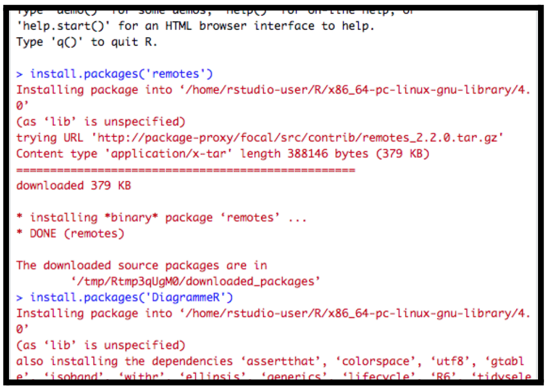
TD DATA LINEAGE - MiniCours

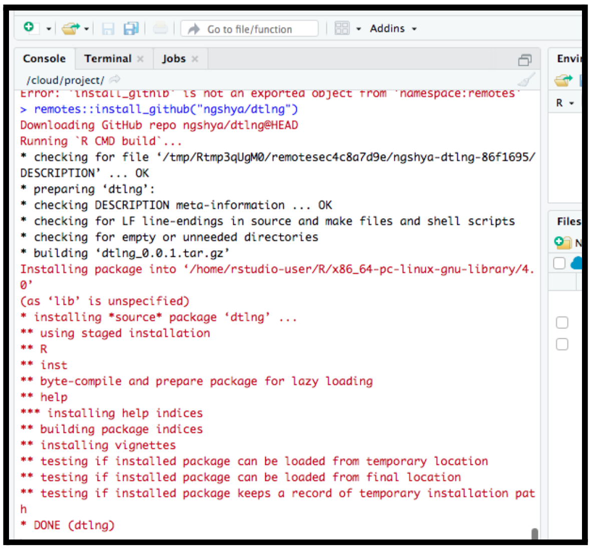
CASANOVA S. , RAMOS Y.

1. Installer les packages suivants: remotes, DiagrammeR



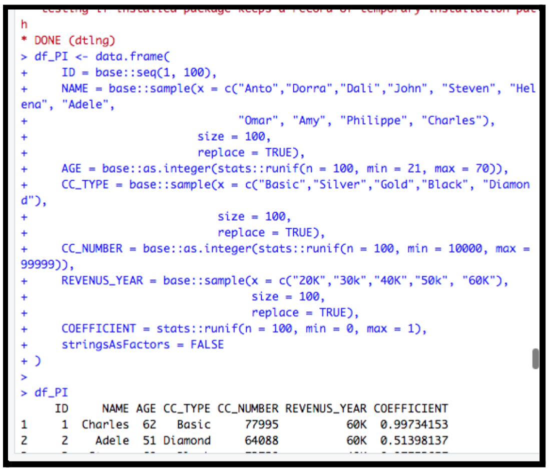
```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> install.packages('remotes')
Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.
0'
(as 'lib' is unspecified)
trying URL 'http://package-proxy/focal/src/contrib/remotes_2.2.0.tar.gz'
Content type 'application/x-tar' length 388146 bytes (379 KB)
==================================================
downloaded 379 KB

* installing *binary* package 'remotes' ...
* DONE (remotes)

The downloaded source packages are in
	'/tmp/Rtmp3qUgM0/downloaded_packages'
> install.packages('DiagrammeR')
Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.
0'
(as 'lib' is unspecified)
also installing the dependencies 'assertthat', 'colorspace', 'utf8', 'gtabl
e', 'isoband', 'withr', 'ellipsis', 'generics', 'lifecycle', 'R6', 'tidysele
```

2. Importer la librairie “dtlng” de github via cette commande: remotes::install_githib(“ngshya/dtlng”)

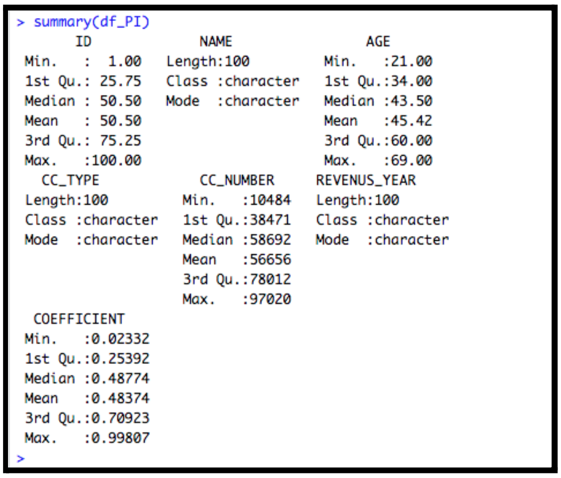


```
Error: 'install_githib' is not an exported object from 'namespace:remotes'
> remotes::install_github("ngshya/dtlng")
Downloading GitHub repo ngshya/dtlng@HEAD
Running `R CMD build`...
* checking for file '/tmp/Rtmp3qUgM0/remotesec4c8a7d9e/ngshya-dtlng-86f1695/
DESCRIPTION' ... OK
* preparing 'dtlng':
* checking DESCRIPTION meta-information ... OK
* checking for LF line-endings in source and make files and shell scripts
* checking for empty or unneeded directories
* building 'dtlng_0.0.1.tar.gz'
Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.
0'
(as 'lib' is unspecified)
* installing *source* package 'dtlng' ...
** using staged installation
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** installing vignettes
** testing if installed package can be loaded from temporary location
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation pat
h
* DONE (dtlng)
```

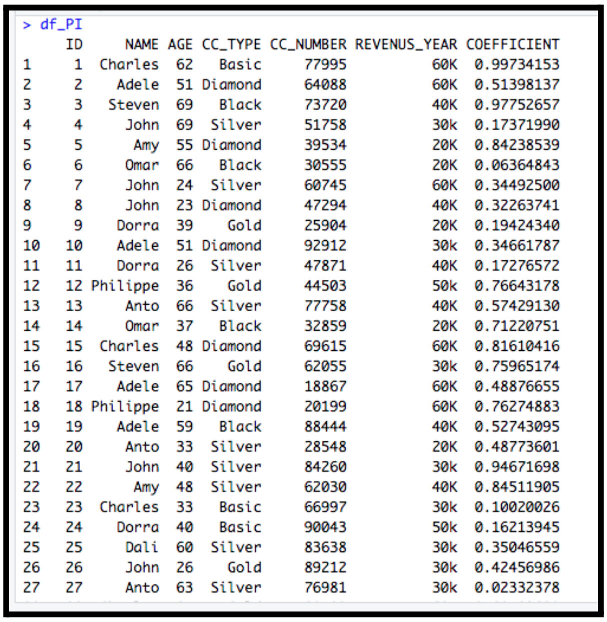
3. Création du dataframe 1 contenant les informations personnelles comme indiqué dans le code R suivant:



```
** testing if installed package keeps a record of temporary installation pat
h
* DONE (dtlng)
> df_PI <- data.frame(
+     ID = base::seq(1, 100),
+     NAME = base::sample(x = c("Anto","Dorra","Dali","John", "Steven", "Hel
ena", "Adele",
+                                "Omar", "Amy", "Philippe", "Charles"),
+                          size = 100,
+                          replace = TRUE),
+     AGE = base::as.integer(stats::runif(n = 100, min = 21, max = 70)),
+     CC_TYPE = base::sample(x = c("Basic","Silver","Gold","Black", "Diamon
d"),
+                             size = 100,
+                             replace = TRUE),
+     CC_NUMBER = base::as.integer(stats::runif(n = 100, min = 10000, max =
99999)),
+     REVENUS_YEAR = base::sample(x = c("20K","30k","40K","50k", "60K"),
+                                  size = 100,
+                                  replace = TRUE),
+     COEFFICIENT = stats::runif(n = 100, min = 0, max = 1),
+     stringsAsFactors = FALSE
+ )
>
> df_PI
    ID    NAME AGE CC_TYPE CC_NUMBER REVENUS_YEAR COEFFICIENT
1    1 Charles  62   Basic     77995          60K  0.99734153
2    2   Adele  51 Diamond     64088          60K  0.51398137
```

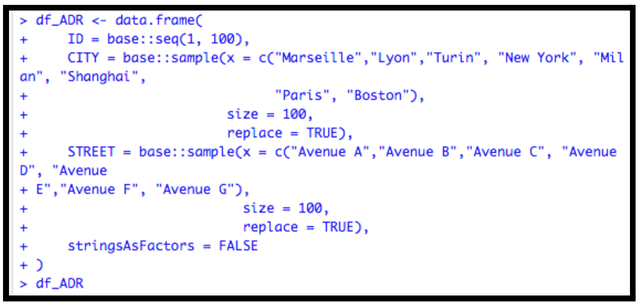


```
> summary(df_PI)
       ID             NAME                AGE
 Min.   :  1.00   Length:100         Min.   :21.00
 1st Qu.: 25.75   Class :character   1st Qu.:34.00
 Median : 50.50   Mode  :character   Median :43.50
 Mean   : 50.50                      Mean   :45.42
 3rd Qu.: 75.25                      3rd Qu.:60.00
 Max.   :100.00                      Max.   :69.00
   CC_TYPE            CC_NUMBER     REVENUS_YEAR
 Length:100         Min.   :10484   Length:100
 Class :character   1st Qu.:38471   Class :character
 Mode  :character   Median :58692   Mode  :character
                    Mean   :56656
                    3rd Qu.:78012
                    Max.   :97020
  COEFFICIENT
 Min.   :0.02332
 1st Qu.:0.25392
 Median :0.48774
 Mean   :0.48374
 3rd Qu.:0.70923
 Max.   :0.99807
>
```

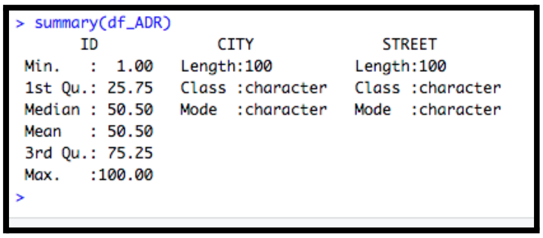


```
> df_PI
   ID     NAME AGE CC_TYPE CC_NUMBER REVENUS_YEAR COEFFICIENT
1   1  Charles  62   Basic     77995          60K  0.99734153
2   2    Adele  51 Diamond     64088          60K  0.51398137
3   3   Steven  69   Black     73720          40K  0.97752657
4   4     John  69  Silver     51758          30k  0.17371990
5   5      Amy  55 Diamond     39534          20K  0.84238539
6   6     Omar  66   Black     30555          20K  0.06364843
7   7     John  24  Silver     60745          60K  0.34492500
8   8     John  23 Diamond     47294          40K  0.32263741
9   9    Dorra  39    Gold     25904          20K  0.19424340
10 10    Adele  51 Diamond     92912          30k  0.34661787
11 11    Dorra  26  Silver     47871          40K  0.17276572
12 12 Philippe  36    Gold     44503          50k  0.76643178
13 13     Anto  66  Silver     77758          40K  0.57429130
14 14     Omar  37   Black     32859          20K  0.71220751
15 15  Charles  48 Diamond     69615          60K  0.81610416
16 16   Steven  66    Gold     62055          30k  0.75965174
17 17    Adele  65 Diamond     18867          60K  0.48876655
18 18 Philippe  21 Diamond     20199          60K  0.76274883
19 19    Adele  59   Black     88444          40K  0.52743095
20 20     Anto  33  Silver     28548          20K  0.48773601
21 21     John  40  Silver     84260          30k  0.94671698
22 22      Amy  48  Silver     62030          40K  0.84511905
23 23  Charles  33   Basic     66997          30k  0.10020026
24 24    Dorra  40   Basic     90043          50k  0.16213945
25 25     Dali  60  Silver     83638          30k  0.35046559
26 26     John  26    Gold     89212          30k  0.42456986
27 27     Anto  63  Silver     76981          30k  0.02332378
```

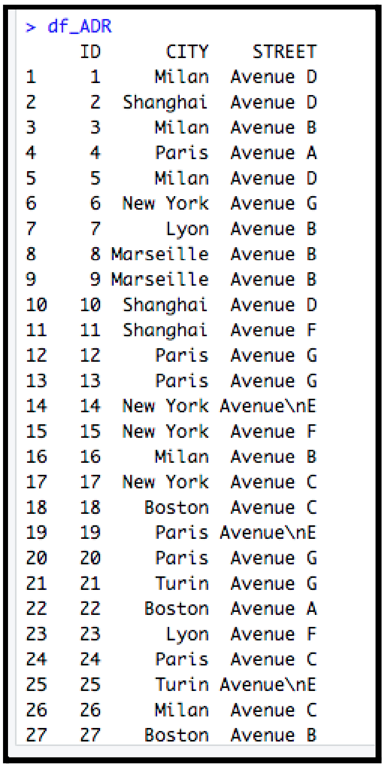
4. Création du dataframe 2 contenant les données de villes et adresses comme indiqué dans le code R suivant:



```
> df_ADR <- data.frame(
+     ID = base::seq(1, 100),
+     CITY = base::sample(x = c("Marseille","Lyon","Turin", "New York", "Mil
an", "Shanghai",
+                               "Paris", "Boston"),
+                         size = 100,
+                         replace = TRUE),
+     STREET = base::sample(x = c("Avenue A","Avenue B","Avenue C", "Avenue
D", "Avenue
+ E","Avenue F", "Avenue G"),
+                           size = 100,
+                           replace = TRUE),
+     stringsAsFactors = FALSE
+ )
> df_ADR
```

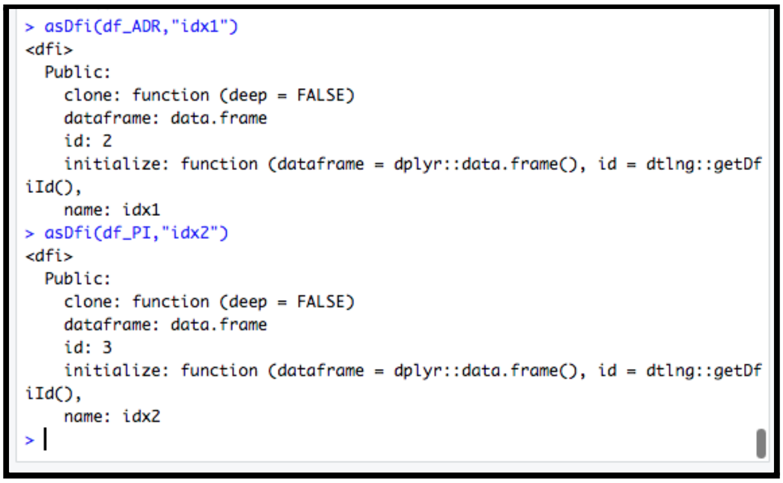


```
> summary(df_ADR)
       ID             CITY              STREET         
 Min.   :  1.00   Length:100         Length:100        
 1st Qu.: 25.75   Class :character   Class :character  
 Median : 50.50   Mode  :character   Mode  :character  
 Mean   : 50.50                                        
 3rd Qu.: 75.25                                        
 Max.   :100.00                                        
>
```



```
> df_ADR
    ID      CITY    STREET
1    1     Milan  Avenue D
2    2  Shanghai  Avenue D
3    3     Milan  Avenue B
4    4     Paris  Avenue A
5    5     Milan  Avenue D
6    6  New York  Avenue G
7    7      Lyon  Avenue B
8    8 Marseille  Avenue B
9    9 Marseille  Avenue B
10  10  Shanghai  Avenue D
11  11  Shanghai  Avenue F
12  12     Paris  Avenue G
13  13     Paris  Avenue G
14  14  New York Avenue\nE
15  15  New York  Avenue F
16  16     Milan  Avenue B
17  17  New York  Avenue C
18  18    Boston  Avenue C
19  19     Paris Avenue\nE
20  20     Paris  Avenue G
21  21     Turin  Avenue G
22  22    Boston  Avenue A
23  23      Lyon  Avenue F
24  24     Paris  Avenue C
25  25     Turin Avenue\nE
26  26     Milan  Avenue C
27  27    Boston  Avenue B
```

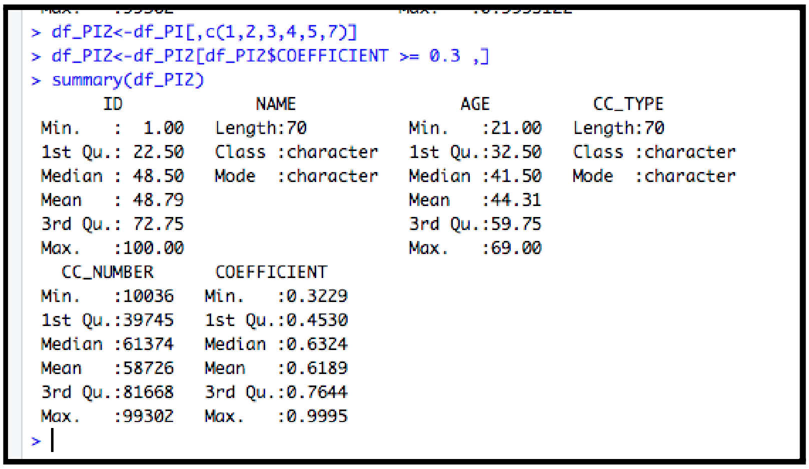
5.Créer des index sur les 2 dataframes avec asDfi()



```
> asDfi(df_ADR,"idx1")
<dfi>
  Public:
    clone: function (deep = FALSE)
    dataframe: data.frame
    id: 2
    initialize: function (dataframe = dplyr::data.frame(), id = dtlng::getDf
iId(),
    name: idx1
> asDfi(df_PI,"idx2")
<dfi>
  Public:
    clone: function (deep = FALSE)
    dataframe: data.frame
    id: 3
    initialize: function (dataframe = dplyr::data.frame(), id = dtlng::getDf
iId(),
    name: idx2
> 
```

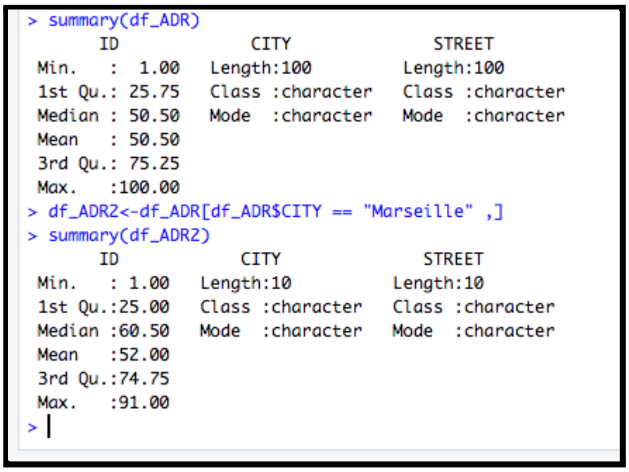
6. Manipulation des dataframes: lancer les requêtes suivantes

1. Select tous les colonnes sauf les Revenus annuelles (REVENUS_YEAR) pour les individus avec coeff > = 0.3 (COEFFICIENT)



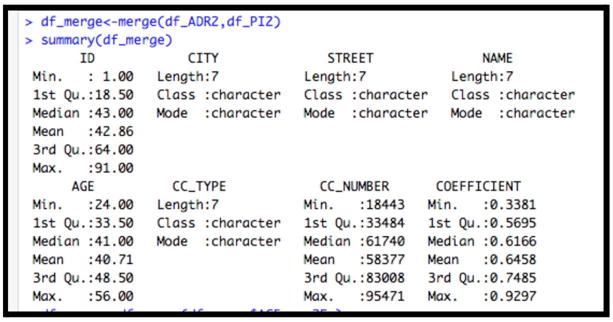
```
> df_PI2<-df_PI[,c(1,2,3,4,5,7)]
> df_PI2<-df_PI2[df_PI2$COEFFICIENT >= 0.3 ,]
> summary(df_PI2)
       ID              NAME               AGE           CC_TYPE         
 Min.   :  1.00   Length:70          Min.   :21.00   Length:70         
 1st Qu.: 22.50   Class :character   1st Qu.:32.50   Class :character  
 Median : 48.50   Mode  :character   Median :41.50   Mode  :character  
 Mean   : 48.79                      Mean   :44.31                     
 3rd Qu.: 72.75                      3rd Qu.:59.75                     
 Max.   :100.00                      Max.   :69.00                     
   CC_NUMBER      COEFFICIENT    
 Min.   :10036   Min.   :0.3229  
 1st Qu.:39745   1st Qu.:0.4530  
 Median :61374   Median :0.6324  
 Mean   :58726   Mean   :0.6189  
 3rd Qu.:81668   3rd Qu.:0.7644  
 Max.   :99302   Max.   :0.9995  
> 
```

2. Sélectionner avec un filtre la ville de Marseille

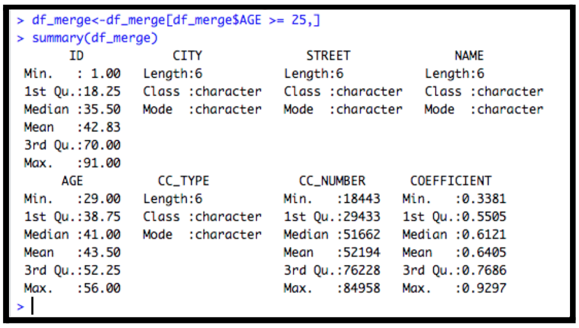


```
> summary(df_ADR)
       ID             CITY              STREET         
 Min.   :  1.00   Length:100         Length:100        
 1st Qu.: 25.75   Class :character   Class :character  
 Median : 50.50   Mode  :character   Mode  :character  
 Mean   : 50.50                                        
 3rd Qu.: 75.25                                        
 Max.   :100.00                                        
> df_ADR2<-df_ADR[df_ADR$CITY == "Marseille" ,]
> summary(df_ADR2)
       ID            CITY              STREET         
 Min.   : 1.00   Length:10          Length:10         
 1st Qu.:25.00   Class :character   Class :character  
 Median :60.50   Mode  :character   Mode  :character  
 Mean   :52.00                                        
 3rd Qu.:74.75                                        
 Max.   :91.00                                        
> 
```

3. Effectuer un inner_join sur l'ID et filtrer tous ce qui ont une AGE >=25 sur l'ID

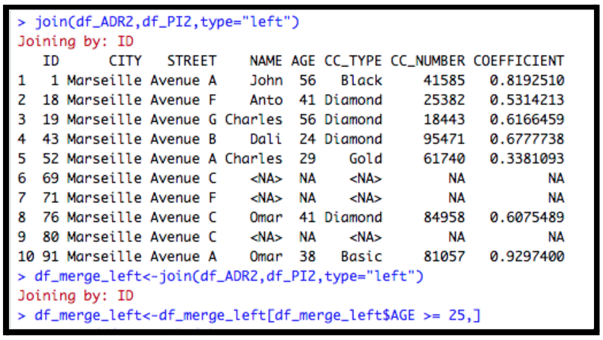


```
> df_merge<-merge(df_ADR2,df_PI2)
> summary(df_merge)
       ID             CITY              STREET              NAME          
 Min.   : 1.00   Length:7           Length:7           Length:7          
 1st Qu.:18.50   Class :character   Class :character   Class :character  
 Median :43.00   Mode  :character   Mode  :character   Mode  :character  
 Mean   :42.86                                                           
 3rd Qu.:64.00                                                           
 Max.   :91.00                                                           
      AGE          CC_TYPE            CC_NUMBER      COEFFICIENT    
 Min.   :24.00   Length:7           Min.   :18443   Min.   :0.3381  
 1st Qu.:33.50   Class :character   1st Qu.:33484   1st Qu.:0.5695  
 Median :41.00   Mode  :character   Median :61740   Median :0.6166  
 Mean   :40.71                      Mean   :58377   Mean   :0.6458  
 3rd Qu.:48.50                      3rd Qu.:83008   3rd Qu.:0.7485  
 Max.   :56.00                      Max.   :95471   Max.   :0.9297  
```

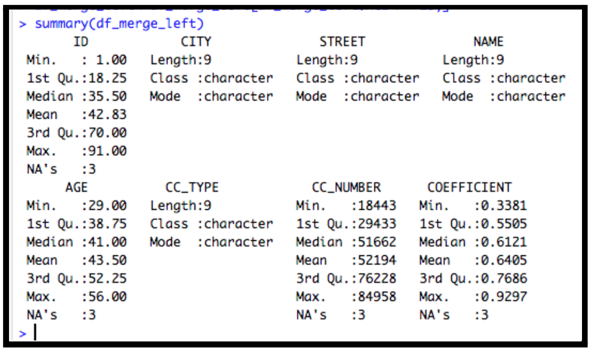


```
> df_merge<-df_merge[df_merge$AGE >= 25,]
> summary(df_merge)
       ID             CITY              STREET              NAME          
 Min.   : 1.00   Length:6           Length:6           Length:6          
 1st Qu.:18.25   Class :character   Class :character   Class :character  
 Median :35.50   Mode  :character   Mode  :character   Mode  :character  
 Mean   :42.83                                                           
 3rd Qu.:70.00                                                           
 Max.   :91.00                                                           
      AGE          CC_TYPE            CC_NUMBER      COEFFICIENT    
 Min.   :29.00   Length:6           Min.   :18443   Min.   :0.3381  
 1st Qu.:38.75   Class :character   1st Qu.:29433   1st Qu.:0.5505  
 Median :41.00   Mode  :character   Median :51662   Median :0.6121  
 Mean   :43.50                      Mean   :52194   Mean   :0.6405  
 3rd Qu.:52.25                      3rd Qu.:76228   3rd Qu.:0.7686  
 Max.   :56.00                      Max.   :84958   Max.   :0.9297  
> |
```

4. Effectuer un left_join sur l'ID et selectionner tous ce qui ont une AGE >=25 sur l'ID

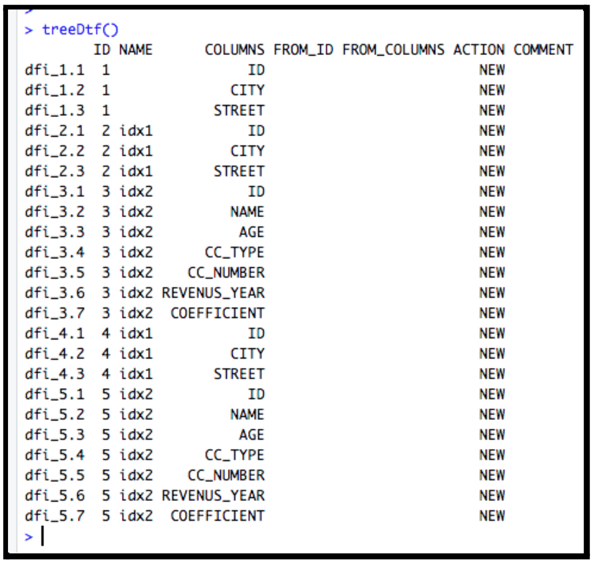


```
> join(df_ADR2,df_PI2,type="left")
Joining by: ID
   ID      CITY    STREET    NAME AGE CC_TYPE CC_NUMBER COEFFICIENT
1   1 Marseille Avenue A     John  56   Black     41585   0.8192510
2  18 Marseille Avenue F     Anto  41 Diamond     25382   0.5314213
3  19 Marseille Avenue G Charles  56 Diamond     18443   0.6166459
4  43 Marseille Avenue B     Dali  24 Diamond     95471   0.6777738
5  52 Marseille Avenue A Charles  29    Gold     61740   0.3381093
6  69 Marseille Avenue C     <NA>  NA    <NA>        NA          NA
7  71 Marseille Avenue F     <NA>  NA    <NA>        NA          NA
8  76 Marseille Avenue C     Omar  41 Diamond     84958   0.6075489
9  80 Marseille Avenue C     <NA>  NA    <NA>        NA          NA
10 91 Marseille Avenue A     Omar  38   Basic     81057   0.9297400
> df_merge_left<-join(df_ADR2,df_PI2,type="left")
Joining by: ID
> df_merge_left<-df_merge_left[df_merge_left$AGE >= 25,]
```



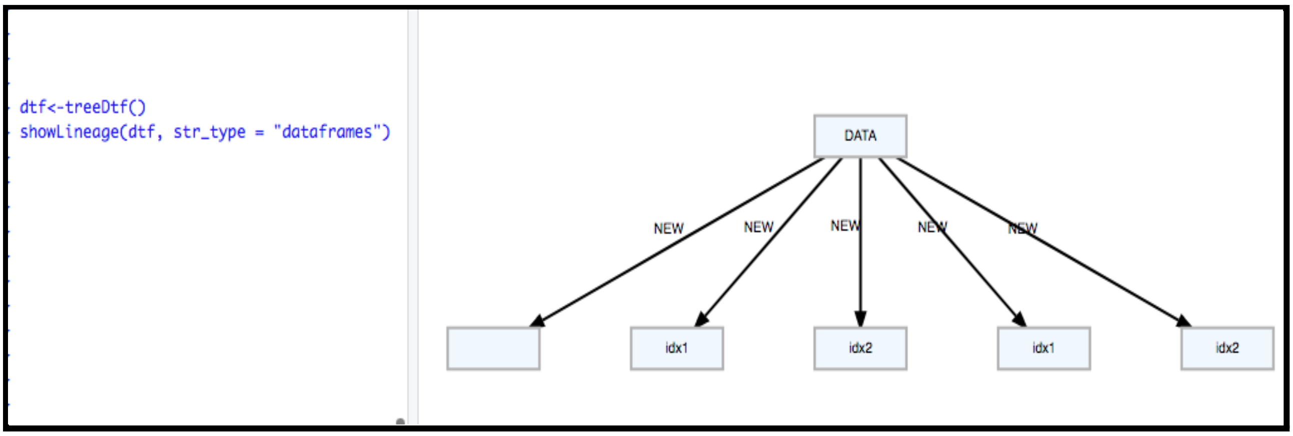
```
> summary(df_merge_left)
       ID             CITY              STREET              NAME          
 Min.   : 1.00   Length:9           Length:9           Length:9          
 1st Qu.:18.25   Class :character   Class :character   Class :character  
 Median :35.50   Mode  :character   Mode  :character   Mode  :character  
 Mean   :42.83                                                           
 3rd Qu.:70.00                                                           
 Max.   :91.00                                                           
 NA's   :3                                                               
      AGE          CC_TYPE            CC_NUMBER      COEFFICIENT    
 Min.   :29.00   Length:9           Min.   :18443   Min.   :0.3381  
 1st Qu.:38.75   Class :character   1st Qu.:29433   1st Qu.:0.5505  
 Median :41.00   Mode  :character   Median :51662   Median :0.6121  
 Mean   :43.50                      Mean   :52194   Mean   :0.6405  
 3rd Qu.:52.25                      3rd Qu.:76228   3rd Qu.:0.7686  
 Max.   :56.00                      Max.   :84958   Max.   :0.9297  
 NA's   :3                          NA's   :3       NA's   :3       
> |
```

5. Générer l'arbre de lineage avec la fonction treeDtf()



```
> treeDtf()
        ID NAME      COLUMNS FROM_ID FROM_COLUMNS ACTION COMMENT
dfi_1.1  1                ID                        NEW        
dfi_1.2  1              CITY                        NEW        
dfi_1.3  1            STREET                        NEW        
dfi_2.1  2 idx1           ID                        NEW        
dfi_2.2  2 idx1         CITY                        NEW        
dfi_2.3  2 idx1       STREET                        NEW        
dfi_3.1  3 idx2           ID                        NEW        
dfi_3.2  3 idx2         NAME                        NEW        
dfi_3.3  3 idx2          AGE                        NEW        
dfi_3.4  3 idx2      CC_TYPE                        NEW        
dfi_3.5  3 idx2    CC_NUMBER                        NEW        
dfi_3.6  3 idx2 REVENUS_YEAR                        NEW        
dfi_3.7  3 idx2  COEFFICIENT                        NEW        
dfi_4.1  4 idx1           ID                        NEW        
dfi_4.2  4 idx1         CITY                        NEW        
dfi_4.3  4 idx1       STREET                        NEW        
dfi_5.1  5 idx2           ID                        NEW        
dfi_5.2  5 idx2         NAME                        NEW        
dfi_5.3  5 idx2          AGE                        NEW        
dfi_5.4  5 idx2      CC_TYPE                        NEW        
dfi_5.5  5 idx2    CC_NUMBER                        NEW        
dfi_5.6  5 idx2 REVENUS_YEAR                        NEW        
dfi_5.7  5 idx2  COEFFICIENT                        NEW        
> |
```

6.Visualiser le data lineage au niveau de dataframes



```
dtf<-treeDtf()
showLineage(dtf, str_type = "dataframes")
```

7.Visualiser le lineage pour chaque colonne

```
s1<-showLineage(dtf, str_type = "columns")
```