

Chapitre 9 Correction des exercices

9.1 Exercice 4.5.2

Exercice : Les données mensuelles sitadel

A partir du fichier sitadel de février 2017 (ROES_201702.xls), sur la région Pays de la Loire (code région 52), livrer un fichier contenant pour chaque mois, pour les logements individuels ($i_AUT = ip_AUT + ig_AUT$) :

- le cumul des autorisations sur 12 mois glissants (i_AUT_cum12)
- le taux d'évolution du cumul sur 12 mois ($i_AUT_cum_evo$, en %)
- la part de ce cumul dans celui de l'ensemble des logements autorisés (log_AUT), en pourcentage

```
rm (list = ls ())
```

```
sitadel <- read_excel("data/ROES_201702.xls", sheet = "AUT_REG",
                     col_types = c("text", "text", "numeric", "numeric", "numeric", "nume
```

```
sitadel52 <- filter (sitadel, REG == "52")
```

```
sitadel52 <- mutate (sitadel52,
```

```
  i_AUT = ip_AUT + ig_AUT, # somme des logements individuels autoris
  i_AUT_cum12 = roll_sumr (i_AUT, 12), # cumul sur 12 mois
  i_AUT_cum12_lag12 = lag (i_AUT_cum12, 12), # décalage de 12 mois
  i_AUT_cum12_delta = i_AUT_cum12 - i_AUT_cum12_lag12,
  i_AUT_cum_evo = round (100 * i_AUT_cum12_delta / i_AUT_cum12_lag1
  log_AUT_cum12 = roll_sumr (log_AUT, 12), # somme des logements au
  i_AUT_cum_part = round (100 * i_AUT_cum12 / log_AUT_cum12, 1) # p
)
```

9.2 Exercice 4.5.3

Convertir les données de la table exercice pertinentes au format date.

```
rm (list = ls ())  
load (file = "data/FormationPreparationDesDonnées.RData")  
exercice <- mutate (exercice,  
                     date_prelevement = ymd (date_prelevement),  
                     date_creation = ymd (date_creation),  
                     date_formatee = format (date_prelevement, "%d/%m/%Y")) # plus joli,
```

9.3 Exercices 4.10

9.3.1 Sitadel

Sur les données “sitadel”, effectuer les opérations suivantes en utilisant l’opérateur %>% :

- les mêmes calculs que ceux réalisés sur la région 52, mais sur chacune des régions
- les agrégations par année civile pour chacune des régions, puis leur taux d’évolution d’une année sur l’autre (exemple : $(val_{2015} - val_{2014}) / val_{2014}$)

```
rm (list = ls())

sitadel <- read_excel ("data/ROES_201702.xls", sheet = "AUT_REG",
                      col_types = c ("text","text","numeric","numeric","numeric","numeric"),
                      group_by (REG) %>%
mutate (i_AUT = ip_AUT + ig_AUT,
        i_AUT_cum12 = roll_sumr (i_AUT, 12),
        i_AUT_cum12_lag12 = lag (i_AUT_cum12, 12),
        i_AUT_cum12_delta = i_AUT_cum12 - i_AUT_cum12_lag12,
        i_AUT_cum_evo = round (100 * i_AUT_cum12_delta / i_AUT_cum12_lag12, 1),

        log_AUT_cum12 = roll_sumr (log_AUT, 12),
        i_AUT_cum_part = round (100 * i_AUT_cum12 / log_AUT_cum12, 1)
)

sitadel <- read_excel ("data/ROES_201702.xls", sheet = "AUT_REG",
                      col_types = c ("text","text","numeric","numeric","numeric","numeric"),
                      mutate (annee = str_sub (date, 1, 4),
                              i_AUT = ip_AUT + ig_AUT) %>%
                      group_by (REG, annee) %>%
                      summarise(
                        log_AUT_cum = sum (log_AUT),
                        i_AUT_cum = sum (i_AUT)) %>%
                      ungroup () %>%
                      group_by (REG) %>%
                      mutate (i_AUT_cum_lag = lag (i_AUT_cum, 1), # décalage de 1 année
                              i_AUT_cum_delta = i_AUT_cum - i_AUT_cum_lag,
                              i_AUT_cum_evo = round (100 * i_AUT_cum_delta / i_AUT_cum_lag, 1),# taux d'évo

                              i_AUT_cum_part = round (100 * i_AUT_cum / log_AUT_cum, 1) # part de l'individu
                      )
```

9.3.2 Pesticides

Sur les données “FormationPreparationDesDonnées.RData”, table “exercice” :

- calculer le taux de quantification pour chaque molécule (code_parametre), chacune des années : nombre de fois où elle a été retrouvée (code_remarque=1) sur le nombre de fois où elle a été cherchée (code_remarque = 1,2,7 ou 10)
 - créer la variable "annee"
 - créer la variable de comptage des présences pour chaque analyse (1=présent, 0=absent)
 - créer la variable de comptage des recherches pour chaque analyse (1=recherchée, 0=non recherchée)
 - pour chaque combinaison année x code_parametre, calculer le taux de quantification
- trouver pour chaque station, sur l'année 2016, le prélèvement pour lequel la concentration cumulée, toutes substances confondues, est la plus élevée (~ le prélèvement le plus pollué)
 - filtrer les concentrations quantifiées (code_remarque=1) et l'année 2016
 - sommer les concentrations (resultat_analyse) par combinaison code_station x code_prelevement
 - ne conserver que le prélèvement avec le concentration maximale

```
rm (list = ls ())
load (file = "data/FormationPreparationDesDonnées.RData")
taux_de_quantification <- exercice %>%
  mutate (year = year (date_prelevement),
           num = 1 * (code_remarque == 1),
           denom = 1 * (code_remarque %in% c (1,2,7,10))) %>%
  group_by (year, code_parametre) %>%
  summarise (taux_de_quantification = 100 * sum (num) / sum (denom))

datatable (taux_de_quantification)
```

Show entries

Search:

	year	code_parametre	taux_de_quantification
1	1991	1107	100
2	1991	1129	0
3	1991	1130	0
4	1991	1136	0
5	1991	1176	0
6	1991	1199	0
7	1991	1203	100
8	1991	1208	50
9	1991	1209	100
10	1991	1212	0

Showing 1 to 10 of 6,760 entries

Previous

1

2

3

4

5

...

676

Next

```
pire_echantillon_par_station_en_2016 <- exercice %>%
  filter (code_remarque == 1, year (date_prelevement) == 2016) %>%
  group_by (libelle_station, code_prelevement) %>%
    summarise (concentration_cumulee = sum (resultat_analyse)) %>%
  group_by (libelle_station) %>%
    filter (concentration_cumulee == max (concentration_cumulee)) %>%
  ungroup ()

datatable (pire_echantillon_par_station_en_2016)
```

Show 10 ▾ entries

Search:

	libelle_station	code_prelevement	concentration_cumulee
1	ANGLE GUIGNARD-RETENUE	43003	0.04
2	ANXURE SAINT-GERMAIN-D'ANXURE	42230	0.381
3	APREMONT-RETENUE	42892	0.074
4	ARAIZE CHATELAIS	41450	0.044
5	ARON MOULAY	41357	0.1
6	AUBANCE LOUERRE	41567	0.099
7	AUBANCE MURS-ERIGNE	41540	0.448
8	AUBANCE SAINT-SATURNIN-SUR-LOIRE	41573	0.579
9	AUTHION LES PONTS-DE-CE	42532	0.27
10	AUTISE SAINT-HILAIRE-DES-LOGES	41998	0.048

Showing 1 to 10 of 191 entries

Previous

1

2

3

4

5

...

20

Next

9.4 Exercice 5.1

- reconstituer le dataframe “exercice” à partir des données contenues dans les tables “analyse”, “prelevement” et “station” (jointures)
- calculer le nombre d’analyses réalisées sur des molécules (code_parametre) présentes dans le référentiel
- produire une liste des **code_parametre** associés à des analyses mais absents du référentiel
- produire une table des analyses “orphelines”, c’est-à-dire qui ne correspondent pas à un prélèvement

```

rm (list = ls ())

load (file = "data/FormationPreparationDesDonnées.RData")

recalcul_exercice <- analyse %>%
  inner_join (prelevement) %>%
  inner_join (station) %>%
  mutate (date_creation = as.character (date_creation),
          annee = year (date_prelevement))

nb_analyses_presentes_dans_referentiel <- analyse %>%
  inner_join (parametre) %>%
  summarise (n = count (.)) %>%
  pull (n)

nb_analyses_presentes_dans_referentiel2 <- analyse %>%
  inner_join (parametre) %>%
  nrow ()

codes_modecules_absents_du_referentiel <- analyse %>%
  anti_join (parametre) %>%
  group_by (code_parametre) %>%
  tally ()

analyses_avec_code_prelevement_non_retrouve_dans_table_prelevement <- analyse %>%
  anti_join (prelevement)

analyse_avec_code_prelevement_non_retrouve_dans_table_prelevement2 <- analyse %>%
  filter(!(code_prelevement %in% unique(prelevement$code_prelevement)))

```

9.5 Exercice 7

Calculer à partir des tables fournies dans le fichier *majic.RData* issues des [fichiers fonciers](#) un indicateur d'étalement urbain entre 2009 et 2014 à la commune et à l'epci sur la région Pays de la Loire.

```
rm (list = ls ())
library(ggplot2)
load("data/majic.RData")
#pour chaque millésime de majic, on remet les données sur la nouvelle carte des territoires
```

```
majic_2009 <- bind_rows (majic_2009_com44, majic_2009_com49, majic_2009_com53, majic_2009_com54)
  left_join (com2017, by = c ("idcom" = "depcom")) %>%
  select (-idcom, -idcomtxt) %>%
  group_by (epci_2017, depcom2017) %>%
  summarise_all (funs (sum)) %>%
  ungroup %>%
  mutate (artif_2009=dcnt07+dcnt09+dcnt10+dcnt11+dcnt12+dcnt13) %>%
  select(-starts_with("dcnt"))
```

```
majic_2014 <- bind_rows (majic_2014_com44, majic_2014_com49, majic_2014_com53, majic_2014_com54)
  left_join (com2017, by = c ("idcom" = "depcom")) %>%
  select (-idcom, -idcomtxt) %>%
  group_by (epci_2017, depcom2017) %>%
  summarise_all (funs (sum)) %>%
  ungroup %>%
  mutate (artif_2014=dcnt07+dcnt09+dcnt10+dcnt11+dcnt12+dcnt13) %>%
  select(-starts_with("dcnt"))
```

#on passe également les données de population sur la nouvelle carte des territoires

```
p_2009 <- population_2009 %>%
  left_join (com2017, by = c ("idcom" = "depcom")) %>%
  select (-idcom) %>%
  group_by (epci_2017, depcom2017) %>%
  summarise(population_2009=sum(Population)) %>%
  ungroup
p_2014 <-population_2014 %>%
  left_join (com2017, by = c ("idcom" = "depcom")) %>%
  select (-idcom) %>%
  group_by (epci_2017, depcom2017) %>%
  summarise(population_2014=sum(Population)) %>%
```


ungroup

#indicateur à la commune

on joint les 4 tables précédentes par commune et on calcul les indicateurs

```
etalement_urbain_commune <- majic_2009 %>%
  left_join(majic_2014) %>%
  left_join (p_2009) %>%
  left_join (p_2014) %>%
  mutate (evoarti = 100 * artif_2014 / artif_2009 - 100,
           evopop = 100 * population_2014 / population_2009 - 100,
           indicateur_etalement_simple=evoarti/evopop,
           indicateur_etalement_avance = case_when (
             evoarti < 0 & evopop >= 0 ~ "1",
             evoarti >= 0 & evopop >= 0 & (evoarti / evopop <= 1 | evopop==0) ~ "2a",
             evoarti < 0 & evopop < 0 & evoarti / evopop > 1 ~ "2b",
             evopop < 0 & evoarti / evopop >= 0 & evoarti / evopop <= 1 ~ "2c",
             evopop > 0 & evoarti > 0 & evoarti <= 4.9 & evoarti / evopop > 1 ~ "3",
             evopop > 0 & evoarti> 4.9 & evoarti / evopop > 1 & evoarti / evopop <= 2 ~ '
             evopop > 0 & evoarti> 4.9 & evoarti / evopop > 2 ~ "5",
             evopop < 0 & evoarti / evopop < 0 ~ "6"
           )
  )
```

#indicateur à l'epci

on joint les 4 tables précédentes par commune, on agregge les compteurs par epci et on

```
etalement_urbain_epci <- majic_2009 %>%
  left_join(majic_2014) %>%
  left_join (p_2009) %>%
  left_join (p_2014) %>%
  select(-depcom2017) %>%
  group_by(epci_2017) %>%
  summarise_all(funs(sum(.))) %>%
  mutate (evoarti = 100 * artif_2014 / artif_2009 - 100,
           evopop = 100 * population_2014 / population_2009 - 100,
           indicateur_etalement_simple=evoarti/evopop,
           indicateur_etalement_avance = case_when (
             evoarti < 0 & evopop >= 0 ~ "1",
```

```

evoarti >= 0 & evopop >= 0 & (evoarti / evopop <= 1 | evopop==0) ~ "2a",
evoarti < 0 & evopop < 0 & evoarti / evopop > 1 ~ "2b",
evopop < 0 & evoarti / evopop >= 0 & evoarti / evopop <= 1 ~ "2c",
evopop > 0 & evoarti > 0 & evoarti <= 4.9 & evoarti / evopop > 1 ~ "3",
evopop > 0 & evoarti > 4.9 & evoarti / evopop > 1 & evoarti / evopop <= 2 ~ '
evopop > 0 & evoarti > 4.9 & evoarti / evopop > 2 ~ "5",
evopop < 0 & evoarti / evopop < 0 ~ "6"
)
)
# Deux graphiques de visualisation de notre indicateur
ggplot(data=etalement_urbain_epci) +
  geom_point(aes(x=evoarti,y=evopop,color=indicateur_etalement_avance))+
  theme_minimal() +
  labs(title="Indicateur d'étalement urbain sur les epci de la région Pays de la Loire",

ggplot(data=etalement_urbain_commune) +
  geom_point(aes(x=evoarti,y=evopop,color=indicateur_etalement_avance),size=.5,alpha=.5)
  theme_minimal()+
  labs(title="Indicateur d'étalement urbain sur les communes de la région Pays de la Loi

```