

# GGPLOT2

Outil de visualisation de données

# Pourquoi utiliser une librairie de visualisation ?

1. Transformer des **données brutes** en **informations visuelles** compréhensibles.
2. Aider à **explorer les données**, détecter des tendances, anomalies, corrélations.
3. Faciliter la **communication des résultats** à un public non technique.
4. Produire des **graphes reproductibles et personnalisables** via du code.
5. Gagner du **temps** et assurer la **cohérence** dans la présentation des graphiques.

# Comment fonctionne ces librairies ?

Elles utilisent une **logique de couches** :

- ❶ On fournit les **données**.
- ❷ On définit les **variables à représenter** (axes, couleurs, tailles...).
- ❸ On ajoute des **éléments visuels** (*geom\_point*, *geom\_bar*, etc.).
- ❹ On personnalise le **style** (titres, thèmes, échelles...).

# Exemples de bibliothèques de visualisation



ggplot2



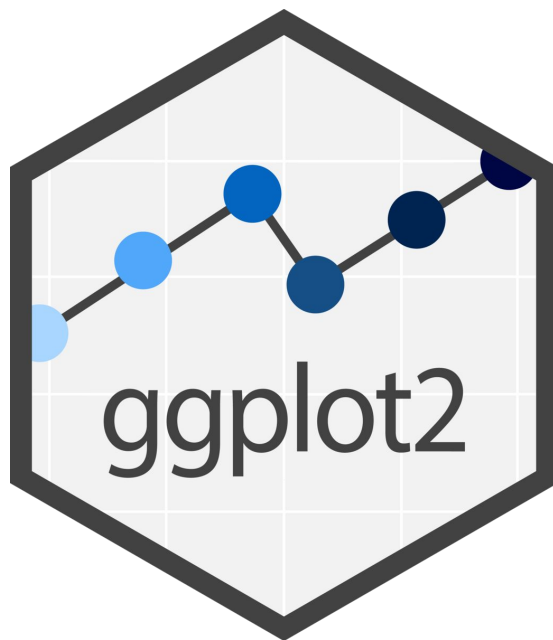
matplotlib, seaborn, plotly

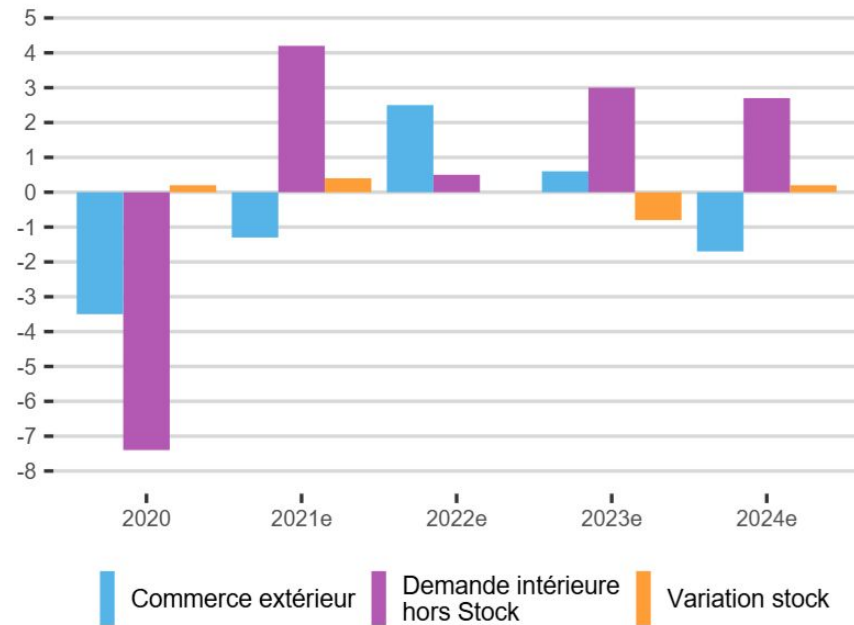
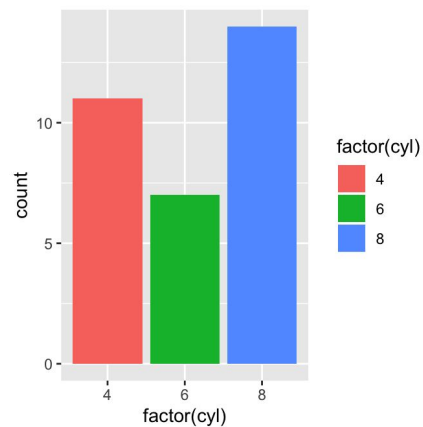
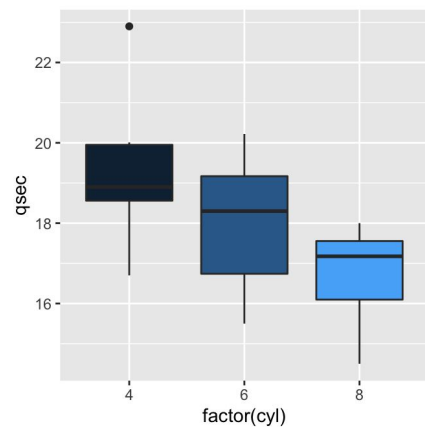
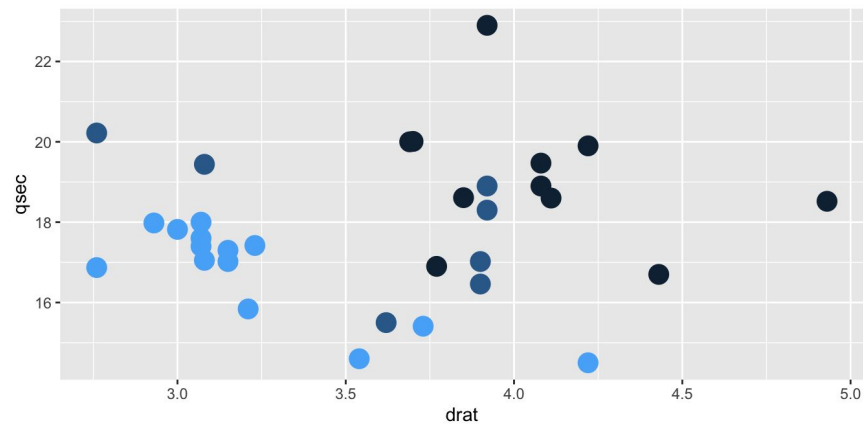


[D3.js](#), Chart.js



[Makie.jl](#), Plots.jl





Source : Comptes économiques rapides - ISPF, INSEE

# GGPLOT2

TP Démo

# **Objectif de ce TP**

Présentation des différents graphiques



# Exercice 1 - Base de données

```
# install.packages("ggplot2") # à faire une seule fois
# install.packages("Amelia")
# install.packages("tinytex")
library(dplyr)
library(Amelia)
library(ggplot2) # charger la librairie
data(mpg)
```

```
{r}
head(mpg)
```

A tibble: 6 × 11

manufacturer <chr>	model <chr>	displ <dbl>	year <int>	cyl <int>	trans <chr>	drv <chr>	cty <int>	hwy <int>	fl <chr>	
audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	
audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	
audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	
audi	a4	2.0	2008	4	auto(av)	f	21	30	p	
audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	
audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	

6 rows | 1-10 of 11 columns

```
str(mpg)
```

```
tibble [234 × 11] (S3: tbl_df/tbl/data.frame)
```

```
$ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...  
$ model       : chr [1:234] "a4" "a4" "a4" "a4" ...  
$ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...  
$ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...  
$ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...  
$ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...  
$ drv         : chr [1:234] "f" "f" "f" "f" ...  
$ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...  
$ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...  
$ fl          : chr [1:234] "p" "p" "p" "p" ...  
$ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

manufacturer : constructeur de la voiture  
model : modèle  
displ : cylindrée (L)  
year : année  
cyl : nombre de cylindres  
trans : type de transmission  
drv : type de traction (f avant, r arrière, 4 4 roues)  
cty : consommation ville (mpg)  
hwy : consommation route (mpg)  
fl : type de carburant  
class : catégorie de véhicule (suv, compact, etc.)

```
```{r}
```

```
mpg <- mpg %>%  
  mutate(drv = recode(drv,  
    "4" = "4 roues",  
    "f" = "traction avant",  
    "r" = "traction arrière"))
```

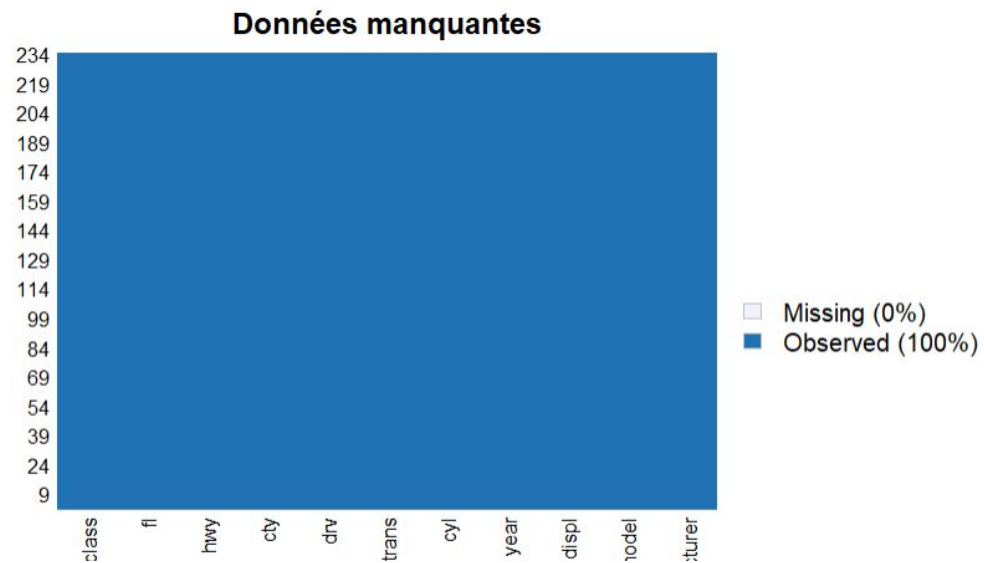
```
mpg <- mpg %>%  
  mutate(fl = recode(fl,  
    "p" = "essence premium",  
    "r" = "essence regular",  
    "d" = "diesel",  
    "e" = "ethanol",  
    "c" = "CNG"))
```

```
str(mpg)
```

```
tibble [234 × 11] (S3: tbl_df/tbl/data.frame)
```

```
$ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...  
$ model       : chr [1:234] "a4" "a4" "a4" "a4" ...  
$ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...  
$ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...  
$ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...  
$ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...  
$ drv         : chr [1:234] "traction avant" "traction avant" "traction avant" ...  
$ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...  
$ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...  
$ fl          : chr [1:234] "essence premium" "essence premium" "essence premium" ...  
$ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

```
## {r}  
missmap(mpg, main = "Données manquantes")  
##
```



# Syntaxe de GGPLOT2

```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION>(  
    mapping = aes(<MAPPINGS>),  
    stat = <STAT>,  
    position = <POSITION>)+  
  <COORDINATE_FUNCTION>+  
  <FACET_FUNCTION>+  
  <SCALE_FUNCTION>+  
  <THEME_FUNCTION>
```

# Elements obligatoires

ggplot(data = <DATA>) +

<GEOM\_FUNCTION>(  
 mapping = aes(<MAPPINGS>),  
 stat = <STAT>,  
 position = <POSITION>)+

<COORDINATE\_FUNCTION>+

<FACET\_FUNCTION>+

<SCALE\_FUNCTION>+

<THEME\_FUNCTION>

**<DATA>** Jeu de données

**<GEOM\_FUNCTION>** Géométrie ou type de graphe Exemple :  
bar,scatter

**<MAPPING>** Correspondance visuelle entre les variables  
Exemple : x,y,color,fill,size,shape

# Elements non obligatoires

```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION>(  
    mapping = aes(<MAPPINGS>),  
    stat = <STAT>,  
    position = <POSITION>)+  
  <COORDINATE_FUNCTION>+  
  <FACET_FUNCTION>+  
  <SCALE_FUNCTION>+  
  <THEME_FUNCTION>
```

## **<STAT> Transformation statistique**

Exemple :

stat = “identity” : valeur par défaut

stat = “count” : compter le nombre d’occurrence

stat = “density” : calculer une densité

## **<POSITION> Position des éléments graphiques quand ils se chevauchent**

Exemple :

position = “stack” : empiler les barres

position = “dodge” : côte à côte

position = “fill” : empiler en normalisant

## **<COORDINATE\_FUNCTION> SYSTEME DE COORDONNEES**

## **<FACET\_FUNCTION> CREATION DE SOUS-GRAPHIQUE**

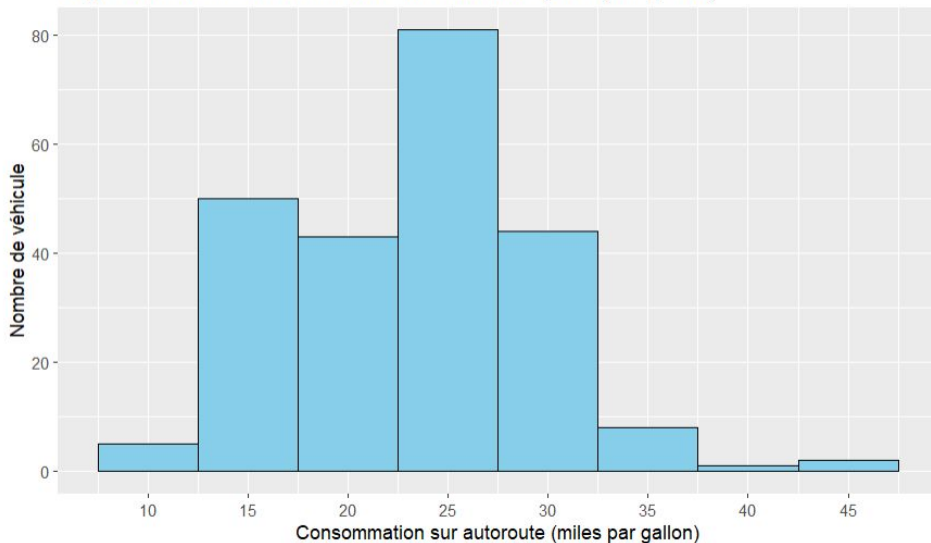
## **<SCALE\_FUNCTION> PERSONALISER LES AXES, COULEURS, ETC**

## **<THEME\_FUNCTION> APPARENCE GENERAL DU GRAPHIQUE**

# Histogramme

```
ggplot(mpg) +  
  geom_histogram(aes(x=hwy), binwidth = 5, fill = "skyblue", color = "black") +  
  scale_x_continuous(breaks = seq(0,50,5)) +  
  labs(  
    title = "Répartition de la consommation sur autoroute (miles par gallon)",  
    x = "Consommation sur autoroute (miles par gallon)",  
    y = "Nombre de véhicule"  
  )  
  )
```

Répartition de la consommation sur autoroute (miles par gallon)

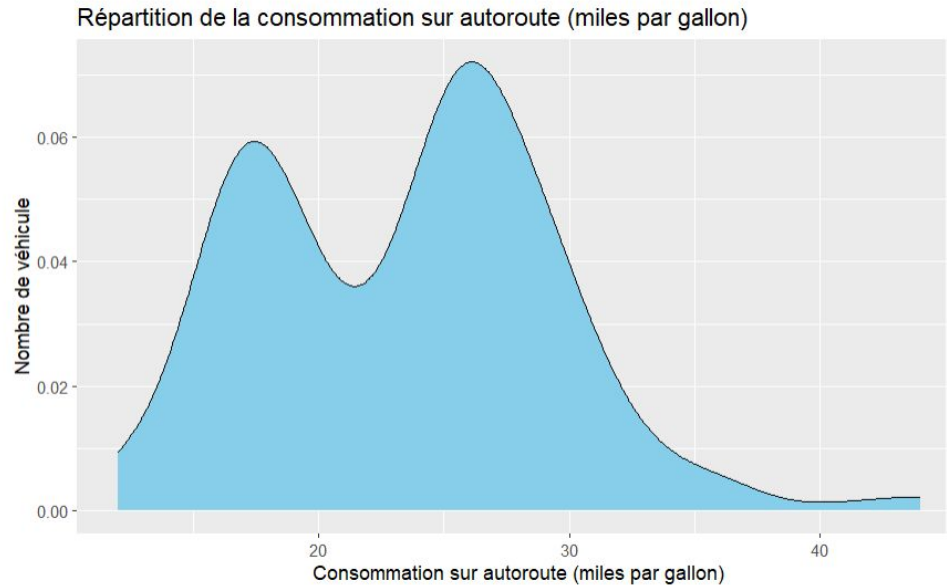


1. `ggplot(mpg) +`
2. `geom_histogram(aes(x=hwy), binwidth = 5, fill = "skyblue", color = "black") +`
3. `scale_x_continuous(breaks = seq(0,50,5)) +`
4. `labs(  
 title = "Répartition de la consommation sur autoroute (miles par gallon)",  
 x = "Consommation sur autoroute (miles par gallon)",  
 y = "Nombre de véhicule"  
)`

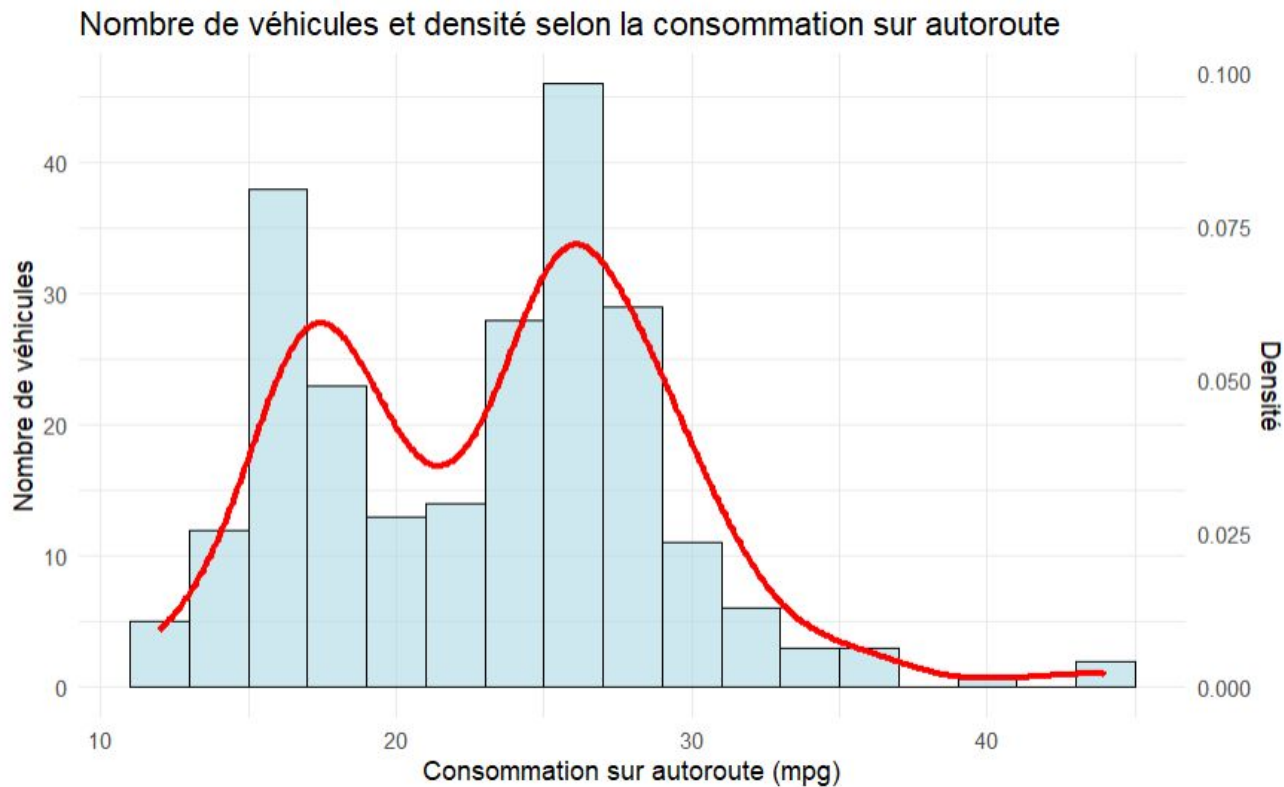


# Fonction de densité

```
ggplot(mpg) +  
  geom_density(aes(x=hwy), fill = "skyblue") +  
  labs(  
    title = "Répartition de la consommation sur autoroute (miles par gallon)",  
    x = "Consommation sur autoroute (miles par gallon)",  
    y = "Nombre de véhicule"  
  )  
  ...
```



# Histogramme et fonction de densité



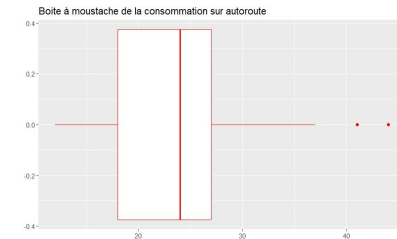
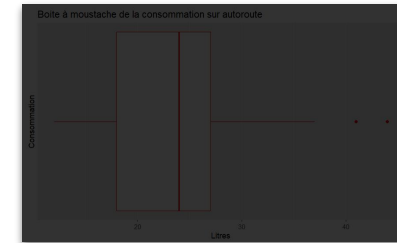
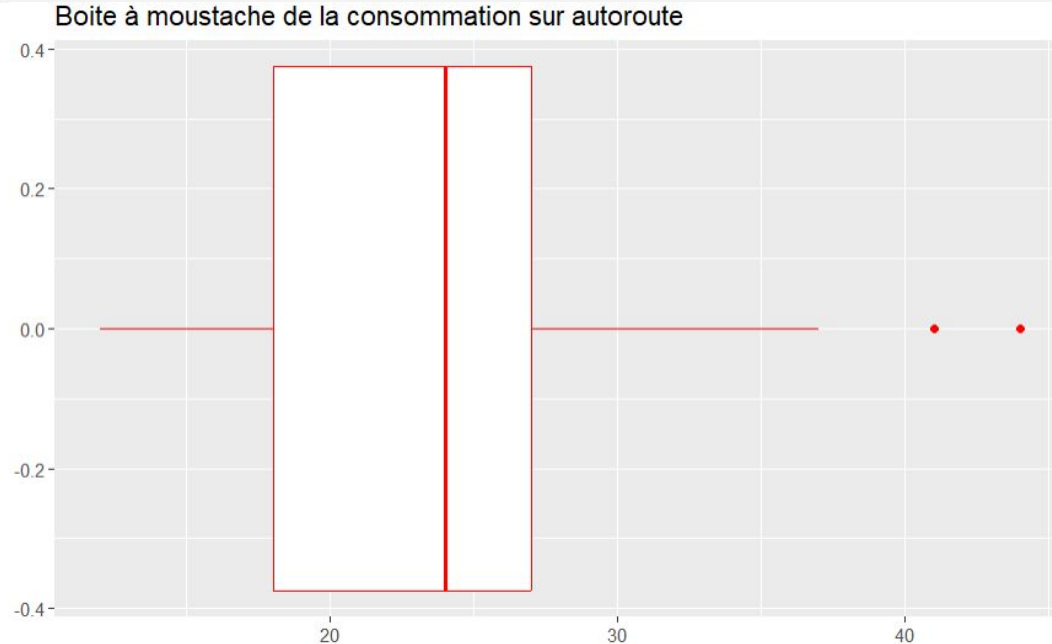
# Histogramme et fonction de densité

```
```{r}
# Largeur des classes pour histogramme
binwidth <- 2

# Histogramme avec densité sur axe secondaire
ggplot(mpg, aes(x = hwy)) +
  geom_histogram(aes(y = ..count..), binwidth = binwidth,
                 fill = "lightblue", color = "black", alpha = 0.6) +
  geom_density(aes(y = ..density.. * nrow(mpg) * binwidth), # conversion pour correspondre aux counts
               color = "red", size = 1.2) +
  scale_y_continuous(
    name = "Nombre de véhicules",
    sec.axis = sec_axis(~ . / (nrow(mpg) * binwidth), name = "Densité")
  ) +
  labs(title = "Nombre de véhicules et densité selon la consommation sur autoroute",
       x = "Consommation sur autoroute (mpg)") +
  theme_minimal()
```
```

# Boite à moustache

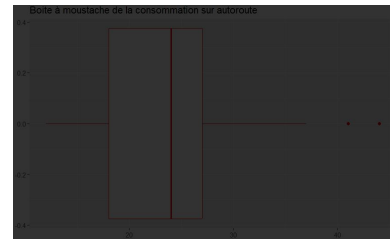
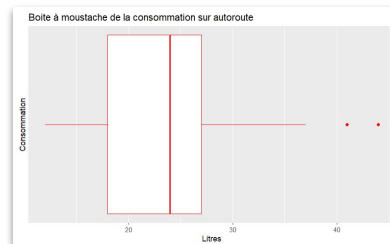
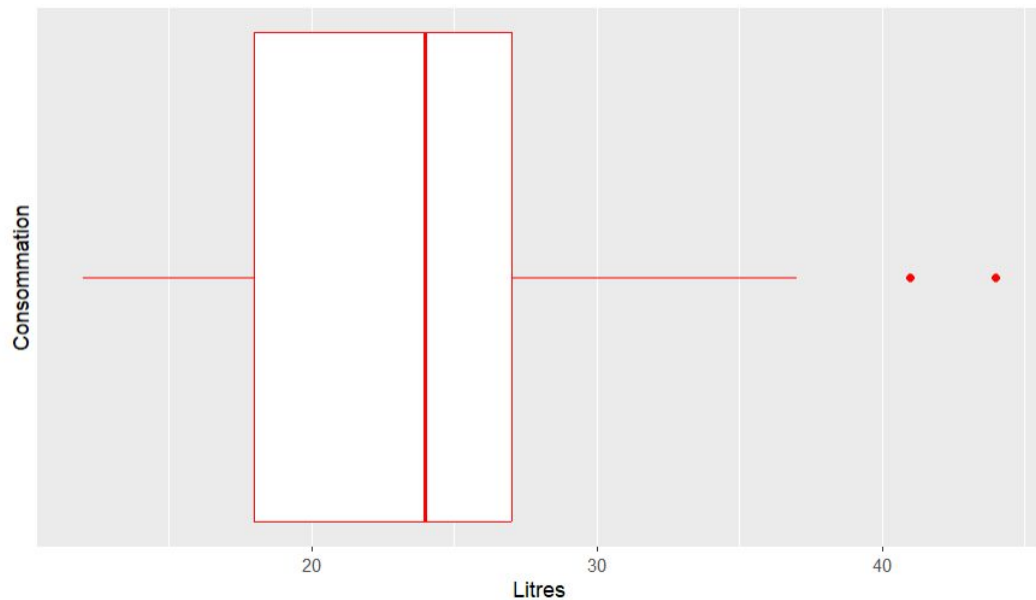
```
ggplot(mpg) +  
  geom_boxplot(aes(x=hwy),color="red") + labs(  
    title = "Boite à moustache de la consommation sur autoroute",  
    x = NULL,  
    y = NULL  
  )
```



# Boite à moustache

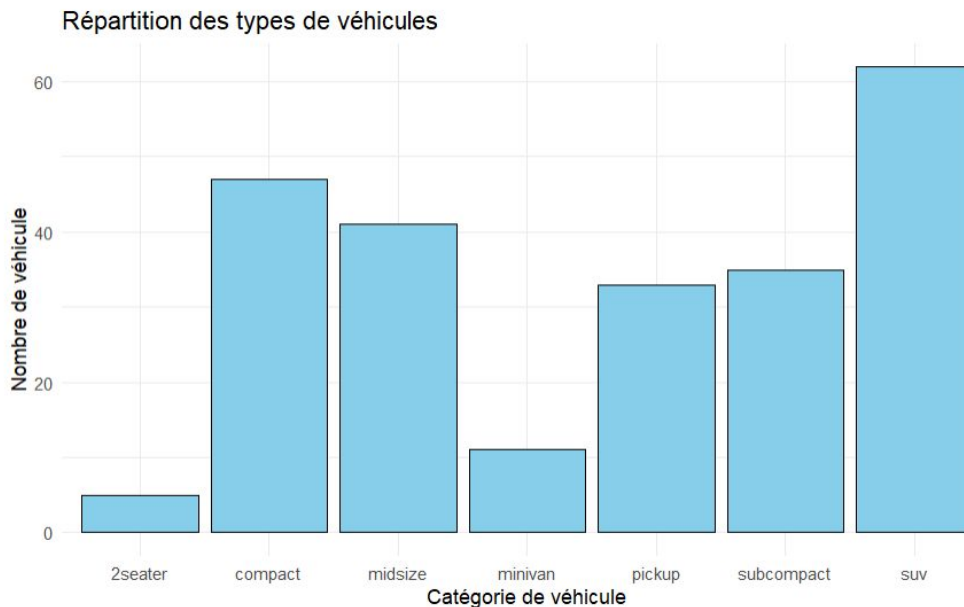
```
ggplot(mpg) +  
  geom_boxplot(aes(x=hwy), color="red") +  
  scale_y_continuous(breaks = NULL) + labs(  
    title = "Boite à moustache de la consommation sur autoroute",  
    x = "Litres",  
    y = "Consommation")
```

Boite à moustache de la consommation sur autoroute



# Graphique en bar pour variable qualitative norminale

```
ggplot(mpg) +  
  geom_bar(aes(x = class), fill = "skyblue", color = "black") +  
  labs(title = "Répartition des types de véhicules",  
        x = "Catégorie de véhicule",  
        y = "Nombre de véhicule") +  
  theme_minimal()  
...
```



# Graphique en bar pour variable qualitative ordinale

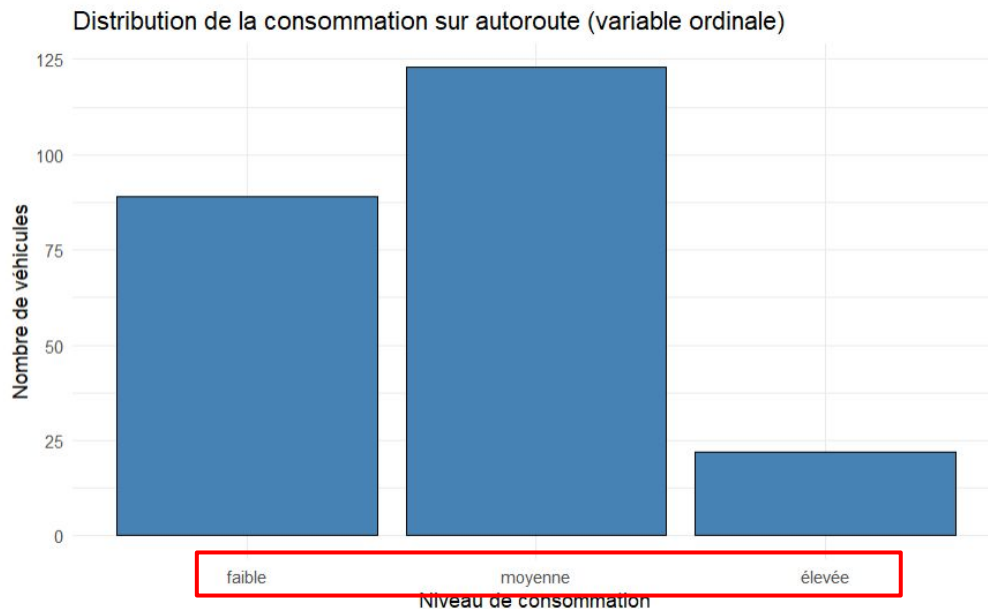
```
# Création d'une variable ordinale (catégories de consommation)  
mpg$conso_cat <- cut(  
  mpg$hwy,  
  breaks = c(0, 20, 30, 45),  
  labels = c("faible", "moyenne", "élevée"),  
  ordered_result = TRUE  
)
```

**L'ordre des modalités d'une variable est à réaliser dans la base de données. Ci-dessous, une autre méthode pour ordonner les modalités**

```
mpg$conso_cat <- factor(mpg$conso_cat,  
  levels = c("faible", "moyenne", "élevée"))
```

# Graphique en bar pour variable qualitative ordinale

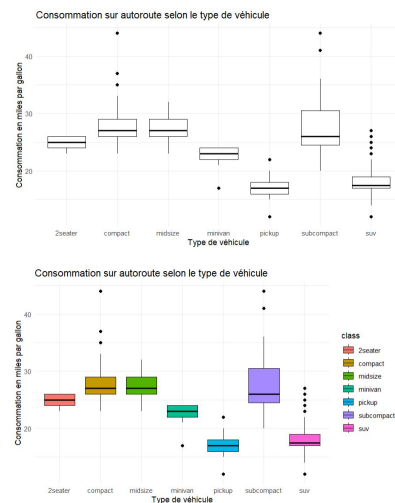
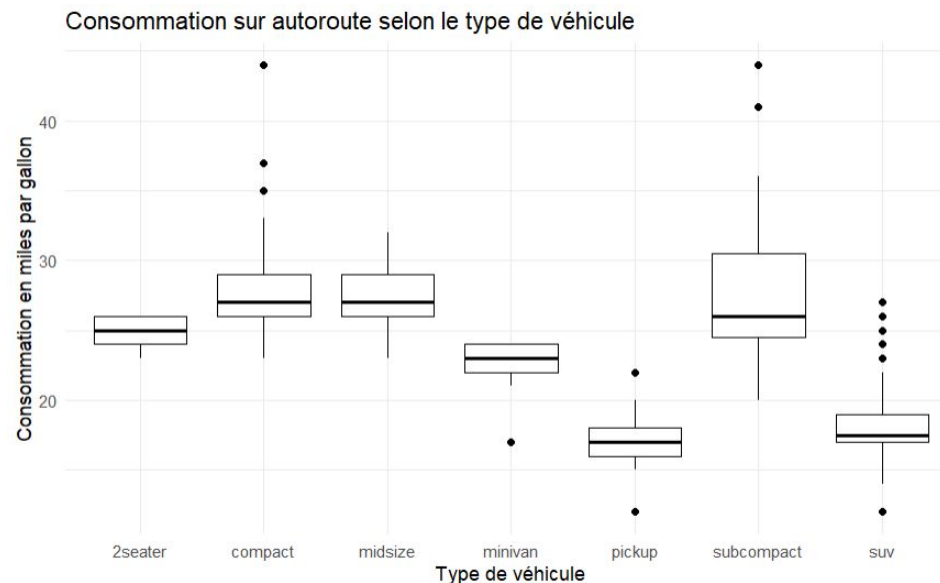
```
# Diagramme en barres ordonné
ggplot(mpg) +
  geom_bar(aes(x = conso_cat), fill = "steelblue", color = "black") +
  labs(title = "Distribution de la consommation sur autoroute (variable ordinale)",
        x = "Niveau de consommation",
        y = "Nombre de véhicules") +
  theme_minimal()
```





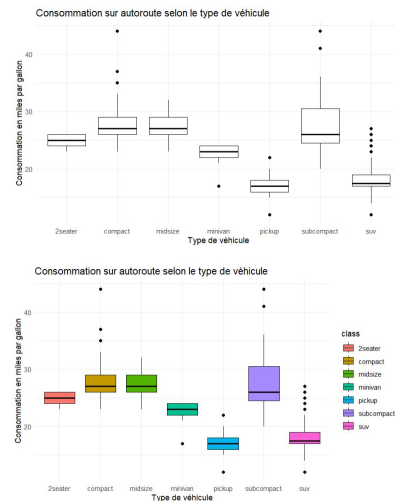
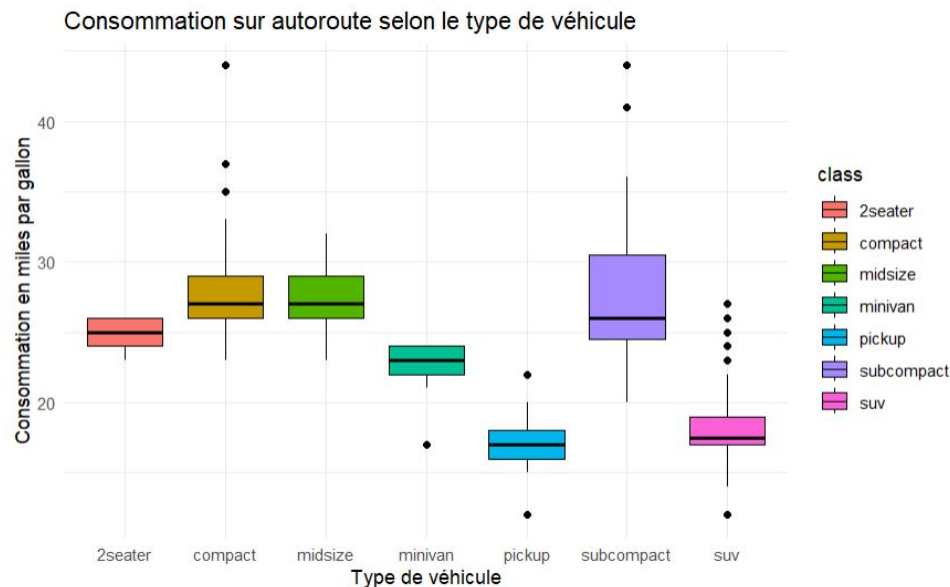
# Boite à moustache : quantitative x qualitative nominale

```
ggplot(mpg) +  
  geom_boxplot(aes(x=class,y=hwy),color="black") +  
  labs(title = "Consommation sur autoroute selon le type de véhicule",  
        x = "Type de véhicule",  
        y = "Consommation en miles par gallon") +  
  theme_minimal()
```



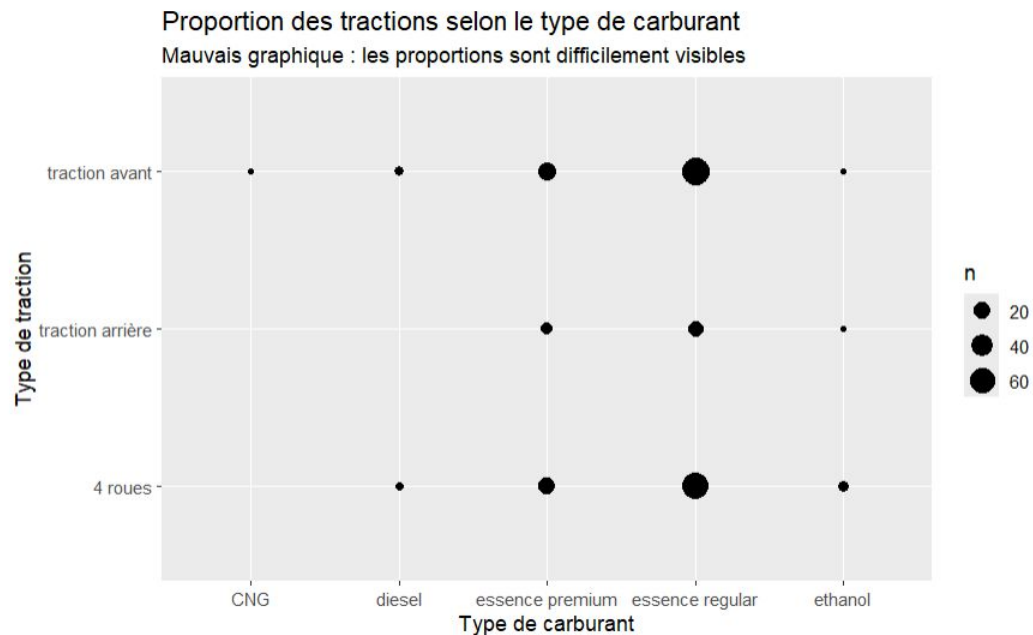
# Boite à moustache : quantitative x qualitative nominale

```
ggplot(mpg) +  
  geom_boxplot(aes(x = class, y = hwy, fill=class), color = "black") +  
  labs(title = "Consommation sur autoroute selon le type de véhicule",  
        x = "Type de véhicule",  
        y = "Consommation en miles par gallon") +  
  theme_minimal()
```



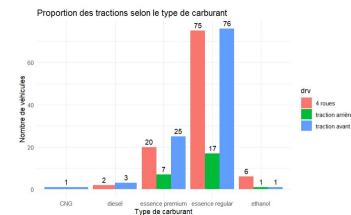
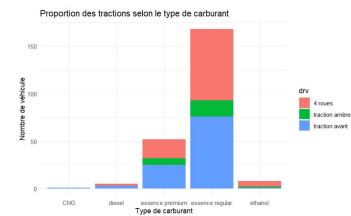
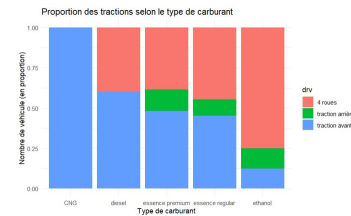
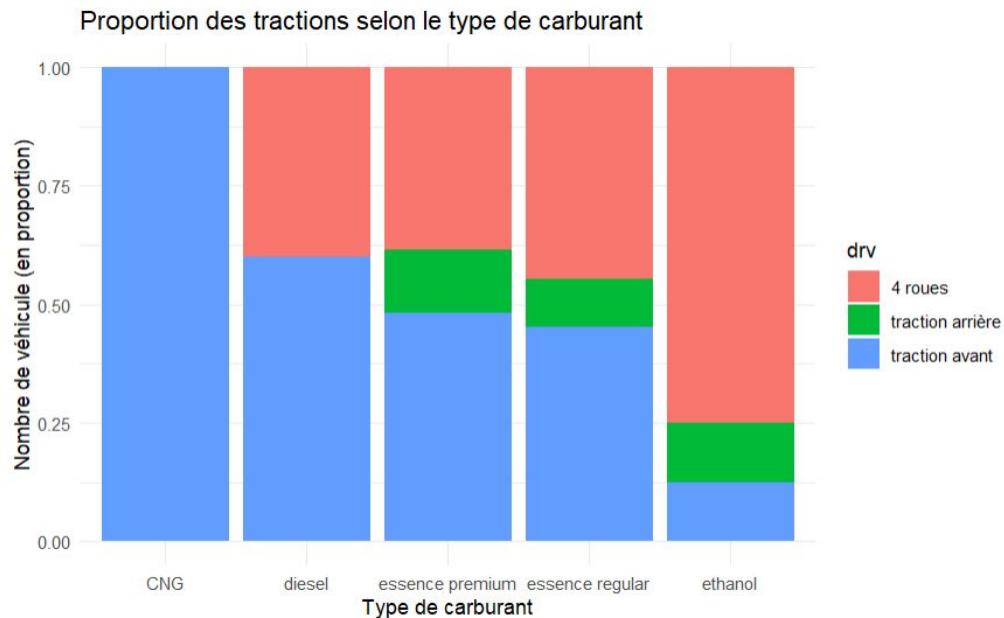
# Graphe pour qualitative nominale x qualitative nominale

```
ggplot(mpg)+  
  geom_count(aes(fl,drv)) +  
  labs(title = "Proportion des tractions selon le type de carburant",  
        subtitle = "Mauvais graphique : les proportions sont difficilement visibles",  
        x = "Type de carburant",  
        y = "Type de traction")
```



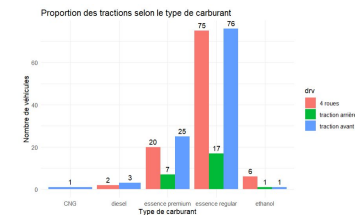
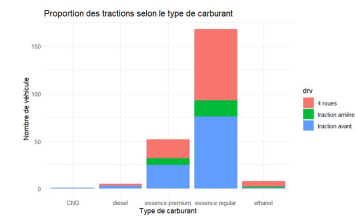
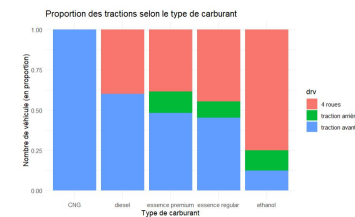
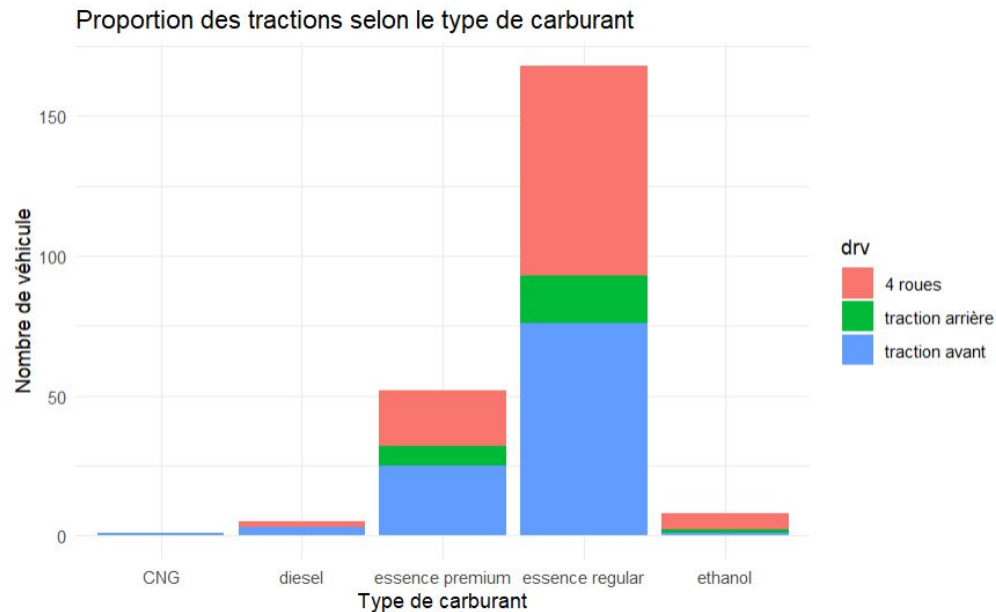
# Graphe pour qualitative nominale x qualitative nominale

```
ggplot(mpg) +  
  geom_bar(aes(f1, fill=drv), position="fill") +  
  labs(  
    title = "Proportion des tractions selon le type de carburant",  
    y = "Nombre de véhicule (en proportion)",  
    x = "Type de carburant"  
  ) +  
  theme_minimal()
```



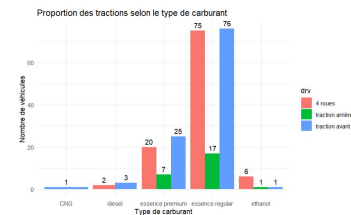
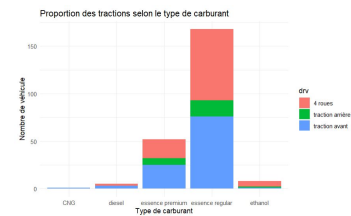
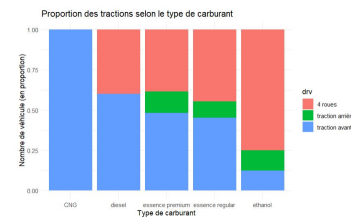
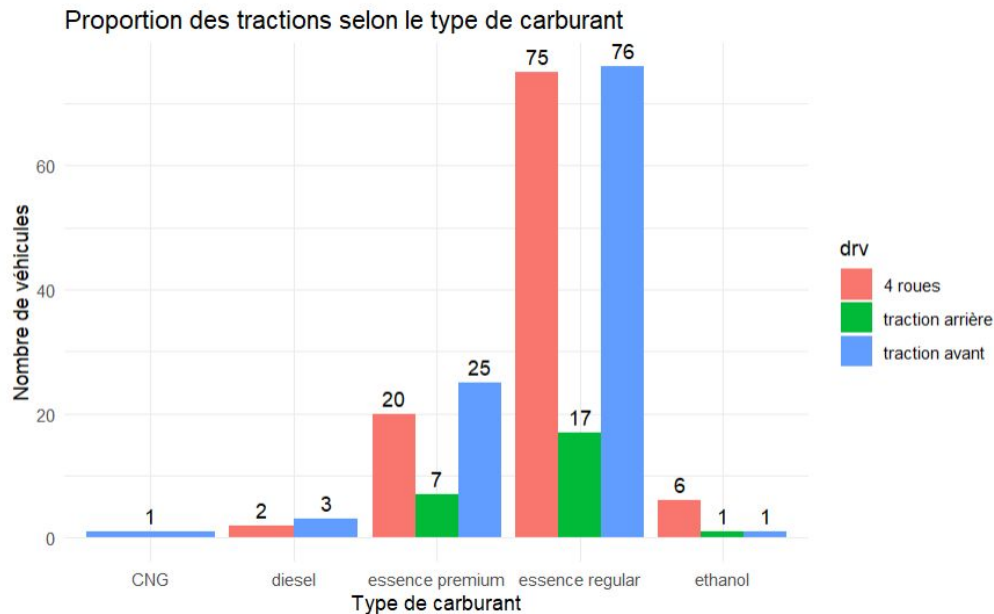
# Graphe pour qualitative nominale x qualitative nominale

```
ggplot(mpg) +  
  geom_bar(aes(f1, fill=drv)) +  
  labs(  
    title = "Proportion des tractions selon le type de carburant",  
    y = "Nombre de véhicule",  
    x = "Type de carburant"  
  ) +  
  theme_minimal()
```



# Graphe pour qualitative nominale x qualitative nominale

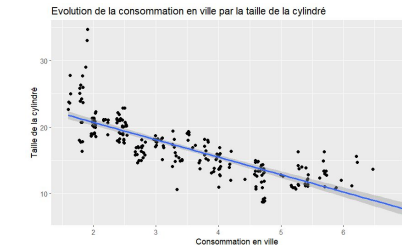
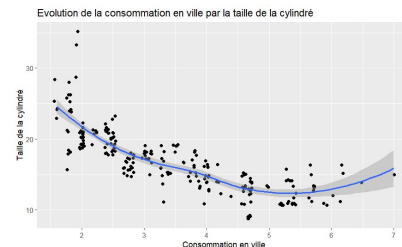
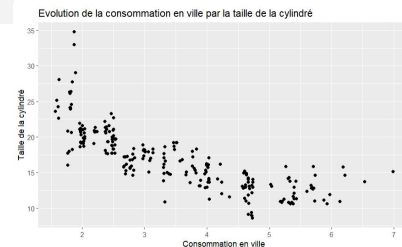
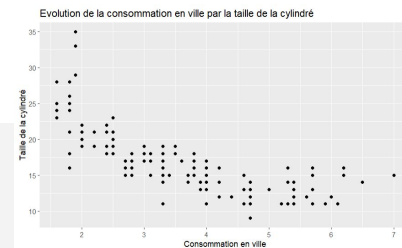
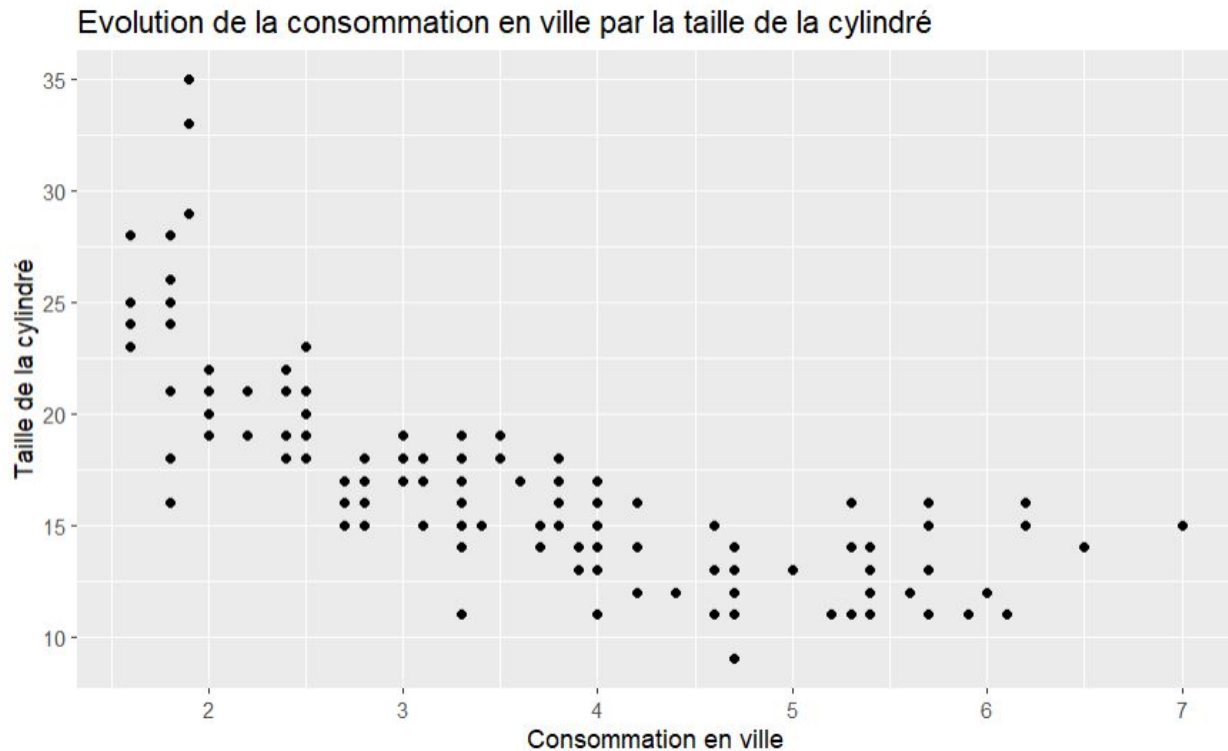
```
dodge <- position_dodge(width = 0.9) # largeur du dodge pour aligner labels
ggplot(mpg, aes(x = fl, fill = drv)) +
  geom_bar(position = dodge) +
  geom_text(stat = "count", aes(label = ..count..),
            position = dodge, vjust = -0.5) +
  labs(
    title = "Proportion des tractions selon le type de carburant",
    y = "Nombre de véhicules",
    x = "Type de carburant"
  ) +
  theme_minimal()
```



Graphe pour quantitative x quantitative

# Geom\_Point : Points se superposent

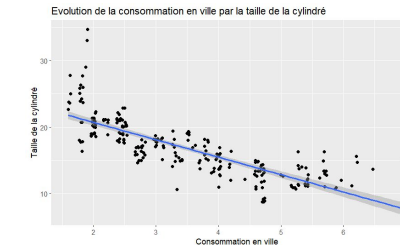
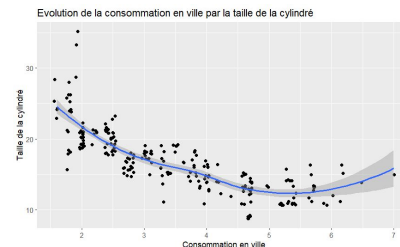
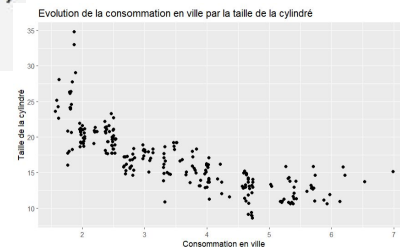
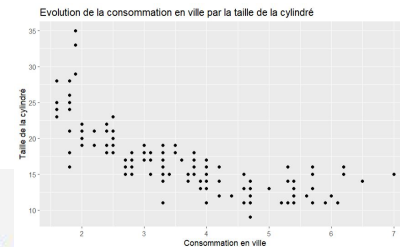
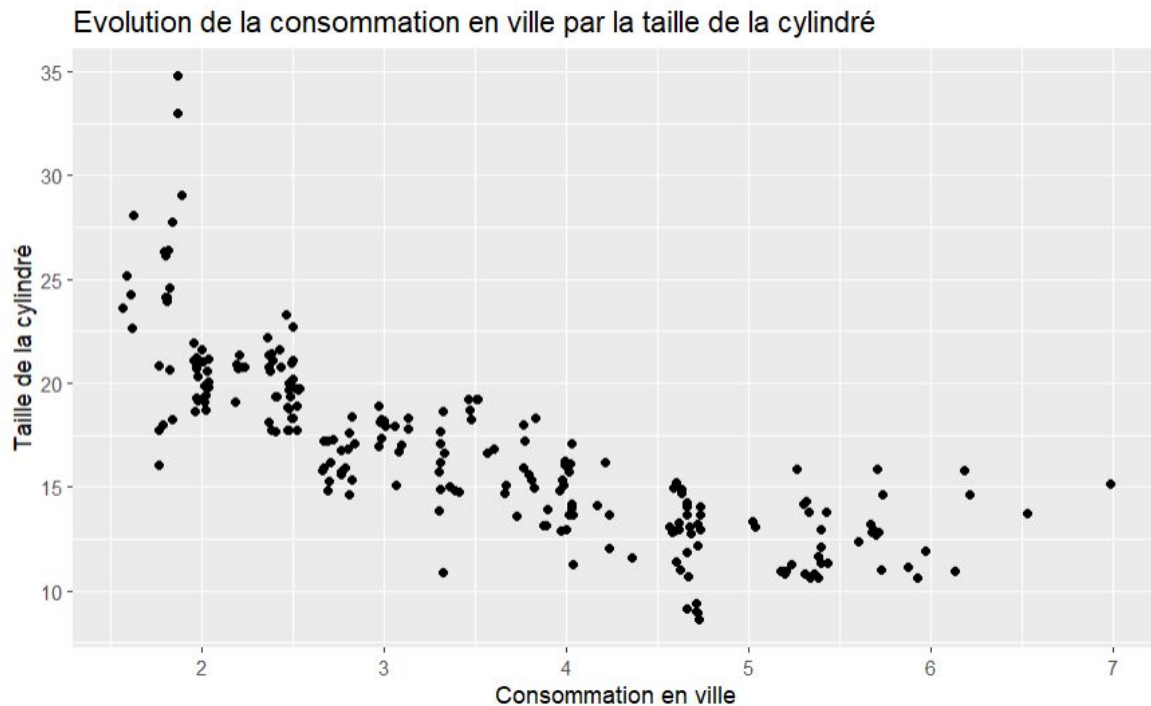
```
ggplot(mpg)+  
  geom_point(aes(displ,cty)) +  
  labs(title = "Evolution de la consommation en ville par la taille de la cylindr ",  
        x = "Consommation en ville",  
        y = "Taille de la cylindr ")
```





# Geom\_Jitter : Petite dispersion aléatoire des points

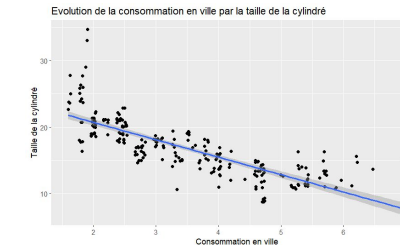
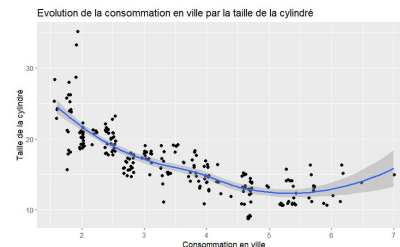
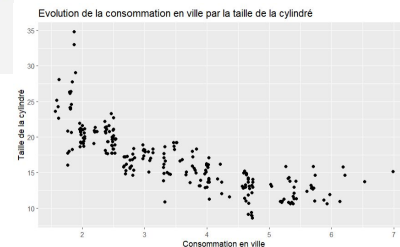
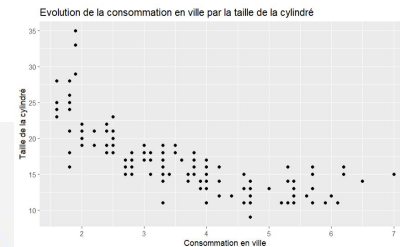
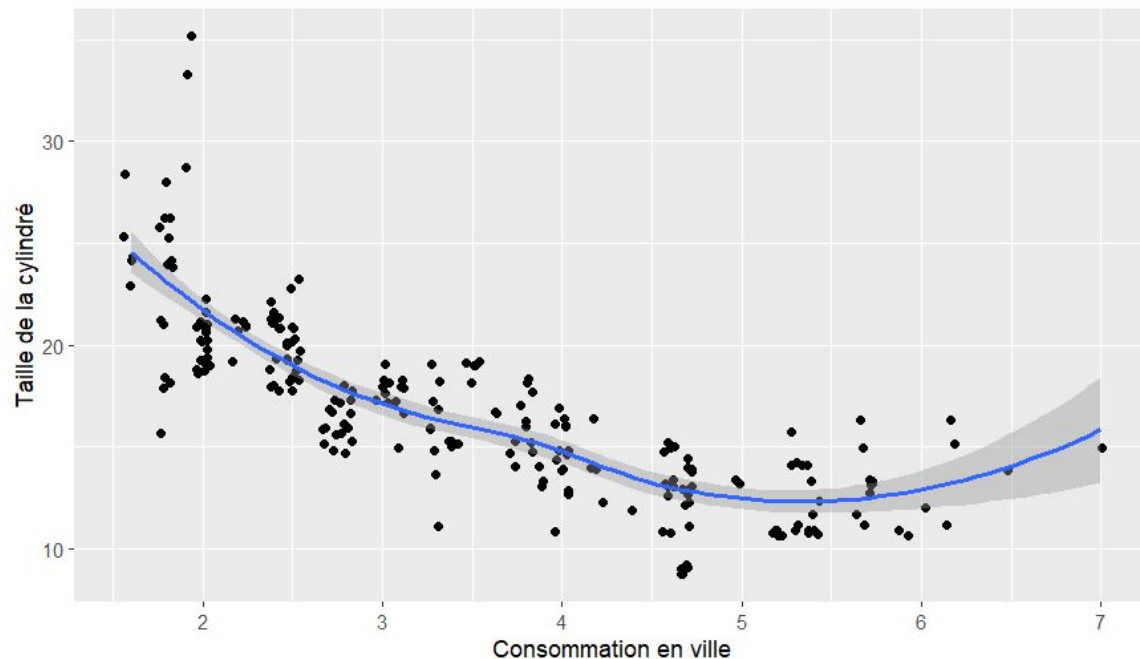
```
ggplot(mpg)+  
  geom_jitter(aes(displ,cty))+  
  labs(title = "Evolution de la consommation en ville par la taille de la cylindr ",  
        x = "Consommation en ville",  
        y = "Taille de la cylindr ")
```



# Courbe de tendance par régression locale

```
ggplot(mpg, aes(displ, cty)) +  
  geom_jitter() +  
  geom_smooth() +  
  labs(title = "Evolution de la consommation en ville par la taille de la cylindr ",  
        x = "Consommation en ville",  
        y = "Taille de la cylindr ")
```

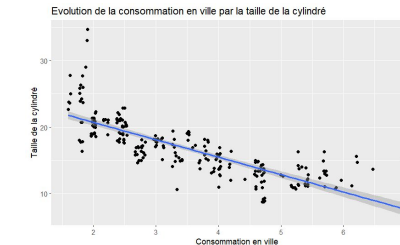
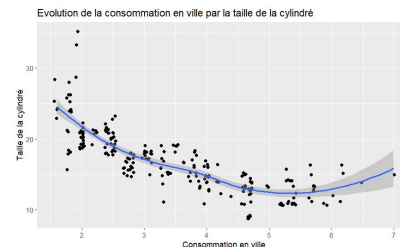
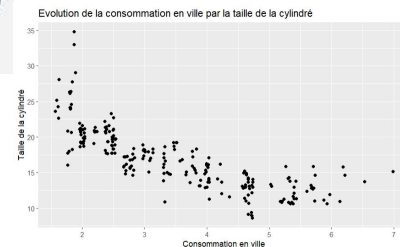
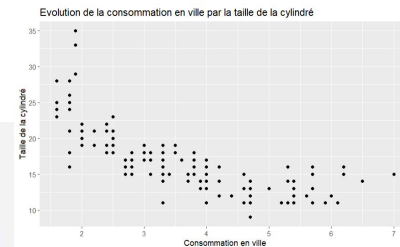
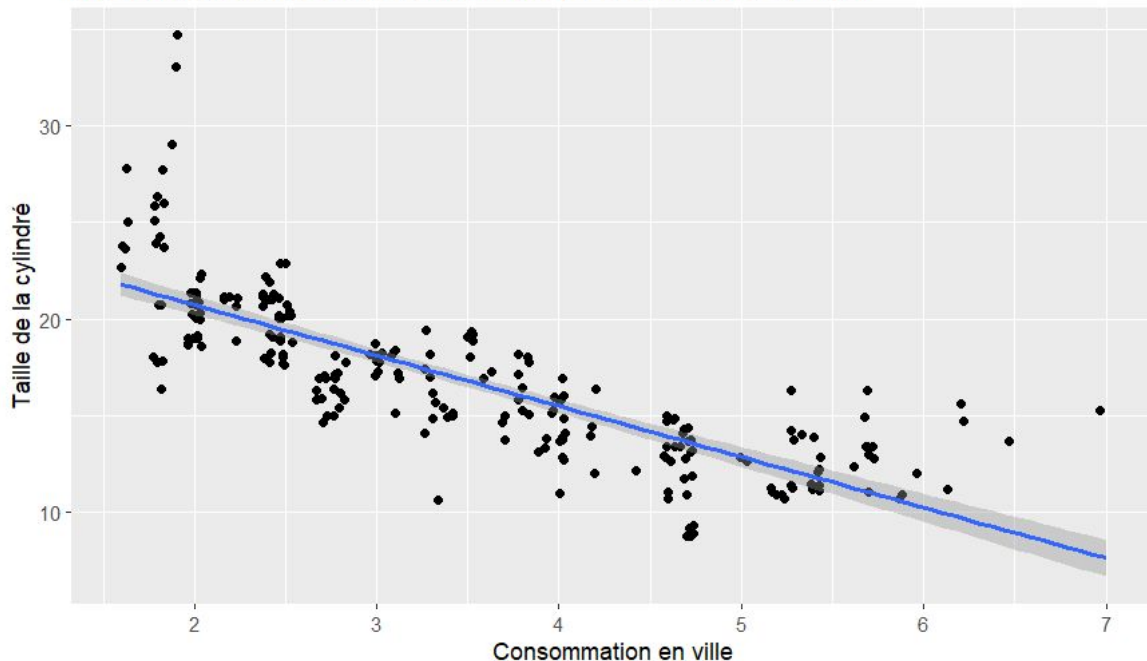
Evolution de la consommation en ville par la taille de la cylindr 



# Courbe de tendance par régression linéaire

```
ggplot(mpg, aes(displ, cty)) +  
  geom_jitter() +  
  geom_smooth(method="lm") +  
  labs(title = "Evolution de la consommation en ville par la taille de la cylindrée",  
        x = "Consommation en ville",  
        y = "Taille de la cylindrée")
```

Evolution de la consommation en ville par la taille de la cylindrée



Merci pour votre attention

# GGPLOT2

TP Etude des ménages