

Comparing buffalo and simulated data using summary statistics

Scott Forrest

2024-03-04

Load packages

```
options(scipen=999)

library(tidyverse)
packages <- c("ggh4x", "patchwork", "terra", "ggExtra",
            "cowplot", "ggpubr", "viridis", "scales")
walk(packages, require, character.only = T)
```

Import environmental layers

```
# unscaled rasters
ndvi_stack <- rast("mapping/cropped rasters/ndvi_GEE_projected_watermask20230207.tif")
ndvi_2018_dry <- ndvi_stack[8:10]
ndvi_2019_dry <- ndvi_stack[[19:21]]
ndvi_dry <- terra::mean(c(ndvi_2018_dry, ndvi_2019_dry))
names(ndvi_dry) <- "ndvi_dry"
# plot(ndvi_dry)

canopy <- rast("mapping/cropped rasters/canopy_cover.tif")
herby <- rast("mapping/cropped rasters/veg_herby.tif")
slope <- rast("mapping/cropped rasters/slope_raster.tif")
```

Hourly summary statistics

Import the summary statistic data frames from the observed and simulated datasets that were outputted by the trajectory validation scripts, and combine into a single data frame for plotting

```
# read in the observed data
summaries_hourly_buffalo <-
  read_csv("outputs/buffalo_summaries_hourly_habitat_2024-02-07.csv") %>%
  mutate(id = as.factor(id))

# read in the simulated data
summaries_hourly_0p <-
  read_csv("outputs/sim_0p_memALL_summaries_hourly_habitat_2024-02-07.csv") %>%
  mutate(id = as.factor(id))
summaries_hourly_1p <-
  read_csv("outputs/sim_1p_memALL_summaries_hourly_habitat_2024-02-07.csv") %>%
  mutate(id = as.factor(id))
```

```

summaries_hourly_2p <-
  read_csv("outputs/sim_2p_memALL_summaries_hourly_habitat_2024-02-07.csv") %>%
  mutate(id = as.factor(id))
summaries_hourly_3p <-
  read_csv("outputs/sim_3p_memALL_summaries_hourly_habitat_2024-02-07.csv") %>%
  mutate(id = as.factor(id))

summaries_hourly_all <- bind_rows(summaries_hourly_buffalo,
                                    summaries_hourly_0p,
                                    summaries_hourly_1p,
                                    summaries_hourly_2p,
                                    summaries_hourly_3p)

summaries_hourly_all_long <- summaries_hourly_all %>%
  dplyr::select(!...1) %>%
  pivot_longer(cols = !c(data, hour, id),
               names_to = "variable",
               values_to = "value") %>%
  mutate(Data = factor(str_to_title(data),
                       levels = c("Buffalo", "0p", "1p", "2p", "3p")))

head(summaries_hourly_all)

## # A tibble: 6 x 23
##   ...1 data     hour id      n step_length_mean step_length_median step_length_sd ndvi_mean ndvi_sd
##   <dbl> <chr>    <dbl> <fct> <dbl>           <dbl>           <dbl>           <dbl>           <dbl>
## 1     1 buffalo    1 2005    99       63.7          7.61        117.         0.281
## 2     2 buffalo    1 2014    75      149.          9.82        311.         0.267
## 3     3 buffalo    1 2018    96      206.          15.6        413.         0.263
## 4     4 buffalo    1 2021    97      129.          84.0        214.         0.219
## 5     5 buffalo    1 2022    86      106.          18.8        169.         0.293
## 6     6 buffalo    1 2024    87      265.          71.4        567.         0.297
## # i 12 more variables: herby_mean <dbl>, herby_median <dbl>, herby_sd <dbl>, canopy_mean <dbl>, canopy_sd <dbl>, slope_mean <dbl>, slope_median <dbl>, slope_sd <dbl>, memory_mean <dbl>, memory_sd <dbl>
head(summaries_hourly_all_long)

## # A tibble: 6 x 6
##   data     hour id   variable      value Data
##   <chr>    <dbl> <fct> <chr>     <dbl> <fct>
## 1 buffalo    1 2005 n            99    Buffalo
## 2 buffalo    1 2005 step_length_mean 63.7  Buffalo
## 3 buffalo    1 2005 step_length_median 7.61 Buffalo
## 4 buffalo    1 2005 step_length_sd    117. Buffalo
## 5 buffalo    1 2005 ndvi_mean      0.281 Buffalo
## 6 buffalo    1 2005 ndvi_median    0.268 Buffalo

```

Plot the hourly habitat selection between the observed and simulated data from different models

In these plots we are assessing how well the simulated trajectories capture the movement dynamics and habitat use of the observed data.

There's a few different ways we can plot this, so we'll provide some examples here.

With each hour represented as a boxplot

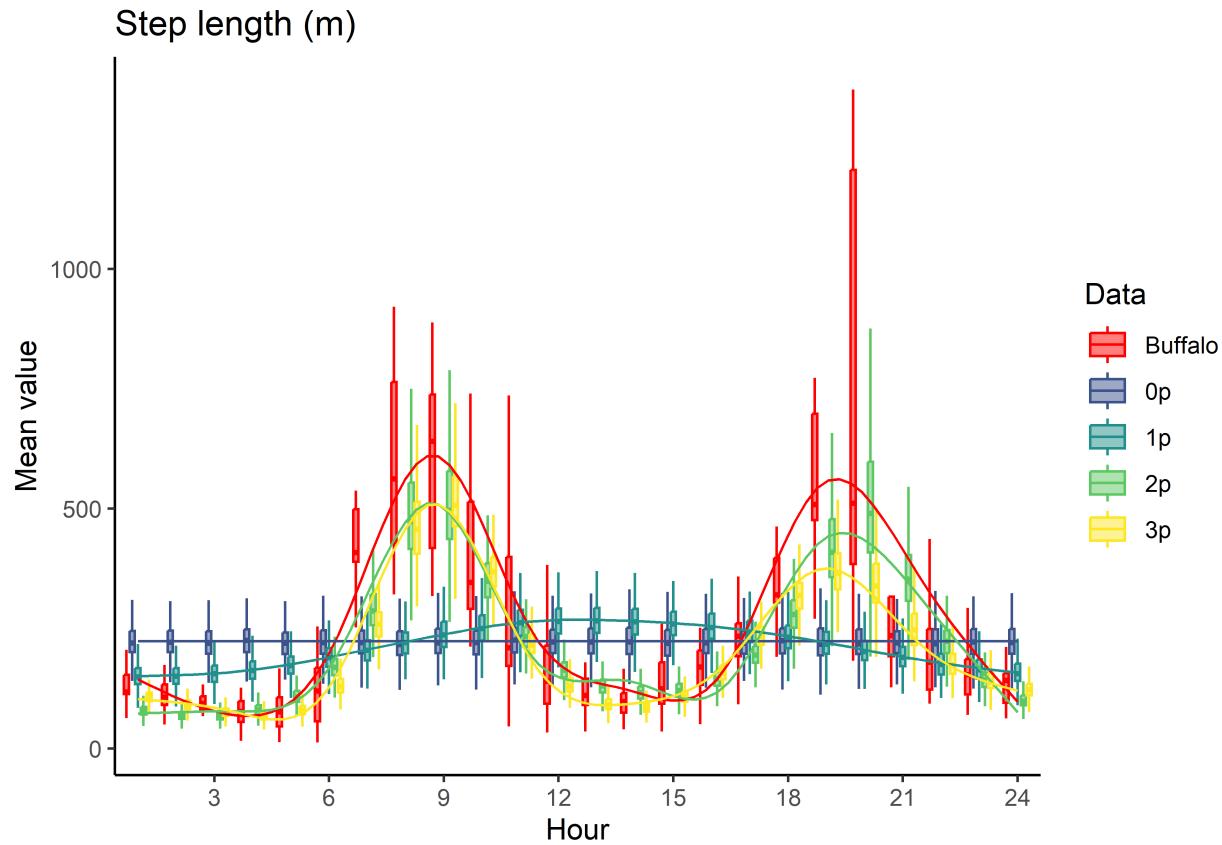
Here, each ‘point’ in a boxplot represents the mean value of step length for that hour for one of the trajectories (observed or simulated). Therefore, the boxplot represents the distribution of these means across all trajectories for each of the models or the observed data. We also include a smoothed line to show the trend.

The boxplots get hard to interpret with the habitat covariates.

```
# Create color mapping
unique_groups <- unique(summaries_hourly_all_long$Data)
colors <- viridis(length(unique_groups))
names(colors) <- unique_groups
colors["Buffalo"] <- "red"

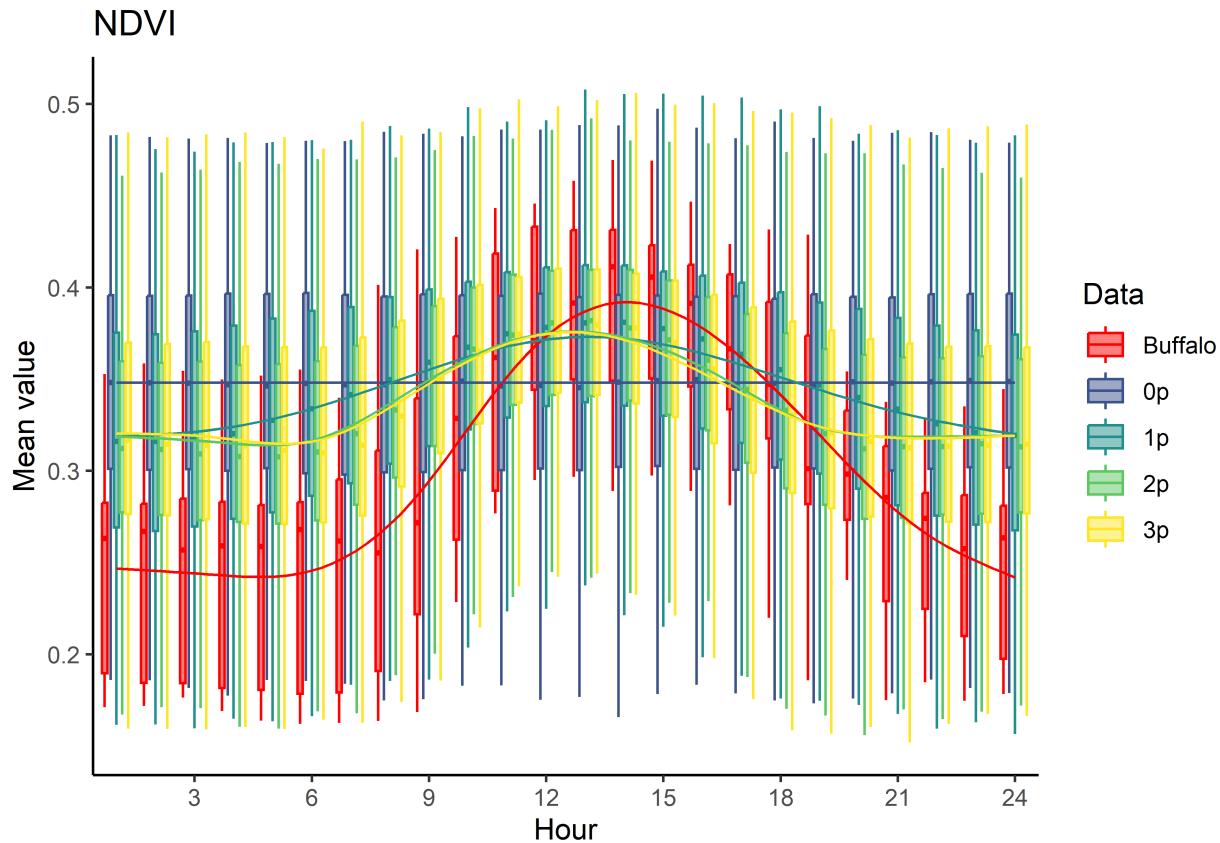
# Step length
ggplot() +
  geom_boxplot(data = summaries_hourly_all_long %>% filter(variable == "step_length_mean"),
               aes(x = factor(hour), y = value, colour = Data, fill = Data),
               alpha = 0.5,
               outlier.shape = NA) +
  geom_smooth(data = summaries_hourly_all_long %>% filter(variable == "step_length_mean"),
              aes(x = hour, y = value, colour = Data, fill = Data),
              size = 0.5,
              se = FALSE,
              alpha = 0.75) +
  scale_fill_manual(values = colors) +
  scale_colour_manual(values = colors) +
  scale_x_discrete("Hour", breaks = seq(0,24,3)) +
  scale_y_continuous("Mean value") +
  ggtitle("Step length (m)") +
  theme_classic()

## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



```
# NDVI
ggplot() +
  geom_boxplot(data = summaries_hourly_all_long %>% filter(variable == "ndvi_mean"),
               aes(x = factor(hour), y = value, colour = Data, fill = Data),
               alpha = 0.5,
               outlier.shape = NA) +
  geom_smooth(data = summaries_hourly_all_long %>% filter(variable == "ndvi_mean"),
              aes(x = hour, y = value, colour = Data, fill = Data),
              size = 0.5,
              se = FALSE,
              alpha = 0.75) +
  scale_fill_manual(values = colors) +
  scale_colour_manual(values = colors) +
  scale_x_discrete("Hour", breaks = seq(0,24,3)) +
  scale_y_continuous("Mean value") +
  ggtitle("NDVI") +
  theme_classic()

## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



With a path approach

Here we show a single ‘path’ per observed or simulated individual, which is comprised of the mean step length (or other covariate) for each hour. We also include a smoothed line to show the trend. The 13 observed buffalo are shown in red, and the simulated trajectories are coloured based on the model that was fitted (0p, 1p, 2p or 3p to denote 0 - 3 pairs of harmonics).

Again, these are a bit messy, but we can see every trajectory. It’s clear that there is a lot of variability in both the observed and simulated data, which is likely due to where in the landscape the observed or simulated buffalo were (and what habitat is ‘available’ to them). We can see that the dynamic models (2p and 3p for the step length, and 1p, 2p and 3p for NDVI) capture the dynamic trends in movement behaviour and habitat selection.

```
# subset the number of simulations
#(650 per model in our case = 13 buffalo x 50 simulations)
summaries_hourly_all_long_subset <- summaries_hourly_all_long %>%
  dplyr::group_by(data, hour, variable) %>% slice_head(n = 650)

summaries_hourly_all_long_subset %>% dplyr::group_by(data, id)

## # A tibble: 1,191,528 x 6
## # Groups:   data, id [2,613]
##   data    hour   id     variable     value Data
##   <chr>  <dbl> <fct>    <chr>      <dbl> <fct>
## 1 0p       1 2005_10_1 canopy_mean 0.570 0p
## 2 0p       1 2005_10_2 canopy_mean 0.537 0p
## 3 0p       1 2005_11_2 canopy_mean 0.444 0p
```

```

## 4 0p      1 2005_12_2 canopy_mean 0.514 0p
## 5 0p      1 2005_13_2 canopy_mean 0.494 0p
## 6 0p      1 2005_14_2 canopy_mean 0.334 0p
## 7 0p      1 2005_15_2 canopy_mean 0.527 0p
## 8 0p      1 2005_16_2 canopy_mean 0.376 0p
## 9 0p      1 2005_17_2 canopy_mean 0.519 0p
## 10 0p     1 2005_18_2 canopy_mean 0.564 0p
## # i 1,191,518 more rows

# set path alpha
sim_path_alpha <- 0.1
buff_path_alpha <- 0.25

# set smooth alpha
sim_smooth_alpha <- 1
buff_smooth_alpha <- 1

# linewidth
sim_path_linewidth <- 0.25
buff_path_linewidth <- 0.5

# smooth linewidth
sim_smooth_linewidth <- 1
buff_smooth_linewidth <- 1

# Step length
ggplot() +

  geom_path(data = summaries_hourly_all_long_subset %>%
    filter(!Data == "Buffalo" & variable == "step_length_mean"),
    aes(x = hour, y = value, colour = Data, group = interaction(id, Data)),
    alpha = sim_path_alpha,
    linewidth = sim_path_linewidth) +
  
  geom_path(data = summaries_hourly_all_long_subset %>%
    filter(Data == "Buffalo" & variable == "step_length_mean"),
    aes(x = hour, y = value, colour = Data, group = interaction(id, Data)),
    alpha = buff_path_alpha,
    linewidth = buff_path_linewidth) +
  
  geom_smooth(data = summaries_hourly_all_long %>%
    filter(!Data == "Buffalo" & variable == "step_length_mean"),
    aes(x = hour, y = value, colour = Data, fill = Data),
    method = "gam",
    size = sim_smooth_linewidth,
    se = FALSE,
    alpha = sim_smooth_alpha) +
  
  geom_smooth(data = summaries_hourly_all_long %>%
    filter(Data == "Buffalo" & variable == "step_length_mean"),
    aes(x = hour, y = value, colour = Data, fill = Data),
    method = "gam",
    size = buff_smooth_linewidth,
    se = FALSE,
    alpha = buff_smooth_alpha) +

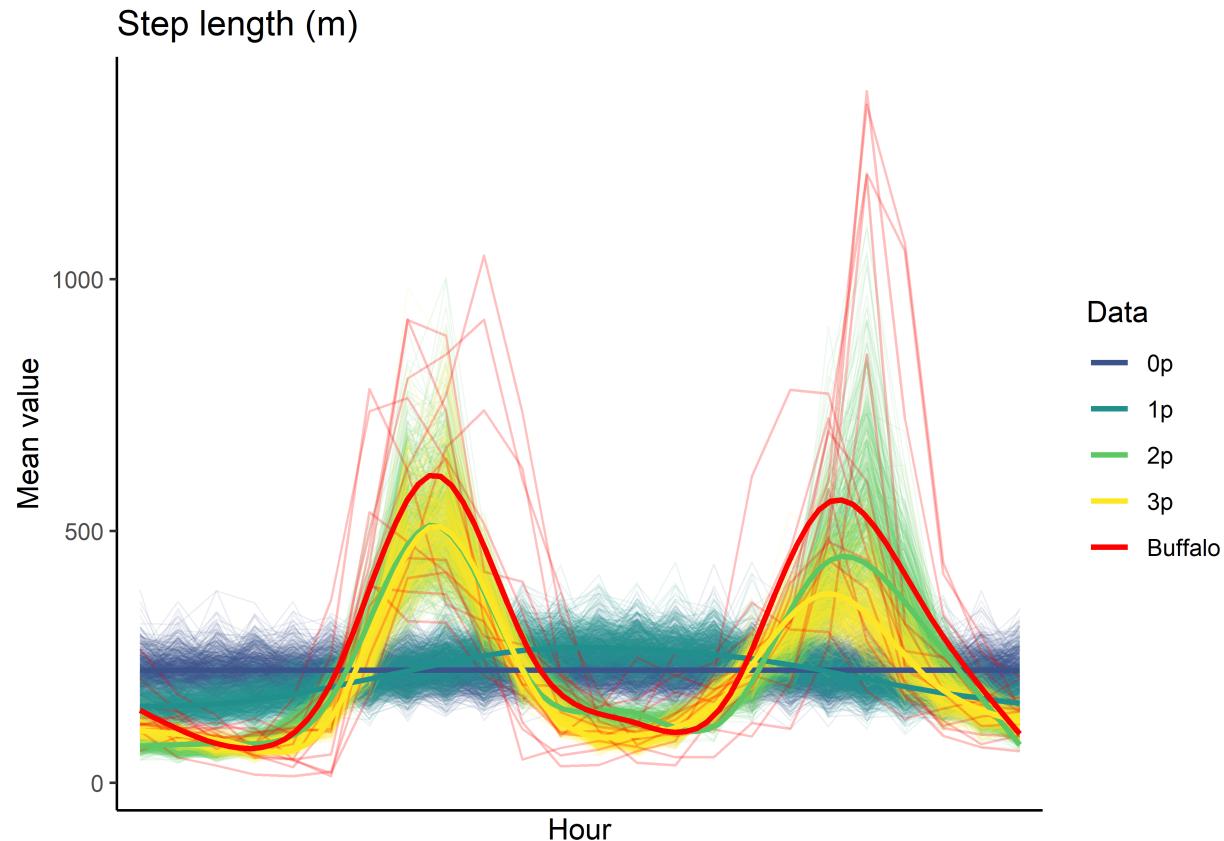
```

```

scale_fill_manual(values = colors) +
scale_colour_manual(values = colors) +
scale_x_discrete("Hour", breaks = seq(0,24,3)) +
scale_y_continuous("Mean value") +
ggtitle("Step length (m)") +
theme_classic()

## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'

```



```

# NDVI
ggplot() +

  geom_path(data = summaries_hourly_all_long_subset %>%
    filter(!Data == "Buffalo" & variable == "ndvi_mean"),
    aes(x = hour, y = value, colour = Data, group = interaction(id, Data)),
    alpha = sim_path_alpha,
    linewidth = sim_path_linewidth) +

  geom_path(data = summaries_hourly_all_long_subset %>%
    filter(Data == "Buffalo" & variable == "ndvi_mean"),
    aes(x = hour, y = value, colour = Data, group = interaction(id, Data)),
    alpha = buff_path_alpha,
    linewidth = buff_path_linewidth) +

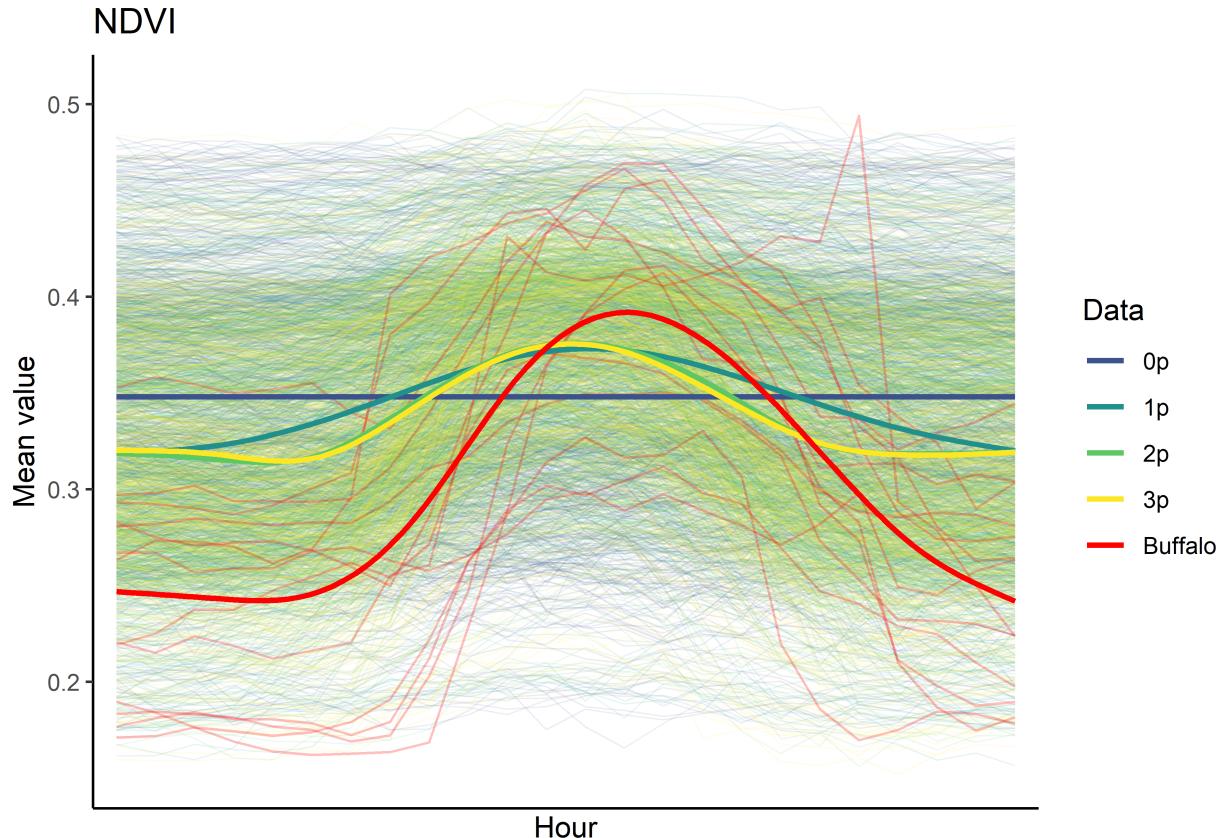
```

```

geom_smooth(data = summaries_hourly_all_long %>%
  filter(!Data == "Buffalo" & variable == "ndvi_mean"),
  aes(x = hour, y = value, colour = Data, fill = Data),
  method = "gam",
  size = sim_smooth_linewidth,
  se = FALSE,
  alpha = sim_smooth_alpha) +
geom_smooth(data = summaries_hourly_all_long %>%
  filter(Data == "Buffalo" & variable == "ndvi_mean"),
  aes(x = hour, y = value, colour = Data, fill = Data),
  method = "gam",
  size = buff_smooth_linewidth,
  se = FALSE,
  alpha = buff_smooth_alpha) +
scale_fill_manual(values = colors) +
scale_colour_manual(values = colors) +
scale_x_discrete("Hour", breaks = seq(0,24,3)) +
scale_y_continuous("Mean value") +
ggtitle("NDVI") +
theme_classic()

## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'

```



A quantile ribbon approach

To express the stochasticity of the simulations, here we show the 25th to 50th quantiles and the 2.5th to 97.5th quantiles of the data. Remember that the ‘data’ are the means for each hour for each trajectory, so the quantiles are calculated across the means for each hour.

This is the plotting approach that we used in the paper.

Here we create a ribbon from the 25th to 50th quantiles and from the 2.5th to 97.5th quantiles of the data

```
hourly_summary_quantiles <- summaries_hourly_all_long %>%
  dplyr::group_by(Data, hour, variable) %>%
  summarise(n = n(),
            mean = mean(value, na.rm = TRUE),
            sd = sd(value, na.rm = TRUE),
            q025 = quantile(value, probs = 0.025, na.rm = TRUE),
            q25 = quantile(value, probs = 0.25, na.rm = TRUE),
            q50 = quantile(value, probs = 0.5, na.rm = TRUE),
            q75 = quantile(value, probs = 0.75, na.rm = TRUE),
            q975 = quantile(value, probs = 0.975, na.rm = TRUE))
```

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

Here we show the hourly step length and selection of three of the habitat covariates. We use a dashed-line ribbon for the 95% interval and a solid-line ribbon for the 50% interval. We show the mean as a solid line.

It's quite a long plotting script...

```
# set plotting parameters here that will change in each plot
buff_path_alpha <- 0.1
ribbon_95_alpha <- 0.1
ribbon_50_alpha <- 0.15
path_95_alpha <- 1

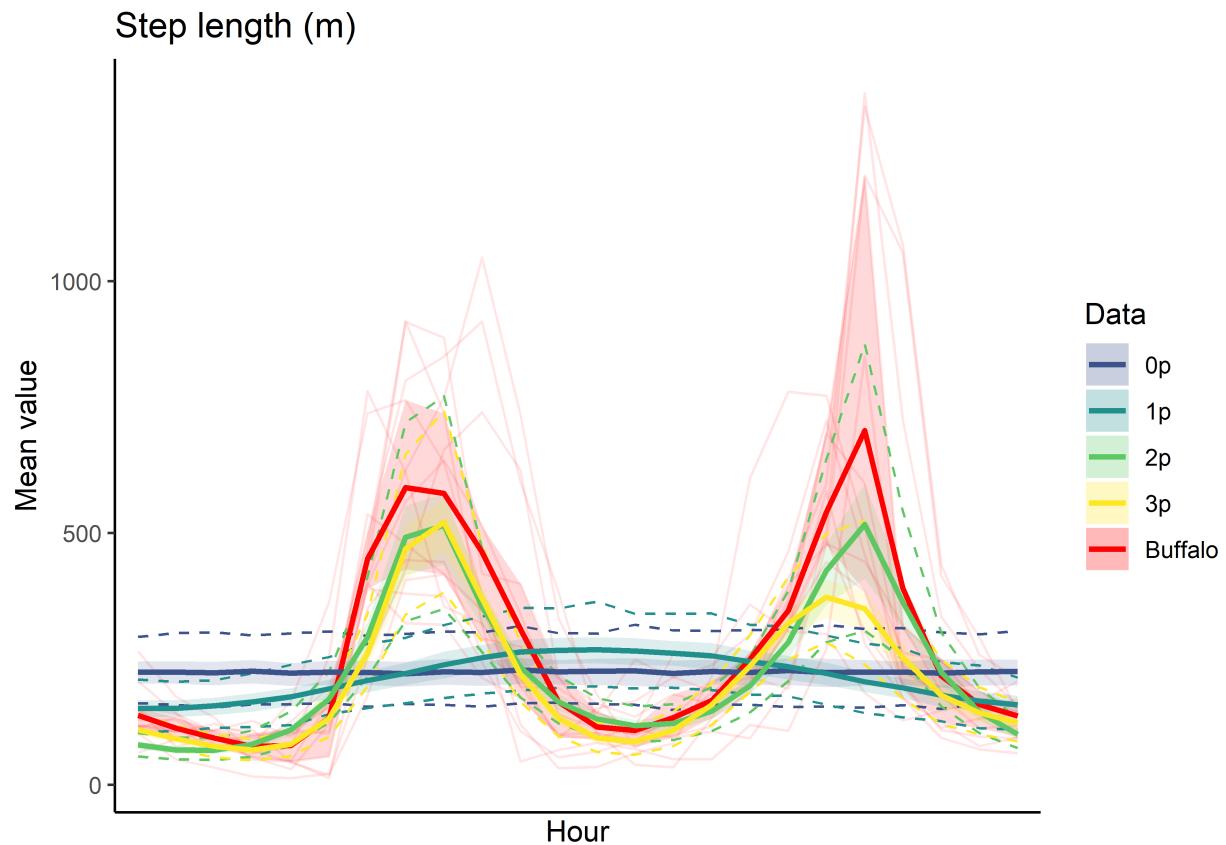
hourly_path_sl_plot <- ggplot() +
  geom_ribbon(data = hourly_summary_quantiles %>%
    filter(Data == "Buffalo" & variable == "step_length_mean"),
    aes(x = hour, ymin = q25, ymax = q75, fill = Data),
    alpha = ribbon_50_alpha) +
  geom_ribbon(data = hourly_summary_quantiles %>%
    filter(!Data == "Buffalo" & variable == "step_length_mean"),
    aes(x = hour, ymin = q25, ymax = q75, fill = Data),
    alpha = ribbon_50_alpha) +
  geom_path(data = summaries_hourly_all_long_subset %>%
    filter(Data == "Buffalo" & variable == "step_length_mean"),
    aes(x = hour, y = value, colour = Data, group = interaction(id, Data)),
    alpha = buff_path_alpha,
    linewidth = buff_path_linewidth) +
  geom_path(data = hourly_summary_quantiles %>%
    filter(!Data == "Buffalo" & variable == "step_length_mean"),
    aes(x = hour, y = q025, colour = Data),
    linetype = "dashed",
    alpha = path_95_alpha) +
```

```

geom_path(data = hourly_summary_quantiles %>%
  filter(!Data == "Buffalo" & variable == "step_length_mean"),
  aes(x = hour, y = q975, colour = Data),
  linetype = "dashed",
  alpha = path_95_alpha) +
  geom_path(data = hourly_summary_quantiles %>%
  filter(Data == "Buffalo" & variable == "step_length_mean"),
  aes(x = hour, y = mean, colour = Data),
  linewidth = 1) +
  geom_path(data = hourly_summary_quantiles %>%
  filter(!Data == "Buffalo" & variable == "step_length_mean"),
  aes(x = hour, y = mean, colour = Data),
  linewidth = 1) +
  scale_fill_manual(values = colors) +
  scale_colour_manual(values = colors) +
  scale_x_discrete("Hour", breaks = seq(0,24,3)) +
  scale_y_continuous("Mean value") +
  ggtitle("Step length (m)") +
  theme_classic()

hourly_path_sl_plot

```



```

hourly_path_ndvi_plot <- ggplot() +

  geom_ribbon(data = hourly_summary_quantiles %>%
    filter(Data == "Buffalo" & variable == "ndvi_mean"),
    aes(x = hour, ymin = q25, ymax = q75, fill = Data),
    alpha = ribbon_50_alpha) +

  geom_ribbon(data = hourly_summary_quantiles %>%
    filter(!Data == "Buffalo" & variable == "ndvi_mean"),
    aes(x = hour, ymin = q25, ymax = q75, fill = Data),
    alpha = ribbon_50_alpha) +

  geom_path(data = summaries_hourly_all_long_subset %>%
    filter(Data == "Buffalo" & variable == "ndvi_mean"),
    aes(x = hour, y = value, colour = Data, group = interaction(id, Data)),
    alpha = buff_path_alpha,
    linewidth = buff_path_linewidth) +

  geom_path(data = hourly_summary_quantiles %>%
    filter(!Data == "Buffalo" & variable == "ndvi_mean"),
    aes(x = hour, y = q025, colour = Data),
    linetype = "dashed",
    alpha = path_95_alpha) +

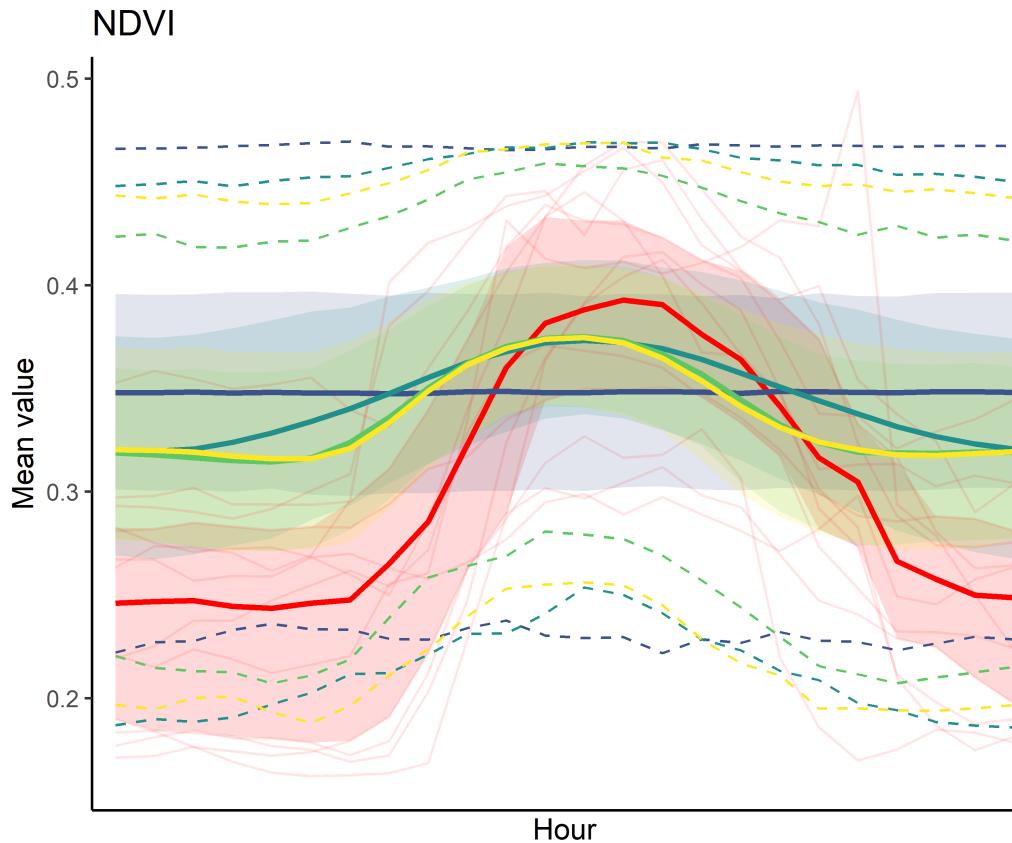
  geom_path(data = hourly_summary_quantiles %>%
    filter(!Data == "Buffalo" & variable == "ndvi_mean"),
    aes(x = hour, y = q975, colour = Data),
    linetype = "dashed",
    alpha = path_95_alpha) +

  geom_path(data = hourly_summary_quantiles %>%
    filter(Data == "Buffalo" & variable == "ndvi_mean"),
    aes(x = hour, y = mean, colour = Data),
    linewidth = 1) +

  geom_path(data = hourly_summary_quantiles %>%
    filter(!Data == "Buffalo" & variable == "ndvi_mean"),
    aes(x = hour, y = mean, colour = Data),
    linewidth = 1) +
  scale_fill_manual(values = colors) +
  scale_colour_manual(values = colors) +
  scale_x_discrete("Hour", breaks = seq(0,24,3)) +
  scale_y_continuous("Mean value") +
  ggtitle("NDVI") +
  theme_classic()

hourly_path_ndvi_plot

```



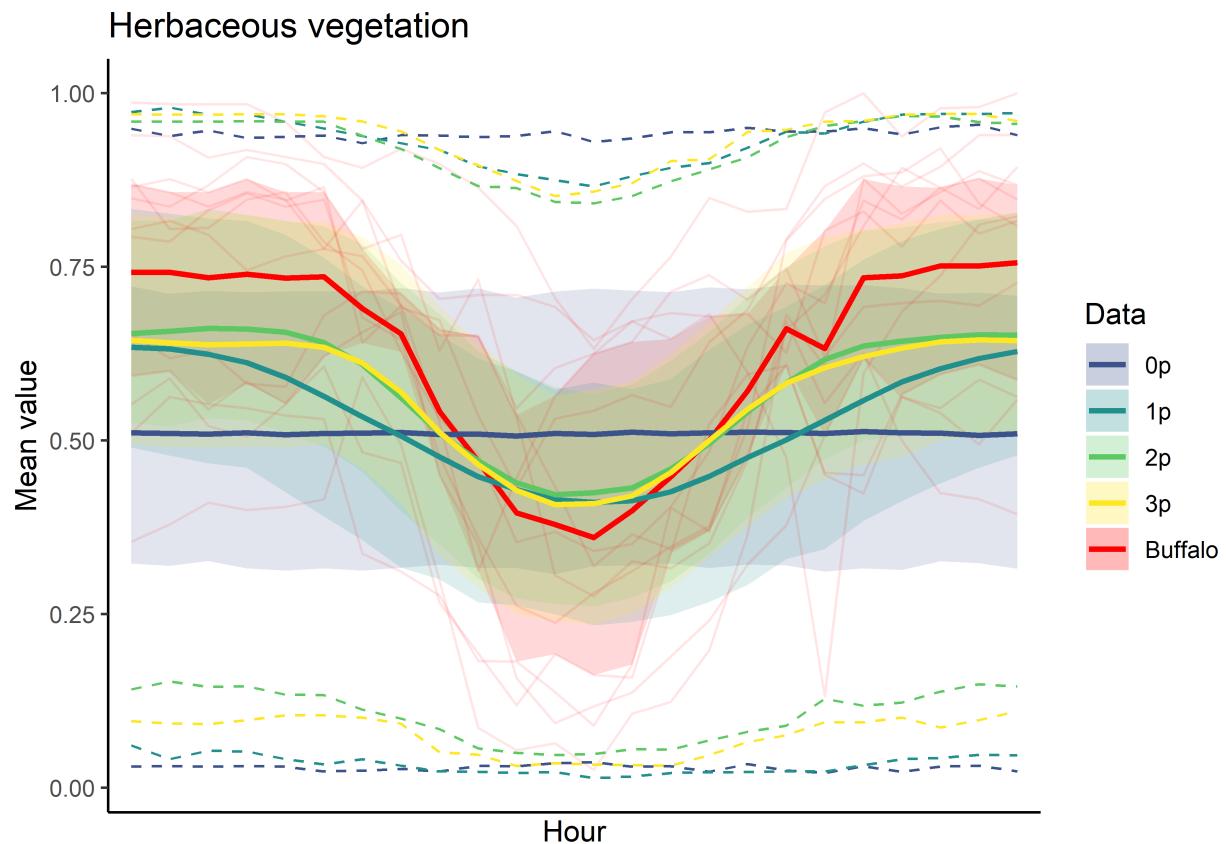
```
hourly_path_herby_plot <- ggplot() +
  geom_ribbon(data = hourly_summary_quantiles %>%
    filter(Data == "Buffalo" & variable == "herby_mean"),
    aes(x = hour, ymin = q25, ymax = q75, fill = Data),
    alpha = ribbon_50_alpha) +
  geom_ribbon(data = hourly_summary_quantiles %>%
    filter(!Data == "Buffalo" & variable == "herby_mean"),
    aes(x = hour, ymin = q25, ymax = q75, fill = Data),
    alpha = ribbon_50_alpha) +
  geom_path(data = summaries_hourly_all_long_subset %>%
    filter(Data == "Buffalo" & variable == "herby_mean"),
    aes(x = hour, y = value, colour = Data, group = interaction(id, Data)),
    alpha = buff_path_alpha,
    linewidth = buff_path_linewidth) +
  geom_path(data = hourly_summary_quantiles %>%
    filter(!Data == "Buffalo" & variable == "herby_mean"),
    aes(x = hour, y = q025, colour = Data),
    linetype = "dashed",
    alpha = path_95_alpha) +
  geom_path(data = hourly_summary_quantiles %>%
```

```

filter(!Data == "Buffalo" & variable == "herby_mean"),
aes(x = hour, y = q975, colour = Data),
linetype = "dashed",
alpha = path_95_alpha) +
geom_path(data = hourly_summary_quantiles %>%
  filter(Data == "Buffalo" & variable == "herby_mean"),
  aes(x = hour, y = mean, colour = Data),
  linewidth = 1) +
geom_path(data = hourly_summary_quantiles %>%
  filter(!Data == "Buffalo" & variable == "herby_mean"),
  aes(x = hour, y = mean, colour = Data),
  linewidth = 1) +
scale_fill_manual(values = colors) +
scale_colour_manual(values = colors) +
scale_x_discrete("Hour", breaks = seq(0,24,3)) +
scale_y_continuous("Mean value") +
ggtitle("Herbaceous vegetation") +
theme_classic()

hourly_path_herby_plot

```

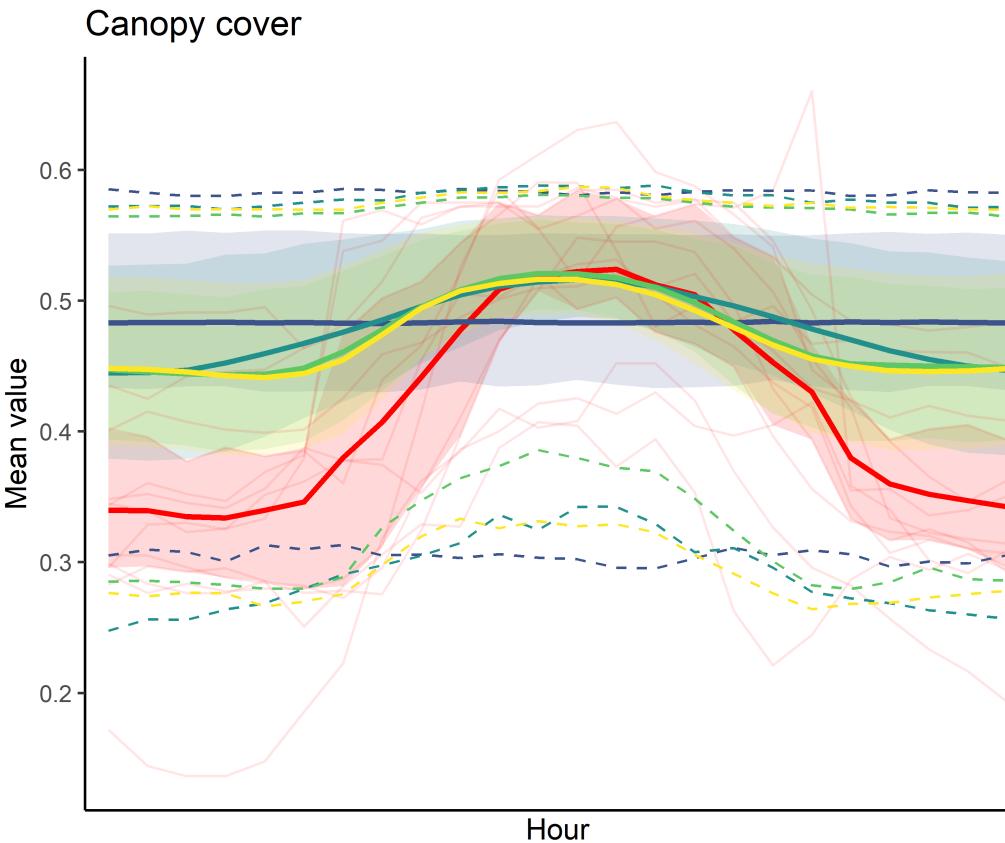


```

hourly_path_canopy_plot <- ggplot() +
  geom_ribbon(data = hourly_summary_quantiles %>%
    filter(Data == "Buffalo" & variable == "canopy_mean"),
    aes(x = hour, ymin = q25, ymax = q75, fill = Data),
    alpha = ribbon_50_alpha) +
  geom_ribbon(data = hourly_summary_quantiles %>%
    filter(!Data == "Buffalo" & variable == "canopy_mean"),
    aes(x = hour, ymin = q25, ymax = q75, fill = Data),
    alpha = ribbon_50_alpha) +
  geom_path(data = summaries_hourly_all_long_subset %>%
    filter(Data == "Buffalo" & variable == "canopy_mean"),
    aes(x = hour, y = value, colour = Data, group = interaction(id, Data)),
    alpha = buff_path_alpha,
    linewidth = buff_path_linewidth) +
  geom_path(data = hourly_summary_quantiles %>%
    filter(!Data == "Buffalo" & variable == "canopy_mean"),
    aes(x = hour, y = q025, colour = Data),
    linetype = "dashed",
    alpha = path_95_alpha) +
  geom_path(data = hourly_summary_quantiles %>%
    filter(!Data == "Buffalo" & variable == "canopy_mean"),
    aes(x = hour, y = q975, colour = Data),
    linetype = "dashed",
    alpha = path_95_alpha) +
  geom_path(data = hourly_summary_quantiles %>%
    filter(Data == "Buffalo" & variable == "canopy_mean"),
    aes(x = hour, y = mean, colour = Data),
    linewidth = 1) +
  geom_path(data = hourly_summary_quantiles %>%
    filter(!Data == "Buffalo" & variable == "canopy_mean"),
    aes(x = hour, y = mean, colour = Data),
    linewidth = 1) +
  scale_fill_manual(values = colors) +
  scale_colour_manual(values = colors) +
  scale_x_discrete("Hour", breaks = seq(0,24,3)) +
  scale_y_continuous("Mean value") +
  ggtitle("Canopy cover") +
  theme_classic()

hourly_path_canopy_plot

```



Combining the hourly plots

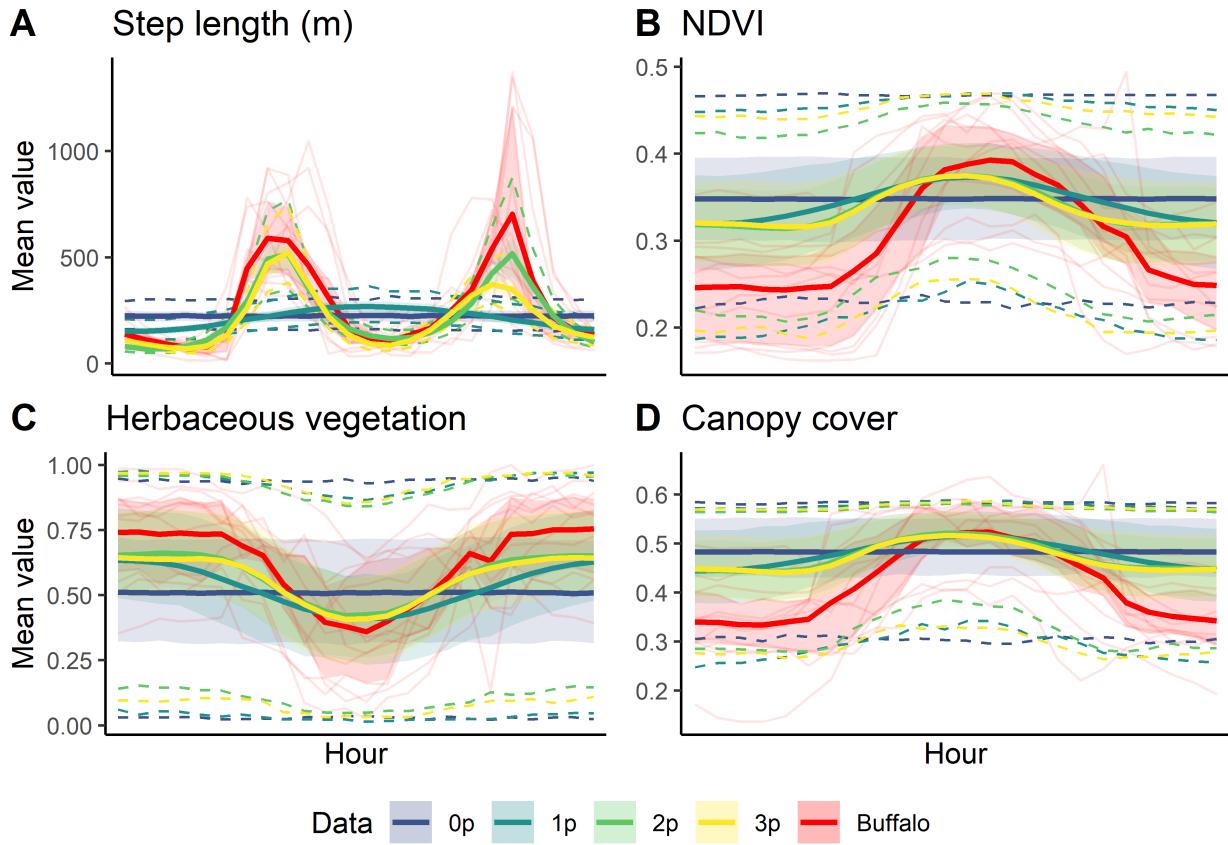
```
ggarrange(hourly_path_sl_plot +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank()),

  hourly_path_ndvi_plot +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title.y = element_blank()),

  hourly_path_herby_plot,

  hourly_path_canopy_plot +
  theme(axis.title.y = element_blank()),

  labels = c("A", "B", "C", "D"),
  ncol = 2, nrow = 2,
  legend = "bottom",
  common.legend = TRUE)
```



```
ggsave(paste0("outputs/plots/manuscript_figs/memALL_hourly_path_mean_all_",
              Sys.Date(), ".png"),
       width=150, height=120, units="mm", dpi = 600)
```

Full trajectory summary statistics

Here we compare between the observed and simulated trajectories using summary statistics that result in a single value per trajectory (rather than one per hour as above). These summary statistics were also used by Abrahms et al. (2017) and Kays et al. (2023), although to compare between taxa rather than between observed and simulated data.

Import the summary statistic data frames from the observed and simulated datasets and combine into a single data frame for plotting

```
summaries_buffalo <-
  read_csv("outputs/buffalo_summary_statistics_df_2024-02-07.csv")

summaries_0p <-
  read_csv("outputs/sim_0p_memALL_daily_summary_statistics_df_2024-02-07.csv")
summaries_0p <- summaries_0p %>% mutate(sim = "0p")

summaries_1p <-
  read_csv("outputs/sim_1p_memALL_daily_summary_statistics_df_2024-02-07.csv")
summaries_1p <- summaries_1p %>% mutate(sim = "1p")
```

```

summaries_2p <-
  read_csv("outputs/sim_2p_memALL_daily_summary_statistics_df_2024-02-07.csv")
summaries_2p <- summaries_2p %>% mutate(sim = "2p")

summaries_3p <-
  read_csv("outputs/sim_3p_memALL_daily_summary_statistics_df_2024-02-07.csv")
summaries_3p <- summaries_3p %>% mutate(sim = "3p")

summaries_all <- bind_rows(summaries_buffalo,
                            summaries_0p,
                            summaries_1p,
                            summaries_2p,
                            summaries_3p)

summaries_all <- summaries_all %>% mutate(residence_time_log = log(residence_time),
                                             msd_log = log(msd))

summaries_all_long <- summaries_all %>%
  pivot_longer(cols = !c(traj, id, data, sim),
               names_to = "summary", values_to = "value") %>%

  mutate(sim = factor(sim, levels = c("obs", "0p", "1p", "2p", "3p"),
                      labels = c("Buffalo", "0p", "1p", "2p", "3p")),

  # categorise the summary statistics for plotting
  category = case_when(summary == "step_length_median" ~ "step",
                        summary == "step_length_mean" ~ "step",
                        summary == "gamma_shape" ~ "step",
                        summary == "gamma_scale" ~ "step",
                        summary == "vm_kappa" ~ "step",

                        summary == "straightness" ~ "path",
                        summary == "sinuosity" ~ "path",

                        summary == "tac" ~ "life_history",
                        summary == "cum_dist" ~ "life_history",
                        summary == "tot_dist" ~ "life_history",
                        summary == "msd" ~ "life_history",
                        summary == "msd_log" ~ "life_history",
                        summary == "intensity_use" ~ "life_history",
                        summary == "residence_time" ~ "life_history",
                        summary == "residence_time_log" ~ "life_history",
                        summary == "hr_area_50" ~ "life_history",
                        summary == "hr_area_75" ~ "life_history",
                        summary == "hr_area_95" ~ "life_history",
                        summary == "hr_monthly_overlap_ba" ~ "life_history",
                        summary == "hr_monthly_overlap_vi" ~ "life_history",
                        summary == "ba_overlap" ~ "life_history",

                        summary == "ndvi_mean" ~ "habitat",
                        summary == "ndvi_median" ~ "habitat",
                        summary == "ndvi_sd" ~ "habitat",
                        summary == "herby_mean" ~ "habitat",

```

```

        summary == "herby_sd" ~ "habitat",
        summary == "canopy_mean" ~ "habitat",
        summary == "canopy_sd" ~ "habitat",
        summary == "slope_mean" ~ "habitat",
        summary == "slope_median" ~ "habitat",
        summary == "slope_sd" ~ "habitat"

    )))

head(summaries_all_long)

```

```

## # A tibble: 6 x 7
##   traj      id data sim     summary      value category
##   <chr>    <dbl> <chr> <fct>    <chr>      <dbl> <chr>
## 1 obs      2005 obs Buffalo step_length_median 92.5    step
## 2 obs      2005 obs Buffalo step_length_mean   194.    step
## 3 obs      2005 obs Buffalo ndvi_mean        0.307   habitat
## 4 obs      2005 obs Buffalo ndvi_median      0.294   habitat
## 5 obs      2005 obs Buffalo ndvi_sd         0.0669  habitat
## 6 obs      2005 obs Buffalo herby_mean       0.764   habitat

```

```
tail(summaries_all_long)
```

```

## # A tibble: 6 x 7
##   traj      id data sim     summary      value category
##   <chr>    <dbl> <chr> <fct>    <chr>      <dbl> <chr>
## 1 2393_9_26 2393 sim  3p hr_area_95    33.4   life_history
## 2 2393_9_26 2393 sim  3p hr_monthly_overlap_ba 0.482   life_history
## 3 2393_9_26 2393 sim  3p hr_monthly_overlap_vi 0.243   life_history
## 4 2393_9_26 2393 sim  3p ba_overlap        0.290   life_history
## 5 2393_9_26 2393 sim  3p residence_time_log 1.94    life_history
## 6 2393_9_26 2393 sim  3p msd_log          15.8   life_history

```

Calculate the mean of the summary statistic values for plotting

```

summary_mean_values <- summaries_all_long %>%
  group_by(summary, sim) %>%
  summarise(mean_value = mean(value, na.rm = TRUE),
            median_value = median(value, na.rm = TRUE))

```

‘summarise()’ has grouped output by ‘summary’. You can override using the ‘.groups’ argument.

```

summaries_all_long %>%
  group_by(summary) %>%
  summarise(mean = mean(value, na.rm = TRUE),
            sd = sd(value, na.rm = TRUE),
            median = median(value, na.rm = TRUE),
            min = min(value, na.rm = TRUE),
            max = max(value, na.rm = TRUE)) %>%
  arrange(summary)

```

```

## # A tibble: 31 x 6
##   summary      mean       sd     median      min      max
##   <chr>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>
## 1 ba_overlap  0.267    0.197    0.259  1.43e-9  0.850
## 2 canopy_mean 0.475    0.0743   0.485  2.23e-1  0.610

```

```

## 3 canopy_sd      0.125      0.0432      0.126 1.11e-2      0.278
## 4 cum_dist     477113.    53218.    475607.    3.11e+5 1001533.
## 5 gamma_scale   467.      53.7       461.    3.45e+2    1017.
## 6 gamma_shape    0.461     0.0172     0.460 3.57e-1      0.524
## 7 herby_mean     0.546     0.233      0.574 3.42e-3      0.995
## 8 herby_sd       0.432     0.0833     0.464 5.84e-2      0.500
## 9 hr_area_50     11.9      20.9       4.58 8.22e-1     298.
## 10 hr_area_75    28.5      47.9       10.7  1.73e+0     580.
## # i 21 more rows

```

Step measurements

```

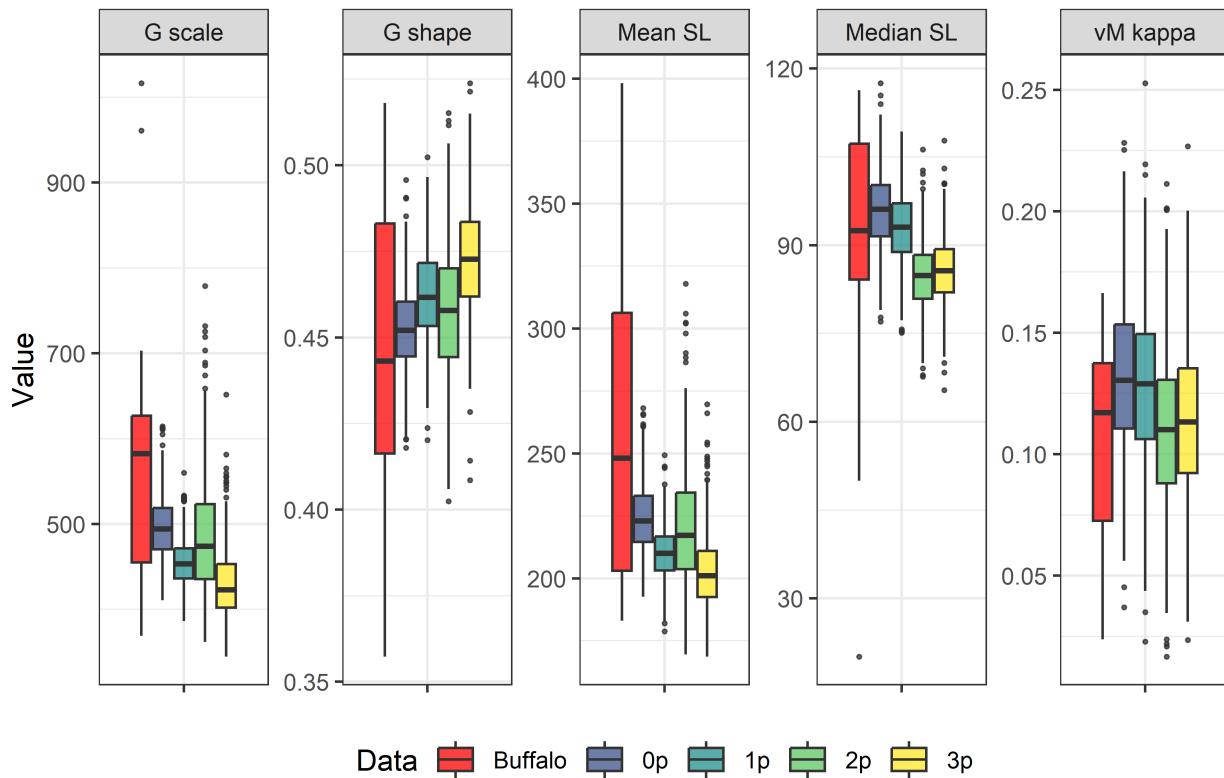
# Create color mapping
unique_groups <- unique(summaries_all_long$sim)
colors <- viridis(length(unique_groups))
names(colors) <- unique_groups
colors["Buffalo"] <- "red"

step_summary_labs <- c("gamma_shape" = "G shape",
                       "gamma_scale" = "G scale",
                       "step_length_mean" = "Mean SL",
                       "step_length_median" = "Median SL",
                       "vm_kappa" = "vM kappa")

ggplot() +
  geom_boxplot(data = summaries_all_long %>%
                filter(category == "step" #&
                      # !str_detect(summary, "gamma")
                      ),
                aes(x = summary, y = value, fill = sim), alpha = 0.75,
                outlier.size = 0.5) +
  ggh4x::facet_grid2(scales = "free", cols = vars(summary),
                     independent = "y",
                     labeller = labeller(summary = step_summary_labs)) +
  ggtitle("Step summary statistics") +
  scale_fill_manual("Data", values = colors) +
  scale_y_continuous("Value") +
  theme_bw() +
  theme(legend.position = "bottom",
        axis.text.x = element_blank(),
        axis.title.x = element_blank())

```

Step summary statistics



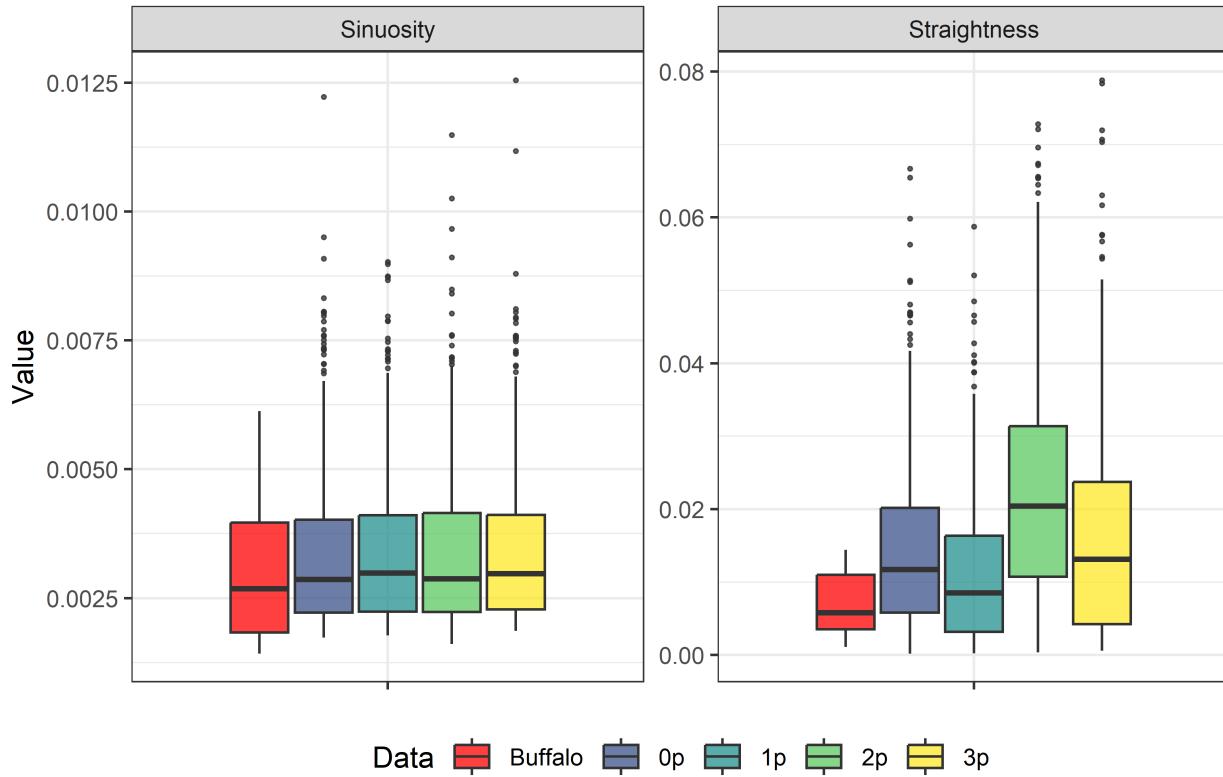
```
# save the plot using ggplot
ggsave(paste0("outputs/plots/manuscript_figs/memALL_summary_comparison_steps_",
              Sys.Date(), ".png"),
       width=150, height=90, units="mm", dpi = 600)
```

Path measurements

```
summary_labs <- c("sinuosity" = "Sinuosity", "straightness" = "Straightness")

ggplot() +
  geom_boxplot(data = summaries_all_long %>%
                filter(category == "path"),
                aes(x = summary, y = value, fill = sim), alpha = 0.75,
                outlier.size = 0.5) +
  ggh4x::facet_grid2(scales = "free", cols = vars(summary), independent = "y"
                      , labeller = labeller(summary = summary_labs)
                      ) +
  ggtitle("Path summary statistics") +
  scale_fill_manual("Data", values = colors) +
  scale_y_continuous("Value") +
  theme_bw() +
  theme(legend.position = "bottom",
        axis.text.x = element_blank(),
        axis.title.x = element_blank())
```

Path summary statistics



```
# save the plot using ggplot
ggsave(paste0("outputs/plots/manuscript_figs/memALL_summary_comparison_paths_",
              Sys.Date(), ".png"),
        width=150, height=90, units="mm", dpi = 600)
```

Life-history measurements

```
lh_summary_labs <- c("intensity_use" = "Intensity of Use",
                      "msd" = "MSD (m^2)",
                      "residence_time" = "RT (hours)",
                      "tac" = "Mean TAC",
                      "tot_dist" = "Tot Dist (m)")

ggplot() +
  geom_boxplot(data = summaries_all_long %>%
                filter(category == "life_history" &
                           (str_detect(summary, "tac") |
                           str_detect(summary, "residence_time") |
                           str_detect(summary, "tot_d") |
                           str_detect(summary, "intensity_use")) &
                           !str_detect(summary, "log")),
                aes(x = summary, y = value, fill = sim), alpha = 0.75,
                outlier.size = 0.5) +
  ggh4x::facet_grid2(scales = "free", cols = vars(summary), independent = "y",
```

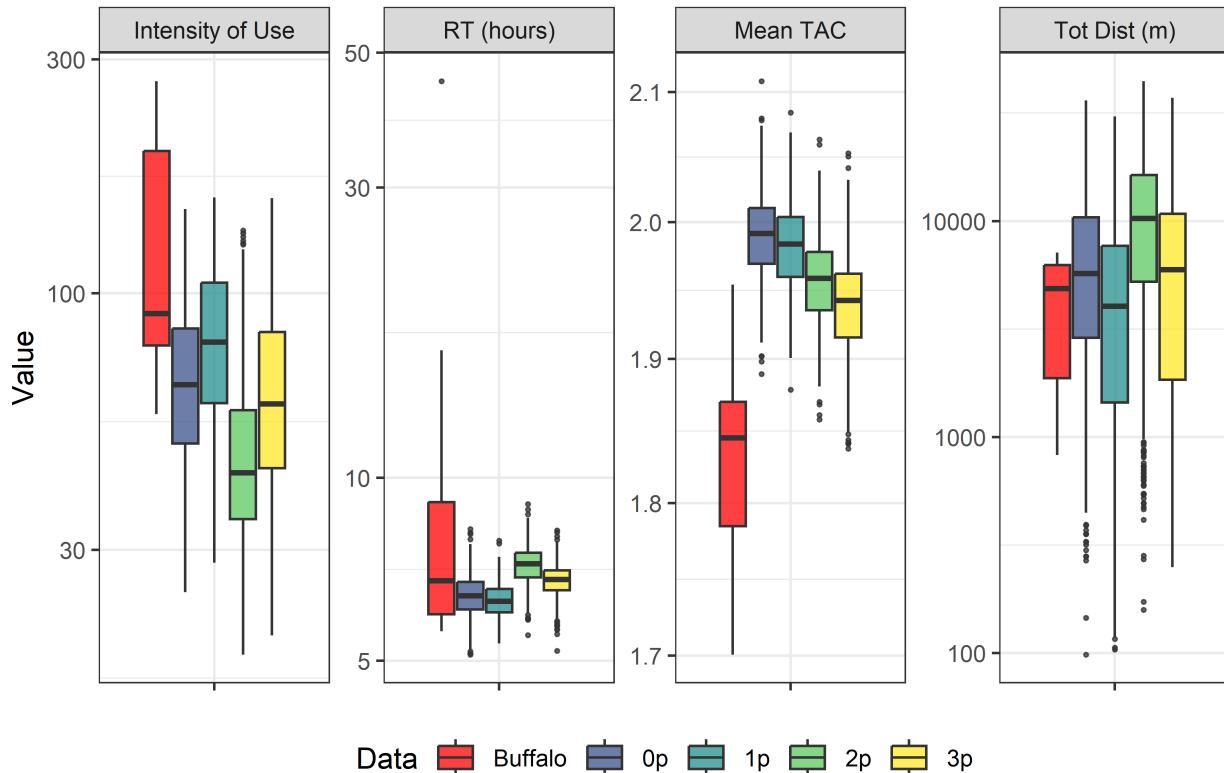
```

    labeller = labeller(summary = lh_summary_labs)) +
ggtitle("Life-history summary statistics") +
scale_fill_manual("Data", values = colors) +
scale_y_log10("Value") +
# scale_y_continuous("Value") +
theme_bw() +
theme(legend.position = "bottom",
axis.text.x = element_blank(),
axis.title.x = element_blank())

## Warning: Removed 7 rows containing non-finite values ('stat_boxplot()').

```

Life-history summary statistics



```

# save the plot using ggplot
ggsave(paste0("outputs/plots/manuscript_figs/memALL_summary_comparison_life_history_",
Sys.Date(), ".png"),
width=150, height=90, units="mm", dpi = 600)

```

Warning: Removed 7 rows containing non-finite values ('stat_boxplot()').

Home range area (KDE) and monthly overlap

```

hr_summary_labs <- c("hr_area_50" = "KDE 50%",
"hr_area_75" = "KDE 75%",
"hr_area_95" = "KDE 95%",
"hr_monthly_overlap_ba" = "Monthly overlap")

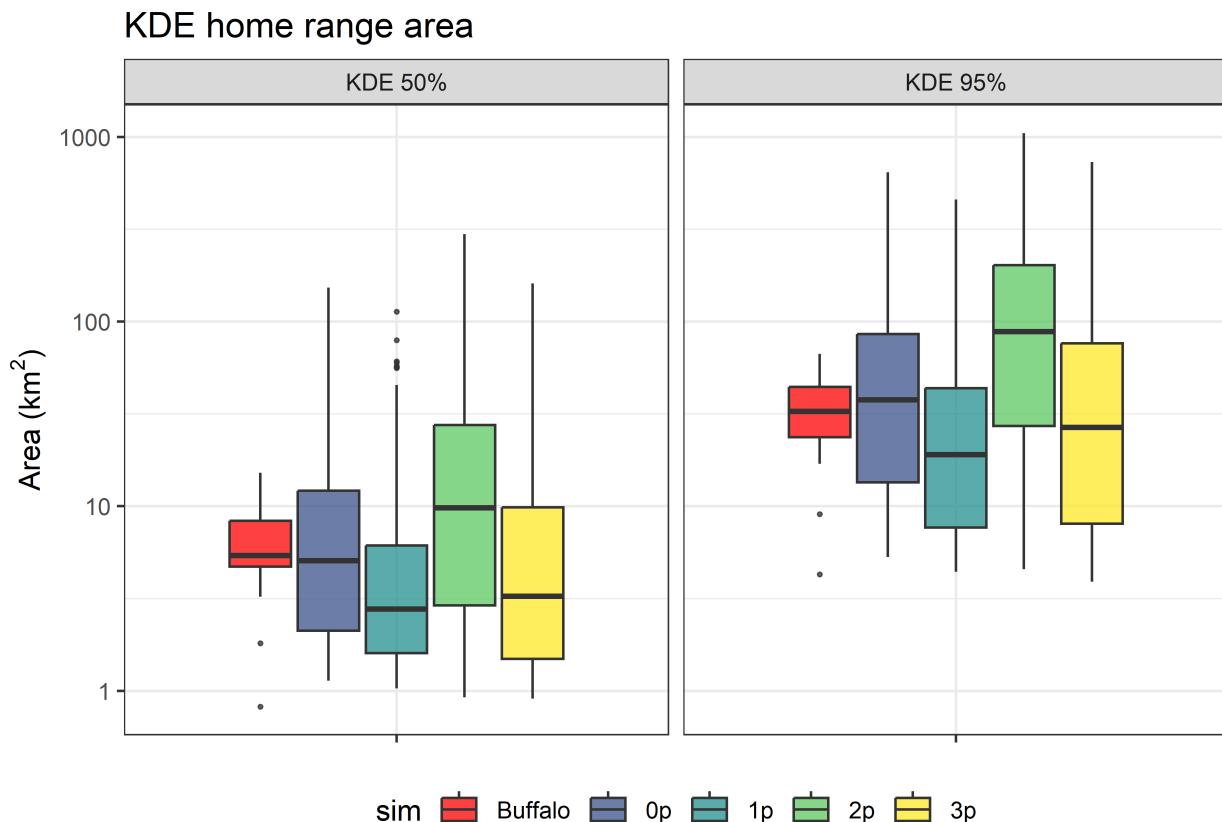
```

```

kde_hr_plot <- ggplot() +
  geom_boxplot(data = summaries_all_long %>%
    filter(category == "life_history" &
           str_detect(summary, "hr_area") &
           !str_detect(summary, "75")),
    aes(x = summary, y = value, fill = sim), alpha = 0.75,
    outlier.size = 0.5) +
  ggh4x::facet_grid2(scales = "free",
    cols = vars(summary),
    # independent = "y",
    remove_labels = "all",
    labeller = labeller(summary = hr_summary_labs)
  ) +
  ggtitle("KDE home range area") +
  scale_fill_manual(values = colors) +
  scale_y_log10(expression("Area (km" ^ 2 * "))) +
  theme_bw() +
  theme(legend.position = "bottom",
    axis.text.x = element_blank(),
    axis.title.x = element_blank())

```

kde_hr_plot

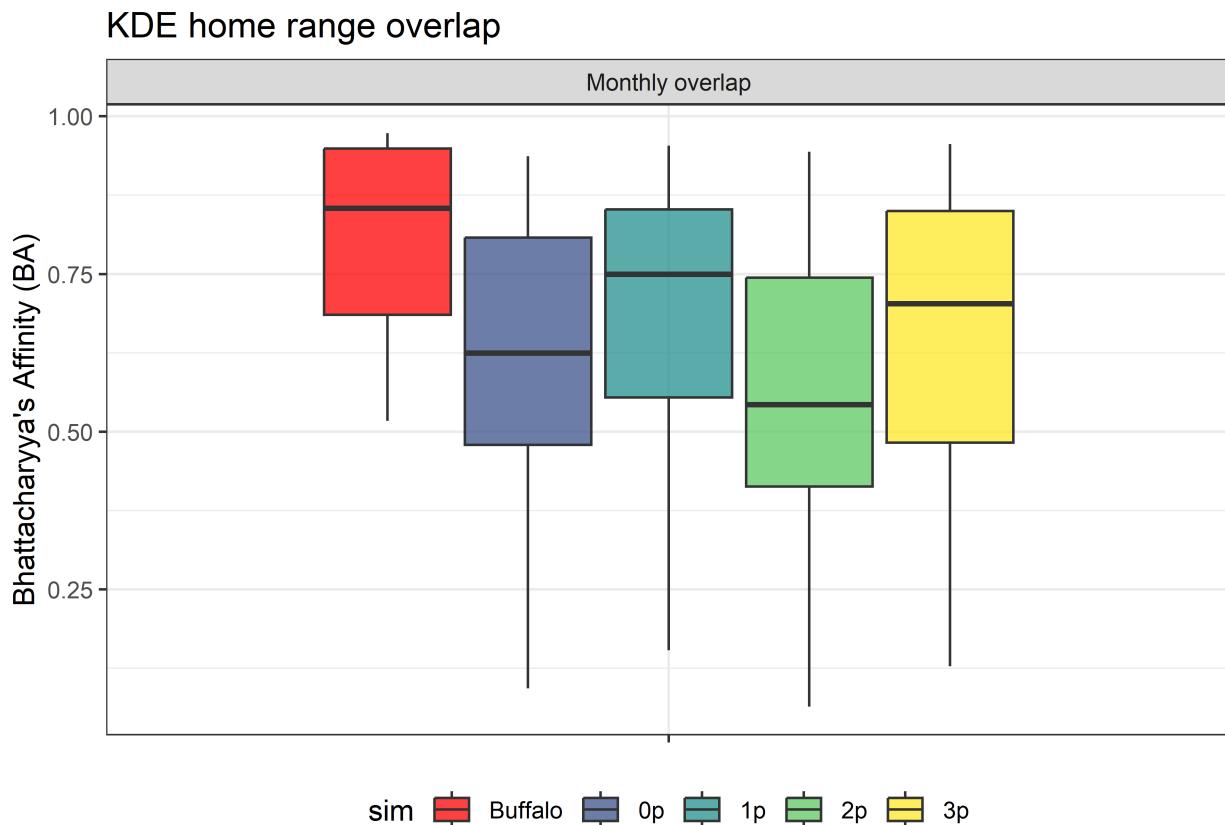


```

kde_hr_overlap_plot <- ggplot() +
  geom_boxplot(data = summaries_all_long %>%
    filter(category == "life_history" &
           str_detect(summary, "hr_monthly_overlap_ba")),
    aes(x = summary, y = value, fill = sim), alpha = 0.75,
    outlier.size = 0.5) +
  ggh4x::facet_grid2(scales = "free", cols = vars(summary), independent = "y",
    labeller = labeller(summary = hr_summary_labs))
  ) +
  ggtitle("KDE home range overlap") +
  scale_fill_manual(values = colors) +
  scale_y_continuous("Bhattacharyya's Affinity (BA)") +
  theme_bw() +
  theme(legend.position = "bottom",
    axis.text.x = element_blank(),
    axis.title.x = element_blank())

```

kde_hr_overlap_plot



Combining plots

```

ggarrange(kde_hr_plot +
  ggtitle("Home range area") +

```

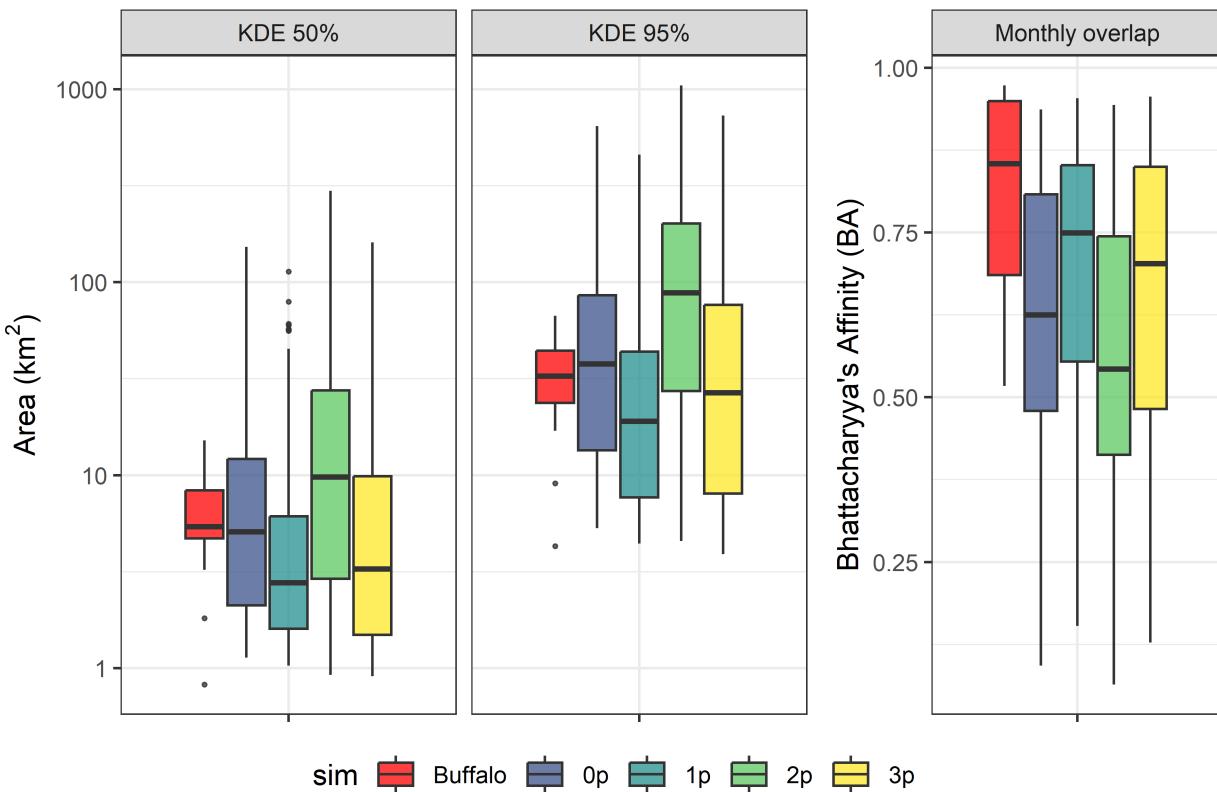
```

    theme(axis.title.x = element_blank()),

    kde_hr_overlap_plot +
      ggtitle("") +
      theme_bw() +
      theme(axis.title.x = element_blank(),
            axis.text.x = element_blank()),
  ncol = 2, nrow = 1,
  widths = c(2,1),
  legend = "bottom",
  common.legend = TRUE)

```

Home range area



```

ggsave(paste0("outputs/plots/manuscript_figs/memALL_kde_overlap_",
              Sys.Date(), ".png"),
       width=150, height=120, units="mm", dpi = 600)

```

Habitat summary statistics comparison

```

habitat_summaries_N0slope_plot <- ggplot() +
  geom_boxplot(data = summaries_all_long %>%
                filter(category == "habitat" &
                       str_detect(summary, "mean")) %>%
                filter(!str_detect(summary, "slope")),
                aes(x = summary, y = value, fill = sim), alpha = 0.75,

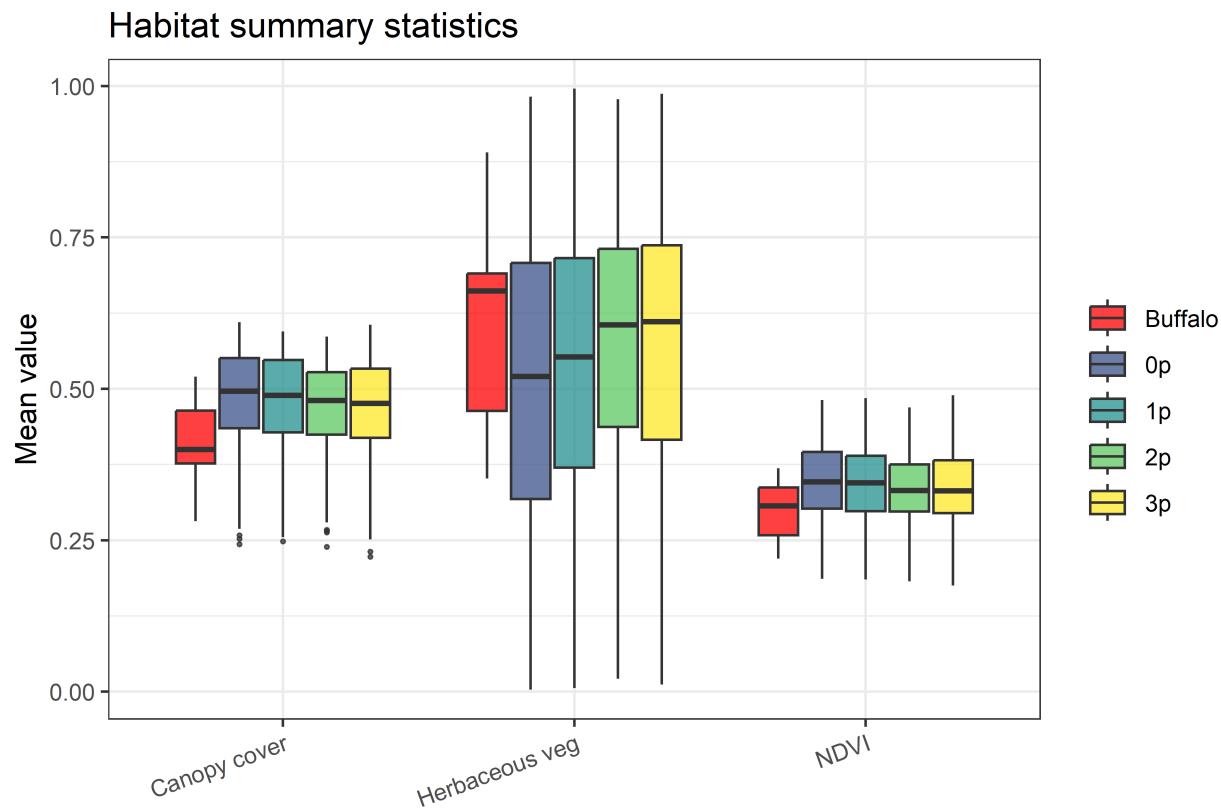
```

```

        outlier.size = 0.5) +
ggtitle("Habitat summary statistics") +
scale_fill_manual(values = colors, "") +
scale_x_discrete("", labels = c("Canopy cover", "Herbaceous veg", "NDVI")) +
scale_y_continuous("Mean value") +
theme_bw() +
theme(legend.position = "right",
axis.text.x = element_text(angle = 20, hjust = 1))

habitat_summaries_N0slope_plot

```



```

habitat_summaries_slope_plot <- ggplot() +
  geom_boxplot(data = summaries_all_long %>%
                filter(category == "habitat" &
                       str_detect(summary, "mean")) %>%
                filter(str_detect(summary, "slope")),
                aes(x = summary, y = value, fill = sim), alpha = 0.75,
                outlier.size = 0.5) +
  ggtitle(" ") +
  scale_fill_manual(values = colors, "") +
  scale_x_discrete("", labels = c("Slope")) +
  scale_y_continuous("Mean value") +
  theme_bw() +
  theme(legend.position = "right",

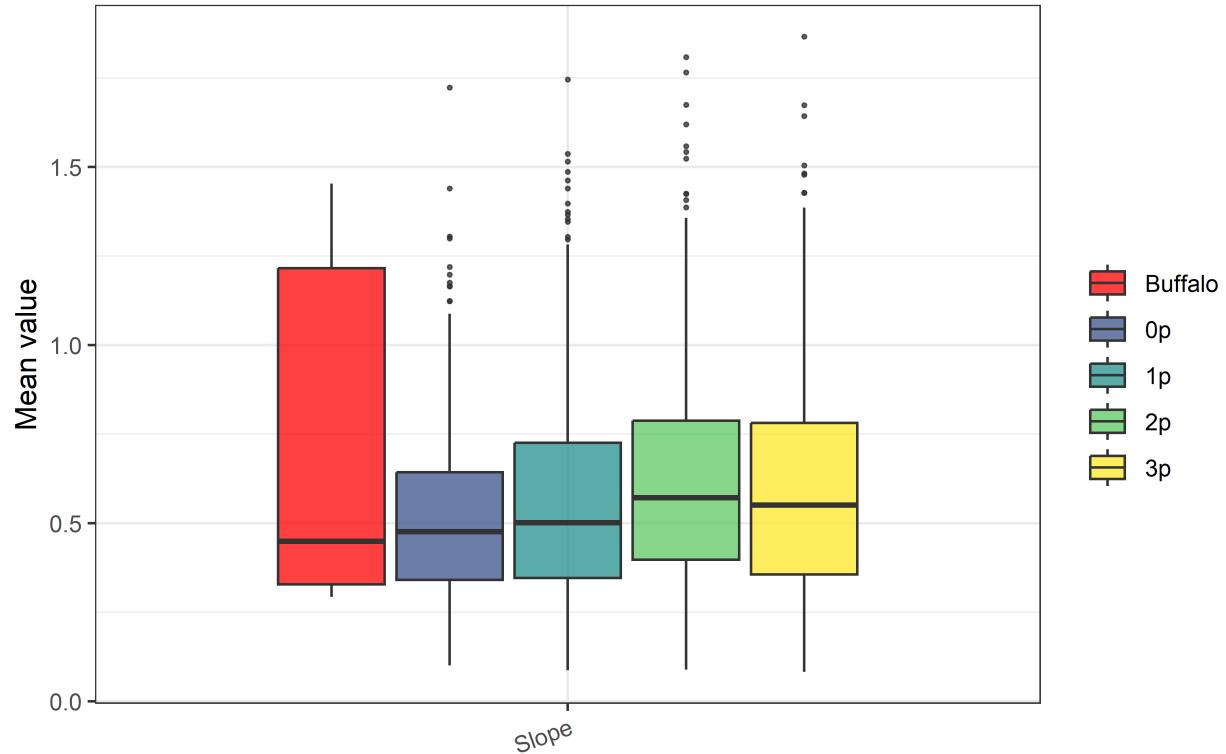
```

```

axis.text.x = element_text(angle = 20, hjust = 1))

habitat_summaries_slope_plot

```



Combining plots

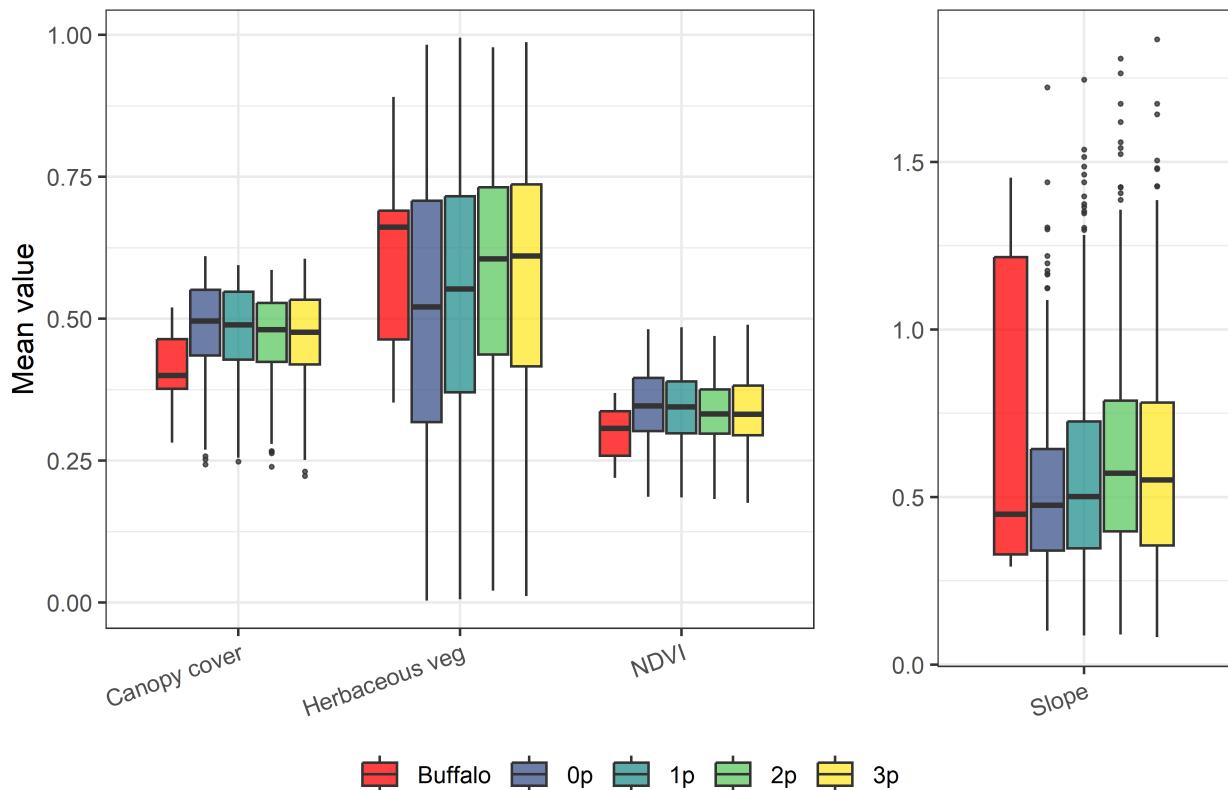
```

ggarrange(habitat_summaries_N0slope_plot + theme(axis.title.x = element_blank()),

          habitat_summaries_slope_plot + theme(axis.title.x = element_blank(),
                                                axis.title.y = element_blank()),
          ncol = 2, nrow = 1,
          widths = c(2,1),
          align = "v",
          legend = "bottom",
          common.legend = TRUE)

```

Habitat summary statistics



```
ggsave(paste0("outputs/plots/manuscript_figs/memALL_overall_habitat_overlap_",
              Sys.Date(), ".png"),
       width=150, height=120, units="mm", dpi = 600)
```

PCA of summary statistics

Similar to the approach of Abrahms et al. (2017) and Kays et al. (2023), we can use PCA to reduce the dimensionality of the summaries and assess the differences between the observed and simulated data in the latent space.

```
# remove some of the columns that should not be included in the PCA, such as id etc,
# and some that are covered by another summary, such as the monthly overlap with
# volume of intersection, as this information is largely contained in the Bhattacharyya's
# affinity overlap
summaries_all_pca <- summaries_all %>%
  dplyr::select(-traj, -id, -data, -sim, -residence_time_log,
               -msd_log, -ba_overlap, -hr_monthly_overlap_vi) %>%
  na.omit() %>%
  prcomp(center = TRUE, scale. = TRUE)

# summaries_all_pca

#PCA diagnostics
summary(summaries_all_pca)
```

```

## Importance of components:
##          PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9     PC10
## Standard deviation 2.7786 2.0941 1.6678 1.45843 1.28446 1.11794 1.01910 0.99455 0.96955 0.91554
## Proportion of Variance 0.2859 0.1624 0.1030 0.07878 0.06111 0.04629 0.03847 0.03663 0.03482 0.03105
## Cumulative Proportion 0.2859 0.4484 0.5514 0.63016 0.69126 0.73755 0.77602 0.81265 0.84747 0.87851
##          PC13    PC14    PC15    PC16    PC17    PC18    PC19    PC20    PC21    PC22
## Standard deviation 0.68815 0.61144 0.53778 0.47387 0.42460 0.31401 0.26209 0.25837 0.19067 0.15981
## Proportion of Variance 0.01754 0.01385 0.01071 0.00832 0.00668 0.00365 0.00254 0.00247 0.00135 0.00098
## Cumulative Proportion 0.94837 0.96222 0.97293 0.98125 0.98792 0.99158 0.99412 0.99659 0.99794 0.99821
##          PC25    PC26    PC27
## Standard deviation 0.07107 0.05445 0.03721
## Proportion of Variance 0.00019 0.00011 0.00005
## Cumulative Proportion 0.99984 0.99995 1.00000

# biplot(summaries_all_pca)
# screeplot(summaries_all_pca, type = "l")

# proportion of variance explained
prop_var <- summaries_all_pca$sdev^2 / sum(summaries_all_pca$sdev^2)

# to create of vector of the simulation labels
# this should remove the same columns as above, besides sim
sims <- summaries_all %>%
  dplyr::select(-traj, -id, -data, -residence_time_log, -msd_log,
               -ba_overlap, -hr_monthly_overlap_vi) %>%
  na.omit() %>%
  pull(sim)

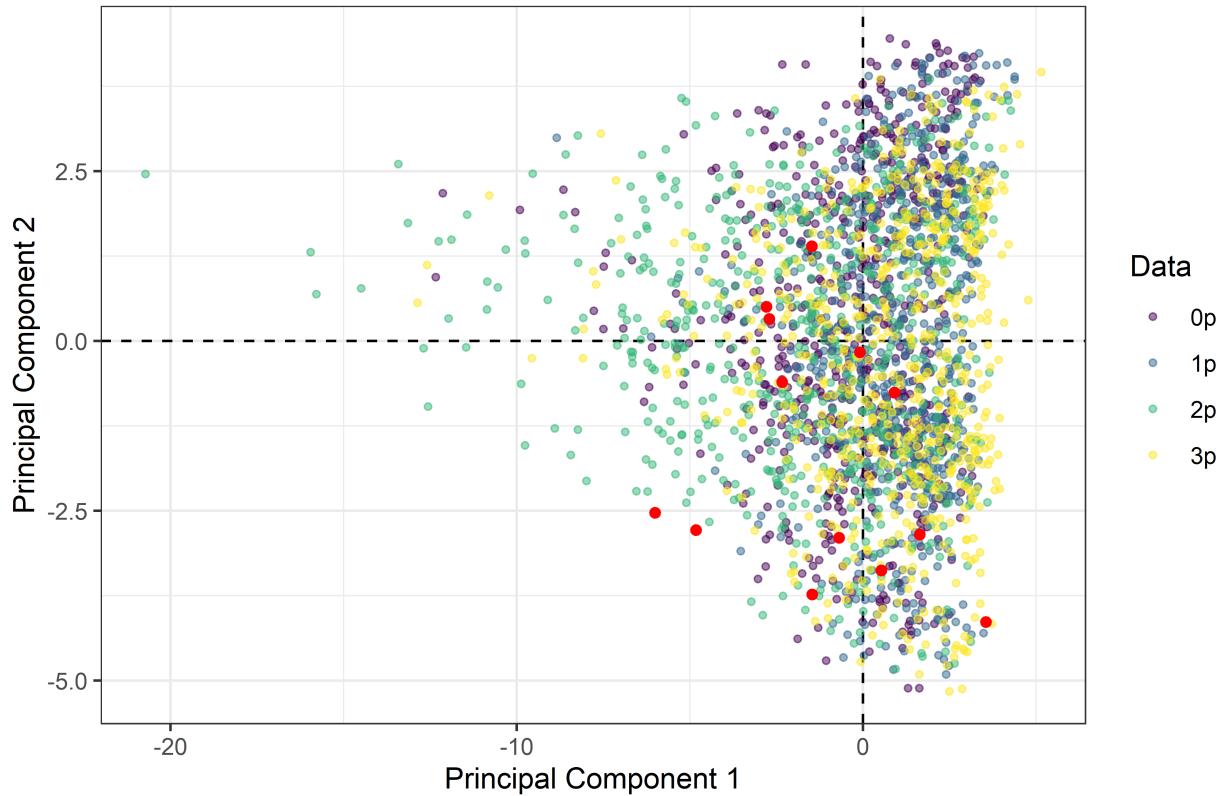
# Create a data frame for ggplot
pca_data <- data.frame(sim = sims, summaries_all_pca$x)

# reorder levels for the legend
pca_data$sim <- factor(pca_data$sim, levels = c("obs", "0p", "1p", "2p", "3p"),
                       labels = c("Buffalo", "0p", "1p", "2p", "3p"))

# Biplot
ggplot() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_vline(xintercept = 0, linetype = "dashed") +
  geom_point(data = pca_data %>% filter(sims != "obs"),
             aes(PC1, PC2, colour = sim),
             size = 1, alpha = 0.5) +
  geom_point(data = pca_data %>% filter(sims == "obs"),
             aes(PC1, PC2),
             colour = "red") +
  scale_colour_viridis_d("Data") +
  ggtitle("PCA Biplot") +
  xlab("Principal Component 1") +
  ylab("Principal Component 2") +
  theme_bw()

```

PCA Biplot



We see that the simulated and observed summaries overlap in the PCA space, and there are few differences between the simulated trajectories from different models, suggesting that the simulations are largely replicating the trends observed in the data, given our set of summary statistics.

We can add a few more plotting elements for further comparison.

```
# Function to compute convex hull and return a data frame
get_hull_coordinates <- function(data, group) {
  points <- data[data$sim == group,]
  hull <- chull(points$PC1, points$PC2)
  return(data.frame(points[hull, ]))
}

# Compute hulls for each group
hulls <- do.call(rbind, lapply(unique(pca_data$sim), get_hull_coordinates, data = pca_data))
```

Add density to the x and y axes.

```
# Generate Viridis colors for three groups
viridis_colors <- viridis::viridis(4)
# Create a named vector of colors for all groups
colors <- setNames(c('red', viridis_colors), c("Buffalo", "0p", "1p", "2p", "3p"))

# Biplot with polygons
pca_plot <- ggplot() +
  geom_point(data = pca_data,
             aes(PC1, PC2, color = sim),
```

```

        alpha = 0.5) +
geom_polygon(data = hulls %>% filter(sim == "Buffalo"),
              aes(x = PC1, y = PC2, group = sim),
              alpha = 0.3, fill = "red") +
theme_bw() +
ggtitle("PCA of all summary statistics") +
xlab(paste0("PC1: ", round(prop_var[1]*100, 1), "% expl. variance")) +
ylab(paste0("PC2: ", round(prop_var[2]*100, 1), "% expl. variance")) +
scale_color_manual("Data", values = colors) +
scale_y_continuous(limits = c(-7.5, 5)) +
geom_hline(yintercept = 0, linetype = "dashed") +
geom_vline(xintercept = 0, linetype = "dashed") +
theme(legend.position = "bottom")

# pca_plot

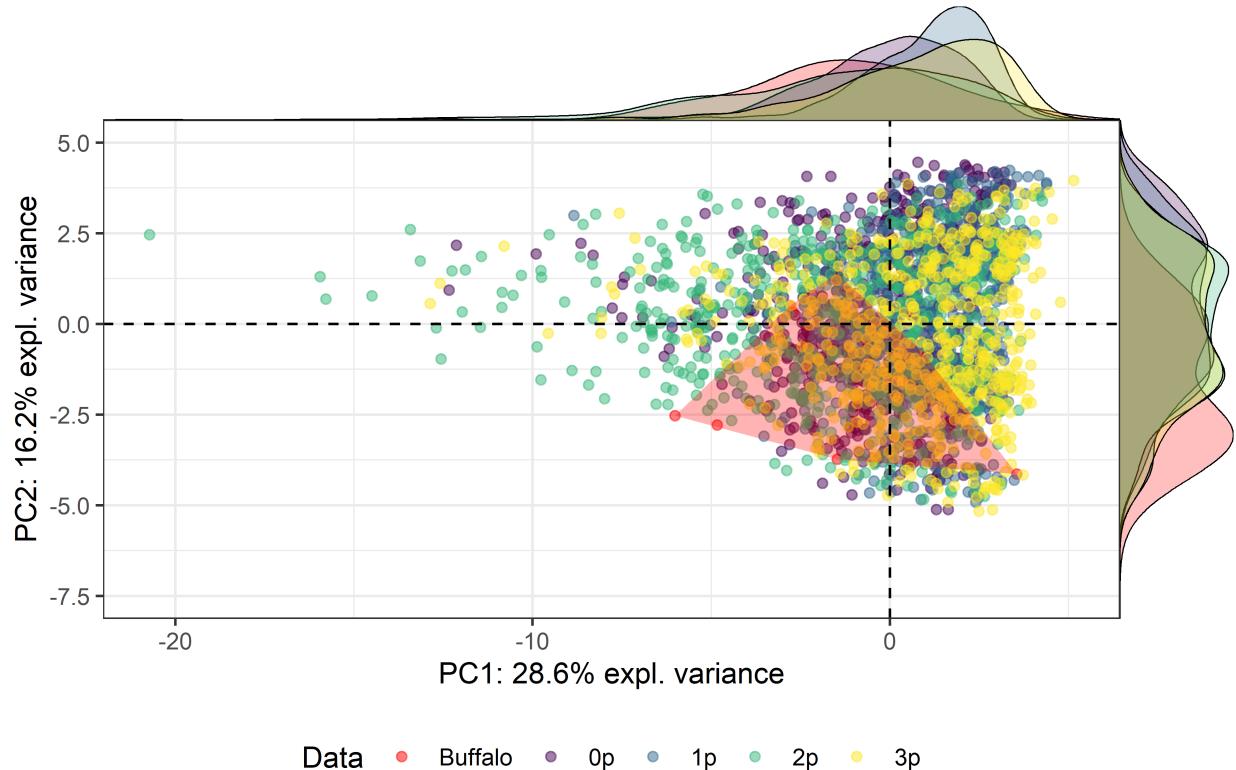
# Add marginal density plots
xdens <-
axis_canvas(pca_plot, axis = "x") +
geom_density(data = pca_data, aes(x = PC1, fill = sim), size = 0.25, alpha = 0.25) +
scale_fill_manual(values = colors)

ydens <-
axis_canvas(pca_plot, axis = "y", coord_flip = TRUE) +
geom_density(data = pca_data, aes(x = PC2, fill = sim), size = 0.25, alpha = 0.25) +
scale_fill_manual(values = colors) +
coord_flip()

pca_plot %>%
insert_xaxis_grob(xdens, grid::unit(1.5, "cm"), position = "top") %>%
insert_yaxis_grob(ydens, grid::unit(1.5, "cm"), position = "right") %>%
ggdraw()

```

PCA of all summary statistics



```
# save the plot using ggplot
ggsave(paste0("outputs/plots/manuscript_figs/memALL_PCA_all_summaries_",
              Sys.Date(), ".png"),
       width=150, height=120, units="mm", dpi = 600)
```

References

- Abrahms, Briana, Dana P Seidel, Eric Dougherty, Elliott L Hazen, Steven J Bograd, Alan M Wilson, J Weldon McNutt, et al. 2017. “Suite of Simple Metrics Reveals Common Movement Syndromes Across Vertebrate Taxa.” *Movement Ecology* 5 (1): 12. <https://doi.org/10.1186/s40462-017-0104-2>.
- Kays, Roland, Ben Hirsch, Damien Caillaud, Rafael Mares, Shauhin Alavi, Rasmus Worsøe Havmøller, and Margaret Crofoot. 2023. “Multi-Scale Movement Syndromes for Comparative Analyses of Animal Movement Patterns.” *Movement Ecology* 11 (1): 61. <https://doi.org/10.1186/s40462-022-00365-y>.

Session info

```
sessionInfo()

## R version 4.2.1 (2022-06-23 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_New Zealand.utf8  LC_CTYPE=English_New Zealand.utf8    LC_MONETARY=English_New Zealand.UTF-8
## [4] LC_NUMERIC=C                           LC_TIME=English_New Zealand.utf8
```

```

## [4] LC_NUMERIC=C                                LC_TIME=English_New Zealand.utf8
##
## attached base packages:
## [1] stats      graphics   grDevices  utils      datasets   methods    base
##
## other attached packages:
## [1] cowplot_1.1.1      ggExtra_0.10.1     ggh4x_0.2.6      Rfast_2.0.7      RcppZiggurat_0.1
## [6] formatR_1.14       scales_1.2.1      glmmTMB_1.1.8    clogitL1_1.5     Rcpp_1.0.10
## [11] ecospat_3.5        TwoStepCLogit_1.2.5 survival_3.5-5    viridis_0.6.2     viridisLite_0.4
## [16] matrixStats_1.0.0   patchwork_1.1.2    ggpubr_0.6.0     adehabitatHR_0.4.21 adehabitatLT_0.1
## [21] CircStats_0.2-6    boot_1.3-28.1    MASS_7.3-59      adehabitatMA_0.3.16 ade4_1.7-22
## [26] sp_1.6-0          ks_1.14.0       beepr_1.3       tictoc_1.2       terra_1.7-23
## [31] amt_0.2.1.0       lubridate_1.9.2  forcats_1.0.0    stringr_1.5.0    dplyr_1.1.2
## [36] purrrr_1.0.1      readr_2.1.4      tidyverse_2.0.0   tibble_3.2.1     ggplot2_3.4.2
## [41] tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
## [1] utf8_1.2.3          tidyselect_1.2.0   lme4_1.1-32      htmlwidgets_1.6.2
## [5] grid_4.2.1          pROC_1.18.0       munsell_0.5.0    codetools_0.2-19
## [9] ragg_1.2.5          units_0.8-1       miniUI_0.1.1.1   withr_2.5.0
## [13] audio_0.1-10       colorspace_2.1-0   highr_0.10      knitr_1.42
## [17] rstudioapi_0.14     ggsignif_0.6.4     Rdpack_2.4       labeling_0.4.2
## [21] emmeans_1.8.5       TeachingDemos_2.12 bit64_4.0.5     farver_2.1.1
## [25] coda_0.19-4        vctrs_0.6.2       generics_0.1.3   TH.data_1.1-2
## [29] circular_0.4-95    xfun_0.39        timechange_0.2.0 randomForest_4.7-1.1
## [33] R6_2.5.1          isoband_0.2.7     cachem_1.0.7    reshape_0.8.9
## [37] promises_1.2.0.1   vroom_1.6.1      multcomp_1.4-23 nnet_7.3-18
## [41] gtable_0.3.3       mda_0.5-3       sandwich_3.0-2   rlang_1.1.0
## [45] systemfonts_1.0.4  splines_4.2.1    rstatix_0.7.2    TMB_1.9.10
## [49] earth_5.3.2       broom_1.0.4      checkmate_2.1.0  biomod2_4.2-2
## [53] yaml_2.3.7         reshape2_1.4.4   abind_1.4-5     backports_1.4.1
## [57] httpuv_1.6.9       Hmisc_5.0-1      tools_4.2.1     nabor_0.5.0
## [61] ellipsis_0.3.2     raster_3.6-20   jquerylib_0.1.4 RColorBrewer_1.1-3
## [65] proxy_0.4-27       plyr_1.8.8       base64enc_0.1-3 classInt_0.4-9
## [69] rpart_4.1.19       zoo_1.8-12      cluster_2.1.4   tinytex_0.48
## [73] magrittr_2.0.3     data.table_1.14.8 mvtnorm_1.1-3   fitdistrplus_1.1-8
## [77] mime_0.12          hms_1.1.3       evaluate_0.20   xtable_1.8-4
## [81] mclust_6.0.0       gridExtra_2.3   compiler_4.2.1  KernSmooth_2.23-20
## [85] crayon_1.5.2       minqa_1.2.5     htmltools_0.5.5 later_1.3.0
## [89] mgcv_1.8-42        tzdb_0.3.0      Formula_1.2-5  DBI_1.1.3
## [93] sf_1.0-12          Matrix_1.6-5    car_3.1-2      permute_0.9-7
## [97] cli_3.6.1          rbibutils_2.2.13 parallel_4.2.1  pkgconfig_2.0.3
## [101] numDeriv_2016.8-1.1 foreign_0.8-84  foreach_1.5.2   bslib_0.4.2
## [105] estimability_1.4.1 plotmo_3.6.2    digest_0.6.31  pracma_2.4.2
## [109] vegan_2.6-4        rmarkdown_2.21   htmlTable_2.4.1 PresenceAbsence_1.1.11
## [113] shiny_1.7.4        gtools_3.9.4    nloptr_2.0.3   lifecycle_1.0.3
## [117] nlme_3.1-162      jsonlite_1.8.4  carData_3.0-5  fansi_1.0.4
## [121] pillar_1.9.0       lattice_0.21-8 fastmap_1.1.1  plotrix_3.8-2
## [125] glue_1.6.2         gbm_2.1.8.1    iterators_1.0.14 bit_4.0.5
## [129] class_7.3-21      stringi_1.7.12 sass_0.4.5    maxnet_0.1.4
## [133] textshaping_0.3.6 poibin_1.5     e1071_1.7-13  ape_5.7-1

```