

CENTRE ANTOINE LACASSAGNE

DÉPARTEMENT ÉPIDÉMIOLOGIE, BIOSTATISTIQUES ET DONNÉES DE SANTÉ (DEBDS)

Responsable: Dr Emmanuel CHAMOREY

Yann CHATEAU - Data Manager ~ Renaud SCHIAPPA - Data Manager

David RENER - Data Manager ~ Dr Jocelyn GAL - Biostatisticien

Brice THAMPHYA - Bioinformaticien

Version du: 06/01/2021

RAPPORT STATISTIQUE FINAL

GDA – Célya FRANC, Juliette MOINEAU, Tom MAUGIER

Le biostatisticien et le data manager devront être co-auteur et avoir validé toute publication utilisant les résultats de ce rapport statistique



Table of Contents

I) Import et Pré-traitement avant analyse de la base de données	3
II) Analyse descriptive.....	3
1) Analyse des variables de la cohorte totale	3
A) Analyse des NA.....	3
B) Analyse des variables qualitatives.....	5
C) Analyse des variables quantitatives	8
2) Création de sous-groupes en fonction du centre	8
A) Analyse descriptive de la cohorte d'apprentissage.....	9
(a) Analyse de toutes les variables 2 à 2	9
(b) Analyse descriptive des variables qualitatives.....	10
(c) Analyse descriptive des variables quantitatives.....	13
B) Analyse descriptive de la cohorte de validation externe (Toulouse)	13
(a) Analyse descriptive des variables qualitatives.....	13
(b) Analyse descriptive des variables quantitatives	15
C) Analyse descriptive de la cohorte de validation interne (30% Nice).....	16
(a) Analyse descriptive des variables qualitatives.....	16
(b) Analyse descriptive des variables quantitatives	18
III) Analyse univariée (uniquement sur la cohorte d'apprentissage)	19
1) Suppression des modalités non observées.....	19
2) Suppression des variables avec une modalité observée inférieure ou égale à 10 %	19
3) Analyse statistique de toutes les variables en fonction de la variable cancer	19
4) Récupération des indices des colonnes correspondant aux variables significatives.....	22
5) Tableau récapitulatif des valeurs significatives	22
IV) Analyse multivariée (uniquement sur la cohorte d'apprentissage)	24
1) Analyse de la colinéarité entre les variables.....	24
2) Modèle initial	25
A) Minimisation de l'AIC	25
B) Modèle final	28
C) Analyse des résidus du modèle final	29
D) Evaluation du modèle	33
3) Validation du modèle.....	35
A) Accuracy sur la cohorte d'apprentissage	35



B) Accuracy sur la cohorte de validation interne Nice	35
C) Accuracy sur la cohorte de validation externe Toulouse	36
D) Courbe de ROC.....	36
V) Bonus : Test XGB avec cross validation.....	37

VALIDATION DU RAPPORT D'ANALYSE FINALE

I) Import et Pré-traitement avant analyse de la base de données

II) Analyse descriptive

```
dim(df)
```

```
## [1] 1335 17
```

```
set.seed(12345) #Fixation de La graine
```

Pour commencer, la dimension de la base de données est de 17 colonnes et 1335 lignes. Nous avons donc 17 variables pour 1 335 patients.

1) Analyse des variables de la cohorte totale




A) Analyse des NA

A présent, nous allons mettre en évidence les variables utilisables pour la suite de l'étude. C'est-à-dire celles avec un pourcentage de valeurs manquantes inférieur à 10 %.

```
md_table(as.data.frame(skim(df)))
```

skim_type	skim_variable	n_missing	complete_rate	factor_ordered	factor_nunique	factor_top_counts	numeric_mean	numeric_sd	numeric_p0	numeric_p25	numeric_p50	numeric_p75	numeric_p100	numeric_hist
factor	Y	0	1.00000000	FALSE	2	0: 862, 1: 473								
factor	X2	0	1.00000000	FALSE	6	1: 487, 4: 264, 6: 236, 5: 131								
factor	X3	0	1.00000000	FALSE	2	2: 1044, 1: 291								
factor	X4	0	1.00000000	FALSE	2	<55: 706, =55: 629								
factor	X5	108	0.91910112	FALSE	2	0: 787, 1: 440								
factor	X6	121	0.90936330	FALSE	2	0: 1195, 1: 19								
factor	X7	124	0.90711610	FALSE	2	No: 1100, Yes: 111								



skim_type	skim_variable	n_missing	complete_rate	factor_order	factor.n_unique	factor.top_counts	numeric.mean	numeric.sd	numeric.p0	numeric.p25	numeric.p50	numeric.p75	numeric.p100	numeric.hist
factor	X8	0	1.00000000	FALSE	2	No: 1301, Yes: 34								
factor	X9	55	0.95880150	FALSE	2	Yes: 983, No: 297								
factor	X10	79	0.94082397	FALSE	2	No: 1227, Yes: 29								
factor	X12.Scoring	2	0.99850187	FALSE	4	4A: 490, 4B: 392, 2-3: 302, 5: 149								
factor	X15	0	1.00000000	FALSE	3	4: 693, 5: 389, 3: 253								
factor	X16	34	0.97453184	FALSE	2	0: 1291, 1: 10								
factor	X17_group	131	0.90187266	FALSE	2	0: 867, 1: 337								
numeric	X11	493	0.63071161				1.800632	3.406129	0.003	0.8185	1.4	2.147	76.88	
numeric	X13	8	0.99400749				20.774981	11.278509	2.000	12.5000	18.0	27.000	77.00	
numeric	X14	1,279	0.04194757				10.908393	9.635908	0.000	4.7500	7.3	13.925	51.72	

Suite à cette analyse, nous avons mis en évidence que nous ne pouvons pas utiliser les variables suivantes lors des analyses univariée et multivariée:

- X14: Cette variable présente 95.81% de NA.
- X11: Cette variable présente 36.93% de NA.

Nous utiliserons donc toutes les autres variables pour ces deux analyses. Sachant que les patients éligibles ont déjà été sélectionné dans le pré-traitement.

```
missmap(df, col=c("blue", "red"), legend=T)
```





Ces résultats sont confirmés visuellement par la missmap. Celle-ci met également en évidence la potentielle corrélation entre les variables X6 et X7. En effet, lorsqu'un NA est observé pour X6 il est souvent observé pour X7, et inversement.

B) Analyse des variables qualitatives

```
index=c(1:ncol(df))
ft=sortiequali(as.data.frame(df),index)
ft=md_table(ft)
ft
```



Variable	Modalite	Nombre	Frequence	Vide
Y				0 (0%)
	0	862	64.57%	
	1	473	35.43%	
X2				0 (0%)
	1	487	36.48%	
	2	129	9.66%	
	3	88	6.59%	
	4	264	19.78%	
	5	131	9.81%	
	6	236	17.68%	
X3				0 (0%)
	1	291	21.8%	
	2	1044	78.2%	
X4				0 (0%)
	<55	706	52.88%	
	=55	629	47.12%	
X5				108 (8.09%)
	0	787	64.14%	
	1	440	35.86%	
X6				121 (9.06%)
	0	1195	98.43%	
	1	19	1.57%	
X7				124 (9.29%)
	No	1100	90.83%	
	Yes	111	9.17%	



Variable	Modalite	Nombre	Frequence	Vide
X8				0 (0%)
	No	1301	97.45%	
	Yes	34	2.55%	
X9				55 (4.12%)
	No	297	23.2%	
	Yes	983	76.8%	
X10				79 (5.92%)
	No	1227	97.69%	
	Yes	29	2.31%	
X12.Score				2 (0.15%)
	2-3	302	22.66%	
	4A	490	36.76%	
	4B	392	29.41%	
	5	149	11.18%	
X15				0 (0%)
	3	253	18.95%	
	4	693	51.91%	
	5	389	29.14%	
X16				34 (2.55%)
	0	1291	99.23%	
	1	10	0.77%	
X17_regroup				131 (9.81%)
	0	867	72.01%	
	1	337	27.99%	
rm(ft,index)				



Dans un premier temps, on se focalise sur les vides. Pour les variables X5, X6, X7, X17_regroup, les pourcentages de vides oscillent entre 8 et 10%. Ces variables seront donc sans doute supprimées du modèle final.

Dans un second temps, on regarde la fréquence des modalités de chaque variable. On observe que pour X6 et X16 la modalité 0 semble corrélée avec un cancer bénin.

Pour les variables X7, X8 et X10 c'est la modalité "no" qui semble corrélée avec un cancer bénin. On peut émettre ces conclusions car la fréquence de ces modalités est supérieure à 90%.

C) Analyse des variables quantitatives

```
index=c(11,13,14)
ft=sortiequanti(as.data.frame(df),index)
ft=md_table(ft)
ft
```

Variable	Vide	Moyenne	Quartiles	EcartType
X11	493	1.8	[0-0.82-1.4-2.15-76.88]	3.41
X13	8	20.77	[2-12.5-18-27-77]	11.28
X14	1279	10.91	[0-4.75-7.3-13.93-51.72]	9.64

```
rm(ft,index)
```

Pour les variables X14 et X11, on observe un nombre de vide très important (surtout X14), ce qui suggère comme précédemment que ces variables seront supprimées du modèle final.

Pour X11 on observe que la moyenne est proche de la médiane, ce qui est en corrélation avec le faible écart-type observé, traduisant ainsi une faible variabilité. En revanche, pour X13 et X15, la moyenne s'éloigne un peu plus de la médiane, ce qui est en corrélation avec des écart-types beaucoup plus importants.

2) Création de sous-groupes en fonction du centre

L'objectif de cette partie est de créer les cohortes de validation et d'apprentissage.

Il y a deux cohortes de validation:

- Validation externe : Toulouse
- Validation interne : 30% des patients de Nice

La cohorte d'apprentissage contient les 70% restants des patients niçois.




```

valid<-rbind(df[(df$X2==4),],df[(df$X2==5),]) #Toulouse
Nice<-rbind(df[(df$X2==1),],df[(df$X2==2),],df[(df$X2==3),],df[(df$X2==6),])#
Nice
appren_int<-sample(nrow(Nice),nrow(Nice)*0.7)#r cup ration al atoire des indi
ces des lignes de l' chantillon
appren<-Nice[appren_int,]
valid_2<-Nice[-appren_int,]
dim(appren) #cohorte d'apprentissage 70% Nice.

## [1] 658 17

dim(valid) #cohorte validation externe (Toulouse)

## [1] 395 17

dim(valid_2) #cohorte de validation interne (30% Nice)

## [1] 282 17

```

On obtient alors une cohorte d'apprentissage avec 658 patients, ainsi, qu'une cohorte de validation externe de 395 patients puis, une cohorte de validation interne de 282 patients.

A) Analyse descriptive de la cohorte d'apprentissage

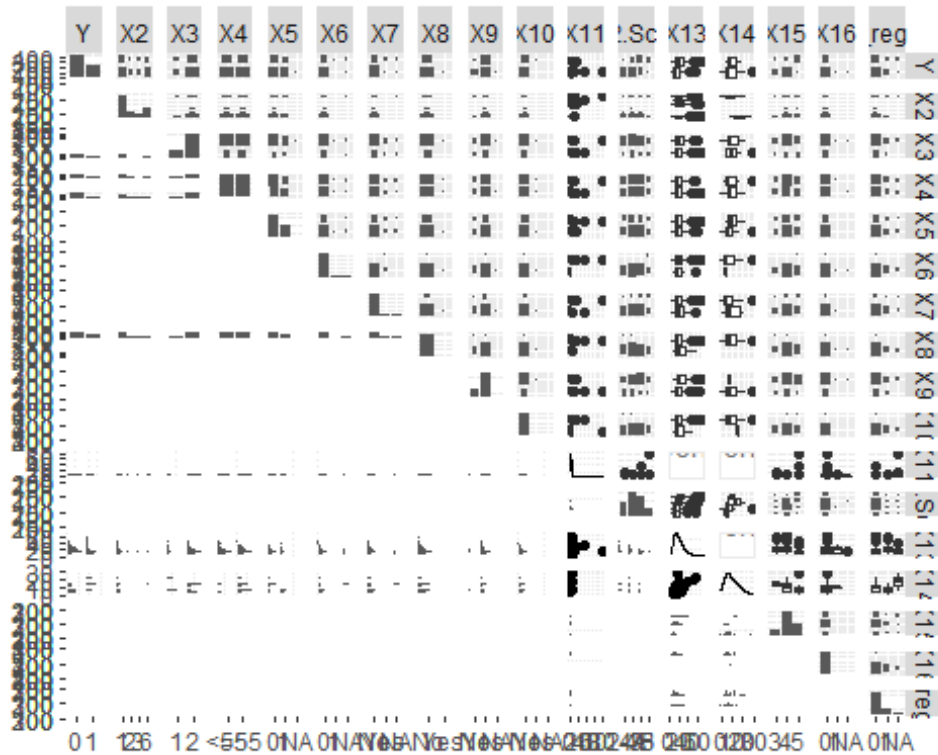
(a) Analyse de toutes les variables 2   2

```

ggp<-ggpairs(as.data.frame(appren))
ggp

```





L'objectif de cette fonction est de nous donner un aspect plus visuel d'une variable par rapport à une autre. On observe pour la variable X8 une majorité de non, à l'inverse des variables X6 et X7 qui ont une majorité de yes.

(b) Analyse descriptive des variables qualitatives

```
index=c(1:ncol(appren))
ft=sortiequali(as.data.frame(appren),index)
ft=md_table(ft)
ft
```

Variable	Modalite	Nombre	Frequence	Vide
Y				0 (0%)
	0	427	64.89%	
	1	231	35.11%	
X2				0 (0%)
	1	350	53.19%	
	2	82	12.46%	
	3	63	9.57%	



Variable	Modalite	Nombre	Frequence	Vide
X3	4	0	0%	
	5	0	0%	
	6	163	24.77%	
X3				0 (0%)
X4	1	151	22.95%	
	2	507	77.05%	
				0 (0%)
X5	<55	323	49.09%	
	=55	335	50.91%	
				5 (0.76%)
X6	0	425	65.08%	
	1	228	34.92%	
				55 (8.36%)
X7	0	592	98.18%	
	1	11	1.82%	
				55 (8.36%)
X8	No	529	87.73%	
	Yes	74	12.27%	
				0 (0%)
X9	No	643	97.72%	
	Yes	15	2.28%	
				5 (0.76%)
X10	No	157	24.04%	
	Yes	496	75.96%	
				2 (0.3%)



Variable	Modalite	Nombre	Frequence	Vide
X12.Score	No	643	98.02%	0 (0%)
	Yes	13	1.98%	
	2-3	131	19.91%	
	4A	237	36.02%	
	4B	211	32.07%	
X15	5	79	12.01%	0 (0%)
	3	101	15.35%	
	4	367	55.78%	
X16	5	190	28.88%	4 (0.61%)
	0	644	98.47%	
X17_regroup	1	10	1.53%	47 (7.14%)
	0	473	77.41%	
	1	138	22.59%	

rm(ft,index)

Concernant l'analyse des vides. Pour les variables X5, X6, X17_regroup, les pourcentages de vides sont compris entre 7 et 9%. Ces variables seront donc sans doute supprimées du modèle final, comme remarqué précédemment sur la cohorte totale.

Dans un second temps, on regarde la fréquence des modalités de chaque variable. On observe que pour X6 et X16 la modalité 0 semble corrélée avec un cancer bénin.

Pour les modalités X8 et X10 c'est la modalité "no" qui semble corrélée avec un cancer bénin. On peut émettre ces conclusions car la fréquence de ces modalités est supérieure à 90%.



(c) Analyse descriptive des variables quantitatives

```
index=c(11,13,14)
ft=sortiequanti(as.data.frame(appren),index)
ft=md_table(ft)
ft
```

Variable	Vide	Moyenne	Quartiles	EcartType
X11	235	1.96	[0-0.9-1.49-2.2-76.88]	4.17
X13	4	20.46	[2-12-18-26.5-72]	11.35
X14	628	10.12	[0-5.43-8.1-13.83-28.5]	6.52

```
rm(ft,index)
```

Pour les variables X14 et X11, on observe un nombre de vides très important (surtout X14), ce qui suggère comme précédemment que ces variables seront supprimées du modèle final.

Pour X11, X13, X14 on observe que la moyenne est proche de la médiane, ce qui est en corrélation avec le faible écart-type observé pour X11 notamment. Concernant X14, l'écart-type est légèrement plus important. Pour X13 en revanche on observe un écart-type beaucoup plus important, donc on ne peut pas le mettre en corrélation avec le fait que la moyenne soit proche de la médiane.

B) Analyse descriptive de la cohorte de validation externe (Toulouse)

(a) Analyse descriptive des variables qualitatives

```
index=c(1:ncol(valid))
ft=sortiequali(as.data.frame(valid),index)
ft =md_table(ft)
ft
```

Variable	Modalite	Nombre	Frequence	Vide
Y				0 (0%)
	0	255	64.56%	
	1	140	35.44%	
X2				0 (0%)
	1	0	0%	
	2	0	0%	
	3	0	0%	
	4	264	66.84%	
	5	131	33.16%	



Variable	Modalite	Nombre	Frequence	Vide
X3	6	0	0%	0 (0%)
	1	77	19.49%	
X4	2	318	80.51%	0 (0%)
	<55	226	57.22%	
	=55	169	42.78%	
X5	0	169	56.9%	98 (24.81%)
	1	128	43.1%	
X6	0	354	98.88%	37 (9.37%)
	1	4	1.12%	
X7	No	348	98.03%	40 (10.13%)
	Yes	7	1.97%	
X8	No	384	97.22%	0 (0%)
	Yes	11	2.78%	
X9	No	67	19.36%	49 (12.41%)
	Yes	279	80.64%	
X10	No	312	97.81%	76 (19.24%)
	Yes	7	2.19%	



Variable	Modalite	Nombre	Frequence	Vide
X12.Score				2 (0.51%)
	2-3	111	28.24%	
	4A	143	36.39%	
	4B	96	24.43%	
	5	43	10.94%	
X15				0 (0%)
	3	103	26.08%	
	4	172	43.54%	
	5	120	30.38%	
X16				30 (7.59%)
	0	365	100%	
	1	0	0%	
X17_regroup				58 (14.68%)
	0	208	61.72%	
	1	129	38.28%	

rm(ft,index)

Dans un premier temps, on se focalise sur les vides. Pour la cohorte de validation externe on observe un pourcentage de vide égal à 0 (ou proche) pour les variables X2, X3, X4, X8, X12.Score et X15. Les autres présentent un pourcentage de vides compris entre 7% et 25% (X5 notamment).

Dans un second temps, on regarde la fréquence des modalités de chaque variable. On observe que pour X6 la modalité 0 semble corrélée avec un cancer bénin et pour X16, il n'y a aucun doute que la modalité 0 soit corrélée avec un cancer bénin puisque celle-ci est représentée à 100%.

Pour les variables X7, X8 et X10 c'est la modalité "no" qui semble corrélée avec un cancer bénin. On peut émettre ces conclusions car la fréquence de ces modalités est supérieure à 90%.

(b) Analyse descriptive des variables quantitatives

```
index=c(11,13,14)
ft=sortiequanti(as.data.frame(valid),index)
ft=md_table(ft)
ft
```



Variable	Vide	Moyenne	Quartiles	EcartType
X11	158	1.5	[0-0.77-1.3-1.93-7.77]	1
X13	3	21.13	[6-13-18.6-26-77]	11.02
X14	384	14.07	[3-4.69-7-11.3-51.72]	16.36

```
rm(ft,index)
```

Ici, on observe pour les deux variables X11 et X13 que la moyenne est proche de la médiane. Toutefois, ceci est corrélé avec l'écart-type de X11 (très faible) mais pas pour X13 dont l'écart-type est trop important.

Cependant, pour la variable X14 la valeur de la moyenne est le double de la médiane, ce qui est en corrélation avec un écart-type très élevé. Cela nous indique donc que cette variable ne sera sans doute pas dans notre modèle final.

C) Analyse descriptive de la cohorte de validation interne (30% Nice)

(a) Analyse descriptive des variables qualitatives

```
index=c(1:ncol(valid_2))
ft=sortiequali(as.data.frame(valid_2),index)
ft=md_table(ft)
ft
```

Variable	Modalité	Nombre	Frequence	Vide
Y				0 (0%)
	0	180	63.83%	
	1	102	36.17%	
X2				0 (0%)
	1	137	48.58%	
	2	47	16.67%	
	3	25	8.87%	
	4	0	0%	
	5	0	0%	
	6	73	25.89%	
X3				0 (0%)
	1	63	22.34%	



Variable	Modalite	Nombre	Frequence	Vide
X4	2	219	77.66%	0 (0%)
	<55	157	55.67%	
X5	=55	125	44.33%	5 (1.77%)
	0	193	69.68%	
X6	1	84	30.32%	29 (10.28%)
	0	249	98.42%	
X7	1	4	1.58%	29 (10.28%)
	No	223	88.14%	
X8	Yes	30	11.86%	0 (0%)
	No	274	97.16%	
X9	Yes	8	2.84%	1 (0.35%)
	No	73	25.98%	
X10	Yes	208	74.02%	1 (0.35%)
	No	272	96.8%	
X12.Score	Yes	9	3.2%	0 (0%)
	2-3	60	21.28%	
	4A	110	39.01%	



Variable	Modalite	Nombre	Frequence	Vide
X15	4B	85	30.14%	0 (0%)
	5	27	9.57%	
	3	49	17.38%	
	4	154	54.61%	
	5	79	28.01%	
X16				0 (0%)
	0	282	100%	
	1	0	0%	
X17_regroup				26 (9.22%)
	0	186	72.66%	
	1	70	27.34%	

rm(ft, index)

A propos des vides, pour les variables X6, X7 et X17_regroup les pourcentages de vides sont compris entre 9 et 11%. Ces variables seront donc sans doute supprimées du modèle final.

Dans un second temps, on regarde la fréquence des modalités de chaque variable. On observe que pour X6 la modalité 0 semble corrélée avec un cancer bénin et pour X16, il n'y a aucun doute que la modalité 0 soit corrélée avec un cancer bénin puisque celle-ci est représentée à 100%.

Pour les modalités X8 et X10 c'est la modalité "no" qui semble corrélée avec un cancer bénin. On peut émettre ces conclusions car la fréquence de ces modalités est supérieure à 90%.

(b) Analyse descriptive des variables quantitatives

```
index=c(11,13,14)
ft=sortiequanti(as.data.frame(valid_2),index)
ft=md_table(ft)
ft
```

Variable	Vide	Moyenne	Quartiles	EcartType
X11	100	1.82	[0-0.8-1.35-2.18-45.66]	3.45
X13	1	21.03	[4-12.5-19-27.5-66]	11.49
X14	267	10.17	[0-3.8-6.8-15.3-27]	8.71



```
rm(ft,index)
```

Ici, on observe pour les deux variables X11 et X13 que la moyenne est proche de la médiane. Ceci est corrélé avec l'écart-type de X11 qui est faible mais, pas pour X13 dont l'écart-type est plus important.

Cependant, pour la variable X14 la moyenne est plus éloignée de la médiane, comparé à X11 et X13, ce qui est en corrélation avec un écart-type important.

III) Analyse univariée (uniquement sur la cohorte d'apprentissage)

1) Suppression des modalités non observées

Certaines variables présentent des modalités qui ne sont pas représentées dans la cohorte d'apprentissage. C'est pourquoi, dans un premier temps nous avons décidé de les supprimer.

Dans un deuxième temps, nous avons supprimé les variables qui ont au moins une modalité toujours associée à la même modalité de la variable cancer.

En l'absence de ces suppressions, il est impossible de réaliser les tests statiques avec tableStack.

```
appren<-drop_level_0_and_unassociate(appren,1)
```

2) Suppression des variables avec une modalité observée inférieure ou égale à 10 %

L'analyse univariée requiert la suppression des variables présentant des modalités avec une fréquence d'apparition inférieure ou égale à 10%.

```
cols<-numcol(c("X6", "X8", "X10"), appren)
appren<-appren[, -cols]
dim(appren)

## [1] 658 14
```

Suite à ces suppressions, il nous reste donc 13 variables, plus la variable cancer pour 658 patients.

3) Analyse statistique de toutes les variables en fonction de la variable cancer

L'objectif ensuite est de déterminer quelles sont les variables significatives par rapport à la variable cancer. Pour cela on utilise la fonction tableStack, qui réalise les tests statistiques adaptés.

```
tt=tableStack(vars=3:ncol(appren),
              by=1,
              dataFrame=as.data.frame(appren),
              na.col=T)
bold_lines=which(tt[,1] %in% colnames(appren))#Met Les noms en gras
colnames(tt)[1:3]=c("CANCER", "Benin", "Malin") #On attribue Les noms des colonnes
md_table(tt, bold.names=bold_lines)
```



CANCER	Benin	Malin	Test stat.	P value	Vide
Total	427	231			
X3			Chisq. (1 df) = 6.86	0.009	0 (0%)
1	84 (19.7%)	67 (29%)			
2	343 (80.3%)	164 (71%)			
X4			Chisq. (1 df) = 4.59	0.032	0 (0%)
<55	196 (45.9%)	127 (55%)			
=55	231 (54.1%)	104 (45%)			
X5			Chisq. (1 df) = 1	0.318	5 (0.76%)
0	269 (63.6%)	156 (67.8%)			
1	154 (36.4%)	74 (32.2%)			
X7			Chisq. (1 df) = 58.27	< 0.001	55 (8.36%)
No	383 (95%)	146 (73%)			
Yes	20 (5%)	54 (27%)			
X9			Chisq. (1 df) = 34.86	< 0.001	5 (0.76%)
No	133 (31.4%)	24 (10.4%)			
Yes	290 (68.6%)	206 (89.6%)			
X11			Ranksum test	0.222	235 (35.71%)
median(IQR)	1.4 (0.9,2.2)	1.6 (1,2.2)			



CANCER	Benin	Malin	Test stat.	P value	Vide
X12.Score			Chisq. (3 df) = 222.81	< 0.001	0 (0%)
2-3	128 (30%)	3 (1.3%)			
4A	193 (45.2%)	44 (19%)			
4B	97 (22.7%)	114 (49.4%)			
5	9 (2.1%)	70 (30.3%)			
X13			Ranksum test	< 0.001	4 (0.61%)
median(IQR)	20 (14,28)	13.9 (9,21.7)			
X14			t-test (28 df) = 0.14	0.886	628 (95.44%)
mean(SD)	9.9 (5)	10.3 (7.7)			
X15			Chisq. (2 df) = 295.23	< 0.001	0 (0%)
3	84 (19.7%)	17 (7.4%)			
4	315 (73.8%)	52 (22.5%)			
5	28 (6.6%)	162 (70.1%)			
X16			Fisher's exact test	0.33	4 (0.61%)
0	421 (98.8%)	223 (97.8%)			
1	5 (1.2%)	5 (2.2%)			
X17_regroup			Chisq. (1 df) = 9.94	0.002	47 (7.14%)
0	300 (73.5%)	173 (85.2%)			
1	108 (26.5%)	30 (14.8%)			



4) Récupération des indices des colonnes correspondant aux variables significatives

```
signif=col_significatives(appren,tt,indice=T,nb_début_vars_TableStack = 3)  
signif
```

```
## [1] 3 4 6 7 9 10 12 14
```

5) Tableau récapitulatif des valeurs significatives

Création d'un nouveau data-frame contenant les variables significatives et la variable cancer.

```
appren_net<-appren[,c(1,signif)]  
  
tt=tableStack(vars=2:ncol(appren_net),  
              by=1,  
              dataFrame=as.data.frame(appren_net),  
              na.col=T)  
bold_lines=which(tt[,1] %in% colnames(appren_net))#Met les noms en gras  
colnames(tt)[1:3]=c("CANCER","Benin","Malin") #on attribue les noms des colon  
nes  
md_table(tt,bold.names=bold_lines)
```

CANCER	Benin	Malin	Test stat.	P value	Vide
Total	427	231			
X3			Chisq. (1 df) = 6.86	0.009	0 (0%)
1	84 (19.7%)	67 (29%)			
2	343 (80.3%)	164 (71%)			
X4			Chisq. (1 df) = 4.59	0.032	0 (0%)
<55	196 (45.9%)	127 (55%)			
=55	231 (54.1%)	104 (45%)			
X7			Chisq. (1 df) = 58.27	< 0.001	55 (8.36%)
No	383 (95%)	146 (73%)			
Yes	20 (5%)	54 (27%)			



CANCER	Benin	Malin	Test stat.	P value	Vide
X9			Chisq. (1 df) = 34.86	< 0.001	5 (0.76%)
No	133 (31.4%)	24 (10.4%)			
Yes	290 (68.6%)	206 (89.6%)			
X12.Score			Chisq. (3 df) = 222.81	< 0.001	0 (0%)
2-3	128 (30%)	3 (1.3%)			
4A	193 (45.2%)	44 (19%)			
4B	97 (22.7%)	114 (49.4%)			
5	9 (2.1%)	70 (30.3%)			
X13			Ranksum test	< 0.001	4 (0.61%)
median(IQR)	20 (14,28)	13.9 (9,21.7)			
X15			Chisq. (2 df) = 295.23	< 0.001	0 (0%)
3	84 (19.7%)	17 (7.4%)			
4	315 (73.8%)	52 (22.5%)			
5	28 (6.6%)	162 (70.1%)			
X17_regroup			Chisq. (1 df) = 9.94	0.002	47 (7.14%)
0	300 (73.5%)	173 (85.2%)			
1	108 (26.5%)	30 (14.8%)			

On obtient alors 8 variables significatives, ainsi que la variable Y.



IV) Analyse multivariée (uniquement sur la cohorte d'apprentissage)

1) Analyse de la colinéarité entre les variables

Dans un premier temps, on supprime les lignes comportant au moins un NA pour éviter les problèmes lors de l'analyse.

```
appren_net<-na.omit(appren_net)
# dim(appren_net)# = 553 12
```

Pour débiter l'analyse de la colinéarité on crée un Modèle Linéaire Généralisé (GLM) en tant que modèle initial. Ce modèle logistique permettra l'analyse des Facteurs d'Inflation de Variance (VIF).

```
formula = as.formula(paste("Y ~ ", paste(names(appren_net)[-1], collapse= "+") ))
model = glm(formula, data=appren_net,family = binomial())
```

Nous avons d'abord regardé si certaines variables étaient aliassées.

```
ld.vars <- attributes(alias(model)$Complete)$dimnames[[1]]
ld.vars
## NULL
```

Aucune d'entre elles ne le sont, ce qui signifie qu'il n'y a pas de corrélation parfaite avec la variable Y (cancer). Cela nous permet alors de poursuivre l'analyse avec les VIF.

```
corr=vif(model)
tabl=cbind("Variable"=rownames(corr),corr)
colnames(tabl)[2]="VIF"
md_table(tabl)
```

Variable	VIF	Df	GVIF^(1/(2*Df))
X3	1.02577517038091	1	1.01280559357703
X4	1.01398039213058	1	1.00696593394741
X7	1.07899850809183	1	1.03874852976639
X9	1.0945755076009	1	1.04621962684749
X12.Score	1.19523913531299	3	1.03017055897019
X13	1.19384566164276	1	1.09263244581275
X15	1.21746180982744	2	1.05042206697161
X17_regroup	1.02380740371307	1	1.01183368382016

Par la suite nous avons mis en évidence qu'aucune des variables ne semble trop corrélée avec Y puisqu'aucun des VIF n'est supérieur à 3.



2) Modèle initial

A) Minimisation de l'AIC

Après avoir vérifié la corrélation entre les variables et la variable cancer, on recrée un modèle logistique.

```
formula = as.formula(paste("Y ~ ", paste(names(appren_net)[1:-1], collapse= "+") ))
model_initial = glm(formula, data=appren_net, family = binomial())
summary(model_initial)

##
## Call:
## glm(formula = formula, family = binomial(), data = appren_net)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.2235  -0.5186  -0.3062   0.3013   3.1686
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.91176    0.89877  -5.465 4.63e-08 ***
## X32           -0.21135    0.30845  -0.685 0.493217
## X4=55         -0.26174    0.27269  -0.960 0.337135
## X7Yes          1.29354    0.42429   3.049 0.002298 **
## X9Yes          0.57117    0.37769   1.512 0.130464
## X12.Score4A    2.60279    0.68573   3.796 0.000147 ***
## X12.Score4B    3.61476    0.67812   5.331 9.79e-08 ***
## X12.Score5     4.45023    0.78586   5.663 1.49e-08 ***
## X13            0.01276    0.01255   1.017 0.309290
## X154          -0.34397    0.38214  -0.900 0.368062
## X155           2.88638    0.45451   6.351 2.15e-10 ***
## X17_regroup1 -0.10495    0.34086  -0.308 0.758154
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 685.63  on 552  degrees of freedom
## Residual deviance: 360.57  on 541  degrees of freedom
## AIC: 384.57
##
## Number of Fisher Scoring iterations: 6
```

On obtient alors un AIC initial de :

```
round(AIC(model_initial), digits = 1)

## [1] 384.6
```



On essaye alors de diminuer cet AIC par une méthode pas à pas descendante. Pour cela, on utilise la fonction stepAIC du module MASS.

```
model_final<-stepAIC(model_initial)
```



```

## Start:  AIC=384.57
## Y ~ X3 + X4 + X7 + X9 + X12.Score + X13 + X15 + X17_regroup
##
##           Df Deviance    AIC
## - X17_regroup  1   360.67 382.67
## - X3           1   361.03 383.03
## - X4           1   361.49 383.49
## - X13          1   361.59 383.59
## <none>         360.57 384.57
## - X9           1   362.95 384.95
## - X7           1   370.02 392.02
## - X12.Score    3   426.21 444.21
## - X15          2   467.77 487.77
##
## Step:  AIC=382.67
## Y ~ X3 + X4 + X7 + X9 + X12.Score + X13 + X15
##
##           Df Deviance    AIC
## - X3           1   361.13 381.13
## - X4           1   361.61 381.61
## - X13          1   361.68 381.68
## <none>         360.67 382.67
## - X9           1   363.05 383.05
## - X7           1   370.09 390.09
## - X12.Score    3   427.73 443.73
## - X15          2   471.11 489.11
##
## Step:  AIC=381.13
## Y ~ X4 + X7 + X9 + X12.Score + X13 + X15
##
##           Df Deviance    AIC
## - X4           1   362.09 380.09
## - X13          1   362.26 380.26
## <none>         361.13 381.13
## - X9           1   363.54 381.54
## - X7           1   370.23 388.23
## - X12.Score    3   429.07 443.07
## - X15          2   473.52 489.52
##
## Step:  AIC=380.09
## Y ~ X7 + X9 + X12.Score + X13 + X15
##
##           Df Deviance    AIC
## - X13          1   363.34 379.34
## <none>         362.09 380.09
## - X9           1   364.52 380.52
## - X7           1   371.26 387.26
## - X12.Score    3   429.73 441.73

```



```
## - X15          2    476.89 490.89
##
## Step: AIC=379.34
## Y ~ X7 + X9 + X12.Score + X15
##
##              Df Deviance    AIC
## - X9          1    365.10 379.10
## <none>         363.34 379.34
## - X7          1    374.47 388.47
## - X12.Score   3    429.84 439.84
## - X15         2    478.46 490.46
##
## Step: AIC=379.1
## Y ~ X7 + X12.Score + X15
##
##              Df Deviance    AIC
## <none>         365.10 379.10
## - X7          1    376.56 388.56
## - X12.Score   3    439.88 447.88
## - X15         2    479.66 489.66
```

B) Modèle final

```
round(AIC(model_final),digits=1)

## [1] 379.1

summary(model_final)
```



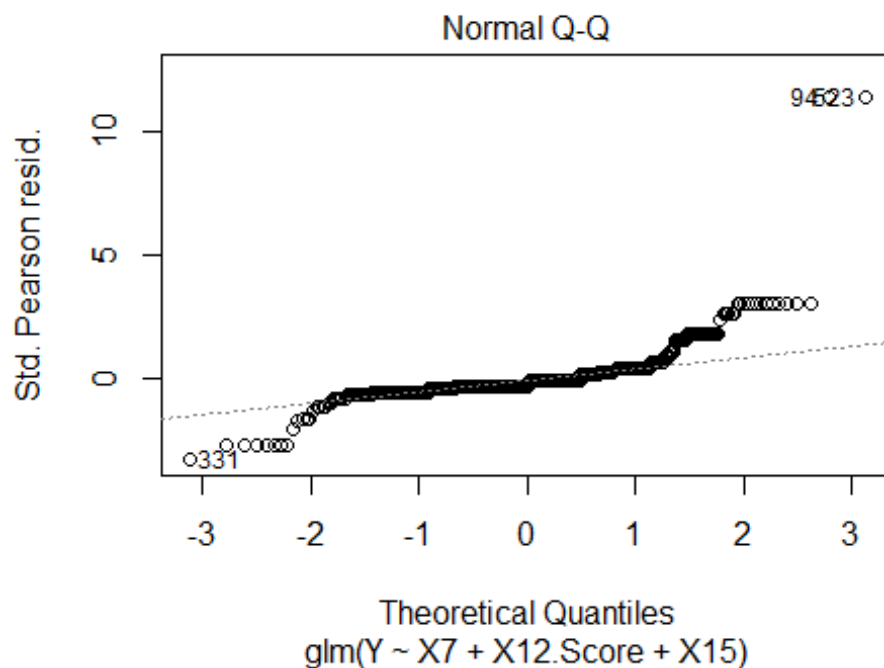
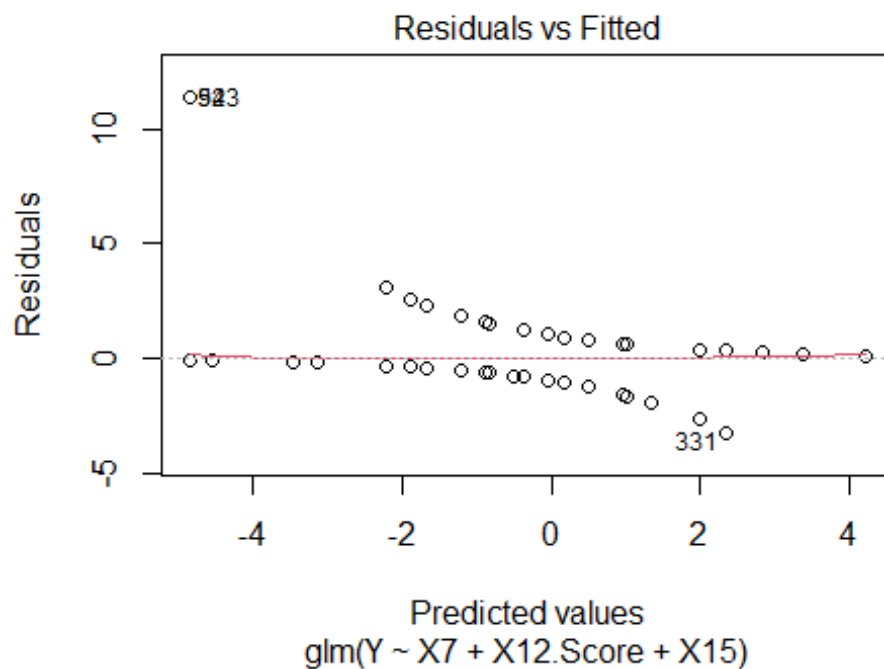
```
##
## Call:
## glm(formula = Y ~ X7 + X12.Score + X15, family = binomial(),
##      data = appren_net)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.2106  -0.5266  -0.4526   0.3406   3.1209
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.5411     0.7041  -6.450 1.12e-10 ***
## X7Yes         1.3897     0.4132   3.363 0.000771 ***
## X12.Score4A    2.6355     0.6729   3.916 8.99e-05 ***
## X12.Score4B    3.6524     0.6610   5.526 3.29e-08 ***
## X12.Score5     4.4906     0.7678   5.849 4.95e-09 ***
## X154          -0.3211     0.3724  -0.862 0.388569
## X155           2.8686     0.4438   6.464 1.02e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 685.63  on 552  degrees of freedom
## Residual deviance: 365.10  on 546  degrees of freedom
## AIC: 379.1
##
## Number of Fisher Scoring iterations: 6
```

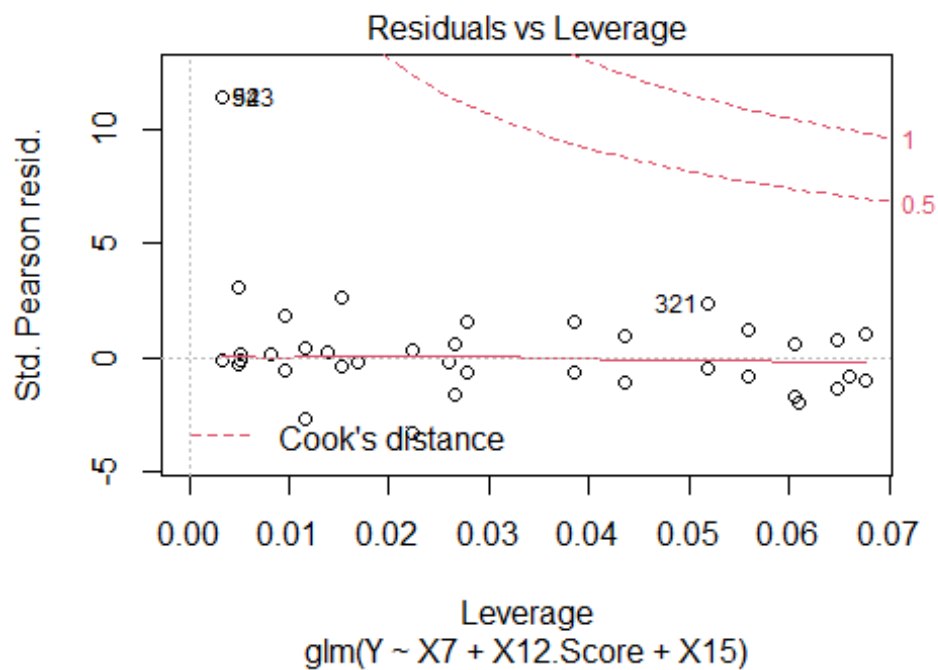
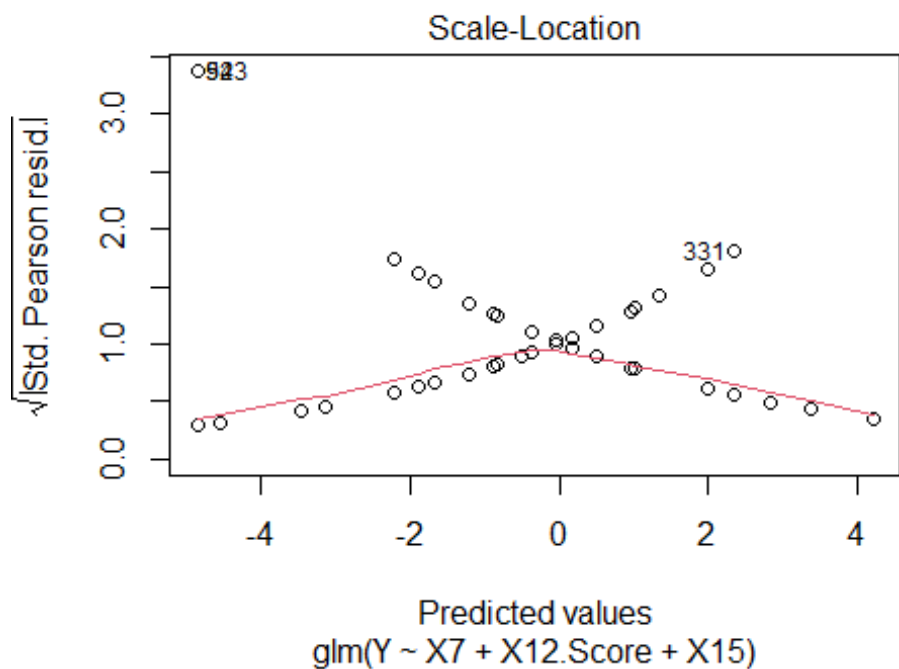
On arrive alors à obtenir une valeur d'AIC plus faible de 379.1, avec trois variables : X7, X12.Score et X15.

C) Analyse des résidus du modèle final

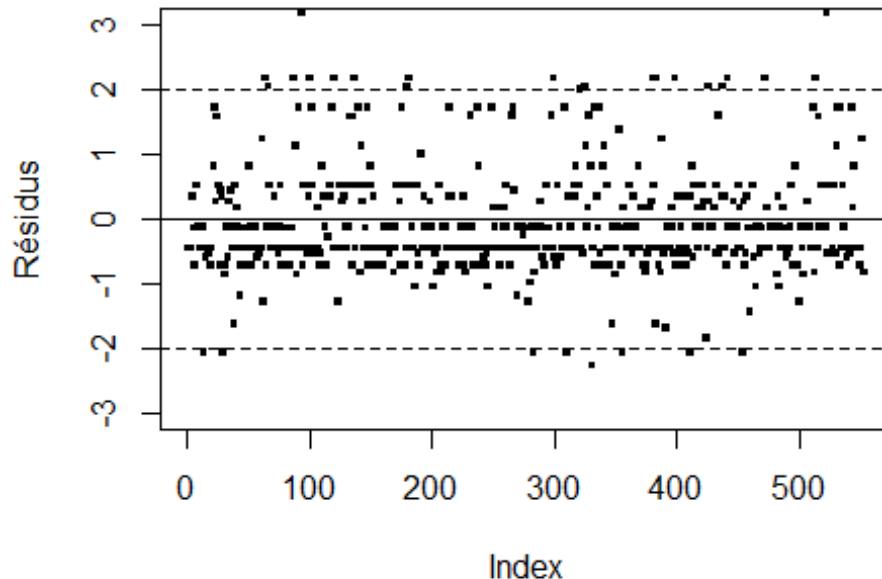
```
plot(model_final)
```







```
res<-rstudent(model_final)
plot(res,pch=15,cex=.5,ylab="Résidus",main="",ylim=c(-3,3))
abline(h=c(-2,0,2),lty=c(2,1,2))
```



Ces graphiques nous permettent de mettre en évidence la normalité et l'homoscédasticité. En effet, on observe que ceux-ci suivent globalement une loi normale (alignés sur le graphique "Normal Q-Q") et qu'ils sont répartis de façon homogène sur le dernier graphique. Cela démontre la régularité de la variance sur le domaine de la variable et donc, qu'il n'y a pas de biais.

On supprime alors les variables qui ne se retrouvent pas dans le modèle final.

```
cols_to_supress=c(2,3,5,7,9)
appren_net<- subset(appren_net,select = -cols_to_supress)
interval_conf<-confint(model_final)
## Waiting for profiling to be done...
md_table(as.data.frame(cbind(c("Intercept","X7Yes","X12.Score4A","X12.Score4B",
"X12.Score5","X154","X155"),interval_conf)))
```

V1	2.5 %	97.5 %
Intercept	-6.12499271168801	-3.30544842355211
X7Yes	0.585217693998381	2.21290834705361



V1	2.5 %	97.5 %
X12.Score4A	1.45457333042284	4.16523546319421
X12.Score4B	2.49960359710749	5.16459949692923
X12.Score5	3.10481563761675	6.17280693011127
X154	-1.03243077630744	0.436255769697223
X155	2.02988496417608	3.77718325104787

D) Evaluation du modèle

Pour évaluer le modèle final, on regarde en premier lieu s'il y a bien une différence entre le modèle initial et le modèle final. Pour cela on réalise un test du chi2 :

```
anova(model_initial, model_final, test = "Chisq")

## Analysis of Deviance Table
##
## Model 1: Y ~ X3 + X4 + X7 + X9 + X12.Score + X13 + X15 + X17_regroup
## Model 2: Y ~ X7 + X12.Score + X15
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      541      360.57
## 2      546      365.10 -5   -4.5327   0.4755
```

On obtient alors une p-value > 0.05, ce qui signifie qu'il n'y a pas de différence significative entre le modèle initial et le modèle final. Cela montre alors que nos deux modèles sont cohérents car peu de variables ont été supprimées lors de la méthode de minimisation de l'AIC.

Dans un second temps on réalise un test statistique de Wald, pertinent au vu de l'utilisation du modèle GLM.

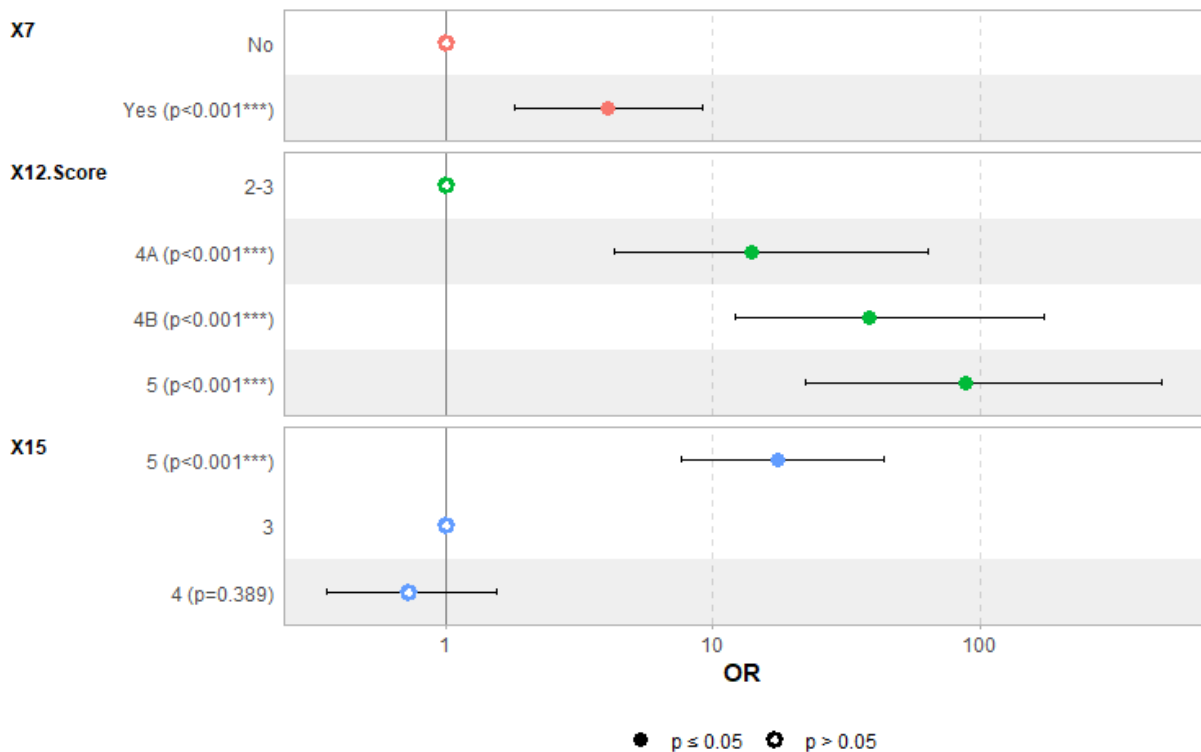
```
car::Anova(model_final, test.statistic = "Wald")

## Analysis of Deviance Table (Type II tests)
##
## Response: Y
##           Df  Chisq Pr(>Chisq)
## X7          1 11.309  0.0007714 ***
## X12.Score    3 47.084  3.336e-10 ***
## X15          2 84.438  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

On retrouve bien que les variables X7, X12.Score et X15 influencent significativement la valeur de la variable cancer, comme on pouvait le voir avec le summary du modèle final au début de cette partie.

```
ggcoef_model(model_final, exponentiate = TRUE)
```





```
forest_model(model_final)
```

Variable	N	Odds ratio	p
X7	No 490	Reference	
	Yes 63	4.01 (1.80, 9.14)	<0.001
X12.Score	2-3 118	Reference	
	4A 210	13.95 (4.28, 64.41)	<0.001
	4B 167	38.57 (12.18, 174.97)	<0.001
	5 58	89.18 (22.31, 479.53)	<0.001
X15	3 87	Reference	
	4 331	0.73 (0.36, 1.55)	0.4
	5 135	17.61 (7.61, 43.69)	<0.001

Pour continuer l'évaluation de notre modèle nous avons ensuite analysé les odds ratios.



Sur le graphique on peut observer l'effet délétère ou bénéfique d'une variable par rapport à la variable cancer. En effet, lorsque le point est à droite de la modalité de référence, qui a un effet neutre sur la valeur de la variable cancer, on observe un effet délétère. On retrouve visuellement cela pour les variables X7 et X12 par exemple, ce qui est confirmé par le fait que les p-value sont inférieures à 0.05.

A l'inverse, lorsqu'il est à gauche cela traduit un effet bénéfique, comme observé pour la modalité 4 de la variable X15 mais, cela ne semble pas significatif au vu de la valeur de la p-value.

3) Validation du modèle

Pour finir, nous avons essayé de valider notre modèle en calculant les accuracy sur les différentes cohortes et en réalisant des prédictions.

L'accuracy correspond à la somme des vrais positifs et vrais négatifs divisée par le nombre total et multipliée par 100 pour exprimer les résultats sous forme de pourcentage. Cette valeur nous permet d'évaluer la qualité du modèle.

En plus de cela on analyse la spécificité et la sensibilité. La spécificité correspond au taux de vrais négatifs et la sensibilité correspond quant-à elle au taux de vrais positifs. Ces deux paramètres sont importants car ils permettent d'analyser la capacité du test à catégoriser les patients. Pour nos données, ces deux paramètres ont une importance équivalente, on va donc regarder s'ils sont proches. On réalise aussi des matrices de confusion.

A) Accuracy sur la cohorte d'apprentissage

```
pred_appren<-predict.glm(model_final,appren_net, type="response")
accuracy_appren<-accuracy(pred_appren,appren_net$Y)
md_table(as.data.frame(accuracy_appren), fit=F)
```

Accuracy	Spécificité	Sensibilité
87.3	87	87.5

Premièrement, on observe que l'accuracy sur la cohorte d'apprentissage, à partir de laquelle nous avons créé notre modèle, est plutôt importante : 87.3%. Cela signifie que la qualité de notre modèle est optimale.

On observe que les pourcentages de spécificité et sensibilité sont très importants, respectivement 87% et 87.5%. Au vu des proportions de malins/ bénins dans le modèle initial, le fait que la spécificité et la sensibilité soient proches montre que la normalisation a bien été effectuée et que l'on peut alors prédire aussi bien ces deux modalités.

B) Accuracy sur la cohorte de validation interne Nice

```
pred_valid_interne<-predict.glm(model_final,valid_2, type="response")
accuracy_valid_interne<-accuracy(pred_valid_interne,valid_2$Y)
md_table(as.data.frame(accuracy_valid_interne), fit=F)
```

Accuracy	Spécificité	Sensibilité
----------	-------------	-------------



Accuracy	Spécificité	Sensibilité
76.6	87.5	84.7

C) Accuracy sur la cohorte de validation externe Toulouse

```
pred_valid_externe<-predict.glm(model_final,valid,type="response")
accuracy_valid_externe<-accuracy(pred_valid_externe,valid$Y)
md_table(as.data.frame(accuracy_valid_externe),fit=F)
```

Accuracy	Spécificité	Sensibilité
71.9	77.4	81.5

Deuxièmement, on s'intéresse aux accuracy sur les cohortes de validation interne et externe. Toutes deux étant supérieures à 70%, respectivement 76.6% et 71.9%, ce qui confirme la qualité de notre modèle.

Pour les deux cohortes de validation, on observe que les spécificités et sensibilités sont assez élevées, bien que plus faibles que celles de la cohorte d'apprentissage (résultat attendu). De plus, celle-ci sont toujours assez similaires, ce qui correspond à ce que l'on attendait et qui confirme la qualité de notre modèle.

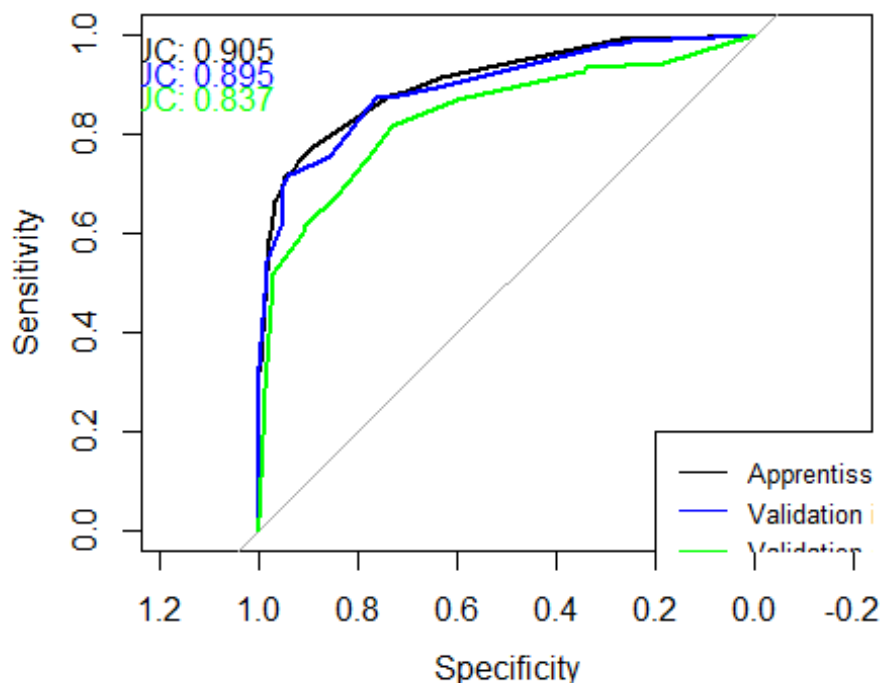
D) Courbe de ROC

Pour finir, on réalise une courbe de ROC pour analyser visuellement la spécificité et la significativité de notre modèle.

```
roc_appren<-roc(appren_net$Y,pred_appren)
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
roc_valid_int<-roc(valid_2$Y,pred_valid_interne)
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
roc_valid_ext<-roc(valid$Y,pred_valid_externe)
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

plot(roc_appren,print.auc=T,print.auc.x=1.3,print.auc.y=1)
plot(roc_valid_int,print.auc=T,col="Blue",print.auc.col="Blue",add=T,print.auc.x=1.3,print.auc.y=0.95)
plot(roc_valid_ext,print.auc=T,col="green",print.auc.col="green",add=T,print.auc.x=1.3,print.auc.y=0.9)
legend(0.2, 0.2, c("Apprentissage","Validation interne","Validation externe"),
,col=c("black","Blue","green"), lty=1, cex=0.8)
```





#Calcul des intervalles de confiance des AUC de chacune des cohortes

```
ci.auc(roc_appren)
```

```
## 95% CI: 0.8761-0.9334 (DeLong)
```

```
ci.auc(roc_valid_int)
```

```
## 95% CI: 0.8512-0.9383 (DeLong)
```

```
ci.auc(roc_valid_ext)
```

```
## 95% CI: 0.7886-0.8848 (DeLong)
```

On observe alors que les aires sous la courbe (AUC) des 3 courbes (3 cohortes) sont assez importantes et similaires. On vérifie bien que les intervalles de confiance se recoupent entre eux, ce qui termine l'évaluation de la qualité de notre modèle. Le fait que les aires soient importantes démontre un certain équilibre entre spécificité et significativité de notre modèle.

V) Bonus : Test XGB avec cross validation

** ressource : <https://www.kaggle.com/rtatman/machine-learning-with-xgboost-in-r/notebook> **

```
df$Y<-as.numeric(df$Y)-1
df$X7<-as.numeric(df$X7)
df$X12.Score<-as.numeric(df$X12.Score)
df$X15<-as.numeric(df$X15)
```



```

df_xgb_imp<-df[,c("Y", "X15", "X7", "X12.Score")]

df_remove_Y<-df_xgb_imp %>% select(-df_xgb_imp$Y)

diseaseLabels <- df_xgb_imp %>% select(Y) %>% magrittr::not()

diseaseInfo_numeric <- df_remove_Y
diseaseInfo_matrix <- data.matrix(diseaseInfo_numeric)
numberOfTrainingSamples <- round(length(diseaseLabels) * .7) #permet de séparer en 30%-70 pour cross validation

train_data <- diseaseInfo_matrix[1:numberOfTrainingSamples,]
train_labels <- diseaseLabels[1:numberOfTrainingSamples]

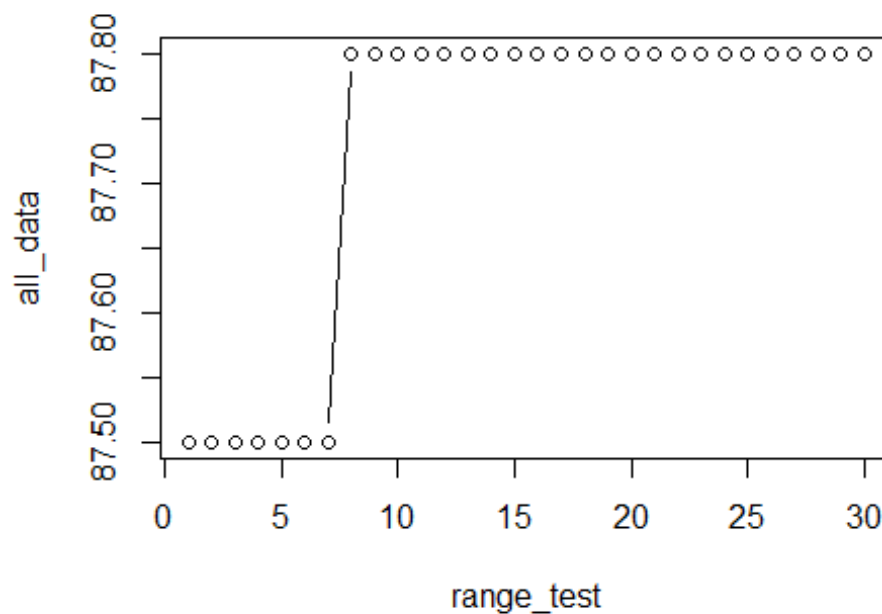
test_data <- diseaseInfo_matrix[-(1:numberOfTrainingSamples),] #tableau apprentissage avec toutes les variables explicatives
test_labels <- diseaseLabels[-(1:numberOfTrainingSamples)] #tableau d'apprentissage avec les lignes de la variable à expliquer

#création de matrice des données utilisables par XGB
dtrain <- xgb.DMatrix(data = train_data, label= train_labels)
dtest <- xgb.DMatrix(data = test_data, label= test_labels)

bst<-bestxgb(nround_init = 1,nround_final = 30,pas=1)

```





```
md_table(bst)
```

bst_accuracy	bst_nround
87.8	8

On choisit donc `n_round = 8` car l'accuracy semble être maximisée dans ce cas-là.

```
model <- xgboost(data = dtrain,
                 nround = bst$bst_nround,
                 objective = "binary:logistic",
                 verbose=0)
```

```
pred_i <- predict(model, dtest)
```

```
accuracy_calc_xgb<-accuracy(pred_i,test_labels)
md_table(accuracy_calc_xgb)
```

Accuracy	Spécificité	Sensibilité
87.8	87.7	88.1

On a une meilleure accuracy avec le modèle XGB qu'avec le modèle glm.



```

df_test<-as.data.frame(test_labels)
df_test$test_labels [df_test$test_labels == "TRUE"] <- 1
df_test$test_labels [df_test$test_labels == "FALSE"] <- 0
df_test$test_labels<-as.factor(df_test$test_labels)

df_test<-cbind(df_test,as.data.frame(pred_i))

roc_xgb_i<-roc(df_test$test_labels,pred_i)

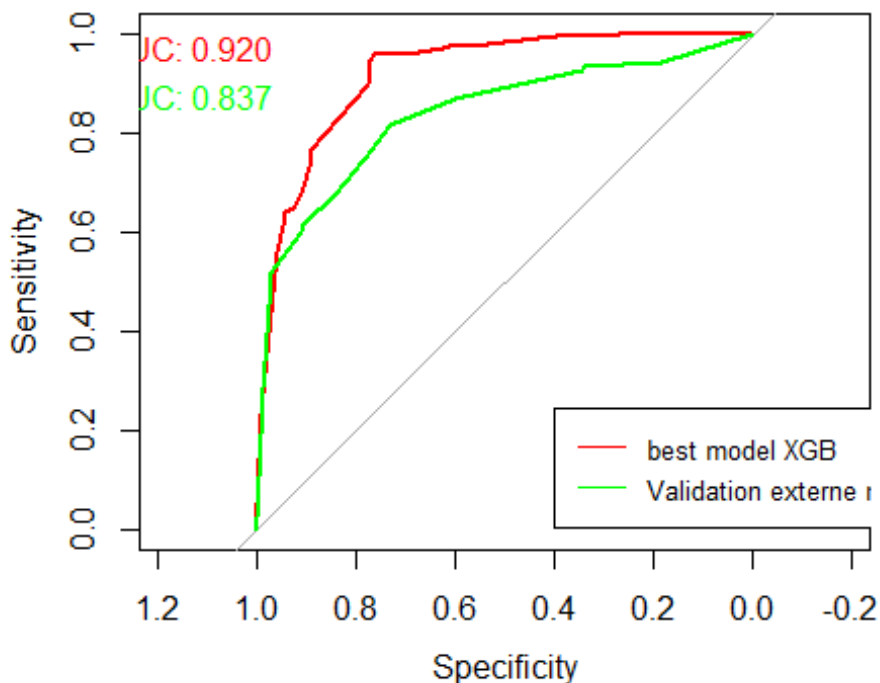
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

roc_valid_ext<-roc(valid$Y,pred_valid_externe)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

plot(roc_xgb_i,print.auc=T,col="red",print.auc.x=1.3,print.auc.y=1)
plot(roc_valid_ext,print.auc=T,col="green",print.auc.col="green",add=T,print.
auc.x=1.3,print.auc.y=0.9)
legend(0.40, 0.245, c("best model XGB","Validation externe modèle glm"),col=c
("Red","green"), lty=1, cex=0.8)

```



On observe visuellement que le modèle XGB est beaucoup plus performant pour prédire si le patient testé a ou non un cancer.

