REPUBLIQUE DU SENEGAL

un peuple-un but-une foi

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR DE LA RECHERCHE ET DE L'INNOVATION

UNIVERSITE DE THIES

UFR DES SCIENCES ECONOMIQUES ET SOCIALES / SCIENCES ET TECHNOLOGIE

DEPARTEMENT MATHEMATIQUES

****

****

MASTER I EN SCIENCES DES DONNEES ET APPLICATIONS / OPTION STATISTIQUES- ECONOMETRIE / INTELLIGENCE ARTIFICIELLE

THEME : PROJET D'ANALYSE DES DONNEES

Annee Academies: 2019-2020

Présenté par :

DJIBRIL DIALLO
SEYNI KAIRE
KHOUDIA MBODJI
MATHIAM FAYE
CHEIKH MBACKE DIOUF

Encadré par :

DR Mme. SALL

SOMMAIRE

INTRODUCTION

L'analyse des données (aussi appelée **analyse exploratoire des** données ou **AED**) est une famille de méthodes statistiques dont les principales caractéristiques sont d'être multidimensionnelles et descriptives.
Dans cette perspective, nous allons étudier une base de données dénommée CEREALES en format CSV en régression simple et multiple comme indiqué dans notre sommaire. Il sera objet ainsi de faire le calcul des estimateurs, des coefficients de corrélation, établir le tableau ANOVA, de détecter les points influents pour en faire des tests et en déduire des prévisions selon des intervalles de confiance. Il est important de noter que contrairement à la régression simple, la régression multiple sera beaucoup plus approfondie car il sera question de tenir en compte du type de variable **(catégoriel comme dans notre cas ici)** et voir si ces dernières ne sont pas fortement corrélées entre elles **(problème de multi colinéarité)** dont le but sera de choisir le meilleur modèle possible par le biais d'une sélection de variables.

THEME 1 REGRESSION SIMPLE (RAPPEL ET THEORIE)

Définition 1 (Modèle de régression linéaire simple) Un modèle de régression linéaire simple est défini par une équation de la forme :

$\forall i \in \{1,...,n\} \qquad y_i = \beta_1 + \beta_2 x_i + \varepsilon_i$

Les quantités ε_i viennent du fait que les points ne sont jamais parfaitement alignés sur une droite. On les appelle les erreurs (ou bruits) et elles sont supposées aléatoires. Pour pouvoir dire des choses pertinentes sur ce modèle, il faut néanmoins imposer des hypothèses les concernant. Voici celles que nous ferons dans un premier temps

(H) $(H_1) : [\varepsilon_i] = 0$ pour tout indice i $(H_2) : Cov(\varepsilon_i,\varepsilon_j) = \delta_{ij}\sigma^2$ pour tout couple (i,j)

- **IMPORTONS NOTRE BASE DE DONNEES**

```
> library(readr)
```

> cereales <- read_csv("Desktop/PROJET ANALYSE DES DONNEES/ProjetàRendre2021/cereales.csv")

Notre base de données est composée de 77 observations pour 16 variables

La commande de sortie ci-dessous nous en donne la certitude.

> dim(cereales)
[1] 77 16

1 – CALCUL DES ESTIMATEURS

Vu que nous sommes dans le cadre d'une régression simple, le choix de 2 variables au maximum nous est imposé.

Nous allons choisir comme variable endogène Y **(sugars)** et la variable exogène X **(sodium)**

La modélisation de notre problème reviendra à faire la régression suivante :

$$Sugars = B_O + B_1(sodium) + E_i$$

Avec Y= sugars
X= sodium
E_i= erreur ou bruit
B_0 et B_1 sont les paramètres à estimer

Avant de commencer notre régression, nous devons affecter Y aux données de la variables sugars et X aux données de la variable sodium. Cette affectation est faisable grâce aux commandes suivantes :

> Y<- cereales$sugars
> X<- cereales$sodium

Maintenant nous pouvons procéder au calcul des estimateurs en établissant notre régression que nous appellerons **a1**

> a1<-lm(Y~x, data= cereales)

> print(a1)

Call:
lm(formula = Y ~ x, data = cereales)

Coefficients:
(Intercept) x
5.468613 0.007639

- ✓ Les estimateurs par la méthode des moindres carrées ordinaires **B_O et B_1 sont respectivement 5,468613 et 0,007639**

2- ETABLIR LES COEFFICIENTS DE CORRELATION

Le coefficient de corrélation entre Y et x montre le niveau de relation entre les 02 variables. Plus il est proche de -1 ET 1 ; plus on en déduit que la relation entre la variable Y et x est forte.

> cor(Y,x)
[1] 0.1535349

- ✓ Dans notre cas d'étude, la corrélation entre Y et x est de **0,153549. On peut en déduire que Y et x sont faiblement corrélées.**

3- LES ECARTYPES POUR LES ESTIMATEURS.

En langage courante, l'ecartype est définie comme la racine carre de la variance. Autrement dit c'est un des éléments qui forment les paramètres de dispersion comme la variance.

On peut obtenir ces résultats en faisant sur R le résumé de notre régression **a1** par la commande de sortie **« summary(a1) »** :

> summary(a1)

Call:
lm(formula = Y ~ x, data = cereales)

```
Residuals:
    Min      1Q  Median      3Q     Max
-6.6840 -4.1493 -0.4915  3.4619  9.1876

Coefficients:
            Estimate   Std. Error   t value   Pr(>|t|)
(Intercept) 5.468613   1.010680     5.411      7.21e-07 ***
x           0.007639   0.005677     1.346      0.182
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.401 on 75 degrees of freedom
Multiple R-squared:  0.02357,    Adjusted R-squared:  0.01055
F-statistic: 1.811 on 1 and 75 DF,  p-value: 0.1825
```

- ✓ L'ecartype de nos estimateurs correspond aux résultats de la colonne std. Error. Nous en tirons les valeurs suivantes **1,010680 et 0,005677**

5- LA TABLE D'ANOVA

La fonction à utiliser est aov (). Comme pour le modèle de régression, l'ANOVA fonctionne avec des for- mules R. Il faut donc spécifier le modèle à utiliser ou bien de façons automatiques faire la commande suivante :

```
> anova(a1)

Analysis of Variance Table

Response: Y
           Df  Sum      Sq Mean    Sq F value  Pr(>F)
x          1   35.07    35.066     1.8107   0.1825
Residuals  75  1452.47  19.366
```

6- LES INDIVIDUS HORS NORMES, LES POINTS LEVIERS ET INFLUENTS

- **Calcul des résidus studentise de la régression**

```
> res.student <- rstudent(a1)
> print(res.student)

          1           2           3           4           5           6           7           8           9
         10
-0.10496645  0.55951165 -0.56424853 -1.50837186  0.22847050
0.71999802  1.75822603  0.21124717 -0.22687471 -0.47289165
         11          12          13          14          15          16          17          18
         19          20
 1.11532303 -1.56805645  0.43954357  0.10496115  1.41834511 -
1.06917629 -1.32736865  1.34866425  1.41834511  0.10496115
         21          22          23          24          25          26          27          28
         29          30
-1.40624183 -0.95189991  0.78997424 -0.43726705  1.51836225
0.91632343  0.35545496 -0.11391184  1.08245135  1.26340711
         31          32          33          34          35          36          37          38
         39          40
 2.17521759  0.32084157 -0.34980939 -0.86036340 -0.46701924
0.88265407  0.60169632  0.95056040 -0.17440209  0.50824453
         41          42          43          44          45          46          47          48
         49          50
-1.02903595 -0.13963907  1.18316491 -0.57378125 -0.58029130 -
0.58029130  1.47186442 -0.26218327  0.47376290 -0.03404071
         51          52          53          54          55          56          57          58
         59          60
-1.09202850  0.73729505  1.62203554 -1.15317889 -1.28226717 -
1.28226717 -0.11363006 -1.52351581  1.13200906  0.33243824
         61          62          63          64          65          66          67          68
         69          70
 0.12324929 -1.22430990 -1.08968824 -1.28226717 -1.28226717 -
1.28226717  2.11784015 -0.97082339 -0.13474944 -0.91470216
         71          72          73          74          75          76          77
 1.63920636 -0.91470216 -1.00937881  1.25426135 -0.97082339 -
0.91470216  0.22847050
```

- **<u>Calcul du seuil critique des résidus studentises au seuil de 10%</u>**

On utilisera une loi de student a n-p-2 degrés de liberté avec n=nombre d'observation, p=nombre de variables explicatives et alpha= 0,1(10% seuil conventionnel)
Dans notre régression simple, n=77 ; p=1 et alpha = 0,1

Par application sous R le seuil critique est de :

```
> alpha <- 0.1
> seuil.student <- qt(1-alpha/2,77-1-2)
> print(seuil.student)
[1] 1.665707
```

La valeur de notre seuil critique pour les résidus studentises est de 1,665707

- **<u>Calcul des points atypiques a ce seuil</u>**

```
> atypiques.rstudent <- (res.student < -seuil.student | res.student > +seuil.student)
> ab.student <- cereales[atypiques.rstudent,]
> print(ab.student)
# A tibble: 3 x 16
  name         mfr   type  calories protein   fat sodium fiber carbo sugars potass vitamins shelf weight  cups rating
  <chr>        <chr> <chr>    <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>  <dbl>  <dbl>    <dbl> <dbl>  <dbl> <dbl>  <dbl>
1 Apple_Jacks  K     C      110      2         0    125     1    11     14     30       25     2      1  1      33.2
2 Golden_Crisp P     C      100      2         0     45     0    11     15     40       25     1      1  0.88   35.3
3 Smacks       K     C      110      2         1     70     1     9     15     40       25     2      1  0.75   31.2
```

- **Calcul du levier pour chaque observation**

Le levier est donné par :
> indicateurs <- influence.measures(a1)

- **On s'intéresse a la matrice infmat**

> print(indicateurs$infmat)

	dfb.1_	dfb.x	dffit	cov.r	cook.d	hat
1	-0.0088853386	0.0033491228	-0.012503492	1.0414749	7.921320e-05	0.01399080
2	0.1218982525	-0.1031039545	0.122022218	1.0670187	7.513529e-03	0.04540251
3	0.0352912843	-0.0779921041	-0.101744621	1.0515415	5.223456e-03	0.03149095
4	-0.1106319936	0.0285206698	-0.175387528	0.9799149	1.512326e-02	0.01333977
5	0.0013017648	0.0135051543	0.029523126	1.0428902	4.413854e-04	0.01642375
6	0.0203409233	0.0237996703	0.085993560	1.0274174	3.721342e-03	0.01406429
7	0.1587851184	-0.0675359760	0.212830675	0.9603478	2.203403e-02	0.01444114
8	-0.0011831509	0.0152480232	0.028683398	1.0448872	4.166764e-04	0.01810279
9	-0.0012926725	-0.0134108257	-0.029316917	1.0429106	4.352454e-04	0.01642375
10	0.0026485666	-0.0341337722	-0.064209804	1.0398535	2.083012e-03	0.01810279
11	-0.0188729726	0.0951232986	0.159797383	1.0139211	1.272621e-02	0.02011469
12	0.1525646983	-0.2801531535	-0.334471437	1.0059835	5.486834e-02	0.04351825
13	-0.0024617910	0.0317266762	0.059681762	1.0407066	1.800323e-03	0.01810279
14	0.0076984076	-0.0019846315	0.012204469	1.0407877	7.546970e-05	0.01333977

15 0.0400701782 0.0468836648 0.169401223 0.9874452 1.415741e-02 0.01406429
16 0.0915734982 -0.1765180520 -0.215893245 1.0368123 2.326055e-02 0.03917628
17 0.1291468791 -0.2371512273 -0.283131961 1.0245757 3.967877e-02 0.04351825
18 0.1755430871 -0.1134611403 0.192290535 0.9984072 1.828814e-02 0.01992361
19 0.0400701782 0.0468836648 0.169401223 0.9874452 1.415741e-02 0.01406429
20 0.0076984076 -0.0019846315 0.012204469 1.0407877 7.546970e-05 0.01333977
21 -0.1991829108 0.1367920002 -0.212080671 0.9965971 2.219976e-02 0.02223898
22 0.0161076034 -0.0811853226 -0.136383014 1.0230874 9.311820e-03 0.02011469
23 0.0579409014 -0.0149370292 0.091855088 1.0237577 4.239931e-03 0.01333977
24 -0.0074247356 -0.0201455335 -0.054101798 1.0375659 1.479457e-03 0.01507759
25 0.1371230577 -0.0583224652 0.183795517 0.9802326 1.660144e-02 0.01444114
26 0.0052209699 0.0541649337 0.118407987 1.0210594 7.025246e-03 0.01642375
27 0.0838769014 -0.0728222530 0.083876901 1.0807132 3.559126e-03 0.05274514
28 -0.0264652188 0.0228657629 -0.026466172 1.0822803 3.548997e-04 0.05121670
29 -0.0429227363 0.1208442519 0.173817720 1.0211046 1.507180e-02 0.02513708
30 0.0998024792 -0.0320978887 0.148480402 0.9978838 1.093628e-02 0.01362368
31 0.3965204409 -0.3126421143 0.401607684 0.9383872 7.682214e-02 0.03296410
32 -0.0274796451 0.0529700570 0.064785881 1.0661245 2.124010e-03 0.03917628
33 -0.0256568767 0.0066142829 -0.040674456 1.0376630 8.370001e-04 0.01333977

34 -0.0339997964 -0.0172551673 -0.100207103 1.0206234 5.038182e-03 0.01338386
35 -0.0688397316 0.0485076143 -0.072482947 1.0457794 2.654563e-03 0.02352149
36 -0.0149358578 0.0752795063 0.126461848 1.0265664 8.019923e-03 0.02011469
37 -0.0307324218 0.0751510308 0.102399477 1.0466938 5.287805e-03 0.02814758
38 0.0268546241 0.0314209533 0.113530969 1.0168782 6.452938e-03 0.01406429
39 -0.0068920127 -0.0034977513 -0.020312728 1.0402888 2.090054e-04 0.01338386
40 0.0200847810 0.0101931862 0.059195582 1.0339134 1.769558e-03 0.01338386
41 0.0643617098 -0.1422363982 -0.185554533 1.0308946 1.720173e-02 0.03149095
42 -0.0086658916 0.0008265892 -0.016039294 1.0402138 1.303334e-04 0.01302160
43 0.0334260178 0.0391097392 0.141312282 1.0035352 9.931628e-03 0.01406429
44 -0.1353954751 0.1175508797 -0.135395475 1.0748218 9.248685e-03 0.05274514
45 -0.1341166842 0.1156838747 -0.134125355 1.0723052 9.075061e-03 0.05071387
46 -0.1341166842 0.1156838747 -0.134125355 1.0723052 9.075061e-03 0.05071387
47 0.0913427575 -0.0087126566 0.169062040 0.9823996 1.407214e-02 0.01302160
48 0.0044365422 -0.0223609992 -0.037564185 1.0463509 7.144046e-04 0.02011469
49 0.0080444303 0.0218269511 0.058617325 1.0366365 1.735946e-03 0.01507759
50 0.0005760209 -0.0029032525 -0.004877166 1.0482630 1.205391e-05 0.02011469
51 -0.0431547258 -0.0219013669 -0.127189293 1.0083817 8.067848e-03 0.01338386
52 0.0291363874 0.0147869486 0.085873249 1.0260146 3.709682e-03 0.01338386

53 0.0092419320 0.0958803902 0.209600626 0.9738802 2.149869e-02 0.01642375
54 0.1530261797 -0.2536652021 -0.287559290 1.0529007 4.116415e-02 0.05854134
55 -0.3025772873 0.2626987811 -0.302577287 1.0377776 4.538666e-02 0.05274514
56 -0.3025772873 0.2626987811 -0.302577287 1.0377776 4.538666e-02 0.05274514
57 -0.0089761738 0.0028868644 -0.013354236 1.0410340 9.035702e-05 0.01362368
58 -0.3595048605 0.3121235222 -0.359504860 1.0194512 6.350328e-02 0.05274514
59 -0.0063401445 0.0817094984 0.153705570 1.0108358 1.176854e-02 0.01810279
60 0.0243827837 -0.0062858247 0.038654606 1.0379949 7.560560e-04 0.01333977
61 0.0290832022 -0.0252501496 0.029083202 1.0839618 4.285434e-04 0.05274514
62 0.0485478915 -0.1366812589 -0.196597061 1.0122722 1.919749e-02 0.02513708
63 0.1060216658 -0.1946866107 -0.232433973 1.0402925 2.694544e-02 0.04351825
64 -0.3025772873 0.2626987811 -0.302577287 1.0377776 4.538666e-02 0.05274514
65 -0.3025772873 0.2626987811 -0.302577287 1.0377776 4.538666e-02 0.05274514
66 -0.3025772873 0.2626987811 -0.302577287 1.0377776 4.538666e-02 0.05274514
67 0.3244041058 -0.2339607190 0.338340445 0.9364651 5.469543e-02 0.02488722
68 0.0274455353 -0.0955660674 -0.147154112 1.0245459 1.083547e-02 0.02245945
69 -0.0293572456 0.0248309394 -0.029387101 1.0755379 4.375286e-04 0.04540251
70 -0.0052117324 -0.0540690986 -0.118198485 1.0211404 7.000686e-03 0.01642375
71 0.0278335033 0.0755206389 0.202814302 0.9711293 2.011438e-02 0.01507759

72 -0.0052117324 -0.0540690986 -0.118198485 1.0211404 7.000686e-03 0.01642375
73 0.0515553349 -0.1260700047 -0.171780778 1.0284459 1.475061e-02 0.02814758
74 0.0919941813 -0.0237158852 0.145840561 0.9982047 1.055408e-02 0.01333977
75 0.0274455353 -0.0955660674 -0.147154112 1.0245459 1.083547e-02 0.02245945
76 -0.0052117324 -0.0540690986 -0.118198485 1.0211404 7.000686e-03 0.01642375
77 0.0013017648 0.0135051543 0.029523126 1.0428902 4.413854e-04 0.01642375

On voit que toutes les 77 observations sont bien représentées.

- **<u>On récupère la colonne Hat qui correspond au levier</u>**

```
> res.hat <- indicateurs$infmat[,"hat"]
```

- **<u>Calcul du seuil au sens de levier</u>**

Ce seuil peut être calculé par la formule suivante : 2x(p+1)/n
Dans notre modèle a1 : n=77, p=1
Par application :

```
> seuil.hat <- 2*(1+1)/77
> print(seuil.hat)
[1] 0.05194805
```

Notre seuil pour le sens de levier est fixe à 0,05194805.

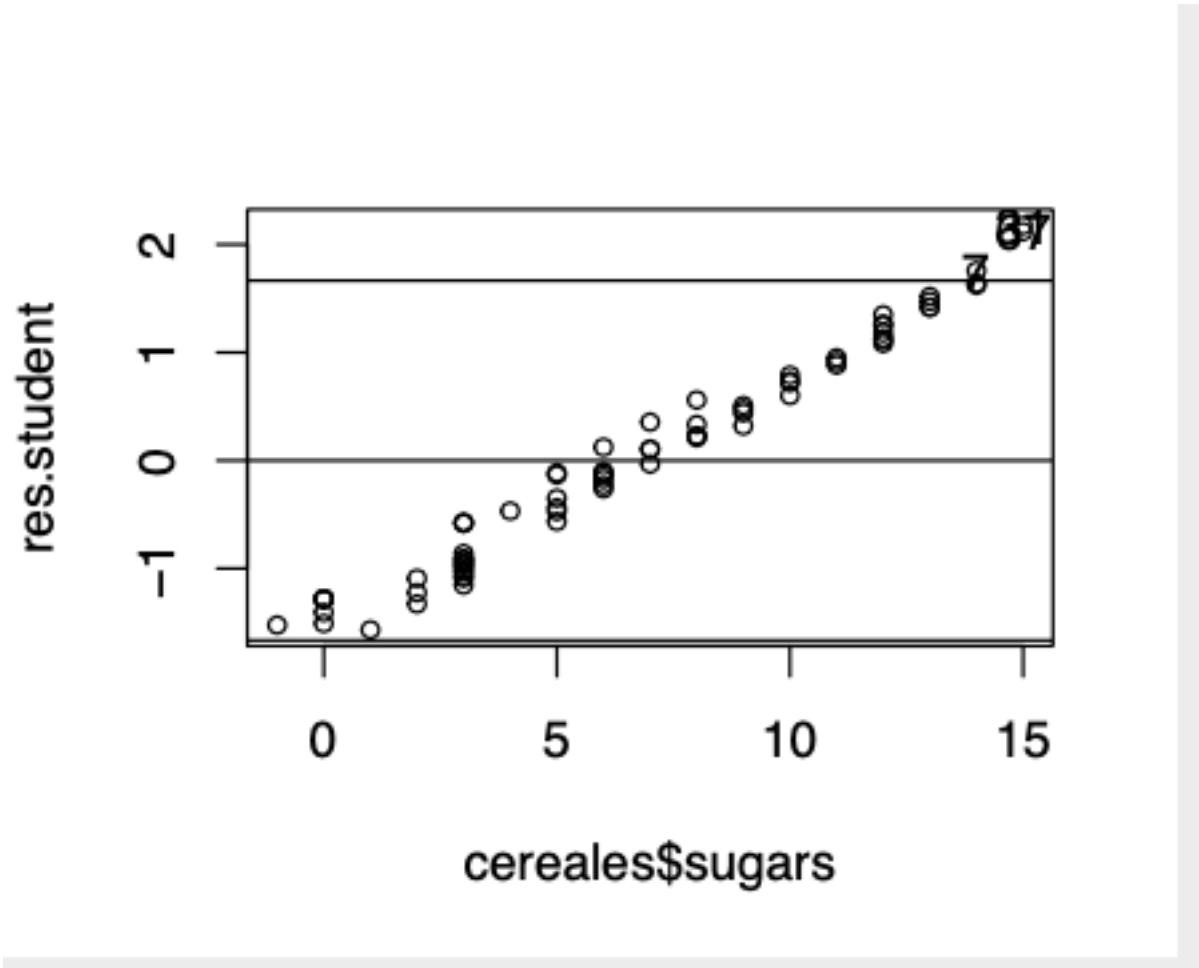
- **<u>Calcul des points atypiques au sens du levier.</u>**

```
> atypiques.levier <- (res.hat > seuil.hat)
> ab.hat <- cereales[atypiques.levier,]
> print(ab.hat)
# A tibble: 10 x 16
```

```
   name              mfr   type  calories protein   fat sodium fiber carbo
sugars potass vitamins shelf weight  cups rating
   <chr>             <chr> <chr> <chr>    <chr>   <dbl>  <dbl> <dbl>
<dbl>  <dbl>  <dbl>    <dbl> <dbl>  <dbl> <dbl>  <dbl>
 1 Frosted_Mini-Whe… K     C     100      3         0      0   3      14      7
100       25     2   1     0.8     58.3
 2 Maypo             A     H     100      4         1      0   0      16      3     95
25     2   1     1      54.9
 3 Product_19        K     C     100      3         0    320   1      20      3     45
100     3   1     1      41.5
 4 Puffed_Rice       Q     C     50       1         0      0   0      13      0     15
0     3   0.5   1      60.8
 5 Puffed_Wheat      Q     C     50       2         0      0   1      10      0     50
0     3   0.5   1      63.0
 6 Quaker_Oatmeal    Q     H     100      5         2      0   2.7    -1     -1
110       0     1   1     0.67    50.8
 7 Raisin_Squares    K     C     90       2         0      0   2      15      6    110
25     3   1     0.5     55.3
 8 Shredded_Wheat    N     C     80       2         0      0   3      16      0
95       0     1   0.83  1      68.2
 9 Shredded_Wheat__… N     C     90       3         0      0   4      19      0
140       0     1   1     0.67    74.5
10 Shredded_Wheat_s… N     C     90       3         0      0   3      20      0
120       0     1   1     0.67    72.8
```

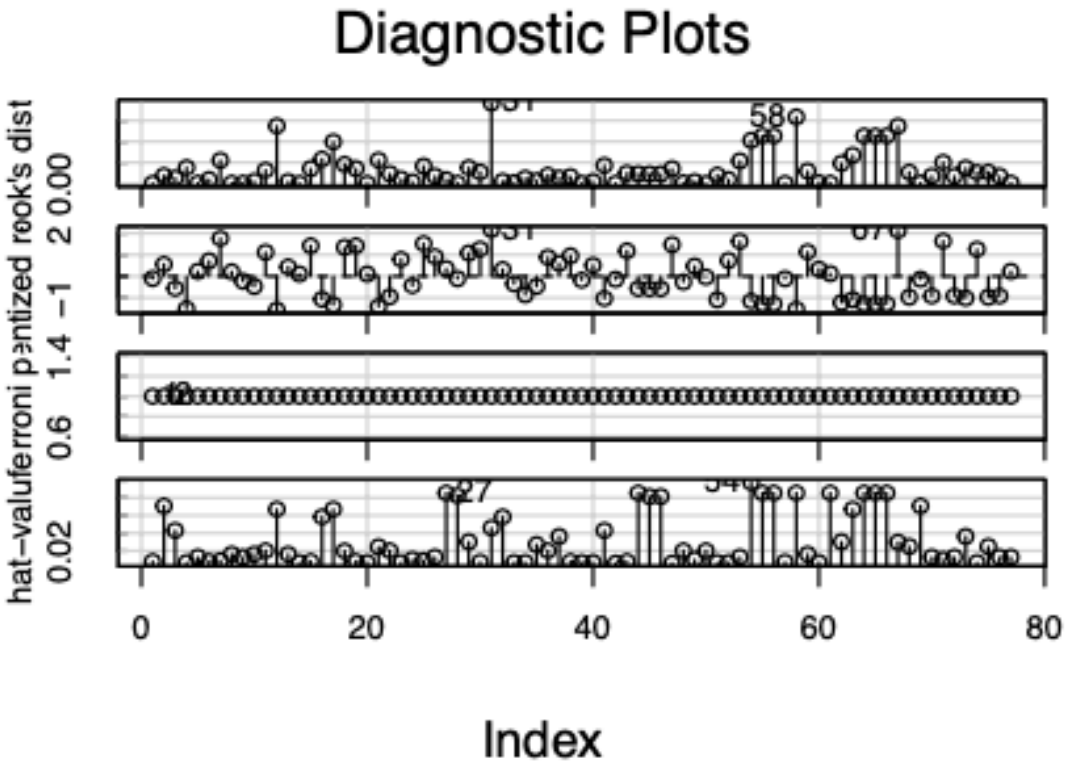
- **<u>On peut procéder par une visualisation des résidus studentises pour afficher les individus hors norme</u>**

```
> plot(cereales$sugars,res.student,cex=0.75)
> abline(h=-seuil.student)
> abline(h=+seuil.student)
> abline(h=0)
>text(cereales$sugars[atypiques.rstudent],res.student[atypiques.rstudent],
rownames(cereales)[atypiques.rstudent])
```



On peut s'apercevoir de l'individus 7 et 31 sont juges influents et leur distance de Cook est supérieur a 1 comme nous le montre le graphe si dessous obtenu par la librairie CAR qui nous affichera 04 graphes pour une meilleure visualisation :

> library(carData)
> library(car)
> infIndexPlot(a1)



Le **package "car"** dispose d'une fonction très utile pour détecter ces données différentes, il s'agit de la **fonction "influenceIndexPlot".** Cette fonction renvoie 4 graphs :

- le premier (en partant du bas), celui des **hat value**, reflète l'**effet de levier** (ou poids) de chaque donnée sur sa propre estimation. Une donnée est considérée comme atypique lorsque cette valeur est inférieure à 0.05.
- le second plot celui des **p-value de Bonferroni** permet de mettre en évidence les **outliers**. Est considérée comme **outlier une donnée ayant une p-value inférieure à 0.05**.
- le troisième plot, celui **des résidus studentizés** permet également de mettre en évidence les **outliers**
- le dernier plot, celui **des distance de Cook** permet d'évaluer l'**influence des données sur les paramètres de régression.** La distance de Cook mesure le changement dans l'estimation des paramètres de régression lorsque la donnée n'est pas prise en compte par les moindres carrés. **Plus la distance est élevée, plus la modification des paramètres de régression est importante. Le**

seuil de 1 est couramment utilisé pour considérer qu'une donnée est très influente. Vous trouverez plus de détail sur l'effet de levier et les distances de Cook

7- FAIRE LA REGRESSION

Dans cette partie, l'idée sera de créer un nouveau data frame ou une nouvelle base de données dans laquelle les observations atypiques au sens de levier ou au sens des résidus studentises seront exclues afin de faire une nouvelle régression.

Par application :

- **Supprimons les points atypiques de la base**

```
> excluded <- (atypiques.rstudent | atypiques.levier)
> print(excluded)
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
16   17   18   19   20
FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE
  21   22   23   24   25   26   27   28   29   30   31   32   33   34
35   36   37   38   39   40
FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE
  41   42   43   44   45   46   47   48   49   50   51   52   53   54
55   56   57   58   59   60
FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE
TRUE FALSE FALSE
  61   62   63   64   65   66   67   68   69   70   71   72   73   74
75   76   77
 TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

Les TRUES sont ceux à exclure.

- **<u>Récupération de notre nouvelle base de données en gardant les non exclus</u>**

```
> cereales.clean <- cereales[!excluded,]
> dim(cereales.clean)
[1] 64 16
```

Ainsi nous obtenons une nouvelle base de données à 64 observations et 16 variables.

- **<u>NOUVELLE REGRESSION</u>**

Avec ce nouveau data frame, on peut procéder à notre régression simple toujours en utilisant les mêmes variables que le modèle a1 afin de voir la pertinence du modèle après le retrait des points atypiques
Nous allons choisir comme variable endogène y **(sugars)** et la variable exogène X **(sodium)**

La modélisation de notre problème reviendra à faire la régression suivante :

$$Sugars = B_O + B_1(sodium) + E_i$$

Avec y= sugars
X= sodium
E_i= erreur ou bruit
B_0 et B_1 sont les paramètres à estimer

Avant de commencer notre régression, nous devons affecter y aux données de la variables sugars et Xaux données de la variable sodium. Cette affectation est faisable grâce aux commandes suivantes :

```
> y<-cereales.clean$sugars
> X<-cereales.clean$sodium
```

La régression du modèle que nous appellerons a2 est :

```
> a2<-lm(y~X,data=cereales.clean)
> summary(a2)

Call:
lm(formula = y ~ X, data = cereales.clean)

Residuals:
    Min      1Q  Median      3Q     Max
-7.1751 -3.9124 -0.5271  3.3006  7.0025

Coefficients:
             Estimate   Std. Error   t value   Pr(>|t|)
(Intercept)  7.293553   1.365921     5.340     1.4e-06 ***
X            -0.001480  0.007181     -0.206    0.837
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.97 on 62 degrees of freedom
Multiple R-squared:  0.0006848,    Adjusted R-squared:  -0.01543
F-statistic: 0.04249 on 1 and 62 DF,  p-value: 0.8374
```

8- INFERENCE DANS LA REGRESSION

1. Test de student pour y et X

```
> t.test(X,y)

        Welch Two Sample t-test

data:  X and y
t = 19.514, df = 63.403, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 152.7479 187.5958
sample estimates:
mean of x mean of y
177.20312   7.03125
```

2- intervalle de confiance de la pente de la droite de régression

```
> confint(a2)
                2.5 %      97.5 %
(Intercept)  4.56311638 10.02398989
X           -0.01583577  0.01287529
```

3-intervalle de confiance pour la moyenne y (sugar)

```
> t.test(cereales.clean$sugars)

        One Sample t-test

data:  cereales.clean$sugars
t = 14.277, df = 63, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 6.047092 8.015408
sample estimates:
mean of x
  7.03125
```

La valeur moyenne en sugar (y) est de 7,03125 avec un intervalle de 95% de [6,047092 ; 8,015408]

4-intervalle de prédiction de y pour une valeur aléatoire de x

Pour une valeur de sodium X0= 15 ; l'intervalle de prédiction pour y (sugar) se présente comme suit :

```
> Xo<-15
> predict(a2,data.frame(X=Xo),interval="prediction")
      fit       lwr      upr
1 7.27135 -1.058765 15.60146
```

9-VERIFIONS LES HYPHOTESES DE LA REGRESSION

Pour rappel, nous avions un premier modèle a1 avec lequel nous avions estime nos paramètres Bo et B1 et en faire une étude de corrélation. Au cours de notre étude, il s'est avéré que certains individus avez des influences qui nous ont conduit à des points atypiques au sens de leviers au sens des résidus studentises dont nous étions obliges d'exclure du modèle de régression afin d'obtenir un nouveau data frame ou base de données nommée **cereales.clean** dont nous utiliserons dans la suite de nos travaux ,d'où la genèse du modèle a2.

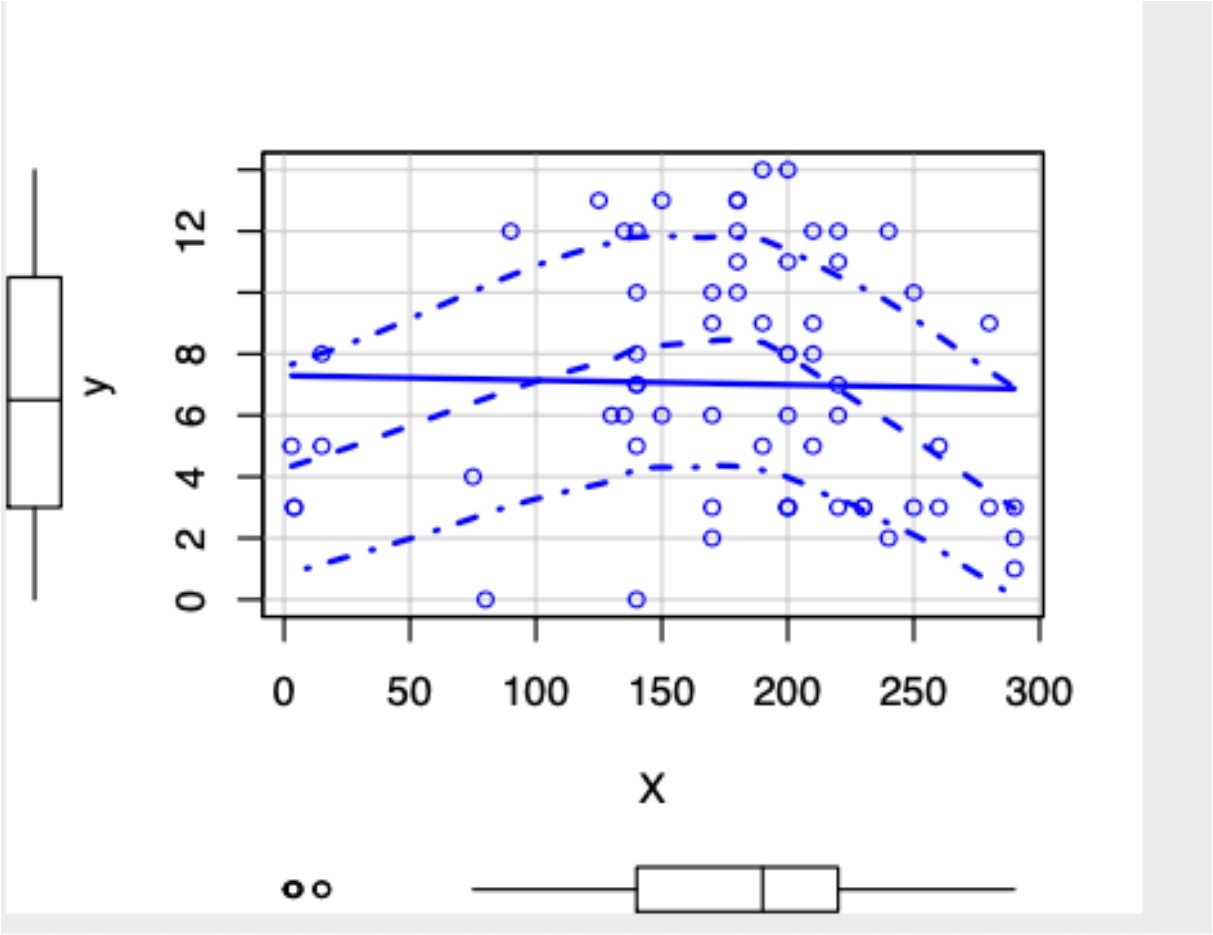
La régression du modèle a2 se présente comme suit :

```
> y<-cereales.clean$sugars
> X<-cereales.clean$sodium
> a2<-lm(y~X,data=cereales.clean)
> summary(a2)

Call:
lm(formula = y ~ X, data = cereales.clean)

Residuals:
   Min      1Q  Median      3Q     Max
-7.1751 -3.9124 -0.5271  3.3006  7.0025

Coefficients:
             Estimate      Std. Error      t value      Pr(>|t|)
(Intercept)  7.293553   1.365921        5.340          1.4e-06 ***
X            -0.001480     0.007181     -0.206          0.837
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.97 on 62 degrees of freedom
Multiple R-squared:  0.0006848,    Adjusted R-squared:  -0.01543
F-statistic: 0.04249 on 1 and 62 DF,  p-value: 0.83
```

➢ VERIFIONS LES HYPOTHESES DU MODELE DE REGRESSION

- EVALUATION VISUELLE DE LA LINEARITE

> library(car)
> scatterplot(y~X, data=cereales.clean)



La régression linéaire simple permet d'évaluer la significativité du lien linéaire entre deux variables. **La forme linéaire entre les deux variables est donc présupposée**. Autrement dit, on fait l'hypothèse que la forme de la relation entre les variables est linéaire. Néanmoins, il est préférable de **vérifier si cette hypothèse est acceptable**, ou non, car si ce n'est pas le cas, les résultats de l'analyse n'auront pas de sens.

La ligne en **trait plein est la droite de régression linéaire** (définie par la méthode des moindres carrés) entre les deux variables. **La ligne centrale en pointillé est la courbe de régression locale de type lowess.**
Elle indique la tendance globale entre les deux variables. Les deux lignes extérieures représentent un intervalle de confiance de la courbe lowess

Ici, **la droite de régression est comprise dans l'intervalle de confiance de la courbe lowess, l'hypothèse de linéarité est donc acceptable**.

- **EVALUATION DES HYPOTHESE DE VALIDITE DES RESULTAS**

Le test d'évaluation de la significativité du lien linéaire entre les deux variables est valide, si les résidus :

- sont **indépendants**
- sont **distribués selon une loi Normale de moyenne 0**
- sont distribués de façon homogènes, c'est à dire, avec u**ne variance constante.**

- **EVALUATION DE L'HYPOTHESE D'INDEPENDANCE DES RESIDUS**

En général, l'hypothèse d'indépendance des résidus est validée ou rejetée en fonction du protocole expérimental.

Le **test de Durbin-Watson** peut être employé pour évaluer la présence d'une autocorrélation. L'hypothèse d'indépendance des résidus est rejetée lorsque la p-value du test est inférieure à 0.05.

PAR application nous obtenons les résultats suivants :

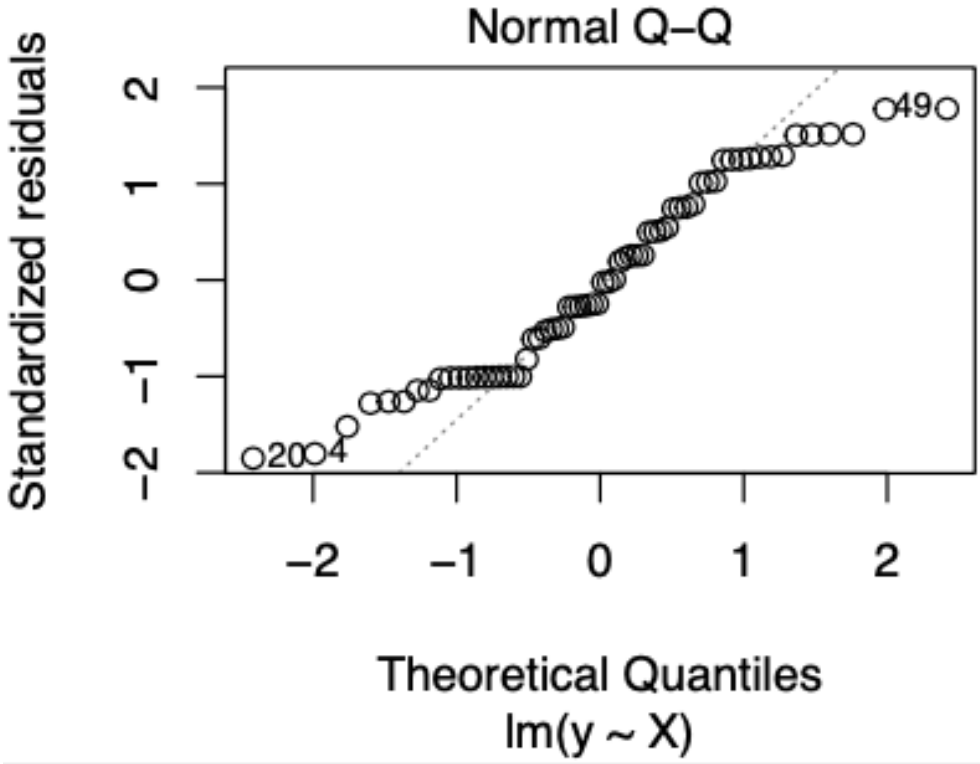
```
> durbinWatsonTest (a2)
 lag Autocorrelation D-W Statistic p-value
```

1 -0.04783925 2.093409 0.652
Alternative hypothesis: rho != 0

La p-value est supérieure à 0,05, on accepte l'hypothèse d'indépendance des résidus.

- **<u>EVALUATION DE L'HYPOTHESE DE NORMALITE DES RESIDUS</u>**

Cette hypothèse peut s'**évaluer graphiquement à l'aide d'un QQplot**. Si les résidus sont bien **distribués le long de la droite figurant sur le plot**, alors l'**hypothèse de normalité est acceptée**. A l'inverse, s'ils s'en écartent, alors l'hypothèse de normalité est rejetée.



Le **test de Shapiro-Wilk** peut également être employé pour **évaluer la normalité des résidus**. L'hypothèse de normalité est **rejetée si la p-value est inférieure à 0.05**.

> shapiro.test(residuals(a2))

Shapiro-Wilk normality test

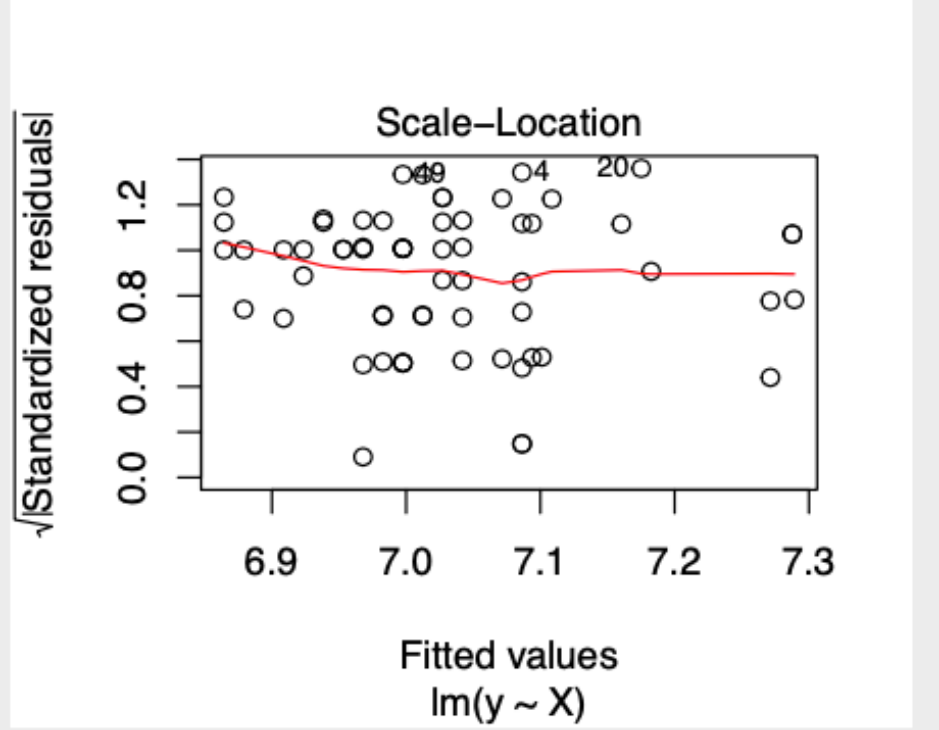
data: residuals(a2)
W = 0.94502, p-value = 0.006552

La p-value est inferieure a 0,05 donc l'hypothèse de normalité des résidus est rejetée.

- **EVALUATION D'HYPHOTHESE D'HOMOGENEITE DES RESIDUS**

Là encore, cette hypothèse peut se **vérifier de façon visuelle,** pour cela il faut réaliser un "**residuals vs fitted plot"**. Les "**fitted**" correspondent aux **réponses prédites par le modèle, pour les valeurs observées de la variable prédicitive**.

> plot(a2, 3)



Ici, la courbe rouge, qui est aussi une courbe de régression locale, est globalement plate. Ceci montre que les résidus ont tendance à être répartis de façon homogène tout le long du gradient des

valeurs de prestige prédites. Et donc que l'hypothèse d'homogénéité des résidus est acceptée.

Il est également possible d'évaluer cette hypothèse en employant le test de Breush-Pagan. L'hypothèse d'homogénéité est rejetée si la p-value est inférieure à 0.05.

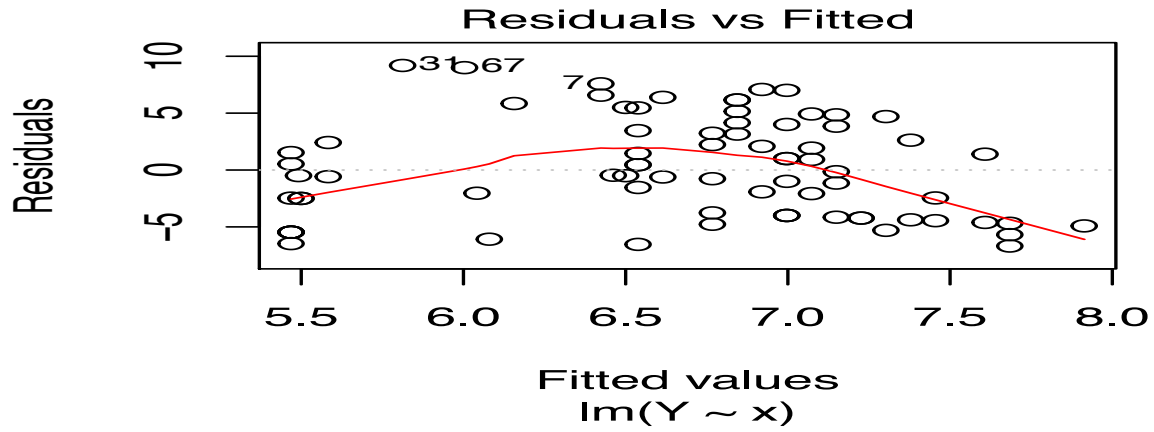
```
> ncvTest(a2)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.02398539, Df = 1, p = 0.87692
```

Ici le test ne rejette pas l'hypothèse d'homogénéité des résidus car la p-value est supérieure a 0,05.

➢ EVALUATION A PISTERIORI DE L'HYPOTHESE DE LINEARITE

Cette hypothèse peut s'évaluer sur les résidus à l'aide du plot suivant :

```
> plot(a1,1)
```



Ici, le plot nous montre que lorsque les réponses prédites par le modèle (fitted values) augmentent, les **résidus restent**

globalement uniformément distribués de part et d'autre de 0. Cela montre, qu'**en moyenne, la droite de régression, est bien adaptée aux données, et donc que l'hypothèse de linéarité est acceptable**.

THEME 2 REGRESSION MULTIPLE ET CONSTRUCTION D'UN MODELE

Le modèle de régression linéaire multiple est une généralisation du modèle de régression simple lorsque les variables explicatives sont en nombre quelconque. Nous supposons donc que les données collectées suivent le modèle suivant :

$$Y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i,\ i=1,...,n$$

où :

— les x_{ij} sont des nombres connus, non aléatoires, la variable x_{i1} valant souvent 1 pour tout i;

— les paramètres β_j du modèle sont inconnus, mais non aléatoires ;

— les ε_i sont des variables aléatoires inconnues.

1- REGRESSION LINEAIRE MULTIPLE

En partant de la modélisation énoncée ci-dessus et en tenant compte que les variables quantitatives (de type double) de notre base de données CEREALES.CSV, notre modèle de régression multiple se présentera comme suit :

Notre variable endogène est **fat.** Elle sera étudiée en fonction des variables explicatives quantitatives telles que s**odium, fiber, potass, carbo, sugars, vitamins, shelf, weight, cups, et rating**

Par application :

$Fat = Bo + B_1(sodium) + B_2(fiber) + B_3(carbo) + B_4(sugars) + B_5(potass) + B_6(vitamins) + B_7(shelf) + B_8(weight) + B_9(cups) + B_{10}(rating) + E_I$

```
>modele<lm(fat~sodium+fiber+carbo+sugars+potass+vitamins+shelf
+weight+cups+rating,data= cereales)
> summary(modele)

Call:
lm(formula = fat ~ sodium + fiber + carbo + sugars + potass +
    vitamins + shelf + weight + cups + rating, data = cereales)

Residuals:
   Min      1Q  Median      3Q     Max
-5.6813 -0.7964  0.0779  0.6711  3.9775

Coefficients:
             Estimate   Std. Error   t value      Pr(>|t|)
(Intercept)  6.185881     2.553412   2.423      0.018163 *
sodium      -0.012363     0.003164  -3.908      0.000222 ***
fiber        0.696334     0.258218   2.697      0.008879 **
carbo       -0.089544     0.032007  -2.798      0.006739 **
sugars      -0.499735     0.087277  -5.726      2.74e-07 ***
potass      -0.014781     0.006265  -2.359      0.021285 *
vitamins    -0.006860     0.008288  -0.828      0.410822
shelf       -0.690901     0.063043  -10.959     < 2e-16 ***
weight      10.863660     0.452495  24.008      < 2e-16 ***
cups        -0.305101     0.902170  -0.338      0.736297
rating      -0.183292     0.042711  -4.291      5.94e-05 ***
---
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.457 on 66 degrees of freedom
Multiple R-squared: 0.9975, Adjusted R-squared: 0.9971
F-statistic: 2650 on 10 and 66 DF, p-value: < 2.2e-16

Les fluctuations du modèle sont expliquées a 99,75% par les variables explicatives retenues et significatives.
On peut juger ce modèle pertinent de par la qualité de son ajustement égale à 99,71 %.

Ainsi toutes variables dont sa probabilité (Pr(>|t|)) < 0.05 (seuil conventionnel) sont tous jugées significatives au seuil de 5%.

2- INFERENCE GAUSSIENNE

Les résultats fournis par summary () se présentent de façon identique à ceux de la régression linéaire simple. On y retrouve les estimations des paramètres de régression dans la colonne Estimate.

Les valeurs réalisées des statistiques des tests de Student associes aux

Hypothèses $H_0 : \beta_i = 0$ versus $H_1 : \beta_i \neq 0$ se trouvent dans la colonne t value, les valeurs-p associées dans la colonne Pr(>|t|). Residual standard error fournit l'estimation de σ ainsi que le nombre de degrés de liberté associes $n - p - 1$. On trouve enfin le coefficient de détermination r^2 (Multiple R-squared) ainsi qu'une version ajustée (Adjusted R-squared). Enfin, on trouve la réalisation du test de Fisher global (F-statistic) ainsi que sa valeur-p (p-value) associée.

➢ INTERPRETATION DES RESULTATS SUR L'ETUDE FAT.

- Au vu des résultats du test global de Fisher (valeur p-valu= 2.2e-16), nous pouvons conclure qu'au moins une des variables explicatives est associée à la variable endogène **fat,** ajustée sur les autres variables.

- Les tests individuels de student nous montrent que :

Les variables **sodium, fiber, sugars, potass, carbo, shelf, weight, et rating sont linéairement associées à la variable far** avec un risque d'erreur inferieur a 5 % (**(Pr(>|t|))** < 0.05) ajusté sur les variables **cups et vitamins.**

Les variables **cups et vitamins ne sont pas significativement associées linéairement a la variable fat** lors qu'on a déjà pris en compte les autres variables. Leur risque d'erreurs est supérieur à 5% (**(Pr(>|t|))** > 0.05).

- **INTERVALLE DE CONFIANCE POUR LA REGRESSION**

```
> confint(modele)
                 2.5 %         97.5 %
(Intercept)  1.08782992  11.283931604
sodium      -0.01867975  -0.006045976
fiber        0.18078434   1.211883938
carbo       -0.15344702  -0.025640673
sugars      -0.67398832  -0.325481027
potass      -0.02728998  -0.002271744
vitamins    -0.02340833   0.009687867
shelf       -0.81676997  -0.565031301
weight       9.96022425  11.767095609
cups        -2.10634240   1.496140573
rating      -0.26856764  -0.098016259
```

Remarque importante :

. On aurait pu conclure de façon équivalente en présentant une estimation par intervalle de confiance et en regardant si la valeur z zéro est contenue ou non dans cet intervalle. Si zéro n'appartient pas à l'intervalle de confiance, alors il existe un apport significatif

de la variable dans le modèle ajusté sur les autres variables.

- **TABLEAU D'ANALYSE DE LA VARIANCE**

> anova(modele)

Analysis of Variance Table

Response: fat

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
sodium	1	6566	6566	3092.1001	< 2.2e-16 ***
fiber	1	31	31	14.8325	0.0002682 ***
carbo	1	43377	43377	20428.4906	< 2.2e-16 ***
sugars	1	216	216	101.5177	5.574e-15 ***
potass	1	177	177	83.5306	2.460e-13 ***
vitamins	1	36	36	16.8641	0.0001129 ***
shelf	1	4236	4236	1995.0096	< 2.2e-16 ***
weight	1	1594	1594	750.7381	< 2.2e-16 ***
cups	1	3	3	1.5036	0.2244797
rating	1	39	39	18.4163	5.936e-05 ***
Residuals	66	140	2		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

- **INTERVALLE DE CONFIANCE ET DE PREDICTION POUR UNE NOUVELLE VALEUR ALEATOIRE DES VARIABLES EXPLICATIVES**

Prenons les valeurs aléatoires suivantes :

Sodium=10 ; fiber= 9 ; carbo= 8 ; sugar = 7 ; potass= 60 ; vitamins= 50 ; shelf= 4 ; weight = 3 ; cups = 2 ; rating= 30

```
>newdata<-
data.frame(sodium=10,fiber=9,carbo=8,sugars=7,potass=60,vitamins
=50,shelf=4,weight=3,cups=2,rating=30)

> predict(modele,newdata,interval="prediction")
       fit      lwr      upr
1 30.60332 24.38521 36.82143
```

3- REGRRESSION AVEC LES VARIABLES PREDICTIVES CATEGORIELLES

Nos variables prédictives catégorielles sont : mfr, type, calories et protein.
Elles sont de type caractères ainsi leur étude nécessite une conversion soigneuse du type caractère en numérique que nous ferons en 02 étapes D'abord par la fonction **factor () et ensuite par la fonction as. numeric() pour pouvoir en faire une régression multiple pour l'ensemble des variables de la base céréales.**

Par application :

```
> MFR<-factor(cereales$mfr)
> MFR<-as.numeric(MFR)
> MFR
 [1]  7  9  6  6 10  5  6  5 10  8  9  5  5  5  5 10  6  6  5  6  7  6  5 10  6  6  6
3  6  8  8  5  8  8  8  9  5  8  6
```

```
[40]  6  5  9  5  4  1  2  6  5  6  6  6  5  8  6  9  9  9  9  6  5  6 10  6  7  7  7
6  6  7  5  5  5  5  5 10  5  5
```

```
> TYPE<-factor(cereales$type)
> TYPE<-as.numeric(TYPE)
> TYPE
 [1] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 4 4 4 4 4 4 3 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 5 1 2 4 4 4 4 4 4 4 4 4 4 5 4
[60] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
```

```
> PROTEIN<-factor(cereales$protein)
> PROTEIN<-as.numeric(PROTEIN)
> PROTEIN
 [1] 4 3 4 4 2 2 2 3 2 3 1 6 1 3 1 2 2 1 1 3 3 2 2 2 2 1 3 7 3 1 2 1 3 3 3 1 3
1 2 3 2 4 2 4 7 7 3 2 2 3 3 3 3 3 1 2 4 5 3
[60] 3 2 1 2 2 3 3 2 6 2 2 3 3 2 1 3 3 2
```

```
> CALORIES<-factor(cereales$calories)
> CALORIES<-as.numeric(CALORIES)
> CALORIES
 [1]  8  3  8  7  2  2  2  4 10 10  3  2  3  2  2  2  1  2  2  2  1  2  1  1  2  2  1
11  3  2  1  2  1  2  3  3  2  2  2
[40]  5  2  1  2  1 12 12  6  1  3  5 10  4  3  1  7  7  1  1  3  1 10  2  2  9 10
10  2  2 10  2  5  1  2  2  1  1  2
```

maintenant nous pouvons appliquer notre régression pour l'ensemble des variables de la base.

```
> modele1<-
lm(fat~MFR+TYPE+CALORIES+PROTEIN+sodium+fiber+carbo+
sugars+potass+vitamins+shelf+weight+cups+rating,data=cereales)
> summary(modele1)
```

```
Call:
lm(formula = fat ~ MFR + TYPE + CALORIES + PROTEIN + sodium +
    fiber + carbo + sugars + potass + vitamins + shelf + weight +
    cups + rating, data = cereales)

Residuals:
   Min      1Q  Median      3Q     Max
-2.8888 -0.8588  0.1642  0.7121  3.7649

Coefficients:
             Estimate   Std. Error  t value    Pr(>|t|)
(Intercept) 24.876853   4.950963    5.025      4.56e-06 ***
MFR         -0.046344   0.100802   -0.460      0.64730
TYPE        -3.827512   0.875605   -4.371      4.80e-05 ***
CALORIES    -0.093791   0.077103   -1.216      0.22843
PROTEIN      0.161598   0.229677    0.704      0.48432
sodium      -0.018227   0.003248   -5.612      4.97e-07 ***
fiber        0.806988   0.249140    3.239      0.00193 **
carbo       -0.023498   0.033376   -0.704      0.48404
sugars      -0.575006   0.081045   -7.095      1.47e-09 ***
potass      -0.012739   0.006512   -1.956      0.05496 .
vitamins    -0.013628   0.007837   -1.739      0.08700 .
shelf       -0.605144   0.060336   -10.030     1.33e-14 ***
weight       9.417065   0.559408   16.834      < 2e-16 ***
cups         0.008872   0.827822    0.011      0.99148
rating      -0.223134   0.042601   -5.238      2.06e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.309 on 62 degrees of freedom
Multiple R-squared:  0.9981, Adjusted R-squared:  0.9977
F-statistic:  2346 on 14 and 62 DF,  p-value: < 2.2e-16
```

Nous avons un modèle globalement pertinent a 99% et plus.
Le modèle est bien ajusté par les données du céréale.

4- ETUDE DE LA MULTI COLINEARITE

Dans une régression, la multi colinéarité est un problème qui survient lorsque certaines variables de prévision du modèle mesurent le même phénomène.

L'approche le plus classique consiste à examiner les facteurs d'inflation de la variance ou variance inflation factor (VIF) en anglais.

Ainsi si vif est proche de 1, il n'y a pas de problème de multi colinéarité

Mais si le vif est élevé, le problème de multi colinéarité est évoqué.

(Notons à ce point il n'y pas de consensus car certains auteurs fixent des valeurs de 2,5 ; 5 et 10 même pour parler de multi colinéarité des variables explicatives)

Par application :

```
> library(carData)
> library(car)
> vif(modele1)

     MFR         TYPE     CALORIES      PROTEIN       sodium        fiber
1.582540     7.550854     2.736978     4.458394     3.696847    15.334708
      carbo       sugars       potass     vitamins        shelf
29.649957     5.699682     9.430939     1.390445   194.988906
      weight         cups       rating
302.770027     7.096679    21.093146
```

Au vu des résultats, on peut belle et bien parler de problème de multi colinéarité car nos vifs paraissent élever dans l'ensemble.

5- LES METHODES DE SELECTIONS DES VARIABLES

Parmi le grand nombre de variables explicatives potentielles, il s'agit de sélectionner celles qui sont le plus `à même d'expliquer Y. Cela permet d'économiser le nombre de prédicteurs (et ainsi d'obtenir un modèle parcimonieux) et d'obtenir un bon pouvoir prédictif en éliminant les variables redondantes qui augmentent le facteur d'inflation de la variance (VIF).

Plus le nombre de paramètres augmente (nombre important de variables explicatives), plus l'ajustement aux données est bon (r^2 proche de 1). En contre-partie, l'estimation des paramètres est détériorée (la variance des estimateurs augmente) `à cause des problèmes de colinéarité.

-La méthode du meilleur sous-ensemble (best subset) :

Lorsque le nombre p de variables explicatives n'est pas trop grand, on peut étudier toutes les possibilités.

Il s'agit de la procédure leaps and bounds. A p fixe, on choisira le modèle de régression qui fournit le r le plus grand. Pour deux modèles de régression ayant un nombre diffèrent de variables explicatives, on peut choisir celle qui fournit le r_a^2 ajuste le plus grand.

La fonction R à utiliser est la fonction leaps () disponible dans le package leaps.

-La méthode pas à pas ascendante (forward sélection)

La régression pas à pas ascendante (ou méthode par additions successives) est une méthode itérative. Elle consiste à sélectionner à chaque étape la variable explicative la plus significative (au seuil α) lorsque l'on régresse Y sur toutes les variables explicatives sélectionnées aux étapes précédentes et la nouvelle variable choisie, tant que l'apport marginal de cette dernière est significatif.

Notons le fonctionnement de cette procédure se fait à l'aide de la fonction add1() pour un seuil $\alpha = 0.05$.

-La méthode pas à pas descendante (backward selection)

Cette méthode est aussi appelée régression par éliminations successives. On part cette fois du modèle complet et on élimine `a chaque étape la variable ayant la plus

petite valeur pour la statistique du test de Student (valeur-*p* la plus grande) en valeur absolue, `à condition qu'il soit non significatif (au seuil α choisi).

Notons le fonctionnement de cette procédure se fait à l'aide de la fonction drop1() pour un seuil $\alpha = 0.05$.

-La méthode pas à pas (stepwise)

Cet algorithme est un perfectionnement de la m´méthode ascendante. Il consiste à effectuer en plus, a chaque´ étape, des tests du type Student ou Fisher, ou encore à optimiser un certain critère, pour ne pas introduire une variable non significative et pour´ eliminer´ éventuellement des variables déjà introduites qui ne seraient plus informatives compte tenu de la dernière variable sélectionnée. L'algorithme s'arrête quand on ne peut plus ajouter ni retrancher de variables.

Nous présentons la fonction step () permettant d'effectuer une méthode du type pas à pas en utilisant à chaque´ étape de la procédure une selection faite par le critère bien connu AIC (An Information Criterion). On peut aussi utiliser le critère bien connu BIC (Bayesian Information Criterion) au moyen de l'argument k=log(n) de la fonction step ().

Remarque importante : En effet, il est important de noter que diverses méthodes de selection automatique peuvent ne pas conduire aux mêmes choix de variables explicatives `à retenir dans le modèle final. Elles ont l'avantage d'être faciles à utiliser, et de traiter le problème de la selection de variables de fa¸con systématique. En revanche, l'inconvénient majeur est que les variables sont retenues ou´ éliminées du modèle sur la base de critères uniquement statistiques, sans tenir compte de l'objectif de l'étude. On aboutit généralement à un modèle qui peut être satisfaisant sur le plan purement statistique, alors que les variables retenues ne sont pas les plus pertinentes pour comprendre et interpréter les données de l'enquête.

6-APPLICATION DES METHODES DE SELECTIONS DES VARIABLES SUR LES DONNEES CEREALES.

Par application, on peut réaliser une selection des variables « backward » optimisant le critère AIC.

Ainsi nous utiliserons la librairie **Mass pour** procéder à la selection des

variables.

De ce fait le meilleur modèle sera celui qui présentera un **AIC** plus petit

```
> library(MASS)
> selection.variable <- stepAIC(modele1,direction="backward")

Start:  AIC=54.82
fat ~ MFR + TYPE + CALORIES + PROTEIN + sodium + fiber + carbo +
    sugars + potass + vitamins + shelf + weight + cups + rating

           Df Sum of Sq    RSS     AIC
- cups      1      0.00 106.28  52.817
- MFR       1      0.36 106.65  53.079
- PROTEIN   1      0.85 107.13  53.430
- carbo     1      0.85 107.13  53.430
- CALORIES  1      2.54 108.82  54.633
<none>                 106.28  54.817
- vitamins  1      5.18 111.47  56.484
- potass    1      6.56 112.84  57.429
- fiber     1     17.99 124.27  64.855
- TYPE      1     32.76 139.04  73.503
- rating    1     47.03 153.31  81.027
- sodium    1     54.00 160.28  84.451
- sugars    1     86.29 192.57  98.584
- shelf     1    172.44 278.72 127.054
- weight    1    485.79 592.07 185.066
```

```
Step:  AIC=52.82
fat ~ MFR + TYPE + CALORIES + PROTEIN + sodium + fiber + carbo +
    sugars + potass + vitamins + shelf + weight + rating

           Df Sum of Sq    RSS     AIC
- MFR       1      0.37 106.65  51.083
- carbo     1      0.86 107.14  51.436
- PROTEIN   1      0.87 107.15  51.444
- CALORIES  1      2.54 108.82  52.634
<none>                  106.28  52.817
- vitamins  1      5.29 111.58  54.559
- potass    1      6.57 112.85  55.436
- fiber     1     19.13 125.41  63.561
- TYPE      1     33.18 139.46  71.736
- rating    1     49.79 156.08  80.404
- sodium    1     56.43 162.72  83.612
- sugars    1     87.18 193.46  96.937
- shelf     1    173.90 280.18 125.456
- weight    1    533.94 640.23 189.087
```

```
Step:  AIC=51.08
fat ~ TYPE + CALORIES + PROTEIN + sodium + fiber + carbo +
sugars +
    potass + vitamins + shelf + weight + rating
```

```
           Df Sum of Sq    RSS     AIC
- PROTEIN   1      0.87 107.52  49.709
- carbo     1      0.94 107.59  49.761
- CALORIES  1      2.61 109.26  50.942
<none>                  106.65  51.083
- vitamins  1      4.93 111.58  52.562
- potass    1      6.24 112.89  53.461
- fiber     1     18.82 125.47  61.594
- TYPE      1     32.87 139.52  69.769
- rating    1     49.59 156.24  78.482
- sodium    1     56.16 162.81  81.655
- sugars    1     87.84 194.49  95.346
- shelf     1    178.01 284.66 124.678
- weight    1    562.54 669.19 190.494
```

```
Step:  AIC=49.71
fat ~ TYPE + CALORIES + sodium + fiber + carbo + sugars + potass
+
    vitamins + shelf + weight + rating

           Df Sum of Sq    RSS     AIC
- carbo     1      1.61 109.13  48.855
<none>                  107.52  49.709
- CALORIES  1      3.96 111.48  50.491
- vitamins  1      4.50 112.02  50.865
- potass    1      5.42 112.94  51.494
- fiber     1     18.09 125.61  59.682
- TYPE      1     32.77 140.29  68.194
- rating    1     51.69 159.21  77.935
- sodium    1     55.88 163.40  79.936
- sugars    1     87.66 195.18  93.618
- shelf     1    183.98 291.50 124.505
- weight    1    700.62 808.14 203.022
```

Step: AIC=48.86
fat ~ TYPE + CALORIES + sodium + fiber + sugars + potass + vitamins + shelf + weight + rating

	Df	Sum of Sq	RSS	AIC
<none>			109.13	48.855
- potass	1	5.48	114.61	50.625
- CALORIES	1	5.80	114.93	50.842
- vitamins	1	5.80	114.94	50.844
- fiber	1	24.40	133.54	62.395
- TYPE	1	47.81	156.94	74.830
- rating	1	67.82	176.96	84.072
- sodium	1	91.35	200.49	93.684
- sugars	1	98.43	207.57	96.357
- shelf	1	211.84	320.97	129.921
- weight	1	716.73	825.87	202.692

AU Total 05 modèles de régressions nous ont été présentes par notre méthode de selection et nous pouvons en déduire que le meilleur modèle est celui du plus petit AIC estime à 48,86 appliques aux données céréales.

Sa régression se présente comme suit :

fat ~ TYPE + CALORIES + sodium + fiber + sugars + potass + vitamins + shelf + weight + rating

C'est un modèle a 10 variables explicative comme indiqué ci-dessus

7-L'INDICATEUR STATISTIQUE Cp MALLOW

Le Cp de Mallows vous aide à choisir entre plusieurs modèles de régression. Il vous permet de trouver un juste équilibre concernant le nombre de prédicteurs figurant

dans le modèle. Le Cp de Mallows compare la précision et le biais du modèle complet à ceux de modèles contenant un sous-ensemble des prédicteurs.

En général, vous devez rechercher les modèles où le Cp de Mallows est faible et proche du nombre de prédicteurs du modèle plus la constante (p). Un Cp de Mallows faible indique que le modèle est relativement précis (avec une variance faible) dans son estimation des coefficients de régression réels et sa prévision des réponses futures. Une valeur de Cp de Mallows proche du nombre de prédicteurs plus la constante indique que le modèle est relativement précis et non biaisé concernant l'estimation des véritables coefficients de régression et la prévision des futures réponses. La valeur du Cp de Mallows des modèles présentant une inadéquation de l'ajustement et un biais est supérieure à p.

```
> RES <-
regsubsets(fat~MFR+TYPE+CALORIES+PROTEIN+sodium+fiber+carbo+su
gars+potass+vitamins+shelf+weight+cups+rating,data=cereales, nbest = 1,nvmax
= NULL,force.in = NULL,force.out = NULL,method = "exhaustive")
```

```
> res.legend <-subsets(RES, statistic="cp", legend = FALSE, min.size = 5,main =
"Cp Mallow")

> res.legend

   Abbreviation

MFR                M

TYPE               T

CALORIES           C

PROTEIN            P

sodium             sd

fiber              f

carbo              cr
```

sugars sg

potass p

vitamins v

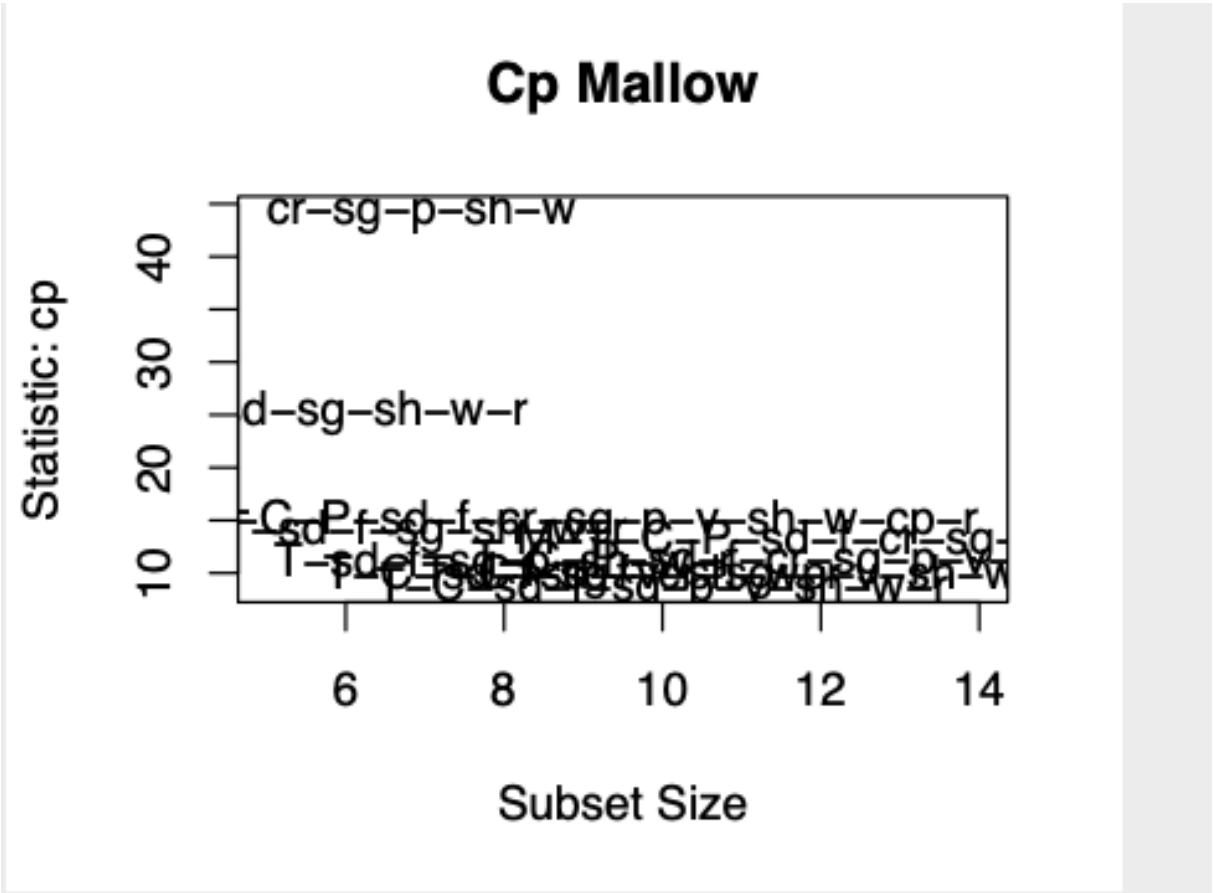
shelf sh

weight w

cups cp

rating r

> abline(a = 1, b = 1, lty = 2)



8- LES CRITERES DE SELECTION DES VARIABLES

Tout dépend en partie de la méthode de selection utilisée. Dans notre cas, la méthode appliquée est **La méthode pas à pas descendante (backward selection).** Cette dernière nous fournit un modèle a plus petit AIC qui sera jugée meilleure.
Pour notre selection le résultat de selection nous fournit le modèle suivant : fat ~ TYPE + CALORIES + sodium + fiber + sugars + potass + vitamins + shelf + weight + rating.

Un modèle a 10 variables explicatives dont avec la variables dépendante fat, l'ajustement du modèle est de 0.9978 soit 99,78% par les données céréales. Et une significativité globale des variables prédictives au seuil de 5% selon la statistique de Fischer < 0,05.

9- UTILISATION DES COMPOSANTES COMME VARIABLES PREDICTIVES.

L'analyse **en composantes principales** (**ACP**), ou *principal component analysis* (*PCA*) en anglais, permet d'analyser et de visualiser un jeu de données contenant des individus décrits par plusieurs variables quantitatives.
C'est une méthode statistique qui permet d'explorer des données dites multivariées (données avec plusieurs variables). Chaque variable pourrait être considérée comme une dimension différente. Si vous avez plus de 3 variables dans votre jeu de données, il pourrait être très difficile de visualiser les données dans une "hyper-espace" multidimensionnelle.

L'analyse en composantes principales est utilisée pour extraire et de visualiser les informations importantes contenues dans une table de données multivariées. L'ACP synthétise cette information en seulement quelques nouvelles variables appelées **composantes principales**. Ces nouvelles variables correspondent à une combinaison linéaire des variables originels. Le nombre de composantes principales est inférieur ou égal au nombre de variables d'origine.
L'information contenue dans un jeu de données correspond à la variance ou l'*inertie totale* qu'il contient. L'objectif de l'ACP est d'identifier les directions (i.e., *axes principaux* ou composantes principales) le long desquelles la variation des données est maximale.
En d'autres termes, l'ACP réduit les dimensions d'une donnée multivariée à deux ou trois composantes principales, qui peuvent être visualisées graphiquement, en perdant le moins possible d'information.

Notez que l'ACP est particulièrement utile lorsque les variables, dans le jeu de données, sont fortement corrélées. La corrélation indique qu'il existe une redondance dans les données. En raison de cette redondance, l'ACP peut être

utilisée pour réduire les variables d'origine en un nombre plus petit de nouvelles variables (= **composantes principales**), ces dernières expliquant la plus grande partie de la variance contenue dans les variables d'origine.

- **FORMAT DES DONNEES**

Selon la terminologie de l'ACP, nos données contiennent une observation de 77 individus repartis en 11 variables quantitatives **(fat, sodium, fiber, carbo, sugars, potass, vitamins, shelf, weight, cups, rating)** sur une base de données à l'origine céréales de 77 0bservations pour 16 variables à caractère qualitative et quantitative.
On Supposera dans notre étude que :

-Tous nos individus sont actives (lignes 1 à 77) ;

- Nos variables actives (utilisées pour l'ACP) sont issues des colonne 6 à 10 c'est-à-dire 11 variables quantitatives partant de la colonne fat a la colonne rating)

Nous commençons par extraire les individus actifs et les variables actives pour l'ACP :

```
> donnees.actives=cereales[1:77, 6:16]
> donnees.actives
# A tibble: 77 x 11
    fat sodium fiber carbo sugars potass vitamins shelf weight  cups rating
  <dbl>  <dbl> <dbl> <dbl>  <dbl>  <dbl>    <dbl> <dbl>  <dbl> <dbl>  <dbl>
1     1    130  10     5      6    280       25     3      1  0.33    68.4
2     5     15   2     8      8    135        0     3      1  1       34.0
3     1    260   9     7      5    320       25     3      1  0.33    59.4
4     0    140  14     8      0    330       25     3      1  0.5     93.7
5     2    200   1    14      8     -1       25     3      1  0.75    34.4
6     2    180   1.5  10.5   10     70       25     1      1  0.75    29.5
7     0    125   1    11     14     30       25     2      1  1       33.2
8     2    210   2    18      8    100       25     3      1.33  0.75    37.0
9     1    200   4    15      6    125       25     1      1  0.67    49.1
10     0    210   5    13      5    190       25     3      1  0.67    53.3
# … with 67 more rows
```

- **STANDARDISATION DES DONNEES**

Dans l'analyse en composantes principales, les variables sont souvent normalisées. Ceci est particulièrement recommandé lorsque les variables sont mesurées dans différentes unités (par exemple : kilogrammes, kilomètres, centimètres, …); sinon, le résultat de l'ACP obtenue sera fortement affecté.

L'objectif est de rendre les variables comparables. Généralement, les variables sont normalisées de manière à ce qu'elles aient au final i) un écart type égal à un et ii) une moyenne égale à zéro.

Techniquement, l'approche consiste à transformer les données en soustrayant à chaque valeur une valeur de référence (la moyenne de la variable) et en la divisant par l'écart type. A l'issue de cette transformation les données obtenues sont dites *données centrées-réduites*. L'ACP appliquée à ces données transformées est appelée *ACP normée*.
La standardisation des données est une approche beaucoup utilisée dans le contexte de l'analyse des données d'expression de gènes avant les analyses de type PCA et de clustering.

Calculer l'ACP sur les individus/variables actifs:

```
> library(ggplot2)
> library(FactoMineR)
> library(factoextra)
> res.pca <- PCA(donnees.actives, graph = FALSE)
> res.pca

**Results for the Principal Component Analysis (PCA)**
The analysis was performed on 77 individuals, described by 11 variables
*The results are available in the following objects:

   name              description
1  "$eig"            "eigenvalues"
2  "$var"            "results for the variables"
3  "$var$coord"      "coord. for the variables"
4  "$var$cor"        "correlations variables - dimensions"
5  "$var$cos2"       "cos2 for the variables"
6  "$var$contrib"    "contributions of the variables"
7  "$ind"            "results for the individuals"
8  "$ind$coord"      "coord. for the individuals"
9  "$ind$cos2"       "cos2 for the individuals"
10 "$ind$contrib"    "contributions of the individuals"
11 "$call"           "summary statistics"
12 "$call$centre"    "mean of the variables"
```

13 "$call$ecart.type" "standard error of the variables"
14 "$call$row.w" "weights for the individuals"
15 "$call$col.w" "weights for the variables"

- ✓ **VISUALISATION ET INTERPRETATION**

  get_eigenvalue(res.pca): Extraction des valeurs propres / variances des composantes principales

  fviz_eig(res.pca): Visualisation des valeurs propres

  get_pca_ind(res.pca), get_pca_var(res.pca): Extraction des résultats pour les individus et les variables, respectivement.

  fviz_pca_ind(res.pca), fviz_pca_var(res.pca): visualisez les résultats des individus et des variables, respectivement.

  fviz_pca_biplot(res.pca): Création d'un biplot des individus et des variables

- ➢ **VALEURS PROPRES ET VARIANCES**

Les valeurs propres (Eigen values en anglais) mesurent la quantité de variance expliquée par chaque axe principal. Les valeurs propres sont grandes pour les premiers axes et petits pour les axes suivants. Autrement dit, les premiers axes correspondent aux directions portant la quantité maximale de variation contenue dans le jeu de données.

Nous examinons les valeurs propres pour déterminer le nombre de composantes principales à prendre en considération. Les valeurs propres et la proportion de variances (i.e. information) retenues par les composantes principales peuvent être extraites à l'aide de la fonction get_eigenvalue() [package factoextra].

> valeurs_propres <- get_eigenvalue(res.pca)

> valeurs_propres

	eigenvalue	variance.percent	cumulative.variance.percent
Dim.1	5.2932313249	48.120284772	48.12028
Dim.2	2.2991557190	20.901415627	69.02170
Dim.3	1.4455362796	13.141238905	82.16294
Dim.4	1.0212170258	9.283791144	91.44673
Dim.5	0.5987314016	5.443012741	96.88974

Dim.6	0.1379238345	1.253853041	98.14360
Dim.7	0.0975935681	0.887214255	99.03081
Dim.8	0.0758870853	0.689882594	99.72069
Dim.9	0.0202402977	0.184002706	99.90470
Dim.10	0.0100577215	0.091433832	99.99613
Dim.11	0.0004257421	0.003870383	100.00000

La somme de toutes les valeurs propres donne une variance totale de 11(nombre de dimensions).

La proportion de variance expliquée par chaque valeur propre est donnée dans la deuxième colonne. Par exemple, 5.2932313249 divisé par 11 est égal à 0,481228, ou, environ 48.12% de la variation est expliquée par cette première valeur propre. Le pourcentage cumulé expliqué est obtenu en ajoutant les proportions successives de variances expliquées. Par exemple, 48.12% plus 20.90% sont égaux à 69.02%, et ainsi de suite. Par conséquent, environ 69.02% de la variance totale est expliquée par les deux premières valeurs propres.

Les valeurs propres peuvent être utilisées pour déterminer le nombre d'axes principaux à conserver après l'ACP (Kaiser 1961) :

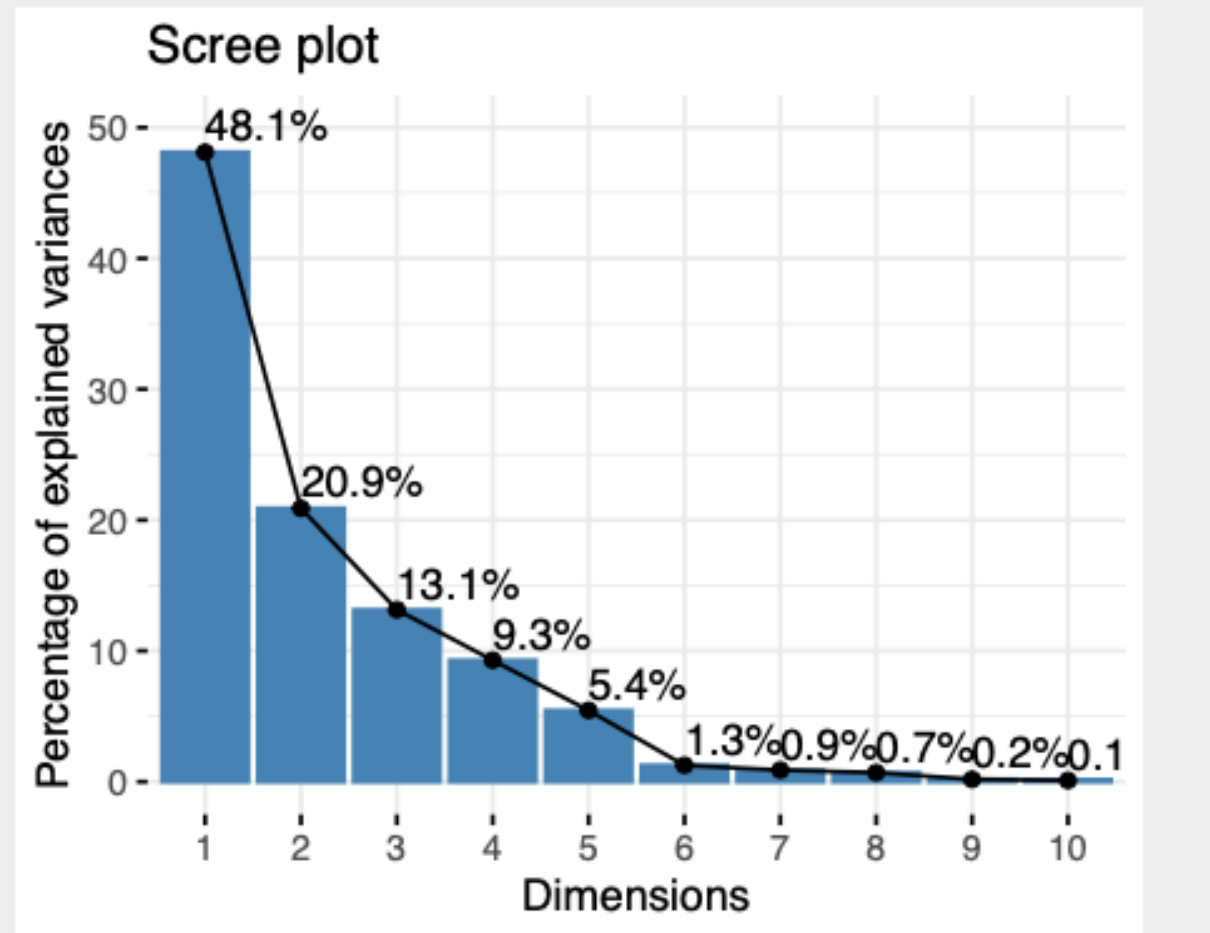
- Une *valeur propre* > 1 indique que la composante principale (PC) concernée représente plus de variance par rapport à une seule variable d'origine, lorsque les données sont standardisées. Ceci est généralement utilisé comme seuil à partir duquel les PC sont conservés. A noter que cela ne s'applique que lorsque les données sont normalisées.
- Vous pouvez également limiter le nombre d'axes à un nombre qui représente une certaine fraction de la variance totale. Par exemple, si vous êtes satisfaits avec 70% de la variance totale expliquée, utilisez le nombre d'axes pour y parvenir.

Malheureusement, il n'existe pas de méthode objective bien acceptée pour décider du nombre d'axes principaux qui suffisent. Cela dépendra du domaine d'application spécifique et du jeu de données spécifiques. Dans la pratique, on a tendance à regarder les premiers axes principaux afin de trouver des profils intéressants dans les données.

Dans notre analyse, les trois premières composantes principales expliquent 82% de la variation. C'est un pourcentage acceptable.

Une autre méthode pour déterminer le nombre de composantes principales est de regarder le graphique des valeurs propres (appelé scree plot). Le nombre d'axes est déterminé par le point, au-delà duquel les valeurs propres restantes sont toutes relativement petites et de tailles comparables (Jollife 2002, Peres-Neto, Jackson, and Somers (2005)).
Le graphique des valeurs propres peut être généré à l'aide de la fonction *fviz_eig* () ou *fviz_screeplot* () [package *factoextra*].

> fviz_eig(res.pca, addlabels = TRUE, ylim = c(0, 50))



Du graphique ci-dessus, nous pourrions vouloir nous arrêter à la troisième composante principale. 82% des informations(variances) contenues dans les données sont conservées par les trois premières composantes principales.

➢ GRAPHIQUE DES VARIABLES ET CORRELATION

Une méthode simple pour extraire les résultats, pour les variables, à partir de l'ACP est d'utiliser la fonction *get_pca_var*() [package *factoextra*]. Cette fonction retourne

une liste d'éléments contenant tous les résultats pour les variables actives (coordonnées, corrélation entre variables et les axes, cosinus-carré et contributions)

```
> var <- get_pca_var(res.pca)

> var

Principal Component Analysis Results for variables

================================================
===

  Name       Description

1 "$coord"   "Coordinates for the variables"

2 "$cor"     "Correlations between variables and dimensions"

3 "$cos2"    "Cos2 for the variables"

4 "$contrib" "contributions of the variables"
```

Les composants de get_pca_var() peuvent être utilisés dans le graphique des variables comme suit:

- var$coord: coordonnées des variables pour créer un nuage de points.
- var$cos2: *cosinus carré* des variables. Représente la qualité de représentation des variables sur le graphique de l'ACP. Il est calculé comme étant les coordonnées au carré: var.cos2 = var.coord * var.coord.
- var$contrib: contient les contributions (en pourcentage), des variables, aux composantes principales. La contribution d'une variable (var) à une composante principale donnée: (var.cos2 * 100) / (total cos2 du composant).

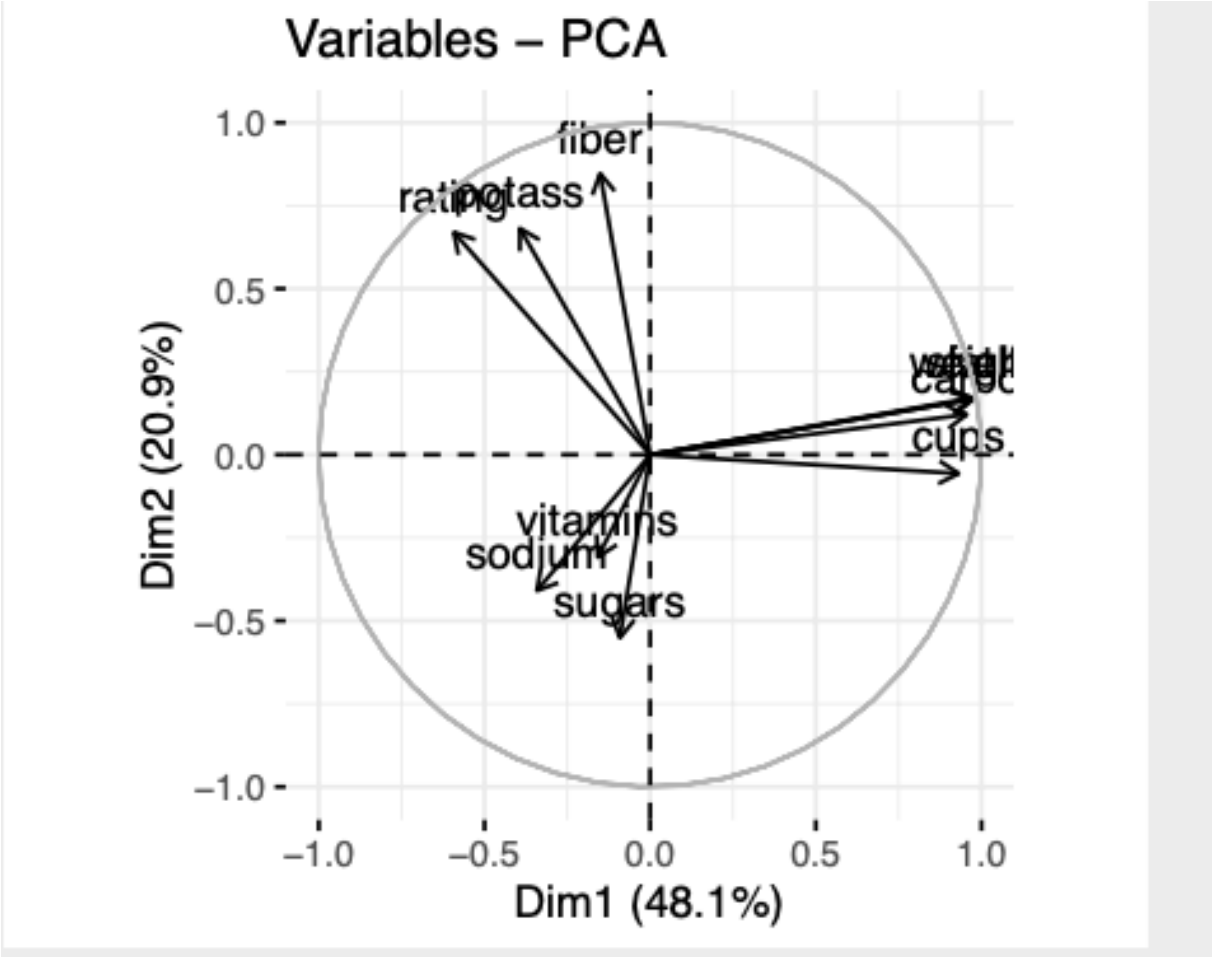
➢ CERCLE DE CORRELATION

La corrélation entre une variable et une composante principale (PC) est utilisée comme coordonnées de la variable sur la composante principale. La représentation des variables diffère de celle des observations : les observations sont représentées par leurs projections, mais les variables sont représentées par leurs corrélations (Abdi and Williams 2010).

COORDONNEES DES VARIABLES :

```
> head(var$coord)
              Dim.1      Dim.2      Dim.3       Dim.4       Dim.5
fat      0.96853805  0.1662277 0.11226273 -0.007337007  0.01805825
sodium -0.34164618 -0.4091250 0.46085412  0.429722454  0.56233812
fiber  -0.15113376  0.8466368 0.46823590 -0.074052796  0.04842415
carbo   0.95541397  0.1212646 0.07650349  0.112863863  0.03614667
sugars -0.09134475 -0.5511427 0.48072258 -0.651087203 -0.09307760
potass -0.39326644  0.6816200 0.55457659 -0.159376707  0.03371872
```

VISUALISATION DES VARIABLES



Le graphique ci-dessus est également connu sous le nom de graphique de corrélation des variables. Il montre les relations entre toutes les variables. Il peut être interprété comme suit :

- Les variables positivement corrélées sont regroupées.
- Les variables négativement corrélées sont positionnées sur les côtés opposés de l'origine du graphique (quadrants opposés).
- La distance entre les variables et l'origine mesure la qualité de représentation des variables. Les variables qui sont loin de l'origine sont bien représentées par l'ACP.

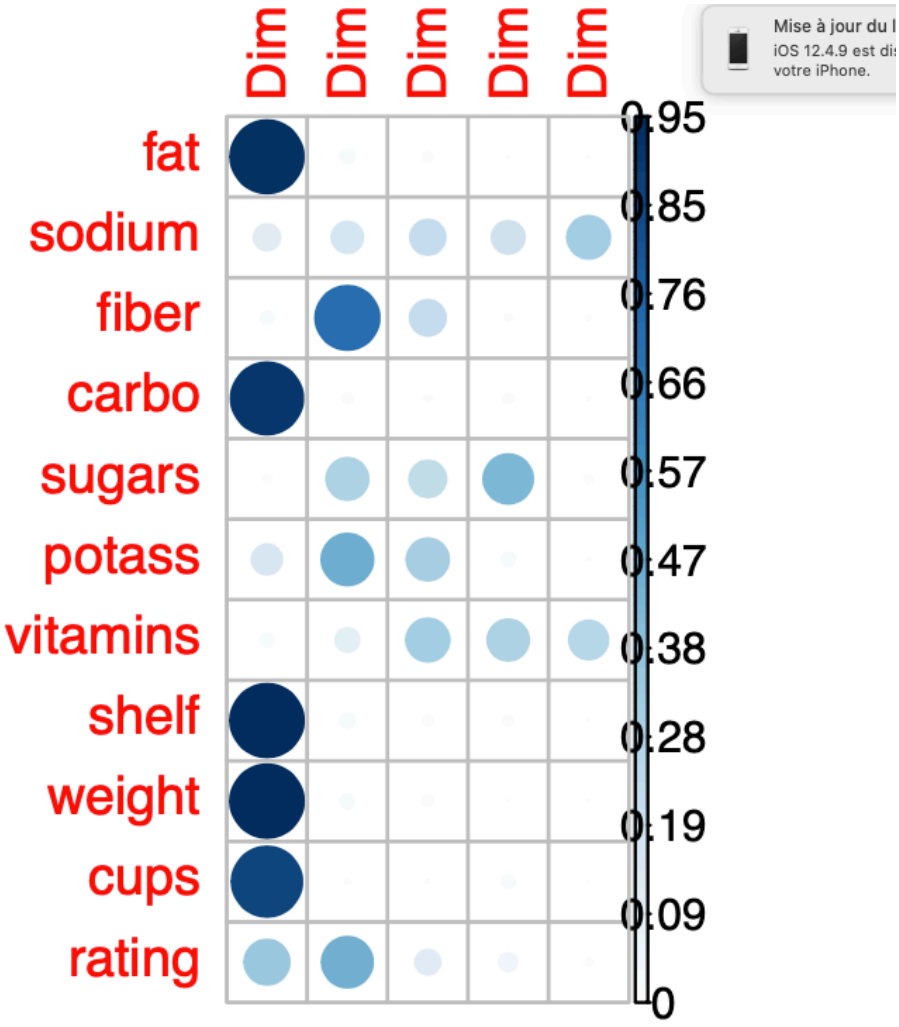
➢ QUALITE DE LA REPRESENTATION

La qualité de représentation des variables sur la carte de l'ACP s'appelle cos2 (*cosinus carré*). Nous pouvons accéder au cos2 comme suit :

```
> head(var$cos2)
              Dim.1      Dim.2       Dim.3        Dim.4        Dim.5
fat     0.938065962 0.02763164 0.012602920 5.383167e-05 0.0003261003
sodium  0.116722111 0.16738326 0.212386522 1.846614e-01 0.3162241617
fiber   0.022841413 0.71679393 0.219244858 5.483817e-03 0.0023448982
carbo   0.912815851 0.01470511 0.005852785 1.273825e-02 0.0013065817
sugars  0.008343863 0.30375825 0.231094199 4.239145e-01 0.0086634393
potass  0.154658490 0.46460581 0.307555194 2.540093e-02 0.0011369523
```

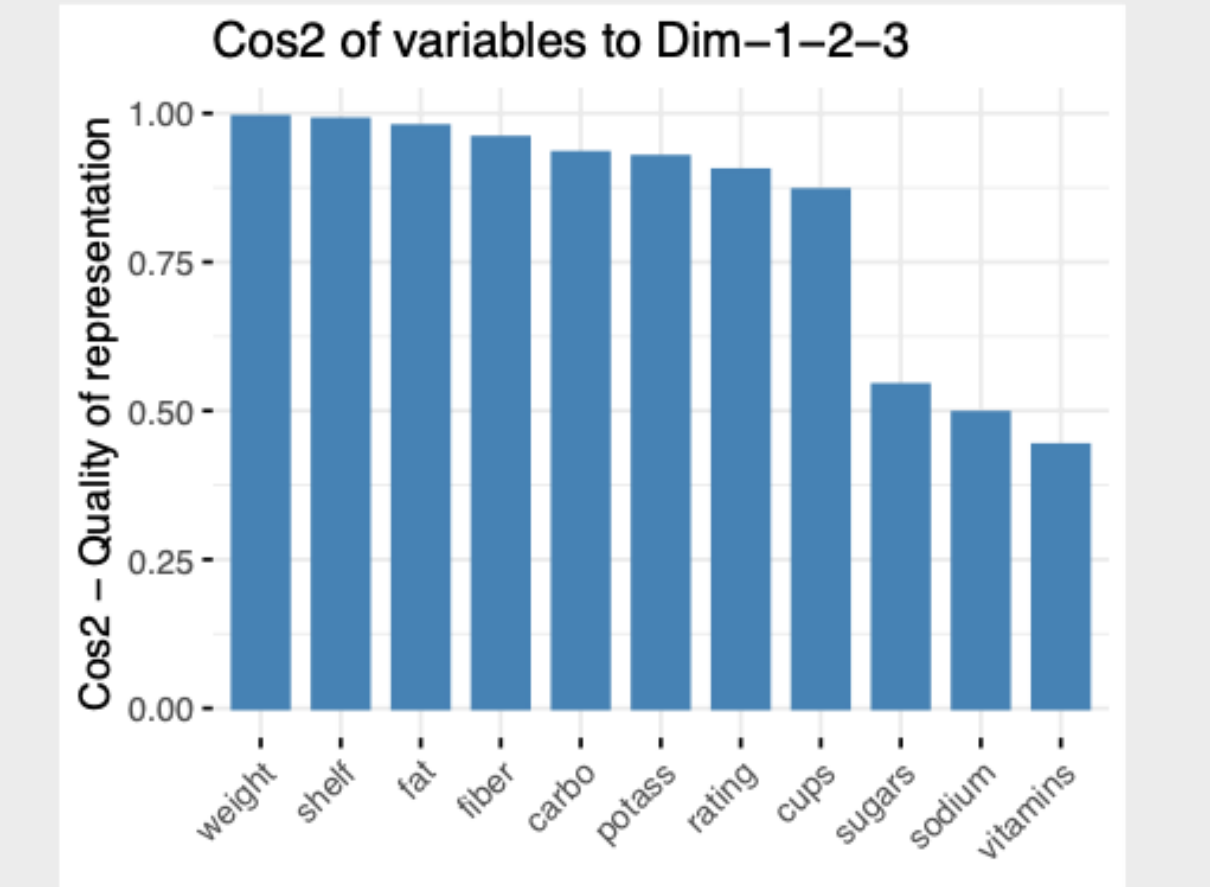
Nous pouvons visualiser le cos2 des variables sur toutes les dimensions en utilisant le package corrplot :

```
> library(corrplot)
> corrplot(var$cos2, is.corr=FALSE)
```



Il est également possible de créer un bar plot du cosinus carré des variables en utilisant la fonction fviz_cos2() [dans factoextra] pour une dimension d'ordre 3 comme nous l'explique les 82% des 03 premiers composantes principales :

> fviz_cos2(res.pca, choice = "var", axes = 1:3)



Notez que,

- Un cos2 élevé indique une bonne représentation de la variable sur les axes principaux en considération. Dans ce cas, la variable est positionnée à proximité de la circonférence du cercle de corrélation.
- Un faible cos2 indique que la variable n'est pas parfaitement représentée par les axes principaux. Dans ce cas, la variable est proche du centre du cercle.

Pour une variable donnée, la somme des cos2 sur toutes les composantes principales est égale à 1.

Si une variable est parfaitement représentée par seulement deux composantes principales (Dim.1 & Dim.2), la somme des cos2 sur ces deux axes est égale à 1. Dans ce cas, les variables seront positionnées sur le cercle de corrélation.

Pour certaines des variables, plus de 2 axes peuvent être nécessaires pour représenter parfaitement les données. Dans ce cas, les variables sont positionnées à l'intérieur du cercle de corrélation.

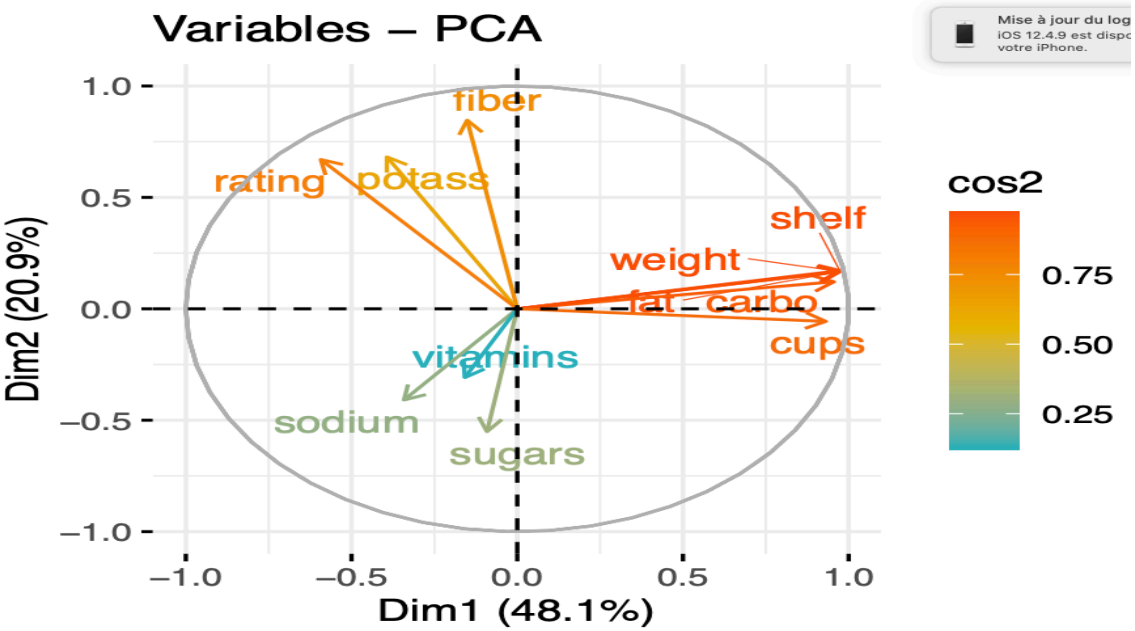
En résumé:

- Les valeurs de cos2 sont utilisées pour estimer la qualité de la représentation
- Plus une variable est proche du cercle de corrélation, meilleure est sa représentation sur la carte de l'ACP (et elle est plus importante pour interpréter les composantes principales en considération)
- Les variables qui sont proche du centre du graphique sont moins importantes pour les premières composantes.

Il est possible de colorer les variables en fonction de la valeur de leurs cos2 à l'aide de l'argument col.var = "cos2". Cela produit un gradient de couleurs. Dans ce cas, l'argument gradient.cols peut être utilisé pour spécifier une palette de couleur personnalisée. Par exemple, gradient.cols = c("white", "blue", "red") signifie que:

- les variables à faible valeur de cos2 seront colorées en "white" (blanc)
- les variables avec les valeurs moyennes de cos2 seront colorées en "blue" (bleu)
- les variables avec des valeurs élevées de cos2 seront colorées en "red" (rouge)

> fviz_pca_var(res.pca, col.var = "cos2",gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),repel = TRUE # Évite le chevauchement de texte))



➢ CONTRIBUTION DES VARIABLES AUX PRINCIPAUX AXES

Les contributions des variables dans la définition d'un axe principal donné, sont exprimées en pourcentage.

- Les variables corrélées avec PC1 (i.e., Dim.1), PC2 (i.e., Dim.2), PC3 (i.e., Dim.3) sont les plus importantes pour expliquer la variabilité dans le jeu de données.

- Les variables qui ne sont pas en corrélation avec un axe ou qui sont corrélées avec les derniers axes sont des variables à faible apport et peuvent être supprimées pour simplifier l'analyse globale.

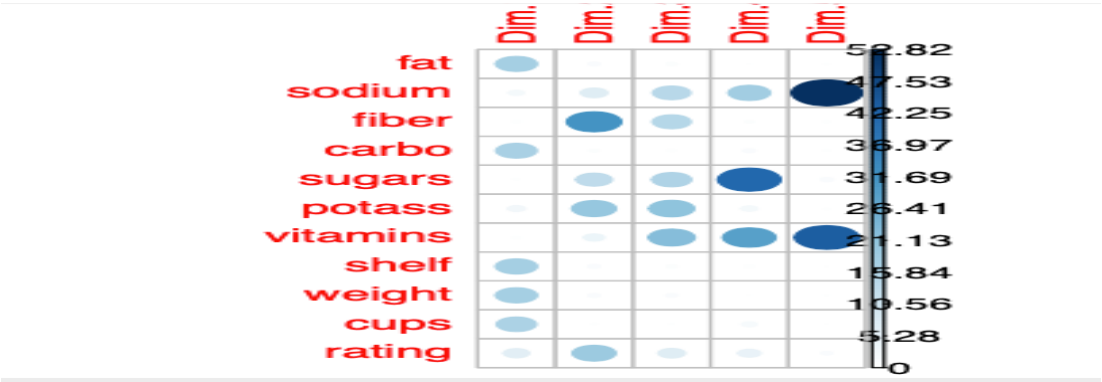
La contribution des variables peut être extraite comme suit :

```
> head(var$contrib)
              Dim.1      Dim.2      Dim.3       Dim.4      Dim.5
fat      17.7219907  1.2018167  0.8718508  0.005271325  0.0544652
sodium    2.2051202  7.2802055 14.6925764 18.082482254 52.8156968
fiber     0.4315212 31.1763977 15.1670256  0.536988370  0.3916444
carbo    17.2449643  0.6395873  0.4048867  1.247359890  0.2182250
sugars    0.1576327 13.2117304 15.9867450 41.510720594  1.4469659
potass    2.9218162 20.2076705 21.2762003  2.487319931  0.1898936
```

Notons que plus la valeur de la contribution est importante, plus la variable contribue à la composante principale en question.

Il est possible d'utiliser la fonction corrplot() [package corrplot] pour mettre en évidence les variables les plus contributives pour chaque dimension:

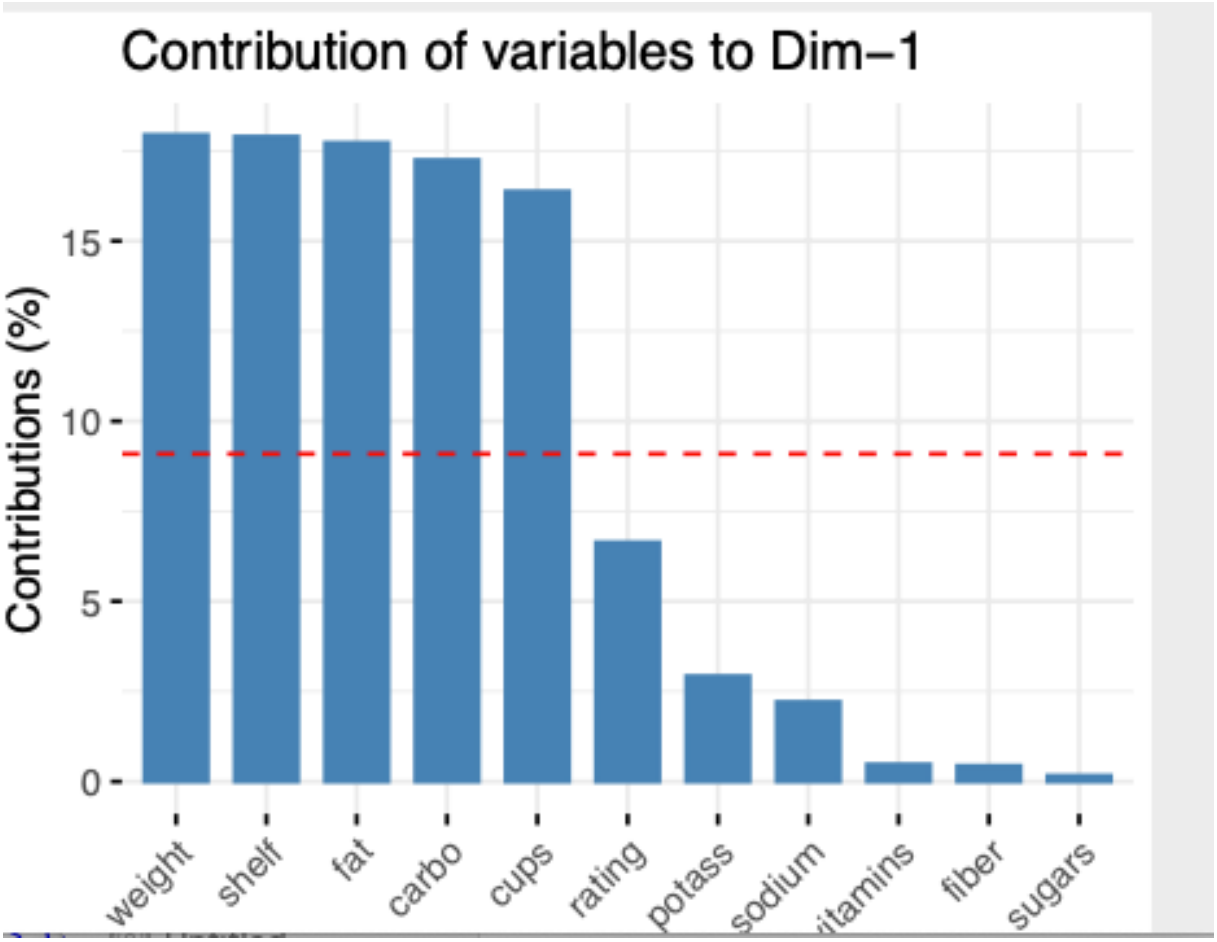
```
> corrplot(var$contrib, is.corr=FALSE)
```



La fonction fviz_contrib() [package factoextra] peut être utilisée pour créer un bar plot de la contribution des variables. Vu que nos données contiennent de nombreuses variables, nous pouvons décider de ne montrer que les principales variables contributives. Le code R ci-dessous montre le top 11 des variables contribuant le plus aux composantes principales :

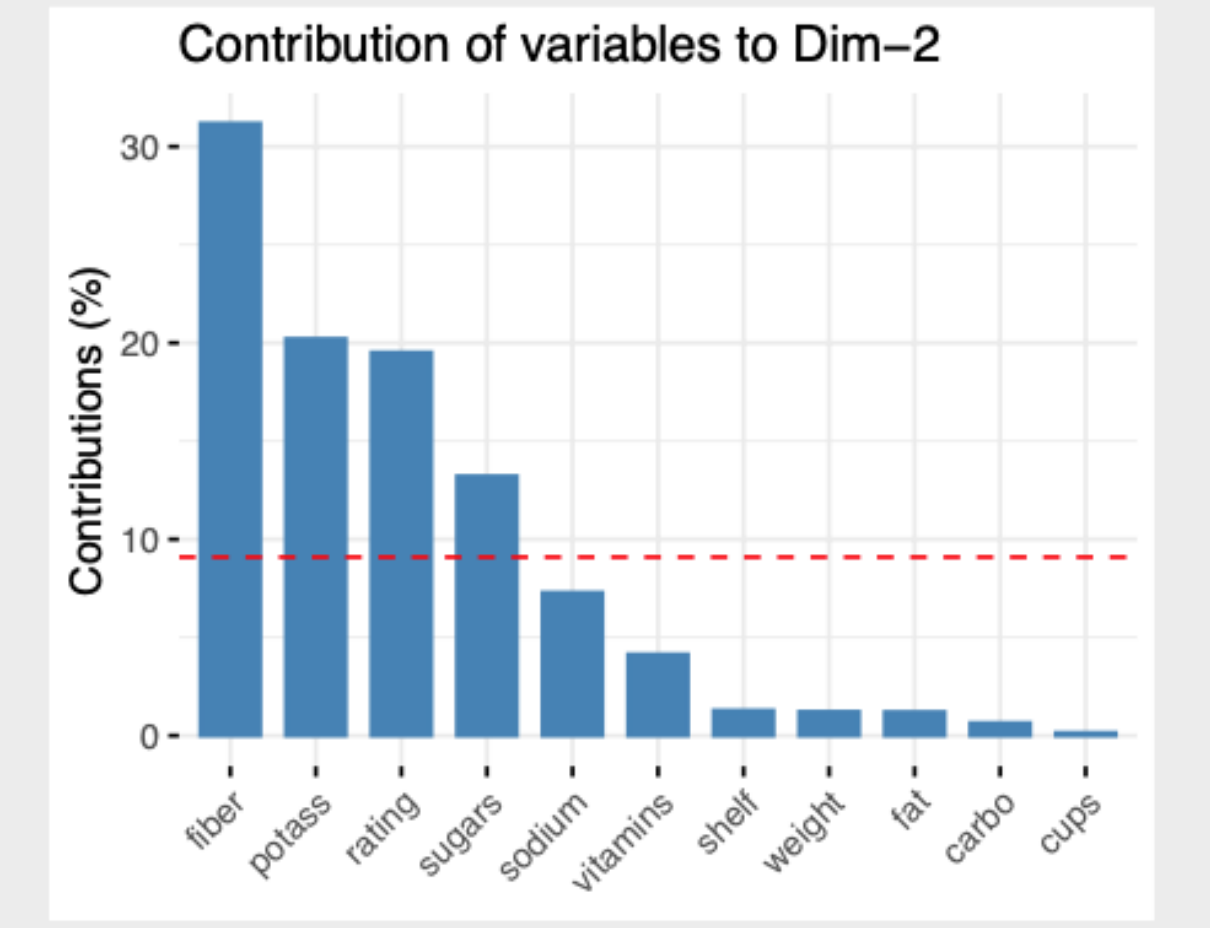
- ✓ Contribution des variables a pc1

> fviz_contrib(res.pca, choice = "var", axes = 1, top = 11)



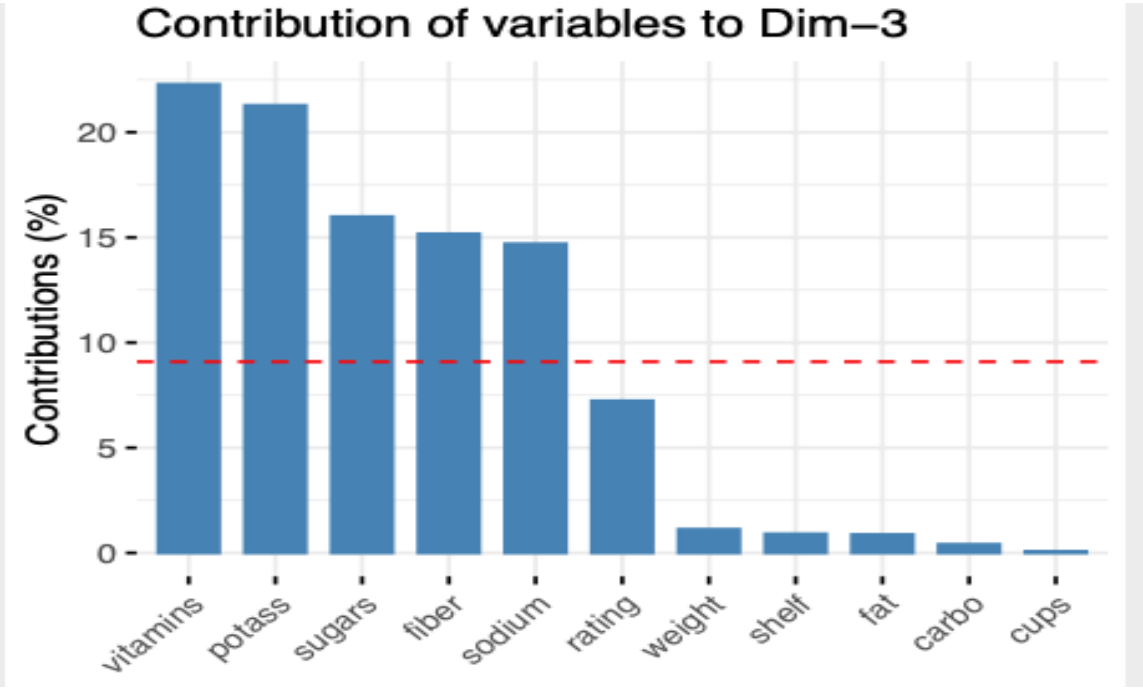
- ✓ Contribution des variables a pc2 :

> fviz_contrib(res.pca, choice = "var", axes = 2, top = 11)



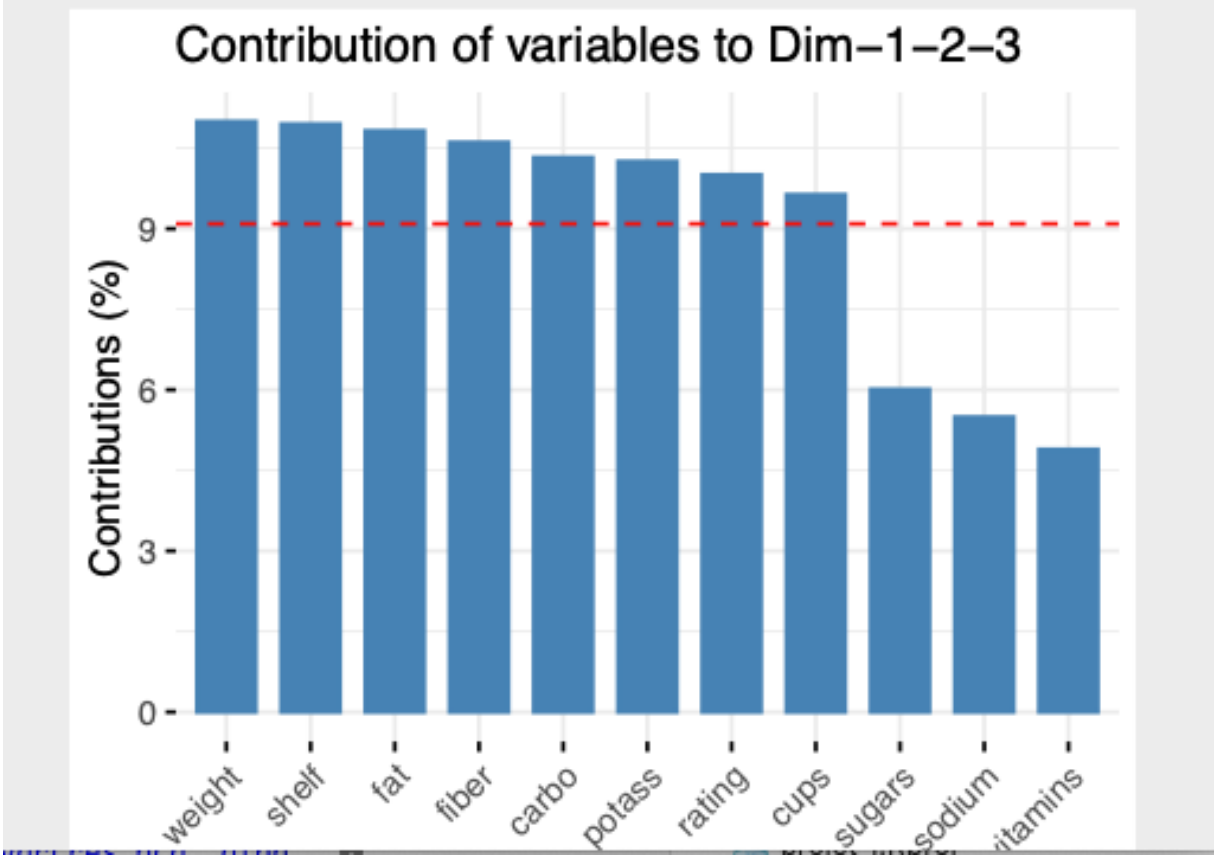
✓ Contribution des variables a pc3 :

> fviz_contrib(res.pca, choice = "var", axes = 3, top = 11)



La contribution totale à PC1, PC2 et PC3 est obtenue avec le code R suivant :

> fviz_contrib(res.pca, choice = "var", axes = 1:3, top = 11)



La ligne en pointillé rouge, sur le graphique ci-dessus, indique la contribution moyenne attendue. Si la contribution des variables était uniforme, la valeur attendue serait 1/length(variables) = 1/11 = 9%. Pour une composante donnée, une variable avec une contribution supérieure à ce seuil pourrait être considérée comme importante pour contribuer à la composante.

- **Significativités et correlation des variables avec la p-values**

La fonction dimdesc() [dans FactoMineR], pour dimension description (en anglais), peut être utilisée pour identifier les variables les plus significativement associées avec une composante principale donnée . Elle peut être utilisée comme suit :

```
> res.desc <- dimdesc(res.pca, axes = c(1,3), proba = 0.05)
> res.desc
$Dim.1
$quanti
         correlation     p.value
weight   0.9745392   1.879428e-50
```

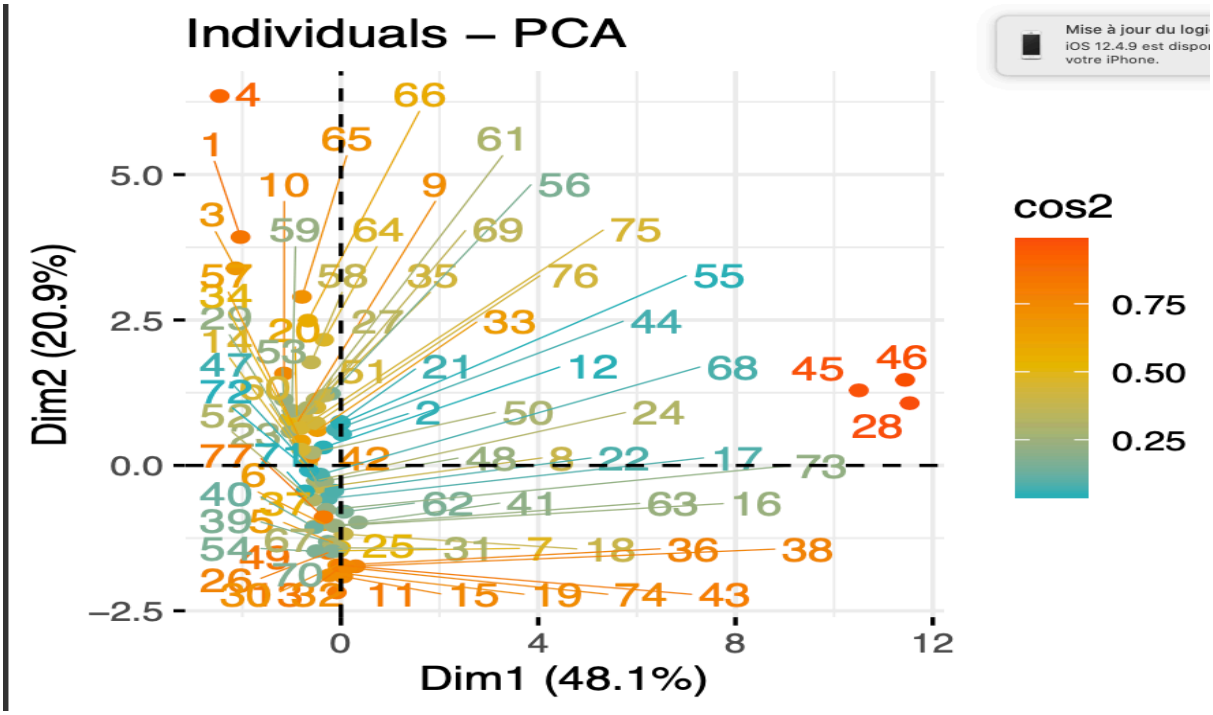
shelf 0.9732449 1.178839e-49
fat 0.9685381 4.718740e-47
carbo 0.9554140 1.772722e-41
cups 0.9309710 1.490889e-34
sodium -0.3416462 2.358238e-03
potass -0.3932664 4.029328e-04
rating -0.5925741 1.367430e-08

➢ GRAPHIQUE DES INDIVIDUS

> ind <- get_pca_ind(res.pca)
> ind
Principal Component Analysis Results for individuals

==========
===
Name Description
1 "$coord" "Coordinates for the individuals"
2 "$cos2" "Cos2 for the individuals"
3 "$contrib" "contributions of the individuals"

> fviz_pca_ind (res.pca, col.ind = "cos2",
+ gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
+ repel = TRUE # Évite le chevauchement de texte
+)

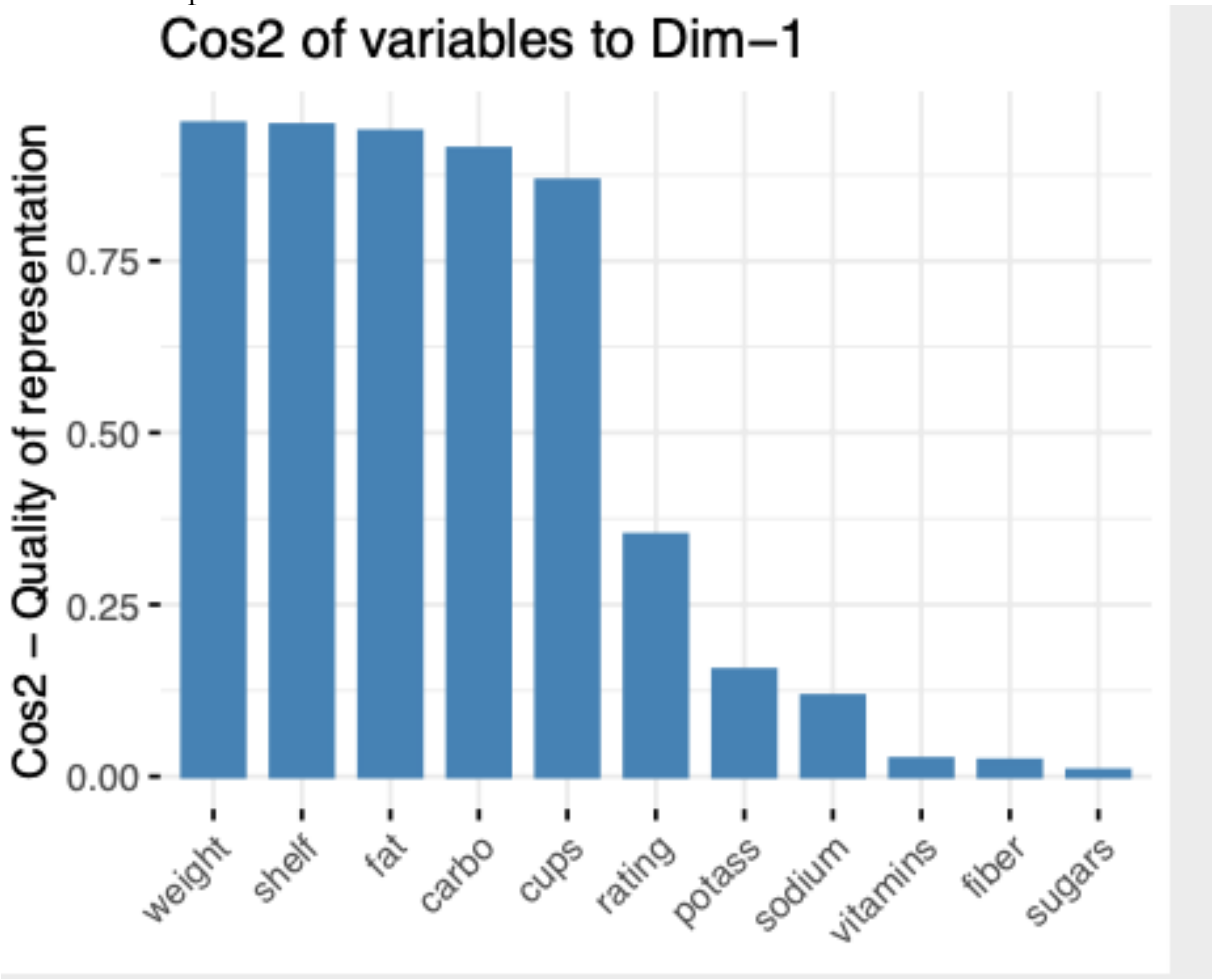


- **REGRESSION POUR LES COMPOSANTES PRINCIPALES**

82% des informations sont expliquées par les 03 premières composantes principales.

Ainsi on peut se baser pour chaque composante afin d'en déduire sa régression suivant la qualité de représentation du cos^2.

Pour la composante PCA1 :



par application :

weight = ß0 + ß1(shelf) + ß2(fat) + ß3(carbo) + ß4(cups) + ß5(rating) + ß6(potass) + ß7(sodium) + ß8(vitamins) + ß9(fiber) + ß10(sugars) + E_i

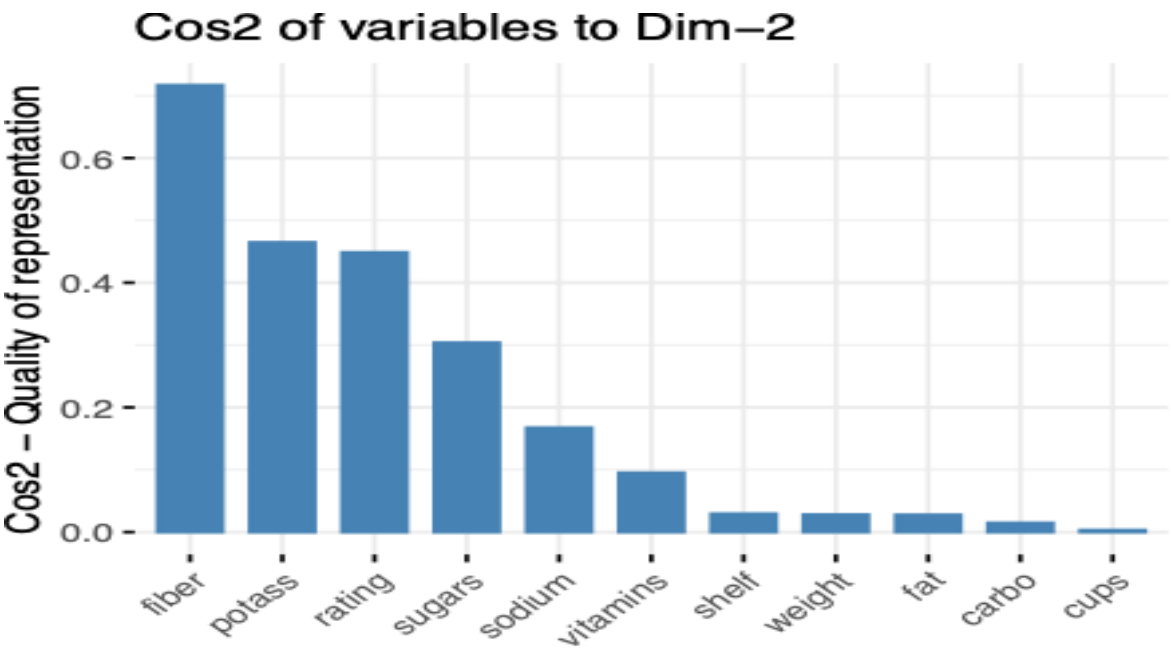
Sous R nous obtenons la sortie de commande suivante :

>composante1=lm(weight~shelf+fat+carbo+cups+rating+potass+sodium+vitamins+fiber+sugars,data=cereales)
> summary(composante1)

Call:

```
lm(formula = weight ~ shelf + fat + carbo + cups + rating + potass +
    sodium + vitamins + fiber + sugars, data = cereales)

Residuals:
     Min       1Q   Median       3Q      Max
-0.32285 -0.06992 -0.00154  0.07275  0.44334

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.3120639  0.2291311  -1.362  0.17785
shelf        0.0692315  0.0035470  19.518  < 2e-16 ***
fat          0.0825928  0.0034402  24.008  < 2e-16 ***
carbo        0.0071542  0.0028172   2.539  0.01346 *
cups         0.0855314  0.0780242   1.096  0.27697
rating       0.0119940  0.0039446   3.041  0.00338 **
potass       0.0011221  0.0005518   2.033  0.04605 *
sodium       0.0009125  0.0002848   3.204  0.00209 **
vitamins     0.0004168  0.0007246   0.575  0.56714
fiber       -0.0416451  0.0231624  -1.798  0.07676 .
sugars       0.0364656  0.0081567   4.471 3.14e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1271 on 66 degrees of freedom
Multiple R-squared:  0.9994,    Adjusted R-squared:  0.9993
F-statistic: 1.027e+04 on 10 and 66 DF,  p-value: < 2.2e-16
```

Pour la composante PCA 2:

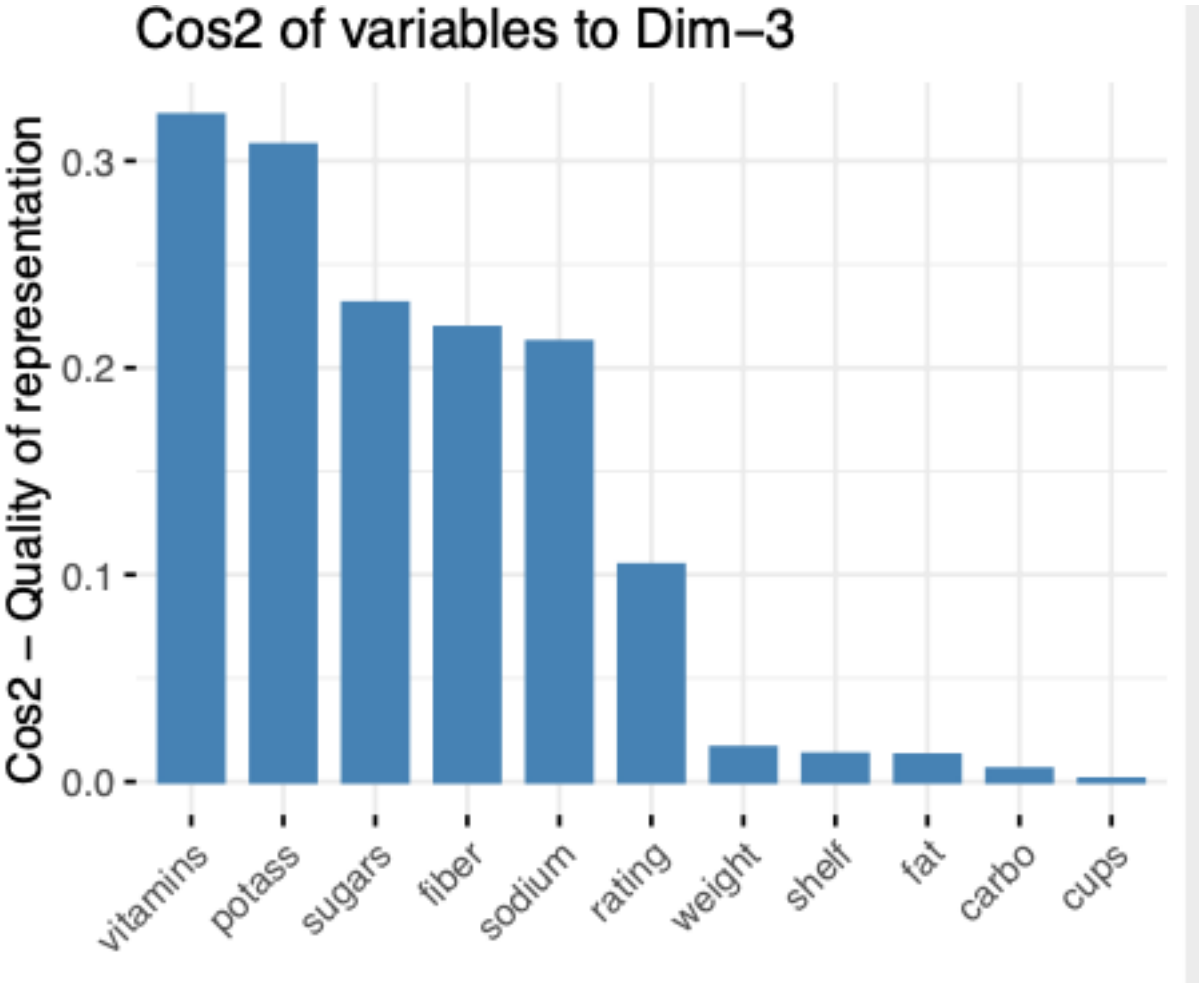


fiber= ß0 + ß1(shelf) + ß2(fat) + ß3(carbo) + ß4(cups) + ß5(rating) + ß6(potass) + ß7(sodium) + ß8(vitamins) + ß9(weight) + ß10(sugars) + E_i

par application sous R:

```
>composante2=lm(fiber~shelf+fat+carbo+cups+rating+potass+sodium+vitamin
s+weight+sugars,data=cereales)
> summary(composante2)

Call:
lm(formula = fiber ~ shelf + fat + carbo + cups + rating + potass +
    sodium + vitamins + weight + sugars, data = cereales)

Residuals:
     Min       1Q   Median       3Q      Max
-1.58511 -0.41987  0.02988  0.38573  1.99563

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.328904   0.919949  -6.880 2.65e-09 ***
shelf        0.118085   0.045636   2.588  0.01187 *
fat          0.142529   0.052854   2.697  0.00888 **
carbo       -0.025510   0.014990  -1.702  0.09349 .
cups        -0.670426   0.400093  -1.676  0.09853 .
rating       0.131317   0.014708   8.929 5.82e-13 ***
potass       0.018917   0.001814  10.429 1.36e-15 ***
sodium       0.007183   0.001320   5.443 8.26e-07 ***
vitamins     0.003445   0.003745   0.920  0.36103
weight      -1.121204   0.623598  -1.798  0.07676 .
sugars       0.216319   0.040307   5.367 1.11e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6593 on 66 degrees of freedom
Multiple R-squared:  0.9323,       Adjusted R-squared:  0.922
F-statistic: 90.84 on 10 and 66 DF,  p-value: < 2.2e-16
```

Pour la composante PCA3:



vitamins= ß0 + ß1(shelf) + ß2(fat) + ß3(carbo) + ß4(cups) + ß5(rating) + ß6(potass) + ß7(sodium) + ß8(fiber) + ß9(weight) + ß10(sugars) + E_i

par application sous R:

>composante3=lm(vitamins~shelf+fat+carbo+cups+rating+potass+sodium+fiber+fiber+sugars,data=cereales)
> summary(composante3)

Call:
lm(formula = vitamins ~ shelf + fat + carbo + cups + rating +
    potass + sodium + fiber + fiber + sugars, data = cereales)

Residuals:

Min	1Q	Median	3Q	Max
-29.117	-10.769	-6.230	1.187	67.064

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	43.45075	38.26509	1.136	0.260
shelf	-0.42880	0.59573	-0.720	0.474
fat	-0.51176	0.57663	-0.887	0.378
carbo	0.47423	0.47143	1.006	0.318
cups	16.17874	13.00556	1.244	0.218
rating	-0.94095	0.65504	-1.436	0.156
potass	0.02985	0.09297	0.321	0.749
sodium	0.03118	0.04786	0.651	0.517
fiber	3.19148	3.88568	0.821	0.414
sugars	-1.17067	1.36776	-0.856	0.395

Residual standard error: 21.42 on 67 degrees of freedom
Multiple R-squared: 0.2078, Adjusted R-squared: 0.1014
F-statistic: 1.953 on 9 and 67 DF, p-value: 0.05905

- **<u>TABLEAU DE RESULTAT DES COMPOSANTES PRINCIPALES A 03 DIMENSIONS</u>**

	R^2	**R^2 ajusté**	**P-value Fisher**
Composante 1	0,9994	0,9993	$2{,}2e^{-16}$
Composante 2	0,9323	0,922	$2{,}2e^{-16}$
Composante 3	0,2078	0,1014	0,05905

La composantes 1 est le meilleur modèle selon le critère d'ajustement du R^2. La pertinence de la régression est évaluée à 99,94% et cette dernière est globalement significative au seuil de 5% car la statistique p-value de Fischer < 0,05.
La composante 2 est dans la même logique de pertinence suivant un R^2 ajusté estime a 92,2% que celle de la composante 1. Son modèle est

globalement significatif au seuil de 5% tout comme la composante 1avec une probabilité inférieure au seuil des 5%.

Quant à la composante 3 ; c'est le modèle le moins pertinent des 03 car sa pertinence est moindrement expliquée par les variables prédictives. Son ajustement est de 10,14% et sa probabilité associée a la statistique de Fischer est supérieure à 0,05. En conclusion, le modèle n'est pas globalement significatif au seuil de 5%.
La variable vitamins est faiblement expliquée par les variables prédictives telles que fat, sodium, fiber, carbo, sugars, potass, shelf, weight, cups et rating.

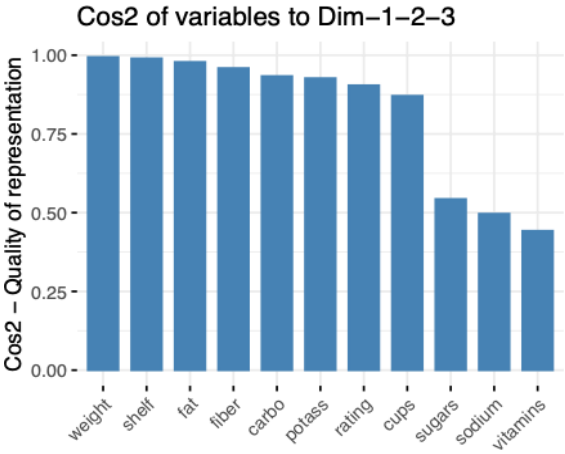
CONCLUSION GENERALE

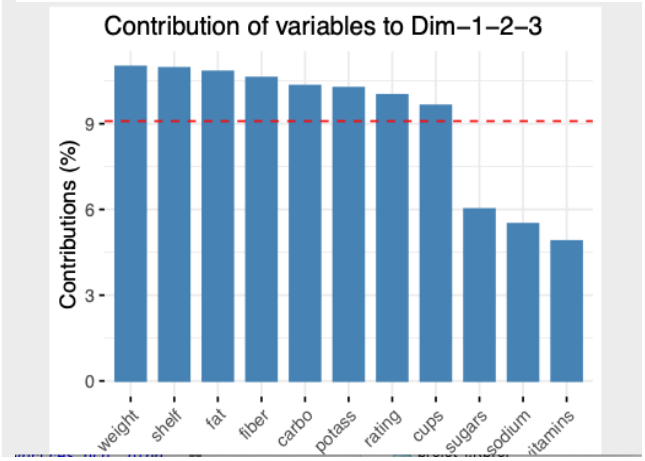
En somme, nous avons étudié les données céréales en 02 thèmes distinctes : régression linéaire simple et multiple.
Dans chacun des thèmes, des résultats ont été obtenus. De là nous pouvons juger que notre modèle de régression simple nous a fourni des résultats globalement satisfaisants car seule la normalité des résidus a été rejetée **(p-value selon la statistique de shapiro-wilk p-value = 0.006552 < 0,05),** par ailleurs les tests d'hypothèses de linéarité, d'homogénéité et d'indépendance des résidus ont été tous acceptés.

La régression multiple quant à elle s'est avérée plus complexe. Cela s'explique en partie par la typologie de la base céréales avec des variables a caractères divers (variables de type caractère () et type double ()). Mais néanmoins avec une bonne gestion de la base suivant une conversion de certaines variables en format numérique, l'application de régression multiple devient simple. Ainsi nos résultats s'avèrent pertinents avec comme variables endogène **la variable fat** qui évalue notre régression multiple a **99% suivant son R^2**. Ce qui justifie la présence possible de multi colinéarité entre nos variables explicatives.

Notre étude sera bouclée par une analyse en composante principale qui qui nous permettra d'avoir des informations sur les relations entre nos

différentes variables quantitatives de la base et d'en déduire la pertinence et la contribution de chacune d'elles dans notre étude.
De façon générale, **l'information est donnée a 80%** par nos trois premières composantes principales et globalement les variables les plus pertinentes sont **weight, shelf, fat, fiber** qui ont une plus grande contribution et donnent une meilleure qualité de représentation des données comme nous le stipule les graphes à 03 dimensions ci-dessous :





BONNE RECEPTION !!!!
« La réussite au bout de l'effort »
« l'effort fait les forts »