Homework 10

Ameth FAYE

2024-03-30

```
#Exercice de maison 10:

source("C:/Users/HP/Desktop/ISEP2CoursR2024/Homework 7.R")
```

```
## New names:
## * `` -> `...8`
## * `` -> `...9`
```

```
## Warning: le package 'dplyr' a été compilé avec la version R 4.3.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
## Rows: 11,114
## Columns: 14
## $ interview__key         <chr> "55-75-97-43", "55-75-97-43", "55-75-97-43", "1~
## $ interview__id          <chr> "3827f11c978e42e6aa4b5b8ed9aa1348", "3827f11c97~
## $ cereales__id           <dbl+lbl>  1,  7, 20,  4, 22,  1,  7, 16, 21,  1, 20,~
## $ s07Bq02_autre_cereales <chr> "", "", "", "", "", "", "", "", "", "", "",~
## $ s07Bq03a_cereales      <dbl> 9.00, 7.00, 2.00, 7.50, 42.00, 14.00, 10.00, 0.~
## $ s07Bq03b_cereales      <dbl+lbl> 100, 100, 139, 100, 568, 100, 100, 100, 568~
## $ s07Bq03c_cereales      <dbl+lbl> 0, 0, 2, 0, 3, 0, 0, 0, 3, 0, 2, 3, 0, 2, 0~
## $ s07Bq04_cereales       <dbl> 0, 7, NA, NA, 0, 0, 10, 0, 0, 0, NA, 0, 0, 7, 0~
## $ s07Bq05_cereales       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ s07Bq06_cereales       <dbl+lbl> 2, 4, 2, 1, 1, 2, 4, 1, 1, 1, 2, 1, 4, 4, 2~
## $ s07Bq07a_cereales      <dbl> 1.00, NA, 2.00, 2.50, 6.00, 2.00, NA, 0.25, 1.0~
## $ s07Bq07b_cereales      <dbl+lbl> 100,  NA, 139, 100, 568, 100,  NA, 100, 568~
## $ s07Bq07c_cereales      <dbl+lbl>  0, NA,  2,  0,  3,  0, NA,  0,  3,  0,  2,~
## $ s07Bq08_cereales       <dbl> 350, NA, 600, 750, 1050, 325, NA, 200, 200, 120~
## Rows: 11,104
## Columns: 15
## $ interview__key <chr> "55-75-97-43", "55-75-97-43", "55-75-97-43", "14-16-03-~
## $ interview__id  <chr> "3827f11c978e42e6aa4b5b8ed9aa1348", "3827f11c978e42e6aa~
```

```
## $ cereales__id   <dbl+lbl>  1,  7, 20,  4, 22,  1,  7, 16, 21,  1, 20, 21,  1,~
## $ AutresCereales <chr> "", "", "", "", "", "", "", "", "", "", "", "", "",~
## $ Qtty_cons      <dbl> 9.00, 7.00, 2.00, 7.50, 42.00, 14.00, 10.00, 0.25, 7.00~
## $ Unite_cons     <dbl+lbl> 100, 100, 139, 100, 568, 100, 100, 100, 568, 100, 1~
## $ Taille_cons    <dbl+lbl> 0, 0, 2, 0, 3, 0, 0, 0, 3, 0, 2, 3, 0, 2, 0, 3, 0, ~
## $ AutoCons       <dbl> 0, 7, NA, NA, 0, 0, 10, 0, 0, 0, NA, 0, 0, 7, 0, 0, 0, ~
## $ AutresProv     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ DernierAchat   <dbl+lbl> 2, 4, 2, 1, 1, 2, 4, 1, 1, 1, 2, 1, 4, 4, 2, 1, 1, ~
## $ Qtty_achat     <dbl> 1.00, NA, 2.00, 2.50, 6.00, 2.00, NA, 0.25, 1.00, 3.00,~
## $ Unite_achat    <dbl+lbl> 100,  NA, 139, 100, 568, 100,  NA, 100, 568, 100, 1~
## $ Taille_achat   <dbl+lbl>  0, NA,  2,  0,  3,  0, NA,  0,  3,  0,  2,  3, NA,~
## $ Value_achat    <dbl> 350, NA, 600, 750, 1050, 325, NA, 200, 200, 1200, 800, ~
## $ produit1       <fct> 1, 7, 20, 4, 22, 1, 7, 16, 21, 1, 20, 21, 1, 6, 7, 21, ~
## Rows: 11,104
## Columns: 16
## $ interview__key <chr> "55-75-97-43", "55-75-97-43", "55-75-97-43", "14-16-03-~
## $ interview__id  <chr> "3827f11c978e42e6aa4b5b8ed9aa1348", "3827f11c978e42e6aa~
## $ cereales__id   <dbl+lbl>  1,  7, 20,  4, 22,  1,  7, 16, 21,  1, 20, 21,  1,~
## $ AutresCereales <chr> "", "", "", "", "", "", "", "", "", "", "", "", "",~
## $ Qtty_cons      <dbl> 9.00, 7.00, 2.00, 7.50, 42.00, 14.00, 10.00, 0.25, 7.00~
## $ Unite_cons     <dbl+lbl> 100, 100, 139, 100, 568, 100, 100, 100, 568, 100, 1~
## $ Taille_cons    <dbl+lbl> 0, 0, 2, 0, 3, 0, 0, 0, 3, 0, 2, 3, 0, 2, 0, 3, 0, ~
## $ AutoCons       <dbl> 0, 7, NA, NA, 0, 0, 10, 0, 0, 0, NA, 0, 0, 7, 0, 0, 0, ~
## $ AutresProv     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ DernierAchat   <dbl+lbl> 2, 4, 2, 1, 1, 2, 4, 1, 1, 1, 2, 1, 4, 4, 2, 1, 1, ~
## $ Qtty_achat     <dbl> 1.00, NA, 2.00, 2.50, 6.00, 2.00, NA, 0.25, 1.00, 3.00,~
## $ Unite_achat    <dbl+lbl> 100,  NA, 139, 100, 568, 100,  NA, 100, 568, 100, 1~
## $ Taille_achat   <dbl+lbl>  0, NA,  2,  0,  3,  0, NA,  0,  3,  0,  2,  3, NA,~
## $ Value_achat    <dbl> 350, NA, 600, 750, 1050, 325, NA, 200, 200, 1200, 800, ~
## $ produit1       <fct> 1, 7, 20, 4, 22, 1, 7, 16, 21, 1, 20, 21, 1, 6, 7, 21, ~
## $ produit        <fct> "Riz local brisé", "Mil", "Pâtes alimentaires", "Riz im~
```

```
Table_de_conversion <- read_excel("C:/Users/HP/Desktop/ISEP2CoursR2024/Base/Table de conversion phase 2
```

```
## New names:
## * `` -> `...8`
## * `` -> `...9`
```

```
colnames(Table_de_conversion) [1:6] <- c("cereales__id","Nom_Prod",
                                         "Unite_cons","Nom_Unite",
                                         "Taille_cons","Nom_Taille")
cereales <- merge(cereales, Table_de_conversion,
                  by = c("cereales__id", "Unite_cons", "Taille_cons"), all.x = TRUE)


# Traitement de la base de données

library(data.table)
```

```
##
## Attachement du package : 'data.table'
##
## Les objets suivants sont masqués depuis 'package:lubridate':
```

```
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## Les objets suivants sont masqués depuis 'package:dplyr':
##
##     between, first, last
##
## L'objet suivant est masqué depuis 'package:purrr':
##
##     transpose
```

```
cereales4 <- data.table(cereales)
setnames(cereales4 ,"poids","poids_cons")

cereales4 [,poids_cons:=as.numeric(poids_cons)] # Conversion de la variable poids_cons
cereales4 [,qtty_cons_kg:= poids_cons*Qtty_cons/1000] ## Quantité consommée en kg

#' calculer la quantite achete en kg;

cereales4 <- cereales4 [, Unite_achat := as.double(Unite_achat)]
cereales4 <- cereales4 [, Taille_achat := as.double(Taille_achat)]
cereales4 <- cereales4 [, cereales__id := as.double(cereales__id)]

colnames(Table_de_conversion) [1:6] <- c("cereales__id","Nom_Prod",
                                         "Unite_achat","Nom_Unite",
                                         "Taille_achat","Nom_Taille")
glimpse(Table_de_conversion)
```

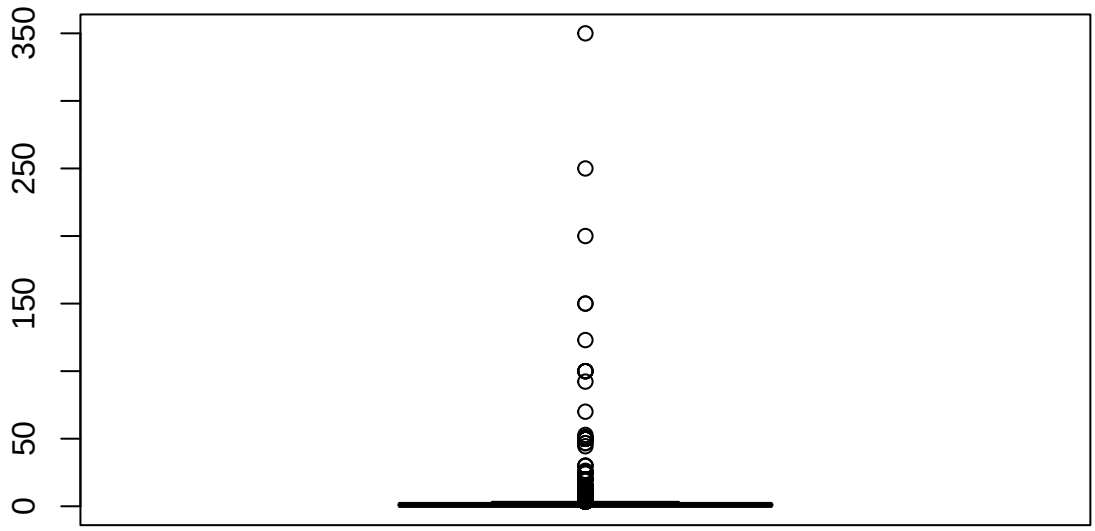
```
## Rows: 1,367
## Columns: 9
## $ cereales__id <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, ~
## $ Nom_Prod     <chr> "Riz local brisé", "Riz local brisé", "Riz local brisé", ~
## $ Unite_achat  <dbl> 100, 108, 108, 108, 108, 136, 138, 139, 100, 108, 108, 10~
## $ Nom_Unite    <chr> "Kg", "Boite de tomate/Pot", "Boite de tomate/Pot", "Boit~
## $ Taille_achat <dbl> 0, 1, 2, 3, 7, 0, 0, 2, 0, 1, 2, 3, 7, 0, 0, 2, 0, 1, 2, ~
## $ Nom_Taille   <chr> "taille unique", "Petit", "Moyen", "Grand", "Tres petit",~
## $ poids        <chr> "1000", "521.5", "935", "1997.5", "222", "25000", "50000"~
## $ ...8         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ ...9         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
```

```
cereales4 <- merge(cereales4 , Table_de_conversion,
                   by = c("cereales__id", "Unite_achat", "Taille_achat"), all.x = TRUE)

cereales4 <- data.table(cereales4 )
glimpse(cereales4)
```

```
## Rows: 11,104
## Columns: 33
## $ cereales__id      <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ Unite_achat       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ Taille_achat      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ Unite_cons        <dbl+lbl> 100, 100, 100, 100, 100, 100, 100, 100, 100, 100~
```

```
## $ Taille_cons       <dbl+lbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ interview__key    <chr> "44-02-17-24", "66-45-81-17", "86-55-10-95", "63-35-~
## $ interview__id     <chr> "b469453f8410449290ba3cd8ca1c4833", "3e6feed83915499~
## $ AutresCereales    <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", ~
## $ Qtty_cons         <dbl> 14.00, 1.50, 10.50, 12.00, 10.50, 25.00, 7.00, 7.00,~
## $ AutoCons          <dbl> 0.00, 1.50, 0.00, 0.00, 0.00, 25.00, 0.00, 0.00, 10.~
## $ AutresProv        <dbl> 0.0, 0.0, 10.5, 0.0, 0.0, 0.0, 0.0, 7.0, 0.0, 0.0, 0~
## $ DernierAchat      <dbl+lbl> 4, 4, 4, 4, 4, 5, 4, 4, 4, 5, 4, 4, 4, 4, 4, 4, ~
## $ Qtty_achat        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ Value_achat       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ produit1          <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ produit           <fct> "Riz local brisé", "Riz local brisé", "Riz local bri~
## $ unite_cons        <fct> Kg, Kg, Kg, Kg, Kg, Kg, Kg, Kg, Kg, Kg, Kg, Kg, Kg, ~
## $ taille_cons       <fct> Taille unique, Taille unique, Taille unique, Taille ~
## $ classCereal       <fct> Tres faible, Tres faible, Tres faible, Tres faible, ~
## $ classCereal_RizKg <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ Nom_Prod.x        <chr> "Riz local brisé", "Riz local brisé", "Riz local bri~
## $ Nom_Unite.x       <chr> "Kg", "Kg", "Kg", "Kg", "Kg", "Kg", "Kg", "Kg", "Kg"~
## $ Nom_Taille.x      <chr> "taille unique", "taille unique", "taille unique", "~
## $ poids_cons        <dbl> 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000~
## $ ...8.x            <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ ...9.x            <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ qtty_cons_kg      <dbl> 14.00, 1.50, 10.50, 12.00, 10.50, 25.00, 7.00, 7.00,~
## $ Nom_Prod.y        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ Nom_Unite.y       <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ Nom_Taille.y      <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ poids             <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ ...8.y            <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ ...9.y            <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
```

```
setnames(cereales4 , "poids_cons", "poids_achat")
cereales4 <- cereales4 [,poids_achat:=as.numeric(poids_achat)]


cereales4 [, qtty_achat_kg := Qtty_achat*poids_achat/1000]
boxplot(cereales4 $qtty_achat_kg)
```




```
#' calculer le prix unitaire ;
## Un prix unitaire pour chaque combinaison (produit,unite, taille)

cereales4 $pu <- cereales4 $Value_achat/cereales4 $Qtty_achat

### Extraire les Prix


prixunitaire <- subset(cereales4 , !is.na(pu),
                       select =c("cereales__id", "Unite_achat", "Taille_achat", "pu") )

## Traitement des pu aberrants ;

idc <- unique(cereales4 $cereales__id)

library(dplyr)
cereales4 <- cereales4 %>%
  group_by(cereales__id) %>%
  mutate(pu = ifelse(!is.na(pu) & pu > quantile(pu, 0.75, na.rm = TRUE), quantile(pu, 0.75, na.rm = TRU

# Calculer la moyenne et la médiane de 'pu' pour chaque combinaison (p,u,t)
library(dplyr)

resultats <- prixunitaire %>%
  group_by(pu, Unite_achat, Taille_achat) %>%
  summarise(
```

```
    mediane_pu = median(pu, na.rm = TRUE)
  )
```

```
## `summarise()` has grouped output by 'pu', 'Unite_achat'. You can override using
## the `.groups` argument.
```

```
# Calculer le prix 'p' pour chaque combinaison (p,u,t)
prixunitaire2 <- prixunitaire %>%
  group_by(pu, Unite_achat, Taille_achat) %>%
  summarise(
    p = mean(pu, na.rm = TRUE)
  )
```

```
## `summarise()` has grouped output by 'pu', 'Unite_achat'. You can override using
## the `.groups` argument.
```

```
#' Ramener cette sous-base dans la base cereales4 pour calculer
#' les depenses de consommations ;

library(dplyr)

# Joindre la sous-base 'prixunitaire' à la base 'cereales4'
cereales4 <- cereales4 %>%
  left_join(prixunitaire2, by = c("pu", "Unite_achat", "Taille_achat"))

# Calculer les dépenses de consommation
cereales4 <- cereales4 %>%
  mutate(depenses = p * Qtty_achat)

#' 1:: evaluer le taux de matching : n(Pc,Uc,Tc) aynt un prix P sur le
#' le nombre total de combinaison n(Pc,Uc,Tc);


# Calculer le nombre total de combinaisons (Pc,Uc,Tc)
total_combinaisons <- nrow(cereales4 )

# Calculer le nombre de combinaisons (Pc,Uc,Tc) ayant un prix P
combinaisons_avec_prix <- cereales4 %>%
  filter(!is.na(p)) %>%
  nrow()

# Calculer le taux de correspondance
taux_correspondance <- combinaisons_avec_prix / total_combinaisons

# Afficher le taux de correspondance
print(taux_correspondance)
```

```
## [1] 0.8444705
```

```
#' 2:: Reflechir a comment valoriser ces quantites n'ayant de prix
```

```r
# Imputer les valeurs manquantes par la médiane
cereales4 $pu[is.na(cereales4 $pu)] <- median(cereales4 $pu, na.rm = TRUE)

#' Valeurs aberrantes :: corrections ;
Q1 <- quantile(cereales4 $pu, 0.25, na.rm = TRUE)
Q3 <- quantile(cereales4 $pu, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1

# Remplacer les valeurs aberrantes par la médiane
cereales4 $pu[cereales4 $pu < (Q1 - 1.5 * IQR) | cereales4 $pu > (Q3 + 1.5 * IQR)] <- median(cereales4 $
```