Effectif_freeze_Regroup

Manon Santrisse

2022-10-18

Récupération du jeu Effectifs_freeze.Rdata

```
load('../Donnees/Effectifs_freeze.Rdata')
head(effectif)
```

```
##   Num.Ctr.Coll.Anonyme Code.Grp.Assures Produit.Anonyme
## 1                    1           ACTIFS               1
## 2                    1           ACTIFS               1
## 3                    1           ACTIFS               1
## 4                    1           ACTIFS               1
## 5                    1           ACTIFS               1
## 6                    1           ACTIFS               1
##   Date.Effet.Adhesion.Contrat.Coll Date.Effet.Radiation.Contrat.Coll Code.Ape
## 1                       2022-01-01                       2050-12-31    2932Z
## 2                       2022-01-01                       2050-12-31    2932Z
## 3                       2022-01-01                       2050-12-31    2932Z
## 4                       2022-01-01                       2050-12-31    2932Z
## 5                       2022-01-01                       2050-12-31    2932Z
## 6                       2022-01-01                       2050-12-31    2932Z
##   Departement REGROUP_PROD_1 REGROUP_PROD_4.Anonyme REGROUP_PROD_5
## 1        <NA>          ERCAC                      1           BASE
## 2        <NA>          ERCAC                      1           BASE
## 3        <NA>          ERCAC                      1           BASE
## 4        <NA>          ERCAC                      1           BASE
## 5        <NA>          ERCAC                      1           BASE
## 6        <NA>          ERCAC                      1           BASE
##   Num.Personne.Anonyme Num.Ctr.Indiv.Anonyme Date.Effet.Adhesion.Num.Personne
## 1                    1                     1                       2022-01-01
## 2                    2                     2                       2022-01-01
## 3                    3                     3                       2022-01-01
## 4                    4                     4                       2022-01-01
## 5                    5                     5                       2022-01-01
## 6                    6                     6                       2022-01-01
##   Date.Effet.Radiation.Num.Personne Type.Assure Sexe Date.Naissance R.NR
## 1                        2050-12-31      ENFANT    F     2005-03-02    R
## 2                        2050-12-31      ENFANT    F     2010-05-21    R
## 3                        2050-12-31      ENFANT    M     1999-01-08    R
## 4                        2050-12-31      CONJOI    F     1991-08-27    R
## 5                        2050-12-31      ENFANT    F     2017-12-02    R
## 6                        2050-12-31      ASSPRI    M     1980-10-10    R
##   Lien.entreprise.Anonyme Numéro.contrat.coll.Grands.comptes Indexation.2018
## 1                       1                               TRUE              NA
```

```
## 2                       1                                TRUE              NA
## 3                       1                                TRUE              NA
## 4                       1                                TRUE              NA
## 5                       1                                TRUE              NA
## 6                       1                                TRUE              NA
##   Indexation.2019 Indexation.2020 Indexation.2021 Indexation.2022
## 1              NA              NA              NA              NA
## 2              NA              NA              NA              NA
## 3              NA              NA              NA              NA
## 4              NA              NA              NA              NA
## 5              NA              NA              NA              NA
## 6              NA              NA              NA              NA
##   Indexation.2023 Renégo.2020 Renégo.2021 Renégo.2022
## 1              NA       FALSE       FALSE       FALSE
## 2            0.06       FALSE       FALSE       FALSE
## 3            0.06       FALSE       FALSE       FALSE
## 4            0.06       FALSE       FALSE       FALSE
## 5            0.06       FALSE       FALSE       FALSE
## 6            0.06       FALSE       FALSE       FALSE
```

Définition du format pour chaque variable

- Gp assurés : mettre majorité
- Produit : Il y a plus de 2000 produits donc on ne peut pas faire une variable par produit. Une entreprise semble avoir au maximum 3 produits et chaque produit mène à une indexation différente donc il serait intéressant de garder les 3 produits les plus représenté par ordre et leur indexation
- Date d'adhésion : pas d'intérêt / mettre année la plus vieille
- Code APE : mettre majorité
- Département : mettre majorité
- Regroup 1 : mettre majorité
- Regroup 4 : semble 1 entreprise = 1 regroup 4
- Regroup 5 : Créer 4 catégories (Base, Option, Asso, Surcomp) et mettre la majorité
- Num pers/ num Ind: calculer nombre de personnes et de famille. Attention ! Enlever les personnes radiées
- Date adhésion pers : pas d'intérêt
- Date radiation : voir précédement
- Type assuré : 3 catégories = variables avec leurs effectifs
- Sexe : 2 catégories = variables avec leurs effectifs
- Date de naissance : pas d'intérêt / sinon tranche d'âge
- R/RN : 2 catégories = variables avec leurs effectifs
- Num contrat grandes entreprises: faire 1 variable oui/non
- Indexation 2018/2019/2020 : faire pour les 3 produits
- Renégo 2020/2021/2022 : variable oui/non

Fonctions pour attribuer une ligne par entreprise

Dataframes de test

Garder le max représenté

Pour les variables :

- Code APE (6)
- Département (7)

- Regroup1 (8)
- Regroup4 (9)

```
max_repr <- function(data,ind){
  if (length(table(data[,ind]))!=0){
    frequences=as.data.frame(table(data[,ind]))
    ind_level_max=which(table(data[,ind])==max(table(data[,ind])))
    level_max=frequences[ind_level_max,1]
    return (as.character(level_max))
  }else{
    return (NA)
  }
}
```

Test unitaire de max_repr

```
max_repr(L1,6)
```

```
## [1] "2932Z"
```

```
max_repr(L1,7)
```

```
## [1] NA
```

```
max_repr(L1,8)
```

```
## [1] "ERCAC"
```

```
max_repr(L1,9)
```

```
## [1] "1"
```

Garder les 3 produits les plus présents

-Produits

```
max3_repr <- function(data) {
  frequences=as.data.frame(table(data[,3]))
  ordre=tail(sort(frequences[,2]),3)
  n=length(ordre)
  ind_first=which(frequences[,2]==ordre[n])
  first=as.integer(as.character(frequences[ind_first,1]))
  if(length(ordre)>1){
    ind_second=which(frequences[,2]==ordre[n-1])
    second=as.integer(as.character(frequences[ind_second,1]))
  }else{
    second=NA
  }
  if(length(ordre)>2){
    ind_third=which(frequences[,2]==ordre[n-2])
    third=as.integer(as.character(frequences[ind_third,1]))
  }else{
    third=NA
  }
  return (c(first,second,third))
}
```

```
max3_repr(L1)
```

```
## [1]  1 NA NA
```

```
max3_repr(L1002)
```

```
## [1] 2828   53   NA
```

```
max3_repr(L9)
```

```
## [1]  9 NA NA
```

Indexations correspondantes aux 3 produits

```
index_prod<-function(data,prod){
  if(!is.na(prod)){
    sub_prod=subset(data,Produit.Anonyme=prod)
    indexation2018_2023=sub_prod[1,21:26]

  }else{
    indexation2018_2023=rep(NA,6)
  }
  return(indexation2018_2023)
}
```

```
index_prod(L1,1)
```

```
##   Indexation.2018 Indexation.2019 Indexation.2020 Indexation.2021
## 1              NA              NA              NA              NA
##   Indexation.2022 Indexation.2023
## 1              NA              NA
```

```
index_prod(L1,NA)
```

```
## [1] NA NA NA NA NA NA
```

```
index_prod(L1002,2828)
```

```
##        Indexation.2018 Indexation.2019 Indexation.2020 Indexation.2021
## 272951            0.03               0            0.01               0
##        Indexation.2022 Indexation.2023
## 272951            0.03            0.15
```

```
index_prod(L1002,53)
```

```
##        Indexation.2018 Indexation.2019 Indexation.2020 Indexation.2021
## 272951            0.03               0            0.01               0
##        Indexation.2022 Indexation.2023
## 272951            0.03            0.15
```

Catégories et effectifs

Pour les varaibles suivantes : - Type assuré - Sexe - R/RN - Regroup5 - Code groupe assuré

```
#transpose(aggregate(L2$Type.Assure,list(L2$Type.Assure),length))
type=aggregate(L2$Type.Assure,list(L2$Type.Assure),length)
sexe=aggregate(L2$Sexe,list(L2$Sexe),length)
r.rn=aggregate(L2$R.NR,list(L2$R.NR),length)
r.rn=aggregate(L9$R.NR,list(L9$R.NR),length)
type
```

```
##   Group.1  x
## 1  ASSPRI 38
## 2  CONJOI 14
```

```
sexe
```

```
##   Group.1  x
## 1       F 29
## 2       M 23
```

```
r.rn
```

```
##   Group.1  x
## 1      NR 74
```

Mise en pourcentage

```
pourcentage<-function(data,eff){
  eff[,2]=eff[,2]/nrow(data)*100
  return(eff)
}
```

```
pourcentage(L2,type)
```

```
##   Group.1        x
## 1  ASSPRI 73.07692
## 2  CONJOI 26.92308
```

```
pourcentage(L2,sexe)
```

```
##   Group.1        x
## 1       F 55.76923
## 2       M 44.23077
```

```
pourcentage(L2,r.rn)
```

```
##   Group.1        x
## 1      NR 142.3077
```

Date d'adhésion lien

Pour la variable : annee moyen d'adhésion de l'entreprise au lien avec la ponderation en du nombre d'effectifs dans l'entreprise.

```
annee_adh_L<-function(data){
  n=length(levels(data[,1]))
  tab <- data.frame(Annee_Entreprise = integer(n),
                    Effectif_Entreprise = integer(n))
  for (i in 1:n){
    sub_entreprise<-subset(data,Num.Ctr.Coll.Anonyme==as.integer(levels(data[,1])[i]))
    tab$Annee_Entreprise[i]=year(sub_entreprise[,4])[1]
    tab$Effectif_Entreprise[i]=nrow(sub_entreprise)

  }
  annee=sum(tab$Annee_Entreprise*tab$Effectif_Entreprise)/nrow(data)
  return(as.integer(annee))
}
```

Test unitaire

```
annee_adh_L(L1)
```

```
## [1] 2022
```

```
annee_adh_L(L9)
```

```
## [1] 2009
```

```
annee_adh_L(L1002)
```

```
## [1] 2021
```

```
annee_adh_L(L2)
```

```
## [1] 1976
```

Temps d'adhésion lien

Calculer le temps moyen d'adhésion d'un lien en calculant le temps moyen d'adhésion des entreprises dans un lien en le pondérant par effectifs de l'entreprise

```
temps_adh_L<-function(data){
  n=length(levels(data[,1]))
  tab <- data.frame(Tmps_Entreprise = integer(n),
                    Effectif_Entreprise = integer(n))
  for (i in 1:n){
    sub_entreprise<-subset(data,Num.Ctr.Coll.Anonyme==as.integer(levels(data[,1])[i]))
    if(year(sub_entreprise[1,5])==2050){

      temps=as.numeric(Sys.Date()-sub_entreprise[1,4])
    }else{
      temps=as.numeric(sub_entreprise[1,5]-sub_entreprise[1,4])
    }

    tab$Tmps_Entreprise[i]=as.integer(temps)

    tab$Effectif_Entreprise[i]=nrow(sub_entreprise)
  }

  adh=sum(tab$Tmps_Entreprise*tab$Effectif_Entreprise)/nrow(data)

  return(as.integer(adh))
}

#retour en jours
```

```
print(temps_adh_L(L1))
```

```
## [1] 324
```

```
print(temps_adh_L(L9))
```

```
## [1] 4932
```

Effectif personnes et familles

Pour les variables : - Num pers - Num ind

Avec retrait des personnes et familles radiées.

```
effectif_pers_fam_actu<-function(data){
  data_sub=subset(data, year(data[,14])==2050)
  nb_pers=nrow(data_sub)
  data_sub=droplevels(data_sub)
  nb_fam=length(levels(data_sub[,12]))
  return (c(nb_pers,nb_fam))
}
```

Test unitaire

```
effectif_pers_fam_actu(L1)
```

```
## [1] 258 109
```

```
effectif_pers_fam_actu(L9)
```

```
## [1] 26 19
```

```
#nb pers, nb_fam
```

Temps d'adhésion moyenne des personnes

Temps moyen d'adhésion des personnes à un lien en le pondérant par l'effectif dans chaque annee.

```
temps_moy_adh_P<-function(data){
  adh=0

  #calcul du nombre d'entrepise dans le lien
  n=length(levels(data[,1]))
  tab <- data.frame(Tmps_Entreprise = integer(n),
                    Effectif_Entreprise = integer(n))
  for (i in 1:n){
    sub_entreprise<-subset(data,Num.Ctr.Coll.Anonyme==as.integer(levels(data[,1])[i]))
#calcul temps adhesion de toutes les personnes dans  une entreprise
    if(year(data[i,14])=='2050'){
      adh=adh+as.numeric(Sys.Date()-data[i,13])
    }else{
      adh=adh+as.numeric(data[i,14]-data[i,13])
    }

    tab$Tmps_Entreprise[i]=adh
    tab$Effectif_Entreprise[i]=nrow(sub_entreprise)
  }
  #total
  adh_moyenne=sum(tab$Tmps_Entreprise*tab$Effectif_Entreprise)/nrow(data)
  return (as.integer(adh_moyenne))
}
#en jour
```

```
temps_moy_adh_P(L2)
```

```
## [1] 4617
```

```
temps_moy_adh_P(L1002)
```

```
## [1] 184
```

Ancienneté moyenne de personne dans un lien

```
ancien_pers<-function(data){
  n=length(levels(data[,1]))
  tab <- data.frame(Anciennete_Entreprise = integer(n),
                    Effectif_Entreprise = integer(n))
  for (i in 1:n){
    sub_entreprise<-subset(data,Num.Ctr.Coll.Anonyme==as.integer(levels(data[,1])[i]))
    tab$Anciennete_Entreprise[i]=mean(year(sub_entreprise$Date.Effet.Adhesion.Num.Personne))
    tab$Effectif_Entreprise[i]=nrow(sub_entreprise)
  }
  anciennete=sum(tab$Anciennete_Entreprise*tab$Effectif_Entreprise)/nrow(data)
  return (as.integer(anciennete))
}
```

```
print(ancien_pers(L1))
```

```
## [1] 2022
```

```
print(ancien_pers(L9))
```

```
## [1] 2011
```

```
print(ancien_pers(L1002))
```

```
## [1] 2018
```

0/1 pour Grands comptes

```
# true or false
gd_compte<-function(data){
  return (data[1,20])
}
```

Test unitaire

```
gd_compte(L2)
```

```
## [1] FALSE
```

```
gd_compte(L1)
```

```
## [1] TRUE
```

Récupération du nombre d'entreprises et de liens

```
paste("Il y a ", length(levels(effectif$Num.Ctr.Coll.Anonyme)), "entreprises.")
```

```
## [1] "Il y a  3789 entreprises."
```

```
paste("Il y a ",length(levels(effectif$Lien.entreprise.Anonyme))," liens")
```

```
## [1] "Il y a  1007  liens"
```

```
nb_liens=length(levels(effectif$Lien.entreprise.Anonyme))
```

Création d'un dataframe pour récupérer ces nouvelles informations

```
tab <- data.frame(Lien.entreprise.Anonyme = integer(nb_liens),
                  Actifs = integer(nb_liens),
                  Non_actifs = integer(nb_liens),
                  Portabilite = integer(nb_liens),
                  Produit1= integer(nb_liens),
                  Produit2= integer(nb_liens),
                  Produit3= integer(nb_liens),
                  Annee_adh_E_au_L = integer(nb_liens),
                  Tps_adh_E_au_L = integer(nb_liens),
                  Code_ape = character(nb_liens),
                  Departement= character(nb_liens),
                  Regroup1= character(nb_liens),
                  Regroup4=character(nb_liens),
                  BASE_percent=numeric(nb_liens), # en pourcentage
                  OPTION_percent = numeric(nb_liens), # en pourcentage
                  Nb_pers = integer(nb_liens),
                  Nb_fam =integer(nb_liens),
                  Anciennete_pers_ds_L = integer(nb_liens),
                  Tps_adh_moy_pers_ds_L = integer(nb_liens),
                  Asspri = integer(nb_liens),# effectif
                  Asspri_percent= numeric(nb_liens),#pourcentage
                  Conjoint = integer(nb_liens), #eff
                  Conjoint_percent = numeric(nb_liens), # pourcentage
                  Enfant = integer(nb_liens), #eff
                  Enfant_percent = numeric(nb_liens), #pourcentage
                  Femme = integer(nb_liens), # eff
                  Femme_percent = numeric(nb_liens), # pourcentage
                  Homme = integer(nb_liens), #eff
                  Homme_percent = numeric(nb_liens), #pourcentage
                  Respo = integer(nb_liens), # eff
                  Respo_percent = numeric(nb_liens), # pourcentage
                  Non_respo = integer(nb_liens), #eff
                  Non_respo_percent = numeric(nb_liens), #pourcentage
                  VIP = logical(nb_liens),
                  Prod1_Ind2018 = double(nb_liens),
                  Prod1_Ind2019 = double(nb_liens),
                  Prod1_Ind2020 = double(nb_liens),
                  Prod1_Ind2021 = double(nb_liens),
                  Prod1_Ind2022 = double(nb_liens),
                  Prod1_Ind2023 = double(nb_liens),
                  Prod2_Ind2018 = double(nb_liens),
                  Prod2_Ind2019 = double(nb_liens),
                  Prod2_Ind2020 = double(nb_liens),
                  Prod2_Ind2021 = double(nb_liens),
                  Prod2_Ind2022 = double(nb_liens),
                  Prod2_Ind2023 = double(nb_liens),
                  Prod3_Ind2018 = double(nb_liens),
                  Prod3_Ind2019 = double(nb_liens),
                  Prod3_Ind2020 = double(nb_liens),
                  Prod3_Ind2021 = double(nb_liens),
                  Prod3_Ind2022 = double(nb_liens),
                  Prod3_Ind2023 = double(nb_liens),
```

```
                Renégo2020 = logical(nb_liens),
                Renégo2021 = logical(nb_liens),
                Renégo2022= logical(nb_liens))
```

Subdataset par lien entreprise

```
for (i in 1:1007){
  lien=levels(effectif$Lien.entreprise.Anonyme)[i]
  sub_lien=subset(effectif,Lien.entreprise.Anonyme==lien)
  sub_lien=droplevels(sub_lien)
  #Lien
  tab$Lien.entreprise.Anonyme[i]=lien

  #Code groupe assures
  code_gp_as=aggregate(sub_lien[,2],list(sub_lien[,2]),length)
  ind_actifs=which(code_gp_as=="ACTIFS")
  ind_non_actifs=which(code_gp_as=="NON_ACTIFS")
  ind_portabi=which(code_gp_as=="PORTABILITE")
  if(length(ind_actifs)){
    tab$Actifs[i]=code_gp_as[ind_actifs,2]
  }else{
    tab$Actifs[i]=0
  }
  if(length(ind_non_actifs)){
    tab$Non_actifs[i]=code_gp_as[ind_non_actifs,2]
  }else{
    tab$Non_actifs[i]=0
  }
  if(length(ind_portabi)){
    tab$Portabilite[i]=code_gp_as[ind_portabi,2]
  }else{
    tab$Portabilite[i]=0
  }

  #produits 1 à 3
  tab[i,5:7]=max3_repr(sub_lien)

  #année d'adhésion du lien
  tab$Annee_adh_E_au_L[i]=annee_adh_L(sub_lien)

  #temps adésien entreprise
  tab$Tps_adh_E_au_L[i]=temps_adh_L(sub_lien)

  #code ape le plus présent
  tab$Code_ape[i]=max_repr(sub_lien,6)

  #departement le plus present
  tab$Departement[i]=max_repr(sub_lien,7)

  # regroup1 le plus present
  tab$Regroup1[i]=max_repr(sub_lien,8)

  #regroup4 le plus present
```

```
tab$Regroup4[i]=max_repr(sub_lien,9)

#base et option (regroup5)
regroup5=aggregate(sub_lien[,10],list(sub_lien[,10]),length)
regroup5_percent=pourcentage(sub_lien,regroup5)
ind_base=which(regroup5=="BASE")
ind_option=which(regroup5=="OPTION")
if(length(ind_base)){
  tab$BASE_percent[i]=regroup5_percent[ind_base,2]
}else{
  tab$BASE_percent[i]=0
}
if(length(ind_option)){
  tab$OPTION_percent[i]=regroup5_percent[ind_option,2]
}else{
  tab$OPTION_percent[i]=0
}

#nb_pers et famille
tab[i,16:17]=effectif_pers_fam_actu(sub_lien)

#anciennete moyenne des personnes
tab$Anciennete_pers_ds_L[i]=ancien_pers(sub_lien)

#temps d'adhésion moyen des personnes
tab$Tps_adh_moy_pers_ds_L[i]=temps_moy_adh_P(sub_lien)

#assure conjoint enfant
type_ass=aggregate(sub_lien[,15],list(sub_lien[,15]),length)
type_ass_percent=pourcentage(sub_lien,type_ass)
ind_as=which(type_ass=="ASSPRI")
ind_conj=which(type_ass=="CONJOI")
ind_enf=which(type_ass=="ENFANT")
if(length(ind_as)){
  tab$Asspri[i]=type_ass[ind_as,2]
  tab$Asspri_percent[i]=type_ass_percent[ind_as,2]
}else{
  tab$Asspri[i]=0
  tab$Asspri_percent[i]=0
}
if(length(ind_conj)){
  tab$Conjoint[i]=type_ass[ind_conj,2]
  tab$Conjoint_percent[i]=type_ass_percent[ind_conj,2]
}else{
  tab$Conjoint[i]=0
  tab$Conjoint_percent[i]=0
}
if(length(ind_enf)){
  tab$Enfant[i]=type_ass[ind_enf,2]
  tab$Enfant_percent[i]=type_ass_percent[ind_enf,2]
}else{
  tab$Enfant[i]=0
  tab$Enfant_percent[i]=0
```

```
}

#sexe
sexe=aggregate(sub_lien[,16],list(sub_lien[,16]),length)
sexe_percent=pourcentage(sub_lien,sexe)
ind_m=which(sexe=="M")
ind_f=which(sexe=="F")
if(length(ind_m)){
  tab$Homme[i]=sexe[ind_m,2]
  tab$Homme_percent[i]=sexe_percent[ind_m,2]
}else{
  tab$Homme[i]=0
  tab$Homme_percent[i]=0
}
if(length(ind_f)){
  tab$Femme[i]=sexe[ind_f,2]
  tab$Femme_percent[i]=sexe_percent[ind_f,2]
}else{
  tab$Femme[i]=0
  tab$Femme_percent[i]=0
}

#r.rn
RNR=aggregate(sub_lien[,18],list(sub_lien[,18]),length)
RNR_percent=pourcentage(sub_lien,RNR)
ind_r=which(RNR=="R")
ind_rn=which(RNR=="NR")
if(length(ind_r)){
  tab$Respo[i]=RNR[ind_r,2]
  tab$Respo_percent[i]=RNR_percent[ind_r,2]
}else{
  tab$Respo[i]=0
  tab$Respo_percent[i]=0
}
if(length(ind_rn)){
  tab$Non_respo[i]=RNR[ind_rn,2]
  tab$Non_respo_percent[i]=RNR_percent[ind_rn,2]
}else{
  tab$Non_respo[i]=0
  tab$Non_respo_percent[i]=0
}

#vip
tab$VIP[i]=gd_compte(sub_lien)

#index prod1
tab[i,35:40]=index_prod(sub_lien,tab[i,5])

#indexation prod2
tab[i,41:46]=index_prod(sub_lien,tab[i,6])

#indexation prod3
tab[i,47:52]=index_prod(sub_lien,tab[i,7])
```

```
  #renego
  tab[i,53:55]=sub_lien[1,27:29]
}
```

```
## Warning in tab$Code_ape[i] <- max_repr(sub_lien, 6): le nombre d'objets à
## remplacer n'est pas multiple de la taille du remplacement

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in tab$Code_ape[i] <- max_repr(sub_lien, 6): le nombre d'objets à
## remplacer n'est pas multiple de la taille du remplacement

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in tab$Code_ape[i] <- max_repr(sub_lien, 6): le nombre d'objets à
## remplacer n'est pas multiple de la taille du remplacement

## Warning in tab$Code_ape[i] <- max_repr(sub_lien, 6): le nombre d'objets à
## remplacer n'est pas multiple de la taille du remplacement

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in tab$Code_ape[i] <- max_repr(sub_lien, 6): le nombre d'objets à
## remplacer n'est pas multiple de la taille du remplacement

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in tab$Regroup4[i] <- max_repr(sub_lien, 9): le nombre d'objets à
## remplacer n'est pas multiple de la taille du remplacement

## Warning in matrix(value, n, p): la longueur des données [4] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in matrix(value, n, p): la longueur des données [4] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in tab$Code_ape[i] <- max_repr(sub_lien, 6): le nombre d'objets à
## remplacer n'est pas multiple de la taille du remplacement
```

```
## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in matrix(value, n, p): la longueur des données [5] n'est pas un
## diviseur ni un multiple du nombre de colonnes [3]

## Warning in tab$Code_ape[i] <- max_repr(sub_lien, 6): le nombre d'objets à
## remplacer n'est pas multiple de la taille du remplacement

## Warning in tab$Departement[i] <- max_repr(sub_lien, 7): le nombre d'objets à
## remplacer n'est pas multiple de la taille du remplacement
```

```
tab$Lien.entreprise.Anonyme<-as.factor(tab$Lien.entreprise.Anonyme)
tab$Code_ape<-as.factor(tab$Code_ape)
tab$Departement<-as.factor(tab$Departement)
tab$Regroup1<-as.factor(tab$Regroup1)
tab$Regroup4<-as.factor(tab$Regroup4)
tab$Produit1<-as.factor(tab$Produit1)
tab$Produit2<-as.factor(tab$Produit2)
tab$Produit3<-as.factor(tab$Produit3)
```

Save en Effectif_final.Rdata

```
save(tab, file='../Donnees/Effectif_final.Rdata')
```

```
write.table(tab,file='../Donnees/Effectif_final.csv',sep=";",row.names = FALSE)
```