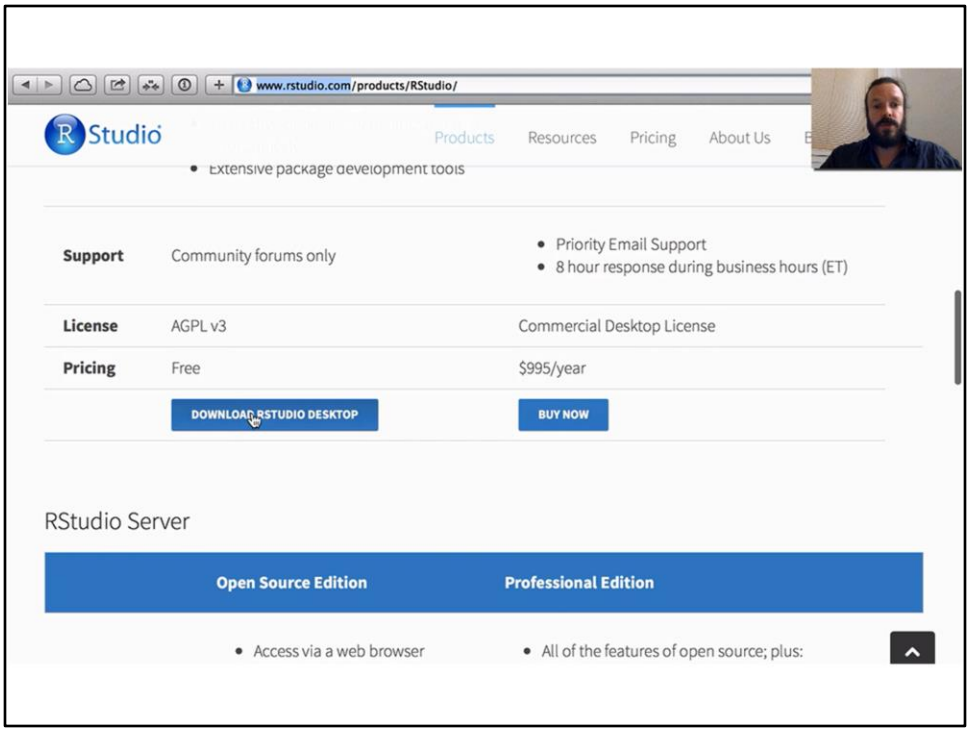
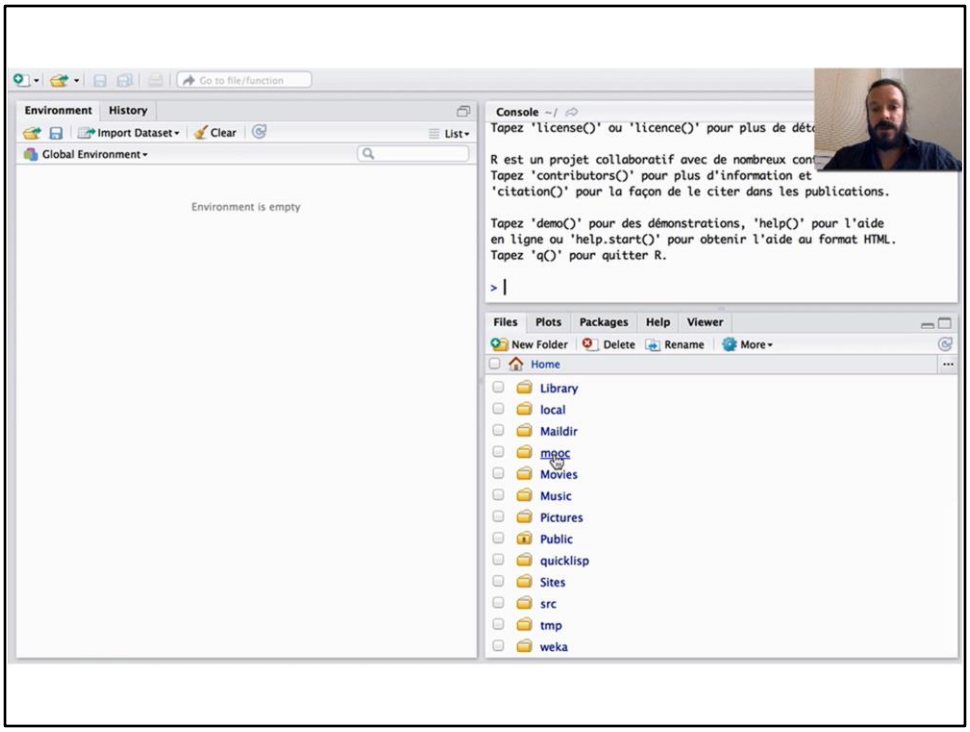
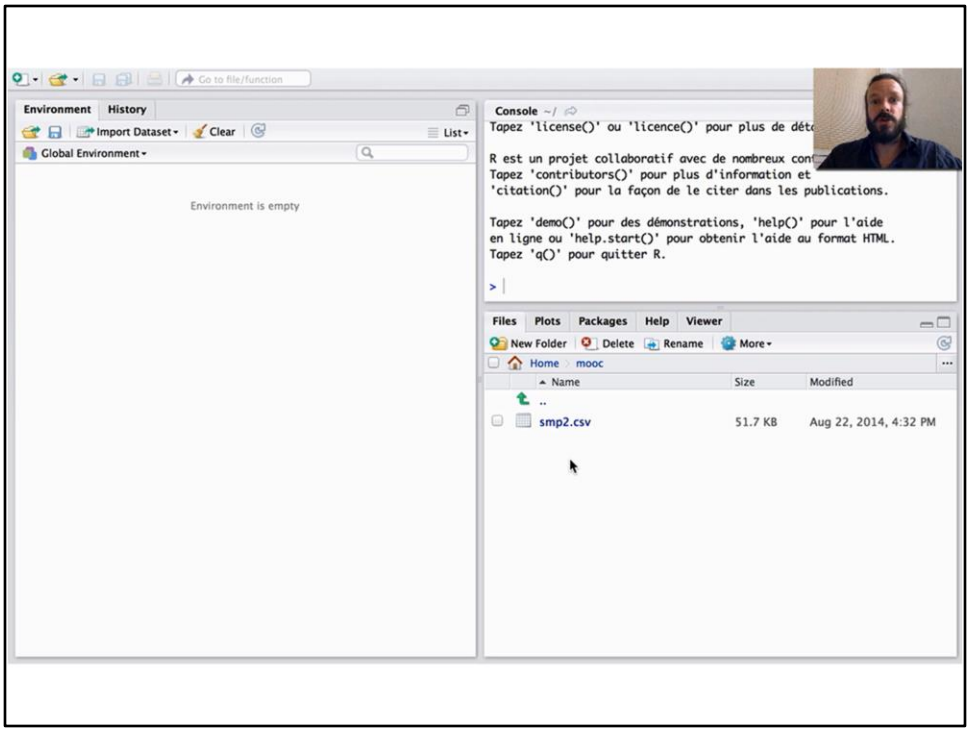
[00:01] Dans cette première session, on va s'intéresser au langage de base donc :
- comment importer des données enregistrées, par exemple, dans un fichier Excel ?
- comment manipuler des variables de type numérique et des variables de type catégoriel ?



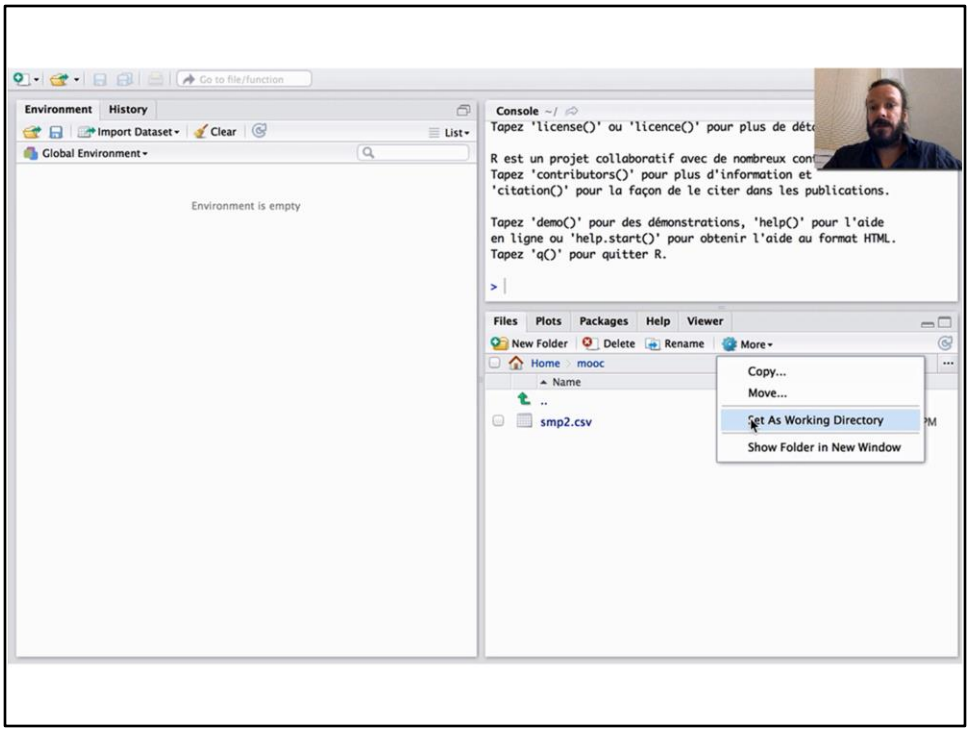
[00:14] Pour interagir avec R, on va utiliser le logiciel R Studio qui se télécharge sur le site rstudio.com. En bas de la page, on trouvera un lien pour le téléchargement. C'est un logiciel qui fonctionne sous Linux, sous Windows et sous Mac et qui, globalement, facilite grandement l'interaction avec R.



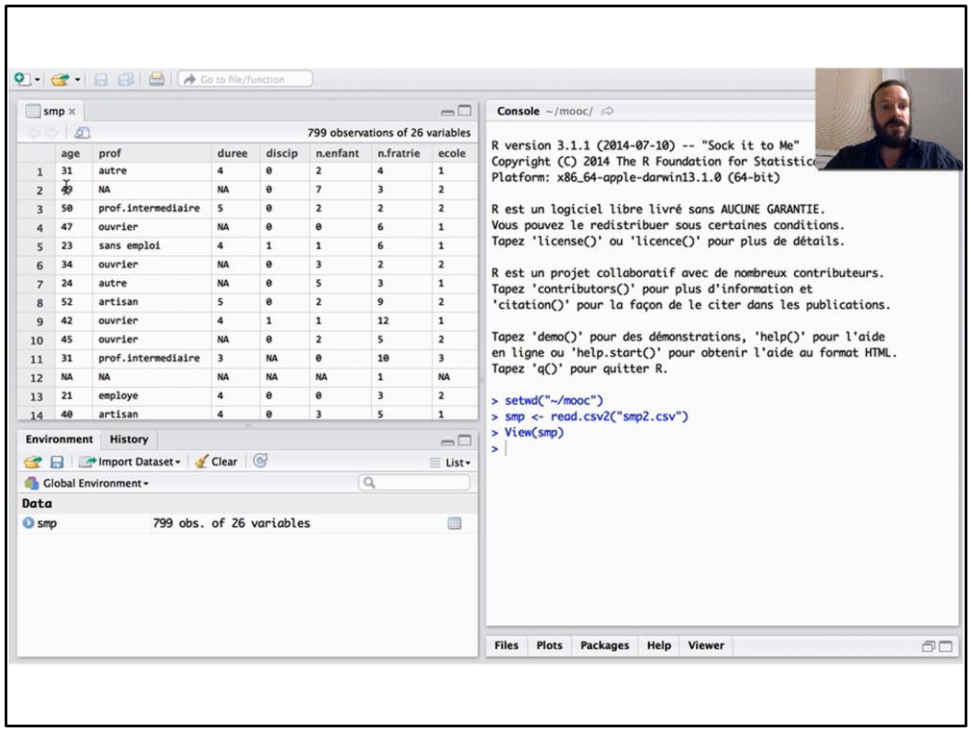
[00:30] On a ici un panneau qui s'appelle « console » (et qui est globalement la console que vous avez lorsque vous lancez R depuis Mac ou Windows), dans laquelle on peut taper des commandes. On va avoir accès à un explorateur de fichiers ; donc, pour toute la durée de ces sessions, on va créer un fichier ou on va supposer que vous avez créé un répertoire qui s'appelle « mooc »



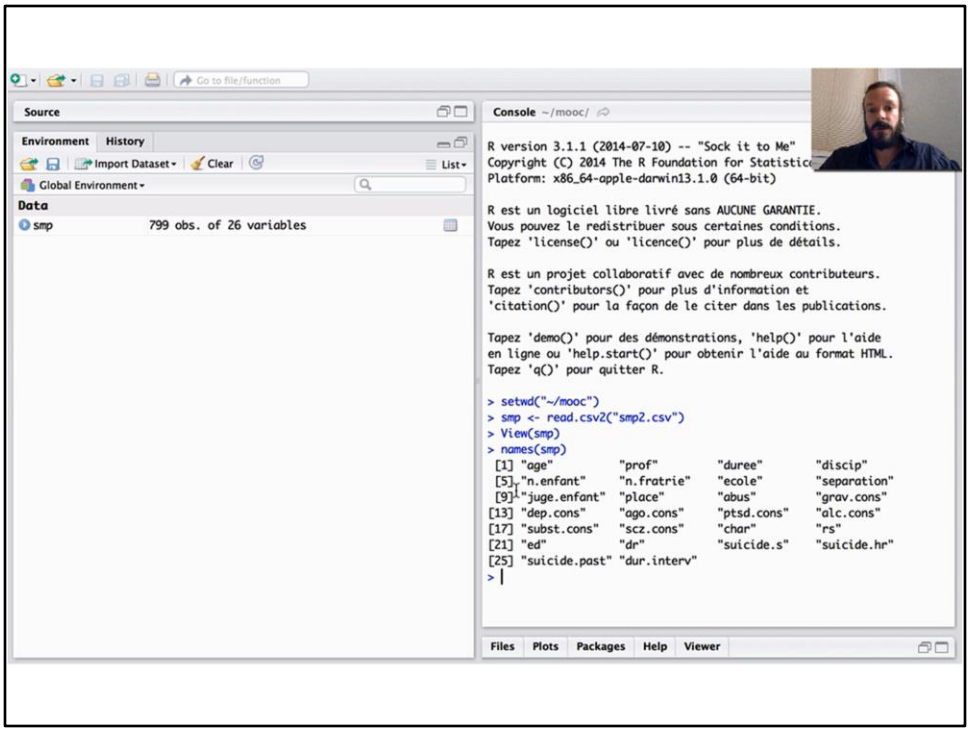
[00:50] dans lequel vous avez enregistré le fichier « smp2.csv » que vous pouvez trouver en téléchargement sur le site du cours. C'est un fichier tabulé qui a été, par exemple, généré à partir d'Excel et qui comporte des variables en colonnes et des observations en lignes.



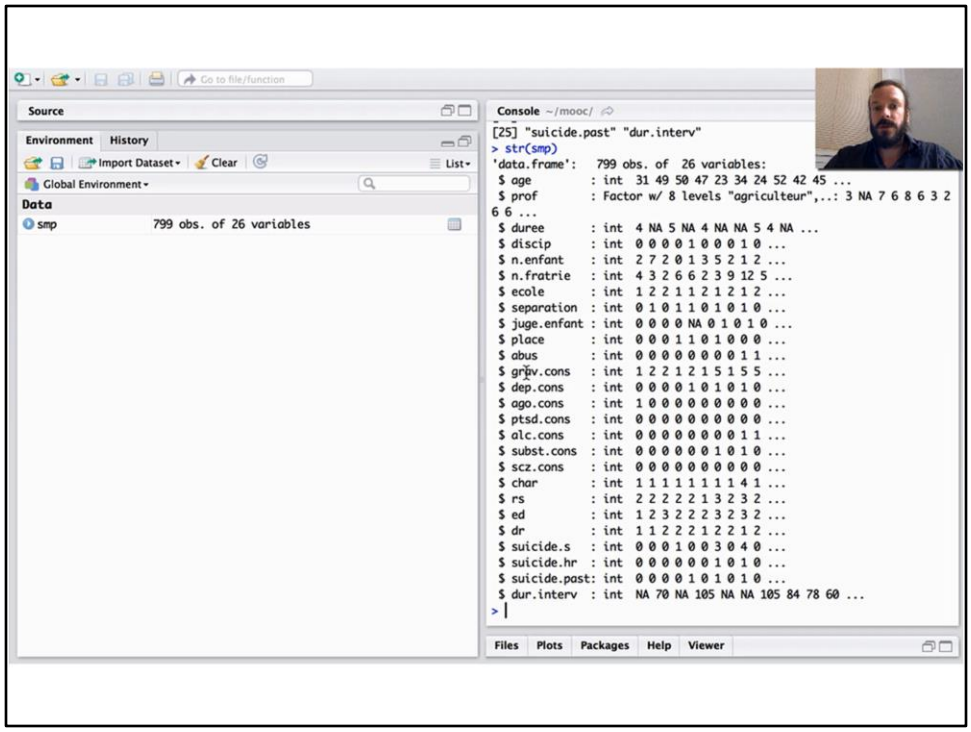
[01:07] Ce que l'on va faire dans un premier temps, c'est définir ce répertoire-là comme le répertoire de travail



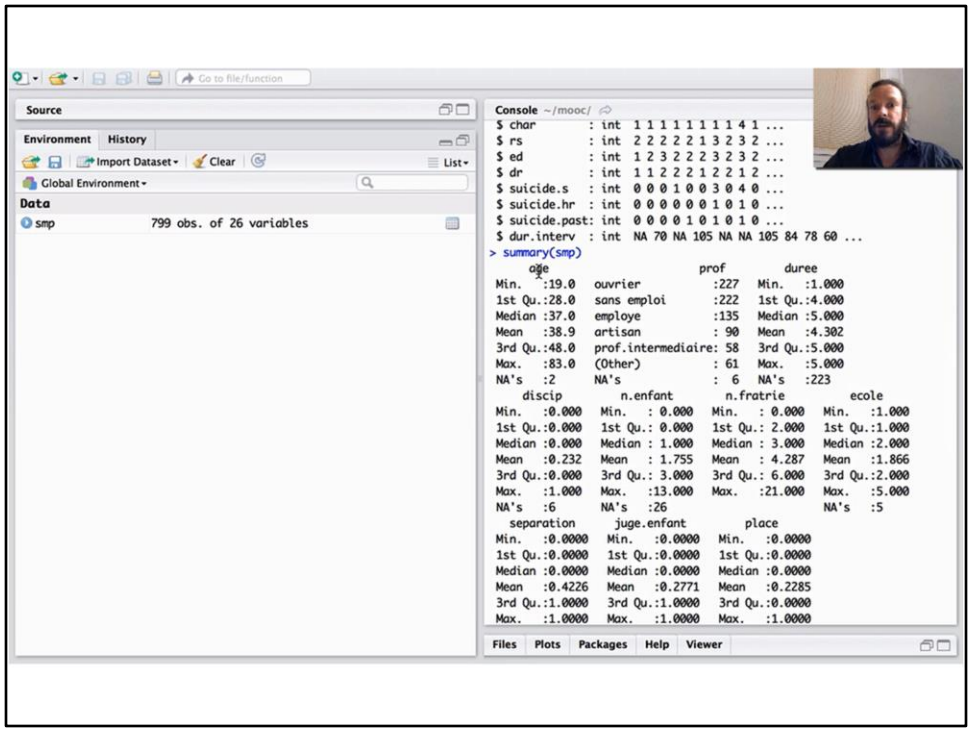
[01:12] et charger le fichier en tapant la commande `read.csv2()`. On va associer ce fichier à une variable que l'on appellera « smp » et on utilisera toujours ce nom de variable pour tous les labs. Donc ici on va taper simplement le début du nom de fichier et taper sur la touche « tab » qui permet de compléter automatiquement les noms de fichier ou les noms de commande. On voit que la variable « smp », qui est ce que l'on appelle un « data frame » sous R, comprend 799 observations et 26 variables. On peut même visualiser directement les données à l'aide du visualisateur interne. On a par exemple la première variable, ici « age », pour laquelle on a les observations. Donc le premier individu a 31 ans, le deuxième a 49 ans.



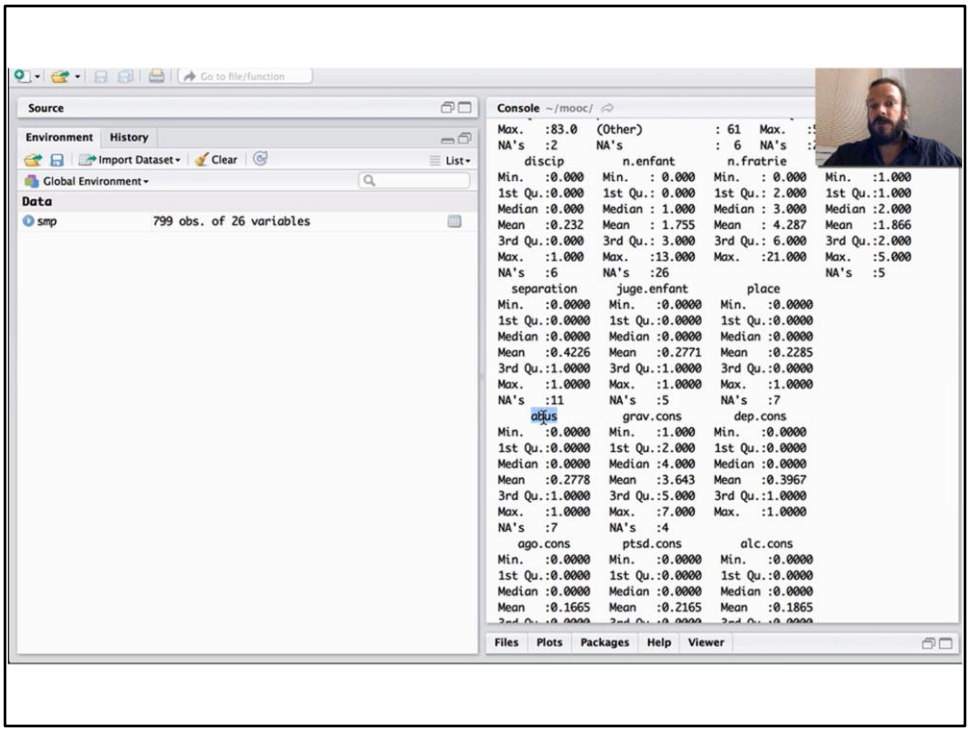
[01:59] Pour avoir accès au nom des variables, on utilise la commande `names()` et R va nous renvoyer le nom de l'ensemble des variables qui sont contenues dans le data frame.



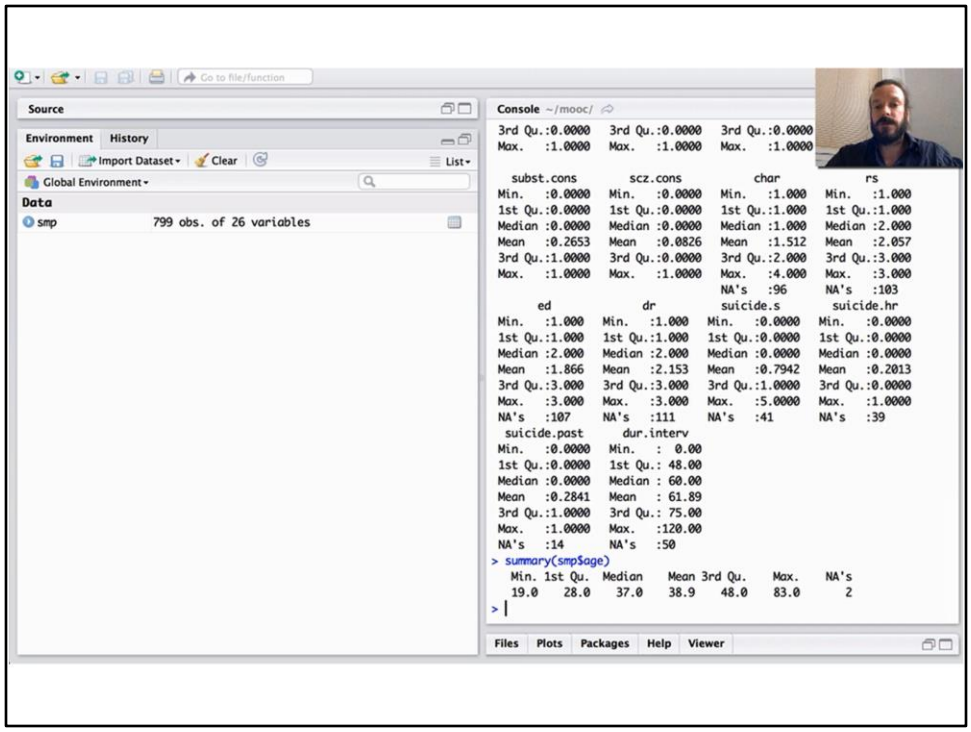
[02:08] On peut également utiliser la commande `str()` qui nous permet d'afficher, pour chacune des variables, son mode de représentation. Ici une variable quantitative, donc des nombres, et pour la profession, une variable qualitative que R appelle des « facteurs » avec 8 niveaux. On a généralement accès à un aperçu des premières observations.



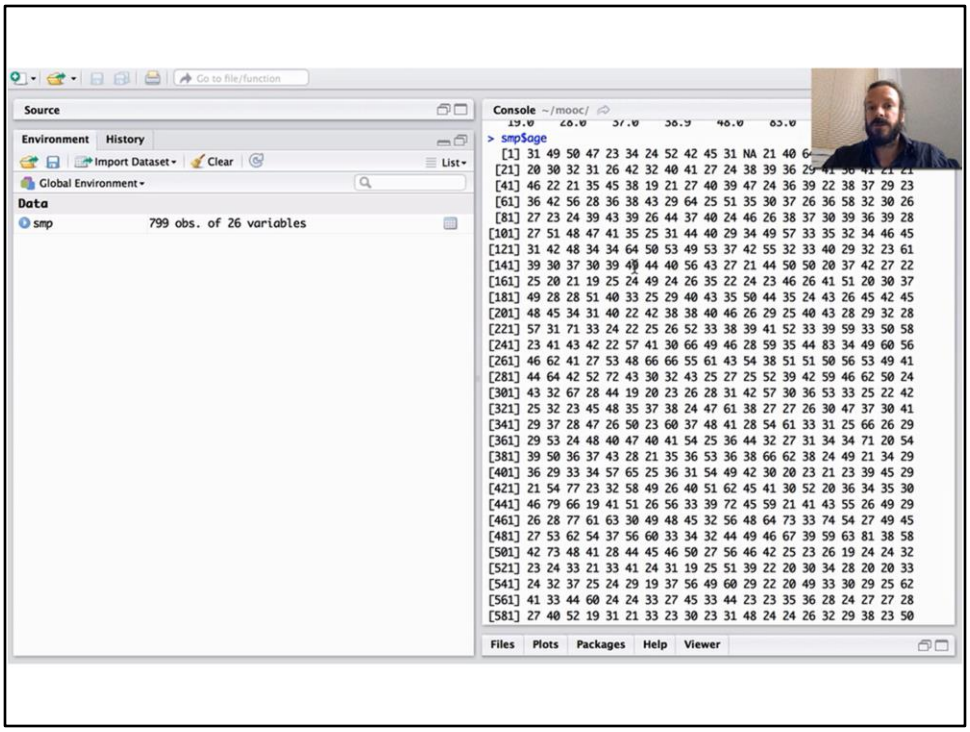
[02:32] On peut également utiliser la commande `summary()` qui va nous fournir un résumé numérique univarié pour chacune des variables (pour les variables numériques : les indicateurs de tendance centrale, de dispersion et d'étendue et pour les variables qualitatives : un tableau d'effectifs associés à chacune des modalités). On voit par exemple qu'ici on a une variable numérique, ici une variable catégorielle



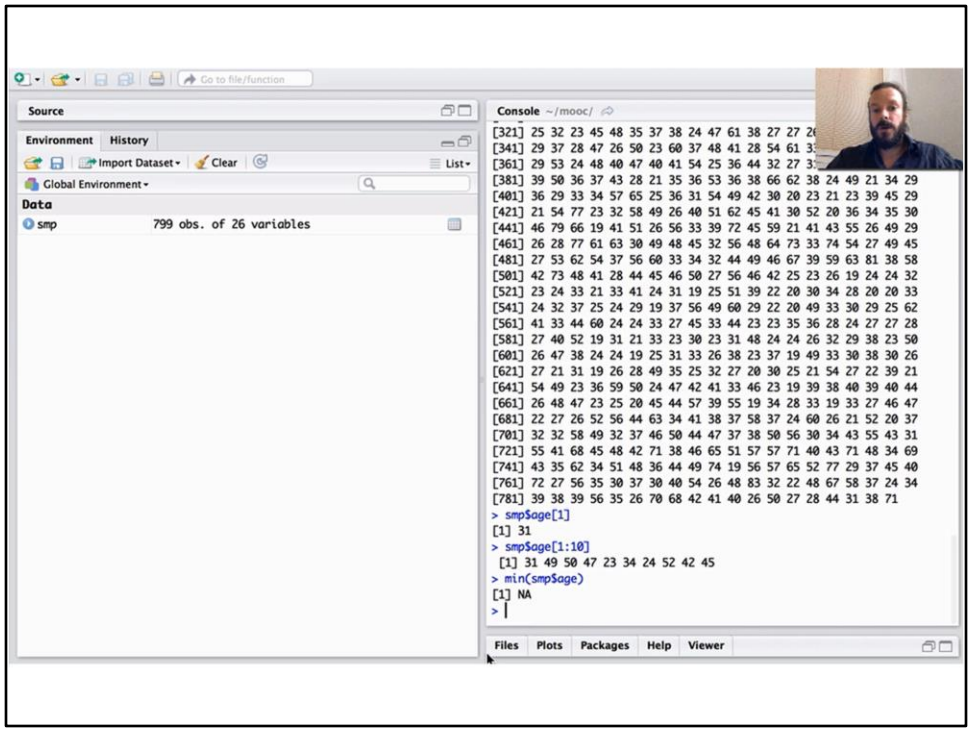
[02:55] et par exemple la variable « abus » est une variable binaire mais R, ici, l'a traitée comme une variable numérique ; on y reviendra juste après.



[03:01] La commande `summary()` fonctionne également pour les variables directement et non pas seulement pour les data frames. Pour accéder à une variable sous R, on tapera le nom du data frame suivi de « dollar » et suivi du nom de la variable qui nous intéresse. Donc ici, la commande `summary()` est appliquée directement à la variable « age ». Le minimum vaut 19, le maximum vaut 83 et on a ici deux valeurs manquantes qui sont représentées par le symbole « NA ».

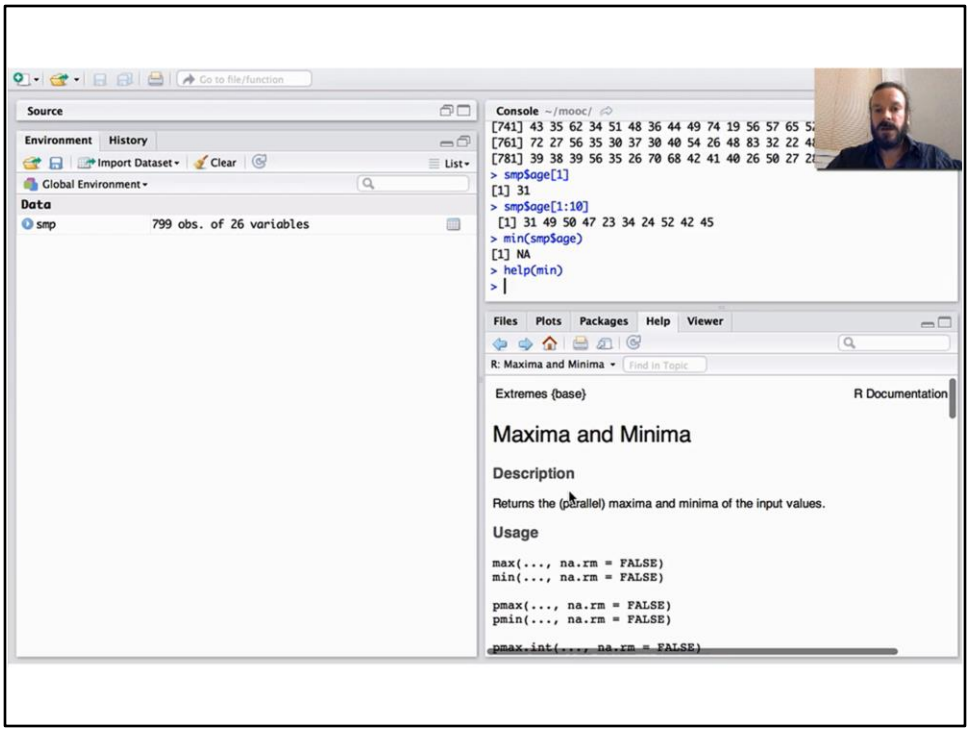


[03:28] On peut très bien taper directement `smp$age` mais dans ces cas-là, R va nous renvoyer l'ensemble des observations, ce qui n'est pas toujours très pratique.

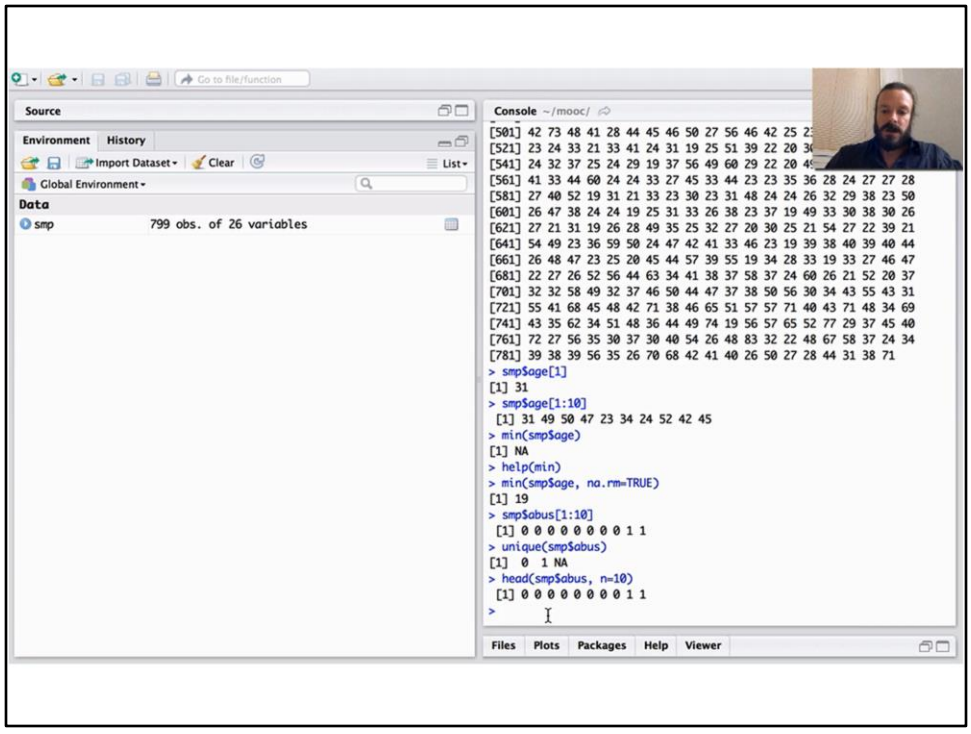


[03:28] On peut, en revanche, n'afficher que la première observation. Dans ces cas-là, on mettra entre crochets le numéro d'observation qui nous intéresse. Ici la première observation ; on peut vérifier qu'il s'agit bien d'un âge de 31. On peut également indiquer 1:10, c'est-à-dire de la première à la dixième observation.
Notez ici que ce n'est pas toujours la peine de retaper systématiquement les commandes. En utilisant les flèches « haut » et « bas », vous pouvez naviguer dans l'historique des commandes.

On peut par exemple chercher la valeur minimale pour l'âge et on s'aperçoit finalement que R va nous renvoyer la valeur « NA ».

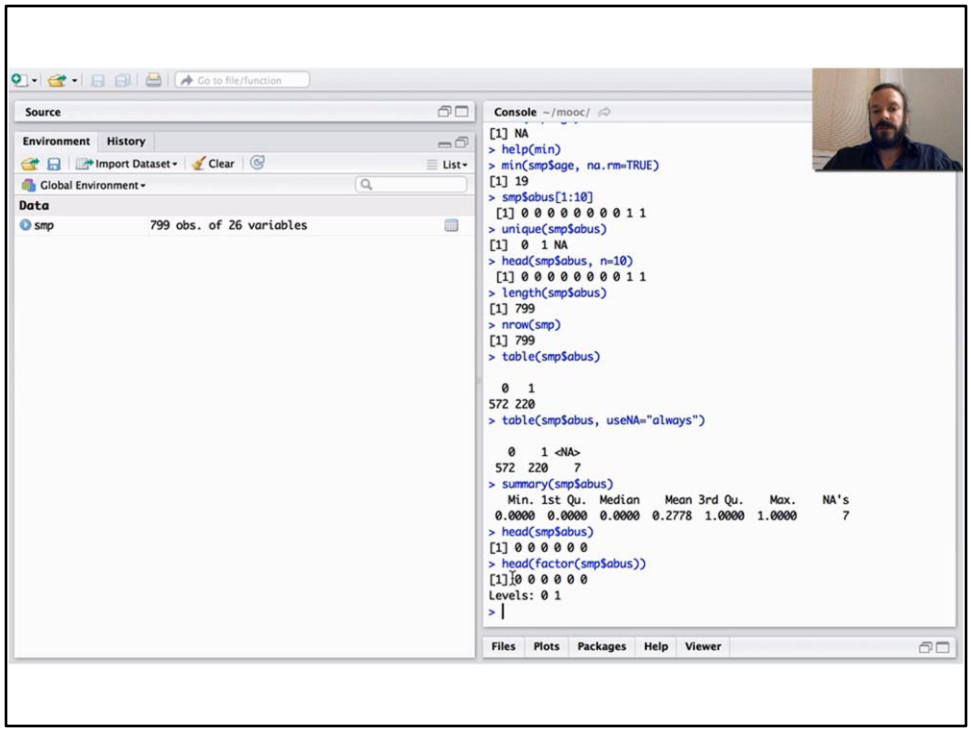


[04:16] Pourquoi ? Alors il suffit d'aller regarder l'aide, à l'aide de la commande `help()`, et de taper le nom de la commande qui nous intéresse. On s'aperçoit que lorsque R calcule le minimum, il n'enlève pas les valeurs manquantes. Dans ces cas-là, il renverra une valeur « NA » pour dire qu'il ne peut pas calculer l'âge minimum.



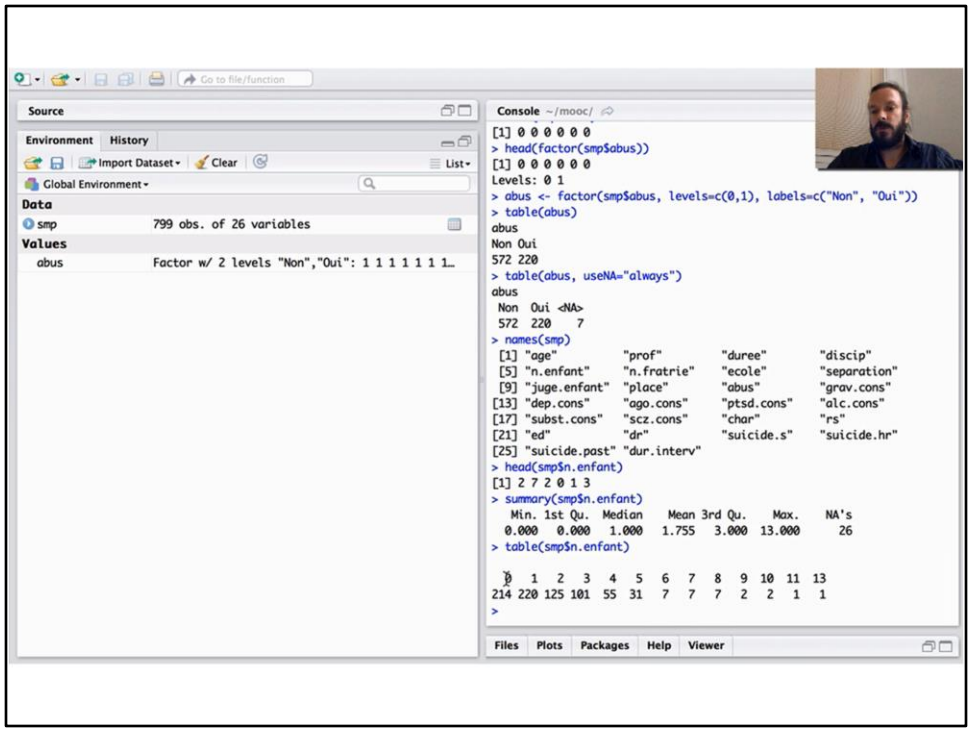
[04:33] On peut par contre préciser que, pour calculer la valeur minimale, on souhaite enlever les valeurs manquantes, en rajoutant l'option `na.rm=TRUE`.

Si on s'intéresse maintenant à la variable « abus », on peut regarder par exemple les premières valeurs de la variable « abus ». On voit qu'on a des valeurs en 0 et 1. On peut d'ailleurs utiliser la commande `unique()` pour lister l'ensemble des modalités uniques qui sont observées pour cette variable. D'ailleurs, plutôt que de taper directement `smp$abus[1:10]`, on peut très bien utiliser ~~la commande smp~~, la commande `head()` pardon, avec smp et indiquer qu'on veut afficher les dix premières valeurs.



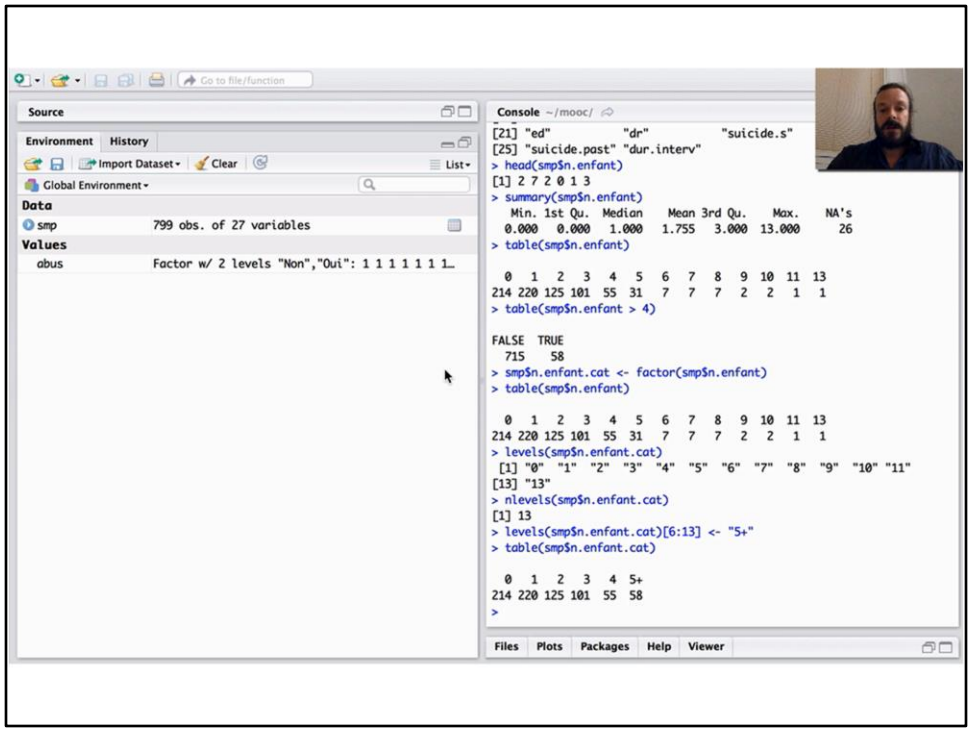
[05:24] Nous avons donc une variable, « abus », qui est contenue dans le data frame « smp ». Le nombre total d'observations s'obtient avec la commande `length()` par exemple. Ça correspond globalement au nombre de lignes de notre tableau « smp ». On peut utiliser la commande `table()` qui va nous renvoyer le tableau d'effectifs associés à chacune des modalités. Or on voit ici qu'une des modalités qui avait été listée est la valeur « NA », ce qui suggère qu'il y a des valeurs manquantes pour notre variable « smp$abus ». Donc, lorsqu'on utilise la commande `table()`, on utilisera toujours `useNA="always"` comme option pour être sûr de bien afficher les valeurs manquantes. Ici, R nous liste 7 valeurs manquantes pour cette variable-là.

La variable ici est toujours traitée comme une variable numérique. D'ailleurs si on fait `summary(smp$abus)`, on a bien un résumé avec les autres indicateurs d'étendue. Souvent on préfèrerait que cette variable-là soit bien traitée comme une variable qualitative et pour ça, on va utiliser la commande `factor()`. Donc, lorsqu'on regarde les premières observations, on va juste remplacer notre variable en utilisant la commande `factor()`. Ce qui va changer : R ne change rien aux valeurs de la variable mais il va lui associer des niveaux et ici les niveaux sont « 0 » et « 1 ».

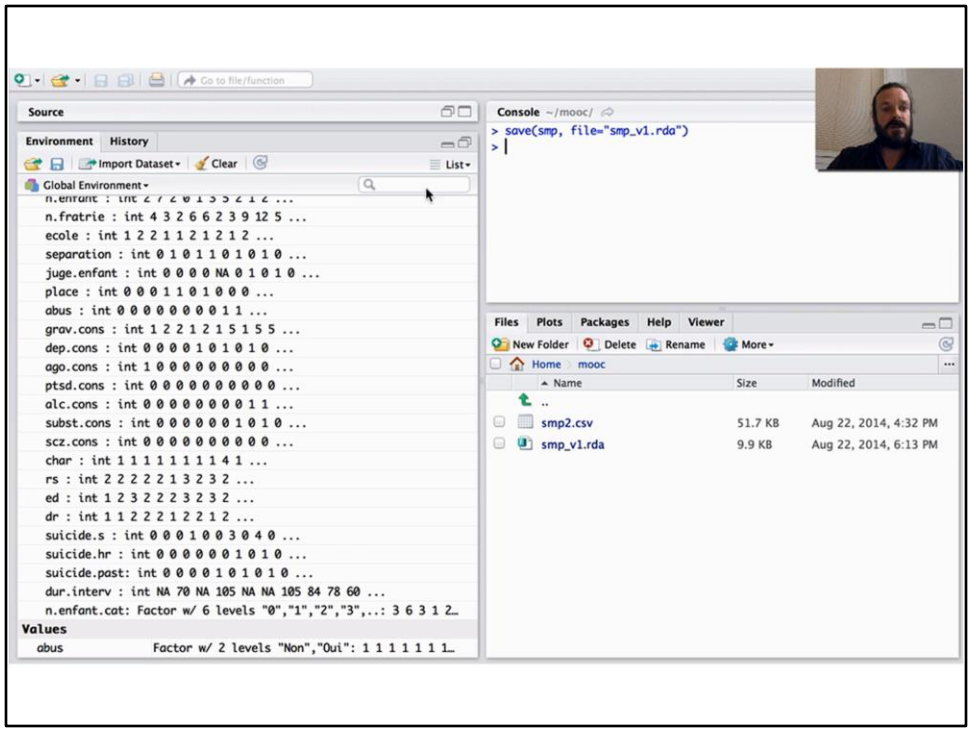


[06:47] Donc ce qu'on va faire, c'est par exemple créer une nouvelle variable et dire que c'est la variable « smp$abus » mais traitée comme facteur. Il faut rajouter par contre, c'est que l'on va lui spécifier que les niveaux qu'il a associés, donc « 0 » et « 1 », vont être associés aux étiquettes « non » et « oui ». Donc la variable « abus » a été créée dans l'espace de travail ; elle est séparée, distincte, du data frame et on voit maintenant que notre variable a donc les modalités « non » et « oui » qui ont été associées aux niveaux « 0 » et « 1 ». On peut toujours lister les valeurs manquantes séparément.

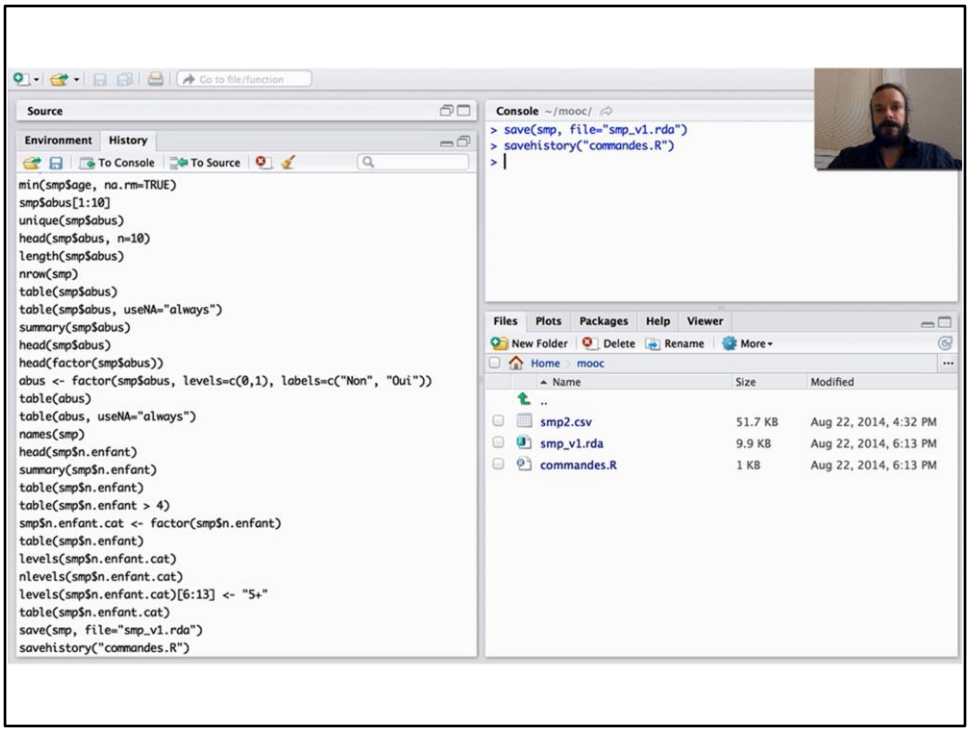
[07:31] Regardons maintenant une autre variable qualitative, par exemple le nombre d'enfants. Le nombre d'enfants qui a été rapporté par les répondants. Donc si on regarde les premières observations de cette variable-là, on voit qu'on a des valeurs numériques. D'ailleurs on peut utiliser `summary()` et vérifier que R considère que cette variable-là est une variable numérique. Mais maintenant regardons effectivement la répartition des effectifs : on s'aperçoit que le nombre d'enfants minimal vaut « 0 » et le nombre d'enfants maximal vaut « 13 ». C'est ce que l'on avait dans le résumé numérique précédent.



[08 :12] Et on peut regarder également le nombre d'enfants qui sont supérieurs à « 4 » par exemple. Donc on a 58 valeurs qui remplissent la condition « le nombre d'enfants est supérieur à 4 », ce qui correspond globalement à l'ensemble de ces valeurs-là. Donc ce qu'on pourrait très bien faire, c'est créer une nouvelle variable « nombre d'enfants » qu'on va appeler « cat », qui est en fait notre variable « nombre d'enfants » traitée cette fois-ci comme un facteur et pour laquelle on va simplement dresser un tableau d'effectifs. Cette fois-ci on peut vérifier que la variable a bien des niveaux qui lui sont associés. Donc ici j'ai pris la valeur « n.enfant », il faudrait prendre la valeur « n.enfant.cat ». Donc on a bien 13 niveaux qui ont été associés. On peut d'ailleurs vérifier le nombre de niveau avec la commande `nlevels()`. Supposons maintenant qu'on souhaite agréger les derniers niveaux ; on va simplement reprendre l'instruction `levels()` et on va indiquer que pour les niveaux allant de 6 à 13, on va considérer que c'est une modalité unique qui s'appelle « 5+ ». Si maintenant on redresse un tableau d'effectifs de notre nouvelle variable (si on appuie sur la touche « tab » on a deux choix possibles, donc on va prendre « n.enfant.cat »), on voit bien que les effectifs ici ont été agrégés dans la même classe, donc ça correspond à tous ces effectifs-là. On peut faire exactement la même chose avec une variable numérique comme c'est indiqué dans le tutoriel en version pdf pour ce lab (fichier labs.pdf).



[09:48] Maintenant on va simplement sauvegarder notre fichier de données « smp » en format R (on voit qu'on a une variable « n.enfant.cat » qui a été ajoutée à ce fichier) en utilisant la commande `save()`. On va lui donner le nom du data frame suivi du nom du fichier. On va appeler ça « smp_v1.rda » et on peut vérifier que dans notre répertoire de travail, on a bien un fichier « smp_v1.rda » qui a été créé.



[10:18] On peut faire la même chose avec l'historique (ici R enregistre automatiquement toutes les commandes qu'on tape) et pour ça on va utiliser la commande `savehistory()` et on va simplement donner le nom d'un fichier, qu'on va appeler ici « commandes.R ».