

SPECT data script

Damien Legallois et Alain Manrique

30 septembre 2022

Contents

Considérations générales	2
Configuration	2
Fonctions	3
Fonction de désaccentuation	3
Fonction pour tout mettre en minuscule	3
Lecture des données sources	3
Import des données	3
Réorganisation des données	3
Codage du sexe	4
Codage des périodes	4
Champ de recherche et anonymisation	5
Nettoyage	5
Tri des examens	5
Traitement des données	6
Cardiopathie	6
CMD	6
CMI	6
ATCD Infarctus	7
Pontage coronaire	7
PCI	8
Revascularisation	8
Nettoyage	8
Antécédents et caractéristiques	8
Antecedent de Cancer	8
COVID	9
Nettoyage	9
Indications de l'examen	9
Nettoyage	10
FDRCV	10
HTA	10
Diabète	11
Tabagisme	11
Dyslipidémie	11
AOMI	11
Hérédité coronaire	11
Nettoyage	12
Données cliniques	12

BBG	12
FA	12
IMC	12
Nettoyage	13
Traitements	13
Antiaggrégants	13
Anticoagulants	13
Bétabloquants	13
Ivabradine	14
Statines	14
Autres hypolipémians	14
Anti-ald, ARA2 et IEC	14
Diuretiques	14
Anticalciques	14
Insuline	14
ADO	15
Nettoyage	15
Signes fonctionnels, caractérisation de la douleur	15
Douleur thoracique	15
Dyspnée	18
Nettoyage	19
Type de stress	19
Nettoyage	20
Volumes et fonction VG	20
FEVG de stress	20
VTD stress	21
VTS stress	21
FEVG de repos	22
VTD repos	22
VTS repos	22
Nettoyage	23
Perfusion	23
Indication à une coronarographie	23
Perfusion normale	23
Perfusion anormale (ischémie myocardique)	23
Perfusion anormale (infarctus)	25
Nettoyage	25
Probabilité pré-test	25
Export csv	27
Nettoyage	27
Analyse statistique	27
Comparaison périodes	27

Considérations générales

Configuration

On commence par effacer les données en mémoire, puis détermination du répertoire courant

```
rm(list=ls())
setwd("~/Documents/Cardio/Publis-Protocoles/2004 - Data mining SPECT/v1-R_markdown/")
filename <- "scinti_fusion.xls"
```

```
filename_prenoms <- "nat2021.csv"
```

Fonctions

Fonction de désaccentuation

```
Unaccent <- function(text) {  
  text <- gsub("[`^~\\"]", " ", text)  
  text <- iconv(text, to="ASCII//TRANSLIT//IGNORE")  
  text <- gsub("[`^~\\"]", "", text)  
  return(text)  
}
```

Fonction pour tout mettre en minuscule

```
ToutEnMinuscule <- function(input) {  
  gsub(pattern = "\\s", replacement = "", x = input) %>%  
    str_to_lower() %>%  
    {gsub('[:punct:] ]+', '', .)} %>%  
    {iconv(., to="ASCII//TRANSLIT//IGNORE")} %>%  
    {gsub("[`^~\\"]", "", .)} ->  
    output  
}
```

Lecture des données sources

Import des données

Lecture du fichier Excel importé de GERA, et transformation en data.frame pour manipuler plus facilement les colonnes.

```
spect <- as.data.table(read_xls(filename))  
spect <- as.data.frame(spect)
```

On peut ensuite définir un filtre sur le prénom ou le nom, cela permet surtout de travailler sur un morceau de base plus petit, plus pratique en phase de développement. Annoter ces lignes pour utiliser l'ensemble de la base

```
# spect <- subset(spect, Nom %like% "LAI*")  
# spect <- subset(spect, Prénom=="Dominique")
```

Réorganisation des données

On renomme les colonnes importées de GERA. Les champs nom et prénom sont suivi d'un underscore pour signaler le fait qu'il s'agit de champ temporaire. Dans la mesure où la base doit être anonymisée, ces champs devront être détruits.

```
names(spect)[1] <- 'date'  
names(spect)[2] <- 'npp'  
names(spect)[3] <- 'dossier'  
names(spect)[4] <- 'nom_' # _ car temporaire, sera anonymisé  
names(spect)[5] <- 'prenom_' # _ car temporaire, sera anonymisé  
names(spect)[6] <- 'ddn'  
names(spect)[7] <- 'protocole'
```

```
names(spect)[8] <- 'service'
names(spect)[9] <- 'statut'
```

Les colonnes créées par GERA sont ensuite en nombre aléatoire - sans que leur nombre ne suive en apparence une certaine logique. En revanche, les informations qu'elles contiennent sont fondamentales (indication de l'examen, données cliniques, résultats de la scintigraphie myocardique de perfusion, etc.). On va donc renommer une à une toutes ces colonnes en les appelant avec le préfixe `data_spect` au moyen d'une boucle. Puis, on concatène toutes les colonnes qui commencent par ce préfixe `data_spect`. La colonne aux données concaténées est conservée mais toutes les colonnes temporaires sont effacées.

```
pattern = "data_spect"
for (i in 10:ncol(spect)) {
  names(spect)[i] <- paste0(pattern,i)
}
spect$spectdata = do.call(paste, spect[,grepl(pattern, names(spect))])
spect <- spect %>%
  select(-starts_with("data_spect"))
```

Codage du sexe

On importe le fichier csv pour comparer le prénom à l'année de naissance et définir quel est le sexe le plus probable. Ensuite on décharge ce gros fichier.

```
gender_ <- read.csv2(filename_prenoms, sep = ",")
gender_$concatenation <- Unaccent(ToutEnMinuscule(gender_$concatenation))
gender_ <- distinct(gender_)
spect$concatenation <- paste0(year(spect$ddn),Unaccent(ToutEnMinuscule(spect$prenom)))
spect<-merge(x=spect,y=gender_,by="concatenation",all.x=TRUE)
```

Codage des périodes

Codage des périodes, par exemple p2019 pour un examen réalisé en 2019, etc. Puis, on code en fonction du jour de l'année si nous sommes en pré-lockdown, en lockdown, etc. et les niveaux sont ordonnées pour que la séquence soit logique (pré-lockdown, puis lockdown, puis post-lockdown, etc.). Les données sont filtrées pour supprimer les examens qui ne nous intéressent pas et on élimine les classes qui ne nous intéressent pas avec un `droplevel`.

```
as.POSIXct(spect$date, format="%Y-%m-%d") %>%
  year() %>%
  {paste0("p", .)} %>%
  as.factor() ->
  spect$periode
spect$date <- floor_date(spect$date, "day")

date_lockdown <- ymd("2020-03-17")

spect$groupe <- time_length(
  interval(start = date_lockdown, end = spect$date), unit = "days"
)

spect$groupe<- as.factor(case_when(
  spect$groupe> 0 & spect$groupe< 56 ~ "lockdown",
  spect$groupe> 55 & spect$groupe< 111 ~ "post+1",
  spect$groupe> 110 & spect$groupe< 166 ~ "post+2",
  spect$groupe> 165 & spect$groupe< 221 ~ "post+3",
```

```

spect$groupe> 220 & spect$groupe< 276 ~ "post+4",
spect$groupe> -56 & spect$groupe< 1 ~ "pre-1",
spect$groupe> -111 & spect$groupe< -55 ~ "pre-2",
spect$groupe> -166 & spect$groupe< -110 ~ "pre-3",
spect$groupe> -221 & spect$groupe< -165 ~ "pre-4",
TRUE ~ "trash"
))

spect$groupe<- ordered(spect$groupe, levels = c("pre-4", "pre-3", "pre-2", "pre-1", "lockdown", "post+1", "post+2", "post+3", "post+4"))

#spect <- subset(spect, spect$periode != "pNA")
#spect <- subset(spect, spect$groupe != "trash")
spect <- droplevels(spect)

```

Champ de recherche et anonymisation

On met en place un champ nommé `saerch` dans lequel on cherchera le texte. La fonction `ToutEnMinuscule` est appliquée au champ concaténé créé quelques lignes au dessus. Ensuite, on anonymise et on calcule l'âge à partir de la date de naissance et la date de l'examen.

```

spect$search <- ToutEnMinuscule(spect$spectdata)

spect$nom <- substr(spect$nom,1,3)
spect$prenom <- substr(spect$prenom,1,2)
spect$date_exam <- as.Date(spect$date`)

interval(spect$ddn, spect$date_exam) %>% #calcul de l'âge
  {time_length(., "years")} %>%
  floor() %>%
  as.numeric() ->
  spect$age

```

Nettoyage

```

rm(pattern, i, gender_)
spect <- spect %>%
  select(-c(nom_, prenom_, date, npp, dossier, service, statut, concatenation))

```

Tri des examens

On retire les protocoles où l'objectif n'est pas de chercher de l'ischémie.

```

gsub(pattern = "\\s", replacement = "", x = spect$protocole) %>%
  str_to_lower() ->
  spect$protocole_temp
pattern <- "viab|tepscan|amylose|f.e.v.g|os"
spect <- subset(spect, spect$protocole_temp %like% pattern == FALSE)

```

On retire également les patients ayant une cardiopathie congénitale.

```

pattern <- "doublediscord|transposition|vdsystemique"
spect <- subset(spect, spect$search %like% pattern == FALSE)
rm(pattern)

```

Traitement des données

Cardiopathie

Les termes fréquemment retrouvés en majuscule et qui seront bien plus difficiles à identifier si tout est concaténé en minuscule sont identifiés au départ, le texte est ensuite transformé sans accents, sans ponctuation et sans majuscule. On coupe également après le terme acquisition afin de ne pas retrouver des mots évocateurs de cardiopathie dans la conclusion, ce qui ferait - à tort - classer le patient comme étant coronarien alors que ce n'est pas le cas.

```
gsub(pattern = " SCA", replacement = "atcdsyndromecoronarienaigu", x = spect$spectdata) %>%
  {gsub(pattern = " CD", replacement = "coronairedroite", x = .)} %>%
  {gsub(pattern = " IVA", replacement = "interventriculaireanterieure", x = .)} %>%
  {gsub(pattern = " ST\\+", replacement = "infarctus", x = .)} %>%
  ToutEnMinuscule() %>%
  Unaccent() %>%
  {gsub("acquisition.*", "", .)} ->
  spect$search_sca
```

CMD

```
pattern <- "cmd|cardiopathiedilate|cardiomyopathiedilate"
spect$cmd <- ifelse(spect$search %like% pattern == TRUE, "cmd", "no_cmd")
spect$cmd <- ordered(spect$cmd, levels = c("no_cmd", "cmd"))
```

CMI

On recherche soit des patterns évocateurs de CMI ou au contraire en faveur d'une absence d'ATCD CV.

```
pattern <- "suividecmi|suivicmi|scatt|scast|suividunecmi|syndromecoronarienaigu"
spect$cmi <- ifelse(spect$search_sca %like% pattern == TRUE, "CAD", "inconnu")
pattern <- "atcdcv0|atcdcvaucun"
spect$cmi <- ifelse(
  spect$cmi=="inconnu", ifelse(
    spect$search_sca %like% pattern == TRUE, "No_CAD", "inconnu"
  ),
  as.character(spect$cmi)
)
```

S'il y a un terme coro/angioplastie/stent + un nom de vaisseau, alors il existe probablement une CMI. Les mots IVA et CD ont été transformés avant pour éviter de chercher "iva" qu'on retrouverait par exemple dans ivabradine ou rivaroxaban si on ne prenait pas cette précaution. Idem pour la CD. Si on a une coro (1 point) et un nom de vaisseau (1 point) alors il est probable qu'il y ait une CMI si on a 2 points. Pour les patients qui ont encore un statut inconnu quant à l'existence d'une CMI, on considère alors qu'il n'y en a pas, les recherches ayant été vaines.

```
pattern <- "coro|angioplastie|stent"
pattern2 <- "cx|interventriculaire|coronairedroite|circonflexe|diagonale|marginale|retroventriculaire"
cmi0 <- ifelse(spect$search_sca %like% pattern == TRUE, 1, 0)
cmi1 <- ifelse(spect$search_sca %like% pattern2 == TRUE, 1, 0)
spect$cmi <- ifelse(spect$cmi=="inconnu", ifelse(cmi0+cmi1 == 2, "CAD", "inconnu"), as.character(spect$cmi))
spect$cmi <- ifelse(spect$cmi == "inconnu", "No_CAD", as.character(spect$cmi))
spect$cmi <- ordered(spect$cmi, levels = c("No_CAD", "CAD"))
```

ATCD Infarctus

On supprime les termes “décédé(e) d’un” car ces termes ne sont utilisés que pour décrire des ATCD familiaux et ils ont l’inconvénient d’éloigner l’individu (ex: père) et le terme (ex: infarctus), ce qui (i) complique la recherche d’ATCD familiaux, et (ii) risque à tort de qualifier le patient comme ayant un ATCD d’infarctus alors qu’il s’agissait d’un parent. On cherche ensuite un maximum de patterns évocateurs d’un ATCD familial de CMI et on remplace ce terme - hétérogène - par “hereditecor”, ce qui permettra plus facilement de rechercher les ATCD familiaux quand on les classera un peu plus loin.

```
pattern_hereditecor <- "decededun|decedeedun"
spect$search_sca <- gsub(pattern = pattern_hereditecor, replacement = "", x = spect$search_sca)
pattern_hereditecor <- "idmchez|idmpere|idmmere|idmfrere|idmsoeur|stentchez|stentpere|stentmere|stentfrere"
spect$search_sca <- gsub(pattern = pattern_hereditecor,
  replacement = "hereditecor", x = spect$search_sca)
```

Après avoir éliminé les termes qui pouvaient - par erreur - faire dire que le patient a des antécédents d’infarctus, alors on cherche des patterns d’infarctus

```
pattern_AMI <- "infarctus|idm|atcdsyndromecoronarienaigu"
spect$history_MI <- ifelse(
  spect$cmi!="NON", ifelse(
    spect$search_sca %like% pattern_AMI == TRUE, "AMI", "No_AMI"
  ),
  "No_AMI"
)
spect$history_MI <- ordered(spect$history_MI, levels = c("No_AMI", "AMI"))
```

Pontage coronaire

1. On va éviter de rechercher l’existence d’un PAC trop proche d’un apparenté, on va donc chercher à localiser un de ces termes et on va remplacer tout le texte autour par ATCDFAM, afin de détruire de notre chaîne de recherche un PAC chez un membre de la famille. Si la recherche n’a rien donné, on peut continuer à chercher dans le champ search_sca qu’on a utilisé précédemment.

```
ToutEnMinuscule(spect$spectdata) %>%
  Unaccent() %>%
  {gsub(pattern = "pere|mere|soeur|frere", replacement = "apparenté", x = .)} ->
  spect$search_CABG

pere <- as.data.frame(str_locate(spect$search_CABG, "apparenté"))
str_sub(spect$search_CABG, pere$start -15, pere$start +35) <- "ATCDFAM"
spect$search_CABG <- ifelse(
  is.na(spect$search_CABG), spect$search_sca, spect$search_CABG
)
```

2. On supprime les autres situations où on peut croiser les lettres “pac”

```
pattern_pac <- "pacemaker|pacilitax|capacit|noncompaction"
pattern_CABG <- "ponte|doublepontage|triplepontage|pontagemig|pontagemid|pontagemammaire|pontageaortocoronaire"
spect$search_CABG <- gsub(pattern = pattern_pac, replacement = "", x = spect$search_CABG)
spect$cabg <- ifelse(
  spect$search_CABG %like% pattern_CABG == TRUE, "CABG", "No_CABG"
)
spect$cabg <- ordered(spect$cabg, levels = c("No_CABG", "CABG"))
```

PCI

On fait la recherche sur le champ search_CABG car le rationnel est le même : éviter que des stents chez le père ne fasse croire que le patient index a déjà eu des stents. Ensuite, on supprime les termes de revascularisation périphérique qui peuvent aussi être confusant. Enfin, on qualifie l'existence ou non d'une revascularisation percutanée.

```
pattern_aomi<- "angioplastiefemora|angioplastieilia|stentfemora|stentilia|aomistent"
pattern_PCI <- "stent|angioplastie|thrombolyse|fibrinolyse"
spect$search_CABG <- gsub(pattern = pattern_aomi, replacement = "", x = spect$search_CABG)
spect$pci <- ifelse(
  spect$cmi != "NON", ifelse(
    spect$search_CABG %like% pattern_PCI == TRUE, "PCI", "No_PCI"
  ),
  "No_PCI"
)
spect$pci <- ordered(spect$pci, levels = c("No_PCI", "PCI"))
```

Revascularisation

Tout simplement fonction de la présence ou non d'une angioplastie et/ou d'un PAC. S'il existe une revascularisation coronaire, on force le champ CMI si la CMI n'avait pas été identifié au préalable.

```
spect$revascularization <- ifelse(
  spect$cabg == "No_CABG", ifelse(
    spect$pci == "No_PCI", "no-revascularization", "revascularization"
  ),
  "revascularization"
)

spect$cmi <- ifelse(
  spect$revascularization == "revascularization", "CAD", as.character(spect$cmi)
)
spect$cmi <- ordered(spect$cmi, levels = c("No_CAD", "CAD"))

spect$revascularization <- ordered(spect$revascularization, levels = c("no-revascularization", "revascularization"))
```

Nettoyage

Le camp search_sca n'est pas éliminé à ce stade car il resservira plus tard.

```
rm(pere,cmi0,cmi1)
rm(list = apropos("pattern"))
spect <- spect %>%
  select(-c(search_CABG,protocole_temp))
```

Antécédents et caractéristiques

Antecedent de Cancer

```
pattern <- "adenocar|adk|epidermoid|tumeur|cancer|remission|chimio|radiother|metastase|carcinome|lymphome"
spect$cancer <- ifelse(spect$search %like% pattern == TRUE, "cancer", "no_cancer")
spect$cancer <- ordered(spect$cancer, levels = c("no_cancer", "cancer"))
```


COVID

```
pattern <- "covid|sarscov2"
spect$covid <- ifelse(spect$search %like% pattern == TRUE, "covid", "no_covid")
spect$covid <- ordered(spect$covid, levels = c("no_covid", "covid"))
```

Nettoyage

```
rm(list = apropos("pattern"))
```

Indications de l'examen

On commence par les bilans pré-op car on peut être en bilan pré-op avant tout sur CMI

```
pattern_indication_preop <- "bilanpreop|bilanpredilatation|avantchirurgie|prechirurgie|avanttransplantation"
spect$indication_exam <- ifelse(
  spect$search %like% pattern_indication_preop == TRUE, "Bilan pre-op", "inconnu"
)
```

Puis, si c'est écrit "recherche d'ischémie" => recherche d'ischémie

```
pattern_indication_diag <- "recherchedischemie|rechercheischemie|rechercheduneischemie|depistageischemie"
spect$indication_exam <- ifelse(
  spect$indication_exam=="inconnu", ifelse(
    spect$search %like% pattern_indication_diag == TRUE, "Recherche d'ischemie", "inconnu"
  ),
  spect$indication_exam
)
```

Sinon, c'est un suivi de CMI si on retrouve ce pattern de texte ou si les patients ont une CMI connue et que l'indication n'est pas encore classée

```
pattern_indication_suividecmi <- "suividecmi|suivicmi|suividunecmi"
spect$indication_exam <- ifelse(
  spect$indication_exam=="inconnu", ifelse(
    spect$search %like% pattern_indication_suividecmi == TRUE, "Suivi de CMI", "inconnu"
  ),
  spect$indication_exam
)

spect$indication_exam <- ifelse(
  spect$indication_exam=="inconnu", ifelse(
    spect$cmi == "CAD", "Suivi de CMI", "inconnu"
  ),
  spect$indication_exam
)
```

Indications particulières. Sinon, recherche d'ischémie.

```
pattern_indication_diag <- "viabilite"
spect$indication_exam <- ifelse(
  spect$indication_exam=="inconnu", ifelse(
    spect$search %like% pattern_indication_diag == TRUE, "Viabilite", "inconnu"
  ),
  spect$indication_exam
)
```

```

pattern_indication_diag <- "amylose"
spect$indication_exam <- ifelse(
  spect$indication_exam=="inconnu", ifelse(
    spect$search %like% pattern_indication_diag == TRUE,"suspicion amylose ATTR","inconnu"
  ),
  spect$indication_exam
)

pattern_indication_diag <- "asymptomatique|depistage|coroscaner|bilandedyspnee|evaluationdelareserve|r
spect$indication_exam <- ifelse(
  spect$indication_exam=="inconnu", ifelse(
    spect$search %like% pattern_indication_diag == TRUE,"Recherche d'ischemie","inconnu"
  ),
  spect$indication_exam
)

pattern_indication_diag <- "sarcoidose"
spect$indication_exam <- ifelse(
  spect$indication_exam=="inconnu", ifelse(
    spect$search %like% pattern_indication_diag == TRUE,"Sarcoidose","inconnu"
  ),
  spect$indication_exam
)

spect$indication_exam <- ordered(spect$indication_exam, levels = c("Suivi de CMI", "Recherche d'ischemie

```

Nettoyage

```
rm(list = apropos("pattern"))
```

FDRCV

On commence par remplacer quelques abréviations qui ne seront pas facile à retrouver par la suite. Ensuite, on va dérouler les FDRCV un à un en commençant systématiquement par rechercher des termes négatifs.

```

gsub(pattern = " HCL| HCT", replacement = "dyslipidemie", x = spect$spectdata) %>%
  {gsub(pattern = " DNID| DT2| DID| DT1", replacement = "diabete", x = .)} %>%
  ToutEnMinuscule() ->
  spect$search_fdrcv

```

HTA

```

spect$search_fdrcv <- str_remove(spect$search_fdrcv, "pasdhta|nihta")
pattern_HTA <- "hta"
spect$Hypertension <- ifelse(
  spect$search_fdrcv %like% pattern_HTA == TRUE,"HTN","No_HTN"
)
spect$Hypertension <- ordered(spect$Hypertension, levels = c("No_HTN", "HTN"))
spect$CVRF <- as.numeric(ifelse(
  spect$Hypertension == "HTN", 1, 0
))

```

Diabète

```
spect$search_fdrvc <- str_remove(spect$search_fdrvc, "pasdediabetes|nidiabetes")
pattern_Diabetes <- "diabete"
spect$diabetes <- ifelse(
  spect$search_fdrvc %like% pattern_Diabetes == TRUE, "Diabetes", "No_Diabetes"
)
spect$diabetes <- ordered(spect$diabetes, levels = c("No_Diabetes", "Diabetes"))
spect$CVRf <- as.numeric(ifelse(
  spect$diabetes == "Diabetes", spect$CVRf+1, spect$CVRf
))
```

Tabagisme

```
spect$search_fdrvc <- str_remove(spect$search_fdrvc, "pasdetaba|nitaba")
pattern_Tabacco <- "tabagisme|tabac"
spect$tabacco <- ifelse(
  spect$search_fdrvc %like% pattern_Tabacco == TRUE, "Tabacco", "No_Tabacco"
)
spect$tabacco <- ordered(spect$tabacco, levels = c("No_Tabacco", "Tabacco"))
spect$CVRf <- as.numeric(ifelse(
  spect$tabacco == "Tabacco", spect$CVRf+1, spect$CVRf
))
```

Dyslipidémie

```
spect$search_fdrvc <- str_remove(spect$search_fdrvc, "pasedyslipidemie|nidyslipi|nihyperchol")
pattern_HCT <- "dyslipidemie|cholesterol|hypertriglycerid"
spect$Dyslipidemia <- ifelse(
  spect$search_fdrvc %like% pattern_HCT == TRUE, "Dyslipidemia", "No_Dyslipidemia"
)
spect$Dyslipidemia <- ordered(spect$Dyslipidemia, levels = c("No_Dyslipidemia", "Dyslipidemia"))
spect$CVRf <- as.numeric(ifelse(
  spect$Dyslipidemia == "Dyslipidemia", spect$CVRf+1, spect$CVRf
))
```

AOMI

```
spect$search_fdrvc <- gsub(pattern = "iaominime", replacement = "IA", x = spect$search_fdrvc)
pattern_AOMI <- "aomi|carotid|endarteriec|iliaque|aaa|claudication|pontagefemoro|pontageaortofemo|pontag
spect$PAR <- ifelse(
  spect$search_fdrvc %like% pattern_AOMI == TRUE,"PAD","No_PAD"
)
spect$PAR <- ordered(spect$PAR, levels = c("No PAD", "PAD"))
```

Hérédité coronaire

Il faut transformer le terme “coronarienne” qui est utilisé exclusivement pour hérédité coronarienne, on le change en “hereditacor”, qui sera utilisé plus loin pour reconnaître une hérédité coronarienne. C’est pour cette recherche qu’on a gardé `search_sca` jusque là.

```
spect$search_sca <- gsub(pattern = "pasdhereditecoronarienne|nihereditecoronarienne", replacement = "",
spect$search_sca <- gsub(pattern = "coronarienne", replacement = "heredite", x = spect$search_sca)
```

```

pattern_heredite <- "heredite"
spect$familial_history_cad <- ifelse(
  spect$search_sca %like% pattern_heredite == TRUE, "Familial_history_CAD", "No_FH_CAD"
)
spect$familial_history_cad <- ordered(spect$familial_history_cad, levels = c("No_FH_CAD", "Familial_his
spect$CVRF <- as.numeric(ifelse(
  spect$familial_history_cad == "Familial_history_CAD", spect$CVRF+1, spect$CVRF
))

```

Nettoyage

```

rm(list = apropos("pattern"))
spect <- spect %>%
  select(-c(search_sca, search_fdrcv))

```

Données cliniques

BBG

```

pattern_BBG <- "bbg|blocdebranchegauche|blocgauche"
spect$LBBB <- ifelse(
  spect$search %like% pattern_BBG == TRUE, "LBBB", "No_LBBB"
)
spect$LBBB <- ordered(spect$LBBB, levels = c("No_LBBB", "LBBB"))

```

FA

```

pattern_FA <- "acfa|faperman|faparoX|fapersis|flutter|fibrillationa"
gsub(pattern = " FA ", replacement = "faparoX", x = spect$spectdata) %>%
  ToutEnMinuscule() %>%
  Unaccent() %>%
  {ifelse(. %like% pattern_FA == TRUE, "AF", "No_AF")} ->
  spect$AF
spect$AF <- ordered(spect$AF, levels = c("No_AF", "AF"))

```

IMC

La valeur est limitée à 2 chiffres.

```

str_to_lower(gsub(pattern = "\\s", replacement = "", x = spect$search)) %>%
  {gsub(pattern = ":", replacement = "", x = .)} %>%
  {gsub(".*imc", "", .)} %>%
  {gsub("patient.*", "", .)} %>%
  {gsub("recher.*", "", .)} %>%
  {gsub("suivi.*", "", .)} %>%
  {gsub("bilan.*", "", .)} %>%
  {gsub("kg.*", "", .)} %>%
  {gsub(pattern = "a", replacement = "", x = .)} %>%
  {substr(., 1, 2)} %>%
  as.character() %>%
  as.numeric() ->
  spect$BMI

```

```
## Warning in str_to_lower(gsub(pattern = "\\s", replacement = "", x =
## spect$search)) %>% : NAs introduits lors de la conversion automatique

spect$overweight <- ifelse(spect$BMI >= 30, "obesity", ifelse(spect$BMI < 25, "no_overweight", "overwei
spect$overweight <- ordered(spect$overweight, levels = c("no_overweight", "overweight", "obesity"))

spect$CVRF <- as.numeric(ifelse(
  spect$overweight != "no_overweight", spect$CVRF+1, spect$CVRF
))
```

Nettoyage

```
rm(list = apropos("pattern"))
```

Traitements

Il faut couper à partir des résultats pour ne pas qu'un traitement proposé en conclusion soit considéré comme présent au début. Il faut aussi garder une trace de quelques termes comme IEC, pour ne pas être confondu avec scintigraphiecardiaque.

```
gsub(pattern = "K160|K75|AAP", replacement = "aspirine", x = spect$spectdata) %>%
  {gsub(pattern = "AVK|AOD|NACO", replacement = "warfarin", x = .)} %>%
  {gsub(pattern = "IEC|ARA2|ARAI", replacement = "entresto", x = .)} %>%
  {gsub("acquisition.*", "", .)} %>%
  ToutEnMinuscule() %>%
  Unaccent() ->
  spect$ttt
```

Antiagrégants

```
pattern_AAP <- "aspirine|kardegic|K160|K75|duoplavin|plavix|aspegic|AAP|clopidogrel|acetylsali|brilique
spect$aap <- ifelse(
  spect$ttt %like% pattern_AAP == TRUE, "AAP", "No_AAP"
)
spect$aap <- ordered(spect$aap, levels = c("No_AAP", "AAP"))
```

Anticoagulants

```
pattern_Antico <- "coumadine|eliquis|naco|xarelto|previscan|sintro|warfarin|avk|pradaxa|dabigatran|xaban
spect$antico <- ifelse(
  spect$ttt %like% pattern_Antico == TRUE, "Antico", "No_Antico"
)
spect$antico <- ordered(spect$antico, levels = c("No_Antico", "Antico"))
```

Bétabloquants

```
pattern_BB <- "olol|tenormine|cardentiel|cardensiel|kredex|temerit|bisocel|seloken|lodoz|detensiel|sectr
spect$bb <- ifelse(
  spect$ttt %like% pattern_BB == TRUE, "BB", "No_BB"
)
spect$bb <- ordered(spect$bb, levels = c("No_BB", "BB"))
```

Ivabradine

```
pattern_ivabradine <- "ivabradine|procoralan"
spect$ivabradine <- ifelse(
  spect$ttt %like% pattern_ivabradine == TRUE, "ivabradine", "No_ivabradine"
)
spect$ivabradine <- ordered(spect$ivabradine, levels = c("No_ivabradine", "ivabradine"))
```

Statines

```
pattern_statine <- "statine|tahor|crestor|zocor|lodales|inegy"
spect$statine <- ifelse(
  spect$ttt %like% pattern_statine == TRUE, "statine", "No_statine"
)
spect$statine <- ordered(spect$statine, levels = c("No_statine", "statine"))
```

Autres hypolipémiants

```
pattern_hypolipemiant <- "ezetrol|liptruzet|ezetimib|inegy|lipanthyl|fibrate"
spect$autre_hypolipemiant <- ifelse(
  spect$ttt %like% pattern_hypolipemiant == TRUE, "hypolipemiant", "No_hypolipemiant"
)
spect$autre_hypolipemiant <- ordered(spect$autre_hypolipemiant, levels = c("No_hypolipemiant", "hypolipemiant"))
```

Anti-aldo, ARA2 et IEC

```
pattern_IEC <- "sacubitril|entresto|sartan|pril|micardis|dactone|spirono|eplerenone|rasilez|aliskiren|a
spect$iec <- ifelse(
  spect$ttt %like% pattern_IEC == TRUE, "ACEI", "No_IEC"
)
spect$iec <- ordered(spect$iec, levels = c("No_IEC", "ACEI"))
```

Diuretiques

```
pattern_diuretiques <- "diuretique|lasilix|esidrex|coaprovel|hyperium|furosemide|indapamide|hydrochloro
spect$diuretiques <- ifelse(
  spect$ttt %like% pattern_diuretiques == TRUE, "diuretics", "No_diuretic"
)
spect$diuretiques <- ordered(spect$diuretiques, levels = c("No_diuretic", "diuretics"))
```

Anticalciques

```
pattern_ICa <- "loxen|zanidip|lercan|amlor|dipine|coveram|exforge|verapamil|isoptine|zanextra|iperten|t
spect$ica <- ifelse(
  spect$ttt %like% pattern_ICa == TRUE, "Ca-blockers", "No_Ca-blockers"
)
spect$ica <- ordered(spect$ica, levels = c("No_Ca-blockers", "Ca-blockers"))
```

Insuline

Pour l'insuline et pour les ADO, on considère que les si les patients ont ce type de traitement, alors, ils sont diabétiques.

```
spect$ttt <- gsub(pattern = "noninsulinor", replacement = "", x = spect$ttt)
pattern_insuline <- "insuline|lantus|novorapid|humalog|insulinor"
spect$insuline <- ifelse(
  spect$ttt %like% pattern_insuline == TRUE, "insuline", "No_insuline"
)
spect$insuline <- ordered(spect$insuline, levels = c("No_insuline", "insuline"))
spect$diabetes <- ifelse(
  spect$insuline == "insuline", "Diabetes", as.character(spect$diabetes)
)
```

ADO

```
pattern_ADO <- "metformine|diamicron|glucophage|xelevia|gliptine|gliclazide|stagid|januvia|gavus"
spect$ado <- ifelse(
  spect$ttt %like% pattern_ADO == TRUE, "OAD", "No_OAD"
)
spect$ado <- ordered(spect$ado, levels = c("No_OAD", "OAD"))
spect$diabetes <- ifelse(
  spect$ado == "OAD", "Diabetes", as.character(spect$diabetes)
)
spect$diabetes <- ordered(spect$diabetes, levels = c("No_Diabetes", "Diabetes"))
```

Nettoyage

```
rm(list = apropos("pattern"))
spect <- spect %>%
  select(-c(ttt))
```

Signes fonctionnels, caractérisation de la douleur

Douleur thoracique

```
gsub(pattern = " DT", replacement = " douleur thoracique", x = spect$spectdata) %>%
  ToutEnMinuscul() %>%
  Unaccent() %>%
  {gsub(pattern = "precordialgie", replacement = "douleurthoracique", x = .)} %>%
  #on doit mettre le critère repos/effort après le caractère constrictive sinon, cela ne sera pas trouvé
  {gsub(pattern = "reposconstrictive", replacement = "constrictivederepos", x = .)} %>%
  {gsub(pattern = "effortconstrictive", replacement = "oppressionthoraciquealeffort", x = .)} ->
  spect$search_sf
```

On cherche en premier les “pas de douleur thoracique” ou “asymptomatique”

```
pattern <- "pasdedouleurthoracique|pasdedouleursthoracique|asymptomatique|pasdangor|absencededouleurtho
spect$dt <- ifelse(spect$search_sf %like% pattern == TRUE, "aucune", "inconnu")
```

Ensuite on va chercher les douleurs thoraciques, les douleurs précordiales sachant que parfois il existe un ou plusieurs petits mots entre douleur et thoracique, il y a aussi la problématique des pluriels, parfois douleurs est au pluriel, parfois c’est thoracique, parfois c’est les deux et parfois tout est au singulier. On enlève donc tous les S, et on cherche si on a “thora” (permet d’avoir thoracique et thorax) ou “cordial” (permet d’avoir précordial et précordial avec ou sans e). On coupe après acquisition, sinon l’évocation d’appeler le SAMU en cas de douleurs dans les résultats est suffisant pour faire dire qu’il y a une DT, même s’il n’y en a pas.

```
str_remove_all(spect$search_sf, "s") %>%
  {gsub("acquisition.*", "", .)} ->
  spect$search_sf

pattern <- "ymptomatologieangoreue|ymptomatologieangineue|oppreionthoracique|oppreionretroternale|genet
spect$dt <- ifelse(
  spect$dt=="inconnu",ifelse(
    spect$search_sf %like% pattern == TRUE,"oui","inconnu"
  ),
  as.character(spect$dt)
)
```

Ensuite, on prend les 25 caractères qui suivent “douleur” et on va chercher des racines de mots évocatrices d’une douleur thoracique

```
pattern <- "thora|troternal|cordial|later"
gsub(".*douleur", "", spect$search_sf) %>%
  {str_sub(.,1,25)} %>%
  {ifelse(spect$dt=="inconnu",ifelse(. %like% pattern == TRUE,"oui","inconnu"),as.character(spect$dt))}
  spect$dt
spect$dt <- ordered(spect$dt, levels = c("oui", "aucune", "inconnu"))
```

On cherche ensuite un lien avec l’effort. On prend les caractères qui suivent douleur parce que la description du lien avec l’effort est souvent un peu plus loin que 25 caractères. Le problème, c’est qu’on va couper la chaîne à “douleur” et qu’il va prendre la dernière occurrence et qu’on va perdre de l’info notamment sur le caractère lié à l’effort. L’astuce consiste à remplacer le terme équivalent à douleur par douleurs avec un s. On cherche ensuite à couper ce terme avec un s (on le trouvera facilement car les s ont été supprimés). On remplace par le terme angor, ce qui permettra plus tard de statuer sur la caractère typique ou non. On considère en effet que angor = typique, donc il faut en garder une trace.

```
pattern_constrictif <- "ymptomatologieangoreue|ymptomatologieangineue|oppreionthoracique|oppreionretrot
pattern_douleur_non_constrictif <- "genethoracique"
gsub(pattern = pattern_constrictif, replacement = "douleursangor", x = spect$search_sf) %>%
  {gsub(pattern = pattern_douleur_non_constrictif, replacement = "douleurs", x = .)} ->
  spect$search_sf
```

On supprime le terme d’épreuve d’effort pour ne pas classifier le caractère “à l’effort” d’une douleur sur ce terme. Ensuite, comme expliqué au-dessus, on coupe à “douleurs” sinon “douleur”.

```
pattern_epreuve_effort <- "epreuveeffort|testdeffort"
spect$search_sf <- gsub(pattern = pattern_epreuve_effort, replacement = "", x = spect$search_sf)

pattern <- "douleurs"
spect$search_sf <- ifelse(
  spect$search_sf %like% pattern == TRUE, gsub(".*douleurs", "", spect$search_sf), gsub(".*douleur", ""
)
```

Le caractère lié à l’effort ne vaut que s’il y a douleur, donc il faut qu’il existe une douleur pour ensuite se poser la question si elle est liée ou non à l’effort. Souvent, il est écrit une dyspnée d’effort, ce qui ne nous intéresse pas ici car cela sera extrait dans la classe NYHA. On supprime tout ce qui suit le terme “dyspnée” et après “acquisition”.

```
gsub("dypnee.*", "", spect$search_sf) %>%
{gsub("acquisition.*", "", .)} ->
spect$search_sf
```

On commence par les patterns qui éliminent le lien à l’effort, plus longues. Puis, on cherche le caractère lié à

l'effort de la douleur. Si les patterns sont étranges, c'est parce que les "s" ont été supprimés. Les patients qui restent n'ont pas de douleurs d'effort.

```
pattern <- "anlienavecleffort|anrapportavecleffort|anrapportavecleeffort|nonlieealeffort|nonlieeauxeffort"
spect$effort <- ifelse(
  spect$dt=="oui", ifelse(
    spect$search_sf %like% pattern == TRUE, "non", "inconnu"
  ),
  "pas de DT"
)

pattern <- "lamarche|effort"
spect$effort <- ifelse(
  spect$dt=="oui", ifelse(
    spect$effort=="inconnu", ifelse(
      spect$search_sf %like% pattern == TRUE, "oui", "inconnu"
    ),
    spect$effort
  ),
  "pas de DT"
)

spect$effort <- ifelse(
  spect$dt=="oui", ifelse(
    spect$effort=="inconnu", "non précisé", spect$effort
  ),
  "pas de DT"
)

spect$effort <- ordered(spect$effort, levels = c("oui", "non", "non précisé", "pas de DT"))
```

On recherche le type "constrictif" de la symptomatologie. On reprend le concept du lien avec les douleurs, on va chercher que ce qu'il y a entre douleur et dyspnée. On regarde s'il y a le mot atypique, sinon typique ou des caractéristiques en faveur. S'il y a le mot atypique, il faut le repérer puis le supprimer pour ne pas tagguer typique.

```
pattern_atypique <- "atypique"
spect$typique <- ifelse(
  spect$dt=="oui", ifelse(
    spect$search_sf %like% pattern_atypique == TRUE, "atypique", "inconnu"
  ),
  "pas de DT"
)

spect$search_sf <- gsub(pattern = pattern_atypique, replacement = "", x = spect$search_sf)

pattern_typique <- "typique|contractive|angor|serrement|efforttrinit"
spect$typique <- ifelse(
  spect$dt=="oui", ifelse(
    spect$typique=="inconnu", ifelse(
      spect$search_sf %like% pattern_typique == TRUE, "typique", "non précisé"
    ),
    spect$typique
  ),
  "pas de DT"
)
```

```
)
```

```
spect$typique <- ordered(spect$typique, levels = c("typique", "atypique", "non précisé", "pas de DT"))
```

Dyspnée

Pour différencier les NYHA I-II des NYHA III, il faut que l'on garde une trace du tiret sinon ça fera nyhaïï dans les deux cas. Les tirets et les "à" sont remplacés par des "z", qu'on pourra retrouver facilement.

```
gsub(pattern = "-", replacement = "z", x = spect$search) %>%  
  ToutEnMinuscule() %>%  
  {gsub(pattern = "à", replacement = "z", x = .)} %>%  
  Unaccent() ->  
  spect$search_sf
```

```
pattern_dyspnee = "dyspneestade|dyspneestade"
```

```
spect$search_sf <- gsub(pattern = pattern_dyspnee, replacement = "dyspneenyha", x = spect$search_sf)
```

On cherche en premier les "pas de dyspnée ou aucune dyspnée" puis ensuite on coupe la chaîne à nyha puis on cherche les patterns qui décrivent la classe NYHA. On part de la chaîne la plus longue à la plus courte.

```
pattern_dyspnee <- "pasdedyspnee|aucunedyspnee|sandsypnee|nidedyspnee"  
spect$dyspnee <- ifelse(spect$search_sf %like% pattern_dyspnee == TRUE, "aucune", "inconnu")  
spect$search_sf <- str_remove(spect$search_sf, "pasdedyspnee")  
spect$search_sf <- str_remove(spect$search_sf, "aucunedyspnee")  
spect$search_sf <- str_remove(spect$search_sf, "sandsypnee")  
spect$search_sf <- str_remove(spect$search_sf, "nidedyspnee")
```

```
spect$search_sf <- gsub(".*nyha", "", spect$search_sf)
```

```
temp_nyha6 <- as.character(str_sub(spect$search_sf, 1, 6))  
temp_nyha5 <- as.character(str_sub(spect$search_sf, 1, 5))  
temp_nyha4 <- as.character(str_sub(spect$search_sf, 1, 4))  
temp_nyha3 <- as.character(str_sub(spect$search_sf, 1, 3))  
temp_nyha2 <- as.character(str_sub(spect$search_sf, 1, 2))  
temp_nyha1 <- as.character(str_sub(spect$search_sf, 1, 1))
```

```
temp_dyspnee <- case_when(  
  temp_nyha6 == "iiziii" ~ "NYHA II-III",  
  temp_nyha6 == "iiiziv" ~ "NYHA III-IV",  
  temp_nyha5 == "iiiiii" ~ "NYHA II-III",  
  temp_nyha5 == "iiiiv" ~ "NYHA III-IV",  
  temp_nyha4 == "izii" ~ "NYHA I-II",  
  temp_nyha3 == "iii" ~ "NYHA III",  
  temp_nyha3 == "1z2" ~ "NYHA I-II",  
  temp_nyha3 == "2z3" ~ "NYHA II-III",  
  temp_nyha3 == "3z4" ~ "NYHA III-IV",  
  temp_nyha2 == "12" ~ "NYHA I-II",  
  temp_nyha2 == "23" ~ "NYHA II-III",  
  temp_nyha2 == "34" ~ "NYHA III-IV",  
  temp_nyha2 == "ii" ~ "NYHA II",  
  temp_nyha2 == "iv" ~ "NYHA IV",  
  temp_nyha1 == "1" ~ "NYHA I",  
  temp_nyha1 == "2" ~ "NYHA II",  
  temp_nyha1 == "3" ~ "NYHA III",
```

```

temp_nyhal == "4" ~ "NYHA IV",
temp_nyhal == "I" ~ "NYHA I",
TRUE ~ "inconnu"
)

spect$dyspnee <- ifelse(
  spect$dyspnee=="inconnu", ifelse(
    temp_dyspnee!="inconnu", temp_dyspnee, "inconnu"
  ),
  spect$dyspnee
)

```

Pour finir avec la dyspnée, parfois il n'est pas écrit NYHA, simplement dyspnée à la marche ou à l'effort. On va regarder dans les 25 caractères qui suivent

```

gsub(".*dyspnee", "", spect$search_sf) %>%
  {str_sub(.,1,25)} ->
  spect$search_sf

pattern_dyspnee_marche <- "zlamarche"
spect$dyspnee <- ifelse(
  spect$dyspnee=="inconnu", ifelse(
    spect$search_sf %like% pattern_dyspnee_marche == TRUE, "à la marche", "inconnu"
  ),
  spect$dyspnee
)

pattern_dyspnee_oui <- "associee|effort"
spect$dyspnee <- ifelse(
  spect$dyspnee=="inconnu", ifelse(
    spect$search_sf %like% pattern_dyspnee_oui == TRUE, "oui", "inconnu"
  ),
  spect$dyspnee
)

spect$dyspnee <- ifelse(spect$dyspnee=="inconnu", "aucune", spect$dyspnee)

```

Nettoyage

```

rm(list = apropos("pattern"))
rm(list = apropos("temp"))
spect <- spect %>%
  select(-c(search_sf))

```

Type de stress

On commence par dire que toutes les dynamiques sont faites au regadenoson, puis on cherche dans l'ordre les indices cochés (x) puis les autres (il faut respecter cet ordre sinon c'est le bordel). De cette façon, il n'y a plus de "rest only", juste quelques "inconnus" liés à des fautes de frappes. Enfin, il faut remplacer "wmax" par autre chose qui se termine pas par "x"

```

pattern <- "Coeur DYN"
spect$type_stress <- ifelse(
  spect$protocole %like% pattern == TRUE, "Regadenoson 1", "inconnu"
)

```

```

)

ToutEnMinuscule(spect$search) %>%
  Unaccent() %>%
  {gsub(pattern = "wmax", replacement = "watt", x = .)} ->
  spect$temp_ee

temp_ee2 <- case_when(
  spect$temp_ee %like% "xtestmixte" ~ "Combined",
  spect$temp_ee %like% "xpersantine" ~ "Dipyridamole",
  spect$temp_ee %like% "xregadenoson" ~ "Regadenoson",
  spect$temp_ee %like% "xeffort" ~ "Exercise",
  spect$temp_ee %like% "epreuevestresstestmixteassociantpersantine056mgkgetepreueveffortsurbicyclette" ~ "Combined",
  spect$temp_ee %like% "epreuevestresspersantinerealiseepar" ~ "Dipyridamole",
  spect$temp_ee %like% "epreuevestresspersantine056mgkgrealiseepar" ~ "Dipyridamole",
  spect$temp_ee %like% "epreuevestressregadenosonrealiseepar" ~ "Regadenoson",
  spect$temp_ee %like% "epreueveffortsurbicycletteexplorant" ~ "Exercise",
  spect$temp_ee %like% "epreuevestresstestmixteassociantpersantine056mgkgetepreueveffortsurbicyclette" ~ "Combined",
  TRUE ~ "inconnu"
)

spect$type_stress <- ifelse(
  spect$type_stress=="inconnu", ifelse(
    temp_ee2 != "inconnu", temp_ee2, "inconnu"
  ),
  spect$type_stress
)

spect$type_stress <- ordered(spect$type_stress, levels = c("Exercise", "Dipyridamole", "Regadenoson", "Combined"))

```

Nettoyage

```

rm(list = apropos("pattern"))
rm(list = apropos("temp"))
spect <- spect %>%
  select(-c(temp_ee))

```

Volumes et fonction VG

FEVG de stress

Il faut enlever tout ce qu'il y a avant 99mtc car sinon, on travaille sur la technique et non sur les résultats. Il faut garder uniquement ce qu'il y a entre "acquisition de stress" et "acquisition de repos". Il peut arriver que les patients n'aient pas eu de repos, dans ce cas, on va jusqu'à la fin.

```

ToutEnMinuscule(spect$search) %>%
  Unaccent() %>%
  {gsub(".*99mtc", "", .)} ->
  spect$vg_stress

start_cut <- as.data.table(str_locate(spect$vg_stress, "acquisitiondestress"))
end_cut <- as.data.table(str_locate(spect$vg_stress, "acquisitionderepos"))
spect$vg_stress_temp <- str_sub(spect$vg_stress, start_cut$start, end_cut$end)

```

```

spect$vg_stress <- ifelse(
  is.na(spect$vg_stress_temp), str_sub(spect$vg_stress, start_cut$start), spect$vg_stress_temp
)

# On attaque pour la FEVG de stress
FEVG <- as.data.table(str_locate(spect$vg_stress, "fevg"))
spect$fevg_stress <- str_sub(spect$vg_stress, FEVG$end)

# On vires progressivement les scories qu'on remplace par un "." car il y en a déjà beaucoup
spect$fevg_stress <- gsub(pattern = "gdestress:", replacement = ".", x = spect$fevg_stress)
spect$fevg_stress <- gsub(pattern = "g:", replacement = ".", x = spect$fevg_stress)
spect$fevg_stress <- gsub(pattern = ">", replacement = ".", x = spect$fevg_stress)

# On cherche le tag "." et on se met de côté la localisation du "."
tag_EFstr <- as.data.table(str_locate(spect$fevg_stress, "."))

# Ensuite on va chercher les deux caractères qui sont après et on le sauvegarde dans le champ FEVG_stress
spect$fevg_stress <- as.numeric(as.character(
  str_sub(spect$fevg_stress, tag_EFstr$start+1, tag_EFstr$start+2))
)

```

VTD stress

```

VTD <- as.data.table(str_locate(spect$vg_stress, "vtd"))
spect$vtd_stress <- str_sub(spect$vg_stress, VTD$end)

# on cherche le tag "d=" et on se met de côté sa localisation puis on récupère les chiffres
tag_VTDstr <- as.data.table(str_locate(spect$vtd_stress, "d"))
spect$vtd_stress <- ifelse(
  str_sub(spect$vtd_stress, tag_VTDstr$end+4, tag_VTDstr$end+4)=="m",
  as.numeric(as.character(str_sub(spect$vtd_stress, tag_VTDstr$end+1, tag_VTDstr$end+3))),
  as.numeric(as.character(str_sub(spect$vtd_stress, tag_VTDstr$end+1, tag_VTDstr$end+2)))
)

```

VTS stress

```

VTS <- as.data.table(str_locate(spect$vg_stress, "vts"))
spect$vts_stress <- str_sub(spect$vg_stress, VTS$end)

# on cherche le tag "s=" et on se met de côté sa localisation puis on récupère les chiffres
tag_VTSstr <- as.data.table(str_locate(spect$vts_stress, "s"))
spect$vts_stress <- ifelse(
  str_sub(spect$vts_stress, tag_VTSstr$end+3, tag_VTSstr$end+3)=="m",
  as.numeric(as.character(str_sub(spect$vts_stress, tag_VTSstr$end+1, tag_VTSstr$end+2))),
  ifelse(str_sub(spect$vts_stress, tag_VTSstr$end+4, tag_VTSstr$end+4)=="m",
    as.numeric(as.character(str_sub(spect$vts_stress, tag_VTSstr$end+1, tag_VTSstr$end+3))),
    as.numeric(as.character(str_sub(spect$vts_stress, tag_VTSstr$end+1, tag_VTSstr$end+1)))
  )
)

```

FEVG de repos

Il faut enlever tout ce qu'il y a avant 99mtc car sinon, on travaille sur la technique et non sur les résultats. Puis, enlever ce qu'il y a avant "FEVG de repos".

```
ToutEnMinuscule(spect$search) %>%
  Unaccent() %>%
  {gsub(pattern = "fevgderepos", replacement = "fevgaurepos", x = .)} ->
  spect$vg_repos

FEVG_repos <- as.data.table(str_locate(spect$vg_repos, "fevgaurepos"))
spect$vg_repos <- str_sub(spect$vg_repos, FEVG_repos$end)

# le tag était pour le stress un ".". je le remplace par un "s" de "...fevgderepos"
# on vire progressivement les scories qu'on remplace par un "s"
spect$vg_repos <- gsub(pattern = "s:", replacement = "s", x = spect$vg_repos)
spect$vg_repos <- gsub(pattern = ">", replacement = "s", x = spect$vg_repos)

# on cherche le tag "s" et on se met de côté la localisation du "s"
tag_EFrest <- as.data.table(str_locate(spect$vg_repos, "s"))

# ensuite on va chercher les deux caractères qui sont après et on le sauvegarde dans le champ FEVG_stre
spect$fevg_repos <- as.numeric(as.character(
  str_sub(spect$vg_repos, tag_EFrest$start+1, tag_EFrest$start+2))
)
```

VTd repos

```
VTDr <- as.data.table(str_locate(spect$vg_repos, "vtd"))
spect$vtd_repos <- str_sub(spect$vg_repos, VTDr$end)

# on cherche le tag "d=" et on se met de côté sa localisation puis on récupère les chiffres
tag_VTDrep <- as.data.table(str_locate(spect$vtd_repos, "d"))
spect$vtd_repos <- ifelse(
  str_sub(spect$vtd_repos, tag_VTDrep$end+4, tag_VTDrep$end+4)=="m",
  as.numeric(as.character(str_sub(spect$vtd_repos, tag_VTDrep$end+1, tag_VTDrep$end+3))),
  as.numeric(as.character(str_sub(spect$vtd_repos, tag_VTDrep$end+1, tag_VTDrep$end+2)))
)
```

VTs repos

```
VTsr <- as.data.table(str_locate(spect$vg_repos, "vts"))
spect$vts_repos <- str_sub(spect$vg_repos, VTsr$end)

# on cherche le tag "s=" et on se met de côté sa localisation puis on récupère les chiffres
tag_VTSrep <- as.data.table(str_locate(spect$vts_repos, "s"))
spect$vts_repos <- ifelse(
  str_sub(spect$vts_repos, tag_VTSrep$end+3, tag_VTSrep$end+3)=="m",
  as.numeric(as.character(str_sub(spect$vts_repos, tag_VTSrep$end+1, tag_VTSrep$end+2))), ifelse(
    str_sub(spect$vts_repos, tag_VTSrep$end+4, tag_VTSrep$end+4)=="m",
    as.numeric(as.character(str_sub(spect$vts_repos, tag_VTSrep$end+1, tag_VTSrep$end+1))))
)
```

Nettoyage

```
rm(end_cut, FEVG, FEVG_repos, start_cut, VTD, VTDr, VTS, VTSr)
rm(list = apropos("tag_"))
spect <- spect %>%
  select(-c(vg_stress, vg_stress_temp, vg_repos))
```

Perfusion

On recrée un champ de recherche en enlevant tout ce qu'il y a avant "tomo", ainsi on a bien les résultats. Parfois le terme tomo n'est pas présent, dans ce cas on coupe à acquisition, en 2e intention.

```
ToutEnMinuscule(spect$search) %>%
  Unaccent() %>%
  {gsub(".*acquisition", "", .)} %>%
  {gsub(".*tomo", "", .)} ->
  spect$conclusion
```

Indication à une coronarographie

```
pattern <- "coronarographie"
spect$sent_to_angio <- ifelse(spect$conclusion %like% pattern == TRUE, "sent-to-angiography", "no")
```

Perfusion normale

```
pattern <- "scintigraphiedeperfusionnormale|scintigraphiemyocardiquenormale|scintigraphiedynamiquenorma
spect$spect_perfusion <- ifelse(
  spect$conclusion %like% pattern == TRUE, "Normal perfusion", "inconnu"
)
pattern <- "attenuation"
spect$spect_perfusion <- ifelse(
  spect$spect_perfusion == "inconnu", ifelse(
    spect$conclusion %like% pattern == TRUE, "Attenuation artefact", "Abnormal perf"
  ),
  spect$spect_perfusion
)
```

Perfusion anormale (ischémie myocardique)

On va chercher "ischémie myocardique", on coupe et on garde les 50 caractères qui suivent. Mais parfois il y a deux fois le terme ischémie et si on coupe au 2e, on perd de l'information, il faut donc couper au premier, ce qui n'est pas possible avec la syntaxe que l'on utilise. On va donc ruser en utilisant "sub" au lieu de "gsub" qui va remplacer la première occurrence d'ischémie en ischemia puis on coupe à ischemia. C'est vraiment pas élégant mais je ne vois pas de façon plus simple d'arriver au résultat recherché

```
spect$temp_perfusion <- sub("ischémie", "ischemia", spect$conclusion)
pattern <- "sansischemia"
spect$temp_perfusion2 <- ifelse(
  spect$temp_perfusion %like% pattern == TRUE, "non", "a definir"
)
spect$temp_perfusion <- sub(".*ischemia", "", spect$temp_perfusion)
spect$temp_perfusion <- str_sub(spect$temp_perfusion, 1, 50)
```

On commence à quantifier les défauts perfusionnels. On commence par remplacer les coquetteries du style "sur 1 segment" ou "d'1 segment" au lieu de "de 1 segment". Puis on quantifie l'étendue de l'ischémie en fonction

des caractères qui précèdent le mot “segment”. Si NA, on met 0, il n’y a pas d’ischémie. On finit aussi par définir ce qu’est une ischémie étendue.

```
nb_segment <- as.data.frame(str_locate(spect$temp_perfusion, "segment"))
spect$nb_segment_isch <- ifelse(
  spect$temp_perfusion2 == "a definir",
  str_sub(spect$temp_perfusion, nb_segment$start-7, nb_segment$start-1), 0
)
spect$nb_segment_isch2 <- ifelse(
  spect$temp_perfusion2 == "a definir",
  str_sub(spect$temp_perfusion, nb_segment$start-3, nb_segment$start-1), 0
)
spect$nb_segment_isch3 <- ifelse(
  spect$temp_perfusion2 == "a definir",
  str_sub(spect$temp_perfusion, nb_segment$start-2, nb_segment$start-1), 0
)
spect$nb_segment_isch4 <- ifelse(
  spect$temp_perfusion2 == "a definir",
  str_sub(spect$temp_perfusion, nb_segment$start-1, nb_segment$start-1), 0
)

ischemie <- case_when(
  spect$nb_segment_isch == "oinsde1" ~ "0.5",
  spect$nb_segment_isch == "moinsd1" ~ "0.5",
  spect$nb_segment_isch == "oinsde2" ~ "1.5",
  spect$nb_segment_isch == "oinsde3" ~ "2.5",
  spect$nb_segment_isch == "oinsde4" ~ "3.5",
  spect$nb_segment_isch == "oinsde5" ~ "4.5",
  spect$nb_segment_isch == "oinsde6" ~ "5.5",
  spect$nb_segment_isch == "oinsde7" ~ "6.5",
  spect$nb_segment_isch == "oinsde8" ~ "7.5",
  spect$nb_segment_isch2 == "1a2" ~ "1.5",
  spect$nb_segment_isch2 == "2a3" ~ "2.5",
  spect$nb_segment_isch2 == "3a4" ~ "3.5",
  spect$nb_segment_isch2 == "4a5" ~ "4.5",
  spect$nb_segment_isch2 == "5a6" ~ "5.5",
  spect$nb_segment_isch2 == "6a7" ~ "6.5",
  spect$nb_segment_isch2 == "7a8" ~ "7.5",
  spect$nb_segment_isch2 == "8a9" ~ "8.5",
  spect$nb_segment_isch3 == "12" ~ "1.5",
  spect$nb_segment_isch3 == "23" ~ "2.5",
  spect$nb_segment_isch3 == "34" ~ "3.5",
  spect$nb_segment_isch3 == "45" ~ "4.5",
  spect$nb_segment_isch3 == "56" ~ "5.5",
  spect$nb_segment_isch3 == "67" ~ "6.5",
  spect$nb_segment_isch3 == "78" ~ "7.5",
  spect$nb_segment_isch3 == "89" ~ "8.5",
  spect$nb_segment_isch4 == "1" ~ "1",
  spect$nb_segment_isch4 == "2" ~ "2",
  spect$nb_segment_isch4 == "3" ~ "3",
  spect$nb_segment_isch4 == "4" ~ "4",
  spect$nb_segment_isch4 == "5" ~ "5",
  spect$nb_segment_isch4 == "6" ~ "6",
  spect$nb_segment_isch4 == "7" ~ "7",
```



```

spect$nb_segment_isch4 == "8" ~ "8",
spect$nb_segment_isch4 == "9" ~ "9"
)
spect$nb_segment_isch <- as.numeric(as.character(ischemie))
spect$nb_segment_isch[is.na(spect$nb_segment_isch)] <- 0

spect$signif_ischemia <- ifelse(
  spect$nb_segment_isch == 0, "pas d'ischémie", ifelse(
    spect$nb_segment_isch > 1.5, "ischémie >= 2 segments", "ischémie <2 segments"
  )
)

```

Perfusion anormale (infarctus)

On va chercher “infarctus”, on coupe et on garde les 20 caractères qui suivent.

```

gsub(pattern = "necrosemyocardique", replacement = "infarctus", x = spect$conclusion) %>%
  {gsub(".*infarctus", "", .)} %>%
  {str_sub(.,1,30)} %>%
  {gsub(pattern = "d1", replacement = "01", x = .)} %>%
  # au cas où on a un truc du genre 2/17 segments
  {gsub(pattern = "17", replacement = "segment", x = .)}->
  temp_infarctus

nb_segment <- as.data.table(str_locate(temp_infarctus, "segment"))

as.numeric(as.character(str_sub(temp_infarctus, nb_segment$start-1, nb_segment$start-1))) %>%
  {ifelse(==0,10,.)} %>%
  {ifelse(is.na(.), 0, .)} ->
  spect$nb_segment_IDM

```

Nettoyage

```

rm(ischemie, pattern, temp_infarctus, nb_segment)
spect <- spect %>%
  select(-c(protocole, search, conclusion, temp_perfusion, temp_perfusion2, nb_segment_isch2, nb_segmen

```

Probabilité pré-test

On cherche à multiplier entre eux les 3 paramètres : sexe, age et symptomatologie. Dès lors que chaque paramètre est associé à un nombre premier, le produit des 3 permettra d’identifier dans quelle case des recos on se trouve et donc d’aller chercher la proba pré-test associée. Les coronariens déjà connus sont exclus (le produit fait 0).

```

proba_temp <- ifelse(spect$indication_exam=="Suivi de CMI",0,1)
proba_temp <- ifelse(proba_temp != 0, ifelse(spect$gender=="M",1,ifelse(spect$gender=="F",2,0)), 0)

age_par_dizaine <- floor((spect$age)*(0.1))

proba_temp2 <- ifelse(
  age_par_dizaine==3, 3, ifelse(age_par_dizaine==4,5,ifelse(
    age_par_dizaine==5, 7, ifelse(
      age_par_dizaine==6, 11, ifelse(

```

```

        age_par_dizaine>6, 13, 0)
    )
  )
)

proba_temp3 <- ifelse(
  spect$typique=="typique", 17, ifelse(
    spect$typique=="atypique", 19, 23))
# Si pas de douleur thoracique mais dyspnée, alors on attribue le chiffre premier 29
# @Alain, les NYHA I sont considérés comme dyspnéique... souvent on considère les SF à NYHA II+
proba_temp3 <- ifelse(
  proba_temp3==23, ifelse(
    spect$dyspnee!="aucune", 29, 23
  ),
  proba_temp3
)

proba_overall <- proba_temp * proba_temp2 * proba_temp3

spect$proba <- case_when(
  proba_overall == 51 ~ 0.03,
  proba_overall == 85 ~ 0.22,
  proba_overall == 119 ~ 0.32,
  proba_overall == 187 ~ 0.44,
  proba_overall == 221 ~ 0.52,
  proba_overall == 102 ~ 0.05,
  proba_overall == 170 ~ 0.10,
  proba_overall == 238 ~ 0.13,
  proba_overall == 374 ~ 0.16,
  proba_overall == 442 ~ 0.27,
  proba_overall == 57 ~ 0.04,
  proba_overall == 95 ~ 0.10,
  proba_overall == 133 ~ 0.17,
  proba_overall == 209 ~ 0.26,
  proba_overall == 247 ~ 0.34,
  proba_overall == 114 ~ 0.03,
  proba_overall == 190 ~ 0.06,
  proba_overall == 266 ~ 0.06,
  proba_overall == 418 ~ 0.11,
  proba_overall == 494 ~ 0.19,
  proba_overall == 69 ~ 0.01,
  proba_overall == 115 ~ 0.03,
  proba_overall == 161 ~ 0.11,
  proba_overall == 253 ~ 0.22,
  proba_overall == 299 ~ 0.24,
  proba_overall == 138 ~ 0.01,
  proba_overall == 230 ~ 0.02,
  proba_overall == 322 ~ 0.03,
  proba_overall == 506 ~ 0.06,
  proba_overall == 598 ~ 0.10,
  proba_overall == 87 ~ 0.00,
  proba_overall == 145 ~ 0.12,

```

```

proba_overall == 203 ~ 0.20,
proba_overall == 319 ~ 0.27,
proba_overall == 377 ~ 0.32,
proba_overall == 174 ~ 0.03,
proba_overall == 290 ~ 0.03,
proba_overall == 406 ~ 0.09,
proba_overall == 638 ~ 0.14,
proba_overall == 754 ~ 0.12,
# sinon, 99 pour ceux qui ne sont pas analysables pour les exclure après
# (car NA ici ne fonctionne pas)
proba_overall == 0 ~ 99
)

spect$proba <- as.numeric(as.character(ifelse(
  spect$proba==99, NA, spect$proba))
)

spect$PPT_range <- ifelse(
  spect$proba>0.15,">0.15",ifelse(
    spect$proba>0.05,"0.05-0.15",ifelse(
      spect$proba>-0.01,"<0.05",NA
    )
  )
)
)
spect$PPT_range <- ordered(spect$PPT_range, levels = c("<0.05", "0.05-0.15", ">0.15"))

```

Export csv

```
write.csv2(spect, 'resultats_coeur_juil_21.csv', row.names=TRUE)
```

Nettoyage

```
rm(spect_proba, table, age_par_dizaine)
rm(list = apropos("proba_"))
```

Analyse statistique

On drop les champs qui ne pourront pas être entrés dans une table

```

spect <- spect %>%
  select(-c(spectdata, nom, prenom, date_exam))
spect <- droplevels(spect)

```

Comparaison périodes

```

varsToFactor <- c("gender")
spect[varsToFactor] <- lapply(spect[varsToFactor], factor)

# Create a variable list
vars <- c("age", "gender", "cancer", "LBBB", "AF", "BMI", "overweight", "PAR", "Hypertension", "diabetes",
"familial_history_cad", "CVRF", "cmd", "cmi", "history_MI", "cabg", "pci", "revascularization", "dt", "
"typique", "dyspnee", "indication_exam",

```

```

"aap", "antico", "bb", "ivabradine", "statine", "autre_hypolipemiant", "iec", "diuretiques", "ica", "in
"vts_stress", "fevg_repos", "vtd_repos", "vts_repos", "spect_perfusion", "nb_segment_isch", "signif_isch
table <- CreateTableOne(vars = vars, data = spect, strata=c("groupe"), includeNA = FALSE)
table <- print(table, printToggle = FALSE, noSpaces = TRUE)

kbl(table) %>%
kable_styling(latex_options = c("striped", "scale_down"))

```

	pre-4	pre-3	pre-2	pre-1	lockdown	post+1	post+2	post+3	post+4	trash	p	test
n	504	617	574	625	220	499	446	626	584	2386		
age (mean (SD))	67.27 (10.91)	65.94 (10.70)	66.51 (10.54)	67.38 (9.85)	66.03 (10.61)	67.32 (9.89)	67.43 (10.47)	67.74 (10.63)	66.90 (10.10)	66.85 (10.63)	0.082	
gender = M (%)	298 (60.9)	380 (63.7)	369 (65.7)	383 (63.9)	153 (70.5)	319 (66.2)	287 (66.9)	390 (63.8)	378 (66.8)	1515 (65.5)	0.377	
cancer = cancer (%)	30 (6.0)	28 (4.5)	19 (3.3)	29 (4.6)	17 (7.7)	38 (7.6)	33 (7.4)	69 (11.0)	28 (4.8)	96 (4.0)	<0.001	
LBBB = LBBB (%)	30 (6.0)	24 (3.9)	30 (5.2)	34 (5.4)	14 (6.4)	26 (5.2)	27 (6.1)	36 (5.8)	34 (5.8)	159 (6.7)	0.504	
AF = AF (%)	60 (11.9)	60 (9.7)	61 (10.6)	77 (12.3)	21 (9.5)	64 (12.8)	49 (11.0)	61 (9.7)	64 (11.0)	254 (10.6)	0.730	
BMI (mean (SD))	27.69 (5.51)	28.62 (5.52)	28.58 (5.59)	28.59 (5.96)	28.66 (6.16)	28.58 (5.70)	28.97 (6.29)	28.13 (5.47)	28.43 (5.86)	28.52 (5.73)	0.083	
overweight (%)											0.543	
no_overweight	141 (28.9)	135 (22.8)	127 (22.6)	145 (23.7)	53 (25.6)	117 (24.3)	103 (25.6)	144 (26.3)	139 (24.6)	516 (23.5)		
overweight	181 (37.1)	231 (39.0)	233 (41.4)	233 (38.1)	82 (39.6)	184 (38.3)	142 (35.2)	218 (39.8)	225 (39.8)	899 (40.9)		
obesity	166 (34.0)	227 (38.3)	203 (36.1)	234 (38.2)	72 (34.8)	180 (37.4)	158 (39.2)	186 (33.9)	202 (35.7)	781 (35.6)		
PAR = PAD (%)	69 (13.7)	77 (12.5)	83 (14.5)	88 (14.1)	17 (7.7)	57 (11.4)	70 (15.7)	107 (17.1)	76 (13.0)	354 (14.8)	0.025	
Hypertension = HTN (%)	329 (65.3)	423 (68.6)	366 (63.8)	392 (62.7)	148 (67.3)	320 (64.1)	202 (58.7)	370 (59.1)	396 (67.8)	1583 (66.8)	0.001	
diabetes = Diabetes (%)	151 (30.0)	189 (30.6)	180 (31.4)	199 (31.8)	64 (29.1)	158 (31.7)	141 (31.6)	197 (31.5)	176 (30.1)	723 (30.3)	0.995	
tabacco = Tabacco (%)	262 (52.0)	370 (60.0)	303 (52.8)	288 (46.1)	104 (47.3)	250 (50.1)	203 (45.5)	254 (40.6)	248 (42.5)	1267 (53.1)	<0.001	
Dyslipidemia = Dyslipidemia (%)	305 (60.5)	364 (59.0)	347 (60.5)	360 (57.6)	120 (54.5)	276 (55.3)	232 (52.0)	309 (49.4)	340 (58.2)	1456 (61.0)	<0.001	
familial_history_cad = Familial_history_CAD (%)	164 (32.5)	205 (33.2)	136 (23.7)	124 (19.8)	49 (22.3)	118 (23.6)	67 (15.0)	81 (12.9)	115 (19.7)	650 (27.2)	<0.001	
CVRF (mean (SD))	3.14 (1.41)	3.34 (1.31)	3.10 (1.31)	2.96 (1.27)	2.99 (1.27)	3.01 (1.27)	2.83 (1.26)	2.78 (1.32)	2.96 (1.30)	3.21 (1.31)	<0.001	
cmd = cmd (%)	3 (0.6)	14 (2.3)	16 (2.8)	12 (1.9)	5 (2.3)	13 (2.6)	14 (3.1)	20 (3.2)	17 (2.9)	45 (1.9)	0.109	
cmd = CAD (%)	234 (46.4)	259 (42.0)	259 (45.1)	247 (39.5)	96 (43.6)	207 (41.5)	180 (40.4)	273 (43.6)	245 (42.0)	1047 (43.9)	0.375	
history_MI = AMI (%)	115 (22.8)	151 (24.5)	133 (23.2)	115 (18.4)	47 (21.4)	93 (18.6)	88 (19.7)	107 (17.1)	128 (21.9)	479 (20.1)	0.029	
cabg = CABG (%)	18 (3.6)	19 (3.1)	20 (3.5)	23 (3.7)	12 (5.5)	7 (1.4)	11 (2.5)	13 (2.1)	25 (4.3)	126 (5.3)	<0.001	
pai = PCI (%)	149 (29.6)	176 (28.5)	174 (30.3)	155 (24.8)	64 (29.1)	142 (28.5)	127 (28.5)	173 (27.6)	136 (23.3)	691 (29.0)	0.155	
revascularization = revascularization (%)	159 (31.5)	189 (30.6)	187 (32.6)	171 (27.4)	72 (32.7)	144 (28.9)	131 (29.4)	180 (28.8)	149 (25.5)	752 (31.5)	0.111	
dt (%)											<0.001	
oui	133 (26.4)	177 (28.7)	152 (26.5)	198 (31.7)	85 (38.6)	143 (28.7)	117 (26.2)	148 (23.6)	148 (25.3)	688 (28.8)		
aucune	343 (68.1)	400 (64.8)	393 (68.5)	412 (65.9)	124 (56.4)	339 (67.9)	317 (71.1)	459 (73.3)	412 (70.5)	1522 (63.8)		
inconnu	28 (5.5)	40 (6.5)	29 (5.1)	15 (2.4)	11 (5.0)	17 (3.4)	12 (2.7)	19 (3.0)	24 (4.1)	176 (7.4)		
effort (%)											<0.001	
oui	35 (6.9)	56 (9.1)	28 (4.9)	42 (6.7)	17 (7.7)	34 (6.8)	27 (6.1)	26 (4.2)	44 (7.5)	174 (7.3)		
non	68 (13.5)	77 (12.5)	75 (13.1)	79 (12.6)	36 (16.4)	43 (8.6)	36 (8.1)	45 (7.2)	51 (8.7)	310 (13.0)		
non précisé	30 (6.0)	44 (7.1)	49 (8.5)	77 (12.3)	32 (14.5)	66 (13.2)	54 (12.1)	77 (12.3)	53 (9.1)	204 (8.5)		
pas de DT	371 (73.6)	440 (71.3)	422 (73.5)	427 (68.3)	135 (61.4)	356 (71.3)	329 (73.8)	478 (76.4)	436 (74.7)	1698 (71.2)		
typique (%)											<0.001	
typique	61 (12.1)	80 (13.0)	51 (8.9)	69 (11.0)	33 (15.0)	29 (5.8)	22 (4.9)	33 (5.3)	42 (7.2)	241 (10.1)		
atypique	31 (6.2)	27 (4.4)	16 (2.8)	30 (4.8)	21 (9.5)	43 (8.6)	26 (5.8)	41 (6.5)	57 (9.8)	203 (8.5)		
non précisé	41 (8.1)	70 (11.3)	85 (14.8)	99 (15.8)	31 (14.1)	71 (14.2)	69 (15.5)	74 (11.8)	49 (8.4)	244 (10.2)		
pas de DT	371 (73.6)	440 (71.3)	422 (73.5)	427 (68.3)	135 (61.4)	356 (71.3)	329 (73.8)	478 (76.4)	436 (74.7)	1698 (71.2)		
dyspnée (%)											<0.001	
aucune	290 (57.5)	337 (54.6)	308 (53.7)	346 (55.4)	120 (54.5)	256 (51.3)	260 (58.3)	398 (63.6)	341 (58.4)	1214 (50.9)		
NYHA I	33 (6.5)	43 (7.0)	20 (3.5)	19 (3.0)	5 (2.3)	22 (4.4)	27 (6.1)	36 (5.8)	23 (3.9)	312 (13.1)		
NYHA I-II	0 (0.0)	3 (0.5)	0 (0.0)	0 (0.0)	0 (0.0)	2 (0.4)	1 (0.2)	7 (1.1)	0 (0.0)	23 (1.0)		
NYHA II	107 (21.2)	151 (24.5)	151 (26.3)	153 (24.5)	74 (33.6)	129 (25.9)	107 (24.0)	105 (16.8)	105 (18.0)	543 (22.8)		
NYHA II-III	15 (3.0)	3 (0.5)	5 (0.9)	3 (0.5)	0 (0.0)	8 (1.6)	6 (1.3)	19 (3.0)	16 (2.7)	55 (2.3)		
NYHA III	44 (8.7)	62 (10.0)	54 (9.4)	68 (10.9)	20 (9.1)	48 (9.6)	23 (5.2)	32 (5.1)	69 (11.8)	175 (7.3)		
NYHA III-IV	2 (0.4)	2 (0.3)	1 (0.2)	1 (0.2)	0 (0.0)	3 (0.6)	3 (0.7)	0 (0.0)	0 (0.0)	7 (0.3)		
NYHA IV	0 (0.0)	1 (0.2)	0 (0.0)	3 (0.5)	0 (0.0)	1 (0.2)	1 (0.2)	1 (0.2)	2 (0.3)	7 (0.3)		
oui	13 (2.6)	15 (2.4)	35 (6.1)	32 (5.1)	1 (0.5)	30 (6.0)	18 (4.0)	28 (4.5)	28 (4.8)	50 (2.1)		
indication_exam (%)											0.584	
Suivi de CMI	175 (37.1)	207 (35.6)	196 (35.8)	188 (32.0)	84 (39.6)	170 (35.8)	136 (32.1)	201 (34.3)	184 (32.9)	780 (34.7)		
Recherche d'ischémie	295 (62.5)	372 (63.9)	350 (64.0)	399 (68.0)	127 (59.9)	301 (63.4)	285 (67.2)	384 (65.5)	371 (66.2)	1449 (64.4)		
Sarcoidose	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)		
inconnu	2 (0.4)	3 (0.5)	1 (0.2)	0 (0.0)	1 (0.5)	3 (0.6)	3 (0.7)	1 (0.2)	5 (0.9)	18 (0.8)		
asp = AAP (%)	296 (58.7)	349 (56.6)	317 (55.2)	330 (52.8)	123 (55.9)	267 (53.5)	194 (43.5)	254 (40.6)	301 (51.5)	1356 (56.8)	<0.001	
antico = Antico (%)	62 (12.3)	75 (12.2)	85 (14.8)	87 (13.9)	28 (12.7)	73 (14.6)	71 (15.9)	79 (12.6)	81 (13.9)	309 (13.0)	0.689	
bb = BB (%)	247 (49.0)	287 (46.5)	280 (48.8)	286 (45.8)	110 (50.0)	262 (52.5)	174 (39.0)	207 (33.1)	238 (40.8)	1156 (48.4)	<0.001	
ivabradine = ivabradine (%)	3 (0.6)	8 (1.3)	13 (2.3)	9 (1.4)	6 (2.7)	5 (1.0)	8 (1.8)	8 (1.3)	5 (0.9)	31 (1.3)	0.265	
statine = statine (%)	283 (56.2)	310 (50.2)	336 (58.5)	312 (49.9)	122 (55.5)	289 (57.9)	208 (46.6)	238 (38.0)	273 (46.7)	1292 (54.1)	<0.001	
autre_hypolipemiant = hypolipemiant (%)	59 (11.7)	80 (13.0)	65 (11.3)	73 (11.7)	18 (8.2)	71 (14.2)	41 (9.2)	54 (8.6)	67 (11.5)	255 (10.7)	0.078	
iec = ACEI (%)	291 (57.7)	317 (51.4)	304 (53.0)	339 (54.2)	117 (53.2)	279 (55.9)	204 (45.7)	265 (42.3)	308 (52.7)	1272 (53.3)	<0.001	
diuretiques = diuretics (%)	104 (20.6)	120 (19.4)	115 (20.0)	112 (17.9)	34 (15.5)	106 (21.2)	88 (19.7)	103 (16.5)	103 (17.6)	518 (21.7)	0.056	
ica = Ca-blockers (%)	130 (25.8)	139 (22.5)	146 (25.4)	150 (24.0)	62 (28.2)	122 (24.4)	97 (21.7)	123 (19.6)	147 (25.2)	596 (25.0)	0.134	
insuline = insuline (%)	57 (11.3)	46 (7.5)	33 (5.7)	28 (4.5)	12 (5.5)	47 (9.4)	44 (9.9)	60 (9.6)	37 (6.3)	217 (9.1)	<0.001	
ado = OAD (%)	59 (11.7)	111 (18.0)	72 (12.5)	50 (8.0)	11 (5.0)	82 (16.4)	82 (18.4)	120 (19.2)	76 (13.0)	289 (12.1)	<0.001	
type_stress (%)											<0.001	
Exercice	34 (7.0)	34 (5.6)	16 (3.0)	40 (6.6)	3 (1.5)	28 (5.9)	9 (2.1)	12 (2.1)	12 (2.3)	130 (5.7)		
Dipyridamole	296 (60.9)	334 (55.0)	326 (60.1)	308 (51.1)	16 (8.0)	195 (41.3)	285 (65.8)	384 (65.9)	282 (53.5)	1280 (56.0)		
Regadenoson	83 (17.1)	150 (24.7)	128 (23.6)	139 (23.1)	153 (76.5)	193 (40.9)	98 (22.6)	135 (23.2)	152 (28.8)	580 (25.4)		
Combined	63 (13.0)	71 (11.7)	45 (8.3)	90 (14.9)	12 (6.0)	40 (8.5)	30 (6.9)	26 (4.5)	50 (9.5)	223 (9.8)		
inconnu	10 (2.1)	18 (3.0)	27 (5.0)	26 (4.3)	16 (8.0)	16 (3.4)	11 (2.5)	26 (4.5)	31 (5.9)	74 (3.2)		
fvsg_stress (mean (SD))	70.67 (12.25)	70.36 (12.45)	69.65 (12.10)	70.21 (12.43)	68.73 (12.57)	70.00 (11.39)	70.41 (12.58)	70.35 (11.05)	70.17 (11.90)	70.12 (12.67)	0.811	
vtd_stress (mean (SD))	88.40 (33.74)	88.88 (39.70)	90.57 (36.17)	90.56 (35.74)	97.10 (42.53)	90.40 (33.75)	94.40 (43.81)	88.29 (34.60)	91.25 (35.02)	91.72 (36.30)	0.041	
vts_stress (mean (SD))	29.45 (24.40)	30.69 (30.99)	31.37 (26.64)	31.82 (27.77)	35.54 (32.61)	30.02 (22.57)	31.09 (31.87)	28.38 (23.13)	30.79 (23.79)	32.10 (27.44)	0.045	
fvsg_repos (mean (SD))	65.56 (11.61)	64.88 (12.77)	65.36 (12.27)	64.02 (12.94)	63.48 (11.93)	65.46 (11.74)	64.73 (13.45)	64.58 (11.97)	64.04 (12.29)	65.34 (12.80)	0.303	
vtd_repos (mean (SD))	96.64 (34.82)	98.71 (46.37)	96.89 (39.97)	96.30 (38.28)	101.13 (42.35)	100.68 (36.78)	103.27 (49.11)	98.62 (36.47)	99.38 (37.87)	98.03 (41.98)	0.448	
vts_repos (mean (SD))	36.13 (24.87)	39.32 (37.85)	37.46 (31.31)	38.60 (29.87)	40.63 (33.06)	37.83 (27.27)	41.69 (41.81)	37.32 (25.87)	38.81 (27.02)	38.49 (35.53)	0.673	
spect_perfusion (%)											<0.001	
Abnormal perf	264 (52.4)	309 (50.1)	306 (53.3)	338 (54.1)	130 (59.1)	285 (57.1)	240 (53.8)	260 (41.5)	235 (40.2)	1353 (56.7)		
Attenuation artefact	0 (0.0)	0 (0.0)	1 (0.2)	0 (0.0)	0 (0.0)	0 (0.0)	5 (1.1)	14 (2.2)	4 (0.7)	5 (0.2)		
Normal perfusion	240 (47.6)	308 (49.9)	267 (46.5)	287 (45.9)	90 (40.9)	214 (42.9)	201 (45.1)	352 (56.2)	345 (59.1)	1028 (43.1)		
nb_segment_isch (mean (SD))	0.80 (1.19)	0.83 (1.23)	0.82 (1.19)	0.78 (1.13)	0.68 (0.94)	0.86 (1.15)	0.80 (1.15)	0.54 (0.98)	0.53 (0.98)	0.82 (1.15)	<0.001	
signif_ischemia (%)											<0.001	
ischémie <2 segments	124 (24.6)	132 (21.4)	151 (26.3)	170 (27.2)	60 (27.3)	155 (31.1)	107 (24.0)	111 (17.7)	103 (17.6)	639 (26.8)		
ischémie >= 2 segments	95 (18.8)	130 (21.1)	107 (18.6)	112 (17.9)	37 (16.8)	94 (18.8)	89 (20.0)	86 (13.7)	74 (12.7)	479 (20.1)		
pas d'ischémie	285 (56.5)	355 (57.5)	316 (55.1)	343 (54.9)	123 (55.9)	250 (50.1)	250 (50.1)	429 (68.5)	407 (69.7)	1268 (53.1)		
nb_segment_IDM (mean (SD))	0.39 (1.28)	0.42 (1.27)	0.45 (1.31)	0.44 (1.28)	0.53 (1.38)	0.37 (1.						