ACME camera script 9-2-2024

Marissa A. Dyck

2024-02-09

Contents

> IMPORTANT the first two chunks of this r markdown file **after** the r setup allow for plot zooming, but it also means that the html file must be opened in a browser to view the document properly. When it knits in RStudio the preview will appear empty but the html when opened in a browser will have all the info and you can click on each plot to Zoom in on it.

These chunks (only visible in RStudio) allow for plot zooming once knitted and opened in browser, can delete if you don't want in your R markdown doc

Before you begin

Notes

A few notes about this script.

If you are running this with the 2022-2023 data make sure you download the whole (OSM_2022-2023 GitHub repository)[https://github.com/ACMElabUvic/OSM_2022-2023] from the ACMElabUvic GitHub. This will ensure you have all the files, data, and proper folder structure you will need to run this code and associated analyses.

Also make sure you open RStudio through the R project (OSM_2022-2023.Rproj) this will automatically set your working directory to the correct place (wherever you saved the repository) and ensure you don't have to change the file paths for some of the data.

If you have question please email the most recent author, currently

Marissa A. Dyck
Postdoctoral research fellow
University of Victoria
School of Environmental Studies
Email: marissadyck17@gmail.com

Netdrive access

This script relies on the user having access to the ACME lab Netdrive (you can view the .html output of this file if you don't have access and just want to see what the script did).

Helpful instructions for connecting to and navigating the Netdrive can also be found here: https://docs.google.com/document/d/1Z72IrlIXO8MUHCoVztcMrMdL10R2tHrHThEgfow1Cu0/edit.

R and RStudio

Before starting you should ensure you have the latest version of R and RStudio downloaded. This code was generated under R version 4.2.3 and with RStudio version 2024.04.2+764.

You can download R and RStudio HERE

R markdown

This script is written in R markdown and thus uses a mix of coding markup languages and R. If you are planning to run this script with new data or make any modifications you will want to be familiar with some basics of R markdown.

Below is an R markdown cheatsheet to help you get started,
R markdown cheatsheet

Install packages

If you don't already have the following packages installed, use the code below to install them. *NOTE this will not run automatically as eval=FALSE is included in the chunk setup (i.e. I don't want it to run every time I run this code since I have the packages installed)

```
install.packages(tidyverse)
install.packages(withr)
```

Load libraries

Then load the packages to your library.

```
library('tidyverse') # data tidying, visualization, and much more; this will load all tidyverse package
library('withr') # used to temporarily set wd
```

Deployment data

Let's start by importing the deployment data so we know how many cameras there are, info on the site names and arrays that were used for 2022-2023. We will use this data to compare with the other data files (e.g. timelapse, covariates, etc.) to ensure those were entered correctly

Import deployment data

Let's import the deployment data file, there is also a deployment site data file which we don't need for this analysis but the code is included to upload it if needed in the future. Although we could read both of these files in as a list, they are different enough and I want to perform a few cleaning operations that are different for each during the data import step so it is easier to import them separately

```
# read in deployment data and camera data files individually

# deployment data
deploy <- read_csv('data/raw/OSM_Deployment_Data_2022.csv',
```

```
                  # specify how we want the columns read in
                  col_types = cols(Project.ID = col_factor(),
                                   Deployment.Location.ID = col_factor(),
                                   Camera.Deployment.Begin.Date. = col_date(
                                     format = "%d-%b-%y"),
                                   Camera.Deployment.End.Date = col_date(
                                     format = "%d-%b-%y"),
                                   .default = col_character())) %>%
                  # the date columns could be read in as such if we needed but I don't think we use th

  # set the column names to lower case and replace the '.' with '_' (these are both personal preference
  set_names(
    names(.) %>%
      tolower() %>%

      # replace the '.' with '_'
      str_replace_all(pattern = '\\.', # provide the character pattern to look for (if you don't keep t
                      replacement = '_'))  # what you want the pattern to be replaced with


# # deployment site data
# cameras <- read_csv('data/raw/OSM_2022_Deployment_Site_Data.csv',
#
#                     # specify how we want the columns read in
#                     col_types = cols('Deploy Date' = col_date(
#                       format = "%d-%b-%y"),
#                       'Deploy Time' = col_time(),
#                       Lat = col_number(),
#                       Long = col_number(),
#                       Grade = col_integer(),
#                       Elevation = col_integer(),
#                       'Distance to trail (m)' = col_number(),
#                       'Distance to Lure' = col_number(),
#                       'Comments and Access Notes' = col_character(),
#                       .default = col_factor()
#                     )) %>%
#
#   # set the column names to lower case and replace the spaces with '_' (these are both personal prefe
#   set_names(
#     names(.) %>%
#       tolower()%>%
#       str_replace_all(pattern = ' ',
#                       replacement = '_'))
#
# # not sure if we need this data, may remove this section later if not needed.
```

Data checks

This section will likely need to be altered and amended each year as various year-specific issues with the data arise, but the functions in this section offer a good starting point to take a look at the data and ensure things imported correctly.

Let's take a look at this data and make sure everything is okay.

We'll start with the deployment data (deploy). We want to check that the data imported properly (each column is the type it should be) and that all the sites are accounted for and look correct. The deployment data files are what we will base our data checking for other files on because they should be the first ones that were generated and thus the most accurate (we hope).

Structure

Make sure columns imported correctly

```
# make sure the columns read in properly

str(deploy)
```

```
## spc_tbl_ [155 x 6] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ project_id                   : Factor w/ 4 levels "OSM_LU13","OSM_LU15",..: 1 1 1 1 1 1 1 1 1 1 .
##  $ deployment_location_id       : Factor w/ 155 levels "LU13_18","LU13_15",..: 1 2 3 4 5 6 7 8 9 10
##  $ camera_deployment_begin_date_: Date[1:155], format: "2022-09-18" "2022-09-18" ...
##  $ camera_deployment_end_date   : Date[1:155], format: "2023-09-27" "2023-09-27" ...
##  $ deployment_id                : chr [1:155] "D1" "D1" "D1" "D1" ...
##  $ camera_failure_details       : chr [1:155] NA NA NA NA ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   .default = col_character(),
##   ..   Project.ID = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   Deployment.Location.ID = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   Camera.Deployment.Begin.Date. = col_date(format = "%d-%b-%y"),
##   ..   Camera.Deployment.End.Date = col_date(format = "%d-%b-%y"),
##   ..   Deployment.ID = col_character(),
##   ..   Camera.Failure.Details = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
# everything looks good
```

Arrays and sites

Let's check the levels for the landscape units (project_id) and the sites (deployment_location_id) to make sure they look correct, but this data file should be correct and the one we will base other files on

```
# let's check the levels for the landscape units (project_id) and the sites (deployment_location_id) to

levels(deploy$project_id)
```

```
## [1] "OSM_LU13" "OSM_LU15" "OSM_LU21" "OSM_LU01"
```

```
levels(deploy$deployment_location_id)
```

```
##   [1] "LU13_18"  "LU13_15"  "LU13_03"  "LU13_34"  "LU13_57"  "LU13_16"
##   [7] "LU13_21"  "LU13_37"  "LU13_55"  "LU13_47"  "LU13_51"  "LU13_27"
##  [13] "LU13_05"  "LU13_26"  "LU13_35"  "LU13_13"  "LU13_128" "LU13_131"
```

```
##  [19] "LU13_12"  "LU13_33"  "LU13_32"  "LU13_30"  "LU13_53"  "LU13_22"
##  [25] "LU13_19"  "LU13_45"  "LU13_08"  "LU13_43"  "LU13_41"  "LU13_59"
##  [31] "LU13_06"  "LU13_14"  "LU13_52"  "LU13_36"  "LU13_38"  "LU13_20"
##  [37] "LU13_17"  "LU13_56"  "LU13_49"  "LU13_70"  "LU13_11"  "LU15-44"
##  [43] "LU15_11"  "LU15_19"  "LU15_07"  "LU15_34"  "LU15_37"  "LU15_18"
##  [49] "LU15_43"  "LU15_10"  "LU15_58"  "LU15_08"  "LU15_12"  "LU15_17"
##  [55] "LU15_01"  "LU15_26"  "LU15_27"  "LU15_28"  "LU15_02"  "LU15_29"
##  [61] "LU15_30"  "LU15_31"  "LU15_32"  "LU15_20"  "LU15_21"  "LU15_23"
##  [67] "LU15_41"  "LU15_25"  "LU15_09"  "LU15_24"  "LU15_46"  "LU15_40"
##  [73] "LU15_16"  "LU15_04"  "LU15_36"  "LI15_03"  "LU15_15"  "LU15_14"
##  [79] "LU15_22"  "LU15_61"  "LU21_106" "LU21_10"  "LU21_16"  "LU21_107"
##  [85] "LU21_41"  "LU21_63"  "LU21_27"  "LU21_23"  "LU21_21"  "LU21_82"
##  [91] "LU21_871" "LU21_164" "LU21_36"  "LU21_14"  "LU21_32"  "LU21_126"
##  [97] "LU21_122" "LU21_56"  "LU21_57"  "LU21_09"  "LU21_98"  "LU21_119"
## [103] "LU21_52"  "LU21_59"  "LU21_68"  "LU21_109" "LU21_93"  "LU21_114"
## [109] "LU21_105" "LU21_100" "LU21_74"  "LU21_78"  "LU21_153" "LU21_06"
## [115] "LU21_116" "LU21_97"  "LU01_06"  "LU01_10"  "LU01_11"  "LU01_13"
## [121] "LU01_22"  "LU01_25"  "LU01_27"  "LU01_30"  "LU01_32"  "LU01_36"
## [127] "LU01_40"  "LU01_41"  "LU01_43"  "LU01_44"  "LU01_45"  "LU01_46"
## [133] "LU01_47"  "LU01_48"  "LU01_60"  "LU01_63"  "LU01_64"  "LU01_66"
## [139] "LU01_67"  "LU01_70"  "LU01_71"  "LU01_72"  "LU01_73"  "LU01_74"
## [145] "LU01_75"  "LU01_76"  "LU01_77"  "LU01_78"  "LU01_79"  "LU01_80"
## [151] "LU01_82"  "LU01_83"  "LU01_84"  "LU01_85"  "LU01_86"
```

Arrays look good, but there are two sites with a '-' instead of a '_' , we will fix this in next step

NAs

Let's check that there aren't any NAs, if everything worked w/ no issues and was entered correctly there shouldn't be any NAs in this file

```
# check that there aren't any NAs
summary(deploy)
```

```
##     project_id deployment_location_id camera_deployment_begin_date_
##  OSM_LU13:41   LU13_18:  1            Min.   :2022-06-13
##  OSM_LU15:39   LU13_15:  1            1st Qu.:2022-07-30
##  OSM_LU21:36   LU13_03:  1            Median :2022-09-13
##  OSM_LU01:39   LU13_34:  1            Mean   :2022-08-23
##                LU13_57:  1            3rd Qu.:2022-09-17
##                LU13_16:  1            Max.   :2022-09-20
##                (Other):149
##  camera_deployment_end_date deployment_id      camera_failure_details
##  Min.   :2022-09-22         Length:155         Length:155
##  1st Qu.:2023-09-10         Class :character   Class :character
##  Median :2023-09-14         Mode  :character   Mode  :character
##  Mean   :2023-09-03
##  3rd Qu.:2023-09-25
##  Max.   :2023-10-02
##
```

Data manipulation

This code will also likely need to be changed as year-specific issues arise but gives a starting point for reformatting the data to match some of the other data files and fixing the issues we found in the previous step.

I like to do as much of my data manipulation I can in one *dplyr* pipe (i.e. code chunk) to avoid extra coding and assigning intermediate objects to the environment that I don't need, but if this format doesn't make sense to you, each step can be done individually if you pull the code out of the pipeline and reference the data within each function. I do write each step individually and check that it's working correctly as I go.

In the pipe below we

1. first rename some of the columns to be shorter/match previous years' data files
2. Then we fix the names of the specific site entries that were different from the detection data/entered incorrectly
3. Finally, we select only the columns we need (dropping the deployment_id and camera_failure_details)

Then we double check that the site names match the other data frame and they do!

```
deploy_fixed <- deploy %>%

  # rename start and end date so they are shorter
  # rename project_id and deployment_location_id so they match previous years' columns
  rename(start_date = camera_deployment_begin_date_,
         end_date = camera_deployment_end_date,
         array = project_id,
         site = deployment_location_id) %>%

  # rename site entries and remove prefix OSM from array
  mutate(site = as.factor(case_when(site == 'LU15-44' ~ 'LU15_44',
                                    site == 'LI15_03' ~ 'LU15_03',
                                    TRUE  ~ site)),
         array = str_remove(array,
                            pattern = "OSM_")) %>%

  # remove columns we don't need
  select(!c(camera_failure_details,
            deployment_id))



# check data again
head(deploy_fixed)
```

```
## # A tibble: 6 x 4
##   array site    start_date end_date
##   <chr> <fct>   <date>     <date>
## 1 LU13  LU13_18 2022-09-18 2023-09-27
## 2 LU13  LU13_15 2022-09-18 2023-09-27
## 3 LU13  LU13_03 2022-09-18 2023-09-27
## 4 LU13  LU13_34 2022-09-18 2023-09-27
```

```
## 5 LU13  LU13_57 2022-09-18 2023-09-24
## 6 LU13  LU13_16 2022-09-18 2023-09-27
```

```
# check levels
levels(deploy_fixed$site)
```

```
##   [1] "LU01_06"  "LU01_10"  "LU01_11"  "LU01_13"  "LU01_22"  "LU01_25"
##   [7] "LU01_27"  "LU01_30"  "LU01_32"  "LU01_36"  "LU01_40"  "LU01_41"
##  [13] "LU01_43"  "LU01_44"  "LU01_45"  "LU01_46"  "LU01_47"  "LU01_48"
##  [19] "LU01_60"  "LU01_63"  "LU01_64"  "LU01_66"  "LU01_67"  "LU01_70"
##  [25] "LU01_71"  "LU01_72"  "LU01_73"  "LU01_74"  "LU01_75"  "LU01_76"
##  [31] "LU01_77"  "LU01_78"  "LU01_79"  "LU01_80"  "LU01_82"  "LU01_83"
##  [37] "LU01_84"  "LU01_85"  "LU01_86"  "LU13_03"  "LU13_05"  "LU13_06"
##  [43] "LU13_08"  "LU13_11"  "LU13_12"  "LU13_128" "LU13_13"  "LU13_131"
##  [49] "LU13_14"  "LU13_15"  "LU13_16"  "LU13_17"  "LU13_18"  "LU13_19"
##  [55] "LU13_20"  "LU13_21"  "LU13_22"  "LU13_26"  "LU13_27"  "LU13_30"
##  [61] "LU13_32"  "LU13_33"  "LU13_34"  "LU13_35"  "LU13_36"  "LU13_37"
##  [67] "LU13_38"  "LU13_41"  "LU13_43"  "LU13_45"  "LU13_47"  "LU13_49"
##  [73] "LU13_51"  "LU13_52"  "LU13_53"  "LU13_55"  "LU13_56"  "LU13_57"
##  [79] "LU13_59"  "LU13_70"  "LU15_01"  "LU15_02"  "LU15_03"  "LU15_04"
##  [85] "LU15_07"  "LU15_08"  "LU15_09"  "LU15_10"  "LU15_11"  "LU15_12"
##  [91] "LU15_14"  "LU15_15"  "LU15_16"  "LU15_17"  "LU15_18"  "LU15_19"
##  [97] "LU15_20"  "LU15_21"  "LU15_22"  "LU15_23"  "LU15_24"  "LU15_25"
## [103] "LU15_26"  "LU15_27"  "LU15_28"  "LU15_29"  "LU15_30"  "LU15_31"
## [109] "LU15_32"  "LU15_34"  "LU15_36"  "LU15_37"  "LU15_40"  "LU15_41"
## [115] "LU15_43"  "LU15_44"  "LU15_46"  "LU15_58"  "LU15_61"  "LU21_06"
## [121] "LU21_09"  "LU21_10"  "LU21_100" "LU21_105" "LU21_106" "LU21_107"
## [127] "LU21_109" "LU21_114" "LU21_116" "LU21_119" "LU21_122" "LU21_126"
## [133] "LU21_14"  "LU21_153" "LU21_16"  "LU21_164" "LU21_21"  "LU21_23"
## [139] "LU21_27"  "LU21_32"  "LU21_36"  "LU21_41"  "LU21_52"  "LU21_56"
## [145] "LU21_57"  "LU21_59"  "LU21_63"  "LU21_68"  "LU21_74"  "LU21_78"
## [151] "LU21_82"  "LU21_871" "LU21_93"  "LU21_97"  "LU21_98"
```

Finish with deployment data

Save data

Now that this data is cleaned up we should save it to the data/processed folder in case we need the cleaned version later so we don't have to repeat the above steps.

Make sure when naming files we follow the best data managements practices for the ACME lab outlined here.

```
# save to processed data folder

write_csv(deploy_fixed,
          'data/processed/OSM_deployment_2022.csv')
```

Remove messy deployment data

Now that we've fixed the deployment data we should remove the old file from the environment so we don't accidentally use it

```
rm(deploy)
```

Camera operability

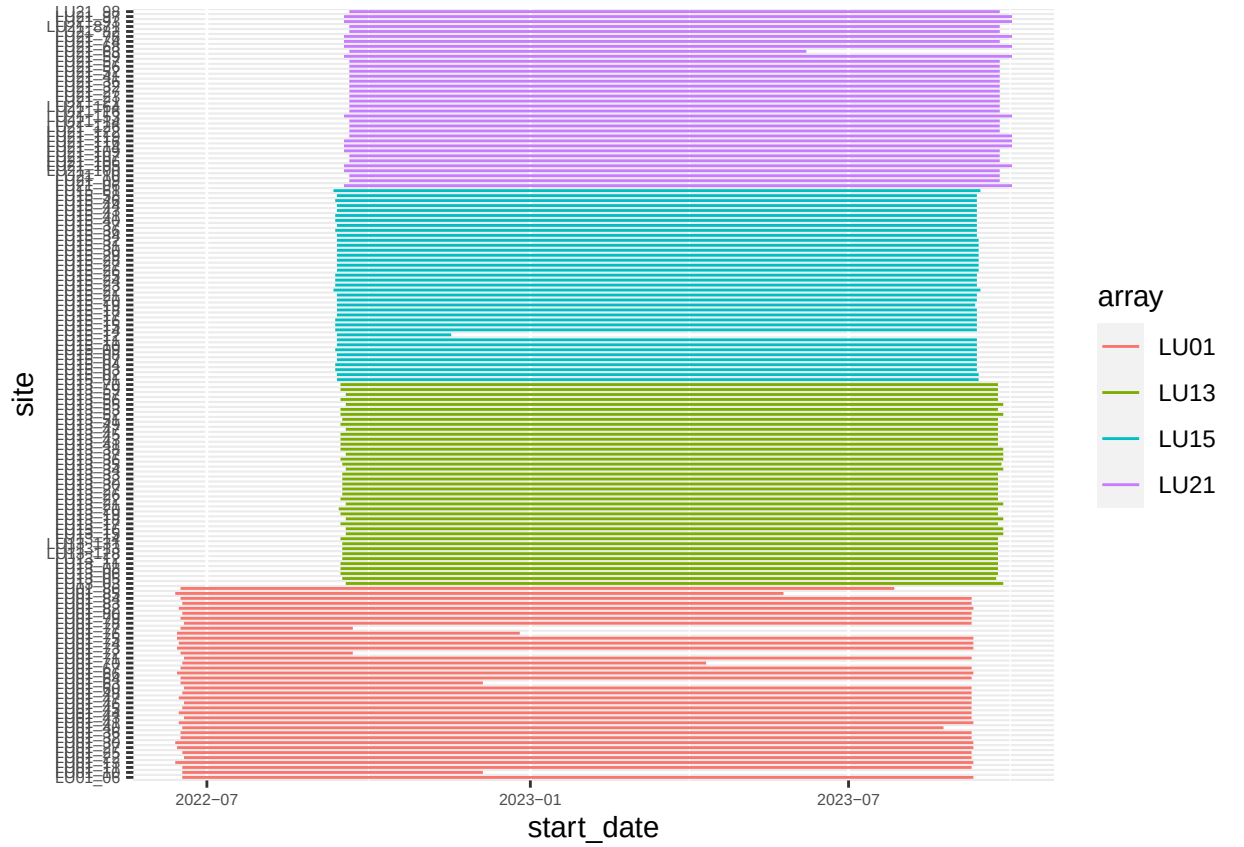
We can check the length each camera was operating using the cleaned deployment data, this is important for calculating the proportional presence/absences for analysis later on so we need to make sure nothing looks inaccurate here.

Let's plot the camera operability with `ggplot()` to look at this

```
# if starting from this point read in data
deploy_fixed <- read_csv('data/processed/OSM_deployment_2022.csv') %>%

  # make sure site re-reads in as a factor to compare with other data sets
  mutate(site = as.factor(site))
```

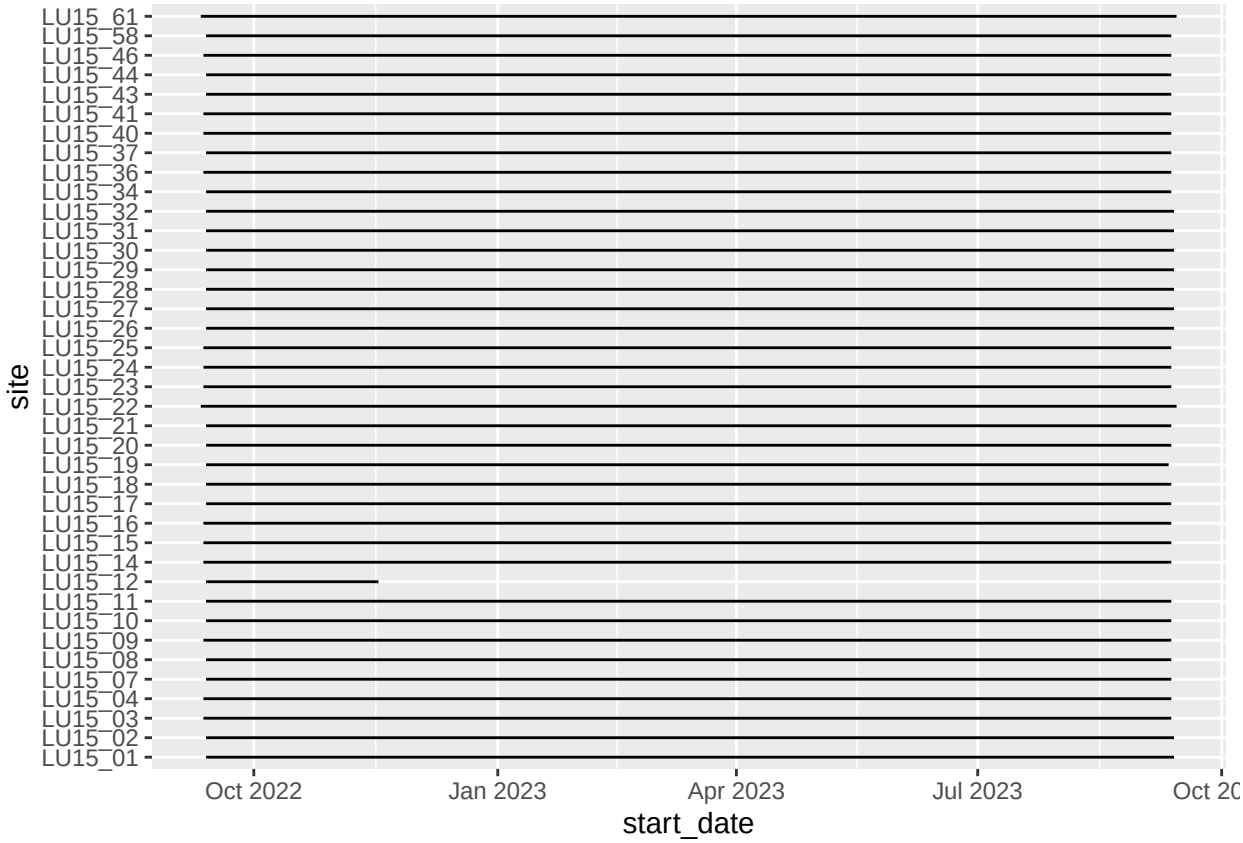
```
## Rows: 155 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): array, site
## date (2): start_date, end_date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# create graph of camera operability

ggplot(deploy_fixed, aes(color = array))+

  geom_segment(aes(x = start_date,
                   xend = end_date,
                   y = site,
                   yend = site)) +

  theme(axis.text = element_text(size = 6))
```



Something weird is happening with the end of one of the LU15 sites it has a later end date than the others by a lot. (This may not show up once the correction has been made to the raw data file). Let's plot just LU15 so we can see which site it is, it is probably a typo in the deployment data we will need to fix in the raw file.

```
deploy_fixed %>%

  filter(array == 'LU15') %>%

 ggplot(., ) +

  geom_segment(aes(x = start_date,
                   xend = end_date,
                   y = site,
                   yend = site))
```



the site we need to check in the raw data is LU15_12

Timelapse data

If you opened the OSM_2022-2023.Rproj file you should have your working directory set to the GitHub repository that you downloaded to your hard drive.

We need to temporarily set your working drive to the ACME lab Netdrive to import the Timelapse data files (there are too many to efficiently store on GitHub or GoogleDrive each year). We will use the `with_dir()` function in the *withr* package to do this.

Import timelapse data

Option 1

This code will import all of the timelapse data files and merge them into one data frame.

Make a list of all the data files and use `map_drf()` to read them in and join them to one data frame (map is a function in the *purrr* package that performs iterations and map_dfr returns data frames by row-binding objects together).

```
# temporarily set the working directory to import from the NetDrive
with_dir(new = '/Volumes/acmelab/1.Resources/2.Arrays/Alberta/OSM/2022-2023',
```

```
        # make a list of all .csv files in the 2. Timelapse Files folder
        OSM_2022_data <- list.files(
          path = '2. Timelapse Files', # provide the folder/s within the working drive where the files
          pattern = '*\\.csv*', # provide the extension in case there are other files saved in that fo
          full.names = TRUE) %>%

          map_dfr(~.x %>%
                    read_csv(.,

                             # specify how to read in the various columns, if you don't specify this R
                             col_types = cols(RootFolder = col_character(),
                                              File = col_character(),
                                              RelativePath = col_character(),
                                              Dark = col_logical(),
                                              DeleteFlag = col_logical(),
                                              Site = col_factor(),
                                              Classifier = col_factor(),
                                              Snow = col_factor(),
                                              Species = col_factor(),
                                              Event = col_character(),
                                              Empty = col_logical(),
                                              CoatColour = col_character(),
                                              CameraMalfunction = col_factor(),
                                              OtherSpecify = col_character(),
                                              Comments = col_character(),
                                              Noteworthy = col_logical(),
                                              DateTime = col_datetime(format = "%Y-%m-%d %H:%M:%S"),
                                              .default = col_integer()))) %>%  #.default sets any unsp

          # set the column names to lowercase, this makes it easier to avoid case-sensitive mistakes w
          set_names(
            names(.) %>%
              tolower()))

# finished importing data, this code will return warnings related to 'parsing issues'. Don't panic it i
```

This code may return warnings related to 'parsing issues'. Don't panic, it is fine.

Option 2

This code is probably not that useful but I spent a bit of time figuring out how to make it work before I fixed the code for option 1 so I have kept it here in case it is of use to someone in the future, but have commented it out so it doesn't run every time.

This will read in all the data files as separate data frames in a list and name them based on the file names (e.g. the landscape units for 2022-2023 data).

I've commented out this code so it doesn't run every time, you can select all the code and hit command + c to uncomment everything

```
# option two: read files in as a list and keep them separated by landscape unit (LU). This is useful if
```

```
# temporarily set the working directory to import from the NetDrive

# with_dir(new = '/Volumes/acmelab/1.Resources/2.Arrays/Alberta/OSM/2022-2023',
#
#          # make a list of all .csv files in the 2. Timelapse Files folder
#          OSM_2022_data_files <- list.files(
#            path = '2. Timelapse Files', # provide the folder/s within the working drive where the fil
#            pattern = '*\\.csv*', # provide the extension in case there are other files saved in that
#            full.names = TRUE) %>%
#
#            # set the names to the base name of each file w/o the .csv file extension and read in all
#            {setNames(map(., read_csv), sub("\\.csv$", "", basename(.)))})
```

Summaries

Some summaries we may need for reports etc.

```
# overall data summary
str(OSM_2022_data)
```

```
## spc_tbl_ [316,636 x 39] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ rootfolder       : chr [1:316636] "Deployment 1" "Deployment 1" "Deployment 1" "Deployment 1" ...
##  $ file             : chr [1:316636] "IMG_0001.JPG" "IMG_0002.JPG" "IMG_0003.JPG" "IMG_0004.JPG" ...
##  $ relativepath     : chr [1:316636] NA NA NA NA ...
##  $ deleteflag       : logi [1:316636] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ site             : Factor w/ 155 levels "LU01-06","LU01-10",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ classifier       : Factor w/ 4 levels "ERA","SM","SLM",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ snow             : Factor w/ 5 levels "0","100","Deep",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ species          : Factor w/ 39 levels "Staff","White-tailed deer",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ total            : int [1:316636] 2 2 2 2 2 2 2 2 2 2 ...
##  $ male             : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ female           : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ unknownsex       : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ adult            : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ yly              : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ yoy              : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ unknownage       : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ group_count      : int [1:316636] 2 2 2 2 2 2 2 2 2 2 ...
##  $ g_male           : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ g_female         : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ g_unknownsex     : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ g_adult          : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ g_yly            : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ g_yoy            : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ gunknownage      : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ event            : chr [1:316636] "Start" NA NA NA ...
##  $ empty            : logi [1:316636] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ coatcolour       : chr [1:316636] NA NA NA NA ...
##  $ leftantler       : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ rightantler      : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ lcount           : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ rcount           : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
```

```
##  $ cameramalfunction: Factor w/ 8 levels "Camera dead",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ otherspecify     : chr [1:316636] NA NA NA NA ...
##  $ comments         : chr [1:316636] NA NA NA NA ...
##  $ noteworthy       : logi [1:316636] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ datetime         : POSIXct[1:316636], format: "2022-06-17 10:01:52" "2022-06-17 10:01:58" ...
##  $ ...37            : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ dark             : logi [1:316636] NA NA NA NA NA NA ...
##  $ ...38            : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   .default = col_integer(),
##   ..   RootFolder = col_character(),
##   ..   File = col_character(),
##   ..   RelativePath = col_character(),
##   ..   DeleteFlag = col_logical(),
##   ..   Site = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   Classifier = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   Snow = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   Species = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   Total = col_integer(),
##   ..   Male = col_integer(),
##   ..   Female = col_integer(),
##   ..   UnknownSex = col_integer(),
##   ..   Adult = col_integer(),
##   ..   YLY = col_integer(),
##   ..   YOY = col_integer(),
##   ..   UnknownAge = col_integer(),
##   ..   Group_count = col_integer(),
##   ..   G_Male = col_integer(),
##   ..   G_Female = col_integer(),
##   ..   G_UnknownSex = col_integer(),
##   ..   G_Adult = col_integer(),
##   ..   G_YLY = col_integer(),
##   ..   G_YOY = col_integer(),
##   ..   GUnknownAge = col_integer(),
##   ..   Event = col_character(),
##   ..   Empty = col_logical(),
##   ..   CoatColour = col_character(),
##   ..   LeftAntler = col_integer(),
##   ..   RightAntler = col_integer(),
##   ..   LCount = col_integer(),
##   ..   RCount = col_integer(),
##   ..   CameraMalfunction = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   OtherSpecify = col_character(),
##   ..   Comments = col_character(),
##   ..   Noteworthy = col_logical(),
##   ..   DateTime = col_datetime(format = "%Y-%m-%d %H:%M:%S"),
##   ..   ...37 = col_integer()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
summary(OSM_2022_data)
```

```
##   rootfolder            file           relativepath       deleteflag
```

```
##  Length:316636      Length:316636      Length:316636      Mode :logical
##  Class :character   Class :character   Class :character   FALSE:316636
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##        site         classifier       snow                      species
##  LU15-27 : 30111   ERA :144553   0   :228134   White-tailed deer: 39081
##  LU21-59 : 15043   SM  :164260   100 : 51773   Black bear       : 36341
##  LU15-29 :  9180   SLM :  2480   Deep:  2519   Moose            : 26920
##  LU21-105:  7564   MB  :  5342   >50 : 23106   Staff            : 18340
##  LU15-40 :  6229   NA's:     1   <50 : 10768   Coyote           :  8583
##  (Other) :248508                 NA's:   336   (Other)          : 24976
##  NA's    :     1                               NA's             :162395
##      total             male             female          unknownsex
##  Min.   :0.00     Min.   :0        Min.   :0.00     Min.   :0.00
##  1st Qu.:1.00     1st Qu.:1        1st Qu.:1.00     1st Qu.:1.00
##  Median :1.00     Median :1        Median :1.00     Median :1.00
##  Mean   :1.25     Mean   :1        Mean   :0.91     Mean   :1.05
##  3rd Qu.:1.00     3rd Qu.:1        3rd Qu.:1.00     3rd Qu.:1.00
##  Max.   :8.00     Max.   :4        Max.   :4.00     Max.   :8.00
##  NA's   :162305   NA's   :288066   NA's   :275263   NA's   :249128
##      adult             yly              yoy            unknownage
##  Min.   :0.00     Min.   :0.00     Min.   :0.00     Min.   :0.00
##  1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00
##  Median :1.00     Median :1.00     Median :1.00     Median :1.00
##  Mean   :1.01     Mean   :1.12     Mean   :1.05     Mean   :0.93
##  3rd Qu.:1.00     3rd Qu.:1.00     3rd Qu.:1.00     3rd Qu.:1.00
##  Max.   :8.00     Max.   :2.00     Max.   :4.00     Max.   :3.00
##  NA's   :207285   NA's   :315762   NA's   :295699   NA's   :312583
##   group_count         g_male           g_female       g_unknownsex
##  Min.   :1.00     Min.   :1.00     Min.   :0.00     Min.   :0.00
##  1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00
##  Median :1.00     Median :1.00     Median :1.00     Median :1.00
##  Mean   :1.41     Mean   :1.09     Mean   :1.07     Mean   :1.27
##  3rd Qu.:2.00     3rd Qu.:1.00     3rd Qu.:1.00     3rd Qu.:1.00
##  Max.   :8.00     Max.   :4.00     Max.   :4.00     Max.   :8.00
##  NA's   :162370   NA's   :288005   NA's   :275236   NA's   :248985
##     g_adult           g_yly            g_yoy          gunknownage
##  Min.   :0.00     Min.   :1.00     Min.   :0.00     Min.   :0.00
##  1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00
##  Median :1.00     Median :1.00     Median :1.00     Median :1.00
##  Mean   :1.12     Mean   :1.31     Mean   :1.56     Mean   :1.09
##  3rd Qu.:1.00     3rd Qu.:2.00     3rd Qu.:2.00     3rd Qu.:1.00
##  Max.   :8.00     Max.   :2.00     Max.   :4.00     Max.   :5.00
##  NA's   :207297   NA's   :315714   NA's   :295723   NA's   :312485
##     event             empty          coatcolour          leftantler
##  Length:316636      Mode :logical   Length:316636      Min.   : 1.00
##  Class :character   FALSE:146852    Class :character   1st Qu.: 2.00
##  Mode  :character   TRUE :169784    Mode  :character   Median : 4.00
##                                                        Mean   : 3.83
##                                                        3rd Qu.: 5.00
##                                                        Max.   :14.00
```

```
##                                                       NA's   :304688
##   rightantler         lcount           rcount
##  Min.   : 1.0     Min.   : NA      Min.   : NA
##  1st Qu.: 2.0     1st Qu.: NA      1st Qu.: NA
##  Median : 4.0     Median : NA      Median : NA
##  Mean   : 3.8     Mean   :NaN      Mean   :NaN
##  3rd Qu.: 5.0     3rd Qu.: NA      3rd Qu.: NA
##  Max.   :11.0     Max.   : NA      Max.   : NA
##  NA's   :304754   NA's   :316636   NA's   :316636
##            cameramalfunction  otherspecify         comments
##  Partially obscured :  4770   Length:316636      Length:316636
##  Repositioned       :  2436   Class :character   Class :character
##  Fully obscured     :  2163   Mode  :character   Mode  :character
##  OtherSpecify       :   421
##  Trigger malfunction:    95
##  (Other)            :     5
##  NA's               :306746
##  noteworthy         datetime                          ...37
##  Mode :logical   Min.   :1979-12-31 23:00:00.00   Min.   : NA
##  FALSE:316551    1st Qu.:2022-11-03 03:06:16.50   1st Qu.: NA
##  TRUE :85        Median :2023-04-14 12:00:00.00   Median : NA
##                  Mean   :2023-03-19 22:34:15.61   Mean   :NaN
##                  3rd Qu.:2023-07-09 13:16:36.00   3rd Qu.: NA
##                  Max.   :2023-10-02 11:10:20.00   Max.   : NA
##                                                   NA's   :316636
##     dark             ...38
##  Mode :logical   Min.   : NA
##  FALSE:209368    1st Qu.: NA
##  NA's :107268    Median : NA
##                  Mean   :NaN
##                  3rd Qu.: NA
##                  Max.   : NA
##                  NA's   :316636
```

```
# mammal specific
levels(OSM_2022_data$species)
```

```
##  [1] "Staff"               "White-tailed deer"   "Black bear"
##  [4] "Snowshoe hare"       "Moose"               "Coyote"
##  [7] "Unknown mustelid"    "Unknown"             "Other birds"
## [10] "Fisher"              "Unknown deer"        "Red squirrel"
## [13] "Marten"              "Striped skunk"       "Raven"
## [16] "Unknown canid"       "Unknown ungulate"    "Grey wolf"
## [19] "Red fox"             "Cougar"              "Lynx"
## [22] "Ruffed grouse"       "Short-tailed weasel" "Human"
## [25] "Domestic dog"        "Porcupine"           "Spruce grouse"
## [28] "Otter"               "Grey jay"            "Owl"
## [31] "Other"               "Beaver"              "ATVer"
## [34] "Wolverine"           "Caribou"             "Long-tailed weasel"
## [37] "Hunter"              "Snowmobiler"         "Canada goose"
```

```
# make vector of mammals
mammal_species <- c('Black bear',
```

```
                    'Caribou',
                    'Coyote',
                    'Fisher',
                    'Grey wolf',
                    'Lynx',
                    'Moose',
                    'Red fox',
                    'White-tailed deer',
                    'Snowshoe hare',
                    'Red squirrel',
                    'Marten',
                    'Striped skunk',
                    'Cougar',
                    'Porcupine',
                    'Short-tailed weasel',
                    'Otter',
                    'Beaver',
                    'Wolverine',
                    'Long-tailed weasel')

# how many photos of just mammals
all_mammals <- OSM_2022_data %>%

  filter(species %in% mammal_species)



# focal species
focal_species <- c('Black bear',
                   'Caribou',
                   'Coyote',
                   'Fisher',
                   'Grey wolf',
                   'Lynx',
                   'Moose',
                   'Red fox',
                   'White-tailed deer')

# detections of focal species
focal_mammals <- OSM_2022_data %>%

  filter(species %in% focal_species)
```

Data checks

This section will also very likely need to be altered and amended each year as various year-specific issues with the data arise, but the functions in this section offer a good starting point to take a look at the data and ensure things imported correctly.

Data structure

Check the internal structure of the data using the `str()` function.

This should all be good since we specified how to read in each variable above, but if new columns are added from the Timelapse program/process, that could change things each year so we should double check anyways.

```
# check the internal structure
str(OSM_2022_data)
```

```
## spc_tbl_ [316,636 x 39] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ rootfolder       : chr [1:316636] "Deployment 1" "Deployment 1" "Deployment 1" "Deployment 1" ...
##  $ file             : chr [1:316636] "IMG_0001.JPG" "IMG_0002.JPG" "IMG_0003.JPG" "IMG_0004.JPG" ...
##  $ relativepath     : chr [1:316636] NA NA NA NA ...
##  $ deleteflag       : logi [1:316636] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ site             : Factor w/ 155 levels "LU01-06","LU01-10",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ classifier       : Factor w/ 4 levels "ERA","SM","SLM",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ snow             : Factor w/ 5 levels "0","100","Deep",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ species          : Factor w/ 39 levels "Staff","White-tailed deer",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ total            : int [1:316636] 2 2 2 2 2 2 2 2 2 2 ...
##  $ male             : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ female           : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ unknownsex       : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ adult            : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ yly              : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ yoy              : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ unknownage       : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ group_count      : int [1:316636] 2 2 2 2 2 2 2 2 2 2 ...
##  $ g_male           : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ g_female         : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ g_unknownsex     : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ g_adult          : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ g_yly            : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ g_yoy            : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ gunknownage      : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ event            : chr [1:316636] "Start" NA NA NA ...
##  $ empty            : logi [1:316636] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ coatcolour       : chr [1:316636] NA NA NA NA ...
##  $ leftantler       : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ rightantler      : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ lcount           : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ rcount           : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ cameramalfunction: Factor w/ 8 levels "Camera dead",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ otherspecify     : chr [1:316636] NA NA NA NA ...
##  $ comments         : chr [1:316636] NA NA NA NA ...
##  $ noteworthy       : logi [1:316636] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ datetime         : POSIXct[1:316636], format: "2022-06-17 10:01:52" "2022-06-17 10:01:58" ...
##  $ ...37            : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  $ dark             : logi [1:316636] NA NA NA NA NA NA ...
##  $ ...38            : int [1:316636] NA NA NA NA NA NA NA NA NA NA ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   .default = col_integer(),
##   ..   RootFolder = col_character(),
##   ..   File = col_character(),
##   ..   RelativePath = col_character(),
##   ..   DeleteFlag = col_logical(),
##   ..   Site = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
```

```
##   ..   Classifier = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   Snow = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   Species = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   Total = col_integer(),
##   ..   Male = col_integer(),
##   ..   Female = col_integer(),
##   ..   UnknownSex = col_integer(),
##   ..   Adult = col_integer(),
##   ..   YLY = col_integer(),
##   ..   YOY = col_integer(),
##   ..   UnknownAge = col_integer(),
##   ..   Group_count = col_integer(),
##   ..   G_Male = col_integer(),
##   ..   G_Female = col_integer(),
##   ..   G_UnknownSex = col_integer(),
##   ..   G_Adult = col_integer(),
##   ..   G_YLY = col_integer(),
##   ..   G_YOY = col_integer(),
##   ..   GUnknownAge = col_integer(),
##   ..   Event = col_character(),
##   ..   Empty = col_logical(),
##   ..   CoatColour = col_character(),
##   ..   LeftAntler = col_integer(),
##   ..   RightAntler = col_integer(),
##   ..   LCount = col_integer(),
##   ..   RCount = col_integer(),
##   ..   CameraMalfunction = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   OtherSpecify = col_character(),
##   ..   Comments = col_character(),
##   ..   Noteworthy = col_logical(),
##   ..   DateTime = col_datetime(format = "%Y-%m-%d %H:%M:%S"),
##   ..   ...37 = col_integer()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

The variables all look like the uploaded with the correct format

Data summary

This is a good way to look at all the data and notice if there are any glaring issues/outliers. etc. for all of the columns

```
summary(OSM_2022_data)
```

```
##   rootfolder            file           relativepath       deleteflag
##  Length:316636      Length:316636      Length:316636      Mode :logical
##  Class :character   Class :character   Class :character   FALSE:316636
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##        site        classifier      snow                     species
```

```
##  LU15-27 : 30111   ERA :144553   0   :228134   White-tailed deer: 39081
##  LU21-59 : 15043   SM  :164260   100 : 51773   Black bear       : 36341
##  LU15-29 :  9180   SLM :  2480   Deep:  2519   Moose            : 26920
##  LU21-105:  7564   MB  :  5342   >50 : 23106   Staff            : 18340
##  LU15-40 :  6229   NA's:     1   <50 : 10768   Coyote           :  8583
##  (Other) :248508                 NA's:   336   (Other)          : 24976
##  NA's    :     1                               NA's             :162395
##      total             male            female         unknownsex
##  Min.   :0.00     Min.   :0        Min.   :0.00     Min.   :0.00
##  1st Qu.:1.00     1st Qu.:1        1st Qu.:1.00     1st Qu.:1.00
##  Median :1.00     Median :1        Median :1.00     Median :1.00
##  Mean   :1.25     Mean   :1        Mean   :0.91     Mean   :1.05
##  3rd Qu.:1.00     3rd Qu.:1        3rd Qu.:1.00     3rd Qu.:1.00
##  Max.   :8.00     Max.   :4        Max.   :4.00     Max.   :8.00
##  NA's   :162305   NA's   :288066   NA's   :275263   NA's   :249128
##      adult             yly              yoy           unknownage
##  Min.   :0.00     Min.   :0.00     Min.   :0.00     Min.   :0.00
##  1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00
##  Median :1.00     Median :1.00     Median :1.00     Median :1.00
##  Mean   :1.01     Mean   :1.12     Mean   :1.05     Mean   :0.93
##  3rd Qu.:1.00     3rd Qu.:1.00     3rd Qu.:1.00     3rd Qu.:1.00
##  Max.   :8.00     Max.   :2.00     Max.   :4.00     Max.   :3.00
##  NA's   :207285   NA's   :315762   NA's   :295699   NA's   :312583
##   group_count         g_male          g_female       g_unknownsex
##  Min.   :1.00     Min.   :1.00     Min.   :0.00     Min.   :0.00
##  1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00
##  Median :1.00     Median :1.00     Median :1.00     Median :1.00
##  Mean   :1.41     Mean   :1.09     Mean   :1.07     Mean   :1.27
##  3rd Qu.:2.00     3rd Qu.:1.00     3rd Qu.:1.00     3rd Qu.:1.00
##  Max.   :8.00     Max.   :4.00     Max.   :4.00     Max.   :8.00
##  NA's   :162370   NA's   :288005   NA's   :275236   NA's   :248985
##     g_adult           g_yly            g_yoy         gunknownage
##  Min.   :0.00     Min.   :1.00     Min.   :0.00     Min.   :0.00
##  1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00
##  Median :1.00     Median :1.00     Median :1.00     Median :1.00
##  Mean   :1.12     Mean   :1.31     Mean   :1.56     Mean   :1.09
##  3rd Qu.:1.00     3rd Qu.:2.00     3rd Qu.:2.00     3rd Qu.:1.00
##  Max.   :8.00     Max.   :2.00     Max.   :4.00     Max.   :5.00
##  NA's   :207297   NA's   :315714   NA's   :295723   NA's   :312485
##     event             empty          coatcolour          leftantler
##  Length:316636      Mode :logical   Length:316636      Min.   : 1.00
##  Class :character   FALSE:146852    Class :character   1st Qu.: 2.00
##  Mode  :character   TRUE :169784    Mode  :character   Median : 4.00
##                                                        Mean   : 3.83
##                                                        3rd Qu.: 5.00
##                                                        Max.   :14.00
##                                                        NA's   :304688
##   rightantler         lcount           rcount
##  Min.   : 1.0     Min.   : NA      Min.   : NA
##  1st Qu.: 2.0     1st Qu.: NA      1st Qu.: NA
##  Median : 4.0     Median : NA      Median : NA
##  Mean   : 3.8     Mean   :NaN      Mean   :NaN
##  3rd Qu.: 5.0     3rd Qu.: NA      3rd Qu.: NA
##  Max.   :11.0     Max.   : NA      Max.   : NA
```

```
##  NA's   :304754   NA's   :316636   NA's   :316636
##            cameramalfunction  otherspecify         comments
##  Partially obscured :  4770   Length:316636      Length:316636
##  Repositioned       :  2436   Class :character   Class :character
##  Fully obscured     :  2163   Mode  :character   Mode  :character
##  OtherSpecify       :   421
##  Trigger malfunction:    95
##  (Other)            :     5
##  NA's               :306746
##  noteworthy         datetime                          ...37
##  Mode :logical   Min.   :1979-12-31 23:00:00.00   Min.   : NA
##  FALSE:316551    1st Qu.:2022-11-03 03:06:16.50   1st Qu.: NA
##  TRUE :85        Median :2023-04-14 12:00:00.00   Median : NA
##                  Mean   :2023-03-19 22:34:15.61   Mean   :NaN
##                  3rd Qu.:2023-07-09 13:16:36.00   3rd Qu.: NA
##                  Max.   :2023-10-02 11:10:20.00   Max.   : NA
##                                                   NA's   :316636
##     dark             ...38
##  Mode :logical   Min.   : NA
##  FALSE:209368    1st Qu.: NA
##  NA's :107268    Median : NA
##                  Mean   :NaN
##                  3rd Qu.: NA
##                  Max.   : NA
##                  NA's   :316636
```

From this if you notice any issues explore them more closely in the code chunks below

Column names

We can use the `names()` function to check that all the column names are correct and match with other year's of OSM data

```
names(OSM_2022_data)
```

```
##  [1] "rootfolder"        "file"              "relativepath"
##  [4] "deleteflag"        "site"              "classifier"
##  [7] "snow"              "species"           "total"
## [10] "male"              "female"            "unknownsex"
## [13] "adult"             "yly"               "yoy"
## [16] "unknownage"        "group_count"       "g_male"
## [19] "g_female"          "g_unknownsex"      "g_adult"
## [22] "g_yly"             "g_yoy"             "gunknownage"
## [25] "event"             "empty"             "coatcolour"
## [28] "leftantler"        "rightantler"       "lcount"
## [31] "rcount"            "cameramalfunction" "otherspecify"
## [34] "comments"          "noteworthy"        "datetime"
## [37] "...37"             "dark"              "...38"
```

Because there are slight differences in the number/order of columns between files, R creates a couple extra columns (...37 & ...38) which we will delete later.

We also need to add a month and year column for future data processing steps, we can extract these columns from the datetime column in the data formatting below

Lastly we need an array column which we can extract from the site column

Dates

The minimum date in the summary data showed 1979 which is obviously incorrect. Let's see how many entries have a wrong date by filtering for years prior to 2022

```
OSM_2022_data %>%

  filter(year(datetime) <2022 )
```

```
## # A tibble: 2 x 39
##   rootfolder  file  relativepath deleteflag site  classifier snow  species total
##   <chr>       <chr> <chr>        <lgl>      <fct> <fct>      <fct> <fct>   <int>
## 1 Deployment~ IMG_~ <NA>         FALSE      LU01~ SM         0     Staff       2
## 2 Deployment~ IMG_~ <NA>         FALSE      LU01~ SM         0     Staff       2
## # i 30 more variables: male <int>, female <int>, unknownsex <int>, adult <int>,
## #   yly <int>, yoy <int>, unknownage <int>, group_count <int>, g_male <int>,
## #   g_female <int>, g_unknownsex <int>, g_adult <int>, g_yly <int>,
## #   g_yoy <int>, gunknownage <int>, event <chr>, empty <lgl>, coatcolour <chr>,
## #   leftantler <int>, rightantler <int>, lcount <int>, rcount <int>,
## #   cameramalfunction <fct>, otherspecify <chr>, comments <chr>,
## #   noteworthy <lgl>, datetime <dttm>, ...37 <int>, dark <lgl>, ...38 <int>
```

Luckily it appears that only 2 images have a year before 2022 and they are both staff images so we can probably safely remove these and it won't mess with any analyses.

Sites

For 2022-2023 data there should be 155 (there were 156 sites originally but LU01-31 had no data for some reason so we are left with 155). It looks like we have them all based on the number of levels printed with `str()`, but let's make sure there isn't anything wonky with any of the sites or site names.

```
# check that all the sites are accounted for
# for 2022-2023 data there should be 155 (there were 56 sites originally but LU01-31 had no data)
levels(OSM_2022_data$site)
```

```
##   [1] "LU01-06"  "LU01-10"  "LU01-11"  "LU01-13"  "LU01-22"  "LU01-25"
##   [7] "LU01-27"  "LU01-30"  "LU01-32"  "LU01-36"  "LU01-40"  "LU01-41"
##  [13] "LU01-43"  "LU01-44"  "LU01-45"  "LU01-46"  "LU01-47"  "LU01-48"
##  [19] "LU01-60"  "LU01-63"  "LU01-64"  "LU01-66"  "LU01-67"  "LU01-70"
##  [25] "LU 01-71" "LU01-72"  "LU01-73"  "LU01-74"  "LU01-75"  "LU01-76"
##  [31] "LU01-77"  "LU01-78"  "LU01-79"  "LU01-80"  "LU01-82"  "LU01-83"
##  [37] "LU01-84"  "LU01-85"  "LU01-86"  "LU13-03"  "LU13-05"  "LU13-06"
##  [43] "LU13-08"  "LU13-11"  "LU13-12"  "LU13-128" "LU13-13"  "LU13-131"
##  [49] "LU13-14"  "LU13-15"  "LU13-16"  "LU13-17"  "LU13-18"  "LU13-19"
##  [55] "LU13-20"  "LU13-21"  "LU13-22"  "LU13-26"  "LU13-27"  "LU13-30"
##  [61] "LU13-32"  "LU13-33"  "LU13-34"  "LU13-35"  "LU13-36"  "LU13-37"
##  [67] "LU13-38"  "LU13-41"  "LU13-43"  "LU13-45"  "LU13-47"  "LU13-49"
##  [73] "LU13-51"  "LU13-52"  "LU13-53"  "LU13-55"  "LU13-56"  "LU13-57"
##  [79] "LU13-59"  "LU13-70"  "LU15-01"  "LU15-02"  "LU15-03"  "LU15-04"
```

```
##  [85] "LU15-07"  "LU15-08"  "LU15-09"  "LU15-10"  "LU15-11"  "LU15-12"
##  [91] "LU15-14"  "LU15-15"  "LU15-16"  "LU15-17"  "LU15-18"  "LU15-19"
##  [97] "LU15-20"  "LU15-21"  "LU15-22"  "LU15-23"  "LU15-24"  "LU15-25"
## [103] "LU15-26"  "LU15-27"  "LU15-28"  "LU15-29"  "LU15-30"  "LU15-31"
## [109] "LU15-32"  "LU15-34"  "LU15-36"  "LU15-37"  "LU15-40"  "LU15-41"
## [115] "LU15-43"  "LU15-44"  "LU15-46"  "LU15-58"  "LU15-61"  "LU21-06"
## [121] "LU21-09"  "LU21-10"  "LU21-100" "LU21-105" "LU21-106" "LU21-107"
## [127] "LU21-109" "LU21-114" "LU21-116" "LU21-119" "LU21-122" "LU21-126"
## [133] "LU21-14"  "LU21-153" "LU21-16"  "LU21-164" "LU21-21"  "LU21-23"
## [139] "LU21-27"  "LU21-32"  "LU21-36"  "LU21-41"  "LU21-52"  "LU21-56"
## [145] "LU21-57"  "LU21-59"  "LU21-63"  "LU21-68"  "LU21-74"  "LU21-78"
## [151] "LU21-82"  "LU21-871" "LU21-93"  "LU21-97"  "LU21-98"
```

```
# need to fix entry LU 01-71 to LU01-71 (has unnecessary space)
```

Looks like there is one site that was entered with an unnecessary space (LU 01-71), we can convert this to match the format of the others (LU01-71) in the data manipulation section.

Otherwise there is the correct number, let's compare with the deployment data we loaded earlier to see if the site names match up.

Since we set site as a factor when we imported the depoyment data and the covariate data we can check which sites were imported with the `levels()` function, we can compare this with the deployment data found we imported earlier using the `setdiff()`. *If you are getting a NULL output for either of the `setdiff()` lines of code, then you need to make sure that site is actually a factor in both the data sets.* Sometimes data manipulation steps change how the variable is reading in R so you made need to fix this.

```
# check which sites are in timelapse data that are not in deployment
setdiff(levels(OSM_2022_data$site),
        levels(deploy_fixed$site))
```

```
##   [1] "LU01-06"  "LU01-10"  "LU01-11"  "LU01-13"  "LU01-22"  "LU01-25"
##   [7] "LU01-27"  "LU01-30"  "LU01-32"  "LU01-36"  "LU01-40"  "LU01-41"
##  [13] "LU01-43"  "LU01-44"  "LU01-45"  "LU01-46"  "LU01-47"  "LU01-48"
##  [19] "LU01-60"  "LU01-63"  "LU01-64"  "LU01-66"  "LU01-67"  "LU01-70"
##  [25] "LU 01-71" "LU01-72"  "LU01-73"  "LU01-74"  "LU01-75"  "LU01-76"
##  [31] "LU01-77"  "LU01-78"  "LU01-79"  "LU01-80"  "LU01-82"  "LU01-83"
##  [37] "LU01-84"  "LU01-85"  "LU01-86"  "LU13-03"  "LU13-05"  "LU13-06"
##  [43] "LU13-08"  "LU13-11"  "LU13-12"  "LU13-128" "LU13-13"  "LU13-131"
##  [49] "LU13-14"  "LU13-15"  "LU13-16"  "LU13-17"  "LU13-18"  "LU13-19"
##  [55] "LU13-20"  "LU13-21"  "LU13-22"  "LU13-26"  "LU13-27"  "LU13-30"
##  [61] "LU13-32"  "LU13-33"  "LU13-34"  "LU13-35"  "LU13-36"  "LU13-37"
##  [67] "LU13-38"  "LU13-41"  "LU13-43"  "LU13-45"  "LU13-47"  "LU13-49"
##  [73] "LU13-51"  "LU13-52"  "LU13-53"  "LU13-55"  "LU13-56"  "LU13-57"
##  [79] "LU13-59"  "LU13-70"  "LU15-01"  "LU15-02"  "LU15-03"  "LU15-04"
##  [85] "LU15-07"  "LU15-08"  "LU15-09"  "LU15-10"  "LU15-11"  "LU15-12"
##  [91] "LU15-14"  "LU15-15"  "LU15-16"  "LU15-17"  "LU15-18"  "LU15-19"
##  [97] "LU15-20"  "LU15-21"  "LU15-22"  "LU15-23"  "LU15-24"  "LU15-25"
## [103] "LU15-26"  "LU15-27"  "LU15-28"  "LU15-29"  "LU15-30"  "LU15-31"
## [109] "LU15-32"  "LU15-34"  "LU15-36"  "LU15-37"  "LU15-40"  "LU15-41"
## [115] "LU15-43"  "LU15-44"  "LU15-46"  "LU15-58"  "LU15-61"  "LU21-06"
## [121] "LU21-09"  "LU21-10"  "LU21-100" "LU21-105" "LU21-106" "LU21-107"
## [127] "LU21-109" "LU21-114" "LU21-116" "LU21-119" "LU21-122" "LU21-126"
## [133] "LU21-14"  "LU21-153" "LU21-16"  "LU21-164" "LU21-21"  "LU21-23"
```

```
## [139] "LU21-27"  "LU21-32"  "LU21-36"  "LU21-41"  "LU21-52"  "LU21-56"
## [145] "LU21-57"  "LU21-59"  "LU21-63"  "LU21-68"  "LU21-74"  "LU21-78"
## [151] "LU21-82"  "LU21-871" "LU21-93"  "LU21-97"  "LU21-98"
```

```
# and switch the order to check if there are extras in deployment data compared to timelapse
setdiff(levels(deploy_fixed$site),
        levels(OSM_2022_data$site))
```

```
##   [1] "LU01_06"  "LU01_10"  "LU01_11"  "LU01_13"  "LU01_22"  "LU01_25"
##   [7] "LU01_27"  "LU01_30"  "LU01_32"  "LU01_36"  "LU01_40"  "LU01_41"
##  [13] "LU01_43"  "LU01_44"  "LU01_45"  "LU01_46"  "LU01_47"  "LU01_48"
##  [19] "LU01_60"  "LU01_63"  "LU01_64"  "LU01_66"  "LU01_67"  "LU01_70"
##  [25] "LU01_71"  "LU01_72"  "LU01_73"  "LU01_74"  "LU01_75"  "LU01_76"
##  [31] "LU01_77"  "LU01_78"  "LU01_79"  "LU01_80"  "LU01_82"  "LU01_83"
##  [37] "LU01_84"  "LU01_85"  "LU01_86"  "LU13_03"  "LU13_05"  "LU13_06"
##  [43] "LU13_08"  "LU13_11"  "LU13_12"  "LU13_128" "LU13_13"  "LU13_131"
##  [49] "LU13_14"  "LU13_15"  "LU13_16"  "LU13_17"  "LU13_18"  "LU13_19"
##  [55] "LU13_20"  "LU13_21"  "LU13_22"  "LU13_26"  "LU13_27"  "LU13_30"
##  [61] "LU13_32"  "LU13_33"  "LU13_34"  "LU13_35"  "LU13_36"  "LU13_37"
##  [67] "LU13_38"  "LU13_41"  "LU13_43"  "LU13_45"  "LU13_47"  "LU13_49"
##  [73] "LU13_51"  "LU13_52"  "LU13_53"  "LU13_55"  "LU13_56"  "LU13_57"
##  [79] "LU13_59"  "LU13_70"  "LU15_01"  "LU15_02"  "LU15_03"  "LU15_04"
##  [85] "LU15_07"  "LU15_08"  "LU15_09"  "LU15_10"  "LU15_11"  "LU15_12"
##  [91] "LU15_14"  "LU15_15"  "LU15_16"  "LU15_17"  "LU15_18"  "LU15_19"
##  [97] "LU15_20"  "LU15_21"  "LU15_22"  "LU15_23"  "LU15_24"  "LU15_25"
## [103] "LU15_26"  "LU15_27"  "LU15_28"  "LU15_29"  "LU15_30"  "LU15_31"
## [109] "LU15_32"  "LU15_34"  "LU15_36"  "LU15_37"  "LU15_40"  "LU15_41"
## [115] "LU15_43"  "LU15_44"  "LU15_46"  "LU15_58"  "LU15_61"  "LU21_06"
## [121] "LU21_09"  "LU21_10"  "LU21_100" "LU21_105" "LU21_106" "LU21_107"
## [127] "LU21_109" "LU21_114" "LU21_116" "LU21_119" "LU21_122" "LU21_126"
## [133] "LU21_14"  "LU21_153" "LU21_16"  "LU21_164" "LU21_21"  "LU21_23"
## [139] "LU21_27"  "LU21_32"  "LU21_36"  "LU21_41"  "LU21_52"  "LU21_56"
## [145] "LU21_57"  "LU21_59"  "LU21_63"  "LU21_68"  "LU21_74"  "LU21_78"
## [151] "LU21_82"  "LU21_871" "LU21_93"  "LU21_97"  "LU21_98"
```

It looks like all the sites are different because the timelapse data is formatted with a dash '-' instead of an underscore '_' between the LU and the site name. We will fix this in the data manipulation and then recheck.

Species names

Let's check that all the species names were entered correctly and we don't have any duplicates with different spelling or something, which is common with wildlife data.

Since we set species as a factor when we imported the data we can also use the the `levels()` function to see all the species names.

```
# check that all the species names were entered correctly
levels(OSM_2022_data$species)
```

```
##  [1] "Staff"               "White-tailed deer"   "Black bear"
##  [4] "Snowshoe hare"       "Moose"               "Coyote"
```

```
##  [7] "Unknown mustelid"    "Unknown"             "Other birds"
## [10] "Fisher"              "Unknown deer"        "Red squirrel"
## [13] "Marten"              "Striped skunk"       "Raven"
## [16] "Unknown canid"       "Unknown ungulate"    "Grey wolf"
## [19] "Red fox"             "Cougar"              "Lynx"
## [22] "Ruffed grouse"       "Short-tailed weasel" "Human"
## [25] "Domestic dog"        "Porcupine"           "Spruce grouse"
## [28] "Otter"               "Grey jay"            "Owl"
## [31] "Other"               "Beaver"              "ATVer"
## [34] "Wolverine"           "Caribou"             "Long-tailed weasel"
## [37] "Hunter"              "Snowmobiler"         "Canada goose"
```

```
# no glaring issues with species entries
```

No glaring issues here.

Check for NAs

There are a lot of NAs in this data set, and most of them are fine but we should check that there aren't NAs for some of the more critical information like the site and datetime columns. We can use the `summary()` function to get a printout of all the variables.

This is also a great way to check for any other glaring issues such as miscounted groups (really large or really small max/min numbers) etc.

```
# check for NAs in columns that shouldn't have NAs, looking at the summary is also a good way to check
summary(OSM_2022_data)
```

```
##   rootfolder            file           relativepath       deleteflag
##  Length:316636      Length:316636      Length:316636      Mode :logical
##  Class :character   Class :character   Class :character   FALSE:316636
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##        site        classifier      snow                      species
##  LU15-27 : 30111   ERA :144553   0   :228134   White-tailed deer: 39081
##  LU21-59 : 15043   SM  :164260   100 : 51773   Black bear       : 36341
##  LU15-29 :  9180   SLM :  2480   Deep:  2519   Moose            : 26920
##  LU21-105:  7564   MB  :  5342   >50 : 23106   Staff            : 18340
##  LU15-40 :  6229   NA's:     1   <50 : 10768   Coyote           :  8583
##  (Other) :248508                 NA's:   336   (Other)          : 24976
##  NA's    :     1                               NA's             :162395
##      total             male            female         unknownsex
##  Min.   :0.00     Min.   :0        Min.   :0.00     Min.   :0.00
##  1st Qu.:1.00     1st Qu.:1        1st Qu.:1.00     1st Qu.:1.00
##  Median :1.00     Median :1        Median :1.00     Median :1.00
##  Mean   :1.25     Mean   :1        Mean   :0.91     Mean   :1.05
##  3rd Qu.:1.00     3rd Qu.:1        3rd Qu.:1.00     3rd Qu.:1.00
##  Max.   :8.00     Max.   :4        Max.   :4.00     Max.   :8.00
##  NA's   :162305   NA's   :288066   NA's   :275263   NA's   :249128
##      adult             yly              yoy           unknownage
```

```
##  Min.   :0.00     Min.   :0.00     Min.   :0.00     Min.   :0.00
##  1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00
##  Median :1.00     Median :1.00     Median :1.00     Median :1.00
##  Mean   :1.01     Mean   :1.12     Mean   :1.05     Mean   :0.93
##  3rd Qu.:1.00     3rd Qu.:1.00     3rd Qu.:1.00     3rd Qu.:1.00
##  Max.   :8.00     Max.   :2.00     Max.   :4.00     Max.   :3.00
##  NA's   :207285   NA's   :315762   NA's   :295699   NA's   :312583
##   group_count         g_male          g_female       g_unknownsex
##  Min.   :1.00     Min.   :1.00     Min.   :0.00     Min.   :0.00
##  1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00
##  Median :1.00     Median :1.00     Median :1.00     Median :1.00
##  Mean   :1.41     Mean   :1.09     Mean   :1.07     Mean   :1.27
##  3rd Qu.:2.00     3rd Qu.:1.00     3rd Qu.:1.00     3rd Qu.:1.00
##  Max.   :8.00     Max.   :4.00     Max.   :4.00     Max.   :8.00
##  NA's   :162370   NA's   :288005   NA's   :275236   NA's   :248985
##     g_adult           g_yly            g_yoy         gunknownage
##  Min.   :0.00     Min.   :1.00     Min.   :0.00     Min.   :0.00
##  1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00     1st Qu.:1.00
##  Median :1.00     Median :1.00     Median :1.00     Median :1.00
##  Mean   :1.12     Mean   :1.31     Mean   :1.56     Mean   :1.09
##  3rd Qu.:1.00     3rd Qu.:2.00     3rd Qu.:2.00     3rd Qu.:1.00
##  Max.   :8.00     Max.   :2.00     Max.   :4.00     Max.   :5.00
##  NA's   :207297   NA's   :315714   NA's   :295723   NA's   :312485
##     event             empty          coatcolour          leftantler
##  Length:316636      Mode :logical   Length:316636      Min.   : 1.00
##  Class :character   FALSE:146852    Class :character   1st Qu.: 2.00
##  Mode  :character   TRUE :169784    Mode  :character   Median : 4.00
##                                                        Mean   : 3.83
##                                                        3rd Qu.: 5.00
##                                                        Max.   :14.00
##                                                        NA's   :304688
##   rightantler         lcount           rcount
##  Min.   : 1.0     Min.   : NA      Min.   : NA
##  1st Qu.: 2.0     1st Qu.: NA      1st Qu.: NA
##  Median : 4.0     Median : NA      Median : NA
##  Mean   : 3.8     Mean   :NaN      Mean   :NaN
##  3rd Qu.: 5.0     3rd Qu.: NA      3rd Qu.: NA
##  Max.   :11.0     Max.   : NA      Max.   : NA
##  NA's   :304754   NA's   :316636   NA's   :316636
##             cameramalfunction  otherspecify         comments
##  Partially obscured :  4770   Length:316636      Length:316636
##  Repositioned       :  2436   Class :character   Class :character
##  Fully obscured     :  2163   Mode  :character   Mode  :character
##  OtherSpecify       :   421
##  Trigger malfunction:    95
##  (Other)            :     5
##  NA's               :306746
##  noteworthy         datetime                         ...37
##  Mode :logical   Min.   :1979-12-31 23:00:00.00   Min.   : NA
##  FALSE:316551    1st Qu.:2022-11-03 03:06:16.50   1st Qu.: NA
##  TRUE :85        Median :2023-04-14 12:00:00.00   Median : NA
##                  Mean   :2023-03-19 22:34:15.61   Mean   :NaN
##                  3rd Qu.:2023-07-09 13:16:36.00   3rd Qu.: NA
##                  Max.   :2023-10-02 11:10:20.00   Max.   : NA
```

```
##                                                 NA's   :316636
##     dark             ...38
##  Mode :logical   Min.   : NA
##  FALSE:209368    1st Qu.: NA
##  NA's :107268    Median : NA
##                  Mean   :NaN
##                  3rd Qu.: NA
##                  Max.   : NA
##                  NA's   :316636
```

```
# there is 1 NA in the site column, let's check what other data is associated with this entry to make s
OSM_2022_data %>%

  filter(is.na(site))
```

```
## # A tibble: 1 x 39
##   rootfolder  file  relativepath deleteflag site  classifier snow  species total
##   <chr>       <chr> <chr>        <lgl>      <fct> <fct>      <fct> <fct>   <int>
## 1 Deployment~ IMG_~ <NA>         FALSE      <NA>  <NA>       <50   <NA>       NA
## # i 30 more variables: male <int>, female <int>, unknownsex <int>, adult <int>,
## #   yly <int>, yoy <int>, unknownage <int>, group_count <int>, g_male <int>,
## #   g_female <int>, g_unknownsex <int>, g_adult <int>, g_yly <int>,
## #   g_yoy <int>, gunknownage <int>, event <chr>, empty <lgl>, coatcolour <chr>,
## #   leftantler <int>, rightantler <int>, lcount <int>, rcount <int>,
## #   cameramalfunction <fct>, otherspecify <chr>, comments <chr>,
## #   noteworthy <lgl>, datetime <dttm>, ...37 <int>, dark <lgl>, ...38 <int>
```

```
# it is not an entry with an animal image so I wouldn't worry about fixing the site entry
```

It looks like there was one entry with an NA for the site column but it wasn't associated with an image of an animal so there's really no need to fix it or remove it at this point.

Data manipulation

The following code will fix any data issues we found in the data check steps. This code will need to be modified each year as well as year-specific and R version specific issues arise but this provides a good starting point.

As with the deployment data, I like to do as much of my data manipulation I can in one *dplyr* pipe (i.e. code chunk) to avoid extra coding and assigning intermediate objects to the environment that I don't need, but if this format doesn't make sense to you, each step can be done individually if you pull the code out of the pipeline and reference the data within each function. I do write each step individually and check that it's working correctly as I go.

In the pipe below we

1. remove the extra columns created from reading in the files at once
2. Fix the site that was entered with a space and fix all the sites so they have underscores instead of dashes
3. Create missing columns (year, month, and array) form existing data

And then we do the same data check steps as above to make sure everything worked.

```r
# Data manipulation timelapse data ------------------------------------------------

# can add code/remove code within the code chunk below to fix any issues that were found from the data

# first make sure to assign an object to the environment that will be your new fixed data. I usually st
OSM_2022_data_fixed <- OSM_2022_data %>%

  # removed extra columns (see R markdown for info on why these columns got added)
  select(!c(...37,
            dark,
            ...38)) %>%

  # filter out the two entries with years before 2022
  filter(year(datetime) >= 2022) %>%

  # fix issues with site column
  mutate(
    # fix site entry with unnecessary space
    site = recode(site,
                  # old entry followed by new entry
                  'LU 01-71' = 'LU01-71'),

    # change format of sites to include '_' instead of '-'
    site = str_replace(site,
                       pattern = '\\-',
                       replacement = '_'),

    # site needs to be a factor and for some reason the code above changes it to a character
    site = as.factor(site),

    # add month and year columns from the datetime data for merging with other files later
    month = month(datetime),
    year = year(datetime)) %>%

    # also split the site column (but keep original) into the LU and site
    separate_wider_delim(site,
                         delim = '_',
                         names = c('array',
                                   'camera'),
                         cols_remove = FALSE)

# use code below  to check that each step worked

# columns (removed extra columns w/ NAs)
names(OSM_2022_data_fixed)
```

```
##  [1] "rootfolder"        "file"              "relativepath"
##  [4] "deleteflag"        "array"             "camera"
##  [7] "site"              "classifier"        "snow"
## [10] "species"           "total"             "male"
## [13] "female"            "unknownsex"        "adult"
## [16] "yly"               "yoy"               "unknownage"
## [19] "group_count"       "g_male"            "g_female"
```

```
## [22] "g_unknownsex"      "g_adult"           "g_yly"
## [25] "g_yoy"             "gunknownage"       "event"
## [28] "empty"             "coatcolour"        "leftantler"
## [31] "rightantler"       "lcount"            "rcount"
## [34] "cameramalfunction" "otherspecify"      "comments"
## [37] "noteworthy"        "datetime"          "month"
## [40] "year"
```

```
# sites (fixed LU 01-71)
levels(OSM_2022_data_fixed$site)
```

```
##   [1] "LU01_06"  "LU01_10"  "LU01_11"  "LU01_13"  "LU01_22"  "LU01_25"
##   [7] "LU01_27"  "LU01_30"  "LU01_32"  "LU01_36"  "LU01_40"  "LU01_41"
##  [13] "LU01_43"  "LU01_44"  "LU01_45"  "LU01_46"  "LU01_47"  "LU01_48"
##  [19] "LU01_60"  "LU01_63"  "LU01_64"  "LU01_66"  "LU01_67"  "LU01_70"
##  [25] "LU01_71"  "LU01_72"  "LU01_73"  "LU01_74"  "LU01_75"  "LU01_76"
##  [31] "LU01_77"  "LU01_78"  "LU01_79"  "LU01_80"  "LU01_82"  "LU01_83"
##  [37] "LU01_84"  "LU01_85"  "LU01_86"  "LU13_03"  "LU13_05"  "LU13_06"
##  [43] "LU13_08"  "LU13_11"  "LU13_12"  "LU13_128" "LU13_13"  "LU13_131"
##  [49] "LU13_14"  "LU13_15"  "LU13_16"  "LU13_17"  "LU13_18"  "LU13_19"
##  [55] "LU13_20"  "LU13_21"  "LU13_22"  "LU13_26"  "LU13_27"  "LU13_30"
##  [61] "LU13_32"  "LU13_33"  "LU13_34"  "LU13_35"  "LU13_36"  "LU13_37"
##  [67] "LU13_38"  "LU13_41"  "LU13_43"  "LU13_45"  "LU13_47"  "LU13_49"
##  [73] "LU13_51"  "LU13_52"  "LU13_53"  "LU13_55"  "LU13_56"  "LU13_57"
##  [79] "LU13_59"  "LU13_70"  "LU15_01"  "LU15_02"  "LU15_03"  "LU15_04"
##  [85] "LU15_07"  "LU15_08"  "LU15_09"  "LU15_10"  "LU15_11"  "LU15_12"
##  [91] "LU15_14"  "LU15_15"  "LU15_16"  "LU15_17"  "LU15_18"  "LU15_19"
##  [97] "LU15_20"  "LU15_21"  "LU15_22"  "LU15_23"  "LU15_24"  "LU15_25"
## [103] "LU15_26"  "LU15_27"  "LU15_28"  "LU15_29"  "LU15_30"  "LU15_31"
## [109] "LU15_32"  "LU15_34"  "LU15_36"  "LU15_37"  "LU15_40"  "LU15_41"
## [115] "LU15_43"  "LU15_44"  "LU15_46"  "LU15_58"  "LU15_61"  "LU21_06"
## [121] "LU21_09"  "LU21_10"  "LU21_100" "LU21_105" "LU21_106" "LU21_107"
## [127] "LU21_109" "LU21_114" "LU21_116" "LU21_119" "LU21_122" "LU21_126"
## [133] "LU21_14"  "LU21_153" "LU21_16"  "LU21_164" "LU21_21"  "LU21_23"
## [139] "LU21_27"  "LU21_32"  "LU21_36"  "LU21_41"  "LU21_52"  "LU21_56"
## [145] "LU21_57"  "LU21_59"  "LU21_63"  "LU21_68"  "LU21_74"  "LU21_78"
## [151] "LU21_82"  "LU21_871" "LU21_93"  "LU21_97"  "LU21_98"
```

```
print(OSM_2022_data_fixed,
      n = 15)
```

```
## # A tibble: 316,634 x 40
##    rootfolder  file  relativepath deleteflag array camera site  classifier snow
##    <chr>       <chr> <chr>        <lgl>      <chr> <chr>  <fct> <fct>      <fct>
##  1 Deployment~ IMG_~ <NA>         FALSE      LU01  06     LU01~ ERA        0
##  2 Deployment~ IMG_~ <NA>         FALSE      LU01  06     LU01~ ERA        0
##  3 Deployment~ IMG_~ <NA>         FALSE      LU01  06     LU01~ ERA        0
##  4 Deployment~ IMG_~ <NA>         FALSE      LU01  06     LU01~ ERA        0
##  5 Deployment~ IMG_~ <NA>         FALSE      LU01  06     LU01~ ERA        0
##  6 Deployment~ IMG_~ <NA>         FALSE      LU01  06     LU01~ ERA        0
##  7 Deployment~ IMG_~ <NA>         FALSE      LU01  06     LU01~ ERA        0
##  8 Deployment~ IMG_~ <NA>         FALSE      LU01  06     LU01~ ERA        0
##  9 Deployment~ IMG_~ <NA>         FALSE      LU01  06     LU01~ ERA        0
```

```
## 10 Deployment~ IMG_~ <NA>         FALSE      LU01  06     LU01~ ERA        0
## 11 Deployment~ IMG_~ <NA>         FALSE      LU01  06     LU01~ ERA        0
## 12 Deployment~ IMG_~ <NA>         FALSE      LU01  06     LU01~ ERA        0
## 13 Deployment~ IMG_~ <NA>         FALSE      LU01  06     LU01~ ERA        0
## 14 Deployment~ IMG_~ <NA>         FALSE      LU01  06     LU01~ ERA        0
## 15 Deployment~ IMG_~ <NA>         FALSE      LU01  06     LU01~ ERA        0
## # i 316,619 more rows
## # i 31 more variables: species <fct>, total <int>, male <int>, female <int>,
## #   unknownsex <int>, adult <int>, yly <int>, yoy <int>, unknownage <int>,
## #   group_count <int>, g_male <int>, g_female <int>, g_unknownsex <int>,
## #   g_adult <int>, g_yly <int>, g_yoy <int>, gunknownage <int>, event <chr>,
## #   empty <lgl>, coatcolour <chr>, leftantler <int>, rightantler <int>,
## #   lcount <int>, rcount <int>, cameramalfunction <fct>, ...
```

We should also double check that the sites match up against the deployment data now that we've fixed the site naming

```
# check which sites are in timelapse data that are not in deployment
setdiff(levels(OSM_2022_data_fixed$site),
        levels(deploy_fixed$site))
```

```
## character(0)
```

```
# and switch the order to check if there are extras in deployment data compared to timelapse
setdiff(levels(deploy_fixed$site),
        levels(OSM_2022_data_fixed$site))
```

```
## character(0)
```

Now all the sites match between the two data files, yay!

Finish timelapse data

Save data

In case someone wants all of the raw timelapse data without the issues that we fixed in the data manipulation section we should save this to a folder so we don't have to run the code again.

```
# Save clean timelapse data ------------------------------------------------

# if someone needs this data later
write_csv(OSM_2022_data_fixed,
          'data/processed/OSM_timelapse_2022.csv')
```

Remove messy data

We should also remove the messy data from the environemnt so we don't accidentall use it

```
rm(OSM_2022_data)
```

Independent detections

For the models we want to run we need data for the independent detections of each species at each site. We've defined independent detections as those at least 30 minutes apart.

The data manipulation and loop below will do this for us.

```
# we will continue working with the OSM_2022_data but you could start here and import the data again if

# can select code chunk and use command + shift + c to uncomment or comment a large portion of code

# OSM_2022_data_fixed <- read_csv('data/processed/OSM_2022_timelapse.csv')
#
# # ignoring parsing issues warning, this is just referring to some columns it's expecting logical data
#
# # check internal structure, even though we specified everything above with the fixed data when we exp
# str(OSM_2022_data_fixed)
#
# # datatime read correctly but we will need to change site and species to factors again
#
# OSM_2022_data_fixed <- OSM_2022_data_fixed %>%
#
#   mutate(species = as.factor(species),
#          site = as.factor(site))


# prep the data for calculating independent detections
OSM_2022_det <- OSM_2022_data_fixed %>%

  # select only variables of interest
  select(array,
         site,
         species,
         datetime,
         month,
         year) %>%

  # remove rows with no species info
  drop_na(species) %>%

  # now we need to create a new variable called timediff
  # first make sure data are arrange in proper order
  arrange(site, species, datetime) %>% # this will NOT work if not in correct order (early-late date)

  # create groups for each species at each site
  group_by(species, site) %>%

  # create new variable timediff that will calculate the difference
  mutate(timediff = as.numeric(difftime(datetime,lag(datetime),
                                        units = "mins")))


# set the independent detection threshold to 30 minutes
mins <- 30
```

```
# loop that assigns group ID
# identifies when there are photos/rows that are more than 30 mins apart
# Attributes an event ID
OSM_2022_det$event_id <- 9999
seq <- as.numeric(paste0(nrow(OSM_2022_det),0))
seq <- round(seq,-(nchar(seq)))

for (i in 2:nrow(OSM_2022_det)) {
  OSM_2022_det$event_id[i-1]  <- paste0("E",format(seq, scientific = F))
  if(is.na(OSM_2022_det$timediff[i]) | abs(OSM_2022_det$timediff[i]) > (mins)){
    seq <- seq + 1
  }
}

if(OSM_2022_det$timediff[nrow(OSM_2022_det)] < (mins)|
   is.na(OSM_2022_det$timediff[nrow(OSM_2022_det)])){
  OSM_2022_det$event_id[nrow(OSM_2022_det)] <- OSM_2022_det$event_id[nrow(OSM_2022_det)-1]
} else{OSM_2022_det$event_id[nrow(OSM_2022_det)] <- paste0("E",format(seq+1, scientific = F))
}

# now create a new data frame with a single row for each event
OSM_2022_det_ind <- OSM_2022_det %>%
  group_by(event_id) %>%
  filter(row_number()==1)

OSM_2022_det_ind
```

```
## # A tibble: 14,063 x 8
## # Groups:   event_id [14,063]
##    array site    species       datetime            month  year timediff event_id
##    <chr> <fct>   <fct>         <dttm>              <dbl> <dbl>    <dbl> <chr>
##  1 LU01  LU01_06 Staff         2022-06-17 10:01:52     6  2022      NA  E0
##  2 LU01  LU01_06 Staff         2023-09-10 12:51:15     9  2023  648166. E1
##  3 LU01  LU01_06 White-tailed~ 2022-06-18 11:09:19     6  2022      NA  E2
##  4 LU01  LU01_06 White-tailed~ 2022-07-10 13:56:10     7  2022   31847. E3
##  5 LU01  LU01_06 White-tailed~ 2022-07-25 11:04:44     7  2022   21429. E4
##  6 LU01  LU01_06 White-tailed~ 2022-07-31 06:38:06     7  2022    8356. E5
##  7 LU01  LU01_06 White-tailed~ 2022-08-01 09:45:28     8  2022    1627. E6
##  8 LU01  LU01_06 White-tailed~ 2022-08-01 15:51:01     8  2022     364. E7
##  9 LU01  LU01_06 White-tailed~ 2022-08-05 06:49:48     8  2022    5218. E8
## 10 LU01  LU01_06 White-tailed~ 2022-08-26 08:36:07     8  2022   30345. E9
## # i 14,053 more rows
```

Save independent detections

Let's also save this data file for later

```
write_csv(OSM_2022_det_ind,
          'data/processed/OSM_ind_det_2022.csv')
```

Graphs

Indpendent detections for mammals

Now lets create a few quick figures to look at the detection data

```
# Data visualization independent detections-----------------------------------------------

# read in saved detection data if starting here
detections <- read_csv('data/processed/OSM_ind_det_2022.csv') %>%

  # change site, species and event_id to factor
  mutate_if(is.character,
            as.factor)
```

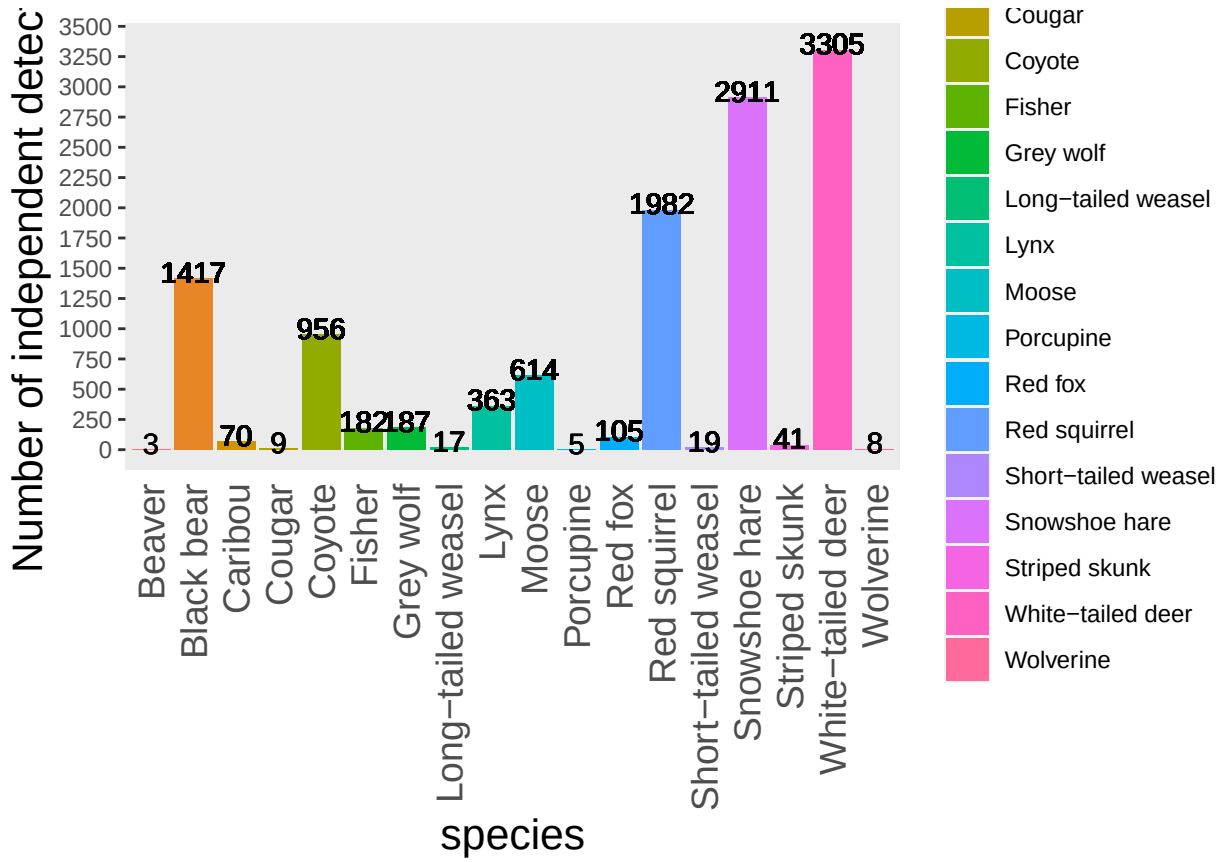
```
## Rows: 14063 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): array, site, species, event_id
## dbl  (3): month, year, timediff
## dttm (1): datetime
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# check number of different species
levels(OSM_2022_det_ind$species)
```

```
##  [1] "Staff"               "White-tailed deer"   "Black bear"
##  [4] "Snowshoe hare"       "Moose"               "Coyote"
##  [7] "Unknown mustelid"    "Unknown"             "Other birds"
## [10] "Fisher"              "Unknown deer"        "Red squirrel"
## [13] "Marten"              "Striped skunk"       "Raven"
## [16] "Unknown canid"       "Unknown ungulate"    "Grey wolf"
## [19] "Red fox"             "Cougar"              "Lynx"
## [22] "Ruffed grouse"       "Short-tailed weasel" "Human"
## [25] "Domestic dog"        "Porcupine"           "Spruce grouse"
## [28] "Otter"               "Grey jay"            "Owl"
## [31] "Other"               "Beaver"              "ATVer"
## [34] "Wolverine"           "Caribou"             "Long-tailed weasel"
## [37] "Hunter"              "Snowmobiler"         "Canada goose"
```

```
# create a vector of the list of mammals to use for quick data visualization/exploration. Could also cr
mammals <- c('White-tailed deer',
             'Black bear',
             'Snowshoe hare',
             'Moose',
             'Coyote',
             'Fisher',
             'Red squirrel',
             'Striped skunk',
             'Grey wolf',
```

```
              'Red fox',
              'Cougar',
              'Lynx',
              'Short-tailed weasel',
              'Porcupine',
              'Beaver',
              'Martin',
              'Wolverine',
              'Caribou',
              'Long-tailed weasel')

# remove NAs and select just images with mammals first then pipe new data into ggplot
det_graph <- detections %>%

  # remove less useful species
  filter(species %in% mammals) %>%

  # get the number of individual detections per species to add to graph
  group_by(species) %>%

  mutate(n = n()) %>%

  ungroup() %>%

  ggplot(.,
         aes(x = species)) +

  # create bar graph of the counts of each spp in the data
  geom_bar(aes(fill = species)) +

  # add the number of detections above each bar using the variable n we calculated earlier
  geom_text(aes(label = n,
                y = n + 50),
            size = 4) +

  # change y axis label
  labs(y = 'Number of independent detections') +

  # change breaks for y axis
  scale_y_continuous(breaks = seq(0,3500, by = 250)) +

  # change theme elements
  theme(axis.text.x = element_text(angle = 90,
                                   vjust = 0.5,
                                   hjust = 1,
                                   size = 14),
        axis.title = element_text(size = 16),
        axis.ticks.x = element_blank(),
        panel.grid = element_blank())

# print graph
det_graph
```



If we want to save the plot for easier viewing later we can use the code below

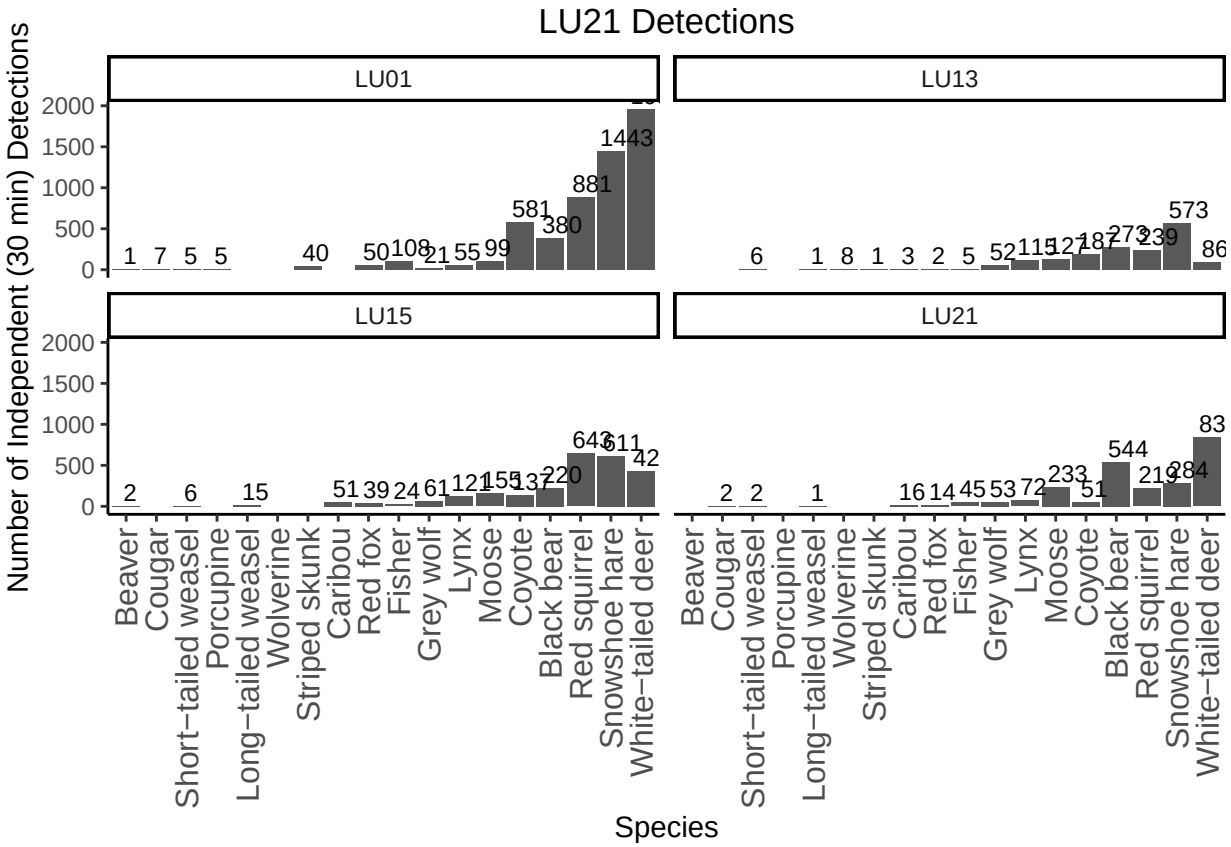
```
# save graph as jpeg (can also save as tiff, png, pdf by changing the file extension) but don't use .ti
ggsave('2022_indv_det_graph.jpeg',
       det_graph,
       path = 'outputs',
       width = 12,
       height = 10,
       units = 'in',
       dpi = 600)
```

Independent detections per LU

let's also create one that graphs each LU in it's own panel using facet_wrap

```
# let's also create one that graphs each LU in it's own panel using facet_wrap
det_plot_LUs <- detections %>%

  # remove less useful species
  filter(species %in% mammals) %>%

 # group by array and species to calculate dets per spp per LU
 group_by(array, species) %>%

  # calculate a column with unique accounts of each species
```

```
  reframe(count = n_distinct(event_id)) %>%

  # pipe to ggplot and set aesthetics mapping
  ggplot(aes(x = reorder(species, count), y = count)) +

  # plot as bar graph
  geom_col() +

  # plot each LU in own panel
  facet_wrap(vars(array)) +

  # add the number of detections at the end of each bar
  geom_text(aes(label = count),
            color = "black",
            size = 3,
            hjust = 0.2,
            vjust = -0.3) +

  # label x and y axis with informative titles
  labs(x = 'Species',
       y = 'Number of Independent (30 min) Detections') +

  # add title to plot with LU name

  ggtitle("LU21 Detections")+

  # set the theme
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.text.x = element_text(angle = 90,
                                   vjust = 0.5,
                                   hjust = 1,
                                   size = 12))

# view plot
det_plot_LUs
```



We can also save this plot if we want it for reports etc.

```
# save this plot

ggsave('figures/OSM_ind_det_per_LU_2022.jpg',
       det_plot_LUs,
       dpi = 600,
       width = 10,
       height = 12,
       units = 'in')
```

Covariate data

Import covariate data

These data files have a similar format so we will read them in together using the `map()` function in the *purrr* package. Reminder, the `map()` function let's us perform iterations. The `~.x` after the function is a placeholder that refers to the data before the last pipe (e.g. the .csv files we supplied) and all operations within the `map()` will be performed on all of these objects.

We are doing a few data manipulation steps here to make the data checks easier since I know how I want some of the columns/entries formatted.

The code below will

1. Provide the path and filenames of the two csv files we need

2. Read them in and specify the column types

3. Set the column names to lowercase, remove the feature_ty prefix in each column, and replace dashes with underscores

4. then create two additional columns, array and camera from the site column information

5. finally set the names of each item in the list as HFI for human footprint inventory and VEG for and landcover data

```
# these data files have a similar format so we will read them in together using the map() function in t

covariate_data <-
  # provide file path (e.g. folders to find the data)
  file.path('data/raw',

            # provide the file names
            c('OSM_LU01_LU13_LU15_LU21_HFI_2022_2024-04-19.csv',
              'OSM_LU01_LU13_LU15_LU21_VEG_2022_2024-04-19.csv')) %>%

  # use purrr map to read in files, the ~.x is a placeholder that refers to the object before the last
  map(~.x %>%
        read_csv(.,

                 # specify how to read in the various columns
                 col_types = cols(Site = col_factor(),
                                  BUFF_DIST = col_integer(),
                                  .default = col_number())) %>%



        # set the column names to lower case
        set_names(
          names(.) %>%
            tolower())) %>%


  # set the names of the two files in the list, if you don't run this they will be named numerically (e
  purrr::set_names('HFI',
                   'VEG')

# will get a warning about parsing issues, don't panic it is fine
```

Data checks

Strucutre

Even though we set some of the columns to read in as a specific type in the data import step it's always a good idea to check internal structure.

```r
str(covariate_data)
```

```
## List of 2
##  $ HFI: spc_tbl_ [3,100 x 107] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##   ..$ site                         : Factor w/ 155 levels "LU13_18","LU13_15",..: 1 2 3 4 5 6 7 8 9 10
##   ..$ buff_dist                    : int [1:3100] 250 250 250 250 250 250 250 250 250 250 ...
##   ..$ vegetated-edge-roads         : num [1:3100] 0 0.0858 0 0 0 ...
##   ..$ harvest-area                 : num [1:3100] 0 0 0.687 0.337 0 ...
##   ..$ road-gravel-1l               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ conventional-seismic         : num [1:3100] 0 0.03277 0 0.00889 0.01144 ...
##   ..$ tame_pasture                 : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ pipeline                     : num [1:3100] 0 0.068 0 0 0.0301 ...
##   ..$ road-gravel-2l               : num [1:3100] 0 0 0 0 0 ...
##   ..$ trail                        : num [1:3100] 0.00588 0.0028 0 0.00196 0 ...
##   ..$ well-bitumen                 : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ rough_pasture                : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ well-aband                   : num [1:3100] 0 0 0 0 0.0322 ...
##   ..$ road-unclassified            : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ crop                         : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ low-impact-seismic           : num [1:3100] 0 0 0 0 0.0523 ...
##   ..$ clearing-unknown             : num [1:3100] 0.0923 0.0697 0 0 0 ...
##   ..$ cultivation_abandoned        : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road-paved-undiv-2l          : num [1:3100] 0 0.0174 0 0 0 ...
##   ..$ road-unimproved              : num [1:3100] 0 0 0 0 0 ...
##   ..$ truck-trail                  : num [1:3100] 0 0 0 0.0139 0 ...
##   ..$ dugout                       : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road-paved-undiv-1l          : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ well-gas                     : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ vegetated-edge-railways      : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ harvest-area-white-zone      : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ country-residence            : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ borrowpit-dry                : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ rural-residence              : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ borrowpit-wet                : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ borrowpits                   : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ grvl-sand-pit                : num [1:3100] 0 0.0873 0 0 0 ...
##   ..$ ris-reclaimed-temp           : num [1:3100] 0 0.0477 0 0 0 ...
##   ..$ ris-clearing-unknown         : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris-drainage                 : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris-mines-oilsands           : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris-overburden-dump          : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris-facility-operations      : num [1:3100] 0 0 0 0 0 ...
##   ..$ transmission-line            : num [1:3100] 0.0642 0 0 0 0.091 ...
##   ..$ ris-tailing-pond             : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ clearing-wellpad-unconfirmed: num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ mines-oilsands               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris-soil-replaced            : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road-paved-1l                : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris-oilsands-rms             : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris-facility-unknown         : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris-borrowpits               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris-transmission-line        : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris-soil-salvaged            : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
```

```
##   ..$ ris-road                    : num [1:3100] 0 0 0 0 0 ...
##   ..$ ris-plant                   : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ urban-residence             : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ facility-other              : num [1:3100] 0 0 0 0 0 ...
##   ..$ airp-runway                 : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ runway                      : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris-reclaimed-permanent     : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ urban-industrial            : num [1:3100] 0.291 0 0 0 0 ...
##   ..$ lagoon                      : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ facility-unknown            : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ residence_clearing          : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ well-cased                  : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road-unpaved-2l             : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road-paved-3l               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ surrounding-veg             : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ rlwy-sgl-track              : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road-winter                 : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ sump                        : num [1:3100] 0 0 0 0 0 ...
##   ..$ greenspace                  : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road-paved-2l               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ well-other                  : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ canal                       : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ reservoir                   : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ well_cleared_not_confirmed  : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ misc-oil-gas-facility       : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ camp-industrial             : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris-camp-industrial         : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ oil-gas-plant               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ well-unknown                : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris-utilities               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ cfo                         : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ recreation                  : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ campground                  : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ peat                        : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ golfcourse                  : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ landfill                    : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ transfer_station            : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ mill                        : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road-paved-div              : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ rlwy-spur                   : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ well_cleared_not_drilled    : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ open-pit-mine               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ well-oil                    : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road-paved-4l               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ mines-pitlake               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris-reclaimed-certified     : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris-windrow                 : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ tailing-pond                : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ rlwy-mlt-track              : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ rlwy-dbl-track              : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   .. [list output truncated]
##   ..- attr(*, "spec")=
##   .. .. cols(
##   .. ..   .default = col_number(),
```

```
##   .. ..   Site = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   .. ..   BUFF_DIST = col_integer(),
##   .. ..   `VEGETATED-EDGE-ROADS` = col_number(),
##   .. ..   `HARVEST-AREA` = col_number(),
##   .. ..   `ROAD-GRAVEL-1L` = col_number(),
##   .. ..   `CONVENTIONAL-SEISMIC` = col_number(),
##   .. ..   TAME_PASTURE = col_number(),
##   .. ..   PIPELINE = col_number(),
##   .. ..   `ROAD-GRAVEL-2L` = col_number(),
##   .. ..   TRAIL = col_number(),
##   .. ..   `WELL-BITUMEN` = col_number(),
##   .. ..   ROUGH_PASTURE = col_number(),
##   .. ..   `WELL-ABAND` = col_number(),
##   .. ..   `ROAD-UNCLASSIFIED` = col_number(),
##   .. ..   CROP = col_number(),
##   .. ..   `LOW-IMPACT-SEISMIC` = col_number(),
##   .. ..   `CLEARING-UNKNOWN` = col_number(),
##   .. ..   CULTIVATION_ABANDONED = col_number(),
##   .. ..   `ROAD-PAVED-UNDIV-2L` = col_number(),
##   .. ..   `ROAD-UNIMPROVED` = col_number(),
##   .. ..   `TRUCK-TRAIL` = col_number(),
##   .. ..   DUGOUT = col_number(),
##   .. ..   `ROAD-PAVED-UNDIV-1L` = col_number(),
##   .. ..   `WELL-GAS` = col_number(),
##   .. ..   `VEGETATED-EDGE-RAILWAYS` = col_number(),
##   .. ..   `HARVEST-AREA-WHITE-ZONE` = col_number(),
##   .. ..   `COUNTRY-RESIDENCE` = col_number(),
##   .. ..   `BORROWPIT-DRY` = col_number(),
##   .. ..   `RURAL-RESIDENCE` = col_number(),
##   .. ..   `BORROWPIT-WET` = col_number(),
##   .. ..   BORROWPITS = col_number(),
##   .. ..   `GRVL-SAND-PIT` = col_number(),
##   .. ..   `RIS-RECLAIMED-TEMP` = col_number(),
##   .. ..   `RIS-CLEARING-UNKNOWN` = col_number(),
##   .. ..   `RIS-DRAINAGE` = col_number(),
##   .. ..   `RIS-MINES-OILSANDS` = col_number(),
##   .. ..   `RIS-OVERBURDEN-DUMP` = col_number(),
##   .. ..   `RIS-FACILITY-OPERATIONS` = col_number(),
##   .. ..   `TRANSMISSION-LINE` = col_number(),
##   .. ..   `RIS-TAILING-POND` = col_number(),
##   .. ..   `CLEARING-WELLPAD-UNCONFIRMED` = col_number(),
##   .. ..   `MINES-OILSANDS` = col_number(),
##   .. ..   `RIS-SOIL-REPLACED` = col_number(),
##   .. ..   `ROAD-PAVED-1L` = col_number(),
##   .. ..   `RIS-OILSANDS-RMS` = col_number(),
##   .. ..   `RIS-FACILITY-UNKNOWN` = col_number(),
##   .. ..   `RIS-BORROWPITS` = col_number(),
##   .. ..   `RIS-TRANSMISSION-LINE` = col_number(),
##   .. ..   `RIS-SOIL-SALVAGED` = col_number(),
##   .. ..   `RIS-ROAD` = col_number(),
##   .. ..   `RIS-PLANT` = col_number(),
##   .. ..   `URBAN-RESIDENCE` = col_number(),
##   .. ..   `FACILITY-OTHER` = col_number(),
##   .. ..   `AIRP-RUNWAY` = col_number(),
```

```
##   .. ..   RUNWAY = col_number(),
##   .. ..   `RIS-RECLAIMED-PERMANENT` = col_number(),
##   .. ..   `URBAN-INDUSTRIAL` = col_number(),
##   .. ..   LAGOON = col_number(),
##   .. ..   `FACILITY-UNKNOWN` = col_number(),
##   .. ..   RESIDENCE_CLEARING = col_number(),
##   .. ..   `WELL-CASED` = col_number(),
##   .. ..   `ROAD-UNPAVED-2L` = col_number(),
##   .. ..   `ROAD-PAVED-3L` = col_number(),
##   .. ..   `SURROUNDING-VEG` = col_number(),
##   .. ..   `RLWY-SGL-TRACK` = col_number(),
##   .. ..   `ROAD-WINTER` = col_number(),
##   .. ..   SUMP = col_number(),
##   .. ..   GREENSPACE = col_number(),
##   .. ..   `ROAD-PAVED-2L` = col_number(),
##   .. ..   `WELL-OTHER` = col_number(),
##   .. ..   CANAL = col_number(),
##   .. ..   RESERVOIR = col_number(),
##   .. ..   WELL_CLEARED_NOT_CONFIRMED = col_number(),
##   .. ..   `MISC-OIL-GAS-FACILITY` = col_number(),
##   .. ..   `CAMP-INDUSTRIAL` = col_number(),
##   .. ..   `RIS-CAMP-INDUSTRIAL` = col_number(),
##   .. ..   `OIL-GAS-PLANT` = col_number(),
##   .. ..   `WELL-UNKNOWN` = col_number(),
##   .. ..   `RIS-UTILITIES` = col_number(),
##   .. ..   CFO = col_number(),
##   .. ..   RECREATION = col_number(),
##   .. ..   CAMPGROUND = col_number(),
##   .. ..   PEAT = col_number(),
##   .. ..   GOLFCOURSE = col_number(),
##   .. ..   LANDFILL = col_number(),
##   .. ..   TRANSFER_STATION = col_number(),
##   .. ..   MILL = col_number(),
##   .. ..   `ROAD-PAVED-DIV` = col_number(),
##   .. ..   `RLWY-SPUR` = col_number(),
##   .. ..   WELL_CLEARED_NOT_DRILLED = col_number(),
##   .. ..   `OPEN-PIT-MINE` = col_number(),
##   .. ..   `WELL-OIL` = col_number(),
##   .. ..   `ROAD-PAVED-4L` = col_number(),
##   .. ..   `MINES-PITLAKE` = col_number(),
##   .. ..   `RIS-RECLAIMED-CERTIFIED` = col_number(),
##   .. ..   `RIS-WINDROW` = col_number(),
##   .. ..   `TAILING-POND` = col_number(),
##   .. ..   `RLWY-MLT-TRACK` = col_number(),
##   .. ..   `RLWY-DBL-TRACK` = col_number(),
##   .. ..   `RIS-WASTE` = col_number(),
##   .. ..   `INTERCHANGE-RAMP` = col_number(),
##   .. ..   `ROAD-PAVED-5L` = col_number(),
##   .. ..   `RIS-AIRP-RUNWAY` = col_number(),
##   .. ..   `FRUIT-VEGETABLES` = col_number(),
##   .. ..   `ROAD-UNPAVED-1L` = col_number(),
##   .. ..   `RIS-RECLAIM-READY` = col_number(),
##   .. ..   `RIS-TANK-FARM` = col_number()
##   .. .. )
```

```
##   ..- attr(*, "problems")=<externalptr>
##  $ VEG: spc_tbl_ [3,100 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##   ..$ site       : Factor w/ 155 levels "LU13_18","LU13_15",..: 1 2 3 4 5 6 7 8 9 10 ...
##   ..$ buff_dist  : int [1:3100] 250 250 250 250 250 250 250 250 250 250 ...
##   ..$ lc_class20 : num [1:3100] 0.0361 0 0 0 0 ...
##   ..$ lc_class32 : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ lc_class33 : num [1:3100] 0 0.101 0 0 0 ...
##   ..$ lc_class34 : num [1:3100] 0 0.0916 0 0 0 ...
##   ..$ lc_class50 : num [1:3100] 0.316 0 0.559 0 0 ...
##   ..$ lc_class110: num [1:3100] 0.193 0.348 0 0 0.178 ...
##   ..$ lc_class120: num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ lc_class210: num [1:3100] 0.456 0.358 0.186 1 0.822 ...
##   ..$ lc_class220: num [1:3100] 0 0 0 0 0 ...
##   ..$ lc_class230: num [1:3100] 0 0.101 0.255 0 0 ...
##   ..- attr(*, "spec")=
##   .. .. cols(
##   .. ..   .default = col_number(),
##   .. ..   Site = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   .. ..   BUFF_DIST = col_integer(),
##   .. ..   LC_class20 = col_number(),
##   .. ..   LC_class32 = col_number(),
##   .. ..   LC_class33 = col_number(),
##   .. ..   LC_class34 = col_number(),
##   .. ..   LC_class50 = col_number(),
##   .. ..   LC_class110 = col_number(),
##   .. ..   LC_class120 = col_number(),
##   .. ..   LC_class210 = col_number(),
##   .. ..   LC_class220 = col_number(),
##   .. ..   LC_class230 = col_number()
##   .. .. )
##   ..- attr(*, "problems")=<externalptr>
```

From a quick glance everything looks good.

Sites

Now let's check that all the sites are accounted for, there should be 155 just like with the timelapse data

```
# check that the sites are all there and entered correctly, there should be 155

# since the data sets are in a list we need to call the list first, then the data name in the list, the
levels(covariate_data$HFI$site)
```

```
##   [1] "LU13_18"  "LU13_15"  "LU13_03"  "LU13_34"  "LU13_57"  "LU13_16"
##   [7] "LU13_21"  "LU13_37"  "LU13_55"  "LU13_47"  "LU13_51"  "LU13_27"
##  [13] "LU13_05"  "LU13_26"  "LU13_35"  "LU13_13"  "LU13_128" "LU13_131"
##  [19] "LU13_12"  "LU13_33"  "LU13_32"  "LU13_30"  "LU13_53"  "LU13_22"
##  [25] "LU13_19"  "LU13_45"  "LU13_08"  "LU13_43"  "LU13_41"  "LU13_59"
##  [31] "LU13_06"  "LU13_14"  "LU13_52"  "LU13_36"  "LU13_38"  "LU13_20"
##  [37] "LU13_17"  "LU13_56"  "LU13_49"  "LU13_70"  "LU13_11"  "LU15_44"
##  [43] "LU15_11"  "LU15_19"  "LU15_07"  "LU15_34"  "LU15_37"  "LU15_18"
##  [49] "LU15_43"  "LU15_10"  "LU15_58"  "LU15_08"  "LU15_12"  "LU15_17"
##  [55] "LU15_01"  "LU15_26"  "LU15_27"  "LU15_28"  "LU15_02"  "LU15_29"
```

```
##  [61] "LU15_30"  "LU15_31"  "LU15_32"  "LU15_20"  "LU15_21"  "LU15_23"
##  [67] "LU15_41"  "LU15_25"  "LU15_09"  "LU15_24"  "LU15_46"  "LU15_40"
##  [73] "LU15_16"  "LU15_04"  "LU15_36"  "LU15_03"  "LU15_15"  "LU15_14"
##  [79] "LU15_22"  "LU15_61"  "LU21_106" "LU21_10"  "LU21_16"  "LU21_107"
##  [85] "LU21_41"  "LU21_63"  "LU21_27"  "LU21_23"  "LU21_21"  "LU21_82"
##  [91] "LU21_871" "LU21_164" "LU21_36"  "LU21_14"  "LU21_32"  "LU21_126"
##  [97] "LU21_122" "LU21_56"  "LU21_57"  "LU21_09"  "LU21_98"  "LU21_119"
## [103] "LU21_52"  "LU21_59"  "LU21_68"  "LU21_109" "LU21_93"  "LU21_114"
## [109] "LU21_105" "LU21_100" "LU21_74"  "LU21_78"  "LU21_153" "LU21_06"
## [115] "LU21_116" "LU21_97"  "LU01_13"  "LU01_30"  "LU01_85"  "LU01_66"
## [121] "LU01_76"  "LU01_73"  "LU01_75"  "LU01_47"  "LU01_74"  "LU01_44"
## [127] "LU01_41"  "LU01_64"  "LU01_86"  "LU01_79"  "LU01_63"  "LU01_72"
## [133] "LU01_67"  "LU01_77"  "LU01_32"  "LU01_36"  "LU01_84"  "LU01_25"
## [139] "LU01_83"  "LU01_80"  "LU01_70"  "LU01_48"  "LU01_45"  "LU01_11"
## [145] "LU01_40"  "LU01_06"  "LU01_10"  "LU01_43"  "LU01_71"  "LU01_22"
## [151] "LU01_60"  "LU01_78"  "LU01_46"  "LU01_82"  "LU01_27"
```

```
levels(covariate_data$VEG$site)
```

```
##   [1] "LU13_18"  "LU13_15"  "LU13_03"  "LU13_34"  "LU13_57"  "LU13_16"
##   [7] "LU13_21"  "LU13_37"  "LU13_55"  "LU13_47"  "LU13_51"  "LU13_27"
##  [13] "LU13_05"  "LU13_26"  "LU13_35"  "LU13_13"  "LU13_128" "LU13_131"
##  [19] "LU13_12"  "LU13_33"  "LU13_32"  "LU13_30"  "LU13_53"  "LU13_22"
##  [25] "LU13_19"  "LU13_45"  "LU13_08"  "LU13_43"  "LU13_41"  "LU13_59"
##  [31] "LU13_06"  "LU13_14"  "LU13_52"  "LU13_36"  "LU13_38"  "LU13_20"
##  [37] "LU13_17"  "LU13_56"  "LU13_49"  "LU13_70"  "LU13_11"  "LU15_44"
##  [43] "LU15_11"  "LU15_19"  "LU15_07"  "LU15_34"  "LU15_37"  "LU15_18"
##  [49] "LU15_43"  "LU15_10"  "LU15_58"  "LU15_08"  "LU15_12"  "LU15_17"
##  [55] "LU15_01"  "LU15_26"  "LU15_27"  "LU15_28"  "LU15_02"  "LU15_29"
##  [61] "LU15_30"  "LU15_31"  "LU15_32"  "LU15_20"  "LU15_21"  "LU15_23"
##  [67] "LU15_41"  "LU15_25"  "LU15_09"  "LU15_24"  "LU15_46"  "LU15_40"
##  [73] "LU15_16"  "LU15_04"  "LU15_36"  "LU15_03"  "LU15_15"  "LU15_14"
##  [79] "LU15_22"  "LU15_61"  "LU21_106" "LU21_10"  "LU21_16"  "LU21_107"
##  [85] "LU21_41"  "LU21_63"  "LU21_27"  "LU21_23"  "LU21_21"  "LU21_82"
##  [91] "LU21_871" "LU21_164" "LU21_36"  "LU21_14"  "LU21_32"  "LU21_126"
##  [97] "LU21_122" "LU21_56"  "LU21_57"  "LU21_09"  "LU21_98"  "LU21_119"
## [103] "LU21_52"  "LU21_59"  "LU21_68"  "LU21_109" "LU21_93"  "LU21_114"
## [109] "LU21_105" "LU21_100" "LU21_74"  "LU21_78"  "LU21_153" "LU21_06"
## [115] "LU21_116" "LU21_97"  "LU01_13"  "LU01_30"  "LU01_85"  "LU01_66"
## [121] "LU01_76"  "LU01_73"  "LU01_75"  "LU01_47"  "LU01_74"  "LU01_44"
## [127] "LU01_41"  "LU01_64"  "LU01_86"  "LU01_79"  "LU01_63"  "LU01_72"
## [133] "LU01_67"  "LU01_77"  "LU01_32"  "LU01_36"  "LU01_84"  "LU01_25"
## [139] "LU01_83"  "LU01_80"  "LU01_70"  "LU01_48"  "LU01_45"  "LU01_11"
## [145] "LU01_40"  "LU01_06"  "LU01_10"  "LU01_43"  "LU01_71"  "LU01_22"
## [151] "LU01_60"  "LU01_78"  "LU01_46"  "LU01_82"  "LU01_27"
```

There are 155 for both data sets and I don't see any glaring issues but we can use a function in R to check that these match perfectly.

If you are getting a NULL output for any of the `setdiff()` lines of code, then you need to make sure that site is actually a factor in both the data sets. Sometimes data manipulation steps change how the variable is reading in R so you made need to fix this.

```
# there are 155 for both and don't see any glaring issues but let's check that all these site names mat
setdiff(levels(covariate_data$VEG$site),
        levels(covariate_data$HFI$site))
```

```
## character(0)
```

```
# no mismatches

# we need to check that they also match the osm_2022_Det
setdiff(levels(detections$site),
        levels(covariate_data$HFI$site))
```

```
## character(0)
```

```
# [1] "LU12_51" "LU15_35"
# there seems to be two site names that are different between the covariate data sets and the detection

# reverse the order to see which two are extras in the covariate data
setdiff(levels(covariate_data$HFI$site),
        levels(detections$site))
```

```
## character(0)
```

```
# [1] "LU13_51" "LU13_35" it looks like the landscape units might have gotten typed in wrong. # checked
```

> We fixed the site issue in the raw data since it's important that the site names are correct so you won't see a difference between the data sets now but I've left the code as an example.

Column names

We should check that the column names all look good, there are a ton for the HFI data frame so we won't look at each of the features individually but check that the general formatting/naming is okay

```
names(covariate_data$HFI)
```

```
##  [1] "site"                      "buff_dist"
##  [3] "vegetated-edge-roads"      "harvest-area"
##  [5] "road-gravel-1l"            "conventional-seismic"
##  [7] "tame_pasture"              "pipeline"
##  [9] "road-gravel-2l"            "trail"
## [11] "well-bitumen"              "rough_pasture"
## [13] "well-aband"                "road-unclassified"
## [15] "crop"                      "low-impact-seismic"
## [17] "clearing-unknown"          "cultivation_abandoned"
## [19] "road-paved-undiv-2l"       "road-unimproved"
## [21] "truck-trail"               "dugout"
## [23] "road-paved-undiv-1l"       "well-gas"
## [25] "vegetated-edge-railways"   "harvest-area-white-zone"
## [27] "country-residence"         "borrowpit-dry"
## [29] "rural-residence"           "borrowpit-wet"
```

```
##  [31] "borrowpits"                    "grvl-sand-pit"
##  [33] "ris-reclaimed-temp"            "ris-clearing-unknown"
##  [35] "ris-drainage"                  "ris-mines-oilsands"
##  [37] "ris-overburden-dump"           "ris-facility-operations"
##  [39] "transmission-line"             "ris-tailing-pond"
##  [41] "clearing-wellpad-unconfirmed" "mines-oilsands"
##  [43] "ris-soil-replaced"             "road-paved-1l"
##  [45] "ris-oilsands-rms"              "ris-facility-unknown"
##  [47] "ris-borrowpits"                "ris-transmission-line"
##  [49] "ris-soil-salvaged"             "ris-road"
##  [51] "ris-plant"                     "urban-residence"
##  [53] "facility-other"                "airp-runway"
##  [55] "runway"                        "ris-reclaimed-permanent"
##  [57] "urban-industrial"              "lagoon"
##  [59] "facility-unknown"              "residence_clearing"
##  [61] "well-cased"                    "road-unpaved-2l"
##  [63] "road-paved-3l"                 "surrounding-veg"
##  [65] "rlwy-sgl-track"                "road-winter"
##  [67] "sump"                          "greenspace"
##  [69] "road-paved-2l"                 "well-other"
##  [71] "canal"                         "reservoir"
##  [73] "well_cleared_not_confirmed"    "misc-oil-gas-facility"
##  [75] "camp-industrial"               "ris-camp-industrial"
##  [77] "oil-gas-plant"                 "well-unknown"
##  [79] "ris-utilities"                 "cfo"
##  [81] "recreation"                    "campground"
##  [83] "peat"                          "golfcourse"
##  [85] "landfill"                      "transfer_station"
##  [87] "mill"                          "road-paved-div"
##  [89] "rlwy-spur"                     "well_cleared_not_drilled"
##  [91] "open-pit-mine"                 "well-oil"
##  [93] "road-paved-4l"                 "mines-pitlake"
##  [95] "ris-reclaimed-certified"       "ris-windrow"
##  [97] "tailing-pond"                  "rlwy-mlt-track"
##  [99] "rlwy-dbl-track"                "ris-waste"
## [101] "interchange-ramp"              "road-paved-5l"
## [103] "ris-airp-runway"               "fruit-vegetables"
## [105] "road-unpaved-1l"               "ris-reclaim-ready"
## [107] "ris-tank-farm"
```

These look okay but we should replace the dash '-' with and underscore '_' to match formatting of other files and because it's easier for R to work with.

We also want to add array and camera columns which we can do using the site data.

Let's check the VEG data too

```
names(covariate_data$VEG)
```

```
##  [1] "site"        "buff_dist"   "lc_class20"  "lc_class32"  "lc_class33"
##  [6] "lc_class34"  "lc_class50"  "lc_class110" "lc_class120" "lc_class210"
## [11] "lc_class220" "lc_class230"
```

We also need to add array and camera columns which we can do using the site data.

NAs

Let's check the summary for any NAs that shouldn't be in the data, mostly we are looking for NAs in the site or buff_dist columns

```
summary(covariate_data$HFI)
```

```
##       site        buff_dist    vegetated-edge-roads  harvest-area
##  LU13_18:  20   Min.   : 250   Min.   :0.000000     Min.   :0.00000
##  LU13_15:  20   1st Qu.:1438   1st Qu.:0.001371     1st Qu.:0.00000
##  LU13_03:  20   Median :2625   Median :0.006426     Median :0.00000
##  LU13_34:  20   Mean   :2625   Mean   :0.011329     Mean   :0.01842
##  LU13_57:  20   3rd Qu.:3812   3rd Qu.:0.015557     3rd Qu.:0.01229
##  LU13_16:  20   Max.   :5000   Max.   :0.147883     Max.   :0.83674
##  (Other):2980
##  road-gravel-1l     conventional-seismic  tame_pasture          pipeline
##  Min.   :0.000000   Min.   :0.000000     Min.   :0.0000000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.002687     1st Qu.:0.0000000   1st Qu.:0.00000
##  Median :0.001659   Median :0.006601     Median :0.0000000   Median :0.01351
##  Mean   :0.003375   Mean   :0.006732     Mean   :0.0006704   Mean   :0.01937
##  3rd Qu.:0.004672   3rd Qu.:0.009984     3rd Qu.:0.0000000   3rd Qu.:0.02812
##  Max.   :0.038085   Max.   :0.045512     Max.   :0.1130783   Max.   :0.28896
##
##  road-gravel-2l          trail            well-bitumen      rough_pasture
##  Min.   :0.0000000   Min.   :0.000e+00   Min.   :0.000000   Min.   :0.0000000
##  1st Qu.:0.0000000   1st Qu.:5.278e-05   1st Qu.:0.000000   1st Qu.:0.0000000
##  Median :0.0000000   Median :4.617e-04   Median :0.000000   Median :0.0000000
##  Mean   :0.0012398   Mean   :8.525e-04   Mean   :0.006976   Mean   :0.0002369
##  3rd Qu.:0.0007902   3rd Qu.:1.236e-03   3rd Qu.:0.005984   3rd Qu.:0.0000000
##  Max.   :0.0438815   Max.   :1.977e-02   Max.   :0.187398   Max.   :0.0828324
##
##    well-aband        road-unclassified        crop   low-impact-seismic
##  Min.   :0.0000000   Min.   :0.000e+00   Min.   :0   Min.   :0.000000
##  1st Qu.:0.0002483   1st Qu.:0.000e+00   1st Qu.:0   1st Qu.:0.000000
##  Median :0.0017255   Median :0.000e+00   Median :0   Median :0.000000
##  Mean   :0.0051658   Mean   :1.663e-06   Mean   :0   Mean   :0.004302
##  3rd Qu.:0.0079168   3rd Qu.:0.000e+00   3rd Qu.:0   3rd Qu.:0.001362
##  Max.   :0.3045402   Max.   :4.325e-04   Max.   :0   Max.   :0.087576
##
##  clearing-unknown    cultivation_abandoned road-paved-undiv-2l
##  Min.   :0.0000000   Min.   :0.000e+00     Min.   :0.0000000
##  1st Qu.:0.0000000   1st Qu.:0.000e+00     1st Qu.:0.0000000
##  Median :0.0001044   Median :0.000e+00     Median :0.0000000
##  Mean   :0.0052000   Mean   :3.281e-05     Mean   :0.0004598
##  3rd Qu.:0.0029293   3rd Qu.:0.000e+00     3rd Qu.:0.0000000
##  Max.   :0.4023522   Max.   :3.115e-02     Max.   :0.0431664
##
##  road-unimproved      truck-trail            dugout
##  Min.   :0.0000000   Min.   :0.0000000   Min.   :0.000e+00
##  1st Qu.:0.0000000   1st Qu.:0.0000000   1st Qu.:0.000e+00
##  Median :0.0000851   Median :0.0000000   Median :0.000e+00
##  Mean   :0.0008519   Mean   :0.0006638   Mean   :2.836e-06
##  3rd Qu.:0.0010399   3rd Qu.:0.0002740   3rd Qu.:0.000e+00
##  Max.   :0.0532898   Max.   :0.0386512   Max.   :9.757e-04
```

```
##
##  road-paved-undiv-1l    well-gas          vegetated-edge-railways
##  Min.   :0.0000000   Min.   :0.0000000   Min.   :0
##  1st Qu.:0.0000000   1st Qu.:0.0000000   1st Qu.:0
##  Median :0.0000000   Median :0.0000000   Median :0
##  Mean   :0.0001018   Mean   :0.0001398   Mean   :0
##  3rd Qu.:0.0000000   3rd Qu.:0.0000000   3rd Qu.:0
##  Max.   :0.0214676   Max.   :0.0121347   Max.   :0
##
##  harvest-area-white-zone country-residence   borrowpit-dry
##  Min.   :0.0000000       Min.   :0.000e+00   Min.   :0.000e+00
##  1st Qu.:0.0000000       1st Qu.:0.000e+00   1st Qu.:0.000e+00
##  Median :0.0000000       Median :0.000e+00   Median :0.000e+00
##  Mean   :0.0002575       Mean   :7.275e-05   Mean   :8.612e-04
##  3rd Qu.:0.0000000       3rd Qu.:0.000e+00   3rd Qu.:2.485e-05
##  Max.   :0.0543438       Max.   :1.714e-02   Max.   :1.039e-01
##
##  rural-residence     borrowpit-wet         borrowpits        grvl-sand-pit
##  Min.   :0.000e+00   Min.   :0.0000000   Min.   :0.0000000   Min.   :0.000000
##  1st Qu.:0.000e+00   1st Qu.:0.0000000   1st Qu.:0.0000000   1st Qu.:0.000000
##  Median :0.000e+00   Median :0.0000000   Median :0.0000000   Median :0.000000
##  Mean   :1.299e-05   Mean   :0.0005352   Mean   :0.0002596   Mean   :0.002136
##  3rd Qu.:0.000e+00   3rd Qu.:0.0000000   3rd Qu.:0.0000000   3rd Qu.:0.000000
##  Max.   :1.842e-03   Max.   :0.1890071   Max.   :0.1163709   Max.   :0.557858
##
##  ris-reclaimed-temp  ris-clearing-unknown  ris-drainage
##  Min.   :0.0000000   Min.   :0.0000000    Min.   :0.000e+00
##  1st Qu.:0.0000000   1st Qu.:0.0000000    1st Qu.:0.000e+00
##  Median :0.0000000   Median :0.0000000    Median :0.000e+00
##  Mean   :0.0002021   Mean   :0.0003988    Mean   :8.739e-05
##  3rd Qu.:0.0000000   3rd Qu.:0.0000000    3rd Qu.:0.000e+00
##  Max.   :0.0476953   Max.   :0.0493557    Max.   :1.682e-02
##
##  ris-mines-oilsands  ris-overburden-dump ris-facility-operations
##  Min.   :0.000e+00   Min.   :0.0000000   Min.   :0.0000000
##  1st Qu.:0.000e+00   1st Qu.:0.0000000   1st Qu.:0.0000000
##  Median :0.000e+00   Median :0.0000000   Median :0.0000000
##  Mean   :8.053e-05   Mean   :0.0001444   Mean   :0.0003609
##  3rd Qu.:0.000e+00   3rd Qu.:0.0000000   3rd Qu.:0.0000000
##  Max.   :5.667e-02   Max.   :0.0211145   Max.   :0.1274343
##
##  transmission-line  ris-tailing-pond   clearing-wellpad-unconfirmed
##  Min.   :0.000000   Min.   :0.000000   Min.   :0.0000000
##  1st Qu.:0.000000   1st Qu.:0.000000   1st Qu.:0.0000000
##  Median :0.000000   Median :0.000000   Median :0.0000000
##  Mean   :0.005587   Mean   :0.001151   Mean   :0.0003957
##  3rd Qu.:0.007075   3rd Qu.:0.000000   3rd Qu.:0.0004178
##  Max.   :0.173950   Max.   :0.173817   Max.   :0.0723607
##
##  mines-oilsands      ris-soil-replaced   road-paved-1l ris-oilsands-rms
##  Min.   :0.0000000   Min.   :0.0000000   Min.   :0     Min.   :0.0000000
##  1st Qu.:0.0000000   1st Qu.:0.0000000   1st Qu.:0     1st Qu.:0.0000000
##  Median :0.0000000   Median :0.0000000   Median :0     Median :0.0000000
##  Mean   :0.0008976   Mean   :0.0001588   Mean   :0     Mean   :0.0002206
```

```
##  3rd Qu.:0.0000000   3rd Qu.:0.0000000   3rd Qu.:0     3rd Qu.:0.0000000
##  Max.   :0.1223456   Max.   :0.0244751   Max.   :0     Max.   :0.0334971
##
##  ris-facility-unknown ris-borrowpits      ris-transmission-line
##  Min.   :0.000e+00    Min.   :0.000e+00   Min.   :0.000e+00
##  1st Qu.:0.000e+00    1st Qu.:0.000e+00   1st Qu.:0.000e+00
##  Median :0.000e+00    Median :0.000e+00   Median :0.000e+00
##  Mean   :5.029e-08    Mean   :2.983e-05   Mean   :9.810e-06
##  3rd Qu.:0.000e+00    3rd Qu.:0.000e+00   3rd Qu.:0.000e+00
##  Max.   :2.780e-05    Max.   :5.063e-03   Max.   :2.667e-03
##
##  ris-soil-salvaged     ris-road           ris-plant urban-residence
##  Min.   :0.000000   Min.   :0.0000000   Min.   :0   Min.   :0.000e+00
##  1st Qu.:0.000000   1st Qu.:0.0000000   1st Qu.:0   1st Qu.:0.000e+00
##  Median :0.000000   Median :0.0000000   Median :0   Median :0.000e+00
##  Mean   :0.000141   Mean   :0.0001807   Mean   :0   Mean   :6.162e-05
##  3rd Qu.:0.000000   3rd Qu.:0.0000000   3rd Qu.:0   3rd Qu.:0.000e+00
##  Max.   :0.041476   Max.   :0.0218055   Max.   :0   Max.   :1.157e-02
##
##  facility-other       airp-runway     runway          ris-reclaimed-permanent
##  Min.   :0.0000000   Min.   :0    Min.   :0.000e+00   Min.   :0.0000000
##  1st Qu.:0.0000000   1st Qu.:0    1st Qu.:0.000e+00   1st Qu.:0.0000000
##  Median :0.0000000   Median :0    Median :0.000e+00   Median :0.0000000
##  Mean   :0.0008448   Mean   :0    Mean   :3.595e-05   Mean   :0.0006082
##  3rd Qu.:0.0000000   3rd Qu.:0    3rd Qu.:0.000e+00   3rd Qu.:0.0000000
##  Max.   :0.2009920   Max.   :0    Max.   :1.525e-02   Max.   :0.0534939
##
##  urban-industrial       lagoon          facility-unknown    residence_clearing
##  Min.   :0.000000   Min.   :0.0000000   Min.   :0.000e+00   Min.   :0.000e+00
##  1st Qu.:0.000000   1st Qu.:0.0000000   1st Qu.:0.000e+00   1st Qu.:0.000e+00
##  Median :0.000000   Median :0.0000000   Median :0.000e+00   Median :0.000e+00
##  Mean   :0.001642   Mean   :0.0001599   Mean   :7.673e-05   Mean   :1.126e-05
##  3rd Qu.:0.000000   3rd Qu.:0.0000000   3rd Qu.:0.000e+00   3rd Qu.:0.000e+00
##  Max.   :0.335749   Max.   :0.0218390   Max.   :8.132e-03   Max.   :3.113e-03
##
##    well-cased        road-unpaved-2l road-paved-3l surrounding-veg
##  Min.   :0.0000000   Min.   :0       Min.   :0     Min.   :0.000e+00
##  1st Qu.:0.0000000   1st Qu.:0       1st Qu.:0     1st Qu.:0.000e+00
##  Median :0.0000000   Median :0       Median :0     Median :0.000e+00
##  Mean   :0.0001998   Mean   :0       Mean   :0     Mean   :5.107e-06
##  3rd Qu.:0.0000000   3rd Qu.:0       3rd Qu.:0     3rd Qu.:0.000e+00
##  Max.   :0.0148472   Max.   :0       Max.   :0     Max.   :2.187e-03
##
##  rlwy-sgl-track  road-winter      sump             greenspace
##  Min.   :0       Min.   :0    Min.   :0.000e+00   Min.   :0.000e+00
##  1st Qu.:0       1st Qu.:0    1st Qu.:0.000e+00   1st Qu.:0.000e+00
##  Median :0       Median :0    Median :7.935e-05   Median :0.000e+00
##  Mean   :0       Mean   :0    Mean   :2.453e-03   Mean   :9.307e-06
##  3rd Qu.:0       3rd Qu.:0    3rd Qu.:2.165e-03   3rd Qu.:0.000e+00
##  Max.   :0       Max.   :0    Max.   :3.111e-01   Max.   :2.285e-03
##
##  road-paved-2l   well-other           canal     reservoir
##  Min.   :0     Min.   :0.000000   Min.   :0   Min.   :0.000e+00
##  1st Qu.:0     1st Qu.:0.000000   1st Qu.:0   1st Qu.:0.000e+00
```

```
##  Median :0     Median :0.000000   Median :0   Median :0.000e+00
##  Mean   :0     Mean   :0.001166   Mean   :0   Mean   :1.253e-05
##  3rd Qu.:0     3rd Qu.:0.001003   3rd Qu.:0   3rd Qu.:0.000e+00
##  Max.   :0     Max.   :0.116479   Max.   :0   Max.   :7.894e-03
##
##  well_cleared_not_confirmed misc-oil-gas-facility camp-industrial
##  Min.   :0.0000000          Min.   :0.0000000     Min.   :0.0000000
##  1st Qu.:0.0000000          1st Qu.:0.0000000     1st Qu.:0.0000000
##  Median :0.0000000          Median :0.0000000     Median :0.0000000
##  Mean   :0.0003304          Mean   :0.0023058     Mean   :0.0002457
##  3rd Qu.:0.0000000          3rd Qu.:0.0009032     3rd Qu.:0.0000000
##  Max.   :0.0829690          Max.   :0.2055261     Max.   :0.0374370
##
##  ris-camp-industrial oil-gas-plant      well-unknown       ris-utilities
##  Min.   :0           Min.   :0.00000   Min.   :0.000e+00   Min.   :0.000e+00
##  1st Qu.:0           1st Qu.:0.00000   1st Qu.:0.000e+00   1st Qu.:0.000e+00
##  Median :0           Median :0.00000   Median :0.000e+00   Median :0.000e+00
##  Mean   :0           Mean   :0.00117   Mean   :1.435e-05   Mean   :7.639e-06
##  3rd Qu.:0           3rd Qu.:0.00000   3rd Qu.:0.000e+00   3rd Qu.:0.000e+00
##  Max.   :0           Max.   :0.07128   Max.   :4.314e-03   Max.   :2.539e-03
##
##       cfo              recreation   campground             peat     golfcourse
##  Min.   :0.000e+00   Min.   :0    Min.   :0.000e+00   Min.   :0   Min.   :0
##  1st Qu.:0.000e+00   1st Qu.:0    1st Qu.:0.000e+00   1st Qu.:0   1st Qu.:0
##  Median :0.000e+00   Median :0    Median :0.000e+00   Median :0   Median :0
##  Mean   :8.114e-07   Mean   :0    Mean   :3.973e-05   Mean   :0   Mean   :0
##  3rd Qu.:0.000e+00   3rd Qu.:0    3rd Qu.:0.000e+00   3rd Qu.:0   3rd Qu.:0
##  Max.   :1.217e-03   Max.   :0    Max.   :1.337e-02   Max.   :0   Max.   :0
##
##     landfill transfer_station      mill   road-paved-div        rlwy-spur
##  Min.   :0   Min.   :0        Min.   :0   Min.   :0.000e+00   Min.   :0
##  1st Qu.:0   1st Qu.:0        1st Qu.:0   1st Qu.:0.000e+00   1st Qu.:0
##  Median :0   Median :0        Median :0   Median :0.000e+00   Median :0
##  Mean   :0   Mean   :0        Mean   :0   Mean   :6.770e-06   Mean   :0
##  3rd Qu.:0   3rd Qu.:0        3rd Qu.:0   3rd Qu.:0.000e+00   3rd Qu.:0
##  Max.   :0   Max.   :0        Max.   :0   Max.   :1.936e-03   Max.   :0
##
##  well_cleared_not_drilled open-pit-mine         well-oil road-paved-4l
##  Min.   :0                Min.   :0.000000   Min.   :0   Min.   :0
##  1st Qu.:0                1st Qu.:0.000000   1st Qu.:0   1st Qu.:0
##  Median :0                Median :0.000000   Median :0   Median :0
##  Mean   :0                Mean   :0.000569   Mean   :0   Mean   :0
##  3rd Qu.:0                3rd Qu.:0.000000   3rd Qu.:0   3rd Qu.:0
##  Max.   :0                Max.   :0.038946   Max.   :0   Max.   :0
##
##  mines-pitlake ris-reclaimed-certified  ris-windrow         tailing-pond
##  Min.   :0     Min.   :0               Min.   :0.000e+00   Min.   :0.000e+00
##  1st Qu.:0     1st Qu.:0               1st Qu.:0.000e+00   1st Qu.:0.000e+00
##  Median :0     Median :0               Median :0.000e+00   Median :0.000e+00
##  Mean   :0     Mean   :0               Mean   :3.354e-05   Mean   :2.034e-05
##  3rd Qu.:0     3rd Qu.:0               3rd Qu.:0.000e+00   3rd Qu.:0.000e+00
##  Max.   :0     Max.   :0               Max.   :1.595e-02   Max.   :4.008e-03
##
##  rlwy-mlt-track rlwy-dbl-track   ris-waste interchange-ramp road-paved-5l
```

```
##  Min.   :0      Min.   :0      Min.   :0   Min.   :0        Min.   :0
##  1st Qu.:0      1st Qu.:0      1st Qu.:0   1st Qu.:0        1st Qu.:0
##  Median :0      Median :0      Median :0   Median :0        Median :0
##  Mean   :0      Mean   :0      Mean   :0   Mean   :0        Mean   :0
##  3rd Qu.:0      3rd Qu.:0      3rd Qu.:0   3rd Qu.:0        3rd Qu.:0
##  Max.   :0      Max.   :0      Max.   :0   Max.   :0        Max.   :0
##
##  ris-airp-runway fruit-vegetables road-unpaved-1l ris-reclaim-ready
##  Min.   :0       Min.   :0        Min.   :0       Min.   :0
##  1st Qu.:0       1st Qu.:0        1st Qu.:0       1st Qu.:0
##  Median :0       Median :0        Median :0       Median :0
##  Mean   :0       Mean   :0        Mean   :0       Mean   :0
##  3rd Qu.:0       3rd Qu.:0        3rd Qu.:0       3rd Qu.:0
##  Max.   :0       Max.   :0        Max.   :0       Max.   :0
##
##  ris-tank-farm
##  Min.   :0
##  1st Qu.:0
##  Median :0
##  Mean   :0
##  3rd Qu.:0
##  Max.   :0
##
```

```
summary(covariate_data$VEG)
```

```
##       site        buff_dist      lc_class20        lc_class32
##  LU13_18:  20   Min.   : 250   Min.   :0.00000   Min.   :0.000e+00
##  LU13_15:  20   1st Qu.:1438   1st Qu.:0.00000   1st Qu.:0.000e+00
##  LU13_03:  20   Median :2625   Median :0.00000   Median :0.000e+00
##  LU13_34:  20   Mean   :2625   Mean   :0.02741   Mean   :1.749e-05
##  LU13_57:  20   3rd Qu.:3812   3rd Qu.:0.03363   3rd Qu.:0.000e+00
##  LU13_16:  20   Max.   :5000   Max.   :0.51965   Max.   :1.176e-02
##  (Other):2980
##    lc_class33          lc_class34         lc_class50       lc_class110
##  Min.   :0.0000000   Min.   :0.000000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.0000000   1st Qu.:0.000000   1st Qu.:0.02385   1st Qu.:0.01974
##  Median :0.0000000   Median :0.004044   Median :0.05722   Median :0.03876
##  Mean   :0.0046114   Mean   :0.030309   Mean   :0.07933   Mean   :0.04838
##  3rd Qu.:0.0005707   3rd Qu.:0.038147   3rd Qu.:0.11539   3rd Qu.:0.06222
##  Max.   :0.3240282   Max.   :0.557095   Max.   :0.60826   Max.   :0.73189
##
##   lc_class120         lc_class210      lc_class220        lc_class230
##  Min.   :0.0000000   Min.   :0.0000   Min.   :0.000000   Min.   :0.00000
##  1st Qu.:0.0000000   1st Qu.:0.4607   1st Qu.:0.002329   1st Qu.:0.01220
##  Median :0.0000000   Median :0.6750   Median :0.044961   Median :0.03595
##  Mean   :0.0007558   Mean   :0.6324   Mean   :0.113321   Mean   :0.06342
##  3rd Qu.:0.0000000   3rd Qu.:0.8364   3rd Qu.:0.154818   3rd Qu.:0.08412
##  Max.   :0.1212155   Max.   :1.0000   Max.   :0.971737   Max.   :0.72109
##
```

Everything looks good!

Data formatting

As with the previouos sections this section will likely change each year but offers a good starting point, and I do all the data manipualtion in one code chunk but run each portion individually as I build the chunk to make sure it's working.

This code will do the following data formatting on both files simultaneously using purrr::map

1. Change the column names - remove the feature_ty prefix in each column and replace dashes with underscores

2. then create two additional columns, array and camera from the site column information

3. finally set the new variables as factors

```
covariate_data_fixed <- covariate_data %>%

 map(
   ~.x %>%

     set_names(
       names(.) %>%
         # remove the FEATURE_TY in front of all the column names because it's not helpful
         str_remove(pattern = "feature_ty") %>%

         # replace the '.' with '_' in the feature column names
         str_replace_all(pattern = '-', # provide the character pattern to look for (if you don't keep
                         replacement = '_')) %>%  # what you want the pattern to be replaced with

     separate_wider_delim(site,
                          delim = '_',
                          names = c('array',
                                    'camera'),
                          cols_remove = FALSE) %>%

     # specify format of new columns
     mutate(
       array = as.factor(array),
       camera = as.factor(camera)
     ))
```

Now let's recheck the data, data structure, and the sites with the deployment data, you can run each of these individually or all at once and review each one

```
# check structure of variables
str(covariate_data_fixed)
```

```
## List of 2
##  $ HFI: tibble [3,100 x 109] (S3: tbl_df/tbl/data.frame)
##   ..$ array                      : Factor w/ 4 levels "LU01","LU13",..: 2 2 2 2 2 2 2 2 2 2 ...
##   ..$ camera                     : Factor w/ 96 levels "01","02","03",..: 32 27 3 48 67 29 35 51 65
##   ..$ site                       : Factor w/ 155 levels "LU13_18","LU13_15",..: 1 2 3 4 5 6 7 8 9 1
##   ..$ buff_dist                  : int [1:3100] 250 250 250 250 250 250 250 250 250 250 ...
```

```
##   ..$ vegetated_edge_roads        : num [1:3100] 0 0.0858 0 0 0 ...
##   ..$ harvest_area                : num [1:3100] 0 0 0.687 0.337 0 ...
##   ..$ road_gravel_1l              : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ conventional_seismic        : num [1:3100] 0 0.03277 0 0.00889 0.01144 ...
##   ..$ tame_pasture                : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ pipeline                    : num [1:3100] 0 0.068 0 0 0.0301 ...
##   ..$ road_gravel_2l              : num [1:3100] 0 0 0 0 0 ...
##   ..$ trail                       : num [1:3100] 0.00588 0.0028 0 0.00196 0 ...
##   ..$ well_bitumen                : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ rough_pasture               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ well_aband                  : num [1:3100] 0 0 0 0 0.0322 ...
##   ..$ road_unclassified           : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ crop                        : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ low_impact_seismic          : num [1:3100] 0 0 0 0 0.0523 ...
##   ..$ clearing_unknown            : num [1:3100] 0.0923 0.0697 0 0 0 ...
##   ..$ cultivation_abandoned       : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road_paved_undiv_2l         : num [1:3100] 0 0.0174 0 0 0 ...
##   ..$ road_unimproved             : num [1:3100] 0 0 0 0 0 ...
##   ..$ truck_trail                 : num [1:3100] 0 0 0 0.0139 0 ...
##   ..$ dugout                      : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road_paved_undiv_1l         : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ well_gas                    : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ vegetated_edge_railways     : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ harvest_area_white_zone     : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ country_residence           : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ borrowpit_dry               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ rural_residence             : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ borrowpit_wet               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ borrowpits                  : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ grvl_sand_pit               : num [1:3100] 0 0.0873 0 0 0 ...
##   ..$ ris_reclaimed_temp          : num [1:3100] 0 0.0477 0 0 0 ...
##   ..$ ris_clearing_unknown        : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris_drainage                : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris_mines_oilsands          : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris_overburden_dump         : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris_facility_operations     : num [1:3100] 0 0 0 0 0 ...
##   ..$ transmission_line           : num [1:3100] 0.0642 0 0 0 0.091 ...
##   ..$ ris_tailing_pond            : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ clearing_wellpad_unconfirmed: num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ mines_oilsands              : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris_soil_replaced           : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road_paved_1l               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris_oilsands_rms            : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris_facility_unknown        : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris_borrowpits              : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris_transmission_line       : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris_soil_salvaged           : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris_road                    : num [1:3100] 0 0 0 0 0 ...
##   ..$ ris_plant                   : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ urban_residence             : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ facility_other              : num [1:3100] 0 0 0 0 0 ...
##   ..$ airp_runway                 : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ runway                      : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris_reclaimed_permanent     : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
```

```
##   ..$ urban_industrial            : num [1:3100] 0.291 0 0 0 0 ...
##   ..$ lagoon                      : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ facility_unknown            : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ residence_clearing          : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ well_cased                  : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road_unpaved_2l             : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road_paved_3l               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ surrounding_veg             : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ rlwy_sgl_track              : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road_winter                 : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ sump                        : num [1:3100] 0 0 0 0 0 ...
##   ..$ greenspace                  : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road_paved_2l               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ well_other                  : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ canal                       : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ reservoir                   : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ well_cleared_not_confirmed  : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ misc_oil_gas_facility       : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ camp_industrial             : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris_camp_industrial         : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ oil_gas_plant               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ well_unknown                : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris_utilities               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ cfo                         : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ recreation                  : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ campground                  : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ peat                        : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ golfcourse                  : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ landfill                    : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ transfer_station            : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ mill                        : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road_paved_div              : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ rlwy_spur                   : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ well_cleared_not_drilled    : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ open_pit_mine               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ well_oil                    : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ road_paved_4l               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ mines_pitlake               : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris_reclaimed_certified     : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ ris_windrow                 : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ tailing_pond                : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   .. [list output truncated]
##  $ VEG: tibble [3,100 x 14] (S3: tbl_df/tbl/data.frame)
##   ..$ array      : Factor w/ 4 levels "LU01","LU13",..: 2 2 2 2 2 2 2 2 2 2 ...
##   ..$ camera     : Factor w/ 96 levels "01","02","03",..: 32 27 3 48 67 29 35 51 65 59 ...
##   ..$ site       : Factor w/ 155 levels "LU13_18","LU13_15",..: 1 2 3 4 5 6 7 8 9 10 ...
##   ..$ buff_dist  : int [1:3100] 250 250 250 250 250 250 250 250 250 250 ...
##   ..$ lc_class20 : num [1:3100] 0.0361 0 0 0 0 ...
##   ..$ lc_class32 : num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ lc_class33 : num [1:3100] 0 0.101 0 0 0 ...
##   ..$ lc_class34 : num [1:3100] 0 0.0916 0 0 0 ...
##   ..$ lc_class50 : num [1:3100] 0.316 0 0.559 0 0 ...
##   ..$ lc_class110: num [1:3100] 0.193 0.348 0 0 0.178 ...
##   ..$ lc_class120: num [1:3100] 0 0 0 0 0 0 0 0 0 0 ...
```

```
##   ..$ lc_class210: num [1:3100] 0.456 0.358 0.186 1 0.822 ...
##   ..$ lc_class220: num [1:3100] 0 0 0 0 0 ...
##   ..$ lc_class230: num [1:3100] 0 0.101 0.255 0 0 ...
```

```
# take a look at the column names
names(covariate_data_fixed$HFI)
```

```
##   [1] "array"                          "camera"
##   [3] "site"                           "buff_dist"
##   [5] "vegetated_edge_roads"           "harvest_area"
##   [7] "road_gravel_1l"                 "conventional_seismic"
##   [9] "tame_pasture"                   "pipeline"
##  [11] "road_gravel_2l"                 "trail"
##  [13] "well_bitumen"                   "rough_pasture"
##  [15] "well_aband"                     "road_unclassified"
##  [17] "crop"                           "low_impact_seismic"
##  [19] "clearing_unknown"               "cultivation_abandoned"
##  [21] "road_paved_undiv_2l"            "road_unimproved"
##  [23] "truck_trail"                    "dugout"
##  [25] "road_paved_undiv_1l"            "well_gas"
##  [27] "vegetated_edge_railways"        "harvest_area_white_zone"
##  [29] "country_residence"              "borrowpit_dry"
##  [31] "rural_residence"                "borrowpit_wet"
##  [33] "borrowpits"                     "grvl_sand_pit"
##  [35] "ris_reclaimed_temp"             "ris_clearing_unknown"
##  [37] "ris_drainage"                   "ris_mines_oilsands"
##  [39] "ris_overburden_dump"            "ris_facility_operations"
##  [41] "transmission_line"              "ris_tailing_pond"
##  [43] "clearing_wellpad_unconfirmed"   "mines_oilsands"
##  [45] "ris_soil_replaced"              "road_paved_1l"
##  [47] "ris_oilsands_rms"               "ris_facility_unknown"
##  [49] "ris_borrowpits"                 "ris_transmission_line"
##  [51] "ris_soil_salvaged"              "ris_road"
##  [53] "ris_plant"                      "urban_residence"
##  [55] "facility_other"                 "airp_runway"
##  [57] "runway"                         "ris_reclaimed_permanent"
##  [59] "urban_industrial"               "lagoon"
##  [61] "facility_unknown"               "residence_clearing"
##  [63] "well_cased"                     "road_unpaved_2l"
##  [65] "road_paved_3l"                  "surrounding_veg"
##  [67] "rlwy_sgl_track"                 "road_winter"
##  [69] "sump"                           "greenspace"
##  [71] "road_paved_2l"                  "well_other"
##  [73] "canal"                          "reservoir"
##  [75] "well_cleared_not_confirmed"     "misc_oil_gas_facility"
##  [77] "camp_industrial"                "ris_camp_industrial"
##  [79] "oil_gas_plant"                  "well_unknown"
##  [81] "ris_utilities"                  "cfo"
##  [83] "recreation"                     "campground"
##  [85] "peat"                           "golfcourse"
##  [87] "landfill"                       "transfer_station"
##  [89] "mill"                           "road_paved_div"
##  [91] "rlwy_spur"                      "well_cleared_not_drilled"
##  [93] "open_pit_mine"                  "well_oil"
```

```
##  [95] "road_paved_4l"                "mines_pitlake"
##  [97] "ris_reclaimed_certified"      "ris_windrow"
##  [99] "tailing_pond"                 "rlwy_mlt_track"
## [101] "rlwy_dbl_track"               "ris_waste"
## [103] "interchange_ramp"             "road_paved_5l"
## [105] "ris_airp_runway"              "fruit_vegetables"
## [107] "road_unpaved_1l"              "ris_reclaim_ready"
## [109] "ris_tank_farm"
```

```
names(covariate_data_fixed$VEG)
```

```
##  [1] "array"       "camera"      "site"        "buff_dist"   "lc_class20"
##  [6] "lc_class32"  "lc_class33"  "lc_class34"  "lc_class50"  "lc_class110"
## [11] "lc_class120" "lc_class210" "lc_class220" "lc_class230"
```

Join covariate data

Now we need to join the HFI and VEG files together

```
covariates_all <- covariate_data_fixed$HFI %>%

  #use full join in case any issues with missing observations but we should be good since we checked th
  full_join(covariate_data_fixed$VEG,
            by = c('array', 'camera', 'site', 'buff_dist'))


head(covariates_all)
```

```
## # A tibble: 6 x 119
##   array camera site   buff_dist vegetated_edge_roads harvest_area road_gravel_1l
##   <fct> <fct>  <fct>      <int>                <dbl>        <dbl>          <dbl>
## 1 LU13  18     LU13_~       250               0             0                  0
## 2 LU13  15     LU13_~       250               0.0858        0                  0
## 3 LU13  03     LU13_~       250               0             0.687              0
## 4 LU13  34     LU13_~       250               0             0.337              0
## 5 LU13  57     LU13_~       250               0             0                  0
## 6 LU13  16     LU13_~       250               0             0                  0
## # i 112 more variables: conventional_seismic <dbl>, tame_pasture <dbl>,
## #   pipeline <dbl>, road_gravel_2l <dbl>, trail <dbl>, well_bitumen <dbl>,
## #   rough_pasture <dbl>, well_aband <dbl>, road_unclassified <dbl>, crop <dbl>,
## #   low_impact_seismic <dbl>, clearing_unknown <dbl>,
## #   cultivation_abandoned <dbl>, road_paved_undiv_2l <dbl>,
## #   road_unimproved <dbl>, truck_trail <dbl>, dugout <dbl>,
## #   road_paved_undiv_1l <dbl>, well_gas <dbl>, ...
```

Finish covariates data

Save data

Let's also save this for future use

```
# save joined data
write_csv(covariates_all,
          'data/processed/OSM_covariates_2022.csv')
```

We may want each buffer to have it's own column for each variable (e.g. create a wide format of this data) for modeling purposes, we can do that with the `pivot_wider()` function.

```
# we also may want to pivot wider so that each column is for a different buffer for modeling purposes,

covariates_all_wide <- covariates_all %>%

  pivot_wider(.,
              names_from = buff_dist,
              values_from = c(vegetated_edge_roads:lc_class230))

head(covariates_all_wide)
```

```
## # A tibble: 6 x 2,303
##   array camera site    vegetated_edge_roads_250 vegetated_edge_roads_500
##   <fct> <fct>  <fct>                      <dbl>                     <dbl>
## 1 LU13  18     LU13_18                  0                       0.00186
## 2 LU13  15     LU13_15                  0.0858                  0.0898
## 3 LU13  03     LU13_03                  0                       0
## 4 LU13  34     LU13_34                  0                       0
## 5 LU13  57     LU13_57                  0                       0.0443
## 6 LU13  16     LU13_16                  0                       0.00104
## # i 2,298 more variables: vegetated_edge_roads_750 <dbl>,
## #   vegetated_edge_roads_1000 <dbl>, vegetated_edge_roads_1250 <dbl>,
## #   vegetated_edge_roads_1500 <dbl>, vegetated_edge_roads_1750 <dbl>,
## #   vegetated_edge_roads_2000 <dbl>, vegetated_edge_roads_2250 <dbl>,
## #   vegetated_edge_roads_2500 <dbl>, vegetated_edge_roads_2750 <dbl>,
## #   vegetated_edge_roads_3000 <dbl>, vegetated_edge_roads_3250 <dbl>,
## #   vegetated_edge_roads_3500 <dbl>, vegetated_edge_roads_3750 <dbl>, ...
```

Let's also save this data

```
# save wide format data
write_csv(covariates_all_wide,
          'data/processed/OSM_covariates_wide_2022.csv')
```

Response metrics

there are several response metrics we can calculate, the ones we will cover here are.

1. Total independent detections per species/site
2. Presence/absence per species/site
3. Proportion of monthly detections

   Generally we only need #3 (proportional monthly detections) but we provided data for ES 482/582 class and wanted them to have multiple responses metrics to choose from for modeling purposes.

Data

For this we need the deployment and independent detection data we created earlier, if you are still working through this script its the 'deploy_fixed' & 'detections' objects

```
# deploy
deploy_fixed <- read_csv('data/processed/OSM_deployment_2022.csv')
```

```
## Rows: 155 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): array, site
## date (2): start_date, end_date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# detections
detections <- read_csv('data/processed/OSM_ind_det_2022.csv') %>%

  # change site, species and event_id to factor
  mutate_if(is.character,
            as.factor)
```

```
## Rows: 14063 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): array, site, species, event_id
## dbl  (3): month, year, timediff
## dttm (1): datetime
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

For plotting and formatting proportional monthly detections we need to create a subset of the species in the detections data to just include several focal species we are interested in

```
# create a list of focal species for filtering the data/plots
 focal_species <- c('Black bear',
                    'Caribou',
                    'Cougar',
                    'Coyote',
                    'Fisher',
                    'Grey wolf',
                    'Lynx',
                    'Moose',
                    'Red fox',
                    'White-tailed deer',
                    'Wolverine',
                    'Snowshoe hare' # this one was added for ES 482/582 class data, we don't normally u
                    )
```

1. Total independent detections

The first response metric we will calculate is the total number of independent detections per species per site.

To do this we use the detections data we created earlier from the raw Timelapse data. We need to group by site and species and then we can use the `summarise()` function with the `n()` function to count the total detections After that we ungroup the data so if we run an analysis or make a plot it doesn't stay grouped and then we use the `pivot_wider()` function to make a column for each species and a row for each site.

Finally we need to replace any NAs with zeros (the `n()` function function won't insert a zero if there aren't any observations to count so these NAs are indeed zeros)

```
total_detections <- detections %>%

  # group by site and species to count detections
  group_by(site, species) %>%

  # use summarise to count detections per species per site
  summarise(detections = n()) %>%

  ungroup() %>%

  pivot_wider(names_from = species,
              values_from = detections) %>%

  # replace NAs with 0 in all species columns
  mutate(across(
    where(is.numeric),
    ~ replace_na(., 0)))
```

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

Now that this data is formatted we should save it to the data/processed folder for use later

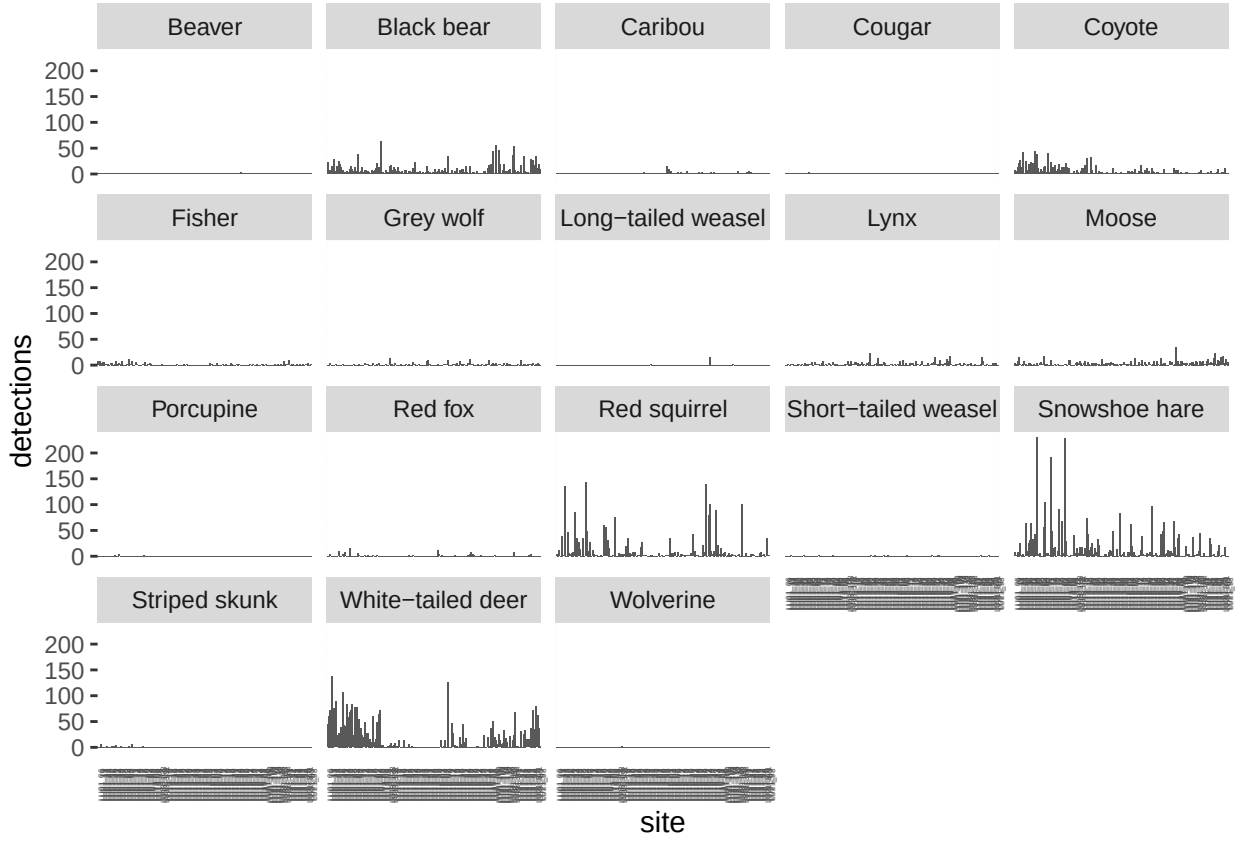
```
# save csv file to processed data folder
write_csv(total_detections,
          'data/processed/OSM_total_detections_2022.csv')
```

We can also plot this data to see what it looks like

In the code below we create an object called site_detections_plot where we pipe the total detection data into the `ggplot()` function after doing some formatting to make it plot

```
site_detections_plot <-  total_detections %>%

  # we need to pivot longer to create species column again for plotting
  pivot_longer(cols = 2:40,
               names_to = 'species',
               values_to = 'detections') %>%

  # remove less useful species using a list created in the 'Graph independent detections' section
  filter(species %in% mammals) %>%
```

```
  # pipe into ggplot function
ggplot(.,
       aes(x = site,
           y = detections)) +

  geom_col() +

  # use facet wrap to make separate plots for each species
  facet_wrap(vars(species)) +

  # shift axis text to 90 degrees so site name are readable
  theme(axis.text.x = element_text(angle = 90,
                                   size = 3),
        axis.ticks.x = element_blank())


# view plot
site_detections_plot
```



This is not the most readable plot because some species are skewing the x axis really high but it works for exploratory purposes.

If we want to save the plot for easier viewing later we can use the code below

```
# save graph as jpeg (can also save as tiff, png, pdf by changing the file extension)
ggsave('OSM_total_detections_site_2022.jpeg',
```

```
       site_detections_plot,
       path = 'figures',
       width = 12,
       height = 10,
       units = 'in',
       dpi = 600)
```

2. Presence/absences

A second response metric we may want to use is simply presence/absence data. Here we can use the total_detections data and replace any values greater than 0 with 1s to create a binary variable.

```
# we can use the data from above to create a similar response metric of simply presences and absences p

species_presence <- total_detections %>%

  # replace all values above 0 with 1s
  mutate_if(is.numeric,
            ~1 * (. > 0))

# now we have presence absence data for all species
```

We can save this to data/processed

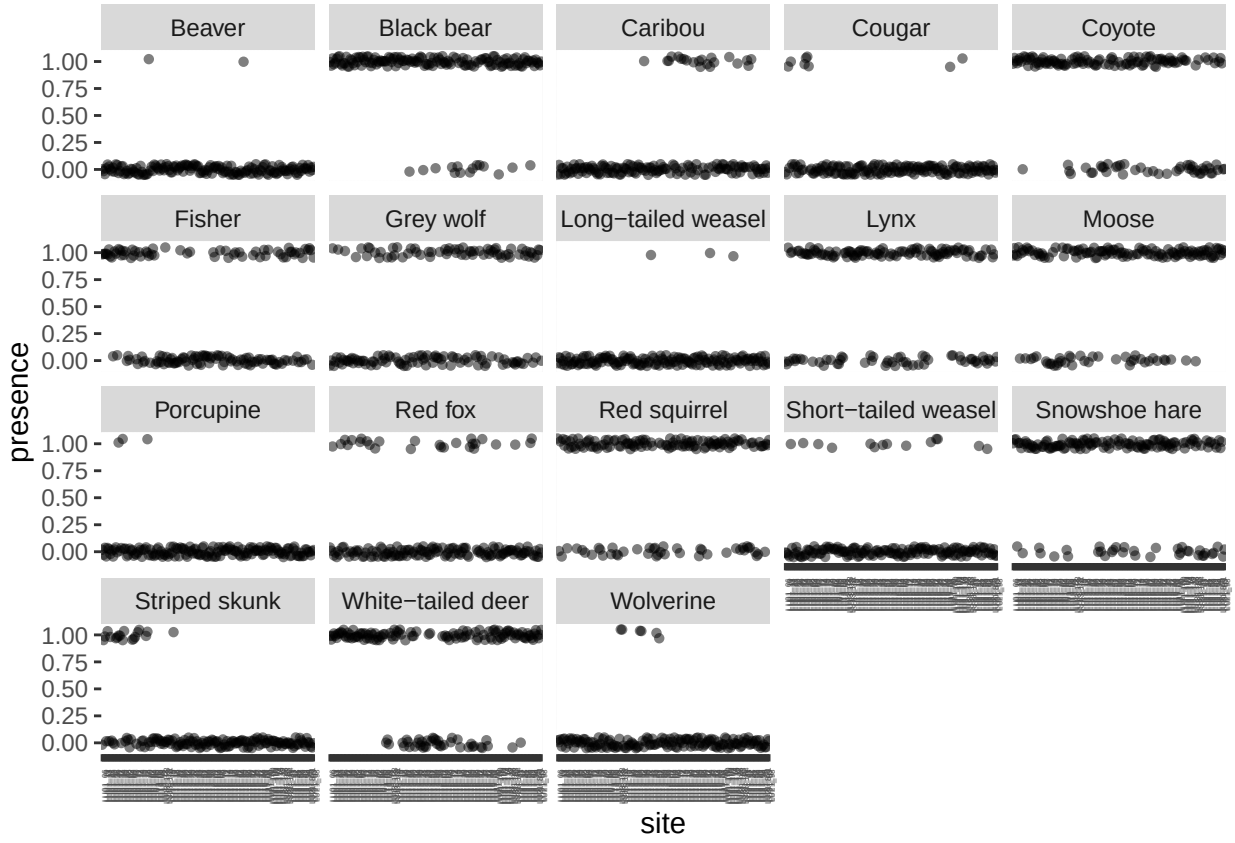
```
# save csv file to processed data folder
write_csv(species_presence,
          'data/processed/OSM_total_presence_absence_2022.csv')
```

We may also want to plot this similarly to the total detections per site to look at the data easier

As before we create a new object with our plot title name and then pipe the data after some formatting into the `ggplot()` function.

```
# plot this data in ggplot
species_presence_plot <- species_presence %>%

  # first we need to pivot the data longer again so we have a species column for plotting
  pivot_longer(cols = 2:40,
               names_to = 'species',
               values_to = 'presence') %>%

  # remove less useful species using a list created in the 'Graph independent detections' section
  filter(species %in% mammals) %>%

  # pipe into ggplot function
ggplot(.,
       aes(x = site, y = presence)) +

  # use geom_jitter instead of geom_point so we can shift points on y-axis to make them easier to view
  geom_jitter(shape = 16,
              size = 1.5,
              width = 0,
```

```
              height = 0.05,
              alpha = 0.5) +

  # use facet_wrap to make separate plots for each species so data is easier to look at
  facet_wrap(vars(species)) +

  # shift axis text to 90 degrees so site name are readable
  theme(axis.text.x = element_text(angle = 90,
                                   size = 3))



# view plot
species_presence_plot
```



If we want to save the plot for easier viewing later we can use the code below

```
# save graph as jpeg (can also save as tiff, png, pdf by changing the file extension)
ggsave('OSM_total_presence_absence_2022.jpeg',
       species_presence_plot,
       path = 'figures',
       width = 12,
       height = 10,
       units = 'in',
       dpi = 600)
```

3. Proportion monthly detections

We need to use the deployment data to determine how many days each camera was active for

The script below, modified from Becca Smith's MSc should do just that!

First we create a new data frame from the deploy_fixed (deployment data frame that's been cleaned) with some of the same columns and a new column 'day' that goes from the start date to the end date of each camera deployment and increases by intervals of 1 (for each day the camera was active). We use this to calculate the number of days/month each camera was active for and create a new variable for this called days_month. and then we create another column based on the days/month the camera was active that classifies each month of data for a camera as keep or remove based on whether there were at least 15 active camera days in that month. This is because we don't want to estimate monthly data for a camera that was working less than half the time.

```
deploy_active <- deploy_fixed %>%

  # for each row, create a sequence between the start and end dates, and make a new row for that for ea
  rowwise() %>%
  do(data.frame(array = .$array,
                site = .$site,
                start = .$start_date,
                end = .$end_date,
                day = seq(.$start_date, .$end_date, by = "1 day"))) %>%
  # Create a new column that determines which month each of your dates is in
  mutate(month = month(day),
         year = year(day)) %>%

  # group by site, month and year
  group_by(site, month, year) %>%

  # Determine number of days per month camera is active
  mutate(days_month = length(unique(day))) %>%

  # get distinct rows for each
  distinct(site, month, year,
           .keep_all = TRUE) %>%

  # mark which months have < 15 days active to be removed later
  mutate(remove = case_when(days_month <15 ~ 'remove',
                            days_month >=15 ~ 'keep'))
```

Now we calculate the total number of months each camera was active for based on the new column we created (remove) for those active at least 15 days/month.

We will use this data again later

```
# calculate the total number of months each camera was active for including only those active for >15 d

deploy_months_active <- deploy_active %>%

  # keep only months camera active >15 days
  filter(remove == 'keep') %>%

  # group by site and month
```

```
  group_by(site) %>%

  # count total number of months active
  summarise(months_active = n())

# we will use this data later
```

Now that we have identified cameras that were not active long enough each month to reliably extract data from we can use that column to remove this data from the detections data frame.

Then from the data we keep we can create a new data frame that has 1 row per camera and a column for each species indicating how many of the active months each species was detected at that camera.

```
# now that we have  identified cameras that were not active long enough each month to reliably extract

proportional_detections <- detections %>%

  # join data to the deploy_active data frame
  left_join(deploy_active,
            by = c('site',
                   'month',
                   'year')) %>%

  # filter by only those we identified as 'keep' (i.e. camera working >=15 days/month)
  filter(remove == 'keep') %>%

  # get a distinct row for each species at each site for each month and year
  distinct(site,
           species,
           month,
           year) %>%

  # group by site and species to create data frame with one row per site x species combo
  group_by(site,
           species) %>%

  summarise(months_present = n()) %>%

  # ungroup data
  ungroup() %>%

  # filter to only species in the list of focal species we created earlier
  # NOTE when we do this we lose 3 sites because there weren't any of these species detected at those s
  filter(species %in% focal_species) %>%

  # pivot the data wider so there is a column for each species and 1 row per site
  pivot_wider(names_from = species,
              values_from = months_present) %>%

  # replace NAs with zeros in all species columns
  mutate(across(
    where(is.numeric),
    ~ replace_na(., 0))) %>%
```

```
  # add column for total months each camera was active from the deploy_months_active object we created

  left_join(deploy_months_active,
            by = 'site') %>%

  # ensure all species columns are numeric not integer
  mutate_if(is.integer,
            as.numeric)
```

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

Now we can run the function below to create a second column for each species that will represent the number of absences (months the species was not detected) at each camera from the active months.

```
# run a function to create columns for absences based on presence data and how many months the camera w

# first convert data to data frame not a tibble for function to work
proportional_detections <- as.data.frame(proportional_detections)

# create a vector of the species columns for the loop
# use all species columns (this value may change year to year)
cols <- 2:13

for (col in cols) {
  if (is.numeric(proportional_detections[,col]) & is.numeric(proportional_detections[,14]))
    {new_col_name <- paste0("absent_", colnames(proportional_detections)[col])
    proportional_detections[new_col_name] <- proportional_detections[,14] - proportional_detections[,col
  }
}
```

We want to rename the species columns so they don't have spaces (r doesn't like spaces for column names but it was fine when they were entries in the data)

```
# rename columns for species because R doesn't like spaces
proportional_detections <- proportional_detections %>%

  # set the column names to lower case and replace the spaces with '_' (these are both personal prefere
  set_names(
    names(.) %>%
      tolower()%>%
      str_replace_all(pattern = ' ',
                      replacement = '_'))
```

Now we have to do a bit of final wrangling of the data to fix the bear columns because we don't want to consider the months that bears are not active in the months active columns.

```
# fix bear data

# before we can use this data we need to adjust the columns for bears since they are hibernating we don

# now let's recalculate the number of active months
```

```
months_active_bears <- deploy_active %>%

  # filter to months bears are active (April - November)
  dplyr::filter(month %in% c("4", "5", "6", "7", "8", "9", "10", "11")) %>%

  # get distinct rows for each
  distinct(site, month, year,
           .keep_all = TRUE) %>%

  # group by site
  group_by(site) %>%

  # count the number of months active during bear active season and save as new column
  summarise(months_active_bears = n())




# now we overwrite the absent column for black bears using new info

proportional_detections_bears <- proportional_detections %>%

  # join the bear active data
  left_join(months_active_bears,
            by = 'site') %>%

  # overwrite absent black bear column
  mutate(absent_black_bear = months_active_bears - black_bear) %>%

  # get rid of unnecessary columns for active months
  select(-c(months_active,
            months_active_bears))
```

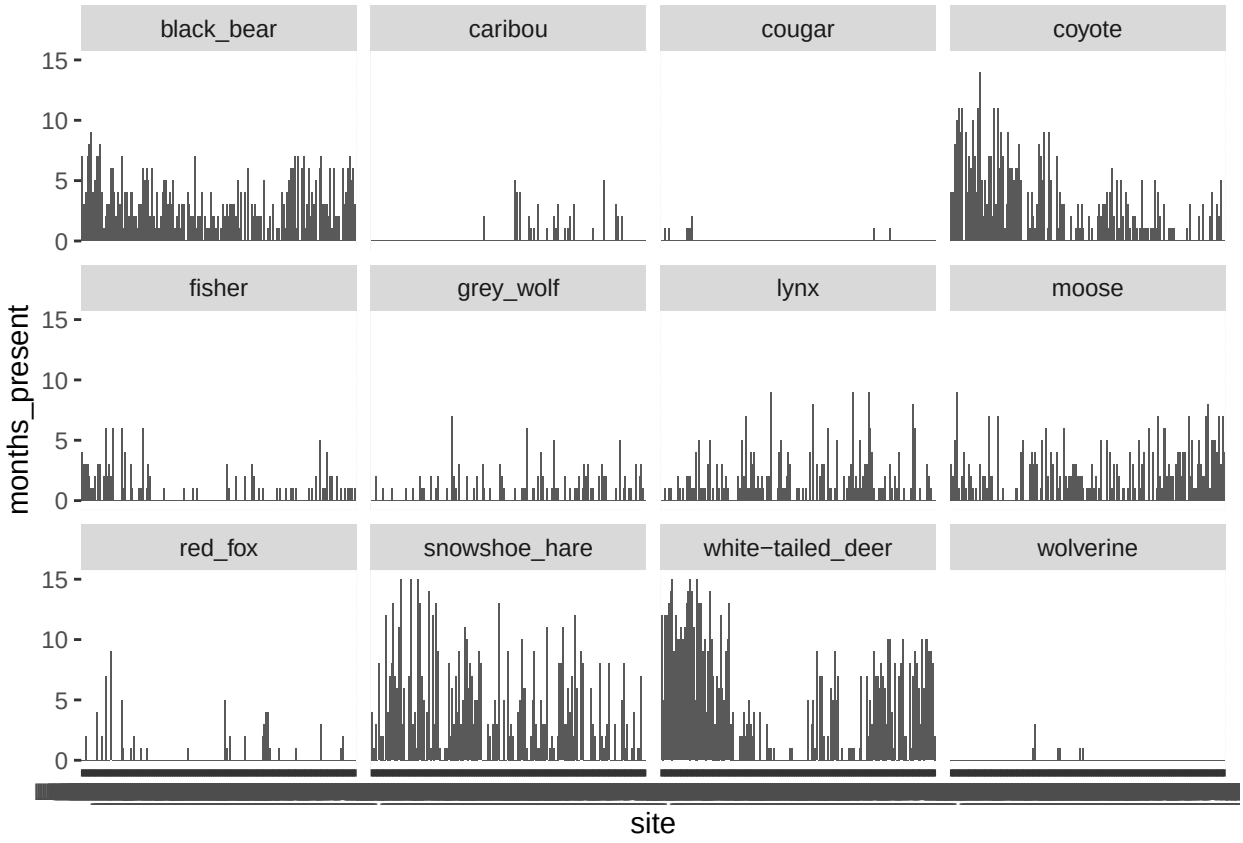
Finally we can save this data

```
# save data
write_csv(proportional_detections_bears,
          'data/processed/OSM_proportional_detections_2022.csv')
```

Let's also try to plot the presence data at least for each species so we can see which species we likely have enough data for to model

```
# first reformat the data so species presence is a column
proportional_detection_plot <- proportional_detections_bears %>%

  pivot_longer(cols = c('black_bear':'caribou'),
               names_to = 'species_presence',
               values_to = 'months_present') %>%

  select(site, species_presence, months_present) %>%

  ggplot(aes(x = site, y = months_present)) +
```

```
  # plot as bars
  geom_col() +

  # use facet wrap to generate separate plots for each species
  facet_wrap(vars(species_presence))

# view plot
proportional_detection_plot
```



If we want to save the plot for easier viewing later we can use the code below

```
# save graph as jpeg (can also save as tiff, png, pdf by changing the file extension)
ggsave('OSM_proportional_detections_2022.jpeg',
       proportional_detection_plot,
       path = 'figures',
       width = 12,
       height = 10,
       units = 'in',
       dpi = 600)
```