

Note book GGplot_2

Utilisation de GGplot_2

GGplot2 est une extension du tidyverse qui permet de générer des graphiques avec une syntaxe cohérente et puissante. Elle nécessite l'apprentissage d'un mini-langage supplémentaire, mais permet la construction de graphiques complexes de manière efficace.

Une des particularités de GGplot2 est que l'extension part du principe que les données relatives à un graphique sont stockées dans un tableau de données (data frame, tibble ou autre).

Les prérequis :

Il faut tout d'abord avoir un data frame du type

```
library(ggplot2)
head(mpg)
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year  cyl trans      drv   cty   hwy fl  class
##   <chr>          <chr> <dbl> <int> <int> <chr>   <chr> <int> <int> <chr> <chr>
## 1 audi          a4      1.8  1999    4 auto(l5)  f     18    29 p  compa~
## 2 audi          a4      1.8  1999    4 manual(m5) f     21    29 p  compa~
## 3 audi          a4      2    2008    4 manual(m6) f     20    31 p  compa~
## 4 audi          a4      2    2008    4 auto(av)   f     21    30 p  compa~
## 5 audi          a4      2.8  1999    6 auto(l5)  f     16    26 p  compa~
## 6 audi          a4      2.8  1999    6 manual(m5) f     18    26 p  compa~
```

Une fois le tableau initialisé, il est temps de jouer avec les données. La manipulation passe par deux axes clés :

- la définition des variables sur leur axe respectifs
- le type de graphique à utiliser.

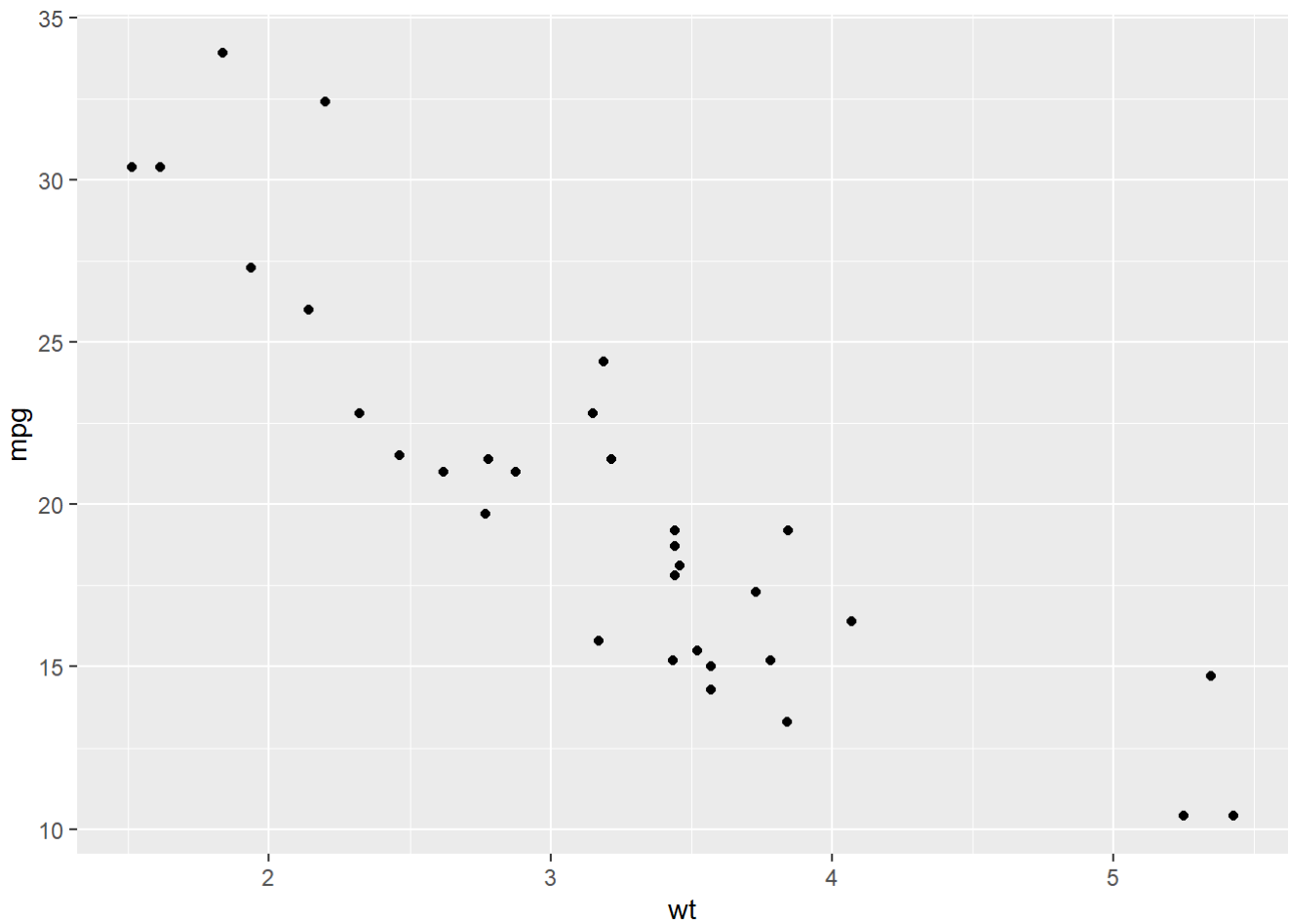
Il existe une multitude de graphiques qui ont chacun une fonction bien précise.

Le nuage de point

Definition: en statistiques, un nuage de points est une représentation de données dépendant de plusieurs variables. Il permet de mettre en évidence le degré de corrélation entre au moins deux variables liées. Les différentes observations des nuages de points permettent de déterminer : Des tendances Des dépendances. Utilisable avec la fonction : `geom_point` (qui permet de modifier : taille, couleur et forme des points).

Une fois l'initialisation terminée, il faut utiliser la fonction suivante :

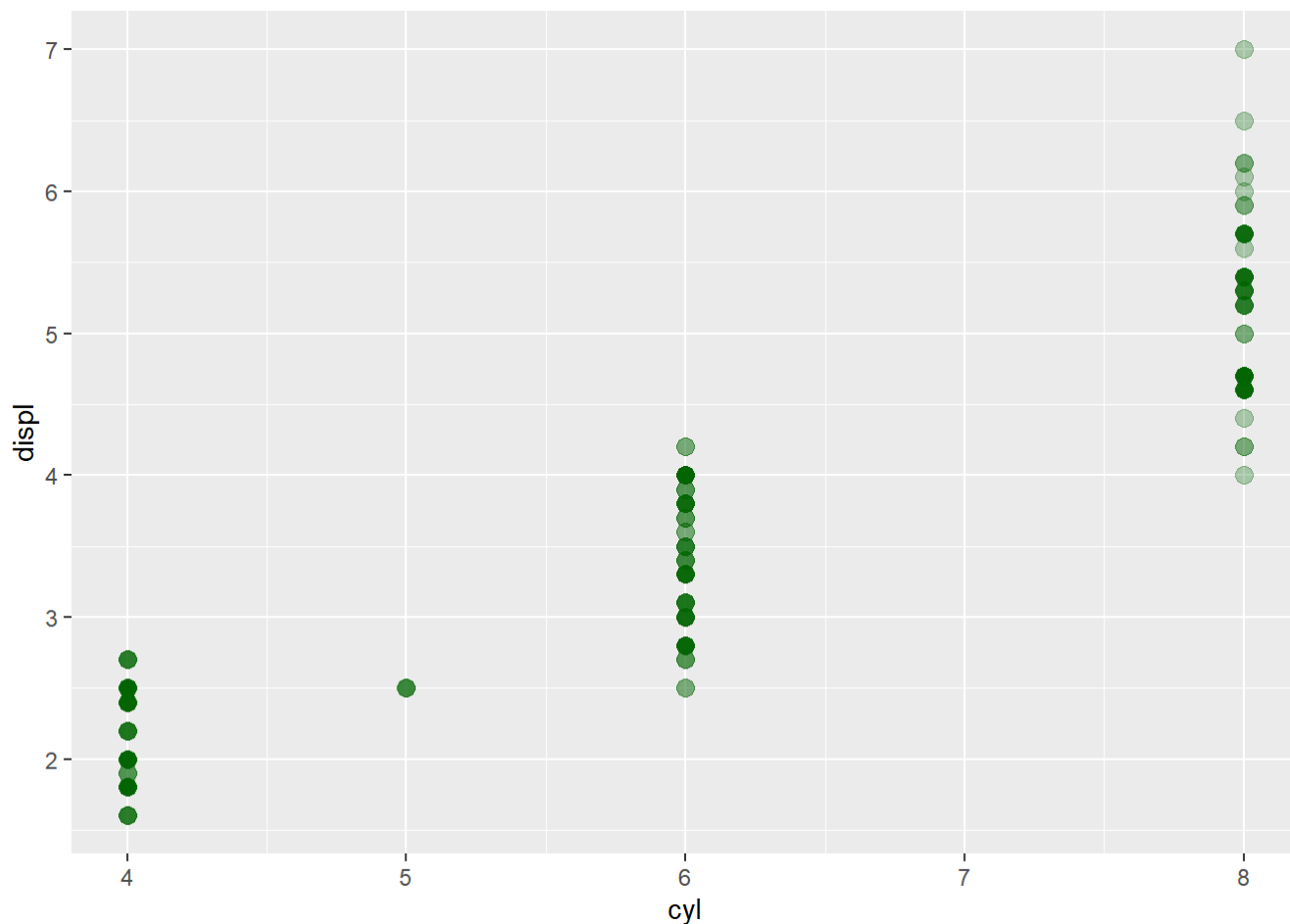
```
ggplot(mtcars, aes(x=wt, y=mpg))+ geom_point()
```



```
#(definition du DataFrame, definition des axes : aes (x = , y = )+ geom_point()
```

Pour la question des points, tout est dans la definition du “geom_point”, personnalisable à l'infini

```
ggplot(mpg) +  
  geom_point(aes(x = cyl, y = displ),  
             color = "darkgreen", size = 3, alpha = 0.3)
```

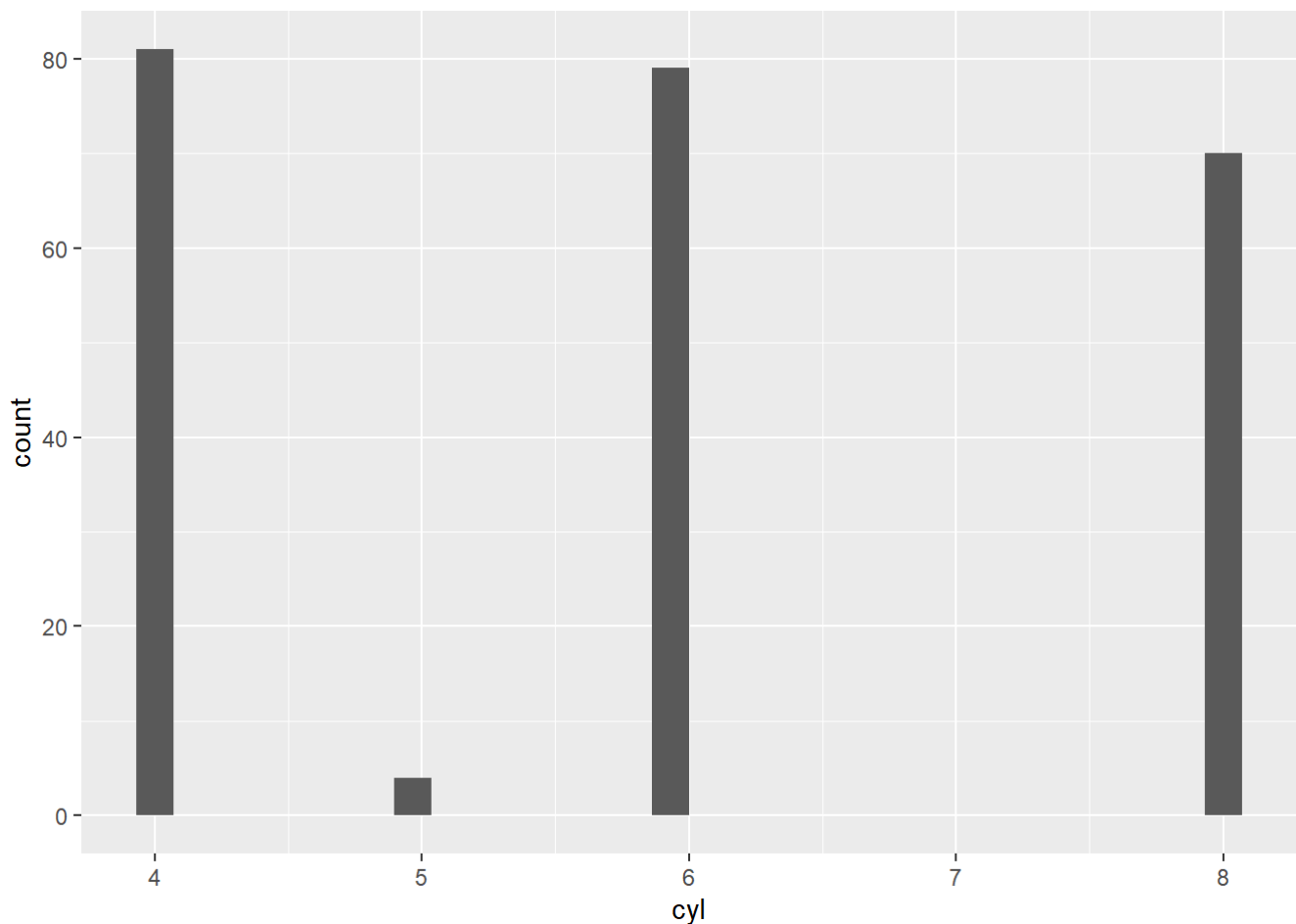


Les Histogrammes:

Definition : Représentation graphique des fréquences ou effectifs relatifs à un caractère quantitatif continu à l'aide d'une série de rectangles dont la base constitue un intervalle de variation des valeurs du caractère et la surface l'effectif correspondant. Pour cela, il faut utiliser la commande : `geom_histogram` qui se configure de la manière suivante :

```
ggplot(mpg) + geom_histogram(aes(x = cyl))
```

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



Définir X permet d'afficher la donnée que l'on souhaite représenter.

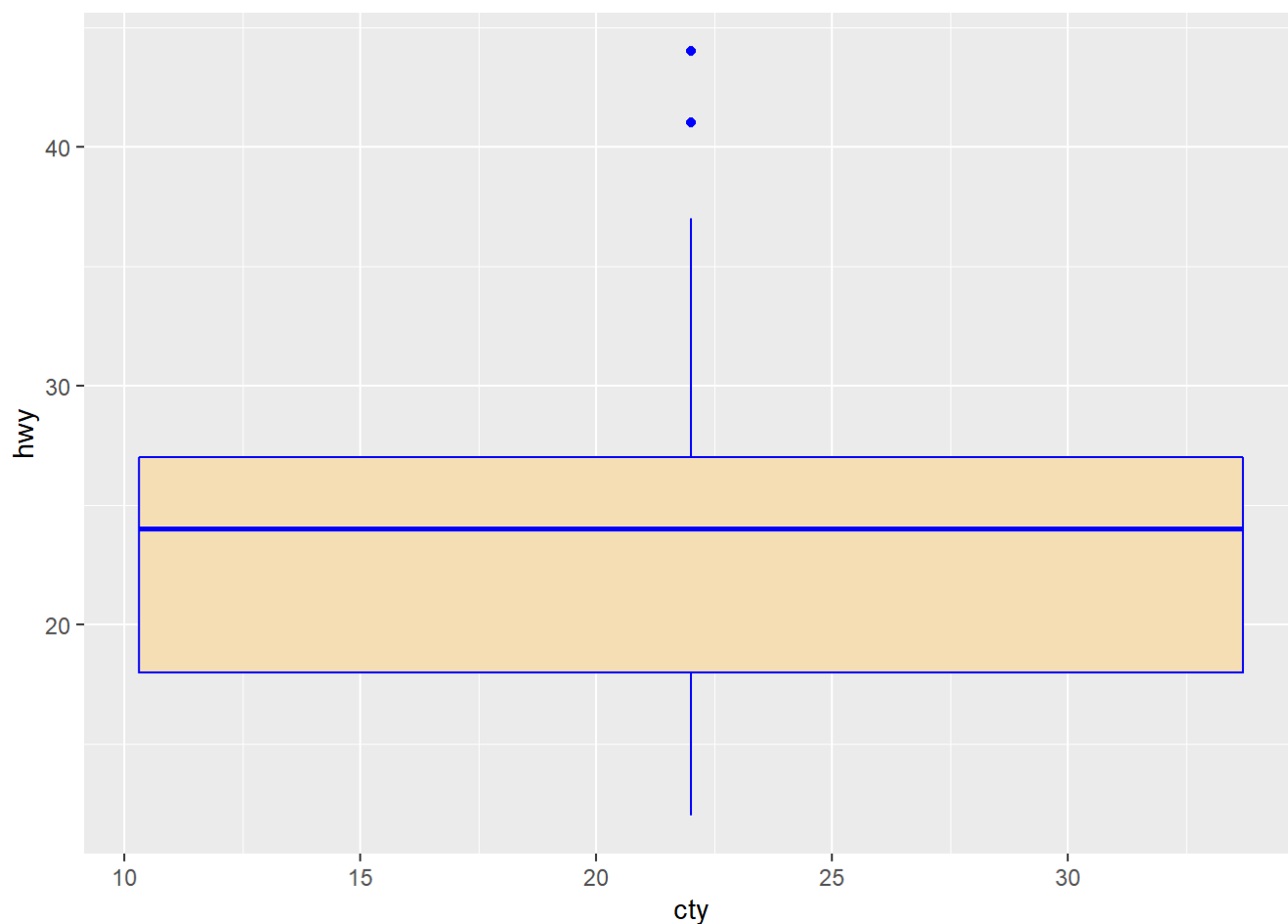
LES BOITES A MOUSTACHE :

Definition : Dans les représentations graphiques de données statistiques, la boîte à moustaches est un moyen rapide de figurer le profil essentiel d'une série statistique quantitative. Elle a été inventée en 1977 par John Tukey, mais peut faire l'objet de certains aménagements selon les utilisateurs.

utilisable avec la commande : "geom_boxplot" :

```
ggplot(mpg) + geom_boxplot(aes(x = cty , y = hwy), fill = "wheat", color = "blue")
```

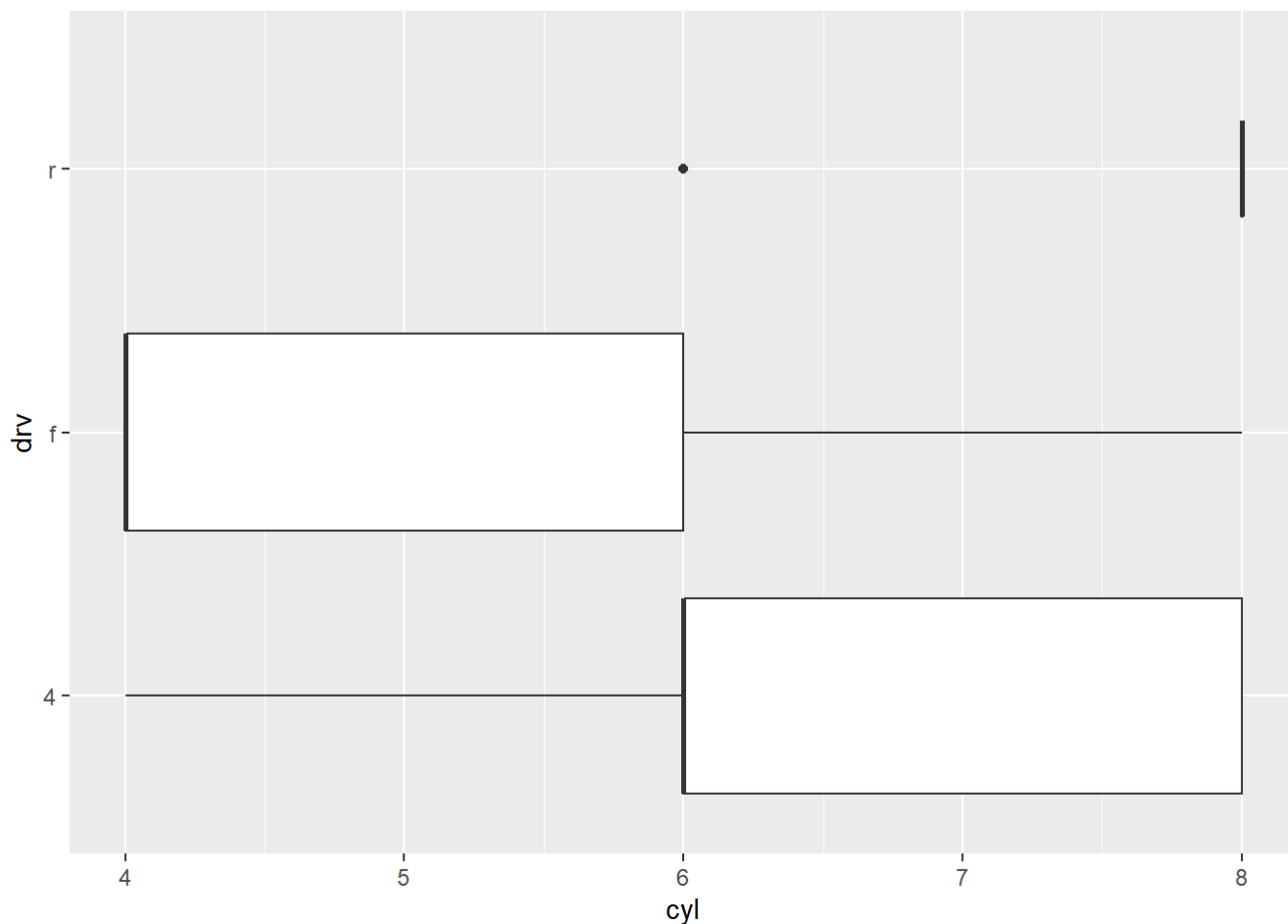
```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



exemple toujours avec le tableau ci dessus Personnalisable avec les commandes :

Toujours avec les boites à moustaches, la commande “varwidth” permet de varier la longueur des boites en fonctions des effectifs de la catégorie. Pour l’appliquer, on fait :

```
ggplot(mpg) + geom_boxplot(aes(x = cyl, y = drv), varwidth = TRUE)
```



Pour cela, il faut utiliser la commande : `geom_histogram` qui se configure de la manière suivante

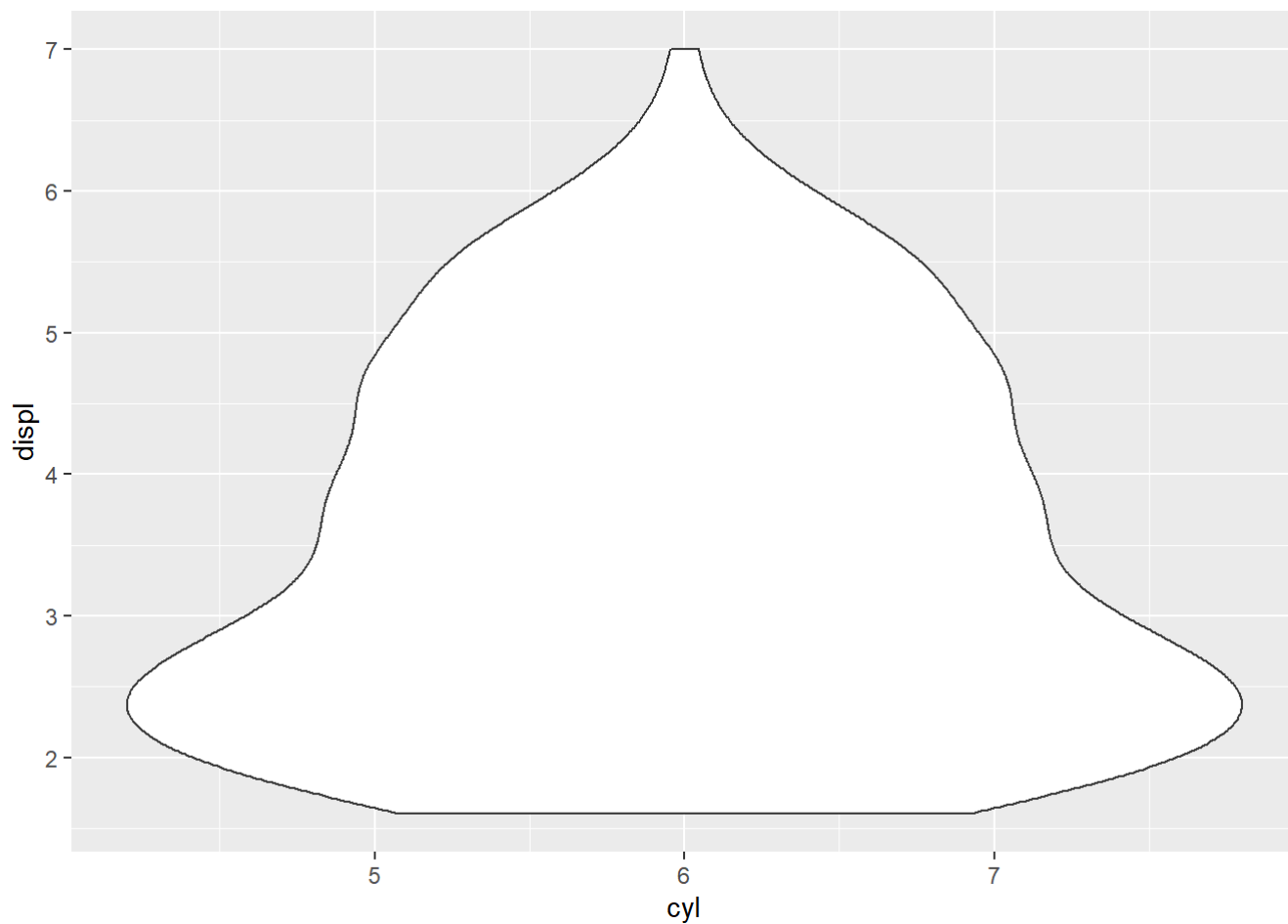
```
geom_histogram(aes(x = mpg))
```

Définir X permet d'afficher la donnée que l'on souhaite représenter.

Geom_violin :

`geom_violin` est très semblable à `geom_boxplot`, mais utilise des graphes en violon à la place des boîtes à moustache. Les graphes en violon peuvent donner une lecture plus fine des différences de distribution selon les classes.

```
ggplot(mpg) + geom_violin(aes(x = cyl, y = displ))
```



##Pour plus d'information : <https://juba.github.io/tidyverse/08-ggplot2.html#ressources-1>
(<https://juba.github.io/tidyverse/08-ggplot2.html#ressources-1>)