OSM Figures 2021-2022

Marissa Dyck

2024-10-03

Contents

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.

Before you begin

Notes

A few notes about this script.

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

(*update/add authors as needed*)

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('ggpubr')
install.packages('rphylopic')
install.packages('broom')
install.packages('grid')
devtools::install_github("JLSteenwyk/ggpubfigs")
```

Load libraries

Then load the packages to your library so they are usable for this session.

```r
library(tidyverse) # data tidying, visualization, and much more; this will load all tidyverse packages,
library(ggpubr) # make modifications to plot for publication (arrange plots)
library(rphylopic) # add animal silhouettes to graphs
library(broom) # extracting odds ratios in a tidy format
library(grid)
library(ggpubfigs)
```

Data

Before we can make any figures we need to make a data frame that include the species, top buffer width, and any other info I may want in the plot. I don't know of any easy way to extract this from all the model summaries so I'm just generating it by hand in the code chunk below.

```r
buffer_data <- tibble(

  #define number of rows
  .rows = 9,

  # create species column
  species = c('Black bear',
              'Caribou',
              'Coyote',
              'Fisher',
              'Grey wolf',
              'Lynx',
              'Moose',
              'Red fox',
              'White-tailed deer'),

  # create buffer column w/ buffer corresponding to the top model for each species
  global_buffer_width = c(250, # bear
                          1000, # caribou
                          5000, # coyote
                          1000, # fisher
                          3500, # wolf
                          250, # lynx
                          250, # moose
                          4750, # fox
                          1500), # deer

   # add phylopic uuid for each species for plotting
  image = c(get_uuid(name = 'Ursus americanus'),
            get_uuid(name = 'Rangifer tarandus'),
            get_uuid(name = 'Canis latrans'),
            get_uuid(name = 'Pekania pennanti'),
            get_uuid(name = 'Canis lupus'),
            get_uuid(name = 'Lynx lynx'),
            get_uuid(name = 'Alces alces'),
            get_uuid(name = 'Vulpes vulpes'),
            get_uuid(name ='Odocoileus virginianus'))

)
```

Buffer size

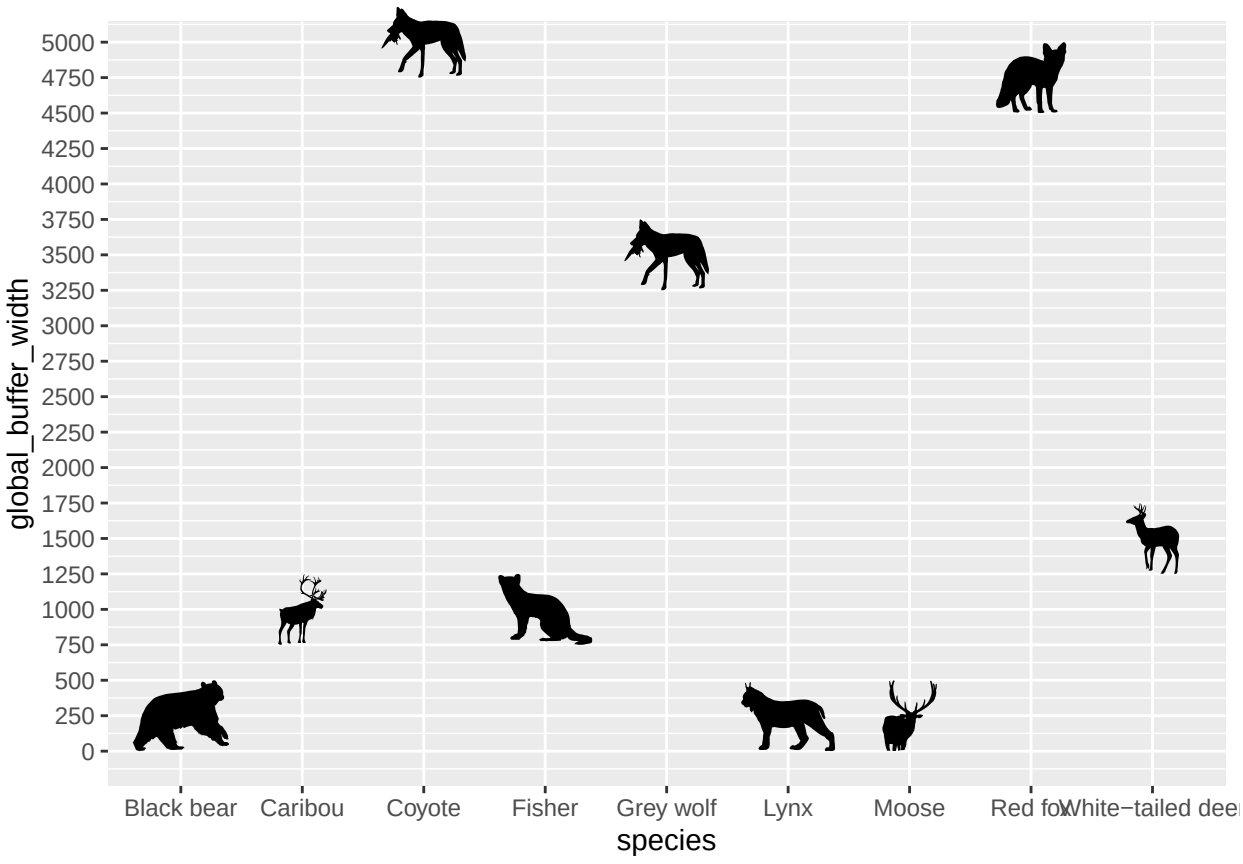
What I want to do here is create a figure that plots a point for each species at the buffer size that was in the top model. I want to use phylopic silhouettes instead of points to make the figure more fun and easier to read quickly.

Global analysis

I will plot the results for the global analysis first.

Plot

```
ggplot(buffer_data,
       aes( x = species,
            y = global_buffer_width)) +

  # add points as phylopic silhouettes
  geom_phylopic(aes(uuid = image),
                size = 500) +

  scale_y_continuous(breaks = seq(0, 5000, by = 250)) +

  coord_cartesian(ylim = c(0, 4900))
```



This looks alright but I don't love all the default phylopic images that are used.

Fix phylopics

We can go to the Phylopic website and check out other options and then use the code below to search for the uuid from those options for the ones we want to replace

Alternatively you can also use the `pick_phylopic()` function to generate a preview of the avaiable silhouettes for each species and interactively select which you want to use. However this feature doesn't work to knit in the rmarkdown html and is a bit slow so I've included the code below with eval = FALSE so it won't run and send an error when I try to knit but is there for future reference.

```
# don't try to run in r markdown, instead copy and paste into console
?pick_phylopic

pick_phylopic("Ursus americanus", view = 2)
```

I opted to use the website and manually select which silhouette I wanted by getting the uuid for a specific silhouette. I searched the website, found how many options there were, printed the uuid for all of them and then copied the one I wanted from the console into the code below to edit the data. Not the most efficient, but it works for now.

```
# Caribou could be bigger
get_uuid(name = 'Rangifer tarandus',
         n = 6)
```

```
## [1] "09c5d6bf-8ed9-4dc3-86f1-6b75dbe63952"
## [2] "a7e8f29e-a4e2-499a-b913-491f78f44f1d"
## [3] "ad1de77a-c89b-41ad-801c-1ef47c960c83"
## [4] "d2e268c1-1937-4308-9fa4-9adc55cc5e8b"
## [5] "e6e864fd-8e3d-435f-9db3-dc6869c589f1"
## [6] "19864ae2-b740-4a41-9f91-e3a195ee7cbb"
```

```
# I like the 5th one best

# Coyote doesn't need to have a food item
get_uuid(name = 'Canis latrans',
         n = 6)
```

```
## [1] "113d2520-9f92-456f-b305-52ee3986172d"
## [2] "3492f4ca-01f0-4609-a84f-084a84bf4e95"
## [3] "5a0398e3-a455-4ca6-ba86-cf3f1b25977a"
## [4] "76c8fdec-d0af-47b9-b949-dc610419d832"
## [5] "d451e353-585a-4543-84e7-7ef2f90aa407"
## [6] "da5faa63-085f-4523-a542-e71cb386c999"
```

```
# I like the 6th one best

# Grey wolf shouldn't be the same as coyote
get_uuid(name = 'Canis lupus',
         n = 5)
```

```
## [1] "113d2520-9f92-456f-b305-52ee3986172d"
## [2] "11658f8c-e0c2-4612-85ef-bdc44acdae0b"
## [3] "3492f4ca-01f0-4609-a84f-084a84bf4e95"
## [4] "5a0398e3-a455-4ca6-ba86-cf3f1b25977a"
## [5] "76c8fdec-d0af-47b9-b949-dc610419d832"
```

```
# I like 2 or 5

# Moose needs a body
get_uuid(name = 'Alces alces',
         n = 3)
```

```
## [1] "df2d0ad0-adb0-49d7-afe5-edc6cad21064"
## [2] "1a20a65d-1342-4833-a9dd-1611b9fb383c"
## [3] "74eab34a-498c-4614-aece-f02361874f79"
```

```
# 3 is best

# red fox is too big
get_uuid(name = 'Vulpes vulpes',
         n = 7)
```

```
## [1] "a1116e25-7b50-4666-bef5-de18b6e2778c"
## [2] "76352962-1eeb-4197-acdd-e3c7eeab839d"
## [3] "9b3e3567-2238-40bf-ab5d-bcb6ce7b32d5"
## [4] "a5e2a085-c895-4fdd-b39e-bbac8ca94d7d"
## [5] "d67d3bf6-3509-4ab6-819a-cd409985347e"
## [6] "40541f06-01d8-4585-a271-40268d24057d"
## [7] "40a5954e-ede1-4af6-9f65-209d1780e602"
```

```
# 4 is best

# deer is weird
get_uuid(name = 'Odocoileus virginianus',
         n = 6)
```

```
## [1] "4584be20-4514-4673-a3e8-97e2a6a10e57"
## [2] "49a5a5db-047e-4e17-849b-9f96a93f0d2b"
## [3] "56f6fdb2-15d0-43b5-b13f-714f2cb0f5d0"
## [4] "9df2ed00-b918-4d3a-82bd-1a94283f7c0f"
## [5] "6038e80c-398d-47b2-9a69-2b9edf436f64"
## [6] "8569838c-c725-4772-b0a3-b5eb04baaada"
```

```
# 5 is best
```

Let's alter the code from above to change the phylopic images

```
buffer_data <- tibble(

  #define number of rows
  .rows = 9,
```

```
# create species column
species = c('Black bear',
            'Caribou',
            'Coyote',
            'Fisher',
            'Grey wolf',
            'Lynx',
            'Moose',
            'Red fox',
            'White-tailed deer'),

# create buffer column w/ buffer corresponding to the top model for each species
global_buffer_width = c(250, # bear
                        1000, # caribou
                        5000, # coyote
                        1000, # fisher
                        3500, # wolf
                        250, # lynx
                        250, # moose
                        4750, # fox
                        1500), # deer


 # add phylopic uuid for each species for plotting
id = c(get_uuid(name = 'Ursus americanus'),
          'e6e864fd-8e3d-435f-9db3-dc6869c589f1', # caribou 5
          'e6a2fa4b-85df-43b4-989c-34a65ba7eee3', # coyote 6
          get_uuid(name = 'Pekania pennanti'),
          '76c8fdec-d0af-47b9-b949-dc610419d832', # wolf 5
          get_uuid(name = 'Lynx lynx'),
          '74eab34a-498c-4614-aece-f02361874f79', # moose 3
          'd67d3bf6-3509-4ab6-819a-cd409985347e', # red fox 4
          '6038e80c-398d-47b2-9a69-2b9edf436f64'), # white-tailed deer 6

# testing this section for some plots need the actual image not uuid
img = c(get_phylopic(uuid = get_uuid(name = 'Ursus americanus')),
        get_phylopic(uuid = 'e6e864fd-8e3d-435f-9db3-dc6869c589f1'),
        get_phylopic(uuid = 'e6a2fa4b-85df-43b4-989c-34a65ba7eee3'),
        get_phylopic(uuid = get_uuid(name = 'Pekania pennanti')),
        get_phylopic(uuid = '76c8fdec-d0af-47b9-b949-dc610419d832'),
        get_phylopic(uuid = get_uuid(name = 'Lynx lynx')),
        get_phylopic(uuid = '74eab34a-498c-4614-aece-f02361874f79'),
        get_phylopic(uuid = 'd67d3bf6-3509-4ab6-819a-cd409985347e'),
        get_phylopic(uuid = '6038e80c-398d-47b2-9a69-2b9edf436f64')))
```

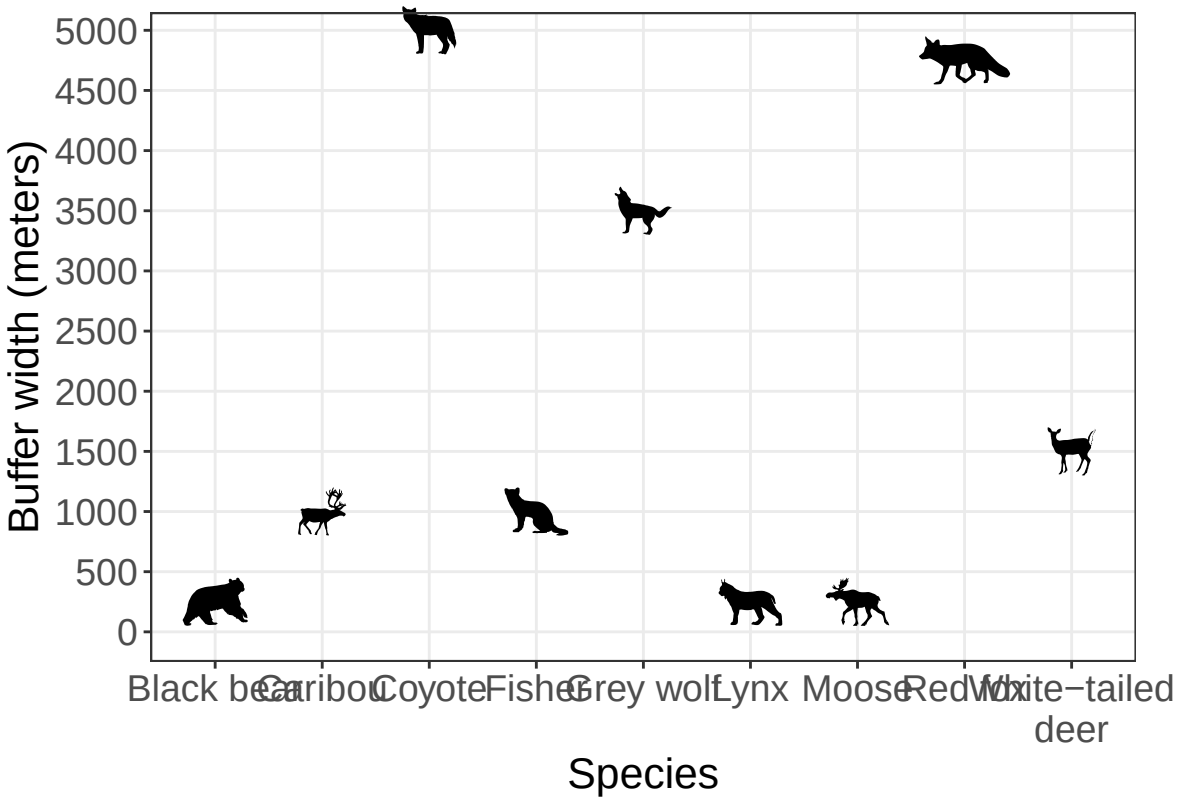
Plot again

Using the code from above with the fixed images let's plot it again

```
global_buffer_plot <- ggplot(buffer_data,
                             aes( x = species,
                                  y = global_buffer_width)) +
```

```
  # add points as phylopic silhouettes
  geom_phylopic(aes(uuid = id),
                size = 400) +

  # set axis limits and breaks
  scale_y_continuous(breaks = seq(0, 5000, by = 500)) +

  # wrap long species names
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +

  # set graph limits
  coord_cartesian(ylim = c(0,4900)) +

  # make nicer labels
  labs(y = 'Buffer width (meters)',
       x = 'Species') +

  # set theme
  theme_bw() +
  theme(panel.grid.minor = element_blank(),
        axis.text = element_text(size = 14),
        axis.title = element_text(size = 16),
        plot.margin = margin(15, 20, 15, 25))
```

```
global_buffer_plot
```



BEAUTIFUL!!! If I do say so myself

Save plot

```
ggsave('figures/OSM_species_global_buffers_2022.jpeg',
       global_buffer_plot,
       width = 18,
       height = 12,
       units = 'in',
       dpi = 600)
```

Anthropogenic analysis

Now I will generate the same plot for the anthropogenic analysis.

Edit data

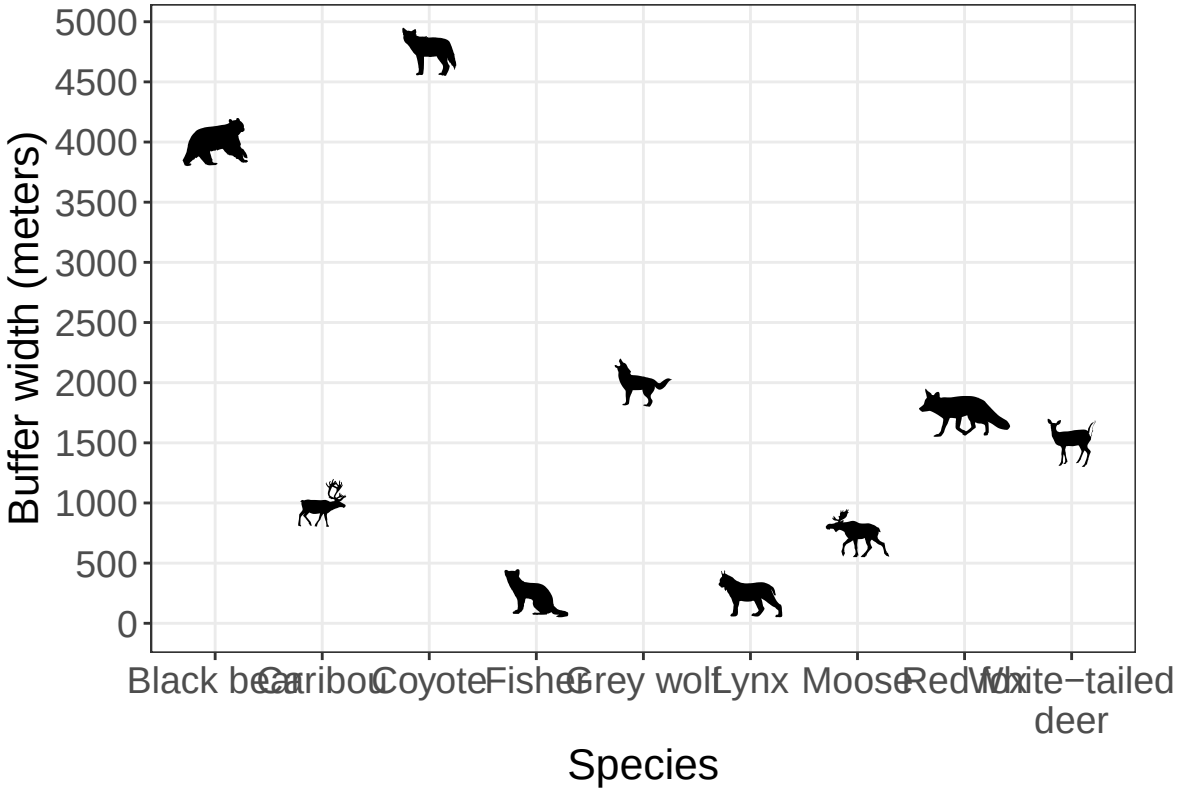
First I need to edit the buffer_data to add a column for the anthropogenic model results.

```
buffer_data <- buffer_data %>%

  # add column for top buffer size of anthropogenic models
  mutate(anthro_buffer_width = c(4000,
                                 1000,
                                 4750,
                                 250,
                                 2000,
                                 250,
                                 750,
                                 1750,
                                 1500))
```

Plot

```
anthro_buffer_plot <- ggplot(buffer_data,
                             aes(x = species,
                                 y = anthro_buffer_width)) +

  # add points as phylopic silhouettes
  geom_phylopic(aes(uuid = id),
                size = 400) +

  # set axis limits and breaks
  scale_y_continuous(breaks = seq(0, 5000, by = 500)) +

  # wrap long species names
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +
```

```
  # set graph limits
  coord_cartesian(ylim = c(0,4900)) +

  # make nicer labels
  labs(y = 'Buffer width (meters)',
       x = 'Species') +

  # set theme
  theme_bw() +
  theme(panel.grid.minor = element_blank(),
        axis.text = element_text(size = 14),
        axis.title = element_text(size = 16),
        plot.margin = margin(15, 20, 15, 25)
        )

anthro_buffer_plot
```



Save plot

```
ggsave('figures/OSM_species_anthro_buffers_2022.jpg',
       anthro_buffer_plot,
       width = 18,
       height = 12,
```

```
       units = 'in',
       dpi = 600)
```

Landscape analysis

Lastly we will make this plot again for the landscape models

Edit data frame

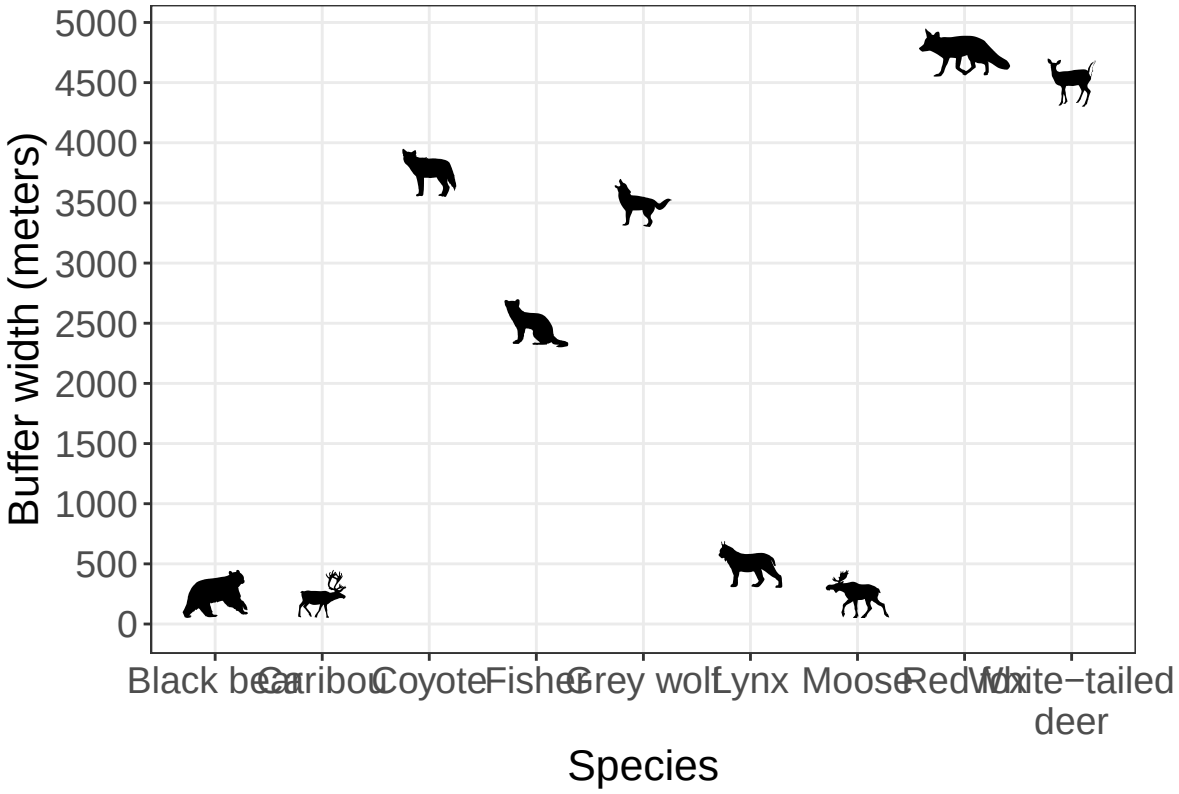
first we need to add the data to the data frame we generated earlier

```
buffer_data <- buffer_data %>%

  # add column for top buffer size of anthropogenic models
  mutate(landscape_buffer_width = c(250,
                                    250,
                                    3750,
                                    2500,
                                    3500,
                                    500,
                                    250,
                                    4750,
                                    4500))
```

Plot

```
land_buffer_plot <- ggplot(buffer_data,
                           aes(x = species,
                               y = landscape_buffer_width)) +

  # add points as phylopic silhouettes
  geom_phylopic(aes(uuid = id),
                size = 400) +

  # set axis limits and breaks
  scale_y_continuous(breaks = seq(0, 5000, by = 500)) +

  # wrap long species names
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +

  # set graph limits
  coord_cartesian(ylim = c(0,4900)) +

  # make nicer labels
  labs(y = 'Buffer width (meters)',
       x = 'Species') +

  # set theme
  theme_bw() +
  theme(panel.grid.minor = element_blank(),
```

```
        axis.text = element_text(size = 14),
        axis.title = element_text(size = 16),
        plot.margin = margin(15, 20, 15, 25)
        )

land_buffer_plot
```



Save plot

```
ggsave('figures/OSM_species_land_buffers_2022.jpg',
       land_buffer_plot,
        width = 18,
       height = 12,
       units = 'in',
       dpi = 600)
```

Body size plot

Now I want to see if their are trends between body size and which bfuffer distance performed best. I'll first need to add a body size column to the data.

Global analysis

Edit data

To do this I'm going to google the average body size of each species and then add it as a column using mutate

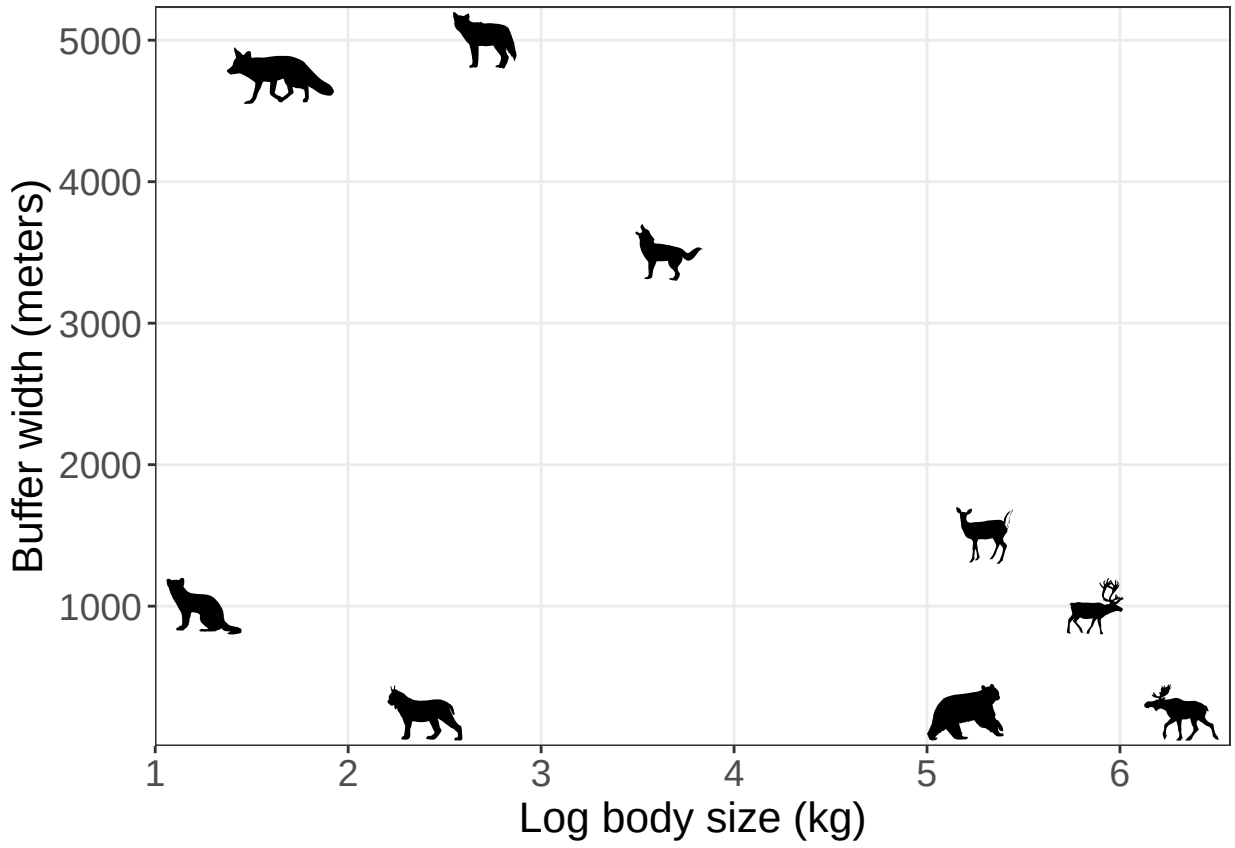
> ** IMPORTANT - this is just for data visualization purposes!. Since we don't actually have data on body size of the individuals in our study I can't run any analysis on this but I can use it to visualize if there appears to be any trends worth investigating.

```
# first add body size to buffer data
body_size_data <- buffer_data %>%

  # use mutate to create new column
  mutate(body_size = case_when(species == 'Black bear' ~ 181,
                               species == 'Caribou' ~ 355,
                               species == 'Coyote' ~ 15,
                               species == 'Fisher' ~ 3.5,
                               species == 'Grey wolf' ~ 39,
                               species == 'Lynx' ~ 11,
                               species == 'Moose' ~ 555,
                               species == 'Red fox' ~ 5.2,
                               species == 'White-tailed deer' ~ 200),
         log_body_size = log(body_size))
```

Plot

Now let's plot this as a scatterplot

```
body_size_plot <- ggplot(body_size_data, aes(x = log_body_size,
                                             y = global_buffer_width)) +

  # add points as phylopic silhouettes
  geom_phylopic(aes(uuid = id),
                size = 400) +

  labs(x = 'Log body size (kg)',
       y = 'Buffer width (meters)') +

  # set theme
  theme_bw() +
  theme(panel.grid.minor = element_blank(),
        axis.text = element_text(size = 14),
        axis.title = element_text(size = 16))

body_size_plot
```



Save plot

```
ggsave('figures/OSM_body_size_buffers_2022.jpg',
       body_size_plot,
       width = 18,
       height = 12,
       units = 'in',
       dpi = 600)
```

Final figures for manuscript

In this section I will join and tidy up some of the plots to make nice final figures for the manuscript

Figure 1

Figure 1 is a map of the study area which is not generated in this script

Figure 2

Figure 2 will depicts the top model (buffer size) for each analysis. This will either be 1 figure with with a panel for each analysis (A = global, B = Anthropogenic, C = landscape), or one figure with the points colored

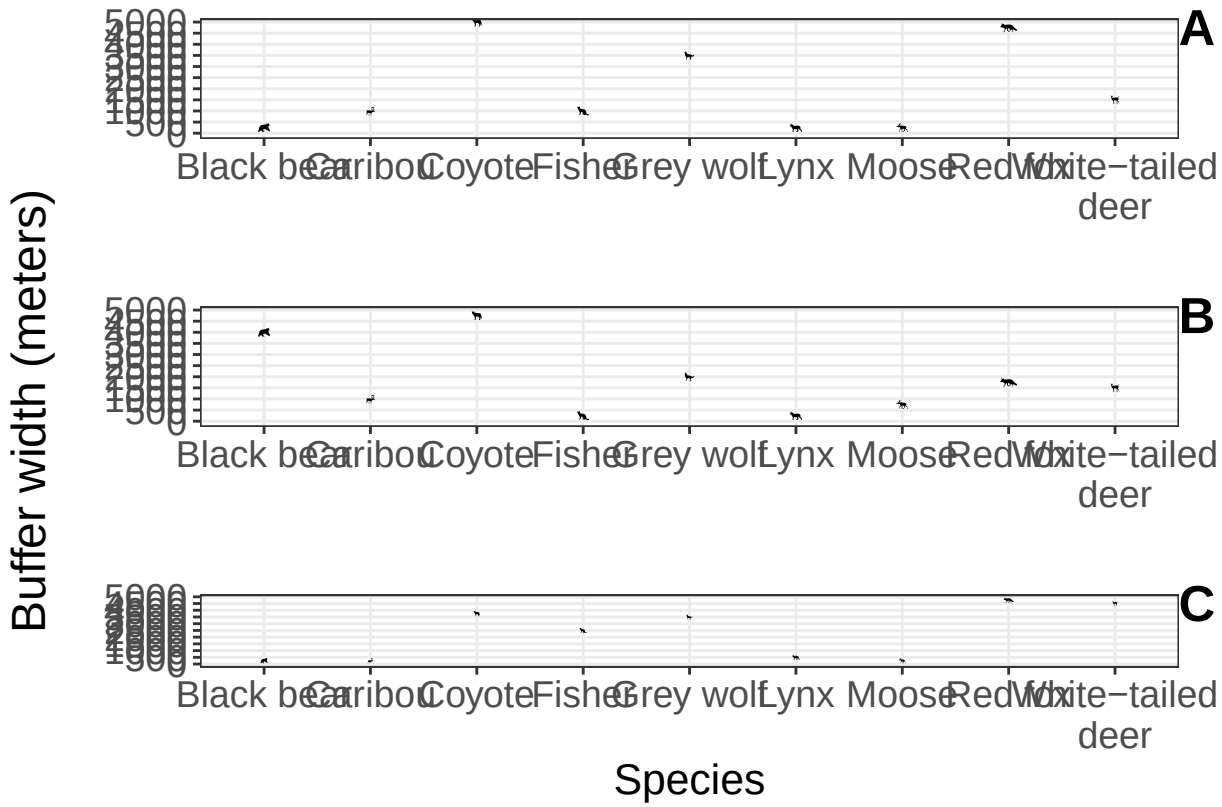
for each analysis. I will generate both here and see which displays the data/conclusions most effectively

Option A

The first option is simple because we already have the 3 plots generated from earlier we just have to combine them and add the labels for each panel.

Plot I decided I like the second option better so have not updated this plot

```
figure_2_a <-
  ggarrange(global_buffer_plot
            + rremove('ylab') +
              rremove('xlab'),
            anthro_buffer_plot
             + rremove('ylab') +
              rremove('xlab'),
            land_buffer_plot
            + rremove('ylab'),
            ncol = 1,
            labels = 'AUTO',
            label.x = 0.94,
            label.y = 0.96,
            font.label = list(size = 19),
            widths = c(1,6))  %>%

# add annotation to figure
  annotate_figure(.,
                  left = textGrob("Buffer width (meters)",
                                  rot = 90,
                                  vjust = 1,
                                  gp = gpar(cex = 1.5)))

figure_2_a
```



Export Let's export this plot to see how it looks at a proper scaled

```
ggsave('figures/publication_figures/figure_2_a.jpg',
       figure_2_a,
       height = 15,
       width = 13,
       units = 'in',
       dpi = 600)
```

That took way longer than it should have to arrange everything and adjust the margins for the common legend but it looks pretty good now.

Option B

The second option requires a bit of rearranging the data to make the plot

Data First we need to make the data long format so there is a column for the analysis type (e.g. globa, anthropogenic, landscape) which will allow us to plot these as variables

```
buffer_data_long <- buffer_data %>%

  pivot_longer(.,
               cols = c(global_buffer_width, anthro_buffer_width, landscape_buffer_width),
```

```
               names_to = 'analysis',
               values_to = 'top_buffer')
```

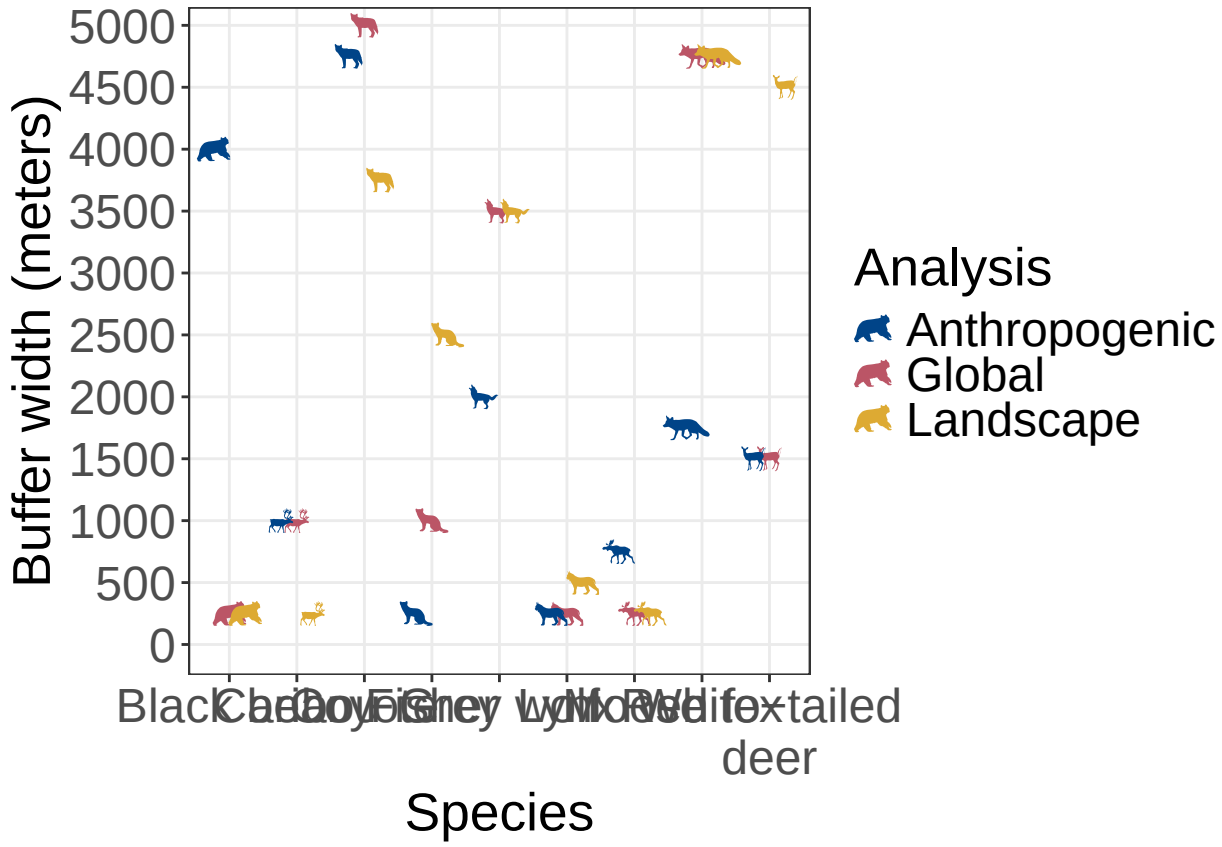
Looks good!

Plot Now let's plot this, hopefully without too much work

```
figure_2_b <-
ggplot(buffer_data_long, aes(x = species,
                        y = top_buffer)) +

  # add points as phylopic silhouettes
  geom_phylopic(aes(img = img,
                    fill = analysis),
                size = 200,
                position = position_dodge(width = 0.7),
                show.legend = TRUE,
                key_glyph = phylopic_key_glyph(img = buffer_data_long$img),
                size = 10) +

  # make the silhouettes in the legend a reasonable size, without this they plot HUGE
  guides(fill = guide_legend(override.aes = list(size = 5))) +

  scale_fill_manual('Analysis',
                    # not colorblind friendly colors
                    # values = c('black','cadetblue3', 'palegreen4'),
                    # colorblind friendly palette
                    values = friendly_pal('contrast_three'),
                    labels = c('Anthropogenic', 'Global', 'Landscape')) +
  labs(x = 'Species',
       y = 'Buffer width (meters)') +

  # set axis limits and breaks
  scale_y_continuous(breaks = seq(0, 5000, by = 500)) +

  # wrap long species names
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +

  # set graph limits
  coord_cartesian(ylim = c(0,4900)) +

  # set theme
  theme_bw() +
  theme(legend.key.size = unit(0.01, 'cm'),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 18),
        axis.title = element_text(size = 20),
        legend.title = element_text(size = 20),
        legend.text = element_text(size = 18))
```

```
## Warning: Duplicated aesthetics after name standardisation: size
```

```
figure_2_b
```




```
ggsave('figures/publication_figures/figure_2_b.jpg',
       figure_2_b,
       height = 13,
       width = 15,
       units = 'in',
       dpi = 600)
```

Save plot

Figure 3

Data

First we need to take the body size data and turn it long format like we did with the buffer data for figure 2_b

```
body_size_data_long <- body_size_data %>%

  # pivot longer
```

```
  pivot_longer(.,
               cols = c(global_buffer_width, anthro_buffer_width, landscape_buffer_width),
               names_to = 'analysis',
               values_to = 'top_buffer')

# look at data
head(body_size_data_long)
```

```
## # A tibble: 6 x 7
##   species    id            img       body_size log_body_size analysis top_buffer
##   <chr>      <chr>         <list>        <dbl>         <dbl> <chr>         <dbl>
## 1 Black bear 369a7880-479~ <Picture>       181          5.20 global_~        250
## 2 Black bear 369a7880-479~ <Picture>       181          5.20 anthro_~       4000
## 3 Black bear 369a7880-479~ <Picture>       181          5.20 landsca~        250
## 4 Caribou    e6e864fd-8e3~ <Picture>       355          5.87 global_~       1000
## 5 Caribou    e6e864fd-8e3~ <Picture>       355          5.87 anthro_~       1000
## 6 Caribou    e6e864fd-8e3~ <Picture>       355          5.87 landsca~        250
```

Plot

This will be similar to the plot above bud on the x axis will be log body size instead of species.

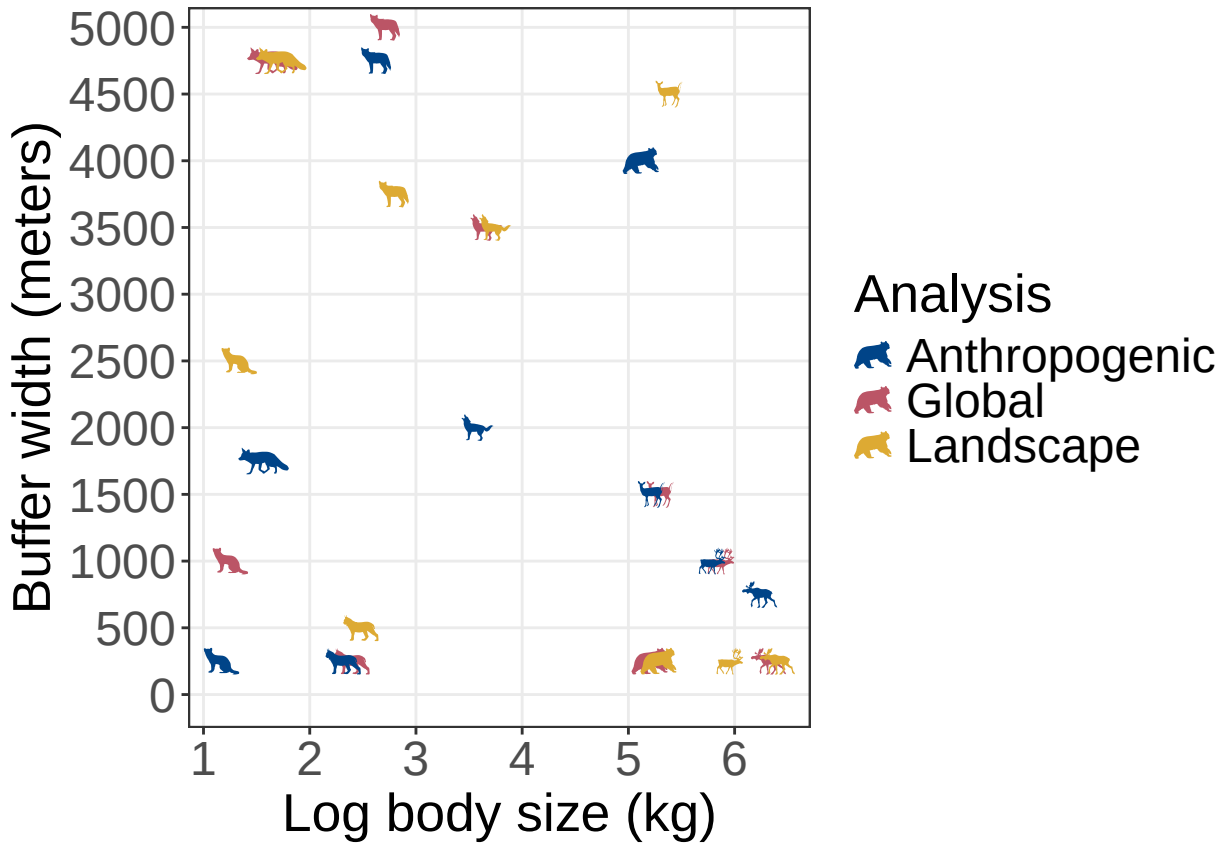
```
figure_3 <-
ggplot(body_size_data_long, aes(x = log_body_size,
                        y = top_buffer)) +

  # add points as phylopic silhouettes
  geom_phylopic(aes(img = img,
                    fill = analysis),
                size = 200,
                position = position_dodge(width = 0.25),
                show.legend = TRUE,
                key_glyph = phylopic_key_glyph(img = buffer_data_long$img),
                size = 10) +

  # make the silhouettes in the legend a reasonable size, without this they plot HUGE
  guides(fill = guide_legend(override.aes = list(size = 5))) +

  scale_fill_manual('Analysis',
                    # not colorblind friendly colors
                    # values = c('black','cadetblue3', 'palegreen4'),
                    # colorblind friendly palette
                    values = friendly_pal('contrast_three'),
                    labels = c('Anthropogenic', 'Global', 'Landscape')) +
  labs(x = 'Log body size (kg)',
       y = 'Buffer width (meters)') +

  # set axis limits and breaks
  scale_y_continuous(breaks = seq(0, 5000, by = 500)) +
  scale_x_continuous(breaks = seq(0, 7, by = 1)) +

  # set graph limits
```

```
  coord_cartesian(ylim = c(0,4900)) +

  # set theme
  theme_bw() +
  theme(legend.key.size = unit(0.01, 'cm'),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 18),
        axis.title = element_text(size = 20),
        legend.title = element_text(size = 20),
        legend.text = element_text(size = 18))
```

```
## Warning: Duplicated aesthetics after name standardisation: size
```

```
figure_3
```

```
## Warning: `position_dodge()` requires non-overlapping x intervals.
```



Save

```
ggsave('figures/publication_figures/figure_3.jpg',
       figure_3,
       height = 13,
```

```
       width = 15,
       units = 'in',
       dpi = 600)
```

```
## Warning: `position_dodge()` requires non-overlapping x intervals.
```

Supporting information figures

For the supporting information, we want a figure that depicts the model selection results from each analysis so we can see if there is a range of similar values that encompass most of the weight for each species

To do this we need a data frame with the results from the model selection for each analysis and each species. I generated exactly this file in the last script 3_OSM_glm_analysis_2022.rmd and saved it to the data processed folder so we can read it in now

Data

There is a separate csv file for each analysis but they are all stored in the data processed folder so we can upload all 3 at the same time as a list element to work with later

```
osm_model_sel_data <- file.path('data/processed/',

                                # provide list of files
                                c('osm_glm_global_model_sel_data.csv',
                                  'osm_glm_anthro_model_sel_data.csv',
                                  'osm_glm_land_model_sel_data.csv')) %>%

  # use purrr::map to read them all in
  map(~.x %>%
        read_csv(.,

                 col_types = cols(Model = col_factor(),
                                  species = col_factor())) %>%

        # remove some text from model names
        mutate(Model = str_replace(Model, " meter buffer", ""),
               Model = as.numeric(Model))) %>%

  # name each list element according to the analysis
  purrr::set_names('global',
                   'anthro',
                   'land')
```

Let's take a look at one of the data files to make sure it's all set to work with.

The main columns we carea bout are the model name, weight, and species

```
str(osm_model_sel_data$global)
```

```
## tibble [180 x 20] (S3: tbl_df/tbl/data.frame)
##  $ Model                                 : num [1:180] 250 3750 4000 3500 3250 4250 2750 3000 2500
```

```
##  $ cond((Int))                              : num [1:180] -0.598 -0.599 -0.599 -0.598 -0.598 ...
##  $ disp((Int))                              : chr [1:180] "+" "+" "+" "+" ...
##  $ cond(scale(harvest))                     : num [1:180] 0.00799 0.06374 0.06691 0.05559 0.05768 ...
##  $ cond(scale(lc_developed))                : num [1:180] -0.188 -0.122 -0.117 -0.12 -0.118 ...
##  $ cond(scale(lc_forest))                   : num [1:180] -0.11733 -0.01858 -0.01916 -0.00456 0.00532
##  $ cond(scale(lc_grassland))                : num [1:180] -0.00163 -0.03364 -0.02177 -0.02707 -0.0278
##  $ cond(scale(osm_industrial))              : num [1:180] 0.033 -0.1106 -0.1109 -0.0935 -0.081 ...
##  $ cond(scale(pipeline_transmission_lines)): num [1:180] -0.0725 -0.1156 -0.1273 -0.1251 -0.134 ...
##  $ cond(scale(seismic_lines_3D))            : num [1:180] -0.117 -0.122 -0.116 -0.114 -0.11 ...
##  $ cond(scale(seismic_lines))               : num [1:180] -0.0178 -0.1212 -0.1274 -0.1058 -0.1 ...
##  $ cond(scale(trails))                      : num [1:180] 0.1265 0.1038 0.1041 0.0967 0.0901 ...
##  $ cond(scale(wells))                       : num [1:180] -0.0112 0.2851 0.2882 0.2618 0.2616 ...
##  $ df                                       : num [1:180] 12 12 12 12 12 12 12 12 12 12 ...
##  $ logLik                                   : num [1:180] -449 -452 -452 -452 -452 ...
##  $ AICc                                     : num [1:180] 923 928 929 930 930 ...
##  $ delta                                    : num [1:180] 0 5.15 5.31 6.36 6.55 ...
##  $ weight                                   : num [1:180] 0.6781 0.0517 0.0477 0.0282 0.0257 ...
##  $ Dataset                                  : chr [1:180] "osm_final_df_2021_2022" "osm_final_df_2021
##  $ species                                  : Factor w/ 9 levels "bear","caribou",..: 1 1 1 1 1 1 1 1
```

Looks good enough for plotting

Species labels

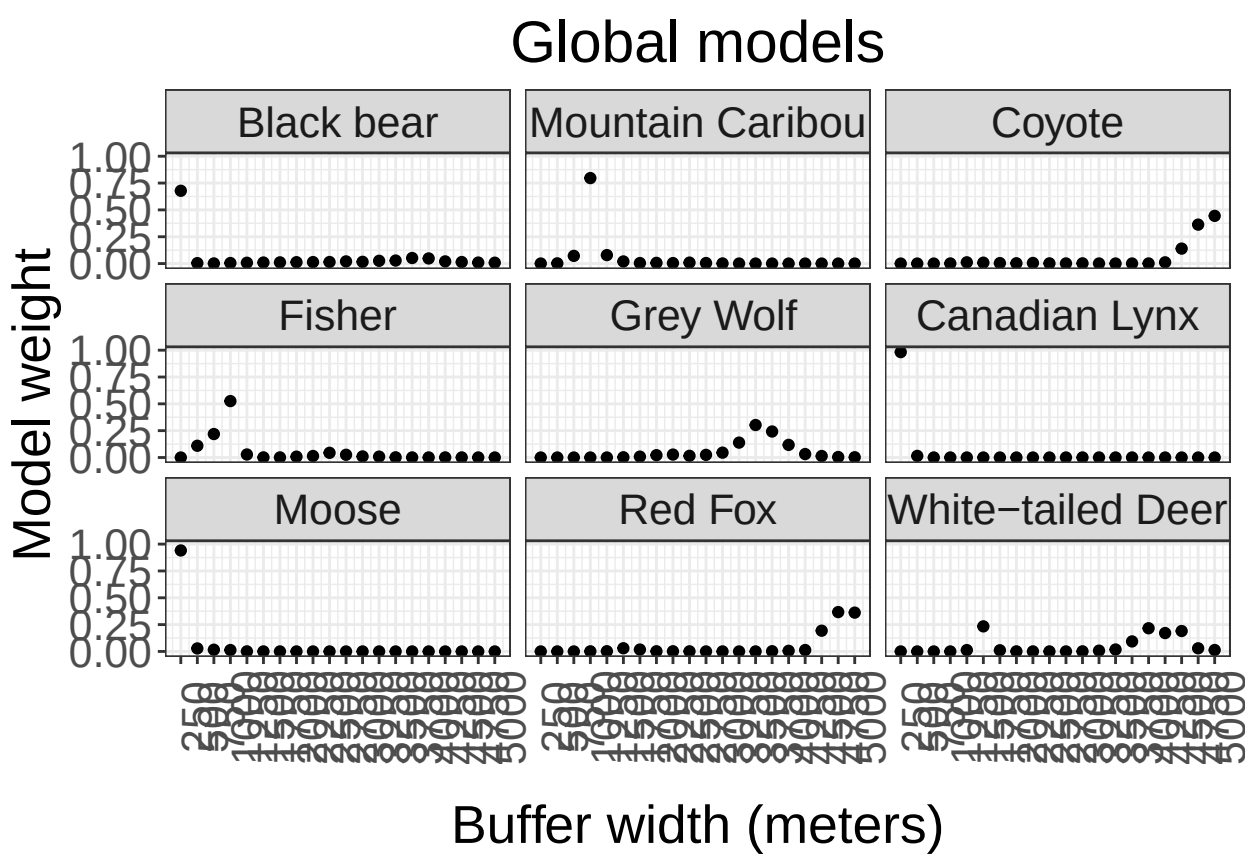
For these plots I want to customize the species labels so they are cleaner, here I am generating a vector of the names I will use in the following 3 plots

```
species_labels <- c(
  bear = 'Black bear',
  caribou = 'Mountain Caribou',
  coyote = 'Coyote',
  fisher = 'Fisher',
  wolf = ' Grey Wolf',
  lynx = 'Canadian Lynx',
  moose = 'Moose',
  fox = 'Red Fox',
  deer = 'White-tailed Deer')
```

Figure S1

```
figs1 <-

  ggplot(osm_model_sel_data$global,
         aes(x = Model,
             y = weight)) +

  # add points for each buffer size
  geom_point() +

  # make indv plots for each species
  facet_wrap(~species,
```

```
            labeller = as_labeller(species_labels)) +

  # x axis breaks
  scale_x_continuous(breaks = seq(250, 5000, by= 250)) +

  # format axis labels
  labs(x = 'Buffer width (meters)',
       y = 'Model weight',
       title = 'Global models') +

  # format theme elements of graph
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5,
                                  size = 22),
        axis.text.x = element_text(angle = 90),
        axis.title.x = element_text(margin = margin(t = 15)),
        axis.text = element_text(size = 16),
        axis.title = element_text(size = 20),
        strip.text = element_text(size = 16))  # Adjust the text size of facet labels

figs1
```



Now lets save this plot

```
ggsave('figures/publication_figures/figure_S1.jpg',
       figs1,
```

```
       height = 12,
       width = 15,
       units = 'in',
       dpi = 600
       )
```

Figure S2

Now we repeat for the anthropogenic models

```
figs2 <-

  ggplot(osm_model_sel_data$anthro,
         aes(x = Model,
             y = weight)) +

  # add points for each buffer size
  geom_point() +

  # make indv plots for each species
  facet_wrap(~species,
             labeller = as_labeller(species_labels)) +

  # x axis breaks
  scale_x_continuous(breaks = seq(250, 5000, by= 250)) +

  # format axis labels
  labs(x = 'Buffer width (meters)',
       y = 'Model weight',
       title = 'Anthropogenic models') +

  # format theme elements of graph
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5,
                                  size = 22),
        axis.text.x = element_text(angle = 90),
        axis.title.x = element_text(margin = margin(t = 15)),
        axis.text = element_text(size = 16),
        axis.title = element_text(size = 20),
        strip.text = element_text(size = 16))  # Adjust the text size of facet labels

figs2
```

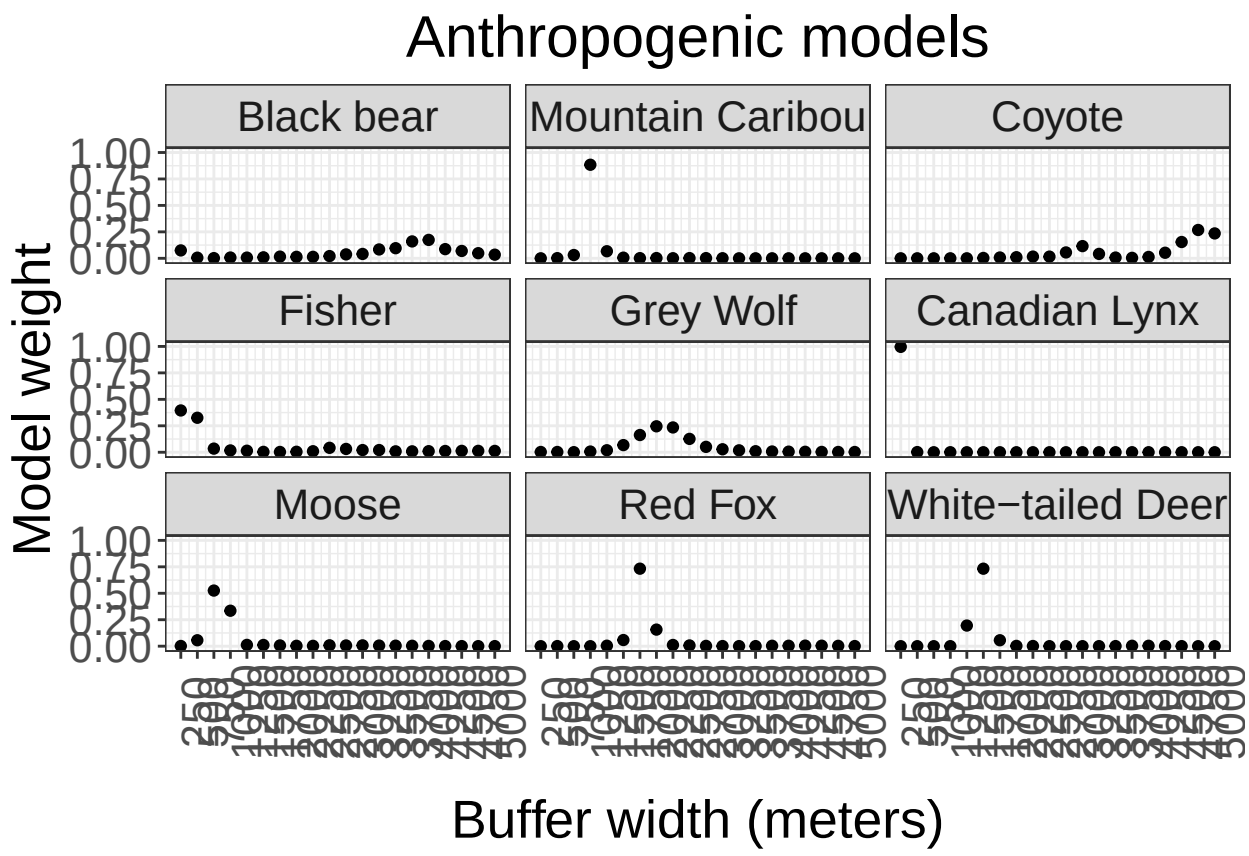
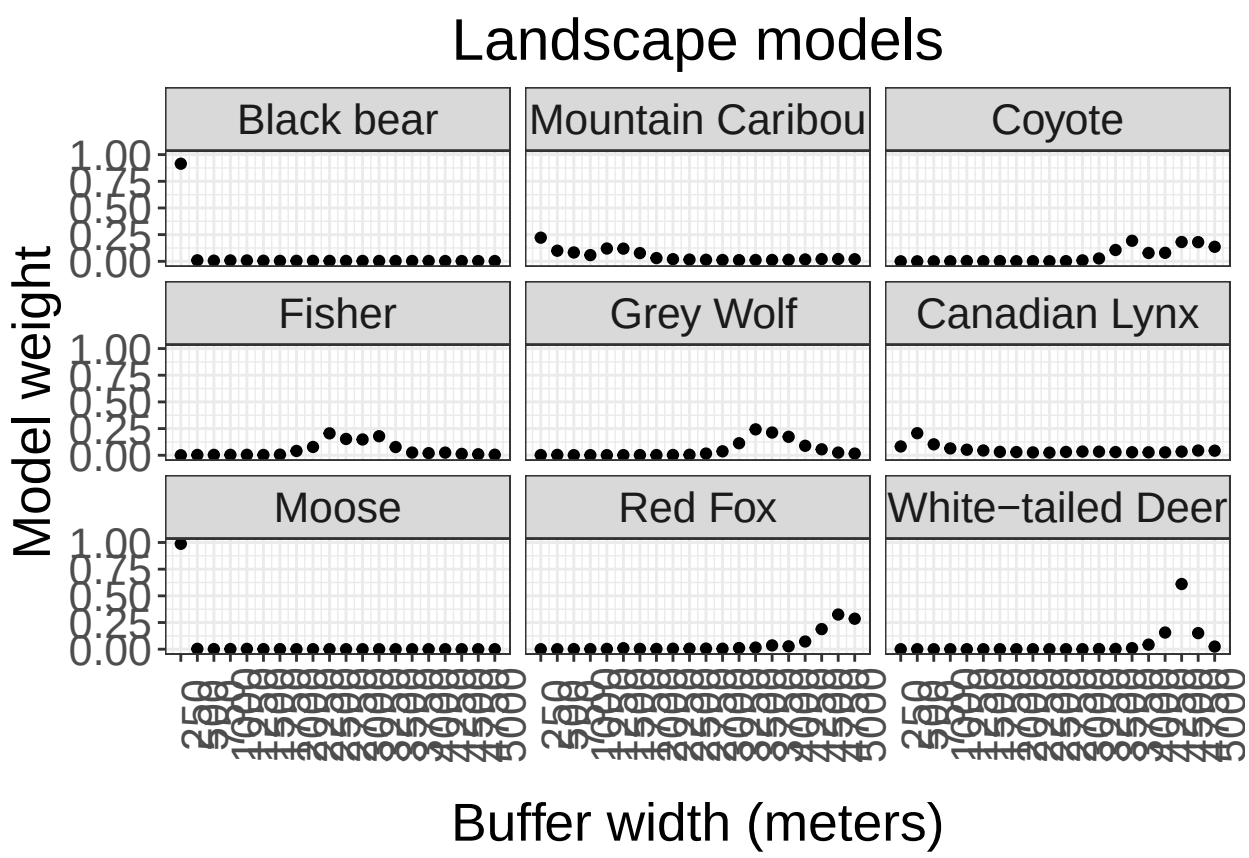


Figure S3

And repeat again for the landscape models. We could probably have found a way to do this in purrr but oh well.

```
figs3 <-

  ggplot(osm_model_sel_data$land,
         aes(x = Model,
             y = weight)) +

  # add points for each buffer size
  geom_point() +

  # make indv plots for each species
  facet_wrap(~species,
             labeller = as_labeller(species_labels)) +

  # x axis breaks
  scale_x_continuous(breaks = seq(250, 5000, by= 250)) +

  # format axis labels
  labs(x = 'Buffer width (meters)',
       y = 'Model weight',
       title = 'Landscape models') +
```

```
  # format theme elements of graph
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5,
                                  size = 22),
        axis.text.x = element_text(angle = 90),
        axis.title.x = element_text(margin = margin(t = 15)),
        axis.text = element_text(size = 16),
        axis.title = element_text(size = 20),
        strip.text = element_text(size = 16))  # Adjust the text size of facet labels

figs3
```



Now lets save this plot

```
ggsave('figures/publication_figures/figure_S3.jpg',
       figs3,
       height = 12,
       width = 15,
       units = 'in',
       dpi = 600
       )
```