TD3

LEGRAND.Axel_SY.Abiboulaye

16/10/2020

1. Combien de lignes et colonnes

On commence par charger le jeu de données.L'instruction help(mtcars) nous renvoie qu'il y a 32 lignes et 11 colonnes

```
data("mtcars")
#View(mtcars)
```

2. Données manquantes ?

```
sum(is.na(mtcars))
```

```
## [1] 0
```

Il n'y a pas de valeurs manquantes

3.Nature des variables

```
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

Le dataframe contient 32 observations pour 11 variables. Certaines variables ne sont pas quantitatives et sont des variables qualitatives. Elles sont représentées par deux(0,1),trois et même plusieurs chiffres.C'est le cas de la variable am qui signifie(automatique ou manuel)et vs qui a deux modalités aussi, gear est une variable qualitative à 3 modalités (3,4,5)ainsi que gear (4,6,8) et carb qui a 6 modalités. Tout ceci est obtenu avec la fonction help(mtcars)

Pour ces 5 variables nous allons les transfromer par leur nature en utilisant la fonction mtcars2 de la fonction help

```
mtcars2 <- within(mtcars,{
            vs <- factor(vs, labels =c("V","S"))
            am <- factor(am, labels = c("automatic","manuel"))
            cyl <- ordered(cyl)
            gear <- ordered(gear)
            carb <- ordered(carb)
})
summary(mtcars2)
```

```
##       mpg        cyl         disp             hp             drat      
##  Min.   :10.40   4:11   Min.   : 71.1   Min.   : 52.0   Min.   :2.760  
##  1st Qu.:15.43   6: 7   1st Qu.:120.8   1st Qu.: 96.5   1st Qu.:3.080  
##  Median :19.20   8:14   Median :196.3   Median :123.0   Median :3.695  
##  Mean   :20.09          Mean   :230.7   Mean   :146.7   Mean   :3.597  
##  3rd Qu.:22.80          3rd Qu.:326.0   3rd Qu.:180.0   3rd Qu.:3.920  
##  Max.   :33.90          Max.   :472.0   Max.   :335.0   Max.   :4.930  
##        wt             qsec       vs              am     gear   carb    
##  Min.   :1.513   Min.   :14.50   V:18   automatic:19   3:15   1: 7  
##  1st Qu.:2.581   1st Qu.:16.89   S:14   manuel   :13   4:12   2:10  
##  Median :3.325   Median :17.71                         5: 5   3: 3  
##  Mean   :3.217   Mean   :17.85                                4:10  
##  3rd Qu.:3.610   3rd Qu.:18.90                                6: 1  
##  Max.   :5.424   Max.   :22.90                                8: 1
```

4. dans quel pays se situe ce jeu de données ? (les unités ?)

En tapant la ligne de commande: ?mtcars, on nous renseigne que ce jeu de données est extrait aux états-unis («The data was extracted from the 1974 Motor Trend US magazine») et les unités sont obtenues avec cette même ligne de commande

[, 1] mpg Miles/(US) gallon [, 2] cyl Number of cylinders [, 3] disp Displacement (cu.in.) [, 4] hp Gross horsepower [, 5] drat Rear axle ratio [, 6] wt Weight (1000 lbs) [, 7] qsec 1/4 mile time [, 8] vs Engine (0 = V-shaped, 1 = straight) [, 9] am Transmission (0 = automatic, 1 = manual) [,10] gear Number of forward gears [,11] carb Number of carburetors

```
?mtcars
```

```
## starting httpd help server ... done
```

5 . Calculer la matrice des corrélations linéaires de Pearson en me faisant apparaître la variable non pas en première ligne mais en dernière ligne.

On applique la fonction reverse(rev) à notre dataframe pour faire apparaître la variable mpg en dernière ligne

```
(cor(rev(mtcars[])))
```

```
##             carb       gear          am         vs        qsec         wt
## carb  1.00000000  0.2740728  0.05753435 -0.5696071 -0.65624923  0.4276059
## gear  0.27407284  1.0000000  0.79405876  0.2060233 -0.21268223 -0.5832870
## am    0.05753435  0.7940588  1.00000000  0.1683451 -0.22986086 -0.6924953
## vs   -0.56960714  0.2060233  0.16834512  1.0000000  0.74453544 -0.5549157
## qsec -0.65624923 -0.2126822 -0.22986086  0.7445354  1.00000000 -0.1747159
## wt    0.42760594 -0.5832870 -0.69249526 -0.5549157 -0.17471588  1.0000000
## drat -0.09078980  0.6996101  0.71271113  0.4402785  0.09120476 -0.7124406
## hp    0.74981247 -0.1257043 -0.24320426 -0.7230967 -0.70822339  0.6587479
## disp  0.39497686 -0.5555692 -0.59122704 -0.7104159 -0.43369788  0.8879799
## cyl   0.52698829 -0.4926866 -0.52260705 -0.8108118 -0.59124207  0.7824958
## mpg  -0.55092507  0.4802848  0.59983243  0.6640389  0.41868403 -0.8676594
##             drat         hp       disp        cyl        mpg
## carb -0.09078980  0.7498125  0.3949769  0.5269883 -0.5509251
## gear  0.69961013 -0.1257043 -0.5555692 -0.4926866  0.4802848
## am    0.71271113 -0.2432043 -0.5912270 -0.5226070  0.5998324
## vs    0.44027846 -0.7230967 -0.7104159 -0.8108118  0.6640389
## qsec  0.09120476 -0.7082234 -0.4336979 -0.5912421  0.4186840
## wt   -0.71244065  0.6587479  0.8879799  0.7824958 -0.8676594
## drat  1.00000000 -0.4487591 -0.7102139 -0.6999381  0.6811719
## hp   -0.44875912  1.0000000  0.7909486  0.8324475 -0.7761684
## disp -0.71021393  0.7909486  1.0000000  0.9020329 -0.8475514
## cyl  -0.69993811  0.8324475  0.9020329  1.0000000 -0.8521620
## mpg   0.68117191 -0.7761684 -0.8475514 -0.8521620  1.0000000
```

On peut constater que les variables wt,cyl et disp sont fortement corrélées avec la variable mpg avec une corrélation négative. qsec est la variable qui a la corrélation la plus faible avec mpg suivie de gear

6.Calculer les tests de corrélation linéaire associés à la question 5). Que concluez-vous ?

Par définition de la régression linéaire multiple, nous avons une variable quantitative réponse qui sera expliquée par deux ou plusieurs variables quantitaves indépendantes. On va voir s'il existe une corrélation entre cette réponse et les autres variables quantitatives On commence par créer un sous ensemble contenant que les variables quantitatives avant d'appliquer la fonction mvn et le test de Mardia

Soit mtcars_quanti la nouvelle variable ne contenant que les variables quantitavives continues

```
mtcars_quanti <- mtcars[,c(1,3,4,5,6,7)]

library(MVN)

## Warning: package 'MVN' was built under R version 3.6.3

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
## sROC 0.1-2 loaded
```

```
result <- mvn(data = mtcars_quanti,mvnTest ="mardia")
result$multivariateNormality
```

```
##
##              Test         Statistic            p value Result
## 1 Mardia Skewness  77.6519454787057 0.0293424460277968     NO
## 2 Mardia Kurtosis 0.241701622075929   0.80901137258975    YES
## 3             MVN              <NA>               <NA>     NO
##
```

Commentaire :Les résultats du test sont ceux obtenus avec la commande result$multivariateNormality.

Nous constatons que le test de Mardia Skewness rejette l'hypothèse de multinormalité contrairement au test de Mardia Kurtosis. Et la dernière ligne de test rejette l'hypothèse de multinormalité donc nous allons utiliser un test de permutation avec une correction de Bonferroni, donc la fonction perm.cor.test du package ModStatR. Comme il y a 6 variables et 2 couples à choisir pour faire le test, le nombre de tests possibles n'est rien d'autre qu'une combinaison de 2 parmi 6. Soit nombres de test = 6C2 = 15. On obtient ce résultat avec la fonction choose qui permet de réaliser des combinaisons

```
library(ModStatR)
```

```
## Warning: package 'ModStatR' was built under R version 3.6.3
```

```
library(corrplot)
```

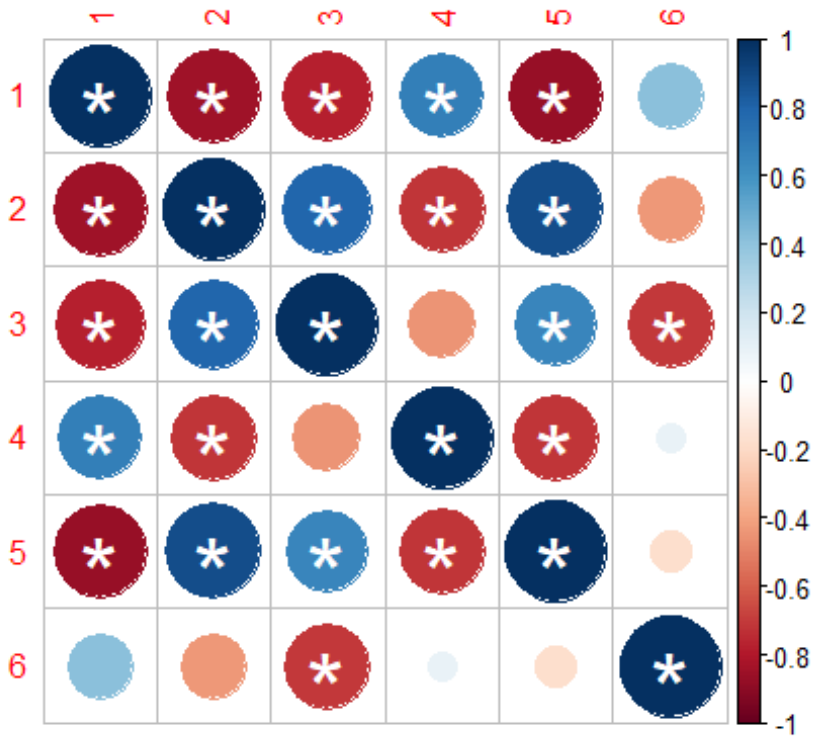
```
## Warning: package 'corrplot' was built under R version 3.6.3
```

```
## corrplot 0.84 loaded
```

```
permmtcars <- perm.cor.mtest(mtcars_quanti,num.sim=5000)
permmtcars$p < 0.05/choose(ncol(mtcars_quanti),2)
```

```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
## [1,]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
## [2,]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
## [3,]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
## [4,]  TRUE  TRUE FALSE  TRUE  TRUE FALSE
## [5,]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
## [6,] FALSE FALSE  TRUE FALSE FALSE  TRUE
```

```
corrplot(permmtcars$cor,p.mat=permmtcars$p,pch.col="white",insig="label_sig",
        sig.level=0.05/choose(ncol(mtcars_quanti),2))
```



LA sortie de la commande permmtcars$p montrent que les variables qui sont significativement liés avec mpg représenté par [1,].

On conclut que toutes les variables sont liées significativement à la variable mpg sauf **qsec**. Par conséquent nous ne l'utiliserons pas dans notre modèle

6.Déterminer le meilleur modèle explicatif et expliquer comment vous l'obtenez ? Quels sont les critères que vous avez choisi pour le sélectionner et pourquoi ?

```
model3456 <- lm(mpg~disp+hp+drat+wt,data =mtcars_quanti)
model3456

##
## Call:
## lm(formula = mpg ~ disp + hp + drat + wt, data = mtcars_quanti)
##
## Coefficients:
## (Intercept)         disp           hp         drat           wt
##   29.148738     0.003815    -0.034784     1.768049    -3.479668

library(olsrr)

## Warning: package 'olsrr' was built under R version 3.6.3

##
## Attaching package: 'olsrr'
```

```
## The following object is masked from 'package:datasets':
##
##     rivers
```

```
ols_step_best_subset(model3456)
```

```
##    Best Subsets Regression
## ----------------------------
## Model Index    Predictors
## ----------------------------
##      1         wt
##      2         hp wt
##      3         hp drat wt
##      4         disp hp drat wt
## ----------------------------
##
##                                                        Subsets Regression Summ
ary
## ------------------------------------------------------------------------------
----------------------------------------------------------
##                         Adj.        Pred
## Model    R-Square    R-Square    R-Square     C(p)        AIC        SBIC
SBC         MSEP       FPE       HSP       APC
## ------------------------------------------------------------------------------
----------------------------------------------------------
##   1        0.7528      0.7446      0.7087    13.1004    166.0294    74.233
6    170.4266    296.9167    9.8572    0.3199    0.2801
##   2        0.8268      0.8148      0.7811     2.8031    156.6523    66.481
6    162.5153    215.5104    7.3563    0.2402    0.2091
##   3        0.8369      0.8194      0.7816     3.1247    156.7311    67.292
3    164.0598    210.4688    7.3801    0.2430    0.2097
##   4        0.8376      0.8136      0.7641     5.0000    158.5837    69.554
9    167.3781    217.5591    7.8298    0.2605    0.2225
## ------------------------------------------------------------------------------
----------------------------------------------------------
## AIC: Akaike Information Criteria
##  SBIC: Sawa's Bayesian Information Criteria
##  SBC: Schwarz Bayesian Criteria
##  MSEP: Estimated error of prediction, assuming multivariate normality
##  FPE: Final Prediction Error
##  HSP: Hocking's Sp
##  APC: Amemiya Prediction Criteria
```

Essayons d'avoir les graphiques

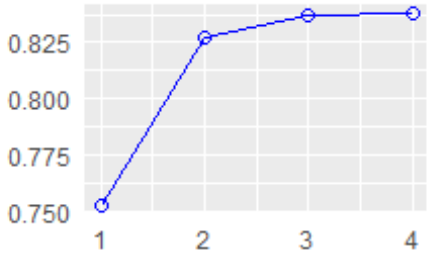
```
cor.test(mtcars$wt,mtcars$mpg)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mtcars$wt and mtcars$mpg
## t = -9.559, df = 30, p-value = 1.294e-10
```

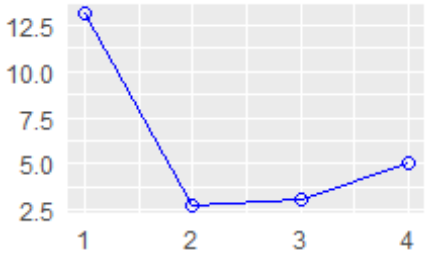
```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9338264 -0.7440872
## sample estimates:
##        cor
## -0.8676594

var <- ols_step_best_subset(model3456)
plot(var)
```

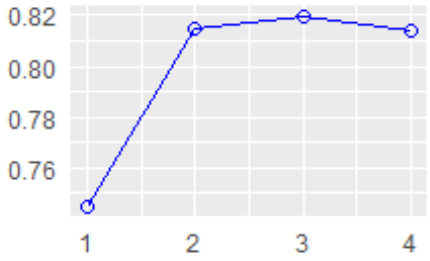
R-Square



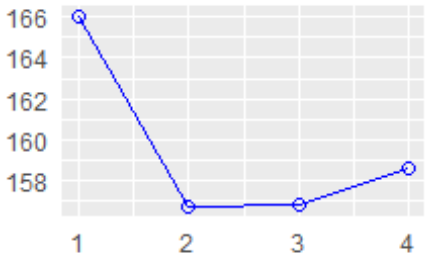
C(p)



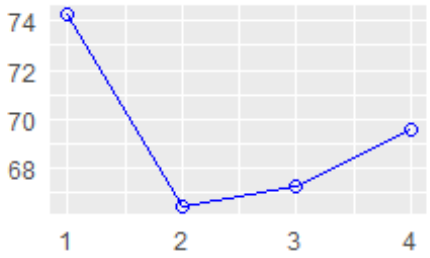
Adj. R-Square



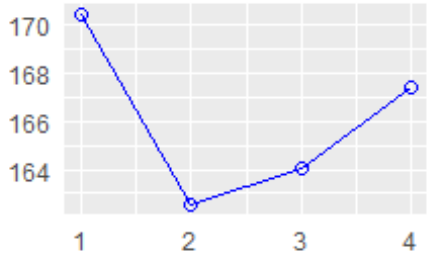
AIC



SBIC



SBC



Commentaire : On cherche à déterminer le meilleur modèle explicatif, comme le BIC est un critère explicatif par définition , on en déduit que le modèle 2 est le meilleur modèle explicatif car ayant le BIC le plus faible

Etudions le modèle 2

```
model2<- lm(mpg~ hp+wt,data =mtcars_quanti)
shapiro.test(residuals(model2))

##
##  Shapiro-Wilk normality test
##
## data:  residuals(model2)
## W = 0.92792, p-value = 0.03427
```

La p-valeur est inférieure à 0.05 donc le test sur la normalité des erreurs est significatif. Par conséquent, nous rejettons l'hypothèse de normalité des erreurs (H0) et nous décidons que les erreurs ne suivent pas une loi normale(hypothèse H1) avec un risque d'erreur alpha

```
summary(model2)

##
## Call:
## lm(formula = mpg ~ hp + wt, data = mtcars_quanti)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.941 -1.600 -0.182  1.050  5.854
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 37.22727    1.59879  23.285  < 2e-16 ***
## hp          -0.03177    0.00903  -3.519  0.00145 **
## wt          -3.87783    0.63273  -6.129 1.12e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.593 on 29 degrees of freedom
## Multiple R-squared:  0.8268, Adjusted R-squared:  0.8148
## F-statistic: 69.21 on 2 and 29 DF,  p-value: 9.109e-12
```

Sur le modèle 2 : mpg_chapeau = 37.23 -0.03*hp-3.88*hp

8. Réaliser une ACP sur les variables de mtcars sauf sur Y. Avec les variables données par l'ACP càd les composantes principales, réaliser un autre modèle

Conditions préalables pour réaliser l'ACP

1. variables quantitatives continues

2. centrer les données impérativement mais comme les mesures de données de mtcars sont hétérogènes, nous allons réaliser une ACP normée Pour la réalisation de l'ACP, nous allons directement utiliser le package FactoShiny sur le jeu de données mtcars2 pour pouvoir mettre les variables qualitatives en supplémentaires et mpg en variable quantitative supplémentaire

```
library(ade4)

## Warning: package 'ade4' was built under R version 3.6.3

library(FactoMineR)

## Warning: package 'FactoMineR' was built under R version 3.6.3

##
## Attaching package: 'FactoMineR'

## The following object is masked from 'package:ade4':
##
##     reconst

res.PCA<-PCA(mtcars2,quali.sup=c(2,8,9,10,11),quanti.sup=c(1),graph=FALSE)
#summary(res.PCA)
#library(Factoshiny)
#resu.shiny <- PCAshiny(mtcars)
```

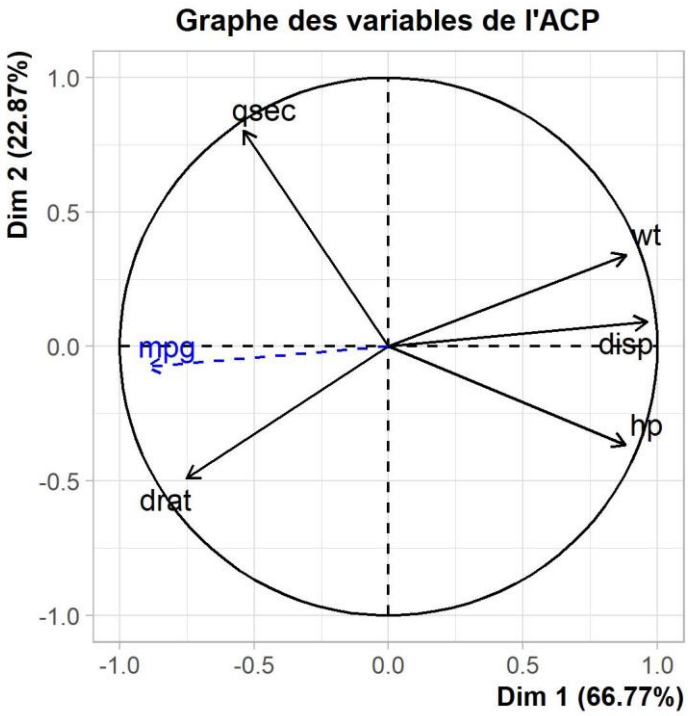


Figure : Cercle de corrélation des variables actives et supplémentaires

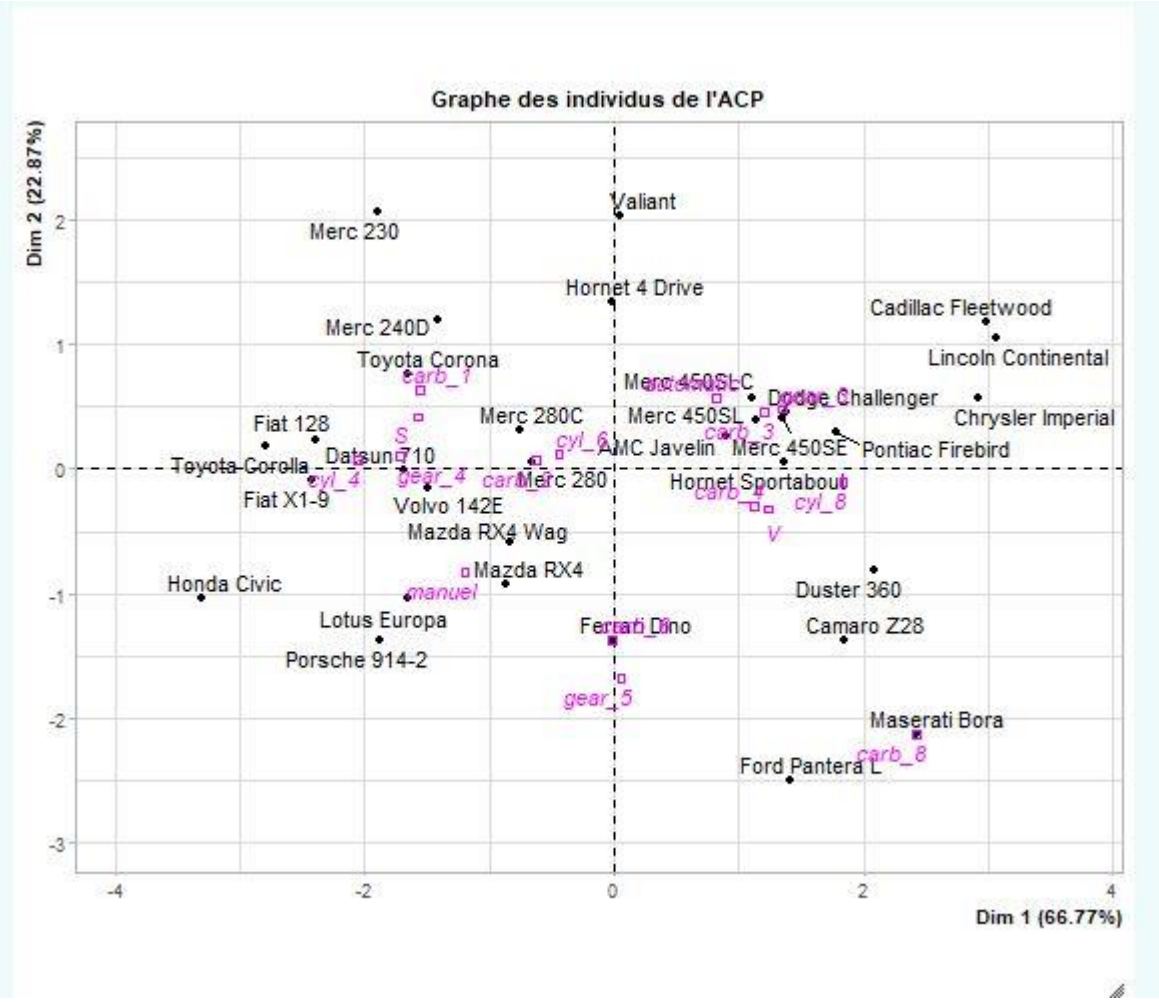


Figure : Graphe des individus avec les variables qualitatives comme supplémentaires

Les deux premières composantes conservent à elles-seules 89.64% de l'information totale.

Rappelons qu'une composante principale est construite comme une combinaison linéaire des variables initiales et que les composantes principales sont deux à deux orthogonaux(propriété des matrices symétriques).Ils sont obtenus sur R en tapant la ligne de command suivante : res.PCA$*var*$coord. Nous allons conserver que 3 chiffres après la virgule comme sur le fichier mtcars en utilisant la fonction round

```
# Définissons d'abord la fonction norme
norm_vec <- function(x) sqrt(sum(x^2))
CP1 <- round(res.PCA$var$coord[,1]/norm_vec(res.PCA$var$coord[,1]),3)
CP2 <- round(res.PCA$var$coord[,2]/norm_vec(res.PCA$var$coord[,1]),3)
CP3 <- round(res.PCA$var$coord[,3]/norm_vec(res.PCA$var$coord[,3]),3)
CP4 <- round(res.PCA$var$coord[,4]/norm_vec(res.PCA$var$coord[,4]),3)
CP5 <- round(res.PCA$var$coord[,5]/norm_vec(res.PCA$var$coord[,5]),3)
new_mpg <- round (res.PCA$quanti.sup$coord,3)
```

```
###Corrélation linéaire entre new_mpg(la coordonnée de mpg dans la base des composantes principales) et les composantes principales
variables_acp <-data.frame(CP1,CP2,CP3,CP4,CP5,new_mpg)
```

```
```{r}
library(MVN)
variables_acp <-data.frame(CP1,CP2,CP3,new_mpg)
result_bis <- mvn(data = variables_acp,mvnTest ="mardia")
result_bis$multivariateNormality
```
```

Test <fctr>	Statistic <fctr>	p value <fctr>	Result <fctr>
Mardia Skewness	20	0.457929714471852	YES
Mardia Kurtosis	-1.29099444873581	0.196705602458947	YES
MVN	NA	NA	YES

3 rows

Commentaire : Aucun des deux tests de de multinormalité(Mardia Skewness et Kurtosis) de mardia n'est significatif au seuil alpha =5%. Ce qui est résumé dans la dernière ligne de MV N qui contient la valeur YES. Nous conservons l'hypothèse de normalité bivariée(H0).L'erre ur de cette décision est un risque de deuxième espèce Béta que nous considérons suffisame nt petit car la taille de l'échantillon dépasse 30.

Nous pouvons utiliser la fonction cor.mtest avec correction de Bonferroni pour tester la null ité du coefficicient de corrélation de Pearson

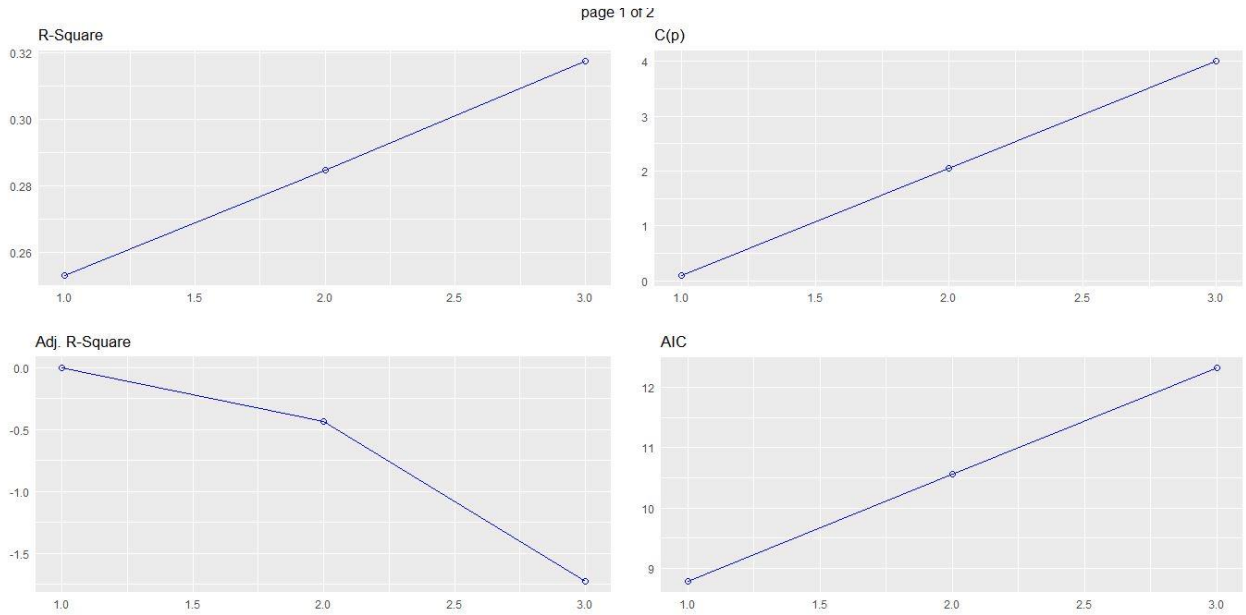
```
> cor.mtest(variables_acp)$p <0.05/6
      [,1]  [,2]  [,3]  [,4]
[1,]  TRUE FALSE FALSE FALSE
[2,] FALSE  TRUE FALSE FALSE
[3,] FALSE FALSE  TRUE FALSE
[4,] FALSE FALSE FALSE  TRUE
```

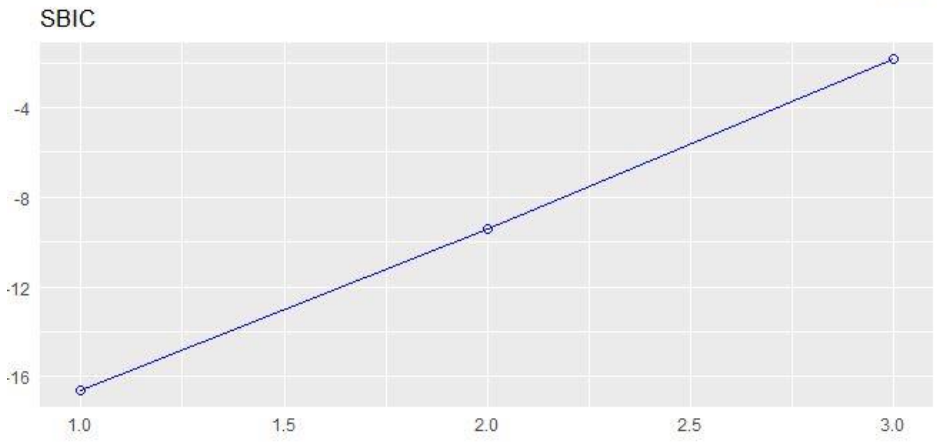
Même si le test mentionne qu'aucun des varaibles n'est significativement corrélée à new_m pg(la coordonnée de mpg dans les nouvelles composantes), ce qui est peut être dû au fait q ue mpg étant une variable supplémentaire, elle ne participe pas à la construction des axes. Nous allons étudier le modèle linéaire

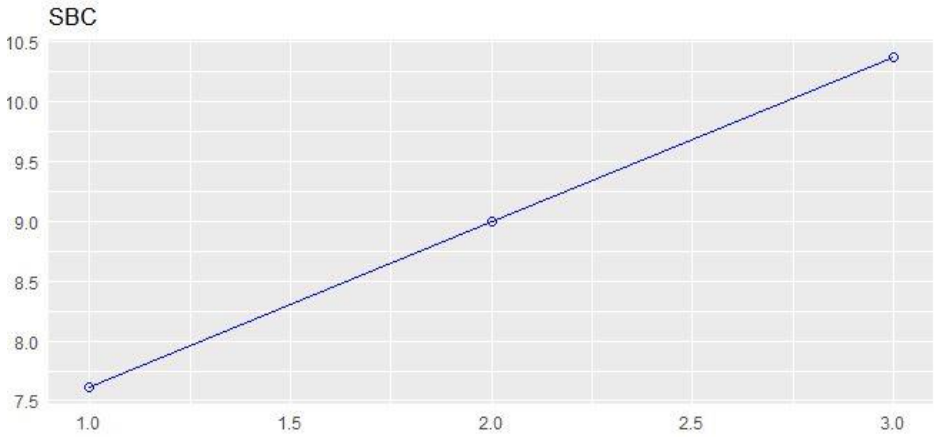
```
#- Modèle linéaire avec les composantes principales
model_acp <- lm(new_mpg~CP1+CP2+CP3, data=mtcars)
shapiro.test((residuals(model_acp)))

##
##  Shapiro-Wilk normality test
##
## data:  (residuals(model_acp))
## W = 0.95069, p-value = 0.7421
```

```
#- Meilleur modèle avec les composantes principales
(var1 <- ols_step_best_subset(model_acp))
plot(var1)
```







Commentaire : Le modèle 1 reste le meilleur modèle explicatif(ayant le bic le plus faible) , prédictif(ayant l'AIC le plus faible) et a le R-square ajusté le plus élevé

Etudions ce modèle

```
Model1<- lm(new_mpg~ CP1)
shapiro.test(residuals(model2))

##
##  Shapiro-Wilk normality test
##
## data : rsiduals(model1)
## W = 0.88497, p-value = 0.3324
```

Conclusion : La p-valeur étant supérieure à alpha =0.05, le test n'est pas signifiactif donc nous décidons que les erreurs suivent une loi normale. Le risque d'erreur associé est un

risque de deuxième espèce que nous pourrons pas évaluer mais comme notre échanntillon contient 32 variables(supérieur à 30) , nous la considérons suffisament faible.

En faisant le summary du modèle 1 on obtient le résultat suivant :

```
Call:
lm(formula = new_mpg ~ CP1)

Residuals:
  Dim.1   Dim.2   Dim.3   Dim.4   Dim.5
-0.5538  0.2509 -0.1525  0.3110  0.1444

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.1121     0.1970  -0.569    0.609
CP1          -0.4443     0.4406  -1.008    0.388

Residual standard error: 0.4125 on 3 degrees of freedom
Multiple R-squared:  0.2531,   Adjusted R-squared:  0.004153
F-statistic: 1.017 on 1 and 3 DF,  p-value: 0.3876
```

Donc new_mpg_chapeau = -0.11-0.44*CP1

9.Appliquer la procédure stagewise à mtcars (rappelez vous de la procédure : initialisation : Y avec la variable la plus corrélée des variables explicatives et ensuite ce sont les résidus des modèles que vous construisez au fur et mesure qui jouent le rôle de la variable réponse Y)

Nous allons appliquer la méthode de stagewise comme présentée dans le cours 3 :

Première étape : effectuer la régression avec la variable la plus corrélée avec Y La variable wt est celle la plus corrélée

```
droite_correlation <- lm(mpg~wt,data=mtcars)
summary(droite_correlation)

##
## Call:
## lm(formula = mpg ~ wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5432 -2.3647 -0.1252  1.4096  6.8727
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
## wt           -5.3445     0.5591  -9.559 1.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.046 on 30 degrees of freedom
## Multiple R-squared:  0.7528, Adjusted R-squared:  0.7446
## F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

2ème étape : Calcul des résidus obtenues avec cette régression

```
residus <- residuals(droite_correlation)
residus
```

```
##           Mazda RX4       Mazda RX4 Wag          Datsun 710      Hornet 4
Drive
##          -2.2826106          -0.9197704          -2.0859521           1.29
73499
##   Hornet Sportabout             Valiant          Duster 360           Merc
240D
##          -0.2001440          -0.6932545          -3.9053627           4.16
37381
##            Merc 230            Merc 280           Merc 280C          Merc
450SE
##           2.3499593           0.2998560          -1.1001440           0.86
68731
##          Merc 450SL         Merc 450SLC  Cadillac Fleetwood Lincoln Contin
ental
##          -0.0502472          -1.8830236           1.1733496           2.10
32876
##   Chrysler Imperial            Fiat 128         Honda Civic      Toyota Co
rolla
##           5.9810744           6.8727113           1.7461954           6.42
19792
##       Toyota Corona    Dodge Challenger         AMC Javelin          Camar
o Z28
##          -2.6110037          -2.9725862          -3.7268663          -3.46
23553
##    Pontiac Firebird           Fiat X1-9       Porsche 914-2        Lotus E
uropa
##           2.4643670           0.3564263           0.1520430           1.20
10593
##      Ford Pantera L        Ferrari Dino       Maserati Bora          Volvo
142E
##          -4.5431513          -2.7809399          -3.2053627          -1.02
74952
```

3ème étape : Considérer ensuite ces résidus comme une nouvelle variable dépendante que l'on veut expliquer à l'aide des variables explicatives restantes

```r
mtcars_res <- cbind(mtcars,residus)

#test de multinormalité
result1 <- mvn(data = mtcars_res[,c(3,4,5,12)],mvnTest="mardia")
result1
```

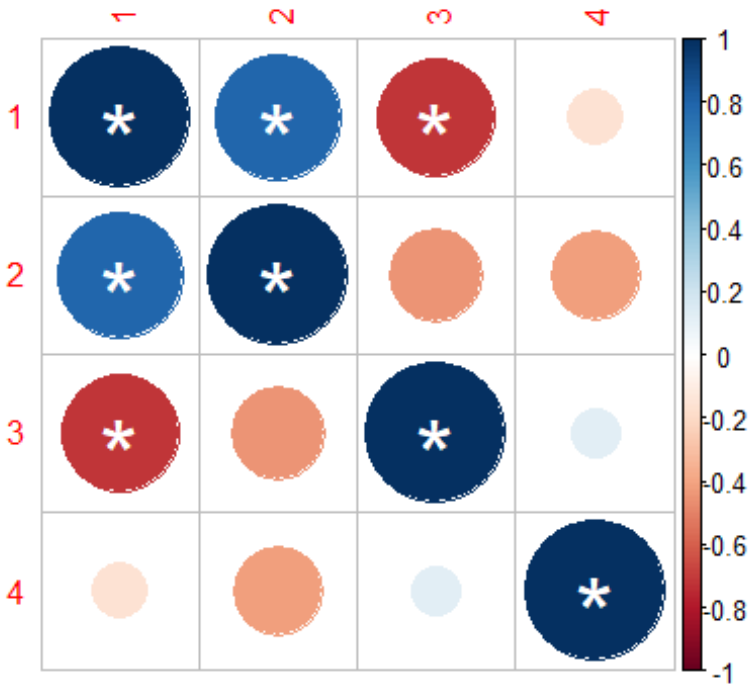
```
## $multivariateNormality
##              Test        Statistic             p value Result
## 1 Mardia Skewness 42.8138272711027 0.00216331710399521     NO
## 2 Mardia Kurtosis 1.32611376555257   0.184802017172272    YES
## 3             MVN             <NA>                 <NA>     NO
##
## $univariateNormality
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk   disp       0.9200    0.0208    NO
## 2 Shapiro-Wilk    hp        0.9334    0.0488    NO
## 3 Shapiro-Wilk   drat       0.9459    0.1101    YES
## 4 Shapiro-Wilk  residus     0.9451    0.1044    YES
##
## $Descriptives
##          n          Mean     Std.Dev      Median       Min        Max
## disp    32  2.307219e+02 123.9386938 196.3000000 71.100000 472.000000
## hp      32  1.466875e+02  68.5628685 123.0000000 52.000000 335.000000
## drat    32  3.596563e+00   0.5346787   3.6950000  2.760000   4.930000
## residus 32 -5.114724e-17   2.9963523  -0.1251956 -4.543151   6.872711
##               25th       75th      Skew   Kurtosis
## disp    120.825000 326.000000 0.3816570 -1.2072119
## hp       96.500000 180.000000 0.7260237 -0.1355511
## drat      3.080000   3.920000 0.2659039 -0.7147006
## residus  -2.364709   1.409561 0.6367705 -0.3004514
```

Nous allons utiliser un test de permutation avec une correction de Bonferroni puisque l'hypothèse de binormalité est rejettée par l'un des tests

```r
permresidus1 <- perm.cor.mtest(mtcars_res[,c(3,4,5,12)])
permresidus1$p < 0.05/choose(ncol(mtcars_res[,c(3,4,5,12)]),2)
```

```
##       [,1]  [,2]  [,3]  [,4]
## [1,]  TRUE  TRUE  TRUE FALSE
## [2,]  TRUE  TRUE FALSE FALSE
## [3,]  TRUE FALSE  TRUE FALSE
## [4,] FALSE FALSE FALSE  TRUE
```

```r
corrplot(permresidus1$cor,p.mat=permresidus1$p,pch.col="white",insig="label_sig",
         sig.level=0.05/choose(ncol(mtcars_res[,c(3,4,5,12)]),2))
```



Conclusion :Aucune des variables n'est significativement corrélée avec résidus 1, donc le procédure de stagewise s'arrête là.

10.Que reprochez-vous à ce jeu de données ? Y a t il des voitures atypiques ?

Le jeu de données contient trop de valeurs atypiques .Les voitures atypiques sont obtenues avec la commande result*multivariate*outliers. Il y a 7 voitures atypiques mais Maserati et Ford restent les plus atypiques car leur distance est trop grande par rapport aux autres

```
result$multivariateOutliers

result <- mvn(data=mtcars_quanti,mvnTest = "mardia",univariateTest = "SW",
              multivariatePlot = "qq",multivariateOutlierMethod = "adj",
    showOutliers = T,showNewData = T)


result$multivariateOutliers
```

	Observation <fctr>	Mahalanobis Distance <dbl>	Outlier <chr>
Maserati Bora	Maserati Bora	107.537	TRUE
Ford Pantera L	Ford Pantera L	85.115	TRUE
Camaro Z28	Camaro Z28	35.230	TRUE
Duster 360	Duster 360	30.679	TRUE
Merc 240D	Merc 240D	20.804	TRUE
Merc 230	Merc 230	16.813	TRUE
Fiat 128	Fiat 128	16.123	TRUE

7 rows