DEVOIR D'EPIDEMIOLOGIE :

UTILISATION DU SCORE DE PROPENSION

Sarah F FELDMAN
M2MSR – avril 2017

SOMMAIRE

INTRODUCTION

Le cathétérisme cardiaque droit (CCD) est un examen invasif utilisé en soins intensif pour mesurer directement la fonction cardiaque, ce qui permettrait selon certains médecins une meilleure prise en charge du patient et donc une augmentation de la survie. Cependant pour montrer une telle causalité il faudrait un essai randomisé. Or nous avons ici les données d'une cohorte prospective, non interventionnelle, de patients hospitalisés en réanimation, certains ayant été cathétérisés, d'autres non. Le traitement par CCD n'a pas été randomisé, il y a donc lieu de penser que l'examen a été réalisé préférentiellement chez certains types de patients, potentiellement les patients les plus graves qui ont donc plus de risque de décéder. Cela constitue un biais d'indication qui rend l'analyse de l'efficacité du CCD ininterprétable. Pour pouvoir analyser l'efficacité du CCD par sonde de Swan Ganz, nous allons prendre en compte les biais d'indication probables à l'aide d'un score de propension, nous permettant d'améliorer le niveau de causalité de la relation CCD/Décès si elle existe.

I - ETAPES PRELIMINAIRES

1) Description de la base de données

La base de données comporte les informations de 5735 patients pour 63 variables. Ce sont des données transversales avec une ligne par patient (pas de doublon), chaque variable ayant été mesurées une seule fois.

2) Data management

Je fais une première étape de data management "basique" pour mettre les variables au bon format (dates, numérique et facteur), m'assurer que le numéro de patient est en caractère, créer une variable SWAN en 0/1 et une autre en TRUE/FALSE selon les différents modèles utilisés par la suite. Je change également les "" en "no category"" pour la variable CAT2 ou en NA pour les dates. Je choisis No comme référence pour la variable CA.

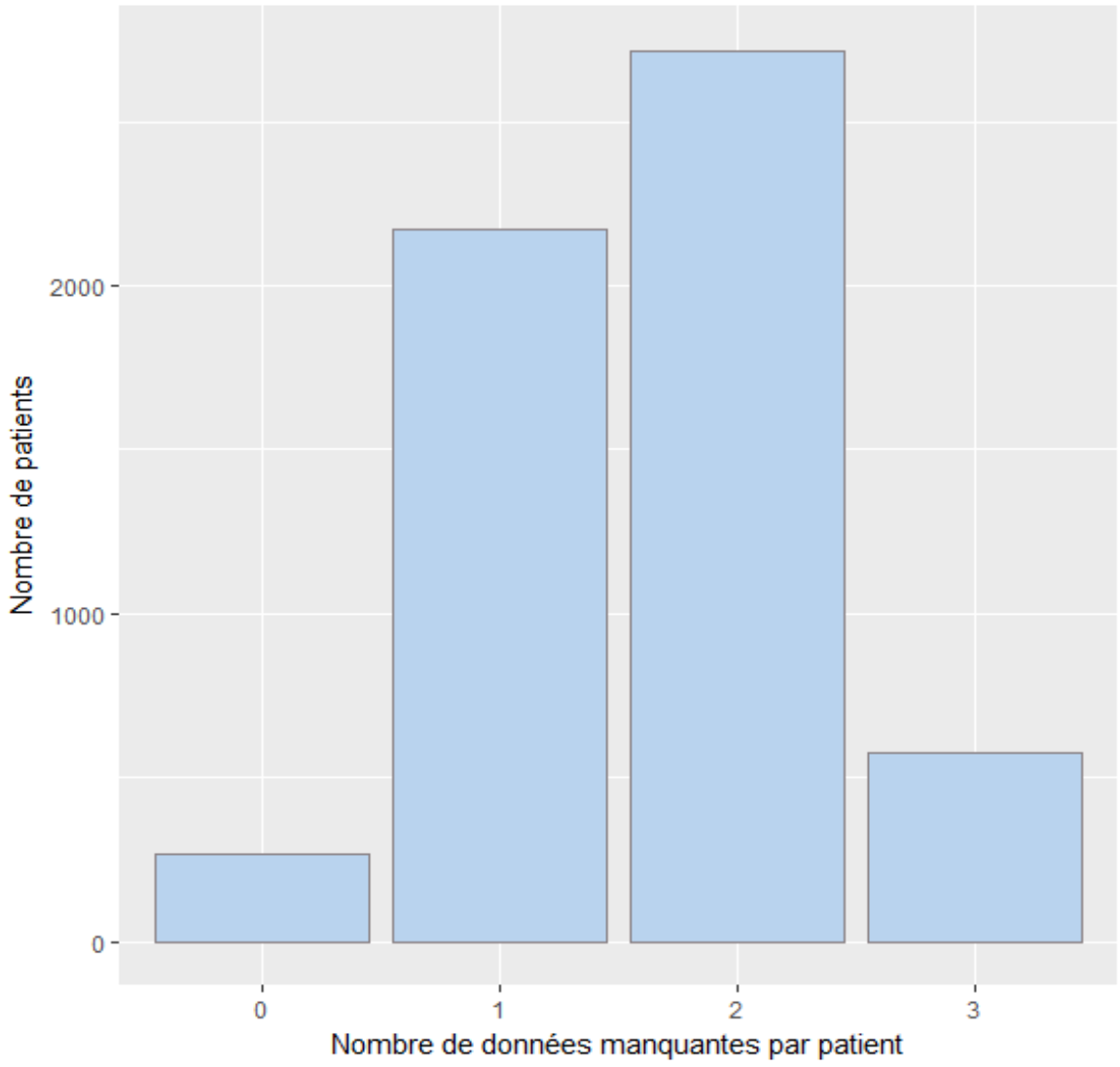
3) Vérification de la durée de suivi

Je vérifie les variables dates de décès, date de dernières nouvelles, et durée de suivi pour l'outcome décès à 30 jours :

- 3 patients pour lesquels la date de décès est antérieure à la date de dernière nouvelle et 2151 pour lesquels date de dernières nouvelles est antérieure au décès. Je modifie la variable date de dernières nouvelles ; si la date de décès est non nulle, alors la date de dernières nouvelles est la date de décès.
- Nous prendrons comme outcome le décès à 30 jours, je dois donc utiliser les variables DTH30, qui est le décès à 30 jours et T3D30 qui est le temps de suivi adapté à cet outcome. Afin de vérifier ces 2 variables, je recrée la variable temps de suivi pour l'outcome décès à 30 jours.

Pour cela je crée d'abord la variable temps de suivi comme la différence entre la date des dernières nouvelles et la date d'admission dans l'étude. Puis je modifie cette variable telle que si le temps de suivi est supérieur à 30 jours alors je le ramène à 30 jours. Et la variable décès à 30 jours est modifiée comme ceci à partir de la variable décès : si le décès est survenu après 30 jours, alors la variable décès à 30 jours vaut 0. En comparant les variables DTH30 et T3D30 avec celles recrées, je trouve une différence pour 8 patients qui ont un temps de suivi inférieur à 30 jours et qui sont pourtant noté avec un temps de suivi de 30 jours pour un outcome décès à 30 jours. Et ça ne vient pas de la modification de la variable date de dernières nouvelles. Pour la suite du devoir, je décide de prendre T3D30 comme variable de temps de suivi pour le décès à 30 jours.

4) Données manquantes



Figue 1. Nombre de valeurs manquantes par patient.

La base de données est globalement de bonne qualité, en effet aucun sujet n'a plus de 4 valeurs manquantes (sur les 63 variables). (Figure 1)

Si je regarde maintenant le nombre de valeurs manquantes pour chaque variable, je vois que ADLD3P (échelle ADL) et URIN1 (volume urinaire des 24h) ont respectivement 75% et 53% de données manquantes tandis que les autres variables n'ont aucune ou moins de 5% de données manquantes.

5) Analyse descriptive préliminaire afin de repérer les incohérences

a- Présentation des variables de l'examen clinique et biologique :

- ADLD3P = Activities of daily living (ADL) : échelle d'autonomie de 0 à 12 (score>6 signe une dépendance).
- DAS2D3PC = Duke Activity Status Index (DASI) : auto-questionnaire de 12 items mesurant l'activité fonctionnelle. Le score va de 0 à 58.2, plus le score est élevé, meilleur est l'activité fonctionnelle.
- DNR1 = Do not resuscitate : 1 pour une interdiction de réanimation cardiopulmonaire, 0 sinon.
- SURV2MD1 = Probabilité de survie à 2 mois, estimée par model : de 0 à 1.
- APS1 = APACHE III Acute Physiology scores : score de prediction du risque de mortalité de patients hospitalisés en soins intensifs. Plus le score est élevé plus le risque de mortalité est important. Le score va de 0 à 299 mais ici seule la partie physiologie du score est utilisée.
- SCOMA1 = score de Glasgow. Ce score évalue l'état de conscience du patient: un score de 3 équivaut à un coma profond, un score de 15 est un état de conscience normal.
- PAFI1 = rapport PAO2/FIO2. Il permet de diagnostiquer une agression pulmonaire aigue(rapport>300), un rapport <200 definissant le syndrome de detresse respiratoire aigue et l'ECMO est envisageable en cas de rapport <50.
- PH1 = pH. Un pH normal varie entre 7,38 et 7.42.
- HEMA1 = taux d'hématocrite, normalement compris entre 41 et 50% environ.
- PACO21 = Pression artérielle en CO2. La PaCO2 normale varie de 35 à 45mmHg.
- ALB1 = taux d'albumine. Le taux d'albumine normal varie entre 25 et 44 g/L.
- WTKILO1 = poids
- TEMP1 = température corporelle
- HRT1 = fréquence cardiaque
- MEANBP1 = pression artérielle moyenne
- RESP1 = fréquence respiratoire
- SOD1 = natrémie
- POT1 = kaliémie
- CREA1 = créatininémie
- BILI1 = bilirubinémie
- URIN1 = volume urinaire des 24h

b- valeurs aberrantes

Pour détecter les valeurs aberrantes je dispose de plusieurs méthodes. Je regarde le tableau descriptif en calculant pour chaque variable quantitative la moyenne, l'intervalle interquartile et surtout le range (tableau non présenté). Je croise également les variables fréquence cardiaque, pression artérielle et fréquence respiratoire à la recherche d'incohérence.

Enfin je regarde la distribution des variables ce qui me permet de repérer éventuellement d'autres incohérences (voir annexe : Figure A-1). Par exemple nous voyons un pic à 0 pour la variable WTKILO1 (nous l'avions déjà vu grâce au tableau descriptif, mais c'est un deuxième filet de sécurité). On observe également une pause dans les inclusions (variable SADMDTE), mais je ne sais pas comment le prendre en compte.

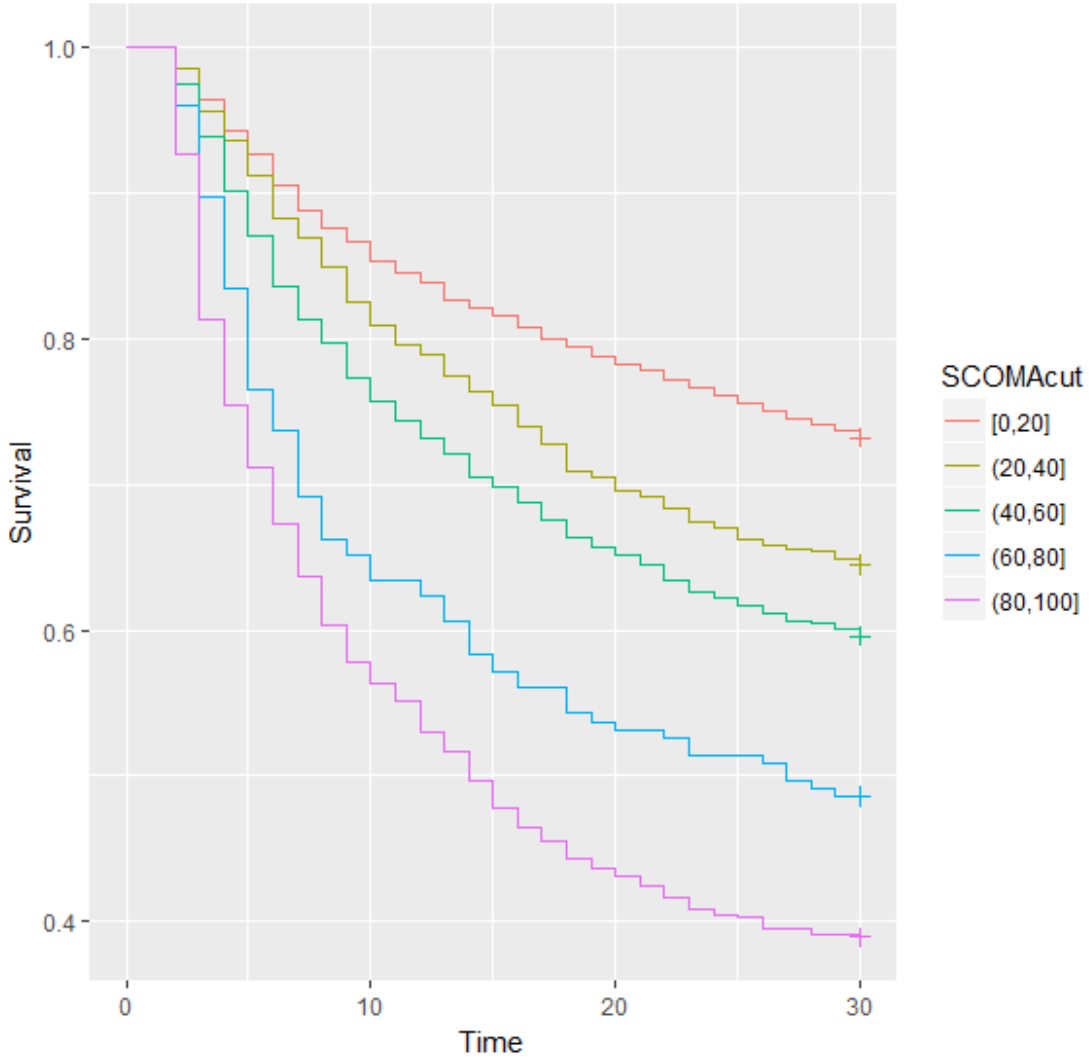


Figure 2. Décès en fonction du score de Glasgow modifié(de 0 à 100) et découpé en quintile.

Les scores qui ont été mesurés, semblent tous avoir des résultats plausibles excepté le score de Glasgow qui a été modifié : d'un score allant de 3 à 15 on passe à un score allant de 0 à 100. Le score de 100 est probablement lié à un coma sévère car les patients avec un score plus élevé ont un risque de décès augmenté (alors que dans l'échelle d'origine, 3 correspond au coma sévère, et 15 à une conscience normale) (Figure 2).
Je laisse la variable telle quelle car elle a de toute évidence été modifiée, je ne veux pas la remodifier une seconde fois.
La température corporelle peut aller de 27° en cas d'hypothermie profonde à 42-43° en cas de forte fièvre, donc je ne modifie pas.
Un patient ne peut pas avoir un poids de 0, ces 515 patients sont recodés en NA pour le poids.
Un patient peut être en aplasie mais je ne suis pas sûr qu'une hyperleucocytose de plus de 40 leucocytes/10^9L soit possible, cependant dans le doute concernant le taux maximum en cas de leucémie et au vue de la distribution de la variable qui semble plausible, je laisse telle quelle.
Hématocrite : une anémie chronique peut entrainer une anémie profonde avec une hématocrite de 10%, en dessous de cette valeur je mets NA. (Un taux de 66% est possible).
Un taux de PACO2 de 156 mmHg me semble complètement impossible, au-delà de 100 mmHg je mets NA. Un pH de 6.6 est extrêmement faible et n'est pas viable mais je ne suis pas sûre qu'on ne puisse pas l'observer en réanimation.
Une créatininémie de 2200umol/L ou 25 mg/dL est possible en cas d'insuffisance rénale terminale. Un volume urinaire de 0 et de 9000mL est possible également (mais cette variable n'est de toute façon pas utilisée pour l'analyse).
Une valeur normale d'albumine se situe entre 3.5 et 5 g/dL. Les patients avec un taux d'albumine supérieur à 10g/dL auront une valeur NA pour l'albumine. Les valeurs de natrémie, kaliémie, fréquence respiratoire, fréquence cardiaque, pression artérielle moyenne et bilirubine semblent plausibles.

En observant ensemble la fréquence cardiaque, la fréquence respiratoire et la tension artérielle, je note certaines incohérences : parfois 2 de ces variables peuvent valoir 0 mais pas la troisième, et parfois une seule de ces variables vaut 0. Ainsi 1 patient à une tension artérielle à 0 mais une fréquence cardiaque et une fréquence respiratoire différente de 0, 47 patients ont une fréquence respiratoire à 0 mais une fréquence cardiaque et une tension artérielle différente de 0 et 3 patients ont une fréquence cardiaque à 0 mais une fréquence respiratoire et une tension artérielle différente de 0.

Je modifie donc ces 3 variables selon les règles suivantes :

- Si deux de ces variables valait 0, alors la valeur de 0 était systématiquement attribuée à la troisième.

- Si une seule de ces variables valait 0 alors elle était systématiquement transformée en valeur manquante.

II - IMPUTATION DES VALEURS MANQUANTES

Les valeurs manquantes doivent absolument être prises en charge pour la suite du projet car on ne peut pas faire de score de propension pour les patients ayant une ou plusieurs variables explicatives manquantes.

1) Règles d'imputation

Règles de décision concernant l'imputation ou non des variables :

- Une variable avec plus de 20% de données manquantes ne sera pas imputée et ne sera pas inclue dans l'analyse.
- Un patient avec plus de 50% de données manquantes ne sera pas inclus dans l'analyse.
- Un patient n'ayant pas d'information concernant la variable Swan Ganz qui est la variable explicative d'intérêt ne sera pas inclus dans l'analyse car non informatif.
- J'imputerai les variables avec des valeurs manquantes uniquement si plus de 5% des patients ont des valeurs manquantes, sinon je supprimerai simplement ces patients.

2) Description des valeurs manquantes (valeurs aberrantes transformées en valeurs manquantes)

Je vais donc dans un premier temps observer les données manquantes par variable et par sujet, après avoir transformé les valeurs aberrantes en valeurs manquantes.

Comme vu précédemment, Le score d'autonomie ADL(ADLD3P) a environ 75% de données manquantes et la variable volume d'excrétion urinaire (URIN1) a plus de 50% de données manquantes. Je supprime donc ces deux variables de l'analyse.
250 patients n'ont aucune donnée manquante, 5485(95.6%) patients ont au moins 1 donnée manquante dont : 1930 patients avec 1 donnée manquante, 2698 patients avec 2 de données manquantes, 796 patients avec 3 données manquantes, 64 patients avec 4 données manquantes et 1 patient avec 5 données manquantes. Tous les patients ont donc moins de 50% de données manquantes.
Il n'y a pas de valeurs manquantes concernant la variable explicative d'intérêt SWAN (SWAN = 0 pas de traitement par CCD, SWAN = 1 traitement par CCD).

En supprimant ADLD3P et URIN1 de l'analyse, j'ai 608 patients avec au moins une donnée manquante, soit plus de 5 % de patients. Je ne peux donc pas simplement supprimer les patients avec des données manquantes. 6 variables ont au moins une donnée manquante : la tension artérielle (15 NA), la fréquence cardiaque (60 NA), l'albuminémie (2 NA) l'hématocrite (8 NA), la PaCO2 (21 NA) et le poids (515 NA).

J'impute les variables explicatives avec le package mice en faisant l'hypothèse que les données manquantes le sont aléatoirement. J'utilise la technique d'imputation multiples mais je ne réalise qu'un seul jeu de de données imputé. Toutes les variables du jeu de

données sont prédictives (y compris l'outcome DTH30, le temps de suivi jusqu'à cet outcome et SWAN, selon Jonathan A C Sterne et al 2009 BMJ) excepté les dates. Par contre je ne prends pas en compte DEATH qui n'est pas notre outcome ni le temps de suivi jusqu'à DEATH.

3) Description des variables imputées avant et après imputation

Je compare les variables qui ont été imputées, avant et après imputation. Je ne compare que visuellement, je ne fais pas de test.

	value	missing values	range
MEANBP1	63 (50-115)	15 (0.3%)	0 - 259
HRT1	124 (100-142)	60 (1%)	0 - 250
ALB1	3.5 (2.6-3.5)	2 (0%)	0.299987793 - 6.599609375
HEMA1	30 (26.1-36.3)	8 (0.1%)	10 - 66.1875
PACO21	37 (31-42)	21 (0.4%)	1 - 100
WTKILO1	72.2 (60.4-85.2)	515 (9%)	19.5 - 244

Table 1. Variables qui ont été imputées, avant imputation

	value	missing values	range
MEANBP1	63 (50-114.5)	0 (0%)	**0 - 259**
HRT1	124 (100-142)	0 (0%)	**0 - 250**
ALB1	3.5 (2.6-3.5)	0 (0%)	**0.299987793 - 6.599609375**
HEMA1	30 (26.1-36.3)	0 (0%)	**10 - 66.1875**
PACO21	37 (31-42)	0 (0%)	**1 - 100**
WTKILO1	**72.2 (60.4-85.2)**	**0 (0%)**	**19.5 - 244**

Table 2. Variables qui ont été imputées, après imputation

La distribution des variables a l'air semblable avant et après imputation.

III - DESCRIPTION DE LA POPULATION DE L'ETUDE (après imputation)

	Pas de CCD	CCD	valeur
SEX_Female	1637 (46.1%)	906 (41.5%)	2543 (44.3%)
SEX_Male	1914 (53.9%)	1278 (58.5%)	3192 (55.7%)
RACE_black	585 (16.5%)	335 (15.3%)	920 (16%)

RACE_other	213 (6%)	142 (6.5%)	355 (6.2%)
RACE_white	2753 (77.5%)	1707 (78.2%)	4460 (77.8%)
INCOME_$11-$25k	713 (20.1%)	452 (20.7%)	1165 (20.3%)
INCOME_$25-$50k	500 (14.1%)	393 (18%)	893 (15.6%)
INCOME_> $50k	257 (7.2%)	194 (8.9%)	451 (7.9%)
INCOME_Under $11k	2081 (58.6%)	1145 (52.4%)	3226 (56.3%)
NINSCLAS_Medicaid	454 (12.8%)	193 (8.8%)	647 (11.3%)
NINSCLAS_Medicare	947 (26.7%)	511 (23.4%)	1458 (25.4%)
NINSCLAS_Medicare & Medicaid	251 (7.1%)	123 (5.6%)	374 (6.5%)
NINSCLAS_No insurance	186 (5.2%)	136 (6.2%)	322 (5.6%)
NINSCLAS_Private	967 (27.2%)	731 (33.5%)	1698 (29.6%)
NINSCLAS_Private & Medicare	746 (21%)	490 (22.4%)	1236 (21.6%)
AGE*	64.59 (50.08-74.97)	63.5 (50.21-72.65)	64.05 (50.15-73.93)
EDU*	12 (10-13)	12 (10-14)	12 (10-13)
CAT1_ARF	1581 (44.5%)	909 (41.6%)	2490 (43.4%)
CAT1_CHF	247 (7%)	209 (9.6%)	456 (8%)
CAT1_Cirrhosis	175 (4.9%)	49 (2.2%)	224 (3.9%)
CAT1_Colon Cancer	6 (0.2%)	1 (0%)	7 (0.1%)
CAT1_Coma	341 (9.6%)	95 (4.3%)	436 (7.6%)
CAT1_COPD	399 (11.2%)	58 (2.7%)	457 (8%)
CAT1_Lung Cancer	34 (1%)	5 (0.2%)	39 (0.7%)
CAT1_MOSF w/Malignancy	241 (6.8%)	158 (7.2%)	399 (7%)
CAT1_MOSF w/Sepsis	527 (14.8%)	700 (32.1%)	1227 (21.4%)
CAT2_Cirrhosis	27 (0.8%)	11 (0.5%)	38 (0.7%)
CAT2_Colon Cancer	1 (0%)	1 (0%)	2 (0%)
CAT2_Coma	70 (2%)	20 (0.9%)	90 (1.6%)
CAT2_Lung Cancer	13 (0.4%)	2 (0.1%)	15 (0.3%)
CAT2_MOSF w/Malignancy	171 (4.8%)	58 (2.7%)	229 (4%)
CAT2_MOSF w/Sepsis	406 (11.4%)	420 (19.2%)	826 (14.4%)
CAT2_NoCAT2	2863 (80.6%)	1672 (76.6%)	4535 (79.1%)
CA_No	2652 (74.7%)	1727 (79.1%)	4379 (76.4%)

CA_Metastatic	261 (7.4%)	123 (5.6%)	384 (6.7%)
CA_Localized	638 (18%)	334 (15.3%)	972 (16.9%)
CARDIOHX_1	567 (16%)	446 (20.4%)	1013 (17.7%)
CHFHX_1	596 (16.8%)	425 (19.5%)	1021 (17.8%)
DEMENTHX_1	413 (11.6%)	151 (6.9%)	564 (9.8%)
PSYCHHX_1	286 (8.1%)	100 (4.6%)	386 (6.7%)
CHRPULHX_1	774 (21.8%)	315 (14.4%)	1089 (19%)
RENALHX_1	149 (4.2%)	106 (4.9%)	255 (4.4%)
LIVERHX_1	265 (7.5%)	136 (6.2%)	401 (7%)
GIBLEDHX_1	131 (3.7%)	54 (2.5%)	185 (3.2%)
MALIGHX_1	872 (24.6%)	444 (20.3%)	1316 (22.9%)
IMMUNHX_1	907 (25.5%)	636 (29.1%)	1543 (26.9%)
TRANSHX_1	335 (9.4%)	327 (15%)	662 (11.5%)
AMIHX_1	105 (3%)	95 (4.3%)	200 (3.5%)
DAS2D3PC*	19.66 (15.73-23.46)	19.92 (16.71-23.36)	19.75 (16.06-23.43)
DNR1_1	499 (14.1%)	155 (7.1%)	654 (11.4%)
SURV2MD1*	0.64 (0.49-0.76)	0.6 (0.45-0.72)	0.63 (0.47-0.74)
APS1*	50 (38-62)	60 (47-74)	54 (41-67)
SCOMA1*	0 (0-41)	0 (0-37)	0 (0-41)
WTKILO1*	70 (58.3-82.9)	75.3 (64.1-88.02)	72.2 (60.4-85.2)
TEMP1*	38.09 (36.2-39)	38.09 (36.09-39)	38.09 (36.09-39)
MEANBP1*	68 (52-119)	57 (47-73)	63 (50-114.5)
RESP1*	30 (20-39)	28 (12-37)	30 (14-38)
HRT1*	121 (79-140)	125 (106-145)	124 (100-142)
PAFI1*	224 (148.8-333.31)	168.44 (110-266.62)	202.5 (133.31-316.62)
PACO21*	38 (32-44)	36 (30-40)	37 (31-42)
PH1*	7.4 (7.35-7.46)	7.4 (7.32-7.46)	7.4 (7.34-7.46)
WBLC1*	13.6 (8.2-19.4)	14.7 (8.6-21.2)	14.1 (8.4-20.05)
HEMA1*	31 (26.6-39)	29 (26-33.4)	30 (26.1-36.3)
SOD1*	136 (133-142)	136 (132-141)	136 (132-142)
POT1*	3.8 (3.4-4.6)	3.8 (3.4-4.6)	3.8 (3.4-4.6)
CREA1*	1.3 (0.9-2)	1.8 (1.2-3)	1.5 (1-2.4)
BILI1*	1.01 (0.7-1.2)	1.01 (1-1.7)	1.01 (0.8-1.4)
ALB1*	3.5 (2.7-3.5)	3.5 (2.4-3.5)	3.5 (2.6-3.5)

RESP_1	1481 (41.7%)	632 (28.9%)	2113 (36.8%)
CARD_1	1007 (28.4%)	924 (42.3%)	1931 (33.7%)
NEURO_1	575 (16.2%)	118 (5.4%)	693 (12.1%)
GASTR_1	522 (14.7%)	420 (19.2%)	942 (16.4%)
RENAL_1	147 (4.1%)	148 (6.8%)	295 (5.1%)
META_1	172 (4.8%)	93 (4.3%)	265 (4.6%)
HEMA_1	239 (6.7%)	115 (5.3%)	354 (6.2%)
SEPS_1	515 (14.5%)	516 (23.6%)	1031 (18%)
TRAUMA_1	18 (0.5%)	34 (1.6%)	52 (0.9%)
ORTHO_1	3 (0.1%)	4 (0.2%)	7 (0.1%)

Table 3. Caractéristiques de base des patients (après imputation des valeurs manquantes). *variable quantitative

Il me manque une information essentielle : je ne sais pas si les variables ont été mesurée à l'admission ou après pose de la sonde de Swan Ganz. Je vais considérer que c'est avant pose de la sonde, sinon ça n'a pas de sens.

Je regarde les différences de caractéristiques de base entre le groupe des patients cathétérisés et le groupe des patients non cathétérisés (Table 3). Je ne réalise pas de test car une petite différence numériquement peut être significative du fait de la taille de l'échantillon.
Les patients traités par CCD sont plus souvent des hommes, avec une couverture médicale moins précaire, ils ont plus souvent une comorbidité cardiaque, avec une pression artérielle moyenne plus faible et une fréquence cardiaque plus élevée, plus souvent admis pour pathologie cardiaque que les patients non traités par CCD, et moins souvent admis pour pathologie respiratoire, ils ont une PAO2/FIO2 plus faible(ratio inférieur à 200 signe un syndrome de détresse respiratoire aigu), une pression artérielle en CO2 inferieur à la normale et plus faible que le groupe non traité, une créatininémie plus élevée et un score d'apache plus élevé.

Donc globalement, les patients traités par CCD sont dans un état cardiaque et rénal plus sévère et avait donc de base plus de risque de décéder. Il y a donc un biais d'indication et il faut donc absolument prendre en compte cette différence d'état de base en considération lorsque l'on teste si le traitement par CCD diminue le décès à 30 jours.
Pour cela, nous allons donc réaliser un appariement sur le score de propension.

IV - Appariement sur le score de propension

Le score de propension permet d'avoir pour chaque patient sa probabilité d'être traité par CCD, en fonction de ses caractéristiques de bases. On apparie ensuite un patient traité par CCD avec un patient non traité par CCD qui avait la même probabilité d'être traité que le patient effectivement traité. On se place donc dans la situation d'un essai clinique

randomisé ou chaque patient à la même probabilité de recevoir l'un ou l'autre des traitements.

1) Sélection des variables à intégrer : tests bivariés

Je dois intégrer dans le score de propension les variables qui sont associées significativement avec le décès à 30 jours ou avec le décès à 30 jours et le traitement par cathétérisme. Pour cela je réalise deux séries de tests bivariés. Une première série testant l'association entre chaque variable et le décès à 30 jours et une deuxième série testant l'association entre chaque variable et le traitement par cathétérisme. Pour rappel, les variables ADLD3P et URIN1 ne sont pas dans l'analyse. J'utilise des modèles de régression logistique à une variable explicative, la variable à expliquer étant soit la variable cathétérisme, soit la variable décès à 30 jours (variables toutes deux binaires) et la variable explicative étant chacune des variables à tester.

Les conditions de validité sont toujours respectées car j'ai 9 classes au maximum (variable CAT1) pour 1918 évènements, donc la condition des 5 à 10 variables par variable est toujours respectée.

2 variables sont liées au décès uniquement : TEMP1 et LIVERHX.

34 variables sont liées au décès et au traitement par cathétérisme : RESP, CARD, NEURO, GASTR, HEMA, SEPS, CAT1, CAT2, CA, DAS2D3PC, DNR1, SURV2MD1, APS1, SCOMA1, WTKILO1, MEANBP1, PAFI1, PACO21, PH1, WBLC1, HEMA1, CREA1, BILI1, ALB1, CARDIOHX, CHFHX, DEMENTHX, PSYCHHX, CHRPULHX, GIBLEDHX, MALIGHX, AGE, INCOME, NINSCLAS.

J'aurai donc 36 variables dans le score de propension.

Les 10 variables liées uniquement au cathétérisme ne sont pas prises en comptes dans le score : RENAL, TRAUMA, RESP1, HRT1, SOD1, IMMUNHX, TRANSHX, AMIHX, SEX, EDU.

2) Calcul du score de Propension

Je calcule le score de propension à partir d'un modèle de régression logistique avec comme outcome le traitement par Swan Ganz et comme variables explicatives toutes les variables liées au décès à 30j uniquement (facteurs pronostiques) ou au décès et au traitement (facteurs de confusion). Je n'intègre pas les variables liées uniquement au traitement pour ne pas perdre de puissance à la fin.

Condition de validité :
C'est une régression logistique, je dois avoir 5 à 10 évènement par variable. J'ai 36 variables explicatives dans le score de propension dont 5 variables qualitatives : 9 classes pour CAT1, 7 classes pour CAT2, 3 classes pour CA, 4 classes pour INCOME, 6 classes pour NINSCLAS. Ces variables qualitatives seront donc transformées en 9+7+3+4+6-5= 24 variables binaires. Soit l'équivalent de 36-5+24 = 55 variables dans le modèle logistique pour 1918 évènements. J'ai donc plus de 10 évènements par variable explicative, les conditions de validité sont respectées.

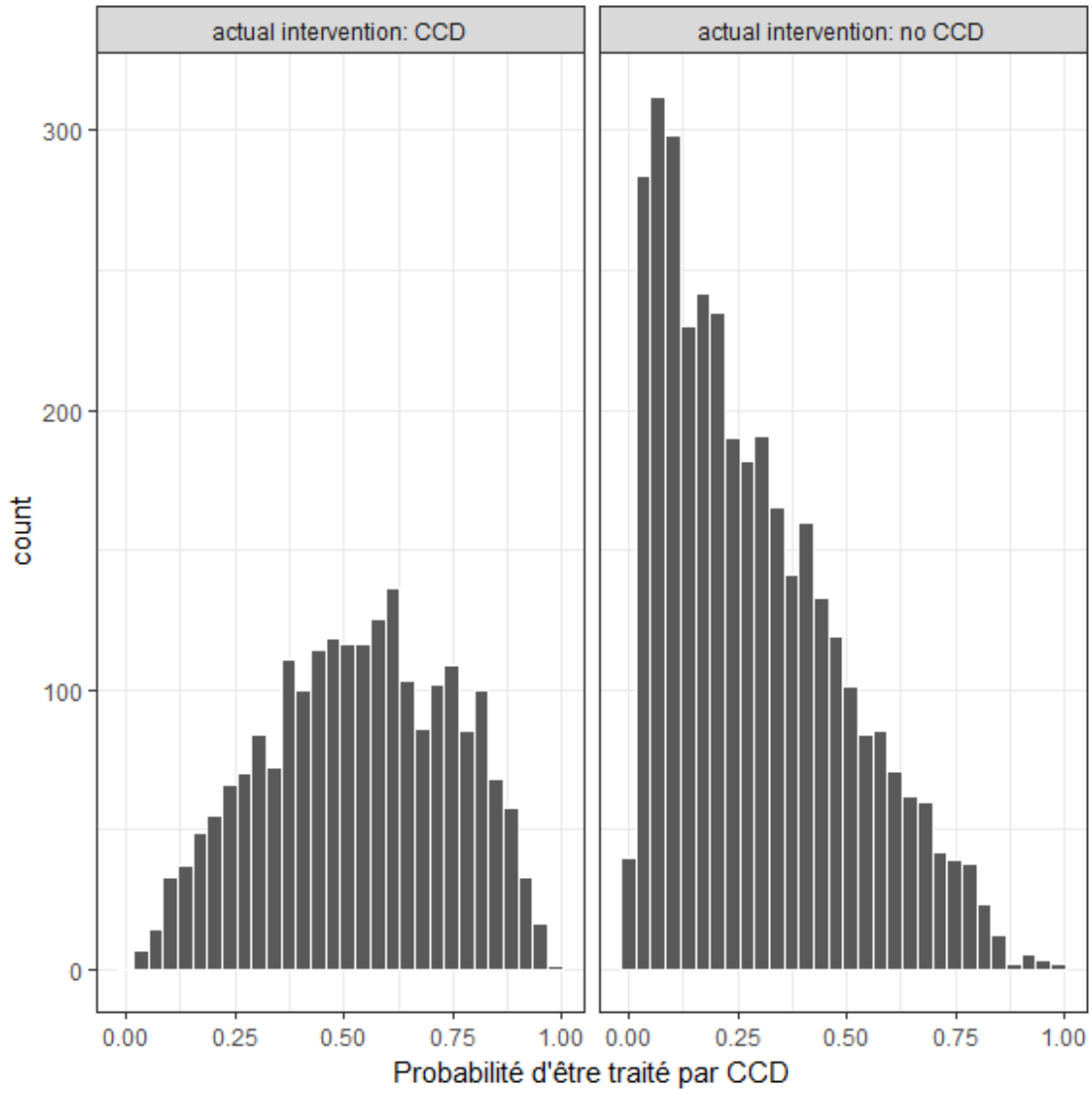


Figure 3. Distribution de la probabilité d'être traité par CCD selon le traitement effectif par CCD.

On voit que la distribution du score n'est pas la même parmi les patients ayant été cathétérisé et ceux n'ayant pas été cathétérisé : beaucoup de patient n'ayant pas été cathétérisé avait une faible probabilité d'être cathétérisé (distribution en L), alors que la distribution est plutôt en cloche centrée autour de 0.5 pour les patients ayant été cathétérisés (Figure 3). On comprend donc que nécessairement l'appariement va supprimer de nombreux patients.

3) Appariement sur le score de propension

A partir du score de propension, j'apparie un sujet cathétérisé avec un sujet non cathétérisé ayant un score de propension proche c'est à dire une probabilité d'être cathétérisé proche. Les sujets non appariés sont écartés de l'analyse. Les sujets devraient donc se ressembler dans les 2 groupes.

Deux packages principalement nous permette de réaliser l'appariement ; le package Matching et le package MatchIt. Le package Matching est celui que je trouve le plus simple pour reconstituer le numéro de paire et qui conserve le plus de patients en gardant les mêmes paramètres.

J'utilise un caliper de 0.2, c'est à dire un seuil d'appariement de 0.2 x sd(logit(score de propension)). En utilisant le package Matching (fonction Match), sans remise, sans ex aequo, avec un ratio 1:1 et avec un caliper de 0.2, je conserve 3142 patients, 1571 dans chaque groupe.

4) Vérification de l'équilibre des variables entre les deux groupes après appariement

L'appariement a normalement permis d'avoir des patients globalement comparables en terme de caractéristiques de base, car ayant la même probabilité d'être traité par CCD dans les deux groupes. Il se peut cependant que certaines variables soit mal équilibrée, et c'est ce que nous allons vérifier ici. Seules les variables ayant servie à construire le score de propension doivent être regardées.

Tout d'abord, je transforme mes variables qualitatives en n-k variables binaires, k étant le nombre de classe de la variable qualitative.

J'ai ensuite plusieurs méthodes possibles pour regarder si l'appariement a équilibré la distribution des variables dans les groupes traité par CCD et non traité par CCD :

- Méthode 1 : je regarde la moyenne de chaque variable en fonction du score de propension (annexe : figure A-2). Je vois donc pour des patients de ces 2 groupes ayant la même probabilité d'être traité, si la moyenne de la variable est semblable. Bien sûr si les courbes se superposent parfaitement, on peut dire que la distribution est semblable dans les deux groupes. Si elles sont disjointes à certaines valeurs d'abscisse, nous pouvons dire que pour les individus ayant tel probabilité d'être traité, la moyenne des variables diffèrent. Ainsi dans les courbes présentées en annexe, on voit ainsi que les individus avec une forte probabilité d'être traité ont une moyenne qui semble différer entre les deux groupes pour INCOME, NINCLAS, RESP et SEPS, et que les individus avec une faible probabilité d'être traité ont une moyenne qui semble différer pour les variables RESP, HEMA et DNR1. Or il me semble qu'on ne cherche pas à ce que les individus ayant la même probabilité d'être traité soit exactement semblable 2 à 2 (même si bien sûr c'est idéal) mais plutôt que les populations des groupes traités et non traités soit homogène, comme c'est le cas lorsque l'on randomise. Il me semble donc plus pertinent de voir si globalement les distributions sont les mêmes dans les deux groupes, et les deux méthodes présentées ci-dessous répondent à cette question.

- Méthode 2 : je regarde la différence standardisée des moyennes (SMD), c'est à dire la différence entre la moyenne dans le groupe cathétérisé et la moyenne dans le groupe non cathétérisé divisé par la variance commune. J'ai séparé les schémas en variables quantitatives non binaires (figure 4) et variables binaires et qualitatives binarisées

(Figure 5) pour plus de lisibilité. En instaurant un seuil de SMD à 0.1 comme conseillé dans la littérature, je vois que l'appariement établi un équilibre entre les deux groupes pour toutes les variables ayant servi à calculer le score de propension.

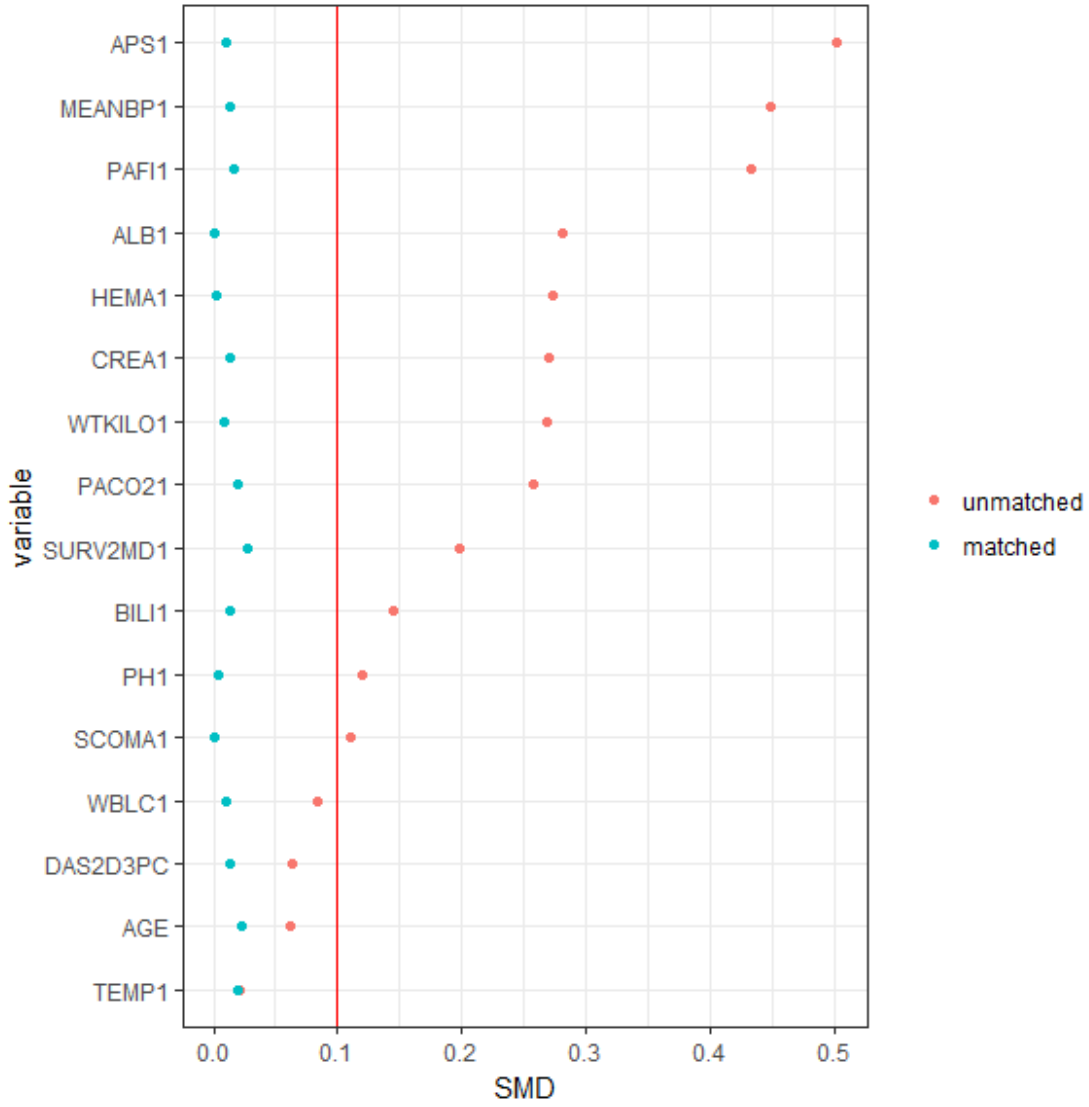


Figure 4. Différence standardisée des moyennes des variables quantitatives.

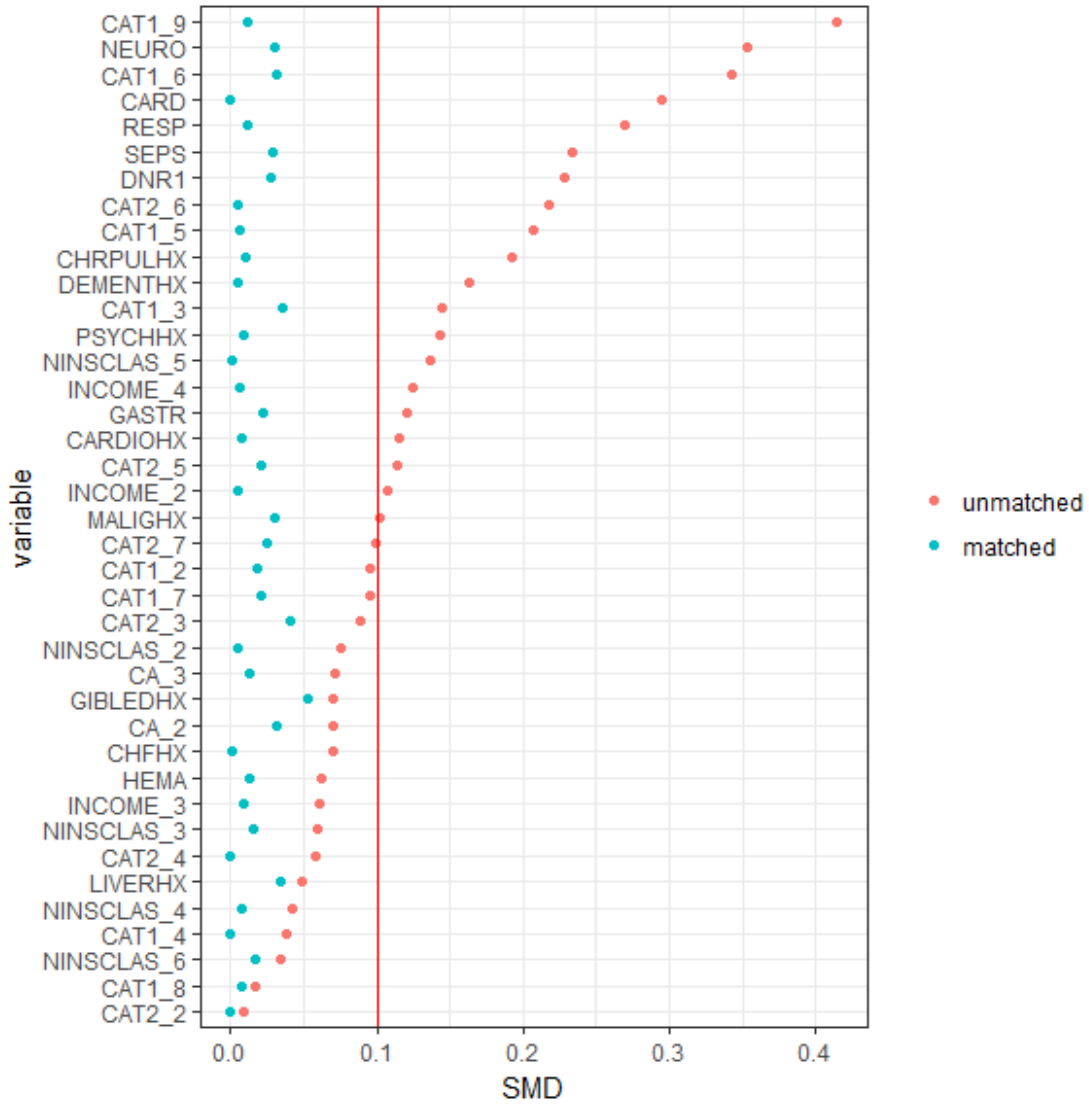


Figure 5. Différence standardisée des moyennes des variables qualitatives (binarisées pour regarder l'équilibre) et des variables binaires.

- Méthode 3 : je regarde les intervalles de confiances des différentes variables, j'ai réalisé les intervalles de confiance des moyennes pour les variables quantitatives (figure 6) et des fréquences de 1 pour les variables binaires et qualitatives binarisées (figure 7). Les variables quantitatives sont standardisées, ce qui permet une lecture plus aisée du graphique des variables quantitatives. Je ne sais pas comment réaliser un équivalent de cette standardisation avec les variables binaires et le graphique est donc moins facilement analysable. A noter que les variables pour lesquelles on n'a pas pu calculer d'intervalle de confiance pour non-respect des conditions de validité (np>=5 et n(1-p)>=5) ne sont pas présente dans le graphe. Ce sont un des niveaux de CAT1 et deux des niveaux de CAT2. On voit là aussi que globalement les variables sont plutôt bien équilibrées après appariement car les intervalles de confiance se recouvrent deux à deux.

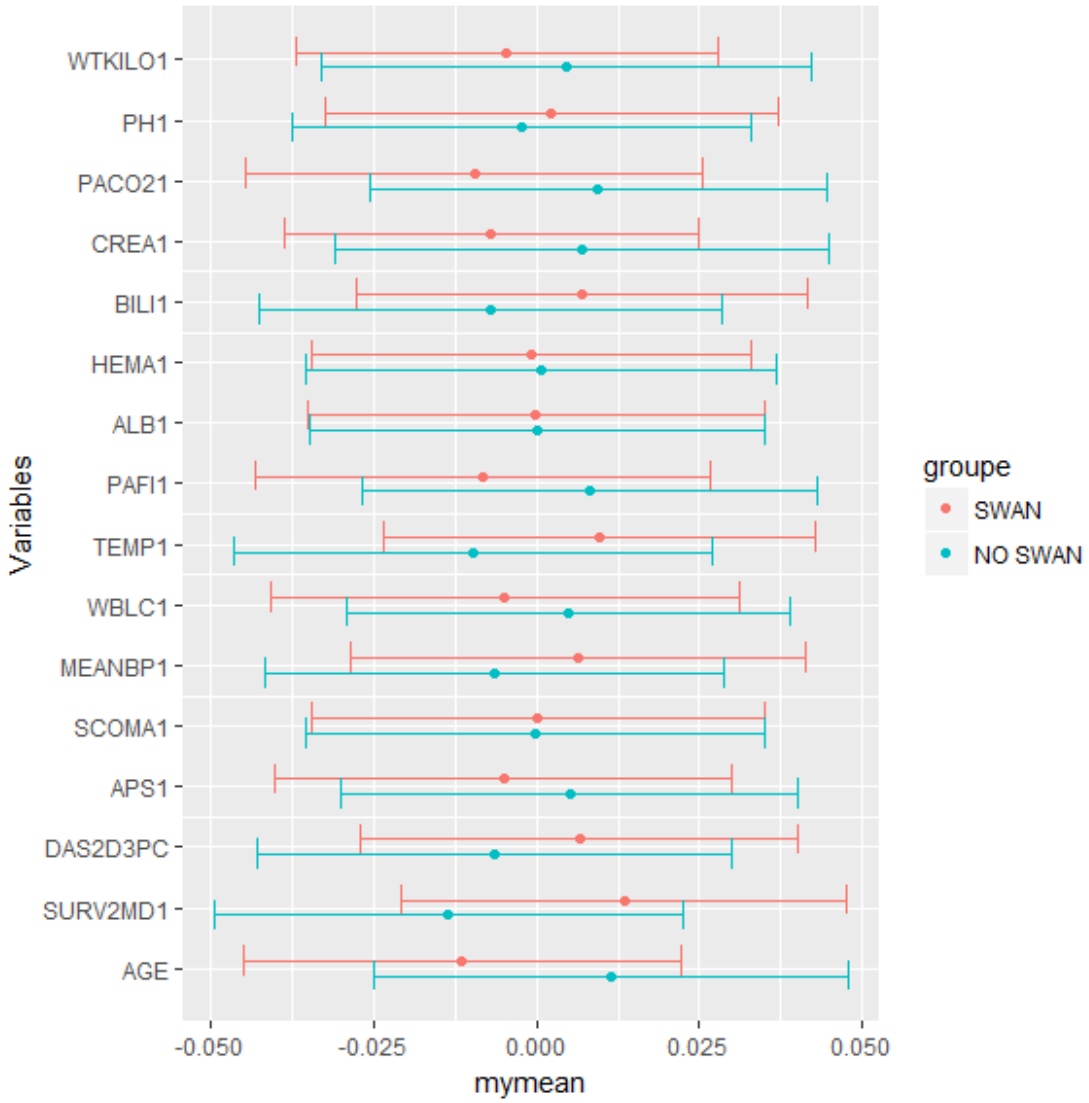


Figure 6. Intervalle de confiance des variables quantitatives utilisées pour calculer le score de propension, en fonction du traitement par CCD.

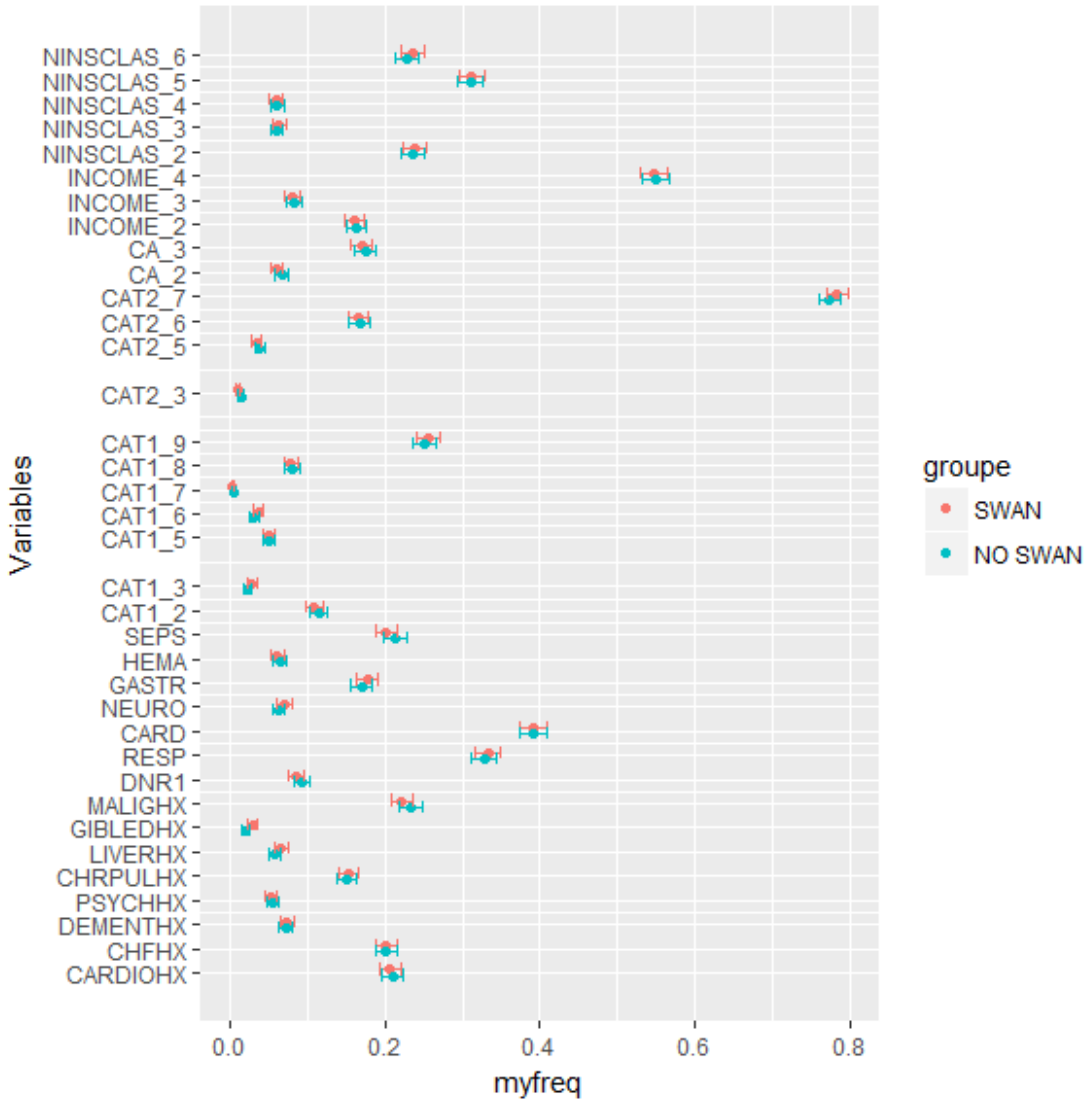


Figure 7. Intervalle de confiance des variables qualitatives et binaires utilisées pour calculer le score de propension, en fonction du traitement par CCD (les variables qualitatives ont été binarisées).

Dans le futur, je préférerai quand même l'analyse de l'équilibre par la méthode SMD qui est plus rapide à coder et plus facile à lire. C'est d'ailleurs la méthode que j'ai le plus souvent rencontrée dans les publications.

V - Analyses dans la population appariée

Pour répondre à la question "le cathétérisme cardiaque droit modifie-t-il la survie à 30 jours ?", je peux utiliser deux méthodes différentes. Soit j'utilise une régression logistique conditionnelle, soit j'utilise un modèle de Cox. Dans les deux cas, j'utilise mon échantillon apparié et je stratifie sur le numéro de paire.

1) Régression logistique conditionnelle

La variable à expliquer est la mort à 30 jours, la variable explicative est le cathétérisme cardiaque et je stratifie sur la paire.

Je n'ai pas trouvé de source indiquant explicitement quelles sont les conditions de validité de la régression logistique conditionnelle. Par défaut j'ai donc considérée que c'étaient les mêmes conditions que pour la régression logistique et elles sont ici respectées : j'ai plus de 5 à 10 évènements par variable explicative (la stratification sur la paire n'est pas une variable explicative).

Le coefficient du cathétérisme est significatif (pvalue<0.001), le cathétérisme cardiaque droit a donc un effet significatif sur le risque de décès à 30 jours. Le coefficient vaut 0.21, l'exponentiel du coefficient me permet d'obtenir l'odds ratio du risque de décès associé au cathétérisme. L'OR est de 1.24 avec un intervalle de confiance à 95% [1.10-1.40]. Le décès n'est pas un évènement rare, je ne peux donc pas interpréter l'OR comme un RR mais je peux dire que le risque de décès est augmenté lorsque le patient est cathétérisé.

2) Analyse de survie : Modèle de Cox

Une deuxième manière de réaliser le calcul est par modèle de Cox. Dans ce cas il est problématique de prendre en compte l'appariement par une stratification sur la paire car FE Harrell nous précise dans son livre Regression Modeling Strategies (2nd Ed) p.482 que si le nombre de strates est très grand par rapport au nombre total d'évènement, on perd en efficience (? "Loss of efficiency" dans le texte, je ne sais pas bien comment le traduire). Or on a ici 1569 paires ou strates pour 1055 évènements, ce qui peut être considéré comme un grand nombre de strates relativement au nombre d'évènements bien qu'aucun seuil ne soit précisé. Et de plus FE Harrell nous précise qu'une strate qui ne contient aucun évènement ne contribue pas à l'information et qu'une telle situation doit donc être évitée si possible. Or ici 44% des paires sont sans évènements et ne contribue pas à l'information avec donc j'imagine une perte de puissance. J'utiliserai donc l'option cluster(paire) plutôt que strata(paire) pour prendre en compte l'appariement dans le modèle de Cox. Cette option prend en compte le design apparié, et on regarde alors la variance robuste et le test du score robuste. Cette méthode calcule une vraisemblance globale et non pas strate par strate comme lorsque l'on stratifie, évitant donc de perdre de l'information.

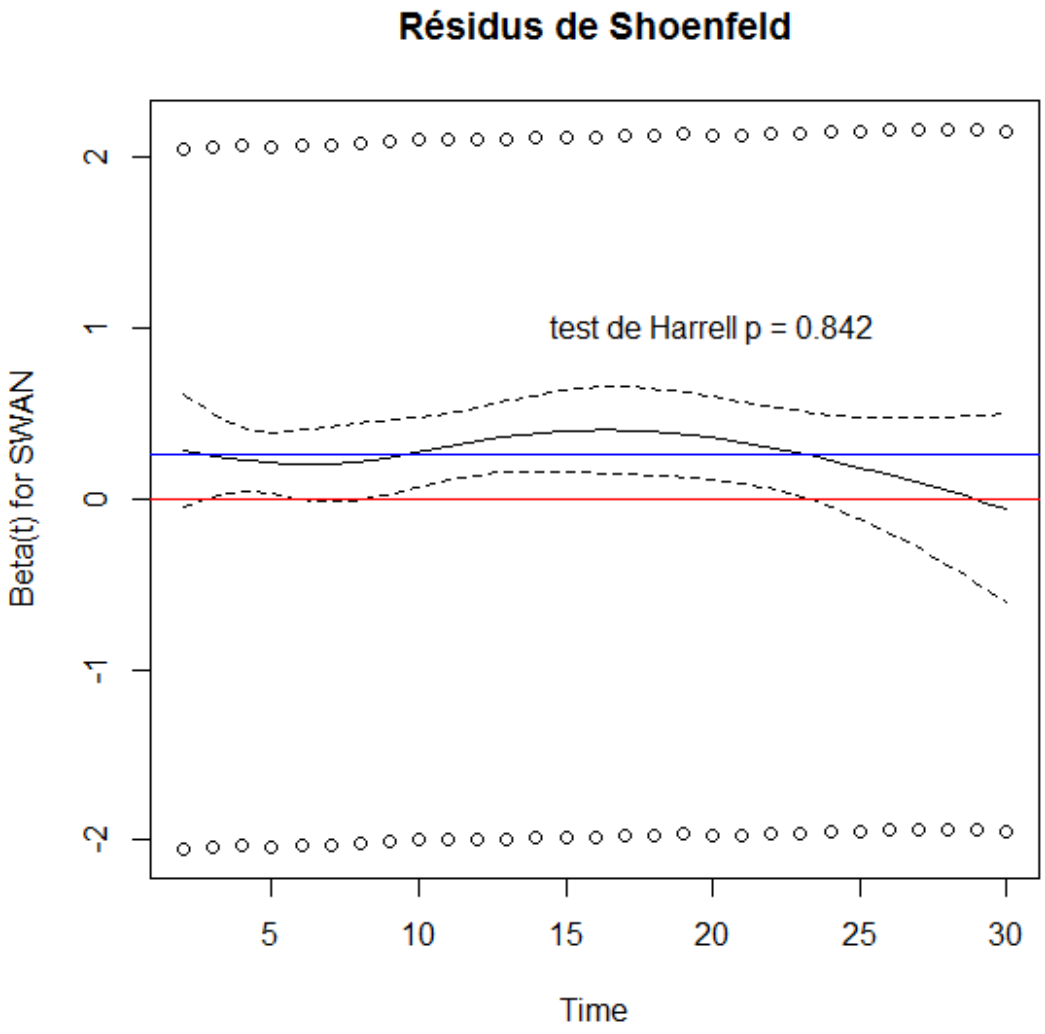


Figure 8. Résidus de Shoenfeld en fonction du temps. P value à partir du test de Harrell.

L'hypothèse des risques proportionnels n'est pas vérifiée. En effet l'intervalle de confiance des résidus de Shoenfeld n'incluent pas 0 en tous points (figure 8). Il faut donc en toute rigueur que j'ajoute une variable SWAN dépendante du temps. Cependant je n'ai pas trouvé de transformation qui convienne, aucun coefficient de paramètre dépendant du temps n'étant significatif lorsqu'on l'ajoute au modèle (les résidus de Shoenfeld des modèles ne doivent donc même pas être regardés). Voici les transformations essayées : log, racine carrée, *temps, 1/temps, racine cube, carré, cube et puissance de 0.7. J'analyse donc le modèle de Cox sans ajouter de variable dépendante du temps, mais il faudra avoir en tête que le résultat est erroné.

L'ajout d'un cluster sur la paire me permet d'avoir une variance robuste prenant en compte l'appariement. J'utilise cette variance robuste pour calculer l'intervalle de confiance à 95% du hazard ratio : 1.299 [1.297-1.302]. Je regarde le test du score robuste pour conclure : $p<0.001$. Le risque de décès est donc significativement différent selon que le patient est

cathétérisé ou non. Et c'est dans le sens d'un risque plus grand chez les patients cathétérisés avec un risque de décès à 30 jours multiplié par 1.299.

Je retrouve cette information graphiquement en traçant une courbe de survie en fonction du traitement par cathétérisme cardiaque droit.

3) Représentation graphique de la survie à 30 jours en fonction du traitement par CCD par la méthode de Kaplan-Meier

La méthode de Kaplan Meier permet de représenter graphiquement les courbes de survie (une courbe par groupe).
Condition de validité de la méthode de Kaplan Meier :
- censure indépendante de la probabilité de survenue de l'évènement
- probabilité de survie indépendante du moment de recrutement dans l'étude
- censure indépendante du groupe

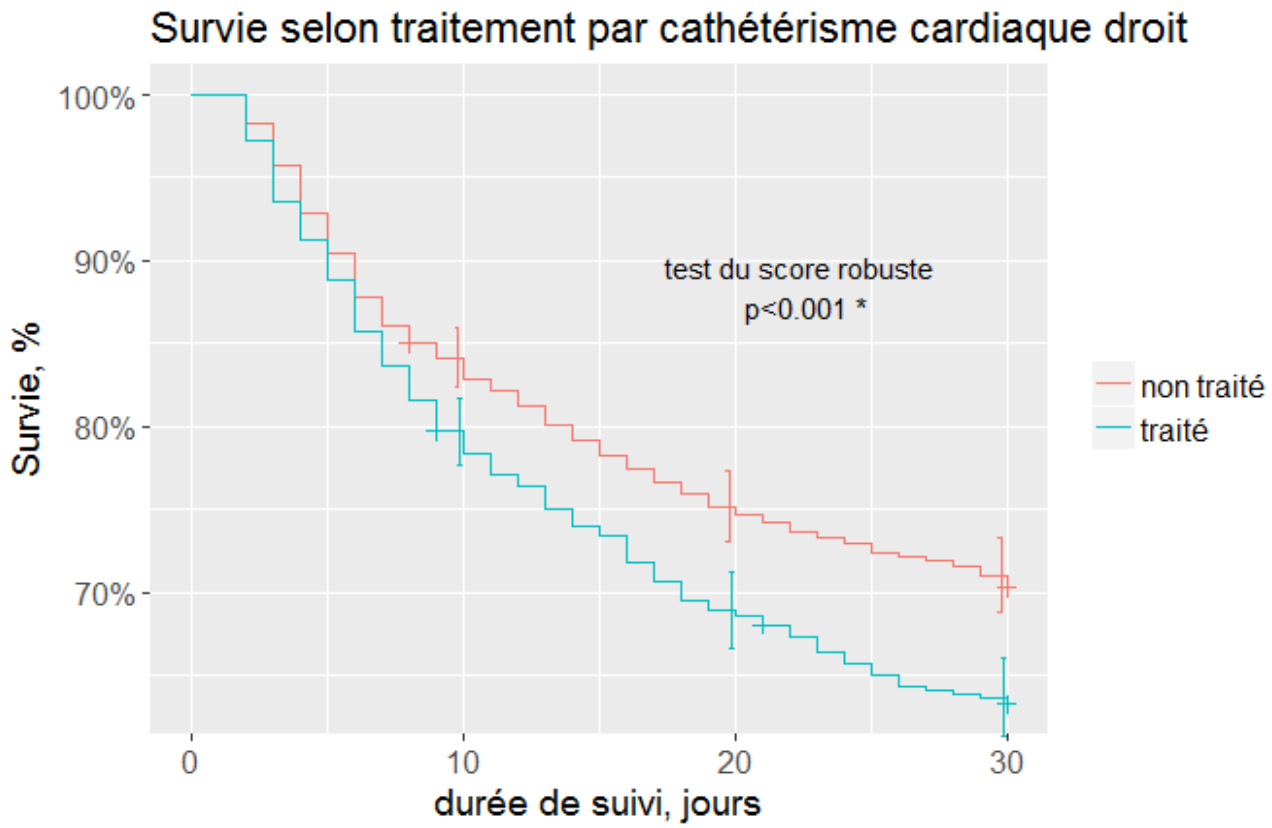


Figure 9. Courbes de survie par la méthode de Kaplan Meier de la survie à 30 jours en fonction du traitement par cathétérisme cardiaque droit. P value calculée par modèle de Cox avec cluster sur la paire.

	J0	J10	J20	J30
n.risk	1571	1321	1180	**1114**
n.event	0	269	129	**68**
surv	**100%**	**83 %**	**75 %**	**70 %**

Table 4. Table de survie des patients non traités par cathétérisme cardiaque droit

	J0	J10	J20	J30
n.risk	1571	1251	1082	**998**
n.event	0	340	154	**83**
surv	**100%**	**78 %**	**69 %**	**63 %**

Table 5. Table de survie des patients traités par cathétérisme cardiaque droit

Sur la figure 9, on observe 2 courbes de survie, une pour le groupe non traité pour CCD et une pour le groupe traité par CCD. A chaque temps, les "marches d'escalier" représentent les patients qui ont eu l'évènement, faisant diminuer le nombre de personnes à risque d'évènement (c'est à dire les non répondeurs). Les croix représentent à chaque temps la présence de censure (perdus de vue ou exclus vivant). Par exemple dans le groupe non traité, je passe de près de 85% de survie à J10 à près de 75% de survie à J20 dans le groupe non traité. Dans la table de survie (table 4) je vois que ça correspond à 126 patients ayant eu l'évènement entre J10 et J20. Pour le groupe traité, je lis sur la courbe que je passe de 80% environ de survie à J10 à 70% environ à J20. Dans la table de survie (table 5) cela correspond à 154 patients ayant eu l'évènement. La courbe de Kaplan Meier nous permet d'observer graphiquement que la survie est meilleure lorsque les patients ne sont pas traités par cathétérisme cardiaque droit.

CONCLUSION

Nous voulions savoir si le cathétérisme cardiaque droit(CCD) améliorait la survie à 30 jours des patients admis en réanimation. Les données étaient observationnelles, sans randomisation de l'intervention. On ne pouvait donc pas exclure la présence de biais de confusion et d'attrition, qui ont d'ailleurs été mis en évidence dans le devoir : les patients traités par CCD étaient dans un état plus grave. En l'état aucune analyse ne pouvait être faite car elle aurait été biaisée. Afin de tenir compte des caractéristiques de bases nous avons donc réalisé un score de propension et apparié sur ce score. Ainsi chaque individus traité et non traité appariés avaient la même probabilité d'être traités. Après appariement et en tenant compte de cet appariement dans l'analyse, nous mettons en évidence une association entre le cathétérisme cardiaque droit par sonde de Swan Ganz et le risque de mortalité à 30 jours : le cathétérisme cardiaque droit est associé à une augmentation du risque de mortalité. Mais bien que l'utilisation du score de propension augmente le niveau de causalité, on ne peut pas conclure avec le même niveau de causalité qu'un essai randomisé.

REFERENCES

Connors, A. F., T. Speroff, N. V. Dawson, C. Thomas, F. E. Harrell, D. Wagner, N. Desbiens, et al. “The Effectiveness of Right Heart Catheterization in the Initial Care of Critically Ill Patients. SUPPORT Investigators.” JAMA 276, no. 11 (September 18, 1996): 889–97.

Sterne, J. A C, I. R White, J. B Carlin, M. Spratt, P. Royston, M. G Kenward, A. M Wood, and J. R Carpenter. “Multiple Imputation for Missing Data in Epidemiological and Clinical Research: Potential and Pitfalls.” BMJ 338, no. jun29 1 (September 1, 2009): b2393–b2393. doi:10.1136/bmj.b2393.

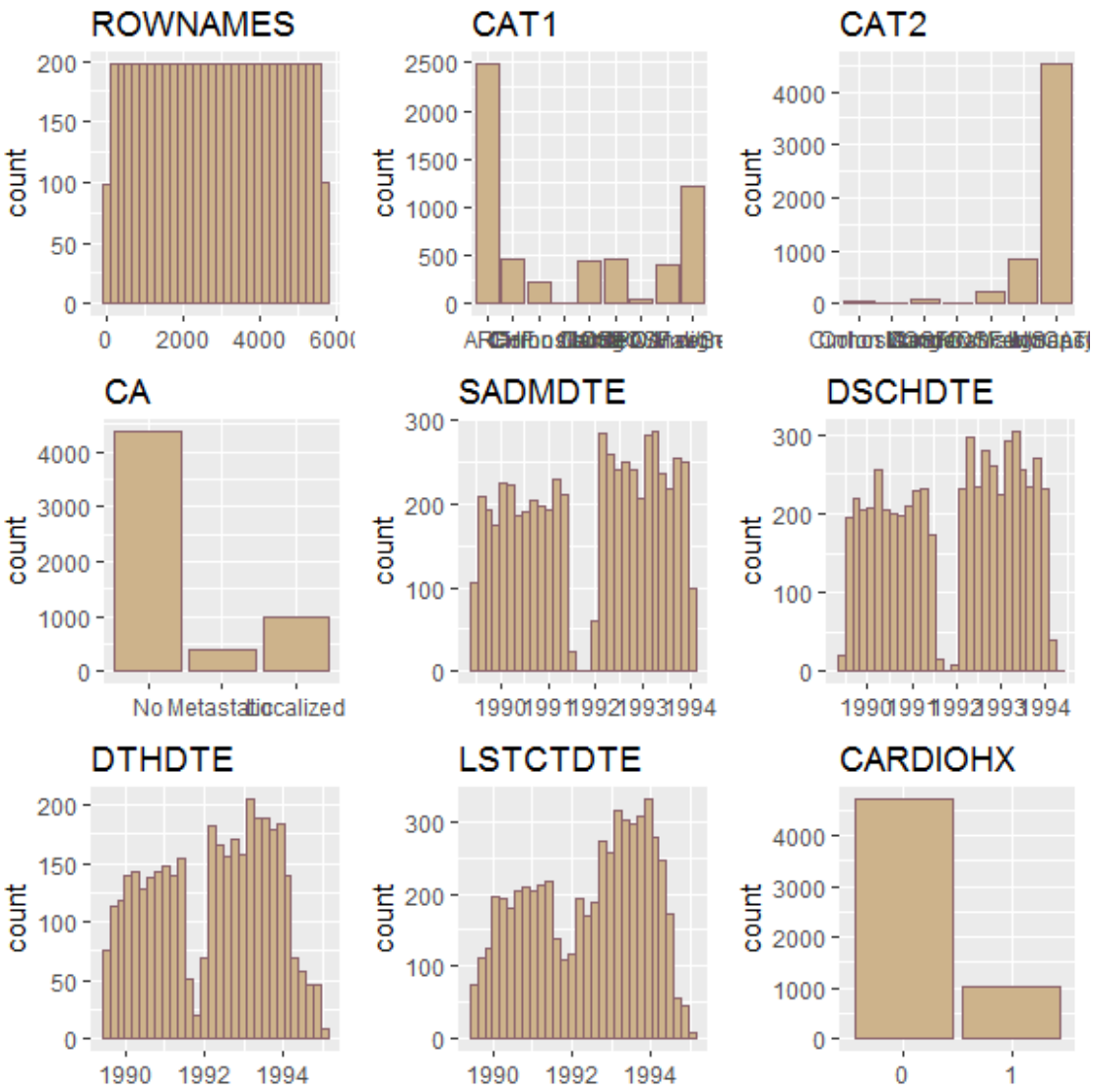
Cours du Dr David Hajage "Evaluation de l’effet d’un traitement en condition réelle d’utilisation (scores de propension et scores pronostiques)", janvier 2017.

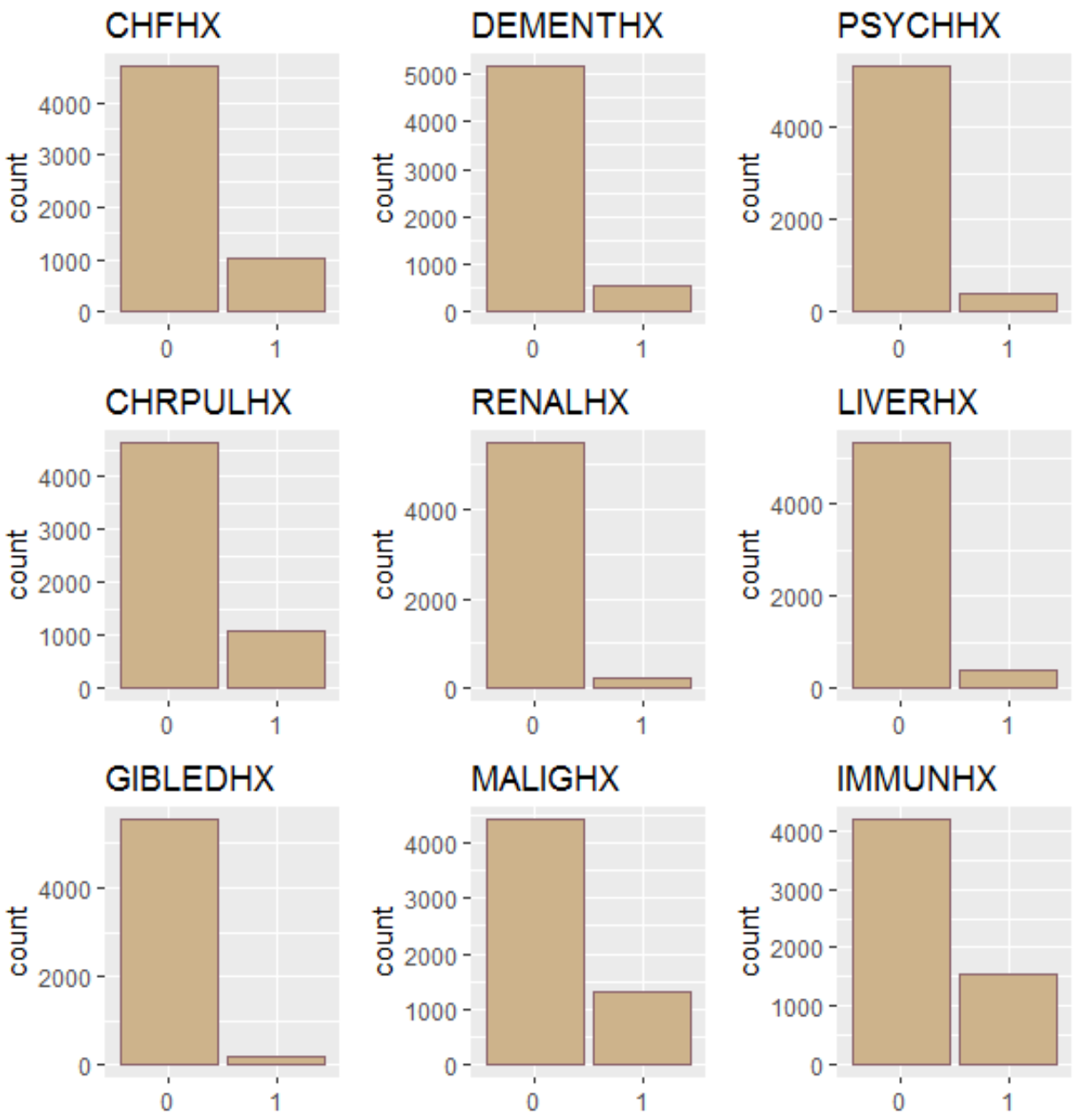
Austin, Peter C. “An Introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies.” Multivariate Behavioral Research 46, no. 3 (May 31, 2011): 399–424. doi:10.1080/00273171.2011.568786.

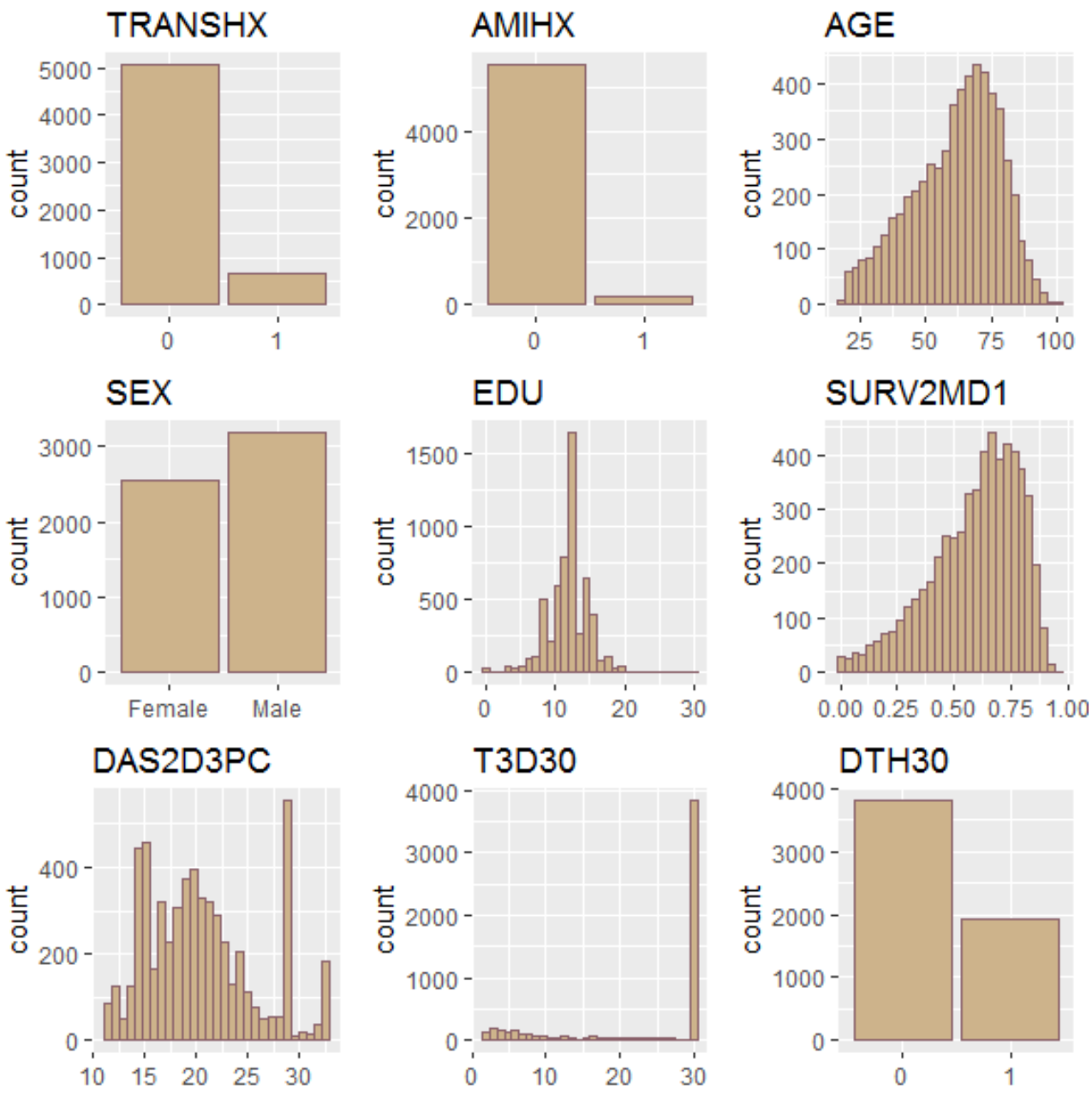
Ho, Daniel E., Kosuke Imai, Gary King, and Elizabeth A. Stuart. “MatchIt: Nonparametric Preprocessing for Parametric Causal Inference.” Journal of Statistical Software 42, no. 8 (2011). doi:10.18637/jss.v042.i08.

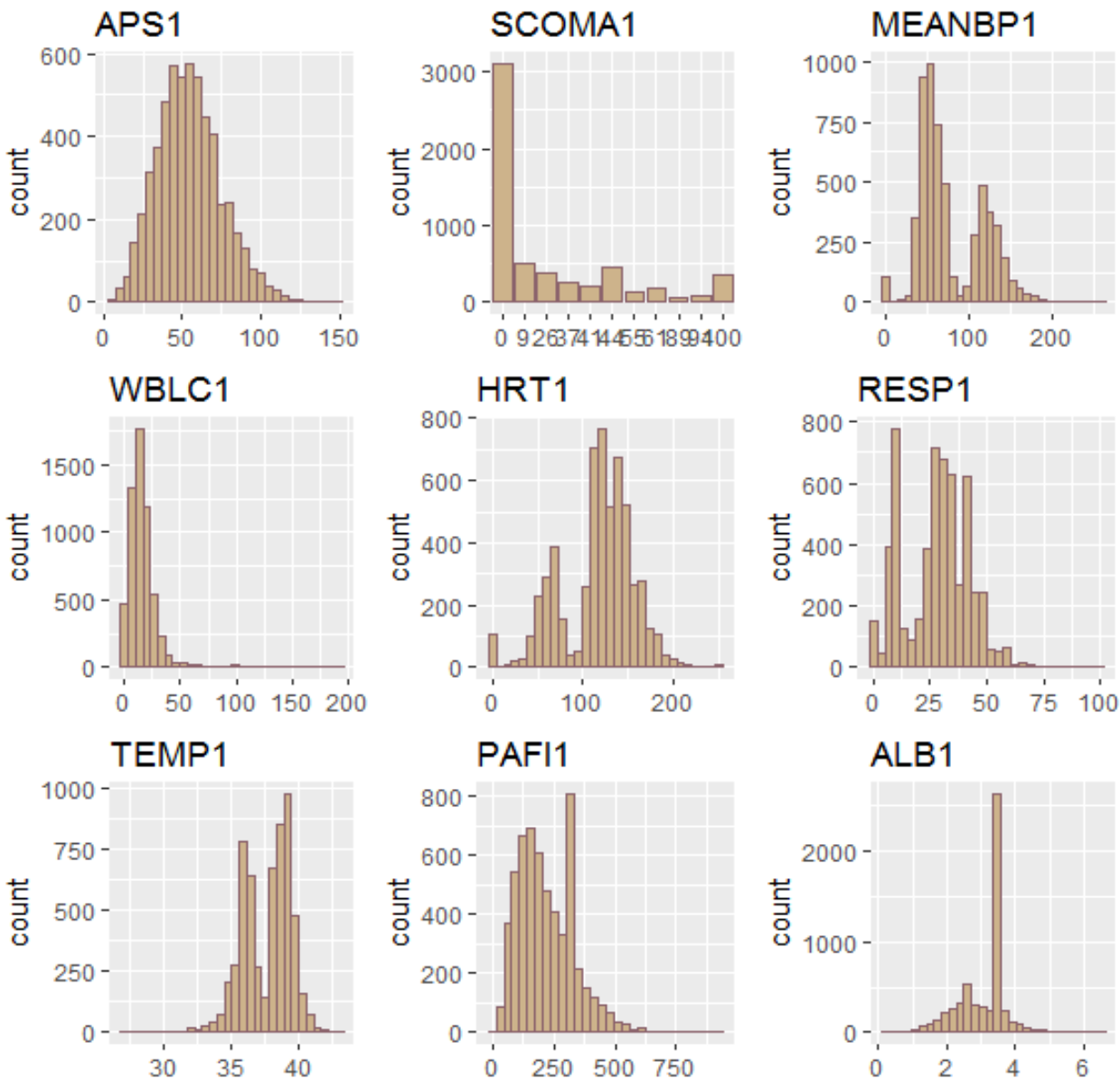
ANNEXES

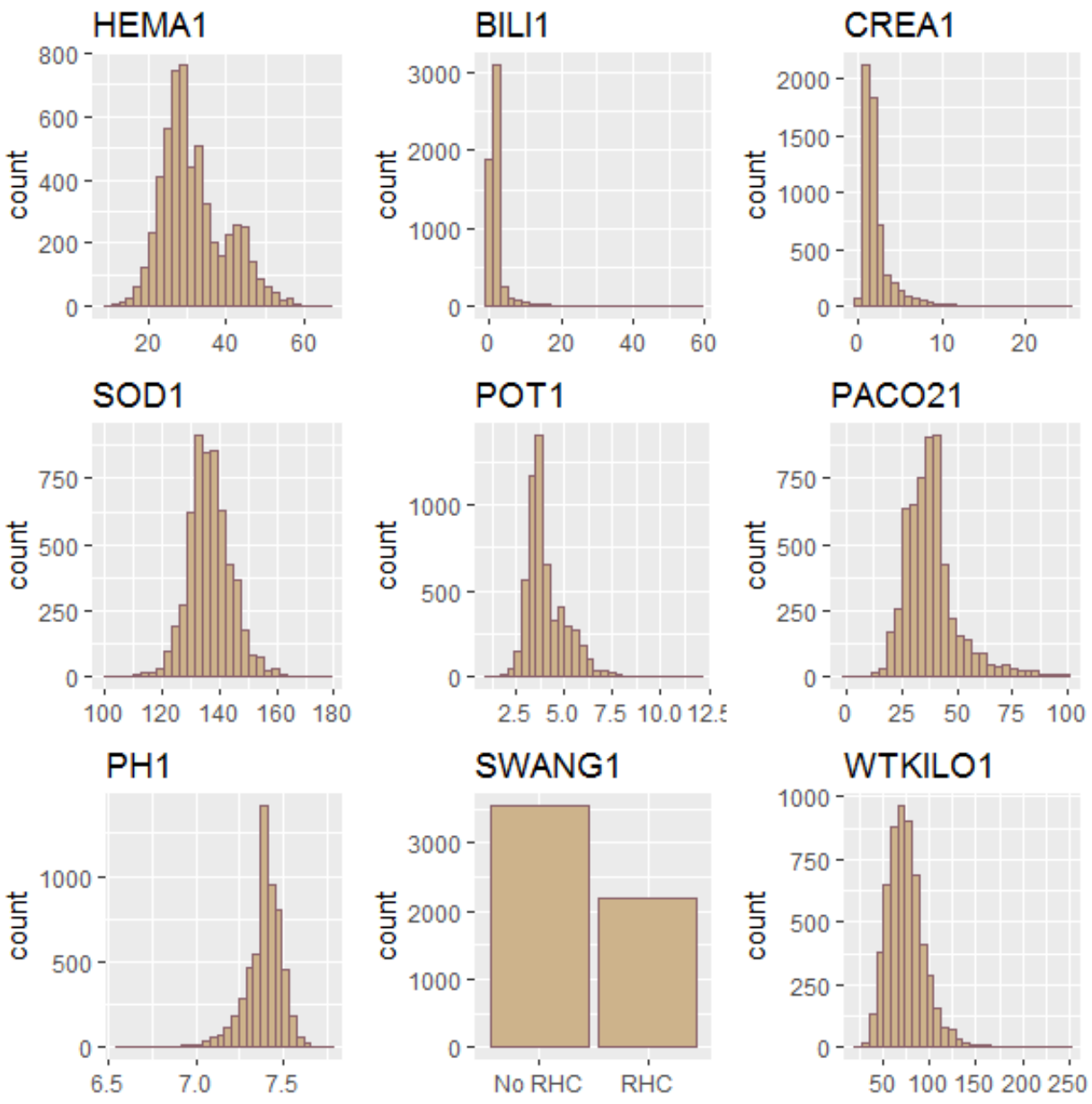
Figure A-1. Distribution des variables avant modification et imputation

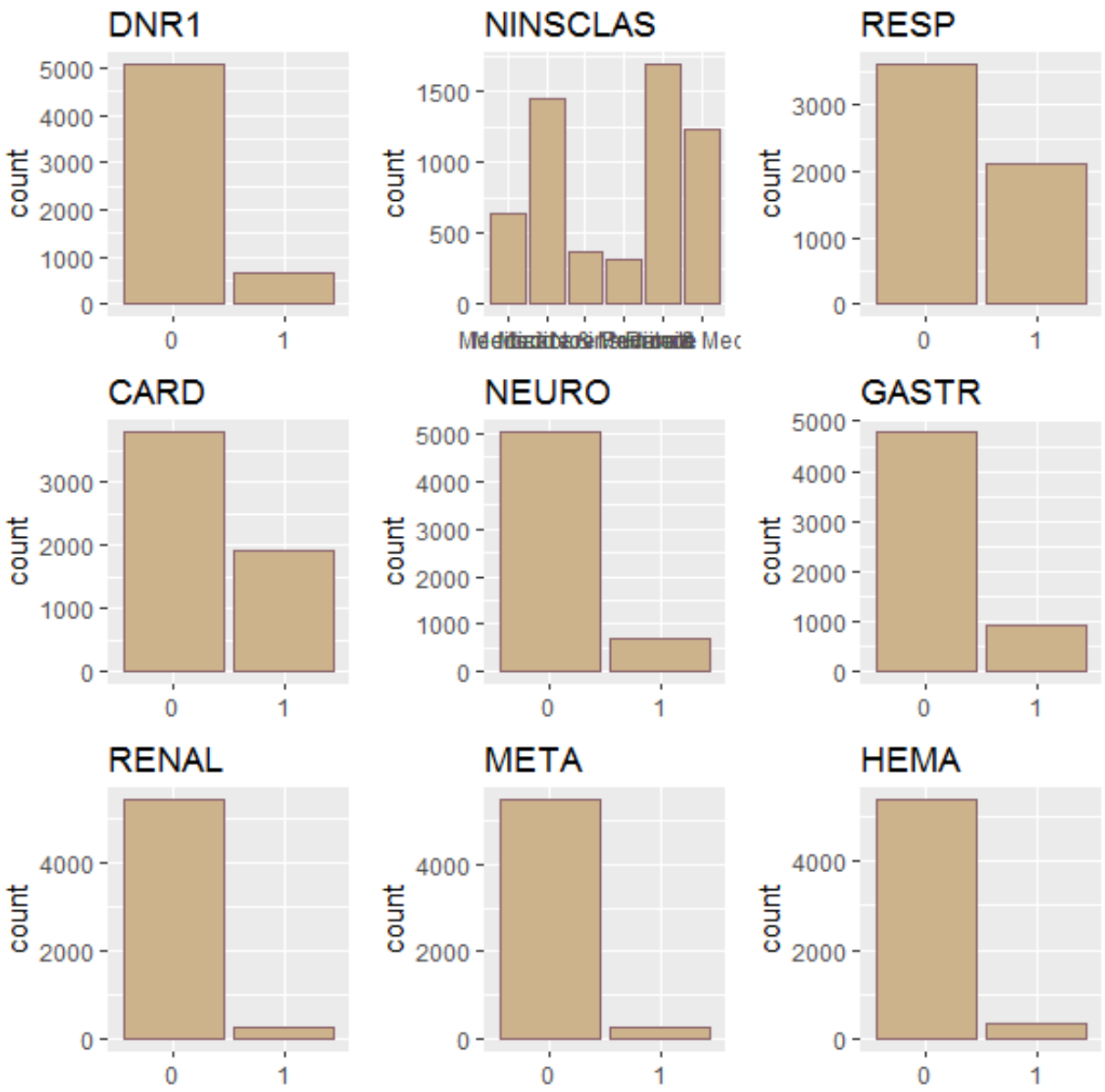

ROWNAMES
CAT1
CAT2
CA
SADMDTE
DSCHDTE
DTHDTE
LSTCTDTE
CARDIOHX
count
No Metastatic
Localized
1990
1992
1994
0
1


CHFHX
DEMENTHX
PSYCHHX
CHRPULHX
RENALHX
LIVERHX
GIBLEDHX
MALIGHX
IMMUNHX
count
0
1000
2000
3000
4000
5000
0
1


TRANSHX
count
5000
4000
3000
2000
1000
0
0
1
AMIHX
count
4000
2000
0
0
1
AGE
count
400
300
200
100
0
25
50
75
100
SEX
count
3000
2000
1000
0
Female
Male
EDU
count
1500
1000
500
0
0
10
20
30
SURV2MD1
count
400
300
200
100
0
0.00
0.25
0.50
0.75
1.00
DAS2D3PC
count
400
200
0
10
15
20
25
30
T3D30
count
4000
3000
2000
1000
0
0
10
20
30
DTH30
count
4000
3000
2000
1000
0
0
1


APS1
SCOMA1
MEANBP1
WBLC1
HRT1
RESP1
TEMP1
PAFI1
ALB1
count


HEMA1
BILI1
CREA1
SOD1
POT1
PACO21
PH1
SWANG1
WTKILO1
count
No RHC
RHC


DNR1
count
5000
4000
3000
2000
1000
0
0
1
NINSCLAS
count
1500
1000
500
0
RESP
count
3000
2000
1000
0
0
1
CARD
count
3000
2000
1000
0
0
1
NEURO
count
5000
4000
3000
2000
1000
0
0
1
GASTR
count
5000
4000
3000
2000
1000
0
0
1
RENAL
count
4000
2000
0
0
1
META
count
4000
2000
0
0
1
HEMA
count
4000
2000
0
0
1

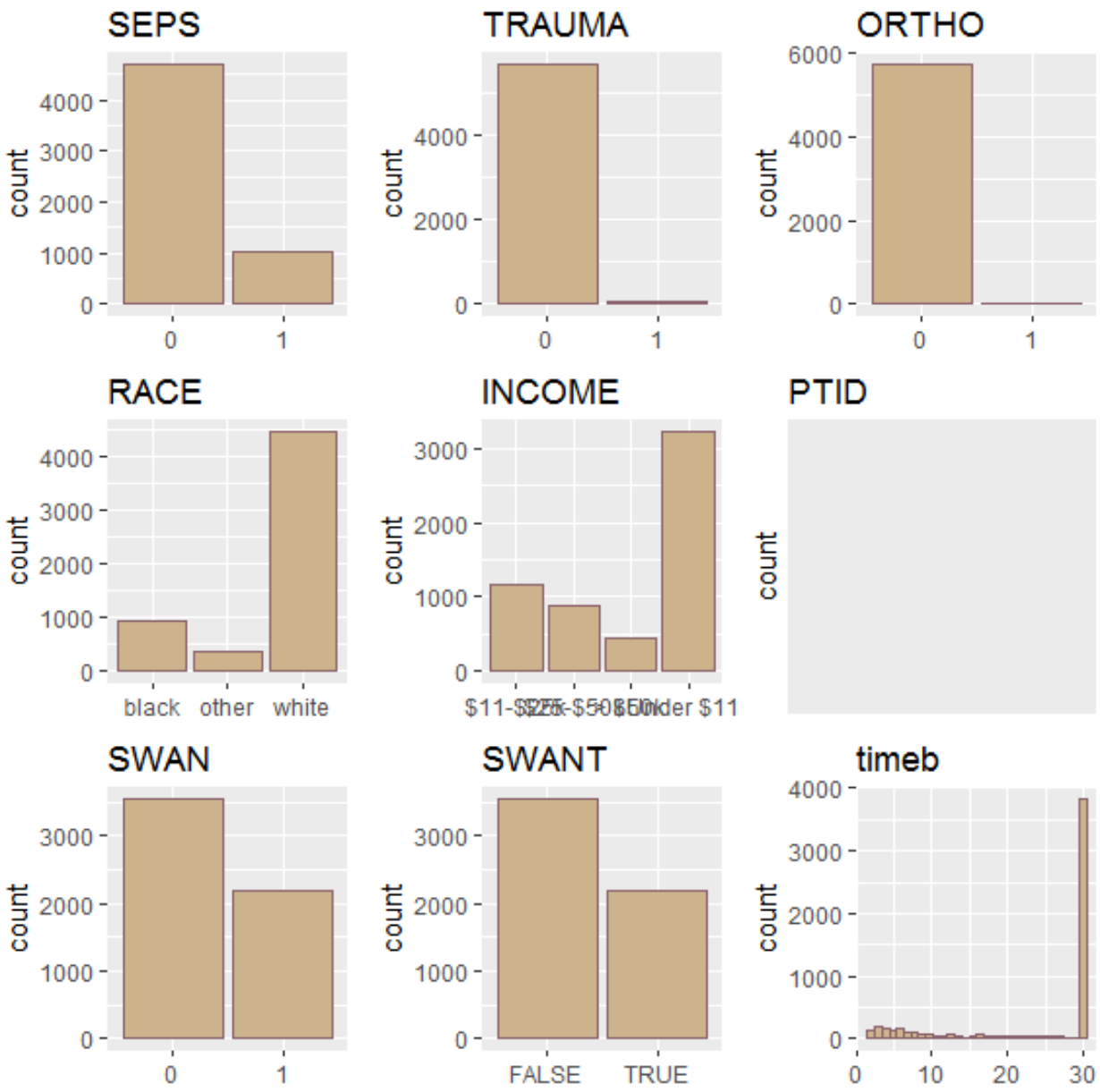
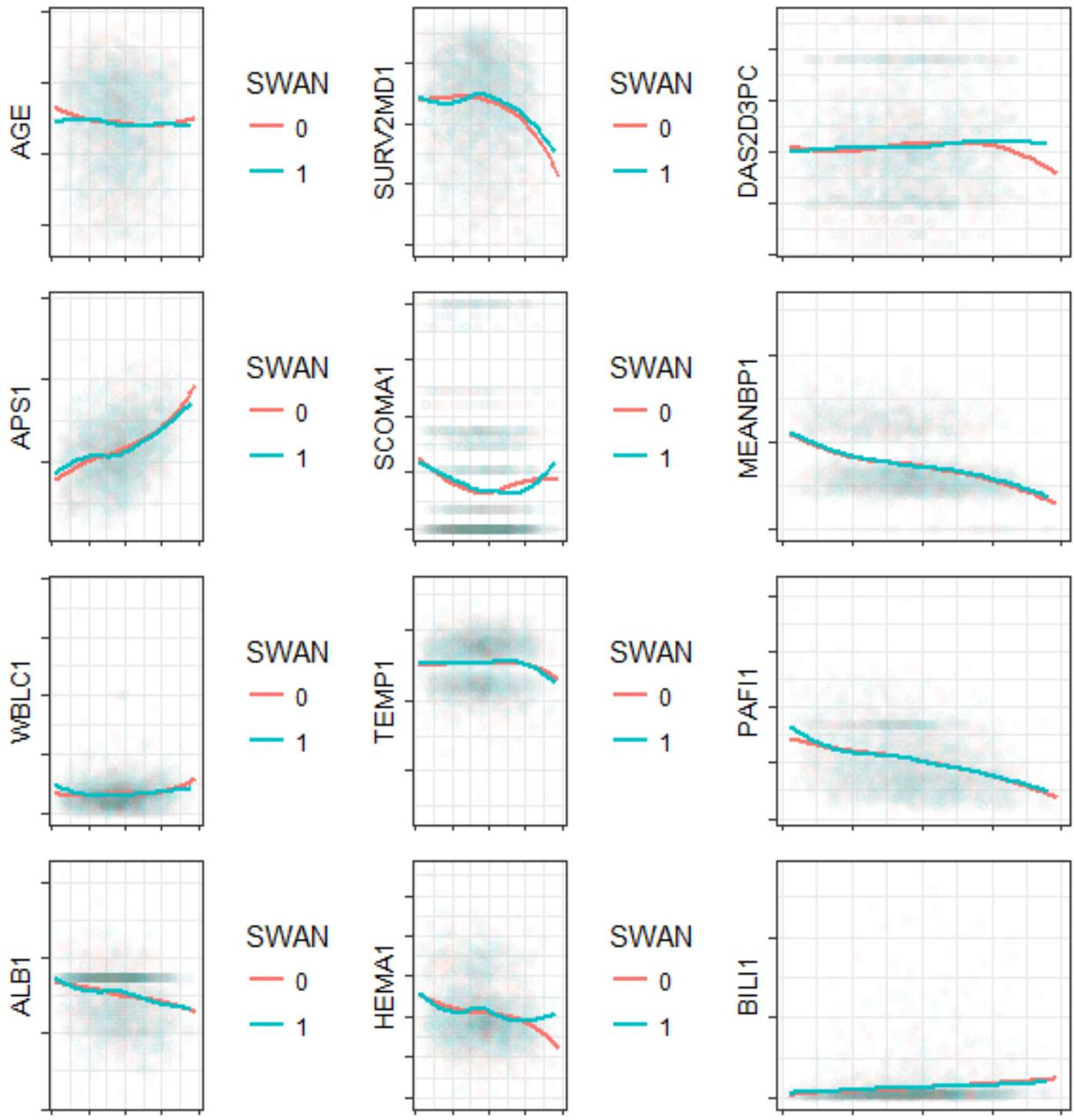
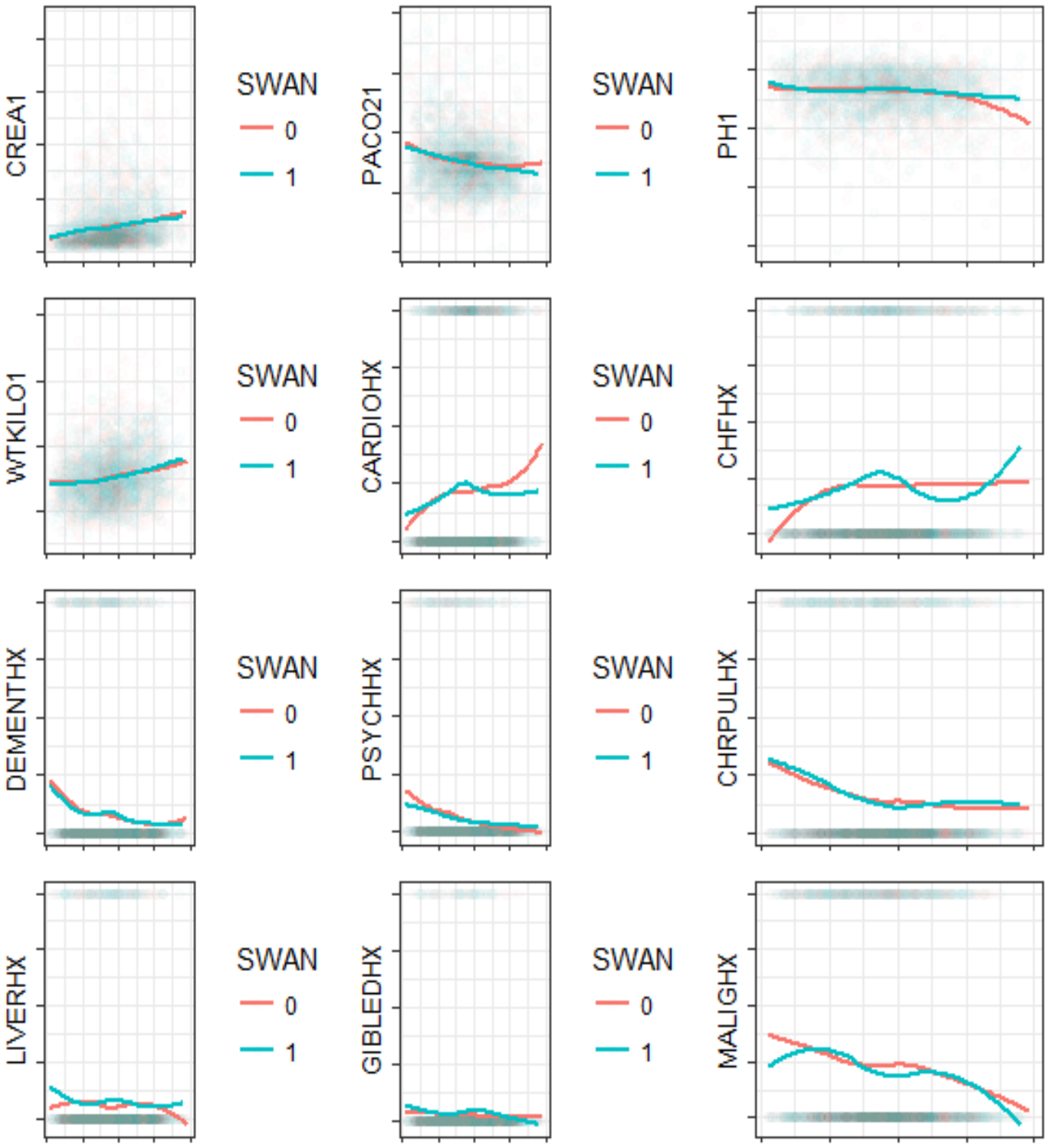
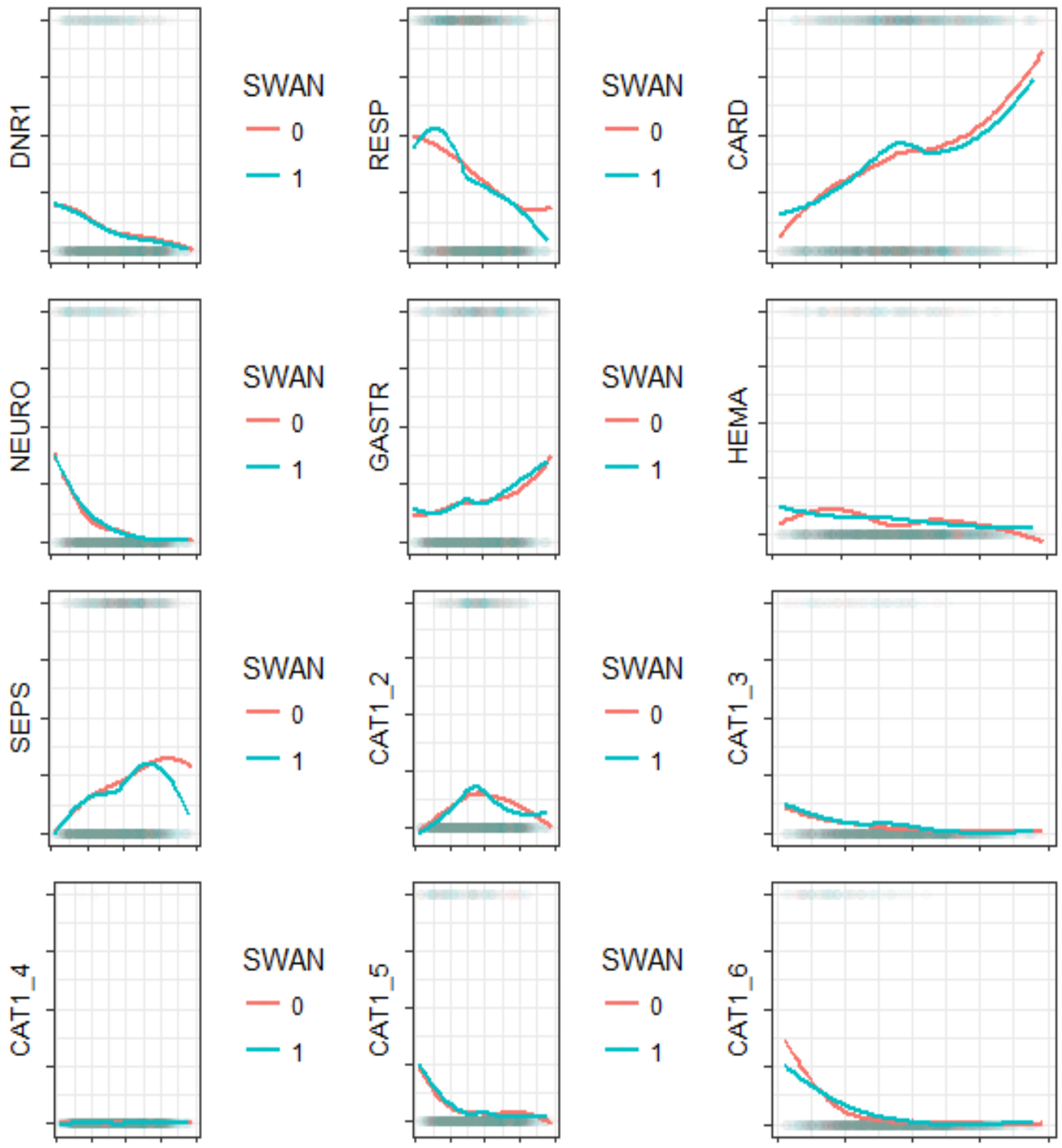
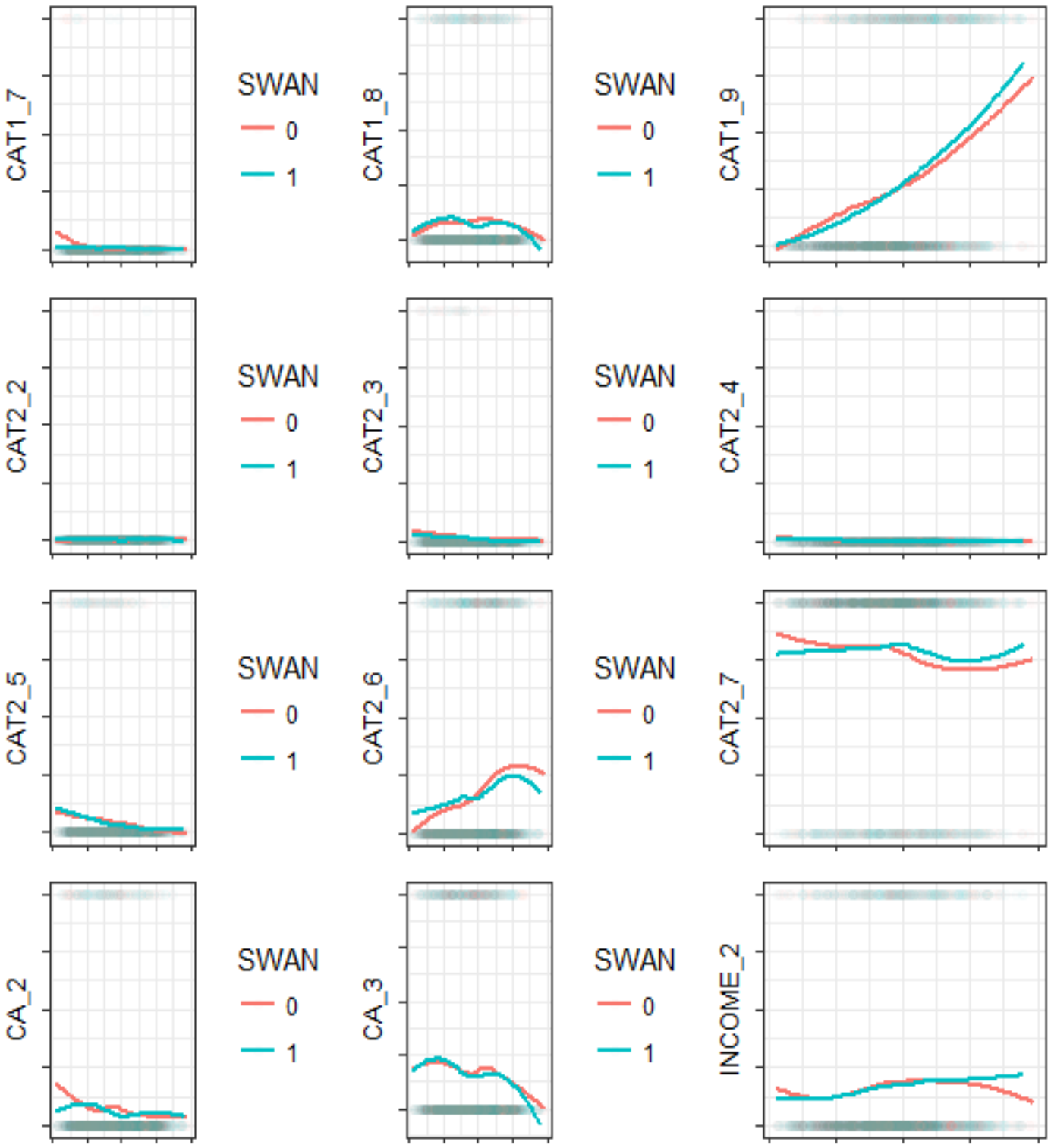


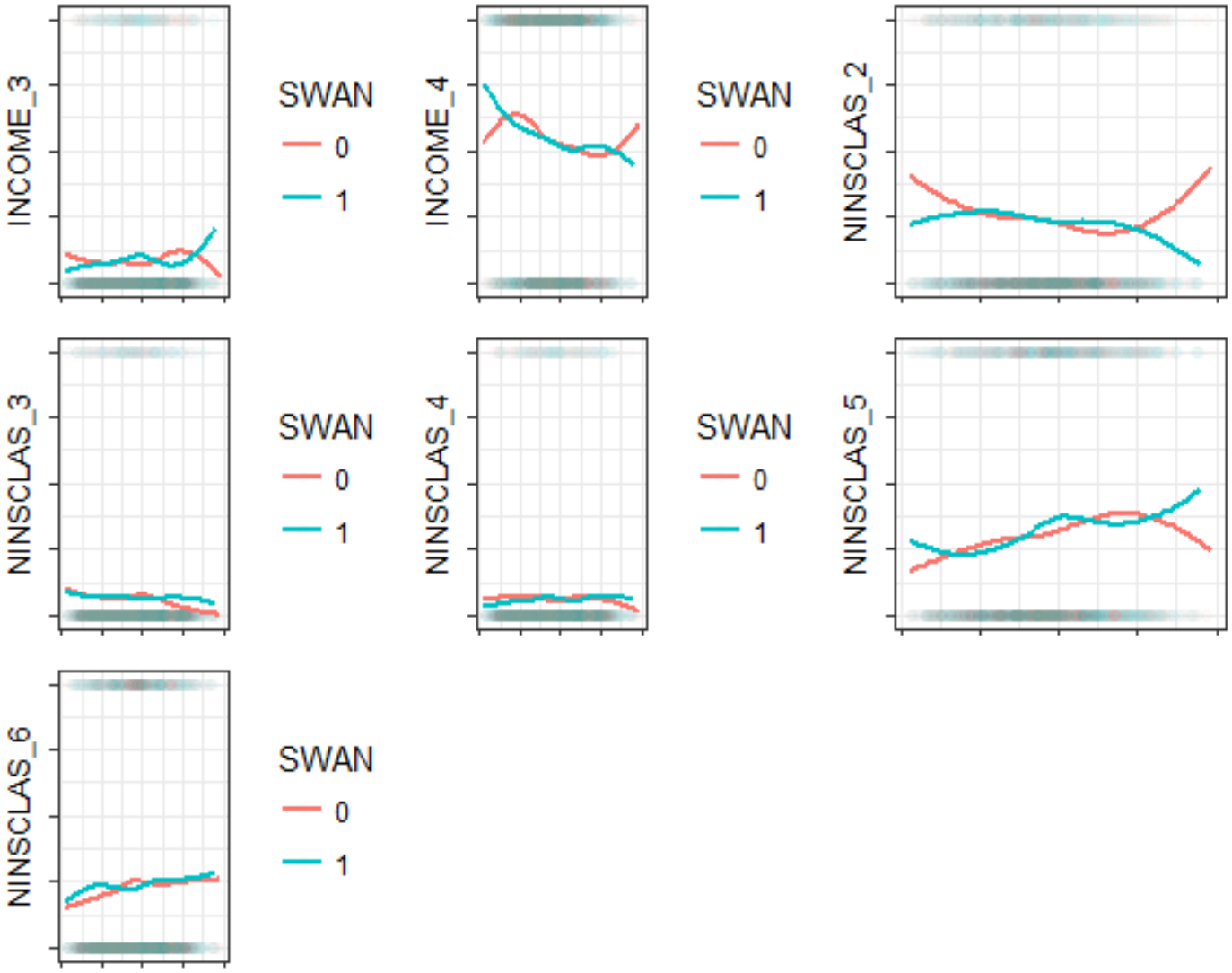
Figure A-2. Equilibre des variables après appariement : moyenne pour chaque variable en fonction du score de propension


AGE
SWAN
0
1
SURV2MD1
SWAN
0
1
DAS2D3PC
APS1
SWAN
0
1
SCOMA1
SWAN
0
1
MEANBP1
WBLC1
SWAN
0
1
TEMP1
SWAN
0
1
PAFI1
ALB1
SWAN
0
1
HEMA1
SWAN
0
1
BILI1


CREA1
SWAN
0
1
PACO21
SWAN
0
1
PH1
WTKILO1
SWAN
0
1
CARDIOHX
SWAN
0
1
CHFHX
DEMENTHX
SWAN
0
1
PSYCHHX
SWAN
0
1
CHRPULHX
LIVERHX
SWAN
0
1
GIBLEDHX
SWAN
0
1
MALIGHX


DNR1
SWAN
0
1
RESP
SWAN
0
1
CARD
NEURO
SWAN
0
1
GASTR
SWAN
0
1
HEMA
SEPS
SWAN
0
1
CAT1_2
SWAN
0
1
CAT1_3
CAT1_4
SWAN
0
1
CAT1_5
SWAN
0
1
CAT1_6


CAT1_7
SWAN
0
1
CAT1_8
SWAN
0
1
CAT1_9
CAT2_2
SWAN
0
1
CAT2_3
SWAN
0
1
CAT2_4
CAT2_5
SWAN
0
1
CAT2_6
SWAN
0
1
CAT2_7
CA_2
SWAN
0
1
CA_3
SWAN
0
1
INCOME_2


INCOME_3
SWAN
0
1
INCOME_4
SWAN
0
1
NINSCLAS_2
NINSCLAS_3
SWAN
0
1
NINSCLAS_4
SWAN
0
1
NINSCLAS_5
NINSCLAS_6
SWAN
0
1

---
title: 'DEVOIR EPIDEMIOLOGIE : UTILISATION DU SCORE DE PROPENSION'
author: "Sarah F FELDMAN"
date: "27 février 2017"
output:
  word_document: default
  pdf_document: default
  html_document: default

```{r setup, include=FALSE}
source("01-library_EPID.R")
source("02-functions_EPID.R")
knitr::opts_chunk$set(echo = TRUE)
opts_chunk$set(echo=FALSE, comment=NA, warning= FALSE, fig.width=6, fig.height=6)
#opts_chunk$set(fig.width=6, fig.height=6)
```

INTRODUCTION

Le cathétérisme cardiaque droit (CCD) est un examen invasif utilisé en soins intensif pour mesurer directement la fonction cardiaque, ce qui permettrait selon certains médecins une meilleure prise en charge du patient et donc une augmentation de la survie. Cependant pour montrer une telle causalité il faudrait un essai randomisé. Or nous avons ici les données d'une cohorte prospective, non interventionnelle, de patients hostpitalisés en réanimation, certains ayant été cathétérisés, d'autres non. La cathétérisation n'a pas été randomisée, il y a donc lieu de penser que l'examen a été réalisé préférentiellement chez certains types de patients, potentiellement les patients les plus graves qui ont donc plus de risque de décéder. Cela constitue un biais d'indication qui rend l'analyse de l'efficacité du CCD ininterprêtable. Pour pouvoir analyser l'efficacité du CCD par sonde de swan Ganz, nous allons prendre en compte les biais d'indication propables à l'aide d'un score de propension, nous permettant d'améliorer le niveau de causalité de la relation CCD/Décès si elle existe.

I - ETAPES PRELIMINAIRES

1) Description de la base de donnees

```{r}
.dir <- dirname(getwd())

d <- read.csv2(paste0(.dir,"/data/rhc_devoir_epidemio.csv"))

#dim(d)
#length(unique(d$PTID)) #Pas de doublons
```
La base de données comporte les informations de 5735 patients pour 63 variables. Ce sont des données transversales avec une ligne par patient (pas de doublon), chaque variable ayant été mesurées une seule fois.

2) Data management
```{r}
#data management

d$SADMDTE <- as.Date(d$SADMDTE,"%d/%m/%Y")
d$DSCHDTE <- as.Date(d$DSCHDTE,"%d/%m/%Y")
d$DTHDTE <- as.Date(d$DTHDTE,"%d/%m/%Y")
d$LSTCTDTE <- as.Date(d$LSTCTDTE,"%d/%m/%Y")

for (i in c("DEATH", "DTH30", "DNR1", "RESP", "CARD", "NEURO", "GASTR", "RENAL", "META", "HEMA", "SEPS", "TRAUMA", "ORTHO", "T3D30")) {
  d[ ,i] <- as.character(d[,i])
  d[,i] <- ifelse (d[ ,i]=="No", 0, d[ ,i])
  d[,i] <- ifelse (d[ ,i]=="Yes", 1, d[ ,i])
  d[,i] <- as.numeric(d[,i])
}

#Pour transformer les "" en NA
#plus long que apply mais garde la bonne structure
#Je repère les variables avec ""
for (x in colnames(d)){
  if(is.factor(d[ ,x])) {
    if(length(d[d[ ,x]=="", x])!=0) {
      #print(x)
    }
  }
}
#CAT2

#Pour CAT2 je crée à la place une cathégorie "NoCAT", pour DSCHDTE et DTHDTE je transforme "" en NA
d$CAT2 <- as.character(d$CAT2)
d$CAT2 <- ifelse (d$CAT2=="", "NoCAT2", d$CAT2)
d$CAT2 <- as.factor(d$CAT2)

#Je préfère que PTID soit un caractère pour éviter un gag si le numéro de ligne n'est pas le même que PTID
d$PTID <- paste0("A", d$PTID)

#variables cathétérisme:
#Variable SWAN Ganz en en 0/1 pour matchit
d$SWAN <- d$SWANG1
levels(d$SWAN) <- c(0, 1)
#J'en fais une autre en T/F pour le package Matching
d$SWANT <- ifelse(d$SWAN==1, T, F)

#Outcomes
d$DEATH <- as.numeric(as.character(d$DEATH))
d$DTH30 <- as.numeric(as.character(d$DTH30))

# d$DEATH2 <- ifelse(!is.na(d$DTHDTE), 1, 0)
# table(d$DEATH==d$DEATH2) #5735 true 0 false => la variable DEATH est correcte


#Je renomme les niveaux de CA,
levels(d$CA) <- c("Metastatic", "No", "Localized")
d$CA <- relevel(d$CA, ref = "No")
```

Je fais une première étape de data management "basique" pour mettre les variables au bon format (dates, numérique et facteur), m'assurer que le numero de patient est en caractère, créer une variable SWAN en 0/1 et une autre en TRUE/FALSE selon les différents modèles utilisés par la suite. Je change également les "" en "no category"" pour la variable CAT2 ou en NA pour les dates. Je choisis No comme référence pour la variable CA.

3) Vérification de la durée de suivi

```{r, include = FALSE}
#Duree de suivi :

#verification de la variable date de dernieres nouvelles (LSTCTDTE)
#table(is.na(d$LSTCTDTE)) #pas de perdu de vu pour la date de dernier contact

#3 patients pour lesquels la date de décès est antérieure à la date de derniere nouvelle
#table(d$DTHDTE < d$LSTCTDTE)

#2151 pour lesquels date de dernières nouvelles est antérieure au décès.
#table(d$DTHDTE > d$LSTCTDTE)

#=> Je modifie LSTCTDTE : si décès, alors la date de dernière nouvelle est la date de décès.
d$LSTCTDTE_bck <- d$LSTCTDTE # back up pour comparer ensuite.
d$LSTCTDTE <- as_date(ifelse(is.na(d$DTHDTE), d$LSTCTDTE, d$DTHDTE))

#duree de suivi en j :  date des dernières nouvelles - date d'admission
d$time <- as.numeric(d$LSTCTDTE - d$SADMDTE)

#je recree la variable duree de suivi si l'outcome est deces a 30 jours :
d$timeb <- ifelse(d$time>30, 30, d$time)


#je compare cette vatiable a celle deja existante :
#table(d$T3D30==d$timeb)
d[d$T3D30 != d$timeb, c("timeb", "time", "T3D30", "DEATH", "DTH30")]
#      timeb time T3D30 DEATH DTH30
# 453       4    4    30     0     0
# 488       6    6    30     0     0
# 2338     21   21    30     0     0
# 2811     10   10    30     0     0
# 2907      2    2    30     0     0
# 4783      9    9    30     0     0
# 5123     28   28    30     0     0
# 5437      8    8    30     0     0

#8 patients ont une duree de suivi discordante.

#Cela ne vient d'une modification de la date des dernières nouvelles
d[d$T3D30 != d$timeb, c("timeb", "time", "T3D30", "DEATH", "DTH30", "LSTCTDTE", "LSTCTDTE_bck")]

#Je prendrai la variable T3D30 pour la suite

#Je supprime la variable LSTCTDTE_bck
d$LSTCTDTE_bck <- NULL
```

Je vérifie les variables dates de décès, date de dernieres nouvelles, et durée de suivi pour l'outcome décès à 30 jours :

- 3 patients pour lesquels la date de décès est antérieure à la date de derniere nouvelle et 2151 pour lesquels date de dernières nouvelles est antérieure au décès. Je modifie la variable date de dernières nouvelles ; si la datee de décès est non nulle, alors la date de dernières nouvelles est la date de décès.

- Nous prendrons comme outcome le décès à 30 jours, je dois donc utiliser les variables DTH30, qui est le décès à 30 jours et T3D30 qui est le temps de suivi adapté à cet outcome. Afin de vérifier ces 2 variables, je recree la variable temps de suivi pour l'outcome décès à 30 jours. Pour cela je cree d'abord la variable temps de suivi comme la différence entre la date des dernières nouvelles et la date d'admission dans l'étude. Puis je modifie cette variable telle que si le temps de suivi est supérieur à 30 jours alors je le ramène à 30 jours. Et la variable décès à 30 jours est modifiée comme ceci à partir de la variable décès : si le décès est survenu après 30 jours, alors la variable décès à 30 jours vaut 0. En comparant les variables DTH30 et T3D30 avec celles recrées, je trouve une différence pour 8 patients qui ont un temps de suivi inférieur à 30 jours et qui sont pourtant noté avec un temps de suivi de 30 jours pour un outcome décès à 30 jours.Et ça ne vient pas de la modification de la variable date de dernières nouvelles. Pour la suite du devoir, je décide de prendre T3D30 comme variable de temps de suivi pour le décès à 30 jours.

4) Données manquantes
```{r, eval = FALSE}
#Nb de sujets avec 0, 1, 2, 3, 4 NA
table(apply(apply(d,2,is.na),1,sum))
```

```{r}
qplot(as.factor(apply(apply(d,2,is.na),1,sum)), xlab = "Nombre de données manquantes par patient", ylab = "Nombre de patients", fill=
I("slategray2"), col =I("lavenderblush4"))
```

Figue 1. Nombre de valeurs manquantes par patient.

La base de données est globalement de bonne qualité, en effet aucun sujet n'a plus de 4 valeurs manquantes (sur les 63 variables). (Figure 1)

```{r, eval = FALSE}
#cb de NA pour chaque colonne
apply(apply(d,2,is.na),2,sum)
table(apply(apply(d,2,is.na),2,sum))

prop.table(table(is.na(d$ADLD3P)))
prop.table(table(is.na(d$URIN1)))
```
Si je regarde maintenant le nombre de valeurs manquantes pour chaque variable, je vois que ADLD3P (échelle ADL) et URIN1 (volume urinaire des 24h) ont respectivivement 75% et 53% de données manquantes tandis que les autres variables n'ont aucune ou moins de 5% de données manquantes.

5) Analyse descriptive préliminaire afin de repérer les incohérences

a- Présentation des variables de l'examen clinique et biologique :

- ADLD3P = Activities of daily living (ADL) : échelle d'autonomie de 0 à 12 (score>6 signe une dépendance).
- DAS2D3PC = Duke Activity Status Index (DASI) : auto-questionnaire de 12 items mesurant l'activité fonctionnelle. Le score va de 0 à 58.2, plus le score est élevé, meilleur est l'activité fonctionnelle.
- DNR1 = Do not resuscitate : 1 pour une interdiction de réanimation cardiopulmonaire, 0 sinon.
- SURV2MD1 = Probabilité de survie à 2 mois, estimée par model : de 0 à 1.
- APS1 = APACHE III Acute Physiology scores : score de prediction du risque de mortalité de patients hospitalisés en soins intensifs. Plus le score est élevé plus le risque de mortalité est important. Le score va de 0 à 299 mais ici seule la partie physiologie du score est utilisée.
- SCOMA1 = score de Glasgow. Ce score évalue l'état de conscience du patient: un score de 3 équivaut à un coma profond, un score de 15 est un état de conscience normal.
- PAFI1 = rapport PAO2/FIO2. Il permet de diagnostiquer une agression pulmonaire aigue(rapport>300), un rapport <200 definissant le syndrome de detresse respiratoire aigue et l'ECMO est envisageable en cas de rapport <50.
- PH1 = pH. Un pH normal varie entre 7,38 et 7.42.

- HEMA1 = taux d'hématocrite, normalement compris entre 41 et 50% environ.
- PACO21 = Pression artérielle en CO2. La PaCO2 normale varie de 35 à 45mmHg.
- ALB1 = taux d'albumine. Le taux d'albumine normal varie entre 25 et 44 g/L.
- WTKILO1 = poids
- TEMP1 = température corporelle
- HRT1 = fréquence cardiaque
- MEANBP1 = pression artérielle moyenne
- RESP1 = fréquence respiratoire
- SOD1 = natrémie
- POT1 = kaliémie
- CREA1 = créatininémie
- BILI1 = bilirubinémie
- URIN1 = volume urinaire des 24h

```{r}
#Je regroupe mes variables pour faciliter la présentation de l'analyse

#variables socio demo
vardes <- c("AGE", "SEX", "RACE", "EDU", "INCOME", "NINSCLAS")
var_quali_des <- c("SEX", "RACE", "INCOME", "NINSCLAS")
var_quanti_des <- c("AGE","EDU")
#variables antécédents médicaux
var_atcd <-  c("CAT1", "CAT2", "CA")
#date
var_date <- c("SADMDTE", "DSCHDTE", "DTHDTE", "LSTCTDTE")
# variables de comorbidités
var_com <- c("CARDIOHX", "CHFHX", "DEMENTHX", "PSYCHHX", "CHRPULHX", "RENALHX",
             "LIVERHX", "GIBLEDHX", "MALIGHX", "IMMUNHX", "TRANSHX", "AMIHX")
#variables de l'examen clinique et paraclinique
var_exam <- c("ADLD3P", "DAS2D3PC", "DNR1", "SURV2MD1", "APS1", "SCOMA1", "WTKILO1", "TEMP1",
              "MEANBP1", "RESP1", "HRT1", "PAFI1", "PACO21", "PH1", "WBLC1", "HEMA1", "SOD1", "POT1",
              "CREA1", "BILI1", "ALB1", "URIN1")
#var admission diagnosis
var_ad <- c("RESP", "CARD", "NEURO", "GASTR", "RENAL", "META", "HEMA", "SEPS", "TRAUMA", "ORTHO")

#Je vérifie qu'elles sont toutes là
#colnames(d)[!colnames(d) %in% c(var_quali_des, var_quanti_des, var_atcd, var_date, var_com, var_exam, var_ad)]
```

b- valeurs aberrantes

Pour détecter les valeurs aberrantes je dispose de plusieurs méthodes. Je regarde le tableau descriptif en calculant pour chaque variable quantitative la moyenne, l'intervalle interquartile et surtout le range (tableau non présenté). Je croise également les variables fréquence cardiaque, pression artérielle et fréquence respiratoire à la recherche d'incohérence.

Enfin je regarde la distribution des variables ce qui me permet de réperérer éventuellement d'autres incohérences (voir annexe : Figure A-1). Par exemple nous voyons un pic à 0 pour la variable WTKILO1 (nous l'avions déjà vu grâce au tableau descriptif, mais c'est un deuxième filet de sécurité). On observe également une pause dans les inclusions(variable SADMDTE), mais je ne sais pas comment le prendre en compte.

```{r, eval=FALSE}
# Je fais un tableau descriptif pour répérer d'éventuels incohérence (resultats non présenté en tableau mais décrit dans le texte)
write.table(print(do.call(rbind, lapply(c(var_quali_des, var_quanti_des, var_atcd, var_com, var_exam, var_ad), describe_all, d))),
file="clipboard", sep="\t")
range(d$LSTCTDTE)
range(d$SADMDTE)
```

```{r, include = FALSE}
# Je croise les variables pour chercher d'autres incohérences

#creation d'une variables permettant d'observer le profil des patients ayant FC, et/ou FR et/ou TA =0
HR0 <- ifelse(d$HRT1==0, "FC0", ifelse(!is.na(d$HRT1),0,NA))
FC0 <- ifelse(d$RESP1==0, "FR0", ifelse(!is.na(d$RESP1),0,NA))
TA0 <- ifelse(d$MEANBP1==0, "TA0", ifelse(!is.na(d$MEANBP1),0,NA))
#pas de NA de toutes façons pour ces valeurs
d$cleCR <- paste(HR0, FC0, TA0, sep="|")
table(d$cleCR)
d[d$cleCR=="FC0|FR0|0", c("HRT1", "RESP1", "MEANBP1", "DTH30", "SCOMA1", "TEMP1")]
```

```{r}
#SURVIE EN FONCTION DU SCORE DE GLASGOW

#je coupe SCOMA1 en classe
d$SCOMAcut <- cut(d$SCOMA1, breaks=20*(0:5), include.lowest = TRUE)

#plus score est élevé, plus le risque de deces est eleve.
ggsurv(survfit(Surv(T3D30, DTH30) ~ SCOMAcut, data=d), order.legend =FALSE)

#Je supprime la variable SCOMACUT du dataset
d$SCOMAcut <- NULL
```

Figure 2. Décès en fonction du score de glasgow découpé en quintile.

```{r}
# Modification des valeurs aberrantes


#"0|0|0" :aucune valeure ne vaut 0, ok
#"0|0|TA0" : LA tension artérielle vaut 0 mais pas le reste : probable erreur de mesure
d$MEANBP1 <- ifelse(d$cleCR=="0|0|TA0", NA, d$MEANBP1)
#"0|FR0|0"  : réa probablement pas d'accord sur quoi mesurer faire en cas de respirateur : chercher variable respirateur, sinon garder telle quelle?
#"0|FR0|TA0" : TA 0 et respi 0, je mets la FC à 0 aussi
d$HRT1 <- ifelse(d$cleCR=="0|FR0|TA0", 0, d$HRT1)
#"FC0|0|0" : seul la FC vaut0 : erreur, FC =NA
d$HRT1 <- ifelse(d$cleCR=="FC0|0|0", NA, d$HRT1)
#"FC0|0|TA0" : probablement encore une histoire de respirateur : si TA et FC vaut 0, alors FR aussi
d$RESP1 <- ifelse(d$cleCR=="FC0|0|TA0", 0, d$RESP1)
#"FC0|FR0|0" : quelq'un qui ne respire pas et n'a pas de pouls a une tenion nulle également
```

```
d$MEANBP1 <- ifelse(d$cleCR=="FC0|FR0|0", 0, d$MEANBP1)
#"FC0|FR0|TA0" : toutes les valeurs sont à 0 ok

#Je supprime la clé
d$cleCR <- NULL

#recodage du poids:
d$WTKILO1 <- ifelse(d$WTKILO1==0, NA, d$WTKILO1) #kilo max 244 ok (j'en ai vu) #515 poids = 0!
#recodage de l'hématocrite
d$HEMA1 <- ifelse(d$HEMA1<10, NA, d$HEMA1)
#recodage de la PACO2
d$PACO21 <- ifelse(d$PACO21>100, NA, d$PACO21)
#recodage de l'albumine
d$ALB1 <- ifelse(d$ALB1>10, NA, d$ALB1)

```

Les scores qui ont été mesurés, semblent tous avoir des résultats plausibles excepté le score de Glasgow qui a été modifié : d'un score allant de 3 à 15 on passe à un score allant de 0 à 100. Le score de 100 est probablement lié à un coma sévère car les patients avec un score plus élevé ont un risque de décès augmenté (alors que dans l'échelle d'origine, 3 correspond au coma sévère, et 15 à une conscience normale) (Figure 2).
Je laisse la variable telle quelle car elle a de toute évidence été modifiée, je ne veux pas la remodifier une seconde fois.
La température corporelle peut aller de 27° en cas d'hypothermie profonde à 42-43° en cas de forte fièvre, donc je ne modifie pas.
Un patient ne peut pas avoir un poids de 0, ces 515 patients sont recodés en NA pour le poids.
Un patient peut être en aplasie mais je ne suis pas sûr qu'une hyperleucocytose de plus de 40 leucocytes/10^9L soit possible, cependant dans le doute concernant le taux maximum en cas de leucémie et au vue de la distribution de la variable qui semble plausible, je laisse telle quelle.
Hematocrite : une anémie chronique peut entrainer une anémie profonde avec une hématocrite de 10%, en dessous de cette valeur je mets NA. (un taux de 66% est possible).
Un taux de PAcO2 de 156 mmHg me semble complètement impossible, au delà de 100 mmHg je met NA.
Un pH de 6.6 est extrêmement faible et n'est pas viable mais je ne suis pas sûre qu'on ne puisse pas l'observer en réanimation.
Une créatininémie de 2200umol/L ou 25 mg/dL est possible en cas d'insuffisance rénale terminale. Un volume urinaire de 0 et de 9000mL est possible également(mais cette variable n'est de toute façon pas utilisée pour l'analyse).
Une valeur normale d'albumine se situe entre 3.5 et 5 g/dL. Les patients avec un taux d'albumine supérieur à 10g/dL auront une valeur NA pour l'albumine.
Les valeurs de natrémie, kaliémie, fréquence respiratoire, fréquence cardiaque, pression artérielle moyenne et bilirubine semblent plausibles.

En observant ensemble la fréquence cardiaque, la fréquence respiratoire et la tension artérielle, je note certaines incohérences : parfois 2 de ces variables peuvent valoir 0 mais pas la troisième, et parfois une seule de ces variables vaut 0. Ainsi 1 patient à une tension artérielle à 0 mais une fréquence cardiaque et une fréquence respiratoire différente de 0, 47 patients ont une fréquence respiratoire à 0 mais une fréquence cardiaque et une tension artérielle différente de 0 et 3 patients ont une fréquence cardiaque à 0 mais une fréquence respiratoire et une tension artérielle différente de 0.

Je modifie donc ces 3 variables selon les règles suivantes :

- Si deux de ces variables valait 0, alors la valeur de 0 était systématiquement attribuée à la troisième.
- Si une seule de ces variables valait 0 alors elle était systématiquement transformée en valeur manquante.

```{r}
quanti <- c("AGE","EDU", "ADLD3P", "DAS2D3PC","SURV2MD1", "APS1", "SCOMA1", "WTKILO1", "TEMP1", "MEANBP1", "RESP1", "HRT1", "PAFI1", "PACO21",
"PH1", "WBLC1", "HEMA1", "SOD1", "POT1", "CREA1", "BILI1", "ALB1", "URIN1")
binaire <-  binaire <- c("SEX", "CARDIOHX", "CHFHX", "DEMENTHX", "PSYCHHX", "CHRPULHX", "RENALHX", "LIVERHX", "GIBLEDHX", "MALIGHX", "IMMUNHX",
"TRANSHX", "AMIHX", "RESP", "CARD", "NEURO", "GASTR", "RENAL", "META", "HEMA", "SEPS", "TRAUMA", "ORTHO")
quali <- c("RACE", "INCOME", "NINSCLAS", "CAT1", "CAT2", "CA")
```

II - IMPUTATION DES VALEURS MANQUANTES

Les valeurs manquantes doivent absolument être prises en charge pour la suite du projet car on ne peut pas faire de score de propension pour les patients ayant une ou plusieurs variables explicatives manquantes.

1) Règles d'imputation

Règles de décision concernant l'imputation ou non des variables :

- une variable avec plus de 20% de données manquantes ne sera pas imputée et ne sera pas inclue dans l'analyse.
- un patient avec plus de 50% de données manquantes ne sera pas inclu dans l'analyse.
- un patient n'ayant pas d'information concernant la variable swan ganz qui est la variable explicative d'interet ne sera pas inclu dans l'analyse car non informatif.
- j'imputerai les variables avec des valeurs manquantes uniquement si plus de 5% des patients on des valeurs manquantes, sinon je supprimerai simplement ces patients.

2) Description des valeurs manquantes (valeurs aberrantes transformées en valeurs manquantes)

Je vais donc dans un premier temps observer les données manquantes par variable et par sujet, après avoir transformé les valeurs aberrantes en valeurs manquantes.

```{r, include= FALSE}
d2 <- d[ , ! colnames(d) %in% c(var_date)] #les variables dates ne sont pas des variables explicatives
nacol <- apply(d2, 2,function(x)sum(is.na(x))/length(x)*100)
nacol[nacol>20]
napat1 <- apply(d2, 1, function(x)sum(is.na(x)))
table(napat1)

#la fonction md.pattern()de la librairie mice fait la même chose mais la sortie est illisible ici car nous avons trop de sujets et trop de variables.
```

Comme vu précédemment, Le score d'autonomie ADL(ADLD3P) a environ 75% de données manquantes et la variable volume d'excrétion urinaire (URIN1) a plus de 50% de données manquantes. Je supprime donc ces deux variables de l'analyse.
250 patient patients n'ont aucune donnée manquante, 5485(95.6%) patients ont au moins 1 donnée manquante dont : 1930 patients avec 1 donnée manquante, 2698 patients avec 2 de données manquantes, 796 patients avec 3 données manquantes, 64 patients avec 4 données manquantes et 1 patient avec 5 données manquantes. Tous les patients ont donc moins de 50% de données manquantes.
Il n'y a pas de valeurs manquantes concernant la variable explicative d'intérêt SWAN (SWAN = 0 pas de traitement par CCD, SWAN = 1 traitement par CCD).

```{r}
#je supprime les variables avec plus de 20% de donnees manquantes.
d <- subset(d, select=-c(ADLD3P, URIN1))
```

```{r, include = FALSE}
d2 <- d[ , ! colnames(d) %in% c(var_date)]
nacol <- apply(d2, 2,function(x)sum(is.na(x)))
nacol[nacol>0]
napat1 <- apply(d2, 1, function(x)sum(is.na(x)))
table(napat1)
```

En supprimant ADLD3P et URIN1 de l'analyse, j'ai 608 patients avec au moins une donnée manquante, soit plus de 5 % de patients. Je ne peux donc pas simplement supprimer les patients avec des données manquantes.
6 variables ont au moins une donnée manquante : la tension artérielle (15 NA), la fréquence cardiaque (60 NA), l'albuminémie (2 NA) l'hématocrite (8 NA), la PaCO2 (21 NA) et le poids (515 NA).

J'impute les variables explicatives avec le package mice en faisant l'hypothèse que les données manquantes le sont aléatoirement. J'utililse la technique d'imputation multiples mais je ne réalise qu'un seul jeu de de données imputé. Toutes les variables du jeu de données sont prédictrices (y compris l'outcome DTH30, le temps de suivi jusqu'à cet outcome et SWAN, selon Jonathan A C Sterne et al 2009 BMJ) excepté les dates. Par contre je ne prend pas en compte DEATH qui n'est pas notre outcome ni le temps de suivi jusqu'à DEATH.

```{r}
#Je supprime DEATH et time du jeu de données car on ne s'en servira plus
d <- subset(d, select=-c(DEATH, time))
```

```{r}
#back up des données avant imputation
d2 <- d
```

```{r, message = FALSE}
#mice bug si je laisse les dates
init = mice(d[ ,!colnames(d)%in%var_date], maxit=0)
meth = init$method
predM = init$predictorMatrix

#Je ne veux pas que les variables ROWNAMES et PTID soit des prédicteurs.(et SWANT est redondante)
predM[ ,c("ROWNAMES", "PTID", "SWANT")]=0

#Je ne veux pas que soit prédites ni l'outcome ni la variable d'interet (elles ne sont pas NA de toutes façon mais par rigueur je l'écris quand meme)
meth[c("DTH30", "SWAN", "SWANT")]=""

#J'impute par imputation multiple
set.seed(103)
imputed = mice(d[ ,!colnames(d)%in%var_date], method=meth, predictorMatrix=predM, m=1)
imputed <- complete(imputed)
d[ ,!colnames(d)%in%var_date] <- imputed


# #NB: autre possibilité d'imputation, non réalisée ici : imputer les données manquantes par la médiane pour chaque variable
# #dput(names(nacol[nacol>0])) #c("MEANBP1", "HRT1", "ALB1", "HEMA1", "PACO21", "WTKILO1")
# for (i in c("MEANBP1", "HRT1", "ALB1", "HEMA1", "PACO21", "WTKILO1")){
#   d[ , i] <- ifelse(is.na(d[ , i]), median(d[ , i], na.rm = T), d[ , i])
# }
```

3) Descrition des variables imputées avant et après imputation
Je compare les variables qui ont été imputées, avant et après imputation. Je ne compare que visuellement, je ne fais pas de test.

```{r}
#avant imputation
kable(do.call(rbind, lapply(names(nacol[nacol>0]), describe_all, d2)))
```

Table 1. Variables qui ont été imputées, avant imputation
```{r}
#apres imputation
kable(do.call(rbind, lapply(names(nacol[nacol>0]), describe_all, d)))
```

Table 2. Variables qui ont été imputées, après imputation

La distribution des variables a l'air semblable avant et après imputation.

III - DESCRIPTION DE LA POPULATION DE L'ETUDE (après imputation)
```{r}
#Je retire ADLD3P et URIN1 de var_exam
var_exam <- var_exam [! var_exam %in% c("ADLD3P", "URIN1")]

#description chez les swan=0
des_noSWAN <- do.call(rbind, lapply(c(var_quali_des, var_quanti_des, var_atcd, var_com, var_exam, var_ad), describe_all, d[d$SWAN==0, ]))
#description chez les swan=1
des_SWAN <- do.call(rbind, lapply(c(var_quali_des, var_quanti_des, var_atcd, var_com, var_exam, var_ad), describe_all, d[d$SWAN==1, ]))
#description tout traitement confondu
des_all <- do.call(rbind, lapply(c(var_quali_des, var_quanti_des, var_atcd, var_com, var_exam, var_ad), describe_all, d))

#mise en forme
des_tot <- cbind(des_noSWAN[ , 1], des_SWAN[ , 1], des_all[ , 1:2])
des_tot <- des_tot[str_sub(rownames(des_tot), -1, -1)!=0, ]
kable(des_tot)
```

Table 3. Caractéristiques de base des patients (après imputation des valeurs manquantes).
*variable quantitative

Il me manque une information essentielle : je ne sais pas si les variables ont été mesurée à l'admission ou après pose de la sonde de swan ganz. Je vais considérer que c'est avant pose de la sonde, sinon ça n'a pas de sens.

Je regarde les différences de caractéristiques de base entre le groupe des patients cathétérisés et le groupe des patients non cathétérisés (Table 3). Je ne réalise pas de test car une petite différence numériquement peut être significative du fait de la taille de l'échantillon.
Les patients traités par CCD sont plus souvent des hommes, avec une couverture médicale moins précaire, ils ont plus souvent une comorbidité

cardiaque, avec une pression artérielle moyenne plus faible et une fréquence cardiaque plus élevée, plus souvent admis pour pathologie cardiaque que les patients non traités par CCD, et moins souvent admis pour pathologie respiratoire, ils ont une PAO2/FIO2 plus faible(ratio inferieur à 200 signe un syndrome de détresse respiratoire aigu), une pression artérielle en CO2 inferieur à la normale et plus faible que le groupe non traité, une creatininémie plus élevée et un score d'apache plus élevé.

Donc globalement, les patients traités par CCD sont dans un état cardiaque et rénal plus sévère et avait donc de base plus de risque de décéder. Il y a donc un biais d'indication et il faut donc absolument prendre en compte cette différence d'état de base en considération lorsque l'on teste si le traitement par CCD diminue le décès à 30 jours.
Pour cela, nous allons donc réaliser un appariemment sur le score de propension.

IV - Appariemment sur le score de propension

Le score de propension permet d'avoir pour chaque patient sa probabilité d'être traité par CCD, en fonction de ses charactéristiques de bases. On apparie ensuite un patient traité par CCD avec un patient non traité par CCD qui avait la même probabilité d'être traité que la patient effectivement traité. On se place donc dans la situation d'un essai clinique randomisé ou chaque patient à la même probabilité de recevoir l'un ou l'autre des traitements.

1) Sélection des variables à intégrer : tests bivariés

Je dois intégrer dans le score de propension les variables qui sont associées significativement avec le décès à 30 jours ou avec le décès à 30 jours et le traitement par cathétérisme. Pour cela je réalise deux séries de tests bivariés. Une première série testant l'association entre chaque variable et le décès à 30 jours et une deuxième série testant l'association entre chaque variable et le traitement par cathétérisme. Pour rappel, les variables ADLD3P et URIN1 ne sont pas dans l'analyse.
J'utilise des modèles de régression logistique à une variable explicative, la variable à expliquer étant soit la variable cathétérisme, soit la variable décès à 30 jours (variables toutes deux binaires) et la variable explicative étant chacune des variables à tester.

Les conditions de validité sont toujours respectées car j'ai 9 classes au maximum (variable CAT1) pour 1918 évènements, donc la condition des 5 à 10 variables par variable est toujours respectée.

```{r, include = FALSE}
#TESTS BIVARIES :

#Je retire ADLD3P et URIN1
var_exam <- c("DAS2D3PC", "DNR1", "SURV2MD1", "APS1", "SCOMA1", "WTKILO1", "TEMP1", "MEANBP1", "RESP1", "HRT1", "PAFI1", "PACO21", "PH1", "WBLC1",
"HEMA1", "SOD1", "POT1", "CREA1", "BILI1", "ALB1")
#var_exam <- var_exam [! var_exam %in% c("ADLD3P", "URIN1")]

#avec swanganz
list_swan <- lapply(c(var_ad, var_atcd, var_exam, var_com, vardes), function(x){
  #print(x)
  d$var <- d[,x]
  if (all(levels(as.factor(d$var)) %in% c(0,1))) d$var <- as.factor(d$var)
  mod <- glm(SWAN~var,d, family="binomial")
  test <- summary(mod)

  if (nrow(coef(test))>2){ #cas variable explicative qualitative
    test <- drop1(mod, .~., test="Chisq")
    ab <- test$`Pr(>Chi)`[2]
  } else {
    ab <- coef(test)[2, "Pr(>|z|)"]
  }
  ab <- round(ab, 3)
  ab <- data.frame(ab)
  ab$signif <- ifelse(ab$ab<0.05,"*","")

  rownames(ab) <- x
  colnames(ab) <- c("coef pvalue SWAN","significatif SWAN")
  return(ab)
})
list_swan <- do.call(rbind, list_swan)

list_death <- lapply(c(var_ad, var_atcd, var_exam, var_com, vardes), function(x){
  #print(x)
  d$var <- d[,x]
  if (all(levels(as.factor(d$var)) %in% c(0,1))) d$var <- as.factor(d$var)
  #mod <- glm(DEATH~var,d, family="binomial")
  mod <- glm(DTH30~var,d, family="binomial")
  test <- summary(mod)

  if (nrow(coef(test))>2){ #cas variable explicative qualitative
    test <- drop1(mod, .~., test="Chisq")
    ab <- test$`Pr(>Chi)`[2]
  } else {
    ab <- coef(test)[2, "Pr(>|z|)"]
  }
  ab <- round(ab, 3)
  ab <- data.frame(ab)
  ab$signif <- ifelse(ab$ab<0.05,"*","")
  #ab$ab <- ifelse(ab$ab<0.001, "<0.001", ab$ab)
  rownames(ab) <- x
  colnames(ab) <- c("coef pvalue DEATH","significatif DEATH")
  return(ab)
})
list_death <- do.call(rbind, list_death)

list_pval <- cbind(list_swan, list_death)

#prendre les valeurs : liées soit uniquement au décès, soit liées à la sonde et au décès (ce qui revient à prendre variable significative pour le décès ici)
list_pval$select <- ifelse (list_pval$`coef pvalue DEATH`<0.05, 1, 0)
list_pval[list_pval$select==1, ]
nrow(list_pval[list_pval$select==1, ])
dput(rownames(list_pval[list_pval$select==1,]))
#c(rownames(list_pval[list_pval$select==1,])[c(1:8,12:36)],"CA","INCOME","NINSCLAS","CAT1")
#dput(rownames(list_pval[list_pval$select==1, ])) #pour éviter de tout taper à la main!ya plus qu'à copier coller

#var avec DEATH
# varps <- c("RESP", "GASTR", "RENAL", "HEMA", "SEPS", "TRAUMA", "ADLD3P",
#            "DAS2D3PC", "DNR1", "CA", "SURV2MD1", "APS1", "SCOMA1", "WTKILO1",
#            "TEMP1", "MEANBP1", "PACO21", "PH1", "HEMA1", "POT1", "CREA1",
#            "BILI1", "ALB1", "URIN1", "CARDIOHX", "CHFHX", "DEMENTHX", "PSYCHHX",
#            "CHRPULHX", "LIVERHX", "MALIGHX", "IMMUNHX", "TRANSHX", "AGE",
#            "INCOME", "NINSCLAS", "CAT1", "CAT2")

#var avec DTH30
```

```
varps <- rownames(list_pval[list_pval$select==1,])


#variables liées au décès et au traitement
rownames(list_pval[list_pval$`coef pvalue DEATH`<0.05 & list_pval$`coef pvalue SWAN`<0.05, ])
#variables liées uniquement au décès
rownames(list_pval[list_pval$`coef pvalue DEATH`<0.05 & list_pval$`coef pvalue SWAN`>=0.05, ])
#variables liées uniquement à la swan
rownames(list_pval[list_pval$`coef pvalue DEATH`>=0.05 & list_pval$`coef pvalue SWAN`<0.05, ])
```

2 variables sont liées au décès uniquement : TEMP1 et LIVERHX.

34 variables sont liées au décès et au traitement par cathétérisme : RESP, CARD, NEURO, GASTR, HEMA, SEPS, CAT1, CAT2, CA, DAS2D3PC, DNR1, SURV2MD1, APS1, SCOMA1, WTKILO1, MEANBP1, PAFI1, PACO21, PH1, WBLC1, HEMA1, CREA1, BILI1, ALB1, CARDIOHX, CHFHX, DEMENTHX, PSYCHHX, CHRPULHX, GIBLEDHX, MALIGHX, AGE, INCOME, NINSCLAS.

J'aurai donc 36 variables dans le score de propension.

Les 10 variables liées uniquement au cathétérisme ne sont pas prises en comptes dans le score : RENAL, TRAUMA, RESP1, HRT1, SOD1, IMMUNHX, TRANSHX, AMIHX, SEX, EDU.

2) Calcul du score de Propension

Je calcule le score de propension à partir d'un modèle de régression logistique avec comme outcome le traitement par swan ganz et comme variables explicatives toutes les variables liées au décès à 30j uniquement(facteurs pronostiques) ou au décès et au traitement(facteurs de confusion). Je n'intègre pas les variables liées uniquement au traitement pour ne pas perdre de puissance à la fin.

Condition de validité :
C'est une régression logistique, je dois avoir 5 à 10 évènement par variable. J'ai 36 variables explicatives dans le score de propension dont 5 variables qualitatives : 9 classes pour CAT1, 7 classes pour CAT2, 3 classes pour CA, 4 classes pour INCOME, 6 classes pour NINSCLAS. Ces variables qualitatives seront donc transformées en 9+7+3+4+6-5= 24 variables binaires. Soit l'équivalent de 36-5+24 = 55 variables dans le modèle logistique pour 1918 évènements. J'ai donc plus de 10 évènements par variable explicative, les conditions de validité sont respectées.

```{r, include = FALSE}
#score de propension
ps <- glm(formula(paste0("SWAN ~ ",paste(varps,collapse="+"))), data = d, family="binomial")

d2 <- d[apply(apply(d[ ,varps], 2, is.na),1,sum)==0, ] #J'elimine les lignes avec au moins 1 NA dans les variables slectionnes varps
d2$logitps <- as.vector(predict(ps, type = "response")) #response is the default for binomial model

#conditions de validité : nombre d'evenements :
table(d$DTH30)
```
```{r}
#Histogramme du score de propension en fonction du groupe de traitement
prs_df <- data.frame(pr_score = predict(ps, type = "response"),
                     SWAN = ps$model$SWAN)
labs <- paste("actual intervention:", c("no CCD", "CCD"))
prs_df %>%
  mutate(SWAN = ifelse(SWAN == 1, labs[2], labs[1])) %>%
  ggplot(aes(x = pr_score)) +
  geom_histogram(color = "white") +
  facet_wrap(~SWAN) +
  xlab("Probabilité d'être traité par CCD") +
  theme_bw()
```

Figure 3. Distribution de la probablité d'être traité par CCD selon le traitement effectif par CCD.

On voit que la distribution du score n'est pas la même parmi les patients ayant été cathétérisé et ceux n'ayant pas été cathétérisé : beaucoup de patient n'ayant pas été cathétérisé avait une faible probabilité d'être cathétérisé (distribution en L), alors que la distribution est plutôt en cloche centré autour de 0.5 pour les patients ayant été cathétérisé (Figure 3). On comprend donc que nécessairement l'appariemment va supprimer de nombreux patients.

3) Appariemment sur le score de propension

A partir du score de propension, j'apparie un sujet cathétérisé avec un sujet non cathétérisé ayant un score de propension proche c'est à dire une probabilité d'être cathétérisé proche. Le sujets non appariés sont écartés de l'analyse. Les sujets devrait donc se ressembler dans les 2 groupes.

Deux packages principalement nous permette de réaliser l'appariemment; le package Matching et le package MatchIt. Le package Matching est celui que je trouve le plus simple pour reconstituer le numéro de paire et qui conserve le plus de patients en gardant les memes paramètres.

```{r}
#============================
#Package Matching :
#cours de David Hajage, MD PhD, département de biostatistiques de la Pitié Salpétrière

#-----
#Le traitement (ici SWan Ganz) doit être en true false pour le package matching, je prend donc la variable "SWANT"

#-----
#appariement
tmp <- Match(Tr = d2$SWANT, X = d2$logitps, M = 1, replace = FALSE, caliper = 0.2, ties = FALSE)
#NB : ties=FALSE est l'équivalent de ratio=1 de MatchIt

#-----
#reconstitution du tableau avec les sujets appariés
d2.app <- d2[c(tmp$index.treated, tmp$index.control),]#index.treated et index.control donne le numero des lignes selectionnees par le matching. Le premier individu de index.treated est matchée avec le 1er de index.ctrl.

#-----
#reconstitution du numéro de paire
d2.app$paire <- rep(1:length(tmp$index.treated), 2) #lignes de d2.app : d'abord les traités de chaque paire puis les control de chaque paire, donc on répète le numéro de paire
d2.app <- d2.app[order(d2.app$paire, d2.app$SWAN==1),] #on réordonne selon la paire SG(1)NSG(1) SG(2)NSG(2) etc
#=> donc ce tableau prend toutes les variables, uniquement les lignes correspondant aux individus matché et pour chaque individu on connait son numéro de paire


#============================
#package MatchIt
```

```
#https://stanford.edu/~ejdemyr/r-tutorials-archive/tutorial8.html#exercise
#https://stanford.edu/~ejdemyr/r-tutorials-archive/matching.R

#-----
#appariement
mod_match <- matchit(formula(paste0("SWAN ~ ",paste(varps,collapse="+"))),
                     method = "nearest", replace = FALSE, ratio = 1, m.order = "smallest", caliper=0.2, data = d2[,c("SWAN",varps,"PTID")])
#MatchIt ne sait pas gérer les NA (même si les colonnes de la formule n'ont pas de NA) => Je dois préciser les colonnes qui m'interesse dans data
(celles où je sais qu'il n'y a pas de NA)
#par exemple j'ai toujours des NA pour DTH30

#-----
#reconstitution du numéro de paire
matches<-data.frame(mod_match$match.matrix)

# > dim(matches)
# [1] 2025    1
#2025 lignes, qui correspondent aux 2025 individus traités (avant matching)

# > head(matches)
#       X1
#   2  <NA>
#   5   927
#   10 2383
#le patient SWAN de la ligne nommée 2 du tableau d2  n'est matché avec aucun patient non SWAN : il faut eliminer la ligne
#le patient SWAN de la ligne nommée 5 du tableau d2 est matché avec le patient non SWAN de la ligne nommée 927

#J'élimine les les lignes avec NA (correspond aux traités qui n'ont pas été appariée)
matches <- na.omit(matches)
#position de chaque patient traité dans le tableau d2
groupSG1<-match(row.names(matches), row.names(d2))
#position de chaque patient non traité dans d2
groupSG0<-match(matches$X1, row.names(d2))

#-----
#reconstitution du tableau avec les sujets appariés et leur numéro de paire
d2.appbis <- d2[c(groupSG1, groupSG0),]
d2.appbis$paire <- rep(1:length(groupSG1), 2)
d2.appbis <- d2.appbis[order(d2.appbis$paire, d2.appbis$SWAN==1), ]

#si jamais j'ai finalement besoin des distances :
dta_m <- match.data(mod_match)
dtm <- merge(d2.appbis, dta_m[,c("PTID","distance","weights")], by="PTID", all=T)

# #après vérification, distance est en fait logitps...
# mean(dtm$distance)
# mean(dtm$logitps)


#============================
#J'ai donc deux tableaux :
#dtm pour celui crée avec le package MatchIt
#d2.app pour celui crée avec le package Matching
#la seule différence est que Matching garde plus d'individu et qu'il a l'air plus simple d'utilisation.

```

J'utilise un caliper de 0.2, c'est à dire un seuil d'appariemment de 0.2 x sd(logit(score de propension)).
En utilisant le package Matching (fonction Match), sans remise, sans ex aequo, avec un ratio 1:1 et avec un caliper de 0.2, je conserve 3142 patients, 1571 dans chaque groupe.

4) Vérification de l'équilibre des variables entre les deux groupes après appariemment

L'appariemment a normalement permis d'avoir des patients globalement comparables en terme de charactéristiques de base, car ayant la même probabilité d'être traité par CCD dans les deux groupes. Il se peut cependant que certaines variables soit mal équilibrée, et c'est ce que nous allons vérifier ici. Seules les variables ayant servie à construire le score de propension doivent être regardées.

Tout d'abord, je transforme mes variables qualitatives en n-k variables binaires, k étant le nombre de classe de la variable qualitative.

```{r, include = FALSE}
# Je transforme les qualitatives en binaires.

varps_quanti <- varps[sapply(varps, function(variable) length(levels(dtm[,variable]))<=2)]
varps_quali <- varps[!varps%in% varps_quanti]

#----------------------
#jeu avant appariemment
dbis <- d2
for (j in varps_quali){
  num <- which(varps_quali==j)
  a <- model.matrix( ~ dbis[ ,j])
  #pour avoir un nom de variable reconnaissable dans les schéma (si on laisse tel quel ça donne dbis[ ,j] CHF par exemple)
  colnames(a) <- gsub("dbis",j,  colnames(a))
  colnames(a) <- gsub("\\[", "",  colnames(a))
  colnames(a) <- gsub("\\]", "",  colnames(a))
  colnames(a) <- gsub("\\,", "",  colnames(a))
  colnames(a) <- gsub("j", "",  colnames(a))
  colnames(a) <- gsub(" ", "_",  colnames(a))
  #créer les  variables binaires
  for (i in 1:(length(colnames(a))-1)){
    dbis[ ,colnames(a)[i+1]] <- a[ ,i+1]
  }
  #créer un vecteur avec les noms de variables bianires crées
  vec_tmp <- colnames(a)[-1]
  vec_var <- if(num==1) vec_tmp else c(vec_tmp, vec_var)
}
#retirer les variables qualitatives non binarisees
dbis[ ,varps_quali] <- NULL
d2_b <- dbis

#----------------------
#jeu apparié (package matching)
```

```
dbis <- d2.app
for (j in varps_quali){
  num <- which(varps_quali==j)
  a <- model.matrix( ~ dbis[ ,j])
  #pour avoir un nom de variable reconnaissable dans les schéma (si on laisse tel quel ça donne dbis[ ,j] CHF par exemple)
  colnames(a) <- gsub("dbis",j,  colnames(a))
  colnames(a) <- gsub("\\[", "",  colnames(a))
  colnames(a) <- gsub("\\]", "",  colnames(a))
  colnames(a) <- gsub("\\,", "",  colnames(a))
  colnames(a) <- gsub("j", "",  colnames(a))
  colnames(a) <- gsub(" ", "_",  colnames(a))
  #créer les  variables binaires
  for (i in 1:(length(colnames(a))-1)){
    dbis[ ,colnames(a)[i+1]] <- a[ ,i+1]
  }
  #créer un vecteur avec les noms de variables bianires crées
  vec_tmp <- colnames(a)[-1]
  vec_var <- if(num==1) vec_tmp else c(vec_tmp, vec_var)
}
#retirer les variables qualitatives non binarisees
dbis[ ,varps_quali] <- NULL
d2_app_b <- dbis


#-------------------------
#Je prend donc ces deux jeux de données pour regarder l'appariemment:
d2_app_b
d2_b
```


```{r}
#nouvelles variables varps (inclus les variables binaires nouvellement créées)
varps_new <- names(d2_b)[names(d2_b) %in% varps]
new_bin <- names(d2_b)[! names(d2_b) %in% names(d2)]
varps_new <- c(varps_new, new_bin)

#je sépare les variables binaires des autres variables quantatives
varps_quanti2 <- varps_new[sapply(varps_new, function(variable) length(levels(as.factor(d2_b[,variable]))) > 2)] #maintenant qu'il n'y a plus que des quanti binaire et non binaire, je peux transformer les quanti en facteur et dire que si plus de 2 levels, c'est une quanti non binaire.
varps_binaire <- varps_new[!varps_new%in% varps_quanti2]

```

J'ai ensuite plusieurs méthodes possibles pour regarder si l'appariemment a équilibré la distribution des variables dans les groupes traité par CCD et non traité par CCD:

- Méthode 1 : je regarde la moyenne de chaque variable en fonction du score de propension (annexe : figure A-2). Je vois donc pour des patients de ces 2 groupes ayant la même probabilité d'être traité, si la moyenne de la variable est semblable. Bien sûr si les courbes se superposent parfaitement, on peut dire que la distribution est semblable dans les deux groupes.Si elles sont disjointes à certaines valeures d'abscisse, nous pouvont dire que pour les individus ayant tel probabilité d'être traité, la moyenne des variables diffèrent. Ainsi dans les courbes présentées en annexe, on voit ainsi que les individus avec une forte probabilité d'être traité ont une moyenne qui semble différer entre les deux groupes pour INCOME, NINCLAS, RESP et SEPS, et que les individus avec une faible probabilité d'être traité ont une moyenne qui semble différer pour les variables RESP, HEMA et DNR1. Or il me semble qu'on ne cherche pas à ce que les individus ayant la même probabilité d'être traité soit exactement semblable 2 à 2 (même si bien sûr c'est idéal) mais plutôt que les populations des groupes traités et non traités soit homogène, comme c'est le cas lorsque l'on randomise. Il me semble donc plus pertinent de voir si globablement les distribution sont les mêmes dans les deux groupes, et les deux méthodes présentées ci dessous répondent à cette question.

- Méthode 2 : je regarde la différence standardisée des moyennes (SMD), c'est à dire la différence entre la moyenne dans le groupe cathétérisé et la moyenne dans le groupe non cathétérisé divisé par la variance commune. J'ai séparé les schémas en variables quantitatives non binaires (figure 4) et variables binaires et qualitatives binarisées (Figure 5) pour plus de lisibilité. En instaurant un seuil de smd à 0.1 comme conseillé dans la littérature, je vois que l'appariemment établi un équilibre entre les deux groupes pour toutes les variables ayant servi à calculer le score de propension.


```{r, eval = FALSE}
#METHODE 2 : standardized mean difference avec matchit (variance du groupe traitement)

#https://cran.r-project.org/web/packages/tableone/vignettes/smd.html
#https://github.com/kaz-yos/tableone/blob/1d47ec186b2e351937e5f9712dad3881380ab12e/vignettes/smd.Rmd

#Le package matchit fournit une différence standardisée
smd1 <- summary(mod_match, standardize = TRUE) #fait la balance pour chaque binaire tirée de la variable quali
dataplot <- data.frame(variable = rownames(smd1$sum.all),
                       unmatched = abs(smd1$sum.all[,"Std. Mean Diff."]),
                       matched = abs(smd1$sum.matched[,"Std. Mean Diff."]))
dataplotmelt <- melt(data          = dataplot,
                     id.vars       = c("variable"),
                     variable.name = "Method",
                     value.name    = "SMD")
colnames(dataplotmelt) <- c("variable","Method","SMD")
varNames <- as.character(dataplot$variable)[order(dataplot$unmatched)]
#organise les variables dans le plot en fonction de la valeur de smd unmatched
dataplotmelt$variable <- factor(dataplotmelt$variable,
                                levels = varNames)
ggplot(data = dataplotmelt[!dataplotmelt$variable %in% c("distance","SWAN"), ], mapping = aes(x = variable, y = SMD,
                                      group = Method, color = Method)) +
  geom_point() +
  geom_hline(yintercept = 0.1, color = "red", size = 0.1) +
  coord_flip() +
  theme_bw() + theme(legend.key = element_blank())


#mais en vérifiant avec la variable AGE je vois qu'ils utilisent la formule avec la variance du groupe traitement et non pas la variance commune (Austin et al 2011).
mymean <- as.numeric(by(d2[,"AGE"], d2$SWAN, mean))
mysd <- as.numeric(by(d2[,"AGE"], d2$SWAN, sd))
(mymean[2]- mymean[1])/sqrt((mysd[2]^2+mysd[1]^2)/2) #smd avec la variance commune
(mymean[2]- mymean[1])/mysd[2] #smd avec l'écart-type du groupe traitement : correspond au smd donné par matchit

#et les variables quali sont transformées en k variables binaire et non k-1, avec k le nombre de classes.
#de plus j'ai finalement préféré utiliser le package Matching qui ne fournit pas de différence standardisée.

#J'ai donc refait cette différence standardisée "à la main"
```

```{r}
#================================================================
#METHODE 2 BIS standardized mean difference avec variance commune

#----------------
#----------------
#CALCUL DE LA SMD

#-----
#jeu non apparié
dbis <- d2_b

#calcul de smd pour les variables quantitatives non binaires
smd_quanti_um <- data.frame(smd = sapply(varps_quanti2, function(x){
  mymean <- as.numeric(by(dbis[,x], dbis$SWAN, mean))
  mysd <- as.numeric(by(dbis[,x], dbis$SWAN, sd))
  mysmd <- abs((mymean[2]- mymean[1])/sqrt((mysd[2]^2+mysd[1]^2)/2))
  #mysmd <- (mymean[2]- mymean[1])/mysd[2]
  return(mysmd)
}), variable = varps_quanti2)

#calcul de smd pour les variables binaires (et qualitatives binarisées)
smd_bin_um <- data.frame(smd = sapply(varps_binaire, function(x){
  mymean <- as.numeric(by(dbis[,x], dbis$SWAN, mean))
  mysmd <- abs((mymean[2]- mymean[1])/sqrt((mymean[2]*(1 - mymean[2]) + mymean[1]*(1 - mymean[1]))/2))
  #mysmd <- (mymean[2]- mymean[1])/mysd[2]
  return(mysmd)
}), variable = varps_binaire)
smdum <- rbind(smd_quanti_um, smd_bin_um)

#-----
#jeu apparié
dbis <- d2_app_b

#calcul de smd pour les variables quantitatives non binaires
smd_quanti_m <- data.frame(smd = sapply(varps_quanti2, function(x){
  mymean <- as.numeric(by(dbis[,x], dbis$SWAN, mean))
  mysd <- as.numeric(by(dbis[,x], dbis$SWAN, sd))
  mysmd <- abs((mymean[2]- mymean[1])/sqrt((mysd[2]^2+mysd[1]^2)/2))
  #mysmd <- (mymean[2]- mymean[1])/mysd[2]
  return(mysmd)
}), variable = varps_quanti2)

#calcul de smd pour les variables binaires (et qualitatives binarisées)
smd_bin_m <- data.frame(smd = sapply(varps_binaire, function(x){
  mymean <- as.numeric(by(dbis[,x], dbis$SWAN, mean))
  mysmd <- abs((mymean[2]- mymean[1])/sqrt((mymean[2]*(1 - mymean[2]) + mymean[1]*(1 - mymean[1]))/2))
  #mysmd <- (mymean[2]- mymean[1])/mysd[2]
  return(mysmd)
}), variable = varps_binaire)
smdm <- rbind(smd_quanti_m, smd_bin_um)

```

```{r}
#----------------
#----------------
#CONSTRUCTON DES GRAPHES

#quanti
dataplot <- data.frame(variable = smd_quanti_m$variable, unmatched = smd_quanti_um$smd, matched = smd_quanti_m$smd)
dataplotmelt <- melt(data = dataplot, id.vars = "variable", variable.name = "Method", value.name = "SMD")
colnames(dataplotmelt) <- c("variable","Method","SMD")
varNames <- as.character(dataplot$variable)[order(dataplot$unmatched)]
#organise les variables dans le plot en fonction de la valeur de smd unmatched
dataplotmelt$variable <- factor(dataplotmelt$variable,
                                levels = varNames)
ggplot(data = dataplotmelt[!dataplotmelt$variable %in% c("distance","SWAN"), ],
       mapping = aes(x = variable, y = SMD, group = Method, color = Method)) +
  geom_point() +  geom_hline(yintercept = 0.1, color = "red", size = 0.1) +
  coord_flip() +  theme_bw() + theme(legend.key = element_blank(), legend.title = element_blank())

```

Figure 4. Différence standardisée des moyennes des variables quantitatives.


```{r}
#binaire
dataplot <- data.frame(variable = smd_bin_m$variable, unmatched = smd_bin_um$smd, matched = smd_bin_m$smd)
dataplotmelt <- melt(data = dataplot, id.vars = "variable", variable.name = "Method", value.name = "SMD")
colnames(dataplotmelt) <- c("variable","Method","SMD")
varNames <- as.character(dataplot$variable)[order(dataplot$unmatched)]
#organise les variables dans le plot en fonction de la valeur de smd unmatched
dataplotmelt$variable <- factor(dataplotmelt$variable,
                                levels = varNames)
ggplot(data = dataplotmelt[!dataplotmelt$variable %in% c("distance","SWAN"), ],
       mapping = aes(x = variable, y = SMD, group = Method, color = Method)) +
  geom_point() +  geom_hline(yintercept = 0.1, color = "red", size = 0.1) +
  coord_flip() +  theme_bw() + theme(legend.key = element_blank(), legend.title = element_blank())

#==============================
```

Figure 5. Différence standardisée des moyennes des variables qualitatives (binarisées pour regarder l'équilibre) et des variables binaires.

- Méthode 3 : je regarde les intervalles de confiances des différentes variables, j'ai réalisé les intervalles de confiance des moyennes pour les variables quantitatives (figure 6) et des fréquences de 1 pour les variables binaires et qualitatives binarisées (figure 7). Les variables quantitatives sont standardisées, ce qui permet une lecture plus aisée du graphique des variables quantitatives. Je ne sais pas comment réaliser un équivalent de cette standardisation avec les variables binaires et le graphique est donc moins facilement analysable. A noter que les variables pour lesquelles on n'a pas pu calculer d'intervalle de confiance pour non respect des conditions de validité (np>=5 et n(1-p)>=5) ne sont pas présente dans le graphe. Ce sont un des niveaux de CAT1 et deux des niveaux de CAT2. On voit là aussi que globalement les variables sont plutôt bien équilibrée après appariemment car les intervalles de confiance se recouvrent deux à deux.

```{r}

#METHODE 3 IC à la main : variables quantitatives

getMIC_quanti <- function(.data, .vec_group){ #.vec_group en 0/1
  dt <- lapply(names(.data), function(var){
    num <- which(var==names(.data))
    #standardisation
    .data[,var] <- (.data[,var] - mean(.data[,var]))/sd(.data[,var])

    x <- .data[.vec_group == 1, var]
    df1 <- data.frame(mymean = mean(x),
                      ICminSG = mean(x) - 1.96*sqrt(var(x)/nrow(.data)),
                      ICmaxSG = mean(x) + 1.96*sqrt(var(x)/nrow(.data)),
                      myy = num*2-0.5,
                      groupe = "SWAN",
                      colour = 1)

    x <- .data[.vec_group == 0, var]
    df0 <- data.frame(mymean = mean(x),
                      ICminSG = mean(x) - 1.96*sqrt(var(x)/nrow(.data)),
                      ICmaxSG = mean(x) + 1.96*sqrt(var(x)/nrow(.data)),
                      myy = num*2+0.5,
                      groupe = "NO SWAN",
                      colour = 2)

    df <- rbind(df1,df0)
  })
  dt <- do.call(rbind, dt)
  dt$var <- rep(names(.data)[names(.data) %in% varps_new],each=2)
  return(dt)
}

#plot des var quanti
#df <- getMIC_quanti(d2_app_b[ ,c("SWAN", varps_quanti2)])
df <- getMIC_quanti(d2_app_b[ ,c(varps_quanti2)], d2_app_b$SWAN)
df <- na.omit(df) #retire les var quali dont les IC n'ont pas été calculés

m<-tapply(df$myy, df$var, mean)
m<-data.frame(var=names(m), y=m)

df2<-merge(df, m, by="var", all=T)
df2<-df2[order(df2$var, df2$group),]
df2$y<-ifelse(df2$groupe=="SWAN", 0.25, -0.25)+df2$y

laby<-unique(df2$var)

col <- hue_pal()(length(1:2))

g <- ggplot(data=df2, aes(x=mymean, xmin = ICminSG, y = y, xend = ICmaxSG, colour=groupe)) + geom_point()
g <- g + labs(y = "variable")
g<-g+scale_y_continuous(name="Variables", breaks=m$y, labels=m$var)

for (i in 1:nrow(df)){
  g <- g + geom_segment(x = df2$ICminSG[i], y = df2$y[i], xend = df2$ICmaxSG[i], yend = df2$y[i], colour = col[df2$colour[i]])
  g <- g + geom_segment(x = df2$ICminSG[i], y = df2$y[i]-0.5, xend = df2$ICminSG[i], yend = df2$y[i]+0.5, colour = col[df2$colour[i]])
  g <- g + geom_segment(x = df2$ICmaxSG[i], y = df2$y[i]-0.5, xend = df2$ICmaxSG[i], yend = df2$y[i]+0.5, colour = col[df2$colour[i]])
}
g

```

Figure 6. Intervalle de confiance des variables quantitatives utilisées pour calculer le score de propension, en fonction du traitement par CCD.

```{r}
#METHODE 3 IC à la main : variables binaires

getMIC_quali <- function(.data, .vec_group){

  dt <- lapply(names(.data), function(var){
    num <- which(var==names(.data))
    x <- .data[.vec_group == 1, var]
    P <- as.numeric(prop.table(table(x))[2]) #length(x[x==1])/length(x)

    if(!is.na(P) & (nrow(.data) * P) >= 5 & (nrow(.data) * (1-P)) >= 5){
      df1 <- data.frame(myfreq = P,
                        ICminSG = P - 1.96*sqrt(P*(1-P)/nrow(.data)),
                        ICmaxSG = P + 1.96*sqrt(P*(1-P)/nrow(.data)),
                        myy = num*2-0.5,
                        np = nrow(.data) * P,
                        nq = nrow(.data) * (1-P),
                        groupe = "SWAN",
                        colour = 1)
    } else {
      df1 <- data.frame(myfreq = P,
                        ICminSG = NA, ICmaxSG = NA,
                        myy = num*2-0.5,
                        np = nrow(.data) * P,
                        nq = nrow(.data) * (1-P),
                        groupe = "SWAN",
                        colour = 1)
    }


    x <- .data[.vec_group == 0, var]
    P <- as.numeric(prop.table(table(x))[2])
    if(!is.na(P) & (nrow(.data) * P) >= 5 & (nrow(.data) * (1-P)) >= 5){
      df0 <- data.frame(myfreq = P,
                        ICminSG = P - 1.96*sqrt(P*(1-P)/nrow(.data)),
                        ICmaxSG = P + 1.96*sqrt(P*(1-P)/nrow(.data)),
                        myy = num*2+0.5,
                        np = nrow(.data) * P,
                        nq = nrow(.data) * (1-P),
                        groupe = "NO SWAN",
                        colour = 2)
```

```
    } else {
      df0 <- data.frame(myfreq = P,
                        ICminSG = NA, ICmaxSG = NA,
                        myy = num*2-0.5,
                        np = nrow(.data) * P,
                        nq = nrow(.data) * (1-P),
                        groupe = "NO SWAN",
                        colour = 2)
    }

    df <- rbind(df1,df0)
  })
  dt <- do.call(rbind, dt)
  dt$var <- rep(names(.data), each=2)
  return(dt)
}



#plot des var binaire
df <- getMIC_quali(d2_app_b[ ,varps_binaire], d2_app_b$SWAN)
df <- na.omit(df) #retire les var quali dont les IC n'ont pas été calculés

m<-tapply(df$myy, df$var, mean)
m<-data.frame(var=names(m), y=m)

df2<-merge(df, m, by="var", all=T)
df2<-df2[order(df2$var, df2$group),]
df2$y<-ifelse(df2$groupe=="SWAN", 0.25, -0.25)+df2$y

laby<-unique(df2$var)

col <- hue_pal()(length(1:2))

g <- ggplot(data=df2, aes(x=myfreq, xmin = ICminSG, y = y, xend = ICmaxSG, colour=groupe)) + geom_point()
g <- g + labs(y = "variable")
g<-g+scale_y_continuous(name="Variables", breaks=m$y, labels=m$var)

for (i in 1:nrow(df)){
  g <- g + geom_segment(x = df2$ICminSG[i], y = df2$y[i], xend = df2$ICmaxSG[i], yend = df2$y[i], colour = col[df2$colour[i]])
  g <- g + geom_segment(x = df2$ICminSG[i], y = df2$y[i]-0.5, xend = df2$ICminSG[i], yend = df2$y[i]+0.5, colour = col[df2$colour[i]])
  g <- g + geom_segment(x = df2$ICmaxSG[i], y = df2$y[i]-0.5, xend = df2$ICmaxSG[i], yend = df2$y[i]+0.5, colour = col[df2$colour[i]])
}
g

```
```

Figure 7. Intervalle de confiance des variables qualitatives et binaires utilisées pour calculer le score de propension, en fonction du traitement par CCD (les variables qualitatives ont été binarisées).

Dans le futur, je préférerai quand même l'analyse de l'équilibre par la méthode smd qui est plus radide à coder et plus facile à lire. C'est dailleurs la méthode que j'ai le plus souvent rencontrée dans les publications.

V - Analyses dans la population appariée

```{r, eval = FALSE}

#http://imai.princeton.edu/research/files/matchit.pdf
#journals.sfu.ca/jmde/index.php/jmde_1/article/download/431/414
#http://r.iq.harvard.edu/docs/matchit/2.4-20/matchit.pdf
#FE Harrell, Regression Modeling Strategies(2nd Ed)
```

Pour répondre à la question "le cathétérisme cardiaque droit modifie-t-il la survie à 30 jours?", je peux utiliser deux méthodes différentes. Soit j'utilise une régression logistique conditionnelle, soit j'utilise un modèle de Cox. Dans les deux cas, j'utilise mon échantillon apparié et je stratifie sur le numéro de paire.

1) Régression logistique conditionnelle

La variable à expliquer est la mort à 30 jours, la variable explicative est le cathétérisme cardiaque et je stratifie sur le paire.

```{r, include = FALSE}
d2.app$DTH30 <- as.numeric(as.character(d2.app$DTH30))
d2.app$SWAN <- as.numeric(as.character(d2.app$SWAN))
mod1 <- clogit(DTH30~SWAN + strata(paire), method="efron", data=d2.app) #marche avec efron
#mod1 <- clogit(SWAN~DTH30 + strata(paire), method="efron", data=d2.app) #marche avec efron
exp(coefficients(mod1))
exp(confint(mod1))
```

Je n'ai pas trouvé de source indiquant explicitement quelles sont les conditions de validité de la régression logistique conditionnelle. Par défaut j'ai donc considérée que c'étaient les mêmes conditions que pour la régression logistique et elles sont ici respectées : j'ai plus de 5 à 10 évènements par variable explicative (la stratification sur la paire n'est pas une variable explicative).

Le coefficient du cathétérisme est significatif (pvalue<0.01), le cathétérisme cardiaque droit a donc un effet significatif sur le risaue de décès à 30 jours. le coeffcient vaut 0.17, l'exponentiel du coefficient me permet d'obtenir l'Odds ratio du risque de décès associé au cathétérisme. L'OR est de 1.19 avec un intervalle de confiance à 95% [1.05-1.34]. Le décès n'est pas un évènement rare, je ne peux donc pas interpréter l'OR comme un RR mais je peux dire que le risque de décès est augmenté lorsque le patient est cathétérisé.

2) Analyse de survie : Modèle de Cox

```{r, eval = FALSE}
    #nombre de strates
    length(unique(d2.app$paire))

    #nombre d'évènement
    #nombre de strates sans evenements
    noevt <- aggregate(d2.app[ ,c("paire", "DTH30")], by = list(d2.app$paire), sum)
    prop.table(table(noevt$DTH30))

```

Une deuxième manière de réaliser le calcul est par modèle de cox. Dans ce cas il est problématique de prendre en compte l'appariemment par une
```

stratification sur la paire car FE Harrell nous précise dans son livre Regression Modeling Strategies(2nd Ed) p.482 que si le nombre de strates est très grand par rapport au nombre total d'évènement, on perd en efficience(? "Loss of efficiency" dans le texte, je ne sais pas bien comment le traduire). Or on a ici 1569 paires ou strates pour 1055 évènements, ce qui peut être considéré comme un grand nombre de strates relativement au nombre d'évènements bien qu'aucun seuil ne soit précisé. Et de plus FE Harrell nous précise qu'une strate qui ne contient aucun évènement ne contribue pas à l'information et qu'une telle situation doit donc être évitée si possible. Or ici 44% des paires sont sans évènements et ne contribue pas à l'information avec donc j'imagine une perte de puissance.
J'utiliserai donc l'option cluster(paire) plutôt que strata(paire) pour prendre en compte l'appariemment dans le modele de cox. Cette option prend en compte le design apparié, et on regarde alors la variance robuste et le test du score robuste. Cette méthode calcule une vraisemblance globale et non pas strate par strate comme lorsque l'on stratifie, évitant donc de perdre de l'information.

```{r}
mod2 <- coxph(Surv(T3D30,DTH30) ~ SWAN + cluster(paire), data=d2.app) #J'utilise DTH30 afin d'avoir le même échantillon que pour la régression logistique conditionnelle.
  z <- cox.zph(mod2, transform = "rank")
  #cat("Test de Harrell\n\n")
  #print(z)
  pval <- round(z$table[,3],3)
  #cat(paste0("\nTest de Harrell p value: ", pval))
  #non signif si p>=0.05 (condition respectée qd non signif)

  #résidus de Shoenfeld
  z <- cox.zph(mod2, transf="identity")
  plot(z, main="Résidus de Shoenfeld", lwd=1)
  abline(h=0, col="red")
  abline(h=coef(mod2), col="blue")
  text(x = 20, y = 1, labels = paste0("test de Harrell p = ", pval))

```

Figure 8. Résidus de Shoenfeld en fonction du temps. P value à partir du test de Harrell.

```{r, eval = FALSE}
  #transformation en fonction du temps
  #decoupage par les temps d'evenement
   ti <- sort(unique(c(0,d2.app$timeb[d2.app$DTH30==1])))
  slat <- d2.app
  slat$start <- 0
  slat$stop <- slat$timeb
  slat$evt <- slat$DTH30
  slat <- survSplit(Surv(stop,evt)~.,slat,start="start",cut=ti)

  #variable dependante du temps
  slat$at<-slat$SWAN*sqrt(slat$stop)
  slat$at<-slat$SWAN/(slat$stop)
  slat$at <-slat$SWAN*log10(slat$stop)
  slat$at <-slat$SWAN*(slat$stop^0.3)
  slat$at <-slat$SWAN*(slat$stop^0.7)
  slat$at <-slat$SWAN*log(slat$stop)
  slat$at <-slat$SWAN*(slat$stop^3)
  slat$at <-slat$SWAN*(slat$stop^2)
  slat$at <-slat$SWAN*(slat$stop)
  #et idem avec : log, *t, ^2, ^3, ^0.7

  #verification de la significativité de la variable dependante du temps
  coxt <- coxph(Surv(start, stop, evt) ~ SWAN + at + cluster(paire), data=slat)
  test <- summary(coxt)
  pval_at <- test$coefficients["at","Pr(>|z|)"] #si <0.05 on peut regarder les courbesm sinon inutiles
  pval_at
  #Les test des coef des variables dépendantes du temps sont sup à 0.05 pour toutes les transformations (donc les transformations ne conviennent pas)

```

```{r, eval = FALSE}
 #aucune transformation ne convient mais voici quand même les codes pour revérifier les conditions si j'avais trouvé une transformation convenable à l'étape précédente
      zt <- cox.zph(coxt, transf="identity")
    for (i in 1:(nrow(zt$table)-1)){
      iz<-i
      plot(zt[iz])
      abline(h=0, col="red") #l'intervalle de confiance doit contenir 0 en tout abscisse, et ce pour les résidus de swan et de swan dépendant du temps.
    }
       zit <- cox.zph(coxt, transform = "rank")
    pval <- round(zit$table[,3],3)
    #Les 3 p doivent etre >=0.05


    # #interpretation que j'aurai faite si j'avais trouvé une transformation satisfaisante
    # mod <- coxph(Surv(timeb, DTH30) ~ SWAN + at +  strata(paire), data = slat)
    # test <- summary (mod)
    #
    # S <- vcov(mod)
    # b <- coef(mod)
    # t <- 15 #choisir le temps en jours
    # transf <- "*t^0.3"
    # if (transf=="*t^0.7") t_t <- t^0.7
    # if (transf=="log") t_t <- log(t)
    # if (transf=="*t^2") t_t <- t^2
    # if (transf=="*t") t_t <- t
    # if (transf=="*t^3") t_t <- t^3
    # if (transf=="*t^0.3") t_t <- t^0.3
    #
    # variance <- S[1,1]+S[2,2]*(t_t)^2+2*S[1,2]*(t_t)
    # m <- b[1]+b[2]*(t_t) #coef de l'HR
    #
    #
    # cat(paste0("\n\nScore test: ", round(test$sctest["pvalue"],3)))
    # HR <- round(exp(m),3)
    # IC <- round(exp(m + qnorm(0.975)*sqrt(variance) * c(-1,1)),3)
    # cat(paste0("\nHR[95%CI] = ",HR, " [", IC[1], "-", IC[2], "] ", "pour t = ", t, " jours" ))

```

```{r}
# Interpretation du modèle de cox sans transformation
    mod2 <- coxph(Surv(timeb,DTH30) ~ SWAN + cluster(paire), data=d2.app) #J'utilise DTH30 afin d'avoir le même échantillon que pour la régression logistique conditionnelle.
    test <- summary (mod2)
    test$robscore[3]
#calcul du HR et de l'intervalle de confiance en utilisant la variance robuste. J'ai plus de 30 sujets je peux donc faire une approximation par la loi normale, d'après le théorème central limite et utiliser cette formule :
    cHR <- test$coef[1]
    sr <- test$coef[4] #racine de la variance robuste
    error <- qnorm(0.975)*sr/sqrt(nrow(d2.app))
    ICl <- cHR-error
    ICu <- cHR+error
    paste(round(exp(c(cHR, ICl, ICu)),3))
```

L'hypothèse des risques proportionnels n'est pas vérifiée. En effet l'intervalle de confiance des résidus de shoenfeld n'incluent pas 0 en tous points (figure 8). Il faut donc en toute rigueur que j'ajoute une variable SWAN dépendante du temps. Cependant je n'ai pas trouvé de transformation qui convienne, aucun coefficient de paramètre dépendant du temps n'étant significatif lorsqu'on l'ajoute au modèle (les résidus de Shoenfeld des modèles ne doivent donc même pas être regardés). Voici les transformations essayeées : log, racine carrée, *temps, 1/temps, racine cube, carré, cube et puissance de 0.7. J'analyse donc le modèle de Cox sans ajouter de variable dépendante du temps, mais il faudra avoir en tête que le résultat est erroné.

L'ajout d'un cluster sur la paire me permet d'avoir une variance robuste prenant en compte l'appariemment. J'utilise cette variance robuste pour calculer l'intervalle de confiance à 95% du Hazard ratio: 1.236 [1.233-1.239]. Je regarde le test du score robuste pour conclure : p<0.001. Le risque de décès est donc significativement différent selon que le patient est cathétérisé ou non. Et c'est dans le sens d'un risque plus grand chez les patients cathétérisés avec un risque de décès à 30 jours multiplié par 1.236.

Je retrouve cette information graphiquement en traçant une courbe de survie en fonction du traitement par cathéterisme cardiaque droit.

3) Représentation graphique de la survie à 30 jours en fonction du traitement par CCD par la méthode de Kaplan-Meier

La méthode de Kaplan Meier permet de représenter graphiquement les courbes de survie (une courbe par groupe).
Condition de validité de la méthode de Kaplan Meier :
-   censure indépendante de la probabilité de survenue de l'evenement
-   probabilité de survie indépendante du moment de recrutement dans l'étude
-   censure indépendante du groupe

```{r, fig.width=8, fig.height=6}

s <- d2.app
.title <- "Survie selon traitement par cathétérisme cardiaque droit"
vec_time_IC <- c(10, 20, 30)
var <- "CCD"
s$tps <- s[ ,"timeb"]

km <- survfit(Surv(tps,DTH30)~SWAN , data=s, conf.int=.95)
km0 <- survfit(Surv(tps,DTH30)~SWAN , data=s[s$SWAN==0,], conf.int=.95)
km1 <- survfit(Surv(tps,DTH30)~SWAN, data=s[s$SWAN==1,], conf.int=.95)

#pour IC95%
skmi0<-summary(km0, time=vec_time_IC-0.2)
skmi1<-summary(km1, time=vec_time_IC-0.1)

group0 <- paste0("\nIn group ", var, " = 0\n ")
group1 <- paste0("\nIn group ", var, " = 1\n ")

#survies aux tps choisis
cat(group0)
sv <- summary(km0, time=vec_time_IC)
df <- data.frame(time = sv$time, survival = sv$surv*100, LCI = sv$lower*100, UCI = sv$upper*100)
df[,2:4] <- round(df[,2:4], 0)
cat(paste0("At ", df$time, " days, survival[95%CI] ", df$survival, "% [",df$LCI,"% - ",df$UCI, "%]\n"))

cat(group1)
sv <- summary(km1, time=vec_time_IC)
df <- data.frame(time = sv$time, survival = sv$surv*100, LCI = sv$lower*100, UCI = sv$upper*100)
df[,2:4] <- round(df[,2:4], 0)
cat(paste0("At ", df$time, " days, survival[95%CI] ", df$survival, "% [",df$LCI,"% - ",df$UCI, "%]\n"))

#preparation legende
leg<-str_sub(names(km$strata),-1,-1)
leg <- ifelse(leg==0, "non traité", "traité")
col <- hue_pal()(length(leg))

#courbe de survie
g <- ggsurv(km, CI=FALSE, order.legend=FALSE, surv.col=col, cens.col=col) +
  #changement des axes
  scale_x_continuous(breaks=seq(0,max(s$tps),10), labels=(0:(length(seq(0,max(s$tps),10))-1))*10) +
  scale_y_continuous(labels=percent) +
  labs(x="durée de suivi, jours", y="Survie, %", title=.title) +
  #changement legende
  guides (linetype = FALSE) +
  scale_colour_discrete( labels = leg) +
  theme(title = element_text(size=14), axis.text = element_text(size=12), axis.title=element_text(size=14),
        legend.text = element_text(size=12)) +
  theme(legend.position="right", legend.title=element_blank()) +
  #espace autour du schéma
  theme(plot.margin = unit(c(1,1,3,2), "cm")) #+ #top, right, bottom, left

  #theme_bw()

#intervalle de confiance
for (i in 1:length(vec_time_IC)) {
  g <- g + geom_segment(x = skmi0$time[i], y = skmi0$lower[i], xend = skmi0$time[i], yend = skmi0$upper[i], colour = col[1])
  g <- g + geom_segment(x = skmi0$time[i] - 0.1, y = skmi0$lower[i], xend = skmi0$time[i] + 0.1, yend = skmi0$lower[i], colour = col[1])
  g <- g + geom_segment(x = skmi0$time[i] - 0.1, y = skmi0$upper[i], xend = skmi0$time[i] + 0.1, yend = skmi0$upper[i], colour = col[1])

  g <- g + geom_segment(x = skmi1$time[i], y = skmi1$lower[i], xend = skmi1$time[i], yend = skmi1$upper[i], colour = col[2])
  g <- g + geom_segment(x = skmi1$time[i] - 0.1, y = skmi1$lower[i], xend = skmi1$time[i] + 0.1, yend = skmi1$lower[i], colour = col[2])
  g <- g + geom_segment(x = skmi1$time[i] - 0.1, y = skmi1$upper[i], xend = skmi1$time[i] + 0.1, yend = skmi1$upper[i], colour = col[2])
}
```

```
#pvalue
pval <- test$robscore[3] #je reprend le test qui était le summary du modele avec cluster
pval <- ifelse(pval<0.001, "test du score robuste \n p<0.001 *",paste0("test du score robuste \n p=", pval))
g <- g + annotate("text",
                  x=0.75*max(km$time),
                  y=0.90*max(km$surv),
                  label=pval)

gt <- ggplotGrob(g)
gt$layout$clip[gt$layout$name=="panel"] <- "off"
grid.draw(gt)
```

Figure 9. Courbes de survie par la méthode de Kaplan Meier de la survie à 30 jours en fonction du traitement par cathétérisme cardiaque droit. p value calculée par modèle de cox avec cluster sur la paire.

```{r}
#table de survie
at_risk <- summary(km)
dfrisk <- data.frame(at_risk[c(2:5,7)])
dfrisk <- data.frame(at_risk[c("time", "n.risk", "n.event", "surv", "strata")])
dfrisk$surv <- paste(round(dfrisk$surv,2)*100,"%")

#groupe swan=0
dfrisk0 <- dfrisk[dfrisk$strata=="SWAN=0",-5]
dfrisk0b <- data.frame(time=c(10,20,30))
dfrisk0b$n.risk <- dfrisk0$n.risk[dfrisk0$time%in%dfrisk0b$time]
dfrisk0b$n.event <- sapply(dfrisk0b$time, function(x) {
  num <- which (dfrisk0b$time==x)
  if (num==1) res <- sum(dfrisk0$n.event[dfrisk0$time<=x])
  else res <- sum(dfrisk0$n.event[dfrisk0$time>dfrisk0b$time[num-1]  & dfrisk0$time<=x])
  return(res)
})
dfrisk0b$surv <- dfrisk0$surv[dfrisk0$time%in%dfrisk0b$time]

dfrisk0tab <- data.frame(t(dfrisk0b))[-1,]
colnames(dfrisk0tab) <- paste0("J",dfrisk0b$time)
dfrisk0tab$J0 <- c(length(unique(s$PTID[s$SWAN==0])), 0, "100%")
dfrisk0tab <- dfrisk0tab[,c(ncol(dfrisk0tab),1:(ncol(dfrisk0tab)-1))]
kable(dfrisk0tab)
```

Table 4. Table de survie des patients non traités par cathétérisme cardiaque droit

```{r}
#groupe swan=1
dfrisk1 <- dfrisk[dfrisk$strata=="SWAN=1",-5]
dfrisk1b <- data.frame(time=c(10,20,30))
dfrisk1b$n.risk <- dfrisk1$n.risk[dfrisk1$time%in%dfrisk1b$time]
dfrisk1b$n.event <- sapply(dfrisk1b$time, function(x) {
  num <- which (dfrisk1b$time==x)
  if (num==1) res <- sum(dfrisk1$n.event[dfrisk1$time<=x])
  else res <- sum(dfrisk1$n.event[dfrisk1$time>dfrisk1b$time[num-1]  & dfrisk1$time<=x])
  return(res)
})
dfrisk1b$surv <- dfrisk1$surv[dfrisk1$time%in%dfrisk1b$time]

dfrisk1tab <- data.frame(t(dfrisk1b))[-1,]
colnames(dfrisk1tab) <- paste0("J",dfrisk1b$time)
dfrisk1tab$J0 <- c(length(unique(s$PTID[s$SWAN==1])), 0, "100%")
dfrisk1tab <- dfrisk1tab[,c(ncol(dfrisk1tab),1:(ncol(dfrisk1tab)-1))]
kable(dfrisk1tab)
```

Table 5. Table de survie des patients traités par cathétérisme cardiaque droit

Sur la figure 9, on observe 2 courbes de survie, une pour le groupe non traité pour CCD et une pour le groupe traité par CCD. A chaque temps, les "marches d'escalier" représentent les patients qui ont eu l'évènement, faisant diminuer le nombre de personnes à risque d'évènement(c'est à dire les non répondeurs). Les croix représentent à chaque temps la présence de censure (perdus de vue ou exclus vivant). Par exemple dans le groupe non traité, je passe de près de 85% de survie à J10 à près de 75% de survie à J20 dans le groupe non traité. Dans la table de survie (table 4) je vois que ça correspond à 126 patients ayant eu l'évènement entre J10 et J20. Pour le groupe traité, je lis sur la courbe que je passe de 80% environ de survie à J10 à 70% environ à J20. Dans la table de survie (table 5) cela correspond à 154 patients ayant eu l'évènement.
La courbe de Kaplan Meier nous permet d'observer graphiquement que la survie est meilleure lorsque les patients ne sont pas traités par cathétérisme cardiaque droit.

Conclusion

Nous voulions savoir si le cathétérisme cardiaque droit(CCD) améliorait la survie à 30 jours des patients admis en réanimation. Les données étaient observationnelles, sans randomisation de l'intervention. On ne pouvait donc pas exclure la présence de biais de confusion et d'attrition, qui ont d'ailleurs été mis en évidence dans le devoir : les patients traités par CCD étaient dans un état plus grave. En l'état aucune analyse ne pouvais être faite car elle aurait été biaisée. Afin de tenir compte des charactéristiques de bases nous avons donc réalisé un score de propension et apparié sur ce score. Ainsi chaque individus traité et non traité appariés avaient la même probabilité d'être traités. Après appariemment et en tenant compte de cet appariemment dans l'analyse, nous mettons en évidence une association entre le cathétérisme cardiaque droit par sonde de swan ganz et le risque de mortalité à 30 jours : le cathétérisme cardiaque droit est associé à une augmentation du risque de mortalité. Mais bien que l'utilisation du score de propension augmente le niveau de causalité, on ne peut pas conclure avec le même niveau de causalité qu'un essai randomisé.

References

Connors, A. F., T. Speroff, N. V. Dawson, C. Thomas, F. E. Harrell, D. Wagner, N. Desbiens, et al. “The Effectiveness of Right Heart Catheterization in the Initial Care of Critically Ill Patients. SUPPORT Investigators.” JAMA 276, no. 11 (September 18, 1996): 889–97.

Sterne, J. A C, I. R White, J. B Carlin, M. Spratt, P. Royston, M. G Kenward, A. M Wood, and J. R Carpenter. “Multiple Imputation for Missing Data in Epidemiological and Clinical Research: Potential and Pitfalls.” BMJ 338, no. jun29 1 (September 1, 2009): b2393–b2393. doi:10.1136/bmj.b2393.

Cours du Dr David Hajage "Evaluation de l’effet d’un traitement en condition réelle d’utilisation (scores de propension et scores pronostiques)", janvier 2017.

Austin, Peter C. “An Introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies.” Multivariate Behavioral Research 46, no. 3 (May 31, 2011): 399–424. doi:10.1080/00273171.2011.568786.

Ho, Daniel E., Kosuke Imai, Gary King, and Elizabeth A. Stuart. “MatchIt: Nonparametric Preprocessing for Parametric Causal Inference.” Journal of

Statistical Software 42, no. 8 (2011). doi:10.18637/jss.v042.i08.

#Annexes

Figure A-1. Distribution des variables avant modification et imputation

```{r}
#4/ distribution des items avant modification et imputation

.l <- lapply(colnames(d), function(x){
  if (class(d[,x])=="Date") {
    #browser()
    qplot(d[ ,x], main=x, xlab=NULL, fill=I("navajowhite3"), col=I("pink4"))
  }
  else{
    if (length(names(table(d[,x])))>15) qplot(as.numeric(as.character(d[ ,x])), main=x, xlab=NULL, fill=I("navajowhite3"), col=I("pink4"))
    else qplot(as.factor(d[ ,x]), main=x, xlab=NULL, fill=I("navajowhite3"), col=I("pink4"))
  }

})
ml <- marrangeGrob(.l,ncol=3,nrow=3,top = NULL)
#ggsave(file="distrib bef dm.pdf", ml)
print(ml)

```

Figure A-2. Equilibre des variables après appariemment : moyenne pour chaque variable en fonction du score de propension

```{r}
#METHODE 1 checking balance: moyenne pour chaque variable en fonction du score de ponpension

#fonction qui plot la distribution des variables en fonction du score de propension avant et après matching
fn_bal <- function(dta, variable) {
  #browser()
  dta$variable <- dta[, variable]
  dta$SWAN <- as.factor(dta$SWAN)
  ggplot(dta, aes(x = logitps, y = variable, color = SWAN)) +
    geom_point(alpha = 0.01, size = 1.5) +
    geom_smooth(method = "loess", se = F) +
    xlab("Propensity score") +
    ylab(variable) +
    theme_bw() +
     theme(axis.text = element_blank(), axis.title.x= element_blank(), axis.title.y = element_text(size=10))
}

.l <- lapply(c(varps_quanti2, varps_binaire), function(variable){
  #print(variable)
  num <- which(c(varps_quanti2, varps_binaire)==variable)
  if (num %% 3 != 0) fn_bal(d2_app_b, variable)
  else fn_bal(d2_app_b, variable) + theme(legend.position = "none")
})

ml <- marrangeGrob(.l, nrow=4, ncol=3, top = NULL)
print(ml)
#ggsave(file="distrib mean variables after matching 20170402.pdf", ml)

#============================

```

```
####################
# 01-library_EPID.R #
####################


require(lubridate)
require(dplyr)
require(ggplot2)
library(grid)
require(gridExtra)
library(survival)
library(stringr)
library(scales)
library(GGally)
library(MatchIt)
library(Matching)
library(mice)
library(psy)
library(knitr)

############################
#   02- functions_EPID.R   #
############################

describe_qualitative <- function(vec_var, .data){
  table_var_quali <- lapply(vec_var, function(i){
    data <- .data[,i]
    names_levels <- levels(as.factor(data))
    a <- lapply(names_levels, function(x) {
      tmp <- as.numeric(table(data)[x])
      tmpbis <- round(as.numeric(prop.table(table(data))[x]),3)*100
      tmptot <- paste0(tmp," (",tmpbis,"%)")

      nNA <- table(is.na(data))
      pNA <- round(prop.table(table(is.na(data))),3)
      if (is.na(nNA[2]))  {
        if (which(names_levels==x)==1) nNA <- paste0 (0," (0%)") #NA pour ligne 1
        else nNA <- ""
      }
      else {
        browser()
        if (which(names_levels==x)==1){   #NA pour ligne 1
          nNA <- as.numeric (nNA[names(nNA)==TRUE])
          pNA <- as.numeric (pNA[names(pNA)==TRUE])*100
          nNA <- paste0(nNA," (",pNA,"%)")
        }
        else nNA <- ""
      }
      cbind(tmptot,nNA)

    })
    a <- do.call(rbind,a)
    #a <- cbind (a,nNA)
    rownames(a) <- paste0(i,"_",names_levels)
    colnames(a) <- c("valeur","missing values")
    # a <- rbind (a,nNA)
    # rownames(a)[-nrow(a)] <- paste0(i,"_",names_levels)
    return(a)
  })
  table_var_quali <- do.call(rbind,table_var_quali)
  table_var_quali <- data.frame(table_var_quali)
  table_var_quali$range <- NA
  colnames(table_var_quali) <- c("valeur", "missing values", "range")
  return (table_var_quali)
}

describe_quantitative <- function(vec_var, .data){
  table_var_quanti <- lapply(vec_var, function(i){ #median ou moyenne? (sachant qu'on ne verifie pas normalite des baselines)
    data <- .data[,i]
    if (!is.numeric(data)) {
      a <- data.frame("not num", "", "")
      colnames(a) <- c("valeur", "missing values", "range")
      rownames(a) <- i
    } else {
      med <- round(median (data,na.rm=T),2)
      quant <- round(quantile(data,na.rm=T),2)
      Q1 <- quant[2]
      Q3 <- quant[4]
      a <- paste0(med," (",Q1,"-",Q3,")")
      #browser()

      nNA <- table(is.na(data))
      pNA <- round(prop.table(table(is.na(data))),3)
      if (is.na(nNA[2]))  nNA <- paste0 (0," (0%)")
      else {
        nNA <- as.numeric (nNA[names(nNA)==TRUE])
        pNA <- as.numeric (pNA[names(pNA)==TRUE])*100
        nNA <- paste0(nNA," (",pNA,"%)")
      }

      myrange <- range(data, na.rm=T)
      myrange <- paste0(myrange[1]," - ",myrange[2])
      # a <- rbind (a,nNA)
      # rownames(a)[-nrow(a)] <- paste0(i,"*")
      a <- cbind (a, nNA, myrange)
      rownames(a) <- paste0(i,"*")
      colnames(a) <- c("valeur", "missing values", "range")
    }
    return(a)
  })
  table_var_quanti <- do.call(rbind,table_var_quanti)
  return (table_var_quanti)
}

describe_all <- function(var, data){
  vec <- data[ ,var]
```

```
  if (any(!is.na(as.numeric(as.character(vec)))) & length(levels(as.factor(vec))) != 2) res <- describe_quantitative(var, data)#génère warning si
character
  else res <- describe_qualitative(var, data)
  return(res)
}
```