

# Indicateurs de la Biodiversité

*Andrew MacDonald*

*2020-06-19*



# Contents

<b>1</b>	<b>Dependencies</b>	<b>5</b>
<b>2</b>	<b>Download data</b>	<b>7</b>
<b>3</b>	<b>Sites</b>	<b>9</b>



## Chapter 1

# Dependencies

```
library(rcoleo)  
library(tidyverse)  
library(lubridate)
```



## Chapter 2

# Download data

The only point here is to demonstrate the workflow for downloading data

```
# On retire les cellules (classe sf) depuis l'API
cells <- rcoleo::sf_cells()
obs <- rcoleo::get_obs()
sites_dl <- rcoleo::sf_sites()

# until there is a better way to obtain the data and parse the result (perhaps
# in the form of a convenience function in rcoleo) we do this:

obs_df <- obs[[1]] %>% map("body") %>% map_df(~ select(.x, -closed_at))

all_obs <- obs_df %>%
  select(cell_code, site_code, date_obs, type,
         taxa = obs_species.taxa_name,
         var = obs_species.variable,
         val = obs_species.value) %>%
  mutate(date_obs = lubridate::ymd(date_obs),
         # convert cover into pres/abs (right?)
         count = if_else(var == "recouvrement", 1, val))

# CELLULES: On compte le nombre d'observation/nombre espece par type, année et cellule
obs_cells <- all_obs %>%
  group_by(cell_code, date_obs, type) %>%
  summarise(n = sum(count)) %>%
  ungroup

sp_cells <- all_obs %>%
  select(cell_code, date_obs, type, taxa) %>%
  distinct() %>%
```

```
group_by(cell_code, date_obs, type) %>%
  summarise(n = n()) %>%
  ungroup

# CAMPAGNES
sites <- sites_dl %>%
  select(site_code, off_station_code_id,
         type_milieu = type, geometry, site_id = id)

# On prépare les jeux de données pour chacun des types de campagnes

all_obs_con <- all_obs %>%
  filter(taxa != "inconnu")

microfaunes <- subset(all_obs_con, type == "insectes_sol")
papillons <- subset(all_obs_con, type == "papilionidés")
odonates <- subset(all_obs_con, type == "odonates")
vegetation <- subset(all_obs_con, type == "végétation")
```



## Chapter 3

### Sites

Summary data about the different sites

```
obs_cells %>%  
  ggplot(aes(x = cell_code, y = n)) + geom_point() +  
  facet_wrap(~type, scales = "free_y")
```

## Warning: Removed 1 rows containing missing values (geom\_point).

