

Réponse aux TD data visualisation

Laurent Politis

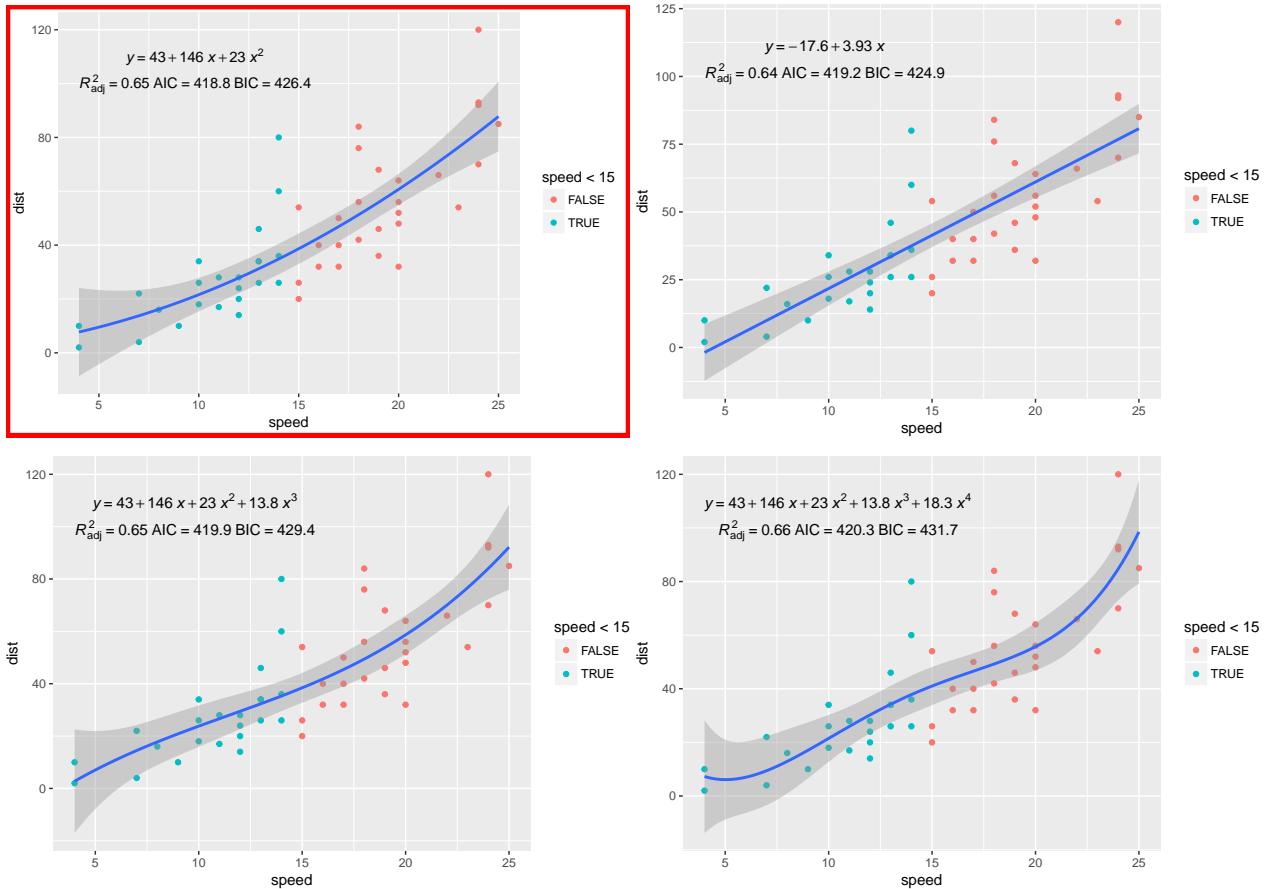
21 février 2017

Exercice 1

Nous avons vu plusieurs modèles la dernière fois pour modéliser la vitesse et la distance de freinage en fonction des différentes métriques d'erreurs présentées au-dessus. Sélectionner le modèle statistique le plus pertinent en modifiant `ma_formule`.

Réponse

Le meilleur modèle est celui qui maximise le R^2 et minimise les critères d'informations Bayésien et d'Akaike.



Exercice 2

- Quelle est le type de cet objet ?

```
class(mpg)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

L'objet est une `data.frame`.

- Que contient cet objet ?

```
?mpg
```

This dataset contains a subset of the fuel economy data that the EPA makes available on <http://fueleconomy.gov>. It contains only models which had a new release every year between 1999 and 2008 - this was used as a proxy for the popularity of the car.

```
head(mpg)
```

```
## # A tibble: 6 × 11
##   manufacturer model displ year cyl trans drv cty hwy fl
##   <chr> <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr>
## 1 audi     a4    1.8  1999     4 auto(15) f    18    29 p
## 2 audi     a4    1.8  1999     4 manual(m5) f    21    29 p
## 3 audi     a4    2.0  2008     4 manual(m6) f    20    31 p
## 4 audi     a4    2.0  2008     4 auto(av)   f    21    30 p
## 5 audi     a4    2.8  1999     6 auto(15) f    16    26 p
## 6 audi     a4    2.8  1999     6 manual(m5) f    18    26 p
## # ... with 1 more variables: class <chr>
```

- Que signifie `displ` et `hwy` ?

displ

engine displacement, in litres

hwy

highway miles per gallon

- Quelle est le type des vecteurs `hwy`, `displ` et `manufacturer` ?

```
class(mpg$hwy)
```

```
## [1] "integer"
```

```
class(mpg$displ)
```

```
## [1] "numeric"
```

```
class(mpg$manufacturer)
```

```
## [1] "character"
```

- Combien y-a-t il de ligne et de colonne dans `mpg` ?

La data frame contient 234 lignes et 11 variables (colonnes).

Exercice 3

Reproduisez le scatterplot ci-dessous entre `hwy` et `displ` puis entre `hwy` et `cyl`; entre `class` et `drv`. Expliquez pourquoi ces graphes ont ces formes.

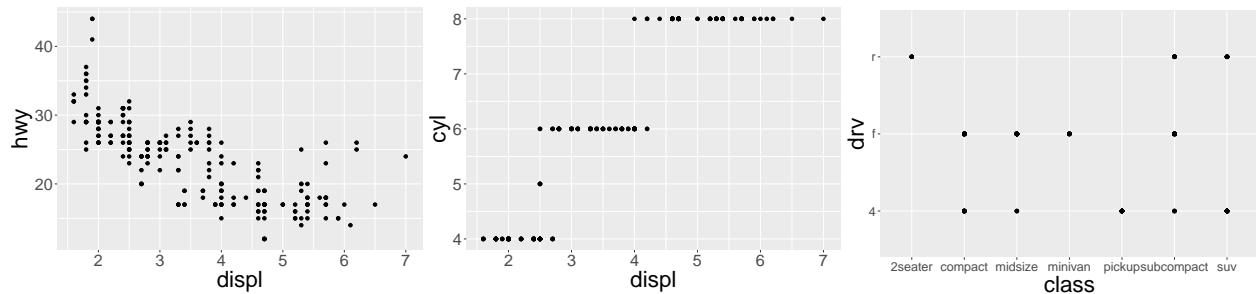
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))+
  theme(plot.subtitle = element_text(vjust = 1),
        plot.caption = element_text(vjust = 1),
        axis.title = element_text(size = 25),
        axis.text = element_text(size = 18))
```

```

ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = cyl))+
  theme(plot.subtitle = element_text(vjust = 1),
  plot.caption = element_text(vjust = 1),
  axis.title = element_text(size = 25),
  axis.text = element_text(size = 18))

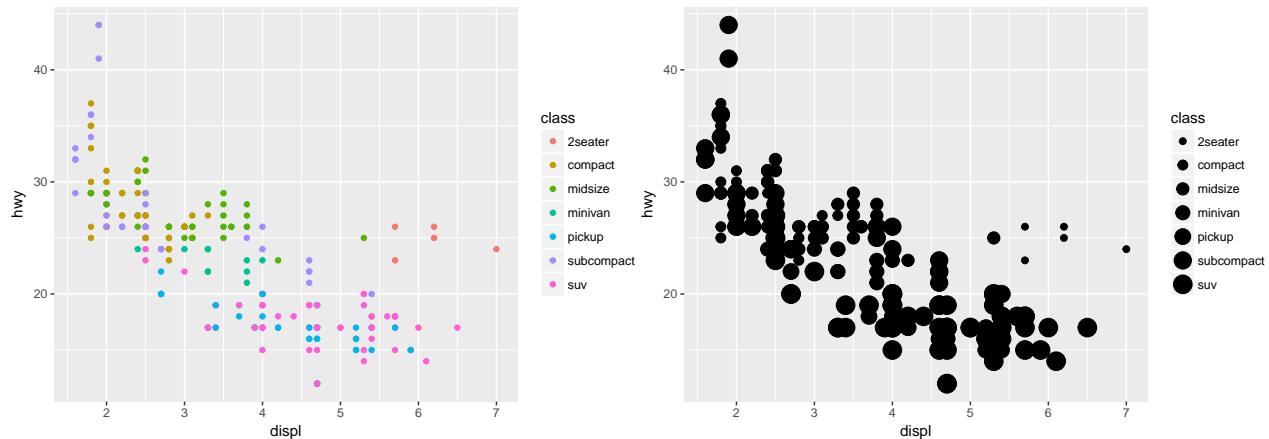
ggplot(data = mpg) +
  geom_point(mapping = aes(x = class, y = drv))+
  theme(plot.subtitle = element_text(vjust = 1),
  plot.caption = element_text(vjust = 1),
  axis.title = element_text(size = 25),
  axis.text = element_text(size = 14))

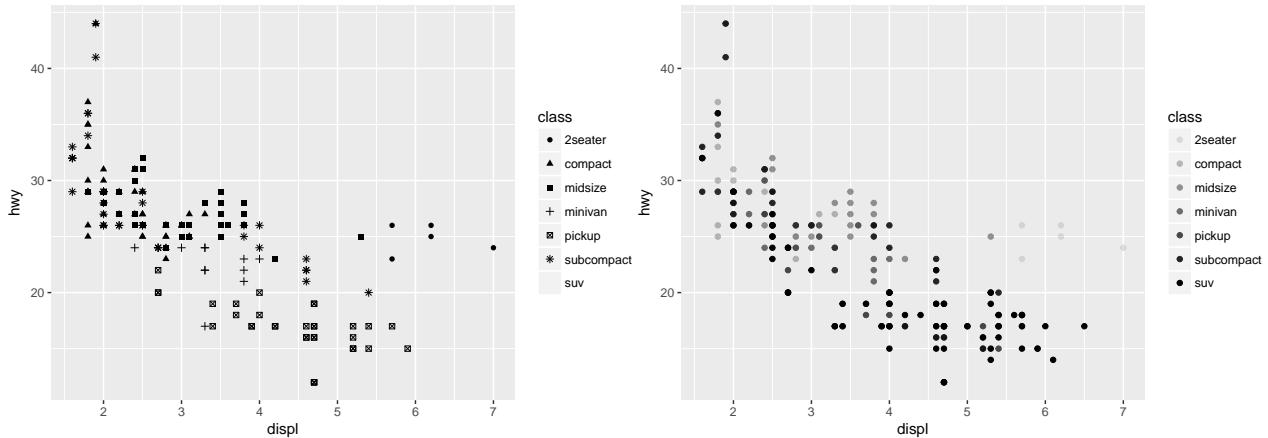
```



Exercice 4

- Dessiner un nuage de points pour chaque esthétique `aes(x,y,shape = class)` défini par la colonne `class` pour les couleurs `color`, les formes `shape`, la taille `size`, et la transparence `alpha`.





- Pourquoi il y a des points qui ont disparu de la classe `suv` dans le graphique où l'on dessine le nuage avec différentes formes de point `aes(x,y,shape=class)` ?

Au-delà de 7 classes, ggplot ne gère plus automatiquement les types de points et de lignes aussi. Il est nécessaire alors d'utiliser la `scale_shape_manual()`

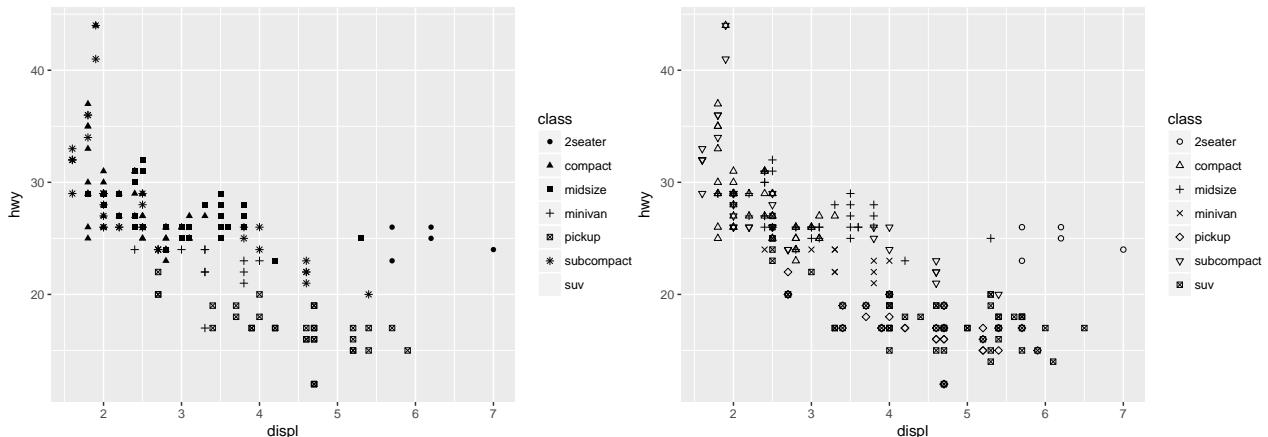
- Que faut-il ajouter pour corriger le graphique ? (indice et regardez le message de warning afficher en console)

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, shape = class, fill=class))
```

```
## Warning: The shape palette can deal with a maximum of 6 discrete values
## because more than 6 becomes difficult to discriminate; you have 7.
## Consider specifying shapes manually if you must have them.

## Warning: Removed 62 rows containing missing values (geom_point).

ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, shape = class, fill=class)) + scale_shape_manual(values=seq(1, 7))
```



- Quel est le problème dans ce code ?

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, colour="class"))
```

Le graphique comprend qu'il y a qu'une classe de couleur pour le nuage de point.

- Quelle est la différence avec les graphiques précédent et quel est l'impact sur les graphes du code ci-dessous ?

```

ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy), colour="blue")

  ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy), size=10)

```

Les caractéristiques du graphe sont spécifiées en dehors de la fonction `aes()`.

Exercice 5

1. A quoi sert le `.` dans la formule de `facet_grid` ? (Tester les exemples en dessous)

```

ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(drv ~ .)

ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(. ~ cyl)

```

2. Quelles sont les avantages et les désavantages d'utiliser les fonctions `facet` par rapport à une esthétique `aes(x,y,color=...)` ?

```

ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +  facet_wrap(facets = ~ class)

ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy,color=class))

```

Les avantages dépendent de vos données : l'avantage principal de `facet` est de pouvoir se concentrer sur chaque type de population de votre échantillon. Si vous avez peu (une dizaine) de sous groupe dans votre échantillon `facet` peut être utilisée. Dans le cas contraire il faut faire attention à la taille de vos graphiques et surtout la visibilité de vos axes qui peuvent devenir illisible en raison du nombre trop important de panneaux générés.

3. Quelles sont les différences entre `facet_grid()` et `facet_wrap()` ? (Indice utiliser ?)

La fonction `facet_wrap` optimise l'espace du graphe grâce à la possibilité de créer des séquences de panneaux en 2 dimensions alors que `facet_grid` constitue une matrice de panneaux en fonction que d'un critère en 1d.

Exercice 6

- Que font les options `se=FALSE` et `show.legend=FALSE` ?

```

ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color = drv)) +
  geom_point(show.legend=F) +
  geom_smooth(se = FALSE,show.legend=FALSE)

```

`se` est le champ des possibles de votre courbe de modélisation statistique à un interval de confiance de 95 %.

`show.legend` si l'est égal à vraie alors la légende est présente, sinon elle n'est pas sur le graphe.

- Ces deux graphes sont ils différents ?

La position de `aes(x = displ, y = hwy)` permet de synthétiser ton code.

```

ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth()

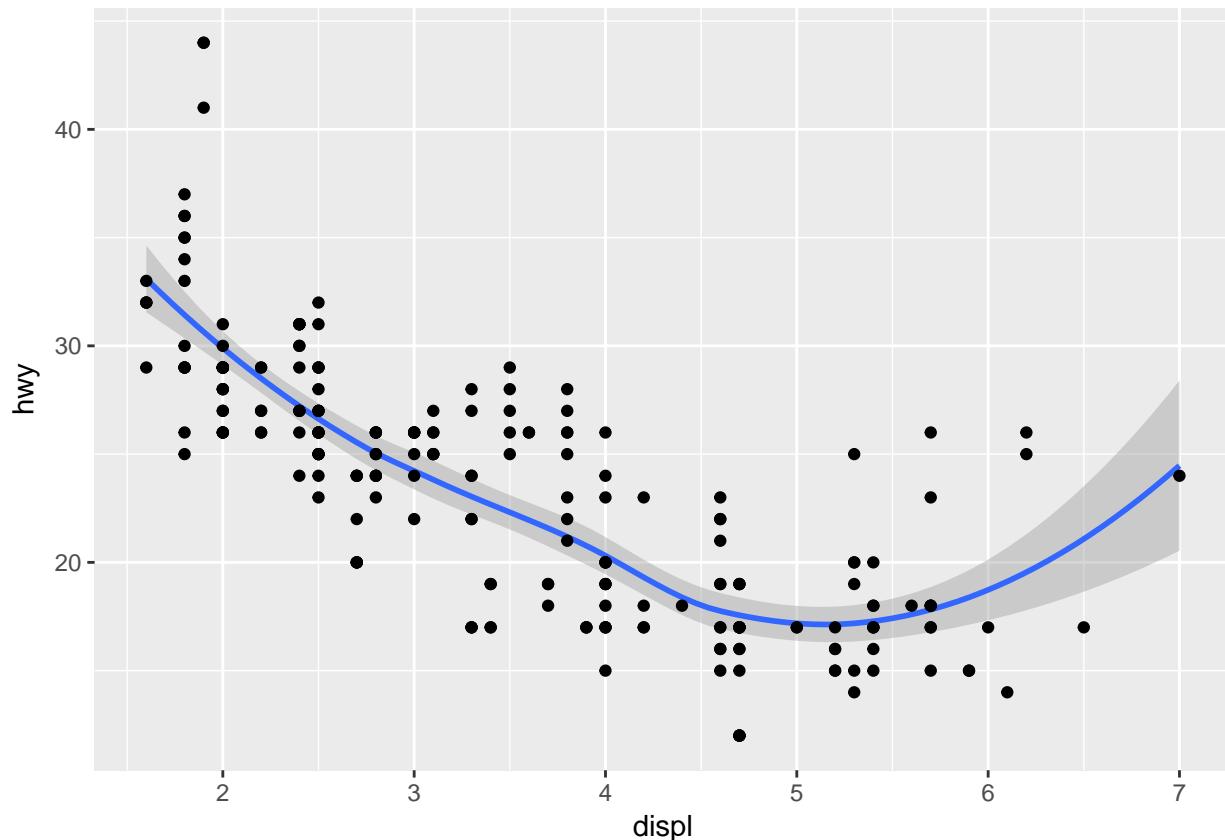
ggplot() +

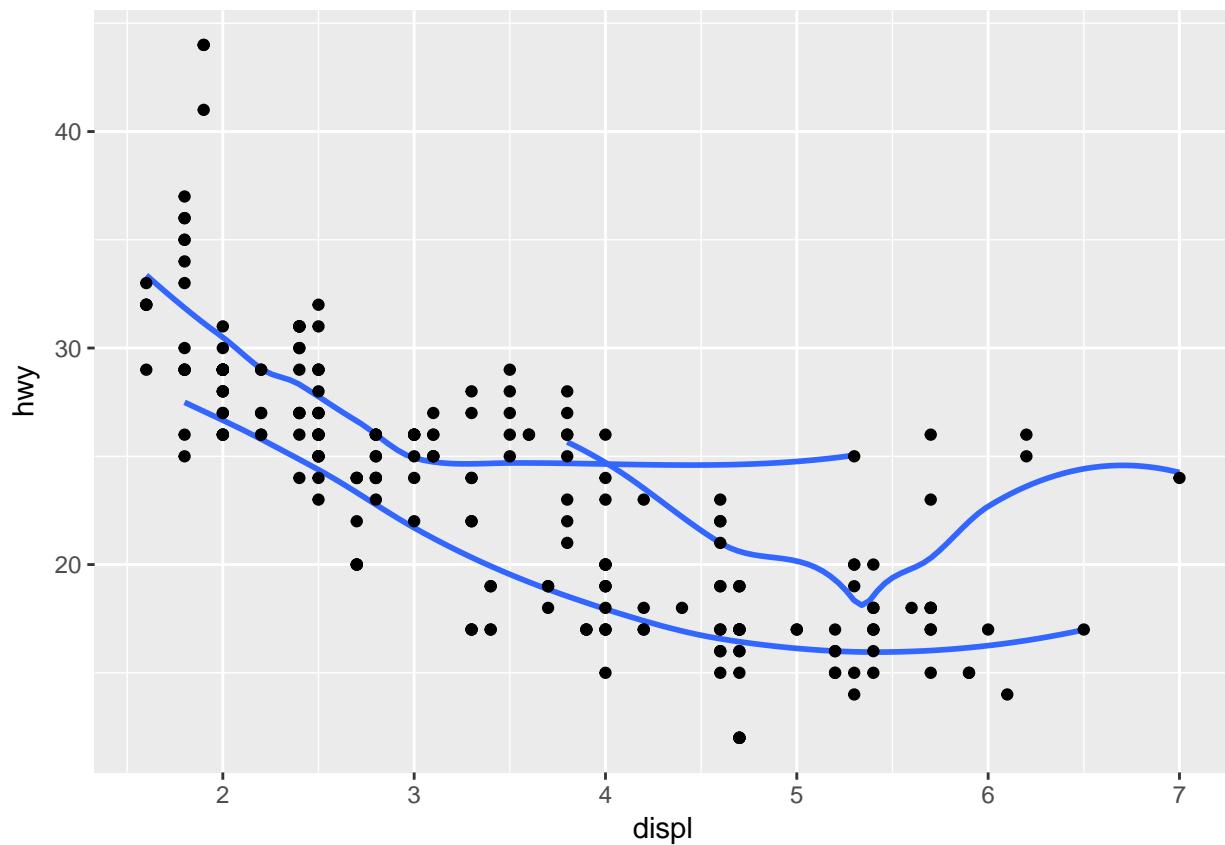
```

```
geom_point(data = mpg, mapping = aes(x = displ, y = hwy)) +  
geom_smooth(data = mpg, mapping = aes(x = displ, y = hwy))
```

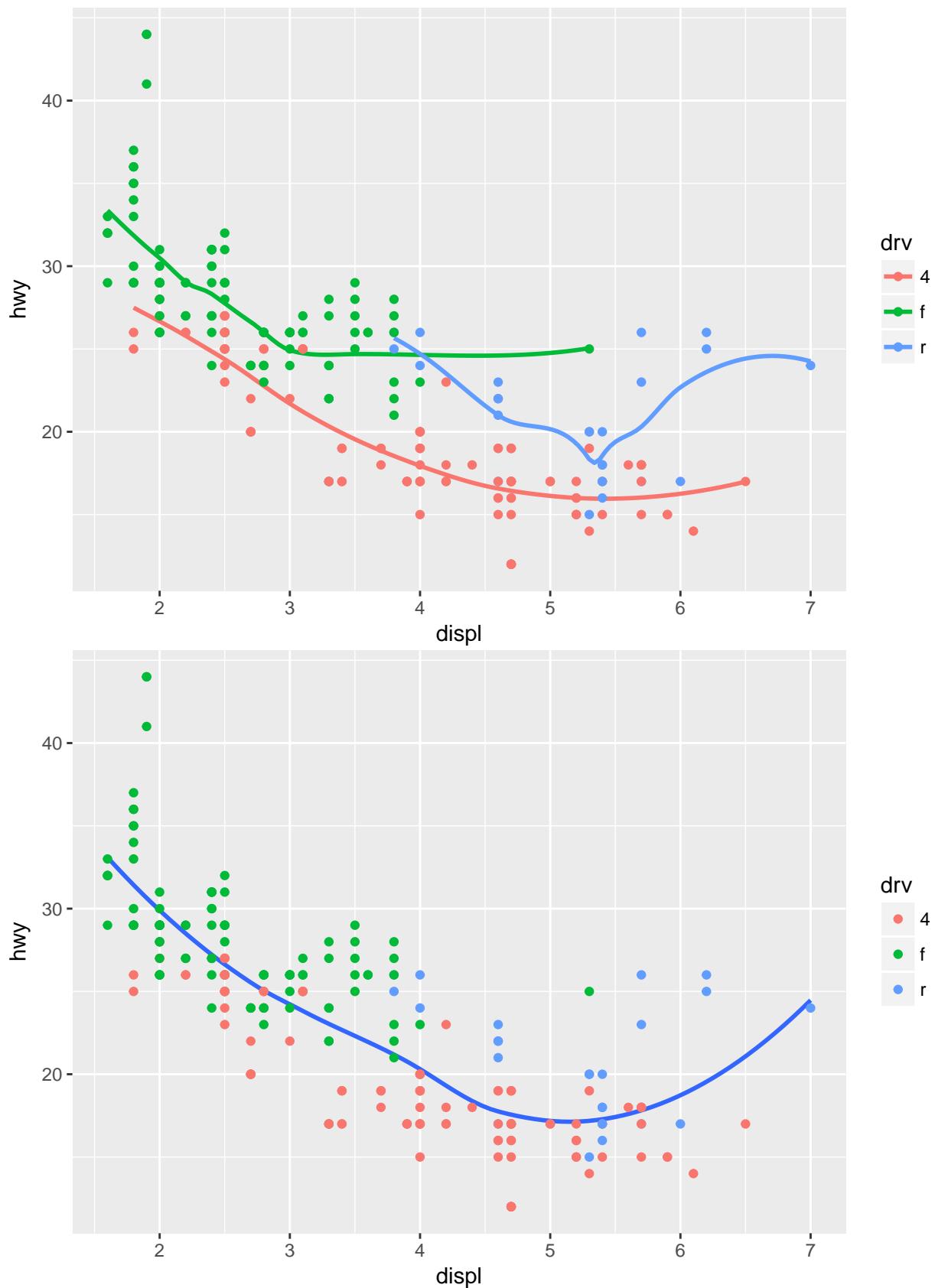
- Refaire les graphes ci-dessous :

```
## `geom_smooth()` using method = 'loess'  
## `geom_smooth()` using method = 'loess'
```



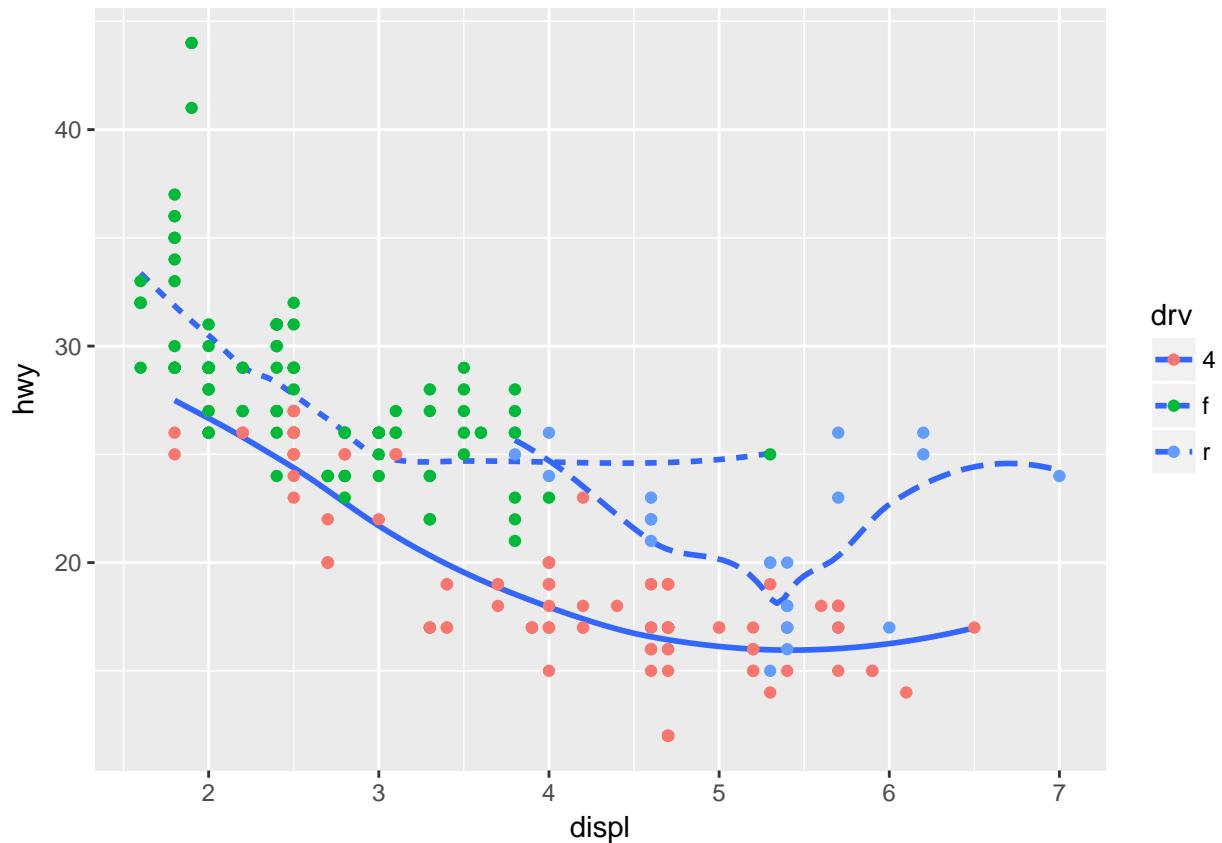


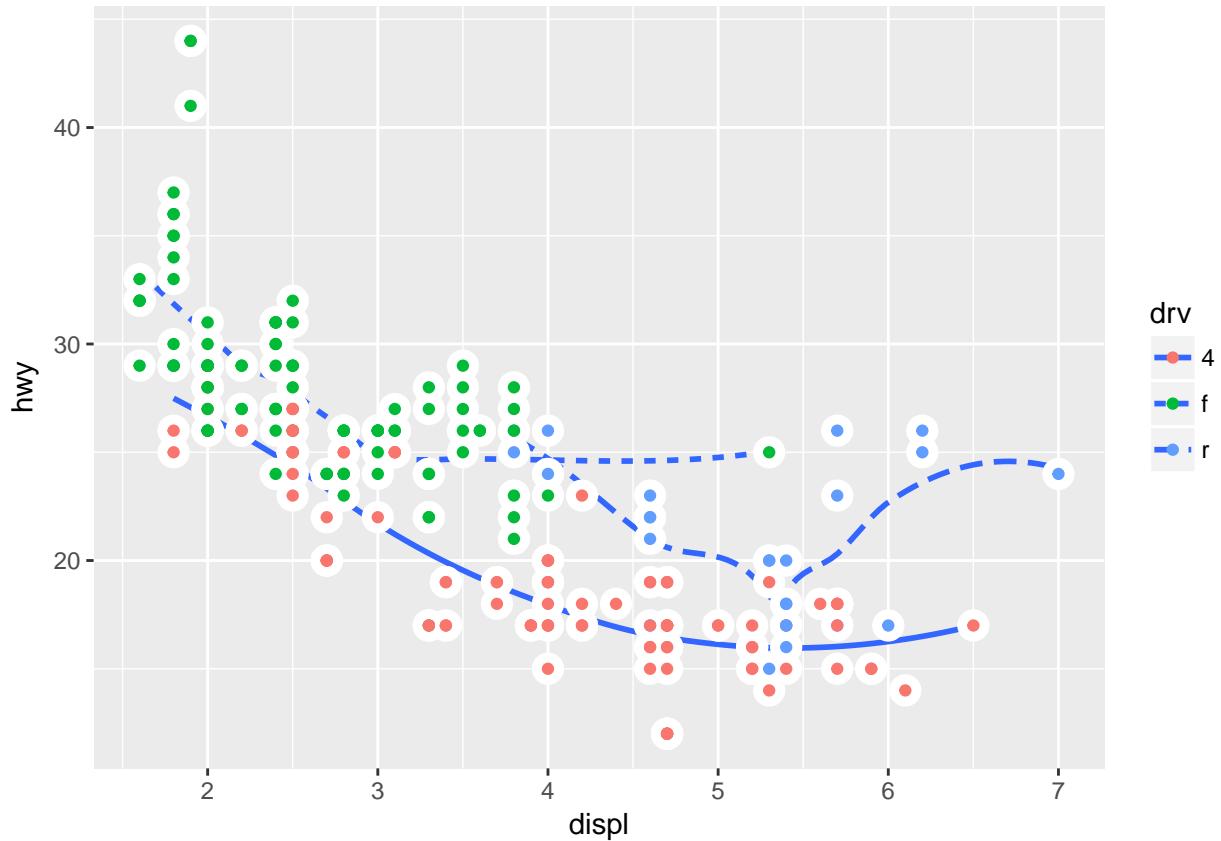
```
## `geom_smooth()` using method = 'loess'  
## `geom_smooth()` using method = 'loess'
```



```
## `geom_smooth()` using method = 'loess'
```

```
## `geom_smooth()` using method = 'loess'
```





Exercice 7

1. Quel est l'équivalent géométrique de `stat_summary()` ? Modifiez le code ci-dessus en utilisant une fonction `geom_`.

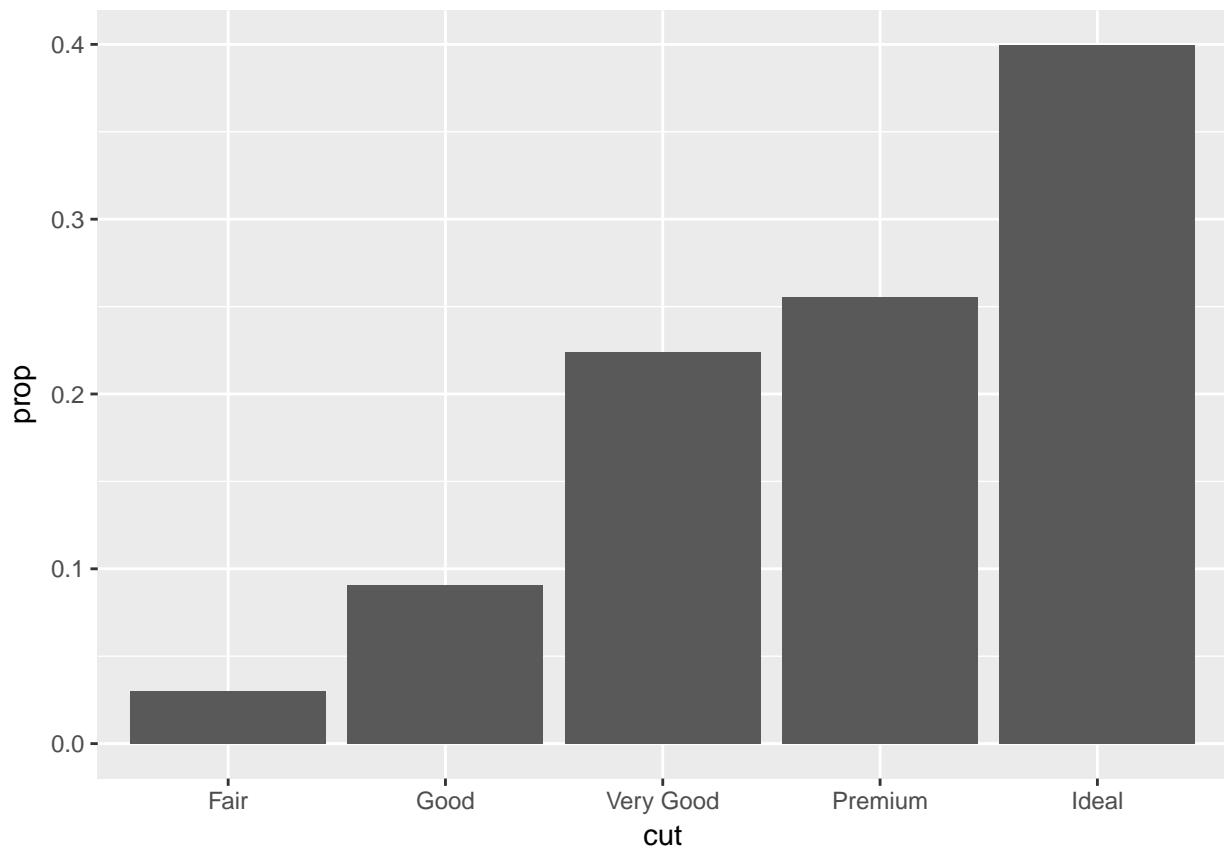
L'équivalent de `stat_summary` est `geom_point`

2. Quelle est la différence entre les fonctions `stat_` et `geom_` ? Listez les paires entre les deux genres de fonctions.

Les fonctions `geom_` définissent les objets de votre graphique à l'instar des fonctions `stat` qui font des transformations .

3. Quelle est le “bug” de ce graphique? Corrigez le.

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, y = ..prop..,group=1))
```

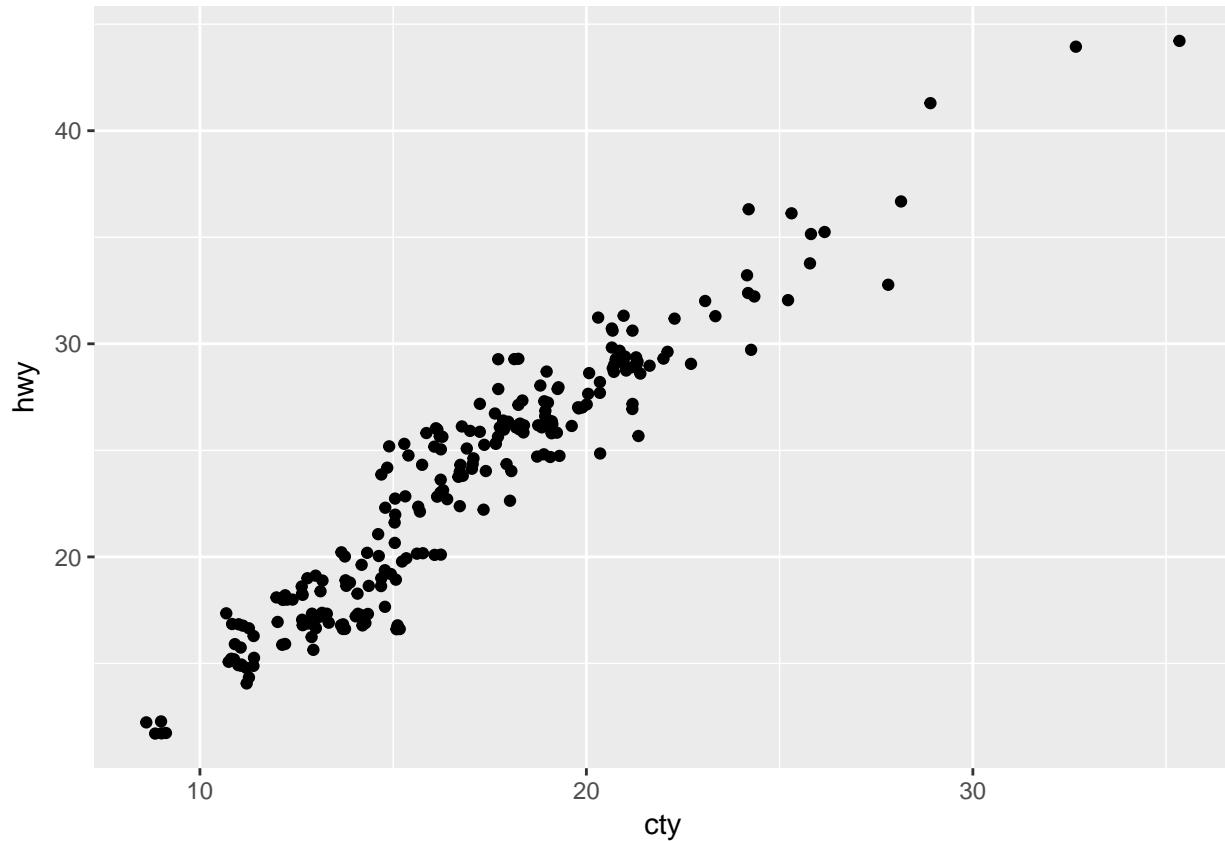


Exercice 8

1. Quel est le problème avec ce graphique ? Comment pouvez vous l'améliorer ?

Utiliser `geom_jitter()`.

```
ggplot(data = mpg, mapping = aes(x = cty, y = hwy)) +  
  geom_jitter()
```

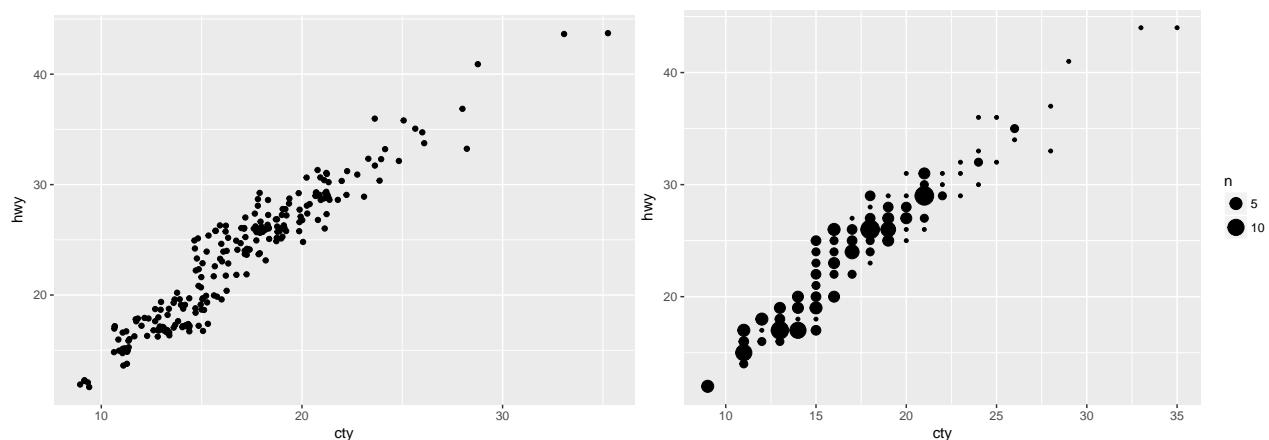


2. Quels sont les arguments de `geom_jitter()`?

?`geom_jitter()` pour connaître les arguments de `geom_jitter`.

3. Comparer `geom_jitter()` et `geom_count()` ?

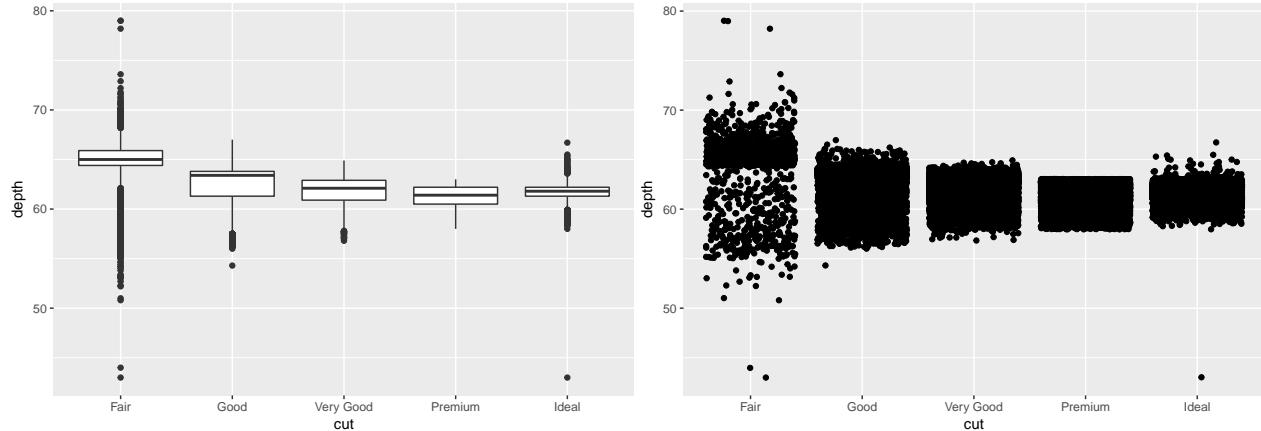
```
ggplot(data = mpg, mapping = aes(x = cty, y = hwy)) +
  geom_jitter()
ggplot(data = mpg, mapping = aes(x = cty, y = hwy)) +
  geom_count()
```



4. Recréer ces graphiques

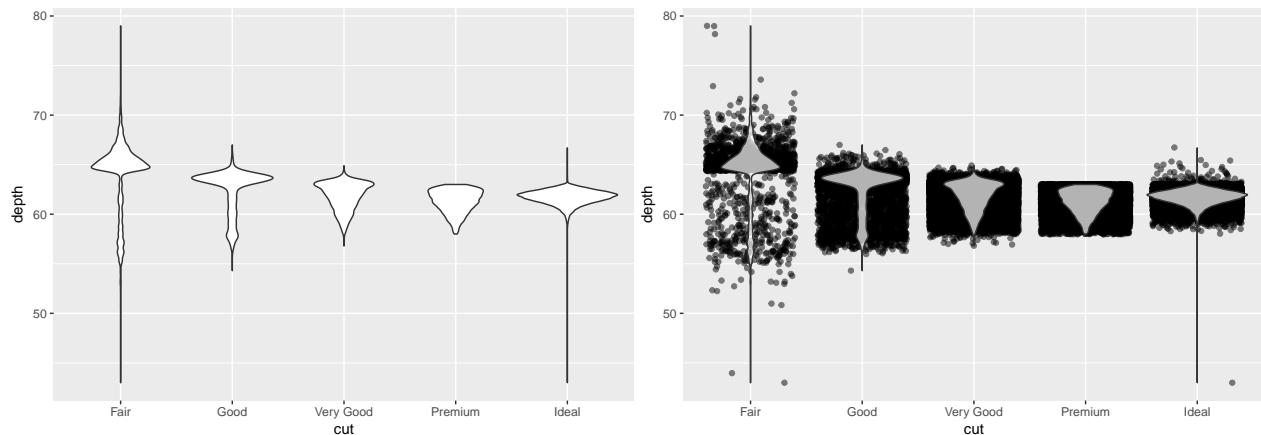
```
ggplot(data = diamonds, mapping = aes( x= cut,y=depth)) +
  geom_boxplot()
```

```
ggplot(data = diamonds, mapping = aes( x= cut,y=depth)) +
  geom_jitter()
```



```
ggplot(data = diamonds, mapping = aes( x= cut,y=depth)) +
  geom_violin()
```

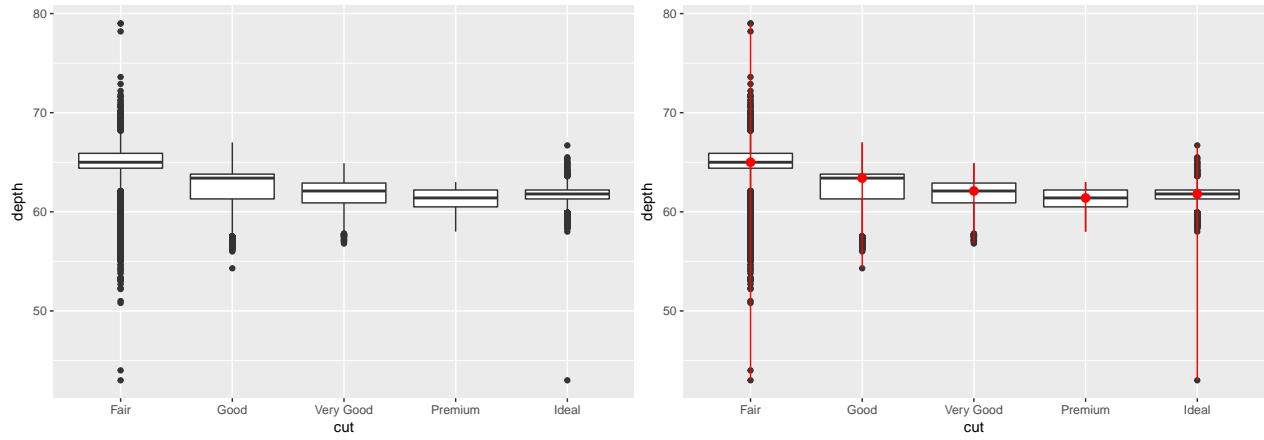
```
ggplot(data = diamonds, mapping = aes( x= cut,y=depth))+  
  geom_jitter(alpha=0.5)+ geom_violin(alpha=0.7)
```



5. A quelles valeurs/métriques correspondent les traits horizontaux de la boîte à moustache ? (Indice utiliser `stat_summary`)

```
ggplot(data = diamonds, mapping = aes(x=cut,y=depth)) +
  geom_boxplot()
```

```
ggplot(data = diamonds, mapping = aes(x=cut,y=depth)) +
  geom_boxplot()+
  stat_summary( color="red", fun.ymin = min,
    fun.ymax = max,
    fun.y = median)
```



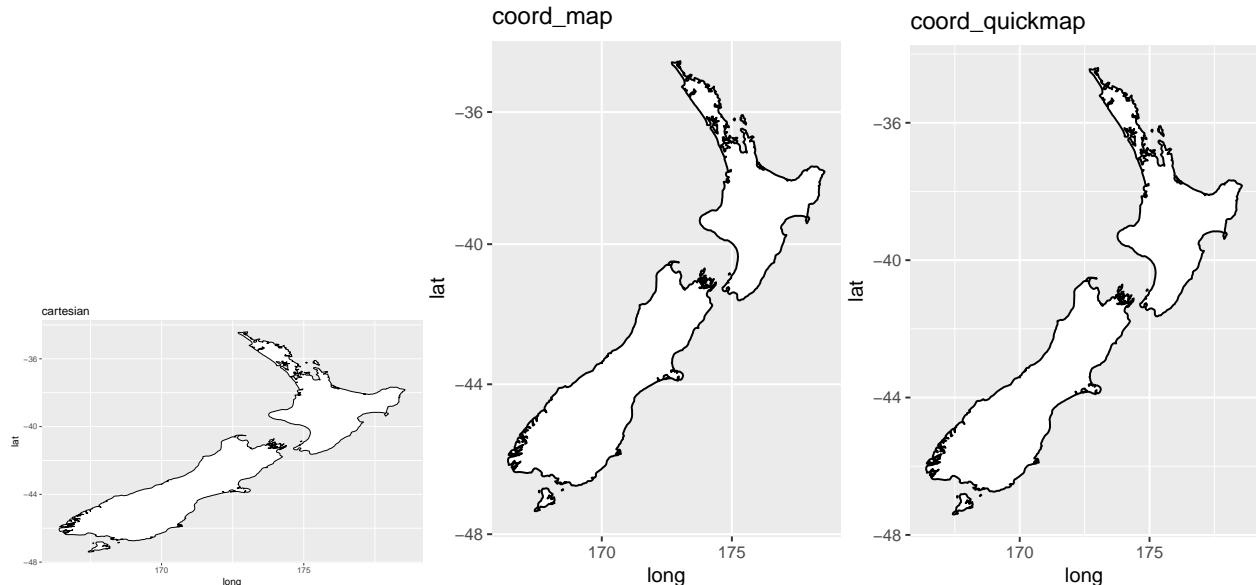
Exercice 9

1. Dessinez le département et une région de votre choix ?
2. Quelle est la différence entre `coord_map()` et `coord_quickmap()` ?

Le temps de calcul est plus long avec `coord_map()` car elle n'est pas aussi approximative que `coord_quickmap()`.
Ex. :

```
nz <- map_data("nz")
# Prepare a map of NZ
nzmap <- ggplot(nz, aes(x = long, y = lat, group = group)) +
  geom_polygon(fill = "white", colour = "black")

# Plot it in cartesian coordinates
nzmap + labs(title="cartesian")
# With correct mercator projection
nzmap + coord_map() + labs(title="coord_map")
# With the aspect ratio approximation
nzmap + coord_quickmap() + labs(title="coord_quickmap")
```



Exercice 10

1. Changer le thème du précédent graphes ?

```
fr <- map_data("france")

ggplot(fr, aes(long, lat, group = group)) +
  geom_polygon(fill = "white", colour = "black") +
  coord_quickmap() + theme_nothing()
```