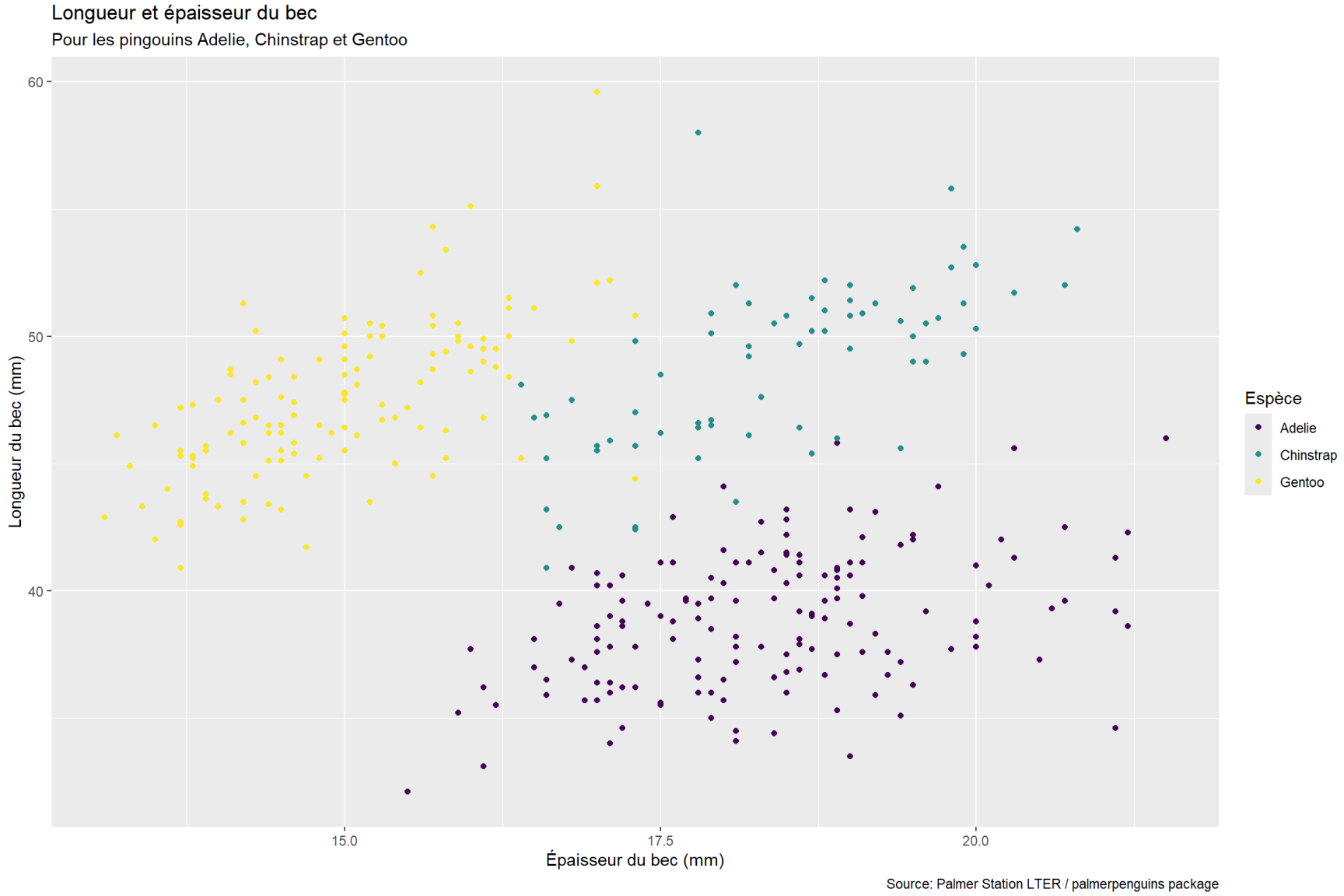
Détaillons tout ça ! (11)

On commence avec le dataset `penguins`, on map l'épaisseur du bec sur l'axe x, et la longueur du bec sur l'axe y. Représente chaque observation par un point et map la couleur des points en fonction de l'espèce. Ajoute comme titre "Longueur et épaisseur du bec", et comme sous titre "Pour les pingouins Adelie, Chinstrap et Gentoo". Nomme les axes x et y, "Épaisseur du bec (mm)" et "Longueur du bec (mm)", et la légende "Espèce". Ajoute une phrase pour indiquer la source des données. `Pour finir, utilise une échelle de couleur adaptée aux personnes daltoniennes.`

```
ggplot(data = penguins,
       mapping = aes(x = bill_depth_mm,
                     y = bill_length_mm,
                     colour = species)) +
  geom_point() +
  labs(title = "Longueur et épaisseur du bec"
       subtitle = "Pour les pingouins Adelie,
       x = "Épaisseur du bec (mm)", y = "Long
       colour = "Espèce",
       caption = "Source: Palmer Station LTER
  scale_colour_viridis_d()
```



Graphe Code


Longueur et épaisseur du bec
Pour les pingouins Adelie, Chinstrap et Gentoo
Longueur du bec (mm)
60
50
40
15.0
17.5
20.0
Épaisseur du bec (mm)
Espèce
Adelie
Chinstrap
Gentoo
Source: Palmer Station LTER / palmerpenguins package

Arguments

Les deux premiers arguments, `data` et `mapping` peuvent être écrits directement pour alléger la notation

```
ggplot(data = penguins,
       mapping = aes(x = bill_depth_mm,
                     y = bill_length_mm,
                     colour = species)) +
  geom_point() +
  scale_colour_viridis_d()
```

```
ggplot(penguins,
       aes(x = bill_depth_mm,
           y = bill_length_mm,
           colour = species)) +
  geom_point() +
  scale_colour_viridis_d()
```

Aesthetics

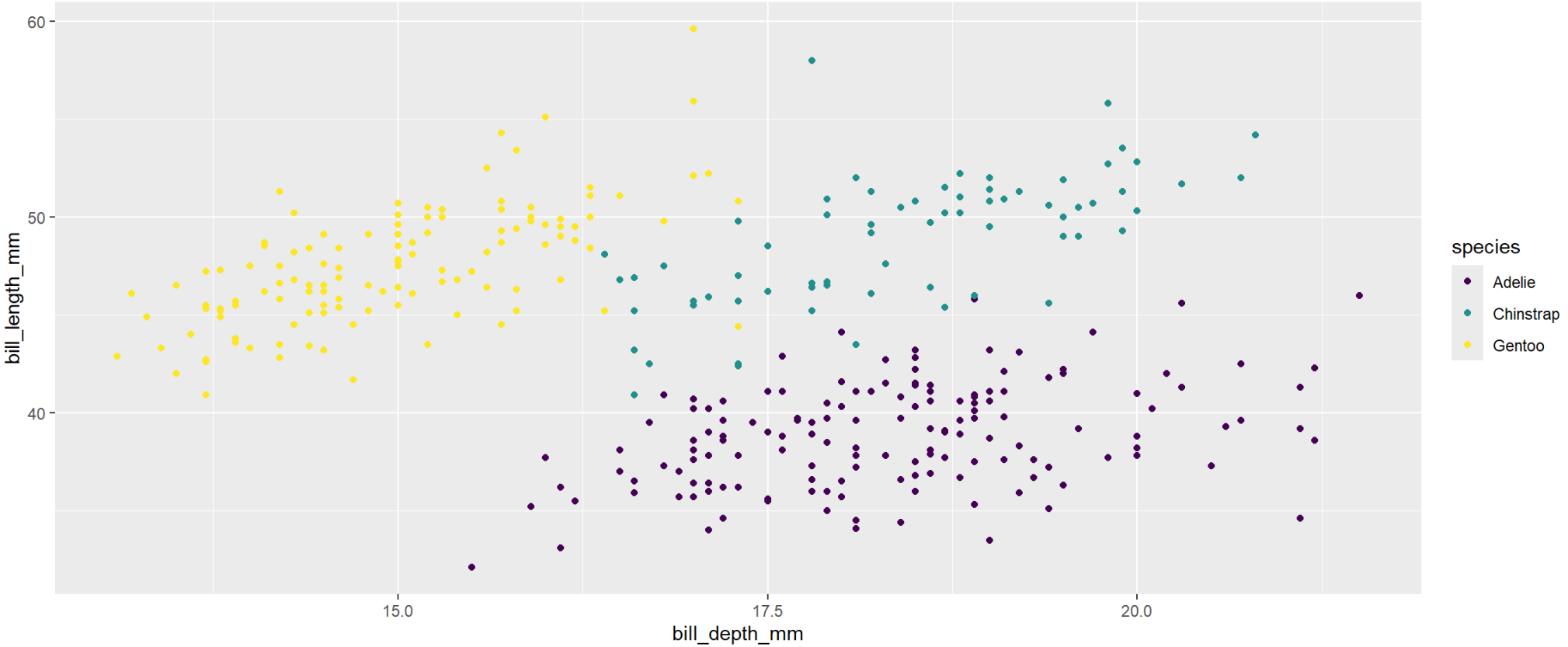
Options

Les caractéristiques principales du graphe peuvent être `envoyées directement sur des variables` dans le jeu de données:

- `colour`
- `shape`
- `size`
- `alpha`

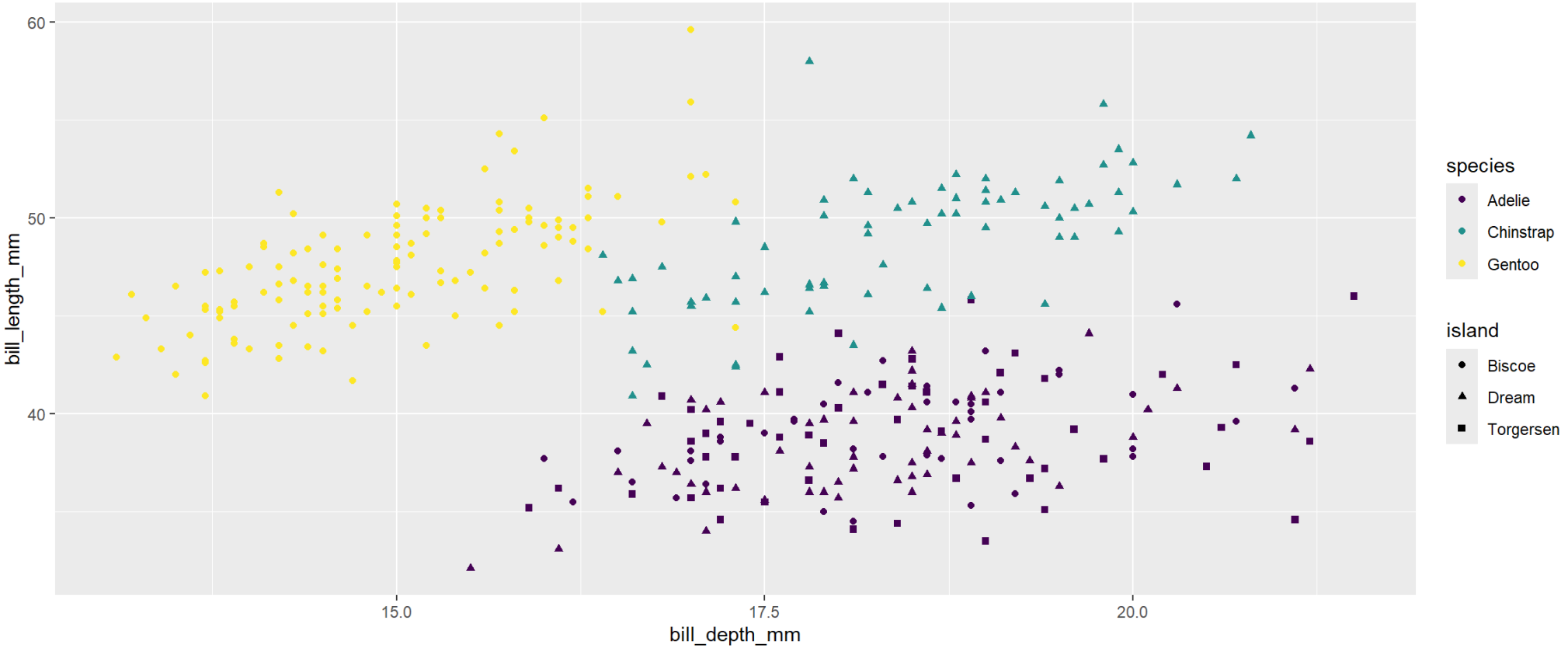
Colour

```
ggplot(data = penguins,
       mapping = aes(x = bill_depth_mm,
                     y = bill_length_mm,
                     colour = species)) +
  geom_point() +
  scale_colour_viridis_d()
```



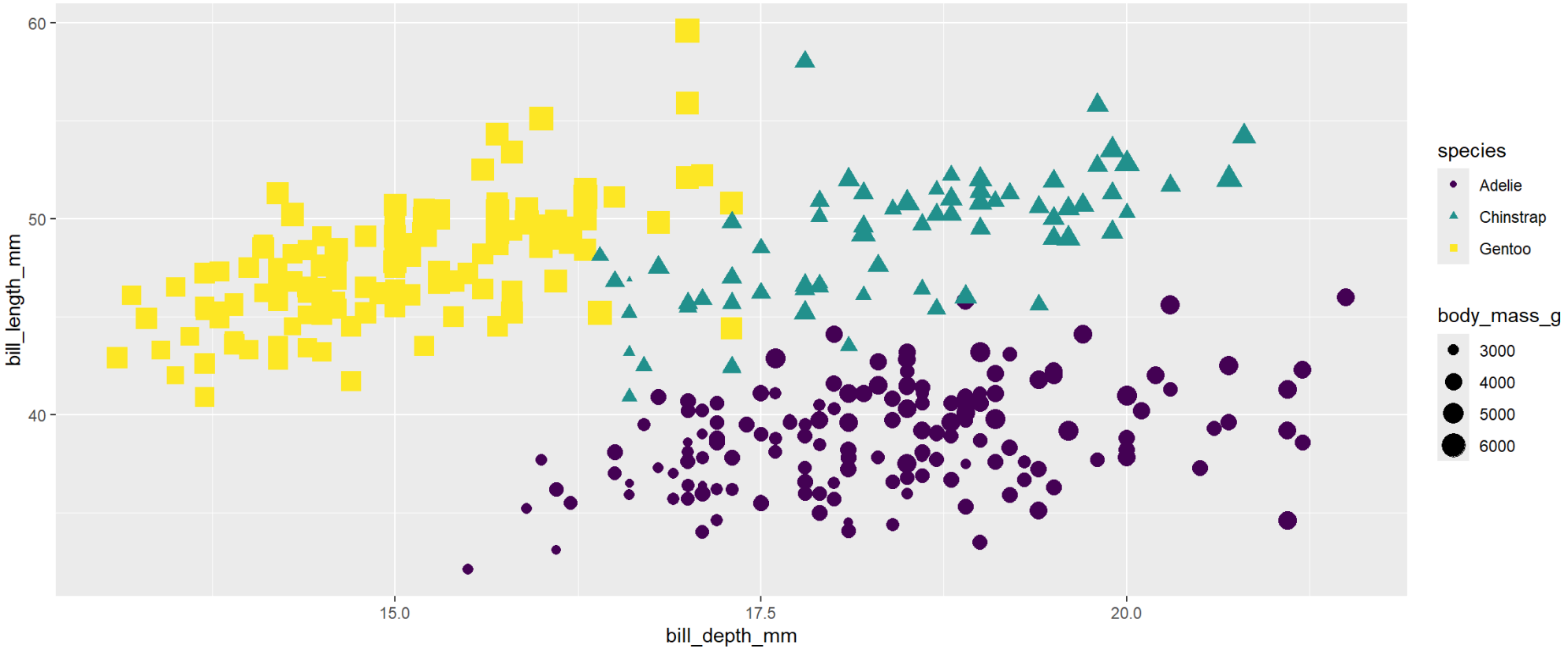
Shape

```
ggplot(penguins,
       aes(x = bill_depth_mm,
           y = bill_length_mm,
           colour = species,
           shape = island)) +
  geom_point() +
  scale_colour_viridis_d()
```



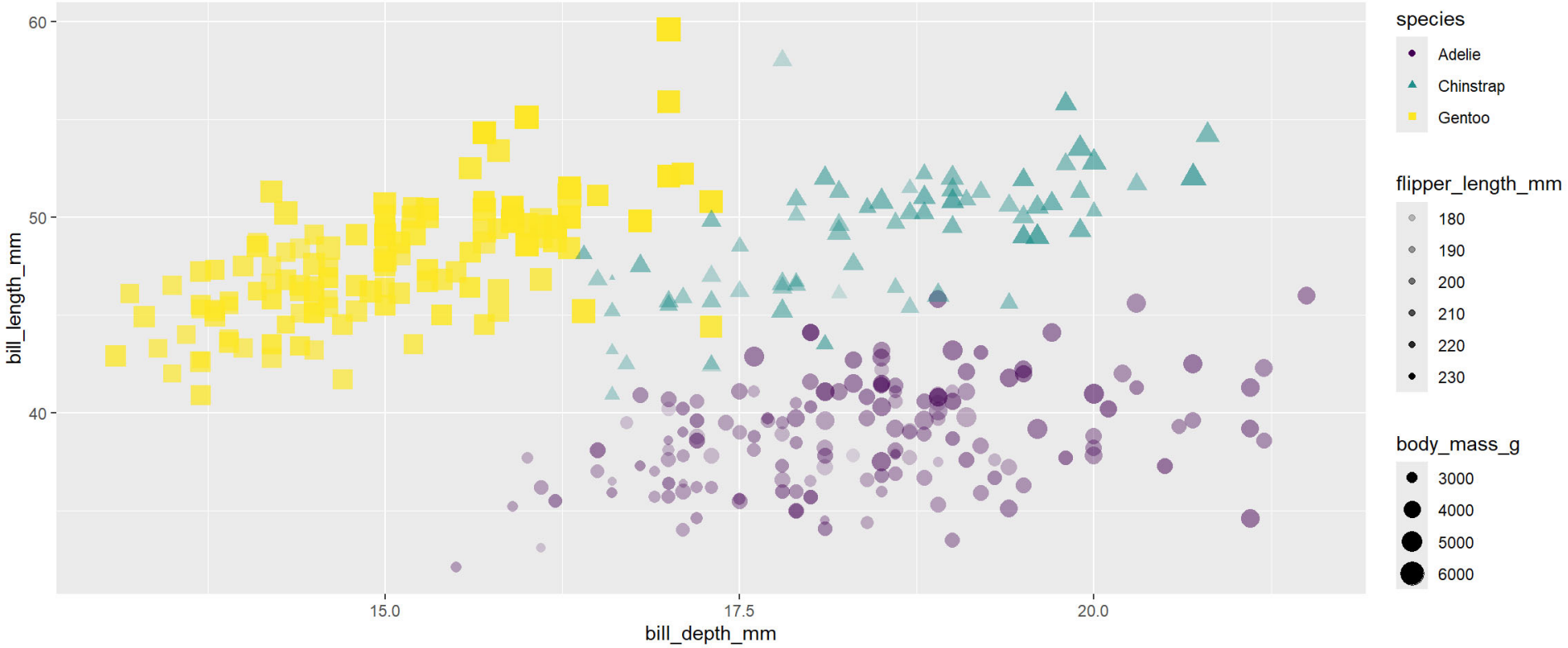
Size

```
ggplot(penguins,
       aes(x = bill_depth_mm,
           y = bill_length_mm,
           colour = species,
           shape = species,
           size = body_mass_g)) +
  geom_point() +
  scale_colour_viridis_d()
```



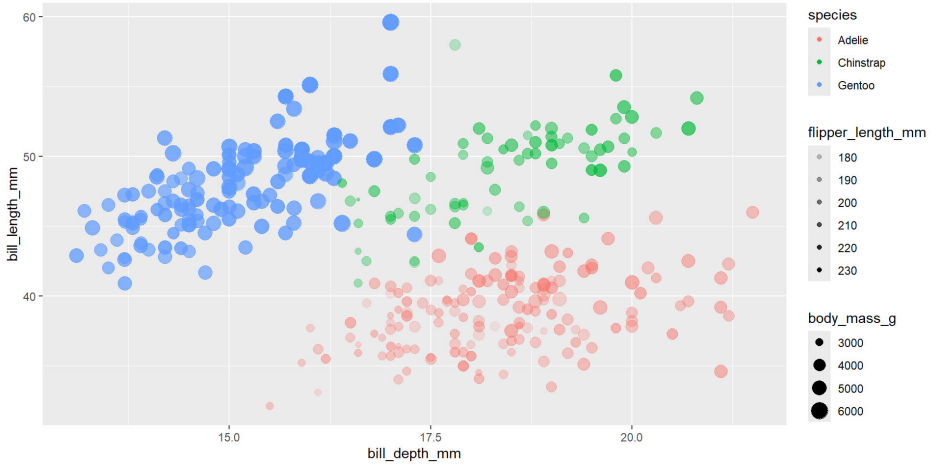
Alpha

```
ggplot(penguins,
       aes(x = bill_depth_mm,
           y = bill_length_mm,
           colour = species,
           shape = species,
           size = body_mass_g,
           alpha = flipper_length_mm)) +
  geom_point() +
  scale_colour_viridis_d()
```

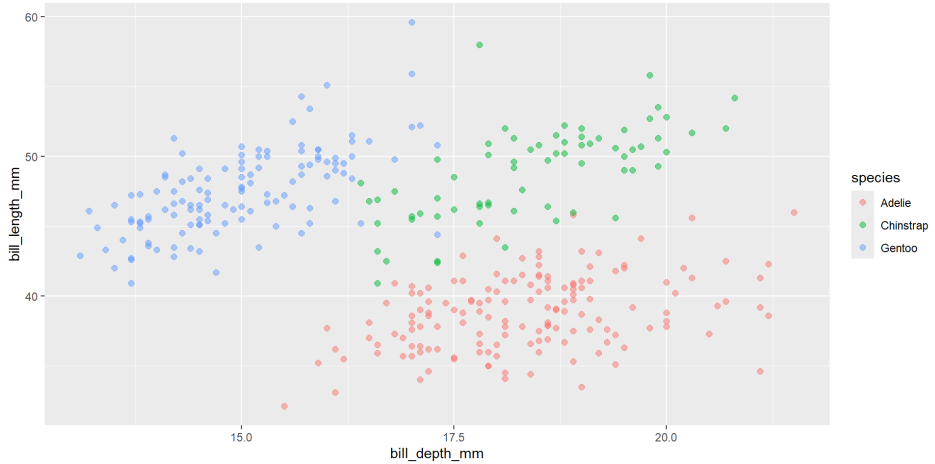


Mapping vs Setting

```
ggplot(penguins,
       aes(x = bill_depth_mm,
           y = bill_length_mm,
           colour = species,
           size = body_mass_g,
           alpha = flipper_length_mm)) +
  geom_point()
```




```
ggplot(penguins,
       aes(x = bill_depth_mm,
           y = bill_length_mm,
           colour = species)) +
  geom_point(size = 2, alpha = 0.5)
```



Mapping vs Setting

`Mapping`: Détermine la propriété (taille, alpha, forme, …) en fonction d'une propriété des données

- Va dans la fonction `aes()`

`Setting`: Fixer une propriété, qui ne dépend pas d'une variable dans les données

- Va dans la fonction `geom_*()`

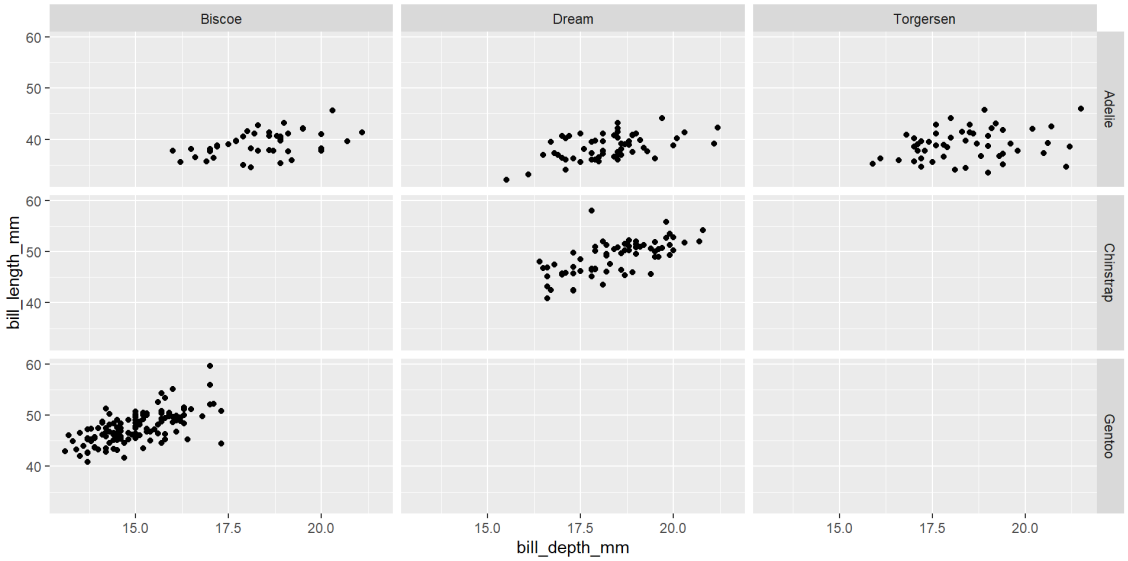
Faceting

Faceting

L'idée est de faire des sous graphes qui vont représentés des sous catégories dans les données.

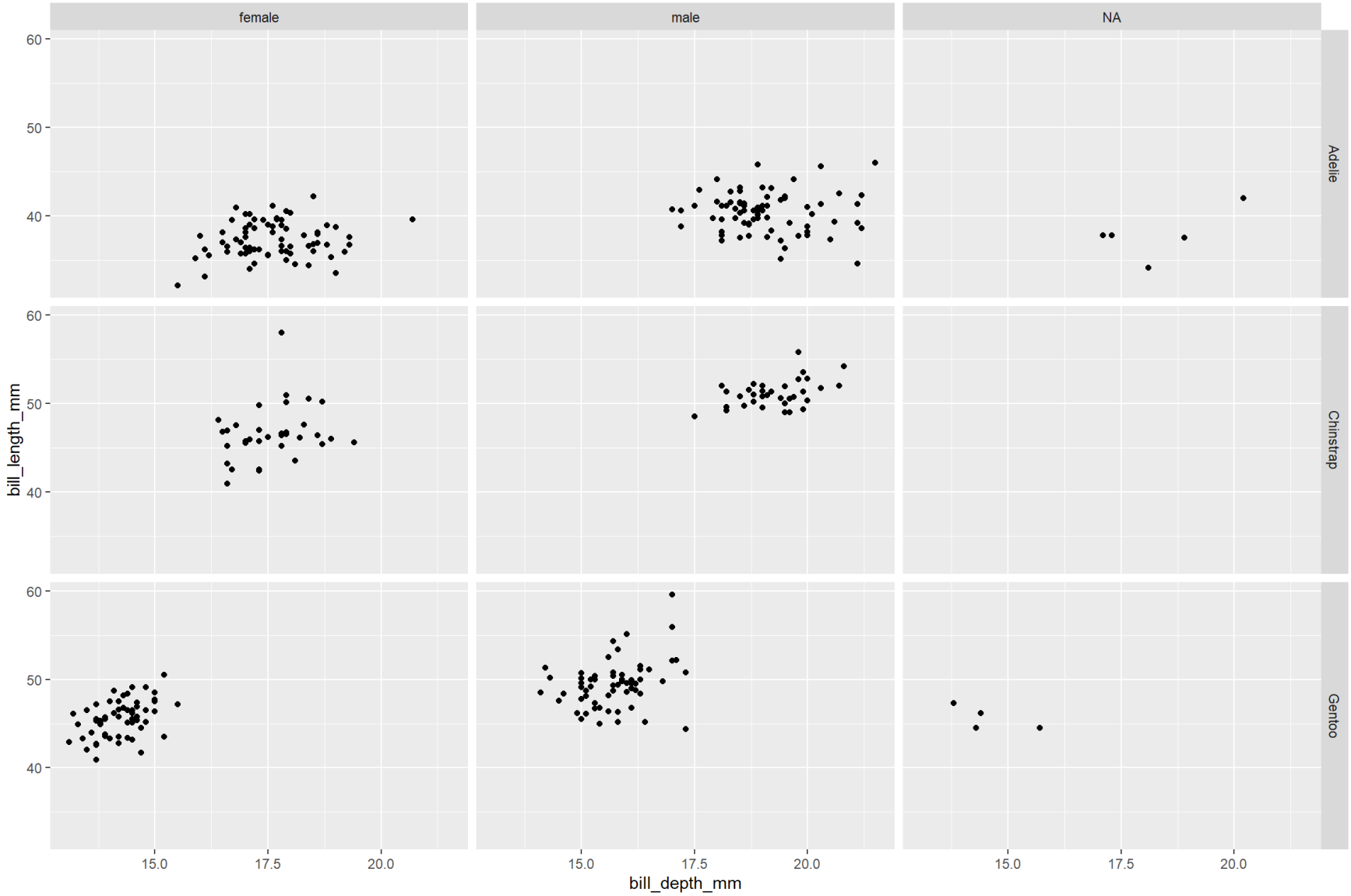
Utile pour explorer des relations conditionnelles dans les données.

```
1  ggplot(penguins,
2         aes(x = bill_depth_mm,
3             y = bill_length_mm)) +
4    geom_point() +
5    facet_grid(species ~ island)
```

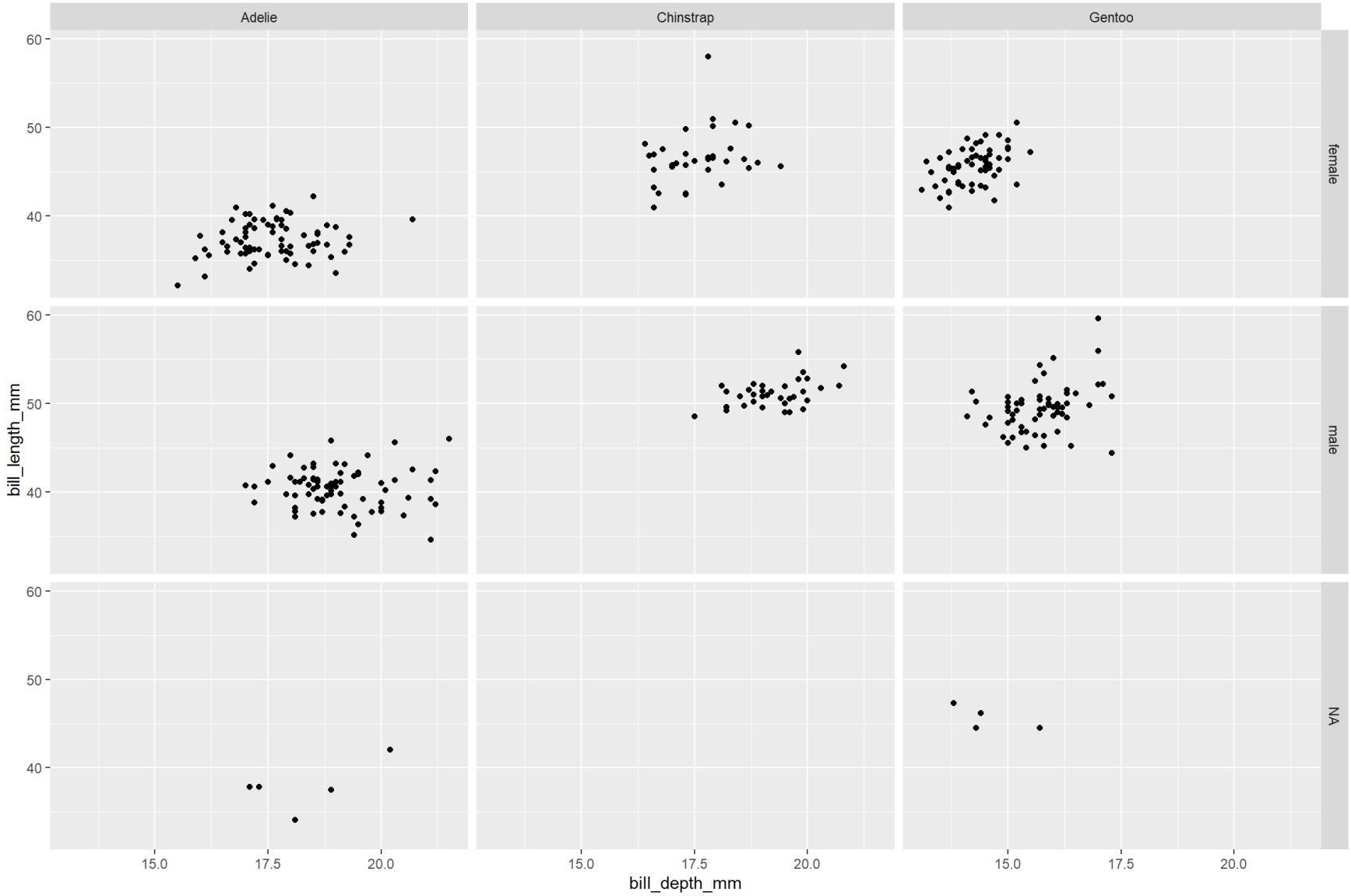


Décrivez ce qu’il se passe dans les graphes en fonction du code

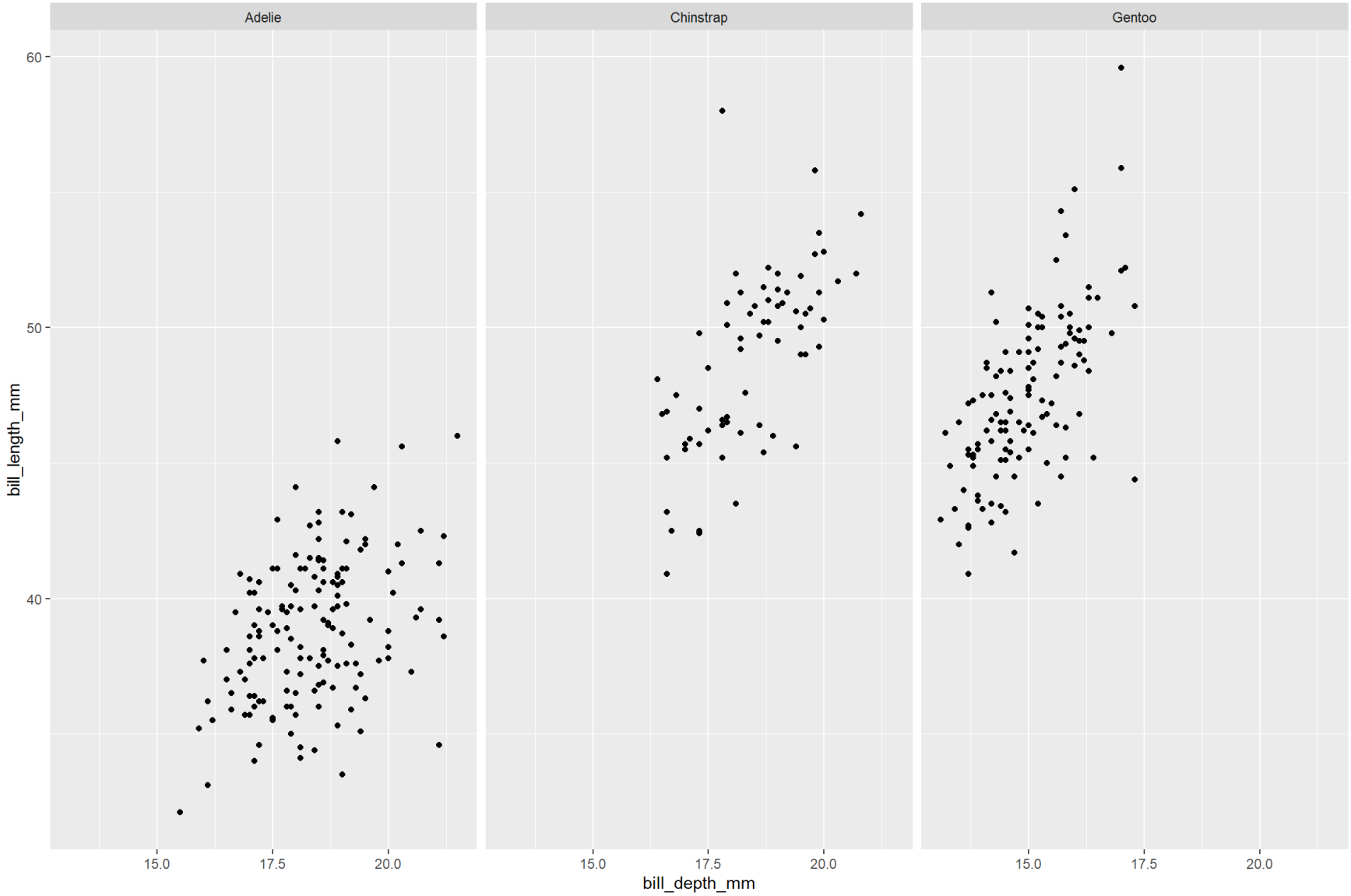
```r
ggplot(penguins, aes(x = bill_depth_mm, y = bill_length_mm)) +
  geom_point() +
  facet_grid(species ~ sex)
```



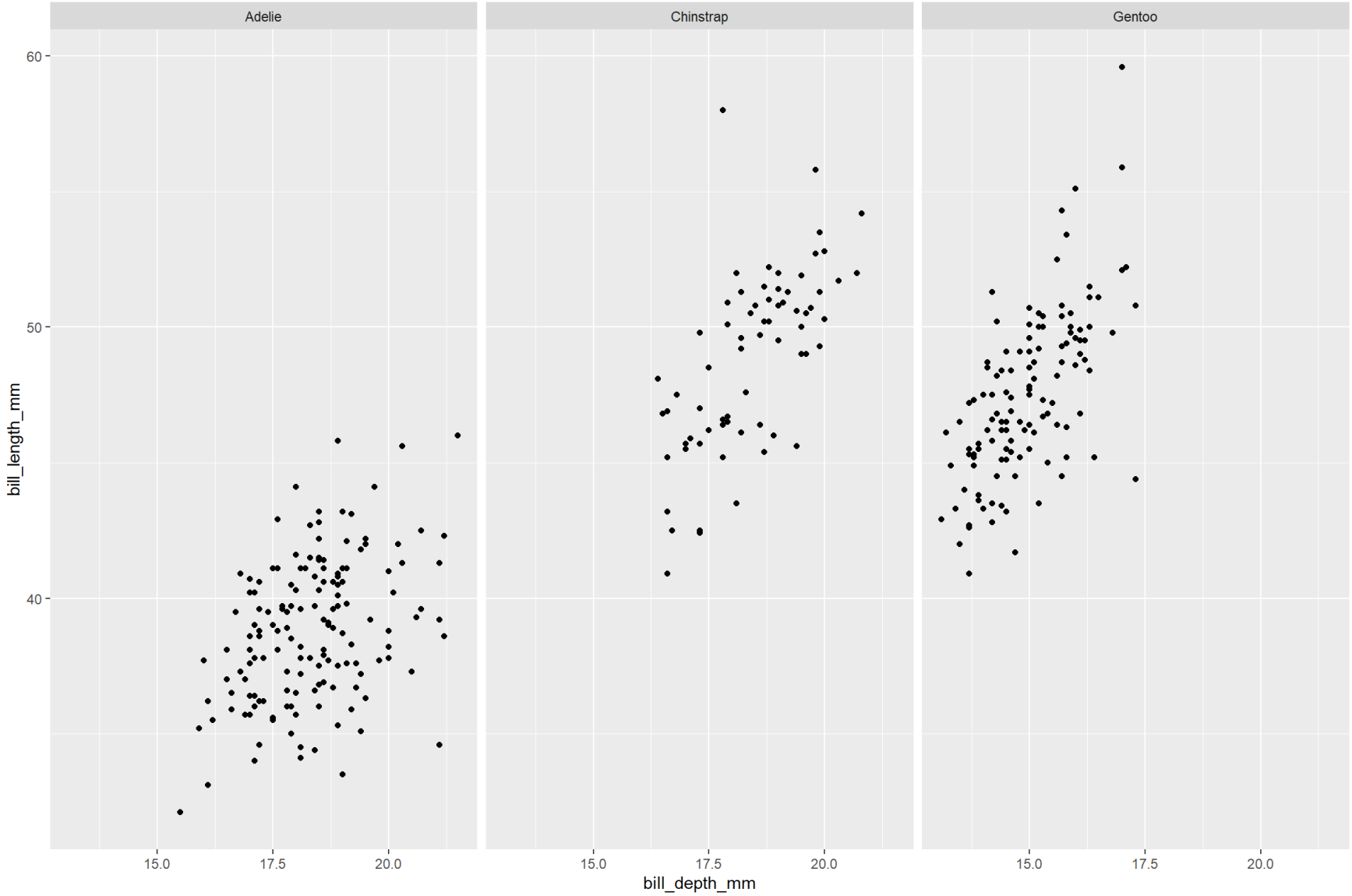
```r
ggplot(penguins, aes(x = bill_depth_mm, y = bill_length_mm)) +
  geom_point() +
  facet_grid(sex ~ species)
```



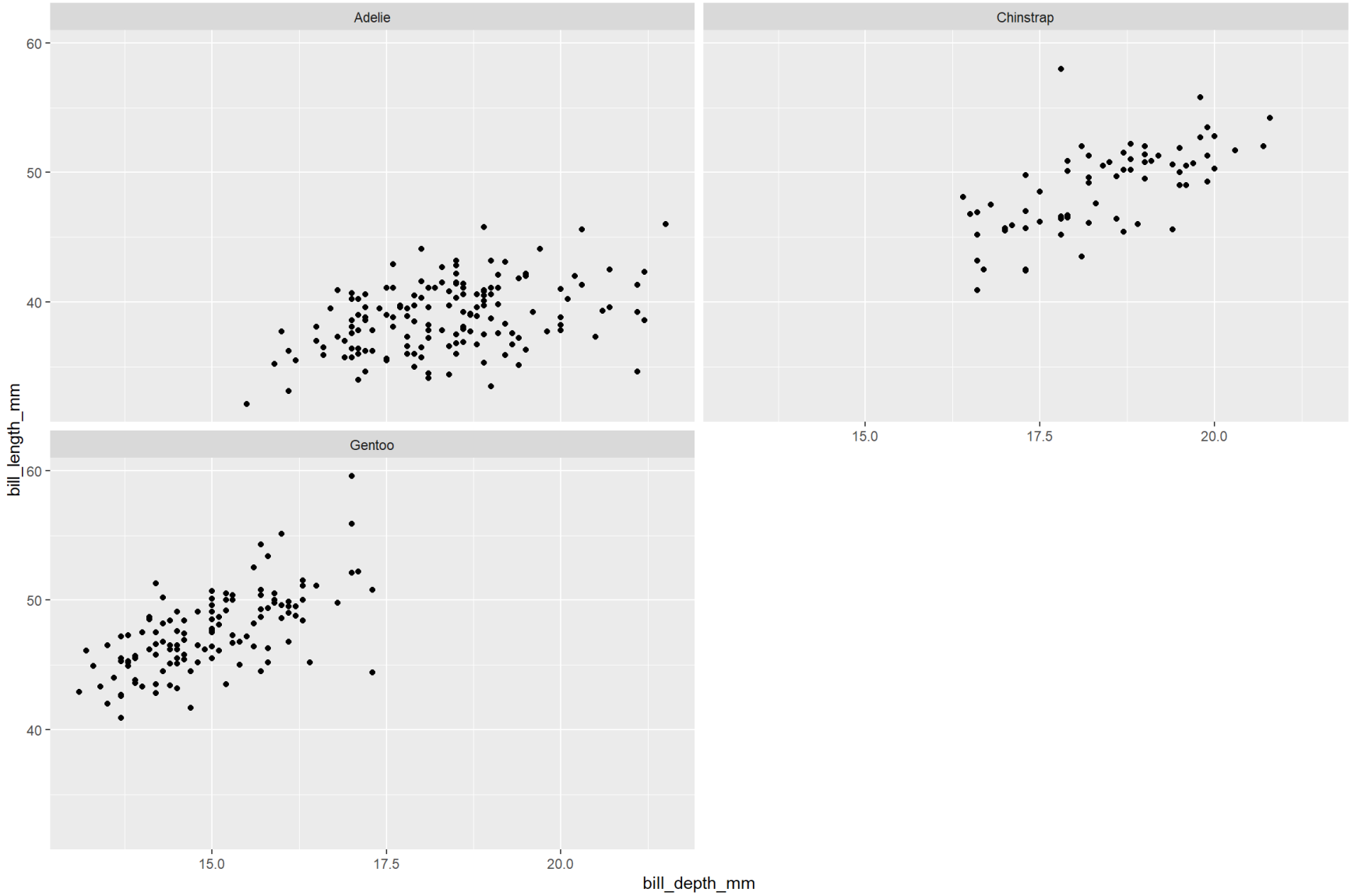
```r
ggplot(penguins, aes(x = bill_depth_mm, y = bill_length_mm)) +
  geom_point() +
  facet_wrap(~ species)
```



```r
ggplot(penguins, aes(x = bill_depth_mm, y = bill_length_mm)) +
  geom_point() +
  facet_grid(. ~ species)
```



```
1 ggplot(penguins, aes(x = bill_depth_mm, y = bill_length_mm)) +
2   geom_point() +
3   facet_wrap(~ species, ncol = 2)
```



Pour résumer

`facet_grid()`:

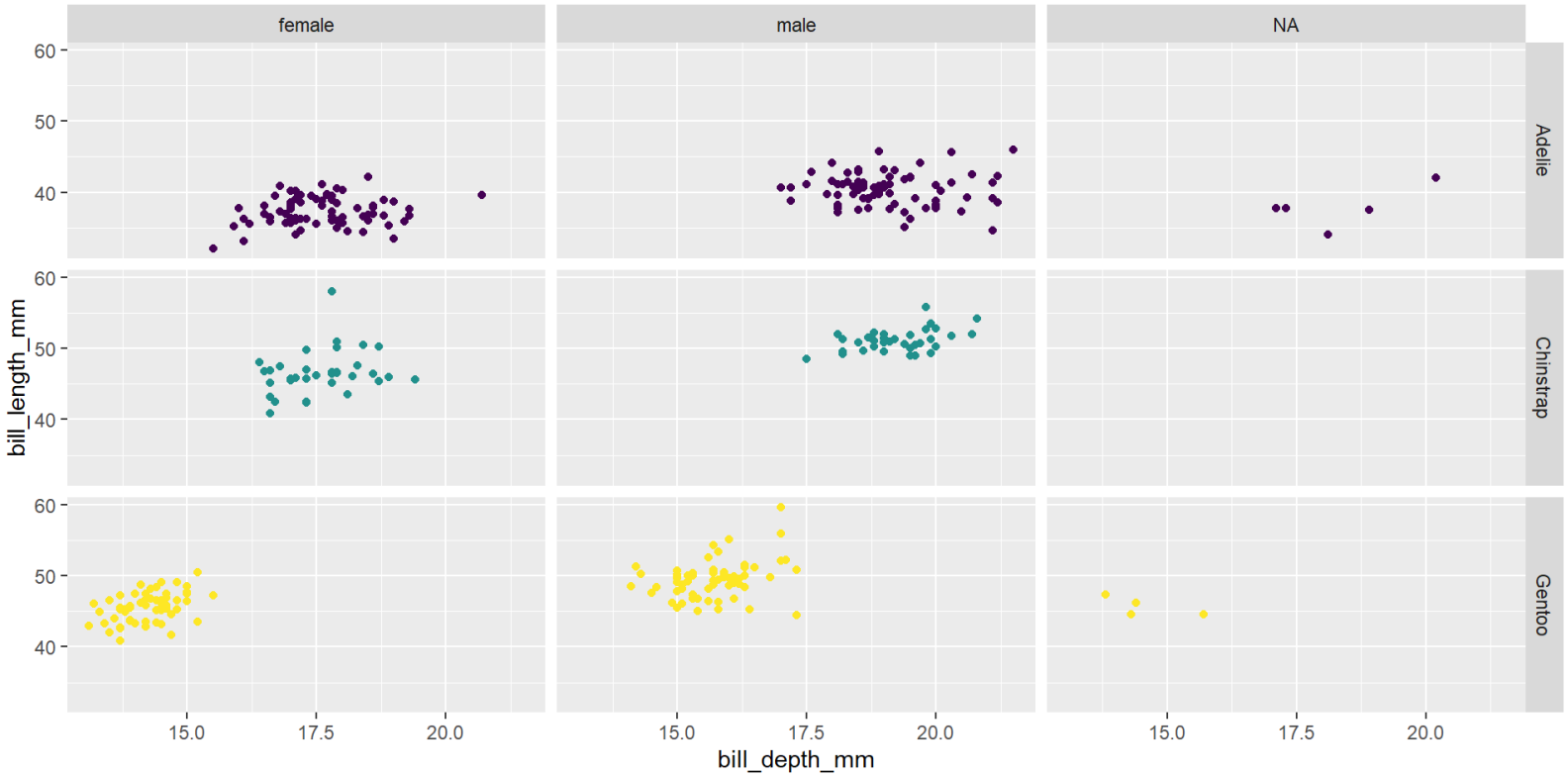
- Grille en 2D
- `lignes ~ colonnes`
- Utilisez le `.` pour faire du 1D

`facet_wrap()`:

- Ruban 1D
- Déroulé selon les arguments spécifiés ou la place à disposition

Combinaison avec la couleur

```
ggplot(
  penguins,
  aes(x = bill_depth_mm,
      y = bill_length_mm,
      color = species)) +
  geom_point() +
  facet_grid(species ~ sex) +
  scale_color_viridis_d() +
  guides(color = "none")
```



Visualisation des données

Terminologie

Analyse :

- `Univariée` - distribution d'une unique variable
- `Bivariée` - Relation entre deux variables
- `Multivariée` - Relation entre plusieurs variables, souvent en se concentrant sur la relation entre deux, tout en les conditionnant selon les autres.

Terminologie

Types de variables :

		Opérations possibles
Qualitative	Nominale	= ≠
	Ordinale	= ≠ < >
Quantitative	Intervalle	= ≠ < > + -
	Ratio	= ≠ < > + - · /

Données

Lending Club

Plateforme pour faire des prêts entre particuliers.

Le jeu de données contient les prêts effectués.

```
1 library(openintro)
2 glimpse(loans_full_schema)
```

```
Rows: 10,000
Columns: 55
$ emp_title                    <chr> "global config engineer ", "warehouse…
$ emp_length                   <dbl> 3, 10, 3, 1, 10, NA, 10, 10, 10, 3, 1…
$ state                        <fct> NJ, HI, WI, PA, CA, KY, MI, AZ, NV, I…
$ homeownership                <fct> MORTGAGE, RENT, RENT, RENT, RENT, OWN…
$ annual_income                <dbl> 90000, 40000, 40000, 30000, 35000, 34…
$ verified_income              <fct> Verified, Not Verified, Source Verifi…
$ debt_to_income               <dbl> 18.01, 5.04, 21.15, 10.16, 57.96, 6.4…
$ annual_income_joint          <dbl> NA, NA, NA, NA, 57000, NA, 155000, NA…
$ verification_income_joint    <fct> , , , , Verified, , Not Verified, , ,…
$ debt_to_income_joint         <dbl> NA, NA, NA, NA, 37.66, NA, 13.12, NA
```

Sélection de variables

```r
loans <- loans_full_schema %>%
  select(loan_amount, interest_rate, term, grade,
         state, annual_income, homeownership, debt_to_income)
glimpse(loans)
```

```
Rows: 10,000
Columns: 8
$ loan_amount    <int> 28000, 5000, 2000, 21600, 23000, 5000, 24000, 20000, 20…
$ interest_rate  <dbl> 14.07, 12.61, 17.09, 6.72, 14.07, 6.72, 13.59, 11.99, 1…
$ term           <dbl> 60, 36, 36, 36, 36, 36, 60, 60, 36, 36, 60, 60, 36, 60,…
$ grade          <fct> C, C, D, A, C, A, C, B, C, A, C, B, C, B, D, D, D, F, E…
$ state          <fct> NJ, HI, WI, PA, CA, KY, MI, AZ, NV, IL, IL, FL, SC, CO,…
$ annual_income  <dbl> 90000, 40000, 40000, 30000, 35000, 34000, 35000, 110000…
$ homeownership  <fct> MORTGAGE, RENT, RENT, RENT, RENT, OWN, MORTGAGE, MORTGA…
$ debt_to_income <dbl> 18.01, 5.04, 21.15, 10.16, 57.96, 6.46, 23.66, 16.19, 3…
```

Variables sélectionnées

Variable	Description
`loan_amount`	Montant du prêt reçu en US Dollards
`interest_rate`	Intérêt sur le prêt, en pourcentage annuel
`term`	Durée du prêt en mois
`grade`	Note du prêt, de A à G, qui représente la qualité du prêt et les changes qu'ils soit remboursé
`state`	État américain dans lequel le prêt a été accordé
`annual_income`	Revenu annuel dudébiteur, en US Dollards
`homeownership`	Indique si la personne est propriétaire, est propriétaire avec un emprunt ou bien loue sa résidence
`debt_to_income`	Ratio Dette/Revenu

Types des variables

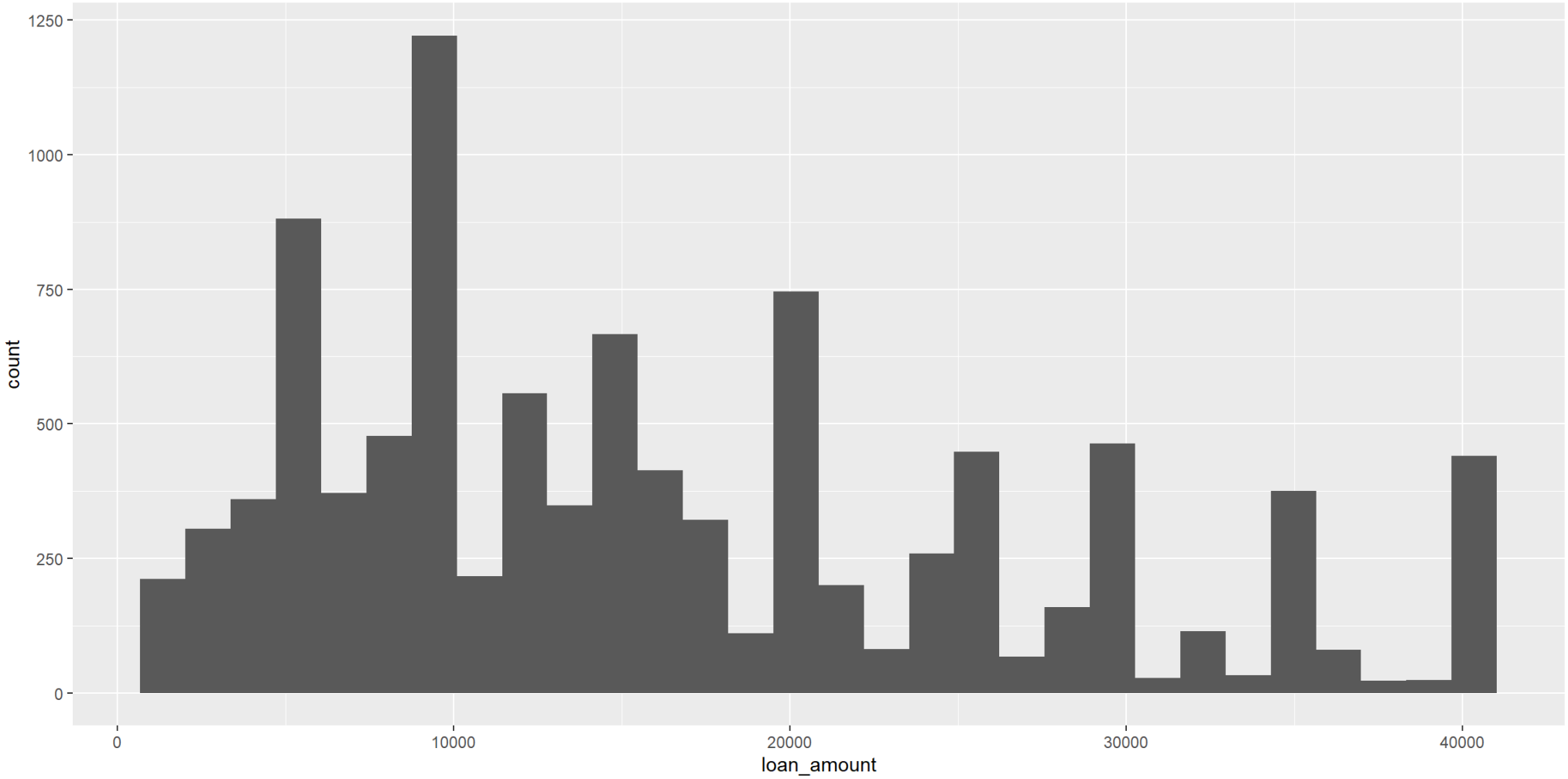
variable	type
`loan_amount`	Quantitatif, Ratio
`interest_rate`	Quantitatif, Ratio
`term`	Quantitatif, Ratio
`grade`	Qualitatif, Ordinal
`state`	Qualitatif, Nominal
`annual_income`	Quantitatif, Ratio
`homeownership`	Qualitatif, Nominal
`debt_to_income`	Quantitatif, Ratio

Données quantitatives

Histogramme

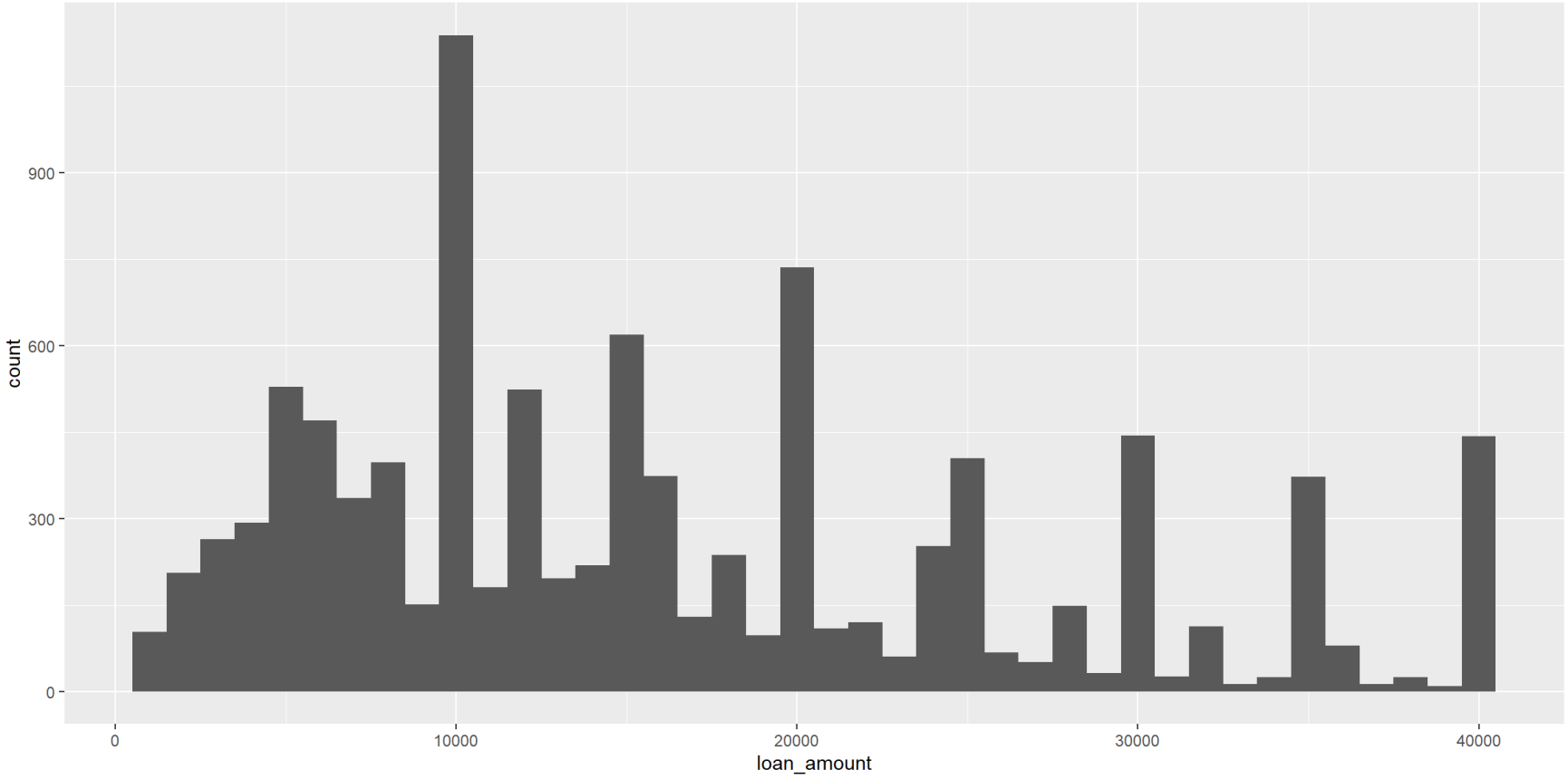
Histogramme

```
1  ggplot(loans, aes(x = loan_amount)) +
2    geom_histogram()
```



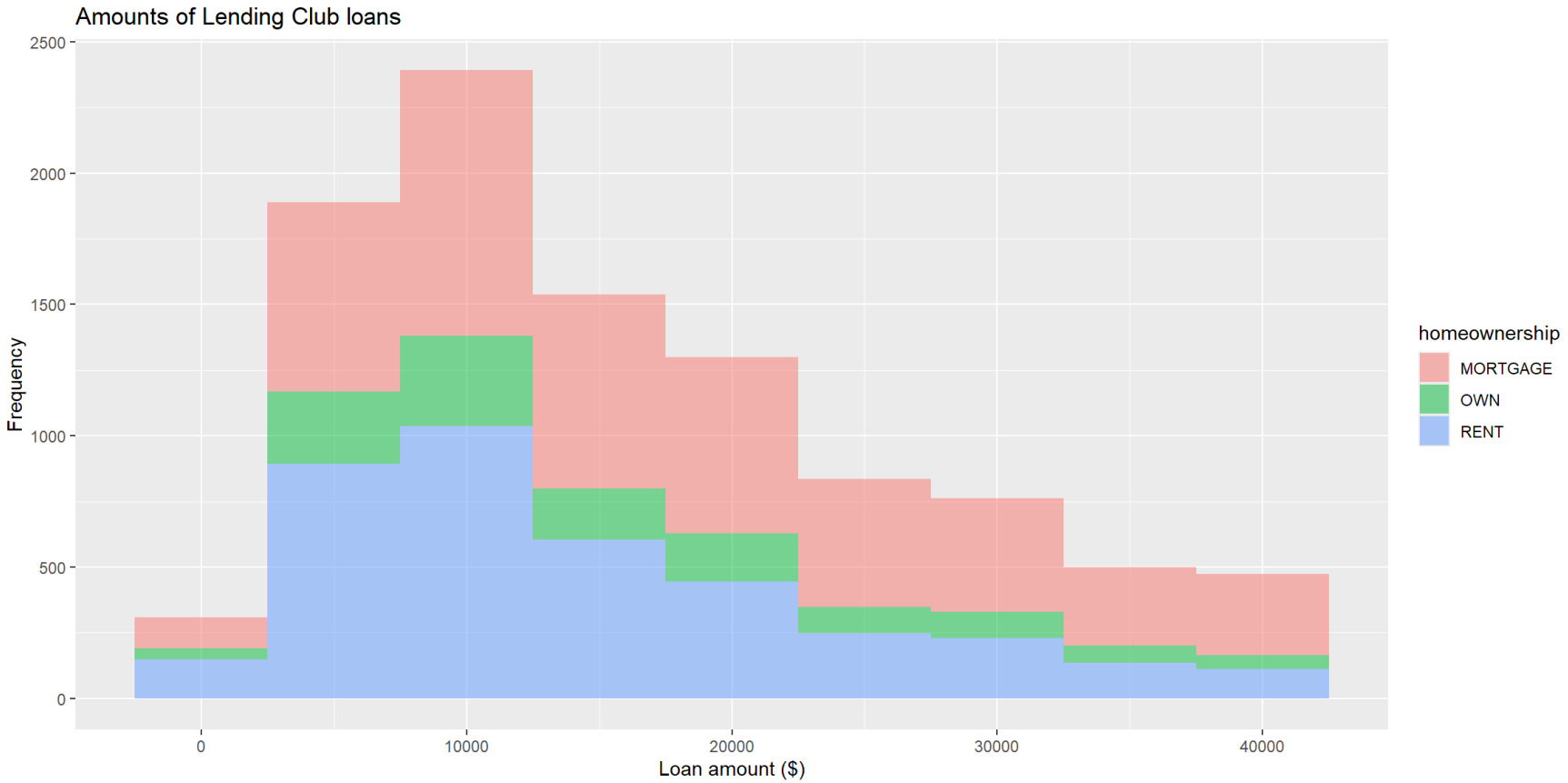
binwidth = 1000 binwidth = 5000 binwidth = 20000

```
1 ggplot(loans, aes(x = loan_amount)) +
2   geom_histogram(binwidth = 1000)
```



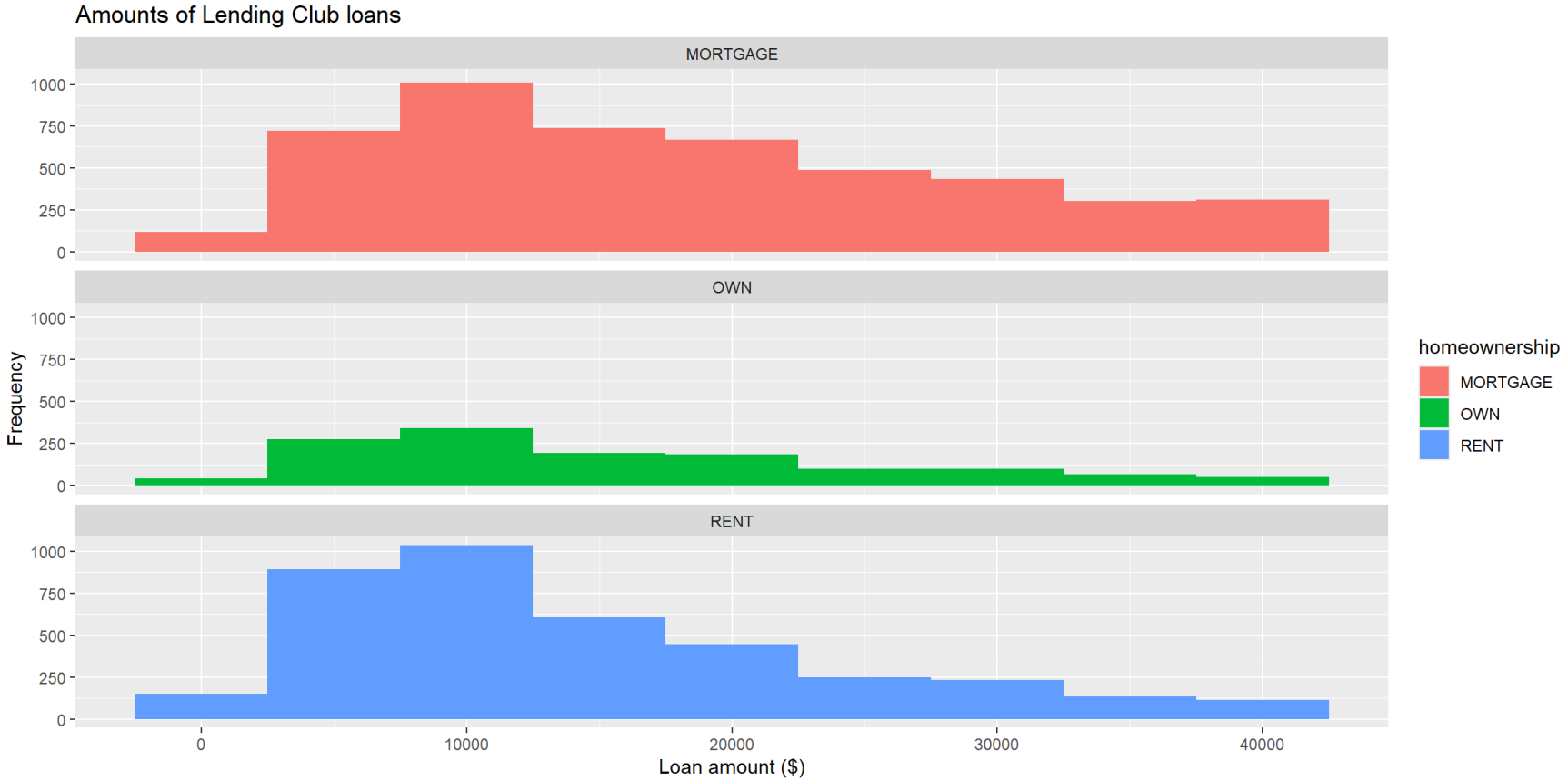
Combinaison avec des variables qualitatives

Plot Code



Avec des facettes

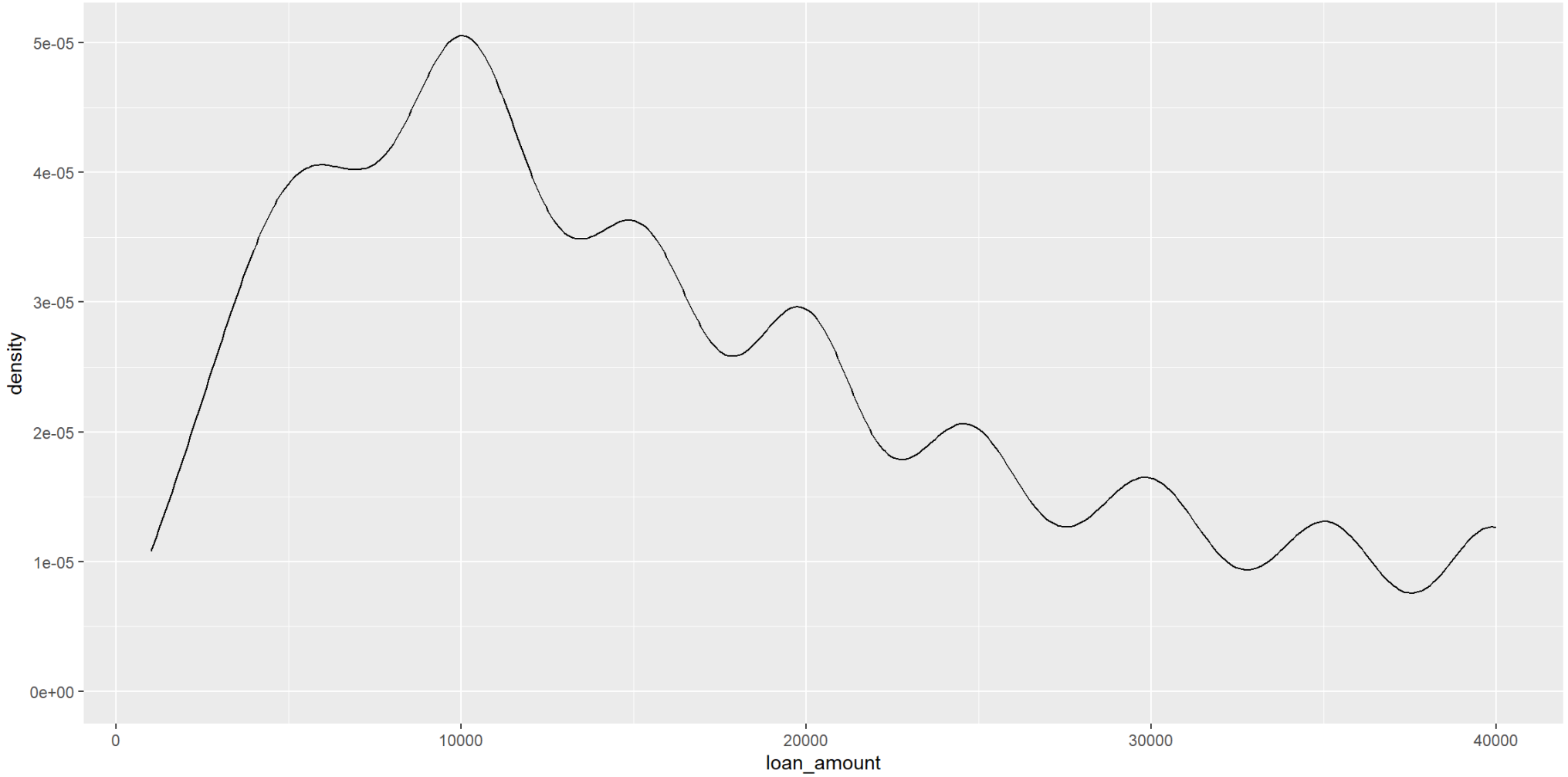
Plot Code



Graphe de densité

Density plot

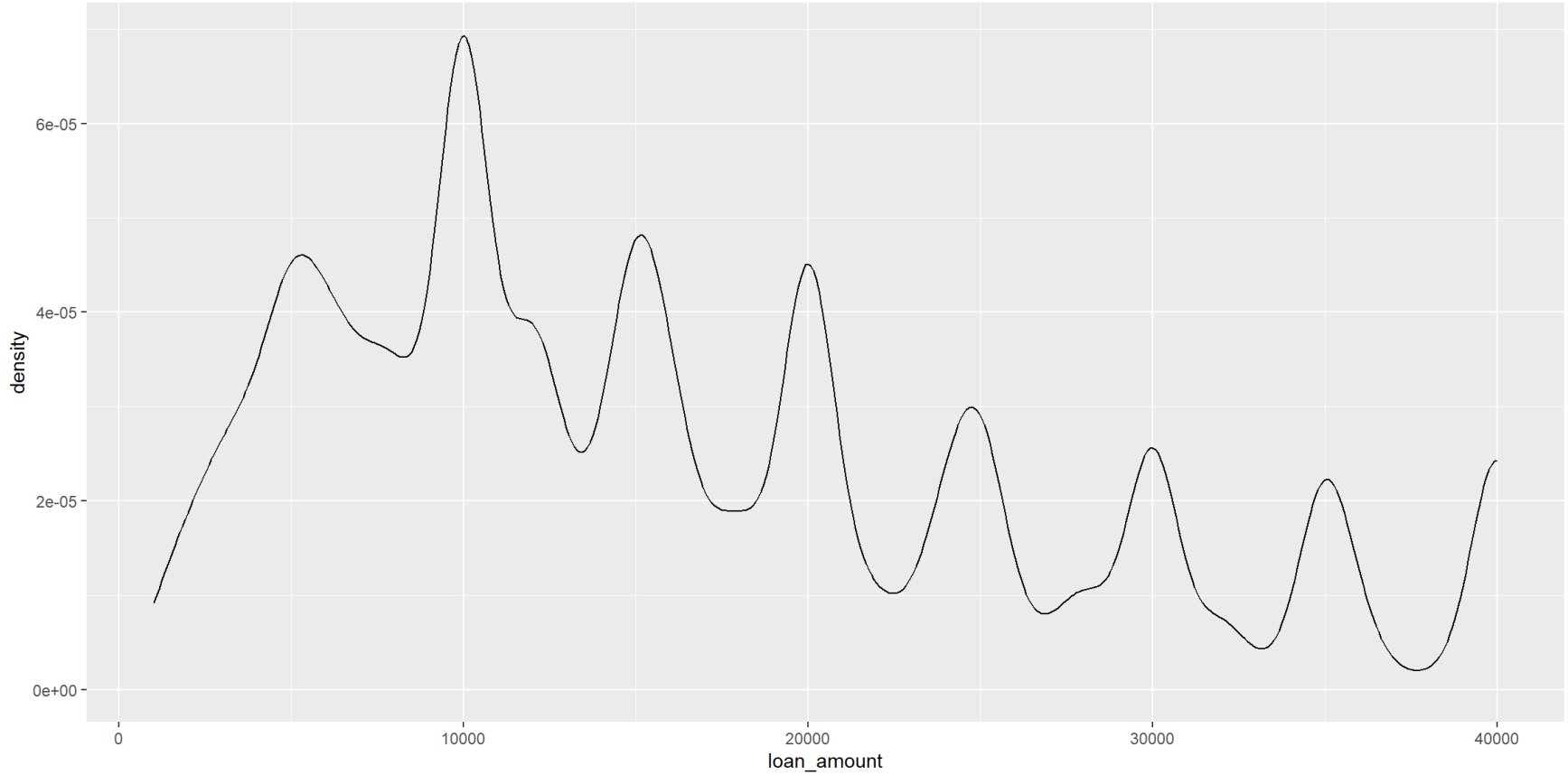
```
1 ggplot(loans, aes(x = loan_amount)) +
2   geom_density()
```



Ajustement de la précision

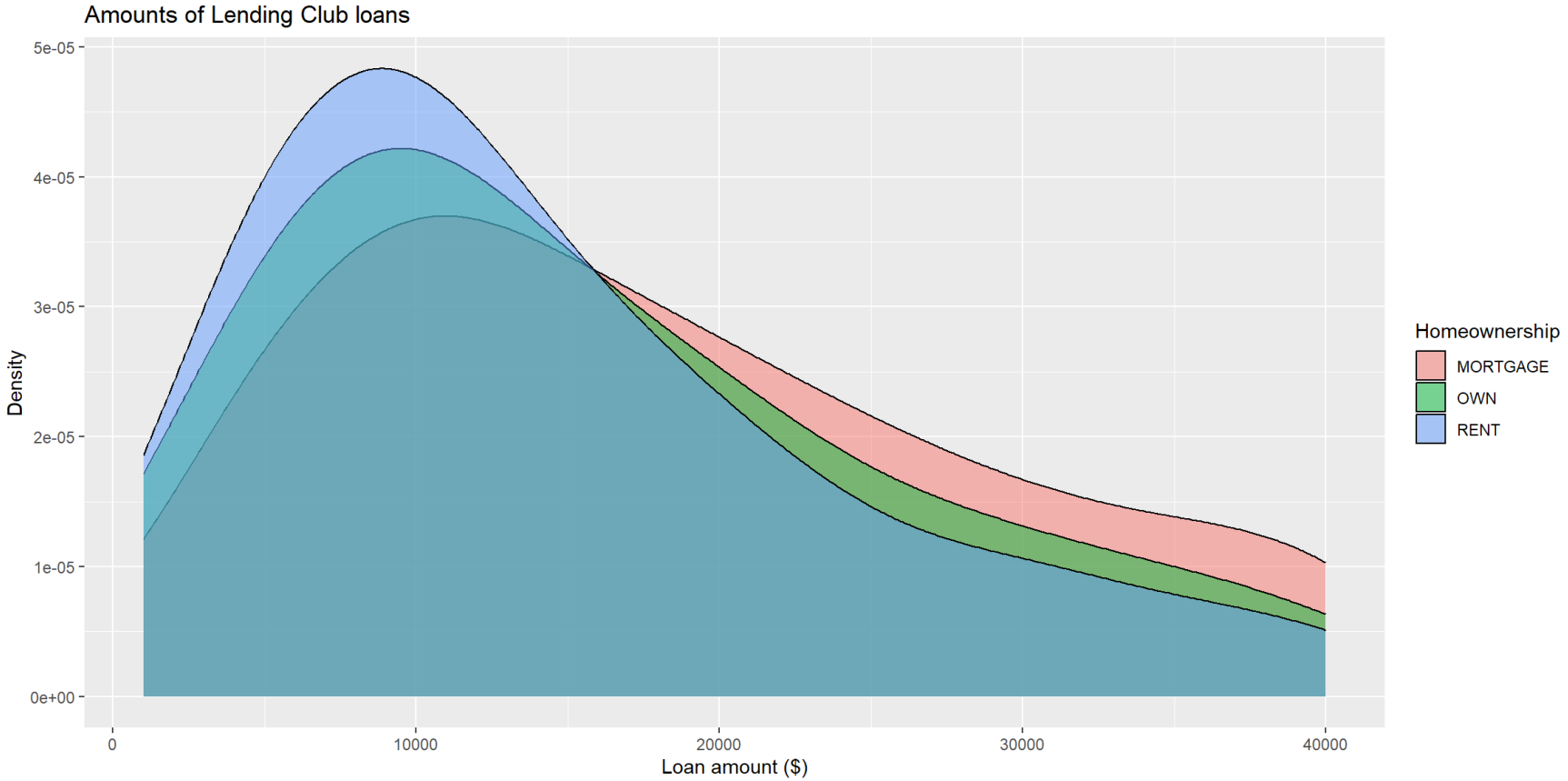
adjust = 0.5 adjust = 1 adjust = 2

```
1 ggplot(loans, aes(x = loan_amount)) +
2   geom_density(adjust = 0.5)
```



Combinaison avec des variables qualitatives

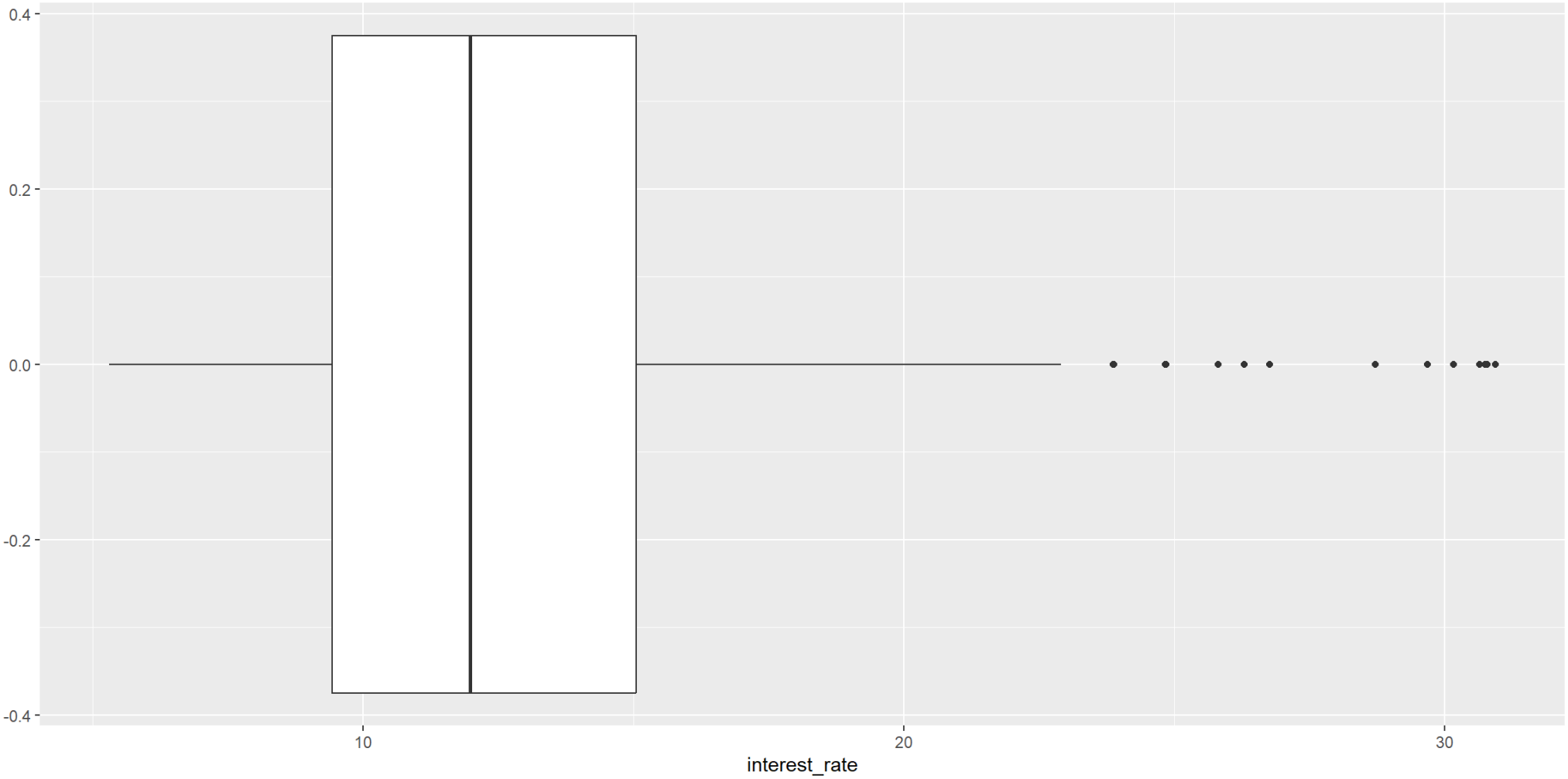
Plot Code



Box plot

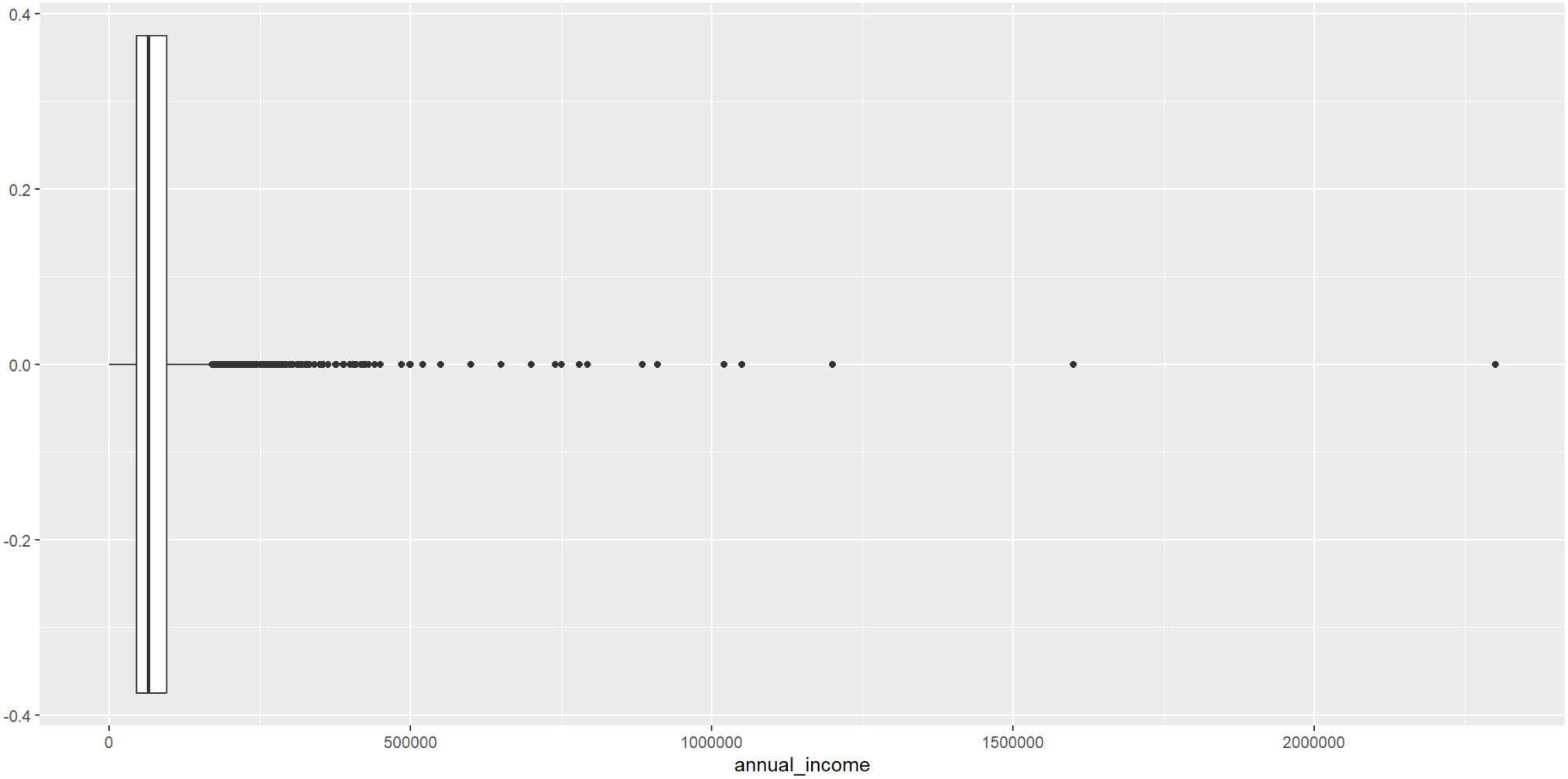
Boîte à moustache

```
1 ggplot(loans, aes(x = interest_rate)) +
2   geom_boxplot()
```



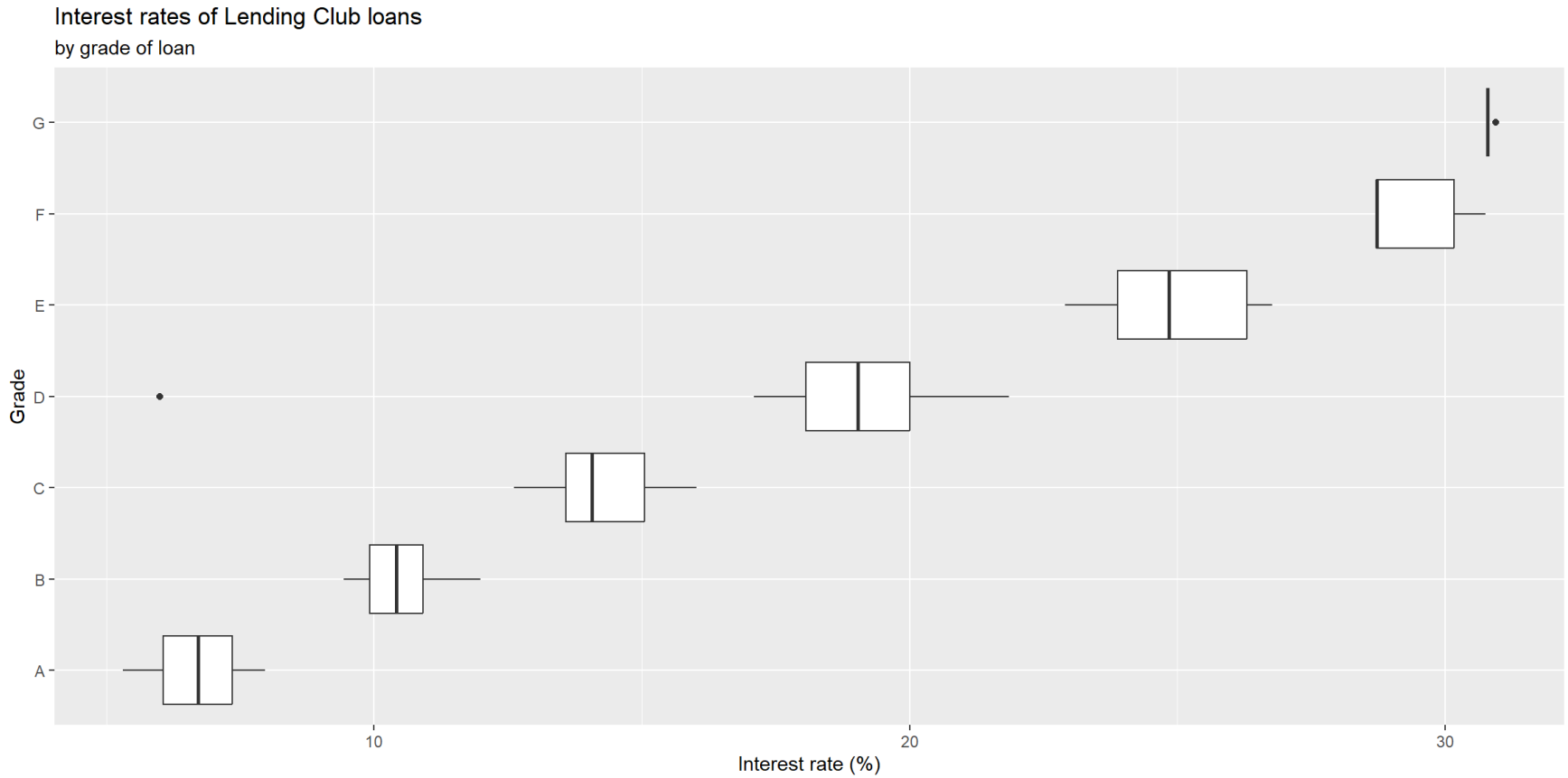
Box plot et outliers

```
1 ggplot(loans, aes(x = annual_income)) +
2   geom_boxplot()
```



Combinaison avec des variables qualitatives

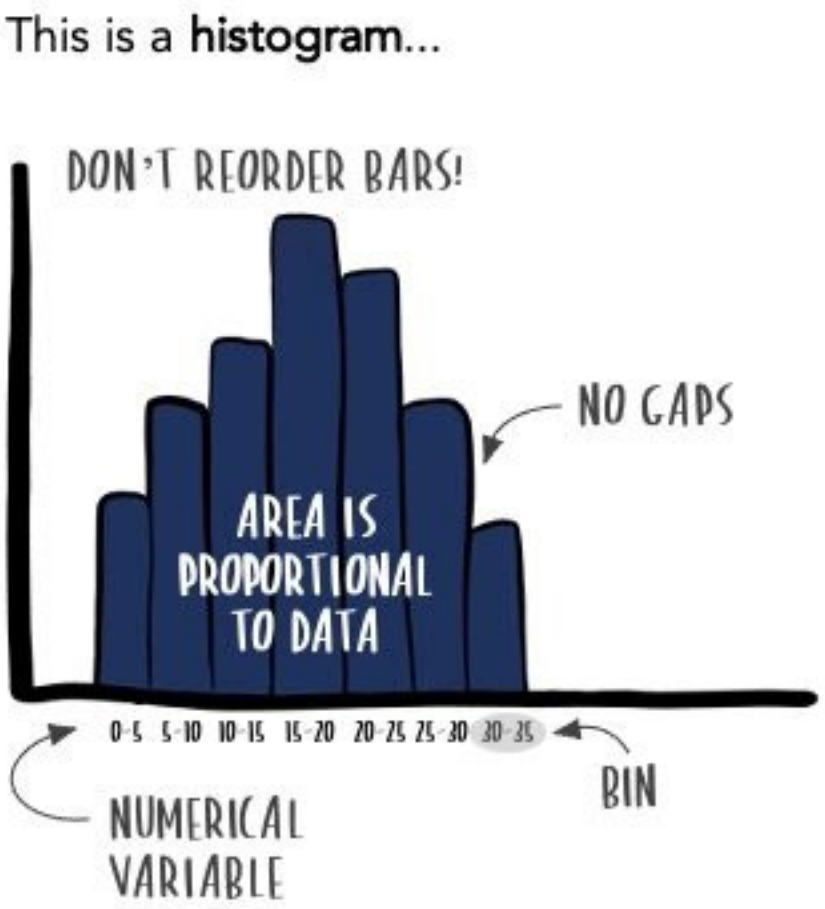
Plot Code

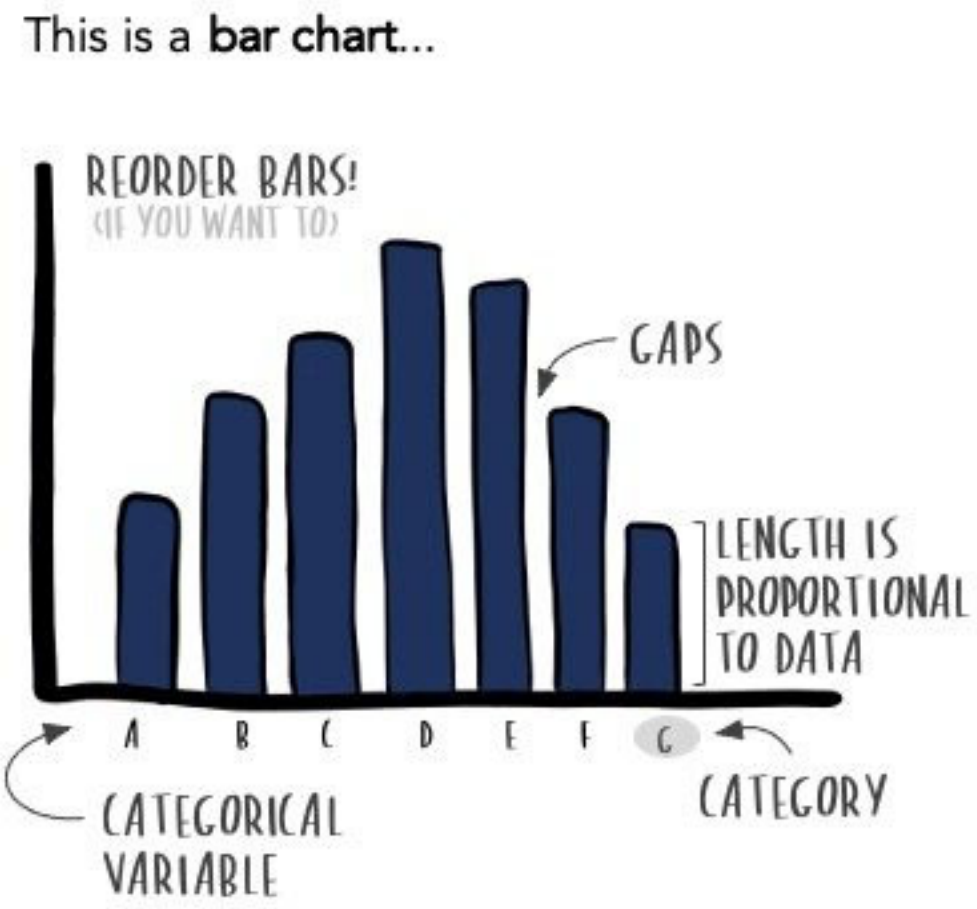


Données qualitatives

Graphes en bar

Histogramme vs bar charts

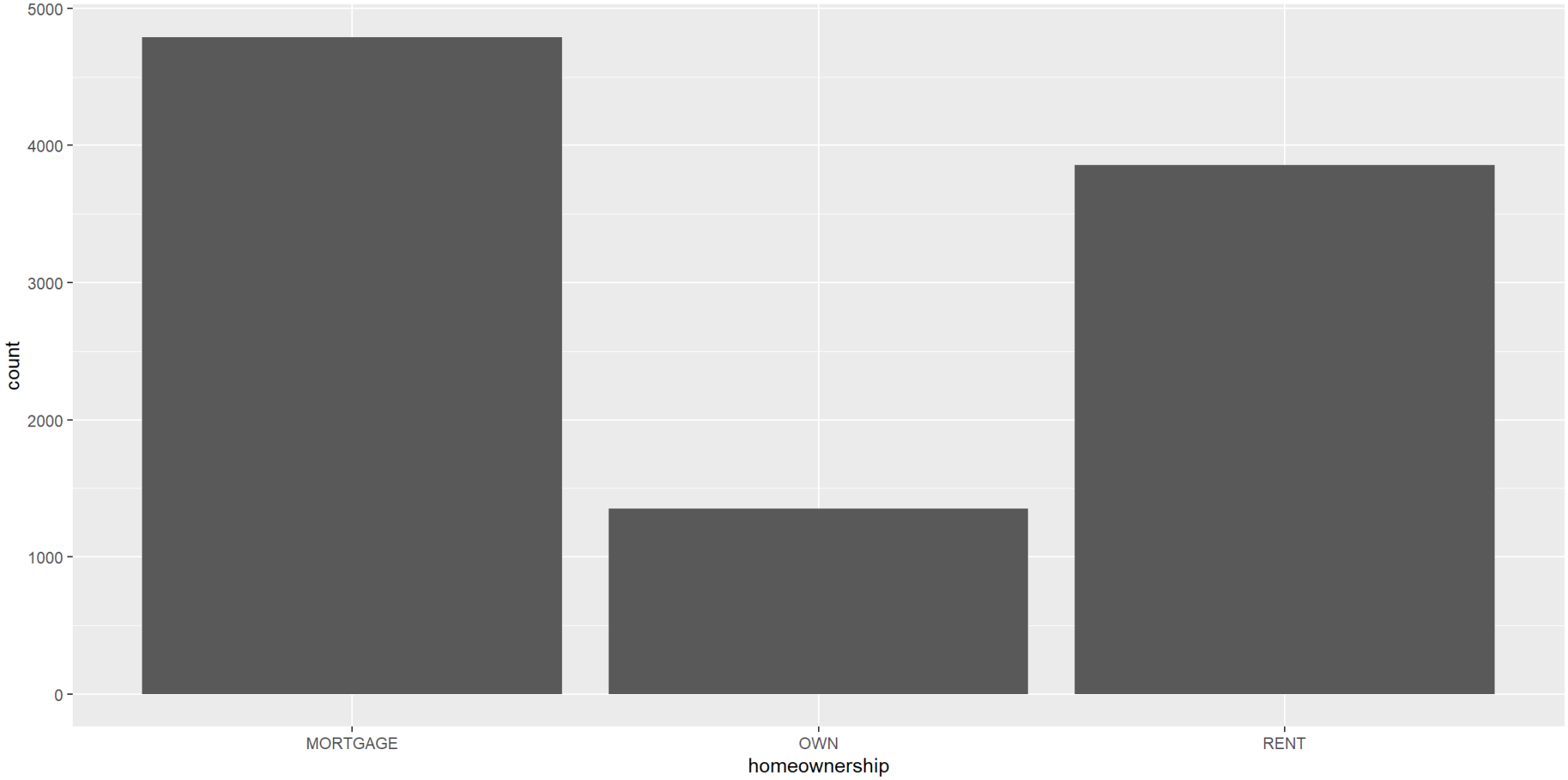




Source

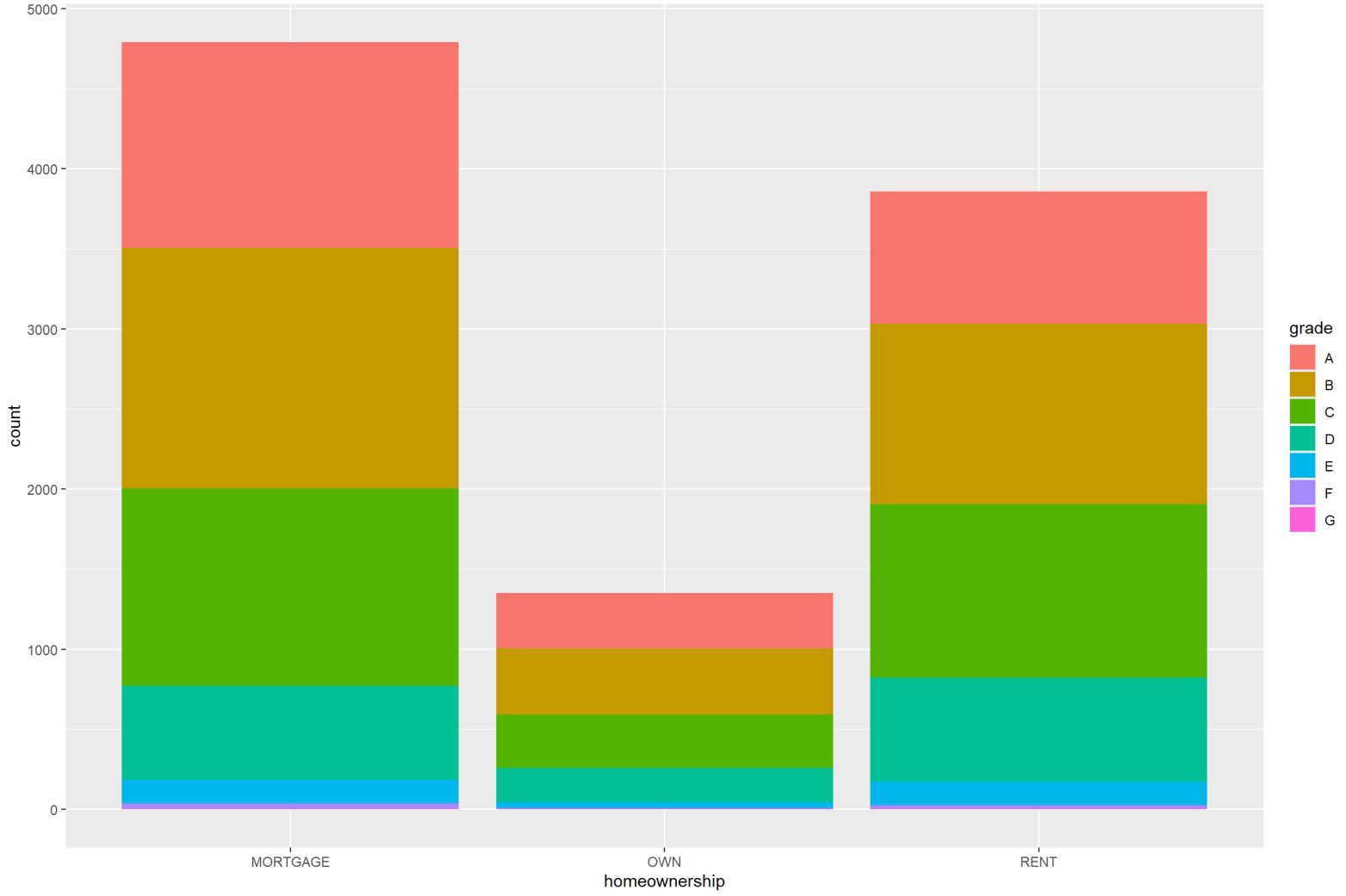
Bar chart

```
1 ggplot(loans, aes(x = homeownership)) +
2   geom_bar()
```



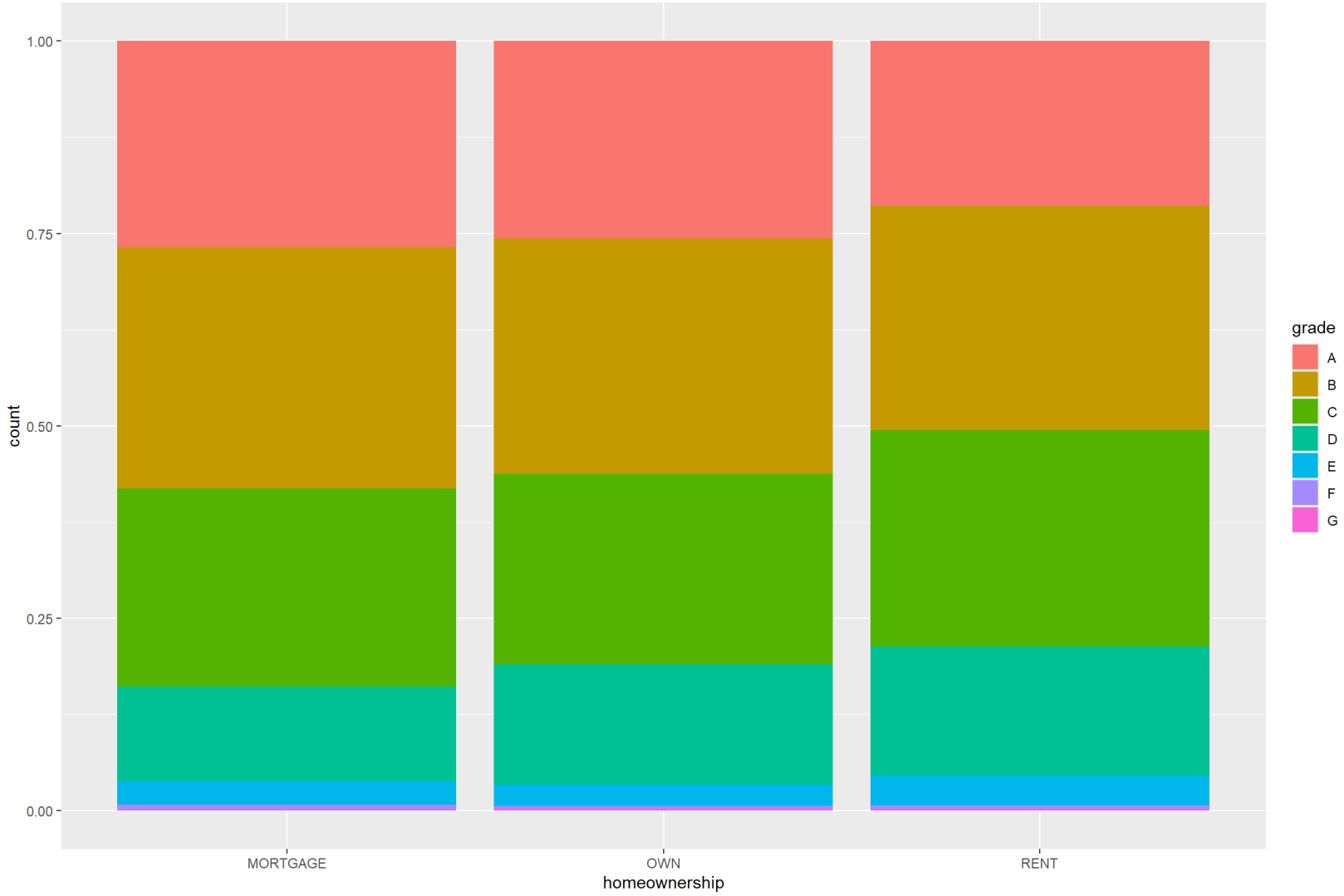
Bar chart segmenté

```
1  ggplot(loans, aes(x = homeownership,
2                    fill = grade)) +
3    geom_bar()
```

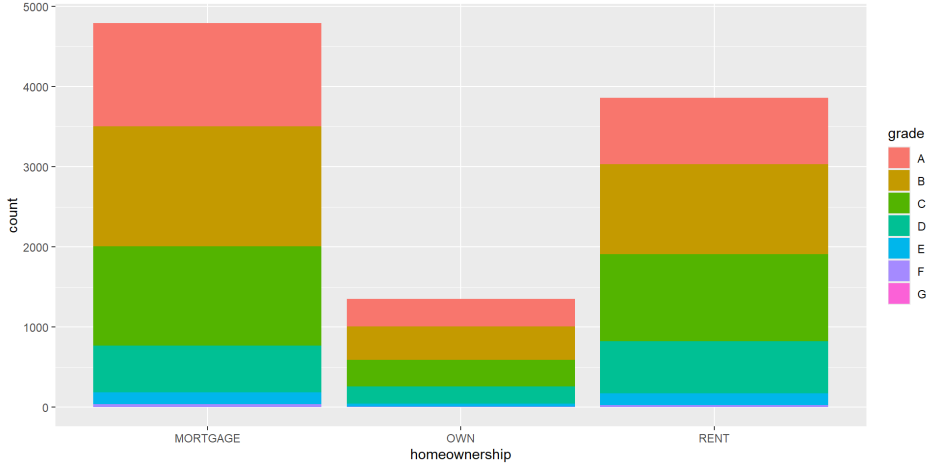


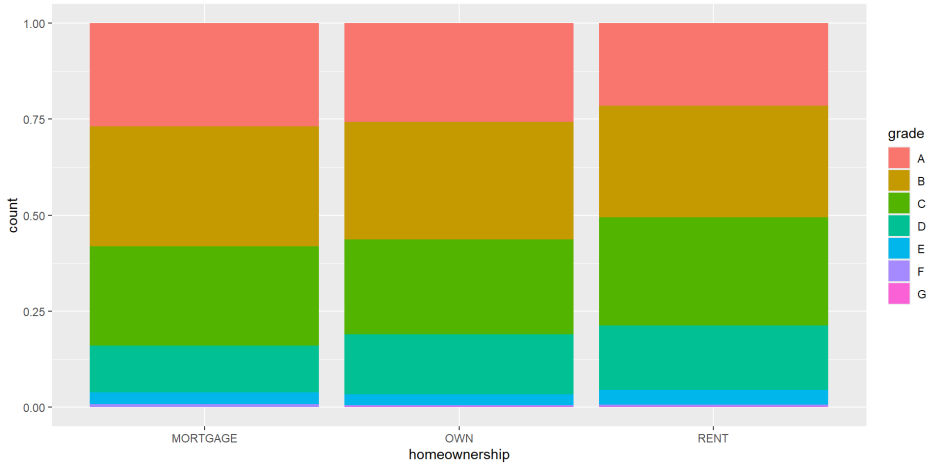
Bar chart segmenté

```
1  ggplot(loans, aes(x = homeownership, fill = grade)) +
2    geom_bar(position = "fill")
```



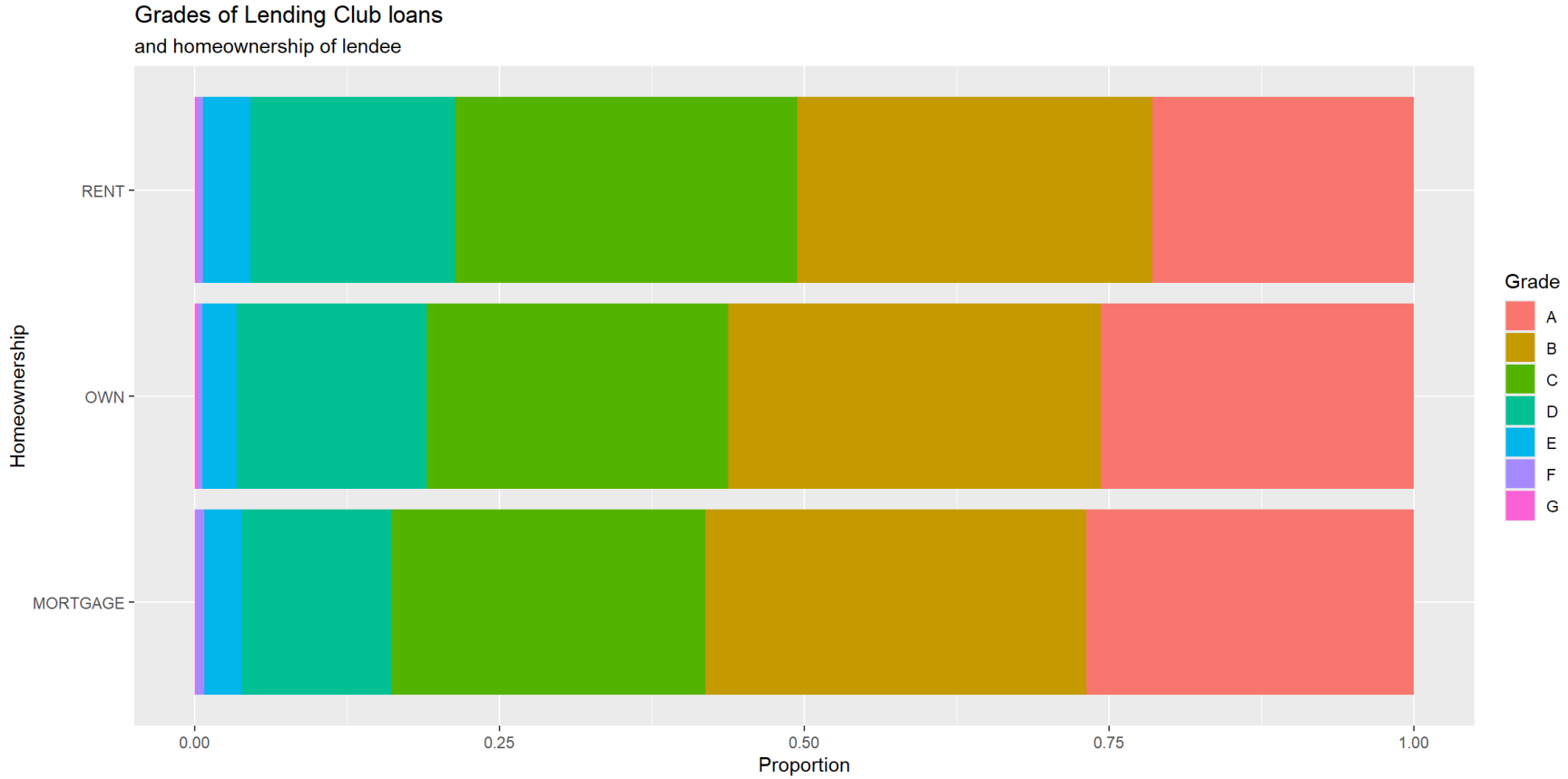
Lequel des deux graphes est plus adapté pour montrer la relation entre le fait d'être propriétaire et les notes de prêt ?





Bar plot horizontal

Plot Code



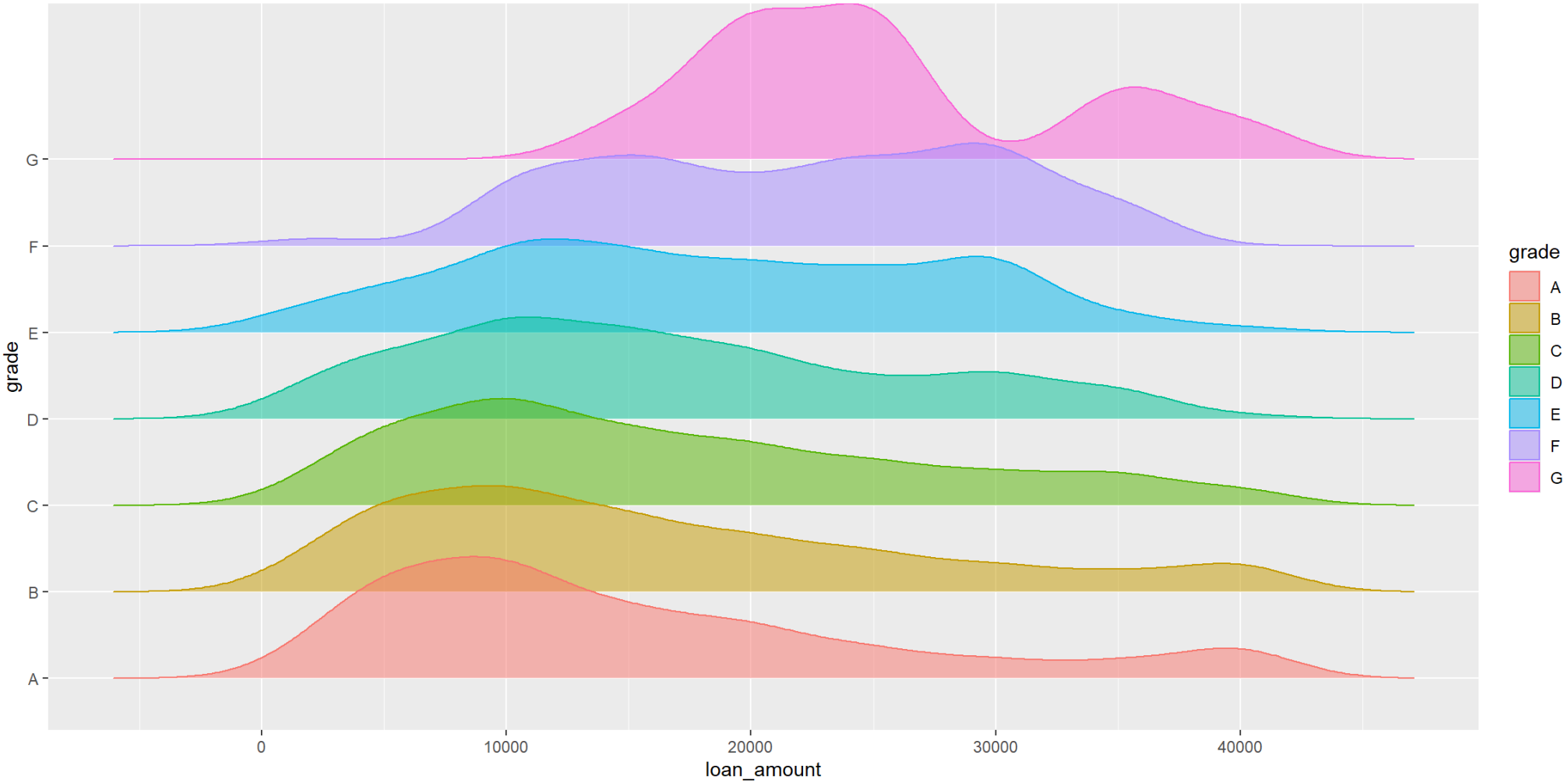
Qualitative vs Quantitative

Violin plots

```
1 ggplot(loans, aes(x = homeownership, y = loan_amount)) +
2   geom_violin()
```

Ridge plots

```
1 library(ggridges)
2 ggplot(loans, aes(x = loan_amount, y = grade, fill = grade, color = grade)) +
3   geom_density_ridges(alpha = 0.5)
```



Références

Wilkinson, L. (2005). *The Grammar of Graphics*. Springer-Verlag. https://doi.org/10.1007/0-387-28695-0