Classification automatique

TP 2

Clustering sur données continues

On dipose du jeu de données suivant, l'objectif est d'effectuer une classificatuon des types de véhicules en fonction de leur motorisation.

```
xtable(mtcars, auto = TRUE)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

1. Data: stats descriptives

```
#strucutre du fichier et nature des variables
z <- data.frame(variable = names(mtcars),
           classe = sapply(mtcars, typeof),
           first_values = sapply(mtcars, function(x) paste0(head(x),  collapse = ", ")),
```

```
            distinct_values = sapply(mtcars, function(x) paste0(length(unique(x)))),
            row.names = NULL)
z
```

variable	classe	first_values	distinct_values
mpg	double	21, 21, 22.8, 21.4, 18.7, 18.1	25
cyl	double	6, 6, 4, 6, 8, 6	3
disp	double	160, 160, 108, 258, 360, 225	27
hp	double	110, 110, 93, 110, 175, 105	22
drat	double	3.9, 3.9, 3.85, 3.08, 3.15, 2.76	22
wt	double	2.62, 2.875, 2.32, 3.215, 3.44, 3.46	29
qsec	double	16.46, 17.02, 18.61, 19.44, 17.02, 20.22	30
vs	double	0, 0, 1, 1, 0, 1	2
am	double	1, 1, 1, 0, 0, 0	2
gear	double	4, 4, 4, 3, 3, 3	3
carb	double	4, 4, 1, 1, 2, 1	6

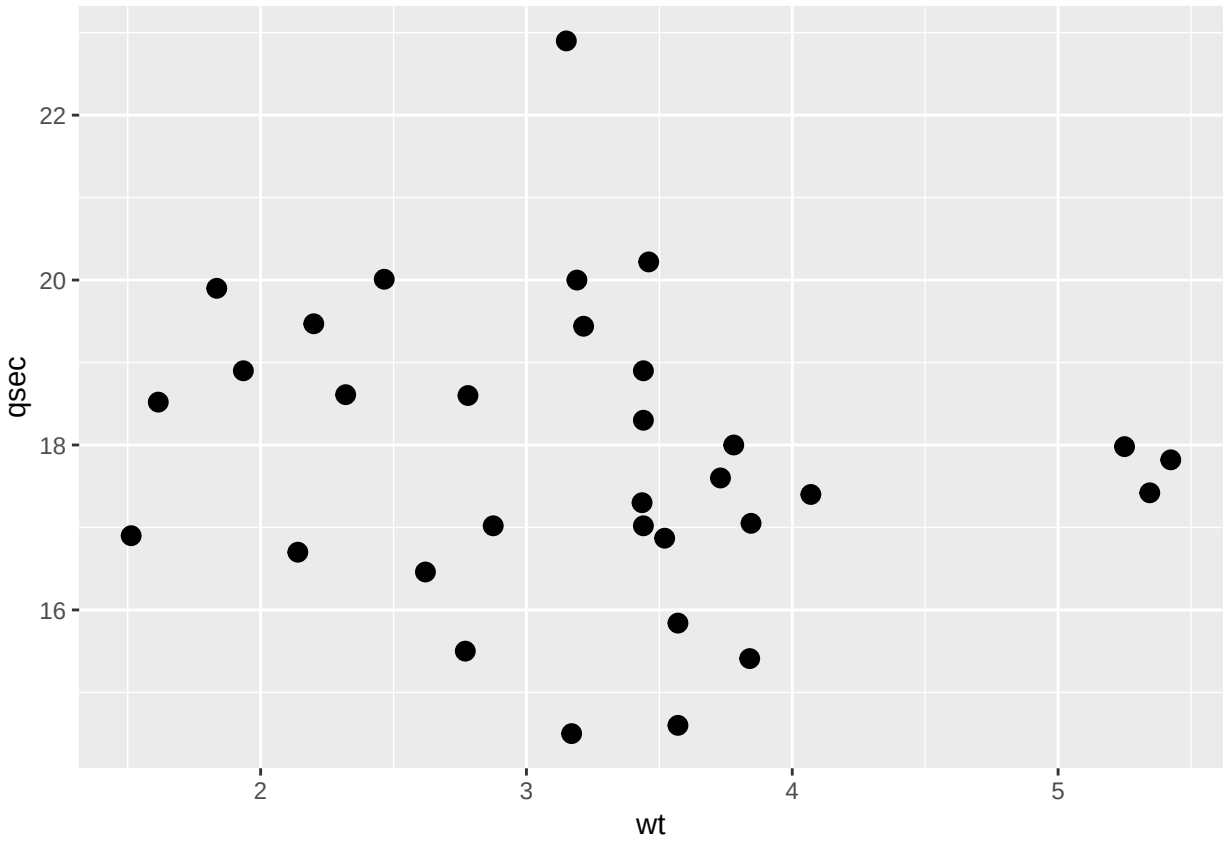
```
#statistiques élémentaires
dtf <- round(sapply(mtcars, each(min, max, mean, sd, var, median, IQR)),3)

xtable(dtf,digits = 2)
```

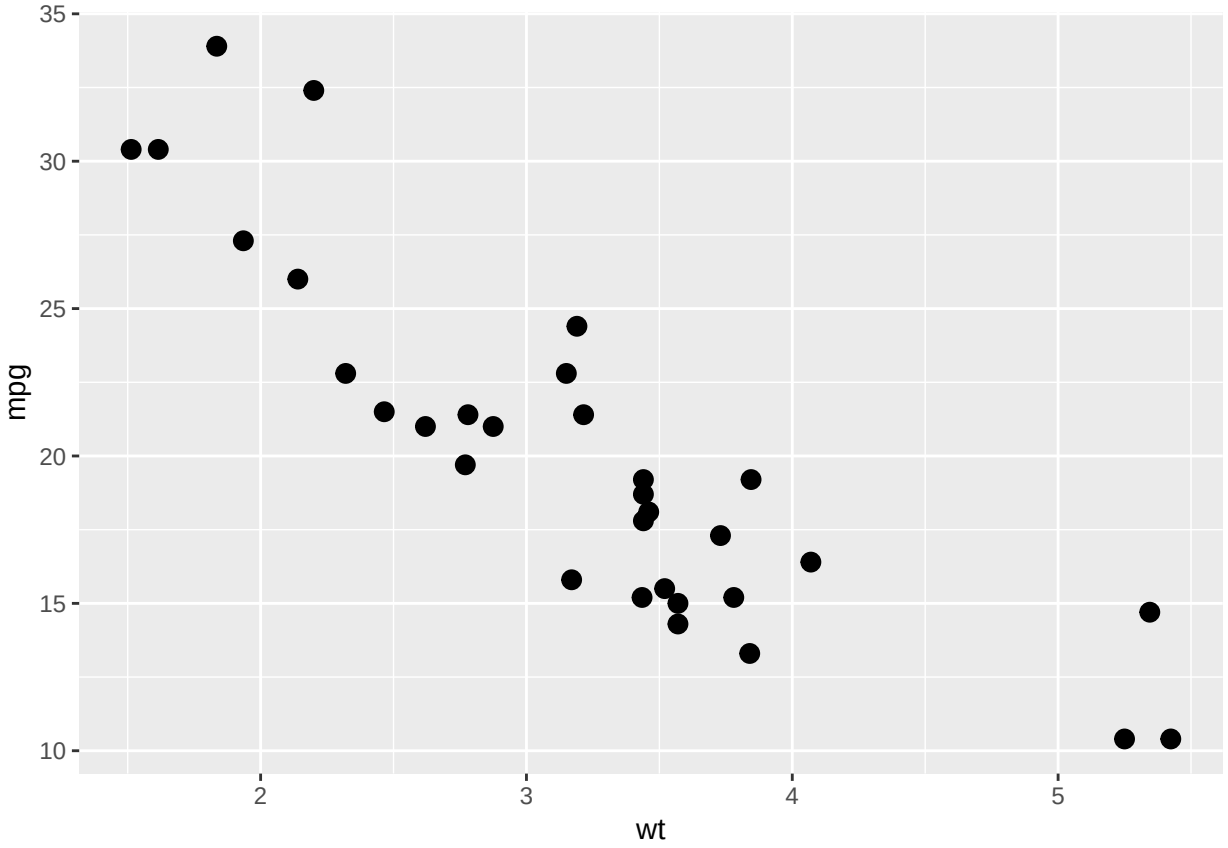
	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
min	10.400	4.000	71.100	52.000	2.760	1.513	14.500	0.000	0.000	3.000	1.000
max	33.900	8.000	472.000	335.000	4.930	5.424	22.900	1.000	1.000	5.000	8.000
mean	20.091	6.188	230.722	146.688	3.597	3.217	17.849	0.438	0.406	3.688	2.812
sd	6.027	1.786	123.939	68.563	0.535	0.978	1.787	0.504	0.499	0.738	1.615
var	36.324	3.190	15360.800	4700.867	0.286	0.957	3.193	0.254	0.249	0.544	2.609
median	19.200	6.000	196.300	123.000	3.695	3.325	17.710	0.000	0.000	4.000	2.000
IQR	7.375	4.000	205.175	83.500	0.840	1.029	2.008	1.000	1.000	1.000	2.000

2. Visualitation en fonction des variables wt et qsec.

```
g <- ggplot(mtcars,aes(x = wt,y = qsec)) +
       geom_point(size = 3)
g
```




```
g1 <- ggplot(mtcars,aes(x = wt,y = mpg)) +
       geom_point(size = 3)
g1
```



Les illustrations suivant les axes ci-dessus laissent penser qu'une segmentation en deux ou trois classes semble appropriée.

3. K-means: méthode de mac queen

```
# kmeans avec les attributs wt et qsec k=3

cars <- mtcars
kmeans.1 <- kmeans(cbind(cars$wt,cars$qsec), centers=3, algorithm=c("MacQueen"))

kmeans.1
```

```
## K-means clustering with 3 clusters of sizes 5, 15, 12
##
## Cluster means:
##       [,1]     [,2]
## 1 3.384000 15.17000
## 2 3.628467 17.32267
## 3 2.633750 19.62250
##
## Clustering vector:
##  [1] 2 2 3 3 2 3 1 3 3 2 3 2 2 2 2 2 2 3 3 3 3 2 2 1 2 3 2 2 1 1 1 3
##
## Within cluster sum of squares by cluster:
## [1]  2.08912 21.42988 20.60248
##  (between_SS / total_SS =  65.7 %)
##
```

```
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
cars$cluster <- factor(kmeans.1$cluster)
cars$cyl     <- factor(cars$cyl, labels =  c('Four cylinder', 'Six cylinder', 'Eight cylinder'))

# Centres des classes
centres  <- data.frame(cluster = factor(seq(1:3)),kmeans.1$centers)
colnames(centres) <- c("cluster","wt","qsec")
centres
```

cluster	wt	qsec
1	3.384000	15.17000
2	3.628467	17.32267
3	2.633750	19.62250

```
# Variances inter-classes et intra-classe
var_tot   <-kmeans.1$totss
var_intra <-kmeans.1$tot.withinss
var_inter <-kmeans.1$betweenss
print(paste("Variance totale: ",var_tot))
```

```
## [1] "Variance totale:  128.666898"
```

```
print(paste("Variance inter: ",var_inter))
```

```
## [1] "Variance inter:  84.5454136833333"
```

```
print(paste("Variance intra: ",var_intra))
```

```
## [1] "Variance intra:  44.1214843166667"
```

```
var_inter+var_intra
```

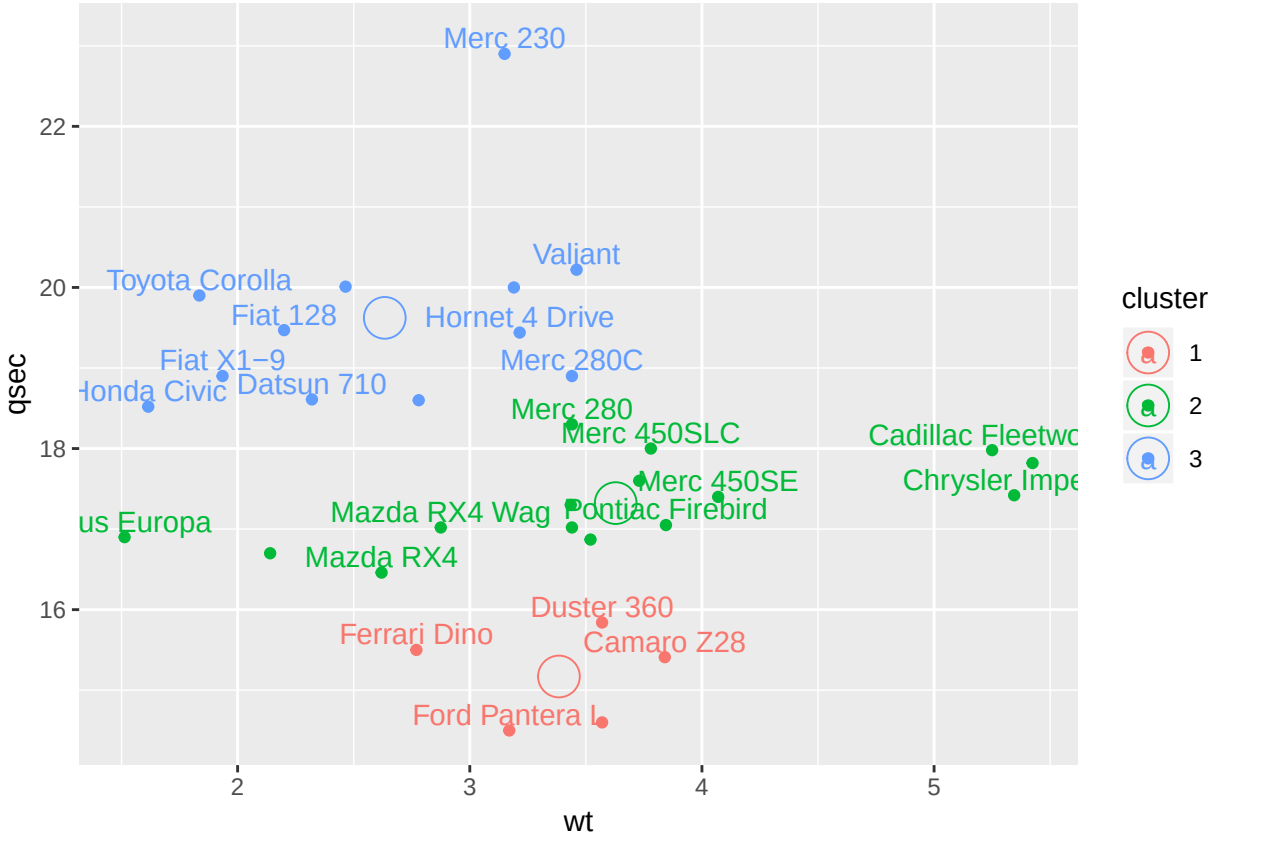
```
## [1] 128.6669
```

```
#Calculer le R2.
R2=var_inter/var_tot

# Visualiser le nuage de points, en affectant á chaque individu une couleur propre à sa classe
#d'affectation, les centres des classes devront se disntinguer des autres points.

gg <- ggplot(cars, aes(x = wt,y = qsec,color=cluster))+
  geom_point() +
  geom_text(data = cars,
            aes(x = wt,
                y = qsec,
                label = row.names(cars),
                color = cluster),
            nudge_y = .2,
            check_overlap = TRUE) +
  geom_point(data = centres,
             mapping = aes(x = wt,
```

```
                        y = qsec,
                        color = cluster),
            size = 7,
            pch = 1)
gg
```




```
#Donner la matrice de confusion et commenter les résultats obtenus.(NE pas en tenir compte)


xtable(table(cars$cluster, cars$cyl))
```

Four cylinder	Six cylinder	Eight cylinder
0	1	4
2	3	10
9	3	0

```
#Décrire et analyser les classes obtenues.
```

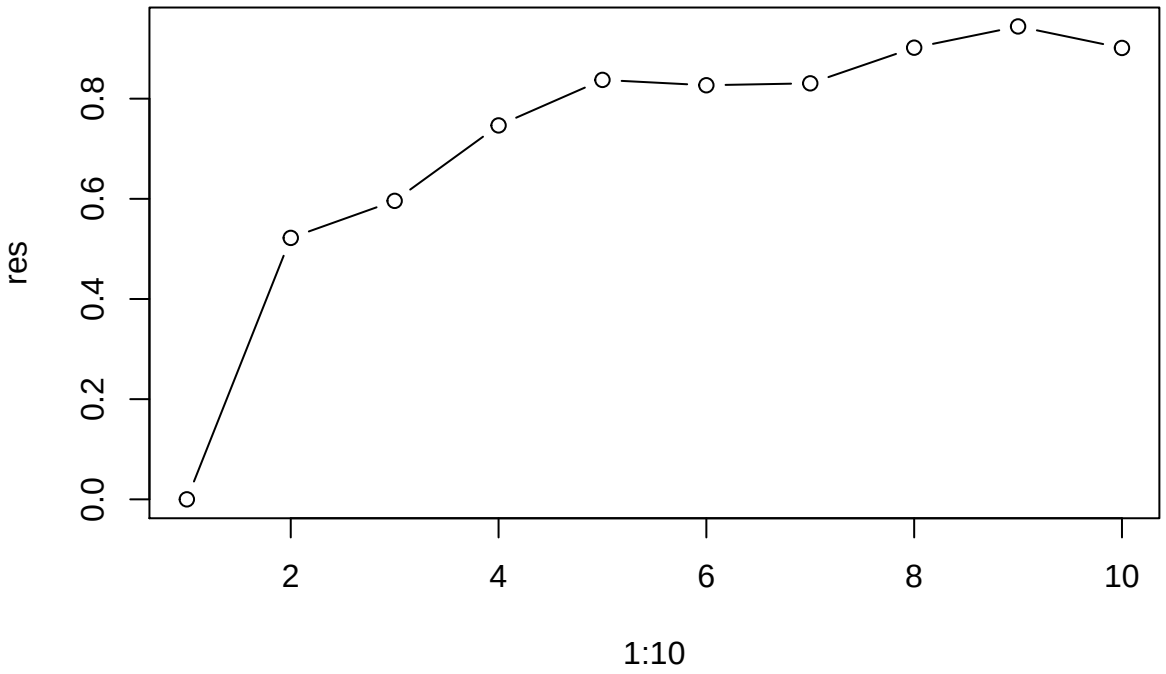
La segmentation obenue distingue les véhicules à faible motorisation(classe 3) des véhicule à forte motorisation(classe 2).

4 cf. deuxième partie question 5

5

```
res <- c()
for (i in 1:10) {
  restmp=kmeans(cbind(cars$wt,cars$qsec),centers=i,algorithm="MacQueen")
 res[i]=restmp$betweenss/restmp$totss
}

plot(1:10,res,type='b')
```




```
n_class <- 3
cars <- mtcars
kmeans.2 <- kmeans(cbind(cars$wt,cars$mpg), centers=n_class, algorithm=c("MacQueen"))
kmeans.2
```

```
## K-means clustering with 3 clusters of sizes 5, 14, 13
##
## Cluster means:
##       [,1]     [,2]
## 1 1.819600 30.88000
## 2 3.971714 14.95714
## 3 2.942308 21.46923
##
## Clustering vector:
##  [1] 3 3 3 3 3 2 2 3 3 3 2 2 2 2 2 2 2 1 1 1 3 2 2 2 3 1 3 1 2 3 2 3
##
## Within cluster sum of squares by cluster:
## [1] 25.00212 79.18771 57.12002
##  (between_SS / total_SS =  86.0 %)
##
## Available components:
```

```
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
cars$cluster <- factor(kmeans.2$cluster)
cars$cyl     <- factor(cars$cyl, labels =  c('Four cylinder', 'Six cylinder', 'Eight cylinder'))

# Centres des classes
centres  <- data.frame(cluster = factor(seq(1:n_class)),kmeans.2$centers)
colnames(centres) <- c("cluster","wt","qsec")
centres
```

cluster	wt	qsec
1	1.819600	30.88000
2	3.971714	14.95714
3	2.942308	21.46923

```
# Variances inter-classes et intra-classe
var_tot   <-kmeans.2$totss
var_intra <-kmeans.2$tot.withinss
var_inter <-kmeans.2$betweenss
print(paste("Variance totale: ",var_tot))
```

```
## [1] "Variance totale:  1155.7259355"
```

```
print(paste("Variance inter: ",var_inter))
```

```
## [1] "Variance inter:  994.416078651648"
```

```
print(paste("Variance intra: ",var_intra))
```

```
## [1] "Variance intra:  161.309856848352"
```

```
var_inter+var_intra
```

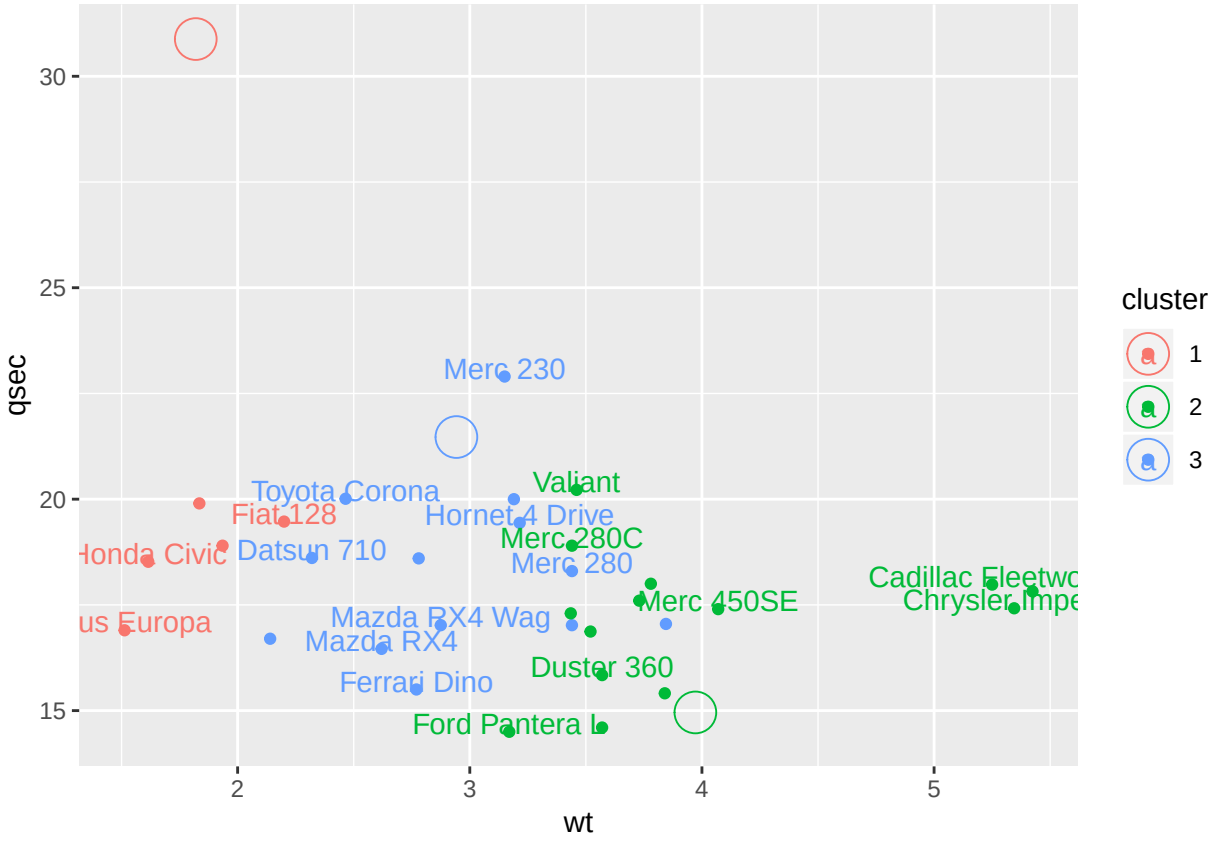
```
## [1] 1155.726
```

```
#Calculer le R2.
R2=var_inter/var_tot

# Visualiser le nuage de points, en affectant á chaque individu une couleur propre à sa classe
#d'affectation, les centres des classes devront se disntinguer des autres points.

gg <- ggplot(cars, aes(x = wt,y = qsec,color=cluster))+
  geom_point() +
  geom_text(data = cars,
            aes(x = wt,
                y = qsec,
                label = row.names(cars),
                color = cluster),
            nudge_y = .2,
            check_overlap = TRUE) +
  geom_point(data = centres,
             mapping = aes(x = wt,
                           y = qsec,
```

```
                          color = cluster),
             size = 7,
             pch = 1)
gg
```




```
xtable(table(cars$cluster, cars$cyl))
```

Four cylinder	Six cylinder	Eight cylinder
5	0	0
0	2	12
6	5	2

6)

```
cars <- mtcars
k7=kmeans(cbind(cars$wt,cars$mpg),3,nstart=30,algorithm="MacQueen")
k7
```

```
## K-means clustering with 3 clusters of sizes 14, 12, 6
##
## Cluster means:
##       [,1]     [,2]
## 1 3.072143 20.64286
## 2 4.058667 14.45833
## 3 1.873000 30.06667
##
```

```
## Clustering vector:
##  [1] 1 1 1 1 1 1 2 1 1 1 1 2 2 2 2 2 2 3 3 3 1 2 2 2 1 3 3 3 2 1 2 1
##
## Within cluster sum of squares by cluster:
## [1] 51.48262 57.60729 44.93300
##  (between_SS / total_SS =  86.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
km.forgy=kmeans(cbind(cars$wt,cars$mpg),3,algorithm="Forgy")
km.forgy
```

```
## K-means clustering with 3 clusters of sizes 12, 14, 6
##
## Cluster means:
##       [,1]     [,2]
## 1 4.058667 14.45833
## 2 3.072143 20.64286
## 3 1.873000 30.06667
##
## Clustering vector:
##  [1] 2 2 2 2 2 2 1 2 2 2 2 1 1 1 1 1 1 3 3 3 2 1 1 1 2 3 3 3 1 2 1 2
##
## Within cluster sum of squares by cluster:
## [1] 57.60729 51.48262 44.93300
##  (between_SS / total_SS =  86.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
km.nuee=kmeans(cbind(cars$wt,cars$mpg),3)
km.nuee
```

```
## K-means clustering with 3 clusters of sizes 14, 6, 12
##
## Cluster means:
##       [,1]     [,2]
## 1 3.072143 20.64286
## 2 1.873000 30.06667
## 3 4.058667 14.45833
##
## Clustering vector:
##  [1] 1 1 1 1 1 1 3 1 1 1 1 3 3 3 3 3 3 2 2 2 1 3 3 3 1 2 2 2 3 1 3 1
##
## Within cluster sum of squares by cluster:
## [1] 51.48262 44.93300 57.60729
##  (between_SS / total_SS =  86.7 %)
##
## Available components:
```

```
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```