Wind support analysis

Hester Bronnvik

2024-01-31

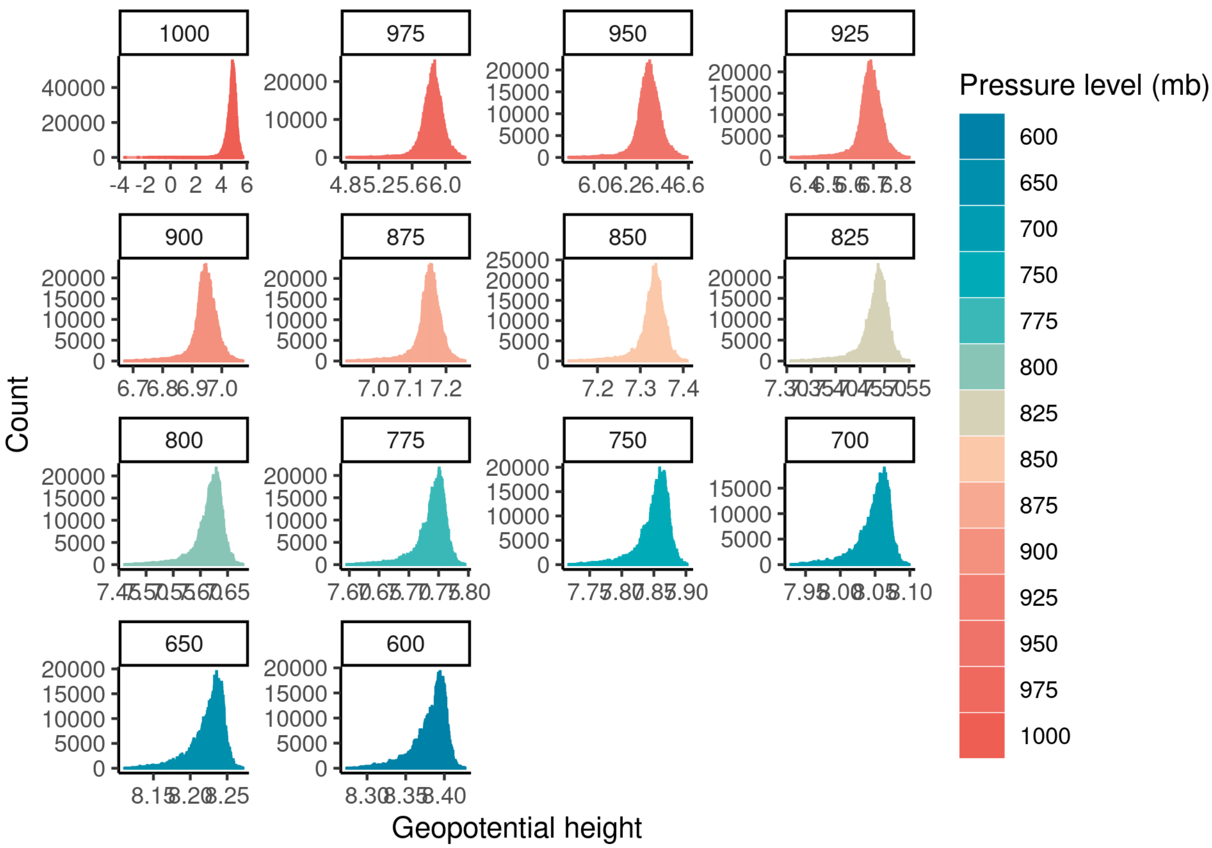
Wind support is an important energy resource for birds (1-4). Wind has been linked to the speed and routes of migrating white storks (*Ciconia ciconia*) (5,6). Thus, we investigated wind support as a predictor of route selection. We did not find it to be a significant predictor, and found neither a signature of selection nor of change over time. We excluded wind support from our final models, but present it here because wind is often considered to be important.

The first step is to determine which wind estimates to use. All of the wind estimates in these weather models are highly correlated. In addition, the height above ellipsoid values sent by the tags are error prone, especially when the tag is cold, and do not necessarily reflect the true height of the bird. For these two reasons, we felt it was sufficient to use only the wind estimate at the pressure level around the approximate average flight height of a migrating stork.

```
# libraries:
library(terra)
library(glmmTMB)
library(tidyverse)
theme_set(theme_classic()+
            theme(axis.text = element_text(color = "black", size = 12),
                  text = element_text(size = 15)))

# functions:
wind_support <- function(u,v,heading) {
  angle <- atan2(u,v) - heading/180*pi
  return(cos(angle) * sqrt(u*u+v*v))
}
cross_wind <- function(u,v,heading) {
  angle <- atan2(u,v) - heading/180*pi
  return(sin(angle) * sqrt(u*u+v*v))
}
wind_speed <- function(u,v) {
  return(sqrt(u*u+v*v))
}

## Determine which level of wind to use by looking at the overall flight heights
# of white storks

# the annotated data file with the uplift and conspecific density values
a_data <- readRDS("/home/hbronnvik/Documents/storkSSFs/annotations/HR_030923.rds")

# all of the full, raw data downloaded from Movebank
files <- list.files("/home/hbronnvik/Documents/storkSSFs/full_data", pattern = ".rds", full.names = T)

# the geoid heights from EGM2008 https://www.agisoft.com/downloads/geoids/
```

```r
egm <- terra::rast("/home/hbronnvik/Documents/storkSSFs/us_nga_egm2008_1.tif")

# take the geoid height (AKA geoid undulation) out of the
# height above ellipsoid to get height above geoid (AKA msl)

# start with a simple collection of all the heights above ellipsoid
# there can be a lot of error in these readings, so we remove the very large or very
# small values (although error exists in the plausible ones as well)
heights <- lapply(files, function(file){
  data <- readRDS(file)
  data <- data %>%
    # take out the bursts
    mutate(td = as.numeric(difftime(timestamp, lag(timestamp), units = "secs"))) %>%
    filter(td >= 300) %>%
    mutate(seq15 = round_date(timestamp, unit = "15 minutes")) %>%
    group_by(seq15) %>%
    slice(1) %>%
    ungroup() %>%
    dplyr::select(-seq15, -td) %>%
    mutate(distance = geosphere::distVincentyEllipsoid(cbind(location.long, location.lat), cbind(lag(lo
           timediff = as.numeric(difftime(timestamp, lag(timestamp), units = "secs")),
           ground_speed_15 = distance/timediff) %>%
    # use only locations that are plausible and in-flight
    filter(ground_speed_15 < 50 & ground_speed_15 > 2)
  hae <- data %>%
    filter(height.above.ellipsoid > -100 & height.above.ellipsoid < 10000) %>%
    dplyr::select(timestamp, location.long, location.lat, height.above.ellipsoid)
  hae
})
heights <- data.table::rbindlist(heights)
# next, extract the geoid undulations at each of the locations that has HAE
heights$geoid_height <- terra::extract(egm, vect(heights,
                                                 geom = c("location.long",
                                                          "location.lat")))[ ,2]

# add on the height above mean sea level
heights$ha_msl <- heights$height.above.ellipsoid - heights$geoid_height

# find the average flight height above mean sea level,
# but try to use only the trustworthy ones by not allowing the birds to fly above the boundary layer
heights <- heights %>%
  filter(ha_msl < max(na.omit(a_data$blh)))

mean(heights$ha_msl)

## [1] 683.2945

# when we annotated the location data with uplift velocity, we had the geopotential height
# for each pressure level at each location (columns 15 to 28)
# take the mean height of each pressure level across locations in the annotations
# and compare it to the mean flight height of the birds
log_heights <- lapply(15:28, function(x){
  level_h <- a_data[, x]
  level <- colnames(a_data)[x]
```

```
  info <- data.frame(level = level, mean = mean(na.omit(log(level_h))), sd = sd(na.omit(log(level_h))))
}) %>% reduce(rbind) %>% arrange(as.numeric(level))

# look at the heights of the levels
colfunc <- colorRampPalette(c("#0081A7", "#0098b0", "#00AFB9", "#7fc4b8", "#FED9B7", "#f7a58f", "#f27e7
pd <- a_data[, c(15:28)]
pd <- pd %>%
  pivot_longer(names_to = "level", values_to = "geoH", cols = names(pd)) %>%
  mutate(level = as.numeric(level),
         # the levels are skewed simply because low pressure exists at higher altitudes
         # we log normalize to take the mean
         log_geoH = log(geoH)) %>%
  arrange(level)
ggplot(pd, aes(log_geoH, color = as.factor(level), fill = as.factor(level))) +
    geom_histogram(bins = 1000) +
    labs(x = "Geopotential height", y = "Count") +
    scale_color_manual("Pressure level (mb)", values = colfunc(14)) +
    scale_fill_manual("Pressure level (mb)", values = colfunc(14)) +
    theme_classic() +
    facet_wrap(~factor(level, levels = c("1000", "975", "950", "925", "900", "875", "850", "825", "800"
```




```
## Look at the mean log heights of the pressure levels
log_heights
```

```
##    level     mean         sd
## 1    600 8.382277 0.02060264
## 2    650 8.224674 0.02117622
```

```
## 3    700 8.049903 0.02191177
## 4    750 7.851134 0.02292309
## 5    775 7.739583 0.02364638
## 6    800 7.617418 0.02466032
## 7    825 7.481895 0.02617267
## 8    850 7.329089 0.02852916
## 9    875 7.153118 0.03231931
## 10   900 6.944613 0.03860282
## 11   925 6.687230 0.04953549
## 12   950 6.348549 0.07050391
## 13   975 5.846427 0.12154267
## 14  1000 4.815541 0.37841110
```

```
## Finally, compare the flight heights to the pressure levels
mean(na.omit(log(heights$ha_msl)))
```

```
## [1] 6.026083
```

```
sd(na.omit(log(heights$ha_msl)))
```

```
## [1] 1.328603
```

```
# read out files to annotate with the wind data from Movebank Env-DATA service at the given level
# https://www.movebank.org/cms/movebank-content/env-data
wind_file <- a_data %>%
  dplyr::select(timestamp, long, lat, individual.id) %>%
  mutate(timestamp = paste0(timestamp, ".000"),
         group = c(rep(c("a"), times = n()/4),
                   rep(c("b"), times = n()/4),
                   rep(c("c"), times = n()/4),
                   rep(c("d"), times = n()/4))) %>%
  rename("location-long" = long,
         "location-lat" = lat)
```

We found that the mean log height of the 950 mb pressure level for our data was 6.35 $\pm$ 0.07 sd and the mean log flight height of storks was approximately 6.02 $\pm$ 1.33 sd.

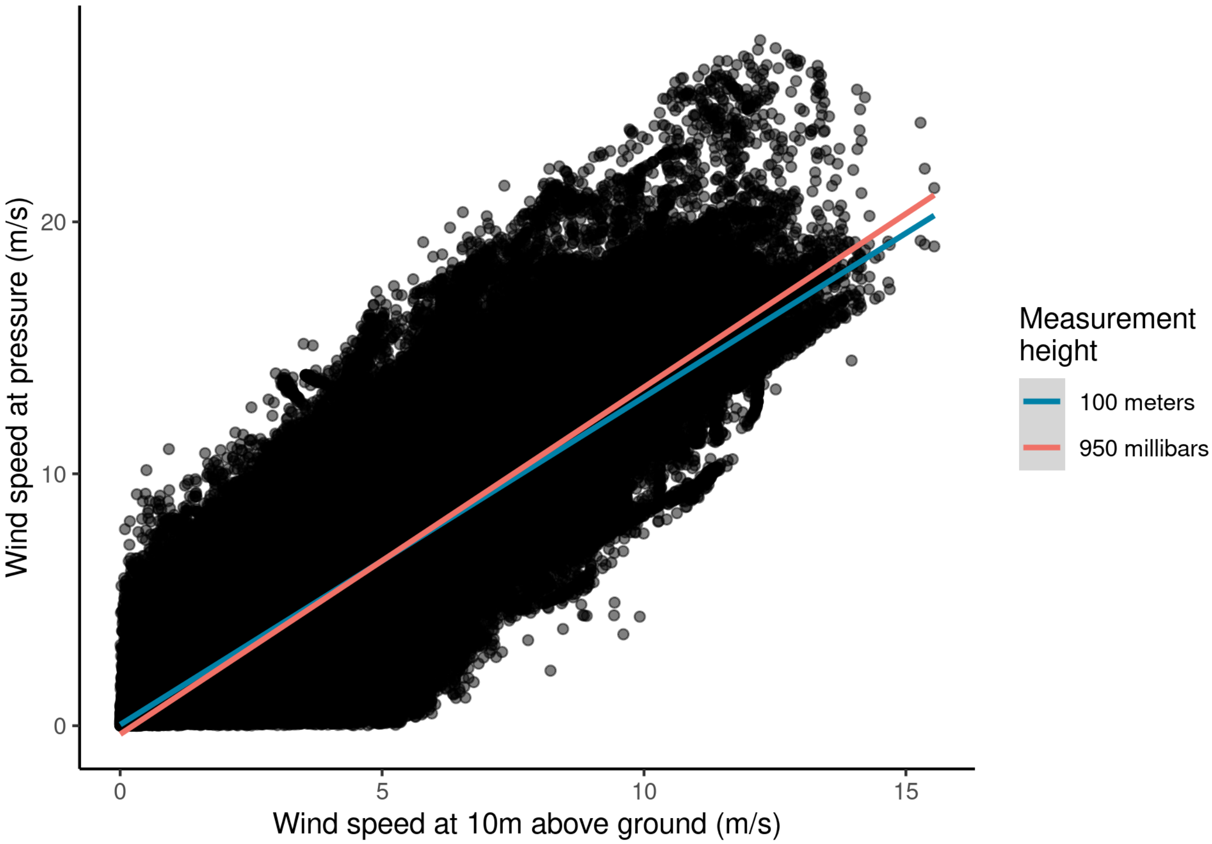
We calculated wind support using the north/south and east/west components of the wind (7) at 950 millibars of pressure.

```
## Explore the wind data to look for patterns

# after annotation, read in the files from Movebank Env-DATA
wind_files <- list.files("/home/hbronnvik/Documents/storkSSFs/ecmwf/winds", pattern = ".csv", full.names

# make the column names convenient
winds <- lapply(wind_files, read.csv) %>% reduce(rbind)
winds$X <- NULL
colnames(winds)
```

```
##  [1] "timestamp"
##  [2] "location.long"
##  [3] "location.lat"
##  [4] "individual.id"
##  [5] "group"
##  [6] "ECMWF.ERA5.PL.U.Wind"
##  [7] "ECMWF.ERA5.PL.V.Wind"
##  [8] "ECMWF.ERA5.SL.Wind..10.m.above.Ground.U.Component."
```

```
##  [9] "ECMWF.ERA5.SL.Wind..100.m.above.Ground.U.Component."
## [10] "ECMWF.ERA5.SL.Wind..100.m.above.Ground.V.Component."
## [11] "ECMWF.ERA5.SL.Wind..10.m.above.Ground.V.Component."
```

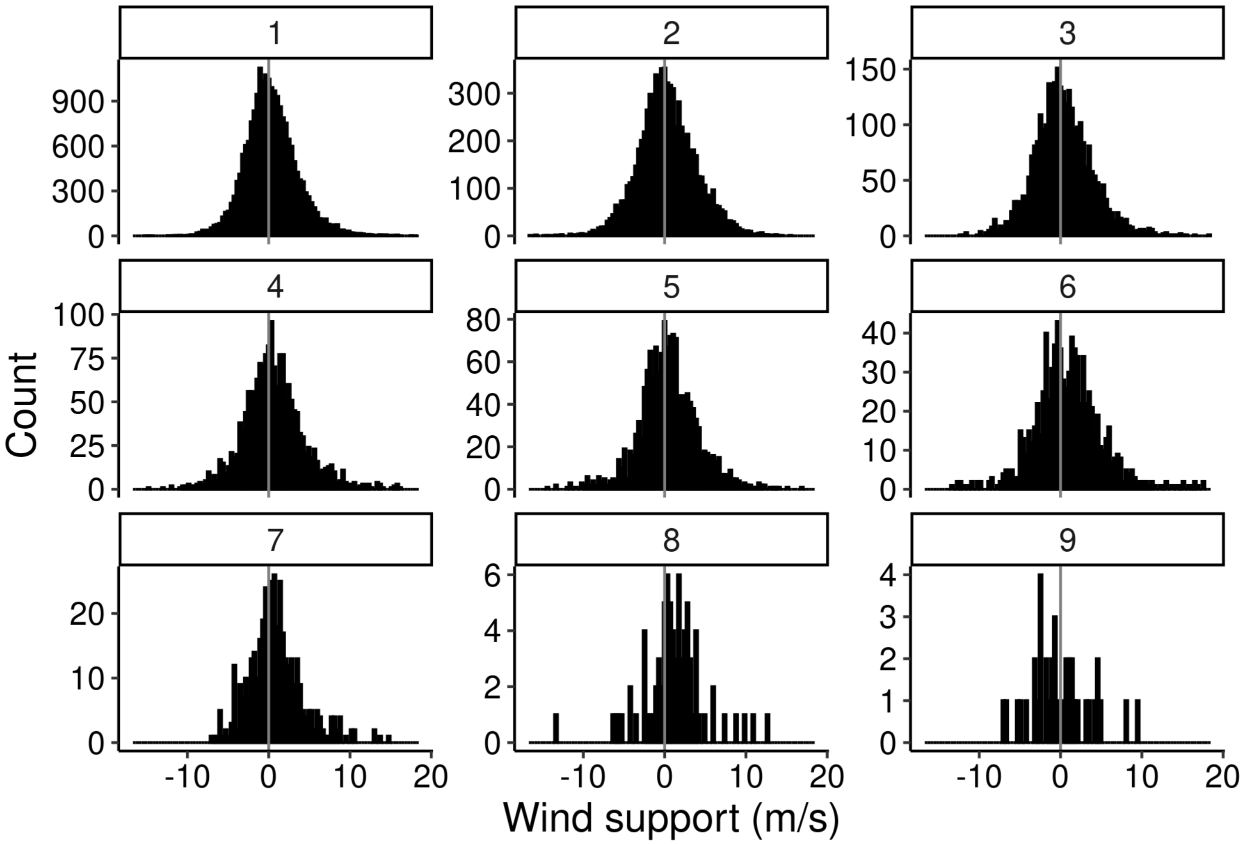
```
colnames(winds)[2:3] <- c("long", "lat")
colnames(winds)[6:11] <- c("u_950", "v_950", "u_10m", "u_100m", "v_100m", "v_10m")
# turn the stamp back to an R format
winds <- winds %>%
  mutate(timestamp = as.POSIXct(timestamp, tz = "UTC"))

# also make the long/lat format match the a_data so that the join will recognize them
a_data <- a_data %>%
  mutate(long = as.character(long),
         lat = as.character(lat)) %>%
  mutate(long = as.numeric(long),
         lat = as.numeric(lat))

# add the wind data to the steps with their social density and w*
a_data <- full_join(a_data, winds) %>%
  mutate(cross_wind = cross_wind(u_950, v_950, heading),
         wind_support = wind_support(u_950, v_950, heading),
         wind_speed = wind_speed(u_950, v_950),
         wind_speed_100m = wind_speed(u_100m, v_100m),
         wind_speed_ground = wind_speed(u_10m, v_10m)) %>%
  rename(ud_pdf = UD_PDF,
         migrations = journey_number) %>%
  mutate(season = ifelse(grepl("fall", track), "post", "pre"))

# double-check our understanding that wind estimates from these weather
# models are highly correlated across pressure levels
check <- a_data %>%
  dplyr::select(wind_speed_ground, wind_speed, wind_speed_100m) %>%
  # reduce the number of data points to make this faster
  group_by(wind_speed_ground) %>%
  slice(1) %>%
  ungroup() %>%
  rename("950 millibars" = wind_speed,
         "100 meters" = wind_speed_100m) %>%
  pivot_longer(cols = c("950 millibars", "100 meters"), names_to = "wind_level",
               values_to = "wind_speed")

ggplot(check, aes(wind_speed_ground, wind_speed)) +
  geom_point(alpha = .5) +
  geom_smooth(aes(group = wind_level, color = wind_level), method = "lm") +
  scale_color_manual(values = c("#0081A7", "#F07167")) +
  labs(x = "Wind speed at 10m above ground (m/s)",
       y = "Wind speed at pressure (m/s)",
       color = "Measurement \nheight") +
  theme_classic()
```



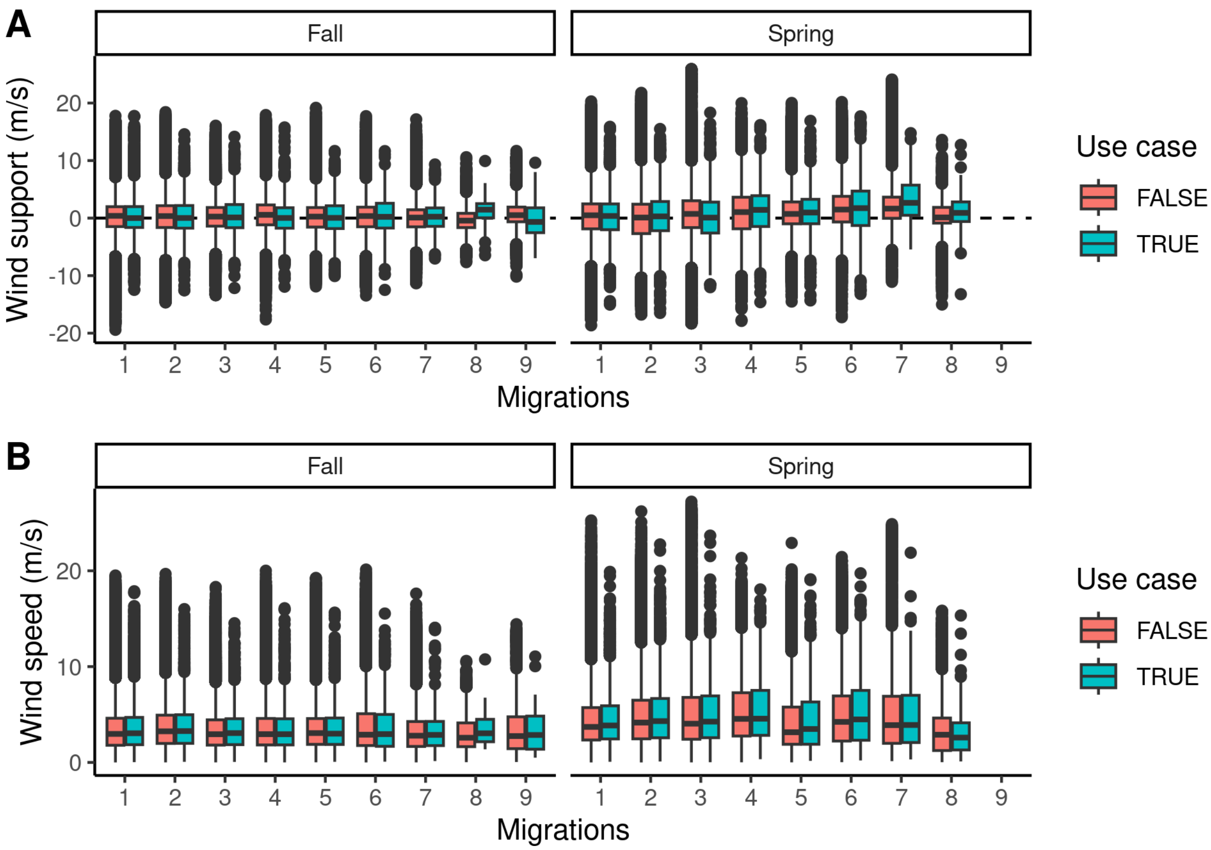

```
# such strong correlation justifies the use of one pressure level rather than
# interpolating to an approximation of the bird's height
a_data %>%
  dplyr::select(c(wind_speed_ground, wind_speed, wind_speed_100m)) %>%
  corrr::correlate()
```

```
## # A tibble: 3 x 4
##   term              wind_speed_ground wind_speed wind_speed_100m
##   <chr>                         <dbl>      <dbl>           <dbl>
## 1 wind_speed_ground            NA          0.900           0.978
## 2 wind_speed                    0.900     NA               0.912
## 3 wind_speed_100m               0.978      0.912          NA
```

```
# look at wind support distributions
ggplot(a_data %>% filter(used == 1), aes(wind_support)) +
  geom_histogram(bins = 100, color = "black") +
  geom_vline(xintercept = 0, color = "gray50") +
  facet_wrap(~migrations, scales = "free_y") +
  labs(x = "Wind support (m/s)", y = "Count")
```




```
# universal facet labels
fac_labs <- c("Fall", "Spring")
names(fac_labs) <- c("post", "pre")
support <- ggplot(a_data %>% group_by(wind_support) %>% slice(1) %>% ungroup(), aes(as.factor(migrations
  geom_hline(yintercept = 0, lty = 2) +
  geom_boxplot() +
  theme_classic() +
  labs(x = "Migrations", y = "Wind support (m/s)", fill = "Use case") +
  scale_color_manual(values = c("#0081A7", "#F07167")) +
  theme_classic() +
  theme(text = element_text(color = "black")) +
  facet_wrap(~season, labeller = labeller(season = fac_labs))
# look at wind speed distributions
speeds <- ggplot(a_data %>% group_by(wind_speed) %>% slice(1) %>% ungroup(), aes(as.factor(migrations),
  geom_boxplot() +
  theme_classic() +
  labs(x = "Migrations", y = "Wind speed (m/s)", fill = "Use case") +
  scale_color_manual(values = c("#0081A7", "#F07167")) +
  theme_classic() +
  theme(text = element_text(color = "black")) +
  facet_wrap(~season, labeller = labeller(season = fac_labs))

cowplot::plot_grid(support, speeds, labels = c("A", "B"), ncol = 1,
          align = 'v', axis = 'l')
```



There is no sign in the distributions that we should expect selection (i.e. no difference between used and available) or to see a change over time (no shift in speeds given age). Still, we can check quantitatively by including wind support in our model.

```
## Model the data and check for significance

# scale the predictors
a_data <- a_data %>%
  mutate_at(c("migrations", "w_star","ud_pdf", "step_length",
              "turning_angle", "blh", "wind_support"),
            list(z = ~(scale(.)))) %>%
  mutate(sqrt_ud = sqrt(ud_pdf),
         sqrt_ud_z = scale(sqrt_ud))

# run the model once for each season
mods <- lapply(split(a_data, a_data$season), function(df){
  df <- df %>%
    mutate(stratum_ID = as.factor(stratum),
           individual.id = as.numeric(individual.id))
  # build the formula
  TMB_struc <- glmmTMB(used ~ -1 + wind_support_z + sqrt_ud_z + w_star_z +
                         migrations_z + step_length_z + turning_angle_z +
                       (1|stratum_ID) +
                       (0 + wind_support_z | individual.id) +
                       (0 + sqrt_ud_z | individual.id) +
                       (0 + w_star_z | individual.id) +
                       (0 + migrations_z | individual.id),
                     family = poisson,
```

```r
                     data = df, doFit = FALSE,
                     map = list(theta = factor(c(NA,1:4))),
                     start = list(theta = c(log(1e3),0,0,0,0)))
  TMB_M <- glmmTMB:::fitTMB(TMB_struc)
})
```

```r
summary(mods[[1]]) # the fall
```

```
##  Family: poisson  ( log )
## Formula:
## used ~ -1 + wind_support_z + sqrt_ud_z + w_star_z + migrations_z +
##     step_length_z + turning_angle_z + (1 | stratum_ID) + (0 +
##     wind_support_z | individual.id) + (0 + sqrt_ud_z | individual.id) +
##     (0 + w_star_z | individual.id) + (0 + migrations_z | individual.id)
## Data: df
##
##       AIC       BIC    logLik  deviance  df.resid
##  444611.1  444738.4 -222295.6  444591.1   2482870
##
## Random effects:
##
## Conditional model:
##  Groups          Name            Variance  Std.Dev.
##  stratum_ID      (Intercept)     1.000e+06 1.000e+03
##  individual.id   wind_support_z 1.986e+00 1.409e+00
##  individual.id.1 sqrt_ud_z       5.612e+02 2.369e+01
##  individual.id.2 w_star_z        3.757e-01 6.129e-01
##  individual.id.3 migrations_z    9.842e-05 9.921e-03
## Number of obs: 2482880, groups:  stratum_ID, 24874; individual.id, 158
##
## Conditional model:
##                  Estimate Std. Error z value Pr(>|z|)
## wind_support_z   -0.172768   0.115080   -1.50    0.133
## sqrt_ud_z        12.841658   1.924152    6.67 2.49e-11 ***
## w_star_z          0.970650   0.065934   14.72  < 2e-16 ***
## migrations_z    -42.387177   0.591782  -71.63  < 2e-16 ***
## step_length_z     0.127719   0.008166   15.64  < 2e-16 ***
## turning_angle_z -12.220437   0.088726 -137.73  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(mods[[2]]) # the spring
```

```
##  Family: poisson  ( log )
## Formula:
## used ~ -1 + wind_support_z + sqrt_ud_z + w_star_z + migrations_z +
##     step_length_z + turning_angle_z + (1 | stratum_ID) + (0 +
##     wind_support_z | individual.id) + (0 + sqrt_ud_z | individual.id) +
##     (0 + w_star_z | individual.id) + (0 + migrations_z | individual.id)
## Data: df
##
##       AIC       BIC    logLik  deviance  df.resid
##  200398.9  200518.3 -100189.4  200378.9   1135033
##
## Random effects:
```

```
##
## Conditional model:
##  Groups          Name           Variance  Std.Dev.
##  stratum_ID      (Intercept)    1.000e+06 1000.0000
##  individual.id   wind_support_z 2.351e+00    1.5332
##  individual.id.1 sqrt_ud_z      1.794e+01    4.2361
##  individual.id.2 w_star_z       8.783e-01    0.9372
##  individual.id.3 migrations_z   2.017e-04    0.0142
## Number of obs: 1135043, groups:  stratum_ID, 11374; individual.id, 72
##
## Conditional model:
##                  Estimate Std. Error z value Pr(>|z|)
## wind_support_z   -0.18158    0.18806   -0.97    0.334
## sqrt_ud_z         3.67355    0.53651    6.85 7.53e-12 ***
## w_star_z          0.54272    0.13390    4.05 5.05e-05 ***
## migrations_z    -41.15466    0.78910  -52.15  < 2e-16 ***
## step_length_z     0.25766    0.01094   23.55  < 2e-16 ***
## turning_angle_z -11.62879    0.12299  -94.55  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As we could expect from the exploratory plots, we find no evidence of selection on wind support in either season.

References

1. Liechti F. 2006 Birds: Blowin'by the wind? Journal of Ornithology 147, 202–211.
2. Vansteelant W, Bouten W, Klaassen R, Koks B, Schlaich A, Van Diermen J, Van Loon E, Shamoun-Baranes J. 2015 Regional and seasonal flight speeds of soaring migrants and the role of weather conditions at hourly and daily scales. Journal of Avian Biology 46, 25–39.
3. Nourani E et al. 2021 The interplay of wind and uplift facilitates over-water flight in facultative soaring birds. Proceedings of the Royal Society B 288, 20211603.
4. Brønnvik H, Safi K, Vansteelant WM, Byholm P, Nourani E. 2022 Experience does not change the importance of wind support for migratory route selection by a soaring bird. Royal Society Open Science 9, 220746.
5. Becciu P et al. 2020 Causes and consequences of facultative sea crossing in a soaring migrant. Functional Ecology 34, 840–852.
6. Bontekoe ID, Hilgartner R, Fiedler W, Flack A. 2023 The price of being late: Short-and long-term consequences of a delayed migration timing. Proceedings of the Royal Society B 290, 20231268.
7. Safi K et al. 2013 Flying with the wind: Scale dependency of speed and direction measurements in modelling wind support in avian flight. Movement Ecology 1, 1–13.